Google Play Store Apps

(Big Data Analytics project)

D'Apoli Clara (531889), Iannarelli Aldo (602009), Macaluso Vitalba (603160),
and Tribuzio Daniele (602020)

Università di Pisa, DI

1 Data semantics

The provided dataset collects app data that users can download from Google Play Store, the Google's official pre-installed app store on Android-certified devices, which provides access to a variety of contents including apps, books, magazines, music, movies and TV shows.
The dataset consists of 10841 records and 13 variables, that we describes as following.

- **App.** The application name in the Google Play Store;
- **Category.** The category the app belongs;
- **Rating.** A quantitative variable for app evaluation, that can only assume values from 1.0 to 5.0;
- **Reviews.** Number of reviews per app;
- **Size.** App dimension, that is expressed in two different units of measurement (Mb and kb);
- **Installs.** Number of download per app, that is expressed as a categorical variable using the notation '###+';
- **Type.** If the app is free or not, that can assume only values 'Free' and 'Paid';
- **Price.** The cost of the app, expressed in dollars using the notation '###$';
- **Content Rating.** Age group the app is targeted at;
- **Genres.** Any multiple genres the app belongs to, containing the main genre, always as first, and the secondary ones;
- **Last Updated.** Date when the app was last updated on Play Store, expressed using the *'%B %d, %Y'* format;
- **Current Ver.** Current version of the app;
- **Android Ver.** Minimum required Android version to download the app.

2 Data quality evaluation

Starting from a syntactic accuracy evaluation, we noticed that Category is a categorical attribute, as we expected, but it contains a value that is not in the domain, actually a *float*. So, we investigated the reason behind the incorrect value and it was found that the 10472$^{\text{nd}}$ record was one tab shifted, therefore we

regularized it. After the alignment, we noticed that Rating and Reviews values have been interpreted as *string* instead of *float* and *integer* types, respectively; so, we re-transformed them in the correct form.
The remained attributes are syntactically and semantically accurate, so other eventual transformations will be applied in order to make data more manageable.

2.1 Missing values detection.

At first, we observed that the target variable Rating had 1474 missing values; so, even if we hasn't investigated data yet, we decided to remove these records because we thought that it was improper handling missing target in any way.
The attributes Category and Genres had only one missing value at the previously aligned record, that actually is the 9117th one, so we took into account all those records that had the same values for Content Rating and Size, since applications belonging to the same category have similar dimensions and are often aimed at the same audience. We noticed that the mode, for these kind of records was 'TOOLS' both for Category and Genres, so we decided to substitute the only missing value using this information.
The attribute Current Ver had four missing values and, since current and recently updated app versions could be supported by newer android version more likely, we combined Last Updated and Android Ver variables to compute the mode of Current Ver, in such a way that missing values were substituted using the obtained statistics. In order to manage the two missing values present in Android Ver, the corresponding Last Updated values was taken into account since newer updates are more likely to be supported by newer Android versions. It has been also observed that apps with either 'March 2018' and 'July 2018', as last updates, had the android version '4.1 and up' as mode, so this information it has been used to handle the missing.

3 Variables transformation

In order to have a common units of measurement for the attribute Size, all values expressed in Kilobyte have been converted to Megabyte. In addition, where the dimension was 'Varies with device', the values has been replaced with '0' and then, all values has been transformed into a *float*. As a means of making the attribute Price more manageable, it has been transformed into a *float*, after removing the dollar symbol ($).
Another transformation involved the attribute Genres, that was composed of one or two words, respectively indicating the main and the eventual secondary genres of the app, separated by a semicolon; so the variable has been split in two different columns, **Main_Genres** and **Secondary_Genres**.
The variable Installs has been rewritten as interval, giving the plus symbol (+) the meaning of "more than...", in order to manage this attribute as an ordinal one [1]. The attribute Last Updated represented the date as a *string*, so it has been transformed into the correct *datetime* format: there were considered only

month and year assuming that the information about the day was not interesting for the analysis.

3.1 New variables creation

Looking at the Category, we observed that ‘FAMILY’ and ‘GAME’ represents the categories with the largest number of records. So, in the same way Google does on the store, we wanted to keep these macro-categories separated from the others creating two new *boolean* attributes:

- **Is_Family**, assumes value 1 if the app falls into the ‘FAMILY’ category, 0 otherwise.
- **Is_Game**, assumes value 1 if the app is a game, 0 otherwise.

The last step of this phase was based on the creation of two additional variables: Compatibility and RTR.
Compatibility, represents the app compatibility according to the Android version. The assumption is that more Android versions are supported by an app, more they could have high ratings, due to the greater usability. In particular, apps which have ‘Varies with device’ are the most usable ones, because developers are able to directly identify which is the best version according to devices. [2].

- ‘Varies with device’ is kept;
- Version between ‘1.0 and up’ and ‘2.3.3 and up’ and also ‘4.4W and up’ are classified to have an high compatibility, because those apps are supported by the oldest android version and smartwatch;
- Version between ‘7.0 and up’ and ‘8.0 and up’ and also ‘5.0 - 6.0’ are classified to have a low compatibility, for the opposed reason of the previous category;
- All the other Android version are classified to have a medium compatibility.

Consonant with what we said before, the idea behind the partition is that apps that support only the newer Android versions cannot be used on a wide range of devices and this could be affect the rating.
The attribute **RTR** (Reviews-Through Rate) was computed based on the fact that some apps can have the same number of reviews, but different number of download and, for this reason, some mistakes can lead because the weight of reviews is different if an app have much more installs than another. The attribute RTR (Reviews-Through Rate) has been computed to give different weights to those apps that have the same number of reviews, but different number of downloads. So, in order to make the most of this intuition, we solved the problem of Installs division into ranges, taking into account the mean of each interval.

4 Elimination of redundant variables and duplicates

4.1 Elimination of redundant variables

Dealing with data quality evaluation, we noticed that the variables Type and Price gave us similar kind of information, but in different form, and we decided to remove the first one because it can be extracted from the second one, but not vice-versa.
The variable Genres can be removed because of the creation of the two distinct Main_Genres and Secondary_Genres. Moreover, Category and Main_Genres seemed to be redundant variables, except for 'FAMILY' and 'GAME' categories, but thanks to Is_Family and Is_Game variables creation, it was possible to drop Category, thus eliminating the problem.

4.2 Handling duplicates

At the end of the data cleaning and preparation phases, we wanted to check for duplicated records and, as they came out, in order to have a dataset where each records identified a single app, we decided to keep the records we supposed to be newer considering the number of reviews. At the end, we obtained a dataset consisting of 8197 distinct apps.

5 Pairwise correlation

Correlation analysis was made using both correlation and scatter matrix. The Fig. 1 shows that the correlation between quantitative variables are close to 0, while it is a little bit higher for Rating and RTR (corr = 0.19). In fact, looking at their plot in the scatter matrix (Fig. 2), it can be noticed that these two variables are the only ones that show a slight trend. Despite there isn't correlation between Rating and Reviews, it is observed that, as the number of reviews increases, the rating also increases until it stabilizes around 4.0. In the scatter matrix diagonal are represented the distributions of the variables, from which emerges that Reviews, Price and RTR are almost uniformly distributed around the lowest values, while Rating and Size seem to have opposite distribution among each other.

Outliers and noise. The attribute Price shows in Fig. 1 that it contains potential outliers because there are some values too much greater than the others, so we decided to investigate them to find out a possible association with the app categories or between apps themselves. As result, in Tab. 1, categories these apps belong to are three different one, but, seeing at the names, we noticed that all them contained the expression 'I am rich'. About that, we discovered that these kind of apps represents such a status symbol, without any functionalities.

It was also observed only one value 'Unrated' in the attribute Content Rating, which is considered a noise, and therefore replaced with the mode according to the Main_Genres the app belongs to.

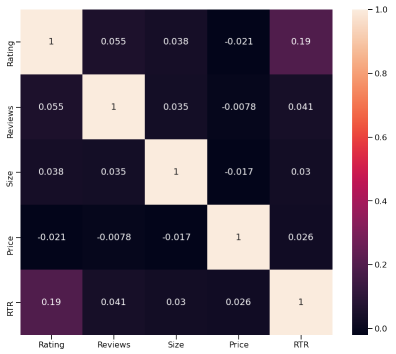

Fig. 1. Correlation matrix

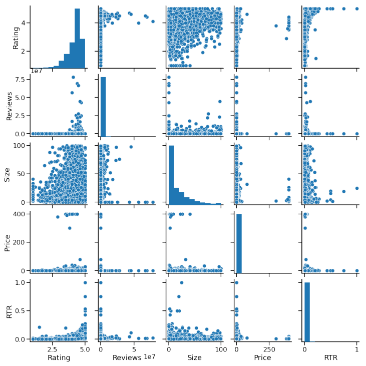

Fig. 2. Scatter matrix

Table 1. Categories for price greater than 79.99$.

Price	Category	Frequency
299.99	LIFESTYLE	1
379.99	LIFESTYLE	1
389.99	FAMILY	1
399.99	FINANCE	6
	FAMILY	3
	LIFESTYLE	2
400.00	LIFESTYLE	1

6 Integration from external sources

The use of information about the apps world wasn't based only on the Android system. On Kaggle, we found a dataset [1], dating back to July 2018, the same period our dataset belong to, concerning the applications available on Apple Store (iOS system). This additional dataset contained attributes not related to data we have, but the presence of some apps in both stores lead us to preserve this information creating a new *boolean* attribute, called exactly **Both_Stores**. The reason behind this choice is the possible presence of differences among apps, in term of rating, due to the tendency of users to give higher evaluation when they have the possibility to use an app not only on the Android system.

7 Distribution of the variables and statistics

The target variable Rating, according to the Google Play Store regulation, can assumes values from 1.0 to 5.0. Its distribution, represented in Fig. 3, shows a negative asymmetric one with a frequency spike around 4.3. Looking at the tail, it can be noticed that frequencies decrease as ratings decrease too, until zero-frequency, increasing only for rating 1.0.

[1] https://www.kaggle.com/ramamet4/app-store-apple-data-set-10k-apps

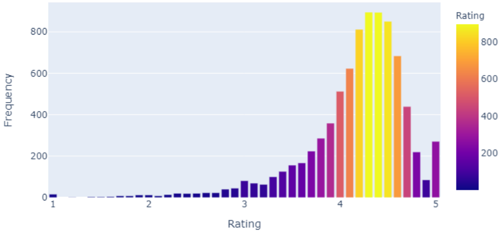

Fig. 3. Distribution of the target variable

In Fig. 4 all the boolean variables are represented in the form of a boxplot with respect to Rating. At first, it can be noticed that it seems there is no difference in the median rating if apps are not classified in family or game categories or they are not present in the Apple store. Instead, what can be said is that, if apps are categorized as games, the median rating slightly decreases, but the main difference is observed for the variable Both_Store: when an app is available both in Google and Apple store, the median rating increases and there are no rating less than 3.0.
The boxplot in Fig. 5 shows the distribution of Rating by the the variable Compatibility: apps that have compatibility which is 'Varies with device' obtains a median rating greater than the others and its distribution seems to be symmetric; on the other hand, apps that belong to the 'Medium' and 'High' compatibility categories, present largest range of rating and more potential outliers through the lowest values. As opposed to our expectation, also apps that have low compatibility obtained high rating, but the lowest ones may not be considered as potential outliers.

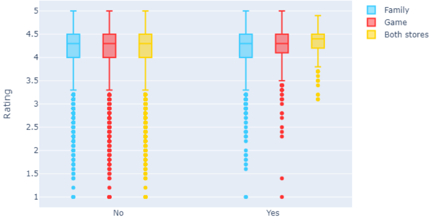

Fig. 4. Boxplots of boolean variable

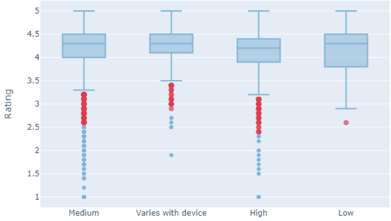

Fig. 5. Boxplot of Compatibility

Analyzing the distribution of RTR relative to Last Updated, distinguishing apps in Free and Paid, as the Fig. 7 shows, more recent updates (in 2018) corresponds to a slight increase for free apps, where these last are much more frequent than the paid ones in the dataset (Fig. 6). However, the RTR distribution is not affected by this imbalance between free and paid apps. Furthermore, what is surprisingly evident is that paid apps have a RTR much more greater than the free ones, indicating the fact that these last have a rather high review rate, probably because users tend to be more motivated to review an app they have paid for.

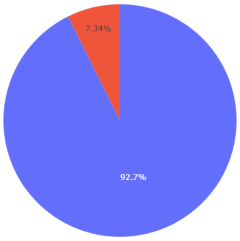

Fig. 6. Distribution of Price

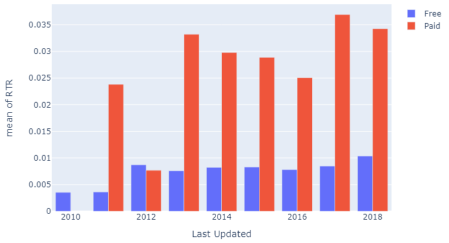

Fig. 7. Distribution of RTR by Last Updated and Price

8 Task proposal

The main goal of this project is to predict ratings of the apps, according to the chosen variables, using a classification model. In particular, looking at the correlation between the app review rate and its rating, the first one variable would seem to be more useful than others for the prediction task, so we plan to verify the importance of RTR in the model we will train and whether at higher RTR values correspond higher app evaluation. Related to this aim, how rate values vary, according to free or paid apps, may reveal information about the goal of the project. At the end, the fact that an app is present in both Google and Apple store is relevant to predict the target variable?

References

1. CPIDroid ‘How Google Displays the App Installs Count on Google Play Listing?’, https://tinyurl.com/yxcsl5mg. Apr 20
2. Google Operating System ‘Varies With Device’, http://googlesystem.blogspot.com/2013/08/varies-with-device.html. Aug 13