EDA & Data Preprocessing on Google App Store Rating Dataset

Domain: Mobile device apps

Context:

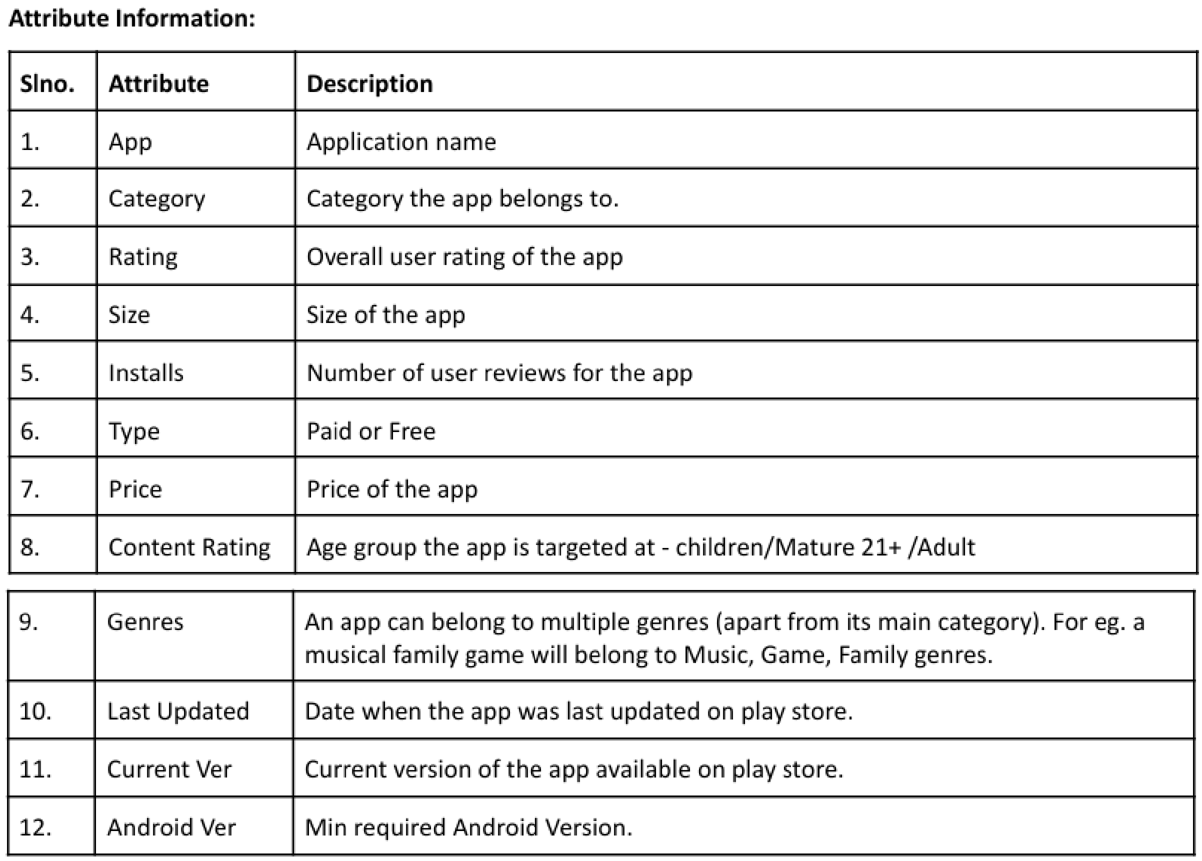
The Play Store apps data has enormous potential to drive app-making businesses to success. However, many apps are being developed every single day and only a few of them become profitable. It is important for developers to be able to predict the success of their app and incorporate features which makes an app successful. Before any such predictive-study can be done, it is necessary to do EDA and data-preprocessing on the apps data available for google app store applications. From the collected apps data and user ratings from the app stores, let's try to extract insightful information.

Objective:

The Goal is to explore the data and pre-process it for future use in any predictive analytics study.

Data set Information:

Web scraped data of 10k Play Store apps for analyzing the Android market. Each app (row) has values for category, rating, size, and more.



Attribute Information:

Slno.	Attribute	Description
1.	App	Application name
2.	Category	Category the app belongs to.
3.	Rating	Overall user rating of the app
4.	Size	Size of the app
5.	Installs	Number of user reviews for the app
6.	Type	Paid or Free
7.	Price	Price of the app
8.	Content Rating	Age group the app is targeted at - children/Mature 21+ /Adult
9.	Genres	An app can belong to multiple genres (apart from its main category). For eg. a musical family game will belong to Music, Game, Family genres.
10.	Last Updated	Date when the app was last updated on play store.
11.	Current Ver	Current version of the app available on play store.
12.	Android Ver	Min required Android Version.

Questions:

1. Import required libraries and read the dataset.

2. Check the first few samples, shape, info of the data and try to familiarize yourself with different features.

3. Check summary statistics of the dataset. List out the columns that need to be worked upon for model building.

4. Check if there are any duplicate records in the dataset? if any drop them.

5. Check the unique categories of the column 'Category', Is there any invalid category? If yes, drop them.

6. Check if there are missing values present in the column Rating, If any? drop them and and create a new column as 'Rating_category' by converting ratings to high and low categories(>3.5 is high rest low)

7. Check the distribution of the newly created column 'Rating_category' and comment on the distribution.

8. Convert the column "Reviews" to numeric data type and check the presence of outliers in the column and handle the outliers using a transformation approach.(Hint: Use log transformation)

9. The column 'Size' contains alphanumeric values, treat the non numeric data and convert the column into suitable data type. (hint: Replace M with 1 million and K with 1 thousand, and drop the entries where size='Varies with device')

10. Check the column 'Installs', treat the unwanted characters and convert the column into a suitable data type.

11. Check the column 'Price' , remove the unwanted characters and convert the column into a suitable data type.

12. Drop the columns which you think redundant for the analysis.(suggestion: drop column 'rating', since we created a new feature from it (i.e. rating_category) and the columns 'App', 'Rating' ,'Genres','Last Updated', 'Current Ver','Android Ver' columns since which are redundant for our analysis)

13. Encode the categorical columns.

14. Segregate the target and independent features (Hint: Use Rating_category as the target)

15. Split the dataset into train and test.

16. Standardize the data, so that the values are within a particular range.