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

WADIA 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

Cognifyz
Where Data Meets Intelligence

Cognifyz Technologies

Internship Completion Certificate

Date -26/01/2024

This is to certify that **Divyanshi Rathore**, **(Intern ID: CTI/A1/C9548)** Currently pursuing a B.E. from Modern Education Society's Wadia College Of Engineering, was working as a **Data Science** with Cognifyz Technologies from December 2023 - January 2024.

During this period, she has served as a Data Science Intern and has displayed remarkable dedication, sincerity, and a strong desire to learn. She has exhibited exceptional coordination skills and effective communication abilities. Moreover, her attention to detail has been truly impressive.

She has consistently approached new assignments and challenges with enthusiasm, showcasing her passion for Data Science. Her commitment and willingness to acquire new knowledge and skills have been evident throughout her internship.

We extend our best wishes to Divyanshi Rathore for a successful future, and we have no doubt that she will continue to excel in the field of Data Science.

With Regards,
Cognifyz Technologies

Cognifyz Technologies
Cognifyz Technologies
Nagpur, India

AICTE
All India Council for Technical Education

MSME
MICRO, SMALL & MEDIUM ENTERPRISES

cognifyztechnologies@gmail.com

www.cognifyz.com

POSTER

M.E.S. Wadia College of Engineering, Pune-411001
Department of Electronics & Telecommunication Engineering
Internship at Cognifyz
Submitted by: Divyanshi Rathore
Name of the Guide: Prof. S. S. Pansare

OBJECTIVES

- Gain hands-on experience in applying theoretical knowledge to real-world projects, enhancing technical skills relevant to the field
- Improve communication skills through interactions with colleagues, clients, and supervisors, learning to articulate ideas effectively in a professional environment
- Acquire a deeper understanding of industry practices, trends, and challenges, by observing and participating in day-to-day operations and projects.
- Build professional relationships and expand your network by connecting with professionals in your field, potentially opening doors for future career opportunities

CONCLUSION

In conclusion, this internship has been an invaluable opportunity for personal and professional growth. Through practical experience and mentorship, I have developed crucial skills relevant to my field. This internship in Data Science has been a transformative experience, offering valuable insights and practical skills essential for navigating the rapidly evolving landscape of Data Science technology. As I conclude this internship, I am confident that the knowledge and experiences gained will serve as a solid foundation for my future endeavours in the exciting field of Data Science.

About Company

Cognifyz Technologies is a leading technology company that specializes in the dynamic field of data science and excel 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 deliveringimpactful projects and solutionsCognifyz Technologies is a leading technology company that specializesin the dynamic field of data science and excelsin deliveringimpactful projects and solutions

Offer Letter:

internship offerletter.pdf

Completion Letter

internship certificate.pdf

SOFTWARE REQUIRED

1. Python Language
2. Jupyter Notebook
3. IDLE

TASKS

- TASK-1 Data Exploration and Preprocessing
- TASK-2 Price Range Analysis
- TASK-3Feature Engineering

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

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.

WEEKLY OVERVIEW OF INTERNSHIP ACTIVITIES

1ST WEEK	DATE	DAY	NAME OF THE TOPIC/MODULE COMPLETED
	22-12-23	Friday	Python Basics
	25-12-23	Monday	Python Basics
	26-12-23	Tuesday	Python Basics
	27-12-23	Wednesday	Python Basics
	28-12-23	Thursday	Python Basics & Data set Analysis

2ND WEEK	DATE	DAY	NAME OF THE TOPIC/MODULE COMPLETED
	01-01-24	Monday	TASK 1
	02-01-24	Tuesday	TASK 1 Analysis
	03-01-24	Wednesday	TASK 1 coding
	04-01-24	Thursday	TASK 1 coding
	05-01-24	Friday	TASK 1 completion

3RD WEEK	DATE	DAY	NAME OF THE TOPIC/MODULE COMPLETED
	08-01-24	Monday	TASK 2 Analysis
	09-01-24	Tuesday	TASK 2 coding
	10-01-24	Wednesday	TASK 2 coding
	11-01-24	Thursday	TASK 2 completion

4TH WEEK	DATE	DAY	NAME OF THE TOPIC/MODULE COMPLETED
	15-01-24	Monday	TASK 3 Analysis
	16-01-24	Tuesday	TASK 3 coding
	17-01-24	Wednesday	TASK 3 coding
	18-01-24	Thursday	TASK 3 coding
	19-01-24	Friday	TASK 3 completion

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.

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.

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.

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)

TECHNOLOGY

1. 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:

2. 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.

3. IDLE

IDLE is Python's Integrated Development and Learning Environment, which is cross-platform and works mostly the same on Windows, Unix, and macOS

Tasks and Outputs

Task 1 : Data Exploration and Preprocessing

a. Determine the percentage of restaurants that offer table booking and online delivery.
b. Compare the average ratings of restaurants with table booking and those without.
c. 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:

```
Number of rows: 9551
Number of columns: 21
Missing Values in Each Column:
Restaurant ID           0
Restaurant Name         0
Country Code            0
City                    0
Address                 0
Locality                0
Locality Verbose        0
Longitude               0
Latitude                0
Cuisines                9
Average Cost for two    0
Currency                0
Has Table booking       0
Has Online delivery     0
Is delivering now       0
Switch to order menu    0
Price range             0
Aggregate rating        0
Rating color            0
Rating text             0
Votes                   0
dtype: int64
Data types after conversion:
Restaurant ID            int64
Restaurant Name         object
Country Code             int64
City                    object
Address                 object
Locality                object
Locality Verbose        object
Longitude              float64
Latitude               float64
Cuisines                object
Average Cost for two     int64
Currency                object
```

Ln: 395 Col: 0

File Edit Shell Debug Options Window Help

```
Average Cost for two      int64
Currency                 object
Has Table booking        object
Has Online delivery      object
Is delivering now        object
Switch to order menu     object
Price range               int64
Aggregate rating        float64
Rating color             object
Rating text              object
Votes                     int64
dtype: object

Class distribution:
Aggregate rating
0.0    2148
3.2     522
3.1     519
3.4     498
3.3     483
3.5     480
3.0     468
3.6     458
3.7     427
3.8     400
2.9     381
3.9     335
2.8     315
4.1     274
4.0     266
2.7     250
4.2     221
2.6     191
4.3     174
4.4     144
2.5     110
4.5      95
2.4      87
4.6      78
4.9      61
2.3      47
4.7      42
```

```
2.3      47
4.7      42
2.2      27
4.8      25
2.1      15
2.0       7
1.9       2
1.8       1
Name: count, dtype: int64
Mean values:
Restaurant ID           9.051128e+06
Country Code            1.836562e+01
Longitude               6.412657e+01
Latitude                2.585438e+01
Average Cost for two    1.199211e+03
Price range             1.804837e+00
Aggregate rating        2.666370e+00
Votes                   1.569097e+02
Name: mean, dtype: float64

Median values:
Restaurant ID           6.004089e+06
Country Code            1.000000e+00
Longitude               7.719196e+01
Latitude                2.857047e+01
Average Cost for two    4.000000e+02
Price range             2.000000e+00
Aggregate rating        3.200000e+00
Votes                   3.100000e+01
Name: 50%, dtype: float64

Standard Deviation values:
Restaurant ID           8.791521e+06
Country Code            5.675055e+01
Longitude               4.146706e+01
Latitude                1.100794e+01
Average Cost for two    1.612118e+04
Price range             9.056088e-01
Aggregate rating        1.516378e+00
Votes                   4.301691e+02
Name: std, dtype: float64

Distribution of Country Code
```

```
Name: std, dtype: float64

Distribution of Country Code:
Country Code
1      8652
216     434
215      80
30       60
214      60
189      60
148      40
208      34
14       24
162      22
94       21
184      20
166      20
191      20
37        4
Name: count, dtype: int64

Distribution of City:
City
New Delhi           5473
Gurgaon             1118
Noida               1080
Faridabad            251
Ghaziabad             25
                     ...
Panchkula              1
Mc Millan              1
Mayfield               1
Macedon                1
Vineland Station       1
Name: count, Length: 141, dtype: int64

Distribution of Cuisines:
Cuisines
North Indian      3960
Chinese           2735
Fast Food         1986
Mughlai            995
```

```
Fast Food          1986
Mughlai             995
Italian             764
                    ...
Fish and Chips        1
Malwani               1
Cuisine Varies        1
Soul Food             1
B?_rek                1
Name: count, Length: 145, dtype: int64

Top Cuisines:
Cuisines
North Indian    3960
Chinese         2735
Fast Food       1986
Mughlai          995
Italian          764
Bakery           745
Continental      736
Cafe             703
Desserts         653
South Indian     636
Name: count, dtype: int64

Top Cities with the Highest Number of Restaurants:
City
New Delhi       5473
Gurgaon         1118
Noida           1080
Faridabad        251
Ghaziabad         25
Bhubaneshwar      21
Amritsar          21
Ahmedabad         21
Lucknow           21
Guwahati          21
Name: count, dtype: int64
```

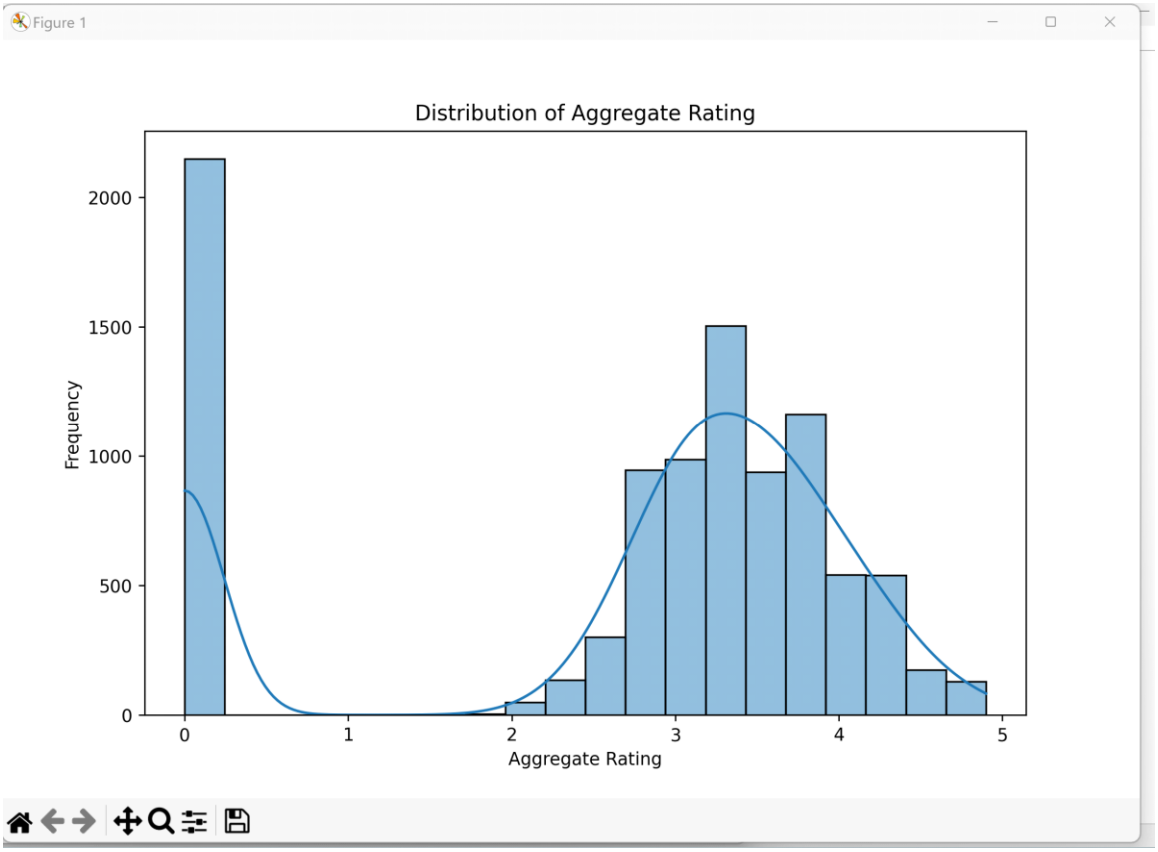


Figure 1
Distribution of Aggregate Rating
Frequency
2000
1500
1000
500
0
0
1
2
3
4
5
Aggregate Rating

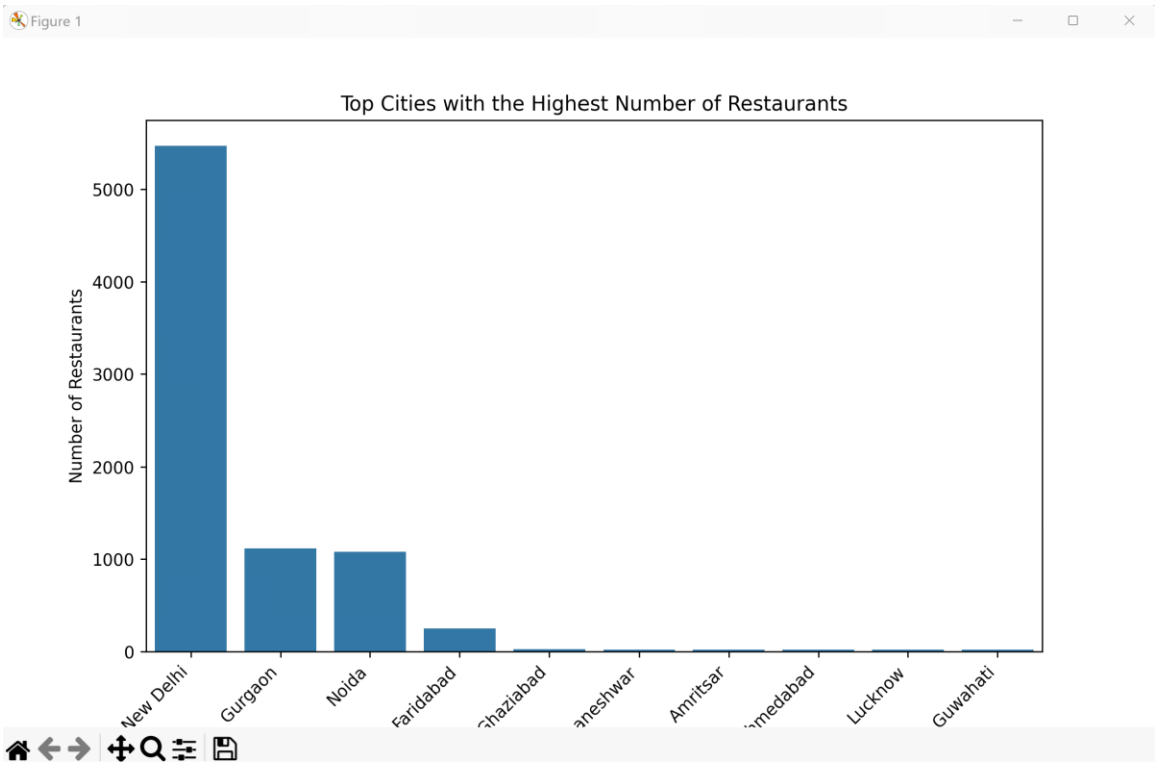


Figure 1
Top Cities with the Highest Number of Restaurants
Number of Restaurants
0
1000
2000
3000
4000
5000
New Delhi
Gurgaon
Noida
Faridabad
Amritsar
Lucknow
Guwahati

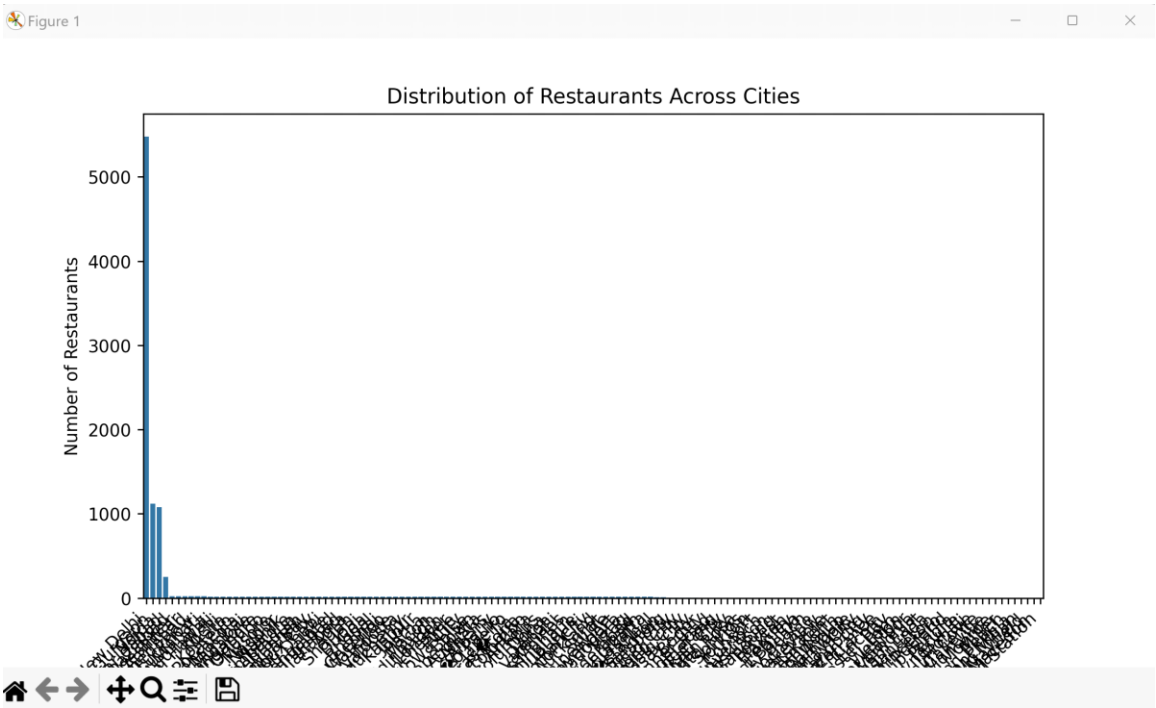


Figure 1
Distribution of Restaurants Across Cities
Number of Restaurants
0
1000
2000
3000
4000
5000

TASK 2: Price Range Analysis

a. Determine the most common price range among all the restaurants.
b. Calculate the average rating for each price range.
c. 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
```

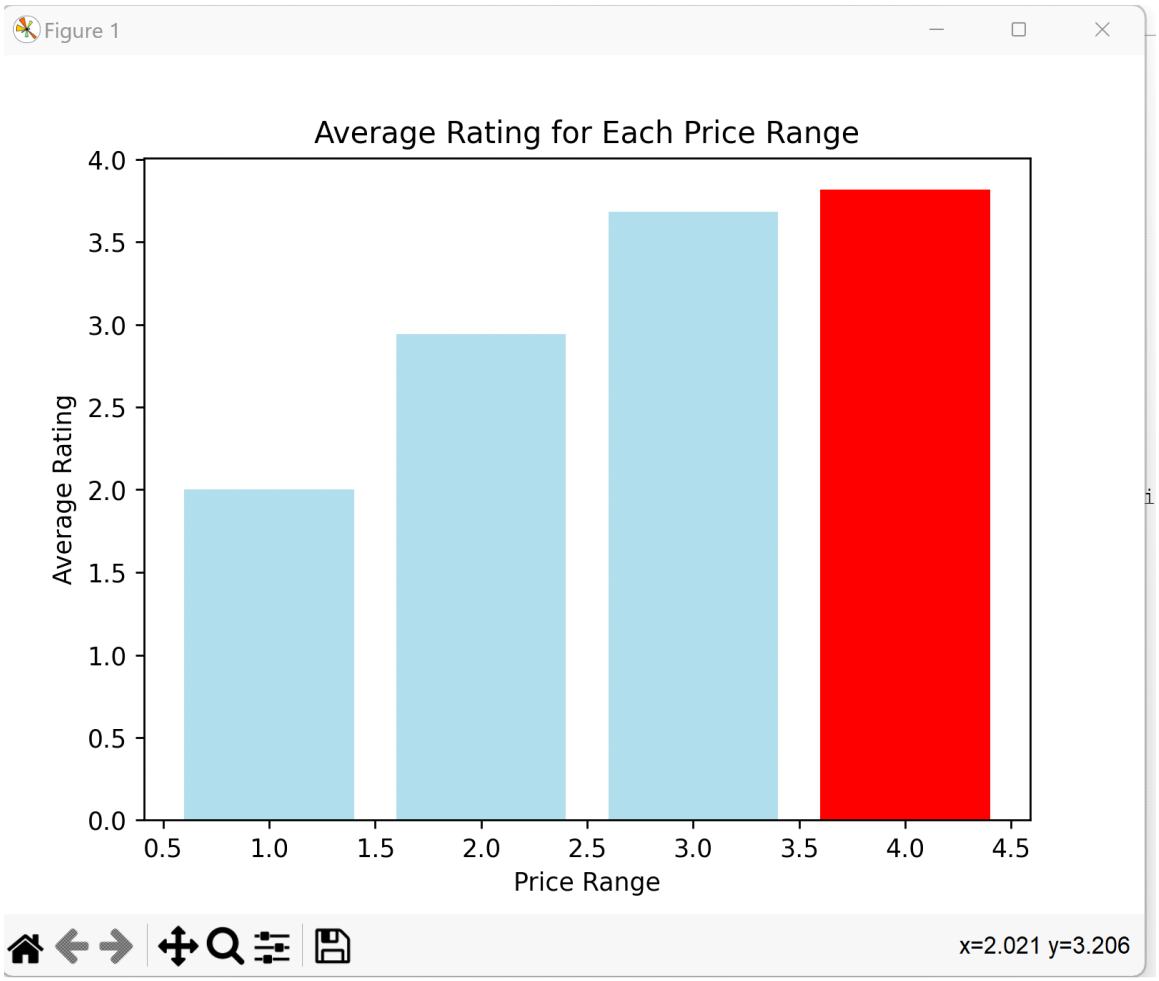
```
table_booking_percentage = (restaurants_with_table_booking / total_restaurants) * 100
online_delivery_percentage = (restaurants_with_online_delivery / total_restaurants) *
100

# 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:

```
Percentage of Restaurants Offering Table Booking: 12.12%
Percentage of Restaurants Offering Online Delivery: 25.66%
Average Rating for Restaurants with Table Booking: 3.44
Average Rating for Restaurants without Table Booking: 2.56
Unique Price Ranges: [3 4 2 1]
Unique Online Delivery Options: ['No' 'Yes']

Availability of Online Delivery Among Restaurants with Different Price R
Has Online delivery        No       Yes
Price range
1                    0.842259  0.157741
2                    0.586894  0.413106
3                    0.708097  0.291903
4                    0.909556  0.090444
Most Common Price Range:  1
Average Rating for Each Price Range:
Price range
1    1.999887
2    2.941054
3    3.683381
4    3.817918
Name: Aggregate rating, dtype: float64
Updated DataFrame with Additional Features:
   Restaurant ID  ... Address Length
0        6317637  ...             71
1        6304287  ...             67
2        6300002  ...             56
3        6318506  ...             70
4        6314302  ...             64

[5 rows x 23 columns]
Updated DataFrame with New Features:
   Restaurant ID  ... Has Online delivery_Yes
0        6317637  ...                   False
1        6304287  ...                   False
2        6300002  ...                   False
3        6318506  ...                   False
4        6314302  ...                   False

[5 rows x 23 columns]
```



TASK 3: Feature Engineering

a. Extract additional features from the existing columns, such as the length of the restaurant name or address.
b. 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)
```

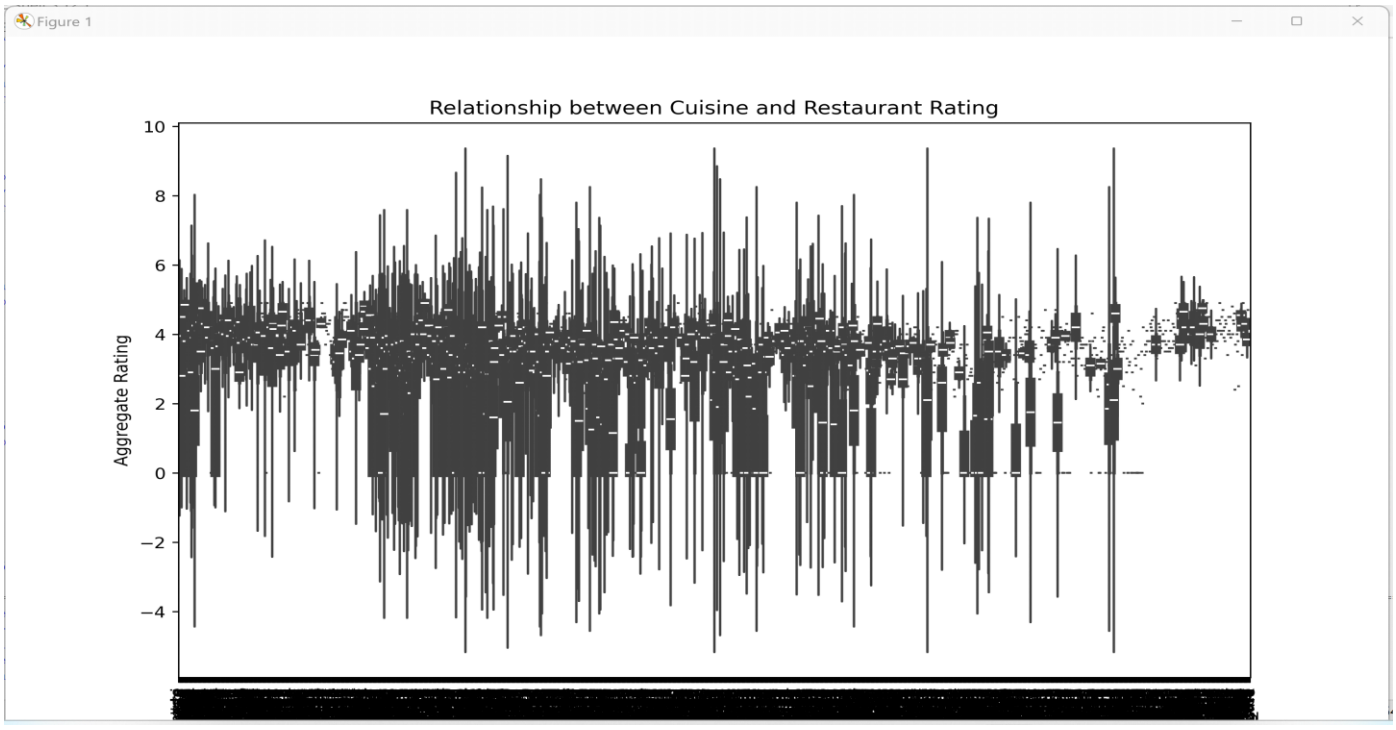
```
r2 = r2_score(y_test, y_pred)

# 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:

```
Mean Squared Error (MSE): 1.76
R-squared (R2): 0.23
Linear Regression - Mean Squared Error (MSE): 1.76, R-squared (R2): 0.23
Decision Tree - Mean Squared Error (MSE): 0.21, R-squared (R2): 0.91
Random Forest - Mean Squared Error (MSE): 0.15, R-squared (R2): 0.94
Most Popular Cuisines Based on Votes:
                  Cuisines  Votes
1514  North Indian, Mughlai  53747
1306          North Indian  46241
1329  North Indian, Chinese  42012
331                   Cafe  30657
497                Chinese  21925
```



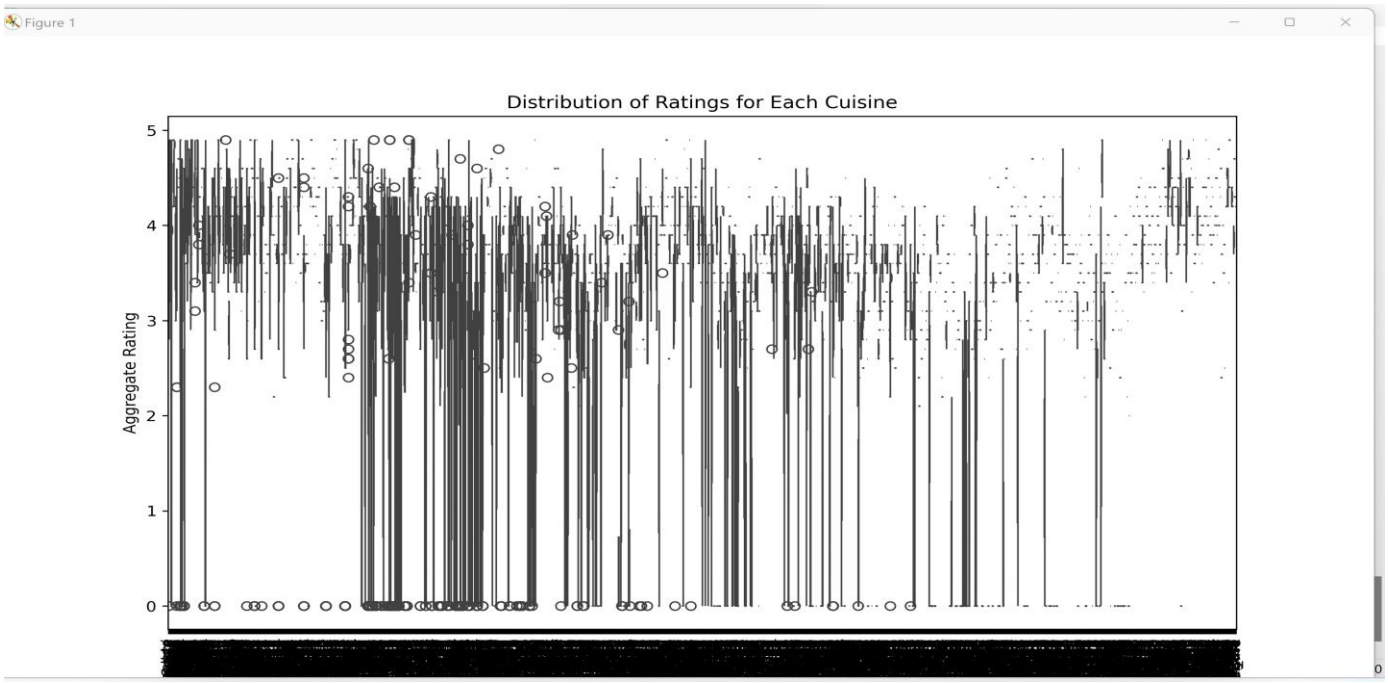


Figure 1
Distribution of Ratings for Each Cuisine
Aggregate Rating
0
1
2
3
4
5

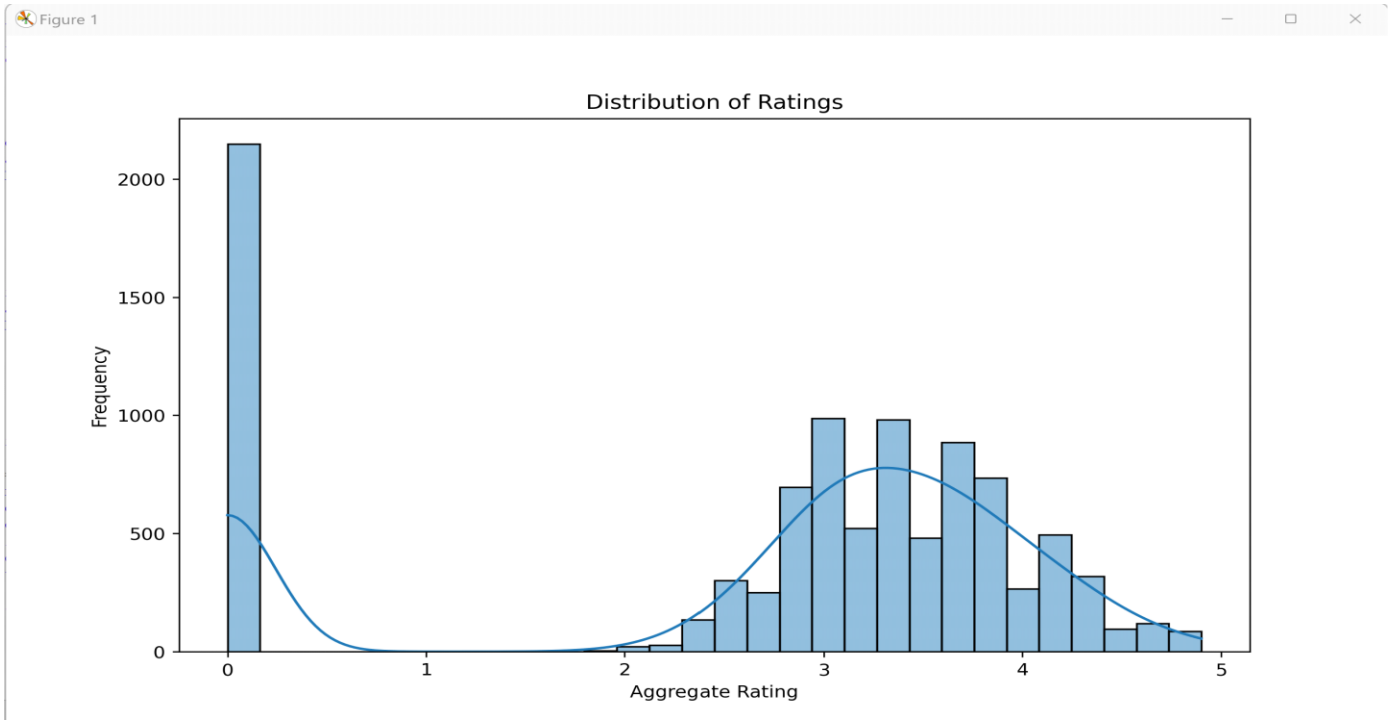


Figure 1
Distribution of Ratings
Frequency
0
500
1000
1500
2000
Aggregate Rating
0
1
2
3
4
5

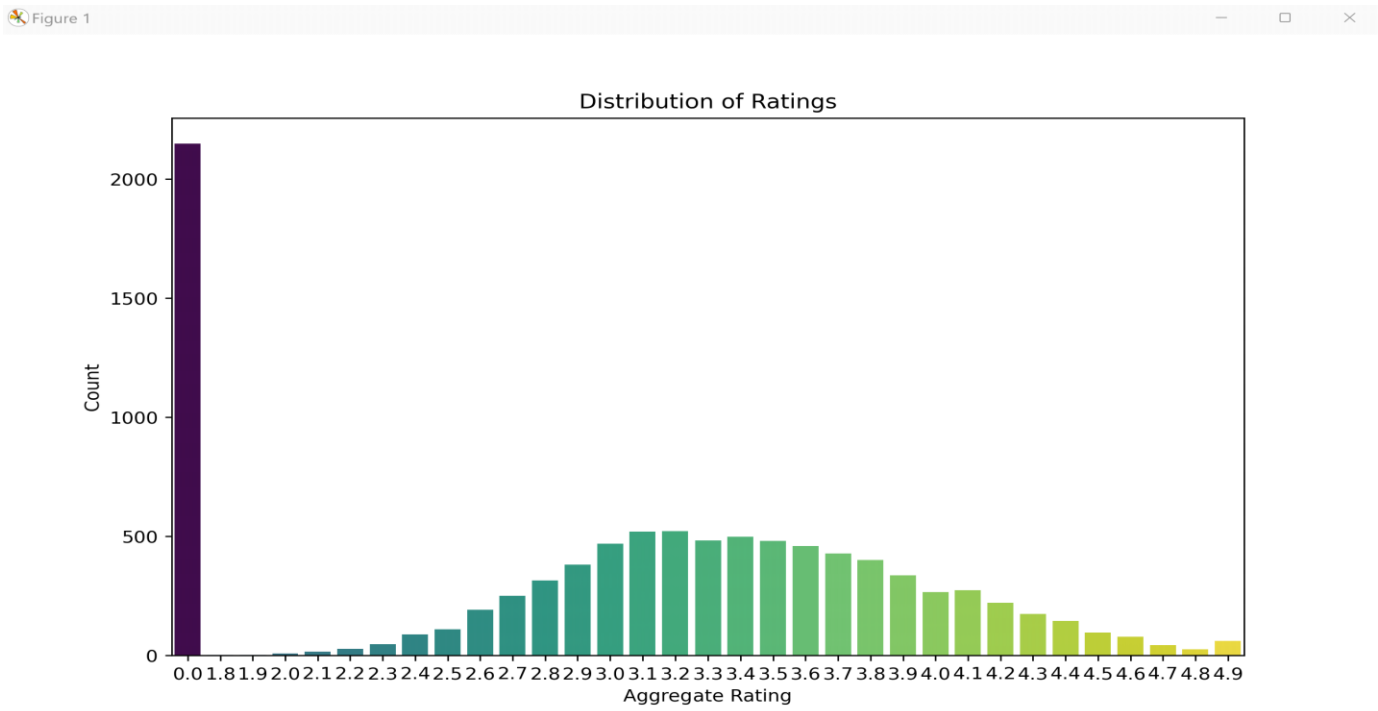


Figure 1
Distribution of Ratings
Count
2000
1500
1000
500
0
0.0 1.8 1.9 2.0 2.1 2.2 2.3 2.4 2.5 2.6 2.7 2.8 2.9 3.0 3.1 3.2 3.3 3.4 3.5 3.6 3.7 3.8 3.9 4.0 4.1 4.2 4.3 4.4 4.5 4.6 4.7 4.8 4.9
Aggregate Rating

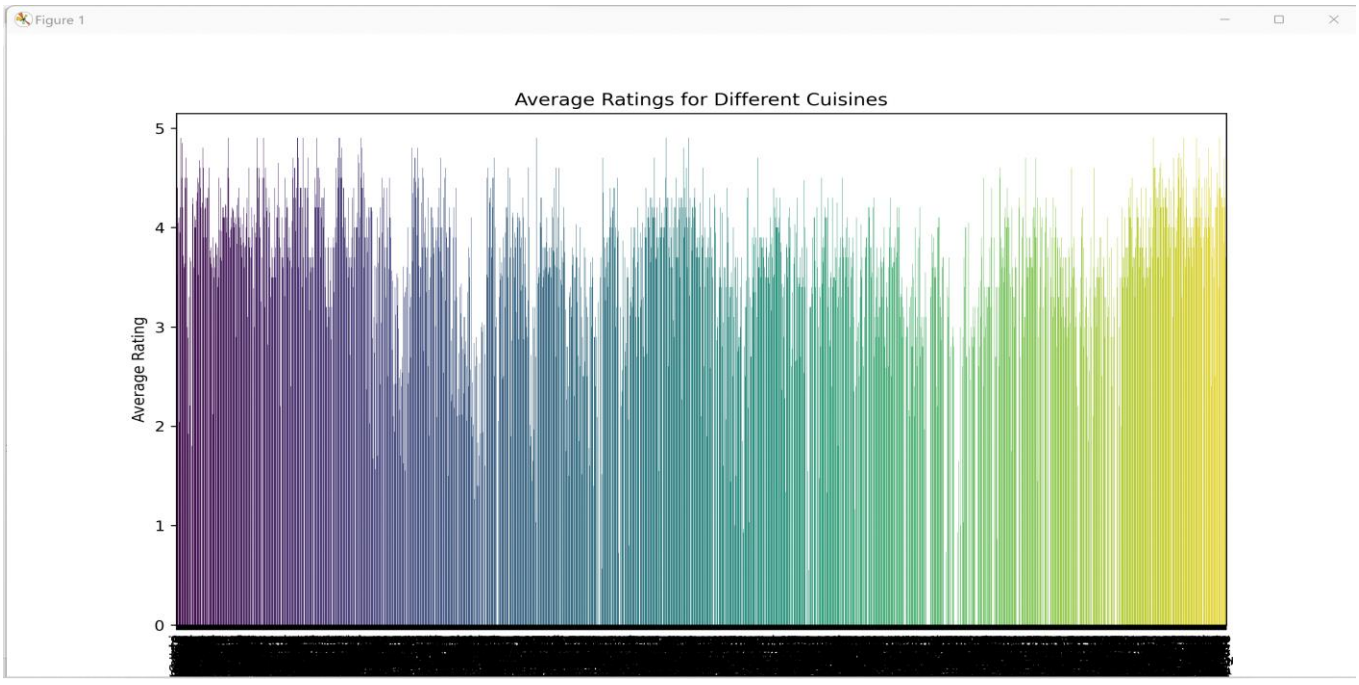


Figure 1
Average Ratings for Different Cuisines
Average Rating
5
4
3
2
1
0

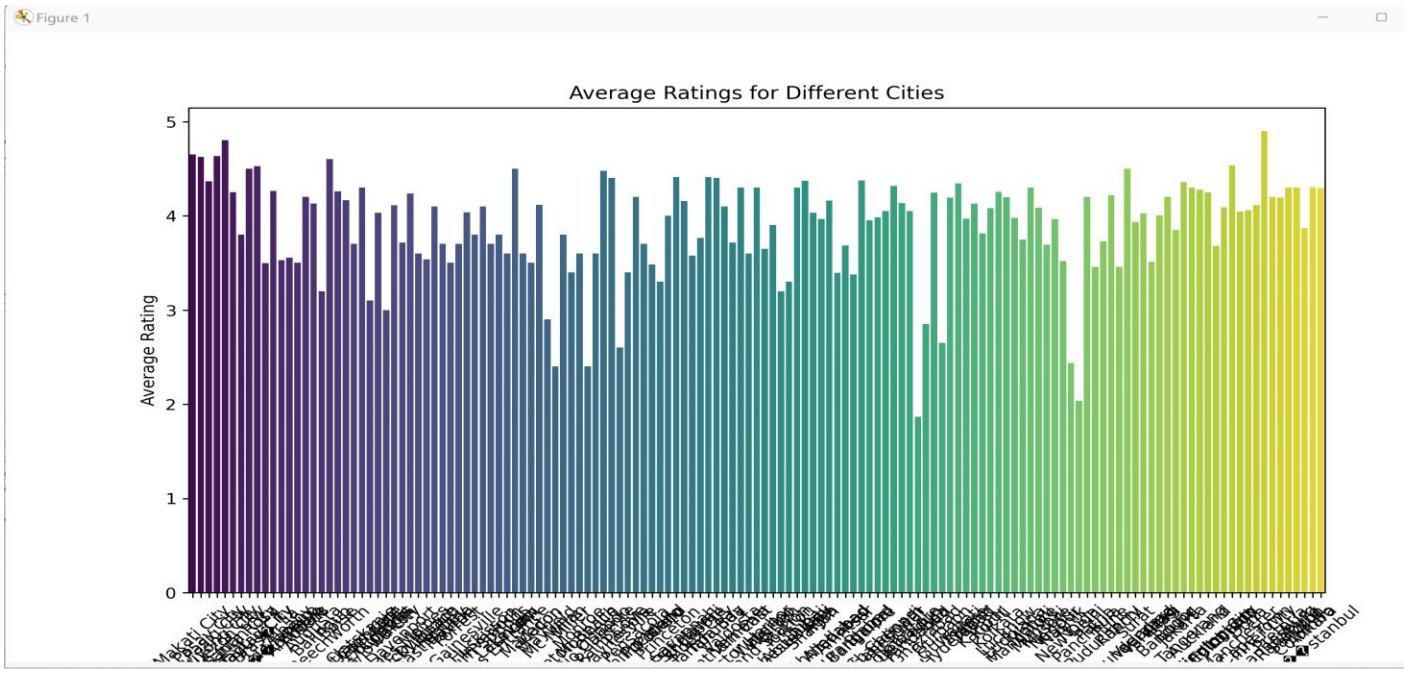


Figure 1
Average Ratings for Different Cities
Average Rating
0
1
2
3
4
5

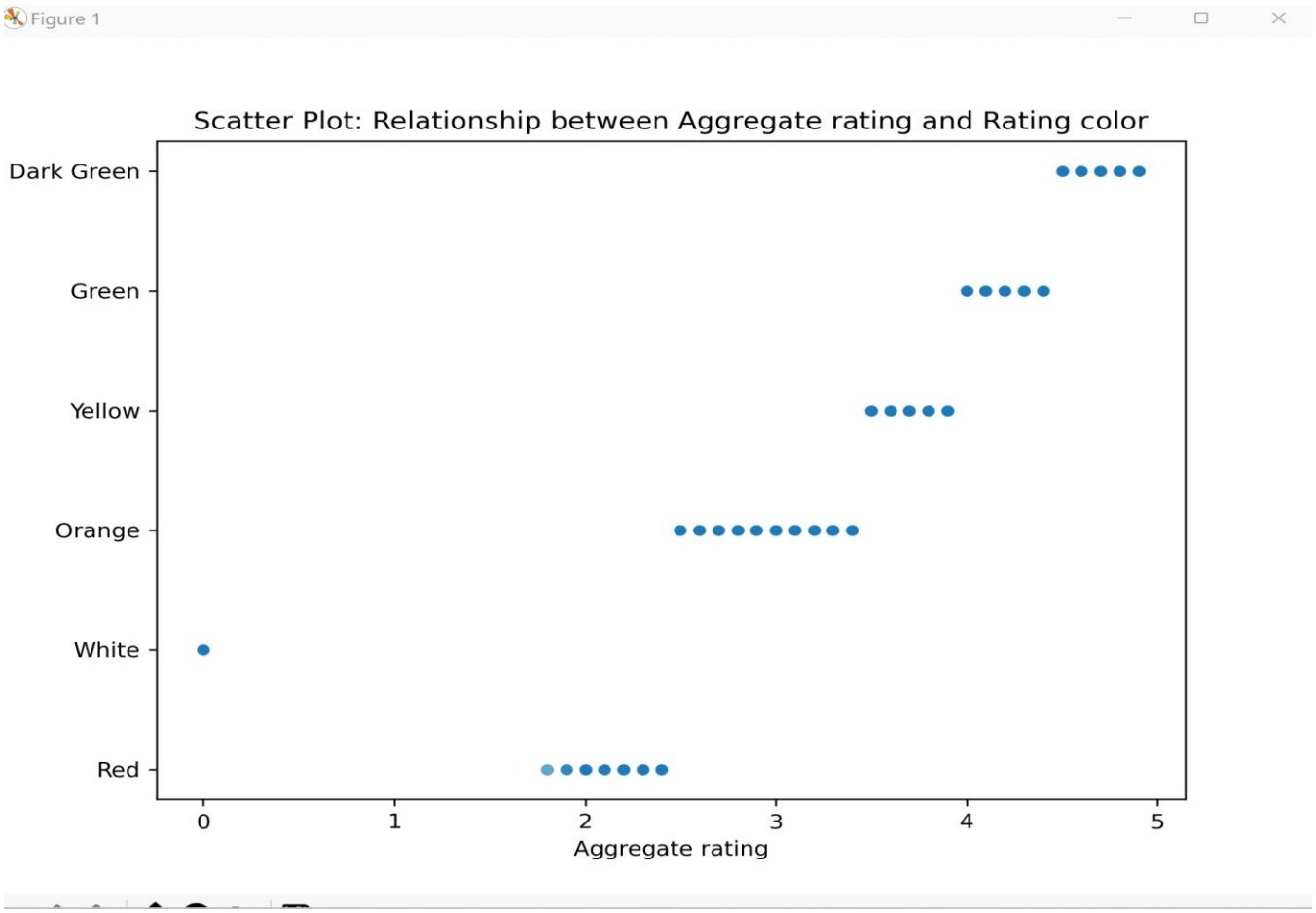


Figure 1
Scatter Plot: Relationship between Aggregate rating and Rating color
Dark Green
Green
Yellow
Orange
White
Red
0
1
2
3
4
5
Aggregate rating

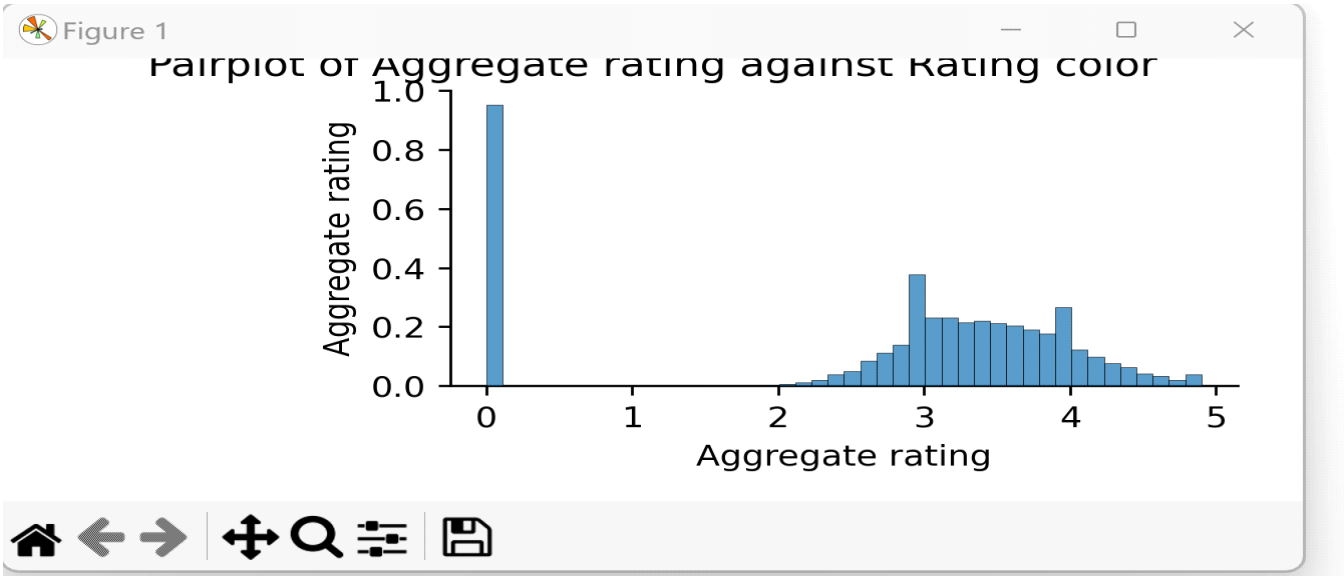


Figure 1
Pairplot of Aggregate rating against Rating color
Aggregate rating
1.0
0.8
0.6
0.4
0.2
0.0
0
1
2
3
4
5
Aggregate rating

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.