CAB FARE PREDICTION USING R

Ayush Kumar Jain

CONTENT

S.NO.	TOPIC	PAGE
1	Problem statement	3
2	Data Pre-Processing	
2.1	Dealing with Insensible Data	3
2.2	Dealing with Missing value and visualization after imputation	6
2.3	Feature Engineering- Creating New features	8
2.4	Feature Selection- Correlation plot, Chi-square test and ANOVA test	11
2.5	Feature Scaling	12
3	Dividing Data into Train and validation data	12
4	Building Model using Different Machine Learning Algorithms	
4.1	Linear Regression	13
4.2	Decision Tree	15
4.3	Random Forest	15
5	Model selection and Prediction	16

1. PROBLEM STATEMENT

The objective of this project is to predict Cab Fare amount. 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.

ABOUT DATA:

TRAIN DATA: Train data is Historical data for which we know the fare_amount. Train data carries total of 16067 observation with attributes pickup_datetime, pickup_latitude, pickup_longitude, dropoff_latitude, dropoff_longitude, passenger_count and fare_amount.

TEST DATA: Test data is the new data for which we have to predict fare_amount value by building model using train data set. Test data carries total 9914 observation with attributes pickup_datetime, pickup_latiutde, pickup_longitude, dropoff_latitude, dropoff_longitude, passenger_count.

About Attributes:

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 value indicating the number of passengers in the cab ride.
fare_amount – float for fare of the trip. This attribute is to be predicted for the test case.

2. DATA PRE-PROCESSING

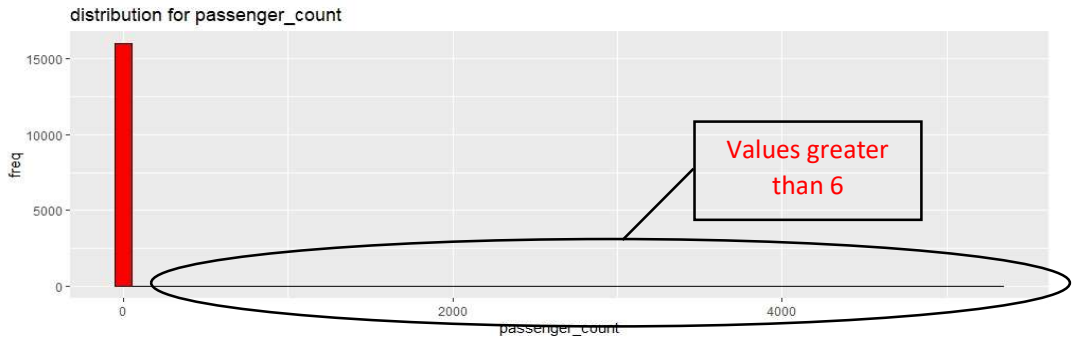
In this step we try to understand the giving data by plotting some visualizations and by using some functions of R. If the data is messy, we will clean it by removing observation or by treating them as missing values, this step is known as Exploratory Data Analysis.

After cleaning the insensible observations, we will go for missing value. After dealing with missing values we will plots some visualization to understand our data more clearly.

2.1 Dealing with Insensible data or observation:

1) Passenger_count:

The passenger_count variable values should be a positive integer. As we can see from our test data that maximum number of passengers is 6. That mean any value which is non integer, negative and more than 6, does not make any sense. So, to understand the passenger_count variable lets observe the its distribution plot as shown below.



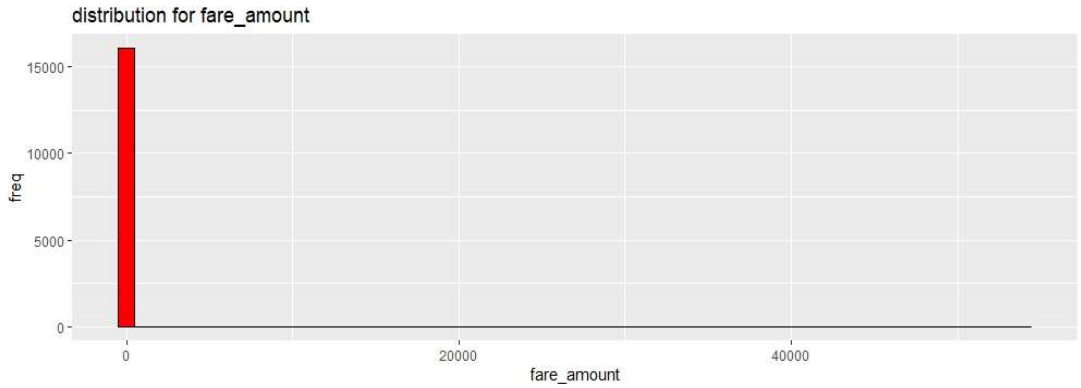
As from above plot we can see that the passenger_count in train data set contains some negative values, non-integer value and also some values greater than 6.

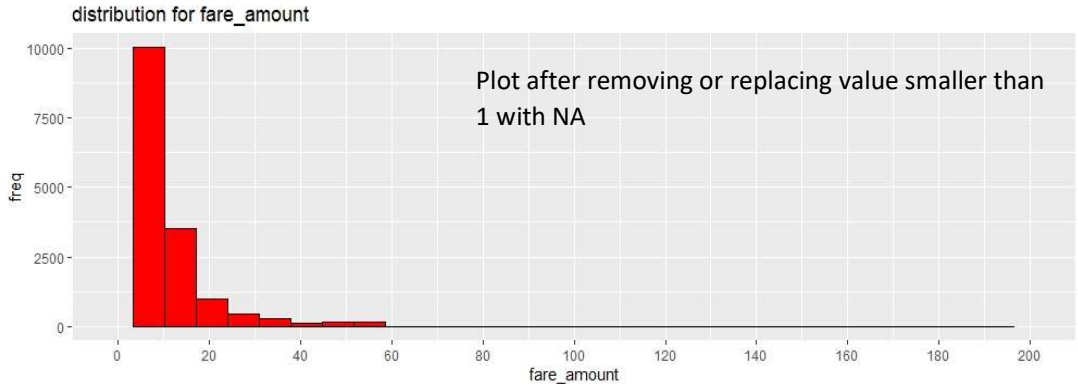
On using some function in R, we found out total 78 observations with negative values and values greater than 6. Now what we can do that we can replace these observations as NA than treat them as missing values or we can remove these observations. We go for both methods one by one than select the method for which our model gives better results.

Also, after doing this step we will Round up the values of passenger_count & convert passenger_count data type to a categorical type data type that is “factor” in R.

2) Fare_amount:

Now, fare_amount is a numeric value that indicates the fare for the trip. This value cannot be a negative value. So, let’s plot some visualization to see how fare_amount is distributed

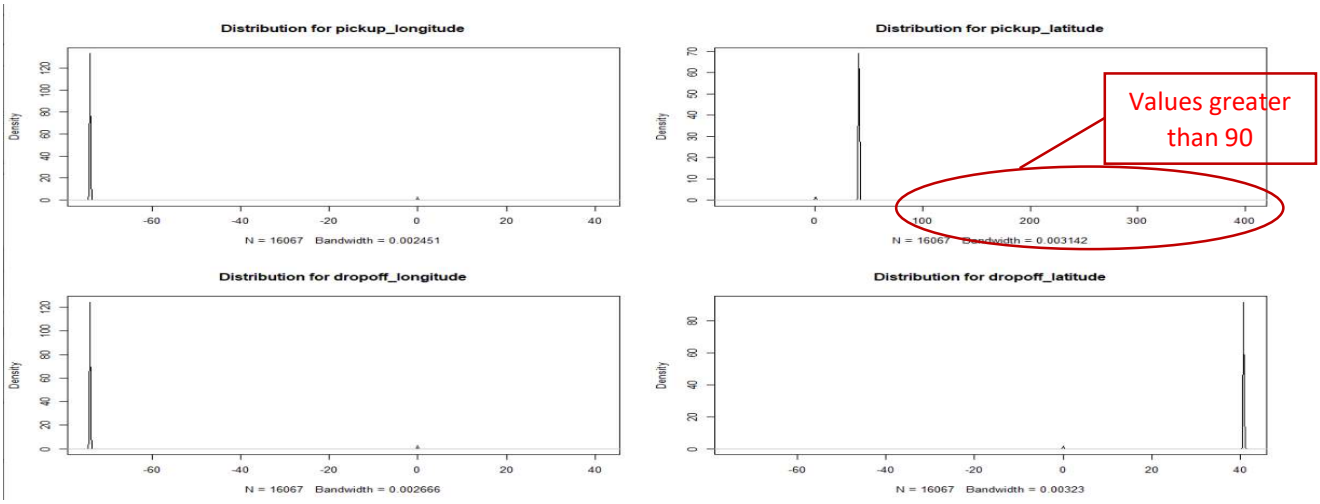




As we can see from above plot fare amount have some negative values, also fare_amount have some abnormally high values. Most of the value for fare_amount is less than 60 and some which are greater than 60 can be possible depending on the other data. On using some functions in R, we find out there are only 4 values with fare_amount greater than 200(taken as maximum value for fare_amount) and 5 value which are less than 1. Now what we can do that we can replace these observations as NA than treat them as missing values or we can remove these observations. We go for both methods one by one than select the method for which our model gives better results.

3) Latitude and Longitude Parameters:

In our data we have pickup_latitude and pickup_longitude which tell us about our pickup location. And also, we have dropoff_latitude and dropoff_longitude which tells us about our drop-off location. Now from our general understanding of latitude and longitude we know that latitude can be between -90 and 90 and longitude can be between -180 and 180. Values out of these values are insensible values, so let's observe these plots given below to check our data have such values or not.



As from above plot we can see that all latitude and longitude are in range except pickup_latitude that have some values greater than 90. On using R, we find out data have only one such value. Also, we can see from above most of latitude values lies near to 40 and longitude values lies near to -72. From latlong.net we find out the given data is for Newyork city. So, by taking range for latitude as (39,42) and longitude range as (-72 to -75) we found out total 337(if we have replaced insensible values for passenger_count and fare_amount as NA, it is 333 when we remove these observations) point lies away from Newyork. So, with these 338 what we can do is we can drop these or treat them as missing value. We do these methods one by one and select the one with better results.

4) Pickup_datetime:

This data contains Date and time at which the trip stars, we can use this attribute to create new attributes lies year, month, day, hour to find out how our fare_amount is varying with time and date.

2.2 Missing value Analysis:

As the name indicate we have analysed the values which are missing in our data.

Let's find out number of missing values in every column by using some function of R.

Number of missing values when we remove all insensible as discussed above:

Variable	Number of missing values	Missing percentage
pickup_latitude	0	0
pickup_longitude	0	0
dropoff_latitude	0	0
dropoff_longitude	0	0
Passenger_count	55	0.3515050
Fare_amount	22	0.14060203
pickupdatetime	0	0.000000

What we can do is we can drop these observations or we can impute them using different missing value imputation methods.

Number of missing values when we impute insensible data as NA.

Variable	Number of missing values	Missing percentage
pickup_latitude	325	2.0227796
pickup_longitude	324	2.0165557
dropoff_latitude	324	2.0165557
dropoff_longitude	322	2.0041078
Passenger_count	133	0.8277837
Fare_amount	34	0.2116139
pickupdatetime	0.00	0.000000

We will impute these missing values using different missing value imputation methods.

Missing value imputation:

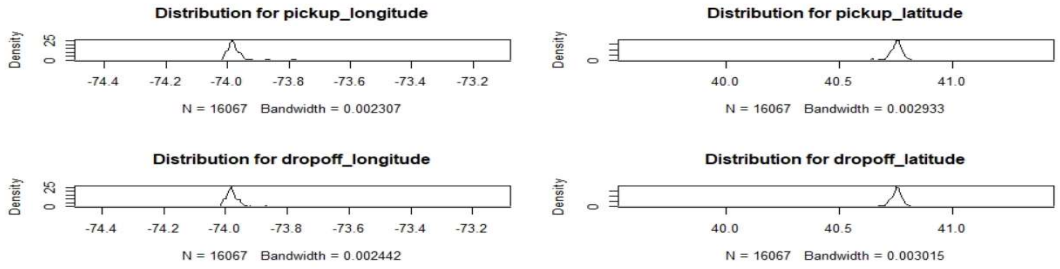
1) passenger_count:
   As we discussed above passenger count can only take integer value between 1 to 6, so it is a categorical type or factor type variable. Hence for imputing missing values for passenger_count have 2 methods: a) we can impute missing value with mode and b) using KNN.
   To find out which method is better we create a missing value in passenger_count at some random observation say 1000, than we impute the value using both methods, than we compare new value with actual value.
   #Actual value for passenger_count at location 1000 is 1
   #Mode for passenger_count is 1
   #When we impute this value using KNN we also get 1 at this observation.

Now, we cannot use the mode method it will bias our data towards passenger_count value 1. Therefore, we use KNN.

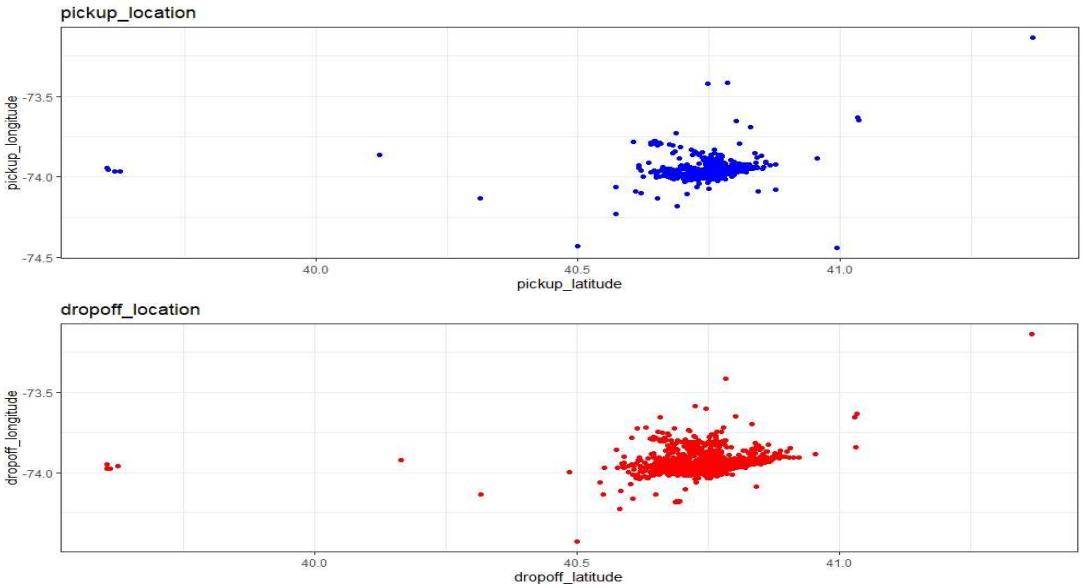
2) fare_amount:
   As we see the fare_amount is numeric type data, we can impute missing value for fare_amount with its mean or with its median or we can impute it using KNN method. To find out which method is better we create a missing value in fare_amount at some random observation say 1000, than we impute the value using both methods, than we compare new value with actual value.
   Actual Value of fare_amount at observation 1000= 5.7,
   Mean of fare_amount= 11.31
   Median of fare_amount = 8.5
   When we impute using KNN=5.43
   As we can see KNN giving better results, hence we use KNN method for imputing missing values for fare_amount.
3) Latitude and longitude variable:
   As we see these data is numeric type data, we can impute missing value with its mean or with its median or we can impute it using KNN method. To find out which method is better we create a missing value at some random observation say 1000, than we impute the value using both methods, than we compare new value with actual value.

	pickup_latitude	pickup_longitude	dropoff_latitude	dropoff_longitude
Actual	40.75843	-73.98846	40.73015	-73.98382
Mean	40.75092	-73.97482	40.75141	-73.97385
Median	40.75332	-73.98205	40.75424	-73.98058
KNN	40.74677	-73.98412	40.76076	-73.97385

   As we can see for pickup_latitude and dropoff_latitude we are getting better results from imputing with median, for dropoff_longitude we are getting better results when we impute using mean and for pickup_longitude we get better results when imputing using KNN.
   So, we use different methods as mentioned above to impute missing values.
   So, after imputing these values lets observe how our latitude and longitude variables are distributed.

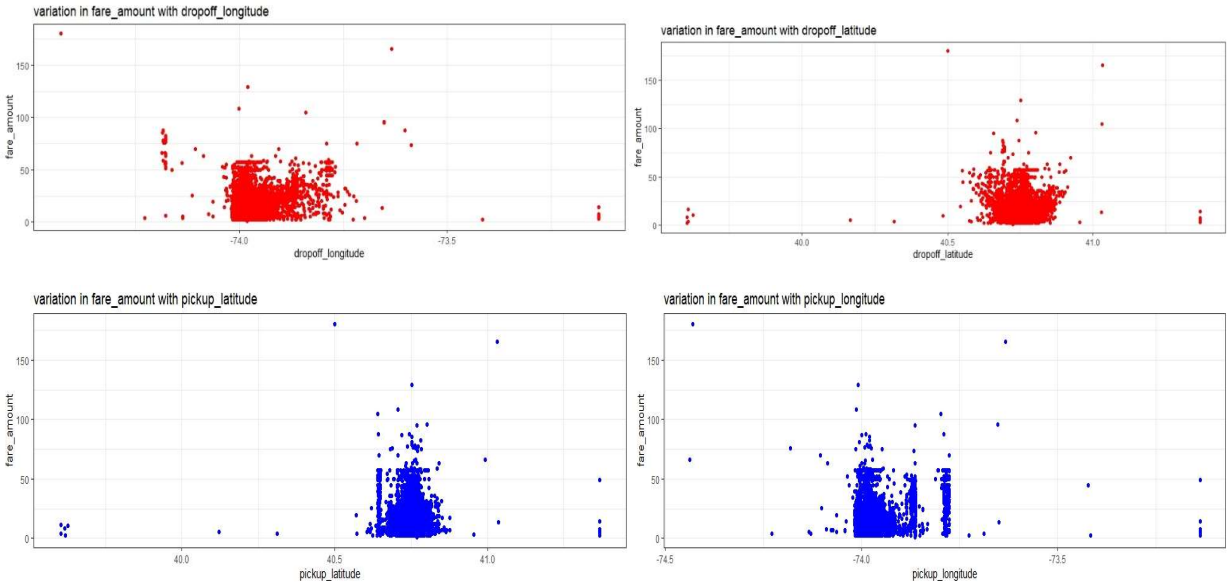


As we can see all values are in range now.
Let's also observe pickup and drop-off locations, by plotting scatter plot.



As we can see most of the values are concentrated between latitude value (40.5,41) and longitude values (-74.3, -73.7).

Let's see how fare_amount varying with latitude and longitude.



As we can see for dropoff_longitude near to -74.2 the fare_amount is higher, also some observation for pickup_longitude near to -74 have higher fare_amount. So, from above we can say that fare_amount is varying with locations.

2.3 Feature Engineering:

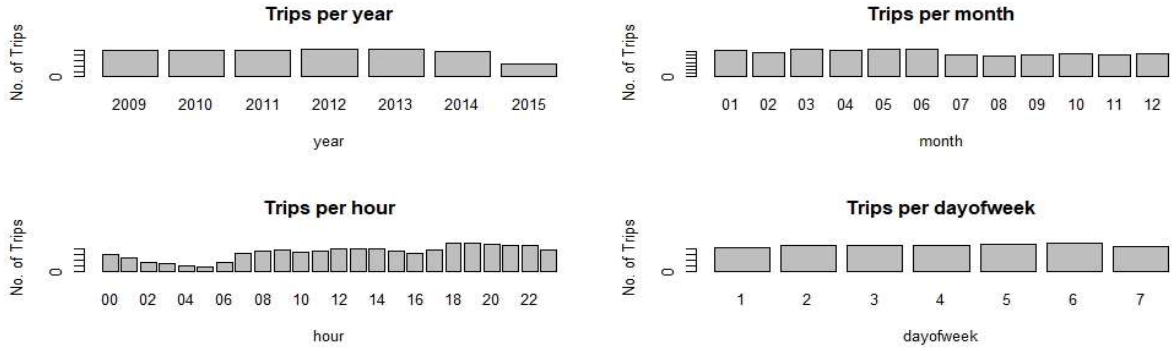
In this step we create new feature with the help of existing feature

1) Year, month, dayofweek and hour:

As we see in our day to day life, a taxi fare can vary because of climate/season, can be different on weekdays and weekends, also be different on same day at peek traffic hours or normal traffic hours. So, in this case to find more about these type of variation

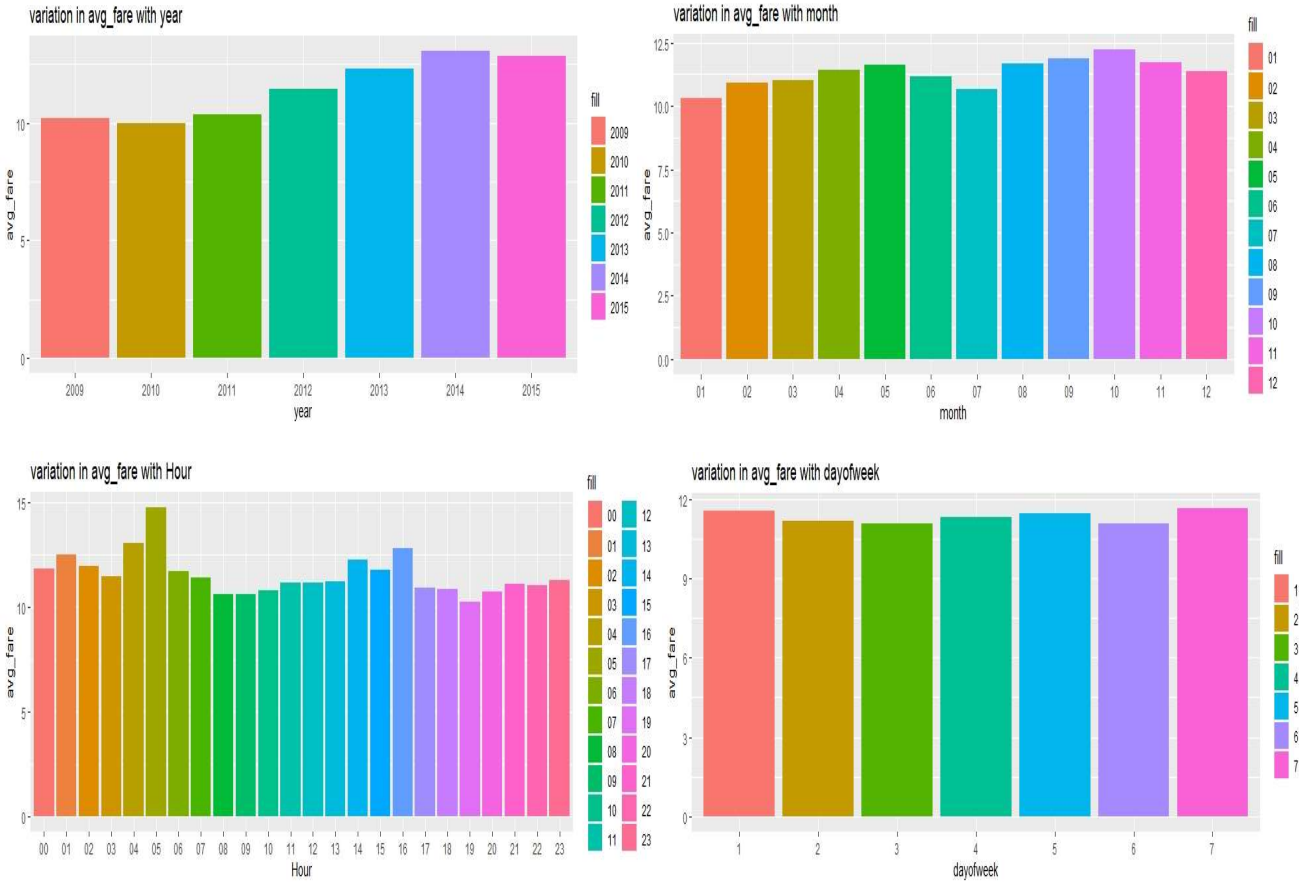
in fare_amount let's create new features as Trip year as year, Trip month as month, Trip day of week as dayofweek and pickup time as hour using pickup_datetime. Now on checking for missing value in these features we found out they have each one missing value for same observation, we will drop this value.

After creating let's see how number of trips varying with these features:



As we can see from above that we have less data for 2015 year, also we can see that from month of January (01) to June (06) there are more trips as compare to other month. Also, from 12:00 AM to 06:00 AM number of trips are very less and from 6:00 PM to 9:00 PM trips are more. Similarly, we can also observe that number of trips on are almost same on every day except for Sunday (7).

Let's see how average fare_amount varying with these features



As we can see from above, we can see that from year 2009 to 2015 the average fare_amount is increasing. From month of august (08) to December (12) Average fare_amount is higher. At late night hours i.e. from 12:00 Am to 05:00 AM average fare_Amount is higher but there is no such effect on average fare_amount with week day.

2) trip_distance:

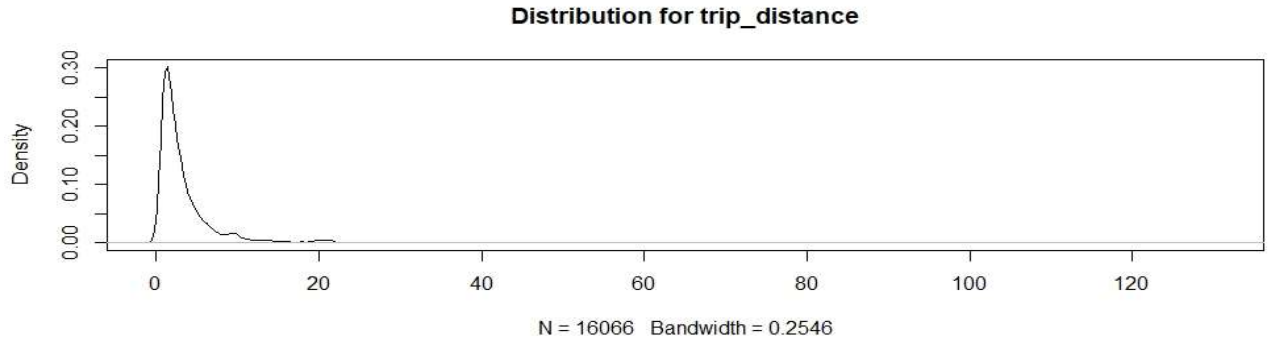
As we know our pickup latitudes and longitude & drop-off latitude and longitude, we can find the distance between these two points by using Haversine method which is describe below (this is the distance between two point on the surface of sphere):

The word "Haversine" comes from the function: haversine(θ) = $\sin^2(\theta/2)$

The haversine formula is a very accurate way of computing distances between two points on the surface of a sphere using the latitude and longitude of the two points. The following equation where ϕ is latitude, λ is longitude, R is earth's radius (mean radius = 6,378km) is how we translate the above formula to include latitude and longitude coordinates. Note that angles need to be in radians to pass to trig functions:

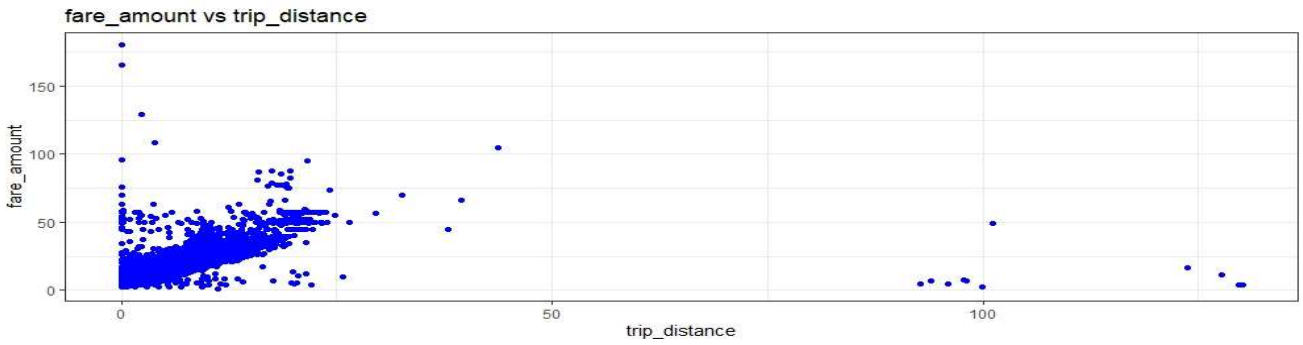
$a = \sin^2(\phi B - \phi A/2) + \cos \phi A * \cos \phi B * \sin^2(\lambda B - \lambda A/2)$
$c = 2 * \text{atan2}(\sqrt{a}, \sqrt{(1-a)})$
$d = R \cdot c$

where A, B are two points and all latitude and longitudes points are in radian. So, after creating this new feature let's take a look at its distribution:



As we can see the distribution is left skewed, also for some observation trip distance is 0. This may be because of a round trip. It is also possible that it is because of cab booking cancelation and fare_amount value at that time is cancelation penalty. Also, on observing trip_distance in test data, we found out it also have 155 observations for which trip_distance is 0. So, we have to considered these observations in train data.

Let's see how fare_amount is varying with trip_distance

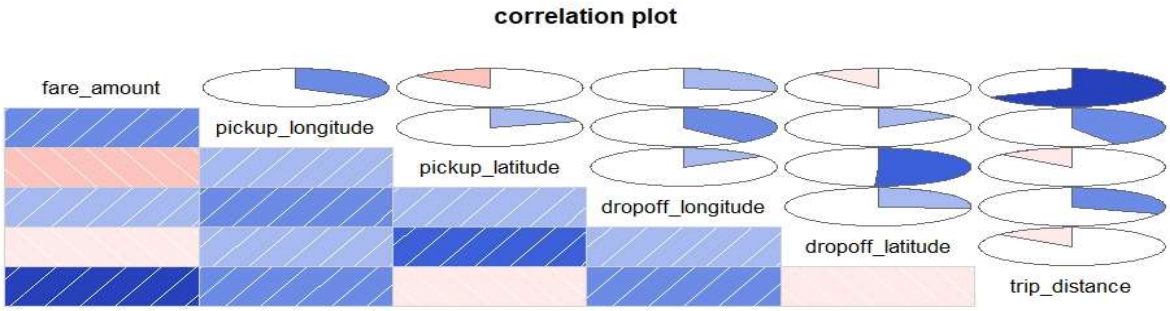


So, as we can see the plot between fare_amount and trip_distance almost following linear pattern that when trip_distance is increasing fare_amount is also increasing.

2.4 Feature selection:

In this step we select the valid feature that is going to drive our model. So, for doing so we first plot correlation plot for Numeric variables, then we do chi-square test between out categorical variable, then ANOVA test to find out effect of categorical variable on our numeric target variable.

Correlation plot between numeric variables:



The Dark Blue colour in above plot is indication of positively correlated and dark red colour is indication of highly negatively correlated. Now, as we can see above there is no pair of independent variables highly correlated to each other. Hence there is no need to drop any variable.

Chi-Square test between categorical variables:

Chi-square test compares 2 categorical variables in a contingency table to see if they are related or not. Assumption for chi-square test: Dependency between Independent variable and dependent variable should be high and there should be no dependency among independent variables.
Null Hypothesis: two variables is independent.
Alternative hypothesis: two variables are not independent.
If p-value from chi-square test comes lower than 0.05 we accept the null hypothesis, otherwise we reject the null hypothesis by saying they are not independent.
From chi-square test we found out (passenger_count and month), (year and hour), (year and dayofweek), (month and hour) and (month and dayofweek) have p-value greater than

0.05. we will only drop those variables which have very low impact on our target variable. So, to find out let's do ANOVA test.

ANOVA Test for independent categorical variable and numeric target variable:

This test helps us to find out the mean of target variable is different for different values of categorical variable or not.

Hypothesis:

Null hypothesis: mean for target variable is same for all values of categorical variable.

Alternative hypothesis: mean for target variable is different for different values of categorical variable.

If p-value from ANOVA test is higher than 0.05 we accept null hypothesis, otherwise reject it. From analysing the results from ANOVA test, shown on next page, we found out dayofweek does not affect our fare_amount mean **so we will drop this feature (starts in the end indicate importance of variable).**

Results from ANOVA test:

	Df	Sum Sq.	Mean Sq.	F value	PR(>F)	
passenger_count	5	2181	436	4.800	0.000219	***
Hour	23	8831	384	4.226	4.68e-11	***
dayofweek	6	612	102	1.122	0.346203	
month	11	4277	389	4.279	2.14e-06	***
year	6	23468	3911	43.048	< 2e-16	***
Residuals	16014	1455015	91			

2.5 Feature Scaling:

In this step we convert the value of independent numeric variables in such a way that they are comparable. For doing so we have two methods: Normalization and standardization.

We can standardization only for those variables for which the distribution is normal distribution. In our case all 5 variables doesn't have normal distribution, so we do feature scaling for all 5 variables i.e. pickup_latitude, pickup_longitude, dropoff_latitude, dropoff_longitude and trip distance using normalization method.

$$\text{New scaled/Normalized value} = \frac{(\text{original value} - \text{minimum value})}{(\text{maximum value} - \text{minimum value})}$$

So, after doing all these pre-processing steps we ended here with total 16066 observations and 10 variables (when we replace insensible data as NA and impute them) or with 15569 observations and 10 variables (when we drop every insensible values and missing values) in our train data. Now let's move towards model building process.

3. DIVIDING DATA:

In this step we divide our data into two parts with 80% data as training set and 20 % data as validation set.

From training set we train our model for predicting the value of target variable and from validation data set we check how accurate our model is.

We divided our train data as x_train (training set) and x_test (validation set).

4. MODEL DEVELOPMENT:

In this step we develop some models using liner regression, decision tree and random forest algorithms. Then we compare our different models using some Error matrix, than we select the best fitted model and use it to predict our fare_amount values for test data.

4.1 Liner Regression:

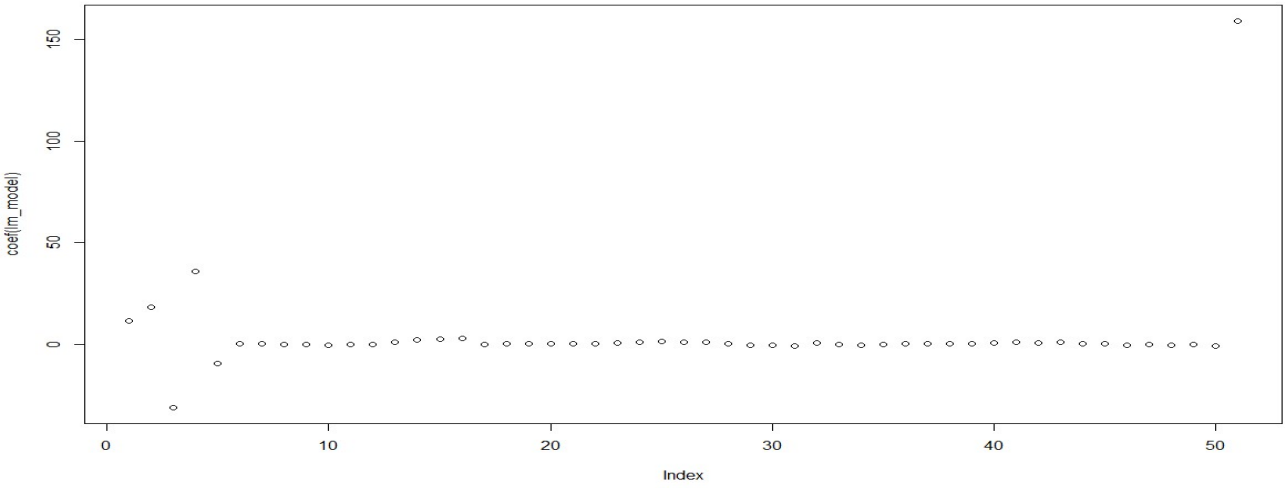
Multicollinearity check– In regression, "multicollinearity" refers to predictors that are correlated with other predictors. Multicollinearity occurs when your model includes multiple factors that are correlated not just to your response variable, but also to each other. For checking we find out VIF (Variance inflation factor) that is measure of multicollinearity in set of multiple regression variables. VIF is always greater or equal to 1.
if VIF is 1, Not correlated to any of the variables.
if VIF is between 1-5, Moderately correlated.
if VIF is above 5, Highly correlated.
If there are multiple variables with VIF greater than 5, only remove the variable with the highest VIF.
We have checked for multicollinearity in our Dataset and all VIF values are below 5.

Model summary:

```
lm(formula = fare_amount ~ ., data = x_train)
Residuals: Min       1Q      Median       3Q      Max
        -181.065   -2.659   -1.122      1.061   186.026
Coefficients:
                   Estimate   Std. Error  t value    PR(>|t|)
(Intercept)        11.79095    2.28987    5.149    2.65e-07 ***
pickup_longitude   18.50774    2.37055    7.807    6.29e-15 ***
pickup_latitude   -31.01028    3.45252   -8.982     < 2e-16 ***
dropoff_longitude  36.05062    2.39842   15.031     < 2e-16 ***
dropoff_latitude   -9.32317    3.40391   -2.739     0.00617 **
passenger_count2    0.56721    0.18301    3.099     0.00194 **
passenger_count3    0.36688    0.31296    1.172     0.24111
passenger_count4    0.24168    0.44465    0.544     0.58677
passenger_count5    0.07211    0.26153    0.276     0.78276
passenger_count6   -0.38494    0.47092   -0.817     0.41370
year2010            0.15888    0.22730    0.699     0.48456
year2011            0.15970    0.22658    0.705     0.48094
year2012            1.35376    0.22623    5.984     2.24e-09 ***
year2013            2.46491    0.22722   10.848     < 2e-16 ***
year2014            2.85830    0.23192   12.325     < 2e-16 ***
year2015            3.09816    0.29318   10.567     < 2e-16 ***
month02             0.20081    0.30115    0.667     0.50492
month03             0.59926    0.29233    2.050     0.04039 *
month04             0.51940    0.29670    1.751     0.08004 .
month05             0.62227    0.29225    2.129     0.03325 *
month06             0.61772    0.29155    2.119     0.03413 *
```

```
month07          0.43330    0.31357    1.382     0.16705
month08          0.78074    0.31783    2.456     0.01405 *
month09          1.34653    0.31173    4.320     1.58e-05 ***
month10          1.53613    0.30672    5.008     5.57e-07 ***
month11          1.30278    0.31005    4.202     2.67e-05 ***
month12          1.29893    0.30831    4.213     2.54e-05 ***
Hour01           0.37829    0.48984    0.772     0.43997
Hour02          -0.20741    0.55219   -0.376     0.70721
Hour03          -0.39798    0.58280   -0.683     0.49470
Hour04          -0.48941    0.64045   -0.764     0.44478
Hour05           0.90129    0.69614    1.295     0.19544
Hour06           0.02591    0.53824    0.048     0.96160
Hour07          -0.29950    0.45675   -0.656     0.51202
Hour08           0.21076    0.44527    0.473     0.63598
Hour09           0.32094    0.43594    0.736     0.46162
Hour10           0.38708    0.45102    0.858     0.39079
Hour11           0.49316    0.44281    1.114     0.26543
Hour12           0.44186    0.43396    1.018     0.30860
Hour13           0.65197    0.43797    1.489     0.13661
Hour14           1.37810    0.43527    3.166     0.00155 **
Hour15           0.64291    0.43920    1.464     0.14327
Hour16           1.10364    0.45695    2.415     0.01574 *
Hour17           0.45293    0.43669    1.037     0.29967
Hour18           0.29963    0.41596    0.720     0.47134
Hour19          -0.30632    0.41864   -0.732     0.46437
Hour20           0.10592    0.41851    0.253     0.80020
Hour21          -0.27460    0.41973   -0.654     0.51297
Hour22           0.06621    0.42440    0.156     0.87603
Hour23          -0.52255    0.43940   -1.189     0.23437
trip_distance  159.00054    2.12130   74.954     < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 7.131 on 12801 degrees of freedom
Multiple R-squared:  0.4666,   Adjusted R-squared:  0.4645
F-statistic: 223.9 on 50 and 12801 DF, p-value: < 2.2e-16
```



From above results we can observe that the R-squared and Adjusted R-square which indicat es how much variance in Target variable is explained by the independent variables, is lower than desired. Also, the different error matrix for validation data is come out as:

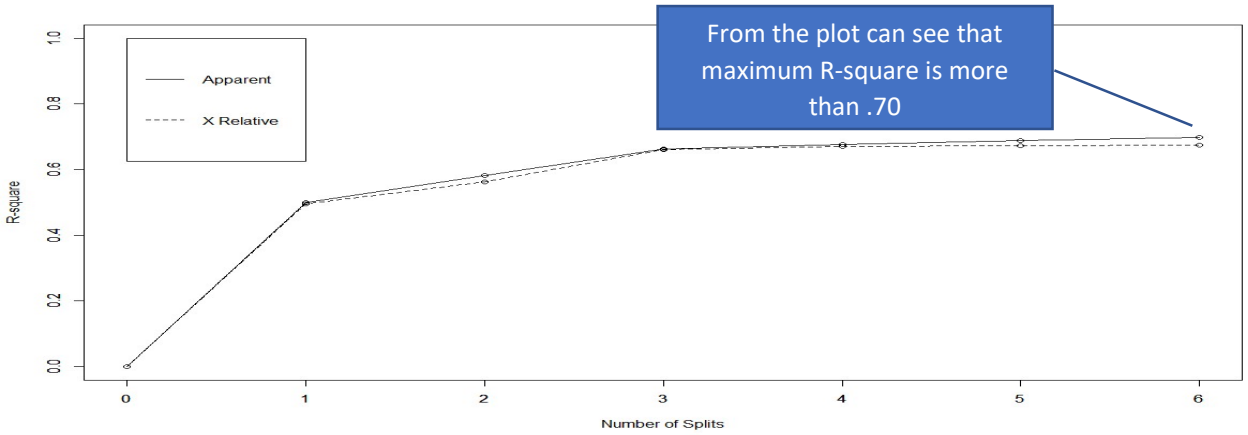
Case	MAE	MSE	RMSE	MAPE
When we drop insensible data	3.1016	57.8934	7.6087	.35299

When We replace insensible data with NA	3.3125	39.9365	6.3195	0.3671

4.2 Decision Tree:

Error matrix for model on Decision tree

Case	MAE	MSE	RMSE	MAPE
When we drop insensible data	2.5199	29.4301	5.4249	0.2616
When We replace insensible data with NA	2.7335	23.6895	4.8671	0.2727

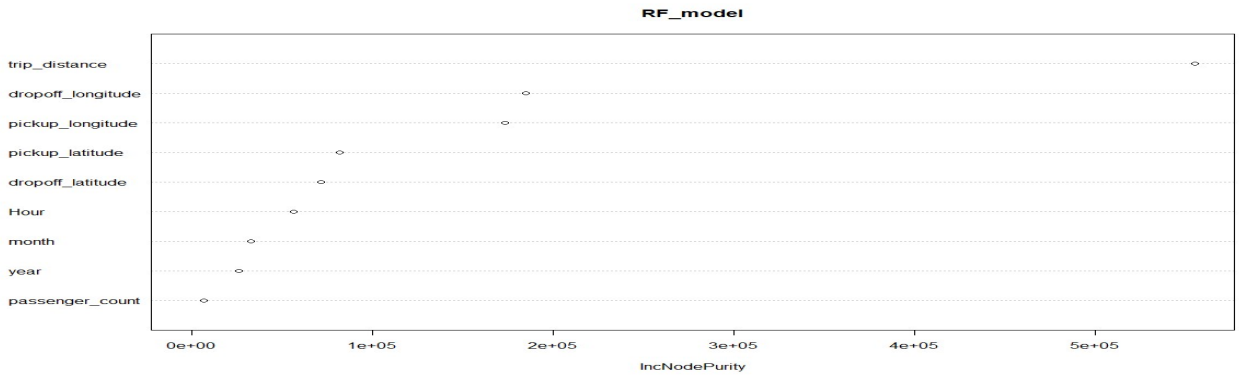


4.3 Random Forest:

Error matrix for model on Decision tree

Case	MAE	MSE	RMSE	MAPE
When we drop insensible data	1.9826	21.6376	4.6516	0.2126
When We replace insensible data with NA	2.0719	14.2777	3.7785	0.22859

Variable importance curve for Random forest is shown below:



On calculating value of R-square for random forest we get 0.826 for case When We replace insensible data with NA and same 0.745 when we drop insensible data.

5. MODEL SELECTION AND PREDICTION

As this is a regression model the R-square value plays important role here.

R-square value is indication of how much variance of target variable is explained by our independent variable. For a good model this R-square should be equal or greater than 0.80.

In case of Linear regression, the value of R-square we get is only 0.466 and value of mean absolute error is from 3.10 to 3.31 and Root mean square error is from 6.3 to 7.6

In case of Decision tree, the value of R-square we get is greater than 0.70 and value of mean absolute error is from 2.5 to 2.75 and Root mean square error is from 4.86 to 5.43

In case of Random forest, the value of R-square we get is only 0.82 and value of mean absolute error is from 1.98 to 2.07 and Root mean square error is from 4.65 to 3.77.

From above, we can see that from all the model we build we get maximum value for R-square i.e. 0.82 from Random forest model which we build on data that we get when we replace all the insensible data with NA and imputed them using missing values imputation method. Also, we can see that from this model we get minimum value for RMSE i.e. 3.77. Hence, we select this model to predict our test data.