Predicting the bike count on seasonal setting

BY: DIPTANSHU GAUTAM

8th February 2019

Table of Content

Introduction 2
Variables 3
Methodology 4
Correlation Matrix and Correlation Plot 6
Dimensionality Reduction 6
Model Selection 8
Appendix 11

Chapter 1

Introduction

Problem Statement

A company wants to calculate the predicted number of bikes which shall be rented for a given point of time. The number of bikes which shall be rented is dependent on the weather conditions and its various parameters. We have taken various seasonal parameters such as temperature, humidity, windspeed etc. to predict the bike count.

Data

The main aim of this project it to create a model which could predict the dependent variable based on the independent variable such as temperature, season, months etc. The dependent variable in this case is the "*Bike Count*".

Bike Count sample data (Column 1 to Column 8)

Instant	Date day	Season	Year	Month	Holiday	Weekday	Working day
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
8	1/8/2011	1	0	1	0	6	0
9	1/9/2011	1	0	1	0	0	0

Bike Count sample data continued (Column 9 to Column 16)

Weather sit	Temperature	Feel Temp	Humidity	Wind speed	Casual	Registered	Count
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	0.0895652	88	1518	1606
2	0.196522	0.208839	0.498696	0.168726	148	1362	1510
2	0.165	0.162254	0.535833	0.266804	68	891	959
1	0.138333	0.116175	0.434167	0.36195	54	768	822

Variables

There are total of 16 variables which are present in the data. "*Count"* is the dependent variable and rest 15 variables are independent. The variables are both categorical and numerical type. Names of the dependent variables are as follows:

Variables
Instant
Date
Season
year
Month
holiday
Week day
Working day
Weather sit
Temperature
Feel temp
Humidity
Wind speed
Casual
Registered

Chapter 2

Methodology

Pre-Processing

If we need to model a data, it cannot be worked as it is in its raw form. It is because, all the models are based on some assumptions and rules and regulations which cannot be broken if the model is to work. However, in the case of raw data, it is not always the case. The data in its raw form has one of the most common issue, which is empty cell. There is always a chance of human error, due to which there is an empty cell with no data. Model cannot be built upon such samples. It is due to which some manipulation of data upto some level is important to give its final touch so that the model can be built upon. Now manipulation has to be taken care of because, the hidden truth behind the data which were trying to figure out is to be preserved. This method is called **pre-processing** the data for **Exploratory data analysis.**

Thus, we will polish the data so that it may be used for the modelling purpose and inferential data could be extracted from it.

Data Structure

We need to understand the data structure or the types of variables present in the data to broadly classify the data which will help us pre-process the data easily. This is because, the factor type data are processed in a different way than the numeric data. This is because the categorical or factor data are usually nominal or ordinal level of data, where as the numeric are usually interval scale or ratio scale.

However, if we check the data type of each variable, it turns out that data has assumed that every variable is numeric type which is clearly not the case. Thus, we convert each variable into appropriate data type. Post that we check the data structure of the data.

It is also noted that there are two variables in the dataset named ***"temperature"*** and "***Feeling Temperature".*** The difference between both the temperature is that ***temperature*** is the actual measure of the heat in the atmosphere where as ***feeling temperature*** is the temperature felt by an individual. This also means that ***feeling temperature*** takes into account other aspect such as windspeed and humidity whereas ***temperature*** does not. Therefore, to make the data more concise and relevant we have removed "*atemp","humidity" and "windspeed"* from the data.

Missing Value Analysis

After converting the data type of the variable, it is very important to address the missing values. As per the company standards, there is a general rule that if any variable has a missing value of more than 30%, we remove that variable because the imputed value would affect the originality of the data way too much to model using that particular variable.

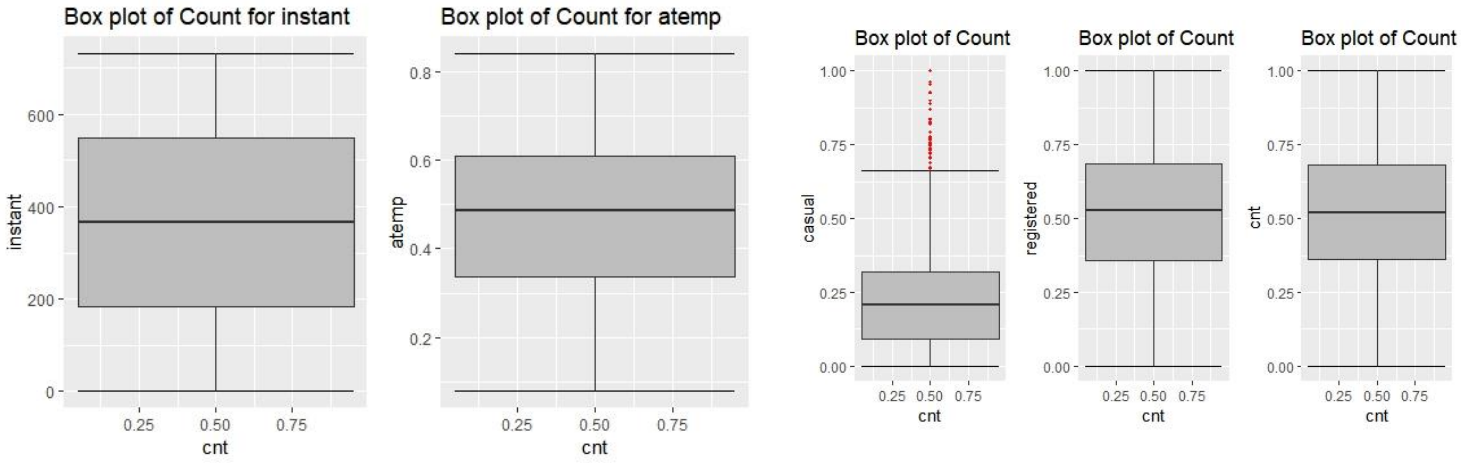
Fortunately, the data which was collected for the modelling purpose does not contain any missing value. Thus, we do not need to impute any missing value here.

Outlier Analysis

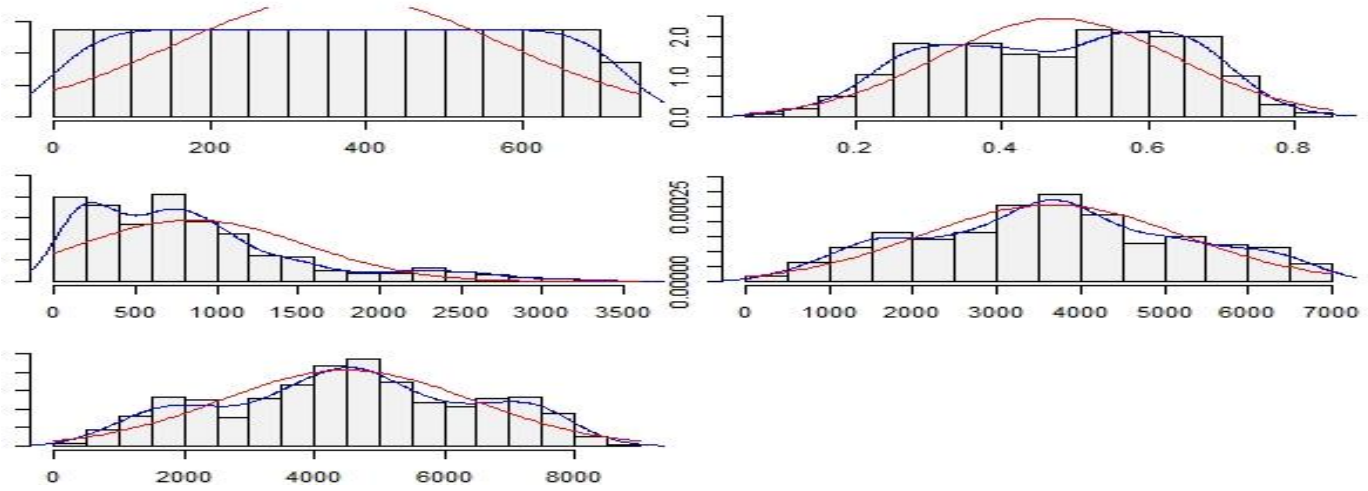
There is a very high possibility that the data which has been taken into consideration has some outlier data. The outlier data is the set of data which is outside the range (mostly interquartile range) of the dataset. It could be on the either side of the range. For e.g., in a cricket match where 8 players score on an average 50 runs, a player hits 150 and one goes to pavilion with nothing but duck! In this case, both of them are an outlier, these data are to be removed because, these data hamper the result in a drastic way. If the statistic is affected by the outlier such as mean, then all the statistic and inference which have mean will be affected by this outlier. Therefore, we tend to find and remove or impute the data.

This is also to be noted that outlier is only possible in numeric data. It can never happen with categorical type of data.

Box Plot Analysis



Multiple Histogram

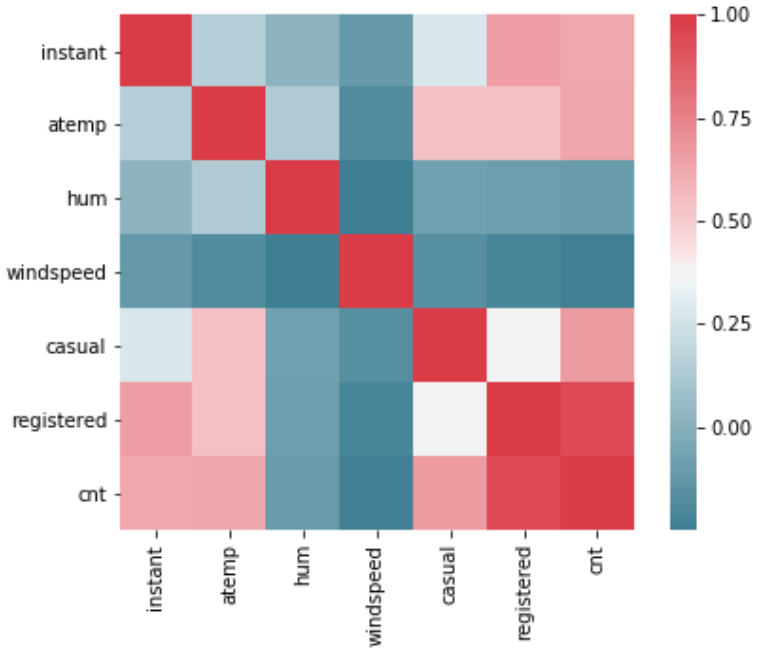


Correlation Matrix and Correlation Plot

One of the major issues in a numeric dataset, is the existence of multicollinearity in the dataset. Multicollinearity is defined when there is a high level of correlation between two independent variables. This is a particular hindrance which should be addressed as soon as possible. Thus, we check the correlation between all the variables and drew a correlation heatmap to make it even more clear.

Correlation heat map

As you can see, the color code has made the representation clearer. It has divided the correlation in mainly two categories, red and blue. Red and its shade denotes positive correlation and blue and its shade denotes negative correlation. As you can see, humidity, windspeed have an inverse relation between each other.



Thus, removing them is all the way more logical and supports our cause as mentioned above.

Dimensionality Reduction

Usually, the variables which are collected are vast in number and in size. Unfortunately, R considers each cell as a single memory block and assign 1 byte for each of them. However, we cannot afford such level of memory consumption, since R runs on Random Access Memory and could seriously make the algorithm work very slow if not taken care of. Thus, we do some statistical analysis, using various test on our variable to test exactly which variables are essentially making a significant impact on the dependent variable. Except for those variables, we remove all the other variables from the data set.

For numeric variable, we shall use Multiple regression model, and for categorical variable we shall use Analysis of variance.

Multiple Linear Regression Model

Summary

```
Residuals:
       Min         1Q     Median         3Q        Max
-2.013e-15 -2.664e-17 -2.470e-18  2.435e-17  1.936e-15

Coefficients:
             Estimate Std. Error    t value Pr(>|t|)
(Intercept)  1.435e-16  1.400e-17  1.025e+01   <2e-16 ***
atemp       -3.026e-16  3.483e-17 -8.688e+00   <2e-16 ***
casual       3.921e-01  2.573e-17  1.524e+16   <2e-16 ***
registered   7.968e-01  2.302e-17  3.462e+16   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.164e-16 on 727 degrees of freedom
Multiple R-squared:      1,    Adjusted R-squared:      1
F-statistic: 8.927e+32 on 3 and 727 DF,  p-value: < 2.2e-16
```

Please note after removing the variables such as temperate, humidity and windspeed, we can see that the p-value of all the variable is extremely less and very close to zero. Thus, we do not need to exclude any more variables anymore.

Analysis of Variance

Analysis of variance, is a method which determines the dependency and significant difference in the given samples. We shall use this method to check the effect of categorical independent variable on the dependent variable.

Summary Output

```
            Df Sum Sq Mean Sq  F value  Pr(>F)
season       3 12.582   4.194  436.234 < 2e-16 ***
yr           1 11.701  11.701 1217.030 < 2e-16 ***
mnth        11  2.479   0.225   23.443 < 2e-16 ***
holiday      1  0.044   0.044    4.553 0.03321 *
weekday      6  0.210   0.035    3.634 0.00147 **
weathersit   2  2.457   1.229  127.800 < 2e-16 ***
Residuals  706  6.788   0.010
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

As per the result, all the variables are significant when compared to the dependent variable i.e. the bike count.

Therefore, we are finally left with the following variables:

Variables
Instant
Date
Season
Year
Month
Holiday
Week day
Working day
Weather
Feeling Temperature
Casual
Registered

Chapter 3

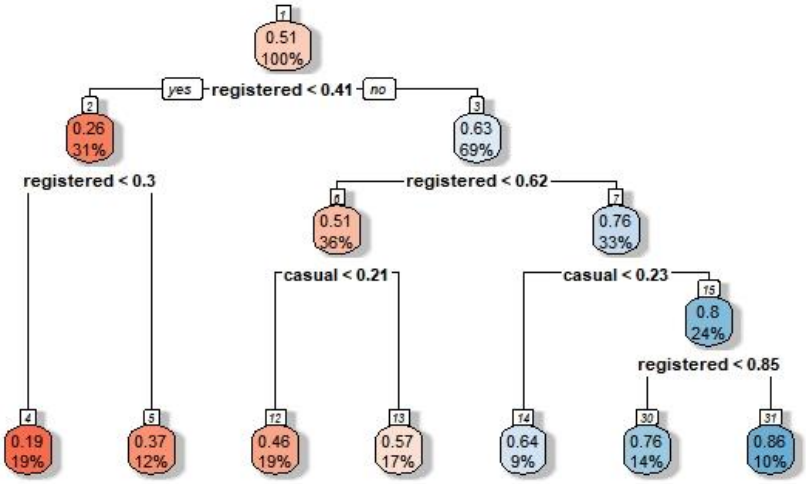
Model Selection

As it is evident, that data contains both numerical and categorical variable, regression analysis or KNN would not be appropriate for this type of data. Therefore, we would be applying random forest, decision tree model and support vector machine model. We will further check the extent of accuracy of the model to check the accuracy of the model.

Decision Tree Modelling

We use the decision tree analysis, to try and analyze the optimum rules which would be applicable for the model. Following is the decision tree which was extracted from the model.

Decision Tree

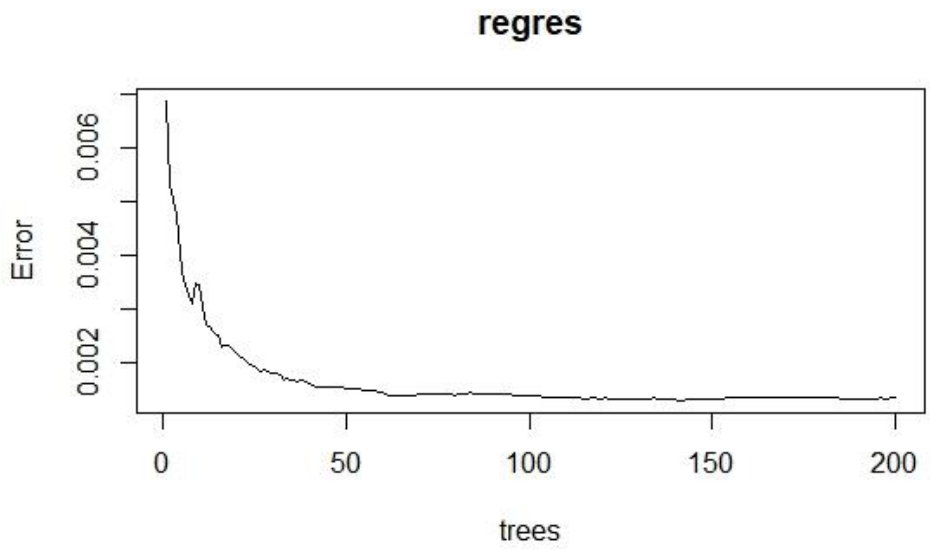


Random Forest Modelling

Unlike the decision tree model, random forest model tend to ensemble multiple decision tree model, and tries to maximize the possible number of results with greater accuracy. This increases the accuracy upto extent. If we plot the curve of the random forest, model, we can see an "***Elbow Point"*** which is supposed to the optimum number of point post which the accuracy won't differ much. Below are the plots of the random forest model along with the predicted plots.

Random Forest Plot

We can see that almost after 100 random forest plots, the results do not vary much. In fact, it is close to 50-60 trees, after which elbow has occurred. We can safely assume 100 trees. Increasing the number of trees will only increase the size of the code which will result in lagging and much time taking.



Support Vector Machine

Support Vector Machine (SVM) is a supervised machine learning algorithm which unlike simple regression model which tries to minimize the error coefficient, it tries to limit the error within the limits of the test. Mathematically, we create a correlation matrix which and then using the matrix it tries to create to predict the variables. The variables are used to predict the values of the test data.

Model Evaluation

Mean Absolute Percentage Error:

Mean absolute percentage error is defined as the sum of deviation from the actual value. It is a measure of accuracy in inferential statistics. Mathematically, it is the percentage mean error of the predicted values from the actual values. The error is as follows:

Model	Error
Decision Tree	9.82%
Random Forest	6.05%
Support Vector Machine	3.767%

Solution:

- All the three models have performed extra ordinarily well as model. However, technically, Support Vector Machine has an edge over Decision Tree and Random Forest Model.

Appendix

R Code

```
#Setting the working directory
setwd("E:\\Notes and ppts\\Edwisor\\Projects\\Project 2")
getwd()

#Uploading the data
day = read.csv("day.csv")

#removing date, temperature, windspeed and humidity column
day = day[,-c(2,10,12,13)]

#exploratory data analysis
str(day)
c = 2:8

#converting the data type
day[c] = lapply(day[c], as.factor)


#segregating numeric and factor data type
numeric.index = sapply(day, is.numeric)
numeric_data = day[,numeric.index]
factor.index = sapply(day, is.factor)
factor_data = day[,factor.index]

#Missing Value Analysis
sum(is.na(day))

#Outlier Analysis
cnames = colnames(numeric_data)
for (i in 1:length(cnames))
{
  assign(paste0("gn",i), ggplot(aes_string(y = (cnames[i]), x = "cnt"), data = subset(day))+
         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="cnt")+
         ggtitle(paste("Box plot of Count for",cnames[i])))
}
```

```
# Plotting plots together
gridExtra::grid.arrange(gn1,gn2,ncol=2)
gridExtra::grid.arrange(gn3,gn4,gn5,ncol=3)

#replacing all the outliers with NA and imputing them using knn method
val = day$casual[day$casual %in% boxplot.stats(day$casual)$out]
day$casual[day$casual %in% val] = NA

sum(is.na(day$casual))
#Number of outliers detected and removed = 44

#Imputing NA values with KNN imputation
library(DMwR)
day = knnImputation(day, k = 5)

#Creating Multiple Histograms
library('psych')
multi.hist(numeric_data, main = NA, dcol = c("Blue", "Red"), dlty = c("solid","solid"), bcol = "grey95")

#Detecting Corelation between variables
library(corrgram)
corrgram(numeric_data, order = FALSE, upper.panel = panel.pie, text.panel = panel.text,
       main = "Correlation Plot")

corre_ana = cor(numeric_data)
write.csv(corre_ana, "Correlation Analysis.csv")

#Normality Check
hist(day$casual)
hist(day$registered)

#Testing the dependence of variable using multiple linear regresssion

mult_lin = lm(cnt~., numeric_data[,-1])
summary(mult_lin)

#Testing the presence of multi collinearity
car::vif(mult_lin)

#ANOVA Testing
anova = aov(day$cnt~.,factor_data)
summary(anova)


#Creating Train and Test Data Set
library(caret)
train.index = createDataPartition(day$cnt, p=0.8, list = FALSE)
train = day[train.index,]
test = day[-train.index,]
```

```
#Decision Tree Model
library(rpart)
set.seed(1234)
regressor = rpart(cnt~., data = train)
rpart.plot::rpart.plot(regressor, box.palette = "RdBu", shadow.col = "gray", nn=TRUE)

#Predicting the test data
y_pred = predict(regressor, test)

#Evaluate the performance of regression model
devi = abs((y_pred - test$cnt)/test$cnt)
concat = as.data.frame(cbind(test$cnt, y_pred, devi))
colnames(concat) = c("Actual", "Predicted", "Deviation")

MAPE = (sum(devi)/length(y_pred))*100
#Mean Absolute Percentage error = 9.82%



#Random Forest Model
library(randomForest)
regres = randomForest(cnt~., data = train, ntree = 50)

#Plotting Random Forest and Result
plot(regres)

#predicting the test data set
pred = predict(regres, test)


#Evaluating the Random Forest Result
devi_RF = abs((pred - test$cnt)/test$cnt)
concat_RF = as.data.frame(cbind(test$cnt, pred, devi_RF))
colnames(concat_RF) = c("Actual", "Predicted", "Deviation")

MAPE_RF = (sum(devi_RF)/length(y_pred))*100
MAPE_RF
#Mean Absolute Percentage Error: 6.05%


#Support Vector Regression
library(e1071)

#Develop model
svm_model = svm(formula = cnt ~ ., data = train, type = "eps-regression")
plot(svm_model)

#predict on test cases
svm_predict = predict(svm_model, test[,1:12])
```

```
#Evaluating the Support Vector Result
devi_svm = abs((svm_predict - test$cnt)/test$cnt)
concat_svm = as.data.frame(cbind(test$cnt, svm_predict, devi_RF))
colnames(concat_RF) = c("Actual", "Predicted", "Deviation")

MAPE_svm = (sum(devi_svm)/length(y_pred))*100
MAPE_svm
#Mean Absolute Percentage Error: 3.767%
```