**Restaurant food cost analysis**

**Problem statement:**

Before entering any restaurant, the first question which comes in my mind is what will be the cost of food. All of us have our favorite restaurant and favorite types of cuisine but not everyday we can afford to go and spend our money. The food cost at any restaurant depends on various factors such as location, cuisines which is offered, city, rating Etc.so before going to any restaurant I consider all this aspect and make my decision weather I can afford the cost of food or not. As digitization reshapes the functional and operational outlook of the life, we can analysis the cost of food at any restaurant even before entering the restaurant. We can use our Data Science skills to investigate what are the factors that really affects the food cost.



**Features of the dataset:**

**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.

There are 9 columns in the dataset and I will be exploring each column and see what affects the cost of food at a restaurant. The first step will be importing all the necessary libraries and loading the csv files which contains all the data about the cost at restaurant.



Pre-processing pipeline:

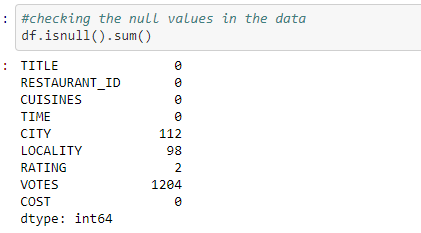
After looking at the dataset it requires a bit of cleaning like;

1)Removing NaN values

2)encoding the categorical values

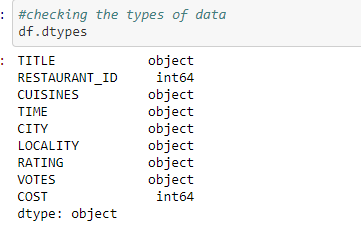
3)dropping some columns

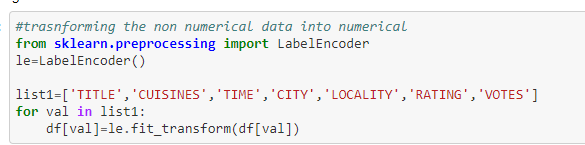
Lets have a look at all the NaN values present in the dataset.



So we can see there are NaN values present in the columns like City, Locality, Rating and Votes. We will be using ffill method to treat the Nan values.

There are 9 columns in the dataset out of which 6 columns are object and two is integer type I will be using label encoder to convert the columns.



I will be importing label encoder from sklearn pre-processing to convert the columns 

I have used For loops since there are multiple columns and passed the values in list1 and converted all the columns into integer type.

I will be dropping the column restaurant id since it does not contribute to the cost of food at restaurant.

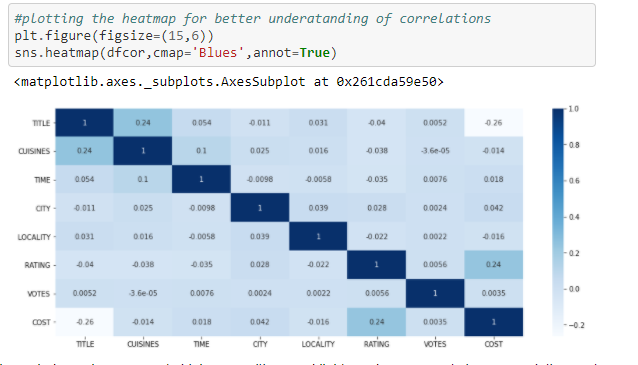
Now the data is cleaned and is ready for plotting.

**EDA process:**

Exploratory Data Analysis refers to the critical process of performing initial investigation on data so as to discover patterns, to spot anomalies, to test hypothesis and to check assumptions with the help of summary statistics and graph representation. It is a good practice to understand the data first and try to gather as many insights from it. EDA is all about making sense of data in hand, before getting them dirty with it.

To start with, I imported necessary libraries like matplotlib and seaborn which is needed for visualization of the data

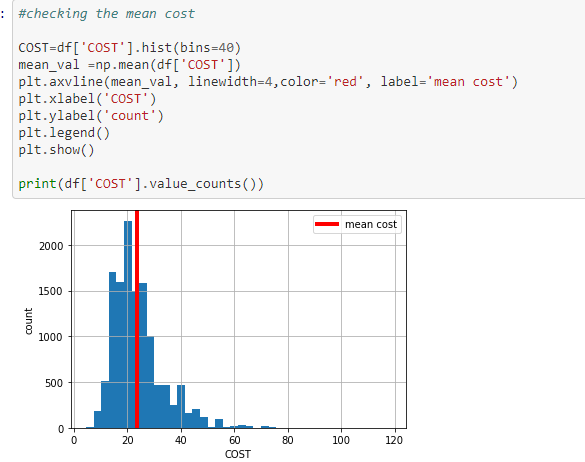
First let’s check the correlation between the dataset.



1. Dark shades represent positive correlation while lighter shades represent negative correlation
2. If you set annot=True, you’ll get values by which features are correlated to each other in grid-cells.
3. Cost and Rating has the highest correlation with each other
4. Other column do not have stronger correlation with cost and there are few columns such as cuisines, titles and locality which has negative correlation with cost.

**Using histogram:**

A histogram is a common choice for an initial visualization of a single variable because it shows the distribution of the data. The x-position is the value of the variable grouped into intervals called bins, and the height of each bar represents the count (number) of data points in each interval.

****

First, here is a general histogram of restaurant food cost. We have plotted the mean fees indicated by the red line. As we can clearly see this takes on standard, normally distributed shape.

But we would want to explore the relationship of restaurant cost with other columns to have a better understanding of what factors affects the cost of food so let’s plot a bar plot and compare the columns with fees.

**Using boxplot:**

A box plot (or box-and-whisker plot) shows the distribution of quantitative data in a way that facilitates comparisons between variables. The box shows the quartiles of the dataset while the whiskers extend to show the rest of the distribution.

The box plot (a.k.a. box and whisker diagram) is a standardized way of displaying the distribution of data based on the five-number summary:

Minimum

First quartile

Median

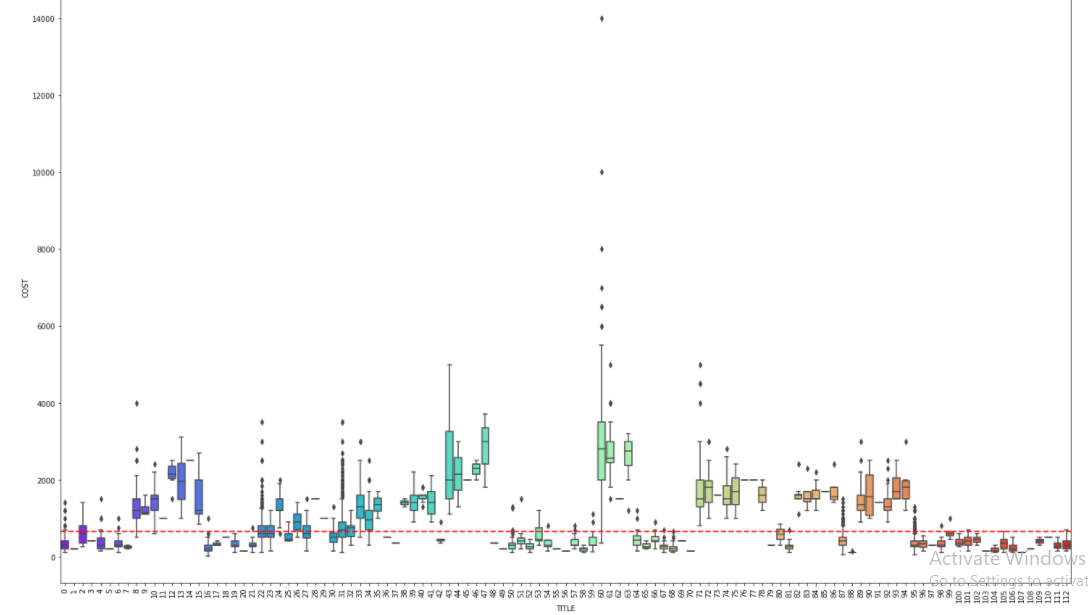
Third quartile

Maximum.

Title vs cost



We are using bivariate analysis to find the relationship between restaurant title and cost.

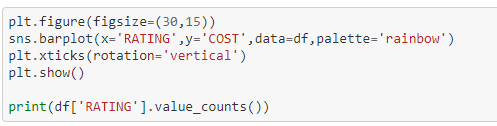


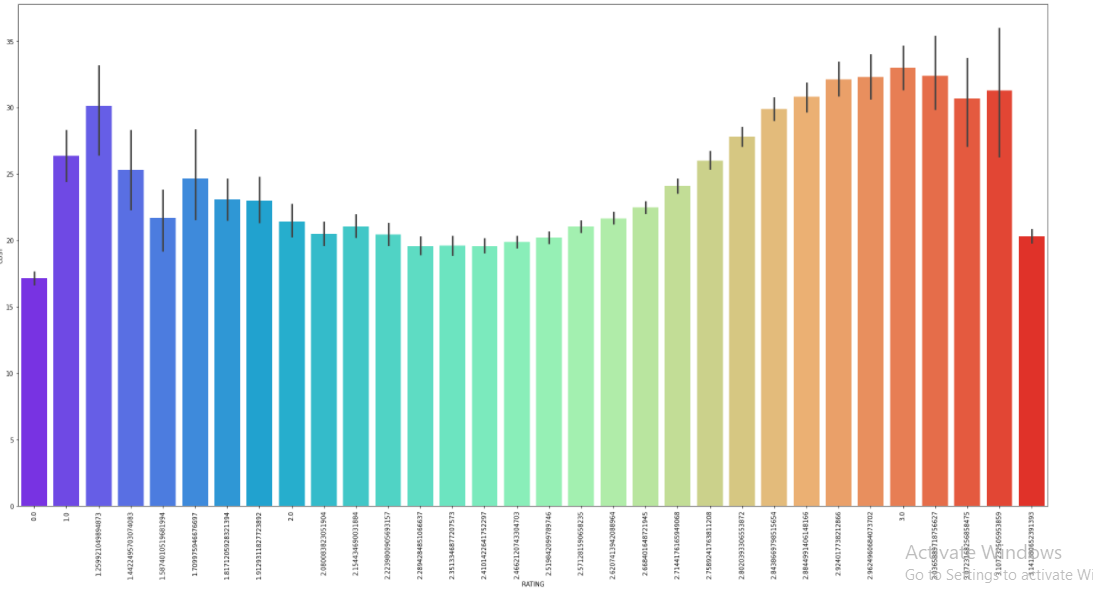
Since we have already converted all the restaurant title into integer type, we can see restaurant title 60 has the highest cost also there are outliers present in the data. We can also see the mean cost which is represented by the red line which is in between 0 to 200.

**Using Barplot:**

A barplot (or barchart) is one of the most common type of plot. It shows the relationship between a **numerical variable** and a **categorical variable**.

Rating vs cost





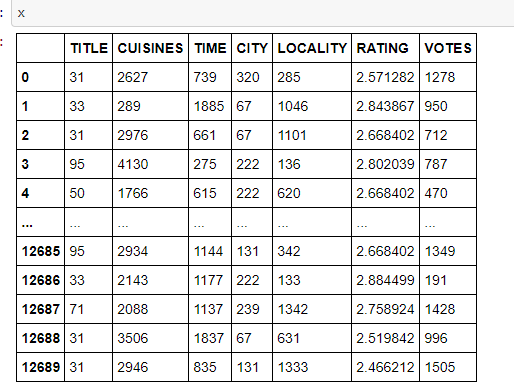
As we can see there is changes in cost with the change in ratings, however it is very dynamic.

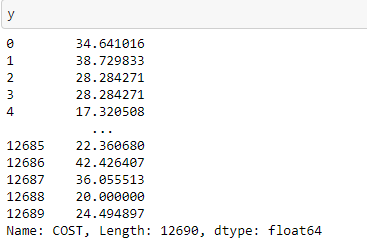
**EDA remarks:**

At this point we, a simple decision of going to a restaurant can be built based on the ratings of the restaurant.it is evident from the graphs and correlation that higher the ratings is higher the cost of the food will be in the restaurant. we have other features but with negative correlation or very low correlations Next step will be scaling the data and building the model where we will test various models to find the best possible outcomes.

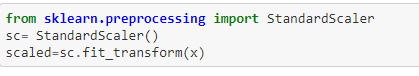
**Splitting the data:**

I am splitting the data into x and y variable. For splitting the data, I have used the drop function where I have dropped our target variable (cost) from the dataset.so now we have our x and y variables-



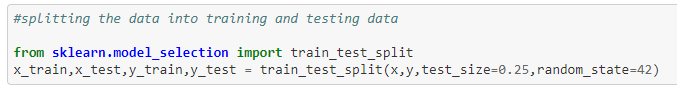


After dropping the columns, we can use standard scaler to transform x variable



We are importing StandardScaler from the scikit learn and scaling all the x variables so we can use this for model building which is our next step but before that we are splitting the data into training and testing.

**Train test split:**

****

We have split the data into train and test with test size of .25 and also provide with the random state of 42.now we can move towards our model building stage and find the accuracy of different models.

**Model building:**

This is one of the most important stage in analysing the data. We will be importing various libraries from scikit learn which will help us in model building. Since it’s a linear regression problem we will be using linear regression algorithms



So we have imported the necessary libraries to predict the best possible outcomes.frist lets understand each machine learning algorithm one by one, how it function and understand how will it help in predicting the doctor consultation fees.

**Linear Regression:**

Linear regression is one of the most popular and best understood algorithms in the machine learning landscape. Since regression task belong to the most common machine learning problems in supervised learning, every machine learning engineer should have a thorough understanding of how it works. This blogpost covers how the linear regression works, so with the help of linear regression and its various tools and techniques we are trying to find the best accuracy.

It is used to estimate real values (cost of houses, number of calls, total sales etc.) based on continuous variable(s). Here, we establish relationship between independent and dependent variables by fitting a best line. This best fit line is known as regression line and represented by a linear equation Y= a \*X + b

In this equation:

Y – Dependent Variable

a – Slope

X – Independent variable

b – Intercept

**regularization method:**

The regularization procedure aims at avoiding the model to overfit the data and thus deals with high variance issues

we have two regularization method which are

lasso: The Lasso is a linear model that estimates sparse coefficients with l1 regularization.

Ridge: Ridge regression addresses some of the problems of Ordinary Least Squares by imposing a penalty on the size of the coefficients with l2 regularization.

We also have elastic net which is a linear regression model trained with both l1 and l2 -norm regularization of the coefficients.

## **Decision Tree:**

This is one of my favourite algorithm and I use it quite frequently. It is a type of supervised learning algorithm that is mostly used for classification problems. Surprisingly, it works for both categorical and continuous dependent variables. In this algorithm, we split the population into two or more homogeneous sets. This is done based on most significant attributes/ independent variables to make as distinct groups as possible

## **kNN (k- Nearest Neighbors):**

It can be used for both classification and regression problems. However, it is more widely used in classification problems in the industry. K nearest neighbours is a simple algorithm that stores all available cases and classifies new cases by a majority vote of its k neighbours. The case being assigned to the class is most common amongst its K nearest neighbours measured by a distance function.

These distance functions can be Euclidean, Manhattan, Minkowski and Hamming distance. First three functions are used for continuous function and fourth one (Hamming) for categorical variables. If K = 1, then the case is simply assigned to the class of its nearest neighbour. At times, choosing K turns out to be a challenge while performing kNN modelling.

## **Random Forest:**

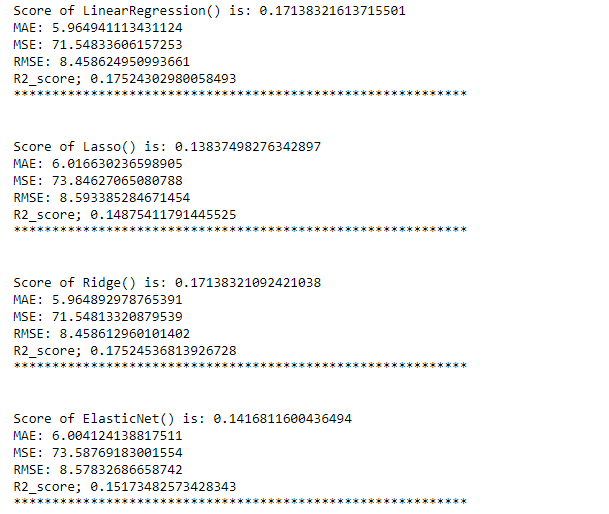
Random Forest is a trademark term for an ensemble of decision trees. In Random Forest, we’ve collection of decision trees (so known as “Forest”). To classify a new object based on attributes, each tree gives a classification and we say the tree “votes” for that class. The forest chooses the classification having the most votes (over all the trees in the forest).

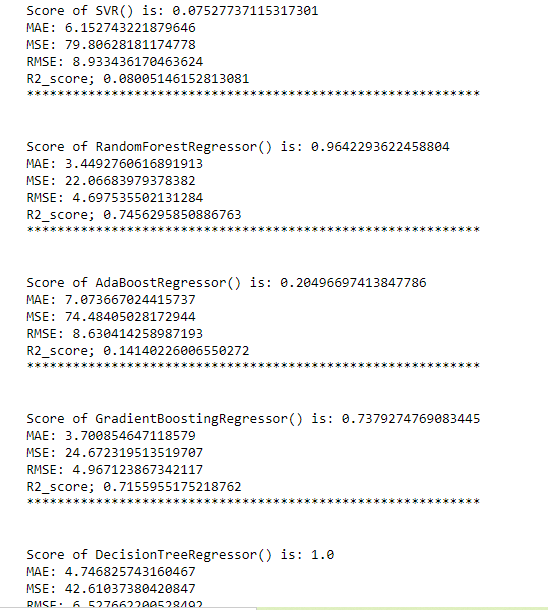
## **Gradient Boosting:**

GBM is a boosting algorithm used when we deal with plenty of data to make a prediction with high prediction power. Boosting is actually an ensemble of learning algorithms which combines the prediction of several base estimators in order to improve robustness over a single estimator. It combines multiple week or average predictors to a build strong predictor.

Alright, so let’s go through this and see which of these models will perform the best and gives the best outcomes.



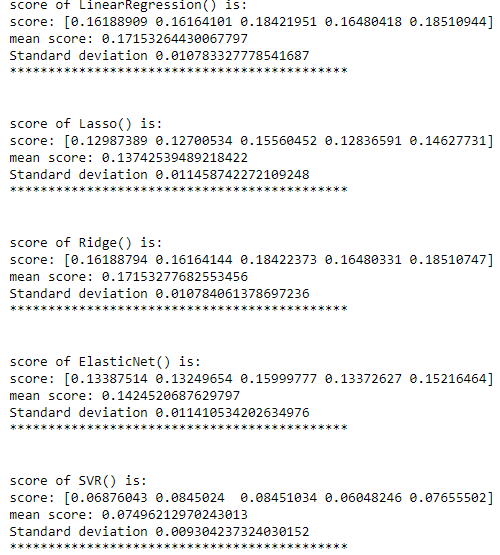
I am using For loop for all the models where I will print the accuracy score, mean absolute error, root mean squared error and r2 score of the data. Let’s see what is the outcome-

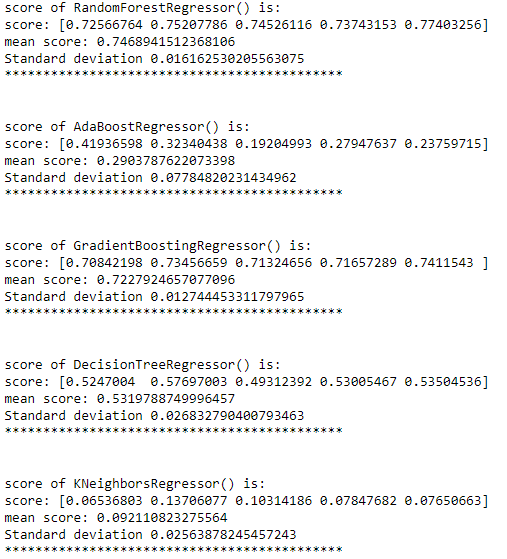


From all the models we can see that Random Forest regressor is performing the best with the accuracy score of 96%, Mean absolute error of 3.4, Mean squared error of 22.0, Root mean squared error of 4.6 and the r2 score of 0.74.

**Cross validation:**

Cross validation is a method that is used to select a model that does not rely too much on the initial training set.  we just can’t assume that it is going to work well on data that it has not seen before. In other words, we can’t be sure that the model will have the desired accuracy and variance in production environment. I have used cross validation to see if there is any overfitting or underfitting of the trained dataset.

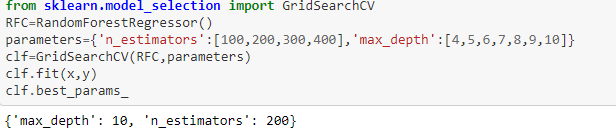




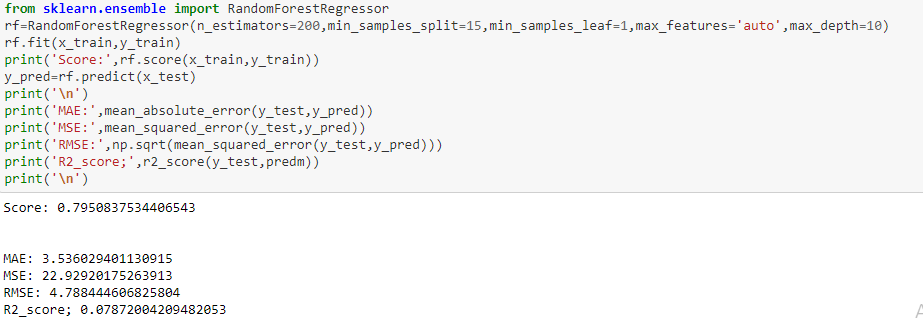
Looking at the cross validation results we can see that random forest regressor Is still performing best even though the accuracy has gone done to 74% but we can consider that there is no overfitting or underfitting of the data.

**Hyperparameter tuning:**

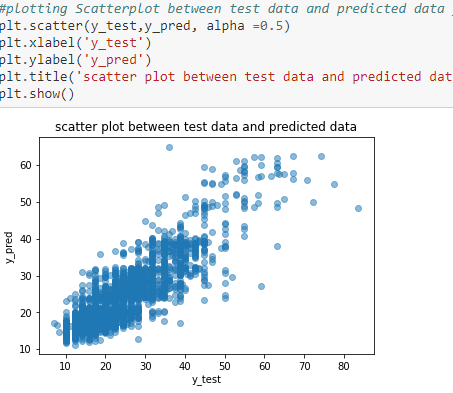
So, I am using hyperparameter tuning to find the best possible parameters to enhance the performance of our best performing model. I will be using grid search method as Grid search is arguably the most basic hyperparameter tuning method. With this technique, we simply build a model for each possible combination of all of the hyperparameter values provided, evaluating each model, and selecting the architecture which produces the best results

****

N\_estimators and max\_depth are the two paramterter for random forest. We can use this parameters to find the best accuracy on our model.



We can see we have improved our accuracy after using the best parameter to 79% with mean absolute error of 3.5, mean squared error of 22.9, root mean squared error of 4.7 and r2 score of 0.07.

**Plotting scatter plot between predicted data and test data:**

We can see the scatterplot between predicted data and test data which is densely concentrated from 10 to 45, and lightly concentrated from 45 to 80.

**Concluding remarks:**

in this blog, we have learned about what affects the cost of food in restaurant. I performed all the mandatory pre-processing pipeline and EDA. There were some data cleaning to be done for example handling the null values, dropping the unnecessary columns, encoding the categorical variables. After that we saw in visualization that what are the features which is affecting the cost of food in a restaurant and rating of the restaurant was highest one. During our model building phase I have used For loops for different machine learning algorithms and I was able to achieve the final accuracy of 79% which I think is good considering I took all the necessary steps of cross validating and hyper tunning.

Blog by

Akchay Pradhan