***CSE3505- Foundations of Data Analytics***

***J COMPONENT-FINAL PROJECT REPORT***

***Google Play Review Analysis***

*By*

Reg. No: 20BCE1105                         Name: Kesav Santhosh

Reg. No: 20BCE1653                             Name: Parth Tiwari

B. Tech CSE

*Submitted to*

**Dr. Priyadarshini R,**

Assistant Professor,

SCOPE, VIT, Chennai

**School of Computer Science and Engineering**



*November 2022*



***BONAFIDE***

This bonafide is to certify that this project report of ‘GOOGLE PLAY REVIEW ANALYSIS’ is the work of KESAV SANTHOSH (20BCE1105) and PARTH TIWARI (20BCE1653) who carried out the project work for the subject Foundations of Data Analytics (CSE3505) under my supervision for the fall semester 2022-23.

Dr. Priyadarshini R

Assistant Professor

SCOPE,VIT Chennai

***ACKNOWLEDGEMENT***

We wish to express our sincere thanks and deep sense of gratitude to our project guide, Dr Priyadarshini R, SCOPE, for her consistent encouragement and valuable guidance offered to us in a pleasant manner throughout the course of the project work and also for motivating us to do the project on the topic.

Secondly, we take this opportunity to thank all the faculties of the school for their support and their wisdom imparted to us throughout the course.

Thirdly, we thank our parents, family, and friends for bearing with us throughout the course of our project and for the opportunity they provided us in undergoing this course in such a prestigious institution.

In the end, we would like to thank our friends who believed in us throughout the project and displayed appreciation for my work.

DATE-13/11/2022 KESAV SANTHOSH-20BCE1105

PARTH TIWARI-20BCE1653

***TABLE OF CONTENTS***

|  |  |  |
| --- | --- | --- |
| **SR.NO** | **CONTENT** | **PAGE NUMBER** |
| 1 | ABSTRACT | 5 |
| 2 | INTRODUCTION | 6 |
| 3 | LITERATURE REVIEW | 7 |
| 4 | ARCHITECTURAL DESIGN & REQUIREMENTS | 10 |
| 5 | DATASETS | 11 |
| 6 | MODULES | 12 |
| 7 | IMPLEMENTATION AND PERFORMANCE ANALYSIS | 15 |
| 8 | CONCLUSION | 30 |
| 9 | FUTURE WORKS | 31 |
| 10 | APPENDIX | 32 |
| 11 | REFERENCES | 38 |
| 12 | WHAT WE HAVE DONE | 38 |

***ABSTRACT***

There are thousands of categories of apps on play store. Each category contains thousands of apps. There are thousands of ratings and reviews for majority of apps. So, it is a tedious task to keep track of all the ratings and reviews of apps on the play store. So, we did our project ‘Google Play Review Analysis’ to help achieve all the statistics of the apps available on Google Play Store. Our project aims at analysing the app details and user reviews of an app on the play store. Our project aims at developing linear regression between number of installs and number of reviews and predicting the number of reviews which would be available on a new app available on Play Store. Our project aims at forming the distribution of rating with respect to all the factors such as Price, Size, Installs, etc and analysing the sentiment of all reviews and plot distribution with respect to the sentiment polarity that is Positive, Negative and Neutral. Our project is helpful in predicting whether an app is good or bad by reviewing the reviews on that app and performing sentimental analysis for each review.

***INTRODUCTION***

Mobile applications are widely available. They are simple to make and may be profitable. These two reasons have led to an increase in the number of apps being created. In our project, we will compare more than ten-thousands of apps in Google Play from various categories to conduct a thorough analysis of the Android app field. In order to develop strategies to promote growth and retention, we will search the data for insights. It’s challenging to evaluate both positive and negative user reviews posted on the Play Store for a particular app, making it challenging for a user to determine whether the app is great or bad just by the ratings for the app and by reading the reviews. We are doing analysis to determine how different factors, such as category, price, and number of reviews, are influencing an app’s rating.

***LITERATURE SURVEY***

**1)**

**TITLE-**

Google Play Store Review Sentiment Analysis

**AUTHOR-**

Vaishnavi Dudhankar, Nitu Sen, Atmja Langde, Vaishnavi Kupade

**GIST-**

What people believe has always had an impact on how we live our daily lives. Our personal beliefs have always been influenced by the ideas and opinions of others. The Web has more and more information on people’s ideas and feelings as it becomes a bigger part of their social life. Opinion mining and sentiment analysis are other terms for the process of extracting insight from this massive volume of unstructured data. Mobile applications are becoming indispensable in our lives due to the quick development of smartphones. Customers, however, find it challenging to stay informed about and comprehend the app world because new apps are always being released. Therefore, Google Play Store sentiment analysis of app reviews will assist app developers in keeping their specific apps up to date in order to maintain their specific applications in the top lists and also assist users in choosing the most well-liked application. Our suggested design would demonstrate that this plan had high scale-out scalability and that by introducing more Name Nodes, the maximum number of files that the system could store could be considerably expanded.

**RESULT-**

The primary location for obtaining and uploading Android applications is now Google Play. Users of Android apps download apps for their own usage. Every app user has a unique experience with the app. Users download and utilise these apps, sharing their opinions in the form of reviews and rating them from one to five. The app’s owner either fix the negative remarks or delete them. We examined all necessary approaches for these projects and presented a new solution to categorise the polarity underlying consumers’ feelings. It has been demonstrated that sentiment analysis is simple to use.

**2)**

**TITLE-**

Sentiment Analysis of Application Reviews On Google Playstore

**AUTHOR-**

Sangeeta Panigrahi, Saarim Momin, Pooja Patil, Prof Prachi Kshirsagar

**GIST-**

What other people believe has an impact on how we live our daily lives. Others’ ideas and beliefs have always influenced our own. The Web has more and more information on people’s ideas and feelings as it becomes a bigger part of their social life. Sentiment analysis is the process of extracting insight from this vast volume of data. Mobile applications are becoming indispensable in our lives due to the quick development of smartphones. Consumers, however, find it challenging to comprehend the app industry. Therefore, sentiment analysis of app reviews on Google Play Store will assist developers in keeping their specific applications up to date in order to maintain them in the top lists and also assist users in choosing the most popular application.

**RESULT-**

The method for obtaining the app features highlighted in user evaluations and the feelings connected to them. These can assist developers in analysing and quantifying user feedback on individual app features and using that data to plan next releases. We’re eager to contribute to the application industry and help Google Play Store consumers find better, free apps that have received positive reviews. The proposed technique is effective from that perspective since user reviews of apps differ from category to category. The outcome is enough for evaluating the Android app, and the programmers may foresee issues and improvements that must be made in order for the app to gain popularity quickly.

**3)**

**TITLE-**

Sentiment Analysis of Application Reviews On Google Playstore

**AUTHOR-**

Sangeeta Panigrahi, Saarim Momin, Pooja Patil, Prof Prachi Kshirsagar

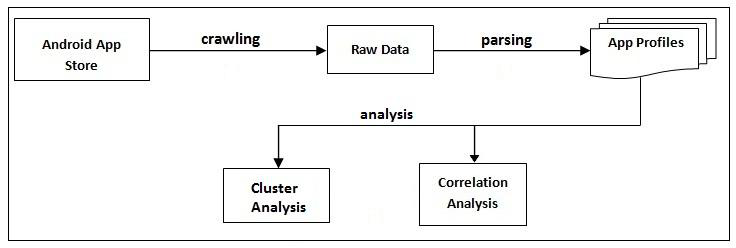
**GIST-**

In the market for mobile applications, maintaining a competitive edge and assessing the requirements of high-quality apps are important. The user reviews of these apps are crucial to the mobile app market. People have the option to participate and express their reviews, ratings, and criticism regarding applications thanks to the quick development of online technology. Various widely used machine learning techniques, including Logistic Regression, Random Forest Classifier, and Multinomial Naive Bayes, were used to measure the statistical data in the findings. Bigram, Trigram, and N-gram were assessed using several metrics, such as accuracy, precision, recall, and F1 score, and the statistical output of these algorithms was compared. Each algorithm is individually examined, and the outcome has been evaluated. Each method is individually examined, and the outcome has been assessed. It has been shown that the optimum method for reviewing and analysing Google Play Store apps is logistic regression. Scientific analysis of the findings reveals the reliability of the logistic regression approach for classifying reviews into three categories: positive, negative, and neutral.

**RESULT-**

On the Google Play Store, developers have uploaded and consumers have downloaded hundreds of thousands of apps. Users have their own unique experiences and utilise these programmes for their own purposes. These programmes are downloaded and used by users, who then rate them on a scale of 0 to 5 and provide their feedback in the form of comments or reviews. We looked at the category of reviews that may be favourable, unfavourable, or neutral. With the use of several machine learning methods, we have examined the application semantics. Contrarily, we found that the most active method with a high precision score was the logistic regression technique. We may utilise this machine-learning approach to assess the user evaluations.

***ARCHITECTURAL DESIGN***



The data is been extracted from the Google Play Store using crawling and converted to raw data and then the raw data is been refined to make app profiles using parsing and then the app profiles in the form of rows are used for analysis. But in the project, we directly got the inbuilt dataset from Kaggle that is the app profiles with thousands of app in rows and the app attributes in columns. We performed the analysis on the app profiles and got the expected result.

***REQUIREMENTS***

1. Jupyter Notebook
2. Python 3.8
3. MATPLOTLIB
4. Pandas
5. Numpy
6. Plotly
7. Seaborn
8. Warnings

***DATASETS***

First dataset consisting of attributes of all app related information such as-

1) app name

2) category

3)rating

4)reviews

5)size

6)installs

7)type

8)price

9)content rating

10)genres

11)last updated

12)current version

13)android version

Second dataset consisting of attributes of all user reviews related data that are-

1)App

2) Review

3)Sentiment

4)Sentiment Polarity

5)Sentiment Subjectivity

***MODULES***

1. **Google Play Store apps and reviews**

In this module, we have imported the panda package and read the dataset apps.csv. We have printed the total number of apps available in the dataset. We have printed a sample of ten rows from the dataset.

**2) Data cleaning**

In this module, we cleaned the dataset by removing the unnecessary columns such as Android Version, Current Version, Last Updated. Also, certain columns such as Installs and Price contained symbols. Price contained dollar ($) symbol and Installs contained plus (+) and comma (,). So, we removed the symbols from the rows which contained the symbols in Installs and Prices and printed the structure of the dataset.

**3) Correcting data types**

In this module, we imported the numpy library and converted the installs column values to float. We also converted the price values to float. Because, we wanted the datatype of the two columns to be float for calculations and graphs. We printed the structure of the dataset.

**4) Exploring app categories**

In this module, we imported the plotly library and established a connection with it. We printed the total number of unique categories. Counted the number of apps in each category. Sorted the categories with respect to the number of apps in that category in descending order and then plotted a bar graph for the category v/s number of apps in that category in descending order.

**5) Distribution of app ratings**

In this module, we have calculated the mean of the rating of the apps and printed the value. The data is been distributed according to the ratings and plotted for rating of apps v/s number of apps rated with that rating.

**6) Size and price of an app**

In this module, we have imported seaborn library as well as warnings. We have selected rows where both rating and size values are not null and have selected the categories with more than 500 apps for better results and have plotted joint plot for size of an app v/s rating of the app. Selected the paid apps and plotted for price of the app v/s rating of the app.

**7) Performance Metrics**

We imported the matplotlib, numpy, sklearn libraries and for the columns Installs and Reviews, we fit the columns into the linear regression model and predicted the reviews for an app with total number of installs 10000.

**8) Relation between app category and app price**

In this module, we printed out the category, app name and app price for the paid apps which costs more than 200 dollars as that are apps which are made for fun or are scam apps like ‘Most expensive app’ and ’I am rich’. We also plotted for price and for certain category using stripplot. The apps that were less than 200 dollars were more focused on.

**9) Filter out junk apps**

In this module, we plotted for all categories and price of paid apps with app price less than 100 dollars using stripplot by filtering out all the junk apps which cost more than 100 dollars.

**10) Popularity of paid apps vs free apps**

In this module, we plotted using box plot for free and paid apps for the number of downloads and found out that the downloads for the free apps are more than the number of downloads for the paid apps. The mean number of downloads of free apps are more than the mean number of downloads of paid apps.

**11) Sentiment analysis of user reviews**

In this module, we have imported the second data set with user reviews values that is user\_reviews.csv. We merged the two data sets apps and user reviews. Removed the NA values from the columns Sentiment and Review and plotted the sentiment polarity for free apps and paid apps using box plot.

**12) Remove null values from user reviews**

In this module, we removed the review rows with NA value and printed the user reviews data set.

**13) Draw histogram to understand user sentiment distribution**

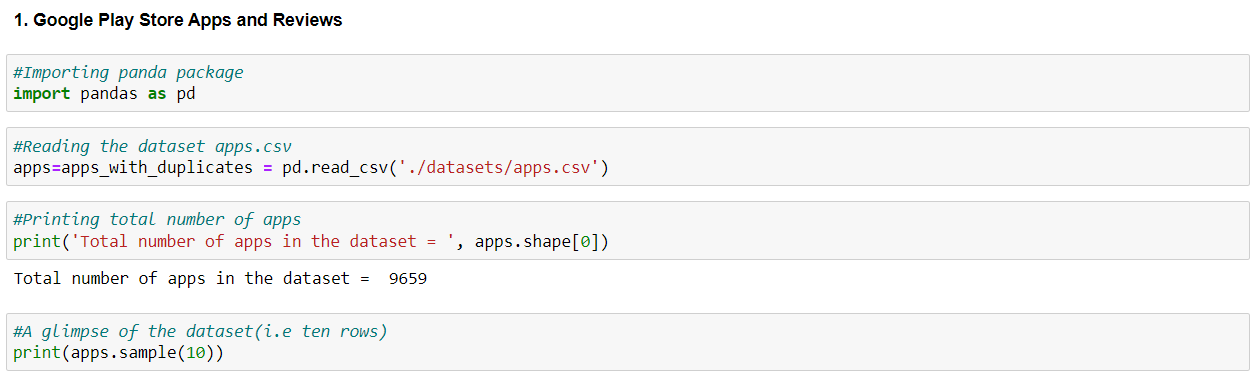
In this module, we plotted a histogram for sentiment distribution into positive reviews, negative reviews and neutral reviews.

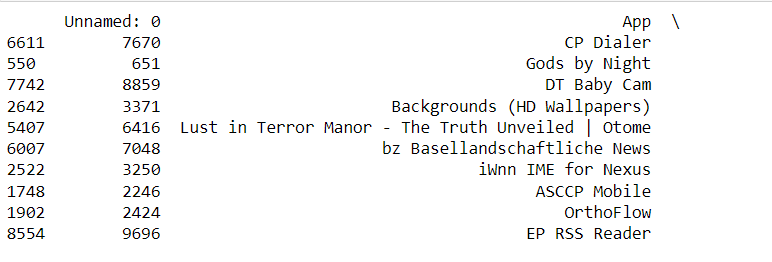
**14) Few positive, neutral and negative reviews**

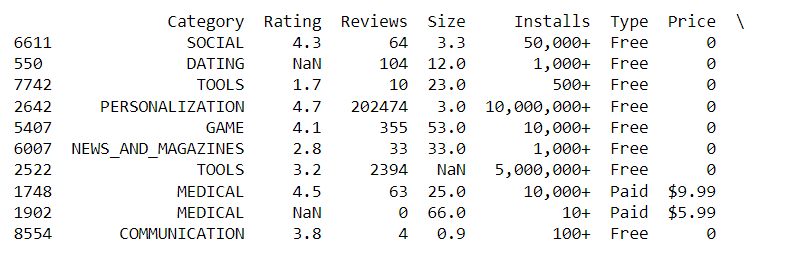
In this module, we printed twenty positive reviews which contained ‘Amazing’ ,’Best way’, etc then we printed twenty negative reviews which contained ‘Horrible’,’ Waste’, etc and then we printed the twenty neutral reviews such as ‘helpful’,’ On test’.

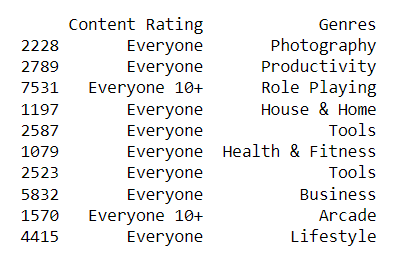
***IMPLEMENTATION AND PERFORMANCE ANALYSIS***

1. **Google Play Store apps and reviews**

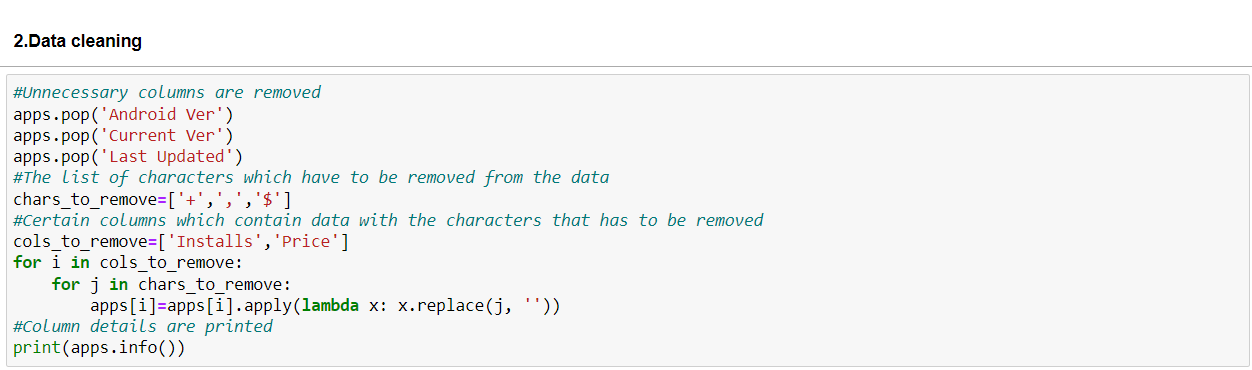
****

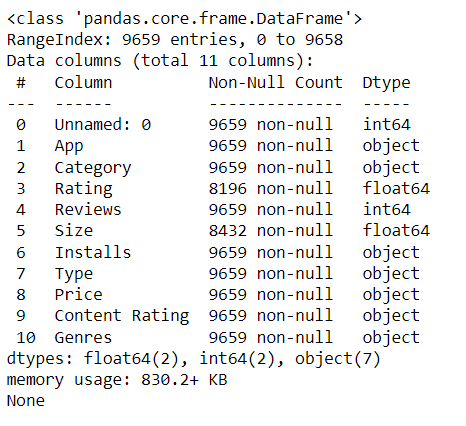
****

****

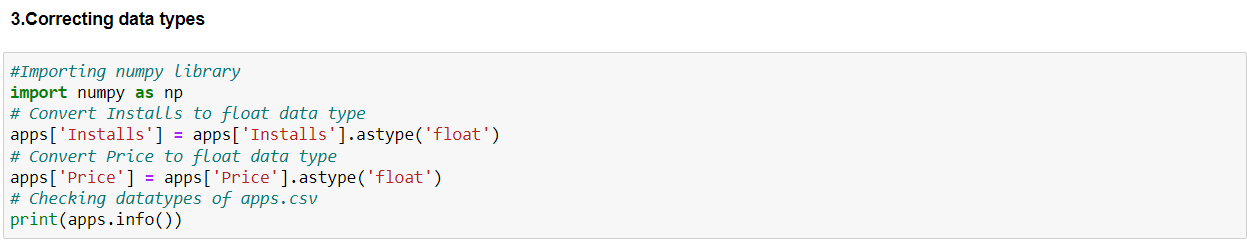
****

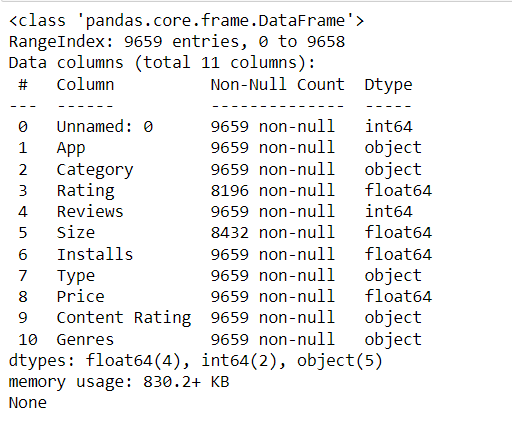
**2) Data cleaning**

****

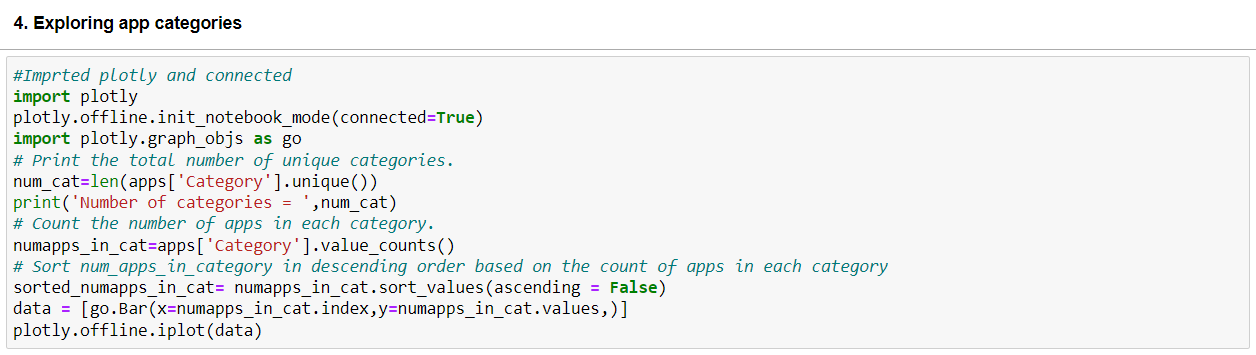
****

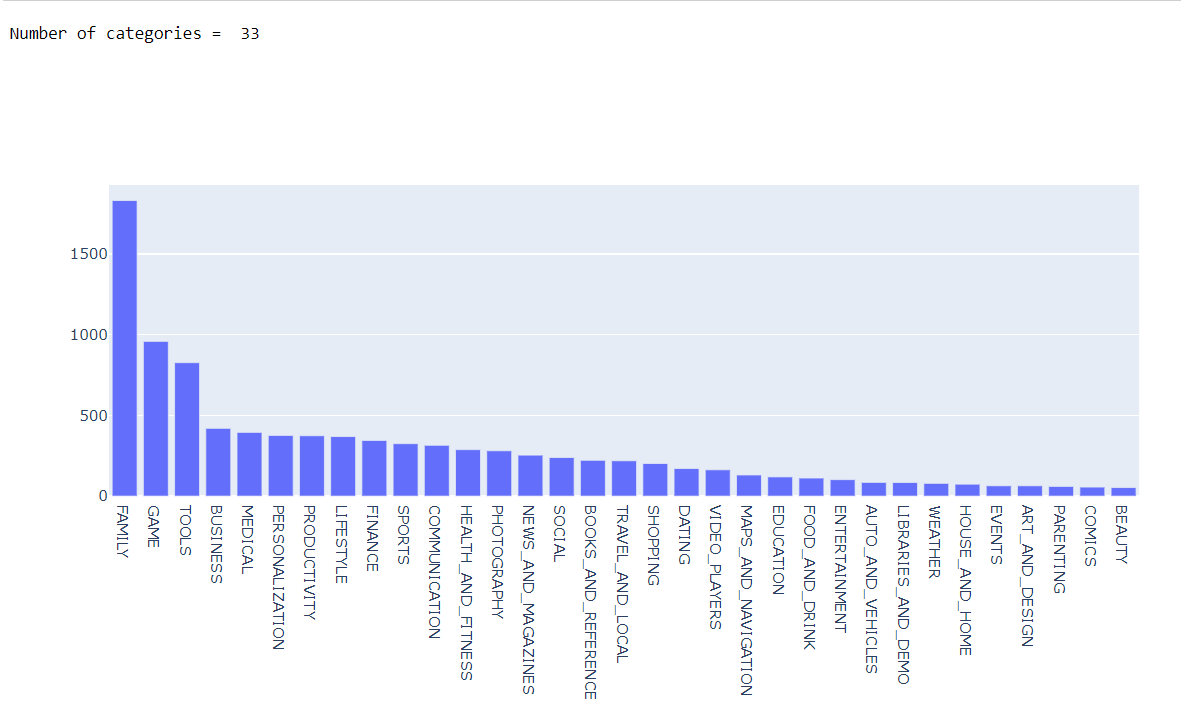
**3) Correcting data types**

****

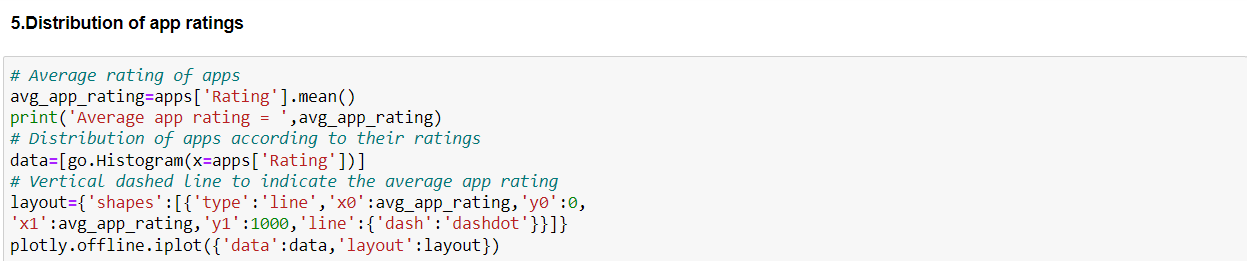
****

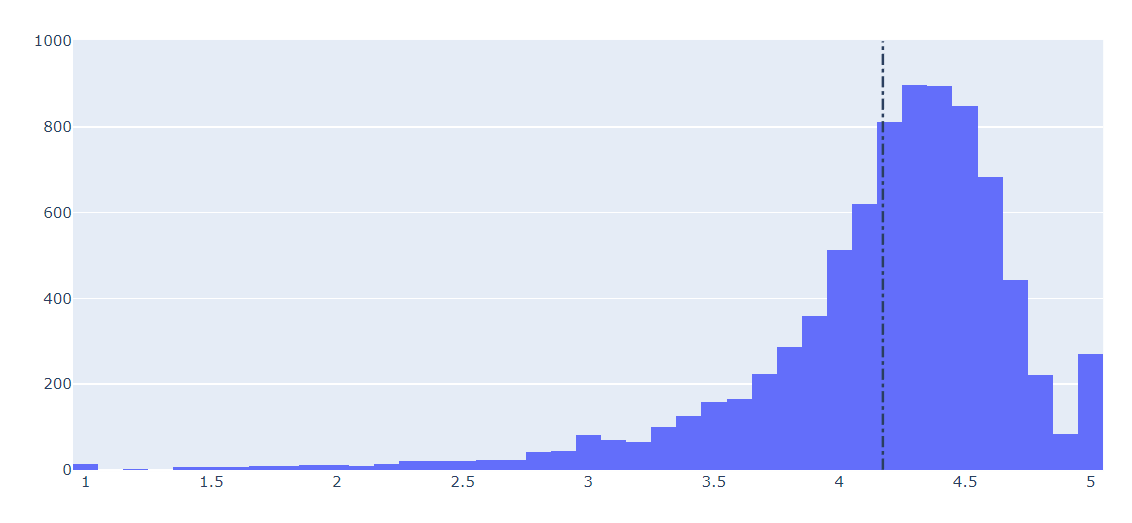
**4) Exploring app categories**

****

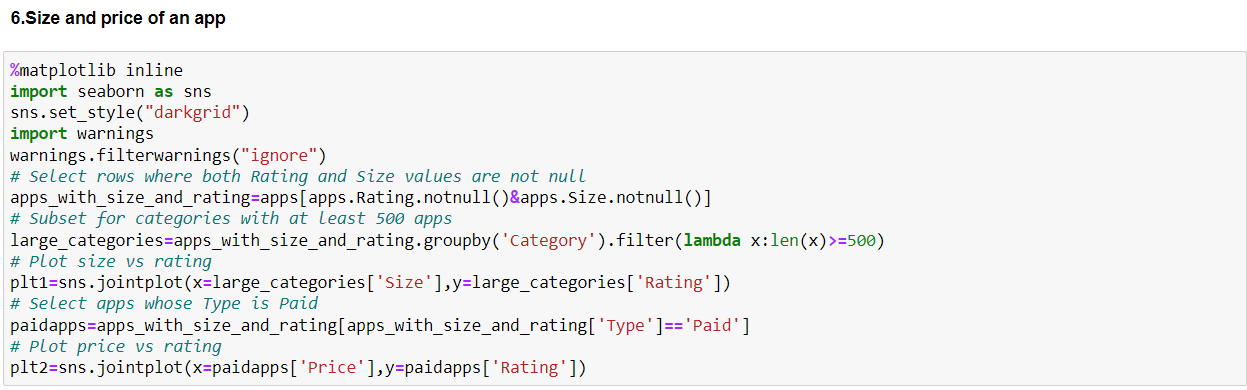
****

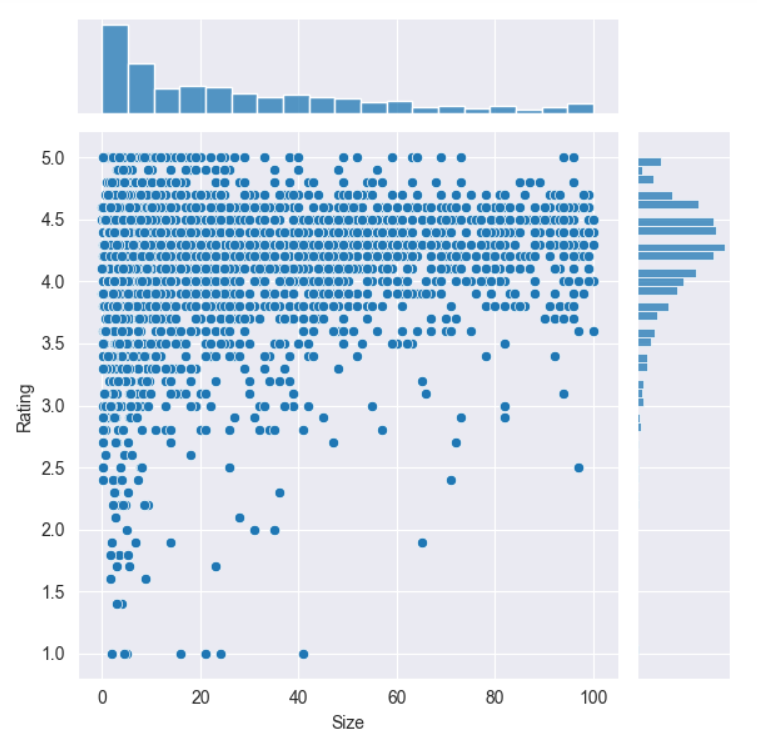
**5) Distribution of app ratings**

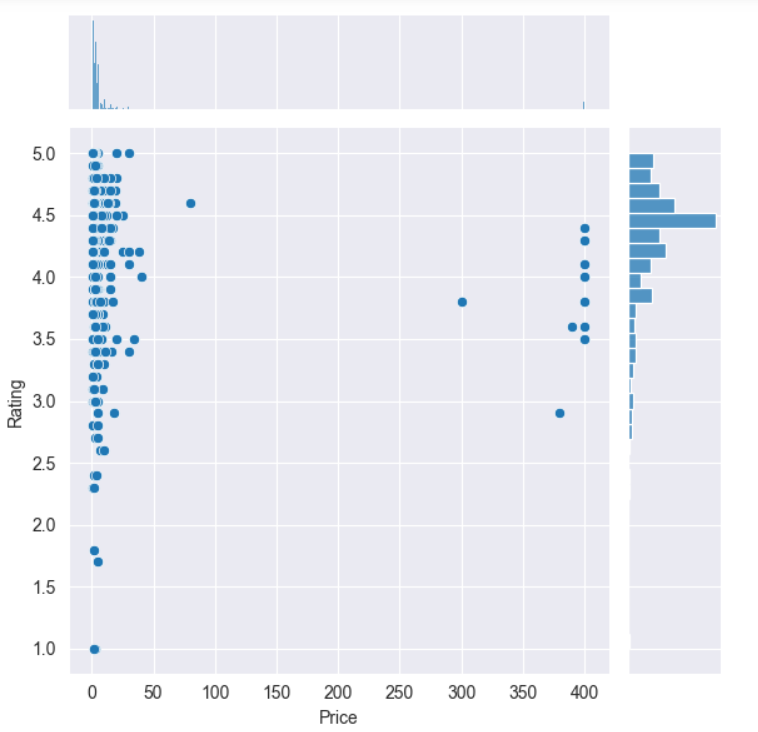
****

****

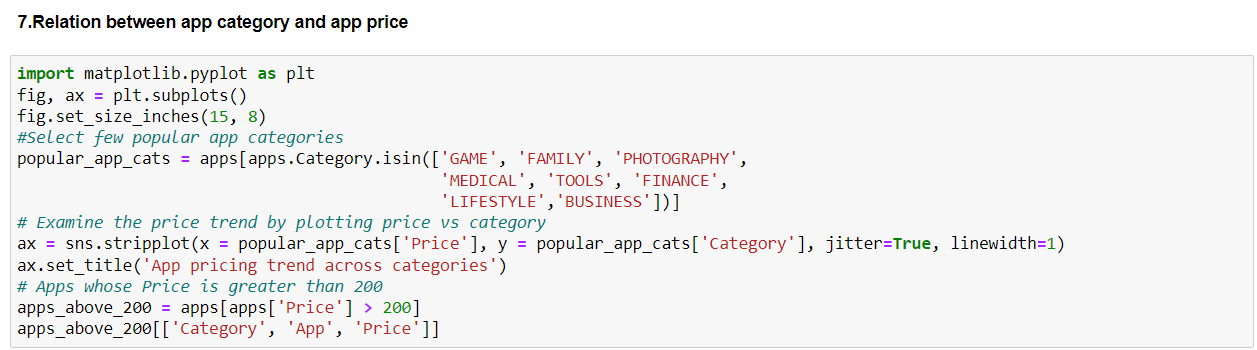
**6) Size and price of an app**

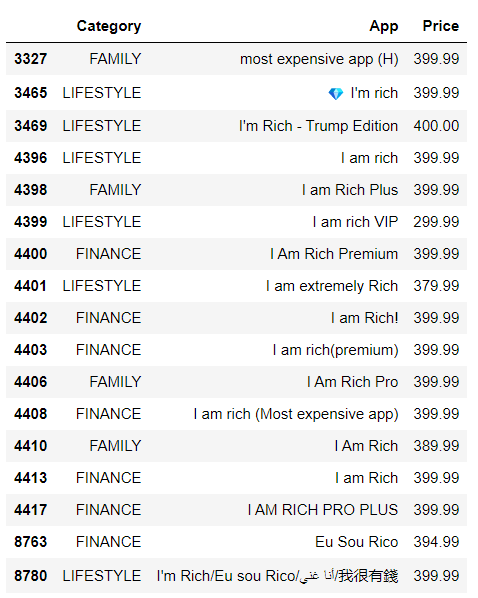
****

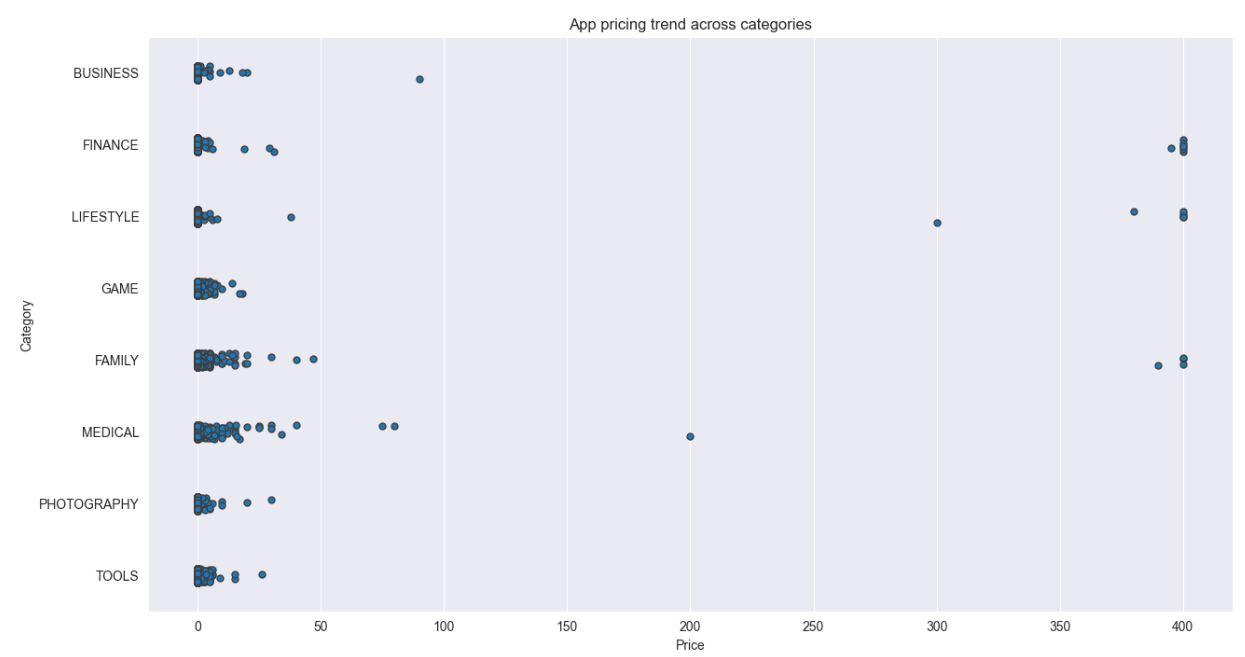
****

****

**7) Relation between app category and app price**

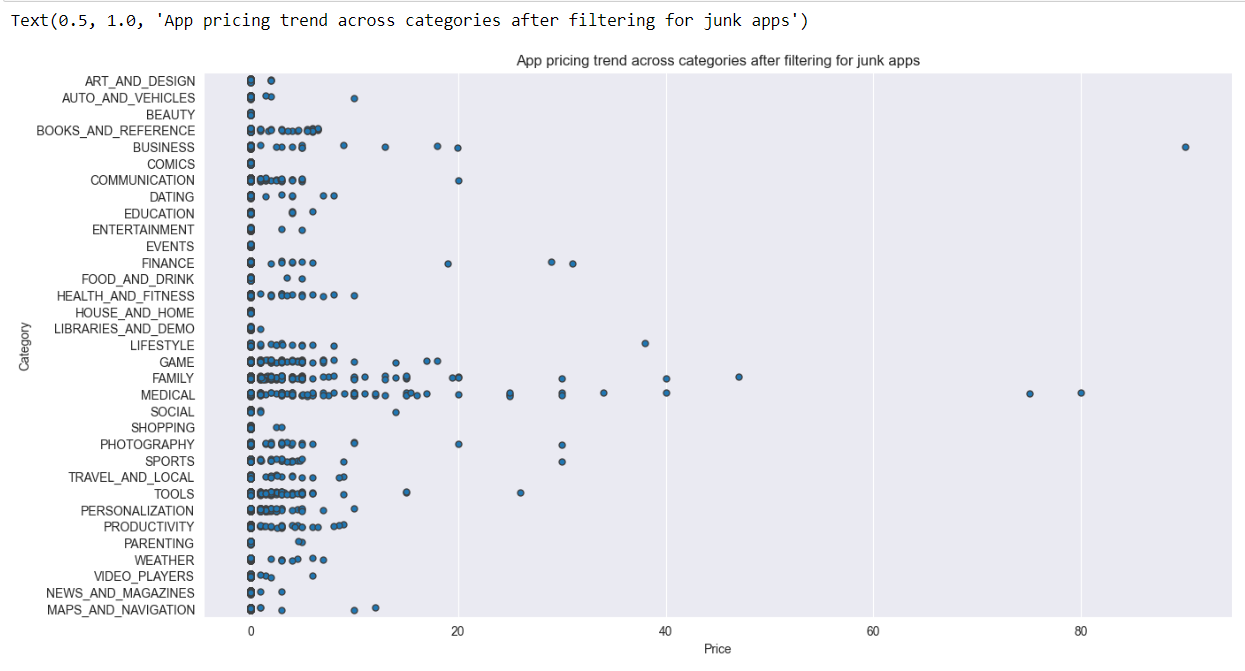
****

****

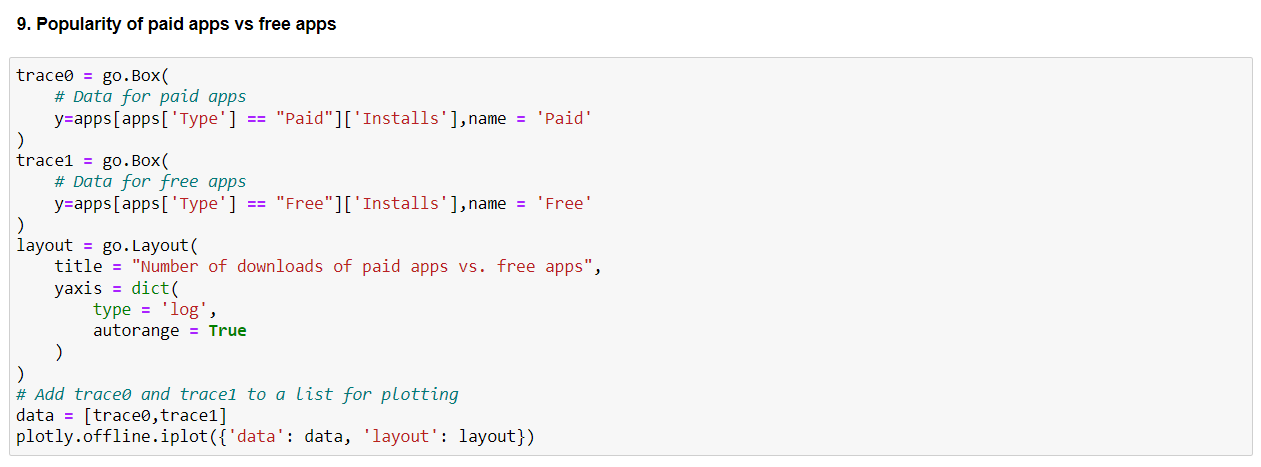
****

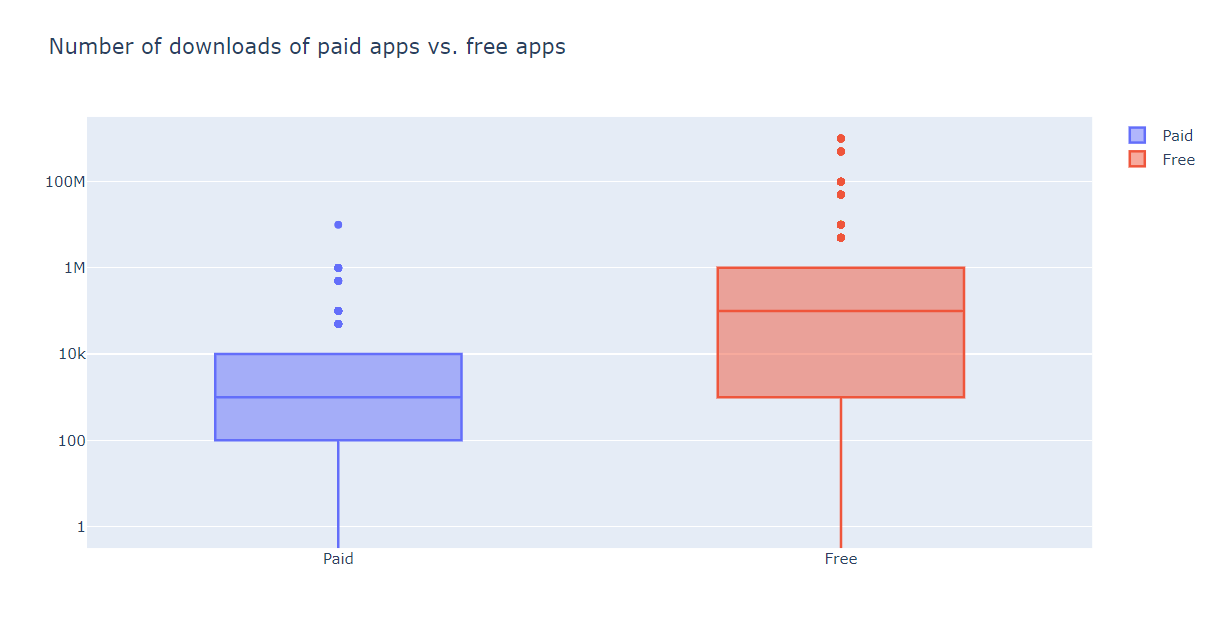
**8) Filter out junk apps**

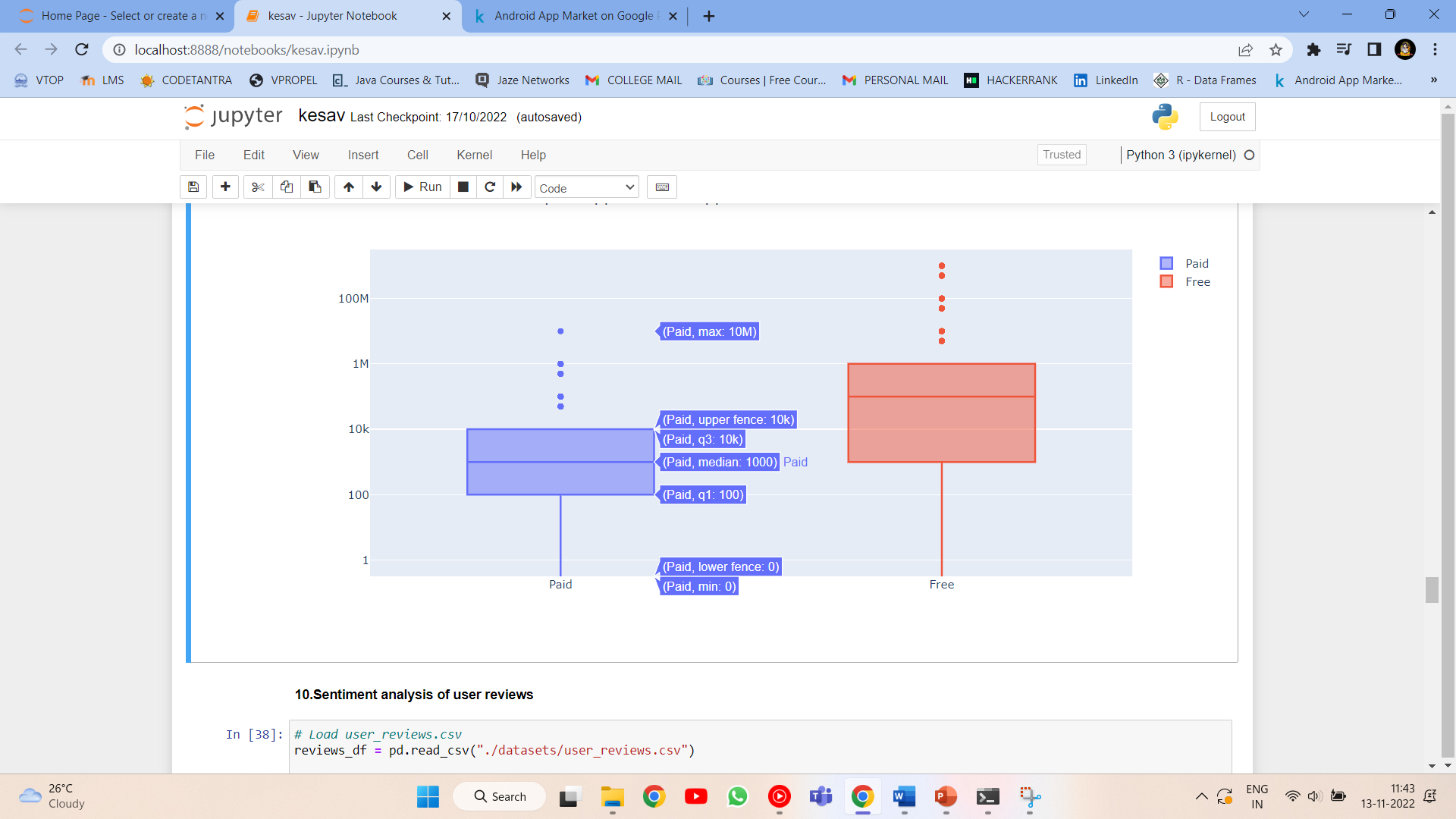
****

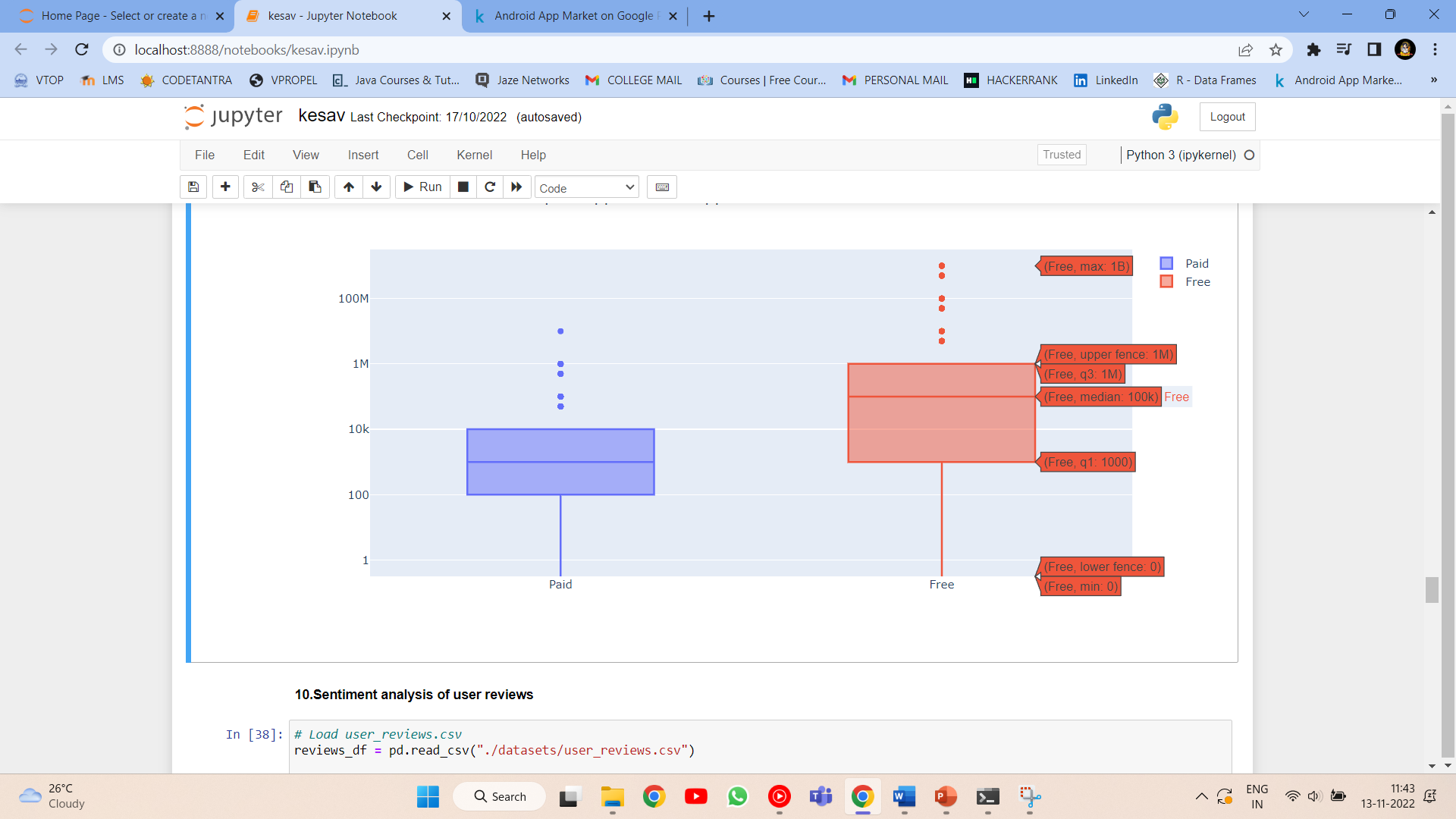
****

**9) Popularity of paid apps vs free apps**

****

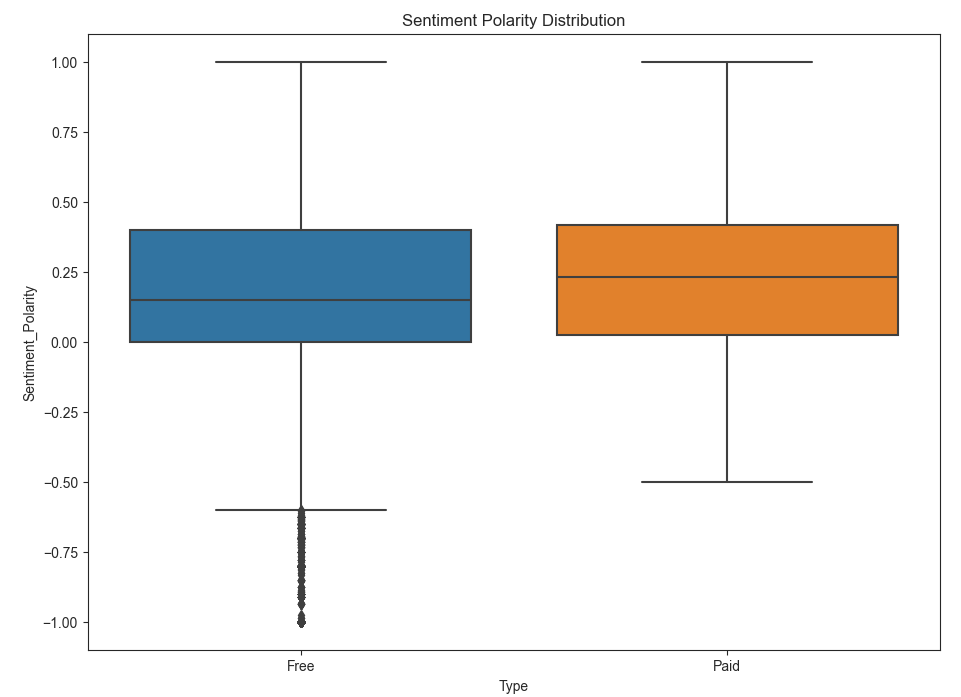
****

****

****

**10) Sentiment analysis of user reviews**

****

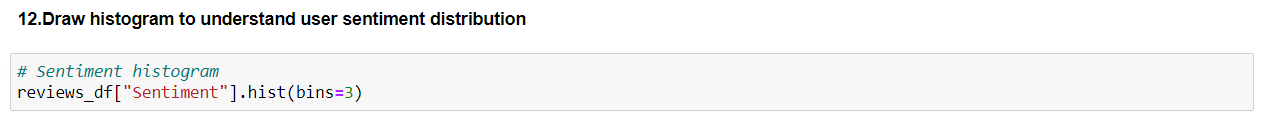
****

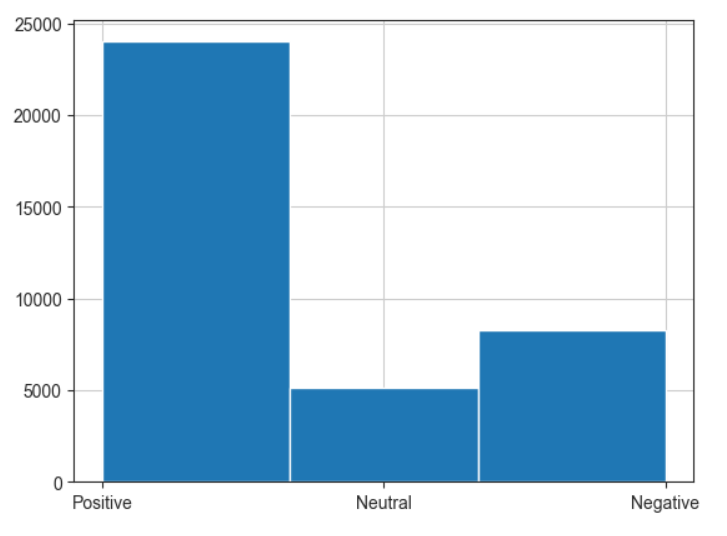
**11) Remove null values from user reviews**

****

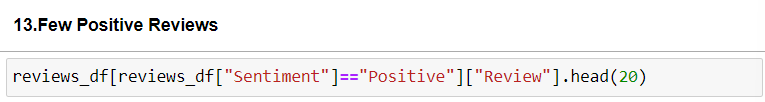
****

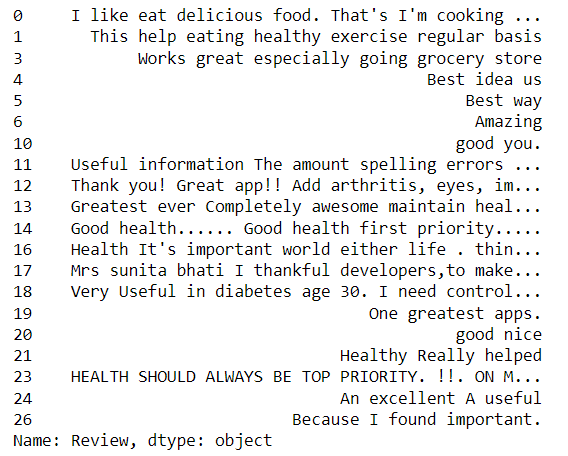
**12) Draw histogram to understand user sentiment distribution**

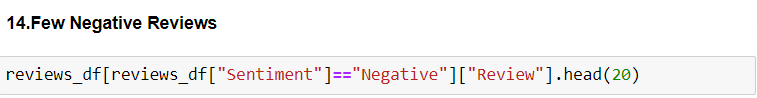
****

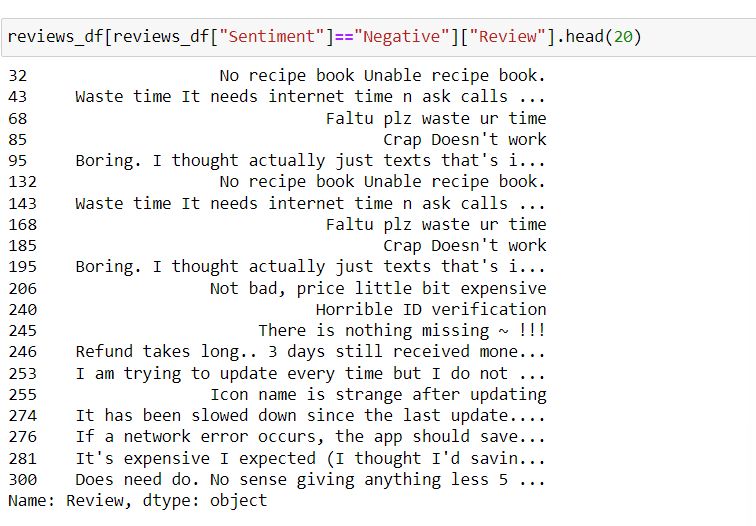
****

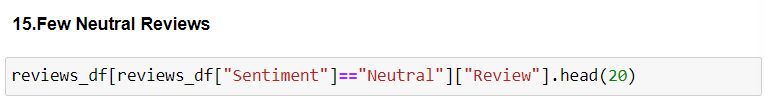
**13) Few positive, neutral and negative reviews**

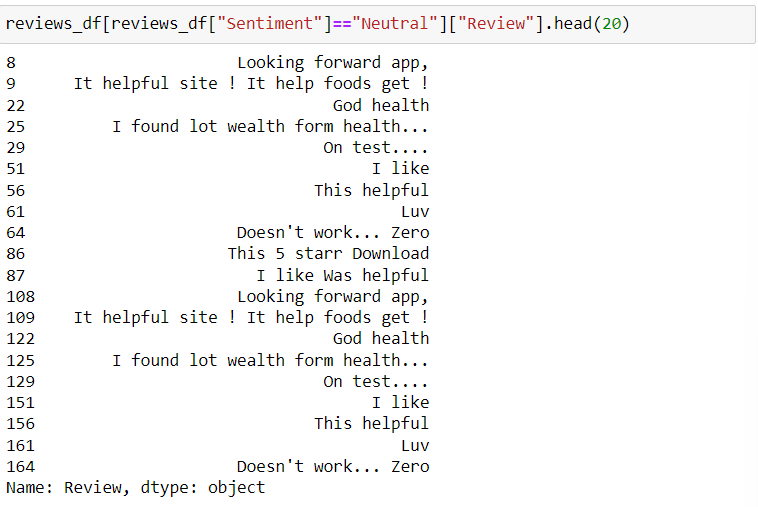












**Performance Metrics**

****

***CONCLUSION***

In this project, we analysed the app features such as size, price, ratings, reviews and category of apps found on Play Store. Our project aimed at finding out the factors which result in finding out which app is good and which app is bad. We studied about the distribution of app reviews into positive, negative and neutral that is sentimental analysis of user reviews. At first, we performed data cleaning by removing all the columns which were not needed for our analysis and then we removed the symbols in certain columns which was preventing us from performing the operations. We compared the popularity of paid and free apps and found out that free apps are more downloaded than free apps. We plotted the price of apps for each category but removed the apps which costs more than 200 dollars as they were junk apps. The variation of ratings with the size and price of an app is plotted. The distribution of app ratings from 0-5 with number of apps with the particular rating is plotted using histogram. Number of apps in each category is plotted using histogram. This project is helpful for predicting whether the app is good in comparison to other apps in that category. The sentiment analysis would lead to improving the app by the app developers after going through all the negative polarized reviews.

***FUTURE WORKS***

In our project, we can add proper sentiment analysis of comments using Machine Learning and Deep Learning by checking for certain keywords and making the machine understand by itself. Having a certain polarity for each word and calculating the total polarity of the sentence by adding all the polarity of the apps with the sign. The rating analysis could be extended to the games, books, movies and tv shows available on Play Store. The best app for a user could be predicted, the game which the user would like to play, the book which the user would like to read and learn, the movies and television shows which the user would love to watch and all can be predicted using the project in future. In the paid apps, we only removed apps which cost more than 200 dollars. There are many other apps which cost less but are junk apps like ‘I am rich’.

***APPENDIX (CODE)***

#Importing panda package

import pandas as pd

#Reading the dataset apps.csv

apps=apps\_with\_duplicates = pd.read\_csv(‘./datasets/apps.csv’)

#Printing total number of apps

print(‘Total number of apps in the dataset = ‘, apps.shape[0])

#A glimpse of the dataset(i.e ten rows)

print(apps.sample(10))

#Unnecessary columns are removed

apps.pop(‘Android Ver’)

apps.pop(‘Current Ver’)

apps.pop(‘Last Updated’)

#The list of characters which have to be removed from the data

chars\_to\_remove=[‘+’,’,’,’$’]

#Certain columns which contain data with the characters that has to be removed

cols\_to\_remove=[‘Installs’,’Price’]

for i in cols\_to\_remove:

for j in chars\_to\_remove:

apps[i]=apps[i].apply(lambda x: x.replace(j, ‘’))

#Column details are printed

print(apps.info())

#Importing numpy library

import numpy as np

# Convert Installs to float data type

apps[‘Installs’] = apps[‘Installs’].astype(‘float’)

# Convert Price to float data type

apps[‘Price’] = apps[‘Price’].astype(‘float’)

# Checking datatypes of apps.csv

print(apps.info())

#Imprted plotly and connected

import plotly

plotly.offline.init\_notebook\_mode(connected=True)

import plotly.graph\_objs as go

# Print the total number of unique categories.

Num\_cat=len(apps[‘Category’].unique())

print(‘Number of categories = ‘,num\_cat)

# Count the number of apps in each category.

Numapps\_in\_cat=apps[‘Category’].value\_counts()

# Sort num\_apps\_in\_category in descending order based on the count of apps in each category

sorted\_numapps\_in\_cat= numapps\_in\_cat.sort\_values(ascending = False)

data = [go.Bar(x=numapps\_in\_cat.index,y=numapps\_in\_cat.values,)]

plotly.offline.iplot(data)

# Average rating of apps

avg\_app\_rating=apps[‘Rating’].mean()

print(‘Average app rating = ‘,avg\_app\_rating)

# Distribution of apps according to their ratings

data=[go.Histogram(x=apps[‘Rating’])]

# Vertical dashed line to indicate the average app rating

layout={‘shapes’:[{‘type’:’line’,’x0’:avg\_app\_rating,’y0’:0,

‘x1’:avg\_app\_rating,’y1’:1000,’line’:{‘dash’:’dashdot’}}]}

plotly.offline.iplot({‘data’:data,’layout’:layout})

%matplotlib inline

import seaborn as sns

sns.set\_style(“darkgrid”)

import warnings

warnings.filterwarnings(“ignore”)

# Select rows where both Rating and Size values are not null

apps\_with\_size\_and\_rating=apps[apps.Rating.notnull()&apps.Size.notnull()]

# Subset for categories with at least 500 apps

large\_categories=apps\_with\_size\_and\_rating.groupby(‘Category’).filter(lambda x:len(x)>=500)

# Plot size vs rating

plt1=sns.jointplot(x=large\_categories[‘Size’],y=large\_categories[‘Rating’])

# Select apps whose Type is Paid

paidapps=apps\_with\_size\_and\_rating[apps\_with\_size\_and\_rating[‘Type’]==’Paid’]

# Plot price vs rating

plt2=sns.jointplot(x=paidapps[‘Price’],y=paidapps[‘Rating’])

import matplotlib.pyplot as plt

import numpy as np

from sklearn.metrics import accuracy\_score

from sklearn.linear\_model import LinearRegression

m=[]

l=[]

m=apps[‘Installs’]

l=apps[‘Reviews’]

m=np.array(m).reshape(-1,1)

l=np.array(l)

model = LinearRegression().fit(m,l)

print(model)

y\_pred = model.predict(np.array([[10000]]))

print(y\_pred)

import matplotlib.pyplot as plt

fig, ax = plt.subplots()

fig.set\_size\_inches(15, 8)

#Select few popular app categories

popular\_app\_cats = apps[apps.Category.isin([‘GAME’, ‘FAMILY’, ‘PHOTOGRAPHY’,’MEDICAL’, ‘TOOLS’, ‘FINANCE’, ‘LIFESTYLE’,’BUSINESS’])]

# Examine the price trend by plotting price vs category

ax = sns.stripplot(x = popular\_app\_cats[‘Price’], y = popular\_app\_cats[‘Category’], jitter=True, linewidth=1)

ax.set\_title(‘App pricing trend across categories’)

# Apps whose Price is greater than 200

apps\_above\_200 = apps[apps[‘Price’] > 200]

apps\_above\_200[[‘Category’, ‘App’, ‘Price’]]

# Select apps priced below $100

apps\_under\_100 = apps[apps.Price < 100]

fig, ax = plt.subplots()

fig.set\_size\_inches(15, 8)

# Examine price vs category with the authentic apps (apps\_under\_100)

ax = sns.stripplot(x = ‘Price’, y = ‘Category’, data = apps\_under\_100, jitter = True, linewidth = 1)

ax.set\_title(‘App pricing trend across categories after filtering for junk apps’)

trace0 = go.Box(

# Data for paid apps

y=apps[apps[‘Type’] == “Paid”][‘Installs’],name = ‘Paid’)

trace1 = go.Box(

# Data for free apps

y=apps[apps[‘Type’] == “Free”][‘Installs’],name = ‘Free’)

layout = go.Layout(

title = “Number of downloads of paid apps vs. free apps”,

yaxis = dict(

type = ‘log’,

autorange = True))

# Add trace0 and trace1 to a list for plotting

data = [trace0,trace1]

plotly.offline.iplot({‘data’: data, ‘layout’: layout})

# Load user\_reviews.csv

reviews\_df = pd.read\_csv(“./datasets/user\_reviews.csv”)

# Join the two dataframes

merged\_df = pd.merge(apps, reviews\_df, on = “App”, how = “inner”)

# Drop NA values from Sentiment and Review columns

merged\_df = merged\_df.dropna(subset = [‘Sentiment’, ‘Review’])

sns.set\_style(‘ticks’)

fig, ax = plt.subplots()

fig.set\_size\_inches(11, 8)

# User review sentiment polarity for paid vs. free apps

ax = sns.boxplot(x = ‘Type’, y = ‘Sentiment\_Polarity’, data = merged\_df)

ax.set\_title(‘Sentiment Polarity Distribution’)

reviews\_df=reviews\_df.dropna()

reviews\_df

# Sentiment histogram

reviews\_df[“Sentiment”].hist(bins=3)

reviews\_df[reviews\_df[“Sentiment”]==”Positive”][“Review”].head(2)

reviews\_df[reviews\_df[“Sentiment”]==”Negative”][“Review”].head(20)

reviews\_df[reviews\_df[“Sentiment”]==”Neutral”][“Review”].head(20)

***REFERENCES***

1) <https://www.kaggle.com/datasets/utshabkumarghosh/android-app-market-on-google-play?resource=download>

2) <https://www.dataquest.io/blog/jupyter-notebook-tutorial/>

3) <https://365datascience.com/tutorials/python-tutorials/jupyter-notebook-tutorial/>

4)<https://www.irjmets.com/uploadedfiles/paper/issue_6_june_2022/26972/final/fin_irjmets1656231688.pdf>

5)<https://www.researchgate.net/publication/322013014_Insights_From_Google_Play_Store_User_Reviews_for_the_Development_of_Weight_Loss_Apps_Mixed-Method_Analysis>

***WHAT WE HAVE DONE***

We have uploaded our project which we did in the jupyter notebook (Python) file in github and pasted the link below. The jupyter notebook file consists of all the modules which we covered in our project in order.

GITHUB LINK- <https://github.com/kesavsanthosh3/FDA-PROJECT/>

***THANK YOU***