PROJECT – ZOMATO-II

Question-2

Submitted by :-

Aryan Sharma

Q2.) User Rating of a restaurant plays a crucial role in selecting a restaurant or ordering the food from the restaurant.

2.1) Write a short detail analysis of how the rating is affected by restaurant due following features: Plot a suitable graph to explain your inference.

2.1.1) Number of Votes given Restaurant

Source Code

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt


data = pd.read_csv('zomato.csv' , encoding = 'latin-1')


data = data[data["Country Code"] == 1]


data = data[data["Rating text"] != "Not rated"]


ratings = np.array(data["Aggregate rating"])
votes = np.array(data["Votes"])
```

```
votes_to_rating_count = {}

for i in range(len(ratings)) :
    if ratings[i] not in votes_to_rating_count :
        votes_to_rating_count[ratings[i]] = [votes[i] , 1]
    else:
        votes_to_rating_count[ratings[i]][0] += votes[i]
        votes_to_rating_count[ratings[i]][1] += 1

votes_to_rating = []
for key,value in votes_to_rating_count.items() :
    avg_votes = value[0]//value[1]
    votes_to_rating.append([key , avg_votes])

votes_to_rating.sort()
```

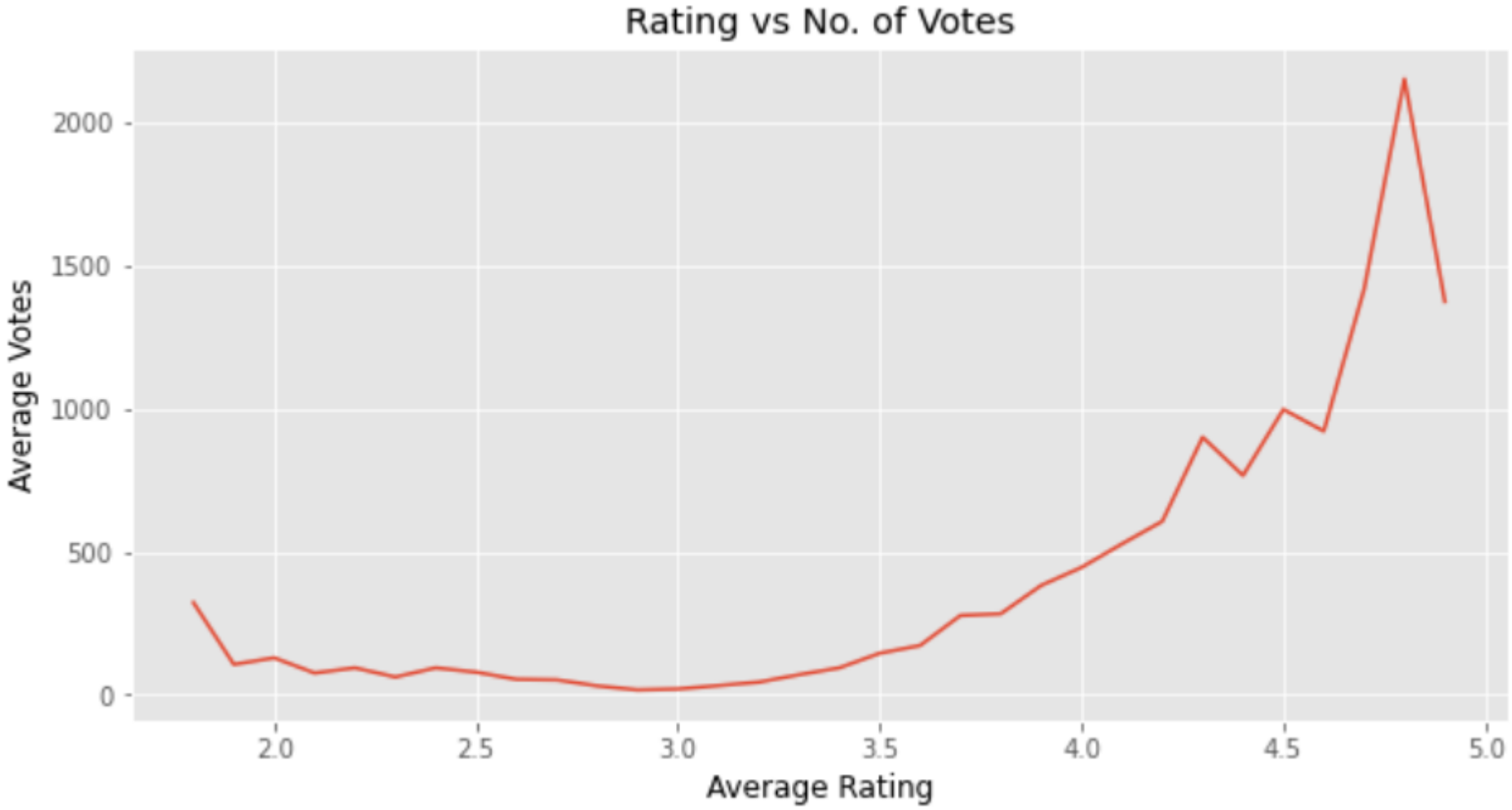
```
rating = [i[0] for i in votes_to_rating ]
votes = [i[1] for i in votes_to_rating ]

plt.style.use('ggplot')
plt.figure(figsize=(10,5))
plt.plot(rating , votes)
plt.xlabel('Average Rating' , color = 'k')
plt.ylabel('Average Votes' , color = 'k')
plt.title('Rating vs No. of Votes')
plt.show()
```

Explanation

- The necessary libraries were included.
- The dataset was loaded and the appropriate encoding was used which happens to be latin-1 in this case .
- The question specfies to analyse the data for India so the data has been filtered by using the Country Code which is 1 for India.
- **IMPORTANT** – Since in these questions we are analysing the user rating it is important to get rid of entries for which there is no rating. This has been done by filtering using the “Rating Text” and removing all the data with rating text as “Not rated”.
- Two numpy arrays were made named ratings and votes to get the Aggregate rating and votes from the data.
- Then a dictionary was made which will keep the rating as the key and the number of votes and the count of restaurants as values.

- The we loop through the length of the ratings array to cover all ratings and form the dictionary. For each rating in the ratings array we find the total number of votes and the total restaurants which contribute these votes.
- We then make a list of lists named votes_to_rating which will store the average votes and ratings .
- We loop through each of the dictionary entries. We find the average votes by dividing the total votes by the no of restaurants, both of which were the 0^{th} and 1^{st} indexed values of the dictionary. Then the rating which was the key of the dictionary and the average votes for that rating were stored in the votes_to_rating list.
- The the list votes_to_rating is sorted based on the rating.
- Two separate lists were made for rating and votes using the votes_to_rating list which would help in plotting the graph.
- Finally a line graph was made between rating and average votes.

Results


Rating vs No. of Votes
Average Votes
2000
1500
1000
500
0
2.0
2.5
3.0
3.5
4.0
4.5
5.0
Average Rating

From the graph it can be concluded that restaurants with higher number of votes have a higher rating.

It can be seen that restaurants with more than 500 votes lie in the rating of 4-5 with restaurants having more the 1500 votes lying in the rating between 4.5 and 5.

So it can be concluded that once must prefer going to a restaurant with higher number of votes to have an overall good experience.

2.1.2.) Restaurant serving more number of cuisines

Source Code

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

data = pd.read_csv('zomato.csv' , encoding = 'latin-1')
data = data[data["Country Code"] == 1]

data = data[data["Rating text"] != "Not rated"]

ratings = np.array(data["Aggregate rating"])
cuisines = np.array(data["Cuisines"])

cuisine_with_rating = {}
```

```
for i in range(len(ratings)) :
    no_of_cuisines = len(cuisines[i].split(", "))

    if no_of_cuisines not in cuisine_with_rating :
        cuisine_with_rating[no_of_cuisines] = [ratings[i] , 1]
    else :
        cuisine_with_rating[no_of_cuisines][0]+=ratings[i]
        cuisine_with_rating[no_of_cuisines][1]+=1
cuisine_with_avg_rating = []

for key,value in cuisine_with_rating.items() :
    total_rating = value[0]
    no_of_res = value[1]
    avg = round(total_rating/no_of_res,2)

    cuisine_with_avg_rating.append([key , avg])

cuisine_with_avg_rating.sort()
```

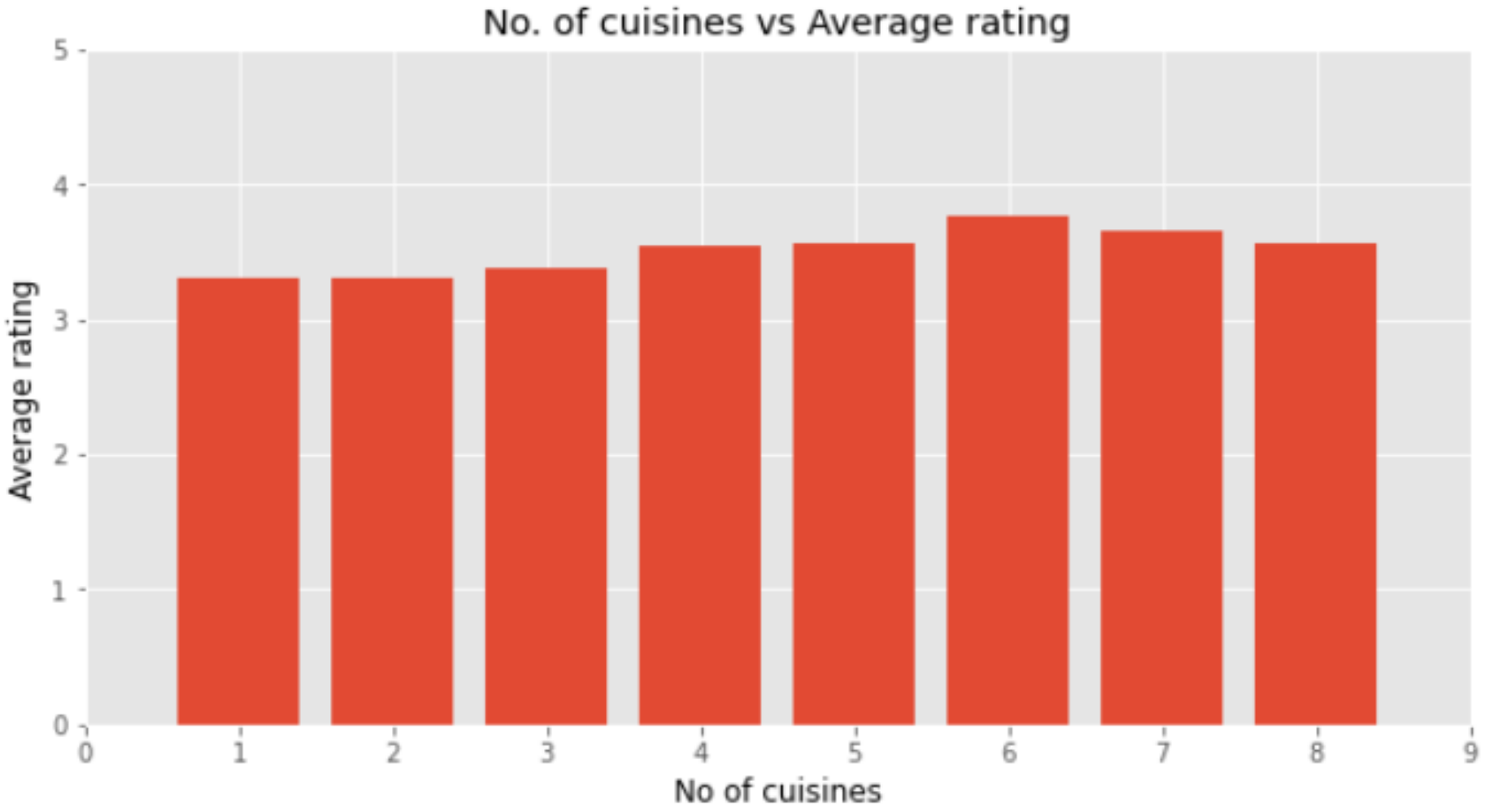
```
cuisine_count = [i[0] for i in cuisine_with_avg_rating]
avg_rating = [i[1] for i in cuisine_with_avg_rating]

plt.style.use('ggplot')
plt.figure(figsize=(10,5))
plt.axis([0,9,0,5])

plt.bar(cuisine_count,avg_rating)
plt.xlabel('No of cuisines' , color = 'k')
plt.ylabel('Average rating' , color = 'k')
plt.title('No. of cuisines vs Average rating’)

plt.show()
```

Explanation

- The necessary libraries were included.
- The dataset was loaded and the appropriate encoding was used which happens to be latin-1 in this case .
- The question specifies to analyze the data for India so the data has been filtered by using the Country Code which is 1 for India.
- **IMPORTANT** – Since in these questions we are analyzing the user rating it is important to get rid of entries for which there is no rating. This has been done by filtering using the “Rating Text” and removing all the data with rating text as “Not rated”.
- Two numpy arrays were made named ratings and cuisines to get the Aggregate rating and cuisines from the data
- A dictionary named cuisine_with_rating which will keep the number of cuisines as key and the rating and number of restaurants as the values.

- We then loop over the length of the ratings array to go over all entries. We find no_of_cuisines by splitting each entry of the cuisines array on “, “ . Then we form the dictionary by keeping the number of cuisines offered as the key and for the values we keep the total rating sum and the number of restaurants as values to find average rating later.
- We then make a list of lists named cuisine_with_avg_rating in which we will store the number of cuisines and the average rating. We find the average rating by diving the total ratings which is the 0^{th} indexed value of the dictionary by the total number of restaurants which is 1^{st} indexed value of the dictionary. The list cuisine_with_avg_rating was thus formed.
- We then sort the list cuisine_with_avg_rating on the basis if number of cuisines offered.
- Then we separate out the number of cuisines and the average rating into two lists to be later used to make the graph.
- A graph was then plotted between the number of cuisines offered and the average rating.

Results


No. of cuisines vs Average rating
Average rating
No of cuisines
0
1
2
3
4
5
6
7
8
9

From the graph we can see the average rating increases with the number of cuisines being offered at a restaurant.

Restaurants offering 6 cuisines had the highest rating and after that on increasing the number of cuisines the rating almost remains the same.

So it can be concluded that higher the number of cuisines offered at restaurant more are the average ratings of those restaurants.

This is also true as a there are higher chances of a customer being satisfied when they have more options to choose from.

2.1.3.) Average Cost of Restaurant

Source Code

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt


data = pd.read_csv('zomato.csv' , encoding = 'latin-1')


data = data[data["Country Code"] == 1]
data = data[data["Rating text"] != "Not rated"]


ratings = np.array(data["Aggregate rating"])
cost_for_two = np.array(data["Average Cost for two"])


rating_and_cost = {}
```

```
for i in range(len(ratings)) :
    if ratings[i] not in rating_and_cost :
        rating_and_cost[cost_for_two[i]] = [ratings[i] , 1]
    else :
        rating_and_cost[cost_for_two[i]][0] += ratings[i]
        rating_and_cost[cost_for_two[i]][1] += 1

cost_with_rating = []

for key , value in rating_and_cost.items() :

    avg = round(value[0]/value[1] , 1)
    cost_with_rating.append([key , avg])

cost_with_rating.sort()
```

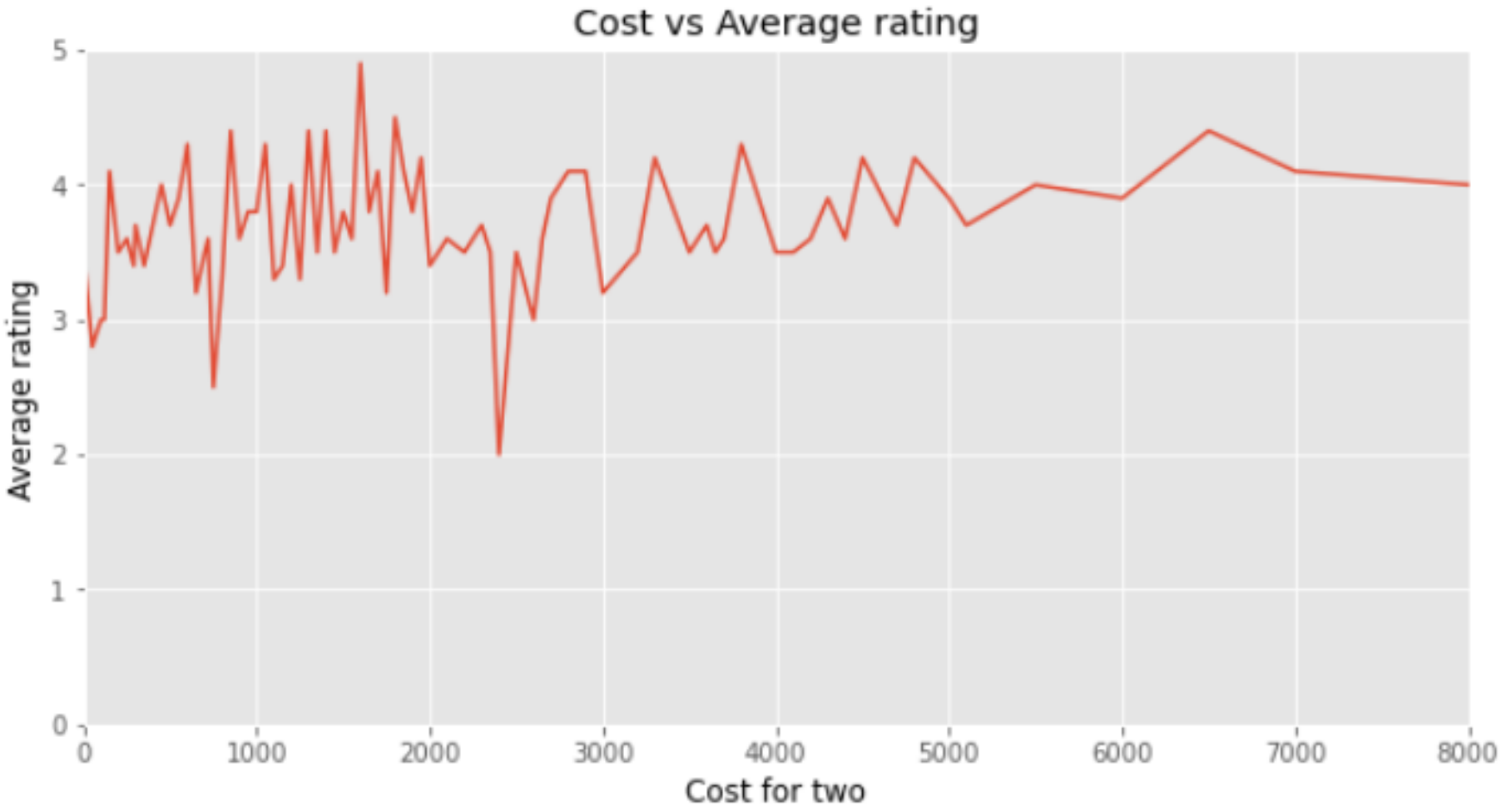
```
cost_for_two = [i[0] for i in cost_with_rating]
avg_rating = [i[1] for i in cost_with_rating]


plt.style.use("ggplot")
plt.figure(figsize=(10,5))
plt.axis([0,8000,0,5])
plt.plot(cost_for_two , avg_rating)
plt.xlabel("Cost for two" , color = 'k')
plt.ylabel("Average rating" , color = 'k')
plt.title("Cost vs Average rating")
plt.show()
```

Explanation

- The necessary libraries were included.
- The dataset was loaded and the appropriate encoding was used which happens to be latin-1 in this case .
- The question specifies to analyze the data for India so the data has been filtered by using the Country Code which is 1 for India.
- **IMPORTANT** – Since in these questions we are analysing the user rating it is important to get rid of entries for which there is no rating. This has been done by filtering using the “Rating Text” and removing all the data with rating text as “Not rated”.
- Two numpy arrays were made named ratings and cost_for_two to get the Aggregate rating and Average Cost for two from the data
- A dictionary named rating_and_cost which will keep the average cost for two as key and the rating and number of restaurants as the values.

- We then loop over the length of the ratings array to go over all entries. Then we form the dictionary by keeping the average cost for tw as the key and for the values we keep the total rating sum and the number of restaurants as values to find average rating later.
- We then make a list of lists named cost_with_rating in which we will store the average cost for two and the average rating. We find the average rating by dividing the total ratings which is the 0^{th} indexed value of the dictionary by the total number of restaurants which is 1^{st} indexed value of the dictionary. The list cost_with_rating was thus formed.
- We then sort the list cost_with_rating on the basis of cost for two.
- Then we separate out the cost for two and the average rating into two lists to be later used to make the graph.
- A graph was then plotted between the cost for two and the average rating.

Results


Cost vs Average rating
Average rating
Cost for two
0
1
2
3
4
5
0
1000
2000
3000
4000
5000
6000
7000
8000

From the graph we can see that the rating rises as the average cost for two increases upto Rs 2000 and then falls a little and rises and stays almost the same there onwards.

From the graph we can see restaurants having average cost for two between Rs 1000 and Rs2000 have the highest rating and have the best cost to value offering among all the restaurants.

2.1.4.) Restaurant serving some specific cuisines

Source Code

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

data = pd.read_csv('zomato.csv' , encoding = 'latin-1')
data = data[data["Country Code"] == 1]
data = data[data["Rating text"] != "Not rated"]

ratings = np.array(data["Aggregate rating"])
cuisines = np.array(data["Cuisines"])
cuisines_to_rating_count = {}
```

```
for i in range(len(ratings)) :
    cuisine_list = cuisines[i].split(", ")
    for c in cuisine_list :
        if c not in cuisines_to_rating_count :
            cuisines_to_rating_count[c] = [ratings[i] , 1]
        else :
            cuisines_to_rating_count[c][0] += ratings[i]
            cuisines_to_rating_count[c][1] += 1
cuisine_to_rating = []
for key,value in cuisines_to_rating_count.items() :

    avg = round(value[0]/value[1] , 1)
    cuisine_to_rating.append([avg , key])
cuisine_to_rating.sort(reverse = True)
```

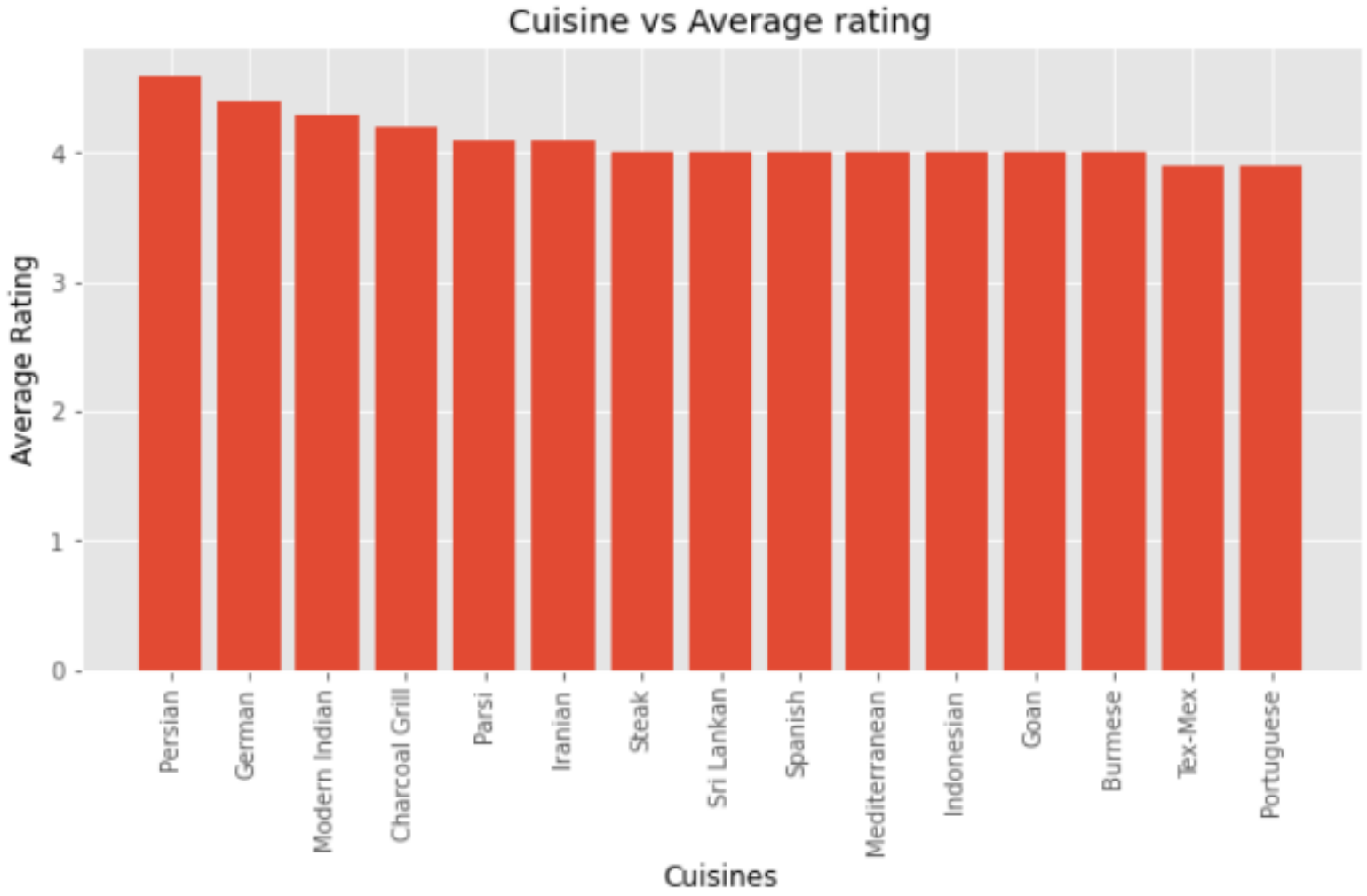
```
cuisine = []
rating = []

for i in cuisine_to_rating :
    cuisine.append(i[1])
    rating.append(i[0])

plt.style.use('ggplot')
plt.figure(figsize=(10,5))
plt.bar(cuisine[:15] , rating[:15] )
plt.xticks(rotation = 90)
plt.xlabel("Cuisines" , color = 'k')
plt.ylabel("Average Rating" , color = 'k')
plt.title('Cuisine vs Average rating')
plt.show()
```

Explanation

- The necessary libraries were included.
- The dataset was loaded and the appropriate encoding was used which happens to be latin-1 in this case .
- The question specifies to analyze the data for India so the data has been filtered by using the Country Code which is 1 for India.
- **IMPORTANT** – Since in these questions we are analyzing the user rating it is important to get rid of entries for which there is no rating. This has been done by filtering using the “Rating Text” and removing all the data with rating text as “Not rated”.
- Two numpy arrays were made named ratings and cuisines to get the Aggregate rating and cuisines from the data
- A dictionary named cuisines_to_rating_count which will keep the cuisine as key and the rating and number of restaurants as the values.

- We then loop over the length of the ratings array to go over all entries. We find the cuisine list by splitting each entry of the cuisines array on “, “ . Then we form the dictionary by keeping the cuisine offered as the key and for the values we keep the total rating sum and the number of restaurants as values to find average rating later.
- We then make a list of lists named cuisine_to_rating in which we will store the average rating and the cuisine. We find the average rating by diving the total ratings which is the 0^{th} indexed value of the dictionary by the total number of restaurants which is 1^{st} indexed value of the dictionary. The list cuisine_to_rating was thus formed.
- We then sort the list cuisine_to_rating on the basis of average rating.
- Then we separate out the cuisines and the average rating into two lists to be later used to make the graph.
- A graph was then plotted between the number of cuisines offered and the average rating.

Results


Cuisine vs Average rating
Average Rating
0
1
2
3
4
Persian
German
Modern Indian
Charcoal Grill
Parsi
Iranian
Steak
Sri Lankan
Spanish
Mediterranean
Indonesian
Goan
Burmese
Tex-Mex
Portuguese
Cuisines

From the graph it can be seen that the Persian cuisine has the highest rating followed by German and Modern Indian.

The top 15 cuisines based in their average rating have been shown.

And we can see how rating is affected by resatuarant serving some specific cuisine.

2.2.1.) Find the weighted restaurant rating of each locality and find out the top 10 localities with more weighted restaurant rating?

Weighted Restaurant Rating=Σ (number of votes * rating) / Σ (number of votes) .

Source Code

```
import numpy as np

import pandas as pd

import matplotlib.pyplot as plt


data = pd.read_csv('zomato.csv' , encoding = 'latin-1')

data = data[data["Country Code"] == 1]

data = data[data["Rating text"] != "Not rated"]


locality_dict = {}


localities = np.array(data['Locality'])

ratings = np.array(data["Aggregate rating"])

votes = np.array(data["Votes"])
```

```
for i in range(len(localities)) :
    if localities[i] not in locality_dict :
        locality_dict[localities[i]] = [votes[i]*ratings[i] , votes[i]]
    else :
        locality_dict[localities[i]][0] += (votes[i]*ratings[i])
        locality_dict[localities[i]][1] += votes[i]

locality_weighted_rating = []
for key , value in locality_dict.items() :
    weighted_rating = round(value[0]/value[1] , 2)

    locality_weighted_rating.append([weighted_rating , key])

locality_weighted_rating.sort(reverse = True)
```

```
localities = [i[1] for i in locality_weighted_rating][:10]
rating = [i[0] for i in locality_weighted_rating][:10]

df = pd.DataFrame(list(zip(localities , rating )), columns = ['Locality', 'Weighted Restaurant Rating'])
display(df)

plt.style.use('ggplot')
plt.figure(figsize=(15,7))
plt.bar(localities , rating)
plt.xticks(rotation = 90)
plt.xlabel('Localities' , color = 'k' , size = 20)
plt.ylabel('Weighted Restaurant Rating' , color = 'k', size = 20)
plt.title('Locality vs Weighted Restaurant Rating' , size = 30)
plt.show()
```

Explanation

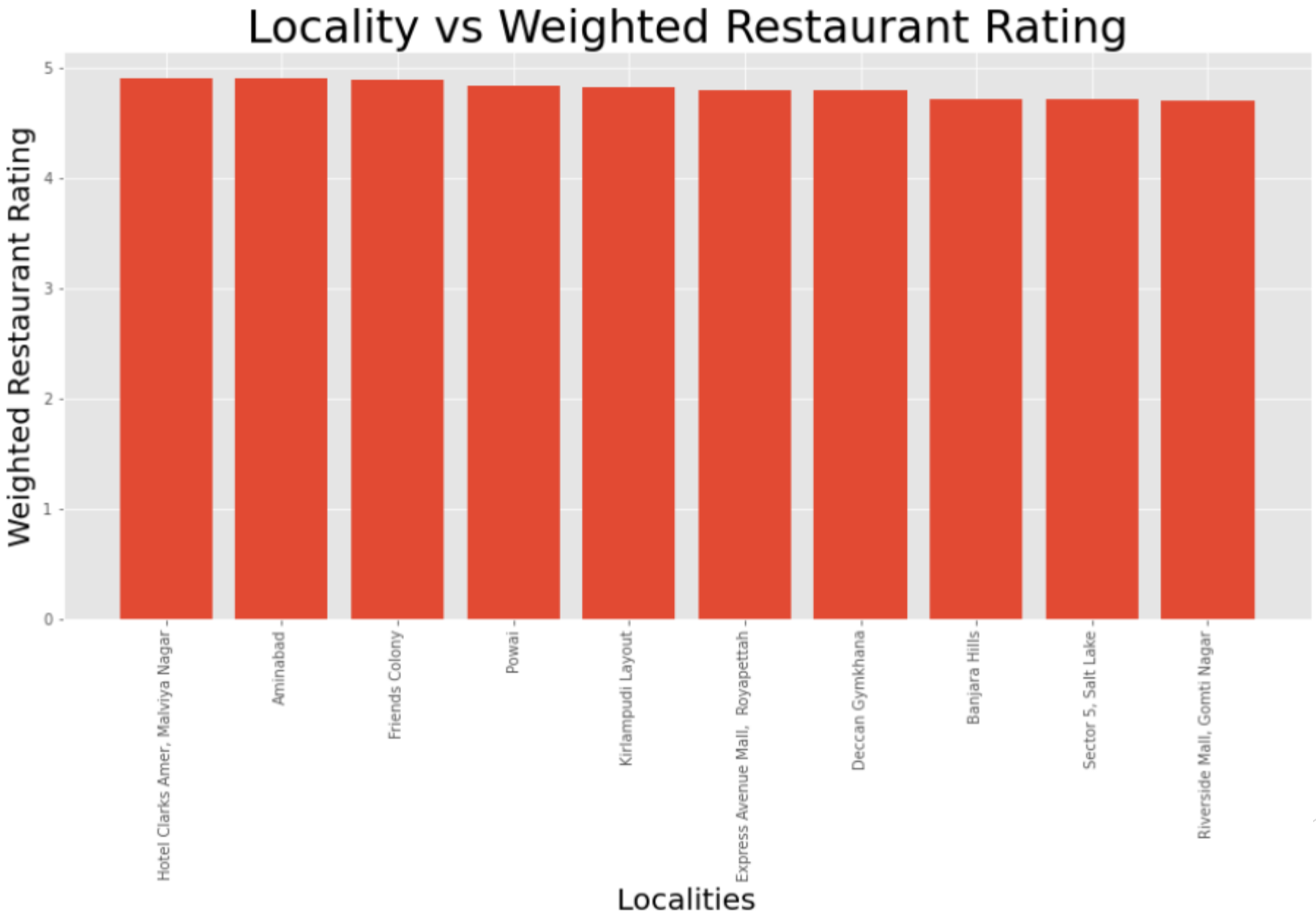
- The necessary libraries were included.
- The dataset was loaded and the appropriate encoding was used which happens to be latin-1 in this case .
- The question specifies to analyze the data for India so the data has been filtered by using the Country Code which is 1 for India.
- **IMPORTANT** – Since in these questions we are analyzing the user rating it is important to get rid of entries for which there is no rating. This has been done by filtering using the “Rating Text” and removing all the data with rating text as “Not rated”.
- A dictionary was made named locality_dict.
- Three numpy arrays are made to store Locality , Aggregate rating and Votes.
- We then run a loop for length of localities array number of times. We now form the dictionary storing the localities as the key and the value as a list of list containing the product Votes*rating sum and total votes.

- We then made a list named locality_weighted_rating which will store the weighted rating and the corresponding location. We calculate the weighted rating according to the given formula and store the result up to two decimal places.
- The we sort the list locality_weighted_rating in reverse order based on the weighted rating.
- Then we make separate lists for localities and rating to be later used to plot graphs.
- We also make a Dataframe just to show the top 10 localities and their weighted ratings in a tabular for and then display it.
- The graph between Localities and weighted restaurant rating is plotted.

Results

Top 10 localities and their weighted restaurant rating

Locality	Weighted Restaurant Rating
Hotel Clarks Amer, Malviya Nagar	4.90
Aminabad	4.90
Friends Colony	4.89
Powai	4.84
Kirlampudi Layout	4.82
Express Avenue Mall, Royapettah	4.80
Deccan Gymkhana	4.80
Banjara Hills	4.72
Sector 5, Salt Lake	4.71
Riverside Mall, Gomti Nagar	4.70


Locality vs Weighted Restaurant Rating
Weighted Restaurant Rating
0
1
2
3
4
5
Hotel Clarks Amer, Malviya Nagar
Aminabad
Friends Colony
Powai
Kirlampudi Layout
Express Avenue Mall, Royapettah
Deccan Gymkhana
Banjara Hills
Sector 5, Salt Lake
Riverside Mall, Gomti Nagar
Localities

We observe that the locality Hotel Clarks Amer, Malviya Nagar has the highest weighted restaurant rating followed by Aminabad and Friends Colony.