Restaurant Data Analysis

This project analyzes a restaurant dataset contained in "Restaurant.xlsx". The dataset likely includes information about restaurant names, locations, cuisines, prices, ratings, and other attributes relevant to food and hospitality. The primary aim is to extract useful insights and patterns relating to restaurant operations, customer preferences, and performance indicators

File Name : Restaurant.xlsx

Data Source : https://drive.google.com/file/d/1rGhBGoFJNWQ1HkFQLh06gIHZA7HaDtI7/view?usp=drive_link

Data Exploration

Importing required Python Libraries

In [7]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import pickle
```

Importing Dataset

In [9]:
```
Restaurant = pickle.load(open("Restaurant.pkl", "rb"))
Restaurant.head()
```

Out[9]:

	Restaurant ID	Restaurant Name	Country Code	City	Address	Locality	Localit Verbos
0	6317637	Le Petit Souffle	162	Makati City	Third Floor, Century City Mall, Kalayaan Avenu...	Century City Mall, Poblacion, Makati City	Century Cit Ma Poblacio Makati Cit Mak
1	6304287	Izakaya Kikufuji	162	Makati City	Little Tokyo, 2277 Chino Roces Avenue, Legaspi...	Little Tokyo, Legaspi Village, Makati City	Little Toky Legasp Villag Makati Cit Ma
2	6300002	Heat - Edsa Shangri-La	162	Mandaluyong City	Edsa Shangri-La, 1 Garden Way, Ortigas, Mandal...	Edsa Shangri-La, Ortigas, Mandaluyong City	Edsa Shangr La, Ortiga Mandaluyon City, Ma
3	6318506	Ooma	162	Mandaluyong City	Third Floor, Mega Fashion Hall, SM Megamall, O...	SM Megamall, Ortigas, Mandaluyong City	SM Megama Ortiga Mandaluyon Cit Mandal
4	6314302	Sambo Kojin	162	Mandaluyong City	Third Floor, Mega Atrium, SM Megamall, Ortigas...	SM Megamall, Ortigas, Mandaluyong City	SM Megama Ortiga Mandaluyon Cit Mandal

5 rows × 21 columns

Level 3

Task : 1 Restaurant Reviews

Text reviews provide rich, unstructured insights into customer experiences, opinions, and emotions that are not always visible through numeric ratings alone. By examining the language customers use in their feedback, it becomes possible to understand what

aspects of a restaurant they appreciate or dislike, and how these perceptions relate to overall satisfaction.

In this task, the focus is on analyzing the text reviews to identify the most common positive and negative keywords, as well as calculating the average length of reviews and exploring whether review length is related to rating. This analysis helps reveal key drivers of customer sentiment, supports service improvement, and enhances decision-making based on both qualitative and quantitative feedback.

Task 1.1 Analyze the text reviews to identify the most common positive and negative keywords.

In [14]:
```
print(Restaurant['Rating text'].unique())
```

```
['Excellent' 'Very Good' 'Good' 'Average' 'Not rated' 'Poor']
```

In [15]:
```
import pandas as pd
from collections import Counter
import nltk
from nltk.corpus import stopwords
import re

# Download stopwords (run once; safe to keep in code)
nltk.download('stopwords')

# Use English stopwords
Stop_Words = set(stopwords.words('english'))

# Combine all reviews into one big text
All_Reviews = ' '.join(Restaurant['Rating text'].dropna().astype(str))

# Remove punctuation and lowercase all words
All_Reviews = re.sub(r'[^\w\s]', '', All_Reviews).lower()

# Split text into words
Words = All_Reviews.split()

# Remove stopwords
Words = [Word for Word in Words if Word not in Stop_Words]

# Count frequencies of all words
Word_Counts = Counter(Words)

# 1) Table: Most common words overall
Most_Common_Words = Word_Counts.most_common(20)
Most_Common_Words_Table = pd.DataFrame(Most_Common_Words, columns=['Word', 'Coun
print("\nMost Common Words overall\n")
display(Most_Common_Words_Table)

# Define positive and negative word lists
Positive_Words = ['good', 'great', 'excellent', 'amazing', 'delicious', 'fantast
Negative_Words = ['bad', 'worst', 'terrible', 'awful', 'poor', 'slow', 'disappoi

# Find positive and negative words in the reviews
Positive_Counter = Counter([Word for Word in Words if Word in Positive_Words])
Negative_Counter = Counter([Word for Word in Words if Word in Negative_Words])
```

```
# 2) Table: Most common positive keywords
Positive_List = Positive_Counter.most_common()
Positive_Words_Table = pd.DataFrame(Positive_List, columns=['Positive Word', 'Co
print("\nMost common positive keywords\n")
display(Positive_Words_Table)

# 3) Table: Most common negative keywords
Negative_List = Negative_Counter.most_common()
Negative_Words_Table = pd.DataFrame(Negative_List, columns=['Negative Word', 'Co
print("\nMost common negative keywords\n")
display(Negative_Words_Table)
```

```
Most Common Words overall
```

```
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\admin\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
```

	Word	Count
0	average	3734
1	good	3174
2	rated	2148
3	excellent	300
4	poor	186

```
Most common positive keywords
```

	Positive Word	Count
0	good	3174
1	excellent	300

```
Most common negative keywords
```

	Negative Word	Count
0	poor	186

Understanding

The reviews show mostly positive sentiment, with “good” and “excellent” appearing far more often than negative words. “Poor” is the main negative term but occurs much less frequently, indicating overall customer satisfaction

Task 1.2 Calculate the average length of reviews and explore if there is a relationship between review length and rating.

In [18]:
```
# Measure the length of each review (number of characters)

Restaurant['Review Length'] = Restaurant['Rating text'].astype(str).apply(len)
```

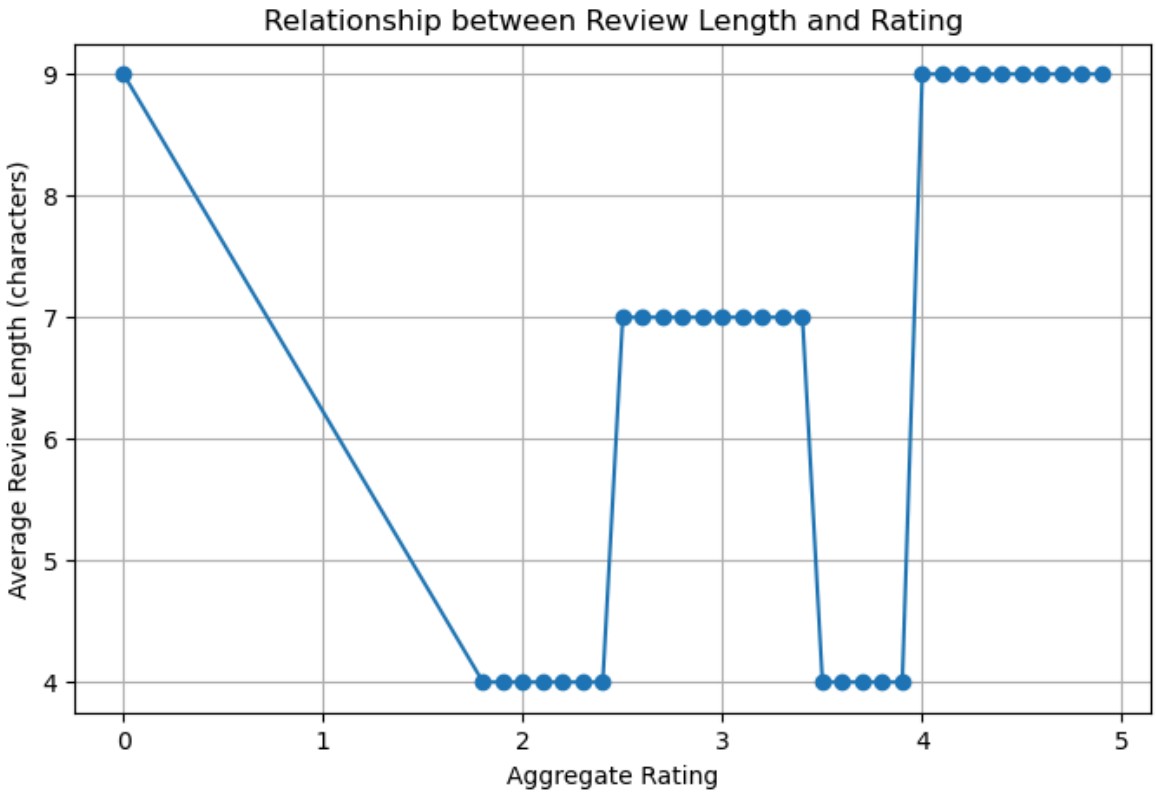
```
# Calculate and print the average review length

Average_Length = Restaurant['Review Length'].mean()

print("Average Review Length is : ", Average_Length)

# Explore the relationship between review length and rating

# Calculate average review length for each rating value

Length_by_Rating = Restaurant.groupby('Aggregate rating')['Review Length'].mean(

# Convert to table format

Length_by_Rating_Table = Length_by_Rating.reset_index()
Length_by_Rating_Table.columns = ['Aggregate Rating', 'Average Review Length']

print("\nAverage review length for each rating:\n")
Length_by_Rating_Table
```

```
Average Review Length is :  7.021588765457976

Average review length for each rating:
```

Out[18]:

	Aggregate Rating	Average Review Length
0	0.0	9.0
1	1.8	4.0
2	1.9	4.0
3	2.0	4.0
4	2.1	4.0
5	2.2	4.0
6	2.3	4.0
7	2.4	4.0
8	2.5	7.0
9	2.6	7.0
10	2.7	7.0
11	2.8	7.0
12	2.9	7.0
13	3.0	7.0
14	3.1	7.0
15	3.2	7.0
16	3.3	7.0
17	3.4	7.0
18	3.5	4.0
19	3.6	4.0
20	3.7	4.0
21	3.8	4.0
22	3.9	4.0
23	4.0	9.0
24	4.1	9.0
25	4.2	9.0
26	4.3	9.0
27	4.4	9.0
28	4.5	9.0
29	4.6	9.0
30	4.7	9.0
31	4.8	9.0
32	4.9	9.0

In [19]:
```
plt.figure(figsize = (8,5))

plt.plot(Length_by_Rating.index, Length_by_Rating.values, marker = 'o')

plt.xlabel('Aggregate Rating')
plt.ylabel('Average Review Length (characters)')
plt.title('Relationship between Review Length and Rating')

plt.grid(True)
plt.show()
```



Understanding

The average review length is about 7 characters, which shows that most reviews are extremely short. The plot also indicates no strong relationship between review length and rating; customers give high or low ratings even with very short text. Overall, rating does not depend on how long the review is.

Task : 2 Votes Analysis

Votes represent how many customers have engaged with a restaurant by sharing their feedback, making them an important indicator of visibility and popularity in the market. Examining both the highest and lowest vote counts helps highlight restaurants that attract significant attention as well as those that remain relatively unnoticed, offering insights into customer reach and engagement levels.

In this task, the goal is to identify the restaurants with the maximum and minimum number of votes and then analyze whether there is a relationship between the number of votes and the restaurant's rating. This analysis helps to understand if highly rated

restaurants naturally receive more votes, or if popularity (votes) and perceived quality (ratings) do not always align, providing useful guidance for performance evaluation and promotional strategies.

Task 2.1 Identify the restaurants with the highest and lowest number of votes.

In [24]:
```
# Find the restaurant with the highest number of votes

Highest_Votes = Restaurant.loc[Restaurant['Votes'].idxmax() , ['Restaurant Name'

# Find the restaurant with the lowest number of votes

Lowest_Votes = Restaurant.loc[Restaurant['Votes'].idxmin() , ['Restaurant Name'

# Combine into one table

Votes_Table = pd.DataFrame([
    {'Type' : 'Highest Votes' , 'Restaurant Name' : Highest_Votes['Restaurant Na
    {'Type' : 'Lowest Votes' ,  'Restaurant Name' : Lowest_Votes['Restaurant Nam
])

print("\nRestaurants with highest and lowest number of votes:\n")
display(Votes_Table)
```

```
Restaurants with highest and lowest number of votes:
```

	Type	Restaurant Name	Votes
0	Highest Votes	Toit	10934
1	Lowest Votes	Cantinho da Gula	0

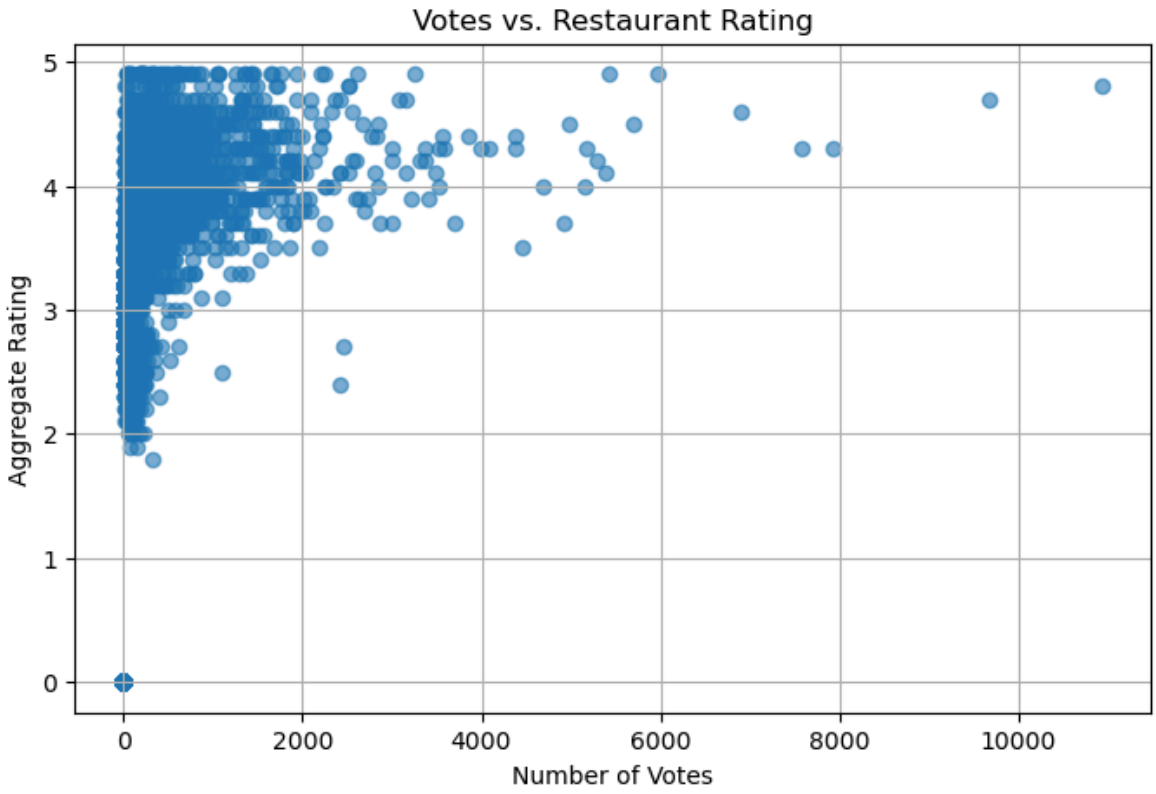
Understanding

The restaurant with the highest number of votes is Toti, with 10,934 votes, showing it is highly popular and frequently reviewed. The restaurant with the lowest number of votes is Cantinho da Gula, with 0 votes, indicating very little or no customer engagement.

Task 2.2 Analyze if there is a correlation between the number of votes and the rating of a restaurant.

In [27]:
```
# Plot a scatter plot to visualize the relationship

plt.figure(figsize = (8,5))
plt.scatter(Restaurant['Votes'] , Restaurant['Aggregate rating'] , alpha = 0.6)

plt.xlabel('Number of Votes')
plt.ylabel('Aggregate Rating')
plt.title('Votes vs. Restaurant Rating')

plt.grid(True)
plt.show()
```

```
# 5. Calculate the correlation coefficient

Correlation = Restaurant['Votes'].corr(Restaurant['Aggregate rating'])
print("\nCorrelation between votes and rating : " , Correlation)
```




```
Correlation between votes and rating :  0.31347418032500046
```

Understanding

There is a weak positive correlation between the number of votes and the restaurant's rating (0.31). This means restaurants with more votes tend to have slightly higher ratings, but the relationship is not very strong. Many restaurants have high ratings even with few votes, showing that rating does not depend heavily on vote count.

Task : 3 Price Range vs. Online Delivery and Table Booking

Price range and service availability are closely linked to how restaurants position themselves in the market and cater to different customer segments. Services like online delivery and table booking can enhance convenience and influence how customers perceive value, especially across different pricing levels

In this task, the objective is to analyze whether there is a relationship between a restaurant's price range and the availability of online delivery and table booking, and to determine if higher-priced restaurants are more likely to offer these services. This analysis helps in understanding how service offerings vary across budget and premium segments, supporting strategic decisions related to service design, competitiveness, and customer experience.

Task 3.1 Analyze if there is a relationship between the price range and the availability of online delivery and table booking

In [32]:
```
# Group data by price range to see the proportion of restaurants with delivery o

Unique_Prices = Restaurant['Price range'].unique()
Summary = []

for Price in Unique_Prices:
    Subset = Restaurant[Restaurant['Price range'] == Price]
    Total = len(Subset)
    Online = (Subset['Has Online delivery'] == 'Yes').mean()
    Booking = (Subset['Has Table booking'] == 'Yes').mean()
    Summary.append({
        'Price range': Price,
        'Total Restaurants': Total,
        'Online Delivery Proportion': round(Online * 100, 1),
        'Table Booking Proportion': round(Booking * 100, 1)
    })

Info = pd.DataFrame(Summary).set_index('Price range')
Info
```

Out[32]:

Price range	Total Restaurants	Online Delivery Proportion	Table Booking Proportion
3	1405	29.3	45.8
4	586	9.0	46.8
2	3113	41.3	7.7
1	4438	15.8	0.0

In [33]:
```
# Visualize these proportions for clarity

plt.figure(figsize = (8,5))

Info[['Online Delivery Proportion', 'Table Booking Proportion']].plot(kind = 'ba

plt.title('Online Delivery & Table Booking Availability by Price Range')
plt.xlabel('Price Range')
plt.ylabel('Proportion of Restaurants (%)')
plt.xticks(rotation = 0)

plt.grid(True)
plt.show()
```

```
<Figure size 800x500 with 0 Axes>
```

Understanding

Higher-priced restaurants (Price Range 3 & 4) offer table booking far more often, while online delivery is more common in mid-range restaurants (Price Range 2). Low-priced restaurants have the least availability of both services. This suggests that service availability varies significantly by price level—premium places focus on table booking, whereas mid-range places focus on delivery.

Task 3.2 Determine if higher-priced restaurants are more likely to offer these services.

In [36]:
```
# Count total restaurants per price range

Total_Counts = Restaurant['Price range'].value_counts().sort_index()

# Count those with online delivery

Delivery_Counts = Restaurant[Restaurant['Has Online delivery'] == 'Yes']['Price

# Count those with table booking

Booking_Counts = Restaurant[Restaurant['Has Table booking'] == 'Yes']['Price ran

# Assemble into a single DataFrame

Result = pd.DataFrame({
    'Total Restaurants': Total_Counts,
    'Online Delivery Fraction': (Delivery_Counts / Total_Counts).fillna(0).round
    'Table Booking Fraction': (Booking_Counts / Total_Counts).fillna(0).round(2)
```

```
})

# If you want percentages:

Result['Online Delivery Fraction'] = (Result['Online Delivery Fraction'] * 100).
Result['Table Booking Fraction'] = (Result['Table Booking Fraction'] * 100).roun

print("\nProportion of services by price range\n")
display(Result)
```

```
Proportion of services by price range
```

Price range	Total Restaurants	Online Delivery Fraction	Table Booking Fraction
1	4438	16.0	0.0
2	3113	41.0	8.0
3	1405	29.0	46.0
4	586	9.0	47.0

In [37]:
```
# Select just the fraction columns for plotting

Result[['Online Delivery Fraction' , 'Table Booking Fraction']].plot(kind = 'bar

plt.title('Service Availability by Price Range')
plt.xlabel('Price Range')
plt.ylabel('Proportion of Restaurants (%)')
plt.xticks(rotation = 0)
plt.legend(['Online Delivery' , 'Table Booking'])

plt.grid(axis = 'y' , linestyle = '--' , alpha = 0.6)


plt.show()
```

Understanding

Higher-priced restaurants (Price Range 3 and 4) consistently offer table booking far more often, as shown in both the table and the graph. Online delivery, however, is most common among mid-priced restaurants (Price Range 2), while lower-priced restaurants provide the fewest services overall. Overall, service availability increases with price range, with premium restaurants focusing mainly on dine-in booking and mid-range restaurants offering a balance of both delivery and booking services.

Conclusion

Most reviews are short and generally positive, with words like “good” and “excellent” appearing far more than negative ones. There is only a weak positive relationship between ratings and votes, meaning popularity doesn’t strongly affect rating. Higher-priced restaurants are much more likely to offer table booking, while mid-priced restaurants provide the most online delivery. Lower-priced restaurants offer the fewest services overall.