# Zomato project

**1.** The dataset is highly skewed toward the cities included in Delhi-NCR. So, we will summarize all the other cities in Rest of India while those in New Delhi, Ghaziabad, Noida, Gurgaon, Faridabad to Delhi-NCR. Doing this would make our analysis turn toward Delhi-NCR v Rest of India.

Basic requirements in 1st cell

Basic libraries required are included prior to 1st part of the 1st question. The following libraries are :

1. pandas

2. numpy

3. matplotlib.pyplot

4. requests

4. json

5. collections

6. HTTPBasicAuth(from requests.auth)

7. figure(from matplotlib.pyplot)

**Code explanation in basic requirements cell**

1.First we are reading the dataset zomato.csv with the help of pandas object pd and the encoding used is latin 1

2.df2 is a copy of original dataframe to avoid accident deletion.

3. In df1 we are taking country code of india which is 1 as our discussion is focused on NCR-Delhi and Rest of India

4. After this we are replacing the city name mentioned under NCR-Delhi to NCR-Delhi with the help of df1.loc .We are doing this make our project easy and other code easily readable and grouping the country helps us in making our calculations fast

# 1.1 Plot the bar graph of number of restaurants present in Delhi NCR vs Rest of India

**Code explanation**

1. NCR_value is storing the restaurant name present in the delhi ncr

2. We are using value_counts() to calculate the the number of each restaurant and sum() to find the total number of restaurants

3.To calculate the number of restaurants present in the Rest of India we are subtracting the NCR_value from total number of restaurants present all over India and we are using similar function value_counts() and sum() to calculate the total number of restaurants present all over India

4. Now we are printing the NCR_value and rest_value with the help of print() command

5.Now talking about the graph

6.height of bar graph is taken as the total sum of restaurants present in NCR-Delhi and rest of India

7. Now we have the set the name of bars as mentioned in the code

8.we have set the title of the graph with the help of plt.title()

9.Now we are using plt.bar() to plot the bar graph

## Conclusion :

From the output we can see that according to this dataset the total restaurants present in the Delhi-NCR is very large as compare to total restaurants present in the rest of India

Delhi-NCR restaurants-7947

Rest of India restaurants-705

## 1.2 Find the cuisines which are not present in restaurant of Delhi NCR but present in rest of India.Check using Zomato API whether this cuisines are actually not served in restaurants of Delhi-NCR or just it due to incomplete dataset.

**1st code explanation :**

1. In NCR_value and rest_value we are storing the name of different cuisines which may or may not be present together .

2. ncr and rest are two empty dictionary created to store the cuisine names with their count .

3.The first for loop is used for finding the values of Delhi-NCR cuisines.

4.The other for loop is used for finding the values of rest of India cuisines.

5. Both the for are working similarly .First we are traversing in NCR_value containing names of ncr cuisenes.we are the using split() and strip to separate the cuisine names which are present together so that we get the actual count.

6. Now we are taking two sets ncr_set and rest_set to store the unique names only .

7. Now to find the cuisines which are present only in rest of India but not in Delhi-NCR we are subtracting the ncr_set form rest_set .

8. We are storing the desired result in diff.

9.Now we are printing the cuisines present only in rest of India.

**2nd code explanation :**

1. we are creating are header in which accept header and user key is passed for authentication

2. Now we are fetching details from the Zomato api to check whether the Cuisines stored in diff are served in Delhi restaurant or not.

3. data is storing the converted data (from json to python )

4.We are converting data with help of json() function

5.d is storing the details of a dictionary cuisines present in the data

6. Now we are using for loop to find whether the cuisine_name is present in cuisines or not

7. output comes out to be True

## Conclusion:

The cuisines which are present only in rest of India are : Malwani,BBQ,Cajun,German

From the above api code it is clear that restaurants present in Delhi-NCR are serving two of these cuisines i.e BBQ and Malwani.Therefore we can say that dataset is incomplete.

## 1.3 Find the top 10 cuisines served by maximum number of restaurants in Delhi NCR and rest of India.

**Code explanation:**

1. In NCR_value and rest_value we are storing the name of different cuisines which may or may not be present together .

2. ncr and rest are two empty dictionary created to store the cuisine names with their count .

3.The first for loop is used for finding the values of Delhi-NCR cuisines.

4.The other for loop is used for finding the values of rest of India cuisines.

5. Both the for are working similarly .First we are traversing in NCR_value containing names of ncr cuisenes.we are the using split() and strip to separate the cuisine names which are present together so that we get the actual count.

6. Now we are sorting the two dictionaries i.e ncr and rest with the help of sorted function which is returning the values in the form of a list of tuples in which name is present at $0^{th}$ value and count is present at $1^{st}$ value .We have set the reverse to False as we want top 10 cuisines and we have done the slicing to get the top 10 cuisines for both the Delhi-NCR and rest of India.

7.Now we are printing the sorted_ncr and sorted_rest with the help of for loop to get the desired top 10 cuisines.

**Conclusion:**

We got the desired result .

## 1.4 Write a short detailed analysis of how cuisine served is different from Delhi NCR to Rest of India. Plot the suitable graph to explain your inference

**Code explanation:**

1. In NCR_value and rest_value we are storing the name of different cuisines which may or may not be present together .

2. ncr and rest are two empty dictionary created to store the cuisine names with their count .

3.The first for loop is used for finding the values of Delhi-NCR cuisines.

4.The other for loop is used for finding the values of rest of India cuisines.

5. Both the for are working similarly .First we are traversing in NCR_value containing names of ncr cuisenes.we are the using split() and strip to separate the cuisine names which are present together so that we get the actual count.

6. Now we are sorting the two dictionaries i.e ncr and rest with the help of sorted function which is returning the values in the form of a list of tuples in which name is present at $0^{th}$ value and count is present at $1^{st}$ value .We have set the reverse to False as we want top 10 cuisines and we have done the slicing to get the top 10 cuisines for both the Delhi-NCR and rest of India.

7.We have created 4 lists to store the cuisines names and their counts for Delhi-NCR as well as rest of India

8.Now talking about the graph plotting. We have plotted pie graphs with the help of plt.pie() and autopct is used for percentage

**Conclusion:**

From the code and graph we can conclude that North Indian and chinese are very popular cuisines in Delhi-NCR as well as in Rest of India.There are some cuisines which are more pupular in rest of India like mexican,cafe. In Delhi-NCR Street food is loved where as its not the case with rest of India.

| Delhi-NCR | Rest of India |
|---|---|
| North Indian | North Indian |
| Chinese | Chinese |
| Fast Food | Continental |
| Mughlai | Italian |
| Bakery | Cafe |
| South Indian | Fast Food |
| Continental | South Indian |
| Desserts | Mughlai |
| Street Food | Desserts |
| Italian | Mexican |

# 2 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

**Code explanation:**

1. We are using figure to set the size facecolor

2. We are plotting scatter graph for aggregate rating and number of votes with plt.scatter(),Taking no. of votes of y-axis and rating on x-axis .

3. We are using plt.show() to display the graph

**Conclusion:**

As we can see from the graph that at 0 rating votes=0. So we can say that no. of votes and rating are interdependent as rating is increasing no. of votes is also increasing . But they are not directly proportional as we can say that we are not getting a straight line.

## 2.1.2 Restaurant serving more number of cuisines

## Code explanation:

df3.Cuisines.fillna('Not Known',inplace=True) # **to fill the nan values with Not Known in Cuisines column**

df4=df3[~df3['Cuisines'].str.contains(',')] # ~ for not condition # **to find the restaurants in which only one cuisine is present**

df5=df3[df3['Cuisines'].str.contains(',')] # **to find the restaurants with multiple cuisines**

print("Average rating of restaurants serving one cuisine : ",df4['Aggregate rating'].mean()) # **printing avg. aggregate rating using mean()**

figure(num=None , figsize=(9,5)) # **setting the figure size**

plt.hist(df4['Aggregate rating'],bins=50) # **plotting histogram for aggregate rating of restaurants with one cuisine only**

plt.title('Rating spread of restaurant serving one cuisine only') # **setting the title**

plt.xlabel("Rating") # **setting the x-label**

plt.show() # **for displaying the graph**

figure(num=None,figsize=(9,5)) # **setting the figure size**

print("Average rating of restaurants serving multiple cuisine : ",df5['Aggregate rating'].mean()) # **calculating mean aggregate rating for restaurants serving multiple cuisines**

plt.hist(df5["Aggregate rating"],bins=50) # # **plotting histogram for aggregate rating of restaurants serving multiple cuisines**

plt.title('Rating spread of restaurants serving multiple cuisne') # **setting the title**

plt.xlabel('Rating') # **setting the x-label**

plt.show() # **to display the graph**

## Conclusion :

Hence we can see that more number of restaurants serving multiple cuisine have a rating between 3-3.5 (peak of the graph) than compared to those serving only one Cuisine. Both multiple and single Cuisine restaurants are almost equally likely to have a rating below 1 Average rating of single Cuisine restaurant is 2.2 whereas for multiple cuisine restaurant is 2.9

## 2.1.3 Average Cost of Restaurant

### Code explanation:

figure(num=None, figsize=(10, 6),facecolor='yellow') # **setting the fgure size**

plt.hist(df3['Average Cost for two'],range=[0,3000], facecolor='black', align='mid',bins=50) # **plotting the histogram for average cost for two and keeping the range 0 to 3000**

plt.show() #to display the graph

figure(num=None, figsize=(10, 6),facecolor='gray') # **setting the figure size**

plt.scatter(df3['Aggregate rating'],df3['Average Cost for two'],c='r',alpha=0.5)# **plotting scatter graph of aggregate rating and average cost for two**

plt.title('Price vs Rating') # **setting the title**

plt.xlabel("Price for two")# **setting the x-label**

plt.ylim(0,5000) # **setting the limit**

### Conclusions:

1. This histogram shows us the spread of price for two. We can see that it peaks before 500 ,therefore we can deduce that majority of the restaurants are in the price range of 0-500

2. This scatter plot shows us that some of the expensive restaurants are in the range of 3000 or 4000 above are generally high rated

## 2.1.4 Restaurant serving some specific cuisines

```
cuisines=df1["Cuisines"] #   taking all the cuisines of india

cuisine=dict()#      empty dictionary with name cuisine

for i in cuisines:   #      now adding cuisines with their counts in dictionary cuisine

  a=i.split(',')

  for j in a:

    j=j.strip()

    if j in cuisine:

      cuisine[j]=cuisine[j]+1

    else:

      cuisine[j]=1

popular=sorted(cuisine.items(),key=lambda kv:kv[1], reverse=True)[:5]    #      sorting the dictionary in descending order to get the popular 5 cuisines

popular_cuisines=[]  #    empty list with name popular_cuisines

for i in popular: #             adding names of top 5 cuisines in our list

  popular_cuisines.append(i[0])

for i in popular_cuisines: #          now extracting the rating in context with top 5 cuisines  with the help of this loop

  rating=[]

  for j, k in zip(df1.Cuisines, df1['Aggregate rating']): #      zipping together cuisines and aggregate rating

    if i in j:

      rating.append(k)

  plt.hist(rating, edgecolor='black', bins=[0, 1, 2, 3, 4, 5])   #             plotting histogram of rating

  plt.xlabel('Rating--->')  #          setting xlabel

  plt.ylabel('Number of restaurants--->')#             setting ylabel

  plt.title('variation of aggregate rating with restaurants serving some specific cuisines(top 5)') #                  setting the title of graph

plt.grid() #              for grid boxes on the graph

plt.legend(labels=popular_cuisines)  #       to distinguish between rating related to different cuisines

plt.show()#               to display the graph
```

**Conclusion:** We can see from the graph that restaurants which are serving popular cuisines have high rating as compare to restaurants serving normal cuisines

## 2.2 Find the weighted restaurant rating of each locality and find out the top 10 localities with more weighted restaurant rating?

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

```
print(df3.groupby('Locality').apply(lambda x: (x['Votes']*x['Aggregate rating']).sum()/(x['Votes'].sum())))

#Weighted Restaurant Rating calculation and printing

print('The top 10 Weighted Restaurant rating of the complete dataset') # print statement

Weightedrating=df3.groupby('Locality').apply(lambda x: (x['Votes']*x['Aggregate rating']).sum()/(x['Votes'].sum()))

Weightedrating.sort_values(ascending=False)[0:10] #  for  top 10 localities with more weighted restaurant rating
```

**Conclusion:**

| Locality | | locality | |
|---|---|---|---|
| Aminabad | 4.900000 | Hotel Clarks Amer, Malviya Nagar | 4.900000 |
| Friends Colony | 4.886916 | Powai | 4.841869 |
| Kirlampudi Layout | 4.820161 | Express Avenue Mall,  Royapettah | 4.800000 |
| Deccan Gymkhana | 4.800000 | Banjara Hills | 4.718762 |
| Sector 5, Salt Lake | 4.707023 | Riverside Mall, Gomti Nagar | 4.700000 |
| Aminabad | 4.900000 | Hotel Clarks Amer, Malviya Nagar | 4.900000 |
| Friends Colony | 4.886916 | Powai | 4.841869 |
| Kirlampudi Layout | 4.820161 | Express Avenue Mall,  Royapettah | 4.800000 |

# 3.Visualization

## 3.1 Plot the bar graph top 15 restaurants have a maximum number of outlets.

**Code Explanation:**

restaurants=df1["Restaurant Name"].value_counts() # **counting the no. Of outlets with the function value_counts(). It return a dictionnary with values and their counts**

#print(restaurants)

sorted_restaurant=sorted(restaurants.items(),key=lambda kv:kv[1], reverse=True)[:15] **# sorting the restaurant dictionary in descending order to get the required 15 restaurants with maximum outlets. Output will be in the form of list of tuple with key and value**

#print(sorted_restaurant)

x1=[]

x2=[]

for i in sorted_restaurant: # **extracting the restaurant name and their counts in two separate lists**

  print(i[0],"",i[1])

  x1.append(i[0])

  x2.append(i[1])

figure(num=None, figsize=(12, 8), dpi=80, facecolor='gray', edgecolor='k') # **setting the figure size**

plt.xticks(rotation=90) # **setting the x-axis**

plt.bar(x1,x2) # **plotting a bar graph between names and the counts**

plt.show() # **for displaying the graph**

**Conclusion:**

The final answer makes good sense as restaurants having most outlets are Huge franchises like - CAFE COFFE DAY, DOMINO'S PIZZA, SUBWAY etc. Through this bar graph we can also see who amongst th ese franchises is the most popular i.e CAFE COFFEE DAY . DOMINO'S PIZZA is a close competitor having 79 outlets respectively.

## 3.2 Plot the histogram of aggregate rating of restaurant( drop the unrated restaurant)

**Code Explanation :**

figure(num=None, figsize=(12, 8), dpi=80, facecolor='gray', edgecolor='k') # **setting the figure size**

plt.hist(df3['Aggregate rating'],color='black',bins=100) # **plotting the histogram of aggregate rating**

plt.xlabel("Aggregate rating") # **naming the x-label**

plt.title("Rating Distribution") # **setting the title**

plt.show() # **to display the graph**

**Conclusion:**

We can say that there are many restaurants having rating 0. Rest of the restaurants rating lie between 3 and 3.5. we got a nearly perfect pyramid shape in the histogram we got from our code

## 3.3 Plot the bar graph top 10 restaurants in the data with the highest number of votes.

**Code explanation:**

votes=df3.groupby("Restaurant Name")["Votes"].sum().sort_values(ascending=False)[0:10] # **we are calculating the votes according to restaurant with help of restaurant name and votes.And sort_values is arranging them in descending order and we are using slicing to get the top 10 restaurant with highest number of votes.Alse total votes of each restaurant are calculated with the help of sum()**

print(votes) # **printing the votes**

restaurant=votes.index # **having the restaurant names**

vote=votes.values # **having the number of votes**

figure(num=None, figsize=(14, 10), facecolor='gray', edgecolor='k') # **setting the figure size**

plt.xticks(rotation=90) # setting the x-axis

plt.bar(restaurant,vote,color='blue')# **plotting bar graph between restaurant name and the votes of the restaurant**

plt.ylabel("Number of votes") # **naming the y-label**

plt.show()# **to display graph**

**Conclusion:**

We are getting the top 10 restaurant with their number of votes. Barbeque Nation is the Restaurant with highest number of votes( 27835 )

## 3.4 Plot the pie graph of top 10 cuisines present in restaurants in the USA

### Code explanation:

```
dfusa=df2[df2['Country Code']==216]   #       setting the country code for us i.e 216

dfusa.Cuisines.fillna('Not Known',inplace=True)#                  filling nan values with Not Known

us_val=df1["Cuisines"].value_counts()#          using value_counts() to get the cuisine name with their count

#print(us_val)

usa=dict() #                    empty dictionary usa

for i in dfusa['Cuisines']:    #              separating cuisine names and adding them to dictionary with their count

   a=i.split(',')

   for j in a:

      j=j.strip()

      if j in usa:

         usa[j]=usa[j]+1

      else:

         usa[j]=1

sorted_usa=sorted(usa.items(),key=lambda kv:kv[1], reverse=True)[:10]  #      sorting the dictionary usa to get the top 10 cuisines

x1=[]

x2=[]

for i in sorted_usa:  #      displaying cuisine name and the count .As we get list of tuples after using sorted() therefore displaying in this
manner

   print("     ",i[0],i[1])

   x1.append(i[0])

   x2.append(i[1])

figure(num=None, figsize=(12, 8), facecolor='w', edgecolor='k')   #     setting the figure size

plt.pie(x2,labels=x1,autopct='%.2f%%',counterclock=False)#        plotting the pie graph of count percentage

plt.show()  #      to display the graph
```

### Conclusion:

American is the top cuisine of the USA and it is served by 112 restaurants

## 3.5 Plot the bubble graph of a number of Restaurants present in the city of India and keeping the weighted restaurant rating of the city in a bubble.

**Code explanation:**

```
city_list=df1['City'].value_counts().index   #      getting the city count

data=list(zip(df1['City'],df1['Aggregate rating'],df1['Votes']))   #   zipping together the city ,aggregate rating  and votes

weightage_rating=[]    #  empty list

for city in city_list:    #    Now calculating weighted restaurant rating  for data with the formula given in 2nd part
   NRate=0
   Tvote=0
   for i in range(len(data)):
      if city in data[i][0]:
         NRate=NRate+(data[i][1]*data[i][2])
         Tvote=Tvote+data[i][2]
   if Tvote!=0:
      weightage_rating.append(NRate/Tvote)
   else:
      weightage_rating.append(0)
figure(num=None, figsize=(16, 10), facecolor='gray', edgecolor='k')  #              setting the figure size

plt.bar(city_list[0:12],weightage_rating[0:12])     # taking for 12 cities

plt.xlabel("City")  #              naming x-label

plt.ylabel("Weighted Restaurant Rating")    #               naming y-label

plt.title("City vs Weighted Restaurant Rating") #         setting  the title

plt.show() #      to display the graph
```

**Conclusion:**

Bangalore has the maximum weighted restaurant rating

# Thank you

1. Thank you for checking my project wisely.
2. Make sure you execute every cell to avoid errors in the code
   Codes are interconnected. And execution of cell with basic
   Requirements and libraries is mandatory.
3. please read all the comments and markdowns carefully
4. And make sure to add complete directory
5. Thank you :)