Google Play Store downloads predictor

Manuel Torres Cid & Eloi Pagès Vives

Índice

Introducción

El objetivo de este proyecto es desarrollar una herramienta capaz de estimar el número de descargas que recibirá una aplicación antes de ser pública en el Google Play Store, para ello aplicaremos los conocimientos adquiridos en el curso Aprendizaje Automático 1 y obtendremos los datos de un dataset obtenido de la popular plataforma de ciencia de datos, Kaggle (link), el cual se trata de un *scraper* de la propia página web, que recoge los datos hasta junio de 2021.

La principal motivación de este proyecto se trata de aportar una herramienta que ayude a los desarrolladores *indie* y empresas a entender el beneficio esperado que pueden obtener antes de comenzar el desarrollo, se ha de resaltar, que una de las variables que tendremos en nuestro *dataset* es el precio de la aplicación. Existen diversos estudios estadísticos que estiman el beneficio que se puede obtener de una aplicación en función de pocos datos, como el tipo de aplicación que se quiere desarrollar (juego de aventuras, red social, etc…), la metodología con la cual se pretende obtener beneficios (microtransacciones, pago único, anuncios). Usaremos estos estudios estadísticos para corroborar que los resultados obtenidos de nuestra herramienta son coherentes y a la hora tomar varias decisiones (por ejemplo, eliminaremos aquellos juegos que tengan menos de 20 *reviews* ya que la mayoría de estos estudios eliminan dichos datos también).

Dividiremos el proyecto en varias secciones con el objetivo de procesar nuestros datos de la forma más adecuada, construir los diferentes modelos de predicción y analizar estos con el objetivo de seleccionar el más adecuado.

Análisis de variables

En esta sección, analizaremos nuestro *dataset*, con el objetivo de entender que variables no serán de utilidad y cuales hemos de descartar para nuestros objetivos, principalmente, determinar el número de descargas.

Nuestro *data set* incluye las siguientes variables:

App Name:

Hemos barajado diferentes posibilidades, creemos que se podría llegar a usar este dato y que además sería útil si usasemos algún tipo de LLM que fuese capaz de evaluar cómo de "eficaz" fuese cada nombre, somos conscientes de que un nombre como "FunFit" o "ignite" pueden llegar a llamar más la atención que "payaso esponja horror horripilante abuelita miedo" (este nombre es real), además, podríamos usar el idioma en el cual este está escrito para realizar un análisis aún más profundo, es razonable por ejemplo que un juego con en Alemán genere más que uno escrito en Griego, por el simple hecho de que las personas germano parlantes tienen un nivel adquisitivo superior.

La principal razón por la que no realizaremos este análisis es debido a una falta de medios, nuestra idea inicial ha sido realizar esta clasificación mediante la API de chat GPT, pero debido al gran tamaño de nuestro *dataset* (millones de observaciones), hemos decidido dejar de lado esta idea. Dejamos este punto como "*Possible extensions and known limitations*".

App ID:

Esta fila será omitida por el simple hecho de que no tiene relación con el número de descargas.

Category:

Es posiblemente una de las principales variables de nuestro *dataset*, es una variable tan crucial que incluso hemos planteado realizar un modelo diferente para cada categoría (tenemos suficientes datos como para realizarlo), pero nuestra decisión final ha sido no realizar esta deserción debido al alto número de categorías existentes.

Rating:

A priori, podríamos pensar que este dato no se conoce hasta el final del desarrollo y la publicación de la aplicación a estimar, pero después de leer las opiniones de varios trabajadores del sector por diferentes medios (principalmente Quora y Reddit, sobre todo hilos sobre el desarrollo de videojuegos), hemos decidido que el *rating* de un juego está más correlacionado con los acabados de este (la calidad de diseño) que con el éxito de este, por ello hemos decidido si incluirlo y dejamos en manos del equipo de desarrolladores estimar este dato de antemano.

Rating count:

Es, al igual que el *rating*, una variable que los desarrolladores no puede conocer de antemano, aunque esta variable tiene una relación muy fuerte con la inversión realiza en marketing, es cierto que también es bastante proporcional al número de descargas, por ello no se tendrá en cuenta a la hora de realizar las predicciones.

A pesar de ello, esta variable sí que será utilizada, como ya se comentaba en la introducción, la mayoría de estudios estadísticos (ejemplo) usan este dato para eliminar *outliers*, es decir, eliminar aquellos juegos con menos de 20 *reviews* o *rating count*, esto elimina los proyectos amateur y deja solo los proyectos serios. Es comprensible que si un equipo de desarrolladores usase una herramienta como esta, es porque están realizando una inversión de tiempo y dinero en el desarrollo, por ello eliminaremos estos "*outliers"*.

Installs:

Se trata de la variable que buscamos predecir. debido a cómo se encuentra el dataset, podremos tanto considerar esta variable como continua como categórica (Google Play Store no muestra el número de descargas exacto, sino el orden de magnitud, por ello solo hay apenas unos 20-30 datos diferentes)

Minimum Installs and maximum Installs:

Minimum installs has the same value as a cleansed "installed", therefore it is useless and can be eliminated.
However, "maximum installs", since it is a numerical approximation of the real number of times the app has been installed, it could be useful for models that predict numerical variables. For this reason, "Maximum installs" will be processed and considered when dealing with regression models and "Installed" will be used for classification models.

Free:

Definitivamente se tendrá en cuenta, es fácil entender que el comportamiento de los consumidores es diferente incluso si la aplicación tiene un coste de pocos céntimos, por el simple hecho de que obliga a los consumidores a registrar métodos de pago, etc…

Es importante destacar que, tenemos más de 2 millones de juegos gratuitos y 45 mil juegos de pago en nuestro *dataset.*

Price and currency:

Deberemos realizar un cambio de moneda ya que hay 5 divisas diferentes. En cuanto al precio, es evidente de que es uno de los datos más relevantes para los desarrolladores, ya que $precio \cdot descargas = beneficios$, lo tendremos en cuenta como variable numérica, pero durante el modelaje, no se descarte crear dos modelos analizando esta variable como categórica.

Size:

Aunque el efecto en el número de descargas sea posiblemente mínimo, debido a que la mayoría de teléfonos móviles de hoy en día tienen un espacio disponible bastante superior al que se necesita para descargar aplicaciones, si que será interesante considerarlo ya que puede que sí

que tenga alguna efecto por menor que sea y además tenemos suficientes datos como para que considerar una variable extra no suponga un problema a la hora de construir el modelo.

Minimum Android:

This variable, for the moment, will not be considered due to low correlation with other variables. Despite the fact that in a few cases in history such as Pokemon Go this variable has significantly affected the downloads of the game, we will, for the moment, not consider this variable for the near future models.

Developer ID, Developer Website, Developer Email y Privacy Policy:

Evidentemente la correlación entre el número de descargas y estas características es mínima, además, esta columnas tienen un número de *unique values* demasiado grande, practicamente cada fila es única, como es lógico.

Released y Last updated:

Son variables muy interesantes, *last updated* nos muestra cuanta dedicación el equipo de desarrolladores muestra una vez la apliación ha sido publicada. *Realeased* en cambio, es también crucial ya que es más que razonable que existe una correlación entre hace cuento tiempo se publico la aplicación y el número de descargas.

Ambas variables serán transformadas a días, por lo que las trataremos como una variable númerica, el objetivo es que si un equipo de desarrolladores quiere predecir el número de descargas después de un periodo de tiempo, intruduzca este periodo en días y que si pretenden actualizar esta aplicación semanalmente por ejemplo, introduzcan 4 días en la variable "*Last updated"* (el tiempo esperado desde la última actualización).

Content Rating:

Esta variable explica el rango de edad para la cual la aplicaciçon esta disponible, tiene las categorías (*Everyone, Everyone 10+, Teen, Mature +17)*, definitivamente se trendrá en cuenta.

Ad Supported y In App Purchases:

El método por el cual se obtendran los beneficios tiene un impacto directo en el número de descargas, es por ello que consideraremos ambas variables.

Editors Choice:

Es una distinción que realiza Google Play store indicando que aplicaciones son recomendables según la tienda. Varios hilos de reddit explican que estas aplicaciones tienden a ser recomendadas con más frecuencia, además esta etiqueta asegura una cierta calidad en la apliación, por todo ello, se tomará en cuenta.

The data exploration process

Sample filtration and elimination
First of all we eliminate the columns that we have considered irrelevant for the prediction.

Secondly we notice that the dataset is very large, N=2.312.944 data samples.
As it has been said in the introduction, we want to eliminate the data which has less than 20 reviews. This leaves our dataset with N=768.108 samples.

Another obvious thing is that there is a very small proportion of paid apps compared to free apps. Therefore it has been decided to eliminate most of the free observations to have more or less the same amount of paid applications and free applications. Now we have two options, we clusterize the data of the free applications and make the selection "uniform" amongst other explicative variables or we make M random selections and train all models on these M random selections and analyze all of them. For the moment we will do the second option, but depending on the performance of the models we train we might switch in the future to the clustering. This could cause some problems in models such as generative classifiers, if we use these type of models in the future it will be taken into account.

Missing Values treatment

To proceed with data cleansing we now have to treat missing values, despite having around 20.000 rows with missing values, this amount is so small compared to the massive size of the dataset that it is not a problem to just eliminate the columns that have some missing values.

We proceed to inspect all columns and graph the values searching for missing values which are not NaN.

- In the 'Currency' explicative variable, notice that there is a currency named "XXX", indicating that the value is missing, therefore we proceed to eliminate the observations with value "XXX".
- In the 'Size' columns, there is an option which is 'Varies with device'. This is not technically a missing value but considering the size of our dataset it can be treated as it and therefore eliminated.

Data type conversion

To proceed with data treatment there we want to convert data types to usable data types for the models.

- In the 'Installs' column, the number of installs is expressed as a string character and the values are categorical in ranges (10+, 1000+, 5000+), therefore it has to be decided whether the variable should be Numerical or Categorical. For the moment the data will be converted to numerical floats and the model will determine which one is used.

- Now 'Currency' and 'Price' need to be treated. Notice that Currency has 5 different values: USD, VND, GBP, EUR and TRY, therefore, the prices are not uniform in units. A simple fix is to convert all prices to USD and delete the Currency column. It is a good idea to use USD because it is the most used currency on the dataset and it will therefore cause less effect on the price variable due to rounding error.
- The size column also has an easily treatable feature. The Bytes are expres in the standard; "Kb, Mb, Gb". Taking into consideration that "size" is a numerical explicative variable it is standard to convert all sizes to "Kb" and eliminate the unit.
- The column "Release" expresses the date in which the game was released. However, for machine learning purposes it is better to consider the variable "Time released" which can be calculated by subtracting "Time-Scrapped" and "Released". Analyzing the dataset it is noticeable that the Time-Scrapped spans a range of 24 hours, which is the time the scrapped took to collect the data. Since 1 day is very small in comparison to the years that the apps have been released for, it is reasonable to neglect this 24 hours and assume that all data has been scrapped at the same time. Converting date to numbers and subtracting them, the new "Released Time" column is obtained, which overwrites "Released".

Adding new calculated variables

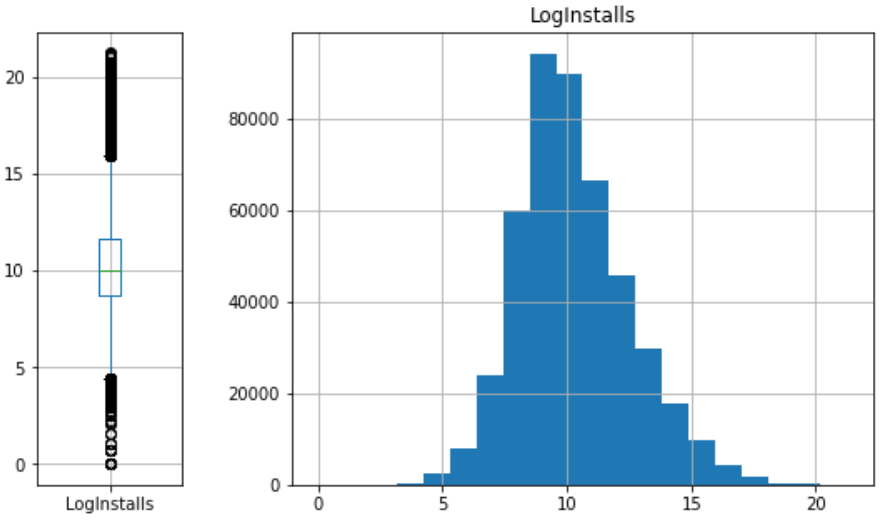
Some explicative variables in the dataset that could be useful for prediction are not collected explicitly, but could be useful for a better prediction if calculated as a function of other variables.

- The first variable added is Downloads/Time which is calculated from the "Installs" and the "Released Time" columns. This new column is useful because it is a better indicator of the popularity of the app than total downloads.

Outlier treatment and normalization

Looking at the graphs it can be seen that for the most part there are barely any outliers. To identify them the boxplots of the variables are observed and if a very far away variable is identified it could be an outlier. However, it is possible that they could be useful for the model and if when using the model the outliers do not have a lot of leverage they will be considered useful and not eliminated.

Moreover, normality for numerical variables can also be a good feature to have. For every variable normality is checked by doing a boxplot and will be further tested with Shapiro tests if the model we use requires normality.

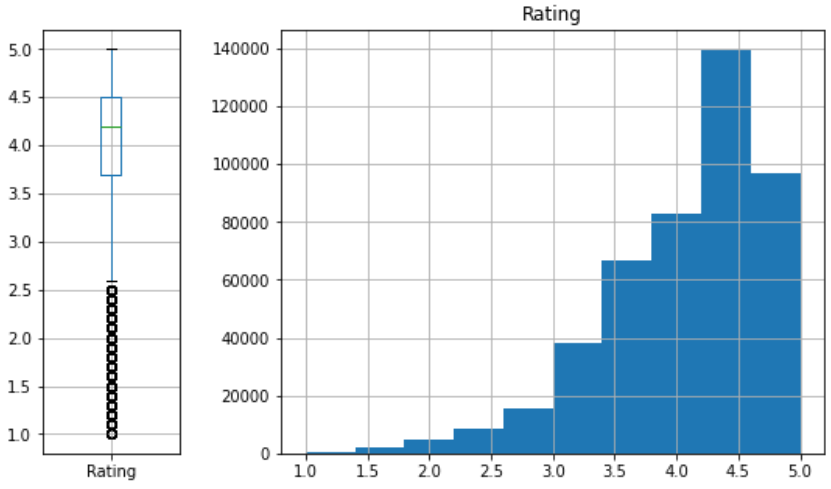


INSTALLS

Starting with the predicted variable, the normality is not present.

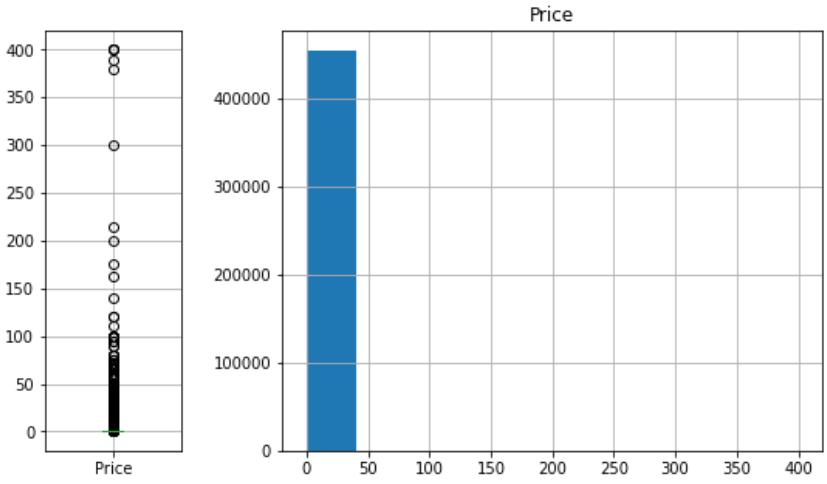
However, considering a modified variable log(installed) it can be seen that it is a nearly perfect normal distribution.

Furthermore, barley any outliers are noticed, since the low probability observations all have a very low amount of occurrences.



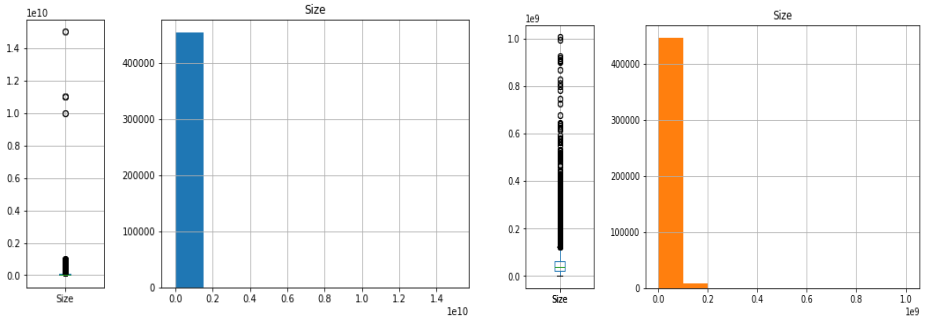
RATING

Observing the graph it is noticeable that most reviews land between 3.5-4.5 stars rating. This is a well known phenomenon in Start Ratings in general. Since it is not an exactly numerical variable, it will not be normalized if not needed for the model.



PRICE

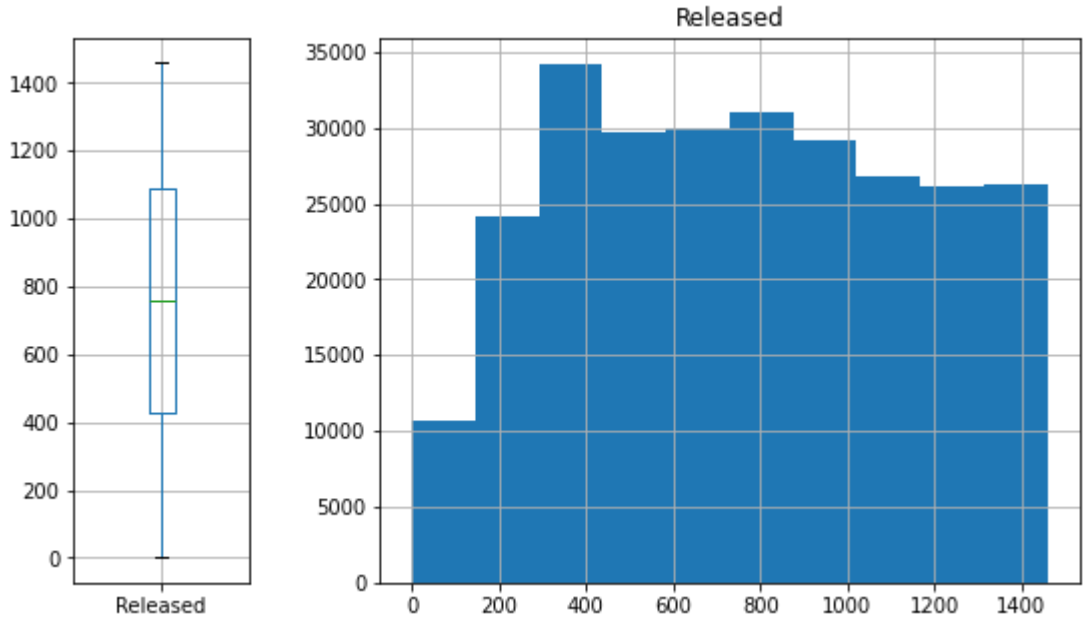
It can be observed that most observations land before the 75 dollar price. Therefore, since the apps that have more than 75 dollar price all have insignificant number of ratings, they will not be considered for the analysis and eliminated.



SIZE:

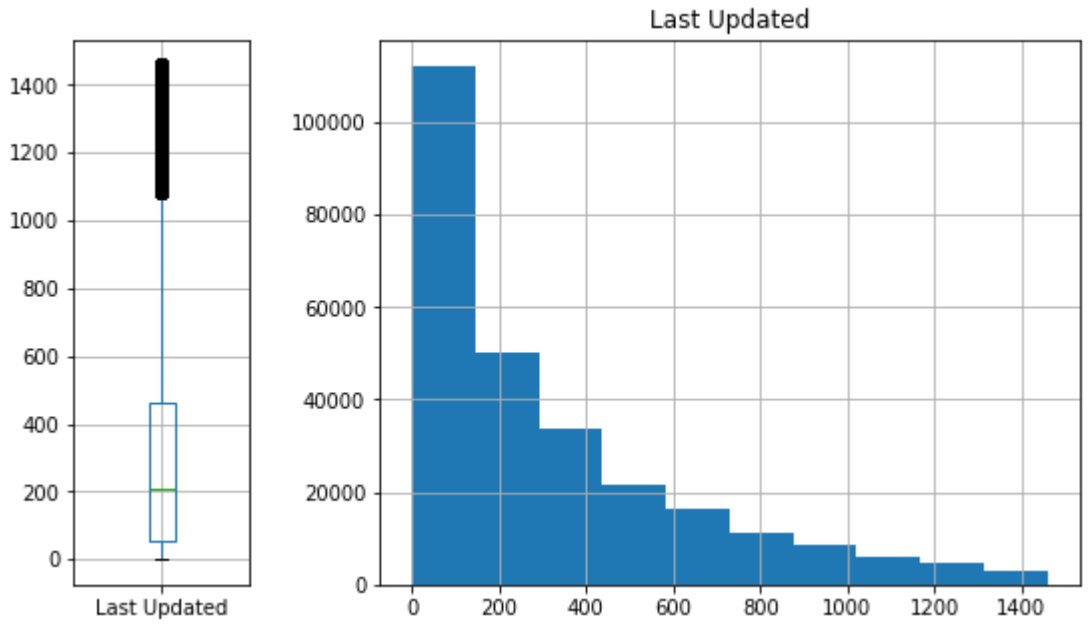
In the size dataset there are some clear outliers. After looking at the other variables of the outliers we notice that they are either errors or apps that do not make sense, therefore we eliminate them and see the resulting dataset.

RELEASED



Released has some kind of uniform distribution. The apps with more than 4 years are eliminated because the market tendency in the last year has augmented considerably and therefore the variable installs/time is very affected. We have conserved 4 years because of the objective of the predictor, to estimate the benefits in the long future.

LAST UPDATED



To end normalization we look at the last update and notice that it needs no transformation because it follows an exponential.