import pandas as pd import nltk from nltk.sentiment.vader import SentimentIntensityAnalyzer from nltk.tokenize import word_tokenize from collections import Counter

In [3]:
```
df=pd.read_csv("C:\\Users\\neeli\\Downloads\\Dataset .csv")
df.head()
```

Out[3]:

	Restaurant ID	Restaurant Name	Country Code	City	Address	Locality	Locality Verbose	Longitude	Latitude	Cuisines	...	Currency	Has Table booking	Has Online delivery	Is delivering now	Switch to order menu	Price range	Aggregate rating	Rating color	Rating text	Votes
0	6317637	Le Petit Souffle	162	Makati City	Third Floor, Century City Mall, Kalayaan Avenu...	Century City Mall, Poblacion, Makati City	Century City Mall, Poblacion, Makati City, Mak...	121.027535	14.565443	French, Japanese, Desserts	...	Botswana Pula(P)	Yes	No	No	No	3	4.8	Dark Green	Excellent	314
1	6304287	Izakaya Kikufuji	162	Makati City	Little Tokyo, 2277 Chino Roces Avenue, Legaspi...	Little Tokyo, Legaspi Village, Makati City	Little Tokyo, Legaspi Village, Makati City, Ma...	121.014101	14.553708	Japanese	...	Botswana Pula(P)	Yes	No	No	No	3	4.5	Dark Green	Excellent	591
2	6300002	Heat - Edsa Shangri-La	162	Mandaluyong City	Edsa Shangri-La, 1 Garden Way, Ortigas, Mandal...	Edsa Shangri-La, Ortigas, Mandaluyong City	Edsa Shangri-La, Ortigas, Mandaluyong City, Ma...	121.056831	14.581404	Seafood, Asian, Filipino, Indian	...	Botswana Pula(P)	Yes	No	No	No	4	4.4	Green	Very Good	270
3	6318506	Ooma	162	Mandaluyong City	Third Floor, Mega Fashion Hall, SM Megamall, O...	SM Megamall, Ortigas, Mandaluyong City	SM Megamall, Ortigas, Mandaluyong City, Mandal...	121.056475	14.585318	Japanese, Sushi	...	Botswana Pula(P)	No	No	No	No	4	4.9	Dark Green	Excellent	365
4	6314302	Sambo Kojin	162	Mandaluyong City	Third Floor, Mega Atrium, SM Megamall, Ortigas...	SM Megamall, Ortigas, Mandaluyong City	SM Megamall, Ortigas, Mandaluyong City, Mandal...	121.057508	14.584450	Japanese, Korean	...	Botswana Pula(P)	Yes	No	No	No	4	4.8	Dark Green	Excellent	229

5 rows × 21 columns

In [4]:
```
df.columns
```

Out[4]:
```
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')
```

In [5]:
```
#level-3
#task-1
#Resturant Reviews
#Analyze the text reviews to identify the most common positive and negative keywords
```

In [6]:
```
import pandas as pd
import nltk
from nltk.sentiment.vader import SentimentIntensityAnalyzer
from nltk.tokenize import word_tokenize
from collections import Counter
nltk.download("vader_lexicon")
sia = SentimentIntensityAnalyzer()
df["Sentiment Score"] = df["Rating text"].apply(lambda x: sia.polarity_scores(x)["compound"])
positive_reviews = df[df["Sentiment Score"] > 0]["Rating text"]
negative_reviews = df[df["Sentiment Score"] < 0]["Rating text"]
positive_tokens = [word.lower() for review in positive_reviews for word in word_tokenize(review)]
negative_tokens = [word.lower() for review in negative_reviews for word in word_tokenize(review)]
positive_keywords = Counter(positive_tokens).most_common(5)
negative_keywords = Counter(negative_tokens).most_common(5)
print("Most common positive keywords:")
print(positive_keywords)
print("\nMost common negative keywords:")
print(negative_keywords)
```

```
[nltk_data] Downloading package vader_lexicon to
[nltk_data]     C:\Users\neeli\AppData\Roaming\nltk_data...
[nltk_data]   Package vader_lexicon is already up-to-date!
Most common positive keywords:
[('good', 3179), ('very', 1079), ('excellent', 301)]

Most common negative keywords:
[('poor', 186)]
```

In [7]:
```
#Calculate the average length of reviews and explore if there is a relationship between review length and rating
```

In [8]:
```
import pandas as pd




df["Review Length"] = df["Rating text"].apply(lambda x: len(x.split()))


average_review_length = df["Review Length"].mean()

correlation = df["Review Length"].corr(df["Aggregate rating"])

print(f"Average review length: {average_review_length:.2f} words")
print(f"Correlation between review length and aggregate rating: {correlation:.2f}")
```

```
Average review length: 1.34 words
Correlation between review length and aggregate rating: -0.60
```

In [9]:
```
#task-2
#Identify the restaurants with the highest and lowest number of votes
```

In [10]:
```
import pandas as pd


# Sort restaurants by votes (ascending order)
df_sorted = df.sort_values(by="Votes")

# Restaurant with the lowest number of votes
lowest_votes_restaurant = df_sorted.iloc[0]

# Restaurant with the highest number of votes
highest_votes_restaurant = df_sorted.iloc[-1]

print(f"Lowest votes restaurant: {lowest_votes_restaurant['Restaurant Name']} ({lowest_votes_restaurant['Votes']} votes)")
print(f"Highest votes restaurant: {highest_votes_restaurant['Restaurant Name']} ({highest_votes_restaurant['Votes']} votes)")
```

```
Lowest votes restaurant: Khalsa Eating Point (0 votes)
Highest votes restaurant: Toit (10934 votes)
```

In [ ]:
```
#Analyze if there is a correlation between thenumber of votes and the rating of a restaurant
```

In [46]:
```
import pandas as pd
import numpy as np
correlation = np.corrcoef(df["Votes"], df["Aggregate rating"])[0, 1]
print(f"Correlation coefficient between votes and rating: {correlation:.2f}")
```

```
Correlation coefficient between votes and rating: 0.31
```

In [ ]:
```
#task-3
#Task: Price Range vs. Online Delivery and Table Booking
#Analyze if there is a relationship between the price range and the availability of online delivery and table booking
```

In [48]:
```
import pandas as pd
import numpy as np
df["Has Table booking"] = df["Has Table booking"].map({"Yes": 1, "No": 0})
df["Has Online delivery"] = df["Has Online delivery"].map({"Yes": 1, "No": 0})
correlation_table_booking = np.corrcoef(df["Price range"], df["Has Table booking"])[0, 1]
correlation_online_delivery = np.corrcoef(df["Price range"], df["Has Online delivery"])[0, 1]
print(f"Correlation with Table booking: {correlation_table_booking:.2f}")
print(f"Correlation with Online delivery: {correlation_online_delivery:.2f}")
```

```
Correlation with Table booking: 0.50
Correlation with Online delivery: 0.08
```

In [ ]:
```
#Determine if higher-priced restaurants are more likely to offer these services
```

In [49]:
```
import pandas as pd
grouped = df.groupby("Price range").agg({
    "Has Table booking": "mean",
    "Has Online delivery": "mean"
})

print(grouped)
```

```
             Has Table booking  Has Online delivery
Price range
1                     0.000225             0.157741
2                     0.076775             0.413106
3                     0.457386             0.291903
4                     0.467577             0.090444
```

In [ ]: