PROJECT OVERVIEW

RESTAURANT DATA ANALYSIS PROJECT

Restaurant datasets encompass various attributes such as names, locations, cuisine types, ratings, review counts, price ranges, and operating hours. They are sourced from online review platforms, food delivery apps, and restaurant websites. Analyses can include descriptive summaries, sentiment analysis, predictive modeling, and geospatial mapping. These insights help restaurant owners improve services, understand customer preferences, and conduct market research. Common challenges involve ensuring data quality, maintaining privacy, and integrating diverse data sources effectively.

```
# importing necessary libraries for data analysis and visualization
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

```
# Loading the restaurant dataset into a dataframe
dataset = pd.read_csv('/content/Dataset  (1).csv')
```

```
# Displaying the first few rows of the dataset to understand the structure
dataset.head()
```

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

5 rows × 21 columns

```
# Displaying nubers of rows and columns in the dataset
dataset.shape
```

```
(9551, 21)
```

```
# Displaying summary of the datset
dataset.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
```

```
# Checking for missing values
dataset.isnull().sum()
```

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

dtype: int64

LEVEL - 3

TASK 1- Restaurant Reviews

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

```
from collections import Counter
import re
```

```
# Extract and clean the 'Rating text' columns
review = dataset['Rating text'].dropna().tolist()
```

```
print(review)
```

```
['Excellent', 'Excellent', 'Very Good', 'Excellent', 'Excellent', 'Very Good', 'Very Good', 'Very Good', 'Excellent', 'Excellent', 'Exce
```

```
# Define function to tokenize and clean text
def tokenize(text):
  text = text.lower() # convert to lowercase
  text = re.sub(r'[^a-z\s]','',text) # Remove non-alphabetic charcters
  tokens = text.split() # split into words
  return tokens

# Tokenize all reviews
all_tokens = []
for r in review:
  all_tokens.extend(tokenize(r))

# count the frequency of each token
token_counts = Counter(all_tokens)

# Display the most common tokens
print("common tokens:",token_counts.most_common(20))

# Define a list of words to ignore
ignore_word = {'rated','very'}

# filtered the ignored words
filtered_counts = {word: count for word, count in token_counts.items() if word not in ignore_word}

# seperate positive and negative keywords
positive_words = {'good','excellent'}
negative_words = {'poor','not'}

# Get frequency of positive and negative keywords
positive_counts = {word: filtered_counts[word] for word in positive_words if word in filtered_counts}
negative_counts = {word: filtered_counts[word] for word in negative_words if word in filtered_counts}

# Display the counts of positive and negative keywords
print('positive words:',positive_counts)
print('negative words:',negative_counts)
```

```
common tokens: [('average', 3737), ('good', 3179), ('not', 2148), ('rated', 2148), ('very', 1079), ('excellent', 301), ('poor', 186)]
positive words: {'good': 3179, 'excellent': 301}
negative words: {'not': 2148, 'poor': 186}
```

CALCULATE THE AVERAGE LENGTH OF REVIEWS AND EXPLORE IF THERE IS A RELATIONSHIP BETWEEN REVIEW LENGTH AND RATING

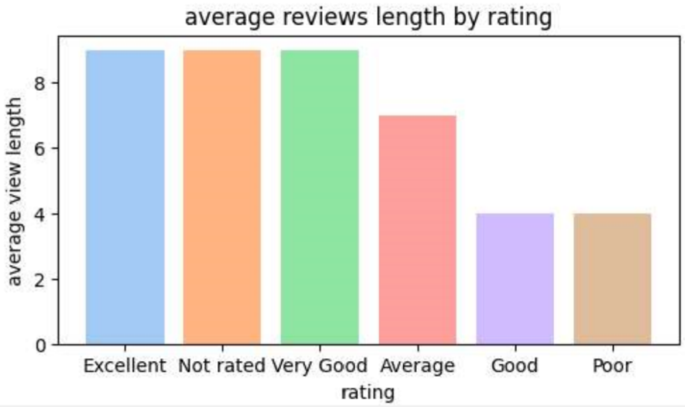
```
dataset['reviews'] = dataset['Rating text'].dropna().str.len()
average_length_review = dataset['reviews'].mean()
print(f"Average length of Reviews:  {average_length_review:.2f} characters")
```

```
Average length of Reviews:  7.02 characters
```

```
rating_length_mean = dataset.groupby("Rating text")['reviews'].mean().sort_values(ascending = False).reset_index()
print(rating_length_mean)
```

```
  Rating text  reviews
0   Excellent      9.0
1   Not rated      9.0
2   Very Good      9.0
3     Average      7.0
4        Good      4.0
5        Poor      4.0
```

```
plt.figure(figsize = (6,3))
colors = sns.color_palette('pastel')
plt.bar(rating_length_mean['Rating text'],rating_length_mean['reviews'],color = colors)
plt.xlabel('rating')
plt.ylabel('average view length')
plt.title('average reviews length by rating')
plt.show()
```



TASK 2 - VOTE ANALYSIS

Identify the restaurant with the highest and lowest number of votes

```
highest_votes = dataset.groupby('Restaurant Name')['Votes'].max().sort_values(ascending=False)
lowest_votes = dataset.groupby('Restaurant Name')['Votes'].min().sort_values(ascending = True)
print("HIGHEST_VOTES",highest_votes)
print("LOWEST_VOTES",lowest_votes)
```

```
HIGHEST_VOTES Restaurant Name
Toit                            10934
Truffles                         9667
Hauz Khas Social                 7931
Peter Cat                        7574
AB's - Absolute Barbecues        6907
                                 ...
Laxmi Dairy                         0
Delhi Foods                         0
Annapurna Caterings                 0
Smily Cakes                         0
Smoke Trailer Grill                 0
Name: Votes, Length: 7446, dtype: int64
LOWEST_VOTES Restaurant Name
Laxmi Food Corner                     0
Healthy Nutrienty                     0
Costa Coffee                          0
The Yolmo Kitchen                     0
Raju De Special Paneer Wale           0
                                    ...
The Black Pearl                    5385
Big Brewsky                        5705
Peter Cat                          7574
Hauz Khas Social                   7931
Toit                              10934
Name: Votes, Length: 7446, dtype: int64
```

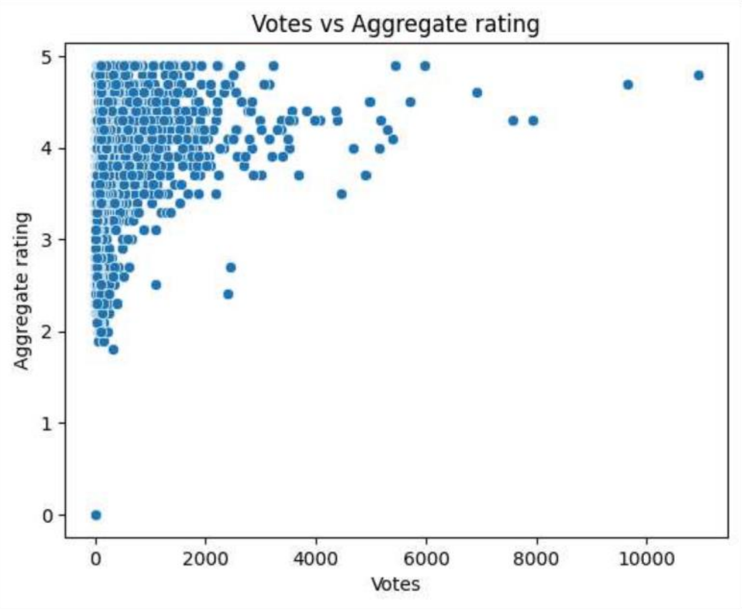
ANALYZE IF THERE IS A CORRELTION BETWEEN THE NUMBER OF VOTES AND THE RATING OF A RESTAURANT

```
correlation = dataset[['Aggregate rating','Votes']].corr()
```

```
print(correlation)
```

```
                  Aggregate rating     Votes
Aggregate rating          1.000000  0.313691
Votes                     0.313691  1.000000
```

```
sns.scatterplot(data = dataset,x='Votes',y='Aggregate rating')
plt.title("Votes vs Aggregate rating")
plt.show()
```



TASK 3 - PRICE RANGE VS ONLINE DELIVERY AND TABLE BOOKING

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

```
# checking the unique values in these columns
print(dataset['Price range'].unique())
print(dataset['Has Online delivery'].unique())
print(dataset['Has Table booking'].unique())
```

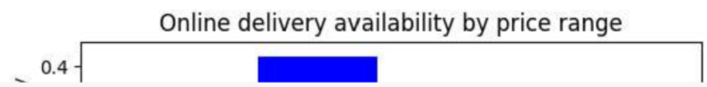
```
[3 4 2 1]
[0 1]
[1 0]
```

```
relation = dataset.groupby('Price range')[['Has Online delivery','Has Table booking']].mean()
```

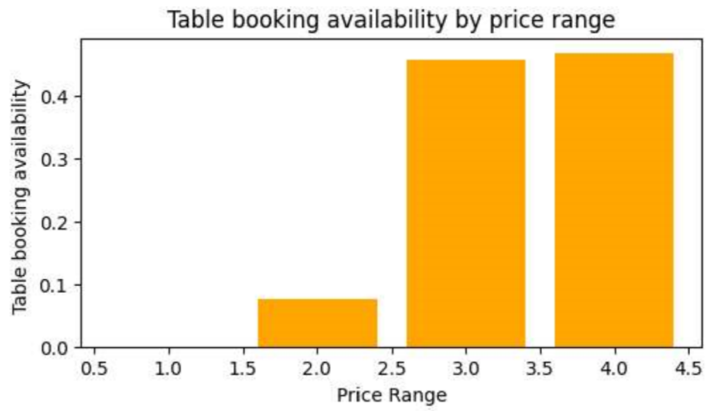
```
relation
```

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

```
plt.figure(figsize = (6,3))
plt.bar(relation.index,relation['Has Online delivery'],color='blue')
plt.xlabel('Price Range')
plt.ylabel('Online delivery availability ')
plt.title('Online delivery availability by price range')
plt.show()
```



```
plt.figure(figsize = (6,3))
plt.bar(relation.index,relation['Has Table booking'],color='Orange')
plt.xlabel('Price Range')
plt.ylabel('Table booking availability ')
plt.title('Table booking availability by price range')
plt.show()
```



DETERMINE IF HIGHER-PRICED RESTAURANTS ARE MORE LIKELY TO OFFER THESE SERVICES

```
high_price = dataset[dataset['Price range'] == 4]
offer_delivery = high_price.groupby('Has Table booking')['Price range'].count()
print(offer_delivery)
```

```
Has Table booking
0    312
1    274
Name: Price range, dtype: int64
```

```
high_range = dataset[dataset['Price range'] == 4]
offer = high_range.groupby('Has Online delivery')['Price range'].count()
print(offer)
```

```
Has Online delivery
0    533
1     53
Name: Price range, dtype: int64
```

```
Start coding or generate with AI.
```