**PROJECT 3**

**Cab Fare Prediction**

**Rishabh Jain**

**Data Analyst, Home Credit India Pvt. Ltd.**

**Content**

1. Introduction
   1. Problem Statement
   2. Business Problem
   3. Data
2. Methodology
   1. Data Checks and Modification
   2. Missing and Invalid Value Computation
   3. Feature Engineering
   4. Data Distribution
   5. Outlier Analysis
   6. Feature Scaling
   7. Final Input DataFrame
3. Modelling
   1. Linear Regression
   2. Decision Tree
   3. Random Forest
   4. Gradient Boosting
   5. Hyper Parameters Tunings for optimizing the results
4. Conclusion
   1. Model Evaluation
   2. Model Selection
5. **Introduction**
   1. Problem Statement

You are a cab rental start-up company. You have successfully run the pilot project and now want to launch your cab service across the country. You have collected the historical data from your pilot project and now have a requirement to apply analytics for fare prediction. You need to design a system that predicts the fare amount for a cab ride in the city.

Dataset Details:

Dataset Characteristics: Multivariant

Number of Attributes: 7

Missing Values: Yes

|  |  |
| --- | --- |
| Variables | Description |
| fare\_amount | Fare that is charged for the ride |
| pickup\_datetime | Cab pickup date with time data |
| pickup\_longitude | Cab pickup location (longitude) |
| pickup\_latitude | Cab pickup location (latitude) |
| dropoff\_longitude | Cab drop-off location (longitude) |
| dropoff\_latitude | Cab drop-off location (latitude) |
| passenger\_count | Number of passengers in Cab |

* 1. Business Problem

The companies these days runs a pilot project to find how the real- world is treating the idea. The idea itself cannot guarantee one that the start-up will reach the dream of the co-founder. In these cases, pilot project can give you a lots of information about how the idea is actually implemented in the real world and how much it’s efficient. At the time of pilot project, the data is captured which then used to gather the insights and future prediction.

* 1. Data

The ‘train\_cab.csv’ file is used for the analysis and modelling. The glimpse of few rows of data is given below



Size of Dataset Provided: - 16067 rows, 7 Columns (including dependent variable)

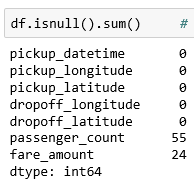
Out of these variables, “fare\_amount” variable is target variable.

1. **Methodology**
   1. Data Checks and Modification

To initialize the process of data modelling, certain checks are performed so that the model doesn’t get biased and predict wrong value of target variable for the certain input variables.

First check is check for NULL values in the dataset. If so, then certain procedures are to be used to fill that NULL values, for example

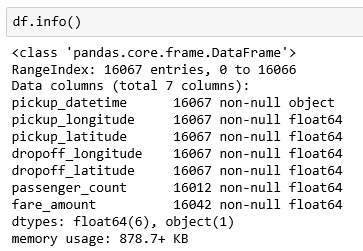
1. Removing NULL value containing row
2. Mean method i.e., filling NULL value with the mean for the values in that column
3. Median method i.e., fiulling NULL values with median value
4. KNN Imputation (K-Nearest Neighbour mehtod)



‘fare\_amount’ is our target variable so NA value in the dataset will be removed as the information will not contribute to build model to predict cab fare.

The above is used in python to check for NULL values in each column of the dataframe.

Second check is to check for the datatypes of the variables in the loaded dataframe.



* 1. Missing and Invalid Value Computation

In statistics, missing data, or missing values, occur when no data value is stored for the variable in an observation. Missing data are a common occurrence and can have a significant effect on the conclusions that can be drawn from the data. If a column has more than 30% of data as missing value either we ignore the entire column or we ignore those observations.

To calculate missing value percentage for each column the following function is used in Python:

*def missing\_perc(data):*

*Missing\_Value = pd.DataFrame((data.isnull().sum()/len(data)\*100))*

*Missing\_Value = Missing\_Value.rename(columns = {0: 'Missing\_percentage'})*

*#Arranging Missing Values in Decreasing Order*

*Missing\_Value = Missing\_Value.sort\_values('Missing\_percentage', ascending = False)*

*print(Missing\_Value)*

in R:

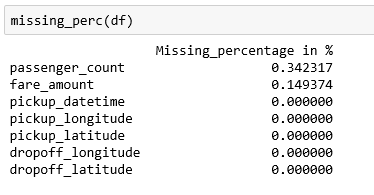
*missing\_perc<-function(data){*

*missing\_value = (as.data.frame(colSums(is.na(data)))\*100/nrow(data))*

*colnames(missing\_value) <- c("Missng Value Percentage")*

*View(missing\_value)*

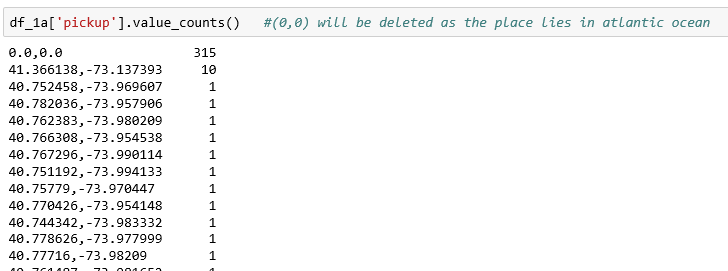
*}*

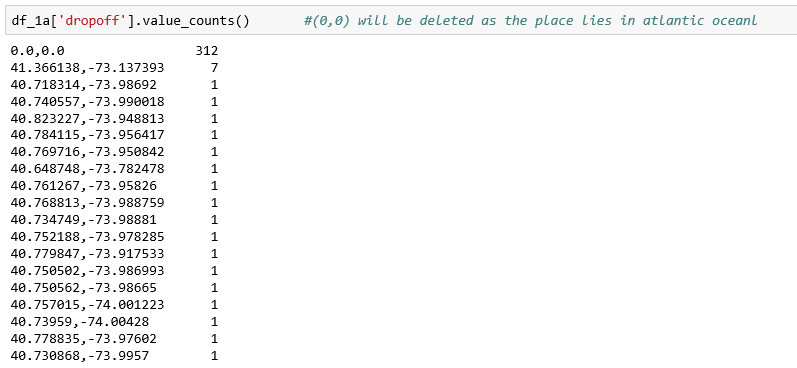


In the given data the maximum percentage of missing value is 0.34% for body mass index column. So, we will drop the missing values as it is hardly 1% of whole data. The data also contains invalid values which needs to be removed. Given below are the variables which had invalid value.

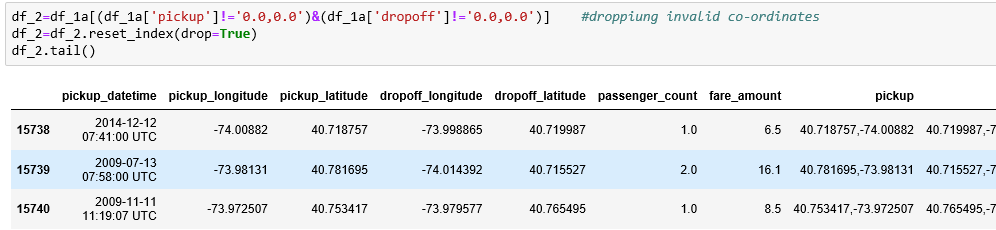
1. Location Co-ordinates (Latitude, Longitude)

From the observations it is noted that the co-ordinates of pickup and drop-off is (0.0,0.0) which is the location co-ordinates that lies in the Atlantic Ocean.





So dropped such co-ordinates:



Another problem with co-ordinate was that the pickup and drop-off co-ordinate are same so the distance will come out to be 0. Thus such values were found to be 156 in data and they are removed from the dataset to avoid any discrepancy.

The longitude should be in range (-180,180) and latitude should be in range (-90,90). There is 1 value whose pickup latitude is greater than 90 i.e., invalid hence removed from the dataset.

1. Number of Passengers

It is noted that the number of passengers are invalid as:

1. Number of passenger is either 0 or negative
2. Number of passenger is way too much whereas big car like SUV can have 6 or 7 passengers at max.
3. Number of passengers is in fractions.

Such values are 74 in count in the dataset which are then removed.

1. Fare Amount

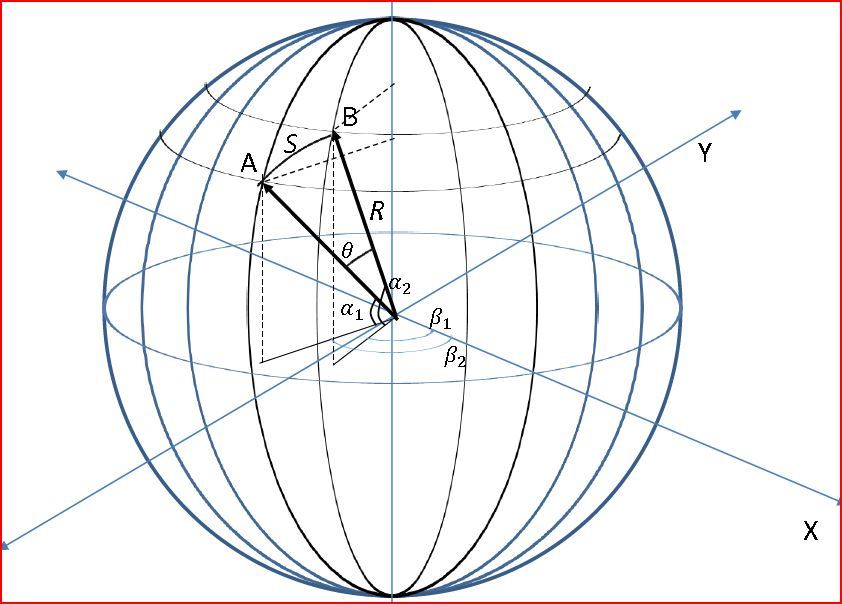
The fare amount for some data rows is 0 or negative which is considered as invalid. 5 such rows are found in the data which is removed.

* 1. Feature Engineering

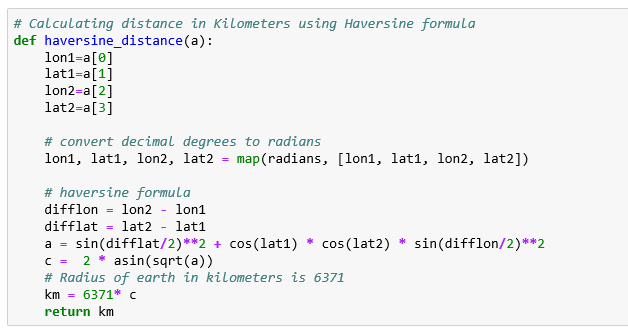
After observing the data, it is noted that more variables need to be added to the dataset to make it more resourceful. Such variables are given as follows:

1. Distance between pickup and drop-off

The distance between pickup and drop-off is calculated using Haversine formula. It determines the great-circle distance between two points on a sphere given their longitudes and latitudes. Important in navigation, it is a special case of a more general formula in spherical trigonometry, the law of haversines, that relates the sides and angles of spherical triangles.

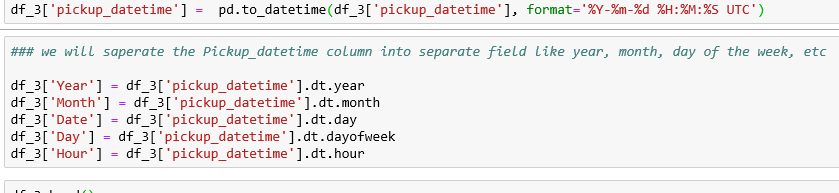


The Haversine distance is being computed using the below mentioned user-defined function:



1. Year, Month, Date, Day of Week & Hour

The data contains variable ‘pickup\_datetime’ which can be used to extract year, month, date, day of week and hour. Below given is the code used to perform this in python:

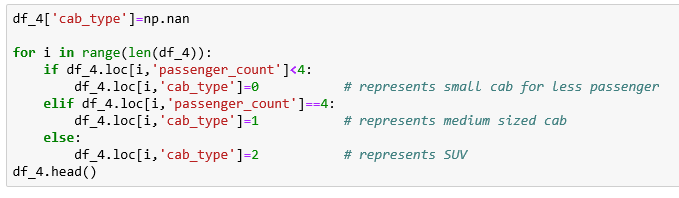


1. Cab Type

As the passenger count increases the type of cab that the passengers will take changes. So to mark that in the data new variable introduced with below logic:

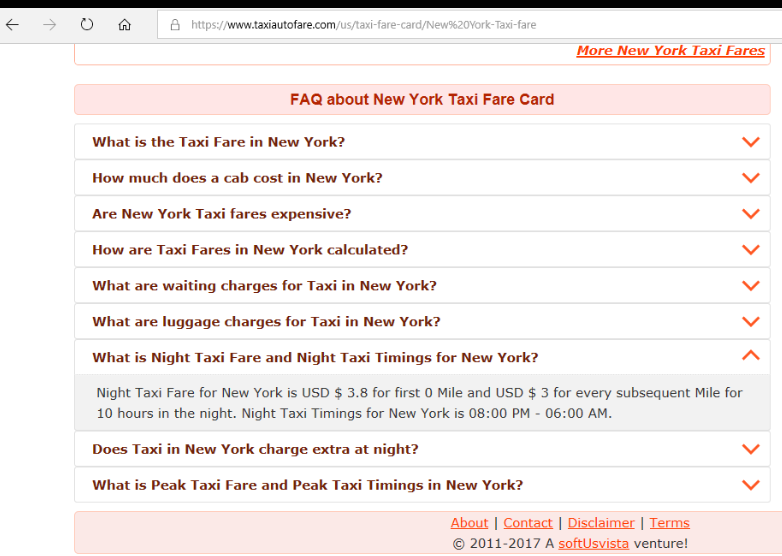
1. For Number of passengers less than 4, ‘cab\_type’ will be 0 i.e., small cab.
2. For Number of passengers equal to 4, ‘cab\_type’ will be 1 i.e., medium cab.
3. For Number of passengers greater than 4, ‘cab\_type’ will be 2 i.e., big cab or SUV.

Below is the logic defined in python:

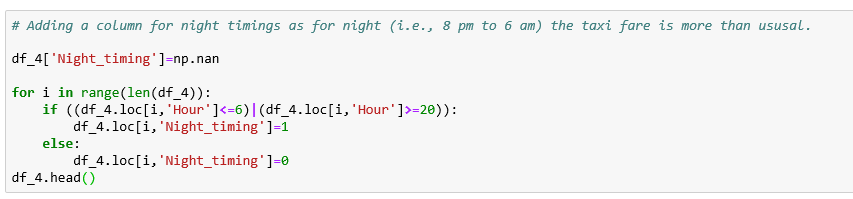


1. Night timing

All location given in the dataset are of New York, and for New York there is extra charges for the cabs taken between 8 p.m. and 6 a.m.



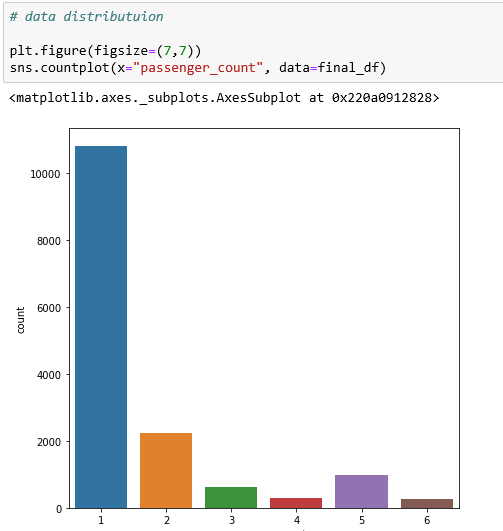
Thus, adding new variable having night timing identification. The logic is mentioned below:



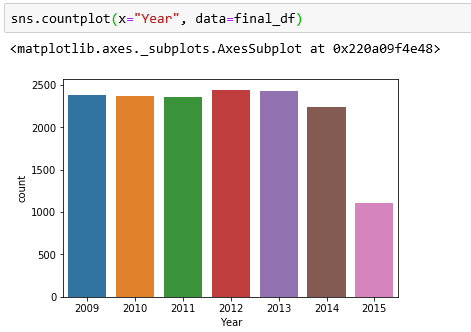
* 1. Data Distribution

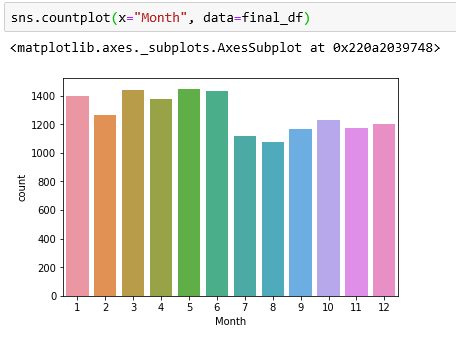
The model is greatly impacted by the data distribution because on the basis of that the model makes rules to predict the target variable. The data distribution for continuous variable should be normalised to develop a better model. So, to have a look at the distribution of the data, Histograms are used with density curves to know how the data is distributed among the dataframe.

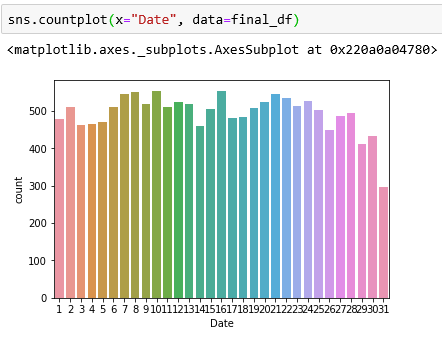
Below are the graph and plot produced for visualization:

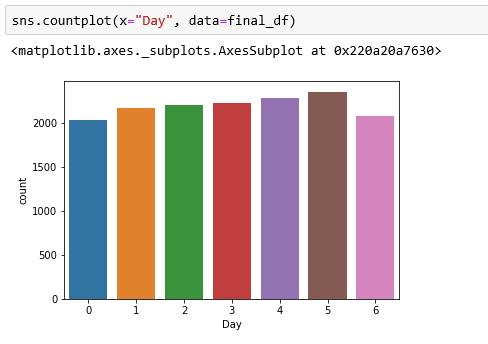


It can be seen that the data for cab rides with single passenger is much greater than the rides with multiple passengers, It can be said that the cabs were mostly used for commute rather than vacation.

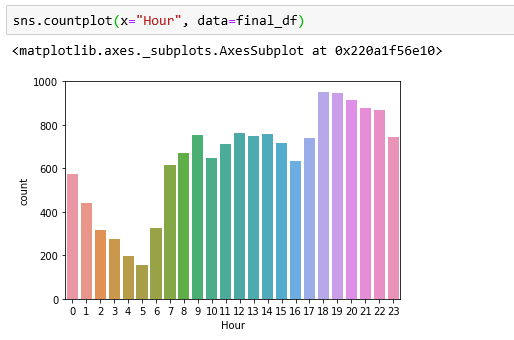






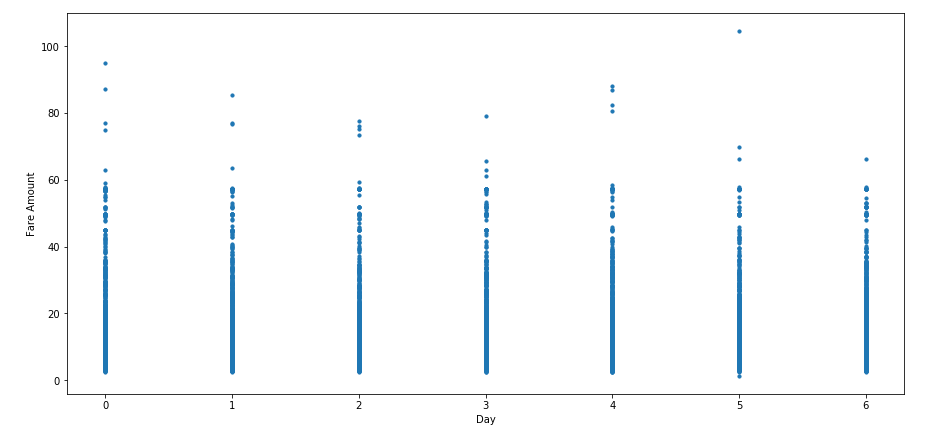


It can be observed from above graphs that datapoints are almost equally distributed among year, month and date.



From above graph it is observed that the cab count is much between 6 pm to 7 pm that is due to the fact that most of the office end between these timings.

**Relationship between Day of week and fare\_amount**



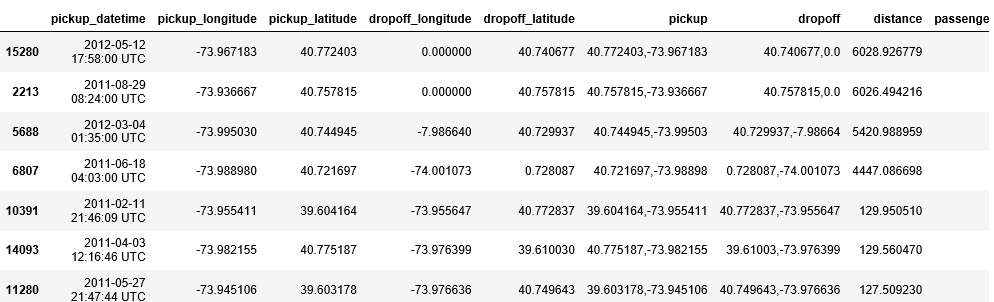
The highest fares seem to be on a Sunday, Monday and Thursday, and the low on Wednesday and Saturday. May be due to low demand of the cabs on saturdays the cab fare is low and high demand of cabs on sunday and monday shows the high fare prices.

* 1. Outlier Analysis

It is observed from these probability distributions that most of the variables are skewed. The skew in these distributions can be most likely explained by the presence of outliers and extreme values in the data. One of the other steps of pre-processing apart from checking for normality is the presence of outliers. Below are the variables that contains outliers:

1. ‘distance’

The column contains outliers that are given follows:

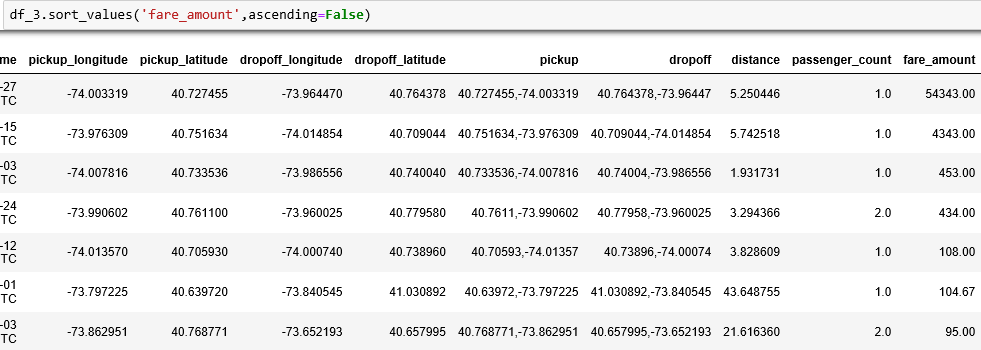


For distance above 130 km all the values seem to be inappropriately large. So removed such outliers form data.

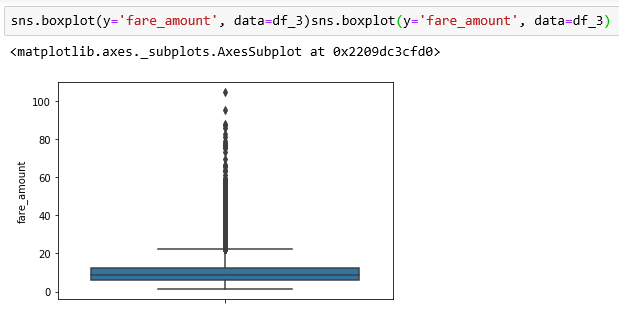
Also, it seems illogical to take cab for haversine distance less than 100 meters. So, removed such values also.

1. ‘fare\_amount’

The column contains outliers which are given as follows:



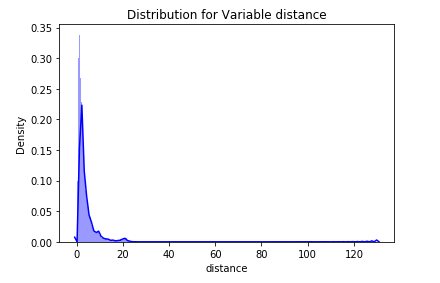
For first 5 rows it can be seen that the ‘fare\_amount’ doesn't corresponds to the distance that the cab had travelled as for distance 1-5 kms, no one can charge such high amount. ‘fare\_amount’ below 108 looks reasonable to the distance that is being traveled by the cab, therefore removing the top 5 rows from data.

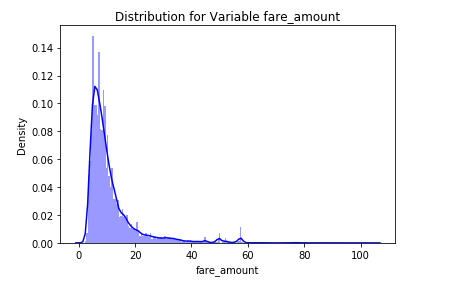


* 1. Feature Scaling

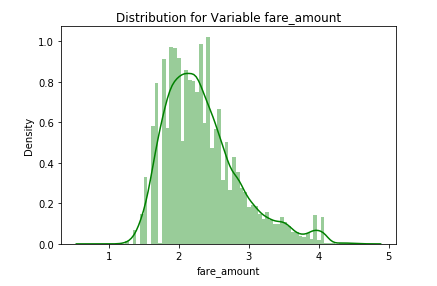
Skewness is asymmetry in a statistical distribution, in which the curve appears distorted or skewed either to the left or to the right. Skewness can be quantified to define the extent to which a distribution differs from a normal distribution. Here we tried to show the skewness of our variables and we find that our target variable absenteeism in hours having is one sided skewed so by using log transform technique we tried to reduce the skewness of the same.

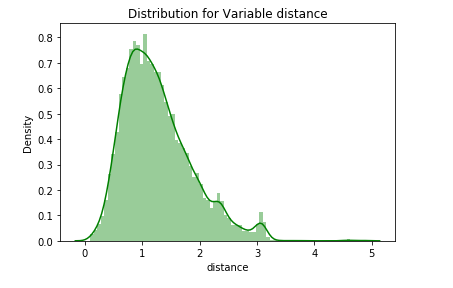
Below mentioned graphs shows the probability distribution plot to check distribution before log transformation:





Below mentioned graphs shows the probability distribution plot to check distribution after log transformation:





* 1. Final Input DataFrame

The main purpose of this modelling is to predict losses that the company will incur in the next year 2011.

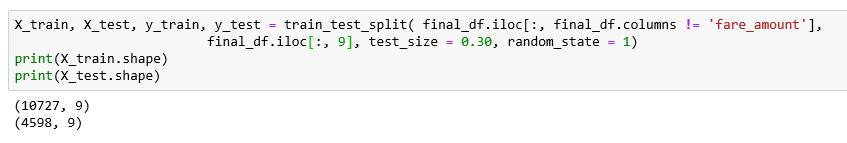
1. **Modelling**

The type of model needs to be implemented mainly depends upon the target variable data type, predictor variable’s data type and type of relationship they carry with each other.

The models applied on the data are given as follows:

1. Linear Regression
2. Decision Tree
3. Random Forest
4. Gradient Boosting
5. Hyper Parameters Tunings for optimizing the results

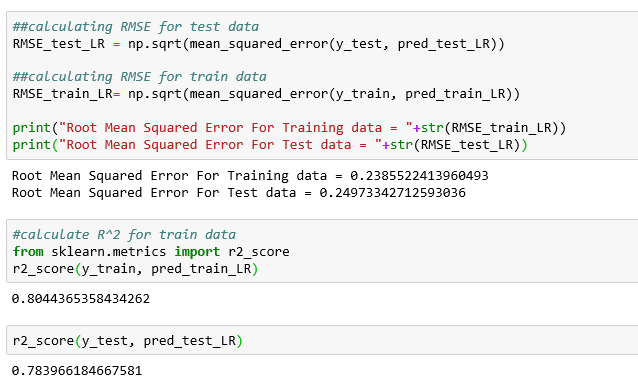
To proceed with implementing the models the data needs to be split into train and test sample. Below is the code is used to perform that:



* 1. Linear Regression

Multiple linear regression is the most common form of linear regression analysis. Multiple regression is an extension of simple linear regression. It is used as a predictive analysis, when we want to predict the value of a variable based on the value of two or more other variables. The variable we want to predict is called the dependent variable (or sometimes, the outcome, target or criterion variable).

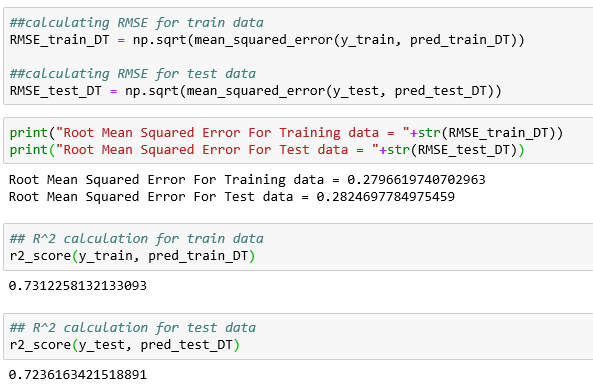
The model performance is mentioned below in the terms of error metrics (RMSE) and R squared value:



* 1. Decision Tree

A tree has many analogies in real life, and turns out that it has influenced a wide area of machine learning, covering both classification and regression. In decision analysis, a decision tree can be used to visually and explicitly represent decisions and decision making. As the name goes, it uses a tree-like model of decisions.

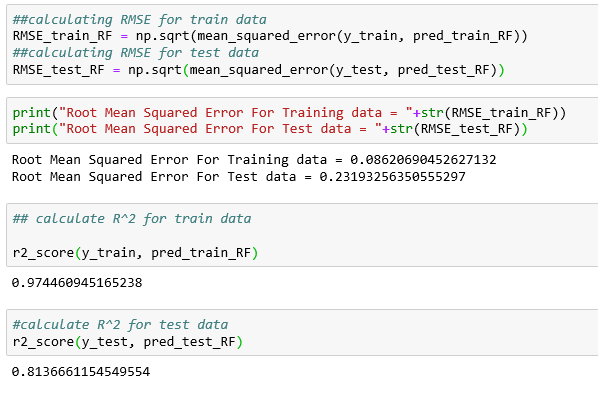
The model performance is mentioned below in the terms of error metrics (RMSE) and R squared value:



* 1. Random Forest

Random forests or random decision forests are an ensemble learning method for classification, regression and other task, that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Random decision forests correct for decision trees' habit of overfitting to their training set.

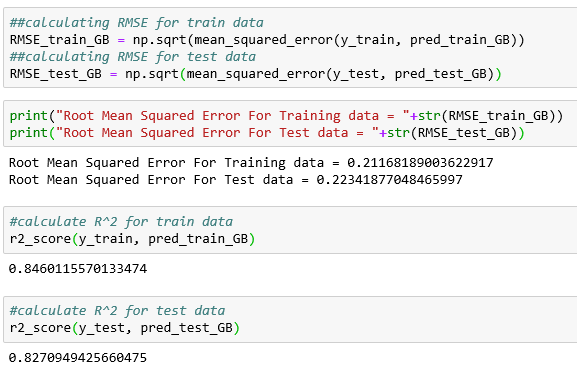
The model performance is mentioned below in the terms of error metrics (RMSE) and R squared value:



* 1. Gradient Boosting

Gradient boosting is a machine learning technique for regression and classification problems, which produces a prediction model in the form of an ensemble of weak prediction models, typically decision trees. It builds the model in a stage-wise fashion like other boosting methods do, and it generalizes them by allowing optimization of an arbitrary differentiable loss function.

The model performance is mentioned below in the terms of error metrics (RMSE) and R squared value:



* 1. Hyper Parameters Tunings for optimizing the results

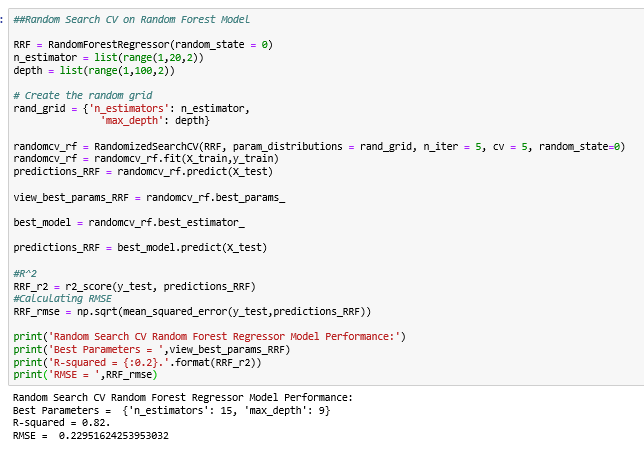
Model hyperparameters are set by the data scientist ahead of training and control implementation aspects of the model. The weights learned during training of a linear regression model are parameters while the number of trees in a random forest is a model hyperparameter because this is set by the data scientist. Hyperparameters can be thought of as model settings. These settings need to be tuned for each problem because the best model hyperparameters for one particular dataset will not be the best across all datasets. The process of hyperparameter tuning (also called hyper parameter optimization) means finding the combination of hyperparameter values for a machine learning model that performs the best - as measured on a validation dataset - for a problem.

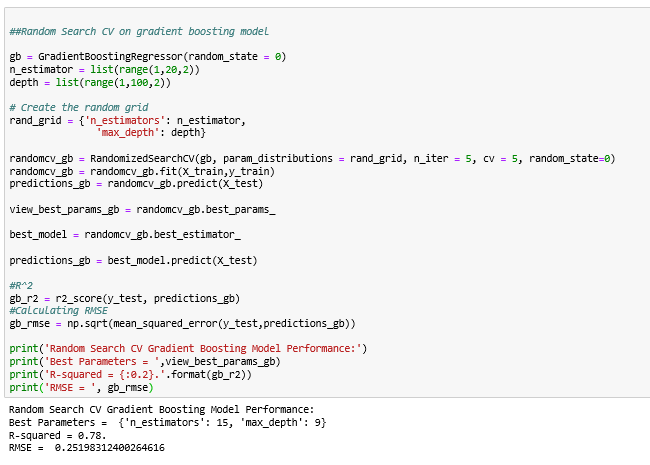
Here we have used two hyper parameters tuning techniques

1. Random Search CV: This algorithm set up a grid of hyperparameter values and select random combinations to train the model and score. The number of search iterations is set based on time/resources.

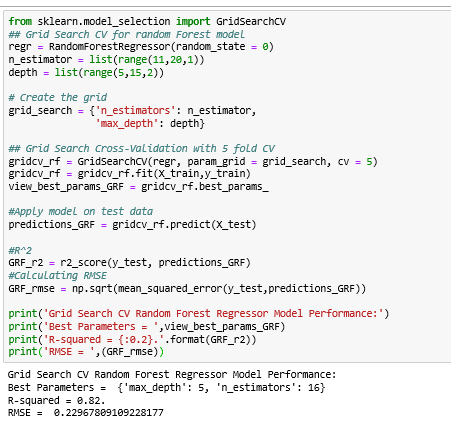
2. Grid Search CV: This algorithm set up a grid of hyperparameter values and for each combination, train a model and score on the validation data. In this approach, every single combination of hyperparameters values is tried which can be very inefficient.

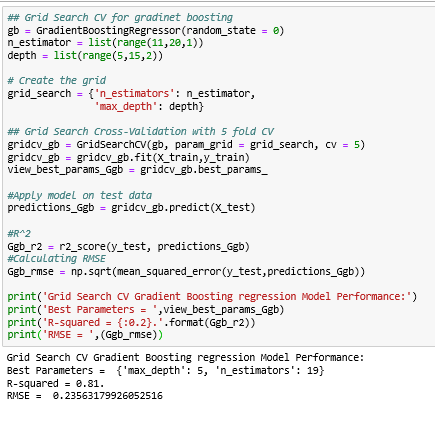
Using Random Search CV on Random forest and gradient boosting model.





Using Grid Search CV on Random forest and gradient boosting model.





1. **Conclusion**
   1. Model Evaluation

To evaluate the machine learning model, two metrics are used and they are given as follows:

1. RMSE (Root Mean Square Error): It is a frequently used measure of the difference between values predicted by a model and the values actually observed from the environment that is being modelled.



1. R Squared (R2): is a statistical measure of how close the data are to the fitted regression line. It is also known as the coefficient of determination, or the coefficient of multiple determination for multiple regression. In other words, we can say it explains as to how much of the variance of the target variable is explained.

Below table shows the model results before applying hyper tuning:

|  |  |  |
| --- | --- | --- |
| Model | RMSE | R squared |
| Linear Regression | 0.24 | 0.78 |
| Decision Tree | 0.28 | 0.72 |
| Random Forest | 0.23 | 0.81 |
| Gradient Boosting | 0.22 | 0.82 |

Below table shows results post using hyper parameter tuning techniques:

|  |  |  |  |
| --- | --- | --- | --- |
|  | Model | RMSE | R squared |
| Random Search CV | Random Forest | 0.22 | 0.82 |
| Gradient Boosting | 0.25 | 0.78 |
| Grid Search CV | Random Forest | 0.22 | 0.82 |
| Gradient Boosting | 0.23 | 0.81 |

Above table shows the results after tuning the parameters of our two best suited models i.e. Random Forest and Gradient Boosting. For tuning the parameters, we have used Random Search CV and Grid Search CV under which we have given the range of n\_estimators, depth and CV folds.

* 1. Model Selection

On the basis RMSE and R Squared results a good model should have least RMSE and max R Squared value. So, from above tables it can be seen that: -

1. Both the models- Gradient Boosting Default and Random Forest perform comparatively well while comparing their RMSE and R-Squared value.
2. After applying tunings Random forest model shows best results compared to gradient boosting.

Finally, it can be said that **Random forest model** is the best method to make prediction for this project with highest explained variance of the target variables and lowest error chances with **parameter tuning technique Grid Search CV**.