UNIVERSIDAD DEL VALLE DE GUATEMALA

Facultad de Ingeniería

Algoritmos inteligentes para reconocimiento de patrones de comportamiento transaccionales

Trabajo de graduación en modalidad de Megaproyecto presentado por:
Ana Lucía Paiz Gómez
para optar al grado académico de Licenciada en Ingeniería Industrial;
Berny Osberto Ixcayau Coguox
para optar al grado académico de Licenciado en Ingeniería en
Ciencia de la Administración;
Diego Alejandro Enriquez Rodríguez,
Joel Alejandro Cantoral Schwartz y
Melinton Antonio Navas González
para optar por el grado académico de Licenciados en Ingeniería en Ciencias de la Computación y Tecnologías de la Información

Guatemala,
2014

Algoritmos inteligentes para reconocimiento de patrones de comportamiento transaccionales

UNIVERSIDAD DEL VALLE DE GUATEMALA

Facultad de Ingeniería

Algoritmos inteligentes para reconocimiento de patrones de comportamiento transaccionales

Trabajo de graduación en modalidad de Megaproyecto presentado por:

Ana Lucía Paiz Gómez

para optar al grado académico de Licenciada en Ingeniería Industrial;

Berny Osberto Ixcayau Coguox

para optar al grado académico de Licenciado en Ingeniería en Ciencia de la Administración;

Diego Alejandro Enriquez Rodríguez,

Joel Alejandro Cantoral Schwartz y

Melinton Antonio Navas González

para optar por el grado académico de Licenciados en Ingeniería en Ciencias de la Computación y Tecnologías de la Información

Guatemala,

2014

Vo. Bo.:

(f) _____
Ing. Lynette García

Tribunal Examinador:

(f) _____
MSc. Douglas Barrios

(f) _____
MSc. Estuardo Sierra

(f) _____
Ing. MBA Celso Crezo

Fecha de aprobación: Guatemala, 25 de noviembre de 2014

CONTENIDO

LISTA DE CUADROS

LISTA DE FIGURAS

RESUMEN

Los avances tecnológicos que poseemos hoy en día, han permitido llevar conceptos que se han definido hace algunas décadas con respecto a la inteligencia artificial, a sistemas inteligentes capaces de tomar decisiones y afirmar o negar una aseveración de acuerdo a un conjunto de datos analizados e interpretados previamente por estos sistemas. La investigación en la detección de transacciones financieras fraudulentas ha sido un área que ha tenido un desarrollo considerable a lo largo del tiempo, es por ello que el presente estudio busca elaborar un algoritmo que modele un sistema inteligente para la detección de fraudes en las transacciones electrónicas. Los algoritmos seleccionados para buscar la solución óptima fueron Redes Neurales, Support Vector Mchines y Redes Bayesianas.

Las redes neuronales están definidas por 9 módulos según los campos seleccionados por transacción. En cada uno de los módulos definidos se varió la función de activación de la capa oculta, la función de activación de la capa de salida, el algoritmo de entrenamiento, el impulso del algoritmo y la tasa de aprendizaje del algoritmo de entrenamiento. Entre los principales resultados se obtuvo que el entrenamiento con el algoritmo de Backpropagation con un porcentaje más alto de falsos negativos que falsos positivos y se alcanzó un 99.18% de efectividad.

La SVM final, tiene un kernel lineal y utiliza el método de descenso de gradiente estocástico para entrenarse. Esta SVM tiene un índice de 18.85% falsos positivos y 17.47% falsos negativos. No se logró disminuir los índices deseados, que actualmente son de 10%.

La red bayesiana consta de seis nodos relacionados entre sí. La hipótesis planteada, buscaba elaborar una red bayesiana que superara el 90% de efectividad, sin embargo, la red presentada únicamente alcanzó un 71.63% de efectividad.

Palabras clave: redes bayesianas, clasificación bayesiana, regla de Bayes, inteligencia artificial, aprendizaje de máquinas, teoría probabilística, detección de fraude.

I. INTRODUCCIÓN

En los últimos años la tendencia a sustituir el efectivo por otro medio de pago ha ido en aumento. Debido a esta situación surge la necesidad de identificar anomalías en las transacciones de un sistema bancario. Con el megaproyecto "Algoritmos inteligentes para reconocimiento de patrones de comportamiento transaccionales" se logró proponer un enfoque más efectivo de detección de fraude transaccional para una empresa con experiencia en el medio. Con esto no solo se propuso una mejora para esta empresa sino para los usuarios de tarjetas de crédito y débito de la red bancaria de Guatemala en general, al ser esta empresa una de las principales del país que brinda a distintos bancos del sistema el software de detección de fraude.

La creación de sistemas inteligentes que puedan proporcionar respuestas instantáneas para detección de fraudes es importante, tomando en cuenta la expansión de las transacciones monetarias hacia el comercio electrónico y medios móviles. En estos casos el volumen de transacciones y la velocidad de atención resultan factores decisivos para el sistema. Además de los beneficios listados se logró abrir una línea de investigación en la Universidad del Valle de Guatemala en la gama de reconocimiento de patrones y análisis de datos utilizando Inteligencia Artificial.

El proyecto consistió en el diseño y programación de tres algoritmos con la capacidad de detectar, con el menor margen de error posible, transacciones monetarias fraudulentas efectuadas con tarjetas de crédito o débito por usuarios de la red bancaria de Guatemala, e identificar patrones de fraude por medio de Clustering. Para esto se utilizó la estructura de transacciones y los datos de prueba que proporcionó la empresa, los cuales fueron el punto de partida para la elaboración del proyecto. El producto final entregado fue un conjunto de tres algoritmos basados en técnicas de aprendizaje de máquinas (Redes Neuronales, Redes Bayesianas y Máquinas de vectores de soporte). Estos modelos permiten ponderar una transacción monetaria recibida por un canal electrónico para verificar su autenticidad, basándose en el historial de transacciones que pertenece al usuario de una tarjeta. Adicionalmente se entregó un reporte de patrones de fraude detectados por medio de Clustering y respaldado por los resultados de análisis de árboles de decisión y regresión lineal. Para los algoritmos se entregó la documentación, diagramación y modelado respectivos como producto del diseño del proyecto. Con base en los resultados finales que cada modelo obtuvo se seleccionó al algoritmo idóneo que se recomienda utilizar. El algoritmo seleccionado fue Redes Neurales, al detectar 99.18% de transacciones correctamente.

Con base en este algoritmo se elaboró una propuesta de modelo de negocio teniendo en cuenta escenarios. Donde se logró identificar el modelo más rentable en el que se potencializarán los beneficios para todos los eslabones de la cadena de valor.

II. OBJETIVOS

A. General

Disminuir el índice de falsos positivos y falsos negativos de los sistemas de detección de transacciones fraudulentas de la empresa con la que se trabajó, aplicando algoritmos inteligentes y reconocimiento de patrones transaccionales.

B. Específicos

- Proponer un modelo de negocio con base en un costeo por escenarios y administrar el Proyecto Algoritmos Inteligentes para Reconocimiento de Patrones de Comportamiento Transaccionales por medio de datos de prueba para una empresa con experiencia en el campo.
- Disminuir el índice de falsos positivos y falsos negativos de los sistemas de detección de transacciones fraudulentas de una empresa con experiencia en el campo aplicando redes neuronales.
- Determinar el índice de falsos positivos y falsos negativos de los sistemas de detección de transacciones fraudulentas de una empresa con experiencia en el campo al aplicar algoritmos de Support Vector Machines.
- Determinar la cantidad de falsos positivos y falsos negativos generados por la implementación del modelo de una red bayesiana, utilizando los datos de prueba proporcionados por una empresa con experiencia en el campo.

III. JUSTIFICACIÓN

Entre las razones principales para la ejecución del proyecto se encuentra aumentar la confiabilidad del sistema de detección de fraude de la empresa, mejorando los casos propuestos con anterioridad y obtener así una disminución en los costos invertidos actualmente en el área de detección de transacciones fraudulentas, al mismo tiempo incrementando la calidad del sistema, para aumentar su competitividad a nivel mundial.

La necesidad de identificar anomalías transaccionales en un sistema bancario surge a partir del crecimiento que ha tenido el mercado en medios distintos al efectivo para realizar transacciones monetarias (Chen, T., 2010). En un estudio realizado por la empresa canadiense TSYS en el año 2014 se obtuvo que un 67% de los encuestados, quienes eran mayores de 18 años y poseían tarjetas de débito o crédito, preferían utilizar sus tarjetas en lugar de efectivo, tarjetas afiliadas a comercios o cheques, entre otros; estos resultados se deben a la trazabilidad, seguridad, conveniencia y comodidad que proveen (TSYS, 2014:3). Las principales empresas que manejan estos medios de pago han demostrado que se manejan cantidades grandes de dinero a través de sus procesos de transacciones. Solamente en el año 2013 la empresa Visa reportó ingresos netos equivalentes a 5 billones de dólares (Visa, 2014), en el mismo período la empresa Mastercard reportó 3.2 billones de dólares (Mastercard, 2014).

En la región iberoamericana se estima que más de 60% de las transacciones realizadas en el año 2012 se han efectuado con tarjetas, este valor se ha mantenido sobre todos los otros medios de pago desde el año 2007 (Afi & Tecnocom, 2013:11). Es importante reconocer que en la región latinoamericana hay un factor que mantiene la predominancia del efectivo en el sector, esto es que «...en Latinoamérica no se tiene acceso a los servicios bancarios [...] la población está empleada de manera informal» (Afi & Tecnocom, 2013:37).

En la actualidad, los sistemas inteligentes y en tiempo real de detección de fraude, son factores decisivos para la expansión de las transacciones monetarias hacia el comercio electrónico y medios móviles en donde el volumen de transacciones y la velocidad de atención resultan factores decisivos para el sistema y la cantidad de transacciones se suma a la de cajeros automáticos, puntos de servicios y otras tecnologías bancarias. Según Paypal, en el 2012, el 15% de las operaciones de comercio electrónico se realizaron a través de dispositivos móviles en Brasil y México. En este mismo año se estimó un valor neto de operaciones por medios electrónicos igual a 13.4 billones, que corresponde a un 42.8% más que en el año 2010 (Afi & Tecnocom, 2013:13). Además, Visa ha realizado pasos importantes de transición hacia un mercado móvil al introducir plataformas de dinero móvil de uso inmediato en Febrero del 2013 y un sitio de billetera móvil para facilitar los pagos en línea en Noviembre del 2012; a su vez Mastercard ha implementado un producto similar llamado MasterPass a través de tecnologías como códigos QR o

Comunicaciones de Rango Corto (NFC), ésta tendencia se observa con recurrencia en meses posteriores implementadas por distintos bancos de la región (Afi & Tecnocom, 2013:20).

Por lo anteriormente establecido, se encuentra la necesidad de explorar el campo de desarrollo de sistemas inteligentes que sean capaces de detectar anomalías en transacciones electrónicas. La importancia de este estudio se encuentra no sólo en la investigación y creación de algoritmos que modelen sistemas inteligentes para cumplir los objetivos anteriormente mencionados, sino también para definir una línea en el campo de la investigación de sistemas inteligentes y marcar la pauta sobre la cual se continua profundizando en esta área.

Es por esto que el desarrollo de este estudio, busca también abrir una línea de investigación en el departamento de Ciencias de la Computación de la Universidad del Valle de Guatemala en la gama de reconocimiento de patrones y análisis de datos utilizando Inteligencia Artificial.

IV. MARCO TEÓRICO

A. Gestión de proyectos informáticos

1. Proyecto. En sentido amplio, un proyecto es un conjunto o secuencia de actividades que desarrolla durante un tiempo un equipo de personas para obtener un resultado.

En general, un proyecto:

- Es un proceso; es decir, un conjunto de actividades interrelacionadas, en las que se trasforman un conjunto de recursos (inputs) en un conjunto de resultados (outputs) que tienen un sentido para alguien (un cliente, interno o externo).
- Un proyecto tiene un objetivo. Normalmente, el resultado u objetivo es también un proceso, o la trasformación de uno que ya existe.
- Tiene una duración, un inicio y un final. La temporalidad es quizá el elemento clave y diferencial de un proyecto frente a otra clase de proceso.
- Es único y diferente. Frente a las operaciones repetitivas, propias de la mayoría de los procesos industriales, cada proyecto es único e irrepetible.
- Es multidisciplinario, involucra recursos y habilidad de diferentes partes de una organización o de varias.
- Tiene recursos limitados y, por lo tanto, una serie de costos directos, indirectos y de oportunidad para la organización.

- Se puede decir que un proyecto es un encargo específico, dirigido y personalizado que realiza una organización a un grupo interno o externo de personas, que se configura para su ejecución (Mínguez, J. *et al*. 2011).

2. Proyectos informáticos. Un proyecto informático es una secuencia de actividades que desarrolla durante un tiempo predeterminado y con unos recursos limitados un equipo de personas, informáticos y no informáticos, para obtener unos resultados sobre la organización y los procesos de trabajo. Una parte sustancial de estas actividades requiere conocimientos y habilidades en las materias de sistemas y tecnologías de la información (Mínguez, J. *et al*. 2011).

Estos proyectos tienen una mayoría de características semejantes a las de los proyectos en genérico, pero tienen algunas peculiaridades o especialidades:

- Son más o menos replicables; es decir, hay muchos parecidos, por los productos (en especial de software) o las metodologías que se utilizan. Muchas metodologías y productos son estándar para resolver determinada clase de problemas o parte de los mismos.
- Los especialistas son informáticos, profesionales que comparten un cuerpo de pensamiento, lenguaje, métodos propios.
- Algunas características de los productos informáticos de hardware y software, referidas a su estabilidad, volatilidad, nivel y extensión del servicio. El cambio tecnológico es más rápido en este entorno que en otros.
- (Mínguez, J. *et al.* 2011).

3. Importancia gestión de proyectos informáticos. Empíricamente, se dice que más del 50% de los proyectos informáticos no responden a los objetivos que tenían planeados o han tenido desviaciones significativas de tiempo o de coste. De acuerdo con un estudio del Standish Group sobre proyectos informáticos en todo el mundo, de los proyectos analizados un 31% fueron cancelados antes de su finalización; en un 88% de los casos, se superó el periodo acordado. Y, lo que es más importante, el volumen económico de sobrecoste alcanzó el 222% de las estimaciones iniciales (Mínguez, J. *et al*. 2011).

En efecto, gestionar con éxito proyectos en general, y los informáticos en particular, es cada vez más difícil porque supone mayores niveles de exigencia (en términos de tiempo, costo y calidad), pero también de riesgo y complejidad, derivados del tamaño, la multidisciplinariedad y el cambio tecnológico acelerado (Mínguez, J. *et al.* 2011).

La gestión de proyectos es la disciplina de conocimiento y experiencia que permite planificar, organizar y gestionar proyectos. Esto quiere decir principalmente dos cosas:

- Asegurar que los proyectos se completan satisfactoriamente y que se consiguen sus productos y resultados últimos.
- Hacerlo de manera que se pueda predecir y controlar su evolución y explicarlo satisfactoriamente al equipo de trabajo y al cliente. (Mínguez, J. *et al*. 2011).

B. Dirección de proyectos

La dirección de proyectos es la aplicación de conocimientos, habilidades, herramientas y técnicas a las actividades del proyecto para cumplir con los requisitos del mismo. Se logra mediante la aplicación e

integración adecuadas de los 42 procesos de la dirección de proyectos, agrupados lógicamente, que conforman los cinco grupos de procesos. Estos cinco grupos de procesos son:

- Iniciación
- Planificación
- Ejecución
- Seguimiento y Control
- Cierre

Dirigir un proyecto por lo general implica:

- Identificar requisitos,
- Abordar las diversas necesidades, inquietudes y expectativas de los interesados según se planifica y efectúa el proyecto,
- Equilibrar las restricciones contrapuestas del proyecto que se relacionan, entre otros aspectos, con:
  - El alcance,
  - La calidad,
  - El cronograma,
  - El presupuesto,
  - Los recursos y
  - El riesgo.

El proyecto específico influirá sobre las restricciones en las que el director del proyecto necesita concentrarse. La relación entre estos factores es tal que si alguno de ellos cambia, es probable que al menos otro se vea afectado. Dada la posibilidad de sufrir cambios, el plan para la dirección del proyecto es iterativo y su elaboración es gradual a lo largo del ciclo de vida del proyecto. La elaboración gradual implica mejorar y detallar constantemente un plan, a medida que se cuenta con información más detallada y específica, y con estimados más precisos. La elaboración gradual permite a un equipo de dirección del proyecto dirigir el proyecto con un mayor nivel de detalle a medida que éste avanza (Project Management Institute, 2008).

C. Matriz de dirección de proyectos

1. Grupos de procesos. Los cinco grupos de procesos son:

- **Iniciación:** Son los procesos que se realizan para definir un nuevo proyecto o una nueva fase de un proyecto ya existente, mediante la autorización para comenzar dicho proyecto o fase.

- **Planificación:** Sirven para establecer el alcance del proyecto, refinar los objetivos y definir el curso de acción necesario para alcanzar los objetivos para cuyo logro se emprendió el proyecto.

- **Ejecución:** Son los que se utilizan para completar el trabajo definido en el plan para la dirección del proyecto a fin de cumplir con las especificaciones del mismo.

- **Seguimiento y control:** Son requeridos para monitorear, analizar y regular el progreso y el desempeño del proyecto, para identificar áreas en las que el plan requiera cambio y para iniciar los cambios correspondientes.

- **Cierre:** Sirven para finalizar todas las actividades a través de todos los grupos de procesos a fin de cerrar formalmente el proyecto o una fase del mismo. (Project Management Institute, 2008).

2. **Área de conocimiento.** Las nueve áreas de conocimiento son:

a. Integración: Incluye los procesos y actividades necesarias para identificar, definir, combinar, unificar y coordinar los diversos procesos y actividades.

b. Alcance: Incluye los procesos necesarios para garantizar que el proyecto incluya todo (y únicamente todo) el trabajo requerido para completarlo con éxito.

c. Tiempo: Incluye los procesos requeridos para administrar la finalización del proyecto a tiempo.

d. Costos: Incluye los procesos involucrados en estimar, presupuestar y controlar los costos de modo que se complete el proyecto dentro del presupuesto aprobado.

e. Calidad: Incluye los procesos y actividades de la organización que determinan responsabilidades, objetivos y políticas de calidad a fin de que el proyecto satisfaga las necesidades por las cuales fue emprendido.

f. Recursos humanos: Incluye los procesos que organizan, gestionan y conducen el equipo del proyecto.

g. Comunicaciones: Incluye los procesos necesarios para asegurar que el proyecto genere, recolecte, distribuya, almacene y disponga de información en tiempo y en forma.

h. Riesgos: Incluye los procesos necesarios para que en caso de que un evento incierto ocurra, se minimicen los efectos negativos o se potencialicen los positivos sobre los objetivos del proyecto.

i. Adquisiciones: Incluye los procesos necesarios para adquirir los bienes y servicios externos a la organización con el fin de lograr el alcance del proyecto. (Project Management Institute, 2008).

3. Grupo de procesos de iniciación

a. Desarrollar el Acta de Constitución del Proyecto: Consiste en desarrollar un documento que autoriza formalmente un proyecto o una fase, y en documentar los requisitos iniciales que satisfacen las necesidades y expectativas de los interesados.

b. Identificar a los interesados: Consiste en identificar a todas las personas u organizaciones que reciben el impacto del proyecto, y en documentar información relevante relativa a sus intereses, participación e impacto en el éxito del proyecto. (Project Management Institute, 2008).

4. Grupos de Procesos de Planificación I.

a. Desarrollar el Plan para la Dirección del Proyecto: Documentar las acciones necesarias para definir, preparar, integrar y coordinar todos los planes subsidiarios.

b. Recopilar requisitos: Definir y documentar las necesidades de los interesados a fin de cumplir con los objetivos del proyecto.

c. Definir el alcance: Consiste en desarrollar una descripción detallada del proyecto y del producto.

d. Crear la EDT (Estructura de Desglose del Trabajo): Es el proceso que consiste en subdividir los entregables y el trabajo del proyecto en componentes más pequeños y más fáciles de manejar.

e. Definir las actividades: Consiste en identificar las acciones específicas a ser realizadas para elaborar los entregables del proyecto.

f. Secuenciar las actividades: Consiste en identificar y documentar las relaciones entre las actividades del proyecto.

g. Estimar los recursos de las actividades: Es el proceso que consiste en estimar el tipo y las cantidades de materiales, personas, equipos o suministros requeridos para ejecutar cada actividad.

h. Estimar la duración de las actividades: Consiste en establecer aproximadamente la cantidad de periodos de trabajo necesarios para finalizar cada actividad con los recursos estimados.

i. Desarrollar el cronograma: Consiste en analizar el orden de las actividades, su duración, los requisitos de recursos y las restricciones del cronograma para crear el cronograma del proyecto.

j. Estimar los costos: Es una aproximación de los recursos monetarios necesarios para completar las actividades del proyecto.

k. Determinar el presupuesto: Consiste en sumar los costos estimados de las actividades individuales o paquetes de trabajo para establecer una línea base de costos autorizados.

l. Planificar la calidad: Es el proceso por el cual se identifican los requisitos de calidad y/o normas para el proyecto y el producto, y se documenta la manera en que el proyecto demostrara el cumplimiento con los mismos.

m. Desarrollar el Plan de RRHH: Consiste en identificar y documentar los roles dentro de un proyecto, las responsabilidades, las habilidades requeridas y las relaciones de comunicación, y se crea el plan de RRHH.

n. Planificar las comunicaciones: Es el proceso para determinar las necesidades de información de los interesados en el proyecto y para definir cómo abordar las comunicaciones.

ñ. Planificar la gestión de riesgos: Es el proceso por el cual se define como realizar las actividades de gestión de riesgos para un proyecto.

o. Identificar los riesgos: Es el proceso por el cual se determinan los riesgos que pueden afectar el proyecto y se documentan sus características.

p. Realizar el análisis cualitativo de riesgos: Es el proceso que consiste en priorizar los riesgos para realizar otros análisis o acciones posteriores, evaluando y combinando la probabilidad de ocurrencia y el impacto de dichos riesgos.

q. Realizar el análisis cuantitativo de riesgos: Es el proceso que consiste en analizar numéricamente el efecto de los riesgos identificados sobre los objetivos generales del proyecto.

r. Planificar la respuesta a los riesgos: Es el proceso por el cual se desarrollan opciones y acciones para mejorar las oportunidades y reducir las amenazas a los objetivos del proyecto.

s. Planificar las adquisiciones: Es el proceso que consiste en documentar las decisiones de compra para el proyecto, especificar el enfoque e identificar posibles vendedores. (Project Management Institute, 2008).

5. Grupo de procesos de ejecución.

a. Dirigir y gestionar la ejecución del proyecto: Consiste en ejecutar el trabajo definido en el plan para la dirección del proyecto para cumplir con los objetivos del mismo.

b. Realizar el aseguramiento de calidad: Consiste en auditar los requisitos de calidad y los resultados obtenidos a partir de medidas de control de calidad, a fin de garantizar que se utilicen definiciones operacionales y normas de calidad adecuadas.

c. Adquirir el equipo del proyecto: Es el proceso para confirmar los recursos humanos disponibles y formar el equipo necesario para completar las asignaciones del proyecto.

d. Desarrollar el equipo del proyecto: Consiste en mejorar las competencias, la interacción de los miembros del equipo y el ambiente general del equipo para lograr un mejor desempeño del proyecto.

e. Dirigir el equipo del proyecto: Consiste en monitorear el desempeño de los miembros del equipo, proporcionar retroalimentación, resolver problemas y gestionar cambios a fin de optimizar el desempeño del proyecto.

f. Distribuir la información: Es el proceso para poner la información relevante a la disposición de los interesados en el proyecto de acuerdo al plan establecido.

g. Gestionar las expectativas de los interesados: Es el proceso que consiste en comunicarse y trabajar en conjunto con los interesados para satisfacer sus necesidades y abordar los problemas conforme se presentan.

h. Efectuar las adquisiciones: Consiste en obtener respuestas de los vendedores, seleccionar un vendedor y adjudicar un contrato. (Project Management Institute, 2008).

6. Grupo de procesos de seguimiento y control.

a. Monitorear y controlar el Trabajo del Proyecto: Es el proceso que consiste en revisar, analizar y regular el avance a fin de cumplir con los objetivos de desempeño definidos en el plan para la dirección del proyecto.

b. Realizar el Control Integrado de Cambios: Es el proceso que consiste en revisar todas las solicitudes de cambios, aprobar los cambios y gestionar los cambios a los entregables, a los activos de los procesos de la organización, a los documentos del proyecto y al plan para la dirección del proyecto.

c. Verificar el alcance: Es el proceso que consiste en formalizar la aceptación de los entregables del proyecto que se han completado.

d. Controlar el alcance: Es el proceso por el cual se monitorea el estado del alcance del proyecto y del producto, y se gestionan cambios a la línea base del alcance.

e. Controlar el cronograma: Es el proceso por el cual se monitorea la situación del proyecto para actualizar el avance del mismo y gestionar cambios a la línea base del cronograma.

f.. Controlar los costos: Es el proceso por el cual se monitorea la situación del proyecto para actualizar el presupuesto del mismo y gestionar cambios a la línea base del costo.

g. Realizar el control de calidad: Es el proceso por el cual se monitorean y se registran los resultados de la ejecución de actividades de control de calidad, a fin de evaluar el desempeño y recomendar cambios necesarios.

h. Informar el desempeño: Es el proceso de recopilación y distribución de información sobre el desempeño, incluyendo los informes de estado, las mediciones del avance y las proyecciones.

i. Monitorear y controlar los riesgos: Es el proceso por el cual se implementan planes de respuesta a los riesgos, se da seguimiento a los riesgos identificados, se da seguimiento a los riesgos

residuales, se identifican nuevos riesgos y se evalúa la efectividad del proceso contra riesgos a través del proyecto.

j. Administrar las adquisiciones: Es el proceso que consiste en gestionar las relaciones de adquisiciones, supervisar el desempeño del contrato y efectuar cambios y correcciones según sea necesario. (Project Management Institute, 2008).

7. Grupo de procesos de cierre

a. Cerrar el proyecto o fase: Es el proceso que consiste en finalizar todas las actividades a través de todos los grupos de procesos de la dirección de proyectos para completar formalmente el proyecto o una fase del mismo.

b. Cerrar las adquisiciones: Es el proceso de finalización de cada adquisición del proyecto. (Project Management Institute, 2008).

D. Proyectos ágiles

El entorno de trabajo de las empresas del conocimiento se parece muy poco al que originó la gestión de proyectos predictiva. Ahora se necesitan estrategias para el lanzamiento de productos orientadas a la entrega temprana de resultados tangibles, y a la respuesta ágil y flexible, necesaria para trabajar en mercados de evolución rápida.

Ahora se construye el producto mientras se modifican y aparecen nuevos requisitos. El cliente parte de una visión medianamente clara, pero el nivel de innovación que requiere, y la velocidad a la que se mueve su sector de negocio, no le permiten predecir con detalle cómo será el resultado final.

La gestión de proyectos ágil no se formula sobre la necesidad de anticipación, sino sobre la de adaptación continua. Quizá ya no hay "productos finales", sino productos en continua evolución y mejora.

La mayoría de los fiascos se producen por aplicar ingeniería secuencial y gestión predictiva tanto en el proceso de adquisición (requisitos, contratación, seguimiento y entrega) como en la gestión del proyecto, en productos que no necesitan tanto garantías de previsibilidad en la ejecución, como respuesta rápida y flexibilidad para funcionar en entornos de negocio que cambian y evolucionan rápidamente. (Palacio, J. 2014).

1. El Manifiesto Ágil. En marzo de 2001, 17 críticos de los modelos de producción basados en procesos, convocados por Kent Beck, que había publicado un par de años antes el libro en el que explicaba la nueva metodología Extreme Programming (Beck, 2000) se reunieron en Salt Lake City para discutir sobre el desarrollo de software. En la reunión se acuñó el término "Métodos Ágiles" para definir a aquellos

que estaban surgiendo como alternativa a las metodologías formales: CMM-SW, (precursor de CMMI) PMI, SPICE (proyecto inicial de ISO 15504), a las que consideraban excesivamente “pesadas” y rígidas por su carácter normativo y fuerte dependencia de planificaciones detalladas, previas al desarrollo.

Los integrantes de la reunión resumieron en cuatro postulados lo que ha quedado denominado como “Manifiesto Ágil”, que son los valores sobre los que se asientan estos métodos.

Hasta 2005, entre los defensores de los modelos de procesos y los de modelos ágiles fueron frecuentes las posturas radicales, más ocupadas en descalificar al otro, que en estudiar sus métodos y conocerlos para mejorar los propios. (Palacio, J. 2014).

En el Manifiesto Ágil se declaran cuatro valores que se listan y se describen a continuación:

a. **Valoramos más a los individuos y su interacción que a los procesos y las herramientas.** Este es el valor más importante del manifiesto. Por supuesto que los procesos ayudan al trabajo. Son una guía de operación. Las herramientas mejoran la eficiencia, pero hay tareas que requieren talento y necesitan personas que lo aporten y trabajen con una actitud adecuada.

La producción basada en procesos persigue que la calidad del resultado sea consecuencia del know-how “explicitado” en los procesos, más que en el conocimiento aportado por las personas que los ejecutan.

Sin embargo en desarrollo ágil los procesos son una ayuda. Un soporte para guiar el trabajo. La defensa a ultranza de los procesos lleva a afirmar que con ellos se pueden conseguir resultados extraordinarios con personas mediocres, y lo cierto es que este principio no es cierto cuando se necesita creatividad e innovación. (Palacio, J. 2014).

b. **Valoramos más el software que funciona que la documentación exhaustiva.** Poder anticipar cómo será el funcionamiento del producto final, observando prototipos previos, o partes ya elaboradas ofrece un "feedback" estimulante y enriquecedor, que genera ideas imposibles de concebir en un primer momento, y difícilmente se podrían incluir al redactar un documento de requisitos detallado en el comienzo del proyecto.

El manifiesto ágil no da por inútil la documentación, sólo la de la documentación innecesaria. Los documentos son soporte de hechos, permiten la transferencia del conocimiento, registran información histórica, y en muchas cuestiones legales o normativas son obligatorios, pero su relevancia debe ser mucho menor que el producto final.

La comunicación a través de documentos no ofrece la riqueza y generación de valor que logra la comunicación directa entre las personas, y a través de la interacción con prototipos del producto. Por eso, siempre que sea posible debe preferirse reducir al mínimo indispensable el uso de documentación, que requiere trabajo sin aportar un valor directo al producto.

Si la organización y los equipos se comunican a través de documentos, además de ocultar la riqueza de la interacción con el producto, forman barreras de burocracia entre departamentos o entre personas. (Palacio, J. 2014).

c. Valoramos más la colaboración con el cliente que la negociación contractual. Las prácticas ágiles se recomiendan para productos cuyo detalle resulta difícil de prever al principio del proyecto; y en caso se detallara de esta forma, el resultado final sería menos relevante que el resultado obtenido a través de la retroalimentación, auditoria y mejora continua del proyecto durante su curso.

También son apropiadas cuando se prevén requisitos inestables por la velocidad de cambio en el entorno de negocio del cliente. El objetivo de un proyecto ágil no es controlar la ejecución conforme a procesos y cumplimiento de planes, sino proporcionar el mayor valor posible al producto. Resulta por tanto más adecuada una relación de implicación y colaboración continua con el cliente, más que una contractual de delimitación de responsabilidades. (Palacio, J. 2014).

d. Valoramos más la respuesta al cambio que el seguimiento de un plan. Para desarrollar productos de requisitos inestables, que tienen como factor inherente el cambio y la evolución rápida y continua, resulta mucho más valiosa la capacidad de respuesta que la de seguimiento y aseguramiento de planes. Los principales valores de la gestión ágil son la anticipación y la adaptación, diferentes a los de la gestión de proyectos ortodoxa: planificación y control que evite desviaciones del plan. (Palacio, J. 2014).

Los 12 principios del manifiesto ágil. El manifiesto ágil, tras los postulados de estos cuatro valores en los que se fundamenta, establece estos 12 principios:

- Nuestra principal prioridad es satisfacer al cliente a través de la entrega temprana y continua de software de valor.
- Son bienvenidos los requisitos cambiantes, incluso si llegan tarde al desarrollo. Los procesos ágiles se doblegan al cambio como ventaja competitiva para el cliente.
- Entregar con frecuencia software que funcione, en periodos de un par de semanas hasta un par de meses, con preferencia en los periodos breves.
- Las personas del negocio y los desarrolladores deben trabajar juntos de forma cotidiana a través del proyecto.

- Construcción de proyectos en torno a individuos motivados, dándoles la oportunidad y el respaldo que necesitan y procurándoles confianza para que realicen la tarea.
- La forma más eficiente y efectiva de comunicar información de ida y vuelta dentro de un equipo de desarrollo es mediante la conversación cara a cara.
- El software que funciona es la principal medida del progreso.
- Los procesos ágiles promueven el desarrollo sostenido. Los patrocinadores, desarrolladores y usuarios deben mantener un ritmo constante de forma indefinida.
- La atención continua a la excelencia técnica enaltece la agilidad.
- La simplicidad como arte de maximizar la cantidad de trabajo que se hace, es esencial.
- Las mejores arquitecturas, requisitos y diseños emergen de equipos que se auto organizan.
- En intervalos regulares, el equipo reflexiona sobre la forma de ser más efectivo y ajusta su conducta en consecuencia. (Palacio, J. 2014).

E. Manufactura esbelta en desarrollo de software

Lean, es un sistema de mejoramiento de procesos de manufactura basado en la eliminación de desperdicios y actividades que no agregan valor al proceso. A continuación los 14 principios:

- Las decisiones del negocio están basadas en una visión a largo plazo, aún a expensas de las pérdidas financieras a corto.
- Los ciclos son cortos y rápidos.
- Se prefieren los sistemas “pull”, que evitan la sobreproducción.
- La carga de trabajo debe ser balanceada (Heijunka).
- La cultura lean comprende detener la producción para arreglar problemas, así como en enseñar el estudio metódico de los problemas (Jidoka).
- Las tareas se estandarizan para lograr la mejora continua (Kaizen).
- La gestión visual simple revela problemas y permite la coordinación.
- Se utiliza solamente tecnología probada que pueda ser provechosa para la gente y su proceso.
- Se forman líderes que comprendan el trabajo, vivan la filosofía de la empresa y la enseñen a otros.
- Se desarrollan equipos y personas excepcionales que siguen la filosofía de la compañía.
- Se respeta la red de colaboradores y proveedores (Keiretsu) , desafiándolos a crecer y ayudándolos a la mejora.
- Se valora que los responsables vayan y miren las situaciones en el lugar de trabajo, para entenderlas y poder ayudar.
- Decisiones basadas en el consenso y la consideración minuciosa de todas las opciones, y su posterior implementación rápida.

- Empresa como organización que aprende a través de la reflexión constante (Hansei) y de la mejora continua (Kaizen). (Palacio, J. 2014).

1. Algunas prácticas comunes en la producción Lean.

- Kanban: o presentación de información visual relativa a la producción (identificación de componentes, estado del proceso, etc) presentada físicamente en el lugar de trabajo implicado y permanentemente actualizada.
- 5S: Metodología de trabajo cuyo nombre procede de las 5 palabras japonesas que la configuran: seiri, sieton, seiso, siketsu and shitsuke (clasificar, ordenar, limpiar, estandarizar y sostener).
- Poka-yoke: o sistema a prueba de error, que busca crear mecanismos que sólo permiten hacer el trabajo de la forma adecuada.
- JIT (Just in time): producir en el momento que es requerido.
- Andon: Sistemas visuales de alertas sobre anomalías, fallos o problemas en el momento y lugar de la producción.
- Jidoka. Instalación en el proceso de sistemas que verifican su calidad.
- Heijunka. Técnicas para adaptar la producción a una demanda fluctuante del cliente. (Palacio, J. 2014).

F. Lean Software Development

Este término se refiere a la aplicación de los principios Lean en el desarrollo del software. Mary y Tom Poppendieck (Poppendieck & Poppendieck, 2003) lo acuñaron. Gracias a sus aportes y los de la comunidad ágil, Lean Software Development está desarrollando un inventario de prácticas útiles para el desarrollo ágil de software (Palacio, J. 2014).

Se basa en siete principios:

1. Eliminar el desperdicio.

- Las actividades que no crean valor no sirven y deben ser eliminadas. Algunos ejemplos:
    - Tareas que no fueron solicitadas por el cliente.
    - Sobre-documentación del proyecto.
    - Procesos de desarrollo que se ejecutan sin analizar su nivel de eficiencia o vigencia.
    - Un mayor número de líneas de código no siempre es mejor, y además requiere mayor esfuerzo de testeo y de mantenimiento.
    - Los errores, bugs y fallos del software son verdadero desperdicio que se debe reducir.

2. Construir con calidad. Incluir en el procedimiento prácticas para mejora de la calidad en el producto (traslación de prácticas "poka-yoke" y "andon" al desarrollo de software). Un procedimiento que respeta la calidad es aquel que es conocido, entendido y mejorado por los propios participantes. Para lograrlo es necesario compromiso y respeto.

Algunos ejemplos de prácticas que se deben contemplar al hacer software.

- Técnicas como TDD (Test Driven Development) permiten que usuarios (clientes), programadores y tester definan claramente los requerimientos y confeccionen pruebas de aceptación antes de escribir el código. Ayuda a la comprensión de los programadores y mejora el entendimiento de los requerimientos.
- El programador es responsable de su propio desarrollo. No debe esperar a que las pruebas o los procedimientos de aseguramiento de calidad descubran los errores.
- Fomentar el desarrollo de pruebas automatizadas.
- Refactorización del código, para lograr simplicidad y eliminar duplicidades.

3. Compartir conocimiento. Conocer lo que necesita el cliente requiere dedicación y esfuerzo, y debe convertirse en el aspecto principal, porque desarrollar un producto que no es útil, es el mayor desperdicio. Hacer software implica un proceso de aprendizaje: entender qué es lo que el cliente quiere y cómo entregar la mejor solución posible. El desarrollo incremental proporciona cuantiosa y frecuente retroinformación.

4. Diferir el compromiso. En los proyectos ágiles que parten con una visión que evoluciona con el desarrollo, el compromiso con el cliente se asienta y evoluciona en la misma medida que se van concretando y comprometiendo los incrementos del producto.

5. Entregar rápido. La gestión evolutiva realiza entregas rápidas a los clientes, que se encuentran con código operativo desde etapas tempranas. Dicho código debe ser desarrollado con calidad ya que no se puede mantener una velocidad importante de entrega si no se cuenta con calidad y un equipo disciplinado, comprometido y confiable.

6. Respetar a las personas. Lean se basa en el respecto por las personas que son el elemento único y diferenciador de cada organización. Deben estar suficientemente capacitadas y ser responsables de los procesos en los que intervienen, de modo que cuando resultan necesarios cambios y mejoras, cada persona colabora en su desarrollo.

7. Optimizar el todo. Lean invita a contemplar el proceso completo, es decir todo el flujo de valor, en lugar de hacerlo en cada etapa. El problema de optimizar cada fase por separado es que genera inventarios grandes en los puntos de transición. En el mundo del software, estos "inventarios" representan al trabajo parcialmente terminado (por ejemplo, requisitos completos, pero sin diseñar, codificar o probar). Lean demostró que un flujo de "una pieza" (por ejemplo, enfocarse en construir un ítem de manera completa) es un proceso más eficiente que concentrarse en construir las partes separadas de forma rápida. (Palacio, J. 2014).

G. Metodología de desarrollo ágil Scrum

Como lo menciona Schwaber, K. (2004:10), Scrum es una metodología de desarrollo eficiente de software que permite mejorar el desempeño de un equipo de trabajo a través de dos conceptos: descentralización y el ciclo de Deming. Evalúa la centralización todas las operaciones en proyectos complejos, de manera que termina exponiendo la capacidad limitada del ente central para poder llevar con detalle las actividades individuales. Scrum busca delegar las tareas en agentes separados capaces de mejorar la efectividad a través de la distribución de tareas. Estas tareas a través de un ciclo corto e iterativo de descubrimiento permiten al equipo de trabajo aprender sobre las tecnologías o etapas del proyecto a medida que se avanza con la ejecución; esta metodología basada en la mejora por iteraciones se basa en el ciclo de Deming.

La metodología Scrum según Schwaber K. & Sutherland J. (2013:3), es un «[...] marco de referencia en el cual las personas pueden dirigirse a problemas complejos adaptativos, mientras de manera productiva y creativa continúan entregando productos del valor más alto posible [...]». Como una metodología basada en el empirismo el objetivo de Scrum es optimizar la previsibilidad y el control de riesgos utilizando un enfoque iterativo e incremental que permite evaluar la distancia entre los objetivos y los resultados obtenidos del proyecto en cada etapa. Schwaber K. & Sutherland, J. (2013:3) definen tres componentes fundamentales en la metodología: la transparencia, la inspección y la adaptación. La transparencia se refiere a que aspectos significativos del proceso deben ser visibles a los que son responsables por el resultado, i.e., los equipos de trabajo que comparten tareas y evaluación de resultados deben poseer métricas públicas y claras que especifiquen la crítica y la finalización de las mismas. La inspección se refiere a que los usuarios de la metodología deben inspeccionar con frecuencia los artefactos y el progreso en base al objetivo de una iteración para detectar variaciones no deseadas o cambios de planificación requeridos; es importante que las inspecciones no obstaculicen el avance. La adaptación se refiere a la identificación de una variación de un componente que sobrepasa el límite aceptable, y si además se tiene la seguridad que el resultado se aleja de lo esperado, el proceso o el material utilizado debe ser ajustado para minimizar la variación no deseada.

El equipo de trabajo en una metodología Scrum se caracteriza por conformarse de un Product Owner, un equipo de desarrollo y un Scrum Master. El Product Owner se encarga de la maximización del valor del producto y el trabajo efectuado por el equipo de desarrollo, y también se encarga de manejar el Product Backlog que representa la lista de tareas o ítems que deben cumplirse por iteración. Sus tareas y responsabilidades suelen ser (Schwaber, K. & Sutherland, J., 2013:5):

- Describir los ítems del backlog
- Ordenar los ítems del backlog para alcanzar objetivos y la misión
- Optimizar el valor del equipo de desarrollo
- Procurar que el backlog sea visible, transparente, claro para todos
- Procurar que el backlog muestre los próximos pasos
- Asegurar que el equipo de desarrollo comprenda los ítems del backlog

El equipo de desarrollo es el encargado de entregar productos que se van aproximando a la categoría de completado en cada iteración. Trabajan como agentes que pueden gestionarse a sí mismos con el objetivo de mejorar su eficiencia y efectividad, y se recomienda que el tamaño del equipo sea pequeño para mantener su agilidad; Schwaber, .K & Sutherland, J. (2013:6) recomiendan el uso de un equipo que tenga entre 3 y 9 integrantes. Las características de un equipo son (Schwaber, K. & Sutherland, J., 2013:5):

- El equipo se organiza a sí mismo
- El equipo es multidisciplinario y contiene todas las habilidades necesarias
- No hay jerarquías o distinciones en base a función; todos son desarrolladores.
- No hay subgrupos dentro del equipo de desarrollo
- Aunque se delegan áreas específicas a miembros del equipo, todo el equipo es responsable del producto.

El Scrum Master tiene la responsabilidad de que la teoría, prácticas y reglas de la metodología Scrum sean cumplidas por todos los integrantes del proyecto. Además funciona como un medio de comunicación entre personas que no son parte del proyecto, v.g., inversionistas o clientes, y los integrantes, y les ayuda a mejorarla indicando que tipo de interacción podría beneficiar al proyecto. El Scrum Master ayuda a tres perfiles dentro de la metodología (Schwaber, K. & Sutherland, J., 2013:6):

- Al Product Owner. Le recomienda técnicas para la gestión del Product Backlog; ayuda a definir ítems claros y concisos; ayuda a entender la filosofía empírica y de agilidad; ayuda a que el Product Backlog genere el mayor valor posible; y ayuda con los eventos de Scrum.
- Al Equipo de Desarrollo. Ayuda al equipo con su auto-organización y su configuración de múltiples disciplinas; ayuda a crear productos de alto valor; elimina obstáculos para el progreso; ayuda a mediar en ambientes organizacionales donde no se comprende o maneja en su totalidad la metodología Scrum.

- A la Organización. Lidera la adopción de la metodología Scrum; planea implementaciones de la metodología en la empresa; ayuda a comprender la filosofía de desarrollo empírico de un producto; elimina obstáculos que afecten al equipo de desarrollo.

H. Estándar de seguridad PCI DSS (Payment Card Industry Data Security Standard)

Según Williams, B.R. et al (2012:2), el estándar PCI ha logrado mejorar el nivel de cumplimiento de los controles de seguridad en las organizaciones con énfasis en la protección de las redes de comunicación. El estándar debería ser parte de los requerimientos mínimos implementados por un comerciante que maneja tarjetas para realizar transacciones monetarias; en el caso contrario se puede incurrir en penalizaciones que llegan hasta la revocación del derecho de procesamiento de tarjetas. Además indica que su importancia reside en que es el único estándar de seguridad de información transaccional para pagos por tarjeta que fue creado y es mantenido por la misma industria. Las empresas más grandes de emisión de tarjetas han unido esfuerzos para implementar el estándar con el objetivo de proteger la información de los usuarios y de los sistemas de pago para evitar pérdidas monetarias por información comprometida y para evitar que su reputación se vea afectada. A pesar de que no es una ley ni una regulación que los gobiernos implementan Williams, B.R. et al (2012:14) recomiendan «[...] Who must comply with the PCI? [...] any organization that accepts payment cards or stores, processes, or transmits credit or debit card data must comply with PCI DSS.», de manera que todo negocio que desea manipular de alguna forma información de las empresas emisoras y sus tarjetas debería procurar adscribir sus procesos a esta metodología. No sólo las compañías interesadas en tener acceso directo tienen la responsabilidad de cumplir con el estándar, también se toma en cuenta a empresas que se encuentran en control de esta información de manera periférica, v.g., empresas proveedoras de servicios, empresas que proveen software cortafuegos o sistemas de detección de intrusos (IDS). (Williams, B.R. et al., 2012:15)

La importancia de generar estándares de seguridad en un ambiente laboral contemporáneo conectado globalmente reside en la reubicación de controles efectivos que permitan disuadir o detectar cuando un individuo explota una vulnerabilidad presente en los sistemas de pago o los medios utilizados para transmitir esta información. La información de las transacciones de un tarjetahabiente suele ser la más afectada por su valor en el mercado negro. La interconectividad ha simplificado el proceso de realizar transacciones pero también ha empoderado al crimen a expandir sus operaciones a un nivel internacional con poco esfuerzo. (Williams, B.R. et al, 2012:14)

I. ISO/IEC 8583

Los campos que son utilizados en el análisis de las transacciones monetarias se han seleccionado como un subgrupo de los parámetros indicados por el estándar ISO 8583. Este estándar se menciona en el glosario de definiciones del estándar PCI, el cual lo describe como el formato para la correspondencia de mensajes entre sistemas que manejan datos transaccionales. Además se seleccionaron otros campos que han sido implementados por la empresa, i.e., no se realiza un análisis de todos los campos que forman parte de cada transacción evaluada por la empresa.

J. Redes neuronales

Cuando se describe un algoritmo que utiliza una red neuronal se establece una analogía al conjunto de neuronas en el cerebro humano con el objetivo de indicar que: (1) se desconoce una forma directa e infalible de determinar la respuesta al problema, y (2) se posee una cantidad finita de ejemplos que muestran qué comportamiento tiene el problema con parámetros determinados. Esto quiere decir que se necesita un modelo, similar al modelo de aprendizaje humano, que permita adquirir y persistir información de cada uno de los ejemplos, y obtenga una generalización que se aproxime al verdadero comportamiento del problema estudiado.

La motivación de utilizar un modelo ingenuo biológico de aprendizaje se basa en la capacidad de la red de neuronas consolidar una serie de entradas o estímulos y transmitirlas hacia otras con mecanismos de adaptabilidad, v.g., inhibición o estimulación, en la velocidad o totalidad de la transmisión del impulso; este modelo se considera ingenuo debido a que toma en consideración solamente a las entradas de estímulo, a una función de activación que no depende de su entorno y a la salida de la misma. En un modelo biológico de una neurona los componentes actúan en una manera más compleja, v.g., la generación de estímulos necesaria para llegar al potencial de activación de una neurona no es causada únicamente por las entradas de la misma, existen células receptoras que afectan directamente el potencial de activación de una neurona en base a información sensorial percibida. Además el procesamiento de los estímulos se realiza de manera paralela, mientras que en un modelo simplificado que utilizado un algoritmo como la propagación de errores hacia atrás es secuencial (Kriesel, D., 2005:46).

Tomando el modelo simplificado propuesto por Kriesel, D. (2005:30) una neurona se compone de los siguientes elementos:

- Entrada vectorial. Vector de entrada a la red neuronal, corresponde a los estímulos sensoriales en la neurona humana.

- Salida escalar. Corresponde al resultado de la red neuronal. La salida suele generarse a través de la agregación de su entrada vectorial a través del mecanismo de umbral análogo al potencial de activación.
- Mecanismo de aprendizaje. Las neuronas se encuentran conectadas a tráves de enlaces que poseen pesos que inhiben o maximizan el estímulo hacia la próxima neurona, éstos son análogos al mecanismo de adaptabilidad en el modelo biológico. El aprendizaje se aplica en el momento en el que los pesos se ajustan para adecuarse a la información proveída en su entrenamiento.
- Naturaleza no-linear. Uno de los aspectos importantes del enfoque en la red neuronal es que no existe una relación lineal entre la entrada a una neurona y la salida, la relación asociativa que se genera entre estos valores permite modelar escenarios más complejos que en una situación de similitud proporcional.

A continuación se describen los componentes formales de una red neuronal en base a Kriesel, D. (2005:35).

Se define una red neuronal como la siguiente tripleta: (N, V, w). En esta tripleta se definen los sets N y V, y la función w. N representa el conjunto de todas las neuronas que pertenecen a la red; V representa el total de conexiones existentes en la red y se define por: $\{(i,j)|i,j \in N\}$, en donde i y j representan neuronas. La función w se define como: $w: V \rightarrow \mathbf{R}$, y representa los pesos asociados a cada enlace que conecta a las neuronas i y j; se simplifica de la siguiente manera: $w_{i,j}$.

1. Parámetros de entrenamiento. Uno de los parámetros importantes que se deben tomar en cuenta para el entrenamiento un perceptrón de múltiples capas es la tasa de aprendizaje. La tasa de aprendizaje incide directamente en la velocidad y exactitud del proceso de aprendizaje. Si este valor es muy alto el porcentaje de error, utilizando un optimización por gradiente descendiente, puede dar saltos muy grandes por la superficie del error y puede afectar la ubicación de mínimos si el intervalo en el que se encuentran es muy pequeño. Sin embargo si se utiliza una tasa de aprendizaje muy pequeña el tiempo que tarda el algoritmo en converger podría incrementarse. Kriesel, D. (2005:92) recomienda el intervalo entre 0.01 y 0.9 para realizar el entrenamiento siempre que sea muy costoso ampliarlo. Otra metodología útil en la fase de entrenamiento es utilizar una tasa de aprendizaje variable en distintas etapas. La forma más efectiva en la que se puede realizar esto es al iniciar con una tasa alta de aprendizaje que converge con rapidez, posteriormente se disminuye en una o más ocasiones el orden de la tasa para asegurar que el algoritmo no se mueva fuera del mínimo alcanzado.

Otro parámetro que puede modificarse para ajustar el entrenamiento es el impulso, en inglés momentum. El impulso consiste en la proporción en la que se agrega una fracción del cambio de la época

anterior a los cambios en la época actual sobre los pesos. La idea detrás de esto es mantener el impulso de descenso que tiene el algoritmo hacia un mínimo y procurar que se dirija hacia el mismo, sin embargo este no es siempre el caso. (Kriesel, D., 2005:97).

2. Propagación hacia atrás elástica (Resilient Backpropagation). Kriesel, D. (2005:93) menciona las desventajas del algoritmo utilizado para el entrenamiento en base a los errores y la distancia del vector de salida y el vector esperado, éstas son: (1) la tasa de aprendizaje puede resultar en malos resultados para la etapa de entrenamiento y (2) por la forma en la que se realiza el algoritmo de propagación hacia atrás, los pesos más alejados de la capa de salida consumen un mayor tiempo en el cálculo del error. Como solución a estos problemas se expone la mejora presentada por Riedmiller, M. (1993 y 1994), llamada propagación hacia atrás elástica; en inglés se conoce como Resilient Backpropagation.

Entre las diferencias principales que identifica Kriesel, D. (2005:93) se encuentran:

- No existe una tasa de aprendizaje global. Esta tasa se define por cada uno de los pesos y se define automáticamente por el algoritmo. Además la tasa es variable a lo largo del entrenamiento en cada época.
- Los pesos se ajustan en base al gradiente del error obtenido en cada época. Además la magnitud de cambio en el peso de cada enlace es directamente calculado a partir de la tasa de aprendizaje en ese momento, esto quiere decir que el cambio no es proporcional al gradiente sino solamente se guía por su signo.
- Existe una etapa posterior al ajuste de pesos en donde se realiza el ajuste de las tasas de aprendizaje.

Estas diferencias propuestas en la modificación del algoritmo son útiles cuando se utilizan perceptrones de múltiples capas con un número creciente de capas ocultas debido a la mejora en la rapidez de la convergencia del algoritmo.

K. Inteligencia artificial

De acuerdo a Alex Champandard en su sitio web Artifical Intelligence Plain and Simple, define la inteligencia artificial como una rama de la ciencia que procura ayudar a las máquinas a encontrar soluciones a problemas complejos de forma similar a como lo haría un humano. Esto generalmente involucra simular características de la inteligencia humana y aplicarlas como algoritmos computacionales de una manera amigable. Allí mismo, Champandard establece que la Inteligencia Artificial generalmente es asociada a las Ciencias de la Computación, sin embargo posee mucho enlaces importantes con otros campos como lo son las Matemáticas, Psicología, Cognición, Biología y Filosofía, entre otros. Por otro lado, Enrique Castillo,

José Manuel Gutiérrez y Ali S. Hadi (1996:13) nos proveen una definición más formal, estableciendo que la Inteligencia Artificial es la parte de la Ciencia que se ocupa del diseño de sistemas de computación inteligentes, es decir, sistemas que exhiben las características que asociamos a la inteligencia en el comportamiento humano que se refiere a la comprensión del lenguaje, el aprendizaje, el razonamiento, la resolución de problemas, etc. También definen que dentro de la Inteligencia Artificial, hoy en día se engloban subáreas tales como los sistemas expertos, la demostración automática de toeremas, el juego automático, el reconocimiento de la voz y de patrones, el procesamiento del lenguaje natural, la visión artificial, la robótica, las redes neuronales, etc. Entre estas áreas es donde se encuentra el campo de investigación de las redes bayesianas, que posteriormente se definirá en esta investigación.

Enrique Castillo, José Manuel Gutiérrez y Ali S. Hadi (1996:14), mencionan un componente importante dentro de la inteligencia artificial, siendo estos los sistemas expertos. Para definirlos utilizan una cita de Stevens (1984:40), en la cual define los sistemas expertos como máquinas que piensan y razonan cómo un experto lo haría en una cierta especialidad o campo. Por ejemplo, un sistema experto en diagnóstico médico requeriría como datos los síntomas del paciente, los resultados de análisis clínicos y otros hechos relevantes, y, utilizando éstos, buscaría en una base de datos la información necesaria para identificar la correspondiente enfermedad. También menciona que un Sistema Experto de verdad, no sólo realiza las funciones tradicionales de manejar grandes cantidades de datos, sino que también manipula estos datos de forma tal que el resultado sea inteligible y tenga significado para responder a preguntas incluso no completamente especificadas. Estos sistemas expertos, necesitan una manera de interpretar y aprender de los datos e información que tienen inicialmente, este concepto se ha estudiado en los últimos años para profundizar en la mejor forma de llevar a cabo esta tarea.

L. Aprendizaje de máquinas

Dentro de estos conceptos surge la necesidad de definir lo que se conoce hoy como aprendizaje de máquinas y engloba las diferentes formas que existen de permitir que un sistema experto aprenda de un conjunto de datos definidos. Según Siddaharth Shrotriya (2013), el aprendizaje de máquinas es una rama de la Inteligencia Artificial, que se enfoca en la construcción y estudio de los sistemas que pueden aprender dado un conjunto de datos. La parte fundamental de este aprendizaje es la representación y generalización de los datos analizados. La generalización es la propiedad que determina que un sistema se desempeñará de una forma correcta en instancias de datos no conocidas; las condiciones sobre las cuales se garantiza esto, es un aspecto clave en el subcampo de teoría de aprendizaje computacional. Por otro lado, David Barber (2011:251) establece que los dos principales subcampos del aprendizaje de máquinas son: el aprendizaje supervisado y el aprendizaje no supervisado. Barber establece que el aprendizaje supervisado se concentra en la predicción precisa, es decir, dado un conjunto de datos donde existe una variable de entrada y una variable de salida, la tarea principal es “aprender” la relación entre la variable de entrada y la variable de salida. En otras palabras, en el aprendizaje supervisado, se enfocan los esfuerzos en la descripción de una

variable resultado, condicionada por una variable de entrada inicial. A diferencia del aprendizaje supervisado, el no supervisado tiene como objetivo encontrar una descripción precisa y compacta de los datos analizados. Un objetivo es cuantificar la precisión de la descripción, por lo que no existe una variable de predicción específica. Desde un punto de vista probabilístico el interés principal es encontrar un modelo que representa la distribución de datos p(x). Es aquí donde el aprendizaje de máquinas y la inteligencia artificial se encuentran con las variables probabilísticas.

M. Teoría de la probabilidad

Para profundizar en las funciones probabilísticas, nos adentramos en los conceptos de la teoría de la probabilidad. Para ello, Paola Sebastiani (2010:3), define que la probabilidad es utilizada para medir la cantidad de certeza de un evento: un hecho para el cual su ocurrencia es incierta. La principal característica al momento de trabajar con eventos y su probabilidad de ocurrencia, son las variables aleatorias. Alex Smola y Vishwanathan (2010:12) establecen el ejemplo del dado, en el que si quisiéramos saber cuáles serían nuestras probabilidades de tener un cierto número al momento de lanzar un dado. Si el dado tiene una posibilidad diferente para cada una de sus seis caras, tendríamos que uno de cada seis resultados debería ser el número del cual se calculó su probabilidad. Es por esto que la teoría de la probabilidad nos permite modelar la incertidumbre en el resultado de dichos experimentos. Formalmente establecemos que la probabilidad de obtener un cierto número ocurre con una probabilidad de $\frac{1}{6}$. Smola y Vishwanathan establecen que en este tipo de experimentos, no todos los resultados pueden ser numéricos ya que si analizamos el lanzamiento de una moneda, el resultado será cara o escudo. Para estos experimentos es necesario definir una variable aleatoria, la cual se comportará de forma que mejor se adapte al experimento realizado. Una de las maneras más importantes de caracterizar una variable aleatoria es asociar probabilidades con los valores que esta tome. Al combinar múltiples variables aleatorias hacia un conjunto de datos se obtiene una distribución, la cual está definida como el comportamiento que el conjunto de datos tiene. Para complementar estos conceptos, Enrique Castillo, José Manuel Gutiérrez, yAli S. Hadi (1996:70) establecen que las redes probabilísticas han ayudado a resurgir de gran manera en relación a los conceptos que se tenían en un inicio. De igual forma citan a Lindley (1987), que establece que: “La única descripción satisfactoria de la incertidumbre es la probabilidad. Esto quiere decir que toda afirmación incierta debe estar en forma de una probabilidad, ue varias incertidumbres deben ser combinadas usando las reglas de la probabilidad, y que el cálculo de probabilidades es adecuado para manejar situaciones que implican incertidumbre. En particular, las descripciones alternativas de la incertidumbre son innecesarias.” Con esto, se define que la teoría probabilística se torna en una de nuestras principales bases de interés para la realización de este proyecto.

N. Términos estadísticos y matemáticos

Para comprender el resto de definiciones del marco teórico es necesario conocer ciertos términos estadísticos y matemáticos, que se definirán a continuación.

1. **Varianza.** En estadística la varianza determina que tan extendido se encuentra un conjunto de números.

2. **Norma.** Se refiere a una función que asigna un valor positivo, que representa su longitud, a un vector en un espacio vectorial.

3. **Optimización de descenso por gradiente.** Este método es utilizado para el cálculo del mínimo local. Para ello se reduce dicho cálculo a una serie de problemas de búsqueda lineal.

4. **Hiperplano.** Un hiperplano es un subespacio de una dimensión menor que un espacio. Por ejemplo para un espacio de tres dimensiones un hiperplano es un plano y para un espacio de dos dimensiones un hiperplano es una línea.

Ñ. Pre-procesamiento de datos

El pre-procesamiento de datos incluye la preparación de los datos y la reducción de los mismos. El resultado del pre-procesamiento es un conjunto de datos final, que se considera correcto y útil, como entrada a un algoritmo.

1. **Preparación de datos.** La preparación de los datos se compone de integración, limpieza, normalización y transformación de los datos. Comprende todos los pasos necesarios para estandarizar los datos y asegurarse de que puedan ser procesados por el algoritmo.

2. **Normalización de datos.** La normalización de datos transforma la distribución de los datos de entrada a un nuevo grupo de datos, con las propiedades deseadas, sin generar nuevos atributos.

3. **Normalización por unidad de longitud.** En este trabajo se tomará normalización por unidad de longitud como la traducción de scaling to unit lenght. Este método consiste en tomar cada uno de los datos de un conjunto, con al menos un elemento que no sea cero, y escalarlo, independientemente de otros conjuntos, para que la norma del resultado sea uno.

4. **Normalización estándar.** En este trabajo se tomará normalización estándar como la traducción de standar scaler. Este método se utiliza para estandarizar conjuntos de datos removiendo la media y escalando con respecto a la varianza unitaria.

5. **Reducción de datos.** La reducción de datos puede ser selección de características, reducción de instancias, discretización, etc. El resultado de la reducción de datos mantiene la misma estructura que al inicio, pero cuenta con una menor cantidad de datos.

O. Support vector machine

Support Vector Machines (SVM) son modelos de aprendizaje supervisado utilizado para resolver problemas de clasificación. Una SVM realiza clasificación al construir un hiperplano N-dimensional que óptimamente separe los datos en las clases necesarias, incrementando los márgenes lo más posible (ver Figura 2).

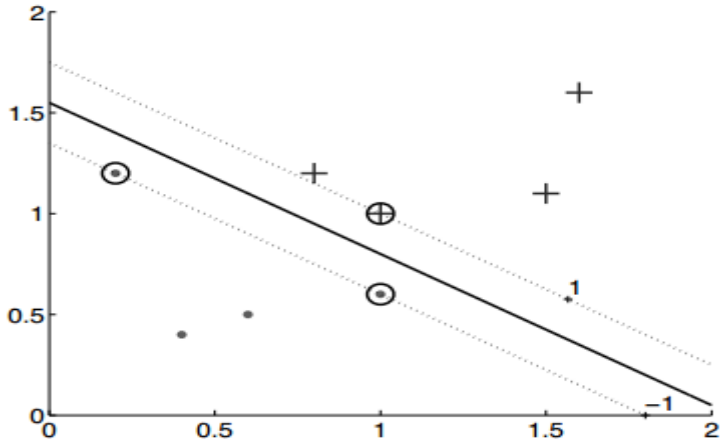
1. **Margen.** El margen es el espacio entre el hiperplano y el dato más cercano a él. La SVM busca que el margen sea el mayor posible.

2. **Vector de soporte.** Las instancias de los datos que se encuentren justo en un margen se llaman vectores de soporte.

3. **Función Kernel.** Una función kernel mapea sus entradas a un nuevo espacio de mayor dimensión. Para las SVMs estas funciones permiten que las SVMs sean utilizadas para clasificación no lineal (ver Figura 3).

4. **Peso de una clase.** En caso los datos de entrenamiento de una clase contengan muchos más de una clase que de otra, es posible especificar el peso de la clase. Este peso representa la cantidad de datos de una clase en relación con la cantidad de datos global.

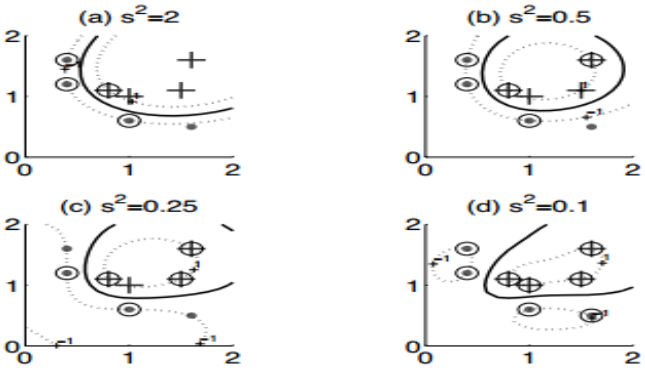
Figura 1 - Para un problema de dos clases donde las instancias de las clases se muestran por medio de signos de suma y puntos, la línea gruesa es el límite (hiperplano) y las líneas punteadas definen los márgenes en cada lado. Las instancias circuladas son vectores de soporte.



Fuente: Alpaydin, 2010:314

5. Método de descenso de gradiente estocástico. Es un método que utiliza la optimización de descenso por gradiente para minimizar una función que es escrita como la suma de funciones diferenciables. Este método se puede utilizar para entrenar una SVM lineal ya que su función objetivo puede ser escrita como una suma de funciones diferenciables. Sin embargo al utilizar otros kernels ya no es posible representar la función de esta forma, por lo que no se puede entrenar utilizando el método del gradiente estocástico.

Figura 2. El límite y márgenes encontrados por el kernel Gaussiano con diferentes valores de propagación, s2. Se encuentran límites más suaves con mayor propagación



Fuente: Alpaydin, 2010:323

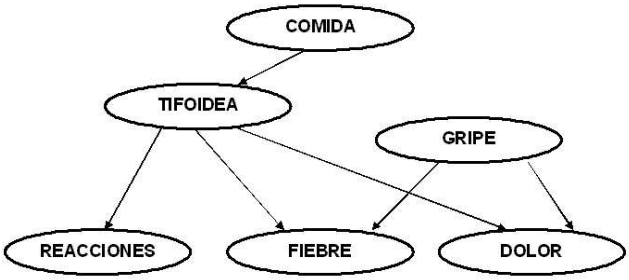
P. Redes Bayesianas (Sistemas de razonamiento probabilístico)

Al unir los conceptos anteriormente explorados, tenemos la creación de sistemas inteligentes que utilizan un razonamiento probabilístico. Stuart Russell y Peter Norvig (1995:436) expresan que la principal ventaja que tienen los sistemas de razonamiento probabilístico sobre los de razonamiento lógico, es que estos primeros permiten a un agente alcanzar decisiones racionales incluso cuando no existe suficiente información para probar que alguna acción será el resultado definitivo. Con esto Russell y Norvig introducen el concepto de red de creencia (también conocida como red bayesiana), la cual busca representar la dependencia entre variables y dar una especificación concisa sobre la distribución de la probabilidad conjunta. Básicamente una red de creencia es un grafo que contiene:

- Un conjunto de variables aleatorias conforman los nodos de la red.
- Un conjunto de enlaces dirigidos (o flechas) que conectan pares de nodos.
- Cada nodo posee una tabla de probabilidad condicional que cuantifica los efectos que los nodos padre tienen sobre el nodo que se está analizando.
- Dicho grafo no posee ciclos dirigidos.

Más a fondo, Luis Enrique Sucar (2012) define que las redes bayesianas modelan un fenómeno mediante un conjunto de variables y las relaciones de dependencia de estas. Dado este modelo, se puede hacer inferencia bayesiana; es decir, estimar la probabilidad posterior de las variables no conocidas, en base a las variables conocidas. En una RB todas las relaciones de independencia condicional representadas en el grafo corresponden a relaciones de independencia en la distribución de probabilidad. Dichas independencias simplifican la representación del conocimiento (menos parámetros) y el razonamiento (propagación de las probabilidades). Una red bayesiana representa en forma gráfica las dependencias e independencias entre variables aleatorias, en particular las independencias condicionales.

Figura 3. Ejemplo de una red bayesiana



Fuente: Sucar,2012

1. Inferencia. Sucar (2012) define que el razonamiento probabilístico o propagación de probabilidades consiste en propagar los efectos de la evidencia a través de la red para conocer la probabilidad a *posteriori*1 de las variables. Es decir, se le dan valores a ciertas variables (evidencia), y se obtiene la probabilidad posterior de las demás variables dadas las variables conocidas (el conjunto de variables conocidas puede ser vacío, en este caso se obtienen las probabilidades a *priori*2). Existen diferentes tipos de algoritmos para calcular las probabilidades posteriores, que dependen del tipo de grafo y de si obtienen la probabilidad de una variable a la vez o de todas. Sucar define los principales tipos de algoritmo de inferencia son:

- Una variable, cualquier estructura: algoritmo de eliminación (variable de eliminación).
- Cualquier variable, estructuras sencillamente conectadas: algoritmo de propagación de Pearl.
- Cualquier variable, cualquier estructura: (i) agrupamiento (*junction tree*), (ii) simulación estocástica, y (iii) condicionamiento.

Alex Smola y Vishwanathan (2010:20), menciona cuatro algoritmos de forma más específica, estos son: *Naive Bayes*3, *Nearest Neighbors*, clasificador promedio y el Perceptron. Debido a la orientación de esta investigación, se profundizará únicamente en el algoritmo Naive Bayes.

2. Clasificador Bayesiano (Naive Bayes). El algoritmo de clasificador bayesiano, está basado principalmente en el Teorema de Bayes, que en teoría de la probabilidad, es un resultado enunciado por Thomas Bayes en 1763 que expresa la probabilidad condicional de un evento aleatorio A dado B en términos de la distribución de probabilidad condicional del evento B dado A y la distribución de probabilidad marginal de sólo A. En términos matemáticos, el Teorema de Bayes se encuentra dado por la siguiente fórmula:

$$P(A_i \mid B) = (P(B \mid A_i) P(A_i)) / (P(B))$$

a. Aprendizaje de clasificadores bayesianos. Al profundizar en los clasificadores bayesianos, Sucar (2012:8) establece un clasificador bayesiano que tiene como labor suministrar una función que clasifica un dato, especificado por una serie de características o atributos en una o diferentes clases predefinidas.

Un clasificador bayesiano obtiene la probabilidad posterior de cada clase, Ci, usando la regla de Bayes, como el producto de la probabilidad a priori de la clase por la probabilidad condicional de los atributos (E) dada la clase, dividido por la probabilidad de los atributos:

---

[1]Tipo de conocimiento que, en algún sentido importante, depende de la experiencia.
[2]Tipo de conocimiento que, en algún sentido importante, es independiente de la experiencia.
[3]También conocido como Clasificador Bayesiano

$$E\frac{|C_i)}{P(E)}$$

El clasificador bayesiano simple (naive Bayes classifier, NBC) asume que los atributos son independientes entre sí dada la clase, así que la probabilidad se obtiene por el producto de las probabilidades condicionales individuales de cada atributo dado el nodo clase:

$$P(C_1i \mid E) = P(C_1i)P(E_1 1 \mid C_1i)P(E_1 2 \mid C_1i) \ldots (P(E_1 n \mid C_1i) \mid C_1i))/(P(E))$$

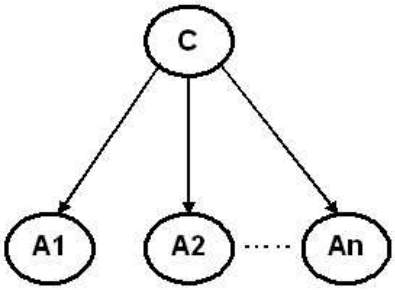
Donde n es el número de atributos. Esto hace que el número de parámetros se incremente linealmente con el número de atributos, en vez de hacerlo en forma exponencial. Gráficamente, un NBC se representa como una red bayesiana en forma de estrella, con un nodo de la raíz, C, que corresponde a la variable de la clase, que está conectada con los atributos, E1, E2, ..., En. Los atributos son condicionalmente independientes dada la clase, de tal manera que no existen arcos entre ellos. Dado que la estructura de un clasificador bayesiano simple está predeterminada, sólo es necesario aprender los parámetros asociados, que son:

$P(C)$: vector de probabilidades a priori para cada clase.

$P(E_1 i \mid C)$: matriz de probabilidad condicional para cada atributo dada la clase.

Estos parámetros se estiman fácilmente, a partir de los datos, en base a frecuencias. El denominador en la ecuación anterior no se requiere, ya que es una constante; es decir, no depende de la clase. Al final simplemente se normalizan las probabilidades posteriores de cada clase (haciendo que sumen uno). Aunque el clasificador bayesiano simple funciona muy bien (tiene una alta precisión en clasificación) en muchos dominios, en ocasiones su rendimiento decrece debido a que los atributos no son condicionalmente independientes como se asume. (Sucar, 2012:9)

Figura 4. Clasificador bayesiano simple.



Fuente: Sucar, 2012

Una vez definidos todos los conceptos relacionados a la elaboración de una red bayesiana, sus conceptos, características y fundamentos probabilísticos de ésta, es necesario definir los conceptos técnicos que la elaboración de un proyecto de redes bayesianas implica en cuestión de software y tecnología.

Q. Python

De acuerdo a la descripción que nos brinda la fundación The Python Software Foundation detrás del desarrollo y mantenimiento de esta herramienta,Python es un lenguaje de programación interpretado, orientado a objetos y de alto nivel con semánticas dinámicas. Su construcción de estructuras de datos de alto nivel, combinadas con su enlace dinámico, hacen de éste un lenguaje atractivo para el Rápido Desarrollo de Aplicaciones4 así como su uso en técnicas de scripting o integración de diferentes componentes en un lugar. Python soporta módulos y paquetes que motivan la modularidad del programa y la reutilización de código.

Fue principalmente el factor de facilidad de integración y scripting, lo que ha hecho que Python cuente con múltiples librerías de especialidades matemáticas, científicas, estadísticas, entre otras, siendo así la opción principal a utilizar en el desarrollo de este proyecto. Debido a que junto a esta herramienta era necesaria un manejador de la información que Python debía consultar, se estudió la opción de MongoDB.

R. MongoDB

De acuerdo a MongoDB, Inc. MongoDB es un sistema de base de datos basado en documentos, escalable, de alto rendimiento y NoSQL5. El nombre MongoDB proviene del término *hu**mongo**us* que es una referencia a algo enorme.

1. Documentos de la base de datos. Un registro en MongoDB es un documento, el cual es una estructura de datos compuesta de campos y pares de valores. Los documentos de MongoDB son similares a los objetos JSON6. Los valores de los campos pueden incluir otros documentos, arreglos, y arreglos de documentos.

---

[4] RAD por sus siglas en inglés *Rapid application development*, es un término utilizado para referirse a alternativas al método convencional de cascada en el desarrollo de software.

[5] Término interpretado generalmente como *Not Only SQL*, que representa una base de datos con mecanismos para el almacenamiento y obtención de datos que están modelados en términos distintos a las relaciones tabulares utilizadas en base de datos relacionales.

[6] JSON (*JavaScript Object Notation*) es un formato de peso liviano para el intercambio de datos.

Figura 5. Documento de MongoDB

Fuente: MongoDB, Inc.

MongoDB Inc, enumera las ventajas de utilizar documentos:

- Los documentos (objetos) corresponden a tipos de datos nativos en muchos lenguajes de programación.
- Los documentos incrustados y arreglos, reducen la necesidad de funciones join que pueden consumir muchos recursos.
- El esquema dinámico soporta polimorfismo7 fluido.

2. Características clave

a. Alto rendimiento. MongoDB provee persistencia de datos en alto rendimiento, particularmente:

- Soporta modelos de datos incrustados que reducen la actividad I/O8 en el sistema de base de datos.
- La indexación soporta consultas más rápidas y puede incluir llaves a partir de documentos incrustados y arreglos de datos.

b. Alta disponibilidad. Para proveer alta disponibilidad, MongoDB facilita la replicación, llamados conjuntos replica, provee:

- Conmutación por error automático.
- Redundancia de datos.

Un conjunto réplica es un grupo de servidores MongoDB que mantiene el mismo conjunto de datos, proporcionando redundancia e incrementando la disponibilidad de los datos. (MongoDB, Inc.)

---

[7] En programación orientada a objetos, es la propiedad por la que es posible enviar mensajes sintácticamente iguales a objetos de tipos distintos.

[8] *Input, Output* por sus siglas en inglés. Funciones de lectura y escritura de datos.

S. Minería de datos

La minería de datos es un conjunto de herramientas y técnicas de análisis de datos que por medio de la identificación de patrones extrae información interesante, novedosa y potencialmente útil de grandes bases de datos que puede ser utilizada como soporte para la toma de decisiones. (Calderón Méndez, 2006)

Si se analiza la definición anteriormente descrita, se dice que la minería de datos es un conjunto de herramientas y técnicas, una gran parte de estas técnicas son una combinación directa de madurez en tecnología de bases de datos y data warehousing, con técnicas de aprendizaje automático y de estadística (Fayyad & Smyth, 1996).

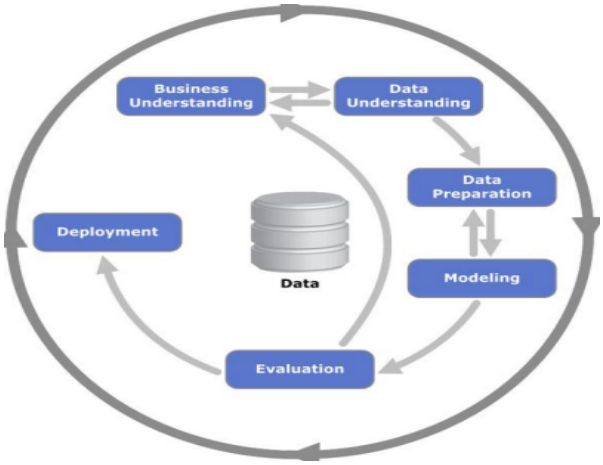
Para descubrir conocimiento de la información se pueden utilizar varias formas de análisis por medio de las cuales se puede llegar a identificar patrones y reglas en los datos para luego crear escenarios, esta información se puede representar por medio de modelos matemáticos sobre datos históricos y con esto se crea un modelo de minería de datos. Después de haber creado un modelo de minería de datos, se puede examinar nueva información a través del modelo evaluando si se apega a los patrones o reglas definidos. (Calderón Méndez, 2006)

1. Proceso de minería de datos. En la medida en que ha evolucionado e incrementado el uso de la minería de datos a nivel de instituciones y empresas, ha sido necesario definir una metodología que provea los lineamientos y mejores prácticas para llevar a cabo un proyecto de esta naturaleza.

En 1996, 3 empresas líderes de la industria (DaimlerBenz, SPSS y NCR) se unieron para realizar aportes en base a su experiencia en el tema de Data Mining y crearon un modelo de proceso llamado CRISP-DM (Cross Industry Standard Process for Data Mining), el cual describe una serie de fases y enfoques para abordar un proyecto de minería de datos (Mendoza Zapeta, 2013).

En la siguiente ilustración se puede observar el ciclo de vida de un proyecto de minería de datos que sigue la metodología CRISP-DM, este diagrama ayuda a entender las fases del proceso y provee una estrategia para ejecutar el proyecto. Las flechas entre las distintas fases indican las dependencias más importantes entre ellas. El círculo exterior enfatiza que es un proceso cíclico que se beneficiará de las experiencias aprendidas durante iteraciones anteriores.

Figura 6. Diagrama del proceso de minería de datos según el método CRISP-DM



Fuente: Mendoza Zapeta, 2013

- Conocimiento del negocio: en este paso se deben analizar los requisitos, analizar el contexto del problema, entender los objetivos del negocio desde una perspectiva no técnica y generar el plan del proyecto.

- Conocimiento de los datos: el objetivo de esta fase es familiarizarse con los datos tomando en cuenta los objetivos del negocio. Se realizan las siguientes tareas con respecto a los datos: recopilación inicial de datos, descripción, exploración y verificación de la calidad de los mismos.

- Preparación de los datos: se deben realizar las tareas necesarias para construir el conjunto final de datos que se utilizarán con las herramientas de modelado. Se seleccionan los datos, se identifica y corrige información faltante, se realiza depuración de aquellos datos que tengan atributos incorrectos. También se puede realizar el enriquecimiento de datos, que consiste en agregar atributos a los datos ya existentes, con el propósito de satisfacer los requisitos del proyecto de minería de datos. Las técnicas de limpieza, transformación y reducción del número de dimensiones de los datos permiten asegurar la calidad de los mismos.

- Modelado: en este momento se seleccionan y aplican las técnicas de minería de datos, ajustando los parámetros a sus valores óptimos. Se generan pruebas, crean los modelos y se interpretan los resultados.

- Evaluación: a partir de los modelos generados en la fase anterior, se debe evaluar si estos son útiles y cumplen con los objetivos del negocio establecidos. Se realiza la evaluación de los resultados, revisión del proceso y se establecen las siguientes acciones a realizar.

- Despliegue: una vez que se han validado los modelos, es crucial explotar la utilidad de los mismos, integrándolos a las tareas de toma de decisiones del negocio. En esta fase se planifica el despliegue del proyecto de minería de datos, el monitoreo y mantenimiento del mismo. Se genera un informe final y se realiza una revisión del proyecto. (Mendoza Zapeta, 2013)

2. Reconocimiento de patrones. Un patrón es una entidad a la que se le puede dar un nombre y que está representada por un conjunto de propiedades medidas y las relaciones entre ellas (vector de características). Por ejemplo, un patrón puede ser una señal sonora y su vector de características el conjunto de coeficientes espectrales extraídos de ella (espectrograma). Otro ejemplo podría ser una imagen de una cara humana de las cuales se extrae el vector de características formado por un conjunto de valores numéricos calculados a partir de la misma. El reconocimiento automático, descripción, clasificación y agrupamiento de patrones son actividades importantes en una gran variedad de disciplinas científicas, como biología, sicología, medicina, visión por computador, inteligencia artificial, teledetección, etc. (Hernandez & Ferri, 2007)

Un sistema de reconocimiento de patrones tiene uno de los siguientes objetivos:

- Identificar el patrón como miembro de una clase ya definida (clasificación supervisada).
- Asignar el patrón a una clase todavía no definida (clasificación no supervisada, agrupamiento o clustering). (Romero & Calonge, 2009)

El reconocimiento de patrones es el estudio de cómo las máquinas pueden observar el ambiente o entorno, aprender a distinguir comportamientos de interés a partir de la experiencia adquirida, y tomar decisiones razonables con respecto a las categorías a las que pertenecen dichos patrones. No hay que dejar atrás que, el mejor reconocedor de patrones conocido hasta ahorita es el ser humano. El Reconocimiento óptico de caracteres (OCR) es uno de los tópicos más antiguos dentro del Reconocimiento de Patrones y una de las áreas de investigación más importante y activa, que en la actualidad presenta desafíos: la precisión en el reconocimiento asociada tanto a caracteres impresos en una imagen degradada o a caracteres manuscritos es aún insuficiente, existiendo errores en el reconocimiento (Seijas, 2011)

Como ya se ha comentado, las técnicas de la minería de datos provienen de la inteligencia artificial y de la estadística, dichas técnicas, no son más que algoritmos, más o menos sofisticados que se aplican sobre un conjunto de datos para obtener unos resultados (García & Azaustre, 2008).

a. Fases para reconocimiento de patrones. El diseño de un sistema de reconocimiento de patrones se lleva a cabo normalmente en tres fases:

1) Adquisición y pre proceso de datos:
   - Se filtran los datos (de forma que se eliminan valores incorrectos, no válidos, desconocidos; según las necesidades y el algoritmo a usar).
   - Se obtienen muestras de los mismos (en busca de una mayor velocidad de respuesta del proceso).
   - Se reducen el número de valores posibles (mediante redondeo, clustering, etc.) (Hernández, 2004).

b. Extracción de características. Mediante una técnica de minería de datos, se obtiene un modelo de conocimiento, que representa patrones de comportamiento observados en los valores de las variables del problema o relaciones de asociación entre dichas variables. También pueden usarse varias técnicas a la vez para generar distintos modelos, aunque generalmente cada técnica obliga a un pre procesado diferente de los datos (Hernández, 2004).

c. Toma de decisiones o agrupamiento. Una vez obtenido el modelo, se debe proceder a su validación, comprobando que las conclusiones que arroja son válidas y suficientemente satisfactorias. En el caso de haber obtenido varios modelos mediante el uso de distintas técnicas, se deben comparar los modelos en busca de aquel que se ajuste mejor al problema (Hernández, 2004).

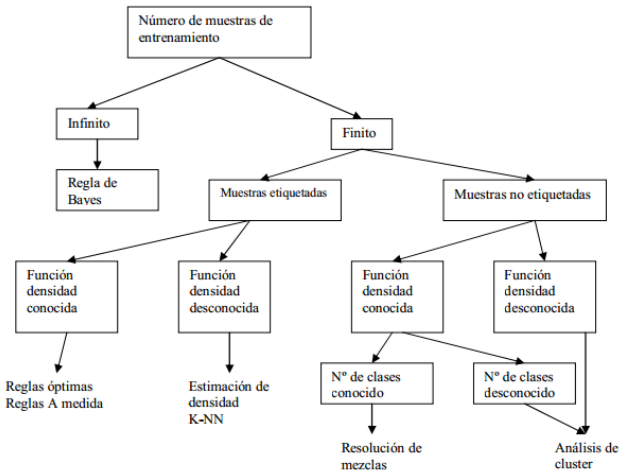
d. Reconocimiento estadístico de patrones. El REP es una disciplina relativamente madura hasta el punto de que existen ya en el mercado un cierto número de sistemas comerciales de reconocimiento de patrones que emplean esta técnica. En REP, un patrón se representa por un vector numérico de dimensión n. De esta forma, un patrón es un punto en un espacio n-dimensional (de características). Un REP funciona en dos modos diferentes: entrenamiento y reconocimiento. En modo de entrenamiento, se diseña el extractor de características para representar los patrones de entrada y se entrena al clasificador con un conjunto de datos de entrenamiento de forma que el número de patrones mal identificados se minimice. En el modo de reconocimiento, el clasificador ya entrenado toma como entrada el vector de características de un patrón desconocido y lo asigna a una de las clases o categorías. (Romero & Calonge, 2009)

El proceso de toma de decisiones en un REP se puede resumir como sigue. Dado un patrón representado por un vector de características

$$X = (x_1, x_2, \ldots, x_n)^T$$

Asignarlo a una de las c clases o categorías $C_1, C_2, \ldots, C_c$. Dependiendo del tipo de información disponible sobre las densidades condicionales de las clases, se pueden diseñar varias estrategias de clasificación. Si todas las densidades condicionales $p(x|C_i), i = 1,2,\ldots,c$ se conocen, la regla de decisión es la de Bayes que establece los límites entre las diferentes clases. Sin embargo, en la práctica las densidades condicionales no se conocen y deben ser estimadas (aprendidas) partiendo de los patrones de entrada. Si se conoce la forma funcional de estas densidades pero no sus parámetros, el problema se llama de toma de decisión paramétrico. En caso contrario, estamos ante un problema de toma de decisión no paramétrico. Las diferentes dicotomías que aparecen al diseñar un sistema de REP se muestran en la siguiente figura..

Figura 7. Diferentes dicotomías al diseñar un sistema REP



Fuente: Romero & Calonge, 2009

3. Tipos de aprendizajes

a. Aprendizaje supervisado. En el aprendizaje supervisado o aprendizaje a partir de ejemplos, el instructor o experto define clases y provee ejemplos de cada una. El sistema debe obtener una descripción para cada clase. Cuando el instructor define una única clase, provee ejemplos positivos (pertenecen a la clase) y negativos (no pertenecen a la clase). En este caso, los ejemplos importantes son los cercanos al límite, porque proveen información útil sobre los límites de la clase. Cuando el instructor define varias clases, el sistema puede optar por realizar descripciones discriminantes o no. Un conjunto de descripciones es discriminante si el total de las descripciones cubren todas las clases, pero una descripción cubre una sola clase en particular (Servante, 2002).

b. Aprendizaje no supervisado. En el aprendizaje no supervisado o aprendizaje a partir de observaciones y descubrimientos, el sistema debe agrupar los conceptos sin ayuda alguna de un

instructor. El sistema recibe los ejemplos, pero no se predefine ninguna clase. Por lo tanto, debe observar los ejemplos y buscar características en común que permitan formar grupos. Como resultado, este tipo de aprendizaje genera un conjunto de descripciones de clases, que juntas cubren todas las clases y en particular describen a una única clase (Servante, 2002).

c. R como herramienta para la minería de datos. La herramienta que se seleccionó para realizar el análisis de datos se llama R, un software estadístico creado por Ross Ihaka y Robert Gentleman de la Universidad de Aukland en Nueva Zelandia y está diseñado para el análisis de datos, gráficos y análisis estadísticos.

La elección de la plataforma computacional R para el desarrollo de esta tesis se debe a que es una plataforma bastante conocida, tiene una facilidad para programar, tiene un buen rendimiento, extensa documentación y su uso está siendo ampliado a campos como bioinformática, finanzas entre otros. A continuación se detallan otras ventajas adicionales y también desventajas que posee el programa R.

1) Ventajas
   - Software de código abierto y multiplataforma (Windows, Linux y MacOS).
   - Completamente programable y extensible por medio de instalación de paquetes que proveen flexibilidad en el análisis de datos.
   - Existe amplia documentación para la utilización de R para minería de datos.
   - Tiene una comunidad de desarrollo activa, que actualiza constantemente el software.
   - Existen actualmente 6 paquetes especializados que incluyen alrededor de 40 algoritmos implementados para el software R, que permiten desarrollar técnicas de data miningcomo: reducción de dimensionalidad, clasificación, Clustering o segmentación de datos, asociación.

2) Desventajas
   - Debido a que la interacción con el usuario se realiza por medio de una interfaz de comandos y no una interfaz gráfica, es necesario conocer o tener un documento de referencia de los comandos que se desean utilizar.
   - Requiere invertir un considerable tiempo inicial para obtener resultados observables.

T. Estadística

La estadística "es la ciencia que tiene que ver con la recolección, organización, presentación, análisis e interpretación de datos" (Webster, 2000).

La estadística según Montiel "desde el punto de vista más amplio, cabe definir la estadística como la ciencia que estudia cómo debe emplearse la información y que pretende dar una guía de acción en situaciones prácticas que entrañan incertidumbre" (Montiel, Rius, & Barón, 1997).

De acuerdo a las definiciones anteriores podemos decir que la estadística se ocupa de los métodos y procedimientos para recoger, clasificar, resumir, hallar regularidades y analizar los datos, siempre y cuando la variable e incertidumbre sea una causa intrínseca de los mismos; así como de realizar inferencias a partir de ellos, con la finalidad de ayudar a la toma de decisiones y en su caso formular predicciones. (Rodríguez Ortiz, 2007)

El campo estadístico es de suma utilidad en el análisis de datos y tendencias. En resumen, este campo tiene dos grandes ventajas:

- Toma de decisiones más acertadas basándose en, datos numéricos, modelos matemáticos y probabilísticos.

- Encontrar distintos tipos de soluciones de problemas, en una forma práctica con un nivel de incertidumbre menor que la forma empírica.

Dentro de la estadística existen diferentes métodos dependiendo el tipo de análisis y el tipo de los datos que se deseen, entre estos existen dos tipos:

1. Estadística descriptiva. Es el proceso de recolectar, agrupar y presentar datos de una manera tal que describa fácil y rápidamente dichos datos. (Morales Peña, 2011). Se analiza a través de las medidas de tendencia central, las medidas de dispersión y las de posición. Otros autores señalan que la estadística descriptiva: describe, analiza y representa un grupo de datos utilizando métodos numéricos y gráficos que resumen y presentan la información contenida en ellos. (Mills, 1981).

2. Medidas de tendencia central. La posición o "tendencia central" de una distribución, se refiere al lugar donde se centra una distribución particular en la escala de valores. Hay tres tipos de medidas de tendencia central:

- Moda: Se define la moda, como el valor que se presenta u ocurre con mayor frecuencia. Es decir la Moda es el valor más común de una distribución. La moda puede no existir en una distribución determinada o bien puede ser única. En una representación gráfica, la moda será el rectángulo más alto, en el caso de un histograma o el pico más alto en el caso de un polígono (García Ferrando, 1989). Si aparecen varias modas, se llamará multimodal, si solo se observan dos, se llamará bimodal, y por último, si sólo observamos una moda será unimodal. Cuando los datos están agrupados en intervalos, a la clase (intervalo) que contiene la moda, se llama clase modal. La moda será el punto intermedio de esa clase modal.

Figura 8. Ejemplo del concepto de moda.

Grupo a	2	3	3	3	5	5	**N=6**
Grupo b	2	2	4	5	5	6	**N=6**

Fuente: García Ferrando, 1989

- Mediana: La mediana es el punto o valor numérico que deja por debajo (y por encima) a la mitad de las puntuaciones de una distribución, o sea, el 50% de valores. Si el número de puntuaciones es impar, el valor de la mediana, es el valor que queda en el centro exactamente (Freud Williams, 1990). De esta forma, la mediana se calcula de la siguiente forma:

Figura 9. Ejemplo de la mediana en set de datos impares.

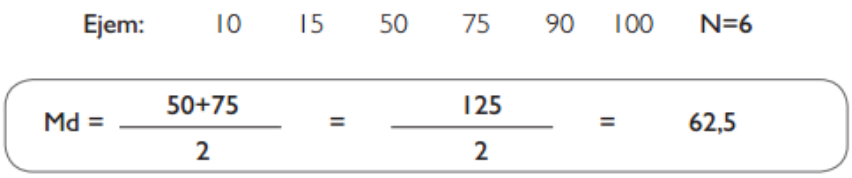
Ejem: 5 6 7 8 9 **N=5**

$$K = \frac{N+1}{2} = \frac{5+1}{2} = 3$$

Fuente: García Ferrando, 1989

En el caso de que la distribución de puntuaciones fuera par, tendríamos que coger las dos puntuaciones centrales, sumarlas y dividirlas entre dos.

Figura 10. Ejemplo de la mediana en set de datos pares

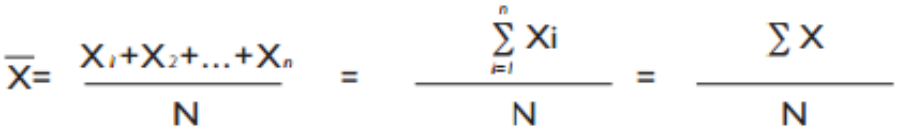


Fuente: García Ferrando, 1989

En este caso la mediana es un valor o puntuación que se encuentra entre las dos puntuaciones centrales de la distribución y se obtiene mediante esta operación y cuyo resultado da exactamente el valor de la mediana.

- Media aritmética: La media aritmética es la suma de todas las puntuaciones de una distribución dividida por el número total de casos. La siguiente ilustración muestra la expresión matemática de la media aritmética, siendo X es la media aritmética, X son los diferentes datos y N es el total de datos proporcionados.

Figura 11. Expresión matemática para la media aritmética



Fuente: García Ferrando, 1989

Es la medida más utilizada, debido a que posee una serie de ventajas, es un buen ejemplo del uso de las razones como una forma válida para hacer comparaciones. De alguna forma realizamos una estandarización, esto permite hacer comparaciones de medias de grupos de diferente tamaño (Berenson, Levine, & Krehbiel, 2001).

Es la medida más utilizada, debido a que posee una serie de ventajas, es un buen ejemplo del uso de las razones como una forma válida para hacer comparaciones. De alguna forma realizamos una estandarización, esto permite hacer comparaciones de medias de grupos de diferente tamaño (Berenson, Levine, & Krehbiel, 2001).

En el caso del cálculo de la media, si nos encontramos con el problema de las puntuaciones extremas. Las puntuaciones con valores más altos tendrán más peso en la distribución, contribuyen en mayor medida a la suma de las puntuaciones que los valores más bajos de la distribución.

Figura 12. Ejemplo de aplicación de la media aritmética

Ejemplo a	2	2	4	6	8	14	20	56/7	$\bar{X}$=8
Ejemplo b	2	2	4	6	8	14	30	66/7	$\bar{X}$=9,4

Fuente: Berenson, Levine &Krehbiel, 2001.

3. Medidas de dispersión. Una segunda propiedad importante para describir un conjunto de datos numéricos es la variación. La variación, es la cantidad de dispersión o separación, que presentan los datos. Dos conjuntos de datos pueden diferir tanto en la tendencia central como en la variación. (Berenson, Levine, & Krehbiel, 2001).

En diversos textos de estadística se hace referencia a la dispersión o variabilidad como la razón de ser de esta disciplina; por ejemplo, “Statistics is about variation.” (De Veaux, Bock, & Velleman, 2003). Así afirma este autor, y en efecto, si no existiese heterogeneidad o dispersión en las variables que estudiamos, sería muy fácil resumir la información de las mismas, no haciendo ninguna falta los métodos estadísticos.

Las medidas de variación incluyen:

- Recorrido: El recorrido es la diferencia entre los datos mayor y menor del conjunto. También se le suele llamar rango. En un conjunto de datos, mientras mayor sea el rango, mayor será su dispersión y, a la inversa, mientras menor sea su rango, menor su dispersión (Contreras López, 2014).

- Rango intercuartil: Es la diferencia entre el primer y tercer cuartil en un conjunto de datos (Berenson, Levine, & Krehbiel, 2001).

- Desviación estándar: Se llama desviación estándar, porque con ella se pueden estandarizar en todos los casos, todas las desviaciones de datos recolectados. La desviación estándar se simboliza con la letra griega σ si se trata de una población y con la letra S si se trata de una

muestra. Cuando los datos están ordenados en una distribución de frecuencias simples, la desviación estándar para una población se calcula mediante la fórmula:

$$\sigma = \sqrt{\frac{\sum f(x-\bar{x})^2}{n}}$$

Siendo:

σ = desviación estándar de la población

f = frecuencia

x = valor nominal

$\bar{x}$ = media aritmética

Cuando los datos están ordenados en una distribución de frecuencias simples, la desviación estándar para una muestra se calcula mediante la fórmula:

$$\sigma = \sqrt{\frac{\sum f(x-\bar{x})^2}{n-1}}$$

Siendo:

σ = desviación estándar de la población

f = frecuencia

x = valor nominal

$\bar{x}$ = media aritmética

- Coeficiente de variación: Es una indicación relativa de la variación, siempre se expresa como un porcentaje, no en términos de las unidades de los datos específicos. El CV mide la dispersión de los datos con relación a la media y se calcula dividiendo la desviación estándar entre la media aritmética, multiplicada por cien (Berenson, Levine, & Krehbiel, 2001).

4. Medidas de posición. Los cuartiles son las medidas de posición "no central" que se utilizan con mayor frecuencia, también se les llama cuantiles, y se emplean sobre todo para resumir o describir las propiedades de conjuntos grandes de datos numéricos. Los cuartiles son medidas descriptivas que parten los datos ordenados en cuatro cuartos. Otros cuartiles que se utilizan a menudo son los deciles, que separan los datos ordenados en diez partes, y los Percentiles, que los dividen en cien partes (Rodríguez Ortiz, 2007).

U. Estadística inferencial

Apoyándose en el cálculo de probabilidades y a partir de datos muestrales, efectúa estimaciones, decisiones, predicciones y otras generalizaciones sobre un conjunto mayor de datos (Mills, 1981). Se analiza a través de la estimación de parámetros y la prueba de hipótesis.

En general, la Inferencia Estadística es una lógica que permite hacer afirmaciones acerca de las características de una población cuando sólo existen datos parciales o muestrales. La estadística Inferencial aborda los temas:

- Inferencia
- Pruebas de hipótesis

V. Regresión lineal

La regresión utiliza valores existentes para pronosticar qué valores son los que se obtendrán más adelante. En un caso simple de regresión se utilizan técnicas estadísticas como la regresión lineal, desafortunadamente muchos problemas de la vida real no son simples proyecciones lineales de los valores previos. Por ejemplo los rangos de fallo en volúmenes de ventas de un determinado stock de productos son bastante difíciles de predecir porque dependen de la interacción de múltiples variables de predicción (Calderón Méndez, 2006). Se adapta a una amplia variedad de situaciones. En la investigación social, el análisis de regresión se utiliza para predecir un amplio rango de fenómenos, desde medidas económicas hasta diferentes aspectos del comportamiento humano. En el contexto de la investigación de mercados puede utilizarse para determinar en cuál de diferentes medios de comunicación puede resultar más eficaz invertir; o para predecir el número de ventas de un determinado producto. En física se utiliza para caracterizar la relación entre variables o para calibrar medidas.

La regresión lineal puede ser dividida, según las variables en dos:

- Regresión simple
- Regresión múltiple

Tanto en el caso de dos variables, regresión simple, como en el de más de dos variables, regresión múltiple, el análisis de regresión lineal puede utilizarse para explorar y cuantificarla relación entre una variable llamada dependiente o criterio Y y una o más variables llamada sin dependientes o predictoras $X_1, X_2, \ldots X_n$, así como para desarrollar una ecuación lineal con fines predictivos. Además, el análisis de regresión lleva asociados una serie de procedimientos de diagnóstico que informan sobre la estabilidad e idoneidad del análisis y que proporcionan pistas sobre cómo perfeccionarlo (Complutense, 2008).

1. Regresión simple. La regresión lineal tiene como objetivo el estudiar cómo los cambios en una variable, no aleatoria, afectan a una variable aleatoria, en el caso de existir una relación funcional entre ambas variables que puede ser establecida por una expresión lineal, es decir, su representación gráfica es una línea recta (Arroyo Cervantes & Camacho Castillo).

Cuando la relación lineal concierne al valor medio o esperado de la variable aleatoria, estamos ante un modelo de regresión lineal simple. La respuesta aleatoria al valor x de la variable controlada se designa por Yx y, según lo establecido, se tendrá:

$$Y_i = \alpha + \beta x_i + \varepsilon_i$$

Donde α y β son coeficientes de regresión.

2. Regresión múltiple. El análisis de regresión múltiple generaliza el modelo de regresión lineal simple permitiendo agregar términos adicionales al intercepto y la pendiente en la función media. Mediante un modelo de regresión lineal múltiple (MRLM) tratamos de explicar el comportamiento de una determinada variable que se denomina variable a explicar, variable endógena o variable dependiente, en función de un conjunto de k variables explicativas mediante una relación de dependencia lineal (Kizys & Juan, 2005)

$$Y = \beta_1 + \beta_2 x_2 + \ldots + \beta_k x_k + \varepsilon$$

Donde los β son coeficientes de regresión y ε es el error.

W. Regresión logística

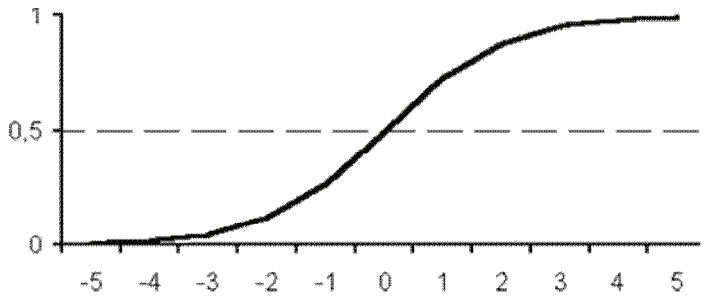
La regresión logística es una técnica estadística multivariable que nos permite estimar la relación existente entre una variable dependiente no métrica, en particular dicotómica y un conjunto de variables independientes métricas o no métricas.

El Análisis de Regresión logística tiene la misma estrategia que el Análisis de Regresión Lineal Múltiple, el cual se diferencia esencialmente del Análisis de Regresión logística por que la variable dependiente es métrica; en la práctica el uso de ambas técnicas tienen mucha semejanza, aunque sus enfoques matemáticos son diferentes. (Hosmer & Lemeshow, 2000)

La variable dependiente o respuesta no es continua, sino discreta (generalmente toma valores 1,0). Las variables explicativas pueden ser cuantitativas o cualitativas; y la ecuación del modelo no es una función lineal de partida, sino exponencial; si bien, por sencilla transformación logarítmica, puede finalmente presentarse como una función lineal. Así pues el modelo será útil en frecuentes situaciones

prácticas de investigación en que la respuesta puede tomar únicamente dos valores: 1, presencia con probabilidad p; y 0, ausencia con probabilidad 1-p (Salcedo Poma, 2002).

Figura 13. Forma gráfica de la función logística.



Fuente: Salcedo Poma, 2002

1. Finalidad de la regresión logística. El objetivo primordial de esta técnica es el de modelar cómo influyen las variables regresoras en la probabilidad de ocurrencia de un suceso particular.

Sistemáticamente tiene dos objetivos:

- Investigar cómo influye en la probabilidad de ocurrencia de un suceso, la presencia o no de diversos factores y el valor o nivel de los mismos.
- Determinar el modelo más parsimonioso y mejor ajustado que siendo razonable describa la relación entre la variable respuesta y un conjunto de variables regresoras.

2. Comparación de regresión logística con otros métodos. El análisis de regresión lineal múltiple y el análisis discriminante son dos métodos eficaces pero plantean problemas cuando la variable respuesta es binaria. En el análisis de regresión lineal múltiple cuando la variable respuesta toma solo dos valores, se violan los supuestos de necesarios para efectuar inferencias, los problemas que se plantean son:

- La distribución de los errores aleatorios no es normal.
- Los valores predichos no pueden ser interpretados como probabilidades como en la regresión logística, porque no toman valores dentro del intervalo [0,1].

El análisis discriminante permite la predicción de pertenencia de la unidad de análisis a uno de los dos grupos pre-establecidos, pero se requiere que se cumplan los supuestos de multinormalidad de las variables regresoras y la igualdad de matrices de covarianzas de los dos grupos, pueden ser diferentes también; para que la regla de predicción sea óptima (Cook & S., 1982).

X. Árboles de decisión

Un árbol de decisión es un modelo de predicción utilizado en el ámbito de la inteligencia artificial, dada una base de datos se construyen estos diagramas de construcciones lógicas, muy similares a los sistemas de predicción basados en reglas, que sirven para representar y categorizar una serie de condiciones que suceden de forma sucesiva, para la resolución de un problema (García & Azaustre, 2008).

El aprendizaje de árboles de decisión está englobado como una metodología del aprendizaje supervisado. La representación que se utiliza para las descripciones del concepto adquirido es el árbol de decisión, que consiste en una representación del conocimiento relativamente simple y que es una de las causas por la que los procedimientos utilizados en su aprendizaje son más sencillos que los de sistemas que utilizan lenguajes de representación más potentes, como redes semánticas, representaciones en lógica de primer orden etc. No obstante, la potencia expresiva de los árboles de decisión es también menor que la de esos otros sistemas. El aprendizaje de árboles de decisión suele ser más robusto frente al ruido y conceptualmente sencillo, aunque los sistemas que han resultado del perfeccionamiento y de la evolución de los más antiguos se complican con los procesos que incorporan para ganar fiabilidad. La mayoría de los sistemas de aprendizaje de árboles suelen ser no incrementales, pero existe alguna excepción (Utgoff, 1988).

1. Algoritmo ID3. Este sistema ha sido el que más impacto ha tenido en la Minería de Datos. Desarrollado en los años ochenta por Quinlan, ID3 significa Induction Decision Trees, y es un sistema de aprendizaje supervisado que construye árboles de decisión a partir de un conjunto de ejemplos. Estos ejemplos son tuplas compuestas por varios atributos y una única clase. El dominio de cada atributo de estas tuplas está limitado a un conjunto de valores. Las primeras versiones del ID3 generaban descripciones para dos clases: positiva y negativa. En las versiones posteriores, se eliminó esta restricción, pero se mantuvo la restricción de clases disjuntas. ID3 genera descripciones que clasifican cada uno de los ejemplos del conjunto de entrenamiento (Servante, 2002).

Este sistema tiene una buena performance en un amplio rango de aplicaciones, entre las cuales podemos nombrar, aplicaciones de dominios médicos, artificiales y el análisis de juegos de ajedrez. El nivel de precisión en la clasificación es alto. Sin embargo, el sistema no hace uso del conocimiento del dominio. Además, muchas veces los árboles son demasiado frondosos, lo cual conlleva a una difícil interpretación. En estos casos pueden ser transformados en reglas de decisión para hacerlos más comprensibles. (Servante, 2002)

2. Algoritmo C4.5. El C4.5 es una extensión del ID3 que permite trabajar con valores continuos para los atributos, separando los posibles resultados en dos ramas: una para aquellos A_i<=N y otra para A_i>N. Este algoritmo fue propuesto por Quinlan en 1993. El algoritmo C4.5 genera un árbol de decisión a partir de los datos mediante particiones realizadas recursivamente. El árbol se construye mediante la estrategia de profundidad-primero (depth-first). El algoritmo considera todas las pruebas posibles que pueden dividir el conjunto de datos y selecciona la prueba que resulta en la mayor ganancia de información. Para cada atributo discreto, se considera una prueba con n resultados, siendo n el número de valores posibles que puede tomar el atributo. Para cada atributo continuo, se realiza una prueba binaria sobre cada uno de los valores que toma el atributo en los datos. (Servante, 2002)

3. CART. Partimos de una muestra de entrenamiento donde cada Xi es un vector con p variables aleatorias, que pueden ser todas continuas, todas discretas o mezclas de ambas.

$$[(X]_1, Y_1), [(X]_2, Y_2), \ldots, [(X]_n, Y_n)$$

Las variables Yi son unidimensionales, discretas o continuas. Con la muestra de entrenamiento, construimos una estructura del tipo árbol en dos etapas bien diferenciadas, en la primera, determinamos el llamado árbol máximo y en la segunda, aplicamos un procedimiento denominado de poda.

Para construir el árbol máximo, comenzamos con toda la muestra en el nodo raíz y vamos obteniendo los nodos interiores por particiones sucesivas, mediante una cierta pregunta o regla que involucra a uno de los p atributos. Se trata de árboles binarios, por lo cual en función de la respuesta, cada nodo se parte en dos nodos hijos. Por convención, asignamos el nodo izquierdo al caso afirmativo y el derecho contrario. Por último, se elige algún criterio de parada, para saber cuando un nodo deja de partirse y queda constituyendo un nodo terminal llamado hoja. (Roche, 2009)

Una de las características fundamentales de CART, es que luego de obtenido un árbol máximo se inicia una etapa de poda, en la cual se eliminan algunas de sus ramas. (Roche, 2009)

Y. Clustering

Clustering es la tarea de agrupar conjuntos de objetos, de manera que objetos dentro del mismo grupo sean más similares entre sí en comparación con los objetos de otros grupos, de acuerdo a alguna métrica en particular (Martinez, 2012).

Las definiciones formales varían entre los diferentes algoritmos y campos de estudio, ya que la noción de clustering no es uniforme entre los diferentes trasfondos que la aplican. Por lo tanto, lo que constituye un clúster puede variar entre algoritmos o autores, e incluso los propósitos para utilizar clustering se pueden tomar desde enfoques distintos. En minería de datos, el interés está en la generación de grupos

resultantes a través de agrupamiento, mientras que en clasificación automática el interés está en poder discriminar a qué grupo pertenece un objeto particular (Alejandro, Vega, & Ruiz, 2012).

El principal propósito de la clasificación no supervisada consiste formar grupos de objetos tomando como base la semejanza. Se desea descubrir clases "significativas" (grupos, conglomerados, clústers); pero no se posee información sobre las características de estas clases, o incluso cuál es su cantidad. Por tanto, en este caso las cuestiones de qué y cómo hacerlo resultan de interés. Esto apunta al cuidado que se debe tener en cuanto a la interpretación de los resultados que se obtienen a luz de las suposiciones hechas y de las limitaciones de las técnicas que se utilicen. (Hartigan, 1975)

1. Algoritmo de K-medias. El análisis de conglomerados de K medias es un método de agrupación de casos que se basa en las distancias existentes entre ellos en un conjunto de variables. Versiones anteriores del procedimiento comenzaban el análisis con la asignación de los K primeros casos a los centros de los K conglomerados (los centros multivariantes de los conglomerados se denominan centroides). En la versión actual se comienza seleccionado los K casos más distantes entre sí (el usuario debe determinar inicialmente el número K de conglomerados que desea obtener). Y a continuación se inicia la lectura secuencial del archivo de datos asignando cada caso al centro más próximo y actualizando el valor de los centros a medida que se van incorporando nuevos casos. Una vez que todos los casos han sido asignados a uno de los K conglomerados, se inicia un proceso iterativo para calcular los centroides finales de esos K conglomerados. (Universidad, 2008)

V. ANTECEDENTES

En el ámbito internacional, se han encontrado estudios e investigaciones basadas en técnicas de clasificación supervisada aplicadas en diversas áreas distintas a la línea de interés de este trabajo. Se puede hacer mención del trabajo realizado por Martínez, F., Díaz, M.C., Martín, M.T., Rivas, V.M. y Ureña, L.A. (2002), en el cual estudia la aplicación de redes neuronales y redes bayesianas en la detección de multipalabras para tareas IR. La red neuronal en este caso, se utiliza para clasificar parejas de términos de los cuales se busca determinar si son o no multipalabras. Por otro lado, se utiliza la red bayesiana para obtener la confianza en que las parejas de términos sean multipalabras.Esta clasificación se basó en diferentes estimadores utilizados como entradas a las dos redes. El resultado obtenido en la clasificación, se utilizó en tareas de recuperación de información.

La implementación que utilizó este estudio para la red bayesiana, se basó en nodos cuyos valores eran conocidos y se propagó el conocimiento a través de la red mediante ciertas reglas probabilísticas. La red bayesiana consistía de cinco nodos de forma lineal, es decir que cada uno de éstos se utilizaba como nodo de entrada para el nodo final que en este caso sería el de Multipalabra. Para el entrenamiento de la red bayesiana se utilizaron listas de palabras de las cuales se conocían eran multipalabras y se determinó la información necesaria de las mismas para contemplar los nodos. A pesar que los resultados que expone el estudio establecen que estos métodos propuestos son útiles en la mejora de la precisión alcanzada por un sistema IR, siendo también la red bayesiana la que mejores resultados ofrece. Sin embargo, una de las principales características de la red bayesianas es que está diseñada para tener múltiples nodos de entrada y que con base a estos se pueda inferir una mayor cantidad de información de la que actualmente se tiene. Es por ello que es importante procurar que una red bayesiana no sea plana, ya que no se aprovecharía las diferentes características que ésta pueda proporcionar.

Por otro lado, López J. y García J. (2008), demuestran las capacidades de una red bayesiana para mejorar el desarrollo de sistemas de tutorización inteligente. En dicho estudio, se profundiza en los conceptos que una red bayesiana permite explorar y cómo estos pueden intervenir de forma positiva a diversos sistemas de tutorización inteligente de forma que, con el uso de redes bayesianas, se puede modelar un flujo de aprendizaje similar al que las personas tienen. Por esto, desarrollar una red bayesiana basada en el conocimiento que poseen las personas acerca del "¿cómo aprendemos?" alineado con los conceptos psicológicos básicos, mejoraría el grado de efectividad de estos sistemas de tutorización.

Otras investigaciones como Abbasi, A. *et al* (2012:2) se enfoca en el fraude financiero que ha ocurrido en distintas partes del mundo en distintas empresarias bancarias que trabajan con sistemas transaccionales, e indica que en la mayoría de estudios en Estados Unidos las tasas de detección son menores al 70%, identificando que la confidencialidad de la información del sector privado de sus bases de

datos de transacciones fraudulentas representa un problema para las investigaciones que no tienen acceso a esta información. Abbasi, A. *et al* (2012:2) indica que en un ambiente en donde los fraudes financieros aún no pueden ser detectados en su totalidad con la información escasa que se encuentra pública para las instituciones interesadas, los sistemas de detección son soluciones que pueden asistir en el descubrimiento y localización de transacciones fraudulentas de manera confiable. Chan, P.K. *et al* (1999:1) identifica otro problema existente con la detección de fraudes descrito como el sesgo existente en la información estudiada, *i.e.*, hay una mayor cantidad de transacciones legítimas en comparación a la cantidad de transacciones fraudulentas. Esto podría afectar la tarea de predicción de fraudes al pronosticar transacciones legítimas y obtener un alto grado de exactitud en los resultados, *i.e*, en un ambiente donde la proporción de transacciones fraudulentas contra transacciones legítimas es de 1:1000, las técnicas de identificación pueden sesgarse ante la mayoría de los casos. Estudios tomados desde 1995 hasta el 2011 por Abbasi, A. *et al* (2012:4) muestran una selección heterogénea de técnicas utilizadas para detectar fraudes, entre las principales se encuentran: regresión logística, redes bayesianas, redes neuronales y, entre las más efectivas, support vector machines con modificaciones.

En el campo de la seguridad informática, Zurutuza U. y UribeetxeberriaR. (2005) evalúan el estado actual del uso de minería de datos para la detección de intrusiones por medio de dispositivos de seguridad (IDS[9]). En este estudio, se analizan múltiples métodos de minería de datos, entre ellos las redes bayesianas que son analizados como clasificadores bayesianos, cuya función es presentar las reglas o firmas de los IDS en forma de relaciones de probabilidad condicional, permitiendo así definir modelos de decisión y razonamiento bajo incertidumbre. El estudio concluyó que es difícil determinar un algoritmo que se adecue de mejor forma para todos los escenarios de la detección de intrusiones debido a que el desempeño de las técnicas de clasificación depende mucho del problema a resolver. Es por ello que muchas veces sería más importante seleccionar adecuadamente las características que intervienen en el proceso de aprendizaje y clasificación o agrupamiento, que el propio algoritmo o técnica que se vaya a utilizar.

Un área que continuamente ha mostrado interés en la investigación y el estudio de algoritmos de minería de datos es la medicina. Abad-Grau, M., Ierache J. y Cervino, C. han realizado un estudio que inspecciona la aplicación de redes bayesianas en sistemas expertos de triaje[10] en servicios de urgencias médicas. Esta investigación contempló múltiples algoritmos de aprendizaje para verificar cuál muestra una mejor efectividad en la correcta clasificación del triaje y estudiar el funcionamiento de las redes bayesianas en el ambiente de urgencias. Este trabajo concluye que la efectividad mostrada por las redes bayesianas utilizando los diferentes algoritmos es de gran potencial en la simulación realizada con datos de prueba, obteniendo un promedio de ochenta y cuatro por ciento de precisión en la clasificación de triajes, lo cual

[9] Sistema de detección de intrusiones (*Intrusion Detection System* por sus siglas en inglés)

[10] Proceso que permite una gestión del riesgo clínico para manejar adecuadamente y con seguridad los flujos de pacientes cuando la demanda y las necesidades clínicas superan a los recursos. (W. Soler, M. Gómez Muñoz, E. Bragulat, A. Álvarez, 2010)

motiva a continuar la investigación en esta línea, utilizando datos reales y expandiendo el sistema a otras categorías sintomáticas relacionadas con enfermedades de distintos orígenes.

Por último, una investigación a nivel de posgrado en inteligencia artificial realizada por Silva F. (2011) profundiza en el funcionamiento de las redes bayesianas en la misma línea que contempla esta investigación, ya que busca elaborar un modelo de aprendizaje automático para la detección de fraudes electrónicos en transacciones financieras. Esta investigación diverge del presente trabajo de forma que el propósito de la misma fue la de establecer un modelo utilizando algoritmos de minería de datos, no específicamente de redes bayesianas. Sin embargo, el estudio demuestra técnicas y observaciones que pueden soportar de forma positiva el presente proyecto y su construcción. La investigación de Silva F., utiliza algoritmos de clasificación supervisada con modelos probabilísticos, tales como Naïve Bayes y Redes Bayesianas. Para poner a prueba el modelo a realizar, el estudio utiliza el sistema Weka como medio para realizar las simulaciones. Silva F. propone un modelo de análisis el cual este centrado únicamente en las transacciones de un comercio, es decir, una red diseñada para funcionar únicamente con un comercio en común. El estudio utilizó un total de 45,975 registros para realizar la simulación, dos tercios de estos fueron utilizados para el entrenamiento y el resto para la simulación del algoritmo en tiempo real. El estudio concluye con un resultado de 99.73% en la exactitud al momento de categorizar las transacciones como fraudulentas o íntegras, mostrándose así eficientes para descubrir conocimientos implícitos estudiando grandes cantidades de datos a partir de una muestra de estos, permitiendo inferir como una variable o atributo puede incidir en otros.

En el contexto nacional, de acuerdo a búsquedas realizadas en trabajos de graduación y estudios de la biblioteca de la Universidad de San Carlos de Guatemala, Universidad del Valle, entre otras, no se pueden referenciar investigaciones y trabajos relacionados con el proceso de descubrimiento, clasificación o aprendizaje basado en modelos estadísticos, específicamente de redes bayesianas.

Es por ello que los estudios anteriormente mencionados y otras publicaciones, pueden utilizarse como marco referencial a la presente investigación, sin representar un antecedente a la misma.

Entre los estudios anteriores se encuentra “Detecting credit card fraud by using support vector machines and neural networks”, por Rong-Chang Chen, Luo Shu-Ting y Li Shiue-Shiun. Este estudio realiza una comparación entre SVMs y redes neurales. El estudio llega a la conclusión de que ambos algoritmos ofrecen buenas soluciones para detectar fraude en tarjetas de crédito, pero que las SVMs tienen mejor desempeño que las redes neurales, cuando los datos son pequeños.

En el estudio Detecting Credit Card Fraud by Decision Trees and Support Vector Machines, por Y. Sahin y E. Duman, se realizó una comparación entre SVMs con distintos tipos de kernel y árboles de decisión, para el problema específico de fraude en tarjetas de crédito. Llegaron a la conclusión de que los

árboles de decisión tienen mejor desempeño que las SVM, pero que al incrementar la cantidad de datos de entrenamiento, la SVM alcanza el rendimiento de un árbol de decisión. Cabe mencionar que este estudio fue realizado con cantidades de datos relativamente bajas (menor a 10,000 datos).

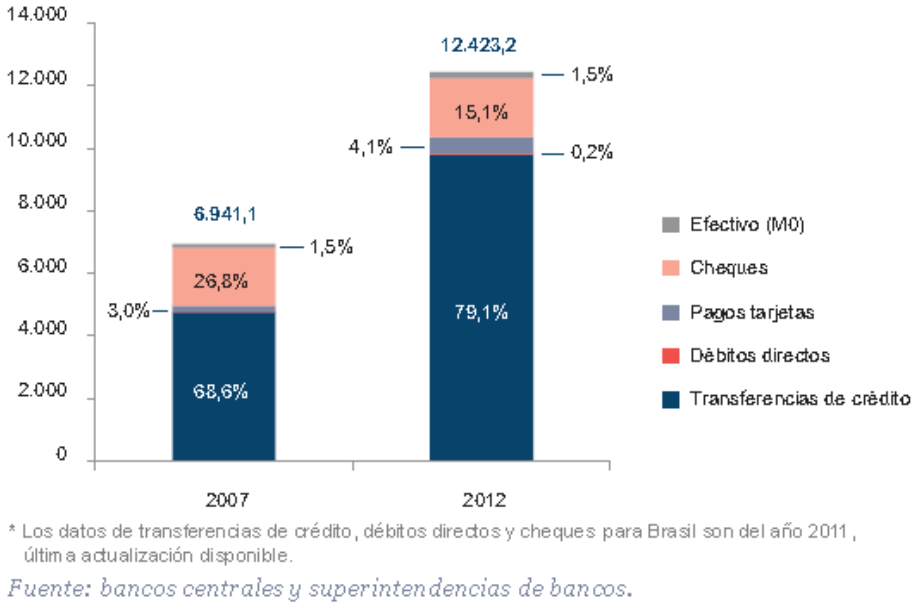
Otro estudio es el de Sitaram Patel y Sunita Gond, Supervised Machine (SVM) Learning for Credit Card Fraud Detection. En su artículo se realiza la comparación entre SMVs con distintos kernels, linear, cuadrático y RBF, utilizando Matlab para realizar el análisis. Llegan a la conclusión de que el mejor kernel es el RBF, para este problema

A. Tendencias actuales en los medio de pago en América Latina

1. Instrumentos de pago. En América Latina, el desarrollo de importantes operaciones corporativas regionales y la llegada de cambios en el entorno tecnológico han impulsado un proceso de sustitución gradual de los instrumentos de pago basados en papel (efectivo y cheques) por otros medios y canales, como Internet y el teléfono móvil. La innovación desarrollada por las empresas privadas avala el auge de estos canales, sobre todo en lo relativo a los pagos en comercio electrónico (que prácticamente se han duplicado en la región en los últimos años) y a los pagos móviles, donde se ha producido una transición desde los meros pilotos al lanzamiento de algunas iniciativas comerciales como Zuum en Brasil, o Transfer en Colombia. Al mismo tiempo, crece el interés por las soluciones de terminal punto de venta en dispositivos móviles (mPOS), como demuestran las apuestas inversoras de Banco Santander y BBVA en iZettle y SumUp, respectivamente. Aunque la presencia de este tipo de dispositivos en el mercado es todavía testimonial, su impacto en entornos de baja penetración de terminales, como los que se dan en algunos puntos de América Latina, será muy elevado. A ello contribuirán los proyectos normativos aprobados recientemente en América Latina, como los reglamentos de dinero electrónico y de tarjetas de pago en Perú, el reglamento de pagos móviles en Brasil, o el anuncio de la próxima reforma financiera en México.

Las transferencias de crédito protagonizan las transacciones realizadas en América Latina, acaparando un 79,1% del monto total de las operaciones de pago registradas en 2012 (USD 9,8 billones). Sin embargo, tal y como refleja la Figura 1, la importancia de los débitos directos (cargos autorizados de forma previa por el pagador) en la región es todavía muy reducida, acaparando un monto de operaciones que rebasaba ligeramente los USD 24,5 miles de millones en 2012. Su uso (atendiendo al valor de las operaciones) ha crecido en los últimos cinco años en países como Colombia o México, mientras que en otros como Brasil o República Dominicana permanece estancado (Valcárcel, J. Informe Tecnocom sobre tendencias en medio de pago 2013).

Figura 14. Valor de las operaciones de pago en América Latina en 2007 y 2012, por instrumento, miles de millones

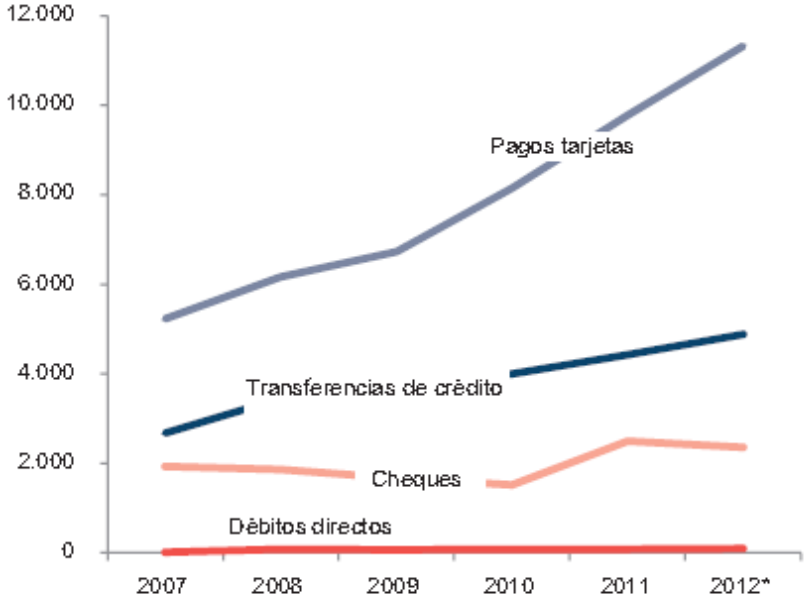


Fuente: Valcárcel, J. Informe Tecnocom sobre tendencias en medio de pago 2013.

Por su parte, las tarjetas de crédito y de débito han ganado importancia durante los últimos años, pasando a representar el 4,1% del monto total de las transacciones. Conocidas sus ventajas, son cada vez más los consumidores que recurren al dinero de plástico para instrumentar sus operaciones de compra, lideradas por las tarjetas que emiten los bancos y otras instituciones financieras bajo marcas globales establecidas por las organizaciones de tarjetas (principalmente Visa, Master-Card, American Express, Diners Club). A ellas se unen las tarjetas que emiten las casas comerciales, que en el caso de Chile ostentan una representatividad superior a las tarjetas emitidas por entidades financieras. De hecho, por número de operaciones, las tarjetas lideran el ranking de medios de pago, tal y como se puede observar en la Figura 2. Al examinar estos datos, se puede ver cómo la tarjeta es utilizada en más de la mitad de las transacciones, concretamente en un 60,7%.

A una distancia considerable se encuentra el cheque, resultado de su sustitución gradual por las transferencias electrónicas de crédito, que ocupan la segunda posición. Por su parte, el número de débitos directos en América Latina sigue siendo reducido en comparación con el resto de medios de pago, pero su evolución en el periodo considerado anota un incremento notable en cuanto a su participación sobre el total de operaciones: 0,5% en 2012 frente al 0,1% en 2007, poniendo de manifiesto su potencial (Valcárcel, J. Informe Tecnocom sobre tendencias en medio de pago 2013).

Figura 15. Número de las operaciones de pago en América Latina, 2007-2012, millones



* Los datos de transferencias de crédito, débitos directos y cheques para Brasil son del año 2011, última actualización disponible.

Fuente: bancos centrales u superintendencias de bancos.

Fuente: Valcárcel, J., Informe Tecnocom sobre tendencias en medio de pago, 2013.

Algunos países de América Latina también registran un ascenso significativo del uso fraudulento de plásticos (mayoritariamente clonaciones y delitos relacionados como el phishing6), lo que justifica que se esté exigiendo la migración a EMV para evitar el fraude con tarjetas de crédito, un proceso que comenzó en 2004 en dos de los países más grandes de la región: Brasil y México. Este proceso también afecta a las redes de cajeros automáticos, que progresivamente han de adaptarse para poder autenticar a través del chip las tarjetas, y así lo están promoviendo los gobiernos a través de diferentes regulaciones; la más reciente, el Proyecto de Reglamento de Tarjetas de Crédito y Débito del Perú. Con este tipo de medidas, los gobiernos buscan fomentar la utilización de instrumentos minoristas de pago electrónico y ampliar la infraestructura física de cajeros automáticos, que han registrado un notable ascenso durante los últimos años. La mayor parte del crecimiento de las operaciones se ha generado en las sucursales bancarias, aunque en los últimos años se ha podido apreciar un aumento en la demanda de las ubicaciones off-premise14, es decir, aquellas que se encuentran fuera de los bancos, en lugares como estaciones de servicio, supermercados, centros comerciales, farmacias o estaciones de metro, autobuses y trenes.

Las ubicaciones off-premise permiten atender a clientes no bancarizados, que en lugar de tener que acudir a las sucursales para formalizar pagos, lo pueden hacer a través de los ATM situados en diferentes puntos. Por ejemplo, el banco mexicano Bancomer, además de ofrecer la posibilidad de que sus clientes

paguen las cuentas de servicios básicos a través de cajeros automáticos, posibilita que éstos se utilicen para ofrecer créditos de consumo (Valcárcel, J. Informe Tecnocom sobre tendencias en medio de pago 2013).

2. Comercio electrónico. A ambos lados del Atlántico, el aumento del uso de dispositivos móviles para acceder a la red incide positivamente en el mayor protagonismo que está adquiriendo la banca por Internet y la banca móvil. La renovación del equipamiento móvil continúa dándose de manera generalizada en todos los países: mientras que se reducen considerablemente los teléfonos celulares convencionales (sin acceso a Internet, pantalla táctil, etc.), los smartphones siguen creciendo año tras año. Esta tendencia adquiere una relevancia especial en América Latina, gracias a su potencial de impactar a millones de personas en la región y permitir la inclusión financiera de segmentos sub-atendidos o no atendidos por los medios tradicionales. En este sentido, las previsiones de penetración móvil para 2017 apuntan que rebasará el 60% de la población latinoamericana, según la Asociación Mexicana de Internet (AMIPCI), lo cual permitirá que aquellos consumidores actualmente excluidos del sistema financiero puedan acceder al comercio móvil a través de sus dispositivos móviles. La importancia de Internet y la banca móvil impulsa la retroalimentación existente entre el sector de las telecomunicaciones y el negocio bancario, lo cual ofrece oportunidades de colaboración entre los operadores de ambos sectores. Este es el caso de las españolas CaixaBank, Santander y Telefónica, que en el segundo trimestre de 2013 acordaron la creación de la primera alianza entre banca y operadoras de telecomunicaciones en Europa, para desarrollar nuevos negocios digitales.

El comercio electrónico también experimenta un comportamiento positivo en las regiones analizadas, poniendo de manifiesto la importancia exponencial de otros medios distintos del efectivo. En América Latina, el volumen de ventas realizadas por este canal se ha duplicado en los últimos años, en línea con la mejora de los índices de inclusión financiera (índices de bancarización y de penetración de los medios de pago electrónicos) y la mayor seguridad del canal. De hecho, los bancarizados son los principales compradores online en todos los países y, en muchos casos, los no bancarizados apenas están incorporados a este canal. La principal excepción a esta regla es Brasil, donde 4 de cada 10 no bancarizados ha realizado alguna compra online gracias al boleto bancario, que permite realizar pagos seguros sin disponer de un vínculo formal con las entidades financieras.

Incluso se podría comenzar a hablar de un proceso de migración del comercio electrónico al móvil, ya que en 2012 el 15% de las ventas en Brasil y México se realizaron a través de canales móviles, según PayPal. Una tendencia corroborada por la plataforma Groupon, que canalizó entre el 5% y el 10% de sus ventas en América Latina a través de un medio móvil (Valcárcel, J. Informe Tecnocom sobre tendencias en medio de pago 2013).

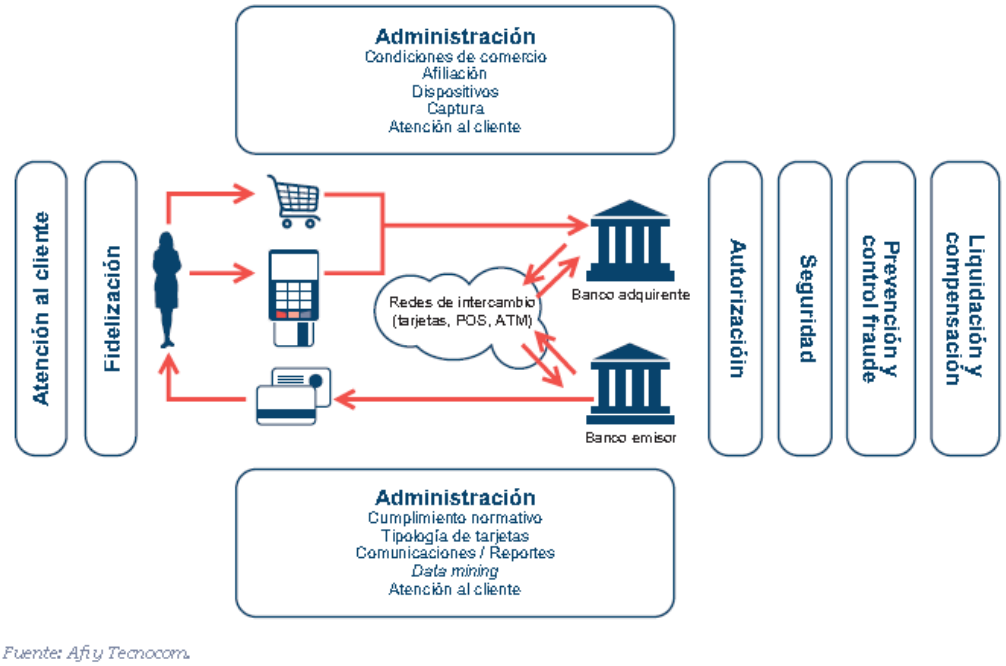
3. Tercerización de procesos. Los datos anteriores sitúan al consumidor en un entorno cambiante, principalmente influido por dos tendencias. Por un lado, los procesos de bancarización e inclusión financiera están atrayendo a segmentos de población con patrones de uso distintos a los de los clientes tradicionales (ticket promedio más bajo en operaciones a débito, mayor uso de la financiación a crédito) y que están expuestos a asimetrías de información mayores que los segmentos socioeconómicos altos. Por otro lado, el crecimiento del comercio electrónico y la telefonía móvil, que facilitan las operaciones de pequeña cuantía (micropagos) y tienden a la coexistencia de tarjetas físicas y virtuales. Ambas tendencias condicionan la reducción del coste de los servicios de procesamiento de pagos para favorecer los pagos de menor importe. En este sentido, la estrategia a seguir por los emisores puede ser diferente según sus condiciones particulares. Mientras que algunas entidades disponen de las capacidades y los recursos para mantener sus operaciones de procesamiento internalizadas, otros emisores (los de menor tamaño, por ejemplo) carecen generalmente de la escala necesaria para competir de manera eficiente, de modo que la subcontratación de los procesos de negocio - Business Process Outsourcing (BPO) - involucrados en el procesamiento de pagos puede ser una opción interesante. En la práctica, muchas entidades financieras optan por realizar sus actividades de forma mixta: desarrollando internamente parte de los procesos y subcontratando otros en función de sus necesidades.

Así, más allá de las funciones de autorización y compensación/liquidación, el procesamiento desde la óptica del emisor de tarjetas puede abarcar un amplio espectro de funciones de negocio, que van desde la emisión física de las tarjetas hasta la gestión del cobro.

Los procesos de mayor subcontratación suelen ser aquellos más alejados del core del negocio de las entidades, como la atención al cliente (contratando una empresa de servicios de call center), el marketing, la contabilidad, las finanzas, la administración de la tecnología y el cumplimiento normativo, especialmente en materia de seguridad de las transacciones y prevención del fraude.

Así, los proveedores de BPO han evolucionado desde los modelos iniciales de pura gestión transaccional, ocupándose de las funciones situadas en la parte superior de la Figura 3, hacia nuevas propuestas de valor añadido y personalización de oferta, mejorando su apuesta como alternativa a las soluciones internas. (Valcárcel, J. Informe Tecnocom sobre tendencias en medio de pago 2013).

Figura 16. Ámbitos de BPO en procesamiento de pagos



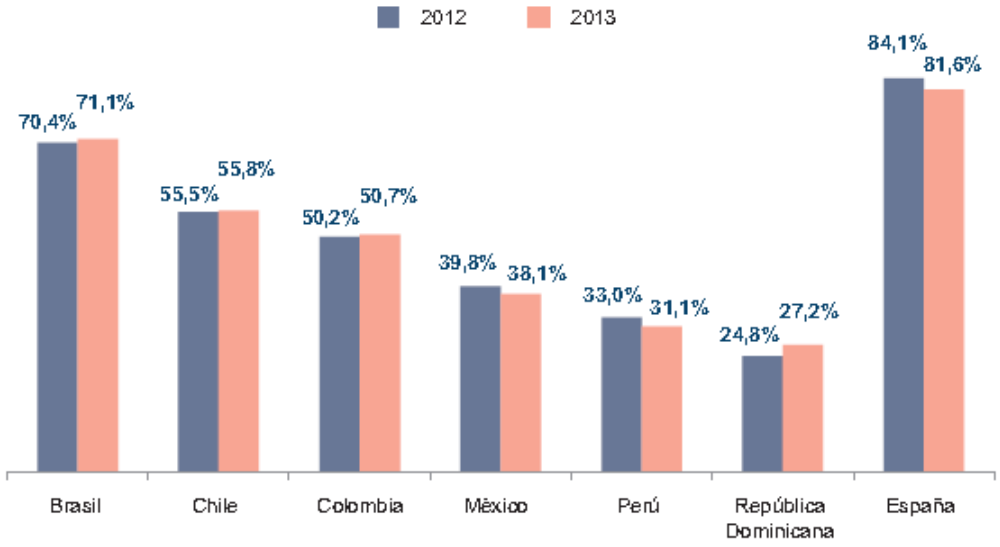
Fuente: Valcárcel, J. Informe Tecnocom sobre tendencias en medio de pago 2013.

En este contexto, observamos un panorama en el que la intensa concentración entre emisores ha dado como resultado algunas entidades de grandes dimensiones que han internalizado el grueso de las tareas del procesamiento de pagos electrónicos. Por su parte, la concentración en la industria del procesamiento ha producido el efecto contrario, permitiendo a los emisores más pequeños beneficiarse de las economías de escala que dicha concentración de proveedores les permite disfrutar. En todo caso, la decisión de externalizar o no el procesamiento es compleja y puede responder a múltiples criterios determinados por la estrategia de cada entidad (tamaño, búsqueda de especialización en determinados productos o segmentos, etc). Así encontramos entidades financieras emisoras de tarjetas de crédito con participación en alguna procesadora de pagos, que realizan buena parte de sus transacciones a través de ésta de forma exclusiva. Otras, en cambio, desarrollan y realizan internamente los servicios de procesamiento de pagos de sus clientes.

En definitiva, la evolución experimentada por la industria de las tarjetas plantea un entorno que tiende hacia mayores volúmenes transaccionales y menores importes promedio, márgenes más estrechos y mayor intensidad en el grado de competencia tecnológica (tanto en hardware como en software), lo que obliga a los servicios de procesamiento a intensificar su capacidad tecnológica y garantizar la seguridad de las transacciones (estándares PCI DSS).

En los últimos años se ha observado la mejora generalizada del número de titulares de tarjetas entre la población bancarizada de América Latina (Figura 4), pese a que no se logra llegar consistentemente a masas nuevas de población. Del conjunto de países destaca el comportamiento de Brasil, donde dos de cada tres personas dispone de algún plástico, lo que le convierte en el país de la región con más titulares de tarjetas. De hecho, se posiciona como el gran país de América Latina en su relación con la banca, seguido por Chile y Colombia, donde la mitad de la población dispone de este medio de pago. A cierta distancia podríamos encuadrar un tercer grupo de países compuesto por México, Perú y la República Dominicana, donde una mayoría de la población todavía no está bancarizada. Cabe destacar la evolución del país caribeño, que por primera vez contabiliza más de un cuarto de su población en posesión de tarjetas (Valcárcel, J. Informe Tecnocom sobre tendencias en medio de pago 2013).

Figura 17. Posesión de tarjetas de débito y/o crédito, por país

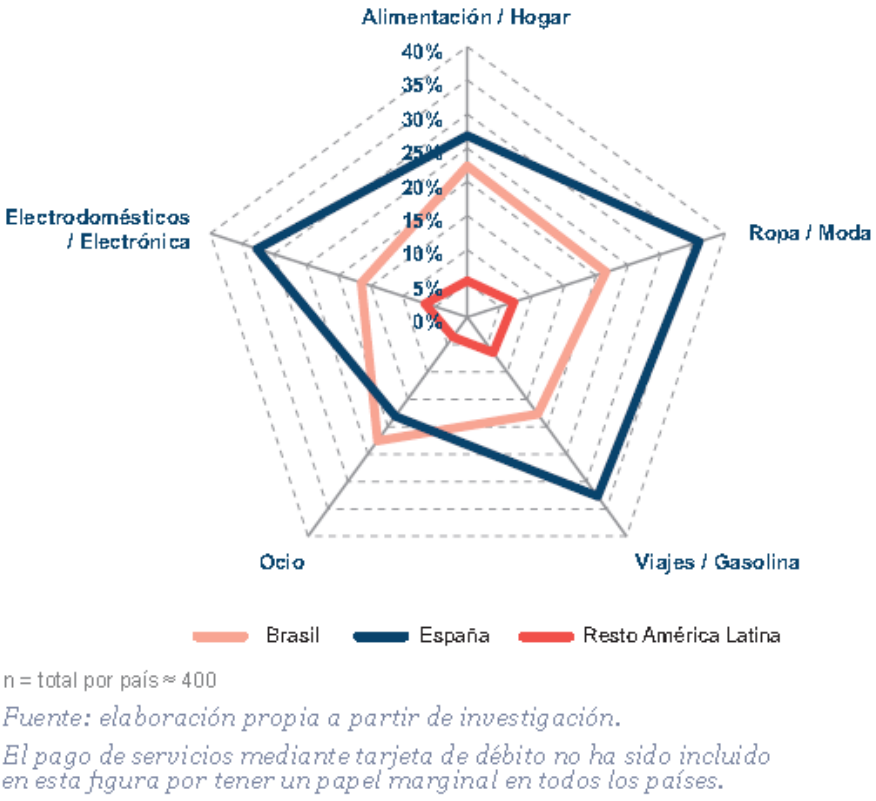


Fuente: Valcárcel, J. Informe Tecnocom sobre tendencias en medio de pago 2013.

4. Medio de pago por categorías. Por tipología, tanto el crédito como el débito anotan crecimiento en casi todos los casos, siendo esta última modalidad la que se está posicionando como el medio de pago más popular en comercios después del efectivo. En España, el retroceso registrado por el débito (pierde más de 4 puntos porcentuales frente al año anterior), probablemente como consecuencia de la integración de algunas entidades financieras, no impide que entre un 20% y un 30% de la población ya lo utilice como forma de pago más habitual (Figura 5).

Asimismo un 20% de brasileños lo utilizan de manera preferente en alimentación y hogar, ocio, y ropa o moda. El uso de este tipo de tarjetas crece en línea con su penetración, existiendo una clara correlación entre posesión y uso mensual (aproximadamente un 87,0% de quienes poseen una tarjeta de pago inmediato hace un uso mensual de ella). En cambio, el análisis de investigación apunta que a menor penetración, mayor es la influencia de la oferta de tarjetas de débito en la elección de la entidad financiera. Sin ser siempre un factor fundamental, se puede afirmar que tiene peso relevante en todos los países analizados. Destacan especialmente Perú y la República Dominicana (los países con menos débito) donde más de la mitad de los titulares de este tipo de tarjeta declaran que ha tenido algún o bastante peso a la hora de tomar la decisión sobre la entidad elegida (Valcárcel, J. Informe Tecnocom sobre tendencias en medio de pago 2013).

Figura 18. Tarjeta de débito como medio de pago más habitual por categorías



Fuente: Valcárcel, J. Informe Tecnocom sobre tendencias en medio de pago 2013.

Además, la perspectiva de contratación de tarjetas durante el próximo año en América Latina es positiva (Figura 6), principalmente en países con un mayor número de población optimista respecto a su economía. Así, por ejemplo, en República Dominicana una de cada cinco personas está considerando la contratación de un plástico, seguido de Perú (17,3%), Brasil (13,5%) y Colombia (13,3%). Incluso en países

con un optimismo más moderado, como Chile o México, aparecen porcentajes del 10% de la población considerando la contratación de medios de pago. Sólo entre los españoles resulta claro que su disposición a la contratación de medios de pago es residual (apenas un 3,1%). En general, el tipo de tarjeta a contratar en casi todos los países será la tarjeta de crédito bancario, con porcentajes que varían entre el 55% (México) y el 75% (Brasil).

Por su parte, la tarjeta de crédito de establecimiento presenta cierta dispersión del interés, ya que en el caso de Colombia y República Dominicana apenas tiene una demanda relevante. El débito alcanza un mayor interés en Colombia (27,8%) y México (21,0%), mientras que las tarjetas prepago no son apenas consideradas en ningún país (Valcárcel, J. Informe Tecnocom sobre tendencias en medio de pago 2013).

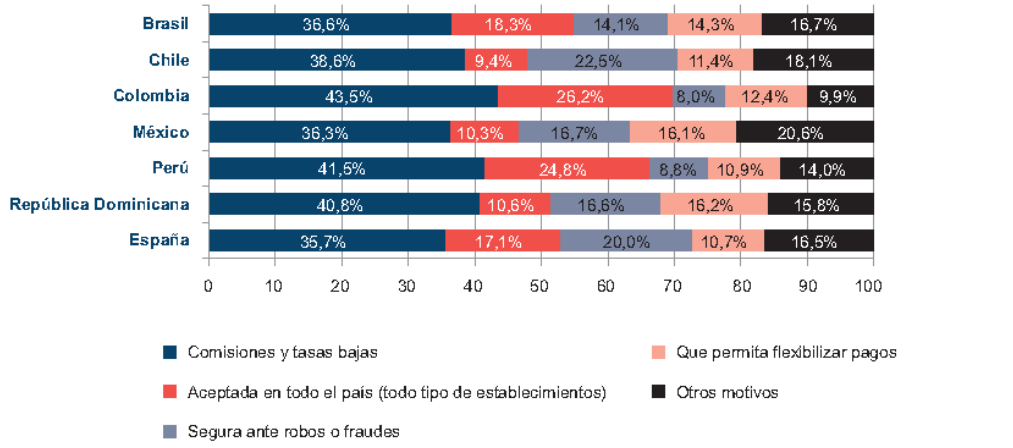
Figura 19. Población que está considerando contratar alguna tarjeta en el próximo año

Fuente: Valcárcel, J. Informe Tecnocom sobre tendencias en medio de pago 2013.

5. Criterios de decisión en la elección de una tarjeta de crédito. Los principales criterios de decisión utilizados por los clientes en la elección de una tarjeta de crédito. El primer lugar lo ocupa el precio sobre las comisiones y tasas que se cobran, ya que casi 4 de cada 10 encuestados refirieron este factor como el más determinante. A una distancia considerable encontramos el segundo factor de elección: la aceptación en los establecimientos del país; un factor que podría considerarse básico, pero que en países como Colombia resulta relevante. El tercer lugar lo ocupa la seguridad frente a robos y fraudes en la tarjeta, un factor discriminador especialmente considerado en Chile, donde el 22,5% de los titulares de tarjetas de crédito valora este aspecto por encima del resto. La flexibilidad de los pagos es un cuarto factor

de relevancia, más destacado en República Dominicana. Otros factores de menor peso serían la aceptación en el extranjero, o las ventajas adicionales que puedan ofrecer. En este sentido, las ventajas con mayor capacidad de influencia son: las promociones, los descuentos, el cash-back y los programas de millas. Respecto a los dos primeros, casi un tercio de los consultados lo considera su opción de mayor interés, especialmente en Colombia. En línea con este interés por los descuentos, el cashback (la devolución de parte de las compras realizadas con tarjeta) es la siguiente ventaja relevante, aunque con un peso menor, quizás debido a que el uso de esta tarjeta en las compras habituales es menor, y resulta más difícil evidenciar un beneficio tan importante como el anterior (Valcárcel, J. Informe Tecnocom sobre tendencias en medio de pago 2013).

Figura 20. Aspectos clave en la elección de una tarjeta de crédito



n = disponen de tarjeta de crédito
La categoría "Otros motivos" incluye varias opciones con resultados menores (en concreto "Aceptada en el extranjero" "Que sean las que ofrece su banco", "Que ofrezca ventajas adicionales" "Que sea de la cadena comercial donde realizo mis compras")

Fuente: elaboración propia a partir de investigación.

Fuente: Valcárcel, J., Informe Tecnocom sobre tendencias en medio de pago 2013.

VI. MARCO METODOLÓGICO

Los recursos principales que requirió el desarrollo de este megaproyecto es el recurso humano especializado en el área de detección de patrones de comportamiento y recurso tecnológico que permita aplicar las habilidades del recurso humano. El proyecto tuvo el apoyo de la empresa, quien es el cliente, y el apoyo de la Universidad del Valle de Guatemala, ya que un equipo de cinco estudiantes fueron los encargados de desarrollar este Megaproyecto. El equipo de trabajo estaba conformado por una estudiante de Ingeniería Industrial, encargada de la administración, gestión del proyecto y propuesta de modelo de negocio, un estudiante de Ingeniería en Ciencia de la Administración encargado del tema de Segmentación (Clustering), tres estudiantes de Ingeniería en Ciencia de la Computación y Tecnologías de la Información, los cuales fueron los encargados de desarrollar los algoritmos con diferentes técnicas de Inteligencia Artificial, para encontrar el algoritmo que pudiera obtener el mejor desempeño en el sistema de detección de fraude de la empresa y una asesora de parte del departamento de Ciencias de la Computación y Tecnologías de la Información.

El Megaproyecto estuvo dividido en 5 fases, las cuales se distribuyeron a lo largo del año y medio de duración del proyecto. Iniciando en julio de 2013 y terminando en noviembre de 2014.

A. Fase 1: Aprobación

Esta fue la fase inicial, la que marcó la pauta para establecer el Megaproyecto en la Universidad del Valle de Guatemala con el objetivo de sugerir posibles soluciones a la problemática actual del sistema de detección de fraude en tarjetas de crédito y débito, en el sistema vigente de la empresa. Esta fase concluyó con la creación del megaproyecto en la universidad, la formación del grupo de estudiantes que lo desarrolló y la entrega de un documento de parte de la empresa con la que se trabajó donde se expuso, en forma general, el objetivo del proyecto (ver en Anexo 1).

B. Fase 2: Definición

Una vez concluida la fase de aprobación, se dio inicio a esta fase, la cual tuvo como objetivo que el equipo de trabajo del megaproyecto conociera las características generales y específicas del sistema actual de detección de fraude en tarjetas de crédito y débito de la empresa para poder encontrar los posibles puntos de mejora del sistema. Así como también conocer el entorno interno y externo del cliente para proponer mejoras que se adaptaran a sus necesidades actuales y futuras.

En base a estos análisis e investigaciones se determinaron los roles que tendría que desempeñar cada integrante del equipo, para que cada quien pudiera enfocarse en recabar información para la solución dentro de su área de responsabilidad.

C. Fase 3: Planificación

Una vez definido el objetivo general de cada rol, la primera etapa fue de investigación para poder recabar información útil sobre los recursos y conocimientos que serían necesarios para desempeñar cada tarea. Se coordinaron reuniones internas con el equipo y reuniones con la empresa. Se construyó un Gantt para agendar las fechas de las reuniones y el entregable de cada una de estas.

Al final de esta fase, se logró definir claramente, en conformidad con la empresa, las técnicas que se aplicaron para solucionar el problema y la forma en que se gestionó el proyecto. Se establecieron tres algoritmos para desarrollar:

1. Redes Neurales
2. Redes Bayesianas
3. Support Vector Machines (SVM)

Se construyó un Gantt para definir las fechas estimadas de cada subfase de la ejecución. Se coordinaron los entregables con base en Balanced Scorecards (BSC).

D. Fase 4: Ejecución

En esta fase se desarrollaron los modelos y las pruebas respectivas con las técnicas de Inteligencia Artificial y Minería de Datos que se especificaron en la fase de planificación.

Para la realización de estas pruebas se utilizaron datos de prueba proporcionados por la empresa, y para lograr proponer una mejora, se tomaron como parámetros mínimos el desempeño del sistema actual de empresa.

Para asegurar el mejor resultado, con base en técnicas de mejora continua, se realizaron diversos prototipos antes de llegar a la versión final del producto. Al concluir cada prototipo se identificaron las fortalezas y debilidades de este y se realizó un plan de acción de mejora para el siguiente prototipo, hasta llegar a la versión final, la cual es una versión que suple de una manera más óptima a la actual las necesidades de la cadena de valor.

Durante esta fase se documentarán las acciones realizadas en versión del producto, los procesos realizados y los planes de acción para cada enfoque que se esté analizando como posible solución al problema. Se documentaron y justificaron también los cambios y decisiones tomadas en esta fase.

Se agendaron reuniones con la empresa con la frecuencia que se consideró necesaria con base en los resultados que se fueron obteniendo. Adicionalmente se les dio acceso a la plataforma virtual donde se planificó el proyecto: Asana, a fin que pudieran tener acceso a información actualizada del proyecto en el

momento que lo desearan. En Asana también se coordinaron las entregas y se llevaban control de las mismas en un formato tipo Balanced Scored Card, utilizando códigos de color para identificar visualmente el status del proyecto a todo momento. Se llevó control sobre las tareas atrasadas (rojo), en fase de desarrollo en tiempo (azul) y terminadas (verde).

E. Fase 5: Cierre

Esta fase dio inicio cuando se concluyó el prototipo final de todos los algoritmos y técnicas utilizadas y se realizó un análisis comparativo de todos los algoritmos con base en los parámetros de desempeño definidos con anterioridad y con esto se determinó cual era el algoritmo solución. En este caso "Redes Neurales" fue el algoritmo elegido como óptimo entre las tres opciones disponibles.

En esta fase se desarrolló también la propuesta de modelo de negocio con base en escenarios, utilizando como base los resultados y posibles escenarios que se pueden generar con el algoritmo elegido como óptimo, es decir con las "Redes Neurales".

Para concluir la fase, se ordenó, preparó y generó la documentación requerida para realizar los entregables finales tanto al cliente de la modalidad de megaproyecto como a la Universidad del Valle de Guatemala.

F. Metodología de algoritmos de inteligencia artificial

1. Diseño y análisis

a. Redes neurales. **C**ada red modular se trabaja con base en los campos seleccionados del modelo de transacción propuesto por Plus Technologies and Innovations, Inc., por lo que se define un módulo por conjuntos de la selección de campos. Junto a la estructura de la transacción fue proveído el conjunto de transacciones que es de un total de 5,723,270. El contexto de las transacciones las ubica como un conjunto de transacciones realizadas en compras en línea durante los meses de mayo a agosto del año 2012.

De estos campos se ha seleccionado un subconjunto en donde se eliminan datos que se encuentran contenidos o pre-procesados en otros, o que no proveen datos representativos para el análisis.

Según los campos seleccionados se procedió a generar una serie de escenarios respectivos de los cuales se obtendría cada uno de los módulos. Se identificó dos tipos de escenarios que podrían ser construidos: escenarios generales que generarían relaciones sin tomar en cuenta el comportamiento individual de un cliente y escenarios por cliente que generarían relaciones tomando en cuenta el identificador del cliente. Además se identificó que se podrían identificar situaciones cíclicas al agregar

campos como el día de la transacción, el mes de la transacción o la hora de la transacción. La definición de lo escenarios podría llegar a mejorarse a través del uso de una red neuronal sin supervisión de mapas auto-organizativos como lo menciona Schmidt, A. (1996) para la clasificación de entradas. Los módulos propuestos y los campos que se analizan por escenario son los siguientes, *i.e.*, los campos que funcionan como la entrada vectorial de cada red modular. Además se incluye la estructura propuesta para la red de decisiones:

- Módulo 1
  - Día de la transacción
  - Mes de la transacción
  - Hora de la transacción
- Módulo 2
  - Día de la transacción
  - Mes de la transacción
  - País del comercio
  - País de la transacción
- Módulo 3
  - Día de la transacción
  - Mes de la transacción
  - Marca de la tarjeta
- Módulo 4
  - Día de la transacción
  - Mes de la transacción
  - BIN
- Módulo 5
  - Día de la transacción
  - Mes de la transacción
  - Indicador de día de feriado
  - Semana de la transacción
- Módulo 6
  - Día de la transacción
  - Mes de la transacción
  - Tipo de producto de la tarjeta
- Módulo 7
  - Día de la transacción
  - Mes de la transacción
  - Merchant Category Code

- Módulo 8
  - Día de la transacción
  - Mes de la transacción
  - Monto de la transacción
  - Identificador del cliente
- Módulo 9
  - Día de la transacción
  - Mes de la transacción
  - País del comercio
  - Identificador del cliente
- Red de decisiones
  - Módulo 1
  - Módulo 2
  - Módulo 3
  - Módulo 4
  - Módulo 5
  - Módulo 6
  - Módulo 7
  - Módulo 8
  - Módulo 9

b. Redes Bayesianas

1) Estudio y análisis de datos. Con los datos proporcionados por Plus Technologies, se procedió a analizar la importancia que cada uno de ellos tiene dentro de la transacción. Posteriormente, se determinaron los campos que poseen un mayor impacto en la determinación de un fraude dentro de la transacción. Por último, se filtró la cantidad de campos a analizar debido al tipo de análisis que realiza las redes bayesianas, es decir, esta red debía iterar sobre cada uno de estos campos y sus posibles registros, por lo que no permitía tener una cantidad grande de nodos a analizar.

2) Construcción de diagrama de relaciones entre nodos. Una vez analizados los tipos de datos con los que se cuentan, se procedió a generar un diagrama de relaciones, cuyo propósito es darle una representación gráfica a la red bayesiana a generar. Esto consiste en representar nodos de entrada y de salida, que estaban relacionados directamente a los campos que se utilizarían de los datos proporcionados. Sin embargo, antes de la definición de los

nodos finales, fue necesario realizar un procesamiento previo para que los datos pudiesen ser de mayor utilidad.

a) Tratamiento de datos. Inicialmente, los datos proporcionados poseían campos que serían innecesarios para llevar a cabo este proyecto. Por ello, se realizó una lista de 14 campos que se utilizarían para el objetivo de este proyecto. Junto a esto, se determinó que se utilizaría únicamente 65% de los datos (3.7 millones de transacciones) para entrenamiento y el otro 35% sería reservado únicamente para pruebas.

Una vez definidos los campos, se procedió a realizar diferentes consultas en la base de datos MSSQL Server proporcionada y de esta forma filtrar únicamente los campos que se necesitarían. Una vez filtrados los campos que se utilizarían, se procedió a exportar los datos a un archivo CSV para que posteriormente pudieran ser importados en el motor de base de datos MongoDB que se utilizaría para el análisis de las transacciones. Al verificar los campos finales en MongoDB, se determinó que el identificador de comercio se presentaba como una cadena de caracteres (*string*) y esto complicaría las iteraciones a través de los campos, por lo que se optó a sustituir cada uno de los campos de identificador de comercio por números únicos. Este proceso de sustitución llevo alrededor de 7 días, debido a que los comercios existentes eran aproximadamente 157 mil y por cada uno de éstos, se debía realizar una consulta a la base de datos que tomaba aproximadamente 4 segundos para reemplazar todas las posibles ocurrencias.

Al realizar los diferentes cálculos para determinar la cantidad de iteraciones que necesitaba cada nodo para entrenar la red bayesiana, se encontró que ésta requeriría de un poder de computación que escapa del recurso y *hardware* a disposición para realizar la investigación, ya que un nodo promedio requería aproximadamente 138 trillones de iteraciones para determinar todas las probabilidades de ocurrencia existentes. Al extrapolar estos datos y calcular una posible variable de tiempo, se obtuvo que si cada iteración toma un promedio de 2 segundos (basado en pruebas de iteraciones realizadas previamente), el análisis completo del nodo tomaría un total de 9 millones de años en completar, lo cual era virtualmente imposible. Es por ello que se decidieron realizar múltiples ajustes a los datos por analizar, entre estos ajustes se encontraron:

- Los nodos iterarían únicamente sobre datos existentes dentro de los proporcionados, es decir, no se realizarían permutaciones a fuerza bruta tomando todos los posibles valores que un campo pueda adoptar. Esto procura que no se realicen consultas al motor de base de datos, que devuelvan resultados vacíos y que aumentan considerablemente el tiempo de computación del nodo a nivel general.

- Se redujo la muestra a analizar de 5.7 millones de registros a 200 mil. Esto fue necesario, ya que se realizaron pruebas de rendimiento, las cuales se iban disminuyendo en cantidad de

registros utilizados hasta alcanzar un balance en el tiempo requerido para la computación de los resultados, como en la cantidad de registros analizados, para que proporcionen una base de aprendizaje aceptable para la red bayesiana. Esto nos permite mantener la uniformidad de la red bayesiana con una base de aprendizaje sólida, mientras se acorta significativamente la cantidad de datos necesarios.

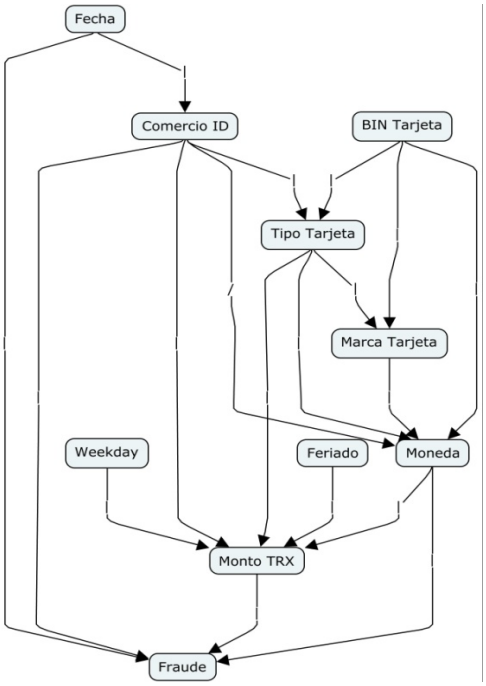
- El campo de identificador de comercio, se agrupó utilizando como referencia el valor esperado de consumo que cada comercio obtuviera. Esto permitió disminuir la cantidad de identificadores únicos de 157 mil valores aproximadamente a 10. A pesar que la introducción de esta medida significaría una pérdida de información, se determinó que beneficiaría el estudio debido a que a lo largo del mismo se definió que mientras la cantidad de valores únicos que tomaría un campo se disminuyera, existiría una mejor distribución de los datos analizados en su probabilidad de ocurrencia.

Una vez finalizada la fase de tratamiento de datos, se concluyó que el análisis sería completado de forma satisfactoria, utilizando 6 nodos dentro de la red bayesiana a construir. La cantidad de nodos fue reducida, con el fin de minimizar la cantidad de iteraciones a realizar para el análisis respectivo. Estos nodos serían:

1. Identificador de comercio
2. Fraude
3. Marca de la tarjeta
4. Moneda de la transacción
5. Monto de la transacción
6. Tipo de Tarjeta

Utilizando estos nodos, se determinó un diagrama relacional, en los cuales se establecían dependencias entre nodos y determinaba el camino que se debía tomar al momento de entrenar la red bayesiana por medio de las iteraciones sobre los datos. En la siguiente figura se puede observar el diagrama final que se utilizaría para modelar la red bayesiana.

Figura 21. Diagrama de relaciones entre nodos



Fuente: Elaboración propia

c. Support Vector Machines. En primer lugar se delimitó como sería la SVM a implementar para que posteriormente determinar con qué herramientas se implementaría la misma.

Debido a que las SVMs son algoritmos de clasificación, se identificaron las clases del problema a resolver. Las clases identificadas fueron dos, la clase de transacciones fraudulentas y la de transacciones no fraudulentas. Se seleccionaron estas clases ya que el objetivo es identificar que transacciones son fraudulentas y cuáles no.

Por otro lado la SVM funciona únicamente con datos numéricos, por lo que hizo ver la necesidad de mapear los campos de texto a campos numéricos.

Adicionalmente, ya que se trata de un sistema con una alta cantidad de transacciones, es necesaria que la SVM sea entrenada por lotes, para asegurar que pueda manejar grandes cantidades de datos en un ambiente real.

3) Leguaje de programación a utilizar. Se tomó Python como el lenguaje a utilizar ya que es un lenguaje que brinda la flexibilidad y facilidad de uso de un lenguaje de alto nivel sin

perder eficiencia. Adicionalmente es liviano, por lo que se puede instalar en distintos dispositivos y tiene la capacidad de correr librerías en lenguajes de menor nivel, como C++.

4) Librería a utilizar. Para Python resaltan dos librerías para SVMs que tienen amplio soporte y se encuentran en continuo desarrollo. La primera es LibSVM, cuya base se encuentra en C++ por lo que se asegura su eficiencia, y la segunda es scikit-learn que contiene muchas más aplicaciones de aprendizaje de máquinas. Se decidió empezar utilizando LibSVM para determinar su eficiencia y posteriormente se cambió por scikit-learn ya que funcionó con mayor eficiencia y se utilizaron otras de sus aplicaciones de aprendizaje de máquinas.

2. Implementación

a. Redes neurales. Para llevar a cabo el entrenamiento de los módulos y por último la red de decisiones se utilizo el lenguaje de programación python con la librería Pybrain. Los datos se encontraban en una instancia de servidor de MongoDB para fácil acceso y manipulación, y se obtenían a través de una biblioteca de código en python. El proceso de entrenamiento y validación se realizó por cada red modular y luego por agregación con la red de decisiones. La fase de entrenamiento y validación se realizaron por separado al obtener las muestras en conjuntos como sugiere el algoritmo de validación cruzada con k-iteraciones.

Las muestras de entrenamiento se consolidaban en un sólo conjunto y se realizaba el entrenamiento por un total máximo de 100 épocas; para poder evaluar los mejores parámetros para la estructura de las redes neuronales se realizaron 116 ejecuciones con variaciones de ciertos parámetros y se comparó en base al porcentaje de validaciones que el modelo realizaba correctamente. Al terminar la fase de entrenamiento se consolidaron las muestras de validación en un sólo conjunto y se compararon con los resultados de los modelos generados. En la validación se obtuvo la distribución de los resultados del modelo, para los casos en donde la transacción era fraudulenta, y se definió un rango en el que se separaban el tipo de transacción obtenida.

1) Metodología de integración continua. Para asegurar que el proceso de desarrollo, y el constante cambio en los módulos y lógica del proyecto, no fuera afectado se utilizó una serie de tecnologías de forma paralela. Inicialmente, se creo un repositorio en el sitio web github.com en el cual se llevo a cabo el control de versiones de todo el proyecto, incluyendo la fase de pre-procesamiento. El repositorio se creó de tipo público para poder realizar la integración con Travis CI. Esta segunda herramienta fue propuesta para llevar a cabo la ejecución de las pruebas unitarias en cada actualización de código, y la implementación en el sitio Heroku.com.

b. Redes Bayesianas. La construcción de esta red bayesiana, se hizo complementado con un sistema de semáforo. Esta red toma cada transacción que se quiere analizar y la procesa por cada una de las reglas existentes para cada nodo. Si no existe una regla que aplicase, simplemente la probabilidad de ocurrencia es 0%.

Por cada nodo que se procesaba, se obtuvo la probabilidad de ocurrencia de esta transacción. Dependiendo del nodo en el que se encontraba, se le asignó dos valores importantes: valor del semáforo y ponderación del nodo. El valor del semáforo es un entero entre 1 y 3 que determina qué tan probable es la ocurrencia de dicha transacción en términos del nodo que la está analizando, el valor 1 representa que su probabilidad de ocurrencia es alta, el valor 2 que es media y el valor 3 que es baja. Por otro lado, la ponderación del nodo, es un número (0.3, 0.5, 0.7 y 1.0) que determina la importancia del nodo para la red bayesiana, es decir, qué tan relevante es el nodo tratado para determinar si la transacción es fraudulenta o no.

Por último, el programa recolecta cada uno de estos valores para cada uno de los nodos involucrados, los opera y los suma. La multiplicación se utiliza como posible filtro para que los nodos con menor ponderación no tuviesen tanta incidencia como los nodos de mayor ponderación en la suma final. Al obtener la suma final, se divide este resultado entre la cantidad de nodos evaluados (6) lo que devuelve un número entre 0 y 3 al cual se le aplica el mismo criterio del semáforo, con la diferencia que si el número es menor o igual a 1, la transacción no es fraudulenta; si el número es menor o igual a 2, pero mayor a 1, la transacción no es fraudulenta, pero se emite una alerta; y por último, si el número es mayor a 2, se categoriza como transacción fraudulenta.

Este algoritmo, guarda una historial de lo que se realiza y del resultado de cada transacción para verificar que decisión se tomó en cada transacción y en cada nodo de la misma.

c. Support Vector Machines. La experimentación se dividió en fases para poder mejorar continuamente la SVM y obtener resultados de forma continua para determinar el estado de la misma. En cada fase se utiliza una librería para crear la SVM y se realizan pruebas para determinar cómo mejorarla. En todas las fases se utilizó Python 2.7. Cada nueva fase se inició cuando se detectó un problema en la SVM. Se consideró como un problema la alta detección de falsos positivos o negativos (mayor al 20%), si el tiempo de entrenamiento o detección de la SVM era prolongado (mayor a 1 hora) y bajos porcentajes de aciertos (menor a 80%).

En cada fase se realizan pruebas utilizando un porcentaje de los datos. En estas pruebas se calculó para cada SVM su porcentaje de aciertos, porcentaje de falsos positivos y porcentaje de falsos negativos.

3. Pruebas y comparaciones

a. Redes neurales. Entre los objetivos del desarrollo de la red de decisiones se encontraba la creación de un entorno en el ual se pueda realizar predicciones a través de un protocolo HTTP utilizando la red neuronal resultante de las etapas de entrenamiento y validación. Para poder cumplir con este objetivo se utilizó la biblioteca de código en python llamada Tornado, la cual es un servidor web asíncrono que trabaja con cargas de muchas transacciones. El servidor web se programó de manera que se pudiera utilizar como un webservice Restful, y se proveían dos puntos de acceso, uno asíncrono y otro síncrono, para evaluar su diferencia. En las pruebas de desempeño se utilizó la herramienta httperf para generar la cantidad de transacciones necesarias para probar el límite del servidor web. La configuración utilizada para httperf fue de 1000 conexiones abiertas por época, 35 época en una prueba y el aumento de 500 conexiones por cada época que avanzaba. De los datos obtenidos se graficó los resultados de la cantidad de peticiones que se pueden realizar por segundo, la cantidad de peticiones que se pueden realizar por milisegundo y el total de peticiones realizadas que terminaron con un código de estatus exitoso, en el protocolo HTTP este es 200.

b. Redes Bayesianas. Una vez desarrollada la red bayesiana, se dispuso a ejecutar las 200 mil transacciones reservadas anteriormente y determinar la efectividad de la misma. Esto se logró por medio de un algoritmo que itera sobre la base de datos con la información a probar, para luego procesarla por los diferentes nodos elaborados anteriormente. Este algoritmo da como resultado un conjunto de archivos de texto con información acerca de los aciertos, falsos positivos y falsos negativos que se obtienen en el análisis. Con esta información se puede proceder al análisis de resultados. .

c. Support Vector Machines.

Fase 1. En la primera fase se utilizó la librería LibSVM para implementar una SVM. En esta fase no se utilizó pre-procesamiento para los datos, ya que sólo se necesitaba crear una SVM funcional capaz de solucionar el problema. El entrenamiento de la SVM se realizó con el 77% de las transacciones que nos propició Plus TI y las pruebas se realizaron con el restante 23%.

Luego de las pruebas se determinó que la SVM no estaba detectando transacciones fraudulentas. Sin embargo muchos de los datos tenían campos vacíos, por lo que se atribuyó a esta razón la falta de detección de transacciones fraudulentas. Adicionalmente el tiempo en el que se entrenaba esta SVM era prolongado, por lo que se utilizó otra librería en la siguiente fase.

Fase 2. En esta fase se cambió la librería utilizada por, la que fue utilizada en el resto de las fases, scikit-learn. Se utilizó normalización como método de pre-procesamiento para mejorar

los resultados y agilizar el entrenamiento. El entrenamiento de la SVM se realizó con el 77% de las transacciones que nos propició Plus TI y las pruebas se realizaron con el restante 23%.

De nuevo, luego de las pruebas se determinó que la SVM no estaba detectando transacciones fraudulentas. Sin embargo los datos completos aún no habían sido recibidos por lo que de nuevo se atribuyó la falta de detección a la falta de campos de las transacciones. Al utilizar esta librería se consiguió una mejora considerable de tiempo, por lo que se utilizó durante el resto del proyecto.

Fase 3. En esta fase se recibieron los datos de prueba completos. Debido a la cantidad de datos, 5.7 millones, y para adecuar el algoritmo a los requisitos necesarios, se implementó una SVM capaz de ser entrenada por lotes, esta SVM se entrena mediante el método gradiente descendente estocástico. De nuevo se utilizó el método de normalización para pre-procesar los datos. El entrenamiento se realizó con el 70% de las transacciones y las pruebas con el 30% restante.

Por tercera vez, se determinó, luego de las pruebas, que la SVM no detectaba transacciones fraudulentas. Al tener los datos con los campos completos se determinó que se debía mejorar el pre-procesamiento para que la SVM pudiera ser entrenada de mejor forma y fuera capaz de detectar distintos tipos de transacciones.

Fase 4. En esta fase se cambió el método de normalización utilizado (de normalización por unidad de longitud a normalización estándar). Adicionalmente se creó un diccionario para cada uno de los campos de texto de las transacciones, para poder normalizarlos de igual manera que al resto de datos y agilizar el proceso de normalización. Se utilizó el 70% de los datos para entrenar la SVM y el 30% para las consecuentes pruebas.

Se realizó una nueva fase ya que el porcentaje de falsos negativos fue de 50%. Se determinó que el alto porcentaje de falsos negativos se debió a que menos de 1% de las transacciones eran transacciones fraudulentas.

Fase 5. En esta fase se realizaron pruebas de tres tipos, para determinar la configuración de los datos en la que los porcentajes de falsos positivos y falsos negativos fueran menores. En todas las pruebas se utilizó como pre-procesamiento normalización estándar y un diccionario para cada uno de los campos de texto. Se utilizó el 70% de los datos para entrenar la SVM y el 30% para las pruebas. Luego de realizar estas pruebas se seleccionó la SVM, que tuviera menor porcentaje de falsos positivos y negativos, como la SVM final.

a) Pruebas tipo 1. El primer tipo de pruebas consistió en tomar todos los datos de entrenamiento y cambiar la cantidad de transacciones no fraudulentas utilizadas. Estas pruebas se realizaron con el objetivo de darle más importancia, en la SVM, a las transacciones no fraudulentas y de esta manera disminuir el porcentaje de falsos positivos.

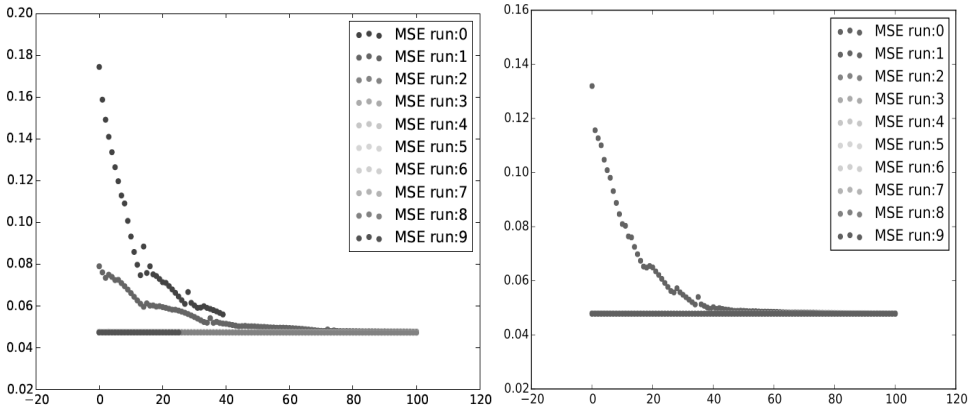
b) Pruebas Tipo 2. El segundo tipo de pruebas consistió en tomar el 70% de las transacciones fraudulentas y tomar cierta cantidad de transacciones no fraudulentas, aleatoriamente. Estas pruebas se realizaron con el objetivo de disminuir la cantidad de falsos positivos al tomar transacciones de todos los meses disponibles en los datos.

c) Pruebas tipo 3. En este tipo de pruebas se tomó el 70% de las transacciones fraudulentas y se utilizaron para entrenar la SVM ocho veces. Es decir por cada 500,548 transacciones no fraudulentas se volvieron a ingresar todas las transacciones fraudulentas. Adicionalmente se cambiaron los pesos de las clases. Estas pruebas se realizaron para aumentar la importancia de las transacciones fraudulentas y disminuir el porcentaje de falsos positivos.

VII. RESULTADOS

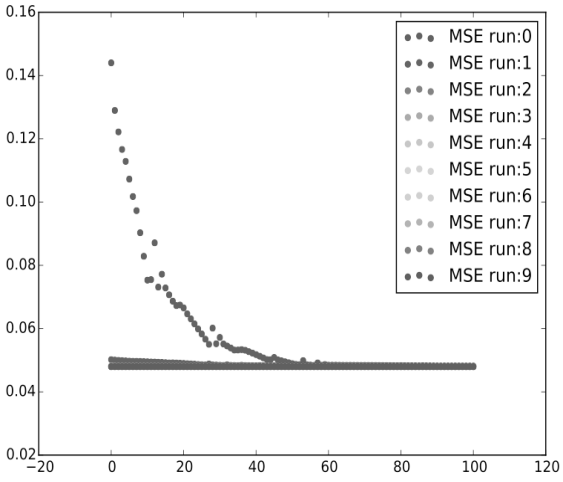
A. Resultados de redes neuronales

Figura 22. Valores de error del algoritmo de entrenamiento de un módulo de red neuronal graficados versus un total de 200 épocas de ejecución. (A) Error calculado en Escenario 8 (B) Error calculado en Escenario 6



(A) (B)

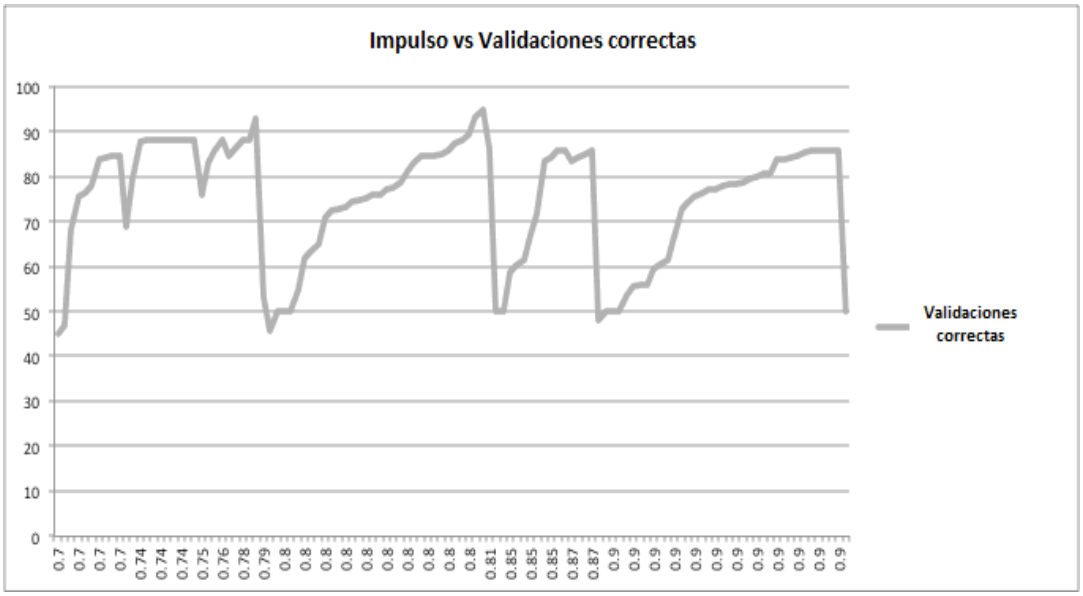
Figura 23. Impulso del algoritmo de propagación hacia atrás vs el porcentaje de validaciones correctas en 116 ejecuciones.



(C)

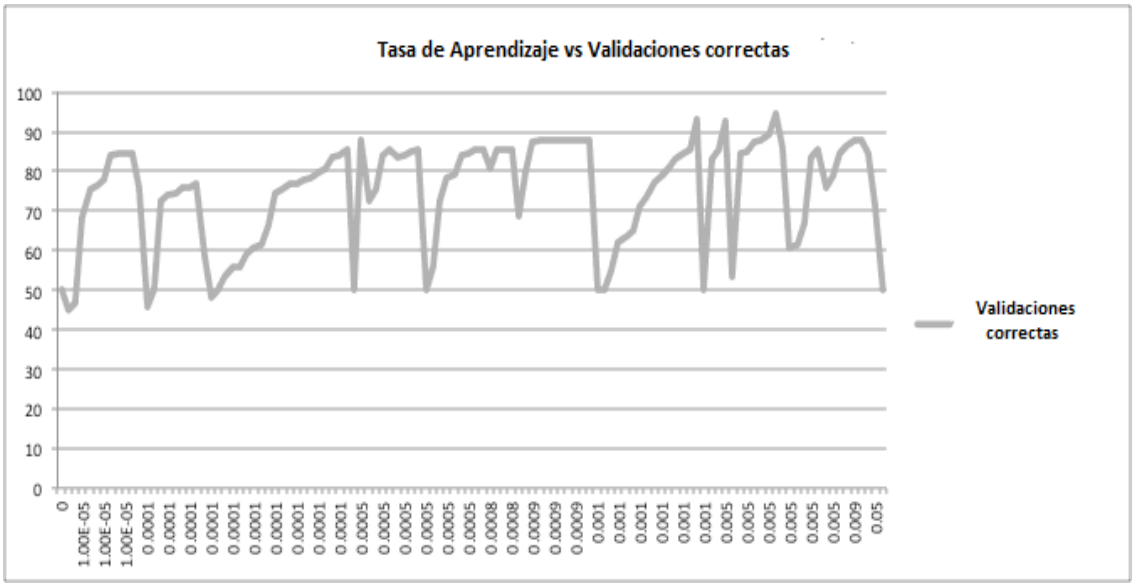
Fuente: Elaboración propia

Figura 24. Tasa de aprendizaje del algoritmo de propagación hacia atrás vs. el porcentaje de validaciones correctas en 116 ejecuciones.



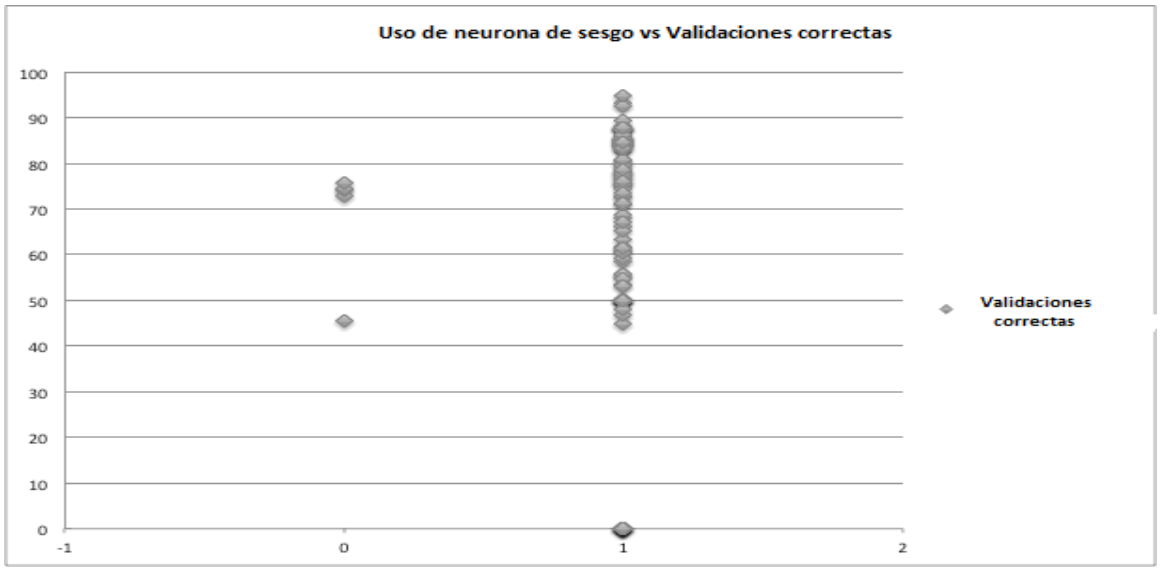
Fuente: Elaboración propia

Figura 25. Uso de neuronas de sesgo en la estructura de la red neuronal vs. el porcentaje de validaciones correctas en 116 ejecuciones. (0 Representa una estructura sin uso de neuronas de sesgo y una estructura que utiliza neuronas de sesgo).



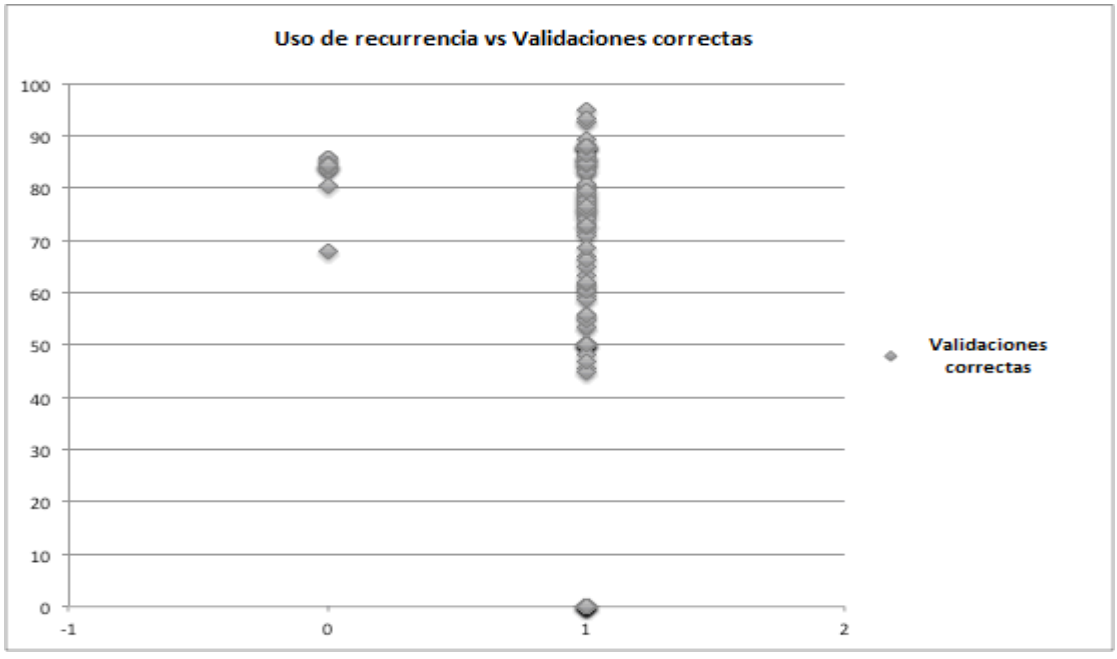
Fuente: Elaboración propia

Figura 26. Uso de Recurrencia a sí misma en neuronas de la estructura de la red neuronal vs el porcentaje de validaciones correctas en 116 ejecuciones. (0 Representa una estructura que no utiliza recurrencia y 1 una estructura que utiliza recurrencia).



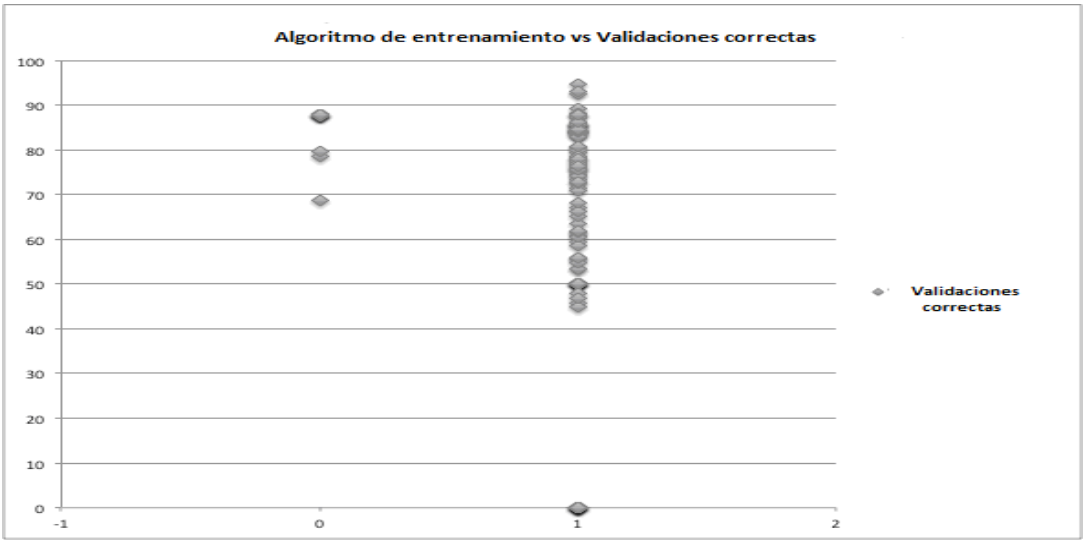
Fuente: Elaboración propia

Figura 27. Algoritmo de entrenamiento de la red neuronal vs el porcentaje de validaciones correctas en 116 ejecuciones. (0 Representa neuronas que utilizan el algoritmo de propagación hacia atrás y 1 neuronas que utilizan el algoritmo de propagación hacia atrás



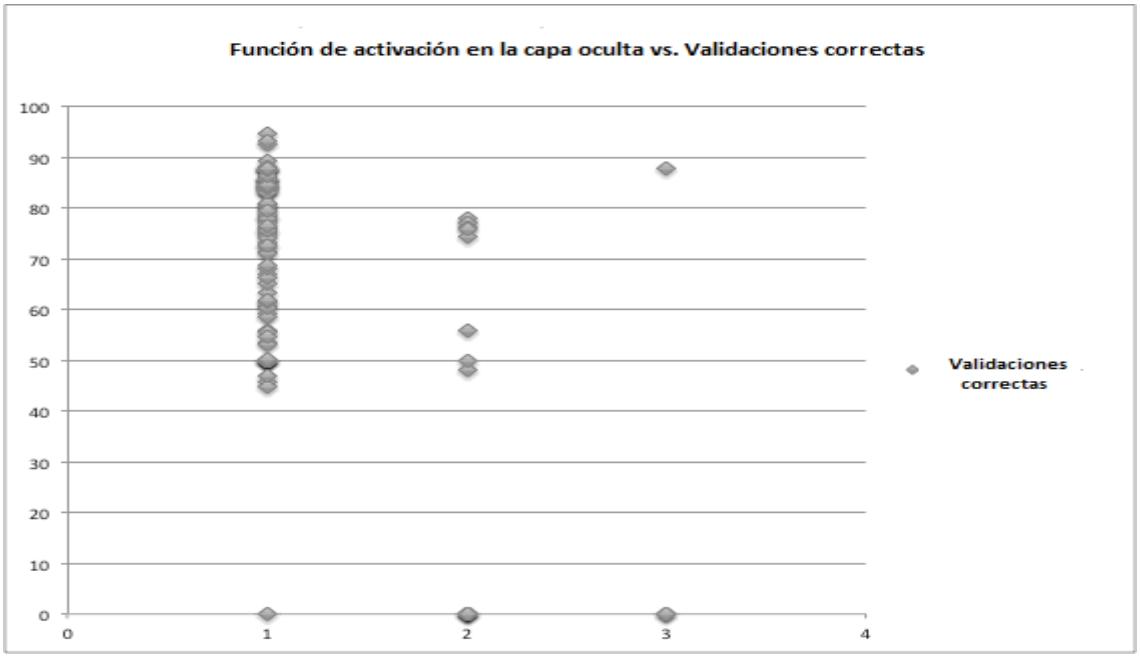
Fuente: Elaboración propia

Figura 28. Función de activación de las neuronas en la capa oculta de la estructura de la red neuronal vs el porcentaje de validaciones correctas en 116 ejecuciones. (0 Representa la función de activación Sigmoidal, 1 representa la función de activación Softmax



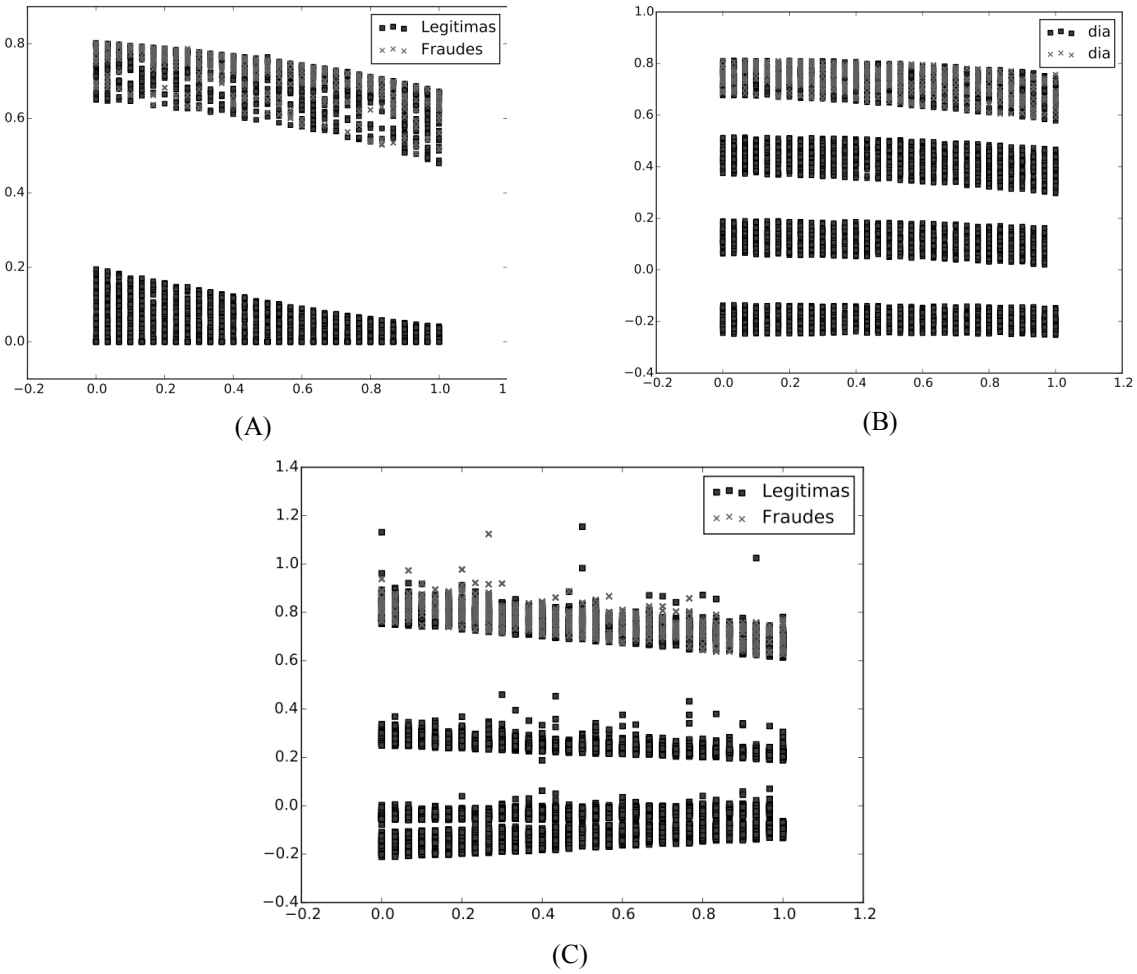
Fuente: Elaboración propia

Figura 29. Comparación entre distintas funciones de activación para la capa de salida de la estructura de la red neuronal; utilizando 100 épocas de entrenamiento. (A) Función Sigmoidal (B) Función Softmax (C) Función Tangente Hiperbólica.



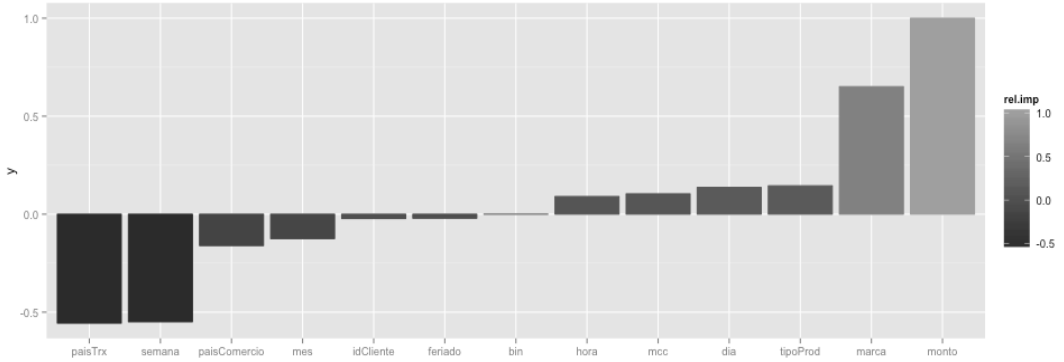
Fuente: Elaboración propia

Figura 30. Análisis de importancia de variables utilizadas en los escenarios para entrenar las redes modulares



Fuente: Elaboración propia

Figura 31. Análisis de importancia de variables donde los nombres de los escenarios se codifican de acuerdo a la variable más significativa del módulo.



Fuente: Elaboración propia

Cuadro 1. Asociación de variables a módulos en base al análisis de importancia de variables

Módulo	Variable
1	Hora
2	PaisTrx
3	Marca
4	*BIN*
5	Semana
6	TipoProd
7	MCC
8	Monto
9	Pais comercio

Fuente: Elaboración propia

1. Resultados del entrenamiento. Se llevó a cabo una serie de ejecuciones, por cada uno de los escenarios propuestos como módulos conectados a una red de decisiones y los porcentajes obtenidos. Este proceso se realizó una vez con el algoritmo de propagación de errores hacia atrás y otras con la versión elástica del mismo. Los resultados se resumen en el siguiente cuadro.

Cuadro 2. Resultados de entrenamiento y validación de transacciones utilizando dos algoritmos de entrenamiento distintos por cada red modular

Módulo	Algoritmo Propagación hacia atrás (Backpropagation)	Algoritmo Propagación hacia atrás elástica (Resilient Backpropagation)

Fuente: Elaboración propia

Cuadro 3. Módulo 1

Resultado	Backpropagation	Resilient Backpropagation
Porcentaje de transacciones identificadas exitosamente	85.69596328 %	87.8851247735 %
Porcentaje de transacciones identificadas incorrectamente	14.30403672 %	12.1148752265 %
Proporción de identificación	1.666666667:10	1.25:10
Porcentaje de falsos positivos	0 %	0.02348643 %
Porcentaje de falsos negativos	3.737213404 %	4.512004175 %

Fuente: Elaboración propia

Cuadro 4. Módulo 2

Resultado	Backpropagation	Resilient Backpropagation
Porcentaje de transacciones identificadas exitosamente	85.69596328 %	87.9618011989%
Porcentaje de transacciones identificadas incorrectamente	14.30403672 %	12.0381988011%
Proporción de identificación	1.666666667:10	1.25:10
Porcentaje de falsos positivos	0 %	0.02348643%
Porcentaje de falsos negativos	3.737213404 %	4.483298539%

Fuente: Elaboración propia.

Cuadro 5. Módulo 3

Resultado	Backpropagation	Resilient Backpropagation
Porcentaje de transacciones identificadas exitosamente	84.39314162 %	87.93391886 %
Porcentaje de transacciones identificadas incorrectamente	15.60685838 %	12.06608114 %
Proporción de identificación	1.666666667:10	1.25:10
Porcentaje de falsos positivos	2.888888889 %	0.02348643 %
Porcentaje de falsos negativos	1.188712522 %	4.493736952 %

Fuente: Elaboración propia

Cuadro 6. Módulo 4

Resultado	Backpropagation	Resilient Backpropagation
Porcentaje de transacciones identificadas exitosamente	65.06682868 %	87.98271295 %
Porcentaje de transacciones identificadas incorrectamente	34.93317132 %	12.01728705 %
Proporción de identificación	5:10	1.25:10
Porcentaje de falsos positivos	9.095238095 %	0.04697286 %
Porcentaje de falsos negativos	0.031746032 %	4.451983299 %

Fuente: Elaboración propia

Cuadro 7. Módulo 5

Resultado	Backpropagation	Resilient Backpropagation
Porcentaje de transacciones identificadas exitosamente	61.86040232 %	87.04168409 %
Porcentaje de transacciones identificadas incorrectamente	38.13959768 %	12.95831591 %

Continuación Cuadro 7. Módulo 5

Resultado	Backpropagation	Resilient Backpropagation
Proporción de identificación	5:10	1.428571429:10
Porcentaje de falsos positivos	8.746031746 %	0.493215031 %
Porcentaje de falsos negativos	1.218694885 %	4.358037578 %

Fuente: Elaboración propia

Cuadro 8. Módulo 6

Resultado	Backpropagation	Resilient Backpropagation
Porcentaje de transacciones identificadas exitosamente	82.98231403 %	87.87815419 %
Porcentaje de transacciones identificadas incorrectamente	17.01768597 %	12.12184581 %
Proporción de identificación	2:10	1.25:10
Porcentaje de falsos positivos	4.380952381 %	0.02348643 %
Porcentaje de falsos negativos	0.065255732 %	4.514613779 %

Fuente: Elaboración propia

Cuadro 9. Módulo 7

Resultado	Backpropagation	Resilient Backpropagation
Porcentaje de transacciones identificadas exitosamente	94.77521264 %	87.89906594 %
Porcentaje de transacciones identificadas incorrectamente	5.224787363 %	12.10093406 %
Proporción de identificación	0:10	1.25:10
Porcentaje de falsos positivos	0.634920635 %	0.02348643 %
Porcentaje de falsos negativos	0.73015873 %	4.506784969 %

Fuente: Elaboración propia

Cuadro 10. Módulo 8

Resultado	Backpropagation	Resilient Backpropagation
Porcentaje de transacciones identificadas exitosamente	87.94786003 %	87.91300711 %
Porcentaje de transacciones identificadas incorrectamente	12.05213997 %	12.08699289 %
Proporción de identificación	1.25:10	1.25:10
Porcentaje de falsos positivos	0.015873016 %	0.02348643 %
Porcentaje de falsos negativos	3.0335097 %	4.501565762 %

Fuente: Elaboración propia

Cuadro 11. Módulo 9

Resultado	Backpropagation	Resilient Backpropagation
Porcentaje de transacciones identificadas exitosamente	78.66882679 %	87.9339188624 %
Porcentaje de transacciones identificadas incorrectamente	21.33117321 %	12.0660811376 %
Proporción de identificación	2.5:10	1.25:10
Porcentaje de falsos positivos	1.380952381 %	0.02348643 %
Porcentaje de falsos negativos	4.192239859 %	4.493736952 %

Fuente: Elaboración propia

Cuadro 12. Resultados de entrenamiento y validación de transacciones utilizando dos algoritmos de entrenamiento distintos por la red de decisiones

Resultado	Backpropagation	Resilient Backpropagation
Porcentaje de transacciones identificadas exitosamente	99.1837180009 %	70.4795761885 %
Porcentaje de transacciones identificadas incorrectamente	0.816281999147%	29.5204238115 %
Proporción de identificación	0:10	3.33333333333:10
Porcentaje de falsos positivos	0.6122115 %	0.02348643 %
Porcentaje de falsos negativos	0.2040705 %	11.02818372 %

Fuente: Elaboración propia

B. Resultados de SVM

Los resultados de las pruebas tipo 1 se pueden observar en el Cuadro 13. En las pruebas tipo 1 el porcentaje de falsos positivos tiene una relación con el porcentaje de las transacciones no fraudulentas utilizadas durante el entrenamiento. Adicionalmente el porcentaje de falsos negativos es muy bajo.

Cuadro 13. Resultados de pruebas tipo 1 realizadas.

Porcentaje de transacciones no fraudulentas utilizadas	Porcentaje de Aciertos	Porcentaje de falsos positivos	Porcentaje de falsos negativos
100.00%	86.47%	12.88%	54.99%
49.95%	56.29%	42.15%	0.00%
24.97%	51.11%	46.60%	0.00%
0.07%	**47.75%**	**49.97%**	**0.00%**
0.15%	46.41%	51.44%	0.00%
1.25%	46.10%	51.82%	0.00%

Fuente: Elaboración propia

El Cuadro 14 muestra los resultados de las pruebas tipo 2 realizadas. Se puede notar que los resultados no varían de los resultados de las pruebas tipo 1, realizadas con la misma cantidad de transacciones no fraudulentas.

Cuadro 14. Resultados de pruebas tipo 2 realizadas.

Porcentaje de transacciones no fraudulentas utilizadas	Porcentaje de aciertos	Porcentaje de falsos positivos	Porcentaje de falsos negativos
49.95%	56.29%	42.15%	0.00%
1.25%	45.85%	52.19%	0.00%

Fuente: Elaboración propia

El Cuadro 15 muestra los resultados de las pruebas tipo 3 realizadas. Las SVMs que fueron denominadas como mejores se encuentran en negrita. La SVM seleccionada como final (por ser más balanceada) se encuentra subrayada. Los mejores resultados se obtuvieron al dejar el peso de las transacciones no fraudulentas en 1 y disminuir el peso de las transacciones fraudulentas. Como se puede ver en la Figura 5, el peso de las transacciones fraudulentas tiene un impacto directo en el porcentaje de falsos positivos y negativos que detecta la SVM. Adicionalmente se puede notar que existe una relación inversa entre el porcentaje de falsos negativos y el de falsos positivos.

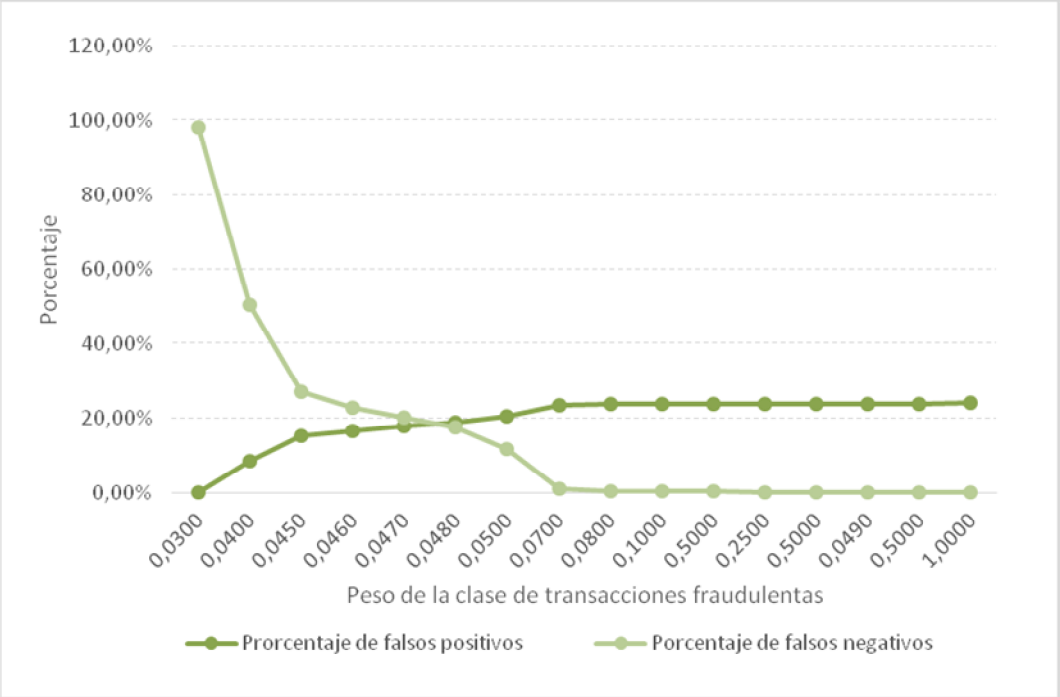
Finalmente en las Figuras 33 y 36 se puede ver la distancia de las transacciones, que se utilizaron para las pruebas, a los hiperplanos de las SVMs final y la número 10 de las pruebas de tipo 3 (la cuarta en negrita del Cuadro 13, en la que se utilizó 0.07 como peso para las transacciones fraudulentas), que se encuentra en 0. Las Figuras 34 y 35 muestran la misma información en un acercamiento entre -1 y 1. Las transacciones no fraudulentas se muestran en color azul y las fraudulentas en color rojo, en las cuatro gráficas.

Cuadro 15. Resultados de pruebas tipo 3 realizadas

Peso de transacciones fraudulentas	Peso de transacciones no fraudulentas	Porcentaje de aciertos	Porcentaje de falsos positivos	Porcentaje de falsos negativos
0.0100	1.0000	99.95%	0.00%	100.00%
0.0054	0.9946	99.95%	0.00%	100.00%
0.0300	1.0000	99.85%	0.10%	98.16%
0.0400	**1.0000**	**91.04%**	**8.65%**	**50.68%**
0.0450	1.0000	84.10%	15.34%	27.24%
0.0460	1.0000	82.65%	16.72%	22.85%
0.0470	1.0000	81.40%	17.91%	20.31%
0.0480	**1.0000**	**80.41%**	**18.85%**	**17.47%**
0.0500	1.0000	78.65%	20.51%	11.72%
0.0700	**1.0000**	**75.38%**	**23.62%**	**1.14%**
0.0800	1.0000	75.30%	23.70%	0.44%
0.1000	1.0000	75.25%	23.75%	0.33%
0.5000	1.5000	75.10%	23.89%	0.22%
0.2500	1.0000	75.08%	23.91%	0.15%
0.5000	1.0000	75.06%	23.93%	0.07%
0.0490	1.0000	75.05%	23.94%	0.07%
0.5000	1.0000	75.01%	23.97%	0.04%
1.0000	1.0000	74.98%	24.01%	0.04%
0.9946	0.0054	64.04%	35.32%	0.00%

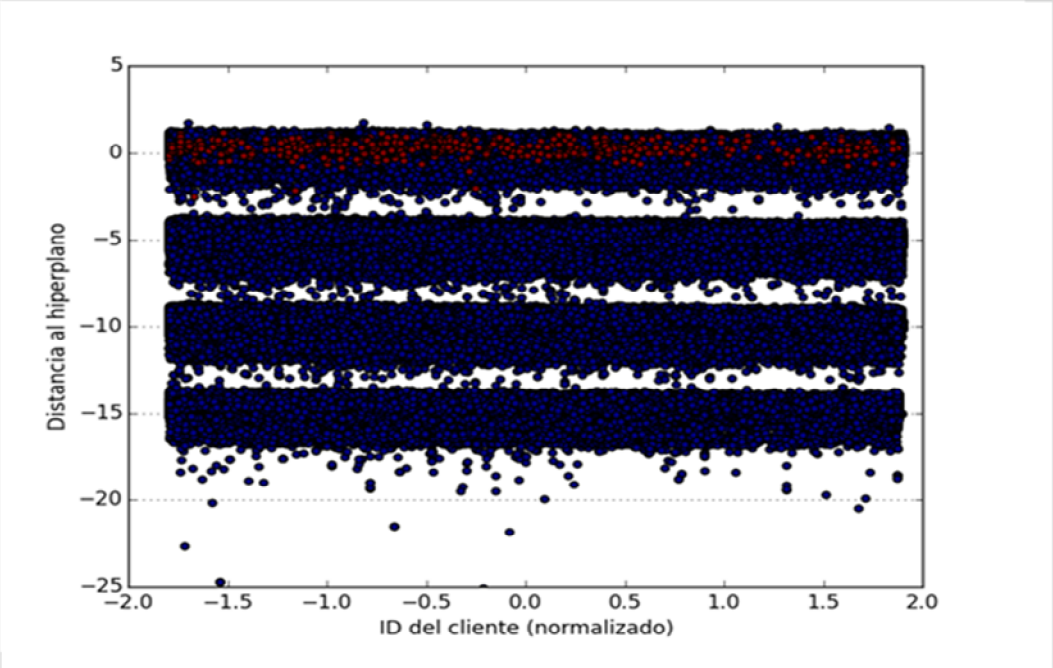
Fuente: Elaboración propia

Figura 32. Relación entre el peso de las transacciones fraudulentas y el porcentaje de falsos positivos y falsos negativos



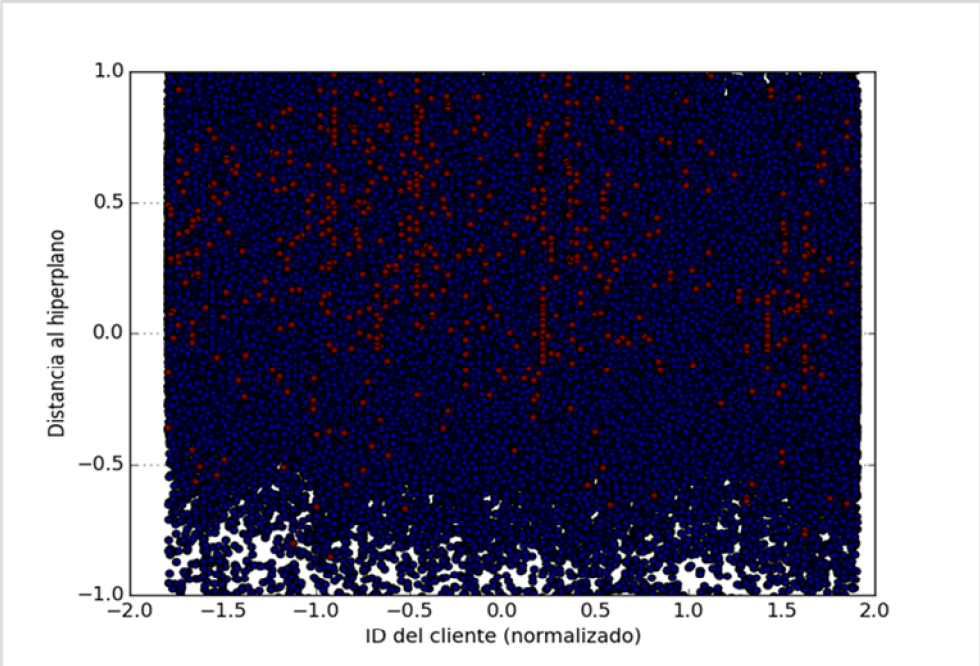
Fuente: Elaboración propia

Figura 33. Distancia de las transacciones al hiperplano de la SVM final.



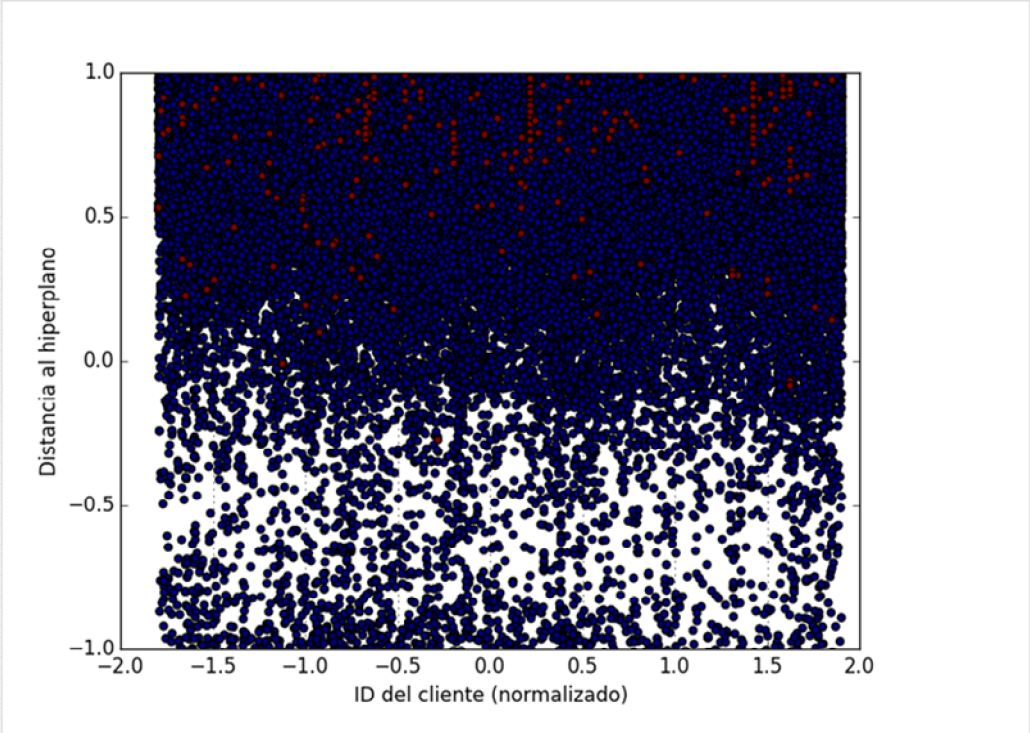
Fuente: Elaboración propia

Figura 34. Distancia de las transacciones al hiperplano de la SVM final (acercado).



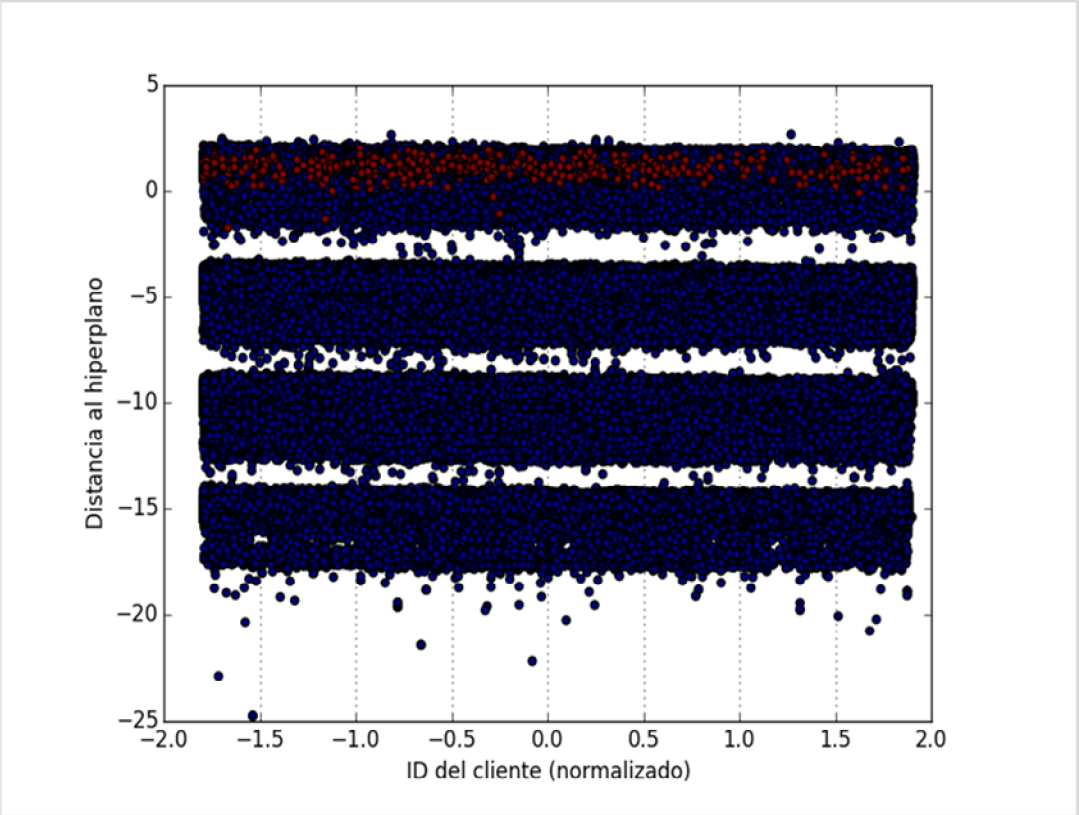
Fuente: Elaboración propia

Figura 35. Distancia de las transacciones al hiperplano de la SVM (acercado).



Fuente: Elaboración propia

Figura 36. Distancia de las transacciones al hiperplano de la SVM.



Fuente: Elaboración propia

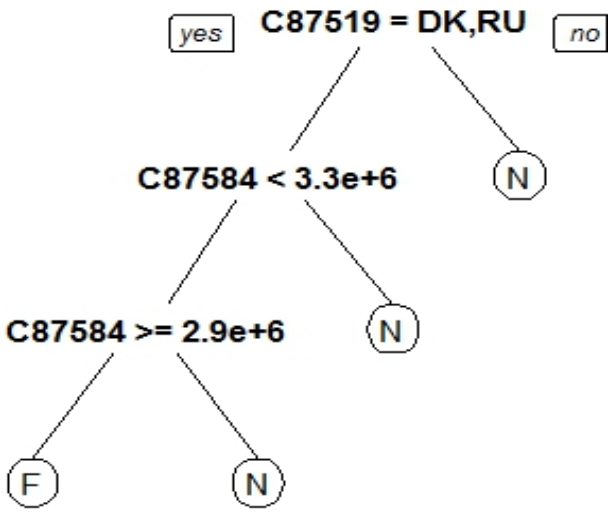
C. Resultados de árboles de decisión

La primera técnica utilizada para el reconocimiento de patrones es la de árboles de decisión. Este método es útil para el reconocimiento de patrones porque toma decisiones en base a los datos de entrada. Este modelo nos ayudará para predecir las variables que explicarán de mejor manera al campo C87601 que es el que define si una transacción es fraude o no.

En todos los árboles se utilizará la herramienta R para determinar y graficar los árboles de probabilidad. La exactitud del modelo original es de 0.9984

1. Árbol # 1. Se procedió a la obtención del árbol para predecir la columna de fraude. Según el principio de parsimonia entre más sencillo sea el modelo será mejor, siempre y cuando se obtenga una exactitud mayor. Este es el árbol obtenido:

Figura 37. Árbol de decisiones en corrida #1.



Fuente: Elaboración propia

Figura 38. Exactitud del árbol de decisiones #1.

```
> arbol <- rpart(C87601 ~ ., data=trxTrain, method="class", control=rpart.control(minbucket=4))
> # graficamos el arbol para ver la estuctura
> prp(arbol)
> # ahora creamos la prediccion sobre el set de datos de validacion
> # ahora creamos la prediccion sobre el set de datos de validacion
> pred <- predict(arbol,newdata=trxTest,type="class")
> # calculamos la matriz de confusion y la precision del modelo
> confumat <- table(trxTest$C87601, pred)
> confumat
   pred
        F      N
  F    17    387
  N     0 260355
> sum(diag(confumat))/sum(confumat)
[1] 0.9985159
```

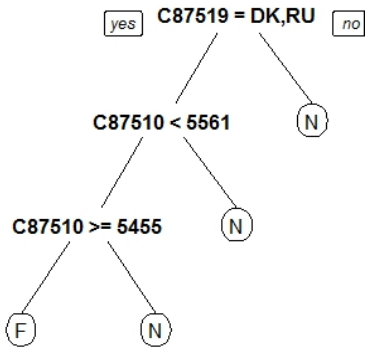
Fuente: Elaboración propia

La exactitud del modelo visto anteriormente es: 99.851% de exactitud y un recall de 1

2. **Árbol # 2.** En el segundo modelo se toman en cuenta todas las variables, la diferencia con el modelo anterior es que no se toma en cuenta la variable:

- C87584: ID Cliente

Figura 39. Árbol de decisiones en corrida #2.



Fuente: Elaboración propia

Figura 40. Exactitud del árbol de decisiones #2.

```
> arbol2 <- rpart(C87601 ~. -C87584, data=trxTrain, method="class", control=rpart.control(minbucket=4))
> # graficamos el arbol para ver la estuctura
> prp(arbol2)
> pred2 <- predict(arbol2,newdata=trxTest,type="class")
> # calculamos la nueva matriz de confusion y la precision del nuevo modelo
> confumat2 <- table(trxTest$C87601, pred2)
> confumat2
   pred2
        F      N
  F    15    389
  N     0 260355
> sum(diag(confumat2))/sum(confumat2)
[1] 0.9985082
```

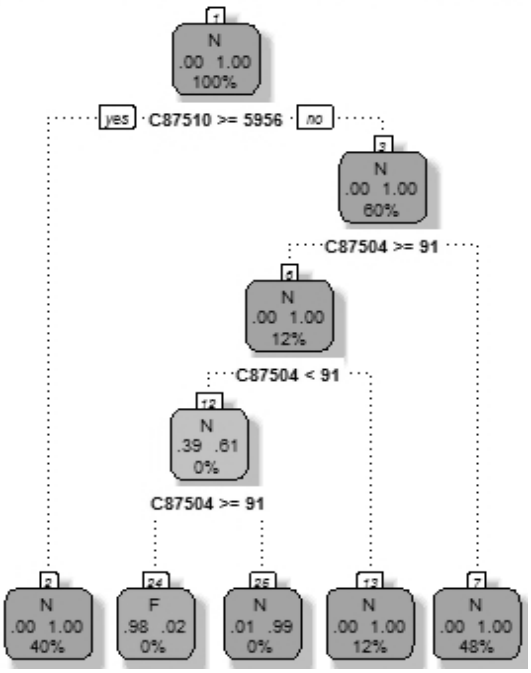
Fuente: Elaboración propia

La exactitud del modelo es de 0.9985 y un recall de 1

3. **Árbol # 3.** Se realiza otro modelo, en este caso se eliminaron las siguientes variables:

- ID Cliente
- Código de agencia
- Código país de origen

Figura 41. Árbol de decisiones en corrida # 3.



Fuente: Elaboración propia

Figura 42. Exactitud del árbol # 3.

Real	F	N
F	12	321
N	1	223500

Fuente: Elaboración propia

Exactitud de este modelo es 0.99856. y un recall de 0.99999

Cuadro 16. Exactitud de los modelos de árboles de decisión.

	Exactitud	Exactitud comparado con modelo inicial	Recall
Modelo base	0.99846	1.00007	
Modelo 1	0.9985273	1.00005	1
Modelo 2	0.9985082	1.00005	1
Modelo 3	0.9985614	1.00011	0.999996

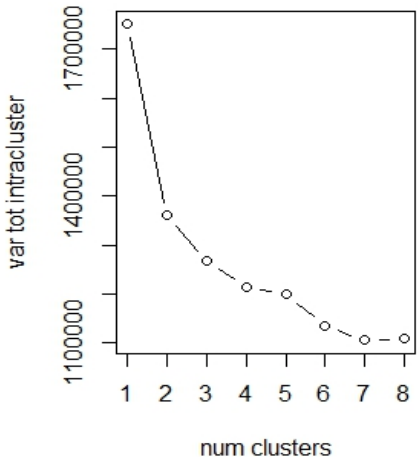
Fuente: Elaboración propia

D. Resultados de clustering

Clustering es una técnica de reconocimiento de patrones que se basa en la división de los datos en grupos similares o con las mismas características. El método a utilizar será no supervisado ya que no sabemos el resultado final que queremos tener por lo que el algoritmo que utilizaremos será K-means, este algoritmo trabaja basado en la similitudes entre clústers los cuales son medidos con respecto al valor promedio llamado centroide o centro de gravedad.

1. Modelo # 1.

Figura 43. Reducción en la variación total que se logra al incrementar un clúster.



Fuente: Elaboración propia

Se realizaron corridas con cuatro clústers:

Figura 44. Prueba con cuatro divisiones de clústers

```
> table(clustrx$cluster, trx$C87601)

      1      2
1   106 207659
2   370 209515
3   438 162822
4   238 164963
```

Fuente: Elaboración propia

Las pruebas respectivas con cinco clústers:

Figura 45. Prueba con cinco divisiones de clústers

```
> # ---------5clusters--------
> set.seed(1234567)
> clustrx1 <- kmeans(trx2,5)
> table(clustrx1$cluster, trx$C87601)

        1      2
1    106 200213
2    317 171714
3    388 136233
4    153  93770
5    188 143029
```

Fuente: Elaboración propia

Pruebas respectivas con seis clústers:

Figura 46. Prueba con seis divisiones de clústers

```
> # ---------6clusters--------
> set.seed(1234567)
> clustrx2 <- kmeans(trx2,6)
> table(clustrx2$cluster, trx$C87601)

        1      2
1     93 191120
2    274 148850
3    339 145179
4    146  90099
5    173 141536
6    127  28175
```

Fuente: Elaboración propia

Esta es la eficiencia del modelo # 4 con cuatro, cinco y seis clústers respectivamente:

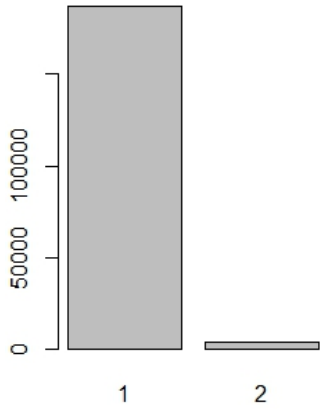
Figura 47. Porcentaje de efectividad con divisiones de cuatro, cinco y seis clústers

```
> #Porcentaje de cluster 1
> porcentaje1=table(trx$C87601,clustrx$cluster)
> porcentaje1[2,]/(porcentaje1[1,]+porcentaje1[2,])
        1         2         3         4
0.9994898 0.9982371 0.9973172 0.9985593
> #Porcentaje de cluster 2
> porcentaje2=table(trx$C87601,clustrx1$cluster)
> porcentaje2[2,]/(porcentaje2[1,]+porcentaje2[2,])
        1         2         3         4         5
0.9994708 0.9981573 0.9971600 0.9983710 0.9986873
> #Porcentaje de cluster 3
> porcentaje3=table(trx$C87601,clustrx2$cluster)
> porcentaje3[2,]/(porcentaje3[1,]+porcentaje3[2,])
        1         2         3         4         5         6
0.9995136 0.9981626 0.9976704 0.9983822 0.9987792 0.9955127
```

Fuente: Elaboración propia

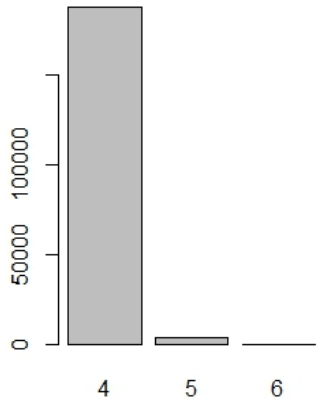
Se elige el clústers con mayor porcentaje de efectividad, por lo que en este caso estaría ubicado en la corrida con seis clústers en el grupo 5, el cual tiene efectividad de 0.9987. A continuación se presentarán las gráficas que muestran las características de las variables contenidas dentro del mismo.

Figura 48. Crédito o débito en corrida # 3 de clustering



Fuente: Elaboración propia

Figura 49. Marca o franquicia en corrida #3 de clustering



Fuente: Elaboración propia

Figura 50. País de origen en corrida #3 de clustering.

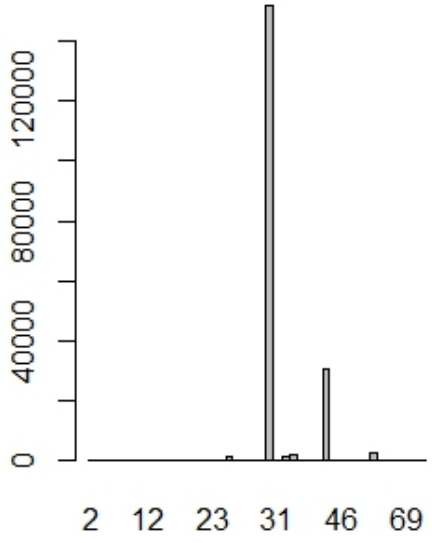
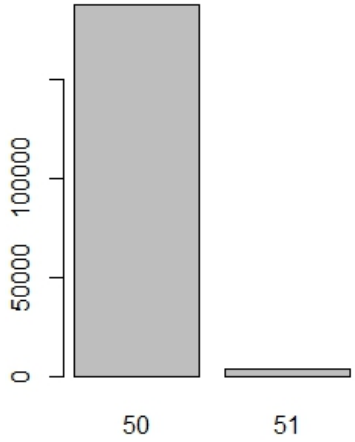
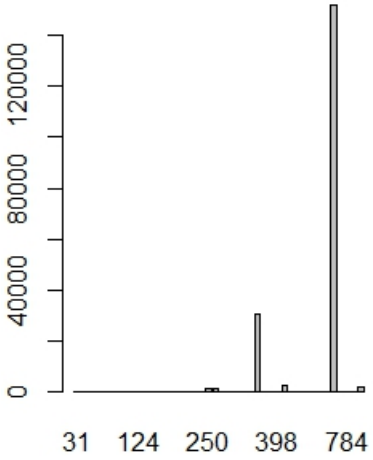


Figura 51. Condición del punto de venta en corrida #3 de clustering.



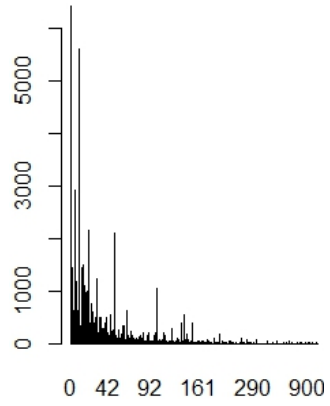
Fuente: Elaboración propia

Figura 52. Código de país adquirente en corrida #3 de clustering.



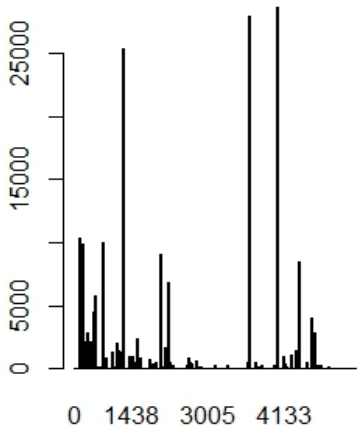
Fuente: Elaboración propia

Figura 53. Monto de transacción original en corrida #3 de clustering.



Fuente: Elaboración propia

Figura 54. Merchant category code (MCC) en corrida #3 de clustering.



Fuente: Elaboración propia

2. Modelo # 2. Estos son los dos clústers que el algoritmo relacionó. Existen dos grandes grupos dentro de los datos de entrada. Para los datos se utilizaron datos fraudulentos y no fraudulentos.

El primero, entre los campos más importantes se puede mencionar que:

- Monto original promedio es: 236.63
- La marca o franquicia promedio es 4
- Las transacciones fue en promedio en la semana 18
- La variable de las condiciones del punto es 50
- La mayoría de las transacciones se encuentran entre las categorías de negocios cercanas a 5441.

El segundo clúster podemos mencionar:

- Monto original promedio es: 169
- La marca o franquicia promedio es 4
- Las transacciones fue en promedio en la semana 18
- La variable de las condiciones del punto es 50
- La mayoría de las transacciones se encuentran entre las categorías de negocios cercanas a 5522.

3. Modelo # 3: Análisis de clúster con mayor fraude. Quedan los mismos clústers únicamente verificamos el clúster con mayor porcentaje de fraudes.

Figura 55. Matriz de porcentaje de fraude por clúster

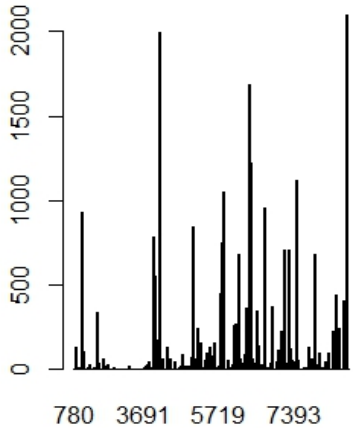
```
> porcentaje1[1,]/(porcentaje1[1,]+porcentaje1[2,])
          1           2           3           4
0.0005101918 0.0017628701 0.0026828372 0.0014406692
> #Porcentaje de cluster 2
> porcentaje2=table(trx$C87601,clustrx1$cluster)
> porcentaje2[1,]/(porcentaje2[1,]+porcentaje2[2,])
          1           2           3           4           5
0.000529156 0.001842691 0.002839973 0.001628994 0.001312693
> #Porcentaje de cluster 3
> porcentaje3=table(trx$C87601,clustrx2$cluster)
> porcentaje3[1,]/(porcentaje3[1,]+porcentaje3[2,])
          1           2           3           4           5           6
0.0004863686 0.0018373971 0.0023296087 0.0016178182 0.0012208117 0.0044873154
```

Fuente: Elaboración propia

El clúster con mayor fraude está ubicado en la corrida con tres clústers en el número 6 con 0.004487.

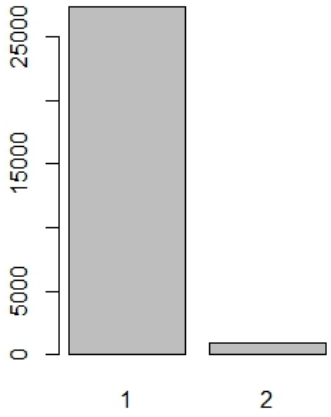
El siguiente análisis gráfico refleja patrones transaccionales y tendencias que siguiente los datos en este determinado clúster.

Figura 56. Merchant category code (MCC) en datos fraudulentos



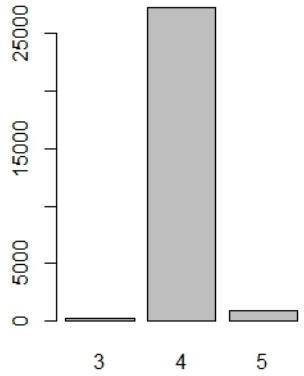
Fuente: Elaboración propia

Figura 57. Crédito y débito en datos fraudulentos



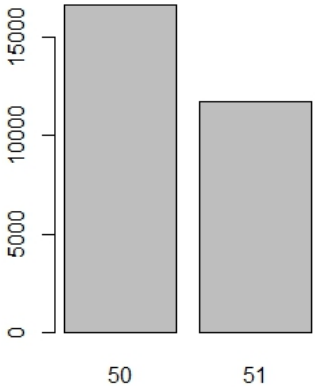
Fuente: Elaboración propia

Figura 58. Marca o franquicia en datos fraudulentos.



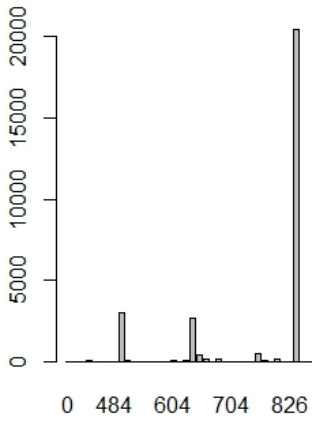
Fuente: Elaboración propia

Figura 59. Condición del punto de venta en datos fraudulentos.



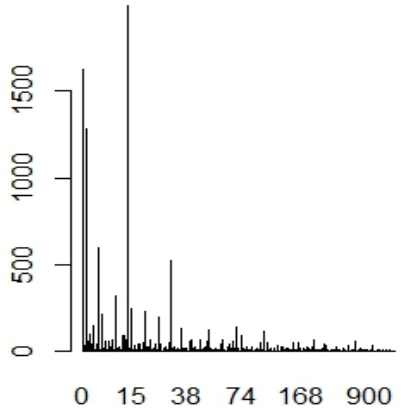
Fuente: Elaboración propia

Figura 60. Código país adquirente en datos fraudulentos.



Fuente: Elaboración propia

Figura 61. Monto original transacción en datos fraudulentos.



Fuente: Elaboración propia

E. Regresión logística

La regresión logística es otro método para determinar patrones de comportamiento en los datos, es útil cuando la variable que se va a predecir posee dos valores. En este caso es funcional porque la variable a predecir será el campo que define si la transacción es fraude o no, este campo únicamente tomará dos valores F o N respectivamente. Para estos modelos se utilizó una muestra de 1, 000,000 de datos debido a que se requería mayor recursos computacionales para entrenar a los algoritmos.

1. Modelo #1. En este modelo se incluyen todas las variables descritas anteriormente. Podemos ver que ciertas variables no son significativas para predecir el fraude. A continuación veremos que variables son las más significativas.

Figura 62. Variables utilizadas en corrida #1 de regresión logística

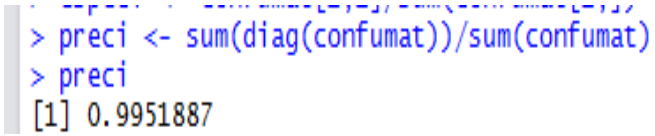
```
Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -2.591e+05  1.269e+05  -2.041 0.041204 *
C87500       9.191e-14  2.823e-14   3.256 0.001131 **
C87550       2.575e-01  2.650e-01   0.972 0.331222
C87504       1.962e-06  8.498e-06   0.231 0.817405
C87506       3.210e-06  7.858e-07   4.085 4.40e-05 ***
C87507       1.290e-02  6.308e-03   2.045 0.040839 *
C87510      -2.378e-04  2.558e-05  -9.295  < 2e-16 ***
C87543      -6.316e-05  2.647e-04  -0.239 0.811430
C87511      -1.957e+00  4.492e+01  -0.044 0.965241
C87512      -2.863e-01  1.291e-01  -2.219 0.026507 *
C87513       1.594e-04  2.114e-05   7.542 4.64e-14 ***
C87531      -2.324e-10  1.861e-10  -1.249 0.211605
C8756779    -3.072e+02  1.199e+03  -0.256 0.797765
C87567BC    -3.190e+02  9.614e+01  -3.318 0.000907 ***
C87567CL    -3.183e+02  9.619e+01  -3.309 0.000937 ***
C87567EL    -3.152e+02  9.621e+01  -3.277 0.001051 **
C87567EM    -3.172e+02  9.628e+01  -3.294 0.000987 ***
C87567EU    -3.198e+02  9.613e+01  -3.327 0.000878 ***
C87567GD    -3.182e+02  9.620e+01  -3.308 0.000941 ***
C87567IF    -3.192e+02  9.604e+01  -3.323 0.000890 ***
C87567PL    -3.172e+02  9.624e+01  -3.296 0.000981 ***
C87547       3.176e-03  4.678e-04   6.788 1.13e-11 ***
C87566       2.220e+00  1.060e+00   2.095 0.036187 *
C87535      -9.008e-04  2.677e-03  -0.336 0.736506
C87584      -6.694e-09  5.142e-09  -1.302 0.192977
C87593      -7.278e+00  2.010e-01 -36.216  < 2e-16 ***
C87594S      1.338e+00  1.417e-01   9.446  < 2e-16 ***
C87675      -2.308e-09  2.662e-07  -0.009 0.993083
C87714S      8.578e-01  4.956e-01   1.731 0.083471 .
```

Fuente: Elaboración propia

- C87500: Llave primaria de control (Tarjeta o ID Cliente)
- C87506: Hora TRX
- C87507: Fecha TRX
- C87510: MCC
- C87512: Condición Pto Venta
- C87513: ID Adq
- C87567: Tipo Prod TC
- C87547: Cod Moneda Trx
- C87566: Marca o Franquicia
- C87593: Semana del Año
- C87594: Grupo Día
- C87675: Bin & MCC
- C87714: Evaluación Dispositivo Chip

Según el modelo de RL estas son las variables que mejor explican el fraude. Este modelo alcanza una precisión de 99.5188%

Figura 63. Precisión del modelo #1 de regresión logística



Fuente: Elaboración propia

2. Modelo # 2. Para este modelo se utiliza los mismos datos únicamente que se omiten algunas variables no significativas del modelo anterior. Las variables significativas para el modelo son:

- C87500: Llave primaria de control (Tarjeta o ID Cliente)
- C87506: Hora TRX
- C87507: Fecha TRX
- C87510: MCC
- C87512: Condición Pto Venta
- C87513: ID Adq
- C87567: Tipo Prod TC
- C87547: Cod Moneda Trx
- C87566: Marca o Franquicia
- C87593: Semana del Año

- C87594: Grupo Día

Al eliminar una de las variables no significativas podemos ver que otras variables que antes aportaban al modelo ahora ya no. Con este modelo se alcanza una precisión de: 99.518%

Figura 64. Precisión del modelo #2 de regresión logística.

```
        F         N
0.4117887 0.9947613
> # Matriz de Confusion para un umbral de 0.7
> confumat2 <- table(tranTrain$C87601, predictTrain2 > 0.7)
> # Sensitividad, especificidad y precison del modelo
> preci <- sum(diag(confumat2))/sum(confumat2)
> preci
[1] 0.9951887
```

Fuente: Elaboración propia

3. **Modelo # 3 y 4.** Al valuar y eliminar algunas de las variables que no aportan al modelo se obtuvo la misma precisión por lo que no se incluye los modelos porque la precisión alcanzada es la misma

4. **Modelo # 5.** Para este modelo se utiliza los mismos datos únicamente que se omiten algunas variables no significativas del modelo anterior. Las variables significativas para el modelo son:

- C87506: Hora TRX
- C87507: Fecha TRX
- C87510: MCC
- C87512: Condición Pto Venta
- C87513: ID Adq
- C87547: Cod Moneda Trx
- C87566: Marca o Franquicia
- C87584: ID Cliente
- C87593: Semana del Año
- C87594: Grupo Día
- C87714: Evaluación Dispositivo Chip

Al eliminar una de las variables no significativas podemos ver que otras variables que antes aportaban al modelo ahora ya no. Con este modelo se alcanza una precisión de: 99.52%

Figura 65. Precisión del modelo #5 de regresión logística

```
preci5 <- sum(diag(confumat5))/sum(confumat5)
preci5
[1] 0.9952019
```

Fuente: Elaboración propia

5. Modelo # 6. Para este modelo se utiliza los mismos datos únicamente que se omiten algunas variables no significativas del modelo anterior. Las variables significativas para el modelo son:

- C87506: Hora TRX
- C87507: Fecha TRX
- C87510: MCC
- C87512: Condición Pto Venta
- C87513: ID Adq
- C87547: Cod Moneda Trx
- C87584: ID Cliente
- C87593: Semana del Año
- C87594: Grupo Día

Figura 66. Variables utilizadas para la corrida # 6 de regresión logística.

```
Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept) -2.388e+05  1.268e+05  -1.884 0.059591 .
C87506       2.820e-06  7.681e-07   3.672 0.000241 ***
C87507       1.188e-02  6.301e-03   1.885 0.059435 .
C87510      -2.207e-04  2.423e-05  -9.106  < 2e-16 ***
C87512      -3.026e-01  1.280e-01  -2.365 0.018040 *
C87513       1.446e-04  1.908e-05   7.581 3.44e-14 ***
C87547       3.150e-03  4.349e-04   7.244 4.34e-13 ***
C87584      -8.814e-09  5.103e-09  -1.727 0.084128 .
C87593      -7.262e+00  1.965e-01 -36.951  < 2e-16 ***
C87594S      1.332e+00  1.413e-01   9.427  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 10236.7  on 75862  degrees of freedom
Residual deviance:  4619.2  on 75853  degrees of freedom
AIC: 4639.2

Number of Fisher Scoring iterations: 9
```

Fuente:Elaboración propia

Al eliminar una de las variables no significativas podemos ver que otras variables que antes aportaban al modelo ahora ya no. Con este modelo se alcanza una precisión de: 99.52%

Figura 67. Precisión del modelo # 6 de regresión logística

```
preci6 <- sum(diag(confumat6))/sum(confumat6)
preci6
1] 0.9952019
```

Fuente: Elaboración propia

6. Modelo # 7. Para este modelo se utiliza los mismos datos únicamente que se omiten algunas variables no significativas del modelo anterior. Las variables significativas para el modelo son:

- C87506: Hora TRX
- C87510: MCC
- C87512: Condición Pto Venta
- C87513: ID Adq
- C87547: Cod Moneda Trx
- C87593: Semana del Año
- C87594: Grupo Día

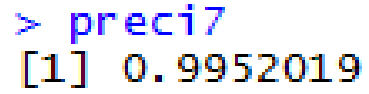
Figura 68. Variables utilizadas para la corrida # 7 de regresión logística.

```
    Min       1Q   Median       3Q      Max
-3.8796   0.0603   0.0874   0.1160   8.4904

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  1.463e+02  7.431e+00  19.690  < 2e-16 ***
C87506       2.692e-06  7.653e-07   3.517 0.000436 ***
C87510      -2.212e-04  2.423e-05  -9.128  < 2e-16 ***
C87512      -3.112e-01  1.278e-01  -2.435 0.014906 *
C87513       1.415e-04  1.901e-05   7.442 9.89e-14 ***
C87547       3.121e-03  4.357e-04   7.162 7.96e-13 ***
C87593      -7.142e+00  1.891e-01 -37.768  < 2e-16 ***
C87594S      1.274e+00  1.347e-01   9.454  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 10236.7  on 75862  degrees of freedom
Residual deviance:  4615.8  on 75855  degrees of freedom
AIC: 4631.8

Number of Fisher Scoring iterations: 9
```

Fuente: Elaboración propia

Al eliminar una de las variables no significativas podemos ver que otras variables que antes aportaban al modelo ahora ya no. Con este modelo se alcanza una precisión de: 99.52%

Figura 69. Precisión del modelo # 7 de regresión logística.



Fuente: Elaboración propia

F. Resultados de redes bayesianas

El análisis final se llevó a cabo utilizando 200,000 transacciones ajenas a las utilizadas para el entrenamiento de las redes bayesianas. Estas transacciones contenían 199,941 transacciones válidas (no fraudulentas) y 59 transacciones fraudulentas.

Como resultado final al analizar el conjunto de datos descrito anteriormente, se obtuvo que:

- 143,264 transacciones fueron categorizadas correctamente. Es decir, si una transacción era fraudulenta fue marcada como tal, de lo contrario no se marcó como fraude. Esta cantidad de transacciones representa una efectividad de 71.63% del total de transacciones evaluadas.
- 56,736 transacciones fueron categorizadas de forma errónea. Es decir, si una transacción fue marcada como fraude, ésta originalmente era no fraudulenta, o bien, si la transacción original era fraude, la red bayesiana no la detectó como tal. Esta cantidad de transacciones representa un 28.37% del total de transacciones evaluadas.

Cuadro 17. Distribución de resultados

	Cantidad	**Porcentaje**
Aciertos	143,264	71.63%
Desaciertos	56,736	28.37%
Total	**200,000**	**100%**

Fuente: Elaboración propia

- 56,696 transacciones se catalogaron como falsos positivos. Es decir, la transacción original no era fraudulenta, sin embargo, la red bayesiana la categorizó como fraude. Esta cantidad de transacciones representa un 99.93% del total de transacciones categorizadas de forma errónea.
- 40 transacciones se catalogaron como falsos negativos. Es decir, la transacción original era fraudulenta, sin embargo, la red bayesiana la categorizó como transacción válida. Esta cantidad de transacciones representa un 0.07% del total de transacciones categorizadas de forma errónea.

Cuadro 18. Distribución de desaciertos

	Cantidad	Porcentaje
Falsos positivos	56,696	99.93%
Falsos negativos	40	0.07%
Total	56,736	100%

Fuente: Elaboración propia

- 19 transacciones originalmente fraudulentas fueron marcadas correctamente. Es decir, la transacción original era fraude y el análisis de la red bayesiana la categorizó como tal. Esta cantidad de transacciones representa un 32.20% del total de transacciones originalmente fraudulentas.

Cuadro 19. Distribución de aciertos en transacciones originalmente fraudulentas

	Cantidad	Porcentaje
Aciertos	19	32.20%
Desaciertos	40	67.80%
Total	59	100%

Fuente: Elaboración propia

VIII. ANÁLISIS DE RESULTADOS

Como se vio anteriormente los datos fueron proporcionados por la empresa, se solicitó un set de datos con historial por cliente, teniendo alrededor de 1, 000,000 de clientes. El set de datos proporcionados únicamente fueron transacciones aisladas de clientes con un determinado tiempo. Los análisis se vieron muy limitados debido al set de datos proporcionados ya que no se contaba con historial por cliente por lo que no se logró realizar un análisis más profundo como se deseaba. La empresa solicitó un análisis de reconocimiento de patrones y clustering que por la naturaleza de las variables tampoco se logró llegar a un mejor análisis. No obstante, se logró determinar características importantes y con un cierto grado de relevancia para la empresa. A pesar de las dificultades con los datos se presentan a continuación la serie de resultados obtenidos a lo largo de esta investigación.

A. Análisis de resultados de redes neuronales

A continuación se describe la estructura de redes neuronal propuesta para todas las redes modulares en los escenarios planteados anteriormente. La lógica de las selecciones se basa en los resultados de esta sección.

Cuadro 20. Selección de parámetros de entrenamiento y estructura de la red neuronal.

Parámetro	Valor	Motivo de selección
Impulso	0.74	El impulso se inicia como un valor alto, en un rango entre 0 a 1. Esto se realiza en base a la utilidad del impulso en el algoritmo de descenso utilizado sobre la superficie del error de la fase de entrenamiento. En la Fig. 11, se puede observar como en este valor existe un resultado promedio estable con respecto a los porcentajes de validaciones correctas.
Tasa de aprendizaje	0.0009	La tasa de aprendizaje debe ser un número pequeño, el rango propuesto por Kriesel, D. (2005) puede ser ampliado debido a la capacidad del sistema en el que se llevo a cabo la ejecución, este equipo poseía 8 GB de memoria RAM para realizar los cálculos. La tasa es útil al mantenerse en un valor pequeño debido a que indicada la variación del algoritmo con respecto

Continuación Cuadro 20. Selección de parámetros de entrenamiento y estructura de la red neuronal.

Parámetro	Valor	Motivo de selección
		a la ubicación de mínimos en la gráfica del error de entrenamiento. A medida que el valor sea más pequeño la aproximación hacia el mínimo trabaja con intervalos podrían variar fuertemente. En la Figura 12, se puede observar que el resultado es más estable en la selección indicada.
Capas ocultas	4	Las capas ocultas fueron definidas en base a la recomendación de Kriesel, D. (2005), en donde establecía que después de 3 capas ocultas era más difícil observar la mejoría en el algoritmo en base a la capacidad de ubicación de regiones sobre la información.
Total de épocas	100	A través de las ejecuciones realizadas se generó gráficas, ver Figura 10, con los valores de error a medida que se entrenaba la red neuronal con un máximo de 200 épocas. Se pudo observar que la convergencia a un error mínimo tomado como absoluto, al no encontrar un valor mínimo mayor, ocurría mucho antes de llegar al máximo de épocas definidas en el algoritmo de entrenamiento. Es importante notar que la convergencia entre el algoritmo de propagación hacia atrás de errores y el algoritmo de propagación hacia atrás elástica de errores diferían, siendo el algoritmo elástico el que convergía con mayor exitoso.
Uso de neuronas de sesgo	Si	En la Figura 13 se puede observar que se obtuvo mejores resultados con estructuras que incluían neuronas de sesgo. Como explica Kriesel, D. (2005) esto puede explicarse por la capacidad de adecuación de los umbrales de cada neurona y la capacidad de simular problemas más complejos.
Uso de recurrencia en la red neuronal	Sí	En la Figura 14 se puede observar que se obtuvo mejores resultados utilizando la recurrencia como parte de la descripción de las neuronas en la red. En el marco teórico se había visto que este parámetro puede llegar a intensificar el valor original de una variable en base a la realimentación obtenida del algoritmo de propagación hacia atrás.

Continuación Cuadro 20. Selección de parámetros de entrenamiento y estructura de la red neuronal.

Parámetro	Valor	Motivo de selección
Algoritmo de entrenamiento	Propagación hacia atrás elástica (Resilient Backpropagation)	En la selección del algoritmo utilizado para entrenar las redes neuronales se eligió la opción modificada de propagación hacia atrás. Esta versión probó converger a un mínimo en la superficie de error del algoritmo más rápido que la versión sin modificaciones. En base a Kriesel, D. (2005), se puede concluir que esta diferencia en la convergencia reside en las modificaciones realizadas al cálculo del cambio de los pesos. Debido a que se realiza una simplificación del método el cálculo se optimiza.
Función de activación de las neuronas en la capa oculta	Sigmoidal	En la Figura 16 se puede observar los resultados de las ejecuciones con respecto a la función de activación utilizada en las neuronas de la capa oculta de la red neuronal. En estos resultados se observa que los mejores porcentajes de validaciones se obtienen utilizando la función de activación Sigmoidal.
Función de activación de las neuronas en la capa de salida	Sigmoidal	En la Figura 17 se observan distintas ejecuciones realizadas con variadas funciones de activación en las neuronas de la capa de salida. Como se puede observar hay funciones de activación que permiten clasificar de mejor manera los resultados en base al mismo número de épocas transcurridas. La función que mejor logra clasificar los datos en dos grupos, fraudulenta y legítima, es la función de activación Sigmoidal visible en (A).

Fuente: Elaboración propia

El análisis que se realizo para obtener la importancia de las variables mostró que los campos que describían a la tarjeta utilizada en la transacción incidían más en el resultado de clasificación de la red neuronal. *V.g.* la marca a la que pertenecía la tarjeta, el tipo de producto de la tarjeta y el monto. Las variables que menos incidencia tenían con el resultado de la red neuronal eran las características de ubicación y la variable de semana de la transacción.

En los resultados del entrenamiento y validación de las redes modulares y la red de decisiones se pudo observar un patrón más estable en el algoritmo de propagación hacia atrás elástico. La mayoría de los resultados estuvieron en 87% de efectividad de identificación de la clase a la que pertenecía la transacción al momento de validarla. La mejor proporción obtenida de identificaciones falsas por total de transacciones

fue de 1.25 falsos por cada 10 transacciones y el peor fue de 1.42 falsos por cada 10 transacciones. En contraste el entrenamiento de las redes modulares con el algoritmo sin modificación mostraron resultados más variados, con porcentajes de validaciones correctas entre 61% Y 94% y con la mejor proporción de 0 falsos por cada 10 transacciones y la peor con 5 falsos por cada 10 transacciones. Sin embargo el resultado final de la red de decisiones comprobó que el algoritmo de propagación de errores hacia atrás sin modificación fue más efectivo que la versión elástica siendo 99% contra 70%. Es importante notar que la proporción de los falsos negativos y falsos positivos con respecto al total de transacciones evaluadas no excedió más del 8% de las transacciones. Estos resultados indican que para la estructura modular utilizada el cálculo completo del error del vector resultante de la red neuronal es más efectivo que la versión simplificada del algoritmo elástico. Además la modificación que recibe el algoritmo elástico en donde modifica su valor de tasa de aprendizaje de manera individual por cada peso podría haber sobre-ajustado el valor en los casos de la data de entrenamiento. Esta podría ser la razón por la cuál el resultado no fue tan variable en este caso. En todos los casos de entrenamiento el porcentaje de falsos negativos resultó más alto que el porcentaje de falsos positivos, esto recalca que la distribución sesgada de los datos fue un componente fundamental en el proceso de creación de las redes modulares y la red de decisiones.

B. Análisis de resultados SVM

Las pruebas tipo 1 no generaron las mejores SVM ya que al disminuir el número de transacciones no fraudulentas utilizadas el porcentaje de falsos positivos aumentó, aunque el porcentaje de falsos negativos fue de cero en la mayoría de las pruebas. Al momento de plantear este tipo de pruebas ya se había tomado en cuenta este factor y por esta razón se crearon las pruebas de tipo 2. Sin embargo estas tampoco dieron buenos resultados ya que no mejoraron los resultados obtenidos en las pruebas de tipo 1. Ya que ninguna de este tipo de pruebas logró su objetivo, que era disminuir el número de falsos positivos, se enfocaron los esfuerzos en las pruebas de tipo 3.

Las pruebas tipo 3 realizadas mostraron mayor flexibilidad que los otros tipos de pruebas. Estas pruebas produjeron tanto SVMs que podían detectar correctamente todas las transacciones fraudulentas, generando una alta cantidad de falsos positivos, como SVMs que podían detectar correctamente todas las transacciones no fraudulentas, generando una alta cantidad de falsos negativos. Por tanto, dependiendo del peso que se le asignara a las transacciones fraudulentas, se pudo generar una SVM especializada o una con un mayor balance.

Se determinaron cuáles eran las mejores SVMs gracias a su porcentaje de aciertos general así como del porcentaje de los falsos positivos y falsos negativos. La SVM más balanceada fue elegida como la SVM final ya que el objetivo general es disminuir tanto los falsos positivos como los falsos negativos.

Se tomó la SVM final como la más balanceada ya que, como se ve en la Figura 32, la relación que existe entre el porcentaje de falsos positivos y el de falsos negativos no permite que una de estos porcentajes baje sin que el otro suba. Adicionalmente al ver las Figuras 33 y 34, que muestran la distancia al hiperplano, se puede notar que no existe una clara separación entre las transacciones fraudulentas y las no fraudulentas, por lo que se hace más evidente la relación entre falsos positivos y falsos negativos. Esto se debe a que si se mueve el hiperplano para permitir que más transacciones sean identificadas como fraudulentas (como se puede ver en las Figuras 36 y 35) muchas de las transacciones no fraudulentas serán tomadas como fraudulentas.

C. Análisis de resultados de árboles de decisiones.

Los árboles de decisión será de suma utilidad para este proyecto ya que nos dará un criterio para seleccionar que transacción será fraudulenta y cual no. Los árboles de decisiones solo analizarán los datos y darán el criterio para definir las transacciones.

1. Árbol de decisión # 1. Esta primera corrida nos dice que durante el proceso de análisis de datos, el algoritmo detectó 2 variables que explican de manera muy acertada que transacción es fraudulenta y cual no. Según esta primera corrida las variables que explican de mejor manera los fraudes son:

- C87519: País de origen
- C87584: ID Cliente

A simple vista lo que el árbol nos muestra es que existen dos posibilidades dentro de los datos, el modelo toma los siguientes criterios para determinar cuando la transacción es fraude en base al campo C87601 que es el que define si una transacción es fraude o no. El criterio de selección es el siguiente:

Si la transacción no proviene de los países DK, RU no serán fraude. Mientras que si pertenecen a estos países pasará a un segundo nivel, verifica si el ID del cliente es menor a 3.3e^6 no será fraude mientras que si el ID es mayor a esa cifra pasará a un 3er nivel. Luego pregunta si el ID del cliente es mayor o igual a 2.9e^6 no será fraude mientras que si es menor será marcado como fraude. El modelo muestra una efectividad del árbol de 99.852% lo que significa que un casi detectó correctamente las transacciones, este modelo tan solo se equivocó un 0.1480% de veces.

Este primer modelo nos muestra que el país de origen es importante para definir si una transacción es fraude o no aunque es poco útil ya que el ID del cliente únicamente es un identificar, no agrega valor a la información por lo que no es de gran ayuda para la investigación.

2. Árbol de decisión # 2. Para el análisis del modelo # 2 de árbol de decisiones no se toma en cuenta la variable ID de cliente ya que únicamente es un identificador que no aporta valor al modelo por lo que se deja fuera para verificar otras variables relevantes.

Este modelo de árboles de decisión nos muestra que únicamente dos variables son las importantes nuevamente:

- C87519: País de origen
- C87510: Merchant Category Code

Lo que el árbol quiere decirnos es que si la transacción no proviene de los países DK, RU no serán fraude. Mientras que si pertenecen a estos países pasará a un segundo nivel, verifica si la categoría del negocio (MCC) es mayor a 5661 no será fraude mientras que si la categoría del comercio es menor a esa cifra pasará a un 3er nivel. Luego pregunta si el MCC es menor o igual a 5455 no será fraude mientras que si es mayor será marcado como fraude. La precisión del modelo es: 99.850% de efectividad, comparando con el modelo anterior ligeramente es superior a este por lo que podemos decir que el modelo anterior es mejor.

Lo importante de este modelo es que establece un rango de categoría de negocios, 5455-5561, en los cuales están abarcados la mayor cantidad de fraudes en los datos. La empresa proporcionó un catalogo de campos pero no se define, por seguridad, el nombre exacto de la categoría al que pertenecen el rango determinado por el modelo, por lo que únicamente se nombró el código de las categorías de negocios. El tener un rango de MCC es muy importante porque quiere decir que existen ciertas categorías de negocios en donde el índice de fraude es mayor, por lo que es tarea de las franquicias de tarjetas de crédito o débito aumentar las medidas de seguridad en este tipo de negocios. Este segundo modelo es de mayor relevancia para la investigación ya que nos proporcionó valor a la investigación y se determinó variables que pueden ser trascendentales en la búsqueda de la eliminación del fraude en tarjetas de crédito o débito.

D. Análisis de resultados de Clustering

1. Modelo # 1 de Clustering general. Como parte de la investigación se realizó un estudio del comportamiento de los datos para tener una visión más amplia y entender como están distribuido los mismos. El modelo # 1 se basa en dar un repaso general de los datos y encontrar el clúster con mayor impacto en los datos.

Se realizó un análisis con 4, 5 y 6 clústers para determinar las diferentes características de los grupos. Se determinó esta cantidad de clústers debido a que se realizó una gráfica de reducción en la variación total que se logra al incrementar un clúster, claramente sabemos que el clúster con mayor

reducción es de 1 a 2 clúster pero queremos analizar que otros clústers podemos encontrar, por lo que se decidió tomar de 4 a 6 clústers.

En las figuras de la 44 a la 46 podemos ver la distribución por clústers de la variable de fraude, lo que necesitamos es buscar el clúster con mayor porcentaje de efectividad. Esto nos dice que clúster agrupó mejor la variable de fraude. Esto se obtuvo con la Figura 47, la cual muestra la efectividad por clúster. El análisis de variables se realizará en este clúster porque es el que mejor agrupo la variable, el mejor clústers es el de corrida 6 en el grupo 5, el cual tiene efectividad de 0.9987.

Las variables alojadas en este clúster nos revelan datos interesantes, la mayoría de transacciones pertenecen a la línea de tarjetas de crédito, siendo Visa Inc. la mayor casa emisora de tarjetas de este grupo. Por temas de confidencialidad se limitará a decir que el país con mayor emisión es el 31, las condiciones del punto de venta se limita a ser 50 y la mayor cantidad de transacciones se encuentra entre $0-$90.

Este es un resumen breve de los datos en general. Los siguientes análisis se enfocarán en recopilar información más precisa con referente a los datos fraudulentos.

2. Modelo # 2 de Clustering segmentado en dos grupos. De antemano sabemos que existen dos clasificaciones grandes entre los datos, transacciones fraudulentas y transacciones no fraudulentas. Esta es la razón por el cual se optó para realizar el primer análisis de clustering con dos clústers únicamente.

El primero, entre los campos más importantes se puede mencionar que:

- Monto Original promedio es: 236.63
- La marca o franquicia promedio es 4
- Las transacciones fue en promedio en la semana 18
- La variable de las condiciones del punto es 50
- La mayoría de las transacciones se encuentran entre las categorías de negocios cercanas a 5441.

El segundo clúster podemos mencionar:

- Monto Original promedio es: 169
- La marca o franquicia promedio es 4
- Las transacciones fue en promedio en la semana 18
- La variable de las condiciones del punto es 50

- La mayoría de las transacciones se encuentran entre las categorías de negocios cercanas a 5522.

Nos damos cuenta que el primer clúster muestra las datos que no son fraudulentos, ya que concuerda con el análisis de árboles de decisión, el MCC del primer clúster está fuera del rango de categorías de negocios que poseen transacciones fraudulentas. Este clúster nos dice que el monto de las transacciones no fraudulentas está alrededor de $235.63, la franquicia de tarjetas que tiene este clúster pertenece a la empresa Visa Inc.

En el caso del segundo clúster notamos que vuelve a coincidir con la interpretación del análisis de arboles de decisión ya que el MCC está dentro del rango de transacciones fraudulentas. El monto promedio de las transacciones es de $169, notamos un decremento en el monto de la transacción en comparación con el clúster 1, ambos clústeres están manejados por la franquicia de tarjetas Visa Inc.

De acuerdo a los datos proporcionados podemos inferir que la mayoría de transacciones fraudulentas no excedieron el monto de $169 lo que nos indica que probablemente no se tiene controles de seguridad estrictos cuando son montos relativamente pequeños por lo que las personas que se dedican al fraude utilizan montos pequeños para llevar a cabo el robo.

3. Modelo #3 de Clustering con mayor fraude. En el modelo # 3 se analizará el clúster con mayor fraude detectado. Nuevamente la Figura 47 nos muestra el porcentaje de efectividad al detectar los datos fraudulentos. Por la naturaleza de la base de datos se cuenta con un número muy bajo de transacciones fraudulentas en comparación con las transacciones no fraudulentas por lo que el clúster con mayor fraude está ubicado en la corrida con 3 clústers en el número 6 con 0.004487.

El análisis nuevamente es gráfico, el clúster nos muestra que nuevamente existe un rango muy importante en el MCC, el rango que muestra la gráfica es 3691-7393 por lo que nuevamente vemos que esta variable es sumamente importante en la detección de una transacción fraudulenta, no significa que no haya en otros MCC´s pero la mayoría de datos están alojados en este rango. Un alto porcentaje de los datos fraudulentos fueron realizados con tarjetas de crédito. La casa emisora de tarjetas nos refleja que sólo tres empresas están en este clúster, Diner's Club, Visa, Master card, de estas tres empresas Visa presenta el mayor índice de transacciones fraudulentas en los datos. En términos de las condiciones del punto de venta podemos recalcar que la mayoría de datos tienen una condición de 50 aunque no es grande la diferencia con los datos con condiciones de 51. La Figura 52 muestra datos muy importantes ya que la mayoría de datos fraudulentos se encuentran con código de país adquirente de 826, esto puede ser muy significativo para la investigación ya que muestra bastante diferencia este país con referente a los demás. Los montos de las transacciones muestran información importante ya que la mayoría de montos fueron menores a $38, esto

reafirma lo que los demás modelos han mostrado. Pueda que sea un modo de operación de falsificadores, utilizar transacciones con montos bajos para no ser detectados ni ser objeto de procedimientos de seguridad en comercios.

E. Análisis de resultados de regresión logística

1. Modelo # 1. Para el primer modelo se incluyen todas las variables descritas al inicio del trabajo, la Figura 62 muestra las variables utilizadas en este modelo, claramente vemos que varias variables no poseen significancia ya que no tienen algún asterisco de significancia como muestra la ilustración. Para el primer modelo la precisión es de 0.9951. Este primer modelo nos abre la ventana para realizar las demás pruebas e ir eliminando las variables que no aportan significancia al modelo.

2. Modelo # 2. Se procede a eliminar variables con poca o nula significancia esto nos lleva a quedarnos con las siguientes variables:

- C87500: Llave primaria de control (Tarjeta o ID Cliente)
- C87506: Hora TRX
- C87507: Fecha TRX
- C87510: MCC
- C87512: Condición Pto Venta
- C87513: ID Adq
- C87567: Tipo Prod TC
- C87547: Cod Moneda Trx
- C87566: Marca o Franquicia
- C87593: Semana del Año
- C87594: Grupo Día

Se depura el modelo, pero el cambio en la precisión es tan pequeña que casi no es notoria se notó que aún siguen habiendo variables que no aportan significancia al modelo por lo que se procederá a seguir depurando al modelo.

3. Modelo # 3 y 4. Se realizaron corridas para los modelos 3 y 4, pero la significancia sigue siendo la misma que la del modelo 2 por lo que incluirlo en el trabajo no aportará valor agregado.

4. Modelo # 5. Se realizó el modelo # 5 los cuales incluían variables como:

- C87506: Hora TRX
- C87507: Fecha TRX

- C87510: MCC
- C87512: Condición Pto Venta
- C87513: ID Adq
- C87547: Cod Moneda Trx
- C87566: Marca o Franquicia
- C87584: ID Cliente
- C87593: Semana del Año
- C87594: Grupo Día
- C87714: Evaluación Dispositivo Chip

Con esta depuración se logró mejorar la precisión del modelo, alcanzando un 0.9952. Esto muestra un gran avance aunque siguen habiendo variables que aportan poca significancia al modelo.

5. Modelo # 6. El modelo # 6 alcanzó la misma precisión que el modelo anterior, como se nota en la Figura 67, aun existen variables que aportan poco al modelo por lo que se realizó un modelo más.

6. Modelo # 7. El modelo # 7 fue el mejor modelo encontrado, las variables que se utilizaron fueron las siguientes:

- C87506: Hora TRX
- C87510: MCC
- C87512: Condición Pto Venta
- C87513: ID Adq
- C87547: Cod Moneda Trx
- C87593: Semana del Año
- C87594: Grupo Día

Como se observa en la Figura 68 se encontró un modelo en donde todas las variables fueron significativas aunque se logró la misma precisión anterior se determina que este es un mejor modelo porque todas las variables aportan significancia y es un modelo mucho más sencillo que los vistos anteriormente.

El modelo refleja una importancia a variables como la hora, esto nos revela muchas de las transacciones fraudulentas se realizan en horas específicas por lo que es una variable a considerar para la empresa.

El mayor hallazgo encontrado durante este trabajo está enmarcado en el MCC. Se reafirma lo que los modelos anteriores mostraron, la variable MCC es uno de los mayores predictores de operaciones

fraudulentas ya que nuevamente está entre los resultados finales. Se debe realizar un análisis de los rangos de MCC mencionados anteriormente para determinar que categorías de negocios caben dentro de estos rangos para tomar medidas de seguridad más drástica y así evitar la propagación de operaciones fraudulentas.

Como hemos visto anteriormente la variable de condición de punto de venta también es importante para la detección de fraude, es importante verificar y garantizar las condiciones óptimas para la realización de las transacciones.

F. Análisis de resultados de redes bayesianas

Al analizar los resultados se determinó que la hipótesis no se alcanzó, debido a que la efectividad obtenida no había alcanzado el 90% requerido, así como la relación de falsos positivos sobre pasaba la relación establecida anteriormente la cual era de 10 a 1.

Es importante notar que en la fase de procesamiento previo de datos, éstos sufrieron una adecuación relevante debido a que los métodos utilizados para disminuir la cantidad de iteraciones (y con ello, el tiempo de procesamiento), poseían perdida de precisión. Esto se evidencia al momento que la data se agrupa de tal forma que 156 mil comercios, se correlacionan a 10 comercios únicos en base al valor esperado de consumo que los clientes tendrían sobre los mismos. Esta operación representa una pérdida de información relevante debido a que ya no se posee una alta precisión en cuanto al identificador de comercio, sino que en su lugar se han consensuado los datos en base a un parámetro en común.

A pesar que la pérdida de información y precisión en los datos era significativa, se obtuvo un 71.63% de efectividad, lo cual representa un valor a tomar en cuenta al hablar de aciertos en las transacciones. Sin embargo, cuando se analizaron las transacciones que se habían evaluado, se encontró que de las 200 mil transacciones, únicamente 59 eran catalogadas como fraude originalmente. Estas transacciones representan un 0.0295% de todos los datos analizados. Desafortunadamente, se encontró que dentro del conjunto de 5.7 millones de transacciones originales, únicamente un aproximado de mil transacciones eran catalogadas como fraude. Esta dispersión mostrada en los datos que se analizaron, pudo repercutir de manera positiva a la efectividad de la red bayesiana, ya que existe la posibilidad que de haber existido una mayor cantidad de transacciones fraudulentas, los resultados hubiesen variado. A pesar de esto, es importante notar que el algoritmo generado para este sistema de clasificación basado en redes bayesianas, fue construido de tal forma que fuera mucho más sencillo y utilizara de mejor forma los recursos para realizar un análisis respectivo en tiempo real. Un análisis más a profundidad, que pudiera desarrollarse en un ámbito de tiempo más grande, se mejoraría significativamente los resultados de este estudio.

Como se planteó anteriormente, los resultados del conjunto de datos utilizado para la fase de aprendizaje de la red bayesiana, habían influido de forma negativa en la efectividad de la red bayesiana, ya que de un total de 59 transacciones originalmente fraudulentas, la red bayesiana construida únicamente acertó 19, lo cual representa un 32.20% del total de transacciones fraudulentas. Esto marca un descenso de 39.43 puntos en la efectividad de la red bayesiana. Es claro que el factor de dispersión de datos afectó tanto el entrenamiento de la red bayesiana, como la evaluación de la misma. Una posible explicación para este resultado es que, las reglas de la red bayesiana se construyeron en su mayoría para detectar transacciones válidas; debido a que estas reglas estaban construidas en base a una probabilidad de ocurrencia representada como un porcentaje, era de esperar que se presentara una mayor cantidad de casos haciendo que la repartición de estos porcentajes fuera más parejo y permitiendo que se desensibilizara la red bayesiana para detectar transacciones fraudulentas que salieran de un determinado comportamiento. En otras palabras, las transacciones eran tantas que la probabilidad que se presentara una transacción fraudulenta de forma única, disminuía de forma inversamente proporcional a la cantidad de transacciones evaluadas después de un determinado punto.

Observando los resultados presentados como desaciertos, se identificó que su gran mayoría se catalogaron como falsos positivos, siendo estos 56,696 transacciones, y únicamente se clasificaron 40 transacciones como falsos negativos.

Al momento de considerar estos dos escenarios en un ámbito real, un falso negativo representa una mayor cantidad de daño a la organización, que un falso positivo. Esto se debe a que la red bayesiana aceptó una transacción que debía ser rechazada ya que se trataba de un fraude, lo cual ocasionaría pérdidas monetarias para la organización o para el usuario final. Por otro lado, un falso positivo, únicamente presentaría un inconveniente al usuario final ya que no realizaría la transacción deseada. Esto se resolvería de forma que el usuario final llamase a su entidad bancaria para liberar la transacción. Este error no incurriría en la pérdida monetaria de ningún tipo.

Al analizar los datos presentados como falsos positivos, se observó que éstos representan un 99.93% de los datos presentados como desaciertos. Establecida anteriormente la importancia de los falsos positivos sobre los falsos negativos, este se marca como un resultado importante. Sin embargo, como anteriormente se encontró que la dispersión de los datos no era la más adecuada, se procedió a realizar un análisis en relación a la representatividad de los falsos negativos y falsos positivos con su máximo posible. Es decir que debido a que únicamente hay 59 transacciones fraudulentas en los datos analizados, la cantidad máxima de falsos positivos que presentaría la red bayesiana es de 59. De igual forma los falsos positivos tendrían un límite máximo de 199,941 (asumiendo que no se acierta ninguna transacción).

Al calcular los porcentajes representativos de esta forma, se determinó que los resultados variaban considerablemente, ya que el porcentaje de falsos negativos presentados era de 67.80% sobre su posible

máximo (40 desaciertos sobre 59 posibles) y el porcentaje de falsos positivos era de 28.36% (56,696 desaciertos sobre 199,941 posibles). Estos datos nos muestran que la efectividad de la red bayesiana, disminuye considerablemente sobre la presentada en un inicio, ya que a pesar que el porcentaje de aciertos se mostraba aceptable, un análisis más profundo nos muestra que la cantidad de falsos negativos presentadas es inaceptable para un modelo que se ejecute en tiempo real con transacciones del día a día.

IX. RESULTADOS DE LA ADMNISTRACIÓN DEL PROYECTO

A. Administración del proyecto

La coordinación y organización del proyecto se realizó aplicando e integrando adecuadamente los 42 procesos de la dirección de proyectos establecidos y aceptados por el Project Management Institute (PMI)9. Estos procesos se agruparon en cinco grupos y cada uno de estos se clasificó en ocho áreas de conocimiento, tal como se puede observar en forma matricial en el siguiente cuadro.

Cuadro 21. Procesos de dirección de proyectos

ÁREAS DE CONOCIMIENTO	INICIACIÓN	PLANIFICACIÓN	EJECUCIÓN	SEGUIMIENTO Y CONTROL	CIERRE
INTEGRACIÓN	1. Acta de constitución / iniciación del proyecto	1. Desarrollar el plan	1.Dirigir y gestionar	1. Monitorear y controlar 2. Control integrado	Cerrar el proyecto
ALCANCE		2. Recopilar requisitos 3. Definir el alcance 4. Crear EDT		3. Verificar alcance 4. Controlar alcance	
TIEMPO		5. Definir actividades 6. Secuenciar actividades 7. Estimar recursos 8. Estimar duración 9. Desarrollar cronograma		5. Controlar el cronograma	
COSTOS		10. Estimar costos 11. Determinar presupuesto		6. Controlar los costos	
CALIDAD		12. Planificar calidad	2. Asegura-miento de calidad	7. Controlar la calidad	

Continuación Cuadro 21. Procesos de dirección de proyectos

ÁREAS DE CONOCIMIENTO	INICIACIÓN	PLANIFICACIÓN	EJECUCIÓN	SEGUIMIENTO Y CONTROL	CIERRE
RECURSO HUMANO		13. Desarrollar el plan de Recurso Humano	3. Adquirir el equipo / definir el equipo 4. Desarrollarlo 5. Gestionarlo		
COMUNICACIÓN	2. Identificar a los involucrados	14. Planificar la comunicación	6. Distribuir la información 7. Gestionar las expectativas	8. Informar el desempeño	
RIESGOS		15. Planificar 16. Identificar 17. Análisis cualitativo 18. Análisis cuantitativo 19. Respuesta a los riesgos		9. Monitorear y controlar	
ADQUISICIONES		20. Planear adquisiciones	8. Efectuar	10.Administrar	2. Cerrar

Fuente: Elaboración propia

La metodología utilizada en el proyecto consta de cinco fases (aprobación, definición, planificación, ejecución y cierre), la cual se adaptó a los cinco grupos de procesos establecidos por el Project Management Institute (PMI) para la correcta gestión de cualquier proyecto. En la siguiente tabla se hace una analogía entre las cinco fases de la metodología utilizada y los cinco grupos de procesos del PMI.

Cuadro 22. Metodología adaptada a los grupos de procesos del PMI

METODOLOGÍA		GRUPO DE PROCESOS
1. Aprobación	→	1. Iniciación
2. Definición	→	2. Planificación
3. Planificación		
4. Ejecución	→	3. Ejecución 4. Seguimiento y control
5. Cierre	→	5. Cierre

Fuente: Elaboración propia

1. Iniciación

a. Integración: Acta de iniciación del proyecto. El Megaproyecto dio inicio en julio de 2013. El nombre inicial fue:

"Sistemas inteligentes para reconocimiento de Patrones de comportamiento transaccionales. Se utilizarán en sistemas de detección de operaciones fraudulentas en ambientes financieros"

Se creó un acta de constitución para poder obtener la aprobación de la Universidad y dar inicio al megaproyecto (Ver Anexos: Acta de Constitución).

En este documento se especificó el nombre del proyecto, el objetivo general que fue: "Crear sistemas inteligentes, capaces de reconocer patrones de comportamiento en sistemas transaccionales, segmentación por patrones y sugerir reglas de detección de patrones específicos" y los objetivos específicos. También se incluyó una descripción general del megaproyecto, el equipo sugerido, los beneficios que se obtendrían al formar parte del proyecto, el departamento que coordinaría el megaproyecto (Departamento en Ciencias de la Computación) y la información de contacto en representación de la empresa.

Este documento fue la base para el inicio del megaproyecto y fue el punto de partida para definir con mayor detalle posteriormente el proyecto y determinar los puntos claves de investigación para lograr el objetivo del mismo y así satisfacer las necesidades y expectativas de los interesados.

b. Comunicación: Identificar a los involucrados. Se realizó una matriz de análisis de los involucrados, donde se listaron todas las partes interesadas en el proyecto (directa e indirectamente) que recibirían un impacto por la realización del proyecto, descripción de sus intereses y el nivel de interacción que tendrían a lo largo del desarrollo del proyecto. La evaluación del impacto se clasificó en base al peligro que representara para los objetivos internos de la gestión de proyectos (costo, alcance, tiempo, calidad), los cuales son necesarios para alcanzar el éxito de un proyecto. El impacto se consideró como ALTO si la mala gestión del interesados afectaba 3 o más de los objetivos internos, medio si afectaba 2 objetivos internos y BAJO si afectaba 1 objetivo interno.

Cuadro 23. Matriz de análisis de los involucrados

Interesado	Inquietud(es) del interesado en el proyecto	Evaluación del impacto	Estrategias potenciales para obtener apoyo o reducir obstáculos
A. Cliente de la modalidad de megaproyecto	Que el proyecto logre proponer un método con mejor desempeño que el sistema actual	ALTO Afecta los cuatro objetivos internos	Definir claramente el alcance del proyecto y las metodologías a utilizar y mantener una comunicación constante y actualiza sobre los avances y resultados de las diferentes fases del proyecto

Continuación Cuadro 23. Matriz de análisis de los involucrados

Interesado	Inquietud(es) del interesado en el proyecto	Evaluación del impacto	Estrategias potenciales para obtener apoyo o reducir obstáculos
B. Universidad del Valle de Guatemala	Lograr suplir las necesidades del cliente de la modalidad de megaproyecto y obtener resultados que cumplan con sus necesidades	ALTO Afecta los cuatro objetivos internos	Definir claramente el alcance del proyecto, fases y calendarización de las mismas y mantener una comunicación constante y actualizada con los encargados del proyecto sobre los avances del mismo
C. Instituciones financieras (bancos) clientes de la empresa con la que se trabajó	Lograr mantener y aumentar los niveles de satisfacción al cliente (usuarios de tarjetas de crédito y débito)	BAJO Afecta el alcance del proyecto	Informarse sobre la situación actual, y los parámetros que buscan mejorar para aumentar la satisfacción del cliente
D. Usuarios de tarjetas de crédito y débito	Tener la confianza de utilizar sus tarjetas de crédito y/o débito sin tener que preocuparse por posibles fraudes o cancelación su tarjeta por error.	BAJO Afecta el alcance del proyecto	Informase sobre las necesidades de los clientes y el tipo de problemática actual para poder enfocar el proyecto en mejorar la satisfacción del cliente final
E. Comercios con opción a pago con tarjetas de crédito y débito	Aumentar la confianza en el cliente al momento de realizar sus compras con tarjetas de crédito y/o débito	BAJO Afecta el alcance del proyecto	Informarse sobre la situación actual y tomar en cuenta los parámetros que ayudarían a mejorar la situación actual

Fuente: Elaboración propia

Luego de haber identificado a los involucrados y la forma obtener el apoyo se realizó una Matriz de poder / Interés de los involucrados, donde se clasificó a la empresa y a la Universidad del Valle de Guatemala como involucrados con alto poder y alto interés debido a que son los únicos que pueden cambiar directamente algún factor del proyecto, caso contrario de las instituciones financieras, usuario de tarjetas de crédito y/o débito y comercio, aunque su interés es alto y se tomaron en cuenta sus necesidades para el

proyecto, estos no tuvieron una relación directa con el proyecto, por lo que no podían realizar cambios en la estructura del mismo.

Cuadro 24. Matriz de poder / Interés de los involucrados.

P O D E R	**ALTO**		A,B
	BAJO		C, D, E
		BAJO	**ALTO**
		INTERÉS	

Fuente: Elaboración propia

Se determinó que a los interesados A y B (La empresa y UVG respectivamente) se les debe gestionar atentamente y a los interesados C, D, E se deben monitorear.

2. Planificación

a. Integración: Desarrollar el plan. Se desarrolló el plan para la dirección del proyecto el cual contiene las siguientes acciones necesarias para definir, preparar, integrar y coordinar el proyecto:

1) Entregables del proyecto:

- Los meses de julio a noviembre de 2013 se definieron como la etapa de investigación a fin de seleccionar y definir los algoritmos y métodos que se estarán desarrollando en las fases de ejecución por lo que el entregable al final de esta etapa será el "Reporte de información general" (ver Anexos: Reporte de información general). Los respectivos cuatro reportes (uno por cada algoritmo o método a desarrollar) se pueden encontrar en la sección de RESULTADOS. En este documento se definen los algoritmos o métodos que se va a desarrollar (ya que estos fueron elegidos de mutuo acuerdo entre el grupo y la empresa), las razones de su elección y el lenguaje de programación que se va a estar utilizando.
- Los meses de enero a septiembre de 2014 se definieron como la etapa de ejecución y cierre del proyecto. En este caso se definieron entregables por cada **sprint**[12]realizado, a los cuales se les definió un mes de duración aproximado para cada uno. Al finalizar cada sprint cada uno de los desarrolladores entregaba el "Reporte de fase terminada" (ver Anexos: Reporte de fase terminada). Los respectivos juegos de reporte de cada uno de los cuatro

algoritmos y métodos desarrollados (uno por cada algoritmo o método a desarrollar) se pueden encontrar en la sección de RESULTADOS.

2) Reuniones con los miembros del equipo asignado al proyecto:

- Se definieron en todas las etapas del proyecto reuniones presenciales semanales de una hora y media de duración en la Universidad del Valle de Guatemala para revisión de avances y definición de las siguientes acciones a realizar.
- Comunicación adicional de baja prioridad o importancia se puede comunicar por medio de correo electrónico a el(los) interesado(s).
- Reuniones extraordinarias fueron coordinadas de manera virtual por medio de Google Hangout.

3) Gestión y utilización de los recursos:

- El Recurso Humano (conocimiento y tiempo) de los involucrados en el proyecto es el principal recurso que se requirió para este proyecto. Por esa razón se distribuyó el trabajo total en intervalos de tiempo con una carga de trabajo moderada en cada etapa.

- Los recursos materiales requeridos para el desarrollo era equipo de cómputo, el cual en la mayoría de las ocasiones se podían utilizar las computadoras personales de cada uno de los miembros del equipo. En los casos cuando se requirió mayor capacidad de procesamiento se hizo uso de las computadoras y servidores de la Universidad del Valle de Guatemala.

4) Implementación de los métodos y normas planificadas:

- Cada entregable del proyecto se documentó en Asana, la cual fue la plataforma virtual para gestión de proyectos seleccionada para llevar el control del proyecto y verificación que cada etapa se desarrollara dentro del tiempo definido.

5) Establecimiento y gestión de los canales de comunicación del proyecto:

- El principal canal de comunicación para los entregables del proyecto fue Asana, ahí se plasmaron las calendarizaciones independientes de cada uno de los desarrolladores y se guardaron las copias de los entregables al final de cada etapa.

- Se tenía comunicación directa con todo el equipo en las reuniones semanales y cualquier otro tipo de comunicación adicional y/o extraordinaria se realizó por correo o por reuniones virtuales en Google Hangout.

5) Generación de los datos del proyecto:

- Se definió una entrega final al terminar la etapa de investigación y definición de algoritmos y métodos y luego se definieron entregas mensuales al finalizar cada sprint en las etapas de ejecución.
- Se manejó en Asana un Gantt general del Megaproyecto y 4 Gantt adicionales para personalizar con los sprints de cada uno de los desarrolladores.

6) Solicitudes de cambios y adaptación de los cambios aprobados:

- Como se manejó una gestión ágil en el proyecto, cuando surgían solicitudes de cambios en los algoritmos o métodos, ya fueran solicitudes por parte de los desarrolladores o por parte del cliente de la modalidad de megaproyecto, estas eran analizadas inmediatamente y si era factible la realización del cambio este se documentaba en el reporte de la fase y se adaptaba lo más pronto posible. El tiempo exacto dependía del tipo de cambio solicitado, la complejidad de este y la fase específica de desarrollo en la que se encontrara el algoritmo o método.

7) Gestión de riesgos:

- Los riesgos se identificaron y gestionaron con base en matrices de manejo de riesgos, las cuales se describen en la sección de riesgos de este capítulo.

8) Gestión con el cliente:

- Se le dio acceso al cliente de la modalidad de megaproyecto al proyecto en Asana de modo que este pudiera estar enterado de los avances en cada uno de los algoritmos y métodos que se estaban desarrollando.
- Adicionalmente se mantenía comunicación constante por medio de correo electrónico para notificar avances y/o cambios importantes realizados.
- Cuando se habían alcanzado avances significativos se realizaban reuniones virtuales con el cliente y todos los miembros del equipo por medio de Webbex o reuniones en las instalaciones del cliente.

9) Recopilación y documentación de las lecciones aprendidas para próximos proyectos

- Cada cambio significativo y/o fallo que sería útil conocer para futuros proyectos de seguimiento del presente proyecto o proyectos similares fueron documentados por el desarrollador que encontró el fallo y se incluyeron en el reporte final consolidado del megaproyecto.

3. Alcance: Recopilar requisitos. Los requisitos definidos por el cliente para el proyecto fueron los siguientes:

Mejorar las siguientes tres métricas:

- **Porcentaje de efectividad** (asertividad en la clasificación de cada transacción, ya sea en la categoría de no fraude o de fraude).
- Porcentaje de desaciertos (transacciones identificadas en la categoría incorrecta).
- Porcentaje de **Falsos positivos**.

En los casos anteriores, se busca un Falso Positivo de 10% o menos y un 90% de **porcentaje de asertividad** en la detección correcta de las transacciones (efectividad). Si el algoritmo genera alertas al procesar las transacciones, este porcentaje de generación de Alertas no debe superar el 1% de transacciones alertadas.

No existen requisitos relativos al lenguaje de programación a utilizar, a fin de no limitar el desempeño o utilización de algoritmos debido a esto.

4. Alcance: Definir el alcance. Se realizó una investigación exhaustiva, se construyó un modelo y se ejecutaron pruebas de desempeño en base a los parámetros de requisitos para los siguientes algoritmos:

- Redes neurales
- Support Vector Machines (SVM)
- Redes Bayesinas

Adicionalmente, se realizó un informe de Clustering para la detección de patrones de comportamiento.

Para estos algoritmos y métodos se entregó toda la documentación relativa al funcionamiento de los modelos y el código relativo al mismo para que el cliente pueda replicar los modelos en su ambiente de datos para realizar pruebas e implementar a futuro el algoritmo que se sugirió como óptimo (Redes Neurales). El proyecto no abarca la implementación de ninguno de los algoritmos en los sistemas del

cliente, razón por la cual no existió ningún tipo de restricción relativa al lenguaje de programación a utilizar y la compatibilidad de este con el sistema actual.

a. Alcance: Crear EDT (Estructura de Desglose del Trabajo). La fase de desarrollo de los prototipos de los tres algoritmos y análisis de clustering se gestionó en base a la metodología ágil de proyectos informáticos, es decir se definieron fases relativamente cortas llamadas Sprints. Los Sprints tienen por lo general una duración entre dos semanas hasta dos meses. Se definieron Sprints de un mes de duración aproximada para cada algoritmo y método desarrollado. Se definieron nueve Sprints en cada caso, los cuales por lo general tenían una duración de un mes cada uno, pero este podía tener una duración menor o mayor según el objetivo de cada fase.

b. Tiempo: Definir actividades. Se definieron nueve fases para llegar al prototipo final de cada uno de los algoritmos y métodos desarrollados. Para cada caso específico se definieron fechas y duración de cada fase apropiada para el algoritmo y método. Las fechas específicas y los objetivos de cada fase en cada caso se encuentran documentadas en la sección de resultados.

c. Tiempo: Secuenciar actividades. El paso inicial fue la selección de los algoritmos o métodos sobre los cuales se basaría el megaproyecto. Una vez definidos se procedió a realizar los prototipos de cada uno de los algoritmos y métodos. Cada uno de estos se podía realizar de forma paralela ya que no dependían uno del otro. Al haber concluido los prototipos finales de los algoritmos estos se compararon bajo los criterios de desempeño definidos con el cliente con anterioridad y en base al que obtuvo el desempeño global mejor, Redes neurales, se procedió a realizar el análisis financiero por escenarios necesario para la conclusión del proyecto.

d. Tiempo: Estimar recursos. Se estimaron como recursos necesarios solo recursos humanos, no se estimó el equipo de cómputo o materiales adicionales que se pudieran requerir para el desarrollo de cada actividad debido a que cada persona puso un precio al servicio brindado, el cual además del tiempo invertido debía incluir los recursos y herramientas utilizadas para desarrollar el producto.

Se desarrolló el siguiente cuadro para la estimación de los recursos:

Cuadro 25. Estimación de los recursos

Actividad a realizar	Cantidad de personas requeridas	Persona(s) designadas
Administración del proyecto y creación de análisis financiero por escenarios	1 estudiante de Ingeniería Industrial	Ana Lucía Paiz

Continuación Cuadro 25. Estimación de los recursos

Actividad a realizar	Cantidad de personas requeridas	Persona(s) designadas
Prototipo utilizando Redes neurales	1 estudiante de Ingeniería en Ciencia de la Computación	Joel Cantoral
Prototipo utilizando Support Vector Machines	1 estudiante de Ingeniería en Ciencia de la Computación	Diego Enríquez
Prototipo utilizando Redes Bayesianas	1 estudiante de Ingeniería en Ciencia de la Computación	Melinton Navas
Análisis de Clustering para identificar patrones de comportamiento	1 estudiante de Ingeniería en Ciencia de la Administración	Berny Ixcayau

Fuente: Elaboración propia

e. Tiempo: Estimar duración. Se desarrolló la siguiente tabla para la estimación de la duración durante el tiempo disponible para el proyecto el cual inició en julio de 2013 y concluyó en octubre de 2014:

Cuadro 26. Listado de actividades y duración de cada periodo

Actividad a realizar	Periodos necesarios	Duración de cada periodo
Administración del proyecto y creación de análisis financiero por escenarios	1 periodo de gestión del proyecto y análisis financiero por escenarios	16 meses
Actividad a realizar	**Periodos necesarios**	**Duración de cada periodo**
Prototipo utilizando Redes neurales	1 periodo definición del algoritmo	6 meses
Prototipo utilizando Support Vector Machines **Prototipo utilizando Redes Bayesianas**	9 periodos de ejecución de prototipos	9 meses
Análisis de Clustering para identificar patrones de comportamiento	1 periodo de cierre del proyecto	1 mes

Fuente: Elaboración propia

Este cuadro muestra un estimado general de periodos y duración de cada periodo para las actividades que se realizaron. En la sección de cierre se muestra el detalle de los periodos y la duración de cada periodo

dependiendo del algoritmo o modelo. No se puede estandarizar el número de periodos y duración en todos los algoritmos y modelos debido a que cada algoritmo y modelo se comporta de manera diferente y tiene diferente estructura. Definir estándares hubiera disminuido la calidad de los resultados finales y limitado el alcance final. Sin embargo si se definió como estándar la fecha límite de finalización de la ejecución (1 de octubre de 2014) y se puso como restricción que ningún periodo podía exceder de 2 meses de duración.

f. Tiempo: Desarrollar cronograma. Consolidado en las fases generales el cronograma general, es el siguiente:

Cuadro 27. Cronograma de actividades de julio 2013 a noviembre 2013

DESCRIPCIÓN	julio-2013					agosto-2013				septiembre-2013				octubre-2013					noviembre-2013			
Fase 1: Aprobación del proyecto y coordinación de reuniones con la empresa																						
Fase 2: Definición de los enfoques a utilizar para brindar una solución a la problemática actual																						
Fase 3: Planificación del desarrollo de prototipos necesarios y las fases requeridas en el proyecto																						

Fuente: Elaboración propia

Cuadro 28. Cronograma de actividades de enero 2014 a mayo 2014

DESCRIPCIÓN	enero-2014	febrero-2014	marzo-2014	abril-2014	mayo-2014
Fase 4: Ejecución de los prototipos y análisis para obtener el producto final					

Fuente: Elaboración propia

Cuadro 29. Cronograma de actividades de julio 2014 a noviembre 2014

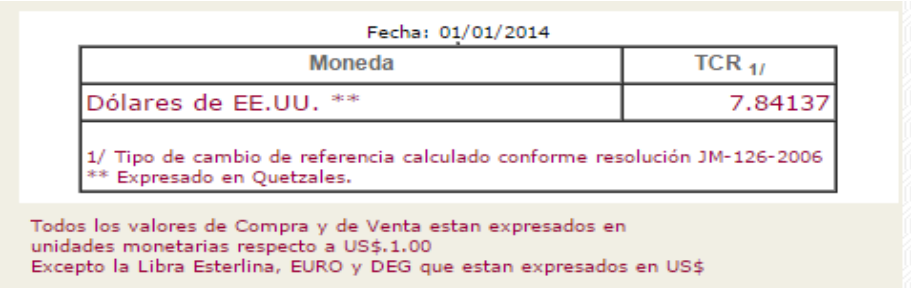
DESCRIPCION	Julio-2014	Agosto-2014	Septiembre-2014	Octubre-2014	Noviembre-2014
Fase 4: Ejecución de los prototipos y análisis para obtener el producto final					
Fase 5: Cierre del proyecto con el producto final consolidado y la propuesta del modelo de negocio en base a escenarios					

Fuente: Elaboración propia

En el Anexo se pueden encontrar el Gantt de la fase de ejecución del proyecto, detallado por las tareas de todos los desarrolladores y un Gantt individual para cada desarrollador, tomando como línea guía este cronograma.

g. Costos: Estimar costos. Para las conversiones entre dólares y quetzales se tomó como referencia el tipo de cambio de enero de 2014 (fecha en la que se inició la fase de ejecución del proyecto) según el Banco de Guatemala, el cual era de $1 = Q.7.84

Figura 70. Tipo de cambio de (Dólares de EE.UU. a Quetzales) enero de 2014



Fecha: 01/01/2014

Moneda	TCR 1/
Dólares de EE.UU. **	7.84137

1/ Tipo de cambio de referencia calculado conforme resolución JM-126-2006
** Expresado en Quetzales.

Todos los valores de Compra y de Venta estan expresados en unidades monetarias respecto a US$.1.00
Excepto la Libra Esterlina, EURO y DEG que estan expresados en US$

Fuente: (Banco de Guatemala. 2014).

En promedio se calcularon nueve fases por persona. El presupuesto asignado por el cliente para cada miembro del equipo fue de Q. 27,440.00 ($3,500.00). Inicialmente se definieron un promedio de nueve fases por algoritmo o modelo desarrollado, por lo que en ese caso cada fase tendría un presupuesto asignado de Q. 3,048.90 el cual podía ser mayor o menor dependiendo de la duración de la fase y las tareas

realizadas durante la misma. Cada miembro del equipo sabía desde el inicio el presupuesto asignado para lograr el producto final, entonces cada persona debía costear las fases de su algoritmo en base al tiempo y recursos que requería cada fase específica, tomando en cuenta que el costo total (sumatoria de todas las fases) no debía superar los Q. 27,440.00

Periódicamente se llevaba un control del costo de cada fase para verificar que ningún desarrollador sobrepasara el presupuesto asignado.

El presupuesto asignado por el cliente no fue considerado directamente como un pago por el producto final, sino como una "beca" brindada a los miembros del equipo para apoyar el desarrollo académico de los miembros del equipo. Debido a esto se asignó un presupuesto equitativo a cada miembro del equipo, independientemente del rol específico de cada persona en el proyecto.

En los costeos específicos por fase y el costeo final de cada algoritmo o modelo desarrollado se buscó no exceder el presupuesto asignado y lograr que los costos fueran lo más cercano al presupuesto como fuera posible.

En la sección de cierre se detalla el costo total de cada algoritmo desarrollado, desglosado por el costo de cada fase. En el caso del análisis del Clustering se definió solamente el costo final, debido a que un análisis incompleto no tenía utilidad, el beneficio se obtiene del análisis completo. A diferencia de los algoritmos, donde se fueron creando varías versiones antes de la versión final, pero desde la versión 1 se obtuvo un producto funcional. En estos casos la versión final era la versión optimizada (la que obtenía el mejor desempeño).

h. Costos: Determinar presupuesto. Se cuenta con un presupuesto total de Q. 27,440.00 ($3,500.00) por persona integrante del equipo durante todo el proyecto. Es decir, un presupuesto total de Q.137,200.00 ($17,500.00). Se distribuyó el presupuesto equitativamente entre todos los integrantes del equipo, debido a que el cliente lo definió de esa manera, al clasificar el presupuesto asignado como una "beca de estudio" para los integrantes del megaproyecto.

i. Calidad: Planificar calidad. Se definió como un producto de calidad el que cumpliera con los requisitos mínimos de rendimiento definidos por el cliente, los cuales básicamente son lograr una asertividad mayor al 90% y una proporción de falsos positivos del 10% o menos.

La calidad se fue revisando al final de cada sprint, y los objetivos del siguiente sprint iban enfocados al mejoramiento de la calidad, hasta alcanzar la calidad deseada. (Ver Anexos con los resultados de cada fase).

j. Recursos Humanos: Desarrollar el plan de Recursos Humanos. Se describió por medio de la siguiente tabla los roles y responsabilidades del grupo.

Cuadro 30. Definición de roles y responsabilidades del equipo

Rol	Persona encargada	Responsabilidades
Directora de proyecto	Ana Lucía Paiz	Gestión del proyecto para asegurar el éxito del mismo
Desarrollador "Redes neurales"	Joel Cantoral	Desarrollo en tiempo y con la calidad esperada del modelo utilizando "Redes Neurales"
Desarrollador "Support Vector Machines"	Diego Enríquez	Desarrollo en tiempo y con la calidad esperada del modelo utilizando "Support Vector Machines"
Desarrollador "Redes Bayesianas"	Melinton Navas	Desarrollo en tiempo y con la calidad esperada de modelo utilizando "Redes Bayesianas"
Desarrollador de análisis de detección de patrones de comportamiento transaccionales por medio de Clustering	Berny Ixcayau	Desarrollo en tiempo y con la calidad esperada de análisis de detección de patrones de comportamiento transaccionales por medio de Clustering

Fuente: Elaboración propia

En el siguiente cuadro, se describen las habilidades que cada rol requiere para poder alcanzar el éxito del proyecto y las relaciones de comunicación que cada rol debe tener.

Cuadro 31. Habilidades requeridas en los roles y relaciones de comunicación

Rol	Habilidades requeridas	Relaciones de comunicación
Directora de proyecto	Conocimientos avanzados de dirección de proyectos informáticos	Comunicación directa y constante con el cliente de la modalidad de megaproyecto para gestionar las expectativas y comunicación directa y constante con todo el equipo para garantizar entregas en tiempo y con la calidad esperada
Desarrollador "Redes neurales"	Conocimientos avanzados de programación y entrenamiento de algoritmos inteligentes	Comunicación directa y constante con la directora de proyectos para verificar que se esté cumpliendo con la calidad en el tiempo que se debe y el cliente esté satisfecho, además de comunicación con el resto del grupo para apoyo mutuo en resolución de problemas y mejores técnicas de ejecución

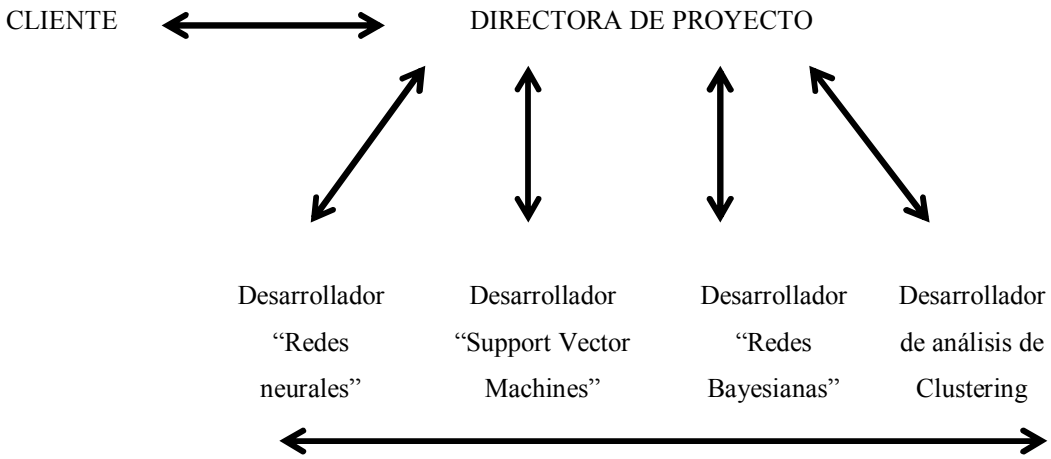
Continuación Cuadro 31. Habilidades requeridas en los roles y relaciones de comunicación

Rol	Habilidades requeridas	Relaciones de comunicación
Desarrollador "Support Vector Machines"	Conocimientos avanzados de programación y entrenamiento de algoritmos inteligentes	Comunicación directa y constante con la directora de proyectos para verificar que se esté cumpliendo con la calidad en el tiempo que se debe y el cliente esté satisfecho, además de comunicación con el resto del grupo para apoyo mutuo en resolución de problemas y mejores técnicas de ejecución
Desarrollador "Redes Bayesianas"	Conocimientos avanzados de programación y entrenamiento de algoritmos inteligentes	Comunicación directa y constante con la directora de proyectos para verificar que se esté cumpliendo con la calidad en el tiempo que se debe y el cliente esté satisfecho, además de comunicación con el resto del grupo para apoyo mutuo en resolución de problemas y mejores técnicas de ejecución
Desarrollador de análisis de detección de patrones de comportamiento transaccionales por medio de Clustering	Conocimientos avanzados de programación y análisis con técnicas de minería de datos	Comunicación directa y constante con la directora de proyectos para verificar que se esté cumpliendo con la calidad en el tiempo que se debe y el cliente esté satisfecho, además de comunicación con el resto del grupo para apoyo mutuo en resolución de problemas y mejores técnicas de ejecución

Fuente: Elaboración propia

k. Comunicación: Planificar la comunicación

Figura 71. Red de comunicaciones del proyecto



Fuente: Elaboración propia

En el siguiente cuadro se describe el plan de comunicación basado en el diagrama anterior de flujo de comunicación.

Cuadro 32. Plan de comunicación

Rol	Plan de comunicación
Directora de proyecto	• Comunicación directa con el cliente de la modalidad de megaproyecto con la frecuencia que lo requiere la fase del proyecto • Comunicación directa con todo el equipo una vez por semana y comunicación virtual según lo requiera cada fase del proyecto
Desarrollador "Redes neurales"	• Comunicación directa con la directora de proyecto una vez por semana y comunicación virtual según lo requiera cada fase del proyecto. • Comunicación directa con todo el equipo una vez por semana y comunicación virtual según lo requiera cada fase del proyecto
Desarrollador "Support Vector Machines"	• Comunicación directa con la directora de proyecto una vez por semana y comunicación virtual según lo requiera cada fase del proyecto. • Comunicación directa con todo el equipo una vez por semana y comunicación virtual según lo requiera cada fase del proyecto
Desarrollador "Redes Bayesianas"	• Comunicación directa con la directora de proyecto una vez por semana y comunicación virtual según lo requiera cada fase del proyecto. • Comunicación directa con todo el equipo una vez por semana y comunicación virtual según lo requiera cada fase del proyecto

Continuación Cuadro 32. Plan de comunicación

Rol	Plan de comunicación
Desarrollador de análisis de detección de patrones de comportamiento transaccionales por medio de Clustering	• Comunicación directa con la directora de proyecto una vez por semana y comunicación virtual según lo requiera cada fase del proyecto. • Comunicación directa con todo el equipo una vez por semana y comunicación virtual según lo requiera cada fase del proyecto

Fuente: Elaboración propia

l. Riesgos: Planificar. Para la planificación de riesgos se determinó realizar un análisis en base a las amenazas que se pudieran encontrar en los siguientes objetivos internos de la gestión del proyecto:

- Costo
- Tiempo
- Alcance
- Calidad

Considerado como riesgo cualquier situación que potencialmente pudiera afectar el resultado final del proyecto.

Se estableció un plan de gestión de riesgos de seis pasos:

1) Identificar todos los posibles riesgos
2) Identificar cuál de los cuatro objetivos (tiempo, calidad, alcance, costo) se vería afectado si ocurriera el riesgo identificado.
3) Ponderar el riesgo en base a su impacto y probabilidad de ocurrencia.
4) Ubicar el riesgo en una matriz y clasificarlo por código de color para darle prioridad visual.
5) Planificar la respuesta específica los riesgos.
6) Analizar otros riesgos.

m. Riesgos: Identificar

Se identificaron los siguientes riesgos:

Cuadro 33. Identificación de riesgos respecto a los objetivos internos de gestión del proyecto

Riesgo	Objetivo al que afecta
1. No terminar el proyecto en tiempo debido a atrasos por parte del equipo.	• Costo • Tiempo • Alcance • Calidad
2..No terminar el proyecto en tiempo debido a atrasos por parte del cliente de la modalidad de megaproyecto.	• Costo • Tiempo • Alcance • Calidad
3..No contar con datos de prueba adecuados para entrenar los modelos.	• Costo • Calidad • Tiempo • Alcance
4..No encontrar un algoritmo que cumpla con todas las especificaciones dadas por el cliente.	• Calidad

Fuente: Fuente propia

n. Riesgos: Análisis cualitativo. Por medio de una matriz de riesgos se clasificó el impacto y la probabilidad de ocurrencia de cada riesgo identificado. Como no hay disponible información histórica de megaproyectos similares, se ponderaron los riesgos con una escala cualitativa (ALTO, MEDIO, BAJO). Adicionalmente se utilizaron códigos de colores (rojo, amarillo y verde) como identificadores visuales de las prioridades de los riesgos.

Cuadro 34. Matriz de impacto y probabilidad de ocurrencia de los riesgos

RIESGO	IMPACTO			PROBABILIDAD DE OCURRENCIA		
	ALTO	MEDIO	BAJO	ALTO	MEDIO	BAJO
No terminar el proyecto en tiempo debido a atrasos por parte del equipo.						
No terminar el proyecto en tiempo debido a atrasos por parte del cliente de la modalidad de megaproyecto						

Continuación Cuadro 34. Matriz de impacto y probabilidad de ocurrencia de los riesgos						
RIESGO	**IMPACTO**			**PROBABILIDAD DE OCURRENCIA**		
	ALTO	**MEDIO**	**BAJO**	**ALTO**	**MEDIO**	**BAJO**
No contar con datos de prueba adecuados para entrenar los modelos						
No encontrar un algoritmo que cumpla con todas las especificaciones dadas por el cliente.						

Fuente: Elaboración propia

ñ. Riesgos: Análisis cuantitativo. Debido a que no se tiene información histórica cuantitativa, no se proporcionan datos numéricos de impacto y la probabilidad de ocurrencia de los riesgos. Sin embargo se creó una ponderación en base a los cuatro objetivos que se deben tomar en cuenta en todo proyecto (tiempo, calidad, alcance, costo). Cada uno de estos recibe una ponderación equitativa del 25%. Se definió una ponderación equitativa debido a que todos los objetivos internos son de igual importancia para lograr los resultados esperados. No se definió ninguna escala de prioridades ni por parte del cliente ni por parte del equipo. Por cada objetivo de estos que se pueda ver afectado por la ocurrencia del riesgo, se sumará 25% al impacto numérico del proyecto.

Cuadro 35. Ponderación de impacto de los riesgos identificados

Riesgo	Objetivo al que afecta	Ponderación de impacto
1. No terminar el proyecto en tiempo debido a atrasos por parte del equipo.	• Costo • Tiempo • Alcance • Calidad	100%
2. No terminar el proyecto en tiempo debido a atrasos por parte del cliente de la modalidad de megaproyecto.	• Costo • Tiempo • Alcance • Calidad	100%
3. No contar con datos de prueba adecuados para entrenar los modelos.	• Costo • Calidad • Tiempo • Alcance	100%

Continuación Cuadro 35. Ponderación de impacto de los riesgos identificados

Riesgo	Objetivo al que afecta	Ponderación de impacto
4. No encontrar un algoritmo que cumpla con todas las especificaciones dadas por el cliente.	• Calidad	25%

Fuente: Elaboración propia

o. Riesgos: Respuesta a los riesgos.

Cuadro 36. Plan de respuesta a los riesgos I

Riesgo	Respuesta al riesgo
1. No terminar el proyecto en tiempo debido a atrasos por parte del equipo	Coordinar por medio de una metodología ágil el proyecto de modo que se pueda organizar por medio de sprints que en promedio tengan una duración de un mes, de modo que cualquier problema no genere un retraso mayor al tiempo del sprint. Un sprint de retraso es manejable debido a que se puede llegar a nivelar este tiempo de retraso de modo que tenga un efecto de 0 retraso en la entrega final. Esto es especialmente importante en el caso de este proyecto, ya que cumple una función como trabajo de graduación, entonces se debe regir también a las fechas de entrega finales establecidas por la Universidad del Valle de Guatemala, por lo que un retraso en la fecha de finalización aunque sea mínimo podría significar el fracaso del proyecto.
2. No terminar el proyecto en tiempo debido a atrasos por parte del cliente de la modalidad de megaproyecto.	Coordinar por medio de una metodología ágil el proyecto de modo que se pueda organizar por medio de sprints que en promedio tengan una duración de 1 mes, de modo que cualquier problema no genere un retraso mayor al tiempo del sprint y se puede identificar al menos con un sprint de anticipación la información que se requiere del cliente. Adicionalmente dependiendo de la importancia o dificultad de conseguir los datos, como directora de proyecto, se gestionó la comunicación de manera de asegurar obtener la información con el menor tiempo posible.

Cuadro 37. Plan de respuesta a los riesgos II

Riesgo	Respuesta al riesgo
1. No contar con datos de prueba adecuados para entrenar los modelos.	En este caso desde el inicio se sabía que se iban a requerir datos de prueba para poder entrenar los algoritmos, sin embargo no se sabía si el formato en que el cliente fuera a brindar los datos iba a ser el idóneo y contener la información específica que necesitaran lo algoritmos. La manera de gestionar este riesgo fue brindar las especificaciones específicas de lo que se necesitaba al cliente desde el momento en que se conocieran los requerimientos. Adicionalmente se creó un plan de respaldo, el cual consistió en determinar un tiempo específico límite para poder esperar a que el cliente brindara los datos de entrenamiento requeridos (2 meses antes de la finalización). Si se llegaba al límite de ese tiempo sin los datos se procedería a utilizar datos genéricos o el equipo iba a generar los datos de prueba, aunque esto significara una disminución en la calidad del producto, al menos no se estarían afectando el resto de los objetivos.
2. No encontrar un algoritmo que cumpla con todas las especificaciones dadas por el cliente.	Este riesgo se identificó desde el inicio y se controló desde el inicio, previo a elegir los algoritmos a desarrollar. Se le dedicó gran parte del tiempo a la investigación inicial para determinar qué algoritmos tendrían el potencial de cumplir con los requisitos del cliente. De esta manera se disminuye la probabilidad de ocurrencia del riesgo, hasta un punto donde es manejable y vale la pena realizar el proyecto.

Fuente: Elaboración propia

p. Adquisiciones: Planear

- No se planearon adquisiciones materiales para el desarrollo del proyecto debido a que cualquier equipo o herramienta requerida para el cumplimiento satisfactorio del rol asignado fue tomado en cuenta por cada miembro del equipo dentro del precio asignado a cada fase.

5. Ejecución

a. Integración: Dirigir y gestionar. Para dirigir el proyecto se utilizó un Gantt para llevar el control de todas las fases.Se creó un Gantt general para visualizar todas las tareas del proyecto y un Gantt por desarrollador, es decir, 1 Gantt general y 4 Gantt personalizados.

De esta manera se llevaba un control sobre el avance del proyecto. Al final de cada fase o sprint cada desarrollador documentaba las acciones y avance alcanzado en cada fase y se verificaba que se hubiera

alcanzado el objetivo de la fase. En caso no se hubiera alcanzado el objetivo de la fase se documentaba la razón y se proponían acciones correctivas en la siguiente fase para no afectar el desempeño del proyecto.

b. Calidad: Aseguramiento de calidad. En la fase de ejecución se definieron sprints de no más de dos meses. El objetivo general de todas las fases fue alcanzar los requerimientos de calidad establecidos para el proyecto en términos de efectividad de detección de transacciones y eliminar en cada versión del producto los errores (bugs) que disminuyeran la calidad del producto final.

c. Recursos Humanos: Definir el equipo. Con el equipo que se definió con anterioridad, se definió que el rol de dirección de proyecto debía estar a cargo de la persona con conocimientos en Ingeniería Industrial, el rol de análisis de Clustering por una persona con conocimientos en Ingeniería en Ciencias de la Administración y los roles de desarrollo de los algoritmos por personas con conocimientos en Ingeniería en Ciencias de la Computación.

d. Recursos Humanos: Desarrollo de las comunicaciones del equipo

- Directora del proyecto: Comunicación directa con los desarrolladores.
- Desarrolladores: Comunicación directa con la directora de proyecto para garantizar el desarrollo correcto y tiempo del proyecto. Comunicación directa con los todos los desarrolladores para obtener y brindar soluciones en base a experiencia del resto del equipo.

e. Recursos Humanos: Gestionar los recursos. Se definieron ciertas acciones para monitorear diferentes situaciones, se describen estas acciones en el cuadro a continuación:

Cuadro 38. Acciones a tomar para la gestión de recursos según la situación presentada

1. Desempeño de los miembros del equipo
Se monitoreó el desempeño revisando los informes proporcionados al final de cada sprint, donde se detallaban las acciones realizadas y los resultados obtenidos, en base a esto se determinaban los objetivos específicos de la siguiente fase.
2. Proporcionar retroalimentación
Se revisaban estos resultados junto con el desarrollador para determinar que el avance fuera significativo para validar los objetivos de la siguiente fase. Las retroalimentaciones se realizaban de forma personal en la siguiente reunión semanal luego haber enviado el reporte de fase terminada a la directora de proyecto.

Continuación Cuadro 38. Acciones a tomar para la gestión de recursos según la situación presentada

3. Resolver problemas
Al momento que un desarrollador detectara un problema que potencialmente pudiera dañar los objetivos del proyecto se comunicaba inmediatamente con la directora de proyecto, por la vía de su preferencia, por lo general vía correo electrónico. A este se le buscaba de manera conjunta para encontrar una solución donde el impacto se disminuyera.
4. Gestionar cambios para optimizar el desempeño del proyecto
Si la solución a un problema requería ayuda por parte del cliente o iba a cambiar de manera significativa alguna característica ofrecida al cliente, la directora de proyecto se comunicaba inmediatamente con el cliente para notificarle solicitar su ayuda o notificarles de los cambios a fin de lograr su aprobación del cambio.

Fuente: Elaboración propia

f. Comunicación: Distribuir la información. Toda información relativa al proyecto se centralizó en la plataforma virtual Asana. Esta contenía toda la información general e información específica de los diferentes módulos del Megaproyecto, y todos los miembros del equipo tenían acceso a esos proyectos para visualizar la información. Adicionalmente se tenía una carpeta compartida en Google Drive donde todos podían subir información importante relativa al proyecto (presentaciones de avances dadas a los clientes, información teórica relevante con el tema del proyecto, etc.).

g. Comunicación: Gestionar las expectativas. En la siguiente tabla se describe la forma en la que se gestionaron las expectativas del cliente y del equipo interno.

Cuadro 39. Gestión de las expectativas

Expectativas del cliente
Al inicio del proyecto se coordinaron reuniones con el cliente para obtener información sobre el producto esperado y también sobre las propuestas de investigación. Estas reuniones no tenían una periodicidad establecida, sino dependían de la cantidad de dudas o información de alguna de las dos parte que fuera relevante compartir. Luego, en la fase de desarrollo se enviaba al cliente los reportes de avances aproximadamente una vez al mes vía correo electrónico y cuando la información era más relevante de lo normal o el cliente tenía dudas sobre el reporte de avances enviado se agendaba una reunión con todo el equipo, ya fuera de forma virtual (vía Webbex) o el equipo iba a las instalaciones del cliente.
Expectativas del equipo interno
Se tenían reuniones presenciales semanales de una hora y media con todos los miembros del equipo de modo que todos estuvieran enterados del desempeño de todos, con ayuda de todo el equipo era más probable encontrar soluciones a las dificultades específicas de cada módulo.

Fuente: Elaboración propia

h. Adquisiciones: Efectuar. Se contó con un equipo de cinco personas, un estudiante de Ingeniería Industrial, un estudiante en Ciencia de la Administración y tres estudiantes de Ciencia de la Computación. No se realizaron adquisiciones materiales, debido a que el equipo y material que cada persona requiriera para llevar a cabo su rol, debía ser incluido en el costo por fase y producto final que cada miembro del equipo definió según el trabajo realizado. La única restricción fue no exceder el monto de la beca académica brindada.

6. Seguimiento y control.

a. Integración: Monitorear y controlar. Se monitoreaba el avance con las reuniones semanales con el equipo de trabajo. Adicionalmente todas las fases y fechas de entrega estaban establecidas en Asana. Se integró la aplicación con un Gantt que mostraba las actividades de acuerdo al código de colores:

- Rojo: Fases retrasadas
- Azul: Fases en tiempo de desarrollo
- Verde: Fases en terminadas y aprobadas por la directora de proyecto

Estos códigos de colores hacen referencia a los colores utilizados en los cuadros de mando integral (BSC), a diferencia que se utiliza el color amarillo en vez del azul. No se utilizó el amarillo debido a que el azul es el color predeterminado que utiliza la plataforma virtual.

b. Integración: Control integrado. Debido a que se gestionó por medio de una metodología ágil, fue posible responder a los cambios de manera rápida. Incluso durante el proceso de desarrollo era factible realizar cambios e identificar problemas de manera rápida. Todos los cambios hechos en cada fase fueron documentados, justificados y reportados en el reporte de fase terminada (Ver en Anexos los reportes de fases terminadas de los algoritmos y modelos desarrollados).

c. Alcance: Verificar el alcance. El alcance era revisado al final de cada sprint mediante el reporte que cada desarrollador entregaba sobre los resultados finales del sprint y en base a esto se establecían los nuevos objetivos del siguiente sprint. Si el sprint no había alcanzado los resultados esperados, se rediseñaba el siguiente sprint a modo de solucionar los problemas sin afectar los objetivos del proyecto. El alcance de cada fase fue reportado en los reportes de fase finalizada.

d. Alcance: Controlar el alcance. El alcance era controlado al finalizar e iniciar una fase por medio de reuniones presenciales con los desarrolladores para revisar el reporte de fase finalizada. Los objetivos de cada fase siempre se construyeron en función de alcanzar el objetivo del proyecto, teniendo en cuenta la línea base del mismo. La metodología ágil permitió realizar los cambios necesarios en el proceso

(aunque fueran cambios grandes) y poder al final alcanzar el objetivo del proyecto. La documentación de las acciones realizadas en cada fase se encuentra en los reporte de fase terminada.

e. Tiempo: Controlar el cronograma. El cronograma se controló por medio de Asana, el cual se integró con Instagantt para llevar un control actualizado del estado y los cambios de inicio o finalización de las fases.

f. Costos: Controlar los costos. Como los costos del proyecto fueron costos por los servicios de desarrollo brindados por los miembros del equipo, cada fase del proyecto incluía el reporte de horas invertidas en las mismas. Cada desarrollador distribuyó la cantidad de trabajo total en las diferentes fases. El desglose de los costos por cada fase se encuentra en la sección de cierre del proyecto. Debido a la naturaleza del proyecto (trabajo de graduación) y el tipo de remuneración obtenida (beca académica) se debía concluir el proyecto sin opción a dejar el proyecto incompleto por falta de presupuesto.

g. Calidad: Controlar la calidad. La calidad se monitoreó por medio de los reportes de fase terminada que cada desarrollador realzaba al concluir una fase, estos eran revisados a fin de garantizar que en base a los resultados de una fase, la siguiente buscara aumentar la calidad de los resultados de la fase anterior.

h. Comunicación: Informar el desempeño. Se centralizó la información en Asana, ahí se incluía la información de cada fase. A esta plataforma tenían acceso todos los miembros del equipo y también el cliente de la modalidad de megaproyecto. De esta manera fue posible mantener a todas las partes involucradas enteradas de la situación general del proyecto.

Adicionalmente se realizaban reuniones con el cliente cada vez que alguna de las partes los considerara necesario para poder describir con mayores detalles el desempeño de cada módulo del proyecto.

i. Riesgos: Monitorear y controlar. En base al análisis de riesgos previamente planteado, se le dio seguimiento a todas las posibles áreas críticas. El riesgo que durante el desarrollo llegó a ser una amenaza real fue la de la obtención de los datos. Inicialmente el cliente brindó una serie de datos en el formato se consideró apropiado, pero a medida que se inició el desarrollo del mismo y los prototipos eran más avanzados, todos los desarrolladores requerían de los datos en un formato diferente, una mayor cantidad y cierta correlación entre los mismos para poder entrenar los modelos para poder obtener el desempeño deseado. Se solicitaron las especificaciones de los datos requeridos al cliente con anticipación, pero este tuvo problemas para obtener los datos en tiempo. Por esta razón inició a analizar el plan de respuesta al riesgo que era el de generar por parte del equipo los datos. Se estableció la fecha límite en la cual se podría esperar que el cliente brindara los datos al equipo a fin de poder iniciar con el plan de respuesta e implementarlo a tiempo para poder cumplir con la fecha de finalización planteada. Se comunicó

al cliente la situación y las fechas límites antes de iniciar con el plan de respuesta. Sin embargo no fue necesario implementar el plan de respuesta porque el cliente agilizó el proceso de obtención de datos luego de informarle sobre la fecha límite y logró brindar los datos días antes de la fecha límite.

j. Adquisiciones: Administrar. Se administraron las adquisiciones de recurso humano por medio del reporte de fases terminadas. Con esto fue posible llevar control sobre el desempeño de cada persona, la calidad de su trabajo e identificar si fuera necesario las fallas para corregirlas a tiempo.

7. Cierre

a. Integración: Cerrar el proyecto. Para cerrar el proyecto se realizó una comparación de los desempeños finales de los tres algoritmos desarrollados y en base a esto se determinó el más adecuado. El que obtuvo un mejor desempeño fue el de Redes Neurales, ya que obtuvo un porcentaje de asertividad de 99.18%.

Cada uno de los desarrolladores realizó un reporte individual con toda la documentación del desarrollo del módulo, los resultados obtenidos y conclusiones y recomendaciones pertinentes a cada caso. Este reporte junto con el código fuente de los modelos fue brindado al cliente de la modalidad de megaproyecto y a la Universidad del Valle de Guatemala.

b. Resumen final de los módulos. En las siguientes tablas se muestran las fases de la ejecución de cada uno de los módulos con sus respectivas fechas de inicio y finalización y el costo total desglosado por el costo por fase. El costo por fase lo definió cada desarrollar en base a la cantidad de tiempo y recursos que debía invertir en cada fase.

Se hace la aclaración que si bien se obtuvo un beneficio económico por el desarrollo del proyecto, este beneficio fue brindado como “beca académica” a los miembros del equipo del megaproyecto, no como un pago en sí, debido a que fue un megaproyecto realizado como trabajo de graduación, no un proyecto con fines de lucro. En base a esta aclaración la única restricción que se tuvo para el costeo total fue que este no excediera el monto de la beca académica brindada. No fue de importancia en este caso la proporción por debajo que estuviera el costo con el presupuesto asignado, ya que el cliente brindó el presupuesto para ser utilizado como una beca académica por los miembros del equipo y definió el presupuesto total de $17,500.00 como precio de mercado por el producto que recibiría al final del megaproyecto. La sección de costos se realizó con el objetivo de poder luego realizar un análisis de modelo de negocio donde se pudiera plantear un posible precio de venta que estuviera por encima del costo del proyecto.

Cuadro 40. Desglose de fases y costo de Redes neurales

REDES NEURALES			
FASE	**FECHA DE INICIO**	**FECHA DE FINALIZACIÓN**	**COSTO**
Red neural v0	26/enero/2014	14/febrero/2014	Q. 2,400.00
Red neural v1.0	15/febrero/2014	10/marzo/2014	Q. 2,760.00
Red neural v1.1	11/marzo/2014	31/marzo/2014	Q. 2,400.00
Red neural v1.2	01/abril/2014	01/mayo/2014	Q. 2,520.00
Red neural v1.3	02/mayo/2014	30/mayo/2014	Q. 2,900.00
Red neural v1.4	01/junio/2014	31/junio/2014	Q. 3,100.00
Red neural v1.5	01/julio/2014	31/Julio/2014	Q. 3,100.00
Red neural v1.6	01/agosto/2014	31/Agosto/2014	Q. 3,720.00
Red neural v1.7	01/septiembre/2014	30/septiembre/2014	Q. 3,600.00
	TOTAL		**Q. 27,400.00**

Fuente: Elaboración propia

Cuadro 41. Desglose de fases y costo de Support Vector Machines (SVM)

SUPPORT VECTOR MACHINES (SVM)			
FASE	**FECHA DE INICIO**	**FECHA DE FINALIZACIÓN**	**COSTO**
Investigación de SVM	15/enero/2014	19/febrero/2014	Q. 2,500.00
Investigación de librerías SVM	20/febrero/2014	03/abril/2014	Q. 1,500.00
SVM versión 1	04/abril/2014	16/mayo/2014	Q. 4,000.00
SVM versión 2	17/mayo/2014	27/junio/2014	Q. 4,000.00
SVM versión 3	28/junio/2014	08/agosto/2014	Q. 4,000.00
SVM versión 4	09/agosto/2014	05/septiembre/2014	Q. 4,000.00
SVM versión final	06/septiembre/2014	01/octubre/2014	Q. 4,000.00
		TOTAL	**Q.24,000.00**

Fuente: Elaboración propia

Cuadro 42. Desglose de fases y costo de Redes Bayesianas

REDES BAYESIANAS			
FASE	**FECHA DE INICIO**	**FECHA DE FINALIZACIÓN**	**COSTO**
1. Investigación de redes bayesianas	17/enero/2014	07/febrero/2014	Q. 450.00
2. Investigación de librerías de BayesPy	08/febrero/2014	21/febrero/2014	Q. 300.00
3. Investigación e implementación de ambiente de desarrollo	22/febrero/2014	21/marzo/2014	Q. 2,500.00
4. Elaboración de mini proyecto aplicando redes bayesianas	22/marzo/2014	11/abril/2014	Q. 1,000.00
5. Estudio de análisis y datos de entrenamiento / avances con redacción de informe	12/abril/2014	06/junio/2014	Q. 1,000.00
6. Análisis de mini proyecto	09/junio/2014	08/agosto/2014	Q. 1,000.00
7. Definición de nodos a utilizar para la implementación de red bayesiana	09/agosto/2014	15/agosto/2014	Q. 2,000.00
8. Definición de diagrama relacional	16/agosto/2014	22/agosto/2014	Q. 1,000.00
9. Definición de reglas probabilísticas	23/agosto/2014	29/agosto/2014	Q. 2,500.00
10. Reestructuración de proyecto	01/septiembre/2014	12/septiembre/2014	Q. 1500.00
11. Generación de umbrales y ponderaciones	16/septiembre/2014	26/septiembre/2014	Q.2,500.00
12. Computación de nodos y cálculo de efectividad	27/septiembre/2014	30/septiembre/2014	Q. 500.00
TOTAL			Q. 17,250.00

Fuente: Elaboración propia

Cuadro 43. Desglose de fases y costo de Reconocimiento de patrones

RECONOCIMIENTO DE PATRONES			
FASE	**FECHA DE INICIO**	**FECHA DE FINALIZACIÓN**	**COSTO**
1. Investigación de redes bayesianas	06/enero/2014	28/febrero/2014	Q. 15,000.00
2. Investigación de librerías de BayesPy	01/marzo/2014	01/abril/2014	
3. Investigación e implementación de ambiente de desarrollo	02/abril/2014	01/mayo/2014	
4. Elaboración de mini proyecto aplicando redes bayesianas	02/mayo/2014	01/junio/2014	
5. Estudio de análisis y datos de entrenamiento / avances con redacción de informe	02/junio/2014	01/julio/2014	
6. Análisis de mini proyecto	02/julio/2014	01/agosto/2014	
7. Definición de nodos a utilizar para la implementación de red bayesiana	02/agosto/2014	01/septiembre/2014	
8. Definición de diagrama relacional	02/septiembre/2014	01/octubre/2014	
		TOTAL	**Q.15,000.00**

Fuente: Elaboración propia

Los resultados finales en efectividad (% porcentaje de aciertos) en cada modelo son los siguientes:

Cuadro 44. Resultados finales de efectividad en cada módulo

RESULTADOS FINALES		
MODELO	% aciertos	% desaciertos
1. Redes neurales	99.18	0.82
2. SVM	80.41	19.59
3. Redes Bayesianas	71.63	28.37

Fuente: Elaboración propia

En el trabajo individual de cada módulo y en el trabajo consolidado del Megaproyecto se incluye el desglose de los resultados finales obtenidos en cada módulo, así como los porcentajes de falsos positivos e información adicional. En la tabla anterior solo se muestran los porcentajes de aciertos y desaciertos obtenidos en cada módulo porque ese fue el criterio determinante para la elección del mejor modelo. Si la diferencia entre los resultados finales de efectividad obtenida entre las opciones hubiera sido menor al 5% se hubieran ponderado otros factores para tomar la decisión final sobre el mejor modelo. Sin embargo en este caso la diferencia entre la primera y segunda opción es de 18.77%, lo que permite dejar en claro que el algoritmo óptimo entre las opciones disponibles es el de **Redes neurales**.

b. Adquisiciones: Cerrar. Se definió como fecha límite de finalización de la fase de ejecución el 1 de octubre de 2014. Para marcar la finalización la ejecución, cada desarrollador entregó el último reporte de fase terminada donde incluyeron los resultados finales obtenidos. A partir de esa fecha se dio inicio a la fase de cierre, donde se recolectó la información del desarrollo del proyecto y se plasmó en reportes individuales y en un reporte consolidado del megaproyecto. Se definió como fecha de entrega el 15 de octubre de 2014 para los reportes individuales y el 30 de octubre para el reporte consolidado.

B. Sistema de control de permisos de software

Se gestionaron los permisos de software para dos secciones:

- Para los lenguajes de programación que se utilizarían para el desarrollo de los algoritmos y el análisis de Clustering.
- La licencia que se seleccionaría para el producto final del proyecto.

Debido a la funcionalidad y librerías disponibles se seleccionó el lenguaje de programación PYTHON como el idóneo para desarrollar los 3 modelos de algoritmos. PYTHON posee la licencia de código abierto “Python Software Foundation License”, la cual es similar a la licencia pública general GNU. Para desarrollar el análisis de Clustering se utilizó el lenguaje de programación R, el cual también se distribuye bajo una licencia de código abierto, la licencia GNU GPL.

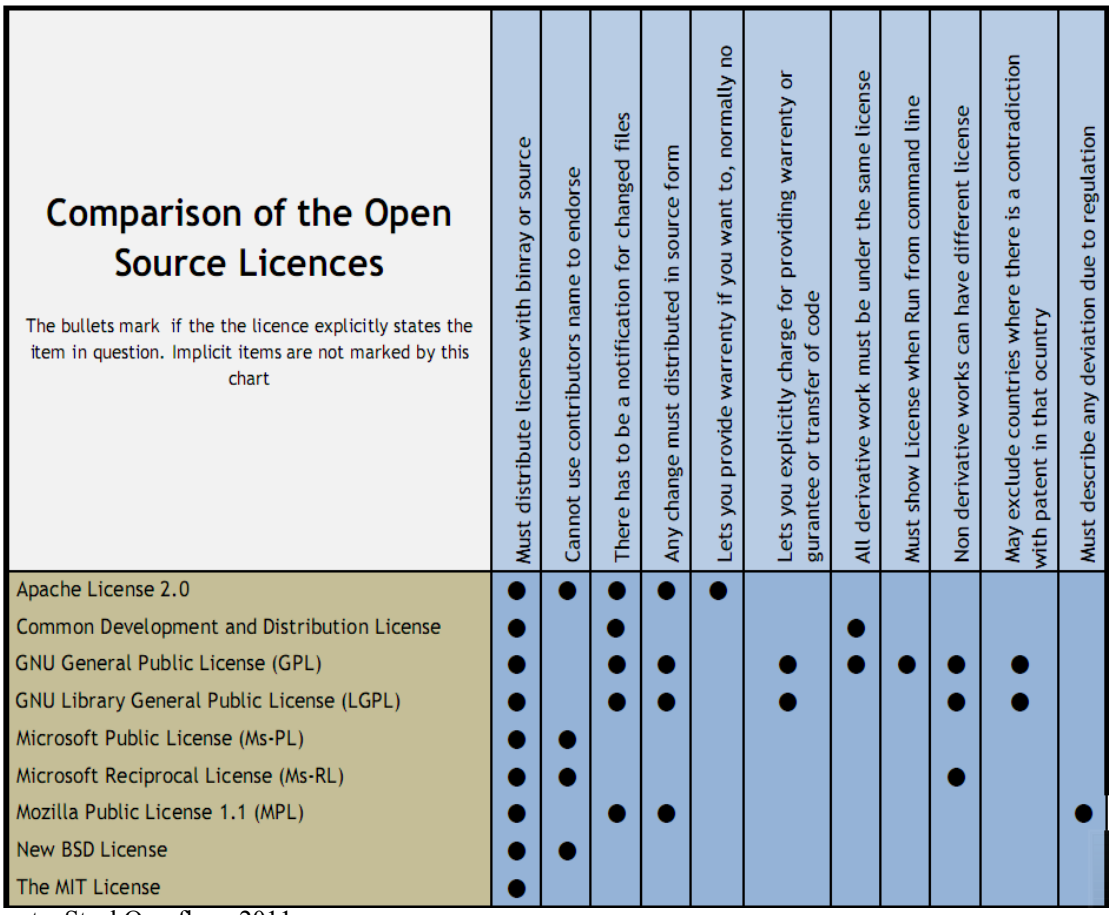
Dado que se seleccionaron lenguajes de código abierto no se requirieron acciones adicionales para poder utilizar los mismos.

Para seleccionar el tipo de licencia de software que se utilizaría para el producto final de los algoritmos que se desarrollaron, se definió inicialmente que se escogería una licencia de código abierto ya que por la naturaleza del proyecto (trabajo de graduación) se necesitaba una tipo de licencia donde se

permitieran modificaciones y redistribuciones libres del código fuente y los archivos binarios sin necesitar el permiso del autor original no tener que pagar a este regalías adicionales por este uso. La elección de una licencia de este tipo resolvía cualquier tipo de conflictos futuros ya que tanto la Universidad como el cliente de la modalidad de megaproyecto podían hacer uso del producto final sin restricciones.

Partiendo de esto se evaluaron las opciones de licencias de código abierto con las ventajas y desventajas de cada una, las cuales se describen en el siguiente cuadro comparativo.

Figura 72. Comparación de licencias de software libre



Fuente: StackOverflow, 2011

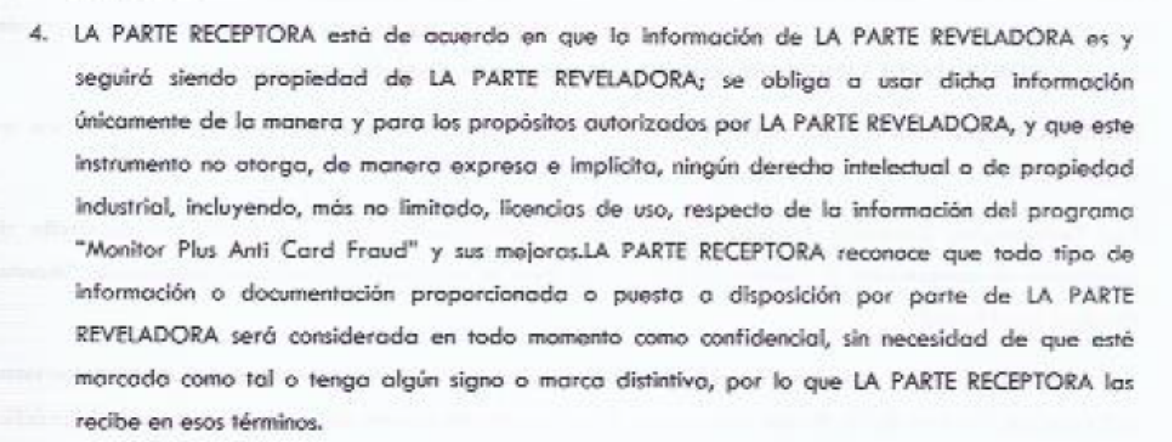
Con base a un consenso democrático con todos los miembros del equipo se decidió utilizar la Licencia MIT (MIT License) debido a ser considerada la que más se aplica para las necesidades del proyecto.

Esta es una de las licencias más básicas y permisivas de código abierto, la característica principal por la que se eligió es que aunque la versión original tiene licencia de software libre otras versiones modificadas por otro usuario no necesariamente deben tener una licencia de código abierto, lo cual se

adapta a las necesidades del cliente, el cual claramente no quisiera tener con licencia de código abierto la versión modificada compatible con su sistema actual.

Posteriormente a haberse definido esto se firmó un contrato de confidencialidad con el cliente de la modalidad de megaproyecto donde también se le concedían los derechos de uso del producto final del proyecto, tal como se puede observar en el siguiente extracto del contrato de confidencialidad (en Anexos se puede encontrar el contrato completo):

Figura 73. Extracto del contrato de confidencialidad



4. LA PARTE RECEPTORA está de acuerdo en que la información de LA PARTE REVELADORA es y seguirá siendo propiedad de LA PARTE REVELADORA; se obliga a usar dicha información únicamente de la manera y para los propósitos autorizados por LA PARTE REVELADORA, y que este instrumento no otorga, de manera expresa e implícita, ningún derecho intelectual o de propiedad industrial, incluyendo, más no limitado, licencias de uso, respecto de la información del programa "Monitor Plus Anti Card Fraud" y sus mejoras.LA PARTE RECEPTORA reconoce que todo tipo de información o documentación proporcionada o puesta a disposición por parte de LA PARTE REVELADORA será considerada en todo momento como confidencial, sin necesidad de que esté marcado como tal o tenga algún signo o marca distintivo, por lo que LA PARTE RECEPTORA las recibe en esos términos.

Fuente: Plus Technologies and Innovations, 2013

Esta fue la resolución final respecto a la licencia del software por lo que ya no fue necesaria la utilización de la licencia seleccionada con anterioridad (MIT license).

C. Resultados del análisis de la cadena de valor y propuesta de modelo de negocio

1. Análisis de la cadena de valor y validación del producto final. Según estudios recientes realizados en América Latina respecto a las tendencias de los medios de pago donde se determinó que:

- El efectivo solo ha ocupado un 1.5% del valor total de las transacciones realizadas en los últimos años.
- Más del 50% del valor total de las transacciones han sido realizadas por medio de transferencia de crédito.

- Dentro de los procesos de bancarización los procesos de mayor subcontratación suelen ser aquellos más alejados del core del negocio de las entidades. Especialmente en materia de seguridad de las transacciones y prevención del fraude.
- En los últimos años ha existido un aumento sostenido en la posesión de tarjetas de débito y/o crédito por país.
- El porcentaje de población que está considerando contratar alguna tarjeta en el próximo año es mayor 10%.
- Dentro de los aspectos clave en la elección de tarjetas de crédito, el tercer lugar lo ocupa la seguridad ante robos o fraudes que esta pueda ofrecer.

Con esta información se realizó el siguiente análisis de la cadena de valor y validación del producto final.

Se definió la siguiente cadena de valor:

a. **Eslabón 1:** desarrollador de software de detección de fraude en tarjetas de crédito y/o débito.

Necesidad: Creación de un software competitivo en el mercado con desempeño similar o mayor al que posee actualmente el mercado.

b. **Eslabón 2:** empresa distribuidora de software de detección de fraude en tarjetas de crédito y/o débito.

Necesidad: Vender un software de detección de transacciones fraudulentas superior a sus competidores para poder obtener ventajas económicas mayores.

c. **Eslabón 3:** entidad financiera que brinda el servicio de tarjetas de crédito y/o débito.

Necesidad: Brindar un producto/servicio (tarjetas de crédito y/o débito) confiable para sus clientes a modo de mantener a los clientes actuales y atraer nuevos.

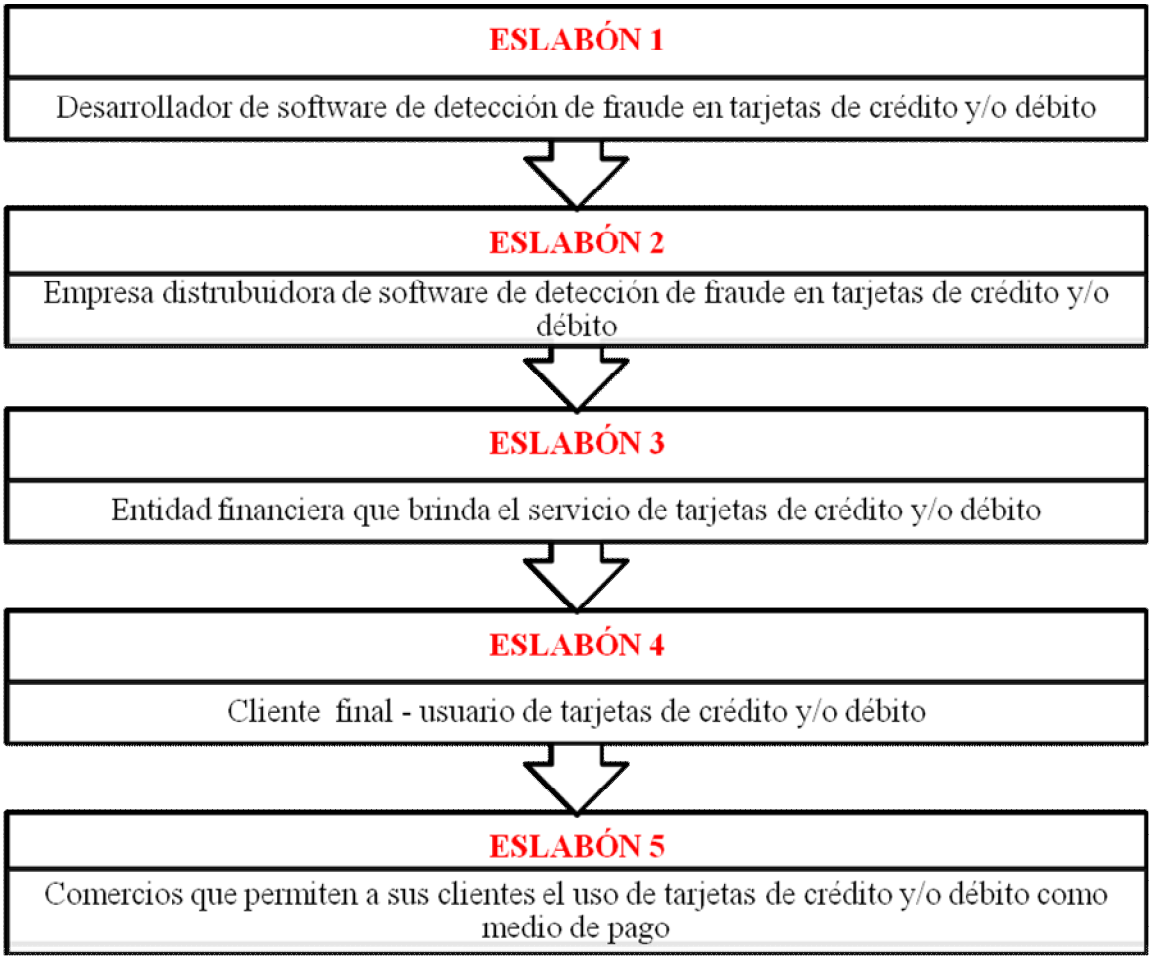
d. **Eslabón 4:** cliente final- usuario de tarjetas de crédito y/o débito.

Necesidad: Contar con un medio de pago no solo práctico, sencillo de utilizar y con beneficios sino también que sea confiable.

e. **Eslabón 5:** comercios que permiten a sus clientes el uso de tarjetas de crédito y/o débito como medio de pago.

Necesidad: Clientes con disponibilidad económica de comprar sus productos y/o servicios. La s tarjetas de crédito y débito suplen esa necesidad.

Figura 73. Cadena de valor del producto final



Fuente: Elaboración propia

Una vez definidos todos los eslabones de la cadena de valor y sus respectivas necesidades, se procede al análisis de los posibles canales de venta del producto, a fin de maximizar los beneficios de todas las partes interesadas. En este análisis se le dará prioridad a la búsqueda del escenario más rentable para el desarrollador de software, ya que esta es la parte que representa el proyecto.

2. Validación del producto. Todos los escenarios que se muestran a continuación parten de la funcionalidad del producto generó con el presente proyecto. En todos los casos es posible utilizar el producto que se creó como base, adaptándolo según requiera cada escenario.

a. OPCIÓN # 1. La primera opción a analizar es la forma actual y tradicional en la que funciona la cadena de valor para este proyecto:

El primero eslabón, que sería el desarrollador del software de detección de fraude de crédito y

débito, quien vende el producto al segundo eslabón, la empresa distribuidora del software de detección de fraude en tarjetas de crédito y débito. Este eslabón es quien lo vende a las entidades financieras y estas se benefician al poder atraer y mantener a sus clientes por brindarles un servicio confiable, lo que a su vez hace que los consumidores utilicen sus tarjetas como principal medio de pago en sus consumos.

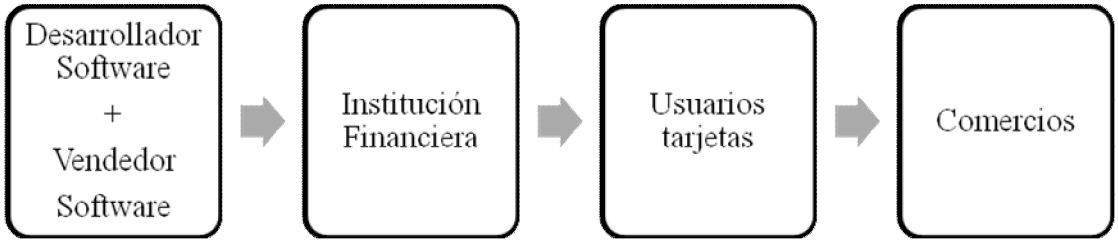
En este escenario todos los eslabones perciben un beneficio, pero el principal beneficio económico lo recibe la empresa que compró el software y la entidad financiera. Esto es debido a que al vender el software se vende este a un precio fijo, independientemente del volumen de transacciones que se analicen y sin importar el tiempo por el cual utilicen el software. Los beneficios económicos percibidos por un mayor volumen de transacciones procesadas y por la determinación de un pedido determinado de tiempo en que se brinda el servicio son percibidos por la empresa que tiene el contacto directo con las entidades financieras y por eso percibe esos beneficios.

Figura 74. Cadena de valor en opción No. 1

Fuente: Elaboración propia

b. OPCIÓN # 2. La segunda opción es fusionar el eslabón uno y dos a fin de desarrollar el software y venderlo directamente a las entidades financieras. En este caso si se perciben beneficios adicionales por mayor volumen de transacciones procesadas por el software y se pueden emitir licencias para limitar el lapso de tiempo por el cual se puede utilizar el software. Una vez finalizado el tiempo de validez se debe renovar la licencia para continuar utilizando el software. En este escenario ni la entidad financiera que utiliza el software ni el resto de la cadena de valor percibe cambios en los beneficios recibidos en comparación con el escenario anterior.

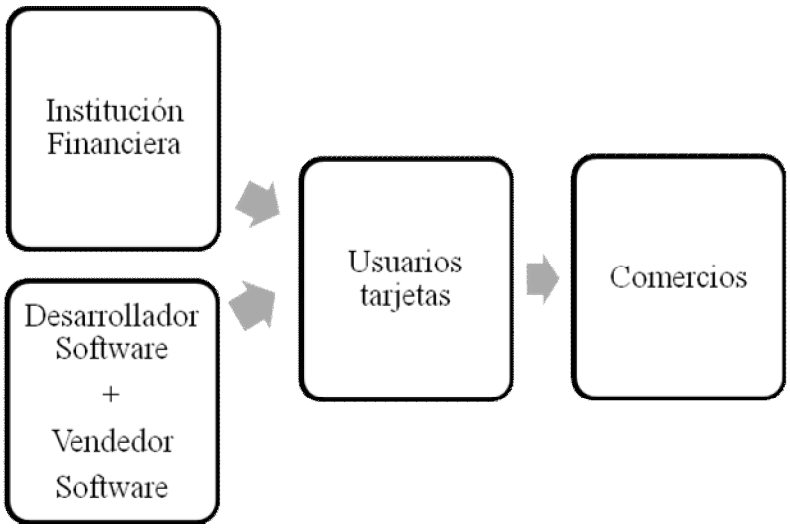
Figura 75. Cadena de valor en opción No. 2



Fuente: Elaboración propia

c. Opción # 3. El siguiente escenario sería ofrecer el producto (software) directamente al cliente final y que este pagara una tarifa mensual (equivalente al seguro que tiene opción a pagar actualmente con la entidad financiera de la cual posee la tarjeta) a cambio de obtener el servicio de seguridad que le estaría brindando el nuevo eslabón conformado por el desarrollador y el vendedor de software. Si el cliente subcontrata el servicio este no tendría que pagar más de lo que paga actualmente porque se buscaría mantener una cuota similar a la actual y así ser una opción competitiva "mismo costo, mayores beneficios". De esa forma se obtendrían ingresos pequeños de forma masiva, lo cual podría ser potencialmente mucho más lucrativo de lo que ya es actualmente. Sin embargo es un canal difícil de establecer ya que no se podría asegurar una cantidad significativa de clientes desde el inicio para poder identificar patrones y detectar fraude. Si bien se podrían utilizar métodos como las redes de mercadeo para alcanzar un número significativo de clientes en poco tiempo, se iniciaría sin datos históricos de todos los clientes (información que sí poseen los bancos) por lo que tomaría cierto tiempo en identificar patrones y durante este tiempo no se podría asegurar que el modelo alcance la asertividad que el mercado busca. Además en esta opción se le haría competencia directa a la entidad financiera respecto al servicio de seguridad, lo cual estos verían como una amenaza en lugar de una alianza estratégica.

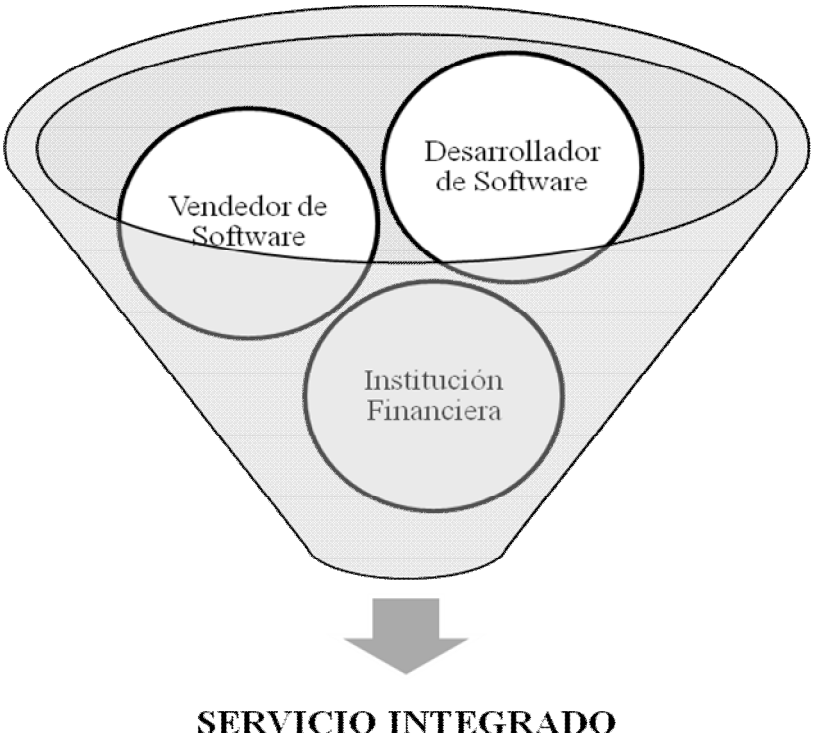
Figura 76. Cadena de valor en opción No. 3



Fuente: Elaboración propia

d. Opción # 4. Por último, otro canal posible sería fusionar los tres primeros eslabones para que este fuera un solo canal. La opción sería brindar un servicio integrado a la institución financiera. No vender el producto al banco sino vender el servicio completo de seguridad a los tarjetahabientes que la institución posea. De esa forma se maximizarían los beneficios en todos los eslabones de la cadena.

Figura 77. Descripción del servicio integrado



Fuente: Elaboración propia

En este caso el banco estaría "cediendo" por completo el departamento de detección de fraude. El pago que recibirían los primeros dos eslabones sería la tarifa que los clientes pagan por el seguro (obteniendo así los beneficios planteados en el escenario anterior, pero de manera factible). El banco tampoco tendría que pagar por los fraudes, ya que ese costo lo absorbería el nuevo departamento tercerizado de detección de fraude. Como obviamente el nuevo departamento quiere maximizar sus ganancias, este se va a encargar de tener un software cada vez más inteligente y asertivo, sin necesidad de esperar a que caduque la licencia para aplicar los cambios.

El cliente final (eslabón 4) por ende estaría más satisfecho con el servicio porque los casos de fraude van a disminuir. Esto aumentaría la confianza en el banco, lo que atraería más clientes para la entidad financiera (eslabón 3) y los comercios en general (eslabón 5) verían un aumento en sus ventas porque más clientes se sentirían en la libertar de realizar más compras con tarjetas porque tienen la certeza de poseer un respaldo de seguridad confiable y funcional.

Esta innovadora forma de ofrecer el servicio se podría potencializar mucho más si se fortalece el nombre de la empresa que brinde este servicio de seguridad a las entidades financieras. Se buscaría posicionar el nombre y prestigio de la empresa de seguridad para que esta fuera un sello distintivo público de calidad para el banco y por ende cada vez más entidades financieras van a querer ese servicio en específico porque los clientes finales quieren poseer este respaldo.

Figura 78. Cadena de valor en opción idónea

Fuente: Elaboración propia

3. Análisis de escenarios de las posibles opciones de costeo del producto. Se plantearon cinco posibles escenarios de software utilizando Redes neurales, ya que fue este el algoritmo seleccionado como el óptimo del proyecto, al haber obtenido el desempeño mayor a los demás algoritmos analizados (ver Tabla 24).

Los escenarios 1-4 son combinaciones entre las diferentes variables que toma en cuenta el escenario 5. El escenario 5 es el elegido como óptimo para el proyecto porque consigue la mejor combinación de variables que permiten obtener un alto porcentaje de efectividad (99.18%).

Como se puede observar en los escenarios 1-4 no se obtuvo una mayor efectividad conforme se tomaban más variables en cuenta. La forma en que funciona el entrenamiento de las redes neurales no es incremental, no necesariamente mejorara si toma más variables en cuenta. Hay variables que van a ayudar a mejorar que el método encuentre un porcentaje de error más bajo antes que otras, ahí radica la complejidad del modelo, en encontrar combinaciones idóneas que aumenten la asertividad de la red.

Para definir el precio de mercado se tomó límite superior el precio que el cliente de la modalidad de megaproyecto pagó por el presente proyecto. El precio de mercado del límite inferior se realizó una entrevista al Ing. Luis Fernando Cordón (Senior Information Security Consultant de Devel Security).

El posible precio de venta del escenario es el precio que el mercado está dispuesto a pagar por un sistema con una asertividad de aproximadamente 75% y el precio del escenario 5 es el precio que el mercado está dispuesto a pagar por un sistema con una asertividad mayor al 99%.

Como solamente se contaba con un precio mínimo y un precio máximo, se dividió la diferencia entre estos precios (Q.122,200.00) en la cantidad de escenarios. Así se pudo obtener un precio aproximado que se esperaría para cada escenario. De esta forma se determinó que por cada mejor escenario se estaría obteniendo en promedio Q.30,550.00 adicionales.

Para cada escenario se muestra el porcentaje de transacciones detectadas correctamente (% efectividad), el porcentaje de transacciones detectadas incorrectamente y el porcentaje de falsos positivos. Todos los datos son expresados con dos números decimales. Se detallan también las variables consideradas en cada escenario y el precio que se esperaría que el mercado pague en cada escenario. No se detallan a profundidad las variables que se incluyen en cada modelo ni la justificación del resultado, ya que esas especificaciones se encuentran en el trabajo individual del módulo "Redes neurales" y en el trabajo consolidado del megaproyecto.

Cuadro 45. Resultados del escenario No.1 de Redes neurales

	Escenario 1		
Red de decisiones	**% Transacciones detectadas correctamente**	**% Transacciones detectadas incorrectamente**	**% Falsos positivos**
Evaluación de fraude	75.17	24.83	25.00

Fuente: Elaboración propia

Cuadro 46. Detalle de la combinación de variables en el escenario No.1 de Redes neurales y su precio de mercado

DETALLE DE VARIABLES INCLUIDAS EN EL ESCENARIO # 1	**PRECIO DE MERCADO**
GENERALES	Q. 15,000.00
1. País del comercio	
2. Feriado(si/no)	
3. Bank Identification Number	
4. Hora de la Trx	
POR CLIENTE	
5. Monto	
6. País del comercio	

Fuente: Elaboración propia

Cuadro 47. Resultados del escenario No.2 de Redes neurales

Escenario 2			
Red de Decisiones	**% Transacciones detectadas correctamente**	**% Transacciones detectadas incorrectamente**	**% Falsos Positivos**
Evaluación de Fraude	78.70	21.30	25.00

Fuente: Elaboración propia

Cuadro 48. Detalle de la combinación de variables en el escenario No.2 de Redes neurales y su precio de mercado

DETALLE DE VARIABLES INCLUIDAS EN EL ESCENARIO # 2	**PRECIO DE MERCADO**
GENERALES	Q. 45,550.00
1. País del comercio	
2. Feriado (si/no)	
3. Bank Identification Number	
POR CLIENTE	
4. Monto	
5. País del comercio	

Fuente: Elaboración propia

Cuadro 49. Resultados del escenario No.3 de Redes neurales

Escenario 3			
Red de decisiones	**% Transacciones detectadas correctamente**	**% Transacciones detectadas incorrectamente**	**% Falsos positivos**
Evaluación de fraude	81.97	18.03	20.00

Fuente: Elaboración propia

Cuadro 50. Detalle de la combinación de variables en el escenario No.3 de Redes neurales y su precio de mercado

DETALLE DE VARIABLES INCLUIDAS EN EL ESCENARIO # 3	PRECIO DE MERCADO
GENERALES	Q.76,100.00
1. País del comercio	
2. Feriado (si/no)	
3. Bank Identification Number	
4. Hora de la Trx	
5. Tipo de producto de la tarjeta	
POR CLIENTE	
6. Monto	
7. País del comercio	

Fuente: Elaboración propia

Cuadro 51. Resultados del escenario No.4 de Redes neurales

	Escenario 4		
Red de decisiones	**% Transacciones detectadas correctamente**	**% Transacciones detectadas incorrectamente**	**% Falsos positivos**
Evaluación de fraude	93.82	6.18	0.00

Fuente: Elaboración propia

Cuadro 52. Detalle la combinación de variables en el escenario No.4 de Redes neurales y su precio de mercado

DETALLE DE VARIABLES INCLUIDAS EN EL ESCENARIO # 4	PRECIO DE MERCADO
GENERALES	Q. 106,650.00
1. País del comercio	
POR CLIENTE	
2. Monto	
3. País del comercio	

Fuente: Elaboración propia

Cuadro 53. Resultados del escenario No.5 de Redes neurales

Escenario 5			
Red de decisiones	**% Transacciones detectadas correctamente**	**% Transacciones detectadas incorrectamente**	**% Falsos positivos**
Evaluación de fraude	99.18	0.82	0.00

Fuente: Elaboración propia

Cuadro 54. Detalle la combinación de variables en el escenario No.5 de Redes neurales y su precio de mercado

DETALLE DE VARIABLES INCLUIDAS EN EL ESCENARIO # 5 (COMPLETO)	**PRECIO DE MERCADO**
GENERALES	Q.137,200.00
1. País del comercio	
2. Feriado (si/no)	
3. Bank Identification Number	
4. Hora de la Trx	
5. Tipo de Producto de la tarjeta	
6. Merchant Category Code	
POR CLIENTE	
7. Monto	
8. País del comercio	

Fuente: Elaboración propia

4. Propuesta del escenario idóneo como modelo de negocio. Según los escenarios analizados se seleccionó como escenario idóneo el escenario#5 del capítulo anterior.

Cuadro 55. Resultados del escenario de Redes neurales determinado como idóneo

Escenario IDÓNEO			
Red de decisiones	**% Transacciones detectadas correctamente**	**% Transacciones detectadas incorrectamente**	**% Falsos positivos**
Evaluación de fraude	99.18	0.82	0.00

Fuente: Elaboración propia

Cuadro 56. Detalle la combinación de variables en el escenario de Redes neurales seleccionado como idóneo y su precio de mercado

DETALLE DE VARIABLES INCLUIDAS EN EL ESCENARIO IDÓNEO	PRECIO DE MERCADO
GENERALES	Q.137,200.00
1. País del comercio	
2. Feriado (si/no)	
3. Bank Identification Number	
4. Hora de la Trx	
5. Tipo de producto de la tarjeta	
6. Merchant Category Code	
POR CLIENTE	
7. Monto	
8. País del comercio	

Fuente: Elaboración propia

Esto es debido a la que con este modelo toda la cadena de valor incrementa los beneficios recibidos. El eslabón # 1 (desarrollador de software de detección de fraude de tarjetas de crédito y/o débito), que es el eslabón que representa el presente proyecto percibe un incremento económico de Q. 122,200.00 en comparación con el que recibiría con un escenario básico de 75% de asertividad. Este aumento significativo de beneficios económicos lo obtiene en el escenario de negocio actual. Los potenciales beneficios que recibirían si se combinara el escenario idóneo con la cadena de valor propuesta como óptima para el proyecto (alianza estratégica entre el desarrollador del software, vendedor del software y las instituciones financieras para brindar un servicio integrado) serían mucho mayores y se estarían explotando los beneficios en todos los eslabones. Todos los demás eslabones aumentan también los beneficios obtenidos con la opción de cadena de valor tradicional (opción # 1) y maximizarían los beneficios con la opción # 4 de la cadena de valor (servicio integrado).

X. CONCLUSIONES

1. Se obtuvo un algoritmo de clasificación basado en redes bayesianas que permite clasificar transacciones electrónicas con un 71.63% de exactitud. Esta efectividad no cumple la hipótesis planteada inicialmente.
2. Por medio de la aplicación de la Regla de Bayes en un conjunto de datos, se puede determinar con precisión la probabilidad de un evento (en este caso catalogado como transacción) para ocurrir en un futuro.
3. Al realizar la serie de aprendizaje de la red bayesiana, se encontró que mientras es mayor la cantidad de datos que se analizan, más pierde sensibilidad dicha red. Es decir, existe una mayor cantidad de valores únicos que, al compararse con el total de transacciones, generan un bajo índice de probabilidad de ocurrencia, a pesar que la cantidad de datos que representa pueda ser muy grande. V.g. diez mil transacciones es un número que haría una probabilidad de ocurrencia relativamente grande, es decir, una transacción que cumpliese con este patrón, no se categorizaría anómala. Sin embargo, si el conjunto de datos es de siete millones de datos, estas diez mil transacciones tendrían una probabilidad de ocurrencia de 0.14%, haciendo así que la transacción sea categorizada como fraude.
4. La técnica de árboles de decisión permitió establece un rango de categoría de negocios, 5455-5561, en los cuales están abarcados la mayor cantidad de fraudes. El tener un rango de MCC es muy importante porque existen ciertas categorías de negocios en donde el índice de fraude es mayor. Aplicar medidas de seguridad en estos comercios puede tener un impacto positivo en la lucha contra el fraude en tarjetas de crédito.
5. Clustering permitió establecer que la mayoría de transacciones fraudulentas no excedieron el monto de $169. Aplicar esta información puede tener implicaciones positivas si se implementa controles de seguridad estrictos cuando son montos relativamente pequeños.
6. Regresión logística reafirmó la importancia del MCC dentro del estudio de transacciones fraudulentas en tarjetas de crédito y débito. Realizar una revisión de los comercios que pertenecen a estos rangos puede reducir los índices de fraudes en tarjetas de crédito y débito.
7. Se demostró que las variables MCC, Condición del punto de venta y País adquirente son variables relevantes en los tres modelos planteados por lo que prestar atención a estas variables puede ser de gran ayuda para la empresa con la que se trabajó en la búsqueda de reducir las transacciones fraudulentas.
8. El entrenamiento reiterado del algoritmo de propagación hacia atrás elástico, determinó que el proceso puede llegar a sobre-ajustar los datos de entrenamiento al comportamiento de la red neuronal.

9. Los mejores resultados de la red de decisión obtenidos fueron del algoritmo de propagación hacia atrás sin modificaciones en comparación al algoritmo modificado que obtuvo mejores resultados individuales por módulo.
10. El uso de técnicas de ajuste para la distribución sesgada de los datos aseguró que la red neuronal no generalizará únicamente la clase mayoritaria en los datos. Esto se comprobó luego con los valores de falsos positivos y falsos negativos.
11. Se determinó que, utilizando el algoritmo de SVM seleccionado, la cantidad de falsos positivos sería de 18.85% y la cantidad de falsos negativos sería 17.47%.
12. Utilizando la SVM seleccionada, no se logra disminuir el índice de falsos positivos ni de falsos negativos que el sistema de detección de transacciones fraudulentas que la empresa posee, actualmente se debajo de 10%. Sin embargo utilizando otras SVMs se logró llegar a menos de 10% en uno de los índices. Por tanto no se considera que las SVMs no puedan disminuir dichos índices sino que deben realizarse nuevos estudios para lograr reducir estos índices.
13. Se determinó que los beneficios de todos los eslabones de la cadena se potencializarían con un modelo que ofreciera un servicio integrado para la prevención y pronta detección de fraude en tarjetas de crédito y débito.
14. Se analizaron, de acuerdo a escenarios, posibles opciones de costeo del producto Se definió el escenario # 5 de Redes neurales como el idóneo al obtener una efectividad de 99.18% y con el cual se puede obtener in ingreso aproximado de Q.137,200.00
15. Se propuso de acuerdo al escenario # 5 de Redes neurales el modelo de negocio adecuado, el cual sería la opción # 4 de la cadena de valor propuesta (servicio integrado).

XI. RECOMENDACIONES

1. Se recomienda que previo a definir los datos que se utilizarán para realizar este tipo de estudio, se realice un análisis estadístico o de sensibilidad, que permita determinar qué nodos tienen mayor influencia en el comportamiento de los datos. De esta forma, se puede garantizar que la definición de datos influya de forma positiva al aprendizaje y posterior análisis de las transacciones.
2. Se debe evaluar la distribución de los datos y patrones significativos para la detección de fraude pueden sugerir estructuras de redes neuronales más elaboradas y efectivas para la identificación.
3. Utilizar métodos alternativos de validación cruzada pueden permitir obtener datos sobre la validez y confianza que se puede obtener del entrenamiento de las redes resultantes. Realizar distintas comparaciones permite cuantificar el grado de sobreajuste al que puede estar sesgado el modelo.
4. El planteamiento de los datos utilizando distintos tipos de redes neuronales pueden mostrar mejorías con respecto al uso de redes modulares de perceptrones de múltiples capas.
5. Se recomienda realizar nuevos estudios con SVMs con diferentes kernels, que permitan generar hiperplanos que puedan acomodarse mejor a las transacciones.
6. Se recomienda realizar un estudio para crear un sistema de SVMs por capas, para mejorar la detección, por ejemplo utilizando SVMs que puedan detectar sólo transacciones fraudulentas y sólo transacciones no fraudulentas como input de una nueva SVM.
7. Para una implementación real, se recomienda utilizar datos de entrenamiento recolectados a lo largo de, al menos, un año, ya que los datos utilizados para este proyecto fueron de tres meses solamente, y con una mayor balance entre la cantidad de transacciones fraudulentas y transacciones no fraudulentas.
8. Crear otro megaproyecto en la Universidad del Valle de Guatemala partiendo de los resultados finales obtenidos con el megaproyecto “Algoritmos Inteligentes para Reconocimiento de Patrones de Comportamiento Transaccionales”. Con este nuevo megaproyecto se buscaría implementar el modelo de Redes neurales identificado como idóneo en el software de detección de fraude de la empresa.
9. Definir desde el inicio los requerimientos de datos de prueba brindados por el cliente y el formato idóneo para obtenerlos, ya que de esto depende en gran parte la calidad del modelo final que se obtenga. Además es el eslabón más débil del proyecto, ya que los retrasos en la obtención de los mismos pueden dañar el alcance del proyecto.

XII. BIBLIOGRAFÍA

Abad-Grau, María M; Ierache, Jorge; Cervino, Claudio. 2007. *Aplicación de Redes Bayesianas en el Modelado de un Sistema Experto de Triaje en Servicios de Urgencias Médicas*. Trelew, Argentina. IX Taller de Investigadores en Ciencias de la Computación. Vol. 1. p. 43–47.

Alejandro, G., Vega, S., & Ruiz, J. 2012. *Algoritmos de agrupamiento conceptuales: un estado de arte.* La Habana, Cuba: Centro de aplicaciones de tecnologías de avanzada.

Afi & Tecnocom. 2013. *Informe Tecnocom sore tendencias en medios de pago 2013*. http://www.tecnocom.es/Documentos%20Web%20Tecnocom/Informe_Tecnocom'13_WEB.pdf [06 de junio de 2013]

Arroyo Cervantes, G., & Camacho Castillo, O. (s.f.). *Regresión lineal simple*. Recuperado el 15 de 10 de 2014, de http://regresionsimple.galeon.com/

Barber, David. 2011. *Bayesian Reasoning and Machine Learning*. Cambridge. Prensa de la Universidad de Cambridge. 590 págs.

Berenson, M., Levine, D., & Krehbiel, T. 2001. *Estadística para administración.* México D.F: Prentice Hall.

Busogain, X. 2008. *Redes neuronales y sus aplicaciones.* Bilbao, España: Escuela superior de ingeniería de Bilbao.

Calderón Méndez, N. d. 2006. *Minería de datos: Una herramienta para la toma de decisiones.* Guatemala: Universidad San Carlos de Guatemala.

Castillo, Enrique; Gutiérrez, José Manuel y Hadi, Ali S. 1996. *Sistemas expertos y modelos de redes probabilísticas*. Madrid. Academia de Ingeniería. 627 págs.

Champandard, Alex. *Introduction to Artificial Intelligence*. http://ai-depot.com/Intro.html [17 de septiembre de 2014]

Charniak, E. 1991. *Bayesian Networks without tears*. http://www.cs.ubc.ca/~murphyk/Bayes/Charniak_91.pdf [10 de enero de 2014]

Chen, T. 2010. *Credit card and debit card transaction volume statistics.* http://www.nerdwallet.com/blog/credit-card-data/credit-card-transaction-volume-statistics/ [01 de junio de 2013]

Complutense, U. 2008. *Análisis de regresión lineal: El procedimiento regresión lineal*. Madrid: Universidad Complutense.

Contreras Lopez, J. 2014. *Medidas de dispersión: Capitulo 15*. Hidalgo: Universidad Michoacana de San Nicolás de Hidalgo.

Cook, R., & S., W. 1982. *Residuals and influence in regression*. Londres: Chapman and Hall.

De Veaux, R., Bock, D., & Velleman, P. 2003. *Intro Stats.* Boston: Addison-Wesley.

Del Brío, B. M., & Carlos, S. C. 1995. *Fundamentos de las redes neuronales artificiales: hardware y software.*Zaragoza, España: Universidad de Zaragoza.

Ezequiel, P. *Redes Bayesianas aplicadas a minería de datos inteligente*. Buenos Aires. Capital Federal: Facultad de Ingeniería. Universidad de Buenos Aires

Fayyad, U., & Smyth, P. 1996. *From data mining to knowledge discovery in Databases: an overview.* Ai Magazine.

Freud Williams, P. 1990. *Estadística para la administración*. Mexico D.F: Prentice Hall Hispanoamérica.

García Ferrando, M. 1989. *Socioestadística: Introducción a la estadística en sociología*. Madrid: Alianza editorial.

García, P., & Azaustre, C. 2008. *Minería de datos aplicada a las redes sociales*. Madrid: Universidad Carlos III de Madrid España.

Hartigan, J. 1975. Clustering Algorithms.New York: Wiley.

Heckerman, D. 1996. *A tutorial on learning with Bayesian Networks.* http://research.microsoft.com/pubs/69588/tr-95-06.pdf [10 de enero de 2014]

Heckerman, D.; Geiger, D. y Chickering, D. 1995. *Learning Bayesian Networks: The Combination of Knowledge and Statistical Data*. Luwer Academic Publishers. Págs 197-243.

Hernández, J. O. 2004. *Introducción a la minería de datos.* Madrid: Pearson education.

Hernandez, J., & Ferri, C. 2007. *Introducción a la minería de datos.*Pearson. Prentice Hall.

Hosmer, D. W., & Lemeshow, S. 2000. *Applied logistic regression.*Estados Unidos: Wiley Interscience.

Kizys, R., & Juan, M. A. 2005. *Modelo de regresión lineal múltiple.* Cataluña: Universidad Oberta de Catalunya.

Licensing - *Is there an Open Source licence matrix?*. (2011, Septiembre 1). Retrievered from http://stackoverflow.com/a/7277525 [2014/05/01]

MacCarthy, J. 2007. *What is artificial intelligence?, Basic Questions*. http://www-formal.stanford.edu/jmc/whatisai/node1.html [08 de febrero de 2014]

Martinez, C. A. 2012. *Aplicación de técnicas de minería de datos para mejorar el proceso de control de gestión en Entel*. Santiago de Chile: Universidad de Chile.

Martinez, F., Díaz, M. C., Rivas, V. M., y Ureña, L. A. 2003. *Aplicación de redes neuronales y redes bayesianas en la detección de multipalabras para tareas IR*. Madrid. Artículo presentado en las II Jornadas de Tratamiento y Recuperación de la Información. 7 págs.

Mastercard. 2014. *Press releases, Mastercard incorporated reports fourth-quarter and full-year 2013 financial results.* http://newsroom.mastercard.com/press-releases/mastercard-incorporated-reports-fourth-quarter-and-full-year-2013-financial-results-2 [06 de junio de 2013]

Matich, D. J. 2001. *Redes neuronales: Conceptos básicos y aplicaciones.* Rosario, Argentina: Universidad tecnológica nacional de Rosario.

Mendoza Zapeta, P. F. 2013. *Minería de datos para encontrar perfiles de colaboradores satisfechos y los factores que inciden en su satisfacción*. Tesis Universidad San Carlos de Guatemala. Guatemala. 106 págs.

Mills, R. 1981. *Estadística para economía y administración*. Bogotá: McGraw-Hill.

MongoDB, Inc. *Introduction to MongoDB*. http://docs.mongodb.org/manual/core/introduction/ [23 de octubre de 2014]

Montaño Moreno, J. J. 2002. *Redes neuronales artificiales aplicadas al anáisis de datos.* Mallorca: Universitat de Les Illes Balears.

Montiel, A., Rius, F., & Barón, F. 1997. *Elementos básicos de estadística económica y empresarial.* Madrid: Prentice Hall.

Morales Peña, O. 2011. *Métodos Cuantitativos II.* Guatemala: Materiales educativos.

MongoDB, Inc. *MongoDB: A Document Oriented Database*. http://www.mongodb.org/about/ [23 de octubre de 2014]

Pearl, J. y Russel, S. 2011. *Bayesian Networks*. California. Department of Statistics, UCLA. http://escholarship.org/uc/item/53n4f34m [10 de enero de 2014]

Project Management Institute. 2008. *Guía de los fundamentos para la dirección de proyectos (Guía del PMBOOK®)*. Cuarta Edición. Estados Unidos. Project Management Institute, Inc. 393 pp.

Puga, Jorge López y García, Juan. 2008. *Sistemas de Tutorización Inteligente Basados en Redes Bayesianas*. Almería. Universidad de Almería. 13 págs.

Python Software Foundation. *What is Python? Executive Summary*. https://www.python.org/doc/essays/blurb/ [23 de octubre de 2014]

Rivera, Miller. 2011. *El papel de las redes bayesianas en la toma de decisiones*. Bogotá. Laboratorio de Modelamiento y Simulación, Universidad del Rosario. 11 págs.

Roche, A. 2009. *Árboles de decisión y series de tiempo.* Montevideo: Universidad de la República.

Rodríguez Ortiz, R. F. 2007. *Análisis de correlación y regresión lineal simple aplicado a la cartera de cuentas por cobrar de una distribuidora farmacéutica.* Guatemala: Universidad San Carlos de Guatemala.

Romero, L. A., & Calonge, T. 2009. *Redes neuronales y reconocimiento de patrones.* Valladolid, España: Universidad de Valladolid.

Rumelhart, D., & McClelland, J. 1986. *Parallel distributed processing.*Boston: MIT Press.

Russell, Stuart J. y Norvig, Peter. 1995. *Artificial intelligence: a modern approach.* Englewood Cliffs, N.J. Prentice Hall. 946 págs.

Salas, R. 2009. *Redes neuronales artificiales*. Valparaiso, Chile: Universidad de Valparaiso.

Salcedo Poma, C. M. 2002. *Estimación de la ocurrencia de incidencias en declaraciones de pólizas de importación.* Lima: Universidad Nacional mayor de San Marcos.

Sebastiani, Paola. 2010. *A Tutorial on Probability Theory*. Amherst. Department of Mathematics and Statistics. University of Massachusetts. 25 págs.

Seijas, L. M. 2011. *Reconocimiento de patrones utilizando tecnicas estadísticas y conexionistas aplicadas a la clasificación de dígitos manuscritos.* Buenos Aires: Facultad de Ciencias Exactas de Naturales de la Universidad de Buenos Aires.

Servante, M. 2002. *Algoritmos TDIDT aplicados a la minería de datos inteligente.* Buenos Aires, Argentina: Universidad de Buenos Aires.

Shrotriya, Siddharth. 2013. *Imitating Humans: A Technical Approach*. S.I. Lulu Com. [02 de mayo de 2014]

Silva, Freddy Miguel. 2011. *Modelo de aprendizaje automático para la detección de fraudes electrónicos en transacciones financieras*. Tesis Universidad Centroccidental Lisandro Alvarado. Barquisimeto, Venezuela. 77 págs.

Smola, Alex y Vishwanathan, S.V.N. 2010. *Introduction to Machine Learning*. Cambridge. Prensa de la Universidad de Cambridge. 226 págs.

Sotolongo, G., & Suárez, C. 1999. *Modular bibliometrics information system with propietary software.* Colima, México: Universidad de Colima.

Sucar, Luis Enrique. 2012. *Capítulo 2: Bayesian Networks and Influence Diagrams. Decision theory models for applications in artificial intelligence: concepts and solutions*. Hershey, PA. Information Science Reference. 28 págs.

StatSoft. 2014. *Neural networks*. Recuperado el 10 de Octubre de 2014, de http://www.statsoft.com

TSYS. 2014. TSYS 2014 *Canadian consumer payment choice study*. http://www.tsys.com/Downloads/upload/2014-TSYS-Canadian-Consumer-Payment-Choice-Study.pdf [06 de junio de 2013]

U. C. 2008. *Análisis de conglomerados: El procedimiento conglomerados de K-medias*. Madrid: Universidad Complutense. 21 págs.

Uribeetxeberria, Roberto y Zurutuza Urko. 2005. *Revisión del estado actual de la investigación en el uso de data mining para la detección de intrusiones.* Escuela Politécnica Superior. Mondragón Unibertsiatea. 8 págs.

Utgoff, P. 1988. *An incremental ID3.* California: Morgan Kaufmann Publishers.

Visa. 2014. News Details, Visa Inc. *Reports Fiscal first quarter 2014 net income of $1.4 billion or $2.20 per diluted share*. http://investor.visa.com/news/news-details/2014/Visa-Inc-Reports-Fiscal-First-Quarter-2014-Net-Income-of-14-billion-or-220-per-diluted-share/default.aspx [06 de junio de 2013]

Webster, A. 2000. Estadística aplicada a los negocios y la economía. Bogotá: McGraw-Hill.

XIII. ANEXOS

A. Código fuente del algoritmo de análisis para la red bayesiana

```
# -*- coding: cp1252 -*-
from pymongo import MongoClient
import gc
import pickle

#Ponderaciones definidas como 0.3, 0.5, 0.7 y 1.0, siendo 1.0 la mayor importancia
BANDERA_ROJA = 3
BANDERA_AMARILLA = 2
BANDERA_VERDE = 1

class ComercioID:

   #Método que define la bandera en que cae el comercio
def definirScore(self):
if(self.porcentaje == 0.0):
self.bandera = BANDERA_ROJA
      elif(self.porcentaje > 0.0 and self.porcentaje < 5.0):
         self.bandera = BANDERA_AMARILLA
      else:
         self.bandera = BANDERA_VERDE

   #Método que define la puntuacion (extrae el porcentaje)
   def computarPuntuacion(self):
      dia,mes,anio = self.fecha.split('-')
match = self.db.comercioID.find_one({ 'COMERCIO_ID':self.comercioID,
'ANIO':anio,
                                      'MES':mes,
                                      'DIA':dia,
})
if match:
```

```
self.porcentaje = match['PORCENTAJE']
else:
            self.porcentaje = 0.0
        self.definirScore()

    #Método inical
def __init__(self, comID, f):
self.connection = MongoClient("localhost", 27017)
        self.db = self.connection.reglas
self.comercioID = comID
        self.fecha = f
        self.ponderacion = 0.5
        self.computarPuntuacion()
self.score = self.ponderacion * self.bandera
self.connection.close()

#=====================================================================

class TipoTarjeta:

    #Método que define la bandera en que cae el TipoTarjeta
def definirScore(self):
if(self.porcentaje == 0.0):
self.bandera = BANDERA_ROJA
        elif(self.porcentaje > 0.0 and self.porcentaje <= 0.10):
            self.bandera = BANDERA_AMARILLA
        else:
            self.bandera = BANDERA_VERDE

    #Método que define la puntuacion (extrae el porcentaje)
    def computarPuntuacion(self):
        match = self.db.tipoTarjeta.find_one({ 'TIPO_TARJETA':self.tipoTarjeta,
'COMERCIO_ID':self.comercioID,
```

```
                                'BIN':self.bin,
                                })
if match:
self.porcentaje = match['PORCENTAJE']
else:
        self.porcentaje = 0.0
    self.definirScore()

  #Método inical
def __init__(self, tipoTar, comID, b):
self.connection = MongoClient("localhost", 27017)
    self.db = self.connection.reglas
self.tipoTarjeta = tipoTar
    self.comercioID = comID
    self.bin = b
    self.ponderacion = 0.5
    self.computarPuntuacion()
self.score = self.ponderacion * self.bandera
self.connection.close()

#================================================================

class MarcaTarjeta:

  #Método que define la bandera en que cae el MarcaTarjeta
def definirScore(self):
if(self.porcentaje <= 5.0):
self.bandera = BANDERA_ROJA
    elif(self.porcentaje > 5.0 and self.porcentaje <= 10.0):
        self.bandera = BANDERA_AMARILLA
    else:
        self.bandera = BANDERA_VERDE
```

```
   #Método que define la puntuacion (extrae el porcentaje)
   def computarPuntuacion(self):
      match = self.db.marcaTarjeta.find_one({ 'MARCA_TARJETA':self.marcaTarjeta,
                                 'TIPO_TARJETA':self.tipoTarjeta,
                                 'BIN':self.bin,
})
if match:
self.porcentaje = match['PORCENTAJE']
else:
         self.porcentaje = 0.0
      self.definirScore()

   #Método inical
def __init__(self, marcaTar, tipoTar, b):
self.connection = MongoClient("localhost", 27017)
self.db = self.connection.reglas
      self.tipoTarjeta = tipoTar
      self.marcaTarjeta = marcaTar
      self.bin = b
      self.ponderacion = 0.7
      self.computarPuntuacion()
      self.score = self.ponderacion * self.bandera
self.connection.close()

#=================================================================
====

class Moneda:

#Método que define la bandera en que cae el Moneda
def definirScore(self):
if(self.porcentaje <= 5.0):
self.bandera = BANDERA_ROJA
```

```
        elif(self.porcentaje > 5.0 and self.porcentaje <= 10.0):
            self.bandera = BANDERA_AMARILLA
        else:
            self.bandera = BANDERA_VERDE

    #Método que define la puntuacion (extrae el porcentaje)
def computarPuntuacion(self):
match = self.db.moneda.find_one({ 'MONEDA':self.moneda,
'TIPO_TARJETA':self.tipoTarjeta,
                                  'BIN':self.bin,
                                   'MARCA_TARJETA':self.marcaTarjeta,
                                   'COMERCIO_ID':self.comercioID
})
if match:
self.porcentaje = match['PORCENTAJE']
else:
            self.porcentaje = 0.0
        self.definirScore()

    #Método inical
def __init__(self, mon, tipoTar, b, marcaTar, comID):
self.connection = MongoClient("localhost", 27017)
self.db = self.connection.reglas
self.moneda = mon
self.tipoTarjeta = tipoTar
        self.marcaTarjeta = marcaTar
self.bin = b
        self.comercioID = comID
self.ponderacion = 1.0
        self.computarPuntuacion()
        self.score = self.ponderacion * self.bandera
self.connection.close()
```

```
        #=============================================================
=====

        class MontoTRX:

        #Método que define la bandera en que cae el MontoTRX
        def definirScore(self):
        if(self.porcentaje <= 5.0):
        self.bandera = BANDERA_ROJA
              elif(self.porcentaje > 5.0 and self.porcentaje <= 10.0):
                 self.bandera = BANDERA_AMARILLA
              else:
                 self.bandera = BANDERA_VERDE

           #Método que define la puntuacion (extrae el porcentaje)
        def computarPuntuacion(self):
        match = self.db.monto.find_one({ 'MONTO_TRX':self.montoTRX,
        'FERIADO':self.feriado,
                                  'TIPO_TARJETA':self.tipoTarjeta,
                                  'MONEDA':self.moneda,
                                  'COMERCIO_ID':self.comercioID
        })
        if match:
        self.porcentaje = match['PORCENTAJE']
        else:
                 self.porcentaje = 0.0
              self.definirScore()

           #Método inical
        def __init__(self, monto, fer, tipoTar, mon, comID):
        self.connection = MongoClient("localhost", 27017)
              self.db = self.connection.reglas
        self.montoTRX = monto
```

```
        self.feriado = fer
        self.tipoTarjeta = tipoTar
        self.moneda = mon
        self.comercioID = comID
        self.ponderacion = 0.7
        self.computarPuntuacion()
self.score = self.ponderacion * self.bandera
self.connection.close()

#=================================================================
=====

class Fraude:

   #Método que define la bandera en que cae el Fraude
def definirScore(self):
if(self.porcentaje <= 10.0):
self.bandera = BANDERA_ROJA
      elif(self.porcentaje > 10.0 and self.porcentaje <= 25.0):
         self.bandera = BANDERA_AMARILLA
      else:
         self.bandera = BANDERA_VERDE

   #Método que define la puntuacion (extrae el porcentaje)
def computarPuntuacion(self):
match = self.db.fraude.find_one({ 'MONTO_TRX':self.montoTRX,
'MONEDA':self.moneda,
                        'COMERCIO_ID':self.comercioID,
                        'CODIGO_TRX':self.codigoTRX,
                        'DIA':self.dia,
'MES':self.mes,
                         'ANIO':self.anio
                        })
```

```
if match:
self.porcentaje = match['PORCENTAJE']
else:
            self.porcentaje = 0.0
        self.definirScore()

    #Método inical
def __init__(self, monto, mon, comID, cod, f):
self.connection = MongoClient("localhost", 27017)
        self.db = self.connection.reglas
        self.montoTRX = monto
self.moneda = mon
        self.comercioID = comID
        self.codigoTRX = cod
self.dia, self.mes, self.anio = f.split("-")
self.ponderacion = 1.0
        self.computarPuntuacion()
        self.score = self.ponderacion * self.bandera
self.connection.close()

#==================ANALISIS DE DATOS ============================

conn = MongoClient("localhost", 27017)
bd = conn.megaproyecto
#test = self.db.MinTestData.find_one()
contador = 0
trxs = bd.MinTestData.find(timeout=False)
aciertos = 0
desaciertos = 0
falsos_positivos = 0
falsos_negativos = 0
trxs_analizadas = 0
for trx in trxs:
```

```
    #variables de la TRX
    #print trx
    comercioID = trx['COMERCIO_ID']
    fecha = str(trx['DIA'])+"-"+str(trx['MES'])+"-"+str(trx['ANIO'])
    tipoTarjeta = trx['TIPO_TARJETA']
    BIN = trx['BIN']
    marcaTarjeta = trx['MARCA_TARJETA']
    moneda = trx['MONEDA']
    monto = trx['RANGO_MONTO']
    feriado = trx['FERIADO']
    codigoTRX = trx['CODIGO_TRX']
    fraude = trx['FRAUDE']

    #creacion de nodos para la TRX
    nodoComercioID = ComercioID(comercioID,fecha)
    nodoTipoTarjeta = TipoTarjeta(tipoTarjeta, comercioID, BIN)
    nodoMarcaTarjeta = MarcaTarjeta(marcaTarjeta, tipoTarjeta, BIN)
    nodoMoneda = Moneda(moneda, tipoTarjeta, BIN, marcaTarjeta, comercioID)
    nodoMonto = MontoTRX(monto, feriado, tipoTarjeta, moneda, comercioID)
    nodoFraude = Fraude(monto, moneda, comercioID, codigoTRX, fecha)

    #analisis de TRX individual
    archivo_trx = open("analisis\\trxs\\trx"+str(contador)+".txt","wb")
    archivo_trx.write("============== COMERCIO ID ==============\r\n")
    archivo_trx.write("Bandera: "+str(nodoComercioID.bandera)+"\r\n")
    archivo_trx.write("Porcentaje: "+str(nodoComercioID.porcentaje)+"\r\n")
    archivo_trx.write("Ponderacion: "+str(nodoComercioID.ponderacion)+"\r\n")
archivo_trx.write("Score: "+str(nodoComercioID.score)+"\r\n")
archivo_trx.write("============== TIPO TARJETA ==============\r\n")
    archivo_trx.write("Bandera: "+str(nodoTipoTarjeta.bandera)+"\r\n")
    archivo_trx.write("Porcentaje: "+str(nodoTipoTarjeta.porcentaje)+"\r\n")
    archivo_trx.write("Ponderacion: "+str(nodoTipoTarjeta.ponderacion)+"\r\n")
    archivo_trx.write("Score: "+str(nodoTipoTarjeta.score)+"\r\n")
```

```
            archivo_trx.write("============== MARCA TARJETA ==============\r\n")
            archivo_trx.write("Bandera: "+str(nodoMarcaTarjeta.bandera)+"\r\n")
            archivo_trx.write("Porcentaje: "+str(nodoMarcaTarjeta.porcentaje)+"\r\n")
            archivo_trx.write("Ponderacion: "+str(nodoMarcaTarjeta.ponderacion)+"\r\n")
            archivo_trx.write("Score: "+str(nodoMarcaTarjeta.score)+"\r\n")
            archivo_trx.write("================== MONEDA ==================\r\n")
            archivo_trx.write("Bandera: "+str(nodoMoneda.bandera)+"\r\n")
            archivo_trx.write("Porcentaje: "+str(nodoMoneda.porcentaje)+"\r\n")
            archivo_trx.write("Ponderacion: "+str(nodoMoneda.ponderacion)+"\r\n")
        archivo_trx.write("Score: "+str(nodoMoneda.score)+"\r\n")
        archivo_trx.write("================== MONTO ==================\r\n")
            archivo_trx.write("Bandera: "+str(nodoMonto.bandera)+"\r\n")
            archivo_trx.write("Porcentaje: "+str(nodoMonto.porcentaje)+"\r\n")
            archivo_trx.write("Ponderacion: "+str(nodoMonto.ponderacion)+"\r\n")
        archivo_trx.write("Score: "+str(nodoMonto.score)+"\r\n")
        archivo_trx.write("================== FRAUDE ==================\r\n")
            archivo_trx.write("Bandera: "+str(nodoFraude.bandera)+"\r\n")
            archivo_trx.write("Porcentaje: "+str(nodoFraude.porcentaje)+"\r\n")
            archivo_trx.write("Ponderacion: "+str(nodoFraude.ponderacion)+"\r\n")
        archivo_trx.write("Score: "+str(nodoFraude.score)+"\r\n")

        #analisis de TRXs acumulado
            lista_nodos = [nodoComercioID, nodoTipoTarjeta, nodoMarcaTarjeta, nodoMoneda,
nodoMonto, nodoFraude]
            score_lista = []
            score_total = 0
            veredicto = 0 #Fraude=1;NoFraude=0
            bandera_alerta = 0 #Esta bandera se levanta cuando cae bandera amarilla (combinación: no
fraude + amarilla)
            for nodo in lista_nodos:
        score_total += nodo.score
        score_lista.append(nodo.score)
        score_promedio = score_total/len(lista_nodos)
```

```
if score_promedio <= 1.0:
veredicto = 0
elif score_promedio <= 2.0 and score_promedio > 1.0:
veredicto = 0
        bandera_alerta = 1
    else:
        veredicto = 1

    archivo_trx.write("================= ANALISIS =================\r\n")
    archivo_trx.write("Veredicto: "+str(veredicto)+", "+str(bandera_alerta)+"\r\n")
    archivo_trx.close()

    if(fraude == "S"):#Esto significa que la TRX debe ser marcada como fraude
        if veredicto == 1:
            aciertos += 1
        else:
            desaciertos +=1
            falsos_negativos +=1
    else:
        if veredicto == 1:
            desaciertos += 1
            falsos_positivos += 1
        else:
            aciertos += 1
    trxs_analizadas += 1

    resultado_parcial = open("analisis\\resultados\\parcial"+str(contador)+".txt","wb")
    resultado_parcial.write("Aciertos: "+str(aciertos)+"\r\n")
    resultado_parcial.write("Desaciertos: "+str(desaciertos)+"\r\n")
    resultado_parcial.write("Falsos positivos: "+str(falsos_positivos)+"\r\n")
    resultado_parcial.write("Falsos negativos: "+str(falsos_negativos)+"\r\n")
    resultado_parcial.write("Total analizados: "+str(trxs_analizadas)+"\r\n")
resultado_parcial.close()
```

```
with open("analisis\\resumen.txt", "a") as resumen:
resumen.write(str(contador)+": "+str(veredicto)+", "+str(bandera_alerta)+"\r\n")
resumen.close()
if contador % 1000 == 0: print contador
contador += 1
gc.collect()
trxs.close()
resultado_txt = open("analisis\\resultado.txt","wb")
resultado_txt.write("Aciertos: "+str(aciertos)+"\r\n")
resultado_txt.write("Desaciertos: "+str(desaciertos)+"\r\n")
resultado_txt.write("Falsos positivos: "+str(falsos_positivos)+"\r\n")
resultado_txt.write("Falsos negativos: "+str(falsos_negativos)+"\r\n")
resultado_txt.write("Total analizados: "+str(trxs_analizadas)+"\r\n")
resultado_txt.close()
```

B. Flujo de decisión en análisis de nodos (semáforos)

Figura 79. Flujo de procesamiento del nodo identificador de comercio

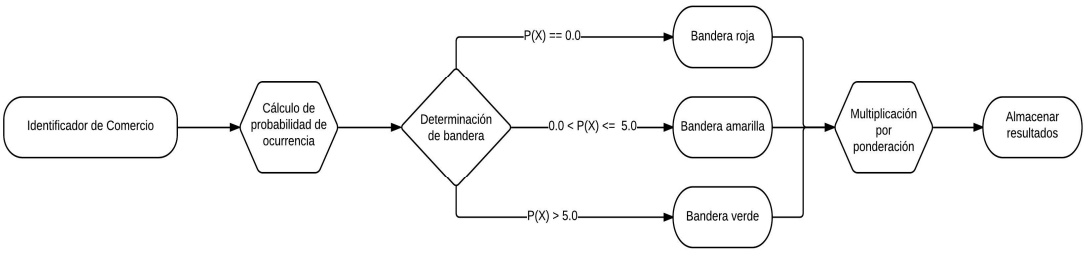


Figura 80. Flujo de procesamiento del nodo fraude

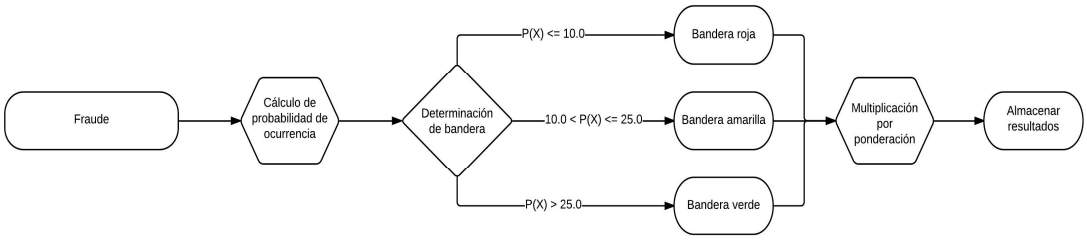


Figura 81. Flujo de procesamiento del nodo marca de tarjeta

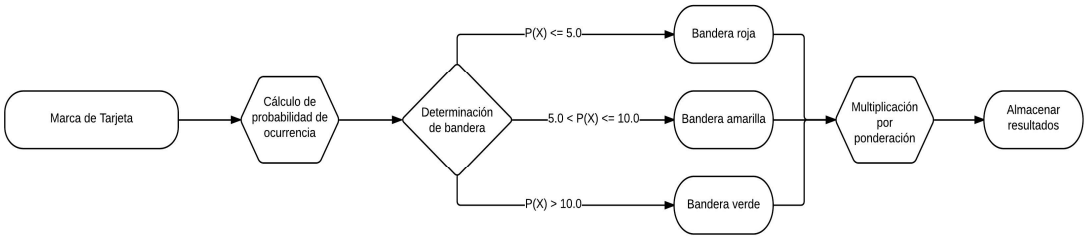


Figura 82. Flujo de procesamiento del nodo moneda

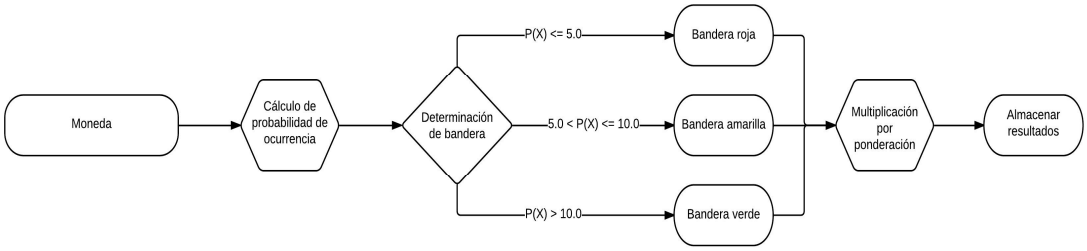


Figura 83. Flujo de procesamiento del nodo monto de transacción

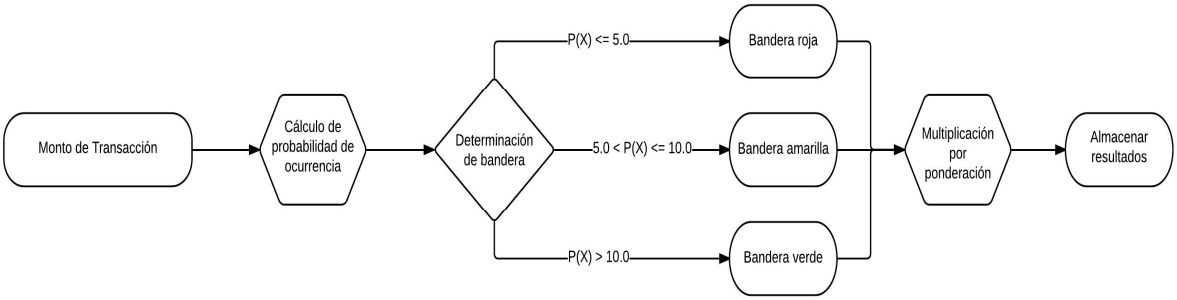


Figura 84. Flujo de procesamiento del nodo tipo de tarjeta

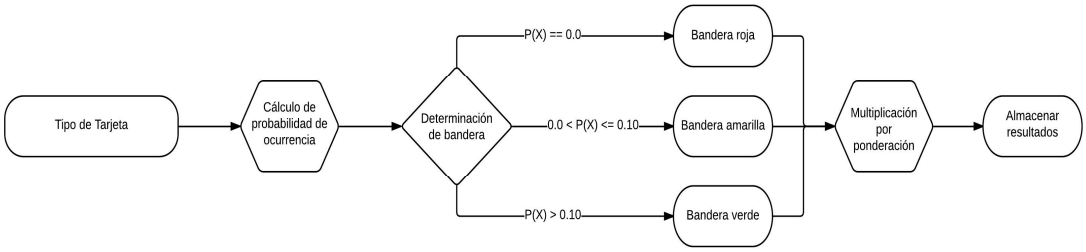
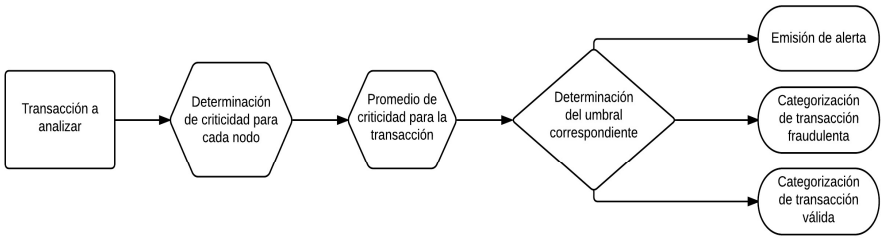


Figura 85. Flujo de programa de detección de transacciones fraudulentas



C. Código de las pruebas tipo 1 de la Fase 5 de SVM (Cargando Todas los datos de entrenamiento)

```
# Se carga el normalizador se abre la coneccion con mongo y se
crea la SVM
scaler= joblib.load("scaler.pkl")
client= MongoClient()
collection= client.megaproyecto.data
clf = SGDClassifier(class_weight={0:1,1:1})

# Se cargan lotes de 1 millon de transacciones
v =5
for i in range(v):
print(str(i)+"/"+ str(v))
print("Cargando datos.")
if i != v:
cursor= collection.find().limit(1000000).skip(i *1000000)
else:
cursor= collection.find().limit(275000).skip(i *1000000)
x =[]
    y =[]

for doc in cursor:
# Se agrega el valor 1 o 0 a la lista
# que contiene si una transaccion es fraudulenta o no.
if doc['C87601']==u'N':
y.append(0)
else:
            y.append(1)
# Se eliminan el id y el indicador de fraude de los atributos
del doc['C87601']
del doc['_id']
# Se agregan a la lista de atributos los atributos de la
transaccion
val= doc.values()
x.append(val)

x = np.array(x)
    y = np.array(y)
print("Datos Cargados")

print("Normalizando")
# Se realiza la normalizacion con el escalador.
    x = scaler.transform(x)
print("Datos normalizados")

print("Entrenando")
# Se realiza el entrenamiento.
```

```
            clf.partial_fit(x, y,[0,1])
        print("Entrenamiento terminado!")
```

D. Código de las pruebas tipo 2 de la Fase 5 de SVM (Cargando 2 millones de transacciones no fraudulentas)

```
        import gc, random
        import numpy as np
        from pymongo import MongoClient
        from sklearn.linear_model import SGDClassifier
        from sklearn.externals import joblib

        # Se carga el normalizador se abre la coneccion con mongo y se
crea la SVM
        scaler= joblib.load("scaler.pkl")
        client= MongoClient()
        collection= client.megaproyecto.data
        clf= SGDClassifier(class_weight='auto')

        # Se cargan las transacciones fraudulentas y se crea una lista
        # aleatorio para seleccionar las transacciones no fraudulentas
        cursor= collection.find({'C87601':'S'}).limit(1900)
        randomArray= random.sample(range(4273009),2000000)
        randomArray.sort()
        x =[]
        y =[]

        cont=0
        oldCont=0
        print("Cargando datos.")
        for i in randomArray:
        # Se carga la transaccion no fraudulenta correspondiente al
        # valor aleatorio.
        cursor1= collection.find({'C87601':'N'}).limit(1).skip(i)
        doc = cursor1[0]
        # Se agrega el valor 0 a la lista
        # que contiene si una transaccion es fraudulenta o no.
            y.append(0)
        # Se eliminan el id y el indicador de fraude de los atributos
        del doc['C87601']
        del doc['_id']
        # Se agregan a la lista de atributos los atributos de la
transaccion
        val= doc.values()
        x.append(val)
        cont+=1
```

```
        if oldCont +10000== cont:
        oldCont= cont
        print cont

        for doc in cursor::
        # Se agrega el valor 1 a la lista
        # que contiene si una transaccion es fraudulenta o no.
            y.append(1)
        # Se eliminan el id y el indicador de fraude de los atributos
        del doc['C87601']
        del doc['_id']
        # Se agregan a la lista de atributos los atributos de la
transaccion
        val= doc.values()
        x.append(val)

        x = np.array(x)
        y = np.array(y)
        print("Datos Cargados")

        print("Normalizando")
        # Se realiza la normalizacion con el escalador.
        x = scaler.transform(x)
        print("Datos normalizados")

        print("Entrenando")
        # Se realiza el entrenamiento.
        clf.partial_fit(x, y,[0,1])
        print("Entrenamiento terminado!")
```

E. Código de las pruebas tipo 3 de la Fase 5 de SVM (Utilizando la SVM final)

```
        import gc
        import numpy as np
        from pymongo import MongoClient
        from sklearn.linear_model import SGDClassifier
        from sklearn.externals import joblib")

        # Se carga el normalizador se abre la coneccion con mongo y se
crea la SVM
        scaler= joblib.load("scaler.pkl")
        client= MongoClient()
        collection= client.megaproyecto.data
        clf = SGDClassifier(class_weight={0:1,1:0.048})

        # Se realizan lotes de 500 mil transacciones no fraudulentas y
        # 1900 transacciones fraudulentas
```

```
        v =8
        for i in range(v):
        print(str(i)+"/"+ str(v))
        print("Cargando datos.")
        # Se cargan las transacciones
        cursor= collection.find({'C87601':'S'}).limit(1900)
        cursor1= collection.find({'C87601':'N'}).limit(500548).skip(i
*500548)
        x =[]
            y =[]

        for doc in cursor1:
        # Se agrega el valor 0 a la lista
        # que contiene si una transaccion es fraudulenta o no.
                y.append(0)
        # Se eliminan el id y el indicador de fraude de los atributos
        del doc['C87601']
        del doc['_id']
        # Se agregan a la lista de atributos los atributos de la
transaccion
        val= doc.values()
        x.append(val)

        for doc in cursor:
        # Se agrega el valor 1 a la lista
        # que contiene si una transaccion es fraudulenta o no.
                y.append(1)
        # Se eliminan el id y el indicador de fraude de los atributos
        del doc['C87601']
        del doc['_id']
        # Se agregan a la lista de atributos los atributos de la
transaccion
        val= doc.values()
        x.append(val)

        x = np.array(x)
            y = np.array(y)
        print("Datos Cargados")

        print("Preprocesando")
        # Se realiza la normalizacion con el escalador.
            x = scaler.transform(x)
        print("Preprocesamiento Terminado")

        print("Entrenando")
        # Se realiza el entrenamiento.
            clf.partial_fit(x, y,[0,1])
        print("Entrenamiento terminado!")
```

F. Script utilizado para árboles de decisión

```
# Setear el directorio de trabajo
setwd("C:/Users/Berny/Desktop/corridamegaproyecto")

# Leer el set de datos

trx <- read.csv("trx_allfields2.csv")

# Ver la estructura del set de datos

str(trx)

# Cargar las librerias que necesitamos

library(caTools)
library(rpart)
library(rpart.plot)

#Necesitamos convertir en factores las variables binarias ya que actualmente
#estan definidas como variables numericas.  Seleccionamos en un vector las
#columnas que corresponden a las variables binarias que usaremos como predictores

factores <- c(2,4:9,11:18)

# Ahora, utilizando un ciclo, convertimos cada una de ellas en factor

for (i in factores) {trx[,i] <- as.factor(trx[,i])}

# definimos otro set de datos eliminando la primera columna ya que solo es un
# correlativo y no tiene valor como predictor

trx2 <- trx[,-1]

# ahora revisamos la estructura del nuevo set de datos y vemos que ya no aparece
# la primera columna

str(trx2)

# Calculamos el baseline.  Para eso construimos la tabla y definimos que el
# baseline sera el valor mas frecuente de la variable respuesta

tab<- table(trx$C87601)
max(tab)/sum(tab)


# ahora procedemos a separar el set de datos para entrenamiento y validación

set.seed(123)
split = sample.split(trx$C87601, SplitRatio=0.75)
```

```
trxTrain = trx[split==TRUE,]
trxTest = trx[split==FALSE,]

# creamos un arbol de clasificacion utilizando todos los predictores

arbol<- rpart(C87601 ~., data=trxTrain, method="class", control=rpart.control(minbucket=4))

# graficamos el arbol para ver la estuctura

prp(arbol)

# ahora creamos la prediccion sobre el set de datos de validacion

pred<- predict(arbol,newdata=trxTest,type="class")

# calculamos la matriz de confusion y la precision del modelo

confumat<- table(trxTest$C87601, pred)
confumat
sum(diag(confumat))/sum(confumat)

# ahora veamos la complejidad del modelo y el error

plotcp(arbol)
arbolpod <- prune(arbol,cp=0.017)

# lo graficamos para ver nuevamente la estructura

prp(arbolpod)

#-----------------------------------------------------PRUEBA 2----------------------
arbol2<- rpart(C87601 ~. -C87518, data=trxTrain, method="class",
control=rpart.control(minbucket=4))

# graficamos el arbol para ver la estuctura

prp(arbol2)
# usamos este nuevo arbol para predecir sobre el set de datos de validacion

pred2<- predict(arbol2,newdata=trxTest,type="class")

# calculamos la nueva matriz de confusion y la precision del nuevo modelo

confumat2<- table(trxTest$C87601, pred2)
confumat2
sum(diag(confumat2))/sum(confumat2)

#-----------------------------------------------------PRUEBA 3----------------------
arbol3<- rpart(C87601 ~. -C87584, data=trxTrain, method="class",
control=rpart.control(minbucket=1))
```

```
# graficamos el arbol para ver la estuctura

prp(arbol3)
# usamos este nuevo arbol para predecir sobre el set de datos de validacion

pred3<- predict(arbol3,newdata=trxTest,type="class")

# calculamos la nueva matriz de confusion y la precision del nuevo modelo

confumat2<- table(trxTest$C87601, pred2)
confumat2
sum(diag(confumat2))/sum(confumat2)
```

G. Script utilizado para clustering

```
#----------------------------BERNY IXCAYAU---------------------

#Utilidamos set de datos.
setwd("C:/Users/Berny/Desktop/corridamegaproyecto")
#cargamos datos
trx<- read.csv("trx_allfields2.csv")

#Convertimos a numeros
factores <- c(1:25)
for (i in factores) {trx[,i] <- as.numeric(trx[,i])}
str(trx)

#Eliminamos las variables que no utilizaremos

trx2<- trx[,-c(1,2,4,11,18,20)]
str(trx2)

#Se normalizan las variables

trx2 <- scale(trx2, center=FALSE)

# ---------4clusters--------

set.seed(1234567)
clustrx<- kmeans(trx2,4)
table(clustrx$cluster, trx$C87601)

# ---------5clusters--------

set.seed(1234567)
clustrx1<- kmeans(trx2,5)
table(clustrx1$cluster, trx$C87601)

# ---------6clusters--------
```

```
set.seed(1234567)
clustrx2<- kmeans(trx2,6)
table(clustrx2$cluster, trx$C87601)

#Porcentaje de cluster 1
porcentaje1=table(trx$C87601,clustrx$cluster)
porcentaje1[1,]/(porcentaje1[1,]+porcentaje1[2,])

#Porcentaje de cluster 2
porcentaje2=table(trx$C87601,clustrx1$cluster)
porcentaje2[1,]/(porcentaje2[1,]+porcentaje2[2,])

#Porcentaje de cluster 3
porcentaje3=table(trx$C87601,clustrx2$cluster)
porcentaje3[1,]/(porcentaje3[1,]+porcentaje3[2,])

#CLuster con mayor reelevencia
trxclus3 <- trx[clustrx2$cluster==6,]

# exploremos graficamente las características

#MCC
barplot(table(trxclus3$C87510))
#Monto original trx
barplot(table(trxclus3$C87504))
#Codigo pais adqiriente
barplot(table(trxclus3$C87543))
#Condicion pto venta
barplot(table(trxclus3$C87512))
#Pais origen
barplot(table(trxclus3$C87519))
#marca o franquicia
barplot(table(trxclus3$C87566))
#credito o debito
barplot(table(trxclus3$C87586))
#grupo dia
barplot(table(trxclus3$C87594))


# Una forma de determinar un numero apropiado de clusters es validar la
# reducción en la variacion total que se logra al incrementar un cluster

# inicializamos la variable que va a guardar la variacion total para cada
# modelo

vtot <- 0
# generamos un modelo para cada numero de clusters desde 1 hasta 8
# y para cada uno, guardamos la variacion total intracluster
```

```
for (i in 1:8) {mod <- kmeans(trx2,i);
vtot[i] <- mod$tot.withinss}

# ploteamos la vtot y el # de clusters y vemos la grafica

plot(1:8, vtot, type="b", xlab="num clusters", ylab= "var tot intracluster")

#----------------------------------------------PRUEBAS CON DATOS SOLO FRAUDE----------------
--------------
#cargamos datos
tr <- read.csv("trxfraude2.csv")

#Convertimos a numeros
factores <- c(1:25)
for (i in factores) {tr[,i] <- as.numeric(tr[,i])}

#Eliminamos las variables que no utilizaremos

tr2<- tr[,-c(1,2,4,11,18,20)]
str(tr2)

#Se normalizan las variables

tr2 <- scale(tr2, center=FALSE)

# ---------4clusters--------

set.seed(1234567)
clustr<- kmeans(tr2,4)
table(clustr$cluster, tr$C87601)

# ---------5clusters--------

set.seed(1234567)
clustr1<- kmeans(tr2,5)
table(clustr1$cluster, tr$C87601)

# ---------6clusters--------

set.seed(1234567)
clustr2<- kmeans(tr2,6)
table(clustr2$cluster, tr$C87601)

#Porcentaje de cluster 1
por1=table(tr$C87510,clustr$cluster)
por1[2,]/(por1[1,]+por1[2,])

#Porcentaje de cluster 2
por2=table(tr$C87510,clustr1$cluster)
por2[2,]/(por2[1,]+por2[2,])
```

```
#Porcentaje de cluster 3
por3=table(tr$C87510,clustr2$cluster)
por3[2,]/(por3[1,]+por3[2,])

#CLuster con mayor reelevencia
trxclus3 <- trx[clustrx2$cluster==1,]

# exploremos graficamente las características

#MCC
barplot(table(trxclus3$C87510))
#Monto original trx
barplot(table(trxclus3$C87504))
#Codigo pais adqiriente
barplot(table(trxclus3$C87543))
#Condicion pto venta
barplot(table(trxclus3$C87512))
#Pais origen
barplot(table(trxclus3$C87519))
#marca o franquicia
barplot(table(trxclus3$C87566))
#credito o debito
barplot(table(trxclus3$C87586))
#grupo dia
barplot(table(trxclus3$C87594))


# Una forma de determinar un numero apropiado de clusters es validar la
# reducción en la variacion total que se logra al incrementar un cluster

# inicializamos la variable que va a guardar la variacion total para cada
# modelo

vtot <- 0

# generamos un modelo para cada numero de clusters desde 1 hasta 8
# y para cada uno, guardamos la variacion total intracluster

for (i in 1:8) {mod <- kmeans(trx2,i);
vtot[i] <- mod$tot.withinss}

# ploteamos la vtot y el # de clusters y vemos la grafica

plot(1:8, vtot, type="b", xlab="num clusters", ylab= "var tot intracluster")
```

H. Script utilizado para regresión logística

```
# Modelo de Regresion Logistica
setwd("C:/Users/Berny/Desktop/corridamegaproyecto")
# Leer los datos
transaccion = read.csv("trxfraude2.csv")

# Examinar la estructura
str(transaccion)
fraude=as.numeric(transaccion$C87601=="F")

# Tabular la variable respuesta
table(fraude)
# La linea base para esto es el de mayor frecuencia, por lo que

99880/(99880+1271)

# libreria para hacer el split de train/test

library(caTools)

# Hacer un split aleatorio de los datos, garantizando reproducibilidad
set.seed(123)
split = sample.split(fraude, SplitRatio = 0.75)

# Crear sets de datos para entrenamiento y validacion
tranTrain = subset(transaccion, split == TRUE)
tranTest = subset(transaccion, split == FALSE)

# Modelo de regresion logistica
tranmod = glm(C87601 ~ .-C87538-C87586-C87586-C87519-C87561, data=tranTrain,
family=binomial)
summary(tranmod)

# Hacer predicciones con el modelo utilizando los datos de entrenamiento
predictTrain = predict(tranmod, type="response")

# Analizar los resultados de las predicciones
summary(predictTrain)
tapply(predictTrain, tranTrain$C87601, mean)


# Matriz de Confusion para un umbral de 0.7
confumat<- table(tranTrain$C87601, predictTrain > 0.7)

# Sensitividad, especificidad y precison del modelo
sensi <- confumat[2,2]/sum(confumat[2,])
especi <- confumat[1,1]/sum(confumat[1,])
preci <- sum(diag(confumat))/sum(confumat)
sensi
```

```
especi
preci

#---------------------------------------RECORTE DE DATOS SOLO FRAUDE-----------------------
--------------
trx<- read.csv("trx_allfields2.csv")
trxfraude<- trx[trx$C87601=="F",]
write.csv(trxfraude,"trxfraude.csv")

#----------------------MODELO 2-----------------------------
# Modelo de regresion logistica
tranmod2 = glm(C87601 ~ .-C87538-C87586-C87586-C87519-C87561-C87714,
data=tranTrain, family=binomial)
summary(tranmod2)

# Hacer predicciones con el modelo utilizando los datos de entrenamiento
predictTrain2 = predict(tranmod2, type="response")


# Matriz de Confusion para un umbral de 0.7
confumat2<- table(tranTrain$C87601, predictTrain2 > 0.7)

# Sensitividad, especificidad y precison del modelo
sensi2 <- confumat2[2,2]/sum(confumat2[2,])
especi2 <- confumat2[1,1]/sum(confumat2[1,])
sensi2
especi2
preci2 <- sum(diag(confumat2))/sum(confumat2)
preci2

#----------------------MODELO 3-----------------------------
# Modelo de regresion logistica
tranmod3 = glm(C87601 ~ .-C87538-C87586-C87586-C87519-C87561-C87714-C87550-
C87543, data=tranTrain, family=binomial)
summary(tranmod3)

# Hacer predicciones con el modelo utilizando los datos de entrenamiento
predictTrain3 = predict(tranmod3, type="response")


# Matriz de Confusion para un umbral de 0.7
confumat3<- table(tranTrain$C87601, predictTrain3 > 0.7)

sensi3 <- confumat3[2,2]/sum(confumat3[2,])
sensi3
especi3 <- confumat3[1,1]/sum(confumat3[1,])
especi3
# Sensitividad, especificidad y precison del modelo
preci3 <- sum(diag(confumat3))/sum(confumat3)
preci3
```

```
#----------------------MODELO 4-----------------------------
# Modelo de regresion logistica
tranmod4 = glm(C87601 ~ .-C87538-C87586-C87586-C87519-C87561-C87714-C87550-
C87543-C87675, data=tranTrain, family=binomial)
summary(tranmod4)

# Hacer predicciones con el modelo utilizando los datos de entrenamiento
predictTrain4 = predict(tranmod4, type="response")


# Matriz de Confusion para un umbral de 0.7
confumat4<- table(tranTrain$C87601, predictTrain4 > 0.7)

sensi4 <- confumat4[2,2]/sum(confumat4[2,])
sensi4
especi4 <- confumat4[1,1]/sum(confumat4[1,])
especi4
preci4 <- sum(diag(confumat4))/sum(confumat4)
preci4

#----------------------MODELO 5-----------------------------
# Modelo de regresion logistica
tranmod5 = glm(C87601 ~ .-C87538-C87586-C87586-C87519-C87561-C87550-C87504-
C87543-C87567, data=tranTrain, family=binomial)
summary(tranmod5)

# Hacer predicciones con el modelo utilizando los datos de entrenamiento
predictTrain5 = predict(tranmod5, type="response")


# Matriz de Confusion para un umbral de 0.7
confumat5<- table(tranTrain$C87601, predictTrain5 > 0.7)

sensi5 <- confumat5[2,2]/sum(confumat5[2,])
sensi5
especi5 <- confumat5[1,1]/sum(confumat5[1,])
especi5
preci5 <- sum(diag(confumat5))/sum(confumat5)
preci5

#----------------------MODELO 6-----------------------------
# Modelo de regresion logistica
tranmod6 = glm(C87601 ~ .-C87714-C87566-C87538-C87586-C87586-C87519-C87561-
C87550-C87504-C87543-C87567-C87500-C87511-C87531-C87675-C87535, data=tranTrain,
family=binomial)
summary(tranmod6)

# Hacer predicciones con el modelo utilizando los datos de entrenamiento
predictTrain6 = predict(tranmod6, type="response")
```

```
# Matriz de Confusion para un umbral de 0.7
confumat6<- table(tranTrain$C87601, predictTrain6 > 0.7)

sensi6 <- confumat6[2,2]/sum(confumat6[2,])
sensi6
especi6 <- confumat6[1,1]/sum(confumat6[1,])
especi6
preci6 <- sum(diag(confumat6))/sum(confumat6)
preci6
#----------------------MODELO 7-----------------------------
# Modelo de regresion logistica
tranmod7 = glm(C87601 ~ .-C87507-C87584-C87714-C87566-C87538-C87586-C87586-
C87519-C87561-C87550-C87504-C87543-C87567-C87500-C87511-C87531-C87675-C87535,
data=tranTrain, family=binomial)
summary(tranmod7)

# Hacer predicciones con el modelo utilizando los datos de entrenamiento
predictTrain7 = predict(tranmod7, type="response")


# Matriz de Confusion para un umbral de 0.7
confumat7<- table(tranTrain$C87601, predictTrain7 > 0.7)

sensi7<- confumat7[2,2]/sum(confumat7[2,])
sensi7
especi7<- confumat7[1,1]/sum(confumat7[1,])
especi7
preci7<- sum(diag(confumat7))/sum(confumat7)
preci7
```

I. Acta de iniciación del proyecto

Universidad del Valle de Guatemala
Julio de 2013

Propuesta de megaproyectos 2013

Información del Megaproyecto

- **Nombre del megaproyecto:**
  - Sistemas Inteligentes para reconocimiento de Patrones de comportamiento transaccionales. Se utilizarán en sistemas de detección de operaciones fraudulentas en ambientes financieros.

- **Objetivos**
  - **Objetivo general:**
    - Crear sistemas inteligentes, capaces de reconocer patrones de comportamiento en sistemas transaccionales, segmentación por patrones y sugerir reglas de detección de patrones específicos.

  - **Objetivos específicos:**
    - Reconocimiento de patrones de comportamiento
    - Segmentación por patrones transaccionales
    - Detección de cambios en la conducta de transaccionalidad.
    - Aprendizaje del modelo en base a los cambios de conducta de los clientes que son confirmados como correctos.

- **Descripción:**
  - Uso de las diferentes tecnologías, técnicas y algoritmos de inteligencia artificial y minería de datos, para la creación de sistemas inteligentes capaces de reconocer patrones de comportamiento, y validación de segmentación transaccional con el objetivo de detectar inusualidades de conducta en el cliente y generar un proceso de investigación para descartar que se trate de operaciones sospechosas.

- **Equipo sugerido:**
  - 4 estudiantes aproximadamente de Ingeniería de Ciencias de la Computación.
  - 2 estudiantes de Ingeniería en Ciencias de la Administración.

- Podrá brindarse una beca de US$ xxx.00 a cada estudiante

- Departamento que coordinará el megaproyecto: Ciencias de la Computación.

Contacto

- Nombre del responsable
  Mario Roberto González

- Correo electrónico
  mgonzalez@plusti.com

- Teléfono
  (502) 2383-1616 ext 670

J. Ejemplo de minutas realizadas durante fase de definición del proyecto (Reuniones UVG-plusTi)

Minuta de Reunión
MEGAPROYECTO UVG

REUNIÓN No. 2
FECHA: Viernes 2 de agosto de 2013.
HORA: 14:00
LUGAR:

Oficinas Plus Technologies

OBJETIVOS DE LA REUNIÓN:

1. Conocer el funcionamiento del sistema actual.
   - Estadísticas de funcionamiento actual del sistema (tasas de detección actual, tasas de falsos positivos, etc.).
   - Dimensiones fundamentales del sistema (alcance, tiempo, disponibilidad).

2. Determinar las prioridades y principales necesidades de la empresa.

PUNTOS A DISCUTIR:

Diferentes técnicas del sistema de Monitor Plus para detección de fraudes.

a. Se utilizan actualmente cinco diferentes técnicas en los sistemas, las cuales son:
   1) Reglas adaptivas.
   2) Secuencias.
   3) Scoring.
   4) Factores de riesgo.
   5) Redes neurales

b. Conocer a personal de diferentes departamentos de Monitor Plus, los cuales estarán formando parte del equipo de apoyo de la empresa para este Megaproyecto.

COMENTARIOS:

Los señores de la empresa mencionaron su interés por enfocar el megaproyecto en mejorar:

1. El modelo actual de redes neurales, dado que es la técnica menos desarrollada entre las que utilizan actualmente.
2. Enfocarnos en nuevas posibles soluciones o mejoras.

CONCLUSIONES:

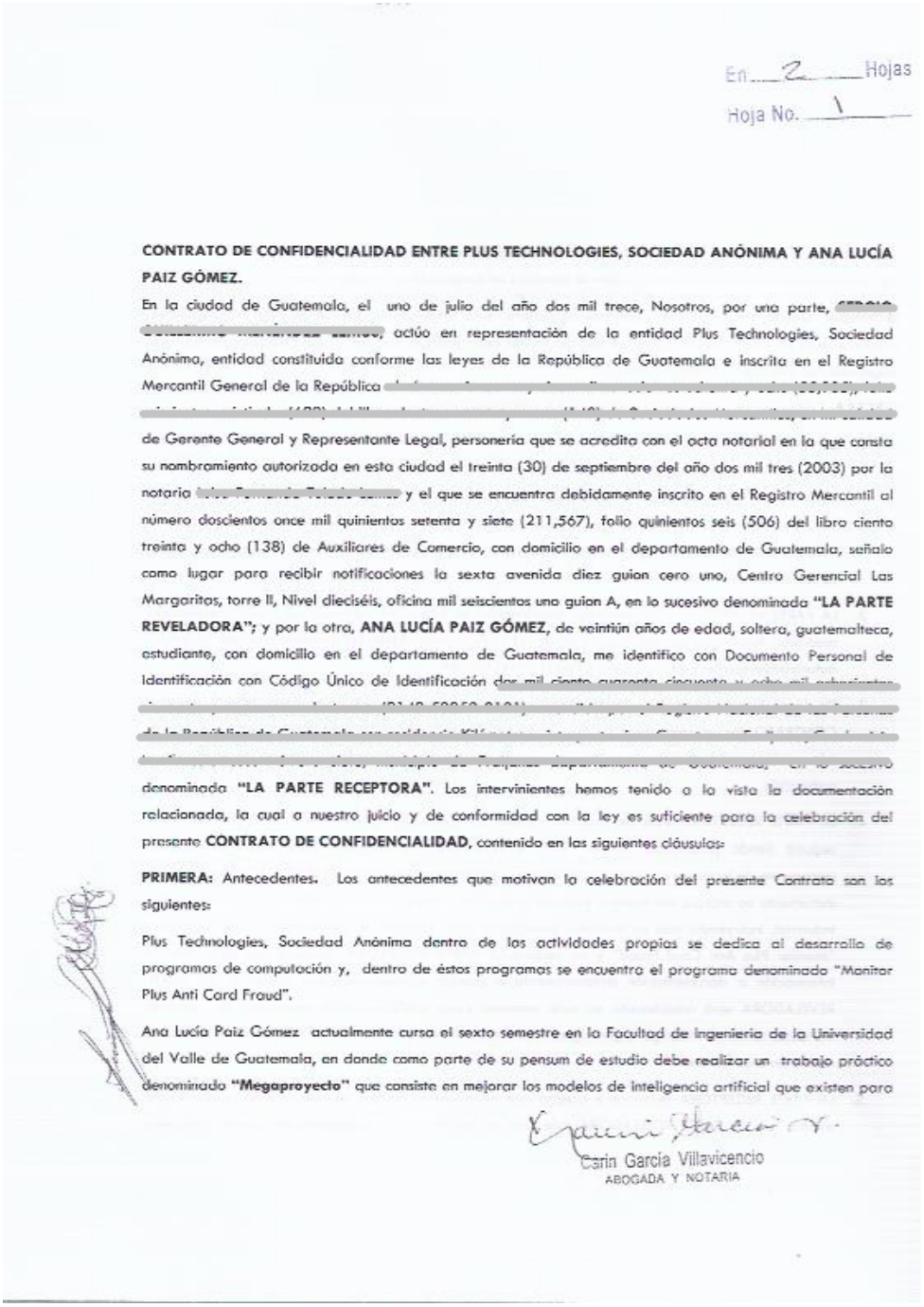
En esta segunda reunión de trabajo de Megaproyecto, se acordaron diversos puntos importantes y se llegaron a las siguientes conclusiones:

1. Enfocar el Megaproyecto en 3 área principales:
   a. Proceso de segmentación (conocer a los clientes).
   b. Detectar nuevas reglas de prevención.
   c. Aumentar el potencial de las tecnologías actuales.

2. La siguiente reunión (No. 3), tendría el objetivo de reunirse (virtualmente) con una de las personas que estuvo directamente relacionado con el diseño del modelo de Redes Neurales con el que cuenta actualmente la empresa, a fin de conocer el alcance del mismo.

3. Queda pendiente la fecha y lugar de reunión, hasta poder gestionar con la Universidad del Valle de Guatemala un día asignado para poder realizar estas reuniones y posteriormente confirmar disponibilidad con el personal de Monitor Plus.

K. Contrato de confidencialidad



En 2 Hojas

Hoja No. 1

CONTRATO DE CONFIDENCIALIDAD ENTRE PLUS TECHNOLOGIES, SOCIEDAD ANÓNIMA Y ANA LUCÍA PAIZ GÓMEZ.

En la ciudad de Guatemala, el uno de julio del año dos mil trece, Nosotros, por una parte, [redacted], actúo en representación de la entidad Plus Technologies, Sociedad Anónima, entidad constituida conforme las leyes de la República de Guatemala e inscrita en el Registro Mercantil General de la República [redacted] de Gerente General y Representante Legal, personería que se acredita con el acta notarial en la que consta su nombramiento autorizado en esta ciudad el treinta (30) de septiembre del año dos mil tres (2003) por la notaria [redacted] y el que se encuentra debidamente inscrito en el Registro Mercantil al número doscientos once mil quinientos setenta y siete (211,567), folio quinientos seis (506) del libro ciento treinta y ocho (138) de Auxiliares de Comercio, con domicilio en el departamento de Guatemala, señalo como lugar para recibir notificaciones la sexta avenida diez guion cero uno, Centro Gerencial Las Margaritas, torre II, Nivel dieciséis, oficina mil seiscientos uno guion A, en lo sucesivo denominada **"LA PARTE REVELADORA"**; y por la otra, **ANA LUCÍA PAIZ GÓMEZ**, de veintiún años de edad, soltera, guatemalteca, estudiante, con domicilio en el departamento de Guatemala, me identifico con Documento Personal de Identificación con Código Único de Identificación [redacted] denominada **"LA PARTE RECEPTORA"**. Los intervinientes hemos tenido a la vista la documentación relacionada, la cual a nuestro juicio y de conformidad con la ley es suficiente para la celebración del presente **CONTRATO DE CONFIDENCIALIDAD**, contenido en las siguientes cláusulas:

PRIMERA: Antecedentes. Los antecedentes que motivan la celebración del presente Contrato son los siguientes:

Plus Technologies, Sociedad Anónima dentro de las actividades propias se dedica al desarrollo de programas de computación y, dentro de éstos programas se encuentra el programa denominado "Monitor Plus Anti Card Fraud".

Ana Lucía Paiz Gómez actualmente cursa el sexto semestre en la Facultad de Ingeniería de la Universidad del Valle de Guatemala, en donde como parte de su pensum de estudio debe realizar un trabajo práctico denominado **"Megaproyecto"** que consiste en mejorar los modelos de inteligencia artificial que existen para

Carin García Villavicencio
ABOGADA Y NOTARIA

detectar de una mejor forma el fraude bancario en tarjetas de crédito, por lo que ha solicitado realizarlo en Plus Technologies, Sociedad Anónima a quien le quedará en propiedad los modelos creados.

SEGUNDA: Objeto. El presente contrato se refiere a la información que LA PARTE REVELADORA proporcione a LA PARTE RECEPTORA, ya sea de forma oral, grafica, escrita o por cualquier medio, que tenga relación o no con el Megaproyecto.

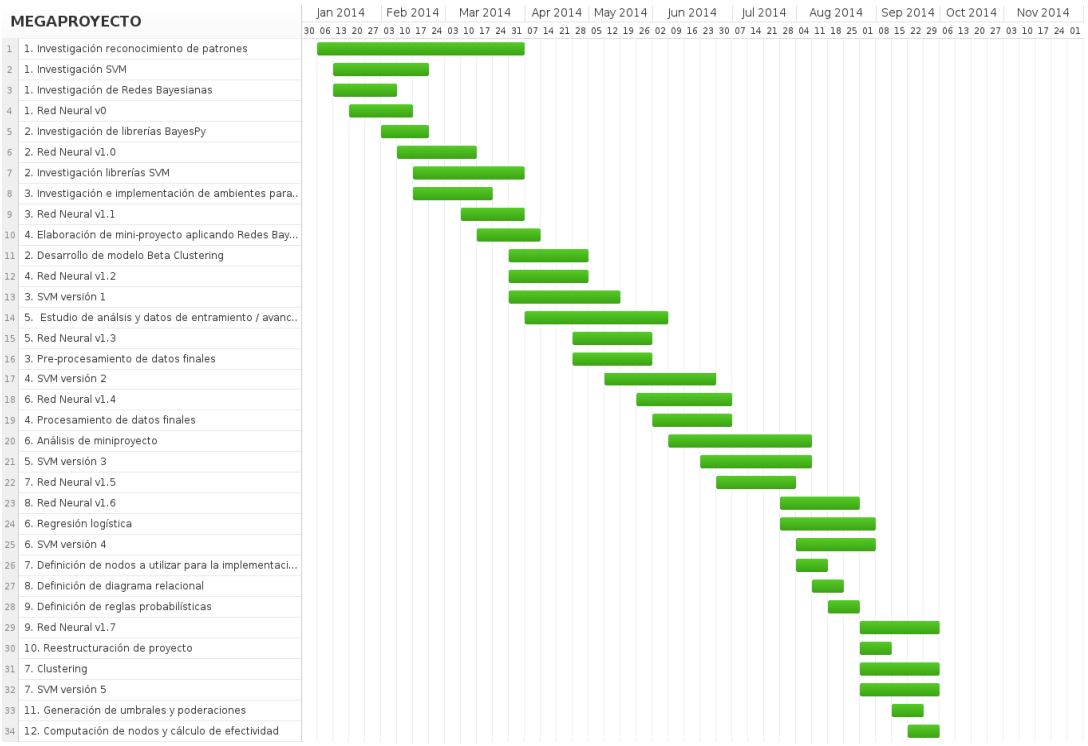
TERCERA: Obligaciones y Responsabilidades de LA PARTE RECEPTORA:

1. LA PARTE RECEPTORA únicamente utilizará la información facilitada por LA PARTE REVELADORA para el fin mencionado en la cláusula segunda de éste contrato, comprometiéndose LA PARTE RECEPTORA a mantener la más estricta confidencialidad respecto de dicha información, advirtiendo de dicho deber de confidencialidad y secreto a sus catedráticos o a cualquier persona que, por su relación con LA PARTE RECEPTORA, deba tener acceso a dicha información para el correcto cumplimiento del Megaproyecto.
2. LA PARTE RECEPTORA o las personas mencionadas en el párrafo anterior no podrán reproducir, modificar, hacer pública o divulgar a terceros la información objeto del presente contrato sin previa autorización escrita y expresa de LA PARTE REVELADORA.
3. De igual forma, LA PARTE RECEPTORA adoptará respecto de la información objeto de este CONTRATO las mismas medidas de seguridad que adoptaría normalmente respecto a la información confidencial de su propiedad, evitando en la medida de lo posible su pérdida, robo o sustracción.
4. LA PARTE RECEPTORA está de acuerdo en que la información de LA PARTE REVELADORA es y seguirá siendo propiedad de LA PARTE REVELADORA; se obliga a usar dicha información únicamente de la manera y para los propósitos autorizados por LA PARTE REVELADORA, y que este instrumento no otorga, de manera expresa e implícita, ningún derecho intelectual o de propiedad industrial, incluyendo, más no limitado, licencias de uso, respecto de la información del programa "Monitor Plus Anti Card Fraud" y sus mejoras.LA PARTE RECEPTORA reconoce que todo tipo de información o documentación proporcionada o puesta a disposición por parte de LA PARTE REVELADORA será considerada en todo momento como confidencial, sin necesidad de que esté marcada como tal o tenga algún signo o marca distintiva, por lo que LA PARTE RECEPTORA las recibe en esos términos.
5. LA PARTE RECEPTORA reconoce y acepta que el revelar o utilizar sin autorización previa y por escrito de LA PARTE REVELADORA este tipo de información o documentación pueda ocasionar

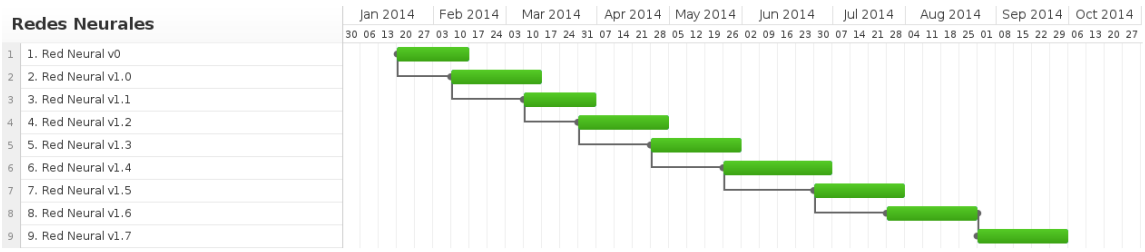
L. Control del tiempo - cronogramas

Los Gantt finales (general y personalizados) para las etapas de ejecución y seguimiento y control fueron los siguientes:

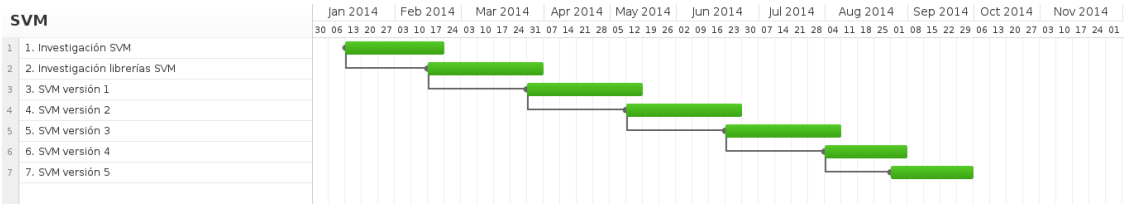
Gantt Megaproyecto fase de ejecución



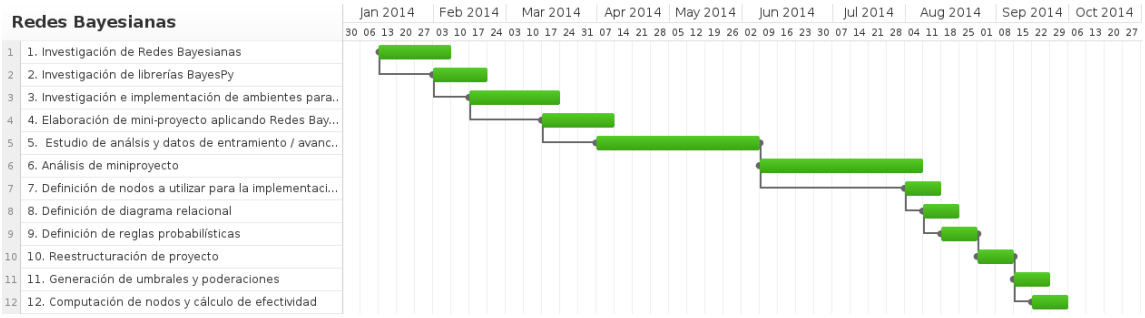
Gantt módulo Redes neurales



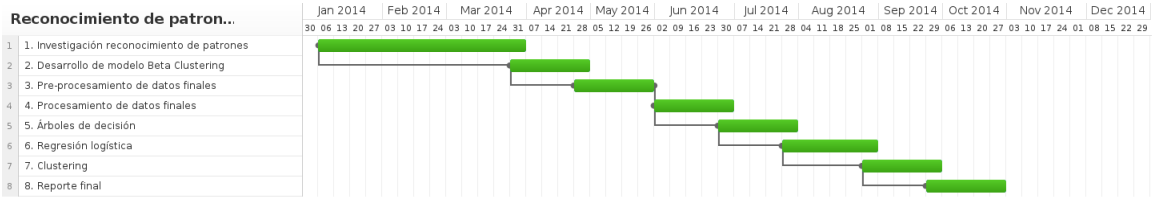
Gantt módulo SVM



Gantt módulo Redes Bayesianas

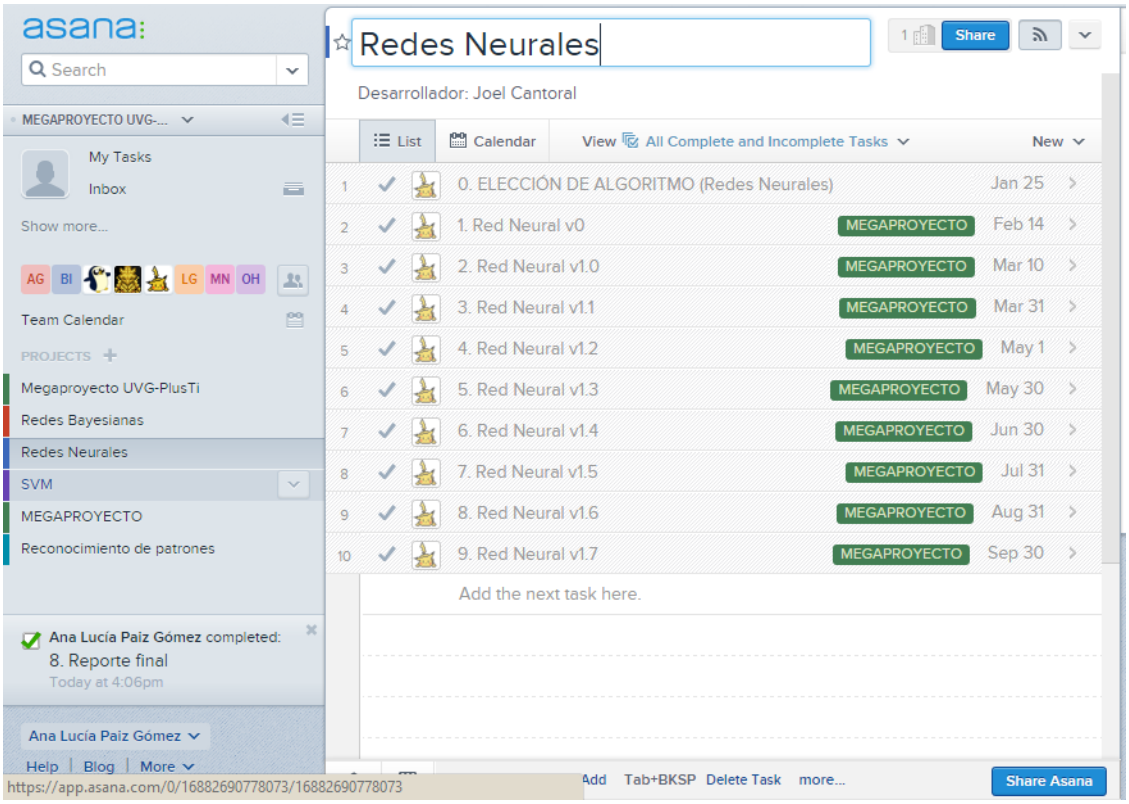


Gantt módulo de Reconocimiento de patrones

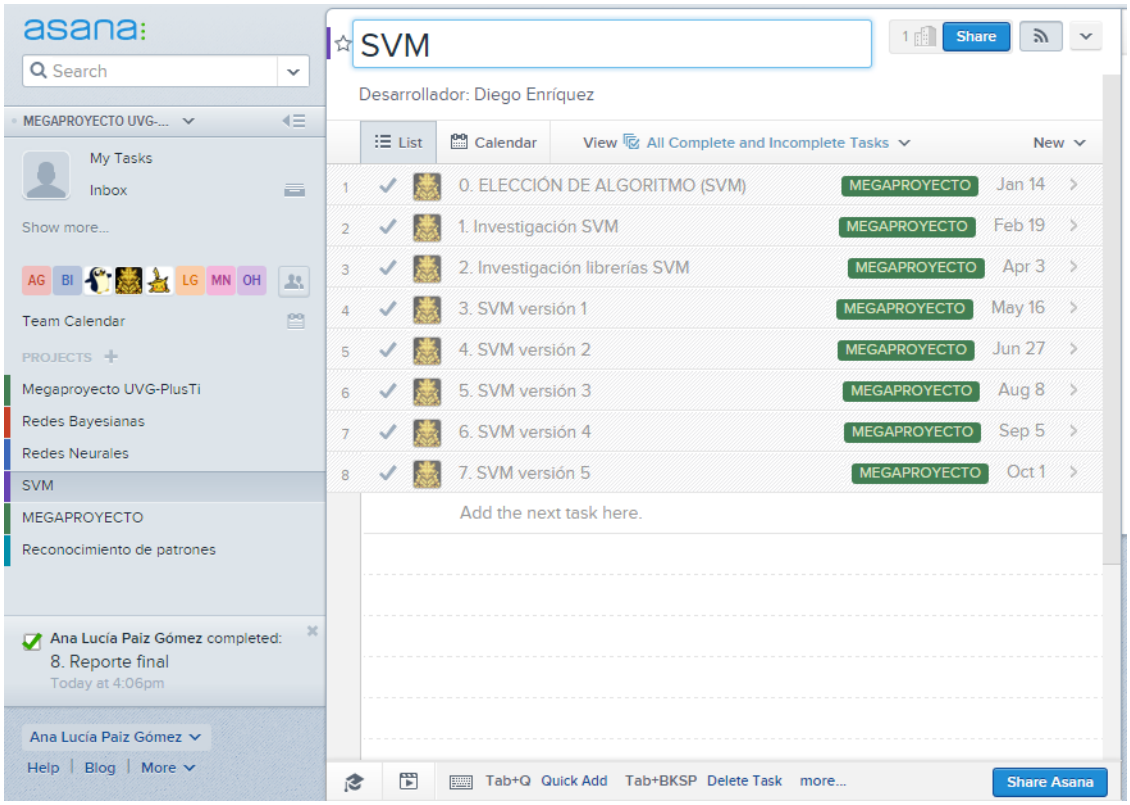


M. Evidencia de proyecto gestionado por medio de Asana

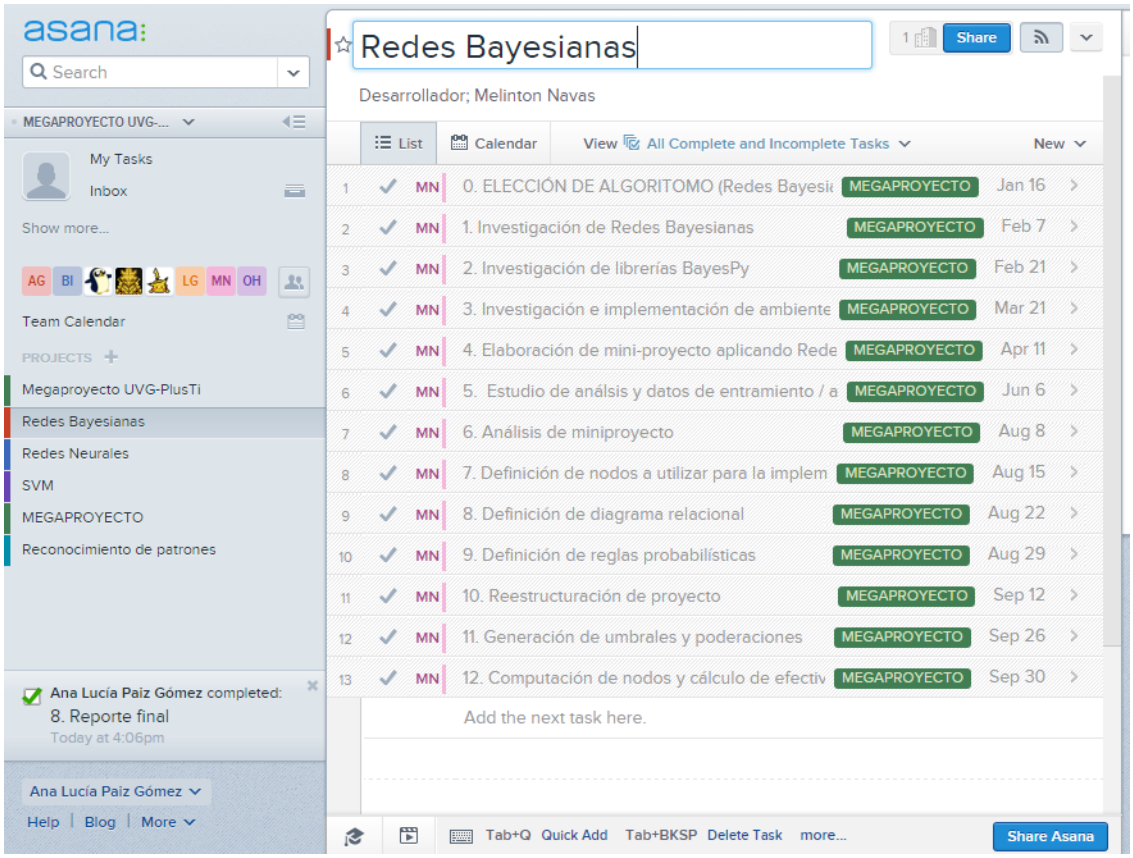
Evidencia de módulo de Redes neurales en Asana



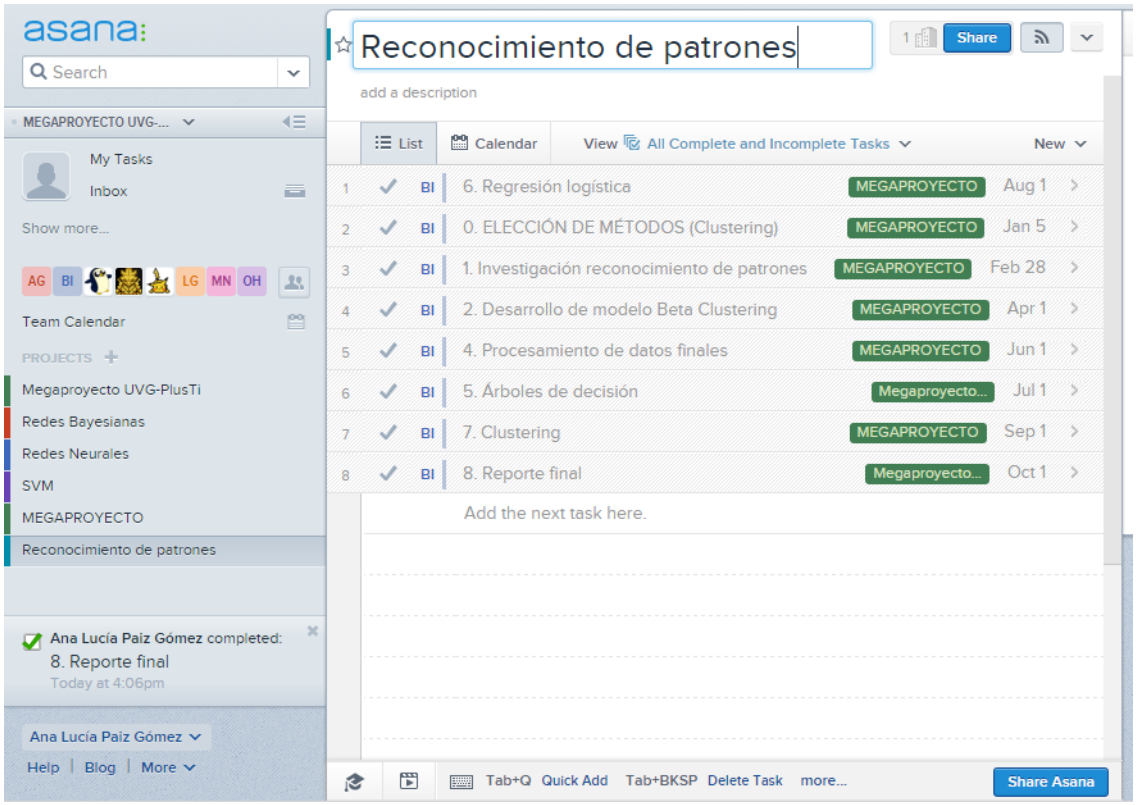
Evidencia de módulo de SVM en Asana



Evidencia de módulo de Redes Bayesianas en Asana



Evidencia de módulo de Reconocimiento de patrones en Asana



N. Plantillla formulario de información general de algoritmo o modelo a desarrollar

PLANTILLA para
Información inicial general

Lenguaje de programación utilizado:	

Justificación de su elección:

(en base a ventajas y desventajas en comparación con otros lenguajes)

Tema de investigación:	

Justificación de su elección:

(Explicación del por qué se considera útil investigar más sobre este enfoque como posible solución a la problemática actual detectada).

Ñ. Plantillla formulario de reporte de fase terminada en los algoritmos y modelos desarrollados.

PLANTILLA para
Reporte fase terminada

Desarrollador:	

Algoritmo:	

Fase actual:	*(ej. red neural versión 2)*
Fase predecesora:	*(ej. red neural versión 1)*
Fase sucesora:	*(ej. red neural versión 3)*

Fecha de desarrollo:	*(de… a…)*

Objetivo(s) de la fase actual:

Resultado(s) obtenidos:

Objetivo(s) de la siguiente fase:

(Acciones a tomar en base a lo obtenido en presente fase)

Detalle de fase actual:

- *Duración de la fase (x semanas, o x días)*
- *Desglose de actividades realizadas durante la fase con duración aproximada de cada una (ser lo más específico posible)*
- *Recursos utilizados*
- *Costo total de la fase*
- *Información adicional (en caso haya alguna sugerencia, cambio importante inesperado realizado, etc.)*

Detalle de la siguiente fase:

- *Duración aproximada*
- *Recursos requeridos*
- *Posible costo total de la fase*
- *Información adicional relevante*

O. Formularios de información general y fases terminadas de redes neurales

Información general

Lenguaje de programación utilizado:	Python

Justificación de su elección:

Se utilizará el lenguaje de programación python, en especifico con la librería de PyBrain, luego de haber realizado una comparación con la segunda alternativa escrita en el lenguaje Javascript, llamada BrainJS. BrainJS es una librería que permite crear y utilizar una red neuronal en el explorador web, a través del compilador de javascript, y en el sistema operativo de una computadora, a través del uso de NodeJS con el compilador V8 utilizado para Google Chrome. La librería de python, PyBrain, contiene una mayor cantidad de documentación, contiene secciones modulares en donde se puede agregar código que se ajuste mejor a las características del problema y cuenta con un equipo de desarrollo más extenso que su contraparte en javascript. Gracias a estas características que le otorgan visibilidad al proyecto se puede asegurar que PyBrain es una solución más robusta y duradera para llevar a cabo la implementación de la red neuronal.

Tema de investigación:	Redes neuronales

Justificación de su elección:

Las redes neuronales son técnicas de aprendizaje de máquinas útiles para la resolución de problemas en donde la metodología no es explícita. Funciona al encontrar patrones que se ajusten a cada dimensión en los datos por medio de algoritmos de entrenamiento como el proceso de propagación hacia atrás (*backpropagation*) que corrige el error de la data de ingreso con respecto al resultado esperado. Técnicas como la modularización de redes neuronales pueden ayudar a reducir tiempos de entrenamiento y a especializar segmentos de una red neuronal con respecto a una dimensión del problema. Esto presenta un campo de estudio el desarrollo de arquitectura de redes neuronales por módulos y su entrenamiento y uso distribuido para la mejora en la efectividad de los modelos de predicción. Esta mejora en la efectividad y la adición de un análisis de preprocesamiento sobre la información a evaluar, a través de ténicas como análisis de importancia de las variables que conforman los nodos de entrada, puede llegar a mejorar los resultados de detección de fraudes en base a la data histórica.

Reporte fase terminada # 1

Desarrollador:	**Joel Cantoral**

Algoritmo:	**Red neuronal**

Fase actual:	*Red neuronal v0*
Fase predecesora:	*N/A*
Fase sucesora:	**Red neuronal v1**

Fecha de desarrollo:	*26/01/2014 hasta 14/02/2014*

Objetivo(s) de la fase actual:

- Evaluar las tecnologías disponibles para modelar redes neuronales. Realizar la operación de reconocimiento de caracteres escritos para ver las funcionalidades que proveen las librerías.

Resultado(s) obtenidos:

- Al comparar PyBrain con BrainJS se puede observar que la librería en Javascript carece de funcionalidades importantes para modelar una red neuronal. No permite entrenar la red más de una vez, lo que limita el aprendizaje y la capacidad de mejorar resultados obtenidos con anterioridad. Utilizar Brain.js a través de NodeJS limita la capacidad de ejecución por la naturaleza de un sólo hilo de ejecución del software. Esto implica que el procesamiento es mucho más intensivo utilizando sistemas en NodeJS y esto puede afectar la evaluación en tiempo real de los resultados del análisis.

Objetivo(s) de la siguiente fase:

- En esta fase se identificó que la red que debe utilizarse para la creación del modelo es PyBrain. En la próxima fase se describe la configuración de la red neuronal.

Detalle de fase actual:

- Duración de la fase: 20 días
- Desglose de actividades realizadas durante la fase
- Selección de herramientas y librerías - 7 días
- Selección de Licencia - 2 días
- Definición de métricas objetivo - 2 días
- Implementación de red neuronal con BrainJS - 7 días
- Revisión de resultados - 2 días
- Recursos utilizados
- Laptop
- Documentación de las librerías
- Libros de referencia
- (Q30*4 horas/día*20 días) = Q 2,400.00
- Información adicional
- Esta fase del proyecto permitió establecer qué librería sería la más indicada para la red neuronal.

Detalle de la siguiente fase:

- Duración aproximada - 23 días
- Recursos requeridos
    - Laptop
    - Información transaccional (Formato de transacción)
- (Q30*4 horas/día*23 días) = Q 2,760.00

Reporte fase terminada # 2

Desarrollador:	**Joel Cantoral**

Algoritmo:	**Red neuronal**

Fase actual:	*Red neuronal v1*
Fase predecesora:	*Red neural v0*
Fase sucesora:	**Red neuronal v1.1**

Fecha de desarrollo:	*15/02/2014 hasta 10/03/2014*

Objetivo(s) de la fase actual:

- Identificar un modelo de red neuronal que permita realiza análisis sobre los datos transaccionales de compras electrónicas utilizando la librería PyBrain.

Resultado(s) obtenidos:

La estructura de la red neuronal se compone de los siguientes nodos de entrada:

- **VWJEFECHAD** - Día de Envío de la TRX desde la interfase
- **VWJEFECHAM** - Mes de Envío de la TRX desde la interfase
- **VWJEFECHAA** - Año de Envío de la TRX desde la interfase
- **VWJEHORA** - Hora de Envío de la TRX desde la interfase
- **SISTEMINUTE** - Minuto de Envío de la TRX desde la interfase
- **87500** - Llave primaria de control (Tarjeta o ID Cliente)
- **87550** - Código de transacción (Tipo de transacción) (Numérico (2))
- **87502** - Indicador de Reverso (N o S)
- **87503** - Monto original de la transacción, Moneda adquirente
- **87506** - Hora de la transacción, hora del adquirente
- **87510** - Merchant Category Code, actividad a la que se dedica el comercio
- **87543** - Código de País institución adquirente (Comercio o ATM)
- **87544** - Código de País institución emisor (Emisor de la Tarjeta)
- **87511** - Modo de entrada del punto de servicio
- **87512** - Código de condición en el punto de venta
- **87514** - Número de autorización
- **87519** - País donde se origino la TRX
- **87567** - Tipo de producto tarjeta (Platinum, Gold, Empresarial, etc.)
- **87535** - Bin (Bank Identification Number)

- **87580** - Código banco origen (Código de compensación local)
- **87584** - Número de identificación cliente
- **87530** - Identificación comercio
- **87566** - Marca o franquicia tarjeta
- **87593** - Semana del año
- **87594** - Día Feriado (Alfanumérico (2))
- **87596** - Clasificador de comercio

Objetivo(s) de la siguiente fase:

Normalizar la información de entrada a la red neuronal para poder manejar resultados más estándares y darle una proporción significativa a los la información de entrenamiento.

Detalle de fase actual:

- Duración de la fase - 23 días
- Desglose de actividades realizadas durante la fase
    - Definición del modelo básico de la red neuronal
    - Selección de nodos de entrada de la data transaccional
    - Selección de técnicas de preprocesamiento por cada nodo de entrada
- Recursos utilizados
    - Laptop
    - Información transaccional (Formato de transacción)
- (Q30*4 horas/día*23 días) = Q 2,760.00
- Información adicional:
    - Hay datos que necesitan ser normalizados y otros codificados en categorías para poder insertarlos como entradas a la red neuronal.

Detalle de la siguiente fase:

- Duración aproximada - 20 días
- Recursos requeridos
    - Laptop
    - Librerías para normalización (scikit)
    - Libros de referencia en estadística
- (Q30*4 horas/día*20 días) = Q 2,400.00

Reporte fase terminada # 3

Desarrollador:	**Joel Cantoral**

Algoritmo:	**Red neuronal**

Fase actual:	*Red neuronal v1.1*
Fase predecesora:	*Red neuronal v1.0*
Fase sucesora:	**Red neuronal v1.2**

Fecha de desarrollo:	*11/03/2014 hasta 31/03/2014*

Objetivo(s) de la fase actual:

- Esta fase continúa con los objetivos de la etapa anterior con la Red neuronal v1.1. Investigación sobre la librería para desarrollo de redes neuronales PyBrain, incluyendo funciones y técnicas utilizadas para mejorar los algoritmos desarrollados

Resultado(s) obtenidos:

- Se pueden agregar nuevas funciones de activación a la librería para modificar el valor de salida de los nodos de la red neuronal
- Se puede utilizar variaciones al método de *backpropagation* para evaluar la mejora en el algoritmo de la convergencia hacia un mejor resultado.

Objetivo(s) de la siguiente fase:

Normalizar la información de entrada a la red neuronal para poder manejar resultados más estándares y darle una proporción significativa a los la información de entrenamiento.

Detalle de fase actual:

- Duración de la fase - 20 días
- Desglose de actividades realizadas durante la fase
  - Revisión de la documentación de la librería
  - Revisión del código necesario para modificar la librería y agregar nuevas funcionalidades
  - Investigar sobre los conceptos asociados a las técnicas de optimización y modificación de redes neuronales que pueden ser añadidos a la librería
- Recursos utilizados
  - Laptop
  - Documentación librería PyBrain
- (Q30*4 horas/día*20 días) = Q 2,400.00
- Información adicional que tengan

Detalle de la siguiente fase:

- Duración aproximada - 20 días
- Recursos requeridos
  - Laptop
  - Librerías para normalización (scikit)
  - Libros de referencia en estadística
- (Q30*4 horas/día*21 días) = Q 2,520.00

Reporte fase terminada # 4

Desarrollador:	**Joel Cantoral**

Algoritmo:	**Red neuronal**

Fase actual:	**Red neuronal v1.2**
Fase predecesora:	*Red neuronal v1.1*
Fase sucesora:	**Red neuronal v1.3**

Fecha de desarrollo:	*01/04/2014 hasta 01/05/2014*

Objetivo(s) de la fase actual:

- Definir arquitectura mínima de la red neuronal para tener un marco de referencia de preprocesamiento de la información
- Iniciar con el preprocesamiento de los datos para poder construir la arquitectura de la red neuronal.

Resultado(s) obtenidos:

- La arquitectura básica de la red neuronal se planteó como una red feedforward de múltiples capas que utiliza backpropagation para corregir el error de el conjunto de datos de prueba. Se estableció, como parte de la arquitectura de la red neuronal, las función de activación de los nodos de capas intermedias y la función de activación de los nodos de salida de la red neuronal. Estas funciones son: Sigmoid para capas intermedias y Heaviside desfasada 0.8 a la derecha en x para los nodos de salida.
- El preprocesamiento necesita acoplarse a la estructura de las funciones de activación para que la diferencia en los valores de entrada sea significativa en el resultado de aprendizaje de la red neuronal. Debido a que el rango óptimo de la función Sigmoid es de 0 a 1 en el eje x la data fue normalizada a este rango conservado la proporción del valor al rango anterior.

Objetivo(s) de la siguiente fase:

- Identificar qué variables tienen mayor peso sobre el resultado de la red neuronal para definir la arquitectura de la red neuronal.

Detalle de fase actual:

- Duración de la fase - 21 días
- Desglose de actividades realizadas durante la fase
  - Definición del arquitectura básica de la red neuronal
  - Selección de función de activación de capas intermedias y de salida
  - Obtención de datos estadísticos correspondientes a cada variable de la red neuronal
  - Normalización de los datos
- Recursos utilizados
  - Laptop
  - Información transaccional (Formato de transacción)
  - Software R Studio para análisis estadístico
- (Q30*4 horas/día*21 días) = Q 2,520.00
- Información adicional que tengan
  - Hay que evaluar si todos los campos definidos son importantes para el análisis, *i.e.*, si no hay duplicación de información.

Detalle de la siguiente fase:

- Duración aproximada - 30 días
- Recursos requeridos
  - Laptop
  - Software estadístico R Studio
  - Literatura sobre preprocesamiento de datos para estadística
- (Q25*4 horas/día*29 días) = Q 2,900.00

Reporte fase terminada # 5

Desarrollador:	**Joel Cantoral**

Algoritmo:	**Red neuronal**

Fase actual:	**Red neuronal v1.3**
Fase predecesora:	*Red neuronal v1.2*
Fase sucesora:	**Red neuronal v1.4**

Fecha de desarrollo:	*01/05/2014 hasta 30/05/2014*

Objetivo(s) de la fase actual:

- Investigar sobre aplicaciones de redes neuronales en el análisis predictivo para seleccionar la arquitectura completa final

Resultado(s) obtenidos:

- Se propone una arquitectura de red neuronal modular para disminuir el tiempo de ejecución necesario para poder procesar los datos en el entramiento de la red neuronal y aprovechar la especialización de los segmentos.
- La arquitectura final de la red neuronal se compone de 7 segmentos modulares de redes neuronales y 2 segmentos enfocados en el análisis filtrando transacciones correspondientes a un cliente individual. Estos módulos se conectan a una red neuronal final que se denomina Red de Decisiones (*DecisionNetwork*).

Objetivo(s) de la siguiente fase:

- Definición de escenarios en base a variables identificadas de importancia

Detalle de fase actual:

- Duración de la fase - 30 días
- Desglose de actividades realizadas durante la fase
  - Se identificaron las variables que podrían tener información duplicada y se eliminaron del conjunto de posibles nodos de entrada
  - Se realizó un análisis de importancia de variables para identificar una correlación entre nodos de entrada y el indicador de fraude
- Recursos utilizados
  - Laptop
  - Información transaccional (Formato de transacción)
  - Literatura y *papers* publicados sobre arquitecturas de redes neuronales
- (Q25*4 horas/día*29 días) = Q 2,900.00

Detalle de la siguiente fase:

- Duración aproximada - 30 días
- Recursos requeridos
  - Laptop
- (Q25*4 horas/día*30 días) = Q 3,000.00

Reporte fase terminada # 6

Desarrollador:	**Joel Cantoral**

Algoritmo:	**Red neuronal**

Fase actual:	**Red neuronal v1.4**
Fase predecesora:	*Red neuronal v1.3*
Fase sucesora:	**Red neuronal v1.5**

Fecha de desarrollo:	*30/05/2014 hasta 30/06/2014*

Objetivo(s) de la fase actual:

- Definición de escenarios en base a variables identificadas de importancia

Resultado(s) obtenidos:

- Los escenarios por módulo son los siguientes:
  - Escenario 1
    - Día de transacción
    - Mes de transacción
    - Hora de transacción
  - Escenario 2
    - Día de transacción
    - Mes de transacción
    - País de Comercio
    - País de Transacción
  - Escenario 3
    - Día de transacción
    - Mes de transacción
    - Marca de la Tarjeta
  - Escenario 4
    - Día de transacción
    - Mes de transacción
    - Bank identification number
  - Escenario 5
    - Día de transacción
    - Mes de transacción
    - Semana

- Es Feriado
- Escenario 6
  - Día de transacción
  - Mes de transacción
  - Tipo del producto
- Escenario 7
  - Día de transacción
  - Mes de transacción
  - Merchant Category Code
- Escenario 8
  - Día de transacción
  - Id del cliente
  - Monto de la transacción
- Escenario 9
  - Día de transacción
  - Id del cliente
  - País de comercio

Objetivo(s) de la siguiente fase:

- Entrenamiento preliminar con un conjunto de transacciones menor a la totalidad de datos de prueba

Detalle de fase actual:

- Duración de la fase - 31 días
- Desglose de actividades realizadas durante la fase
  - Se agruparon las variables de entrada en escenarios para generar la estructura modular de la red neuronal
- Recursos utilizados
  - Laptop
  - Información transaccional (Formato de transacción)
- (Q25*4 horas/día*30 días) = Q 3,000.00

Detalle de la siguiente fase:

- Duración aproximada - 31 días
- Recursos requeridos
  - Laptop
- (Q25*4 horas/día*31 días) = Q 3,100.00

Reporte fase terminada # 7

Desarrollador:	**Joel Cantoral**

Algoritmo:	**Red neuronal**

Fase actual:	**Red neuronal v1.5**
Fase predecesora:	*Red neuronal v1.4*
Fase sucesora:	**Red neuronal v1.6**

Fecha de desarrollo:	*01/07/2014 hasta 31/07/2014*

Objetivo(s) de la fase actual:

Entrenamiento preliminar con un conjunto de transacciones menor a la totalidad de datos de prueba

Resultado(s) obtenidos:

Se decidió realizar un proceso de validación cruzada (cross validation) utilizando las primeras 200,000 transacciones para la fase de entrenamiento utilizando un 70% para el entrenamiento y un 30% de validación.

Los resultados son los siguientes:

Datos con pruebas de 200,000 transacciones			
Escenario	% Trx correctas	% Trx incorrectas	Falsos positivos
Generales			
Bank Identification Number	99.99916295	0.0008370547392	0:10
Hora de la Trx	99.99970496	0.0002950435632	0:10
Tipo de producto de la tarjeta	100	0	0:10
País del comercio	85.74971229	14.25028771	1.4:10
Merchant Category Code	100	0	0:10
¿Es feriado?	99.70535673	0.2946432682	0:10
			Por cliente
Monto	90.58823529	9.411764706	0.67:10
País del comercio	96.47058824	3.529411765	0:10
Red de decisiones			
Evaluación de fraude	93.56396854	6.436031461	0:10

Objetivo(s) de la siguiente fase:

- Entrenamiento con la totalidad de los datos utilizando el mismo esquema de validación cruzada

Detalle de fase actual:

- Duración de la fase - 31 días
- Desglose de actividades realizadas durante la fase
  - Se entrenó cada uno de los módulos y luego la red de decisiones para obtener resultados de ejecución
- Recursos utilizados
  - Laptop
  - Información transaccional (Formato de transacción)
- Posible precio - (Q25*4 horas/día*31 días) = Q 3,100.00
- Información adicional que tengan
  - Se puede haber llegado a sobreajustar (*overfitting*) la red neuronal por entrenarla solamente con un segmento de todas las transacciones.

Detalle de la siguiente fase:

- Duración aproximada - 31 días
- Recursos requeridos
  - Laptop
- Posible precio - (Q30*4 horas/día*31 días) = Q 3,720.00

P. Formularios de información general y fases terminadas de SVM

Información general SVM

Lenguaje de programación utilizado:	Python

Justificación de su elección:

Se utilizó Python por la disponibilidad de documentación, librerías científicas y la facilidad para resolver problemas matemáticos a través de la sintaxis sencilla y el soporte de múltiples paradigmas de programación. Adicionalmente es un lenguaje que conozco, por lo que no será necesario aprenderlo. Existen varias implementaciones de svm en el lenguaje. Python brinda las ventajas de un lenguaje de alto nivel sin perder rendimiento, ya que librerías críticas están escritas en C.

Tema de investigación:	SVM

Justificación de su elección:

Se decidió utilizar la técnica de support vector machines principalmente porque estamos lidiando con un problema de clasificación, para lo cual están hechas las SVM. Adicionalmente es un problema de clasificación en dos grupos (fraudulentas y no fraudulentas) que son el caso para el que las svm fueron creadas. Por otro lado utilizando distintos kernels es posible probar distintas svm que puedan dar mejores resultados, lo que brinda flexibilidad.

Reporte fase terminada # 1

Desarrollador:	**Diego Enríquez**

Algoritmo:	**SVM**

Fase actual:	*Investigación SVM*
Fase predecesora:	**N/A**
Fase sucesora:	*Investigación librerías SVM*

Fecha de desarrollo:	*01/15/2014 a 02/19/2014*

Objetivo(s) de la fase actual:

- Investigar sobre SVM

Resultado(s) obtenidos:

- Se consolidaron los conocimientos sobre SVMs.

Objetivo(s) de la siguiente fase:

- Crear una SVM inicial capaz de resolver el problema.

Detalle de fase actual:

- Duración de la fase: 5 semanas
- Desglose de actividades realizadas
- Investigación de SVMs (5 semanas)
  - Computadora con sistema operativo Windows de 8gb de RAM y procesador core i7 de 3Ghz
- Posible precio: Q.2,500

Detalle de la siguiente fase:

- Duración aproximada: 6 semanas
- Recursos requeridos:
  - Computadora con sistema operativo Windows de 8gb de RAM y procesador core i7 de 3Ghz
- Q. 1,500.00

Reporte fase terminada # 2

Desarrollador:	**Diego Enríquez**

Algoritmo:	**SVM**

Fase actual:	**Investigación librerías SVM**
Fase predecesora:	*Investigación SVM*
Fase sucesora:	*SVM versión 1*

Fecha de desarrollo:	*02/20/2014 a 03/04/2014*

Objetivo(s) de la fase actual:

- Investigar sobre librerías de SVM
- Determinar el lenguaje en el que se realizará la SVM

Resultado(s) obtenidos:

- Se determinó que se utilizaría la librería libsvm en el lenguaje Python. Se investigaron otras librerías para utilizar en caso esta no funcionara correctamente. Se consolidaron los conocimientos sobre SVMs.

Objetivo(s) de la siguiente fase:

- Probar la librería determinada en la fase de investigación
- Determinar cambios que se tengan que realizar para mejorar la SVM

Detalle de fase actual:

- Duración de la fase: 6 semanas
- Desglose de actividades realizadas
  - Investigación de librerías (4 semanas)
  - Investigación de lenguajes (2 semanas)
- Recursos utilizados
  - Computadora con sistema operativo Windows de 8gb de RAM y procesador core i7 de 3Ghz
- Q.1,500.00

Detalle de la siguiente fase:

- Duración aproximada: 4 semanas
- Recursos requeridos:
  - Python
  - libSVM
  - Computadora con sistema operativo Windows de 8gb de RAM y procesador core i7 de 3Ghz
- Q.4,000.00

Reporte fase terminada # 3

Desarrollador:	**Diego Enríquez**

Algoritmo:	**SVM**

Fase actual:	*SVM versión 1*
Fase predecesora:	*Investigación librerías SVM*
Fase sucesora:	*SVM versión 2*

Fecha de desarrollo:	*De 04/04/2014 a 05/16/2014*

Objetivo(s) de la fase actual:

- Crear una SVM inicial capaz de resolver el problema.
- Probar la librería determinada en la fase de investigación
- Determinar cambios que se tengan que realizar para mejorar la SVM

Resultado(s) obtenidos:

- Se creó la SVM inicial, la librería utilizada no resultó óptima ya que se tardaba demasiado en entrenar con los datos obtenidos. Se determinó que era necesario cambiar la librería para mejorar los tiempos. Tenía un buen porcentaje de aciertos de 98%, pero una examinación más detallada mostró que no encontraba ninguna de las transacciones fraudulentas, que eran el 2% de desaciertos. En otras palabras la SVM tomaba todas las transacciones como no fraudulentas.

Objetivo(s) de la siguiente fase:

- Crear otra SVM que resuelva el problema, mejorando el tiempo.
- Utilizar una nueva librería.
- Determinar cambios que se tengan que realizar para mejorar la SVM

Detalle de fase actual:

- Duración de la fase: 6 semanas
- Desglose de actividades realizadas
  - Implementación de la SVM (3 semanas)
  - Entrenamiento (1 semana)
  - Pruebas (2 semanas)
- Recursos utilizados
- Q. 4,000.00

Detalle de la siguiente fase:

- Duración aproximada: 4 semanas
- Recursos requeridos: Python, Datos de Prueba, Librería de SVM
- Q. 4,000.00

Reporte fase terminada # 4

Desarrollador:	**Diego Enríquez**

Algoritmo:	**SVM**

Fase actual:	*SVM versión 2*
Fase predecesora:	*SVM versión 1*
Fase sucesora:	*SVM versión 3*

Fecha de desarrollo:	*De 5/17/2014 a 06/27/2014*

Objetivo(s) de la fase actual:

- Crear otra SVM que resuelva el problema, mejorando el tiempo.
- Utilizar una nueva librería.
- Determinar cambios que se tengan que realizar para mejorar la SVM
- Hacer que la SVM sea persistente

Resultado(s) obtenidos:

- Se creó la SVM, la librería utilizada fue una mejora considerable a la anterior (en cuanto a tiempo). Tenía un buen porcentaje de aciertos de 98%, pero un examen más detallado mostró que no encontraba ninguna de las transacciones fraudulentas, que eran el 2% de desaciertos. En otras palabras la SVM tomaba todas las transacciones como no fraudulentas. Examinando los datos utilizados, muchos de los campos estaban en blanco por lo que se atribuyó estos errores en los aciertos a los datos.

Objetivo(s) de la siguiente fase:

- Utilizar nuevos datos, que tengan todos los campos necesarios
- Determinar cambios que se tengan que realizar para mejorar la SVM
- Utilizar menos campos para mejorar la cantidad de falsos positivos

Detalle de fase actual:

- Duración de la fase: 6 semanas
- Desglose de actividades realizadas
  - Implementación de la SVM (3 semanas)
  - Entrenamiento (1 semana)
  - Pruebas (2 semanas)
- Recursos utilizados
  - Python
  - Scikit Learn
  - Computadora con sistema operativo Windows de 8gb de RAM y procesador core i7 de 3Ghz
- Q. 4,000.00

Detalle de la siguiente fase:

- Duración aproximada: 4 semanas
- Recursos requeridos:
  - Python
  - ScikitLearn
  - Computadora con sistema operativo Windows de 8gb de RAM y procesador core i7 de 3Ghz
- Q. 4,000.00

Reporte fase terminada # 5

Desarrollador:	**Diego Enríquez**

Fase actual:	*SVM versión 3*
Fase predecesora:	*SVM versión 2*
Fase sucesora:	*SVM versión 4*

Fecha de desarrollo:	*De 06/28/2014 a 08/08/2014*

Objetivo(s) de la fase actual:

- Utilizar nuevos datos, que tengan todos los campos necesarios
- Determinar cambios que se tengan que realizar para mejorar la SVM
- Utilizar menos campos para mejorar la cantidad de falsos positivos
- Crear una SVM que se entrene por lotes

Resultado(s) obtenidos:

- Debido a la cantidad de datos no fue posible realizar el entrenamiento en mi computadora personal. Por lo tanto se inició el proceso en el servidor de la UVG, sin embargo el procesamiento no se terminó porque el servidor se reiniciaba y se creó otra SVM. Para poder realizar el procesamiento en mi computadora fue necesario cambiar la forma de entrenarla para que se pudiera entrenar por lotes.
  La svm resultante no fue capaz de determinar cuáles eran las transacciones fraudulentas. Se eliminaron todos los datos que no tenían al menos un valor distinto para mejorar las predicciones. Se determinó que para mejorar los resultados era necesario mejorar el preprocesamiento que se estaba realizando.

Objetivo(s) de la siguiente fase:

- Utilizar otro tipo de preprocesamiento para conseguir que la SVM que sea capaz de identificar las transacciones fraudulentas.

Detalle de fase actual:

- Duración de la fase: 6 semanas
- Desglose de actividades realizadas
  - Implementación de la SVM (3 semanas)
  - Entrenamiento (2 semana)

- Pruebas (1 semana)
- Recursos utilizados
  - Python
  - ScikitLearn
  - Computadora con sistema operativo Windows de 8gb de RAM y procesador core i7 de 3Ghz
- Q.4,000.00

Detalle de la siguiente fase:

- Duración aproximada: 4 semanas
- Recursos requeridos:
  - Python
  - ScikitLearn
  - Computadora con sistema operativo Windows de 8gb de RAM y procesador core i7 de 3Ghz
- Q.4,000.00

Reporte fase terminada # 6

Desarrollador:	**Diego Enríquez**

Algoritmo:	**SVM**

Fase actual:	*SVM versión 4*
Fase predecesora:	*SVM versión 2*
Fase sucesora:	*SVM versión final*

Fecha de desarrollo:	*09/08/2014 a 09/05/2014*

Objetivo(s) de la fase actual:

- Utilizar otro tipo de preprocesamiento para conseguir que la SVM que sea capaz de identificar las transacciones fraudulentas.

Resultado(s) obtenidos:

- Utilizando otro tipo de preprocesamiento fue los resultados fueron aceptables (85% de aciertos) pero se el 50% de las transacciones fraudulentas produjo falsos negativos4. Para mejorar esto se determinó que se realizaría una fase de pruebas en la que se cambiaran los pesos de los falsos negativos y falsos positivos y cambiar la cantidad de datos con las que se entrena.

Objetivo(s) de la siguiente fase:

- Utilizar distintas cantidades de datos para mejorar los resultados de la SVM.
- Cambiar el peso de las transacciones de falsos negativos y positivos en el entrenamiento de la SVM para mejorar sus resultados.

Detalle de fase actual:

- Duración de la fase: 4 semanas
- Desglose de actividades realizadas
  - Implementación de la SVM (1.5 semanas)
  - Entrenamiento (1 semana)
  - Pruebas (3.5 semana)
- Recursos utilizados
  - Python
  - ScikitLearn
  - Computadora con sistema operativo Windows de 8gb de RAM y procesador core i7 de 3Ghz
- Q.4,000.00

Detalle de la siguiente fase:

- Duración aproximada: 4 semanas
- Recursos requeridos:
  - Python
  - ScikitLearn
  - Computadora con sistema operativo Windows de 8gb de RAM y procesador core i7 de 3Ghz
- Q.4,000.00

Reporte fase terminada # 7

Desarrollador:	**Diego Enríquez**

Algoritmo:	**SVM**

Fase actual:	*SVM versión final*
Fase predecesora:	*SVM versión 4*
Fase sucesora:	*N/A*

Fecha de desarrollo:	*09/06/2014 a 10/01/2014*

Objetivo(s) de la fase actual:

- Utilizar distintas cantidades de datos para mejorar los resultados de la SVM.
- Cambiar el peso de las transacciones de falsos negativos y positivos en el entrenamiento de la SVM para mejorar sus resultados.

Resultado(s) obtenidos:

- Se consiguió una SVM con 80.41% de aciertos, 18.85% de falsos negativos y 17.47% de falsos positivos. Esto se logró cambiando los pesos de la SVM para entrenarla e ingresando las transacciones positivas por cada 500,000 transacciones negativas.

Detalle de fase actual:

- Duración de la fase: 4 semanas
- Desglose de actividades realizadas
  - Pruebas (4 semanas)
- Recursos utilizados
  - Python
  - ScikitLearn
  - Computadora con sistema operativo Windows de 8gb de RAM y procesador core i7 de 3Ghz
- Q. 4,000.00

Q. Formularios de información general y fases terminadas de redes bayesianas

Información general

Lenguaje de programación utilizado:	Python

Justificación de su elección:

El lenguaje de programación, se escogió basándose en dos principios:

1. Las librerías disponibles para el manejo y administración de redes bayesianas.
2. La administración de memoria y recursos dentro del sistema operativo para que los algoritmos sean lo más eficientes posible, debido a la gran carga de datos.

Estos dos aspectos, fueron la principal razón para escoger los lenguajes de programación en los que se desarrollo la solución. En este caso, Python es el óptimo para la realización de la inferencia por medio de la red bayesiana utilizando la librería BayesPy.
Debido al gran poder computacional, se utilizo una base de datos en MongoDB conectada a través de Python para que la comunicación sea un tanto más eficiente que la ejecución de consultas en una base de daos relacional.

Tema de investigación:	**Redes Bayesianas**

Justificación de su elección:

Dentro de los algoritmos que se suelen estudiar al momento de buscar predecir los comportamientos de diferentes fenómenos, siempre se buscan tecnologías con algoritmos avanzados que puedan aprender de sí mismos de una manera diferente. Sin embargo, se optó por incluir una implementación que utiliza herramientas más exactas como es el caso de las distribuciones probabilísticas, siendo éstas el eje sobre el cual giran las redes bayesianas.
Las distribuciones probabilísticas son muy importantes a tomar en cuenta debido a que han demostrado ser eficiente en predecir comportamientos, en calcular probabilidades y tener un dato exacto de una posible ocurrencia. Es por ello, que se ha decidido incluir en la realización de este proyecto.

Reporte fase terminada # 1

Desarrollador:	**Melinton Navas**

Algoritmo:	**Redes bayesianas**

Fase actual:	*Investigación Redes Bayesianas*
Fase predecesora:	*N/A*
Fase sucesora:	*Investigación de librería de BayesPy*

Fecha de desarrollo:	*Enero 17/2014 – Febrero 7/2014*

Objetivo(s) de la fase actual:

- Actualizar el tema a Redes Bayesianas y realizar una investigación teórica acerca de los conceptos que componen y respaldan una Red Bayesiana.

Resultado(s) obtenidos:

- Se obtuvo como resultado que las redes bayesianas pueden ser una implementación útil para poder realizar inferencia de datos a un nivel estadístico, en base a un conjunto de datos previamente conocidos.

Objetivo(s) de la siguiente fase:

- En la siguiente fase, se realizará una investigación tomando en cuenta las posibles librerías disponibles para poder trabajar el algoritmo en cuestión

Detalle de fase actual:

- 3 semanas
- Investigación de conceptos y conceptos relacionados siempre de una forma teórica (no técnica)
- No se utilizaron recursos indispensables, más que fuentes de consulta y un equipo para consolidar información.
- Q.450.00

Detalle de la siguiente fase:

- Dos semanas
- No hay recursos indispensables, solo las fuentes de consulta.
- Q.300.00

Reporte fase terminada # 2

Desarrollador:	**Melinton Navas**

Algoritmo:	**Redes bayesianas**

Fase actual:	*Investigación de librería de BayesPy*
Fase predecesora:	*Investigación Redes Bayesianas*
Fase sucesora:	*Investigación e implementación de ambiente de desarrollo*

Fecha de desarrollo:	*Febrero 8/2014 – Febrero 21/2014*

Objetivo(s) de la fase actual:

- Realizar una investigación técnica acerca de las librerías disponibles para utilizar como herramienta para construir la red bayesiana propuesta.

Resultado(s) obtenidos:

- Se obtuvo como resultado que la librería más adecuada para la realización de este proyecto es BayesPy, debido a que posee múltiples herramientas que, a pesar de aún estar en desarrollo, son más completas que otras existentes.

Objetivo(s) de la siguiente fase:

- En la siguiente fase, se realizará una investigación e implementación del ambiente de desarrollo adecuado para la instalación de la librería y sus dependencias.

Detalle de fase actual:

- 2 semanas
- Investigación técnica de librerías orientadas a la construcción y análisis de redes bayesianas.
- No se utilizaron recursos indispensables, más que fuentes de consulta y un equipo para consolidar información.
- Q300.00

Detalle de la siguiente fase:

- Una semanas
- Los recursos a utilizar serán: un equipo con suficientes recursos para soportar software de virtualización y una conexión a internet para la descarga de las herramientas necesarias para construir el ambiente.
- Q500.00

Reporte fase terminada # 3

Desarrollador:	**Melinton Navas**

Algoritmo:	**Redes bayesianas**

Fase actual:	*Investigación e implementación de ambiente de desarrollo*
Fase predecesora:	*Investigación de librería de BayesPy*
Fase sucesora:	*Elaboración de miniproyecto aplicando redes bayesianas*

Fecha de desarrollo:	*Febrero 22/2014 – Marzo 21/2014*

Objetivo(s) de la fase actual:

- Realizar una investigación técnica acerca de los requisitos necesarios para poder llevar a cabo una instalación limpia de las herramientas y librerías necesarias para poder trabajar con redes bayesianas utilizando los resultados obtenidos en las investigaciones previas. Posterior a esto, se realizará la implementación de lo encontrado.

Resultado(s) obtenidos:

- Esta fase se extendió considerablemente del tiempo esperado, ya que las dependencias de diferentes herramientas se veían afectadas por el versionamiento de las mismas. Al finalizar, sí se logró obtener un ambiente de desarrollo apto para poder llevar a cabo el desarrollo de redes bayesianas con la librería BayesPy.

Objetivo(s) de la siguiente fase:

- En la siguiente fase, se realizará un pequeño proyecto de prueba para verificar el funcionamiento de la herramienta investigada.

Detalle de fase actual:

- 4 semanas
- Investigación e implementación de ambiente de desarrollo apto para el trabajo con redes bayesianas.
- Únicamente se utilizó un equipo, software de virtualización y las herramientas necesarias para poder soportar esta librería.
- Q2,000.00

Detalle de la siguiente fase:

- Tres semanas
- Los recursos a utilizar serán: un equipo con suficientes recursos para soportar software de virtualización y ambiente que soporte el uso de BayesPy.
- Q2,500.00

Reporte fase terminada # 4

Desarrollador:	**Melinton Navas**

Algoritmo:	**Redes bayesianas**

Fase actual:	*Elaboración de miniproyecto aplicando redes bayesianas*
Fase predecesora:	*Investigación e implementación de ambiente de desarrollo*
Fase sucesora:	*Estudio de análisis y datos de entrenamiento/ Avances con redacción de informe*

Fecha de desarrollo:	*Marzo 22/2014 – Abril 11/2014*

Objetivo(s) de la fase actual:

- Realizar una implementación de un miniproyecto para poder comprender la forma en que BayesPy funciona y como las diferentes variables aleatorias se comportan dentro de la librería.

Resultado(s) obtenidos:

- Se determinó que en base a este miniproyecto, la decisión de tipo de variable a utilizar para el análisis de datos es una parte crucial para el diseño del proyecto final.

Objetivo(s) de la siguiente fase:

- Estudio y análisis de datos de entrenamiento.

Detalle de fase actual:

- 3 semanas
- Implementación de miniproyecto de redes bayesianas BayesPy.
- Únicamente se utilizó un equipo, software de virtualización y las herramientas necesarias para poder soportar esta librería.
- Q2,500.00

Detalle de la siguiente fase:

- Tres meses
- Los recursos a utilizar serán: un equipo con suficientes recursos para análisis, estudio y manipulación de datos.
- Q5,000.00

Reporte fase terminada # 5

Desarrollador:	**Melinton Navas**

Algoritmo:	**Redes bayesianas**

Fase actual:	*Estudio de análisis y datos de entrenamiento/ Avances con redacción de informe*
Fase predecesora:	*Elaboración de miniproyecto aplicando redes bayesianas*
Fase sucesora:	*Análisis de miniproyecto*

Fecha de desarrollo:	*Abril 12/2014 – Junio 6/2014*

Objetivo(s) de la fase actual:

- Análisis de los datos obtenidos y que servirían de entrada para el entrenamiento de la red bayesiana, junto con el avance en redacción del informe escrito que se presentará al final del proyecto.

Resultado(s) obtenidos:

- Debido a algunos inconvenientes dentro del proyecto, esta fase se extendió en su duración por lo que se incorporó el avance en la redacción del documento final, permitiendo así consolidar en el documento los hallazgos anteriores. Una vez se obtuvo los datos con los que se trabajarían, se procedió a analizar cada uno de éstos y determinar cuáles eran necesarios para que la red pudiera construirse de forma óptima. Después de realizar el análisis debido, se concluyó que se utilizarían catorce campos de la data proporcionada.

Objetivo(s) de la siguiente fase:

- Análisis de los campos seleccionados y construcción del diagrama relacional.

Detalle de fase actual:

- 2 meses
- Estudio y análisis de los datos con los que se trabajarán.
- Únicamente se utilizó un equipo, software de virtualización y las herramientas necesarias para poder soportar el poder de procesamiento y análisis de los datos
- Q1,000.00

Detalle de la siguiente fase:

- Una semana
- Los recursos a utilizar serán: un equipo con suficientes recursos para análisis, estudio y manipulación de datos.
- Q1,000.00

Reporte fase terminada # 6

Desarrollador:	**Melinton Navas**

Algoritmo:	**Redes bayesianas**

Fase actual:	*Análisis de miniproyecto*
Fase predecesora:	*Estudio de análisis y datos de entrenamiento/ Avances con redacción de informe*
Fase sucesora:	*Definición de nodos a utilizar para la implementación de red bayesiana*

Fecha de desarrollo:	*Junio 9/2014 – Agosto 8/2014*

Objetivo(s) de la fase actual:

- Análisis del miniproyecto para poder abstraer el funcionamiento de una red neuronal y de esta forma tener un mejor conocimiento de la construcción de la misma.

Resultado(s) obtenidos:

- Junto al análisis de los resultados de miniproyecto, se analizó la data que se nos proporcionó para poder hacer un modelo que se adapte de la mejor forma al funcionamiento de una red bayesiana.

Objetivo(s) de la siguiente fase:

- Diseño de los nodos a utilizar en la red bayesiana

Detalle de fase actual:

- 2 meses
- Estudio y análisis de los datos con los que se trabajarán.
- Únicamente se utilizó un equipo, software de virtualización y las herramientas necesarias para poder soportar el poder de procesamiento y análisis de los datos
- Q1,000.00

Detalle de la siguiente fase:

- Una semana
- Los recursos a utilizar serán: un equipo con suficientes recursos para análisis, estudio y manipulación de datos.
- Q1,500.00

Reporte fase terminada # 7

Desarrollador:	**Melinton Navas**

Algoritmo:	**Redes bayesianas**

Fase actual:	*Definición de nodos a utilizar para la implementación de red bayesiana*
Fase predecesora:	*Estudio de análisis y datos de entrenamiento/ Avances con redacción de informe*
Fase sucesora:	*Definición de diagrama relacional*

Fecha de desarrollo:	*Agosto 9/2014 – Agosto 15/2014*

Objetivo(s) de la fase actual:

- El objetivo de esta fase es definir los nodos que se utilizarán en la red bayesiana y que compondrán el diagrama de relaciones a generar.

Resultado(s) obtenidos:

- Se obtuvo un total de 13 nodos, de los cuales existe uno que se genera al momento de procesar los datos.

Objetivo(s) de la siguiente fase:

- Análisis de los campos seleccionados y construcción del diagrama relacional.

Detalle de fase actual:

- 1 semana
- Generación de nodos que se utilizarán en la red bayesiana.
- Únicamente se utilizó un equipo, software de virtualización y las herramientas necesarias para poder soportar el poder de procesamiento y análisis de los datos
- Q2,000.00

Detalle de la siguiente fase:

- Una semana
- Los recursos a utilizar serán: un equipo con suficientes recursos para análisis, estudio y manipulación de datos.
- Q1,000.00

Reporte fase terminada # 8

Desarrollador:	**Melinton Navas**

Algoritmo:	**Redes bayesianas**

Fase actual:	*Definición de diagrama relacional*
Fase predecesora:	*Definición de nodos a utilizar para la implementación de red bayesiana*
Fase sucesora:	*Definición de reglas probabilísticas*

Fecha de desarrollo:	*Agosto 16/2014 – Agosto 22/2014*

Objetivo(s) de la fase actual:

- El objetivo de esta fase es crear un diagrama de relaciones que determinen la forma en que los nodos influyen entre sí de forma probabilística. Es decir, qué nodo afecta la probabilidad de ocurrencia de qué otro nodo.

Resultado(s) obtenidos:

- Se obtuvo un diagrama relacional acíclico.

Objetivo(s) de la siguiente fase:

- Generación de reglas probabilísticas para determinar un porcentaje de ocurrencia para cada variable aleatoria que se desea generar.

Detalle de fase actual:

- 1 semana
- Se generó el diagrama relacional acíclico que utiliza los nodos determinados en la fase anterior.
- Únicamente se utilizó un equipo, software de virtualización y las herramientas necesarias para poder soportar el poder de procesamiento y análisis de los datos
- Q1,000.00

Detalle de la siguiente fase:

- Una semana
- Los recursos a utilizar serán: un equipo con suficientes recursos para análisis, estudio y manipulación de datos.
- Q2,000.00

Reporte fase terminada # 9

Desarrollador:	**Melinton Navas**

Algoritmo:	**Redes bayesianas**

Fase actual:	*Definición de reglas probabilísticas*
Fase predecesora:	*Definición de diagrama relacional*
Fase sucesora:	*Reestructuración de proyecto*

Fecha de desarrollo:	*Agosto 23/2014 – Agosto 29/2014*

Objetivo(s) de la fase actual:

- El objetivo de esta fase es crear las reglas probabilísticas que se generan a partir del diagrama la relacional y el comportamiento de las variables entre sí.

Resultado(s) obtenidos:

- Un conjunto de 9 reglas definidas para la generación de las probabilidades de ocurrencia de variables aleatorias, las cuales mostraron ser computacionalmente incalculables para el alcance de este proyecto.

Objetivo(s) de la siguiente fase:

- Reestructuración del proyecto debido a los hallazgos en la fase anterior. SE procura enfocar este proyecto en la detección de por lo menos un nodo seleccionado.

Detalle de fase actual:

- 1 semana
- Se generaron 9 reglas probabilísticas a partir del diagrama relacional.
- Únicamente se utilizó un equipo, software de virtualización y las herramientas necesarias para poder soportar el poder de procesamiento y análisis de los datos
- Q2,500.00

Detalle de la siguiente fase:

- Una semana
- Los recursos a utilizar serán: un equipo con suficientes recursos para análisis, estudio y manipulación de datos.
- Q1,500.00

Reporte fase terminada # 10

Desarrollador:	**Melinton Navas**

Algoritmo:	**Redes bayesianas**

Fase actual:	*Reestructuración de proyecto*
Fase predecesora:	*Definición de reglas probabilísticas*
Fase sucesora:	*Generación de umbrales y ponderaciones*

Fecha de desarrollo:	*Septiembre 1/2014 – Septiembre 12/2014*

Objetivo(s) de la fase actual:

- Se determinó que el objetivo final de la investigación es comprobar el funcionamiento de las redes bayesianas, por lo que se finalizará el proyecto con la estructura definida en el inicio del mismo. Por lo que utilizando las reglas generadas, se creará una red bayesiana sin framework (es decir, sin utilizar BayesPy), debido a que este tipo de red no se adaptaría a la que requiere el framework.

Resultado(s) obtenidos:

- Se definió un conjunto de reglas que nos permite saber el porcentaje de ocurrencia de una transacción en particular. Estas reglas existen para cada uno de los nodos definidos completando en sí un 100% de probabilidad al sumar todas.

Objetivo(s) de la siguiente fase:

- En la siguiente fase se definirá un conjunto de banderas, ponderaciones y umbrales que permitirán determinar si una transacción es o no fraudulenta al momento de recorrer todos los nodos.

Detalle de fase actual:

- 2 semanas
- Se generaron miles de datos porcentuales en base a las reglas probabilísticas.
- Únicamente se utilizó un equipo, y las herramientas.
- Q1,500.00

Detalle de la siguiente fase:

- Una semana
- Los recursos a utilizar serán: un equipo con suficientes recursos para análisis, estudio y manipulación de datos.
- Q1,500.00

Reporte fase terminada # 11

Desarrollador:	**Melinton Navas**

Algoritmo:	**Redes bayesianas**

Fase actual:	*Generación de umbrales y ponderaciones*
Fase predecesora:	*Reestructuración de proyecto*
Fase sucesora:	*Computación de nodos y cálculo de efectividad*

Fecha de desarrollo:	*Septiembre 16/2014 – Septiembre 26/2014*

Objetivo(s) de la fase actual:

- Se determinó un sistema de banderas (roja, amarilla, verde) que define la probabilidad de fraude de cada nodo, ésta se define con diferentes umbrales para cada nodo. Por su parte, cada nodo tiene asociada una ponderación diferente, dependiendo de qué tan representativo es el nodo para que se detecte una transacción fraudulenta.

Resultado(s) obtenidos:

- Se obtuvo un sistema de ponderaciones, umbrales y banderas para cada nodo, que al unirlo determina si la transacción (después de pasar por los nodos) es o no fraudulenta.

Objetivo(s) de la siguiente fase:

- En la siguiente fase se computará la data reservada para prueba y se verificarán los valores de falsos positivos y negativos obtenidos.

Detalle de fase actual:

- 10 días
- Se generó el sistema por el cual pasarán las transacciones que se deseen analizar.
- Únicamente se utilizó un equipo, y las herramientas.
- Q2,500.00

Detalle de la siguiente fase:

- Una semana
- Los recursos a utilizar serán: un equipo con suficientes recursos para análisis, estudio y manipulación de datos.
- Q500.00

Reporte fase terminada # 12

Desarrollador:	**Melinton Navas**

Algoritmo:	**Redes bayesianas**

Fase actual:	*Computación de nodos y cálculo de efectividad*
Fase predecesora:	*Generación de umbrales y ponderaciones*

Fecha de desarrollo:	***Septiembre 27/2014 – Septiembre 30/2014***

Objetivo(s) de la fase actual:

- Se computó la data reservada para probar la red bayesiana para determinar cuántos de estos efectivamente son fraudes y si la red bayesiana fue capaz de determinarlos de esta forma.

Resultado(s) obtenidos:

Aciertos: 143264 (71.63% sobre el total de datos)
Desaciertos: 56736 (28.37% sobre el total de datos)
Falsos positivos: 56696 (99.93% sobre el total de desaciertos)
Falsos negativos: 40 (0.07% sobre el total de desaciertos)
Total analizados: 200000 (100%)
Alertas generadas: 143,285 (71.64%)
==========
Total fraudes en data de prueba: 59
Total no fraudes en data de prueba: 199,941
==========
Porcentaje de aciertos: 71.63%
Porcentaje de aciertos específicamente en data fraudulenta: 32.20%

Detalle de fase actual:

- 4 días
- Se probó la red bayesiana diseñada.
- Únicamente se utilizó un equipo, y las herramientas.
- Q500.00

R. Formularios de información general y fases terminadas de reconocimiento de patrones

Información general

PROGRAMA UTILIZADO	R Y R STUDIO

Justificación de su elección:

-
- R es un sofware gratuito.
- Posee una comunidad académica muy amplia en línea con excelente documentación.
- La capacidad de operación en diferentes áreas estadísticas y de análisis en reconocimiento de patrones.
- Paquetes específicos para el tema de clustering y modelos estadísticos.
- R recrea gráficas con mayor calidad que otros paquetes gratuitos.
- R es funcional en cualquier plataforma (mac, linux, windows).

Tema de investigación:	Reconocimiento de patrones y clustering

Justificación de su elección:

- El reconocimiento de patrones es óptimo para el análisis de las transacciones ya que se utilizan modelos para determinar correlación entre variables y dependencias de las mismas para explicar ciertas variables. Los modelos para reconocimiento toma una variable respuesta y es explicada por otras a la eleccion del empleador. Por lo que si se encuentra un modelo que prediga las variables que detecten con mayor precision los fraudes estaremos aumentando la efectividad de detección de fraudes.

Reporte fase terminada # 1

Desarrollador:	**BERNNY IXCAYAU**

TEMA:	**RECOCONOCIMIENTO DE PATRONES**

Fase actual:	*INVESTIGACION Reconocimiento de patrones*
Fase predecesora:	*N/A*
Fase sucesora:	*Desarrollo de modelo beta Clustering*

Fecha de desarrollo:	*6 ENERO-28 FEBRERO*

Objetivo(s) de la fase actual:

- Recopilación de información sobre reconocimiento de patrones
- Ventajas y desventajas de modelos de predicción

Resultado(s) obtenidos:

- **Modelos posiblemente a utilizar:**
  - Regresión logistica
  - Árboles de decisiones
  - Clustering

Objetivo(s) de la siguiente fase:

- Realizar pruebas con modelos de clustering

Detalle de fase actual:

- Duración: 1 mes y 23 días
- Investigación
- Internet, computadora, libros

Detalle de la siguiente fase:

- Duración: 1 mes
- Computadora

Reporte fase terminada # 2

Desarrollador:	**BERNNY IXCAYAU**

MODELO	**CLUSTERING**

Fase actual:	*Desarrollo de modelo beta clustering*
Fase predecesora:	*INVESTIGACION Reconocimiento de patrones*
Fase sucesora:	*Desarrollo de modelo regresión logística*

Fecha de desarrollo:	*01 marzo – 01 abril*

Objetivo(s) de la fase actual:

- Probar si es factible utilizar el modelo de clustering
- Determinar variables que aportan al modelo

Resultado(s) obtenidos:

- Se podrá utilizar el modelo si se tiene un record por numero de tarjeta, esto para tener un resultado de mayor reelevancia
- Variables importantes corrida 1:
  - 87507 - acf-fecha trx, 87512 - acf-condición pto venta, 87513 - acf-id adq, 87553 - acf-monto dollar
- Variables importantes corrida 2:
  - Vwjefecha, vwjefechamon, vwjefraud, vwscore, vwscore2, vwscore3, vwscore4

Objetivo(s) de la siguiente fase:

- Realizar pruebas con modelos de regresión logística

Detalle de fase actual:

- Duración: 1 mes
- 40 horas
- Desarrollo de primera corrida de clustering
- Internet, computadora.

Detalle de la siguiente fase:

- Duracion: 1 mes
- Computadora

Reporte fase terminada # 3

Desarrollador:	**BERNNY IXCAYAU**

MODELO	**RECOCONOCIMIENTO DE PATRONES**

Fase actual:	*Pre procesamiento de datos finales*
Fase predecesora:	*Desarrollo de modelo clustering beta*
Fase sucesora:	*Procesamiento de datos finales*

Fecha de desarrollo:	*2 abril-1 mayo*

Objetivo(s) de la fase actual:

Obtención y organizar los datos para entrenamiento final

Resultado(s) obtenidos:

- Plus ti proporcionó los datos y se organizaron para el procesamiento.

Objetivo(s) de la siguiente fase:

- Preparar los datos para la corrida final
- Convertir los datos a archivos csv
- Eliminar celdas vacías para entrenar en r

Detalle de fase actual:

- Duración: 1 mes
- Plus ti realizó esta fase

Detalle de la siguiente fase:

- Duración: 1 mes
- Computadora

Reporte fase terminada # 4

Desarrollador:	**BERNNY IXCAYAU**

MODELO	

Fase actual:	*Procesamiento de datos finales*
Fase predecesora:	*Preprocesamiento*
Fase sucesora:	*Árboles de decisión*

Fecha de desarrollo:	*2 mayo-1 junio*

Objetivo(s) de la fase actual:

- Organizar los datos para ser procesados en r

Resultado(s) obtenidos:

- Se eliminó celdas vacías
- Se eliminaron campos con un único valor ya que no agregan valor al resultado
- Se eliminaron campos que no agregaban valor, según el desarrollador, al análisis

Objetivo(s) de la siguiente fase:

- Encontrar patrones que identifiquen si la transacción es fraude.

Detalle de fase actual:

- Duración: 1 mes
- 15 horas debido a la falta de recursos computacionales

Detalle de la siguiente fase:

- Duración: 1 mes
- Computadora

Reporte fase terminada # 5

Desarrollador:	**BERNNY IXCAYAU**

MODELO	**ÁRBOLES DE DECISIÓN**

Fase actual:	*ÁRBOLES DE DECISIÓN*
Fase predecesora:	*PROCESAMIENTO*
Fase sucesora:	*REGRESIÓN LOGISTICA*

Fecha de desarrollo:	*2 junio-1 julio*

Objetivo(s) de la fase actual:

- Reconocer patrones que identifiquen si la operación es un fraude.

Resultado(s) obtenidos:

- El árbol nos dice que únicamente dos variables son las importantes:
  - C87519: País de origen
  - C87584: ID cliente

- Lo que el árbol quiere decirnos es que si la transacción no proviene de los países DK, RU no serán fraude. Mientras que si pertenecen a estos países pasará a un segundo nivel, verifica si el ID del cliente es menor a 3.3e^6 no será fraude mientras que si el ID es mayor a esa cifra pasará a un 3er nivel. Luego pregunta si el ID del cliente es mayor o igual a 2.9e^6 no será fraude mientras que si es menor será marcado como fraude.

- La precisión del modelo visto anteriormente es: 99.851% de efectividad.

- El árbol nos dice que únicamente dos variables son las importantes nuevamente:
  - C87519: País de origen
  - C87510: Merchant Category Code

- Lo que el árbol quiere decirnos es que si la transacción no proviene de los países DK, RU no serán fraude. Mientras que si pertenecen a estos países pasará a un segundo nivel, verifica si la categoría del negocio (MCC) es menor a 5661 no será fraude mientras que si la categoría del comercio es mayor a esa cifra pasará a un 3er nivel. Luego pregunta si el MCC es mayor o igual a 5455 no será fraude mientras que si es menor será marcado como fraude.

- La precisión del modelo es: 99.850% de efectividad, comparando con el modelo anterior ligeramente es

superior a este por lo que podemos decir que el modelo anterior es mejor.

- El árbol nos dice que nuevamente dos variables son las importantes:
  - C87519: País de origen
  - C87584: ID cliente

- Lo que el árbol quiere decirnos es que si la transacción no proviene de los países DK, LB no serán fraude. Mientras que si pertenecen a estos países pasará a un segundo nivel, verifica si el ID del cliente es menor a 3.3e^6 no será fraude mientras que si el ID es mayor a esa cifra pasará a un 3er nivel. Luego pregunta si el ID del cliente es mayor o igual a 2.9e^6 no será fraude mientras que si es menor será marcado como fraude. Básicamente nos refleja el mismo resultado que el primero con la variación de uno de los países.

- La precisión del modelo es: 99.85% de efectividad, esta es la misma precisión que el modelo 1.

Objetivo(s) de la siguiente fase:

- Encontrar patrones que identifiquen si la transacción es fraude

Detalle de fase actual:

- Duración: 1 mes
- 20 horas debido a la falta de recursos computacionales

Detalle de la siguiente fase:

- Duración: 1 mes
- Computadora

Reporte fase terminada # 6

Desarrollador:	**BERNNY IXCAYAU**

MODELO	**REGRESIÓN LOGÍSTICA**

Fase actual:	*Regresión logística*
Fase predecesora:	*Arboles de decisión*
Fase sucesora:	*Clustering*

Fecha de desarrollo:	*2 julio-1 agosto*

Objetivo(s) de la fase actual:

- Reconocer patrones que identifiquen si la operación es fraude

Resultado(s) obtenidos:

- En este modelo se incluyen todas las variables descritas anteriormente. Podemos ver que ciertas variables no son significativas para predecir el fraude. A continuación veremos que variables son las más significativas.
  - C87500: Llave primaria de control (Tarjeta o ID Cliente)
  - C87506: Hora TRX
  - C87507: Fecha TRX
  - C87510: MCC
  - C87512: Condición Pto Venta
  - C87513: ID Adq
  - C87567: Tipo Prod TC
  - C87547: Cod Moneda Trx
  - C87566: Marca o Franquicia
  - C87593: Semana del Año
  - C87594: Grupo Día
  - C87675: Bin & MCC
  - C87714: Evaluacion Dispositivo Chip

Según el modelo de RL estas son las variables que mejor explican el fraude. Este modelo alcanza una precisión de 99.5188%

- Para este modelo se utiliza los mismos datos únicamente que se omiten algunas variables no significativas del modelo anterior. Las variables significativas para el modelo son:
  - C87500: Llave primaria de control (Tarjeta o ID Cliente)
  - C87506: Hora TRX
  - C87507: Fecha TRX
  - C87510: MCC
  - C87512: Condición Pto Venta
  - C87513: ID Adq
  - C87567: Tipo Prod TC
  - C87547: Cod Moneda Trx
  - C87566: Marca o Franquicia
  - C87593: Semana del Año
  - C87594: Grupo Día

Al eliminar una de las variables no significativas podemos ver que otras variables que antes aportaban al modelo ahora ya no. Con este modelo se alcanza una precisión de: 99.518%

- Para este modelo se utiliza los mismos datos únicamente que se omiten algunas variables no significativas del modelo anterior. Las variables significativas para el modelo son:
  - C87506: Hora TRX
  - C87507: Fecha TRX
  - C87510: MCC
  - C87512: Condición Pto. Venta
  - C87513: ID Adq
  - C87547: Cod Moneda Trx
  - C87566: Marca o franquicia
  - C87584: ID Cliente
  - C87593: Semana del año
  - C87594: Grupo día
  - C87714: Evaluación dispositivo chip

Al eliminar una de las variables no significativas podemos ver que otras variables que antes aportaban al modelo ahora ya no. Con este modelo se alcanza una precisión de: 99.52%

Objetivo(s) de la siguiente fase:

Encontrar patrones que identifiquen si la transacción es fraude

Detalle de fase actual:

- Duración: 1 mes

20 horas debido a la falta de recursos computacionales

Detalle de la siguiente fase:

- Duración: 1 mes
- Computadora

Reporte fase terminada # 7

Desarrollador:	**BERNNY IXCAYAU**

MODELO	**CLUSTERING**

Fase actual:	**CLUSTERING**
Fase predecesora:	*REGRESIÓN LOGISTICA*
Fase sucesora:	*REPORTE final*

Fecha de desarrollo:	*1 AGOSTO-1 SEPTIEMBRE*

Objetivo(s) de la fase actual:

- Reconocer patrones que identifiquen si la operación es fraude

Resultado(s) obtenidos:

2 clústers que el algoritmo relacionó. Existen dos grandes grupos dentro de los datos de entrada. Para los datos se utilizaron datos fraudulentos y no fraudulentos.

- El primero, entre los campos más importantes se puede mencionar que:
  - Monto Original promedio es: 236.63
  - La marca o franquicia promedio es 4
  - Las transacciones fue en promedio en la semana 18
  - La variable de las condiciones del punto es 50
  - La mayoría de las transacciones se encuentran entre las categorías de negocios cercanas a 5441.

- El segundo clúster podemos mencionar:
  - Monto Original promedio es: 169
  - La marca o franquicia promedio es 4
  - Las transacciones fue en promedio en la semana 18
  - La variable de las condiciones del punto es 50
  - La mayoría de las transacciones se encuentran entre las categorías de negocios cercanas a 5522.

Datos fraudulentos

- El primero, entre los campos más importantes se puede mencionar que:
  - Monto Original promedio es: 217
  - La marca o franquicia promedio es 4
  - Las transacciones fue en promedio en la semana 19
  - La variable de las condiciones del punto es 50

- La mayoría de las transacciones se encuentran entre las categorías de negocios cercanas a 6427.

- El segundo clúster podemos mencionar:
  - Monto Original promedio es: 104
  - La marca o franquicia promedio es 4
  - Las transacciones fue en promedio en la semana 18
  - La variable de las condiciones del punto es 50
  - La mayoría de las transacciones se encuentran entre las categorías de negocios cercanas a 6312.

Objetivo(s) de la siguiente fase:

- Encontrar patrones que identifiquen si la transacción es fraude

Detalle de fase actual:

- Duración: 1 mes
- 20 horas debido a la falta de recursos computacionales

Detalle de la siguiente fase:

- Duración: 2 semanas