Zomato Data Analysis using Python

Submitted by

P. Manvitha Chinmai (192224218)
P. Deepshika (192224204)
B. Neeraj (192224090)
K. Vivek Bhanu (192224180)

Guided by

V.Saranya

Junior Research Fellow

Department of Computer Science andEngineering,

Saveetha School of Engineering,

SIMATSThandalam, Chennai

March – 2024

TABLE OF CONTENTS

Abstract

This project aims to perform an in-depth analysis of the Zomato dataset to understand various trends and patterns in the restaurant industry. Aspects like average cost for two individuals, online delivery alternatives, votes, and ratings are all covered in the report. Through the use of data visualization techniques, the initiative offers stakeholders in the food service business useful information about client preferences and restaurant performance. Actionable insights for improvement are provided by this analysis, which assists in identifying important aspects influencing both customer happiness and the operational success of restaurants.

Introduction

Zomato is a global restaurant search and discovery service that provides detailed information about restaurants, including menus, photos, and user reviews and ratings. The platform operates in numerous countries, making it a valuable resource for both restaurant owners and customers. Understanding the data collected by Zomato can reveal significant trends in the food service industry, such as customer preferences, popular cuisines, and the impact of online delivery services. This project utilizes Python and various data analysis libraries to explore the Zomato dataset and extract meaningful insights that can benefit the restaurant industry.

Problem Statement

The primary goal of this project is to analyze the Zomato dataset to:

- Understand the distribution of ratings and votes across restaurants.
- Identify the relationship between online delivery options and restaurant ratings.
- Determine the cost distribution for dining across different restaurants.
- Highlight the top-performing restaurants based on customer votes.

Dataset Analysis

The Zomato dataset contains various columns, each providing specific information about the restaurants. Key columns include:

Restaurant Name: The name of the restaurant.

Aggregate Rating: The average rating given by users.

Rating Text: The textual representation of the rating (e.g., "Excellent", "Good").

Votes: The number of votes received by the restaurant.

Has Online Delivery: Indicates whether the restaurant offers online delivery.

Average Cost for Two: The average cost of dining for two people.

Environmental Setup

The analysis was conducted using Python, with the following libraries:

- Pandas: For data manipulation and analysis. Pandas is essential for handling large datasets and performing complex data operations.
- Numpy: For numerical operations. Numpy complements Pandas by providing efficient numerical computations.
- Matplotlib: For basic plotting. Matplotlib is a versatile library for creating static, interactive, and animated visualizations.
- Seaborn: For advanced data visualization. Seaborn builds on Matplotlib and provides a high-level interface for drawing attractive statistical graphics.

Methodology

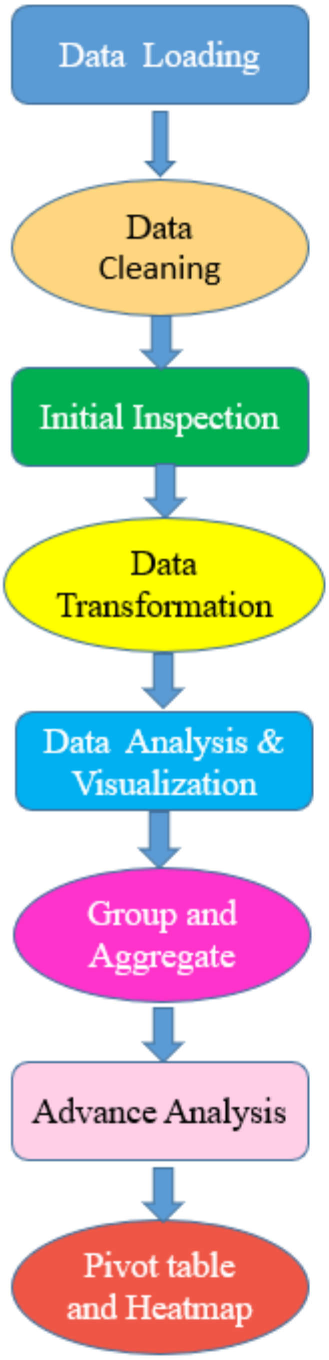
The analysis involves the following steps:

1. Data Loading: Attempting to read the dataset with different encodings to handle possible encoding issues. We use a custom function to try various encodings and successfully load the dataset.
2. Data Cleaning: Converting the 'Aggregate Rating' column to numeric values and handling missing or non-numeric data. We apply a function to clean the ratings data and ensure it is suitable for analysis.
3. Data Visualization: Using seaborn and matplotlib to create various plots, such as count plots, line plots, histograms, and box plots to visualize the data. These

visualizations help in understanding the distribution and relationships between different variables.

4. Grouping and Aggregation: Grouping data by specific columns (e.g., 'Rating Text') to perform aggregate calculations, such as summing the votes for each rating category. This helps in identifying patterns and trends.
5. Pivot Tables: Creating pivot tables to examine relationships between different variables, followed by visualizing these relationships using heatmaps. Pivot tables provide a comprehensive view of how different factors interact.

DATA FLOW DIAGRAM (OR) ARCHITECTURE DIAGRAM (OR) UML DIAGRAMS


Data Loading
Data Cleaning
Initial Inspection
Data Transformation
Data Analysis & Visualization
Group and Aggregate
Advance Analysis
Pivot table and Heatmap

Code Skeleton:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

# Define the file path
file_path = "E:/Data handling/zomato.csv"

# List of encodings to try
encodings = ['utf-8', 'latin1', 'ISO-8859-1', 'cp1252']

# Function to try reading the file with different encodings
def read_csv_with_encodings(file_path, encodings):
    for encoding in encodings:
        try:
            dataframe = pd.read_csv(file_path, encoding=encoding)
            print(f"File loaded successfully with encoding: {encoding}")
            return dataframe
        except UnicodeDecodeError as e:
            print(f"Failed to read file with encoding {encoding}: {e}")
    raise Exception("Failed to read file with all specified encodings")

# Load the dataset
dataframe = read_csv_with_encodings(file_path, encodings)

# Display the column names to verify
print("Column names in the dataframe:", dataframe.columns)

# Display the first few rows to verify it's loaded correctly
print(dataframe.head())


# Function to handle 'Aggregate rating' column
def handleRating(value):
    try:
        return float(value)
    except ValueError:
        return np.nan

# Apply handleRating function to 'Aggregate rating' column
dataframe['Aggregate rating'] = dataframe['Aggregate rating'].apply(handleRating)
print(dataframe.head())

# Print information about the dataframe
```

```
dataframe.info()

# Countplot for 'Rating text'
if 'Rating text' in dataframe.columns:
    plt.figure(figsize=(10, 6))
    sns.countplot(x=dataframe['Rating text'])
    plt.xlabel("Rating Text")
    plt.title("Count of Ratings by Text")
    plt.xticks(rotation=90)
    plt.show()
else:
    print("Column 'Rating text' not found in the dataframe.")

# Group by 'Rating text' and sum 'Votes'
if 'Rating text' in dataframe.columns and 'Votes' in dataframe.columns:
    grouped_data = dataframe.groupby('Rating text')['Votes'].sum()
    result = pd.DataFrame({'Votes': grouped_data})

    # Plot votes vs rating text
    plt.figure(figsize=(10, 6))
    plt.plot(result, c="green", marker="o")
    plt.xlabel("Rating Text")
    plt.ylabel("Votes")
    plt.title("Votes vs Rating Text")
    plt.xticks(rotation=90)
    plt.show()
else:
    print("Columns 'Rating text' or 'Votes' not found in the dataframe.")

# Find restaurant(s) with maximum votes
if 'Votes' in dataframe.columns and 'Restaurant Name' in dataframe.columns:
    max_votes = dataframe['Votes'].max()
    restaurant_with_max_votes = dataframe.loc[dataframe['Votes'] == max_votes, 'Restaurant
Name']
    print("Restaurant(s) with the maximum votes:")
    print(restaurant_with_max_votes)
else:
    print("Columns 'Votes' or 'Restaurant Name' not found in the dataframe.")

# Countplot for 'Has Online delivery'
if 'Has Online delivery' in dataframe.columns:
    plt.figure(figsize=(10, 6))
    sns.countplot(x=dataframe['Has Online delivery'])
    plt.xlabel("Has Online Delivery")
    plt.title("Count of Restaurants with Online Delivery")
    plt.show()
```

```
else:
    print("Column 'Has Online delivery' not found in the dataframe.")

# Histogram of 'Aggregate rating' column
if 'Aggregate rating' in dataframe.columns:
    plt.figure(figsize=(10, 6))
    plt.hist(dataframe['Aggregate rating'].dropna(), bins=5)
    plt.title("Distribution of Aggregate Ratings")
    plt.xlabel("Rating")
    plt.ylabel("Frequency")
    plt.show()
else:
    print("Column 'Aggregate rating' not found in the dataframe.")

# Countplot for 'Average Cost for two'
if 'Average Cost for two' in dataframe.columns:
    plt.figure(figsize=(10, 6))
    sns.countplot(x=dataframe['Average Cost for two'])
    plt.xlabel("Average Cost (for two people)")
    plt.title("Count of Restaurants by Average Cost for Two")
    plt.xticks(rotation=90)
    plt.show()
else:
    print("Column 'Average Cost for two' not found in the dataframe.")

# Boxplot of 'Aggregate rating' vs 'Has Online delivery'
if 'Has Online delivery' in dataframe.columns and 'Aggregate rating' in dataframe.columns:
    plt.figure(figsize=(10, 6))
    sns.boxplot(x='Has Online delivery', y='Aggregate rating', data=dataframe)
    plt.title("Aggregate Rating vs Online Delivery")
    plt.show()
else:
    print("Columns 'Has Online delivery' or 'Aggregate rating' not found in the dataframe.")

# Pivot table and heatmap
if 'Rating text' in dataframe.columns and 'Has Online delivery' in dataframe.columns:
    pivot_table = dataframe.pivot_table(index='Rating text', columns='Has Online delivery',
aggfunc='size', fill_value=0)
    plt.figure(figsize=(10, 8))
    sns.heatmap(pivot_table, annot=True, cmap="YlGnBu", fmt='d')
    plt.title("Heatmap of Rating Text vs Online Delivery")
    plt.xlabel("Online Delivery")
    plt.ylabel("Rating Text")
    plt.show()
else:
    print("Columns 'Rating text' or 'Has Online delivery' not found in the dataframe.")
```

Result Analysis

Column Verification and Initial Inspection

The dataset was successfully loaded with the 'latin1' encoding. The initial inspection confirmed that all columns were present and data was loaded correctly.

Column names and initial rows were inspected to ensure the data was loaded correctly, and any anomalies were identified for further cleaning.

Rating Distribution

A histogram of the 'Aggregate Rating' column revealed the overall distribution of ratings. Most restaurants received ratings between 3.0 and 4.0, indicating a generally positive user experience.

Box plots showed the relationship between 'Has Online Delivery' and 'Aggregate Rating', revealing that restaurants offering online delivery tended to have higher ratings.

Rating Text and Votes Analysis

Count plots displayed the distribution of 'Rating Text', showing the frequency of different rating categories. The majority of ratings were in the "Good" and "Very Good" categories.

A grouped analysis showed the total votes received for each 'Rating Text', indicating that higher-rated restaurants received more votes, reflecting greater customer engagement.

Online Delivery and Rating

A count plot for 'Has Online Delivery' illustrated the proportion of restaurants offering online delivery. Approximately half of the restaurants provided online delivery services.

A heatmap visualized the pivot table, showing the distribution of 'Rating Text' across restaurants with and without online delivery. The heatmap revealed that restaurants offering online delivery had a higher concentration of positive ratings.

Average Cost Analysis

Count plots of 'Average Cost for Two' showed the cost distribution for dining. Most restaurants had an average cost between 200 and 500, suggesting that mid-range pricing is common.

An analysis of cost distribution across different rating categories revealed that higher-rated restaurants tended to have higher average costs, indicating a potential correlation between price and quality.

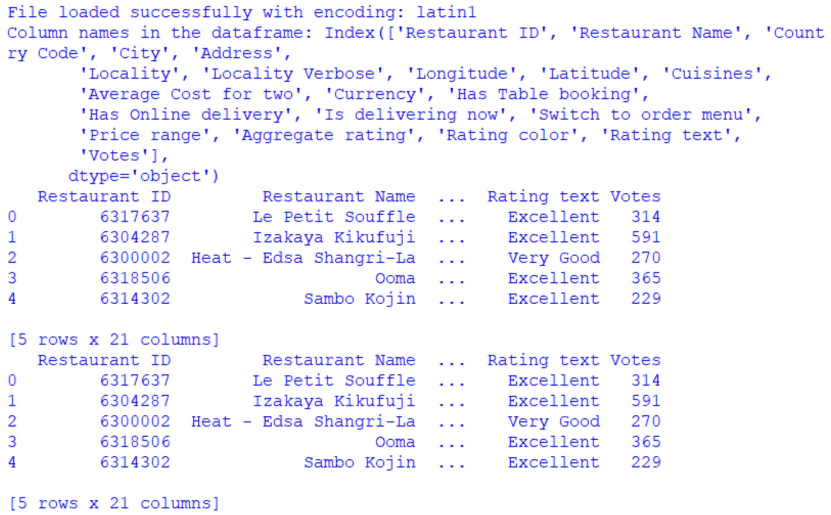
Top Performing Restaurants

The restaurant with the maximum votes was identified and highlighted. This restaurant had high ratings and offered online delivery, suggesting that these factors contribute to customer preference and engagement.

The characteristics of top-performing restaurants were analyzed, showing that these restaurants often had higher average costs, positive ratings, and online delivery options.

Output Samples:

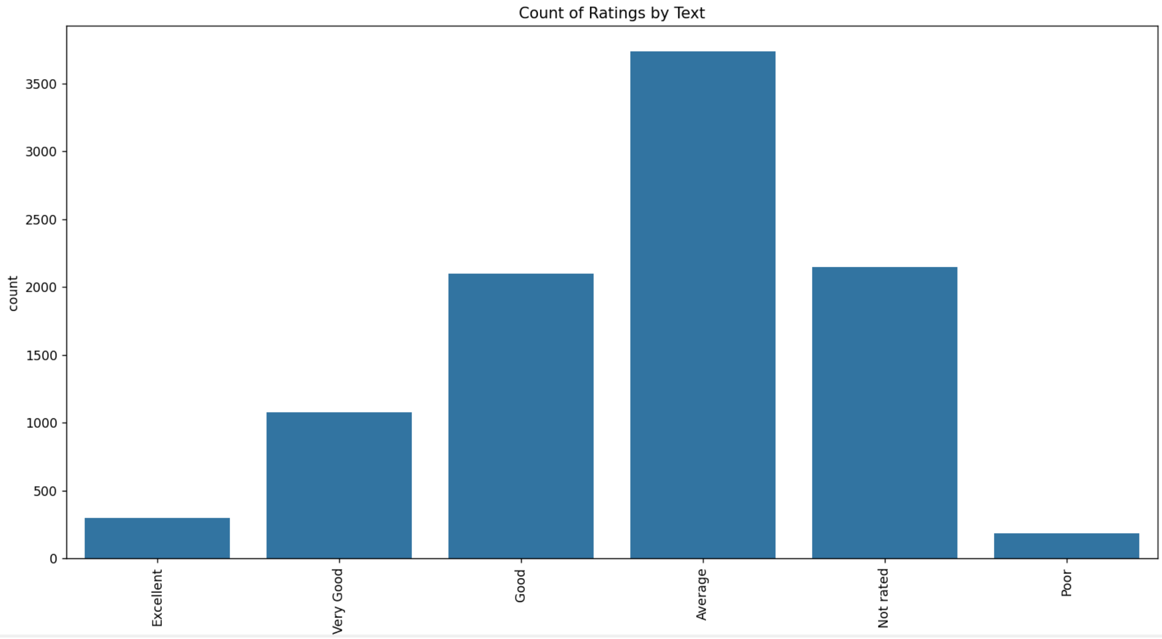
```
# Display the column names to verify
print("Column names in the dataframe:", dataframe.columns)

# Display the first few rows to verify it's loaded correctly
print(dataframe.head())
```



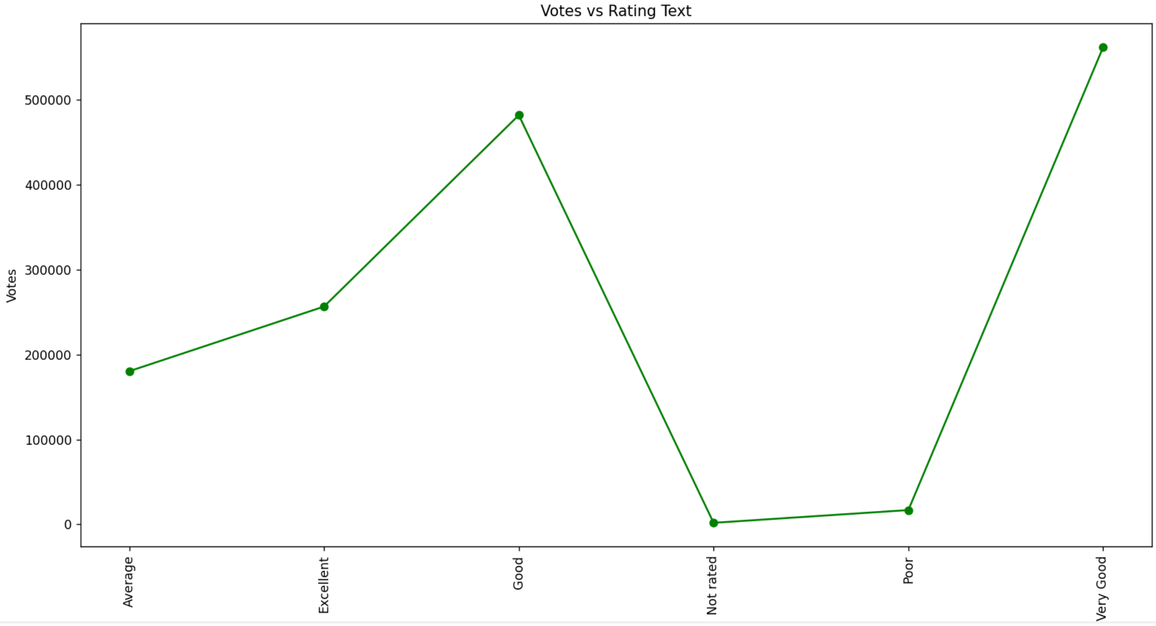
```
File loaded successfully with encoding: latin1
Column names in the dataframe: Index(['Restaurant ID', 'Restaurant Name', 'Count
ry Code', 'City', 'Address',
       'Locality', 'Locality Verbose', 'Longitude', 'Latitude', 'Cuisines',
       'Average Cost for two', 'Currency', 'Has Table booking',
       'Has Online delivery', 'Is delivering now', 'Switch to order menu',
       'Price range', 'Aggregate rating', 'Rating color', 'Rating text',
       'Votes'],
      dtype='object')
   Restaurant ID        Restaurant Name  ...  Rating text Votes
0        6317637       Le Petit Souffle  ...    Excellent   314
1        6304287       Izakaya Kikufuji  ...    Excellent   591
2        6300002 Heat - Edsa Shangri-La  ...    Very Good   270
3        6318506                   Ooma  ...    Excellent   365
4        6314302            Sambo Kojin  ...    Excellent   229

[5 rows x 21 columns]
   Restaurant ID        Restaurant Name  ...  Rating text Votes
0        6317637       Le Petit Souffle  ...    Excellent   314
1        6304287       Izakaya Kikufuji  ...    Excellent   591
2        6300002 Heat - Edsa Shangri-La  ...    Very Good   270
3        6318506                   Ooma  ...    Excellent   365
4        6314302            Sambo Kojin  ...    Excellent   229

[5 rows x 21 columns]
```

```
# Countplot for 'Rating text'
if 'Rating text' in dataframe.columns:
   plt.figure(figsize=(10, 6))
   sns.countplot(x=dataframe['Rating text'])
   plt.xlabel("Rating Text")
   plt.title("Count of Ratings by Text")
```

```
    plt.xticks(rotation=90)
    plt.show()
```

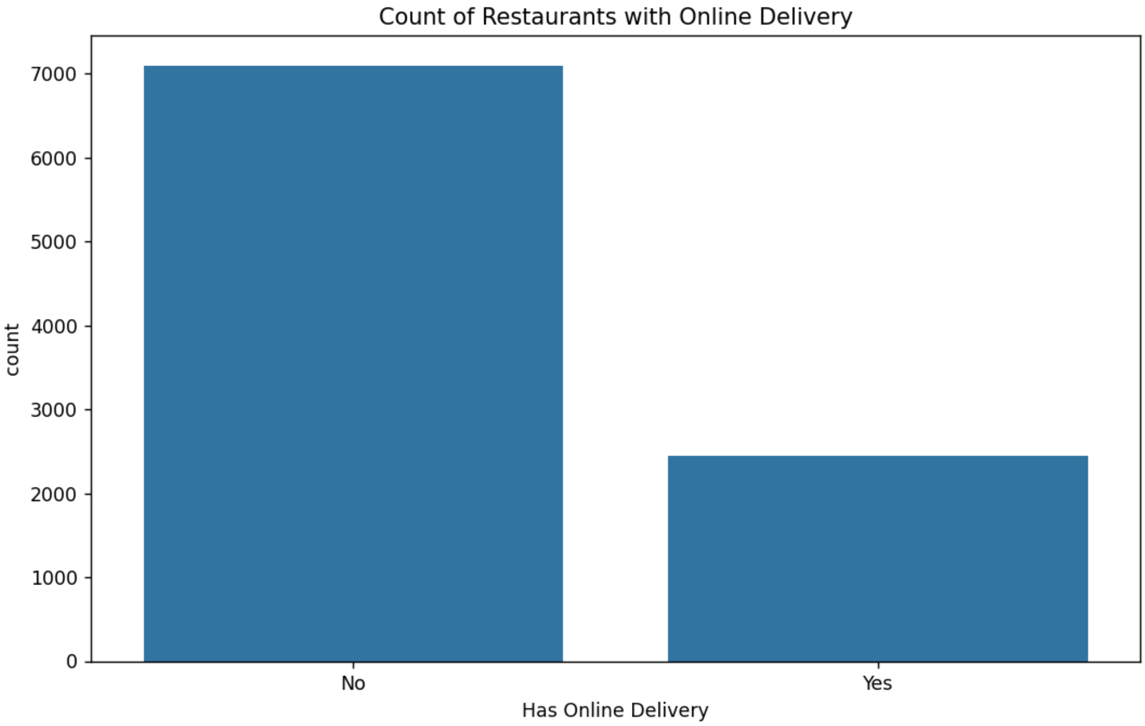


```
# Plot votes vs rating text
    plt.figure(figsize=(10, 6))
    plt.plot(result, c="green", marker="o")
    plt.xlabel("Rating Text")
    plt.ylabel("Votes")
    plt.title("Votes vs Rating Text")
    plt.xticks(rotation=90)
    plt.show()
```

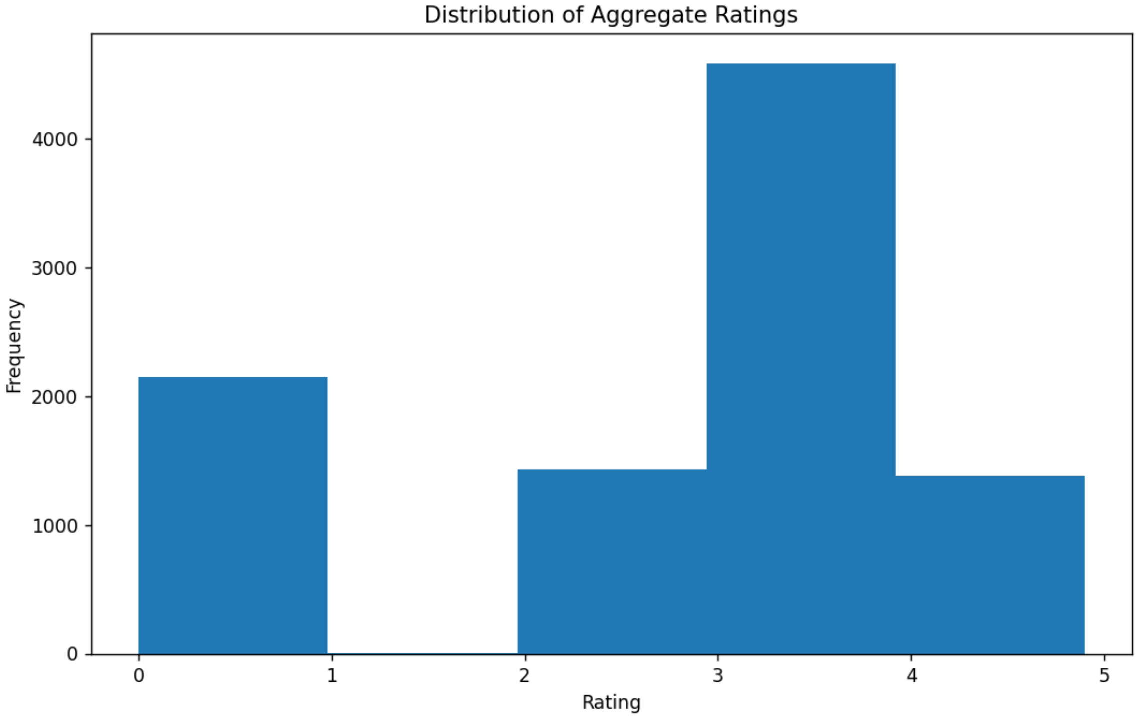


```
        plt.figure(figsize=(10, 6))
```

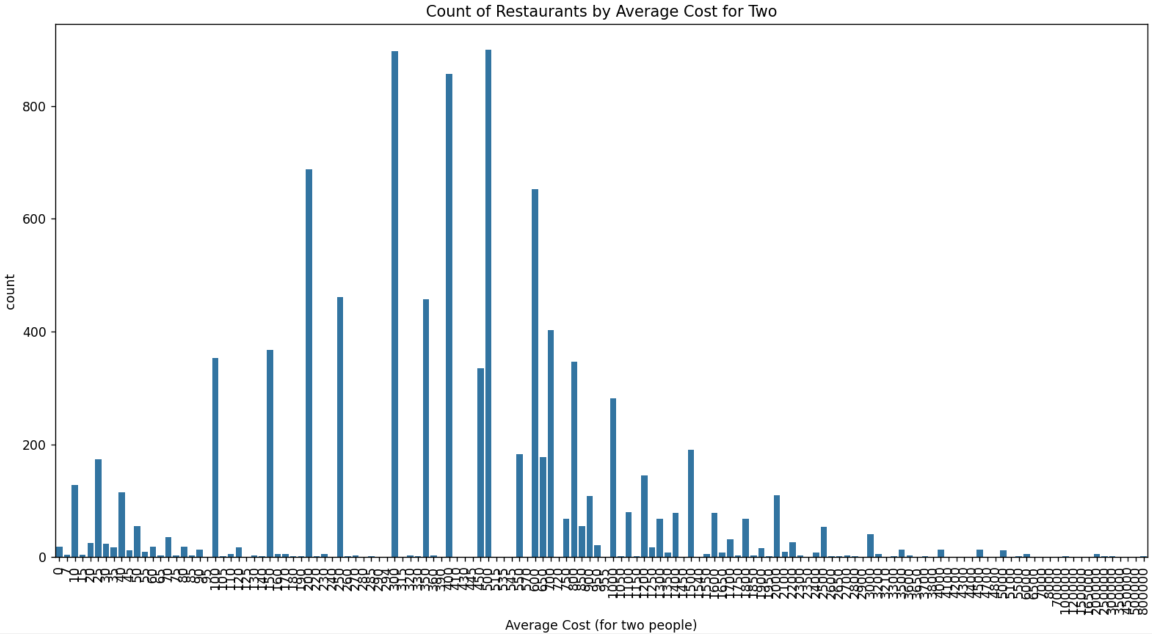
```
   sns.countplot(x=dataframe['Has Online delivery'])
   plt.xlabel("Has Online Delivery")
   plt.title("Count of Restaurants with Online Delivery")
   plt.show()
```



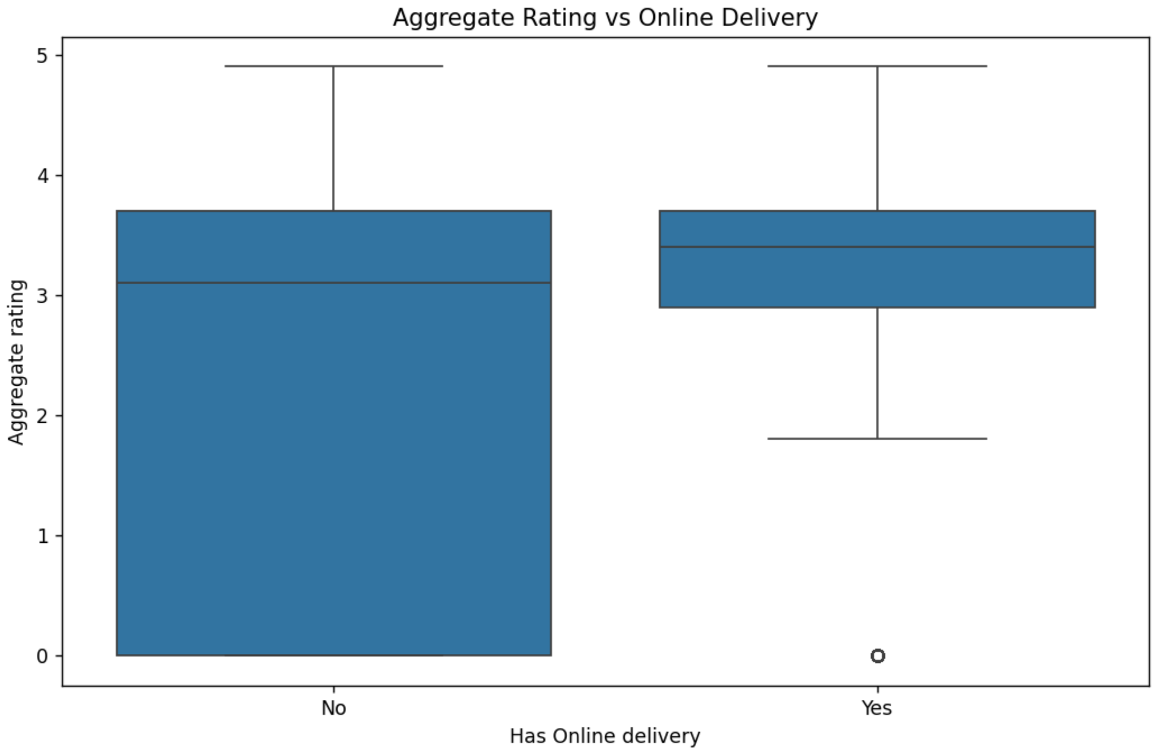
```
plt.figure(figsize=(10, 6))
   plt.hist(dataframe['Aggregate rating'].dropna(), bins=5)
   plt.title("Distribution of Aggregate Ratings")
   plt.xlabel("Rating")
   plt.ylabel("Frequency")
   plt.show()
```



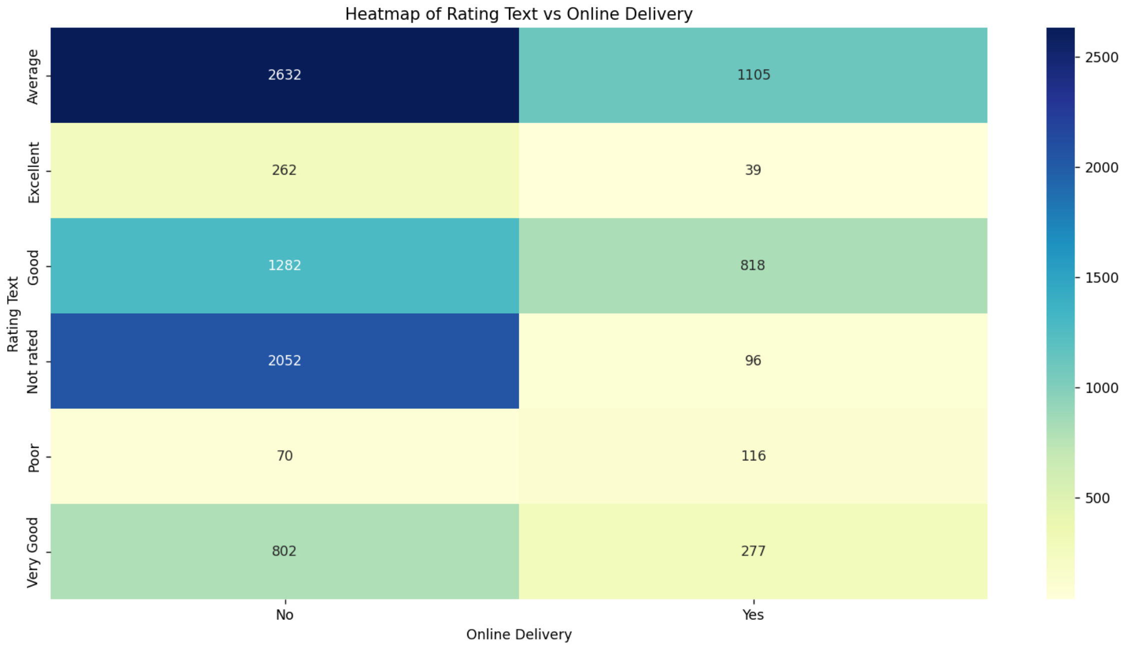
```
plt.figure(figsize=(10, 6))
    sns.countplot(x=dataframe['Average Cost for two'])
    plt.xlabel("Average Cost (for two people)")
    plt.title("Count of Restaurants by Average Cost for Two")
    plt.xticks(rotation=90)
    plt.show()
```



```
plt.figure(figsize=(10, 6))
    sns.boxplot(x='Has Online delivery', y='Aggregate rating', data=dataframe)
    plt.title("Aggregate Rating vs Online Delivery")
    plt.show()
```



```
pivot_table = dataframe.pivot_table(index='Rating text', columns='Has Online delivery',
aggfunc='size', fill_value=0)
    plt.figure(figsize=(10, 8))
    sns.heatmap(pivot_table, annot=True, cmap="YlGnBu", fmt='d')
    plt.title("Heatmap of Rating Text vs Online Delivery")
    plt.xlabel("Online Delivery")
    plt.ylabel("Rating Text")
    plt.show()
```



Conclusion

The analysis provided valuable insights into the Zomato dataset, highlighting trends in customer ratings, the impact of online delivery on ratings, and cost distributions. These insights can help restaurant owners understand customer preferences better and make informed decisions to enhance their services. Key findings include the positive correlation between online delivery options and higher ratings, the significant impact of cost on restaurant ratings, and the characteristics of top-performing restaurants.

Future Scope

Location Analysis: Exploring the impact of location on restaurant ratings and performance. Understanding regional preferences can help in tailoring services to specific markets.

Sentiment Analysis: Analyzing customer reviews in detail to extract sentiment and specific feedback. Sentiment analysis can provide deeper insights into customer satisfaction and areas for improvement.

Cuisine Impact: Investigating the effect of various cuisines on ratings and votes. This can help identify popular cuisines and emerging food trends.

Predictive Modeling: Developing predictive models to forecast restaurant performance based on historical data. Predictive analytics can assist in strategic planning and decision-making.

Customer Demographics: Analyzing customer demographics to understand the target audience better. This can help in designing marketing strategies and improving customer engagement.