cognify-l1-t1

January 20, 2024

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

[ ]:
```
df = pd.read_csv("./L1T1_Dataset.csv")
df.head()
```

[ ]:
```
   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

                                              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…

                                    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

                                  Locality Verbose   Longitude   Latitude  \
0  Century City Mall, Poblacion, Makati City, Mak…  121.027535  14.565443
1  Little Tokyo, Legaspi Village, Makati City, Ma…  121.014101  14.553708
2  Edsa Shangri-La, Ortigas, Mandaluyong City, Ma…  121.056831  14.581404
3  SM Megamall, Ortigas, Mandaluyong City, Mandal…  121.056475  14.585318
4  SM Megamall, Ortigas, Mandaluyong City, Mandal…  121.057508  14.584450

                          Cuisines  …          Currency Has Table booking  \
```

```
0        French, Japanese, Desserts  …  Botswana Pula(P)               Yes
1                            Japanese  …  Botswana Pula(P)               Yes
2  Seafood, Asian, Filipino, Indian  …  Botswana Pula(P)               Yes
3                   Japanese, Sushi  …  Botswana Pula(P)                No
4                  Japanese, Korean  …  Botswana Pula(P)               Yes

  Has Online delivery Is delivering now Switch to order menu Price range  \
0                  No                No                    No           3
1                  No                No                    No           3
2                  No                No                    No           4
3                  No                No                    No           4
4                  No                No                    No           4

   Aggregate rating  Rating color Rating text Votes
0               4.8    Dark Green   Excellent   314
1               4.5    Dark Green   Excellent   591
2               4.4         Green   Very Good   270
3               4.9    Dark Green   Excellent   365
4               4.8    Dark Green   Excellent   229

[5 rows x 21 columns]
```

```
Warning: Total number of columns (21) exceeds max_columns (20) limiting to first
(20) columns.
```

[ ]:
```
df.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
```

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

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

[ ]:
```
columns_df = list(df.columns)
number_of_columns = len(df.columns)
```

[ ]:
```
for i in range(0, len(columns_df)):
  print(i+1, columns_df[i])
```

```
1 Restaurant ID
2 Restaurant Name
3 Country Code
4 City
5 Address
6 Locality
7 Locality Verbose
8 Longitude
9 Latitude
10 Cuisines
11 Average Cost for two
12 Currency
13 Has Table booking
14 Has Online delivery
15 Is delivering now
16 Switch to order menu
17 Price range
18 Aggregate rating
19 Rating color
20 Rating text
21 Votes
```

[ ]:
```
print("Number of Columns in the dataset => {}.".format(number_of_columns))
```

```
Number of Columns in the dataset => 21.
```

[ ]:
```
df.isnull().values.sum()
```

[ ]:
```
9
```

[ ]:
```
for i in range(0, len(columns_df)):
  if df[columns_df[i]].isnull().values.sum() > 0:
    print("No.of null values in {} column is {}".
  ↪format(columns_df[i],df[columns_df[i]].isnull().values.sum()))
  else:
    continue
```

```
No.of null values in Cuisines column is 9
```

[ ]:
```
# Dropped the null valued rows as only 9 rows have null values,
# & removing them won't have impact on the data set
df = df.dropna()
df.isnull().values.sum()
```

[ ]:
```
0
```

[ ]:
```
columns_to_drop = ['Restaurant ID', 'Locality Verbose', 'Locality', 'Switch to␣
  ↪order menu']
df_new = df.drop(columns=columns_to_drop, axis=1)
```

[ ]:
```
df_new.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 9542 entries, 0 to 9550
Data columns (total 17 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   Longitude             9542 non-null   float64
 5   Latitude              9542 non-null   float64
 6   Cuisines              9542 non-null   object
 7   Average Cost for two  9542 non-null   int64
 8   Currency              9542 non-null   object
 9   Has Table booking     9542 non-null   object
 10  Has Online delivery   9542 non-null   object
 11  Is delivering now     9542 non-null   object
 12  Price range           9542 non-null   int64
 13  Aggregate rating      9542 non-null   float64
 14  Rating color          9542 non-null   object
 15  Rating text           9542 non-null   object
 16  Votes                 9542 non-null   int64
dtypes: float64(3), int64(4), object(10)
memory usage: 1.3+ MB
```

[ ]:
```
from sklearn.preprocessing import LabelEncoder

object_columns = ['Has Table booking', 'Has Online delivery', 'Is delivering␣
 ↪now', 'Rating color', 'Rating text']

# Initialize LabelEncoder
label_encoder = LabelEncoder()

# Iterate through object columns and apply label encoding
for column in object_columns:
    df_new[column] = label_encoder.fit_transform(df_new[column])
```

[ ]:
```
label_mapping = {column: {label: index for index, label in␣
 ↪enumerate(label_encoder.classes_)} for column in object_columns}
label_mapping
```

[ ]:
```
{'Has Table booking': {'Average': 0,
  'Excellent': 1,
  'Good': 2,
  'Not rated': 3,
  'Poor': 4,
  'Very Good': 5},
 'Has Online delivery': {'Average': 0,
  'Excellent': 1,
  'Good': 2,
  'Not rated': 3,
  'Poor': 4,
  'Very Good': 5},
 'Is delivering now': {'Average': 0,
  'Excellent': 1,
  'Good': 2,
  'Not rated': 3,
  'Poor': 4,
  'Very Good': 5},
 'Rating color': {'Average': 0,
  'Excellent': 1,
  'Good': 2,
  'Not rated': 3,
  'Poor': 4,
  'Very Good': 5},
 'Rating text': {'Average': 0,
  'Excellent': 1,
  'Good': 2,
  'Not rated': 3,
  'Poor': 4,
  'Very Good': 5}}
```

[ ]:
```
target_stats = df['Aggregate rating'].describe()
print(target_stats)
```

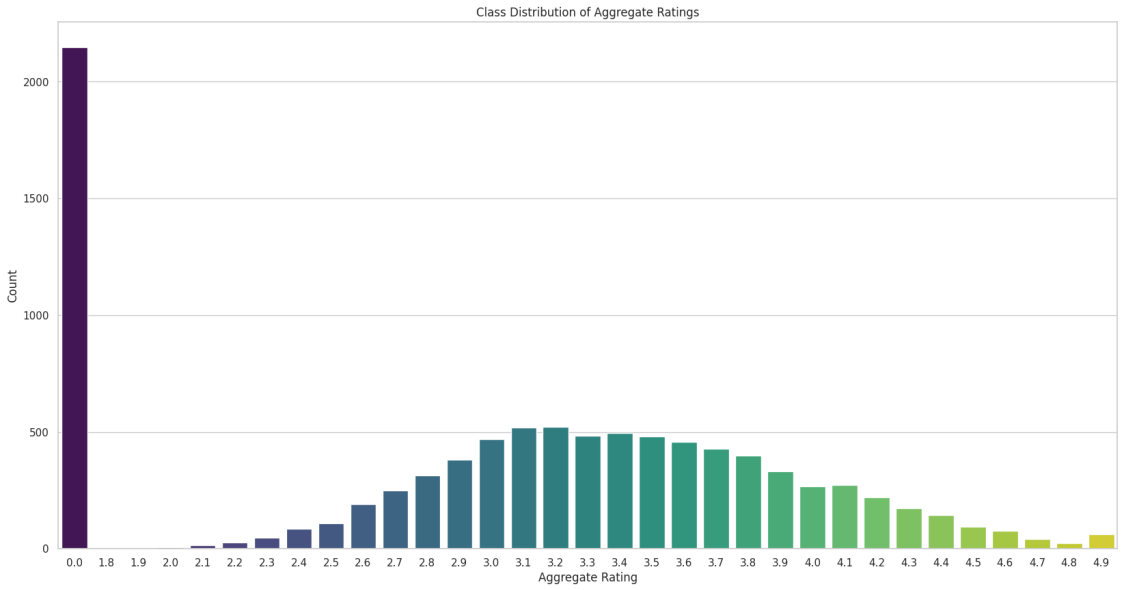
```
count    9542.000000
mean        2.665238
std         1.516588
min         0.000000
25%         2.500000
50%         3.200000
75%         3.700000
max         4.900000
Name: Aggregate rating, dtype: float64
```

[ ]:
```
sns.set(style="whitegrid")

# Plot a countplot to visualize class distribution
plt.figure(figsize=(20, 10))
sns.countplot(data=df, x='Aggregate rating', palette='viridis')

# Add labels and title
plt.xlabel('Aggregate Rating')
plt.ylabel('Count')
plt.title('Class Distribution of Aggregate Ratings')

# Show the plot
plt.show()
```



[ ]:
```
df_new.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 9542 entries, 0 to 9550
Data columns (total 17 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   Longitude             9542 non-null   float64
 5   Latitude              9542 non-null   float64
 6   Cuisines              9542 non-null   object 
 7   Average Cost for two  9542 non-null   int64  
 8   Currency              9542 non-null   object 
 9   Has Table booking     9542 non-null   int64  
 10  Has Online delivery   9542 non-null   int64  
 11  Is delivering now     9542 non-null   int64  
 12  Price range           9542 non-null   int64  
 13  Aggregate rating      9542 non-null   float64
 14  Rating color          9542 non-null   int64  
 15  Rating text           9542 non-null   int64  
 16  Votes                 9542 non-null   int64  
dtypes: float64(3), int64(9), object(5)
memory usage: 1.3+ MB
```

[ ]:
```
df_new.to_csv('L1T2_Dataset.csv', index=False)
```