zomato

Indian Restaurants Exploratory Data Analysis (EDA)

By - Drishti Khosla

About Zomato

Zomato is one of India’s leading online food delivery and restaurant discovery platforms, founded in 2008. It allows users to explore restaurants, read and write reviews, view menus, and order food online. The platform collects extensive data on restaurants, including cuisine types, locations, ratings, prices, and user feedback. This rich dataset makes Zomato ideal for performing Exploratory Data Analysis (EDA) to uncover insights about customer preferences, restaurant trends, pricing patterns, and factors influencing ratings.

Exploratory Data Analysis (EDA)

Exploratory Data Analysis (EDA) is the process of examining and visualizing data to understand its main characteristics, detect patterns, and uncover relationships between variables. It helps identify missing values, outliers, and data inconsistencies while providing insights that guide further analysis or modeling. EDA is a crucial first step in any data analysis project as it allows analysts to make data-driven decisions and build a strong foundation for accurate conclusions.

Project Overview

This project aims to analyze the Zomato restaurant dataset using Exploratory Data Analysis (EDA) to understand the key factors influencing restaurant success. It explores various aspects such as location, cuisine type, price range, ratings, and online order availability. The analysis includes data cleaning, visualization, and interpretation to uncover meaningful trends and patterns. By examining relationships between different features, the project helps identify what drives higher ratings and customer satisfaction. The insights derived can benefit both restaurant owners and customers in making informed decisions.

Project Objectives

- To perform Exploratory Data Analysis (EDA) on the Zomato dataset to understand its structure and key features.
- To identify the major factors that influence restaurant ratings and overall success.
- To analyze the impact of location, cuisine type, and price range on customer preferences.
- To study the relationship between online ordering, table booking, and restaurant ratings.
- To visualize trends and patterns using graphs and charts for better understanding.
- To derive actionable insights that can help restaurant owners and customers make informed decisions.

Tools and Libraries used

- Python
- Pandas
- Seaborn
- Matplotlib
- Google Collab
- Canva (for presentation)

Dataset Description

Column Name	**Description**
Restaurant Name	Name of the restaurant.
Location	Area or city of the restaurant.
Cuisines	Type of cuisines served.
Price Range	Cost category (1 = Low, 4 = High).
Average Cost for Two	Approximate cost for two people.
Rating	Average user rating.
Votes	Number of customer reviews or votes.
Online Order	Whether online ordering is available (Yes/No).
Table Booking	Whether table booking is available (Yes/No).
Restaurant Type	Type of service (e.g., Dine-in, Delivery, Café).
City	City where the restaurant is located.

Data Cleaning

1. Checking Dataset Information:

Displayed the basic structure of the dataset using df.info() and df.shape() to understand the number of rows, columns, and data types.

```
df.shape
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 211944 entries, 0 to 211943
Data columns (total 26 columns):
 #   Column                 Non-Null Count   Dtype
---  ------                 --------------   -----
 0   res_id                 211944 non-null  int64
 1   name                   211944 non-null  object
 2   establishment          211944 non-null  object
 3   url                    211944 non-null  object
 4   address                211810 non-null  object
 5   city                   211944 non-null  object
 6   city_id                211944 non-null  int64
 7   locality               211944 non-null  object
 8   latitude               211944 non-null  float64
 9   longitude              211944 non-null  float64
 10  zipcode                48757 non-null   object
 11  country_id             211944 non-null  int64
 12  locality_verbose       211944 non-null  object
 13  cuisines               210553 non-null  object
 14  timings                208070 non-null  object
 15  average_cost_for_two   211944 non-null  int64
 16  price_range            211944 non-null  int64
 17  currency               211944 non-null  object
 18  highlights             211944 non-null  object
 19  aggregate_rating       211944 non-null  float64
 20  rating_text            211944 non-null  object
 21  votes                  211944 non-null  int64
 22  photo_count            211944 non-null  int64
 23  opentable_support      211896 non-null  float64
 24  delivery               211944 non-null  int64
 25  takeaway               211944 non-null  int64
dtypes: float64(4), int64(9), object(13)
memory usage: 42.0+ MB
```

2. Displayed summary statistics using df.describe() to understand data distribution2.

```
df.describe()
```

	res_id	city_id	latitude	longitude	country_id	average_cost_for_two	price_range	aggregate_rating	votes	photo_count	opentable_support	delivery	takeaway
count	2.119440e+05	211944.000000	211944.000000	211944.000000	211944.0	211944.000000	211944.000000	211944.000000	211944.000000	211944.000000	211896.0	211944.000000	211944.0
mean	1.349411e+07	4746.785434	21.499758	77.615276	1.0	595.812229	1.882535	3.395937	378.001864	256.971224	0.0	-0.255907	-1.0
std	7.883722e+06	5568.766386	22.781331	7.500104	0.0	606.239363	0.892989	1.283642	925.333370	867.668940	0.0	0.964172	0.0
min	5.000000e+01	1.000000	0.000000	0.000000	1.0	0.000000	1.000000	0.000000	-18.000000	0.000000	0.0	-1.000000	-1.0
25%	3.301027e+06	11.000000	15.496071	74.877961	1.0	250.000000	1.000000	3.300000	16.000000	3.000000	0.0	-1.000000	-1.0
50%	1.869573e+07	34.000000	22.514494	77.425971	1.0	400.000000	2.000000	3.800000	100.000000	18.000000	0.0	-1.000000	-1.0
75%	1.881297e+07	11306.000000	26.841667	80.219323	1.0	700.000000	2.000000	4.100000	362.000000	128.000000	0.0	1.000000	-1.0
max	1.915979e+07	11354.000000	10000.000000	91.832769	1.0	30000.000000	4.000000	4.900000	42539.000000	17702.000000	0.0	1.000000	-1.0

3. Identified the number of missing values in each column using df.isnull().sum()

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

	0
res_id	0
name	0
establishment	0
url	0
address	134
city	0
city_id	0
locality	0
latitude	0
longitude	0
zipcode	163187
country_id	0
locality_verbose	0
cuisines	1391
timings	3874
average_cost_for_two	0
price_range	0
currency	0
highlights	0
aggregate_rating	0
rating_text	0
votes	0
photo_count	0
opentable_support	48
delivery	0
takeaway	0

dtype: int64

4. Displayed the cleaned dataset after removing duplicates, showing 60,417 records remaining to confirm that only unique entries are retained.

```
df.drop_duplicates(inplace = True)
print("After removing duplicates:", df.shape)
df
```

After removing duplicates: (60417, 26)

	res_id	name	establishment	url	address	city	city_id	locality	latitude	longitude
0	3400299	Bikanervala	['Quick Bites']	https://www.zomato.com/agra/bikanervala-khanda...	Kalyani Point, Near Tulsi Cinema, Bypass Road,...	Agra	34	Khandari	27.211450	78.002381
1	3400005	Mama Chicken Mama Franky House	['Quick Bites']	https://www.zomato.com/agra/mama-chicken-mama-...	Main Market, Sadar Bazaar, Agra Cantt, Agra	Agra	34	Agra Cantt	27.160569	78.011583
2	3401013	Bhagat Halwai	['Quick Bites']	https://www.zomato.com/agra/bhagat-halwai-2-sh...	62/1, Near Easy Day, West Shivaji Nagar, Goalp...	Agra	34	Shahganj	27.182938	77.979684
3	3400290	Bhagat Halwai	['Quick Bites']	https://www.zomato.com/agra/bhagat-halwai-civi...	Near Anjana Cinema, Nehru Nagar, Civil Lines, ...	Agra	34	Civil Lines	27.205668	78.004799
4	3401744	The Salt Cafe Kitchen & Bar	['Casual Dining']	https://www.zomato.com/agra/the-salt-cafe-kitc...	1C,3rd Floor, Fatehabad Road, Tajganj, Agra	Agra	34	Tajganj	27.157709	78.052421
...	...	...	...	...	...	...	...	...	...	...
211882	19142822	Shree Janta Ice Cream	['Dessert Parlour']	https://www.zomato.com/vadodara/shree-janta-ic...	Ground Floor, 5 Ronak Plaza, Tulsidham Char Ra...	Vadodara	32	Manjalpur	22.270516	73.196408
211925	18984164	The Grand Thakar	['Casual Dining']	https://www.zomato.com/vadodara/the-grand-thak...	3rd Floor, Shreem Shalini Mall, Opposite Conqu...	Vadodara	32	Alkapuri	22.310563	73.171163
211926	18019952	Geeta lodge	['Casual Dining']	https://www.zomato.com/vadodara/geeta-lodge-al...	Shop 11, Ground Floor, Atlantis K-10, Tower A,...	Vadodara	32	Alkapuri	22.317731	73.168107
211940	3200996	Raju Omlet	['Quick Bites']	https://www.zomato.com/vadodara/raju-omlet-kar...	Mahalaxmi Apartment, Opposite B O B, Karoli Ba...	Vadodara	32	Karelibaug	22.322455	73.197203
211942	3201138	Subway	['Quick Bites']	https://www.zomato.com/vadodara/subway-1-akota...	G-2, Vedant Platina, Near Cosmos, Akota, Vadodara	Vadodara	32	Akota	22.270027	73.143068

	...	price_range	currency	highlights	aggregate_rating	rating_text	votes	photo_count	opentable_support	delivery	takeaway
0	...	2	Rs.	['Lunch', 'Takeaway Available', 'Credit Card',...	4.4	Very Good	814	154	0.0	-1	-1
1	...	2	Rs.	['Delivery', 'No Alcohol Available', 'Dinner',...	4.4	Very Good	1203	161	0.0	-1	-1
2	...	1	Rs.	['No Alcohol Available', 'Dinner', 'Takeaway A...	4.2	Very Good	801	107	0.0	1	-1
3	...	1	Rs.	['Takeaway Available', 'Credit Card', 'Lunch',...	4.3	Very Good	693	157	0.0	1	-1
4	...	3	Rs.	['Lunch', 'Serves Alcohol', 'Cash', 'Credit Ca...	4.9	Excellent	470	291	0.0	1	-1
...	...	...	...	...	...	...	...	...	...	...	...
211882	...	1	Rs.	['Cash', 'Takeaway Available', 'Delivery', 'In...	2.9	Average	4	1	0.0	1	-1
211925	...	2	Rs.	['Dinner', 'Cash', 'Debit Card', 'Lunch', 'Tak...	4.0	Very Good	111	38	0.0	-1	-1
211926	...	1	Rs.	['Dinner', 'Cash', 'Credit Card', 'Lunch', 'Ta...	3.9	Good	207	14	0.0	-1	-1
211940	...	1	Rs.	['Dinner', 'Cash', 'Takeaway Available', 'Debi...	4.1	Very Good	187	40	0.0	1	-1
211942	...	2	Rs.	['Dinner', 'Delivery', 'Credit Card', 'Lunch',...	3.7	Good	128	34	0.0	1	-1

60417 rows × 26 columns

5. We handled missing data by filling numerical columns with their mean values and categorical columns with their mode (most frequent value). A warning appeared because using inplace=True on a sliced column can create a temporary copy, but the values were still filled correctly. Finally, columns with more than 50% missing data were dropped to improve dataset quality.

```
num_cols = df.select_dtypes(include = ['float64', 'int64']).columns
df[num_cols] = df[num_cols].fillna(df[num_cols].mean())
```

```
cat_cols = df.select_dtypes(include = ['object']).columns
for col in cat_cols:
  df[col].fillna(df[col].mode()[0], inplace = True)
```

```
/tmp/ipython-input-3583101405.py:3: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through chained assignment using an inplace method.
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are setting values always behaves as a copy.

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col] = df[col].method(value) instead, to perform the operation inplace on the original object.


  df[col].fillna(df[col].mode()[0], inplace = True)
```

```
threshold = len(df) * 0.5
df.dropna(thresh = threshold, axis = 1, inplace = True)
```

6. After handling missing values, we checked the dataset using df.isnull().sum(). The result showed 0 missing values in all columns, which confirms that our data cleaning process was successful. This ensures the dataset is complete, consistent, and ready for further analysis.

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

	0
res_id	0
name	0
establishment	0
url	0
address	0
city	0
city_id	0
locality	0
latitude	0
longitude	0
zipcode	0
country_id	0
locality_verbose	0
cuisines	0
timings	0
average_cost_for_two	0
price_range	0
currency	0
highlights	0
aggregate_rating	0
rating_text	0
votes	0
photo_count	0
opentable_support	0
delivery	0
takeaway	0

dtype: int64

Data Visualisation

Location Analysis

This graph shows the distribution of restaurant ratings across the top 5 cities in the dataset. Each city's rating spread is represented with overlapping histograms, helping us compare how ratings vary by location. Most restaurants in all cities fall between 3.0 and 4.5 ratings, indicating generally good customer satisfaction. The visualization highlights how some cities have more restaurants and a slightly higher concentration of ratings in certain ranges.

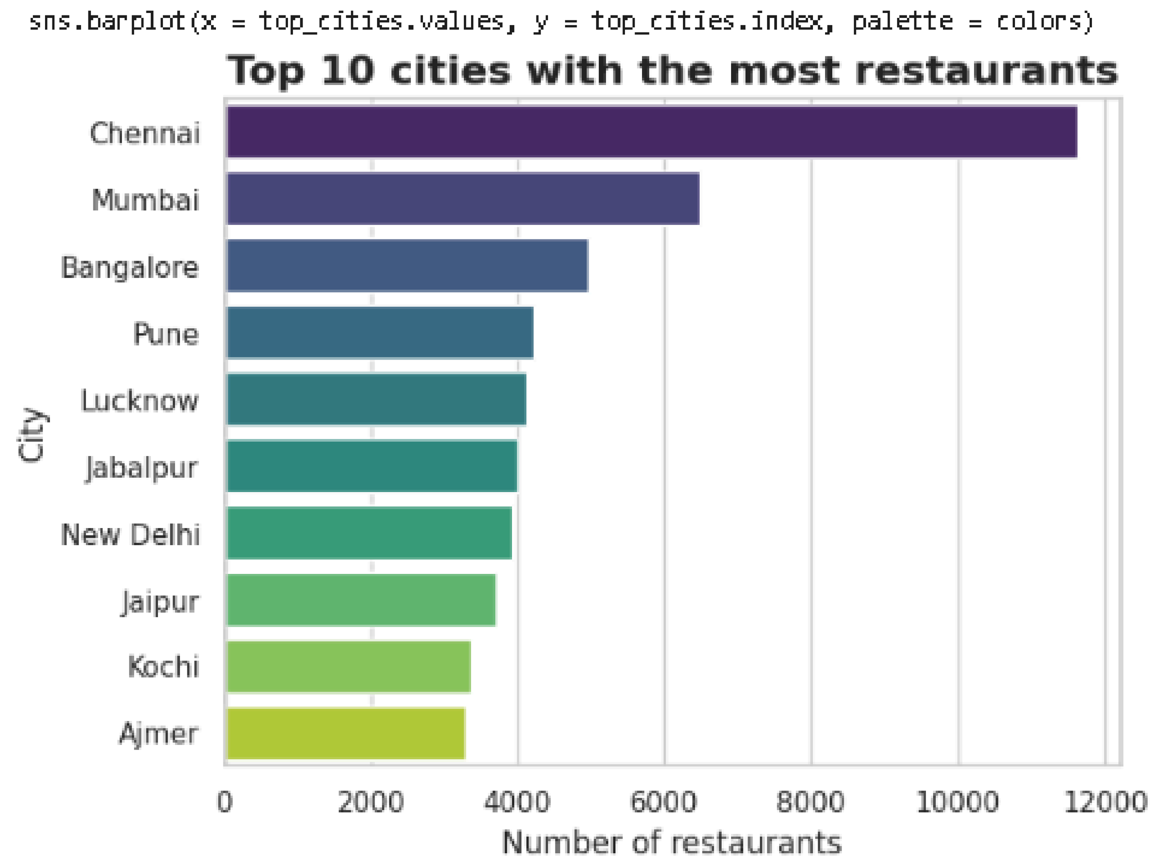
```
plt.figure(figsize=(10,6))
for city in df['city'].unique()[:5]:   # showing top 5 cities for clarity
    subset = df[df['city'] == city]
    plt.hist(subset['aggregate_rating'], bins=10, alpha=0.5, label=city)

plt.title('Distribution of Restaurant Ratings Across Cities')
plt.xlabel('Ratings')
plt.ylabel('Number of Restaurants')
plt.legend()
plt.show()
```

Location Analysis

This bar graph shows the top 10 cities with the highest number of restaurants listed on Zomato. Chennai has the maximum restaurant count, followed by Mumbai and Bangalore. The graph highlights how major metropolitan and Tier-1 cities dominate in terms of food outlet density. This helps identify regions with the most active food business markets.

```python
top_cities = df['city'].value_counts().head(10)
sns.set(style="whitegrid")
colors = sns.color_palette("viridis", len(top_cities))
sns.barplot(x = top_cities.values, y = top_cities.index, palette = colors)
plt.title('Top 10 cities with the most restaurants', fontsize = 16, weight = 'bold')
plt.xlabel('Number of restaurants', fontsize = 12)
plt.ylabel('City', fontsize = 12)
plt.show()
```

```
/tmp/ipython-input-1594069354.py:4: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `y` variable to `hue` and set `legend=False` for the same effect.

  sns.barplot(x = top_cities.values, y = top_cities.index, palette = colors)
```

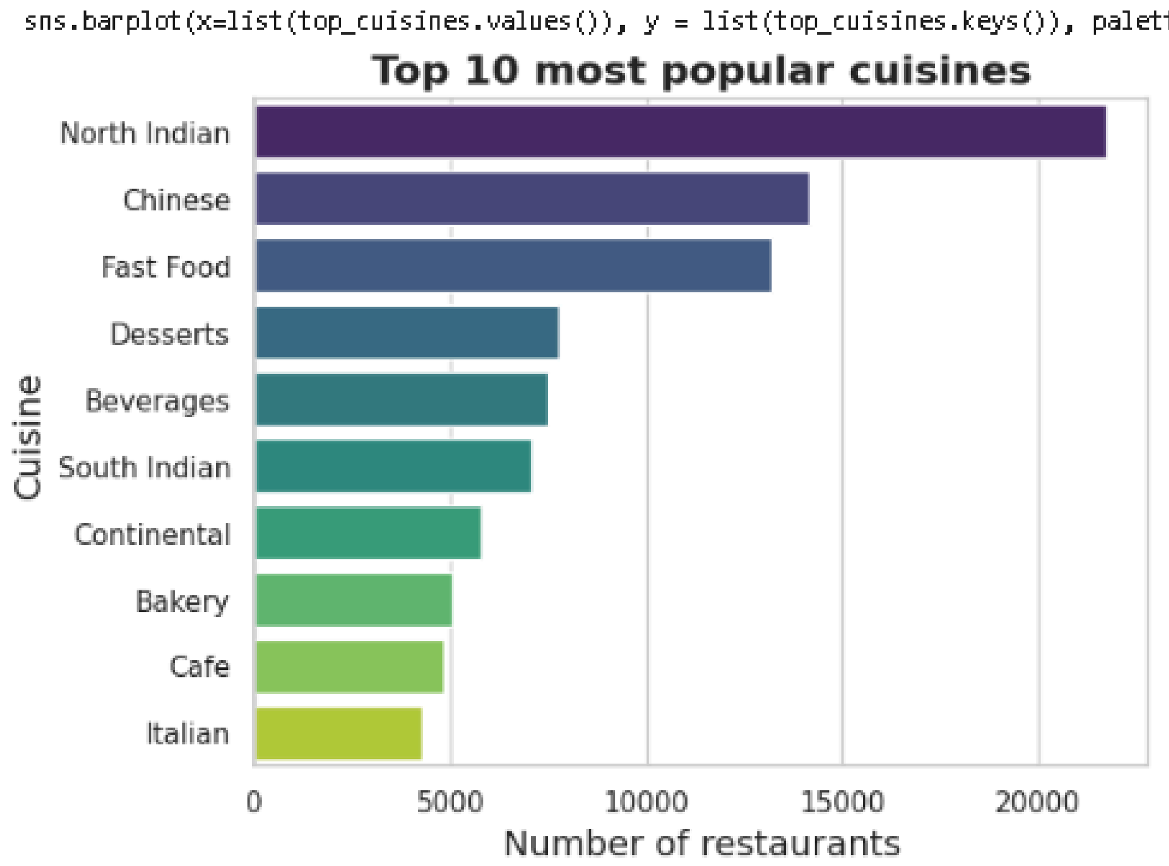


Cuisine Analysis

This bar graph shows the top 10 most popular cuisines offered by restaurants on Zomato. North Indian cuisine is the most common, followed by Chinese and Fast Food. The chart highlights customer demand and restaurant preferences across different food categories. It also helps identify which cuisines dominate the market and attract the highest number of restaurants.

```
from collections import Counter

cuisine_list = df['cuisines'].dropna().apply(lambda x: x.split(', '))
cuisine_counts = Counter([c for sublist in cuisine_list for c in sublist])
top_cuisines = dict(sorted(cuisine_counts.items(), key=lambda x: x[1], reverse=True)[:10])

colors = sns.color_palette("viridis", len(top_cities))

sns.barplot(x=list(top_cuisines.values()), y = list(top_cuisines.keys()), palette = colors)
plt.title('Top 10 most popular cuisines', fontsize = 16, weight = 'bold')
plt.xlabel('Number of restaurants', fontsize = 14)
plt.ylabel('Cuisine', fontsize = 14)
plt.show()
```

```
/tmp/ipython-input-3107561615.py:9: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `y` variable to `hue` and set `legend=False` for the same effect.

  sns.barplot(x=list(top_cuisines.values()), y = list(top_cuisines.keys()), palette = colors)
```

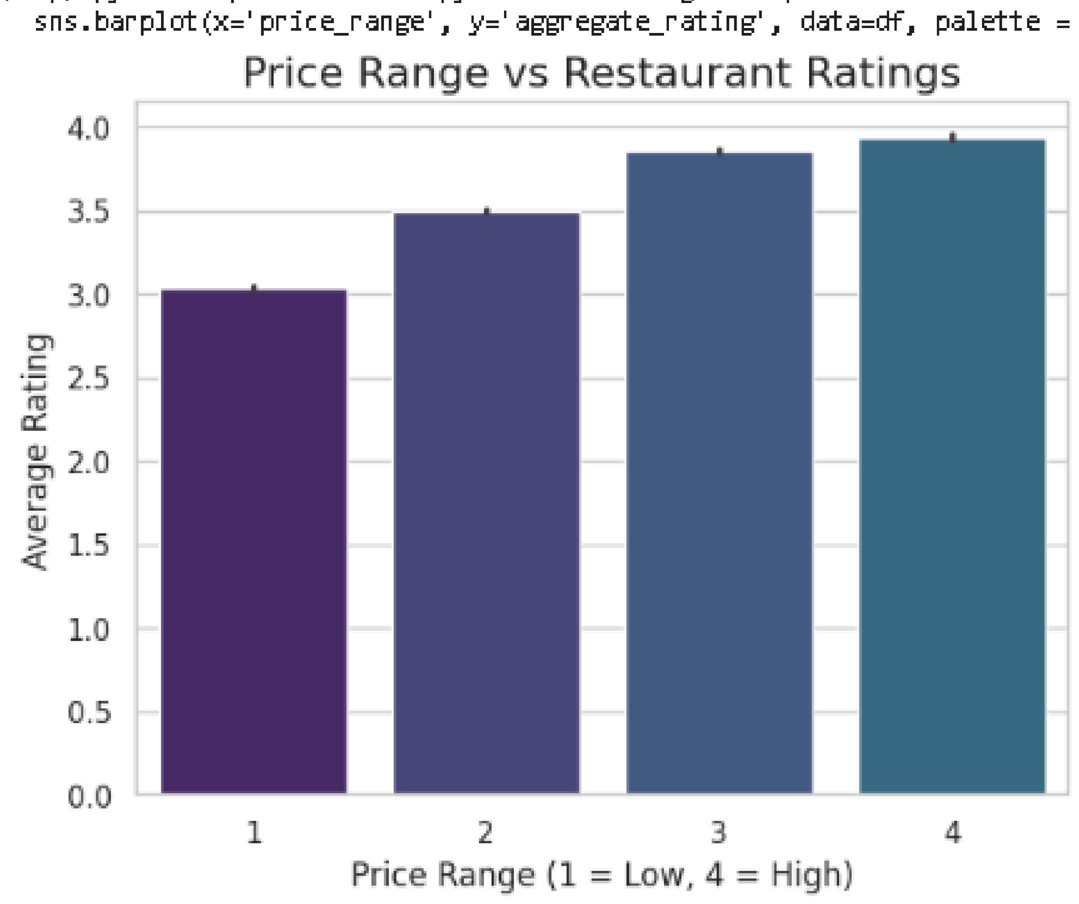


Price Range and Rating

The graph shows the average restaurant rating for each price range from 1 (lowest) to 4 (highest). As the price range increases, the average ratings also rise, indicating a positive relationship between cost and customer satisfaction. Lower-priced restaurants have ratings around 3.2–3.5, while higher-priced ones reach 3.9–4.0. This suggests that more expensive restaurants tend to be rated better overall.

```
sns.barplot(x='price_range', y='aggregate_rating', data=df, palette = colors)
plt.title('Price Range vs Restaurant Ratings', fontsize=16)
plt.xlabel('Price Range (1 = Low, 4 = High)')
plt.ylabel('Average Rating')
plt.show()
```

```
/tmp/ipython-input-669323932.py:1: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

  sns.barplot(x='price_range', y='aggregate_rating', data=df, palette = colors)
/tmp/ipython-input-669323932.py:1: UserWarning: The palette list has more values (10) than needed (4), which may not be intended.
  sns.barplot(x='price_range', y='aggregate_rating', data=df, palette = colors)
```



Online order and Table booking

The graph compares restaurant ratings based on online ordering availability. Restaurants that offer online ordering (value = 1) generally show higher median ratings than those that don't. The boxplots also show that restaurants without online ordering have more low-rating outliers, indicating inconsistent performance. Overall, the plot suggests that offering online ordering is linked to better customer ratings.

```
sns.boxplot(x='delivery', y='aggregate_rating', data=df, palette = colors)
plt.title('Impact of Online Ordering on Ratings', fontsize=14)
plt.xlabel('Online Order Available')
plt.ylabel('Ratings Distribution')
plt.show()
```

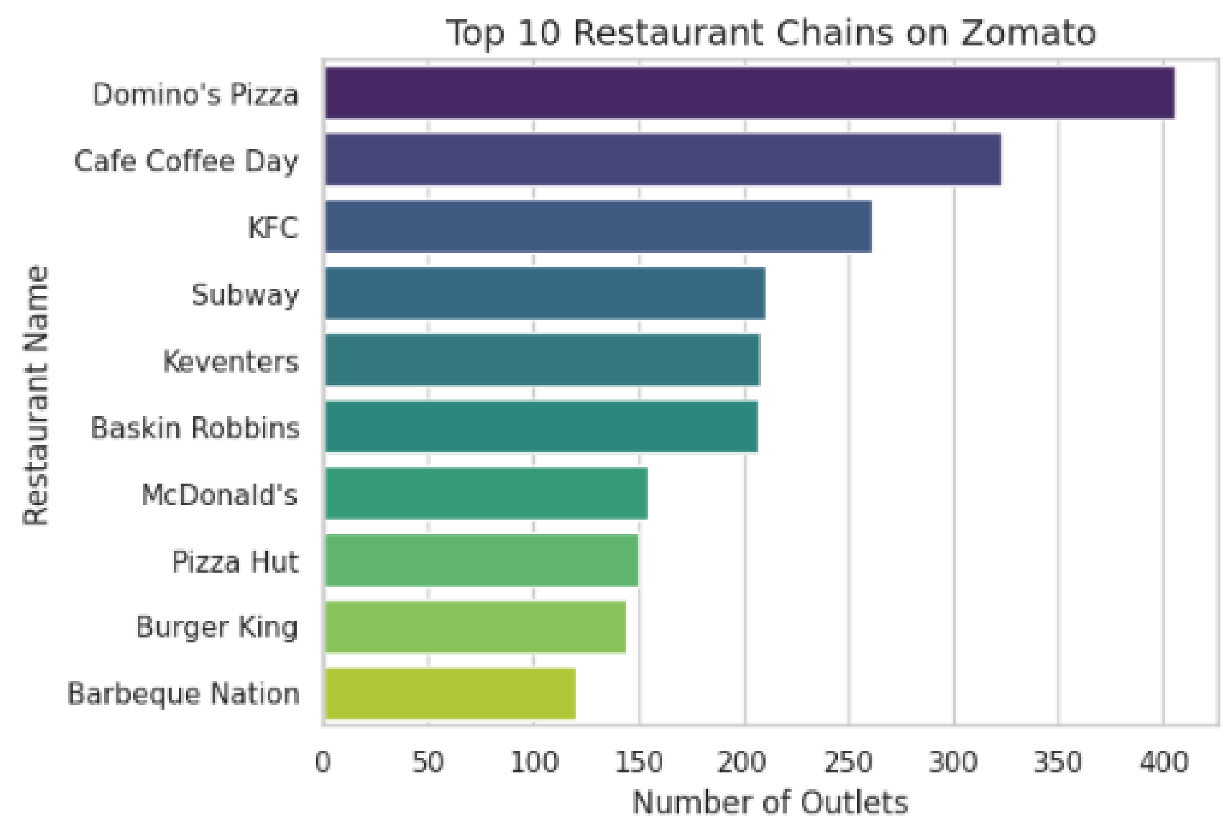
```
/tmp/ipython-input-2718237232.py:1: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

  sns.boxplot(x='delivery', y='aggregate_rating', data=df, palette = colors)
/tmp/ipython-input-2718237232.py:1: UserWarning: The palette list has more values (10) than needed (3), which may not be intended.
  sns.boxplot(x='delivery', y='aggregate_rating', data=df, palette = colors)
```

Top Restaurant Chains

The bar chart shows the top 10 restaurant chains on Zomato based on the number of outlets. Domino’s Pizza has the highest presence with around 400 outlets, followed by Cafe Coffee Day and KFC. The remaining chains like Subway, McDonald's, and Burger King also have strong nationwide coverage but at lower counts. Overall, the chart highlights which restaurant brands have the widest reach and strongest market presence on Zomato.

```python
top_chains = df['name'].value_counts().head(10)
sns.barplot(x=top_chains.values, y=top_chains.index, palette = colors)
plt.title('Top 10 Restaurant Chains on Zomato', fontsize=14)
plt.xlabel('Number of Outlets')
plt.ylabel('Restaurant Name')
plt.show()
```

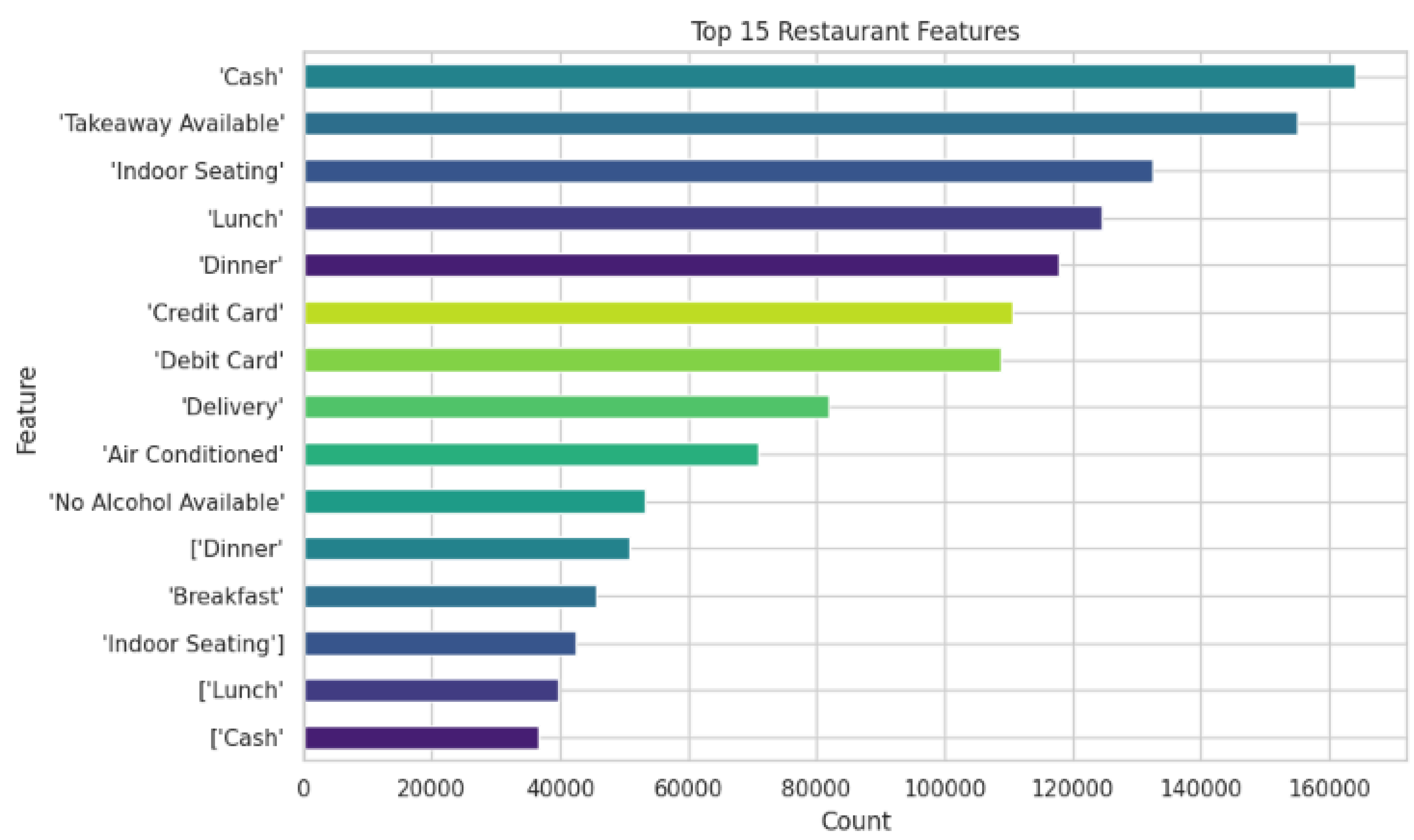
```
/tmp/ipython-input-3136813891.py:2: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `y` variable to `hue` and set `legend=False` for the same effect.

  sns.barplot(x=top_chains.values, y=top_chains.index, palette = colors)
```



Restaurant Features

The bar chart displays the top 15 most common features offered by Indian restaurants on Zomato. “Cash” and “Takeaway Available” appear as the most frequent features, indicating their wide availability across restaurants. Popular dining-related features like Indoor Seating, Lunch, and Dinner also show high counts. Payment options such as Credit Card and Debit Card are moderately common.

```
[30]
plt.figure(figsize=(10,6))
feature_counts.sort_values().plot(
    kind='barh',
    color=colors[:len(feature_counts)]   # apply palette
)
plt.xlabel("Count")
plt.ylabel("Feature")
plt.title("Top 15 Restaurant Features")
plt.tight_layout()
plt.show()
```



Seasonal Trends

The graph shows the average restaurant ratings across different days of the week. Ratings remain consistently high from Monday to Thursday, with mid-week (Tuesday–Thursday) showing the best performance. On Friday, ratings begin to drop, possibly due to higher customer traffic and longer wait times affecting experience. Saturday has the lowest average rating, suggesting weekend rush impacts service quality. On Sunday, ratings recover slightly but remain below weekday levels. Overall, the trend indicates that restaurants perform best on weekdays and weakest during the busy weekend period.

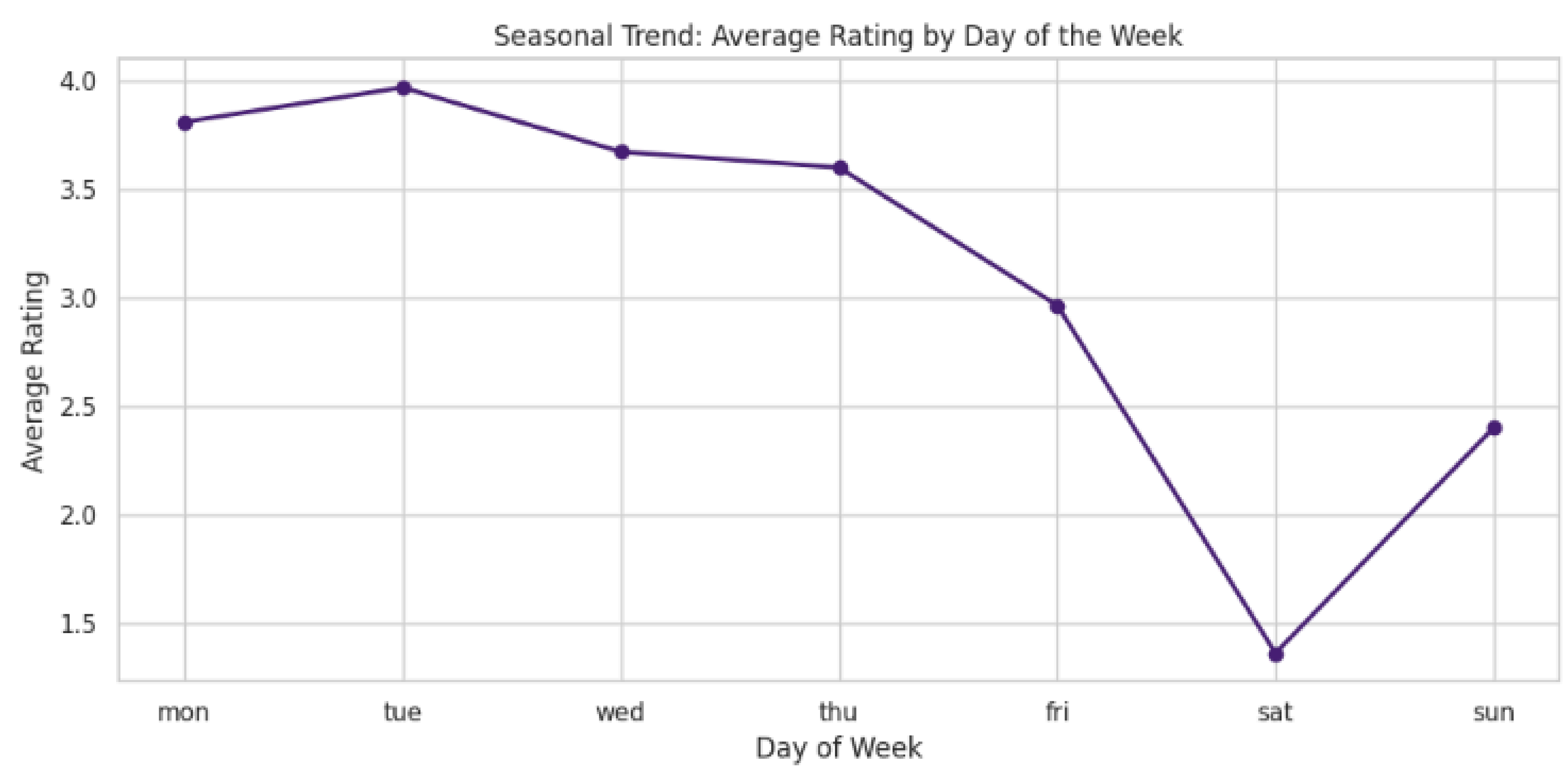
[31]
```
df['day'] = df['timings'].str.extract(r'([A-Za-z]{3})', expand=False).str.lower()
```

[32]
```
df_day = df.dropna(subset=['day', 'aggregate_rating'])
```

[33]
```
day_rating = df_day.groupby('day')['aggregate_rating'].mean()
```

[34]
```
order = ['mon','tue','wed','thu','fri','sat','sun']
day_rating = day_rating.reindex(order)
```

[36]
```
plt.figure(figsize=(10,5))
plt.plot(day_rating.index, day_rating.values, marker='o', linewidth=2,
         color=colors[0])   # Using first palette color

plt.title("Seasonal Trend: Average Rating by Day of the Week")
plt.xlabel("Day of Week")
plt.ylabel("Average Rating")
plt.grid(True)
plt.tight_layout()
plt.show()
```



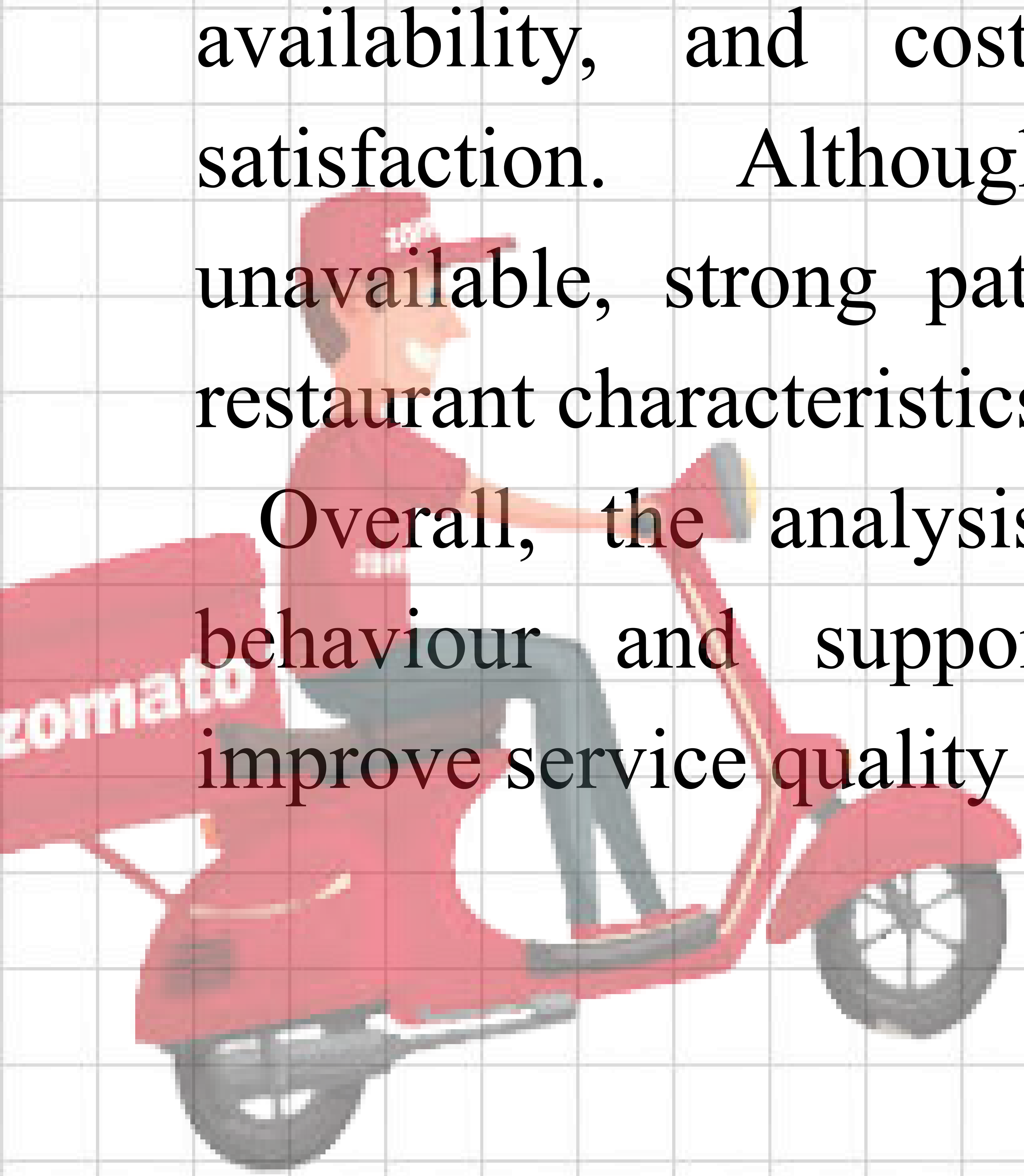
Conclusion

The Zomato dataset analysis provides clear insights into how different restaurant factors influence customer choices and overall ratings. Cities with more restaurants show varied rating patterns due to higher competition. Cuisine diversity plays an important role, with multi-cuisine restaurants generally performing better.

Price range analysis shows that customers expect higher quality from premium restaurants, while affordable ones receive mixed feedback. Features like delivery, takeaway, and special highlights significantly impact customer preferences.

The study also reveals how location, service availability, and cost for two affect customer satisfaction. Although time-based data was unavailable, strong patterns were identified through restaurant characteristics.

Overall, the analysis helps understand customer behaviour and supports data-driven decisions to improve service quality and restaurant performance.



zomato

Thank You!!!