PRÁCTICA DE APRENDIZAJE AUTÓNOMO (APA)

Gross Prediction in Conventional and Social Media Movies

2020-2021

Carolina Middel
Federico Rubinstein

ÍNDICE

1. Introducción

El objetivo de esta práctica consiste en desarrollar un modelo de regresión o clasificación para resolver un problema obtenido de los repositorios recomendados de datasets.

Después de varias ojeadas a las distintas páginas, nos hemos decantado por la web de UCI machine learning para escoger el dataset de nuestro proyecto porque está de entrada ya nos marca para qué tarea específica es, el tipo de atributos de los que consta y el número de instancias y atributos.

En este repositorio hemos encontrado datasets de muchos temas distintos pero finalmente nos hemos decantado por uno de películas de 2014 y 2015, ya que ambos somos grandes cinéfilos.

Una vez escogido el dataset y haber observado y analizado (el análisis se encuentra en el siguiente apartado), hemos decidido centrarnos en regresión lineal e intentar predecir los beneficios de las películas. Además, este dataset nos parece muy interesante ya que cuenta con datos de redes sociales y sentimos curiosidad sobre cuánto pueden influir estos valores en la predicción. Ya que hoy en día, el uso de las redes sociales se ha incrementado ampliamente y tiene mucha importancia la opinión de los usuarios sobre las películas.

El dataset incluye información de distintas redes sociales como Twitter y Youtube, y también de webs como IMDB. Y cumple con todas las restricciones marcadas ya que tiene variables tanto numéricas como categóricas, no está generado sintéticamente ni viene preprocesado, contiene missing values, además cuenta con más de 10 atributos y más de 200 instancias.

Una vez hemos escogido el dataset, hemos hecho un análisis y preprocesamiento de los datos, para después escoger entre varios modelos de regresión el que mejores resultados nos aportaba.

Haciendo una búsqueda previa sobre el dataset que hemos elegido, hemos encontrado un proyecto similar al que queríamos hacer, escrito en R (ver la primera referencia). Este proyecto ha conseguido una R2 de 66.01%, mientras que el modelo que mejores resultados nos ha dado a nosotros ha conseguido una R2 de 67,5078%, por lo que podríamos decir que nuestro modelo es ligeramente mejor.

2. Descripción de los Datos

El dataset con el que trabajaremos consta de 14 variables y 231 ejemplos de películas. A continuación definiremos brevemente la información que aporta cada variable:

- **Movie:** El nombre
- **Year:** El año en que salió
- **Ratings:** La valoración de la película en la plataforma IMDB
- **Genre:** Género de la película
- **Gross:** Ingresos brutos de taquilla de una película en dólares estadounidenses
- **Budget:** Presupuesto para producir la película
- **Screens:** Proyecciones de la película
- **Sequel:** Si es la secuela de otra película y en qué posición está de la lista de secuelas. Si es 1 significa que es la primera
- **Sentiment:** Sentimientos positivos o negativos de la audiencia a través de Twitter
- **Views:** Visualizaciones de la película
- **Likes:** Likes que ha recibido la película en redes sociales
- **Dislikes:** Dislikes que ha recibido la película en redes sociales
- **Comments:** Número de comentarios que ha recibido la película en redes sociales
- **Aggregate Followers:** Número total de seguidores que tiene la película en las redes sociales

3. Análisis de los Datos

Para empezar el trabajo, lo primero que hemos hecho es analizar los datos, este análisis lo hemos hecho utilizando el código de analisis_datos.ipynb

3.1. Visualización de las variables

Lo primero que hemos hecho es comprobar que el problema que queremos hacer sea factible. Para ello hemos buscado relaciones entre las variables y hemos intentado visualizar las relaciones que tienen cada variable con las variables Gross y la variable Ratings, que son las dos variables que queríamos predecir inicialmente.

Para esto, importamos el dataset y creamos un dataframe de pandas, con el que hemos visualizado las siguientes relaciones, mediante un scatterplot:

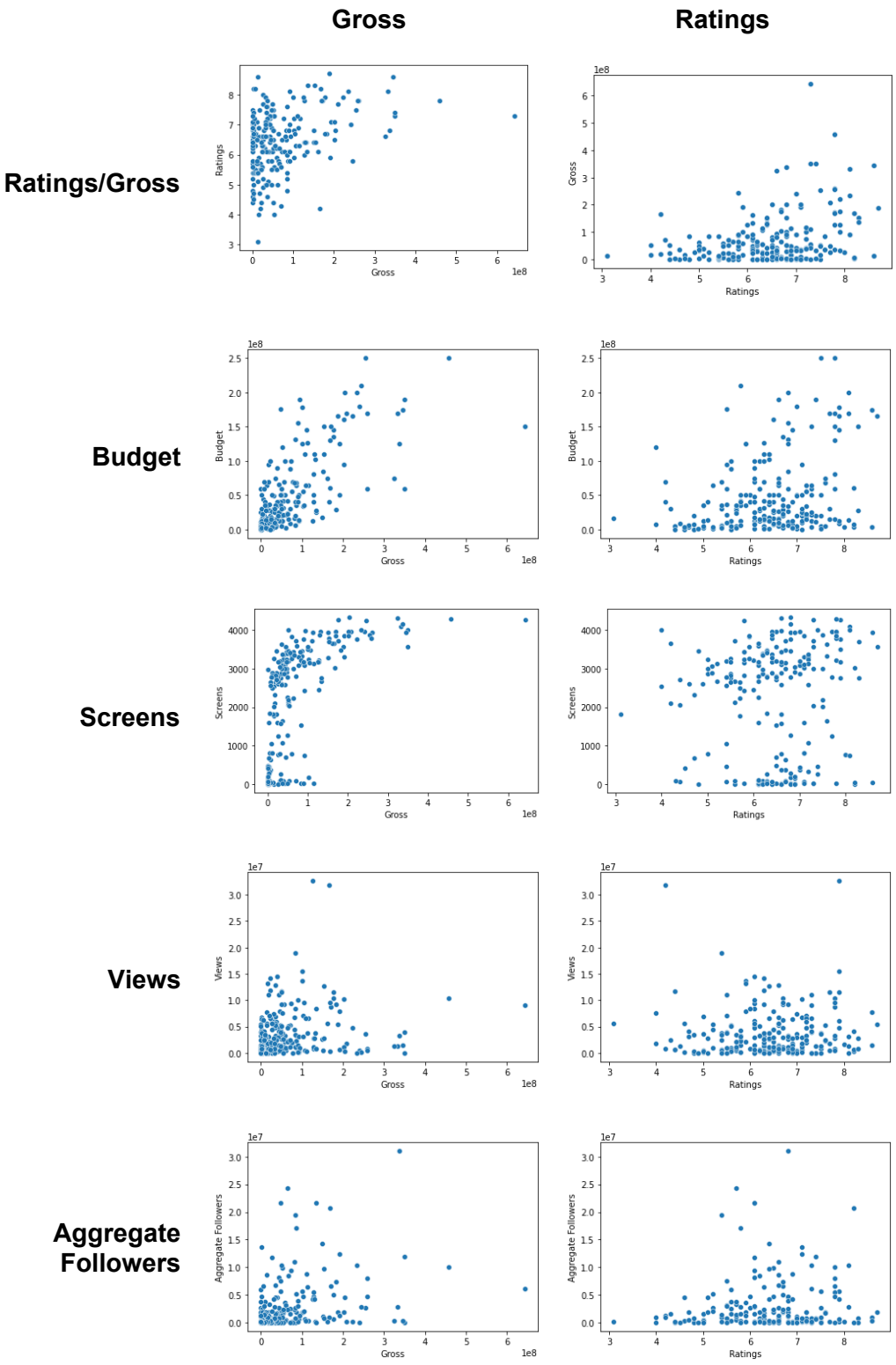


Figura 1: Relación lineal entre las variables y el target

También hemos hecho un pairplot utilizando la librería seaborn para ver más relaciones usando la variable Gross como hue.

3.2. Información sobre las variables

Una vez vistas estas relaciones hemos investigado un poco más sobre qué datos nos ofrece cada variable. Para ello, hemos aprovechado las funciones que nos ofrece pandas y numpy para obtener los tipos de cada variables:

	Type	Nulls	Nulls (%)	Unique Values
Movie	object	0	0.00	231
Year	int64	0	0.00	2
Ratings	float64	0	0.00	45
Genre	int64	0	0.00	11
Gross	int64	0	0.00	215
Budget	float64	1	0.43	105
Screens	float64	10	4.33	201
Sequel	int64	0	0.00	7
Sentiment	int64	0	0.00	36
Views	int64	0	0.00	231
Likes	int64	0	0.00	227
Dislikes	int64	0	0.00	203
Comments	int64	0	0.00	213
Aggregate Followers	float64	35	15.15	191

Figura 2: Características de nuestros atributos

También hemos representado la correlación entre cada variable para ver si era factible resolver el problema de predicción que queríamos hacer.

	Ratings	Gross	Budget	Screens	Sequel	Views	Likes	Dislikes	Comments	Aggregate Followers
Ratings	1.000000	0.342204	0.288157	0.057625	0.105701	0.011710	0.073824	-0.187422	0.015679	0.078545
Gross	0.342204	1.000000	0.719839	0.586447	0.423711	0.176363	0.110432	0.161536	0.125960	0.301808
Budget	0.288157	0.719839	1.000000	0.595684	0.464733	0.114708	0.011701	0.096888	0.090559	0.168874
Screens	0.057625	0.586447	0.595684	1.000000	0.267456	0.256515	0.173473	0.268176	0.213039	0.210822
Sequel	0.105701	0.423711	0.464733	0.267456	1.000000	-0.042763	-0.036089	-0.059792	-0.069333	0.228649
Views	0.011710	0.176363	0.114708	0.256515	-0.042763	1.000000	0.677175	0.776105	0.710507	0.155044
Likes	0.073824	0.110432	0.011701	0.173473	-0.036089	0.677175	1.000000	0.470645	0.917492	0.078575
Dislikes	-0.187422	0.161536	0.096888	0.268176	-0.059792	0.776105	0.470645	1.000000	0.579966	0.052877
Comments	0.015679	0.125960	0.090559	0.213039	-0.069333	0.710507	0.917492	0.579966	1.000000	0.034332
Aggregate Followers	0.078545	0.301808	0.168874	0.210822	0.228649	0.155044	0.078575	0.052877	0.034332	1.000000

Figura 3: Correlación de atributos 1

Figura 4: Correlación de atributos 2

Viendo estos resultados, finalmente nos decantamos por las fuertes relaciones entre Budget y Gross y Screens y Gross. Y por lo tanto decidimos predecir los Gross de las películas.

A partir de esta figura 4, también nos hemos dado cuenta de que entre los atributos LIkes, Dislikes i Comments, existe una fuerte correlación. Hemos decidido eliminarlas ya que esto puede significar que las variables son redundantes.

4. Preprocesamiento de los datos

Para preprocesar los datos, hemos desarrollado el código preprocesado.py
Hemos podido hacer este preprocesado gracias al análisis previo de los datos y a la prueba y error que hemos hecho con los modelos.

4.1. Procesamiento inicial, eliminación de atributos

Como ya hemos visto anteriormente en el apartado de descripción de datos en la tabla de la figura 2, tenemos algunos datos que no usaremos, ya que no nos aportan información o esta no es válida.

Hemos decidido no tener en cuenta el título de la película y tampoco el año ya que el primero no aporta información necesaria para el estudio y el dato del año no aporta nada ya que la base de datos sólo dispone de dos años diferentes. Como se supone que no hay una súper inflación de un año para el otro, por lo tanto también la eliminamos.

Además como se ha comentado en el apartado anterior, debido a la alta correlación entre likes, dislikes i comments, hemos optado por eliminar también estos atributos ya que se vuelven redundantes.

4.2. Procesamiento de missing values

	Nulls	Nulls (%)
Movie	0	0.00
Year	0	0.00
Ratings	0	0.00
Genre	0	0.00
Gross	0	0.00
Budget	1	0.43
Screens	10	4.33
Sequel	0	0.00
Sentiment	0	0.00
Views	0	0.00
Likes	0	0.00
Dislikes	0	0.00
Comments	0	0.00
Aggregate Followers	35	15.15

Figura 5: Missing values de los distintos atributos

En la tabla que hemos mostrado anteriormente en la figura 2 también contábamos con la información de los valores nulls (NaN), es decir los missing values.

Primeramente, nuestra idea ha sido borrar las filas en las que encontramos algún missing value ya que hay poco porcentaje de estos, pero como nuestro dataset es bastante limitado respecto a las instancias finalmente hemos optado por imputar estos valores.

4.3. Procesamiento de variables categóricas

Hemos decidido usar la variable *Genre* como variable categórica. Este atributo, es obvio que aunque esté representado como un int, cada uno de estos representa un género, y por lo tanto deberá ser una variable categórica. También hemos probado de categorizar el atributo Sequel, ya que habíamos pensado que de esta forma representa el tipo de secuela/película que estamos viendo, si es secuela (+ de 1) o no. Pero nos dimos cuenta que esto nos generaba *course of dimensionality*, por lo que finalmente hemos optado por dejarlo como un atributo numérico.

4.4. Estandarización

Llegamos a la última fase pero no la menos importante del preprocesamiento de datos. Hemos decidido estandarizar los valores en vez de normalizar ya que hemos probado ambos, y hemos obtenido mejores resultados con la estandarización.

Para realizarla hemos usado la función StandardScaler() de sklearn.preprocessing.

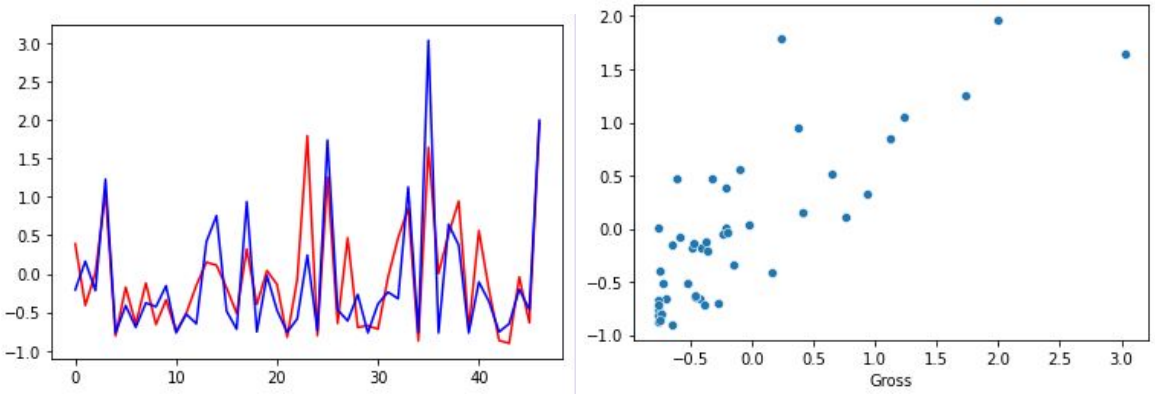
5. Modelos lineales

Hemos decidido usar más de 3 modelos para predecir los datos ya que primeramente los resultados no eran muy buenos y queríamos obtener cuanta más información mejor.

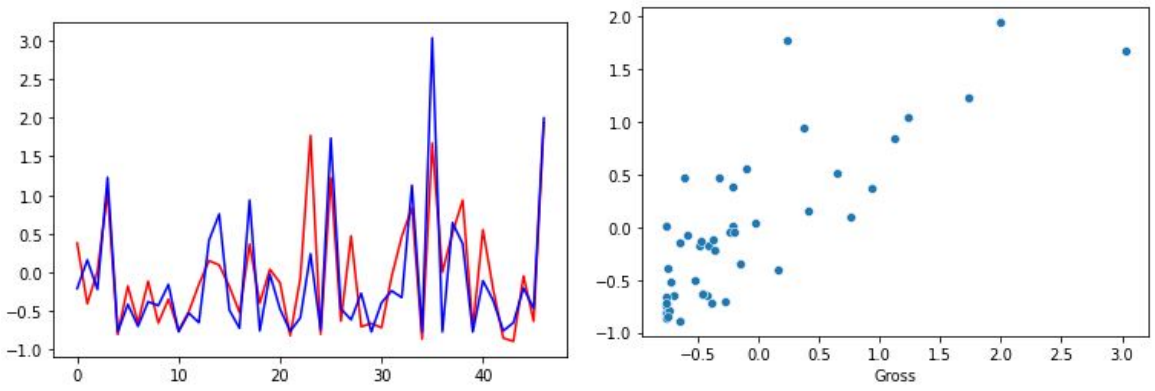
De esta manera hemos podido comparar más resultados y decantarnos por uno de ellos de forma más segura.

En los siguientes apartados mostraremos únicamente la representación de la predicción obtenida mediante el test vs los valores reales del test. En las primeras gráficas se muestra un plot en el que en rojo está representada la predicción y en azul están representados los datos reales. En cambio en el gráfico de la derecha mostramos mediante un scatterplot la relación entre los datos reales de Gross del test (eje x), y los valores de este predichos (eje y).

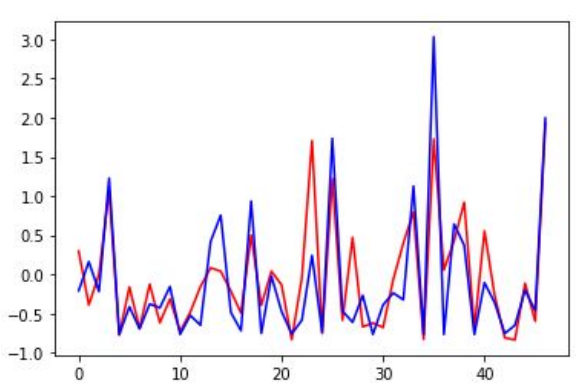
5.1. Linear Regression

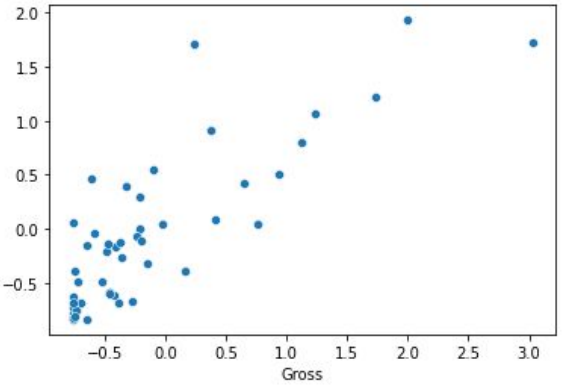


5.2. Ridge Regression

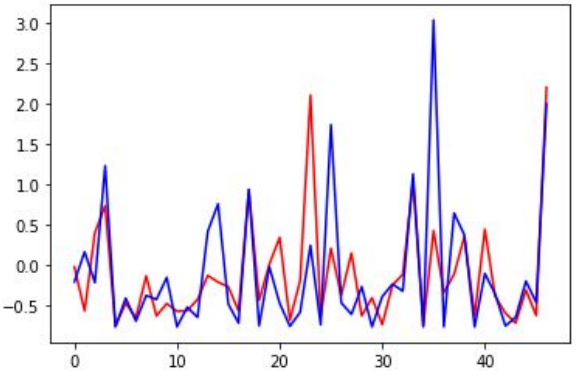


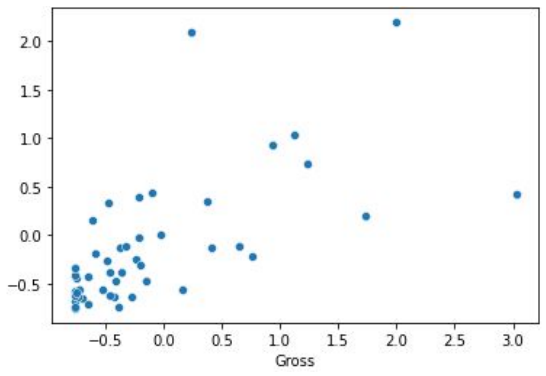
5.3. LASSO Regression



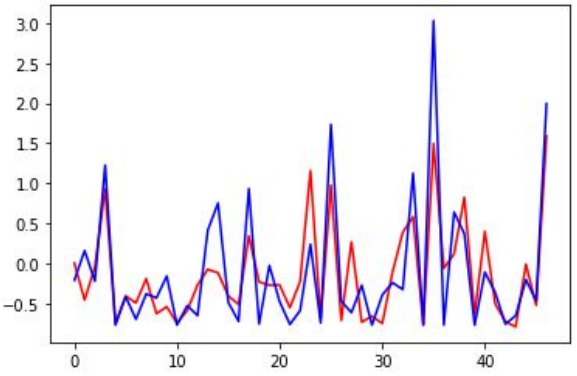


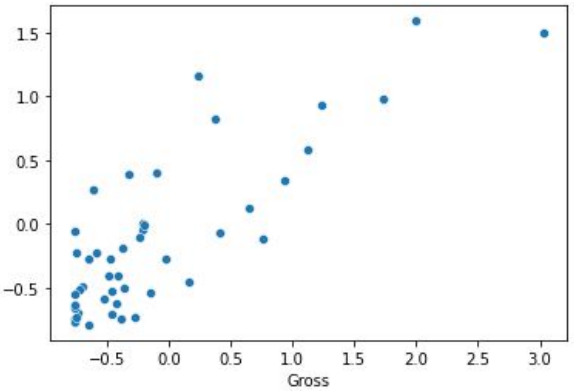
5.4. k-Nearest-Neighbours Regression





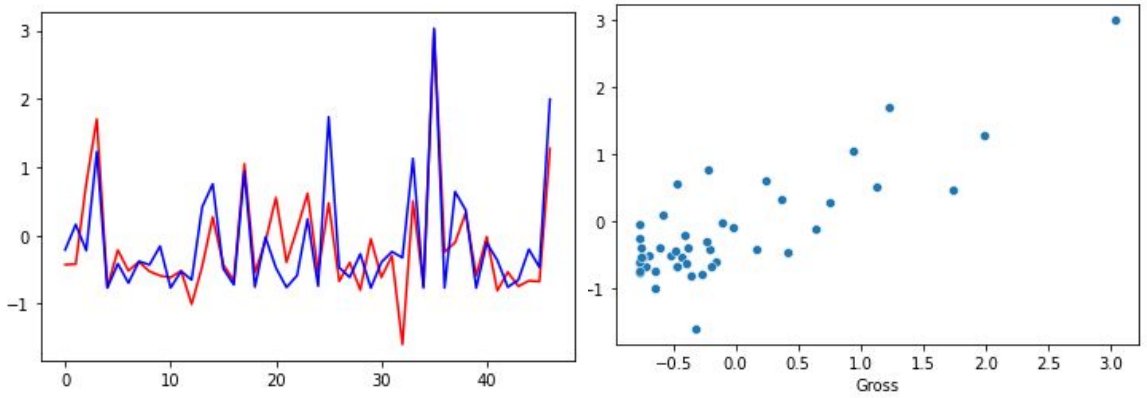
5.5. Linear SVM Regression



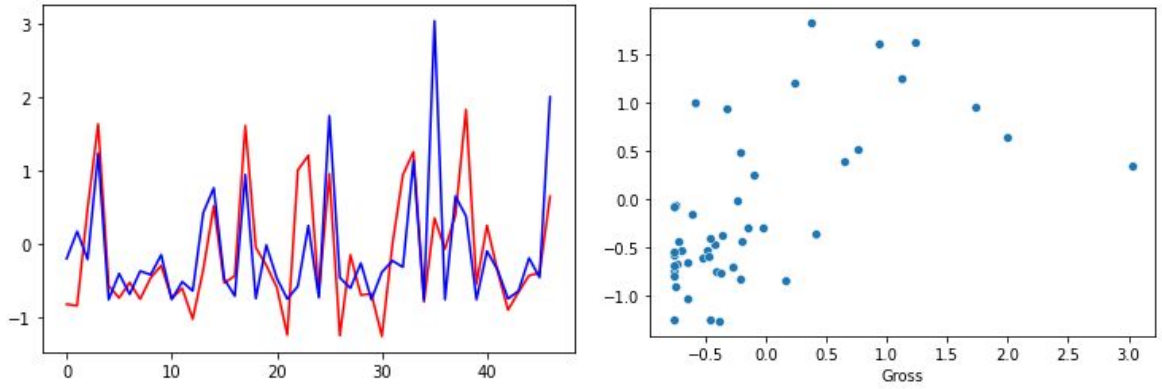


6. Modelos no lineales

6.1. MLP Regression



6.2. SVM (Kernel RBF)



7. Modelo Escogido

Finalmente, una vez estudiados todos los modelos, podemos compararlos con la siguiente tabla:

	R2	MSE	median_absolute_error	mean_absolute_error
LR	0.646537	0.235975	0.237777	0.350423
RIDGE_CV	0.652415	0.232051	0.236851	0.348708
LASSO_CV	0.671146	0.219546	0.225546	0.335961
KNN	0.409489	0.394231	0.203816	0.379763
LinearSVR	0.675078	0.216921	0.23947	0.352277
MLP	0.639187	0.240882	0.216549	0.363765
RBF-SVR	0.234966	0.510744	0.331211	0.493897

Figura 6: Comparación de los resultados de todos los modelos

Esta tabla nos indica que el mejor modelo con el que hemos experimentado ha sido el Linear SVM, con un R2 de 0.675078, seguido muy de cerca por el R2 del LASSO, de 0.671146.

Lo que debería hacerse en este punto es hacer una estimación honesta de su error con unos datos de test nuevos. Aquí es donde nos hemos topado con un problema con nuestro dataset; nuestro dataset no es lo suficientemente grande como para hacer una nueva partición de los datos para testear el modelo, conservando datos suficientes para entrenar y validar los modelos.

8. Conclusiones

8.1. Logros, fallos y conclusiones

Una vez finalizado el proyecto podemos evaluar las carencias y los logros de este. Como logro queremos remarcar el buen uso del código y la exploración de los modelos, ya que hemos trabajado con casi todos ellos para observar las mejoras de nuestros resultados.

Una de las carencias/fallos más importantes que consideramos que tenemos es la elección del dataset. Primeramente nos guiamos sobre si este era interesante y cubría todos los requisitos, pero nos hemos encontrado con un dataset muy pequeño y muy difícil de trabajar para predecir nuestro objetivo. A pesar de esto, pensamos que lo hemos podido resolver bien separando nuestro dataset en solo train y test (y no val), para contar con más valores para el estudio.

Además la estadística de los resultados generados siempre se encuentra en valores buenos. Es decir r2 lo encontramos encima del 0.65 aprox. Y los errores resultantes son bajos.

8.2. Extensiones y limitaciones

Como extensiones para el proyecto nos gustaría haber podido tratar con un dataset mayor, buscando otros con los que poder fusionar el nuestro por ejemplo. Además así poder tratar cosas que en este no hemos tratado por miedo de consumir nuestro dataset como por ejemplo los outliers.
Otra extensión que nos gustaría haber probado, es marcar los atributos sequel como variables categóricas, como pensamos al principio, pero adaptar los datos para que estos sean 1 si no pertenecen a una secuela y 2 si en caso contrario, sí que pertenecen a una, en vez de tener en cuenta la posición de la película en esta.

9. Referencias

GitHub. 2021. *Gaurangaurang/Conventional-And-Social-Media-Movies*. [online] Available at: <https://github.com/gaurangaurang/Conventional-and-Social-Media-Movies > [Accessed December 2020].

Ijirset.com. 2021. [online] Available at: <https://www.ijirset.com/upload/2017/april/192__IJIRSET_BABITA_PN.pdf > [Accessed December 2020].

Computer.org. 2021. *CSDL | IEEE Computer Society*. [online] Available at: <https://www.computer.org/csdl/proceedings-article/smartcity/2015/1893a273/12OmNyKrH6S > [Accessed January 2021].

Raschka, S. and Mirjalili, V., n.d. *Python Machine Learning - Second Edition*.