**BIKE RENTING**

**ASIF RAZA**

**14TH JULY, 2018**

**Table of Content**

|  |  |
| --- | --- |
| **1. Introduction** | **03** |
| 1.1 Problem of statement | **03** |
| **2. Data** | **03** |
| **3. Pre- processing** | **04** |
| 3.1 Exploration of Numerical Variables | **04** |
| 3.2 Checking the Distribution of Numerical Variables | **04** |
| **3.3 Missing Value Analysis** | **06** |
| **3.4 Checking the outliers** | **07** |
| 3.4.1 Boxplot of all numerical variable with dependent or target variables | **07** |
| 3.4.2 Outliers Treatment | **08** |
| **3.5 Feature Selection** | **08** |
| 3.5.1 Correlation Plot | **09** |
| 3.5.2 Chi-Square test | **09** |
| **3.6 Feature Scaling** | **10** |
| 3.6.1 Normalization Method | **10** |
| **4. Building Predictive model** | **10** |
| **4.1 Linear regression Model** | **10** |
| 4.1.1 Summary of the model | **13** |
| 4.1.2 Model Performance or Evaluation for linear regression model | **15** |
| **4.2 Decision Tree Regression** | **15** |
| 4.2.1 Performance of model | **16** |
| 4.2.2 Model Performance | **17** |
| **4.3 Random forest Algorithm** | **17** |
| 4.3.1 Performance of the model | **18** |
| **4.4 KNN Model** | **18** |
| **4.4 Model Selection** | **18** |
| **5. Conclusion** | **18** |
| **6. Complete R-Code** | **18** |

1. **Introduction**
   1. **Problem Statement**

Here we have bike renting company where objective of this Case is to **prediction of bike rental** count on daily based on the environmental and seasonal settings.

1. **Data**

Our task is to build a Regression Model to predict the bike rental on the daily basis. Here we have a target variable in the form if numerical variable in a form of count so it is Regression type Problem.

As we have 1 dataset which contains 731 Observation and 16 variables. First we used the data for predicting Bike rental count for the daily basis.

Bike Rental dataset ( Columns: 1 - 8 )

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | | instant dteday season yr mnth holiday weekday workingday | | | | | | | | | 1 1/1/2011 1 0 1 0 6 0 | | | | | | | | | 2 1/2/2011 1 0 1 0 0 0 | | | | | | | | | 3 1/3/2011 1 0 1 0 1 1 | | | | | | | | | 4 1/4/2011 1 0 1 0 2 1 | | | | | | | | | 5 1/5/2011 1 0 1 0 3 1 |  |  |  |  |  |  |  | |
| Bike rental dataset ( Columns: 9 - 14 )   |  | | --- | | weathersit temp atemp hum windspeed casual | | 2 0.344167 0.363625 0.805833 0.160446 331 | | 2 0.363478 0.353739 0.696087 0.248539 131 | | 1 0.196364 0.189405 0.437273 0.248309 120 | | 1 0.200000 0.212122 0.590435 0.160296 108 | | 1 0.226957 0.229270 0.436957 0.186900 82 | |
|  |
| Bike rental dataset ( Column: 15-16 ) |
| |  |  | | --- | --- | | registered cnt | | | 654 985 | | | 670 801 | | | 1229 1349 | | | 1454 1562 | | | 1518 1600 |  | |

As we can see that we have 12 variables, using which we have correctly predict the bike rental count on particular day.

|  |  |
| --- | --- |
| S.No | Predictor |
| 1 | Dteday |
| 2 | Season |
| 3 | Year |
| 4 | Month |
| 5 | Holiday |
| 6 | Weekday |
| 7 | Working day |
| 8 | Weathersit |
| 9 | Temp |
| 10 | Atemp |
| 11 | Hum |
| 12 | Wind speed |
|  |  |

1. **Pre- Processing**

**3.1 Exploration of Numerical variable:-**

After getting the data although we have total 731 observation and 16 variables. Out of 16 variables, 7 are numerical variables and 9 are categorical variable out of which 1 dependant variable which is in the form of numerical variable and having in the form of count i.e. number of bike rented per day.

**3.2 Checking the Distribution of Numerical Variables:-**

We plot the Histogram to check the distribution of the Numerical Variables weather variables is normally distributed or not.

|  |  |
| --- | --- |
| Atemp.jpeg | Temperature.jpeg |
| humidity.jpeg | wind speed.jpeg |
| casual.jpeg | registered.jpeg |
| Count.jpeg | |

Here we observe that the entire numerical variable are either left skewed or right skewed except Temperature, Atemp, registered, Total count are Normally distributed. Here Humidity are right skewed, Wind Speed and casual are left skewed. We clearly observe these probability distribution some variable are skewed and some are normally distributed.

**3.3 Missing Value Analysis:-**

Here after doing proper Analysis there is no value missing in the dataset which is shown in the form of below table where we observe that no value missing in the respective variables present in the dataset.

|  |  |
| --- | --- |
| Variables | Missing Value |
| instant | 0 |
| dteday | 0 |
| season | 0 |
| yr | 0 |
| mnth | 0 |
| holiday | 0 |
| weekday | 0 |
| workingday | 0 |
| weathersit | 0 |
| temp | 0 |
| atemp | 0 |
| hum | 0 |
| windspeed | 0 |
| casual | 0 |
| registered | 0 |
| cnt | 0 |

**3.4 Checking for Outliers:-**

We already observe from the above property distribution ( Histogram ) that some of the variables are skewed and some are normally distributed. The Skewed of all distribution can be most likely to be explained by the outliers and extreme value of the data.

**3.4.1 Boxplot of Independent Variable with Dependent Variable:-**

|  |
| --- |
| **box 6.jpeg** |
| **box 7.jpeg** |

We have plot the boxplot of all 6 numerical variable with respect to dependent variables. A lot of useful interference can be made from the plot. First we have seen that we have some outliers and extreme value in each of the variables. We have observe from above box plot that temp, atemp and registered, these variables didn’t have outliers.

Boxplot is used to check the outliers. We observe from the above box plot that **Humidity, Wind speed and Casual** has outliers. This is clearly the effect of outliers and extreme values and we observe the outliers in between the numerical independent variable and dependent target variable.

**3.4.2 Outliers Treatment:-**

After performing Outliers Analysis, removing Outliers using Boxplot command line method or detect or delete the variable. We again plot the variable to check the outliers.

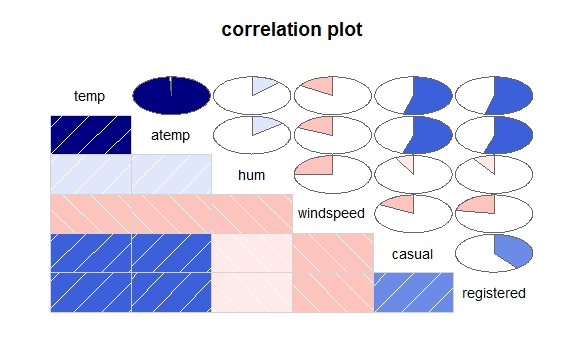
|  |
| --- |
| box after 1.jpeg |
| box 9.jpeg |

After Outliers Treatment, we plot the Boxplot and we observe that almost all the outliers **from Temp, Wind speed and casual variable are removed.**

**3.5 Feature Selection:-**

Before performing any model we need to access the importance of each independent variable. For that we check another pre processing technique to check the correlation analysis between two independent variable and also we check the chi-Square test of Independence between each categorical variable present in the dataset.

**3.5.1 Correlation Plot:-**

 We have plot the correlation plot between all the continuous variable present in the dataset. Here we have 7 continuous variables to check the dependency between the independent variable and dependent variables.

=> Highly Negatively Correlated between two independent variable

=> Highly Positively Correlated between two Independent variables.

Here, we observe that all the independent variable is correlated with dependent variable but no correlation between the two independent variable. So we consider all the numerical independent variable for building the model.

**3.5.2 Chi-Square Test:-**

Here we check the dependency between the 2 categorical variables and can be check by the P( Probability ) Value which we get from the chi-Square Test.

|  |  |
| --- | --- |
| **Categorical Variable** | **P- Value** |
| Instant | 0.245 |
| Dteday | 0.245 |
| Season | 0.5441 |
| Yr(Year) | 0.3677 |
| Month | 0.4918 |
| Holiday | 0.6781 |
| Weekday | 0.4102 |
| Wokingday | 0.4544 |
| Weathersit | 0.6407 |
|  |  |

Here we observe that all the p( Probability ) value > 0.05 which means that we accept the Null Hypothesis saying that all the independent variables are dependent to each other. For Further Analysis we can remove all categorical variables for building the data. Here we have less data we consider all the categorical Variables for building a model.

**3.6 Feature Scaling Method**

**3.6.1 Normalization Method:-**

Here our task is to convert the all numerical variable in the range of 0-1. Here in our dataset we observe that **temperature, Atemp, Humidity & Windspeed** all these 4 variables are already done a feature scaling but casual and registered variable are require a feature Scaling.

1. **Building Predictive Model**

The data is first divided into training and testing set in the proportion of **80:20** ratios for building a model.

**4.1 Linear Regression Model:-**

Provided problem statement is falls under **Regression category.** Linear regression model used for the continuous target variable as we have target variable as a continuous.

We need to find ways to predict the bike rented per day depending on the multiple parameters provided so that we can bike rented per day. We have to split the data into the train and test data.

Now we have divided the data into train and test data. 80% of data is falling under train data category and rest 20% is test data.

Before building linear regression model, we need to check the multi co linearity effect on the data or not.

When two independent variable are highly correlated to each other, then multi collinear effect occurs in the model. It will inflict the variance of different or strong repressor or predictors.

Before building the model, first we need to check the multi collinear effect if the effect is 0 then we feed that model to build the linear regression model.

We have few test which identify the multi collinear effect in the data and one of the test is **Variance Inflation Factor(VIF).**

VIF= 1/ (1-r^2)

where r^2= correlation coefficient

vif(data[,-14])

|  |  |  |
| --- | --- | --- |
| Variables VIF | | |
| 1 season 4.103767 | | |
| 2 yr 2.731931 | | |
| 3 mnth 3.359576 | | |
| 4 holiday 1.096987 | | |
| 5 weekday 1.047200 | | |
| 6 workingday 3.110323 | | |
| 7 weathersit 1.912326 | | |
| 8 temp 2.474220 | | |
| 9 hum 1.931927 | | |
| 10 windspeed 1.225102 | | |
| 11 casual 3.530669 | | |
| 12 registered 5.964035 |  |  |

Here we have a library **USDM**, which help us to influence the VIF factor and implement the VIF factor on all the independent variable. VIF is a function used to check the multi co linearity or any variable which is highly correlated to each other. As we are running this test only for the independent variable and exclude the target variable.

Here we provide a particular value for each of the independent variable by the help of VIF. First VIF take one of the independent variable like **season** then calculate the correlation coefficient with all other **Independent variable** by iterating one by one, and then it will calculate the particular value of independent value based on the correlation value with all other variable.

> vifcor(data[,-13], th= 0.9)

No variable from the 12 input variables has co linearity problem.

The linear correlation coefficients range between:

min correlation ( temp ~ weekday ): -0.0001699624

max correlation ( mnth ~ season ): 0.8314401

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

|  |
| --- |
| Variables VIF |
| season 4.103767 |
| yr 2.731931 |
| mnth 3.359576 |
| holiday 1.096987 |
| weekday 1.047200 |
| workingday 3.110323 |
| weathersit 1.912326 |
| temp 2.474220 |
| hum 1.931927 |
| windspeed 1.225102 |
| casual 3.530669 |
| registered 5.964035 |

We have another metrics which helps us to identity whether to keep the variable or delete the variable from the data set which can be VIFCOR( variance influence factor correlation ), it will take data as first argument, second argument is threshold as we know that correlation ranges from -1 to +1

We used Library USDM for checking the multi collinear effect in the dataset. **Correlation factor range from -1 to 1, if it is -1, then it is highly negatively correlated and if it is +1 then it is highly positively correlated to each other.**

In either case we are not accepting the highly positively or negatively correlated. In this scenario we need to assign a threshold value for which output data is acceptable**. If the correlation value is till the 90% either it is positively correlated or negatively correlated, it is acceptable if the correlation between the two variable is beyond the 0.9 whether that variable is positive or negative we just remove that variable.**

First it finds the pair of variable which has maximum linear correlation, it basically find linear correlation which is actually greater than threshold value and exclude one of them which have greater VIF and remove one of them VIF if high.

VIFCOR repeat this iteration till no variable is highly correlation with other variables. **If VIF > 10 then we have co linearity problem in our variables but here in this case all the variables have VIF < 10 so here there is no correlation effect in dataset.**

After running the collinear test, then Minimum correlation between independent variables (**temp ~ weekday=-0.00016)** which is close to 0< 0.9 and maximum correlation between (**mnth ~ Season** = 0.83) which is less than threshold value. There is no highly positively or negatively correlated variable in our input data.

**Now let us consider these input variables to feed in the model. Now we develop a linear regression model on the top of it.**

> summary(lm\_model)

Call:

lm(formula = cnt ~ ., data = train)

Residuals:

Min 1Q Median 3Q Max

-3.477e-11 -1.960e-13 5.400e-14 3.090e-13 1.632e-11

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Coefficients: |  |  |  |  |  |  |
| Estimate Std. Error t value Pr(>|t|) | | | | | | |
| (Intercept) 2.200e+01 7.134e-13 3.084e+13 < 2e-16 \*\*\* | | | | | | |
| season 2.388e-13 1.652e-13 1.445e+00 0.14897 | | | | | | |
| yr -4.326e-12 3.020e-13 -1.432e+01 < 2e-16 \*\*\* | | | | | | |
| mnth 3.031e-14 4.864e-14 6.230e-01 0.53340 | | | | | | |
| holiday 4.268e-14 5.738e-13 7.400e-02 0.94074 | | | | | | |
| weekday -5.984e-14 4.726e-14 -1.266e+00 0.20591 | | | | | | |
| workingday -1.197e-12 3.508e-13 -3.412e+00 0.00069 \*\*\* | | | | | | |
| weathersit 1.723e-12 2.316e-13 7.440e+00 3.71e-13 \*\*\* | | | | | | |
| temp -9.974e-12 3.688e-12 -2.705e+00 0.00704 \*\* | | | | | | |
| atemp 7.307e-12 4.175e-12 1.750e+00 0.08060 . | | | | | | |
| hum 1.314e-12 8.960e-13 1.466e+00 0.14311 | | | | | | |
| windspeed 6.910e-12 1.319e-12 5.237e+00 2.30e-07 \*\*\* | | | | | | |
| casual 3.408e+03 8.759e-13 3.891e+15 < 2e-16 \*\*\* | | | | | | |
| registered 6.926e+03 9.845e-13 7.035e+15 < 2e-16 \*\*\* | | | | | | |

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.213e-12 on 570 degrees of freedom

Multiple R-squared: 1, Adjusted R-squared: 1

F-statistic: 3.424e+31 on 13 and 570 DF, p-value: < 2.2e-16

**4.1.1 Summary of model:**

Summary can also help you to extract the correlation coefficient of bike count or coefficient of independent variable and also get the r^2 and adjusted r^2. This linear regression model is a statistical model which will calculate the regression coefficient of bike rented count per day which will help us to extract the amount of information of each independent variable to predict the bike rented count per day.

Here we get the summary statistics of error which is called residuals where we got maximum errors 1.632e-11 and minimum errors -3.477e-11.

**Regression Coefficient:**

**Statistical model** does not save the pattern in the memory, we have calculate the regression coefficient which helps us to extract the amount of information each independent variable has contributed.

Estimate is nothing but coefficient of each independent variable. Here we have some estimator value of all the independent variable.

Here we **consider windspeed**, if one unit increases in **windspeed**, then 6.910e-12 times increase in the **bike rented count per day**.

If one unit increase in the **weekday** which leads to 5.984e-14 times decrease in the bike rented count.

This is how we have how we interpret the coefficient.

Here minus ( - )symbol means decrease in the bike rented count and plus ( + ) means increases the bike rented count.

**So from above linear regression model we observe that we increase the bike rented by the help of these variables whose estimator is plus which is**

* Season
* Month
* Holiday
* weathersit
* Atemp
* Humidity
* Windspeed
* Casual
* Registered

Here we have standard error is 2.213e-12 which is also called standard deviation error it measures the average amount that coefficient estimators varies from actual average value of our response variable.

This standard error t-value has contributed to calculate the p-value.

T-value measure how many standard deviation coefficients which are away from 0. Using the t- value we have calculate the p-value

In our model summary, we have get the **p – value** and those variable whose p- value < 0.05 we accept the alternative hypothesis or reject the **NULL Hypothesis** saying that these variable is significant to us.

Here in this model, Variables whose **p-value < 0.05 is**

* **Year(\*\*\*)**
* **Workingday (\*\*\*)**
* **Weathersit(\*\*\*)**
* **Temperature (\*\*)**
* **Windspeed(\*\*\*)**
* **Casual (\*\*\*)**
* **Registered(\*\*\*)**

This above variable is significant to increase the bike rented count but **Year, Workingday, Weathersit, Temperture, Windspeed, Casual, Registered** are highly significant to explain the target variable and **Temperature** is least significant to explain the target variable.

The remaining other variable has p-value>0.05 which means we accept the null hypothesis saying that this independent variables not significant to explain the Bike rented count or target variable.

\*\*\* (3 star)- Highly Significant variable to explain Bike rented count

\*\* (2 Star )- Moderate Significant variable to explain dependent variable

**Adjusted R^2= 1**

**R^2= 1**

Here, 100% of the independent variable has explained the target variable which is acceptable.

Here F – statistics > 1 then it is good relation between predictor and respond or target variable, here we get F- statistics is 3.434e+31 which is reasonably good statistics.

Overall P-Value< 0.05 which is get from F- Statistics model is reasonably good as we get limited number of observation.

**4.1.2 Model Performance or Evaluation for Linear Regression Model**

As this is Regression Problem, so we can take the Mean Absolute Percentage error (RMSE) for predicting the model performance and also get the accuracy of the model.

|  |  |  |  |
| --- | --- | --- | --- |
| **MAE** | **RMSE** | **MAPE** | **MSE** |
| 1.755572e-12 | 2.166289e-12 | 5.034581e-16 | 4.692808e-24 |

**MAPE= 5.034581e-16**

**Accuracy is** quite good accuracy of the model to predict the bike rented count.

**4.2 Decision Tree Regression:-**

Here internally decision tree works by nodes and branches. First important part is we need to select the parent node, that can be done by the information gain .Information gain is the expected amount of information that would be needed.

Out of all the independent variable, the variable which is considered as parent node which is contributing nearly all information of remaining variable. Those variable whose Information gain is high that will be contributing much information compared to all other independent variable.

IG= (Entropy of the system before Split – Entropy of the system after split)

Entropy ( H ) = - ∑iPi log2Pi

Here we find the entropy of the system before split (we keep both target variable and independent variable together to calculate the impurity) & then calculate the entropy after split and then take difference to get the information Gain.

Now the independent variable whose information gain is high is considered as a **Parent Node**. We have total 732 observation and 13 variables. After that we build a model on the train data and then apply that model on the test data to know the accuracy of the model.

Decision tree helps us to Increase the bike rented count, what we predicting that when weather change than what is the probability of in increasing the bike rented count come.

Here we have 12 independent variable after doing proper data exploration, what we do we want to predict the Target variable to know the Bike rented count when new bike comes in the market.

For that we built a decision tree on the top of the data, 1st we need to convert the data into train and test data.

For building a decision tree regression model, we load the library RPART, we are using ANOVA method for building a regression type problem statement.

> fit

n= 584

node), split, n, deviance, yval

\* denotes terminal node

1) root 584 2180865000 4516.765

2) registered< 0.4552411 208 263895000 2514.481

4) registered< 0.3115796 112 42834970 1671.500

8) registered< 0.190875 41 6214940 1066.805 \*

9) registered>=0.190875 71 12970830 2020.690 \*

5) registered>=0.3115796 96 48617090 3497.958

10) casual< 0.2793427 64 8575479 3083.922 \*

11) casual>=0.2793427 32 7127781 4326.031 \*

3) registered>=0.4552411 376 621760800 5624.412

6) registered< 0.6728992 218 149237900 4794.234

12) casual< 0.5259683 181 53624110 4516.983

24) registered< 0.5518337 103 20510640 4204.650 \*

25) registered>=0.5518337 78 9797323 4929.423 \*

13) casual>=0.5259683 37 13639450 6150.514 \*

7) registered>=0.6728992 158 114978400 6769.848

14) casual< 0.2102406 33 4319960 5623.061 \*

15) casual>=0.2102406 125 55802110 7072.600 \*

These models are in the form of tabular but its gets splits. Here we have pattern value, here we have all the decision tree pattern value or rules for the decision tree.

Now we apply decision tree regression model on the test data and find get predicted value of the test data we have actual value then compare with the actual value of the test data and get performance of the model.

.

**4.2.1 Performance of Model**

**We have four type of error which help us to evaluate the performance of these regression model.**  We have 4 error matrices i.e.

* RMSE ( root Mean Square Error)
* MAE (Mean absolute error)
* MAPE ( Mean absolute percentage error)
* MSE ( Mean square error )

Here we have used MAE and MAPE for finding the error matrices because our data is and regression Method.

MAE will give the error in the form of number and MAPE we get the error in the form of percentage so we calculate the MAPE to calculate the error rate.

Here we get the predicted value and compare with the actual value of the target value and got the error which we get through calculating MAPE and get accuracy of the model.

|  |  |  |  |
| --- | --- | --- | --- |
| MAE | RMSE | MAPE | MSE |
| 4.202596e+02 | 5.129908e+02 | 1.229908e-01 | 2.631596e+05 |

Here, we have regr.eval function which is available in library ( DMwR), which helps to calculate the regression error matrix which include MAE(mean absolute error), RMSE(root mean square error), MAPE(mean absolute percentage error ), MSE(mean square error).

**4.2.2Model Performace:-**

Here we get the MAPE= 70% accuracy of the model is good which we get from the decision tree algorithm to explain the Bike Rented Count per day.

* 1. **Random Forest:-**

Now we develop a model by random forest. Now, we run a random forest algorithm. Here we use three fold cross validation in this model due the computational cost. Random forest Packages used for building this model. In this model we get built a model on the train data and apply on the test data.

Here in this case we develop a model and extract to read the 2 rules of the tree.

> exec=extractRules(treelist, train[,-14])

5024 rules (length<=6) were extracted from the first 100 trees.

> rulemetrix[1:2,]

len freq err

[1,] "5" "0.015" "50965.1111111111"

[2,] "6" "0.01" "4060.47222222222"

condition

[1,] "X[,11]<=0.0902288732394366 & X[,11]<=0.0535504694835681 & X[,12]<=0.338362691308114 & X[,12]<=0.167340456251805 & X[,12]<=0.0996246029454231"

[2,] "X[,9]<=0.5971195 & X[,11]<=0.0902288732394366 & X[,11]<=0.0535504694835681 & X[,12]<=0.338362691308114 & X[,12]<=0.167340456251805 & X[,12]>0.0996246029454231"

pred

[1,] "547.666666666667"

[2,] "959.833333333333"

Here, length tells us no of variable value taken in the condition, it means we have 6 variable is included; frequency tells us percentage of data satisfied the condition which is 0.019% in whole data satisfied the condition. Error means amount of percentage error is 50965 which satisfy the data.

**4.3.1Performace of the Model:-**

Here we get the MAPE= 4.753375e-02, accuracy of the model is quite good which we get from the Random Forest algorithm to explain the predict the bike Rented count. Here we draw random forest with 100 trees and after increasing in the tree still we get slight increase in the MAPE, so we can consider the random forest with ntrees=100.

|  |  |  |  |
| --- | --- | --- | --- |
| MAE | RMSE | MAPE | MSE |
| 1.336580e+02 | 2.043152e+02 | 4.753375e-02 | 4.174469e+04 |

* 1. **K Nearest Neighbors:-**

It calculates the distance between each test observation verses all the training observation. In this algorithm I calculate the distance of nearest neighbors. K- Nearest neighbor is the another algorithm based on these we predict the bike rented count per day. Here through KNN model we get the all the extracted all the target value of the test data, store all the predicted value in the KNN Pedictors.

**4.5 Model Selection:-**

We can see that all the model comparatively on average and we select either of the any models without any loss of information

1. **CONCLUSION**

This study analysis the issue of bike rented and explores in detail preventative and corrective actions. Bike Rented per day programs that can be implemented individually or collectively to increase the bike rented per day. Bike Rented count is a serious good business by companies throughout the world. All companies must approach this issue depend on the new bike coming in the market with increase in bike rented count per day. While coming new bike company should do proper advertisement and will increase in the bike rented count and also give proper discount on holiday, weekend or some special occasion to increase the bike rented count.

1. **Complete R-Code**

**rm(list=ls())**

**setwd("C:/users/user")**

**getwd()**

**#load data**

**data= read.csv("day.csv")**

**#dimension of data**

**dim(data)**

**#LOAD Libraries**

**x=c("ggplt2", "corrgram", "DMwR", "Caret", "RandomForest", "unbalance", "C50", "dummies", "e1071", "information", "MASS", "rpart", "gbm", "ROSE")**

**lapply(x, require, character.only=TRUE)**

**names(data)**

**#library for ploting the graph**

**library(scales)**

**library(psych)**

**library(gplots)**

**library(ggplot2)**

**#explore the data**

**str(data)**

**#missing value analysis**

**missing\_val= data.frame(apply(data, 2, function(x)(sum(is.na(x)))))**

**View(missing\_val)**

**#calculate how much missing value in particular variables**

**sum(is.na(data))**

**#view and table of all**

**View(data)**

**table(data$yr)**

**table(data$season)**

**table(data$mnth)**

**table(data$holiday)**

**table(data$weekday)**

**table(data$workingday)**

**table(data$weathersit)**

**table(data$temp)**

**table(data$atemp)**

**table(data$hum)**

**table(data$windspeed)**

**range(data$weathersit)**

**range(data$workingday)**

**range(data$temp)**

**range(data$atemp)**

**range(data$hum)**

**range(data$windspeed)**

**range(data$casual)**

**range(data$registered)**

**range(data$cnt)**

**data[1:5, 1:8]**

**data[1:5, 9:14]**

**data[1:5, 15:16]**

**#histogram plot**

**ggplot(data, aes(x=data$temp))+geom\_histogram(fill="DarkSlateBlue", colour="black")+xlab("Temperature")+ggtitle("Temperature")**

**ggplot(data, aes(x=data$atemp))+geom\_histogram(fill="DarkSlateBlue", colour="black")+ggtitle("Normalized Temperature in Celsius")**

**ggplot(data, aes(x=data$hum))+geom\_histogram(fill="DarkSlateBlue", colour="black")+xlab("Humidity")+ggtitle("Humidity")**

**ggplot(data, aes(x=data$windspeed))+geom\_histogram(fill="DarkSlateBlue", colour="black")+xlab("Wind Speed")+ggtitle("Wind speed")**

**ggplot(data, aes(x=data$casual))+geom\_histogram(fill="DarkSlateBlue", colour="black")+ggtitle("casual")**

**ggplot(data, aes(x=data$registered))+geom\_histogram(fill="DarkSlateBlue", colour="black")+ggtitle("registered")**

**ggplot(data, aes(x=data$cnt))+geom\_histogram(fill="DarkSlateBlue", colour="black")+ggtitle("Total Count")**

**#boxplot distribution and outlier analysis**

**numeric\_index= sapply(data, is.numeric)**

**numeric\_index**

**numeric\_data= data[, numeric\_index]**

**cnames= colnames(numeric\_data)**

**cnames**

**#detect and delete the outliners from all numerical variable by iterating the loop**

**for (i in cnames){**

**print(i)**

**val=data[,i][data[,i] %in% boxplot.stats(data[,i])$out]**

**print(length(val))**

**data=data[which(!data[,i] %in% val),]**

**}**

**library(ggplot2)**

**for (i in 1:length(cnames)) {**

**assign(paste0("gn", i), ggplot(aes\_string((cnames[i]),x= "cnt"), data= subset(data))+**

**stat\_boxplot(geom= "errorbar", width=0.5)+**

**geom\_boxplot(outlier.colour="red", fill="grey", outlier.shape=18,**

**outlier.size=1, notch=FALSE)+**

**theme(legend.position="bottom")+**

**labs(y=cnames[i], x="Bike Rented count")+**

**ggtitle(paste("box plot for", cnames[i])))**

**}**

**gridExtra::grid.arrange(gn9,gn10,gn11, ncol=3)**

**gridExtra::grid.arrange(gn12,gn13, gn14, ncol=3)**

**#detect and delete the outliners from all numerical variable by iterating the loop**

**for (i in cnames){**

**print(i)**

**val=data[,i][data[,i] %in% boxplot.stats(data[,i])$out]**

**print(length(val))**

**data[,i][data[,i] %in% val] =NA**

**}**

**sum(is.na(data))**

**View(data)**

**data=knnImputation(data, k=5)**

**#boxplot analysis after detect and impute outliers**

**for (i in 1:length(cnames)) {**

**assign(paste0("gn", i), ggplot(aes\_string((cnames[i]),x= "cnt"), data= subset(data))+**

**stat\_boxplot(geom= "errorbar", width=0.5)+**

**geom\_boxplot(outlier.colour="red", fill="grey", outlier.shape=18,**

**outlier.size=1, notch=FALSE)+**

**theme(legend.position="bottom")+**

**labs(y=cnames[i], x="Bike rented Count")+**

**ggtitle(paste("box plot for", cnames[i])))**

**}**

**gridExtra::grid.arrange(gn9,gn10,gn11, ncol=3)**

**gridExtra::grid.arrange(gn12,gn13, gn14, ncol=3)**

**#chec shape of data**

**dim(data)**

**data1= subset(data, select= c(temp, atemp, hum, windspeed, casual, registered))**

**data2= subset(data, select=-c(temp, atemp,hum, windspeed, casual, registered))**

**#correlation plot**

**corrgram(data1, order=F,**

**upper.panel=panel.pie, text.panel=panel.txt, main="correlation plot")**

**#chi sqaure of independence and selecting only categorical variable**

**factor\_index=sapply(data, is.factor)**

**factor\_data= data[, factor\_index]**

**factor\_index**

**View(factor\_data)**

**data**

**#chi-square test data**

**#chi-square test**

**for (i in 1:9){**

**print(names(data2)[i])**

**print(chisq.test(table(data2$cnt, data2[,i])))**

**}**

**#dimension Reduction Method**

**data= subset(data, select= -c(instant, dteday, atemp))**

**#normalizaion method**

**cnames1=c( "casual", "registered")**

**for (i in cnames1){**

**print(i)**

**data[,i]= (data[,i]-min(data[,i]))/(max(data[,i]-min(data[,i])))**

**}**

**View(data)**

**dim(data)**

**#Build a model**

**#decision tree alogorithm(regression problm)**

**library(rpart)**

**library(MASS)**

**library(randomForest)**

**#sampling technique**

**train\_index= sample(1:nrow(data), 0.8\*nrow(data))**

**train= data[train\_index,]**

**test=data[-train\_index,]**

**#rpart for regression**

**fit= rpart(cnt~., data=train, method="anova")**

**fit**

**#predict for new test case**

**prediction\_DT= predict(fit, test[,-13])**

**prediction\_DT**

**#calculate MAPE**

**MAPE= function(y, yhat){**

**mean(abs((y-yhat)/y))\*100**

**}**

**MAPE(test[,13], prediction\_DT )**

**#alternative method**

**regr.eval(test[,13], prediction\_DT, stats = c("mae", "rmse", 'mape', 'mse'))**

**#linear regression method**

**#check multicollinearity**

**library(usdm)**

**vif(data[,-13])**

**vifcor(data[,-13], th= 0.9)**

**#Run regression model**

**lm\_model= lm(cnt~., data = train)**

**summary(lm\_model)**

**#Predict the target variable with absence of actual target variable**

**prediction\_LR=predict(lm\_model, test[,1:12])**

**MAPE(test[,13], prediction\_LR)**

**prediction\_LR**

**#Performance of model**

**regr.eval(test[,13], prediction\_LR, stats = c("mae", "rmse", 'mape', 'mse'))**

**#random forest algorithm**

**RF\_model= randomForest(cnt ~., train, importance=TRUE, ntree=500)**

**#extract rules from random forest**

**library(inTrees)**

**treelist= RF2List(RF\_model)**

**exec=extractRules(treelist, train[,-13])**

**exec[1:2,]**

**#make rules more readable**

**readablerules= presentRules(exec, colnames(train))**

**readablerules[1:2,]**

**rulemetrix=getRuleMetric(exec, train[,-13], train$cnt)**

**rulemetrix[1:2,]**

**RF\_prediction= predict(RF\_model, test[, -13])**

**RF\_prediction**

**#Performance of model**

**regr.eval(test[,13], RF\_prediction, stats = c("mae", "rmse", 'mape', 'mse'))**

**#KNN(k nearest neighbour) classification**

**library(class)**

**library(e1071)**

**KNN\_Model = knn(cnt~. , train, trControl = x, metric = "ROC",tuneLength = tunel)**

**knn\_predictions= knn(train[,1:13], test[,1:13], train$cnt, k=7)**

**knn\_predictions**

**#Performance of model**

**regr.eval(test[,13], knn\_predictions, stats = c("mae", "rmse", 'mape', 'mse'))**