

**Cab Fare Prediction**

*Vimalkumar K*

*10 October 2019*

**ABSTRACT**

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.

We are using both R and Python to build the suitable model according to the company’s problem statement. We will try different ML Algorithm and will choose the best model accordingly to help them to know the answers for the questions mentioned above.

# Contents

##### Introduction

* 1. Problem Statement
  2. Data

##### Methodology

* 1. Pre-Processing
     1. Missing Value Analysis
     2. Outlier Analysis
     3. Feature Engineering
     4. Feature Selection
     5. Variance Inflation Factor(VIF)
  2. Modeling
     1. Model Selection
     2. Multiple Linear Regression
     3. Regression Trees
     4. Decision Tree
     5. Random Forest

1. **Conclusion**
   1. Model Evaluation

3.1.1 Root Mean Squared Error (RMSE)

* 1. Model Selection

### visualizations

Visualization on the count of passengers

* 1. Visualization on distribution of trip\_distance

1. **INTRODUCTION**
   1. **Problem statement**

There is a cab rental start-up company which wants to launch a cab service. They have successfully run the pilot project and now want to launch your cab service across the country. They have collected the historical data from their pilot project and now have a requirement to apply analytics for fare prediction. The aim of the project is to design a system that predicts the fare amount for a cab ride in the city.

## Data

Our task is to build models that will predict the fare of the cab depending on certain conditions like location, time and weekdays, etc. Given below is a sample of the data set that we will use to predict the fare amount of cabs :

Table 1.1: Sample Data (Columns: 1-5)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| fare\_amount | pickup\_datetime | pickup\_longitude | pickup\_latitude | dropoff\_longitude |
| 4.5 | 2009-06-15 17:26:21  UTC | -73.8443 | 40.72132 | -73.8416 |
| 16.9 | 2010-01-05 16:52:16  UTC | -74.016 | 40.7113 | -73.9793 |
| 5.7 | 2011-08-18 00:35:00  UTC | -73.9827 | 40.76127 | -73.9912 |
| 7.7 | 2012-04-21 04:30:42  UTC | -73.9871 | 40.73314 | -73.9916 |
| 5.3 | 2010-03-09 07:51:00  UTC | -73.9681 | 40.76801 | -73.9567 |

Table 1.2: Sample Data (Columns: 6-7)

|  |  |
| --- | --- |
| dropoff\_latitude | passenger\_count |
| 40.71228 | 1 |
| 40.782 | 1 |
| 40.75056 | 2 |
| 40.75809 | 1 |
| 40.78376 | 1 |

Attributes present in the dataset are fare\_amount, pickup\_datetime, pickup\_longitude, pickup\_latitude, dropoff\_longitude, dropoff\_latitude and passenger\_count.

The details of dataset attributes are as follows -

* pickup\_datetime - timestamp value indicating when the cab ride started.
* pickup\_longitude - float for longitude coordinate of where the cab ride started.
* pickup\_latitude - float for latitude coordinate of where the cab ride started.
* dropoff\_longitude - float for longitude coordinate 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 Now let’s have a look at the data type of dataset attributes.

fare\_amount object pickup\_datetime object pickup\_longitude float64 pickup\_latitude float64 dropoff\_longitude float64 dropoff\_latitude float64 passenger\_count float64 dtype: object

Here, the datatype of fare\_amount attribute is an object which is not correct. So we converted this attribute into numeric. But while converting it to numeric we found a problem that it contains a string value “-430” at location 1123. So, we basically replaced this value with 430 and then converted it to a numeric datatype. Also, passenger count variable has datatype float so once again we will convert it to object or factor datatype.

# Methodology

We know that data is backbone of data science is Data. We collect data from different sources and converting data I proper format is very necessary. When any new project comes in we spend 80% time in understanding, cleaning and preparing the data as driving the data according to problem is very important. The whole data process is divided into six phases.

1. Business understanding: When any client comes in we should try to understand their problem statement first. It helps us to get proper data for better results.
2. Data understanding: In this we use many statistical techniques, Graphs and visualizations to understand the data so that we can understand the data well and can get relevant data from the client.
3. Data Preparation: This means exploring the raw data we receive from client and understanding what data speaks out. In data science 80% of our time goes in data understanding, cleaning and preparation and 20% in model development and model evaluation. If the quality of data is good the model will predict better and results in high accuracy.
4. Data modeling: There are many machine learning algorithms and we have to select the most appropriate algorithm according to our problem statement.

Evaluation: It helps us to evaluate our model. It tells us whether our model is able to accomplish the business objective or not.

1. Deployment: This is the final phase in which we deploy our model in client premises.

## Pre-Processing

Before developing any model, we first need to look into the data. By, saying look into the data, I mean to explore the data. Look at the datatypes of attributes, find the minimum and maximum values of variables compare it with its mean value. Convert the required datatypes. Finding and imputing missing values using various methods like mean, median, etc. This is nothing but data preprocessing where we analyze the data, clean the data and transform the data. Data preprocessing is the probably most or one of the most important things in model development. So, it needs to be taken care of.

##### Missing Value Analysis

Missing value analysis is a method or technique to find out if there are missing values in the attributes of the dataset. When we applied missing value analysis on our dataset, we found that passenger\_count had most numbers of missing values followed by fare\_amount. Except for these two variables, there were no missing values in any other variables. Missing values can be found by using these syntaxes

* + - * is.null( ).sum() in python
      * function(x){sum(is.na(x))} in R

We got the following result after applying missing value analysis on our dataset

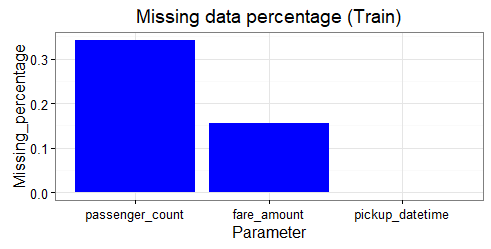
|  |  |  |
| --- | --- | --- |
|  | **Variables** | **Missing\_percentage** |
| **0** | passenger\_count | 0.342317 |
| **1** | fare\_amount | 0.149374 |
| **2** | pickup\_datetime | 0.000000 |
| **3** | pickup\_longitude | 0.000000 |
| **4** | pickup\_latitude | 0.000000 |
| **5** | dropoff\_longitude | 0.000000 |
| **6** | dropoff\_latitude | 0.000000 |

Now, you can see that passenger\_count has 0.342317% missing values which is obviously much small. Similarly, fare\_amount has got 0.149374% missing values otherwise there were no missing values in the dataset.

As we know that passenger\_count is a categorical variable we used mode method to impute missing values of this variable.

But, as missing values present in fare\_amount which is target variable we remove the index or rows of missing value instead of imputing of it.

Let’s look at the visualization of missing values.



* + 1. **Outlier Analysis**

In statistics, an outlier is defined as a data point that differs significantly from other observations. Outlier analysis is a technique to find these points. Outlier analysis can only be done on a numerical variable.

Causes of Outliers

* Poor data quality/contamination
* Low-quality measurements, malfunctioning equipment, manual error
* Correct but exceptional data

In our case, we first analyzed location variables i.e latitude and longitude. As we know the fare of the cab may change from location to location hence, we considered all the locations of train dataset that were outside the locations of test dataset as outlier locations. And then we removed these locations from the train dataset, for example, look at the following R code.

#longitude boundary

min(test$pickup\_longitude, test$dropoff\_longitude) max(test$pickup\_longitude, test$dropoff\_longitude)

# longitude boundary=(-74.26324,-72.98653)

#latitude boundary

min(test$pickup\_latitude, test$dropoff\_latitude) max(test$pickup\_latitude, test$dropoff\_latitude) #latitude boundary=(40.56897,41.70956)

#set boundaries min\_longitude=-74.26324 min\_latitude=40.56897 max\_longitude=-72.98653 max\_latitude=41.70956

train=subset(train, pickup\_longitude >= min\_longitude) train=subset(train,pickup\_longitude <= max\_longitude) train=subset(train,pickup\_latitude >= min\_latitude) train=subset(train,pickup\_latitude<=max\_latitude)

train=subset(train, dropoff\_longitude >= min\_longitude) train=subset(train,dropoff\_longitude <= max\_longitude) train=subset(train,dropoff\_latitude >= min\_latitude) train=subset(train,dropoff\_latitude<=max\_latitude)

So, using these lines of code we removed outlier locations from the train dataset.

Now, we analyzed fare\_amount and saw significantly larger values than mean. For better understanding let’s look at the summary of fare\_amount.

|  |  |
| --- | --- |
| count | 16043.000000 |
| mean | 15.040871 |
| std | 430.459997 |
| min | -3.000000 |
| 25% | 6.000000 |
| 50% | 8.500000 |
| 75% | 12.500000 |
| max | 54343.000000 |

Name: fare\_amount, dtype: float64

So, we can see from the above summary that the maximum value of fare\_amount i.e 54343 is way larger than the mean value which is 15.044. Similarly, if you look at the minimum value in fare\_amount is negative which is not possible. So, we considered all the value below 1 and above 150 in the fare\_amoubt variable as an outlier and removed it from the dataset.

Similarly, if you will look at the summary of passenger\_count you will find that the maximum value in passenger count is huge. Let’s have a look at the summary of the passenger\_count.

|  |  |
| --- | --- |
| count | 16012.000000 |
| mean | 2.625070 |
| std | 60.844122 |
| min | 0.000000 |
| 25% | 1.000000 |
| 50% | 1.000000 |
| 75% | 2.000000 |
| max | 5345.000000 |

Name: passenger\_count, dtype: float64

As you can see maximum passenger\_count is 5345 which is not possible as probably no cab in this world can take these many passengers at once.

Similarly, if you look at the minimum value of passenger\_count you will find that the minimum value is 0 which is of no use. So after analyzing passenger\_count we decided to drop all the observations having count below 1 and above 6. Now when we look at the unique values in passenger\_count we found an odd value.

|  |  |
| --- | --- |
| 1.0 | 11051 |
| 2.0 | 2281 |
| 5.0 | 1023 |
| 3.0 | 662 |
| 4.0 | 320 |
| 6.0 | 295 |
| 1.3 | 1 |

Name: passenger\_count, dtype: int64

As you can see there is one unique value as 1.3 which is not possible as there cannot be 1.3 passengers.

Now, we analyzed the summary of the trip\_distance which we extracted from longitude and latitude data and will discuss more it later in the Feature Engineering part. Now we found minimum trip\_distance value as zero. So, we decided to drop all the observations having trip\_distance value below 0.2, as most of the people do not prefer a cab for distance below 200 meters.

### Feature Engineering

Feature Engineering is used to extract important or valuable features from the data. In our case, we had a timestamp attribute pickup\_datetime which will be of no use if we don’t extract important features from it. We extracted many important features like day, year, month, weekday\_names, hour from this timestamp variable. To do so we used “pd.to\_datetime” in python and “as.Date” in R.

Similarly, we calculated trip\_distance from pickup and dropoff latitudes and longitudes. To calculate this distance we used Haversine distance formula.

For example, look at the below python code

def trip\_distance(lon1, lat1, lon2, lat2):

lon1, lat1, lon2, lat2 = map(np.radians, [lon1, lat1, lon2, lat2]) dlon = lon2 - lon1

dlat = lat2 – lat1

a = np.sin(dlat/2.0)\*\*2 + np.cos(lat1) \* np.cos(lat2) \* np.sin(dlon/2.0)\*\*2 c = 2 \* np.arcsin(np.sqrt(a))

km = 6371 \* c return km

Here, we have used np.radians to convert latitude and longitudes into radian. And from 2nd last line to 4th last line is the Haversine formula. Where 6371 is nothing but the radius of the earth in kilometers.

## Feature Selection

Feature selection is the process of selecting a subset of relevant features (variables, predictors) for use in model construction. In machine learning and statistics feature selection is also known as variable selection, attribute selection or variable subset selection. In our dataset pickup\_datetime which is a timestamp variable is of no use hence dropped this column from our train and test dataset. And selected the rest of the features.

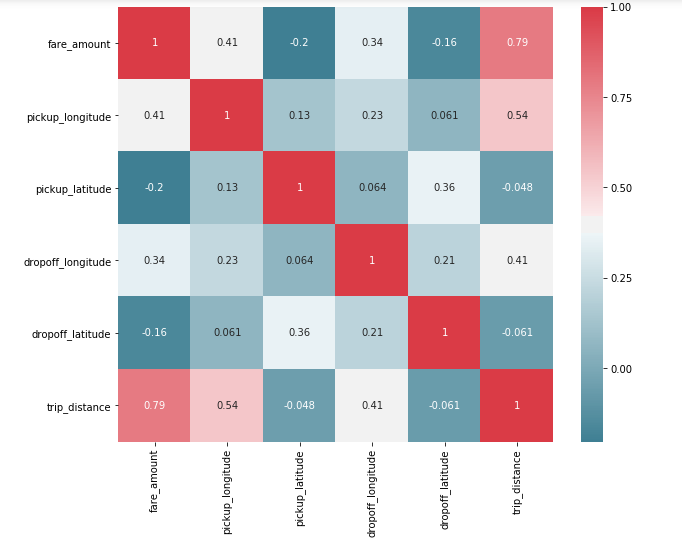
**Correlation analysis:** It is one of the methods for feature selection technique applied only on numerical data. Correlation tells us the association between two continuous variables. It ranges for -1 to +1.

**-1: Highly negatively correlated**

**0: No correlation**

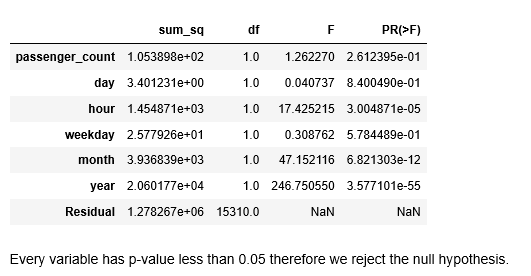
**+1: Highly positively correlated**

In correlation there is an assumption that there should be high dependency between predictor and target variable but there should be low dependency between two predictors.

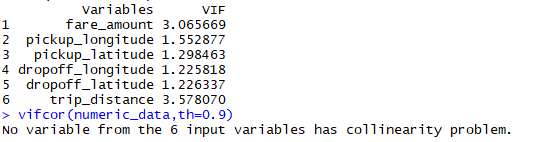


**ANOVA (Analysis of Variance):** It is applied on one categorical and one continuous variable. It is a statistical technique used to compare means of two or more group. We get results in form of p values. If p value is less than 0.05 we will reject NULL HYPOTHESIS (result purely from chance) and accepts ALTERNATE HYPOTHESIS (influenced by some non-random cause).

In our project we have used Correlation plot and ANOVA to select important variables and selected the variables according to the values of correlation and ANOVA. The correlation plot and ANOVA values are given below.



**2.1.5 Variance Inflation Factor(VIF) :** We have performed VIF test using function VIF which is used to check whether variables have multicollinearity .



As there is collinearity problem no need to take any action .

* + 1. **Feature Scaling**

It comes into an action when we are dealing with parameters of different units and scales. It is also known as variable scaling. It is used to limit the range of range of variables so that they can be compared on common basis. Feature scaling is performed only on continuous data. There are two methods to scale the data **Normalization and Standardization.** Normalization is the process of reducing unwanted variation either within or between variables. Normalization brings all the variables into proportion with one another. It ranges between **0 and 1** and are sensitive to outliers. Normalization works on all kind of continuous data whereas standardization works well when data is uniformly distributed.

In our project without scaling we are getting better accuracy so no need to perform any action.

## Modeling

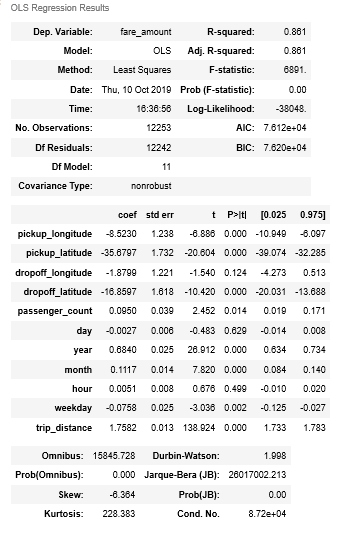
### Model Selection

As we know that we need to predict the fare of the cab we can understand the problem category that this is a forecasting problem. So, we knew that we have to use regression models to predict the target variable. Hence we used the following regression models to predict the result.

* Linear Regression
* Decision Tree
* Random Forest

### Multiple Linear Regression

### model1 = sm.OLS(y\_train, X\_train).fit( ) ### model1.summary( )



As you can see from the adjusted R-squared value, we can explain nearly 86.5 % of our data using our linear model. Which is quite good. Now if you look at the p- value then we can say that the null hypothesis for passenger\_count, day and hour is true.

###

X\_train1,X\_test1,y\_train1,y\_test1=modeling(train,'fare\_amount',drop\_cols=['pickup\_datet ime'],is\_train=True,split=0.3)

### model2 = sm.OLS(y\_train1, X\_train1).fit( ) ### model2.summary( )

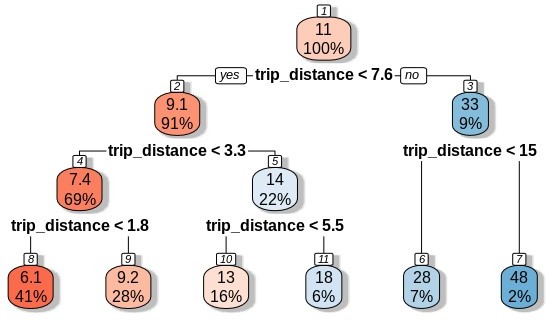
OLS Regression Results

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Dep. Variable:** | fare\_amount | | **R-squared:** | | | 0.862 | |
| **Model:** | OLS | | **Adj. R-squared:** | | | 0.862 | |
| **Method:** | Least Squares | | **F-statistic:** | | | 6077. | |
| **Date:** | Tue, 27 Aug 2019 | | **Prob (F-statistic):** | | | 0.00 | |
| **Time:** | 22:15:39 | | **Log-Likelihood:** | | | -33384. | |
| **No. Observations:** | 10738 | | **AIC:** | | | 6.679e+04 | |
| **Df Residuals:** | 10727 | | **BIC:** | | | 6.687e+04 | |
| **Df Model:** |  | 11 |  |  |  |  | |
| **Covariance Type:** | **coef** | nonrobust  **std err** | **t** | **P>|t|** | **[0.025** | **0.975]** | |
| **pickup\_longitude** | -13.3092 | 1.315 | -10.125 | 0.000 | -15.886 | -10.732 | |
| **pickup\_latitude** | -49.1592 | 1.832 | -26.833 | 0.000 | -52.750 | -45.568 | |
| **dropoff\_longitude** | 2.1519 | 1.315 | 1.636 | 0.102 | -0.426 | 4.730 | |
| **dropoff\_latitude** | -3.5173 | 1.739 | -2.022 | 0.043 | -6.927 | -0.108 | |
| **passenger\_count** | 0.0438 | 0.041 | 1.063 | 0.288 | -0.037 | 0.125 | |
| **day** | -0.0020 | 0.006 | -0.336 | 0.737 | -0.014 | 0.010 | |
| **year** | 0.6591 | 0.027 | 24.188 | 0.000 | 0.606 | 0.713 | |
| **month** | 0.1033 | 0.015 | 6.765 | 0.000 | 0.073 | 0.133 | |
| **hour** | 0.0042 | 0.008 | 0.516 | 0.606 | -0.012 | 0.020 | |
| **weekday** | -0.0970 | 0.027 | -3.626 | 0.000 | -0.149 | -0.045 | |
| **trip\_distance** | 1.7927 | 0.014 | 132.620 | 0.000 | 1.766 | 1.819 | |
| **Omnibus:** | 13667.474 | **Durbin-Watson:** | | 1.992 | | |  |
| **Prob(Omnibus):** | 0.000 | **Jarque-Bera (JB):** | | 23448967.651 | | |  |
| **Skew:** | -6.136 | **Prob(JB):** | | 0.00 | | |  |
| **Kurtosis:** | 231.602 | **Cond. No.** | | 8.67e+04 | | |  |

Even after splitting the dataset into a 70% train and 30% test we didn’t see much improvement.

### Regression Trees

Now, we will use other regression models to predict the fare of a cab ride. Below is a visualization of the decision tree used in our model.



### Decision Tree

We have divided train data into 80% train and 20% test datasets for the decision tree model. Let’s look at the decision tree model development code in python.

### fit\_DT=DecisionTreeRegressor(max\_depth=6,random\_state=42).fit(X\_train, y\_train)

### predictions\_DT = fit\_DT.predict(X\_test)

Here, X\_train is subset data from the train dataset for training and has all independent variables. Similarly, y\_train is a training dataset with only the target variable.

X\_test is test data that is a subset of the train dataset and has all the independent variables.

### Random Forest

For Random Forest also we have divided train data into 80% train and 20% test datasets. Let’s look at the random forest model development code in python.

###fit\_RF=RandomForestRegressor(n\_estimators=50,random\_state=42).fit(X\_train,y\_ train)

### prediction\_RF=fit\_RF.predict(X\_test)

Here, X\_train is a subset data from the train dataset for training and has all independent variables. Similarly, y\_train is a training dataset with only the target variable.

X\_test is test data that is a subset of the train dataset and has all the independent variables.

n\_estimators is nothing but no. Of trees to be used in the random forest.

# Conclusion

z

#### Model Evaluation

Now that we have three models for predicting the cab fare, we need to decide which one to choose. There are several criteria that are used for evaluating and comparing models. We can compare the models using any of the following criteria:

1. Predictive Performance
2. Interpretability
3. Computational Efficiency

In our case, we have used predictive performance criteria to select the best model. That means model which gives the best accuracy we will select that model.

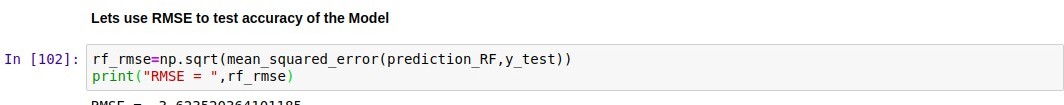
Predictive performance can be measured by comparing Predictions of the models with real values of the target variables and calculating some error metrics.

**3.1.1 Root Mean Squared Error (RMSE)**

RMSE is one of the error measures used to calculate the predictive performance of the model. We will apply this measure to our models that we have developed. This is also called as Root Mean Squared Deviation (RMSD)

– it Squares the errors, find their average and takes the square root

–Time-based measure



In the above code, prediction\_RF is the predicted value and y\_test is the actual value.

It will provide the error percentage of the model.

“Reason, why we selected RMSE for measuring error, is that we wanted to punish the larger errors. For example, if we predict very high fare for a small distance then it will lead to the negative impact of the company on the customers. Hence our focus was to remove such errors.”

RMSE value that we got in python is as follows: -

|  |  |  |
| --- | --- | --- |
| **Model Name** | **Error Rate** | **Accuracy** |
| Linear Regression | 5.10% | 94.90% |
| Decision Tree | 3.61% | 96.39% |
| Random Forest | 3.10% | 96.90% |

RMSE value that we got in R is as follows: -

|  |  |  |
| --- | --- | --- |
| **Model Name** | **Error Rate** | **Accuracy** |
| Linear Regression | 5.98% | 94.02% |
| Decision Tree | 4.64% | 95.46% |
| Random Forest | 3.91% | 96.09% |

#### 3.2 Model Selection

As we can see from the above tables the random forest gave the best accuracy both in R and python. That’s why we selected the Random Forest model for predicting the fare.

After building number of regression models there are criteria by which they can be evaluated and compared. Model evaluation tells us whether our model is able to accomplish the business object or not. There are different metrics for regression model like **MSE (Mean Square Error), RMSE (Root Mean Square Error), MAPE (Mean Absolute Percentage Error), MAE (Mean Absolute Error) etc. MAPE and MAE are used for regression data** whereas MSE and RMSE are used for **transition or time series data also called time series analysis**.

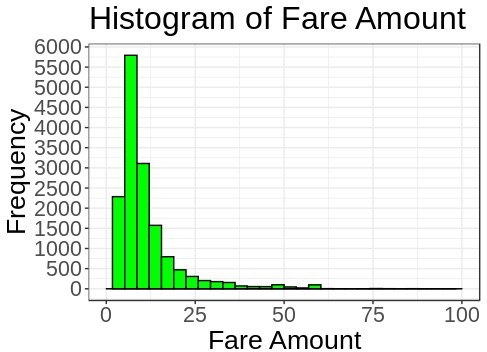
If we want to convert error number in particular percentage, we should go for MAPE. we have used MAPE as error metric. Accuracy can be calculated as:

**Accuracy = 100 – MAPE**

**4.2 END NOTE**

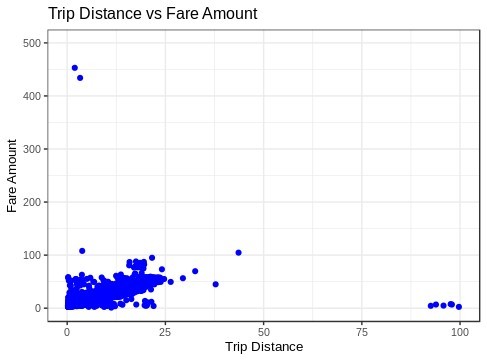
We established significant relationship between several independent variable and Cab fare prediction. We developed a regression model that can be applied to predict cab fare prediction. and also found that the fare\_amount is almost fixed for trips over 80 KM. Also, fare\_amount is few times very high for small distances, also that single passengers booked a cab for most numbers of the time whereas family booking was least. We trained various different models and preformed. we can thus conclude that the developed model can be used to predict the Cab fare prediction.

1. **Visualizations**
   1. **Visualization of the distribution of fare\_amount**



As you can see in the above Histogram that most of the fare\_amounts are somewhere between 5 to 15 dollars.

### Visualization on distribution of fare\_amount over trip\_distance

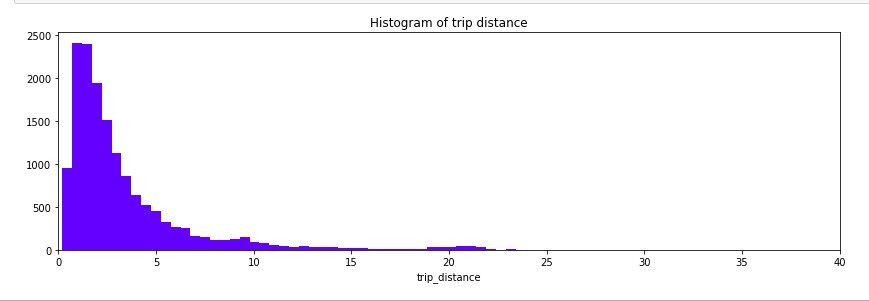


As we can see that in the above scatter plot that fare\_amount is almost fixed for trips over 80 KM. Also, fare\_amount is few times very high for small distances which we considered as an outlier and removed during outlier analysis.

### Visualization on the count of passengers

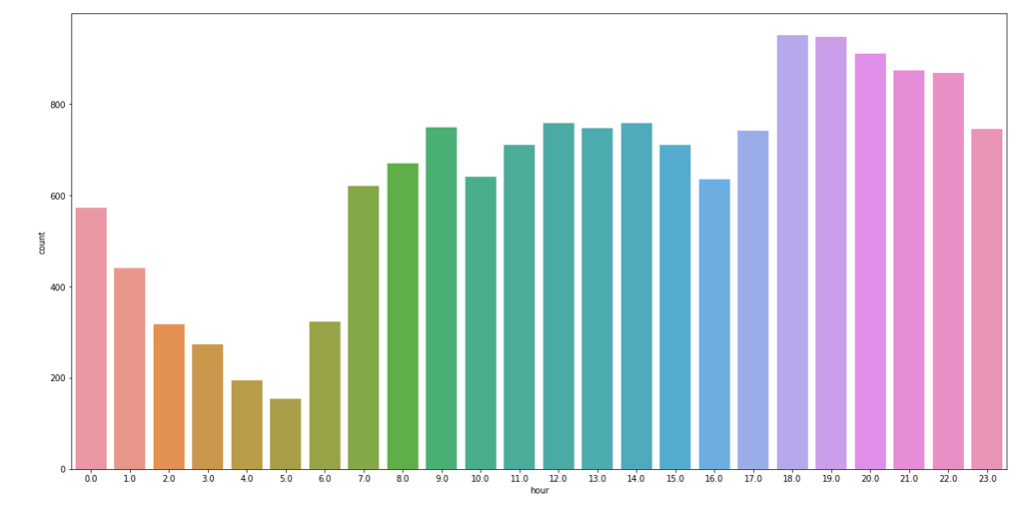
As we can see in the above bar graph that single passengers booked a cab for most numbers of the time whereas family booking was least.

### Visualization on distribution of trip\_distance



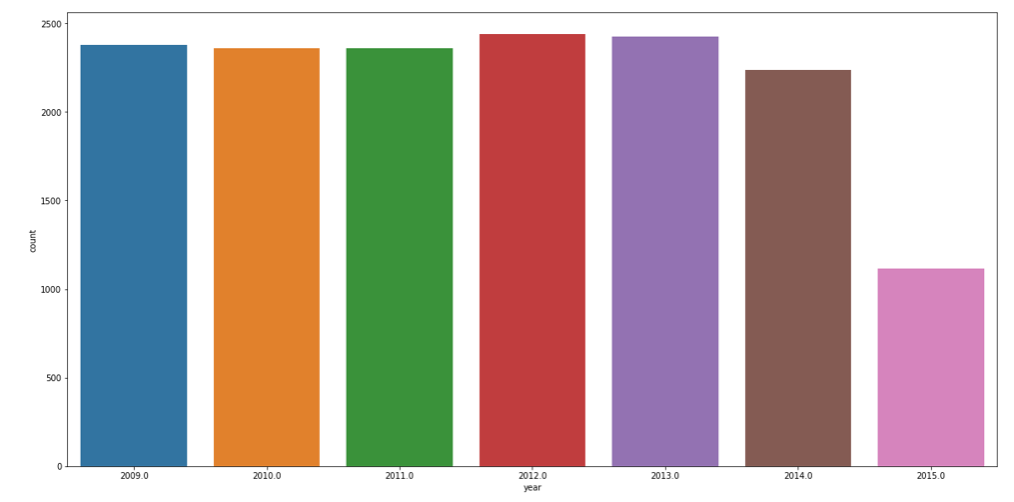
As we can see in the above histogram that most of the trip distance was between

* 1. **Visualization on distribution of trip count vs time**



As we observe the no of trips are high in the evening time i.e; 1700hrs to 2300hrs so company can provide more cabs during that time.

* 1. **Visualization on distribution of trip count vs year**



***APPENDIX – B***

***VARIANCE INFLATION FACTOR***

VIF (Variance Inflation Factor) is used to detect and remove multicollinearity. It is one of the assumptions of linear regression. VIF is used only on independent variable. It is calculated by the formula,

VIF = 1/1-r2

Where, r2 = % variance in variables & 1-r2 also called tolerance of the model.

If r2 is high it means the given variable is redundant. So we need not to bring the given variable in the model. It means the given variable is highly correlated. If r2 is low it means the given variable is not redundant and we should include that variable in our model. It means the given variable is less correlated.

Higher the VIF more collinear is the variable which means we should not include that variable in our model. Lower the VIF less collinear is the variable which means it can be included in our model.

***Basic Output Terms***

**Residual standard error:** It is also called standard deviation error. It measures the average amount that the coefficient estimates vary from actual average value of our response variable. It helps in calculation of p-value.

**t- value:** It measures how many standard deviations our coefficients are away from 0. Coefficients should be far away from zero because if coefficient of any variable is near to 0 it means that variable is not able to explain the target variable i.e. that variable is an irrelevant variable. With help of t-value we calculate p-value.

**p-value:** It helps us to decide whether to accept or reject the variable i.e. a variable is contributing much information or not.

**F-statistics:** It is a good indicator of whether there is a relationship between our predictor and the response variable. F-statistics should be greater than 1.

**Degrees of Freedom:** Number of observation (training data) – Total number of variable

**R Square:** It is numerical value which tells us the amount of variance of the dependent variable is explained by all independent variable. It tells us how much our model is robust and what the strength of model on training data is.

**Adjusted R Square:** It is derived from R-Square values. Adjusted R Square will penalize the effect of additional variables which are not carrying much information. It should be always less than R Square.

**AIC (Alkaline Information Criteria):** It adjusts the loc likelihood based on the number of observation and complexity of the model.

**BIC (Baisen Information Criteria):** It is similar to AIC but has high penalty for models.

**Omnibus:** Provides combined statistical test for the presence of skewness and kurtosis. Basically, it is breakdown of skewness and kurtosis.

**Skew and Kurtosis:** These tests are basically for time series dataset.

**Null Deviance:** It tells us how well the response variable is predicted by the model with intercept only.

**Residual Deviance:** It tells us how well the response variable is predicted by using null deviance and all other independent variables.

***REFERENCE***

1. **“*Edwisor Videos”***
2. **Youtube**