cognifyz-restaurant-dataset-1

May 28, 2025

1 Restaurant Customer Ratings

Name: Zahadana Haneef Thundhakkachi

Organisation: Cognifyz Technology

2 Introduction:

Online reviews and ratings have a significant role in influencing public responds and Business success. This project looks at a dataset of restaurants from different cities and countries, focusing on features like types of cuisine, prices, customer ratings, and services such as online delivery and table booking.

3 About the Dataset:

This dataset contains information contains 9551 no of row and 21 columns about various restaurants across different cities and countries, focusing on customer experiences, service options, and restaurant characteristics.its a Regression Problem which means the values of the Target variable continuous numeric variable.

4 Data Description:

- Restaurant ID : Unique identifier for each restaurant
- Restaurant Name : Name of the Restaurant
- Country code : Numerical code of the Country
- City : City where the Restaurant is Located
- Address: Address of the Restaurant
- Locality/Locality verbose: Detailed general description
- Longitude/Latitude:Geolocation coordinates of the Restaurant Location
- Cuisines:The type or variety of food served at the restaurant
- Average Cost for two:Average meal cost for two people
- Currency: Currency used for pricing

- Has Table booking: Whether table booking is available
- Has Online delivery: Whether online booking is available
- Is delivering now: Whether the restaurant is currently delivering

- Switch to order menu:A flag showing whether the restaurant switching to an online order menu view
- Price range:price Range Rating
- Aggregate rating:Overall customer Ratings
- Rating color: color code associated with a restaurant's rating
- Rating text:Textual label for rating
- Votes:Number of Customers votes

5 Level 1:

Task 1:Data Exploring and Preprocessing

[189]:
```
import pandas as pd
df=pd.read_csv("/content/Restaurant Dataset.csv")
df
```

[189]:

	Restaurant ID	Restaurant Name	Country Code	City	\
0	6317637	Le Petit Souffle	162	Makati City	
1	6304287	Izakaya Kikufuji	162	Makati City	
2	6300002	Heat - Edsa Shangri-La	162	Mandaluyong City	
3	6318506	Ooma	162	Mandaluyong City	
4	6314302	Sambo Kojin	162	Mandaluyong City	
…	…	…	…	…	
9546	5915730	Naml Gurme	208	stanbul	
9547	5908749	Ceviz A ac	208	stanbul	
9548	5915807	Huqqa	208	stanbul	
9549	5916112	A k Kahve	208	stanbul	
9550	5927402	Walter's Coffee Roastery	208	stanbul	

	Address	\
0	Third Floor, Century City Mall, Kalayaan Avenu…	
1	Little Tokyo, 2277 Chino Roces Avenue, Legaspi…	
2	Edsa Shangri-La, 1 Garden Way, Ortigas, Mandal…	
3	Third Floor, Mega Fashion Hall, SM Megamall, O…	
4	Third Floor, Mega Atrium, SM Megamall, Ortigas…	
…	…	
9546	Kemanke Karamustafa Pa a Mahallesi, R ht m …	
9547	Ko uyolu Mahallesi, Muhittin st _nda Cadd…	
9548	Kuru _e me Mahallesi, Muallim Naci Caddesi, N…	

```
9549  Kuru _e me Mahallesi, Muallim Naci Caddesi, N…
9550  Cafea a Mahallesi, Bademalt  Sokak, No 21/B, …

                                      Locality  \
0      Century City Mall, Poblacion, Makati City
1     Little Tokyo, Legaspi Village, Makati City
2     Edsa Shangri-La, Ortigas, Mandaluyong City
3         SM Megamall, Ortigas, Mandaluyong City
4         SM Megamall, Ortigas, Mandaluyong City
…                                           …
9546                                   Karak _y
9547                                  Ko uyolu
9548                                 Kuru _e me
9549                                 Kuru _e me
9550                                        Moda

                                    Locality Verbose   Longitude  \
0     Century City Mall, Poblacion, Makati City, Mak…  121.027535
1     Little Tokyo, Legaspi Village, Makati City, Ma…  121.014101
2     Edsa Shangri-La, Ortigas, Mandaluyong City, Ma…  121.056831
3     SM Megamall, Ortigas, Mandaluyong City, Mandal…  121.056475
4     SM Megamall, Ortigas, Mandaluyong City, Mandal…  121.057508
…                                                  …         …
9546                                Karak _y,  stanbul   28.977392
9547                               Ko uyolu,  stanbul   29.041297
9548                              Kuru _e me,  stanbul   29.034640
9549                              Kuru _e me,  stanbul   29.036019
9550                                    Moda,  stanbul   29.026016

       Latitude                          Cuisines  …          Currency  \
0     14.565443        French, Japanese, Desserts  …  Botswana Pula(P)
1     14.553708                          Japanese  …  Botswana Pula(P)
2     14.581404  Seafood, Asian, Filipino, Indian  …  Botswana Pula(P)
3     14.585318                  Japanese, Sushi  …  Botswana Pula(P)
4     14.584450                 Japanese, Korean  …  Botswana Pula(P)
…         …                                  …  …             …
9546  41.022793                          Turkish  …  Turkish Lira(TL)
9547  41.009847   World Cuisine, Patisserie, Cafe  …  Turkish Lira(TL)
9548  41.055817            Italian, World Cuisine  …  Turkish Lira(TL)
9549  41.057979                  Restaurant Cafe  …  Turkish Lira(TL)
9550  40.984776                             Cafe  …  Turkish Lira(TL)

     Has Table booking Has Online delivery Is delivering now  \
0                  Yes                  No                No
1                  Yes                  No                No
2                  Yes                  No                No
3                   No                  No                No
```

```
4                   Yes                   No                  No
…                   …                     …                   …
9546                 No                   No                  No
9547                 No                   No                  No
9548                 No                   No                  No
9549                 No                   No                  No
9550                 No                   No                  No

     Switch to order menu Price range  Aggregate rating  Rating color  \
0                      No           3               4.8    Dark Green
1                      No           3               4.5    Dark Green
2                      No           4               4.4         Green
3                      No           4               4.9    Dark Green
4                      No           4               4.8    Dark Green
…                     …           …                 …           …
9546                   No           3               4.1         Green
9547                   No           3               4.2         Green
9548                   No           4               3.7        Yellow
9549                   No           4               4.0         Green
9550                   No           2               4.0         Green

     Rating text Votes
0      Excellent   314
1      Excellent   591
2      Very Good   270
3      Excellent   365
4      Excellent   229
…          …   …
9546   Very Good   788
9547   Very Good  1034
9548        Good   661
9549   Very Good   901
9550   Very Good   591

[9551 rows x 21 columns]
```

Made the Dataset copy for the future refference

[190]:
```
data=df.copy()
```

Find the shape of the dataset that means no of row and columns ,Here 9551 is rows and 21 columns.

[191]:
```
df.shape
```

[191]:
```
(9551, 21)
```

[192]:
```
df.columns
```

[192]:
```
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')
```

[193]:
```
df.isnull().sum()  # check the null values in columns
```

[193]:
```
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
```

Here column cuisine has null values.it can be treated by using the function dropna()

[194]:
```
df.dropna(inplace=True)
```

[195]:
```
df.isnull().sum() # again check the null values if its present or not
```

[195]:
```
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
```

[196]:
```
df.duplicated().sum() # check for the duplicates rows
```

[196]:
```
np.int64(0)
```

[197]:
```
df.info() # check dataset informations like data type
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 9542 entries, 0 to 9550
Data columns (total 21 columns):
 #   Column                Non-Null Count  Dtype  
---  ------                --------------  -----  
 0   Restaurant ID         9542 non-null   int64  
 1   Restaurant Name       9542 non-null   object 
 2   Country Code          9542 non-null   int64  
 3   City                  9542 non-null   object 
 4   Address               9542 non-null   object 
 5   Locality              9542 non-null   object 
 6   Locality Verbose      9542 non-null   object 
 7   Longitude             9542 non-null   float64
 8   Latitude              9542 non-null   float64
 9   Cuisines              9542 non-null   object 
 10  Average Cost for two  9542 non-null   int64  
 11  Currency              9542 non-null   object 
 12  Has Table booking     9542 non-null   object 
 13  Has Online delivery   9542 non-null   object 
 14  Is delivering now     9542 non-null   object 
 15  Switch to order menu  9542 non-null   object 
 16  Price range           9542 non-null   int64  
 17  Aggregate rating      9542 non-null   float64
 18  Rating color          9542 non-null   object 
 19  Rating text           9542 non-null   object 
 20  Votes                 9542 non-null   int64  
```

```
dtypes: float64(3), int64(5), object(13)
memory usage: 1.6+ MB
```

Deleting unwanted columns in dataset like Locality Verbose

[198]:
```
df.columns = df.columns.str.strip()
df = df.drop(columns=["Locality Verbose", "Restaurant ID"], axis=1)
```

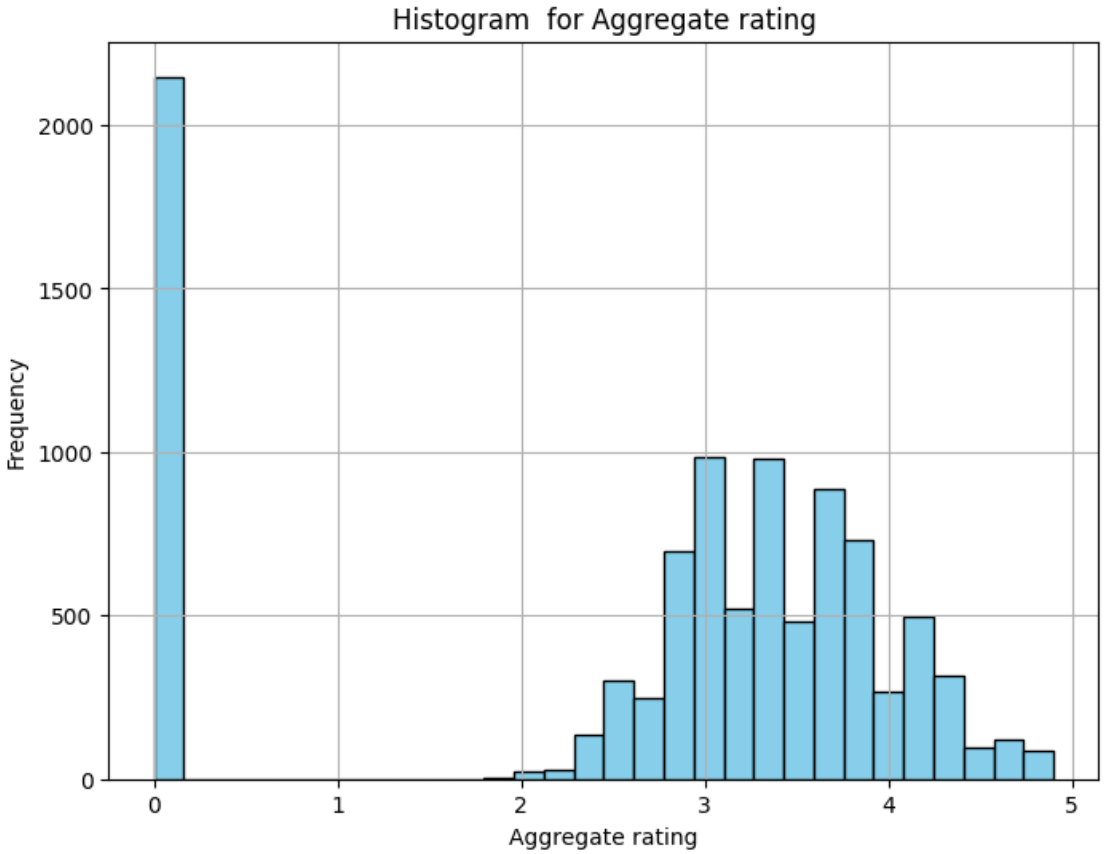
distribution of the target variable

[199]:
```
df['Aggregate rating'].value_counts()
```

[199]:
```
Aggregate rating
0.0    2148
3.2     522
3.1     519
3.4     495
3.3     483
3.5     480
3.0     468
3.6     458
3.7     427
3.8     399
2.9     381
3.9     332
2.8     315
4.1     274
4.0     266
2.7     250
4.2     221
2.6     191
4.3     174
4.4     143
2.5     110
4.5      95
2.4      87
4.6      78
4.9      61
2.3      47
4.7      41
2.2      27
4.8      25
2.1      15
2.0       7
1.9       2
1.8       1
Name: count, dtype: int64
```

[200]:
```
import matplotlib.pyplot as plt
import seaborn as sns

plt.figure(figsize=(8,6))
plt.hist(df['Aggregate rating'],bins=30,color='skyblue',edgecolor='black')
plt.title('Histogram  for Aggregate rating')
plt.xlabel('Aggregate rating')
plt.ylabel('Frequency')
plt.grid()
plt.show()
```



Histogram shows the visual representation of aggregate ratings

- There is a spike at the rating rangeof 0 ,that means Many items have not been rated
- most of the ratings between 2.5 to 4.5 it represented as a bell shape
- the frequency for the range 3 to 4 is the highest when compared to non zero ratings it showing the most item has the same rating values.

Task 2:Descriptive Analysis

[201]:
```
df.describe()
```

[201]:

	Country Code	Longitude	Latitude	Average Cost for two	\
count	9542.000000	9542.000000	9542.000000	9542.000000	
mean	18.179208	64.274997	25.848532	1200.326137	
std	56.451600	41.197602	11.010094	16128.743876	
min	1.000000	-157.948486	-41.330428	0.000000	
25%	1.000000	77.081565	28.478658	250.000000	
50%	1.000000	77.192031	28.570444	400.000000	
75%	1.000000	77.282043	28.642711	700.000000	
max	216.000000	174.832089	55.976980	800000.000000	

	Price range	Aggregate rating	Votes
count	9542.000000	9542.000000	9542.000000
mean	1.804968	2.665238	156.772060
std	0.905563	1.516588	430.203324
min	1.000000	0.000000	0.000000
25%	1.000000	2.500000	5.000000
50%	2.000000	3.200000	31.000000
75%	2.000000	3.700000	130.000000
max	4.000000	4.900000	10934.000000

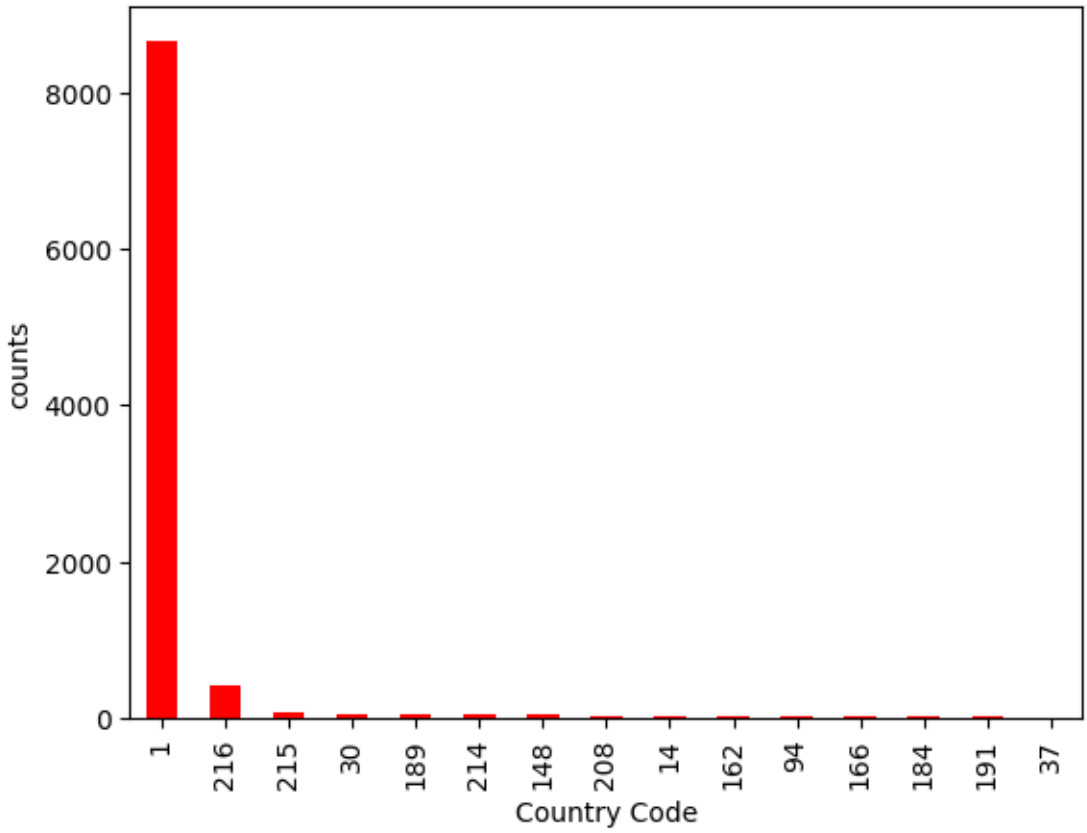
[202]:
```
numerical_cols=df.select_dtypes(include='number')# filter numerical columns
```

[203]:
```
numerical_cols.columns
```

[203]:
```
Index(['Country Code', 'Longitude', 'Latitude', 'Average Cost for two',
       'Price range', 'Aggregate rating', 'Votes'],
      dtype='object')
```

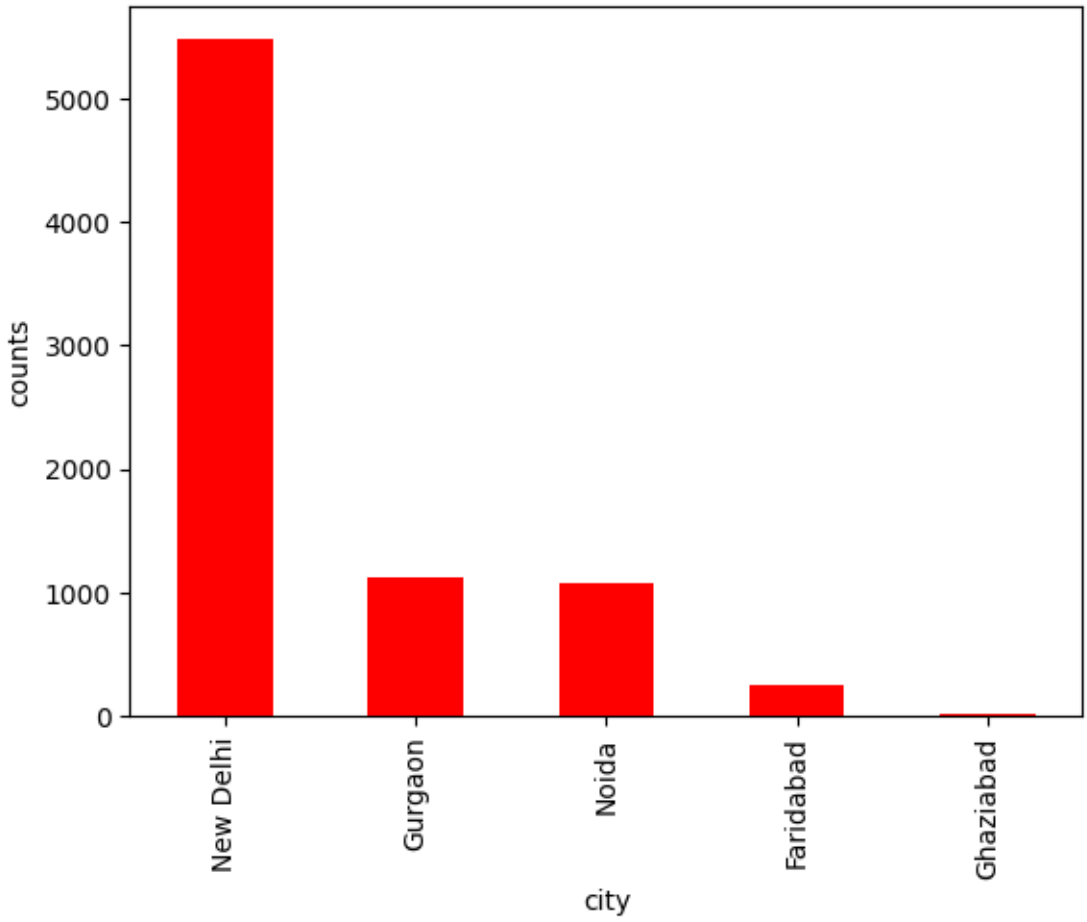
distribution of categorical variables like "Country Code," "City," and "Cuisines."

[204]:
```
df['Country Code'].value_counts().plot(kind='bar',color='red')
plt.xlabel('Country Code')
plt.ylabel('counts')
plt.show()
```



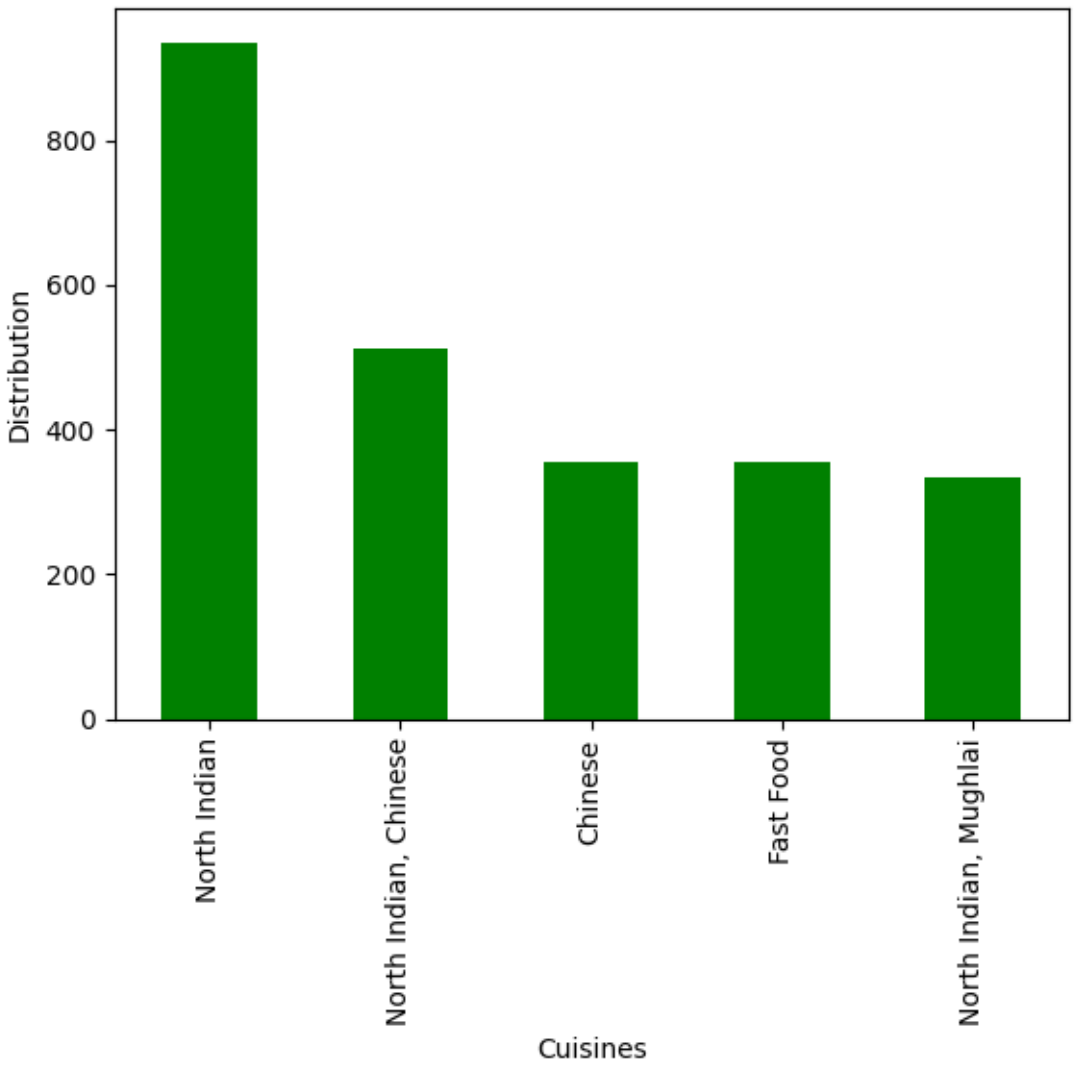
- Country code 1 has the most no of Restaurants other has only few no of restaurant

[205]:
```
df['City'].value_counts().head(5).plot(kind='bar',color='red')
plt.xlabel('city')
plt.ylabel('counts')
plt.show()
```



- New Delhi has the highest count of Restaurants by a significant margin (over 5000).
- Gurgaon and Noida have roughly similar counts, slightly above 1000.

[206]:
```
df['Cuisines'].value_counts().head(5).plot(kind='bar',color='green')
plt.xlabel('Cuisines')
plt.ylabel('Distribution')
plt.show()
```



- It has the highest distribution, with close to 900 entries, indicating it's the most popular or most frequently available cuisine.
- North Indian, Chinese" has the second highest distribution
- Chinese and Fast Food are equally popular:

[207]:
```
df.groupby('Cuisines')['Votes'].sum().reset_index()
```

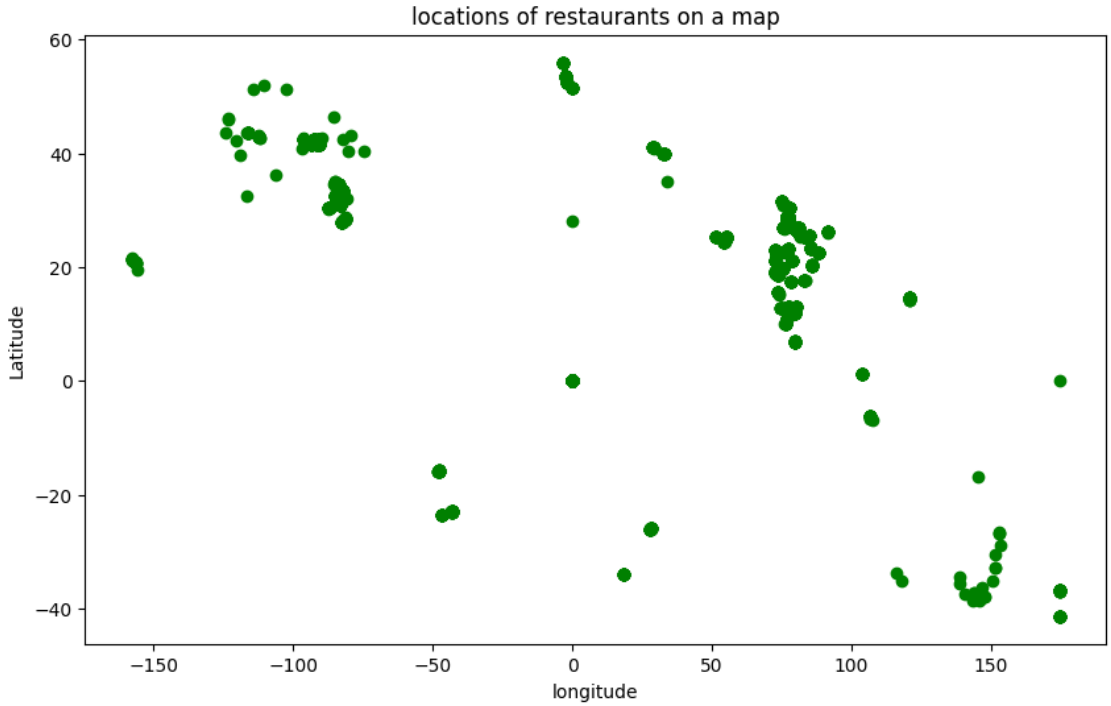
[207]:

	Cuisines	Votes
0	Afghani	39
1	Afghani, Mughlai, Chinese	2
2	Afghani, North Indian	0
3	Afghani, North Indian, Pakistani, Arabian	3
4	African	373

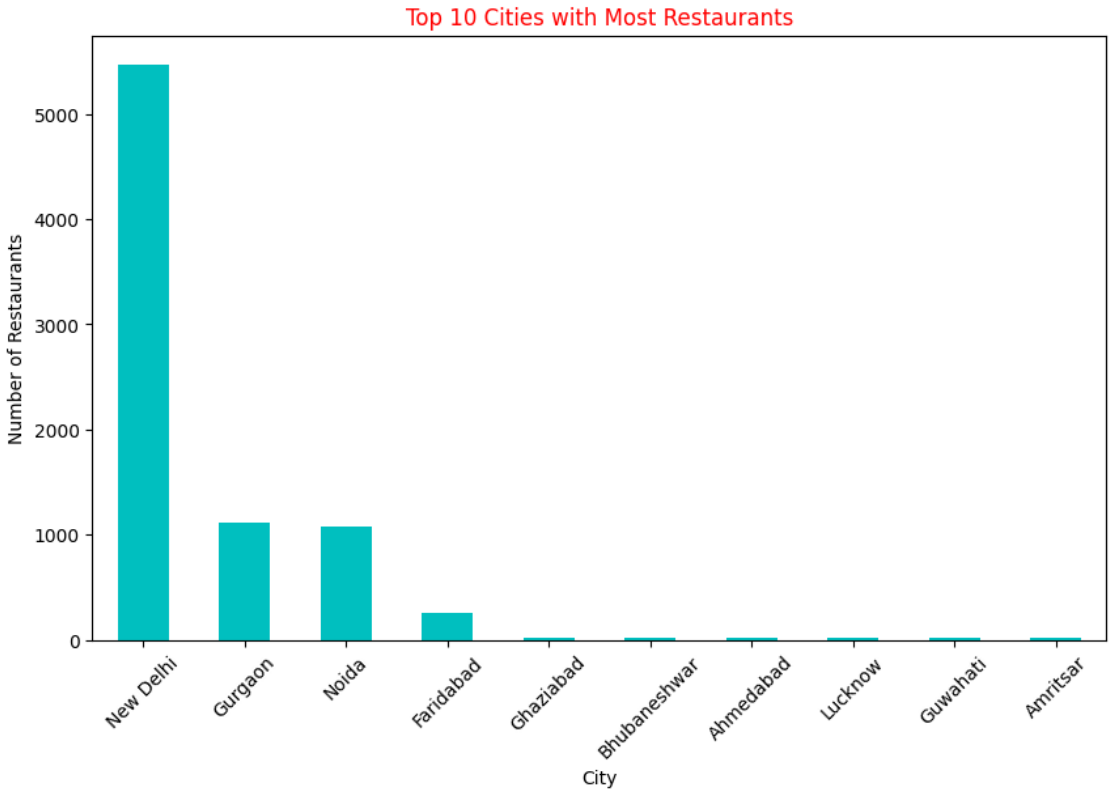
```
…                                          …    …
1820                       Western, Asian, Cafe    259
1821                  Western, Fusion, Fast Food     32
1822                                World Cuisine     95
1823          World Cuisine, Mexican, Italian    115
1824          World Cuisine, Patisserie, Cafe   1034

[1825 rows x 2 columns]
```

[208]:
```
plt.figure(figsize=(10, 6))
plt.scatter(df['Longitude'], df['Latitude'],color='green')
plt.title('locations of restaurants on a map')
plt.xlabel('longitude')
plt.ylabel('Latitude')
plt.show()
```



[209]:
```
highest_city=df['City'].value_counts().head(10)
plt.figure(figsize=(10, 6))
highest_city.plot(kind='bar', color='c')
plt.title("Top 10 Cities with Most Restaurants",color='red')
plt.xlabel("City")
plt.ylabel("Number of Restaurants")
plt.xticks(rotation=45)
plt.show()
```



Correlation Heat map

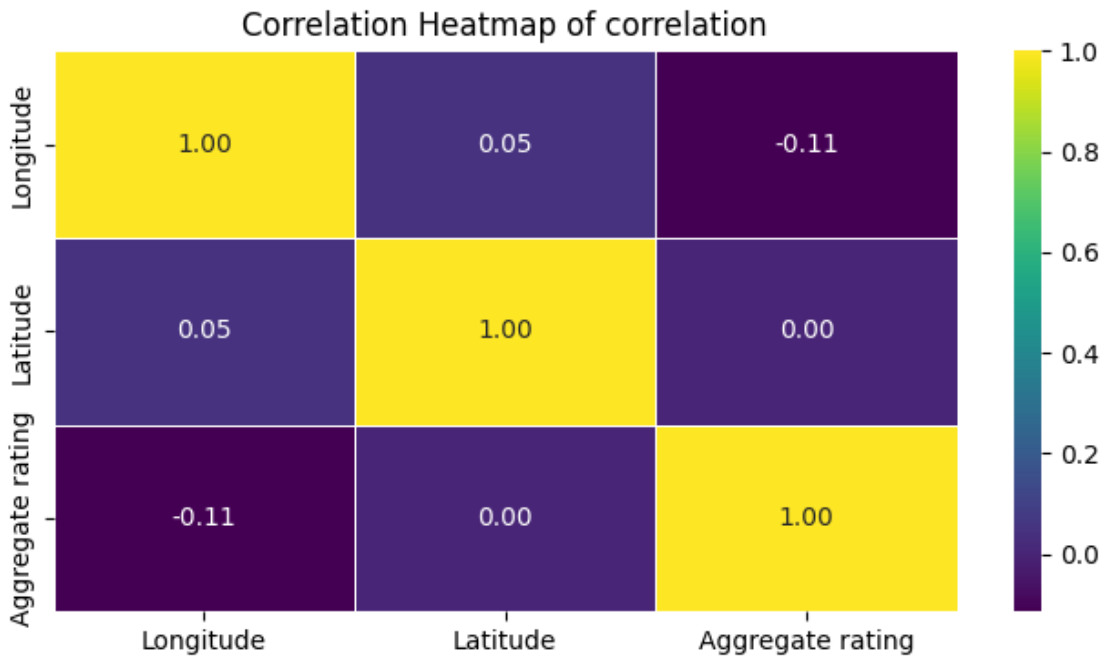
The correlation matrix shows the pairwise correlation coefficients between numerical columns:

[210]:
```
correlation_columns=df[['Longitude','Latitude','Aggregate rating']].corr()
correlation_columns
```

[210]:

	Longitude	Latitude	Aggregate rating
Longitude	1.000000	0.045415	-0.114733
Latitude	0.045415	1.000000	0.000197
Aggregate rating	-0.114733	0.000197	1.000000

[211]:
```
plt.figure(figsize=(8, 4))
sns.heatmap(correlation_columns, annot=True, cmap="viridis", fmt=".2f",␣
↪linewidths=0.5)
plt.title("Correlation Heatmap of correlation")
plt.show()
```



- Longnitude and Rating has 0.16 correlation which means weak
- Latitude and Rating has 0.13 correlation its weak

- Location has little no meaningful correlation with ratings.

6 Level 2:

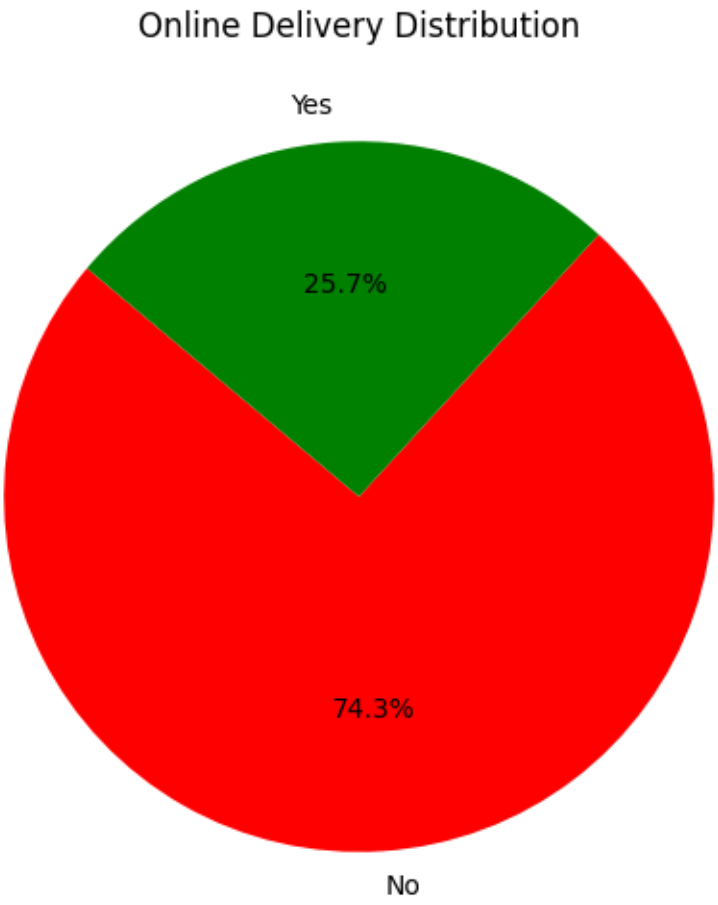
Table Booking and Online Delivery

Online Delivery Distribution

[212]:
```
Online_del=df['Has Online delivery'].value_counts()
Online_del
```

[212]:
```
Has Online delivery
No     7091
Yes    2451
Name: count, dtype: int64
```

[213]:
```
plt.figure(figsize=(6, 6))
plt.pie(Online_del, labels=Online_del.index, autopct='%1.1f%%', startangle=140,␣
 ↪colors=['red', 'green'])
plt.title('Online Delivery Distribution')
plt.show()
```

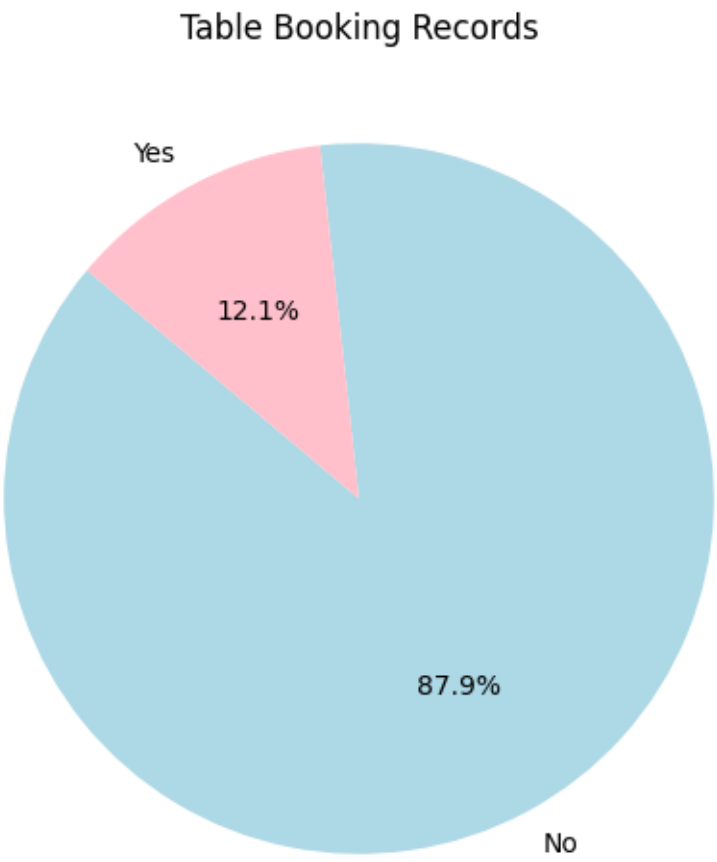


- 74.3% of restaurants do not offer online delivery.
- 25.7% of restaurants do offer online delivery.

[214]:
```
Table_booking=df['Has Table booking'].value_counts()
Table_booking
```

[214]:
```
Has Table booking
No     8384
Yes    1158
Name: count, dtype: int64
```

[215]:
```
plt.figure(figsize=(6, 6))
plt.pie(Table_booking, labels=Table_booking.index, autopct='%1.1f%%',␣
 ↪startangle=140, colors=['lightblue', 'pink'])
plt.title('Table Booking Records')
plt.show()
```



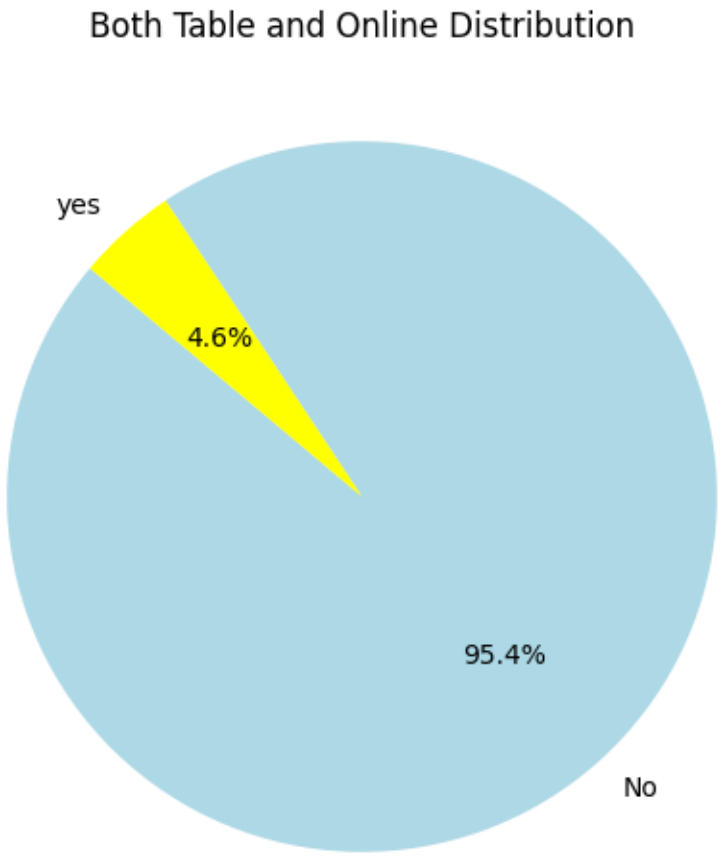
- 87.9% Restaurants not offering table booking
- only 12.1 % Restaurants offering table booking

The Restaurants which have both Table Booking and Online Booking

[216]:
```
Table_online=((df['Has Table booking'] == 'Yes') & (df['Has Online delivery']␣
↪== 'Yes')).value_counts()
Table_online
```

[216]:
```
False    9107
True      435
Name: count, dtype: int64
```

[217]:
```
plt.figure(figsize=(6, 6))
plt.pie(Table_online, labels=['No','yes'], autopct='%1.1f%%', startangle=140,␣
↪colors=['lightblue', 'yellow'])
plt.title('Both Table and Online Distribution')
plt.show()
```



- 95.4% Restarants are not offering the both Table booking and Online booking.
- 4.6% Are offering the both services.

Average Rating of Restaurants

[218]:
```
Average_delivery=df.groupby('Has Online delivery')['Aggregate rating'].mean()
Average_delivery
```

[218]:
```
Has Online delivery
No     2.463517
Yes    3.248837
Name: Aggregate rating, dtype: float64
```

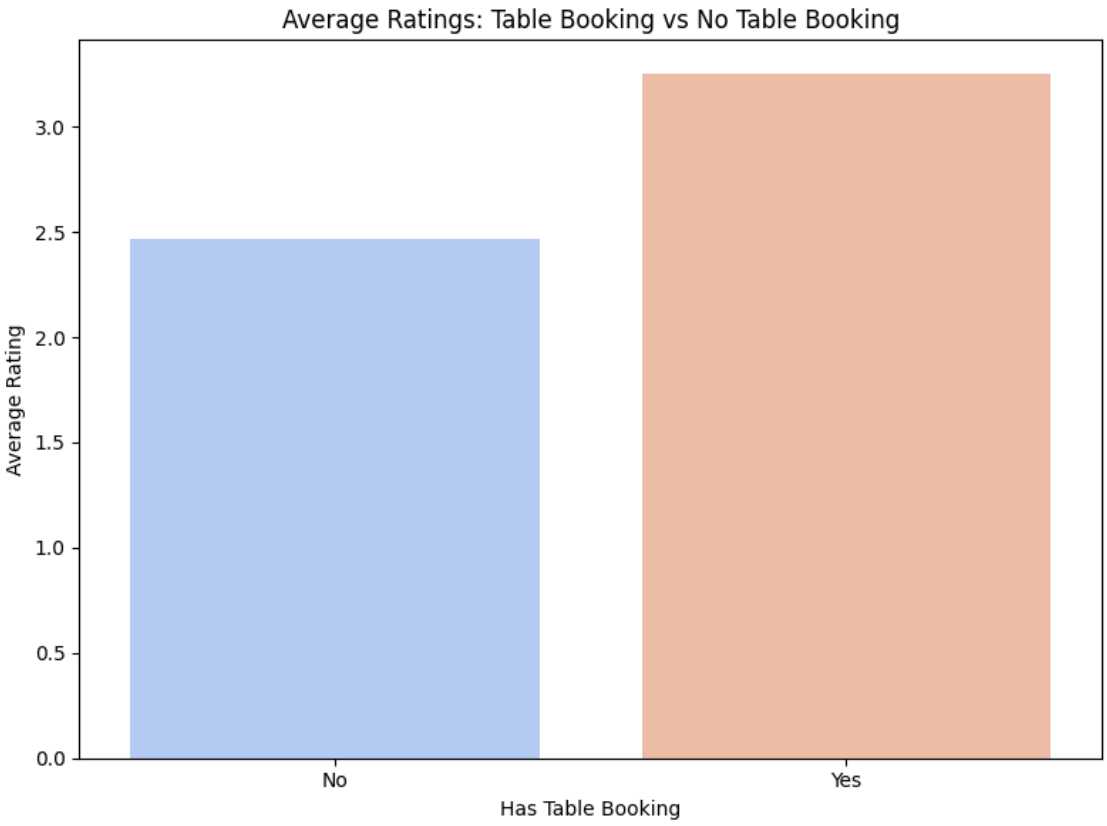
[219]:
```
plt.figure(figsize=(8,6))
sns.barplot(x=Average_delivery.index, y=Average_delivery.values,␣
 ↪palette='coolwarm')
plt.title('Average Ratings: Table Booking vs No Table Booking')
plt.xlabel('Has Table Booking')
```

```
plt.ylabel('Average Rating')
plt.tight_layout()
plt.show()
```

```
<ipython-input-219-6fecc5e6ac88>:2: 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=Average_delivery.index, y=Average_delivery.values,
palette='coolwarm')
```



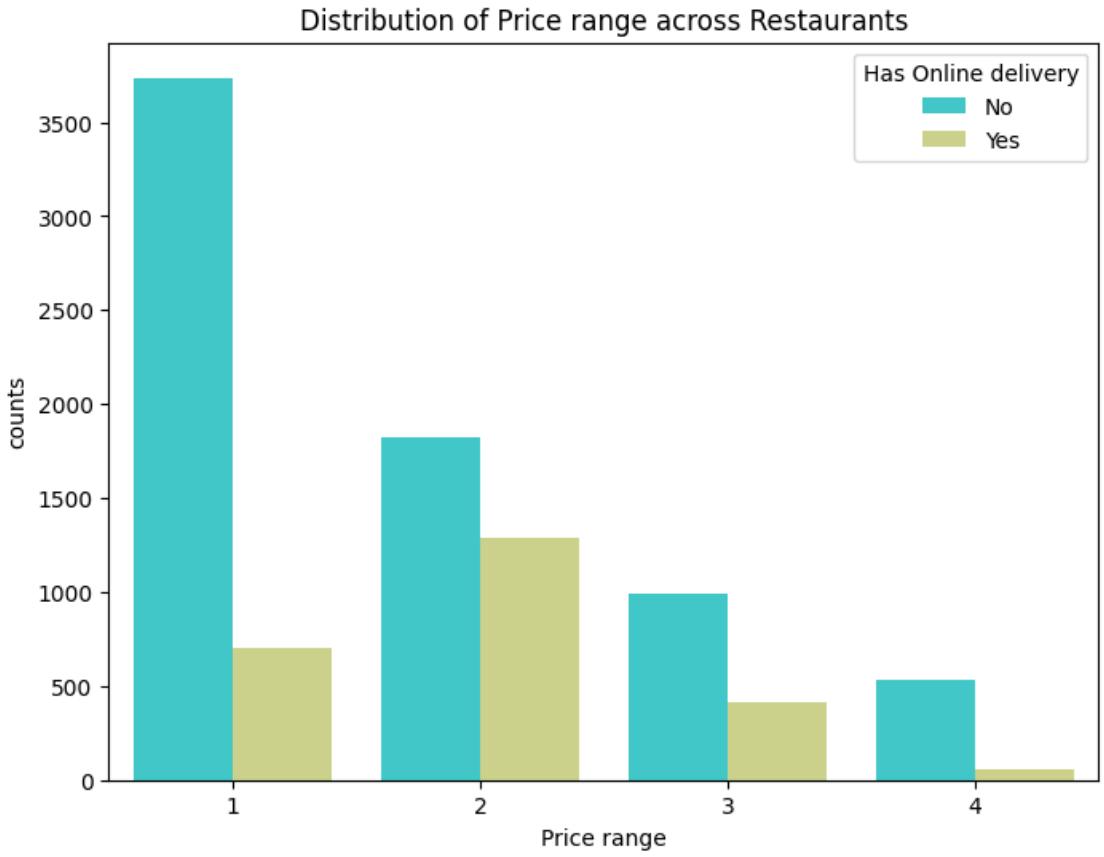
- Restaurants with table booking have an average rating above 3.2.
- Restaurants without table booking average around 2.45.

Online Delivery with Price Range

[220]:
```
online_price=df.groupby('Price range')['Has Online delivery'].value_counts()
online_price
```

[220]:
```
Price range  Has Online delivery
1            No                    3737
             Yes                    701
2            No                    1827
             Yes                   1286
3            No                     994
             Yes                    411
4            No                     533
             Yes                     53
Name: count, dtype: int64
```

[221]:
```
plt.figure(figsize=(8,6))
sns.countplot(df,x='Price range',hue='Has Online delivery',palette='rainbow')
plt.xlabel('Price range')
plt.ylabel('counts')
plt.title('Distribution of Price range across Restaurants')
plt.show()
```



- Price Range 1 has the highest number of restaurants. when compared to other ranges
- As the price range increases, the number of restaurants decreases

- Price range increases the no of restaurants will be decrease

Price Range Analysis

[222]:
```
most_price_range=df['Price range'].mode()[0]
most_price_range
```

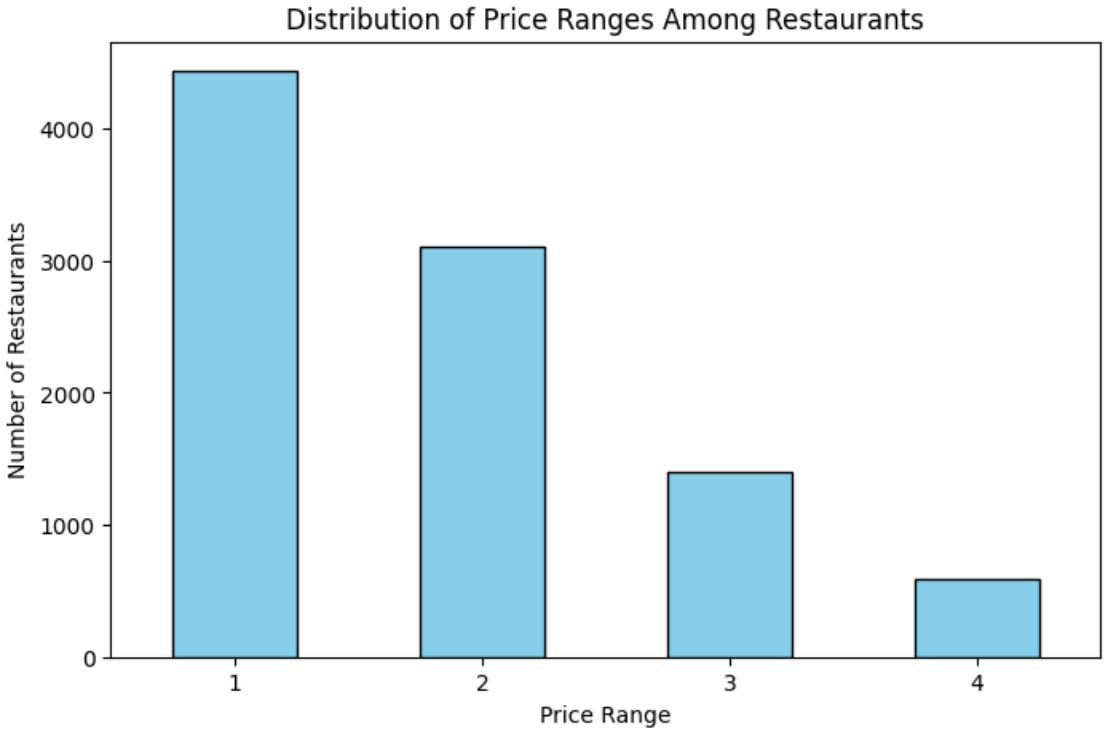
[222]:
```
np.int64(1)
```

[223]:
```
most_occurrences=df['Price range'].value_counts()
most_occurrences
```

[223]:
```
Price range
1    4438
2    3113
3    1405
4     586
Name: count, dtype: int64
```

[224]:
```
plt.figure(figsize=(8, 5))
most_occurrences.plot(kind='bar', color='skyblue', edgecolor='black')

plt.title('Distribution of Price Ranges Among Restaurants')
plt.xlabel('Price Range')
plt.ylabel('Number of Restaurants')
plt.xticks(rotation=0)
plt.show()
```

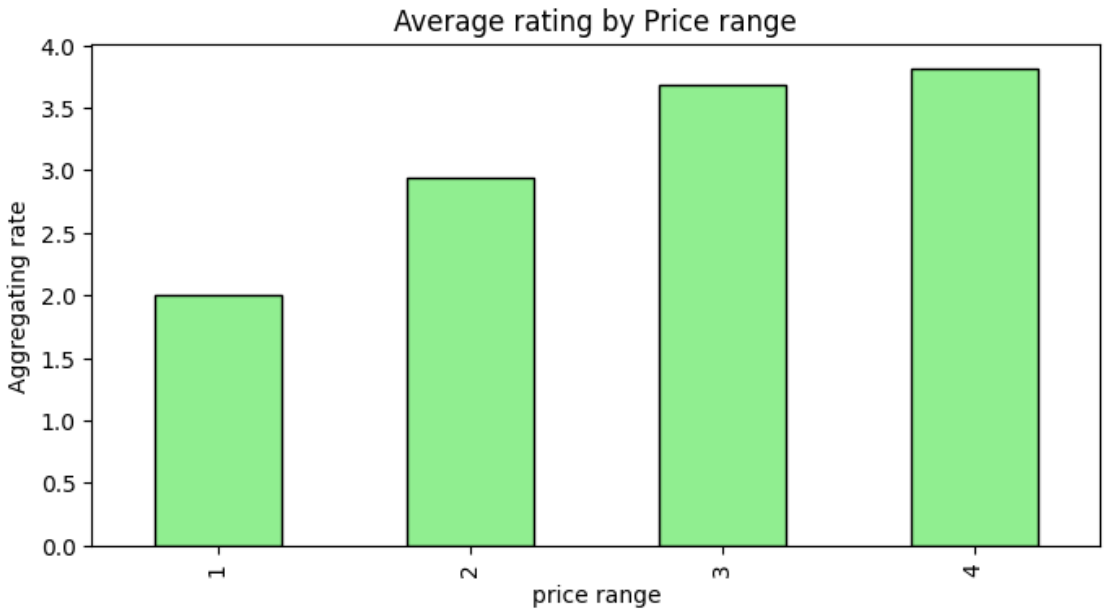


- bar chart showing the distribution of price ranges among restaurants. Price Range 1 is the most common

[225]:
```
avg_rate_price=df.groupby('Price range')['Aggregate rating'].mean().sort_index()
avg_rate_price
```

[225]:
```
Price range
1    1.997476
2    2.941054
3    3.682633
4    3.817918
Name: Aggregate rating, dtype: float64
```

[226]:
```
plt.figure(figsize=(8,4))
avg_rate_price.plot(kind='bar',color='lightgreen',edgecolor='black')
plt.title('Average rating by Price range')
plt.xlabel('price range')
plt.ylabel('Aggregating rate')
plt.show()
```



- Price range 1 has the lowest average rating, around 2.
- Price range 3,4 has the slightly common mean value around 3.8
- when price range increases the mean value also increase

[227]:
```
highest_average_price=avg_rate_price.idxmax()
highest_average_price
```

[227]:
```
np.int64(4)
```

[228]:
```
highest_rating = df[df['Price range'] == highest_average_price]
```

[229]:
```
most_common_color=highest_rating['Rating color'].mode()[0]
most_common_color
```

[229]:
```
'Yellow'
```

[230]:
```
rating_color=df.groupby('Price range')['Rating color'].agg(lambda x: x.
↪mode()[0])
rating_color
```

[230]:
```
Price range
1    Orange
2    Orange
3    Yellow
4    Yellow
Name: Rating color, dtype: object
```

[231]:
```
rating_color.columns=['Price range','Rating color']
avg_rate_price.columns=['Price range','Aggregate rating']
merged_columns=pd.merge(rating_color, avg_rate_price, on='Price range',␣
↪how='inner')
merged_columns
```

[231]:

Price range	Rating color	Aggregate rating
1	Orange	1.997476
2	Orange	2.941054
3	Yellow	3.682633
4	Yellow	3.817918

[232]:
```
top_rating = merged_columns.loc[merged_columns["Aggregate rating"].idxmax()]
top_rating
```

[232]:
```
Rating color          Yellow
Aggregate rating    3.817918
Name: 4, dtype: object
```

- highest price range is 4 with highest rating around 3.81
- highest range color indicating as yellow

Feature Engineering

[233]:
```
df['Address Length'] = df['Address'].astype(str).apply(len)
df[['Address','Address Length']].head(10)
```

[233]:

	Address	Address Length
0	Third Floor, Century City Mall, Kalayaan Avenu…	71
1	Little Tokyo, 2277 Chino Roces Avenue, Legaspi…	67
2	Edsa Shangri-La, 1 Garden Way, Ortigas, Mandal…	56
3	Third Floor, Mega Fashion Hall, SM Megamall, O…	70
4	Third Floor, Mega Atrium, SM Megamall, Ortigas…	64
5	Ground Floor, Mega Fashion Hall, SM Megamall, …	71
6	Building K, SM By The Bay, Sunset Boulevard, M…	83
7	Building B, By The Bay, Seaside Boulevard, Mal…	81
8	Plaza Level, Sofitel Philippine Plaza Manila, …	69
9	Brixton Technology Center, 10 Brixton Street, …	67

[234]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 9542 entries, 0 to 9550
Data columns (total 20 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   Restaurant Name       9542 non-null   object
 1   Country Code          9542 non-null   int64
 2   City                  9542 non-null   object
 3   Address               9542 non-null   object
 4   Locality              9542 non-null   object
 5   Longitude             9542 non-null   float64
 6   Latitude              9542 non-null   float64
 7   Cuisines              9542 non-null   object
 8   Average Cost for two  9542 non-null   int64
 9   Currency              9542 non-null   object
 10  Has Table booking     9542 non-null   object
 11  Has Online delivery   9542 non-null   object
 12  Is delivering now     9542 non-null   object
 13  Switch to order menu  9542 non-null   object
 14  Price range           9542 non-null   int64
 15  Aggregate rating      9542 non-null   float64
 16  Rating color          9542 non-null   object
 17  Rating text           9542 non-null   object
 18  Votes                 9542 non-null   int64
 19  Address Length        9542 non-null   int64
dtypes: float64(3), int64(5), object(12)
memory usage: 1.5+ MB
```

[235]:
```
# deleting unwanted columns in dataset for building model
df.drop(["Restaurant Name","Currency","City","Address"
```

```
,"Locality","Country Code","Longitude","Latitude"],axis=1,inplace=True)
```

[236]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 9542 entries, 0 to 9550
Data columns (total 12 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   Cuisines              9542 non-null   object
 1   Average Cost for two  9542 non-null   int64
 2   Has Table booking     9542 non-null   object
 3   Has Online delivery   9542 non-null   object
 4   Is delivering now     9542 non-null   object
 5   Switch to order menu  9542 non-null   object
 6   Price range           9542 non-null   int64
 7   Aggregate rating      9542 non-null   float64
 8   Rating color          9542 non-null   object
 9   Rating text           9542 non-null   object
 10  Votes                 9542 non-null   int64
 11  Address Length        9542 non-null   int64
dtypes: float64(1), int64(4), object(7)
memory usage: 969.1+ KB
```

[237]:
```
df['Is Delivering Now'] = df['Is delivering now'].apply(lambda x: 1 if x ==␣
 ↪'Yes' else 0)
```

[238]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 9542 entries, 0 to 9550
Data columns (total 13 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   Cuisines              9542 non-null   object
 1   Average Cost for two  9542 non-null   int64
 2   Has Table booking     9542 non-null   object
 3   Has Online delivery   9542 non-null   object
 4   Is delivering now     9542 non-null   object
 5   Switch to order menu  9542 non-null   object
 6   Price range           9542 non-null   int64
 7   Aggregate rating      9542 non-null   float64
 8   Rating color          9542 non-null   object
 9   Rating text           9542 non-null   object
 10  Votes                 9542 non-null   int64
 11  Address Length        9542 non-null   int64
 12  Is Delivering Now     9542 non-null   int64
```

```
dtypes: float64(1), int64(5), object(7)
memory usage: 1.0+ MB
```

one hot encoding the categorical columns

[239]:
```
df=pd.get_dummies(df,dtype=int,drop_first=True)
df
```

[239]:

	Average Cost for two	Price range	Aggregate rating	Votes	\
0	1100	3	4.8	314	
1	1200	3	4.5	591	
2	4000	4	4.4	270	
3	1500	4	4.9	365	
4	1500	4	4.8	229	
…	…	…	…	…	
9546	80	3	4.1	788	
9547	105	3	4.2	1034	
9548	170	4	3.7	661	
9549	120	4	4.0	901	
9550	55	2	4.0	591	

	Address Length	Is Delivering Now	Cuisines_Afghani, Mughlai, Chinese	\
0	71	0	0	
1	67	0	0	
2	56	0	0	
3	70	0	0	
4	64	0	0	
…	…	…	…	
9546	103	0	0	
9547	77	0	0	
9548	73	0	0	
9549	75	0	0	
9550	65	0	0	

	Cuisines_Afghani, North Indian	\
0	0	
1	0	
2	0	
3	0	
4	0	
…	…	
9546	0	
9547	0	
9548	0	
9549	0	
9550	0	

	Cuisines_Afghani, North Indian, Pakistani, Arabian	Cuisines_African	\

```
0                                                    0                  0
1                                                    0                  0
2                                                    0                  0
3                                                    0                  0
4                                                    0                  0
…                                                  …                  …
9546                                                 0                  0
9547                                                 0                  0
9548                                                 0                  0
9549                                                 0                  0
9550                                                 0                  0

      …  Rating color_Green  Rating color_Orange  Rating color_Red  \
0     …                    0                    0                 0
1     …                    0                    0                 0
2     …                    1                    0                 0
3     …                    0                    0                 0
4     …                    0                    0                 0
…   …                  …                    …                 …
9546  …                    1                    0                 0
9547  …                    1                    0                 0
9548  …                    0                    0                 0
9549  …                    1                    0                 0
9550  …                    1                    0                 0

      Rating color_White  Rating color_Yellow  Rating text_Excellent  \
0                      0                    0                      1
1                      0                    0                      1
2                      0                    0                      0
3                      0                    0                      1
4                      0                    0                      1
…                    …                    …                      …
9546                   0                    0                      0
9547                   0                    0                      0
9548                   0                    1                      0
9549                   0                    0                      0
9550                   0                    0                      0

      Rating text_Good  Rating text_Not rated  Rating text_Poor  \
0                    0                     0                 0
1                    0                     0                 0
2                    0                     0                 0
3                    0                     0                 0
4                    0                     0                 0
…                  …                     …                 …
9546                 0                     0                 0
9547                 0                     0                 0
```

```
9548                 1                       0                 0
9549                 0                       0                 0
9550                 0                       0                 0

      Rating text_Very Good
0                          0
1                          0
2                          1
3                          0
4                          0
…                        …
9546                       1
9547                       1
9548                       0
9549                       1
9550                       1

[9542 rows x 1843 columns]
```

Predictive Modeling

[240]:
```
df.columns
```

[240]:
```
Index(['Average Cost for two', 'Price range', 'Aggregate rating', 'Votes',
       'Address Length', 'Is Delivering Now',
       'Cuisines_Afghani, Mughlai, Chinese', 'Cuisines_Afghani, North Indian',
       'Cuisines_Afghani, North Indian, Pakistani, Arabian',
       'Cuisines_African',
       …
       'Rating color_Green', 'Rating color_Orange', 'Rating color_Red',
       'Rating color_White', 'Rating color_Yellow', 'Rating text_Excellent',
       'Rating text_Good', 'Rating text_Not rated', 'Rating text_Poor',
       'Rating text_Very Good'],
      dtype='object', length=1843)
```

Now dataset contains no of columns increased after encoding. better to do feature selection

[241]:
```
from sklearn.preprocessing import StandardScaler
from sklearn.feature_selection import SelectKBest, f_regression
from sklearn.ensemble import RandomForestRegressor
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.tree import DecisionTreeRegressor
from sklearn.metrics import mean_squared_error,mean_absolute_error,r2_score
```

[242]:
```
X = df.drop('Aggregate rating', axis=1)
y = df['Aggregate rating']
```

```
model = RandomForestRegressor(random_state=42)
model.fit(X, y)

importances = model.feature_importances_

# Create a DataFrame with feature scores
feature_importance_df = pd.DataFrame({
    'Feature': X.columns,
    'Importance': importances
}).sort_values(by='Importance', ascending=False)

print(feature_importance_df)
```

```
                                                Feature    Importance
1835                                Rating color_White  3.141690e-01
1839                              Rating text_Not rated  3.140494e-01
2                                                 Votes  2.726146e-01
1833                               Rating color_Orange  5.180078e-02
1840                                  Rating text_Poor  1.200229e-02
…                                                   …           …
1815       Cuisines_Tibetan, South Indian, North Indian  0.000000e+00
6                        Cuisines_Afghani, North Indian  0.000000e+00
1820      Cuisines_Turkish, Arabian, Moroccan, Lebanese  0.000000e+00
5                    Cuisines_Afghani, Mughlai, Chinese  0.000000e+00
1350  Cuisines_North Indian, Chinese, Continental, S… -1.932013e-20

[1842 rows x 2 columns]
```

[243]:
```
# Set threshold value
threshold = 0.03

# Select features with importance above the threshold
selected_features = feature_importance_df[feature_importance_df['Importance'] >␣
 ↪threshold]

print("Selected Features:")
print(selected_features)

# If you want to use the selected features for your dataset X
X_selected = X[selected_features['Feature']]
```

```
Selected Features:
                    Feature  Importance
1835     Rating color_White    0.314169
1839  Rating text_Not rated    0.314049
2                     Votes    0.272615
1833    Rating color_Orange    0.051801
```

splitting data into training and testing:

- Training Set : Used to train the model (learn patterns)
- Testing Set : Used to evaluate the models performance on unseen data. its used for preventing overfit and underfit

[244]:
```
X_train,X_test,y_train,y_test=train_test_split(X_selected, y, test_size=0.2,␣
↪random_state=42)
print("Shape of X_train:",X_train.shape)
print("Shape of X_test:",X_test.shape)
print("Shape of y_train:",y_train.shape)
print("Shape of y_test:",y_test.shape)
```

```
Shape of X_train: (7633, 4)
Shape of X_test: (1909, 4)
Shape of y_train: (7633,)
Shape of y_test: (1909,)
```

Feature Scaling

[245]:
```
scalar=StandardScaler()
X_train_scaled=scalar.fit_transform(X_train)
X_test_scaled=scalar.fit_transform(X_test)
```

[246]:
```
# Initialize models
models = {
    "Linear Regression": LinearRegression(),
    "Decision Tree Regression": DecisionTreeRegressor(random_state=42),
    "Random Forest Regression": RandomForestRegressor(random_state=42)
}

# Create an empty list to store results
results = []

# Train and evaluate each model
for name, model in models.items():
    model.fit(X_train_scaled, y_train)
    y_pred = model.predict(X_test_scaled)

    mse = mean_squared_error(y_test, y_pred)
    mae = mean_absolute_error(y_test, y_pred)
    r2 = r2_score(y_test, y_pred)

    results.append({
        "Model": name,
        "Mean Squared Error": mse,
        "Mean Absolute Error": mae,
        "r2_score": r2
    })
```

```
# Convert results to DataFrame
results_df = pd.DataFrame(results)

# Show the final comparison table
print(results_df)
```

```
                     Model  Mean Squared Error  Mean Absolute Error  r2_score
0        Linear Regression            0.108108             0.206188  0.952792
1  Decision Tree Regression            0.122072             0.220610  0.946694
2  Random Forest Regression            0.235279             0.322596  0.897260
```

- The results shows that Linear regression is the best with model MSE (0.1081) and MAE (0.2062) and r2 score (0.9528)
- Decision Tree Regression came second with very good performance, just slightly below Linear Regression.
- List item

[247]:
```
# finding the best model
best_model = results_df.iloc[0]["Model"]
best_model
```

[247]:
```
'Linear Regression'
```

[248]:
```
# save the model
import joblib
joblib.dump(best_model,"best_model.pkl")
print("Best model is saved as best_model.pkl")
```

```
Best model is saved as best_model.pkl
```

Customer Preference Analysis

[249]:
```
data
```

[249]:
```
      Restaurant ID           Restaurant Name  Country Code              City  \
0           6317637          Le Petit Souffle           162       Makati City
1           6304287          Izakaya Kikufuji           162       Makati City
2           6300002    Heat - Edsa Shangri-La           162  Mandaluyong City
3           6318506                      Ooma           162  Mandaluyong City
4           6314302              Sambo Kojin           162  Mandaluyong City
…             …                       …             …               …
9546        5915730              Naml  Gurme           208          stanbul
9547        5908749             Ceviz A ac           208          stanbul
9548        5915807                   Huqqa           208          stanbul
9549        5916112              A  k Kahve           208          stanbul
9550        5927402  Walter's Coffee Roastery           208          stanbul
```

```
                                                Address  \
0     Third Floor, Century City Mall, Kalayaan Avenu…
1     Little Tokyo, 2277 Chino Roces Avenue, Legaspi…
2     Edsa Shangri-La, 1 Garden Way, Ortigas, Mandal…
3     Third Floor, Mega Fashion Hall, SM Megamall, O…
4     Third Floor, Mega Atrium, SM Megamall, Ortigas…
…                                                   …
9546  Kemanke  Karamustafa Pa a Mahallesi, R ht m …
9547  Ko uyolu Mahallesi, Muhittin  st _nda  Cadd…
9548  Kuru _e me Mahallesi, Muallim Naci Caddesi, N…
9549  Kuru _e me Mahallesi, Muallim Naci Caddesi, N…
9550  Cafea a Mahallesi, Bademalt  Sokak, No 21/B, …

                                      Locality  \
0      Century City Mall, Poblacion, Makati City
1     Little Tokyo, Legaspi Village, Makati City
2     Edsa Shangri-La, Ortigas, Mandaluyong City
3         SM Megamall, Ortigas, Mandaluyong City
4         SM Megamall, Ortigas, Mandaluyong City
…                                             …
9546                                   Karak _y
9547                                  Ko uyolu
9548                                 Kuru _e me
9549                                 Kuru _e me
9550                                        Moda

                                    Locality Verbose   Longitude  \
0     Century City Mall, Poblacion, Makati City, Mak…  121.027535
1     Little Tokyo, Legaspi Village, Makati City, Ma…  121.014101
2     Edsa Shangri-La, Ortigas, Mandaluyong City, Ma…  121.056831
3     SM Megamall, Ortigas, Mandaluyong City, Mandal…  121.056475
4     SM Megamall, Ortigas, Mandaluyong City, Mandal…  121.057508
…                                                   …         …
9546                                Karak _y,  stanbul   28.977392
9547                               Ko uyolu,  stanbul   29.041297
9548                              Kuru _e me,  stanbul   29.034640
9549                              Kuru _e me,  stanbul   29.036019
9550                                     Moda,  stanbul   29.026016

       Latitude                          Cuisines  …          Currency  \
0     14.565443        French, Japanese, Desserts  …  Botswana Pula(P)
1     14.553708                          Japanese  …  Botswana Pula(P)
2     14.581404  Seafood, Asian, Filipino, Indian  …  Botswana Pula(P)
3     14.585318                  Japanese, Sushi  …  Botswana Pula(P)
4     14.584450                 Japanese, Korean  …  Botswana Pula(P)
…         …                                    …  …               …
9546  41.022793                          Turkish  …  Turkish Lira(TL)
```

```
9547  41.009847   World Cuisine, Patisserie, Cafe  …  Turkish Lira(TL)
9548  41.055817            Italian, World Cuisine  …  Turkish Lira(TL)
9549  41.057979                  Restaurant Cafe  …  Turkish Lira(TL)
9550  40.984776                             Cafe  …  Turkish Lira(TL)

     Has Table booking Has Online delivery Is delivering now  \
0                  Yes                  No                No
1                  Yes                  No                No
2                  Yes                  No                No
3                   No                  No                No
4                  Yes                  No                No
…                  …                    …                 …
9546                No                  No                No
9547                No                  No                No
9548                No                  No                No
9549                No                  No                No
9550                No                  No                No

     Switch to order menu Price range  Aggregate rating  Rating color  \
0                      No           3               4.8    Dark Green
1                      No           3               4.5    Dark Green
2                      No           4               4.4         Green
3                      No           4               4.9    Dark Green
4                      No           4               4.8    Dark Green
…                     …         …                 …           …
9546                   No           3               4.1         Green
9547                   No           3               4.2         Green
9548                   No           4               3.7        Yellow
9549                   No           4               4.0         Green
9550                   No           2               4.0         Green

     Rating text Votes
0      Excellent   314
1      Excellent   591
2      Very Good   270
3      Excellent   365
4      Excellent   229
…          …   …
9546   Very Good   788
9547   Very Good  1034
9548        Good   661
9549   Very Good   901
9550   Very Good   591

[9551 rows x 21 columns]
```

[250]:
```
data.head(10)
```

[250]:

	Restaurant ID	Restaurant Name	Country Code	City	Address	Locality	Locality Verbose	Longitude	Latitude
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
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
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
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
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
5	18189371	Din Tai Fung	162	Mandaluyong City	Ground Floor, Mega Fashion Hall, SM Megamall, …	SM Megamall, Ortigas, Mandaluyong City	SM Megamall, Ortigas, Mandaluyong City, Mandal…	121.056314	14.583764
6	6300781	Buffet 101	162	Pasay City	Building K, SM By The Bay, Sunset Boulevard, M…	SM by the Bay, Mall of Asia Complex, Pasay City	SM by the Bay, Mall of Asia Complex, Pasay Cit…	120.979667	14.531333
7	6301290	Vikings	162	Pasay City	Building B, By The Bay, Seaside Boulevard, Mal…	SM by the Bay, Mall of Asia Complex, Pasay City	SM by the Bay, Mall of Asia Complex, Pasay Cit…	120.979333	14.540000
8	6300010	Spiral - Sofitel Philippine Plaza Manila	162	Pasay City	Plaza Level, Sofitel Philippine Plaza Manila, …	Sofitel Philippine Plaza Manila, Pasay City	Sofitel Philippine Plaza Manila, Pasay City, P…	120.980090	14.552990
9	6314987	Locavore	162	Pasig City	Brixton Technology Center, 10 Brixton Street, …	Kapitolyo	Kapitolyo, Pasig City	121.056532	14.572041

```
                            Cuisines  …          Currency  \
0          French, Japanese, Desserts  …  Botswana Pula(P)
1                              Japanese  …  Botswana Pula(P)
2    Seafood, Asian, Filipino, Indian  …  Botswana Pula(P)
3                       Japanese, Sushi  …  Botswana Pula(P)
4                     Japanese, Korean  …  Botswana Pula(P)
5                              Chinese  …  Botswana Pula(P)
6                       Asian, European  …  Botswana Pula(P)
7  Seafood, Filipino, Asian, European  …  Botswana Pula(P)
8              European, Asian, Indian  …  Botswana Pula(P)
9                              Filipino  …  Botswana Pula(P)

  Has Table booking Has Online delivery Is delivering now  \
0               Yes                  No                No
1               Yes                  No                No
2               Yes                  No                No
3                No                  No                No
4               Yes                  No                No
5                No                  No                No
6               Yes                  No                No
7               Yes                  No                No
8               Yes                  No                No
9               Yes                  No                No

  Switch to order menu Price range  Aggregate rating  Rating color  \
0                   No           3               4.8    Dark Green
1                   No           3               4.5    Dark Green
2                   No           4               4.4         Green
3                   No           4               4.9    Dark Green
4                   No           4               4.8    Dark Green
5                   No           3               4.4         Green
6                   No           4               4.0         Green
7                   No           4               4.2         Green
8                   No           4               4.9    Dark Green
9                   No           3               4.8    Dark Green

  Rating text Votes
0   Excellent   314
1   Excellent   591
2   Very Good   270
3   Excellent   365
4   Excellent   229
5   Very Good   336
6   Very Good   520
7   Very Good   677
8   Excellent   621
```

```
 9   Excellent   532

[10 rows x 21 columns]
```

[251]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9551 entries, 0 to 9550
Data columns (total 21 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   Restaurant ID         9551 non-null   int64
 1   Restaurant Name       9551 non-null   object
 2   Country Code          9551 non-null   int64
 3   City                  9551 non-null   object
 4   Address               9551 non-null   object
 5   Locality              9551 non-null   object
 6   Locality Verbose      9551 non-null   object
 7   Longitude             9551 non-null   float64
 8   Latitude              9551 non-null   float64
 9   Cuisines              9542 non-null   object
 10  Average Cost for two  9551 non-null   int64
 11  Currency              9551 non-null   object
 12  Has Table booking     9551 non-null   object
 13  Has Online delivery   9551 non-null   object
 14  Is delivering now     9551 non-null   object
 15  Switch to order menu  9551 non-null   object
 16  Price range           9551 non-null   int64
 17  Aggregate rating      9551 non-null   float64
 18  Rating color          9551 non-null   object
 19  Rating text           9551 non-null   object
 20  Votes                 9551 non-null   int64
dtypes: float64(3), int64(5), object(13)
memory usage: 1.5+ MB
```

[252]:
```
data[['Cuisines', 'Aggregate rating']].head()
```

[252]:
```
                          Cuisines  Aggregate rating
0        French, Japanese, Desserts               4.8
1                          Japanese               4.5
2  Seafood, Asian, Filipino, Indian               4.4
3                   Japanese, Sushi               4.9
4                  Japanese, Korean               4.8
```

[253]:
```
cuisine_count=data['Cuisines'].value_counts()
cuisine_count
```

[253]:
```
Cuisines
North Indian                              936
North Indian, Chinese                     511
Chinese                                   354
Fast Food                                 354
North Indian, Mughlai                     334
                                          ...
World Cuisine, Patisserie, Cafe             1
Burger, Izgara                              1
Desserts, B _rek                            1
Restaurant Cafe, Turkish, Desserts          1
Restaurant Cafe, Desserts                   1
Name: count, Length: 1825, dtype: int64
```

[254]:
```
cuisine_rating=data.groupby('Cuisines')['Aggregate rating'].mean().
 ↪sort_values(ascending=False)
cuisine_rating
```

[254]:
```
Cuisines
Burger, Bar Food, Steak                   4.9
American, Burger, Grill                   4.9
American, Caribbean, Seafood              4.9
American, Coffee and Tea                  4.9
Mexican, American, Healthy Food           4.9
                                          ...
Tibetan, South Indian, North Indian       0.0
Afghani, Mughlai, Chinese                 0.0
Tibetan                                   0.0
Turkish, Arabian, Moroccan, Lebanese      0.0
Tibetan, Chinese, North Indian            0.0
Name: Aggregate rating, Length: 1825, dtype: float64
```

[255]:
```
cuisine_count = cuisine_count.reset_index()
cuisine_count.columns = ['Cuisines', 'count']

cuisine_rating = cuisine_rating.reset_index()
cuisine_rating.columns = ['Cuisines', 'Aggregate rating']

# Merge DataFrames on 'Cuisines'
cui_count = cuisine_count.merge(cuisine_rating, on="Cuisines")

# Sort and take top 10
cui_count.sort_values(by="count", ascending=False, inplace=True)
cuis_ch = cui_count.head(10)

# Show the result
cuis_ch
```

[255]:

	Cuisines	count	Aggregate rating
0	North Indian	936	1.672329
1	North Indian, Chinese	511	2.421722
2	Chinese	354	2.042090
3	Fast Food	354	2.118362
4	North Indian, Mughlai	334	2.888623
5	Cafe	299	2.890970
6	Bakery	218	1.924312
7	North Indian, Mughlai, Chinese	197	2.568528
8	Bakery, Desserts	170	2.317647
9	Street Food	149	2.161745

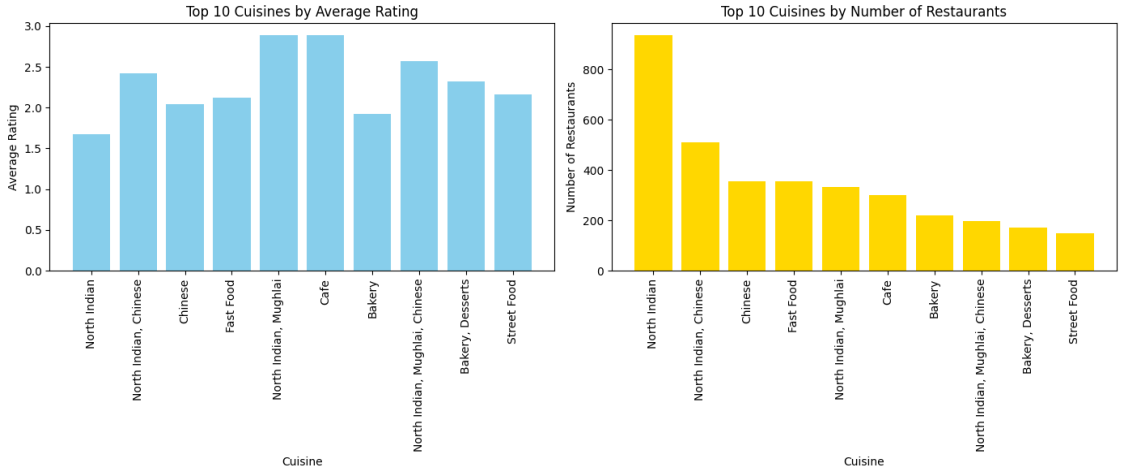
[256]:
```
import matplotlib.pyplot as plt

plt.figure(figsize=(14, 6))

# Bar chart for Average Rating
plt.subplot(1, 2, 1)
plt.bar(cuis_ch['Cuisines'], cuis_ch['Aggregate rating'], color='skyblue')
plt.xlabel("Cuisine")
plt.ylabel("Average Rating")
plt.title("Top 10 Cuisines by Average Rating")
plt.xticks(rotation=90)

# Bar chart for Number of Restaurants
plt.subplot(1, 2, 2)
plt.bar(cuis_ch['Cuisines'], cuis_ch['count'], color='gold')
plt.xlabel("Cuisine")
plt.ylabel("Number of Restaurants")
plt.title("Top 10 Cuisines by Number of Restaurants")
plt.xticks(rotation=90)

plt.tight_layout()
plt.show()
```



-
- North Indian cuisine dominates the market with a very high number of restaurants but their rating is low

1. Less available but well-rated cuisines (e.g., Bakery, Cafe, Mughlai) might be good opportunities for restaurant investment or marketing.
2. cuisine is available in many restaurants doesn't mean customers like it more.

[257]:
```
cuisine_by_vote=data.groupby('Cuisines')['Votes'].sum().reset_index()
cuisine_by_vote
```

[257]:

	Cuisines	Votes
0	Afghani	39
1	Afghani, Mughlai, Chinese	2
2	Afghani, North Indian	0
3	Afghani, North Indian, Pakistani, Arabian	3
4	African	373
…	…	…
1820	Western, Asian, Cafe	259
1821	Western, Fusion, Fast Food	32
1822	World Cuisine	95
1823	World Cuisine, Mexican, Italian	115
1824	World Cuisine, Patisserie, Cafe	1034

[1825 rows x 2 columns]

[258]:
```
popular_cuisines_sorted = cuisine_by_vote.sort_values(by='Votes',␣
↪ascending=False)
popular_cuisines_sorted
```

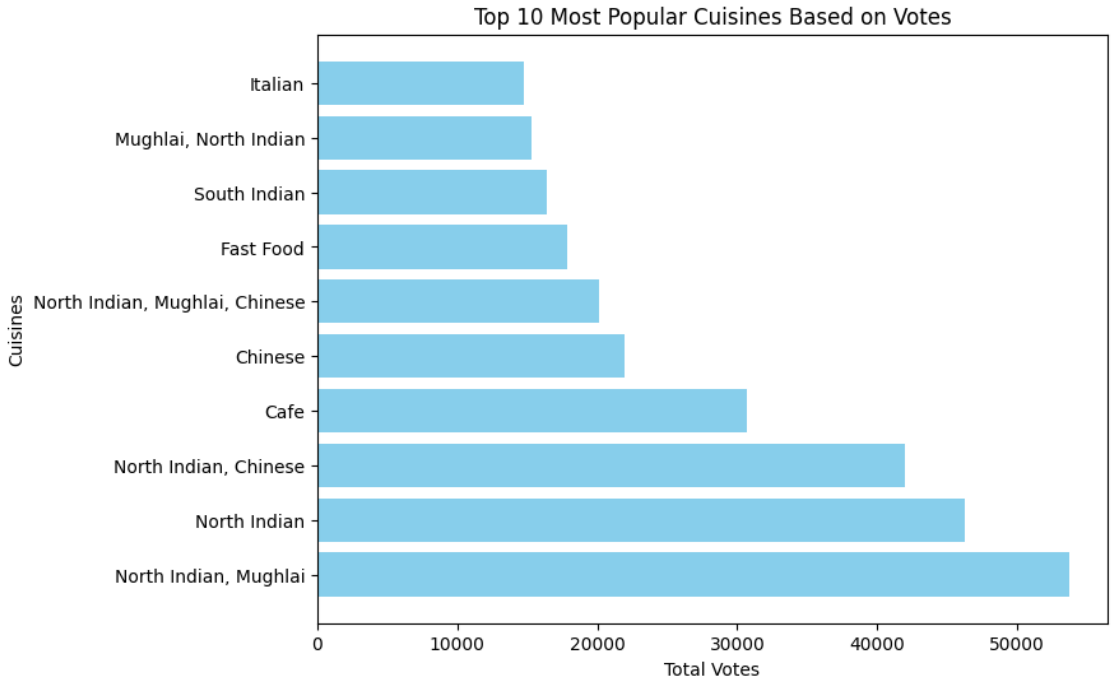
[258]:

	Cuisines	Votes
1514	North Indian, Mughlai	53747
1306	North Indian	46241
1329	North Indian, Chinese	42012
331	Cafe	30657
497	Chinese	21925
…	…	…
885	Fast Food, South Indian, Mithai	0
1509	North Indian, Mithai, Chinese	0
1259	Mithai, South Indian, Chinese, Street Food	0
1816	Turkish, Arabian, Moroccan, Lebanese	0
1811	Tibetan, South Indian, North Indian	0

[1825 rows x 2 columns]

[259]:
```
# top 10 cuisines

top_10_cuisines = popular_cuisines_sorted.head(10)
plt.figure(figsize=(8, 6))
plt.barh(top_10_cuisines['Cuisines'], top_10_cuisines['Votes'], color='skyblue')
plt.xlabel('Total Votes')
plt.ylabel('Cuisines')
plt.title('Top 10 Most Popular Cuisines Based on Votes')
plt.show()
```



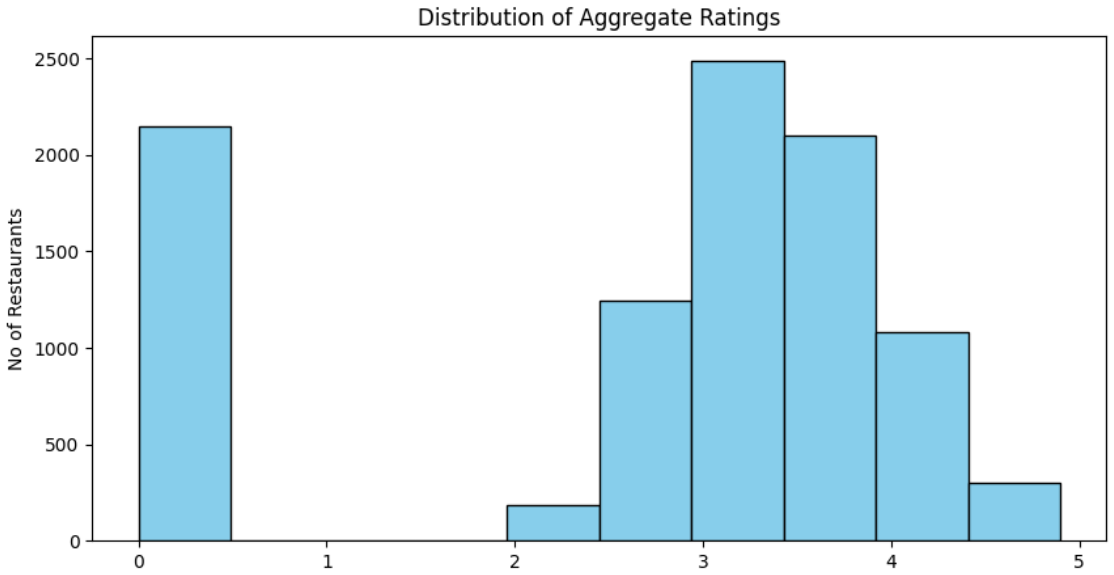
- The cuisnes 'North Indian, Mughlai' has the highest vote received from customer
- North Indian, Chinese, and Cafe also have high popularity.

[260]:
```
filtered_df= data.groupby("Cuisines").filter(lambda x: len(x) >= 10)
result = filtered_df.groupby("Cuisines")["Aggregate rating"].mean().
 ↪sort_values(ascending=False)
result_head = result.head(10)
result_head
```

[260]:
```
Cuisines
Modern Indian                 4.345455
Indian                        4.250000
Seafood                       4.114286
Thai                          4.100000
Cafe, Continental, Italian    4.080000
American, Burger              4.076923
Japanese, Sushi               4.044444
Pizza, Italian                3.668421
American                      3.667742
Italian                       3.657407
Name: Aggregate rating, dtype: float64
```

Level 3:Data Visualization

[261]:
```
plt.figure(figsize=(10, 5))
plt.hist(data['Aggregate rating'], bins=10, color='skyblue', edgecolor='black')
plt.title('Distribution of Aggregate Ratings')
plt.ylabel('No of Restaurants')
plt.show()
```



- A large number of restaurants have an aggregate rating of 0,that means they haven't received any ratings
- The peak of the distribution is in the 3.0 to 4.0 range.

[262]:
```
data['Aggregate rating'].value_counts().head(10)
```

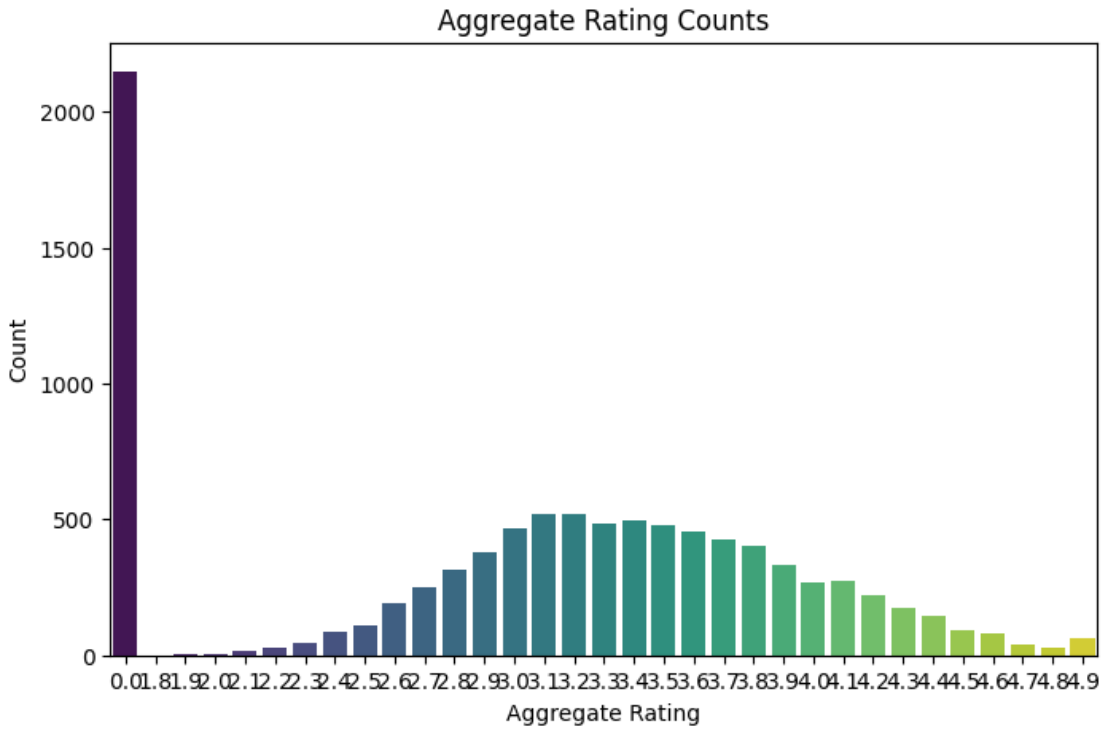
[262]:
```
Aggregate rating
0.0    2148
3.2     522
3.1     519
3.4     498
3.3     483
3.5     480
3.0     468
3.6     458
3.7     427
3.8     400
Name: count, dtype: int64
```

[264]:
```
plt.figure(figsize=(8, 5))
sns.countplot(x='Aggregate rating', data=data, palette='viridis')
plt.title('Aggregate Rating Counts')
plt.xlabel('Aggregate Rating')
plt.ylabel('Count')
plt.show()
```

```
<ipython-input-264-eb15dad9ff07>:2: 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.countplot(x='Aggregate rating', data=data, palette='viridis')
```



- There may be a bias toward average ratings, with extreme ratings being rare.
- Suggests that most customers rate moderately, and very few experiences are deemed either extremely poor or excellent.

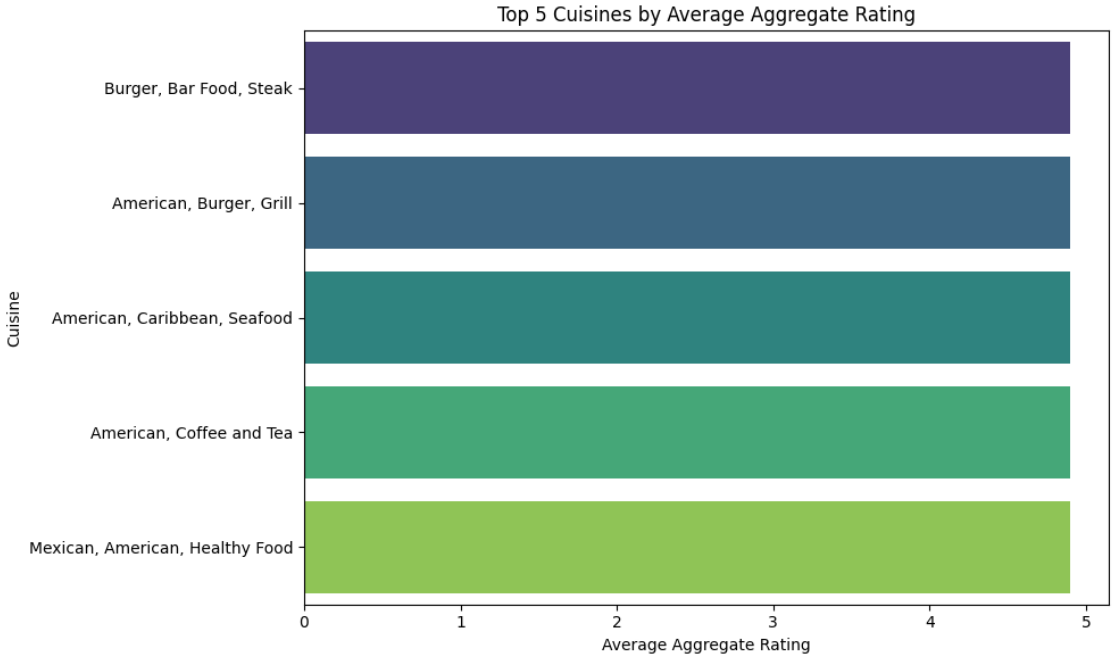
[265]:
```
cuisine_ratings=data[['Cuisines', 'Aggregate rating']].dropna()
cuisine_avg = cuisine_ratings.groupby('Cuisines')['Aggregate rating'].mean().
 ↪sort_values(ascending=False).head()
```

[266]:
```
plt.figure(figsize=(10, 6))
sns.barplot(x=cuisine_avg.values, y=cuisine_avg.index, palette='viridis')
plt.title('Top 5 Cuisines by Average Aggregate Rating')
plt.xlabel('Average Aggregate Rating')
plt.ylabel('Cuisine')
plt.tight_layout()
plt.show()
```

```
<ipython-input-266-66095e42994c>: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=cuisine_avg.values, y=cuisine_avg.index, palette='viridis')
```



[267]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9551 entries, 0 to 9550
Data columns (total 21 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   Restaurant ID         9551 non-null   int64
 1   Restaurant Name       9551 non-null   object
 2   Country Code          9551 non-null   int64
 3   City                  9551 non-null   object
 4   Address               9551 non-null   object
 5   Locality              9551 non-null   object
 6   Locality Verbose      9551 non-null   object
 7   Longitude             9551 non-null   float64
 8   Latitude              9551 non-null   float64
 9   Cuisines              9542 non-null   object
 10  Average Cost for two  9551 non-null   int64
 11  Currency              9551 non-null   object
 12  Has Table booking     9551 non-null   object
 13  Has Online delivery   9551 non-null   object
 14  Is delivering now     9551 non-null   object
 15  Switch to order menu  9551 non-null   object
 16  Price range           9551 non-null   int64
 17  Aggregate rating      9551 non-null   float64
 18  Rating color          9551 non-null   object
```

```
 19  Rating text           9551 non-null   object
 20  Votes                 9551 non-null   int64
dtypes: float64(3), int64(5), object(13)
memory usage: 1.5+ MB
```

- Restaurants offering these combinations are likely to attract higher customer satisfaction
- Platforms or food apps can promote restaurants with these cuisine profiles as "top-rated" or customer favorites

[268]:
```
average_of_two=data.groupby('Aggregate rating')['Average Cost for two'].mean().
↪sort_values(ascending=False)
average_of_two
```

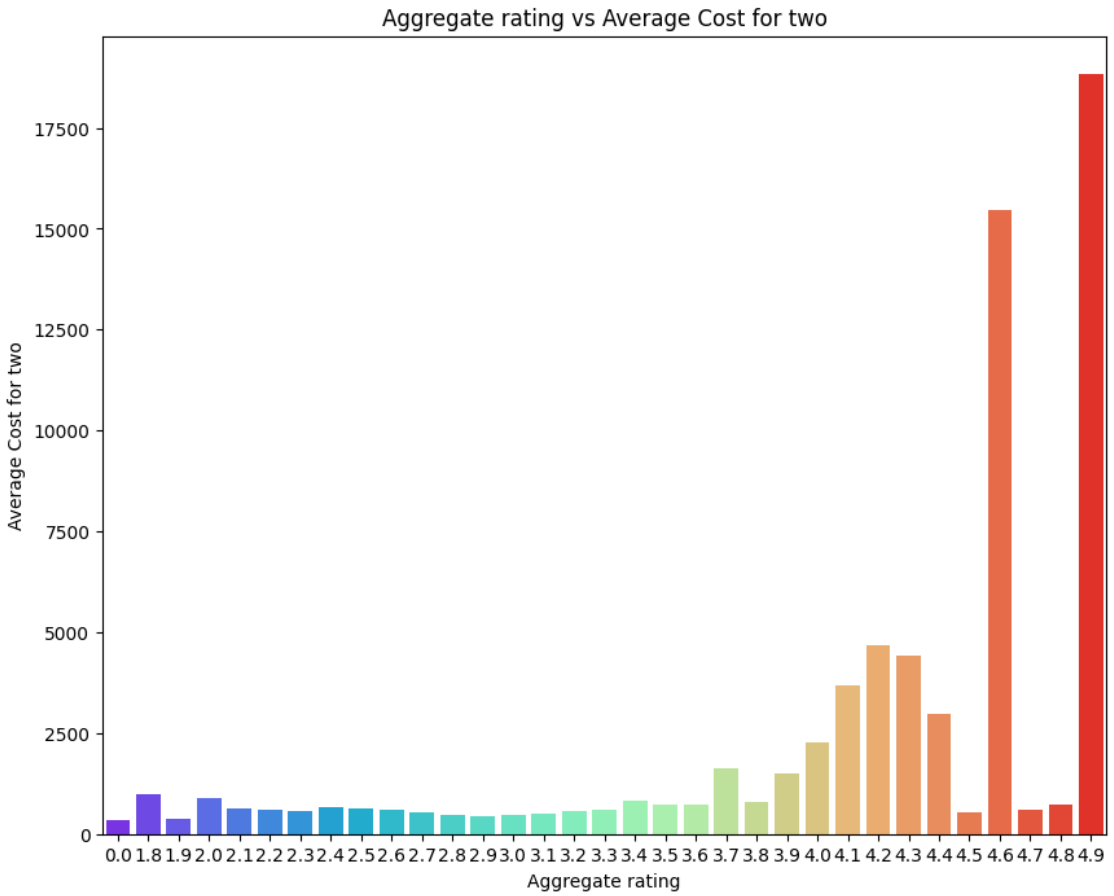
[268]:
```
Aggregate rating
4.9    18833.442623
4.6    15455.128205
4.2     4664.072398
4.3     4430.718391
4.1     3669.543796
4.4     2983.055556
4.0     2256.484962
3.7     1623.854801
3.9     1511.417910
1.8     1000.000000
2.0      892.857143
3.4      824.236948
3.8      792.835000
4.8      727.000000
3.5      726.270833
3.6      717.482533
2.4      670.919540
2.1      633.333333
2.5      622.272727
3.3      608.664596
2.6      601.675393
2.2      599.074074
4.7      597.380952
3.2      575.124521
2.3      565.957447
2.7      546.800000
4.5      542.263158
3.1      490.414258
2.8      480.190476
3.0      473.717949
2.9      450.446194
1.9      375.000000
0.0      340.337523
Name: Average Cost for two, dtype: float64
```

[269]:
```
plt.figure(figsize=(10,8))
sns.barplot(x=average_of_two.index,y=average_of_two.values,palette="rainbow")
plt.xlabel("Aggregate rating")
plt.ylabel("Average Cost for two")
plt.title("Aggregate rating vs Average Cost for two")
plt.show()
```

```
<ipython-input-269-dc6033f94107>:2: 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=average_of_two.index,y=average_of_two.values,palette="rainbow")
```



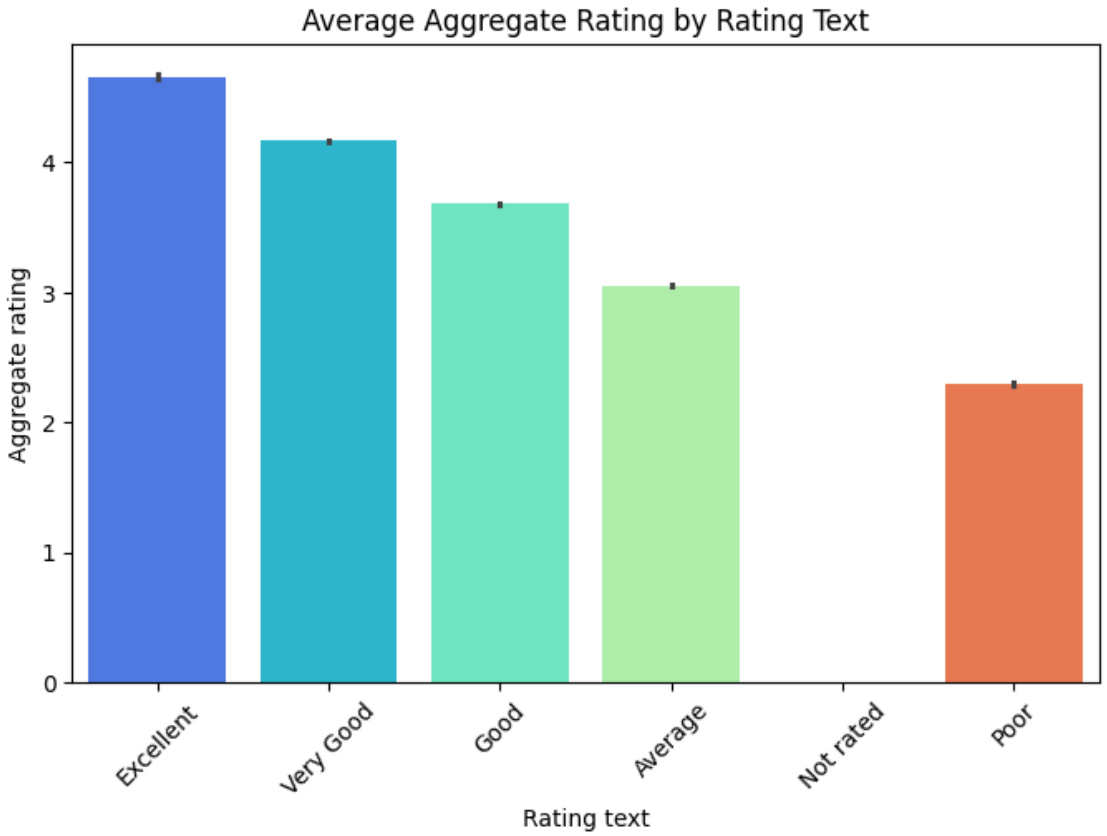
[279]:
```
rating_text=data['Rating text'].value_counts()
aggregate_rate=data['Aggregate rating'].value_counts()
```

[283]:
```
plt.figure(figsize=(8, 5))
sns.barplot(x='Rating text', y='Aggregate rating', data=data, estimator='mean',␣
 ↪palette='rainbow')
plt.title('Average Aggregate Rating by Rating Text')
plt.xticks(rotation=45)
plt.show()
```

```
<ipython-input-283-f0a0cbf73579>:2: 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='Rating text', y='Aggregate rating', data=data,
estimator='mean', palette='rainbow')
```



- most of the are unrated
- As the rating text improves the average aggregate rating increases.
- Excellent: Highest average rating, close to 4.7–4.9, indicating strong customer satisfaction.
- Restaurants aiming for a strong market presence should target the “Excellent” and “Very Good” zones, where the rating is above 4.0.

7 Conclusion:

- In this project, we investigated restaurant data with an emphasis on analyzing how various cuisines rank regarding average customer ratings and popularity. By categorizing data according to cuisine type, we could determine which cuisines receive the highest ratings and which are most frequently available.
- Using three regression models Random Forest, Decision Tree, and Linear Regression—predicted restaurant ratings based on a variety of factors. The linear regression model is the best model performed correlations in the , as evidenced by its greatest R2 score.
- The distribution and performance of different cuisines were examined using a variety of visualizations, including pie charts and bar graphs. These served to draw attention to the cuisines with the greatest and lowest average ratings, as well as the most and least popular. The entire analysis and the inferences made from the data were reinforced by the visual insights.
- All things considered, this analysis offers marketers, food delivery platforms, and restaurant owners useful information that they can use to inform data-driven choices about menu selections, advertising tactics, and service enhancements based on cuisine performance.