Analyzing Customer Preferences and Restaurant Performance Using Zomato Dataset

- Dhai Hisham Ibrahim Manaa
- Christine Gabrail Saad Botros
- Sara Amr Ismail Salama
- Nadine Ehab Jomaa Muhammed
- Nouran Tarek Youssef Mahra

Supervised by Dr. Ahmed Abd-Allatef

With the generous support of AST company

In Today’s competitive food industry, understanding customer preferences is essential. Zomato, a leading restaurants platform, provides valuable data to explore what makes restaurants successful and how pricing varies across cities. This project analyzes the Zomato dataset to uncover key restaurant trends and customers insights.

The dataset contains information on restaurants, includes their names, locations, ratings and more. With over than 51,000 records and 17 columns.

```
df.head()
```

url	address	name	online_order	book_table	rate	votes	phone	location	rest_type	dish_liked	cuisines	approx_cost(for two people)	reviews_list
ore/jalsa-banasha...	942, 21st Main Road, 2nd Stage, Banashankari, ...	Jalsa	Yes	Yes	4.1/5	775	080 42297555\r\n+91 9743772233	Banashankari	Casual Dining	Pasta, Lunch Buffet, Masala Papad, Paneer Laja...	North Indian, Mughlai, Chinese	800	[('Rated 4.0', 'RATED\n A beautiful place to ...
re/spice-elephan...	2nd Floor, 80 Feet Road, Near Big Bazaar, 6th ...	Spice Elephant	Yes	No	4.1/5	787	080 41714161	Banashankari	Casual Dining	Momos, Lunch Buffet, Chocolate Nirvana, Thai G...	Chinese, North Indian, Thai	800	[('Rated 4.0', 'RATED\n Had been here for din...
ngalore? cont...	1112, Next to KIMS Medical College, 17th Cross...	San Churro Cafe	Yes	No	3.8/5	918	+91 9663487993	Banashankari	Cafe, Casual Dining	Churros, Cannelloni, Minestrone Soup, Hot Choc...	Cafe, Mexican, Italian	800	[('Rated 3.0', "RATED\n Ambience is not that ...
'addhuri-udupi...	1st Floor, Annakuteera, 3rd Stage, Banashankar...	Addhuri Udupi Bhojana	No	No	3.7/5	88	+91 9620009302	Banashankari	Quick Bites	Masala Dosa	South Indian, North Indian	300	[('Rated 4.0', "RATED\n Great food and proper...
e/grand-village...	10, 3rd Floor, Lakshmi Associates, Gandhi Baza...	Grand Village	No	No	3.8/5	166	+91 8026612447\r\n+91 9901210005	Basavanagudi	Casual Dining	Panipuri, Gol Gappe	North Indian, Rajasthani	600	[('Rated 4.0', 'RATED\n Very good restaurant ...

The Goal of This Analysis

Identifying factors that contribute to high restaurant ratings

Understanding how location impacts restaurant prices and ratings.

Discovering trends in restaurant categories and customer preferences.

Data Cleaning

a- Handling missing values.

b- Handling duplicates.

c- Handling unique values.

d- Renaming columns.

Dealing with Messing and Duplicated Values


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

```
name                0
online_order        0
book_table          0
rate             7775
votes               0
phone            1208
location           21
rest_type         227
cuisines           45
Cost2people       346
reviews_list        0
Type                0
city                0
Unique_ID           0
dtype: int64
```

```
def Calculate_Null_Percentage(column_name):
    null_percentage = df[column_name].isnull().mean() * 100
    print(f"Percentage of null values in '{column_name}': {null_percentage:.2f}%")
```

```
Calculate_Null_Percentage('rate')
Calculate_Null_Percentage('location')
Calculate_Null_Percentage('rest_type')
Calculate_Null_Percentage('cuisines')
Calculate_Null_Percentage('Cost2people')
Calculate_Null_Percentage('phone')
Calculate_Null_Percentage('dish_liked')
```

```
Percentage of null values in 'rate': 15.03%
Percentage of null values in 'location': 0.04%
Percentage of null values in 'rest_type': 0.44%
Percentage of null values in 'cuisines': 0.09%
Percentage of null values in 'Cost2people': 0.67%
Percentage of null values in 'phone': 2.34%
Percentage of null values in 'dish_liked': 54.29%
```

```
df= df.drop(['url','address','menu_item','dish_liked'], axis=1)
```

```
df.shape
```

```
(51717, 14)
```

```
df.duplicated().sum()
```

```
66
```

```
df.drop_duplicates(inplace= True)
df.shape
```

```
(51651, 14)
```

```
df['rate'].fillna(df['rate'].mean(), inplace=True)
```

```
df.dropna(inplace= True)
```

Dealing with Unique Values in Columns

```
df['rate'].unique()
```

```
array(['4.1/5', '3.8/5', '3.7/5', '3.6/5', '4.6/5', '4.0/5', '4.2/5',
       '3.9/5', '3.1/5', '3.0/5', '3.2/5', '3.3/5', '2.8/5', '4.4/5',
       '4.3/5', 'NEW', '2.9/5', '3.5/5', nan, '2.6/5', '3.8 /5', '3.4/5',
       '4.5/5', '2.5/5', '2.7/5', '4.7/5', '2.4/5', '2.2/5', '2.3/5',
       '3.4 /5', '-', '3.6 /5', '4.8/5', '3.9 /5', '4.2 /5', '4.0 /5',
       '4.1 /5', '3.7 /5', '3.1 /5', '2.9 /5', '3.3 /5', '2.8 /5',
       '3.5 /5', '2.7 /5', '2.5 /5', '3.2 /5', '2.6 /5', '4.5 /5',
       '4.3 /5', '4.4 /5', '4.9/5', '2.1/5', '2.0/5', '1.8/5', '4.6 /5',
       '4.9 /5', '3.0 /5', '4.8 /5', '2.3 /5', '4.7 /5', '2.4 /5',
       '2.1 /5', '2.2 /5', '2.0 /5', '1.8 /5'], dtype=object)
```

```
def handleRate(value):
    if(value== 'NEW' or value =='-'):
        return np.nan
    else:
        value= str(value).split('/')
        return float(value[0])
df['rate']= df['rate'].apply(handleRate)
df['rate'].head()
```

```
0    4.1
1    4.1
2    3.8
3    3.7
4    3.8
Name: rate, dtype: float64
```

```
(df['Cost2people'].unique())
```

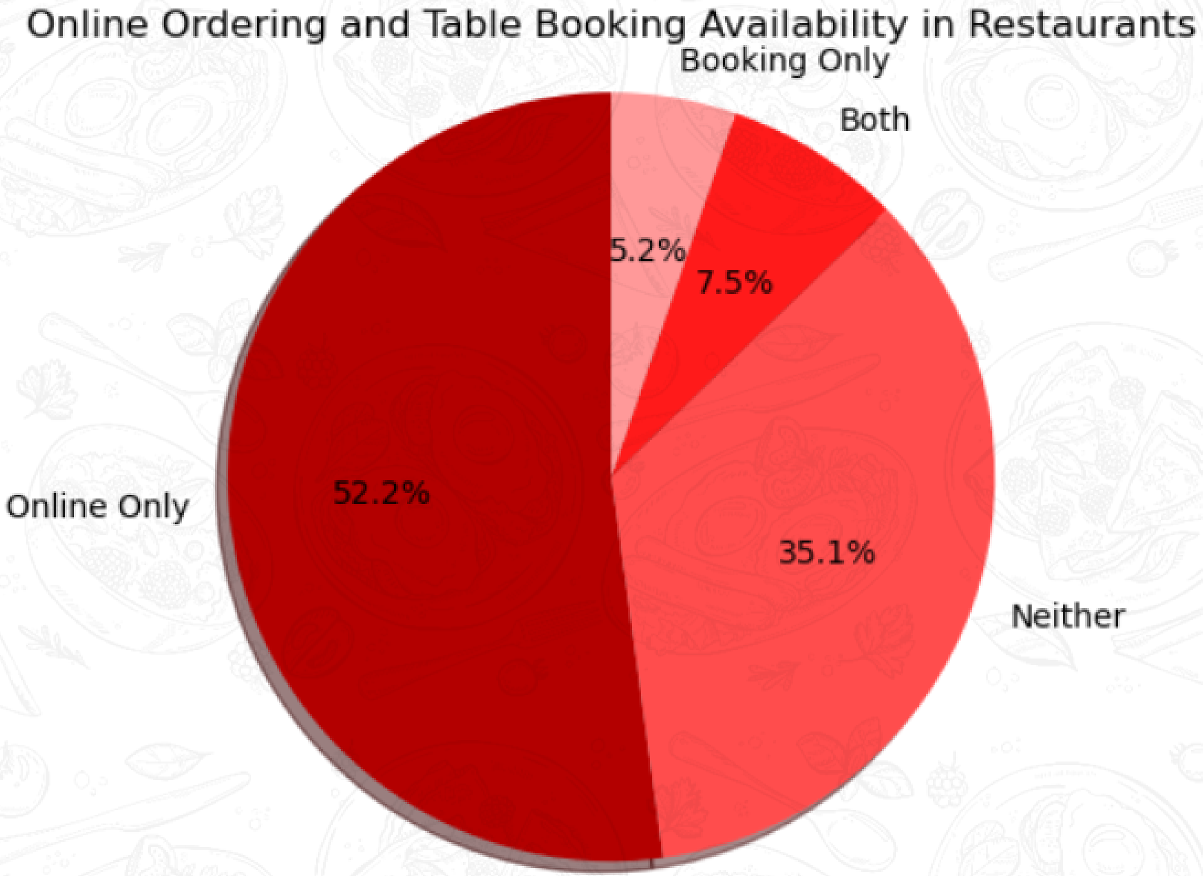
```
array(['800', '300', '600', '700', '550', '500', '450', '650', '400',
       '900', '200', '750', '150', '850', '100', '1,200', '350', '250',
       '950', '1,000', '1,500', '1,300', '199', '80', '1,100', '160',
       '1,600', '230', '130', '50', '190', '1,700', '1,400', '180',
       '1,350', '2,200', '2,000', '1,800', '1,900', '330', '2,500',
       '2,100', '3,000', '2,800', '3,400', '40', '1,250', '3,500',
       '4,000', '2,400', '2,600', '120', '1,450', '469', '70', '3,200',
       '60', '560', '240', '360', '6,000', '1,050', '2,300', '4,100',
       '5,000', '3,700', '1,650', '2,700', '4,500', '140'], dtype=object)
```

```
def handleComma(value):
    if ',' in value:
        value= value.replace(',' , '')
        return float(value)
    else:
        return float(value)

df['Cost2people']= df['Cost2people'].apply(handleComma)
df['Cost2people'].head()
```

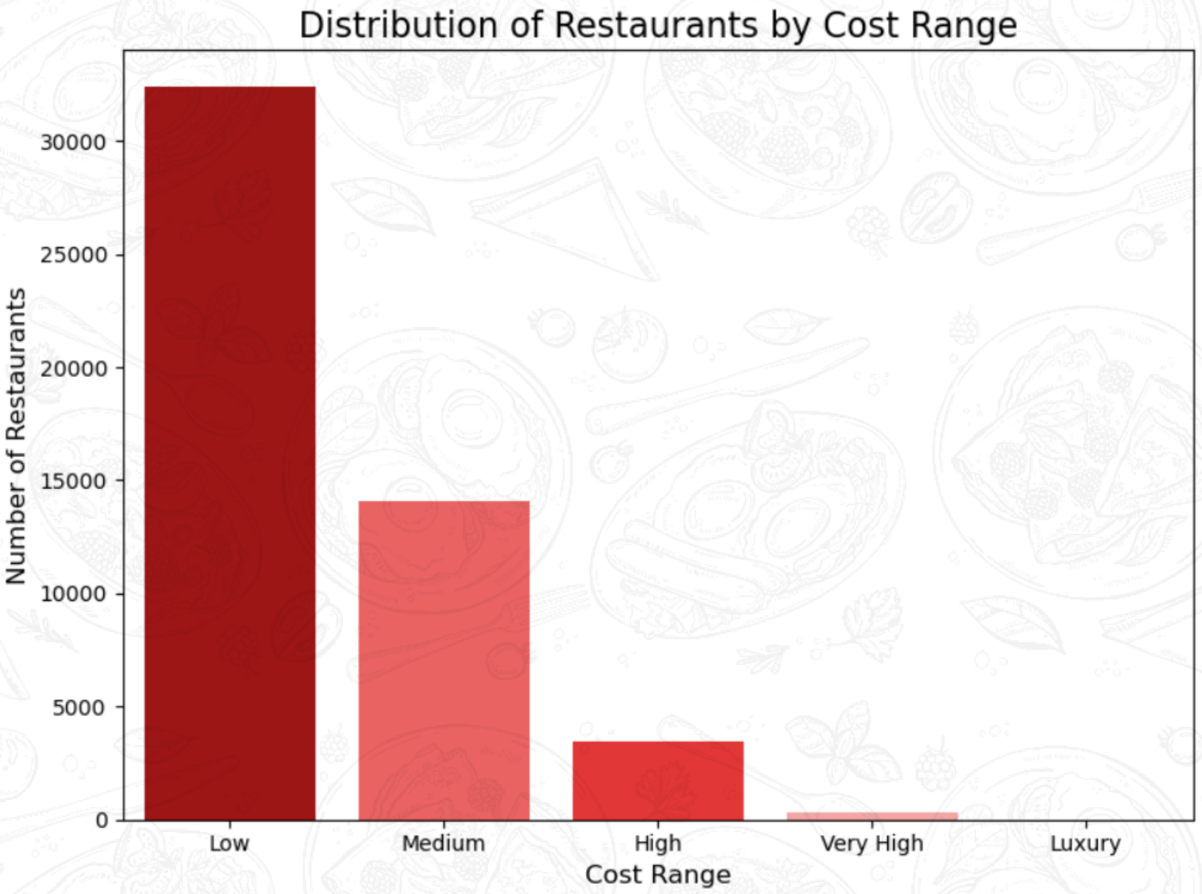
```
0    800.0
1    800.0
2    800.0
3    300.0
4    600.0
Name: Cost2people, dtype: float64
```

Exploratory Data Analysis

Almost half of the restaurants offer online ordering, while only 5% provide booking services, and less than 10% offer both. Approximately 35% of the restaurants do not provide either service.

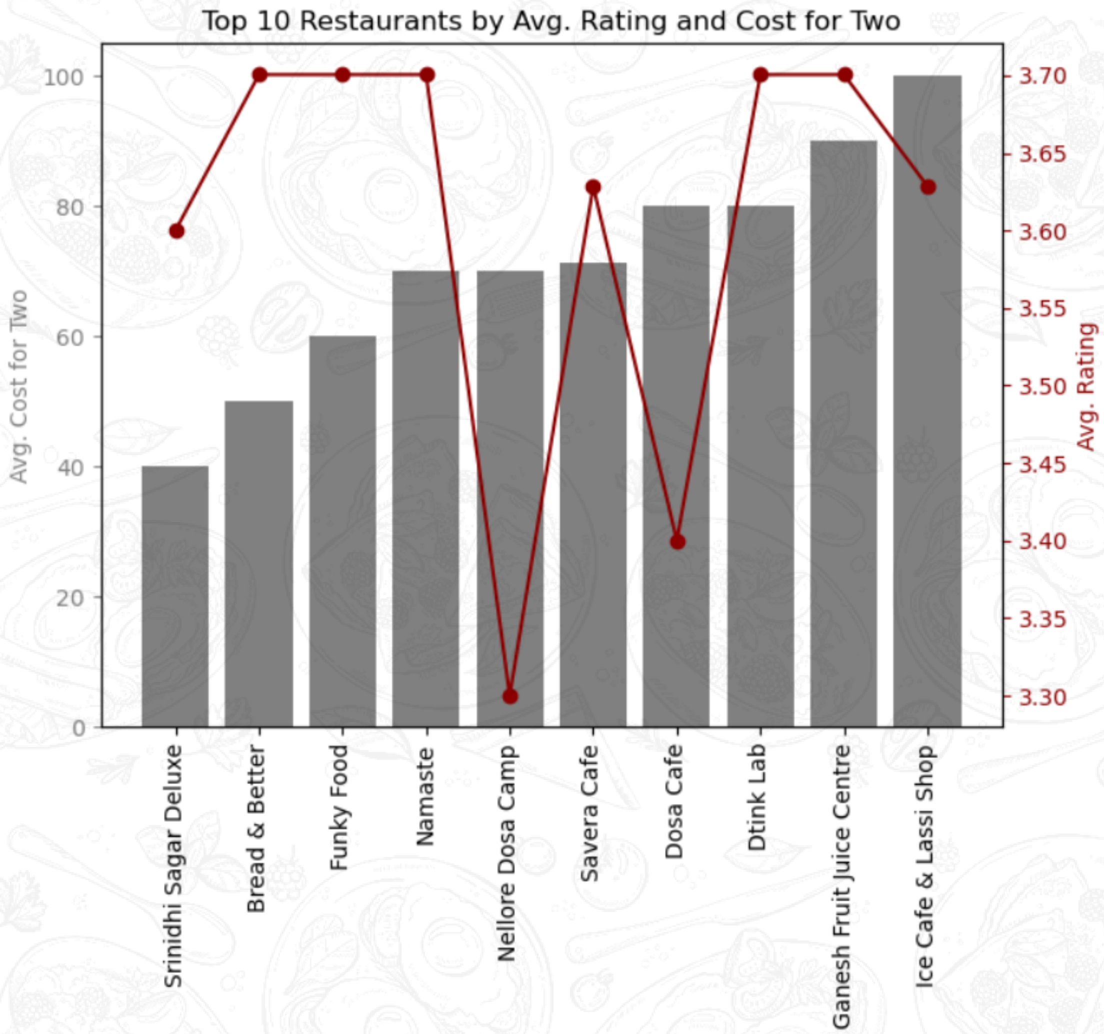


Almost 80% of the restaurants offer affordable meals for two people, with prices falling into the low to medium range. The remaining 20% are priced in the high to very high range

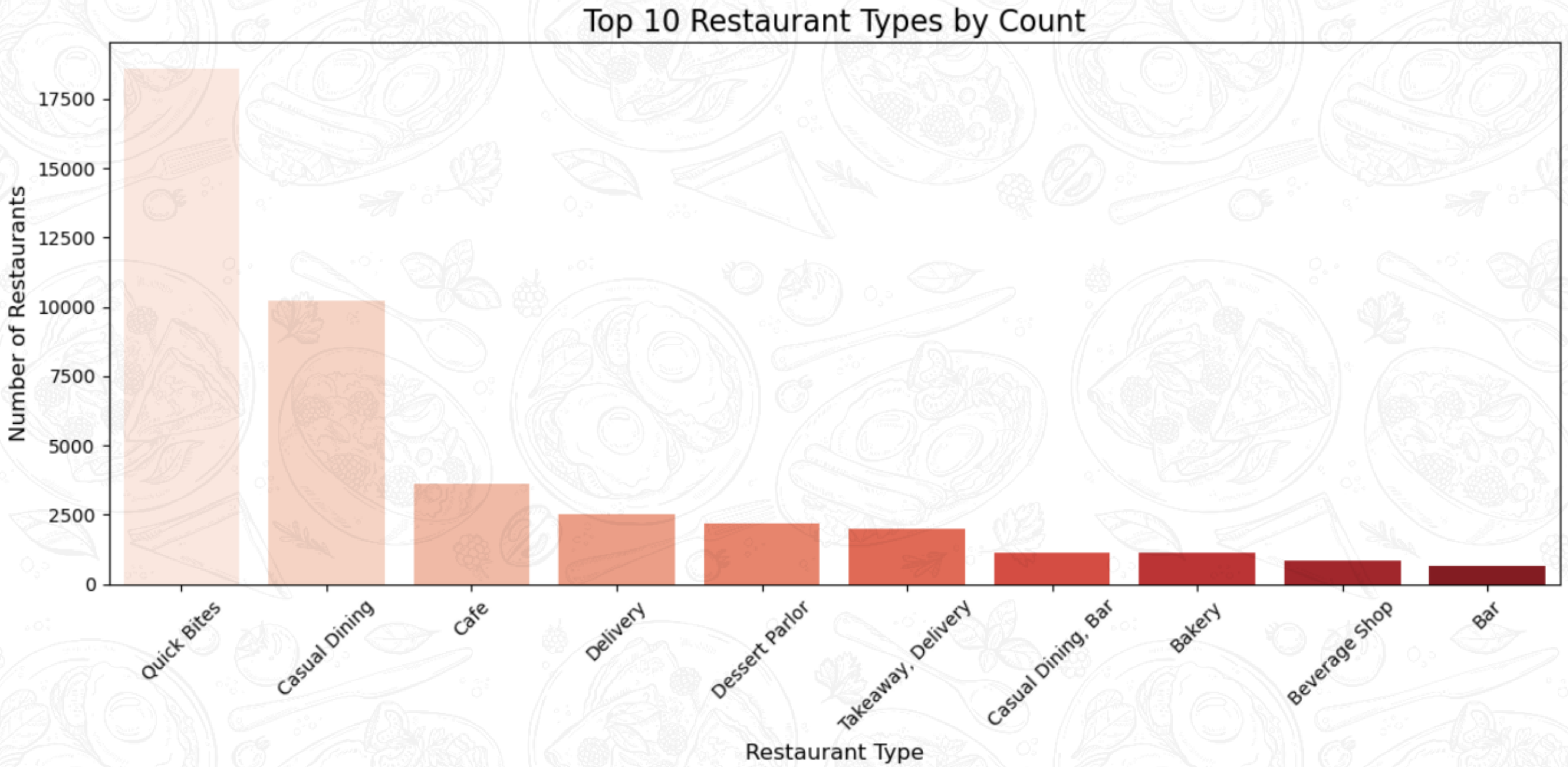


The chart compares the top 10 restaurants based on two variables: the average cost for two people (represented by bars) and the average rating (represented by the red line).

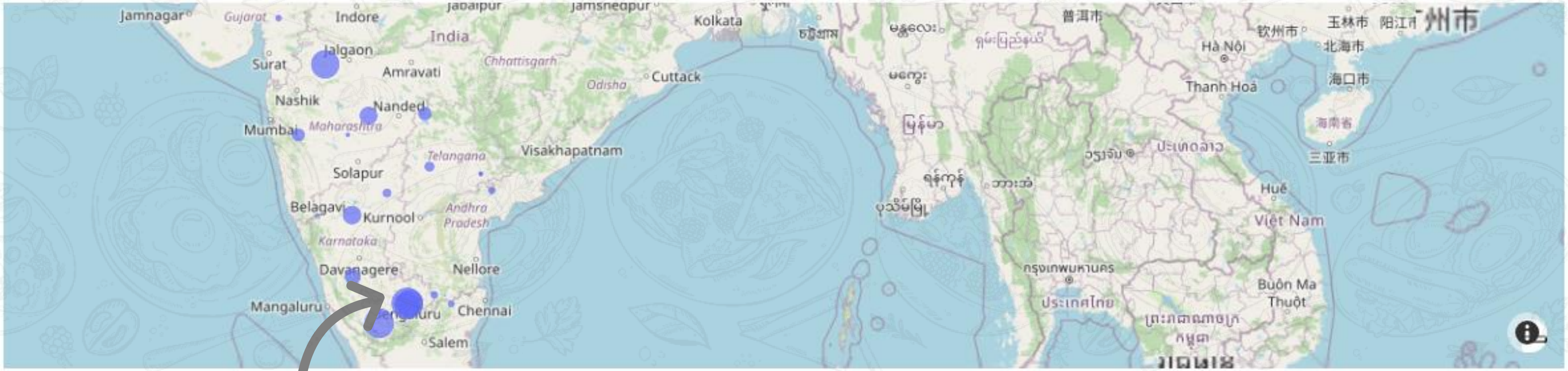
Each restaurant shows a different relationship between cost and rating. For instance, Sindhi Sagar Deluxe has one of the lowest costs and moderate ratings, while Nellore Dosa Camp has the lowest rating despite having a higher cost. This visualization helps in analyzing how cost impacts customer ratings across these restaurants.

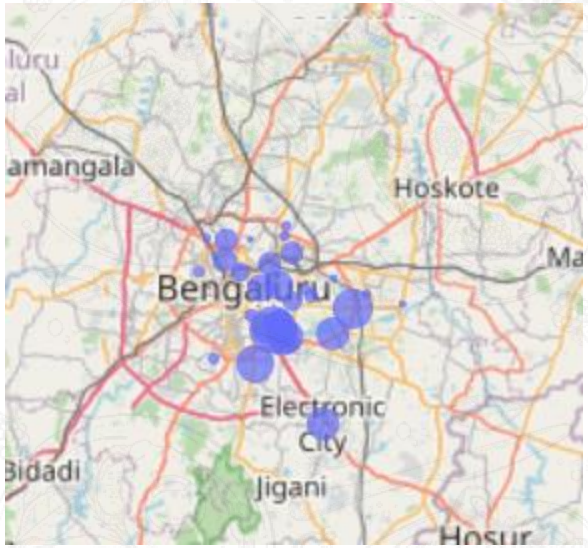


The most popular restaurant type is "Quick Bites," while the least popular is "Bar." This trend may be attributed to people's preference for quick meals due to fast-paced lifestyles and hectic schedules.



Map of Restaurant Count by Location





1- The majority of restaurants are clustered in Bengaluru.

2- Potential restaurant owners can use this information to identify good locations for their ventures.

Machine Learning Model

Goal: Predict restaurant success (ratings) based on features like online_order, book_table, Cost2people, votes, and encoded categorical features (rest_type, cuisines, location).

Model: Random Forest Regressor

Step one :Handling Categorical Data

رواد مصر الرقمية

Handling Categorical Data

```
print(df['online_order'].unique())
```

```
['Yes' 'No']
```

```
from sklearn.preprocessing import OneHotEncoder
```

```
df['online_order'] = df['online_order'].map({'Yes': 1, 'No': 0})
df['book_table'] = df['book_table'].map({'Yes': 1, 'No': 0}) # for bianary columns
```

```
print(df['online_order'].unique())
```

```
[1 0]
```

```
df = pd.get_dummies(df, columns=['rest_type', 'cuisines', 'location'], drop_first=True) # one hot encoding for categorical col
```

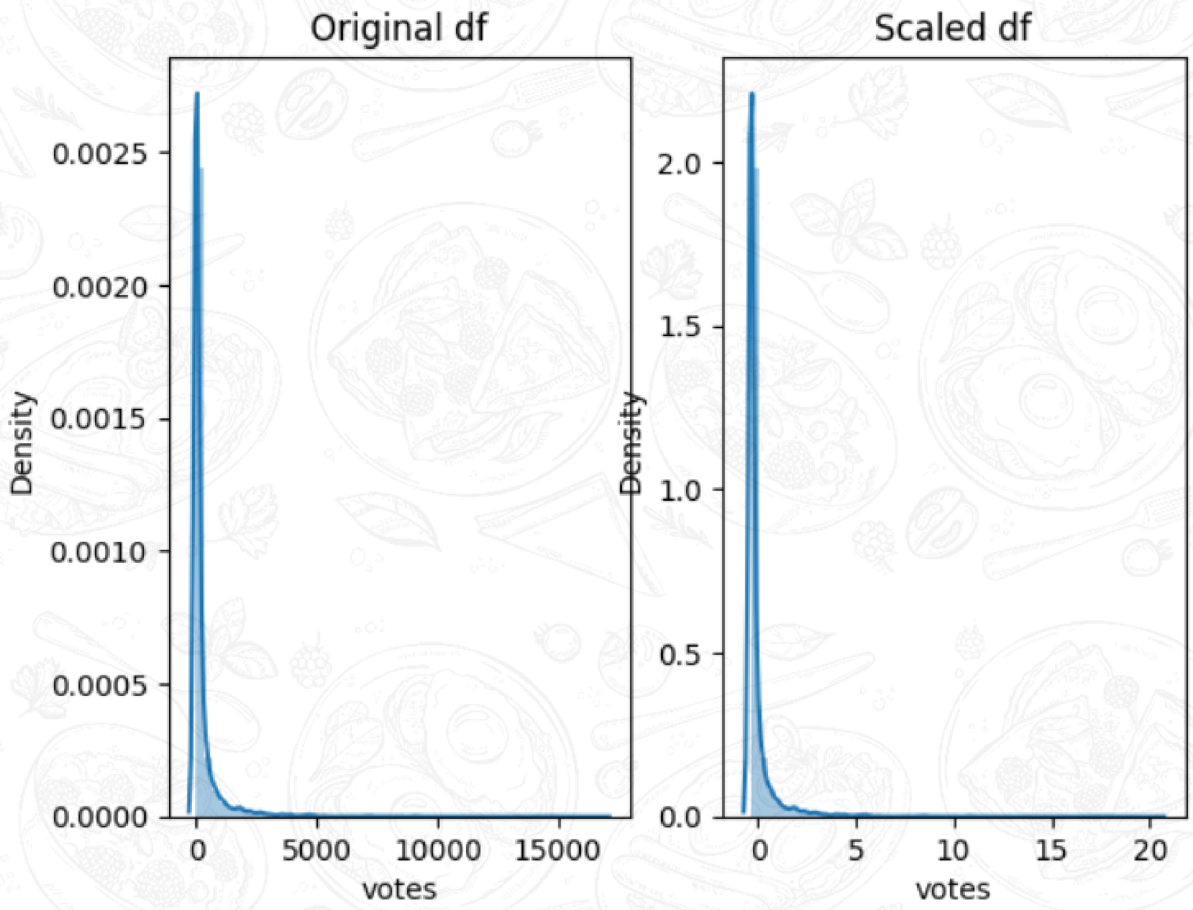
Step Two :Scale and Normalize Data

Scale and Normalize data

```
from sklearn.preprocessing import StandardScaler
old_votes=df['votes'].copy()
old_cost=df['Cost2people'].copy()

scaler = StandardScaler()

df['Cost2people'] = scaler.fit_transform(df[['Cost2people']])
df['votes'] = scaler.fit_transform(df[['votes']])
```

```
fig, ax=plt.subplots(1,2)
sns.distplot(old_votes, ax=ax[0])
ax[0].set_title("Original df")
sns.distplot(df['votes'], ax=ax[1])
ax[1].set_title("Scaled df")
```

```
Text(0.5, 1.0, 'Scaled df')
```



Step Three :Predicting Restaurant Success

رواد مصر الرقمية

Predicting Restaurant Success

```
X = df[['online_order', 'book_table', 'Cost2people', 'votes'] +
        [col for col in df.columns if col.startswith('restaurant_type_')] +
        [col for col in df.columns if col.startswith('cuisine_types_')] +
        [col for col in df.columns if col.startswith('location_')]]
y = df['rate']
```

```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

print(f'Shape of X: {X.shape}')
print(f'Shape of y: {y.shape}')
```

```
Shape of X: (50215, 96)
Shape of y: (50215,)
```

```
from sklearn.ensemble import RandomForestRegressor

rf_regressor = RandomForestRegressor(random_state=42)

# Fit the model to the training data
rf_regressor.fit(X_train, y_train)

# Make predictions on the test set
y_pred = rf_regressor.predict(X_test)
```

Step Four :Model Evaluation

Model Evaluation

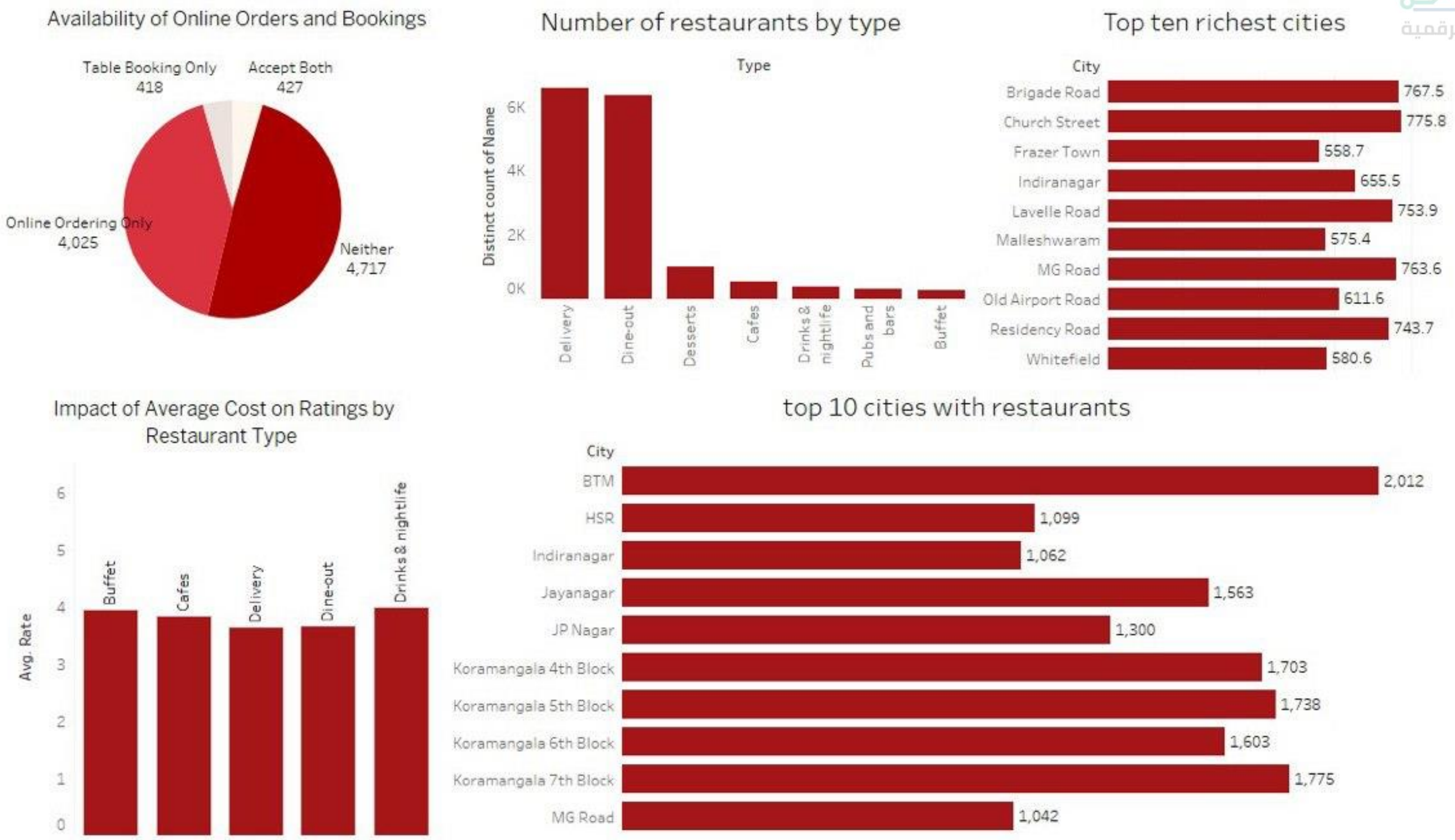
```
from sklearn.metrics import mean_absolute_error, r2_score

r2 = r2_score(y_test, y_pred)


print(f'R-squared:{r2}')
```

```
R-squared:0.8616635597172214
```

Dashboards

We used Tableau to build insightful dashboards

zomato Dashboard

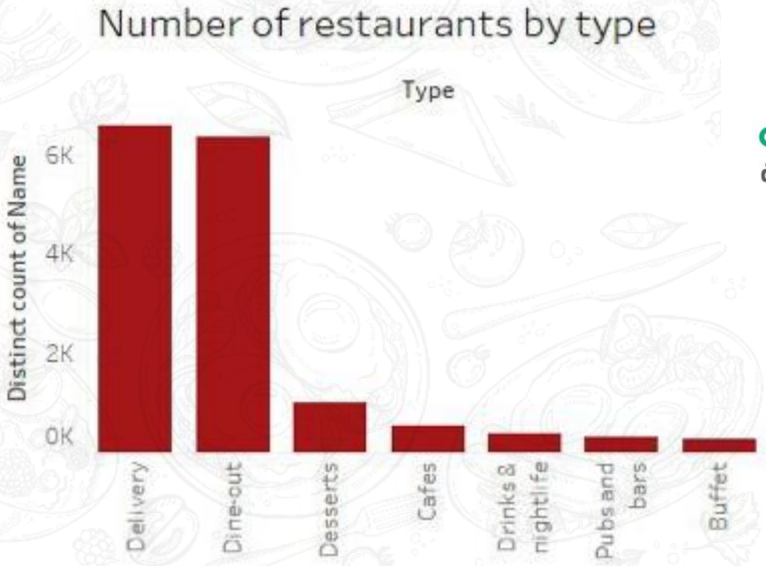


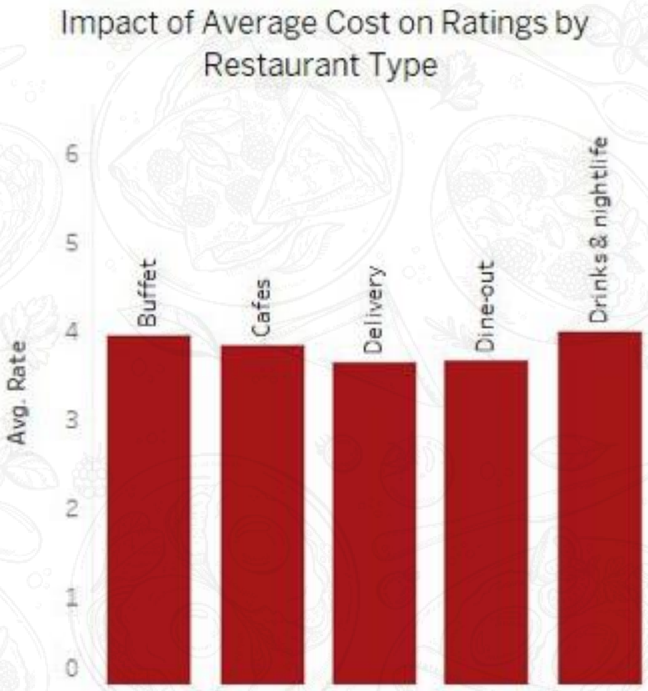
Key Insights Recap:

- Delivery and dine-out dominate, indicating high customer demand for these formats.

- Buffets and bars are relatively rare, which may suggest niche market opportunities.

- Higher average costs, like those for nightlife spots, tend to have better ratings, showing a positive correlation between price and perceived quality.



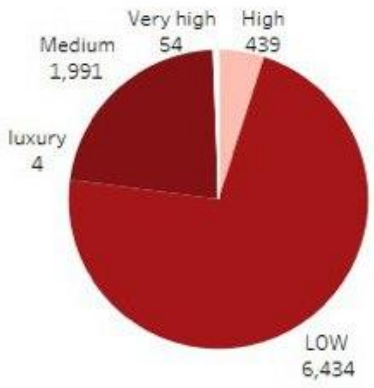


zomato **Dashboard**

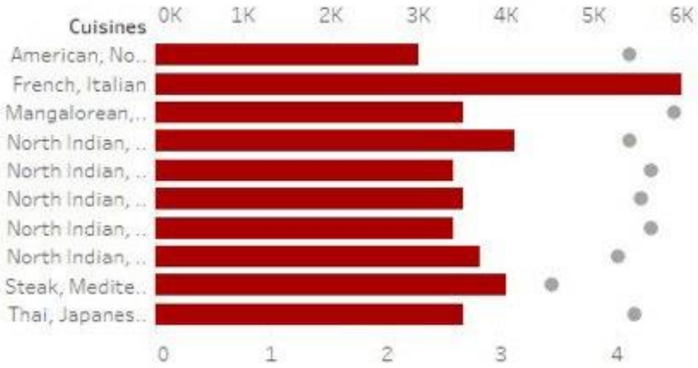
Selected Type: Buffet

Total Votes	Total Restuarants	AVG Ratings
15M	8,723	3.7

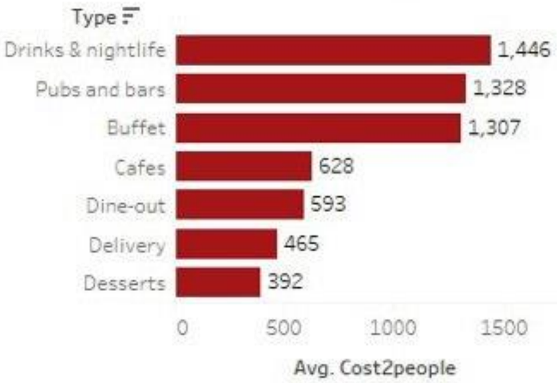
Distribution of Restaurant by cost range



Top 10 cuisines by AVG cost and AVG ratings



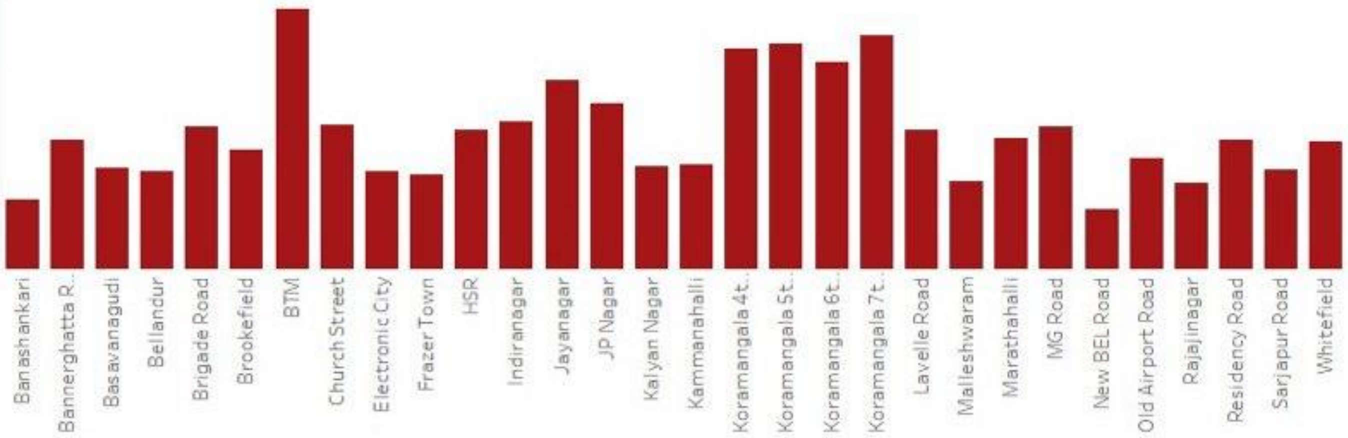
AVG cost of Two by restaurant type



Top 10 restaurants

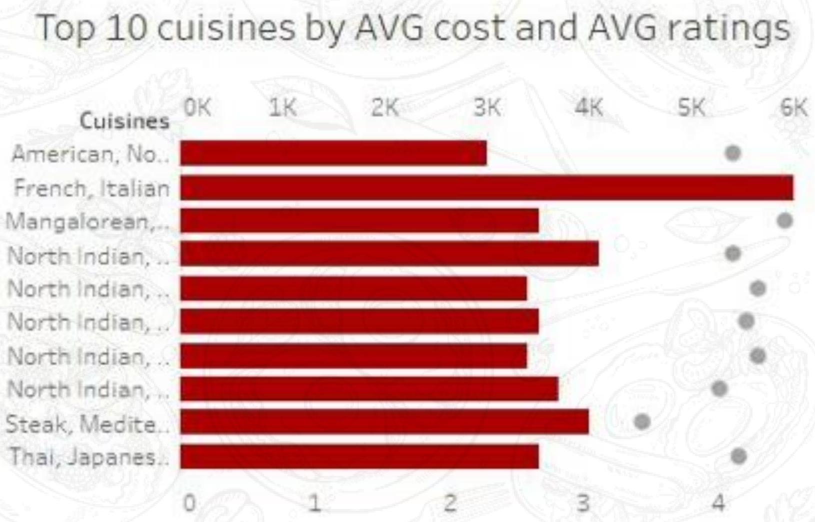
Name	Cuisines	Avg. Rate	Votes
AB's - Absol..	European, Mediterran..	5	86,418
Asia Kitche..	Asian, Chinese, Thai, ..	5	42,273
Barbecue b..	BBQ	5	2,683
Belgian Wa..	Desserts	5	24,882
Byg Brewsk..	Continental, North Ind..	5	99,531
Flechazo	Asian, Mediterranean,..	5	29,956
O.G. Variar ..	Bakery, Desserts	5	2,317
Punjab Grill	North Indian	5	1,822
	North Indian, Mughlai	5	7,838
SantÃ ◦ Â ◦ ..	Healthy Food, Salad, ..	5	246
The Pizza B..	Italian, Pizza, Beverag..	5	10,523

no. of restaurants in each city



Key Insights Recap:

- American and North Indian cuisines dominate, with American cuisine being the most expensive on average.



- Restaurants with higher votes and ratings serve a diverse set of cuisines, suggesting that variety can attract more customers.

Top 10 restaurants

Name	Cuisines	Avg. Rate	Votes
AB's - Absol..	European, Mediterran..	5	86,418
Asia Kitche..	Asian, Chinese, Thai, ..	5	42,273
Barbecue b..	BBQ	5	2,683
Belgian Wa..	Desserts	5	24,882
Byg Brewsk..	Continental, North Ind..	5	99,531
Flechazo	Asian, Mediterranean,..	5	29,956
O.G. Variar ..	Bakery, Desserts	5	2,317
Punjab Grill	North Indian	5	1,822
	North Indian, Mughlai	5	7,838
SantÃ ◦ Â ◦ ..	Healthy Food, Salad, ..	5	246
The Pizza B..	Italian, Pizza, Beverag..	5	10,523

Key Insights Recap:

Nightlife and pubs have the highest average costs, aligning with their premium positioning

Koramangala blocks(well-known neighborhood located in Bengaluru) have a dense concentration of restaurants, indicating intense competition in those areas

Recommendations & Business Implications:

• For businesses: Restaurants in highly competitive areas like Bengaluru should focus on differentiating their services to attract customers.

• Adding Services: Restaurants that don’t offer table bookings might benefit from expanding their offerings to improve customer engagement.

• Nightlife Options: Cities with a lack of nightlife or buffet options could introduce these formats to attract higher-spending customers.

• Location Strategy: Businesses expanding to Koramangala should plan for competitive strategies to stand out in a saturated market.

• Cuisine Focus: Focusing on popular cuisines like North Indian or introducing a unique cuisine can attract more customers.

Thank you for your attention! Do you have any questions about our dashboards, insights, or recommendations?