**Predicting Restaurant Food Cost**

Who doesn’t love food? All of us must have craving for at least a few favourite food items, we may also have a few places where we like to get them, a restaurant which serves our favourite food the way we want it to be. But there is one factor that will make us reconsider having our favourite food from our favourite restaurant, the **cost**. 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 our Data Science skills to investigate the factors that really affect the cost, and who knows maybe We will even gain some very interesting insights that might help us choose what to eat and from where.

So, Let’s start:

Some Information Of Project:-

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

#### **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.

**First step:-**

We need to Import the libraries



Pandas is a software library written for the Python programming language for data manipulation and analysis. In particular, it offers data structures and operations for manipulating numerical tables and time series.

NumPy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays.

The **OS module in Python** provides functions for interacting with the operating system. **OS** comes under **Python's** standard utility **modules**. This **module** provides a portable way of using operating system-dependent functionality. The \***os**\* and \***os**. path\* **modules** include many functions to interact with the file system.

**Warning messages** are displayed by **warn**() function defined in '**warning**' module of **Python's** standard library. **Warning** is actually a subclass of Exception in built-in class hierarchy.

**Second Step:-**

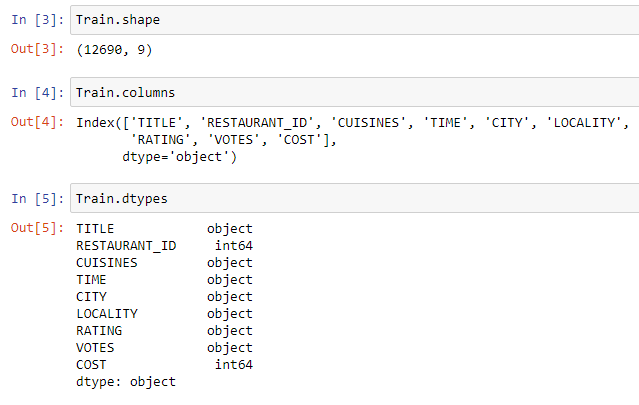
****

We will be using Train as instance for the training dataset.

**Pd.read\_csv** states that it will fetch the file from the given location in the brackets ( ).

**Train.head()** will show the first five rows of the dataset and if the person want to check the last five rows of the dataset the he or she ca use **Train.tail()**

**Third Step:-**

****

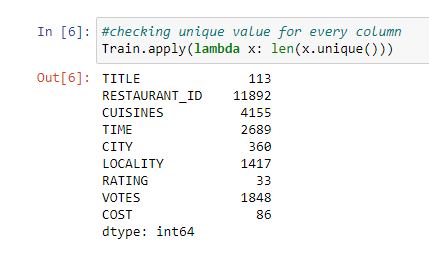
**Train.shape** will show you the number of rows and column in the data set.

Where 12690 represents the rows and 9 represents the columns.

**Train.columns** will show you the all names of the columns.

**Train.dtypes** will show you the datatypes of the columns.

**Fourth step:**



**Train.apply(lambda x: len(x.unique()))** will show the unique value present in every column.

**EDA**

Exploratory data analysis (EDA) is used by data scientists to analyze and investigate data sets and summarize their main characteristics, often employing data visualization methods. It helps determine how best to manipulate data sources to get the answers you need, making it easier for data scientists to discover patterns, spot anomalies, test a hypothesis, or check assumptions.

EDA is primarily used to see what data can reveal beyond the formal modeling or hypothesis testing task and provides a provides a better understanding of data set variables and the relationships between them. It can also help determine if the statistical techniques you are considering for data analysis are appropriate. Originally developed by American mathematician John Tukey in the 1970s, EDA techniques continue to be a widely used method in the data discovery process today.

**Importance of EDA**

The main purpose of EDA is to help look at data before making any assumptions. It can help identify obvious errors, as well as better understand patterns within the data, detect outliers or anomalous events, find interesting relations among the variables.

Data scientists can use exploratory analysis to ensure the results they produce are valid and applicable to any desired business outcomes and goals. EDA also helps stakeholders by confirming they are asking the right questions. EDA can help answer questions about standard deviations, categorical variables, and confidence intervals. Once EDA is complete and insights are drawn, its features can then be used for more sophisticated data analysis or modeling, including [machine learning](https://www.ibm.com/cloud/learn/machine-learning).

There are four primary types of EDA:

Univariate non-graphical. This is simplest form of data analysis, where the data being analyzed consists of just one variable. Since it’s a single variable, it doesn’t deal with causes or relationships. The main purpose of univariate analysis is to describe the data and find patterns that exist within it.

Univariate graphical. Non-graphical methods don’t provide a full picture of the data. Graphical methods are therefore required. Common types of univariate graphics include:

Stem-and-leaf plots, which show all data values and the shape of the distribution.

Histograms, a bar plot in which each bar represents the frequency (count) or proportion (count/total count) of cases for a range of values.

Box plots, which graphically depict the five-number summary of minimum, first quartile, median, third quartile, and maximum.

Multivariate nongraphical: Multivariate data arises from more than one variable. Multivariate non-graphical EDA techniques generally show the relationship between two or more variables of the data through cross-tabulation or statistics.

Multivariate graphical: Multivariate data uses graphics to display relationships between two or more sets of data. The most used graphic is a grouped bar plot or bar chart with each group representing one level of one of the variables and each bar within a group representing the levels of the other variable.

Other common types of multivariate graphics include:

Scatter plot, which is used to plot data points on a horizontal and a vertical axis to show how much one variable is affected by another.

Multivariate chart, which is a graphical representation of the relationships between factors and a response.

**Fifth step:**

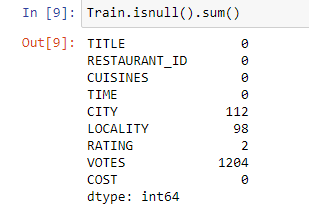


**Seaborn** is a **library** for making statistical graphics in Python.**Seaborn** helps you explore and understand your data. Its plotting functions operate on dataframes and arrays containing whole datasets and internally perform the necessary semantic mapping and statistical aggregation to produce informative plots.

**Matplotlib** is a plotting library for the Python programming language and its numerical mathematics extension NumPy. It provides an object-oriented API.

Above Graph shows you the null value present in the column on the X-axis we can observe the columns the whie dash in the graphs represents the null.

**Sixth step:**



Train.isnull().sum() will show the null values in columns.

As we can observe there are four columns which have null values.

So, we need to use mean,median or mode to fix that

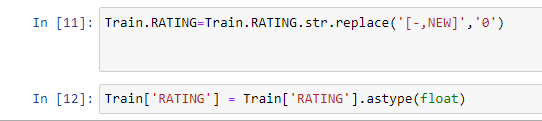
If column’s datatype is categorical use **Mode**.

If it has continuous datatype of value then follow this:-

**Mean**:- If data is normal distributed(Without outliers)

**Median**- If data is not normal distributed(With outliers)

**7 Step:**

****

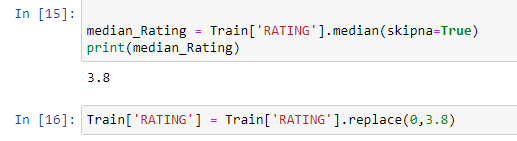
Input [11] will remove the unnecessary strings from the column like – , NEW and replace it with 0.

IN[12] Will change the datatype from object to float

We need to change the datatype because further we will be applying Linear Regression

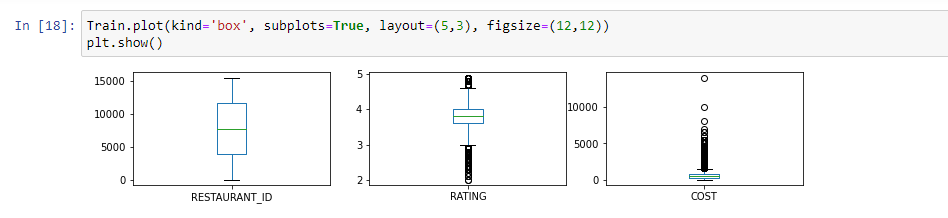
And it doesn’t support object datatype.

**8 Step:**



In[15] will give you the median of the column RATING and in second line of code we will be replacing 0 with the median value.

**9 step:**



This code describes the outliers present in the columns.

**10 step:**

same we will be doing with VOTES column.

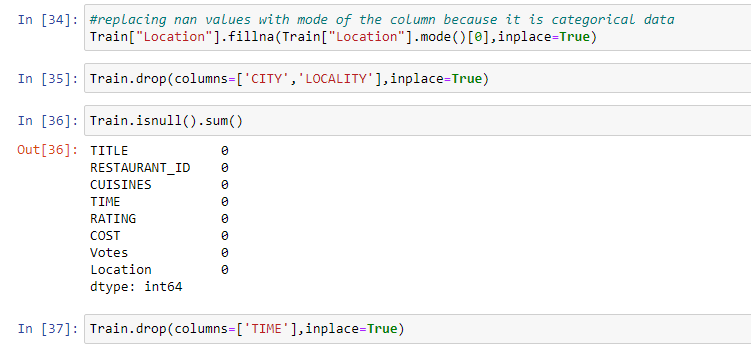
**11 step-**



Here we are concatenating the two columns because both are elated to each other and

City is having more null values compared to Locality.

**12 step**



In[34]

This line of code will replace the 0 value with mode.

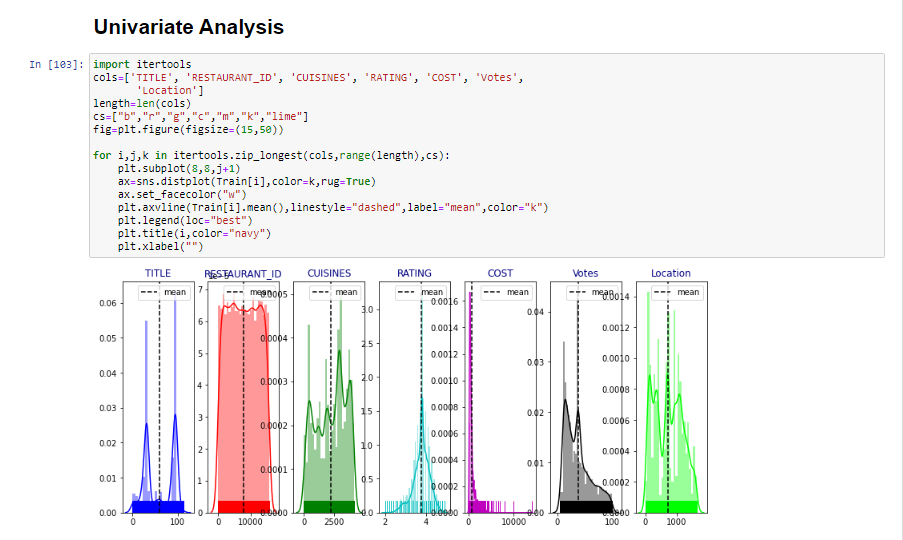
In[35]

Now we will be dropping this column because had already combined City and Locality.

In[36]

This shows all the null values has been removed.

**13 Step-**



Above code will help us to show the skewness of al the columns one by one.

Cs describe the instances colours which will be used in the graphs.

Figsize is for the size of the column.

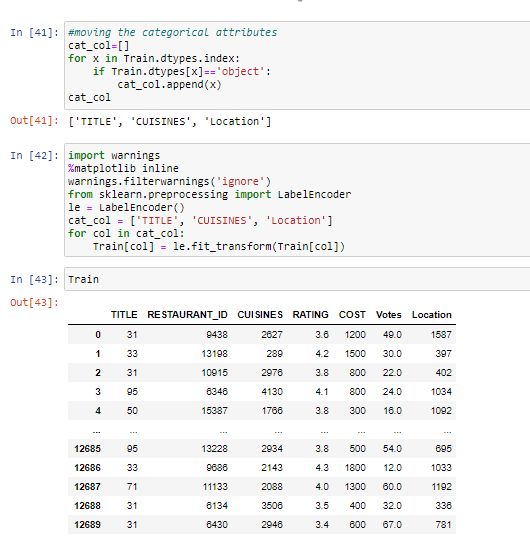
**14 Step:**



This Shows the correlation between the column that Target column has strongest bond with which column and least with which.

This also helps you to understand which column we can drop.

**15 Step-**

****

In[41]

This line of code will move the object columns in one for instance so that we can easily use it

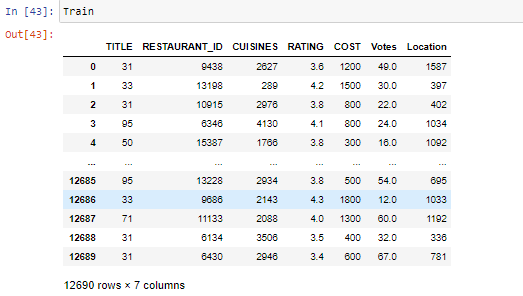
Further to change it in Numerical format.

In[42]

Here we are using **LableEncoder**

**Label Encoding** refers to converting the labels into numeric form so as to convert it into the machine-readable form. Machine learning algorithms can then decide in a better way on how those labels must be operated. It is an important pre-processing step for the structured dataset in supervised learning.

As we can see columns having categorical value now has been changed to Numerical.



**16 step-**

****

X means the value which will gives us the result and Y depend on the values of x.

Now we had use **Standard Scaler**. **StandardScaler** removes the mean and scales each feature/variable to unit variance. This operation is performed feature-wise in an independent way. **StandardScaler** can be influenced by outliers (if they exist in the dataset) since it involves the estimation of the empirical mean and standard deviation of each feature.

**17 Step:-**

Now, we will be importing libraries to do the Machine Learning part so that our model could learn the data that means we can Train the data.

In[47]

This line of code will give us the best state to use.

In[48]

This line of code will seprate the data to Train.

**Step 18:**



Here we are using Different model Lineaer Regression and Random Forest Regressor.

**Linear regression** attempts to model the relationship between two variables by fitting a linear equation to observed data. One variable is considered to be an explanatory variable, and the other is considered to be a dependent variable. For example, a modeler might want to relate the weights of individuals to their heights using a linear regression model.

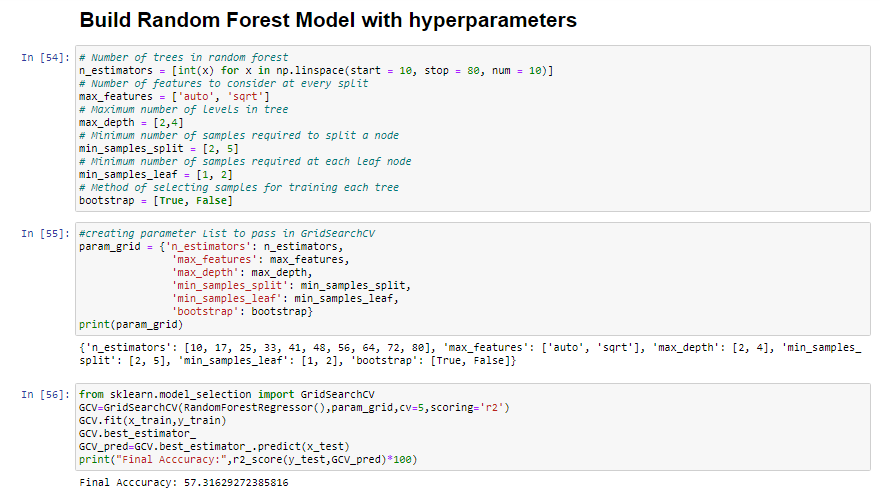
Before attempting to fit a linear model to observed data, a modeler should first determine whether or not there is a relationship between the variables of interest. This does not necessarily imply that one variable *causes* the other (for example, higher SAT scores do not *cause* higher college grades), but that there is some significant association between the two variables. A [scatterplot](http://www.stat.yale.edu/Courses/1997-98/101/scatter.htm) can be a helpful tool in determining the strength of the relationship between two variables. If there appears to be no association between the proposed explanatory and dependent variables (i.e., the scatterplot does not indicate any increasing or decreasing trends), then fitting a linear regression model to the data probably will not provide a useful model. A valuable numerical measure of association between two variables is the [correlation coefficient](http://www.stat.yale.edu/Courses/1997-98/101/correl.htm), which is a value between -1 and 1 indicating the strength of the association of the observed data for the two variables.

A linear regression line has an equation of the form ***Y = a + bX***, where ***X*** is the explanatory variable and ***Y*** is the dependent variable. The slope of the line is ***b***, and ***a*** is the intercept (the value of ***y*** when ***x*** = 0).

**Random Forest Regression** is a supervised learning algorithm that uses **ensemble learning** method for regression. Ensemble learning method is a technique that combines predictions from multiple machine learning algorithms to make a more accurate prediction than a single model.

**AdaBoost** is best **used** to boost the performance of decision trees on binary classification problems. **AdaBoost** was originally called **AdaBoost**. M1 by the authors of the technique Freund and Schapire. More recently it may be referred to as discrete **AdaBoost** because it is **used** for classification rather than regression.

**Step 19**

****

Here we will be using GridsearchCV for hypertuning.

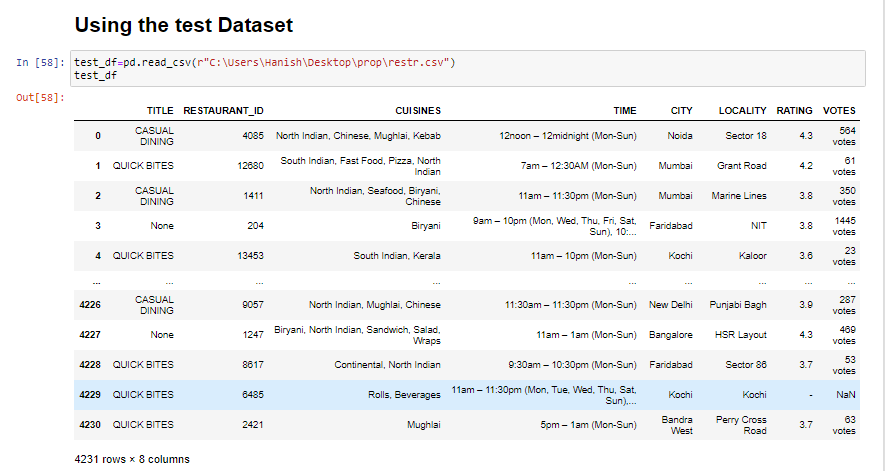
Grid search. Grid search is arguably the most basic **hyperparameter tuning** method. With this technique, we simply build a model for each possible combination of all of the **hyperparameter** values provided, evaluating each model, and selecting the architecture which produces the best results.

**Step 20-**

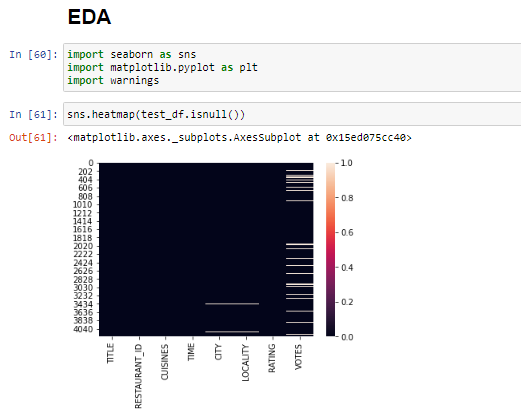
****

This line of code will save the model so that we can use it further for Testin part.

**Now we will proceed with Testing part**

****

Here we are doing the same steps Loading the Test dataset .



Checking if there is any null value or not .

After doing the data cleaning.We wil be running this line of code .Shown Below

