**INTERNSHIP REPORT**

*A report submitted in partial fulfillment of the requirements for the Award of Degree of*

**BACHELOR OF ENGINEERING**

In

### ELECTRONICS & TELECOMMUNICATION ENGINEERING



**SUBMITTED BY: -**

**Divyanshi Ramesh Rathore (T190323137)**

**UNDER THE GUIDANCE OF: -**

**Supervision of Mr. Sunil Pansare**

**(Duration:1 Months)**

**DEPARTMENT OF ELECTRONICS & TELECOMMUNICATION ENGINEERING**

Modern Education Society’s Wadia College of Engineering, Pune Approved by AICTE,

Affiliated to SPPU, Pune

**Maharashtra [2023-24]**

**MODERN EDUCATION SOCIETY’S WA D I A COLLEGE OF**

**ENGINEERING, PUNE**

**DEPARTMENT OF ELECTRONICS & TELECOMMUNICATION ENGINEERING**



***CERTIFICATE***

This is to certify that the “**Internship report”** submitted by **Divyanshi Ramesh Rathore(T190323137)** is work done by her and submitted during 2023-24 academic year,

in partial fulfillment of the requirements for the award of the degree of **BACHELOR OF ENGINEERING in ELECTRONICS & TELECOMMUNICATION ENGINEERING,** MESCOE, Pune.

**Prof. S S Pansare**

**Department Internship Coordinator Examiner**

.

**Dr. P P Mane**

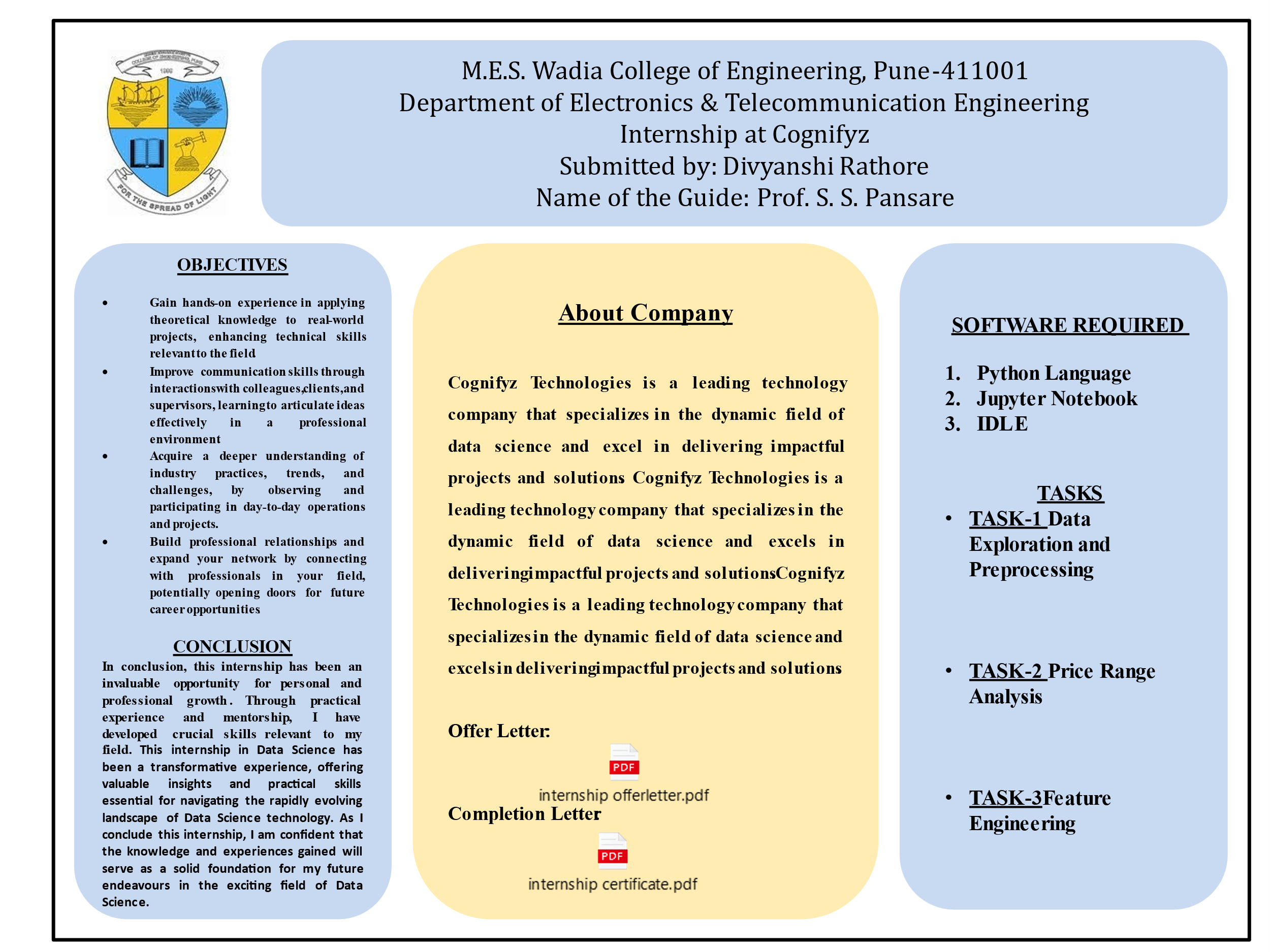
**Head of the Department**

**Date - Place -**

## Certificate Of Completion



**POSTER**



## ACKNOWLEDGEMENT

The internship opportunity I had with Cognifyz was a great chance for learning and professional development. Therefore, I consider myself as a very lucky individual as I was provided with an opportunity to be a part of it. I am also grateful for having a chance to work and explore myself more though this internship period.

I extend my heartfelt appreciation to the entire team at Cognifyz for providing me with the opportunity to intern at their esteemed organization. Special thanks for their guidance, support, and invaluable insights throughout my internship journey.

I offer my sincere phrases of thanks to Prof. S. S. Pansare, Department Internship coordinator for their guidance and constant supervision as well as providing necessary information during Internship work. I express my deepest gratitude to Dr. P. P. Mane, Head of E&TC department for their kind co- operation.

Finally, I would like to express my gratitude towards my parents and all teaching and non-teaching staff members of E&TC department for their kind co-operation and encouragement which help us in completion of this Internship.

**Divyanshi Rathore(T190323137)**

## INDEX

**Sr No. Title Page No.**

1. Learning Objectives/ Internship Objectives 07
2. Weekly Overview of Internship Activities 08
3. Introduction 10
4. Abstract 11
5. Company Information 12
6. Software Requirements 13
7. Technology 14
8. Tasks And Outputs 15

09 Conclusion 41

**Department Of Electronics and Telecommunication Engineering, MESWCOE**

## Learning Objectives/Internship Objectives

* Internships are generally thought of to be reserved for college students looking to gain experience in a particular field. However, a wide array of people can benefit from Training Internships in order to receive real world experience and develop their skills.
* An objective for this position should emphasize the skills you already possess in the area and your interest in learning more.
* Internships are utilized in a number of different career fields, including architecture, engineering, healthcare, economics, advertising and many more.
* Some internship is used to allow individuals to perform scientific research while others are specifically designed to allow people to gain first-hand experience working.
* Utilizing internships is a great way to build your resume and develop skills that can be emphasized in your resume for future jobs. When you are applying for a Training Internship, make sure to highlight any special skills or talents that can make you stand apart from the rest of the applicants so that you have an improved chance of landing the position.

**Department Of Electronics and Telecommunication Engineering, MESWCOE**

**2ND WEEK**

**WEEK**

|  |  |  |  |
| --- | --- | --- | --- |
| **3RD** | 09-01-24 | Tuesday | TASK 2 coding |
|  | 10-01-24 | Wednesday | TASK 2 coding |
|  | 11-01-24 | Thursday | TASK 2 completion |

### WEEKLY OVERVIEW OF INTERNSHIP ACTIVITIES

**K**

|  |  |  |  |
| --- | --- | --- | --- |
| **WEE** | 25-12-23 | Monday | Python Basics |
| **1ST** | 26-12-23 | Tuesday | Python Basics |
|  | 27-12-23 | Wednesday | Python Basics |
|  | 28-12-23 | Thursday | Python Basics & Data set Analysis |

**DATE**

**DAY**

22-12-23

Friday

**NAME OF THE TOPIC/MODULE**

**COMPLETED**

Python Basics

**DATE**

**DAY**

01-01-24

Monday

**NAME OF THE TOPIC/MODULE**

**COMPLETED**

TASK 1

05-01-24

Friday

|  |  |  |
| --- | --- | --- |
| 02-01-24 | Tuesday | TASK 1 Analysis |
| 03-01-24 | Wednesday | TASK 1 coding |
| 04-01-24 | Thursday | TASK 1 coding |

TASK 1 completion

**DATE**

**DAY**

08-01-24

Monday

**NAME OF THE TOPIC/MODULE**

**COMPLETED**

TASK 2 Analysis

**Department Of Electronics and Telecommunication Engineering, MESWCOE**

**4TH WEEK**

**DATE**

**DAY**

15-01-24

Monday

**NAME OF THE TOPIC/MODULE**

**COMPLETED**

TASK 3 Analysis

19-01-24

Friday

TASK 3 completion

|  |  |  |
| --- | --- | --- |
| 16-01-24 | Tuesday | TASK 3 coding |
| 17-01-24 | Wednesday | TASK 3 coding |
| 18-01-24 | Thursday | TASK 3 coding |

**Department Of Electronics and Telecommunication Engineering, MESWCOE**

## INTRODUCTION

The field of data science stands at the forefront of innovation, leveraging

advanced analytics and machine learning algorithms to extract meaningful

insights from vast volumes of data. As a burgeoning discipline, data science

plays a pivotal role in informing strategic decision-making, driving business

growth, and unlocking new opportunities across various industries.

During the duration of 1 month at Cognifyz, I had the privilege of immersing

myself in the dynamic world of data science. Under the guidance of

seasoned professionals and industry experts, my internship journey was

characterized by hands-on learning, collaborative problem-solving, and the

application of cutting-edge techniques to real-world challenges

The primary objective of my internship at Cognifyz was to gain practical

experience and insights into the application of data science methodologies

within a corporate environment. Specifically, I aimed to: Acquire a deeper

understanding of data science principles, techniques, and tools. Enhance my

technical proficiency in programming languages such as Python, R, or SQL,

and data manipulation libraries like pandas and NumPy.

**Department Of Electronics and Telecommunication Engineering, MESWCOE**

## ABSTRACT

As each and every sector of the market is growing, data is building up day by day, we need to keep the record of the data which can be helpful for the analytics and evaluation. Now we don't have data in gigabyte or terabyte but in zettabyte and petabyte and this data cannot be handled with the day-to-day software such as Excel or MATLAB. Therefore, in this report, we will be dealing with large data sets with the high-level programming language 'Python'. The main goal of this project is to aggregate and analyze the data collected from the different data sources available on the internet. These projects mainly focus on the usage of the python programming language in the field of renewable energy. This language has not only it's an application in the field of just analyzing the data but also for the prediction of the upcoming scenarios in the energy field. The purpose of using this specific language is due to its versatility, vast libraries (Pandas, NumPy, Matplotlib, etc.), speed limitations, and ease of learning. We will be analyzing large energy data sets in this project which cannot be easily analyzed in other tools as compared to python. Python does not have it's a limitation to only data analytics but also in many other fields such as Artificial intelligence, Machine learning, and many more.

**Department Of Electronics and Telecommunication Engineering, MESWCOE**

## ABOUT THE COMPANY

Cognifyz Technologies is a leading technology company that specializes

in the dynamic field of data science and excels in delivering impactful

projects and solutions. Cognifyz Technologies is a leading technology

company that specializes in the dynamic field of data science and excels

in delivering impactful projects and solutions. Cognifyz Technologies is

a leading technology company that specializes in the dynamic field of

data science and excels in delivering impactful projects and solutions.

Cognifyz Technologies is a leading technology company that specializes

in the dynamic field of data science and excels in delivering impactful

projects and solutions.

**Department Of Electronics and Telecommunication Engineering, MESWCOE**

## SOFTWARE REQUIREMENTS SPECIFICATIONS

**1.1 System configurations**

**Software Requirements**:

* Operating system: Jupyter Notebook/ Goggle Colab/ IDLE python
* Coding Language: Python

**Hardware Requirements**:

* Ram: 8GB (min)
* Storage: 1TB/512GB
* Processor: i5(min)

**Department Of Electronics and Telecommunication Engineering, MESWCOE**

## TECHNOLOGY

#### Jupyter Notebook

**Project Jupyter** is a project and community whose goal is to "develop open-source software, open-standards, and services for interactive computing across dozens of programming languages". It was spun off from IPython in 2014 by Fernando Pérez and Brian Granger. Project Jupyter's name is a reference to the three core programming languages supported by Jupyter, which are Julia, Python and R, and also a homage to Galileo's notebooks recording the discovery of the moons of Jupiter. Project Jupyter has developed and supported the interactive computing products Jupyter Notebook, JupyterHub, and JupyterLab. 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:

#### Google Colab

Colab is the commonly used abbreviation of the New York City artists' group **Collaborative Projects** which was formed after a series of open meetings between artists of various disciplines.

1. **IDLE**

IDLE is Python’s Integrated Development and Learning Environment,

which is cross-platform and works mostly the same on Windows, Unix,

and macOS

**Department Of Electronics and Telecommunication Engineering, MESWCOE**

# Tasks and Outputs

**Task 1 : Data Exploration and Preprocessing**

* 1. Determine the percentage of restaurants that offer table booking and

online delivery.

* 1. Compare the average ratings of restaurants with table booking and those

without.

* 1. Analyze the availability of online delivery among restaurants with different

price ranges.

**CODE:**

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

import folium

from sklearn.model\_selection import train\_test\_split

from sklearn.linear\_model import LinearRegression

from sklearn.metrics import mean\_squared\_error, r2\_score

from sklearn.preprocessing import StandardScaler

from sklearn.tree import DecisionTreeRegressor

from sklearn.ensemble import RandomForestRegressor

# Replace 'path\_to\_your\_file.csv' with the actual path to your CSV file

file\_path = r'C:\Users\divya\Desktop\cognifyz\Dataset .csv'

# Read the CSV file into a pandas DataFrame

df = pd.read\_csv (file\_path)

# Display the number of rows and columns

num\_rows, num\_columns = df.shape

print(f"Number of rows: {num\_rows}")

print(f"Number of columns: {num\_columns}")

# Check for missing values

missing\_values = df.isnull().sum()

# Display the number of missing values in each column

print("Missing Values in Each Column:")

print(missing\_values)

# Data type conversion (if necessary)

# For example, if "Aggregate rating" is a string, convert it to a numeric type

df['Aggregate rating'] = pd.to\_numeric(df['Aggregate rating'], errors='coerce')

# Check data types after conversion

print("Data types after conversion:")

print(df.dtypes)

# Analyze the distribution of the target variable ("Aggregate rating")

plt.figure(figsize=(10, 6))

sns.histplot(df['Aggregate rating'], bins=20, kde=True)

plt.title('Distribution of Aggregate Rating')

plt.xlabel('Aggregate Rating')

plt.ylabel('Frequency')

plt.show()

# Identify class imbalances

class\_counts = df['Aggregate rating'].value\_counts()

print("\nClass distribution:")

print(class\_counts)

# Select numerical columns

numerical\_columns = df.select\_dtypes(include=['number'])

# Display the descriptive statistics

statistics = numerical\_columns.describe()

# Extract mean, median, and standard deviation from the statistics DataFrame

mean\_values = statistics.loc['mean']

median\_values = statistics.loc['50%']  # 50% corresponds to the median

std\_dev\_values = statistics.loc['std']

# Display the results

print("Mean values:")

print(mean\_values)

print("\nMedian values:")

print(median\_values)

print("\nStandard Deviation values:")

print(std\_dev\_values)

# Explore distribution of "Country Code"

country\_counts = df['Country Code'].value\_counts()

print("\nDistribution of Country Code:")

print(country\_counts)

# Explore distribution of "City"

city\_counts = df['City'].value\_counts()

print("\nDistribution of City:")

print(city\_counts)

# Explore distribution of "Cuisines"

# Note: Split cuisines strings and count each cuisine separately

cuisines = df['Cuisines'].str.split(',').explode().str.strip()

cuisine\_counts = cuisines.value\_counts()

print("\nDistribution of Cuisines:")

print(cuisine\_counts)

# Identify top cuisines

top\_cuisines = cuisine\_counts.head(10)

print("\nTop Cuisines:")

print(top\_cuisines)

# Identify top cities with the highest number of restaurants

top\_cities = city\_counts.head(10)

print("\nTop Cities with the Highest Number of Restaurants:")

print(top\_cities)

# Visualize the distribution of cities

plt.figure(figsize=(12, 6))

sns.barplot(x=top\_cities.index, y=top\_cities.values)

plt.title('Top Cities with the Highest Number of Restaurants')

plt.xlabel('City')

plt.ylabel('Number of Restaurants')

plt.xticks(rotation=45, ha='right')

plt.show()

# Filter out rows with missing latitude or longitude information

df = df.dropna(subset=['Latitude', 'Longitude'])

# Create a map centered around the first restaurant's location

map\_restaurants = folium.Map(location=[df['Latitude'].iloc[0], df['Longitude'].iloc[0]], zoom\_start=12)

# Add markers for each restaurant

for index, row in df.iterrows():

    folium.Marker([row['Latitude'], row['Longitude']], popup=row['Restaurant Name']).add\_to(map\_restaurants)

# Save the map as an HTML file

map\_restaurants.save('restaurant\_map.html')

# Explore distribution of restaurants across cities

city\_counts = df['City'].value\_counts()

# Visualize distribution of restaurants across cities

plt.figure(figsize=(12, 6))

sns.barplot(x=city\_counts.index, y=city\_counts.values)

plt.title('Distribution of Restaurants Across Cities')

plt.xlabel('City')

plt.ylabel('Number of Restaurants')

plt.xticks(rotation=45, ha='right')

plt.show()

# Determine correlation between location and rating

correlation\_matrix = df[['Latitude', 'Longitude', 'Aggregate rating']].corr()

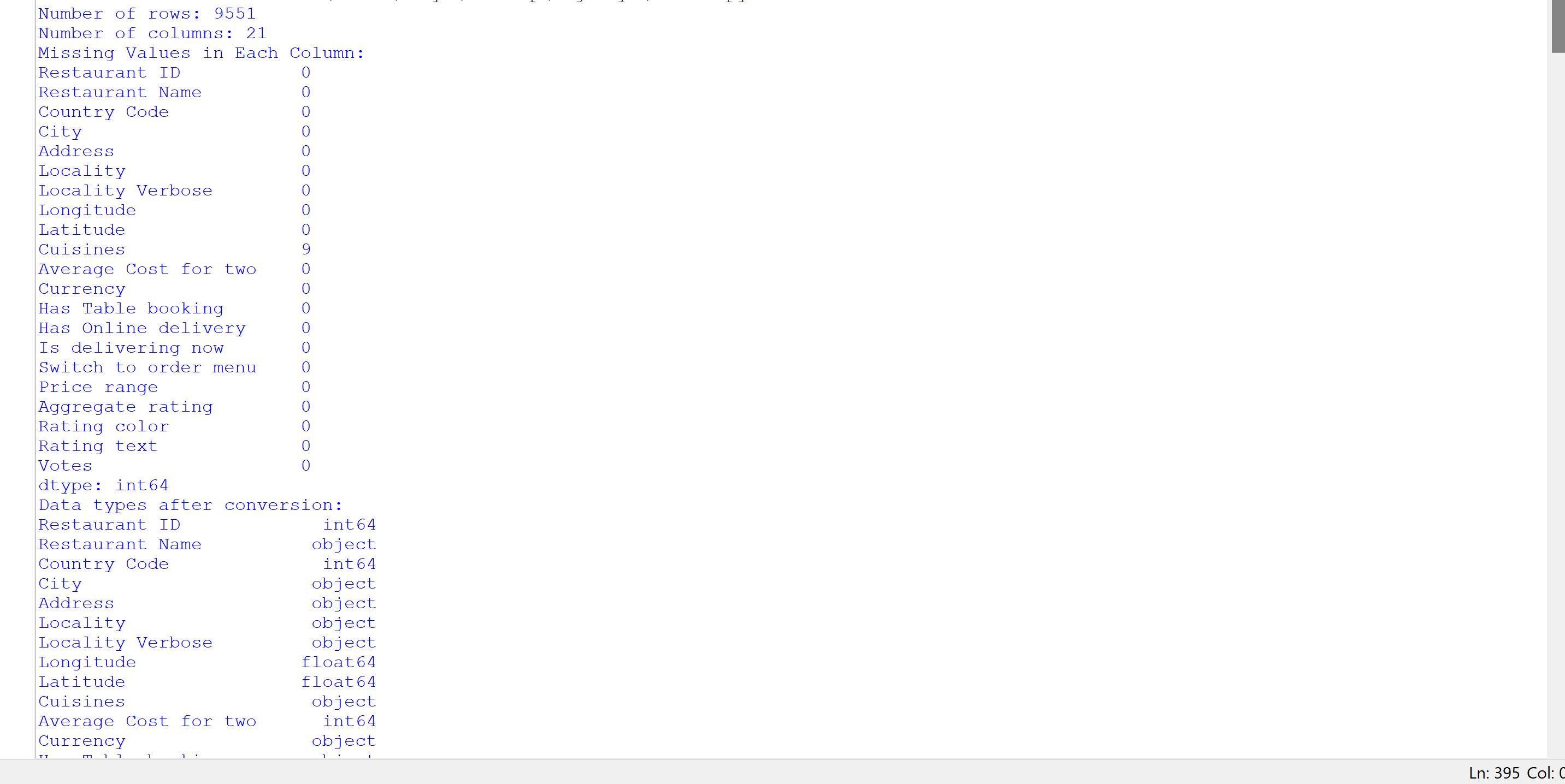
# Visualize the correlation matrix using a heatmap

plt.figure(figsize=(8, 6))

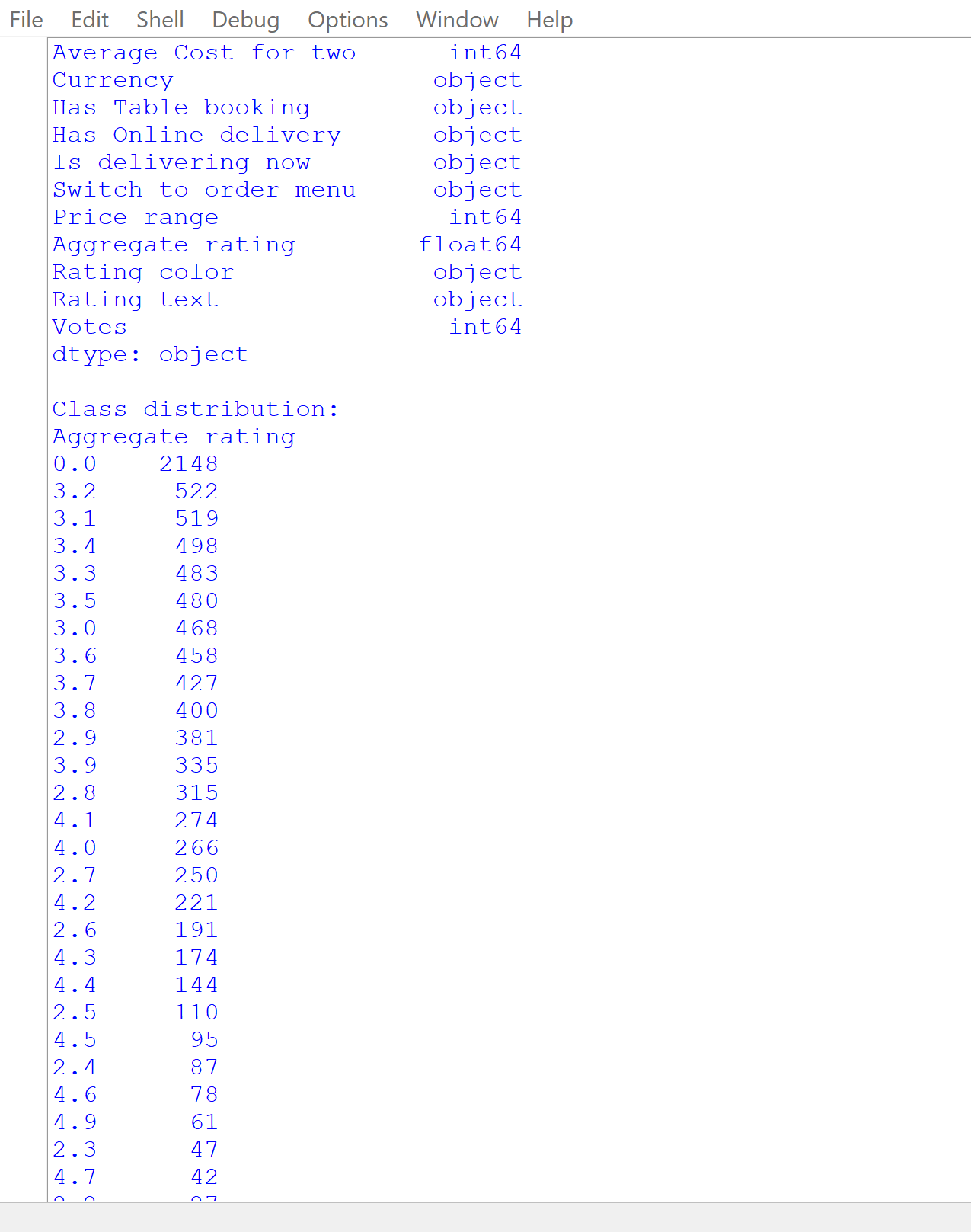
sns.heatmap(correlation\_matrix, annot=True, cmap='coolwarm', linewidths=.5)

plt.title('Correlation Matrix: Location vs. Rating')

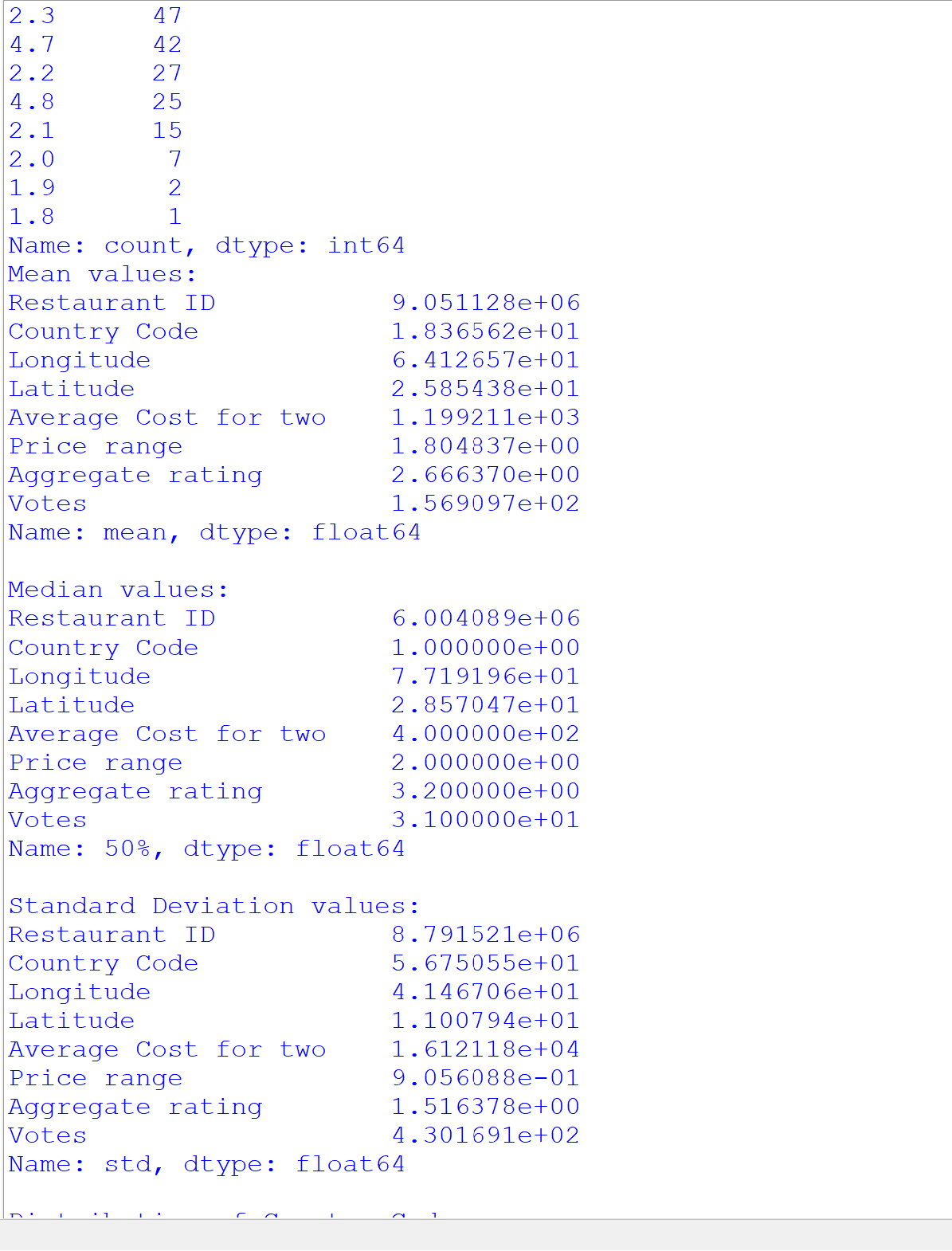
**OUTPUT:**



**Department Of Electronics and Telecommunication Engineering, MESWCOE**



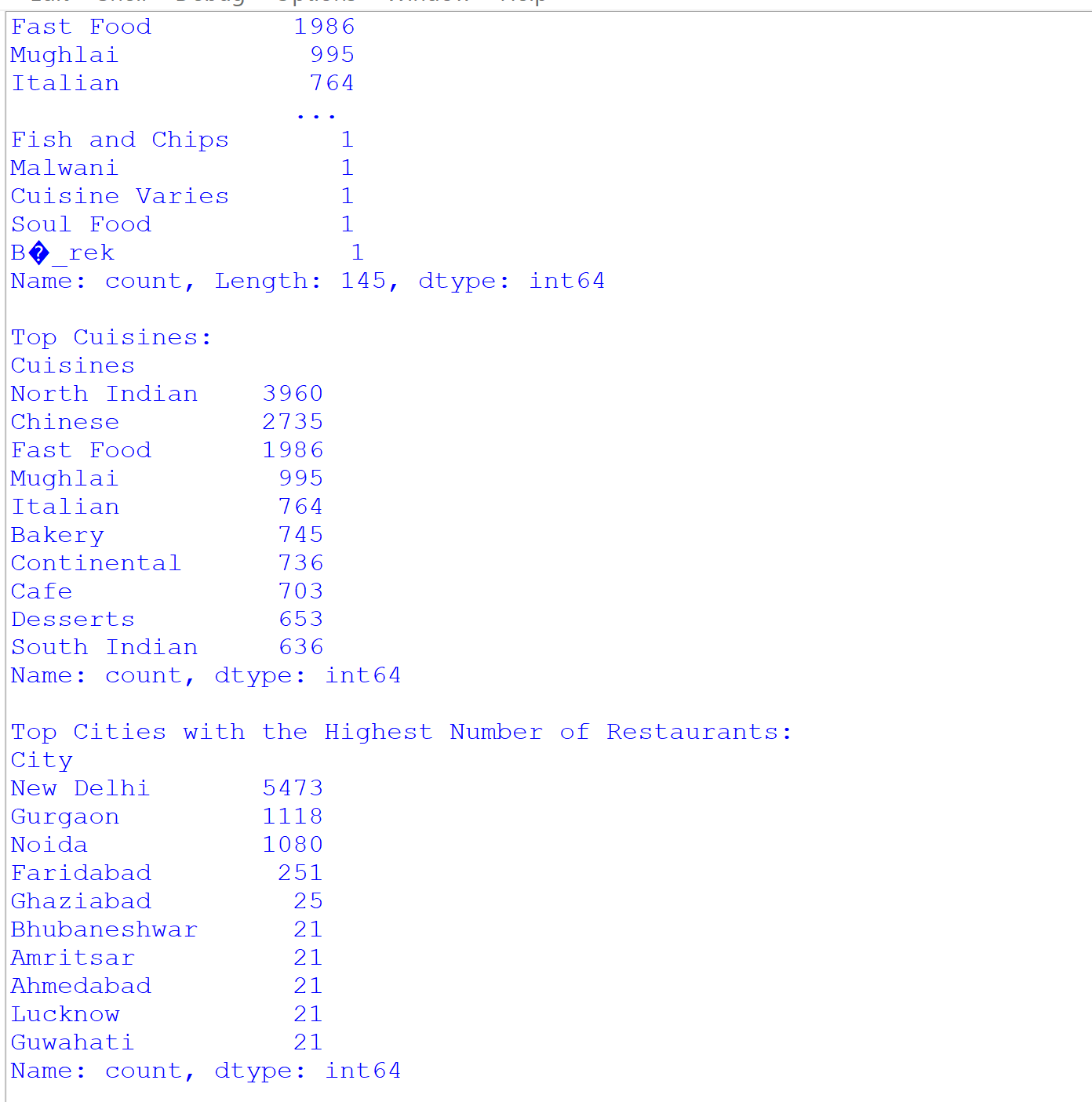
**Department Of Electronics and Telecommunication Engineering, MESWCOE**



**Department Of Electronics and Telecommunication Engineering, MESWCOE**

# 

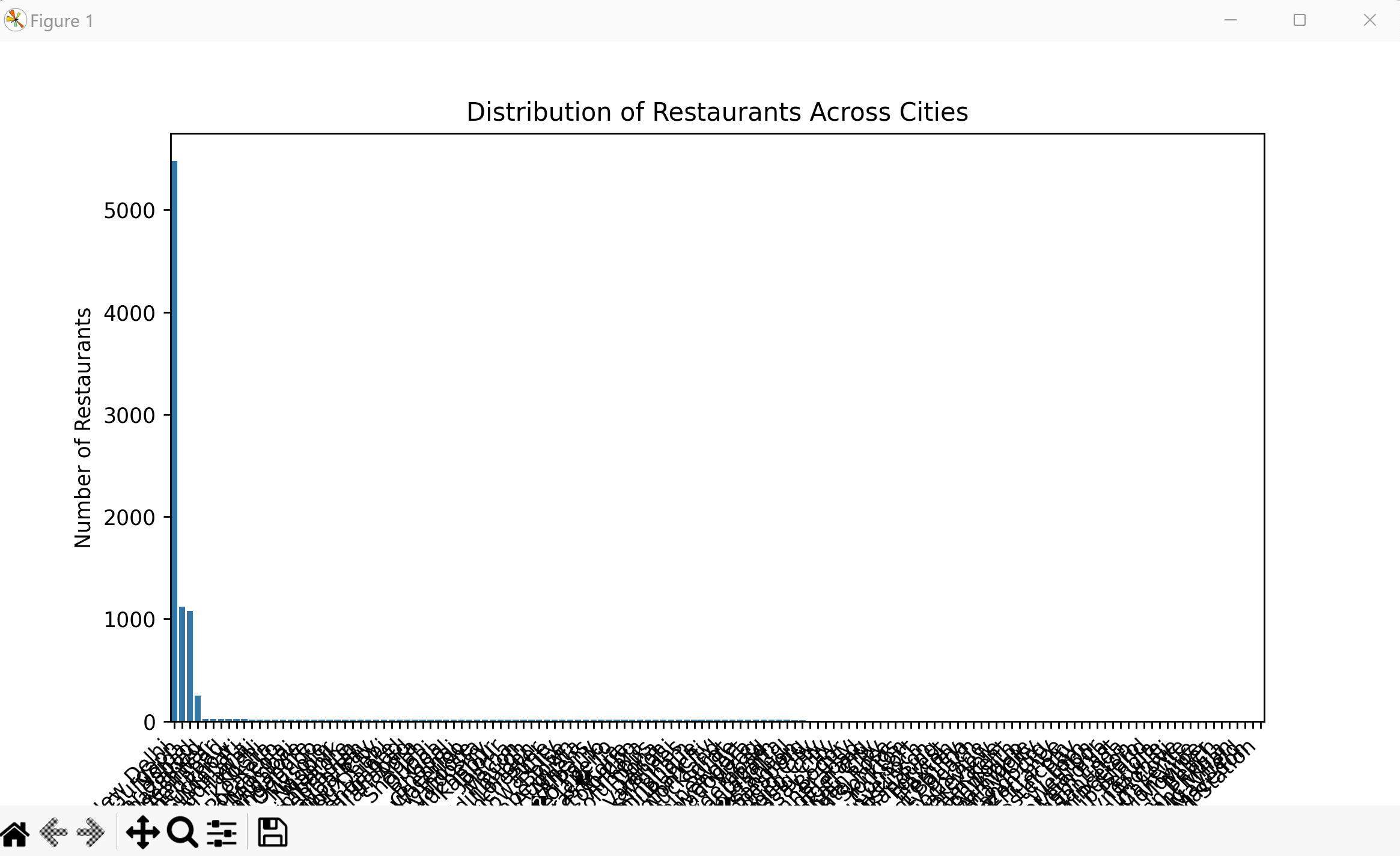
**Department Of Electronics and Telecommunication Engineering, MESWCOE**



**Department Of Electronics and Telecommunication Engineering, MESWCOE**







**TASK 2:** **Price Range Analysis**

* 1. Determine the most common price range among all the restaurants.
  2. Calculate the average rating for each price range.
  3. Identify the color that represents the highest average rating among different

price ranges.

**CODE:**

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

import folium

from sklearn.model\_selection import train\_test\_split

from sklearn.linear\_model import LinearRegression

from sklearn.metrics import mean\_squared\_error, r2\_score

from sklearn.preprocessing import StandardScaler

from sklearn.tree import DecisionTreeRegressor

from sklearn.ensemble import RandomForestRegressor

# Replace 'path\_to\_your\_file.csv' with the actual path to your CSV file

file\_path = r'C:\Users\divya\Desktop\cognifyz\Dataset .csv'

# Read the CSV file into a pandas DataFrame

df = pd.read\_csv (file\_path)

#Task2

# Determine the percentage of restaurants offering table booking and online delivery

total\_restaurants = len(df)

# Count the number of restaurants offering table booking and online delivery

restaurants\_with\_table\_booking = df['Has Table booking'].value\_counts()['Yes']

restaurants\_with\_online\_delivery = df['Has Online delivery'].value\_counts()['Yes']

# Calculate the percentage

table\_booking\_percentage = (restaurants\_with\_table\_booking / total\_restaurants) \* 100

online\_delivery\_percentage = (restaurants\_with\_online\_delivery / total\_restaurants) \* 100

# Display the results

print(f"\nPercentage of Restaurants Offering Table Booking: {table\_booking\_percentage:.2f}%")

print(f"Percentage of Restaurants Offering Online Delivery: {online\_delivery\_percentage:.2f}%")

# Convert 'Aggregate rating' column to numeric

df['Aggregate rating'] = pd.to\_numeric(df['Aggregate rating'], errors='coerce')

# Compare the average ratings

average\_rating\_with\_table\_booking = df[df['Has Table booking'] == 'Yes']['Aggregate rating'].mean()

average\_rating\_without\_table\_booking = df[df['Has Table booking'] == 'No']['Aggregate rating'].mean()

# Display the results

print(f"Average Rating for Restaurants with Table Booking: {average\_rating\_with\_table\_booking:.2f}")

print(f"Average Rating for Restaurants without Table Booking: {average\_rating\_without\_table\_booking:.2f}")

# Explore the unique values in the 'Price range' and 'Has Online delivery' columns

unique\_price\_ranges = df['Price range'].unique()

unique\_online\_delivery = df['Has Online delivery'].unique()

# Display the unique values

print("Unique Price Ranges:", unique\_price\_ranges)

print("Unique Online Delivery Options:", unique\_online\_delivery)

# Analyze the availability of online delivery among restaurants with different price ranges

delivery\_by\_price\_range = df.groupby('Price range')['Has Online delivery'].value\_counts(normalize=True).unstack()

# Display the results

print("\nAvailability of Online Delivery Among Restaurants with Different Price Ranges:")

print(delivery\_by\_price\_range)

# Determine the most common price range

most\_common\_price\_range = df['Price range'].mode().iloc[0]

# Display the result

print("Most Common Price Range: ", most\_common\_price\_range)

# Convert 'Aggregate rating' and 'Price range' columns to numeric

df['Aggregate rating'] = pd.to\_numeric(df['Aggregate rating'], errors='coerce')

df['Price range'] = pd.to\_numeric(df['Price range'], errors='coerce')

# Calculate the average rating for each price range

average\_rating\_by\_price\_range = df.groupby('Price range')['Aggregate rating'].mean()

# Identify the color that represents the highest average rating

highest\_avg\_rating\_color = 'green'  # Default color

if not average\_rating\_by\_price\_range.empty:

    highest\_avg\_rating\_price\_range = average\_rating\_by\_price\_range.idxmax()

    if pd.notnull(highest\_avg\_rating\_price\_range):

        highest\_avg\_rating\_color = 'red'  # Change color to red for the highest average rating

# Display the average rating for each price range

print("Average Rating for Each Price Range:")

print(average\_rating\_by\_price\_range)

# Plot a bar chart with colors

plt.bar(average\_rating\_by\_price\_range.index, average\_rating\_by\_price\_range, color=[highest\_avg\_rating\_color if x == highest\_avg\_rating\_price\_range else 'lightblue' for x in average\_rating\_by\_price\_range.index])

plt.xlabel('Price Range')

plt.ylabel('Average Rating')

plt.title('Average Rating for Each Price Range')

plt.show()

# Extract additional features

df['Restaurant Name Length'] = df['Restaurant Name'].apply(len)

df['Address Length'] = df['Address'].apply(len)

# Display the updated DataFrame with additional features

print("Updated DataFrame with Additional Features:")

print(df.head())

# Save the updated DataFrame to a new CSV file

df.to\_csv('updated\_dataset.csv', index=False)

# Perform one-hot encoding for 'Has Table Booking' and 'Has Online Delivery'

df\_encoded = pd.get\_dummies(df, columns=['Has Table booking', 'Has Online delivery'], drop\_first=True)

# Display the updated DataFrame with new features

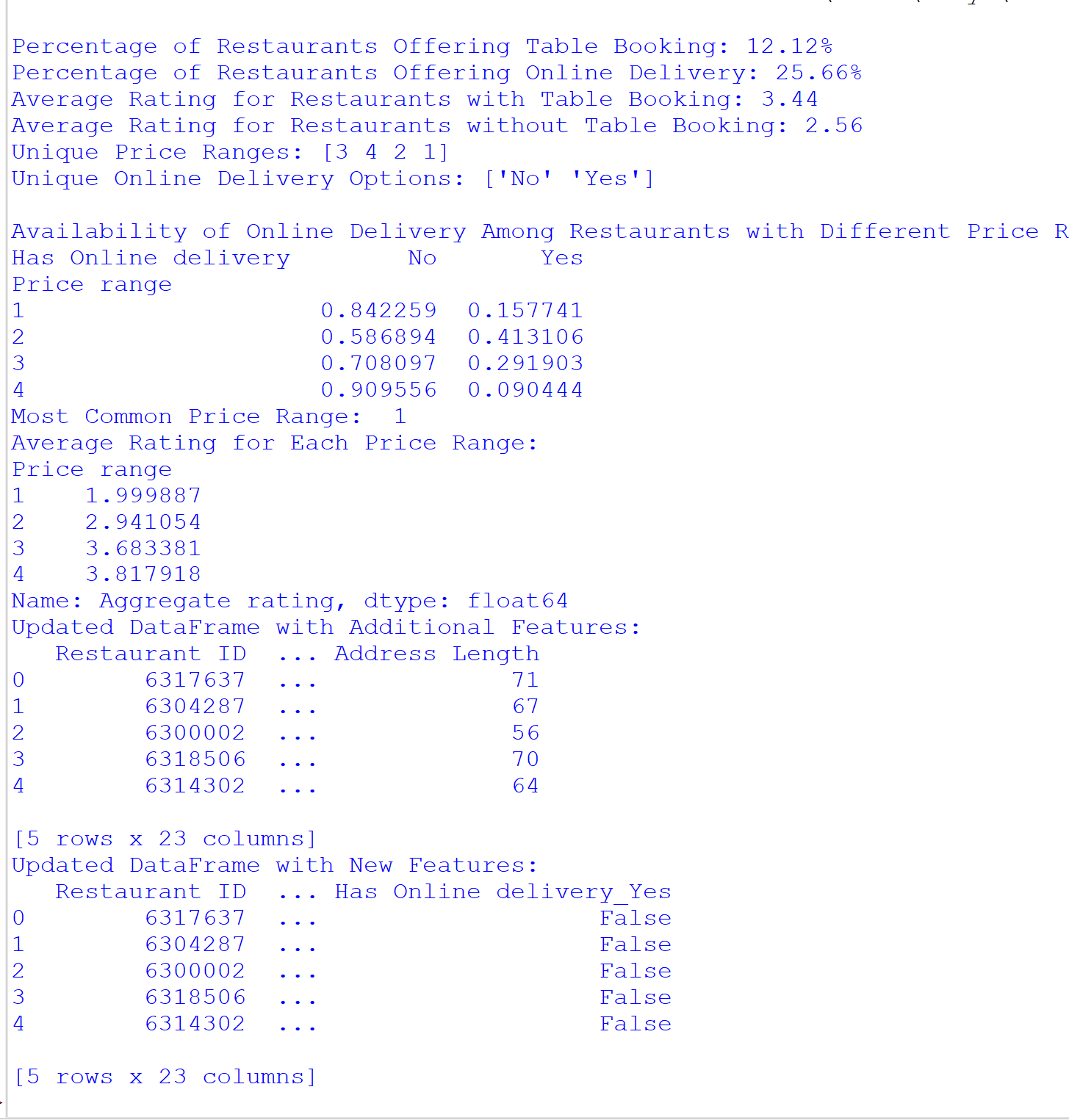
print("Updated DataFrame with New Features:")

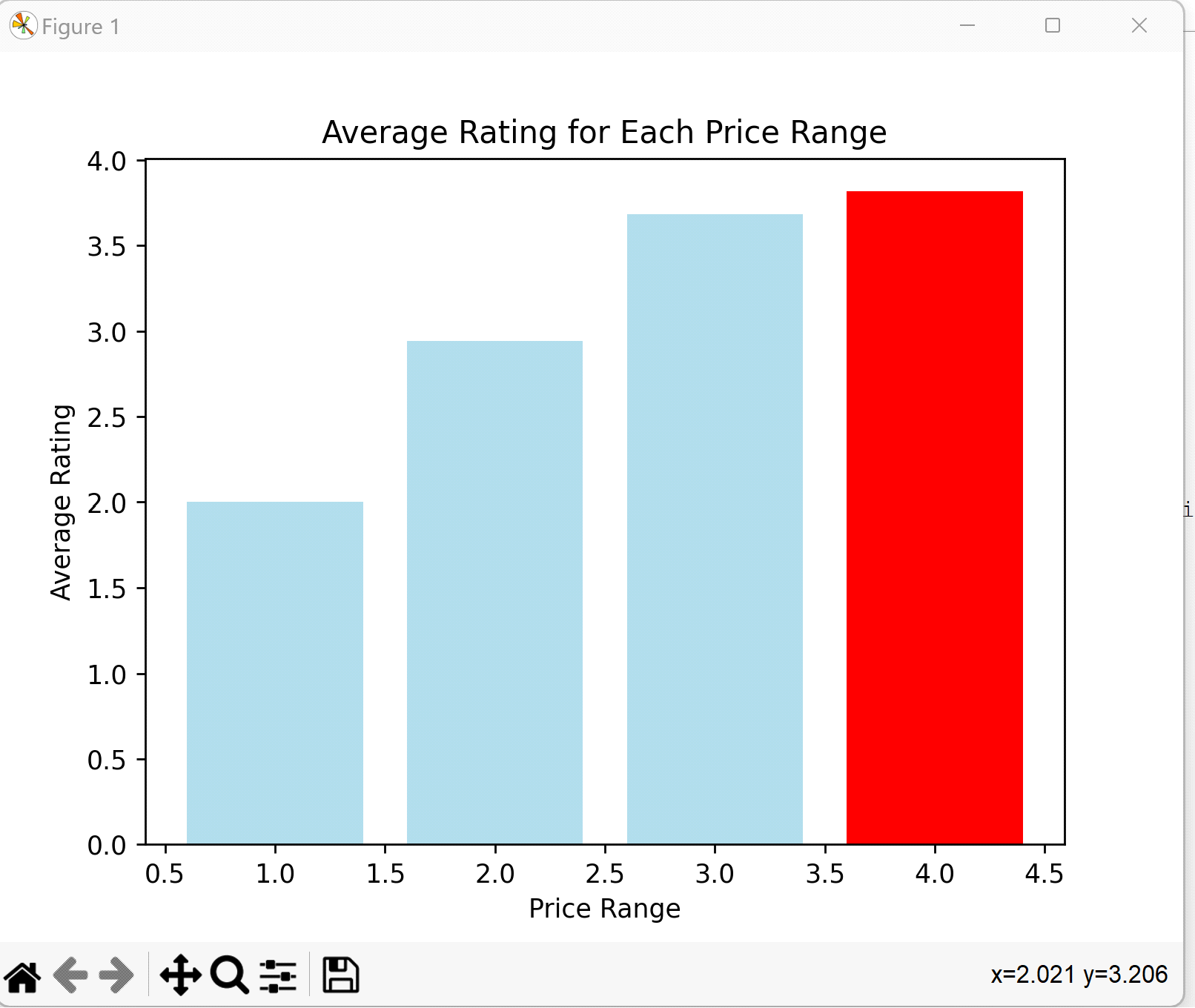
print(df\_encoded.head())

# Save the updated DataFrame to a new CSV file

df\_encoded.to\_csv('encoded\_dataset.csv', index=False)

**OUTPUT:**





**TASK 3:** **Feature Engineering**

* 1. Extract additional features from the existing columns, such as the length

of the restaurant name or address.

* 1. Create new features like "Has Table Booking" or "Has Online Delivery"

by encoding categorical variables

**CODE:**

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

import folium

from sklearn.model\_selection import train\_test\_split

from sklearn.linear\_model import LinearRegression

from sklearn.metrics import mean\_squared\_error, r2\_score

from sklearn.preprocessing import StandardScaler

from sklearn.tree import DecisionTreeRegressor

from sklearn.ensemble import RandomForestRegressor

# Replace 'path\_to\_your\_file.csv' with the actual path to your CSV file

file\_path = r'C:\Users\divya\Desktop\cognifyz\Dataset .csv'

# Read the CSV file into a pandas DataFrame

df = pd.read\_csv (file\_path)

#Task 3

# Assuming you have selected relevant features for prediction

# For example, let's consider features like 'Average Cost for two', 'Votes', 'Price range', etc.

selected\_features = ['Average Cost for two', 'Votes', 'Price range']

# Extract features and target variable

X = df[selected\_features]

y = df['Aggregate rating']

# Split the dataset into training and testing sets

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

# Standardize features using StandardScaler

scaler = StandardScaler()

X\_train\_scaled = scaler.fit\_transform(X\_train)

X\_test\_scaled = scaler.transform(X\_test)

# Build a Linear Regression model

model = LinearRegression()

model.fit(X\_train\_scaled, y\_train)

# Predict on the test set

y\_pred = model.predict(X\_test\_scaled)

# Evaluate the model

mse = mean\_squared\_error(y\_test, y\_pred)

r2 = r2\_score(y\_test, y\_pred)

# Display the evaluation metrics

print(f'Mean Squared Error (MSE): {mse:.2f}')

print(f'R-squared (R2): {r2:.2f}')

# Build and evaluate Linear Regression model

linear\_reg\_model = LinearRegression()

linear\_reg\_model.fit(X\_train, y\_train)

y\_pred\_linear\_reg = linear\_reg\_model.predict(X\_test)

# Build and evaluate Decision Tree model

decision\_tree\_model = DecisionTreeRegressor(random\_state=42)

decision\_tree\_model.fit(X\_train, y\_train)

y\_pred\_decision\_tree = decision\_tree\_model.predict(X\_test)

# Build and evaluate Random Forest model

random\_forest\_model = RandomForestRegressor(random\_state=42)

random\_forest\_model.fit(X\_train, y\_train)

y\_pred\_random\_forest = random\_forest\_model.predict(X\_test)

# Evaluate models

def evaluate\_model(name, y\_true, y\_pred):

    mse = mean\_squared\_error(y\_true, y\_pred)

    r2 = r2\_score(y\_true, y\_pred)

    print(f'{name} - Mean Squared Error (MSE): {mse:.2f}, R-squared (R2): {r2:.2f}')

evaluate\_model('Linear Regression', y\_test, y\_pred\_linear\_reg)

evaluate\_model('Decision Tree', y\_test, y\_pred\_decision\_tree)

evaluate\_model('Random Forest', y\_test, y\_pred\_random\_forest)

# Assuming you have a 'Cuisines' column and an 'Aggregate rating' column

# If your 'Cuisines' column contains multiple cuisines separated by commas, you may need to preprocess it.

# Plot a violin plot to visualize the distribution of ratings for each type of cuisine

plt.figure(figsize=(16, 8))

sns.violinplot(x='Cuisines', y='Aggregate rating', data=df)

plt.xticks(rotation=90)

plt.title('Relationship between Cuisine and Restaurant Rating')

plt.xlabel('Cuisine Type')

plt.ylabel('Aggregate Rating')

plt.show()

# Extracting cuisines and their corresponding votes

cuisine\_votes = df.groupby('Cuisines')['Votes'].sum().reset\_index()

# Sorting cuisines based on the total number of votes in descending order

popular\_cuisines = cuisine\_votes.sort\_values(by='Votes', ascending=False)

# Display the most popular cuisines

print("Most Popular Cuisines Based on Votes:")

print(popular\_cuisines.head())

# Create a boxplot to visualize the distribution of ratings for each cuisine

plt.figure(figsize=(16, 8))

sns.boxplot(x='Cuisines', y='Aggregate rating', data=df)

plt.xticks(rotation=90)

plt.title('Distribution of Ratings for Each Cuisine')

plt.xlabel('Cuisine Type')

plt.ylabel('Aggregate Rating')

plt.show()

#Histogram

# Assuming you have an 'Aggregate rating' column

plt.figure(figsize=(10, 6))

sns.histplot(df['Aggregate rating'], bins=30, kde=True)

plt.title('Distribution of Ratings')

plt.xlabel('Aggregate Rating')

plt.ylabel('Frequency')

plt.show()

#Bar plot

# Assuming you have an 'Aggregate rating' column

plt.figure(figsize=(10, 6))

sns.countplot(x='Aggregate rating', data=df, hue='Aggregate rating', palette='viridis', dodge=False, legend=False)

plt.title('Distribution of Ratings')

plt.xlabel('Aggregate Rating')

plt.ylabel('Count')

plt.show()

#Bar Plot for Cuisines

# Assuming you have 'Cuisines' and 'Aggregate rating' columns

plt.figure(figsize=(14, 8))

sns.barplot(x='Cuisines', y='Aggregate rating', data=df, err\_kws={'linewidth': 0}, palette='viridis', hue='Cuisines', dodge=False, legend=False)

plt.title('Average Ratings for Different Cuisines')

plt.xlabel('Cuisine Type')

plt.ylabel('Average Rating')

plt.xticks(rotation=90)

plt.show()

#Bar Plot for Cities

# Assuming you have 'City' and 'Aggregate rating' columns

plt.figure(figsize=(12, 6))

sns.barplot(x='City', y='Aggregate rating', data=df, errorbar=None, palette='viridis', hue='City', dodge=False, legend=False)

plt.title('Average Ratings for Different Cities')

plt.xlabel('City')

plt.ylabel('Average Rating')

plt.xticks(rotation=45)

plt.show()

plt.figure(figsize=(8, 6))

sns.scatterplot(x='Aggregate rating', y='Rating color', data=df, alpha=0.7)

plt.title('Scatter Plot: Relationship between Aggregate rating and Rating color')

plt.xlabel('Aggregate rating')

plt.ylabel('Rating color')

plt.show()

# Pairplot for multiple features against the target

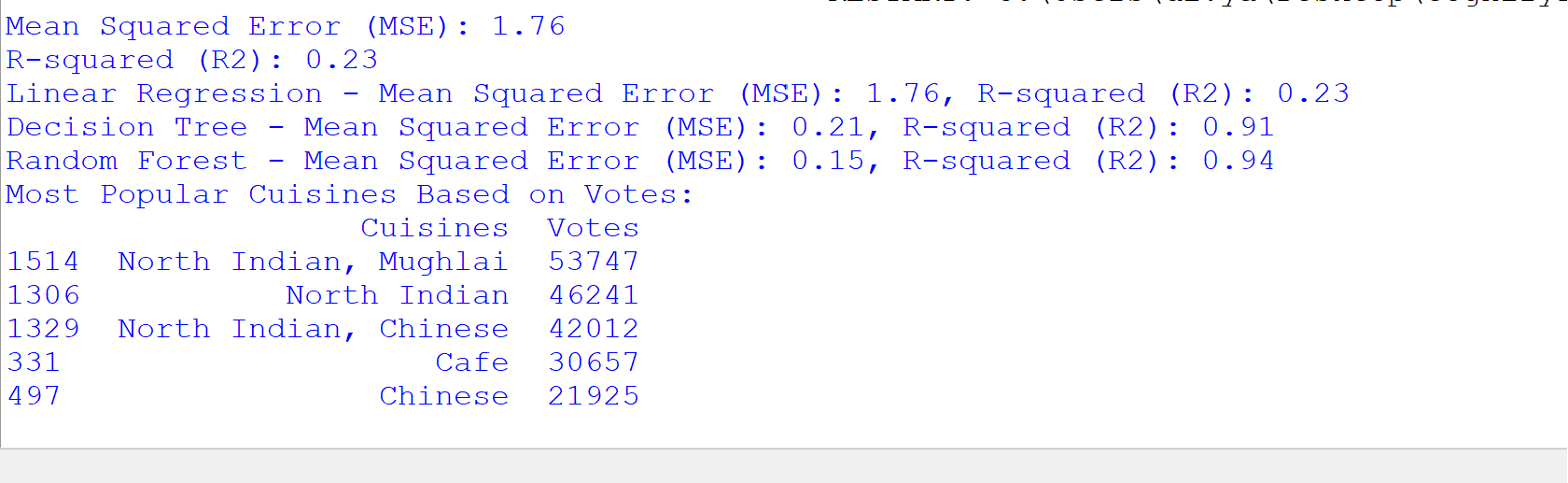
selected\_features = ['Aggregate rating', 'Rating color']

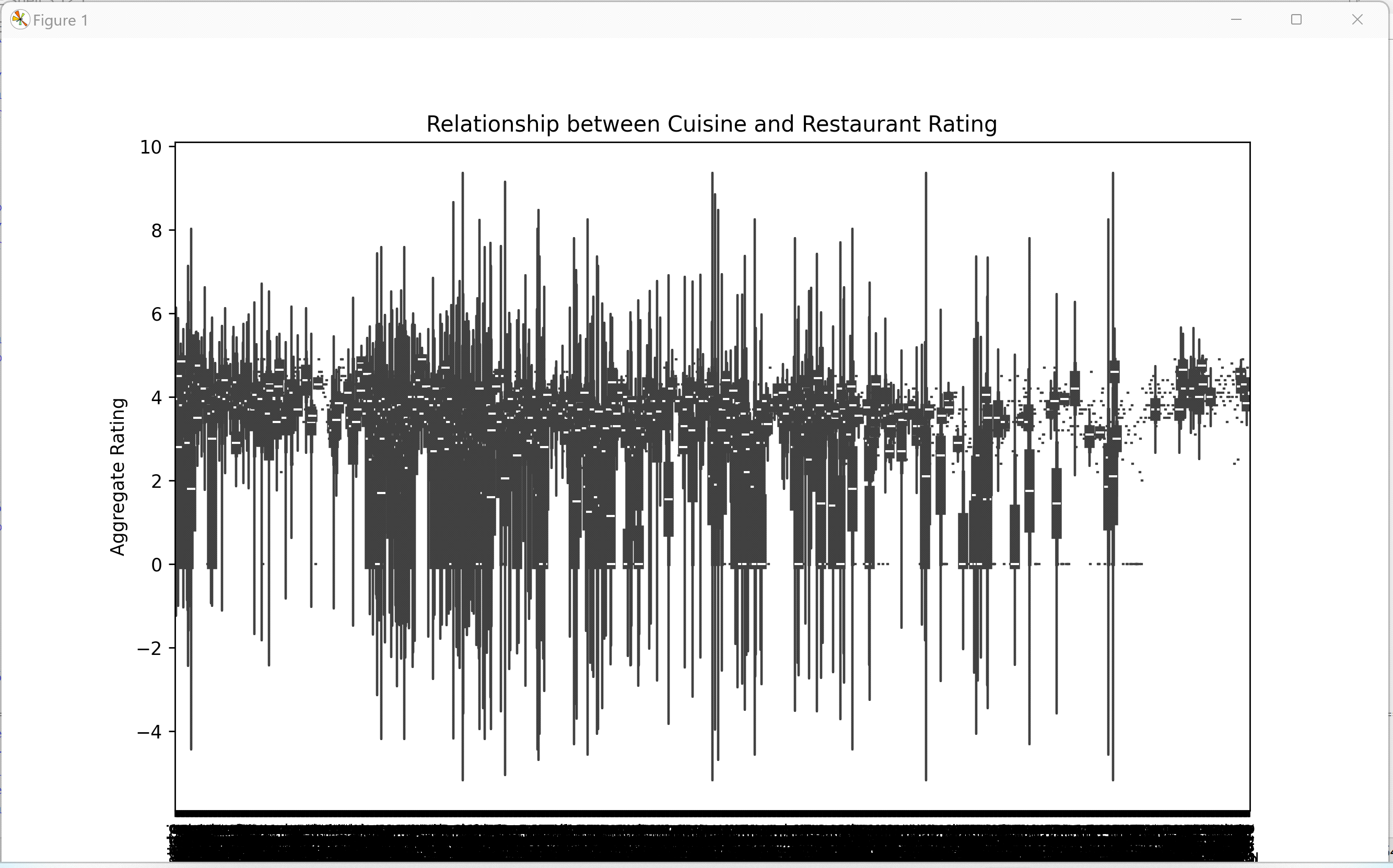
sns.pairplot(df[selected\_features], height=2)

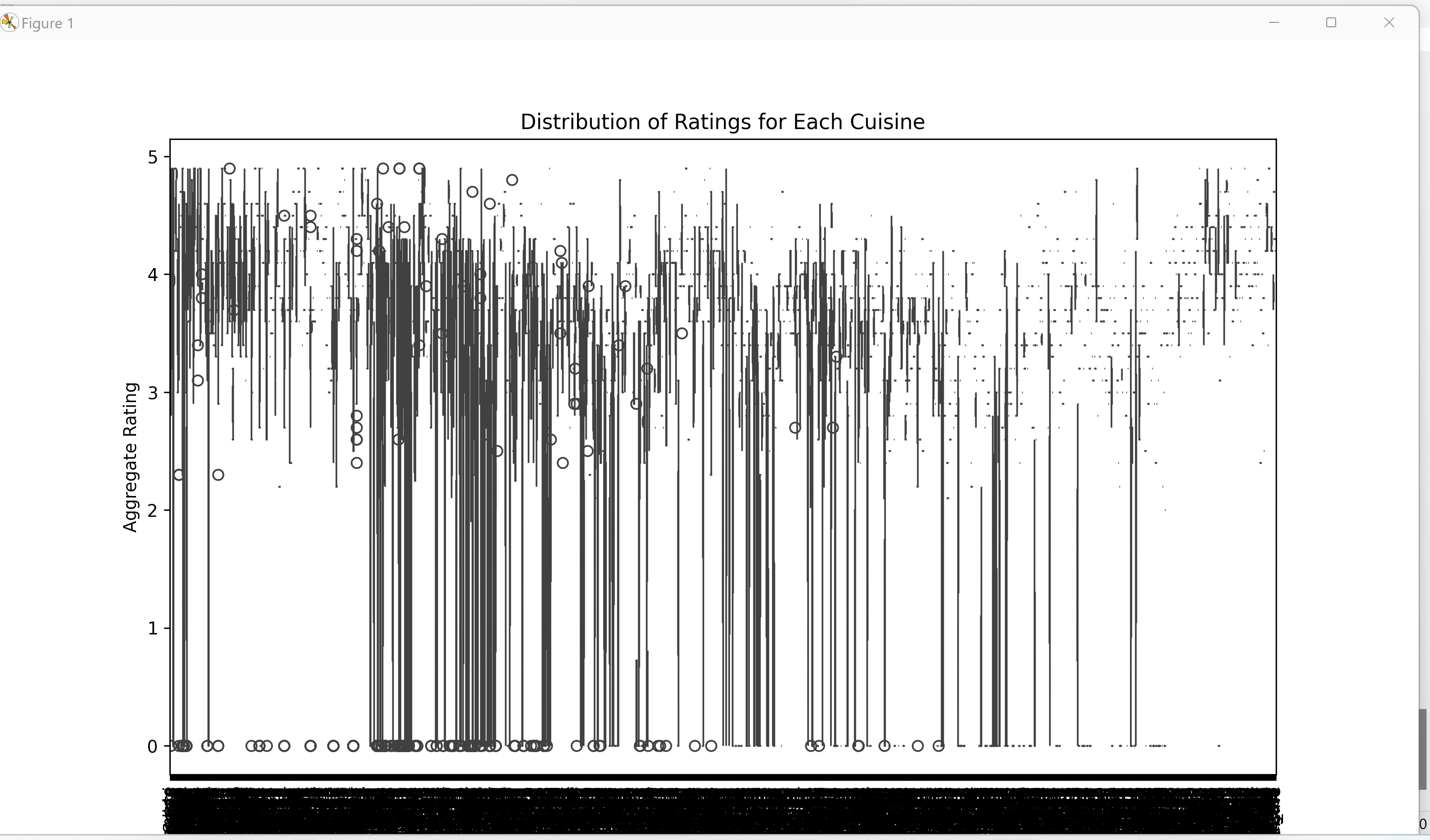
plt.suptitle('Pairplot of Aggregate rating against Rating color', y=1.02)

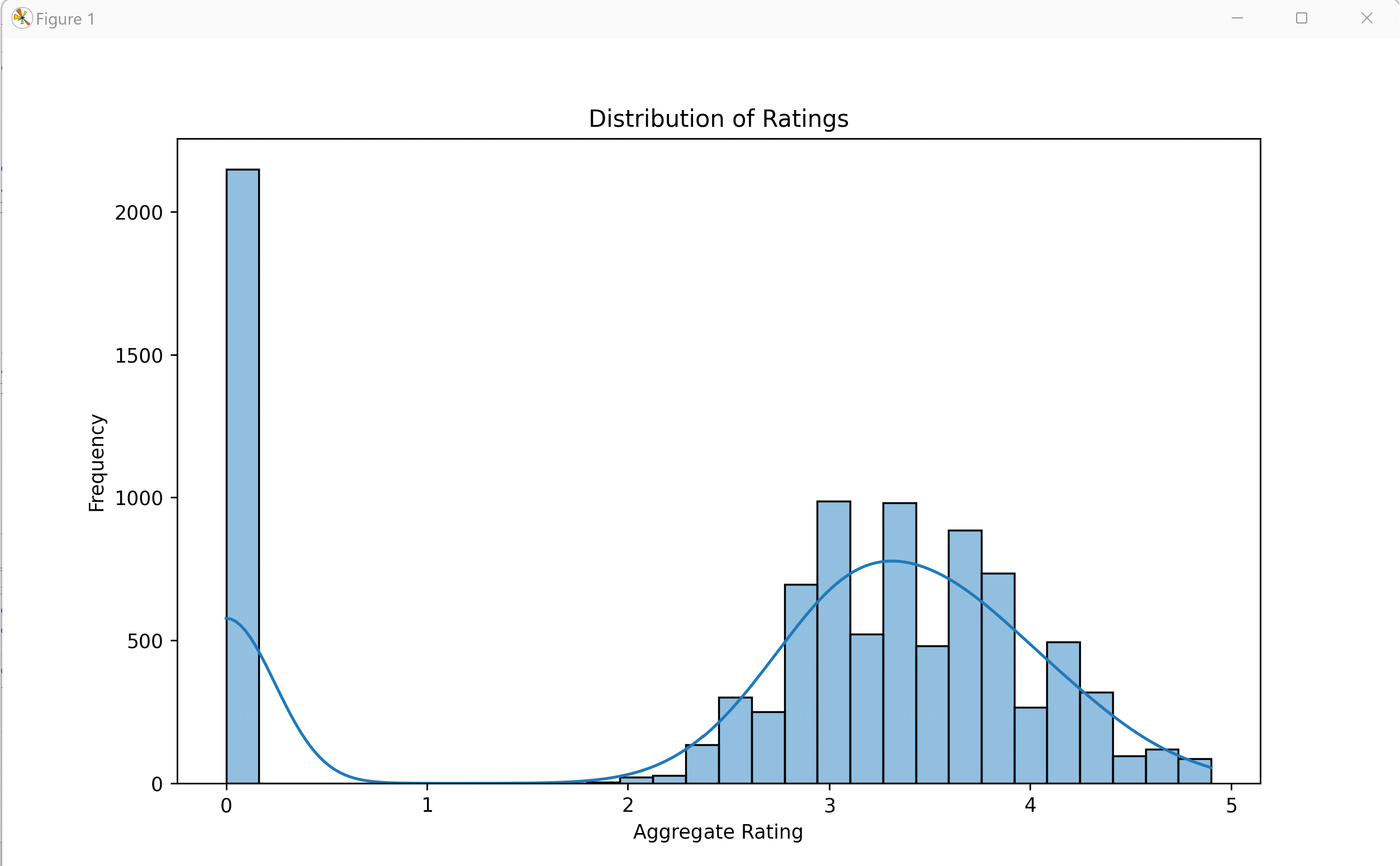
plt.show()

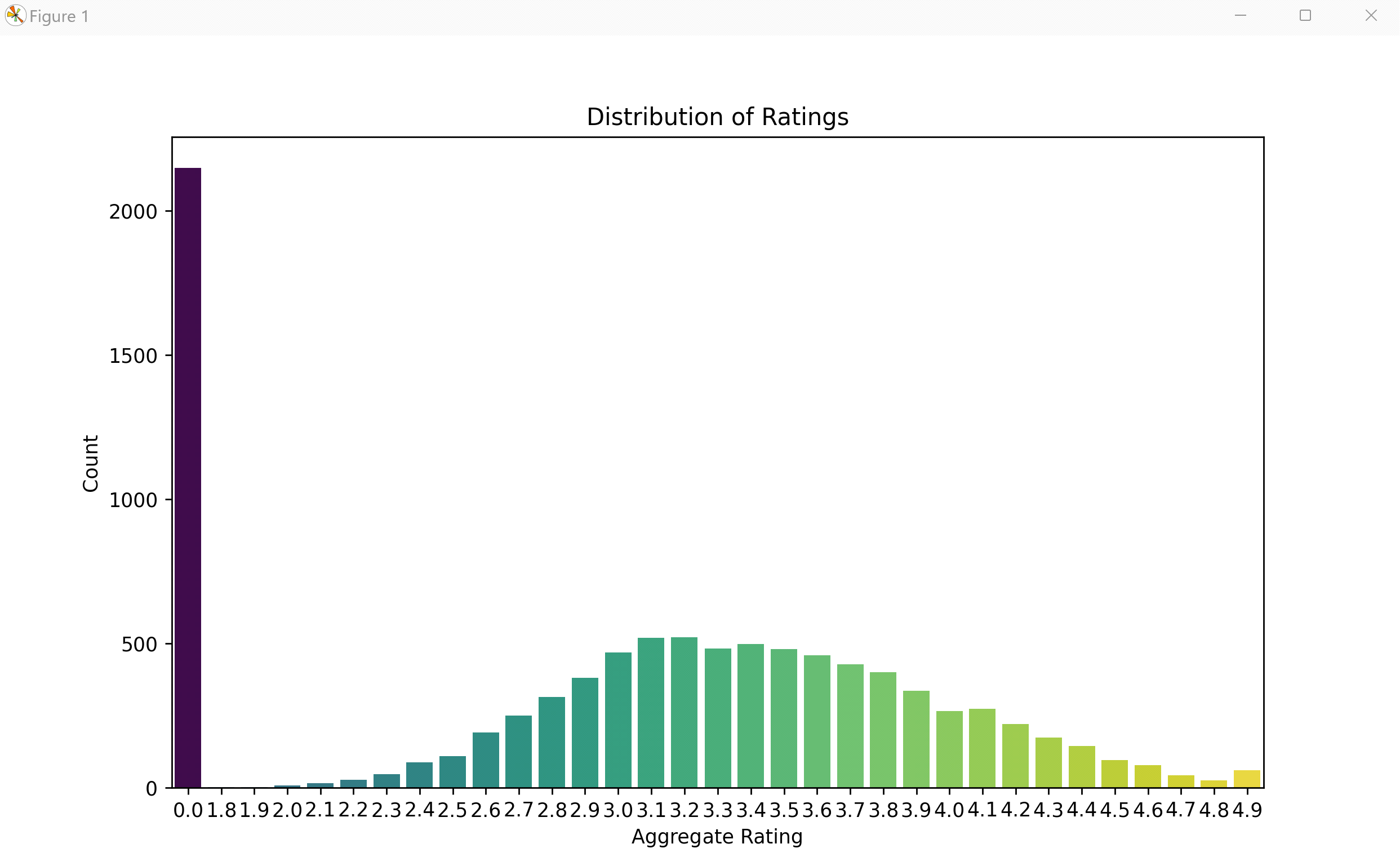
**OUTPUT:**

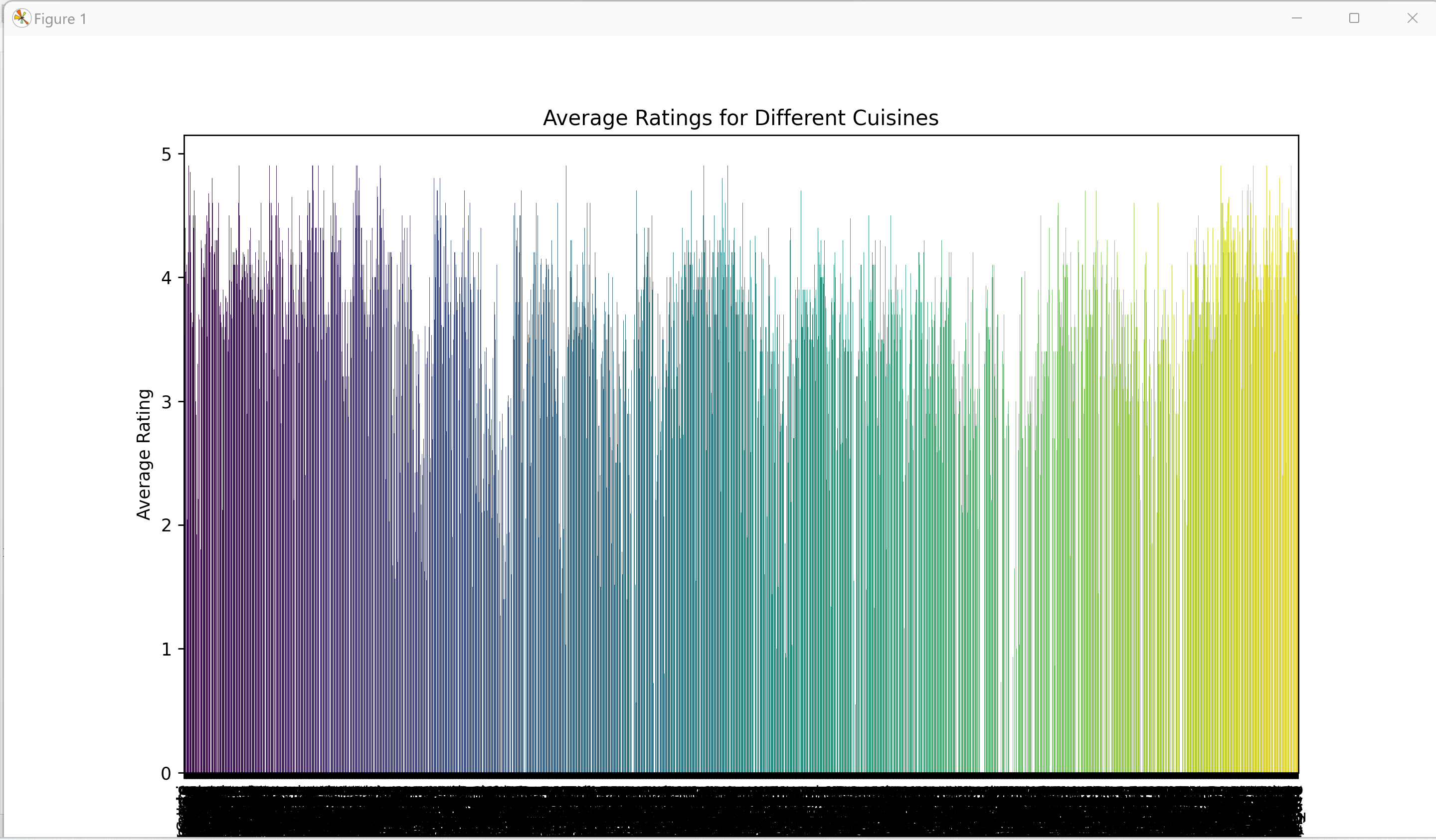


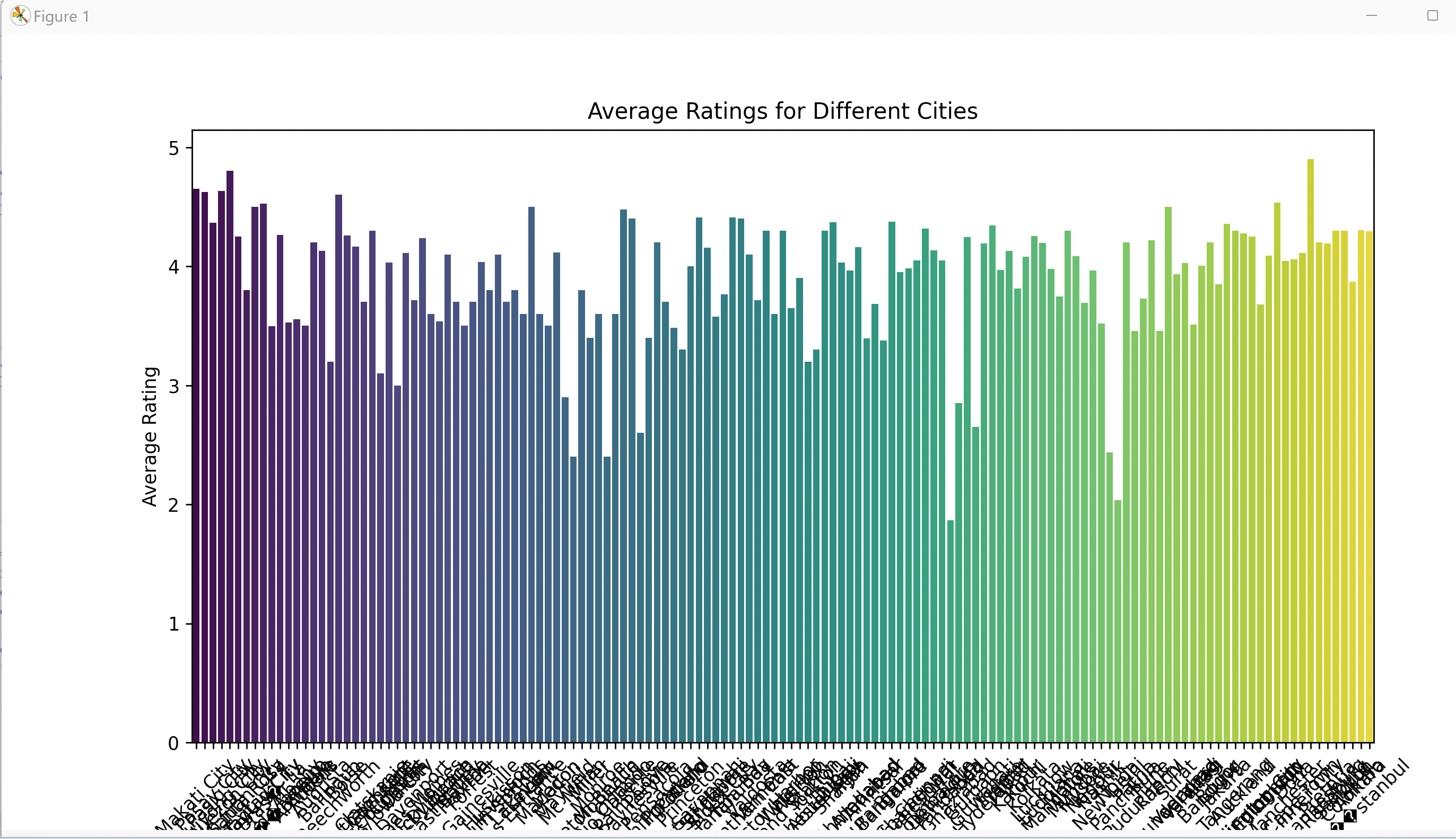


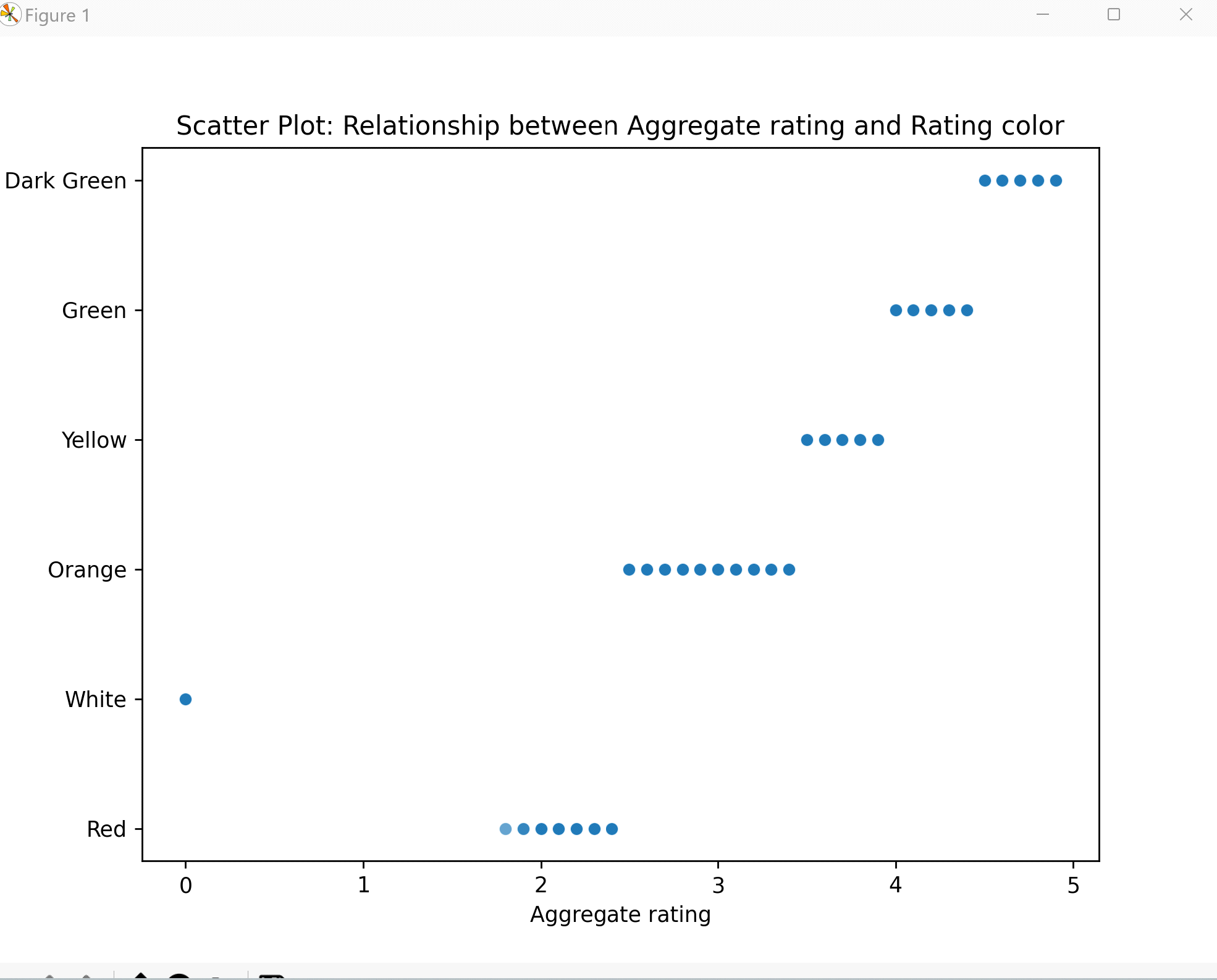


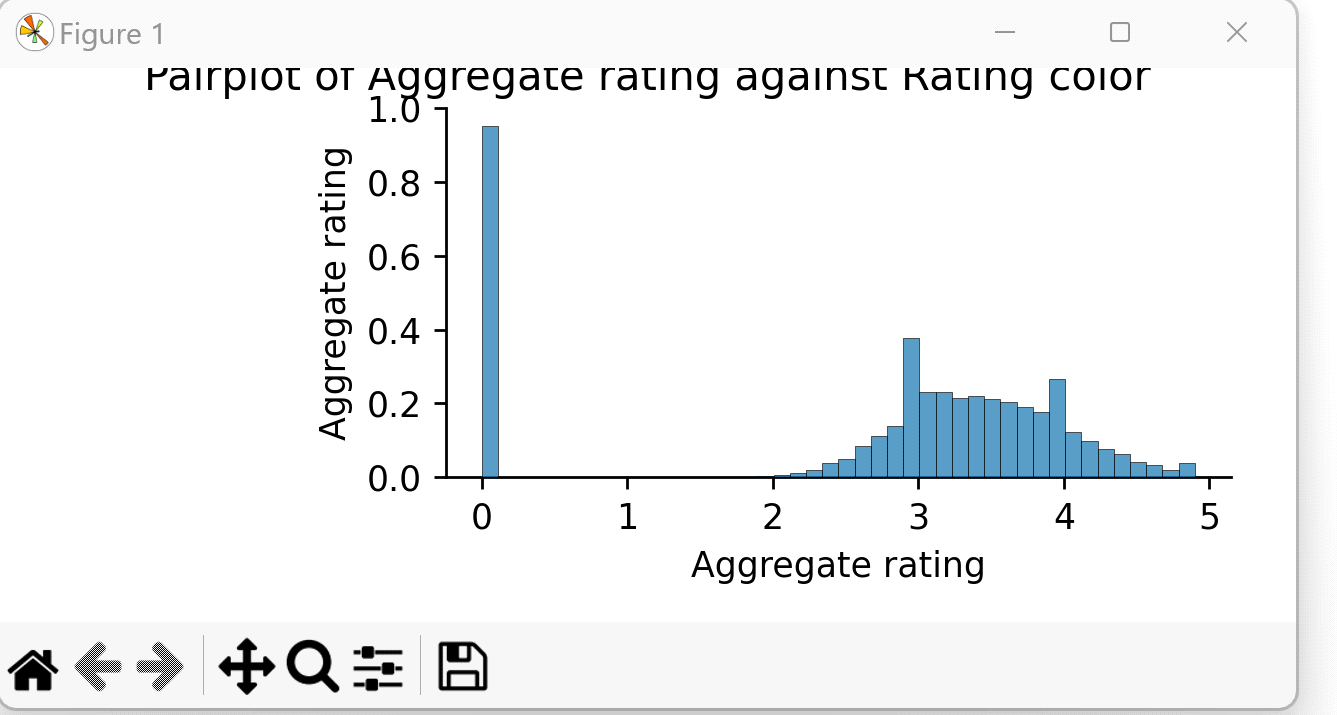












**Department Of Electronics and Telecommunication Engineering, MESWCOE**

# CONCLUSION

On the whole, this internship was a useful experience. I have gained new knowledge, skills. I achieved several of my learning goals. These projects mainly focus on the usage of the python programming language in the field of Data Science. This language has not only its own application in the field of just analysing the data but also for the prediction of the upcoming scenarios in this field. The purpose of using this specific language is due to its versatility, vast libraries (Pandas, NumPy, Matplotlib, etc.), speed limitations, and ease of learning. We will be analysing large energy data sets in this project which cannot be easily analysed in other tools as compared to python. Python does not have its limitation to only data analytics but also in many other fields such as Artificial intelligence, Machine learning, and many more.