Predicting Rental Bikes

Bhupendra Kumar
December 2019

Contents

1. Introduction:

1.1 Problem Statement:

The Bike Rental Data contains the daily count of rental bikes between the year 2011 and 2012 with corresponding weather and seasonal information. We would like to predict the daily count of rental count in order to automate the system.

1.1 Data:

We are supposed to build a Regression model as our target variable 'cnt' is continuous, which gives us daily count of bikes getting rented between year 2011-2012.

Given below is a sample of the data set, we are using to predict the cnt.

Table 1.1: Bike Rental Sample Data (Columns: 1-8)

instant	dteday	season	yr	mnth	holiday	weekday	weathersit
1	1/1/2011	1	0	1	0	6	2
2	1/2/2011	1	0	1	0	0	2
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

Table 1.2: Bike Rental Sample Data (Columns: 9-15)

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

Below are the predictor variables, will help us to predict total number of counts.

Table 1.3: Predictor variables

Sr.no.	Variables
1	dteday
2	season
3	yr
4	mnth
5	holiday
6	weekday
7	workingday
8	weathersit
9	temp
10	atemp
11	hum
12	windspeed
13	casual
14	registered

2. Methodology:

2.1 EDA(Exploratory Data Analysis):

Any predictive modeling requires that we look at the data before we start modeling. However, in data mining terms *looking at data* refers to so much more than just looking. Looking at data refers to exploring the data, cleaning the data as well as visualizing the data through graphs and plots. This is often called as **Exploratory Data Analysis**. To start this process we will first try and look at all the distributions of the Numeric variables. Most analysis like regression, require the data to be normally distributed.

2.1.1 Univariate Analysis:

In Figure **2.1 and 2.2** we have plotted the probability density functions numeric variables present in the data including target variable cnt..

i. Target variable cnt and numeric predictors i.e. temp, atemp and registered are normally distributed.

ii. Though 'registered feature is normally distributed, it's range is different. We need to scale the said feature.

iv. Independent variable 'casual' is highly left right skewed, it's more likely we get outliers there.

Figure 2.2 Distribution of target variable (cnt)

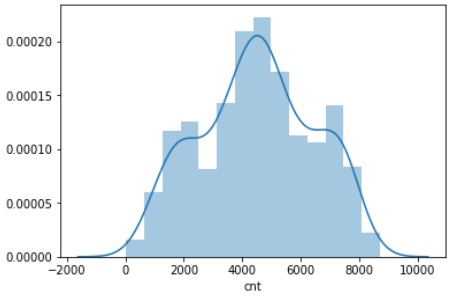
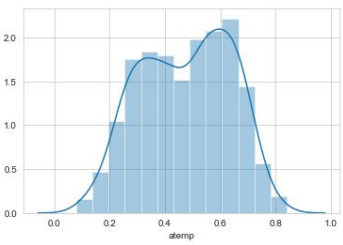
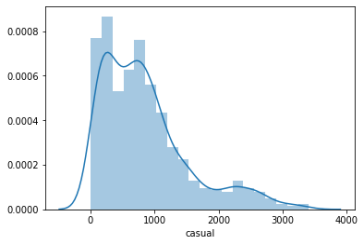
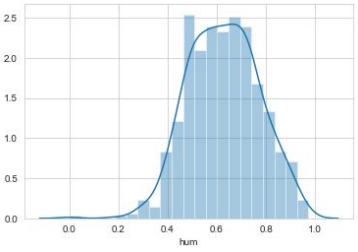
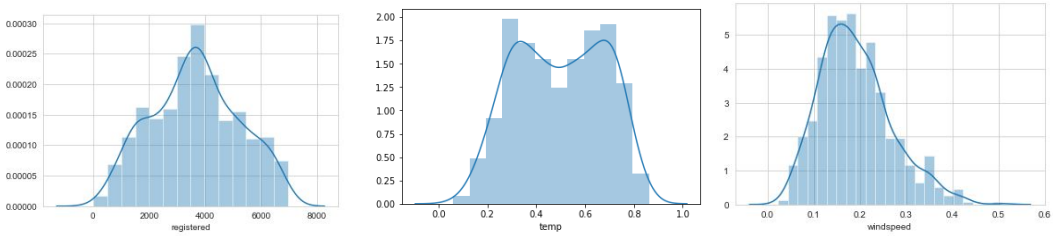


Figure 2.2 Distribution of independent variables (python code in Appendix B)









2.1.2 Multivariate Analysis:

We are doing bi-variate analysis over here, trying to get relationship between each continuous feature amongst each other. Specially, relationship with target variable 'cnt'.

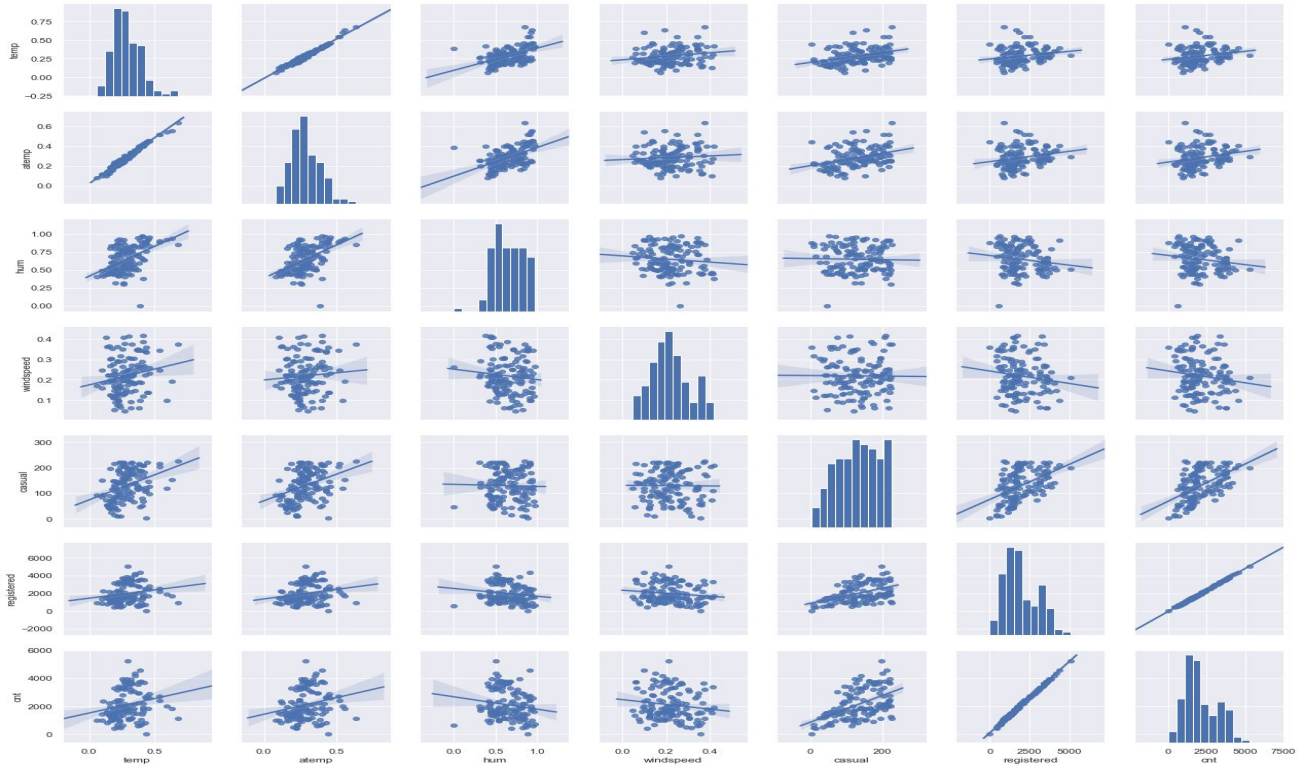
Below figure is pair-plot of numeric independent features.

i. Variables 'temp' and 'atemp' are strongly correlated to each other.

ii. The relationship between 'hum' , 'windspeed' with target variable 'cnt' is less.

iii. We can remove features hum, windspeed and atemp/temp.(While feature importance process). Clearly, they won't help much with the prediction purpose.

Fig 2.3



2.2 Data Pre-Processing

2.2.1 Missing Value Analysis

Missing values in data is a common real world problem, we face while analysing and fitting the model.

The table below shows us that we don't have any missing value in our data.

Features	Counts
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

2.2.2 Outlier Analysis

This is one of the most important data pre-processing part which needs good understanding of the data and careful analysis. Outliers are the points which are far away from the other observations/or from their distribution.

We are going to observe outliers with the help of Box-Plot. While EDA we observed outliers in independent feature 'casual', boxplot of 'casual' in **(Fig 2.4)**.

After removing the outliers BoxPlot of feature casual looks as **Fig 2.5.**

We are almost losing 40 points from our data which is huge data w.r.t the size of the data. We will not treat outliers.

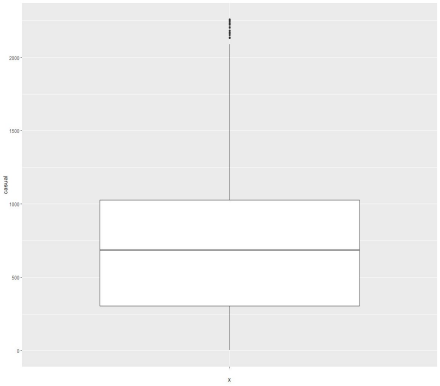


Fig 2.4

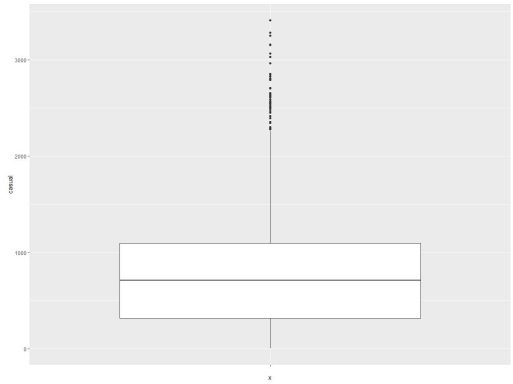


Fig 2.5

2.2.3 Feature Selection

When the number of features are very large. We can't visualize or create correlation heat map to observe, which features are important and which is not. In our case we have known that have only 14 features out of which 6 are continuous features(Relevant ones).

These are the two criteria, which we will use to select our features.

i. The relationship between two independent variables should be less and

ii. The relationship between Independent and Target variables should be high.

Below fig 2.6 illustrates that relationship between all numeric variables using Correlation heat map.

Figure 2.6 correlation heat map of numeric variables (R code in Appendix B)

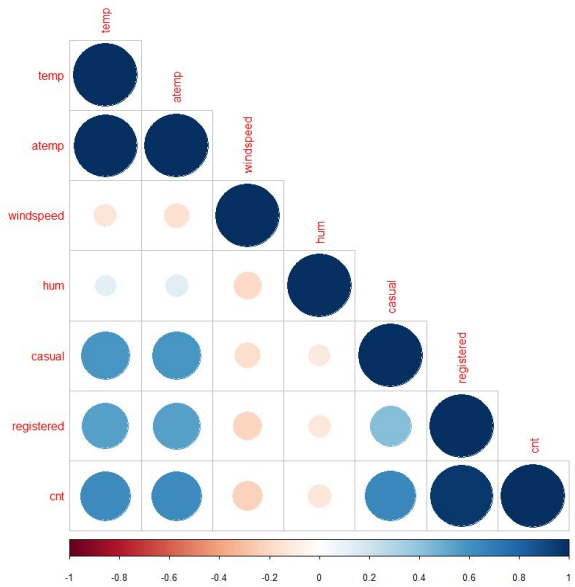


Fig 2.6

Color dark blue indicates there is strong positive relationship and if darkness is decreasing indicates relation between variables are decreasing.

Color dark Red indicates there is strong negative relationship and if darkness is decreasing indicates relationship between variables are decreasing.

Scale is given in x-axis.

i. It's clearly visible that 'hum' and 'windspeep' are not much correlated to target variable 'cnt'. So, we can remove this for our future analysis.

Ii. We have other techniques to detect importance of features using Random forest classifier.

2.2.4 Feature Scaling

Feature scaling is one of the most important pre-processing technique. It majorly involves two techniques named as Normalization and standardization.

Link to Random forest classifier based feature importance is given below,
https://towardsdatascience.com/running-random-forests-inspect-the-feature-importances-with-this-code-2b00dd72b92e

It's important to rescale features else it may lead to wrong predictions, especially in the case of regression problems.

Rescaling data between 0 and 1 is known as feature scaling. In our case independent features like 'temp', 'atemp', hum, 'windspeed' are already normalized. We need to normalize 'casual' and 'registered' features.

Code snippet:

```
# As we see casual and registered column needs scaling
col=['casual','registered']

for ithCol in col:
    Final_BikeData[ithCol]=(Final_BikeData[ithCol] - min(Final_BikeData[ithCol])
                      /(max(Final_BikeData[ithCol]) - min(Final_BikeData[ithCol]
```

2.3 Modeling

2.3.1 Model Selection

In our early stages of analysis during pre-processing we have come to understand that 'cnt' is highly dependent on 'temp', 'casual' and 'registered' features.

The dependent variable can fall in either of the four categories:
1. Nominal
2. Ordinal
3. Interval
4. Ratio

The dependent variable is Interval, so we have to do a Regression analysis.

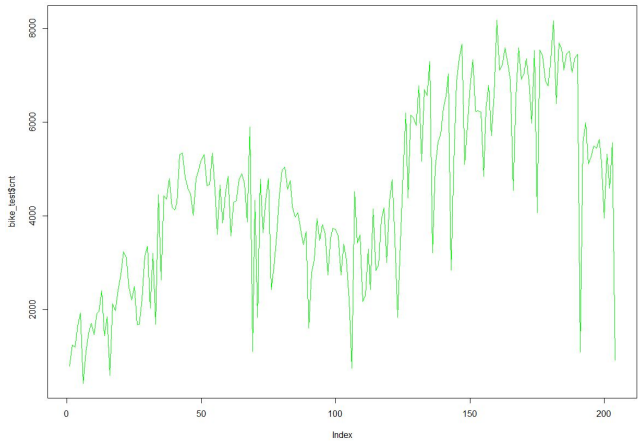
We will start our model building from the most simplest to more complex. Therefore we use Multiple Linear Regression at first.

2.3.2 Multiple Linear Regression

This is the most basic model in ML, it is good only if data is either linearly separable or data is lying on a line. In our case 'registered' feature carries exactly the same distribution as target feature 'cnt'. This might prove to be our best fit algorithm.

```
> model_lm=lm(cnt~ weathersit+temp+casual+registered,data = bike_train)
> summary(model_lm)

Call:
lm(formula = cnt ~ weathersit + temp + casual + registered, data = bike_train)

Residuals:
       Min         1Q     Median         3Q        Max
-8.212e-11 -3.500e-14  1.780e-13  3.990e-13  1.091e-11

Coefficients:
              Estimate Std. Error    t value Pr(>|t|)
(Intercept) -3.775e-12  8.321e-13 -4.537e+00 7.34e-06 ***
weathersit  -1.096e-12  3.556e-13 -3.081e+00  0.00219 **
temp         2.313e-12  1.392e-12  1.662e+00  0.09720 .
casual       1.000e+00  4.447e-16  2.249e+15  < 2e-16 ***
registered   1.000e+00  1.529e-16  6.538e+15  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.986e-12 on 446 degrees of freedom
Multiple R-squared:      1,     Adjusted R-squared:      1
F-statistic: 2.391e+31 on 4 and 446 DF,  p-value: < 2.2e-16
```

Adjusted R-squared value is 1, let's plot line graph fro predicted and actual values of train data set.



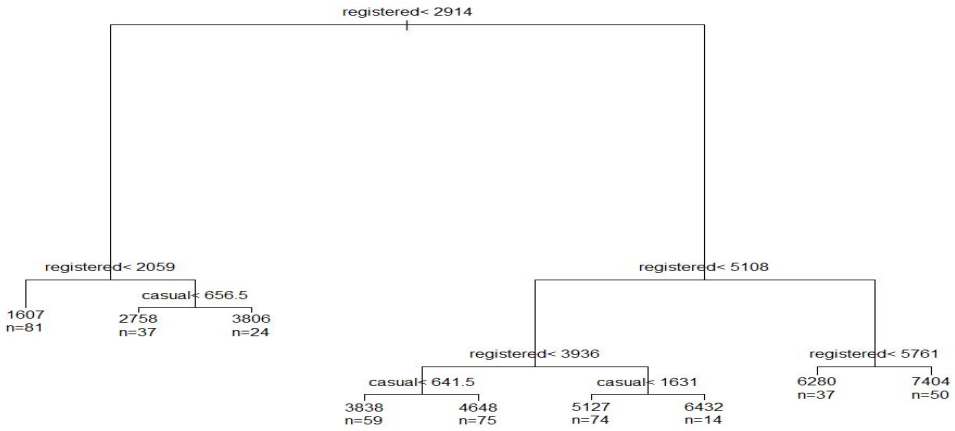
Green line represents actual value and blue predicted value. As we see both lines have overlapped each other. So prediction is quite good with linear regression model.

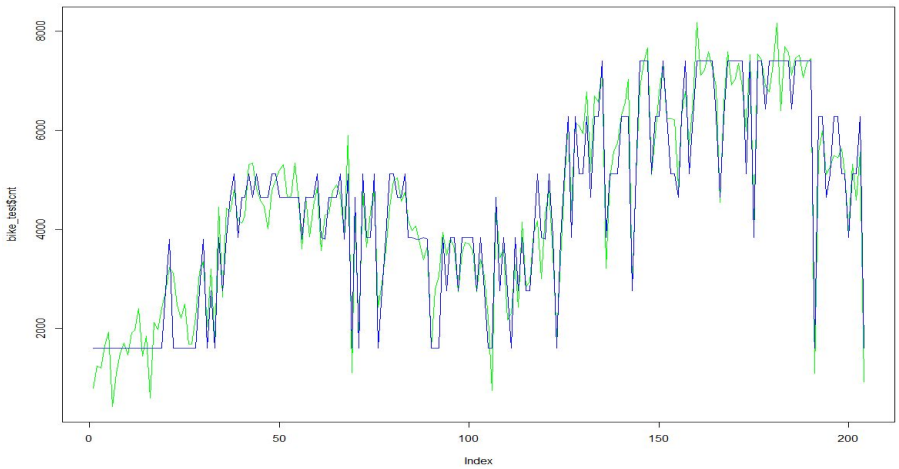
Error Metrics:

```
> RMSE(pred  = predic, obs = bike_test$cnt)
[1] 4.627063e-12
> MAPE(predic,bike_test$cnt)
[1] 1.436967e-15
>
```

2.3.3 Decision Tree

Decision tree to predict target variable 'cnt'.





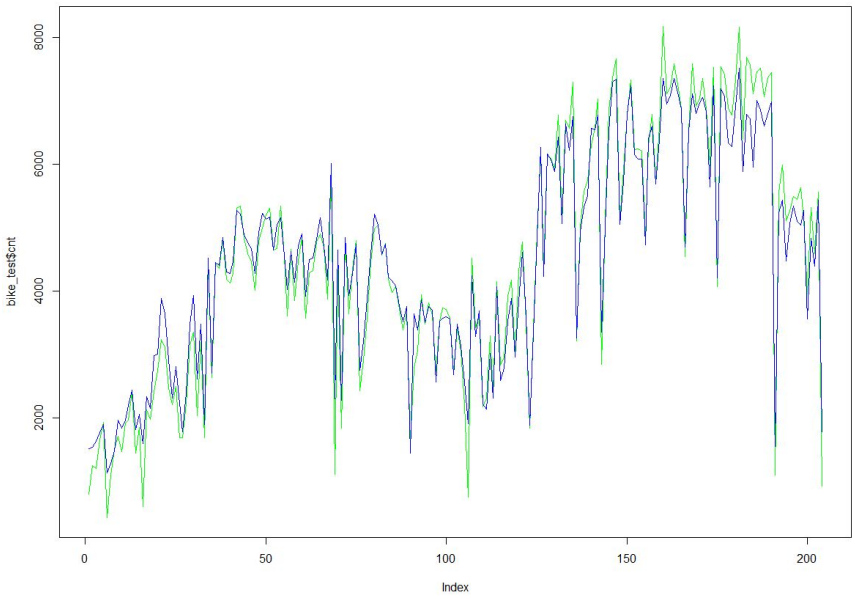
Green line represents actual and blue line predicted values.

Error Metrics:

```
> RMSE(pred  = predic, obs = bike_test$cnt)
[1] 485.812
> MAPE(predic,bike_test$cnt)
[1] 0.1345875
>
```

2.3.4 Random Forest

Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks, that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees.



Green line represents actual and blue line predicted values.

Error Metrics:

```
> RMSE(pred  = predic, obs = bike_test$cnt)
[1] 340.4017
> MAPE(predic,bike_test$cnt)
[1] 0.0974051
```

3. Conclusion

3.1 Model valuation:

a. Mean Absolute Percentage Error (MAPE) : MAPE is one of the error measures used to calculate the predictive performance of the model. We have applied this measure to our models that we have generated in the previous section.

$$M = \frac{1}{n}\sum_{t=1}^{n}\left|\frac{A_t - F_t}{A_t}\right|$$

Where, A(t) = actual value
F(t) = predicted value
n= total number of points

b. Root Mean Squared Error (RMSE) MSE can be obtained as follows

$$RMSE_{fo} = [\sum_{j=1}^{N}(z_{f_i} - z_{o_i})^2/N]^{1/2}$$

Where, z(f) = Forecasted value
z(o) = Original value
N= Total points

We have applied this measure to our models that we have generated in the previous section.

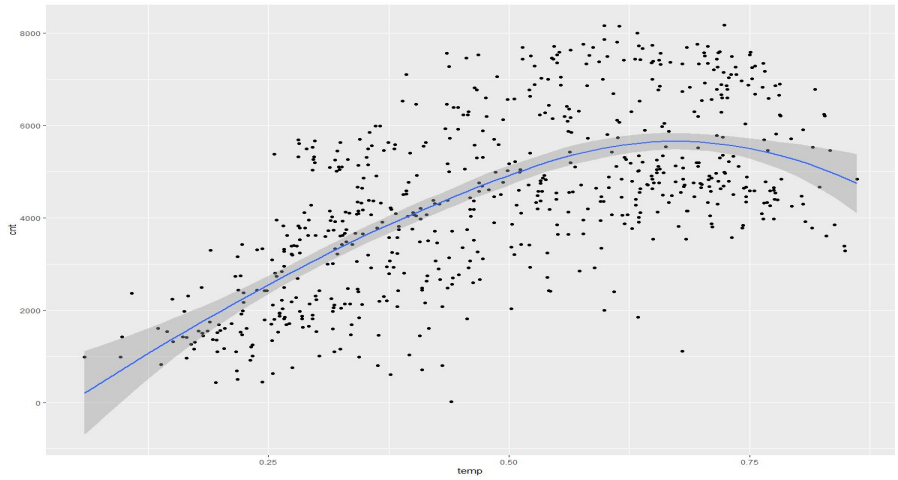
3.2 Model Selection:

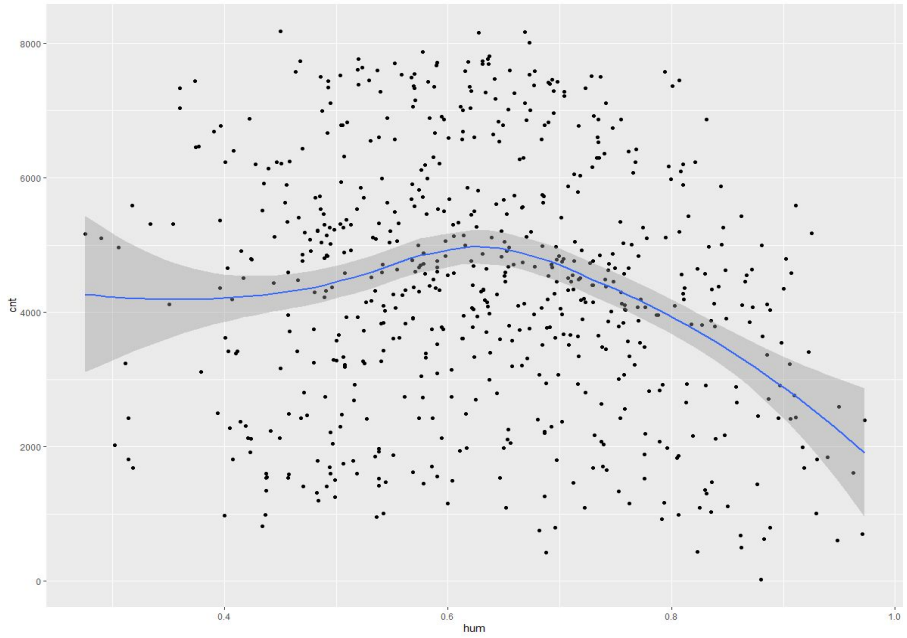
We have trained and applied three models i.e. Decision Tree, Random Forest and Linear Regression as MAPE and RMSE is less for the Linear Regression model compared to other models.

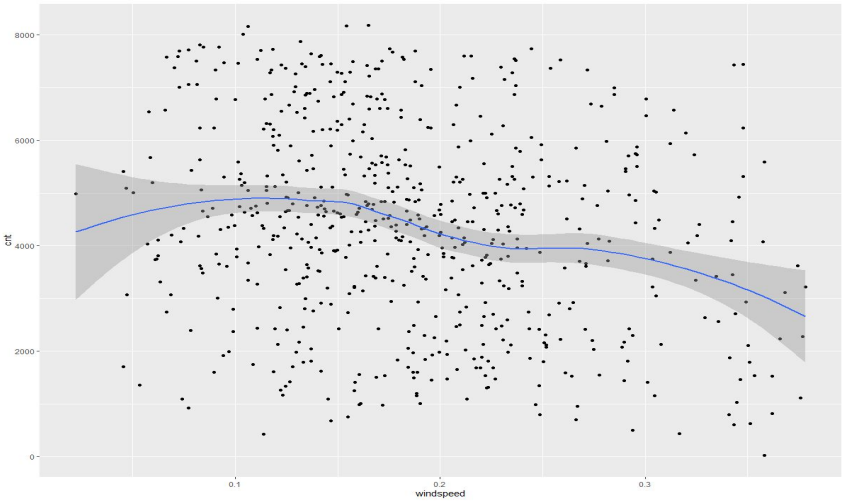
Conclusion: - For the Bike Rental Data, Linear Regression model gives the best fit line. So, we can deploy Linear Regression model and present it to others with confidence.

Appendix A- Extra Figures

Bivariate analysis plots w.r.t target variable ‘cnt’






8000
6000
4000
2000
0
cnt
0.1
0.2
0.3
windspeed

Appendix B (Complete R File)

Loading Libraries

```
rm(list=ls())

getwd()

# We need to predict cnt feature in the given data set
# cnt: total number of bikes rented on any perticular day

#Load Libraries
x = c("ggplot2", "corrplot", "caret", "randomForest", "dplyr", "tidyr", "MLmetrics", "rpart", 'DataCombine')

#install.packages(x)
lapply(x, require, character.only = TRUE)

warnings()
data_Bike<-read.csv("day.csv", header=T)
```

EDA

```
#*********************************************************
# EXPLORATORY DATA ANALYSIS
#*********************************************************

# view first 6 rows of the test data
head(data_Bike)

colnames(data_Bike)

#Get dimesion of the data
dim(data_Bike)
# It has 731 rows and 16 columns

str(data_Bike)
# As we see we don't need to change any data type for any feature/column

# Observations:
# As we can see that temp, atemp, hum, windspeed, casual and registered are continous vfeatures and rest are categorical features.
# Target variable is continous.
# Implies, it is an regression problem.
summary(data_Bike)

#Histogram plot for continouss features

#histogram plot for temp
hist(data_Bike$temp)

#histogram plot for atemp
hist(data_Bike$atemp)

#histogram plot for atemp
hist(data_Bike$hum)

#histogram plot for atemp
hist(data_Bike$windspeed)

#histogram plot for atemp
hist(data_Bike$casual)

# analyze the distribution of  target variable 'cnt'
univariate_numeric(data_Bike$cnt)

#histogram plot for atemp
```

Bi-Variate analysis

```
#Histogram plot for atemp
hist(data_Bike$registered)

#Observation: except casual and registered feature all other features are normally distributed and
#             scale is between 0 and 1. We have to scale features casual and humidity later, before applying models.

# Scatter plots of  cntns variables w.r.t target variable
ggplot(data_Bike, aes(x= temp,y=cnt)) +
  geom_point()+
  geom_smooth()   # temp is not highly correlated to cnt

ggplot(data_Bike, aes(x= atemp,y=cnt)) +
  geom_point()+
  geom_smooth()  # atemp is not highly correlated to cnt
                 # But by looking at the plot of temp and atemp, we can clearly see that both follows same curve w.r.t cnt as y-axis
ggplot(data_Bike, aes(x= hum,y=cnt)) +
  geom_point()+
  geom_smooth()   # Correlation between cnt and hum is very low

ggplot(data_Bike, aes(x= windspeed,y=cnt)) +
  geom_point()+
  geom_smooth()   # Correlation between cnt and windspeed is very low


ggplot(data_Bike, aes(x= casual,y=cnt)) +
  geom_point()+
  geom_smooth()   # Correlation between cnt and casual is very low


ggplot(data_Bike, aes(x= registered,y=cnt)) +
  geom_point()+
  geom_smooth()   # this plot is showing linear behaviour, this feature can be used for univariate analysis.
                  # cnt and registered feature are very highly correlated, and seems linear regression would be a better model if we
                  #  go for univariate analysis.
```

Missing Value Analysis and Outlier Analysis

```
#*************************************************************
# Missing Value Analysis
#*************************************************************

# Count total number of missing values

missing_value=data.frame(apply(data_Bike,2,function(input){sum(is.na(input))}))
missing_value$features=row.names(missing_value)
names(missing_value)[1]= "Missing_Percentage"
missing_value$Missing_Percentage=(missing_value$Missing_Percentage/nrow(data_Bike))
missing_value

# No missing values are present

#*************************************************************
# Outlier Analysis
#*************************************************************

# Checking which feature has outliers
ggplot(data = data_Bike, aes(x = "", y = temp)) +
  geom_boxplot() # No outliers

ggplot(data = data_Bike, aes(x = "", y = atemp)) +
  geom_boxplot() # No outliers

ggplot(data = data_Bike, aes(x = "", y = windspeed)) +
  geom_boxplot()   # It has outliers

ggplot(data = data_Bike, aes(x = "", y = hum)) +
  geom_boxplot() # Outliers are present

ggplot(data = data_Bike, aes(x = "", y = casual)) +
  geom_boxplot() # Outliers are present

ggplot(data = data_Bike, aes(x = "", y = registered)) +
  geom_boxplot() # No outliers

# x[!x %in% boxplot.stats(x)$out], by this formula we can directly remove the outliers from the desired feature.

cnames=colnames(data_Bike)
```

Treating Outliers and Feature Selection

```
cnames=colnames(data_Bike)

 #loop to remove outliers from all variables
 for(i in cnames){
   print(i)
   outliers = data_Bike[,i][data_Bike[,i] %in% boxplot.stats(data_Bike[,i])$out]
   print(length(outliers))
   data_Bike = data_Bike[which(!data_Bike[,i] %in% outliers),]
 }

#Outliers have been removed
ggplot(data = data_Bike, aes(x = "", y = casual)) +
  geom_boxplot() # Outliers are present

dim(data_Bike)
# After removing outliers we are left out with 655 rows out of 731 rows. Which might be a problem, because of good amount
# of data loss


#*****************************************************************
# Feature selection
#*****************************************************************

# Correlation graph
data_New=subset(data_Bike,select = c("temp","atemp","windspeed","hum","casual","registered","cnt"))

model_rf= randomForest(cnt ~ ., data = data_Bike, ntree = 100, keep.forest = FALSE, importance = TRUE)


cr=cor(data_New)
corrplot(cr,type = "lower")
# As we can observe that, our target variable is very less dependent on feature hum and windspeed, so we can remove both
# And also temp and atemp are highly correlated to each other, it's better to remove one of them. I will remove atemp.

# Removal of features
data_Bike = subset(data_Bike,select=-c(dteday,atemp,hum, windspeed))
dim(data_Bike)

colnames(data_Bike)
```

Feature Scaling

```
#*********************************************************************
# Feature Normalization
#*********************************************************************

# As we observed above, casual and registered feature requires normalization

cnames=c("casual","registered")

for(iIterCol in cnames){
  print(iIterCol)
  data_Bike[,iIterCol] = (data_Bike[,iIterCol] - min(data_Bike[,iIterCol]))/
    (max(data_Bike[,iIterCol] - min(data_Bike[,iIterCol])))
}

ggplot(data = data_Bike, aes(x = "", y = casual)) +
  geom_boxplot()
# As we can see casual and registered feature has been normalized

#*************************************************************************************
#  Sampling
#*************************************************************************************

#Clean the environment
rmExcept("data_Bike")

#Dividing the data into train and test using stratified sampling method.
set.seed(3)
id = sample(2,nrow(data_Bike), prob = c(0.7,0.3),replace=TRUE)
bike_train = data_Bike[id==1,]
bike_test  = data_Bike[id==2,]

dim(bike_train)
dim(bike_test)
# Data has been divided correctly
```

Linear Regression and Decision Tree

```
##### Linear Regression

model_lm=lm(cnt~ weathersit+temp+casual+registered,data = bike_train)
summary(model_lm)
# R squared and adjusted R-squared error is 1, it might be possible that our model is over fitting.
# p-value is 2.2e^-16, which is a good sign

na.omit(model_lm)
predic=predict(model_lm,bike_test)

plot(bike_test$cnt,type='l',col="green")
lines(predic,type="l",col="blue")

# Errors
RMSE(pred  = predic, obs = bike_test$cnt)
MAPE(predic,bike_test$cnt)
# Root mean square error is 4.627063e-12
# MAPE is 1.436967e-15
# Let's visualize actual and predicted values, both are over lapping. Implies prediction is without any error.
# Train error and test error both are almost 0.
# This might be the best model, as we have observed while visualizing as well that linear model might be the best fit model.


##### Decision Tree

#Training the model
model_dt=rpart(cnt~workingday+mnth+holiday+weekday +weathersit+temp+casual+registered,data = bike_train,control = list(minsplit = 10, maxdepth = 20, cp = 0.
plot(model_dt,margin=0.1)
# We can tune our model by changing hyper parameters i.e. depth and splits
text(model_dt,use.n = TRUE,pretty=TRUE)#control = list(minsplit = 10, maxdepth = 10, cp = 0.01))

predic <- predict(model_dt, newdata = bike_test)

plot(bike_test$cnt,type='l',col="green")
lines(predic,type="l",col="blue")
# After plotting predicted values and actual values, we can see the error please.

RMSE(pred  = predic, obs = bike_test$cnt)
MAPE(predic,bike_test$cnt)
# Root mean square error is 485.812
# MAPE is 0.1345875
```

Random Forest and Final Observation

```
##### Random Forest

model_rf=randomForest(cnt~workingday+mnth+holiday+weekday +weathersit+temp+casual+registered,data = bike_train,control = list(minsplit = 10, maxdepth = 20,
plot(model_rf,margin=0.1)
# We can tune our model by changing hyper parameters i.e. depth and splits

predic=predict(model_rf,newdata = bike_test,type="class")

# Root mean square error is  334.4366
plot(bike_test$cnt,type='l',col="green")
lines(predic,type="l",col="blue")

RMSE(pred  = predic, obs = bike_test$cnt)
MAPE(predic,bike_test$cnt)
# Root mean square error is  341.242
# MAPE is 0.09948815

# Observations:
# i. Linear regression alg is giving us the best result compare to other Algo's.
# ii. It was visible while exploring the data, since 'registered' feature was linearly dependent on target feature 'cnt'.
#      It was highly observable, that Multiple linear regression is going to be our best fit algorthm.
```