Relación entre las características de una canción y su popularidad en Spotify

Bautista Gómez Juan Pablo
Facultad de ingeniería
Universidad Autónoma de San Luis Potosí
San Luis Potosí, México
A328510@alumnos.uaslp.mx

Resumen – Las plataformas de streaming de música como Spotify han permitido dar mayor alcance a la música y con ello ha surgido la necesidad de saber qué características influyen en que una canción tenga mayor popularidad y por ende mayores reproducciones respecto a otras. Con el objetivo de resolver esta incógnita. En este trabajo se aplicó el método de descubrimiento de conocimiento en bases de datos (KDD), el cual hace uso del data set "Spotify Top 200 Charts (2020-2021)" obtenido de la plataforma de kaggle, el cual contiene las características de cada canción, como, tempo (BPM), tonalidad, genero de la canción, volumen en decibeles, etc. Además, se aplicaron los algoritmos de, regresión lineal, árboles de regresión y un modelo multicapa de red neuronal, para predecir la cantidad de reproducciones de una canción según sus características y con ello poder definir qué características tienen mayor importancia en que una canción tenga más reproducciones.

Palabras clave – KDD, Spotify, regresión lineal, árboles de regresión, redes neuronales.

1. INTRODUCCIÓN

Con el avance de la tecnología y el internet, la industria musical ha sufrido muchos cambios, uno de los más importantes se dio a partir de la aparición de las plataformas de streaming de música, como Spotify, Apple music, Deezer, etc. Ofreciendo acceso a una cantidad inmensa de música a un precio menor en comparación de la compra física. En donde Spotify se coloca como una de las plataformas más usadas para escuchar música, ofreciendo a los usuarios recomendaciones de canciones personalizadas, playlist, podcasts, un catálogo amplio de canciones, etc. Debido al aumento del uso de plataformas como Spotify nace la necesidad de estudiar qué influye en que una canción tenga un mayor éxito (reproducciones) entre los usuarios.

Este trabajo aborda esta problemática a partir del análisis de las diversas características de una canción, tales como el tempo (BPM), la tonalidad, la capacidad de baile, entre otras, a partir del método KDD para extraer información del data set "Spotify Top 200 Charts (2020-2021)"**[1]**, así como hacer uso de los algoritmos de regresión lineal y árboles de regresión que nos permitan predecir las reproducciones de una canción en base a sus características y con ello poder obtener aquellas características que tengan una mayor influencia en las reproducciones de una canción. Y cuya información puede ser usada por artistas y las diversas empresas discográficas para crear canciones que tengan un mayor impacto entre los usuarios o permitirles llegar a un público en específico.

2. TRABAJOS RELACIONADOS

El articulo "**What Makes a Song Trend? Cluster Analysis of Musical Attributes for Spotify Top Trending Songs" [2]** publicado por Zayd et al (2020) aborda la problemática haciendo uso de las características proporcionadas por la API de Spotify del top 100 de los años 2017 y 2018. El data set usado en este trabajo contiene algunos atributos en común con el data set usado por Zayd, como son, tempo (BPM), Energy, Loudness, entre otras. El trabajo propuesto en el trabajo mencionado hace uso del método K Means de clustering, obteniendo como resultado que **las canciones más emocionantes, con sonidos que tienden al pop y canciones más adecuadas para bailar** según el índice "Danceability", **tienden a ser más populares.**

Laura Colley et al (2022) en su artículo "**Elucidation of the Relationship Between a Song's Spotify Descriptive Metrics and its Popularity on Various Platforms**" **[4]** aborda la problemática, tomando como base las métricas de Spotify, añadiendo algunos otros atributos obtenidos de diversas plataformas como Billboard, Google trends, YouTube, etc. En dicho artículo se utilizaron diversos modelos de minería, como VSM, random forest, regresión lineal, regresión polinomial y árboles de decisión, obteniendo una correlación de $R^2 = 0.650$ con el método de random forest, concluyendo que **no hay una correlación tan fuerte entre las variables como para predecir correctamente la popularidad de una canción** en base a los atributos estudiados.

3. PROPUESTA DE ESTUDIO

La metodología propuesta consiste en aplicar las 5 etapas del método KDD **[3]** sobre el data set "Spotify Top 200 Charts (2020-2021)" buscando encontrar la relación entre las características de cada canción y como influyen en que una canción tenga un éxito mayor o menor, tomando como indicador de éxito la cantidad de reproducciones, a partir de la aplicación de los algoritmos: regresión lineal, árboles de regresión y un modelo multicapa de redes neuronales.

Estos algoritmos fueron seleccionados debido a su compatibilidad para predecir valores continuos, debido a que la **variable objetivo**, elegida, corresponde a la **columna "Streams"** que contiene el número de reproducciones de cada canción en Spotify.

Una desventaja respecto a otros trabajos es el tamaño del data set, pues el conjunto de datos utilizado cuenta con poco más de 1500 registros, respecto a los 500k registros utilizados en el trabajo de Laura Colley.

4. METODOLOGÍA

La solución propuesta se basa en el método de descubrimiento de bases de datos (KDD), el cual consiste en cinco etapas, que permiten extraer la información del conjunto de datos.

4.1 Selección

El data set usado para este trabajo fue publicado por Sashank Pilla en la plataforma de Kaggle, y cuyos datos fueron obtenidos de la página **spotifycharts.com** y de la librería **Spotify** según el autor. Estos datos corresponden a las canciones que en algún momento estuvieron en el top 200 mundial en Spotify en los años 2020 y 2021. Teniendo un total de 1556 registros y 23 atributos, donde el atributo **Streams** se definió como la **variable objetivo.** Todos los atributos del conjunto de datos se presentan en la siguiente tabla.

Atributo	Descripción	Valores posibles
Index	Identificador de cada fila	1 a 1556
Highest Charting Position	Posición más alta obtenida en el top 200 entre 2020 y 2021	1 a 200
Number of Times Charted	Número de veces que la canción estuvo en el top 200	1 a 142
Week of Highest Charting	Semana en que la canción estuvo en su posición más alta	Fecha con el formato YYYY – MM – DD
Song Name	Contiene el nombre de la canción	Cadenas con el nombre de la canción
Streams	Número de reproducciones de la canción	4176083 a 48633449
Artist	Nombre del artista o artistas de la canción	Cadena con el nombre del artista(s)
Artist Followers	Número de seguidores del artista principal	4883 a 83337783
Song ID	ID asignado a la canción dentro de Spotify	Cadena de letras y números
Genre	Genero(s) a los que pertenece la canción	Cadena con el género
Release Date	Fecha en la que la canción fue lanzada	Fecha con el formato YYYY – MM – DD
Weeks Charted	Fechas en las que la canción estuvo en el top 200	Fechas con el formato YYYY – MM – DD
Popularity	Popularidad de la canción	1 a 100 (100 indica mayor popularidad)

Danceability	Determina que tan adecuada es la canción para bailar	0 a 1.0
Energy	Indica la intensidad de la canción. Generalmente canciones más rápidas, con mayor sonido	0 a 1.0
Loudness	Indica el volumen de la canción en decibeles	1.6 a -0.26
Speechiness	Indica la presencia de palabras en la canción	0 a 1.0
Acousticness	Indica que tan acústica es la canción (Presencia de instrumentos como guitarra, piano, etc)	0 a 1.0
Liveness	Mide la cantidad de público en la canción (Generalmente para canciones grabadas en vivo)	0 a 1.0
Tempo	Beats por minuto en la canción	46 a 205
Duration (ms)	Duración de la canción en milisegundos	30133 a 588139
Valence	Indica si una canción es más positiva o negatica	0 a 1.0
Chord	La tonalidad principal de la canción	Valores de las clases de tonos (A, B, C, C#, D, D#, E, F, F#, …)

4.2 Preprocesamiento

Esta etapa se realiza una limpieza de los datos, la cual consiste en la identificación de datos nulos, duplicados o desconocidos. En donde se encontraron 11 filas con datos vacíos, las cuales se eliminaron debido a que se en todas ellas faltaban la mayoría de las características de la canción y por tratarse de un conjunto pequeño se decidió eliminarla. De forma que el conjunto de datos se redujo a 1545 registros de los 1556 que había originalmente.

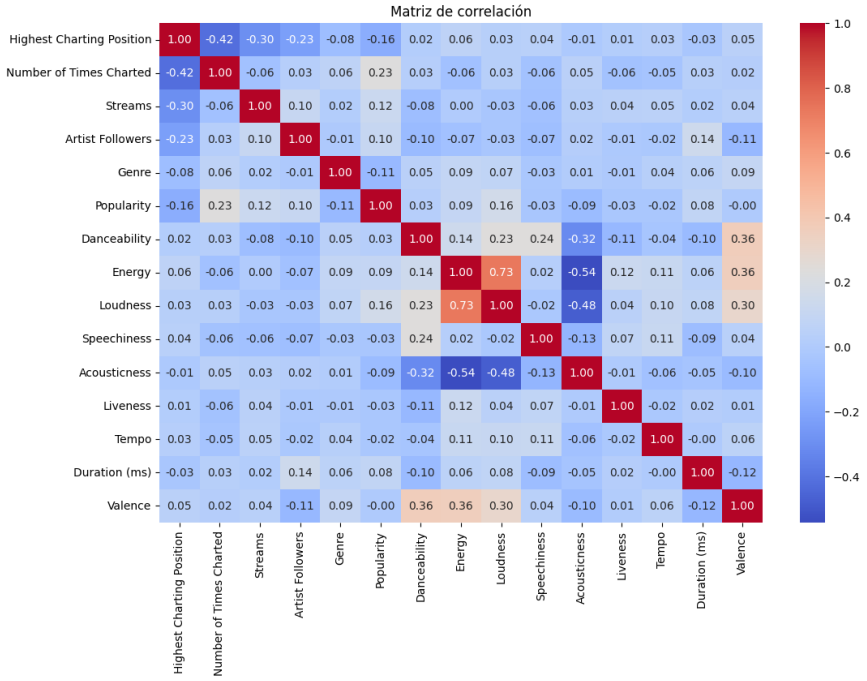
Con los datos resultantes se procedió a transformar el tipo de dato de algunas columnas, debido a que algunas de ellas presentaban un tipo de dato “object”, que no correspondía con el tipo de dato real de la columna, como puede ser String, Int, Float, etc. Dando como resultado las siguientes columnas con su tipo de dato correcto después de la transformación. Cabe mencionar que en este proceso ningún dato fue alterado, solo se realizó la conversión de su respectiva columna al tipo de dato correcto.

```
<class 'pandas.core.frame.DataFrame'>
Index: 1545 entries, 0 to 1555
Data columns (total 23 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   Index                     1545 non-null   int64
 1   Highest Charting Position 1545 non-null   int64
 2   Number of Times Charted   1545 non-null   int64
 3   Week of Highest Charting  1545 non-null   object
 4   Song Name                 1545 non-null   object
 5   Streams                   1545 non-null   int64
 6   Artist                    1545 non-null   object
 7   Artist Followers          1545 non-null   int64
 8   Song ID                   1545 non-null   object
 9   Genre                     1545 non-null   object
 10  Release Date              1545 non-null   object
 11  Weeks Charted             1545 non-null   object
 12  Popularity                1545 non-null   int64
 13  Danceability              1545 non-null   float64
 14  Energy                    1545 non-null   float64
 15  Loudness                  1545 non-null   float64
 16  Speechiness               1545 non-null   float64
 17  Acousticness              1545 non-null   float64
 18  Liveness                  1545 non-null   float64
 19  Tempo                     1545 non-null   float64
 20  Duration (ms)             1545 non-null   int64
 21  Valence                   1545 non-null   float64
 22  Chord                     1545 non-null   object
```

Una vez que las columnas tienen el tipo de dato correcto se realiza la conversión de algunos datos de clase a datos numéricos para que puedan ser usados por los modelos de regresión. La única columna a la que se le aplico esta conversión corresponde a la columna Genre, de forma que ahora cada genero se representa con valores enteros y no con cadenas de texto.

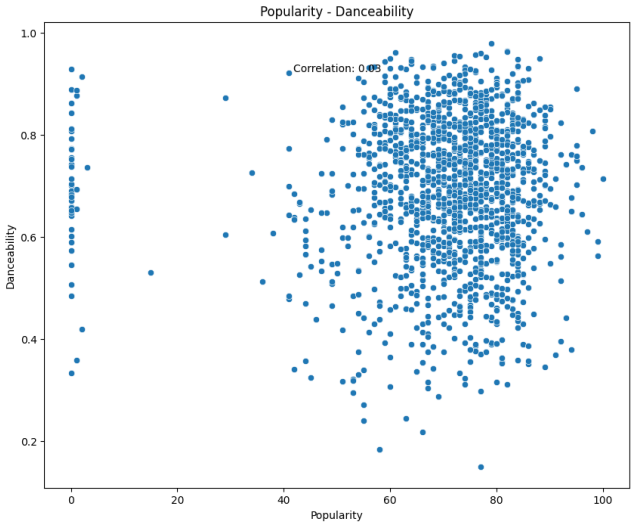
4.3 Transformación

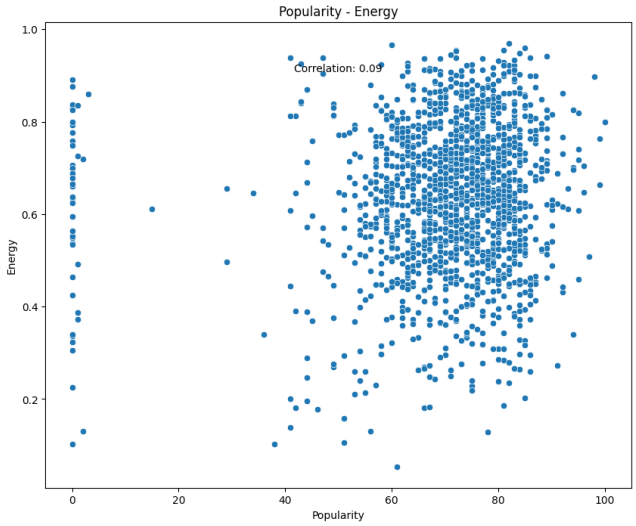
Una vez realizado el preprocesamiento de los datos, se decidió eliminar aquellas columnas que no sean relevantes para la predicción como lo son, nombre de la canción, artista, columnas que almacenan fechas, así como la tonalidad de la canción. De forma que al final se tiene un conjunto de datos con 15 atributos. Y para los cuales se realizó una matriz de correlación.

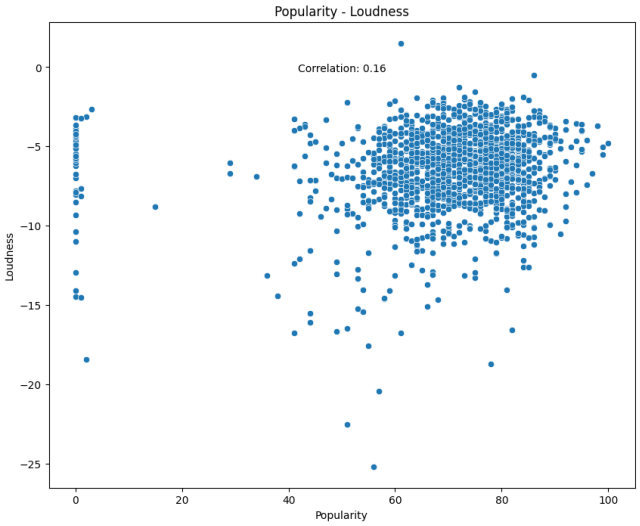


En esta matriz se puede observar que el atributo “Streams” que representa las reproducciones de la canción y que se definió como la variable objetivo presenta una mayor correlación con el atributo de popularidad. Así mismo la popularidad se correlaciona con otras variables como la energía de la canción, su volumen y que tan adecuada es la canción para bailar.

Y esto se comprueba al realizar los diagramas de dispersión con los atributos mencionados. En donde se puede notar una ligera correlación entre los atributos, pues a medida que aumentan, la popularidad lo hace igualmente. Sin embargo, los valores obtenidos en la matriz y los diagramas de dispersión nos indican que la correlación entre estos atributos es muy débil.







4.4 Minería de datos

Para la predicción de las reproducciones se eligieron los algoritmos de regresión lineal, árboles de regresión y perceptrón multicapa, debido a su compatibilidad para predecir valores continuos.

4.4.1 Regresión lineal [5]

Este método se basa en la predicción de una variable independiente a partir de otra u otras variables, a las cuales se les denomina dependientes, de forma que para este trabajo la variable independiente correspondería las reproducciones de cada canción, que es la variable que se quiere predecir, y las variables dependientes serían las características de la canción.
Es por ello que para este problema se hace uso de la formula de regresión lineal múltiple

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_k x_{ik} + \varepsilon_i$$

En donde "y" representa la variable que se desea predecir, B representa las constantes de correlación que indica que tanto aumenta o decrece una variable según el cambio en alguna de las variables independientes, X representa el valor de cada una de las variables independientes y E indica el error, es decir, que tanta distancia hay entre los valores reales y los valores predichos. Y cuya formula una vez ajustada permite predecir la variable independiente según los valores que se le otorguen a las variables dependientes.
Para la implementación de este método se usó la librería de scikit learn en Python, la cual contiene el método de regresión lineal sin modificar los hiperpárametros por default del algoritmo.

4.4.2 Árboles de regresión [6, 7]

El algoritmo de árboles de regresión es un tipo de modelo supervisado que genera una estructura de árbol para predecir una variable continua. Esto se logra a partir la partición de los datos de entrada, con el objetivo de reducir el error cuadrático medio **(MSE).** De forma que para cada partición se genera un modelo de árbol, los cuales son usados como nodos y que según la entrada que se le de al nodo raíz sigue un camino a través del árbol, hasta llegar a un nodo hoja, donde devuelve el promedio de la variable objetivo-almacenada en ese nodo hoja.

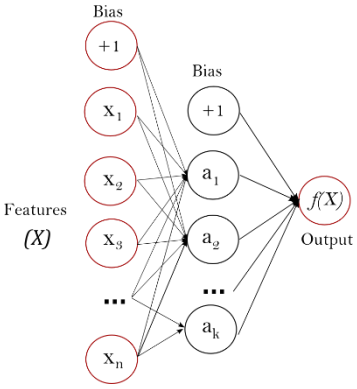
Para el uso de este modelo se usó nuevamente la librería de scikit learn que incluye un algoritmo de árboles de regresión.
Con el fin de obtener los hiperpárametros óptimos se usó la función **RandomizedSearchCV [8]**. Esta función permite encontrar la combinación de hiperpárametros que ofrezcan un mejor resultado. De forma que al aplicar esta función se obtuvieron los siguientes parámetros para el modelo de árboles de regresión:

- min_samples_leaf: 10
- max_features: 6,
- max_depth: 10,
- criterion: squared_error

4.4.3 Perceptrón multicapa (Regresión) [9]

El modelo de perceptrón multicapa, también conocido como Multi Layer Perceptrón Regression (MLPR), es un tipo de algoritmo basado en el modelo de redes neuronales, el cual consiste en un mínimo de tres capas, con sus respectivos nodos (neuronas).

- **Capa de entrada:** Es la primera capa de la red y recibe las características o datos.
- **Capas ocultas:** Las capas subsecuentes, llamadas capas ocultas, procesan la información recibida de la capa anterior y en base a ella permiten extraer características. Estas capas implementan una función de activación que permite la no linealidad en la red, es decir, que la red no produzca la misma salida, incluso para los mismos datos. Pueden existir de 1 a n capas ocultas.
- **Capa de salida:** La última capa, llamada capa de salida, contiene la predicción que ha hecho la red.



Cada una de las capas realiza un proceso llamado retro propagación **(backpropagation) [10, 11]**. Este proceso se encarga, a grandes rasgos, de introducir los datos en la red y ajustar los pesos de los nodos, con el objetivo de reducir el error cuadrático medio. Este proceso se repite hasta que se alcance el número de épocas **(epochs)** o que el error cuadrático medio sea menor al umbral definido, lo que indica que la red encontró una solución óptima para el problema.

La librería scikit learn incluye un modelo como el mencionado anteriormente, con el nombre de MLPR, el cual esta destinado para operaciones de regresión.
En una red neuronal los hiperpárametros toman un papel relevante en el resultado final, pues según el tipo de función de activación que se asigne a los nodos o el tamaño de cada capa permite obtener mejores o peores resultados. Con el fin de obtener los hiperpárametros óptimos se usó nuevamente la función **RandomizedSearchCV**, obteniendo como resultado los siguientes parámetros:

- solver: adam
- learning_rate: constant
- hidden_layer_sizes: (50, 50, 50)
- alpha: 0.05
- activation: relu

4.5 Resultados

Para el entrenamiento de los modelos se separaron los datos en conjuntos de entrenamiento y conjuntos de prueba, en donde, el conjunto de prueba corresponde al 30% de los datos y el 70% restante se usó para entrenar el modelo. Los algoritmos de regresión lineal y árboles de regresión se entrenaron usando los mismos datos de entrenamiento y de prueba, con el objetivo de que la comparación de los resultados sea lo más exacta posible. Por el contrario, para el modelo de perceptrón multicapa, los datos se sometieron a un proceso de estandarización, necesario para el correcto funcionamiento del modelo, sin embargo, el conjunto de datos de prueba y entrenamiento tienen la misma distribución utilizada en los modelos anteriores.

Los resultados obtenidos por cada modelo se obtuvieron a partir del cálculo del coeficiente de determinación entre los datos de prueba y los resultados esperados. A continuación, se muestra una tabla con los resultados obtenidos.

Modelo	**Coeficiente de determinación (R^2)**
Regresión lineal	0.14
Árboles de regresión	0.57
Perceptrón multicapa de regresión	0.75

Según los resultados obtenidos se puede determinar que el modelo de perceptrón multicapa proporciona un mejor coeficiente de determinación, con un valor R^2 de 0.75, aunque para un modelo de regresión este valor representa una baja confiabilidad para hacer predicciones.
Obtener un valor bajo de precisión era predecible, pues en la matriz de correlación y en los diagramas de dispersión se puede

notar que no hay una correlación fuerte con ninguno de los atributos del conjunto de datos.

Posteriormente, a partir de la función **permutation_importance** se obtuvo la importancia de cada atributo en el modelo multicapa. Obteniendo como resultado que los 5 atributos más importantes son: Number of Times Charted, Popularity, Highest Charting Position, Genre y Acousticness.

Característica	Importancia
Number of Times Charted	0.807504
Popularity	0.790953
Highest Charting Position	0.665752
Genre	0.100973
Acousticness	0.091626
Loudness	0.062378
Energy	0.034375
Valence	0.031450
Danceability	0.021487
Tempo	0.012732
Liveness	0.010351
Artist Followers	0.004512
Speechiness	0.003596
Duration (ms)	-0.009772

5. Conclusiones y trabajo futuro

Se puede observar que, en base a los métodos probados, una canción tiende a ser más exitosa si logra entrar múltiples veces dentro del top con un lugar alto. Lo que a su vez indica que una canción es más popular y que también afecta en que una canción pueda ser más exitosa. Sin embargo, no se pude confirmar que una canción con estas características sea exitosa, pues la precisión del mejor modelo probado es baja, por lo que se puede concluir que además de las características de cada canción existen otros factores que influyen en que una canción tenga una mayor cantidad de reproducciones.

El trabajo futuro incluye la búsqueda de data sets de otros años anteriores o más recientes, con el objetivo de tener un solo data set con una mayor cantidad de datos, con diferentes variaciones. Además, hacer uso de métodos más complejos y con ello proporcionar conclusiones más precisas.

Bibliografía

[1] Pillai, S. (2021). Spotify Top 200 Charts (2020-2021). Kaggle. Recuperado 28 de mayo de 2024, de https://www.kaggle.com/datasets/sashankpillai/spotify-top-200-charts-20202021/data

[2] What Makes a Song Trend? Cluster Analysis of Musical Attributes for Spotify Top Trending Songs. (2020). Journal Of Marketing Development And Competitiveness, 14(3). https://doi.org/10.33423/jmdc.v14i3.3065

[3] Pereira, S. R. T., Arteaga, I. H., Zambrano, S. J. C., Troya, A. H., & Pérez, J. C. A. (2016). Descubrimiento de patrones de desempeño académico con árboles de decisión en las competencias genéricas de la formación profesional. https://doi.org/10.16925/9789587600490

[4] Colley, L., Dybka, A., Gauthier, A., Laboissonniere, J., Mougeot, A., Mowla, N., Dick, K., Khalil, H., & Wainer, G. (2022). Elucidation of the Relationship Between a Song's Spotify Descriptive Metrics and its Popularity on Various Platforms. 2022 IEEE 46th Annual Computers, Software, And Applications Conference (COMPSAC). https://doi.org/10.1109/compsac54236.2022.00042

[5] Amazon AWS. (s. f.). ¿Qué es la regresión lineal? - Explicación del modelo de regresión lineal - AWS. Amazon Web Services, Inc. Recuperado 28 de mayo de 2024, de https://aws.amazon.com/es/what-is/linear-regression/

[6] Scikit Learn. (s. f.). 1.10. Decision Trees. Scikit-learn. Recuperado 28 de mayo de 2024, de https://scikit-learn.org/stable/modules/tree.html#tree

[7] IBM. (s. f.). IBM Cognos Analytics 11.1.x. Recuperado 28 de mayo de 2024, de https://www.ibm.com/docs/es/cognos-analytics/11.1.0?topic=tests-regression-tree

[8] Scikit Learn. (s. f.). Comparing randomized search and grid search for hyperparameter estimation. Scikit-learn. Recuperado 28 de mayo de 2024, de https://scikit-learn.org/stable/auto_examples/model_selection/plot_randomized_search.html

[9] Scikit Learn. (s. f.-b). 1.17. Modelos de redes neuronales (supervisadas) — documentación de scikit-learn - 0.24.1. Recuperado 28 de mayo de 2024, de https://qu4nt.github.io/sklearn-doc-es/modules/neural_networks_supervised.html

[10] Interactive Chaos. (s. f.). Backpropagation. Recuperado 28 de mayo de 2024, de https://interactivechaos.com/es/manual/tutorial-de-machine-learning/backpropagation

[11] Solis, D. D. (2021, 15 diciembre). Blog: Cómo funciona el algoritmo de backpropagation. Medium. https://medium.com/@ddiazsolis/blog-c%C3%B3mo-funciona-el-algoritmo-de-backpropagation-22575308f14b