15 de noviembre 2022

Entrega Final Proyecto Análisis del Dataset AnimeList

Ingeniería de Datos – ICC732

Andy Sandoval Neculcura
Christopher Alarcón Herrera

Universidad de La Frontera

Contenido

Entrega Final Proyecto Análisis del Dataset AnimeList

Ingenieria de datos.

Andy Sandoval Neculcura
Facultad de Ingenieria y Ciencias
Universidad de La Frontera
Temuco, Chile
a.sandoval17@ufromail.cl

Christopher Alarcón Herrera
Facultad de Ingenieria y Ciencias
Universidad de La Frontera
Temuco, Chile
c.alarcon18@ufromail.cl

***Resumen*—This electronic document is a "live" template and already defines the components of your paper [title, text, heads, etc.] in its style sheet. **CRITICAL: Do Not Use Symbols, Special Characters, Footnotes, or Math in Paper Title or Abstract.* (*Abstract*)**

I. Introducción

En los últimos años, el anime ha ido creciendo exponencialmente y ha ido llegando a un público cada vez más grande "Puede que el anime esté atravesando una de sus mejores épocas de la historia. El género tiene audiencias en casi cualquier parte del mundo." (Nolan Rada Galindo 9 de junio de 2021) , de esto se han dado cuenta muchos servicios de streaming y estudios de televisión, por lo cual, varios quieren aprovechar esta oportunidad para poder atraer a más gente a sintonizar sus servicios y beneficiarse monetariamente "En las últimas décadas hemos vivido un auténtico renacimiento en el mundo del anime con la animación japonesa volviéndose un hobby mucho más normalizado y "mainstream" … Las películas de anime recaudan millones y millones" (MARILÓ DELGADO 21 Mayo 2022).

Un nuevo estudio de televisión dedicado específicamente a la transmisión de anime, tiene problemas para decidir acerca de qué shows añadir a su transmisión habitual, tienen como objetivo poder captar más público e ir creciendo como estudio alcanzando más rating, fuimos contratados para poder ayudarlos a alcanzar dicho objetivo. Para esto, realizaremos un análisis exploratorio de datos de un dataset histórico que obtuvimos de diversos animes que se han ido estrenando a lo largo de los años, con la finalidad de analizar las variables que nos sirvan y eliminar las que no, ordenar los datos con los que contamos y realizar boxplots y otros tipos de gráficos para examinar y obtener toda la información que podamos del dataset.

Lo que nos motivó a trabajar en conjunto en este estudio fue el querer ayudar a seguir expandiendo la cultura del anime, asistiendo a este estudio de televisión, se logrará que llegue a más televidentes mostrándoles animaciones de calidad, para que se interesen en ellas y sigan de manera autónoma buscando más animes.

II. Metodologia

Requeríamos conocer contra que nos enfrentábamos, era necesario investigar la información que teníamos. Partimos por conocer las dimensiones de la información y su tipo, donde una vez visualizadas nos encontramos con variables nulas, que en ciertas ocasiones tienden a ser denominadas outliers o valores fuera de rango, cuando tienen algún valor asignado. Posteriormente obtenemos los nombres de nuestros atributos para conocer mejor nuestro dataset, se crearon nuevas columnas con el fin de describir mejor la información que se conocía, por ejemplo, se creó la variable "duration_minutes" para guardar la duración de los animes por episodios, reflejadas en minutos, además de separar la columna "aired" la cual era una cadena larga que nos mostraba la fecha de inicio y fecha de término del anime, se optó por separar la respectiva variable.

Iniciando el proceso de la exploración de datos o EDA, tuvimos que comprender la naturaleza de los datos a disposición, para la creación de modelos experimentales. Primero, observamos la matriz de correlación de nuestro dataset (Ilustración 1).

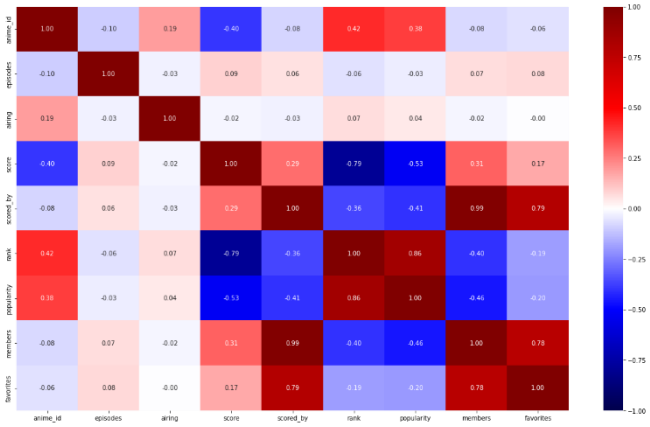

Ilustración 1 Matriz de correlación

De la cual existe una alta correlación entre el atributo Members (Cantidad de miembros que tienen guardado el anime) y Score_by (Puntuación por cantidad de miembro total), de estas se tomó la decisión de eliminar solamente el atributo Score_by debido a que para las problemáticas su aporte es nulo.

Otros atributos por eliminar fueron los siguientes:

- Opening_theme
- Ending_theme
- Licensor
- Related
- Aired_string
- Image_url
- Background
- Tittle_english
- Tittle_japanese
- Tittle_synonyms
- Duration
- Premiered
- Aired

- Broadcast
- Anime_id
- Rating
- Scored_By

Siendo así todos estos atributos irrelevantes para nuestras problemáticas.

Además, algo super importante, se tomó la decisión de categorizar nuestra variable más importante (Score), la cual, corresponde a la puntuación del Anime, lo que es muy importante a la hora de realizar alguna comparación, debido a la relevancia que posee la puntuación sobre un anime. De por sí, el atributo es de tipo numérico los cuales van desde el 0 al 10, este atributo seguirá la siguiente regla para la problemática 1:

- Score (0 - 3): Puntuación mala (0)
- Score)3 – 7): Puntuación regular (1)
- Score)7 – 10): Puntuación buena (2)

También, a petición del estudio de televisión, filtramos los datos con las siguientes reglas para trabajar nuestra segunda problemática:

- duration_minutes =< 60
- episodes<=500 & episodes>0

Esto con la finalidad de no tener en cuenta animes de mucha duración, ni que posean una cantidad tan grande de episodios.

III. PROBLEMATICAS

El primer problema por abordar es el de clasificar la puntuación del anime en base a que si se encuentra en emisión o no. Como se explicó anteriormente, se tomará en cuenta la variable de puntuación (Score) debido que, es el dato relevante a la hora de seleccionar un anime.

El segundo problema consta en encontrar grupos de anime con base en algunas variables del dataset y encontrar a los grupos que se junten en una puntuación buena (Score superior a 7).

IV. RESULTADOS

Para responder la primera problemática, utilizaremos 2 modelos de clasificación, SVM, el cual es modelo de clasificación supervisada y K-Means el cual es un algoritmo de clasificación no supervisada que agrupa objetos en k grupos basándose en sus características.

Predicción

	precision	recall	f1-score	support
0	0.00	0.00	0.00	64
1	0.99	1.00	1.00	7620
2	1.00	1.00	1.00	2909
accuracy			0.99	10593
macro avg	0.66	0.67	0.67	10593
weighted avg	0.99	0.99	0.99	10593

Validación

	precision	recall	f1-score	support
0	0.00	0.00	0.00	19
1	0.99	1.00	1.00	2548
2	1.00	1.00	1.00	964
accuracy			0.99	3531
macro avg	0.66	0.67	0.67	3531
weighted avg	0.99	0.99	0.99	3531

Ilustración 2 Resultado SVM Problemática 1

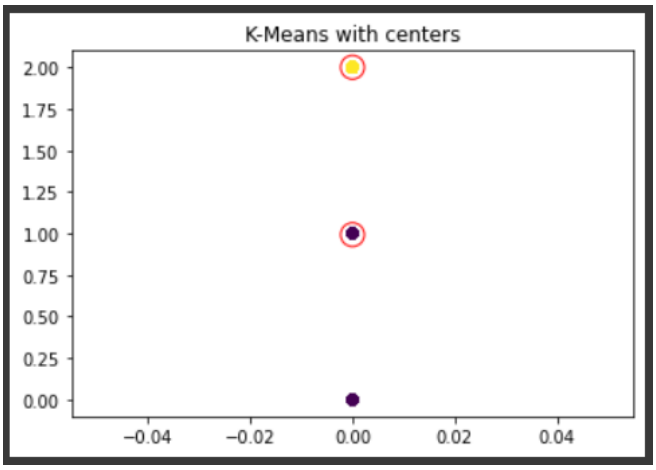

Ilustración 3 Resultado K-Means Problemática 1

Para responder la segunda problemática, utilizaremos dos algoritmos de clustering K-Means y DBSCAN

- Probaremos dos subconjuntos de datos (“score” y “episodes”) & (“score” y “duration_minutes”) para analizar como se agrupan los datos al utilizar información diferente.

- También Evaluaremos los clusters utilizando el enfoque visual y la Matriz de Similitud, para evaluar por proximidad.

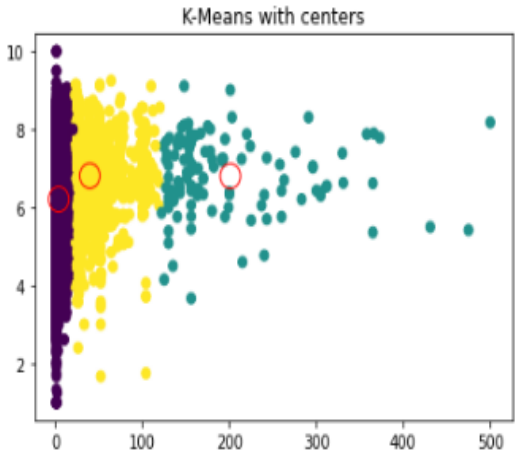

Ilustración 4 Resultado K-Means Problemática 2 "score vs episodes"

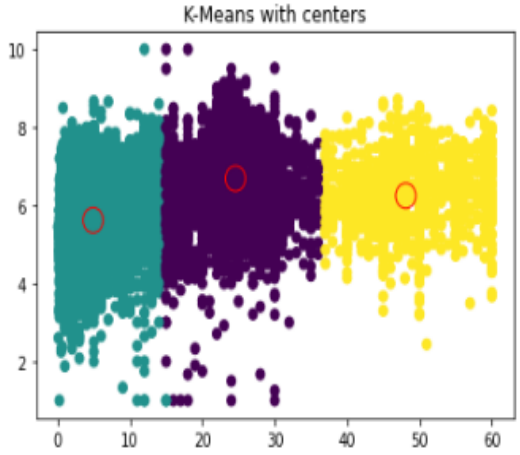

Ilustración 5 Resultado K-Means Problemática 2 "score vs duration_minutes"

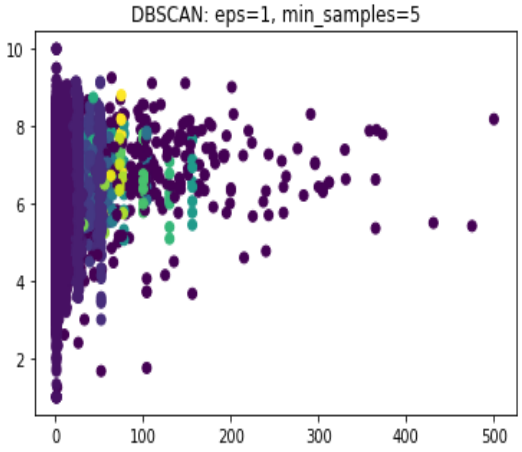

Ilustración 6 Resultado DBSCAN Problemática 2 "score vs episodes"

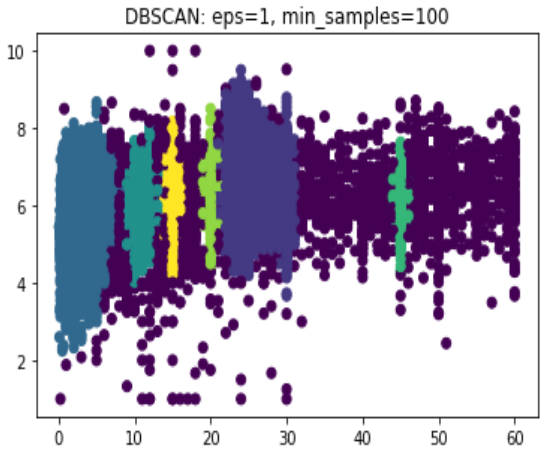

Ilustración 7 Resultado DBSCAN Problemática 2 "score vs duration_minutes"

V. PRUEBA DE LOS ALGORITMOS

Como mencionamos en la propuesta de la segunda problemática, aparte del enfoque visual, utilizaremos la Matriz de Similitud para comprobar la eficacia de los algoritmos de clustering utilizados.

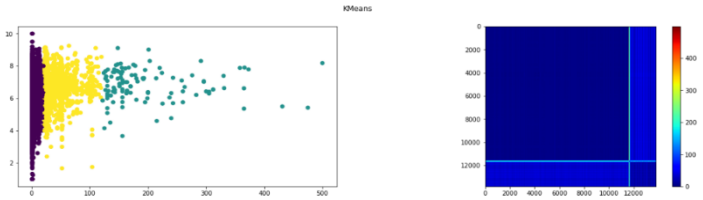

Ilustración 8 Resultado Matriz de Similitud para K-Means Problemática 2 "score vs episodes"

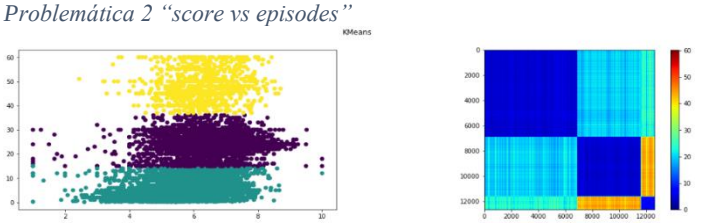

Ilustración 9 Resultado Matriz de Similitud para K-Means Problemática 2 "score vs duration_minutes"

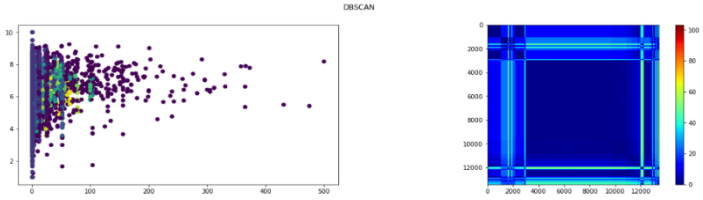

Ilustración 10 Resultado Matriz de Similitud para DBSCAN Problemática 2 "score vs episodes"

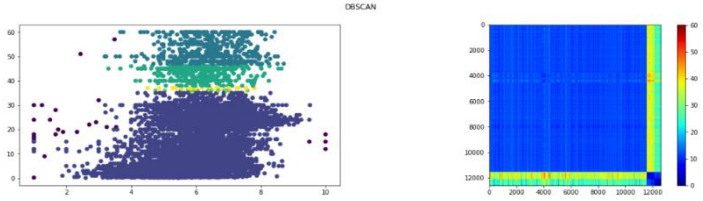

Ilustración 11 Resultado Matriz de Similitud para DBSCAN Problemática 2 "score vs duration_minutes"

VI. ANALISIS DE RESULTADOS

Como podemos observar en la ilustración 2, respecto a la precisión, el modelo tiene un 100% de probabilidad de acertar cuando da como resultado de la clasificación de la puntuación sea Buena, ósea, mayor a 7 y un 99% de probabilidad de acertar para la puntuación Regular, Score entre 3 y 7, sin embargo, es de un 0% en la puntuación Mala, menor que 3. Por otra parte, respecto al recall, vemos que fue capaz de clasificar el 100% de los datos en las puntuaciones Buenas y Regulares, no así, en las puntuaciones Malas, con un 0%. En el f1-score, al igual que el recall, las puntuaciones Buenas y Regulares son capaces de acertar en un 100%, no así en las puntuaciones Malas con un 0%. Finalmente, en cuanto al accuracy, el modelo fue capaz de predecir de forma correcta el 99% de los datos.

En cambio, en la ilustración 3, podemos observar el resultado del algoritmo de K-Means el cual falla, debido a que este algoritmo posee algunas desventajas, como la elección del k, nosotros decidimos cual usar y la posibilidad de que se cometa un error es bastante alta, igualmente este algoritmo es muy sensible a los outliers, los casos extremos hacen que el cluster se vea muy afectado.

Al fijarnos en los resultados de las gráficas, en la ilustración 4, contamos con 3 clusters definidos, los primeros dos, sin embargo, son de nuestro interés debido a que contienen los datos que mayor puntuación (Score) poseen. En la ilustración 5, podemos ver los resultados de comparar las columnas “score” con “duration_minutes” (duración en minutos), también contamos con 3 clusters claramente definidos y en este caso, nos interesa el del medio, debido a que contiene los valores con mayor score. En la ilustración 6 cambiamos el algoritmo de clustering por DBSCAN para graficar los mismos datos, sin embargo, los clusters no se encuentran tan definidos y considera a gran parte de los datos como “ruido”. Por último, al fijarnos en la ilustración 7 observamos que DBSCAN nos detecta una cantidad superior de agrupaciones de datos al utilizar un min_samples tan alto, esto hace que los puntos requieran más “vecinos” para poder ser considerados y no tomados como “ruido”.

Al analizar la prueba de los algoritmos con la Matriz de Similitud, podemos observar que el algoritmo de clustering K-Means, nos es más útil para los datos, sobre todo para agrupar los datos de “score vs duration_minutes” debido a que este no posee una gran cantidad de outliers y podemos ver a los datos claramente definidos en los clusters, por otra parte, DBSCAN nos puede ser un poco más útil agrupando los datos de “score vs episodes” debido a que se pueden visualizar más outliers, sin embargo, sigue siendo mejor utilizar K-Means para una clara distinción entre los datos.

Para concluir, teniendo en cuenta los resultados de la segunda propuesta experimental, consideraremos aquellos animes que se encuentren en un rango de duración en minutos de 20 a 30 minutos y con una cantidad de episodios que se encuentren entre 1 y 50.

VII. CONCLUSION

Podemos decir que el desarrollo del proyecto fue compleja, debido principalmente a la dimensionalidad de los datos, y las dificultados presentadas en la interpretación del comportamiento de las variables con valores vacíos o inexistentes. Para el futuro si se quiere abordar esta problemática, es necesaria la utilización de otros conjuntos de datos para complementar de mejor manera que estadísticas son necesarias para poder focalizar las mejores puntuaciones de los animes en base a más características. También se podrían haber empleado redes neuronales o reglas de asociación para simplificar el comportamiento de los datos.