Clasificación de exoplanetas utilizando modelos de machine learning

Luis Fernando Rascón Calderón

UVG UNIVERSIDAD DEL VALLE DE GUATEMALA

UNIVERSIDAD DEL VALLE DE GUATEMALA
Facultad de Ciencias y Humanidades

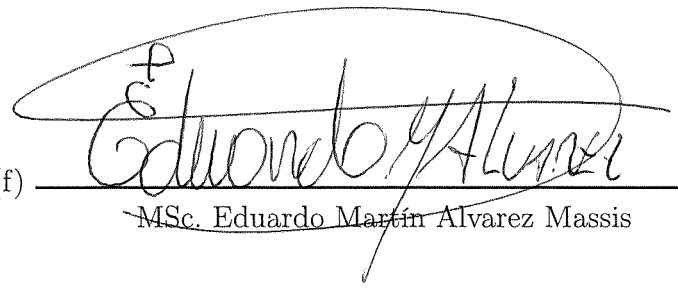
Clasificación de exoplanetas utilizando modelos de machine learning

Trabajo de graduación presentado por
Luis Fernando Rascón Calderón
para optar al grado académico de Licenciado en Física

Guatemala,

2025

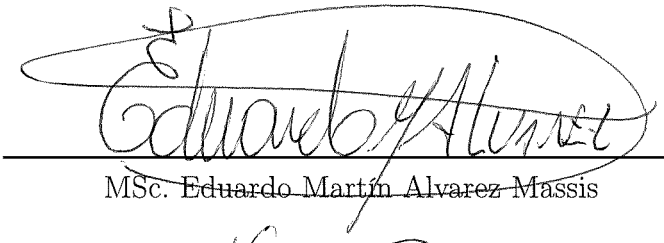
Vo.Bo.:



(f) _____

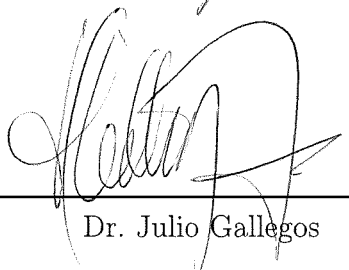
MSc. Eduardo Martín Alvarez Massis

Tribunal Examinador:



(f) _____

MSc. Eduardo Martín Alvarez Massis



(f) _____

Dr. Julio Gallegos

(f) _____

MSc. Antonio Méndez

Fecha de aprobación: Guatemala, 27 de junio de 2025.

Prefacio

El presente trabajo surge a partir de la fascinación por los exoplanetas y los frutos que pueden llegar a dar los estudios de estos; además de la curiosidad por el potencial del uso de las herramientas de Machine Learning en el campo de la astronomía. Esta investigación entrelaza ambos campos, siendo así un aporte más a un puente que cada año se hace más grande y se espera pueda ser un aporte significativo a la comunidad científica relacionada.

Una vez dado por completo el presente trabajo de investigación, se hace el profundo agradecimiento a:

- Mi papá, Miguel Rascón†, que es la razón por la que amo la física y la ciencia en general. Por enseñarme a ser curioso y no quedarme con dudas, solo con respuestas. Por motivarme a levantar la mirada al cielo nocturno y admirar la belleza del universo que nos rodea.
- Mi familia en Guatemala y México, especialmente a mi mamá Jeniffer Calderón, mi hermano José Rascón Calderón y a mis abuelitos Alba Pineda, Eva y Miguel Ángel Rascón, por haberme inspirado a seguir mis sueños y darme el apoyo moral necesario a lo largo de la licenciatura. A mis tías Leli, Tili, Lucy, Wendy y mi tío Christian, su amor incondicional me motivó a siempre seguir adelante sin importar las adversidades.
- Mis mejores amigos y compañeros de carrera, por hacer el tiempo en la licenciatura una experiencia amena y divertida. Siempre me impulsaron a ser una mejor persona y un mejor científico.
- MSc. Eduardo Álvarez, por su apoyo como director de la carrera y por su colaboración como asesor principal del trabajo de investigación. Su ayuda fue como un faro que me permitió tener un camino claro a lo largo de la licenciatura.
- Dr. Julio Gallegos por su apoyo durante el proceso de definición y delimitación de la investigación. Fue su interés y apoyo en el proyecto lo que me motivo a seguir adelante con dicho trabajo.
- Todos los profesores y mentores que me acompañaron en la carrera, por haber usado su valioso tiempo en enseñar a una nueva generación, buscando no solo formar profesionales ejemplares, sino que también personas de bien que lideran la comunidad con la que se rodean.

Índice

Lista de figuras

Figura **Página**

Lista de cuadros

Cuadro **Página**

Resumen

En el presente trabajo se aborda el entrenamiento de algoritmos de machine learning para la clasificación de exoplanetas en cuatro tipos: Terrestres, Súper Tierras, Neptunianos y Gigantes Gaseosos. Los datos empleados fueron extraídos del *NASA Exoplanet Catalog* y del *NASA Exoplanet Archive*. Los algoritmos utilizados fueron XGBoost, Árbol de Decisión, *Random Forest* y *K-Nearest Neighbors*. Tras la preparación de la base de datos y el entrenamiento de los modelos, se procedió a su evaluación. Los métodos basados en árboles de decisión mostraron el mejor desempeño, siendo XGBoost el mejor con un *F1-score* promedio de 0.9822 (desviación estándar: 0.0087) en 10 iteraciones. Todos los modelos presentaron dificultades al distinguir entre Súper Tierras y Neptunianos. El algoritmo *K-Nearest Neighbors* obtuvo el rendimiento más bajo ($f1 - score$ promedio de 0.8100 y desviación estándar de 0.395), atribuible al desequilibrio de clases en la base de datos, es decir, la proporción de exoplanetas Terrestres es notablemente menor a la del resto de clasificaciones. Los métodos basados en árboles identificaron la masa y el radio planetario como las variables más relevantes, dicho resultado es apoyado por la literatura al buscar caracterizar exoplanetas. Como oportunidades de profundización, se propone explorar una mayor cantidad de hiperparámetros para cada algoritmo e investigar variantes de *K-Nearest Neighbors* adaptadas a conjuntos de datos desbalanceados.

Abstract

This work addresses the training of machine learning algorithms for the classification of exoplanets into four categories: Terrestrial, Super-Earths, Neptunian, and Gas Giants. The data were obtained from the NASA Exoplanet Catalog and the NASA Exoplanet Archive. The algorithms applied were XGBoost, Decision Tree, Random Forest, and K-Nearest Neighbors. After preparing the dataset and training the models, their performance was evaluated. Tree-based methods demonstrated the best results, with XGBoost achieving the highest performance, reaching an average F1-score of 0.9822 (standard deviation: 0.0087) across 10 iterations. All models showed difficulties in distinguishing between Super-Earths and Neptunians. The K-Nearest Neighbors algorithm obtained the lowest performance (average $f1\text{-}score$ of 0.8100 and standard deviation of 0.395), mainly due to class imbalance in the dataset, since the proportion of Terrestrial exoplanets is notably smaller than that of the other categories. Tree-based methods identified planetary mass and radius as the most relevant variables, a result consistent with the literature on exoplanet characterization. As future work, it is proposed exploring a wider range of hyperparameters for each algorithm and investigating variants of K-Nearest Neighbors adapted to imbalanced datasets.

CAPÍTULO 1

Introducción

No estamos solos en el Universo. Existen miles de mundos, que así como el nuestro, orbitan una estrella tan única y especial como el Sol. La existencia de estos mundos fuera del sistema solar fue especulada por siglos, afortunadamente, desde el siglo XX contamos con la tecnología adecuada para poder confirmar su presencia.

Gracias al desarrollo de la Astronomía en las décadas siguientes a 1940 se han logrado desarrollar varios métodos que explotan un fenómeno físico específico para poder detectar un planeta, algunos de estos son: variación de velocidad radial, tránsito, microlentes gravitacionales y detección visual directa. A partir de estas mediciones, se puede determinar una gran cantidad de propiedades del exoplaneta y su órbita alrededor de su estrella. Utilizando estas propiedades, se pueden hacer estimaciones del tipo de planeta que es.

A pesar de que no existe una convención oficial sobre las categorías de planetas que existen, la Administración Nacional de Aeronáutica y el Espacio (NASA por sus siglas en inglés) propone en su *NASA Exoplanet Catalog* 4 tipos fundamentales: Terrestre, el cual consiste en planetas rocosos con características similares a la Tierra, pero más pequeños; Súper Tierras, que como indica su nombre, también son rocosos y con similitudes a la Tierra, pero de mayor tamaño; Neptunianos, son planetas gaseoso con tamaño similar a Neptuno o Urano y con atmósferas compuestas principalmente por helio e hidrógeno; Gigantes Gaseosos, son planetas con tamaños similar o mayores a Júpiter o Saturno.

CAPÍTULO 2

Objetivos

2.1. Objetivo general

Desarrollar modelos de machine learning de alta precisión y confianza para la clasificación exoplanetas según datos recopilados por telescopios.

2.2. Objetivos específicos

- Crear una base de datos adecuada para el uso de modelos de machine learning por medio de procesos de preparación y limpieza de datos.
- Evaluar individualmente el desempeño de cada modelo de machine learning utilizado.
- Determinar qué modelo de machine learning tuvo un mejor desempeño y por qué se dio esto.

CAPÍTULO 3

Justificación

La humanidad ha sido caracterizada por su inherente curiosidad y deseo por conocimiento. Lo que una vez llevó a descubrir el fuego, el día de hoy motiva a explorar cada rincón del universo. Dentro de esta exploración se encuentra el descubrimiento y clasificación de exoplanetas. Una mejor comprensión de estos podría llevar a responder preguntas fundamentales como la formación de la Tierra y el sistema solar, así como el lugar de la humanidad en el universo.

El estudio de exoplanetas es un campo reciente y con gran potencial y la aplicación de herramientas modernas como lo es el *machine learning* podría impulsar el desarrollo de nuevas técnicas y descubrimientos trascendentales. El presente trabajo se enfoca en el desafío de clasificación, proponiendo e investigando el uso de modelos de *machine learning* como una herramienta más eficiente y precisa. La importancia de este trabajo radica en la aplicación de dicho aprendizaje automático al campo de la astronomía de exoplanetas, principalmente en su clasificación en categorías.

CAPÍTULO 4

Marco teórico

4.1. Introducción a exoplanetas

Así como el Sol está rodeado por 8 cuerpos celestes a los que llamamos planetas, las demás miles de millones de estrellas en las miles de millones de galaxias también pueden estar acompañadas por sus propios planetas, cada uno tan único como el sistema solar donde se encuentra la Tierra. Desde los tiempos de la Antigua Grecia se pensaba la idea de la existencia de otros mundos. Epicuro de Samos (341 a. C - 270 a. C) dijo "Hay infinitos mundos que pueden ser iguales o completamente distintos al nuestro."(Perryman, 2011).

El primer exoplaneta del cual se sospechó (con base en evidencia observacional) su existencia fue Errai Ab (Perryman, 2011). Este es un gigante de gas que orbita alrededor de "γ Cep". Este fue un hito enorme en el campo de la Astronomía y dio lugar a una explosión de descubrimientos en las décadas siguientes. Otro descubrimiento de gran importancia fue el de "51 Peg b"por Michel Mayor y Didier Queloz (Johnson et al., 2024). Este fue el primero en su clase: exoplanetas orbitando estrellas como el Sol, una estrella de secuencia principal. Hasta Julio de 2024, se conoce la existencia confirmada de 5741 exoplanetas confirmados (NASA, 2024).

4.2. NASA Exoplanet Catalog y Planetary Systems Composite Data

El NASA Exoplanet Catalog es un servicio en línea actualizado constantemente. Esta información de más de 5600 planetas es presentada junto a un modelo interactivo en 3D de una representación artística del planeta según sus características medidas y calculadas. Además, este catálogo permite filtrar los planetas mostrados según su clasificación: Terrestre, Súper Tierra, Neptuniano, Gigante de gas; por el método de descubrimiento (en inglés): *Eclipse Timing Variations*, *Astrometry*, *Transit*, *Pulsar Timing*, *Transit Timing Variations*, *Imaging*, *Microlensing*, *Radial Velocity*, *Disk Kinematics*, *Orbital Brightness Modulation*; y por la misión o instalación que lo descubrió (NASA Exoplanet Science Institute, 2024).

La definición de cada tipo de exoplaneta según la NASA (NASA, 2025) es la siguiente: Los de tipo Terrestres (*Terrestrials*) son planetas rocosos de tamaño similar al de la Tierra o menores. Los planetas de tipo Súper Tierras (*Super Earth*) son planetas del doble de tamaño que la Tierra pero menores al de Neptuno, además, pueden ser planetas rocosos o gaseosos y en cuanto a masa, pueden encontrarse entre el doble de la masa de la Tierra hasta 10 veces la masa de esta. Los Gigantes Gaseosos (*Gas Giants*) son planetas del tamaño de Júpiter y están compuestos principalmente por helio e hidrógeno. Finalmente, los Neptunianos (*Neptune-Like*) son planetas gaseosos de proporciones similares a las de neptuno (unas 4 veces el tamaño de la Tierra y más de 10 veces su masa).

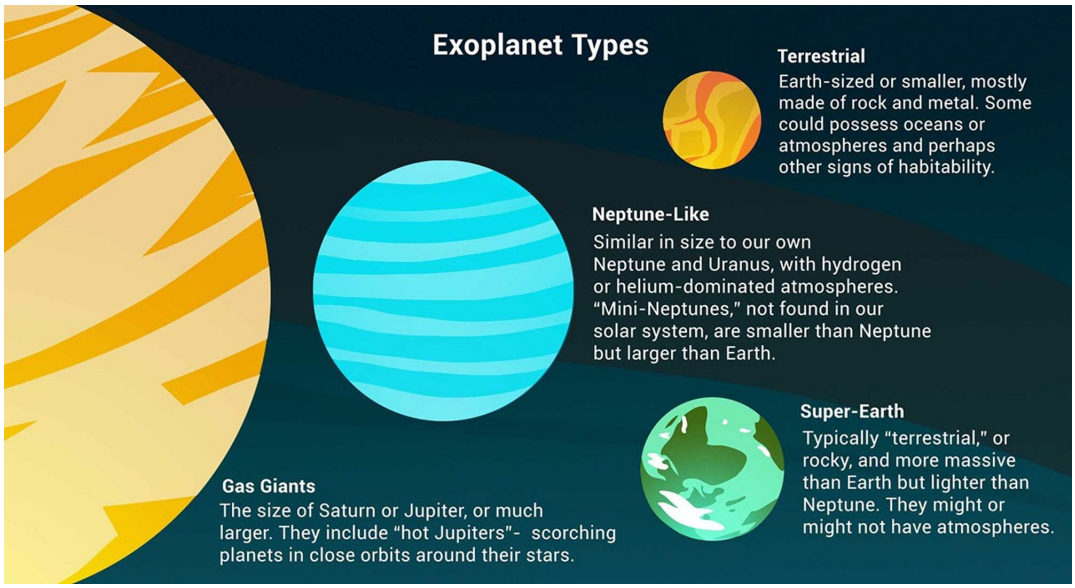


Figura 4.1: Ilustración y explicación de los 4 tipos de planeta según la NASA, NASA/JPL-Caltech, 2025.

(NASA, 2025)

Los métodos consisten en lo siguiente: *Eclipse Timing Variation* es un método utilizado en sistemas binarios, es decir, de dos estrellas orbitando entre sí. Así como la Luna puede eclipsar el Sol, las estrellas se eclipsan desde el punto de vista de la Tierra. Este eclipse ocurre en intervalos concretos, sin embargo, la presencia de un planeta en el sistema puede afectar el periodo (The Planetary Society, n.d.).

Astrometry o astrometría consiste en mediciones de alta precisión de la posición y movimiento de estrellas. A partir de perturbaciones en el movimiento de una estrella, se puede deducir la existencia de un planeta orbitando (European Space Agency, 2019).

Transit o tránsito consiste en detectar un exoplaneta cuando pasa frente a ella, desde la perspectiva de la Tierra. El pasar de un planeta delante de su estrella provocará que el brillo de esta disminuya característicamente en tiempos y duraciones periódicas (The Planetary Society, n.d.). En caso de ser varios planetas, se utiliza el *Transit Timing Variation*. Este utiliza el mismo método de bloqueo de luz de la estrella, pero evalúa cada cuanto tiempo ocurre para determinar si hay disminuciones adicionales que pueden ser ocasionadas por otros planetas presentes en la órbita.

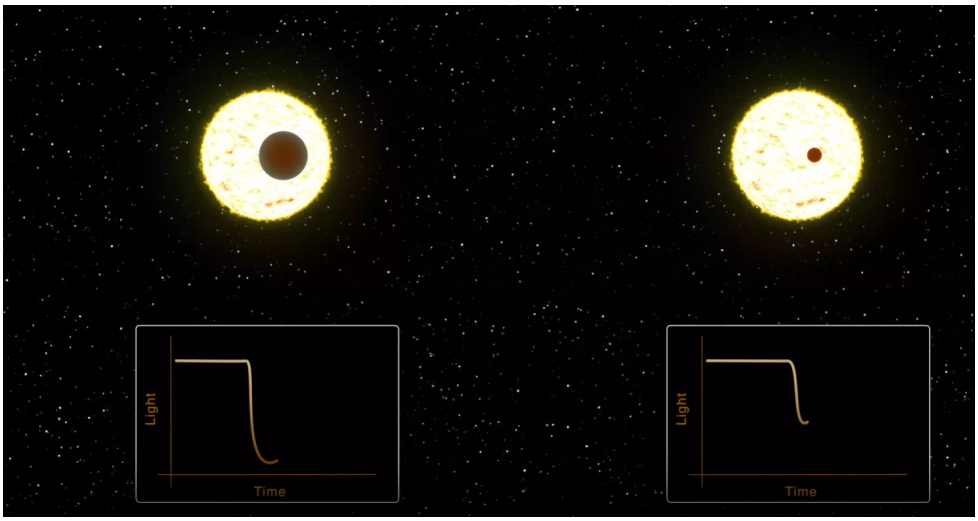


Figura 4.2: Representación de detección por método de tránsito de exoplaneta delante de su estrella. Elaborado por Scott Wiessinger y publicado por NASA Scientific Visualization Studio, 2018.
(NASA Visualization Studio, 2018)

Imaging Microlensing o microlente gravitacional explota un fenómeno predicho por el mismo Albert Einstein. Einstein descubrió que los cuerpos masivos como las estrellas distorsionan el espacio-tiempo a su alrededor. Esta distorsión permite que la luz proveniente de una estrella A, pueda viajar alrededor de la estrella B la cual se encuentra entre A y el observador. Si la estrella B es orbitada por un planeta, este también puede generar el fenómeno de microlente gravitacional y provocar un aparente aumento en el brillo de la estrella A (Optical Gravitational Lensing Experiment, n.d.).

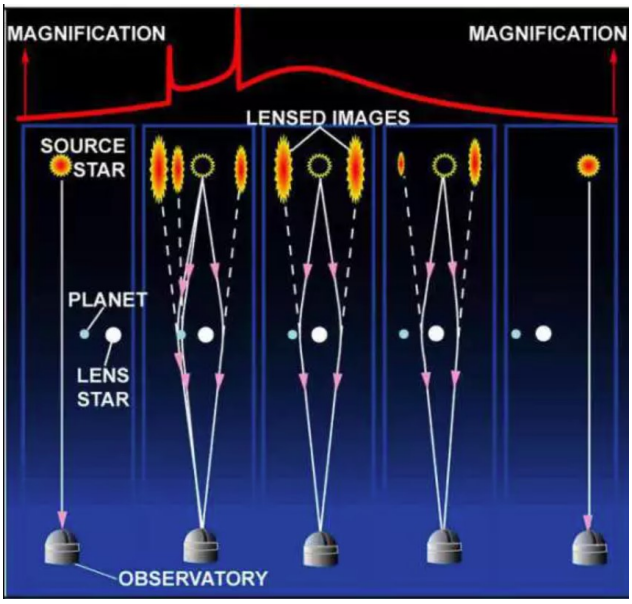


Figura 4.3: Representación gráfica del proceso de detección de exoplanetas por microlente gravitacional. Elaborada por el Optical Gravitational Lensing Experiment (OGLE)
(Optical Gravitational Lensing Experiment, n.d.)

El método de *Radial Velocity* o velocidad radial aprovecha el conocido Efecto Doppler. Cuando un planeta orbita a una estrella, este genera perturbaciones en el movimiento de la estrella, causando así un "baile circular" celestial. Este movimiento puede detectarse al medir la luz que llega de la estrella. Cuando esta se aleja, la luz proveniente se desplaza al rojo, mientras que cuando se acerca se desplaza al azul. Estos cambios son uniformes por lo que se puede estimar la existencia del exoplaneta (European Southern Observatory, 2007).

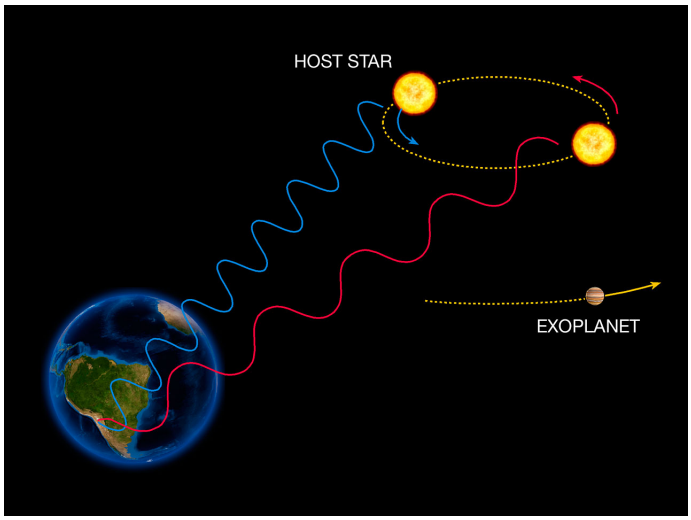


Figura 4.4: Representación de la medición de la luz de una estrella durante su baile con el planeta. Elaborada por el European Southern Observatory, 2007.
(European Southern Observatory, 2007)

Disk Kinematics o cinemática de disco protoplanetario es un método que se basa en la detección directa de planetas durante su formación en el disco de material protoplanetario. El protoplaneta limpia su órbita de polvo y escombros, dejando evidencia clara de su existencia (ALMA et al., n.d.).

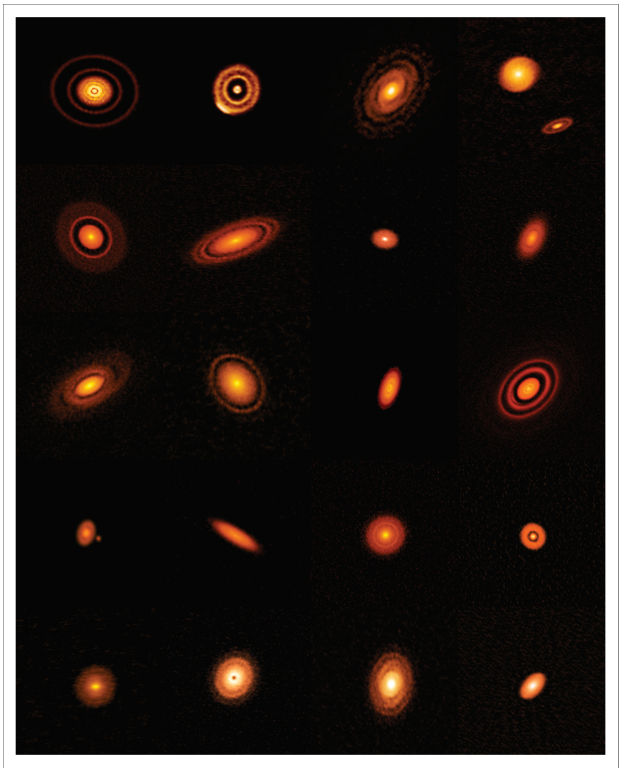


Figura 4.5: 20 discos protoplanetarios capturados por el Atacama Large Millimiter Array (ALMA). (ALMA et al., n.d.)

La base de datos *Planetary Systems Composite Data* se puede encontrar en la página del *NASA Exoplanet Archive*: un servicio en línea del NASA Exoplanet Science Institute. Esta base de datos recopila la data de todos los exoplanetas confirmados y su estrella anfitriona. La página cuenta con herramientas de manipulación de los datos (filtrado principalmente), documentación detallada del significado de cada columna y un manual de usuario. Además, una vez se tenga filtrada y personalizada la data mostrada, se puede descargar en formato CSV, TSV, VOTable o iPAC (NASA Exoplanet Science Institute, 2024).

Con ayuda del doctor Julio Gallegos, se propusieron las columnas a utilizar en el estudio. Estas son: Método de descubrimiento (*Discovery Method*), Periodo orbital en días (*Orbital Period [days]*), Radio del planeta en términos del radio de Júpiter (*Planet Radius [Jupiter Radius]*), Masa del planeta en términos de la masa de Júpiter (*Planet Mass or Mass*sin(i) [Jupiter Mass]*), Densidad del planeta (*Planet Density [g/cm^3]*), Excentricidad (*Eccentricity*), Insolación comparada a la de la Tierra (*Insolation Flux [Earth Flux]*), *Equilibrium Temperature [K]*, *V (Johnson) Magnitud* y *Ks (2MASS) Magnitud*. Se establecieron estos como los parámetros adecuados ya que son las propiedades de un planeta que están relacionadas directamente con su posible clasificación y no generarían ruido innecesario al momento de crear los modelos.

4.3. Modelos de aprendizaje

4.3.1. Árbol de Decisiones

El Árbol de Decisiones es un modelo de *machine learning* supervisado utilizado para clasificación y regresión. Este funciona a partir de lo que se conoce como *nodos de decisión* y *nodos hoja*. El proceso empieza con lo que se conoce como *nodo raíz*, luego se mueve a un nodo de decisión en el cual se evalúa la data de cada instancia para realizar la decisión adecuada y dividir los datos en grupos homogéneos o nodos hoja. Siguiendo este proceso con una cantidad de nodos de decisión y hoja determinados por el usuario del modelo, se puede llegar a un resultado donde todas las instancias de la base de datos son clasificadas en las categorías correspondientes (IBM, 2024).

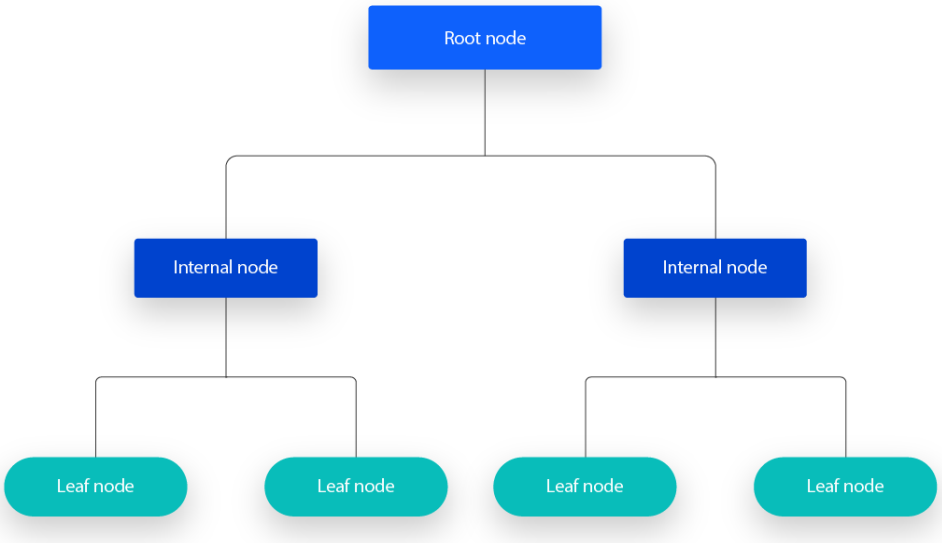


Figura 4.6: Ilustración de funcionamiento de un árbol de decisión.
(IBM, 2024)

Es necesario mencionar que este no es un modelo utilizado ampliamente por profesionales ya que es susceptible al *sobreajuste*. El sobreajuste es cuando el árbol tiene tantos nodos de decisión y hoja que, al momento de analizar un set de datos, este es capaz de crear una clasificación específica para cada instancia. Es decir, conoce casi a completitud cada instancia del set de datos por lo que solo es capaz de predecir datos del set existente y no podría realizar una predicción confiable de nuevos datos introducidos (Kadir et al., 2020). Para prevenir esto se puede modificar la cantidad de nodos que debe utilizar el modelo o la cantidad de instancias mínimos o máximas por nodo hoja, entre otros parámetros.

4.3.2. *Random Forest*

El modelo de clasificación *Random Forest* consiste en construir un arreglo de árboles de decisión y combinar los resultados de cada árbol para poder hacer una predicción más acertada. Cada árbol de decisión tiene el mismo objetivo de clasificación, pero trabaja de forma independiente a los demás. Cuando se han formado los resultados de todos los árboles, estos son promediados para poder presentar una decisión final. Promediar el resultado de todos los árboles ayuda a reducir la variabilidad en los resultados de un solo árbol de decisión y el sobreajuste (IBM, 2024).

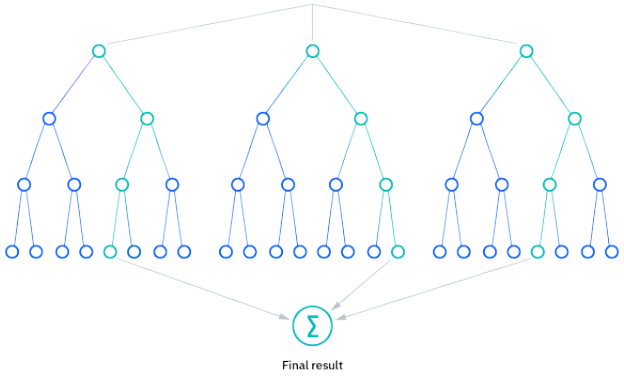


Figura 4.7: Ilustración de un bosque aleatorio.

(IBM, 2024)

4.3.3. *XGBoost*

XGBoost (Extreme Gradient Boosting) es un tipo de modelo de *machine learning* que utiliza *boosted decision trees*. Este modelo aprovecha el proceso de *boosting*. El proceso de *boosting* consiste en formar un *weak-learner*: un árbol de decisiones que tiene resultados ligeramente mejor a una toma de decisiones al azar. Luego, se crea un segundo *weak-learner* a partir de los resultados del primero con el fin de corregir los errores que hizo. Combinando secuencialmente muchos *weak-learners* se puede formar un modelo mucho más confiable y con menor probabilidad de caer en el sobreajuste (IBM, 2024).

Figura 4.8: Ilustración del proceso de *boosting*.

(IBM, 2024)

4.3.4. *K-Nearest Neighbors* (KNN)

Este modelo clasifica instancias de una base de datos según otras instancias "vecinas". Es decir, el modelo evalúa las características de una instancia y le coloca la etiqueta de clasificación de otras instancias cercanas (características similares). La cantidad de vecinos cercanos necesarios para asignar la respectiva etiqueta está definida por el número K, como indica el nombre del modelo. El valor de K puede llevar el modelo a caer al sobreajuste si es muy alto o al subajuste (lo contrario al sobreajuste) si es muy bajo (IBM, 2024).

Para determinar la cercanía de una instancia a otras, es necesario considerar a estas como puntos con coordenadas (x,y) en un plano cartesiano XY. De esta forma, se puede utilizar la ecuación de distancia euclidiana para cuantificar la distancia entre instancias (IBM, 2024):

$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \tag{4.1}$$

Figura 4.9: Ilustración del algoritmo de *K-Nearest Neighbors*. 2024. (IBM, 2024)

4.4. Métricas de evaluación

Existen varias métricas de evaluación de modelos. Es importante utilizar varias métricas ya que cada una evalúa un aspecto distinto del modelo o la relación entre otras métricas, lo que permite tener una imagen más amplia y clara del desempeño del modelo.

Precision es una medida de la cantidad de aciertos positivos que tuvo el modelo dividido por la suma de aciertos positivos y falsos positivos (Dalianis, 2018), *Recall* o *Sensitivity* es la cantidad de aciertos positivos dividido por la cantidad de aciertos positivos y falsos negativos (Dalianis, 2018).

La principal diferencia radica en su enfoque: *precision* busca minimizar los falsos positivos y *recall* busca minimizar los falsos negativos (Dalianis, 2018).

$$\text{Precision} = \frac{\text{Aciertos positivos}}{\text{Aciertos positivos+ Falsos positivos}} \tag{4.2}$$

$$\text{Recall} = \frac{\text{Aciertos positivos}}{\text{Aciertos positivos+ Falsos negativos}} \tag{4.3}$$

Si tanto los falsos positivos como los falsos negativos son importantes, existe la métrica de *F-score*. Esta es un promedio ponderado de las métricas de *precision* y *recall* y utiliza una función peso para determinar a cuál de las dos métricas se le dará prioridad (Dalianis, 2018).

$$F_\beta = (1 + \beta^2)\frac{\text{Precision} \cdot \text{Recall}}{\beta^2 \cdot (\text{Precision} + \text{Recall})} \tag{4.4}$$

Si se le quiere dar prioridad a *precision*, el valor de β debe ser menor a 1, mientras que si se le quiere dar prioridad a *recall*, el valor debe ser mayor a 1. Si ambos tienen la misma prioridad, $\beta = 1$ (Dalianis, 2018).

Una Matriz de Confusión es una herramienta que permite visualizar el desempeño de un modelo de clasificación. Las columnas de la matriz representan el valor predicho por el modelo, mientras que las filas contienen el valor real de la instancia (Murel & Kavlakoglu, 2024). En un modelo de óptimo desempeño, la matriz debería presentar un patrón principalmente diagonal, como el que se puede ver a continuación (Murel & Kavlakoglu, 2024).

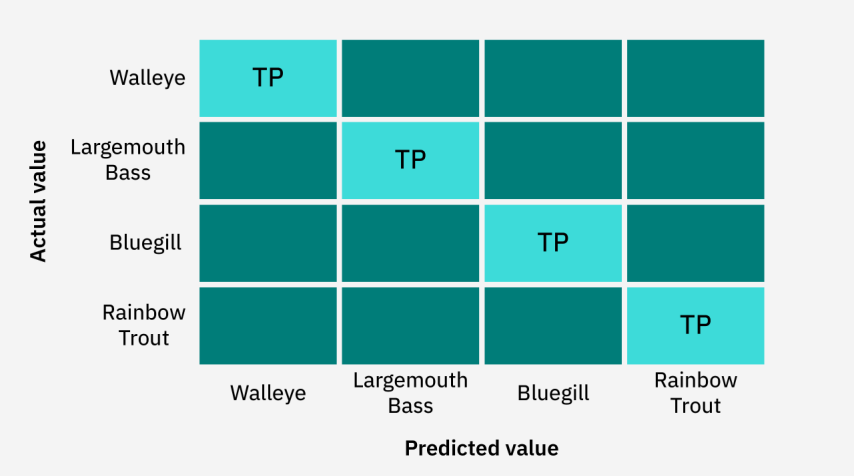


Figura 4.10: Ilustración de una matriz de confusión para un modelo de clasificación multiclase, (Murel & Kavlakoglu, 2024)

4.4.1. *Feature importance*

La métrica conocida como *feature importance* es utilizada para describir qué variables utilizadas fueron las más relevantes para el modelo al momento de tomar decisiones y llegar a los resultados finales de las clasificaciones (Codecademy, 2025). Cada modelo puede tener una forma distinta de calcular su *feature importance*.

Feature importance de Árbol de Decisiones y *Random Forest*

Los árboles de decisión definen la *feature importance* a partir de la reducción de impureza en los nodos. En este caso, la impureza se refiere a la entropía, que representa el grado de desorden en un nodo, es decir, qué tan mezcladas están las instancias de las distintas clases (SciKit Learn, 2024).

De esta forma, las variables con mayor importancia son aquellas que logran dividir los datos en nodos más "puros", es decir, nodos donde predominan instancias de una sola clase. Cuanto mayor sea la reducción de impureza que produce una variable al generar divisiones en el árbol, mayor será su importancia relativa en el modelo (SciKit Learn, 2024).

La entropía se calcula de la siguiente forma (SciKit Learn, 2024):

$$S = -\sum_{c \in C} p(c) \log_2 p(c) \tag{4.5}$$

Este valor de entropía es utilizado para calcular el *information gain*, el cual es la métrica que muestra cuánto varía la entropía del sistema de datos al dividir usando una *feature* específica (SciKit Learn, 2024).

$$IG(S) = S - \sum_{v \in V} \frac{|D_v|}{|D|} S_v \tag{4.6}$$

Donde S es la entropía antes de dividir los datos, v es el posible valor (categoría o rango) de la variable sobre la que se está dividiendo, V es el conjunto de posibles valores de v, D_v es la cantidad de elementos del conjunto de datos original que cumplen con el valor v, y D es la cantidad total de elementos del conjunto de datos. Esto es para un solo nodo; el *feature importance* final es un acumulado del IG de todos los nodos (SciKit Learn, 2024).

Para el caso del *Random Forest*, ya que este es un acumulado de árboles de decisiones, la *feature importance* se obtiene a través de un promedio de los resultados de dicho cálculo de cada clasificador utilizado.

Feature Importance de *XGBoost*

El modelo de *XGBoost* utiliza la métrica de *gain* como *feature importance*. Esta métrica evalúa qué tanto mejora la función de pérdida empleada (en este caso *multi log-loss*) al momento de dividir los datos usando uno de los parámetros. Este valor se calcula de la siguiente forma (Yang, 2020):

$$\text{Gain} = \frac{1}{2} \sum_{k=1}^{K} \left[\frac{G_{L,k}^2}{H_{L,k} + \lambda} + \frac{G_{R,k}^2}{H_{R,k} + \lambda} - \frac{(G_{L,k} + G_{R,k})^2}{H_{L,k} + H_{R,k} + \lambda} \right] - \gamma \tag{4.7}$$

Donde $G_{L,k}$ es la suma de los gradientes de la clase k en el nodo izquierdo, $H_{L,k}$ es la suma de los hessianos de la clase k en el nodo izquierdo, $G_{R,k}$ es la suma de los gradientes de la clase k en el nodo derecho, $H_{R,k}$ es la suma de los hessianos de la clase k en el nodo derecho, K es el número total de clases, λ es el parámetro de regularización, y γ es el parámetro de penalización por cada división añadida al árbol. El gradiente y el hessiano se definen de la siguiente forma (Yang, 2020):

$$G_{i,k} = \frac{\partial L}{\partial \hat{y}_{i,k}} \tag{4.8}$$

$$H_{i,k} = \frac{\partial^2 L}{\partial \hat{y}_{i,k}^2} \tag{4.9}$$

Donde $G_{i,k}$ es el gradiente de la función de pérdida (L representa la función de pérdida) respecto a la predicción $\hat{y}_{i,k}$ de la muestra i (nodo izquierdo o derecho) en la clase k; $H_{i,k}$ es el hessiano, es decir, la segunda derivada de la función de pérdida respecto a la misma predicción. k recorre el número total de clases K. El gradiente indica la dirección y magnitud en la que la predicción debe ajustarse para reducir el error, mientras que el hessiano mide la curvatura de la función de pérdida.

CAPÍTULO 5

Antecedentes

La astronomía y el aprendizaje automático son campos que han estado en contacto previamente, específicamente en el campo de exoplanetas. Los científicos han utilizado *machine learning* para tareas como la detección y, similar a lo que se busca en el presente trabajo, para la clasificación de exoplanetas, llegando a interesantes resultados que servirán de guía en el presente y futuros trabajos:

1. En Portugal, Barboza y Ulmer-Moll (Barboza & Ulmer-Moll, 2020) lograron utilizar el método de *clustering "K-Means"* para clasificar los planetas de su base de datos en 3 clasificaciones principales con base en la masa planetaria y el periodo orbital: Júpiters calientes, gigantes de gas con periodo largo y planetas pequeños. Además, utilizaron un método más complejo conocido como *Uniform Manifold Approximation and Projection (UMAP)*, con lo cual llegaron a una clasificación más diversa de 4 tipos: Júpiters calientes, gigantes de gas con periodo largo, sub-neptunianos y planetas rocosos.

2. Científicos de la Universidad Metodista del Sur (Clayton, Manry, & Rafiqi, 2019), en Estados Unidos, utilizaron la base de datos KOI (*Kepler Objects of Interest*) para entrenar los modelos de *Support Vector Machine/Support Vector Classifier*, *K-Nearest Neighbor* y *Random Forest* con el fin de identificar la probabilidad de una instancia de ser un exoplaneta. Encontraron que el clasificador de *Random Forest* tuvo el mejor desempeño con un *accuracy* de 98 %.

3. En el trabajo de graduación presentado por González (Gonzáles, 2021) se utilizan los modelos de Árbol de Decisiones, *K-Nearest Neighbor* y *Support Vector Machine*, además de explorar la posibilidad de utilizar redes neuronales. El resultado principal de este trabajo recae en la importancia de seleccionar adecuadamente los parámetros que serán utilizados en los modelos, demostrando que dependiendo de esta, se obtienen mejores o peores resultados los cuales varían drásticamente.

Las variables que se usaron para entrenar el modelo fueron las siguientes:

Nombre en base de datos	Nombre en tabla	Descripción
pl_name	Planet name	Nombre predeterminado para el planeta asignado por Exoplanet Archive.
discoverymethod	Discovery Method	Método utilizado para identificar el exoplaneta.
pl_orbper	Orbital Period [days]	Tiempo que toma el planeta en completar una orbita entera alrededor de su estrella o sistema anfitrión .
pl_radj	Candidate Raius [Jupiter radii]	Segmento de línea que empieza en el centro del exoplaneta hasta su superficie. Medido en unidades de radio de Júpiter
pl_bmassj	Planet Mass or Mass*sin(i) [Jupiter Mass]	Mejor estimación disponbible de la masa del planeta.
pl_orbeccen	Eccentricity	Medida que cuantifíca qué tanto la orbita se asemeja a un círculo.
st_spectype	Spectral Type	Clasificación de la estrella según sus caracerísticas siguiendo el sistema de Morgan-Keenan.
st_rad	Setllar Radius [Solar Radii]	Segmento de línea que empieza en el centro de la estrella hasta su superficie. Medido en unidades de radio del Sol.
st_mass	Stellar Mass [Solar mass]	Masa de la estrella anfitriona medida en unidades de masa del Sol.
st_met	Stellar Metallicity [dex]	Comparación de la cantidad de metal en la fotósfera de la estrella en comparación con la cantidad de hidrógeno.

Figura 5.1: Descripción y definición de las variables utilizadas para entrenar los modelos.

Las razones por las que se escogieron estas variables son las siguientes:

- Nombre del planeta: para ser utilizado como un identificador único el cual permitiría unir las bases de datos de clasificación del planeta y sus parámetros físicos.
- Método de descubrimiento: para comprobar si el método de descubrimiento puede ser una pista ante el tipo de planeta descubierto. Por ejemplo, como menciona la *NASA*, el método de velocidad radial facilitó el descubrimiento de planetas conocidos como *"Hot Jupiters"*, que consisten en planetas con características similares a las de Júpiter, pero orbitan a la estrella en una órbita más cercana, aumentando así su temperatura (NASA, 2024).
- Periodo orbital: porque, como se menciona anteriormente con los *"Hot Jupiters"*, estos tienen un periodo y órbita característicos, por lo que en combinación con otras variables, podría ser útil para determinar el tipo de exoplaneta o incluso, abrir las posibilidades a nuevas categorías.

- Masa y radio planetario: como se describió anteriormente en la sección de *Marco Teórico*, las clasificaciones propuestas están relacionadas directamente con estas dos variables.
- Excentricidad: similar a la consideración del periodo, la excentricidad podría revelar información importante sobre un planeta y nuevas posibles clasificaciones.
- Tipo espectral, radio estelar, masa estelar y metalicidad: para determinar si las características principales de una estrella y su posible historia de formación pueden estar relacionadas al tipo de planetas que se pueden formar a su alrededor, como es mencionado por Johnson (Johnson et al., 2024) en su estudio que relaciona la masa estelar y la metalicidad a la existencia de gigantes gaseosos en sus sistemas solares.

CAPÍTULO 6

Metodología

6.1. Diseño experimental

6.1.1. Material y equipo

La computadora utilizada fue una laptop *Lenovo Legion Pro 5* con las especificaciones:

- Procesador: Core i9-13900HX
- 24 núcleos
- RAM: 32.0 GB
- Sistema operativo: *Windows* 11
- Versión de Python utilizada: 3.11.5

Los datos utilizados para el entrenamiento provienen de las siguientes bases de datos.

- Para los parámetros planetarios de cada exoplaneta se utilizaron los datos de *NASA Exoplanet Archive* del *NASA Exoplanet Science Institute*, específicamente de la sección del *Planetary System Composite Data*: `https://exoplanetarchive.ipac.caltech.edu/cgi-bin/TblView/nph-tblView?app=ExoTbls&config=PSCompPars`
- Para tener una clasificación de los planetas, se extrajo la información del catalogo de exoplaneta de la NASA : `https://science.nasa.gov/exoplanets/exoplanet-catalog/`

6.1.2. Librerías

El procedimiento se realizó utilizando Python 3.11.5, Java 11.0.26, PySpark 3.5.5 y las librerías utilizadas fueron las siguientes:

- pyspark (SQL functions)
- pandas
- numpy
- xgboost
- sklearn
- matplotlib
- seaborn

6.2. Procedimiento experimental

La primera etapa del procedimiento experimental consiste en la obtención de la base de datos para poder entrenar los modelos. El proceso de tener una base de datos presentó una complicación, la base de datos *Planetary Composite Data* del *NASA Exoplanet Archive* no cuenta con una clasificación oficial de los planetas presentes en dicha base, sin embargo, existe el *NAS Exoplanet Catalog*. Este es el catálogo oficial de la NASA que cumple como una enciclopedia de exoplanetas, contando con información como radio del planeta, su clasificación (propia de la NASA), método de descubrimiento, masa planetaria, entre otros. No obstante, la información no es presentada en un formato descargable como para poder entrenar un modelo de *machine learning*. Por lo tanto, se requiere de un procedimiento para poder extraer la información de la clasificación de un exoplaneta. Dicho procedimiento es el análisis y extracción de información de una página web, siendo este conocido como *webscrapping*.

El proceso realizado consiste en los siguientes pasos:

1. Descargar la base de datos de NASA Exoplanet Archive como csv, seleccionando las columnas relevantes.

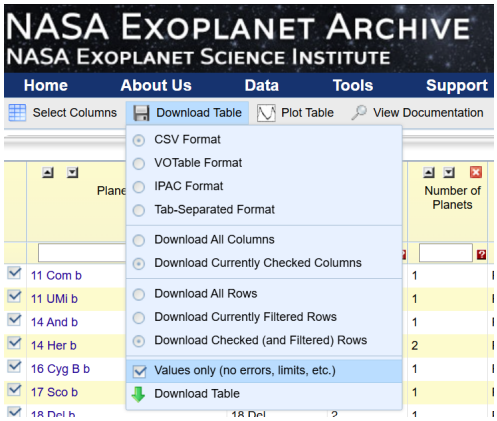


Figura 6.1: Captura de pantalla de las configuraciones de descarga utilizadas.

2. Evaluar la estructura de la página del NASA Exoplanet Catalog. A la fecha de realización de este trabajo, el proceso consiste en: presionar clic derecho e inspeccionar elemento, luego, dirigirse a la ventana de *Network* y aplicar uno de los filtros de tipo de planeta. Se podrá ver que se genera un *request* en la página. En este, se debe viajar a la ventana de *payload* donde se podrá ver en la variable de *"meta_fields"* el efecto del filtro seleccionado.

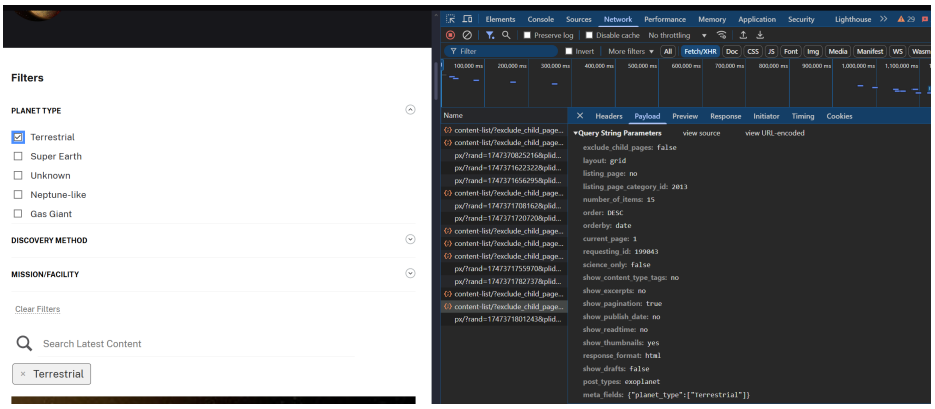


Figura 6.2: Captura de pantalla del *request* luego de seleccionar el filtro de *Terrestrial*.

3. Diseñar y escribir el código de *webscrapping* con base en la modificaciones de la página al aplicar el filtro para extraer la información de la clasificación de los exoplanetas. El resultado de este proceso serán 4 archivos distintos, cada uno con la información de nombre y clasificación de los exoplanetas.

4. Unificar los 4 archivos para tener un solo documento con todos los planetas disponibles y su respectiva clasificación.

5. Unir la base de datos de nombre y clasificación de cada planeta con la base que contiene los parámetros planetarios específicos. La unión y manipulación de estas bases de datos fue realizada utilizando PySpark y Pandas.

Unir los dos archivos finales (nombre y clasificación con parámetros planetarios) presenta un problema y este es que el formato en el que están escritos los nombres de los planetas en el catálogo no coincide completamente con los nombre en el NASA Exoplanet Archive. Por ejemplo, en el catálogo, un planeta puede tener de nombre 16 Cygni B b, mientras que en el archivo de parámetros planetarios el mismo planeta tiene el nombre 16 Cyg B b. Esta discrepancia dificulta el poder unir adecuadamente los valores de cada planeta con su respectiva clasificación, por lo que al momento de acoplar las bases de datos se escogió tomar en cuenta solo los planetas que tienen nombres idénticos en ambas fuentes, lo que redujo el tamaño total de datos disponibles para entrenar el modelo.

Para mejorar este proceso, se recomienda utilizar algoritmos de *approximate string matching* que permiten evaluar y comparar cadenas de caracteres para medir la similitud entre dos palabras. Por ejemplo, analizar la palabra "dinroz determinar que la palabra más cercana es "dinero".

En cuanto al procedimiento para entrenar los modelos, los pasos fueron los siguientes:

1. Leer el archivo csv de la base de datos y convertir a un *Pandas dataframe* para poder utilizar los modelos de la librería de *sklearn*.

2. Separar los datos en dos partes: 80 % para entrenamiento y 20 % para pruebas.

3. Preparar un proceso de *Grid Search* que permite explorar varios múltiples combinaciones de hiperparámetro para encontrar el conjunto que permite llegar a los mejores resultados posibles a través de validación cruzada.

4. Entrenar el modelo con los hiperparámetros sugeridos por la búsqueda de *Grid Search*.

5. Realizar las predicciones del modelo utilizando el 20 % de datos de prueba.

6. Validar el desempeño del modelo utilizando las métricas establecidas.

Los hiperparámetros modificados para cada modelo durante el proceso de *Grid Search* son:

- Modelo *XGBoost*:
  - max_depth: Profundidad máxima que puede alcanzar cada estimador (árbol).
  - n_estimators: Número de estimadores (árboles que puede utilizar el modelo).
  - learning_rate: Velocidad con la que puede aprender el modelo.
  - subsample: Porcentaje de datos que puede utilizar el modelo para cada estimador.
- *Decision Tree*:
  - criterion: Método de partición de datos.
  - max_depth: Profundidad máxima que puede alcanzar el árbol.
  - min_samples_split: Muestra mínima de datos para separar un nodo.
  - min_samples_leaf: Muestra mínima de datos por cada hoja del árbol.
- *Random Forest*:
  - criterion: Método de partición de datos.
  - max_depth: Profundidad máxima que puede alcanzar cada árbol.
  - min_samples_split: Muestra mínima de datos para separar un nodo.
  - min_samples_leaf: Muestra mínima de datos por cada hoja del árbol.
- *K-Nearest Neighbors*:
  - n_neighbors: Número de vecinos cercanos
  - weights: Forma de evaluar el peso de cada instancia sobre sus vecinos.
  - metric: Forma de medir la distancia entre vecinos.

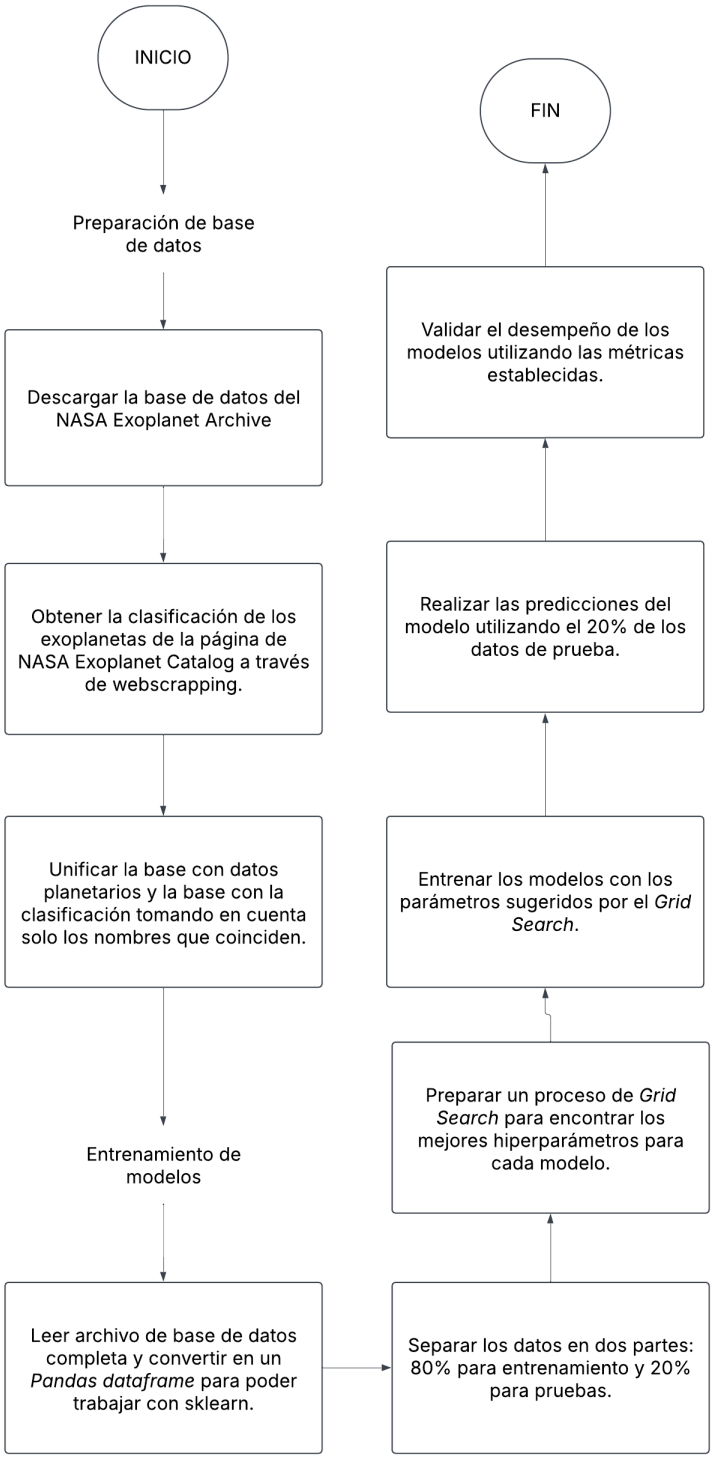


Figura 6.3: Diagrama de flujo del proceso realizado.

CAPÍTULO 7

Resultados

7.1. Árbol de Decisiones

Las siguientes figuras y cuadros muestran los parámetros utilizados en el proceso de *Grid Search* y los resultados de utilizar los parámetros sugeridos por este mismo en el modelo de Árbol de Decisiones.

Hiperparámetro	**Valores a probar**	**Valor seleccionado**
`criterion`	entropy	entropy
`max_depth`	4, 6, 8, 10, 12	10
`min_samples_split`	2, 5, 10, 12	12
`min_samples_leaf`	1, 2, 4, 6	2

Cuadro 7.1: Resultados de GridSearchCV para el modelo de Árbol de Decisiones

Tipo	**Precisión**	**Recall**	**F1-score**	**Soporte**
Gas_giant	1.0	1.00	1.00	268
Neptune-Like	0.96	0.98	0.97	313
Super_Earth	0.98	0.96	0.97	285
Terrestrial	1.0	1.00	1.00	34

Cuadro 7.2: Reporte de clasificación del Árbol de Decisión

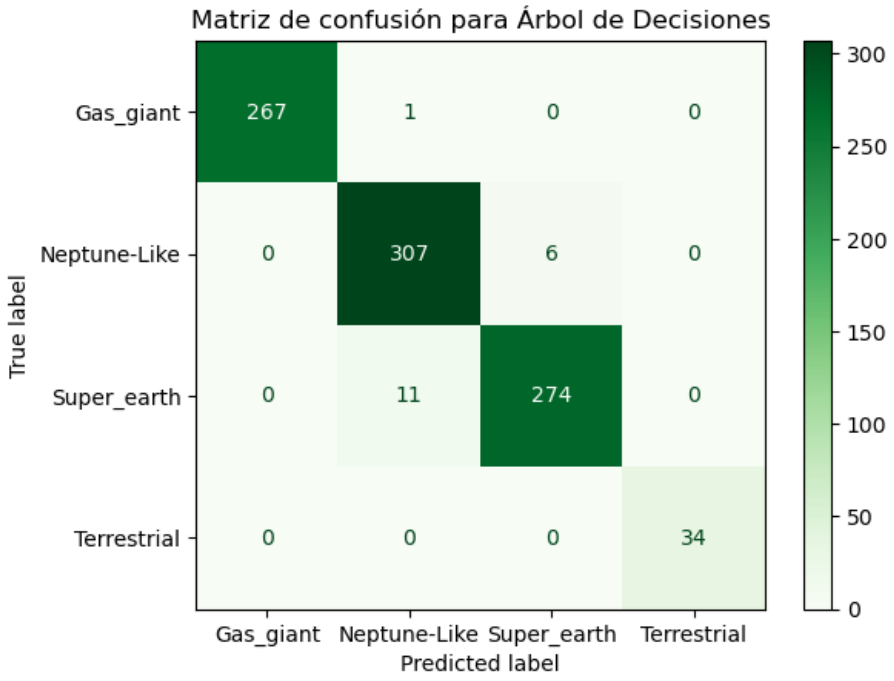


Figura 7.1: Matriz de confusión del modelo de Árbol de Decisiones

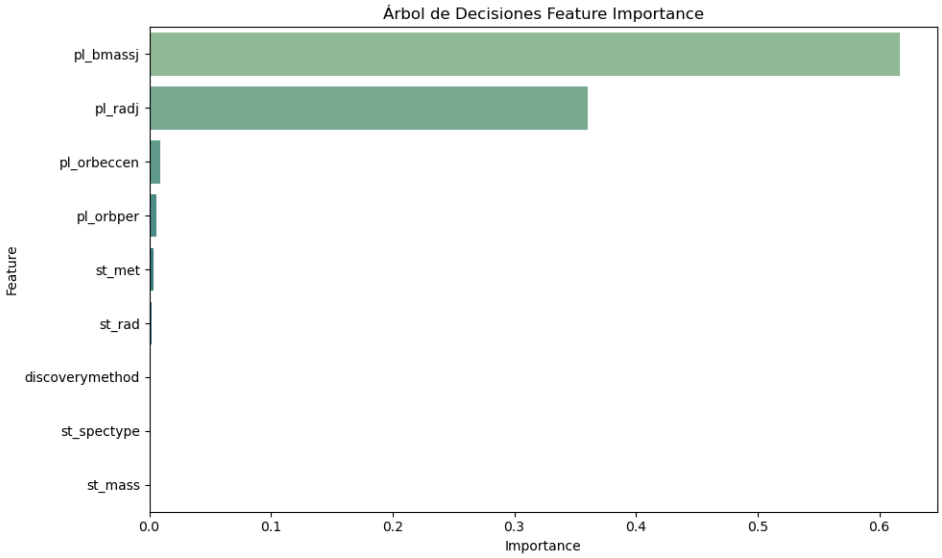


Figura 7.2: Feature Importance del modelo de Árbol de Decisiones

El diagrama del Árbol de Decisioness que se obtuvo se encuentra en el repositorio de *GitHub* en la sección de *Anexos*.

7.2. *Random Forest*

Esta sección contiene los resultados obtenidos al entrenar el modelo de *Random Forest* con los parámetros sugeridos por el proceso de *Grid Search* para maximizar el $F - score$.

Hiperparámetro	**Valores a probar**	**Valor seleccionado**
`criterion`	entropy	entropy
`n_estimators`	10, 15, 20, 100	20
`max_depth`	4, 6, 8, 10, 12	12
`min_samples_split`	2, 5, 10, 12	12
`min_samples_leaf`	1, 2, 4, 6	1

Cuadro 7.3: Resultados de GridSearchCV para el modelo *Random Forest*

Tipo	**Precisión**	**Recall**	**F1-score**	**Soporte**
Gas_giant	1.0	1.00	1.00	268
Neptune-Like	0.96	0.98	0.97	313
Super_Earth	0.98	0.95	0.96	285
Terrestrial	0.97	1.00	0.99	34

Cuadro 7.4: Reporte de clasificación del *Random Forest*

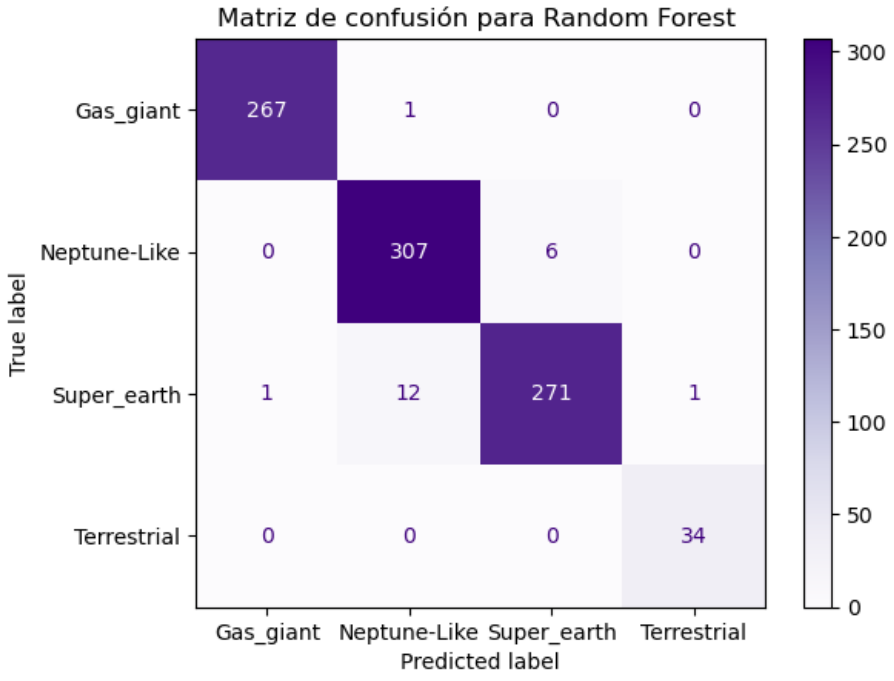


Figura 7.3: Matriz de confusión del modelo de *Random Forest*

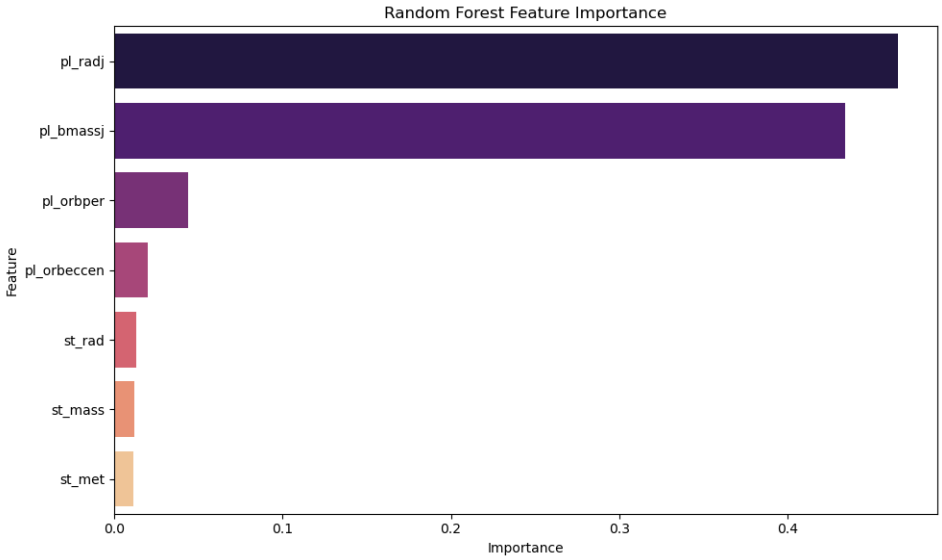


Figura 7.4: Feature Importance del modelo de *Random Forest*

7.3. *XGBoost*

A continuación se presentan los resultados de la optimización del modelo *XGBoost* al terminar el proceso de *Grid Search*, así como las variables más importantes para el modelo.

Hiperparámetro	**Valores a probar**	**Valor seleccionado**
`n_estimators`	10, 50, 70, 100	100
`learning_rate`	0.01, 0.05, 0.1	0.1
`max_depth`	4, 6, 8, 10, 12	8
`subsample`	0.2, 0.6, 0.8, 1.0	1.0

Cuadro 7.5: Resultados de GridSearchCV para el modelo *XGBoost*

Tipo	**Precisión**	**Recall**	**F1-score**	**Soporte**
Gas_giant	1.00	0.98	0.99	268
Neptune-Like	0.96	0.98	0.97	313
Super_Earth	0.98	0.98	0.98	285
Terrestrial	0.97	0.97	0.97	34

Cuadro 7.6: Reporte de clasificación del modelo *XGBoost*

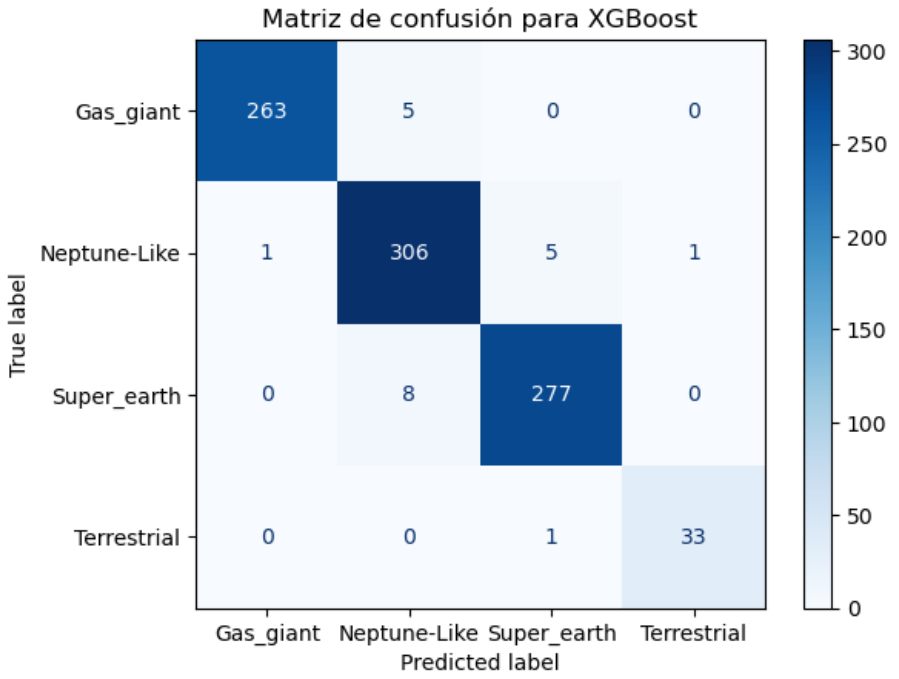


Figura 7.5: Matriz de confusión del modelo *XGBoost*

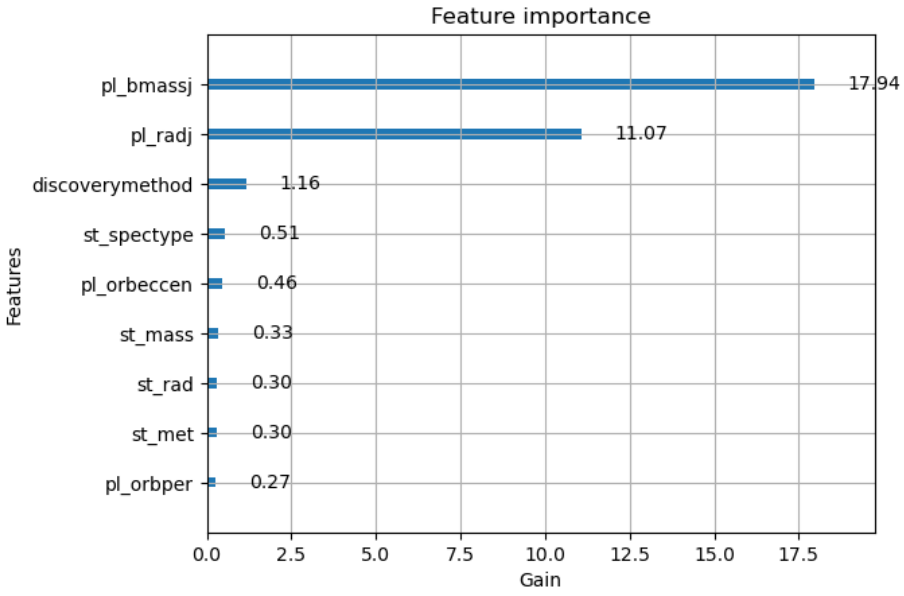


Figura 7.6: Feature Importance del modelo *XGBoost*

7.4. *K-Nearest Neighbors*

Los siguientes resultados corresponden al modelo de *K-Nearest Neighbors* luego de ser entrenado y comprobado utilizando los parámetros propuestos por el proceso de *Grid Search*.

Hiperparámetro	Valores a probar	Valor seleccionado
`n_neighbors`	Valores impares entre 3 y 67	3
`wights`	uniform, distance	distance
`metric`	euclidean, manhattan	manhattan

Cuadro 7.7: Resultados de GridSearchCV para el modelo *K-Nearest Neighbors*

Tipo	Precisión	Recall	F1-score	Soporte
Gas_giant	0.99	0.94	0.97	268
Neptune-Like	0.85	0.85	0.85	313
Super_Earth	0.80	0.87	0.83	258
Terrestrial	0.82	0.53	0.64	34

Cuadro 7.8: Reporte de clasificación del modelo *K-Nearest Neighbors*

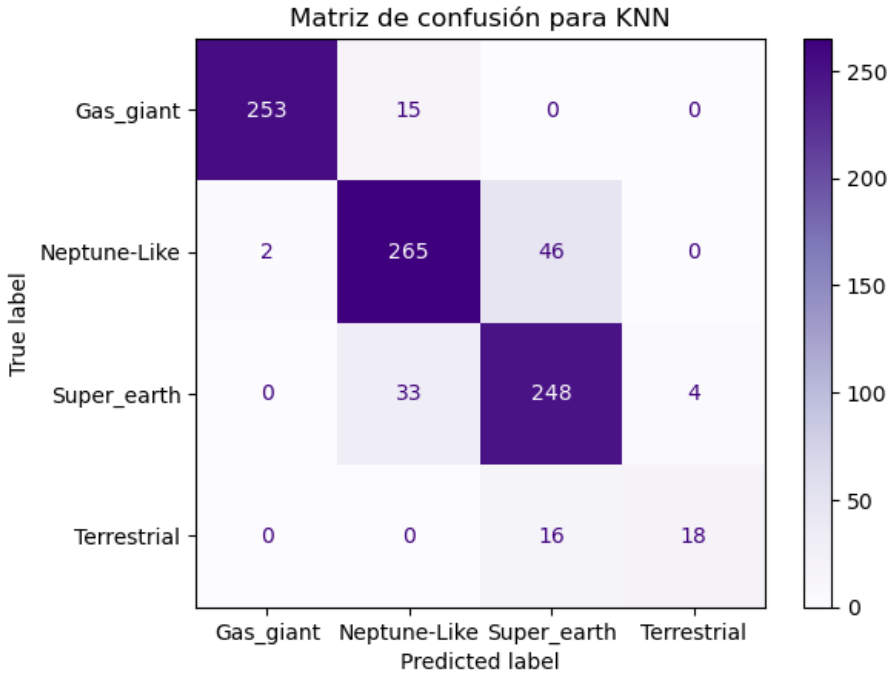


Figura 7.7: Matriz de confusión del modelo *K-Nearest Neighbors*

7.5. Validación cruzada

El presente cuadro presenta la comparación entre los resultados de realizar un proceso de validación cruzada con 10 iteraciones con el fin de validar los parámetros propuestos por el proceso de *Grid Search* para cada modelo. Se presenta la media y la desviación estándar para dar una idea del comportamiento de las iteraciones y qué tan precisos son los resultados.

Modelo	Promedio de F1 score (macro)	Desviación estándar
`Árbol de Decisiones`	0.9819	0.0053
`Random Forest`	0.9783	0.0094
`XGBoost`	0.9822	0.0087
`K-Nearest Neighbors`	0.8100	0.0395

Cuadro 7.9: Resultados de validación cruzada con 10 iteraciones.

CAPÍTULO 8

Discusión y análisis de resultados

El proceso de preparación de datos, entrenamiento, ajuste y evaluación de modelos de *machine learning* requiere de especial cuidado y atención cuando se trata con datos sensibles como lo pueden ser los parámetros planetarios obtenidos por telescopios. A continuación, se presentan los factores que influyen en dicho proceso y se analizan los resultados obtenidos.

8.1. Consideraciones por cada modelo

Los telescopios y *software* utilizados para obtener los datos planetarios no son perfectos, por lo que es normal que no se encuentren absolutamente todos los datos para todos los parámetros. Esto es algo de suma relevancia al momento de entrenar modelos de *machine learning* ya que no todos los modelos interactúan igual con valores faltantes. En el caso de *XGBoost*, este no tiene inconvenientes, sin embargo, los modelos de Árbol de Decisiones, *Random Forest* y *K-Nearest Neighbors* pueden llegar a encontrar dificultades. Para asegurar que todos los modelos trabajan con la misma base de datos, se decidió eliminar todas las instancias que tuviesen un valor nulo ya que procedimientos de manejo de valores nulos como llenar con la media o con ceros pueden no ser apropiados debido a la sensibilidad de los datos y el fenómeno que representan.

8.2. Desempeño de Árbol de Decisiones

Utilizando los parámetros sugeridos por el proceso de *Grid Search* (Tabla 7.1), se puede observar que el modelo tuvo excelentes resultados de $F_1 - score$ al momento de clasificar planetas de tipo *Gas Giant* (Gigantes Gaseosos), al igual que con planetas de tipo *Terrestrial* (Terrestre). Esto se puede visualizar mejor en la matriz de confusión de este modelo (Figura 7.1). No se presentó ningún falso positivo o negativo en estas dos clasificaciones.

Por otro lado, las clasificaciones de *Neptune-Like* (Neptuniano) y *Super Earth* (Súper Tierra) tuvieron resultados iguales de $F_1 - score$ pero no de *precision* y *recall*. La clasificación de *Neptune-Like* tuvo un mejor desempeño en *recall* que en *precision*, lo que indica que el modelo es bueno para clasificar adecuadamente a los planetas de este tipo, pero también puede colocar planetas que no corresponden en esta categoría. Lo contrario pasa en la clasificación de *Super Earth*, tiene menos

falsos positivos pero más falsos negativos, es decir, es menos probable que clasifique como Súper Tierra a un planeta que no lo es, pero es más probable que no clasifique como Súper Tierra a un planeta que sí lo es.

La diferencia de desempeño entre estas clases podría darse por la naturaleza de los planetas. Se puede observar que los planetas que se encuentran en los extremos de características son los que mejor se clasificaron, por otro lado, las dos clasificaciones intermedias, las cuales presentan límites borrosos entre sí, presentaron cierta incertidumbre. Esto se pude apoyar al observar la importancia de las variables que utilizó el modelo (Figura 7.2). La masa planetaria y el radio, en ese orden, fueron las variables más importantes para el modelo. Nótese que estas variables están altamente relacionadas con la definición de cada clasificación, mientras que las variables relacionadas a la naturaleza de la estrella anfitriona fueron de importancia casi negligente.

8.3. Desempeño de *Random Forest*

Utilizando los parámetros descritos en la Tabla 7.3, se observa que el modelo tuvo un comportamiento similar al del Árbol de Decisiones. Tiene un buen desempeño en las clasificaciones de *Gas Giant* y *Terrestrial*, mientras que en las clasificaciones de *Neptune-Like* y *Super Earth* tiene un comportamiento contrario en *precision* y *recall*. Sin embargo, el desempeño en estas últimas dos clasificaciones fue peor, resultando en un $F_1 - score$ menor. En la matriz de confusión puede observarse que el modelo tuvo más problemas en la clasificación de *Super Earth*, confundiendo esta clasificación con todas las demás, especialmente con la clasificación de *Neptune-Like*.

Similar al caso del Árbol de Decisiones, puede intentarse explicar esta diferencia utilizando las variables más importantes. El modelo de *Random Forest* también considera como variables más importantes a la masa planetaria y al radio planetario, sin embargo, le da prioridad al radio. Analizando las definiciones de cada planeta, donde por tamaño la clasificación de *Neptune-Like* y *Super Earth* son adyacentes, es entendible que el modelo pueda llegar a confundir los planetas Neptunianos con una Súper Tierra si le da prioridad al radio como variable más importante; mientras que la diferencias de masas entre una Súper Tierra y un Planeta Neptuniano es mucho más marcada. Esto apoya el razonamiento descrito previamente acerca de los límites de las características que definen a cada clasificación.

8.4. Desempeño de *XGBoost*

Al utilizar los parámetros mostrados en la Tabla 7.5, se obtuvieron resultados similares a los anteriores, sin embargo, no completamente iguales. La clasificación de *Gas Giant* sigue siendo la que presenta mejor desempeño, no obstante, la clasificación de *Terrestrial* no fue igual de eficiente que en los otros modelos, siendo superada por la clasificación de *Super Earth* en cuanto a $F_1 - score$.

Los resultados indican que modelo es eficiente para clasificar adecuadamente los planetas de tipo Neptuniano pero también es el que tiene una mayor cantidad de falsos positivos. Esto se puede ver en la matriz de confusión del modelo (Figura 7.6), donde nuevamente existe una mayor incertidumbre entre las clasificaciones de *Neptune-Like* y *Terrestrial*, aunque es menor a la de los dos modelos anteriores; nótese que este modelo también confundió a los Neptunianos con la clasificación de Gigantes Gaseosos.

En cuanto a las variables más importantes del modelo, estas se pueden observar en la Figura 7.7. Nótese que este modelo también considera a la masa planetaria y al radio planetario como los parámetros más relevantes, y en el mismo orden que el Árbol de Decisiones.

8.5. Desempeño de *K-Nearest Neighbors*

Con los parámetros descritos en la Tabla 7.7 se obtuvieron los resultados de la Tabla 7.8. Al comparar con los modelos mencionados anteriormente, se puede notar que el modelo de *K-Nearest Neighbors* es el de peor desempeño.

La clasificación de *Gas Giant* sigue siendo la de mejores resultados con un $F_1 - score$ de 0.97, indicando que los Gigantes Gaseosos tienen características bien definidas ya que se encuentran a distancias que permite que el modelo los clasifique como vecinos. No obstante, los demás tipos tuvieron un $F_1 - score$ menor al 0.90, llegando hasta 0.64 para *Terrestrial*. Esto es una diferencia abismal al comparar con los modelos previos, con especial énfasis en la clasificación de planetas tipo *Terrestrial* donde el *recall* llega hasta el 0.53, lo que significa que tiene casi la misma cantidad de falsos negativos que de verdaderos positivos. Dicho fenómeno se puede ver mejor en la matriz de confusión (Figura 7.8).

Este es un comportamiento esperado para este modelo debido a la naturaleza de la base de datos. En las Tablas de resultados puede observarse en la columna de soporte que el tipo de exoplaneta *Terrestrial* tiene la menor cantidad de instancias, lo que significa que las categorías no tienen una cantidad uniforme de elementos, es decir, no están balanceadas. Como se menciona en Cover y Hart (1967), el modelo de *K-Nearest Neighbors* tiene problemas al momento de trabajar con este tipo de bases de datos, dando prioridad a las clases con mayor cantidad de instancias y dando menor importancia a las clases con menor cantidad, lo que se ve claramente reflejado en los resultados. Además, en la matriz de confusión (Figura 7.8) se puede ver que este modelo también tiene dificultades al diferenciar la entre *Neptune-Like* y *Terrestrial*.

8.6. Comparación global

En la Tabla 7.9 se puede observar el promedio del $F_1 - score$ y la desviación estándar luego de realizar una validación cruzada de 10 iteraciones con el fin de validar el desempeño de cada modelo. Puede notarse que *XGBoost* y el Árbol de decisiones tienen el mayor promedio de $F_1 - score$, siendo seguidos por *Random Forest* y *K-Nearest Neighbors*. Además, el modelo de Árbol de Decisiones es el que presenta menor variabilidad, mientras que *K-Nearest Neighbors* presentó considerablemente mayor variación.

Como era esperado por la literatura, el modelo de *XGBoost* fue el que tuvo el mejor desempeño de los 4 modelos, mientras que el modelo *K-Nearest Neighbors* fue el peor. Sin embargo, existe una posibilidad de mejora en los resultados si se aumenta la cantidad de valores por parámetro durante el proceso de *Grid Search*, teniendo en cuenta la capacidad computacional que esto requeriría. Esto también podría llegar a formar un modelo de *Random Forest* que tenga un mejor rendimiento que el Árbol de Decisiones (caso contrario a lo presentado en este trabajo), como lo sugiere la literatura.

En cuanto a los resultados de *feature importance*, se encuentra que los primeros tres modelos coinciden en que la masa planetaria y el radio son los parámetros más importantes para lograr los resultados de los modelos, no obstante, los modelos de *XGBoost* y Árbol de Decisiones, que son los que tuvieron mejor desempeño, coinciden en que la masa es el parámetro principal al momento de separar los planetas.

La conexión entre masa y radio de exoplanetas es un fenómeno que ha sido estudiado previamente para la caracterización de exoplanetas como es el caso del estudio por Seager et al. (2007), donde se encuentra una fórmula matemática que relaciona el radio y la masa para planetas sólidos; el estudio de Müller (2011) busca establecer rangos de masa y radio para la clasificación de exoplanetas, así como definir proporciones entre estas dos características. Por lo tanto, el presente trabajo puede considerarse como un aporte a la consolidación de estos dos parámetros como la clave para poder

agrupar y clasificar exoplanetas.

CAPÍTULO 9

Conclusiones

Tras finalizar el análisis del proceso experimental, la presentación de resultados, la comparación tanto individual como global de los modelos de aprendizaje automático, y su correspondiente respaldo teórico en la literatura, se llegó a las siguientes conclusiones:

1. Se encontró que los modelos basados en árboles (Árbol de Decisiones, Random Forest y XGBoost) demostraron tener un excelente desempeño al momento de clasificar exoplanetas en las 4 categorías propuestas por la NASA: Terrestres, Súper Tierras, Neptunianos y Gigantes Gaseosos. Por otro lado, el modelo de K-Nearest Neighbors presenta dificultad al realizar esta clasificación debido al desbalance en la cantidad de planetas por categoría en la base de datos.

2. Se encontró que todos los modelos presentaron ambigüedad al momento de clasificar los planetas bajo las categorías de Súper Tierra y Neptunianos, dicha confusión pudiendo ser atribuida a la naturaleza difusa de la separación entre las características definidas para estas categorías.

3. Se encontró que los modelos basados en árboles encontraron a la masa planetaria y al radio como los parámetros más importantes al momento de clasificar a los planetas, además, los modelos de mejor desempeño coinciden en la masa como el parámetro principal. Adicionalmente, dicho resultado es respaldado por estudios previos que determinan que la relación masa-radio es clave al buscar caracterizar o agrupar exoplanetas.

4. El estudio presenta los recursos y la información necesarios para construir una base de datos adecuada, así como una descripción clara de la configuración y evaluación de modelos que permita llevar a cabo un proceso de replicación de resultados.

CAPÍTULO 10

Recomendaciones

El proceso de experimentación y el análisis y discusión de resultados permitió construir una serie de recomendaciones que podrían seguirse al realizar un estudio similar al presente:

1. Para poder contar con una base de datos con todos los planetas posibles, se recomienda explorar la idea del uso de un algoritmo de *string matching* el cual permitiría mejorar la eficiencia del proceso de unificación de la base de datos con los parámetros planetarios y la base de datos con la clasificación de los exoplanetas.

2. Para buscar una configuración de hiperparámetros que podrían resultar en modelos con mejor desempeño, se recomienda ampliar la variedad de valores por hiperparámetro, así como la cantidad hiperparámetros por modelo durante el proceso de *Grid Search*.

3. Para buscar una mejora en el desempeño del modelo de K-Nearest Neighbors, se recomienda investigar si existen librerías que contengan una variante del modelo que pueda contrarrestar el efecto del desbalance en la base de datos.

4. Para profundizar en la relación entre variables, se propone entrenar distintos modelos donde, en cada iteración, se utiliza variables específicas paradeterminar si existe alguna combinación que pueda dar mejores resultados.

CAPÍTULO 11

Bibliografía

ALMA, Andrews, S., & Dagnello, S. (n.d.). *Twenty Protoplanetary Disks Imaged by ALMA*. https://public.nrao.edu/gallery/twenty-protoplanetary-disks-imaged-by-alma/.

Barboza, A., & Ulmer-Moll, S. (2020). *Classifying Exoplanets with Machine Learning.* https://presentations.copernicus.org/EPSC2020/EPSC2020_833_presentation.pdf.

Bhuva, L. (2025). *Understanding Feature Importance in Machine Learning.* understanding-feature-importance.

Clayton, G., Manry, B., & Rafiqi, S. (2019). *Machine Learning Pipeline for Exoplanet Classification.* scholar.smu.edu/cgi/viewcontent.cgi?article=1070.

Codecademy. (2025). *Feature importance*. https://www.codecademy.com/article/fe-feature-importance-final.

Dalianis, H. (2018). *Evaluation Metrics and Evaluation*. https://www.researchgate.net/publication/325122648_Evaluation_Metrics_and_Evaluation.

European Southern Observatory. (2007). *The Radial Velocity Method (Artist's Impression)*. https://www.eso.org/public/images/eso0722e/.

European Space Agency. (2019). *Exoplanet Detection Methods*. sci.esa.int/web/exoplanets/-/60655-detection-methods.

Gonzáles, J. (2021). *Algoritmos de Aprendizaje Supervisado en la Clasificación de Exoplanetas en Python.* http://repositorio.uan.edu.co/handle/123456789/5839.

IBM. (2024). *What is a Decision Tree?* https://www.ibm.com/topics/decision-trees.

IBM. (2024). *What is a Random Forest?* https://www.ibm.com/topics/random-forest.

IBM. (2024). *What is XGBoost?* https://www.ibm.com/topics/xgboost.

IBM. (2024). *What is the k-Nearest Neighbors (KNN) Algorithm?* ibm.com/topics/knn.

Johnson, J., Aller, K., Howard, A., & Crepp, J. (2024). *Giant Planet Occurrence in the Stellar Mass-Metallicity Plane*. ar5iv.labs.arxiv.org/html/1005.3084.

Kadir, E., Saha, P., Shrim, S., Ahsan, A., & Shoyaib, M. (2020). *A Proximity Weighted Evidential k Nearest Neighbor Classifier for Imbalanced Data.* https://pmc.ncbi.nlm.nih.gov/articles/PMC7206335/.

Müller, S., Baron, J., Helled, R., Bouchy, F., & Parc, L. (2024). *The mass-radius relation of exoplanets revised.* https://www.aanda.org/articles/aa/full_html/2024/06/aa48690-23/aa48690-23.html.

Murel, J., & Kavlakoglu, E. (2024). *¿Qué es una matriz de confusión?* https://www.ibm.com/es-es/topics/confusion-matrix.

NASA. (2024). *NASA Exoplanet Catalog.* https://science.nasa.gov/exoplanets/exoplanet-catalog/.

NASA. (2024). *What is a Gas Giant?* science.nasa.gov/exoplanets/gas-giant.

NASA. (2025). *What is an Exoplanet?* science.nasa.gov/exoplanets/planet-types.

NASA Exoplanet Science Institute. (2024). *NASA Exoplanet Archive.* https://exoplanetarchive.ipac.caltech.edu/.

NASA Visualization Studio. (2018). *Exoplanet Transit Animation.* https://svs.gsfc.nasa.gov/13022.

Optical Gravitational Lensing Experiment. (n.d.). *First Detection of an Extrasolar Planet with Microlensing.* https://ogle.astrouw.edu.pl/cont/4_main/epl/blg235/.

Perryman, M. (2011). *The Exoplanet Handbook.* https://www.w3schools.com/python/numpy/numpy_array_indexing.asp.

Pinte, C., van der Plas, G., Ménard, F., & et al. (2019). *Kinematic Detection of a Planet Carving a Gap in a Protoplanetary Disk. Nature Astronomy*, 3(1109–1114). https://doi.org/10.1038/s41550-019-0852-6.

SciKit Learn. (2024). *Decision Trees.* https://scikit-learn.org/stable/modules/tree.html.

Seager, S., Kuchner, M., Hier-Majumder, C. A., & Militzer, B. (2007). *Mass-Radius relationships for solid exoplanets.* https://iopscience.iop.org/article/10.1086/521346/pdf.

The Planetary Society. (n.d.). *Color-Shifting Stars: The Radial-Velocity Method.* https://www.planetary.org/articles/color-shifting-stars-the-radial-velocity-method.

The Planetary Society. (n.d.). *Space-Warping Planets: The Microlensing Method.* https://www.planetary.org/articles/space-warping-planets-the-microlensing-method.

The Planetary Society. (n.d.). *Timing Variations.* planetary.org/articles/timing-variations.

Yang, Y. (2020). *Intelligent System and Computing.* books.google.com/books?id=RmH9DwAAQBAJ.

CAPÍTULO 12

Anexo A

12.1. Códigos utilizados

En el siguiente repositorio se encuentran los archivos utilizados para extracción de nombres de planetas por medio de webscrapping y para el entrenamiento y evaluación de los modelos:
`https://github.com/Luis-RasconCalderon/ExoplanetClass.git`

Además, se encuentra una imagen en la que se puede ver en alta resolución el Árbol de Decisiones completo que se obtuvo como resultado al entrenar el modelo con los parámetros especificados en la sección 7.1