Task 1: Predictive Modeling

In [42]:
```
import pandas as pd
import numpy as np
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
from sklearn.preprocessing import LabelEncoder
```

In [43]:
```
df = pd.read_csv('Dataset .csv')
```

In [45]:
```
df = df.drop(columns=['Restaurant ID', 'Restaurant Name', 'Address', 'Locality', 'L
                      'Longitude', 'Latitude', 'Switch to order menu'])
```

In [46]:
```
le = LabelEncoder()
categorical_cols = ['Country Code', 'City', 'Cuisines', 'Currency', 'Has Table book
                    'Has Online delivery', 'Is delivering now', 'Rating color', 'Ra
```

In [47]:
```
for col in categorical_cols:
    df[col] = le.fit_transform(df[col])
```

In [48]:
```
X = df.drop('Aggregate rating', axis=1)
y = df['Aggregate rating']
```

In [49]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_sta
```

In [50]:
```
linear_reg = LinearRegression()
decision_tree = DecisionTreeRegressor(random_state=42)
random_forest = RandomForestRegressor(random_state=42)
```

In [51]:
```
linear_reg.fit(X_train, y_train)
y_pred_lr = linear_reg.predict(X_test)
```

In [52]:
```
decision_tree.fit(X_train, y_train)
y_pred_dt = decision_tree.predict(X_test)
```

In [54]:
```
random_forest.fit(X_train, y_train)
y_pred_rf = random_forest.predict(X_test)
```

In [55]:
```
def evaluate_model(y_test, y_pred, model_name):
    print(f"{model_name} Performance:")
    print(f"Mean Squared Error (MSE): {mean_squared_error(y_test, y_pred):.2f}")
    print(f"R-squared (R2): {r2_score(y_test, y_pred):.2f}")
    print("-" * 30)
```

In [56]:
```
evaluate_model(y_test, y_pred_lr, "Linear Regression")
```

```
Linear Regression Performance:
Mean Squared Error (MSE): 1.30
R-squared (R2): 0.43
------------------------------
```

In [57]:
```
evaluate_model(y_test, y_pred_dt, "Decision Tree")
```

```
Decision Tree Performance:
Mean Squared Error (MSE): 0.06
R-squared (R2): 0.97
------------------------------
```

In [58]:
```
evaluate_model(y_test, y_pred_rf, "Random Forest")
```

```
Random Forest Performance:
Mean Squared Error (MSE): 0.03
R-squared (R2): 0.99
------------------------------
```

Task 2: Customer Preference Analysis

In [59]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [60]:
```
df = pd.read_csv('Dataset .csv')
```

In [61]:
```
#  1: Analyze the relationship between the type of cuisine and the restaurant's rat
```

In [62]:
```
df['Cuisines'] = df['Cuisines'].fillna('Unknown')
df['Cuisines'] = df['Cuisines'].str.split(', ')
```

In [63]:
```
df_cuisine = df.explode('Cuisines')
```

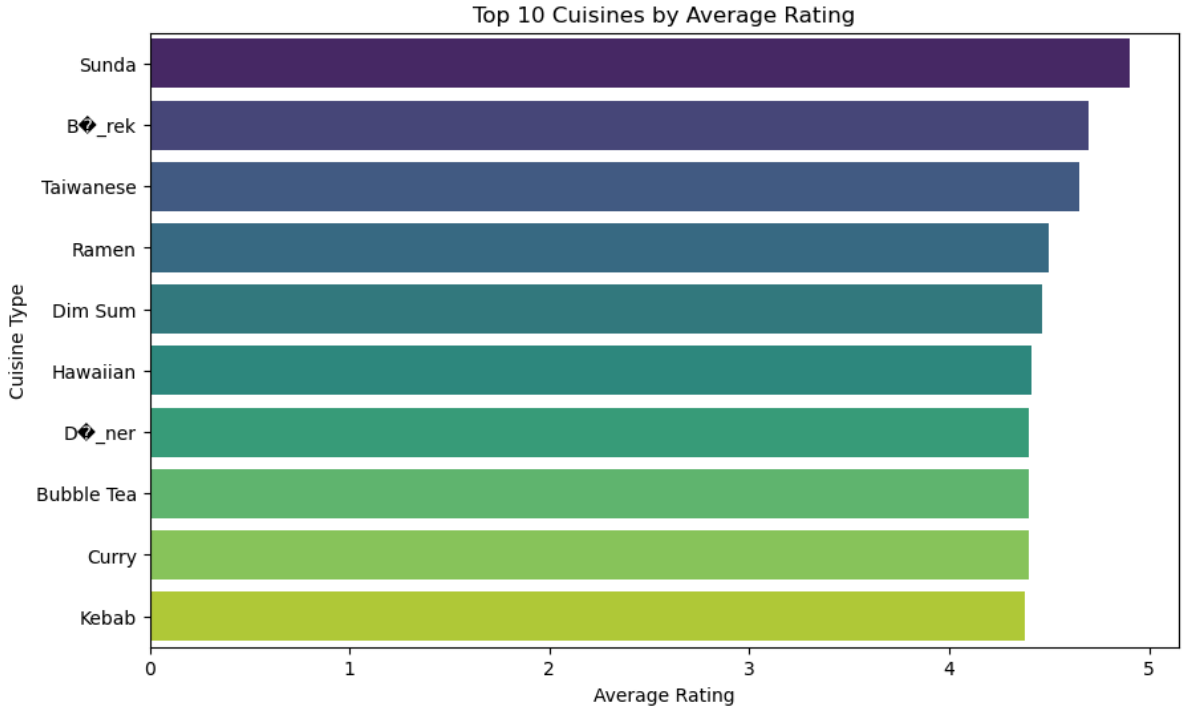
In [64]:
```
cuisine_rating = df_cuisine.groupby('Cuisines')['Aggregate rating'].mean().reset_ir
```

In [65]:
```
cuisine_rating = cuisine_rating.sort_values(by='Aggregate rating', ascending=False)
```

In [66]:
```
print("Top 10 cuisines by average rating:")
print(cuisine_rating.head(10))
```

```
Top 10 cuisines by average rating:
       Cuisines  Aggregate rating
130       Sunda          4.900000
26       B�_rek          4.700000
132   Taiwanese          4.650000
112       Ramen          4.500000
43      Dim Sum          4.466667
61     Hawaiian          4.412500
47       D�_ner          4.400000
23   Bubble Tea          4.400000
40        Curry          4.400000
75        Kebab          4.380000
```

In [67]:
```
plt.figure(figsize=(10,6))
sns.barplot(x='Aggregate rating', y='Cuisines', data=cuisine_rating.head(10), palet
plt.title('Top 10 Cuisines by Average Rating')
plt.xlabel('Average Rating')
plt.ylabel('Cuisine Type')
plt.show()
```



In [68]:
```
# 2: Identify the most popular cuisines based on the number of votes
```

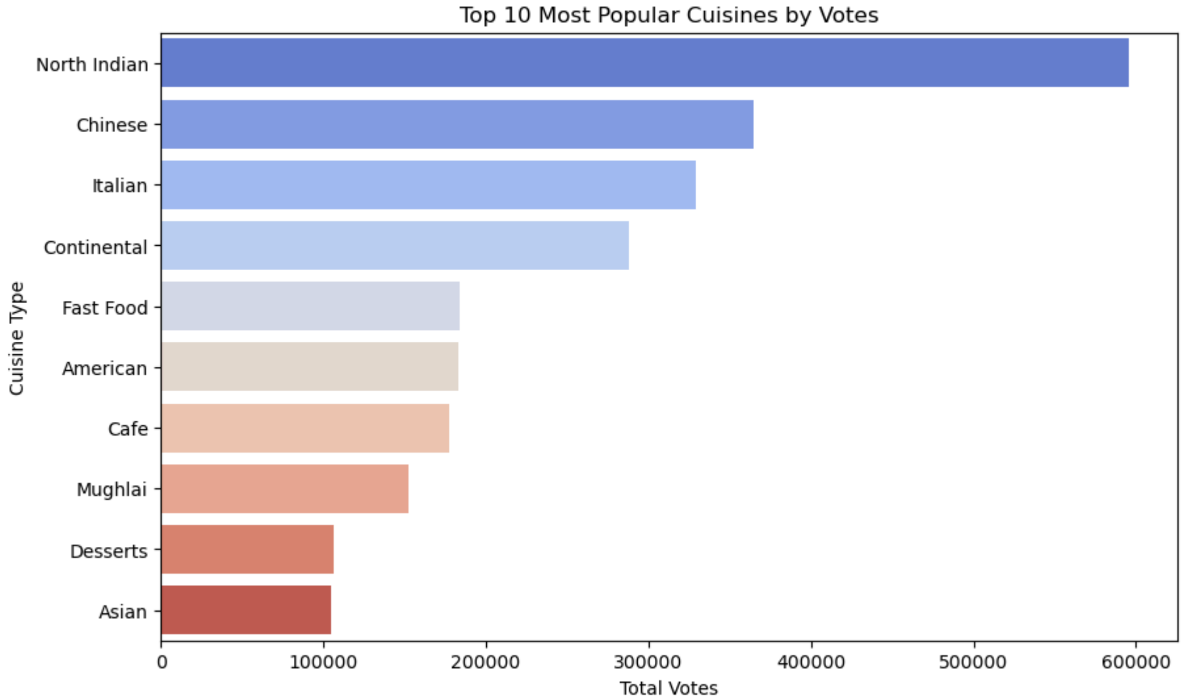
In [69]:
```
cuisine_votes = df_cuisine.groupby('Cuisines')['Votes'].sum().reset_index()
```

In [70]:
```
cuisine_votes = cuisine_votes.sort_values(by='Votes', ascending=False)
```

In [71]:
```
print("Top 10 most popular cuisines by votes:")
print(cuisine_votes.head(10))
```

```
Top 10 most popular cuisines by votes:
         Cuisines   Votes
100  North Indian  595981
34        Chinese  364351
70        Italian  329265
37    Continental  288255
49      Fast Food  184058
2        American  183117
27           Cafe  177568
95        Mughlai  151946
42       Desserts  105889
7           Asian  104303
```

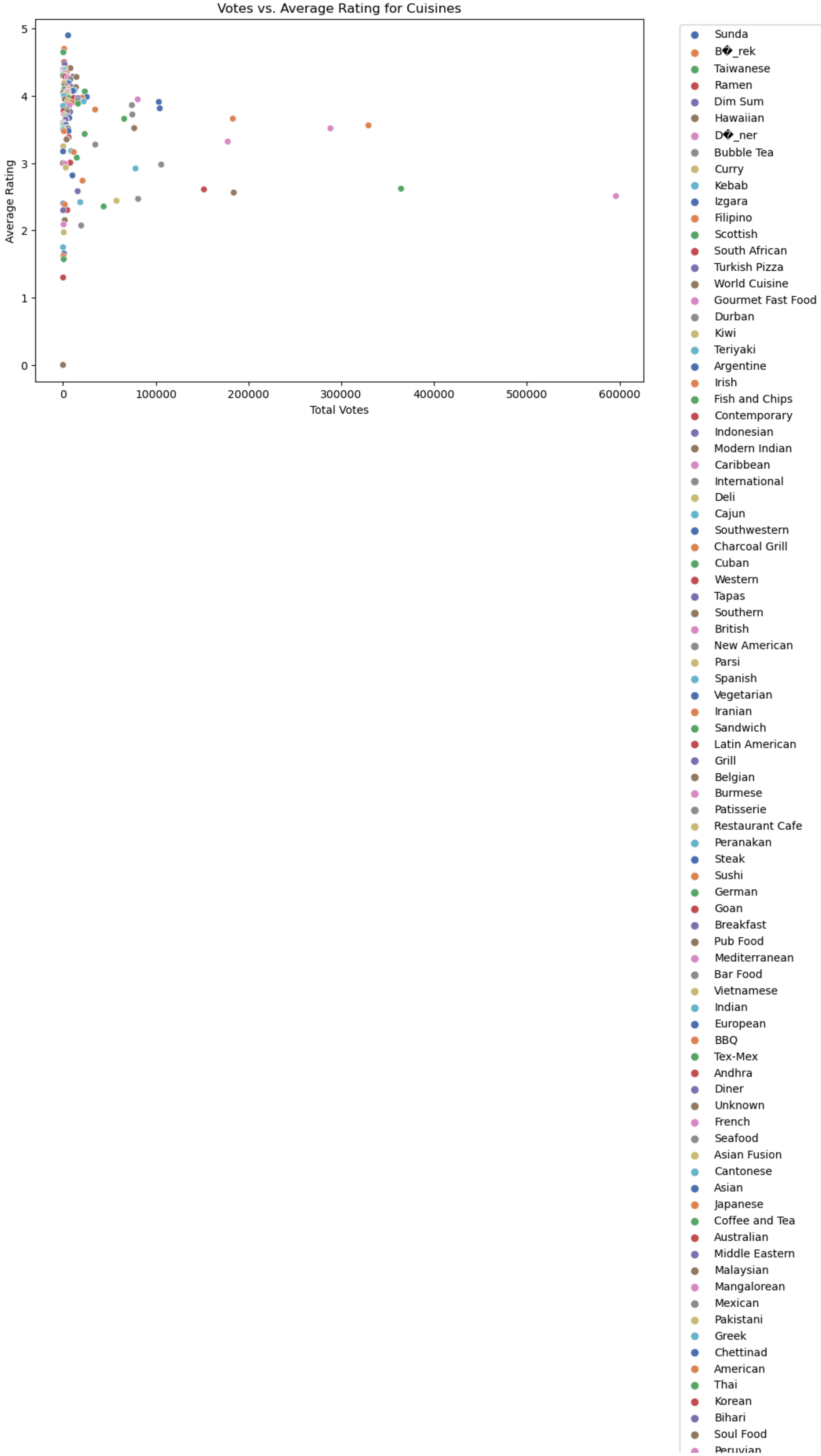
In [72]:
```
plt.figure(figsize=(10,6))
sns.barplot(x='Votes', y='Cuisines', data=cuisine_votes.head(10), palette='coolwarm
plt.title('Top 10 Most Popular Cuisines by Votes')
plt.xlabel('Total Votes')
plt.ylabel('Cuisine Type')
plt.show()
```



In [73]:
```
# 3: Determine if any specific cuisines tend to receive higher ratings
```

In [74]:
```
cuisine_analysis = pd.merge(cuisine_rating, cuisine_votes, on='Cuisines')
```

In [75]:
```
plt.figure(figsize=(10,6))
sns.scatterplot(x='Votes', y='Aggregate rating', hue='Cuisines', data=cuisine_analy
plt.title('Votes vs. Average Rating for Cuisines')
plt.xlabel('Total Votes')
plt.ylabel('Average Rating')
plt.legend(bbox_to_anchor=(1.05, 1), loc='upper left')
plt.show()
```


Votes vs. Average Rating for Cuisines
Average Rating
0
1
2
3
4
5
Total Votes
0
100000
200000
300000
400000
500000
600000
Sunda
B�_rek
Taiwanese
Ramen
Dim Sum
Hawaiian
D�_ner
Bubble Tea
Curry
Kebab
Izgara
Filipino
Scottish
South African
Turkish Pizza
World Cuisine
Gourmet Fast Food
Durban
Kiwi
Teriyaki
Argentine
Irish
Fish and Chips
Contemporary
Indonesian
Modern Indian
Caribbean
International
Deli
Cajun
Southwestern
Charcoal Grill
Cuban
Western
Tapas
Southern
British
New American
Parsi
Spanish
Vegetarian
Iranian
Sandwich
Latin American
Grill
Belgian
Burmese
Patisserie
Restaurant Cafe
Peranakan
Steak
Sushi
German
Goan
Breakfast
Pub Food
Mediterranean
Bar Food
Vietnamese
Indian
European
BBQ
Tex-Mex
Andhra
Diner
Unknown
French
Seafood
Asian Fusion
Cantonese
Asian
Japanese
Coffee and Tea
Australian
Middle Eastern
Malaysian
Mangalorean
Mexican
Pakistani
Greek
Chettinad
American
Thai
Korean
Bihari
Soul Food

In [76]:
```
print("Cuisine Analysis DataFrame:")
print(cuisine_analysis.head(10))
```

```
Cuisine Analysis DataFrame:
     Cuisines  Aggregate rating  Votes
0       Sunda          4.900000   5514
1      B�_rek          4.700000   1305
2   Taiwanese          4.650000    384
3       Ramen          4.500000   1259
4     Dim Sum          4.466667   1755
5    Hawaiian          4.412500   8012
6      D�_ner          4.400000     72
7  Bubble Tea          4.400000    659
8       Curry          4.400000   2059
9       Kebab          4.380000   1536
```

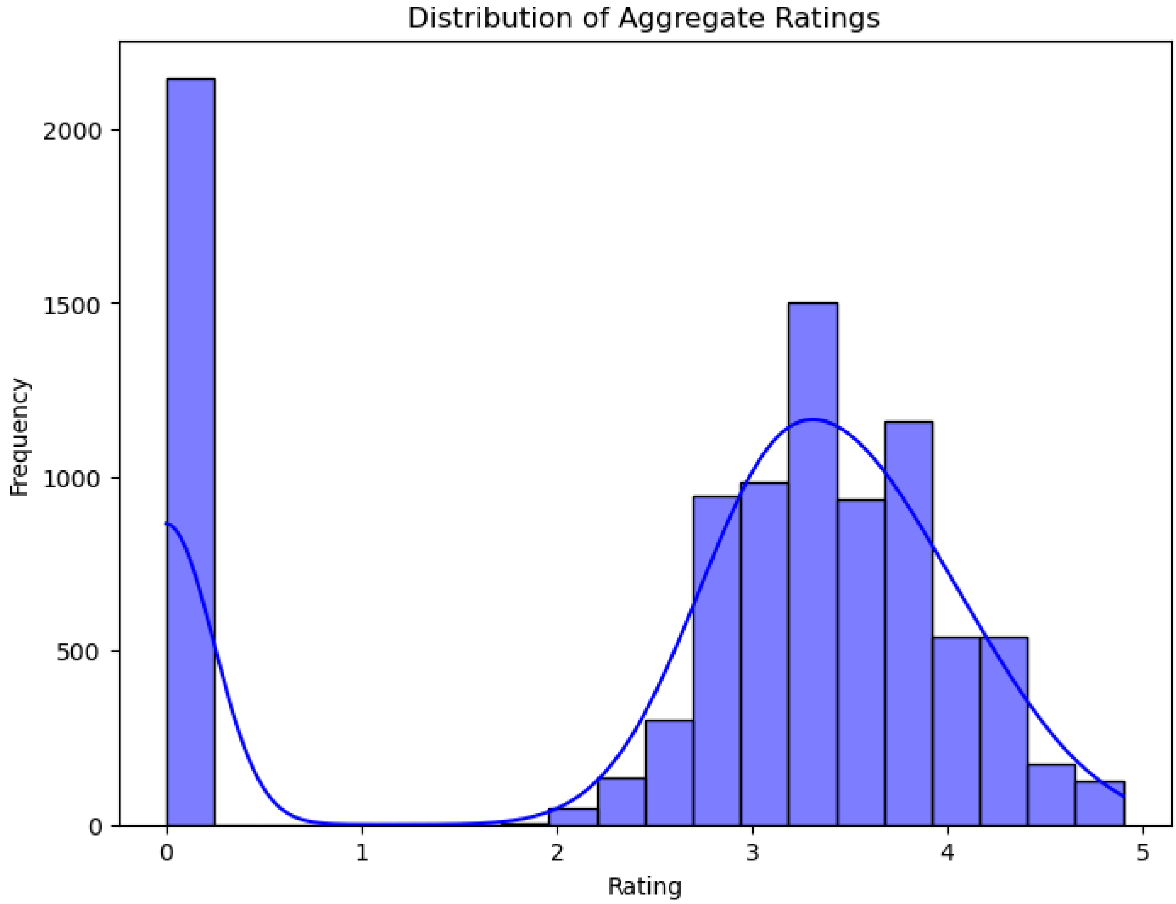
Task 3: Data Visualization

In [77]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

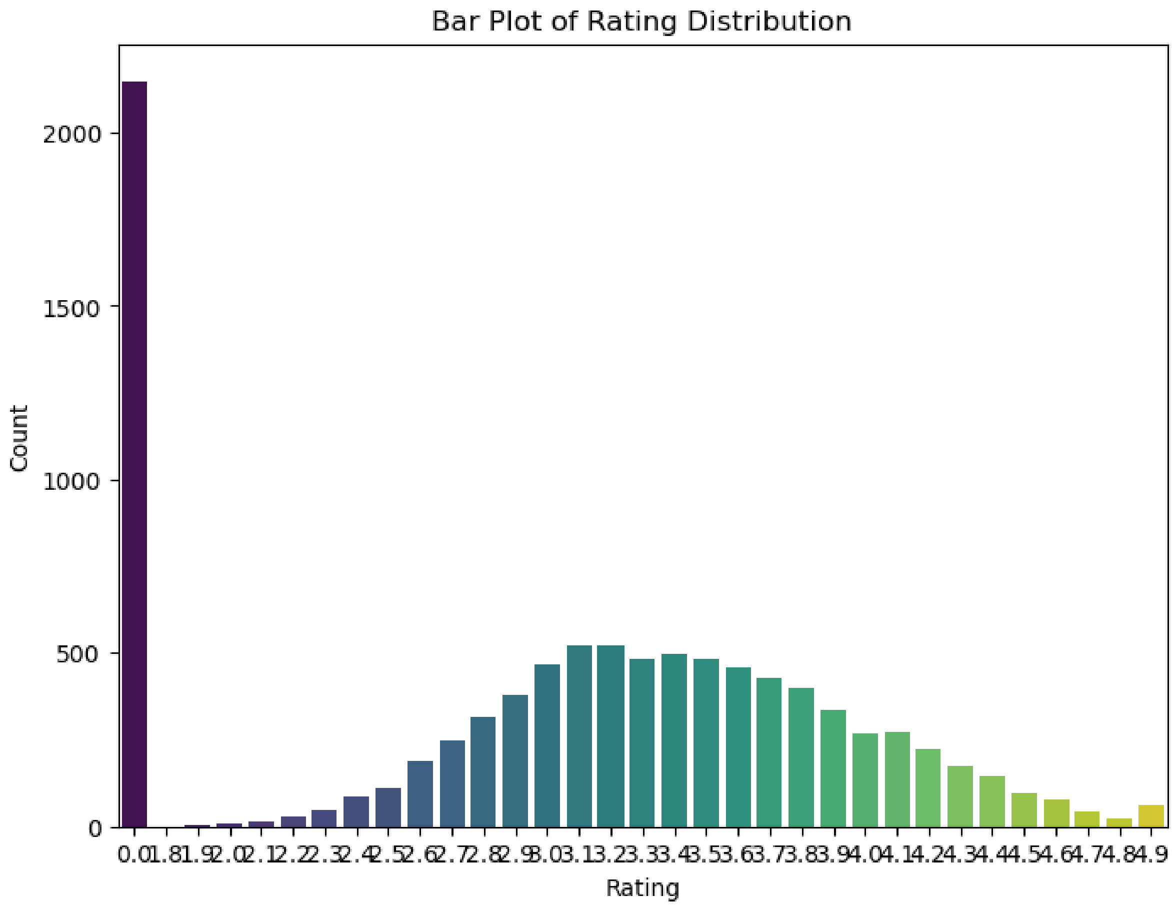
In [78]:
```
df = pd.read_csv('Dataset .csv')
```

In [79]:
```
#1: Visualizing the distribution of ratings
```

In [80]:
```
plt.figure(figsize=(8,6))
sns.histplot(df['Aggregate rating'], bins=20, kde=True, color='blue')
plt.title('Distribution of Aggregate Ratings')
plt.xlabel('Rating')
plt.ylabel('Frequency')
plt.show()
```



In [81]:
```
rating_counts = df['Aggregate rating'].value_counts().sort_index()
plt.figure(figsize=(8,6))
sns.barplot(x=rating_counts.index, y=rating_counts.values, palette='viridis')
plt.title('Bar Plot of Rating Distribution')
plt.xlabel('Rating')
plt.ylabel('Count')
plt.show()
```



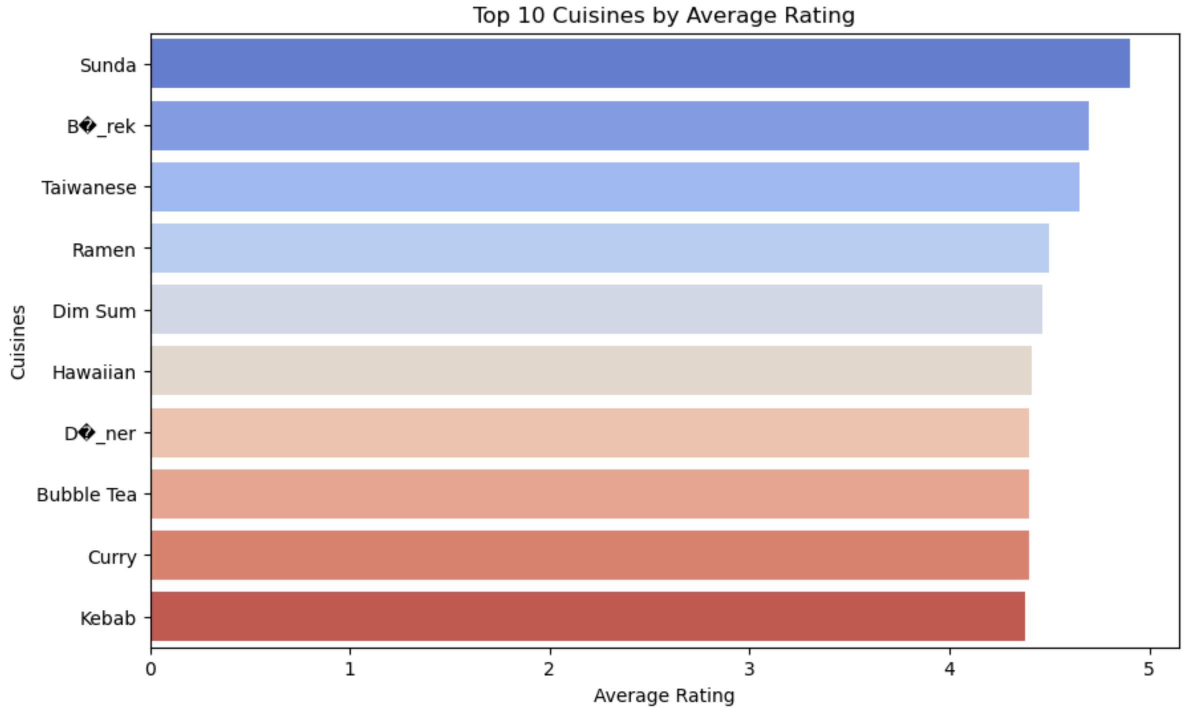
In [82]:
```
# 2: Compare average ratings of different cuisines
```

In [83]:
```
df['Cuisines'] = df['Cuisines'].fillna('Unknown')
df['Cuisines'] = df['Cuisines'].str.split(', ')
```

In [84]:
```
df_cuisine = df.explode('Cuisines')
```

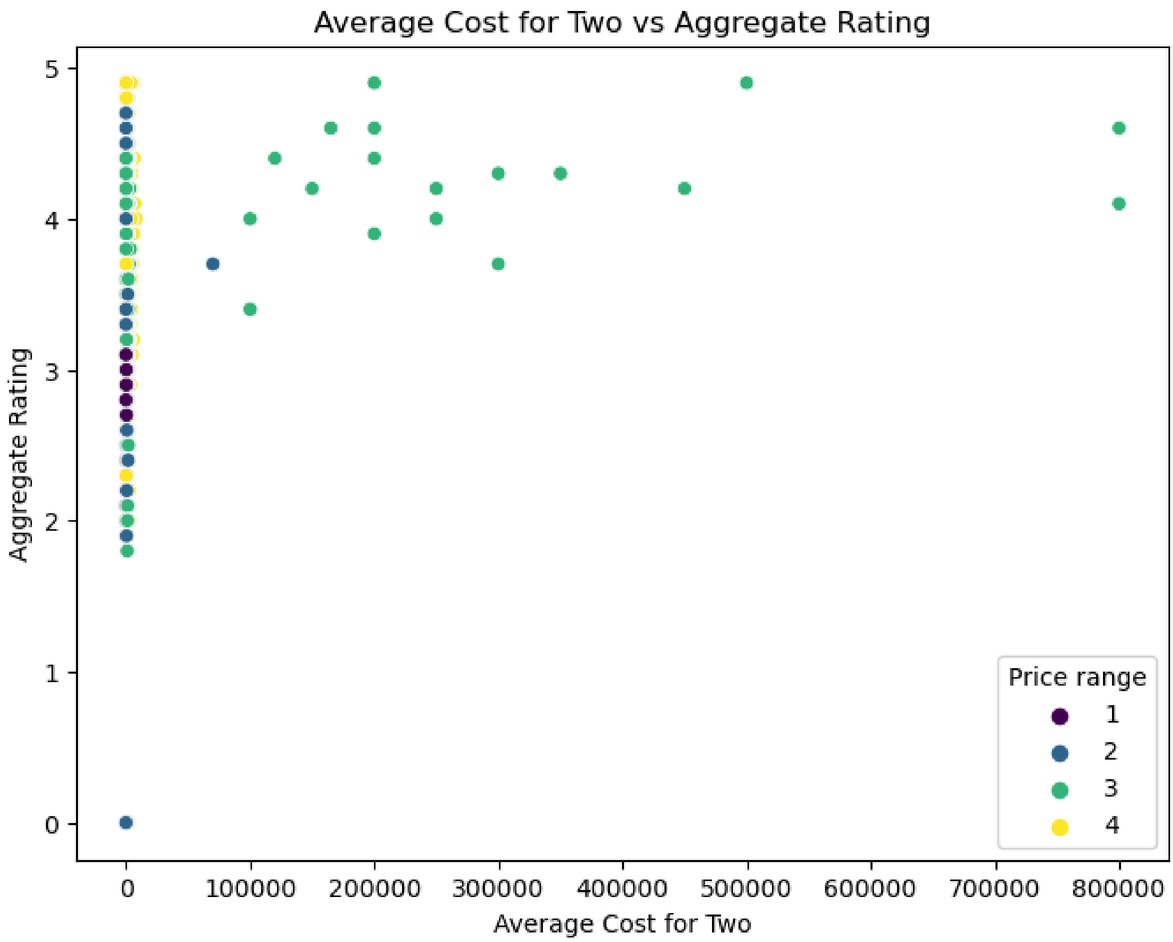
In [85]:
```
avg_cuisine_rating = df_cuisine.groupby('Cuisines')['Aggregate rating'].mean().rese
avg_cuisine_rating = avg_cuisine_rating.sort_values(by='Aggregate rating', ascendir
```

In [86]:
```
plt.figure(figsize=(10,6))
sns.barplot(x='Aggregate rating', y='Cuisines', data=avg_cuisine_rating.head(10), p
plt.title('Top 10 Cuisines by Average Rating')
plt.xlabel('Average Rating')
plt.ylabel('Cuisines')
plt.show()
```

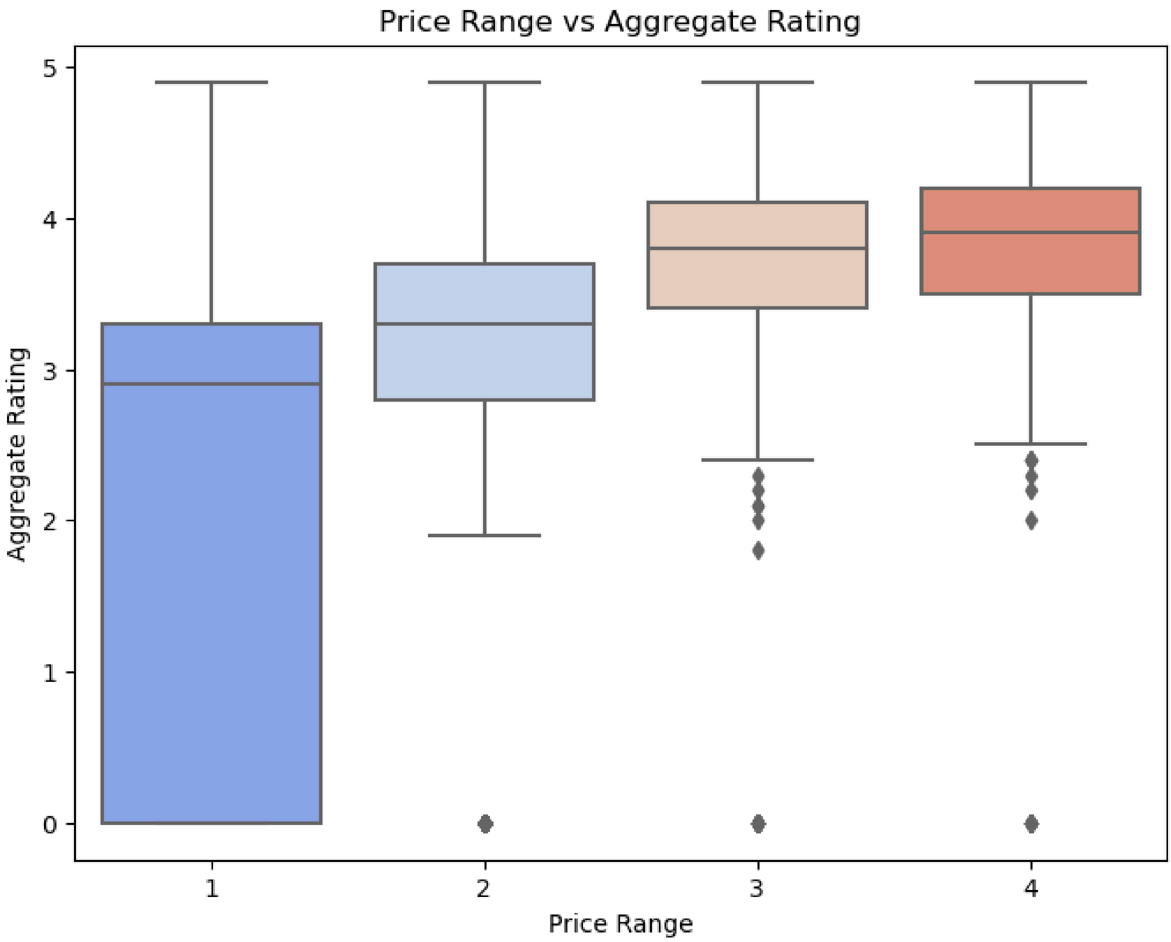


In [87]:
```
# 3: Visualize relationship between various features and aggregate rating
```

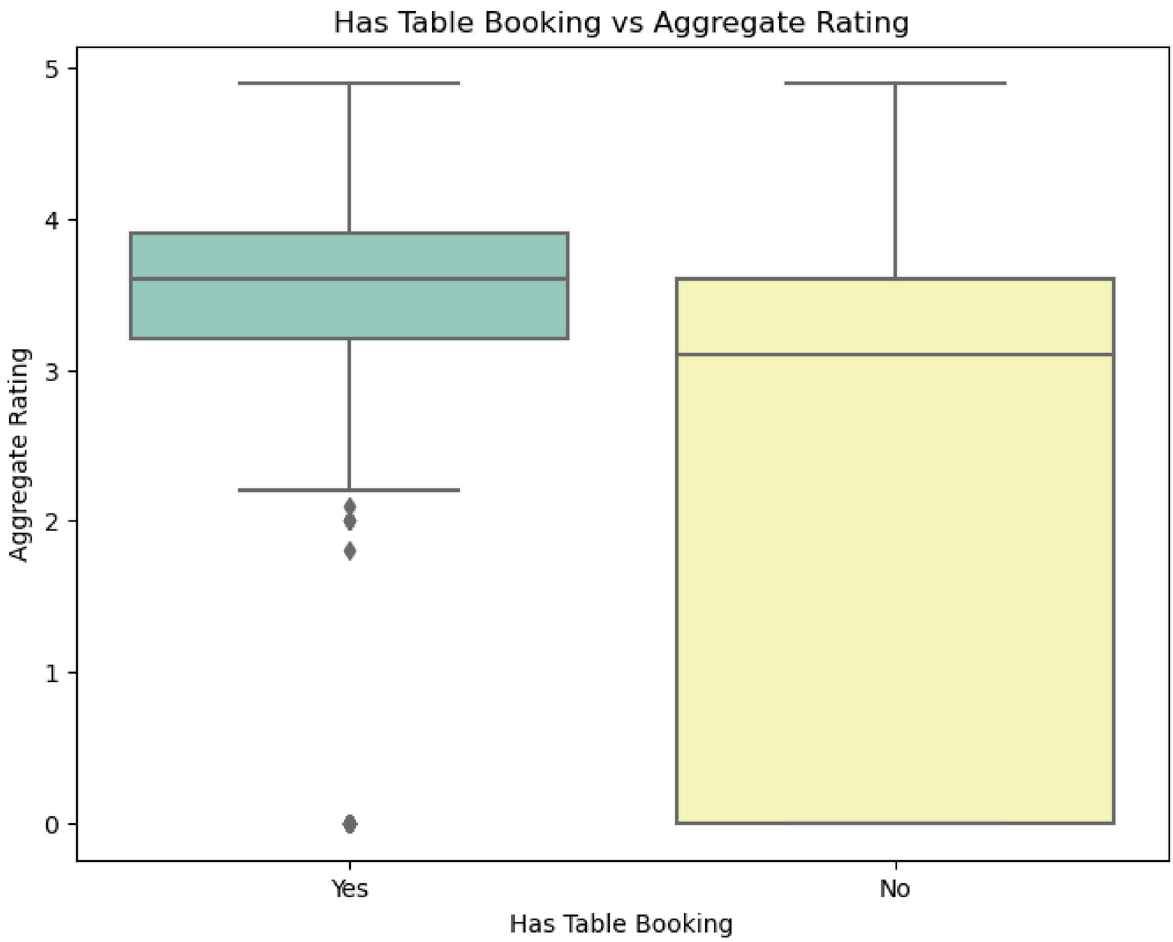
In [88]:
```
plt.figure(figsize=(8,6))
sns.scatterplot(x='Average Cost for two', y='Aggregate rating', data=df, hue='Price
plt.title('Average Cost for Two vs Aggregate Rating')
plt.xlabel('Average Cost for Two')
plt.ylabel('Aggregate Rating')
plt.show()
```



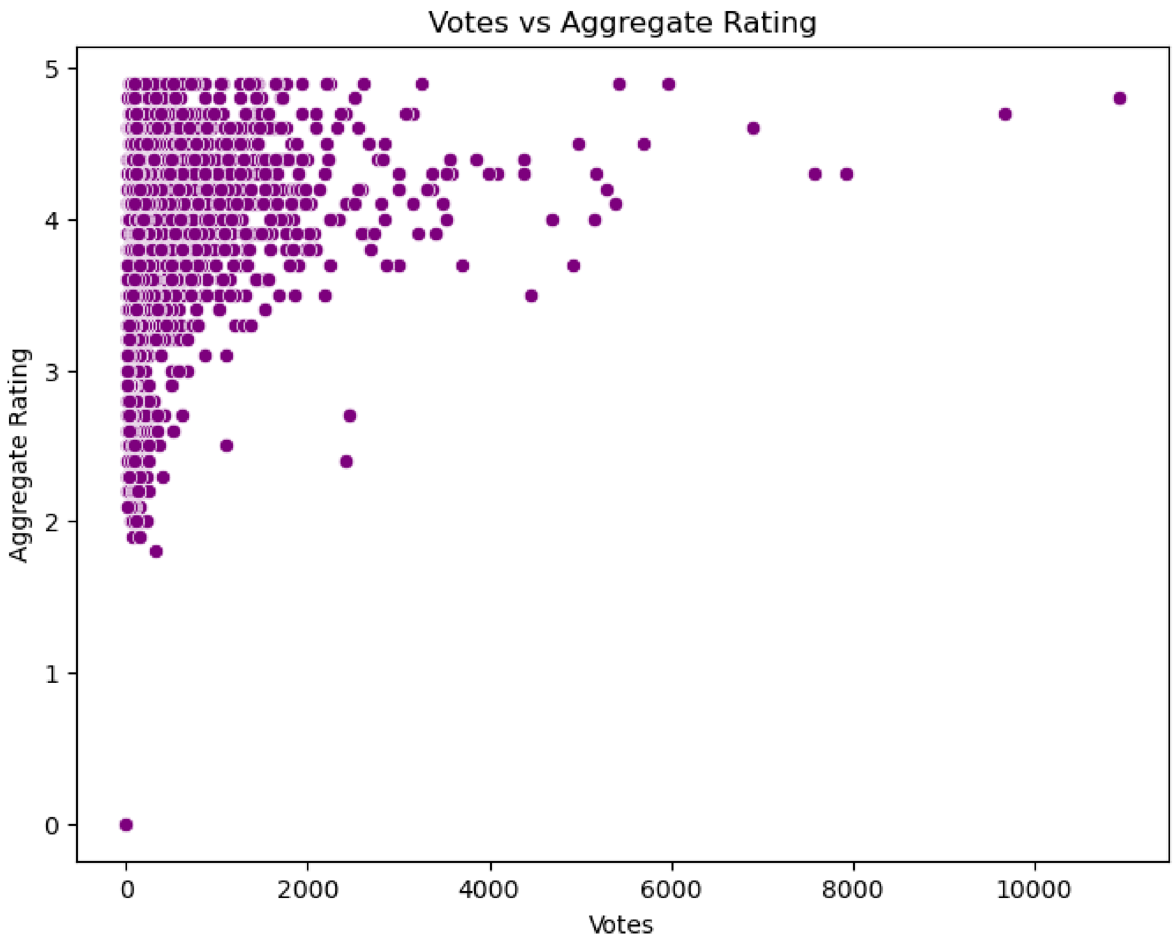
In [89]:
```
plt.figure(figsize=(8,6))
sns.boxplot(x='Price range', y='Aggregate rating', data=df, palette='coolwarm')
plt.title('Price Range vs Aggregate Rating')
plt.xlabel('Price Range')
plt.ylabel('Aggregate Rating')
plt.show()
```



In [90]:
```
plt.figure(figsize=(8,6))
sns.boxplot(x='Has Table booking', y='Aggregate rating', data=df, palette='Set3')
plt.title('Has Table Booking vs Aggregate Rating')
plt.xlabel('Has Table Booking')
plt.ylabel('Aggregate Rating')
plt.show()
```



In [91]:
```
plt.figure(figsize=(8,6))
sns.scatterplot(x='Votes', y='Aggregate rating', data=df, color='purple')
plt.title('Votes vs Aggregate Rating')
plt.xlabel('Votes')
plt.ylabel('Aggregate Rating')
plt.show()
```


Votes vs Aggregate Rating
Aggregate Rating
Votes
0
1
2
3
4
5
0
2000
4000
6000
8000
10000