**RestAurant Food Cost**

I came across another project to design prediction model of Restaurant Food Cost using Machine Learning. The prediction of Cost of Food is the main intent to design this model using other features of the dataset. This dataset having such features having multiple values for a single cell, I needed to implement some codes to extract those values and prepared the dataset for modeling.



**Description:**

The Restaurant Food Cost Dataset contains several features like the ID of the Restaurant, cuisine type, timing, city, locality, votes, rating, cost of two person meal etc. The Dataset has 12690 training records and 4231 testing records.

The features are:

**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.

**Problem Definition:**

We love food so much. We want to go favourite restaurant to get our favourite and delicious food. But there is one factor that will make us reconsider having our favourite food from our favourite restaurant, the **cost.**

Here in this hackathon, I will predict the cost of the food served by the restaurants across different cities in India using Python and Machine Learning.

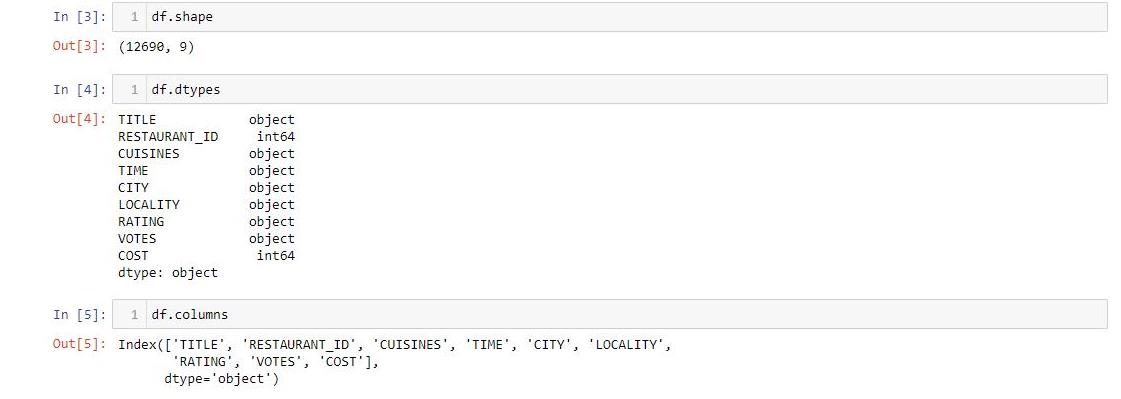
**Data Analysis:**

In the Jupyter Notebook, First of all Warnings library is used to ignore the warnings so that the programmers or viewers won’t be confused. Then the **Resturant Food Cost Data Train** dataset is uploaded by using pandas library code pd.read\_csv as the dataset is a csv file.



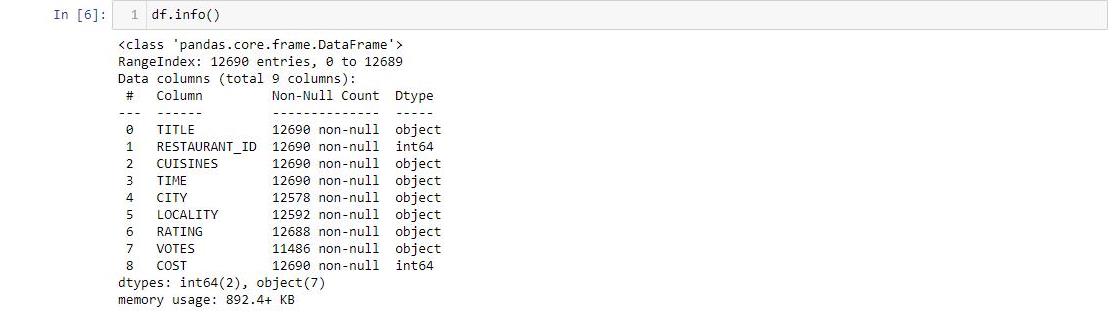


The shape and type of the dataset is determined using df.shape and df.types respectively where df is the DataFrame of the dataset. df.columns is used to list the features/columns of the dataset.



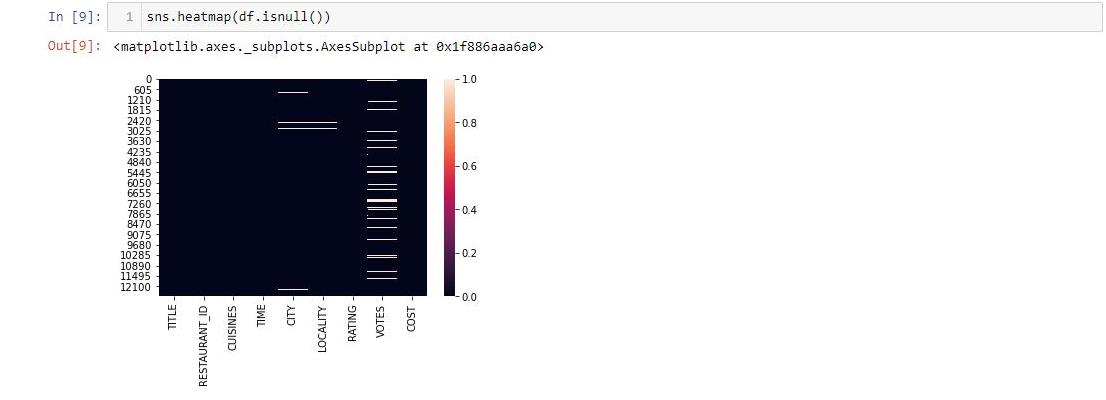
* The dataset has 9 features.
* RESTAURANT\_ID and COST are integer type and rest are object type.

df.info shows the dataset is not null for some columns. And there are some null values.

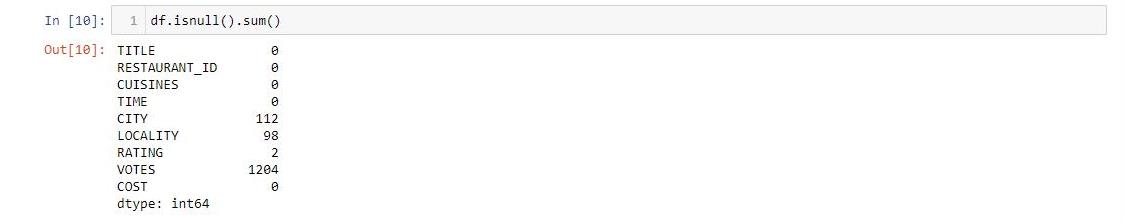


Importing Numpy, Seaborn and Matplotlib for **Exploratory Data Analysis(EDA)**. First command is used to check null values in the dataset.



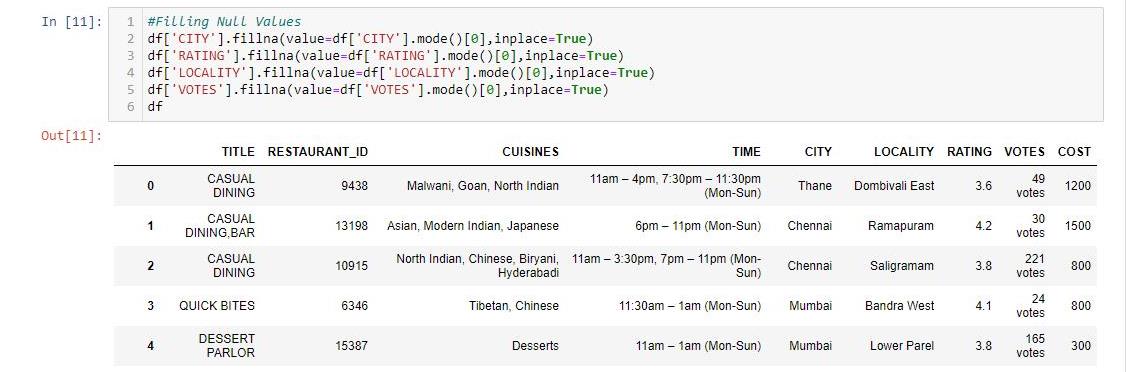


* There are so much null values in the VOTES column and some in CITY and LOCALITY.



* The VOTES has highest null values. CITY, LOCALITY and RATING has also some null values.

Let’s fill the Null Values. As all the null value columns are categorical, the Null value is filled with the mode of that column.



Dropping the ID Column as it is for no use in prediction model.



Converting VOTES column into numeric by removing “ votes” from values.

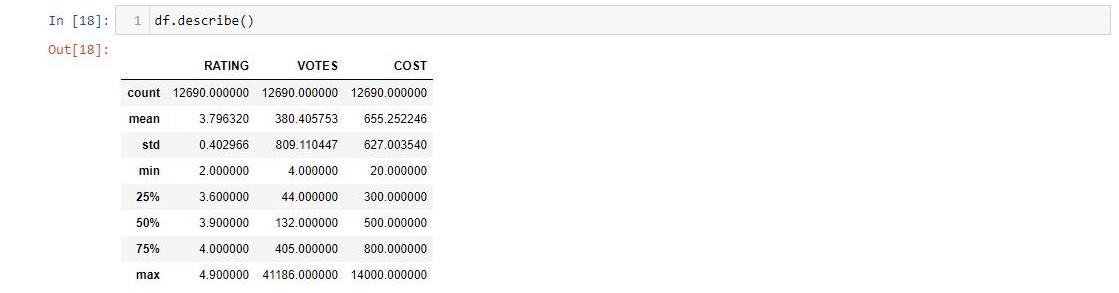


Replacing unsual values with the mode in RATING column



To describe the dataframe, df.describe is used to show the minimum, maximum values of the dataframe for each column data.

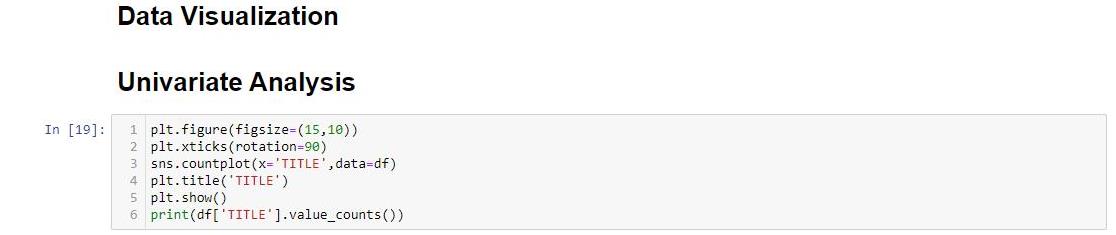
It also gives the 25th,50th,75th percentiles and mean & standard deviation of the dataframe. 50th percentile is known as median.



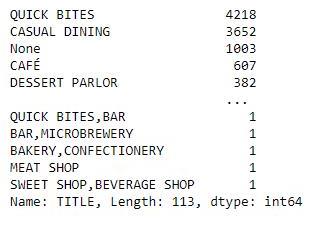
* There is huge difference in 75% and maximum value in VOTES and COST column.
* There is huge difference between mean and 50%(median) in VOTES column.

**Data Visualization** is the major part of EDA process. It shows the graphical representation of the features and relationship between them.

***Univariate Analysis*** is used to represent one feature’s distribution in the dataset and the counts for a categorical feature.

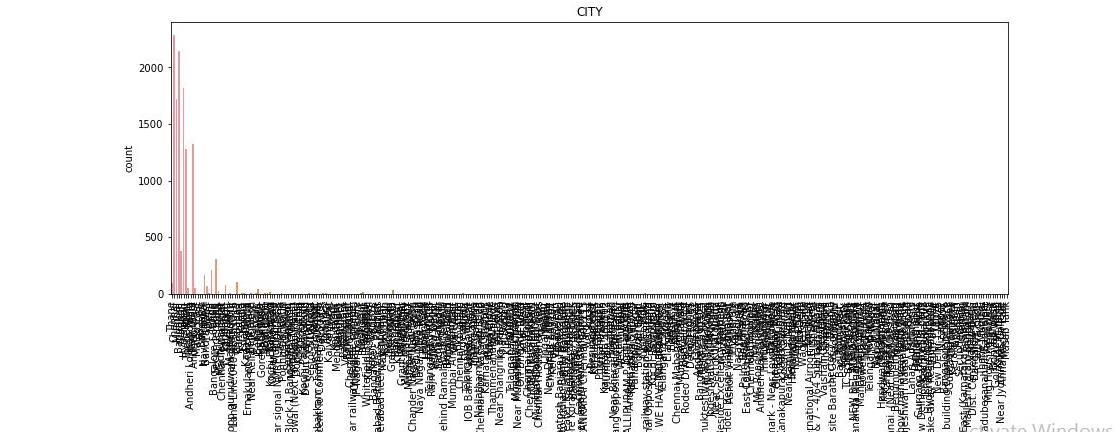
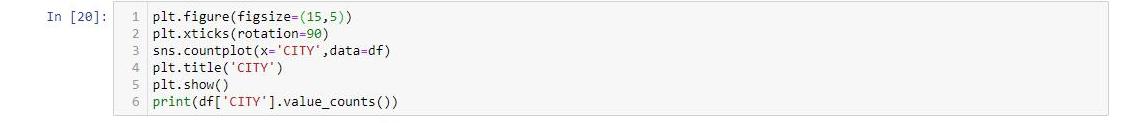






Most of the Resturants are QUICK BITES and CASUAL DINING according to the TITLE Column. But It is not clear how many duplicates are there of in each multivalue values in the entire column.

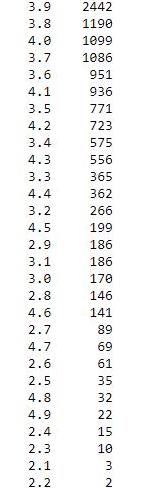
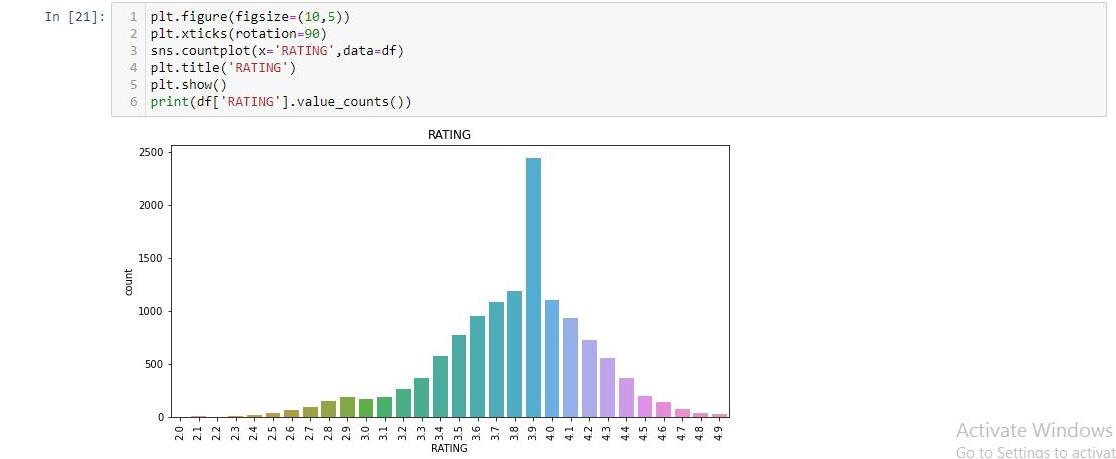
Checking How many Restaurants in each city of India.





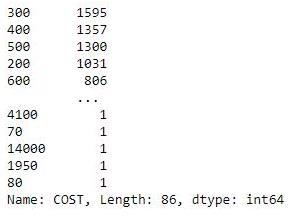
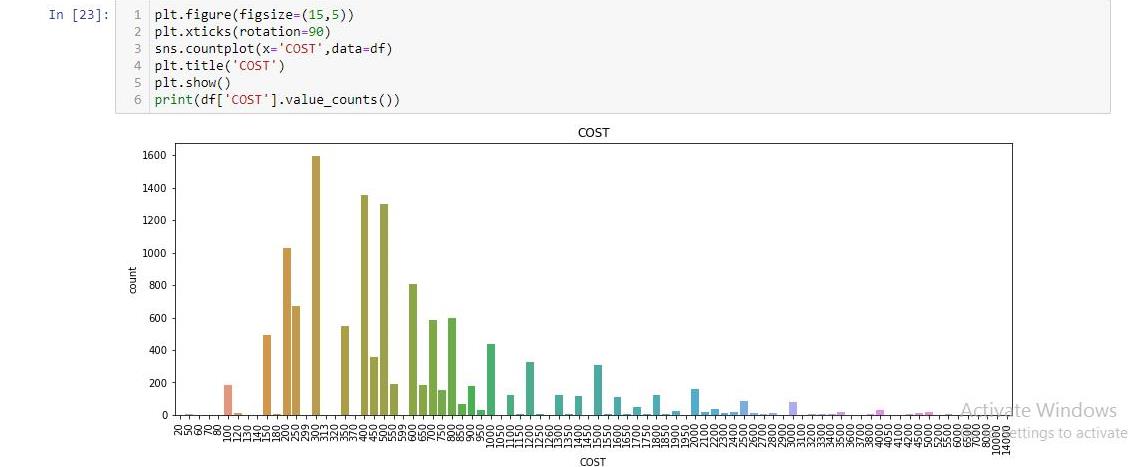
* Most of the Restaurants are in Chennai & Bangalore.
* There are 359 cities in the dataset.

Checking the ratings range in all restaurants of India.



* Most Ratings given to a Restaurants is 3.9.

Checking the COST distribution in the dataset.

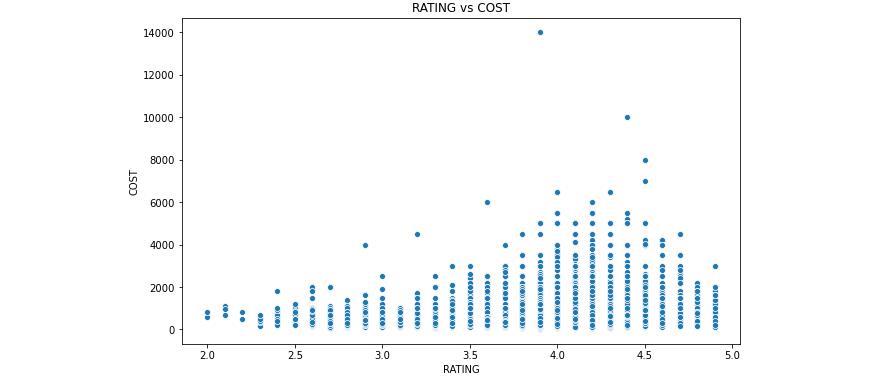


* Most Two Meal food cost is 300.

***Bivariate Analysis*** shows the graphical representation of the relationship between two features in the dataset.

Checking the relationship between RATING and COST





* The RATING between 3.9 to 4.5, the COST of food is high.

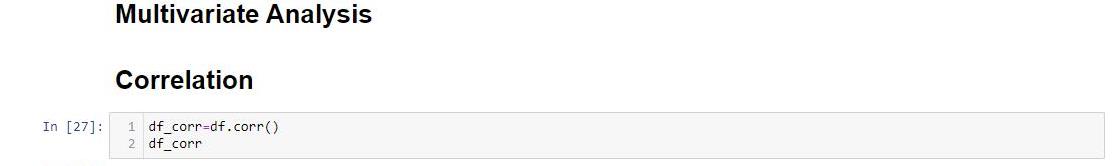
Label Encoding is used to convert categorical data into numeric.

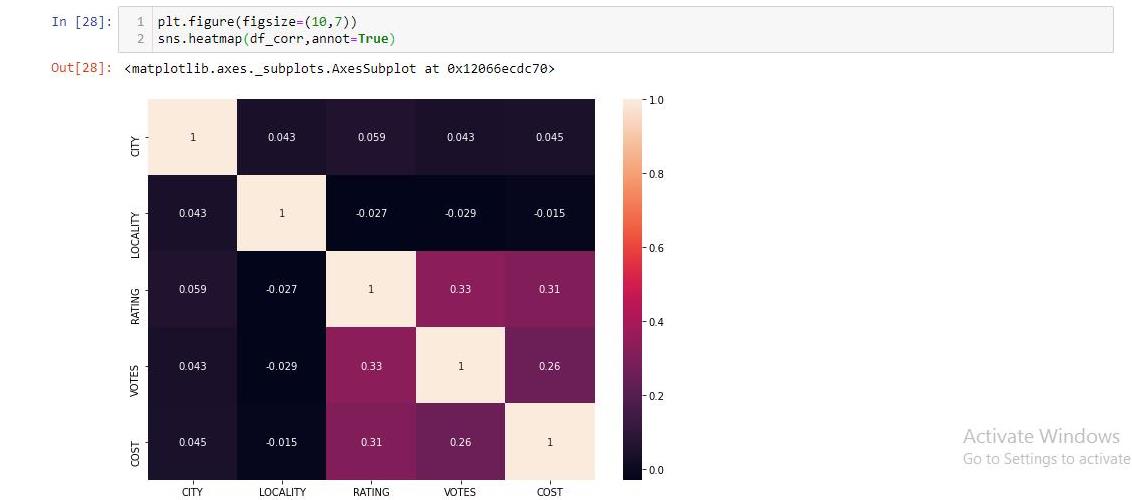
Here only CITY and LOCALITY is Label Encoded.



***Multivariate Analysis*** is done to see the relationship between all features using Correlation.

But in this case, TITLE, CUISINE and TIME have multivalues in each cell which confuses to given accurate correlation results.

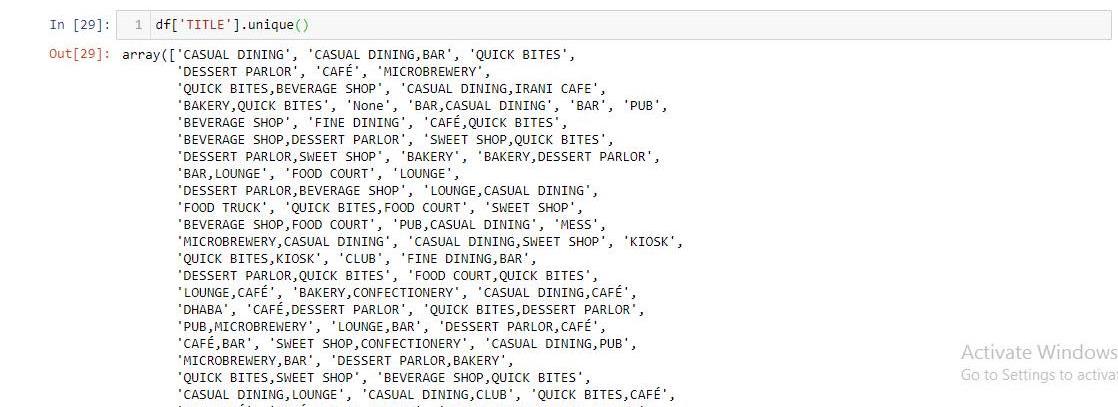


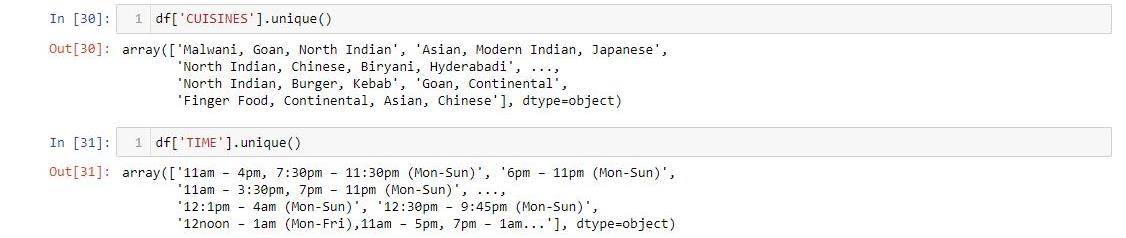


* COST is correlated with RATING and VOTES.

Extracting values from multiple value columns

First to find out the unique multiple values in each column





Splitting the multiple values



Calculating the frequency of each splitted value in each column.

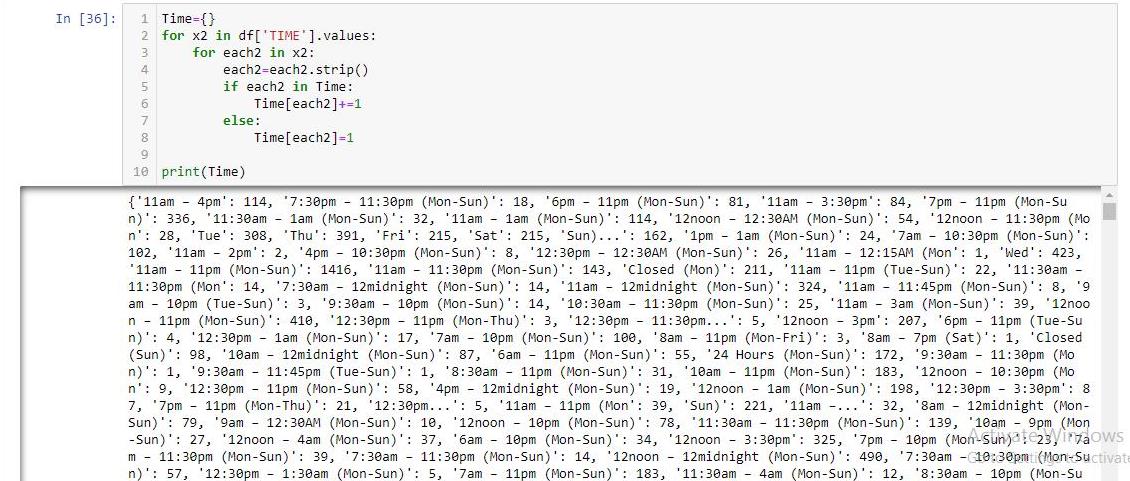
TITLE Column Value Counts:



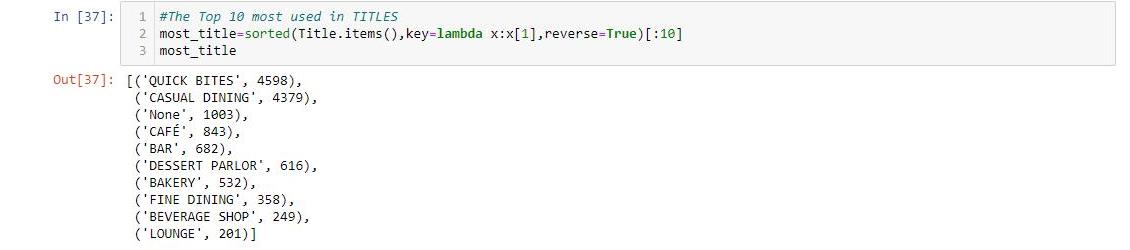
CUISINE Column Value Counts:



TIME Column Value Counts:

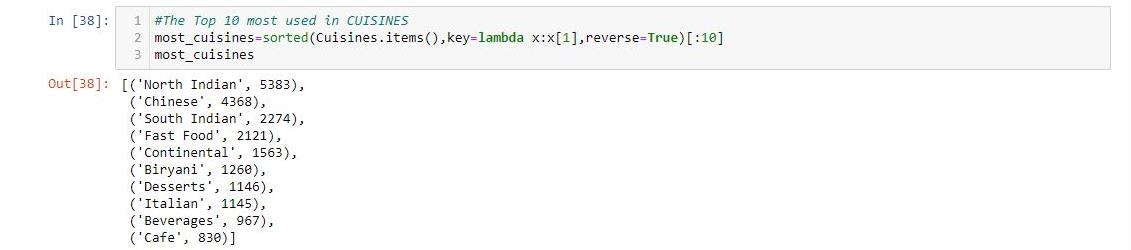


The Top 10 most used values in TITLES:



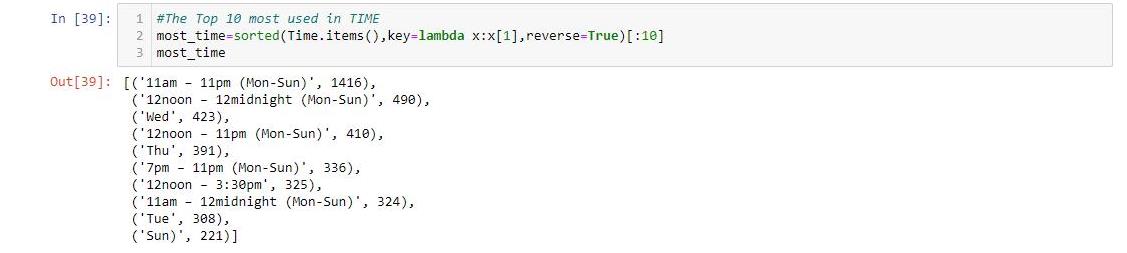
* QUICK BITES & CASUAL DINING are mostly used in TITLE Column.

The Top 10 most used values in CUISINES:



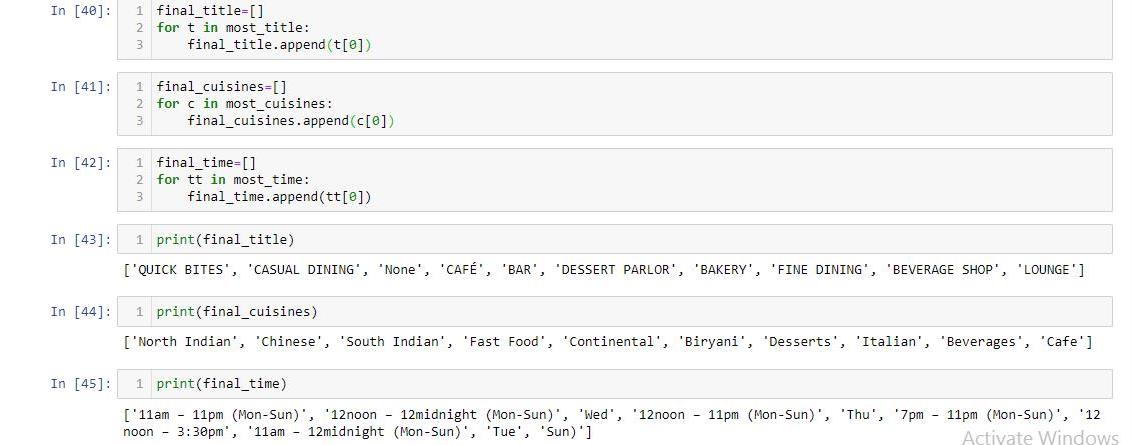
* North Indian & Chinese Cuisines are mostly used in CUISINES Column.

The Top 10 most used values in TIME:



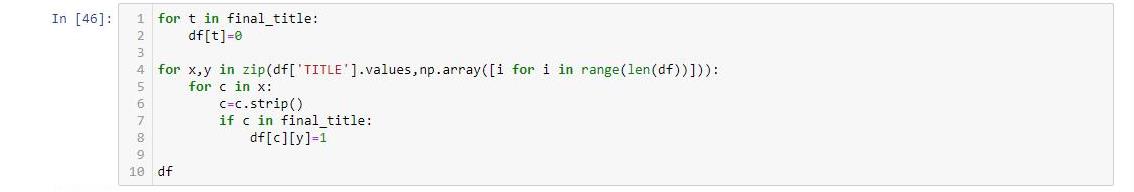
* 11am-11pm(Mon-Sun) is mostly used in TIME Column.

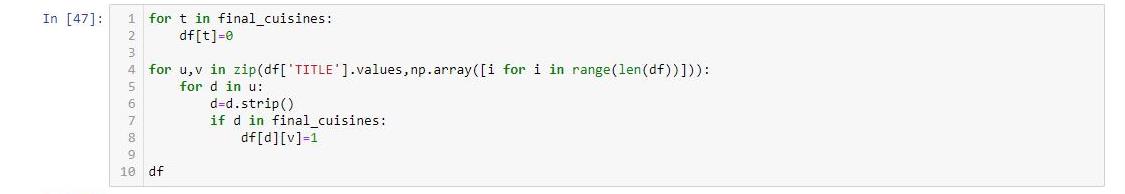
Listing those 10 Top most used values of each column separately

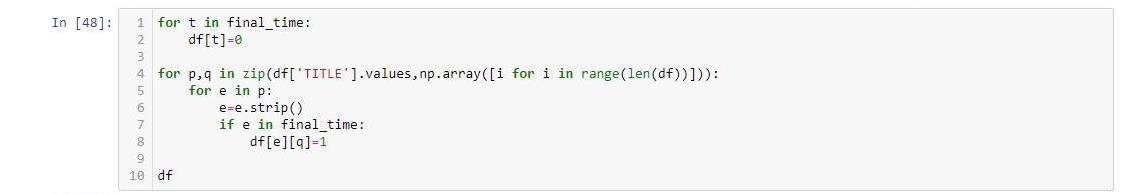


Converting each value of each column into new columns of the dataframe and placing binary values 1 and 0 wherever applicable for each restaurant.

It is one type of One Hot Encoding.







After dropping the RESTAURANT ID Column the DataFrame had 8 Columns.

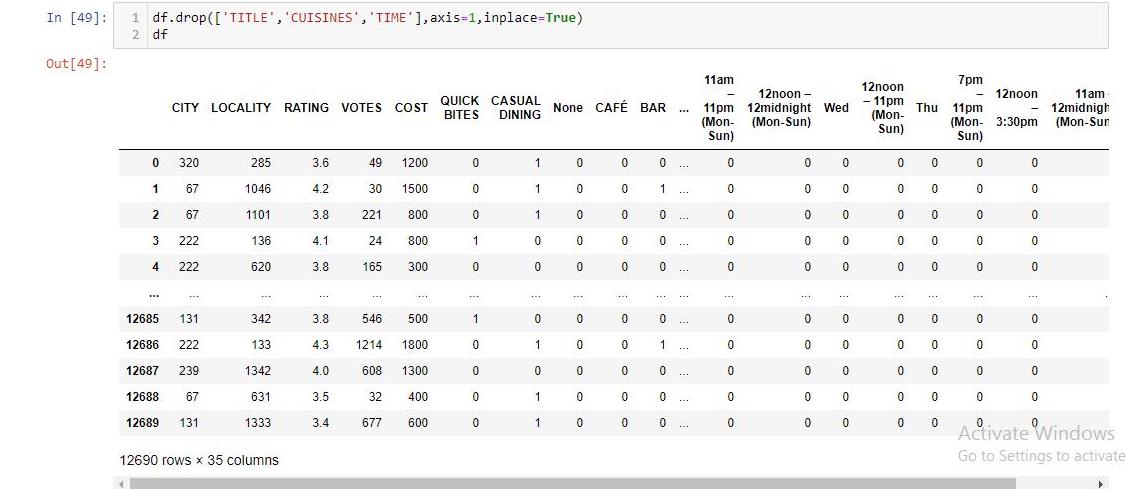
Now after this One Hot Encoding the Dataframe has 38 Columns.

10 Value Columns each from TITLE, CUISINES and TIME are added to the DataFrame.





Now, TITLE, CUISINES & TIME will be dropped as its value columns are already added to the DataFrame.

After dropping main three columns, now the no. of columns reduced to 35.

As all almost features were categorical, so need to remove **outliers** or to check **skewness**.

All the EDA process is completed. The procedure was as follows:

* **Null values were removed.**
* **Data Visualization was done including Univariate, Bivariate and Multivariate Analysis.**
* **CITY, LOCALITY Columns are Label Encoded.**
* **VOTES and RATING Column are converted to Numeric.**
* **As TITLE, CUISINES and TIME Columns had multiple values cells, All single values were extracted and the top 10 most used values for each column were converted to columns of the dataframe using 0 and 1 value as its unique value.**
* **The original TITLE, CUISINES and TIME columns are dropped.**
* **Outliers Removal or Skewness reduction could not be entertained due to high number of catergorical columns and losing data.**

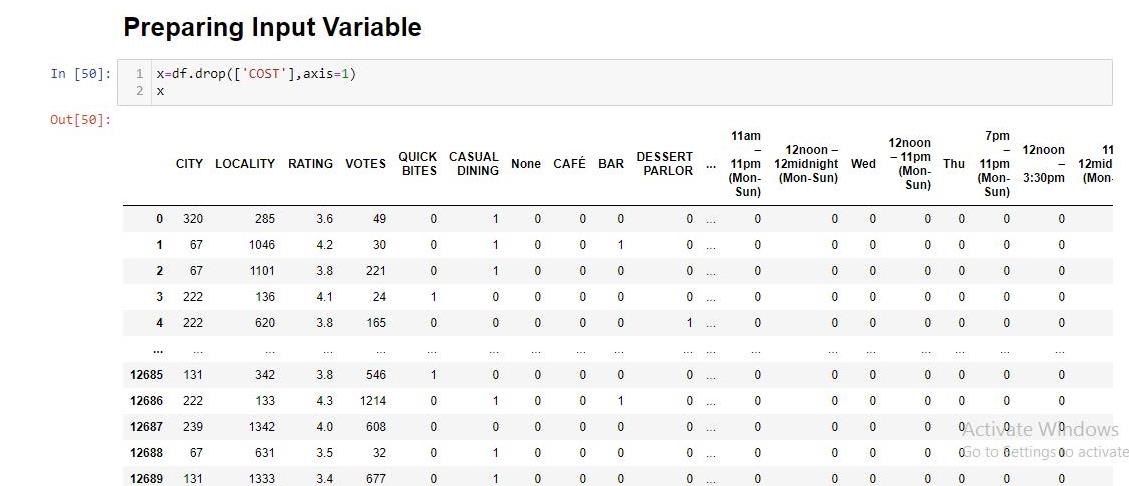
Now the df dataframe is ready for prediction model.

**Preprocessing for Building Model:**

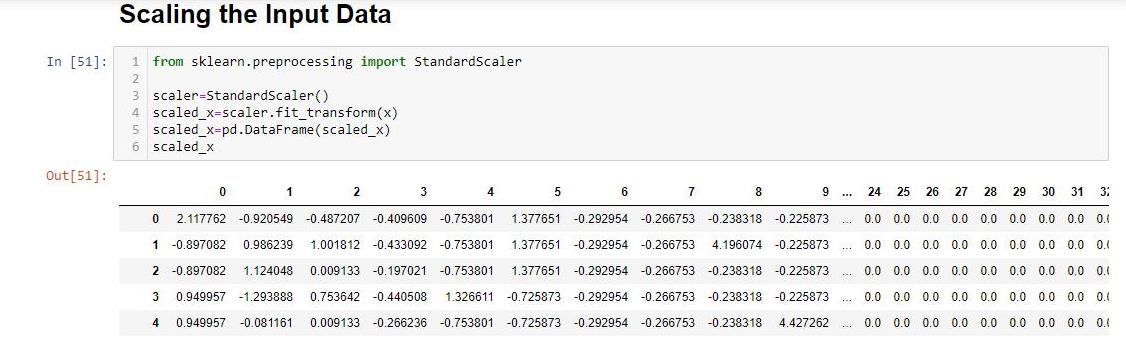
This Model will be used to predict the Food Cost.

This will be a Regression Model.

Preparing Input Variable and it is denoted as x.



Scaling the Input Data for better performance and it is denoted as scaled\_x.



Now the datas are scaled between -1 and 1.

Preparing Target Variable ie COST and it is denoted as y.



**Building a Regression Model for prediction of COST:**

Now, importing train\_test\_split, LinearRegression and r2\_score to find the best random state at which the model will perform the best.



The best random state is at 178.

All regression algorithms are imported such as **Linear Regression, KNeighbor Regression, SVR**.

Ensemble techniques such as **Random Forest Regressor, Gradient Boosting Regressor, AdaBoost Regressor** are also imported.

Regularization algorithms to avoid overfitting and underfitting such as **Lasso, Ridge, ElasticNet** are also imported.

cross\_val\_score is imported for **Cross Validation**.

**Metrics** for Regression are also imported. Such as:

**r2\_score**: It tells the accuracy of the model.

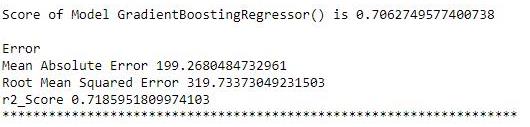
**mean\_squared\_error**: It tells the squared distances between the predicted values and actual values in the graphical representation of Input and Target Variables. The square root of mean\_squared\_error is taken as the metric value to check the performance of the model.

**mean\_absolute\_error**: It tells the average magnitudes of the distances between the predicted values and actual values in the graphical representation of Input and Target Variables.

Now, the Multiple Algorithms are used by a for loop to get the results at a glance.



The best r2\_score is obtained from GradientBoostingRegressor Model having 71.86% accuracy.

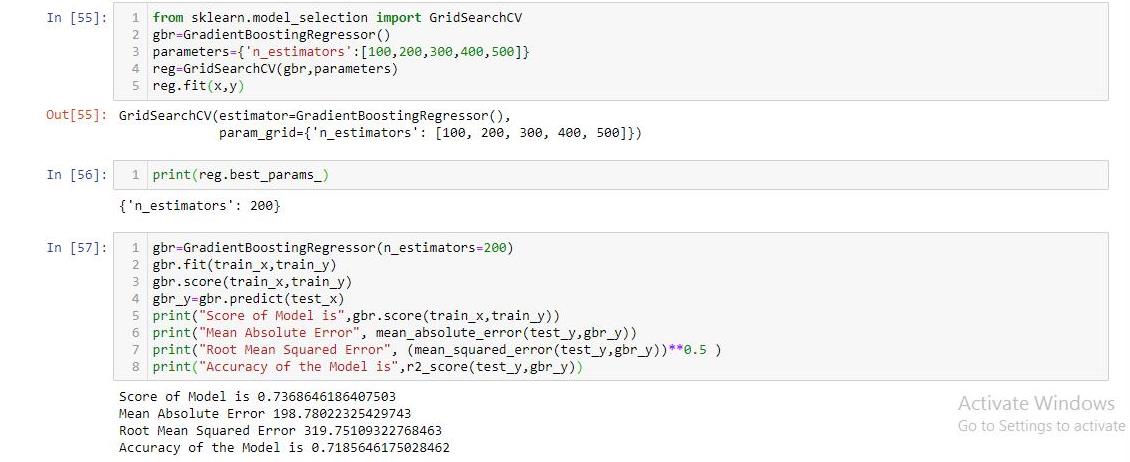


The Mean Absolute Error and Root Mean Squared Error are too high as comparing COST values.

CROSS Validation couldn’t help to improve the model.



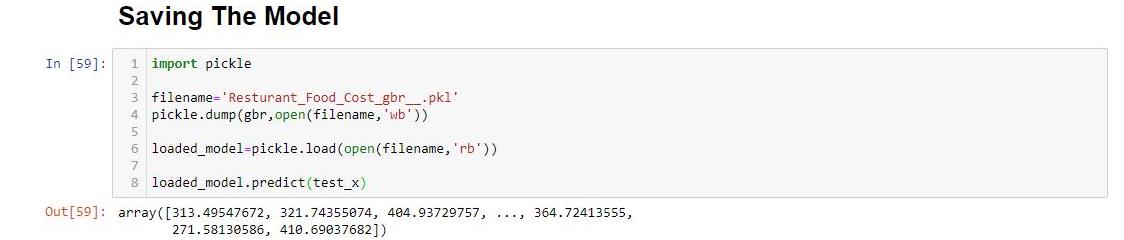
The **Hypertuning** of the best model **GradientBoostingRegressor** is done using **GridSearchCV**.



After Hypertuning the Accuracy is same as previous result ie 71.86%.

Now the Regression Model for COST prediction is ready.

Then the Model is saved.



Loading the Test Data naming as df\_test to fed into the saved Model.



Now doing the EDA of Test data just like as done in Train Data



Here, in the test there were null values. Then it is replaced with the mode of the respective columns as done in train data.

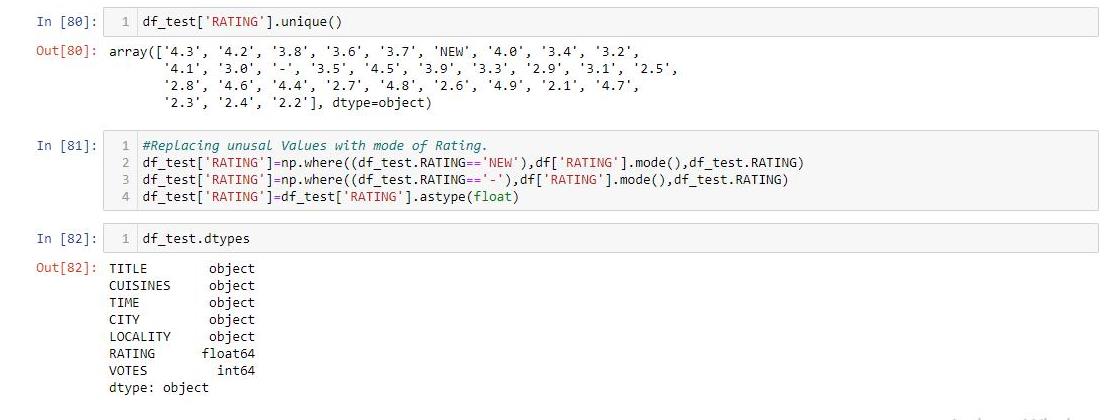
Now Dropping the ID Column ‘RESTAURANT\_ID’



Converting VOTES Column into Numeric as like done in Train Data EDA.



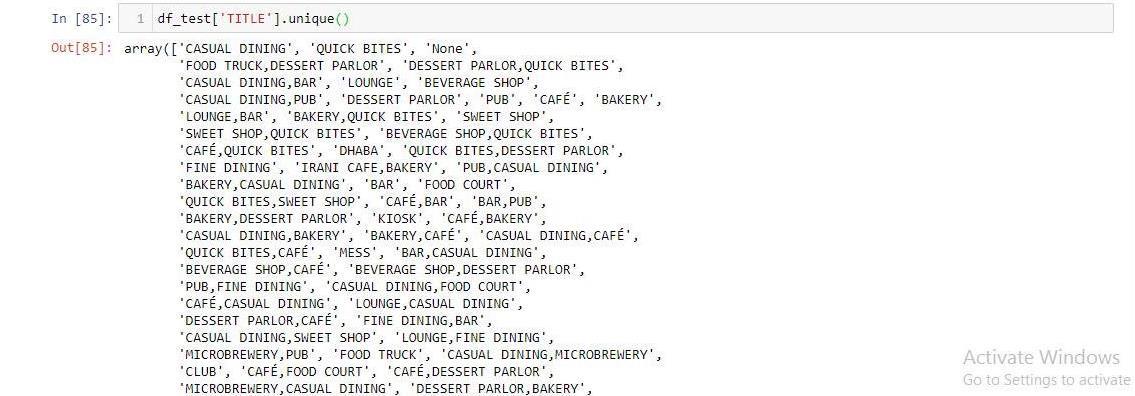
Replacing Unusual Data of RATING Column with the mode of this column and converting RATING Column into float

Label Encoding the same two columns CITY & LOCALITY



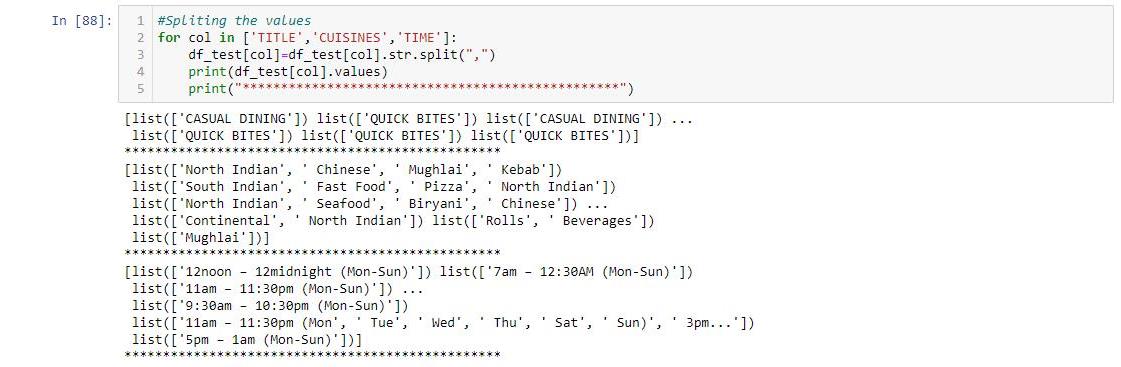
Extracting Values from Multiple Values of the columns TITLE, CUISINES and TIME

First thing to check the unique multiple values of each column as done in Train Data EDA



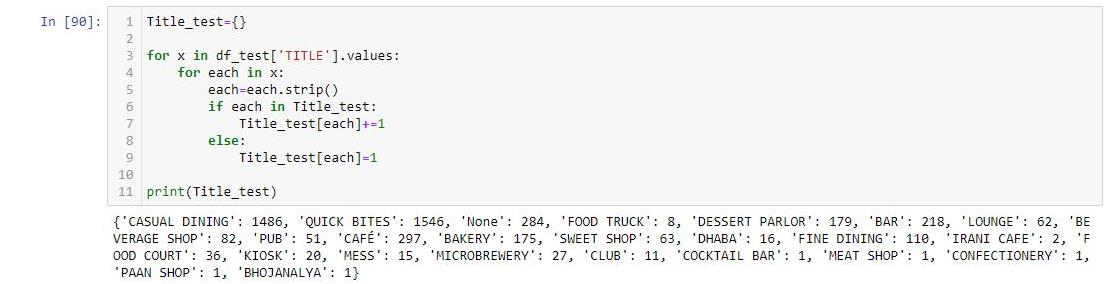


Splitting the multivalues of each column.



Calculating the frequency of each splitted value in each column.

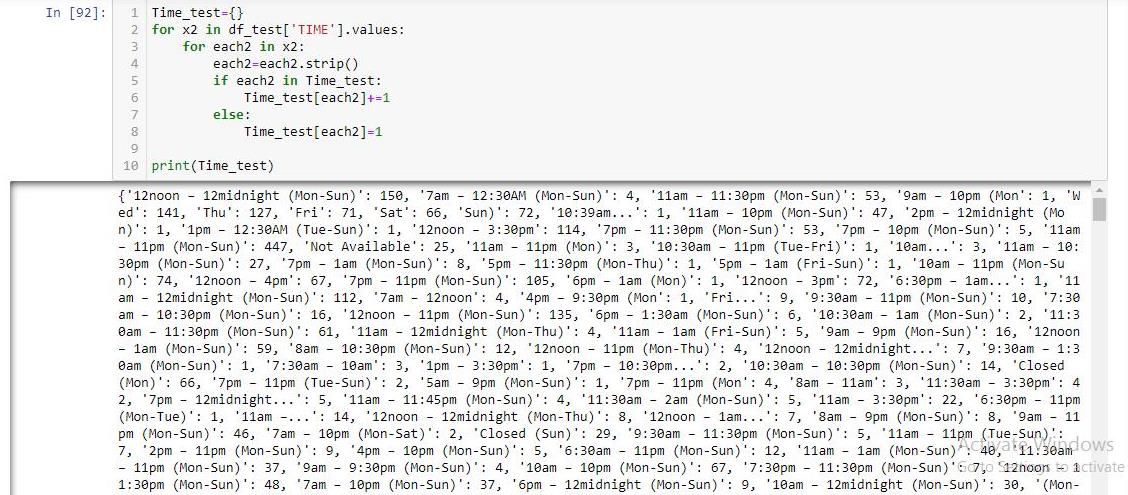
TITLE Column Value Counts:



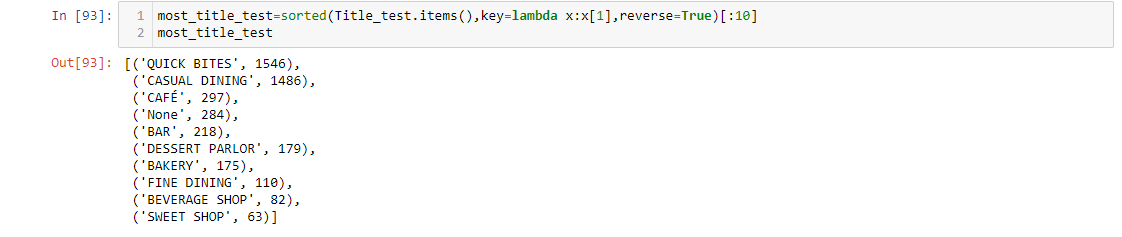
CUISINES Column Value Counts:



TIME Column Value Counts:

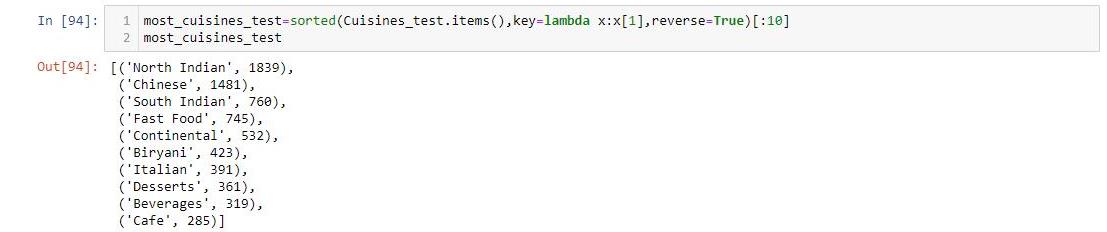


The Top 10 most used values in TITLES:



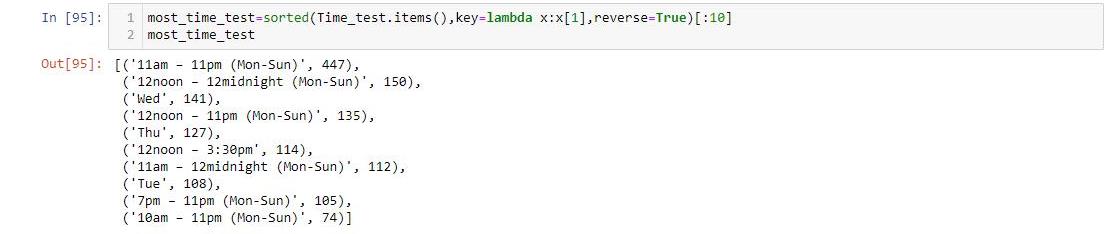
* QUICK BITES & CASUAL DINING are mostly used in TITLE Column.

The Top 10 most used values in CUISINES:



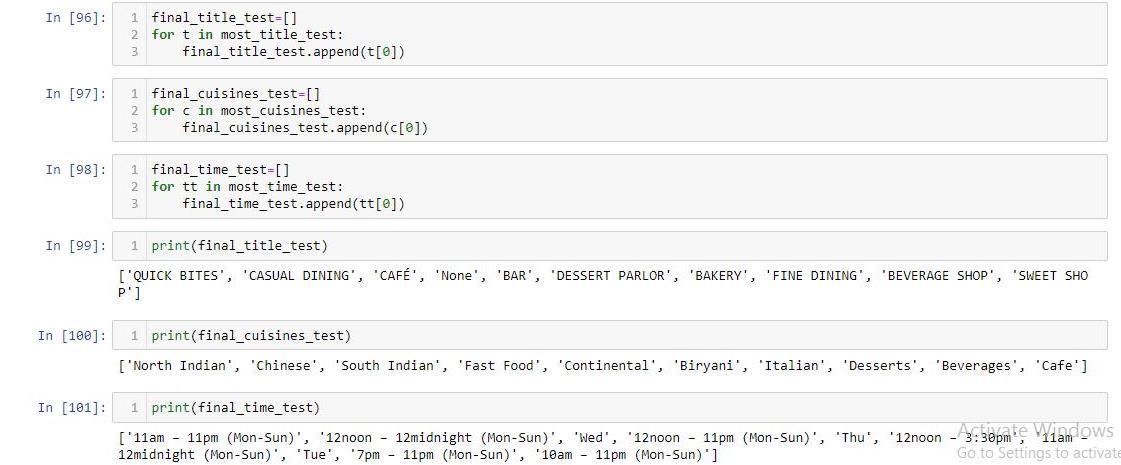
* North Indian & Chinese Cuisines are mostly used in CUISINES Column.

The Top 10 most used values in TIME:



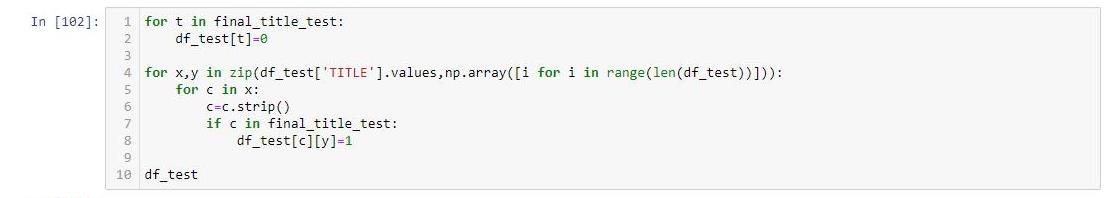
* 11am-11pm(Mon-Sun) is mostly used in TIME Column.

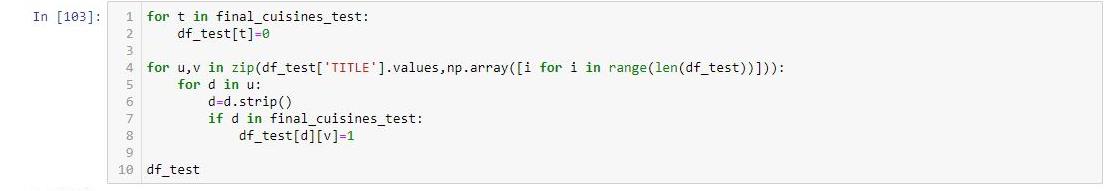
Listing the Top 10 values of each column separately.

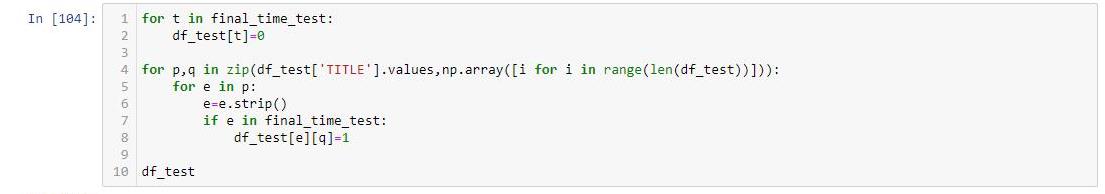


Converting each value of each column into new columns of the dataframe and placing binary values 1 and 0 wherever applicable for each restaurant.

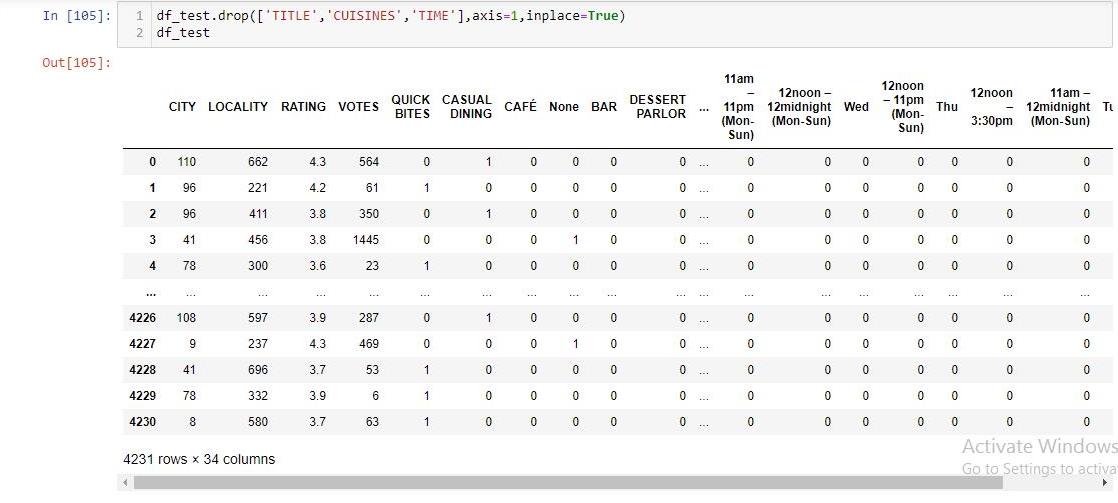
It is one type of One Hot Encoding.







Now, TITLE, CUISINES & TIME will be dropped as its value columns are already added to the DataFrame.



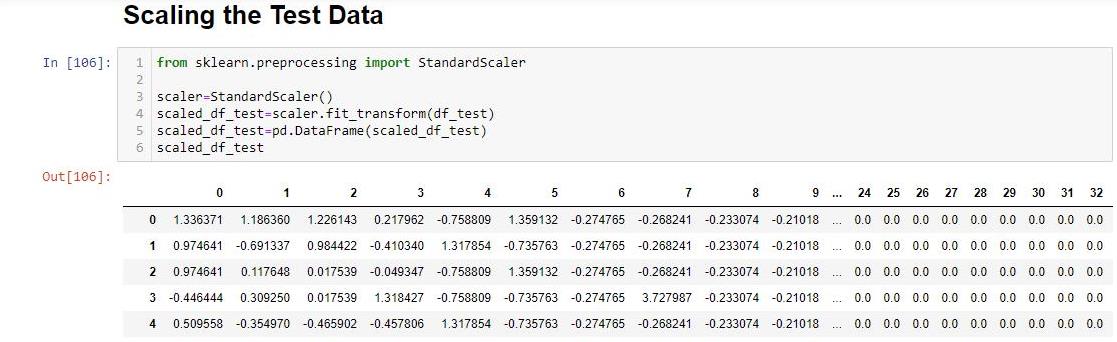
Now, df\_test has 34 columns instead of 35 columns as in Train data because there is no Target Variable.

EDA of Test Data is almost same as the Train Data including all operations.

df\_test itself is an Input data.

It will be fed to the saved model to get predicted Cost as Output.

Now scaling this df\_test. The scaled input data is denoted as scaled\_df\_test.



Now, Scaled input data is ready and to fed to the saved model and gbr is the best model as defined using Gradient Boosting Regressor by Hypertuning using GridSearchCV while saving the model.



Saving the Test Data results into a **CSV** file.



**Conclusion:**

* The Predicted result is only 71.86% accurate.
* I tried different methodologies to improve the Accuracy of the Model but it couldn’t improve.
* The Gradient Boosting Regressor is the best Model for prediction of two Meal Cost of Restaurants.

**Personal Experience:**

I noted down in this hackathon with a new challenge to extract the exact values from multiple values contained by a single cell of three columns. While Coding the confusion arised to make One Hot Encoding using One Hot Encoder library or by coding. So the EDA process was done. Though Univariate, Bivariate and Multivariate Analysis can’t be checked properly with all columns. And the Accuracy couldn’t be improved even after applying different methodologies and coding. This Hackathon came with new challenges and gave new knowledge of doing EDA and Model Building.

The Model improvement may depend on using more values of TITLE, CUISINES and TIME for One Hot Encoding. COST was also fixed with an Integer not float and there were only 86 types of COST in the entire dataset having 12690 COST values. One can try taking COST as a Categorical Variable and making a Classification Model instead. But we have to give or find out a certain Label Encoding for COST Column to convert the desired output into its original form.

**Author**

**Aditya Prasad Pattanaik**