**INTRODUCTION**

All of us must have craving for at least a few favorite food items, we may also have a few places where we like to get them, a restaurant which serves our favorite food the way we want it to be. But there is one factor that will make us reconsider having our favorite food from our favorite restaurant, the cost. Here in this hackathon, you will be predicting the cost of the food served by the restaurants across different cities in India. You will use your Data Science skills to investigate the factors that really affect the cost, and who knows maybe you will even gain some very interesting insights that might help you choose what to eat and from where.

Here we have two datasets train and test. One is Train dataset which contains 12690 rows and 9 columns while other is Test dataset which contains 4231 rows and 8 columns because it does not contain cost column as we have to predict that. Features of the columns 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.

**Task:** Regression

**Target:** Cost Column

**DATA PREPROCESSING**

**Importing Libraries**

First we will import some important libraries which are numpy as np, pandas as pd, matplotlib.pyplot as plt, seaborn as sns.

**NumPy:** NumPy stands for Numerical Python. NumPy is a Python library used for working with arrays. It also has functions for working in domain of linear algebra, fourier transform, and matrices.

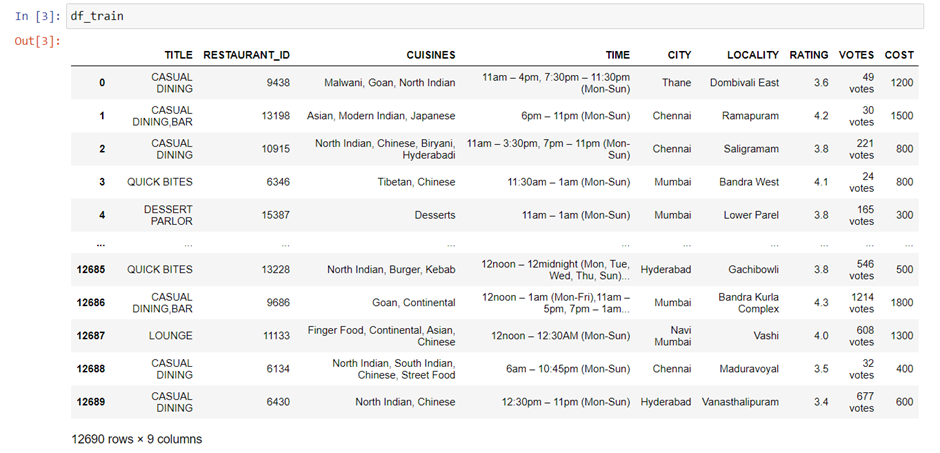
**Pandas:** The name "Pandas" has a reference to both "Panel Data", and "Python Data Analysis" used for working with data sets.It has functions for analysing, cleaning, exploring, and manipulating data.

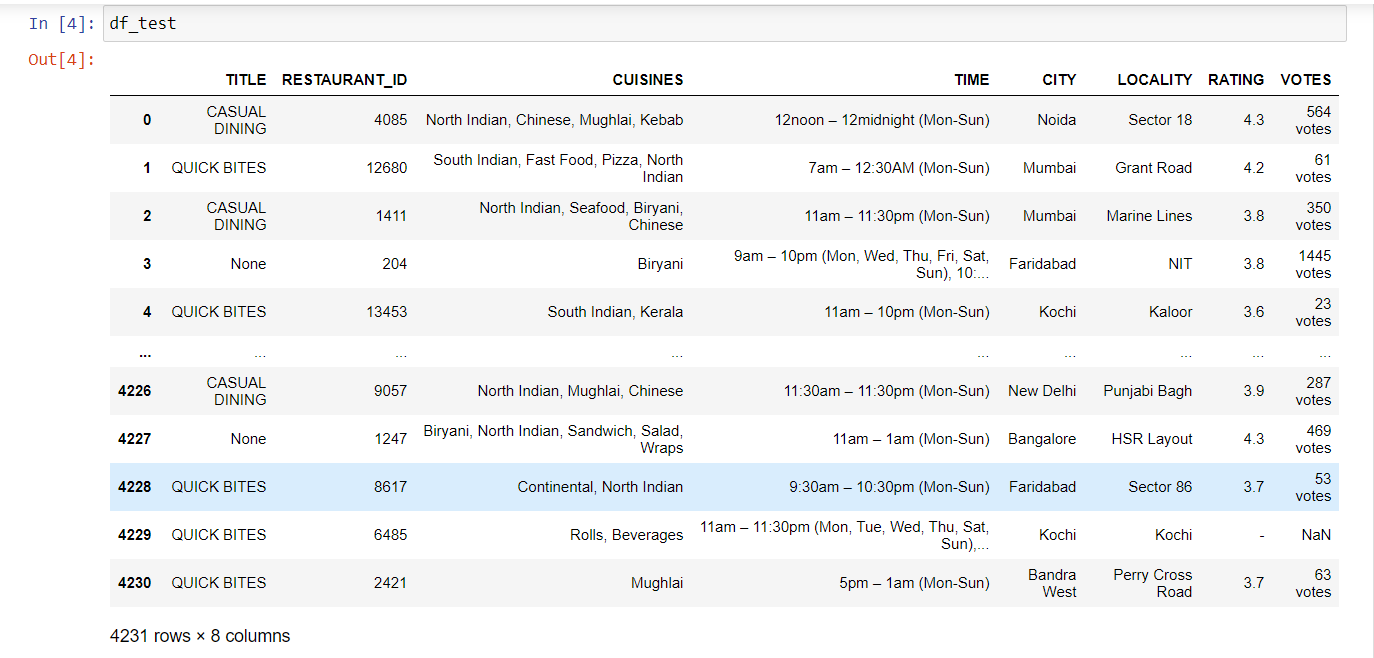
**Matplotlib:** Matplotlib is a low level graph plotting library in python that serves as a visualization utility. Most of the Matplotlib utilities lies under pyplot sub module, and are usually imported under the plt alias.

**Seaborn:** Seaborn is a library that uses Matplotlib underneath to plot graphs. It will be used to visualize random distributions.

**Loading Dataset**

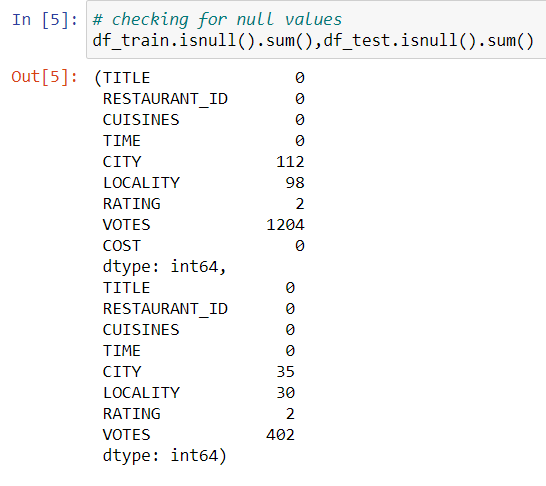
Now we are loading the datasets using pandas. Train and Test dataset are loaded as df\_train and df\_test. Take a look at the dataset...

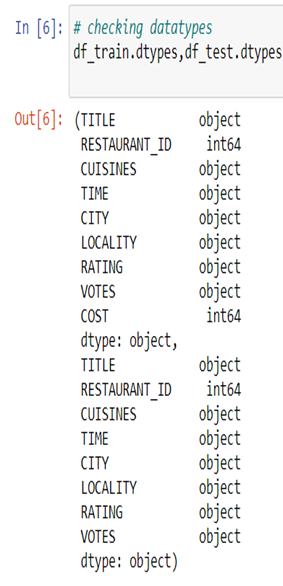




**Checking null values and datatypes**

We are checking that if our datasets contains any null values also check the datatypes and unique values of every column of datasets.





We observe that city, locality, ratings and votes columns contain null values. we also observe that except restaurant id and cost all are object columns which we need to convert into numeric. Now we do data cleaning of these columns one by one.

**Data Cleaning of VOTES Column:**

As we can see that the column contains values like “232 votes”, we remove “votes” from it by using “str.strip” syntax then only numeric value left. After that we change the column from object to numeric then fill all Nan values replacing it with “0”.

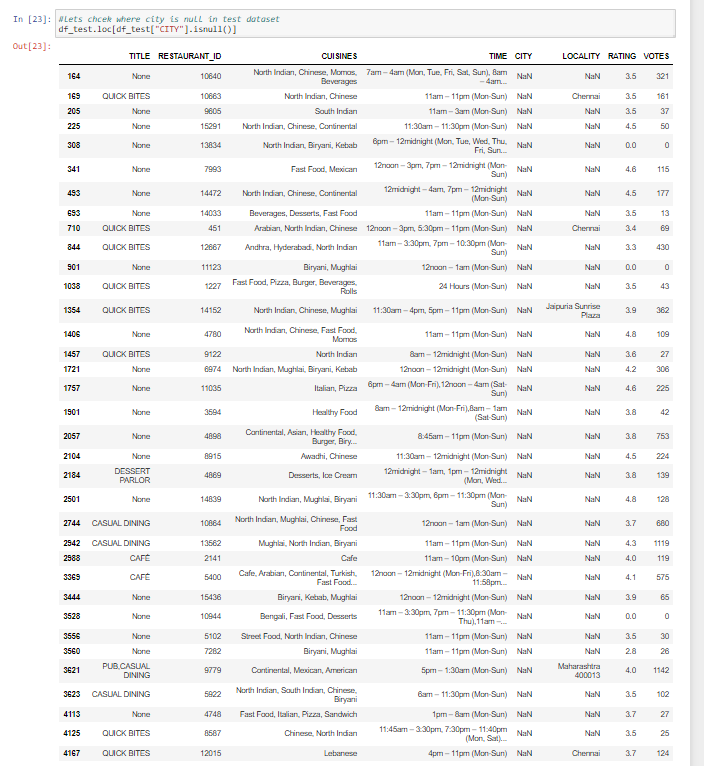
**Data Cleaning of RATINGS Column:**

After checking the value counts of ratings column we observe that it contains ratings in numeric and also contains strings “NEW” and “-”, so we are going to replace ‘NEW’ and “-“ with “0” as new indicates new restaurant which means zero ratings. We also replace Null values with “0”.

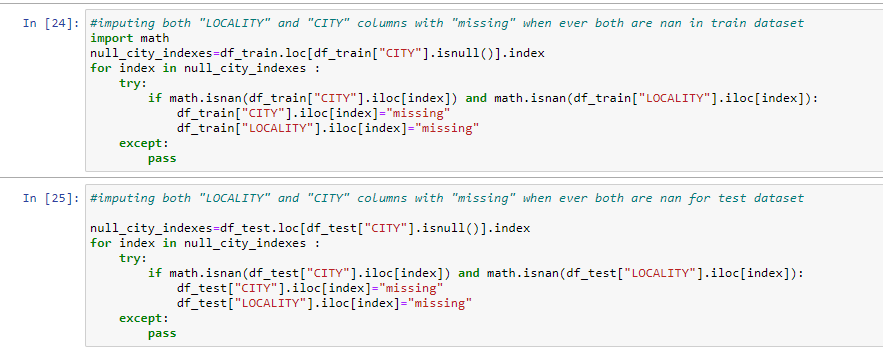
**Data Cleaning of CITY and LOCALITY Columns together**

First, we check the null values in city columns.

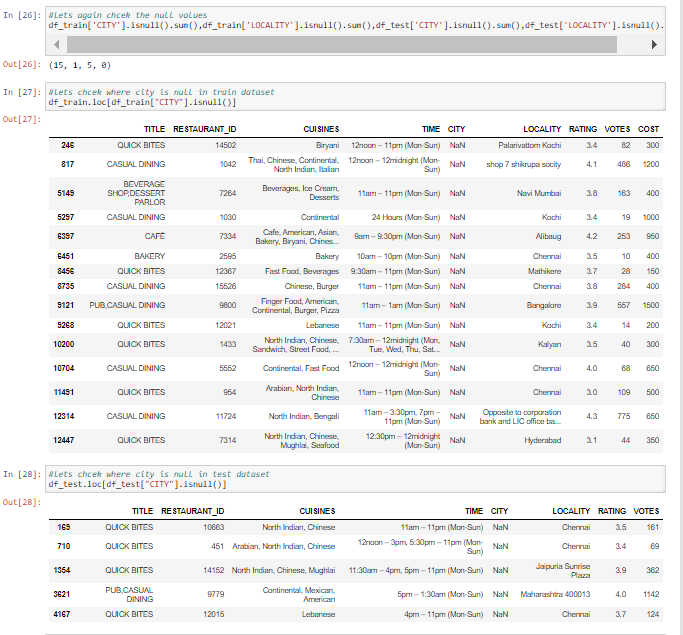




By seeing the data, we came to know in most of the rows when CITY contains null values, LOCALITY also contains null values. So we will impute both columns with "missing" whenever both we will encounter with null values. For this we import maths library than lock the dataset where city column contains null values in a variable named “null\_city\_indexes”, than apply for loop on tis variable and set if condition when both city and locality contains null when its if condition is true replace both city and locality with “missing” else pass. We do this for both train and test datasets.



Let’s check the null values again.



We observe that even after replacing null values with “missing” there are 15 rows in city and 1 row in locality in train dataset while 5 rows in city of test dataset contains null values, we also observe that most of the rows in locality city name is mentioned. So we replace these null values in city with city name mentioned in respective rows. For this we take a variable named “city\_list” which contains all the names of city already mentioned and dropped duplicate values by using “drop.duplicates” than take the another variable “null\_city\_indexes” than lock the dataset where city column contains null values. Now we apply for loop into it and split the values by space in locality column than match it with city\_list by applying if else condition in true case it will replace null value in city column with city name in locality column. We will do this on both train and test dataset separately. After that we will fill remaining null values with “missing” in both datasets.



**Data Cleaning for TITLE Column:**

We observe that title column contains multiple values so we split this column into multiple columns. We will first calculate the maximum length for this we take a variable named “max\_title” and assign value “-1” in it then apply for loop in title column and take another variable named “temp” which contains value equal to number of values in title cell by using “len” and split it by “,”now apply if condition which is true when temp is greater than max\_title and change max\_title equal to temp and print max\_title. Here max\_title is 2.

After that we make lists, number of list is equal to max\_title so here we make 2 lists t1,t2 then apply for loop for title column and use try and except, try for assign value of “temp[0]” in t1 except if doesn’t have any value by assigning “None” . Similarly for t2 try for assign value of “temp[1]” in t1 except if doesn’t have any value by assigning “None” .

At last we add two new columns t1,t2 which contains values same in list t1 and t2.



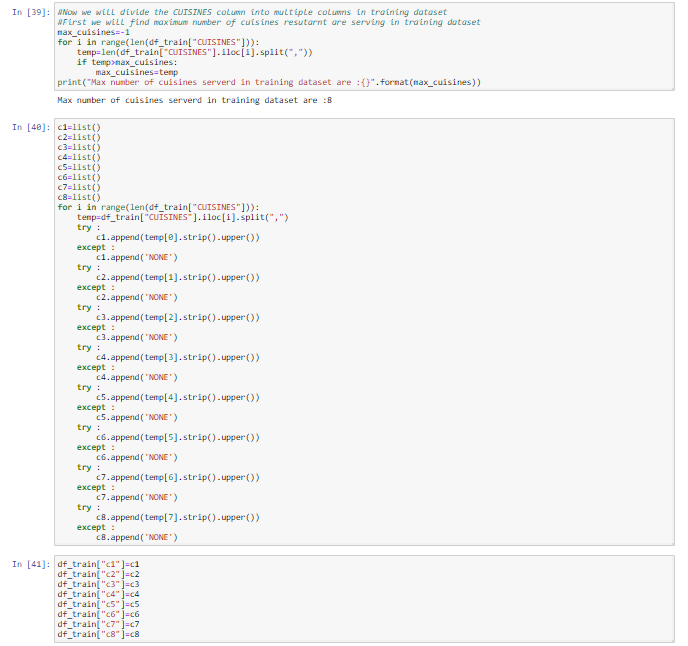
We will do the same procedure for test dataset too.

**Data Cleaning for CUISINE Column:**

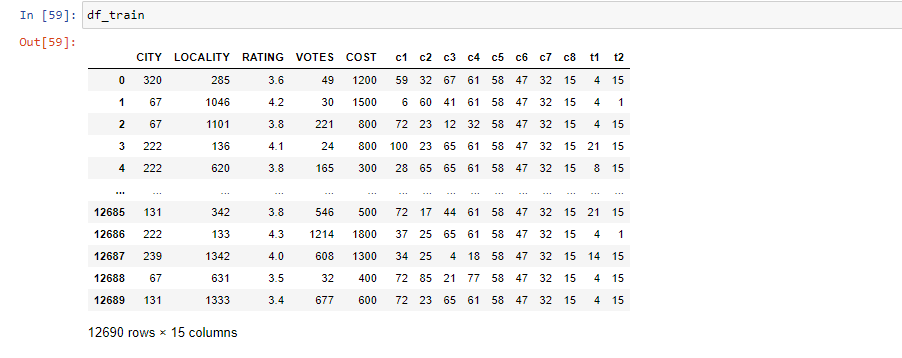
Here also we use same procedure which we used for title. We observe that Cuisine column too contains multiple values so we split this column into multiple columns. We will first calculate the maximum length for this we take a variable named “max\_cuisine” and assign value “-1” in it then apply for loop in title column and take another variable named “temp” which contains value equal to number of values in title cell by using “len” and split it by ‘,’now apply if condition which is true when temp is greater than max\_cuisine and change max\_cuisine equal to temp and print max\_cuisine. Here max\_cuisine is 8.

After that we make lists, number of list is equal to max\_title so here we make 8 lists c1,c2….c8 then apply for loop for title column and use try and except, try for assign value of “temp[0]” in t1 except if doesn’t have any value by assigning “None” . Similarly for t2 try for assign value of “temp[1]” in t1 except if doesn’t have any value by assigning “None” .We will do this till c8.

At last we 8 new columns c1,c2…c8 which contains values same in list c1, c2….c8.



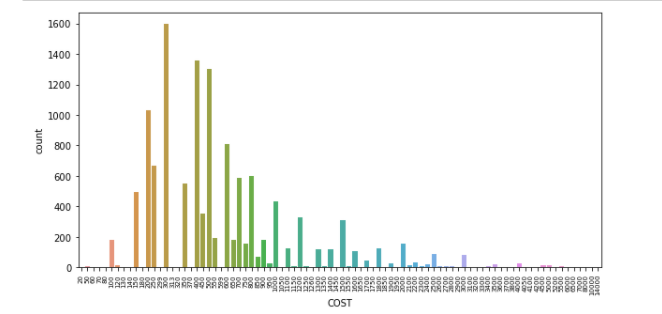
Now we will drop Cuisine , Title, Restaurant\_id and Time columns from both datasets train and test then change the remaining object columns into numeric by using Label Encoder which we have to import from sklearn library. Checking the dataset



**DATA VIZUALISATION**

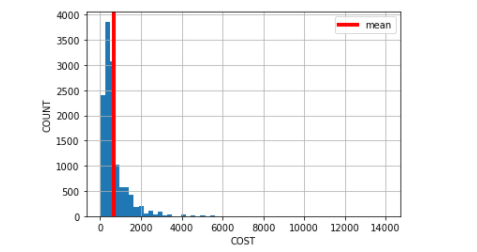
We will do some visualizations on training dataset only by using sns and plt which we already imported at the beginning.

**To check the cost column:**



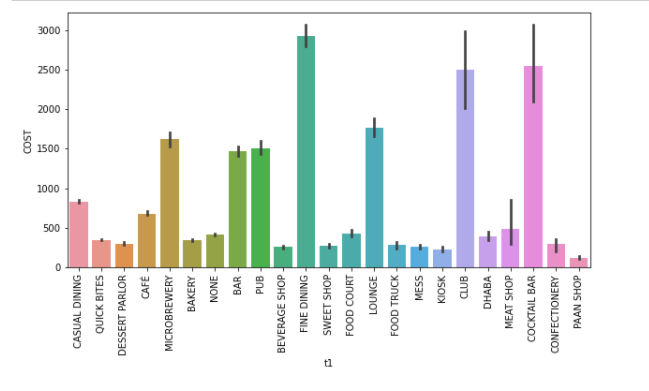
We observe that cost lies between 20 to 14000 and max no of count is of 300.

**To check mean cost:**

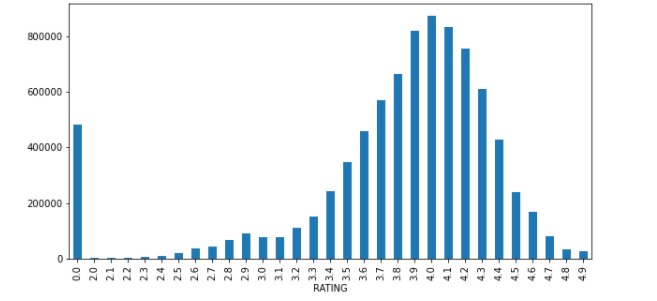


We mean cost is between 600 to 700 indicates through red line.

**To check cost via t1 column:**

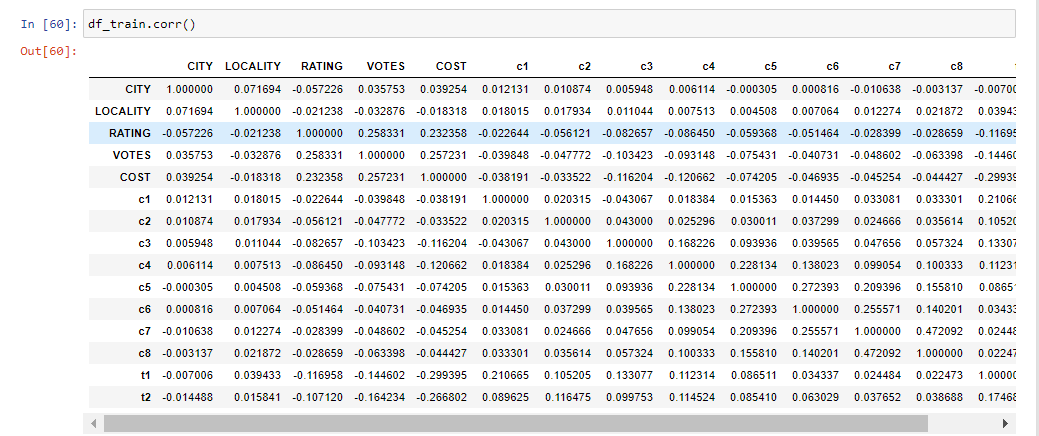


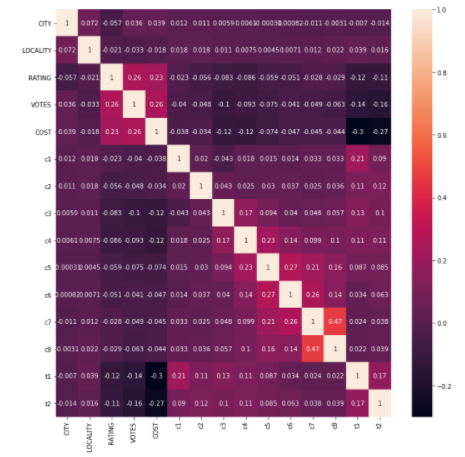
**Ratings plot by sum up the cost**

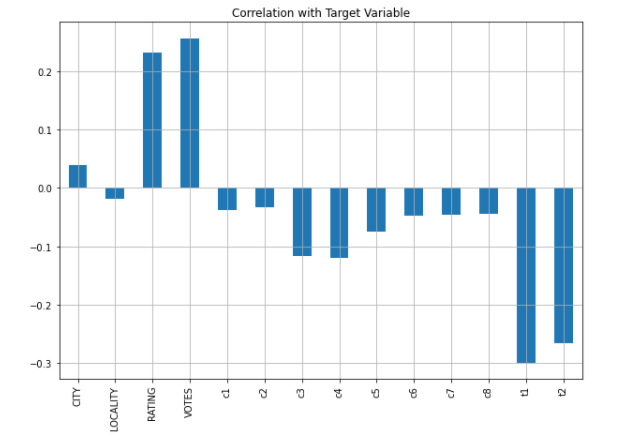


**EDA ON TRAINING DATASET**

We will check the correlation in dataset also use heatmap and visualization techniques to check it deeply.



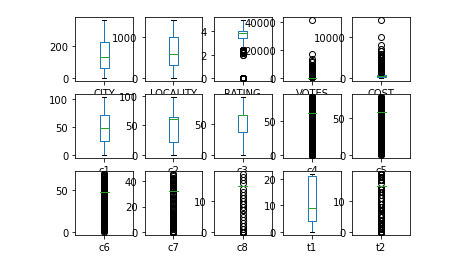




We observe that most of the columns are negatively corelated, so the highly positively correlated column is votes which is 0.26 and negatively correlated column is t1 which is -0.3.

**Checking Outliers**

Now we will check outliers by using boxplot .



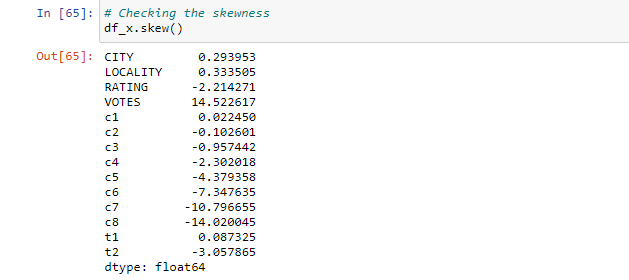
Usually, we remove outliers by using z-score but here we will not remove outliers as it causes removal of data about 70% which indicates that these are not outliers but is the property of our dataset.

**Splitting the Data**

Now we will split our data into two parts x and y where x consists of all the data used to train the model while other y contains target variable which is cost column in this dataset.

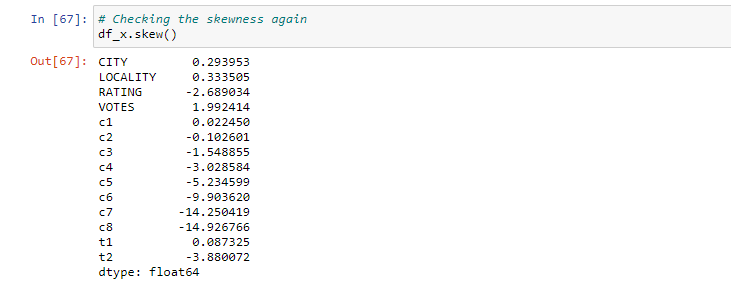
**Checking the skewness:**

We are now checking the skewness of every column. let’s see it below



**Treating the skewness:**

Skewness refers to distortion or asymmetry in a symmetrical bell curve or normal distribution of the set of data. It could be positive, negative, zero or undefined. For normal distribution skewness should be varying between -0.5 to 0.5 so we have to treat the skewness hare to fit it inti these parameters so that it would not be left skewed or right skewed. There are many methods to skewed it like log transform, square root transform, cube root transform, power transform. Here we are using square root transform method to treat the skewness greater than 0.55 and cube root transform method to treat the skewness greater than -0.55. Let’s check skewness again after treating.



We observe that some of the columns like rating, c3 c4 c5 c8 are not skewed its because of dataset some times it happened because it should be a property of dataset, so its ok we can move forward.

**Using Standard Scaler**

Now we use standard scaler to transform our data such that its distribution will have mean value “0” and standard deviation if “1”. In case of multi variate data like in our dataset this is done feature wise which means independently for each column of the data. According to distribution of the data, each value in the dataset will have the mean value subtracted, and then divided by the standard deviation of the whole dataset. We have to import the standard scaler from sklearn and then just pass the data by calling it.

**Splitting the data into train and test:**

Now we will split our data into x\_train, x\_test, y\_train, y\_test in the ratio 80%(for training) and 20%(for testing), for this we have to import train\_test split from sklearn model selection.

**Sending Data to Models:**

Now we are sending our data to various models for training and testing to check the r2 score and errors. Models used here are:

**1. Linear Regression –** The linear regression is a linear approach to modelling the relationship between a scalar response and one or more explanatory variables (also known as dependent and independent variables). The case of one explanatory variable is called simple linear regression; for more than one variable, the process is known as multiple linear regression.

**2. Lasso -** Lasso regression is a linear regression that uses shrinkage. Shrinkage is where data values are shrunk towards a mid point, like the mean. The lasso procedure encourages simple and sparse models. In machine learning model, lasso is a regression analysis method that performs variable selection and regularization in order to bring out the prediction accuracy and interpretability of the statistical model it produces.

**3. Ridge -** Ridge regression is used to create a model when the number of predictor variables in a set exceeds the number of observations in the given dataset, or when a data set has multicollinearity (correlations between predictor variables) in it.

**4. Elastic-Net -** In statistics the fitting of linear or logistic regression models, the elastic net regularizes regression method that linearly combines the L₁ and L₂ penalties of both lasso and ridge methods.

**5. SVR -** As in classification, support vector regression (SVR) is characterized by the use of kernels, sparse solution, and VC control of the margin and the number of support vectors. Support Vector Regression (SVR) is quite different than all other Regression models. The Support Vector Machine (SVM) algorithm, a classification category algorithm is used to predict a continuous variable.

**6. KNeighbors Regressor –** It is a Regression based on k-nearest neighbours. In the KNeighbours model target is predicted by local interpolation of the targets which associated to the nearest neighbours in the training set.

**7. Decision Tree Regressor** **-** Decision tree learning is one of the predictive modelling approaches used in statistics, data mining and machine learning. It uses a decision tree to go from observations about an item to conclusions about the item’s target value.

**Results:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Models** | **R2 Score** | **Mean Absolute Error** | **Mean Squared Error** | **Root Mean Squared Error** |
| Linear | 0.2085052707 | 306.4289567 | 281905.9830 | 530.948194 |
| Ridge | 0.2085289235 | 306.4270056 | 281904.6817 | 530.946985 |
| Lasso | 0.2084352994 | 306.4048731 | 281930.9046 | 530.971668 |
| Elastic-Net | 0.2017230593 | 312.1079985 | 284321.5971 | 533.218695 |
| SVR | 0.0766680093 | 301.0109051 | 328862.3444 | 573.465207 |
| K-Neighbor | 0.3043098632 | 277.9987106 | 247783.3452 | 497.778409 |
| Decision Tree | 0.2059098090 | 261.3087392 | 282830.4061 | 531.818019 |

**Using Ensemble Techniques**

**Ensemble methods** is a machine learning technique that combines several base models in order to produce one optimal predictive model. Ensemble methods are meta-algorithms that combine several machine learning techniques into one predictive model on order to decrease variance (bagging), bias (boosting), or improve predictions (stacking).

**Models Used:**

**1.Random Forest Regressor -** Random Forest uses multiple decision trees as base learning models in the dataset. Random forest is a meta estimator that fits a number of classifying decision trees on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting in the dataset. The main concept of Random Forest is to combine multiple decision trees in determining the final result rather than relying on individual decision trees.

**2.Adaboost Regressor -** An AdaBoost regressor is a meta-estimator that begins by fitting a regressor on the original dataset and then fits additional copies of the regressor on the same dataset but where the weights of instances are adjusted according to the error of the current prediction. As such, subsequent regressors focus more on difficult cases.

**3.GradientBoosting Regressor -** Gradient boosting is a [machine learning](https://en.wikipedia.org/wiki/Machine_learning) technique which produces a prediction model in the form of an [ensemble](https://en.wikipedia.org/wiki/Ensemble_learning) of weak prediction models, typically [decision trees](https://en.wikipedia.org/wiki/Decision_tree_learning). It builds the model in a stage-wise fashion like other [boosting](https://en.wikipedia.org/wiki/Boosting_(machine_learning)) methods do, and it generalizes them by allowing optimization of an arbitrary [differentiable](https://en.wikipedia.org/wiki/Differentiable_function) [loss function](https://en.wikipedia.org/wiki/Loss_function).

**Results:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Models** | **R2 Score** | **Mean Absolute Error** | **Mean Squared Error** | **Root Mean Squared Error** |
| Random Forest | 0.6656677714 | 197.4746369 | 119078.8163 | 345.077984 |
| Adaboost | -0.498248736 | 628.4108895 | 533629.9372 | 730.499058 |
| Gradient Boosting | 0.7055822740 | 198.6538378 | 104862.5029 | 323.824802 |

**Using GridSearchCV**

GridSearchCV is the process of performing hyperparameter tuning in order to determine the optimal values for a given model. The performance of a model significantly depends on the value of hyperparameters and there is no way to know in advance the best values for hyperparameters so ideally, we need to try all possible values to know the optimal values. Doing this manually could take a considerable amount of time and resources and thus we use GridSearchCV to automate the tuning of hyperparameters.

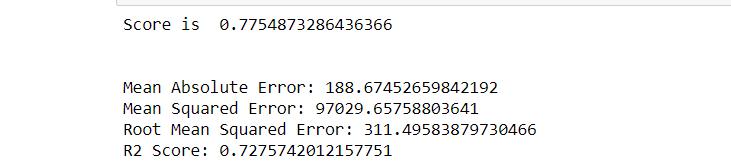
In short GridSearchCV is a library function that is a member of sklearn's model\_selection package. It helps to loop through predefined hyperparameters and fit your estimator (model) on your training set. So, in the end, you can select the best parameters from the listed hyperparameters.

Here we find the best parameters for Gradient boosting as it gives max R2 score with less errors. So best parameters are:

n\_estimators = 300 and criterion: friedman\_mse.

**Best Model**

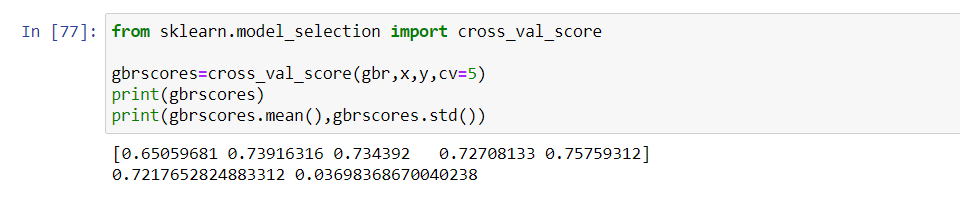
From above w have calculated the best parameters for gradient boosting regressor we will apply that parameters to get best result of Gradient Boosting Regressor. Result after applying is below:



**Cross validation**

Cross validation helps to find out the over fitting and under fitting of the model. In the cross validation the model is made to run on different subsets of the dataset which will get multiple measures of the model. If we take 5 folds, the data will be divided into 5 pieces where each part being 20% of full dataset. While running the Cross validation the 1 st part (20%) of the 5 parts will be kept out as a hold out set for validation and everything else is used for training data. This way we will get the first estimate of the model quality of the dataset. In the similar way further iterations are made for the second 20% of the dataset is held as a hold out set and remaining 4 parts are used for training data during process. This way we will get the second estimate of the model quality of the dataset. These steps are repeated during the cross validation process to get the remaining estimate of the model quality.

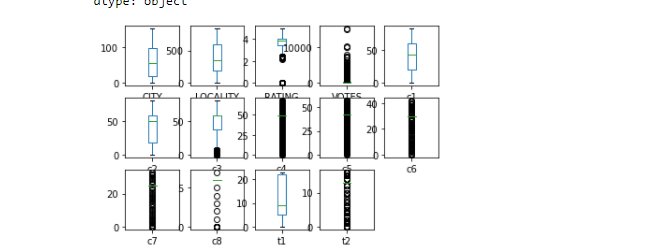
Below is our cross validation result:



**EDA On Test Dataset**

**Checking Outliers**

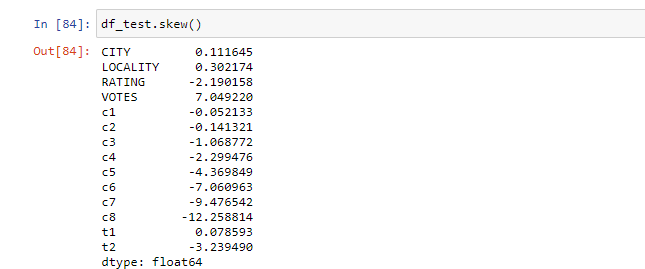
Now we will check outliers by using boxplot .



Usually, we remove outliers by using z-score but here we will not remove outliers as it causes removal of data about 70% which indicates that these are not outliers but is the property of our dataset.

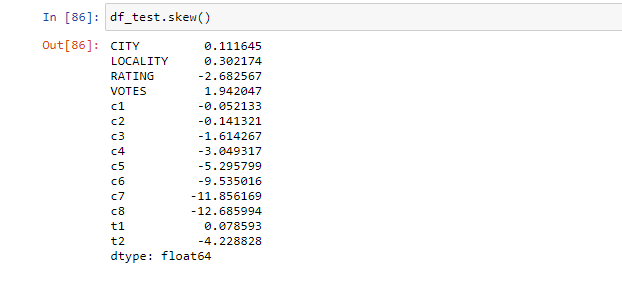
**Checking the skewness:**

We are now checking the skewness of every column which means distribution of each column. let’s see it below



**Treating the skewness**

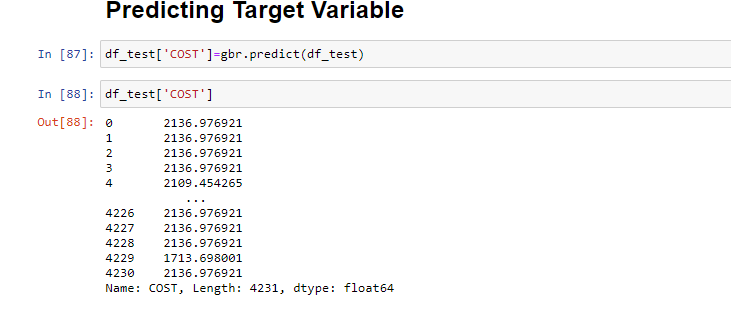
We have treated the skewness using square root transform method to treat the skewness greater than 0.55 and cube root transform method to treat the skewness greater than -0.55. Let’s check skewness again after treating.



We observe that some of the columns like rating, c3 c4 c5 c8 are not skewed its because of dataset some times it happened because it should be a property of dataset, so its ok we can move forward.

**Predict the target variable**

From above we know that gradient boosting regressor is the best model to predict the target variable as it gives high R2 score and less errors compared to others so we use gbr.predict(x) to predict.



**CONCLUSION**

We conclude that dataset contains null values which we have imputed also splits two column into many columns then after doing EDA we have sent our data to models also used ensemble techniques and get to know that the best model for prediction is Gradient Boosting Regressor which we used for prediction in test dataset.