Report: Data Science Project

Name: Susanta Ghosh

Table of Contents

Assignment-1A

Problem statement

Given data (RestoInfo 3.csv) and perform exploratory data analysis(EDA)

Analysis

Lets see data first:

It is restaurant information data, data shape = 2069, 15,

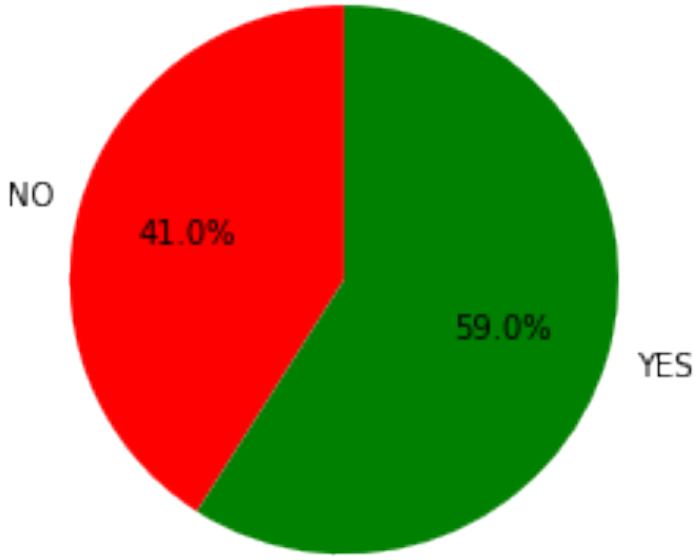
So 2069 restaurants are there, and total 15 columns for information's about those restaurants

Missing value:

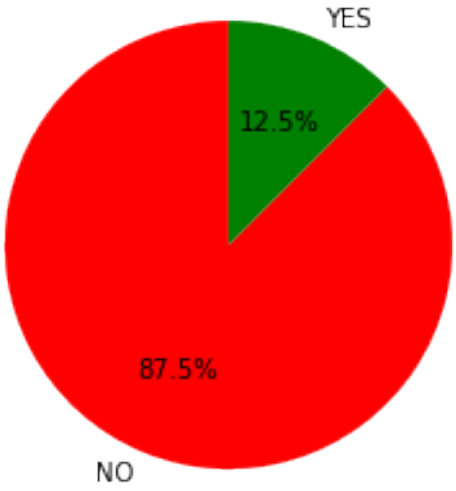
```
Null values % of each columns =  shop_id                        0.000000
name                             0.000000
online_order                     0.000000
book_table                       0.000000
rate                            14.451426
votes                            0.000000
location                         0.000000
rest_type                        0.821653
dish_liked                      53.504108
cuisines                         0.000000
approx_cost(for two people)      0.773320
reviews_list                     0.000000
menu_item                        0.000000
listed_in(type)                  0.000000
listed_in(city)                  0.000000
```

So as you see dish_liked column has around 53% null value, rate also 14%

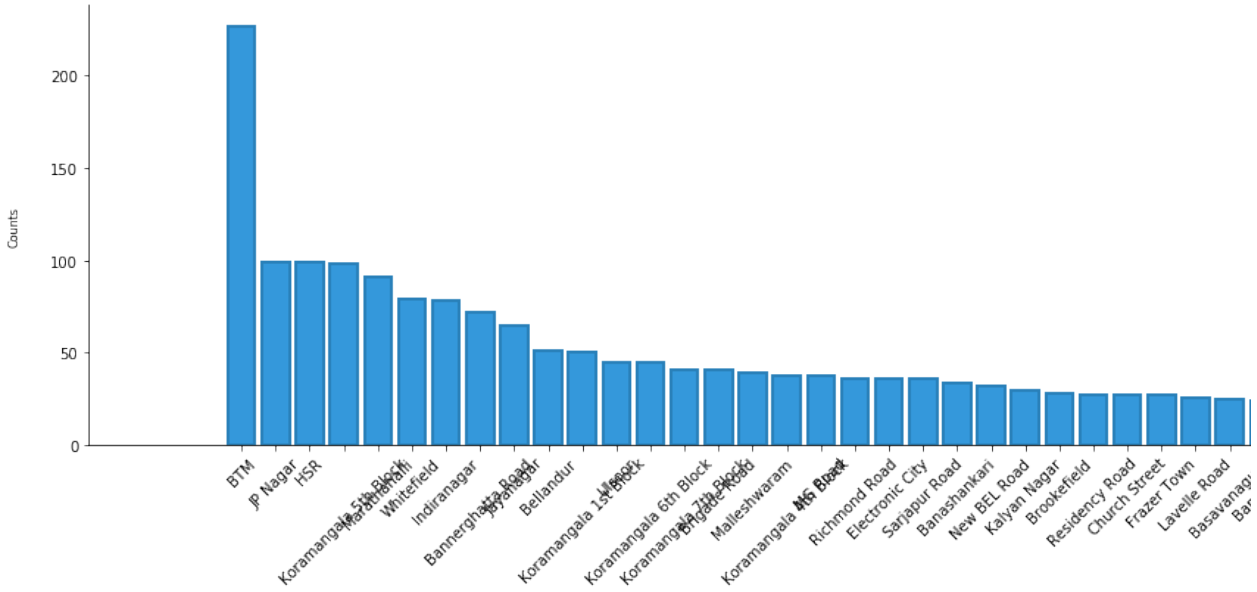
Q.1. How many online order you can see below pie chart ?



Q.2. How many book_table you can see below pie chart?

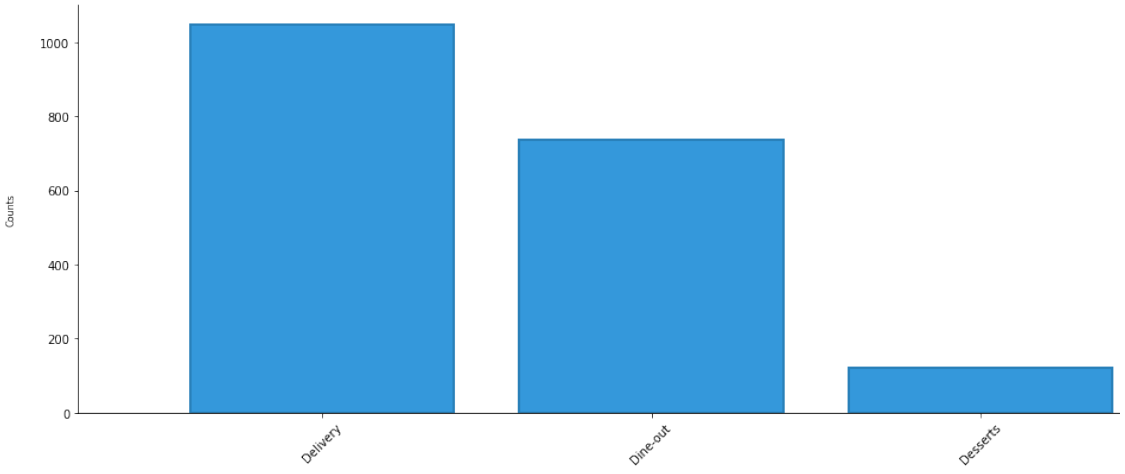


Q.3. Where maximum number of restaurants are located?



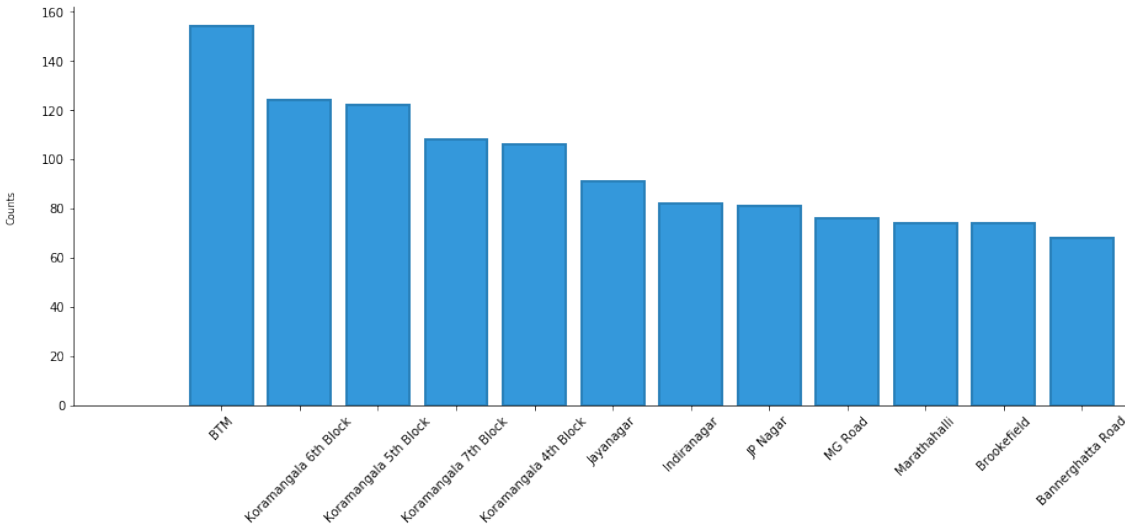
- **As you can see from above graph, maximum number of restaurant are located in BTM**

Q.4. Most of the restaurant are in whech listed type?



- **Ans: Delivery**

Q.5. Most popular listed in city?



Lets explore rest_type column:

There's no way to analyze it the way it currently is.

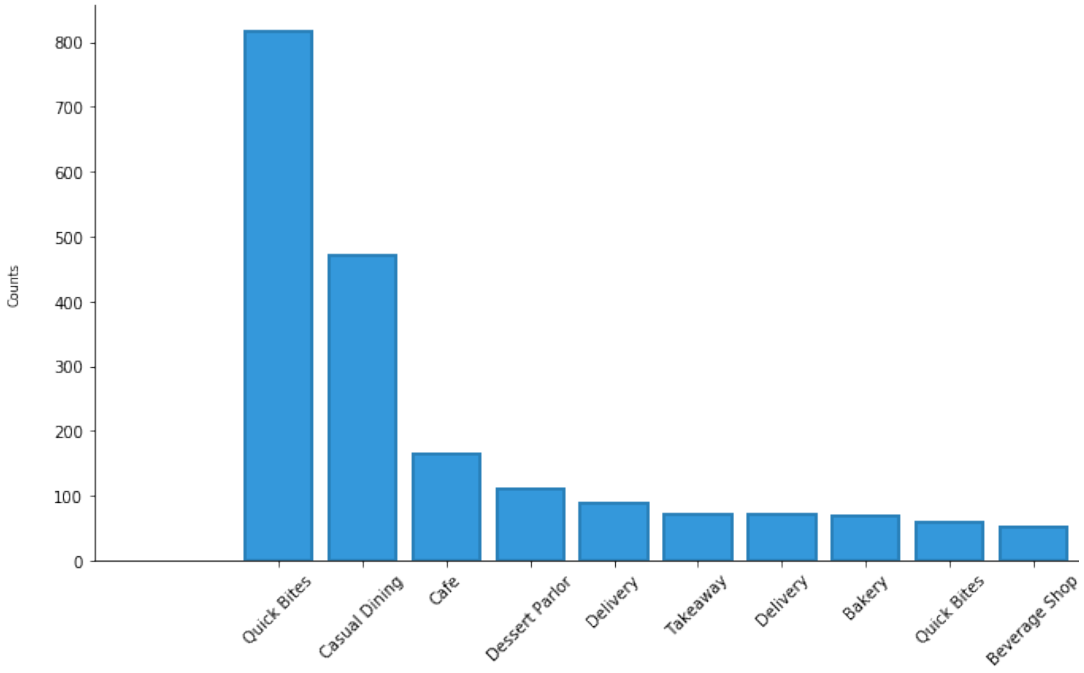
What I want to accomplish is the following:

-- Split the string on the pipe (,) character

-- Create a new entry for each rest_type

So, 1 row of 'Beverage Shop, Quick Bites' should become 2 rows, with other information remaining the same.

Q.6. Most of the restaurant are in which rest_type?



- **Ans: Quick Bites, and Casual Dining**

dish_liked column exploration:

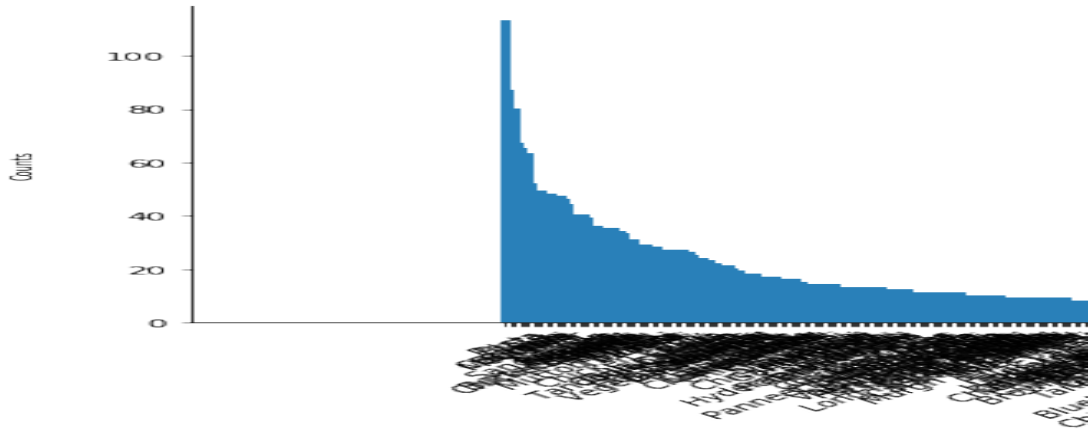
There’s no way to analyze it the way it currently is.

What I want to accomplish is the following:

-- Split the string on the pipe (,) character

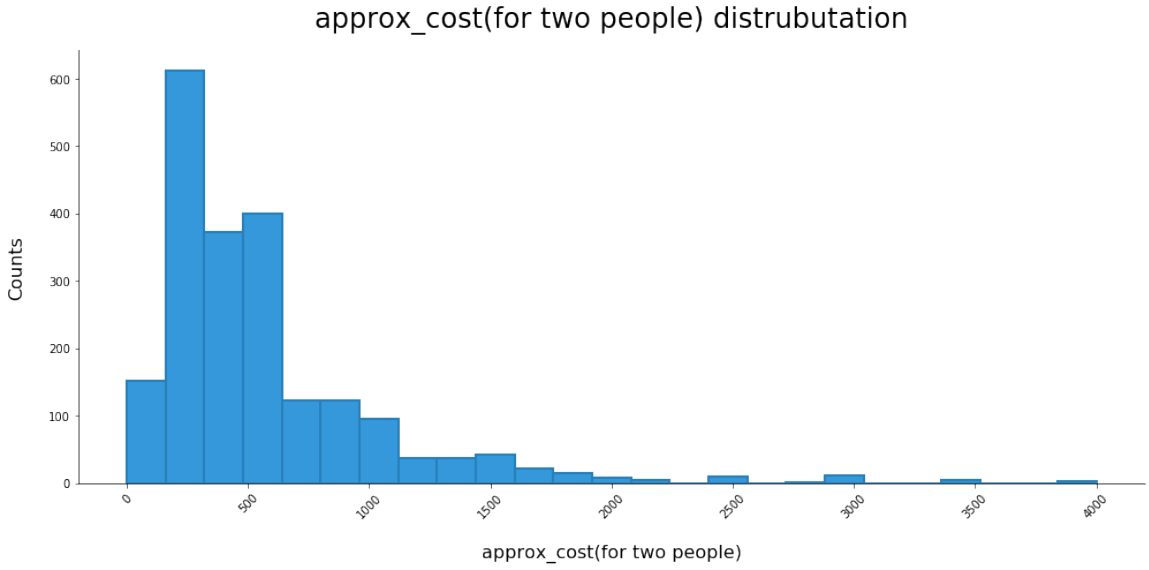
-- Create a new entry for each dish_liked

Q.7. Most popular dishs?



approx_cost(for two people) column exploration:

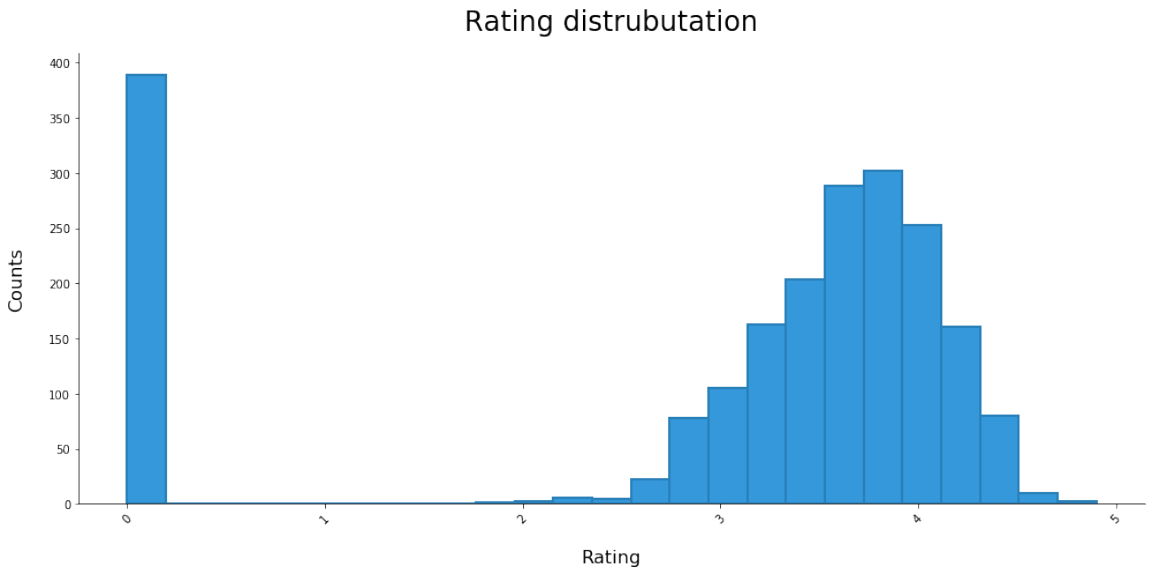
- **Histogram plot**



 # Most of the restaurant cost for two people less than 500.

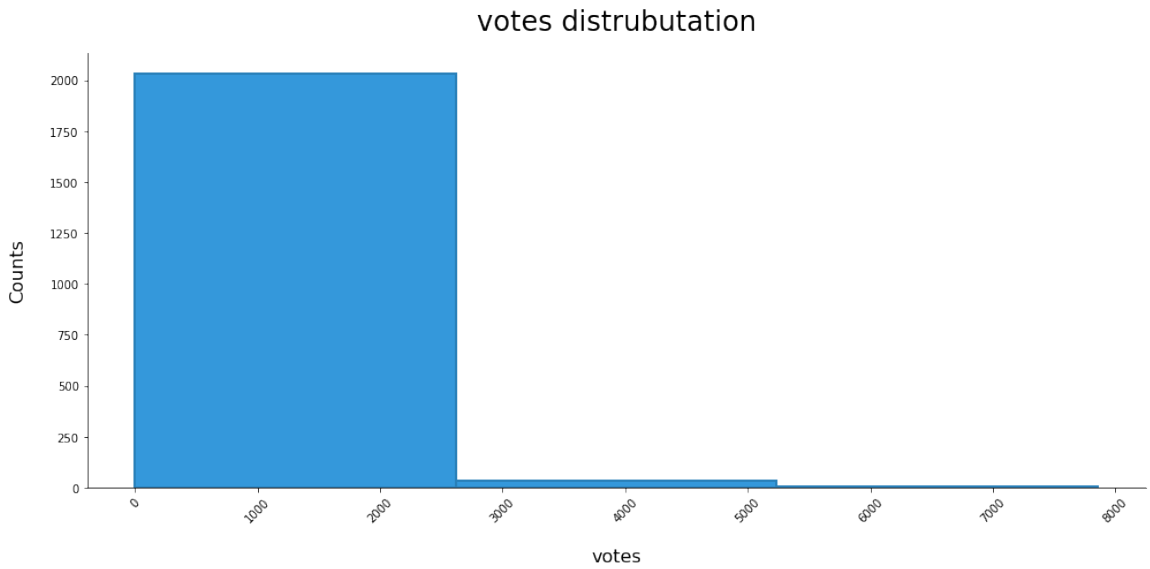
Rating And Votes exploration:

- **Histogram plot for rating**



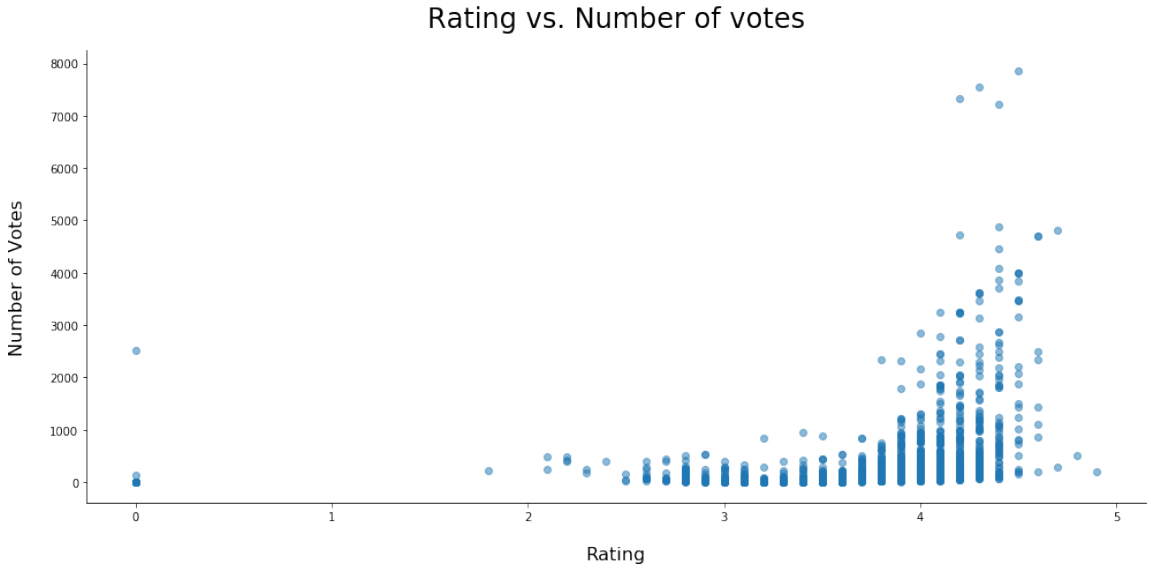
- As rating column has around 17% missing value, so I impute 0 for null value.
- You can see most of the restaurant rating in between 3-4

- **Histogram plot for Votes**



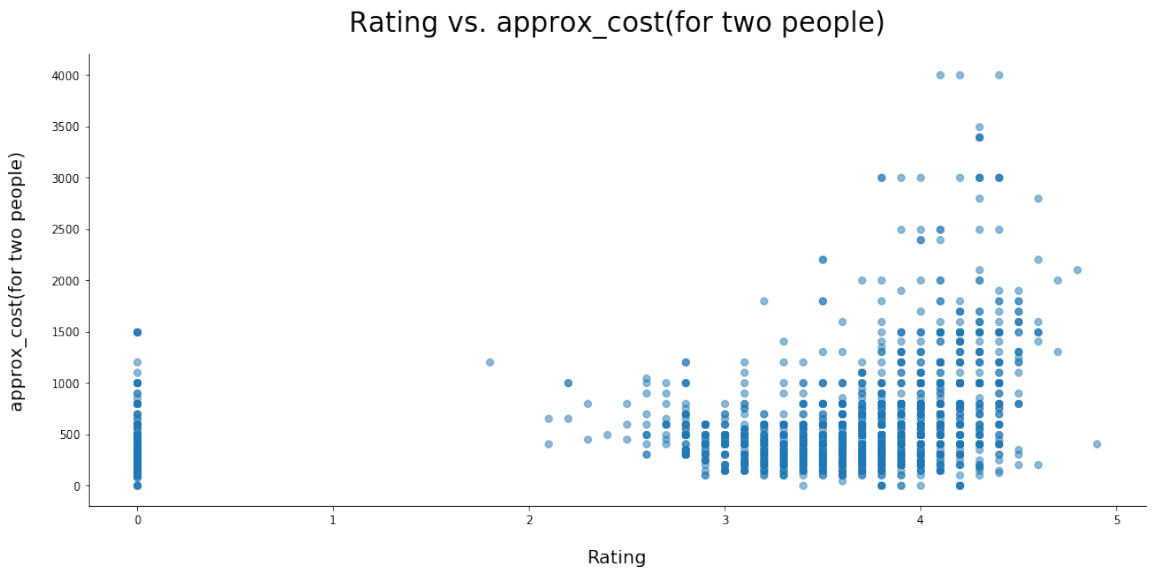
- Most of the restaurant get number of votes in between **0-2000**

- **Scatter plot: Rating vs. Number of votes**



- You can see a trend here — as a restaurant gets more votes it's ratings tends to increase.
- This also makes perfect sense if you think about it. If more and more people are voting a particular restaurant, it probably has a good budget and good marketing, which would mean that it's a good of some sort, and they are generally highly rated.

Scatter plot: Rating vs. approx_cost(for two people)



==========EDA part end here==========

==========Lets go to recommendation part==========

Assignment - 2

Problem statement :

Given restaurant information with some query(location, cuisine, budget, free search text). Goal is to recommend top 3 restaurant based on the query

Example query:

```
location = 'Koramangala'
cost = 500
cuisine = 'North Indian'
free_text = 'good ambiance restaurants, serving fish'
```

Data - RestoInfo 3.csv

Data details:

```
name
online_order
book_table
rate
votes
location
rest_type
dish_liked
cuisines
approx_cost(for two people)
reviews_list
menu_item
listed_in(type)
listed_in(city)
```

Algorithm:

Step – 1: Basic preprocessing

- Rating (3.0/5) convert to ➔ 3.5
- Votes (str(12,00)) convert to → float(1200)

Step – 2: Input Query

- Take input query query(location, cuisine, budget, free search text)

Step – 3: Search in restaurant info data and find how many matches

- So for given query iterate through all the restaurant one by one and collect below info

1. **Location(loc): (0/1)** if location in query and the restaurant is same, location = 1 else 0
2. **Cost(cos): (0/1)** if budget in query is less or equel to restaurant(approx_cost(for two people)), cost = 1 else 0
3. **Cuisines(cosin): (0/1)** if cuisines in query is also present in restaurant cuisines list, cuisines = 1 else 0
4. **free text score(f_score):**
    - split free text by ' ', and generate token list
    - Then search how many token present in restaurant info following columns [dish_liked, reviews_list, menu_item, rest_type]
    - Filnal f_score = #token match/#token in token list
5. **normalized rating score(n_rating):**
    - n_rating = restaurant rating / 5
    - 5 because max it is possible for rating

6. normalized votes score(n_votes):

- n_votes = restaurant voats / max(votes)

So best on the above 6 value we will calculate final recommendation score for each restaurant in nest step

Step – 4: Final recommendation score

- Final score (t_score) = loc + cos + csin + f_score + n_rating + n_votes
- Based on t_score we will take to 3 restaurent and show to the customer.
- Below you can see sample recommendation list

shop_id	f_score	t_score	loc	cos	csin	n_rating	n_votes
32674	0.2	3.992850	1	1	1	0.76	0.032850
36921	0.6	3.921421	0	1	1	0.86	0.461421
33863	0.6	3.921039	0	1	1	0.86	0.461039

Thank You