Task: Data Visualization

Import libraries

In [1]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

Load Dataset

In [2]:
```
df=pd.read_csv('D:\Intern\Cognifyz Intern\Dataset .csv')
```

Data characteristics

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

Out[3]:

	Restaurant ID	Restaurant Name	Country Code	City	Address	Locality	Locality Verbose	Longitud
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...	121.02753
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...	121.01410
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...	121.05683

3 rows × 21 columns

In [4]:
```
df.describe()
```

Out[4]:

	Restaurant ID	Country Code	Longitude	Latitude	Average Cost for two	Price range	Aggr
count	9.551000e+03	9551.000000	9551.000000	9551.000000	9551.000000	9551.000000	9551.0
mean	9.051128e+06	18.365616	64.126574	25.854381	1199.210763	1.804837	2.6
std	8.791521e+06	56.750546	41.467058	11.007935	16121.183073	0.905609	1.5
min	5.300000e+01	1.000000	-157.948486	-41.330428	0.000000	1.000000	0.0
25%	3.019625e+05	1.000000	77.081343	28.478713	250.000000	1.000000	2.5
50%	6.004089e+06	1.000000	77.191964	28.570469	400.000000	2.000000	3.2
75%	1.835229e+07	1.000000	77.282006	28.642758	700.000000	2.000000	3.7
max	1.850065e+07	216.000000	174.832089	55.976980	800000.000000	4.000000	4.9

In [5]:
```
df.info()
```

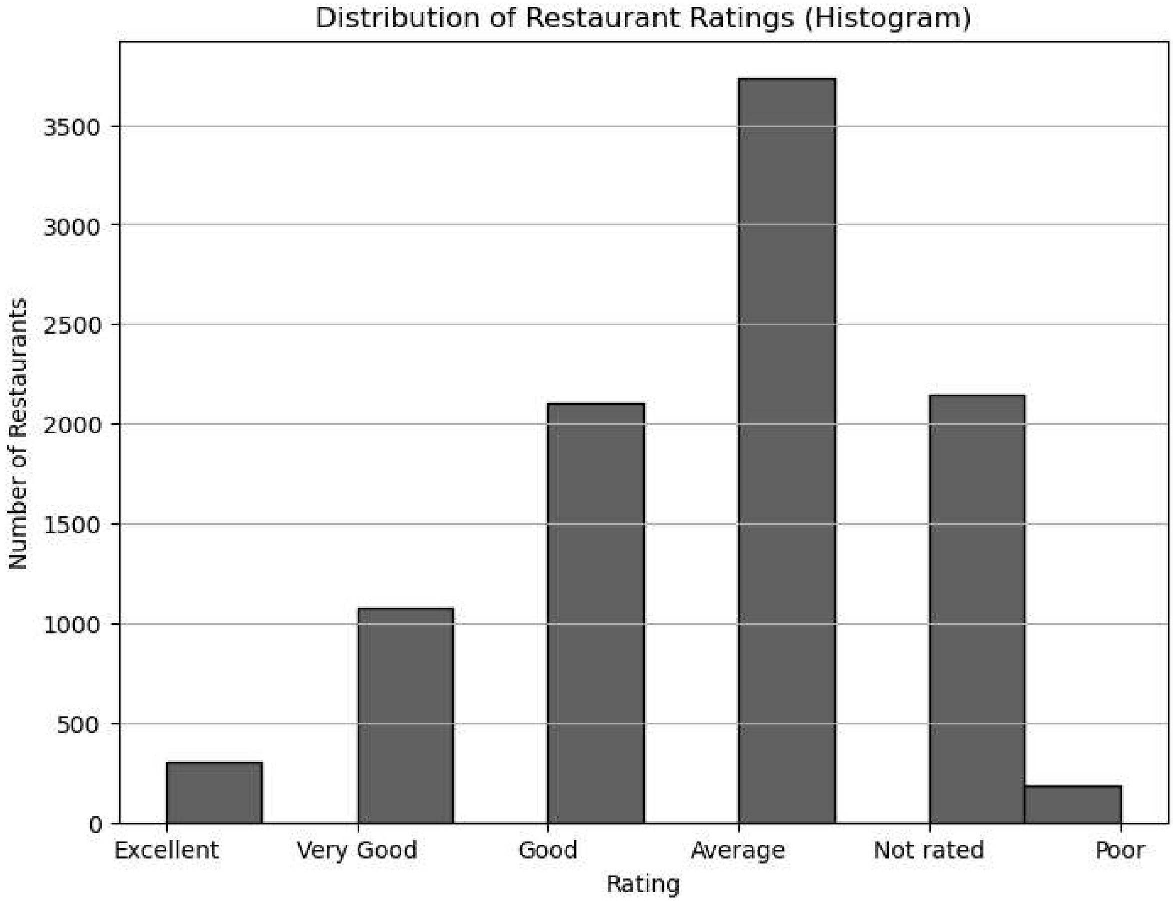
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9551 entries, 0 to 9550
Data columns (total 21 columns):
 #   Column                Non-Null Count  Dtype  
---  ------                --------------  -----  
 0   Restaurant ID         9551 non-null   int64  
 1   Restaurant Name       9551 non-null   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
```

```
In [6]: df.isnull().sum()
```

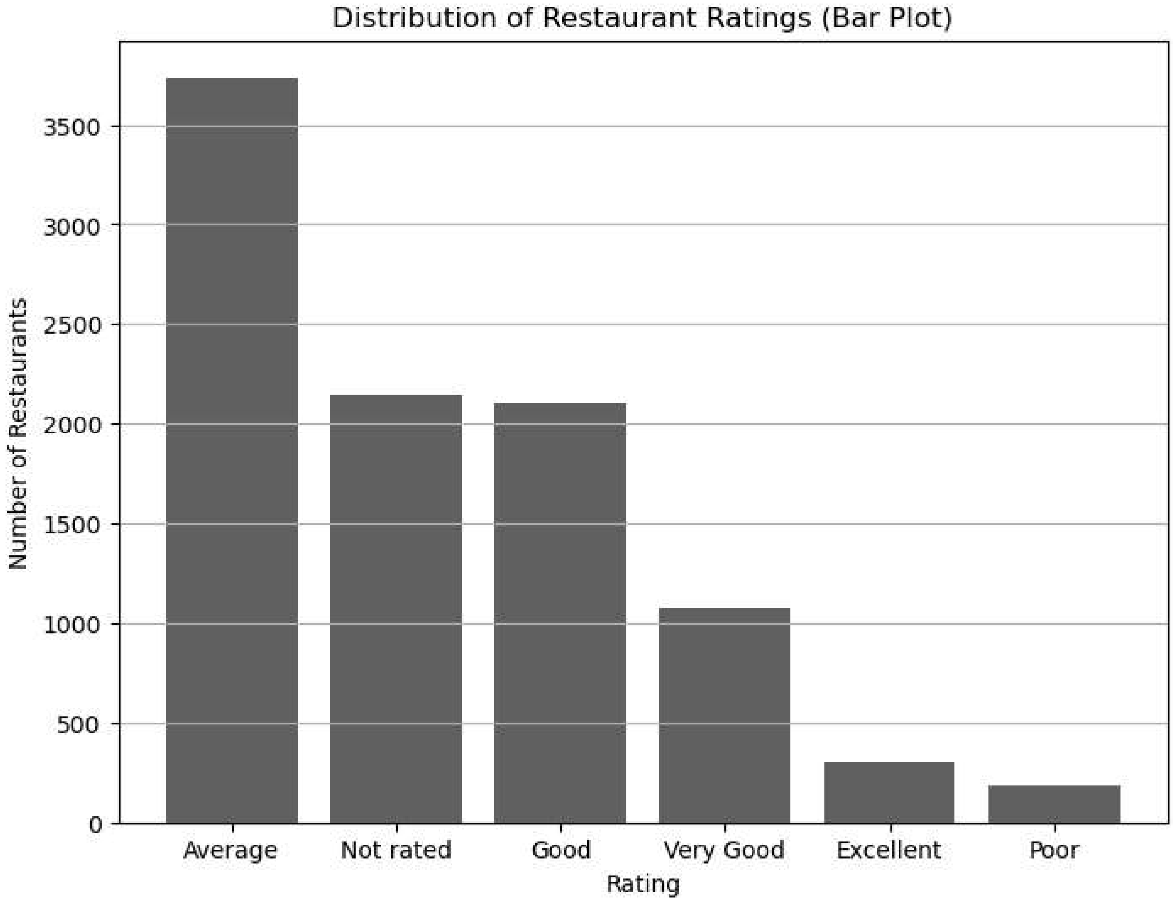
```
Country Code            0
City                    0
Address                 0
Locality                0
Locality Verbose        0
Longitude               0
Latitude                0
Cuisines                9
Average Cost for two    0
Currency                0
Has Table booking       0
Has Online 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
```

Create visualizations to represent the distribution of ratings using different charts (histogram, bar plot)

In [9]:
```
plt.figure(figsize=(8, 6))
plt.hist(df['Rating text'], bins=10, edgecolor='black')
plt.xlabel('Rating')
plt.ylabel('Number of Restaurants')
plt.title('Distribution of Restaurant Ratings (Histogram)')
plt.grid(axis='y')
plt.show()
```

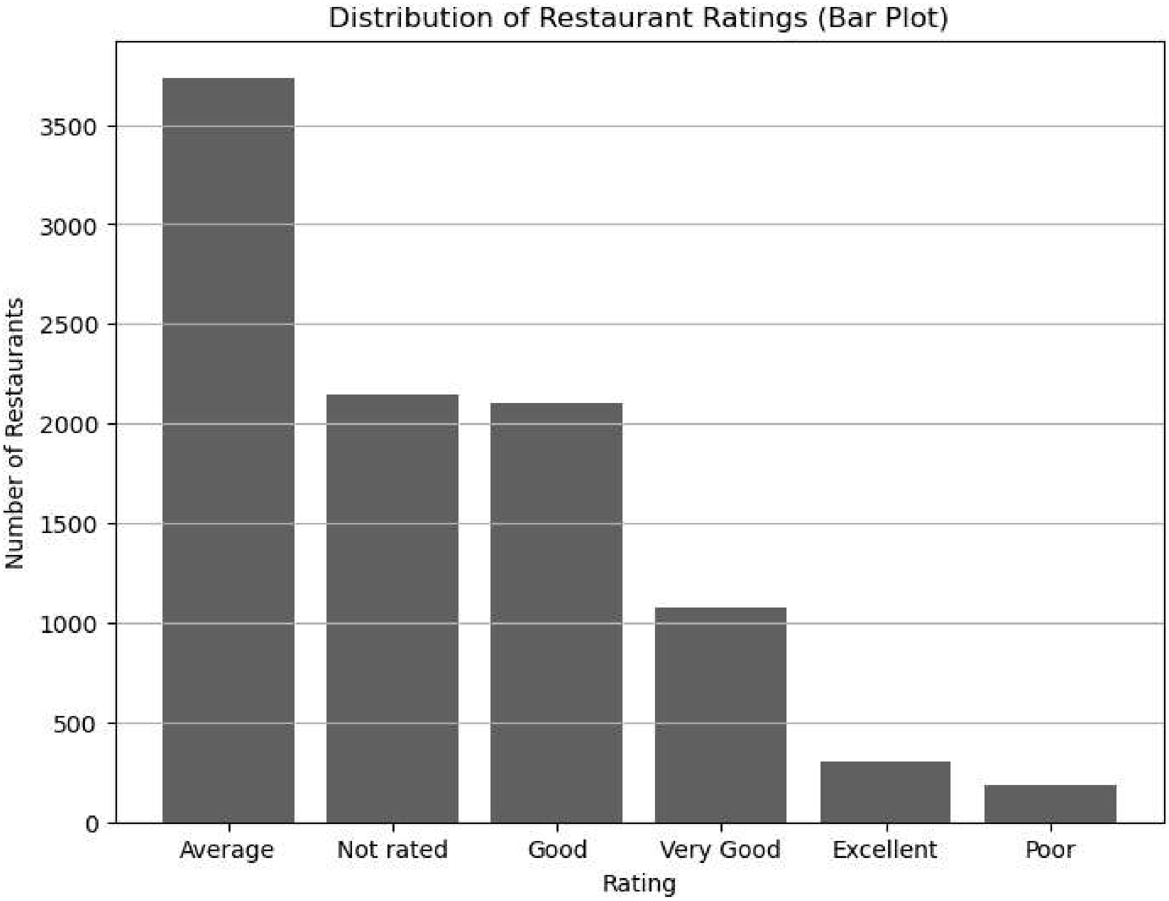


```
In [10]: plt.figure(figsize=(8, 6))
         plt.bar(df['Rating text'].value_counts().index, df['Rating text'].value_counts
         plt.xlabel('Rating')
         plt.ylabel('Number of Restaurants')
         plt.title('Distribution of Restaurant Ratings (Bar Plot)')
         plt.grid(axis='y')
         plt.show()
```

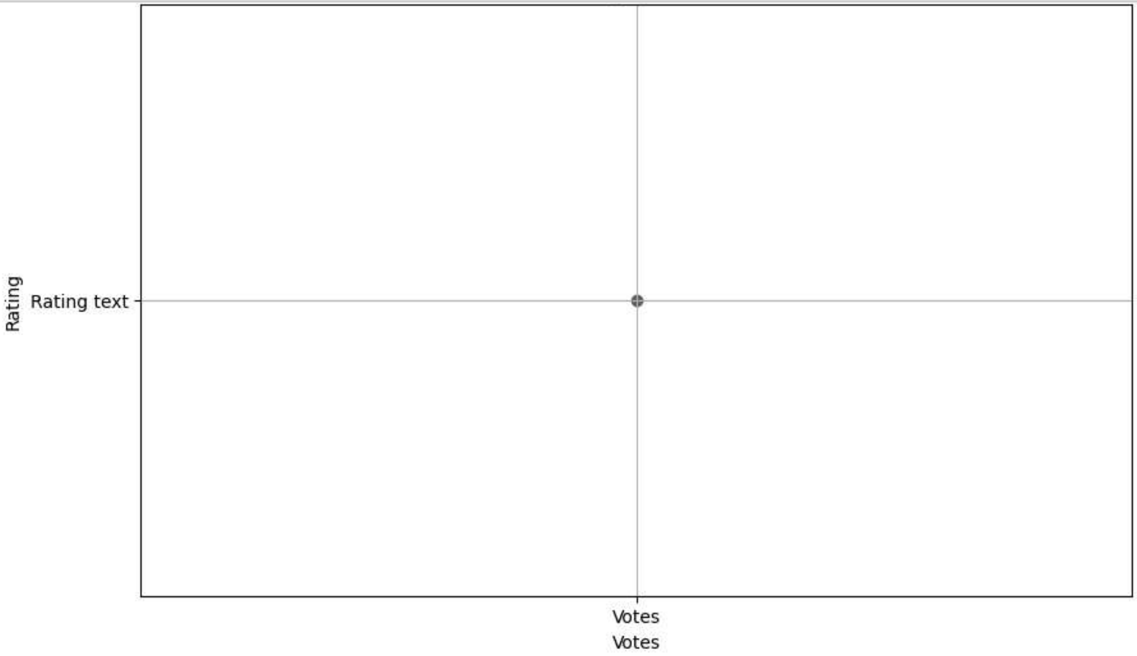


In [11]:

```
plt.figure(figsize=(8, 6))
plt.bar(df['Rating text'].value_counts().index, df['Rating text'].value_counts
plt.xlabel('Rating')
plt.ylabel('Number of Restaurants')
plt.title('Distribution of Restaurant Ratings (Bar Plot)')
plt.grid(axis='y')
plt.show()
```



In [19]:
```
plt.figure(figsize=(10, 6))
plt.scatter(x=['Votes'],y=['Rating text'],data=df)
plt.xlabel('Votes')
plt.ylabel('Rating')
plt.title('Rating vs Votes')
plt.grid()
plt.show()
```



Compare the average ratings of different cuisines or cities using appropriate visualizations

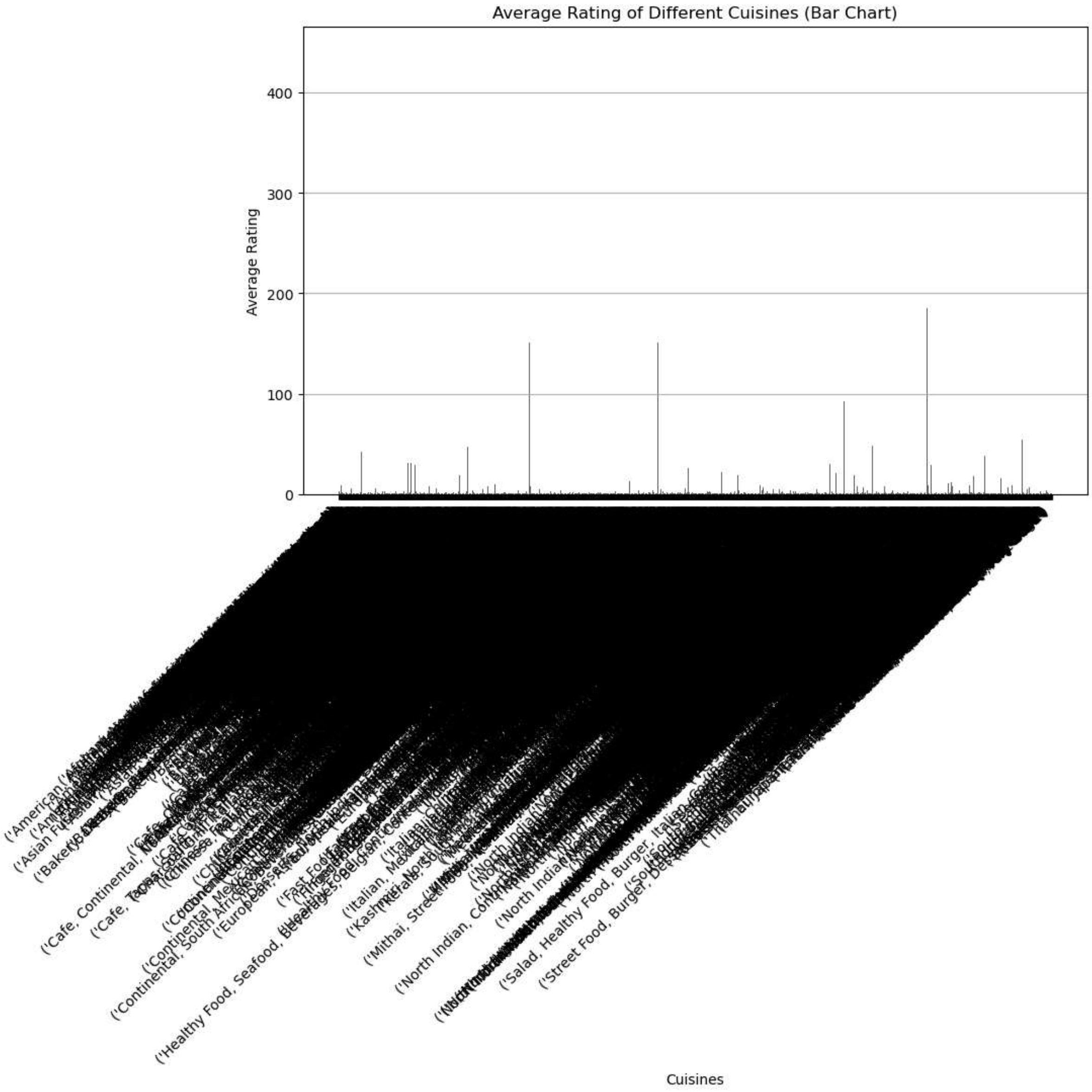
In [23]:
```
avg_rating_cuisine = df.groupby('Cuisines')['Rating text'].value_counts()
print(avg_rating_cuisine)
```

```
Cuisines                                    Rating text
Afghani                                     Not rated      3
                                            Average        1
Afghani, Mughlai, Chinese                   Not rated      1
Afghani, North Indian                       Not rated      1
Afghani, North Indian, Pakistani, Arabian   Not rated      1
                                                          ..
Western, Asian, Cafe                        Very Good      1
Western, Fusion, Fast Food                  Average        1
World Cuisine                               Excellent      1
World Cuisine, Mexican, Italian             Very Good      1
World Cuisine, Patisserie, Cafe             Very Good      1
Name: count, Length: 2616, dtype: int64
```

In [25]:

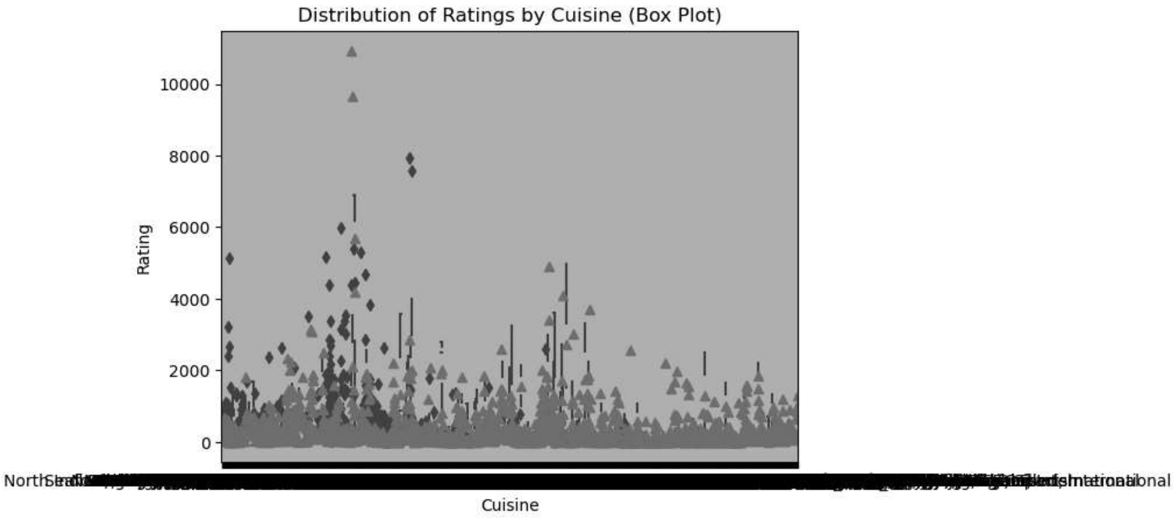
```
cuisine_labels = avg_rating_cuisine.index.to_series().astype(str)

plt.figure(figsize=(10, 6))
plt.bar(cuisine_labels, avg_rating_cuisine.values)
plt.xlabel('Cuisines')
plt.ylabel('Average Rating')
plt.title('Average Rating of Different Cuisines (Bar Chart)')
plt.xticks(rotation=45, ha='right')
plt.grid(axis='y')
plt.show()
```



In [27]:

```
sns.boxplot(
    x =df['Cuisines'],
    y = df['Votes'],
    showmeans=True,
)
plt.xlabel('Cuisine')
plt.ylabel('Rating')
plt.title('Distribution of Ratings by Cuisine (Box Plot)')
plt.grid()
plt.show()
```

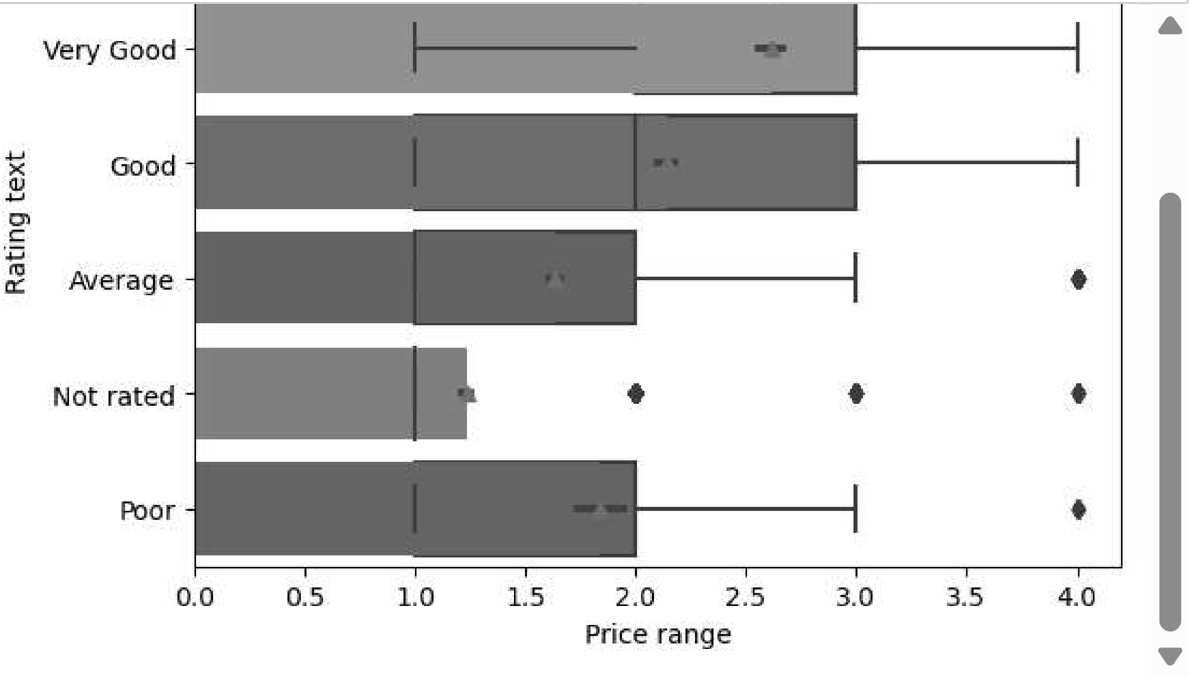


Visualize the relationship between various features and the target variable to gain insights

In [31]:

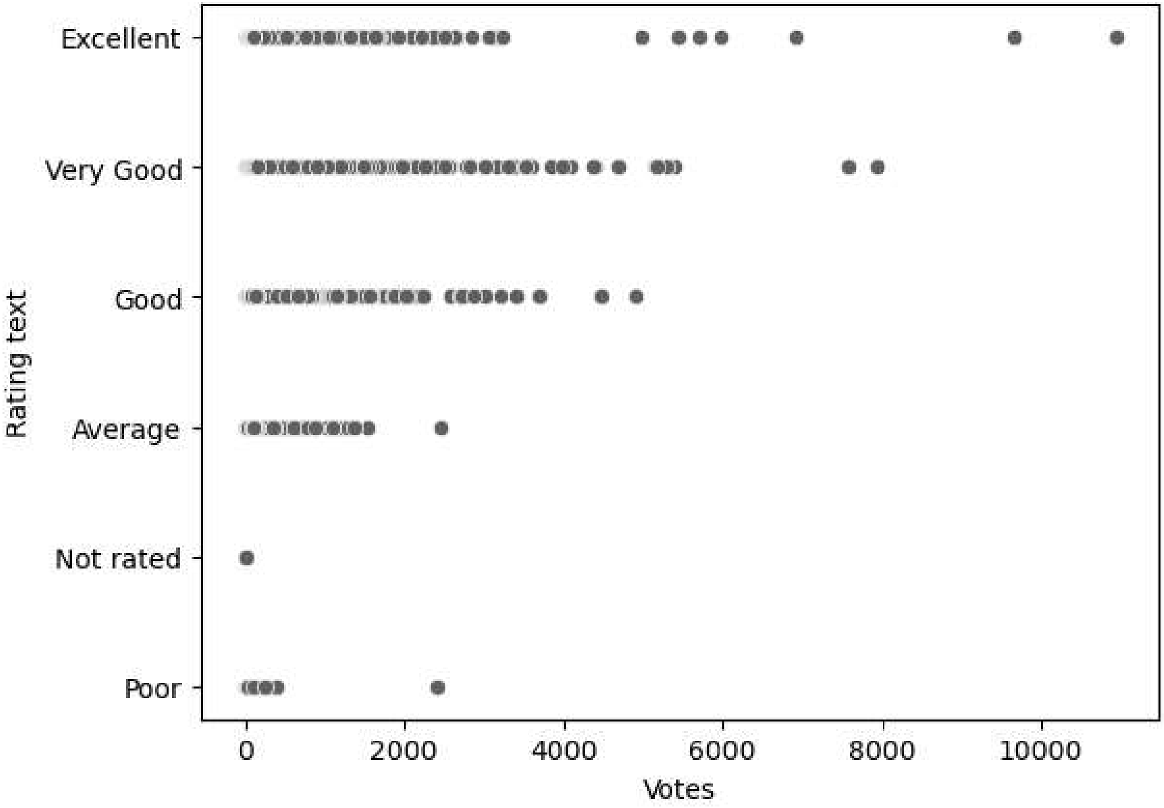
```
target_var = 'Rating text'
```

In [32]:
```
categorical_feature = 'Price range'
sns.barplot(x=categorical_feature, y=target_var, data=df)
sns.boxplot(x=categorical_feature, y=target_var, showmeans=True, data=df)
```



In [35]:
```
sns.scatterplot(x='Votes', y=target_var, data=df)
```

Out[35]:
```
<Axes: xlabel='Votes', ylabel='Rating text'>
```



In [ ]: