Universidad Francisco Marroquín
Facultad de Ciencias Económicas
Curso: Machine Learning Models
Nombre: Juan Fernando González
Fecha: 18-02-2020

Reporte

Introducción / Data

Google Play Store App Data set

El propósito de este reporte es poder responder a las siguientes preguntas: ¿El Rating de una aplicación, depende de las descargas de esta?, ¿Las descargas de una aplicación, dependen del precio?, y por último, ¿El rating de una aplicación, depende del precio del app?; así como también el de analizar y procesar los datos provenientes de la ***Google Play Store App***.

El *dataset* de la Google Play Store App fue adquirido desde la página *kaggle.com*, la cual proporciona varias fuentes de datos para las personas que deseen hacer un análisis de estos. Lavanya Gupta, es la autora de este *dataset*, ella fue la encargada de buscar y almacenar los datos de la Google Play Store App. Los datos fueron recopilados mediante la herramienta de *scraping,* la cual consiste en recaudar datos provenientes de una página web de manera automatizada por medio de un algoritmo especial por cada sitio. Este *dataset* contiene información sobre más de diez mil aplicaciones del mercado de Android del año 2019. Cada aplicación tiene un valor para una categoría, rating, precio, tamaño, entre otras.

Las variables (*features* / *columnas*) que podremos encontrar en el *dataset* son las siguientes:

1. *App*: El nombre de la aplicación.
2. *Category*: La categoría a donde pertenece la aplicación.
3. *Rating*: Contiene el promedio del rating que los usuarios otorgaron a la aplicación.
4. *Reviews*: Numero de comentarios de usuarios a la aplicación.
5. *Size*: El tamaño de almacenamiento de la aplicación.
6. *Installs*: Numero de descargas de la aplicación.
7. *Type*: Tipo de la aplicación, pagada o gratis.
8. *Price*: El precio de la aplicación.
9. *Content Rating*: Grupo de edad al que se dirige la aplicación.
10. *Genres*: El género a la que pertenece la aplicación. Una app puede pertenecer a varios géneros.

11. *Last Update*: Dia en el que la app fue última vez actualizada en la Play Store.
12. *Current Ver*: La versión actual disponible de la aplicación en la Play Store.
13. *Android Ver*: Versión mínima de Android requerida para poder obtener la aplicación.

Por otro lado, hay 10,841 filas (*observations*), que son los valores por cada columna en el dataset. Entre estas 13 *features* y 10,841 *observations*, existen 1,487 datos nulos, es decir, que no tiene un valor, o que no se tiene información sobre ese campo en específico. Más adelante veremos cómo tratamos con estos valores nulos.

Dadas las preguntas que queremos resolver en este reporte, las variables dependientes serán las siguientes: La categoría (*Category*), el *rating*, y las descargas (Installs). Las variables independientes serán: El precio (*Price*), el tamaño de la app (*Size*), el tipo (*Type*), los *reviews* y el género (*Genres*). No tomaremos en cuenta la última vez que la aplicación fue actualizada, o la actual versión de la aplicación, ni tampoco la versión mínima de Android requerida para obtener la aplicación, ya que los datos son de un mismo año (2019), y las versiones o actualizaciones son diferentes por cada aplicación, y mucho dependen de los desarrolladores que hacen la app, por lo que no nos podrán servir de referencia o de punto de comparación para lo que queremos analizar.

La herramienta que se usará para la extracción, la exploración, visualización y el análisis de los datos será *Jupyter lab. Jupyter lab* es un entorno de desarrollo interactivo basado en la web para dispositivos portátiles, enfocado para poder trabajar código, datos y *jupyter notebooks*. En este caso, *jupyter notebook* será en donde manejaremos nuestro *dataset*. El lenguaje de programación que se usará para el manejo y acceso a los datos será *Pyhton*, el cual ya viene por defecto instalado en *Jupyter lab*. *Pyhton* y *Jupyter lab* nos darán las herramientas necesarias para poder trabajar sobre las preguntas, hacer análisis de los datos y visualizar los resultados. Por otro lado, Pyhton también nos proporciona una gran variedad de librerías que nos ayudaran a hacer visualizaciones y cálculos para poder interactuar con los datos.

Algunos de los conceptos de estadística que serán aplicados en este reporte son los siguiente: Histograma (distribución), Correlación, *Skewness*, Regresión lineal simple, Regresión lineal múltiple, media, desviación estándar, entre otros. Comencemos con el histograma, también llamado diagrama de dispersión, es una representación gráfica de una variable numérica o categórica, por lo regular, en forma de barras. En el eje *y* se representa la frecuencia, y en el eje x, los valores de la variable. Esto nos ayudara a poder ver de forma gráfica la distribución de una variable, visualizar que datos son más frecuente, cuales no tanto y cuales no lo son.

La correlación nos indica la fuerza y dirección de una relación lineal y proporcionalidad entre dos variables. Para que dos variables sean consideradas correlacionadas, el coeficiente de correlación debe de estar cercano a 1 o a -1. Si el coeficiente de correlación es cercano a 0, esto quiere decir que no hay relación entre las 2 variables.

La regresión lineal es un modelo matemático que nos permite aproximar la relación de dependencia entre una variable dependiente *y*, y una variable dependiente *x*, además de un término aleatorio. Este será uno de los métodos fundamentales para poder responder las preguntas.

Métodos

Los datos de la Google Play Store App estaban almacenados en un archivo *.csv,* con las herramientas de *Jupyter lab* y de *Python* es posible exportar los datos y manipularlos. La librería que nos ayudara con la exportación y manipulación de los datos del archivo es *Pandas*, una librería de Python que está enfocada a el análisis de datos y manejo de archivos. Llamaremos *dataframe* o *df* a la variable donde almacenaremos los datos del archivo *.csv*. Una vez ya exportados los datos, los podremos visualizar, ver los tipos de datos que tiene almacenada, ver los datos por columna, y podremos imputarlos como mejor nos convenga.

Luego de explorar el *dataframe*, se comenzará la limpieza de los datos. Existen varios tipos de manipulación de datos, en este trabajo se utilizaron tres, el de imputación por valores numéricos por valores categóricos, el de *One – Hot Encoding,* que consiste en volver las *observations* (filas) de una columna, a columnas, y son seleccionadas de manera binaria (0 o 1) y el de eliminación de datos nulos. Para la columna del tipo (*Type*) por ejemplo, utilice la imputación por valores numéricos, en este caso también binario, ya que solo existen 2 valores únicos en esa columna, el cero significa que la app es gratis y en caso contrario, un uno, para indicar que es pagada. Por otro lado, también realizo modificación de los valores de las columnas que son de tipo *string,* para quitarles signos (+, -, _, ,) y remplazar palabras por números, para que luego sean más fáciles de computar y comparar.

Una vez ya hecha la limpieza y la imputación de nuestros datos, se continuo a aplicar los métodos estadísticos y algoritmos matemáticos para visualización y análisis de los datos.

Como primer método, se empleó el histograma o diagrama de dispersión. En este caso, se utilizó la librería para visualización de datos de Python, *matplotlib*, esta librería nos permite hacer un el histograma de una variable con solo instanciar nuestro *dataframe* (*df)*, la *feature* (columna) que deseamos graficar, y la función para graficarlo (*plot*) con el

parámetro de histograma (*hist*), y sí es necesario, algunos otros parámetros para modificar la gráfica. A continuación, se podrá ver un ejemplo de cómo instanciar y graficar el histograma de una columna, en este caso del Precio:

$$df.Price.plot(kind = 'hist')$$

La correlación, la cual fue otro de los métodos empleados en el análisis, fue el método de *Kendall,* ya que este nos permitirá correlacionar nuestras variables ordinales o continuas, y verificar si estas tienen una relación monotónica. En otras palabras, a medida que aumenta el valor de una variable, también aumenta la otra variable, o al revés, si una variable aumenta su valor, el valor de la otra disminuye.
Esta, al igual que la anterior, nos la proveerá una librería de *Python,* en este caso la librería seaborn. Al nosotros instanciar el método que queremos utilizar, automáticamente esta librería nos hace un gráfico, en donde podrá nuestras variables numéricas o continuas se muestren en un mapa de calor con escala, donde 1 es lo más fuerte y 0 es lo más débil.

Por último, pero no menos importante, la regresión lineal siempre y regresión lineal múltiple. Estos métodos fueron los que me ayudaron a resolver las preguntas que se habían presentado en el inicio de este reporte, y escogí estos métodos ya que me ayudan a predecir el resultado de una variable dependiente sobre una variable independiente, así como la relación que estas tienen. *Seaborn* fue la librería que me proporciono la herramienta para poder relacionar mis *features.* Los parámetros que utilice para este método fueron mis variables dependientes y mis variables independientes, las cuales ya fueron descritas anteriormente.

La diferencia entre la regresión lineal simple y la regresión lineal múltiple es que hay una variable dependiente, y 2 o más variables independientes para predecir el resultado y la relación que existe entre las demás variables. Además, en este método en especial, se dividieron los datos en *training* y *test*. ¿Por qué? Porque en este caso, queremos que nuestro modelo se entre con datos ya existentes, y que prediga a partir de los datos de prueba, en este caso sobre el 30% de los datos. Aquí es donde también utilizaremos los datos que imputamos de manera *dummy* (*one – hot encoding*), por lo que haremos 2 predicciones, una con nuestros datos imputados de manera numérica entera, y otra con los datos imputados de manera *one – hot encoding*, y así poder comparar, cuál de los 2 métodos es mejor para la relación entre las *features*. Para esto también utilizaremos la media y la deviación estándar, para ayudarnos a saber que tan precisos son los modelos y que tan alejados están de los valores reales.

Resultados

A continuación, se presentarán los resultados obtenidos de los métodos aplicados al *dataset,* así como también el análisis y deducciones del mismo.

Los primeros resultados que obtuve fueron sacados de la información que me proporcionaban los histogramas, como también de la media o el valor máximo de la variable a analizar.

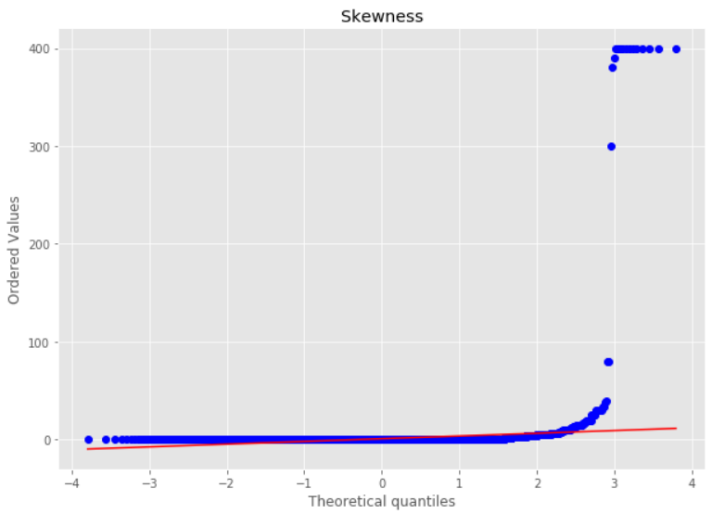
Histograma del Precio:

$$\boldsymbol{Media = \$0.96}$$

$$\boldsymbol{Max = \$400.00}$$

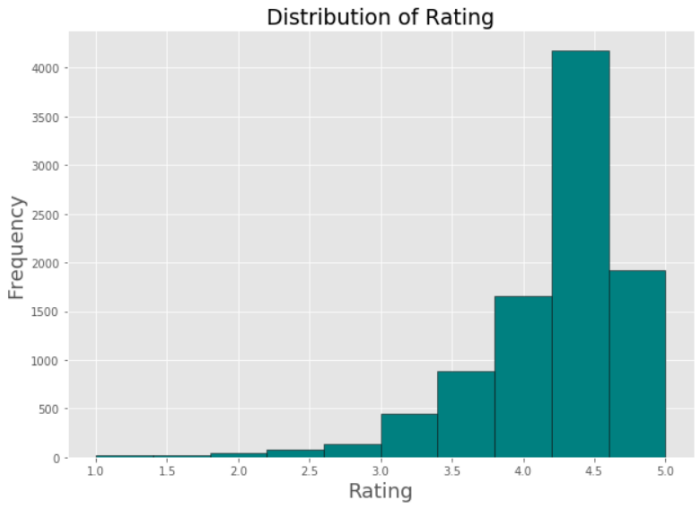
Como se puede observar en este histograma, la mayoría de las aplicaciones son gratis o son menores a los 50 dólares, sin embargo, se pueden observar unos *outliers* que son aplicaciones que tiene un precio muy por encima de la media de las apps. Para ser específicos, la aplicación *I'm rich* es el *outlier* con precio más alto en el *dataset*.

***Skewness* del Precio:**



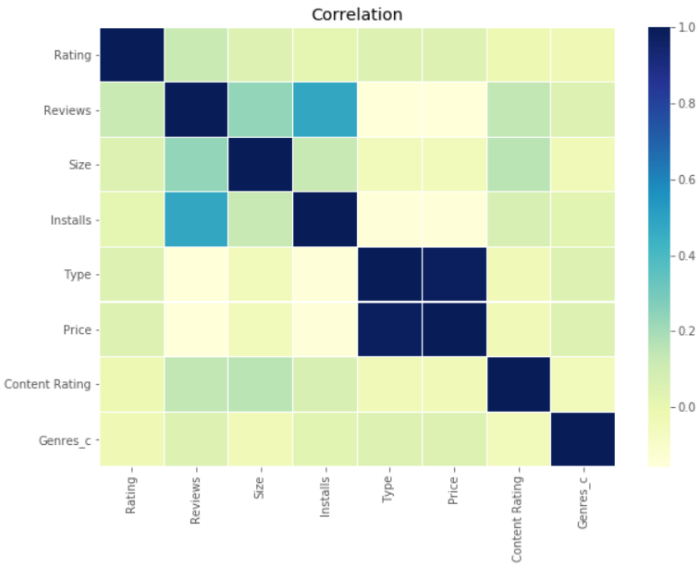
En base a la distribución con los outliers, se realizó una distribución de skewness, y como se puede ver en la gráfica arriba, la cola de la distribución está en la izquierda, lo que significa que la media de los precios se encuentra en precios bajos y luego, por el cuantil teórico 3 se empieza a notar una gran subida en el precio, lo que representa a los outliers. Por lo que la distribución del precio presenta un *left skew*.

Histograma del Rating:



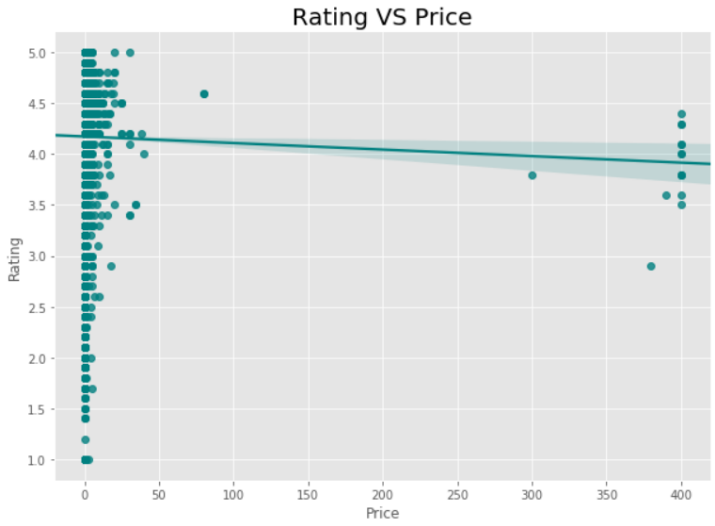
Al hacer el histograma para el *Rating*, pude observar que la mayoría de las aplicaciones tienen un rating entre 4.0 – 4.5, lo que es bastante bueno. Luego, saque la media de los datos del rating, y la media estimada del *rating* está en 4.19.

Correlación:



En el mapa de correlación podremos ver la fuerza de como cada variable numérica se relaciona con las demás. En este caso, las variables que tomaremos en cuenta para el análisis son las descargas, el rating y el precio. Sin embargo, la correlación nos indica que el rating no está relacionado con las descargas, y que las descargas tampoco esta relacionada con el precio, por lo que trataremos de comprobar o contrastar con las regresiones lineales más adelante.

Regresión lineal simple : *Rating vs Price*

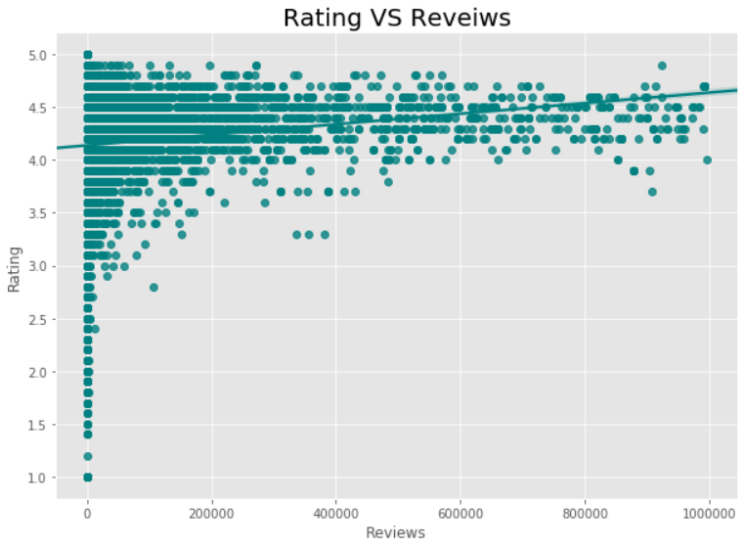


Como se puede notar en la regresión lineal entre el rating y el precio, los outliers parecen indicarnos que el precio si afecta de cierta manera al rating, pero es algo difícil de entender y de visualizar, por lo que realice una categorización de los precios por intervalos, y así poder agruparlos por la media de rating en ese intervalo.

	Price_Cat	Rating
0	1 Free $0	4.186288
1	2 Cheap (0–0.99)	4.300943
2	3 Not cheap (0.99–2.99)	4.292975
3	4 Normal (2.99–4.99)	4.250318
4	5 Expensive (4.99–14.99)	4.269149
5	6 Too expensive (14.99–29.99)	4.252000
6	7 Very expensive (29.99–400.0)	3.923810

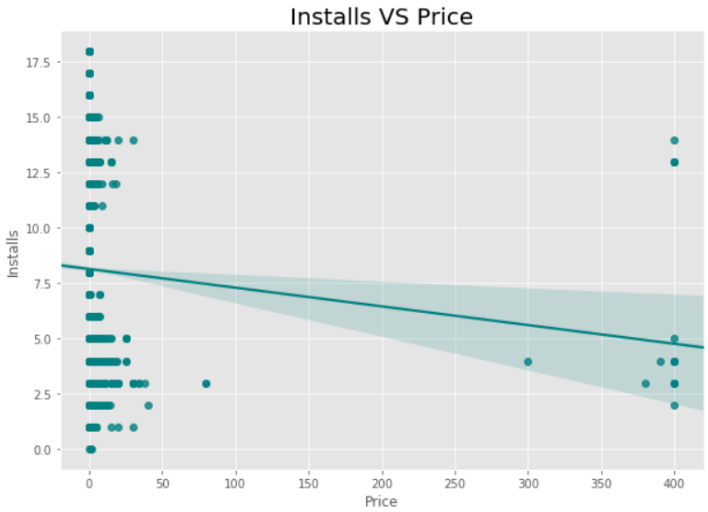
Dada esta nueva visualización de los datos, podemos ver que el rating no varía mucho con los precios, ya que incluso las aplicaciones mas caras son valoradas con un buen rating, en este caso, el ultimo intervalo tiene una media de rating del 3.9, por lo que solo esta 0.2 puntos por debajo de la media del rating.

Regresión lineal simple: *Rating vs Reviews*



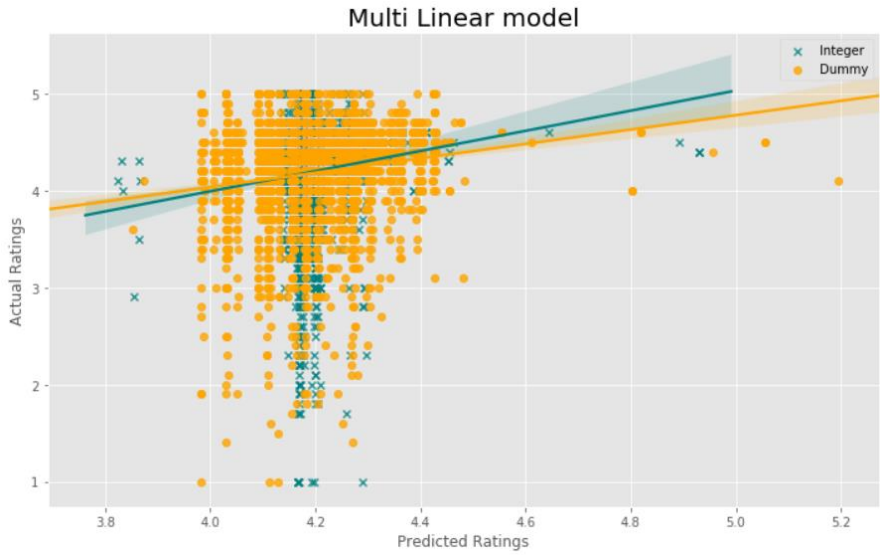
En este caso, la regresión lineal entre el *rating* y los *reviews* es más clara, ya que se puede identificar que a medida que sube la cantidad de reviews en el app, se incrementa el rating, por lo que estas 2 variables sí están relacionadas de manera positiva.

Regresión lineal simple: *Installs vs Price*



La relación entre las *features* del precio y las descargas, a excepción de los outliers, es negativa, ya que a medida que sube el precio, decrementa el número de descargas. Por otro lado, si contamos también a los outliers, el resultado de la regresión lineal sigue siendo el mismo.

Regresión lineal múltiple: *Integer encoding vs Dummy encoding*



Por último, el modelo de regresión lineal múltiple que compara el *Integer encoding* y el *Dummy encoding* para predecir el rating de un app. A simple vista se puede ver que ambas predicciones son bastante parecidas, e incluso las 2 predicen con bastante exactitud el rating de una aplicación, por lo que haremos el cálculo de la media del rating, del *integer encoding* y los dummy *encoding* en los datos predichos, y por último, la desviación estándar de la predicción de ambos métodos.

```
Actual mean of population:4.191837606837606
Integer encoding(mean) :4.191845591333759
Dummy encoding(mean) :4.199235078806515
Integer encoding(std) :0.047355161612185045
Dummy encoding(std) :0.10746292600823242
```

Dicho lo anterior, con estas estadísticas podemos comprobar que ambas son bastante precisas al momento de predecir, sin embargo, el dummy encoding tiene una desviación estándar más alta que el integer encoding.

Conclusiones

Luego del proceso de exploración y análisis de los datos ya antes mencionados, puedo concluir y responder con certeza a mis preguntas del inicio. El rating de una aplicación no depende del numero de descargas de la misma. Tal y como se demostró en los resultados, la variable que mas afecta la calificación del rating es el numero de reviews.

Esto quiere decir que cuando una aplicación tiene un gran numero de reviews, el rating será alto, y en caso contrario, donde una aplicación tenga pocos reviews, el rating será bajo. Como siguiente conclusión, se puede decir que el numero de descargas si se ve afectado por el precio del app. Ya sea tomando o no en cuenta los outliers, podemos concluir que cuando un app tiene un precio muy elevado, entonces no tendrá muchas descargas. Y por último, también se concluyo que el rating de un aplicación no depende del precio. Como se demostró en la tabla de resultados, tanto las aplicaciones gratis como las pagadas tenían un rating parecido y bastante cercano a la media de rating de las demás aplicaciones, por lo que no se puede decir que el precio tiene un efecto en el rating. Durante el proceso de análisis de los datos, me encontré con problemas en mi imputación y en mi limpieza de datos. Las variables categóricas como la categoría y el genero no las estaba manejando del todo bien, ya que estaba tratando de aplicar el método numérico a mis variables categóricas de un modo incorrecto, y fui ahí donde el *One – Hot Econding* me fue de gran ayuda en la hora de hacer imputación y análisis de mis datos, por lo cual lo recomiendo para variables categóricas con muchas observaciones. Y ya para concluir, del lado del mercado, si un usuario quiere hacer una app pagada y con un precio alto, debe de ser lo suficientemente buena para que el usuario final la descargue y pague el precio. Además, si algún usuario busca que su app tenga buen rating, debe preocuparse por el número de reviews que su aplicación tenga.

(Creech, 2019)
(Brownlee, 2018)
(Magiya, 2019)
(Seaborn, 2012)

Apéndice

Diccionarios adicionales del *dataset*:

```
Content Rating Dictionary
{'Everyone': 0,
 'Teen': 1,
 'Everyone 10+': 2,
 'Mature 17+': 3,
 'Adults only 18+': 4,
 'Unrated': 5}
```

```
Genres Dictionary
{'Art & Design': 0,
 'Art & Design;Pretend Play': 1,
 'Art & Design;Creativity': 2,
 'Auto & Vehicles': 3,
 'Beauty': 4,
 'Books & Reference': 5,
 'Business': 6,
 'Comics': 7,
```

Referencias

Brownlee, J. (2018, 05 11). *Machine Learning Maestry*. From https://machinelearningmastery.com/a-gentle-introduction-to-normality-tests-in-python/

Creech, S. (2019). *Statistically SIgnificant* . From https://www.statisticallysignificantconsulting.com/RegressionAnalysis.htm

Magiya, J. (2019, 06 16). *Medium*. From Medium: https://towardsdatascience.com/kendall-rank-correlation-explained-dee01d99c535

Seaborn. (2012). *Seaborn*. From https://seaborn.pydata.org/generated/seaborn.lmplot.html