

# ZOMATO

## STRATEGIC GOAL

- \*Improve customer experience.
- \*Focused item promotions.
- \*Business opportunities for expansion.

## OBJECTIVE

- \*Recommend top ranked restaurants to the customer
- \*Recommend the restaurants to the customer for their menu choices
- \*Analysis of menuitems across various location and recommend less/more competitive location for promotion of menu items
- \*Recommend the location needs on menu item choices, restarurant choices, ... etc

## User and Data Customers

Customers, Restaurant Managers, Business unit Heads

## Explaratory Data Analysis

### Field Definition

Each row represents the restaurant of Bangalore city in which it shows the following information about the restaurants by column wise. Given below,

**URL- Website of restaurants.**

**Address- Address of restaurants.**

**Name - Name of the restaurants.**

**Online\_order - Online\_order facility present or not.**

**Book\_table - Booking table facility present or not.**

**Rate - Rating of the restaurants by people.**

**Votes - Votes given by people for the restaurants.**

**Phone - Contact number of the restaurants.**

**Location - Location of the restaurants.**

**Rest\_type - Type of restaurants.**

**Dish\_liked - Dishes which are best in the restaurants.**

**Cuisines - Dishes by area wise whether it is north indian or south indian etc.,**

**Approx\_cost(for two people) - Approximate cost for two people in the restaurants.**

**Reviews\_list - Reviews given by the people for the restaurants.**

**Menu\_items - Dishes present in the restaurants.**

**Listed\_in(type) - Type of Dish present in the restaurants.**

**Listed\_in(city) - City of the restaurant.**

## Data Cleanup

Identified the fields that are not needed for the analysis to meet the objectives. Removed Fields: URL, Address, Phone, Review list and Menu items.

Removed duplicate restaurant records. Removed empty records.

## After Cleanup

## IMPORTING PACKAGES

```
In [40]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
plt.style.use('ggplot')
```

## IMPORTING DATASET

```
In [41]: df = pd.read_excel('Downloads\zom.xlsx')
```

In [42]: `df.isnull().sum()`

```
Out[42]: Unnamed: 0      0
         address      0
         name        0
         online_order  0
         book_table    0
         rate         0
         location     0
         rest_type     0
         dish_liked    0
         cuisines      0
         approx_cost(for two people)  0
         meal_type     0
         city         0
         votes        0
         dtype: int64
```

## Describing the data

In [43]: `df.describe()`

```
Out[43]:
```

	Unnamed: 0	approx_cost(for two people)	votes
count	19233.000000	19233.000000	19233.000000
mean	9616.000000	708.809858	420.982582
std	5552.233199	503.524302	699.775971
min	0.000000	40.000000	0.000000
25%	4808.000000	400.000000	94.000000
50%	9616.000000	600.000000	199.000000
75%	14424.000000	800.000000	462.000000
max	19232.000000	6000.000000	16345.000000

In [44]: df.info

```
Out[44]: <bound method DataFrame.info of          Unnamed: 0
address \
0          0  # 31, 7th Cross, Opposite Canara Bank, Domlur,...
1          1  # 31, 7th Cross, Opposite Canara Bank, Domlur,...
2          2  # 31, 7th Cross, Opposite Canara Bank, Domlur,...
3          3  # 31, 7th Cross, Opposite Canara Bank, Domlur,...
4          4  #10061 Tower 10, Prestige Shanti Niketan., Whi...
...          ...
19228      19228  Zone By The Park, 26/A, Electronic City, Banga...
19229      19229  Zone by The Park, 26/A, Phase 1, Electronic Ci...
19230      19230  Zone by The Park, 26/A, Phase 1, Electronic Ci...
19231      19231  c/o Foodism, 2nd A Main Road, 445/31, 7th Cros...
19232      19232  c/o Foodism, 2nd A Main Road, 445/31, 7th Cros...

          name online_order book_table  rate \
0          Vinny's          Yes         No  4.1/5
1          Vinny's          Yes         No  4.1/5
2          Vinny's          Yes         No  4.1/5
3          Vinny's          Yes         No  4.1/5
4          Cupcake Couture        No         No  3.8/5
...          ...          ...         ...  ...
19228          Z-Bar          No         Yes  4.1/5
19229  Bazaar - Zone By The Park        No         Yes  4.0/5
19230  Bazaar - Zone By The Park        No         Yes  4.0/5
19231  Artinci Artisanal Ice Cream        Yes         No  4.2/5
19232  Artinci Artisanal Ice Cream        Yes         No  4.2/5

          location  rest_type \
0          Domlur  Casual Dining
1          Domlur  Casual Dining
2          Domlur  Casual Dining
3          Domlur  Casual Dining
4  Whitefield  Delivery
...          ...          ...
19228  Electronic City          Bar
19229  Electronic City  Casual Dining
19230  Electronic City  Casual Dining
19231  Marathahalli  Dessert Parlor
19232  Marathahalli  Dessert Parlor

          dish_liked \
0  Pizza, Pasta, Mocktails, Tiramisu, Garlic Brea...
1  Pizza, Pasta, Mocktails, Tiramisu, Garlic Brea...
2  Pizza, Pasta, Mocktails, Tiramisu, Garlic Brea...
3  Pizza, Pasta, Mocktails, Tiramisu, Garlic Brea...
4          Cup Cake, Chocolate Truffle
...          ...
19228  Cocktails, Tandoori Chicken, Wine, Ice Tea, Fr...
19229  Salads, Chicken Tikka, Roti, Gulab Jamun, Lunc...
19230  Salads, Chicken Tikka, Roti, Gulab Jamun, Lunc...
19231          Coffee, Vanilla Ice Cream, Almond Ice Cream
19232          Coffee, Vanilla Ice Cream, Almond Ice Cream
```

```
          cuisines \
0  Burger, Desserts, Italian, Pizza, Salad
```

```

1      Burger, Desserts, Italian, Pizza, Salad
2      Burger, Desserts, Italian, Pizza, Salad
3      Burger, Desserts, Italian, Pizza, Salad
4      Desserts
...
19228      Finger Food, North Indian
19229      Continental, South Indian, North Indian, Asian
19230      Continental, South Indian, North Indian, Asian
19231      Desserts, Ice Cream
19232      Desserts, Ice Cream

```

```

      approx_cost(for two people)      meal_type      city \
0      600      Delivery      Koramangala 6th Block
1      600      Delivery      Old Airport Road
2      600      Dine-out      Indiranagar
3      600      Dine-out      Old Airport Road
4      300      Delivery      Whitefield
...
19228      1500      Pubs and bars      Electronic City
19229      1400      Buffet      Electronic City
19230      1400      Dine-out      Electronic City
19231      300      Delivery      Marathahalli
19232      300      Desserts      Marathahalli

```

```

      votes
0      366
1      367
2      360
3      366
4      40
...
19228      410
19229      415
19230      415
19231      44
19232      44

```

```
[19233 rows x 14 columns]>
```

## Columns present in the Data

```
In [45]: df.columns
```

```
Out[45]: Index(['Unnamed: 0', 'address', 'name', 'online_order', 'book_table', 'rate',
               'location', 'rest_type', 'dish_liked', 'cuisines',
               'approx_cost(for two people)', 'meal_type', 'city', 'votes'],
              dtype='object')
```

## Shape of the Data

```
In [46]: df.shape
```

```
Out[46]: (19233, 14)
```

## Count of the Data

In [47]: df.count

```
Out[47]: <bound method DataFrame.count of          Unnamed: 0
address \
0          0 # 31, 7th Cross, Opposite Canara Bank, Domlur,...
1          1 # 31, 7th Cross, Opposite Canara Bank, Domlur,...
2          2 # 31, 7th Cross, Opposite Canara Bank, Domlur,...
3          3 # 31, 7th Cross, Opposite Canara Bank, Domlur,...
4          4 #10061 Tower 10, Prestige Shanti Niketan., Whi...
...          ...
19228      19228 Zone By The Park, 26/A, Electronic City, Banga...
19229      19229 Zone by The Park, 26/A, Phase 1, Electronic Ci...
19230      19230 Zone by The Park, 26/A, Phase 1, Electronic Ci...
19231      19231 c/o Foodism, 2nd A Main Road, 445/31, 7th Cros...
19232      19232 c/o Foodism, 2nd A Main Road, 445/31, 7th Cros...

          name online_order book_table  rate \
0          Vinny's          Yes         No  4.1/5
1          Vinny's          Yes         No  4.1/5
2          Vinny's          Yes         No  4.1/5
3          Vinny's          Yes         No  4.1/5
4          Cupcake Couture        No         No  3.8/5
...          ...          ...          ...
19228          Z-Bar          No         Yes  4.1/5
19229      Bazaar - Zone By The Park        No         Yes  4.0/5
19230      Bazaar - Zone By The Park        No         Yes  4.0/5
19231      Artinci Artisanal Ice Cream        Yes         No  4.2/5
19232      Artinci Artisanal Ice Cream        Yes         No  4.2/5

          location      rest_type \
0          Domlur      Casual Dining
1          Domlur      Casual Dining
2          Domlur      Casual Dining
3          Domlur      Casual Dining
4      Whitefield      Delivery
...          ...          ...
19228      Electronic City          Bar
19229      Electronic City      Casual Dining
19230      Electronic City      Casual Dining
19231      Marathahalli      Dessert Parlor
19232      Marathahalli      Dessert Parlor

          dish_liked \
0      Pizza, Pasta, Mocktails, Tiramisu, Garlic Brea...
1      Pizza, Pasta, Mocktails, Tiramisu, Garlic Brea...
2      Pizza, Pasta, Mocktails, Tiramisu, Garlic Brea...
3      Pizza, Pasta, Mocktails, Tiramisu, Garlic Brea...
4          Cup Cake, Chocolate Truffle
...          ...
19228      Cocktails, Tandoori Chicken, Wine, Ice Tea, Fr...
19229      Salads, Chicken Tikka, Roti, Gulab Jamun, Lunc...
19230      Salads, Chicken Tikka, Roti, Gulab Jamun, Lunc...
19231          Coffee, Vanilla Ice Cream, Almond Ice Cream
19232          Coffee, Vanilla Ice Cream, Almond Ice Cream
```

```
          cuisines \
0      Burger, Desserts, Italian, Pizza, Salad
```

```

1      Burger, Desserts, Italian, Pizza, Salad
2      Burger, Desserts, Italian, Pizza, Salad
3      Burger, Desserts, Italian, Pizza, Salad
4      Desserts
...
19228      Finger Food, North Indian
19229      Continental, South Indian, North Indian, Asian
19230      Continental, South Indian, North Indian, Asian
19231      Desserts, Ice Cream
19232      Desserts, Ice Cream

```

```

      approx_cost(for two people)      meal_type      city \
0      600      Delivery      Koramangala 6th Block
1      600      Delivery      Old Airport Road
2      600      Dine-out      Indiranagar
3      600      Dine-out      Old Airport Road
4      300      Delivery      Whitefield
...
19228      1500      Pubs and bars      Electronic City
19229      1400      Buffet      Electronic City
19230      1400      Dine-out      Electronic City
19231      300      Delivery      Marathahalli
19232      300      Desserts      Marathahalli

```

```

      votes
0      366
1      367
2      360
3      366
4      40
...
19228      410
19229      415
19230      415
19231      44
19232      44

```

```
[19233 rows x 14 columns]>
```

## Datatype of each column



```
In [48]: df.dtypes
```

```
Out[48]: Unnamed: 0          int64
address          object
name             object
online_order     object
book_table       object
rate             object
location         object
rest_type        object
dish_liked       object
cuisines         object
approx_cost(for two people)  int64
meal_type        object
city             object
votes            int64
dtype: object
```

### Count of the ratings

Removing the null values present in the rating column.

```
In [49]: df['rate'].value_counts()
df['rate'].isnull().sum()
df['rate'] = df['rate'].apply(lambda x: str(x).split('/')[0])
df['rate'].value_counts()
```

```
Out[49]: 3.9      3012
4.0      2629
4.1      2463
3.8      2288
4.2      1698
3.7      1448
4.3      1148
3.6       762
4.4       598
3.5       424
3.4       320
2.8       286
2.9       261
4.5       259
3.1       237
3.0       236
3.2       209
3.3       190
2.7       162
NEW       135
4.6       119
2.6       112
4.7        53
2.5        44
2.4        44
2.3        33
2.2        23
2.1        11
2.0        11
4.8        10
4.9         8
Name: rate, dtype: int64
```

```

In [50]: g23 = list(df['rate'])
for i in range(0, len(g23)):
    if g23[i] == 'nan':
        g23[i] = 'unrated'
    elif g23[i] == '-':
        g23[i] = 'unrated'
    elif g23[i] == 'NEW':
        g23[i] = 'unrated'

df['rate'] = g23

df['rate'].value_counts()

g23 = list(df['rate'])
for i in range(0, len(g23)):
    if g23[i] == 'unrated':
        g23[i] = None
    else :
        g23[i] = float(g23[i])
df['rate'] = g23
df['rate'].value_counts()

```

```

Out[50]: 3.9    3012
         4.0    2629
         4.1    2463
         3.8    2288
         4.2    1698
         3.7    1448
         4.3    1148
         3.6     762
         4.4     598
         3.5     424
         3.4     320
         2.8     286
         2.9     261
         4.5     259
         3.1     237
         3.0     236
         3.2     209
         3.3     190
         2.7     162
         4.6     119
         2.6     112
         4.7      53
         2.5      44
         2.4      44
         2.3      33
         2.2      23
         2.0      11
         2.1      11
         4.8      10
         4.9       8
         Name: rate, dtype: int64

```

### Count of city

```
In [51]: df['city'].value_counts()
```

```
Out[51]: BTM 1225
Koramangala 7th Block 1132
Koramangala 4th Block 1122
Koramangala 5th Block 1047
Koramangala 6th Block 979
Jayanagar 925
Indiranagar 879
Church Street 802
Brigade Road 779
MG Road 734
JP Nagar 689
Lavelle Road 684
Residency Road 625
HSR 611
Old Airport Road 599
Basavanagudi 562
Brookefield 522
Whitefield 517
Marathahalli 496
Bannerghatta Road 482
Frazer Town 477
Kalyan Nagar 453
Malleshwaram 436
Kammanahalli 434
Bellandur 424
Sarjapur Road 404
Banashankari 347
Rajajinagar 314
Electronic City 288
New BEL Road 245
Name: city, dtype: int64
```

```
In [52]: df['location'].isnull().sum()
```

```
Out[52]: 0
```

```
In [53]: len(df['city'].value_counts())
```

```
Out[53]: 30
```

## UNIVARIATE ANALYSIS

```
In [54]: # plotting number of restaurants that take online_order
sns.set_context("paper", font_scale = 2, rc = {"font.size": 20, "axes.titlesize":
sns.catplot(data = df, kind = 'count', x = 'online_order')
plt.title('Number of restaurants that take order online')
plt.show()
```

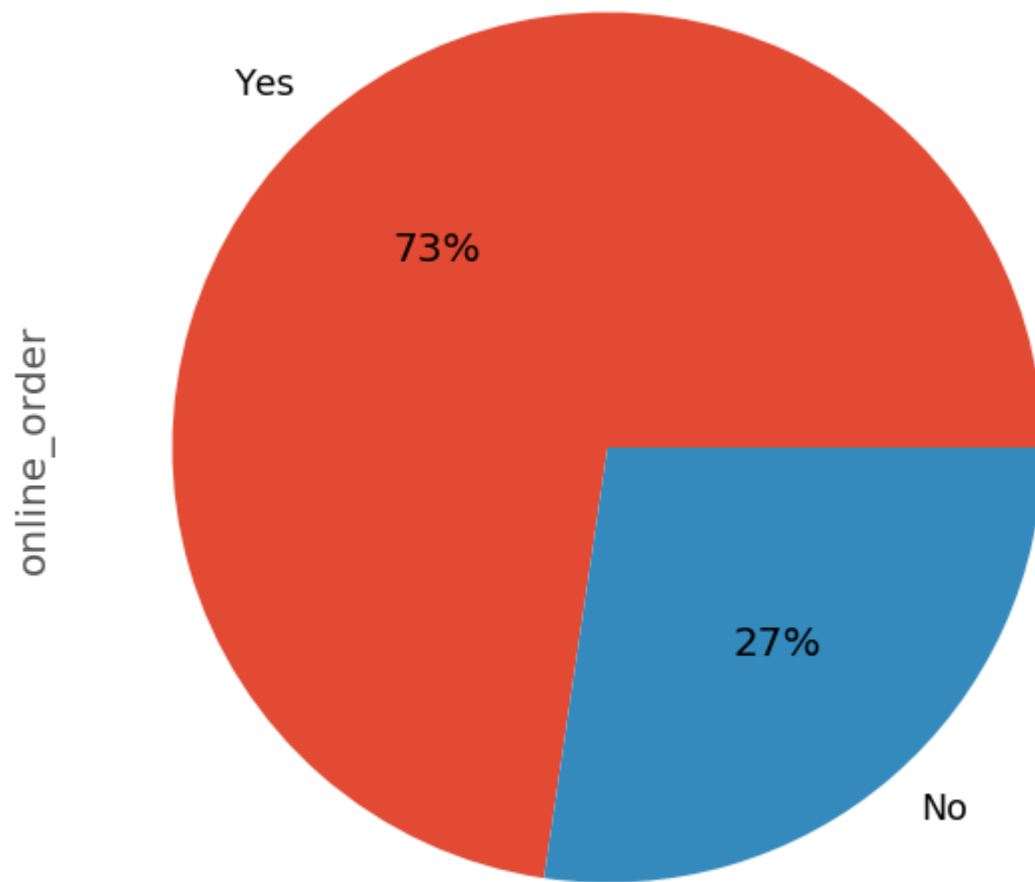
## Number of restaurants that take order online



Type *Markdown* and LaTeX:  $\alpha^2$

```
In [55]: df['online_order'].value_counts().plot(kind='pie', figsize=(10,10), autopct='%1.0
```

```
Out[55]: <matplotlib.axes._subplots.AxesSubplot at 0x1e418378208>
```



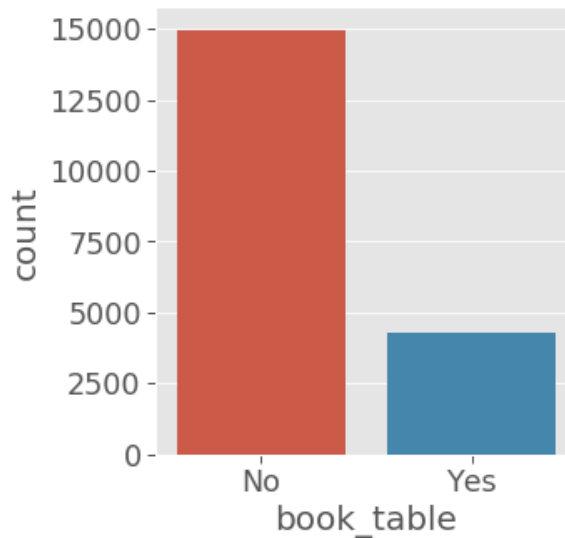
## Conclusion

The IT field has the most value in developing an application for the hotels which are not taking online orders.

and mostly Bangalore people takes online order.

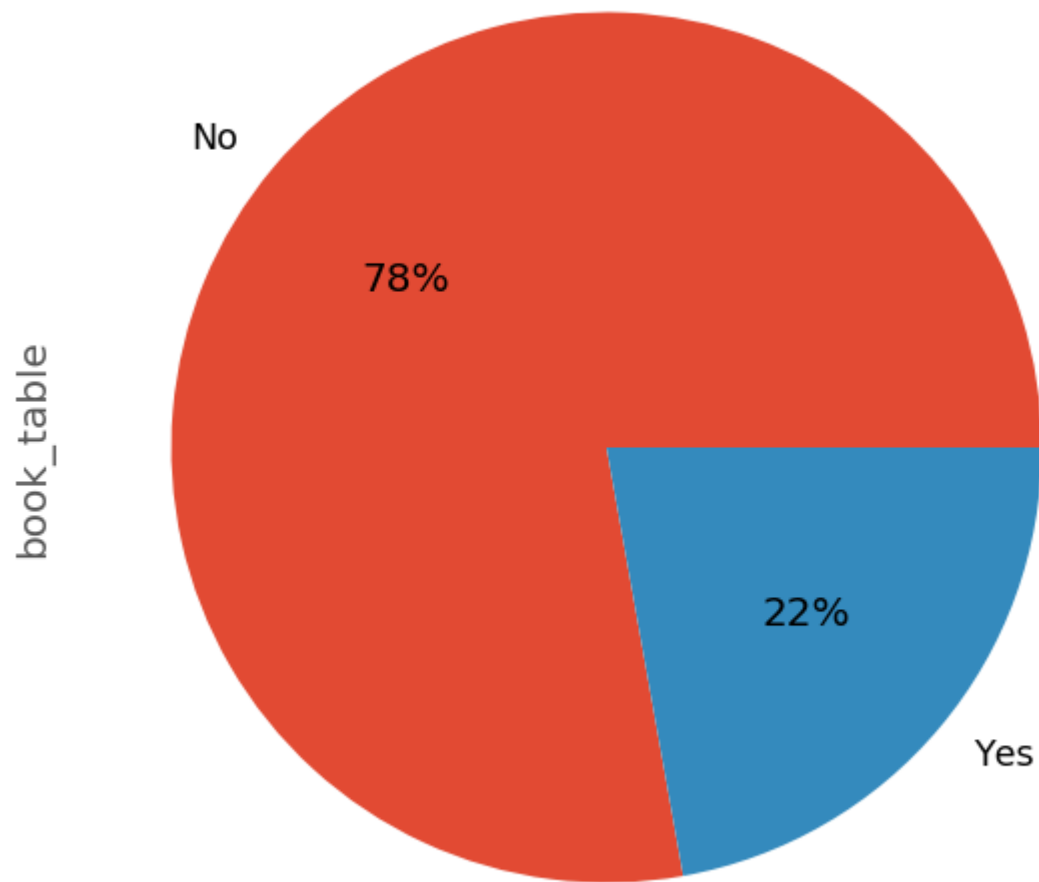
```
In [56]: sns.set_context("paper", font_scale = 2, rc = {"font.size": 20, "axes.titlesize"  
sns.catplot(data = df, kind = 'count', x = 'book_table')  
plt.title('Number of restaurants that have the option to book table')  
plt.show()
```

Number of restaurants that have the option to book table



```
In [57]: df['book_table'].value_counts().plot(kind='pie', figsize=(10,10),autopct='%1.0f%%')
```

```
Out[57]: <matplotlib.axes._subplots.AxesSubplot at 0x1e415de1a48>
```



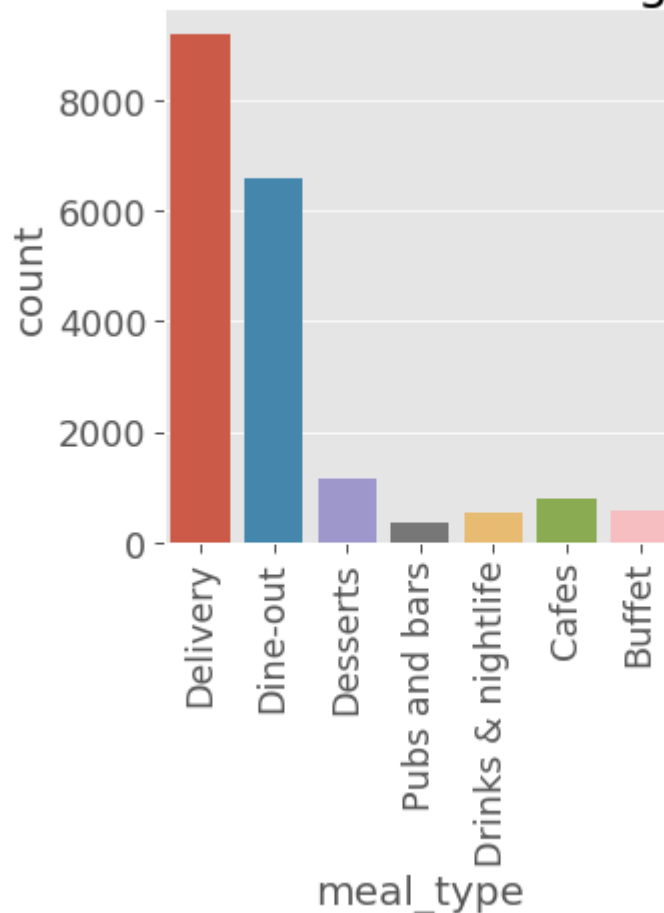


## Conclusion

We see that very high number of restaurants does not offer the service to book table, and very less do. This means that people at Bangalore prefer to eat at their homes or prefer fast food (snacks — quick bites)

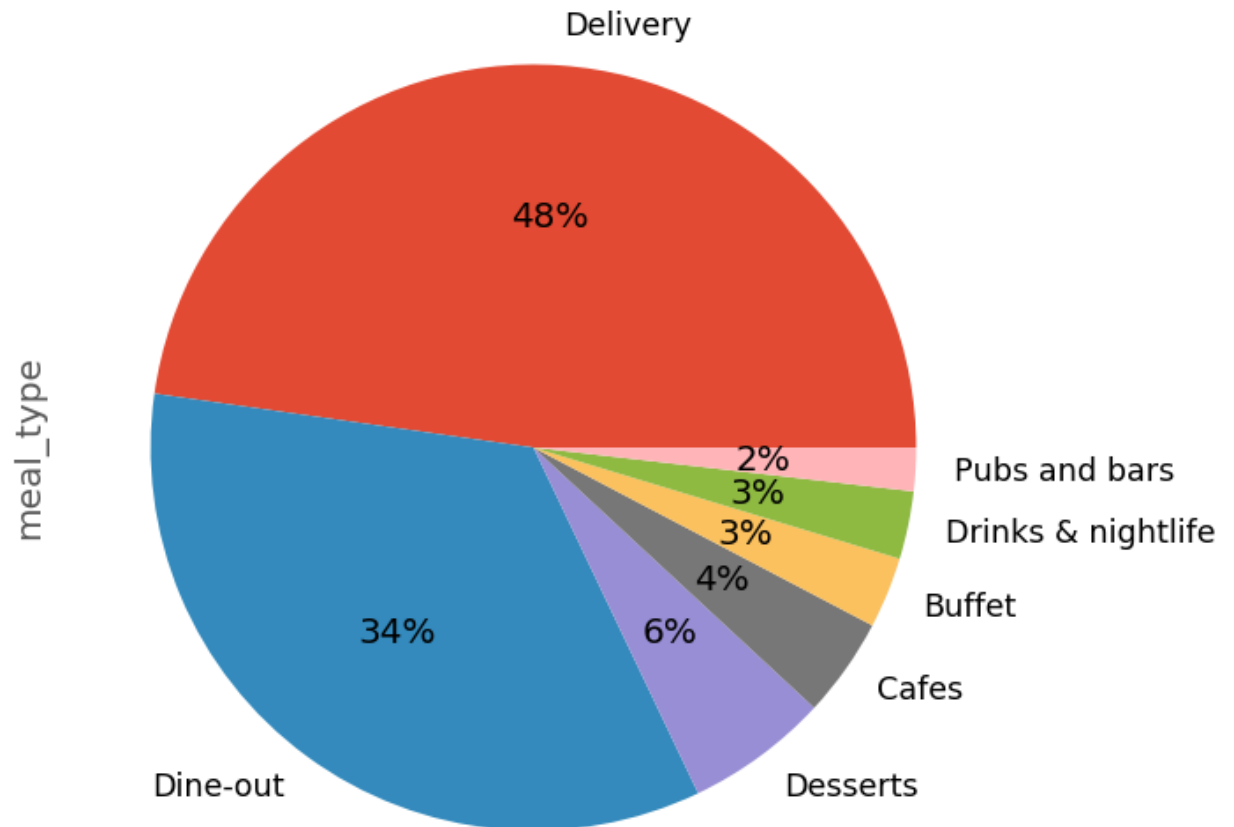
```
In [58]: sns.set_context("paper", font_scale = 2, rc = {"font.size": 20, "axes.titlesize":  
sns.catplot(data = df, kind = 'count', x = 'meal_type')  
plt.title('Number of restaurants according to meal type')  
plt.xticks(rotation=90)  
plt.show()
```

### Number of restaurants according to meal type



```
In [59]: df['meal_type'].value_counts().plot(kind='pie', figsize=(10,10), autopct='%1.0f%%')
```

```
Out[59]: <matplotlib.axes._subplots.AxesSubplot at 0x1e4198e2908>
```



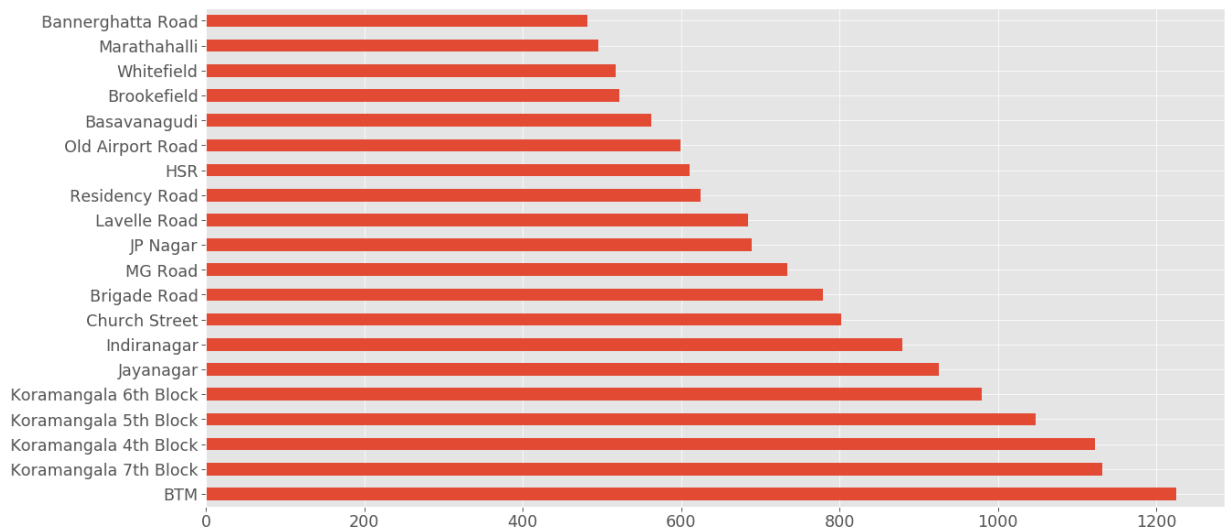
## Conclusion

**\*We see that the majority of restaurants offer delivery, which might account for the above conclusion that people at Bangalore prefer to eat at home.**

**\*Pubs and Bars and Buffet are offered by very less number of restaurants.**

```
In [60]: df['city'].value_counts().head(20).plot(kind='barh', figsize=(20,10))
```

```
Out[60]: <matplotlib.axes._subplots.AxesSubplot at 0x1e419812a48>
```



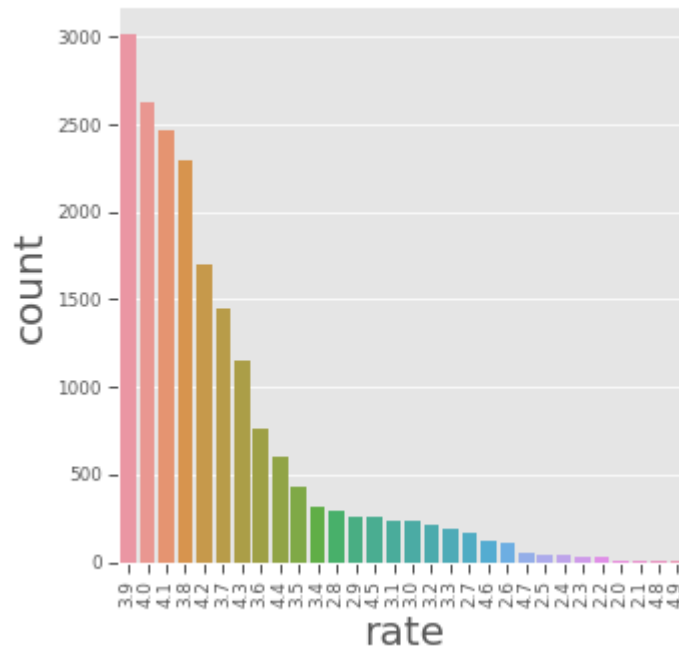
## Conclucion

**\*BTM has the highest number of restaurants, followed by Koramangala 7th Block. New BEL Road has the least number of restaurants, followed by Banashankari.**

**\*It seems that the main foodies live in BTM and Koramangala.**

```
In [61]: sns.set_context("paper", font_scale = 1, rc = {"font.size": 20, "axes.titlesize":
g23 = sns.catplot(data = df, kind = 'count', x = 'rate', order = df['rate'].value
plt.title('Number of restaurants for each rating')
g23.set_xticklabels(rotation = 90)
plt.show()
```

## Number of restaurants for each rating



## Conclucision

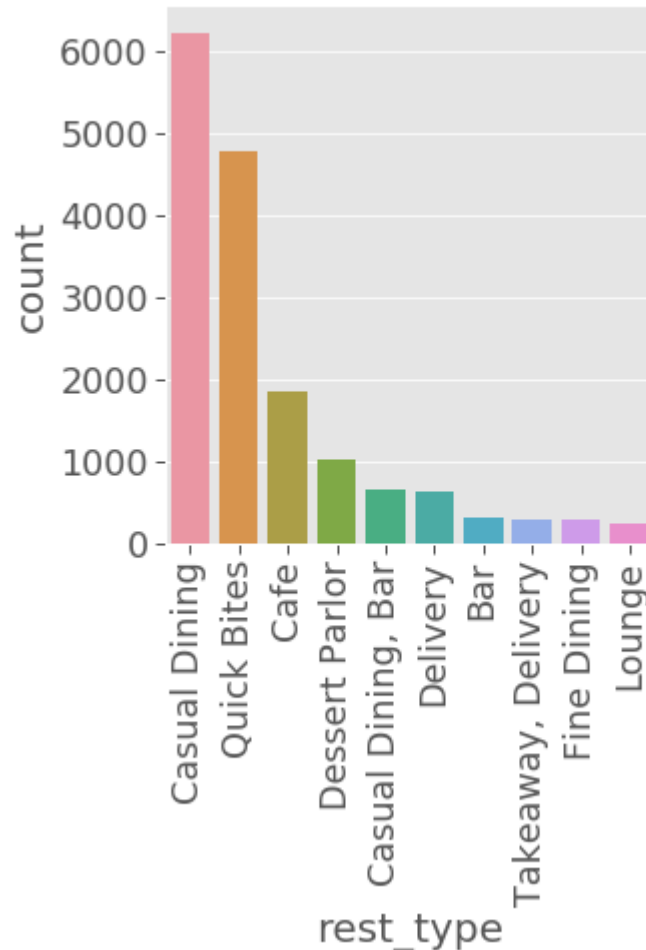
**\*The majority of restaurants are rated 3.9(out of 5), followed by 3.8 and 3.7, which are decent ratings.**

**\*This means that most of the restaurants at Bangalore are liked by the citizens and hence rated above average**

**\*This shows that due to high rise in the number of restaurants in Bangalore, the competition is at its best, as every restaurant is trying to excel in quality and services to increase the number of customers, and hence enhance profit**

```
In [62]: sns.set_context("paper", font_scale = 2, rc = {"font.size": 20, "axes.titlesize":  
b = sns.catplot(data = df, kind = 'count', x = 'rest_type', order = df['rest_type'  
plt.title('Number of restaurants for each type')  
b.set_xticklabels(rotation = 90)  
plt.show()
```

## Number of restaurants for each type



## Conclucision

**\*We see that, the most number of restaurants are of the type casual dining.**

**\*From the above plot it is clear that people at Bangalore prefer 'quick bites' more often.**

**\*This supports our above conclusion, which was drawn from the count plot of book\_table feature, that citizens of Bangalore prefer fast food-quick bites more often, this is why many restaurants do not offer table booking facility**

```
In [63]: df['rest_type'].value_counts()
```

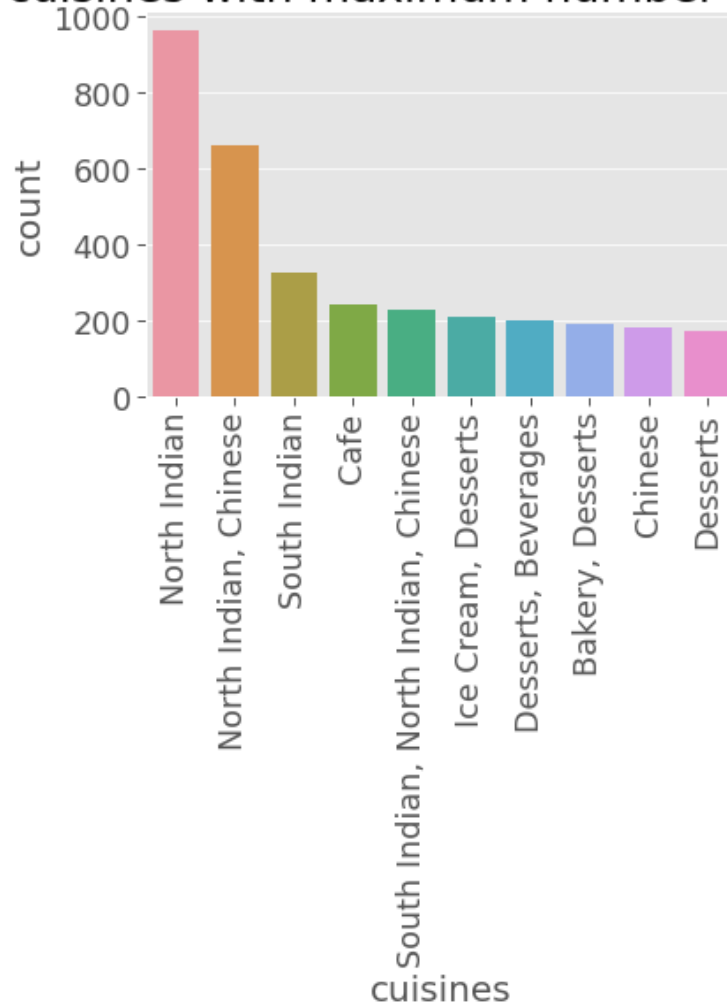
```
Out[63]: Casual Dining          6221
Quick Bites          4779
Cafe                 1839
Dessert Parlor       1005
Casual Dining, Bar    641
...
Dessert Parlor, Kiosk      2
Microbrewery, Bar         2
Cafe, Lounge             2
Dhaba                    1
Food Court, Casual Dining  1
Name: rest_type, Length: 73, dtype: int64
```

```
In [64]: df['cuisines'].value_counts()
```

```
Out[64]: North Indian          964
North Indian, Chinese         658
South Indian                 325
Cafe                       243
South Indian, North Indian, Chinese 227
...
Chinese, Vietnamese, Thai, Malaysian      1
Coffee, South Indian                      1
Biryani, Andhra, North Indian, Seafood     1
Biryani, Kebab, Continental               1
Cafe, Healthy Food, North Indian, Biryani, Continental, Desserts 1
Name: cuisines, Length: 1602, dtype: int64
```

```
In [65]: b = sns.countplot(data = df, x = 'cuisines', order = df.cuisines.value_counts().\n      b.set_xticklabels(b.get_xticklabels(),rotation = 90)\n      plt.title('Top 10 cuisines with maximum number of restaurants')\n      plt.show()
```

Top 10 cuisines with maximum number of restaurants

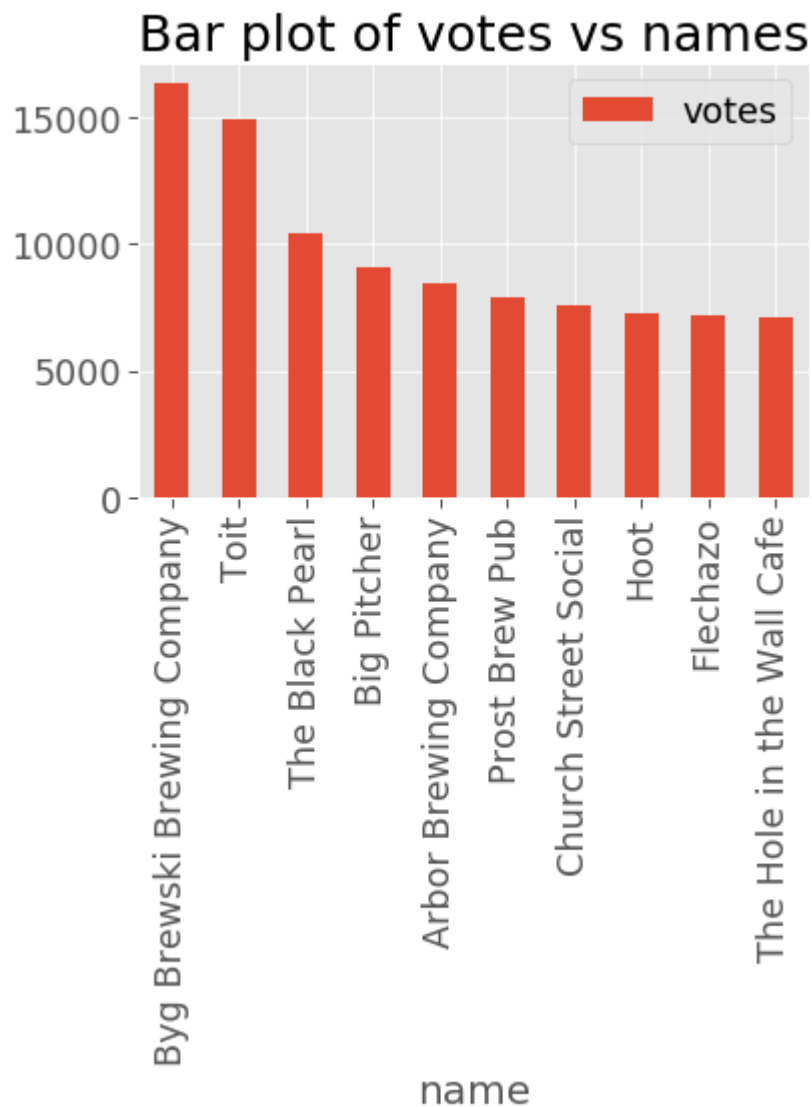


## Conlcusion

\*We see that North Indian cuisine is offered by most number of restaurants.

\*This is worth discussing, as Bangalore is situated in South India and yet people there prefer to eat North Indian cuisine

```
In [66]: df[['votes', 'name']].groupby(['name']).median().sort_values("votes", ascending :  
plt.title('Bar plot of votes vs names')  
plt.show()
```



## Conlcusion

\*We see that Byg Brewski Brewing Company restaurant has the maximum number of upvote.



\*We see that this restaurant has outstanding rating, 4.9 out of 5.0.

\*The restaurant is a Microbrewery type of restaurant, and offers the above mentioned cuisines.

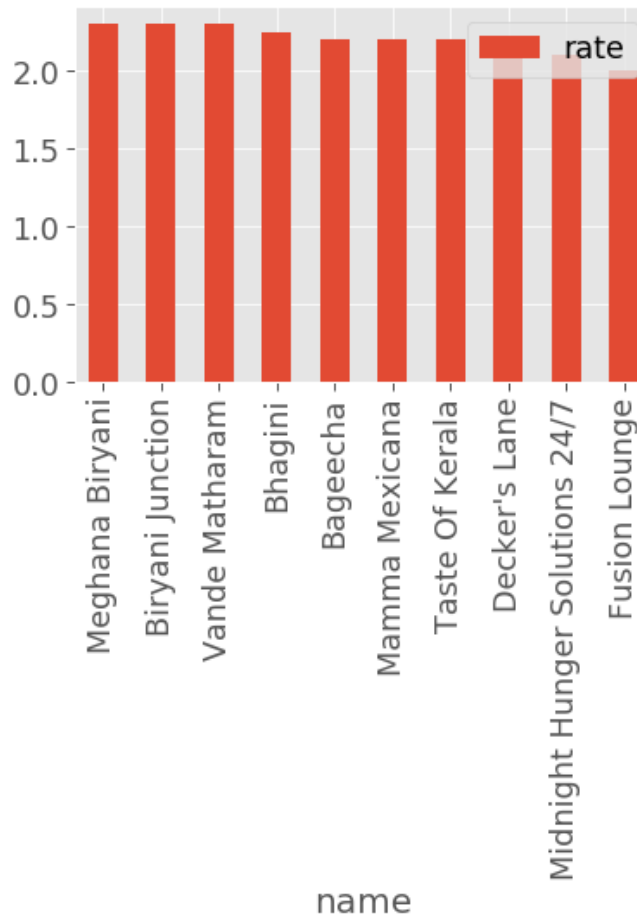
\*The dishes liked by the people are also mentioned.

\*We see that the meal type for different branches of the restaurant include Delivery, Dine-out and Drinks & nightlife, this means that Byg Brewski Brewing Company restaurant is one of those strong competitors in these types of meals.

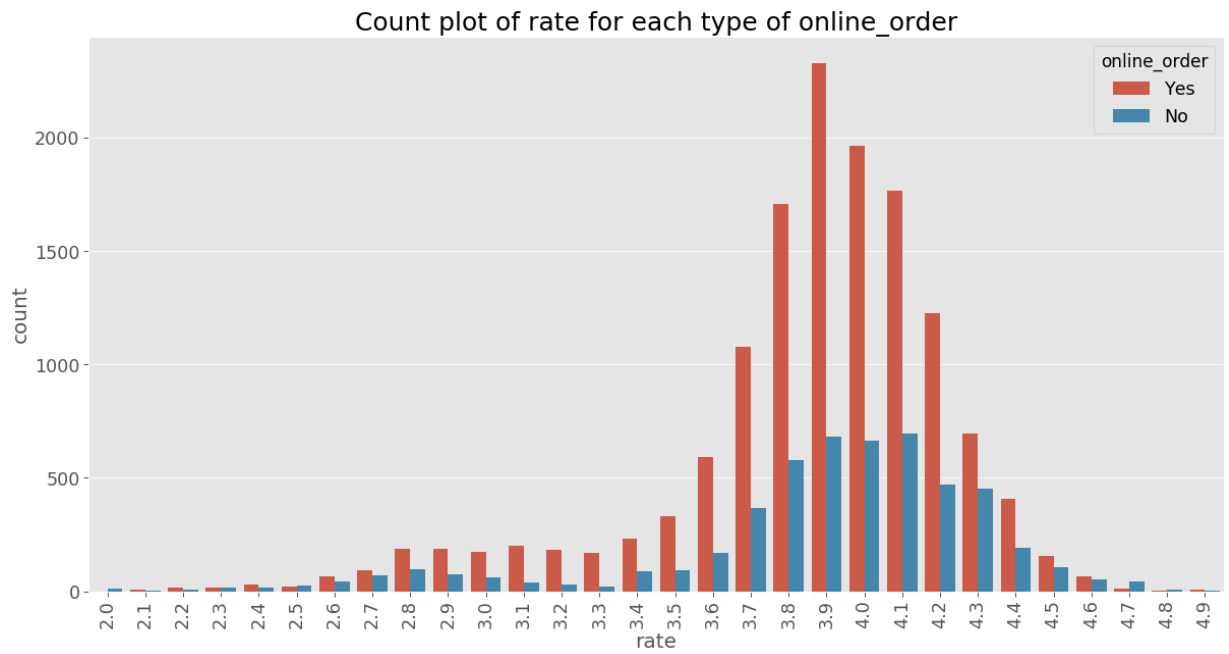
```
In [67]: b=[]
a = df.iloc[:, :].values
for i in range(0, len(a)):
    if a[i][0] == 'Byg Brewski Brewing Company' or a[i][0] == 'Byg Brewski Brewin
        b.append(list(a[i]))
```

```
In [68]: a = pd.DataFrame(df['rate'])
a['name'] = df['name']
a = a.dropna(axis = 0, how = 'any')
a[['rate', 'name']].groupby(['name']).median().sort_values("rate", ascending = F)
plt.title('Bar plot of rate vs names for 10 least rated restaurants')
plt.show()
```

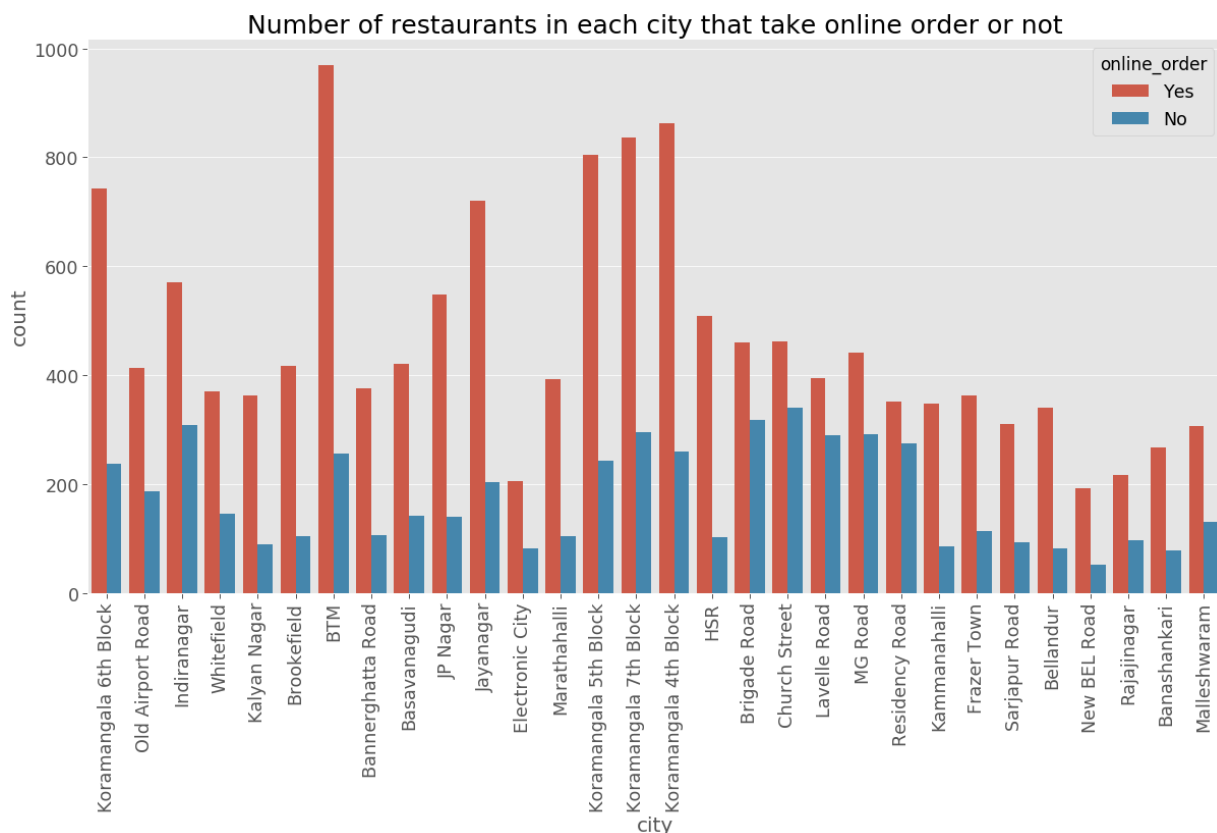
Bar plot of rate vs names for 10 least rated restaurants



```
In [69]: plt.figure(figsize=(20,10))
b=sns.countplot(data = df, hue = 'online_order', x = 'rate')
plt.title('Count plot of rate for each type of online_order')
plt.xticks(rotation = 90)
plt.show()
```



```
In [70]: plt.figure(figsize=(20,10))
sns.set_context("paper", font_scale = 2, rc = {"font.size": 20,"axes.titlesize":
b = sns.countplot(data = df, x = 'city', hue = 'online_order')
plt.title('Number of restaurants in each city that take online order or not')
b.set_xticklabels(b.get_xticklabels(),rotation = 90)
plt.show()
```



**We see that BTM has the highest number of restaurants where you can or can not book table.**

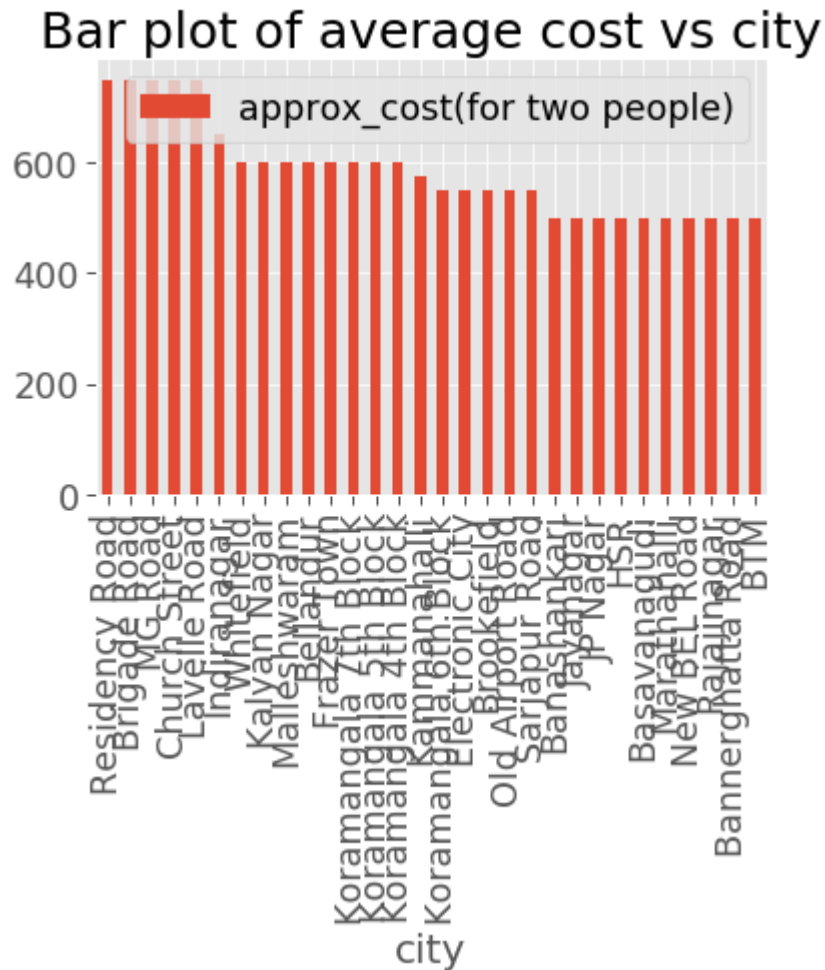
```
In [71]: sns.set_context("paper", font_scale = 2, rc = {"font.size": 20,"axes.titlesize":
b = sns.countplot(data = df, x = 'online_order', hue = 'book_table')
plt.title('Number of restaurants in online order in which you can book a table or not')
b.set_xticklabels(b.get_xticklabels(),rotation = 90)
plt.show()
```

Number of restaurants in online order in which you can book a table or not



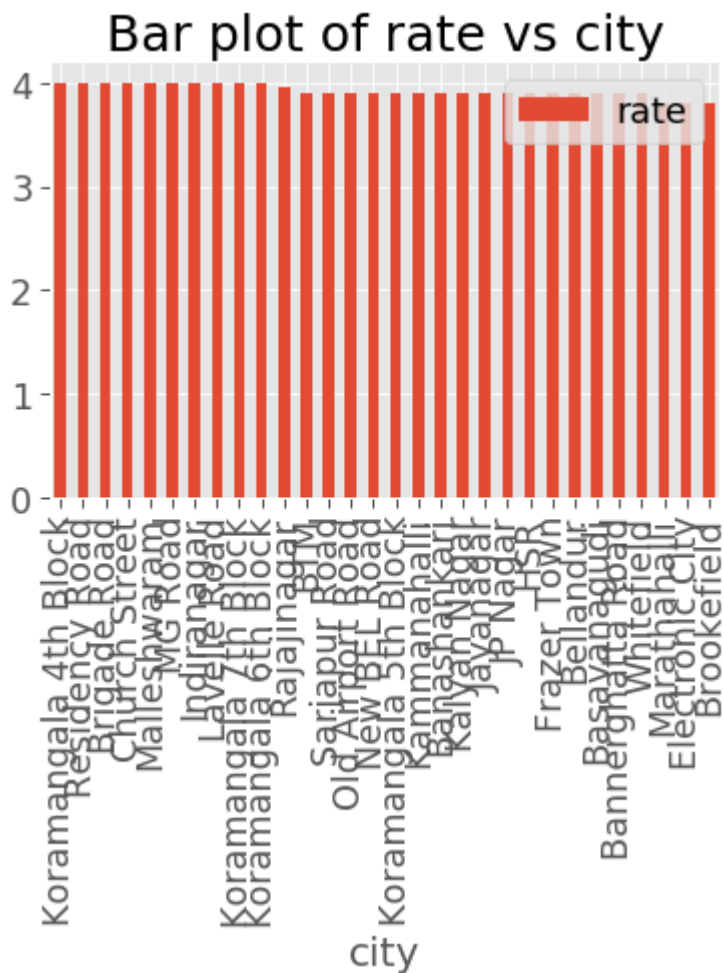
```
In [72]: plt.figure(figsize=(20,10))
df[['approx_cost(for two people)', 'city']].groupby(['city']).median().sort_values()
plt.title('Bar plot of average cost vs city')
plt.show()
```

<Figure size 1440x720 with 0 Axes>



```
In [73]: plt.figure(figsize=(20,10))
df[['rate', 'city']].groupby(['city']).median().sort_values("rate", ascending = True)
plt.title('Bar plot of rate vs city')
plt.show()
```

<Figure size 1440x720 with 0 Axes>



```

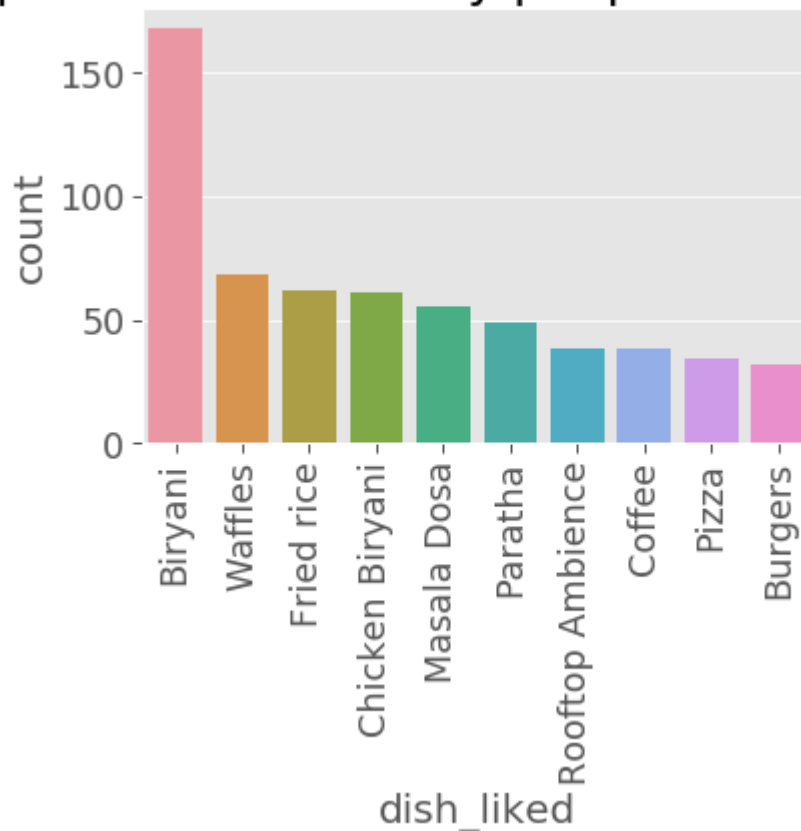
In [74]: a = df.iloc[:, :].values
for i in range(0, len(a)):
    if a[i, 6] == 'Fried rice':
        print(a[i, 6])
        a[i, 6] = None

a = pd.DataFrame(a)
a.columns = df.columns

sns.set_context("paper", font_scale = 2, rc = {"font.size": 20, "axes.titlesize":
b = sns.countplot(data = a, x = 'dish_liked', order = a.dish_liked.value_counts(
b.set_xticklabels(b.get_xticklabels(), rotation = 90)
plt.title('Top 10 dishes liked by people of Bangalore')
plt.show()

```

## Top 10 dishes liked by people of Bangalore



## Conclution

**\*We see that people at Bangalore love Biryani the most, then comes Chicken Biryani and then Fried rice**

In [ ]:

In [ ]:

```
In [1]: import pandas as pd
import numpy as np
```

```
In [2]: df = pd.read_excel('zom.xlsx')
```

```
In [3]: df1 = df.applymap(lambda s:s.lower() if type(s) == str else s)
```

```
In [4]: df2 = df1.drop_duplicates()
```

```
In [5]: df2.columns
```

```
Out[5]: Index(['Unnamed: 0', 'address', 'name', 'online_order', 'book_table', 'rate',
              'location', 'rest_type', 'dish_liked', 'cuisines',
              'approx_cost(for two people)', 'meal_type', 'city', 'votes'],
              dtype='object')
```

```
In [6]: df256=df.drop(['Unnamed: 0', 'address', 'name', 'online_order', 'book_table', 'lo
```

```
In [7]: df256
```

```
Out[7]:
```

	rate	approx_cost(for two people)
0	4.1/5	600
1	4.1/5	600
2	4.1/5	600
3	4.1/5	600
4	3.8/5	300
...	...	...
19228	4.1/5	1500
19229	4.0/5	1400
19230	4.0/5	1400
19231	4.2/5	300
19232	4.2/5	300

19233 rows × 2 columns

```
In [8]: from sklearn.datasets.samples_generator import make_blobs
from sklearn.cluster import KMeans
import matplotlib.pyplot as plt
```



```
In [9]: df256['rate'].value_counts()  
df256['rate'].isnull().sum()  
df256['rate'] = df['rate'].apply(lambda x: str(x).split('/')[0])  
df256['rate'].value_counts()
```

```
Out[9]: 3.9      3012  
4.0      2629  
4.1      2463  
3.8      2288  
4.2      1698  
3.7      1448  
4.3      1148  
3.6       762  
4.4       598  
3.5       424  
3.4       320  
2.8       286  
2.9       261  
4.5       259  
3.1       237  
3.0       236  
3.2       209  
3.3       190  
2.7       162  
NEW       135  
4.6       119  
2.6       112  
4.7        53  
2.4        44  
2.5        44  
2.3        33  
2.2        23  
2.0        11  
2.1        11  
4.8        10  
4.9         8  
Name: rate, dtype: int64
```

```

In [10]: g23 = list(df256['rate'])
for i in range(0, len(g23)):
    if g23[i] == 'NaN':
        g23[i] = 'unrated'
    elif g23[i] == '-':
        g23[i] = 'unrated'
    elif g23[i] == 'NEW':
        g23[i] = 'unrated'

df256['rate'] = g23

df256['rate'].value_counts()

g23 = list(df256['rate'])
for i in range(0, len(g23)):
    if g23[i] == 'unrated':
        g23[i] = None
    else :
        g23[i] = float(g23[i])
df256['rate'] = g23
df256['rate'].value_counts()

```

```

Out[10]: 3.9    3012
         4.0    2629
         4.1    2463
         3.8    2288
         4.2    1698
         3.7    1448
         4.3    1148
         3.6     762
         4.4     598
         3.5     424
         3.4     320
         2.8     286
         2.9     261
         4.5     259
         3.1     237
         3.0     236
         3.2     209
         3.3     190
         2.7     162
         4.6     119
         2.6     112
         4.7      53
         2.5      44
         2.4      44
         2.3      33
         2.2      23
         2.0      11
         2.1      11
         4.8      10
         4.9       8
         Name: rate, dtype: int64

```

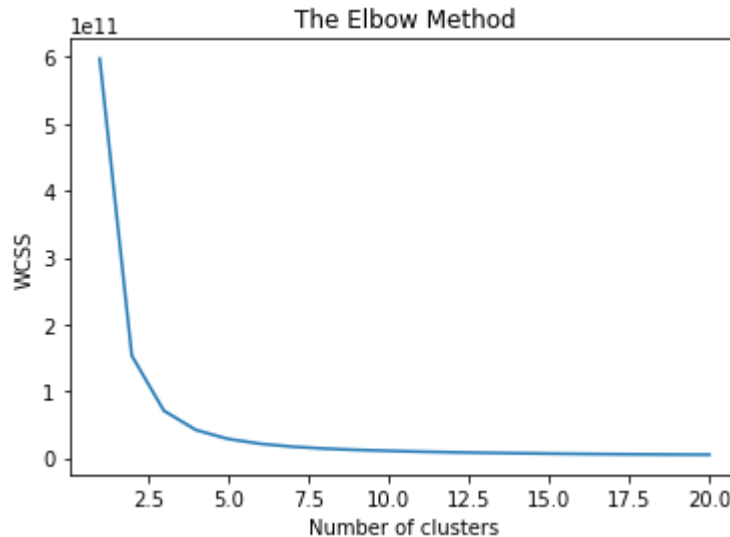
```

In [11]: df257=pd.read_csv('df257.csv')

```

# Elbow Method

```
In [12]: wcss = []
for i in range(1, 21):
    kmeans = KMeans(n_clusters = i, init = 'k-means++', random_state = 5)
    kmeans.fit(df257)
    wcss.append(kmeans.inertia_)
plt.plot(range(1, 21), wcss)
plt.title('The Elbow Method')
plt.xlabel('Number of clusters')
plt.ylabel('WCSS')
plt.show()
```



In [ ]:

```
In [13]: kmeans = KMeans(n_clusters=5, random_state=0).fit(df257)
```

```
In [14]: labels = kmeans.labels_
```

```
In [15]: df257['clusters'] = labels
```

```
In [16]: df257.columns
```

```
Out[16]: Index(['Unnamed: 0', 'rate', 'approx_cost(for two people)', 'clusters'], dtype='object')
```

## k-Means Clustering

```
In [17]: # Fitting K-Means to the dataset
kmeans = KMeans(n_clusters = 5, init = 'k-means++', random_state = 42)
y_kmeans = kmeans.fit_predict(df257)
y_kmeans
```

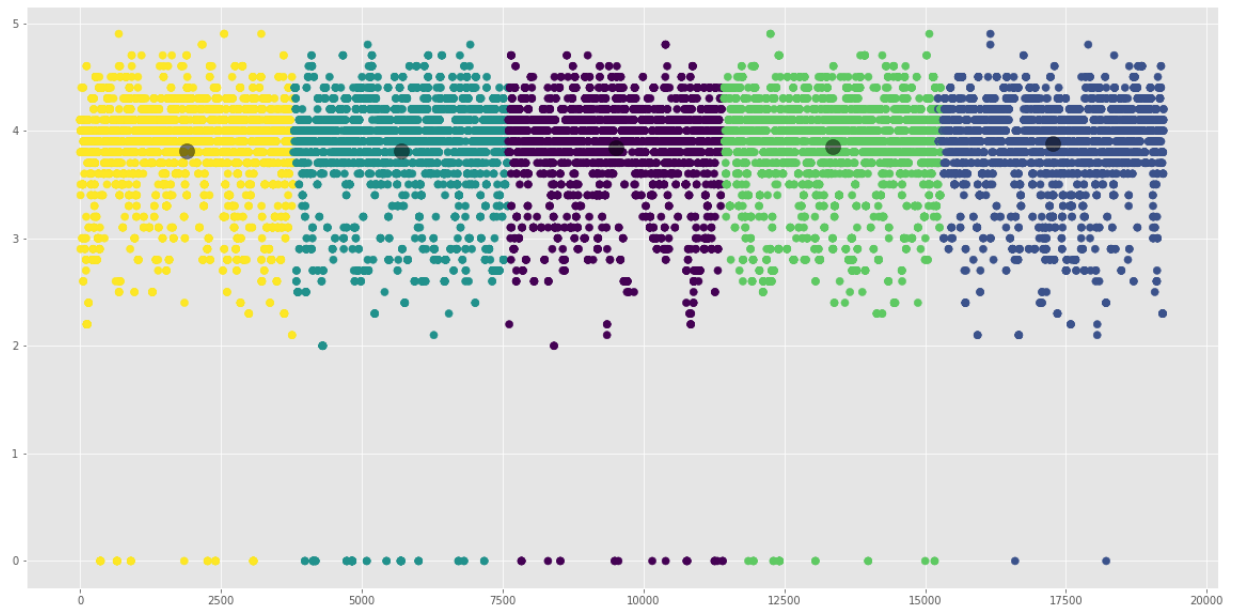
```
Out[17]: array([4, 4, 4, ..., 1, 1, 1])
```

```
In [18]: import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns
```

```
In [19]: import matplotlib.pyplot as plt
plt.style.use('ggplot')
```

```
In [20]: plt.figure(figsize=(20,10))
plt.scatter(df257.iloc[:, 0], df257.iloc[:, 1], c=y_kmeans, s=50, cmap='viridis')

centers = kmeans.cluster_centers_
plt.scatter(centers[:, 0], centers[:, 1], c='black', s=200, alpha=0.5);
```



```
In [ ]:
```

## Conclusion

After clustering i have concluded the restaurants by Price variance.

This shows that there some restaurants above 4.5 rating have price range between 2500 rs.

## SUMMARY OF CONCLUSION

\* Mostly Bangalore population takes online order. Hence, IT field has potential to develop online application for 27% of restaurants.

\*We see that 78%r of restaurants does not offer the service to book table, and very less do.This means that people at Bangalore prefer to eat at their homes or prefer fast

**food(snacks — quick bites)**

**\*We see that 48% of restaurants offer delivery, which might account for the above conclusion that people at Bangalore prefer to eat at home. Pubs and Bars and Buffet are offered by very less number of restaurants.**

**\*BTM has the highest number (approx 1200) of restaurants, followed by Koramangala 7th Block. New BEL Road has the least number of restaurants, followed by Banashankari. It seems that the main foodies live in BTM and Koramangala**

**\* The majority of restaurants are rated 3.9 (out of 5), followed by 3.8 and 3.7, which are decent ratings. This means that most of the restaurants at Bangalore are liked by the citizens and hence rated above average. This shows that due to high rise in the number of restaurants in Bangalore, the competition is at its best, as every restaurant is trying to excel in quality and services to increase the number of customers, and hence enhance profit**

**\*We see that, the most number of restaurants are of the type casual dining. From the above plot it is clear that people at Bangalore prefer 'quick bites' more often. This supports our above conclusion, which was drawn from the count plot of book\_table feature, that citizens of Bangalore prefer fast food-quick bites more often, this is why many restaurants do not offer table booking facility**

**\*We see that North Indian cuisine is offered by most number of restaurants. This is worth discussing, as Bangalore is situated in South India and yet people there prefer to eat North Indian cuisine**

**\*We see that Byg Brewski Brewing Company restaurant has the maximum number of upvotes. We see that this restaurant has outstanding rating, 4.9 out of 5.0. The restaurant is a Microbrewery type of restaurant, and offers the above mentioned cuisines. The dishes liked by the people are also mentioned. We see that the meal type for different branches of the restaurant include Delivery, Dine-out and Drinks & nightlife, this means that Byg Brewski Brewing Company restaurant is one of those strong competitors in these types of meals.**

**\*We see that BTM has the highest number of restaurants where you can or can not book table**

**\*We see that people at Bangalore love Biryani the most, then comes Chicken Biryani and then Fried rice**

**\*After clustering I have concluded the restaurants by Price variance. This shows that there are some restaurants above 4.5 rating have price range between 2500 rs**

In [ ]:

```
In [24]: import pandas as pd  
import numpy as np  
import matplotlib.pyplot as plt  
import seaborn as sns
```

```
In [26]: df = pd.read_csv('Documents\zomato og\dishuniquefull.csv')
```

```
In [27]: df.isnull().sum()
```

```
Out[27]: Unnamed: 0      0
         name          0
         rate          0
         location      0
         votes         0
         dish_liked    0
         approx_cost(for two people)  0
         aalo bhaja    0
         aam panna     0
         aamras        0
         abbabi chicken 0
         achari chicken 0
         achari mushroom 0
         achari paneer  0
         adrak ke panje 0
         afghan chicken 0
         afghani biryani 0
         afghani chaap  0
         afghani kebab  0
         afghani momos  0
         aglio olio pasta 0
         ajwaini paratha 0
         akki rottis    0
         al faham chicken 0
         alfam chicken  0
         almond crunch  0
         almond flakes  0
         almond fudge   0
         almond ice cream 0
         almond pie     0
         ..
         veg dum biryani 0
         veg jaipuri     0
         veg manchow soup 0
         veg momos       0
         veg platter     0
         veg puff        0
         veg pulao       0
         veg salad       0
         veg tandoori momos 0
         veg thali       0
         vegetable biryani 0
         vegetarian     0
         veggie delight  0
         veggie pizza    0
         veggie roll     0
         veggie wrap     0
         virgin colada   0
         waffle cone     0
         waffles         0
         wasabi prawns   0
         watermelon juice 0
         wedges          0
         wheat beer      0
         white pasta     0
```

```
whole meat shawarma      0
wine                     0
wings                    0
wonton                   0
wood fire pizza          0
zinger burger            0
Length: 3313, dtype: int64
```

In [28]:

df.describe()

Out[28]:

	Unnamed: 0	votes	approx_cost(for two people)	aalo bhaja	aam panna	aamras	at chi
count	4332.000000	4332.000000	4332.000000	4332.000000	4332.000000	4332.000000	4332.00
mean	4344.799631	439.021237	1510.350877	0.001154	0.002078	0.001616	0.00
std	2507.835705	861.524719	1902.643209	0.033958	0.045538	0.040170	0.01
min	1.000000	0.000000	80.000000	0.000000	0.000000	0.000000	0.00
25%	2200.500000	91.000000	500.000000	0.000000	0.000000	0.000000	0.00
50%	4318.000000	189.000000	900.000000	0.000000	0.000000	0.000000	0.00
75%	6487.250000	432.250000	1700.000000	0.000000	0.000000	0.000000	0.00
max	8760.000000	16345.000000	30600.000000	1.000000	1.000000	1.000000	1.00

8 rows × 3309 columns

In [29]:

df.info

Out[29]:

<bound method DataFrame.info of			Unnamed: 0
name	rate	\	
0	1		1947 3.9/5
1	2		1947 4.0/5
2	3		1947 4.0/5
3	8		1947 4.1/5
4	9		#1-81 cafe 3.9/5
5	10		#refuel 3.7/5
6	11		1000 b.c 3.2/5
7	12	100âfâfâ,âfâfâ,â,âfâfâfâ,â,âfâ,â,â,âfâfâ,âfâfâ...	3.7/5
8	13		1131 bar + kitchen 4.6/5
9	16		12th main - grand mercure 4.1/5
10	18		1441 pizzeria 4.1/5
11	23		1522 - the pub 4.2/5
12	25		1522 - the pub 4.2/5
13	26		1522 - the pub 4.3/5
14	29		154 breakfast club 4.0/5
15	32		1722 urban bistro 4.1/5
16	33		1980s games cafe 3.4/5
17	34		1980s games cafe 3.4/5



In [7]: df.columns

Out[7]: Index(['Unnamed: 0', 'name', 'rate', 'location', 'votes', 'dish\_liked',  
'approx\_cost(for two people)', 'aalo bhaja', 'aam panna', 'aamras',  
...,  
'watermelon juice', 'wedges', 'wheat beer', 'white pasta',  
'whole meat shawarma', 'wine', 'wings', 'wonton', 'wood fire pizza',  
'zinger burger'],  
dtype='object', length=3313)

In [30]: df.shape

Out[30]: (4332, 3313)

In [31]: df.count

Out[31]: <bound method DataFrame.count of Unnamed: 0

	name	rate	\		
0		1		1947	3.9/5
1		2		1947	4.0/5
2		3		1947	4.0/5
3		8		1947	4.1/5
4		9		#l-81 cafe	3.9/5
5		10		#refuel	3.7/5
6		11		1000 b.c	3.2/5
7		12	100ãfâfã,âfâfâ,ã,âfâfâfã,â,âfâ,ã,â,âfâfã,âfâfâ...		3.7/5
8		13		1131 bar + kitchen	4.6/5
9		16		12th main - grand mercure	4.1/5
10		18		1441 pizzeria	4.1/5
11		23		1522 - the pub	4.2/5
12		25		1522 - the pub	4.2/5
13		26		1522 - the pub	4.3/5
14		29		154 breakfast club	4.0/5
15		32		1722 urban bistro	4.1/5
16		33		1980s games cafe	3.4/5
17		34		1980s games cafe	3.4/5

In [32]: df.dtypes

```
Out[32]: Unnamed: 0      int64
name      object
rate      object
location  object
votes     int64
dish_liked object
approx_cost(for two people) int64
aalo bhaja      int64
aam panna      int64
aamras         int64
abbabi chicken  int64
achari chicken  int64
achari mushroom int64
achari paneer   int64
adrak ke panje  int64
afghan chicken  int64
afghani biryani int64
afghani chaap   int64
afghani kebab   int64
afghani momos   int64
aglio olio pasta int64
ajwaini paratha int64
akki rottis     int64
al faham chicken int64
alfam chicken   int64
almond crunch   int64
almond flakes   int64
almond fudge    int64
almond ice cream int64
almond pie      int64
...
veg dum biryani int64
veg jaipuri     int64
veg manchow soup int64
veg momos       int64
veg platter     int64
veg puff        int64
veg pulao       int64
veg salad       int64
veg tandoori momos int64
veg thali       int64
vegetable biryani int64
vegetarian      int64
veggie delight  int64
veggie pizza    int64
veggie roll     int64
veggie wrap     int64
virgin colada   int64
waffle cone     int64
waffles         int64
wasabi prawns   int64
watermelon juice int64
wedges          int64
wheat beer      int64
white pasta     int64
```

```
whole meat shawarma    int64
wine                   int64
wings                  int64
wonton                 int64
wood fire pizza        int64
zinger burger          int64
Length: 3313, dtype: object
```

```
In [33]: df['rate'].value_counts()
df['rate'].isnull().sum()
df['rate'] = df['rate'].apply(lambda x: str(x).split('/')[0])
df['rate'].value_counts()
```

```
Out[33]: 3.9      635
4.0      583
3.8      509
4.1      472
3.7      353
4.2      329
4.3      234
3.6      209
4.4      138
3.5       96
4.5       88
2.9       74
3.1       71
3.4       71
3.2       65
2.8       64
3.0       62
3.3       50
2.7       47
4.6       39
2.6       32
new       28
4.7       21
2.5       16
2.4       15
2.3        7
4.8        7
4.9        5
2.1        5
2.2        5
2.0         2
Name: rate, dtype: int64
```

```
In [34]: g23 = list(df['rate'])
for i in range(0, len(g23)):
    if g23[i] == 'nan':
        g23[i] = 'unrated'
    elif g23[i] == '-':
        g23[i] = 'unrated'
    elif g23[i] == 'new':
        g23[i] = 'unrated'

df['rate'] = g23

df['rate'].value_counts()

g23 = list(df['rate'])
for i in range(0, len(g23)):
    if g23[i] == 'unrated':
        g23[i] = None
    else :
        g23[i] = float(g23[i])
df['rate'] = g23
df['rate'].value_counts()
```

```
Out[34]: 3.9    635
         4.0    583
         3.8    509
         4.1    472
         3.7    353
         4.2    329
         4.3    234
         3.6    209
         4.4    138
         3.5     96
         4.5     88
         2.9     74
         3.1     71
         3.4     71
         3.2     65
         2.8     64
         3.0     62
         3.3     50
         2.7     47
         4.6     39
         2.6     32
         4.7     21
         2.5     16
         2.4     15
         4.8      7
         2.3      7
         2.2      5
         4.9      5
         2.1      5
         2.0      2
         Name: rate, dtype: int64
```

In [35]:

```
df['log_votes'] = np.log(df['votes'])
print(df)
```

	Unnamed: 0		name	rate	\
0	1		1947	3.9	
1	2		1947	4.0	
2	3		1947	4.0	
3	8		1947	4.1	
4	9		#1-81 cafe	3.9	
5	10		#refuel	3.7	
6	11		1000 b.c	3.2	
7	12	100ãfâfã,âfâfã,ã,âfâfâfã,â,âfâ,ã,â,âfâfã,âfâfã...		3.7	
8	13		1131 bar + kitchen	4.6	
9	16		12th main - grand mercure	4.1	
10	18		1441 pizzeria	4.1	
11	23		1522 - the pub	4.2	
12	25		1522 - the pub	4.2	
13	26		1522 - the pub	4.3	
14	29		154 breakfast club	4.0	
15	32		1722 urban bistro	4.1	
16	33		1980s games cafe	3.4	
17	34		1q1	4.3	

In [37]: dataset= pd.read\_csv('Documents\zomato og\df1.csv')

C:\Users\SPR\Anaconda3\lib\site-packages\IPython\core\interactiveshell.py:3057: DtypeWarning: Columns (3316) have mixed types. Specify dtype option on import or set low\_memory=False.  
 interactivity=interactivity, compiler=compiler, result=result)

In [38]: df24=dataset.sort\_values(['vote\_rate'],ascending=False, inplace=False)

In [39]: df25=df24.head(10)

In [41]: `print(df25)`

	Unnamed: 0	Unnamed: 0.1	name	rate	\
662	662	1358	byg brewski brewing company	4.9	
4050	4050	8168	toit	4.7	
4070	4070	8218	truffles	4.7	
3807	3807	7642	the black pearl	4.7	
75	75	142	ab's - absolute barbecues	4.9	
499	499	1011	big pitcher	4.7	
3808	3808	7643	the black pearl	4.8	
386	386	792	barbeque nation	4.7	
1422	1422	2852	flechazo	4.7	
3676	3676	7354	tbc sky lounge	4.7	

	location	votes	vote_rate	\
662	sarjapur road	16345	47.538219	
4050	indiranagar	14956	45.180479	
4070	koramangala 5th block	14704	45.100612	
3807	koramangala 5th block	10471	43.504915	
75	btm	6490	43.012287	
499	old airport road	9041	42.814768	
3808	marathahalli	7023	42.513340	
386	indiranagar	7261	41.784282	
1422	marathahalli	7154	41.714507	
3676	hsr	6745	41.437817	

	dish_liked	\
662	cocktails, dahi kebab, rajma chawal, butter ch...	
4050	beer, pesto pizza, nachos, cocktails, beef las...	
4070	burgers, pasta, cocktails, american cheese bur...	
3807	chocolate lollipop, chocolate biscuit, fire sh...	
75	tangdi chicken, bbq buffet, chocolate icecream...	
499	beer, cocktails, tiramisu, tawa chicken, mockt...	
3808	dahipuri, jal-jeera, chicken grill, mutton see...	
386	mutton curry, tender coconut payasam, mutton s...	
1422	pizza, lauki halwa, keema pav, mutton kebab, c...	
3676	kulcha, cocktails, peri peri chicken, masala p...	

	approx_cost(for two people)	aalo bhaja	...	wheat beer	white pasta	\
662	1600	0	...	0	0	
4050	1500	0	...	0	0	
4070	900	0	...	0	0	
3807	1400	0	...	0	0	
75	1600	0	...	0	0	
499	3600	0	...	0	0	
3808	1500	0	...	0	0	
386	1600	0	...	0	0	
1422	1400	0	...	0	0	
3676	2000	0	...	0	0	

	whole meat shawarma	wine	wings	wonton	wood fire pizza	\
662	0	0	0	0	0	
4050	0	0	0	0	0	
4070	0	0	0	0	0	
3807	0	0	0	0	0	
75	0	0	0	0	0	
499	0	0	0	0	0	

3808	0	0	0	0	0
386	0	0	0	0	0
1422	0	0	0	0	0
3676	0	0	0	0	0

	zinger burger	Rank	log_votes
662	0	4332.0	9.701677319
4050	0	4331.0	9.612867836
4070	0	4330.0	9.595874845
3807	0	4329.0	9.25636481
75	0	4313.0	8.77801781
499	0	4327.0	9.109525067
3808	0	4317.0	8.856945756
386	0	4321.0	8.890272839
1422	0	4319.0	8.87542692
3676	0	4314.0	8.816556769

[10 rows x 3317 columns]

```
In [44]: import seaborn as sn
df25.head()
```

Out[44]:

	Unnamed: 0	Unnamed: 0.1	name	rate	location	votes	vote_rate	dish_liked	approx_co two pe
662	662	1358	byg brewski brewing company	4.9	sarjapur road	16345	47.538219	cocktails, dahi kebab, rajma chawal, butter ch...	
4050	4050	8168	toit	4.7	indiranagar	14956	45.180479	beer, pesto pizza, nachos, cocktails, beef las...	
4070	4070	8218	truffles	4.7	koramangala 5th block	14704	45.100612	burgers, pasta, cocktails, american cheese bur...	
3807	3807	7642	the black pearl	4.7	koramangala 5th block	10471	43.504915	chocolate lollipop, chocolate biscuit, fire sh...	
75	75	142	ab's - absolute barbecues	4.9	btm	6490	43.012287	tangdi chicken, bbq buffet, chocolate icecream...	

5 rows x 3317 columns



In [48]:

In [ ]: