

UNIVERSIDAD DEL VALLE DE GUATEMALA
FACULTAD DE INGENIERÍA
Departamento de Computación



**COMPARACIÓN DE ALGORITMOS DE APRENDIZAJE DE
MÁQUINA PARA LA CLASIFICACIÓN DE EMOCIONES
HUMANAS**

Trabajo de graduación en modalidad de Tesis presentado por

Freddy Eduardo Martínez Gómez

**para optar al grado académico de Licenciado en Ingeniería en Ciencias de la
Computación y Tecnologías de la Información**

GUATEMALA
2017

UNIVERSIDAD DEL VALLE DE GUATEMALA
FACULTAD DE INGENIERÍA
Departamento de Computación



**COMPARACIÓN DE ALGORITMOS DE APRENDIZAJE DE
MÁQUINA PARA LA CLASIFICACIÓN DE EMOCIONES
HUMANAS**

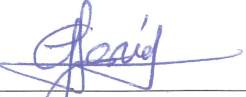
Trabajo de graduación en modalidad de Tesis presentado por

Freddy Eduardo Martínez Gómez

**para optar al grado académico de Licenciado en Ingeniería en Ciencias de la
Computación y Tecnologías de la Información**

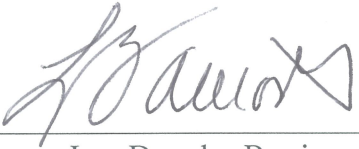
GUATEMALA
2017

Vo. Bo:

(f) 

Ing. Lynette García

Tribunal:

(f) 

Ing. Douglas Barrios

(f) 

PhD. Pablo Barrientos

(f) 

Ing. Lynette García

Fecha de Aprobación: 27 de noviembre de 2017

ÍNDICE

LISTADO DE FIGURAS	viii
LISTADO DE TABLAS	ix
RESUMEN	x
I. INTRODUCCIÓN	1
II. OBJETIVO	2
A. Objetivo general.....	2
B. Objetivos específicos	2
III. JUSTIFICACIÓN	3
IV. MARCO TEÓRICO	4
A. Aprendizaje de máquina	4
B. Aprendizaje de máquina supervisado	4
1. Clasificadores lineales	6
2. Regresión logística.....	6
3. Redes bayesianas	7
4. Máquina de soporte vectorial.....	8
5. Árboles de decisión.....	8
6. Redes neuronales	8
7. K Vecinos más próximos.....	9
C. Aprendizaje de máquina no supervisado	9
1. Agrupamiento	10
a. Agrupamiento jerárquico	10
2. K-Medias.....	11
3. Redes neuronales artificiales.....	13
a. RNA según la arquitectura.....	13
b. RNA según el aprendizaje	14
c. Componentes de la RNA	15
D. Supervisado vs no supervisado	16
1. Ventajas y desventajas de los AAMS	16
2. Ventajas y desventajas de los AAMNS	16
E. Emociones.....	16
F. Tecnologías que se utilizan en el AM.....	18

1.	Python	18
a.	Bibliotecas de Python para el AM	18
2.	R.....	19
a.	Bibliotecas de R para el AM.....	19
3.	Java	19
a.	Bibliotecas de Java para el AM	19
4.	Javascript.....	20
a.	Bibliotecas en Javascript para el AM.....	20
G.	Señales Electroencefalográficas (EEG)	20
H.	Emotiv EPOC.....	21
I.	Electrodos	23
J.	Transformada de Wavelets	23
K.	Validación cruzada.....	23
L.	Extracción de características.....	24
V.	MARCO METODOLÓGICO	26
A.	Investigación previa	26
B.	Conjunto de datos	26
C.	Pre-procesamiento del conjunto de datos	26
D.	Datos de entrenamiento.....	27
E.	Tecnología utilizada.....	27
1.	Python vs Java.....	27
2.	Python vs Javascript.....	27
3.	Python vs R.....	27
F.	Selección de algoritmos	27
1.	Árboles de decisión.....	28
2.	K-Vecinos más Próximos (KNN)	28
3.	Redes Bayesianas.....	28
G.	Módulo de Python utilizado.....	28
H.	Implementación de los algoritmos	29
VI.	RESULTADOS.....	30
A.	Pre-procesamiento de datos	30
B.	Segmentos utilizados	31

C. Datos de entrenamiento.....	31
D. Implementación de los algoritmos	32
E. Matrices de confusión	34
VII. DISCUSIÓN	37
VIII. CONCLUSIONES.....	39
IX. RECOMENDACIONES	40
X. BIBLIOGRAFÍA.....	41
XII. GLOSARIO.....	45

LISTADO DE FIGURAS

Figura 1. Proceso de un algoritmo de aprendizaje de máquina supervisado.	5
Figura 2. Cómputo de las probabilidades en las redes bayesianas	7
Figura 3. Estructura de una red bayesiana con probabilidades representando la relación entre nodos.	7
Figura 4. Cálculo de la distancia Euclidiana entre dos puntos	9
Figura 5. Cálculo de la clase del objeto a clasificar.....	9
Figura 6. Cálculo de la clase del vecino del dato a clasificar	9
Figura 7. Ejemplo de elementos individuales a ser agrupados.	11
Figura 8. Dendograma final con los elementos agrupados en grupos.	11
Figura 9. Se escogen 3 centroides generados aleatoriamente.	12
Figura 10. K grupos son generados asociándole el punto con la media más cercana.....	12
Figura 11. El centroide de cada uno de los K grupos se recalcula.....	13
Figura 12. Se repiten los pasos anteriores hasta la convergencia.	13
Figura 13. Arquitectura de una Red de Hopfield.....	14
Figura 14. Arquitectura de una red multi-capas.....	14
Figura 15. Modelo dimensional de emociones	18
Figura 16. Tipos de Onda detectados por Electroodos	21
Figura 17. Dispositivo Emotiv EPOC.....	22
Figura 18. Esquema k-fold de una validación cruzada, con k=4 y un solo clasificador.....	24
Figura 19. Señal EEG en bruto para el sujeto no.20 de los electrodos F3, F4, AF3 y AF4.	30
Figura 200. Aplicación de la transformada de Wavelets sobre el electrodo AF3.....	30
Figura 211. Segmentos de las señales EEG para los electrodos F3, F4, AF3 y AF4 para el sujeto no.4.....	31

LISTADO DE TABLAS

Tabla 1. Emociones Fundamentales con su respectiva reacción y función	17
Tabla 2. Rangos de frecuencias que componen las señales EEG	24
Tabla 3. División de segmentos obtenidos del pre-procesamiento de datos.....	31
Tabla 4. Cantidad de segmentos utilizados para el entrenamiento con el 70% de los segmentos	32
Tabla 5. Cantidad de segmentos utilizados para el entrenamiento con el 80% de los segmentos	32
Tabla 6. Cantidad de segmentos utilizados para el entrenamiento con el 90% de los segmentos	32
Tabla 7. Efectividad de cada algoritmo con un conjunto de entrenamiento del 70%.....	32
Tabla 8. Efectividad de cada algoritmo con un conjunto de entrenamiento del 80%.....	33
Tabla 9. Efectividad de cada algoritmo con un conjunto de entrenamiento del 90%.....	33
Tabla 10. Resumen de los más altos y más bajos porcentajes presentados por cada algoritmo para cada conjunto de entrenamiento.....	34
Tabla 11. Matriz de confusión para el algoritmo de árboles de decisión con el grupo de entrenamiento del 70% y una efectividad del 80.1%	34
Tabla 12.. Matriz de confusión para el algoritmo de redes bayesianas con el grupo de entrenamiento del 70% y una efectividad del 30.2%	35
Tabla 13. Matriz de confusión para el algoritmo de K-Vecino más próximo con el grupo de entrenamiento del 70% y una efectividad del 88.4%	35
Tabla 14. Matriz de confusión para el algoritmo de árboles de decisión con el grupo de entrenamiento del 80% y una efectividad del 82.6%	35
Tabla 15. Matriz de confusión para el algoritmo de redes bayesianas con el grupo de entrenamiento del 80% y una efectividad del 33.4%	35
Tabla 16. Matriz de confusión para el algoritmo de K-Vecino más próxima con el grupo de entrenamiento del 80% y una efectividad del 89.0%	35
Tabla 17. Matriz de confusión para el algoritmo de árboles de decisión con el grupo de entrenamiento del 90% y una efectividad del 86.2%	36
Tabla 18. Matriz de confusión para el algoritmo de redes bayesianas con el grupo de entrenamiento del 90% y una efectividad del 30.9%	36
Tabla 19. Matriz de confusión para el algoritmo de K-Vecino más próxima con el grupo de entrenamiento del 90% y una efectividad del 89.0%	36

RESUMEN

Durante el siguiente trabajo de graduación se muestra la evaluación de diferentes tipos de Algoritmos de Aprendizaje de Máquina, AMM de ahora en adelante, para determinar, en base a los resultados obtenidos de la implementación de cada uno de ellos, cuál de estos algoritmos clasifica de una manera más exacta y eficiente las emociones humanas. Lo primero que se realizó fue la investigación de los diferentes AMM que existen, y según estos resultados se escogieron los algoritmos utilizados. Luego de tener los algoritmos seleccionados, se procedió a obtener los datos a evaluar en estos algoritmos. Los datos utilizados fueron señales Encefalográficas, señales EEG de ahora en adelante, ya que estas señales son pulsos electricos que el cerebro humano genera al sentir y expresar una emoción. Estas señales EEG mostraban únicamente pulsos de emociones de felicidad y tristeza, las cuales fueron las emociones que se decidieron clasificar.

Al ya tener las señales EGG, los algoritmos a implementar y las emociones seleccionadas para clasificar, se procedió a la implementación de los algoritmos. Para esta implementación se seleccionó el lenguaje de programación a utilizar y se trabajó en ese lenguaje. Luego de este proceso, y al tener resultados del mismo, se logró concluir el algoritmo que clasifica de mejor manera las emociones humanas. Al finalizar se discutió sobre los resultados obtenidos y se crearon conclusiones en base a esta discusión y resultados.

I. INTRODUCCIÓN

Las emociones humanas son parte fundamental de la vida de un ser humano. Además de esto, el reconocimiento de las mismas ha sido un área a conocer y experimentar desde los años 60 en adelante. Asimismo, la inteligencia artificial ha tomado un auge a partir de los años 80 debido a la facilidad y utilidad que se les puede dar. Pero, ¿existe algún tipo de inteligencia artificial que pueda reconocer y a la misma vez, clasificar las emociones humanas?

La inteligencia artificial posee una rama que se encarga de que una computadora pueda aprender a tomar decisiones por su cuenta, sin necesidad de un mediador de las mismas. Con esto se han logrado grandes avances en la ciencia, computación, estadística, matemáticas, etc. Dentro de la inteligencia artificial existen algoritmos dedicados a lo anterior, el aprendizaje de la máquina por cuenta propia. Estos algoritmos se clasifican en según el conjunto de datos que evalúan, existiendo los Algoritmos de Aprendizaje de Máquina Supervisados, no Supervisado y Semi-supervisados.

Dentro de esta investigación se ha decidido implementar, obtener y analizar los resultados de derivados de los algoritmos de aprendizaje de máquina, y comparar su rendimiento en el análisis de ondas encefalográficas. Las ondas encefalográficas son pulsos eléctricos dentro del cerebro en los que se transmite información del ser humano. Parte de esa información que se transporta en los impulsos eléctricos que produce el cerebro contiene información sobre las emociones que una persona tiene en un instante dado. Según lo anterior, esta investigación tomó un conjunto de datos, los evaluó en los algoritmos seleccionados, se compararon los resultados obtenidos y se definió cuál de los algoritmos utilizados es el mejor en la clasificación de emociones humanas. Los algoritmos a implementar se encuentran definidos dentro de la metodología utilizada, así como la descripción del conjunto de datos utilizado y luego, en la parte de resultados, se muestran las respuestas obtenidas de cada algoritmo para lograr un análisis de las mismas y poder concluir acerca del tema.

II. OBJETIVO

A. Objetivo general

- Comparar el desempeño de distintos Algoritmos de Aprendizaje de Máquina (AAM) para clasificar emociones humanas capturadas a través de señales Electroencefalográficas (EEG).

B. Objetivos específicos

- Investigar y seleccionar los AAM a implementar.
- Implementar derivados de los AAM para la clasificación de emociones en base a las señales EEG de personas.
- Comparar los resultados obtenidos de los algoritmos a implementar.

III. JUSTIFICACIÓN

El enfoque principal de este trabajo es la identificación de emociones humanas por medio de la inteligencia artificial, específicamente los Algoritmos de Aprendizaje de Máquina (AAM). Esta es la principal pregunta a responder en el trabajo. Como se sabe, el ser humano es un ser que tiene una infinidad de expresiones gestuales, tanto físicas como emocionales, todas ellas concebidas dentro del cerebro mismo. La comparación de los resultados de diferentes AAM provee una oportunidad de determinar cuál de ellos es el mejor, recibiendo el mismo conjunto de datos como entrada, señales encefalográficas obtenidas de personas.

Estudios anteriores demuestran que los AAM, entre muchos otros, son utilizados para clasificar emociones, pero estos estudios se limitan a la implementación de únicamente un AAM. Autores como Petrantonokis y Hadjileontiadis (Petrantonokis & Hadjileontiadis, 2009) utilizaron algoritmos de Validación Cruzada para la detección de emociones en base a señales EEG. O también, autores como Chai Tong y Woo San (Chai Tong, Woo San, & al) utilizaron algoritmos de Redes Neuronales para la clasificación de emociones. Debido a esto, la presente investigación será una base tanto teórica como práctica al momento de la elección de algoritmos para la clasificación de emociones.

IV. MARCO TEÓRICO

A. Aprendizaje de máquina

El Aprendizaje de Máquina, también conocido como Machine Learning en inglés y de ahora en adelante AM, son una rama de la Inteligencia Artificial que tiene como objetivo crear sistemas capaces de aprender de ellos mismos a partir de un conjunto de datos sin ser programados de forma explícita (Moya, 2016) Asimismo, como apunta Murphy, el AM se puede definir como un conjunto de métodos que son capaces de detectar patrones en los datos, y después hacer uso de estos patrones descubiertos para realizar predicciones de datos futuros o brindar la posibilidad de realizar una toma de decisiones en condiciones de incertidumbre. (Murphy, 2012). Los algoritmos de AM requieren precisión, efectividad y un mínimo margen de error para lograr obtener modelos predictivos correctos, estables, efectivos y acertados en la predicción de información. (S. & Vishwanatan, 2008) Según Kotsiantis, las personas normalmente son susceptibles a cometer errores durante el análisis de la información o hasta posiblemente, tienen más dificultad al momento de establecer relación alguna entre múltiples características de un modelo. (Kotsiantis S. , 2012)

Richiert, define el AM como una clasificación de algoritmos que tienen como objetivo, enseñar a la máquina la forma correcta de llevar a cabo ciertas tareas, a través del análisis de ejemplos concretos, de qué hacer, para ejecutarla (Richiert, 2016). Los algoritmos de aprendizaje de máquina, se dividen según el tipo de aprendizaje que utilizan. Los dos tipos principales son: Aprendizaje Supervisado y Aprendizaje No Supervisado, existiendo un tercero llamado Aprendizaje Semi-Supervisado.

B. Aprendizaje de máquina supervisado

El Aprendizaje de Máquina Supervisado, en inglés Supervised Machine Learning y de ahora en adelante AMS, tiene como objetivo principal el mapear desde una entrada X a una salida Y , dado un conjunto de etiquetas de pares de entrada-salida $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$ donde \mathcal{D} es llamado el conjunto de entrenamiento (training set), y N es el número de muestras de entrenamiento (Murphy, 2012).

En la configuración más simple, cada entrada de entrenamiento X_i , es un vector numérico con D dimensiones. Donde cada uno de los valores del vector son llamados características, atributos o covariables. En general, X_i podría tener una estructura de objetos muy compleja, tales como una imagen, una oración, un mensaje de correo, un conjunto de datos, una forma molecular, un grafo, etc. (Murphy, 2012).

De la misma manera, la forma en que se presenta la salida o la variable de respuesta puede ser cualquier cosa, pero la mayoría de los métodos suponen que Y_i es una variable categórica o nominal de algún conjunto finito de etiquetas. En este caso el problema es conocido como clasificación o reconocimiento de patrones. En cambio, cuando la variable de respuesta representa un valor escalar (como un nivel de ingresos), el problema se conoce como regresión (Murphy, 2012).

El proceso de aplicar un AAMS a una aplicación en el mundo real se puede observar en la Figura no.1

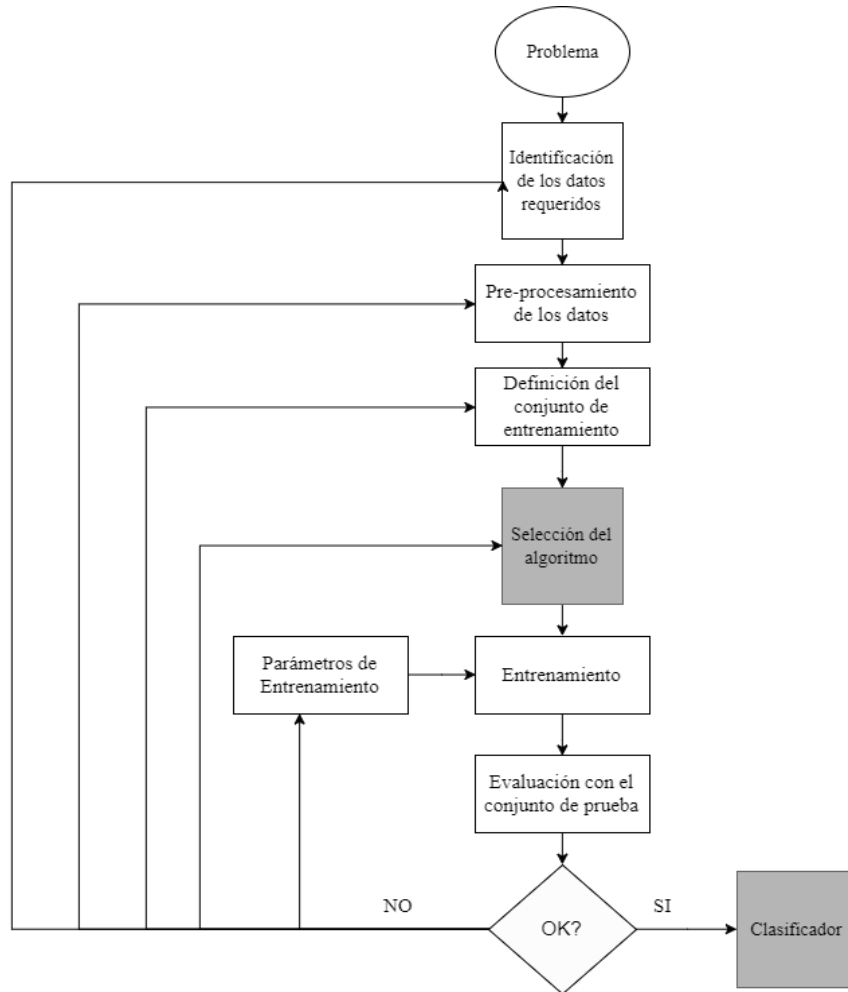


Figura 1. Proceso de un algoritmo de aprendizaje de máquina supervisado. (Zhang, Zhang, & Yang, 2002).

Los pasos de este proceso son. (Zhang, Zhang, & Yang, 2002):

1. Se identifica el problema a resolver.
2. Se identifican los datos a utilizar en el AAMS.
3. Se realiza un pre-procesamiento de los datos.
4. Se define el conjunto de entrenamiento.
5. Se realiza la selección del AAMS a utilizar.
6. Se implementa el algoritmo con el conjunto de entrenamiento de los datos.
7. Se realiza una evaluación con el conjunto de prueba.
8. Si los resultados son satisfactorios, se puede definir un clasificador. Si no, se puede repetir el proceso desde el paso 2, empezando por cualquier paso nuevamente

El primer paso es identificar el problema a resolver. Luego de eso, se selecciona la colección de los datos a utilizar. Si se cuenta con la opinión de un experto en el tema, se tiene que consultar con el mismo para determinar qué información es la relevante para el estudio. Si no se cuenta con esta opinión, se deben tomar en cuenta algunos factores importantes en los datos a utilizar, como por ejemplo, la cantidad de los datos para que los resultados de la implementación del algoritmo sean válidos, la calidad de los datos, para que el algoritmo no tenga problemas con la implementación o el tipo de información de los datos, entre otros. El método más utilizado para la elección de datos, pero poco recomendable, es el método de “Fuerza Bruta” el cual consiste en escoger la información esperando que sea útil para el estudio. Este método no es el más recomendable debido a que la información que se puede obtener, posee algún tipo de ruido o datos que no sean relevantes, provocando que la información pase a una etapa de Pre-procesamiento de información. (Zhang, Zhang, & Yang, 2002). Como siguientes pasos se tiene la definición de los diferentes conjuntos de datos a utilizar. Estos, se clasifican en un conjunto de entrenamiento y uno de pruebas. Al ya tener estos conjuntos definidos, se procede a la implementación del algoritmo seleccionado. Este paso es opcional ya que, al momento de implementar el algoritmo, se puede programar el mismo para que realice esta clasificación de conjuntos y trabaje con los mismos. Luego de la implementación, se procede a la evaluación del algoritmo con el set de entrenamiento y luego con el set de pruebas. Si los datos obtenidos son los requeridos, se puede definir un clasificador como tal, de lo contrario, se puede empezar desde cualquier paso anterior el proceso nuevamente.

Según Taiwo, los Algoritmos de Aprendizaje de Máquina Supervisados más utilizados para la clasificación de información son los siguientes (Taiwo, 2010):

1. Clasificadores lineales

Este tipo de clasificador separa los vectores de entrada en clases utilizando límites de decisión lineales. El principal objetivo de este tipo de clasificadores es agrupar la información de los datos en base a sus características. (Elder, 1997) Este tipo de clasificadores regularmente son utilizados en situaciones en las que la velocidad de la clasificación de los datos es un factor importante a tomar en cuenta, ya que estos clasificadores son considerados los más rápidos entre los AAMS. (Taiwo, 2010). Este tipo de algoritmos toma la decisión de clasificar un data basado en el valor de una combinación lineal de sus características. Típicamente las características de un objeto son presentadas en un vector, llamado “vector de características” (Taiwo, 2010).

2. Regresión logística

Este tipo de clasificadores realizan una función que utiliza un modelo de regresión logística simple multinomial con un solo estimador. Realiza predicciones más fuertes y más detalladas, y se pueden moldear a casi cualquier situación, pero estas predicciones pueden estar equivocadas. La Regresión Logística es una de las herramientas más comunes utilizadas para el análisis de información estadística y de data discreta. (Taiwo, 2010) Asimismo, la Regresión Logística es un tipo de análisis de regresión utilizado para predecir el resultado de una variable que puede tomar un número limitado de categorías, en función de las variables dependientes o predictivas del modelo. (William, 1991). Es útil para modelar la probabilidad de un evento ocurriendo como función de otros factores y es utilizada extensamente en las ciencias médicas y sociales. La

Regresión Logística es un tipo especial de regresión que se utiliza para explicar y predecir una variable categórica binaria, en función de varias variables independientes que a su vez pueden sean cuantitativas o cualitativas. (William, 1991)

3. Redes bayesianas

Este tipo de clasificador está compuesto por grafos acíclicos con un solo padre, el cual representa el nodo sin observar, y con una cantidad N de nodos, los cuales son los nodos a observar, con una fuerte dependencia entre nodos. Las Redes Bayesianas son los modelos más representativos de los algoritmos de aprendizaje estadístico. La característica más destacable de las redes bayesianas es ciertamente la posibilidad de tomar en cuenta información relevante de un problema, en términos de una relación entre sus características (Kotsiantis S. , 2012). Normalmente estos modelos se basan en estimaciones. (Nilsson, 1965). Debido a que los algoritmos de clasificación bayesiana utilizan multiplicaciones para computar probabilidades, véase la Figura no.2, son muy probables a encontrarse con probabilidades con valor de cero. Para evitar este problema, se puede utilizar un estimador de Laplace, añadiendo uno a todos los numeradores y añadiendo la cantidad de unos añadidos al denominador.

$$R = \frac{P(i|X)}{P(j|X)} = \frac{P(i)P(X|i)}{P(j)P(X|j)} = \frac{P(i) \prod P(X_r|i)}{P(j) \prod P(X_r|j)}$$

Figura 2. Cómputo de las probabilidades en las redes bayesianas

Una red bayesiana puede ser representada por un grafo acíclico dirigido S en donde los nodos en S son dependencias uno a uno con las características X , las aristas representan las influencias entre las características y la falta de las mismas representan que no hay una relación entre las mismas, véase la Figura no.3

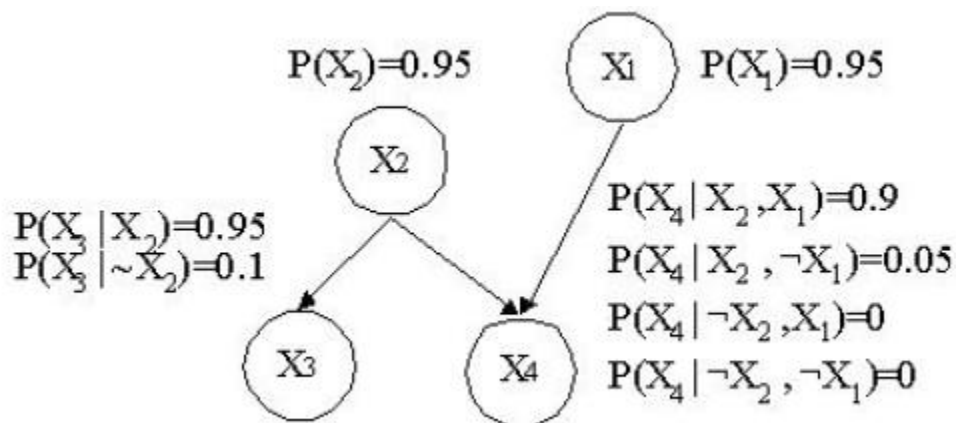


Figura 3. Estructura de una red bayesiana con probabilidades representando la relación entre nodos.

4. Máquina de soporte vectorial

También conocido como Support Vector Machine en inglés y de ahora en adelante SVM se basan principalmente en establecer un hiperplano que separa de forma óptima los subconjuntos de puntos correspondientes a cada una de las clases. Para ello utiliza los vectores de características que han sido proyectados previamente en un espacio de dimensionalidad superior. Asimismo, estos algoritmos son capaces de crear el modelo a partir de un conjunto de datos de entrenamiento. Este proceso convierte a estos algoritmos en una técnica con una gran eficiencia computacional (Sherwood, 2009).

El concepto detrás del algoritmo de las máquinas de vectores de soporte, es maximizar el discriminante marginal, lo cual se logra al proponer una solución utilizando optimización cuadrática por métodos de Lagrange (Sherwood, 2009).

En algunas investigaciones previas, se ha podido comparar el algoritmo de SVM, con otras técnicas de clasificación, llegando a la conclusión que este tiene una mejor precisión al clasificar. Pero lo más importante de todo, es que SVM se considera un clasificador binario, lo cual lo hace muy útil, en escenarios donde se quieren clasificar únicamente dos clases (Kotsiantis S. , 2007).

5. Árboles de decisión

Este tipo de algoritmo crea árboles que clasifican las instancias ordenándolas en base al valor de sus características. Cada nodo en un árbol de decisión representa una característica en un momento dado a ser clasificada y cada arista representa el valor que cada nodo puede tomar. Este tipo de algoritmo utiliza los árboles como un modelo predictivo que mapea las observaciones de un objeto hacia las conclusiones que puede tener ese objeto. (Hastie, Tibshirani, & Friedman, 2001).

6. Redes neuronales

Este tipo de algoritmos depende fundamentalmente de cuatro aspectos: las entradas, las funciones de activación de cada unidad, la arquitectura de la red y el peso de cada conexión entre las unidades. Ya que los primeros dos aspectos pueden ser manipulables, el comportamiento de una red neuronal depende en gran manera del peso que posean las relaciones entre unidades. Normalmente, los pesos de estas redes se inicializan con valores random, y luego, estos valores se cambian en base a que las instancias de los valores del conjunto de entrenamiento son expuestas a la red repetidamente. Los valores de cada entrada son colocados en las unidades de entrada y el resultado de estos valores de la red son comparados con los valores deseados a obtener. Luego, todos los pesos de las unidades se ajustan levemente en la dirección que se desea que tomen los resultados de las iteraciones. (Neocleous & Schizas, 2002).

7. K Vecinos más próximos

El algoritmo de k-vecinos más próximos, en inglés K-Nearest Neighbors, y de ahora en adelante KNN, utiliza el conjunto completo de datos para su entrenamiento. Luego de esto, para cada nueva entrada al algoritmo esta se clasifica en base al mayor número de clases dentro de los datos de entrenamiento (Duda & Hart, 1973). El algoritmo KNN clasifica los datos de entrenamiento y de prueba en base a la distancia que existe entre cada uno ellos y para esto utiliza la distancia Euclidiana (Duda & Hart, 1973).

$$D_{euclidean}(X, Y) = \sqrt{\sum_{i=1}^m d(x_i, y_i)^2} = \sqrt{\sum_{i=1}^m (x_i - y_i)^2}$$

Figura 4. Cálculo de la distancia Euclidiana entre dos puntos

Para clasificar un objeto según sus vecinos se puede hacer por mayoría, es decir, que para cada dato se revisan sus vecinos y la clase que tenga el mayor número de vecinos es la clase del dato a clasificar.

$$c_X = \underset{c}{\operatorname{argmax}} \sum_{n=1}^k I(c, c_n)$$

Figura 5. Cálculo de la clase del objeto a clasificar

Donde k es el número de vecinos y donde $I(c, c_n)$ es igual a

$$I(c_1, c_2) = \begin{cases} 1, & c_1 = c_2 \\ 0, & c_1 \neq c_2 \end{cases}$$

Figura 6. Cálculo de la clase del vecino del dato a clasificar

C. Aprendizaje de máquina no supervisado

Los Algoritmos de Aprendizaje de Máquina No Supervisado, de ahora en adelante AAMNS, están estrechamente relacionados con el problema de la estimación de la densidad en las estadísticas. (Jordan & Bishop, 2004). Una de las maneras para distinguir los AAMNS de los AAMS es que los datos que se le provee al algoritmo no poseen etiquetas, por lo tanto, no hay evaluación de la exactitud de la estructura que es emitida por el algoritmo seleccionado. (Jordan & Bishop, 2004)

Algunos de los enfoques de los AAMNS incluyen:

- Clustering: Entre los que se encuentran:
 - K-Means
 - Clustering jerárquico.
- Redes Neuronales.
 - Según su arquitectura.

- Según su aprendizaje.
- Componentes de la RNA.

1. Agrupamiento

El enfoque de un algoritmo de agrupamiento es congregar un conjunto de objetos de tal manera que los que se encuentran en el mismo grupo (Cluster) son más similares, de una u otra forma, que a los demás objetos en los demás grupos. El Agrupamiento es un gran enfoque que se puede utilizar en minería de datos, y es una técnica común dentro del análisis estadístico, utilizado en varios campos como el aprendizaje de máquina, el reconocimiento de patrones, el análisis de imágenes, bioinformática o en las gráficas de computadora (Pfitzner, Leibbrandt, & Powers, 2009)

Debido a que el concepto de un “cluster” como tal no puede ser definido totalmente existen diferentes algoritmos de agrupamiento. El común denominador entre ellos es el grupo de objetos de información. Los modelos estándar de un cluster incluyen:: (KTian & Raghu, 1994)

- Modelos de conexión: Por ejemplo, el agrupamiento jerárquico construye un modelo basado en la distancia de la conexión.
- Modelo centroide: Por ejemplo, el algoritmo de K-Means representa cada grupo como un vector.
- Modelos de distribución: Los grupos son modelados por medio de distribuciones estadísticas.
- Modelos de densidad: Algunos definen a los grupos como densas regiones dentro del conjunto de datos.
- Subespacio de modelos: En modelos de doble-agrupamiento, también llamados biagrupamiento, cada grupo es modelado con ambos miembros del grupo y sus atributos relevantes.
- Grupo de modelos: Algunos algoritmos no proveen un modelo redefinido para los resultados y solamente proveen información del grupo de datos (KTian & Raghu, 1994).

a. Agrupamiento jerárquico

Este tipo de agrupamiento está basado en la idea principal de que los objetos están más relacionados con otros objetos más cercanos a él, en lugar de los objetos más lejanos (Everitt, 2011). Estos algoritmos conectan los objetos para formar grupos en base a las distancias entre ellos (Everitt, 2011) . Las estrategias para este tipo de agrupamiento se pueden clasificar en dos tipos:

- Aglomerativas: Esta estrategia indica que inicialmente se forman grupos individuales, cada uno de los cuales contiene a un único elemento. En cada iteración se unen los dos grupos más próximos. El procedimiento finaliza cuando solo haya un grupo. (Everitt, 2011)
- Divisivos: Esta estrategia indica que se parte de un único grupo al que pertenecen todos los elementos. En cada iteración se escoge un grupo y se divide; se debe de decidir qué grupo se selecciona para dividir y cómo se divide. El procedimiento finaliza cuando haya tantos grupos como elementos (Everitt, 2011)

Dendogramas: Es la representación gráfica de un clustering jerárquico ya que en él se muestra en qué orden se han unido los grupos y cuál es el grado de proximidad que tienen los que se unen. Los nodos hoja del dendograma representan a los elementos individuales, el nodo raíz representa el grupo al que pertenecen todos los elementos y el resto de nodos representan los grupos que se van formando. (Everitt, 2011)

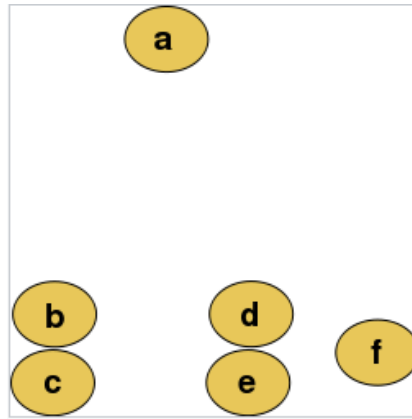


Figura 7. Ejemplo de elementos individuales a ser agrupados.

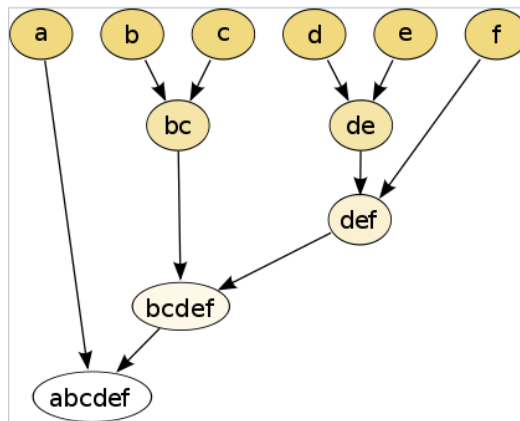


Figura 8. Dendograma final con los elementos agrupados en grupos.

2. K-Medias

Este es un método de cuantificación vectorial utilizado popularmente en el análisis de grupos para minería de datos. El objetivo principal de este algoritmo es la partición de un conjunto n de observaciones en k grupos en el que cada observación pertenece al grupo cuyo valor medio es más cercano (Jain & Dubes, 1988). El algoritmo está basado en la minimización de la distancia interna (la suma de las distancias de los patrones asignados a un agrupamiento, al centroide de dicho agrupamiento). De hecho, este algoritmo minimiza la suma de las distancias al cuadrado de cada patrón al centroide de su agrupamiento. Este tipo de algoritmos tienen una complejidad NP-Hard, sin embargo, hay algoritmos heurísticamente eficientes que convergen a un óptimo local (MacQueen, 1967). La complejidad de un algoritmo está dada en base a la cantidad de recursos computacionales que se tienen que utilizar (Sanjeev & Barak, 2009). El término NP proviene del

inglés “Nondeterministic Polynomial Time” y es el conjunto de los problemas de decisión que contienen los problemas H tales que todo problema L en NP pueda ser transformado polinomialmente. (Sanjeev & Barak, 2009)

Algoritmo Estándar: El algoritmo más común utiliza una técnica de refinamiento iterativo. Dado un conjunto inicial de k centroides $\mathbf{m}_1^{(1)}, \dots, \mathbf{m}_k^{(1)}$ el algoritmo continúa alternando entre dos pasos:

- Paso de asignación: Asigna cada observación al grupo con la media más cercana.

$$S_i^{(t)} = \{x_p : \|x_p - m_i^{(t)}\| \leq \|x_p - m_j^{(t)}\| \forall 1 \leq j \leq k\}$$

Donde cada x_p va exactamente dentro de un $S_i^{(t)}$, incluso aunque pudiera ir en dos de ellos.

- Paso de actualización: Calcular los nuevos centroides como el centroide de las observaciones en el grupo.

$$m_i^{(t+1)} = \frac{1}{|S_i^{(t)}|} \sum_{x_j \in S_i^{(t)}} x_j$$

El algoritmo se considera que ha convergido cuando las asignaciones ya no cambian. El procedimiento del algoritmo se muestra en las Figuras 6, 7, 8 y 9.

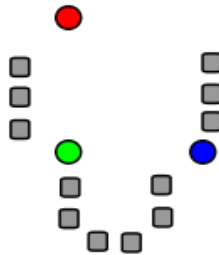


Figura 9. Se escogen 3 centroides generados aleatoriamente.

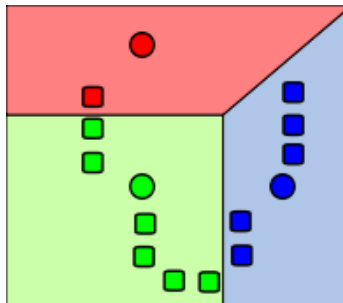


Figura 10. K grupos son generados asociándole el punto con la media más cercana.

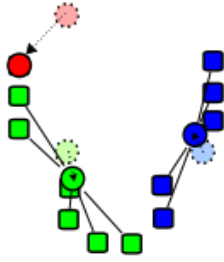


Figura 11. El centroide de cada uno de los K grupos se recalcula.

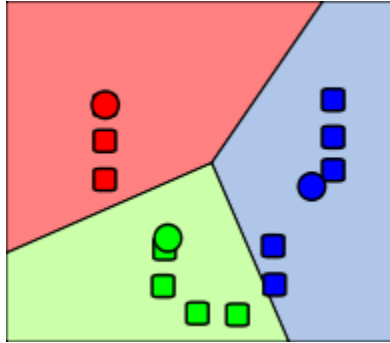


Figura 12. Se repiten los pasos anteriores hasta la convergencia.

3. Redes neuronales artificiales

También llamadas Sistemas de conectividad son sistemas computacionales inspirados en las redes biológicas neuronales del cerebro. Las redes neuronales artificiales, de ahora en adelante RNA, son aquellas redes en las que existen elementos procesadores de información de cuyas interacciones locales depende el comportamiento del conjunto del sistema (Florez & Fernández, 2008). Las RNA tratan de emular el comportamiento del cerebro humano, caracterizado por el aprendizaje a través de la experiencia y la extracción de conocimiento genérico a partir de un conjunto de datos. Los computadores presentan habitualmente una arquitectura basada en un microprocesador muy rápido capaz de ejecutar en serie de instrucciones complejas de forma fiable, mientras que el cerebro está compuesto por millones de procesadores elementales o neuronas, interconectadas entre sí formando redes (Florez & Fernández, 2008). Por su parte, las neuronas biológicas no necesitan ser programadas, sino que aprenden a partir de los estímulos que reciben el entorno y operan siguiendo un esquema masivamente paralelo, distinto al procesamiento en serie típico de los computadores convencionales (Florez & Fernández, 2008). Las RNA también se pueden considerar como AAMNS ya que las implementaciones que pueden tener estos algoritmos pueden ser en busca de reconocimiento de patrones, de imágenes, imágenes faciales, etc, sin necesidad de esperar como resultado un patrón en específico y esperando que las RNA puedan interpretar los patrones por sí solas. Las RNA se pueden clasificar en dos tipos:

a. RNA según la arquitectura

La arquitectura de una red consiste en la disposición y conexionado de las neuronas. En una red se pueden distinguir aspectos como el número de capas, el tipo de capas que posee, ya sean

ocultas o visibles, de entrada, o de salida y la direccionalidad de las conexiones de las neuronas (Jordan & Bishop, 2004). Este tipo de RNA se pueden clasificar en:

- Redes monocapa: cuentan con una capa de neuronas, que intercambian señales con el exterior y que constituyen a un tiempo la entrada y la salida del sistema. Un ejemplo de este tipo de redes es la Red de Hopfield. (Jordan & Bishop, 2004)

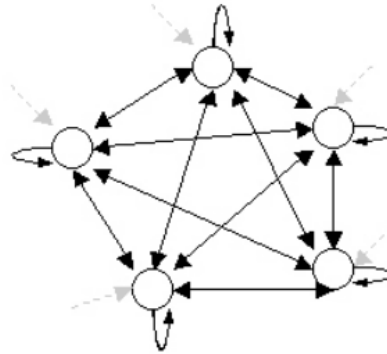


Figura 13. Arquitectura de una Red de Hopfield

- Redes multi-capas: están formadas por dos o más capas de neuronas conectadas entre ellas. Estas a su vez se dividen en:
 - Redes con conexiones hacia delante: Este tipo de redes contienen solo conexiones entre capas hacia delante. Esto implica que una capa no puede tener conexiones a una que reciba la señal antes que ella en la dinámica de la computación. (Jordan & Bishop, 2004)
 - Redes con conexiones hacia atrás: En este tipo de redes pueden existir conexiones de capas hacia atrás y por tanto la información puede regresar a capas anteriores en la dinámica de la red (Jordan & Bishop, 2004)

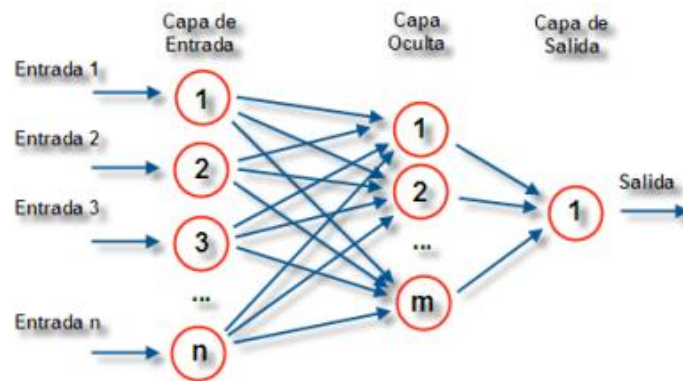


Figura 14. Arquitectura de una red multi-capas

b. RNA según el aprendizaje

El aprendizaje se basa en el entrenamiento de la red con patrones. El proceso de aprendizaje se basa en que la red ejecute los patrones de forma iterativa hasta que se muestren respuestas

satisfactorias. Es decir, los pesos sinápticos se ajustan para dar respuestas óptimas para el conjunto de patrones de entrenamiento. Podemos distinguir tres tipos de aprendizaje:

- Aprendizaje supervisado: la red dispone de los patrones de entrada y de salida que queremos obtener para esa entrada, y en función de ellos se modifican los pesos de las sinopsis para ajustar la entrada a la salida. (Florez & Fernández, 2008)
- Aprendizaje no supervisado: consiste en no proporcionar a la red los patrones de salida, sino sólo los de entrada y dejar que la red los clasifique en función de características comunes que encuentre entre ellos. (Florez & Fernández, 2008)
- Aprendizaje híbrido: No se proporcionan los patrones objetivo, sino que sólo se dice si la respuesta acierta o falla ante un patrón de entrada. (Florez & Fernández, 2008)

c. Componentes de la RNA

Neuronas: Una neurona j que recibe de una entrada $p_j(t)$ de una neurona predecesora tiene los siguientes componentes:

- Un conjunto de unidades de procesamiento: Pueden ser los siguientes:
 - Unidades de entradas: reciben señales del ambiente.
 - Unidades de salida: transmiten las señales fuera de la RNA.
 - Unidades ocultas: solo reciben señales de unidades que pertenecen a la RNA.
- Un estado de activación: Magnitud de la variable de estado de la unidad.
- Una función de salida para cada unidad: La señal de salida de una unidad varía con su activación, de acuerdo con una función de salida, generalmente una función de saturación no lineal.
- Un conjunto de conexiones: Las unidades de procesamiento están interconectadas. Con cada conexión se asocia un peso conocido como peso de conexión o peso sináptico. La magnitud de cada peso representa la eficiencia con la cual la señal de salida de una unidad es transmitida a la otra.
- Un conjunto de reglas de propagación: Una entrada en una unidad puede ser una versión en escala de la señal de salida recibida, pero también puede ser más compleja.
- Una regla de combinación: Necesaria para generar la entrada neta de las señales de salida pesadas o moduladas provistas por la regla de propagación.
- Una regla de activación: Necesaria para combinar la señal de entrada con el estado actual. La regla de activación puede ser una ecuación diferencial con respecto al tiempo, en el caso continuo, o una ecuación diferencial en el caso discreto. Además, la regla de activación puede ser determinística o estocástica. En este último caso, se incluye una fuente de ruido para producir cierta incertidumbre sobre el estado presente de la unidad.
- Una regla de modificación: La modificación del funcionamiento de una RNA en función del tiempo requiere la modificación de patrones de conectividad con el tiempo, usando alguna regla de modificación.
- Un ambiente en el cual opera la RNA: Provee las señales a las unidades de entrada. Las señales de entrada pueden ser discretas o continuas, constantes o dependientes del tiempo, y determinísticas o estocásticas. (Jordan & Bishop, 2004)

D. Supervisado vs no supervisado

Según Bishop, la principal diferencia entre los AAMS y los AAMNS radica en que los primeros reciben un conjunto de datos etiquetados y se tiene un conocimiento sobre las salidas que producen, ya que cuenta con información que especifica qué datos son satisfactorios para el objetivo del aprendizaje, en cambio los segundos, reciben un conjunto de datos sin ningún tipo de etiqueta y se espera que el mismo algoritmo reconozca patrones dentro de los datos ya que no se cuenta con la información que especifique si los resultados son satisfactorios o no. (Bishop C. , 2006).

1. Ventajas y desventajas de los AAMS

Según Hinton algunas de las ventajas de utilizar AMMS son las siguientes:

- Se obtienen salidas específicas sobre la definición de los datos queriendo decir con esto que se puede entrenar al clasificador de tal manera que se obtenga un parámetro perfecto de decisión para distinguir entre los datos.
- Se pueden especificar exactamente cuántas salidas se desean obtener.
- Luego de terminar el entrenamiento no es necesario seguir guardando en algún lugar el conjunto de datos de entrenamiento ya que se puede guardar como una fórmula matemática el parámetro de decisión, pudiendo ser utilizado en nuevas clasificaciones. (Hinton, 1996).

Asimismo, Hinton indica algunas de las desventajas de los AAMS son las siguientes:

- Se pueda dar el caso en que el parámetro de decisión haya sido “sobrentrenado” lo cual significa que se pueden excluir datos en el conjunto de entrenamiento y al momento de evaluar este tipo de datos en la validación, se pueden obtener resultados incorrectos.
- Se tiene que tener una clasificación de datos correcta para entrenar al algoritmo.
- Se requieren varios recursos computacionales para la fase de entrenamiento y de clasificación de los datos.
- El tiempo de clasificación de los datos pueden ser demasiado grande. (Hinton, 1996)

2. Ventajas y desventajas de los AAMNS

De la misma manera Hinton indica ventajas de utilizar los AAMNS son las siguientes:

- Pueden realizar predicciones según los patrones encontrados en su implementación.
- Automatiza y acelera los procesos de toma de decisiones.

Asimismo, las desventajas que Hinton indica son las siguientes:

- El tiempo de espera para que el algoritmo encuentre patrones sea demasiado grande.
- Los patrones encontrados por los AAMNS no sean de ayuda al estudio. (Hinton, 1996)

E. Emociones

Las emociones son tendencias de respuesta con un gran valor adaptativo, que tienen evidentes manifestaciones a nivel fisiológico, en la expresión facial, la experiencia subjetiva, el procesamiento de la información, etc., que son intensas, pero breves en el tiempo, y que surgen

ante la evaluación de algún acontecimiento anteriormente experimentado (Fredrickson B. L., 2001). Esta definición parece ajustarse mejor al estudio de emociones negativas (miedo, ira, asco, tristeza, etc.), que al estudio de emociones positivas (alegría, orgullo, satisfacción, esperanza, fluidez, elevación, etc.) fundamentalmente porque las primeras se pueden asociar a tendencias de respuesta claras y específicas y las segundas no tanto (Fredrickson & Levenson, 2001).

De la misma manera, una emoción puede ser considerada como un estado afectivo que experimenta un individuo, una reacción subjetiva al ambiente que viene acompañada de cambios orgánicos, de origen innato, influidos por la experiencia. En el ser humano, la experiencia de una emoción generalmente involucra a un conjunto de cogniciones, actitudes y creencias sobre el mundo, que utilizamos para valorar una situación concreta y, por tanto, influyen en el modo en el que se percibe dicha situación (Goleman, 1993). En la siguiente tabla se muestran las seis emociones fundamentales, así como la reacción y las funciones que pueden cumplir cada una de ellas:

Tabla 1. Emociones Fundamentales con su respectiva reacción y función

Emoción	Reacción	Función
Miedo	Anticipación de una amenaza o peligro que produce ansiedad, incertidumbre o inseguridad.	Produce tendencia hacia la protección.
Sorpresa	Asombro o desconcierto.	Provoca orientación frente a una nueva situación.
Asco	Disgusto, aversión y se produce un alejamiento del causante de la aversión.	Provoca rechazo hacia aquello que tenemos adelante.
Ira	Enojo, resentimiento, furia.	Induce hacia la destrucción.
Alegría	Euforia, gratificación, bienestar, seguridad.	Provoca un sentido de reproducción de la situación que provoca la emoción.
Tristeza	Pena, soledad, pesimismo	Provoca una nueva reintegración personal.

(Goleman, 1993)

La existencia de una gran cantidad de emociones hace que sea tan difícil el proceso de reconocimiento. Además, no existen como estados totalmente independientes a lo largo del tiempo en un sujeto, sino que pueden traslaparse unas con otras. Un buen modelo de emociones es aquel, que puede separarlas de forma clara y concisa. Diversos modelos han tratado de definir un conjunto de emociones básicas y representarlas en un modelo dimensional. Las seis emociones básicas son: Asco, miedo, sorpresa, ira, alegría y tristeza (Friesen, 1982). El modelo dimensional, comúnmente utilizado se caracteriza por tener dos dimensiones: valencia y excitación. Los rangos de valencia en las emociones pueden ir de negativo a positivo, mientras que los rangos para la excitación van de la calmado a excitado. Este modelo es ampliamente

utilizado en muchos estudios, porque permite expresar y clasificar las emociones de una manera muy sencilla. (Russell, 1980)

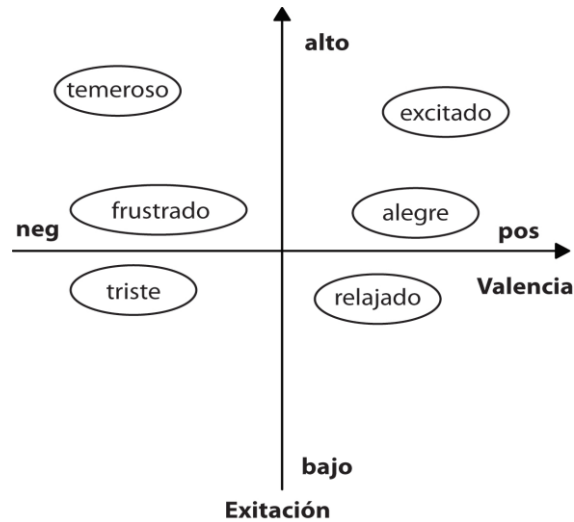


Figura 15. Modelo dimensional de emociones (Horlings, 2008)

F. Tecnologías que se utilizan en el AM

1. Python

Python es un lenguaje de programación orientado a objetos, sencillo, pero de gran potencia, que permite el desarrollo rápido de aplicaciones. Es un lenguaje interpretado y altamente modular, por lo que la etapa de desarrollo se reduce considerablemente y, además, resulta muy sencillo añadir nuevos elementos a posteriori al programa (Bauer, 1996) (Van Rossum & Drake, 2000). Python es un lenguaje de programación interpretado cuya filosofía hace énfasis en una sintaxis que favorezca un código legible. (Martelli, 2007)

a. Bibliotecas de Python para el AM

Entre las bibliotecas que están escritas en Python para el AM se encuentran:

- Scikit-Learn: Es la librería principal que existe escrita en Python, para trabajar con aprendizaje de máquina. Esta librería se puede utilizar para la clasificación de características, extracción de las mismas, regresiones, agrupaciones, reducción de dimensiones, selección de modelos o preprocesamiento de datos. (Lopez, 2015)
- Statsmodels: Esta librería ha énfasis en modelos estadísticos y se utiliza principalmente para el análisis predictivo y exploratorios. Las pruebas estadísticas que ofrece son bastante amplias y abarcan tareas de validación para la mayoría de algoritmos. (Lopez, 2015)
- PyMC: Es un módulo de Python que implementa modelos estadísticos bayesianos. Este módulo ofrece funcionalidades para hacer el análisis bayesiano lo más simple posible ya que incluye los modelos bayesianos, distribuciones estadísticas y herramientas de diagnóstico para la covarianza de los modelos. (Lopez, 2015)

2. R

R es un conjunto integrado de programas para manipulación de datos, cálculos y gráficos (Bates & Watts, 1998). Entre las características que dispone están:

- Almacenamiento y manipulación efectiva de datos.
- Operadores para cálculo sobre variables indexadas, especialmente matrices.
- Una amplia, coherente e integrada colección de herramientas para análisis de datos.
- Posibilidades gráficas para análisis de datos. (Bates & Watts, 1998)

a. Bibliotecas de R para el AM

Entre las bibliotecas escritas en R para el AM están:

- Mice: siglas para “Multivariate Imputation by Chained Equations” y es una librería utilizada para completar los valores perdidos o no disponibles de la base de datos que se esté utilizando. (Shah, 2016)
- Rpart: siglas para “Recursive Partitioning and Regression Trees” y es una librería para realizar árboles de clasificación y regresión. (Shah, 2016).
- Caret: siglas para “Classification And Regression Training” y es una librería que posee un marco común para combinar los modelos de entrenamientos y modelos de predicción. Este paquete ayuda al desarrollador a encontrar los parámetros correctos con experimentos controlados. (Shah, 2016)

3. Java

Java es un lenguaje de programación con un paradigma orientado a objetos diseñado específicamente la menor cantidad de dependencias de implementación como fuere posible. (Goslin & Joy, 2005). Su intención es permitir que los desarrolladores de aplicaciones escriban el programa una vez y lo ejecuten en cualquier dispositivo (conocido en inglés como WORA, o "write once, run anywhere"), lo que quiere decir que el código que es ejecutado en una plataforma no tiene que ser recompilado para correr en otra. (Goslin & Joy, 2005)

a. Bibliotecas de Java para el AM

Entre las bibliotecas escritas en Java para el AM están:

- Weka: siglas para “Waikato Environment for Knowledge Analysis” y es una plataforma que provee un ambiente gráfico, una línea de comandos y un API para el desarrollo de AAM. (Brownlee J. , 2014)
- Java-ML: siglas para “Java Machine Learning” es una librería que provee una colección de AAM ya implementados en Java, asimismo, provee una interfaz gráfica para cada algoritmo con métodos para la manipulación, agrupamiento, selección de características y clasificación. (Brownlee J. , 2014)

- JSAT: siglas para “Java Statistical Analysis Tool” provee implementaciones de AM escritas en Java. Esta desarrollada como una librería con “auto-aprendizaje” para realizar los análisis por sí sola. (Brownlee J. , 2014)

4. Javascript

Javascript es un lenguaje de programación interpretado. Se puede definir como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico. (Flanagan & Ferguson, 2002). El uso primordial de Javascript se da del lado del cliente, implementado como parte de un navegador web permitiendo mejores en la interfaz de usuario y páginas web dinámicas. (Flanagan & Ferguson, 2002)

a. Bibliotecas en Javascript para el AM

Entre las bibliotecas escritas en Javascript para el AM están:

- Brain: Librería que permite crear redes neuronales y entrenarlas en base a un conjunto de datos de entrada y/o salida (Comunidad, tutorialzine.com, 2017)
- Neurojs: Framework encargado de crear sistemas con inteligencia artificial basados en el aprendizaje por reforzamiento (Comunidad, tutorialzine.com, 2017)
- Synaptic: Librería que posee una arquitectura agnóstica, la cual permite a los desarrolladores crear cualquier tipo de red neuronal que deseen (Comunidad, tutorialzine.com, 2017).

G. Señales Electroencefalográficas (EEG)

Una de las funciones del tejido nervioso es producir potenciales eléctricos, ya que estos son la base para la excitabilidad del organismo. Aunque todo el sistema nervioso contiene esta característica, la electroencefalografía se basa en estudiar solo los relacionados con la corteza cerebral y las regiones directamente relacionadas con ella. La electroencefalografía se encarga de registrar y evaluar los potenciales eléctricos generados por la actividad en la corteza cerebral, generados por la actividad cerebral. Esto lo realiza por medio de electrodos. Lo que se detecta en un área de superficie subcortical es actividad eléctrica espontánea. (Barea, 2002).

Los potenciales anteriormente descritos se generan cuando un conjunto de neuronas se sincroniza, porque se activan por alguna reacción del cerebro, para así generar un potencial eléctrico. Con esto, el conjunto de todos los potenciales generados logra conseguir que se detecte una diferencia de potencial sobre la corteza cerebral. La actividad electroencefalográfica puede captarse de distintas formas: Por encima del cuero cabelludo, en la base del cráneo, en cerebro expuesto y en localizaciones cerebrales profundas. Dependiendo de qué forma se utilice, así deben ser los electrodos que se utilicen. Dependiendo también de qué forma se capten las señales, así también será la integridad de la señal recopilada, ya que entre más adentro del cerebro se encuentre el electrodo, mejor será la señal recopilada, siendo así que cuando se obtenga la señal estando por encima del cuero cabelludo, será la forma en que se tendrá menor integridad e intensidad de la señal electroencefalográfica, por la distancia que se tiene al punto en donde se origina la señal. (Barea, 2002)

Las señales electroencefalográficas, abreviadas como EEG por sus siglas en inglés, son fluctuaciones de voltaje que ocurren en el cerebro, las cuales juegan un papel muy importante en la detección de emociones. Se cree que las emociones están relacionadas con un grupo de estructuras en el centro del cerebro llamado sistema límbico, el cual incluye: amígdala cerebral, tálamo, hipotálamo y el hipocampo (Noppadon, 2013). Se dividen en cuatro tipos principales ya que cada una de ellas se caracteriza principalmente por tener un rango de frecuencia asociado y una sección del cerebro donde su presencia es mayor. (Kotsos, 2016). Estas señales son:

- Ondas Alpha: Estas señales se encuentran en un rango de frecuencias de 7-12Hz, su aparición habitual es en el área occipital del cerebro.
- Ondas Beta: Estas señales se encuentran en un rango de 14 a 40 Hz, estas señales se presentan cuando el individuo está en un estado de alerta, o está realizando algún razonamiento lógico.
- Ondas Delta: Esta es la señal con la frecuencia más baja, se encuentra en el rango de 0.3 a 3 Hz. Esta señal se da principalmente en estado de sueño profundo, en el momento donde el cuerpo se está regenerando.
- Ondas Theta: Estas señales están en el rango de 3-7 Hz, aparecen habitualmente en menores a 15 años y en adultos somnolientos. Estas ondas aparecen normalmente en la mayoría de las áreas del cerebro. (Kotsos, 2016)

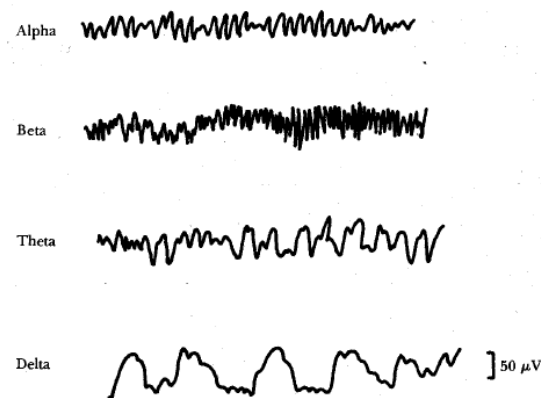


Figura 16. Tipos de Onda detectados por Electrodo
(Kotsos, 2016)

H. Emotiv EPOC

El enfoque del dispositivo “Emotiv” es la lectura de las ondas cerebrales por medio de electrodos, filtrando dichas lecturas para posterior análisis. Emotiv cuenta con siete pares de electrodos colocados de forma simétrica a lo largo del cuero cabelludo.

El cerebro está conformado por más de doce billones de neuronas distribuidas a lo largo del órgano encefálico. Este órgano cuenta con pliegues entrantes y fisuras que proveen una mayor área superficial en donde se encuentran las neuronas. Las neuronas, conformadas por

dendritas, axón, núcleo y pericarion, envían señales a otras neuronas por medio de las neuronas adyacentes. Estas señales son pulsos eléctricos que pueden ser captados por los electrodos.

Es de suma importancia recalcar el hecho que el cerebro tiene evaginaciones^[1] e invaginaciones^[2] características de cada individuo, por lo que un mapeo tridimensional de las señales eléctricas diferirá de un cerebro a otro incluso cuando se obtiene la misma respuesta a un estímulo, ya que la superficie no es idéntica. Sin embargo, los estudios encefalográficos han demostrado que hay distintas regiones que se comportan de manera singular según distintas actividades cerebrales. Es por esto que se ha logrado determinar múltiples regiones del cerebro con distintas funciones a su cargo.

Hay distintos tipos de electrodos con distintos propósitos. Entre estos electrodos se encuentran los superficiales (sobre el cuero cabelludo), los basales (base del cráneo sin procedimiento quirúrgico), y los electrodos quirúrgicos (corticales o intracerebrales). Estas distintas herramientas se utilizan para realizar los electroencefalogramas (EEG), electrocorticograma (ECoG) y/o estéreo electroencefalograma (E-EEG). Las ondas que estos electrodos detectan, se clasifican en distintas categorías según su función y periodicidad. Estas ondas pueden ayudar a catalogar el estado anímico, e incluso el tipo de pensamientos de un individuo. Los tipos de onda son alfa, beta, delta y theta, y aunque se encuentran presentes en todo momento, es la combinación la que varía al cambiar el estado de la persona.

El dispositivo Emotiv EPOC pertenece al área de la interacción cerebro-computadora, o más conocida en inglés como Brain Computer Interaction (BCI) y, además, está conformado por 14 canales de lectura de señales EEG, junto con un giroscopio el cual permite a la cámara del dispositivo seguir en todo momento la dirección en que se mueve la cabeza de la persona que lo utiliza. Los 14 canales que posee son: AF3, F7, F3, FC5, T7, P7, O1, O2, P8, T8, FC6, F4, F8 y AF4. Estos canales están basados en el sistema internacional 10-20 el cual es un sistema encargado de describir y aplicar la localización de los electrodos para las pruebas o experimentos del reconocimiento de señales EEG. Asimismo, el sistema provee un Kit de Desarrollo de Software, o SDK por sus siglas en inglés, para personalizar el uso de la tecnología a las necesidades metodológicas de los investigadores. El costo del dispositivo es de \$799, según la página oficial del mismo (<https://www.emotiv.com/epoc>), contando con una versión más pequeña del dispositivo con 5 canales, en lugar de 14. Este dispositivo fue diseñado y fabricado por la compañía estadounidense “Emotiv Systems”.



Figura 17. Dispositivo Emotiv EPOC

I. Electrodo

Para ilustrar la complejidad del cerebro, hay que mencionar que un solo milímetro cúbico contiene al menos 40,000 neuronas y 1,000 millones de conexiones de fibras nerviosas. Cada neurona y conexión genera campos eléctricos, que pueden llegar a cancelarse o sumarse dependiendo de su posición (Thevenet, 2008). Por eso los neuropsicólogos han desarrollado sistemas de posicionamiento de electrodos, que buscan obtener lecturas óptimas de dichas señales. Entre estos sistemas se encuentra el Sistema Internacional 10-20, antes mencionado.

J. Transformada de Wavelets

La transformada de Wavelets es muy similar a la transformada de Fourier, en la cual se utiliza el concepto de aproximación de señales usando la superposición de otras señales sobre las originales (González, 2011). El análisis utilizando la transformada de Fourier consiste en descomponer la señal original en señales sinusoidales de diferente frecuencia. De manera similar, la transformada de Wavelets descompone la señal en versiones trasladadas en el tiempo, y escaladas de la Wavelet original, conocida como Wavelet Madre (González, 2011).

La transformada de Wavelet combina las dimensiones del tiempo y frecuencia, permitiendo la extracción de los detalles y de las aproximaciones de la señal. Esto no puede ser obtenido aplicando ninguna de las versiones de la transformada de Fourier haciendo de Wavelet una opción bastante importante en el procesamiento de las señales EEG (Mallat, 1989).

K. Validación cruzada

La validación cruzada, en inglés “Cross-Validation”, es un método que se utiliza en el AM para determinar el conjunto de completo de datos para entrenar y validar un algoritmo dado (Brownlee J. , 2016). Básicamente consiste en dividir el conjunto de datos en un determinado número de grupos de igual tamaño denominados “folds”. Entonces, el algoritmo es entrenado a partir de todos los folds exceptuando uno, el cual es utilizado para la validación. Este proceso es repetido hasta utilizar todos los grupos de datos (folds), como datos de validación. Al final la exactitud del algoritmo es obtenida a partir del promedio de todas las exactitudes obtenidas en cada una de las iteraciones realizadas (Brownlee J. , 2016).

Asimismo, es una técnica utilizada para evaluar los resultados de un análisis estadístico y garantizar que son independientes de la partición entre datos de entrenamiento y prueba. Consiste en repetir y calcular la media aritmética obtenida de las medidas de evaluación sobre diferentes particiones (Devijver & Kittler, 1982). Se utiliza en entornos donde el objetivo principal es la predicción y se quiere estimar la precisión de un modelo que se llevará a cabo a la práctica. Es una técnica muy utilizada en proyectos de inteligencia artificial para validar modelos generados (Devijver & Kittler, 1982)

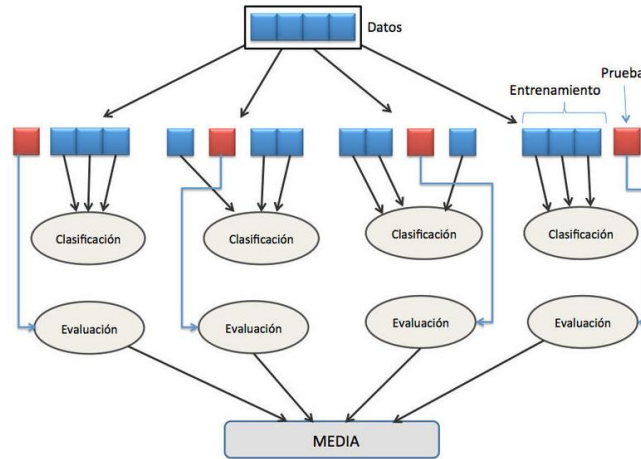


Figura 18. Esquema k-fold de una validación cruzada, con k=4 y un solo clasificador

L. Extracción de características

Se le conoce así al proceso en el cual se obtienen características representativas de una señal durante un tiempo específico, denominado ventana. Existen una gran cantidad de enfoques distintos para la extracción de dichas características. La característica ampliamente utilizada para la detección o clasificación de emociones es PSD (Power Spectrum Density). El PSD consiste en determinar el poder de una señal EEG en un rango de frecuencias específicas. (Noppadon, 2013).

La señal EEG, suele dividirse en rangos de frecuencias muy específicos, debido a que es en esta división, donde realmente se pueden determinar las características básicas presentes en la señal (Noppadon, 2013). Dichos rangos son descritos en la siguiente tabla:

Tabla 2. Rangos de frecuencias que componen las señales EEG

Nombre	Frecuencia (Hz)
Delta	0-4
Theta	4-8
Alpha	8-16
Beta	16-32
Gamma	32-64

(Noppadon, 2013)

Las bandas que se encuentran estrechamente relacionadas con el estado emocional de una persona son únicamente las Alpha y las Beta (Ramírez, 2008). Algunas investigaciones todavía toman en cuenta a la banda Gamma para este propósito, pero en general, las primeras dos son las más utilizadas.

Recientes investigaciones señalan que los electrodos AF3, AF4, F3 y F4, que se ubican en el córtex prefrontal del cerebro, son comúnmente utilizados en la detección de emociones. Debido a que es en ellos, donde se muestra un aumento dominante de ondas Alpha y ondas Beta en estados de excitación, alerta y relajación. Además, están ubicados en el lóbulo prefrontal, el cual

juega un papel crucial en la regulación de emociones y durante estados conscientes (Ramírez, 2008).

V. MARCO METODOLÓGICO

A. Investigación previa

La primera parte del desarrollo de la investigación fue investigar y tener los conocimientos teóricos sobre los temas de la misma, la Inteligencia Artificial, el Aprendizaje de Máquina, las ondas EEG y las emociones humanas. Se realizó la búsqueda de investigaciones que tuvieran el mismo fin, determinar el mejor o los mejores AAM para la clasificación de emociones humanas.

A partir de esto, se determinó la tecnología a utilizar para el desarrollo de la investigación. La tecnología a utilizar tenía que cumplir con factores como el uso minimizado de los recursos computacionales, el tiempo de ejecución de los algoritmos y la efectividad de las implementaciones.

B. Conjunto de datos

El conjunto de datos utilizados en la investigación fueron 20 toma de señales EEG las cuales se obtuvieron del trabajo de graduación “Desarrollo de SDK enfocado en el procesamiento de señales EEG para la clasificación y el reconocimiento de emociones” (García, 2016) desarrollado por un estudiante de la facultad de ingeniería de la Universidad del Valle de Guatemala. Se decidió utilizar este conjunto de datos ya que una de las bases de la presente investigación es comparar otros algoritmos de aprendizaje de máquina contra el utilizado en este trabajo de graduación, el algoritmo de Máquina de Soporte Vectorial. Estas 20 toma de señales se realizaron a 20 sujetos en las que se les realizó una prueba que consistía de 20 estímulos visuales. Estos 20 estímulos estaban divididos en 2 grupos, 10 fueron destinados para provocar un estado de felicidad y los otros 10 destinados a provocar un estado de tristeza en los sujetos. El tiempo de exposición de cada estímulo sobre la persona fue de 5 segundos. Entre cada estímulo visual se le mostró al sujeto uno de tipo neutral, para obtener un estado neutro entre cada estímulo. El dispositivo utilizado para la toma de datos fue Emotiv EPOC y se utilizó la lectura de los electrodos F3, F4, AF3 y AF4.

Los estímulos fueron obtenidos, luego de ser evaluados por 92 personas utilizando una encuesta en línea. Se les solicitaba a los participantes evaluar cada una de las imágenes en una escala de 0 a 10, donde 0 indicaba felicidad y 10 tristeza. Los resultados fueron tabulados y luego se extrajeron las 10 imágenes más representativas para cada emoción.

C. Pre-procesamiento del conjunto de datos

En esta etapa de la investigación se le aplicaron varios filtros al conjunto de datos para eliminar cualquier ruido que pudiera causar algún conflicto al momento de implementar los algoritmos utilizados. Los datos obtenidos se dividieron en segmentos de los datos en segundos que tienen una duración de 1 segundo. Uno de los filtros aplicados a estos segmentos fue el de descartar el primer segundo de cada 5 segundos que duraba un estímulo. Esto se debió a que el primer segundo del estímulo podría estar afectado por el estímulo anterior, alterando los datos y con esto, los algoritmos no podrían identificar la emoción en ese segundo.

Luego de este proceso se aplicó la transformada de Wavelets a los datos resultantes. Dicha transformada es capaz de descomponer la señal en distintos conjuntos de coeficientes de aproximación y de detalle, que representan las distintas bandas de frecuencia que componen la señal. Esta descomposición fue realizada, sobre cada uno de los segmentos resultantes del paso

anterior. Se utilizó esta transformada ya que permite extraer las frecuencias de la señal sin perder la dimensión del tiempo.

D. Datos de entrenamiento

Como se ha mencionado anteriormente en la investigación, para utilizar AAMS se necesitan dos grupos de datos, el de entrenamiento y el de pruebas. Para obtener los datos de entrenamiento, luego de haber pre-procesado todos los datos a utilizar, se utilizó el método de Validación Cruzada.

E. Tecnología utilizada

La tecnología utilizada durante el desarrollo de la presente investigación ha sido el lenguaje de programación llamado “Python” debido a, como mencionó Bauer y Van Rossum junto con Drake, este lenguaje es interpretado y modular, haciendo que el tiempo de desarrollo sea considerablemente menor. Además, comparado con otros lenguajes de programación como Java, el tiempo de ejecución y los recursos computacionales utilizados por Python son menores.

En la página oficial de Python, (www.python.org), se realiza una comparación de este lenguaje contra otros obteniendo los siguientes resultados:

1. Python vs Java

El tiempo de desarrollo de un archivo escrito en Python es menor al tiempo que se tomaría en Java, así como su tamaño, ya que puede llegar a ser entre 3 a 5 veces menos las líneas de código escritas por archivo. (Comunidad, Comparing Python to Other Languages, 2001)

2. Python vs Javascript

Python y Javascript tienen una característica en común, y es que ambos lenguajes soportan un estilo de programación que utiliza funciones simples y variables sin necesidad de que pertenezcan a una clase, como es el caso de Java. Pero, a pesar de esto, Python todavía soporta archivos con líneas de código mucho más grande y hace un mejor uso de la reutilización del código basados en una programación orientada a objetos. (Comunidad, Comparing Python to Other Languages, 2001)

3. Python vs R

La sintaxis de Python es más similar al lenguaje actual y a otros lenguajes de programación, a comparación de la sintaxis de R. Debido a eso a Python se le puede llamar como un “lenguaje hablado”. Además, la flexibilidad de Python hace que sea un lenguaje que se utilice para producción de sistemas, ya que la integración con aplicaciones Web se realiza sin mayores complicaciones. (Radcliffe, 2016)

F. Selección de algoritmos

El siguiente paso de la investigación fue el seleccionar los algoritmos a utilizar. Asimismo, los algoritmos utilizados durante el desarrollo de este trabajo son AAMS debido al conjunto de datos ya obtenidos y utilizados en la investigación. Como se mencionó anteriormente, los AAMS son algoritmos que reciben un conjunto de datos, y luego de aplicar el algoritmo, se reconoce el tipo de información obtenida de él. Para propósitos de la investigación, las respuestas de cada algoritmo tienen que ser la clasificación de las emociones del sujeto en dos,

alegría o tristeza. A pesar de que existen seis emociones fundamentales y básicas que todo ser humano posee, se decidió clasificar el conjunto de datos en únicamente dos emociones, las ya mencionadas anteriormente. Esto se dio así ya que las señales EEG del conjunto de datos utilizado, fueron tomadas específicamente para inducir un estado de felicidad y/o tristeza en la persona, siempre para fines de la investigación.

Luego, ya que se tiene un conocimiento del conjunto de datos a ingresar en cada algoritmo, así como el conocimiento de la respuesta obtenida de cada uno de ellos, los algoritmos más recomendables a utilizar e implementar fueron de tipo supervisado. La característica principal que separa a los AAMS de los AAMNS es que en los primeros se conoce el resultado deseado, y se espera que el algoritmo llegue a ese resultado por cuenta propia, haciendo el uso de estos algoritmos los más recomendable para utilizar en la investigación porque se necesita que el algoritmo clasifique las señales EEG en las emociones definidas para la investigación, y no que encuentre patrones de comportamiento en las señales si tal fuera el uso de los AAMNS.

1. Árboles de decisión

Uno de los algoritmos utilizados en la presente investigación son los “Árboles de decisión” de los AAMS. Este algoritmo tiene como base clasificar las entradas provenientes de un conjunto de datos, en las diferentes salidas definidas basándose en las características de estas entradas. Se eligió este algoritmo debido a que la respuesta, en la mayoría de los casos de implementación, de un árbol de decisión es una respuesta binaria, ya sea falso o verdadero si es un valor booleano, o puede ser el número 1 o número 0, si la respuesta es numérica, etc, y las respuestas que el algoritmo tiene que proveer es la clasificación de la señal ya sea si es una señal EEG de tristeza o una señal EEG de alegría.

2. K-Vecinos más Próximos (KNN)

Se ha elegido el algoritmo de KNN ya que la clasificación de este algoritmo se basa en los vecinos del dato a clasificar. Los datos a clasificar son señales continuas recopiladas de los electrodos F3, F4, AF3 y AF4 del dispositivo Emotiv EPOC. Dado a que son señales continuas, el algoritmo de KNN puede aprovechar esta continuidad y clasificar cada señal en base a ella.

3. Redes Bayesianas

Las Redes bayesianas se han utilizado ya que la característica principal de este algoritmo es encontrar características que relacionen fuertemente a los nodos que compone el grafo resultante formado por el propio algoritmo. Estas relaciones podrían indicar la forma en la que las señales EEG de estados emocionales de una persona interactúan entre sí.

G. Módulo de Python utilizado

Dado que la tecnología utilizada fue Python, se decidió utilizar el módulo llamado “Scikit-learn” (<http://scikit-learn.org/stable/>). Este módulo se eligió principalmente por ya proveer la documentación necesaria para implementar los algoritmos deseados, junto con ejemplos de implementaciones de diferentes conjuntos de datos para su mayor comprensión, además, como ya antes se ha mencionado, “Scikit-learn” por estar escrito en Python, cuenta con un lenguaje interpretado y fácil de utilizar. Una de las razones principales por la cual se decidió utilizar este módulo de Python fue la popularidad que mantiene éste sobre otros, como por ejemplo el módulo de “Pybrain” que ya pasó a ser discontinuado.

H. Implementación de los algoritmos

Luego de tener claro los algoritmos a utilizar, juntamente con los conjuntos de datos de entrenamiento y de pruebas, se procedió a realizar la implementación de cada algoritmo. La implementación de cada algoritmo se dio en dos pasos, entrenar al algoritmo y hacer pruebas en base al entrenamiento del algoritmo. Se realizaron tres diferentes implementaciones de los algoritmos en las que se variaba el tamaño del conjunto de datos de entrenamiento, siendo el 70% para la primera, el 80% para la segunda y el 90% para la tercera implementación. Para cada distinto conjunto de datos de entrenamiento se realizó diez veces la ejecución de cada algoritmo. Estas implementaciones, así como la obtención de los conjuntos de datos de entrenamiento utilizando la validación cruzada, se realizó con los métodos ya establecidos del módulo Scikit-learn de Python.

VI. RESULTADOS

A. Pre-procesamiento de datos

En esta etapa se obtuvieron los datos relevantes a utilizar para la implementación de los algoritmos. En la Figura no.19 se muestra la señal en forma gráfica de los datos antes de ser procesados para el sujeto no.20.

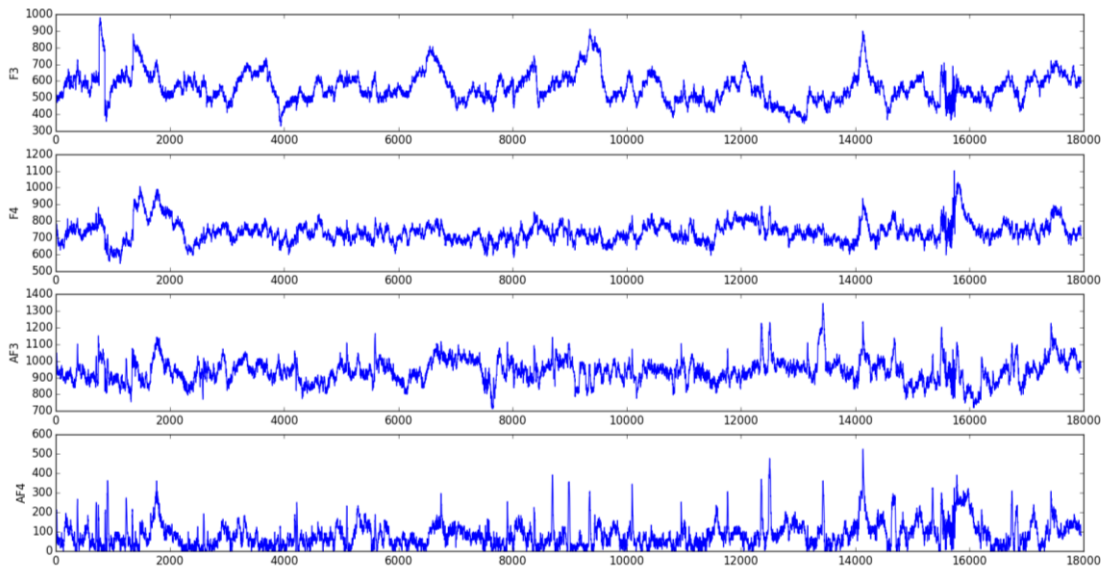


Figura 19. Señal EEG en bruto para el sujeto no.20 de los electrodos F3, F4, AF3 y AF4.

En la Figura no.20 se muestra la aplicación de la transformada de Wavelets, con 4 niveles de descomposición, sobre el electrodo AF3 utilizando una wavelet madre Daubechies 4 (db4), para separar las distintas bandas de frecuencia de las señales EEG.

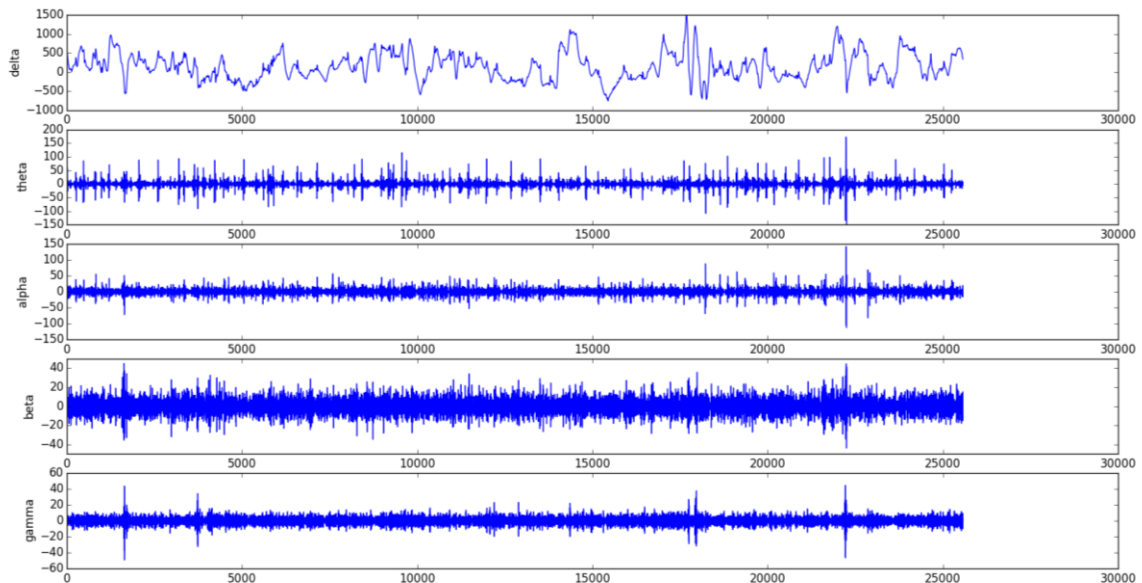


Figura 200. Aplicación de la transformada de Wavelets sobre el electrodo AF3

En la Figura no.21 se muestra la comparación de los segmentos cada electrodo utilizado. El segmento de tristeza se muestra como color rojo, alegría como color azul y neutral como color negro.

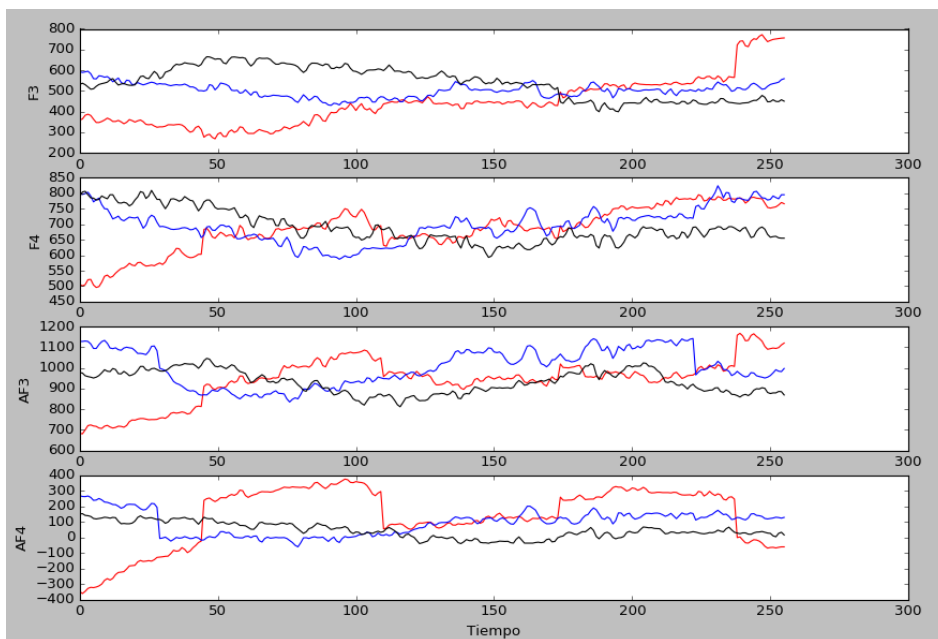


Figura 211. Segmentos de las señales EEG para los electrodos F3, F4, AF3 y AF4 para el sujeto no.4

B. Segmentos utilizados

La cantidad de segmentos utilizados durante la investigación obtenidos luego del pre-procesamiento de datos fue de 1810 segmentos, separados como se muestran en la tabla no.3

Tabla 3. División de segmentos obtenidos del pre-procesamiento de datos

Segmento	Cantidad
HAPPY	436
SAD	456
NEUTRAL	918

C. Datos de entrenamiento

Luego de obtener los segmentos a utilizar, se obtuvieron los conjuntos de entrenamiento y de prueba para implementar los algoritmos. Como se mencionó anteriormente, se realizaron pruebas con tres distintos tamaños de conjunto de datos como se muestran en las tablas no.4, 5 y 6

Tabla 4. Cantidad de segmentos utilizados para el entrenamiento con el 70% de los segmentos

Conjunto	Cantidad de segmentos
Entrenamiento	1267
Prueba	543

Tabla 5. Cantidad de segmentos utilizados para el entrenamiento con el 80% de los segmentos

Conjunto	Cantidad de segmentos
Entrenamiento	1448
Prueba	362

Tabla 6. Cantidad de segmentos utilizados para el entrenamiento con el 90% de los segmentos

Conjunto	Cantidad de segmentos
Entrenamiento	1629
Prueba	181

D. Implementación de los algoritmos

La implementación de los algoritmos se hizo para cada diferente conjunto de datos de entrenamiento obtenidos en el apartado anterior y para cada implementación de los algoritmos se han hecho 10 iteraciones obteniendo los siguientes resultados.

Tabla 7. Efectividad de cada algoritmo con un conjunto de entrenamiento del 70%

No.	Árboles de decisión (%)	Redes Bayesianas (%)	K-Vecino más próximo (%)
1	80.1	33.0	87.7
2	79.9	30.2	86.2
3	76.4	29.7	87.8
4	79.6	34.4	88.0
5	77.0	32.6	87.8
6	79.0	31.5	88.6
7	72.9	31.7	88.6

No.	Árboles de decisión (%)	Redes Bayesianas (%)	K-Vecino más próximo (%)
8	79.0	28.9	867.8
9	76.1	31.1	86.6
10	77.0	29.3	87.1

Tabla 8. Efectividad de cada algoritmo con un conjunto de entrenamiento del 80%

No.	Árboles de decisión (%)	Redes Bayesianas (%)	K-Vecino más próximo (%)
1	82.6	32.6	87.9
2	77.3	34.3	90.3
3	82.9	30.1	90.9
4	79.6	32.9	90.3
5	81.5	29.8	89.5
6	77.6	34.8	90.1
7	82.3	33.4	89.0
8	76.5	32.6	88.7
9	80.1	31.5	89.5
10	79.8	30.9	89.2

Tabla 9. Efectividad de cada algoritmo con un conjunto de entrenamiento del 90%

No.	Árboles de decisión (%)	Redes Bayesianas (%)	K-Vecino más próximo (%)
1	79.6	35.9	92.8
2	82.3	34.3	90.6
3	82.9	32.0	91.7
4	85.1	30.9	92.8
5	86.2	28.7	89.5

No.	Árboles de decisión (%)	Redes Bayesianas (%)	K-Vecino más próximo (%)
6	84.0	31.5	92.8
7	82.3	37.0	89.5
8	82.3	30.9	96.1
9	84.0	29.8	90.6
10	79.6	29.3	90.1

Tabla 10. Resumen de los más altos y más bajos porcentajes presentados por cada algoritmo para cada conjunto de entrenamiento

	Porcentaje del conjunto de entrenamiento	% Más alto	% Más bajo
Árboles de decisión (%)	70	80.1	72.9
Redes Bayesianas (%)	70	33.1	28.9
K-Vecino más próximo (%)	70	88.6	86.4
Árboles de decisión (%)	80	82.9	77.3
Redes Bayesianas (%)	80	34.8	29.8
K-Vecino más próximo (%)	80	90.9	88.7
Árboles de decisión (%)	90	86.2	76.2
Redes Bayesianas (%)	90	35.9	28.7
K-Vecino más próximo (%)	90	96.1	89.5

E. Matrices de confusión

Una matriz de confusión es una herramienta que permite la visualización del desempeño de un algoritmo que se emplea en aprendizaje supervisado. Cada columna de la matriz representa el número de predicciones de cada clase, mientras que cada fila representa a las instancias en la clase real (Breiman, Friedman, & Olshen, 1984). Dicho lo anterior, a continuación se presentan las matrices de confusión de cada algoritmo.

Tabla 11. Matriz de confusión para el algoritmo de árboles de decisión con el grupo de entrenamiento del 70% y una efectividad del 80.1%

	HAPPY	NEUTRAL	SAD
HAPPY	96	21	12
NEUTRAL	16	236	25
SAD	8	26	103

Tabla 12.. Matriz de confusión para el algoritmo de redes bayesianas con el grupo de entrenamiento del 70% y una efectividad del 30.2%

	HAPPY	NEUTRAL	SAD
HAPPY	123	5	1
NEUTRAL	227	31	19
SAD	118	9	10

Tabla 13. Matriz de confusión para el algoritmo de K-Vecino más próximo con el grupo de entrenamiento del 70% y una efectividad del 88.4%

	HAPPY	NEUTRAL	SAD
HAPPY	111	14	4
NEUTRAL	14	255	8
SAD	3	20	114

Tabla 14. Matriz de confusión para el algoritmo de árboles de decisión con el grupo de entrenamiento del 80% y una efectividad del 82.6%

	HAPPY	NEUTRAL	SAD
HAPPY	71	12	8
NEUTRAL	9	154	11
SAD	10	13	74

Tabla 15. Matriz de confusión para el algoritmo de redes bayesianas con el grupo de entrenamiento del 80% y una efectividad del 33.4%

	HAPPY	NEUTRAL	SAD
HAPPY	84	6	1
NEUTRAL	132	28	14
SAD	73	15	9

Tabla 16. Matriz de confusión para el algoritmo de K-Vecino más próxima con el grupo de entrenamiento del 80% y una efectividad del 89.0%

	HAPPY	NEUTRAL	SAD
HAPPY	76	12	3
NEUTRAL	5	136	6
SAD	9	8	80

Tabla 17. Matriz de confusión para el algoritmo de árboles de decisión con el grupo de entrenamiento del 90% y una efectividad del 86.2%

	HAPPY	NEUTRAL	SAD
HAPPY	37	5	4
NEUTRAL	6	78	1
SAD	3	6	41

Tabla 18. Matriz de confusión para el algoritmo de redes bayesianas con el grupo de entrenamiento del 90% y una efectividad del 30.9%

	HAPPY	NEUTRAL	SAD
HAPPY	42	2	2
NEUTRAL	69	8	8
SAD	40	6	4

Tabla 19. Matriz de confusión para el algoritmo de K-Vecino más próxima con el grupo de entrenamiento del 90% y una efectividad del 89.0%

	HAPPY	NEUTRAL	SAD
HAPPY	41	4	1
NEUTRAL	4	77	4
SAD	3	6	41

VII. DISCUSIÓN

Como se ha mencionado anteriormente, los electrodos utilizados para la recolección de los datos a utilizar son F3, F4, AF3 y AF4, siendo elegidos por el aumento de ondas Alpha y Beta que ocurre dentro del cerebro durante los estados de alerta que un individuo puede tener. Como se puede observar en la Figura no.19, las señales EEG graficadas para cada electrodo no presentan ningún tipo de variación con respecto a las demás a pesar de que el sujeto, en este caso el sujeto no. 20, estuvo expuesto a 40 diferentes estímulos, 10 de alegría, 10 de tristeza y 20 neutrales. Debido a esto, se aplicó la transformada de Wavelets a cada electrodo. Un ejemplo de esto se puede notar en la Figura no. 20, en la que aplicó la transformada sobre el electrodo AF3, separando las 5 bandas de frecuencia que da como resultado Wavelets. Como se puede observar en esta figura, la presencia de las ondas Alpha, Beta y Gamma es mayor a las demás, confirmando lo dicho en el marco teórico (Ramírez, 2008)

La segmentación de los datos, luego de ser pre-procesados, fue un paso fundamental del proceso ya que se determina los bloques de información que contienen la información relevante para la investigación. Estos segmentos se pueden observar en la Figura no.21 y como se puede notar, ya se muestran los diferentes segmentos para cada electrodo. Esto se pudo obtener luego de haber aplicado la transformada. Esta parte de segmentación dio como resultado 1810 segmentos, distribuidos como se muestra en la Tabla 3. La mayor parte de los segmentos corresponde a un estado neutral debido a que los estímulos neutrales fueron a los que más tiempo y veces estuvieron expuestos los sujetos.

Como se puede observar en las tablas 4, 5 y 6, se cambió el porcentaje del tamaño del conjunto de entrenamiento para poder comparar la eficacia de cada algoritmo con diferentes conjuntos de entrenamiento. Como se puede observar en el resumen de la Tabla no. 10, cada algoritmo tuvo un porcentaje más alto y uno más bajo en las 10 iteraciones. El mejor porcentaje, con un conjunto de entrenamiento de 70%, lo obtuvo el algoritmo KNN, con 88.6% respectivamente. Esto se debe a la forma en que el algoritmo clasifica los datos. Como se sabe, el algoritmo KNN clasifica en según el vecino más próximo y en este caso, clasifica en base a la señal más próxima. Como son señales continuas, el algoritmo puede encontrar la clase del vecino de la señal sin tanta complejidad, haciendo que su efectividad sea mayor que los demás. A su vez, como se puede observar en la misma tabla, el algoritmo que menor porcentaje de efectividad es el algoritmo de las Redes Bayesianas, con un 28.9% de efectividad. Esto se puede deber a los valores que le colocó el algoritmo a cada arista para para la clasificación. Asimismo, las multiplicaciones que efectúa el algoritmo de Redes Bayesianas pueden dar 0, haciendo que la efectividad del algoritmo disminuya considerablemente, como se ve reflejado en la Tabla 10.

Con respecto al algoritmo de Árboles de decisión, los resultados pueden ser aceptados ya que el mayor porcentaje de este, siempre con un conjunto de entrenamiento del 70%, fue de 80.1%, no muy alejado de la mayor efectividad del algoritmo KNN ya antes mencionada.

En la misma Tabla no.10 se muestran los resultados de los de las implementaciones con conjuntos mayores de entrenamiento, un 80 y 90 por ciento respectivamente, siendo evidente que se mantiene el mismo patrón de resultados en los algoritmos. El mejor de estas implementaciones resulta siendo el algoritmo KNN, teniendo como mejores porcentajes 90.9% para el 80% y 96.1% para el 90%. De esta manera se puede observar que entre más grande sea el

conjunto de entrenamiento, la efectividad del algoritmo se verá aumentada considerablemente. Esto se hace notar en la mejoría que tiene el algoritmo entre el 70 y el 90 por ciento del conjunto de datos para entrenamiento, 5% de mejoría entre uno y otro. Un punto importante a tomar en cuenta al momento de escoger el tamaño del grupo de entrenamiento es el sobreajuste que realiza el algoritmo. El sobreajuste es el efecto de entrenar un algoritmo con datos a los que se conoce su resultado. El efecto principal de el sobreajuste se puede notar cuando un algoritmo se desempeña correctamente con los datos de entrenamiento, pero su precisión es notablemente más baja con los datos de prueba ya que el algoritmo ha memorizado los datos ingresados en el entrenamiento y no ha podido generalizar las reglas para predecir los datos de prueba. (Lopez, 2016). Debido a esto se utilizó la Validación Cruzada al momento de entrenar a los algoritmos ya que uno de los propósitos principal de este es evitar el sobreajuste de cualquier algoritmo.

A su vez, se puede notar que el algoritmo de Redes Bayesianas no varía tanto en su mejoría al momento de clasificar las señales sin importar que el tamaño del conjunto de datos de entrenamiento haya aumentado un 20%. La mejoría presentada por este algoritmo fue de un 2.8 por ciento, ya que aumentó de un 33.1% a un 35.9%, con el 90% del conjunto de datos para entrenamiento.

En las tablas que contienen la información de la matriz de confusión para cada algoritmo cabe destacar la cantidad de aciertos que tuvo cada algoritmo, con cada grupo de entrenamiento. El número que se encuentra en la diagonal de cada matriz es la cantidad de clasificaciones correctas que hizo cada algoritmo. Las demás casillas de la matriz son falsos positivos clasificados así por los algoritmos. El término “Falso positivo” se refiere a la ocurrencia de una clasificación cuando el resultado correcto no es esa. (Jain & Zongker, 1997).

Como se puede notar en las tablas de la 11 a la 19, en todas las matrices de confusión la intersección del estado “NEUTRAL” es la que obtiene el mayor número de aciertos debido a que la cantidad de señales de este tipo era mucho mayor a las de “ALEGRÍA” y “TRISTEZA”.

Dicho lo anterior, se puede observar que cada algoritmo puede clasificar de mejor manera cierta emoción. Si notamos en la Tabla no.11, con el 70% del grupo de entrenamiento y una efectividad del 80.1%, el algoritmo de árboles de decisión puede clasificar de mejor manera las señales de tipo “SAD” ya que de la totalidad de las muestras de esta señal, 137 señales, clasificó correctamente 103 señales.

Hablando específicamente del algoritmo que obtuvo mejores porcentajes de efectividad en los 3 tipos de implementación, el algoritmo KNN, se puede notar visiblemente que en según sus matrices de confusión en las tablas 13, 16 y 19, la emoción que mejor clasifica es para la señales de “HAPPY”.

Para esta investigación es importante mencionar que no se tuvo ningún contacto con alguna de las personas de las que se obtuvieron las señales, ya que fueron señales obtenidas de otra investigación ya antes mencionada. Asimismo, en esa investigación se utilizó el algoritmo de Máquina de Soporte Vectorial, Support Vector Machine en inglés, y en la que se obtuvo un mayor porcentaje de efectividad del 81% con un grupo de entrenamiento del 70% de la muestra de datos. Debido a esto, el algoritmo que mejor clasifica las señales fue el algoritmo KNN, utilizando el 70% de la muestra de los datos.

VIII. CONCLUSIONES

Luego de analizar a detalle los resultados obtenidos en la presente investigación, se puede llegar a las siguientes conclusiones:

- Se investigaron derivados de los AAM y se seleccionaron tres para la investigación: Árboles de Decisión, Redes Bayesianas y el K-Vecino más próximo, o KNN.
- Debido a su bajo rendimiento en la clasificación de las señales, el algoritmo de Redes Bayesianas no es el más recomendable a utilizar para este tipo de clasificación.
- El algoritmo que mejores resultados obtuvo fue el algoritmo KNN debido a la continuidad de las señales que se utilizaron.
- La mejora que obtuvo el algoritmo de KNN fue de 5.8 por ciento. Esta mejora se puede considerar importante ya que la efectividad del algoritmo fue mayor del 95% en la clasificación, y gracias a la Validación Cruzada el sobreajuste no provocaría problemas en esta efectividad.
- A pesar de que el algoritmo de Redes Bayesianas obtuvo una mejora de un 2.8%, siendo casi la mitad de la mejora de la efectividad del algoritmo KNN, este aumento no se puede considerar relevante ya que, a pesar de la mejora, el algoritmo clasifica correctamente la tercera parte de la data que se le ingresa. Que un algoritmo clasifique correctamente uno de cada tres datos no lo hace confiable en la clasificación.
- En base a las matrices de confusión del algoritmo KNN, se puede deducir que este clasifica de mejor manera las señales de tipo “NEUTRAL” y “HAPPY”, obteniendo los mejores porcentajes de clasificación de esta señal.

IX. RECOMENDACIONES

Luego de realizar la presente investigación, se le recomienda al lector seguir las siguientes recomendaciones para el uso de la misma:

- Se recomienda utilizar un mayor número de muestras de personas, ya que una de las características de los AMM es que aprenden de grandes cantidades de datos.
- Si en un futura investigación se decide utilizar el dispositivo Emotiv EPOC, se recomienda utilizar más electrodos que los utilizados en la toma de señales de la investigación, F3, F4, AF3 y AF4, para obtener una mejor calidad de la señal, así como para obtener más emociones y poder clasificar a las mismas.
- Se recomienda utilizar más derivados de los AMM para tener un estudio más amplio de los algoritmos en la clasificación de emociones humanas.

X. BIBLIOGRAFÍA

- Barea, R. (2002). *Instrumentacion Biomedica: Electroencefalografia*. España: Universidad de alcala.
- Bates, D., & Watts, G. (1998). *Nonlinear Regression Analysis and Its Applications*. New York: Duxbury Press.
- Bauer, J. (1996). *An Introduction to Python*. Linux Journal.
- Bishop, C. (1995). *Neural Networks for Pattern Recognition*. New York: Oxford Universiti Press, Inc.
- Bishop, C. (2006). *Pattern Recognition and Machine Learning*. New York: Springer.
- Brownlee, J. (8 de Junio de 2014). *machinelearningmastery.com*. Obtenido de Java Machine Learning: <https://machinelearningmastery.com/java-machine-learning/>
- Brownlee, J. (03 de 08 de 2016). *Machine Learning Mastery*. Obtenido de <http://machinelearningmastery.com/how-to-evaluate-machine-learning-algorithms/>
- Chai Tong, Y., Woo San, S., & al, e. (s.f.). Classification of Human Emotions from EEG Signals using Statistical Features and Neral Network. *International Journal of Integrated Engineering*.
- Comunidad. (2001). *Comparing Python to Other Languages*. Obtenido de pyhton.org: <https://www.python.org/doc/essays/comparisons/>
- Comunidad. (25 de Abril de 2017). *tutorialzine.com*. Obtenido de 10 Machine Learning Examples in JavaScript: <https://tutorialzine.com/2017/04/10-machine-learning-examples-in-javascript>
- Devijver, P., & Kittler, J. (1982). *Pattern Recognition: A Statistical Approach*. Londres: Prentice-Hall.
- Duda, R., & Hart, P. (1973). *Pattern Classification and Scene Analysis*. London: John Wiley & Sons.
- Elder, J. (1997). *Introduction to Machine Learning and Pattern Recognition*. Obtenido de Lassonde University EECS Department York: http://www.eecs.yorku.ca/course_archive/2011-12/F/4404-5327/lectures/01%20Introduction.pdf
- Everitt, B. (2011). *Cluster Analysis*. United Kingdom: Wiley.
- Flanagan, D., & Ferguson, P. (2002). *JavaScript: The Definitive Guide*. London: Read Write Web.
- Flórez, R., & Fernández, J. (2008). *Las Redes Neuronales Artificiales*. La Coruña: Gesbiblo, S.L.

- Fredrickson, B. L. (2001). *Positive Emotions*. Nueva York: Guilford Press.
- Fredrickson, B., & Levenson, R. (2001). *The Role of Positive Emotions in Positive Psychology*. Nueva York: Guilford Press.
- Friesen, P. E. (1982). "Measuring facial movement with the facial action coding system," in *Emotion in the Human Face*. New York: Cambridge University Press.
- García, H. (2016). *Desarrollo de SDK enfocado en el procesamiento de señales EEG para la clasificación y el reconocimiento de emociones*. Ciudad de Guatemala.
- Goleman, D. (1993). *La inteligencia Emocional*. Barcelona: Kairos. Obtenido de <http://www.psicoadactiva.com/emocion.htm>.
- González, E. (2011). Megaproyecto INNOVA. Módulo de Interfaz Cerebro - Máquina. Control no convencional e implementación de un prototipo de una silla de ruedas eléctrica a través de la electroencefalografía. *Universidad del Valle de Guatemala*.
- Good, I. (1951). *Probability and the Weighing of Evidence*. London: Charls Griffin and Company.
- Goslin, J., & Joy, B. (2005). *The Java language specification*. Chicago: Addison-Wesley.
- Hastie, T., Tibshirani, R., & Friedman, J. (2001). *The elements of statistical learning*. New York: Springer Verlag.
- Hinton, G. (1996). *Advances in Neural Information Processing Systems*. New York: MIT Press.
- Horlings, R. (2008). *Emotion Recognition Using Brain Activity, Department of Mediamatics*. Delft University of Technology.
- Jain, A., & Dubes, R. (1988). *Algorithms for Clustering Data*. Prentice Hall.
- Jordan, I., & Bishop, M. (2004). *Neural Networks*. Florida: Chapman & Hall.
- Kotsiantis, S. (2007). Supervised machine learning: a review of classification techniques. *Informática*, 31.
- Kotsiantis, S. (2012). Supervised Machine Learning: A Review of Classification Techniques. *International Journal of Machine Learning and Computing (IJMLC)*, 560-563.
- Kotsos, T. (19 de septiembre de 2016). *Las ondas cerebrales y los estados de la mente*. Obtenido de BibliotecaPleyades: http://www.bibliotecapleyades.net/ciencia/ciencia_brain69.htm
- KTian, Z., & Raghu, R. M. (1994). *An Efficient Data Clustering Method for Very Large Databases*. Santiago, Chile: 20th VLDB Conference.
- Lopez, R. (10 de Octubre de 2015). *Machine Learning con Python*. Obtenido de Raul E. Lopez Briega: <https://relopezbriega.github.io/blog/2015/10/10/machine-learning-con-python/#Scikit-Learn>

- MacQueen, J. (1967). Some Methods for classification and Analysis of Multivariate Observations. En J. MacQueen, *Berkeley Symposium on Mathematical Statistics and Probability* (págs. 281-297). California: University of California Press.
- Mallat, S. (1989). A theory for multiresolution signal decomposition: The Wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. *IEEE*, 11.
- Martelli, A. (2007). *Python. Guía de referencia*. El Salvador: Anaya Multimedia.
- Moya, R. (03 de 08 de 2016). *Jarroba*. Obtenido de <http://jarroba.com/que-es-el-machine-learning/>
- Murphy, K. (2012). *Machine Learning, A Probabilistic Perspective*. London, England: Massachusetts Institute of Technology.
- Neocleous, C., & Schizas, C. (2002). *Artificial Neural Network Learning: A Comparative Review*. Berlin: Springer, Heidelberg.
- Nilsson, N. (1965). *Learning machines*. New York: MacGraw-Hil.
- Noppadon, J. S. (2013). Real-Time EEG-Based Happiness Detection System. *The Scientific World Journal*, 12.
- Petrantonakis, P., & Hadjileontiadis, L. (2009). Emotions recognition from EEG using higher Order crossing. *Information Technology in Biomedicine*, 186-197.
- Pfutzner, D., Leibbrandt, R., & Powers, D. (2009). Characterization and evaluation of similarity measures for pairs of clustering. En *Knowledge and Information Systems* (págs. 361-365). Springer.
- Radcliffe, T. (16 de November de 2016). *Python versys R for machine learning and data analysis*. Obtenido de [opensource.com: https://opensource.com/article/16/11/python-vs-r-machine-learning-data-analysis](https://opensource.com/article/16/11/python-vs-r-machine-learning-data-analysis)
- Ramírez, R. (2008). Detecting emotion from EEG signals using the emotiv epoc device. *University Pompeu Fabra*, 11.
- Richiert, W. a. (2016). *Building Machine Learning System with Python*. Packt Publishing.
- Russell, J. (1980). "A circumplex model of affect," *Journal of Personality and Social Psychology*. Google Scholar.
- S., A., & Vishwanatan, S. (2008). Introduction to Machine Learning. *Cambridge Univestity Press*.
- Sanjeev, A., & Barak, B. (2009). *Computational Complexity*. Cambridge: McGraw-Hill.

- Shah, K. (6 de Junio de 2016). *R-bloggers*. Obtenido de What are the Best Machine Learning Packages in R?: <https://www.r-bloggers.com/what-are-the-best-machine-learning-packages-in-r/>
- Sherwood, J. &. (2009). On classification ability of wavelet features for EEG based brain-computer interfaces. *IEEE*, 23.
- Taiwo, O. (2010). *Types of Machine Learning Algorithms, New Advances in Machine Learning*. Reino Unido: Universidad de Portsmouth, Inglaterra.
- Tapas, D. (2002). *A local search approximation algorithm for k-means clustering*. Barcelona: ACM Press.
- Thevenet, D. (2008). *Instrumentación Biomédica con Seguridad*. Montevideo: Universidad de la República de Uruguay.
- Van Rossum, G., & Drake, F. (2000). *An Introduction to Python*. Nueva York: Network Theory LTD.
- William, G. (1991). *Handbook of the Logistic Distribution*. New York: Marcek Dekker, Inc.
- Zhang, S., Zhang, C., & Yang, Q. (2002). Data Preparation for Data Mining. En S. Zhang, C. Zhang, & Q. Yang, *Applied Artificial Intelligence*. (págs. 451-462).

XII. GLOSARIO

[1] Evaginación: Protuberancia o saliente hueco de un conducto o cavidad orgánicos.

[2] Invaginación: Formación del repliegue de una membrana, capa de tejido u hoja blastodérmica que se dirige hacia el interior de una pared intestinal.