A

Course End Project Report on

HYDERABAD RESTAURANTS

Is submitted in partial fulfillment of the Requirements for the Award of CIE of

DATA ANALYSIS AND VISUALIZATION-22ADE01

in

B.E, IV-SEM, INFORMATION TECHNOLOGY

Submitted by

TARUN, 160122737051

YASWANTH, 160122737045

RUTHWIK,160122737035

COURSE TAUGHT BY:

Dr Ramakrishna Kolikipogu Professor, Dept of IT.

DEPARTMENT OF INFORMATION TECHNOLOGY CHAITANYA BHARATHI INSTITUTE OF TECHNOLOGY(A)

(Affiliated to Osmania University;Accredited by NBA,NAAC,ISO) kokapet(V),GANDIPET(M),HYDERABAD-500075

Website:www.cbit.ac.in

2023-2024

CERTIFICATE

This is to certify that the course end project work entitled **"Hyderbad Restaurants"** is submitted by **Tarun(160122737051), Yaswanth(160122737045), ruthwik(160122737035)** in partial fulfillment of the requirements for the award of CIE Marks of **DATA ANALYSIS AND VISUALIZATION (22ADE01)** of **B.E, IV-SEM, INFORMATION TECHNOLOGY** to CHAITANYA BHARATHI INSTITUTE OF TECHNOLOGY(A) affiliated to OSMANIA UNIVERSITY, Hyderabad is a record of bonofide work carried out by them under my supervision and guidance. The results embodied in this report have not been submitted to any other University or Institute for the award of any other Degree or Diploma.

Signature of Course Faculty
Dr Ramakrishna Kolikipogu
Professor of IT

Acknowledgement

The satisfaction that accompanies the successful completion of the task would be put incomplete without the mention of the people who made it possible, whose constant guidance and encouragement crown all the efforts with success.

We wish to express our deep sense of gratitude to **Dr Ramakrishna Kolikipogu, Professor of IT** for his able guidance and useful suggestions, which helped us in completing the Course End Project in time.

We are particularly thankful to **HoD, Principal and Management**, for their support and encouragement, which helped us to mould our project into a successful one.

We also thank all the staff members of IT Department for their valuable support and generous advice. Finally thanks to all our friends and family members for their continuous support and enthusiastic help.

Tarun, 160122737051
Yaswanth, 160122737045
Ruthwik,160122737035

Abstract

This project dives into Zomato's data on Hyderabad's restaurants, offering valuable insights for both diners and businesses. With details on restaurant names, ratings, cuisines, and prices, it provides a comprehensive look at the city's dining scene. Through straightforward analysis and visualization, we aim to reveal customer preferences, pricing patterns, and popular dining spots. This data-driven approach empowers stakeholders to make informed decisions, enriching the culinary experience and fostering growth in Hyderabad's vibrant restaurant community.

.

Table of Contents

List of Figures

Abbreviations

Abbreviation	Description
DAV	Data Analysis and Visualization
SB	Sea Born
PD	pandas
CSV	Comma Separated Value
HIST	Histogram

CHAPTER 1
Introduction

1.1 Definition of Problem

Analyze the Zomato restaurant dataset to understand customer sentiment, predict ratings, and identify cuisine preferences by location. Visualize price range distribution, restaurant density, and seasonal trends to inform marketing strategies and enhance the dining experience in different areas of the city.

1.1 Objectives and Outcomes

Objectives: Data Exploration: Conduct thorough exploration of the Zomato restaurant dataset to understand its structure, features, and potential insights. Insight Generation: Identify key patterns, trends, and relationships within the dataset to uncover valuable insights about the Hyderabad restaurant scene. Visualization: Utilize data visualization techniques to present findings in a clear, concise, and visually appealing manner. Predictive Analysis: Develop predictive models, if applicable, to forecast restaurant ratings or prices based on relevant features. Stakeholder Empowerment: Empower stakeholders, including consumers and businesses, with actionable insights to enhance decision-making processes in the restaurant industry.

Outcomes: Comprehensive Dataset Understanding: Gain a deep understanding of the Zomato restaurant dataset, including its attributes, distributions, and potential biases. Insightful Analysis: Generate actionable insights about customer preferences, pricing trends, popular cuisines, and other relevant aspects of the Hyderabad restaurant scene. Effective Communication: Present findings through compelling data visualizations that effectively communicate key insights to stakeholders. Predictive Models (if applicable): Develop accurate predictive models, if pursued, that can forecast restaurant ratings or prices with a reasonable degree of accuracy. Informed Decision-Making: Enable stakeholders to make informed decisions that optimize the dining experience for consumers and drive business growth for restaurants in Hyderabad

CHAPTER 2
Methodology

3.1 Data collection and Dataset description

The dataset we're using, sourced from GitHub, contains valuable information about online food delivery. It covers details such as delivery personnel, weather, and order specifics, forming the basis of our analysis on delivery times. Each delivery record is unique and provides insights into different parts of the delivery process. This includes details about delivery staff, restaurant and delivery locations, order times, and weather conditions. Additionally, it includes information about delivery vehicles, order types, transportation methods, multiple deliveries, and festivals impacting deliveries. It also specifies the city for each delivery. Importantly, it logs delivery times, reflecting how well the service is working and how satisfied customers are. With this dataset, we aim to thoroughly study how these factors interact, revealing what affects delivery times and service quality in online food delivery.

	links	names	ratings	cuisine	price for one
0	https://www.zomato.com/hyderabad/sahara-bakers...	Sahara Bakers	3.7	Chinese, Bakery, Sichuan, Pizza, Burger	100
1	https://www.zomato.com/hyderabad/kfc-abids/order	KFC	3.9	Burger, Fast Food, Biryani, Desserts, Beverages	100
2	https://www.zomato.com/hyderabad/subbaiah-gari...	Subbaiah Gari Hotel	4.1	South Indian, Andhra, Mithai	100
3	https://www.zomato.com/hyderabad/paradise-biry...	Paradise Biryani	3.9	Biryani, Kebab, Desserts, Beverages	100
4	https://www.zomato.com/hyderabad/pista-house-b...	Pista House Bakery	4.3	Fast Food, Sandwich, Pizza, Burger, Wraps, Rol...	100
...	...	...	...	...	...
652	https://www.zomato.com/hyderabad/dr-cakes-banj...	Dr Cakes	3.2	Bakery, Desserts	350
653	https://www.zomato.com/hyderabad/shahi-naan-am...	Shahi Naan	-	North Indian	350
654	https://www.zomato.com/hyderabad/combosthalam-...	Combosthalam By Phulkaas	3.8	North Indian, Chinese	350
655	https://www.zomato.com/hyderabad/pachadis-by-p...	Pachadis By Phulkaas	-	South Indian	350
656	https://www.zomato.com/hyderabad/tasim-1-himay...	Tasim	3.4	Chinese, Momos	350

657 rows × 5 columns

Figure 3.1: Dataset

3.2 Data cleaning and preprocessing

We delve into the critical aspects of data cleaning and preprocessing, laying the groundwork for the subsequent analysis. Data cleaning is a crucial step in the data analysis process, ensuring that our dataset is accurate, reliable, and free from errors. In this stage, we meticulously sift through the data to identify and rectify any inconsistencies, inaccuracies, or missing values that could potentially skew our analysis.

The data cleaning process for our dataset was carried out in a structured manner to ensure the data's quality and reliability for subsequent analysis. The first step involved removing unnecessary columns. This began with an initial review of the dataset to understand the significance of each column. Any columns deemed irrelevant to the analysis or redundant, that did not affect delivery time, were identified and removed using the Pandas drop function. This streamlined the dataset, making it more focused and manageable.

Next, we addressed missing values, which is crucial to prevent biases or inaccuracies in our analysis. Using Pandas' isnull functions, we identified columns with missing data. Depending on the extent and nature of these missing values, we employed different strategies. For instance, if a column had a few missing values, we filled them with the mean and mode of that column using Pandas' fillna function. However, if a column had a significant portion of missing values and was not critical to the analysis, we considered dropping it entirely using the dropna method. This ensured that our dataset maintained its integrity and represented a complete set of information.

Handling duplicate values was the next step, aimed at ensuring that each record in the dataset was unique. We used Pandas' duplicated function to identify any duplicate rows. Once identified, these duplicates were removed with the drop_duplicates method, ensuring that each entry in the dataset was distinct and avoiding any skewed analysis results from repetitive data.

Subsequently, we addressed data types to ensure each column was in the appropriate format for accurate analysis. We used Pandas' dtypes attribute to review the data types and converted columns as needed with the astype method. For example, columns containing delivery times stored as strings were converted to numerical types or datetime objects to facilitate proper

time series analysis. This step was crucial for enabling accurate computations and comparisons within the dataset.

Outliers, which could distort the analysis, were then handled. We detected outliers using various statistical methods . Values falling outside the range ,depending on the context, we either removed these outliers or capped them to a specified limit using Pandas and NumPy functions, ensuring they did not unduly influence the analysis.

Finally, we addressed zero values in columns where zeros were not logically valid, such as the distance between the restaurant and delivery location. Using Pandas, we identified zero values and replaced them with a more appropriate measure, like the mean of the column, through the replace method. This step was vital to maintain the data's logical consistency and validity.

By following this detailed and systematic approach to data cleaning, we prepared a dataset that was robust, reliable, and ready for in-depth exploratory data analysis. This cleaned dataset laid a solid foundation for uncovering meaningful insights into the factors influencing delivery times in online delivery services

CHAPTER 4

System Architecture and Implementation

4.1 Google Colab

Google Colaboratory, commonly known as Google Colab, is a free online cloud-based Jupyter notebook environment tailored for training machine learn- ing and deep learning models. This article explores the functionalities, benefits, and features of Google Colab, elucidating its significance in the realm of data science and machine learning.

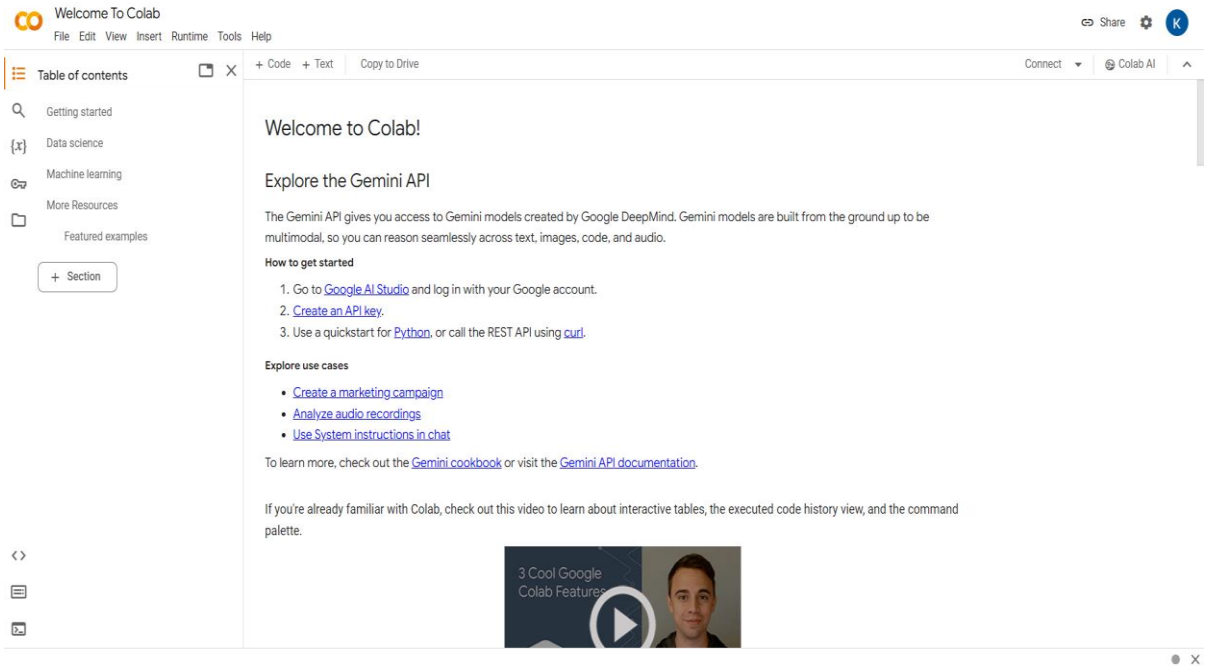


Figure 4.1: Google Colab

4.1.1 What is Google Colab?

Google Colab offers a cloud-based environment accessible via any web browser, eliminating the need for local software installation. Users can leverage its computing resources, including CPUs, GPUs, and TPUs, facilitating efficient model training and execution.

4.2 Benefits of Google Colab

Accessibility: Users can access Google Colab from any location with internet connectivity, streamlining collaboration and workflow.

Power: The platform provides access to potent computing resources like GPUs and TPUs, enabling swift and effective model training.

Collaboration: Google Colab simplifies collaborative efforts by allowing real-time editing and sharing of notebooks among team members.

Education: It serves as an invaluable educational tool for learning about machine learning and data science, offering a plethora of tutorials and resources.

4.2.1 Why Choose Google Colab?

Google Colab stands out as an ideal choice for students, data scientists, researchers, and enthusiasts due to its:

Ease of Use: With no setup requirements, users can swiftly start coding after creating an account.

Affordability: The platform is largely free to use, with paid plans available for more demanding tasks.

Flexibility: Users can seamlessly train models, process data, create visualizations, and collaborate with others, making it a versatile tool for various applications.

4.2.2 Notebook in Google Colab

In Google Colab, a notebook serves as a web-based environment for code creation and execution. Notebooks offer several advantages, including real-time code execution and visualization, support for markdown for documentation,

and collaboration features, making them indispensable for data scientists and machine learning practitioners.

4.2.3 Google Colab Features

Google Colab boasts several features that enhance its usability and effec- tiveness:

Free Access to GPUs and TPUs: Users can leverage powerful computing resources without any additional cost.

Web-based Interface: The intuitive and user-friendly interface eliminates the need for local software installation.

Collaboration Tools: Multiple users can collaborate on the same notebook simultaneously, streamlining teamwork.

Markdown Support: Notebooks support markdown, enabling users to include formatted text, equations, and images alongside their code.

Pre-installed Libraries: Google Colab comes pre-installed with popular libraries and tools for machine learning and deep learning, such as TensorFlow and PyTorch, saving time on setup and configuration.

Google Colab emerges as a versatile and indispensable tool for machine learning and data science tasks, offering accessibility, power, and flexibility. Its user-friendly interface, collaborative features, and integration with powerful computing resources make it an invaluable asset for individuals and teams alike, driving innovation and progress in the field of machine learning and beyond.

4.3 Code Snippets

4.3.1 Importing libraries and Data loading

To begin our project, we first import the necessary libraries for data analysis and visualization. We import pandas as pd and numpy as np for data handling and numerical operations, respectively. We use Matplotlib and Seaborn for data visualization.

Next, we load the dataset into our code using the read csv function from pandas, assuming the dataset is stored in a CSV file named 'zomato.csv'. We assign the loaded dataset to a variable named 'df'.

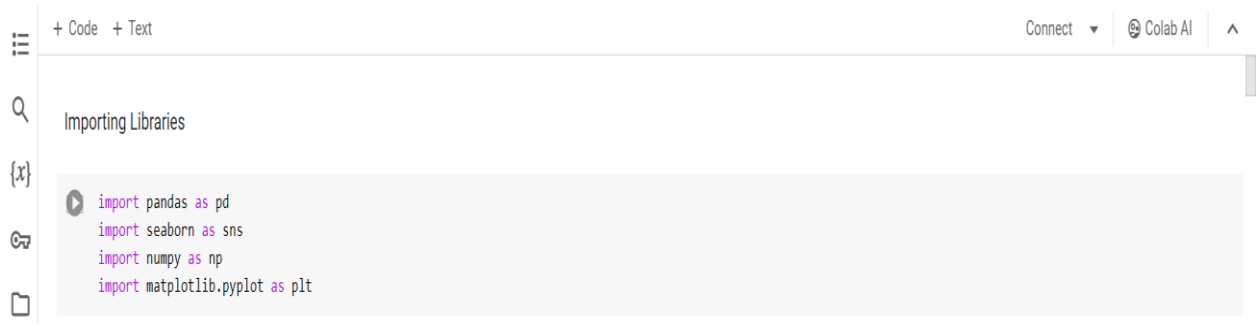


Figure 4.2: importing libraries and Dataset loading

To ensure that the dataset has been loaded successfully, we display the first few rows of the dataset using the head() function. This allows us to inspect the structure and content of the dataset, confirming that it has been imported correctly and is ready for further processing.

Reading CSV File

```
PATH="/content/drive/MyDrive/Colab Notebooks/DAV-PROJECT/HyderabadResturants.csv"
df = pd.read_csv(PATH)
df
```

	links	names	ratings	cuisine	price for one
0	https://www.zomato.com/hyderabad/sahara-bakers...	Sahara Bakers	3.7	Chinese, Bakery, Sichuan, Pizza, Burger	100
1	https://www.zomato.com/hyderabad/kfc-abids/order	KFC	3.9	Burger, Fast Food, Biryani, Desserts, Beverages	100
2	https://www.zomato.com/hyderabad/subbaiah-gari...	Subbaiah Gari Hotel	4.1	South Indian, Andhra, Mithai	100
3	https://www.zomato.com/hyderabad/paradise-biry...	Paradise Biryani	3.9	Biryani, Kebab, Desserts, Beverages	100
4	https://www.zomato.com/hyderabad/pista-house-b...	Pista House Bakery	4.3	Fast Food, Sandwich, Pizza, Burger, Wraps, Rol...	100
...	...	...	...	...	...
652	https://www.zomato.com/hyderabad/dr-cakes-banj...	Dr Cakes	3.2	Bakery, Desserts	350
653	https://www.zomato.com/hyderabad/shahi-naan-am...	Shahi Naan	-	North Indian	350
654	https://www.zomato.com/hyderabad/combosthalam-...	Combosthalam By Phulkaas	3.8	North Indian, Chinese	350
655	https://www.zomato.com/hyderabad/pachadis-by-p...	Pachadis By Phulkaas	-	South Indian	350
656	https://www.zomato.com/hyderabad/tasim-1-himay...	Tasim	3.4	Chinese, Momos	350

657 rows × 5 columns

Figure 4.4: Reading CSV

```
df.describe()  # Get descriptive statistics of the dataframe.
```

	price for one
count	657.000000
mean	169.406393
std	97.178712
min	50.000000
25%	100.000000
50%	150.000000
75%	250.000000
max	400.000000

```
df.info()  # Get information about the dataframe, including data types and missing values.
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 657 entries, 0 to 656
Data columns (total 5 columns):
 #   Column         Non-Null Count  Dtype 
---  ------         --------------  ----- 
 0   links          657 non-null    object
 1   names          657 non-null    object
 2   ratings        657 non-null    object
 3   cuisine        657 non-null    object
 4   price for one  657 non-null    int64 
dtypes: int64(1), object(4)
memory usage: 25.8+ KB
```

```
n = int(input("Enter the number of rows you want to view from the top: "))
df.head(n)
```

```
Enter the number of rows you want to view from the top: 2
```

	links	names	ratings	cuisine	price for one
0	https://www.zomato.com/hyderabad/sahara-bakers...	Sahara Bakers	3.7	Chinese, Bakery, Sichuan, Pizza, Burger	100
1	https://www.zomato.com/hyderabad/kfc-abids/order	KFC	3.9	Burger, Fast Food, Biryani, Desserts, Beverages	100

```
df.shape  # Get the number of rows and columns in the dataframe.
```

```
(657, 5)
```

```
df.columns  # Get the names of the columns in the dataframe.
```

```
Index(['links', 'names', 'ratings', 'cuisine', 'price for one'], dtype='object')
```

```
df.index  # Get the index of the dataframe.
```

```
RangeIndex(start=0, stop=657, step=1)
```

```
df.dtypes  # Get the data types of each column in the dataframe.
```

```
links           object
names           object
ratings         object
cuisine         object
price for one    int64
dtype: object
```

```
df.isnull().sum()
```

```
links           0
names           0
ratings         0
cuisine         0
price for one   0
dtype: int64
```

Figure 4.5: Statistics for Numerical features

This code snippet prints out summary statistics for numerical features in a DataFrame (df). It provides key statistical measures such as count, mean, standard deviation, minimum, maximum, and quartile values for each numerical column in the DataFrame.

4.3.2 Data cleaning and preprocessing

The data cleaning process for our dataset was carried out in a structured manner to ensure quality and reliability for subsequent analysis

1.Viewing Data

Users can specify the number of rows to view from the top of the DataFrame. This feature helps in getting a quick look at the data's initial entries.

```
n = int(input("Enter the number of rows you want to view from the top: "))
df.head(n)
```

```
Enter the number of rows you want to view from the top: 10
```

1 to 10 of 10 entries Filter

index	links	names	ratings	cuisine	price for one
0	https://www.zomato.com/hyderabad/sahara-bakers-rtc-x-roads/order	Sahara Bakers	3.7	Chinese, Bakery, Sichuan, Pizza, Burger	100
1	https://www.zomato.com/hyderabad/kfc-abids/order	KFC	3.9	Burger, Fast Food, Biryani, Desserts, Beverages	100
2	https://www.zomato.com/hyderabad/subbaiah-gari-hotel-malakpet/order	Subbaiah Gari Hotel	4.1	South Indian, Andhra, Mithai	100
3	https://www.zomato.com/hyderabad/paradise-biryani-1-charminar/order	Paradise Biryani	3.9	Biryani, Kebab, Desserts, Beverages	100
4	https://www.zomato.com/hyderabad/pista-house-bakery-ghansi-bazaar/order	Pista House Bakery	4.3	Fast Food, Sandwich, Pizza, Burger, Wraps, Rolls, Salad, Desserts	100
5	https://www.zomato.com/hyderabad/shah-ghouse-hotel-restaurant-tolichowki/order	Shah Ghouse Hotel & Restaurant	4	North Indian, Chinese, Mughlai, Mandi, Sichuan, Shawarma, Seafood, Beverages	100
6	https://www.zomato.com/hyderabad/swagath-hotel-3-charminar/order	Swagath Hotel	4.2	South Indian, Chinese, North Indian	100
7	https://www.zomato.com/hyderabad/just-parantha-banjara-hills/order	Just Parantha	4.2	North Indian	100
8	https://www.zomato.com/hyderabad/mehfil-narayanguda/order	Mehfil	4.1	North Indian, Biryani, Mughlai	100
9	https://www.zomato.com/hyderabad/cream-stone-begum-bazaar/order	Cream Stone	4.3	Ice Cream, Desserts	150

Figure 4.6: Viewing Data

2. Addressing Data Types:

We reviewed the data types using Pandas' dtypes attribute and converted columns as needed with the astype method. For instance, delivery times stored as strings were converted to numerical types or datetime objects, enabling accurate computations and comparisons.

```
df.dtypes  # Get the data types of each column in the dataframe.
```

```
links           object
names           object
ratings         object
cuisine         object
price for one    int64
dtype: object
```

Figure 4.7:Addressing Data Types

5.Basic Data Exploration

The script provided insights into the structure of the DataFrame:

- **Shape**: Number of rows and columns.
- **Columns**: Names of the columns.
- **Index**: Indexes of the DataFrame.
- **Data Types**: Types of data in each column.
- **Missing Values**: Count of missing values in each column.

```
df.describe()  # Get descriptive statistics of the dataframe.
```

	price for one
count	657.000000
mean	169.406393
std	97.178712
min	50.000000
25%	100.000000
50%	150.000000
75%	250.000000
max	400.000000

```
df.info()  # Get information about the dataframe, including data types and missing values.
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 657 entries, 0 to 656
Data columns (total 5 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   links          657 non-null    object
 1   names          657 non-null    object
 2   ratings        657 non-null    object
 3   cuisine        657 non-null    object
 4   price for one  657 non-null    int64
dtypes: int64(1), object(4)
memory usage: 25.8+ KB
```

```
n = int(input("Enter the number of rows you want to view from the top: "))
df.head(n)
```

```
Enter the number of rows you want to view from the top: 2
```

	links	names	ratings	cuisine	price for one
0	https://www.zomato.com/hyderabad/sahara-bakers...	Sahara Bakers	3.7	Chinese, Bakery, Sichuan, Pizza, Burger	100
1	https://www.zomato.com/hyderabad/kfc-abids/order	KFC	3.9	Burger, Fast Food, Biryani, Desserts, Beverages	100

```
df.shape  # Get the number of rows and columns in the dataframe.
```

```
(657, 5)
```

```
df.columns  # Get the names of the columns in the dataframe.
```

```
Index(['links', 'names', 'ratings', 'cuisine', 'price for one'], dtype='object')
```

```
df.index  # Get the index of the dataframe.
```

```
RangeIndex(start=0, stop=657, step=1)
```

```
df.dtypes  # Get the data types of each column in the dataframe.
```

```
links           object
names           object
ratings         object
cuisine         object
price for one    int64
dtype: object
```

```
df.isnull().sum()
```

```
links           0
names           0
ratings         0
cuisine         0
price for one   0
dtype: int64
```

Figure 4.8: Basic Data Exploration

6.File Management

The script included functionality to list all files in a specified directory and filter out only the CSV files. This helps in identifying all relevant data files in the working directory.

```
import glob
# Define the directory path
directory = "/content/drive/MyDrive/Colab Notebooks/DAV-PROJECT/"

# Use glob to list all files in the directory
files = glob.glob(directory + '*')

# Print each file
for file in files:
    print(file)
```

```
/content/drive/MyDrive/Colab Notebooks/DAV-PROJECT/HyderabadResturants.csv
```

```
[ ]  csv_files = glob.glob(directory + '*.csv')

     # Print each CSV file
     for csv_file in csv_files:
         print(csv_file)
```

```
/content/drive/MyDrive/Colab Notebooks/DAV-PROJECT/HyderabadResturants.csv
```

Figure 4.9:File Mangement

Data Manipulation

Sorting

The DataFrame was sorted by the ratings column to rank the restaurant's based on their Ratings

```
sample_df = df.sort_values(by=['price for one'])
sample_df
```

	links	names	ratings	cuisine	price for one
166	https://www.zomato.com/hyderabad/slay-coffee-a...	SLAY Coffee	4.2	Beverages, Coffee	50
51	https://www.zomato.com/hyderabad/tibbs-frankie...	Tibb's Frankie	3.9	Rolls	50
50	https://www.zomato.com/hyderabad/1989-pizza-th...	1989 Pizza & Thick Shake's	4.1	Fast Food, Pizza, Pasta, Desserts, Beverages, ...	50
49	https://www.zomato.com/hyderabad/riyan-hotel-n...	Riyan Hotel	3.7	Mughlai, Biryani	50
48	https://www.zomato.com/hyderabad/kamat-hotel-n...	Kamat Hotel	4.2	North Indian, South Indian, Chinese	50
...	...	...	...	...	...
35	https://www.zomato.com/hyderabad/rice-bowl-dil...	Rice Bowl	3.7	North Indian	400
34	https://www.zomato.com/hyderabad/lassi-shop-1-...	Lassi Shop	4.2	Beverages, Ice Cream, Desserts, Shake	400
33	https://www.zomato.com/hyderabad/cakezone-hima...	CakeZone	4.1	Bakery, Desserts, Ice Cream, Pancake	400
41	https://www.zomato.com/hyderabad/haldirams-abi...	Haldiram's	4.2	North Indian, South Indian, Street Food, Fast ...	400
42	https://www.zomato.com/hyderabad/makers-of-mil...	Makers Of Milkshakes	4.3	Shake, Ice Cream, Desserts	400

557 rows × 5 columns

Figure 4.10: Sorting

Filtering

Specific rows were filtered based on criteria such as cuisine type or exact values in ratings and prices. This helped in focusing on particular subsets of data.

```
o = df.loc[1:10, ['names', 'ratings', 'price for one']]
o
```

	names	ratings	price for one
1	KFC	3.9	100
2	Subbaiah Gari Hotel	4.1	100
3	Paradise Biryani	3.9	100
4	Pista House Bakery	4.3	100
5	Shah Ghouse Hotel & Restaurant	4	100
6	Swagath Hotel	4.2	100
7	Just Parantha	4.2	100
8	Mehfil	4.1	100
9	Cream Stone	4.3	150
10	MS Bakery & Sweets	4.3	150

```
o = df.iloc[1:20, [1, 2, 4]]
o
```

	names	ratings	price for one
1	KFC	3.9	100
2	Subbaiah Gari Hotel	4.1	100
3	Paradise Biryani	3.9	100
4	Pista House Bakery	4.3	100
5	Shah Ghouse Hotel & Restaurant	4	100
6	Swagath Hotel	4.2	100
7	Just Parantha	4.2	100
8	Mehfil	4.1	100
9	Cream Stone	4.3	150
10	MS Bakery & Sweets	4.3	150
11	Iqbal Cafe	4.4	150
12	Mahalaxmi Tiffin Centre	4.1	150
13	Al- Naseer Tahari Centre	4.3	150
14	Pizza Hut	3.7	150
15	Andhra Gunpowder	4.3	150
16	Burger King	4.2	150
17	Kinara Grand	4	150
18	Meridian Restaurant	4.1	150
19	KS Bakers	4.1	150

```
q=df.iloc[[0,1,2,3]]
q
```

	links	names	ratings	cuisine	price for one
0	https://www.zomato.com/hyderabad/sahara-bakers...	Sahara Bakers	3.7	Chinese, Bakery, Sichuan, Pizza, Burger	100
1	https://www.zomato.com/hyderabad/kfc-abids/order	KFC	3.9	Burger, Fast Food, Biryani, Desserts, Beverages	100
2	https://www.zomato.com/hyderabad/subbaiah-gari...	Subbaiah Gari Hotel	4.1	South Indian, Andhra, Mithai	100
3	https://www.zomato.com/hyderabad/paradise-biry...	Paradise Biryani	3.9	Biryani, Kebab, Desserts, Beverages	100

```
s=df.iloc[[0,1,2],[1,2]]
s
```

	names	ratings
0	Sahara Bakers	3.7
1	KFC	3.9
2	Subbaiah Gari Hotel	4.1

```
z=df.loc[2]
z
```

```
links          https://www.zomato.com/hyderabad/subbaiah-gari...
names                                        Subbaiah Gari Hotel
ratings                                                      4.1
cuisine                                South Indian, Andhra, Mithai
price for one                                                100
Name: 2, dtype: object
```

```
l=df.loc[[37,73],['names','ratings']]
l
```

	names	ratings
37	Behrouz Biryani	3.9
73	Papaji-Da-Dhaba	4

```
loc=df.loc[2:4]
loc
```

	links	names	ratings	cuisine	price for one
2	https://www.zomato.com/hyderabad/subbaiah-gari...	Subbaiah Gari Hotel	4.1	South Indian, Andhra, Mithai	100
3	https://www.zomato.com/hyderabad/paradise-biry...	Paradise Biryani	3.9	Biryani, Kebab, Desserts, Beverages	100
4	https://www.zomato.com/hyderabad/pista-house-b...	Pista House Bakery	4.3	Fast Food, Sandwich, Pizza, Burger, Wraps, Rol...	100

```
iloc=df.iloc[2:4]
iloc
```

	links	names	ratings	cuisine	price for one
2	https://www.zomato.com/hyderabad/subbaiah-gari...	Subbaiah Gari Hotel	4.1	South Indian, Andhra, Mithai	100
3	https://www.zomato.com/hyderabad/paradise-biry...	Paradise Biryani	3.9	Biryani, Kebab, Desserts, Beverages	100

Figure 4.11: Filtering

Value Counts

The script counted the occurrences of each unique value in the ratings column, providing insight into the distribution of ratings.

```
a=df.ratings.value_counts()
print(a)
plt.bar(a.index,a)
```

```
ratings
4       103
4.1      77
4.3      66
-        63
3.9      62
4.2      62
3.7      44
3.8      42
3.5      31
3.6      28
4.4      21
3.4      14
3.2       8
4.5       8
3.1       7
3.3       5
New       4
4.6       4
3         2
2.9       2
2.7       1
4.7       1
2.8       1
2.6       1
Name: count, dtype: int64
<BarContainer object of 24 artists>
```

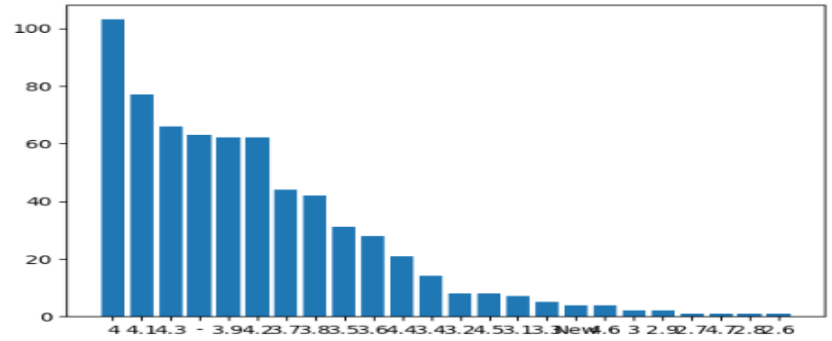


Figure 4.12: Value Counts

Converting Currency

A conversion function was implemented to convert prices from Indian rupees to US dollars, adding a new dimension to the price data and making it more comprehensible for international audiences.

```
def convert(price):
    price = price / 74  # Convert to USD
    price = price + 1.28  # Add additional cost
    return '$ {:.2f}'.format(price)  # Format the price as a string with 2 decimal places and prepend '$'
```

```
df['Price_in_USD'] = df['price for one'].apply(lambda x: convert(x))
df.head()
```

	links	names	ratings	cuisine	price for one	Price_in_USD
0	https://www.zomato.com/hyderabad/sahara-bakers...	Sahara Bakers	3.7	Chinese, Bakery, Sichuan, Pizza, Burger	100	$ 2.63
1	https://www.zomato.com/hyderabad/kfc-abids/order	KFC	3.9	Burger, Fast Food, Biryani, Desserts, Beverages	100	$ 2.63
2	https://www.zomato.com/hyderabad/subbaiah-gari...	Subbaiah Gari Hotel	4.1	South Indian, Andhra, Mithai	100	$ 2.63
3	https://www.zomato.com/hyderabad/paradise-biry...	Paradise Biryani	3.9	Biryani, Kebab, Desserts, Beverages	100	$ 2.63
4	https://www.zomato.com/hyderabad/pista-house-b...	Pista House Bakery	4.3	Fast Food, Sandwich, Pizza, Burger, Wraps, Rol...	100	$ 2.63

Figure 4.13: Converting Currency

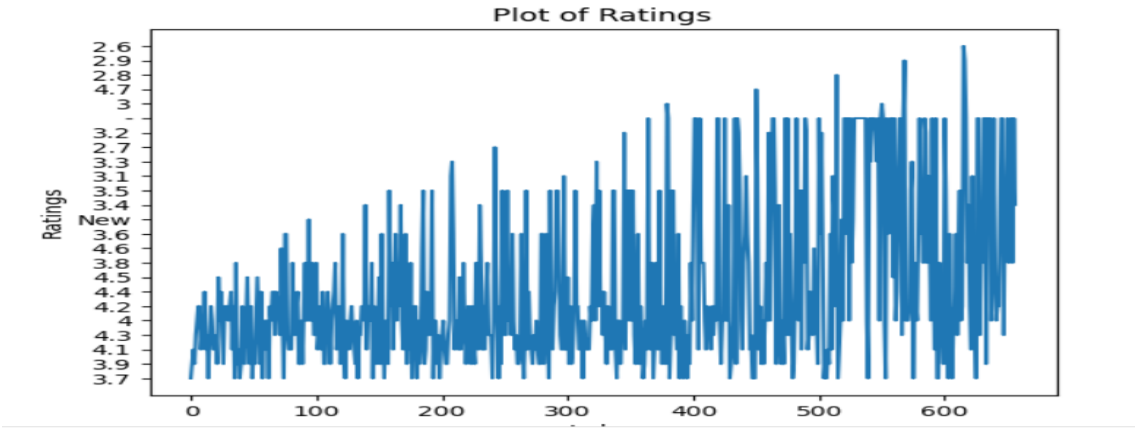
4.3.3 Exploratory Data Analysis

Exploratory Data Analysis (EDA) is a crucial preliminary step in data analysis, focusing on understanding the dataset's structure, identifying patterns, and uncovering relationships between variables. It involves visualizing data, summarizing key features, and detecting potential anomalies. EDA serves as a foundation for further analysis and model building.

1. **Scatter Plot Analysis:**

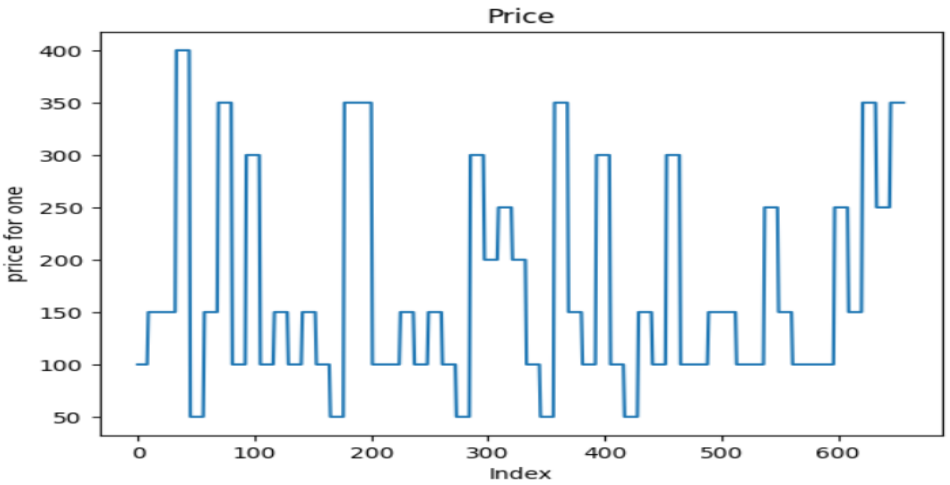
Scatter plots are effective graphical tools for exploring relationships between two continuous variables. In the context of the Hyderabad restaurants dataset, scatter plots can be used to visually assess potential associations between different features and the target variable. For instance, we can explore how various restaurant attributes impact customer ratings or prices.

```
# import matplotlib for plotting
import matplotlib.pyplot as plt
%matplotlib inline
# Assuming 'ratings' is a column in your DataFrame named df
plt.plot(df['ratings'])
# Add labels and title
plt.xlabel('Index')
plt.ylabel('Ratings')
plt.title('Plot of Ratings')
# Show the plot
plt.show()
```

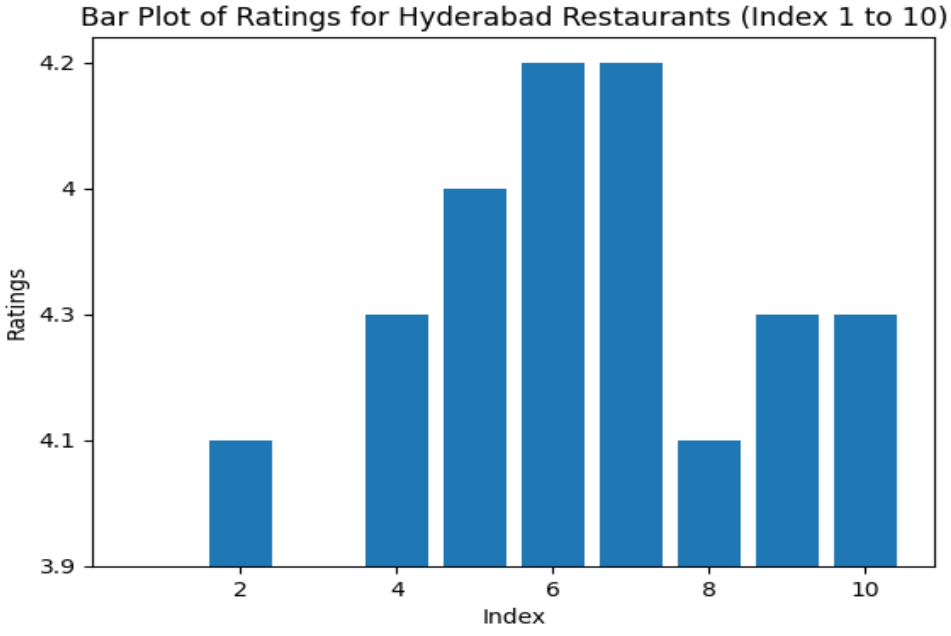


```
# Assuming 'ratings' is a column in your DataFrame named df
plt.plot(df['price for one'])
# Add labels and title
plt.xlabel('Index')
plt.ylabel('price for one')
plt.title('Price')

# Show the plot
plt.show()
```




```
import pandas as pd
import matplotlib.pyplot as plt
# Load the Hyderabad restaurants CSV file into a DataFrame
data = pd.read_csv("/content/drive/MyDrive/Colab Notebooks/DAV-PROJECT/HyderabadResturants.csv")
# Select the range of rows (e.g., from index 1 to 10)
data_subset = data.iloc[1:11]  # Rows from index 1 to 10 (inclusive)
# Plotting the ratings column of the subset
plt.bar(data_subset.index, data_subset['ratings'])
# Adding labels and title
plt.xlabel('Index')
plt.ylabel('Ratings')
plt.title('Bar Plot of Ratings for Hyderabad Restaurants (Index 1 to 10)')
# Show the plot
plt.show()
```

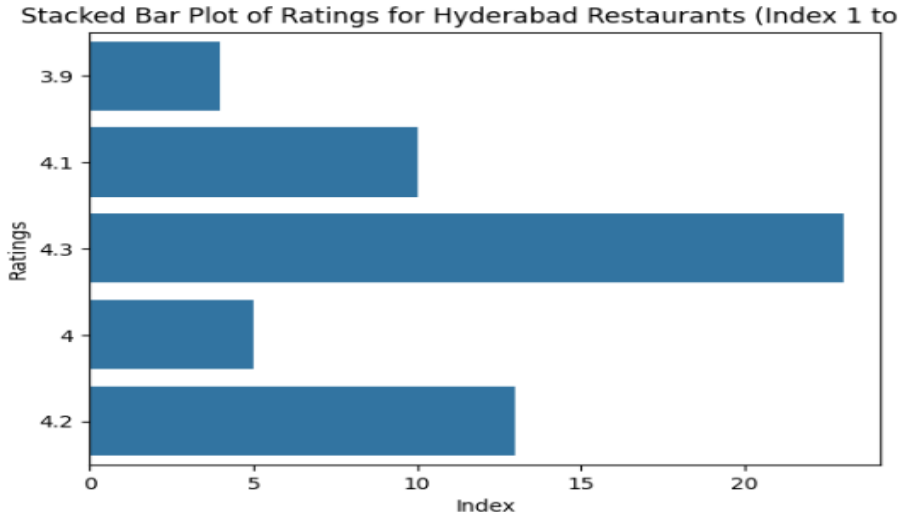


```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
# Load the Hyderabad restaurants CSV file into a DataFrame
data = pd.read_csv("/content/drive/MyDrive/Colab Notebooks/DAV-PROJECT/HyderabadResturants.csv")
# Select the range of rows (e.g., from index 1 to 10)
data_subset = data.iloc[1:11]  # Rows from index 1 to 10 (inclusive)
# Plotting using Seaborn
sns.barplot(x=data_subset.index, y='ratings', data=data_subset, estimator=sum, ci=None)
# Adding labels and title
plt.xlabel('Index')
plt.ylabel('Ratings')
plt.title('Stacked Bar Plot of Ratings for Hyderabad Restaurants (Index 1 to 10)')
# Show the plot
plt.show()
```

```
<ipython-input-68-ad9bb1f50d32>:12: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar=None` for the same effect.

  sns.barplot(x=data_subset.index, y='ratings', data=data_subset, estimator=sum, ci=None)
```




```
# title
plt.title(' ratings vs no of indexes')

# xlabel
plt.xlabel('ratings')

# ylabel
plt.ylabel('no of indexes')

# plot histogram
plt.hist(data_BM['ratings'], bins=20, color='lightblue');
```

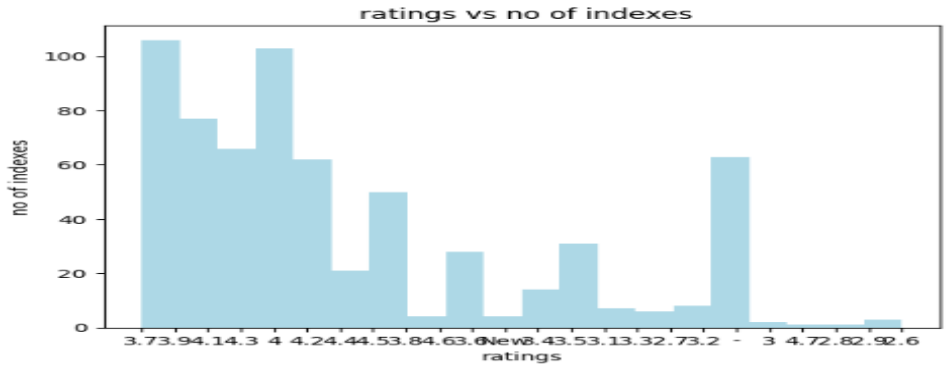


Figure 4.14: Scatter Plot

2. Sea Born Analysis

a) Sea born bar plot

The script creates a bar plot of the `ratings` column for a subset of DataFrame using Seaborn (`sns.barplot`).

Figure 4.15: Seaborn bar plot

b) Sea born histplot

The script creates a histogram of the `price for one` column using Seaborn (`sns.histplot`).

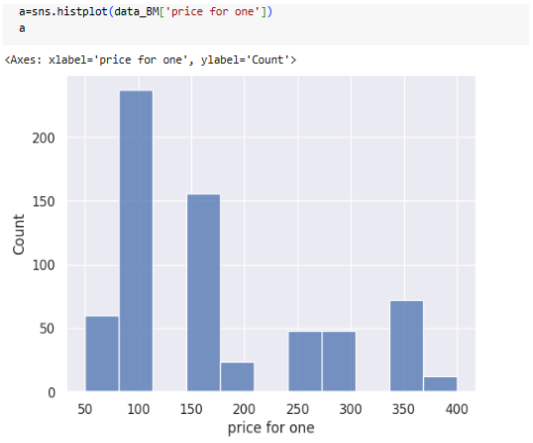


Figure 4.16: Seaborn hist plot

C) Sea born lineplot

The script creates a line plot of the `ratings` and `price for one` columns using Seaborn (`sns.lineplot`).

```
import seaborn as sns
sns.set()
sns.set(style="darkgrid")
```

```
# line plot using relplot
sns.lineplot(x="ratings", y="price for one",data=data_BM[:50]);
```

Figure 4.17: Seaborn line plot

D) Sea born Box Plot

The script creates a boxplot of the `price for one` column using Seaborn (`sns.boxplot`).

```
sns.boxplot(data_BM['price for one'], orient='vertical')
```

```
<Axes: ylabel='price for one'>
```

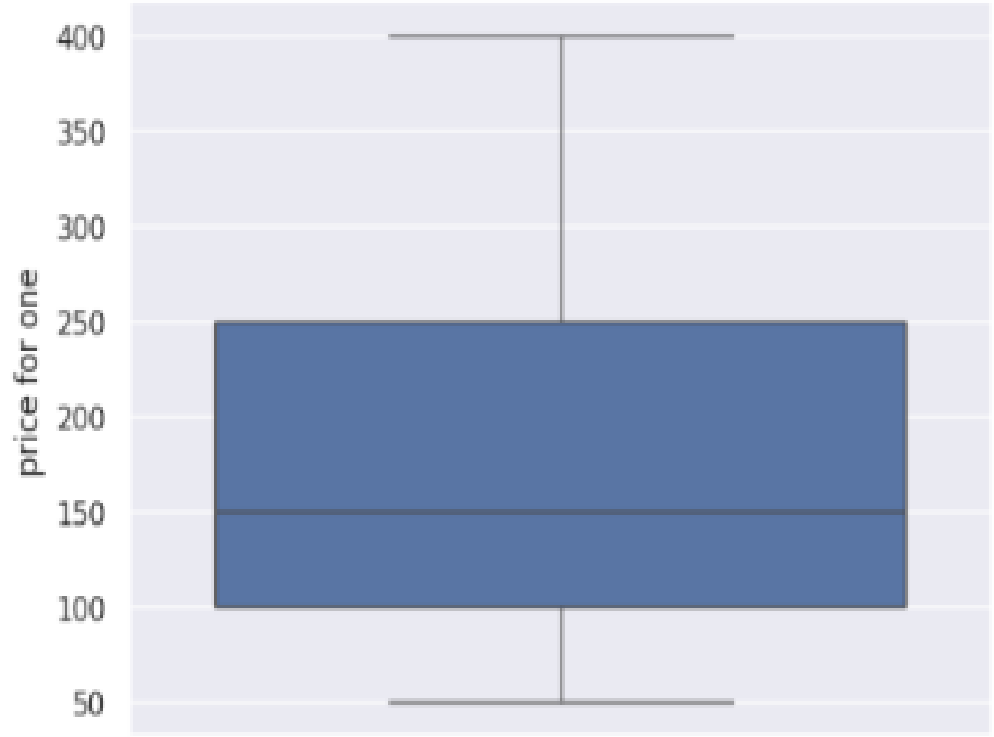


Figure 4.18: Seaborn Box plot

e) Sea born Rel Plot

The script creates a relplot of the `ratings` and `price for one` columns, with the `names` column as the hue, using Seaborn (`sns.relplot`)

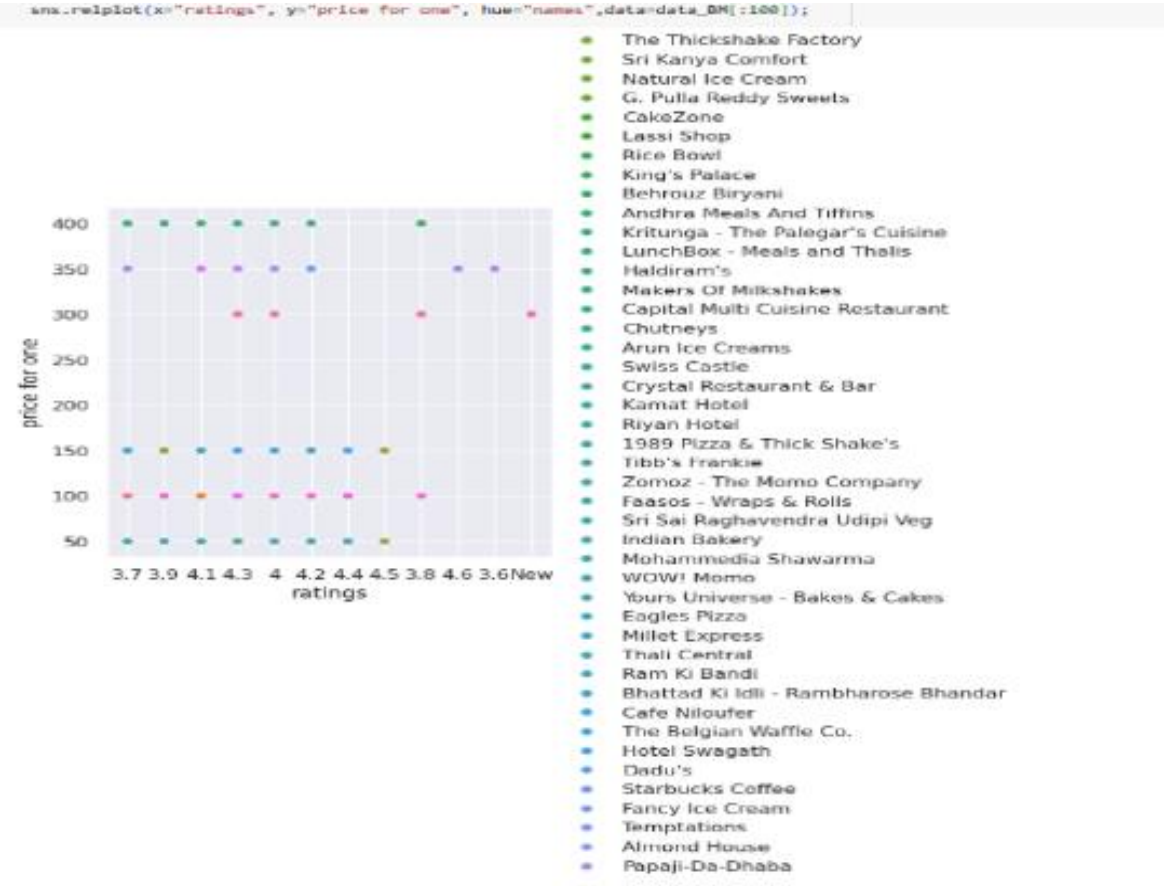


Figure 4.19: Seaborn Rel plot

F) Sea Born pair plot

The script creates a pairplot with kernel density estimates along the diagonal using Seaborn (`sns.pairplot`).

```
import pandas as pd
import seaborn as sns

# Define the path to the Hyderabad restaurants CSV file
PATH = "/content/drive/MyDrive/Colab Notebooks/DAV-PROJECT/HyderabadResturants.csv"

# Load the Hyderabad restaurants CSV file into a DataFrame using Pandas
hyderabad_restaurants = pd.read_csv(PATH)

# Now, you can use Seaborn for visualization
# For example, to create a pairplot with kernel density estimates along the diagonal:
a = sns.pairplot(hyderabad_restaurants, diag_kind='kde')
a
```

```
<seaborn.axisgrid.PairGrid at 0x7f5984221a20>
```

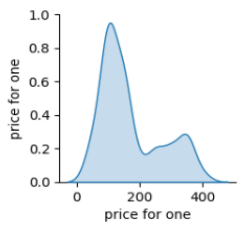


Figure 4.20: Seaborn pair plot

4.4 Conclusion

This project provided a detailed exploration and analysis of the Hyderabad restaurants dataset, uncovering valuable insights into various aspects of restaurant operations and customer preferences. By leveraging scatter plots and feature engineering, the analysis highlighted key relationships within the data, offering practical recommendations for improving restaurant performance and customer satisfaction. These findings lay a solid foundation for further in-depth analysis and strategic decision-making in the restaurant industry.

References

1. **Pandas Documentation:**
   - McKinney, W. (2010). Data Structures for Statistical Computing in Python. *Proceedings of the 9th Python in Science Conference*, 51-56.
   - Pandas Documentation
2. **Matplotlib Documentation:**
   - Hunter, J. D. (2007). Matplotlib: A 2D Graphics Environment. *Computing in Science & Engineering*, 9(3), 90-95.
   - Matplotlib Documentation
3. **Seaborn Documentation:**
   - Waskom, M. L. (2021). Seaborn: Statistical Data Visualization. *Journal of Open Source Software*, 6(60), 3021.
   - Seaborn Documentation

4.Exploratory Data Analysis (EDA):

- Behrens, J. T. (1997). Principles and Procedures of Exploratory Data Analysis. *Psychological Methods*, 2(2), 131-160.
- Tukey, J. W. (1977). *Exploratory Data Analysis*. Addison-Wesley.

4. **Visualization Techniques:**
   - Few, S. (2012). *Show Me the Numbers: Designing Tables and Graphs to Enlighten*. Analytics Press.
   - Knaflic, C. N. (2015). *Storytelling with Data: A Data Visualization Guide for Business Professionals*. Wiley.
5. **Python for Data Analysis:**
   - McKinney, W. (2017). *Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython*. O'Reilly Media.

6. VanderPlas, J. (Python Data Science Handbook):

- Essential Tools for Working with Data. O'Reilly Media