COGNIFYZ DATA SCIENCE INTERNSHIP

LEVEL 3

About Level 3

Level 3 of the Cognifyz Data Science Internship focuses on three key areas:

1. **Predictive Modeling**
2. **Customer Preference Analysis**
3. **Data Visualization**

Task 1: Predictive Modeling

The goal was to develop a regression model to predict a restaurant's aggregate rating based on available features. The steps included:

- Splitting the dataset into training and testing sets.
- Evaluating model performance using appropriate metrics.
- Experimenting with various algorithms such as Linear Regression, Decision Trees, and Random Forest to compare their performance.

Task 2: Customer Preference Analysis

The objective was to analyze the relationship between restaurant ratings and cuisine types. Key tasks included:

- Identifying the most popular cuisines based on the number of customer votes.
- Investigating whether certain cuisines tend to receive higher ratings.

Task 3: Data Visualization

The final task involved creating visualizations to represent the data. Specific goals included:

- Displaying rating distributions through charts (e.g., histograms, bar plots).
- Comparing average ratings across different cuisines or cities.
- Visualizing the relationship between features and the target variable (aggregate rating).

```
#importig all the necessary libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
```

```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error, r2_score, accuracy_score, confusion_matrix
from sklearn.svm import SVR
```

```
#accessing the file
df = pd.read_csv("/content/Dataset .csv")
df.head()
```

	Restaurant ID	Restaurant Name	Country Code	City	Address	Locality	Locality Verbose	Longitude	Latitude	Cuisines	...	Currency	T... book...
0	6317637	Le Petit Souffle	162	Makati City	Third Floor, Century City Mall, Kalayaan Avenu...	Century City Mall, Poblacion, Makati City	Century City Mall, Poblacion, Makati City, Mak...	121.027535	14.565443	French, Japanese, Desserts	...	Botswana Pula(P)	
1	6304287	Izakaya Kikufuji	162	Makati City	Little Tokyo, 2277 Chino Roces Avenue, Legaspi...	Little Tokyo, Legaspi Village, Makati City	Little Tokyo, Legaspi Village, Makati City, Ma...	121.014101	14.553708	Japanese	...	Botswana Pula(P)	
2	6300002	Heat - Edsa Shangri-La	162	Mandaluyong City	Edsa Shangri-La, 1 Garden Way, Ortigas, Mandal...	Edsa Shangri-La, Ortigas, Mandaluyong City	Edsa Shangri-La, Ortigas, Mandaluyong City, Ma...	121.056831	14.581404	Seafood, Asian, Filipino, Indian	...	Botswana Pula(P)	
3	6318506	Ooma	162	Mandaluyong City	Third Floor, Mega Fashion Hall, SM Megamall, O...	SM Megamall, Ortigas, Mandaluyong City	SM Megamall, Ortigas, Mandaluyong City, Mandal...	121.056475	14.585318	Japanese, Sushi	...	Botswana Pula(P)	
4	6314302	Sambo Kojin	162	Mandaluyong City	Third Floor, Mega Atrium, SM Megamall, Ortigas...	SM Megamall, Ortigas, Mandaluyong City	SM Megamall, Ortigas, Mandaluyong City, Mandal...	121.057508	14.584450	Japanese, Korean	...	Botswana Pula(P)	

5 rows × 21 columns

```
#checking for null values
df.isnull().sum()
```

	0
Restaurant ID	0
Restaurant Name	0
Country Code	0
City	0
Address	0
Locality	0
Locality Verbose	0
Longitude	0
Latitude	0
Cuisines	9
Average Cost for two	0
Currency	0
Has Table booking	0
Has Online delivery	0
Is delivering now	0
Switch to order menu	0
Price range	0
Aggregate rating	0
Rating color	0
Rating text	0
Votes	0

dtype: int64

```
1 df.describe()
```

	Restaurant ID	Country Code	Longitude	Latitude	Average Cost for two	Price range	Aggregate rating	Votes
count	9.551000e+03	9551.000000	9551.000000	9551.000000	9551.000000	9551.000000	9551.000000	9551.000000
mean	9.051128e+06	18.365616	64.126574	25.854381	1199.210763	1.804837	2.666370	156.909748
std	8.791521e+06	56.750546	41.467058	11.007935	16121.183073	0.905609	1.516378	430.169145
min	5.300000e+01	1.000000	-157.948486	-41.330428	0.000000	1.000000	0.000000	0.000000
25%	3.019625e+05	1.000000	77.081343	28.478713	250.000000	1.000000	2.500000	5.000000
50%	6.004089e+06	1.000000	77.191964	28.570469	400.000000	2.000000	3.200000	31.000000
75%	1.835229e+07	1.000000	77.282006	28.642758	700.000000	2.000000	3.700000	131.000000
max	1.850065e+07	216.000000	174.832089	55.976980	800000.000000	4.000000	4.900000	10934.000000

```
1 df.columns
```

```
Index(['Restaurant ID', 'Restaurant Name', 'Country 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')
```

```
1 #filling the missing values
2 rest_data = df['Cuisines'].fillna('Unknown', inplace=True)
```

```
1 #re-checking for null values
2 df.isnull().sum()
```

	0
Restaurant ID	0
Restaurant Name	0
Country Code	0
City	0
Address	0
Locality	0
Locality Verbose	0
Longitude	0
Latitude	0
Cuisines	0
Average Cost for two	0
Currency	0
Has Table booking	0
Has Online delivery	0
Is delivering now	0
Switch to order menu	0
Price range	0
Aggregate rating	0
Rating color	0
Rating text	0
Votes	0

dtype: int64

Task 1: Predictive Modeling

predict the aggregate rating

Experiment with different algorithms (e.g., linear regression, decision trees, random forest) and compare their performance.

```
# Select features and target variable
features = ['Average Cost for two', 'Votes']
target = 'Aggregate rating'
```

```
# Split the data into training and testing sets
X = df[features]
y = df[target]
X_train,X_test,Y_train,Y_test = train_test_split(X,y,test_size=0.2,random_state=42)
```

```
# Function to train and evaluate a regression model
def train_and_evaluate_model(model):
    model.fit(X_train, Y_train)
    y_pred = model.predict(X_test)
    mse = mean_squared_error(Y_test, y_pred)
    r2 = r2_score(Y_test, y_pred)
    return mse, r2
```

Linear Regression

```
# Training and evaluating Linear Regression model
linear_regression = LinearRegression()
mse_lr, r2_lr = train_and_evaluate_model(linear_regression)
```

```
#scores of linear model
print("Linear Regression:")
print(f"Mean Squared Error (MSE): {mse_lr:.6f}"
```

```
print(f"R-squared (R2): {r2_lr:.6f}")
```

```
Linear Regression:
Mean Squared Error (MSE): 2.055716
R-squared (R2): 0.096829
```

Decision trees

```
# Training and evaluating decision tree model
dec_model = DecisionTreeRegressor(random_state=42)
mse_dec, r2_dec = train_and_evaluate_model(dec_model)
```

```
#scores of decision tree model
print("\nDecision Tree:")
print(f"Mean Squared Error (MSE): {mse_dec:.6f}")
print(f"R-squared (R2): {r2_dec:.6f}")
```

```
Decision Tree:
Mean Squared Error (MSE): 0.222755
R-squared (R2): 0.902133
```

Random forest

```
# Training and evaluating random forest model
random_forest = RandomForestRegressor(random_state=42)
mse_rf, r2_rf = train_and_evaluate_model(random_forest)
```

```
#scores of random forest model
print("\nRandom Forest:")
print(f"Mean Squared Error (MSE): {mse_rf:.6f}")
print(f"R-squared (R2): {r2_rf:.6f}")

```

```
Random Forest:
Mean Squared Error (MSE): 0.159152
R-squared (R2): 0.930077
```

Support Vector Regression

```
# Training and evaluating Support Vector Regression model
svr = SVR()
mse_svr, r2_svr = train_and_evaluate_model(svr)
```

```
#scores of support vector machine model
print("\nSupport Vector Regression:")
print(f"Mean Squared Error (MSE): {mse_svr:.6f}")
print(f"R-squared (R2): {r2_svr:.6f}")

```

```
Support Vector Regression:
Mean Squared Error (MSE): 2.253043
R-squared (R2): 0.010134
```

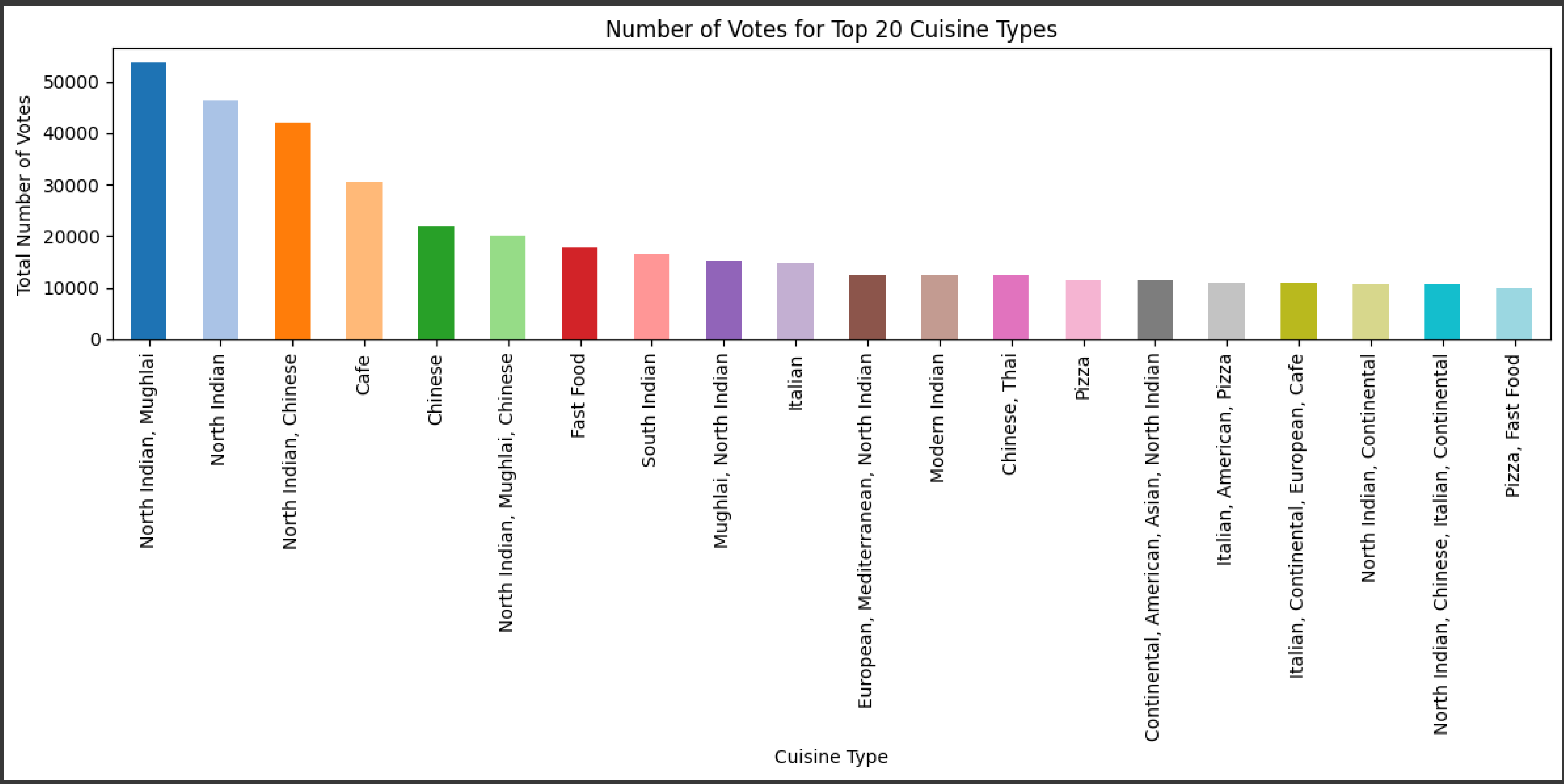
Task 2: Customer Preference Analysis

```
# Identify the most popular cuisines based on the number of customer votes.
cuisines_votes = df.groupby('Cuisines')['Votes'].sum().sort_values(ascending=False)
high_cuisines_votes=cuisines_votes.head(20)
print("Most popular cuisines based on votes:\n", high_cuisines_votes)
```

```
Most popular cuisines based on votes:
 Cuisines
North Indian, Mughlai                    53747
North Indian                             46241
North Indian, Chinese                    42012
Cafe                                     30657
Chinese                                  21925
North Indian, Mughlai, Chinese           20115
Fast Food                                17852
```

```
South Indian                               16433
Mughlai, North Indian                      15275
Italian                                    14799
European, Mediterranean, North Indian      12541
Modern Indian                              12355
Chinese, Thai                              12354
Pizza                                      11537
Continental, American, Asian, North Indian 11404
Italian, American, Pizza                   10934
Italian, Continental, European, Cafe       10853
North Indian, Continental                  10760
North Indian, Chinese, Italian, Continental 10744
Pizza, Fast Food                            9953
Name: Votes, dtype: int64
```

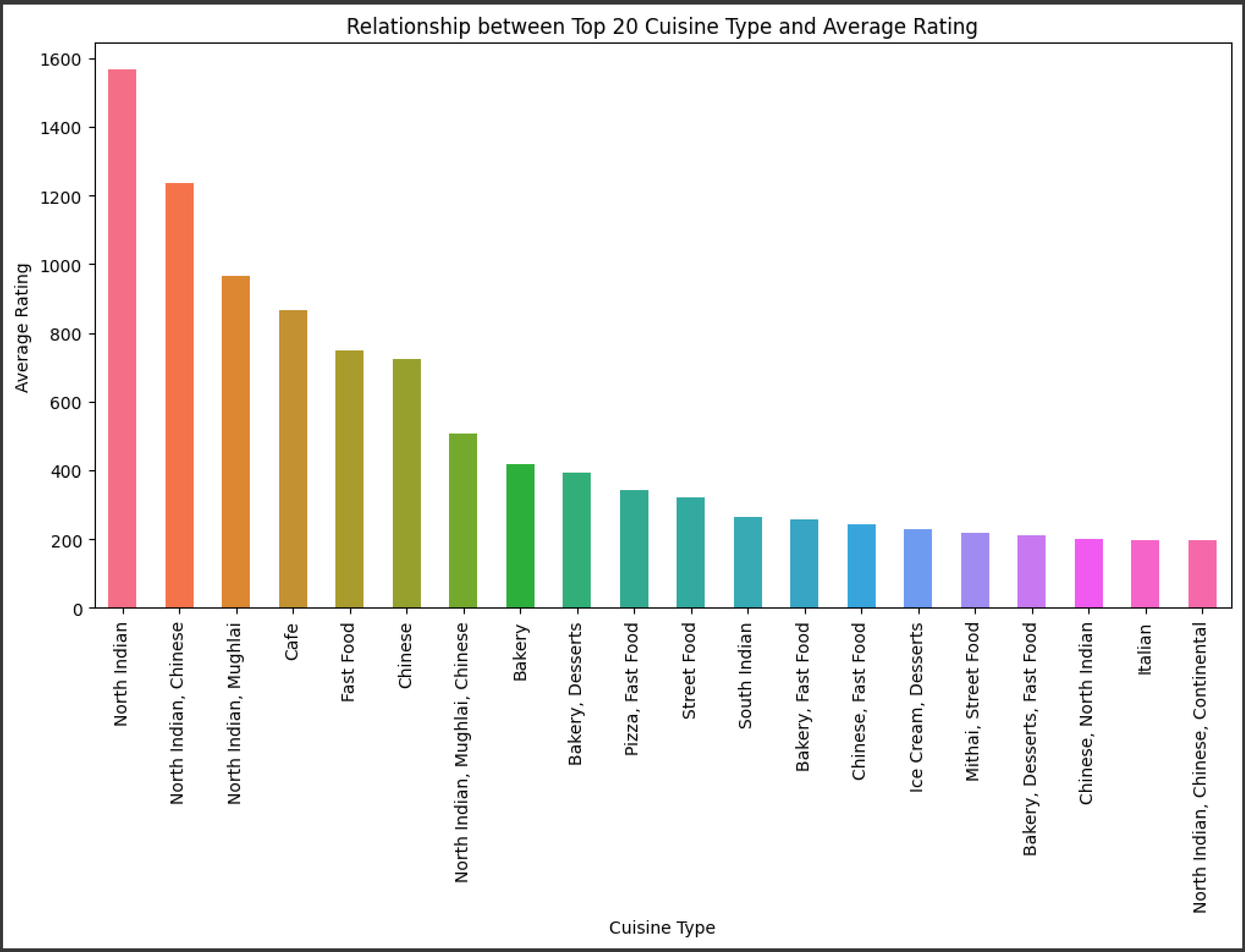
```
# Plot the relationship between Top 20 cuisine type and the number of votes
plt.figure(figsize=(12, 6))
colors = sns.color_palette("tab20", n_colors=len(high_cuisines_votes))
high_cuisines_votes.plot(kind='bar',color=colors)
plt.title('Number of Votes for Top 20 Cuisine Types')
plt.xlabel('Cuisine Type')
plt.ylabel('Total Number of Votes')
plt.tight_layout()
plt.show()
```




```
# Investigating whether certain cuisines tend to receive higher ratings.
cuisines_ratings = df.groupby('Cuisines')['Aggregate rating'].sum().sort_values(ascending=False)
high_cuisines_ratings = cuisines_ratings.head(20)
print("Most popular cuisines based on Aggregate rating:\n", high_cuisines_ratings)
```

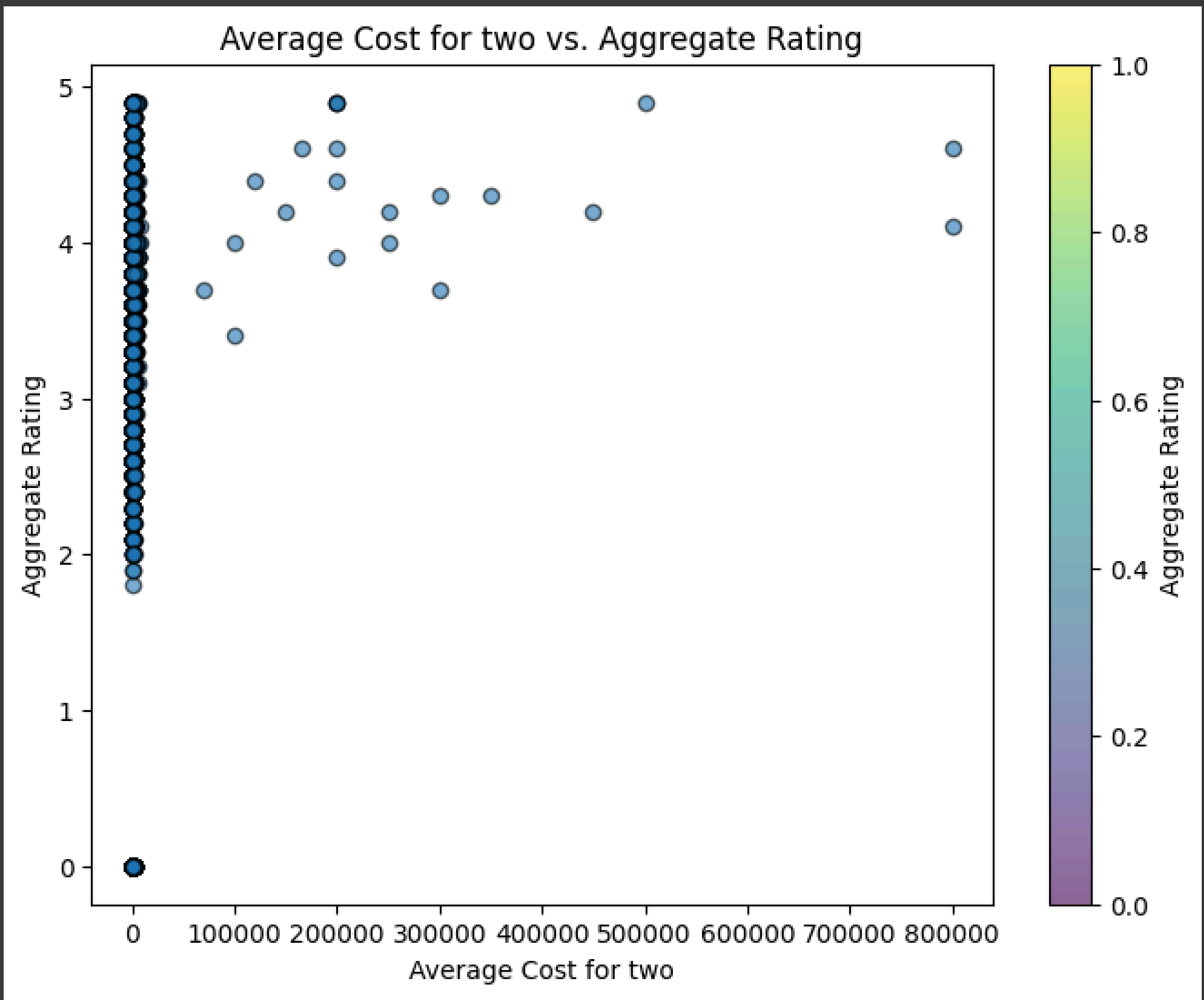
```
Most popular cuisines based on Aggregate rating:
 Cuisines
North Indian                       1565.3
North Indian, Chinese              1237.5
North Indian, Mughlai               964.8
Cafe                                864.4
Fast Food                           749.9
Chinese                             722.9
North Indian, Mughlai, Chinese      506.0
Bakery                              419.5
Bakery, Desserts                    394.0
Pizza, Fast Food                    344.4
Street Food                         322.1
South Indian                        265.5
Bakery, Fast Food                   259.1
Chinese, Fast Food                  242.5
Ice Cream, Desserts                 230.3
Mithai, Street Food                 219.6
Bakery, Desserts, Fast Food         212.1
Chinese, North Indian               199.9
Italian                             197.5
North Indian, Chinese, Continental  197.4
Name: Aggregate rating, dtype: float64
```

```
# Plot the relationship between cuisine type and average rating
plt.figure(figsize=(12, 6))
colors = sns.color_palette("husl", n_colors=len(high_cuisines_ratings))
high_cuisines_ratings.plot(kind='bar',color=colors)
plt.xlabel('Cuisine Type')
plt.ylabel('Average Rating')
plt.title('Relationship between Top 20 Cuisine Type and Average Rating')
plt.show()
```

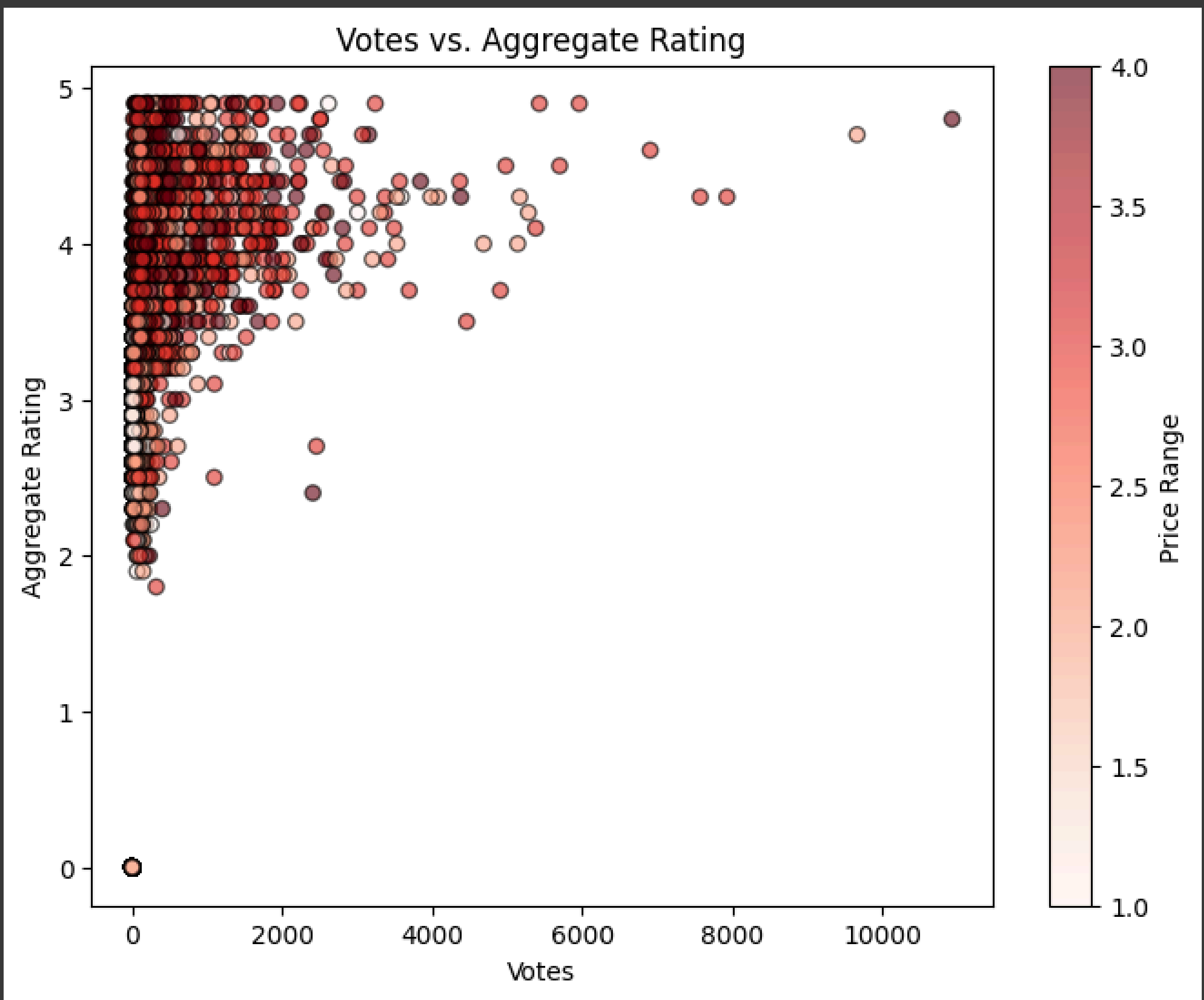


Task 3: Data Visualization

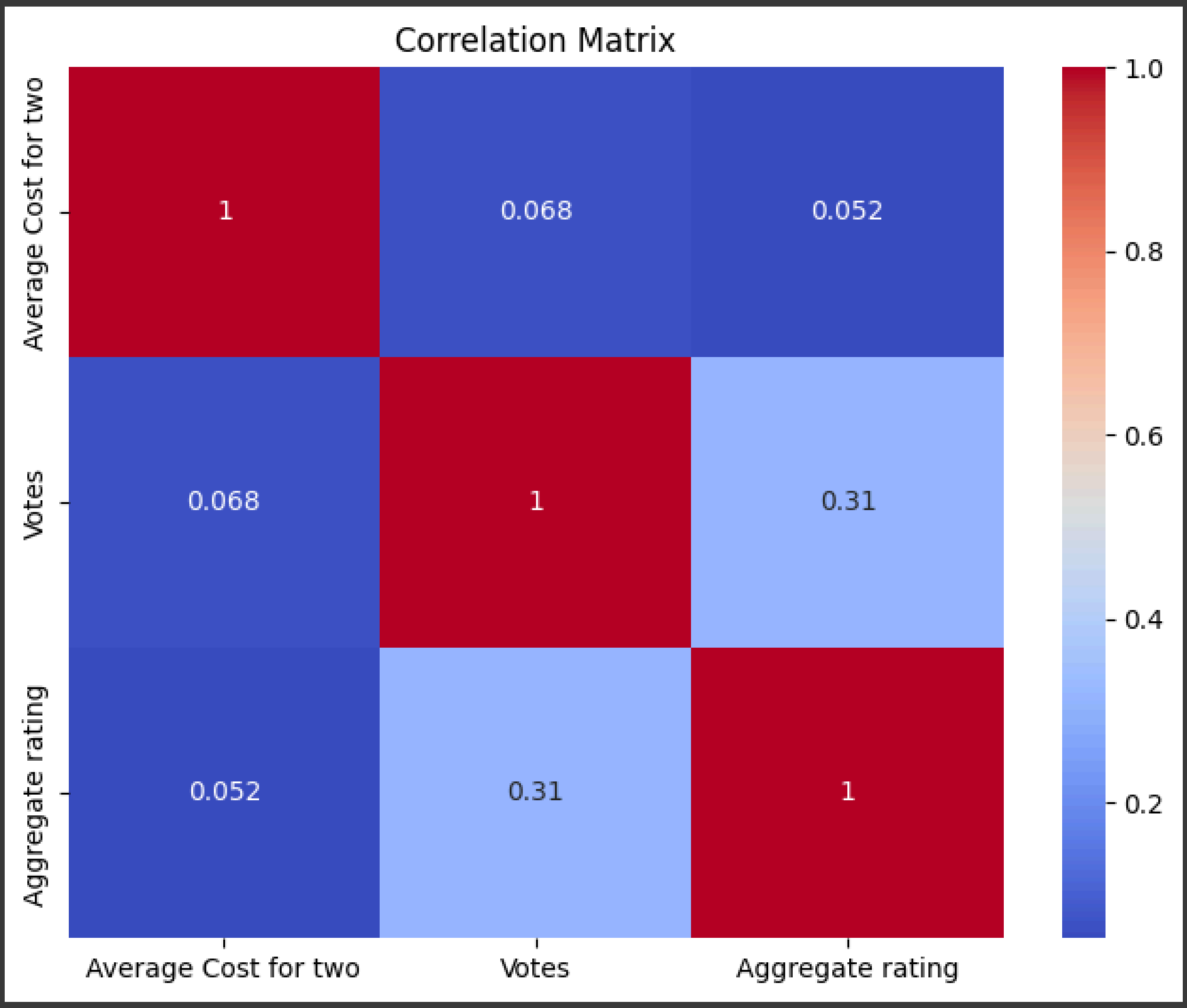
```
# Scatter plot of Average Cost for two vs. Aggregate rating
plt.figure(figsize=(8, 6))
plt.scatter(df['Average Cost for two'], df['Aggregate rating'], cmap='viridis', alpha=0.6, edgecolor='k')
plt.title('Average Cost for two vs. Aggregate Rating')
plt.xlabel('Average Cost for two')
plt.ylabel('Aggregate Rating')
plt.colorbar(label='Aggregate Rating')
plt.show()

```




```python
import matplotlib.pyplot as plt

# Scatter plot of Votes vs. Aggregate rating
plt.figure(figsize=(8, 6))
plt.scatter(df['Votes'], df['Aggregate rating'], c=df['Price range'], cmap='Reds', alpha=0.6, edgecolor='k')
plt.title('Votes vs. Aggregate Rating')
plt.xlabel('Votes')
plt.ylabel('Aggregate Rating')
plt.colorbar(label='Price Range')
plt.show()

```




```python
# Correlation matrix heatmap
correlation_matrix = df[['Average Cost for two', 'Votes',
                         'Aggregate rating']].corr()
plt.figure(figsize=(8, 6))
sns.heatmap(correlation_matrix,annot=True, cmap='coolwarm')
plt.title('Correlation Matrix')
plt.show()
```

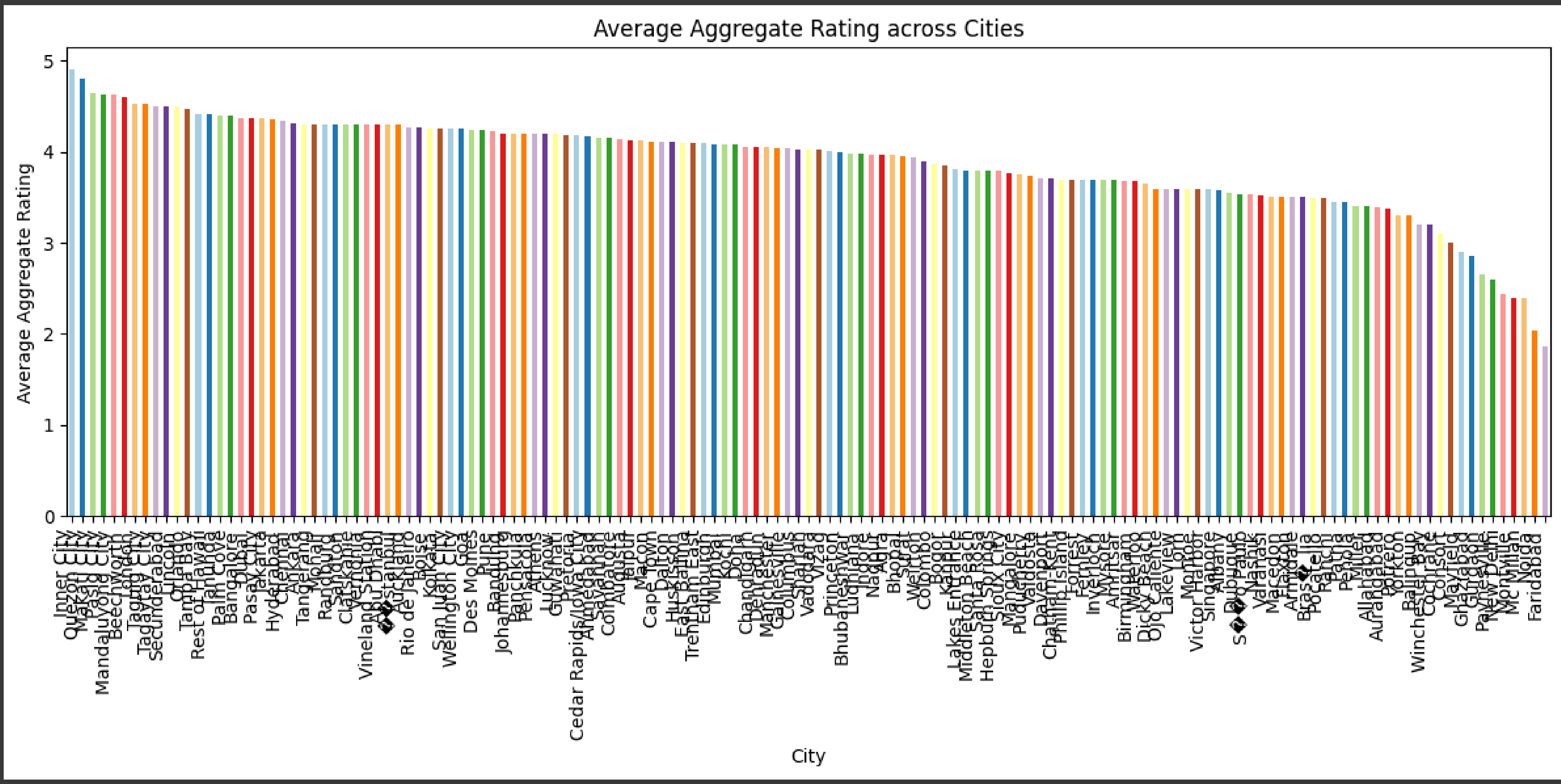



```
# Comparing average ratings across different cities
city_avg_ratings = df.groupby('City')['Aggregate rating'].mean().sort_values(ascending=False)
city_avg_ratings.head(10)
```

City	Aggregate rating
Inner City	4.900000
Quezon City	4.800000
Makati City	4.650000
Pasig City	4.633333
Mandaluyong City	4.625000
Beechworth	4.600000
London	4.535000
Taguig City	4.525000
Tagaytay City	4.500000
Secunderabad	4.500000

dtype: float64

```
# Plot the average ratings across cities
plt.figure(figsize=(12, 6))
colors = sns.color_palette("Paired", n_colors=len(city_avg_ratings))
city_avg_ratings.plot(kind='bar', color=colors)
plt.title('Average Aggregate Rating across Cities')
plt.xlabel('City')
plt.ylabel('Average Aggregate Rating')
plt.xticks(rotation=90, ha='right')
plt.tight_layout()
plt.show()

```



Results

Task 1: Predictive Modeling

Four regression models were developed:

1. **Linear Regression**,
2. **Decision Tree**,
3. **Random Forest**,
4. **Support Vector Machine (SVM)**.

The models were evaluated using Root Mean Square Error (RMSE) and R-squared metrics.

The results are summarized below:

Model	RMSE	R-squared
Linear Regression	2.055716	0.096829
Decision Tree	0.222755	0.902133
Random Forest	0.159152	0.930077
Support Vector Machine	2.253043	0.1105025

From the table, it is evident that the **Random Forest** model performed the best, with the lowest RMSE and highest R-squared, making it the preferred choice for this task.

Task 2: Customer Preference Analysis

- The analysis revealed that **North Indian**, **Mughlai**, and **Chinese** cuisines were the most popular, based on customer votes.
- Additionally, several cuisines, such as **American**, **BBQ**, **Sandwich**, **Burger**, **Grill**, **Caribbean**, **Seafood**, and **Coffee and Tea**, consistently received an average rating of 4.9.

Task 3: Data Visualization

- The majority of restaurant ratings fell between **3 and 4**.
- On a city-level analysis, **Inner City** had the highest average ratings, followed by **Quezon City**, **Makati City**, **Pasig City**, **Mandaluyong City**, and **Beechworth**.
- A correlation analysis indicated that aggregate ratings were positively associated with several features, including **Votes**, **Price Range**, **Has Table Booking**, and **Has Online Delivery**.
- Among these, **Price Range** had the strongest positive correlation.

Conclusion

This project highlighted the critical role of predictive modeling, customer preference analysis, and data visualization in uncovering valuable insights and facilitating strategic decision-making.

The analysis of customer preferences offered deeper understanding of target audience preferences, while data visualization techniques proved

Producing report of this project

```
!pip install ydata_profiling
```

```
Collecting phik<0.13,>=0.11.1 (from ydata_profiling)
  Downloading phik-0.12.4-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (5.6 kB)
Requirement already satisfied: requests<3,>=2.24.0 in /usr/local/lib/python3.10/dist-packages (from ydata-profiling) (2.32.3)
```