Bike Rental Count Prediction

Navin Kumar K

Content

1. Problem Statement

The objective of this Case is to Predication of bike rental count on daily based on the environmental and seasonal settings.

2. Data

The Bike Rental Data contains the daily count of rental bikes between the year 2011 and 2012 with corresponding weather and seasonal information. The task is to build Regression model which will give the daily count of rental bikes based on weather and season

The Sample of data to predict the count of bikes is shown below :

Instant	Dteday	season	Yr	Mnth	Holiday	weekday	workingday	weathersit
1	01-01-2011	1	0	1	0	6	0	2
2	02-01-2011	1	0	1	0	0	0	2
3	03-01-2011	1	0	1	0	1	1	1
4	04-01-2011	1	0	1	0	2	1	1
5	05-01-2011	1	0	1	0	3	1	1
6	06-01-2011	1	0	1	0	4	1	1

Table 2.1 : Bike Rental Sample Data (Columns: 1-8)

Temp	Atemp	Hum	windspeed	casual	registered	cnt
0.344167	0.363625	0.805833	0.160446	331	654	985
0.363478	0.353739	0.696087	0.248539	131	670	801
0.196364	0.189405	0.437273	0.248309	120	1229	1349
0.2	0.212122	0.590435	0.160296	108	1454	1562
0.226957	0.22927	0.436957	0.1869	82	1518	1600
0.204348	0.233209	0.518261	0.0895652	88	1518	1606

Table 2.2: Bike Rental Sample Data (Columns: 9-14)

3. Data Pre Processing

In Data Pre-processing / Exploratory Data Analysis, we used to understand the data by plotting and comparing the features. This method includes data cleaning, merging, sorting, plotting the features. As this is an regression problem, the numerical variable should be normally distributed.

3.1 Distribution of continuous variable

From Fig 3.1.1 to 3.1.7 explain the distribution of numerical variables in given dataset including target variable cnt. The target variable cnt is normally distributed. Independent variables like 'temp', 'atemp', and 'registered' data is distributed normally. Independent variable 'casual' data is slightly skewed to the right so, there is chances of getting outliers. Other Independent variable 'hum' data is slightly skewed to the left. The presence of outliers in the skewed data is detected in outlier detection.

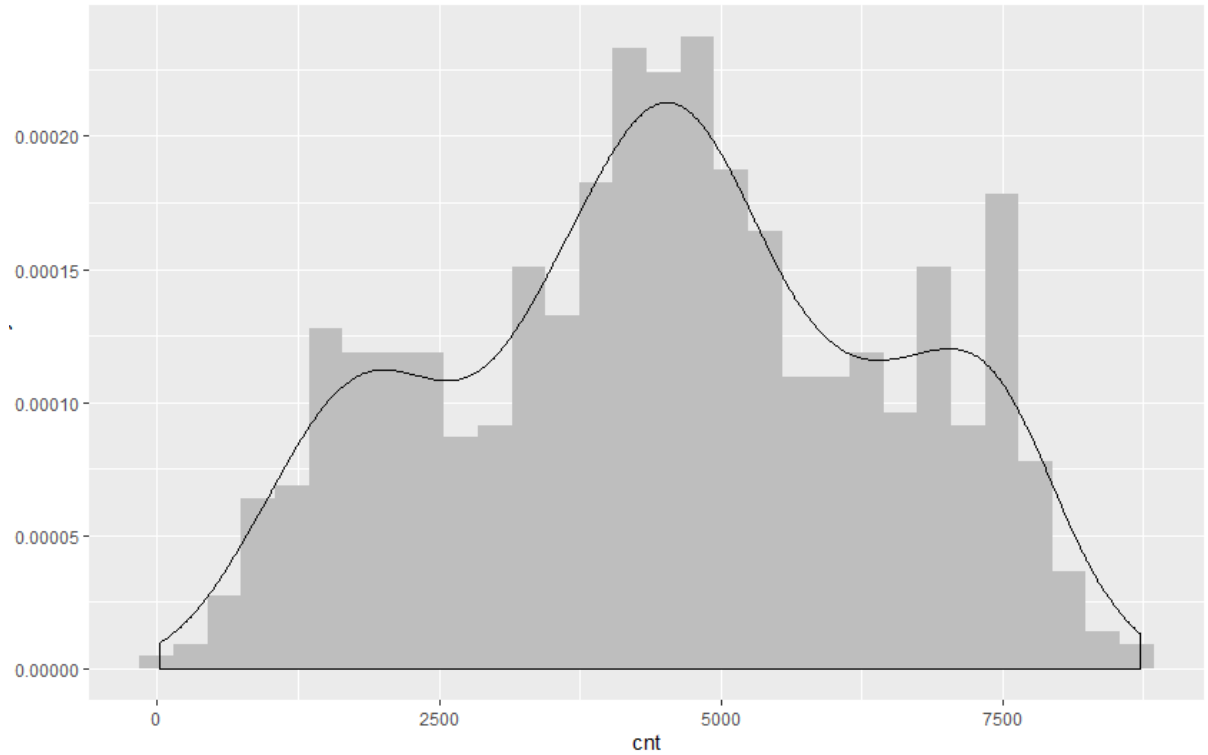

Fig.3.1.1 Distribution of count feature

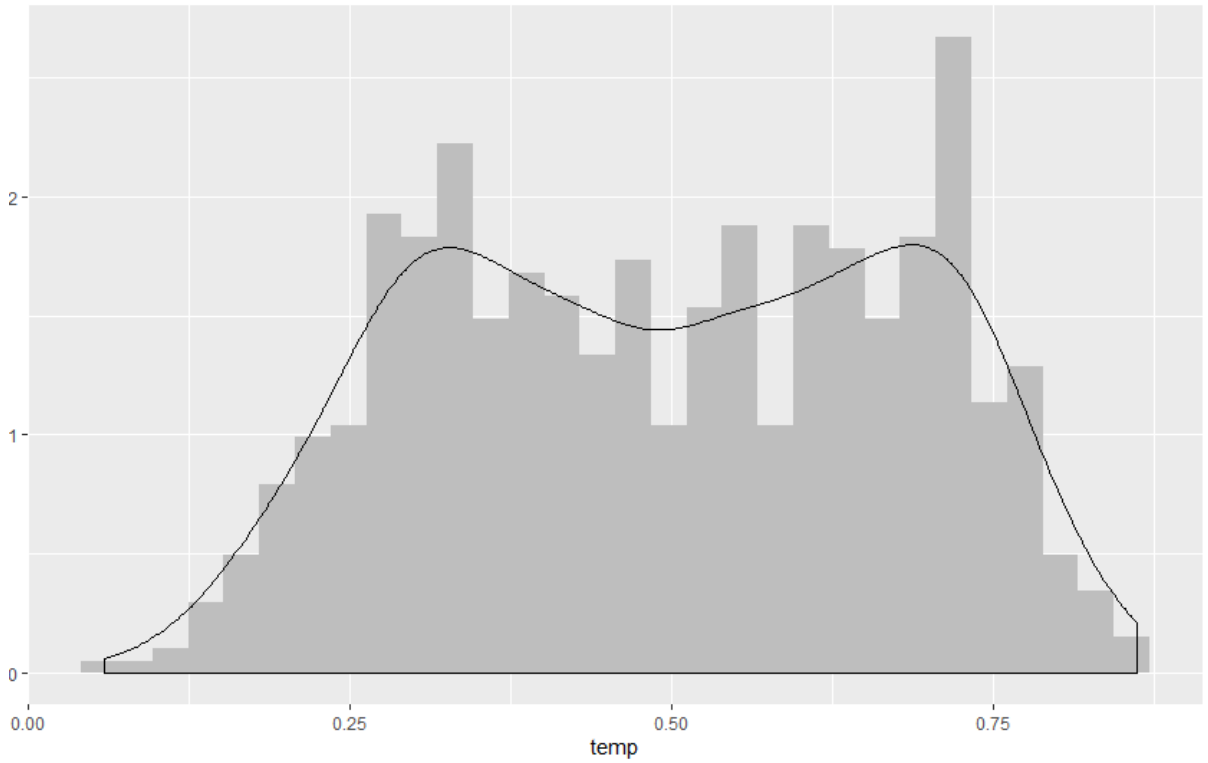

Fig.3.1.2 Distribution of temp feature

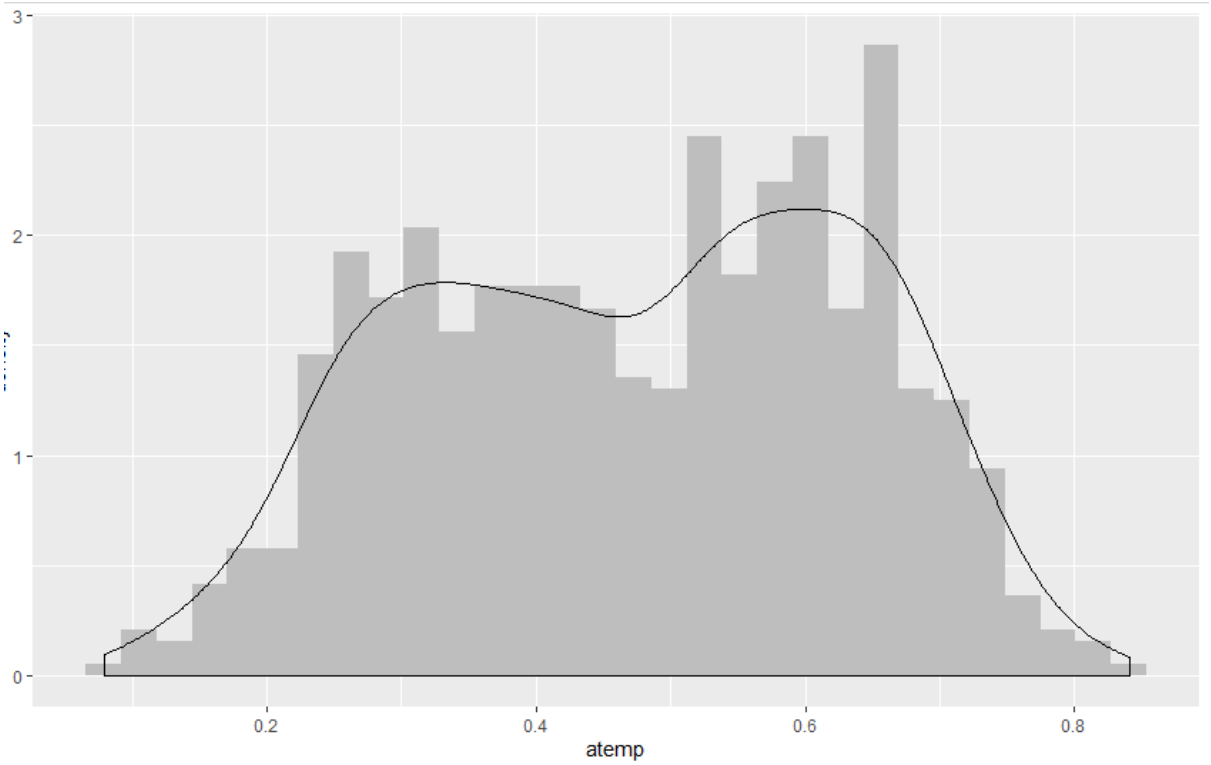

Fig.3.1.3 Distribution of atemp feature

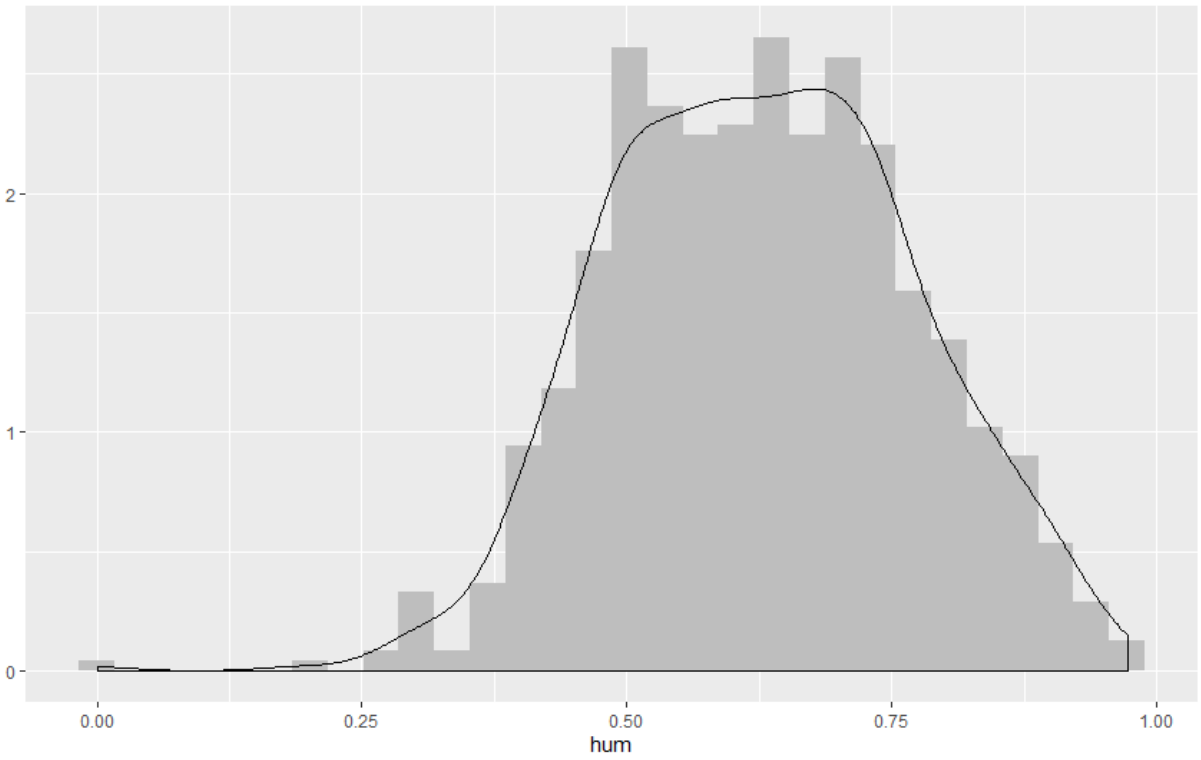

Fig.3.1.4 Distribution of humidity feature

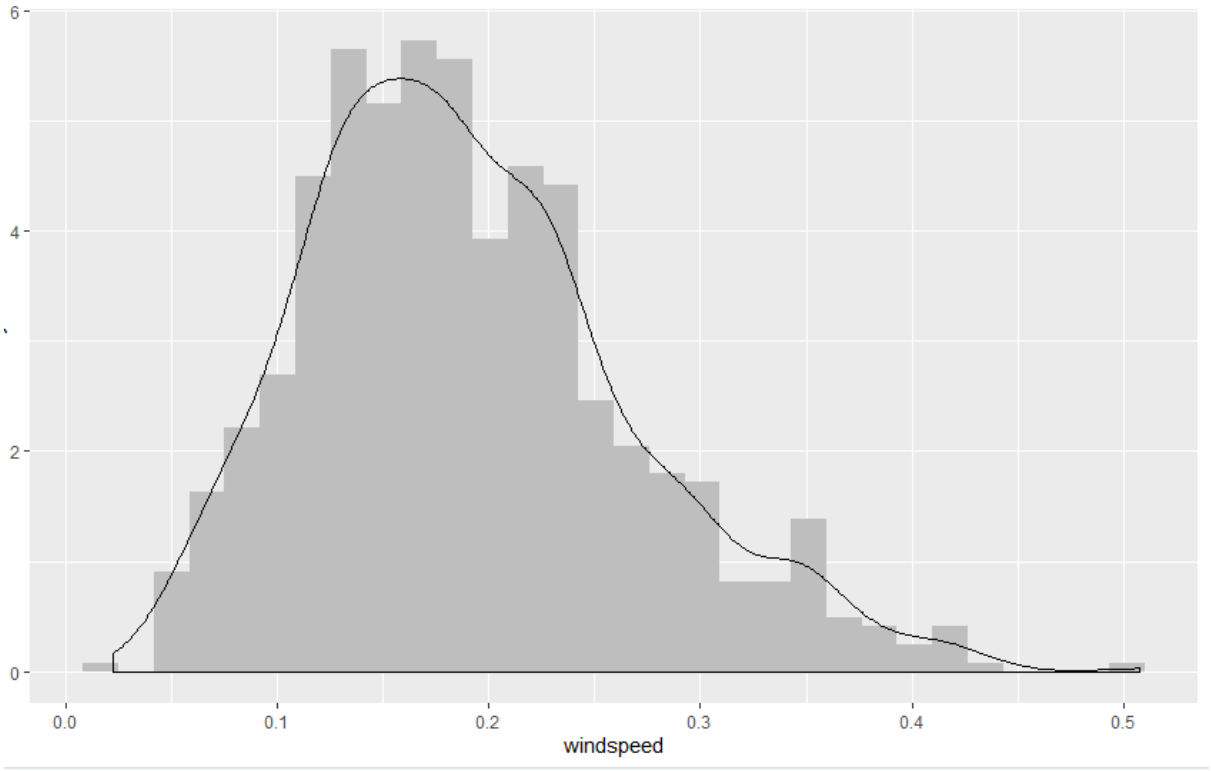

Fig.3.1.5 Distribution of windspeed feature

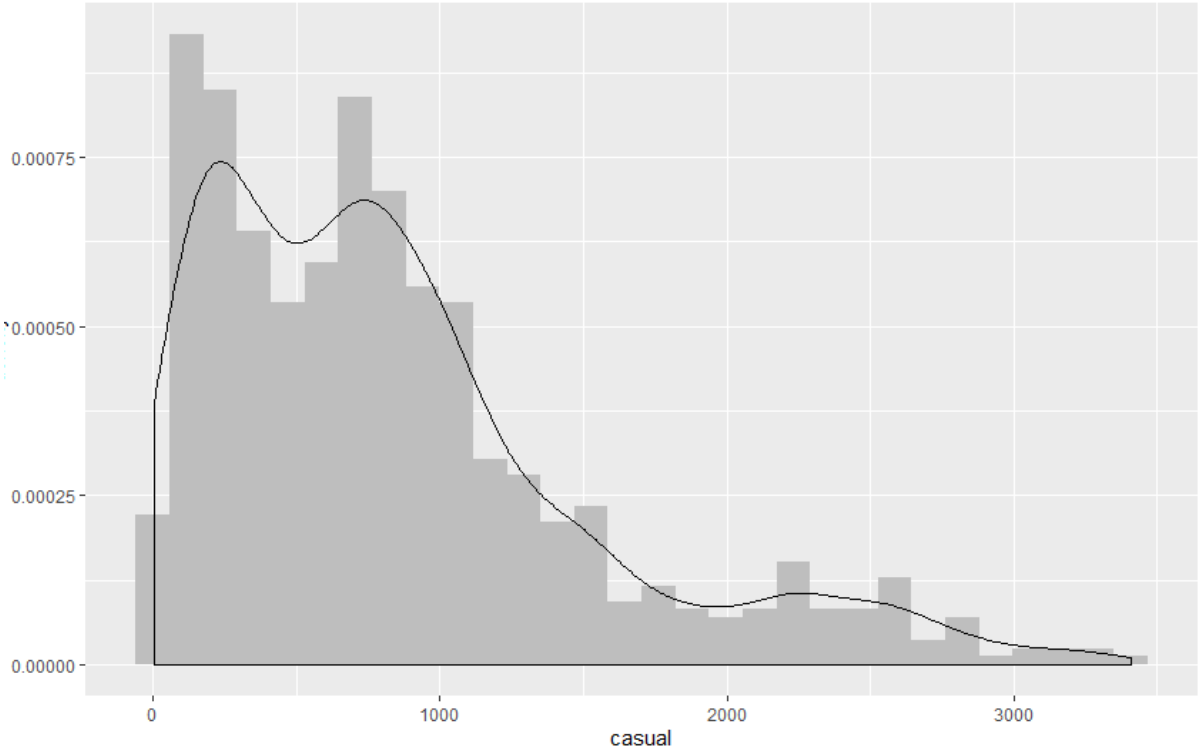

Fig.3.1.6 Distribution of casual feature

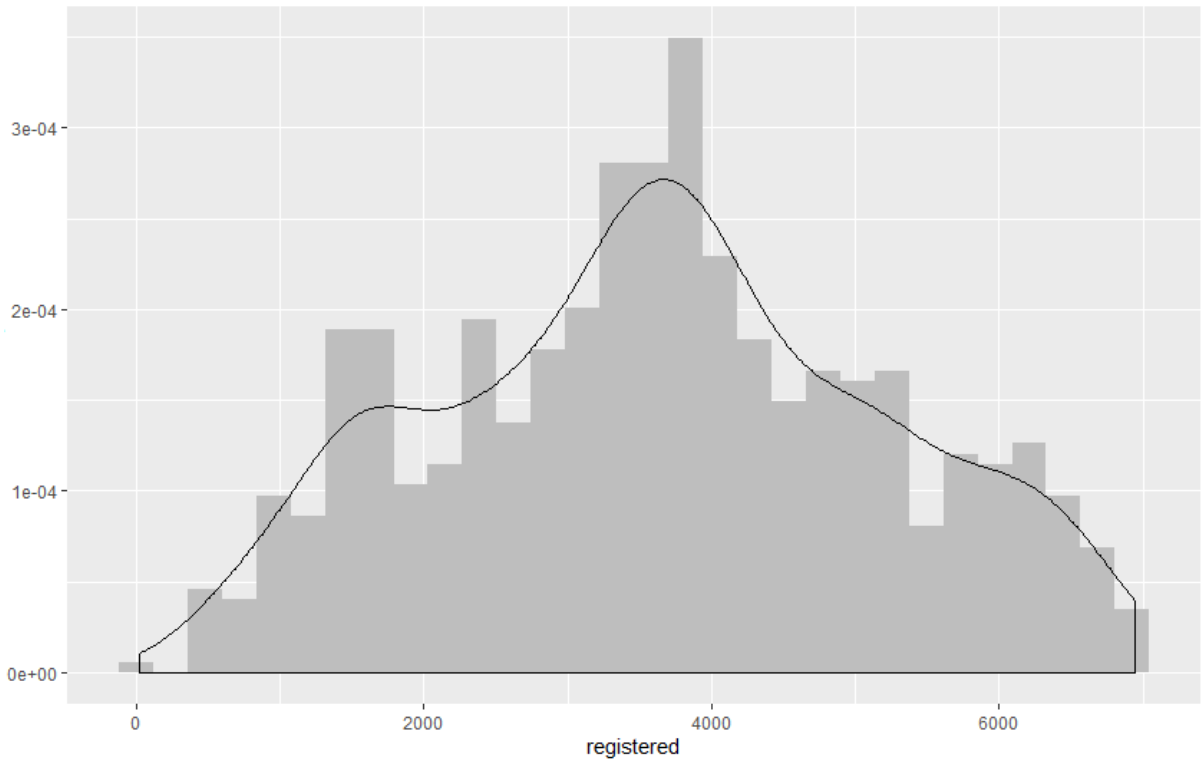

Fig.3.1.7 Distribution of registered feature

3.2 Distribution of Categorical variable

The categorical features are distributed as below,

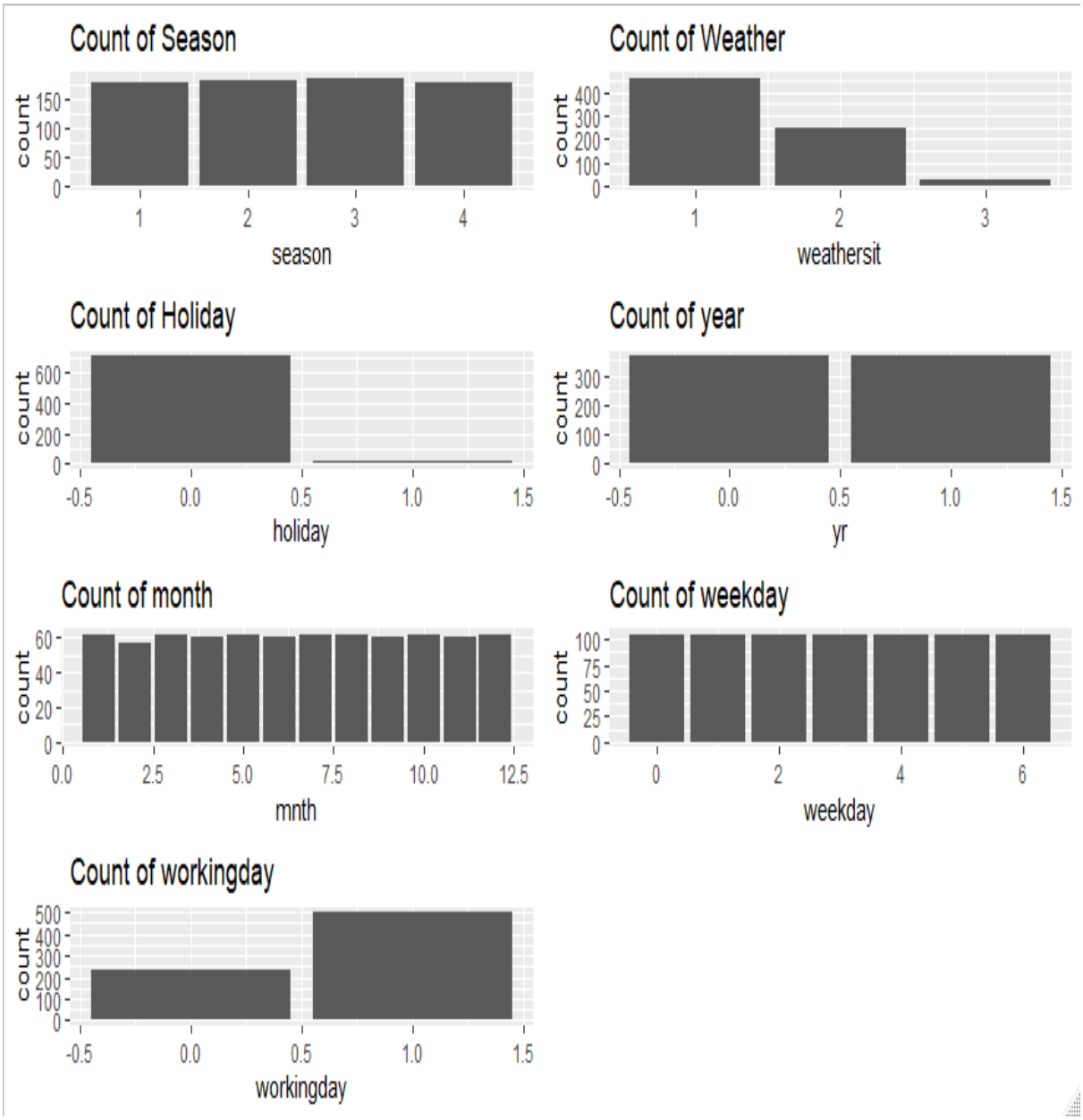

Fig.3.2.1 Distribution of Categorical features

4. Missing Value Analysis

Missing value analysis plays an vital role in building effective model with dataset having Missing/Null values in it's row or column cell. The Data with missing value leads to produce an skewed model. There is no missing values present in this dataset.

S.No	Variables	Missing values
1	Dteday	0
2	Season	0
3	Yr	0
4	Mnth	0
5	Holiday	0

6	weekday	0
7	workingday	0
8	weathersit	0
9	Temp	0
10	Atemp	0
11	Hum	0
12	Windspeed	0
13	Casual	0
14	Registered	0
15	Cnt	0

Table 4.1 : Missing values in given dataset

5. Outlier Analysis

The data point that differ significantly from the observations is known as **outliers**. The outliers can be detected using **boxplot method**. If the outlier has detected in the dataset, then it can be handled either by dropping or by imputing the outliers using Mean, Median, Mode, KNN method.

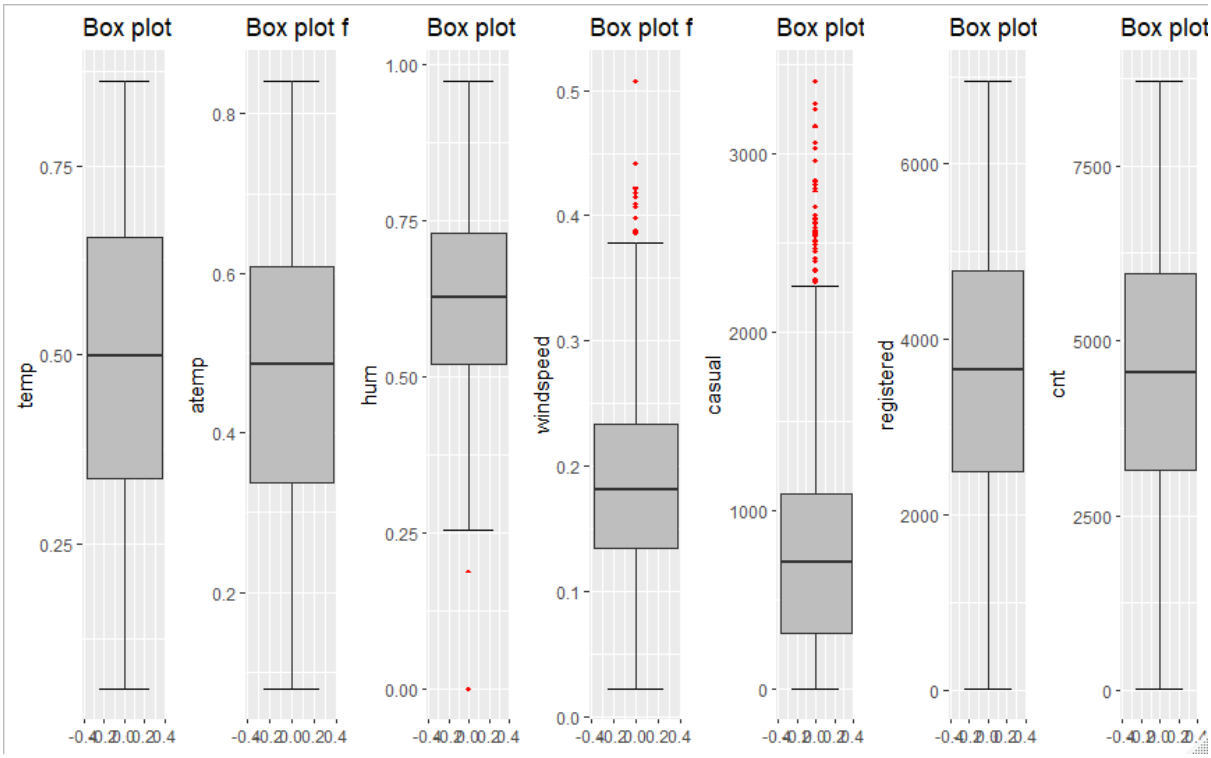

Fig 5.1 Outliers before outlier imputation

The list of independent features having outliers are :

- hum
- windspeed
- casual

As there are 44 outliers in Casual, we are not deleting outliers, which leads to loss of large percent of observation, so we are imputing the outliers using Mean method.

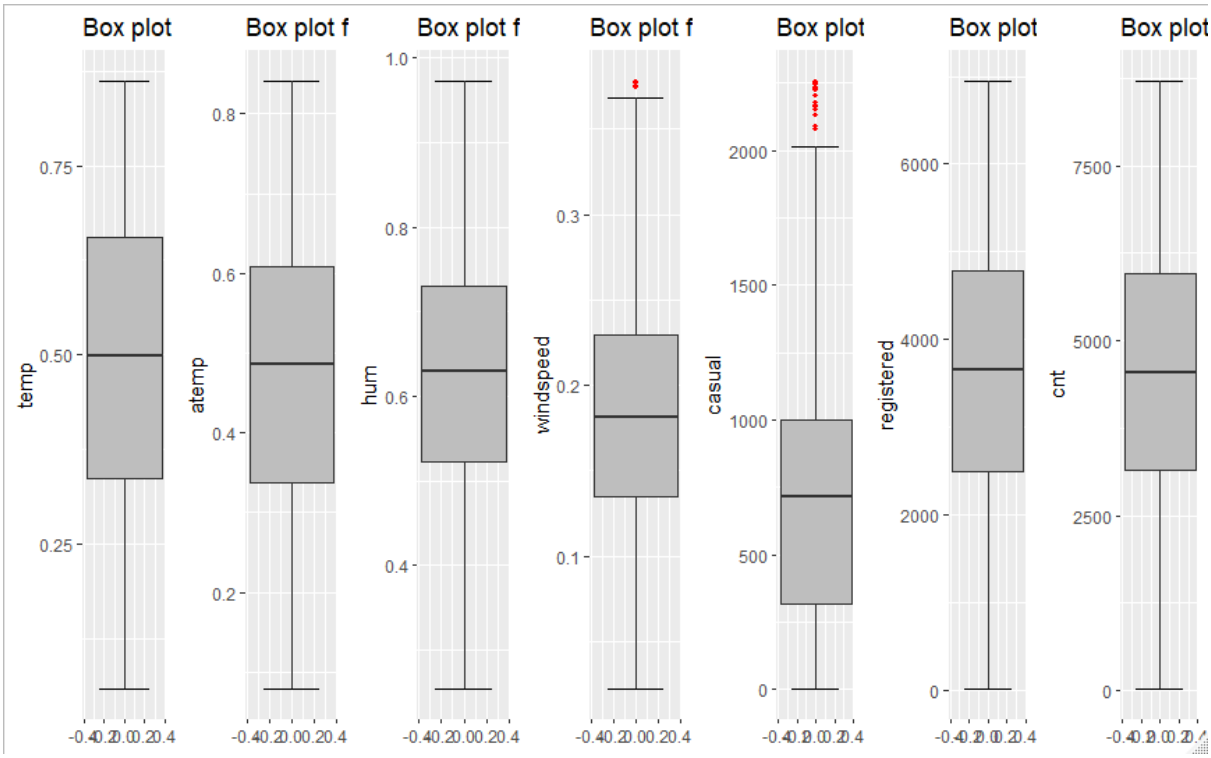

Fig 5.2 : Outliers after Outlier imputation

6. Features Selection

As it is very important to keep only relevant features for the model and the irrelevant features can be removed to produce an powerful model. This becomes even more important when the number of features are very large. The correlation analysis is used to find the relationship between numerical variables.

The relevant feature for model should accept the below criteria :

- The relationship between two independent variable should be less.
- The relationship between Independent and target variables should be high.

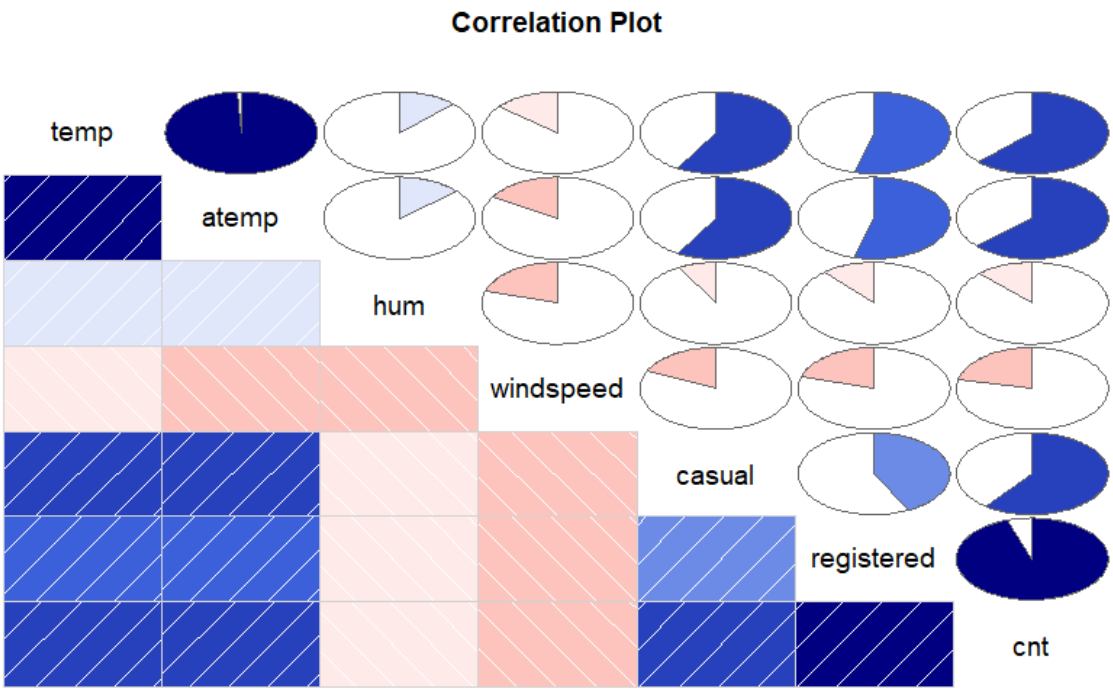

Fig 6.1 : Corrgram plot to illustrates the relationship between all numeric variables.

- Dark blue colour indicates that there is strong positive relationship and if darkness is decreasing indicates relation between variables are decreasing.
- Dark red colour indicates that there is strong negative relationship and if darkness is decreasing indicates relationship between variables are decreasing.

	Temp	Atemp	Hum	windspeed	Casual	registered	cnt
Temp	1	0.9917016	0.12370305	-0.1399995	0.58446835	0.540012	0.627494
Atemp	0.9917016	1	0.13729261	-0.1653145	0.58360342	0.5441918	0.6310657
Hum	0.123703	0.1372926	1	-0.2018941	-0.07980945	-0.1122382	-0.1215178
Windspeed	-0.1399995	-0.165315	-0.20189406	1	-0.18037849	-0.2041117	-0.2164729
Casual	0.5844683	0.5836034	-0.07980945	-0.1803785	1	0.4250802	0.6032946
Registered	0.540012	0.5441918	-0.11223825	-0.2041117	0.42508016	1	0.9455169
Cnt	0.627494	0.6310657	-0.12151781	-0.2164729	0.60329456	0.9455169	1

Table 6.1 : Corrlation value between all numerical values.

Multicollinearity detection using VIF :

VIF quantifies the multicollinearity between the independent variables. Linear regression will work well if multicollinearity between the Independent variables are less. It is acceptable if VIF < 10.

2 variables from the 7 Numerical variables have collinearity problem:

atemp, cnt

After excluding the collinear variables, the linear correlation coefficients ranges betwee
n:
min correlation (casual ~ hum): -0.07980945
max correlation (casual ~ temp): 0.5844683

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

Variables	VIF
temp	1.956175
hum	1.165044
windspeed	1.125537
casual	1.633271
registered	1.557076

7. Dimension Reduction for numeric variables

From Fig 6.1, it shows that there is strong relationship between independent variables ‘temp’ and ‘atemp’. So only one among the two feature is enough to build the model. Casual and registered is the features that we want to find, ignoring these variables. Subsetting independent features atemp, casual, registered, instant, dteday from actual dataset.

```
> # Dimension Reduction
> df = subset(df, select = -c(atemp, casual, registered, instant, dteday))
```

Fig 7.1 : R Code for Dimension Reduction

8. Feature Engineering

It is important to convert the features into required format to produce an effective model. So converting all the categorical features as factor in R and as Category in Python.

'data.frame': 731 obs. of 11 variables:
$ season   : Factor w/ 4 levels "1","2","3","4": 1 1 1 1 1 1 1 1 1 ...
$ yr   : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 ...
$ mnth   : Factor w/ 12 levels "1","2","3","4",..: 1 1 1 1 1 1 1 1 ...
$ holiday   : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 ...
$ weekday   : Factor w/ 7 levels "0","1","2","3",..: 7 1 2 3 4 5 6 7 ...
$ workingday: Factor w/ 2 levels "0","1": 1 1 2 2 2 2 2 1 1 2 ...
$ weathersit: Factor w/ 3 levels "1","2","3": 2 2 1 1 1 1 2 2 1 1 ...
$ temp   : num 0.344 0.363 0.196 0.2 0.227 ...
$ hum   : num 0.806 0.696 0.437 0.59 0.437 ...
$ windspeed : num 0.16 0.249 0.248 0.16 0.187 ...
$ cnt   : num 985 801 1349 1562 1600 ...

9. Creating Dummy variables

For models like linear regression, it is easier for model to process the quantitative variables than qualitative variables. So the categorical variable has been converted into series of zeros and ones by creating dummy variables as below,

"season_1", "season_2", "season_3", "season_4", "mnth_1", "mnth_2", "mnth_3", “mnth_4", ”mnth_5", ”mnth_6", ”mnth_7", ”mnth_8", ”mnth_9", ”mnth_10", "mnth_11", ”mnth_12", ”weekday_0", ”weekday_1", ”weekday_2", ”weekday_3", ”weekday_4", "weekday_5", ”weekday_6", ”weathersit_1", ”weathersit_2", ”weathersit_3", ”yr_0", ”yr_1", "holiday_0", ”holiday_1", ”workingday_0", ”workingday_1", ”temp", ”hum", ”windspeed", "cnt"

10. Data Sampling

The train and test data has been split in 80-20% i.e. 80% in train and 20% in test dataset. The model will build on train data and implement the model in test data to find the target.

- Dimension of Train data : (584,36)
- Dimension of Test data : (147,36)

11. Model Development

In this project, as we are predicting the count of rental bikes. It is an Regression problem. So the below mentioned models has been build to predict the target variables.

- Multiple Linear Regression
- Decision Tree
- Random Forest

The various error metrics as mentioned below has been used to detect the model performance as below,

- **MAPE** : (Mean Absolute Percent Error) measures the size of the error in percentage terms. It is calculated as the average of the unsigned percentage error.

$$MAPE = \frac{1}{n}\sum_{i=1}^{t}\left|\frac{A_t - P_t}{A_t}\right| * 100$$

- **RMSE :** (Root Mean Square Error) is a standard way to measure the error of a model in predicting quantitative data.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}(P_i - A_i)^2}{N}}$$

- **Accuracy :** Measurement of similarity between the predicted and actual value.

11.1. Multiple Linear Regression

Multiple linear regression is the most common form of linear regression analysis. It is used to explain the relationship between one continuous dependent

variable and two or more independent variables. The independent variables can be continuous or categorical.

The Below R Code is used to build and predict the target variable using Multiple lin ear regression.

```
# Train the Linear regression model
lm_model = lm(cnt ~., data = train)

# Summary of the model
summary(lm_model)

# Predict the test data
predictions_LR = predict(lm_model, test[,-36])
```

Fig 11.1 : R Code for Multiple linear regression

Here,
Residual standard error : 763.8 on 556 degrees of freedom
Multiple R-squared : 0.8484,
Adjusted R-squared : 0.8411
F-statistic : 115.3 on 27 and 556 DF,
p-value : < 2.2e-16

Evaluating Regression Model

The Difference between actual and predicted value is known as **Error / Residuals**. This method is used to evaluating performance of model. The below factors are are considered for model performance.

- MAPE value : 21.88%
- RMSE value : 859.869
- Accuracy : 78.12%

11.2. Decision Tree

A Decision Tree is a supervised learning model used to predict a target by learning decision rules from features. In this model the data can be breakdown by making decisions based on number of questions. It is used for both **classification and regression problems**.

The below R code is used to build and predict the target variable using decision tree algorithm,

```
> # Train the  Decision Tree model
> fit = rpart(cnt ~ ., data = train, method = "anova")
>
> # Predict the test data
> predictions_DT = predict(fit, test[,-36])
```

Fig 11.2.1 : R Code for Decision Tree

The variable importance for this model is,

Temp	yr_0	yr_1	season_1	season_3	mnth_12	mnth_1	mnth_2
973923100	638003319	638003319	485342594	190738061	181825627	176199737	154873592

Hum	season_4	Windspeed	mnth_11	mnth_10	weathersit_3	weekday_1
131405658	97207802	74540546	49838600	12655005	12317155	8885461

Table 11.2.1 : Variable importance of Decision Tree model

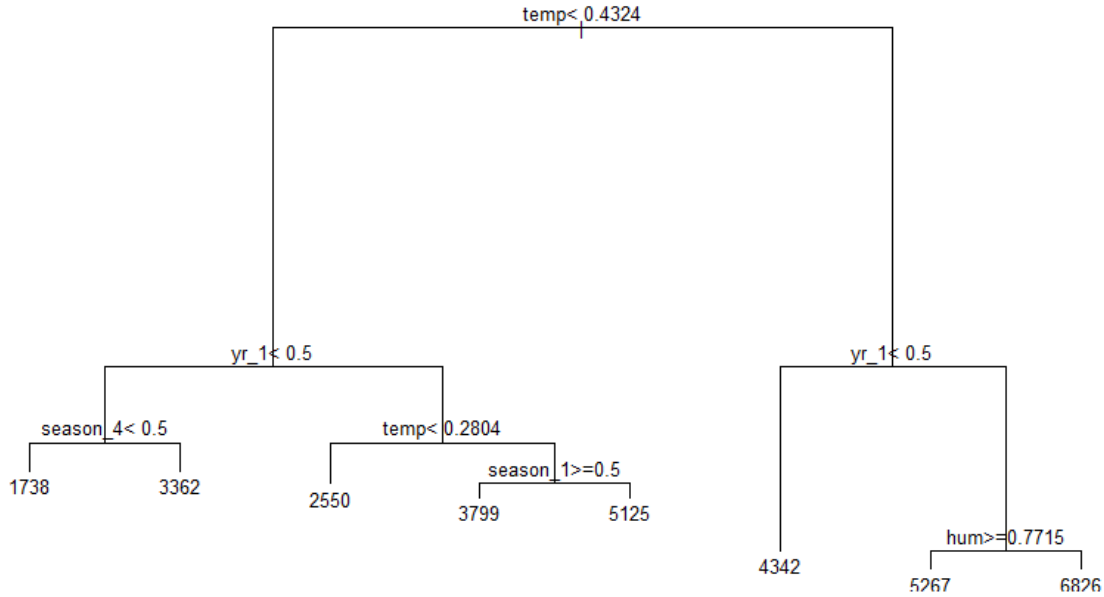

Fig 11.2.2 : Decision Tree for Bike Rental Prediction

Evaluation of Decision Tree Model

- MAPE value : 27.81%
- RMSE value : 1029.00
- Accuracy : 72.19%

11.3. Random Forest

Ensemble learning is a technique that combines the prediction of multiple machine learning algorithms together to make more accurate predictions than individual model. Random forests or random decision forests is an example of ensemble learning method for classification and regression, which aggregates multiple decision trees.

As we saw in section 11.2 Decision tree is overfitting and its accuracy, MAPE and RMSE is also poor, In order to improve the accuracy Random Forest is used.

The below R code is used to build and predict the target variable using random forest algorithm,

```
> # Train the Random Forest model
> rf_model = randomForest(cnt~., data = train, ntree = 500)
>
> # Predict the test data
> RF_Predictions = predict(rf_model, test[,-36])
```

Fig 11.3 : R Code for random forest algorithm

Number of trees : 500
No. of variables tried at each split : 11
% Var explained : 88.06

Our Random Forest model is looking quite good where it utilized maximum variables to predict the count values

Evaluation of Random Forest

- MAPE value : 18.88%
- RMSE value : 736.977
- Accuracy : 81.12%

Conclusion

Based on Error Metrics, the Random Forest Model is best model to predict the Bike rental count.