

# Machine Learning para el tratamiento de datos y la detección de exoplanetas mediante el método de tránsito



ETSI SISTEMAS  
INFORMÁTICOS

## Trabajo de Fin de Grado

Grado en Ingeniería de Sistemas de Información  
Curso 2019/2020

### Autor

José Javier Gómez de Diego

### Tutor

Fernando Ortega Requena



# Trabajo de Fin de Grado

*Machine Learning para el tratamiento de datos y la detección  
de exoplanetas mediante el método de tránsito*

Universidad Politécnica de Madrid  
Julio 2020



**Ad astra**



## Resumen

Uno de los siguientes pasos en la exploración espacial es encontrar planetas más allá del sistema solar que, potencialmente, puedan albergar signos de vida extraterrestre. Estos planetas que orbitan otras estrellas son conocidos como **exoplanetas**. Las complejas técnicas utilizadas para su detección recaban una inmensa cantidad de datos que deben ser cuidadosamente tratados y adecuados para su posterior análisis en busca de estos mundos.

Existen multitud de métodos para la detección de estos exoplanetas que arrojan una gran diversidad de datos. Estos métodos van desde la observación de estrellas en busca de movimientos radiales de las mismas provocados por distorsiones gravitacionales, hasta la observación en bruto del espacio vacío con el objetivo de detectar picos de luz producto de la deformación del espacio-tiempo derivada de la presencia de exoplanetas.

Uno de los métodos más comúnmente utilizados es conocido como el método de **tránsito**. Éste consiste en observar las estrellas en busca de disminuciones de luz provocadas por posibles exoplanetas transitando entre la estrella y el observador. Esta información queda plasmada en los datos recogidos por los telescopios, que deben ser procesados, tratados y analizados. Estas tareas se pueden llevar a cabo de forma masiva y automatizada mediante distintas técnicas de **machine learning**.

El **machine learning** ofrece la posibilidad de gestionar todo tipo de datos gracias a su naturaleza automatizada y a su punto más fuerte: el **aprendizaje**. Debido a la gran variedad de técnicas que se pueden aplicar en el tratamiento de datos, el **machine learning** es utilizado en un gran número de trabajos e investigaciones, entre las que se encuentra, por supuesto, la detección de exoplanetas.

Los datos derivados de estas tareas requieren un alto nivel de procesado y un alto grado de automatización debido a la compleja naturaleza y magnitud de los mismos. Para ello, existen ciertas técnicas que pueden realizar gran parte del **tratamiento** y, en concreto, la **clasificación** de los datos recogidos.

Los problemas que presentan los datos resultantes del método de tránsito suelen ser de dos tipos: un número de dimensiones excesivamente elevado, lo que dificulta el procesado y clasificación de los mismos y cuya solución pasa por la reducción de **dimensiones**; y el **desbalanceamiento** del conjunto de datos, lo que hace que, a la hora de clasificar, los algoritmos puedan desarrollar cierto sesgo hacia el tipo de datos mayoritarios, resultando en una clasificación poco ajustada a la realidad.

En cuanto a la clasificación, existen multitud de algoritmos que pueden ser entrenados partiendo de un conjunto de datos ya tratados para, posteriormente, lograr clasificar correctamente nuevos datos derivados de nuevas observaciones. De esta forma, el trabajo de categorizar los datos se puede llegar a automatizar en gran medida.

En este proyecto se presentarán las principales técnicas y modelos de machine learning para el tratamiento y clasificación de datos, se emplearán algunos de ellos para adecuar conjuntos de datos de observaciones realizadas por la misión *Kepler* de la NASA y, finalmente, se construirá un modelo de predicción y se analizará su precisión a la hora de detectar exoplanetas.

## Abstract

Next step in space exploration is to find planets beyond the Solar System which may potentially harbor life. These types of planets orbiting other stars are known as **exoplanets**. The complex techniques used for their detection collect an immense amount of data that must be carefully processed and adapted for further analysis in search of other worlds.

There are a large variety of methods and techniques for detecting these exoplanets that provides a vast diversity of data. These techniques go from stars observation looking for radial motion derived from gravitational distortions, to the crude monitoring of the empty space in the search of light spikes as a product of the space-time deformation due to exoplanets presence.

One of the most widely used techniques is called **transit**. It consists of observing stars to detect decreases in light brightness derived from a possible exoplanet transiting between the star and the observer. This information is captured with the data collected by telescopes, which must be processed, adapted and analyzed. These tasks can be performed automatically through different **machine learning** methods.

Machine learning offers the possibility to manage all kinds of data thanks to its automation and also its capacity to learn, which is its strongest feature. Owing to the vast diversity of techniques that can be applied to data processing, machine learning is used in a big number of research processes, including exoplanets hunting.

These tasks produce data that requires a high level of processing as well as automation due to its complexity and great magnitude. Therefore, there are certain methods that can carry out a large part of the **processing** and, specifically, the **classification** of the collected data.

The main challenges that data resulting from the transit method are usually two: an excessively high number of dimensions, which makes data processing and classification too heavy and makes it necessary to apply some dimensions **reduction** techniques; and an unbalanced dataset, which can make algorithms biased when it comes to classification, resulting in an inaccurate classification of the data.

Regarding to classification, there are plenty of algorithms that can be trained with a processed dataset to classify accurately new data from observations. Thus, classifying data can become a complete automated work.

In this project, the main machine learning models and techniques for data processing and classification will be introduced, some of them will be used to adapt datasets from the NASA's Kepler mission and, finally, a model for prediction will be built and analyzed how accurately it performs.

# Índice

Resumen .....	1
Abstract .....	2
Índice de figuras .....	5
Índice de tablas .....	6
Preludio .....	7
Introducción .....	8
Objetivos .....	12
General .....	12
Específico .....	12
Machine Learning .....	13
Aprendizaje supervisado .....	14
Árboles de decisión .....	14
Redes bayesianas .....	15
Regresión logística .....	16
Redes neuronales artificiales .....	17
Aprendizaje no supervisado .....	19
K-Medias .....	19
Clusterización Jerárquica .....	20
Density Based Scan Clustering -DBSCAN- .....	21
Aprendizaje por refuerzo .....	22
Detección de exoplanetas .....	23
Método de tránsito .....	24
Datos .....	27
Entrenamiento .....	27
Test .....	28
Problemas .....	28
Soluciones .....	28
Principal Component Analysis -PCA- .....	29
Data augmentation .....	30
Modelado de datos .....	31
Adecuación de los datos .....	32
Entrenamiento .....	34
Test .....	34
Reducción de dimensiones .....	35
Entrenamiento .....	36

<i>Test</i> .....	36
Balanceamiento de datos .....	36
<b>Modelo</b> .....	38
<b>Resultados</b> .....	39
<b>Futuros trabajos</b> .....	41
<b>Impactos ambientales y sociales</b> .....	42
<b>Conclusiones</b> .....	43
<b>Glosario de términos</b> .....	44
<b>Referencias</b> .....	45
<b>Bibliografía</b> .....	46

## Índice de figuras

<b>Figura 1.</b> Un algoritmo es una secuencia de instrucciones .....	13
<b>Figura 2.</b> Representación de un árbol de decisión .....	15
<b>Figura 3.</b> Representación de una red bayesiana con parámetros .....	16
<b>Figura 4.</b> Representación gráfica de la función logística .....	17
<b>Figura 5.</b> Representación gráfica una red neuronal artificial multicapa .....	17
<b>Figura 6.</b> Representación gráfica del funcionamiento de una neurona .....	18
<b>Figura 7.</b> Resultados de agrupaciones por K-Medias [16] .....	19
<b>Figura 8.</b> Resultados de agrupaciones por clusterización jerárquica [17] .....	20
<b>Figura 9.</b> Representación del resultado de la agrupación por DBSCAN .....	21
<b>Figura 10.</b> Resultados de agrupaciones por DBSCAN [18] .....	21
<b>Figura 11.</b> Diagrama del aprendizaje por refuerzo .....	22
<b>Figura 12.</b> Desviación del centro de masa de una estrella respecto al baricentro .....	23
<b>Figura 13.</b> Desviación de la luz provocada por la curvatura del espacio-tiempo .....	24
<b>Figura 14.</b> Curvas de luz producidas por el exoplaneta HAT-P-7b orbitando la estrella HAT-P-7 .....	24
<b>Figura 15.</b> Detalle de una curva de luz producida por el exoplaneta HAT-P-7b orbitando la estrella HAT-P-7: $t_t$ se refiere al tiempo total de tránsito; y $t_f$ , al tiempo en el que toda el área del cuerpo transita la estrella .....	25
<b>Figura 16.</b> Resultado de la aplicación del método PCA .....	29
<b>Figura 17.</b> Croquis del proceso de construcción del modelo .....	31
<b>Figura 18.</b> Flujo de la estrella nº 10 CON exoplanetas sin tratamiento .....	32
<b>Figura 19.</b> Flujo "suavizado" de la estrella nº 10 CON exoplanetas .....	32
<b>Figura 20.</b> Superposición de los anteriores flujos .....	33
<b>Figura 21.</b> Flujo desvinculado de la tendencia de fluctuación de la estrella nº 10 CON exoplanetas .....	33
<b>Figura 22.</b> Flujo desvinculado de la tendencia de fluctuación normalizado .....	33
<b>Figura 23.</b> Flujo desvinculado de la tendencia de fluctuación normalizado y sin datos atípicos superiores .....	34
<b>Figura 24.</b> Varianza acumulada respecto a las dimensiones del dataset de entrenamiento .....	35
<b>Figura 25.</b> Flujo de la estrella nº 10 CON exoplanetas con 603 dimensiones tras aplicar PCA .....	35
<b>Figura 26.</b> Arquitectura de la red neuronal utilizada para la predicción .....	38

## Índice de tablas

<b>Tabla 1.</b> Muestra de los datos de entrenamiento .....	27
<b>Tabla 2.</b> Muestra de los datos de test .....	28
<b>Tabla 3.</b> Muestra de los datos de entrenamiento tras el primer tratamiento .....	34
<b>Tabla 4.</b> Muestra de los datos de test tras el primer tratamiento .....	34
<b>Tabla 5.</b> Muestra de los datos de entrenamiento tras aplicar PCA .....	36
<b>Tabla 6.</b> Muestra de los datos de test tras aplicar PCA .....	36
<b>Tabla 7.</b> Muestra de los datos de test tras el balanceamiento de los datos .....	37
<b>Tabla 8.</b> Clasificación de datos según 'LABEL' .....	37
<b>Tabla 9.</b> Matriz de confusión de las predicciones del modelo .....	39
<b>Tabla 10.</b> Matriz de confusión de las predicciones del modelo con los datos en bruto .....	40

## Preludio

La evolución y desarrollo del ser humano han estado marcados por una intrínseca necesidad de conocimiento. Esta necesidad ha llevado a la humanidad a adentrarse en los confines de lo desconocido, a profundizar en la naturaleza de su realidad y a expandir su conciencia más allá, originando una de las herramientas más poderosas jamás dominadas: la **exploración**.

Desde los primeros seres vivos en dejar atrás el medio acuático, los primeros en viajar a tierras desconocidas, los primeros en cruzar los océanos y los primeros en levantar la mirada a las estrellas, la vida es sinónimo de exploración. Expandir el **conocimiento** hacia otros lugares ha permitido -y permite- al ser humano progresar y coexistir con la naturaleza.

La exploración hacia el conocimiento es un camino arduo y laborioso cuya consecución alberga la mayor de las conquistas: la prosperidad de la humanidad. Tras conquistar la tierra, el mar y el aire, el siguiente reto que se le presenta al ser humano es el que determinará su existencia en el universo: la búsqueda de nuevos mundos.

El futuro del hombre siempre será alcanzable, pero la razón de alcanzarlo nunca debe serlo. Tener un objetivo accesible permite luchar hasta conseguirlo; pero tener una razón, una causa **infinita**, hace que la humanidad jamás deje de luchar, de conocer. Y es que es el conocimiento la causa infinita de la humanidad; es el conocimiento lo que sobrevive al hombre.

Transformar lo desconocido en conocido es el legado que todo ser humano aspira a dejar tras de sí. El conocimiento consigue convertir al hombre en humanidad, y le ha capacitado a dar el salto a los continentes, a los mares, a los océanos, *a las estrellas*.

# Introducción

La búsqueda de exoplanetas es una tarea que requiere de complejos métodos tanto para la recogida de datos como para su tratamiento y procesado. Hasta la fecha, se han descubierto 4171 exoplanetas [1] -07/07/2020- mediante varias técnicas.

Uno de los métodos más comúnmente utilizados es el de velocidades radiales. Consiste en observar las estrellas en busca de movimientos tambaleantes provocados por una desviación entre el centro de masa de las estrellas y el baricentro del sistema. En otras palabras, que una estrella esté girando alrededor de un punto. Esto es sinónimo de que existe un cuerpo lo suficientemente masivo como para constituir un exoplaneta que está orbitando alrededor de la estrella. Debido a las interacciones gravitacionales -la estrella atrae al exoplaneta y viceversa-, el sistema estrella-exoplaneta gira en torno a un punto llamado baricentro que no coincide con el centro de masas de la estrella, haciendo que ésta reproduzca los mencionados movimientos tambaleantes desde el punto de vista de un observador externo. Al observar las estrellas, se recogen los datos de su posición en varios puntos de tiempo para ser analizados y determinar si dicho movimiento existe y si es provocado por un exoplaneta.

Otra técnica conocida es la de la microlente. Este método consiste en observar el espacio vacío con el objetivo de detectar picos de luz producidos por el paso de un exoplaneta por delante de varios objetos. Esto ocurre debido a que las grandes agrupaciones de materia -como son los exoplanetas- provocan una deformación del espacio-tiempo suficientemente marcada como para curvar la luz de forma detectable a su paso. Esta deformación provoca que la luz proveniente de los cuerpos situados detrás del exoplaneta con respecto al observador se curve, desviando su trayectoria y enfocándose hacia donde se encuentra el observador, tal y como hacen las lentes gracias a la curvatura del cristal. Gracias a esto, el observador detectará un incremento de la luz observada en un punto del espacio producto del efecto de lente gravitacional provocada por el paso de un posible exoplaneta.

El problema que presentan los dos métodos de detección de exoplanetas anteriores es que, con la tecnología actual, solo se pueden encontrar exoplanetas varias veces más masivos que La Tierra, que normalmente son gigantes gaseosos. Esto se debe a que la precisión de los instrumentos actuales solo permite percibir de forma evidente los efectos provocados por grandes exoplanetas, pero no de los que poseen una masa parecida a la de La Tierra.

Para solucionar este problema, se utiliza el método conocido como tránsito. Esta técnica consiste en observar la luz proveniente de las estrellas en busca de disminuciones periódicas en su intensidad. Este fenómeno se produce cuando un objeto transita entre la estrella y el observador, que es exactamente en lo que consiste un eclipse. Durante un eclipse de sol, un observador en La Tierra experimenta una reducción en la intensidad de la luz recibida de la estrella ya que la Luna bloquea parcialmente -o, en algunos casos, totalmente- dicha luz. Cuando este fenómeno se da en estrellas lejanas y con exoplanetas rocosos de tamaño medio que las orbitan, esta disminución del brillo percibido de las estrellas resulta muy inferior, pero es lo suficientemente fuerte como para poder ser detectado con los instrumentos disponibles actualmente. El resultado de estas observaciones son mediciones de la intensidad de la luz de las estrellas en varios momentos de tiempo que, en el caso de que un cuerpo transite por delante, presentaran unos valles o curvas de luz derivadas del bloqueo de luz que el cuerpo ha producido. Estos datos deben ser minuciosamente analizados para encontrar dichas curvas de luz y determinar si éstas son provocadas por exoplanetas. Como resultado, no solo se obtiene la respuesta de si existe o no un exoplaneta orbitando una estrella lejana, sino que de las curvas

de luz se puede abstraer una gran cantidad de información acerca del tamaño del planeta, velocidad, período orbital, distancia con su estrella e, incluso, analizando el espectro luz se puede determinar la composición de su atmósfera.

Observar las estrellas recogiendo tales cantidades ingentes de datos y tratar de sacar conclusiones conlleva una serie de procesos de gran complejidad -desde el tratamiento de los datos hasta su clasificación- que requieren una gran capacidad de cálculo. Para realizar estas tareas se hace uso del *machine learning*.

El *machine learning* es una rama de la inteligencia artificial que tiene como objetivo desarrollar algoritmos que sean capaces de aprender de la experiencia y realizar tareas tal y como la inteligencia humana lo haría. Disponer de estos algoritmos supone poder realizar distintos tratamientos y procesados de enormes cantidades de datos de forma automatizada gracias a esa capacidad de aprendizaje. Dependiendo de cómo se realice dicho aprendizaje, los algoritmos de *machine learning* se pueden clasificar principalmente de tres formas: aprendizaje supervisado, aprendizaje no supervisado y aprendizaje por refuerzo.

En el aprendizaje supervisado, el objetivo principal es transformar una entrada en una salida. Para ello, se le suministra al algoritmo un conjunto de datos de entrada conocidos que resultarán en un conjunto de datos de salida. Este *output* es comparado con los datos que el algoritmo debería haber producido -y que, evidentemente, se conocen- para que el algoritmo aprenda y reconfigure sus parámetros con el objetivo de que, la próxima vez que se le suministre ese *input*, produzca la salida deseada. Este proceso se denomina entrenamiento, y prepara al algoritmo para responder acertadamente a cualquier conjunto de datos de entrada. Entre los algoritmos de aprendizaje se encuentran los de regresión -predicciones meteorológicas, estimaciones de esperanza de vida, predicciones de actividad bursátil, etc.-, utilizando técnicas como la regresión logística; y los de clasificación -reconocimiento de imágenes, diagnósticos clínicos, detección de fraude, etc.- En este último grupo se encuentran algoritmos como los árboles de decisión, redes bayesianas y redes neuronales artificiales.

En cuanto al aprendizaje no supervisado, a diferencia del supervisado, no se conocen los datos de salida que el algoritmo debería producir, sino que lo que se busca es encontrar patrones en el conjunto de datos de entrada que permitan clasificarlos según esas regularidades. El aprendizaje no supervisado puede provocar ciertos inconvenientes ya que estos algoritmos pierden el contexto de los datos, pudiendo encontrar correlaciones espurias que, a pesar de encajar de forma precisa, carezcan de total sentido al no existir una correlación real entre los datos. Es por ello que la dificultad en este tipo de aprendizaje radica en el estudio previo de los datos y en determinar qué algoritmo encaja mejor con los mismos. Estos son problemas recurrentes en los modelos de *clustering*, entre los que se encuentran K-medias, *clusterización jerárquica* y *Density Based Scan Clustering -DBSCAN-*.

Por último, el aprendizaje por refuerzo consiste en avanzar hacia un estado más favorable para la consecución de un objetivo deseado. En el caso del ajedrez, por ejemplo, estos algoritmos tendrían como objetivo deseado el jaque mate, por lo que la decisión de qué movimiento realizar se basará en si el estado siguiente a dicho movimiento es un estado que se acerque al objetivo final. Para determinar esto, se utiliza un sistema de recompensas que responden a la acción que ha tomado el algoritmo. Por lo tanto, al realizar una acción, el modelo se retroalimenta con la recompensa suministrada y el estado posterior resultante de la acción.

Todos estos métodos de *machine learning* son imprescindibles para realizar procesos de tratamiento y clasificación de datos complejos como lo son los datos recogidos mediante los

distintos métodos de detección de exoplanetas. El conjunto de datos que será utilizado en este proyecto, producto de la búsqueda de exoplanetas mediante el método de tránsito, presenta una serie de problemas que deben ser resueltos antes de proceder a construir un sistema para su clasificación.

El primero de los problemas presentes en estos datos es el elevado número de dimensiones que presentan: un total de 3197. Cada una de ellas representa una medición de la intensidad de luz percibida de una estrella en cada momento de tiempo. Muchas de estas dimensiones resultan muy parecidas entre sí y no aportan información relevante para el estudio, por lo que se pueden descartar un gran número de las mismas sin perder la información esencial de los datos.

El segundo problema es el desbalanceamiento del conjunto de datos. Esto se debe a que el número de estrellas que poseen algún exoplaneta confirmado representa una ínfima parte de todo el conjunto de datos -menos del 1%- . Esto supone que, a la hora de entrenar el modelo, dicho desajuste provoque un sesgo en el algoritmo que lo lleve a una clasificación acertada dada la carencia de datos relevantes durante en proceso de aprendizaje.

El *machine learning* puede solucionar estos problemas gracias a las distintas técnicas de reducción de dimensiones y *data augmentation*.

En cuanto al problema del número de dimensiones, existe una técnica que realiza un procedimiento estadístico para seleccionar las dimensiones más relevantes para el sistema: *Principal Component Analysis -PCA-*. En pocas palabras, esta técnica consiste en obtener la varianza de cada dimensión, pudiendo así descartar aquellas que no son relevantes y reduciendo en gran medida el número total de las mismas.

Para solventar del desbalanceamiento del conjunto de datos, las técnicas de *data augmentation* permiten tanto aumentar los datos interesantes para el sistema -*oversampling*- como eliminar los datos que se repiten a lo largo de todo el *dataset* y desbalancean todo el conjunto de datos -*undersampling*-. El *oversampling* consiste en replicar los datos relevantes y aplicar ciertas transformaciones lo suficientemente fuertes como para que se diferencien de los originales, pero sin que lleguen a diferenciarse tanto como para desvirtuar la información esencial de los datos. Estas técnicas se suelen aplicar comúnmente en *datasets* de imágenes, en los que se replican aquellas que son relevantes y se les aplican cambios como la inversión, rotación, aplicación de filtros, ruido gaussiano, etc.

Una vez realizada una primera adecuación de los datos y aplicadas las técnicas de reducción de dimensiones y *data augmentation*, se puede construir y entrenar un sistema inteligente con los conjuntos de datos resultantes para su clasificación.

El modelo se trata de una red neuronal artificial cuya configuración en cuanto a número de capas, neuronas, funciones de activación, etc. se establece conforme a los resultados que aportan las distintas parametrizaciones.

Tras entrenar el modelo y analizar los resultados se pueden obtener conclusiones en cuanto a la precisión de modelo y cómo se puede mejorar. Este último paso es muy delicado ya que requiere conocer todo el proceso de tratamiento de datos que se ha llevado a cabo, así como el modelo de clasificación, para poder detectar posibles puntos débiles y cuellos de botella que, de ser solventados, puedan mejorar no solo el resultado final, sino todo el proceso llevado a cabo.

Una vez realizado el análisis de los resultados y obtenidas las conclusiones del proceso, resulta interesante indicar, para futuros trabajos, qué es aquello que funciona bien y qué es lo que se puede mejorar de todo el proceso para optimizar tanto el proceso como los resultados.

## Objetivos

### General

El objetivo último del presente proyecto es lograr **detectar** la presencia de **exoplanetas** mediante el análisis de los datos resultantes de las observaciones de la luz procedente de estrellas mediante del método de **tránsito**.

Los datos utilizados tanto para entrenamiento como para test se han obtenido de Kaggle [2] y corresponden a los recogidos por la NASA en una de las misiones *Kepler*, con los que se plantean una serie de problemas y objetivos.

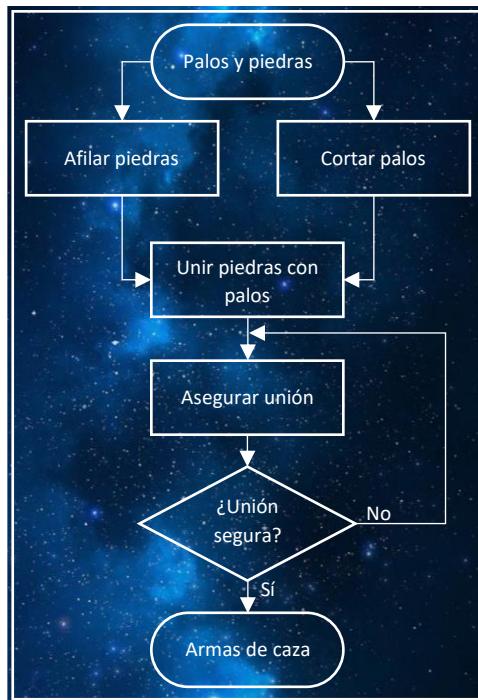
### Específico

Los objetivos específicos que resultan del objetivo general se pueden dividir en los siguientes:

- *Tratamiento y adecuación* de los datos que van a ser analizados a través de varias técnicas seleccionadas tras un estudio previo de los mismos. Estas técnicas servirán para un primer tratamiento de los datos en bruto y una posterior adecuación para compensar los desajustes -PCA, *data augmentation*, etc.-
- *Construcción y entrenamiento* de un sistema inteligente que, a través de los datos ya tratados, sea entrenado y pueda predecir la existencia de un exoplaneta orbitando una estrella.
- *Prueba* del sistema inteligente mediante los datos de test para analizar su fiabilidad y precisión.

## Machine Learning

En el núcleo de la programación reside el concepto de algoritmo. Un algoritmo es una secuencia no ambigua de instrucciones que se deben llevar a cabo para transformar un *input* en un *output* [3]. Esto es lo que los seres humanos llevan haciendo durante toda su existencia: desde transformar un conjunto de palos y piedras en armas para cazar -*Figura 1*-, hasta la creación de vacunas a partir de una serie de elementos químicos y bases nitrogenadas.



*Figura 1. Un algoritmo es una secuencia de instrucciones*

Estas acciones se realizan mediante un número determinado de **instrucciones** y cuyo resultado es fruto de la consecución de todas ellas. Dicho resultado no depende de quién o qué lleve a cabo dichas instrucciones mientras éstas se realicen tal y como están definidas, por lo que cualquiera que pueda seguir esos pasos logrará el resultado esperado. Será la naturaleza de las instrucciones la que limite quién o qué las puede realizar.

Cuando se trata de algoritmos de una naturaleza **lógica**, la computación es capaz de abstraer dicha lógica de las instrucciones a niveles que puede procesar, resultando en el resultado lógico esperado del problema. Sin embargo, los seres humanos son muy efectivos y eficientes resolviendo muchos de estos problemas que las máquinas no pueden resolver, o no lo hacen en una forma y tiempo útiles -reconocer rostros en fotografías, moverse entre multitudes sin colisionar con otros, jugar al ajedrez o conducir-. Estos problemas se denominan **insolubles** o NP-difíciles, y es extremadamente improbable que incluso la mejora de carácter exponencial en la velocidad de cálculo que el hardware ha experimentado -“Ley de Moore”- tenga un impacto significativo en la capacidad para resolver casos complejos de problemas insolubles [4]. De esta problemática surge la **inteligencia artificial**.

La inteligencia artificial es el estudio del diseño de sistemas inteligentes. Su objetivo principal es entender los principios que hacen posible el comportamiento inteligente, tanto en sistemas naturales como artificiales. La hipótesis central se basa en que el razonamiento es

computación. El objetivo, desde el punto de vista de la ingeniería, es especificar métodos para el diseño de artefactos inteligentes útiles [5]. Una de las ramas de la inteligencia artificial es el **machine learning** o aprendizaje automático. El *machine learning* tiene como objetivo desarrollar técnicas para que las máquinas aprendan, entendiendo el concepto de aprender como la mejora del desempeño con la experiencia; es decir, que la habilidad de resolver un problema no estaba presente en la máquina cuando ésta fue concebida [6].

La idea del *machine learning* es lograr resolver los problemas insolubles haciendo que las máquinas aprendan por sí solas de la experiencia. Este proceso comienza con un modelo genérico que posee una serie de parámetros y que, dependiendo de cómo se configuren, éste pueda llegar a realizar todo tipo de tareas. Esta característica del modelo inicial es esencial para que el aprendizaje resulte en un modelo útil, ya que el ejercicio de **aprender** consiste en ajustar esos parámetros tal que el modelo genérico inicial se adapte de la mejor forma a los datos suministrados para el entrenamiento. Tras esto, el modelo con los parámetros ya ajustados se ha especializado en una tarea particular, convirtiéndose en el algoritmo para realizarla [3].

Existen numerosos tipos de modelos cuya forma de aprendizaje se puede clasificar como aprendizaje **supervisado**, aprendizaje **no supervisado** y aprendizaje por **refuerzo**.

## Aprendizaje supervisado

El principal objetivo del aprendizaje supervisado es transformar un *input* en un *output* mediante un proceso de entrenamiento. En este caso se conocen ambos conjuntos de datos, y los datos de salida son contrastados con datos correctos que son provistos por un supervisor. El modelo, por tanto, aprende gracias a esa comparación entre los datos de salida propios y los suministrados externamente, lo que le permite retroalimentarse y ajustar sus parámetros para que, en la siguiente iteración, los datos de salida se ajusten mejor a los datos provistos a modo de ejemplo.

El aprendizaje supervisado se suele emplear principalmente en problemas de **regresión** -predicciones meteorológicas, de esperanza de vida, bursátiles, etc.-, como la regresión lineal; y en problemas de **clasificación** -reconocimiento de imágenes, diagnósticos clínicos, detección de fraude, etc.-. En estos últimos, según cómo se clasifique el *output*, los modelos pueden hacerlo de forma binaria o multiclase. La clasificación binaria consiste en asignar cada dato a un grupo de un total de dos, mientras que la multiclase tiene más de dos opciones para clasificar cada dato. Los algoritmos más comunes de clasificación son los árboles de decisión, la regresión logística, las redes bayesianas y las redes neuronales artificiales, que se exponen a continuación.

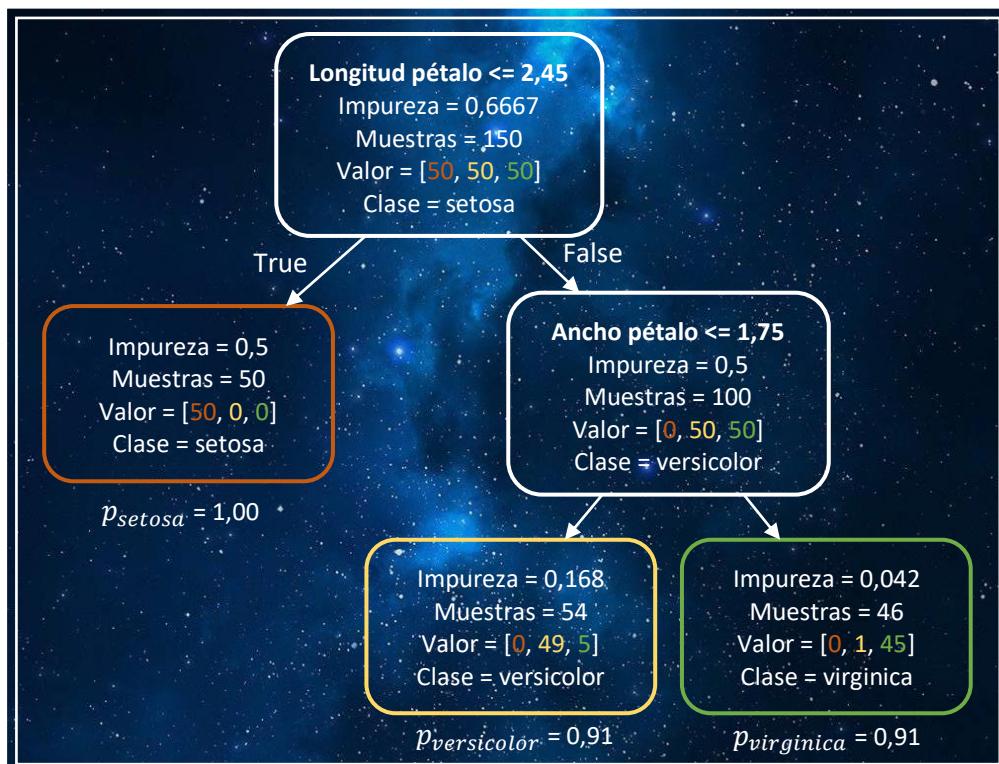
## Árboles de decisión

Este modelo de clasificación consiste en ir dividiendo cada punto de datos -entendido como lista de pares atributo-valor- según sus valores de forma que va siguiendo un camino de nodos de decisión o “rama” hasta llegar a la clasificación final, denominada nodo hoja. Para elegir los atributos por los cuales dividir en cada nodo de decisión, se hace uso del concepto de entropía,

$$\text{Entropía}(S) = \sum_{i=1}^n -p_i \log_2 p_i$$

donde  $S$  es una colección de objetos,  $p_i$  es la probabilidad de los posibles valores,  $i$  es el número de posibles respuestas de los objetos y  $n$  es el número de clases.

Por lo general, el atributo con menor entropía es aquel por el que mejor se van a clasificar las instancias. Esto supone que la entropía, de acuerdo a la propia naturaleza del concepto, de la clasificación tiende a ser mínima y, por lo tanto, resulta en una ordenación óptima, dado que la entropía se puede definir en última instancia como la magnitud que mide la pérdida de información -ver *entropía* en el glosario de términos-. En consecuencia, menor entropía supone mayor información inherente al estado del sistema.



*Figura 2. Representación de un árbol de decisión*

En la *Figura 2* se muestra un ejemplo de un árbol de decisión para clasificar plantas basándose en unos atributos específicos.

Para evitar el sobreajuste del modelo, se utiliza el método de poda. Esto consiste en eliminar las ramas cuya discriminación se sostiene en atributos que no son relevantes -que no disminuyen de forma significativa el error de validación cruzada- y solo se ajustan a un conjunto de datos específico. En adición, este proceso puede ir acompañado, o ser sustituido, por la pre-poda, que consiste en evitar que, durante la construcción del árbol, se creen dichas ramas que se basan en estos atributos irrelevantes para tomar decisiones. Ambos métodos, la poda y la pre-poda, evitan que el modelo se sobreajuste a un conjunto de datos determinado.

Pese a que este modelo es muy susceptible a la parametrización inicial, el árbol de decisión presenta una gran facilidad de uso, así como una gran resistencia a datos atípicos.

## Redes bayesianas

Son una representación gráfica en forma de grafo dirigido de dependencias para razonamiento probabilístico en la cual los nodos representan variables aleatorias; y los arcos, relaciones de dependencia directa entre las variables [7]. Este modelo hace uso del teorema de Bayes,

$$P(A_i|B) = \frac{P(B|A_i)P(A_i)}{P(B)}$$

donde  $A_i$  es la clase y  $B$  un suceso,  $P(A_i)$  es la probabilidad de la clase a priori;  $P(B|A_i)$ , la probabilidad de  $B$  en la hipótesis  $A_i$ ; y  $P(A_i|B)$ , las probabilidades a posteriori.

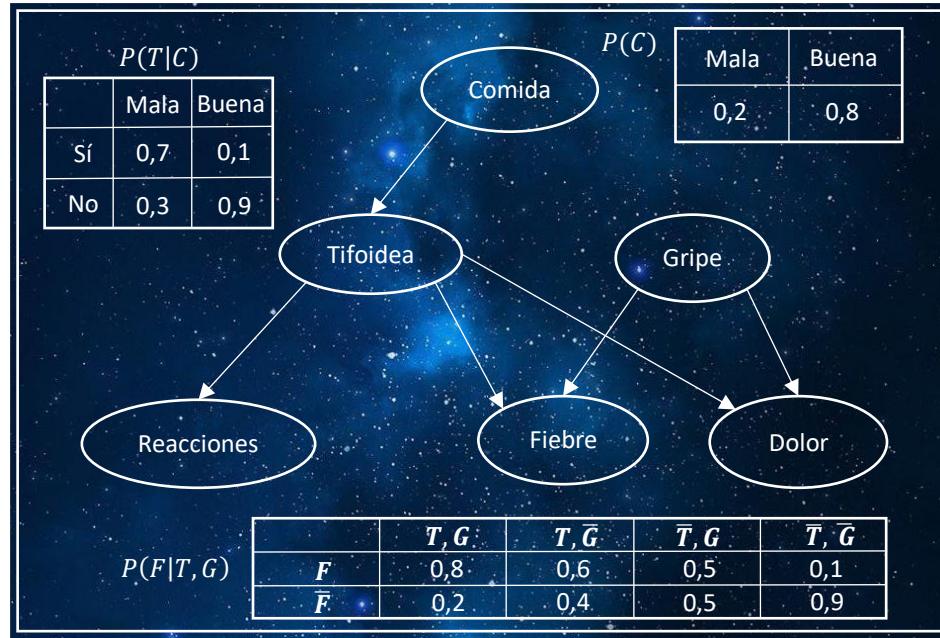


Figura 3. Representación de una red bayesiana con parámetros

El razonamiento probabilístico consiste en propagar los efectos de la evidencia a través de la red para conocer la probabilidad *a posteriori* de las variables. Es decir, se les dan valores a ciertas variables y se obtiene la probabilidad posterior de las demás variables dadas las variables conocidas -el conjunto de variables conocidas puede ser vacío, en cuyo caso se obtienen las probabilidades *a priori*- [7]. En la Figura 3 se muestra un ejemplo de red bayesiana para predecir las causas de ciertos síntomas mediante sus probabilidades condicionadas.

## Regresión logística

Se trata de un modelo estadístico que utiliza una función logística para modelar una variable dependiente binaria, aunque existen otras aplicaciones más complejas [8] como la regresión logística multinomial -variables dependientes no binarias- o la regresión logística ordinal, entre otras. Consiste en estimar los parámetros de un modelo logístico donde el logaritmo de las probabilidades de uno de los valores binarios es una combinación lineal de una o más variables independientes binarias o continuas. Para ello, se hace uso de la función *logit*,

$$\text{logit}(p) = \log\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_n x_n$$

donde  $p$  es la probabilidad de que ocurra uno de los valores binarios de la variable dependiente,  $x_n$  son variables independientes y  $\beta_n$  son parámetros del modelo. Esta ecuación se puede reescribir de tal forma que

$$p = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_n x_n)}}$$

En la Figura 4 se muestra de forma gráfica esta función.

De este modo, se resuelve de forma directa la probabilidad. Este modelo se podría interpretar como una red neuronal de una sola capa.

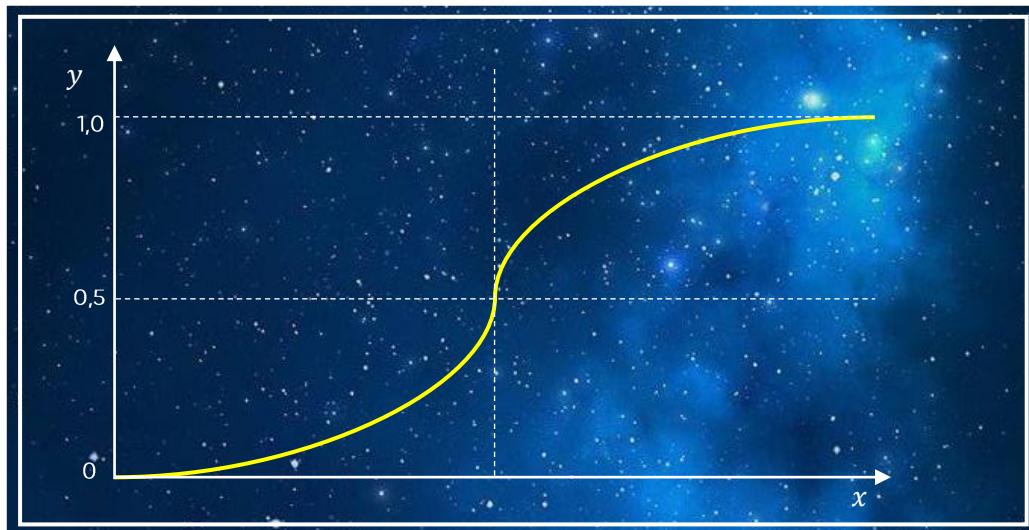


Figura 4. Representación gráfica de la función logística

## Redes neuronales artificiales

Este modelo está inspirado en el comportamiento de las neuronas y sus conexiones en el cerebro. Las redes neuronales artificiales se pueden entender como modelos computacionales paralelos compuestos de unidades de procesamiento adaptativo densamente interconectadas [9]. Estas unidades de procesamiento son las denominadas **neuronas**, que se conectan entre sí formando una o varias **capas** y que, a su vez, conforman una **red** unidireccional y recurrente, puesto que se retroalimenta.

Una neurona artificial se puede definir como un dispositivo simple de cálculo que, a partir de un *input*, proporciona un *output* único. Dependiendo de estos datos, se pueden identificar tres tipos: las neuronas de entrada, que no tienen conexiones entrantes provenientes de otras neuronas sino del entorno exterior; las neuronas de salida, que no tienen conexiones salientes hacia otras neuronas sino hacia el exterior -salidas de la red-; y las neuronas ocultas, que tienen tanto conexiones entrantes como salientes con otras neuronas. Si una red neuronal no tiene ninguna capa de neuronas ocultas, se denomina perceptrón simple; mientras que, si presenta una o más de estas capas, se conoce como perceptrón multicapa -este último está representado en la Figura 5, donde se muestra una red neuronal con dos capas ocultas-.

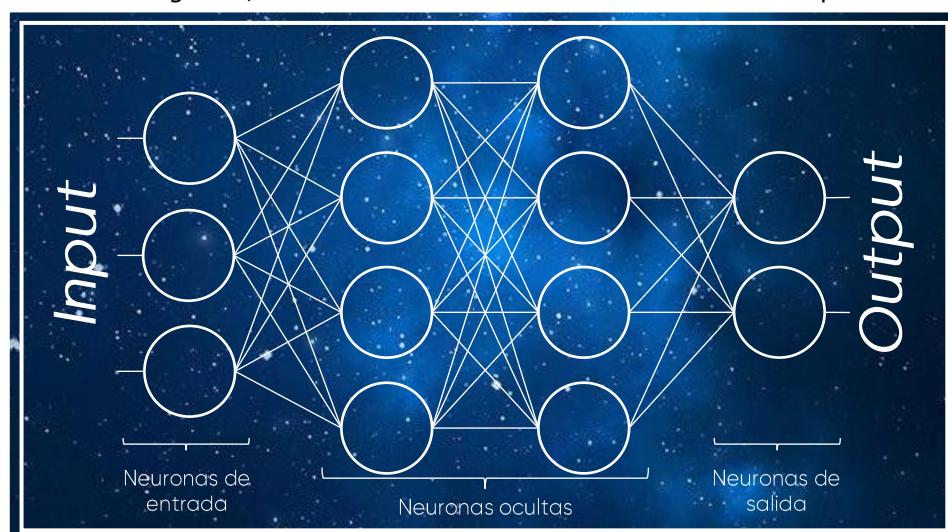


Figura 5. Representación gráfica una red neuronal artificial multicapa

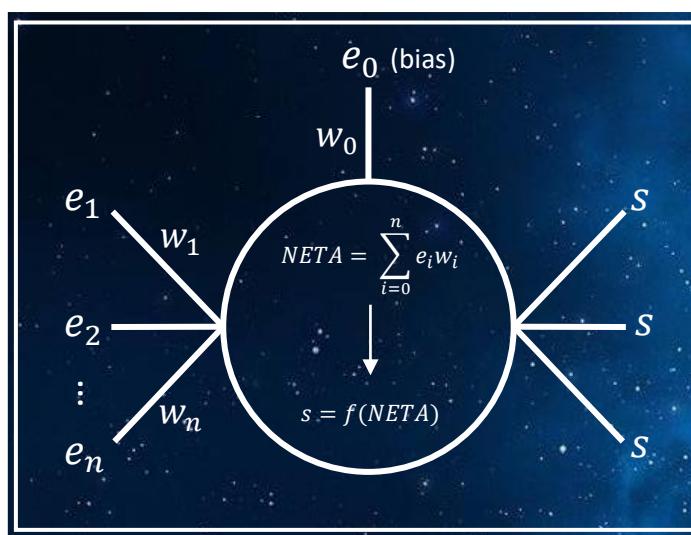
Cada neurona oculta recibe una serie de *inputs* provenientes de otras neuronas que procesa para producir un *output* que es enviado a cada una de las neuronas de la siguiente capa. La entrada neta que recibe se expresa de la siguiente forma:

$$NETA = \sum_{i=0}^n e_i w_i$$

donde  $e_i$  es la entrada  $i$ -ésima;  $w_i$ , el peso  $i$ -ésimo asociado a la conexión  $i$ -ésima con el que las entradas son ponderadas; y  $n$  es el número de conexiones entrantes. Una vez calculada la entrada neta de la neurona, se aplica una función cuyo resultado será el *output* de la neurona,

$$s = f(NETA)$$

donde  $s$  es la salida de la neurona y la función  $f$  se establece previamente como parámetro de la red neuronal artificial. Este proceso está representado en la *Figura 6*, donde se muestra una neurona, sus entradas, sus salidas, la función *NETA* y la función de salida.



*Figura 6. Representación gráfica del funcionamiento de una neurona*

El aprendizaje de una red neuronal se inicia calculando el error de la salida de la red - con respecto a la salida deseada- y corrigiendo los pesos de las conexiones hacia atrás - retropropagación- para hacer que este error sea mínimo. La corrección de los pesos de las conexiones se calcula de tal forma que

$$\Delta w_{ij} = \alpha \delta_j e_i$$

donde  $\Delta w_{ij}$  es la corrección que hay que aplicar al peso  $i$ -ésimo - $w_i$ - de la  $j$ -ésima neurona;  $e_i$ , la entrada  $i$ -ésima de la de la  $j$ -ésima neurona; y  $\alpha$ , el factor de aprendizaje que se establece como parámetro de la red y que regula la velocidad y precisión con la que aprende el sistema.  $\delta_j$  varía dependiendo de si se trata de una neurona oculta o de salida,

$$\delta_j = (d_j - s_j)s_j(1 - s_j), \text{ si } j \text{ es una neurona de salida}$$

$$\delta_j = \sum_k \delta_k w_{jk} s_j(1 - s_j), \text{ si } j \text{ es una neurona oculta},$$

donde  $k$  hace referencia a todas las neuronas de la capa inmediatamente superior a la de la neurona  $j$  [10]. Este tipo de aprendizaje de redes neuronales artificiales es el más utilizado en la práctica.

## Aprendizaje no supervisado

Si en el aprendizaje supervisado se le suministra a un modelo un *output*, asociado al *input*, con los datos correctos para retroalimentarse, en el aprendizaje no supervisado no existen datos de salida para realizar dicha comparación y posterior aprendizaje. Lo que se trata de lograr con el aprendizaje no supervisado es encontrar patrones en los datos de entrada que permitan clasificar dicho *input* según esas regularidades [3].

Los modelos que aprenden de forma no supervisada pueden, en ocasiones, encontrar correlaciones espurias que, si bien encajan de forma muy precisa, carecen de sentido al no existir ningún tipo de correlación entre los datos. Esto se debe a que el modelo es capaz de clasificar los datos y encontrar patrones recurrentes en ellos, pero no es capaz de abstraer la naturaleza de los mismos, perdiendo el sentido de contexto de los datos. Estos son problemas recurrentes de los modelos de ***clustering*** o agrupación. Para tratar de solventarlos, se requiere de un estudio para determinar qué algoritmos van a dar un resultado que se ajuste de forma lógica a los datos.

Entre los principales algoritmos de aprendizaje no supervisado y que se exponen a continuación se encuentran K-medias, *clusterización jerárquica*, *Density Based Scan Clustering - DBSCAN-*; además de métodos de reducción de dimensiones como *isomap*, *t-SNE*, *Singular Value Decomposition -SVD-* y, como se expone más adelante, *Principal Component Analysis -PCA-*.

### K-Medias

Es el más utilizado como punto de referencia para la evaluación de otros algoritmos gracias a su fácil implementación y eficiencia computacional. Funciona de tal manera que, dado un número  $k$  de *clusters*, selecciona de forma aleatoria el centroide de cada uno de ellos. Luego, asigna cada dato al *cluster* cuyo centroide tenga una menor Distancia Cuadrada Euclidiana,

$$d(x, y)^2 = \sum_{j=1}^m (x_j - y_j)^2 = \|x - y\|_2^2$$

donde  $x$  e  $y$  son dos puntos en el espacio de dimensión  $m$ . Tras esto, se calcula el error cuadrático mínimo de los puntos de datos agrupados en cada *cluster* y se sustituye su centroide por ese punto. Por último, se vuelven a recalcular los *clusters* con los nuevos centroides y se vuelve a repetir el proceso un número de veces previamente determinado.

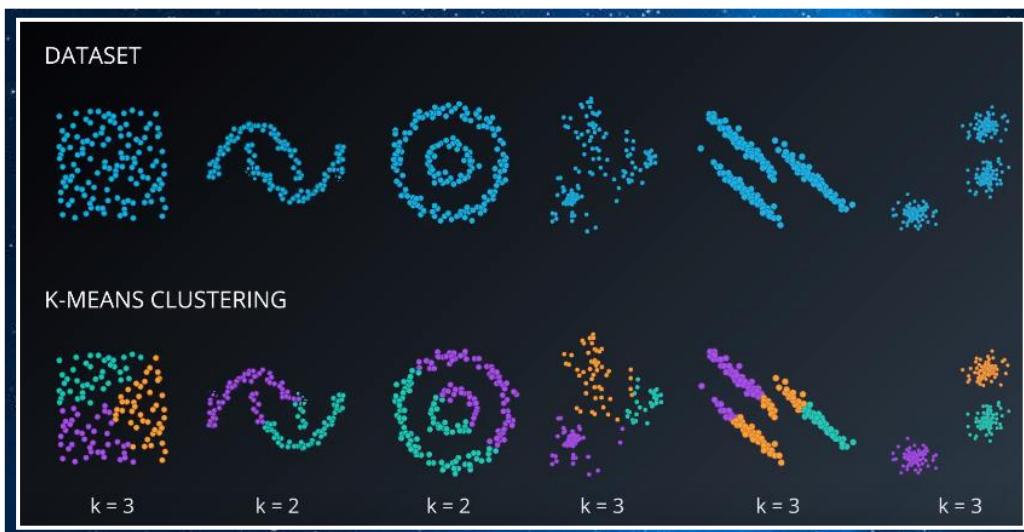


Figura 7. Resultados de agrupaciones por K-Medias [16]

Como se puede observar en la *Figura 7*, este algoritmo resulta bastante acertado clasificando agrupaciones de datos que presentan una distribución esférica -como es el caso de la sexta agrupación-, siempre y cuando se conozcan previamente el número de agrupaciones. En otros casos, el algoritmo K-Medias no es capaz de dividir las agrupaciones de forma intuitiva.

## Clusterización Jerárquica

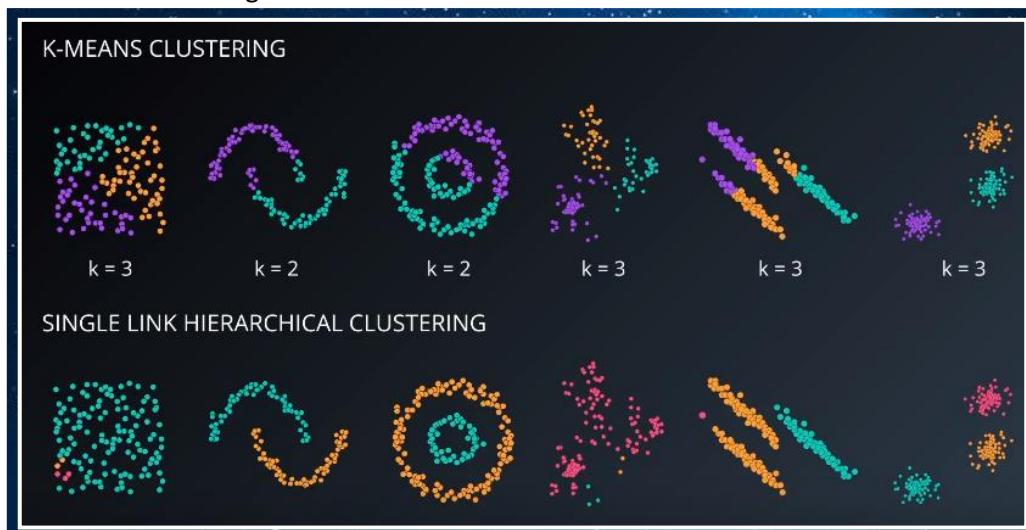
Este algoritmo presenta algunas ventajas con respecto a K-Medias, siendo la principal de ellas que no es necesario conocer de antemano el número de *clusters*. Existen dos aproximaciones en este método de agrupación: divisiva y aglomerativa.

Por un lado, la divisiva comienza clasificando todos los puntos de datos en una sola agrupación para, posteriormente, dividir dicho grupo de forma iterativa hasta llegar a agrupar cada punto en un *cluster*.

Por el contrario, la aglomerativa parte de que cada punto de datos forma un *cluster* en sí mismo y, de forma iterativa, los va fusionando hasta llegar a una sola agrupación de todos los puntos. Para realizar este procedimiento, este algoritmo puede, por un lado, ejecutar el acoplamiento simple, que calcula las distancias entre los puntos -cada uno representa un *cluster* diferente en este momento- fusionando los cúmulos para los cuales la distancia entre los miembros más similares es la más pequeña.; o, por otro lado, el acoplamiento complejo, que es similar al anterior con la diferencia de que la fusión de cúmulos se realiza con los *clusters* más diferentes, y no los más similares.

Este algoritmo no finaliza únicamente con la mejor agrupación posible de los puntos de datos, sino que su *output* constituye una gama de *clusters* entre los cuales uno de ellos es el que mejor se adapta al conjunto de datos, mientras que los restantes representarán agrupaciones más divididas o más conglomeradas, proporcionando desde una división total -cada punto es un *cluster*- hasta una agrupación total -todos los puntos constituyen un *cluster*-.

En la *Figura 8* se puede observar el resultado de la agrupación de varios conjuntos de datos a través de este algoritmo.



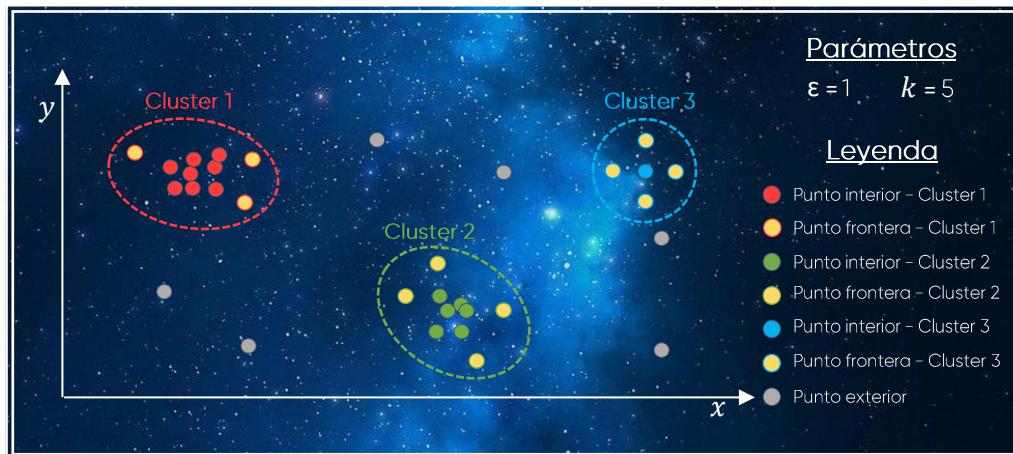
*Figura 8. Resultados de agrupaciones por clusterización jerárquica [17]*

La clusterización jerárquica es capaz de agrupar satisfactoriamente conjuntos de datos más complejos que los que es capaz de agrupar K-Medias, como es el caso de *clusters* que se sitúan dentro de otros -agrupación tercera de la *Figura 8*- Sin embargo, este algoritmo es más sensible a los valores atípicos y es más exigente computacionalmente.

## Density Based Scan Clustering -DBSCAN-

Utiliza un método más intuitivo, pero requiere de una parametrización más ajustada. DBSCAN requiere de dos parámetros: la distancia máxima para la que dos puntos pertenecen a un *cluster* - $\epsilon$ - y el número mínimo de puntos que pueden formar un *cluster* - $k$ -. Este algoritmo puede considerar los puntos de datos de tres formas: interior, frontera y exterior -o ruido-.

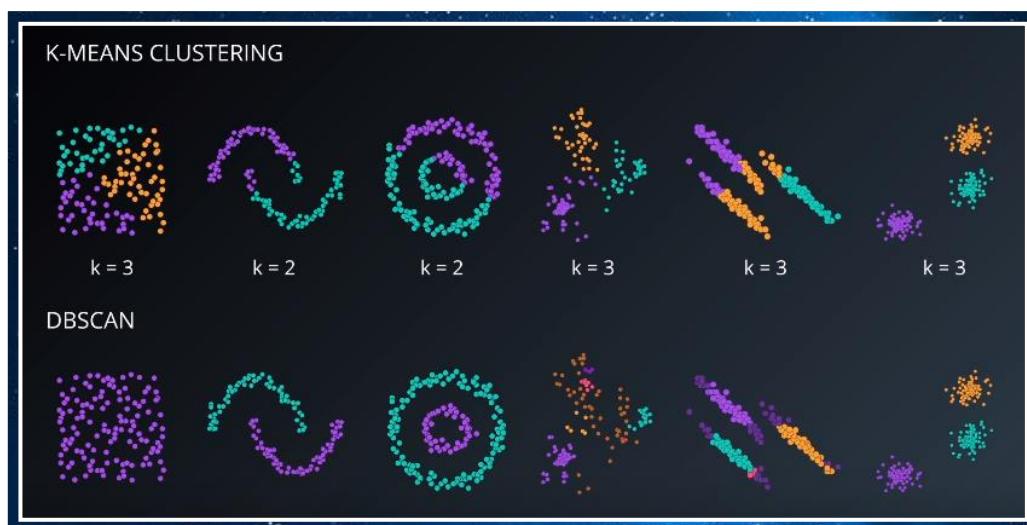
El punto interior es aquel que, en un radio de longitud  $\epsilon$ , tiene un número de puntos mayor que  $k$ . Por su parte, el punto frontera tiene, para el mismo radio, un número de puntos menor que  $k$ , siendo alguno de ellos un punto interior. Y el punto exterior es aquel que no tiene ningún punto interior en un radio  $\epsilon$ . En la *Figura 9* se muestra un ejemplo de todos estos puntos representados en distintas agrupaciones o *clusters*.



*Figura 9.* Representación del resultado de la agrupación por DBSCAN

DBSCAN es un algoritmo que ofrece una gran flexibilidad en cuanto a formas y tamaños de las agrupaciones que puede generar mediante la distinta parametrización de  $\epsilon$  y  $k$ , lo que permite también identificar y trabajar con datos atípicos.

En la *Figura 10* se puede observar el resultado de la clusterización mediante DBSCAN en varios conjuntos de datos.



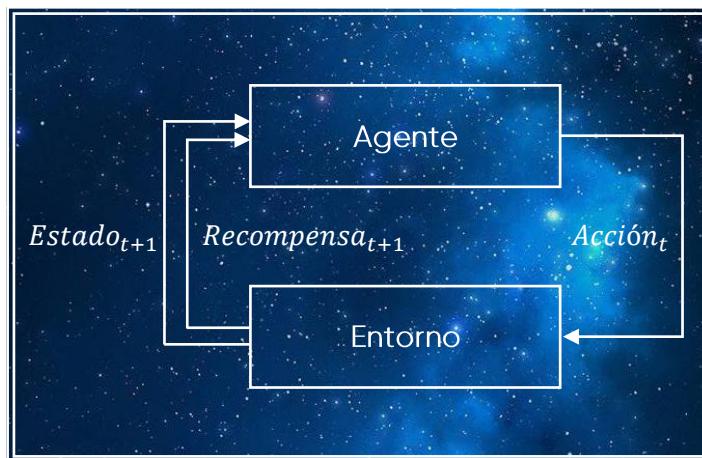
*Figura 10.* Resultados de agrupaciones por DBSCAN [18]

## Aprendizaje por refuerzo

El ajedrez es un juego que, pese a tener unas reglas específicas y sencillas, resulta manifiestamente complejo debido a su inabarcable número de posibles movimientos -número de Shannon-, que se estima entre  $10^{120}$  y  $10^{123}$  [11] -como punto de referencia, el número de átomos en el universo se calcula comúnmente en  $10^{80}$ -. Un escenario que ofrece tal cantidad de estados se presenta difícilmente propenso a ser abordado mediante algoritmos.

El objetivo de los modelos que aprenden por refuerzo es avanzar hacia un estado más favorable para la consecución de un estado deseado. En el caso del ajedrez, este estado es el jaque mate al rival. Para lograrlo, no existen algoritmos que puedan determinar una serie de acciones que resulten en dicho objetivo, sino que, debido a la complejidad del entorno, es necesario construir y seguir una política que definan una serie de acciones correctas que ayuden a conseguir el objetivo. La diferencia entre una política y un algoritmo es que en la primera no existe una mejor opción en un determinado estado intermedio; mientras que un algoritmo, por definición, establece cuál es la mejor acción en cada momento. En el aprendizaje por refuerzo, una acción correcta es la que pertenece a una política alineada con el objetivo final, en cuyo caso el modelo debería tener la capacidad de identificar dicha alineación de las políticas y aprender de secuencias de acciones pasadas para generarlas mediante recompensas.

Otro ejemplo visualizable es la conducción autónoma. En este caso, el objetivo final es alcanzar un lugar deseado. En cada estado intermedio, el sistema o agente puede realizar un gran número de movimientos, de los cuales solo serán correctos aquellos que pertenezcan a una política de acciones que conlleven la llegada del sistema al lugar objetivo de la forma más rápida posible y evitando obstáculos. Para determinar qué políticas están alineadas con esa meta, el sistema debe realizar múltiples pruebas para prender qué acciones participan en la consecución del objetivo y cuáles no [3].



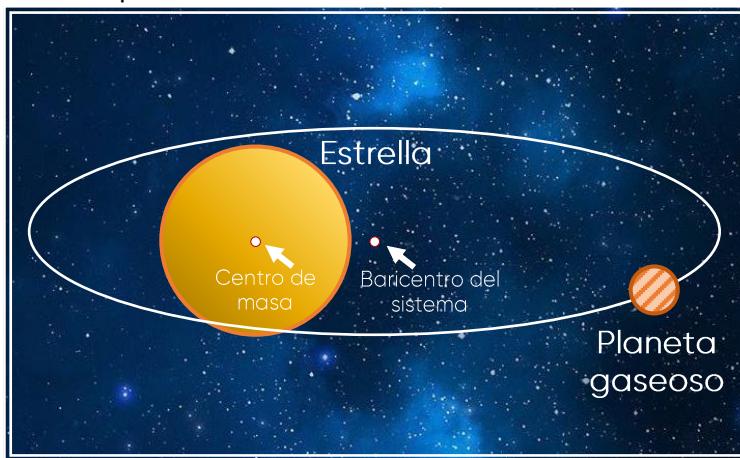
*Figura 11. Diagrama del aprendizaje por refuerzo*

En la *Figura 11* se muestra un diagrama simplificado del aprendizaje por refuerzo. El agente realiza una acción en el instante  $t$  y el entorno devuelve una recompensa y un nuevo estado en el instante  $t + 1$ .

## Detección de exoplanetas

En la Vía Láctea se estima que pueden existir más de 200 mil millones de estrellas, de las cuales miles de millones podrían conformar, cada una, un sistema con uno o varios planetas orbitando a su alrededor. Estos planetas reciben el nombre de **exoplanetas**. Hasta la fecha, se han confirmado 4171 [1] -07/07/2020- de estos mundos a decenas de años luz de distancia.

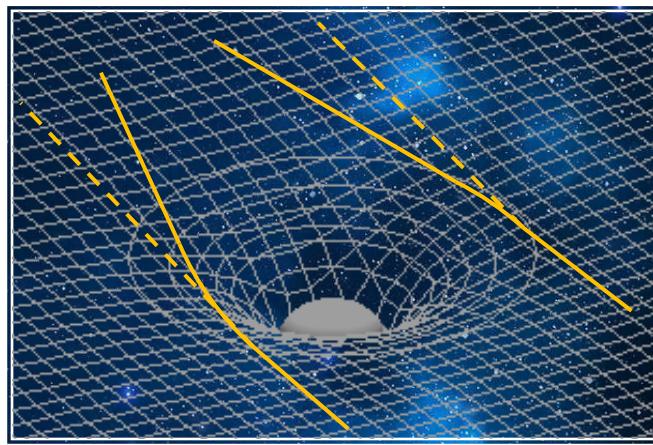
Detectar estos sistemas resulta una tarea ardua y compleja dadas las magnitudes -distancia a estos sistemas, relación de tamaños y luminosidades entre las estrellas y los planetas, etc.-. Para llevar a cabo estos descubrimientos, existen diferentes métodos de detección. El método de **velocidades radiales** consiste en observar las estrellas en busca de movimientos tambaleantes, es decir, que sus centros de masa estén girando alrededor de un baricentro provocado por la presencia de otro cuerpo masivo cercano. Como se puede observar en la *Figura 12*, el movimiento de las estrellas es solo detectable si los cuerpos que las orbitan son lo suficientemente masivos como para alejar el baricentro del sistema lo suficiente como para que el movimiento tambaleante de la estrella sea perceptible, y los únicos cuerpos capaces de provocar esto son grandes planetas gaseosos, pues otros planetas similares a La Tierra originan oscilaciones insuficientes para ser detectados.



*Figura 12. Desviación del centro de masa de una estrella respecto al baricentro*

Otro método comúnmente utilizado es el método de **microlente**, que consiste en aprovechar los efectos de la relatividad general para detectar ciertas variaciones. Según la teoría de Albert Einstein, la presencia de materia produce una curvatura del espacio-tiempo adyacente. Grandes cúmulos de materia, como es el caso de los planetas, producen una curvatura tal que es capaz de provocar una desviación de la luz que atraviesa ese espacio lo suficiente como para poder ser detectada y medida -estas desviaciones son del orden de unos pocos microsegundos-. Por ello, un cuerpo suficientemente masivo es capaz de proyectar toda la luz proveniente de detrás de éste en un único punto, aumentando así la luz percibida por el observador en ese punto y generando un pico o curva de luz que es inversa a la producida en el método de tránsito, explicado más adelante. Este método de detección se centra en observar el espacio en busca de estos picos de luz y deducir la existencia de exoplanetas mediante sus mediciones.

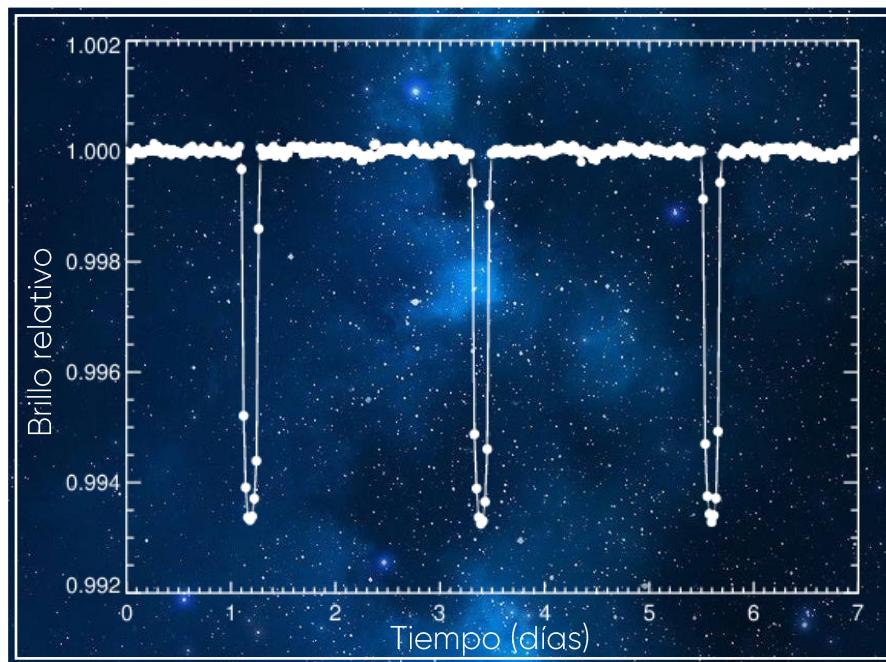
En la *Figura 13* se representa la curvatura del espacio-tiempo provocada por un cuerpo masivo. Esta curvatura desvía la luz que la atraviesa ya que, como la luz viaja por el espacio-tiempo, ésta se ve alterada por sus deformaciones. Toda esta luz se concentra en un solo punto, por lo que el cuerpo masivo actúa como una lente gravitacional.



*Figura 13. Desviación de la luz provocada por la curvatura del espacio-tiempo*

## Método de tránsito

Para detectar planetas más pequeños como La Tierra se utiliza el método de **tránsito**. Cuando se observa una estrella y un cuerpo transita por delante de ésta, la intensidad de la luz recibida disminuye hasta que el cuerpo termina de transitar. Esto genera una ligera variación en la intensidad de la luz recibida que puede ser detectada y medida, pese a ser tan sutil como una mosca pasando por delante de los faros de un vehículo a cientos de kilómetros. La observación de exoplanetas mediante el método de tránsito produce una **curva de luz** resultante del brillo relativo percibido, cuya variación se mueve en un margen del entorno del 1%, lo que demuestra la dificultad para observarlas y la necesidad de precisión en las medidas.



*Figura 14. Curvas de luz producidas por el exoplaneta HAT-P-7b orbitando la estrella HAT-P-7*

En la *Figura 14* se pueden observar varias curvas de luz producidas por el exoplaneta HAT-P-7b al transitar por delante de la estrella HAT-P-7. En el momento del tránsito, la luz percibida procedente de la estrella disminuye.

Estas mediciones poseen una serie de características que arrojan gran información acerca de lo que se está observando. La distancia temporal entre las curvas de luz producidas por el tránsito del mismo cuerpo celeste representa el período orbital de éste, o lo que es lo mismo, la duración de un año. Por lo general, basta con tres curvas de luz detectadas para confirmar la existencia de un exoplaneta. Justo en la mitad de la distancia entre curva y curva, se aprecia otra disminución más pequeña de la intensidad de la luz recibida, que muestra el momento en el que el cuerpo transita por detrás de la estrella y, así, dejando de recibir la luz que refleja dicho cuerpo.

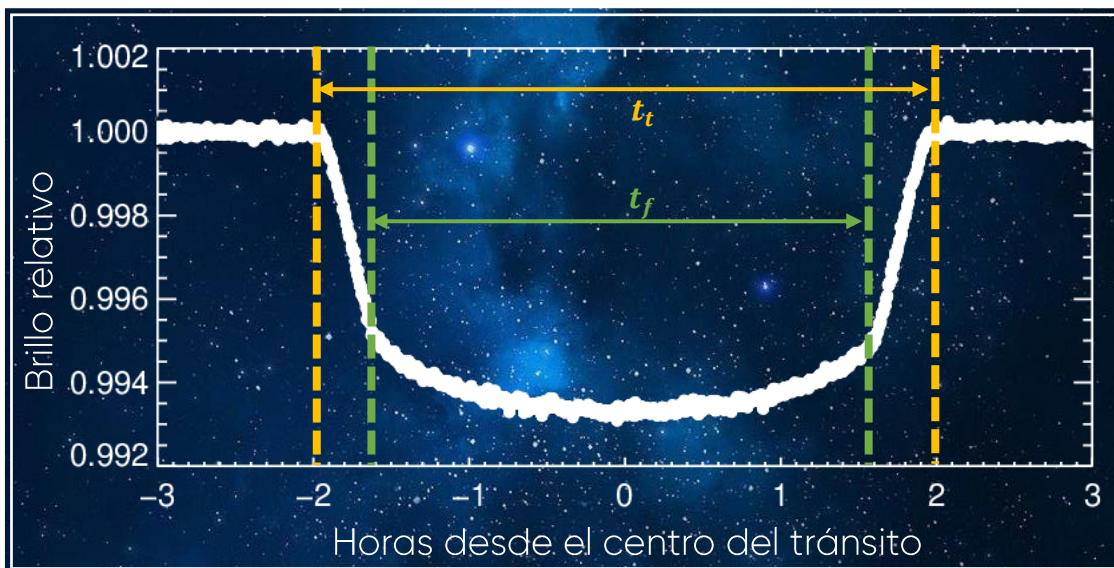
Otra característica del cuerpo celeste que está transitando la estrella que se puede abstraer de este método es su tamaño. Esto se puede calcular gracias a conocer la profundidad de la curva de luz,

$$R_c = R_\star \sqrt{P}$$

donde  $R_c$  es el radio del cuerpo celeste en tránsito;  $R_\star$ , el radio de la estrella siendo orbitada; y  $P$ , la profundidad de la curva de luz en términos relativos.

Analizando la curva de luz con más detalle, se puede observar que ésta no es del todo plana por debajo. Esto se debe a que la luz de una estrella no presenta una distribución homogénea a lo largo de toda el área observada, sino que es más brillante en el centro desde nuestra perspectiva -ya que es donde más materia acumula- y menos brillante cuanto más cercano a sus límites, por lo que cuando el cuerpo celeste transita por delante, eclipsará más intensidad de luz cuando pasa por el centro y menos cuando se aleja de éste.

Otra característica de la curva de luz es que no se produce una disminución del brillo de forma instantánea, es decir, que existe un espacio de tiempo entre el momento en el que empieza a decaer el brillo y el momento en el que se estabiliza la curva -Figura 14, las franjas entre  $t_t$  y  $t_f$ . Este espacio de tiempo es el que tarda el cuerpo en cruzar el borde de la estrella hasta posicionarse por delante de ésta en su totalidad. Gracias a esto, y combinándolo con la profundidad de la curva, se puede estimar la distancia entre la estrella y el cuerpo que la transita.



**Figura 15.** Detalle de una curva de luz producida por el exoplaneta HAT-P-7b orbitando la estrella HAT-P-7:  $t_t$  se refiere al tiempo total de tránsito; y  $t_f$ , al tiempo en el que toda el área del cuerpo transita la estrella

En la Figura 15 se muestra el detalle de una curva de luz donde se pueden observar las características anteriormente mencionadas.

No solo la curva de luz producida por un exoplaneta aporta información acerca de éste, sino que también se puede medir el espectro de luz recibido en el momento del tránsito para deducir la composición de su atmósfera de forma precisa. Con todas estas técnicas, se puede conocer una gran cantidad de información sobre estos inimaginablemente lejanos mundos solamente observando la luz procedente de los mismos.

Para estudiar las estrellas en busca de estos datos, la NASA lleva a cabo varias misiones utilizando telescopios orbitales como *Kepler*, *k2* y la actualmente activa *TESS*, diseñadas para este propósito. Estos satélites realizan campañas, que consisten en observar una porción de 100 grados cuadrados durante 80 días a lo largo de 20 puntos distintos del firmamento, logrando una resolución de 4 arcsec/píxel o, lo que es lo mismo, 0.25 grados/píxel. La luz medida en la misión *k2* es en su mayoría luz visible, moviéndose en un rango de entre 4200Å -ángstroms- y 8800Å [12] -la luz visible va desde los 4000Å hasta los 7000Å-. Estas misiones arrojan una gran cantidad de datos derivados de la observación de objetivos que son propuestos abiertamente por la comunidad.

## Datos

Los datos que van a ser analizados en el presente proyecto fueron obtenidos en la Campaña 3 de la misión *K2* de la NASA en 2016. Estos datos se publican abiertamente por la agencia espacial norteamericana en formato *.fits* a través del portal *Mikulski Archive del Space Telescope Science Institute* [13], y han recibido un primer tratamiento para un mejor estudio.

Primeramente, la NASA lleva a cabo un proceso para eliminar el ruido en los datos producidos por el propio telescopio. A continuación, por cada archivo correspondiente a cada estrella, se ha traspasado cada una de las distintas mediciones de luz a una columna distinta llamada ***FLUX.n***, siendo *n* el número de la medición. Por último, se han etiquetado en la columna ***LABEL*** con un 2 aquellas estrellas -filas- cuyas mediciones han confirmado la existencia de algún exoplaneta en su órbita, y con un 1 aquellas que no han sido confirmadas. Al momento de realizar este proceso, los datos dejaban entrever que apenas existían estrellas con exoplanetas, por lo que se le han añadido algunos datos de estrellas otras campañas que sí han sido confirmadas con exoplanetas. Por último, se han dividido los datos en dos *datasets*, uno para el entrenamiento del modelo y el otro para test.

### Entrenamiento

	LABEL	FLUX.1	FLUX.2	FLUX.3	...	FLUX.3196	FLUX.3197
0	2	93.85	83.81	20.10	...	5.08	-39.54
1	2	-38.88	-33.83	-58.54	...	16.00	19.93
2	2	532.64	535.92	513.73	...	-70.02	-96.67
...	...	...	...	...	...	...	...
5085	1	3.82	2.09	-3.29	...	-6.41	-2.55
5086	1	323.28	306.36	293.16	...	-14.09	27.82

5087 filas x 3198 columnas

Tabla 1. Muestra de los datos de entrenamiento

En la *Tabla 1* se muestra la estructura de los datos de **entrenamiento**, que se dividen en:

- **5087 filas**, representando cada una de las estrellas, de las cuales:
  - 37 tienen al menos un exoplaneta en su órbita -*LABEL* = 2-.
  - 5050 no tienen ningún exoplaneta en su órbita -*LABEL* = 1-.
- **3198 columnas**, conformadas por:
  - 1 columna -*LABEL*- para etiquetar las estrellas.
  - 3197 columnas -***FLUX.1 – FLUX.3197***-, que son las distintas mediciones del brillo de las estrellas, cada una tomada cada 36 minutos aproximadamente o  $0,00046 \text{ Hz}$  -se realizan 3.197 observaciones en 80 días:  $\frac{80}{3197} \approx 0.025 \rightarrow 0.025 \times 24 \times 60 = 36 \text{ minutos} \rightarrow \frac{1}{36 \times 60} \approx 0.00046 \text{ Hz}$ -, pues la unidad de tiempo utilizada para realizar las mediciones es el Tiempo Dinámico Baricéntrico -TDB-. Por su parte, la unidad de medida de estas columnas es fotones por segundo - $\text{e}^{-\text{s}^{-1}}$ -, que mide la radiancia espectral.

### Test

	LABEL	FLUX.1	FLUX.2	FLUX.3	...	FLUX.3196	FLUX.3197
0	2	119.88	100.21	86.46	...	269.43	57.72
1	2	5736.59	5699.98	5717.16	...	-2294.86	-2034.72
2	2	844.48	817.49	770.07	...	-36.79	30.63
...	...	...	...	...	...	...	...
568	1	91.36	85.60	48.81	...	-6.48	17.60
569	1	3071.19	2782.53	2325.47	...	-69.63	121.56

570 filas x 3198 columnas

*Tabla 2. Muestra de los datos de test*

En la *Tabla 2* se muestra la estructura de los datos de **test**, que se dividen en:

- **570 filas**, representando cada una de las estrellas, de las cuales:
  - 5 tienen al menos un exoplaneta en su órbita -**LABEL = 2**-.
  - 565 no tienen ningún exoplaneta en su órbita -**LABEL = 1**-.
- **3198 columnas**, conformadas por:
  - 1 columna -**LABEL**- para etiquetar las estrellas.
  - 3197 columnas -**FLUX.1 – FLUX.3197**-, que son las distintas mediciones del brillo de las estrellas, cada una tomada cada 36 minutos aproximadamente o 0.00046 Hz, pues la unidad de tiempo utilizada para realizar las mediciones es el Tiempo Dinámico Baricéntrico -TDB-. Por su parte, la unidad de medida de estas columnas es fotones por segundo -e·s<sup>-1</sup>-, que mide la radiancia espectral.

### Problemas

Tanto los datos de entrenamiento como los datos de **test** presentan dos problemas. El primero de ellos es el **elevado número de dimensiones** o columnas, representando 3197 mediciones por cada estrella, muchas de las cuales son muy parecidas entre sí y, por tanto, no aportan información útil para el estudio. La gran mayoría de las columnas son mediciones de la intensidad de luz en el momento en el que no se producen importantes variaciones de la misma, por lo que será necesario aplicar técnicas de reducción de dimensiones.

El segundo problema es que ambos *datasets* están **desbalanceados**. El número de estrellas que tienen al menos un exoplaneta confirmado en su órbita con respecto al número total de estrellas en el conjunto de datos de entrenamiento representa el **0.73%** -37 de un total de 5087-. El problema que esto supone a la hora de realizar el entrenamiento es que el modelo no dispondrá de muchos datos relevantes para una posterior clasificación acertada.

### Soluciones

Para abordar ambos problemas, se van a seguir métodos tanto de reducción de dimensiones -**Principal Component Analysis (PCA)**- para descartar las dimensiones de menos relevantes para el modelo, y de balanceamiento de datos -**data augmentation**- para mejorar el entrenamiento del futuro modelo evitando la falta de registros de estrellas con exoplanetas confirmados.

## Principal Component Analysis -PCA-

Este método de reducción de dimensiones es un procedimiento estadístico que emplea una transformación ortogonal que convierte un grupo de dimensiones correlacionadas en un grupo de dimensiones no correlacionadas. De esta forma, se puede **obviar** un gran número de componentes del *dataset* sin perder información relevante.

Principalmente, se generan dos vectores ortogonales -PC1 y PC2- que cruzan el centroide del conjunto de datos para, posteriormente, recoordinar dichos datos en función de estos nuevos vectores. En este nuevo sistema de referencia, la distancia relativa entre los datos se mantiene igual, pero su interpretación puede ser diferente ya que la varianza puede cambiar con respecto a los ejes originales, siendo en algunos casos menor y, por lo tanto, haciendo descartables ciertas dimensiones que se muestran irrelevantes. En otras palabras, *PCA* produce combinaciones lineales de las dimensiones originales para generar nuevos ejes y reinterpretar los datos [14].

Para ello, se calcula la matriz de covarianza de las dimensiones, se obtienen los autovectores y autovalores de dicha matriz, se seleccionan los  $n$  autovectores que se correspondan con los  $n$  autovalores más grandes -siendo  $n$  el posterior número de dimensiones resultantes, que será menor o igual al número de dimensiones inicial-, se construye la matriz de proyección con los  $n$  autovectores seleccionados y se transforma el *dataset* original, a través de dicha matriz, en un nuevo *dataset* con  $n$  dimensiones.

Con este método, el número de columnas del *dataset* se reduce considerablemente, haciendo que su procesamiento resulte menos complejo y más rápido sin perder una gran cantidad de información relevante, ya que las dimensiones seleccionadas son aquellas con una mayor varianza.

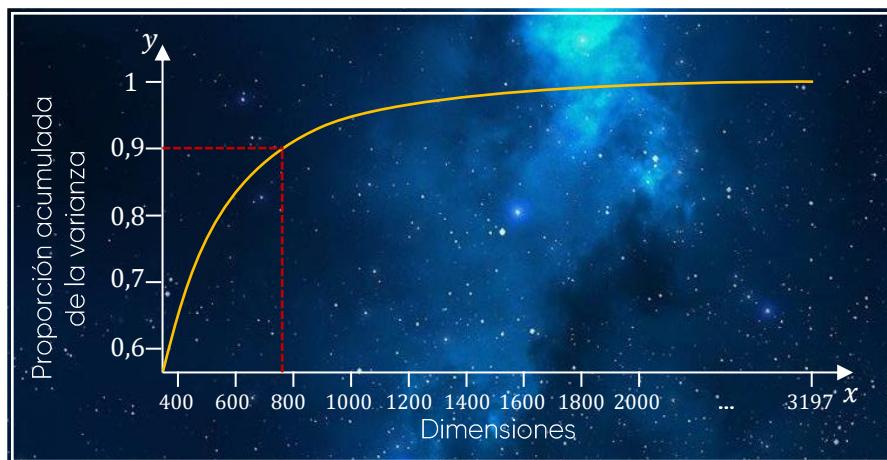


Figura 16. Resultado de la aplicación del método PCA

En el ejemplo de la *Figura 16* -datos no reales- se puede visualizar que, para conseguir una varianza del 90%, solo es necesaria una fracción del número total de dimensiones para llevar a cabo el estudio de forma eficaz. Es decir, todas las demás dimensiones no aportan información relevante acerca de los datos, por lo que se puede descartar.

## Data augmentation

Existen diversas técnicas para aplicar balancear un conjunto de datos que se pueden clasificar en dos grupos dependiendo de las técnicas que se empleen. Por un lado, cuando se presenta un desbalanceamiento en un *dataset*, se pueden reducir aquellos registros de una clase que se repiten de forma excesiva. Esto es conocido como ***undersampling***. Una de las técnicas catalogadas en este grupo es el *undersampling* aleatorio, que consiste eliminar registros de la clase mayoritaria de forma aleatoria, pudiendo sustituirlos y no por registros de otras clases. Este método logra balancear el *dataset*, pero puede llegar a eliminar registros que sí son relevantes.

También se pueden utilizar algoritmos de *clustering* de tal forma que se sustituye una agrupación de registros por el centroide de una agrupación por K-medias, estableciendo el número de *clusters* según el nivel de *undersampling*.

Existen estudios recientes en los que se investiga más a fondo distintas técnicas de *undersampling*, combinándolas incluso con aprendizaje de conjuntos para la obtención de mejores resultados [15].

Por otro lado, para compensar el desbalanceamiento se puede optar por el ***oversampling***. Estas técnicas consisten en **replicar**, con alguna variación, los tipos de datos que son interesantes para entrenar el modelo. En el caso de un modelo de procesado de imágenes - que suele ser donde más se aplica esta técnica-, se pueden replicar aquellas que se muestran relevantes para el entrenamiento mediante el volteo de las imágenes originales, el cambio de ancho y alto de la imagen, alteración de colores, aplicación de filtros, recortes, etc., de forma que el número de estas imágenes se multiplica sin ser éstas iguales entre ellas y, por tanto, mejorando el entrenamiento del modelo.

Una de las técnicas de oversampling más utilizadas es la *Synthetic Minority Over-sampling Technique -SMOTE-*. Este método consiste en seleccionar un registro de la clase minoritaria que se quiera aumentar, calcular los  $k$  vecinos en el espacio dimensional, tomar un vector entre el registro original y uno de los  $k$  vecinos y multiplicar dicho vector por un número aleatorio entre 0 y 1. Como resultado, se obtiene un nuevo registro creado artificialmente.

También, como en el caso del *undersampling*, existe una técnica en la que se seleccionan aleatoriamente unos registros de la clase minoritaria y se replican tantas veces como se considere. El problema de generar registros exactamente iguales es que el modelo de clasificación que sea entrenado con dichos datos puede resultar sesgado.

En el *dataset* de entrenamiento estudiado en el presente proyecto, que se muestra muy desbalanceado -37 estrellas con exoplanetas frente a 5050 sin exoplanetas-, el objetivo es replicar los datos de estrellas con exoplanetas de forma que se introduzcan alteraciones menores sin que lleguen a ser excesivamente diferentes de los originales.

Esto resultaría en un aumento de los datos de estrellas con la columna *LABEL* = 2 - confirmación de existencia de exoplanetas-, lo que balancearía los datos e implicaría un mejor entrenamiento del modelo.

Para llevar a cabo esta técnica, es necesario un estudio previo de los datos que se van a replicar para evitar que las alteraciones en los nuevos datos generados no los hagan completamente diferentes, ya que el entrenamiento del modelo mostraría un resultado muy desajustado con respecto a los datos reales.

## Modelado de datos

El modelo consistirá en una red neuronal artificial de tipo perceptrón multicapa construido en Python que será entrenado con los datos -ya tratados- de entrenamiento y probado con los datos de test.

La solución más óptima consiste en, primeramente, realizar un **tratamiento** inicial de los datos. A continuación, aplicar el método de **PCA** con el objetivo de reducir el número de dimensiones del *dataset* de entrenamiento para, posteriormente, realizar el balanceamiento de datos mediante el método de ***data augmentation***. De esta forma, la generación de nuevos datos para el modelo resultará más sencilla al haber eliminado previamente las dimensiones poco relevantes para el modelo.

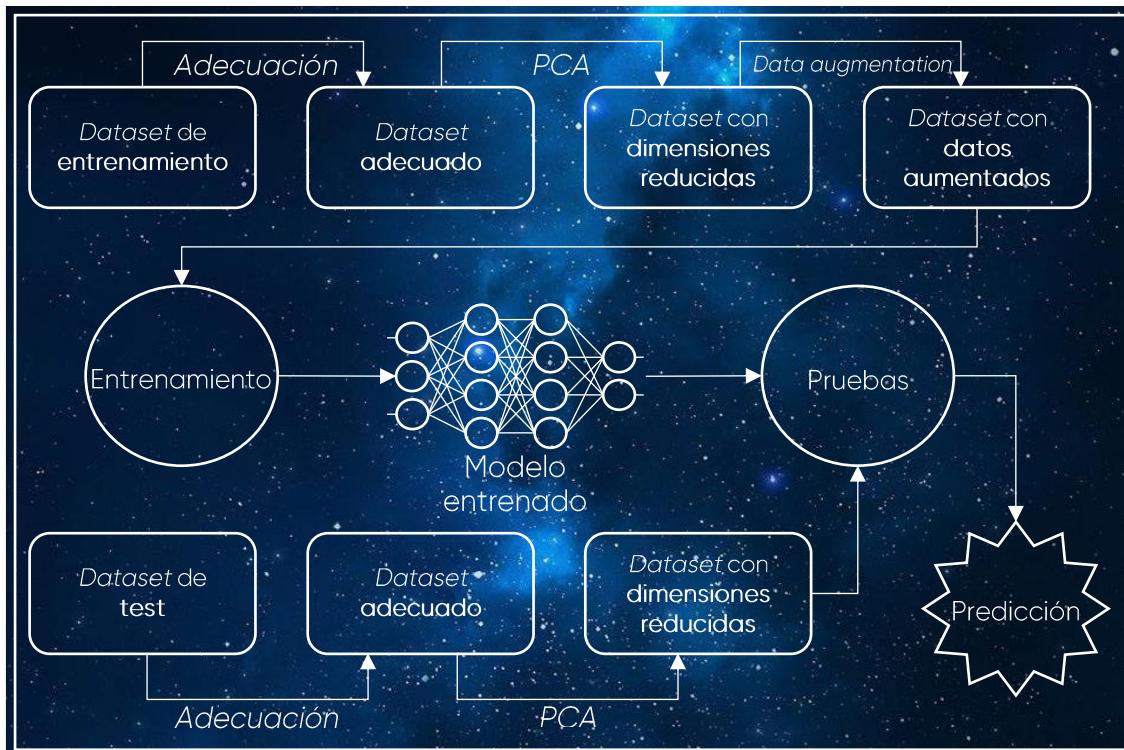


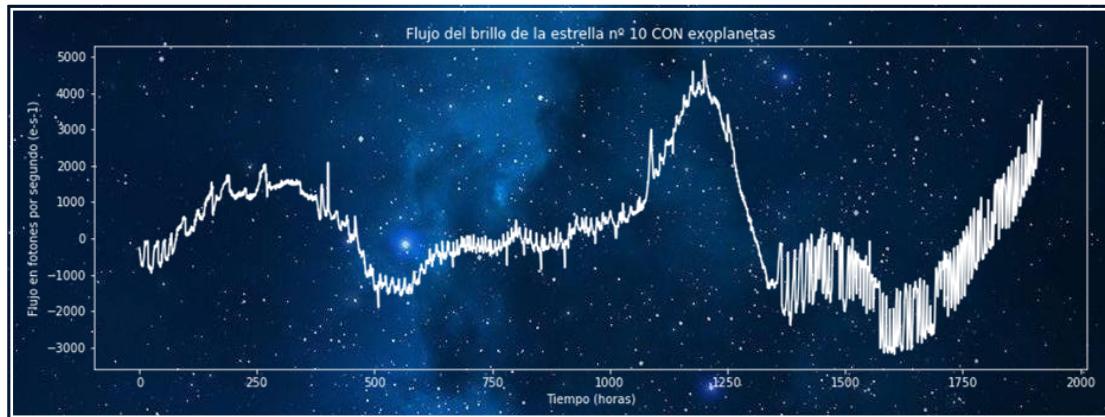
Figura 17. Croquis del proceso de construcción del modelo

En cuanto a los datos de test, es necesario realizar el primer tratamiento de datos y la reducción de dimensiones en el *dataset* de entrenamiento, ya que el modelo estará entrenado con la forma y las dimensiones específicas del entrenamiento. En los datos de test resulta irrelevante aplicar un balanceamiento de datos ya que esto solo afecta al entrenamiento del modelo, que puede verse afectado por dicho desbalanceamiento. En los datos de test esto carece de relevancia. En la *Figura 17* se muestra, de forma gráfica, el proceso completo partiendo de los datos en crudo y finalizando con una predicción.

## Adecuación de los datos

Cada estrella posee una forma única de brillar. Su rotación, superficie, composición, tamaño, etc. hace que la luz de cada estrella varíe de forma heterogénea, y esa variación es captada y reflejada en los datos. Esto se une a las variaciones de luz que puedan derivar de exoplanetas orbitando y transitando delante de ellas, siendo estas variaciones las que sí son relevantes para su detección.

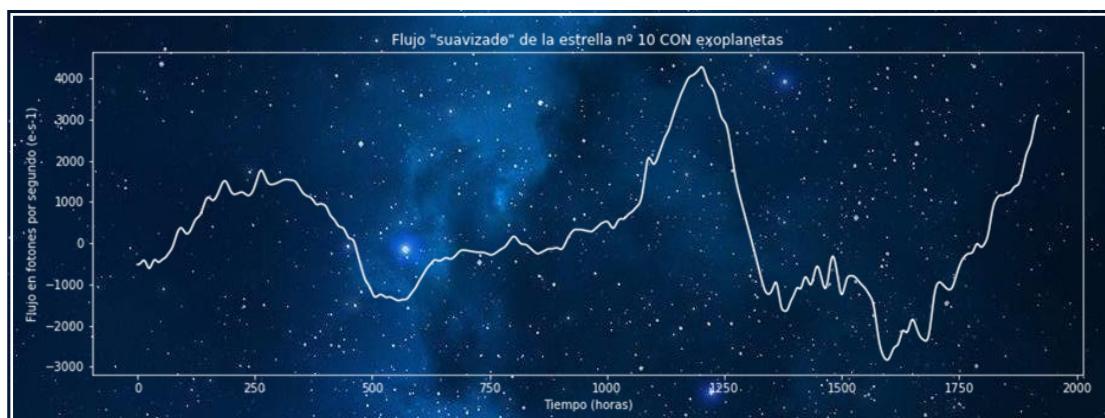
Al entrenar un modelo sin tener esto en cuenta, éste va a resultar sesgado por las estrellas y sus fluctuaciones con las que ha sido entrenado, por lo que es necesario abstraer del flujo original las variaciones de luz resultantes de posibles exoplanetas, eliminando toda posible interferencia resultante de las fluctuaciones particulares de las estrellas.



*Figura 18. Flujo de la estrella nº 10 CON exoplanetas sin tratamiento*

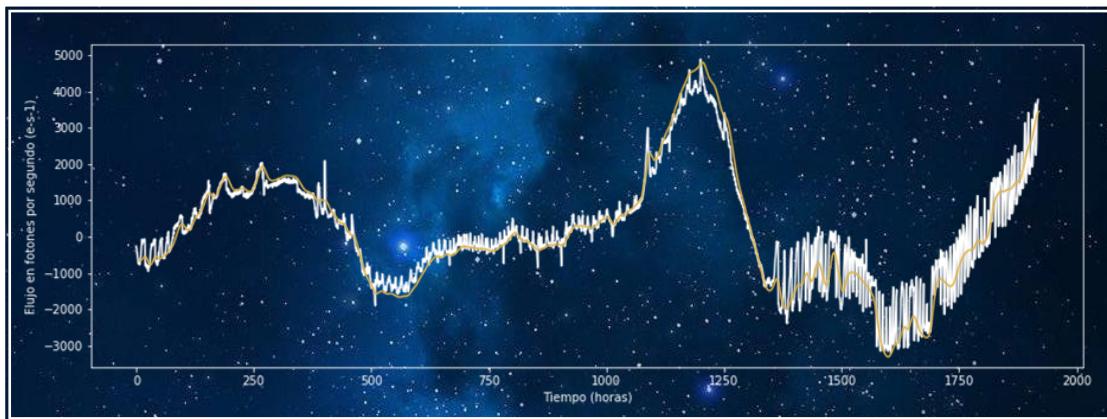
En la *Figura 18*, se muestra el flujo de la estrella nº 10 con exoplanetas confirmados. Como se puede apreciar, las variaciones de luz se muestran muy heterogéneas y presentan grandes fluctuaciones a lo largo del flujo.

Para filtrar los datos eliminando las fluctuaciones derivadas de la propia estrella, es necesario encontrar el flujo general de variaciones de intensidad provocada por la estrella. Para ello, se va a aplicar el método de desenfoque Gaussiano con el objetivo de “suavizar” los datos. El resultado de este proceso se muestra en la *Figura 19*.



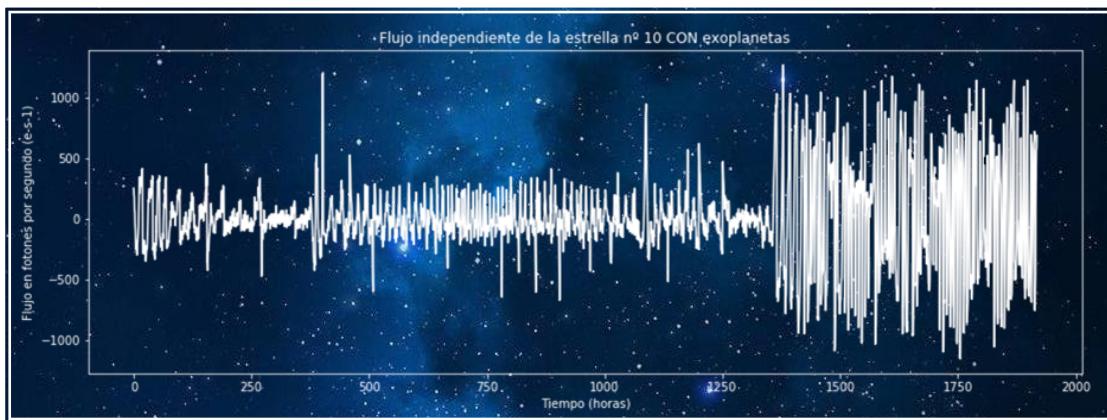
*Figura 19. Flujo “suavizado” de la estrella nº 10 CON exoplanetas*

A continuación, se va a eliminar el flujo “suavizado” del flujo en bruto para abstraer de esta forma las variaciones que sí son interesantes para detectar posibles exoplanetas.



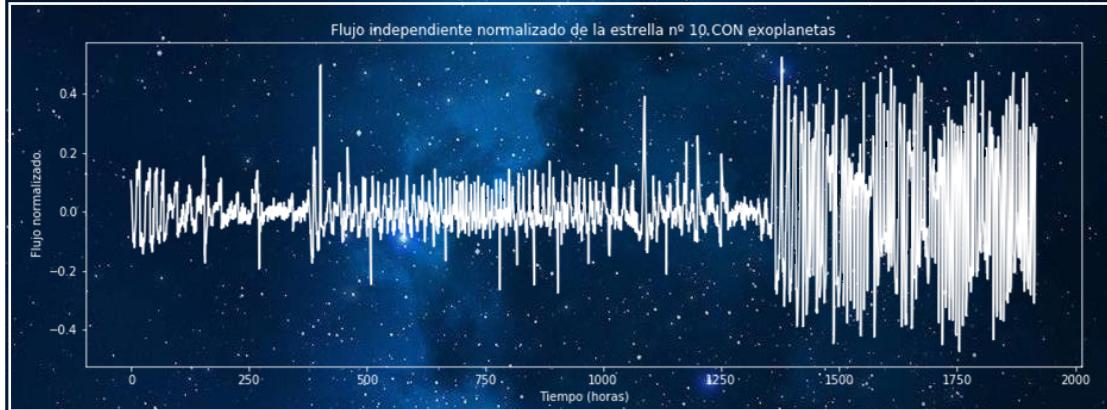
*Figura 20. Superposición de los anteriores flujos*

Para abstraer las variaciones de luz relevantes para el entrenamiento, es necesario eliminar las fluctuaciones de las propias estrellas de los datos. Esto se consigue eliminando la tendencia principal de los datos -que viene dada por dichas fluctuaciones y representada en la Figura 20-, resultando en las variaciones de luz independientes de las fluctuaciones de la estrella -Figura 21-.



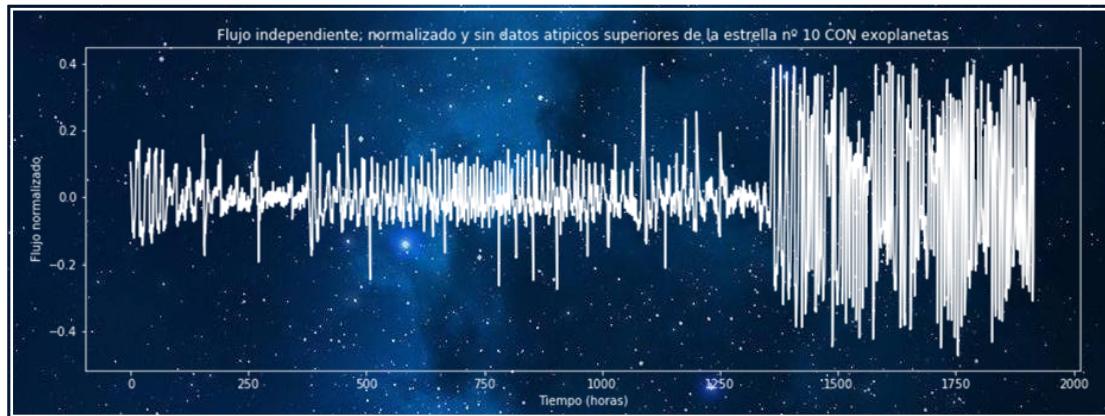
*Figura 21. Flujo desvinculado de la tendencia de fluctuación de la estrella nº 10 CON exoplanetas*

Una vez conseguido esto último, el siguiente paso es normalizar los datos. El resultado de este proceso se muestra en la Figura 22.



*Figura 22. Flujo desvinculado de la tendencia de fluctuación normalizado*

Los exoplanetas provocan una disminución de la luz percibida de la estrella al transitar delante de ella, por lo que es necesario eliminar todos los datos atípicos superiores, ya que son los inferiores los que son relevantes. El resultado de esto se muestra en la *Figura 23*.



*Figura 23. Flujo desvinculado de la tendencia de fluctuación normalizado y sin datos atípicos superiores*

Este proceso de tratamiento de datos se aplica a todo el *dataset* tanto de entrenamiento como de test. En la *Tabla 3* se muestra un ejemplo de los datos de entrenamiento después de realizar este proceso, mientras que en la *Tabla 4* se muestran los datos de test. Se puede apreciar que, ahora, todos los datos se encuentran en la misma magnitud.

## Entrenamiento

	LABEL	FLUX.1	FLUX.2	FLUX.3	...	FLUX.3196	FLUX.3197
0	2	0.1064	0.1	0.0581	...	0.0026	-0.027
1	2	0.0414	0.0507	0.0035	...	0.0398	0.0471
2	2	0.0363	0.0392	0.0188	...	-0.032	-0.057
...	...	...	...	...	...	...	...
5085	1	0.0072	0.004	-0.006	...	-0.002	0.005
5086	1	0.088	0.0759	0.0672	...	-0.014	0.0167

5087 filas x 3198 columnas

*Tabla 3. Muestra de los datos de entrenamiento tras el primer tratamiento*

## Test

	LABEL	FLUX.1	FLUX.2	FLUX.3	...	FLUX.3196	FLUX.3197
0	2	0.1128	0.1937	0.1566	...	-0.035	0.0329
1	2	0.04	0.0381	0.0412	...	-0.088	-0.068
2	2	0.4118	0.346	0.291	...	0.0085	0.0899
...	...	...	...	...	...	...	...
568	1	0.012	0.0109	0.0036	...	-0.001	0.0035
569	1	0.0434	0.0362	0.049	...	-0.018	-0.011

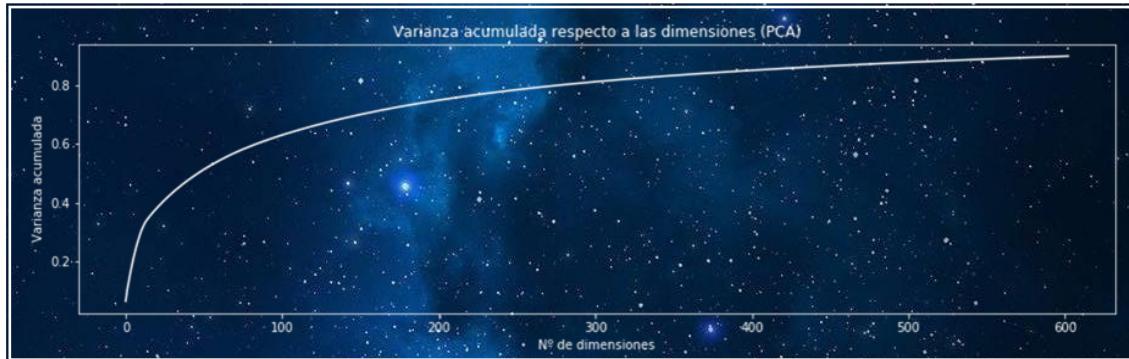
570 filas x 3198 columnas

*Tabla 4. Muestra de los datos de test tras el primer tratamiento*

## Reducción de dimensiones

Una vez se ha realizado el primer tratamiento de datos, el siguiente paso es aplicar la técnica de reducción de dimensiones **PCA**.

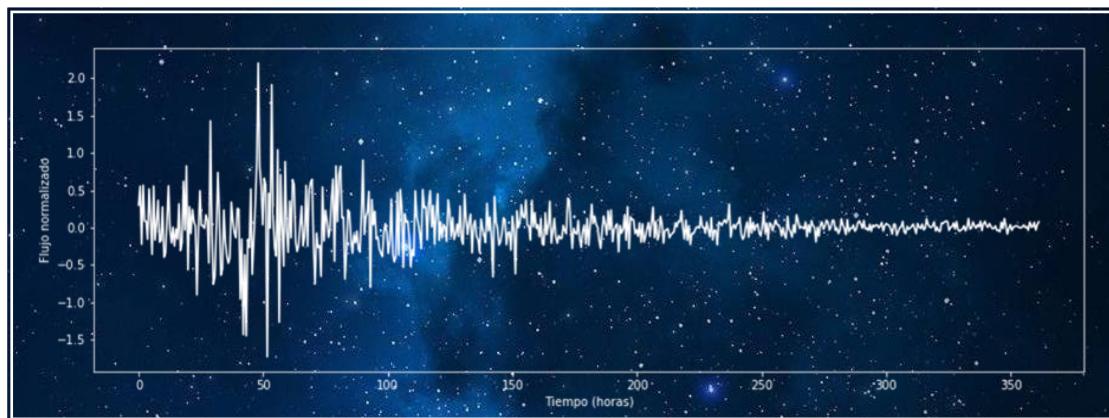
Para ello, se va a utilizar un cálculo de dimensiones tal que la varianza acumulada sea de 0.9. De esta forma, se asegura una varianza alta y un número de dimensiones aceptable para entrenar, posteriormente, el modelo.



*Figura 24. Varianza acumulada respecto a las dimensiones del dataset de entrenamiento*

En la *Figura 24* se muestra la curva de la varianza acumulada con respecto a las dimensiones del *dataset* de entrenamiento. Como se puede observar, para lograr una varianza de 0.9 se necesitan un total de 603 dimensiones. Estas dimensiones, que se encuentran ordenadas de mayor a menor varianza -es decir, por relevancia para el modelo-, serán las que se usarán en los siguientes pasos del proceso de modelado.

En cuanto al *dataset* de test, es importante utilizar la misma función de *PCA* resultante de los datos de entrenamiento para seleccionar exactamente las mismas dimensiones en los datos de test. Esto se debe a que los datos que vayan a evaluar el futuro modelo para predecir la existencia de exoplanetas deben hacerlo sobre las mismas dimensiones de los datos con los que ha sido entrenado. Por lo tanto, a todos los datos de test se les debe aplicar la misma función de reducción de dimensiones.



*Figura 25. Flujo de la estrella nº 10 CON exoplanetas con 603 dimensiones tras aplicar PCA*

En la *Figura 25* se puede observar el flujo de luz de la estrella nº 10 con exoplanetas - datos de entrenamiento- tras aplicar el algoritmo de reducción de dimensiones *PCA*. Se puede apreciar una importante diferencia con el flujo de la *Figura 23* puesto que el número de dimensiones se ha reducido de 3197 a 603.

## Entrenamiento

	LABEL	FLUX.1	FLUX.2	FLUX.3	...	FLUX.602	FLUX.603
0	2	0.0646	0.0967	-0.025	...	-0.007	0.0219
1	2	0.2809	0.0516	0.1324	...	-0.004	0.1323
2	2	-0.095	0.2125	-0.554	...	-0.021	-0.089
...	...	...	...	...	...	...	...
5085	1	-0.004	-0.161	0.0413	...	-0.005	-0.001
5086	1	0.3201	-0.382	-0.145	...	0.0029	0.0281

5087 filas x 604 columnas

Tabla 5. Muestra de los datos de entrenamiento tras aplicar PCA

## Test

	LABEL	FLUX.1	FLUX.2	FLUX.3	...	FLUX.602	FLUX.603
0	2	0.0652	0.0768	0.0227	...	-0.06	0.0975
1	2	0.1557	-0.1562	0.1027	...	-0.08	0.0348
2	2	0.1936	-0.242	0.3057	...	0.1293	0.095
...	...	...	...	...	...	...	...
568	1	0.2523	0.0893	-0.025	...	0.0096	-0.023
569	1	0.334	-0.288	-0.388	...	0.0071	0.0165

570 filas x 604 columnas

Tabla 6. Muestra de los datos de test tras aplicar PCA

En la *Tabla 5* y en la *Tabla 6* se muestran los datos de entrenamiento y de test, respectivamente, después de aplicar la reducción de dimensiones mediante *PCA*.

## Balanceamiento de datos

Para balancear el *dataset* de entrenamiento mediante *data augmentation*, es necesario conocer las características de los datos que van a ser replicados y alterados. Estos datos son los correspondientes a las estrellas con exoplanetas confirmados -*LABEL* = 2-, que son 37 de las 5087 totales -alrededor del 0.7%-.

Para realizar las distintas modificaciones de los datos se pueden combinar dos métodos. El primero consiste en **alterar** todos los datos el **mismo factor**, esto es, sumar o restar la misma cantidad a todos los valores de las dimensiones de cada estrella. De esta forma, todas las dimensiones se alteran en la misma proporción, guardando así las relaciones existentes entre las mismas, siendo dichas relaciones las que albergan la información importante para la detección de exoplanetas.

El segundo método consiste en añadir **ruido** a los flujos de las estrellas. Este ruido debe ser relativamente pequeño con respecto a la magnitud de los datos de los flujos puesto que, al generarse de forma aleatoria y diferente cada factor y ser añadido a cada dimensión, éstas se alteran de forma diferente entre sí, lo que podría modificar la información esencial de cada registro si este ruido es demasiado grande. Un ruido con una varianza entre 0.0001 y 0.00001 sería lo suficientemente fuerte como para alterar de forma significativa los datos, pero no como para modificarlos en exceso perdiendo información.

En cuanto al primer método, una estrategia válida consiste en aumentar y disminuir todos los valores de los datos en los siguientes factores: 0.02, 0.04, 0.06, 0.08 y 0.1. De esta forma, se obtienen un total de 10 modificaciones.

Posteriormente, a cada uno de los registros resultantes más los originales, que son  $10 \times 37 + 37 = 407$ , se les aplican 6 instancias aleatorias de ruido con distinta varianza - 0.000015, 0.00003, 0.000045, 0.00006, 0.00075 y 0.00009-. De esta forma, los registros de estrellas con exoplanetas resultan en un total de  $407 \times 6 + 407 = 2849$ , conformando un  $\frac{2849}{5050+2849} \times 100 \approx 36.1\%$  del total del *dataset*.

Tras estos dos tratamientos diferentes se consigue corregir el desbalanceamiento inicial del *dataset* de entrenamiento. En la *Tabla 7* se muestran los nuevos datos tratados, con dimensiones reducidas y balanceados.

	LABEL	FLUX.1	FLUX.2	FLUX.3	...	FLUX.602	FLUX.603
0	2	0.0646	0.0967	-0.025	...	-0.007	0.0219
1	2	0.2809	0.0516	0.1324	...	-0.004	0.1323
2	2	-0.095	0.2125	-0.554	...	-0.021	-0.089
...	...	...	...	...	...	...	...
7897	1	-0.004	-0.161	0.0413	...	-0.005	-0.001
7898	1	0.3201	-0.382	-0.145	...	0.0029	0.0281

7899 filas x 604 columnas

*Tabla 7.* Muestra de los datos de test tras el balanceamiento de los datos

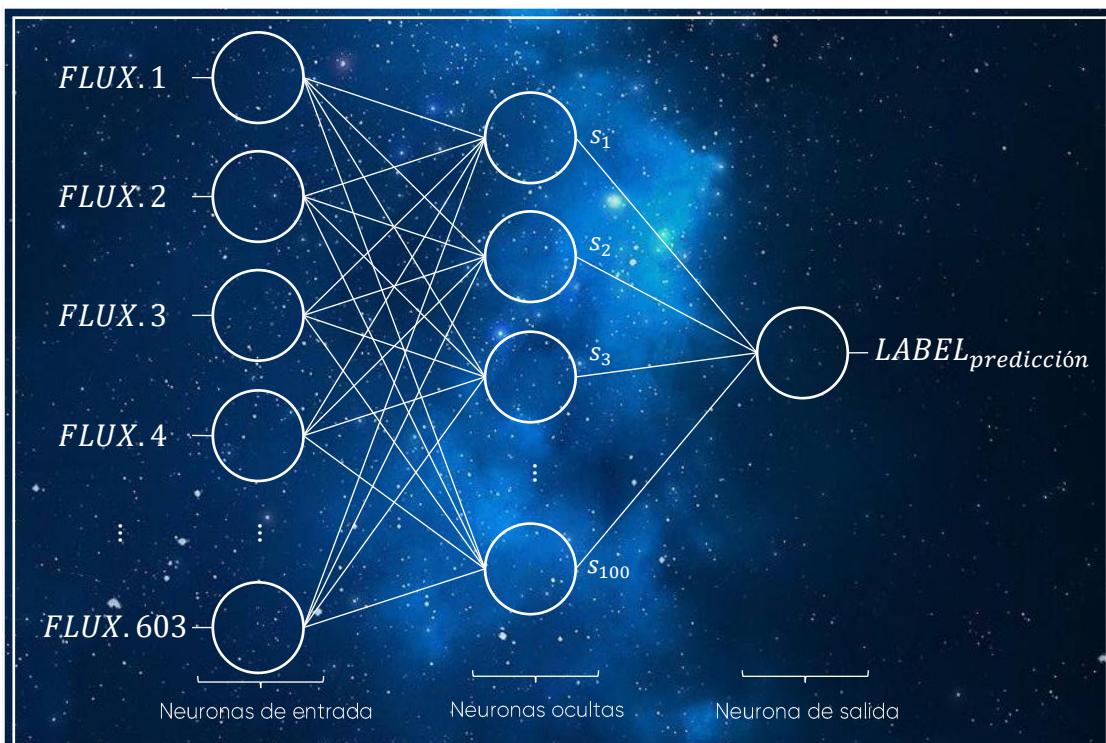
En la *Tabla 8* se muestra cómo están los datos, tanto de entrenamiento como de test, clasificados según los valores de la columna ‘LABEL’, que representa la existencia o no de exoplanetas en cada estrella.

LABEL	Entrenamiento	Test
1	5050	565
2	2849	5

*Tabla 8.* Clasificación de datos según ‘LABEL’

## Modelo

El modelo utilizado es una red neuronal artificial secuencial, es decir, una red neuronal cuyas salidas de cada capa son las entradas de la siguiente capa. Esta red está compuesta por una capa de 603 dimensiones de entrada -una por cada dimensión del conjunto de datos de entrenamiento-, por lo que tiene 603 neuronas, y 100 dimensiones de salida. La capa oculta está conformada, por tanto, de 100 dimensiones de entrada y 1 dimensión de salida. Consecuentemente, la capa de salida está conformada por 1 neurona, que aportará el *output* binario correspondiente. Esta arquitectura se muestra en la *Figura 26*.



*Figura 26. Arquitectura de la red neuronal utilizada para la predicción*

Tras varias pruebas con diferentes arquitecturas, esta es la que mejores resultados ha obtenido. Mientras que con más capas la red caía fácilmente en sobre-entrenamiento. Con el número de neuronas ocurría algo similar, ya que, a más neuronas, más propensa era la red al sobre-entrenamiento. Además, cuántas más capas y más neuronas se le incorporan a la red, la complejidad y el tiempo de entrenamiento aumentan considerablemente.

En cuanto al resto de parámetros, la función de activación de la capa de salida es  $sigmoid(x) = \frac{1}{1+\exp(-x)}$ , que devuelve un valor entre 0 y 1. La *loss function* del modelo se trata de *binary crossentropy*, puesto que se trata de una clasificación binaria. Los datos de entrenamiento con los que la red se entrena se van a propagar en lotes de tamaño 20, asegurando un menor uso de memoria y una mayor velocidad de entrenamiento sin perder precisión durante el proceso. El número de *EPOCHs* es de 10 y se utilizan los datos de test como datos de validación. Todos estos parámetros de ajustan principalmente mediante prueba y error -al igual que la arquitectura de la red-, siendo esta configuración la que mejores resultados obtiene.

Tras varias pruebas con diferentes modelos de clasificación, este es el que mejores resultados logra. Por ello, es el modelo elegido para realizar las predicciones. Los resultados se encuentran en el siguiente apartado.

## Resultados

Una vez entrenado el modelo, las predicciones obtenidas de los datos de test se muestran en la *Tabla 9*.

Predicción Real		SIN exoplanetas	CON exoplanetas
SIN exoplanetas	562	3	
CON exoplanetas	3	2	

*Tabla 9. Matriz de confusión de las predicciones del modelo*

Los resultados muestran que el modelo ha logrado predecir 562 de las 565 estrellas sin exoplanetas y 2 de las 5 estrellas con exoplanetas. Es decir, 562 verdaderos negativos, 2 verdaderos positivos, 3 falsos positivos y 3 falsos negativos. Para medir los resultados del sistema, existen varias fórmulas:

$$\text{precisión} = \frac{VP}{VP + FP}$$

$$\text{exhaustividad} = \frac{VP}{VP + FN}$$

$$F1 = 2 \times \frac{\text{precisión} \times \text{exhaustividad}}{\text{precisión} + \text{exhaustividad}}$$

$$\text{exactitud} = \frac{VP + VN}{VP + VN + FP + FN}$$

donde *VP* son los verdaderos positivos; *FP*, los falsos positivos; *VN*, los verdaderos negativos; y *FN*, los falsos negativos. En este caso, como el conjunto de datos de test está desbalanceado, la medida *exactitud* no resulta interesante puesto que dará un valor cercano al 100% sin medir de forma realista los resultados.

Por lo tanto, el sistema, entrenado con los datos finales de entrenamiento -tras el primer tratamiento, reducción de dimensiones y *data augmentation*- y probado con los datos de test -tras el primer tratamiento y reducción de dimensiones- ha obtenido una *precisión* del **40%**, una *exhaustividad* del **40%**, y un *F1* del **40%**.

Otro clasificador con el que se han realizado pruebas parametrizándolo de distintas formas es una red neuronal de tipo perceptrón multicapa, parecido al modelo anterior, pero que no ha logrado obtener los resultados esperados, ya que todos los datos de test los clasifica como estrellas sin exoplanetas, sin lograr predecir ninguna estrella con exoplanetas. Es decir, los 570 registros son clasificados como sin exoplanetas.

Como método de evaluación de todo el proceso de tratamiento de datos, reducción de dimensiones y balanceamiento del *dataset*, los resultados obtenidos con la misma configuración del modelo, pero con los datos en bruto -sin ningún tipo de tratamiento ni adecuación-, se muestran en la *Tabla 10*.

Predicción		SIN exoplanetas	CON exoplanetas
Real			
SIN exoplanetas		346	219
CON exoplanetas		2	3

**Tabla 10.** Matriz de confusión de las predicciones del modelo con los datos en bruto

Estos resultados contrastan con los obtenidos mediante el entrenamiento con los datos procesados una *precisión* del **1.35%**, una *exhaustividad* del **60%**, y un *F1* del **2.64%**, evidenciando la importancia de este proceso y la mejora significativa en los resultados obtenidos.

## Futuros trabajos

La red neuronal utilizada en este proyecto se muestra muy prometedora con los resultados obtenidos, gracias, además, a su amplio rango de parametrización, pudiendo elegir el número de capas, dimensiones, funciones de activación, *loss function*, número de *EPOCHs*, tamaño de lotes de entrenamientos, etc.

El cuello de botella de todo el proceso se puede encontrar en el tratamiento de datos, más concretamente en el balanceamiento del *dataset*. Siempre se puede encontrar una forma más óptima de replicar y alterar los registros deseados, así como elegir cuándo el conjunto de datos se encuentra lo suficientemente balanceado. Este proceso resulta ser el más crítico a la hora de entrenar el modelo puesto que, de no hacerse correctamente, es muy probable que la red neuronal presente sobre-entrenamiento en el caso de que los datos replicados sean demasiado parecidos entre sí, además de muy numerosos; o, por el contrario, es también muy sencillo que el modelo no obtenga un rango de datos suficientemente amplio y variado como para “aprender” de los mismos.

Por lo tanto, optimizar y mejorar este proceso puede ser clave para mejorar las predicciones, adaptando de igual manera los parámetros del modelo de predicción conforme a las características del *dataset* resultante de este proceso.

## Impactos ambientales y sociales

Las máquinas han conseguido que los medios físicos utilizados para el tratamiento de datos se hayan reducido considerablemente, y con ellos el impacto negativo para el medioambiente. Al ser el presente proyecto un estudio sobre el procesamiento de datos desarrollado íntegramente en medios virtuales con la ayuda de distintas tecnologías como *Python*, el impacto medioambiental se puede considerar nulo.

En el ámbito de lo social, el impacto que puede provocar este estudio sobre el tratamiento de datos y detección de exoplanetas es, en última instancia, inimaginable. Lograr un proceso que contribuya a automatizar y facilitar, con la ayuda del *machine learning*, la detección de exoplanetas supondría un cuantioso avance en la exploración espacial. Avanzar en esta dirección es imprescindible para el ser humano, dado que la búsqueda de nuevos mundos resultará de capital importancia en el futuro del hombre por multitud de motivos: la búsqueda de vida extraterrestre, la comprensión del universo, la expansión del ser humano, etc., lo que tendría grandiosas implicaciones no solo para la sociedad, sino para la humanidad.

## Conclusiones

La complejidad de los conjuntos de datos utilizados hace que las técnicas de tratamiento y adecuación deban ser escogidas y realizadas con precisión tras un escrupuloso estudio de los datos. Estas técnicas pueden variar por completo la predicción del modelo y resultan de suma importancia en la obtención final de resultados.

El *machine learning* es una potente herramienta que, utilizada correctamente, puede facilitar y mejorar notablemente todo el trabajo relativo a los datos. No obstante, de utilizarse de forma imprecisa y poco acertada, resulta del todo inútil, desvirtuando la información. Es por esto que el proceso de tratamiento de datos requiere necesariamente del ingenio humano, basado en el conocimiento, para ser aplicado en la medida y forma más adecuadas dada la propia naturaleza de los datos. Solo después de un proceso bien razonado, analizado y llevado a cabo, se pueden lograr unos acertados modelos de predicción preparados para nuevos datos.

Todo este complejo transcurso referente a los datos puede derivar en el mayor de los premios en lo que concierne a la exploración espacial: el hallazgo de exoplanetas. Nuevos mundos esperan a ser encontrados, y la humanidad está preparada para ello. Solo el trabajo duro conlleva su consecución. *Un camino arduo conduce a las estrellas.*

“*Per aspera ad astra*”

## Glosario de términos

Tránsito: Fenómeno astronómico durante el cual un astro realiza una trayectoria que interseca la línea recta existente entre otro astro y el observador. El fenómeno más conocido de este tipo es el eclipse solar.

Sistema inteligente: Programa computacional que trata de imitar el funcionamiento de la inteligencia humana para resolver problemas.

Input: Conjunto de datos o factores que son introducidos en un proceso.

Output: Conjunto de datos o factores que son resultado de un proceso.

Instrucciones: Secuencia de acciones o reglas para algún fin.

Lógica: Abstracción de las ideas inherentes a la realidad.

Aprendizaje: Proceso por el cual un modelo lógico artificial o natural se reestructura para adaptarse a una situación no experimentada previamente adquiriendo nueva información.

Entropía: magnitud que mide el número de microestados equivalentes para un mismo macroestado de un sistema, existiendo una tendencia a cambiar a macroestados de existencia con mayor multiplicidad y, por ello, más probables y con menor información. A mayor entropía, mayor número de microestados para un mismo macroestado y, por tanto, mayor probabilidad de suceder; así como menor información intrínseca en el sistema.

Ángstrom: unidad de medida del sistema internacional empleada para expresar longitudes de onda.

Tiempo Dinámico Baricéntrico (TDB): escala de tiempo relativista utilizada en astronomía que tiene en cuenta la dilatación temporal provocada por la diferencia de la velocidad o de la situación con respecto a un campo gravitacional definida en la Teoría de la Relatividad General.

Radiancia espectral: unidad de medida de radiación térmica.

Matriz de covarianza: matriz cuadrada que contiene la covarianza entre las dimensiones de un vector.

Autovectores: vectores nulos que, al ser transformados, se convierten en un múltiplo escalar de sí mismos.

Autovalores: valores escalares de los autovectores.

Matriz de proyección: matriz cuadrada que ofrece una proyección de un espacio vectorial a un subespacio vectorial.

Año luz: unidad de medida de longitud equivalente a la distancia que recorre la luz en el vacío en un año.

Espacio-tiempo: espacio cuatridimensional conformado por tres dimensiones espaciales y una temporal formulado en la teoría de la Relatividad General.

## Referencias

- [1] N. E. S. Institute, «NASA Exoplanet Archive,» Caltech, 7 Julio 2020. [En línea]. Available: <https://exoplanetarchive.ipac.caltech.edu/>. [Último acceso: 7 Julio 2020].
- [2] WinterDelta, «Kaggle,» 2017. [En línea]. Available: <https://www.kaggle.com/keplersmachines/kepler-labelled-time-series-data?select=exoTest.csv>. [Último acceso: 2019].
- [3] E. Alpaydin, Introduction to Machine Learning, MIT Press, 2020.
- [4] R. M. & J. D. U. John E. Hopcroft, Introduction to Automata Theory, Languages and Computation, Pearson, 2013.
- [5] D. Pool, Computational Intelligence: A Logical Approach, New York: Oxford University Press, 2018.
- [6] S. J. R. & P. Norving, Artificial Intelligence: A Modern Approach, Prentice Hall, 2009.
- [7] L. E. S. & M. Tonantzintla, Aprendizaje Automático: conceptos básicos y avanzados, INAOE, 2006.
- [8] J. T. & W. J. Meurer, Logistic Regression Relating Patient Characteristics to Outcomes, JAMA, 2016.
- [9] M. H. Hassoun, Fundamentals of artificial neural networks, MIT press, 1995.
- [10] J. P. L. Mangin y R. F. L. & J. M. F. Fernández, Las redes neuronales artificiales, Netbiblo, 2008.
- [11] C. Shannon, «Programming a Computer for Playing Chess,» *Philosophical Magazine*, vol. 41, nº 314, 1950.
- [12] S. T. S. Institue, «Mikulki Archive,» NASA, [En línea]. Available: <https://archive.stsci.edu/missions-and-data/k2>. [Último acceso: Junio 2020].
- [13] S. T. S. Institute, «Mikulski Archive,» NASA, [En línea]. Available: <https://archive.stsci.edu/>. [Último acceso: Junio 2020].
- [14] S. M. Holland, «Principal Components Analysis,» University of Georgia, Athens, GA, 2019.
- [15] M. Zhu, C. Xu y Y.-F. B. Wu, «IFME: information filtering by multiple examples with under-sampling in a digital library environment,» ACM, p. 107–110, 2013.
- [16] J. Beaulieu, «JulienBeaulieu,» 2019. [En línea]. Available: <https://julienbeaulieu.gitbook.io/wiki/sciences/machine-learning/unsupervised-learning/k-means>. [Último acceso: Junio 2020].
- [17] J. Beaulieu, «JulienBeaulieu,» 2019. [En línea]. Available: <https://julienbeaulieu.gitbook.io/wiki/sciences/machine-learning/unsupervised-learning/hierarchical-clustering>. [Último acceso: Junio 2020].
- [18] J. Beaulieu, «JulienBeaulieu,» 2019. [En línea]. Available: <https://julienbeaulieu.gitbook.io/wiki/sciences/machine-learning/unsupervised-learning/dbscan>. [Último acceso: Junio 2020].

# Bibliografía

<https://www.kaggle.com/learn/python> (Enero 2020)

<https://www.kaggle.com/learn/intro-to-machine-learning> (Enero 2020)

<https://pandas.pydata.org/docs/index.html> (Enero 2020)

<https://medium.com/datos-y-ciencia/aprendizaje-no-supervisado-en-machine-learning-agrupaci%C3%B3n-b3n-bb8f25813edc> (Mayo 2020)

<https://julienbeaulieu.gitbook.io/wiki/sciences/machine-learning/unsupervised-learning/> (Mayo 2020)

[https://web.archive.org/web/19990218034308/http://ai.bpa.arizona.edu/papers/mlir93/subsection3\\_7\\_1.html](https://web.archive.org/web/19990218034308/http://ai.bpa.arizona.edu/papers/mlir93/subsection3_7_1.html) (Mayo 2020)

<http://www.niedermayer.ca/node/22> (Mayo 2020)

<https://julienbeaulieu.gitbook.io/wiki/sciences/math/statistics/logistic-regression> (Mayo 2020)

<https://www.kaggle.com/keplersmachines/kepler-labelled-time-series-data#exoTrain.csv> (Diciembre 2019)

<https://github.com/winterdelta/KeplerAI> (Abril 2020)

<https://www.cfa.harvard.edu/~avanderb/tutorial/tutorial2.html> (Abril 2020)

<https://archive.stsci.edu/k2/> (Abril 2020)

[https://github.com/winterdelta/KeplerAI/blob/master/PCA/PCA\\_in\\_Python.ipynb](https://github.com/winterdelta/KeplerAI/blob/master/PCA/PCA_in_Python.ipynb) (Abril 2020)

<https://github.com/winterdelta/KeplerAI/blob/master/Normalise/Normalise.ipynb> (Abril 2020)

<https://github.com/winterdelta/KeplerAI/blob/master/Normalise/NormaliseRapid.ipynb> (Abril 2020)

<https://keplerscience.arc.nasa.gov/k2-observing.html> (Abril 2020)

<https://spaceplace.nasa.gov/other-solar-systems/en/> (Junio 2020)

<https://spaceplace.nasa.gov/barycenter/en/> (Junio 2020)

<https://spaceplace.nasa.gov/all-about-exoplanets/en/> (Junio 2020)

<https://www.cfa.harvard.edu/~avanderb/k2.html> (Junio 2020)

<https://keplerscience.arc.nasa.gov/k2-observing.html> (Junio 2020)

[https://mathworld.wolfram.com/ProjectionMatrix.html#:~:text=A%20projection%20matrix%20is%20an,\(1\)](https://mathworld.wolfram.com/ProjectionMatrix.html#:~:text=A%20projection%20matrix%20is%20an,(1)) (Junio 2020)

<https://www.quora.com/What-is-the-best-way-to-choose-the-number-of-components-in-PCA-during-dimensionality-reduction> (Junio 2020)

<https://towardsdatascience.com/a-complete-guide-to-principal-component-analysis-pca-in-machine-learning-664f34fc3e5a> (Junio 2020)

<https://nanonets.com/blog/data-augmentation-how-to-use-deep-learning-when-you-have-limited-data-part-2/> (Junio 2020)

<http://www.deepspace.ucsb.edu/wp-content/uploads/2012/02/134-Notes-a.pdf> (Julio 2020)

<https://www.kaggle.com/muonneutrino/exoplanet-data-visualization-and-exploration> (Julio 2020)

<https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html> (Julio 2020)

<https://www.aprendemachinelearning.com/comprende-principal-component-analysis/> (Julio 2020)

<https://nanonets.com/blog/data-augmentation-how-to-use-deep-learning-when-you-have-limited-data-part-2/> (Julio 2020)

<https://keras.io/api/layers/activations/> (Julio 2020)

Trabajo de Fin de Grado

*Machine Learning para el tratamiento  
de datos y la detección de exoplanetas  
mediante el método de tránsito*

Autor:  
José Javier Gómez de Diego

Tutor:  
Fernando Ortega Requena

Escuela Técnica Superior de Ingeniería de Sistemas Informáticos  
Universidad Politécnica de Madrid  
Grado en Ingeniería de Sistemas de Información  
Julio 2020