Course End Project Preparation

Subject – Programming Basics & Data Analytics with Python

Project Link - [Course End Project - Jupyter Notebook](http://localhost:8888/notebooks/Course%20End%20Project.ipynb)

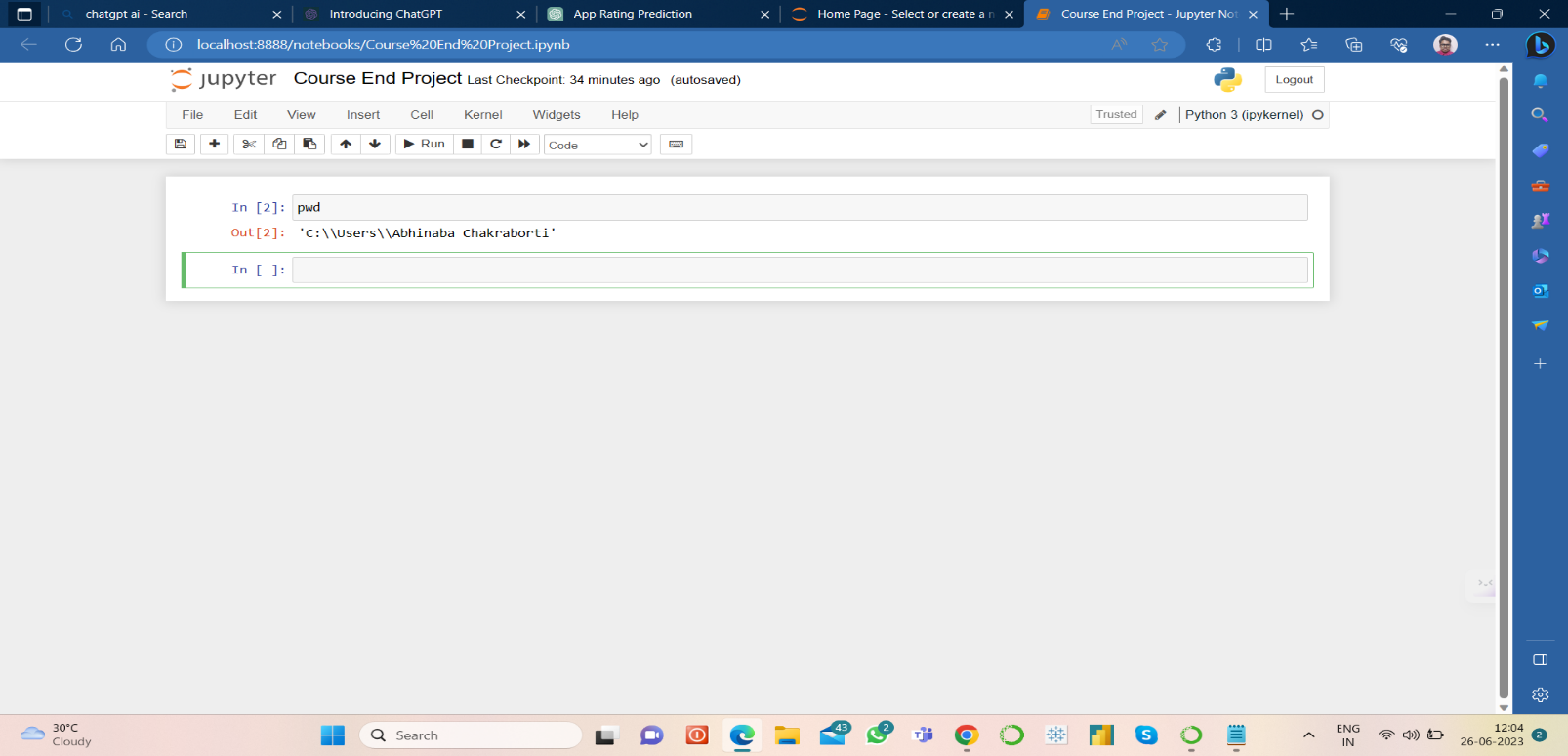
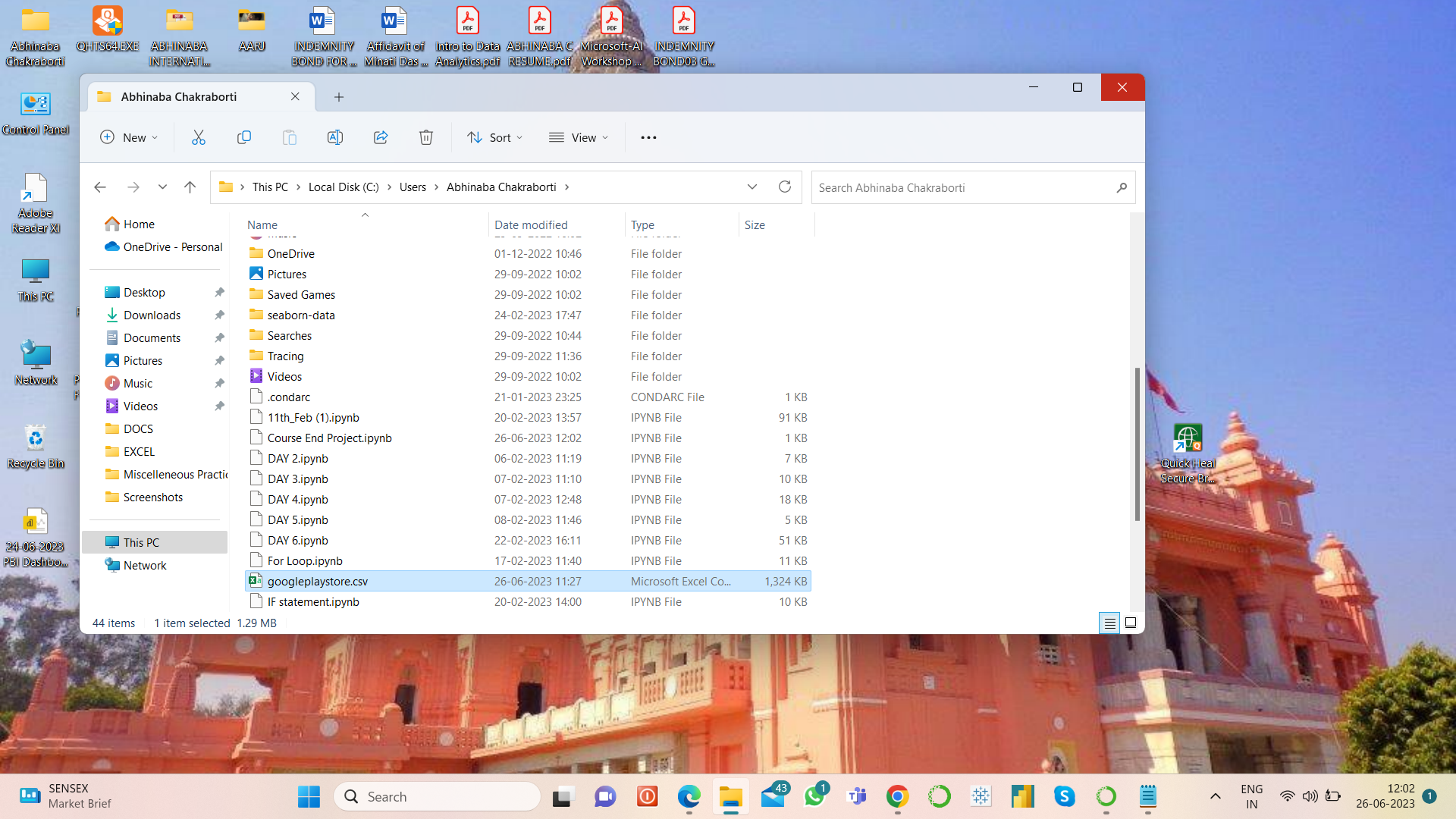


**Name** – Abhinaba Chakraborti

**Mobile** - +91 9007825314

**E-mail ID** – [achakraborti6dollar@gmail.com](mailto:achakraborti6dollar@gmail.com)

STEPS:

1. Download the Dataset from the Portal “googleplaystore.csv”.
2. Open the Jupiter Notebook from Anaconda package and open a new Python 3 ipykernel.
3. Run the code pwd to know the location where the read only .csv excel file is required to be saved.
4. Paste the downloaded .CSV EXCEL file in the location that has been printed.
5. Now, using Pandas, we have to load the data file. For that run the following code:

import pandas as pd

data = pd.read\_csv("googleplaystore.csv")

1. Now, we have to check for the null values available in the data and also find out the number to null values available in each column. For this we have to run the following code:

null\_counts = data.isnull().sum()

1. Now for dropping the records with nulls from any of the columns, we have to run the following code:

data.dropna(inplace=True)

1. Then for Data cleaning and formatting, we have to run the following code:

# Convert Size column to numeric

data['Size'] = data['Size'].apply(lambda x: float(x[:-1]) \* 1000 if 'M' in x else (float(x[:-1]) if x[:-1].isdigit() else -1))

# Convert Reviews column to numeric

data['Reviews'] = data['Reviews'].astype(int)

# Convert Installs column to numeric

data['Installs'] = data['Installs'].apply(lambda x: int(x[:-1].replace(',', '')) if '+' in x else int(x.replace(',', '')))

# Convert Price column to numeric

data['Price'] = data['Price'].apply(lambda x: float(x[1:]) if '$' in x else 0)

1. Now, for Sanity check and cleaning the data further, we have to run the following code:

# Drop rows with rating outside the range [1, 5]

data = data[(data['Rating'] >= 1) & (data['Rating'] <= 5)]

# Drop rows where reviews are greater than installs

data = data[data['Reviews'] <= data['Installs']]

# Drop rows with price greater than 0 for free apps

data = data[(data['Type'] == 'Free') | (data['Price'] == 0)]

1. Now we have to carryout a univariate analysis, that is in a single graph plot, there is going to be one variable at a time. For that of course we have to use Matplotlib Library. Hence, we have to run the following code:

import matplotlib.pyplot as plt

# Boxplot for Price

plt.boxplot(data['Price'])

plt.xlabel('Price')

plt.show()

# Boxplot for Reviews

plt.boxplot(data['Reviews'])

plt.xlabel('Reviews')

plt.show()

# Histogram for Rating

plt.hist(data['Rating'], bins=20)

plt.xlabel('Rating')

plt.ylabel('Count')

plt.show()

# Histogram for Size

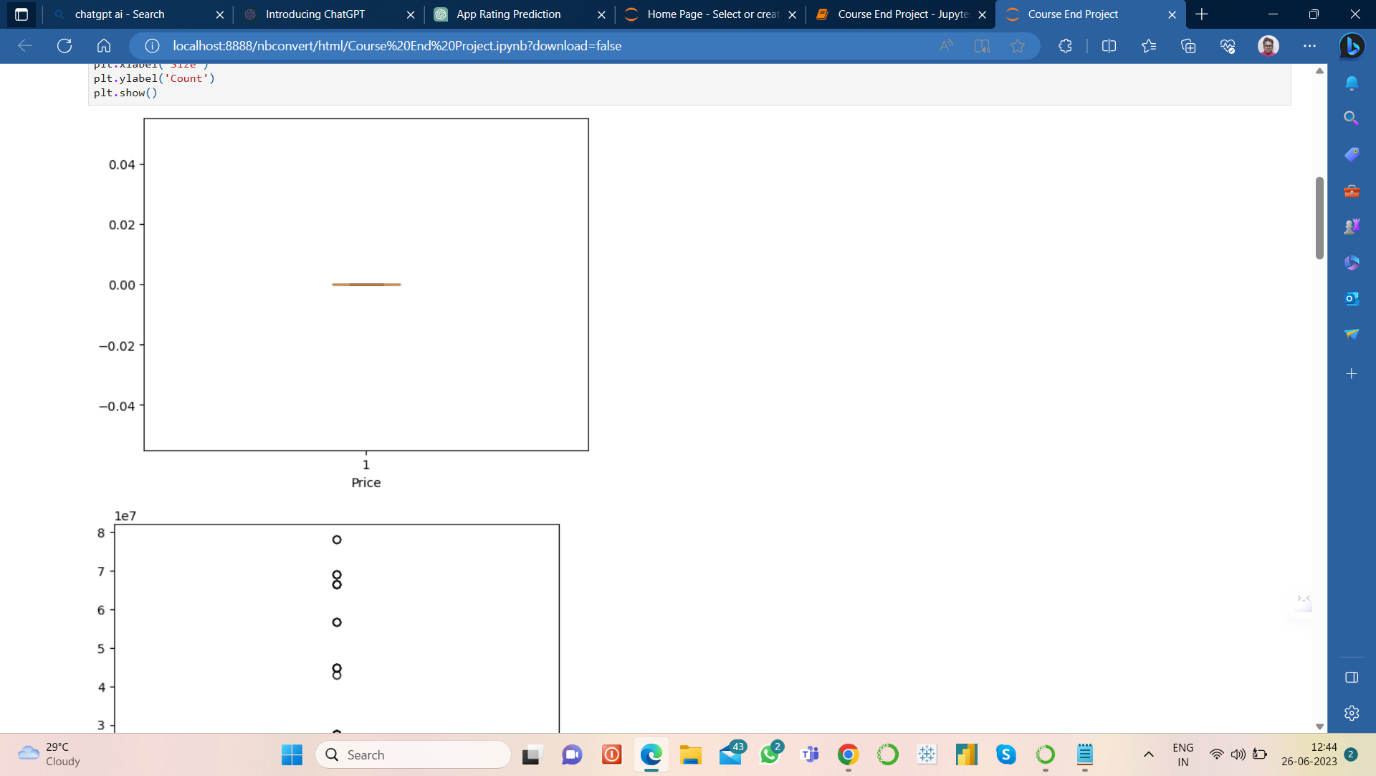
plt.hist(data['Size'], bins=20)

plt.xlabel('Size')

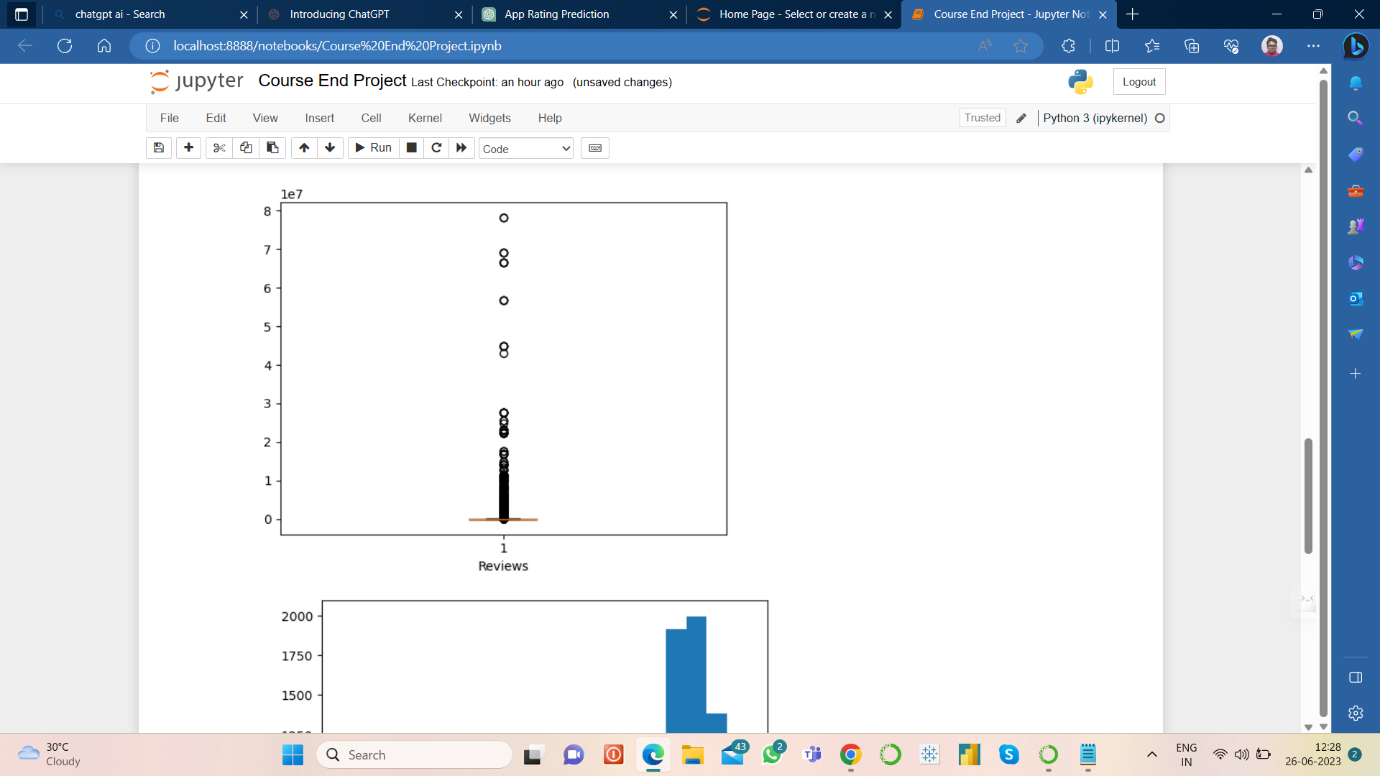
plt.ylabel('Count')

plt.show()

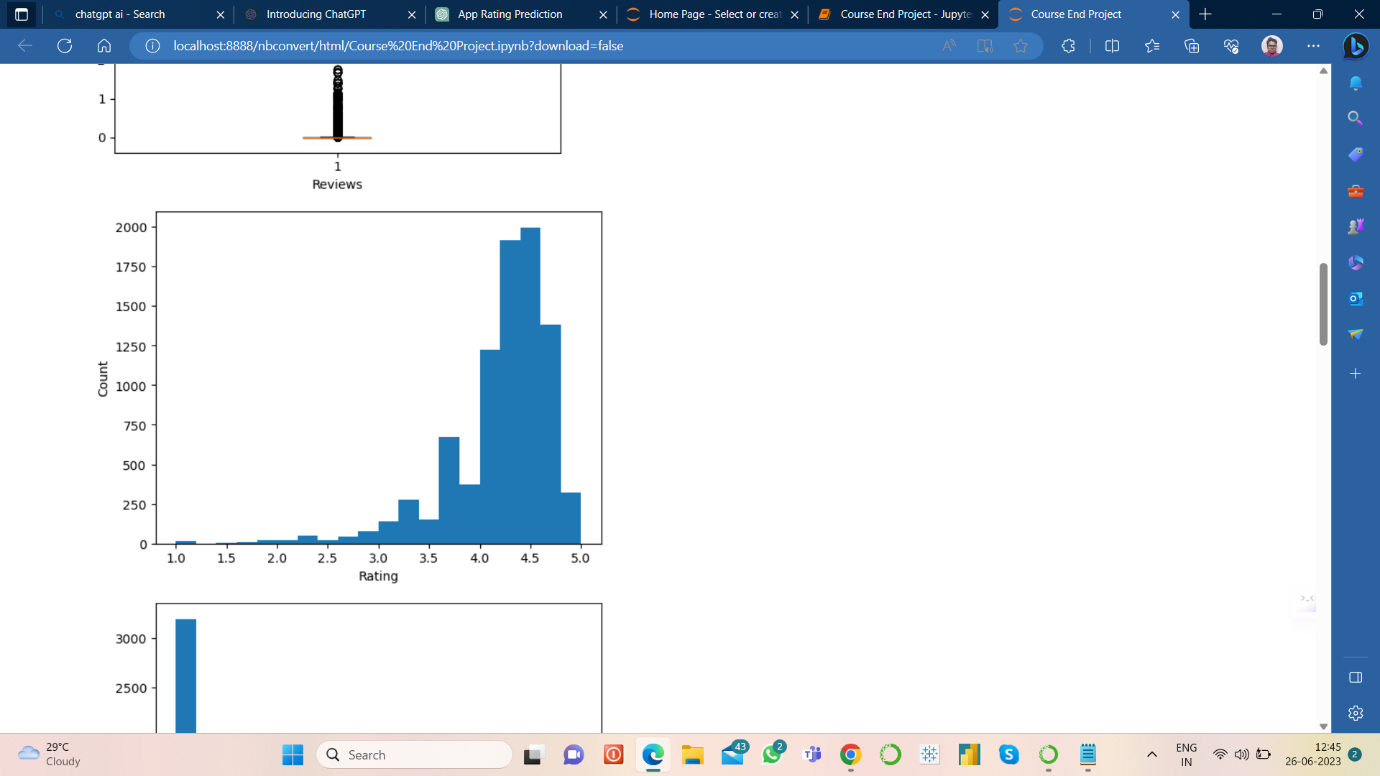
We are going to get the following graph diagrams with required plots:



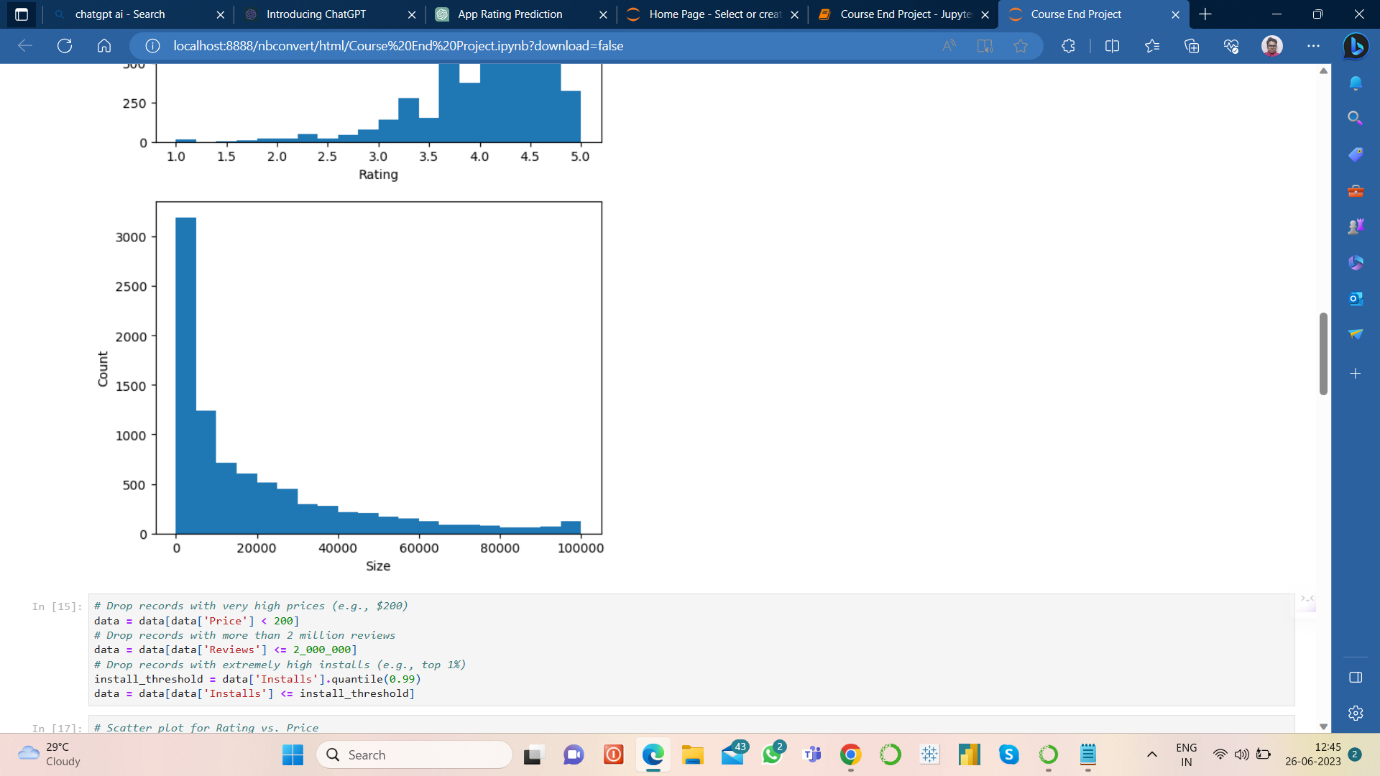
Box Plot for Price



Box Plot for Reviews



Histogram for Rating Count



Histogram for Size Count

1. Then for treating the outliers from ‘Price’, ‘Reviews’ & ‘Installs’, we have to run the following code:

# Drop records with very high prices (e.g., $200)

data = data[data['Price'] < 200]

# Drop records with more than 2 million reviews

data = data[data['Reviews'] <= 2\_000\_000]

# Drop records with extremely high installs (e.g., top 1%)

install\_threshold = data['Installs'].quantile(0.99)

data = data[data['Installs'] <= install\_threshold]

1. After that we have to carry out the bivariate analysis, i.e. the plots will now have two variables. We have the run the following code:

# Scatter plot for Rating vs. Price

plt.scatter(data['Price'], data['Rating'])

plt.xlabel('Price')

plt.ylabel('Rating')

plt.show()

# Scatter plot for Rating vs. Size

plt.scatter(data['Size'], data['Rating'])

plt.xlabel('Size')

plt.ylabel('Rating')

plt.show()

# Scatter plot for Rating vs. Reviews

plt.scatter(data['Reviews'], data['Rating'])

plt.xlabel('Reviews')

plt.ylabel('Rating')

plt.show()

# Boxplot for Rating vs. Content Rating

data.boxplot(column='Rating', by='Content Rating')

plt.xlabel('Content Rating')

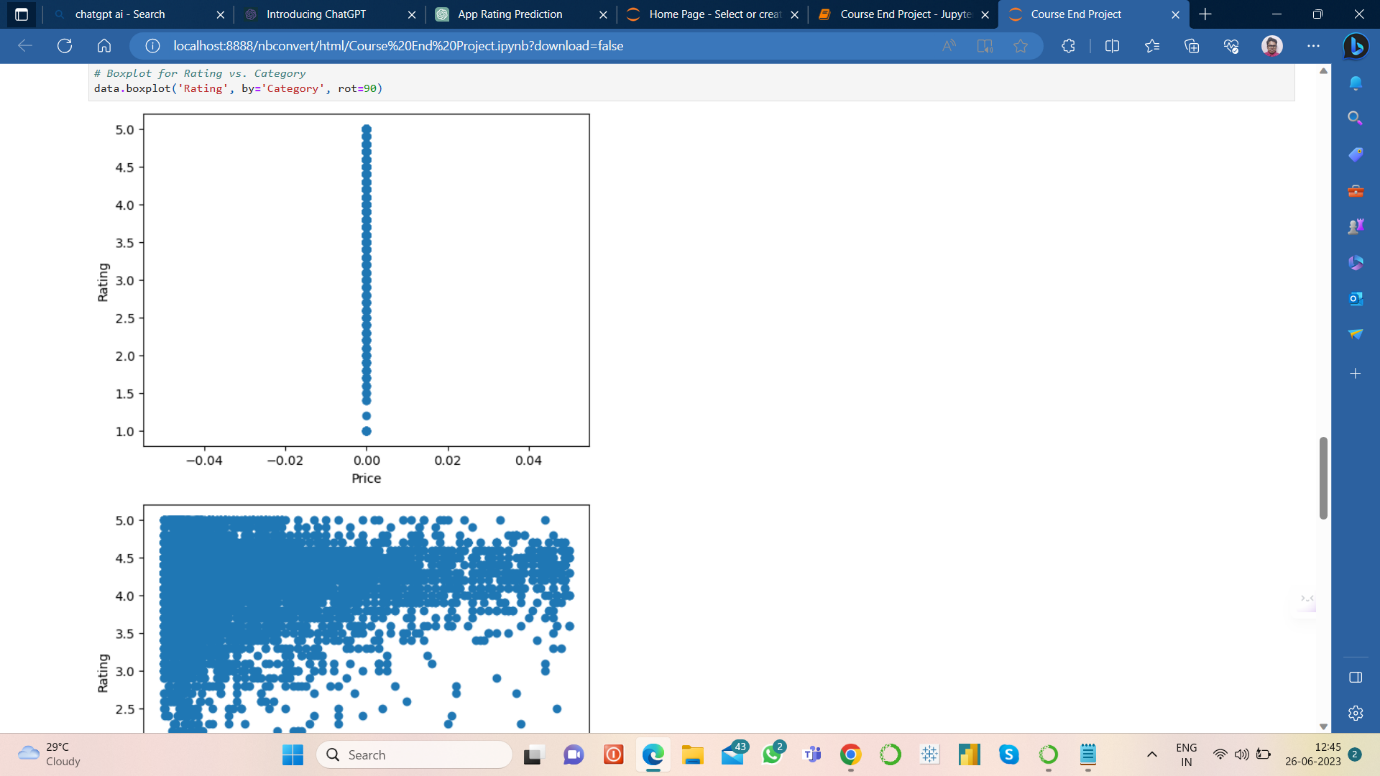
plt.ylabel('Rating')

plt.show()

# Boxplot for Rating vs. Category

data.boxplot('Rating', by='Category', rot=90)

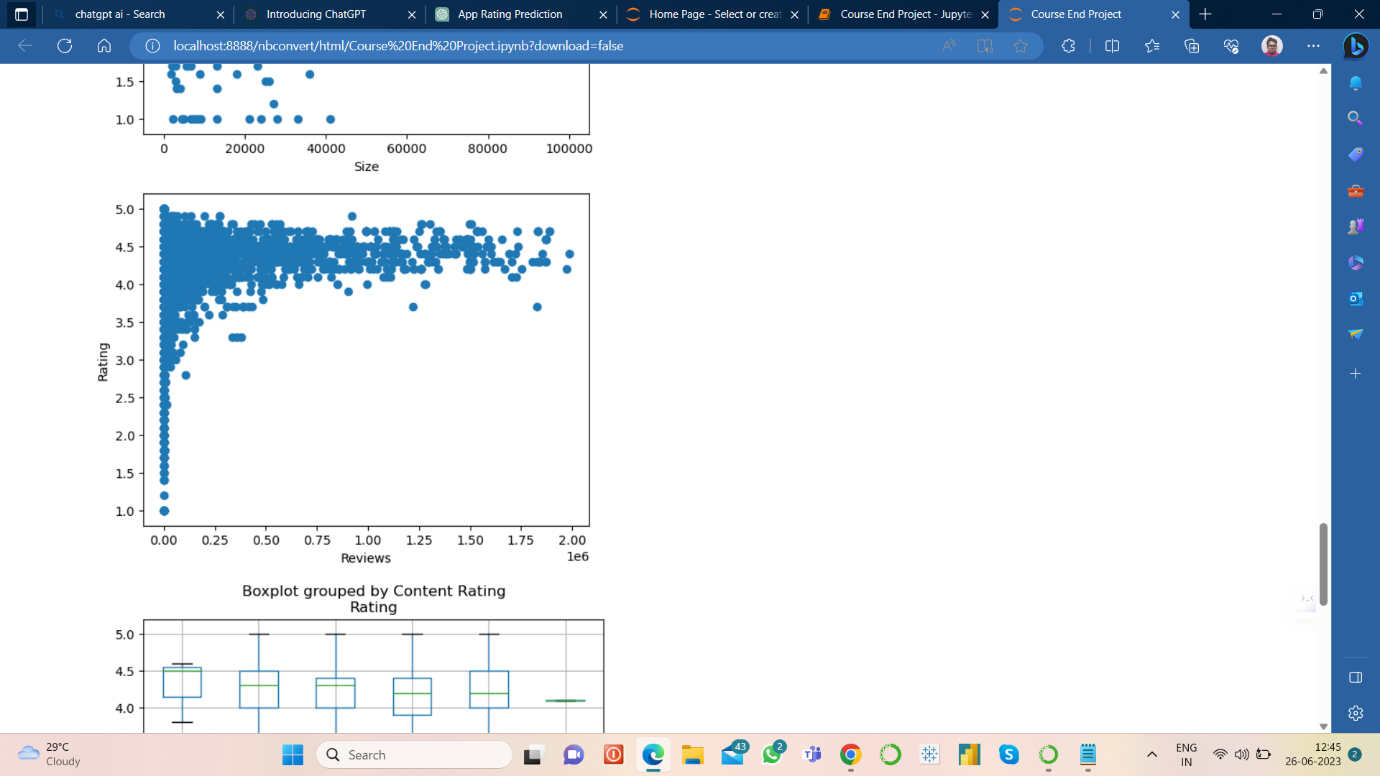
We are going to get the following graph diagrams with required plots:



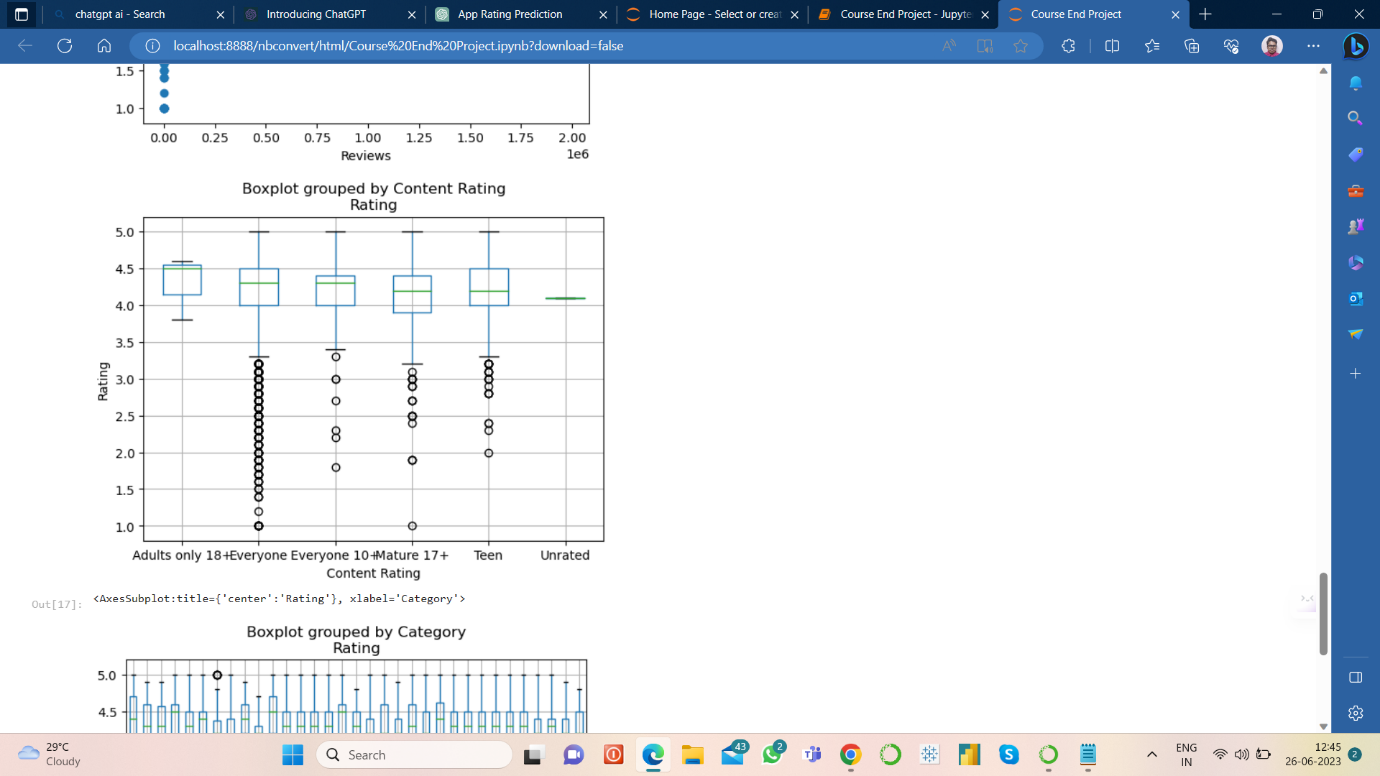
Scatter Plot (Rating Vs Price)



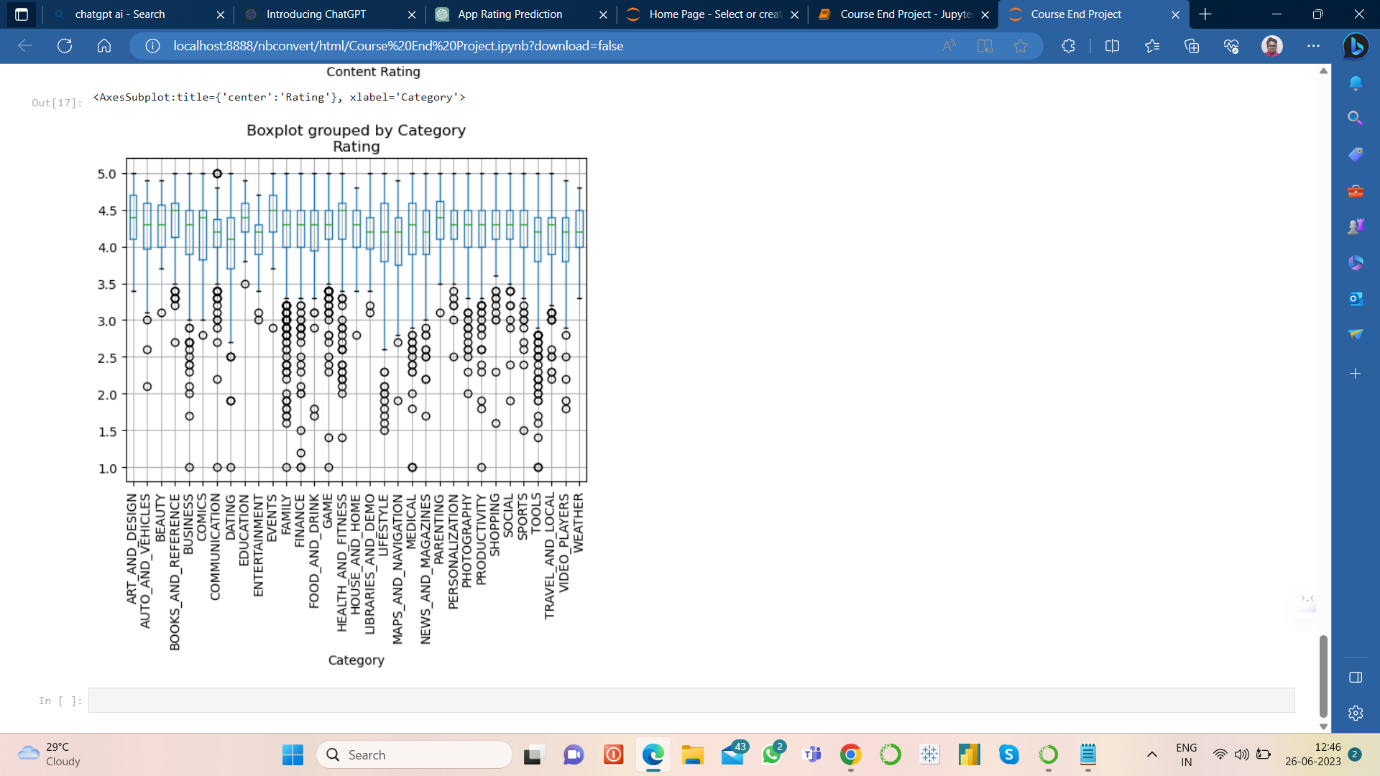
Scatter Plot (Rating Vs Size)



Scatter Plot (Rating Vs Reviews)



Box Plot (Rating Vs Content Rating)



Box Plot (Rating Vs Category)