***INDEX***

**Introduction to the data**

1) Problem Statement 2

2) Data 2

**Methods applied in pre-processing**

1) Missing Values Analysis 4

2) Outliers 5

3) Feature Engineering 10

4) Feature Selection and Feature Scaling

**Modeling**

1) Model Selection 13

2) Linear Regression 13

3) Decision Tree 15

**Model Evaluation**

1) RMSE 17

**Deployment** 18

**Introduction to the data:-**

* **Problem Statement**

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

* **Data**

Our task is to build a good regression model on the data. The data contains 7 variables and 16067 observations.

Here are few observations of our data.

|  |
| --- |
|  |
| |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | | fare\_amount | pickup\_datetime | pickup\_longitude | pickup\_latitude | dropoff\_longitude | dropoff\_latitude | passenger\_count | | | 4.5 | 2009-06-15 17:26:21 UTC | -73.8443 | 40.72132 | -73.8416 | 40.71228 | 1 |  | | 16.9 | 2010-01-05 16:52:16 UTC | -74.016 | 40.7113 | -73.9793 | 40.782 | 1 |  | | 5.7 | 2011-08-18 00:35:00 UTC | -73.9827 | 40.76127 | -73.9912 | 40.75056 | 2 |  | | 7.7 | 2012-04-21 04:30:42 UTC | -73.9871 | 40.73314 | -73.9916 | 40.75809 | 1 |  | | 5.3 | 2010-03-09 07:51:00 UTC | -73.9681 | 40.76801 | -73.9567 | 40.78376 | 1 |  | | 12.1 | 2011-01-06 09:50:45 UTC | -74.001 | 40.73163 | -73.9729 | 40.75823 | 1 |  | | 7.5 | 2012-11-20 20:35:00 UTC | -73.98 | 40.75166 | -73.9738 | 40.76484 | 1 |  | | 16.5 | 2012-01-04 17:22:00 UTC | -73.9513 | 40.77414 | -73.9901 | 40.75105 | 1 |  | |  | 2012-12-03 13:10:00 UTC | -74.0065 | 40.72671 | -73.9931 | 40.73163 | 1 |  | | 8.9 | 2009-09-02 01:11:00 UTC | -73.9807 | 40.73387 | -73.9915 | 40.75814 | 2 |  | |

* **Pre processing**

By using “.value\_counts()” function in python, we can see that the 0o latitude and longitude have highest number of drops and pickups i.e. more than 300 times and from other individual latitude it is less than 10 times, it can be concluded that earning from this latitude and longitude (i.e., 00) would be maximum.

I checked the datatype of each column in the dataset, I found that the *“fare amount”* and *“pickup date and time”* variables have the datatype *“object”* and other variables have datatype *“float64”* in python, and similarly in R datatype for *“fare amount”* and *“pickup date and time”* have *“object”* and other variables have datatype *“num*”.

I converted the “fare amount” variable into the “float” type.

And also converted “pickup\_datetime” variable in datetime64[ns] datatype as it is particularly used for date and time. I also added few columns like “month”, ”day “and “hour”, I think it is important as it will give us important in which year, month, day and hour cab are mostly used and least used. I also drop the “pickup\_datetime” variable as I have derived the important variables from it, and now it is not useful to us.

* **Missing Values**

By using “ is.null().sum()” in python and in R we can see that there are 24 missing values in the “fare\_amount” variable and 55 missing values in the “passenger\_count” variable. So there is need to impute these values.

I have created a dataframe with two variables having names – “variables” and “missing val %”. Have a look of it.

variables missing val %

0 fare\_amount 0.155598

1 pickup\_datetime 0.000000

2 pickup\_longitude 0.000000

3 pickup\_latitude 0.000000

4 dropoff\_longitude 0.000000

5 dropoff\_latitude 0.000000

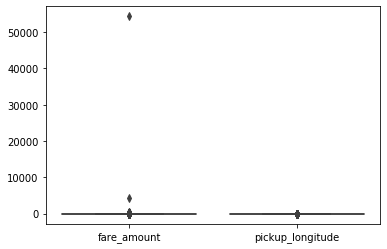
6 passenger\_count 0.342317

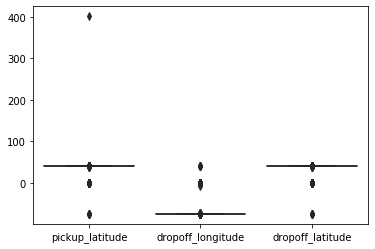
We can see that missing values percentage is very less, so we can remove these values but it is good to have more data in our hand, so I imputed the missing values by using the median method in both the columns- “fare\_amount” and “passenger\_count”. Firstly, I removed 1001th value of “pickup\_latitude” and then try different methods-like mean method, median method and knn method to fill this value and then check which value is closest to it, I got median value closest to the original value and I imputed that value. After this I imputed original value at 1001th place.

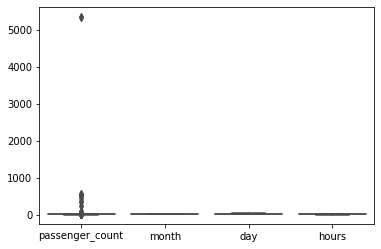
* **Outliers**

It is always good to detect the outliers, as the outliers impact the model and our predictions can go wrong, so I used firstly boxplot to see is there any outlier or not in our dataset. I found outlier in all variables except “day”, “hour” and “month” variables.

Take a look of these outliers:-







In the above pictures we can see that the fare amount of a few observations is above 1000 which are doubted cases that, are these real or human error during entering the values, let’s assume these are real observations but these are few values and are outliers for us and there are few customers who paid this much of amount.

Next come to “pickup\_latitude” here are some values which are 300, I haven’t see the values of latitude above 900N and 900S. So obviously, these are misprinted values again there is a need to remove these values and I have removed these and fill the values by mean method.

Similarly, I have removed all the outlier’s values from other variables like from “dropoff\_latitude”, “dropoff\_longitude” and “passenger\_count” and fill the values using mean method. Here is the chart of missing value, after removing outliers.

Variables missing values

fare\_amount 3362

pickup\_longitude 3356

pickup\_latitude 3356

dropoff\_longitude 3142

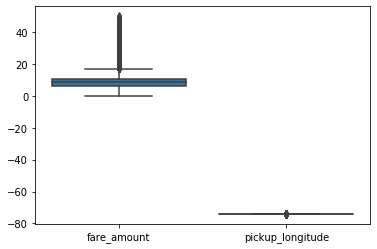
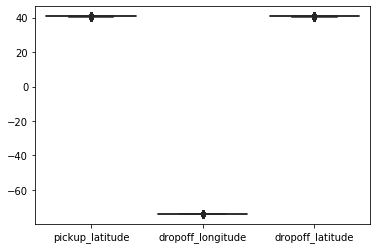
dropoff\_latitude 2592

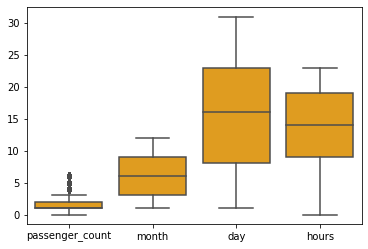
passenger\_count 1696

month 1

day 1

hours 1

After filling these missing values here is some visualization. 



You will wonder that the fare\_amount variable is showing outliers till now, but I have willingly choose that values of fare amount which are less than 50, because if we will go by inter quartile range values (e.g.; max = q75+iqr\*1.5) , where q75th is 75th percentile of the data and iqr is the difference of 75th and 25th percentile of the data then we are losing 3799 values rather than now I am losing 2962 original values and 50 is reasonable value as compared to the value of max in the above formula, which is 22.25.

And in passenger\_count variable I can easily see that these values now are not oultiers, because the maximum value of the passenger is 3.5, so I take 6 in place of max value in this feature and the minimum is 0.

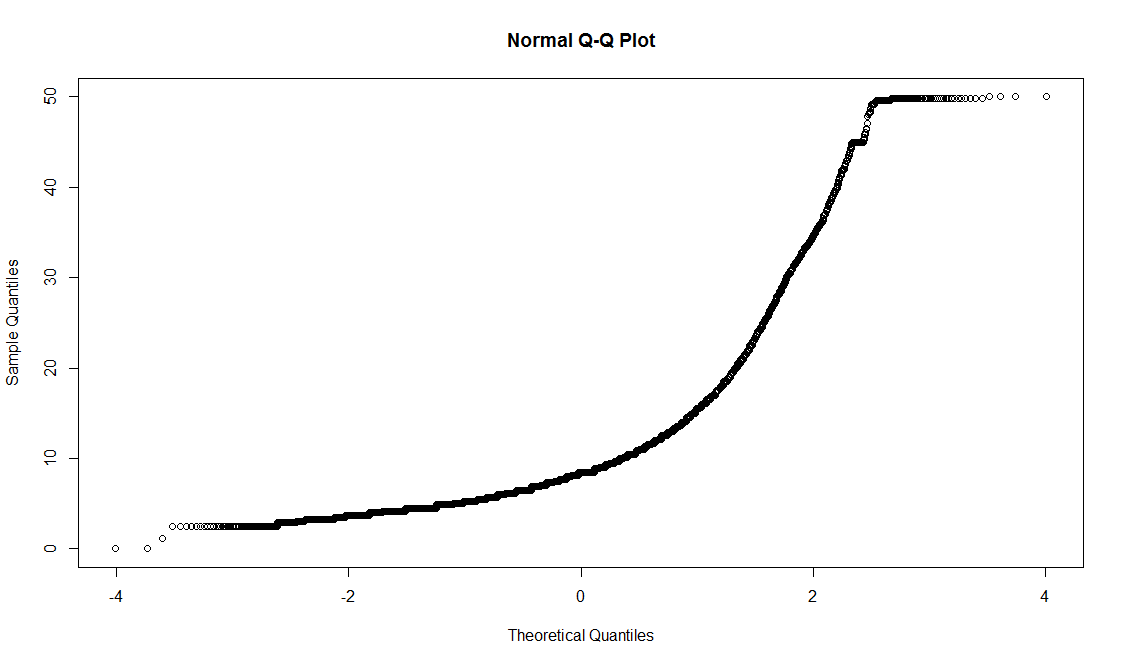
* **Feature Engineering**

We can see that there is a very important variable i.e., “Distance” missing in the dataset, as it can give us important information regarding to get the prediction of cab fare.

So, if we take the absolute value of the difference of “pickup\_lattitude” and “dropoff\_lattitude” and also take the absolute value of the pickup\_longitude and dropoff\_longitude and them by adding them we will get the approximate value of the “distance” variable.

* **Feature selection and feature scaling**

Firstly, take a look of qqplot which shows, whether the data is normally distributed or not.

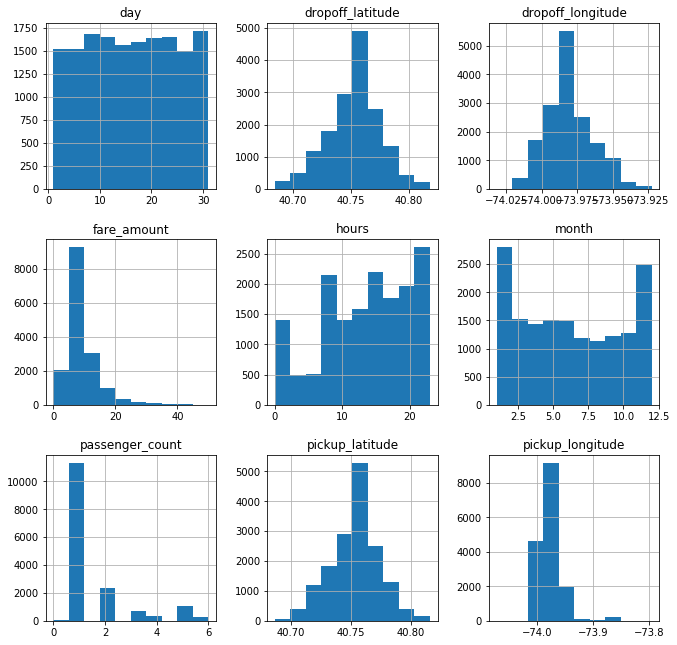


The data is not that much normally distributed.

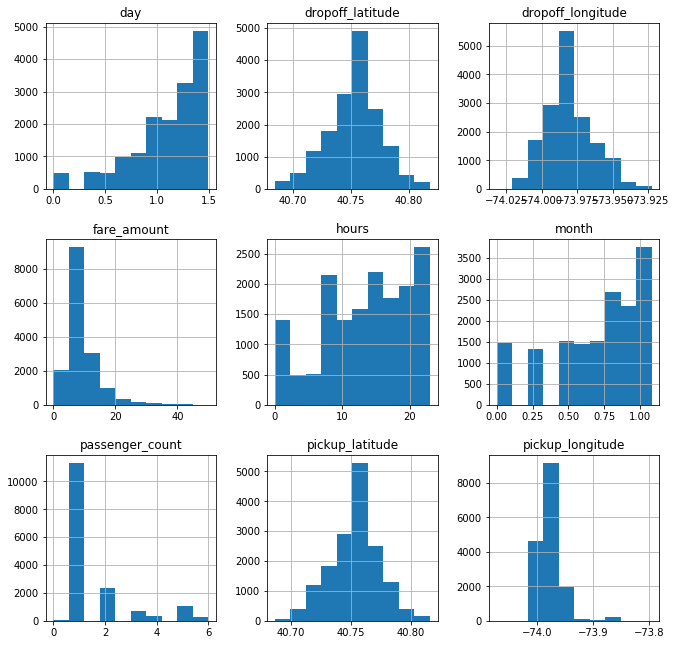
Before selection and scaling of the data I take a look of histogram to check whether the data is normally distributed or not, we can see that some of the data looks like it is normally distributed (like- dropoff\_lattitude, dropoff\_longitude and pickup\_lattitude) some are not (like- day, month and hours).

Then I take the log of variables “day” and “month” to make it in a structure somewhat like normally distributed data but it become a little change, I cannot take the log of “hours” variable as it contains “0” and it is not possible to take the log of this value.

Take a look of the histograms before taking the log of “day” and “month” variables:-



Now again take a look of the histogram after taking the log of “day” and “month” features.



Now, we can see that some features have large values, some have small and even some values are negative, so we will now try to scale this values, I have chosen min-max normalization for this as we can see that all the data is not normally distributed.

I take all the features except “passenger\_count”, “day”, “month” because “passenger-count”, have maximum value “6” and minimum is”0”. Similarly after taking the log of “day” and “month” features these values become small.

Min-max formula:-

(X - Min(X))/ (Max(X) - Min(X))

Where “X” is a general data point and “Min(X)” is minimum value of the variable and “max(X)” is the maximum value of the variable.

The min-max normalization makes the values in the range, 0-1.

* **Modeling**

Our data contains target variable as continuous variable or we can say that, there are cardinals in the dependent variable, so we will go for regression model and choose **linear regression and decision tree** as our model because we have almost all numeric variables.

**I) Linear Regression**

First, let us see the correlation between the independent variables:-

Variables VIF

1 pickup\_longitude 1.142760

2 dropoff\_longitude 1.144069

3 passenger\_count 1.000067

4 Month 1.001215

5 Hour 1.001972

> vifcor(df[,-1],th=0.5)

No variable from the 5 input variables has collinearity problem.

The linear correlation coefficients ranges between:

min correlation ( passenger\_count ~ dropoff\_longitude ): 1.184951e-05

max correlation ( dropoff\_longitude ~ pickup\_longitude ): 0.3393486

---------- VIFs of the remained variables --------

Variables VIF

1 pickup\_longitude 1.133109

2 dropoff\_longitude 1.132086

3 passenger\_count 1.000365

4 Month 1.000071

5 Hour 1.003315

We can see that vif values are very less near “1”, so the independent variables are not correlated with each other.

Linear Regression summary in python

|  |  |  |  |
| --- | --- | --- | --- |
| OLS Regression Results | | | |
| **Dep. Variable:** | y | **R-squared (uncentered):** | 0.785 |
| **Model:** | OLS | **Adj. R-squared (uncentered):** | 0.785 |
| **Method:** | Least Squares | **F-statistic:** | 1.173e+04 |
| **Date:** | Mon, 09 Dec 2019 | **Prob (F-statistic):** | 0.00 |
| **Time:** | 22:38:13 | **Log-Likelihood:** | -38742. |
| **No. Observations:** | 12853 | **AIC:** | 7.749e+04 |
| **Df Residuals:** | 12849 | **BIC:** | 7.752e+04 |
| **Df Model:** | 4 |  |  |
| **Covariance Type:** | nonrobust |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **coef** | **std err** | **t** | **P>|t|** | **[0.025** | **0.975]** |
| **x1** | 27.8509 | 0.445 | 62.573 | 0.000 | 26.978 | 28.723 |
| **x2** | -2.3167 | 0.283 | -8.188 | 0.000 | -2.871 | -1.762 |
| **x3** | 0.1539 | 0.034 | 4.573 | 0.000 | 0.088 | 0.220 |
| **x4** | 0.9415 | 0.126 | 7.492 | 0.000 | 0.695 | 1.188 |

|  |  |  |  |
| --- | --- | --- | --- |
| **Omnibus:** | 5101.363 | **Durbin-Watson:** | 2.011 |
| **Prob(Omnibus):** | 0.000 | **Jarque-Bera (JB):** | 28034.305 |
| **Skew:** | 1.837 | **Prob(JB):** | 0.00 |
| **Kurtosis:** | 9.233 | **Cond. No.** | 25.3 |

**II) Decision Tree**

I also built the decision tree regression model here.

A decision tree is based on hierarchical decision; here hierarchical mean the model is defined by a series of questions that lead to a value when applied to any observation.

A Non-parametric method means that there are no underlying assumptions about the distribution of the errors or the data. It basically means that the model is constructed based on the observed data.

Here the decision tree and Linear Regression give almost same error.

* **Model Evaluation**

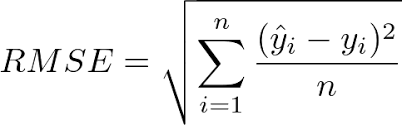
**I) RMSE**

After making any model, it is necessary to evaluate that model, to check that how much accurately your model can predict the future values. So I divide my data into two parts that is train and test.

I make model on train data and test the model on test data.

The metric used by me for model evaluation is RMSE, i.e. root mean square error.

The formula is given as:



As the name suggest it First Square up the difference between the actual value and predicted value, then sum up all square values, then take the mean of all sum square values and lastly take the square rot.

By taking the square root it make the value in the same scale as of the variables.

Error by Linear Regression = 3.1798969799673453

Error by Decision Tree = 3.5201997511345517

* **Deployment**

After modeling and evaluation there is a need to deploy our model in our or client’s premises.

In general there are four ways to deploy the model:-

1) Data science tools (or cloud)

2) Programming language (Java, C, VB, etc.)

3) Database and SQL script (TSQL, PL-SQL, etc.)

4) PMML (Predictive Model Markup Language)

For connecting to database firstly we have to load the libraries like “RMySQL” in R and “pymysql” in python, and then we need user-id, password, and database name, then after connection we can extract data from the database and write queries.

Now it’s times to save the output and lastly close the connection.