Bike Renting Prediction

Harish Ravi

5 January 2019

Contents

Chapter 1

Introduction

1.1 Problem Statement

The aim of the project is to predict the bike rental count on daily basis based on the environmental and seasonal settings. We would like to predict the bike rental count based on previous data.

1.2 Data

Our task is to build regression models which will predict the bike rental count based on previous environmental and seasonal data. Given below is a sample of the data set that we are using to predict the Bike rental count.

Table 1.1: Bike Rental Sample Data (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
6	1/6/2011	1	0	1	0	4	1
7	1/7/2011	1	0	1	0	5	1

Table 1.2: Bike Rental Sample Data (Columns: 9-16)

weathersit	temp	atemp	hum	windspeed	casual	registered	cnt
2	0.344167	0.363625	0.805833	0.160446	331	654	985
2	0.363478	0.353739	0.696087	0.248539	131	670	801
1	0.196364	0.189405	0.437273	0.248309	120	1229	1349
1	0.2	0.212122	0.590435	0.160296	108	1454	1562
1	0.226957	0.22927	0.436957	0.1869	82	1518	1600
1	0.204348	0.233209	0.518261	0.089565	88	1518	1606
2	0.196522	0.208839	0.498696	0.168726	148	1362	1510

The Explanation of Independent variables is as follows

Predictor	Description
dteday	Date in "mm/dd/yyyy " format
season	Four categories-> 1 = spring, 2 = summer, 3 = fall, 4 = winter
yr	For 2011 it is 0 and for 2012 it is 1
holiday	whether the day is a holiday or not (1/0)
workingday	whether the day is working day or holiday (1/0)
temp	Temperature in Celsius
atemp	Average temperature in Celsius
hum	Relative humidity
weathersit	Four Categories of weather 1-> Clear, Few clouds, Partly cloudy, Partly cloudy 2-> Mist + Cloudy, Mist + Broken clouds, Mist + Few clouds, Mist 3-> Light Snow and Rain + Thunderstorm + Scattered clouds, Light Rain + Scattered Clouds 4-> Heavy Rain + Ice Pallets + Thunderstorm + Mist, Snow + Fog
windspeed	Wind speed

As we can see in the table below we have the following 12 variables, using which we have to correctly predict the Bike Rental Count

Table 1.3 Predictor Variables

S.No	Predictor
1	dteday
2	season
3	yr
4	mnth
5	holiday
6	weekday
7	workingday
8	temp
9	atemp
10	hum
11	weathersit
12	windspeed

Chapter 2

Methodology

2.1 Pre Processing

Any predictive modeling requires 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 probability distributions of the variables. Most analysis like regression, require the data to be normally distributed. We can visualize that in a glance by looking at the probability distributions or probability density functions of the variable.

In Figure 2.1 we have plotted the histogram of the variables available in the data. So we can see from the histogram that the distributions of the variables.

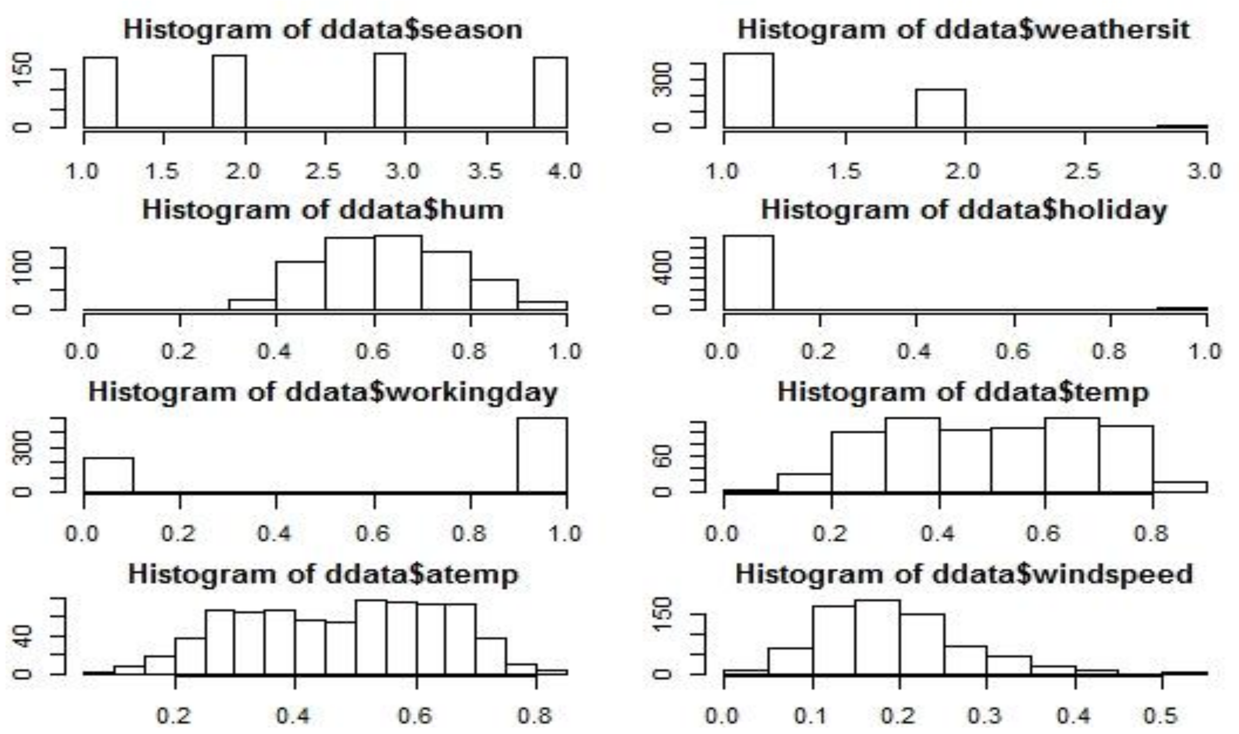


Fig 2.1

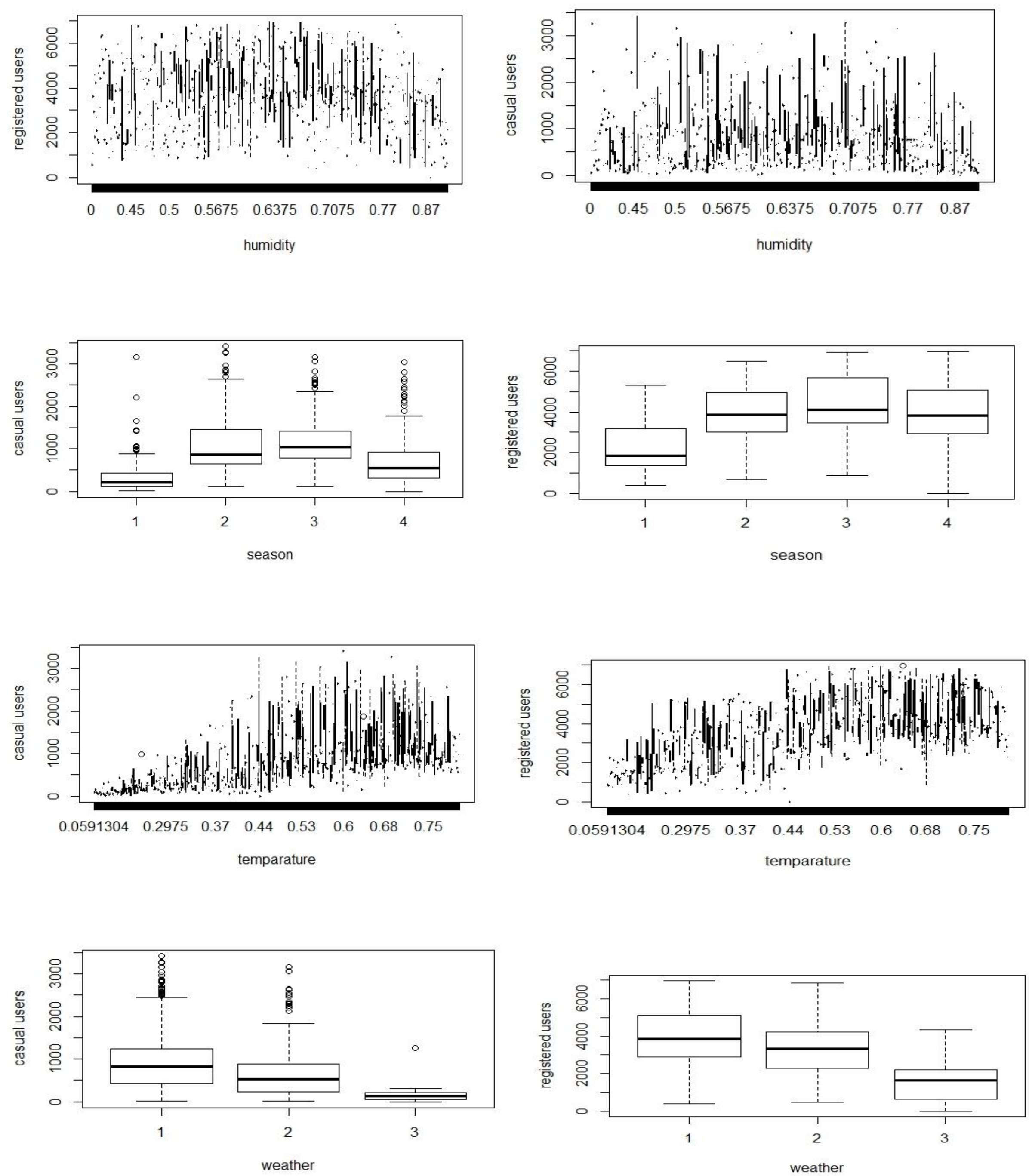
From above Histogram we can understand that

- Season has four categories of almost equal distribution
- Weather 1 has higher contribution i.e. mostly clear weather

2.1.1 Outlier Analysis

One of the other steps of **pre-processing** apart from checking for normality is the presence of outliers. In this case we use a classic approach of removing outliers. We visualize the outliers using *boxplots*.

In figure 2.2 we have plotted the boxplots of the predictor variables with respect to registered and casual users. A lot of useful inferences can be made from these plots. First as you can see, we have a lot of outliers and extreme values in each of the data set.



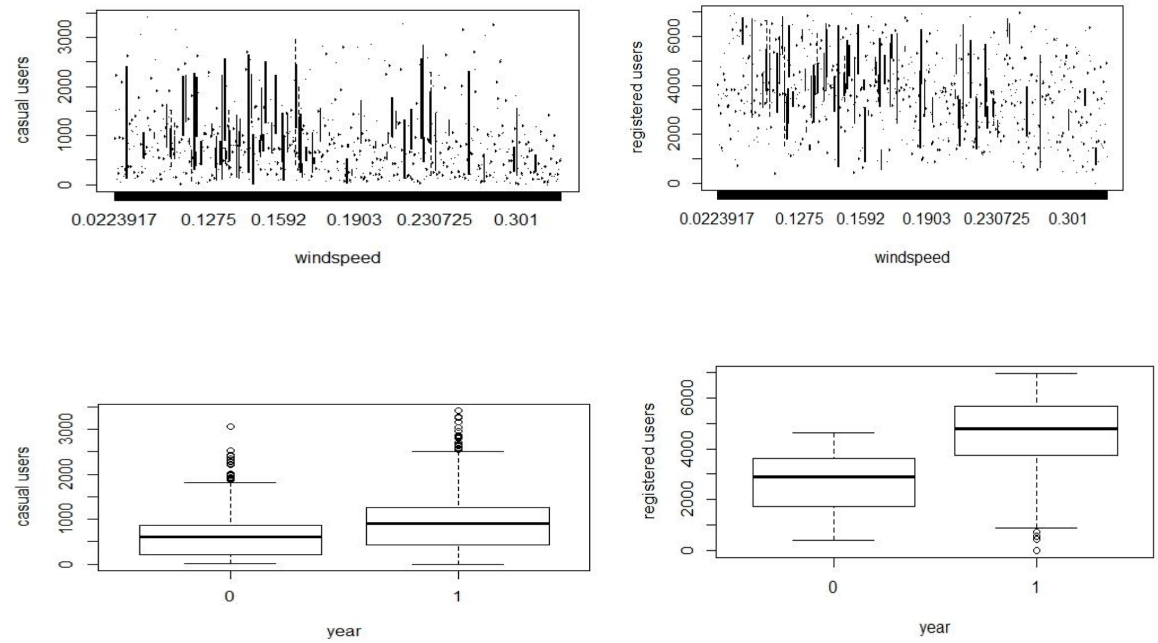


Fig 2.2

From the box plots we can see that there are outliers in casual and registered users with respect to independent variables.

Since the users are not normally or functionally distributed there will be outliers, so there is no need to take any processing action on these outliers in this case.

2.1.2 Correlation Analysis

Another step of Exploratory Data Analysis is to look for highly correlated variables in the data. A very simple way of looking at correlations in the data is plotting Heat map between variables.

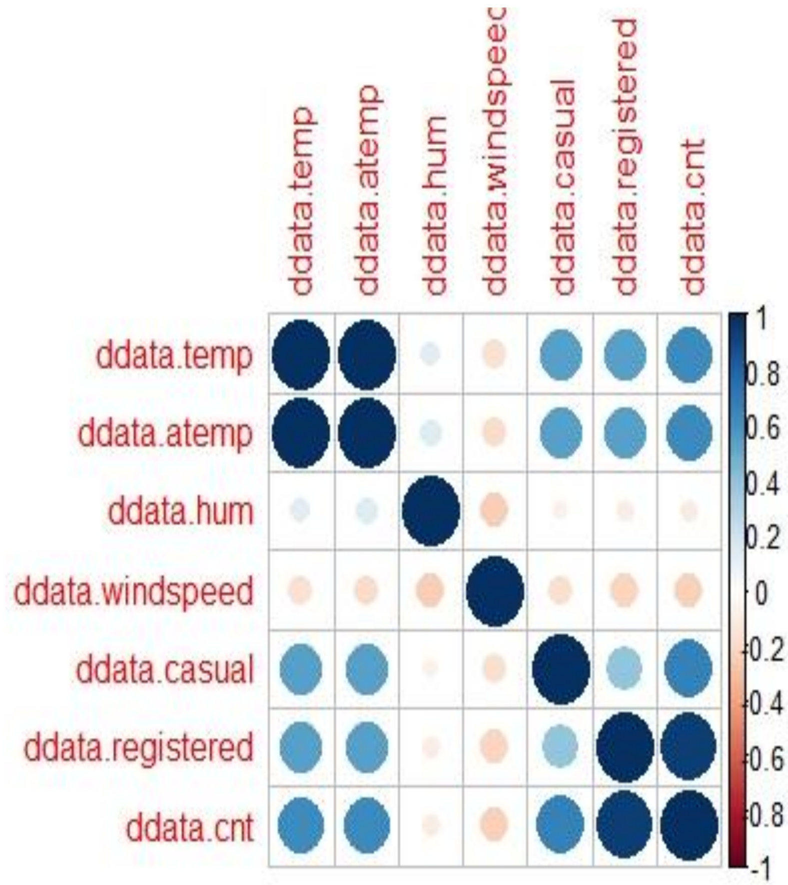


Fig 2.3

From the Heat map we can see that there is higher correlation between temperature and Average temperature(0.8). So we will consider any one of the variable while modeling.

2.1.3 Random Tree plotting

Before doing feature selection lets plot random trees for extracting new variables and binning some existing variables.

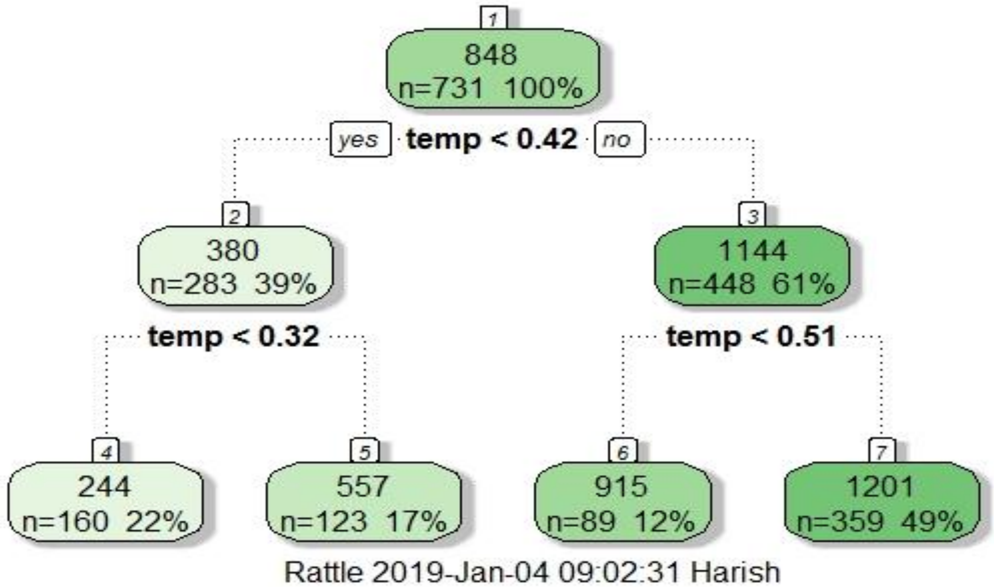


Fig 2.4

Decision tree of Casual Users with respect to temperature

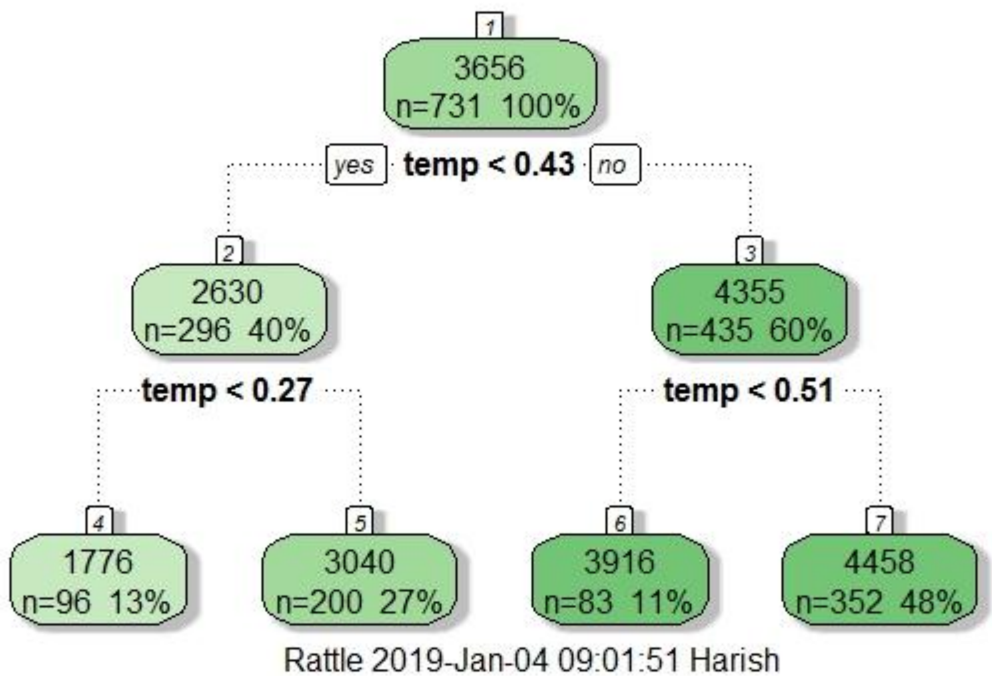


Fig 2.5

Decision tree of Registered Users with respect to temperature

2.1.4 Feature Selection

Before performing any type of modeling we need to assess the importance of each predictor variable in our analysis. There is a possibility that many variables in our analysis are not important at all to the problem. There are several methods of doing that. We have used the above *Random Forests* to perform features selection.

We will create the following variables which will play an important role while predicting.

Temp Bins: We have created bins for temperature for both registered and casuals users. Variables created are (temp_reg and temp_cas).

Year Bins: We had a hypothesis that bike demand will increase over time and we have proved it also. Here I have created 8 bins (quarterly) for two years. Jan-Mar 2011 as 1 …..Oct-Dec2012 as 8.

Day Type: Created a variable having categories like “weekday”, “weekend” and “holiday”.

Weekend: Created a variable for weekend (0/1)

2.2 Modeling

2.2.1 Model Selection

The dependent variable, in our case *Count*, is Continuous Value the only predictive analysis that we can perform is **Regression**.

You always start your model building from the simplest to the complex. Therefore we use Linear Regression.

As we know that dependent variables have natural outliers so we will predict log of dependent variables.

Predict bike demand of registered and casual users separately.

Here we have added 1 to deal with zero values in the casual and registered columns.

2.2.2 Linear Regression

Linear Regression model for predicting the Casual users Count

```
Call:
lm(formula = logcasual ~ workingday + holiday + day_type + hum +
    atemp + windspeed + season + weathersit + weekend + yr +
    year_part + mnth + day_type, data = dtrain)

Residuals:
     Min       1Q   Median       3Q      Max
-1.67505 -0.18565  0.02247  0.21270  1.54767

Coefficients: (10 not defined because of singularities)
                    Estimate Std. Error t value Pr(>|t|)
(Intercept)          5.27627    0.13230  39.881  < 2e-16 ***
workingday1         -0.91137    0.03555 -25.635  < 2e-16 ***
holiday1            -0.27927    0.09933  -2.811 0.005144 **
day_typeweekend           NA         NA      NA       NA
day_typeworking day       NA         NA      NA       NA
hum                 -0.81950    0.16179  -5.065 5.93e-07 ***
atemp                2.82324    0.24714  11.424  < 2e-16 ***
windspeed           -1.09561    0.23187  -4.725 3.07e-06 ***
season2              0.98842    0.12733   7.763 5.50e-14 ***
season3              0.99376    0.12021   8.267 1.51e-15 ***
season4              0.70029    0.09319   7.514 3.04e-13 ***
weathersit2         -0.17625    0.04153  -4.244 2.66e-05 ***
weathersit3         -1.13369    0.11308 -10.025  < 2e-16 ***
weekend1                  NA         NA      NA       NA
yr1                  0.53891    0.06511   8.277 1.40e-15 ***
year_part2           0.15577    0.09160   1.700 0.089722 .
year_part3           0.33782    0.09107   3.710 0.000233 ***
year_part4           0.09203    0.09175   1.003 0.316356

mnth2                0.26802    0.07920   3.384 0.000776 ***
mnth3                0.98023    0.08583  11.421  < 2e-16 ***
mnth4                0.21307    0.08891   2.396 0.016962 *
mnth5                0.24370    0.08234   2.960 0.003240 **
```

```
mnth6                     NA         NA      NA       NA    
mnth7               -0.26592    0.08501  -3.128 0.001872 ** 
mnth8               -0.10582    0.08170  -1.295 0.195915    
mnth9                     NA         NA      NA       NA    
mnth10               0.58227    0.08660   6.724 5.30e-11 ***
mnth11               0.35381    0.07939   4.457 1.05e-05 ***
mnth12                    NA         NA      NA       NA    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.3493 on 457 degrees of freedom
Multiple R-squared:  0.8792,   Adjusted R-squared:  0.8734 
F-statistic: 151.2 on 22 and 457 DF,  p-value: < 2.2e-16
```

Linear Regression for predicting the Count of Registered Users

```
Call:
lm(formula = logregister ~ workingday + holiday + day_type + 
    hum + atemp + windspeed + season + weathersit + weekend + 
    yr + year_part + mnth + day_type, data = dtrain)

Residuals:
     Min       1Q   Median       3Q      Max 
-0.84765 -0.06573  0.01477  0.09278  0.50770 

Coefficients: (10 not defined because of singularities)
                    Estimate Std. Error t value Pr(>|t|)    
(Intercept)          7.02639    0.05819 120.747  < 2e-16 ***
workingday1          0.28448    0.01564  18.193  < 2e-16 ***
holiday1             0.04275    0.04369   0.978  0.32839    
day_typeweekend           NA         NA      NA       NA    
day_typeworking day       NA         NA      NA       NA    
hum                 -0.34142    0.07116  -4.798 2.18e-06 ***
atemp                0.73278    0.10870   6.741 4.75e-11 ***
windspeed           -0.55057    0.10199  -5.398 1.08e-07 ***
season2              0.45988    0.05600   8.211 2.26e-15 ***
season3              0.49076    0.05287   9.282  < 2e-16 ***
season4              0.51862    0.04099  12.652  < 2e-16 ***
weathersit2         -0.10392    0.01827  -5.689 2.28e-08 ***
weathersit3         -0.68230    0.04974 -13.718  < 2e-16 ***
weekend1                  NA         NA      NA       NA    
yr1                  0.81354    0.02864  28.409  < 2e-16 ***
year_part2           0.34381    0.04029   8.533  < 2e-16 ***
year_part3           0.41366    0.04006  10.327  < 2e-16 ***
year_part4           0.37119    0.04036   9.198  < 2e-16 ***
mnth2                0.17920    0.03484   5.144 4.00e-07 ***
mnth3                0.27483    0.03775   7.280 1.47e-12 ***
mnth4               -0.20180    0.03911  -5.160 3.69e-07 ***
mnth5                0.02353    0.03622   0.650  0.51626    
```

```
mnth6                   NA         NA        NA        NA    
mnth7             -0.18116    0.03739  -4.845 1.74e-06 ***
mnth8             -0.11827    0.03594  -3.291  0.00107 ** 
mnth9                   NA         NA        NA        NA    
mnth10             0.07416    0.03809   1.947  0.05216 .  
mnth11             0.02222    0.03492   0.636  0.52478    
mnth12                  NA         NA        NA        NA    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.1536 on 457 degrees of freedom
Multiple R-squared:  0.9067,	Adjusted R-squared:  0.9022 
F-statistic: 201.9 on 22 and 457 DF,  p-value: < 2.2e-16
```

We have built the linear regression models for both casual and registered users and the summary of the models is also listed above.

2.2.3 Random Forest

Now we will try and use another regression model to predict our *Count* target variable. We will use Random Forest to predict the values of our target variable.

```
set.seed(415)
rf_register = randomForest(logregister ~
workingday+holiday+day_type+hum+atemp+windspeed+season+weathersit+weekend+yr+year_part+mnt
h+day_type, data=dtrain,importance=TRUE, ntree=250)
```

```
                Length Class  Mode
call              5    -none- call
type              1    -none- character
predicted       480    -none- numeric
mse             250    -none- numeric
rsq             250    -none- numeric
oob.times       480    -none- numeric
importance       24    -none- numeric
importanceSD     12    -none- numeric
localImportance   0    -none- NULL
proximity         0    -none- NULL
ntree             1    -none- numeric
mtry              1    -none- numeric
forest           11    -none- list
coefs             0    -none- NULL
y               480    -none- numeric
test              0    -none- NULL
inbag             0    -none- NULL
terms             3    terms  call
```

```
set.seed(415)
rf_casual   = randomForest(logcasual ~
workingday+holiday+day_type+hum+atemp+windspeed+season+weathersit+weekend+yr+year_part+mnt
h+day_type, data=dtrain,importance=TRUE, ntree=250)
```

```
                Length Class  Mode
call              5    -none- call
type              1    -none- character
predicted       480    -none- numeric
mse             250    -none- numeric
rsq             250    -none- numeric
oob.times       480    -none- numeric
importance       24    -none- numeric
importanceSD     12    -none- numeric
localImportance   0    -none- NULL
proximity         0    -none- NULL
ntree             1    -none- numeric
mtry              1    -none- numeric
forest           11    -none- list
coefs             0    -none- NULL
y               480    -none- numeric
test              0    -none- NULL
inbag             0    -none- NULL
terms             3    terms  call
```

Chapter 3

Conclusion

3.1 Model Evaluation

Now that we have a few models for predicting the target variable, we need to decide which one to choose. There are several criteria that exist for evaluating and comparing models. We can compare the models using any of the following criteria:

1. Predictive Performance
2. Interpretability
3. Computational Efficiency

In our case of Bike Rental. We will use *Predictive performance* as the criteria to compare and evaluate models. Predictive performance can be measured by comparing Predictions of the models with real values of the target variables, and calculating some average error measure.

3.1.1 Mean Absolute Percentage Error Loss (MAPE)

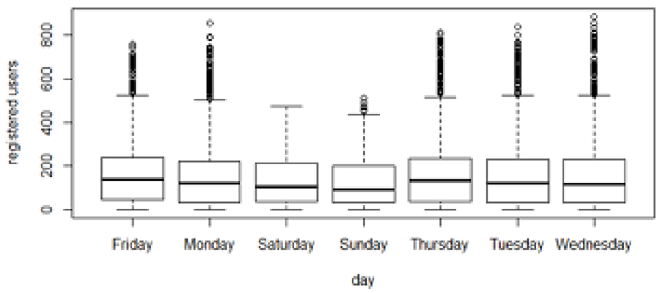
MAPE is one of the error measures used to calculate the predictive performance of the model. We will apply this measure to our models that we have generated in the previous section.

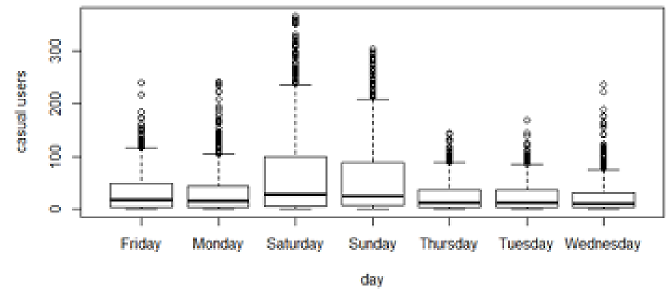
```
Prediction_lmreg = predict(lm_register,dtest)
dtest$logregister= Prediction_lmreg

Prediction_lmcas = predict(lm_casual,dtest)
dtest$logcasual  = Prediction_lmcas

dtest$pregistered =exp(dtest$logregister)-1
dtest$pcasual     =exp(dtest$logcasual)-1

MAPE(dtest$pregistered,dtest$registered)

errorrate = 7.9
accuracy = 92.1

MAPE(dtest$pcasual,dtest$casual)

errorrate = 8.4
accuracy = 91.6
```

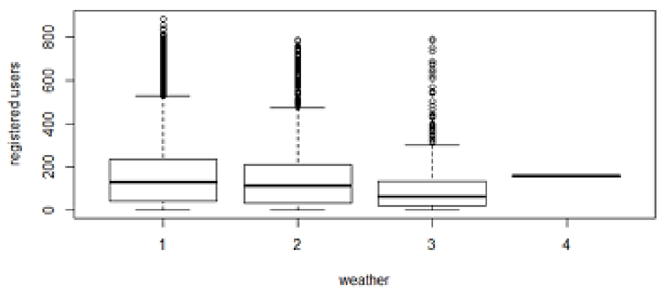
3.2 Model Selection

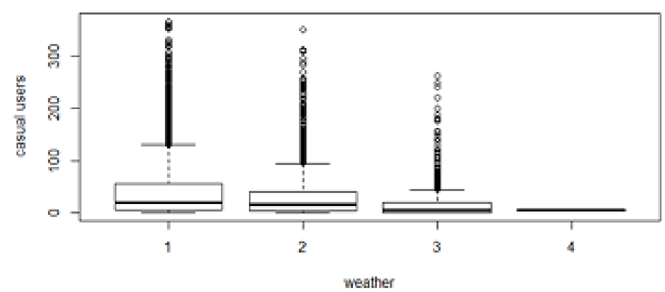
We can see that both models perform comparatively on average and therefore we can select either of the two models without any loss of information.

Appendix A - Extra Figures









Appendix B - R Code

Bike Rental Histograms (Fig: 2.1)

```
par(mfrow=c(4,2))
par(mar = rep(2, 4))
hist(ddata$season)
hist(ddata$weathersit)
hist(ddata$hum)
hist(ddata$holiday)
hist(ddata$workingday)
hist(ddata$temp)
hist(ddata$atemp)
hist(ddata$windspeed)
```

Bike Rental Boxplots(Fig 2.2)

```
boxplot(ddata$registered~ddata$season,xlab="season", ylab="registered users")
boxplot(ddata$casual~ddata$season,xlab="season", ylab="casual users")

boxplot(ddata$registered~ddata$weathersit,xlab="weather", ylab="registered users")
boxplot(ddata$casual~ddata$weathersit,xlab="weather", ylab="casual users")

boxplot(ddata$registered~ddata$temp,xlab="temparature", ylab="registered users")
boxplot(ddata$casual~ddata$temp,xlab="temparature", ylab="casual users")

boxplot(ddata$registered~ddata$yr,xlab="year", ylab="registered users")
boxplot(ddata$casual~ddata$yr,xlab="year", ylab="casual users")

boxplot(ddata$registered~ddata$windspeed,xlab="windspeed", ylab="registered users")
boxplot(ddata$casual~ddata$windspeed,xlab="windspeed", ylab="casual users")

boxplot(ddata$registered~ddata$hum,xlab="humidity", ylab="registered users")
boxplot(ddata$casual~ddata$hum,xlab="humidity", ylab="casual users")
```

Correlation Map (Fig 2.3)

```
d = data.frame( ddata$temp,ddata$atemp,ddata$hum,ddata$windspeed,
          ddata$casual,ddata$registered,ddata$cnt)
M = cor(d)
corrplot(M, method = "circle")
```

Complete R File

```
#Clearing RAM
rm(list = ls())

#importing all the required libraries
library(rpart)
library(corrplot)
library(rpart.plot)
library(RColorBrewer)
library(rattle)
library(tidyverse)
library(modelr)
library(broom)
library(MLmetrics)
library(randomForest)
library(ggplot2)
library(caret)
library(e1071)

#Knowing the working directory
getwd()

#setting the working directory
setwd("C:/Users/Harish/Desktop/Projects")

#Importing the data
ddata  = read.csv("day.csv",sep = ",")

#Understanding the data or summary of the day data

head(ddata,5)
summary(ddata)
View(ddata)

#Knowing the data type of the variables
str(ddata)

#Data pre processing
#Cheking for any missing values in the dataset
sum(is.na(ddata))

#No missing values in our data, so no need of Imputing
#Understanding the Distribution of numeric variables by plotting Histogram

par(mfrow=c(4,2))
par(mar = rep(2, 4))
hist(ddata$season)
hist(ddata$weathersit)
hist(ddata$hum)
hist(ddata$holiday)
hist(ddata$workingday)
hist(ddata$temp)
hist(ddata$atemp)
hist(ddata$windspeed)
```

```
#outlier analysis
#Boxplotting count againest required variables to know the outliers

boxplot(ddata$registered~ddata$season,xlab="season", ylab="registered users")
boxplot(ddata$casual~ddata$season,xlab="season", ylab="casual users")

boxplot(ddata$registered~ddata$weathersit,xlab="weather", ylab="registered users")
boxplot(ddata$casual~ddata$weathersit,xlab="weather", ylab="casual users")

boxplot(ddata$registered~ddata$temp,xlab="temparature", ylab="registered users")
boxplot(ddata$casual~ddata$temp,xlab="temparature", ylab="casual users")

boxplot(ddata$registered~ddata$yr,xlab="year", ylab="registered users")
boxplot(ddata$casual~ddata$yr,xlab="year", ylab="casual users")

boxplot(ddata$registered~ddata$windspeed,xlab="windspeed", ylab="registered users")
boxplot(ddata$casual~ddata$windspeed,xlab="windspeed", ylab="casual users")

boxplot(ddata$registered~ddata$hum,xlab="humidity", ylab="registered users")
boxplot(ddata$casual~ddata$hum,xlab="humidity", ylab="casual users")

#From the Boxplots we can see there are outliers,
#since the users are not normally or functionally distributed
#There is no need to take any action on these outliers

#Correlation Analysis
#Plotting a correlation heatmap

d = data.frame( ddata$temp,ddata$atemp,ddata$hum,ddata$windspeed,
          ddata$casual,ddata$registered,ddata$cnt)
M = cor(d)
corrplot(M, method = "circle")

#from the correlation plot it is clear that there is high correlation
#betwwen temp and atemp, so we can use any one while building the model

#plotting the random tree partitioning with data of users against temparature

f=rpart(registered~t,ddata)
fancyRpartPlot(f)

f1=rpart(casual~temp,ddata)
fancyRpartPlot(f1)

#Applying feature selection
#Most of the available data is processed, let us create new
#Variables for better data feeding
#Feature Engineering the data to create some more meaningful columns

ddata$weekend=0
ddata$weekend[ddata$weekday== 0 | ddata$weekday== 6 ]=1

ddata$temp_reg=0
ddata$temp_reg[ddata$temp<13]=1
ddata$temp_reg[ddata$temp>=13 & ddata$temp<23]=2
ddata$temp_reg[ddata$temp>=23 & ddata$temp<30]=3
ddata$temp_reg[ddata$temp>=30]=4
```

```
ddata$temp_cas=0
ddata$temp_cas[ddata$temp<15]=1
ddata$temp_cas[ddata$temp>=15 & ddata$temp<23]=2
ddata$temp_cas[ddata$temp>=23 & ddata$temp<30]=3
ddata$temp_cas[ddata$temp>=30]=4


ddata$year_part[ddata$yr== 0 ]=1
ddata$year_part[ddata$yr== 0 & ddata$mnth>3]=2
ddata$year_part[ddata$yr== 0 & ddata$mnth>6]=3
ddata$year_part[ddata$yr== 0 & ddata$mnth>9]=4
ddata$year_part[ddata$yr== 1]=5
ddata$year_part[ddata$yr== 1 & ddata$mnth>3]=6
ddata$year_part[ddata$yr== 1 & ddata$mnth>6]=7
ddata$year_part[ddata$yr== 1 & ddata$mnth>9]=8
table(ddata$year_part)

ddata$day_type=0
ddata$day_type[ddata$holiday==0 & ddata$workingday==0]="weekend"
ddata$day_type[ddata$holiday==1]="holiday"
ddata$day_type[ddata$holiday==0 & ddata$workingday==1]="working day"
table(ddata$day_type)

#knowing the datatype of all available variables
str(ddata)

#Converting the variables to the required data format for feeding to model
ddata$season        =as.factor(ddata$season)
ddata$yr            =as.factor(ddata$yr)
ddata$weekday       =as.factor(ddata$weekday)
ddata$holiday       =as.factor(ddata$holiday)
ddata$workingday    =as.factor(ddata$workingday)
ddata$weekend       =as.factor(ddata$weekend)
ddata$weathersit    =as.factor(ddata$weathersit)
ddata$temp_cas      =as.factor(ddata$temp_cas)
ddata$temp_reg      =as.factor(ddata$temp_reg)
ddata$mnth          =as.factor(ddata$mnth)
ddata$day_type      =as.factor(ddata$day_type)
ddata$dteday        =as.factor(ddata$dteday)
ddata$year_part     =as.factor(ddata$year_part)

#data pre processing completed
#Model building to predict the registered and casual users
#dividing the train and test data

dtrain =ddata[as.integer(substr(ddata$dteday,9,10))<21,]
dtest  =ddata[as.integer(substr(ddata$dteday,9,10))>20,]


#registered and casual users count is not normally distributed
#applying log transformation to these skewed variables in the data

dtrain$logcasual   =log(dtrain$casual+1)
dtrain$logregister =log(dtrain$registered+1)
```

```
#we divided the train and test data
#building the models with train data
#For first building i am going with linear regression model

lm_register = lm(logregister ~
workingday+holiday+day_type+hum+atemp+windspeed+season+weathersit+weekend+yr+year_part+mnt
h+day_type,data=dtrain)

lm_casual   = lm(logcasual ~
workingday+holiday+day_type+hum+atemp+windspeed+season+weathersit+weekend+yr+year_part+mnt
h+day_type,data=dtrain)

summary(lm_casual)

summary(lm_register)

Prediction_lmreg = predict(lm_register,dtest)
dtest$logregister= Prediction_lmreg

Prediction_lmcas = predict(lm_casual,dtest)
dtest$logcasual  = Prediction_lmcas

dtest$pregistered =exp(dtest$logregister)-1
dtest$pcasual     =exp(dtest$logcasual)-1

MAPE(dtest$pregistered,dtest$registered)

#errorrate = 7.9
#accuracy = 92.1

MAPE(dtest$pcasual,dtest$casual)

#errorrate = 8.4
#accuracy = 91.6

#Now Building model with random forest

set.seed(415)
rf_register = randomForest(logregister ~
workingday+holiday+day_type+hum+atemp+windspeed+season+weathersit+weekend+yr+year_part+mnt
h+day_type, data=dtrain,importance=TRUE, ntree=250)

set.seed(415)
rf_casual   = randomForest(logcasual ~
workingday+holiday+day_type+hum+atemp+windspeed+season+weathersit+weekend+yr+year_part+mnt
h+day_type, data=dtrain,importance=TRUE, ntree=250)


Prediction_rfreg = predict(rf_register,dtest)
dtest$rflogregister= Prediction_rfreg

Prediction_rfcas = predict(rf_casual,dtest)
dtest$rflogcasual  = Prediction_rfcas
```

```
dtest$rfpregistered =exp(dtest$rflogregister)-1
dtest$rfpcasual     =exp(dtest$rflogcasual)-1

MAPE(dtest$rfpregistered,dtest$registered)

#errorrate = 11.3
#accuracy = 88.7

MAPE(dtest$rfpcasual,dtest$casual)

#errorrate = 10
#accuracy = 90


dtest$fcnt=dtest$pcasual+dtest$pregistered


#Writing the data to the final output file

s = data.frame(day=dtest$dteday,registered=dtest$pregistered,casual=dtest$pcasual,count=dtest$fcnt)
write.csv(s,file="FinalRFile.csv",row.names=FALSE)
```

References

An Introduction to Statistical Learning in application with R. Vol. 7. Springer.

Wickham, Hadley. 2009. *Ggplot2: Elegant Graphics for Data Analysis*. Springer Science & Business Media.

Machine Learning with Python *by* Json Brownlee.