Import Libraries

In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from scipy.spatial.distance import pdist, squareform
```

Loading the Dataset

In [2]:
```
restaurant_data = pd.read_csv("Restaurant.csv")
```

In [3]:
```
restaurant_data.head()
```

Out[3]:

	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 rows × 21 columns

In [4]:
```
restaurant_data.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]:
```
restaurant_data.isna().sum()
```

Out[5]:
```
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
```

In [6]:
```
df_restaurants = restaurant_data[['Restaurant ID', 'Restaurant Name', 'Cuisines', 'Price
df_restaurants
```

Out[6]:

	Restaurant ID	Restaurant Name	Cuisines	Price range	Aggregate rating	Votes
0	6317637	Le Petit Souffle	French, Japanese, Desserts	3	4.8	314
1	6304287	Izakaya Kikufuji	Japanese	3	4.5	591
2	6300002	Heat - Edsa Shangri-La	Seafood, Asian, Filipino, Indian	4	4.4	270
3	6318506	Ooma	Japanese, Sushi	4	4.9	365
4	6314302	Sambo Kojin	Japanese, Korean	4	4.8	229
...	...	...	...	...	...	...
9546	5915730	Naml\ Gurme	Turkish	3	4.1	788
9547	5908749	Ceviz A��acl	World Cuisine, Patisserie, Cafe	3	4.2	1034
9548	5915807	Huqqa	Italian, World Cuisine	4	3.7	661
9549	5916112	A���k Kahve	Restaurant Cafe	4	4.0	901
9550	5927402	Walter's Coffee Roastery	Cafe	2	4.0	591

9551 rows × 6 columns

In [7]:
```
def describe_columns(df):
    column_info = []
```

```
    for col in df.columns:
        column_info.append(
            [col,
             df[col].dtype,
             df[col].isna().sum(),
             round(df[col].isna().sum() / len(df) * 100, 2),
             df[col].nunique(),
             list(df[col].drop_duplicates().sample(min(2, df[col].nunique()), random_sta
        )
    data_description = pd.DataFrame(
        data=column_info,
        columns=['Column', 'Data_Type', 'Missing_Values', 'Percent_Missing', 'Unique_Cou
    )
    return data_description
```

In [8]:
```
describe_columns(df_restaurants)
```

Out[8]:

	Column	Data_Type	Missing_Values	Percent_Missing	Unique_Count	Sample_Values
0	Restaurant ID	int64	0	0.00	9551	[307888, 2004]
1	Restaurant Name	object	0	0.00	7446	[Corp Kitchen, 10 Downing Street]
2	Cuisines	object	9	0.09	1825	[American, Seafood, Southern, Assamese, Chinese]
3	Price range	int64	0	0.00	4	[1, 2]
4	Aggregate rating	float64	0	0.00	33	[0.0, 3.4]
5	Votes	int64	0	0.00	1012	[1670, 853]

In [9]:
```
df_restaurants = df_restaurants.dropna()
```

In [10]:
```
df_restaurants
```

Out[10]:

	Restaurant ID	Restaurant Name	Cuisines	Price range	Aggregate rating	Votes
0	6317637	Le Petit Souffle	French, Japanese, Desserts	3	4.8	314
1	6304287	Izakaya Kikufuji	Japanese	3	4.5	591
2	6300002	Heat - Edsa Shangri-La	Seafood, Asian, Filipino, Indian	4	4.4	270
3	6318506	Ooma	Japanese, Sushi	4	4.9	365
4	6314302	Sambo Kojin	Japanese, Korean	4	4.8	229
...	...	...	...	...	...	...
9546	5915730	Naml Gurme	Turkish	3	4.1	788
9547	5908749	Ceviz A��acl	World Cuisine, Patisserie, Cafe	3	4.2	1034
9548	5915807	Huqqa	Italian, World Cuisine	4	3.7	661
9549	5916112	A���k Kahve	Restaurant Cafe	4	4.0	901
9550	5927402	Walter's Coffee Roastery	Cafe	2	4.0	591

9542 rows × 6 columns

In [11]:
```
df_restaurants.duplicated().sum()
```

Out[11]: 0

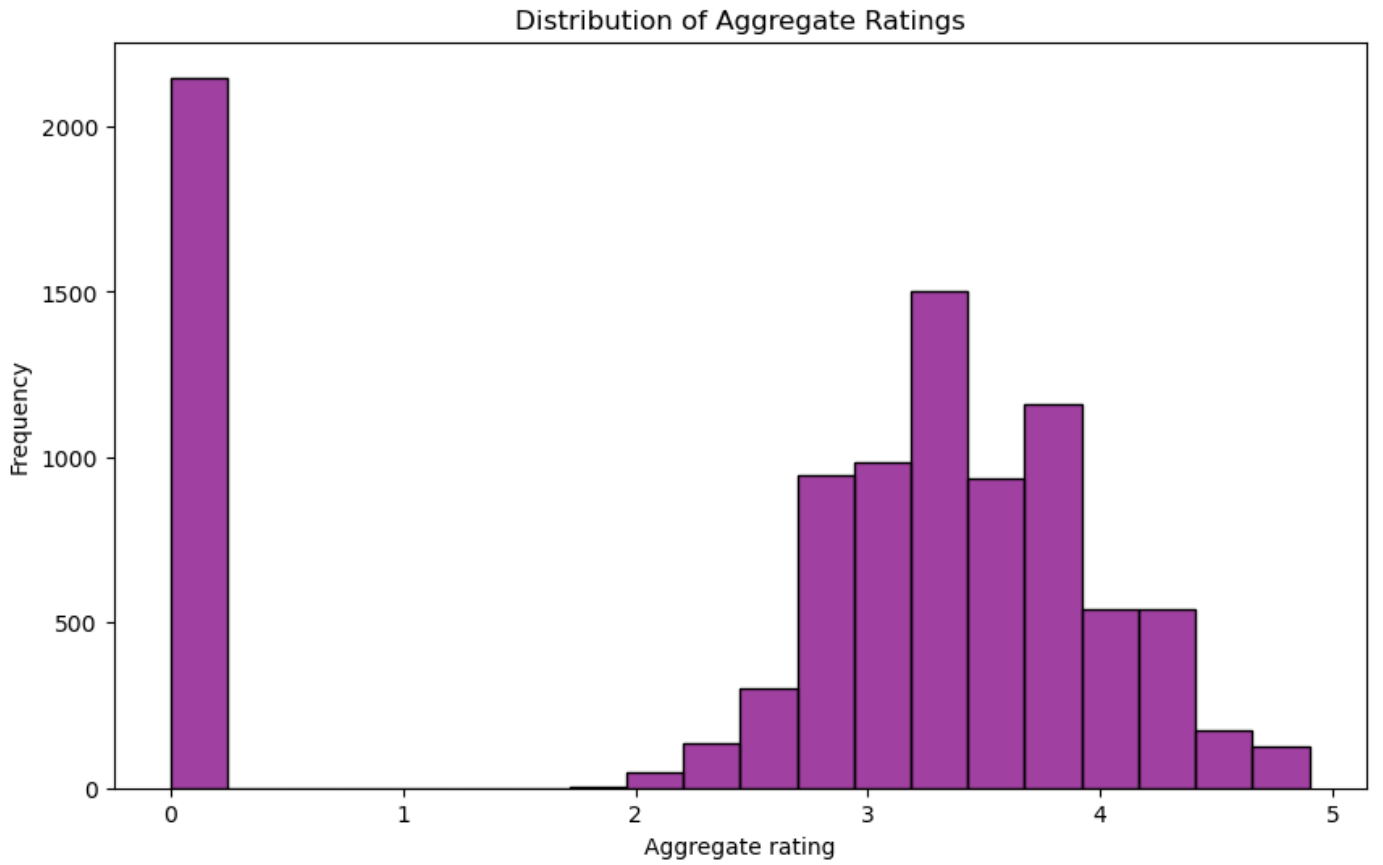
In [12]:
```
df_restaurants['Restaurant Name'].duplicated().sum()
```

Out[12]: 2105

In [13]:
```
df_restaurants['Cuisines'].value_counts()
```

Out[13]:
```
North Indian                                                936
North Indian, Chinese                                       511
Chinese                                                     354
Fast Food                                                   354
North Indian, Mughlai                                       334
                                                           ...
Bengali, Fast Food                                            1
North Indian, Rajasthani, Asian                               1
Chinese, Thai, Malaysian, Indonesian                          1
Bakery, Desserts, North Indian, Bengali, South Indian         1
Italian, World Cuisine                                        1
Name: Cuisines, Length: 1825, dtype: int64
```

In [14]:
```
plt.figure(figsize=(10, 6))
sns.histplot(restaurant_data['Aggregate rating'], bins=20, color='purple')
plt.title('Distribution of Aggregate Ratings')
plt.xlabel('Aggregate rating')
plt.ylabel('Frequency')
plt.show()
```



In [15]:
```
# Generating cross tabulation of restaurant names and cuisines

cross_tab_resto_cuisines = pd.crosstab(df_restaurants['Restaurant Name'],
                                       df_restaurants['Cuisines'])
cross_tab_resto_cuisines
```

Out[15]:

Cuisines	Afghani	Afghani, Mughlai,	Afghani, North	Afghani, North	African	African, Portuguese	American	American, Asian,	American, Asian,	Ame

Restaurant Name	Chinese	Indian	Indian, Pakistani, Arabian				Burger	European, Seafood	I Se
#45	0	0	0	0	0	0	0	0	0
#Dilliwaala6	0	0	0	0	0	0	0	0	0
#InstaFreeze	0	0	0	0	0	0	0	0	0
#OFF Campus	0	0	0	0	0	0	0	0	0
#Urban Caf�?	0	0	0	0	0	0	0	0	0
...	...	...	...	...	...	...	...	...	...
t Lounge by Dilmah	0	0	0	0	0	0	0	0	0
tashas	0	0	0	0	0	0	0	0	0
wagamama	0	0	0	0	0	0	0	0	0
{Niche} - Cafe & Bar	0	0	0	0	0	0	0	0	0
?ukura??a Sofras?	0	0	0	0	0	0	0	0	0

7437 rows × 1825 columns

In [16]:
```
# Checking on restaurant name value
cross_tab_resto_cuisines.loc['feel ALIVE'].values
```

Out[16]:
```
array([0, 0, 0, ..., 0, 0, 0], dtype=int64)
```

In [17]:
```
# Sample of Restaurant Names
df_restaurants['Restaurant Name'].sample(20, random_state=101)
```

Out[17]:
```
989                       Sanjha Chulha
7275                    Cafe Coffee Day
3981                         Coast Cafe
7753                     Slice of Italy
3255                        Rara Dragon
7969                     Meghraj Sweets
855                  Eddie's Patisserie
7259                       Smile Bakers
6561                              Giani
2959                     Me and My Cake
5035      Dhaba Cash 'N' Carry Kitchen
4545                         Desi Villa
8221                    Punjabi Pakwaan
8547                           Lapaalap
7907                Shama Chicken Corner
2536                     Barbeque Nation
6483    The Hubbub Cafe and Restaurant
6942                         Biryani Bot
1678                  Panther Restaurant
2740             SGF - Spice Grill Flame
Name: Restaurant Name, dtype: object
```

In [18]:
```
from sklearn.metrics import jaccard_score
```

```
# Measuring similarity between two restaurants using Jaccard similarity score
print(jaccard_score(
    cross_tab_resto_cuisines.loc["Olive Bistro"].values,
    cross_tab_resto_cuisines.loc["Rose Cafe"].values,
    average='micro'  # or 'macro', 'weighted', depending on your requirement
))
```

```
0.9978106185002736
```

Recommendation

In [19]:

```
import pandas as pd
from sklearn.metrics import jaccard_score

# Sample data: Assuming we have a DataFrame `restaurants_data`
# with binary features representing different characteristics of restaurants
restaurants_data = {
    'Ooma': [1, 0, 1, 0, 1],
    'Desi Villa': [1, 0, 1, 0, 1],
    'Punjabi Pakwaan': [1, 0, 1, 0, 1],
    'Biryani Bot': [1, 0, 1, 0, 1],
    'Me and My Cake': [1, 0, 1, 0, 1]
}
df = pd.DataFrame(restaurants_data).T

# Sample ratings data
ratings_data = {
    'Restaurant Name': ['Ooma', 'Desi Villa', 'Punjabi Pakwaan', 'Biryani Bot', 'Me and
    'Aggregate Rating': [4.5, 4.0, 4.5, 4.2, 4.8]
}
ratings_df = pd.DataFrame(ratings_data).set_index('Restaurant Name')

# Calculate the Jaccard similarity matrix
similarity_matrix = pd.DataFrame(index=df.index, columns=df.index)

for i in df.index:
    for j in df.index:
        similarity_matrix.loc[i, j] = jaccard_score(df.loc[i], df.loc[j])

# Initial restaurant
initial_restaurant = 'Ooma'

# Calculating similarity scores between the initial restaurant and other restaurants
similarity_scores = similarity_matrix.loc[initial_restaurant].sort_values(ascending=Fals

# Creating DataFrame to store similarity scores
similarity_df = pd.DataFrame({'Restaurant Name': similarity_scores.index, 'Similarity Sc

# Merging similarity scores with ratings
result_df = similarity_df.merge(ratings_df, left_on='Restaurant Name', right_index=True)

# Sorting the final DataFrame by similarity scores and resetting index
result_df = result_df.reset_index(drop=True)

print(result_df)
```

```
   Restaurant Name Similarity Score  Aggregate Rating
0            Ooma              1.0               4.5
1      Desi Villa              1.0               4.0
2  Punjabi Pakwaan             1.0               4.5
3      Biryani Bot             1.0               4.2
4   Me and My Cake             1.0               4.8
```

The Data above shows the top 5 recommended restaurants with the best ratings. The ratings are curated to include only those that are 4 or above, ensuring the recommendation system that provides good ratings.

Thank You