CAB FARE PREDICTION

Mohd Abdul Baseer

04 July 2019

Contents

1. Introduction

1.1 Problem Statement

1.2 Data

1.3 Exploratory Data Analysis

2. Methodology

2.1 Data Pre-Processing

2.1.1 Missing value analysis

2.1.2 Date time Conversion

2.1.3 Distance Conversion

2.1.4 Outlier Analysis

2.1.4 Visualizations

2.1.5 Feature Selection

2.1.6 Feature Scaling

2.1.7 Data Sampling

2.2 Modeling

2.2.1 Linear Regression

2.2.2 KNN Regression

2.2.3 Decision Tree Regression

2.2.4 Random Forest

2.2.5 Model Evaluation

3. Conclusion

3.1 Model selection

References

Instructions

1. Introduction

1.1.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.

1.2. Data

There are two datasets Train and Test given to us where Train is used to train the model and Test does not have the target variable , predictions are done on Test data set

Train Dataset
Number of attributes:

- pickup_datetime - timestamp value indicating when the cab ride started.
- pickup_longitude - float for longitude coordinates of where the cab ride started.
- pickup_latitude - float for latitude coordinate of where the cab ride started.
- dropoff_longitude - float for longitude coordinates of where the cab ride ended.
- dropoff_latitude - float for latitude coordinate of where the cab ride ended.
- passenger_count - an integer indicating the number of passengers in the cab ride.
- Fare_amount (Target variable)- amount charged for particular ride

Missing Values: Yes

	fare_amount	pickup_datetime	pickup_longitude	pickup_latitude	dropoff_longitude	dropoff_latitude	passenger_count
0	4.5	2009-06-15 17:26:21 UTC	-73.844311	40.721319	-73.841610	40.712278	1.0

1	16.9	2010-01-05 16:52:16 UTC	-74.016048	40.711303	-73.979268	40.782004	1.0
2	5.7	2011-08-18 00:35:00 UTC	-73.982738	40.761270	-73.991242	40.750562	2.0

Head of Train Dataset (above)

Test Dataset

Number of attributes:

- pickup_datetime - timestamp value indicating when the cab ride started.
- pickup_longitude - float for longitude coordinates of where the cab ride started.
- pickup_latitude - float for latitude coordinate of where the cab ride started.
- dropoff_longitude - float for longitude coordinates of where the cab ride ended.
- dropoff_latitude - float for latitude coordinate of where the cab ride ended.
- passenger_count - an integer indicating the number of passengers in the cab ride.

Missing Values: No

	pickup_datetime	pickup_longitude	pickup_latitude	dropoff_longitude	dropoff_latitude	passenger_count
0	**2015-01-27 13:08:24 UTC**	**-73.973320**	**40.763805**	**-73.981430**	**40.743835**	**1**
1	**2015-01-27 13:08:24 UTC**	**-73.986862**	**40.719383**	**-73.998886**	**40.739201**	**1**
2	**2011-10-08 11:53:44 UTC**	**-73.982524**	**40.751260**	**-73.979654**	**40.746139**	**1**
3	**2012-12-01 21:12:12 UTC**	**-73.981160**	**40.767807**	**-73.990448**	**40.751635**	**1**
4	**2012-12-01 21:12:12 UTC**	**-73.966046**	**40.789775**	**-73.988565**	**40.744427**	**1**

Head of test dataset

Test dataset does not have any missing values and
it does not contain the target variable “fare_amout”

1.3 Exploratory Data Analysis

Here we analyze the data and Sketch out the Required changes which need to be done With respective to the problem statement for effective execution

Describing the Data set

	pickup_longitude	pickup_latitude	dropoff_longitude	dropoff_latitude	passenger_count
count	16067.000000	16067.000000	16067.000000	16067.000000	16012.000000
mean	-72.462787	39.914725	-72.462328	39.897906	2.625070
std	10.578384	6.826587	10.575062	6.187087	60.844122
min	-74.438233	-74.006893	-74.429332	-74.006377	0.000000
25%	-73.992156	40.734927	-73.991182	40.734651	1.000000
50%	-73.981698	40.752603	-73.980172	40.753567	1.000000
75%	-73.966838	40.767381	-73.963643	40.768013	2.000000
max	40.766125	401.083332	40.802437	41.366138	5345.000000

```
#describing train dataset
```

```
#DEscribing the Target Variable fare_amount
```

```
count    16043
unique     468
top        6.5
freq       759
Name: fare_amount, dtype: object
```

Analysis :

We understood that the latitudes and longitudes need to be converted into distance and the pickup_datetime need to be separated into date , time, day , day of week, etc..

Null Hypothesis : After Brainstorming we came to a Null hypothesis that Fare amount increases with the increase in distance and it time of the Ride also affects the Fare amount

2.1 Data Pre-Processing

Data preprocessing is a **data** mining technique that involves transforming raw **data** into an understandable format. Real-world **data** is often incomplete, inconsistent, and/or lacking in certain behaviors or trends, and is likely to contain many errors. **Data preprocessing** is a proven method of resolving such issues.

Steps involved into Data Pre processing

- **Missing value analysis**
- **Outlier analysis**
- **Visualization**
- **Feature selection**
- **Feature selection**
- **Data Sampling**

2.1.1 Missing value analysis

Missing Value Analysis. The Missing Value Analysis procedure performs three primary functions: Describes the pattern of missing data. ... Fills in (imputes) missing values with estimated values using regression or EM methods; however, multiple imputation is generally considered to provide more accurate results.

Now we analyze the MIssing values in our Train data set

	Variables	Missing_percentage
0	passenger_count	0.342317
1	fare_amount	0.155598
2	pickup_datetime	0.000000
3	pickup_longitude	0.000000
4	pickup_latitude	0.000000
5	dropoff_longitude	0.000000

6 dropoff_latitude 0.000000

We can observe that we are having missing values only in Passanger_count and fare_amount @ 0.342% and @ 0.155598 Respectively

Now the Missing values are imputed by using Mean/Median/Knn any of these which Performs better

In this process an observation from the data is removed and replaced by na and then we perform imputation with different methods , whichever method gives closer results to the existing observation , that method is frozen and applied on all the missing values

Now depending upon the observation on which we are testing we get better results using Knn imputation in R and we get better results from Mean in Python

Results It depends upon which observation we are examining

After imputation of missing value analysis we do not have any missing values in our data set

There are no missing values in data now data:

fare_amount 0

pickup_datetime 0

pickup_longitude 0

pickup_latitude 0

dropoff_longitude 0

dropoff_latitude 0

passenger_count 0

dtype: int64

Hence we are done with Missing value analysis

2.1.2 Date time Conversion

We have a variable which is pickup_datetime which is needed to be converted into datetime format and the date , time ,day etc need to be extracted from the particular data

Pickup_datetime after converting into Datetime format

0 2015-01-27 13:08:24

1 2015-01-27 13:08:24
2 2011-10-08 11:53:44
3 2012-12-01 21:12:12
4 2012-12-01 21:12:12

Name: pickup_datetime, dtype: datetime64[ns]

After breaking down our Datetime data into meaningful data we get new variables like **year,month ,day, day of the week, and hour**
Now we can Remove the Date time variable to reduce the complexity

Head of the Train Data after Extracting our new datetime variables and and Removing the Pickup_datetime variable

	fare_amount	pickup_longitude	pickup_latitude	dropoff_longitude	dropoff_latitude	passenger_count	year	month	day	weekday	hour
0	4.5	-73.844311	40.721319	-73.841610	40.712278	1.0	2009.0	6.0	15.0	0.0	17.0
1	16.9	-74.016048	40.711303	-73.979268	40.782004	1.0	2010.0	1.0	5.0	1.0	16.0
2	5.7	-73.982738	40.761270	-73.991242	40.750562	2.0	2011.0	8.0	18.0	3.0	0.0
3	7.7	-73.987130	40.733143	-73.991567	40.758092	1.0	2012.0	4.0	21.0	5.0	4.0
4	5.3	-73.968095	40.768008	-73.956655	40.783762	1.0	2010.0	3.0	9.0	1.0	7.0

Now the datetime data is easily understandable in the form of new variables **year, month,day,day of the week and hour**

Due to this conversion the understanding of the has increased and then we can get the relationship between the New variables and the Target variable which is Fare_amount

Hence after this process we get the simplified version of the data

2.1.3 Distance Conversion

We aree having variables like

pickup_longitude	pickup_latitude	dropoff_longitude	dropoff_latitude

WE can not understand these variables hence we need to convert them ibnto meaningful vareables shuch that we can get the meaning from the vareables

After analysing the varibles we understand that these vareables can be used to extract the distance from them

Now to extarct the distance from these Variables we use the Formula of Haversine

R = 6373.0

dlon = radians(lon2) - radians(lon1)

dlat = radians(lat2) - radians(lat1)

a = (sin(dlat/2))**2 + cos(radians(lat1)) * cos(radians(lat2)) * (sin(dlon/2))**2

c = 2 * atan2(sqrt(a), sqrt(1-a))

distance = R * c

In R we can calculate this Using a library called “geosphere”

Now using the Above methods we calculate the distance between the given latitudes and longitudes in the updated dataset below

	fare_amount	pickup_longitude	pickup_latitude	dropoff_longitude	dropoff_latitude	passenger_count	year	month	day	weekday	hour	distance
0	4.5	-73.844311	40.721319	-73.841610	40.712278	1	2009	6	15	0	17	1.030764
1	16.9	-74.016048	40.711303	-73.979268	40.782004	1	2010	1	5	1	16	8.4501

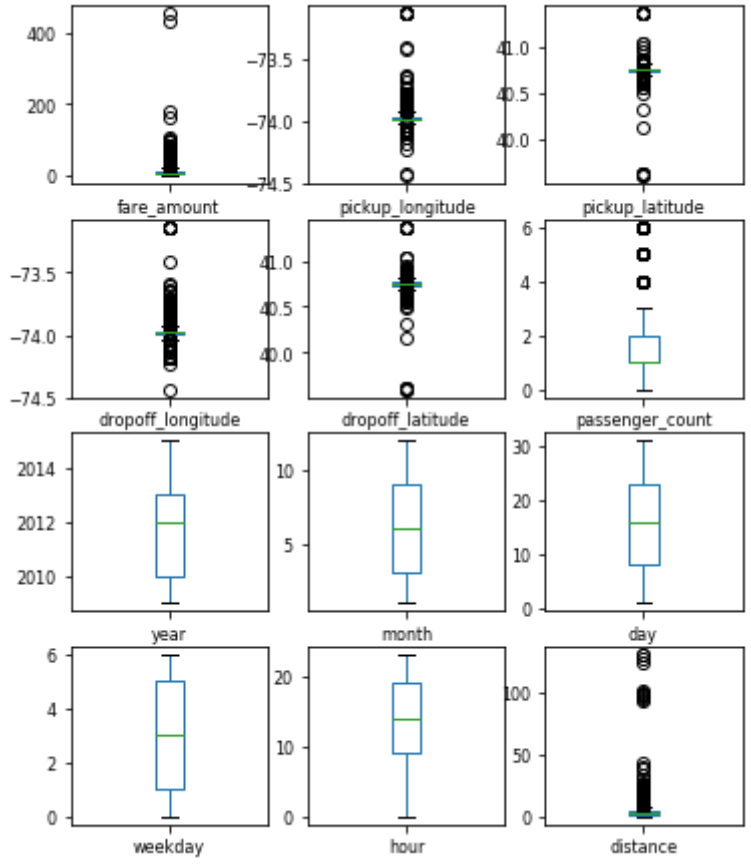
												34
2	5.7	-73.982738	40.761270	-73.991242	40.750562	2	2011	8	18	3	0	1.389525

2.1.4 Outlier Analysis

The analysis of outlier data is referred to as outlier mining. Outliers may be detected using statistical tests that assume a distribution or probability model for the data, or using distance measures where objects that are a substantial distance from any other cluster are considered outliers.

In this process of Outlier analysis we first detect outliers in the data then we impute that outliers with na then we remove them or also can be imputed using the imputation techniques

We use Boxplot to plot the Outliers from tthe data



After plotting the Outliers from the data now we remove the outliers which fall beyond 75% and below 25%

The removed Outliers are replaced with na and then they are removed or they cann be imputed

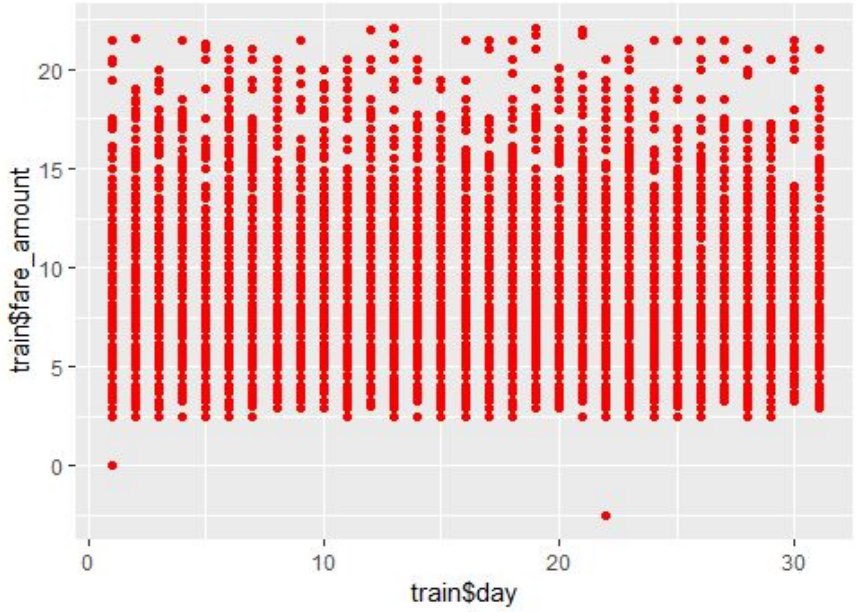
In R and Python we removed the outliers from the data

Once the Outliers are Removed then we get data without any irregularities which is effective for our modeling

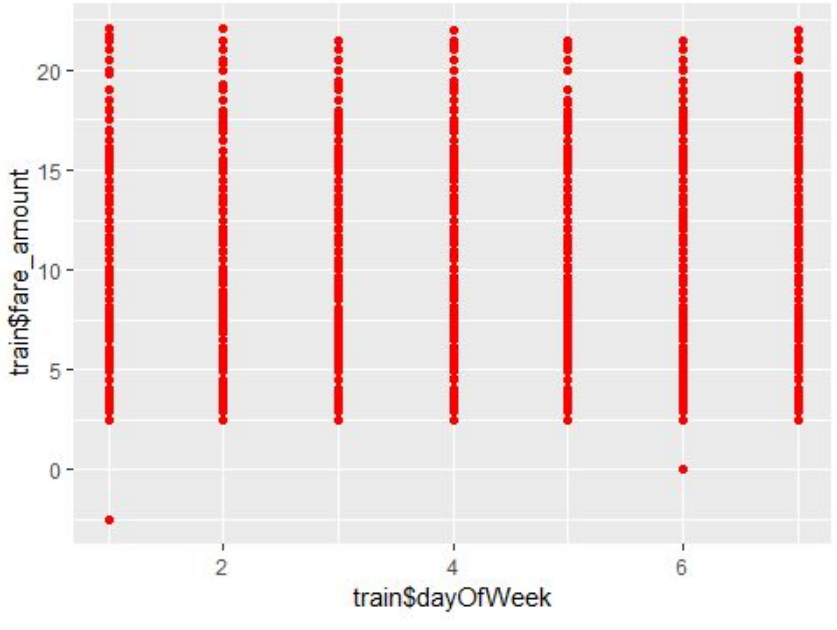
2.1.4 Visualizations

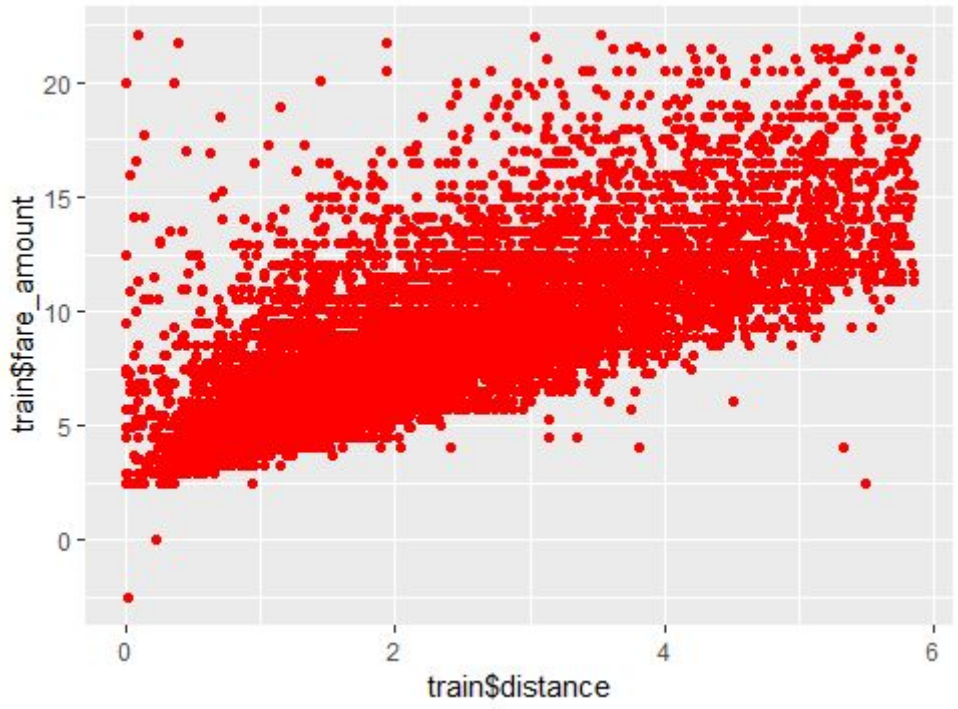
Numerical data may be encoded using dots, lines, or bars, to visually communicate a quantitative message. Effective visualization helps users analyze and reason about data and evidence. It makes complex data more accessible, understandable and usable. Users may have particular analytical tasks, such as making comparisons or understanding causality, and the design principle of the graphic (i.e., showing comparisons or showing causality) follows the task. Tables are generally used where users will look up a specific measurement, while charts of various types are used to show patterns or relationships in the data for one or more variables.

Data visualization is both an art and a science.

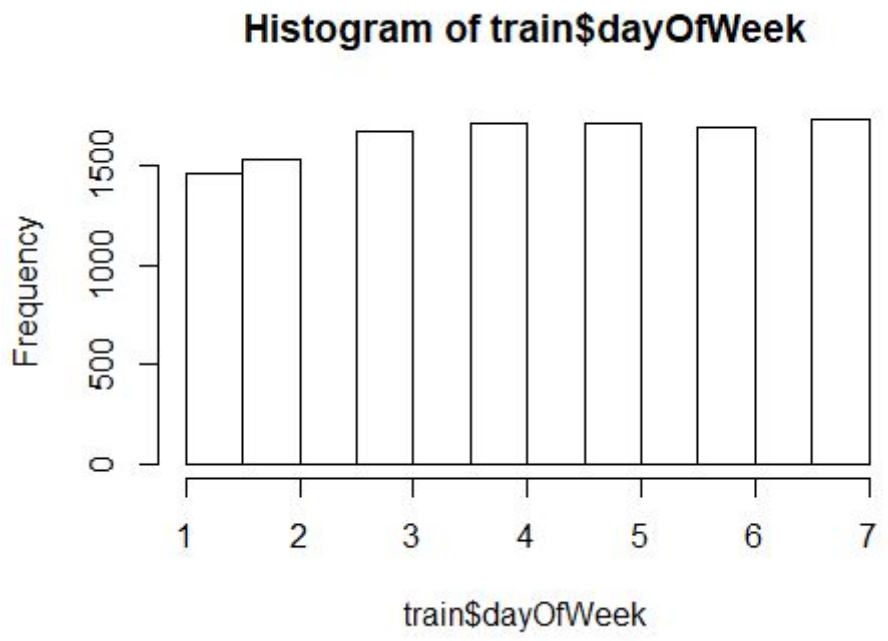


From the above plot we can see that fare amount is a bit less in ending days

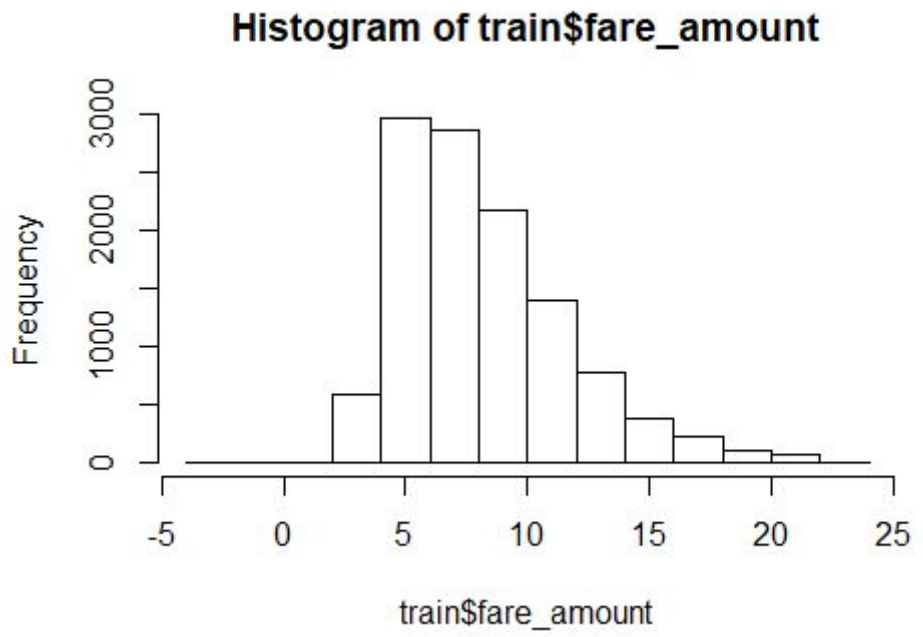




We can clearly see that as the distance increases the amount of the fare also increases

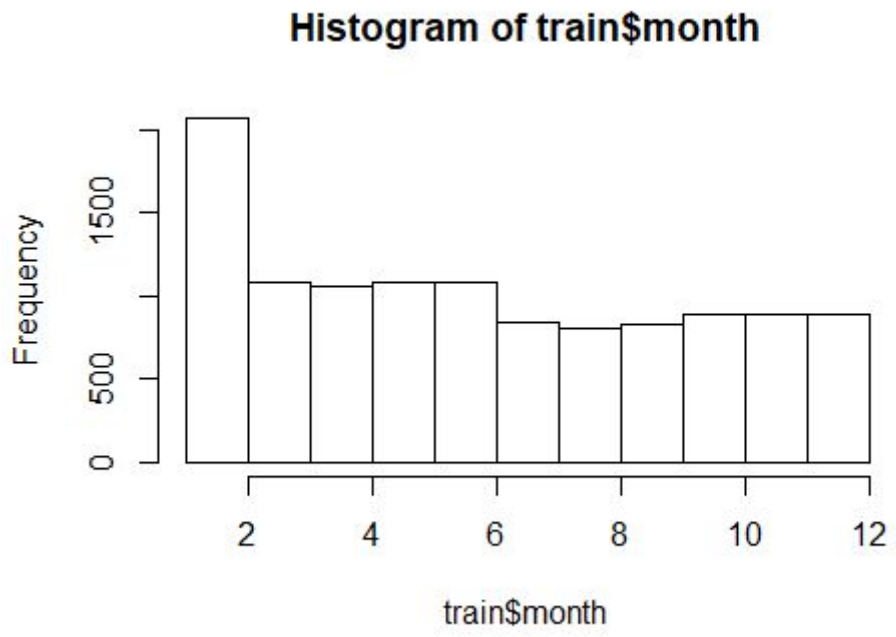


The frequency of rides are a bit less on the on the first two days of the week

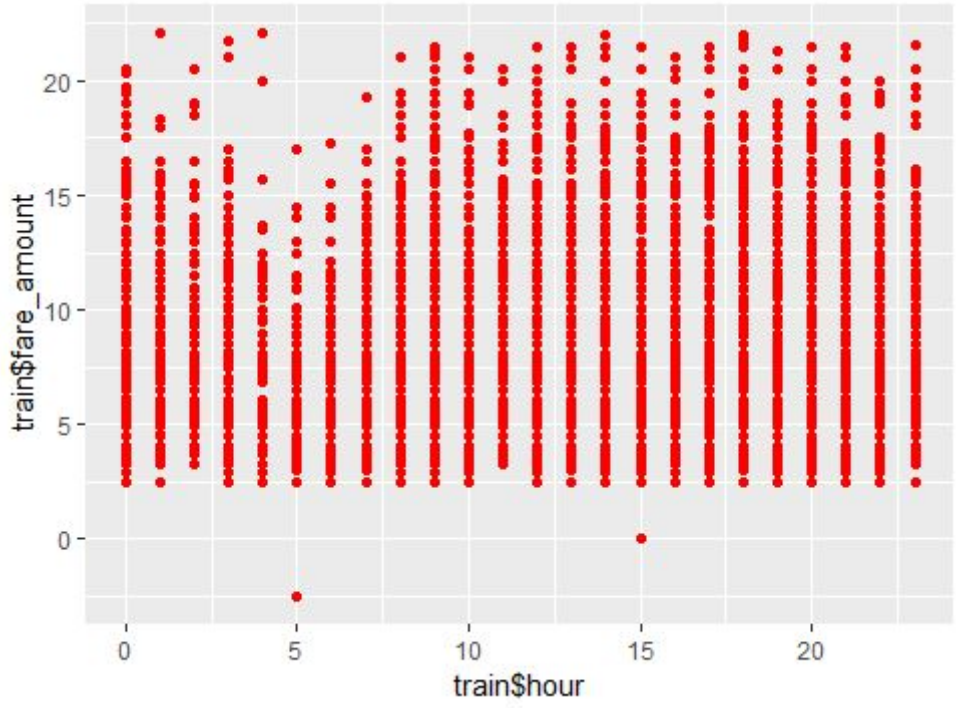


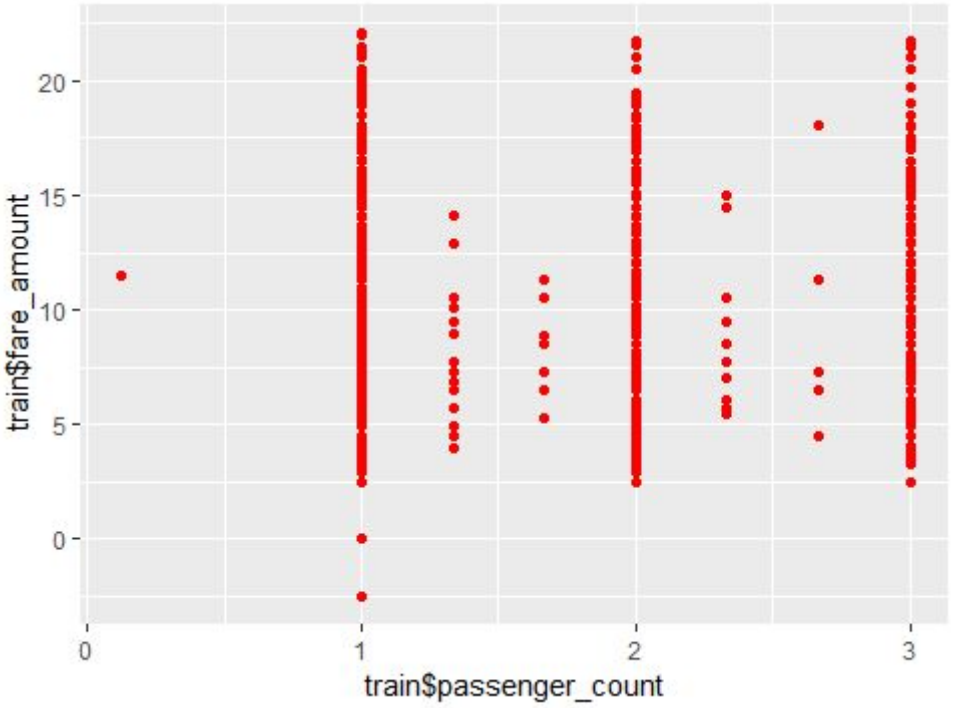
We can observe that the histogram is centerd skewed and frequency increases from 4 to 12

In the hours 2 to 6 the frequency of rides are low as it is night time



Starting two months have higher frequency of raids





Majority of the passengers are 1, 2,3 the other values may have occurred due to imputation of knn

2.1.5 Feature Selection

The Process of selecting the meaningful features in the data and removing the useless data from the data is known as feature selection

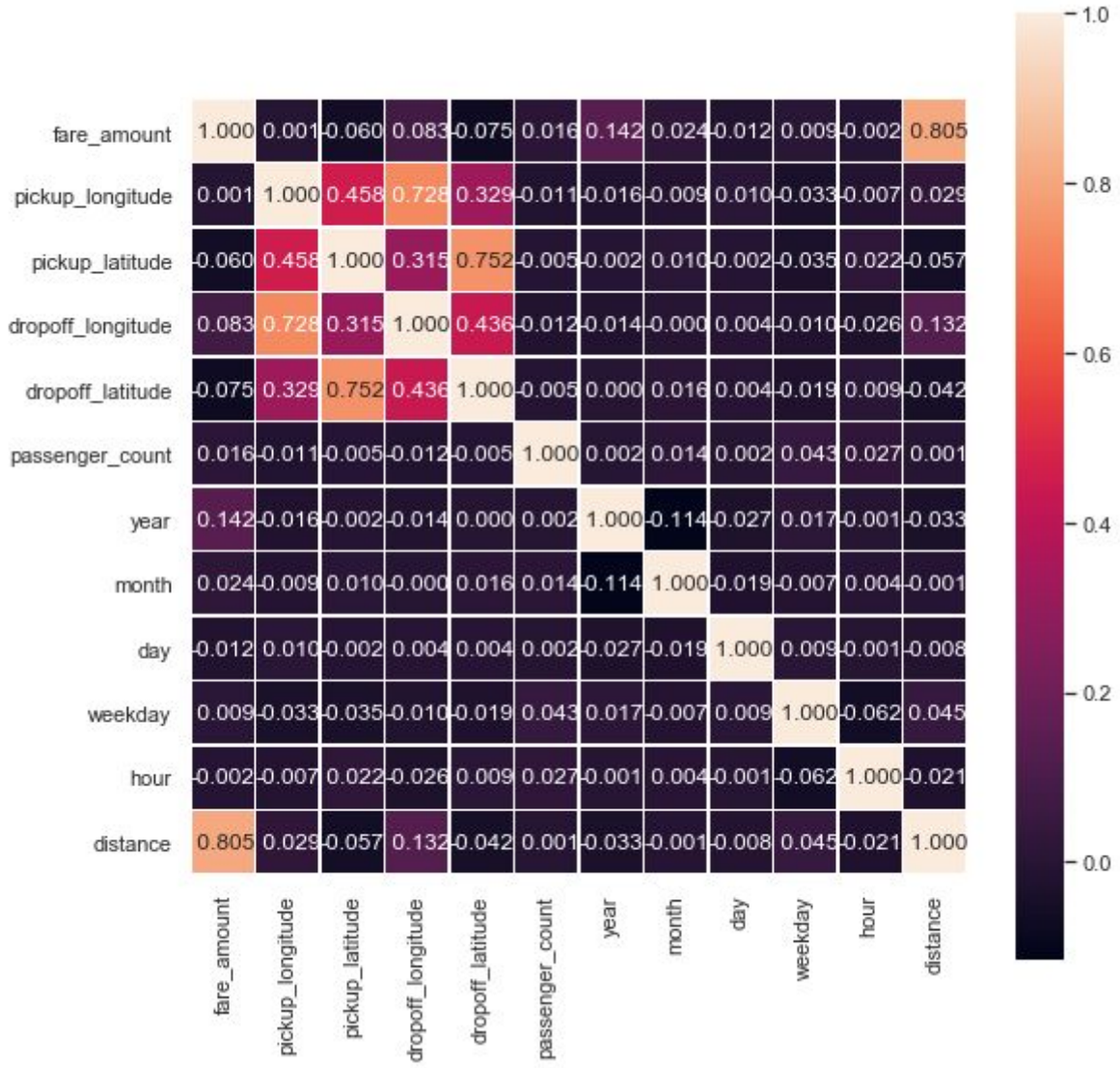
There are two major ways of feature selection

1.correlation plot with backward elimination

2.Chi-square test of indepencies

1. In correlation plot we remove the variables which are very highly or lowly corelated to each other

Correlation Plot of the variables



2. Chi-square test : this test is used to check the independence of each variable in context with the target variable

Through applying this test we remove the variables which have p value greater than 0.05

Dimension Reduction :

analysing the above plot we remove the latitude and longitude variables from the data

And from the chi square test we remove “day” variable and “passenger_count ” variable from the data

Data after dimension reduction

	fare_amount	passenger_count	year	month	weekday	hour	distance
0	4.5	1.0	2009	6	3.486624	17.000000	1.030764
1	16.9	1.0	2010	1	1.000000	16.000000	8.450134
2	5.7	2.0	2011	8	3.000000	14.085337	1.389525
3	7.7	1.0	2012	4	5.000000	4.000000	2.799270
4	5.3	1.0	2010	3	1.000000	7.000000	1.999157

Now the variables are reduced through correlation analysis and chi square test

2.1.6 Feature Scaling

Feature scaling can be done in two types

1. **Normalization :** if the data is not normally distributed then normalization is done and the values are compressed between 0 and 1
2. **Standardization** : if the data is normally distributed then Standardization is done and the values are compressed into -1 and 1

All the data except the target variable is Normalised in our data

	passenger_count	year	month	weekday	hour	distance
9773	0.0	0.666667	0.181818	0.6	0.772727	0.131972
2393	0.8	0.500000	0.272727	0.0	0.000000	0.090939
11093	0.0	0.833333	0.909091	0.0	0.590909	0.059197
15210	0.0	0.166667	0.363636	0.0	0.594788	0.211983

2.1.7 Data Sampling

The Process of dividing the data into test and train is known as data sampling

There are various types of sampling techniques

We used random sampling for our project

The data is divided in a ratio of 8:2

Train dataset gets 80% of data

Test dataset gets 20% of data

The models are first trained on the train dataset and then they are tested on the test dataset and the accuracy or the score is calculated in the test dataset comparing the predicted values to the original values

Hence we are done with the data pre processing which is the most important and crucial part of data science

The next step of data science is Machine learning modelling

2.2 Modeling

2.2.1 Linear Regression

Multiple linear regression (MLR), also known simply as multiple regression, is a statistical technique that uses several explanatory variables to predict the outcome of a response variable. The goal of multiple linear regression (MLR) is to model the linear relationship between the explanatory (independent) variables and response (dependent) variable. In essence, multiple regression is the extension of ordinary least-squares (OLS) regression that involves more than one explanatory variable.

The Formula for Multiple Linear Regression Is

$$yi=\beta 0+\beta 1xi1+\beta 2xi2+...+\beta pxip+\epsilon$$

where, for $i=n$ observations:

***yi*=dependent variable**
***xi*=expanatory variables**
***β*0=y-intercept (constant term)**
***Bp*=slope coefficients for each explanatory variable**
***ϵ*=the model's error term (also known as the residuals)**

Python results
#Leniear Reggresion rmse: 2.367790983098614
#Linear Regression mape: 4.509954219310796
#R^@score = 0.6705294516779501
R Results

#linear Regression MAPE = 18.6%

#this model is dropped in Python as rmse is high when compared
#this model is dropped in R as MAPE is high when compared

2.2.2 KNN Regression

KNN can be used for both classification and regression problems. The algorithm uses

'feature similarity' to predict values of any new data points. This means that the new point is assigned a value based on how closely it resembles the points in the training set. From our example, we know that ID11 has height and age similar to ID1 and ID5, so the weight would also approximately be the same.

Had it been a classification problem, we would have taken the mode as the final prediction. In this case, we have two values of weight – 72 and 77. Any guesses how the final value will be calculated? The average of the values is taken to be the final prediction.

Below is a stepwise explanation of the algorithm:

1. First, the distance between the new point and each training point is calculated
2. The closest k data points are selected (based on the distance). In this example, points 1, 5, 6 will be selected if value of k is 3.
3. The average of these data points is the final prediction for the new point. Here, we have weight of ID11 = (77+72+60)/3 = 69.66 kg.

Python Results

#n_neighbours : 100 ----KNN rmse:
2.6079425339636284

#n_neighbours : 100 ----KNN mape:
5.021265945581793

#R^2score = 0.6003075829633684

#this model is dropped in Python as rmse is high when compared
#this model is dropped in R as MAPE is high when compared

2.2.3 Decision Tree Regression

Decision tree builds regression or classification models in the form of a tree structure. It breaks down a dataset into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed. The final result is a tree with decision nodes and leaf nodes. A decision node (e.g., Outlook) has two or more branches (e.g., Sunny, Overcast and Rainy), each representing values for the attribute tested. Leaf node (e.g., Hours Played) represents a decision on the numerical target. The topmost decision node in a tree which corresponds to the best predictor called root node. Decision trees can handle both categorical and numerical data.

Python Results

#Decision Tree rmse: 2.493424498531671

#Decision Tree mape: 4.766396172879622

#R^2score= 0.6346388897092561

R results

#decision tree MAPE = 20.4%

#this model is dropped in python as rmse is high when compared
#this model is dropped in R as MAPE is high when compared

2.2.4 Random Forest

Let's understand Random Forest step by step:

Step 1: Samples are taken repeatedly from the training data so that each data point is having an equal probability of getting selected, and all the samples have the same size as the original training set.

Let's say we have the following data:

x= 0.1,0.5,0.4,0.8,0.6, y=0.1,0.2,0.15,0.11,0.13 where x is an independent variable with 5 data points and y is dependent variable.

Now Bootstrap samples are taken with replacement from the above data set. n_estimators is set to 3 (no of tree in random forest), then:

The first tree will have a bootstrap sample of size 5 (same as the original dataset), assuming it to be: x1={0.5,0.1,0.1,0.6,0.6} likewise

x2={0.4,0.8,0.6,0.8,0.1}

x3={0.1,0.5,0.4,0.8,0.8}

Step 2: A Random Forest Regressor model is trained at each bootstrap sample drawn in the above step, and a prediction is recorded for each sample.

Step 3: Now the ensemble prediction is calculated by averaging the predictions of the above trees producing the final prediction

#Random Forest rmse: 2.157257716683359

#Random Forest mape: 4.303135135131023

#r^2score= 0.726514732384569

R result

#random forest MAPE = 18.5%

#this model is accepted in Python as rmse is low when compared
#this model is accepted in R as MAPE is low when compared

2.2.5 Model Evaluation :

the process of evaluating the model is known as model evaluation

For regression problems, we have metrics like:

- MSE.
- RMSE.
- MAPE.
- Rsquare.

We are choosing RMSE and R square for our project. Why RMSE over MAPE? Because, in RMSE, as the errors are squared before they are averaged, the RMSE gives a relatively high weightage to large errors, another reason is, RMSE penalizes large errors. For the above-mentioned reasons, we choose RMSE over MAPE

We use MAPE in R and in Python we use r^2 score and rmse to evaluate a model

We are going to use RMSE and R square as our error metrics to evaluate our models. In Python

RMSE – Simply said, it is the sum of calculated errors.

Rsquare – Simply defined, correlation of original and predicted values.

MAPE- Mean absolute percentage error

3. Conclusion

3.1 Model selection

Model selection is done by comparing and evaluating the best performing model of all

In python

####################MODEL SELECTION ##########################

#n_neighbours : 100 ----KNN rmse: 2.6079425339636284

#n_neighbours : 100 ----KNN mape: 5.021265945581793

#R^2score = 0.6003075829633684

#Leniear Reggresion rmse: 2.367790983098614

#Linear Regression mape: 4.509954219310796

#R^@score = 0.6705294516779501

#Decision Tree rmse: 2.493424498531671

#Decision Tree mape: 4.766396172879622

#R^2score= 0.6346388897092561

#Random Forest rmse: 2.157257716683359

#Random Forest mape: 4.303135135131023

#r^2score= 0.7265147323845697

Hence as the rmse and MAPE is minnimum and R^2 score is maximum for random forest we choose Random Forest as the best fitting model

Random forest Predicted Result in Python

	fare_amount
0	7.310591
1	7.370473
2	4.745531

In R

Analyzing mape

#random forest = 18.5%

#linear Regression = 18.6%

#decision tree = 20.4%

#as Random forest is having minimum mape we select Random forest as a best fit for our data

Random forest Predicted results in R

1	2	3
14.51751	14.51751	12.64721

Hence we have successfully predicted the fare_amount of the cab using Random Forest Regression in both R and Python

References:

• Edwisor.com

• Edwisor Community.

• Google

• To get idea to deal with pickup_datetime variable, we used this source - https://stackoverflow.com/questions/46428870/how-to-handle-date-variable-in-machinelearning-data-pre-processing

• Coming to latitude and longitude variables, we got idea how to use them to go forward using these links
https://gis.stackexchange.com/questions/119846/calculating-distance-between-latitudeand-lo

ngitude-points-using-python/119854

############## Instructions ##########################

=>R project is made in R Studio. The user can run it in R Studio or any other environment which uses R.

=>Python project is made in jupyter environment.

=>The file is a .ipynb notebook file extension file.

=>.ipynb file can be checked in spyder environment also.