Machine Learning Applied:  The Predicting Restaurant Food Cost based on location ratings and cuisines type.

# Introduction

We all love food. And it is only normal to have a craving for one of your favorite foods from a special restaurant that we all love to have at least once in a month. However, there is a strong factor that will make us reconsider going back to that special restaurant which is cost. The cost may vary from restaurant to restaurant based on many factors such as size of the restaurant, location, rating based quality of the food, service, type of the cuisines and also the time of the restaurant services.

Here in this hackathon, we will be predicting the cost of the food served by the restaurants across different cities in India. We will use Data Science skills to investigate the factors that really affect the cost, lets also analyze some very interesting insights that might help us choose which restaurant serves food worth for money.

# Context of the project

The [hackathon](https://www.machinehack.com/course/predicting-restaurant-food-cost-hackathon/) is about predicting the average price for a meal for the TEST data set and we will train our best Machine Learning model with TRAIN data set, TEST data set won’t have COST column. Using the Train dataset, I have attempted to

1. Import the database
2. Clean the data
3. Analyze the data
4. Standardize the data
5. Draw up the potential ML models
6. Shortlist / Finalize ML model
7. Hyper-tune the ML model
8. Predict the output
9. Save the ML model

This project is an attempt to realize a solution for the easy prediction of cost in Restaurants across India with the data available. Once the optimal ML model has been created / chosen, it can be deployed as a real-time prediction device for Restaurant cost analysis.

# About the Data Set

The Data set in this hackathon consist of following features

**Size of training set:** 12,690 records

**Size of test set:** 4,231 records

**Columns/Features**.

**TITLE**: The feature of the restaurant which can help identify what and for whom it is suitable for.

**RESTAURANT\_ID**: A unique ID for each restaurant.

**CUISINES**: The variety of cuisines that the restaurant offers.

**TIME**: The open hours of the restaurant.

**CITY**: The city in which the restaurant is located.

**LOCALITY**: The locality of the restaurant

**RATING**: The average rating of the restaurant by customers.

**VOTES**: The overall votes received by the restaurant.

**COST**: The average cost of a two-person meal.

Note: You can find the dataset in the link below.

* <https://github.com/dsrscientist/Data-Science-ML-Capstone-Projects>

# **Insights of the Data**

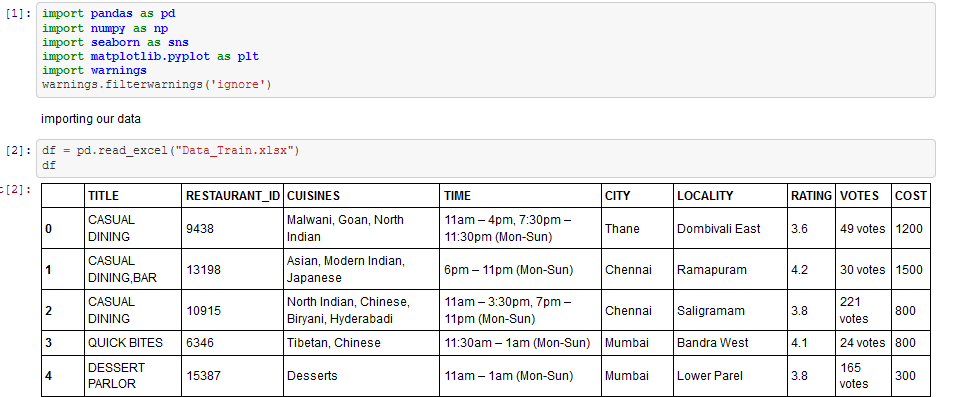
From all the Data available, we can bring out some neat insights or conclusions such as

* Restaurant in which City cost higher?
* Which cuisine type Restaurant cost higher?
* How Time of service related to cost?
* How rating of the Restaurants is related to the cost?
* How votes increase the cost?
* What is the most liked Restaurant type?

# Tools used

1. Python 3.8
2. NumPy
3. Pandas
4. Matplotlib
5. Seaborn
6. Data science
7. Machine learning

# Data Analysis

I have started my analysis of the data by importing required libraries

We will start out Pre-Processing Pipeline by analyzing basic details from the data.

# Exploratory Data Analysis

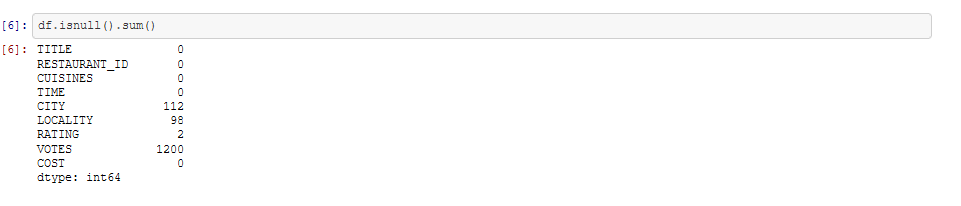
### Duplicate Data.

We will start with identifying the duplicate records in our data and we will remove

those duplicate records since duplicate records will reduce the performance of the data.

The duplicate records in the data are removed from our data set lets identify the Nan values in our data.

### Missing values in the data.

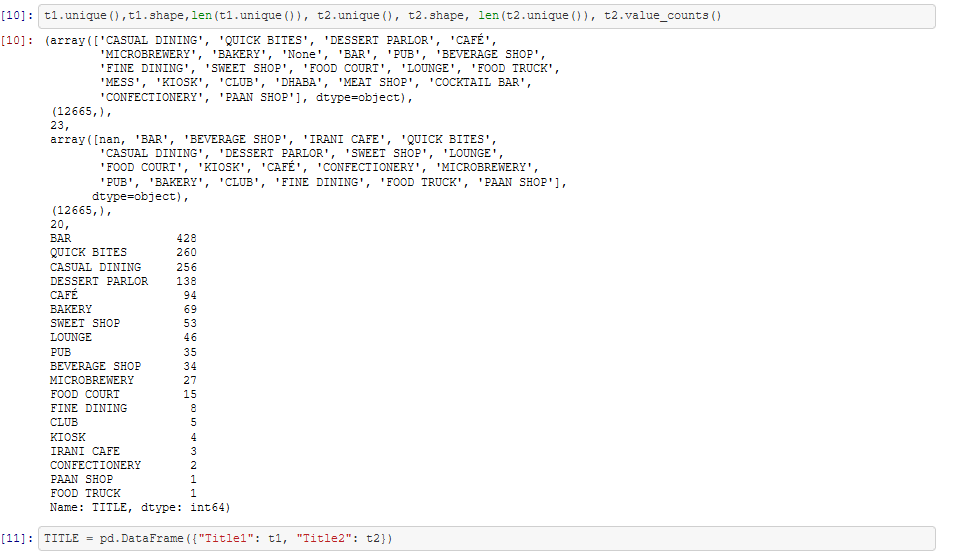
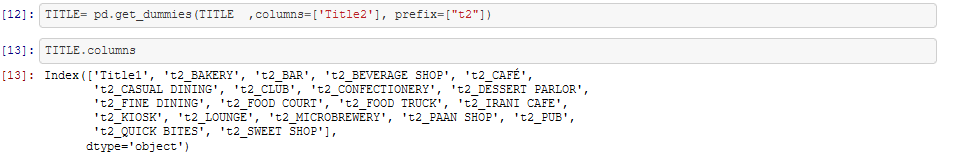
 We will identify the missing/Nan values in our dataset and let’s figure out the how to fill the missing values in our data.

We observe that we have Missing values in four columns namely CITY, LOCALITY, RATINGS, VOTES. We will split our imported data into multiple pieces of Data-Frames with one column in every Data-Frame so we can analyze and clean our data.

### Feature Engineering.

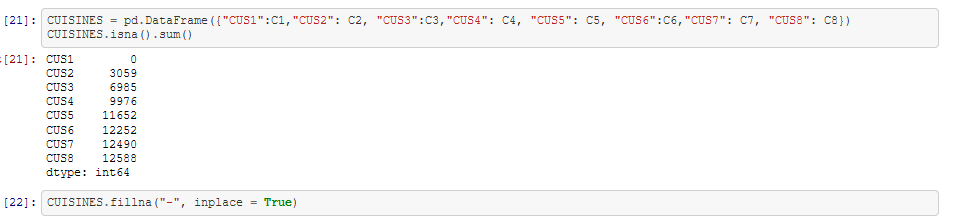
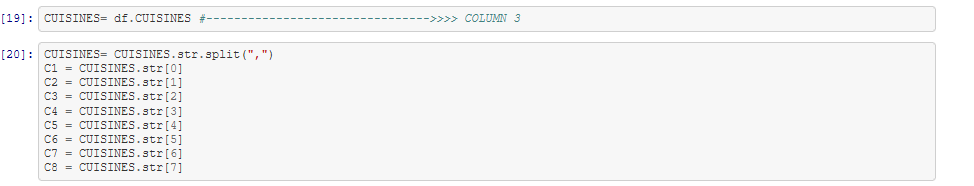
‘TITLE’ is the first piece of data that we have separated from the imported Dataset lets understand the records of data in ‘TITLE’.

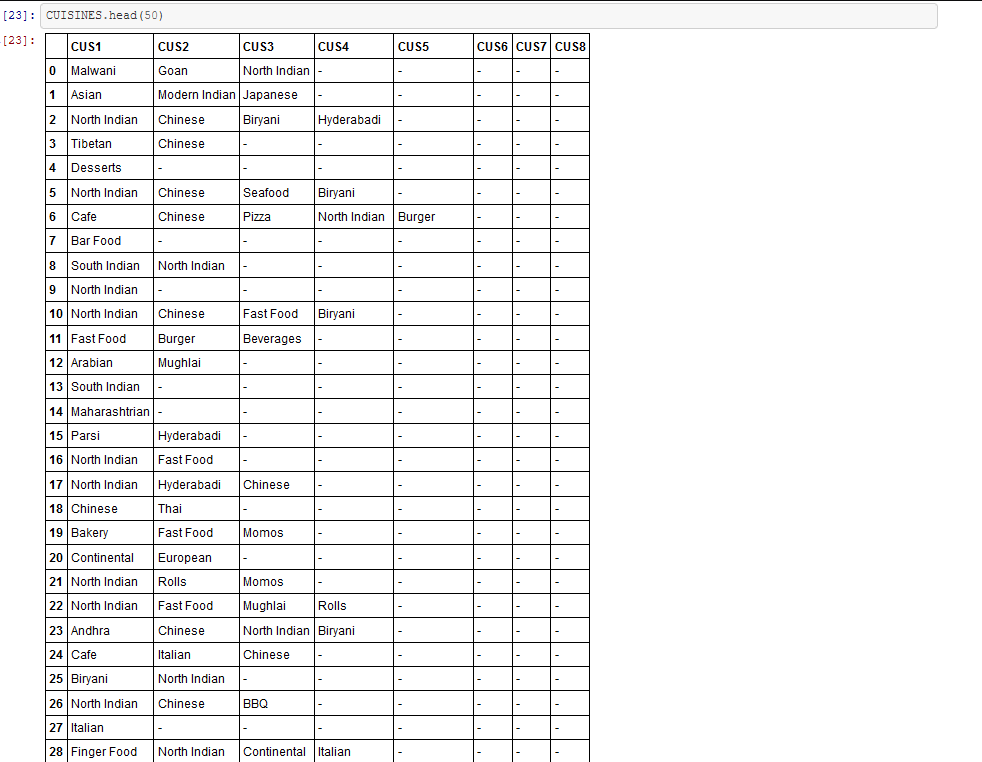
We can observe that ‘TITLE’ is a categorical data but however the common names in the data are manipulated, we are again splitting ‘TITLE’ into two columns t1, t2 as you see above, now we will observe the unique values in t1 and t2 and we will create a new data frame with t1 and t2.

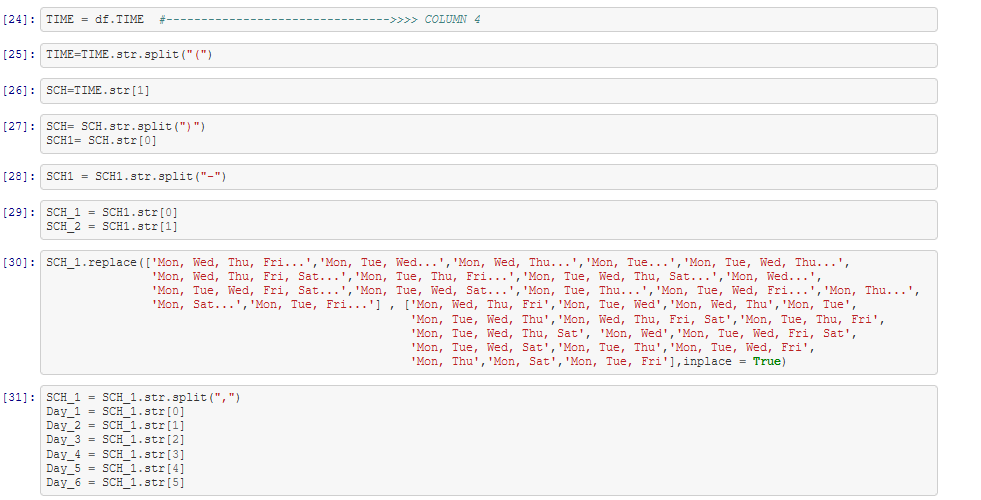
We can see that t2 mostly have addon Features of Restaurant such as BAR, BAKERY, CLUB, PAAN SHOP we will create dummy title for t2 so it will help in our analysis

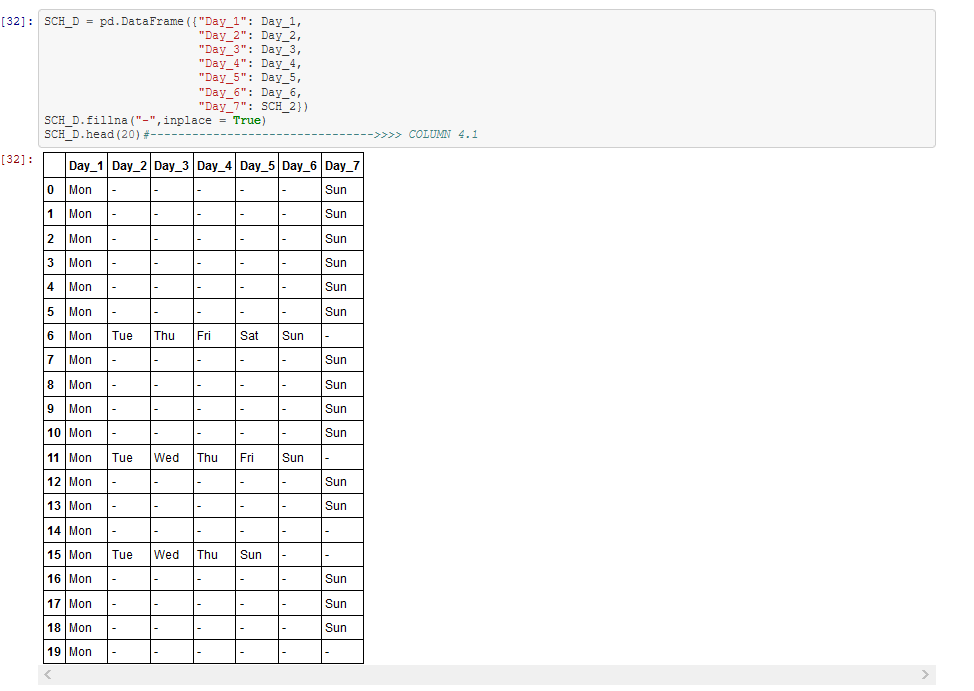
Now we have cleared the records in ‘TITLE’ which will be suitable for prediction similarly we also will clean the remaining columns from imported Dataset.

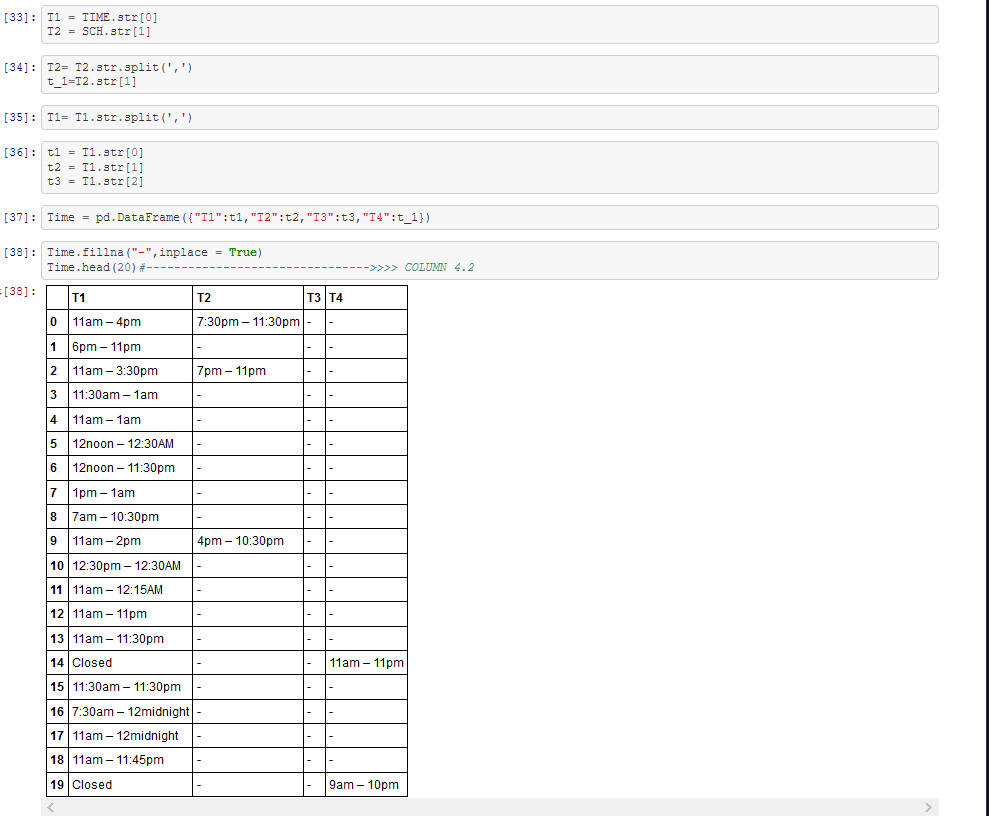
‘CUISINES’ is the second piece that we are splitting from the imported dataset lets analyze and see the records in ‘CUISINES’.

 We are splitting are data by **(“,”)** and we can observe that some records have more than 8 splits but some has only 1. We will further create a Data frame with all these data and we will fill missing values in the data with **“-”**. Initially CUISINES have no missing values but we have splitted our data into 8 pieces because of which we have missing values.

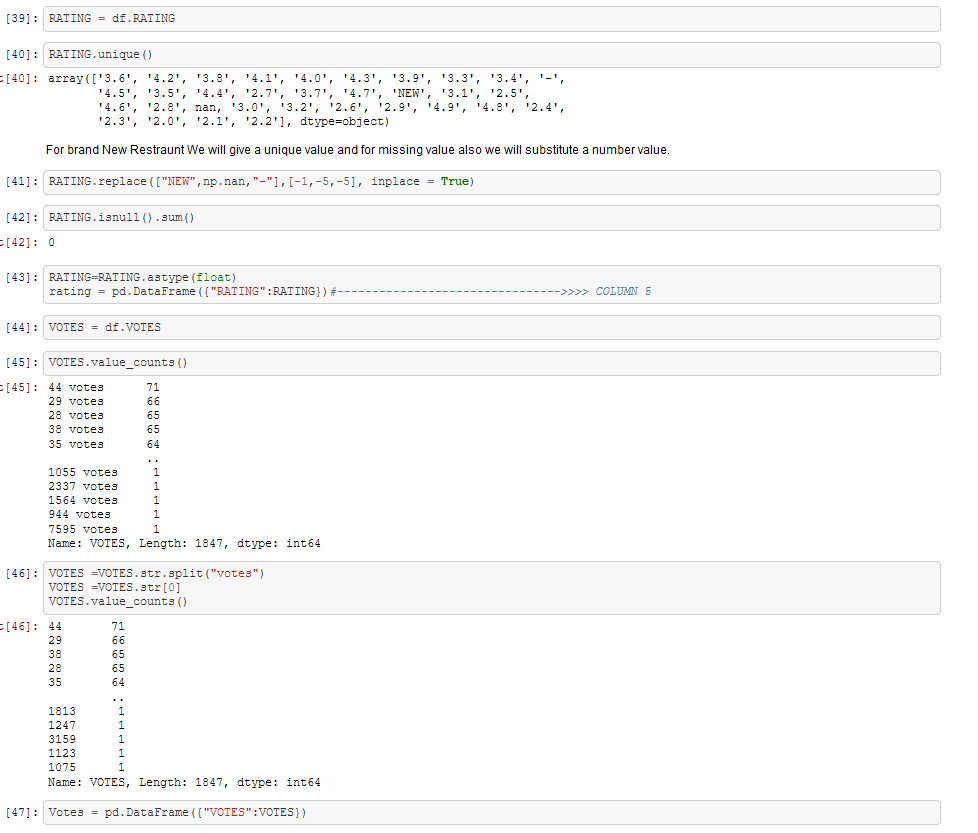
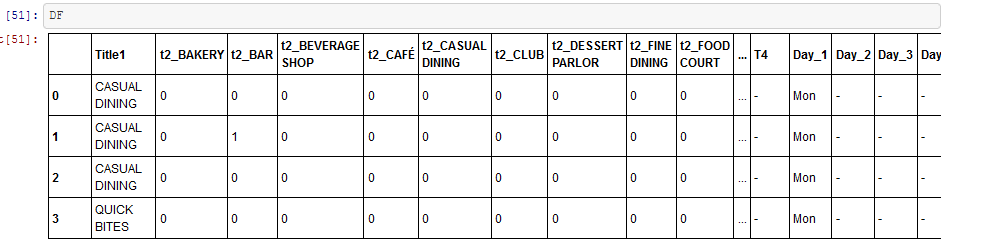
Let’s see the data in CUISINES

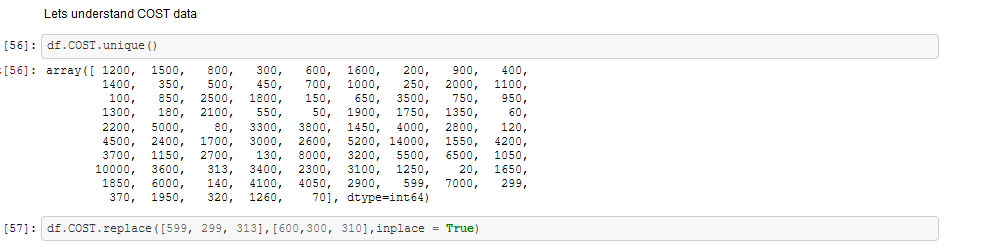
We have cleansed the data in ‘CUISINES’ now we will separate ‘TIME’ and we will split ‘TIME’ into two Data Frames since we have more information in ‘TIME’ and also as like before the Nan values that are getting generated because of splitting the data will be replaced by **“-”**.

We are creating SCH\_D Data Frame and we will place our collected data in SCH\_D

We will filter the remaining data in ‘TIME’ in new Data Frame.

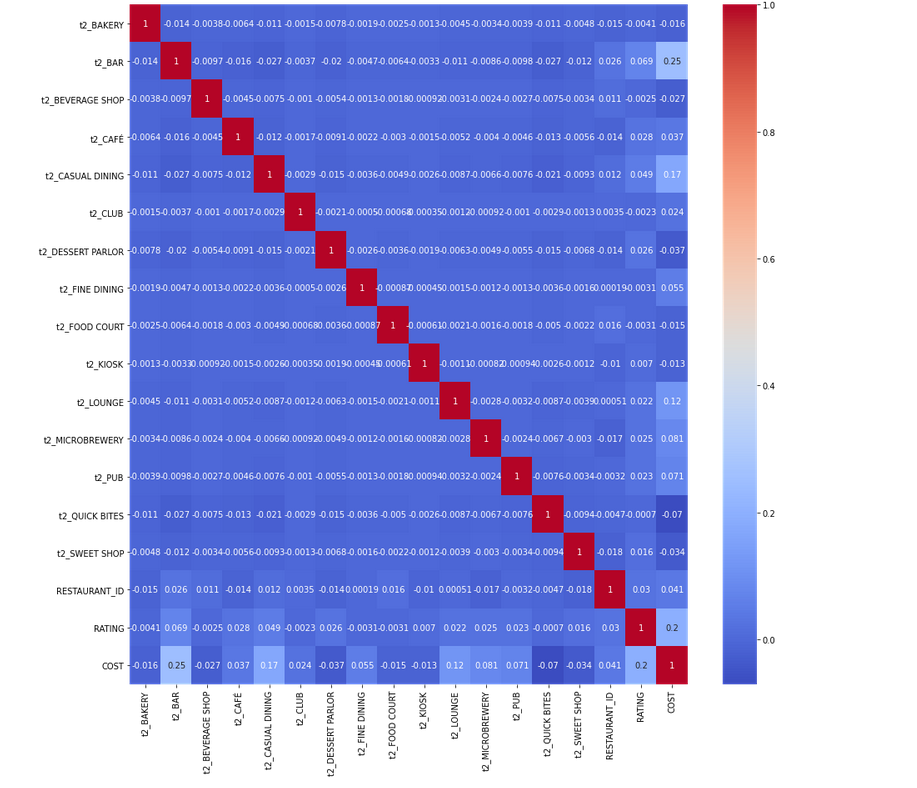
We have cleared records in ‘TIME’ we will now let’s split ‘RATING’ and ‘VOTES’ from imported data and clear the data in ‘RATING’ and ‘VOTES’ so we will them for prediction.

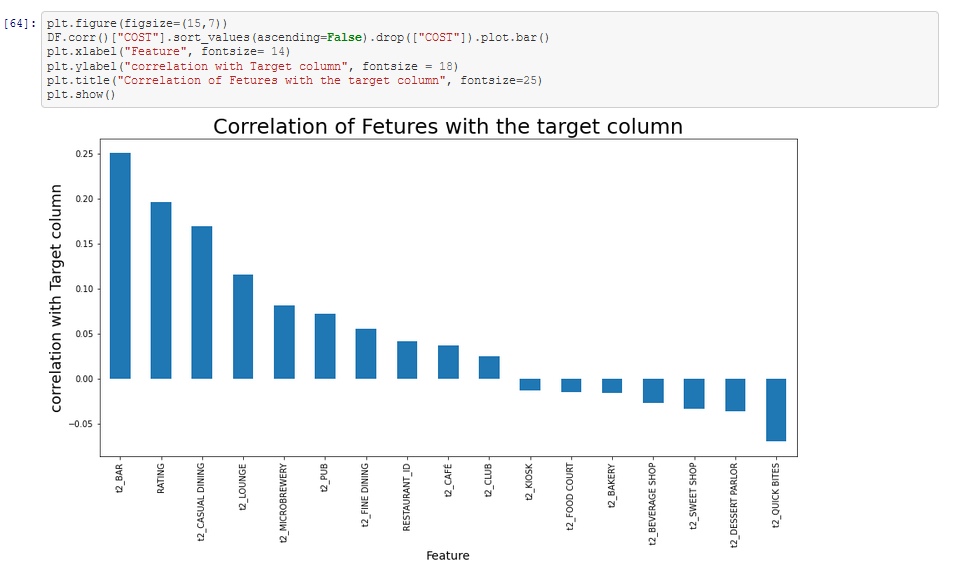
We have now splitted our data into multiple Data Frames and cleared our data to make our data predictable and we have also not included ‘RESTAURANT\_ID’ since it is only unique value and will not help us in our prediction. Now we will join all the pieces of Data Frame and name our collected Data as DF.

In the above Data Frame, we have not included ‘COST’ lets analyze ‘COST’ and also include it into our engineered Data Frame DF.

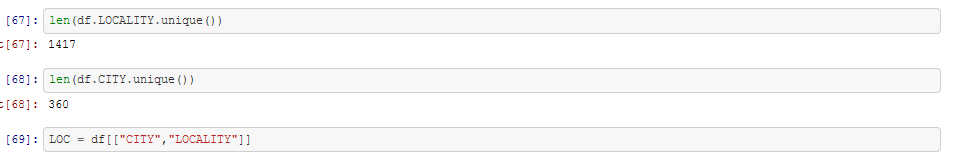
We can see that the data in ‘COST’ is mostly multiples of 10 expect 599,299,313 and we are replacing it with 600,300,310 the nearest multiple numbers of 10 so the we can do better cost prediction.

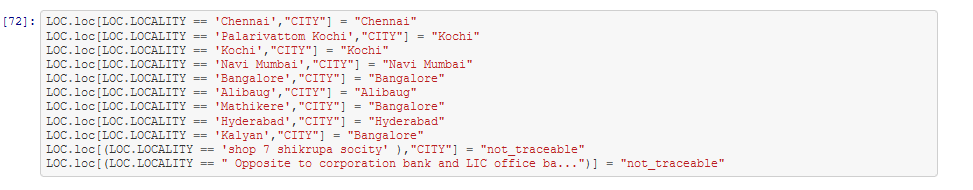
### Data Correlation

Let’s visualize the correlation of the Data with Seaborn Heatmap.

 Key Observation from the correlation is Target variable ‘COST’ have more correlation with RATING and t2\_BAR but let’s visualize the correlation of Feature Variables and Target Variables with bar plot.

We will now separate ‘CITY’ and ‘LOCALITY’ columns and we will clean the data from the original data set that is imported. And also ‘CITY’ has 112 missing records and ‘LOCALITY’ has 98 missing records. We will fill the missing data in ‘CITY’ with the data available in ‘LOCALITY’ and drop ‘LOCALITY’ since both defines the same meaning for prediction.

Let’s analyze the CITY and LOCALITY datasets.

We have defined LOC as a separate Data frame which will have CITY and LOCALITY as columns. And also, like said before we are filling data in CITY with information available in LOCALITY.

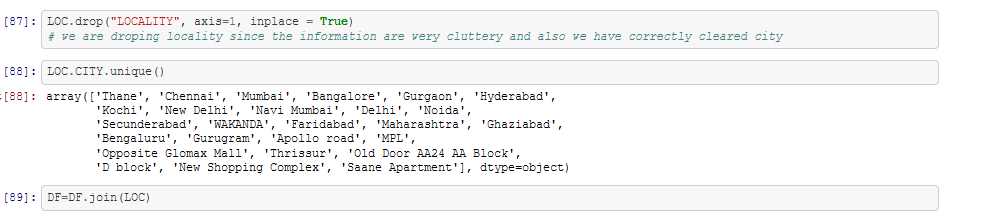


We are now replacing remaining missing values in ‘CITY’ with name ‘WAKANDA’ since ‘WAKANDA’ is not a real place it’s a frictional place appearing in a famous comic.

And also, we are replacing few records available in ‘CITY’ since some records have upper case and lower case for same place and some have spelling and typo errors on replacing the same, we will reduce the number of unique records in ‘CITY’.

Let’s identify the further missing values in LOC Data frame.



We have filled all the data missing in CITY and as said before we also have corrected the typo errors in data. We will now drop LOCALITY since LOCALITY and CITY will give the data same details in the data and also that we have 98 missing values in the data. Post which we will join that with our Feature Engineered Data Frame DF.

# Data Insights.

## Univariate Analysis

1. **We will first analyze ‘COST’ data with Histogram and lets also see the mathematical summary of ‘COST’.**



**MATHEMATICAL SUMMARY OF COST:**

count 12665.000000

mean 655.400711

std 627.389909

min 20.000000

25% 300.000000

50% 500.000000

75% 800.000000

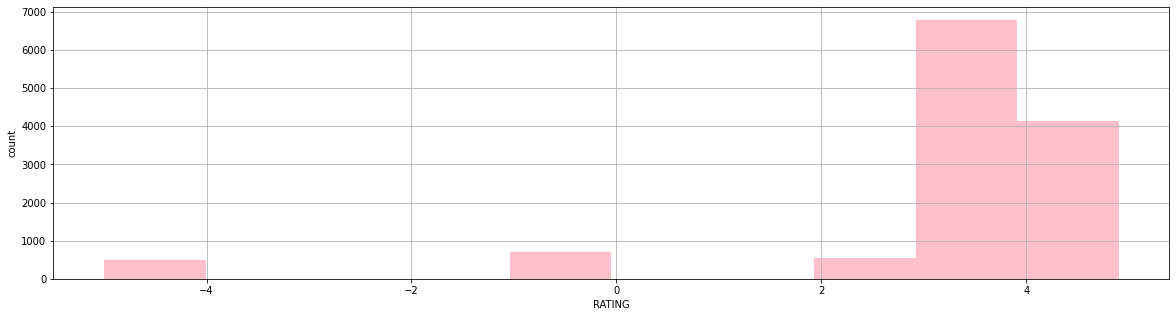
max 14000.000000

Name: COST, dtype: float64

#### **Key Observation:**

From above we can see that most of the restaurant charges within Rs: 1500/- per dine with average of 655 and surprisingly minimum cost is only Rs: 20/- but max is Rs: 14000/-.

1. **Let’s analyze the RATING data and let’s try to gets some insights with mathematical summary of RATING**



**MATHEMATICAL SUMMARY OF RATING:**

count 12665.000000

mean 3.175199

std 2.021152

min -5.000000

25% 3.400000

50% 3.800000

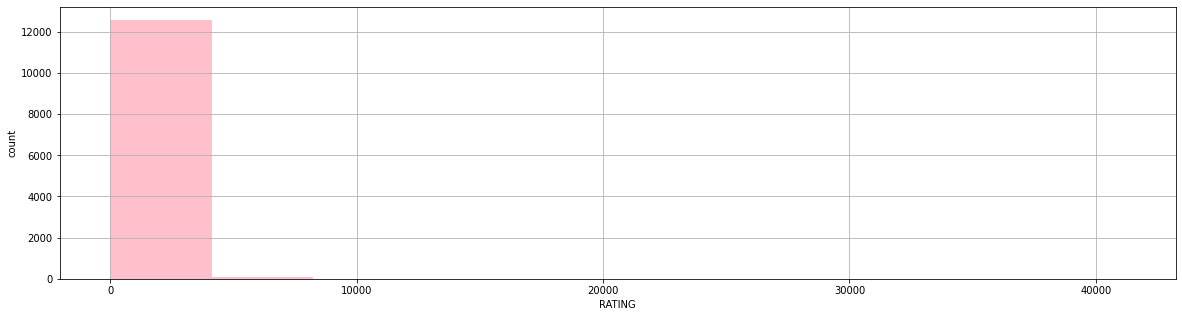
75% 4.000000

max 4.900000

#### **Key Observation:**

We can see most of the restaurant share the ratings between 3-4 and highest being 4.9

1. **Now let’s analyze VOTES data with its mathematical summary.**



**MATHEMATICAL SUMMARY OF VOTES:**

count 12665.000000

mean 376.242637

std 811.549378

min 0.000000

25% 39.000000

50% 132.000000

75% 405.000000

max 41186.000000

Name: VOTES, dtype: float64

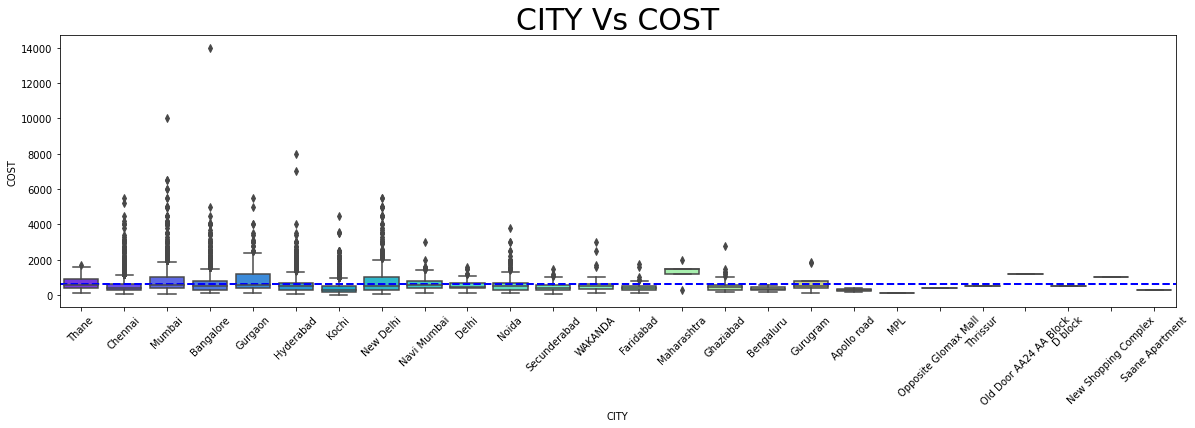
#### **Key Observation:**

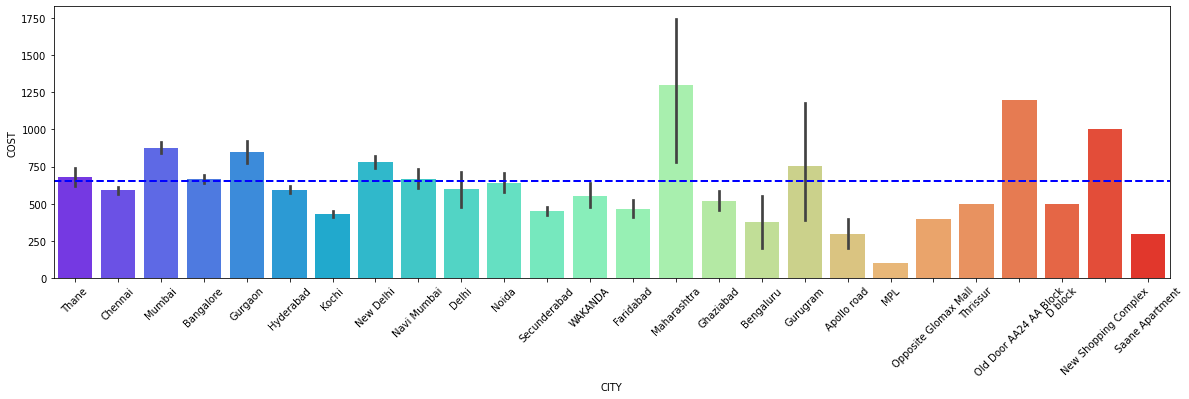
Most of the votes are between 0 - 400 with mean of 376 highest being 41186.

## Multivariate Analysis.

Let’s gets some insights in analyzing Feature Variables with Target Variable.

1. **COST Vs CITY.**



**Key Observation:**

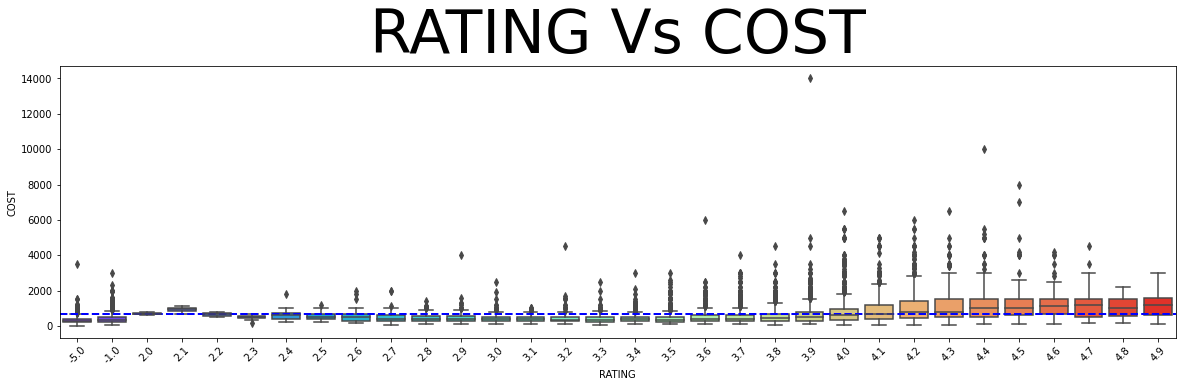
1. Bangalore charges the highest of Rs: 14000/- per Dine.

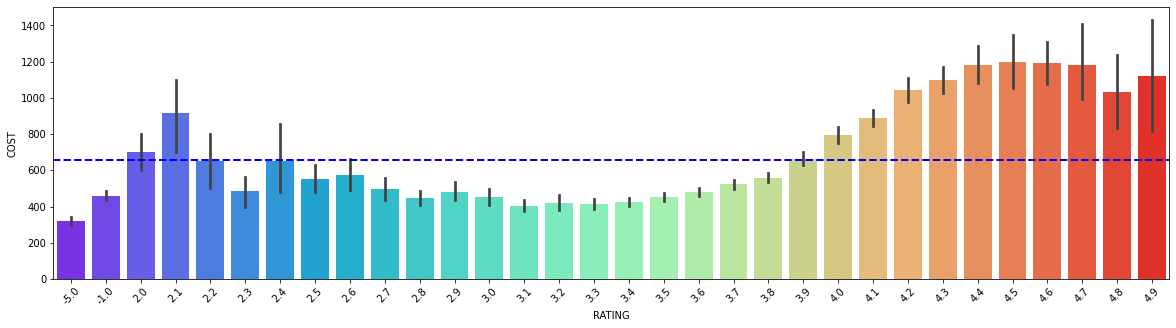
2. Maharashtra overall population of Restaurant charges above the mean of COST and also the total cost of all restaurants put together being the highest in the country.

3. Most of the cities are around the mean.

 4. Wakanda(unknown) charges lesser than the mean. Which is in Wakanda restaurants are not charging high food

1. **RATING Vs COST.**





**Key observation:**

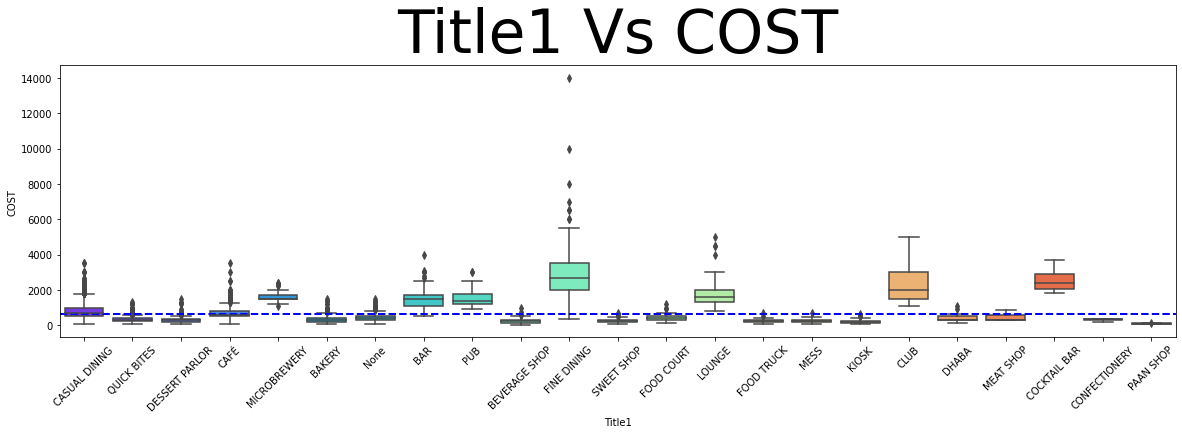
1. We can see as per the Rating increase the cost also increases.

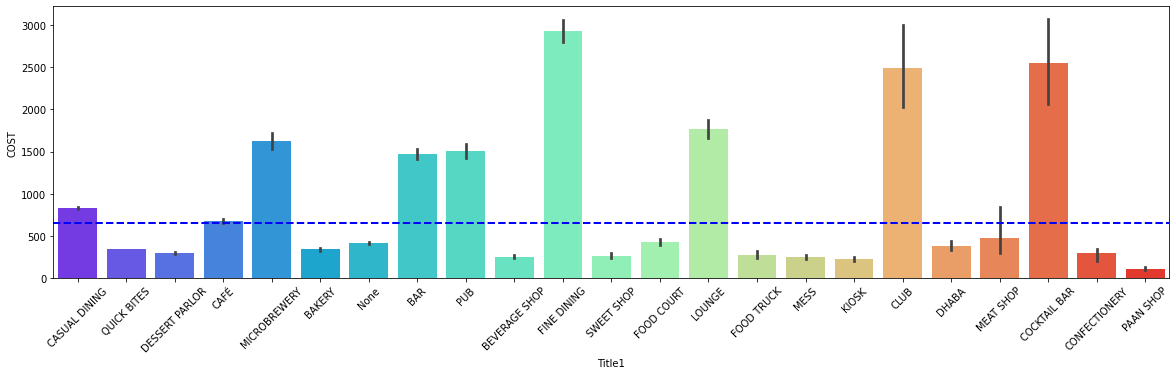
2. The Mediocre rated restaurant actually charges lesser but they are actually lesser in number.

3. Missing valued Restaurant and new restaurant charges lesser.

4. We can observe the rating 2.0, 2.1, 2.1, 2.4 restaurants are charging higher than the restaurant rated between 3 to 4 which is the general audience for the restaurant rated them lesser might be because of the service or because of the quality of the food.

1. **TITLE Vs COST.**

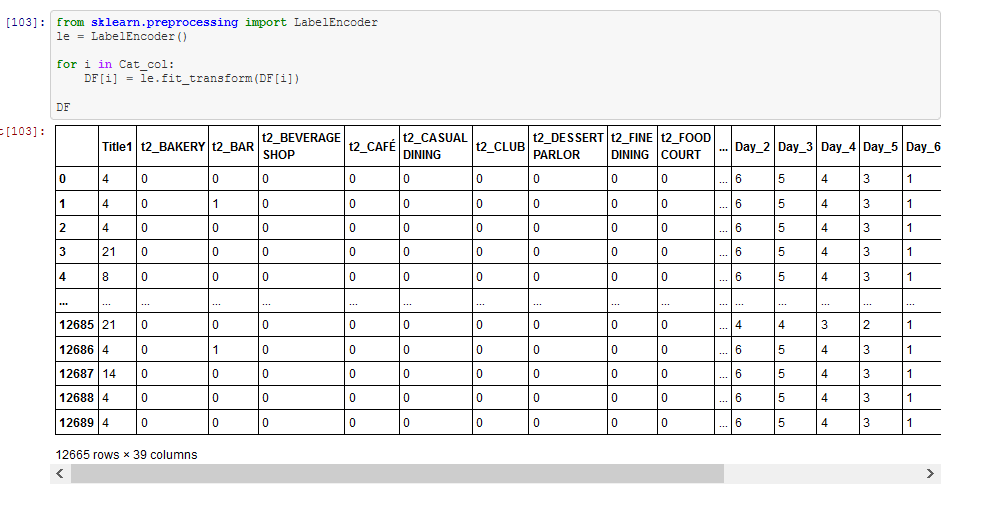
****

****

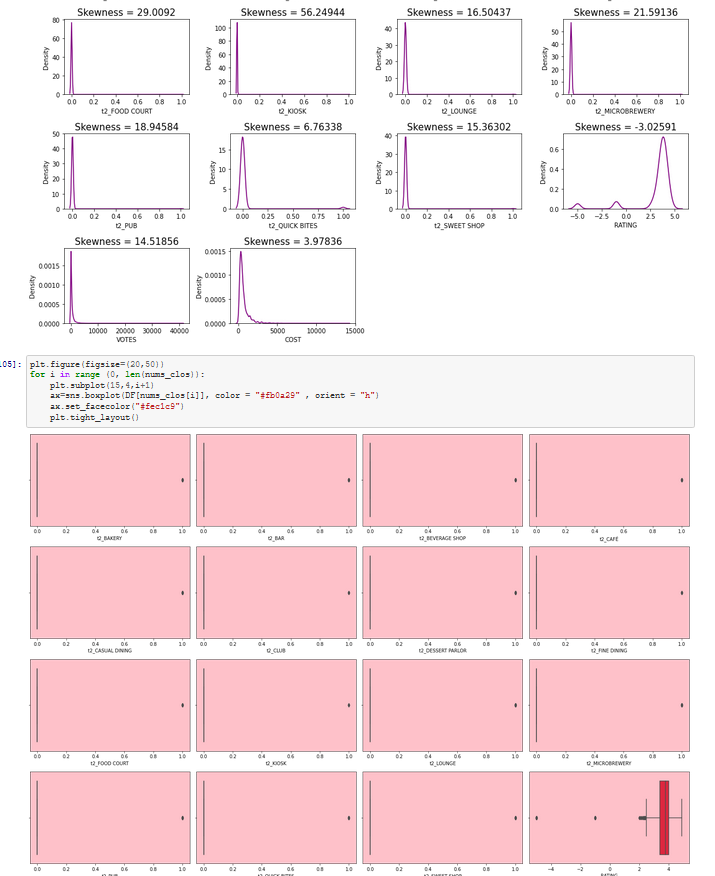
**Key observation:**

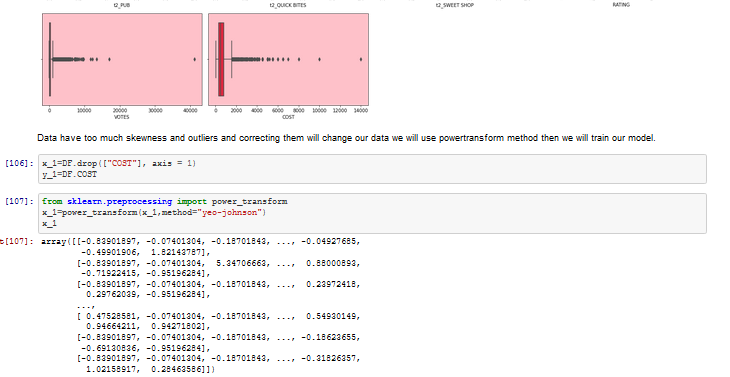
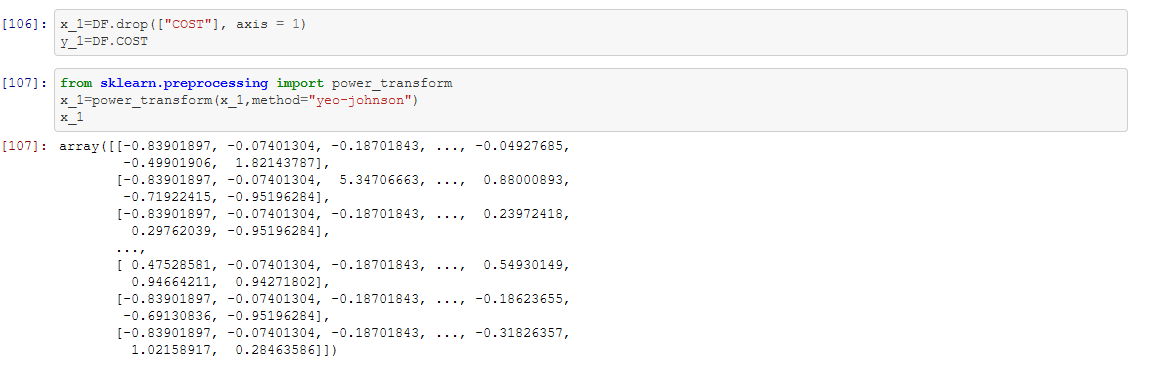
1. From above we can say that Fine Dine Type of restaurant is costlier than others.
2. From all our above observation we can tell our story that restaurant in Bangalore with 3.9 rating with very lesser votes Fine dine type cuisine at the prime time is charging RS:14000/ - per dine

# Label Encoding

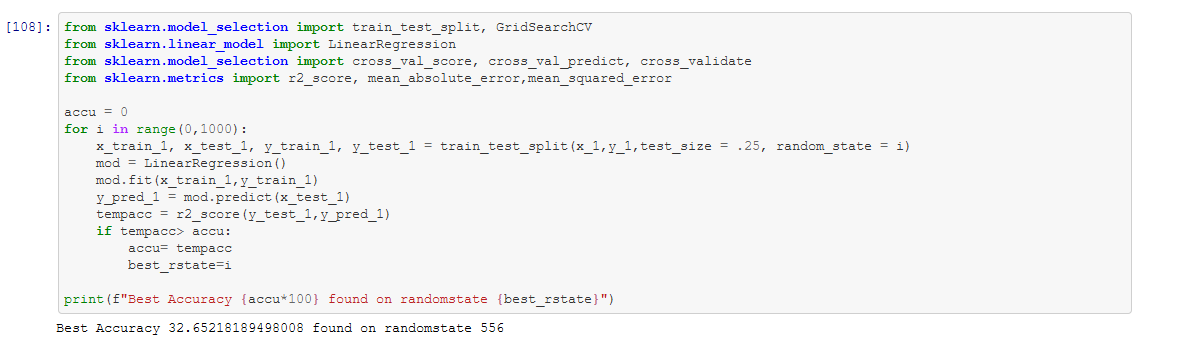
Using sklearn preprocessing Label Encoder we are converting all the categorical columns into numerical values so that we it will be helpful in doing our prediction.

We have converted all our data into numerical values lets finally end our preprocessing pipeline with identifying and correcting Skewness and Outliers.

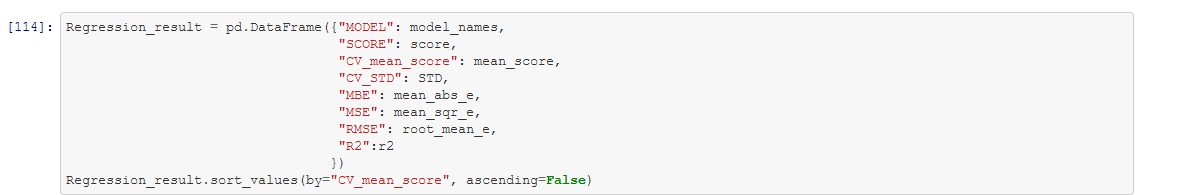


 As we can see from above the data have skewness and Outliers but correcting them will change the definition of the data or we might loss data. But skewness and outliers in the data will reduce the performance of our prediction model we will use Power Transformation (“YEO-JOHNSON”) method to reduce the skewness and will make our data ready for training.

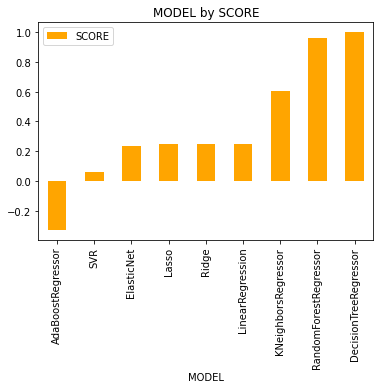
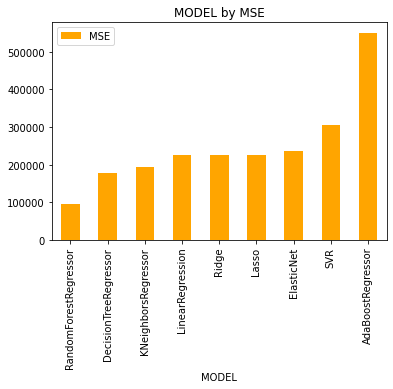
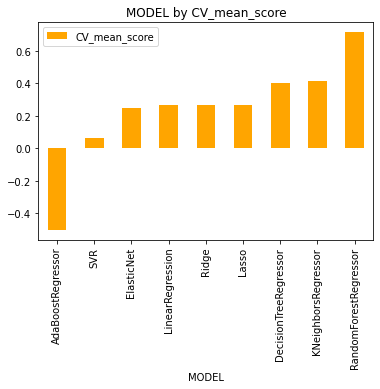
# Building Machine Learning Models.

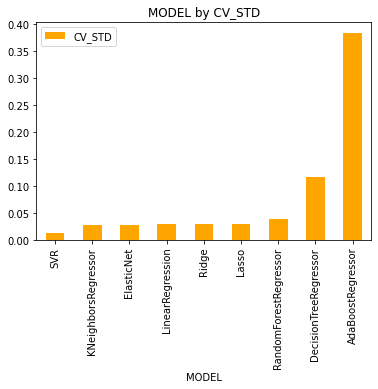
 Our data is ready for training we will split our data into test and train data to analyze the score of prediction and also to start with lets also filter and find best random\_state parameter with Linear Regression model.

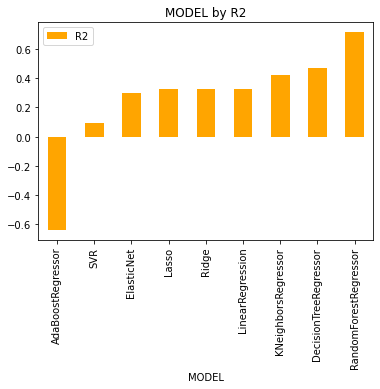
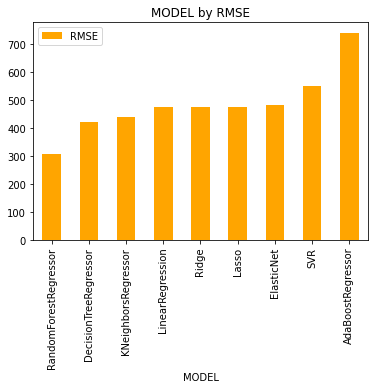
 We got our best random state as 556 now let’s split our data and train our data with nine different Regression models. Test data size will be 25%.

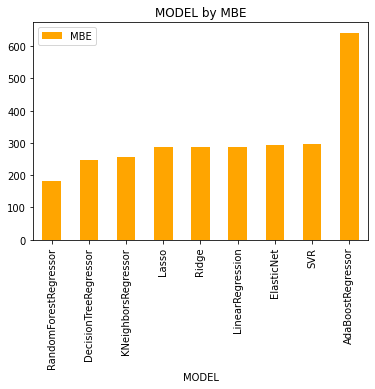
Filtering best of our model with Cross Validation Mean Score, mean absolute error, Mean Squared error, root mean error and R2 score

|  | **MODEL** | **SCORE** | **CV\_mean\_score** | **CV\_STD** | **MBE** | **MSE** | **RMSE** | **R2** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **8** | RandomForestRegressor | 0.959056 | 0.715856 | 0.038838 | 182.044105 | 95006.232934 | 308.230811 | 0.717433 |
| **5** | KNeighborsRegressor | 0.603356 | 0.413735 | 0.027209 | 257.525734 | 194710.651089 | 441.260299 | 0.420893 |
| **6** | DecisionTreeRegressor | 0.999694 | 0.404283 | 0.115743 | 247.787207 | 178421.944675 | 422.400219 | 0.469338 |
| **1** | Lasso | 0.250354 | 0.265359 | 0.030023 | 286.465383 | 226735.923968 | 476.167958 | 0.325643 |
| **2** | Ridge | 0.250591 | 0.265274 | 0.029715 | 286.588132 | 226441.203722 | 475.858386 | 0.326520 |
| **0** | LinearRegression | 0.250591 | 0.265272 | 0.029714 | 286.589834 | 226440.496695 | 475.857643 | 0.326522 |
| **3** | ElasticNet | 0.235966 | 0.251170 | 0.027905 | 292.957281 | 234975.807549 | 484.743032 | 0.301136 |
| **4** | SVR | 0.057973 | 0.067222 | 0.012830 | 296.905068 | 304410.280648 | 551.733886 | 0.094624 |
| **7** | AdaBoostRegressor | -0.327724 | -0.500642 | 0.384598 | 641.588820 | 551536.970929 | 742.655351 | -0.640378 |



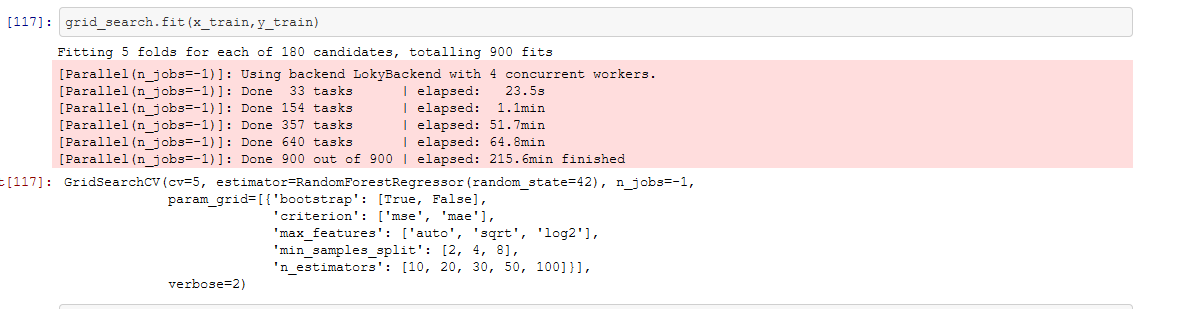
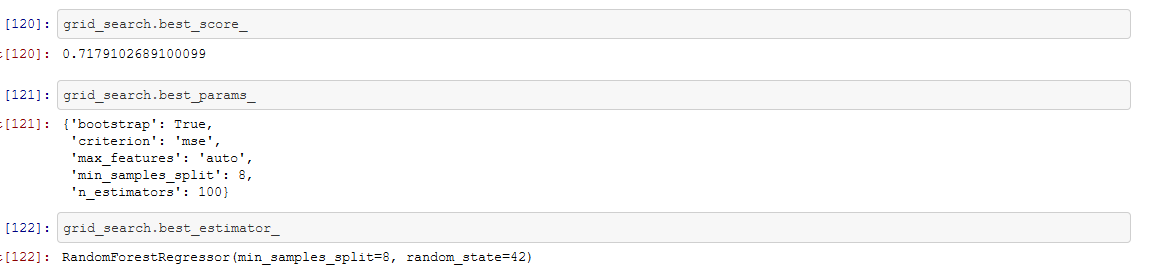


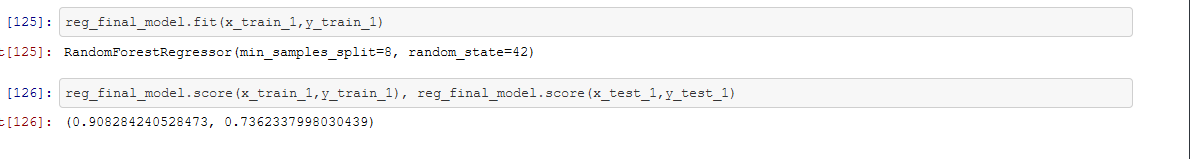




From above its clear that Random Forest Regressor has highest CV Sore of 71% and model score of 95% lets Hyper-Parameter Tune Random Forest Regressor model to reduce the overfitting and save our model for predicting the TEST dataset.

# Hyper parameter tuning

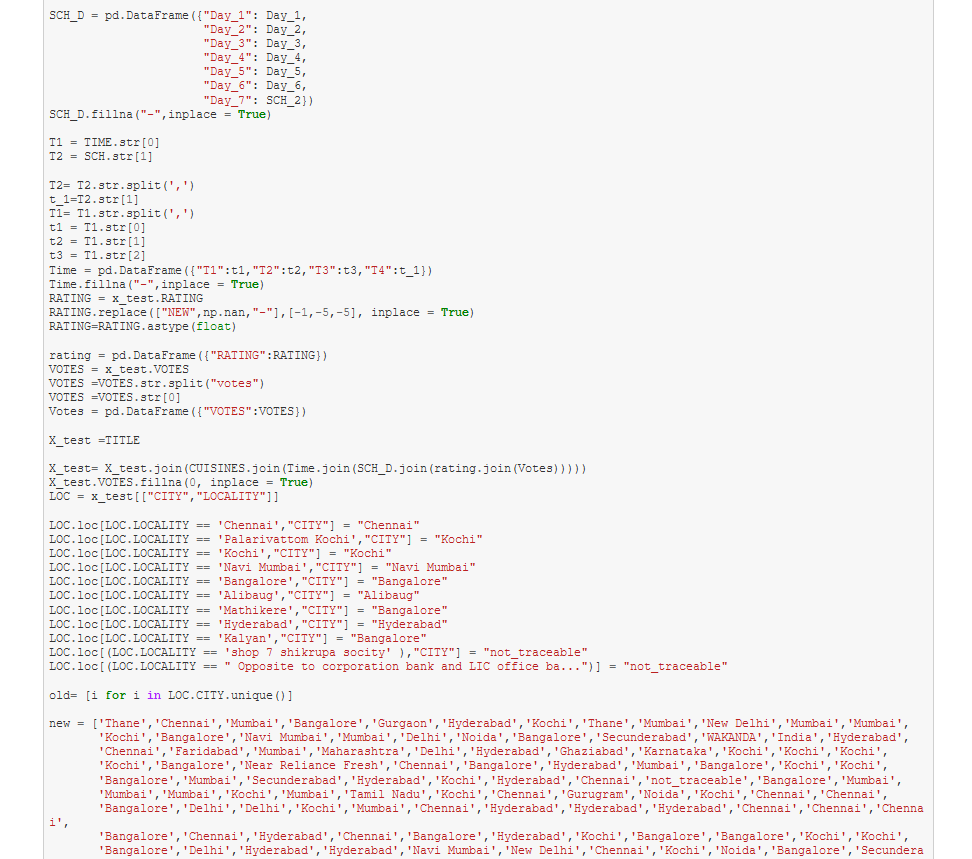
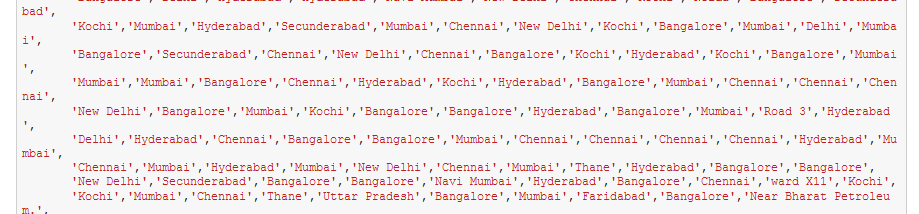
Let’s also observe the grid score and best estimator of our Grid search CV.

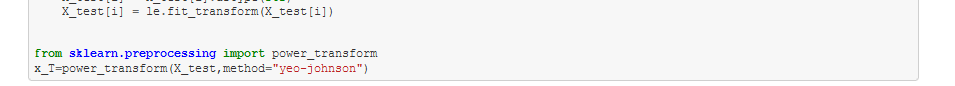
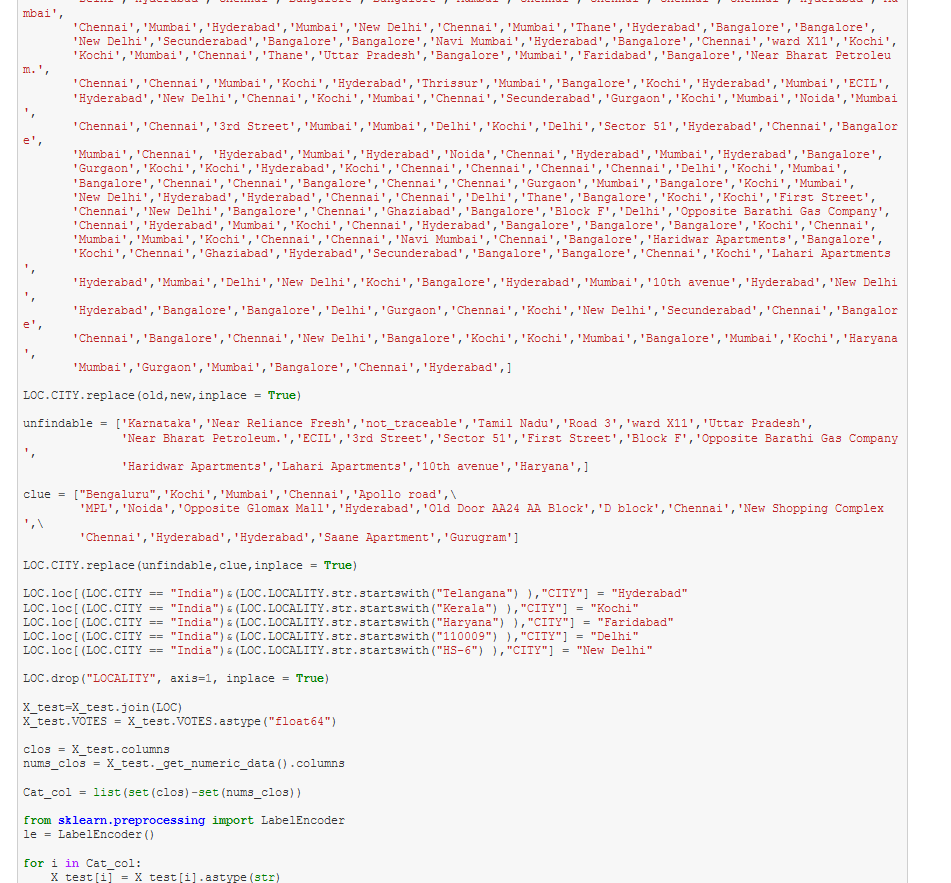
Let’s also see the score of the Hyper parameter tuned model before saving our model.

We have assigned “reg\_final\_model = grid\_search. best\_estimator\_” we can see that our model has reduced scored from 95% to 90% because it has corrected the overfitting. We are saving our model with joblib library.

# Saving the best Model

We have saved our model in as ‘**Restaurant\_price\_prediction.obj**’ obj format with joblib library now let’s import the TEST data set and we will Feature engineer our data as like we did with the TRAIN data set.





We have imported our TEST data and have cleaned it for prediction lets import our saved model and predict the Cost for test dataset.

We have predicted our data and saved it in separate CSV file as “Predicted\_COST.csv”.

# Summary and Conclusion:

We have cleansed our data and prepared for training on splitting it on Target and Feature variable. On training or data with 9 models we have finalize Random Forest Regressor with model score of 95% and with CV-Score of 71%. Further to increase the performance of our model we have hyper tuned our model with GridSearchCV and we have reduced the overfitting our final model score is 0.908 with best score of 0.717. The above study helps us to understand the business in Restaurants. How the cost is varying across the Restaurants, with the Study we can tell how city, Time, Cuisine Type, votes and rating are related to each other and how it is affecting the cost. With the help of the above study, we have figured out which type of restaurant is charging more and which type is charging less.