REPORT ON ZOMATO RATE PREDICTION

1. Introduction

Zomato is an application to discover the best food and drinks in Beirut and different places. There are many restaurants in different places on it. They provide restaurant partners with industry-specific marketing tools which enable them to engage and acquire customers to grow their business while also providing a reliable and efficient last mile delivery service. We also operate a one-stop procurement solution, which supplies high quality ingredients and kitchen products to restaurant partners. They also provide our delivery partners with transparent and flexible earning opportunities.

Business Problems We are Trying to Solve

- Problems we generally face during servicing customers:
    - Which restaurant is more suitable for me
    - How about services of the restaurant?
    - What is the rate of each restaurant?
    - What kind of cuisines they provide?

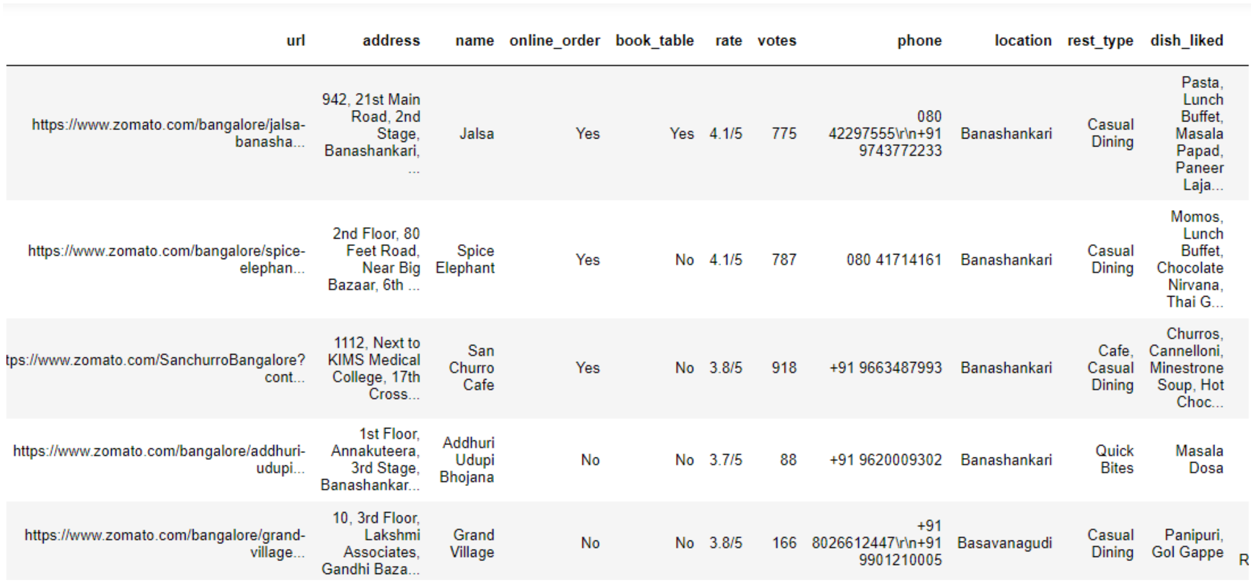
Main Goal

- Create an analytical framework to understand
    - Key factors impacting rate
- Develop a modelling framework
    - To estimate the rate of a restaurant

About Dataset

The data is taken from a popular app in different places. About many different restaurants in many places

This is what the dataset looks like,



url	address	name	online_order	book_table	rate	votes	phone	location	rest_type	dish_liked
https://www.zomato.com/bangalore/jalsa-banasha...	942, 21st Main Road, 2nd Stage, Banashankari, ...	Jalsa	Yes	Yes	4.1/5	775	080 42297555\r\n+91 9743772233	Banashankari	Casual Dining	Pasta, Lunch Buffet, Masala Papad, Paneer Laja...
https://www.zomato.com/bangalore/spice-elephan...	2nd Floor, 80 Feet Road, Near Big Bazaar, 6th ...	Spice Elephant	Yes	No	4.1/5	787	080 41714161	Banashankari	Casual Dining	Momos, Lunch Buffet, Chocolate Nirvana, Thai G...
tps://www.zomato.com/SanchurroBangalore?cont...	1112, Next to KIMS Medical College, 17th Cross...	San Churro Cafe	Yes	No	3.8/5	918	+91 9663487993	Banashankari	Cafe, Casual Dining	Churros, Cannelloni, Minestrone Soup, Hot Choc...
https://www.zomato.com/bangalore/addhuri-udupi...	1st Floor, Annakuteera, 3rd Stage, Banashankar...	Addhuri Udupi Bhojana	No	No	3.7/5	88	+91 9620009302	Banashankari	Quick Bites	Masala Dosa
https://www.zomato.com/bangalore/grand-village...	10, 3rd Floor, Lakshmi Associates, Gandhi Baza...	Grand Village	No	No	3.8/5	166	+91 8026612447\r\n+91 9901210005	Basavanagudi	Casual Dining	Panipuri, Gol Gappe

R

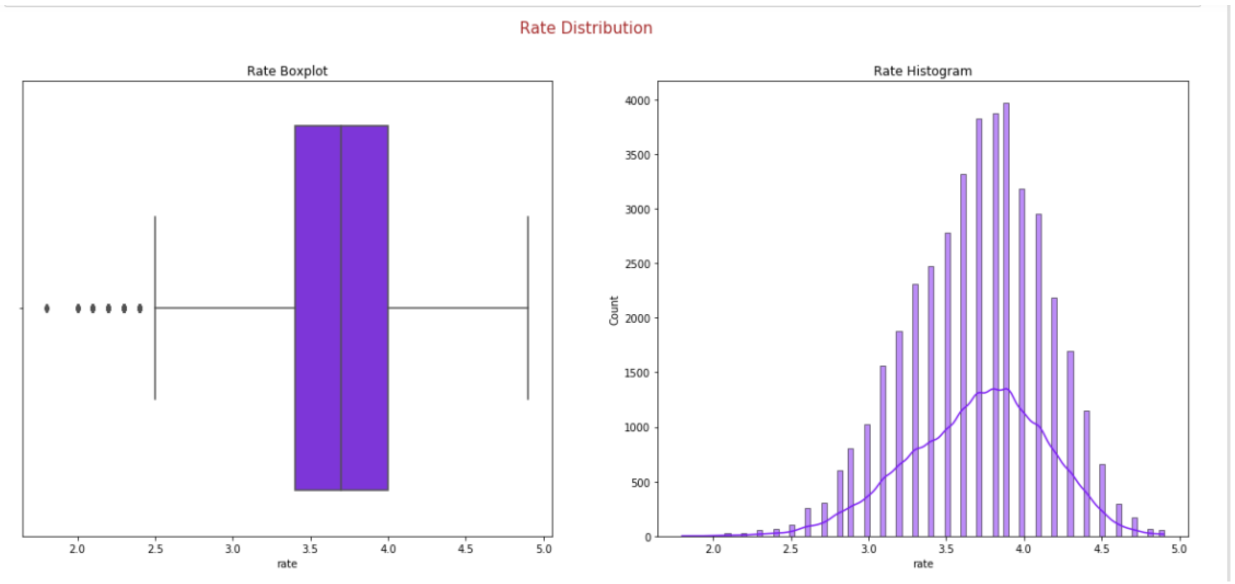
It has lots of rows and columns, which is 49509 rows spread across 17 columns.

2. EDA & Business Implication

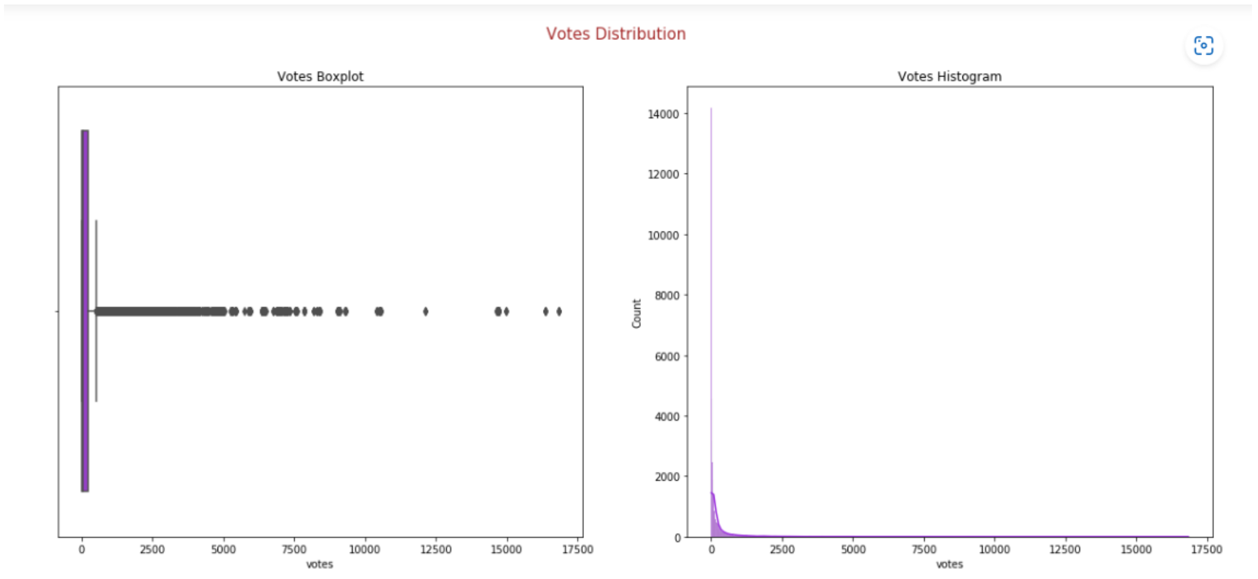
EDA stands for exploratory data analysis where we explore our data and grab insights from it. EDA helps us in getting knowledge in form of various plots and diagrams where we can easily understand the data and its features.

Analysis of Rate

Observation:

- We can see from the graph that the most restaurants have rates from range 3.5 to 4
- The maximum value in rate column is 4.9 and the minimum is 1.8
- 

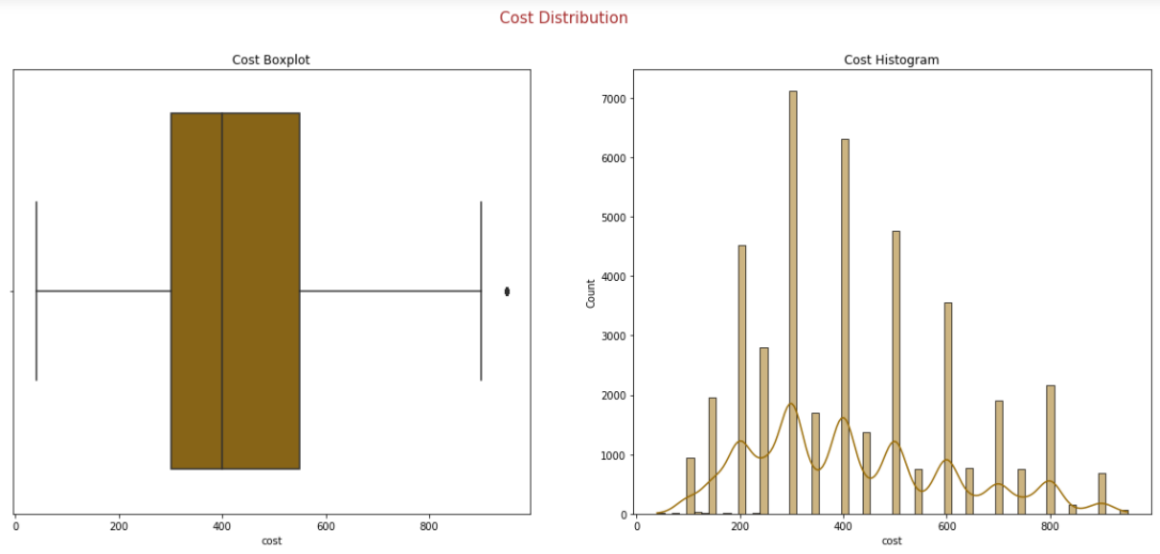
Analysis of Votes:



Observation:

- Votes have many outliers
- The average votes is 269

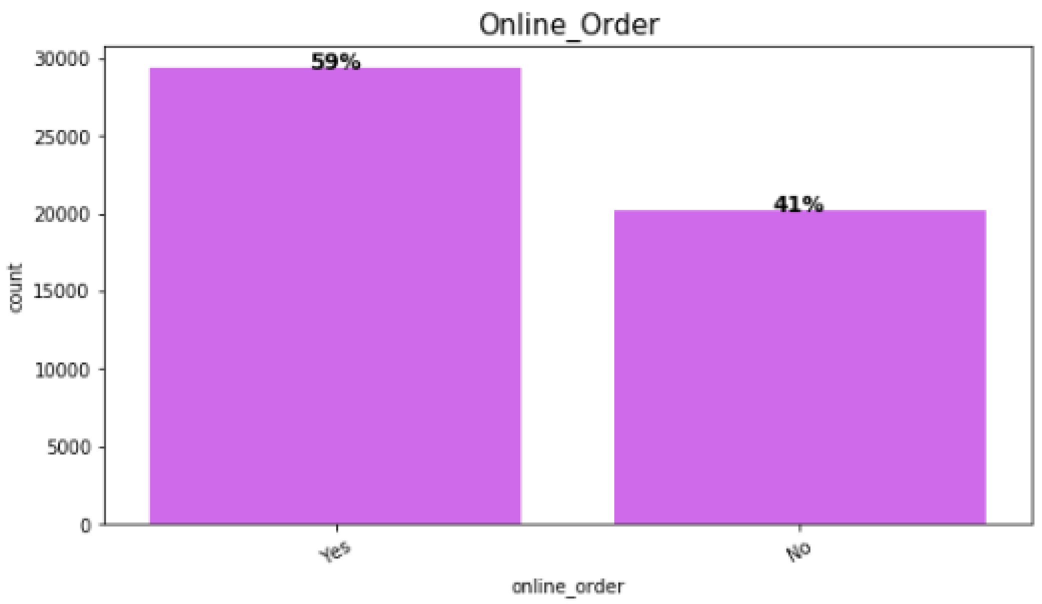
Analysis of Cost:



Observation:

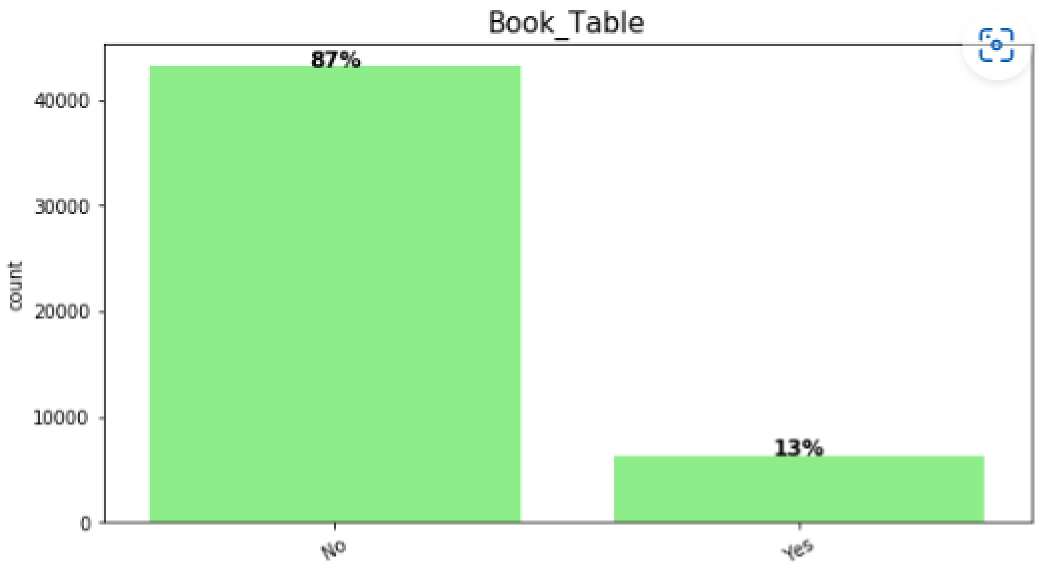
The average cost is 418 and the maximum is 95

Analysis of Online order service:



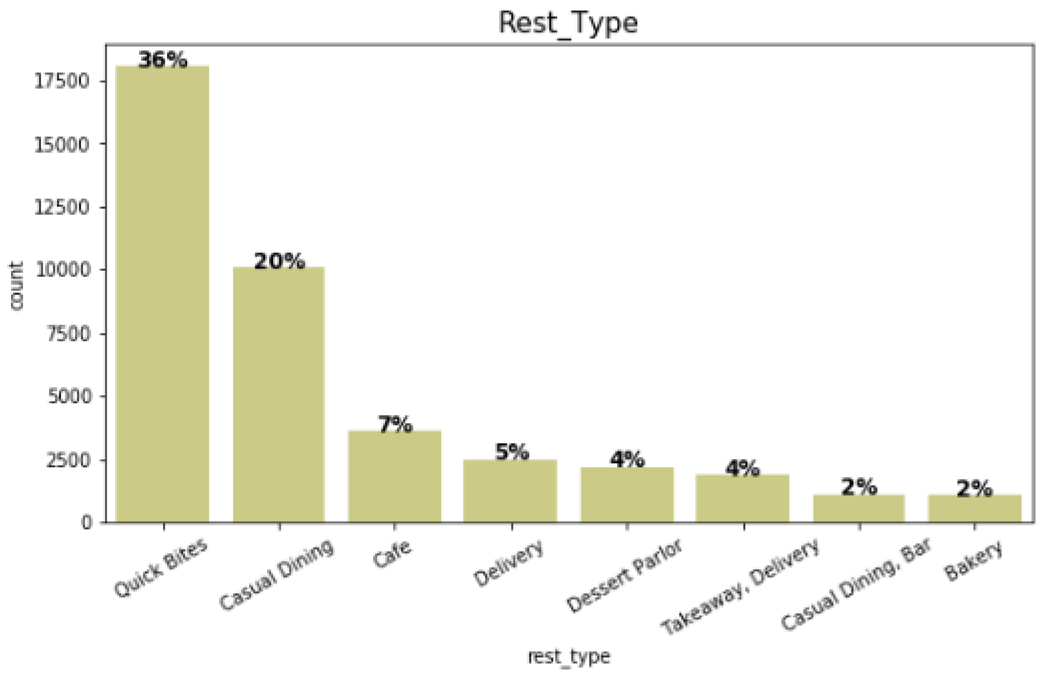
Observation: 59% of restaurants accept online order

Analysis of Body type:



Observation: 87% of restaurants refuse booking tables

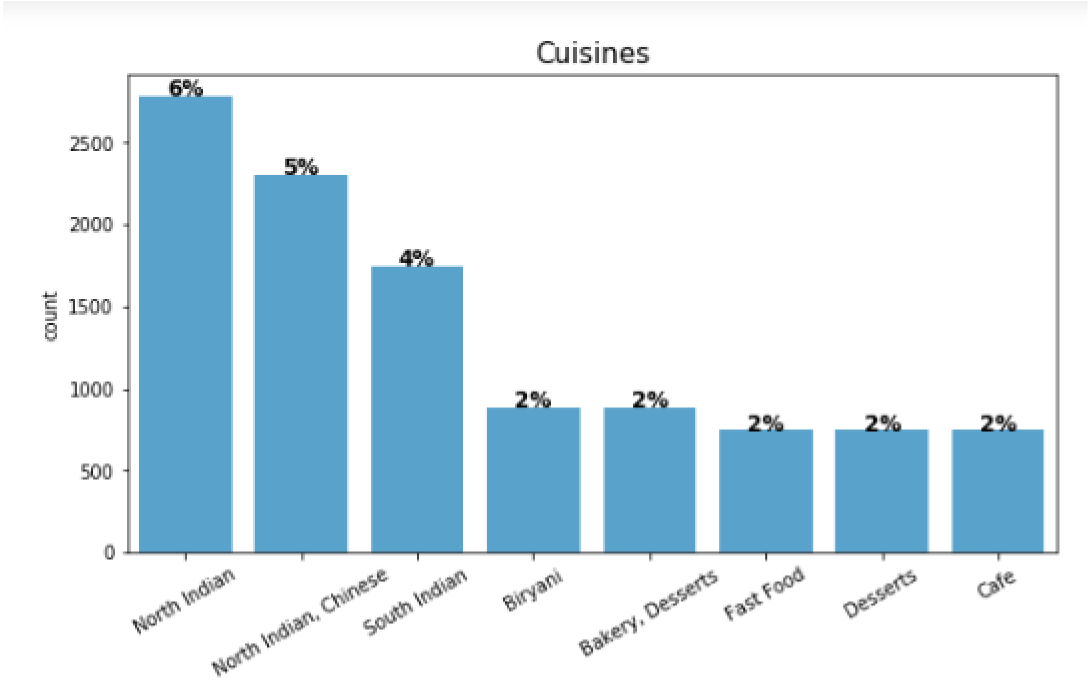
Analysis of Rest Type:



Observation:

Most restaurants type in our data are quick bites which has 36% from this data, and Casual Dining which has 20%

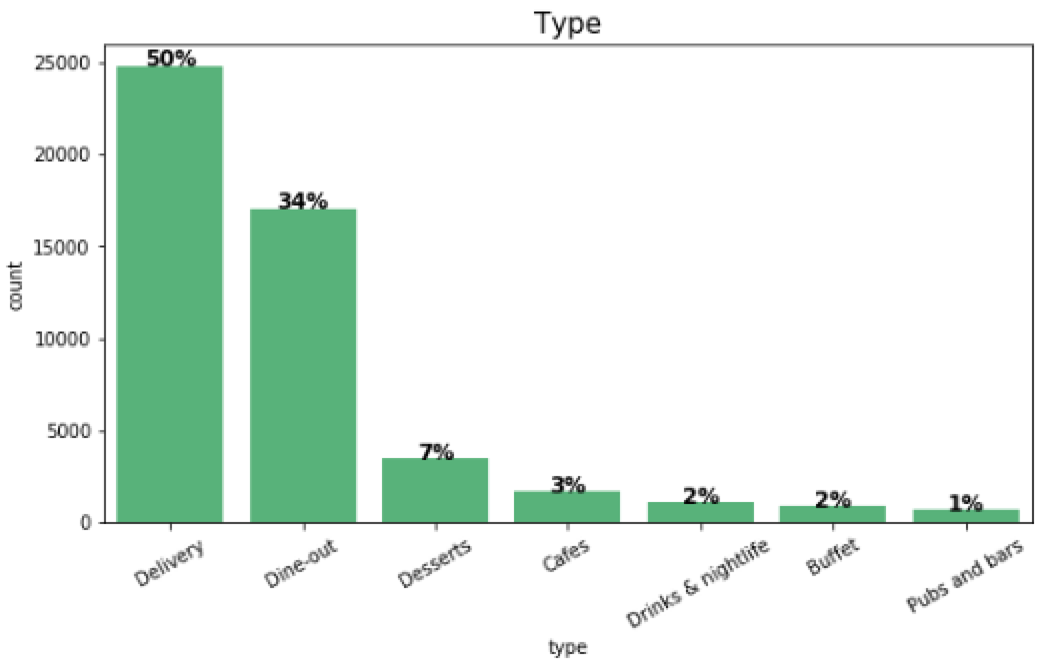
Analysis of Cuisines:



Observation:

Most cuisines popular served by restaurant are **North Indian, Chinese and South Indian**

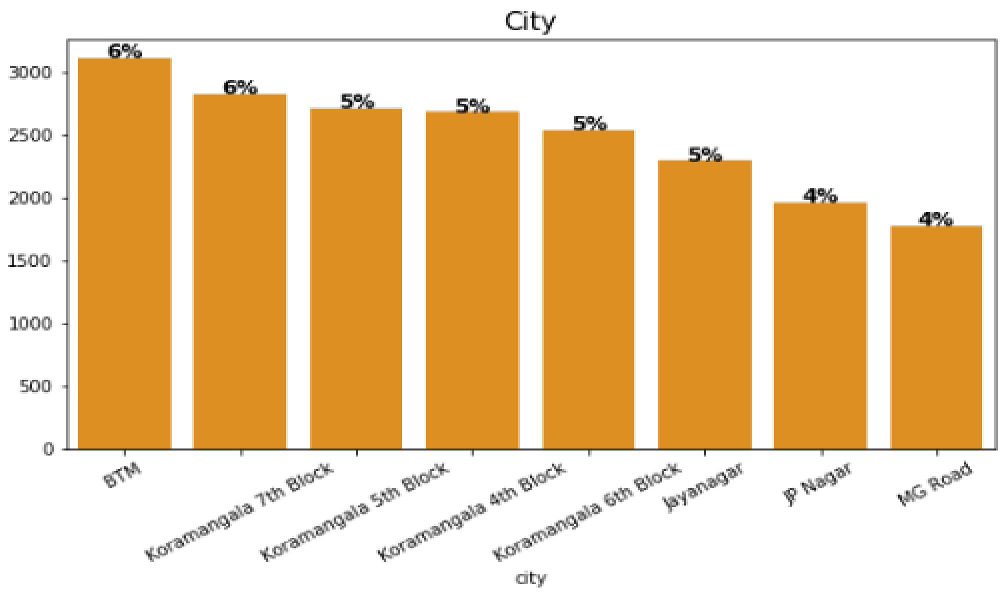
Analysis of Type:



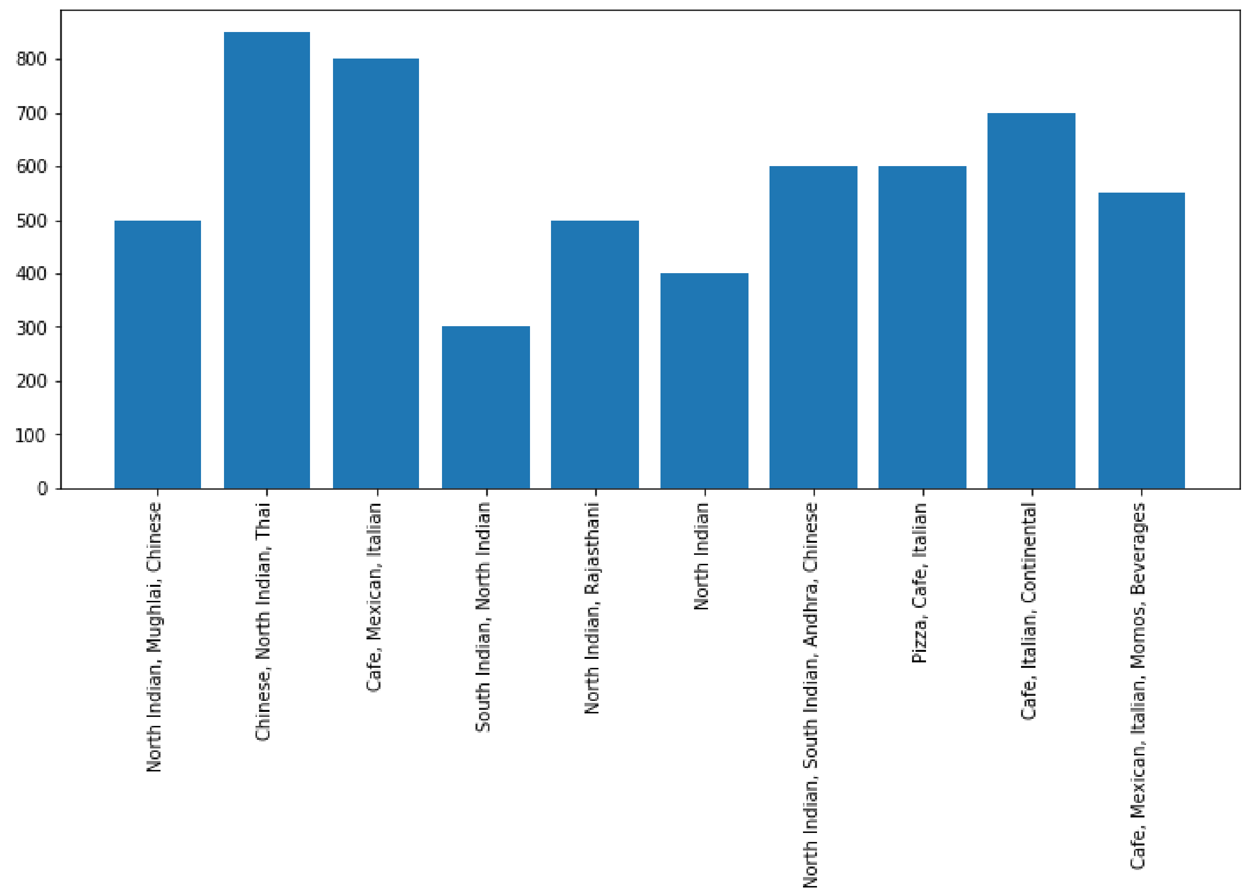
Observation:

Most popular categories of restaurants are **delivery which has 50% of frequency, and dine-out which has 34%**

Analysis of City:

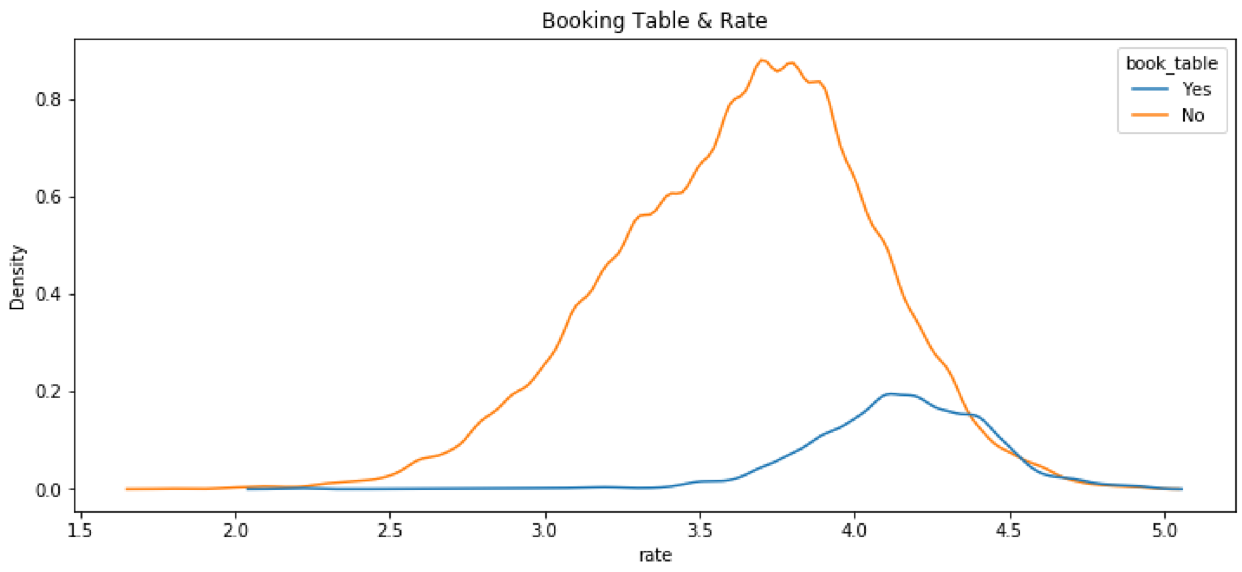


what is the relation between cost and cuisines differences?



we can observe that there is no rule for that because some restaurants serve different types of cuisines but there cost are different because there are other effects

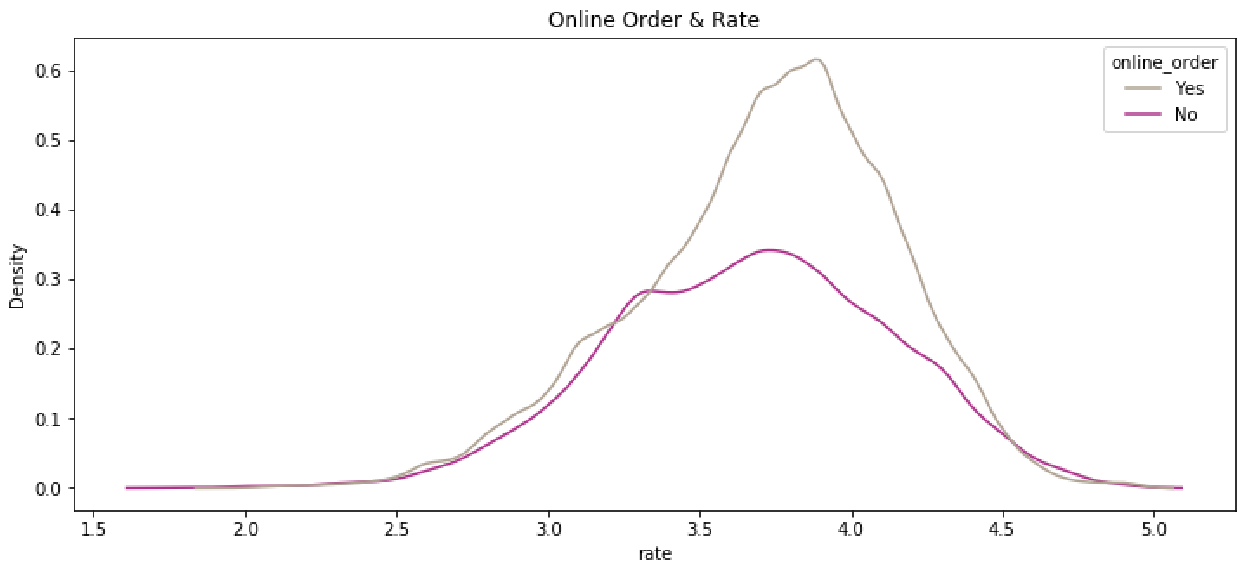
Is booking table service affecting on rate of the restaurant ?



Yes, the booking table service affects on the rate. We can observe the peak of no curve is in range 3.5 to 4 while the peak of yes curve is in range 4 to 4.5

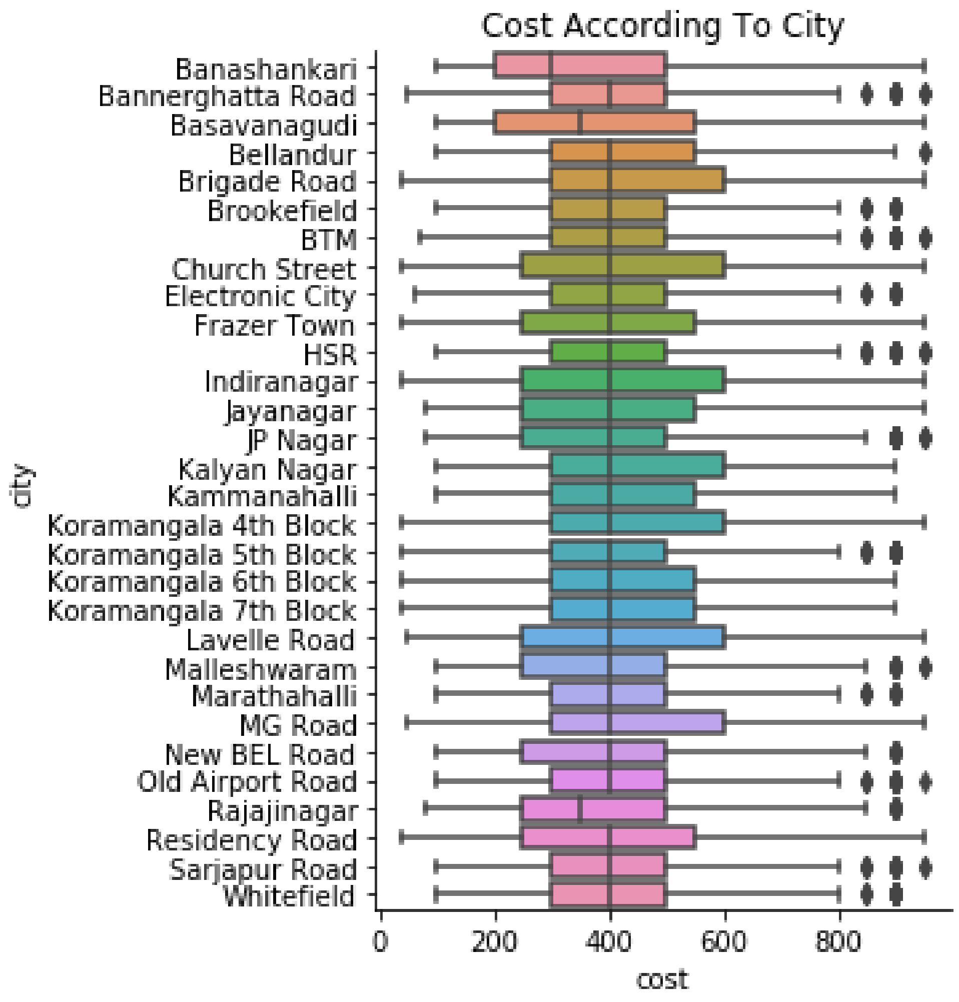
Advice: Add booking service will increase your rating

Relationship between online order and rating of the restaurant



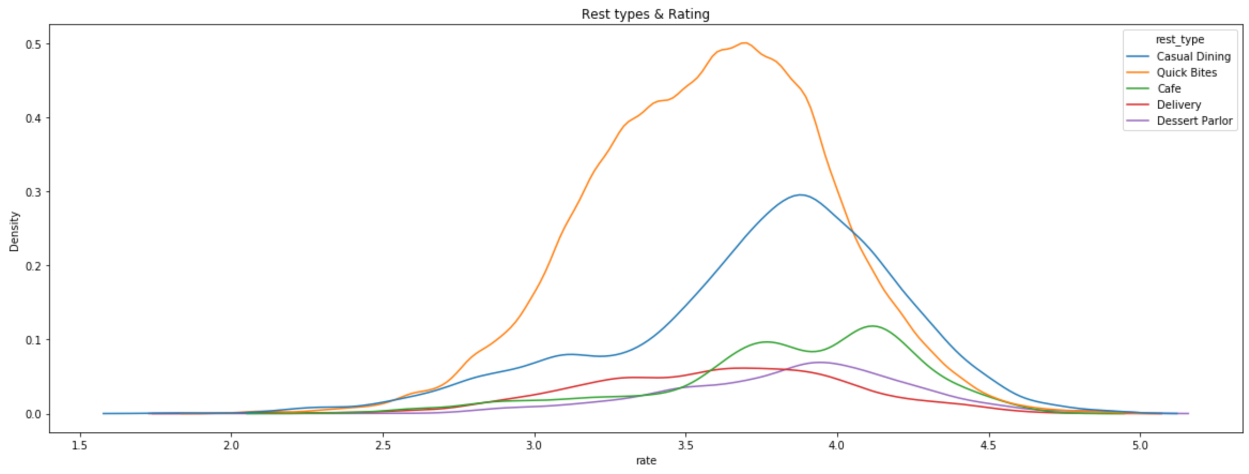
The distribution of two curves are **approximately** the same but the peak value is slight different. In restaurants which have online order service have more high rates

Is cost different from city to another?



There are approximately the same cost

Rest type and highest votes



The highest rate is for cafe.

The second highest are dessert palor and casual dining

3. Data Cleaning & Pre-processing

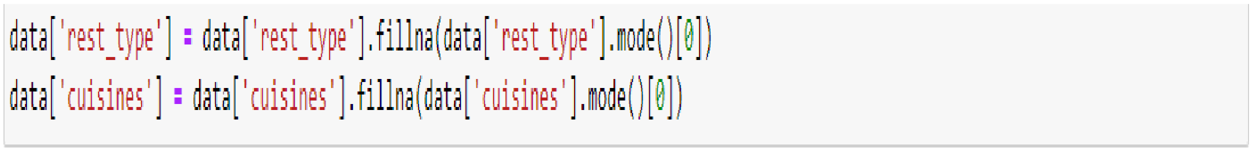
Data Cleaning is an important phase in any data science project, if our data is clean then only we can provide it to our machine learning model. Uncleaned Data can further lead our model with low accuracy. And, If data is incorrect, outcomes and algorithms are unreliable, even though they may look correct. There is no one absolute way to prescribe the exact steps in the data cleaning process because the processes will vary from dataset to dataset.

The approach used for identifying and treating missing values and outlier treatment:-

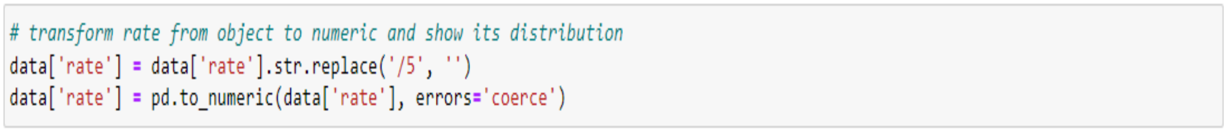
	Number of missing values	Percentage	Dtype
dish_liked	26017	52.55	object
rate	7775	15.70	object
phone	1194	2.41	object
approx_cost(for two people)	345	0.70	object
rest_type	225	0.45	object
cuisines	45	0.09	object
location	21	0.04	object

We have a lot of missing values in dish_liked column, so I dropped it. And I impute object columns with its mode and numeric columns with random values from the column values to make the distribution **approximately**. the same before and after imputing

The second problem is in the data type. Some columns have numeric values but in object format, so I transform these columns from object to numeric

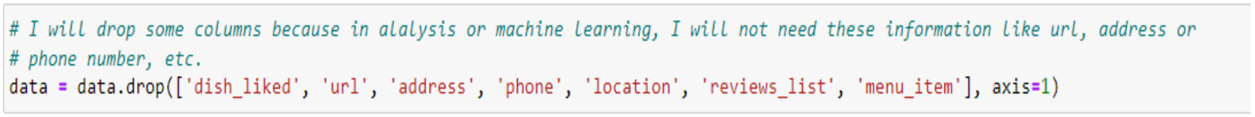


```
data['rest_type'] = data['rest_type'].fillna(data['rest_type'].mode()[0])
data['cuisines'] = data['cuisines'].fillna(data['cuisines'].mode()[0])
```



```
# transform rate from object to numeric and show its distribution
data['rate'] = data['rate'].str.replace('/5', '')
data['rate'] = pd.to_numeric(data['rate'], errors='coerce')
```

Drop unused columns?



```
# I will drop some columns because in alalysis or machine learning, I will not need these information like url, address or
# phone number, etc.
data = data.drop(['dish_liked', 'url', 'address', 'phone', 'location', 'reviews_list', 'menu_item'], axis=1)
```

['dish_liked', 'url', 'address', 'phone', 'location', 'reviews_list', 'menu_item'] these columns I dropped.

4. Modeling Building

Model Selection and Why?

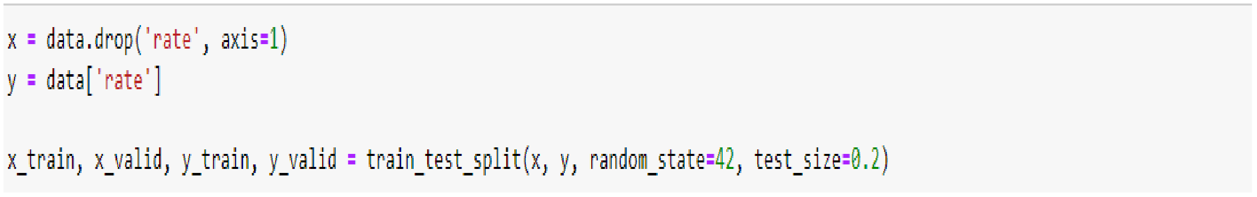
After cleaning and processing the data then comes the modeling part which includes building Machine Learning models, let's first understand in brief what Machine Learning is?

Machine Learning is a technique that analyzes past data and tries to extract meaningful insights and patterns from them which can be further used to perform predictions in future. For example, classifying whether a tumor is benign or malignant, predicting stock prices, etc. One such application which we're using right here is predicting house prices. Before making predictions first we need to build a model and train it using past data.

First, we need to separate the dataset into two parts: features (property attributes) and labels (prices) which is the required format for any model to be trained on.

Then the data needs to be split into 2 sets

1. Training set - This will be the part of the dataset which the model will be using to train itself, the size should be at least 80% of the total data we've.
2. Validation set - This set is used for validating our model's performance for a different set of hyperparameters. After taking out the train set, the remaining set can be split into validation and test set.

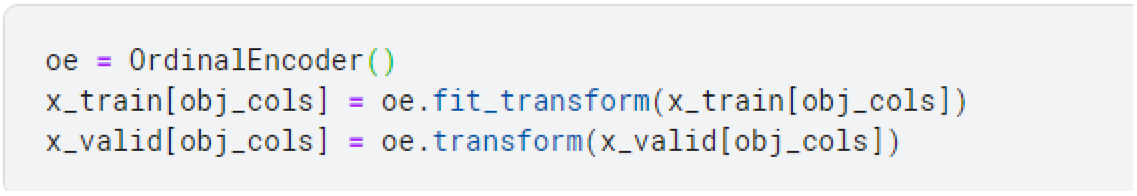


```
x = data.drop('rate', axis=1)
y = data['rate']

x_train, x_valid, y_train, y_valid = train_test_split(x, y, random_state=42, test_size=0.2)
```

We need to build different regression algorithms and using the validation set we can determine which model to keep for making final predictions.

Before we train we encode cat columns.



```
oe = OrdinalEncoder()
x_train[obj_cols] = oe.fit_transform(x_train[obj_cols])
x_valid[obj_cols] = oe.transform(x_valid[obj_cols])
```

Initially, we’ve tried **Linear Regression, RandomForestRegressor**.and **DecisionTreeRegressor**

We can see that the train and valid scores are 0.22 , 0.614 and 0.454 respectively.

Let’s look at some other algorithms, and how they are performing as compared to linear regression.

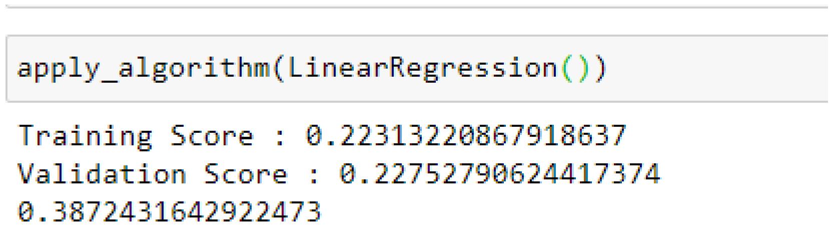
Performance Metrices

Just building is not enough, as we need to evaluate it using different metrics based on the problem we're solving. Model Validation helps us to understand how well the model is generalizing on the real-world data, the data which it has not seen during the training phase.

For regression problems the evaluation metrics we’ve used are:

- **RMSE** (Root Mean Squared Error)
- **MSE** (Mean Squared Error)
- **Adjusted R^2**

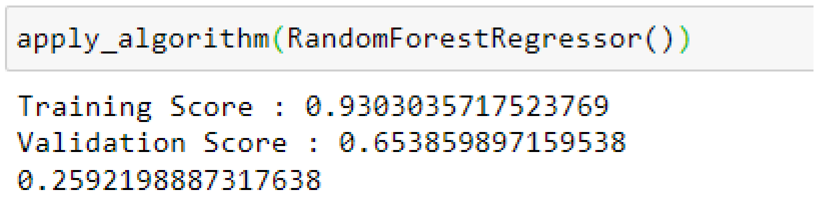
LinearRegession:



```
apply_algorithm(LinearRegression())
```

```
Training Score : 0.22313220867918637
Validation Score : 0.22752790624417374
0.3872431642922473
```

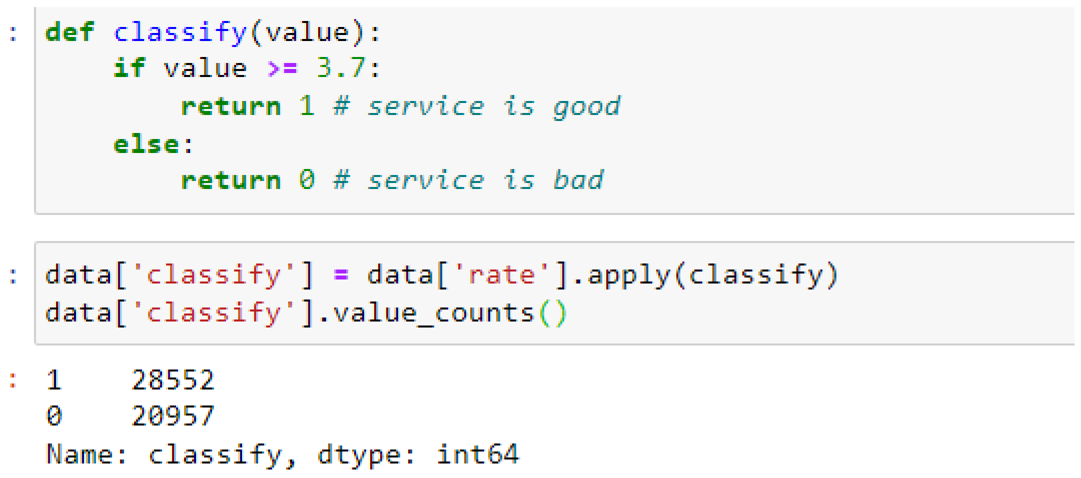
RandomForestRegression



```
apply_algorithm(RandomForestRegressor())
```

```
Training Score : 0.9303035717523769
Validation Score : 0.653859897159538
0.2592198887317638
```

I also work on this data as a classification problem. I add column which define the service is good or bad according to rate value.



```
: def classify(value):
      if value >= 3.7:
          return 1 # service is good
      else:
          return 0 # service is bad
```

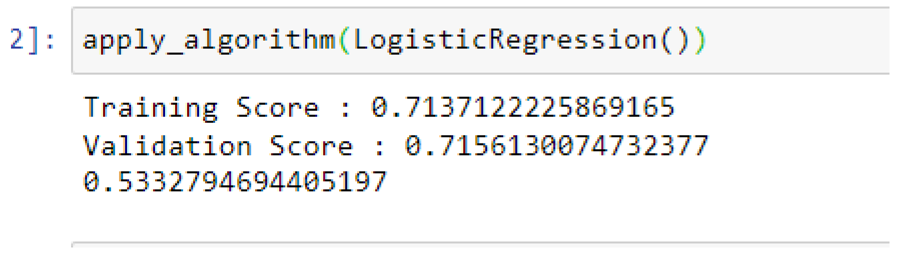
```
: data['classify'] = data['rate'].apply(classify)
  data['classify'].value_counts()
```

```
: 1    28552
  0    20957
  Name: classify, dtype: int64
```

We can observe that data is balanced

Initially, we've tried **Logistic Regression, DecisionTreeClassifier and RandomForestClassifer**

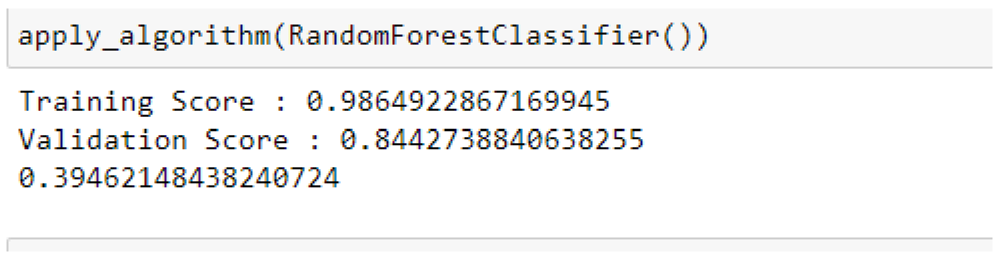
Logistic Regression:



```
2]: apply_algorithm(LogisticRegression())

Training Score : 0.7137122225869165
Validation Score : 0.7156130074732377
0.5332794694405197
```

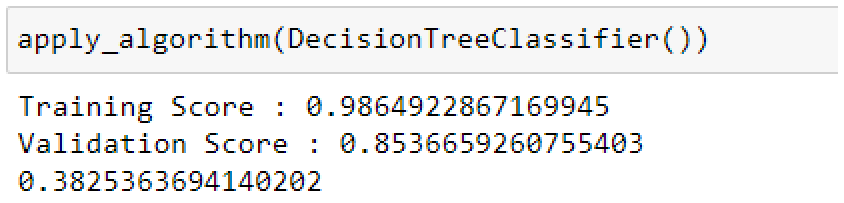
RandomForestClassifier



```
apply_algorithm(RandomForestClassifier())

Training Score : 0.9864922867169945
Validation Score : 0.8442738840638255
0.39462148438240724
```

DecisionTreeClassifer



```
apply_algorithm(DecisionTreeClassifier())

Training Score : 0.9864922867169945
Validation Score : 0.8536659260755403
0.3825363694140202
```