**GROUP – B**

**Assignment No: 1**

====================================================================

**Title:-** Predict the price of the Uber ride from a given pickup point to the agreed drop-off location. Perform following tasks:

1. Pre-process the dataset.

2. Identify outliers.

3. Check the correlation.

4. Implement linear regression and random forest regression models.

5. Evaluate the models and compare their respective scores like R2, RMSE, etc.

=====================================================================**Objective:-**

-To learn about Regression Technique

-To understand about predict the price of Uber ride.

-To design an algorithm which will tell the fare to be charged for a passenger.

=====================================================================

**Theory:-**

* In this practical, world's largest taxi company Uber inc. In these project we're looking to predict the fare for their future transactional cases and this company deliver services to lakhs of customers daily. Now it becomes really important to manage their data properly to come up with new business ideas to get best results. So, it becomes really important to estimate the fare prices accurately.
* Whenever we booked a cab, we were fascinated to know how actually the process for predicting the fare works with respect to like distance of the ride, surge multipliers, pick-up and drop location, weather and wind conditions, traffic and time of the commute. Also cab booking is a booming industry, we were curious to know the demand of cabs on basis of source and destination locations.
* To predict the fare prices need to find out correlation as well as linear regression techniques.
* **Dataset Description**

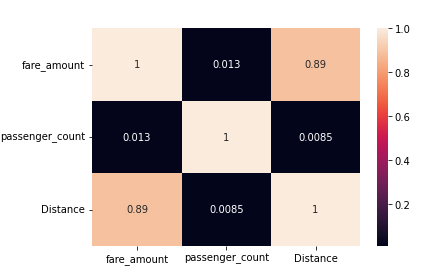
The dataset contains the following fields:

1. key - a unique identifier for each trip
2. Index
3. fare\_amount - the cost of each trip in usd
4. pickup\_datetime - date and time when the meter was engaged
5. passenger\_count - the number of passengers in the vehicle (driver entered value)
6. pickup\_longitude - the longitude where the meter was engaged
7. pickup\_latitude - the latitude where the meter was engaged
8. dropoff\_longitude - the longitude where the meter was disengaged
9. dropoff\_latitude - the latitude where the meter was disengaged.

The dataset contains data of 196629 bookings done with 9 features.

* **Data Preprocessing**

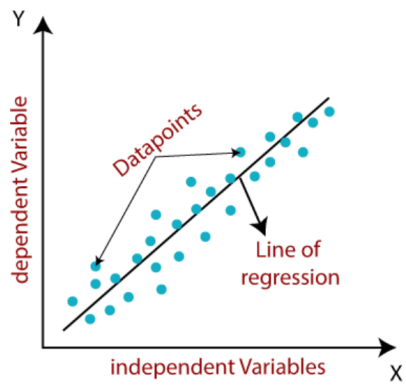
1. Replacing Nan values with their mean: The data has been visualized and analyzed to get the information about the features and their types. Invalid data such as zero prices has also been replaced to null values such that they can be replaced with appropriate values. The Missing values have been handled by replacing the null values with the mean.
2. Correlation between the features of the dataset with respect to price: The features correlations with our target variable i.e. ‘fare\_amount’, ‘distance’ and ‘passenger\_count’ were compared using the correlation matrix.



1. Mapping source and destination location names to their respective latitude and longitude: Additional features have been added to the dataset by mapping source and destination location names to their respective latitude and longitude for better results of fare and demand prediction.

* **Linear Regression**

Linear Regression is an algorithm that belongs to supervised Machine Learning. It tries to apply relations that will predict the outcome of an event based on the independent variable data points. The relation is usually a straight line that best fits the different data points as close as possible. The output is of a continuous form, i.e., numerical value. For example, the output could be revenue or sales in currency, the number of products sold, etc. In the above example, the independent variable can be single or multiple.



Linear regression can be expressed mathematically as:

y= β0+ β 1x+ ε

Here, Y= Dependent Variable

X= Independent Variable

β 0= intercept of the line

β1 = Linear regression coefficient (slope of the line)

ε = random error

The last parameter, random error ε, is required as the best fit line also doesn't include the data points perfectly.

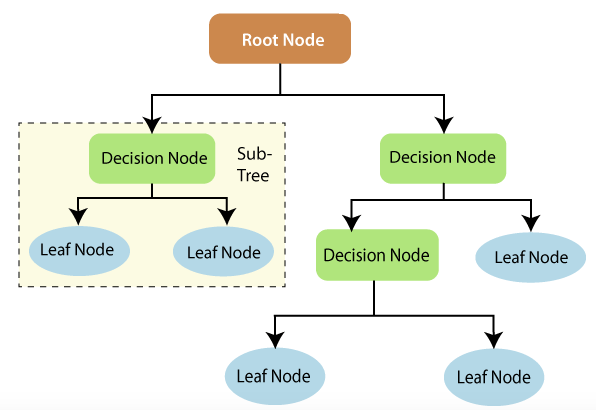
There are five basic steps to implementing linear regression:

1. Import the packages and classes that you need.
2. Provide data to work with, and eventually do appropriate transformations.
3. Create a regression model and fit it with existing data.
4. Check the results of model fitting to know whether the model is satisfactory.
5. Apply the model for predictions.

* **Random Forest Regression Model**

Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of ensemble learning, which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model. Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset. Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output.

Decision trees are the building blocks of a random forest algorithm. A decision tree is a decision support technique that forms a tree-like structure. An overview of decision trees will help us understand how random forest algorithms work. A decision tree consists of three components: decision nodes, leaf nodes, and a root node. A decision tree algorithm divides a training dataset into branches, which further segregate into other branches. This sequence continues until a leaf node is attained. The leaf node cannot be segregated further. The nodes in the decision tree represent attributes that are used for predicting the outcome. Decision nodes provide a link to the leaves. The following diagram shows the three types of nodes in a decision tree.



The information gain concept involves using independent variables (features) to gain information about a target variable (class). The entropy of the target variable (Y) and the conditional entropy of Y (given X) are used to estimate the information gain. In this case, the conditional entropy is subtracted from the entropy of Y.

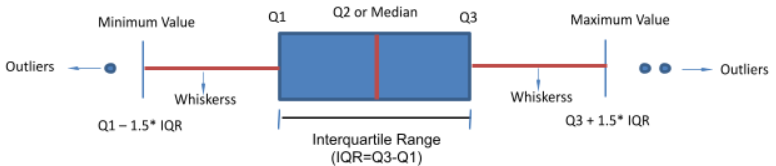
Regression is the other task performed by a random forest algorithm. A random forest regression follows the concept of simple regression. Values of dependent (features) and independent variables are passed in the random forest model.

* Box plot

A single box which gives you a visual idea about 5 components in a dataset. It is also known as box and whiskers plot or simply box plot. It is useful for describing measures of central tendencies and measures of dispersion in a dataset.

Box Plot represents the following points in a dataset.

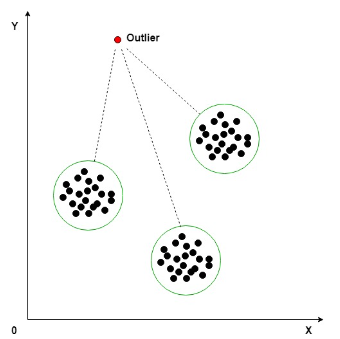
* Minimum Value
* First Quartile (Q1 or 25th Percentile)
* Second Quartile (Q2 or 50th Percentile)
* Third Quartile (Q3 or 75th Percentile)
* Maximum Value



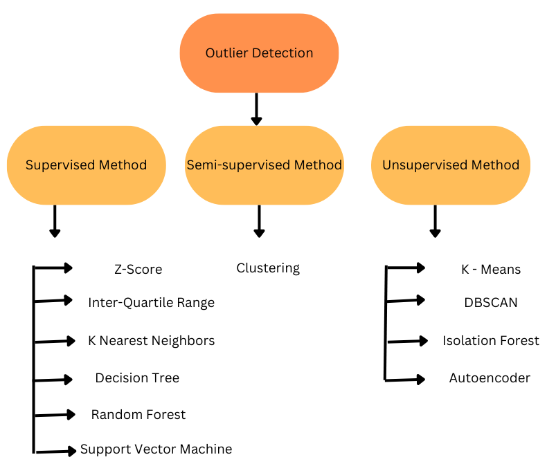
Along with the above 5 components. Boxplot also gives us below information:

* Outliers: Points lying beyond the minimum and maximum values are outliers
* Interquartile range: It is Q3-Q1. It is the spread or range of the middle 50% of the data
* Whiskers: From Minimum Value to Q1 is the first 25% of data
* From Q3 to Maximum value is the last 25% of the data
* Outliers

An outlier is an object that deviates significantly from the rest of the objects. They can be caused by measurement or execution error. The analysis of outlier data is referred to as outlier analysis or outlier mining.

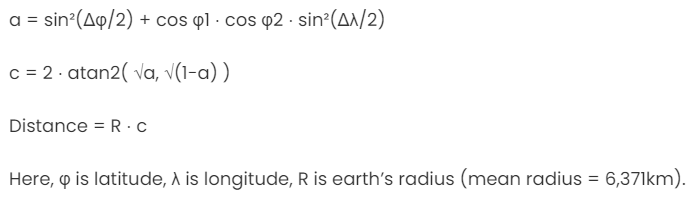


There are various outlier detection techniques in machine learning that are categorized as supervised methods, semi-supervised methods, and unsupervised methods. The below image shows a list of a few ways on how to detect outliers in machine learning under the three categories.



* Haversine

The Haversine formula calculates the shortest distance between two points on a sphere using their latitudes and longitudes measured along the surface. It is important for use in navigation.



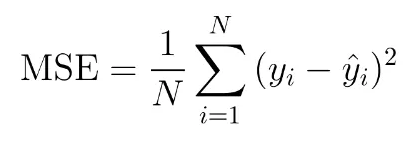
* Mathplotlib

Matplotlib is an open-source plotting library in Python introduced in the year 2003. It is a very comprehensive library and designed in such a way that most of the functions for plotting in MATLAB can be used in Python.

* Mean Squared Error

The Mean Squared Error (MSE) is perhaps the simplest and most common loss function, often taught in introductory Machine Learning courses. To calculate the MSE, you take the difference between your model’s predictions and the ground truth, square it, and average it out across the whole dataset.

The MSE will never be negative, since we are always squaring the errors. The MSE is formally defined by the following equation:



Where N is the number of samples we are testing against.

* Code Explanation:

*#import libraries*

**import** pandas **as** pd

**import** numpy **as** np

**import** matplotlib.pyplot **as** plt

**import** warnings

*#We do not want to see warnings*

warnings**.**filterwarnings("ignore")

Import all libraries required for uber ride prediction

*#import data*

data **=** pd**.**read\_csv("uber.csv")

*#Create a data copy*

df **=** data**.**copy()

*#Print data*

df**.**head

*#Get Info*

df**.**info()

Read values from dataset with the help of pandas library.

*#pickup\_datetime is not in required data format*

df["pickup\_datetime"] **=** pd**.**to\_datetime(df["pickup\_datetime"])

df**.**info()

*#Statistics of data*

df**.**describe()

*#Correlation*

df**.**corr()

corr() function to find the correlation among the columns in the Dataframe using ‘Pearson’ method(standard correlation coefficient).

*#Drop the rows with missing values*

df**.**dropna(inplace**=True**)

plt**.**boxplot(df['fare\_amount'])

Box Plot is the visual representation of the depicting groups of numerical data through their quartiles. Boxplot is also used for detect the outlier in data set. It captures the summary of the data efficiently with a simple box and whiskers and allows us to compare easily across groups. Boxplot summarizes a sample data using 25th, 50th and 75th percentiles. These percentiles are also known as the lower quartile, median and upper quartile.

*#Remove Outliers*

q\_low **=** df["fare\_amount"]**.**quantile(0.01)

q\_hi **=** df["fare\_amount"]**.**quantile(0.99)

df **=** df[(df["fare\_amount"] **<** q\_hi) **&** (df["fare\_amount"] **>** q\_low)]

With the help of quantile() function, we can define range or limit for data to remove an outliers from the dataset.

**from** sklearn.linear\_model **import** LinearRegression

lrmodel **=** LinearRegression()

lrmodel**.**fit(x\_train, y\_train)

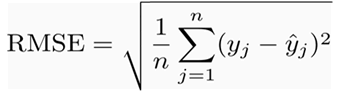
Import the class LinearRegression from sklearn.linear\_model. Use the class sklearn.linear\_model.LinearRegression to perform linear and polynomial regression and make predictions accordingly. The next step is to create a linear regression model and fit it using the existing data. Create an instance of the class LinearRegression. This statement creates the variable model as an instance of LinearRegression. It can provide several optional parameters to LinearRegression. In other words, .fit() fits the model. It returns self, which is the variable model itself.

**from** sklearn.metrics **import** mean\_squared\_error

lrmodelrmse **=** np**.**sqrt(mean\_squared\_error(predict, y\_test))

print("RMSE error for the model is ", lrmodelrmse

Root mean squared error (RMSE) is a quadratic scoring rule that also measures the average magnitude of the error. It’s the square root of the average squared differences between prediction and actual observation.



In the scikit learn library, sklearn.metrics has a mean\_squared\_error function. The RMSE is just the square root of values it returns.

**from** sklearn.ensemble **import** RandomForestRegressor

rfrmodel **=** RandomForestRegressor(n\_estimators **=** 100, random\_state**=**101)

After all the work of data preparation, creating and training the model is pretty simple using Scikit-learn. We import the random forest regression model from skicit-learn, instantiate the model, and fit (scikit-learn’s name for training) the model on the training data.

rfrmodel\_rmse **=** np**.**sqrt(mean\_squared\_error(rfrmodel\_pred, y\_test))

print("RMSE value for Random Forest is:",rfrmodel\_rmse)

sklearn.metrics has a mean\_squared\_error function. The RMSE is just the square root it returns.

=====================================================================

**Conclusion:-**

Thus we have studied how to predict the price of Uber ride.

=====================================================================