**Google Play Store Data Analysis**

**Mohammed Saudh**

**Data science trainee,**

**AlmaBetter, Bangalore**

**Abstract:**

The google play store is a vast store of data. A lot of data regarding the present apps like rating, reviews, no. of installs etc. is available on the play store that can be analyzed and studied while app development companies are undertaking new projects. In this project we will do a similar study to determine what type of app would be most successful among the play store audience.

**1.Problem Statement**

Data is taken from the Google play store dataset. Every row contains various entries regarding a certain app. We will be doing Exploratory data analysis on this data set, which is a very important step in data science cycle, as it not only helps in taking very initial business decisions but also in preparing the data for further modelling for use in machine learning algorithms. Our objective will be to structure the data, clean it and present certain trends that we observe that can help us draw very preliminary conclusions about the probability of success of a newly launched app.

**2. Introduction**

### The data set contains the following columns:

* App: This Column contains the name of the app
* Category: This contains the category to which the app belongs. The category column contains 33 unique values.
* Rating: This column contains the average value of the individual rating the app has received on the play store. Individual rating values can vary between 0 to 5.
* Reviews: This column contains the number of people that have given their feedback for the app.
* Size: This column contains the size of the app i.e. The memory space that the app occupies on the device after installation.
* Installs: This column indicates the number of time that the app has been downloaded from the play store, these are approximate values and not absolute values.
* Type: This column contains only two values- free and paid. They indicate whether the user must pay money to install the app on their device or not.
* Price: For paid apps this column contains the price of the app, for free apps it contains the value 0.
* Content Rating: It indicates the targeted audience of the app and their age group.
* Genre: This column contains to which genre the app belongs to, genre can be considered as a sub division of Category.
* Last updated: This column contains the info about the date on which the last update for the app was launched.
* Current version: Contains information about the current version of the app available on the play store.
* Android version: Contains information about the version of the android OS on which the app can be installed.

## **3. Data Cleaning and Preparation.**

The available data is raw and unusable for Exploratory data analysis, so before we do anything with the data we will have to explore and clean it to prepare it for data analysis.

### **Step1**: We write a function **printinfo(), t**hat will display 3 attributes about all the columns: Data type, Count of null values and The number of unique values in that column.

Table

Description automatically generated

* **Step 2**: we start off with the column ‘Type’ we can see that it has one null value. We checked this row and found out from the play store that it is a free app. We use **filllna()** function of the pandas library to fill this value.
* **Step 3**: We drop the columns ‘Current Ver’, ‘Android Ver’ and ‘last updated’ from our dataset using the **drop()** function of the pandas library.
* **Step 4**:We can see that the ‘Rating’ column has 1474 null values. Due to low variations in the rating values and a lot of repeated values the ‘mode’ would be a suitable statistical indicator to replace the null values with. We calculate the mode of the column using the **mode()** aggregate method, and fill this value in place of null values using the **fllna(**) function.





* **Step 5:** We can see that the ‘Reviews’ column despite being a numerical indicator is of the ‘object’ data type, we will convert this to ‘int’ data type using the **astype(int)** function.



* **Step 6:**  We can see that the size column, which should be numeric, is of the data type ‘object’, it also has characters ‘k’ and ‘M’ in the values which stand for kilobytes and Megabytes, we will replace the ‘k’ with 1000 and ‘M’ with 1000000. Some values also have ‘+’ sign in them, which will be removed.

A picture containing application

Description automatically generated

Next, we will convert this column into ‘int’ datatype using the **pd.to\_numeric()** function.

Text

Description automatically generated

* **Step 7:** The ‘Installs’ column values contain the characters ‘+’ and ‘,’ which are going to prevent us from converting this column into a numeric datatype. We will get rid of these using the **strip()** and **replace()** functions.

Graphical user interface, text, application, chat or text message

Description automatically generated

Next, we will convert this column into ‘int’ data type using the **pd.to\_numeric()** function.



* **Step 8:** The values in the column ‘Price’ might have the ‘$’ sign in some values and the column is of the datatype ‘object’. We will first remove the ‘$’ sign using the **strip()** function and then convert the column into ‘int’ datatype using the **pd.to\_numeric()** function.

**4. Exploratory Data Analysis**

Exploratory Data Analysis is the process of exploring data, generating insights, checking assumptions and revealing underlying hidden patterns in the data.

* **Step1: Statistical Indicators of all numerical columns.**

The statistical indicators of all columns like the percentiles, max and min values, mean and median are displayed using the **describe()** function.

Graphical user interface, table

Description automatically generated

* **Step2: Distribution of ratings across categories.**

The **boxplot()** function is used to plot the distribution of ratings across the categories on a boxplot. (For graph and conclusions refer presentation)

Text

Description automatically generated with medium confidence

* **Step 3: Distribution of apps across categories.**

The Category wise counts are calculated using the **value\_counts()** function and the results are visualized using a bar plot.



Text, letter

Description automatically generated

(For graphical representation and conclusions refer presentation)

* **Step4: Distribution of Installs across Categories**

The category wise installs are calculated by using groupby() function on the ‘Category’ Column and the using the aggregate function sum() on the ‘Installs’ Column.

Then this data is plotted on a bar plot.



Text

Description automatically generated with low confidence

(For graphical representation and conclusions refer presentation)

* **Step 5: Comparing percentage of paid and free apps.**

The percentage of paid vs free apps is calculated using the **groupby()** function on the ‘Type’ column and the **count()** aggregation function on the ‘App’ Column.



Then this data is plotted on a pie chart.

Text, letter

Description automatically generated

(For graphical representation and conclusions refer presentation)

* **Step 6: Comparing installs between paid and free apps.**

The percentage of installs of paid vs free apps is calculated using the **groupby()** function on the ‘Type’ column and the **sum()** aggregation function on the ‘Installs’ Column.



This data is then plotted on a pie chart.

Text, letter

Description automatically generated

(For graphical representation and conclusions refer presentation)

* **Step 7: Comparing apps across Content Ratings.**

The Category wise app count is calculated using the **value\_counts()** function and the results are visualized using a bar plot.



Text

Description automatically generated

(For graphical representation and conclusions refer presentation)

* **Step 8: Comparing Installs across Content Ratings.**

The installs across content ratings are calculated using the **groupby()** function on the column ‘Content Rating’ and then the **sum()** aggregate function on the ‘Installs’ column.



This data is then plotted on a bar plot for visualization.

Text

Description automatically generated

(For graphical representation and conclusions refer presentation)

* **Step 9: Checking the General distribution of app ratings.**

Rating vs Frequency curve was plotted on a continuous distribution curve using the **sns.kdeplot()** function.

Text

Description automatically generated

(For graphical representation and conclusions refer presentation)

* **Step 10: Checking for top 10 installed apps within a certain category:**

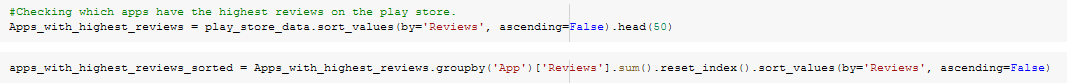
A custom function **findtop10incategory()** that finds and plots the top 10 apps installed in the category that is provided as the argument for the function was written and used to find top 10 installed apps in categories of interest.

Text

Description automatically generated

* **Step 11: Checking for apps with highest reviews:**

Apps with highest reviews are calculated using the **groupby()** function on the ‘App’ column and the **sum()** aggregate function on the reviews column. This data is then plotted on a bar plot for better visualization.



A picture containing text

Description automatically generated

# 

**5. Conclusion:**

Through exploratory data analysis we have observed some trends and have made some assumptions that might lead to app success among the users in the play store.