

Project Based Learning Report

On

GOOGLE PLAYSTORE DATASET ANALYSIS USING PYTHON

Submitted in the partial fulfillment of the requirements

For the Project based learning in (Essentials Of Data
Science)

In

Electronics & Communication Engineering

By

2014111123 Satyam Suresh

2014111107 Shashank

2014111097 Vivek Nagar

Under the guidance of Course In-charge

Prof. Dnyanesh S.Lavhkare

Department of Electronics & Communication Engineering

Bharati Vidyapeeth
(Deemed to be University)
College of Engineering,
Pune – 4110043

Academic Year: 2021-22

**Bharati Vidyapeeth
(Deemed to be University)
College of Engineering,
Pune – 4110436**

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

CERTIFICATE

Certified that the Project Based Learning report entitled, “**Google Playstore Dataset Analysis Using Python_**” is work done by

2014111123	Satyam Suresh
2014111107	Shashank
2014111097	Vivek Nagar

in partial fulfillment of the requirements for the award of credits for Project Based Learning (PBL) in **Essentials of Data Science Course** of Bachelor of Technology Semester IV, Electronics & Communication Engineering.

Date: 24 May 2022

Prof. Dnyanesh S.Lavhkare
Course In-charge

Dr. Tanuja S.Dhope
PBL Co-Ordinator

Dr. Arundhati A.Shinde
Professor & Head
ELECTRONICS & COMMUNICATION ENGINEERING

Index

Page No.	Contents
1-1	Problem Statement with Solution
2-5	Description about project
6-6	Software Used
7-15	Results with Analysis
16	Conclusion & Outcome

Problem Statement :-

What is Data Science? Why learn Data Science?

Solution :-

Data science is the domain of study that deals with vast volumes of data using modern tools and techniques to find unseen patterns, derive meaningful information, and make business decisions. Data science uses complex machine learning algorithms to build predictive models. The data used for analysis can come from many different sources and presented in various formats.

Data science is the field of study that combines domain expertise, programming skills, and knowledge of mathematics and statistics to extract meaningful insights from data. Data science practitioners apply machine learning algorithms to numbers, text, images, video, audio, and more to produce artificial intelligence (AI) systems to perform tasks that ordinarily require human intelligence. In turn, these systems generate insights which analysts and business users can translate into tangible business value.

Reasons to learn Data Science are: -

- 1) Learning about data science provides an opportunity for you to recreate yourself.
- 2) We live in a digital world, everything is data-driven. There is data science in business, accounting, education, science, engineering, healthcare, technology, energy sector, government, and so on.
- 3) Data science is also a very promising field with lots of high paying job opportunities.
- 4) Basic data science skills are important for personal use.
- 5) Great potential to branch out with different options.
- 6) Become a decision-maker, not every job opportunity will give you the power to make informed business decisions. For a data scientist, that is the core responsibility.
- 7) Less competitive because it is a highly analytical role, competition is less, but demand is not. With a limited talent pool, there is always a challenge for businesses to hire in these roles.

Google Playstore Analysis

Google Play, also branded as the Google Play Store and formerly Android Market, is a digital distribution service operated and developed by Google. It serves as the official app store for certified devices running on the Android operating system and its derivatives as well as Chrome OS, allowing users to browse and download applications developed with the Android software development kit (SDK) and published through Google. Google Play also serves as a digital media store, offering music, books, movies, and television programs.^[4] Content that has been purchased on Google Play Movies & TV and Google Play Books can be accessed on a web browser, and through the Android and iOS apps.



Applications are available through Google Play either free of charge or at a cost. They can be downloaded directly on an Android device through the proprietary Google Play Store mobile app or by deploying the application to a device from the Google Play website. Applications utilizing hardware capabilities of a device can be targeted to users of devices with specific hardware components, such as a motion sensor (for motion-dependent games) or a front-facing camera (for online video calling). The Google Play Store had over 82 billion app downloads in 2016 and reached over 3.5 million apps published in 2017,^[5] while after a purge of apps is back to over 3 million.^[6] It has been the subject of multiple issues concerning security, in which malicious software has been approved and uploaded to the store and downloaded by users, with varying degrees of severity.

Google Play was launched on March 6, 2012, bringing together Android Market, Google Music, Google Movies and the Google eBookstore under one brand, marking a shift in Google's digital distribution strategy. Following their re-branding, Google has expanded the geographical support for each of the services. Since 2018, Google has gradually sunsetted the Play brand. Play

Newsstand was rebranded as Google News in 2018, Play Music was discontinued in favor of YouTube Music in 2020, and Play Movies & TV was rebranded as Google TV in 2021. In 2022, Play Games is expected to shut down its mobile app in favor of an Android emulator for Windows with the same name.^[7] The remaining standalone mobile app will be Play Books.

Datasets: -

We have downloaded one dataset googleplaystore.csv about Google Playstore from Kaggle site and performed analysis in Jupyter Notebook.

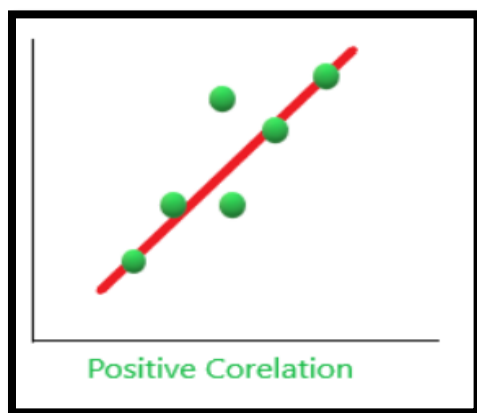
Libraries used: -

- 1) Numpy library - numpy is used to perform various mathematical operations on arrays.
- 2) Pandas Library - pandas provides various data structures and operations for manipulating numerical data and time series.
- 3) Matplotlib library from which pyplot module is used for plotting library used for 2D graphics.
- 4) Seaborn library - seaborn is a library for making statistical graphics in Python.

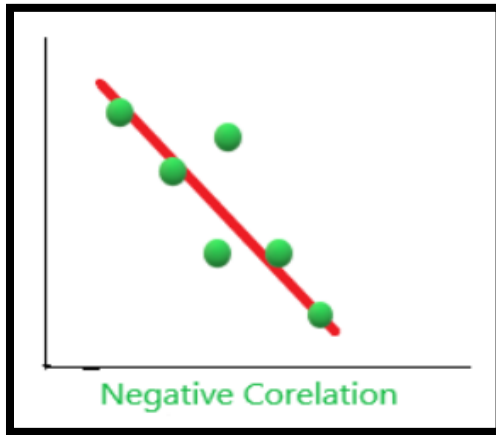
Correlation: -

Correlation means an association; it is a measure of the extent to which two variables are related.

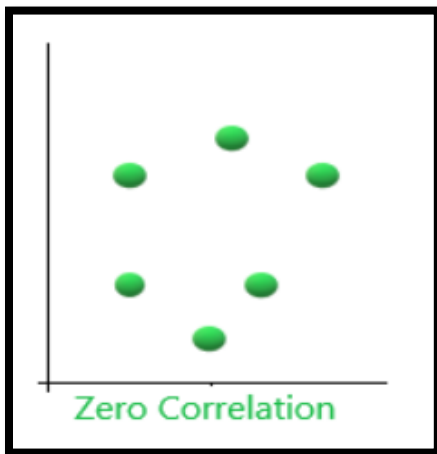
1. **Positive Correlation:** When two variables increase together and decrease together. They are positively correlated. '1' is a perfect positive correlation. For example – demand and profit are positively correlated the more the demand for the product, the more profit hence positive correlation.



2. **Negative Correlation:** When one variable increase and the other variable decreases together and vice-versa. They are negatively correlated. For example, If the distance between magnet increases their attraction decreases, and vice-versa. Hence, a negative correlation. '-1' is no correlation.



3. **Zero Correlation (No Correlation):** When two variables don't seem to be linked at all. '0' is a perfect negative correlation. For Example, the amount of tea you take and level of intelligence.



4. **Box plot:** A box plot helps to maintain the distribution of quantitative data in such a way that it facilitates the comparisons between variables or across levels of a categorical variable. The main body of the box plot showing the quartiles and the median's confidence intervals if enabled. The medians have horizontal lines at the median of each

box and while whiskers have the vertical lines extending to the most extreme, non-outlier data points and caps are the horizontal lines at the ends of the whiskers.

5. **Countplot:** `seaborn.countplot()` method is used to Show the counts of observations in each categorical bin using bars.
6. **Line plot :** The `plot()` function in `pyplot` module of `matplotlib` library is used to make a 2D hexagonal binning plot of points `x`, `y`.

Software Used

Jupyter Notebook: -

Jupyter Notebook (formerly IPython Notebooks) is a web-based interactive computational environment for creating notebook documents. A Jupyter Notebook document is a browser-based REPL containing an ordered list of input/output cells which can contain code, text (using Markdown), mathematics, plots and rich media. Underneath the interface, a notebook is a JSON document, following a versioned schema, usually ending with the “. ipynb” extension. Jupyter notebooks are built upon a number of popular open-source libraries.

Jupyter Notebook can connect to many *kernels* to allow programming in different languages. A Jupyter kernel is a program responsible for handling various types of requests (code execution, code completions, inspection), and providing a reply. Kernels talk to the other components of

Jupyter

using ZeroMQ, and

thus can be on the

same or remote

machines. Unlike

many other

Notebook-like

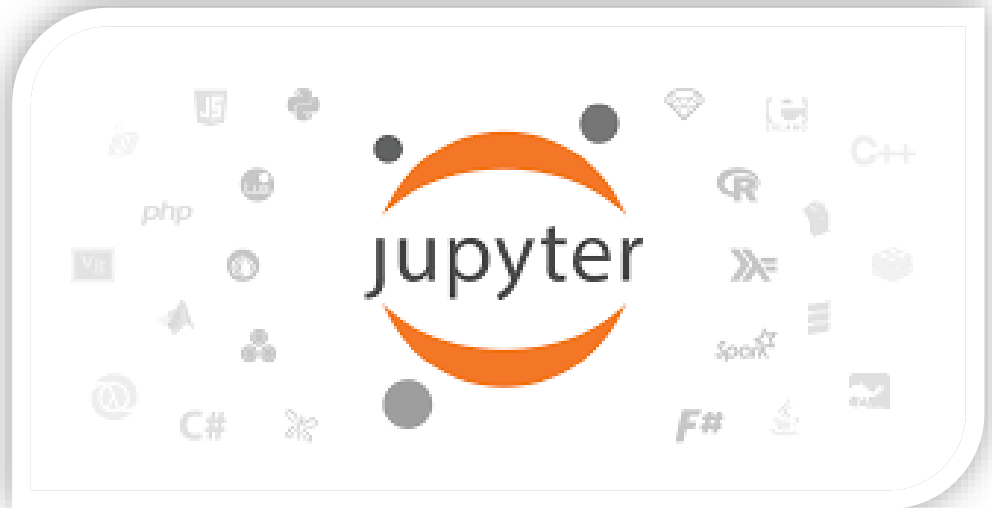
interfaces, in Jupyter,

kernels are not aware

that they are attached

to a specific

document, and can be connected to many clients at once. Usually kernels allow execution of only a single language, but there are a couple of exceptions. By default Jupyter Notebook ships with the IPython kernel. As of the 2.3 release (October 2014), there are 49 Jupyter-compatible kernels for many programming languages, including Python, R, Julia and Haskell.

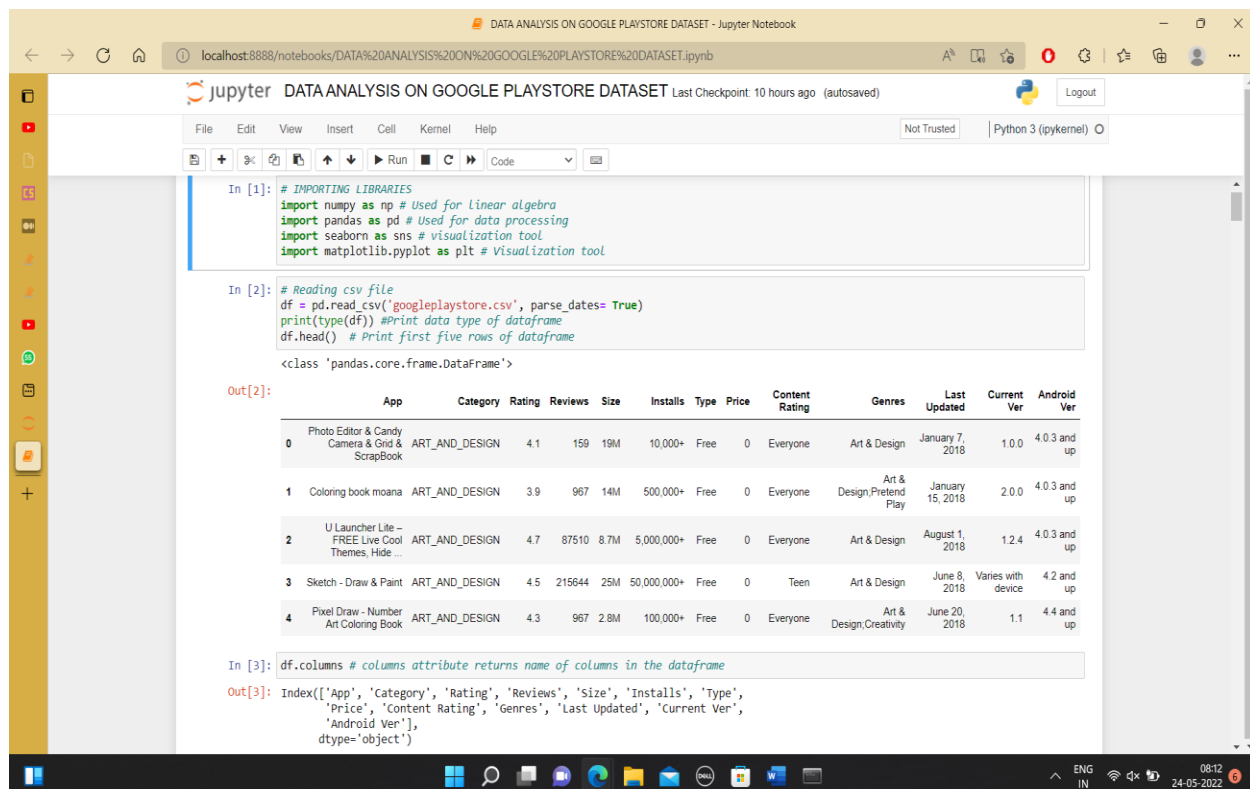


Result with Analysis

Analysis of the code: -

- First, we have imported four libraries – numpy as np, pandas as pd, matplotlib.pyplot as plt and seaborn library as sns.
- Secondly, we have loaded our dataset – googleplaystore.csv using read_csv() function of pandas library and used head() function for displaying first five rows of the dataset.
- Then, for checking null values in the dataset, we have used isnull() function of pandas library.
- We have changed the data type of Installs, Price, Reviews, Rating, Size and Last Updated column using the lambda function.
- After this, we have created a Correlation Map using heatmap() function of Seaborn library to check any type of relation between some of the columns.
- Then, we used plot() function of Matplotlib library, to plot all the data for the Price, Reviews, Installs and Rating. These data are plotted in comparison with each category. Here we analyse the top value in each plot.
- At last, we used countplot() function of Seaborn library to create a countplot for Content Rating.

Screenshots of code: -



```
In [1]: # IMPORTING LIBRARIES
import numpy as np # Used for linear algebra
import pandas as pd # Used for data processing
import seaborn as sns # visualization tool
import matplotlib.pyplot as plt # Visualization tool

In [2]: # Reading csv file
df = pd.read_csv('googleplaystore.csv', parse_dates=True)
print(type(df)) #Print data type of dataframe
df.head() # Print first five rows of dataframe

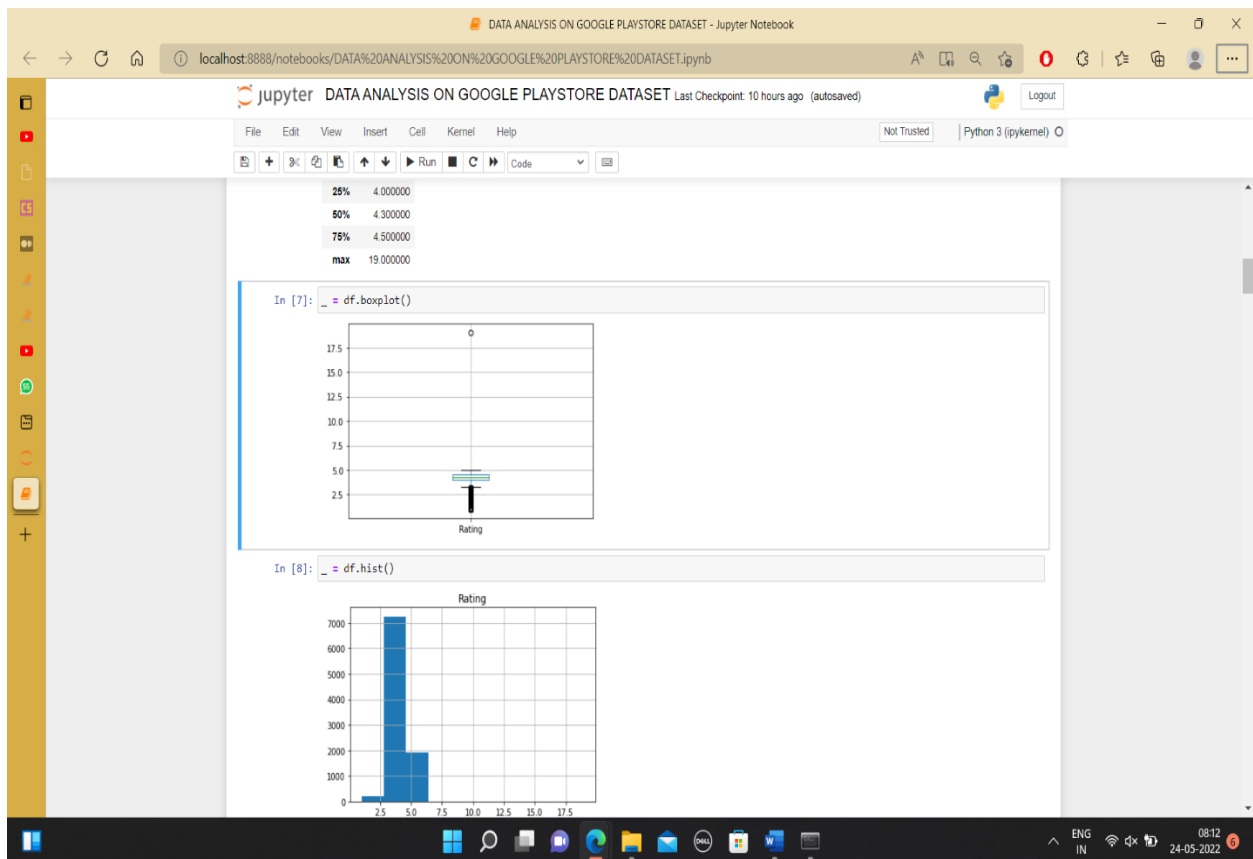
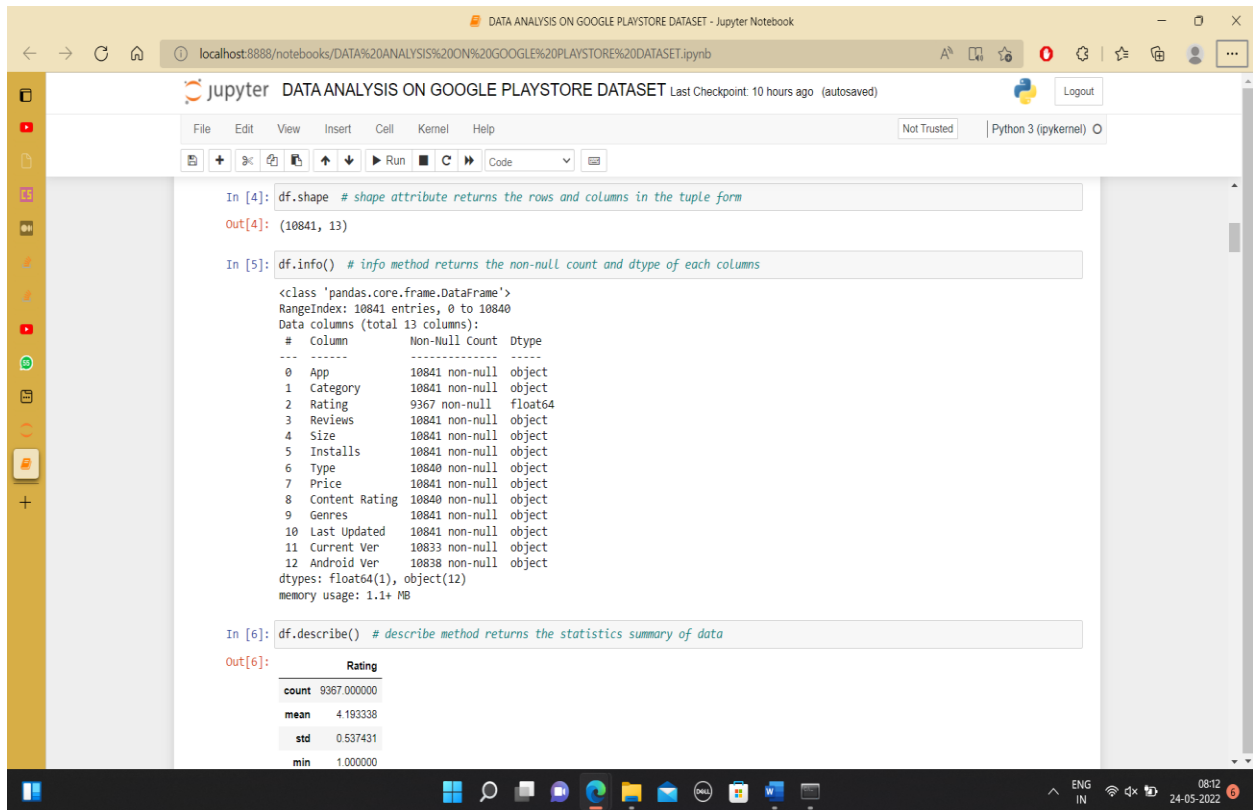
<class 'pandas.core.frame.DataFrame'>

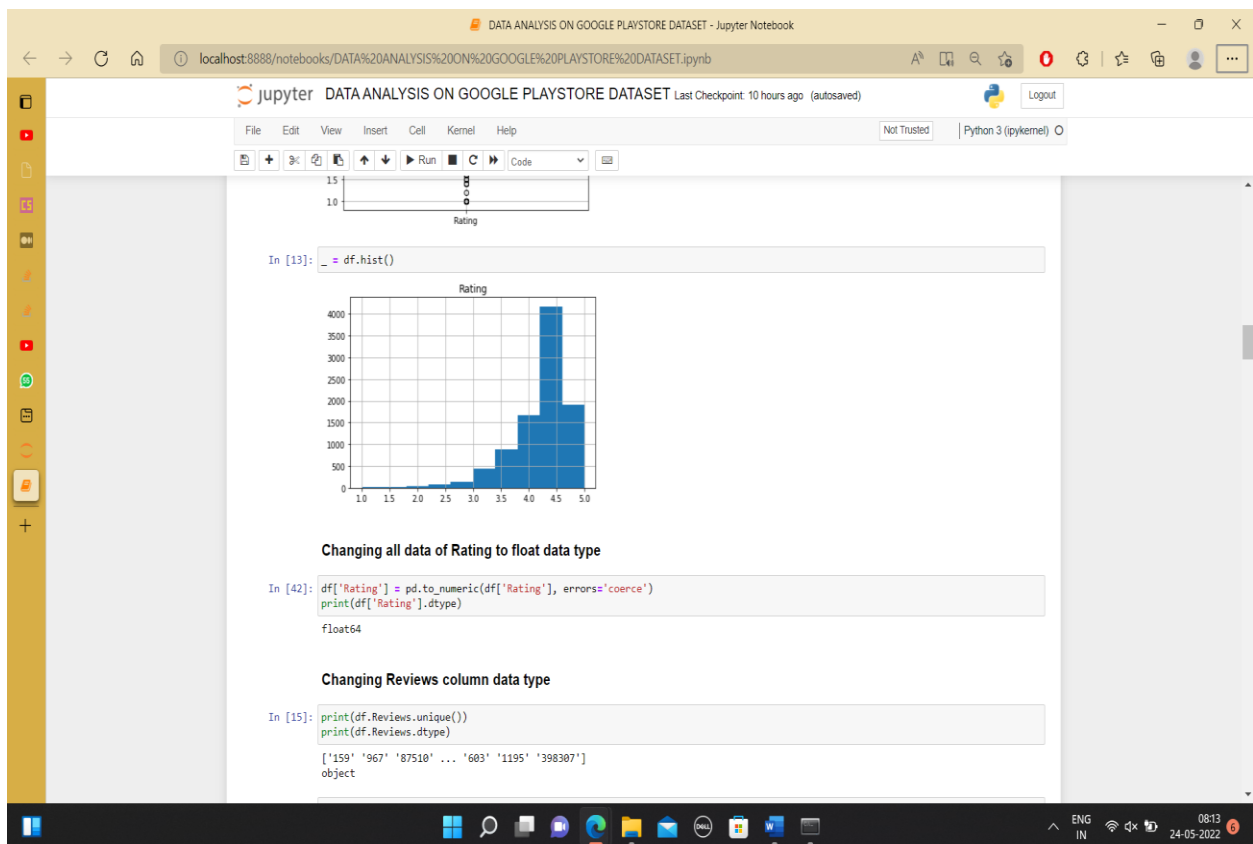
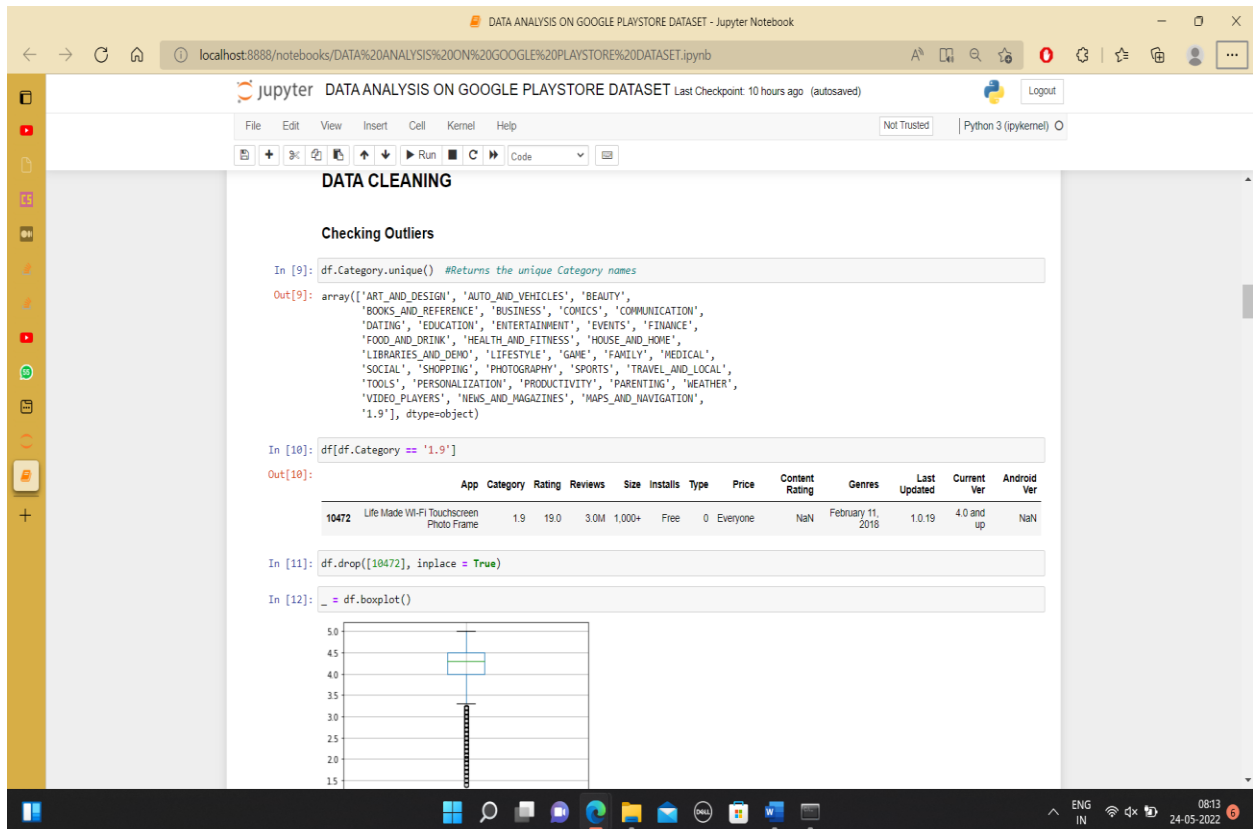
Out[2]:
```

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	Genres	Last Updated	Current Ver	Android Ver
0	Photo Editor & Candy Camera & Grid & ScrapBook	ART_AND_DESIGN	4.1	159	19M	10,000+	Free	0	Everyone	Art & Design	January 7, 2018	1.0.0	4.0.3 and up
1	Coloring book moana	ART_AND_DESIGN	3.9	967	14M	500,000+	Free	0	Everyone	Art & Design, Pretend Play	January 15, 2018	2.0.0	4.0.3 and up
2	U Launcher Lite – FREE Live Cool Themes, Hide ...	ART_AND_DESIGN	4.7	87510	8.7M	5,000,000+	Free	0	Everyone	Art & Design	August 1, 2018	1.2.4	4.0.3 and up
3	Sketch - Draw & Paint	ART_AND_DESIGN	4.5	215644	25M	50,000,000+	Free	0	Teen	Art & Design	June 8, 2018	Varies with device	4.2 and up
4	Pixel Draw - Number Art Coloring Book	ART_AND_DESIGN	4.3	967	2.8M	100,000+	Free	0	Everyone	Art & Design, Creativity	June 20, 2018	1.1	4.4 and up

```
In [3]: df.columns # columns attribute returns name of columns in the dataframe

Out[3]: Index(['App', 'Category', 'Rating', 'Reviews', 'Size', 'Installs', 'Type', 'Price', 'Content Rating', 'Genres', 'Last Updated', 'Current Ver', 'Android Ver'],
              dtype='object')
```





DATA ANALYSIS ON GOOGLE PLAYSTORE DATASET - Jupyter Notebook

localhost:8888/notebooks/DATA%20ANALYSIS%20ON%20GOOGLE%20PLAYSTORE%20DATASET.ipynb

Jupyter DATA ANALYSIS ON GOOGLE PLAYSTORE DATASET Last Checkpoint: 10 hours ago (autosaved)

File Edit View Insert Cell Kernel Help Not Trusted Python 3 (ipykernel)

```
In [16]: df.Reviews = df.Reviews.astype(float)
print(df.Reviews.dtype)

float64
```

Changing Size column data type

```
In [17]: print(df.Size.unique())
print(df.Size.dtype)

['19M' '14M' '8.7M' '25M' '2.8M' '5.6M' '29M' '33M' '3.1M' '28M' '12M'
'20M' '21M' '37M' '2.7M' '5.5M' '17M' '39M' '31M' '4.2M' '7.0M' '23M'
'6.0M' '6.1M' '4.6M' '9.2M' '5.2M' '11M' '24M' 'Varies with device'
'9.4M' '15M' '10M' '1.2M' '26M' '8.0M' '7.9M' '56M' '57M' '35M' '54M'
'201k' '3.6M' '5.7M' '8.6M' '2.4M' '27M' '2.5M' '16M' '3.4M' '8.9M'
'3.9M' '2.9M' '38M' '32M' '5.4M' '18M' '1.1M' '2.2M' '4.5M' '9.8M' '52M'
'9.0M' '6.7M' '30M' '2.6M' '7.1M' '3.7M' '22M' '7.4M' '6.4M' '3.2M'
'8.2M' '9.9M' '4.9M' '9.5M' '5.0M' '5.9M' '13M' '73M' '6.8M' '3.5M'
'4.0M' '2.3M' '7.2M' '2.1M' '42M' '7.3M' '9.1M' '55M' '23k' '6.5M' '1.5M'
'7.5M' '51M' '41M' '48M' '8.5M' '46M' '8.3M' '4.3M' '4.7M' '3.3M' '40M'
'7.8M' '8.0M' '6.6M' '5.1M' '61M' '60M' '79k' '8.4M' '118k' '44M' '695k'
'1.6M' '6.2M' '18k' '53M' '1.4M' '3.0M' '5.0M' '3.0M' '9.6M' '45M' '63M'
'49M' '77M' '4.4M' '4.0M' '70M' '6.9M' '9.3M' '10.0M' '8.1M' '39M' '84M'
'97M' '2.0M' '1.9M' '1.0M' '5.3M' '47M' '556k' '526k' '76M' '7.6M' '59M'
'9.7M' '78M' '72M' '43M' '7.7M' '6.3M' '334k' '34M' '93M' '65M' '79M'
'100M' '58M' '50M' '68M' '64M' '67M' '60M' '94M' '232k' '99M' '624k'
'95M' '8.5k' '41k' '292k' '11k' '80M' '1.7M' '74M' '62M' '69M' '75M'
'98M' '85M' '82M' '96M' '87M' '71M' '86M' '91M' '81M' '92M' '83M' '88M'
'704k' '862k' '899k' '378k' '266k' '375k' '1.3M' '975k' '980k' '4.1M']
float64
```

```
In [18]: df.Size = df.Size.apply(lambda x: str(x).replace("Varies with device",str(np.nan)) if "Varies with device" in str(x) else str(x))
df.Size = df.Size.apply(lambda x: str(x).replace("M","000") if "M" in str(x) else str(x)) # All size values became the kilobyte.
df.Size = df.Size.apply(lambda x: str(x).replace("k","") if "k" in str(x) else str(x))
df.Size = df.Size.astype(float)
print(df.Size.dtype)

float64
```

DATA ANALYSIS ON GOOGLE PLAYSTORE DATASET - Jupyter Notebook

localhost:8888/notebooks/DATA%20ANALYSIS%20ON%20GOOGLE%20PLAYSTORE%20DATASET.ipynb

Jupyter DATA ANALYSIS ON GOOGLE PLAYSTORE DATASET Last Checkpoint: 10 hours ago (autosaved)

File Edit View Insert Cell Kernel Help Not Trusted Python 3 (ipykernel)

```
In [19]: print(df.Installs.unique())
print(df.Installs.dtype)

['10,000+' '500,000+' '5,000,000+' '50,000,000+' '100,000+' '50,000+'
'1,000,000+' '10,000,000+' '5,000+' '100,000,000+' '1,000,000,000+'
'1,000+' '500,000,000+' '50+' '100+' '500+' '10+' '1+' '5+' '0+' '0']
object
```

```
In [20]: df.Installs = df.Installs.apply(lambda x: str(x).replace('+','')) if '+' in str(x) else str(x))
df.Installs = df.Installs.apply(lambda x: str(x).replace(',','')) if ',' in str(x) else str(x))
df.Installs = df.Installs.apply(lambda x: float(x))
print(df.Installs.dtype)

float64
```

Changing Price Column data type

```
In [21]: print(df.Price.unique())
print(df.Price.dtype)

['0' '$4.99' '$3.99' '$6.99' '$1.49' '$2.99' '$7.99' '$5.99' '$3.49'
'$1.99' '$9.99' '$7.49' '$0.99' '$9.00' '$5.49' '$10.00' '$24.99'
'$11.99' '$79.99' '$16.99' '$14.99' '$1.00' '$29.99' '$12.99' '$2.49'
'$10.99' '$1.50' '$19.99' '$15.99' '$33.99' '$74.99' '$30.99' '$3.95'
'$4.49' '$1.70' '$8.99' '$2.00' '$3.08' '$25.99' '$399.99' '$17.99'
'$400.00' '$3.02' '$1.76' '$4.84' '$4.77' '$1.01' '$2.50' '$1.59' '$6.49'
'$1.29' '$5.00' '$13.99' '$299.99' '$379.99' '$37.99' '$18.99' '$389.99'
'$19.90' '$8.49' '$1.75' '$14.00' '$4.85' '$46.99' '$109.99' '$154.99'
'$3.08' '$2.59' '$4.80' '$1.96' '$19.40' '$3.90' '$4.59' '$15.46' '$3.04'
'$4.29' '$2.60' '$3.28' '$4.60' '$28.99' '$2.95' '$2.90' '$1.97'
'$200.00' '$89.99' '$2.56' '$30.99' '$3.61' '$394.99' '$1.26' '$1.20'
'$1.04']
object
```

```
In [22]: df.Price = df.Price.apply(lambda x: str(x).replace('$','')) if '$' in str(x) else str(x))
df.Price = df.Price.apply(lambda x: float(x))
print(df.Price.dtype)
```

DATA ANALYSIS ON GOOGLE PLAYSTORE DATASET - Jupyter Notebook

localhost:8888/notebooks/DATA%20ANALYSIS%20ON%20GOOGLE%20PLAYSTORE%20DATASET.ipynb

Jupyter DATA ANALYSIS ON GOOGLE PLAYSTORE DATASET Last Checkpoint: 10 hours ago (autosaved) Logout

File Edit View Insert Cell Kernel Help Not Trusted Python 3 (ipykernel)

float64

Changing Last Updated Column data type

```
In [23]: print(df['Last Updated'].unique())
print(df['Last Updated'].dtype)

['January 7, 2018' 'January 15, 2018' 'August 1, 2018' ...
'January 20, 2014' 'February 16, 2014' 'March 23, 2014']
object
```

```
In [24]: df['Last Updated'] = pd.to_datetime(df['Last Updated']) # Convert string to datetime datatype
print(df['Last Updated'].dtype)
print(df['Last Updated'])

datetime64[ns]
0      2018-01-07
1      2018-01-15
2      2018-08-01
3      2018-06-08
4      2018-06-20
...
10836   2017-07-25
10837   2018-07-06
10838   2017-01-20
10839   2015-01-19
10840   2018-07-25
Name: Last Updated, Length: 10840, dtype: datetime64[ns]
```

Data Manipulation

```
In [25]: print(df.isnull().sum()) # isnull method replace all the data with boolean values. If the data is null then it changes to True.

App          0
Category     0
Rating      1474
Reviews      0
Size        1695
Type         0
Test-11     0
```

08:13 24-05-2022

DATA ANALYSIS ON GOOGLE PLAYSTORE DATASET - Jupyter Notebook

localhost:8888/notebooks/DATA%20ANALYSIS%20ON%20GOOGLE%20PLAYSTORE%20DATASET.ipynb

Jupyter DATA ANALYSIS ON GOOGLE PLAYSTORE DATASET Last Checkpoint: 10 hours ago (autosaved) Logout

File Edit View Insert Cell Kernel Help Not Trusted Python 3 (ipykernel)

```
price
Content Rating  0
Genres          0
Last Updated    0
Current Ver     8
Android Ver     2
dtype: int64
```

```
In [26]: def impute_median(series):
return series.fillna(series.median)
```

```
In [27]: df.Rating = df.Rating.transform(impute_median)
```

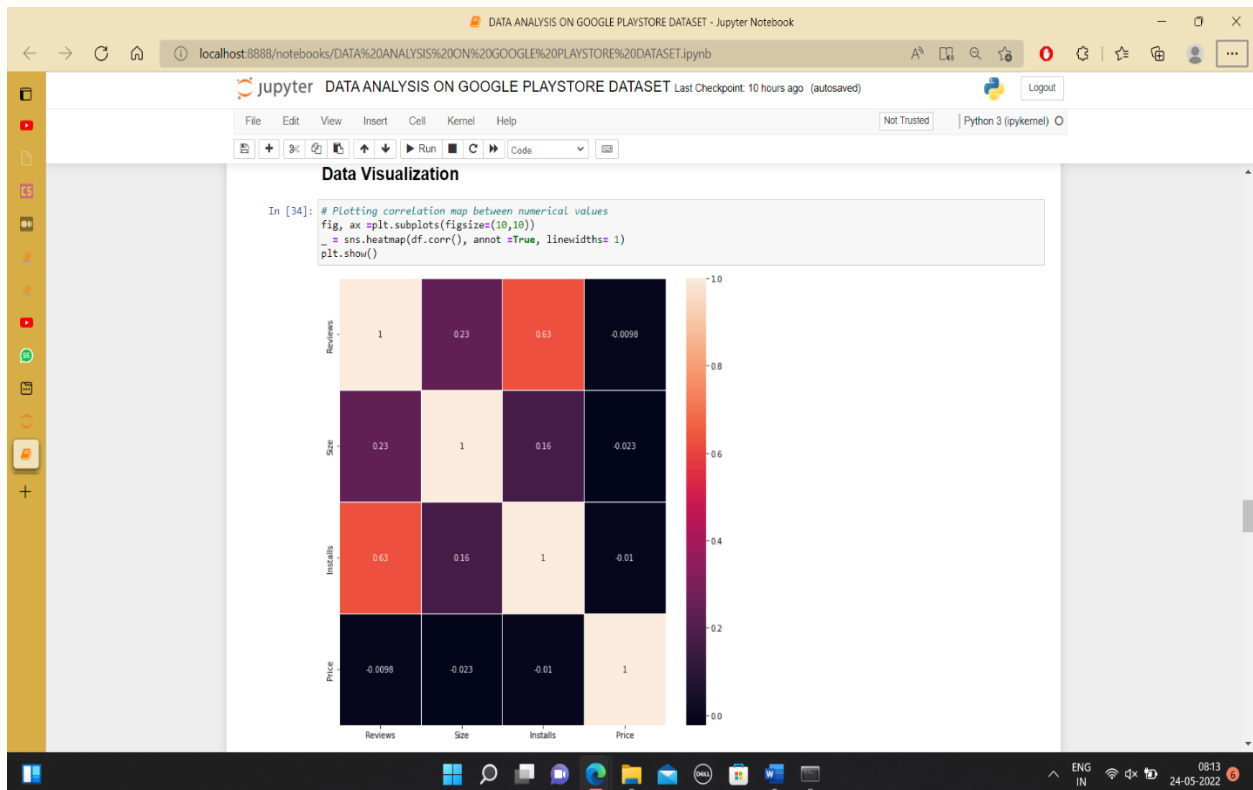
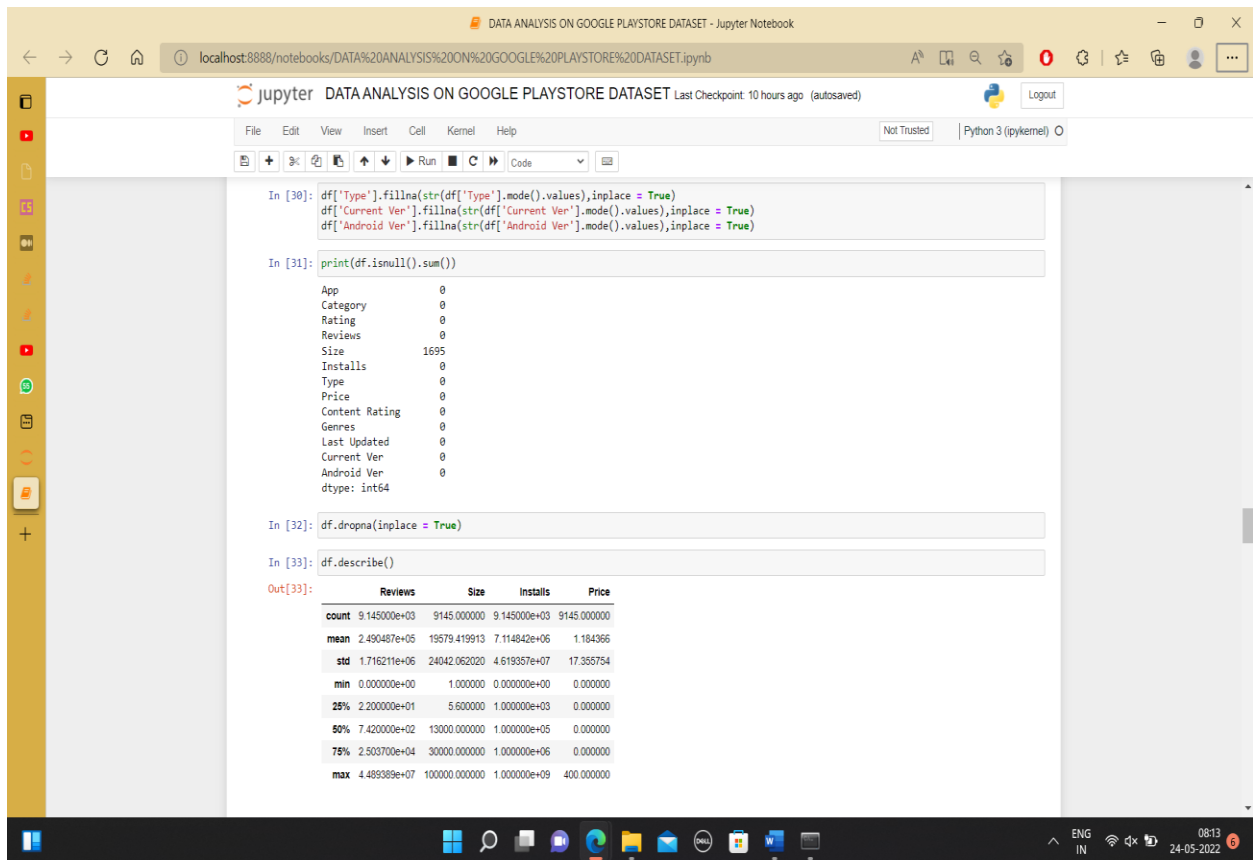
```
In [28]: print(df.isnull().sum())

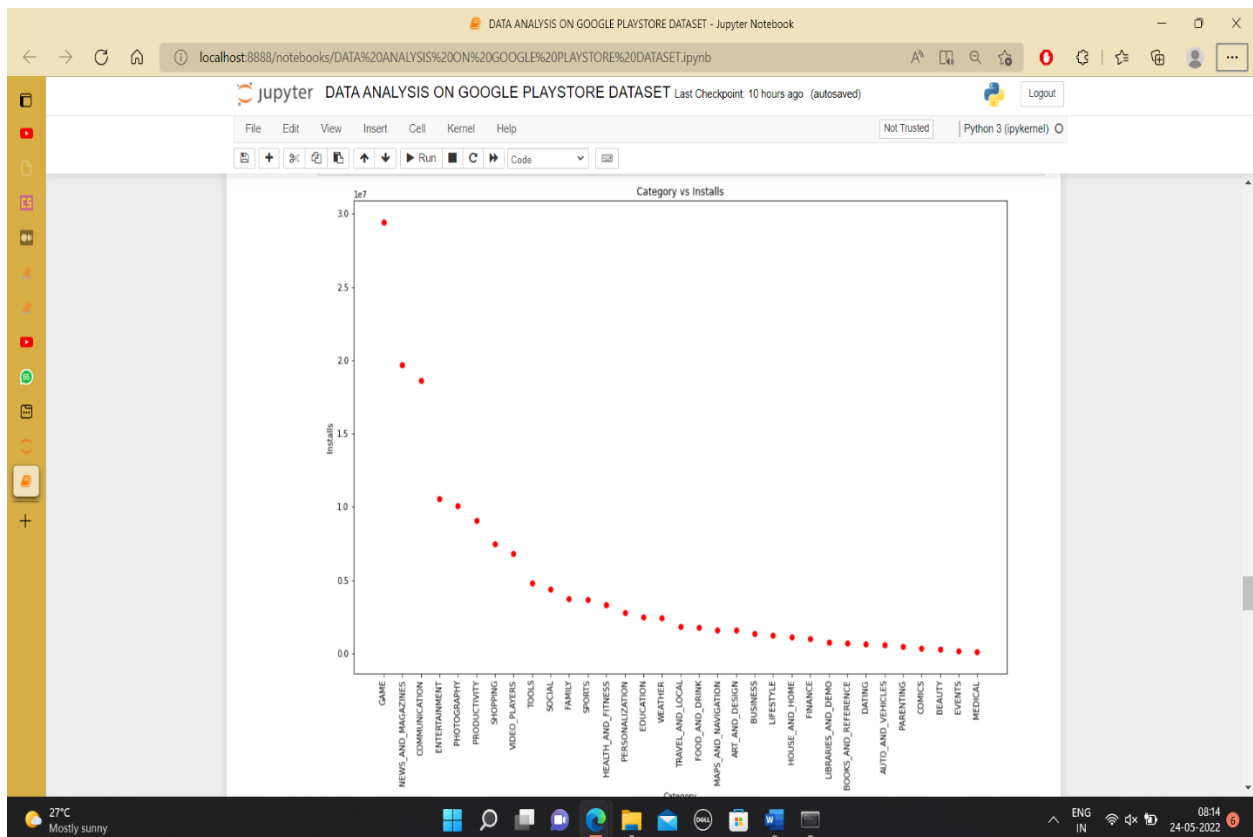
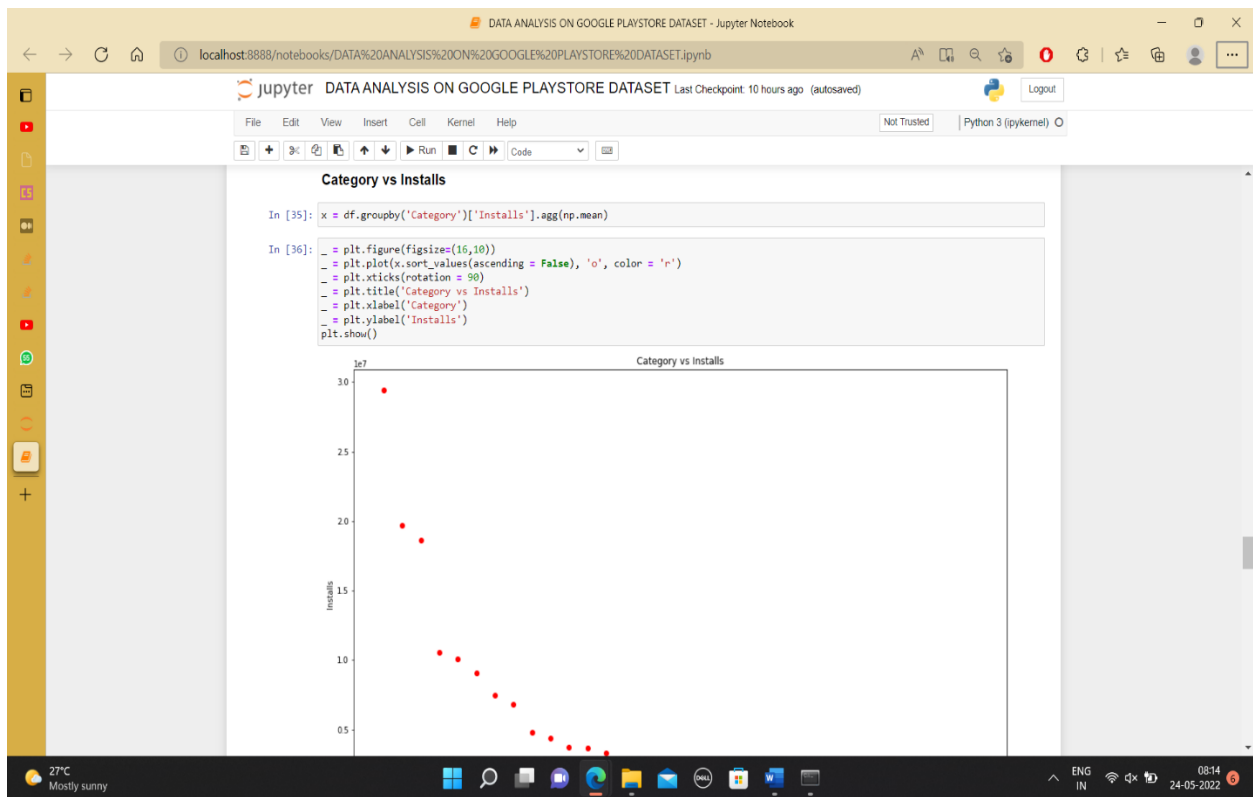
App          0
Category     0
Rating      0
Reviews      0
Size        1695
Installs     0
Type         1
Price        0
Content Rating  0
Genres          0
Last Updated    0
Current Ver     8
Android Ver     2
dtype: int64
```

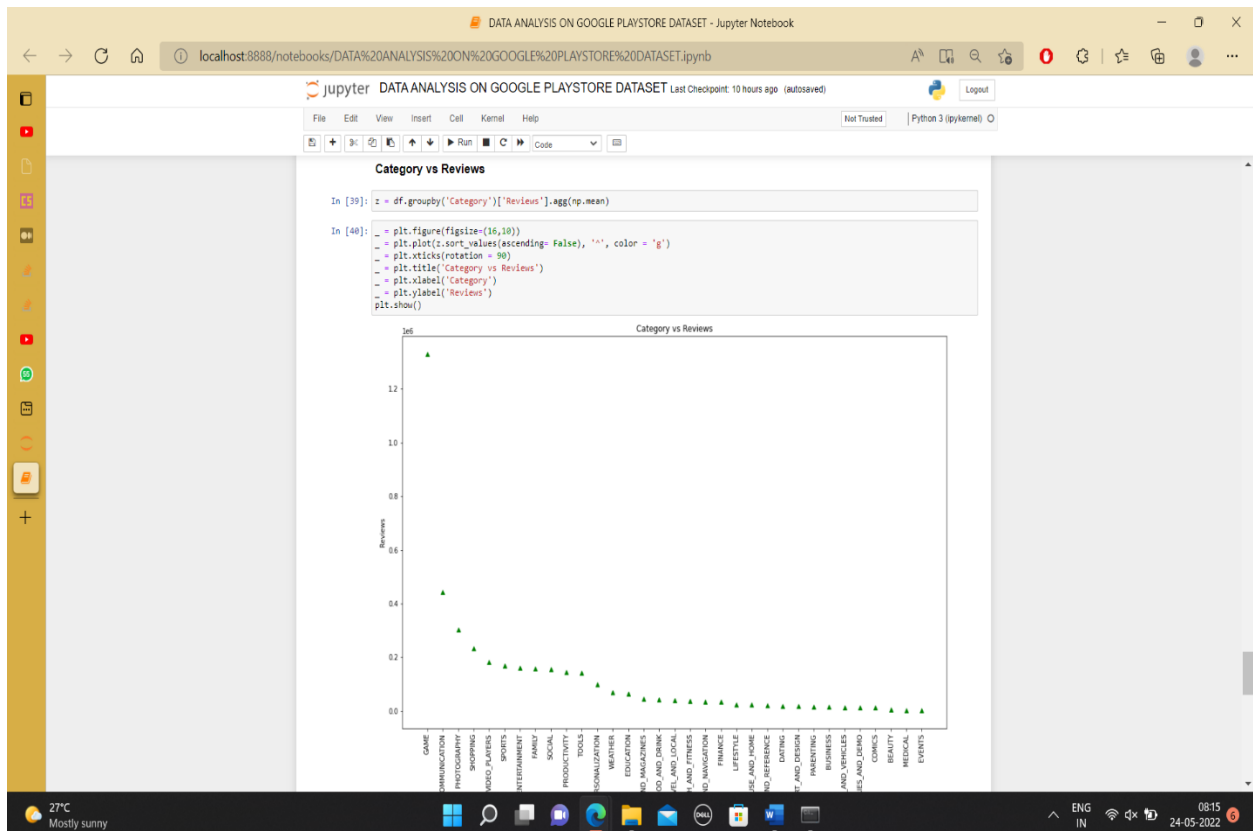
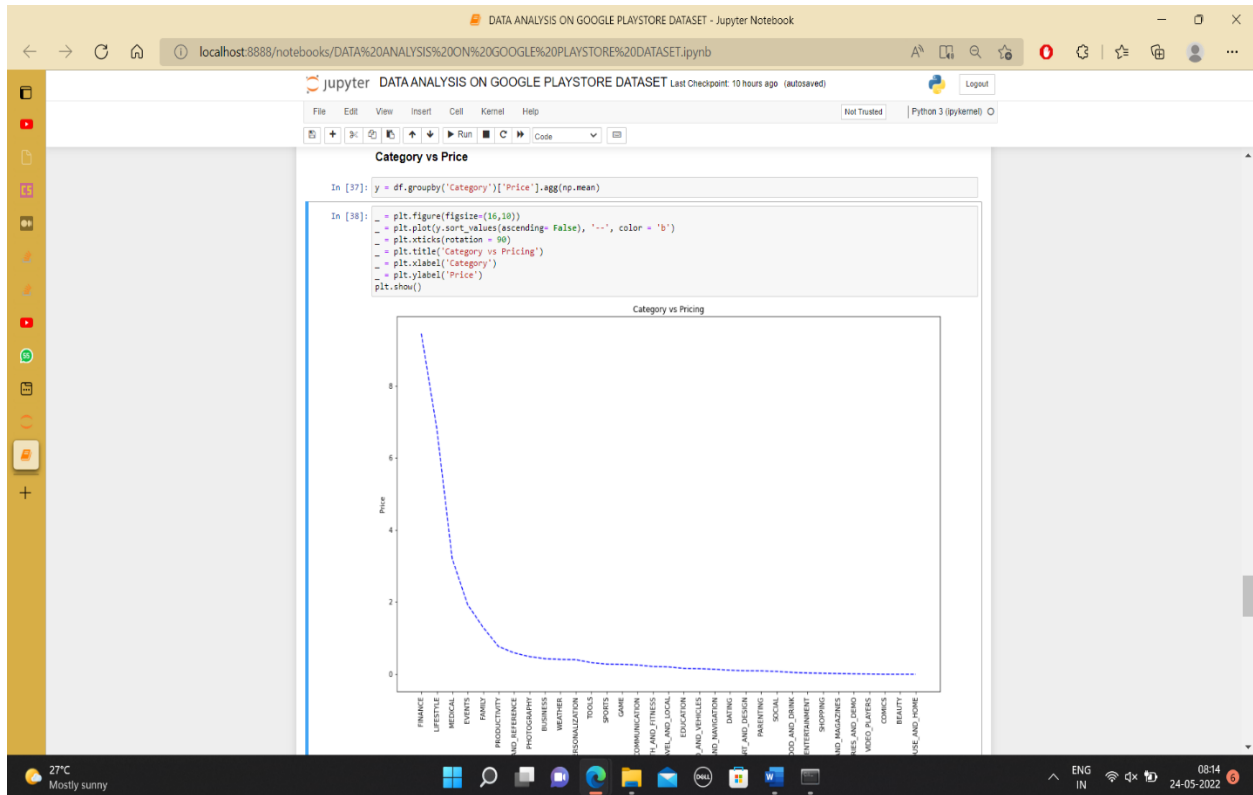
```
In [29]: print(df.Type.mode())
print(df['Current Ver'].mode())
print(df['Android Ver'].mode())

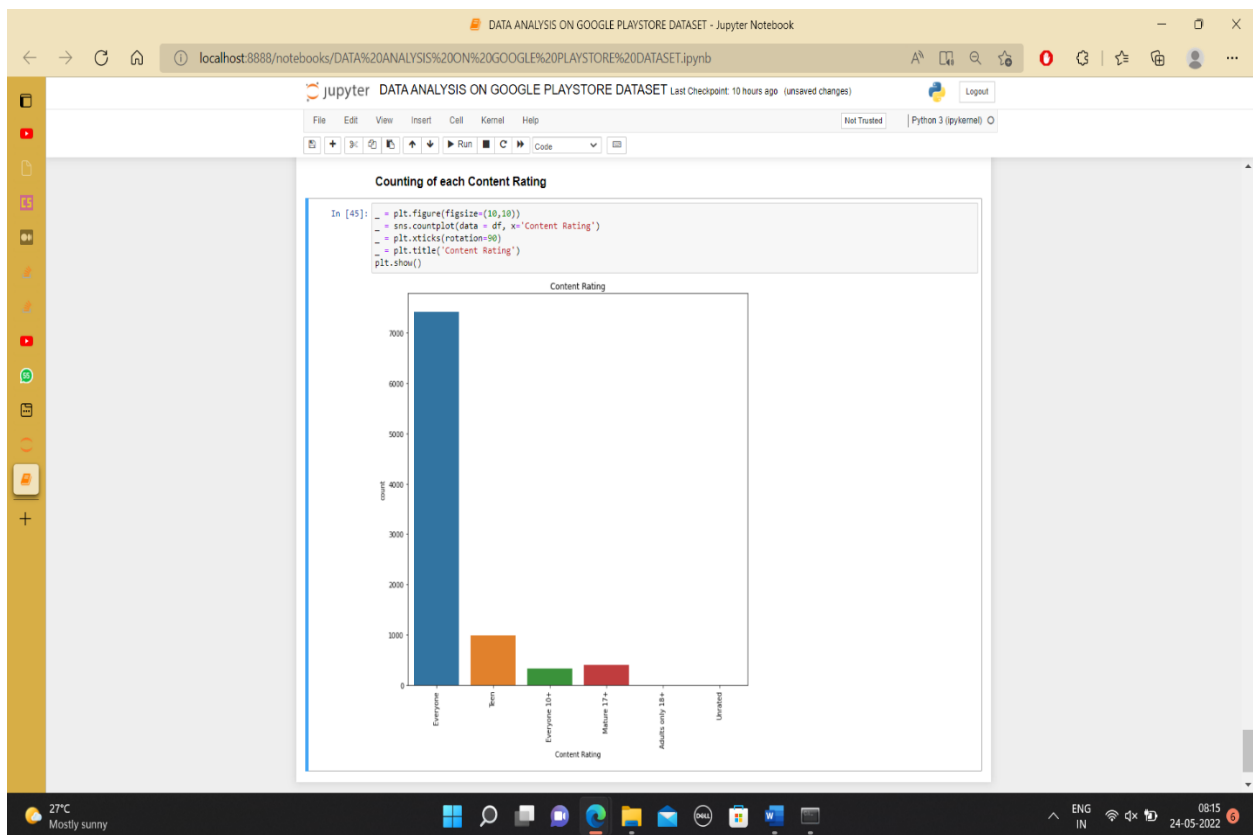
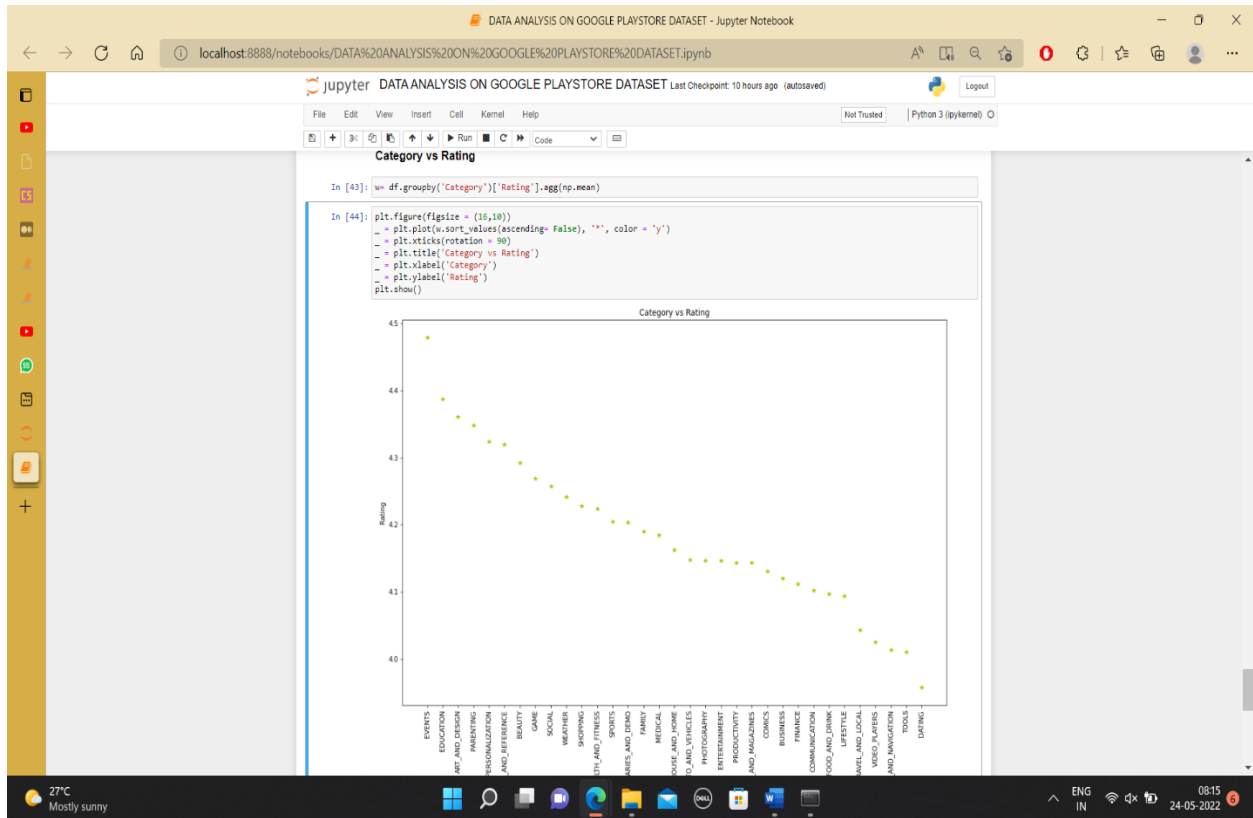
0      Free
Name: Type, dtype: object
0      4.1 and up
Name: Current Ver, dtype: object
0      4.1 and up
Name: Android Ver, dtype: object
```

08:13 24-05-2022









Project Outcome: -

From this project, we learnt to describe a flow process for data science problems and classified data science problems into standard typology. We also learnt about correlating results to the solution approach followed and assessing the solution approach.

Project Conclusion: -

From this project, we gained the knowledge of software – Jupyter Notebook. We learnt to analyse the datasets and afterwards, visualizing them. We learnt about various plots which are – Heatmap, countplot, boxplot, histogram.