Case Study2: Energy Forecasting for Boston

INFO 7390: Advances in Data Sciences/Architecture
Team 9

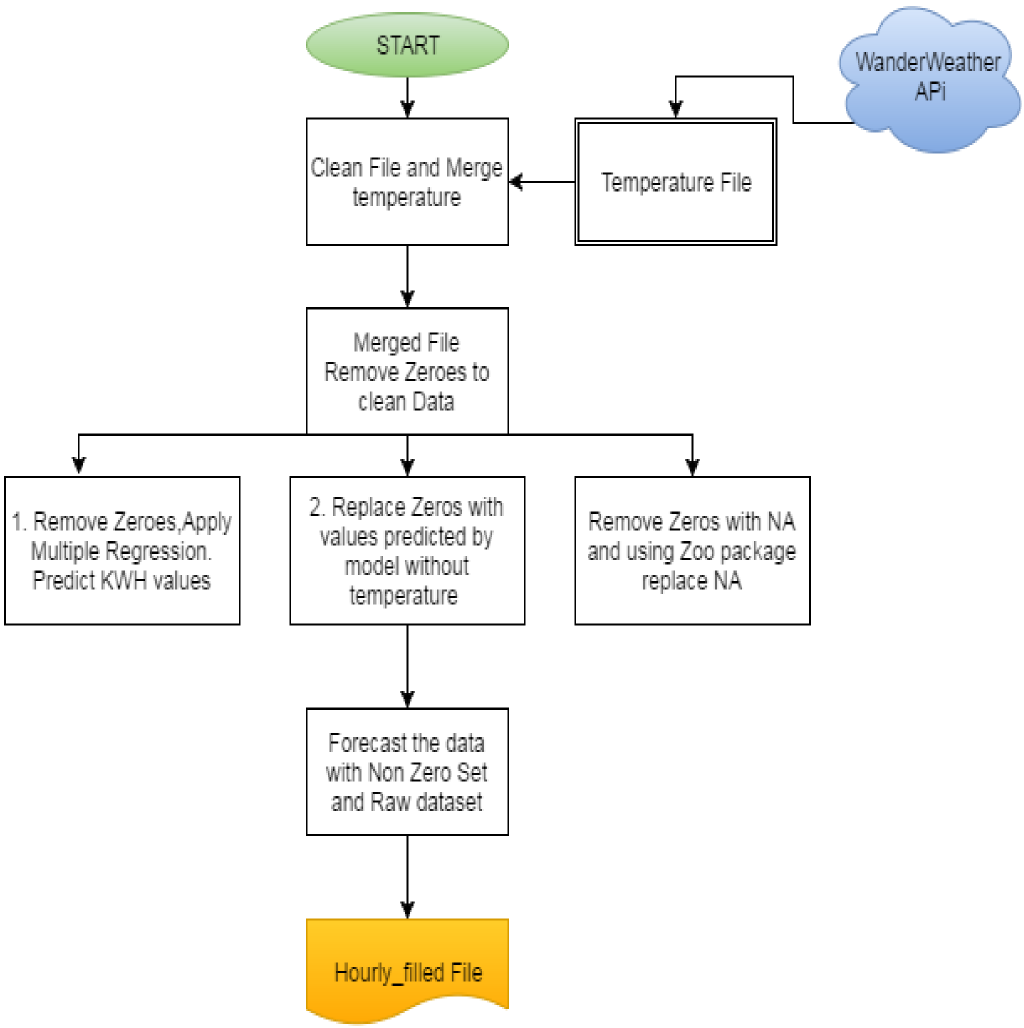
Bhavesh Patel
Ila Nigam
Ayushi Srivastava

Table Of Contents

Goal

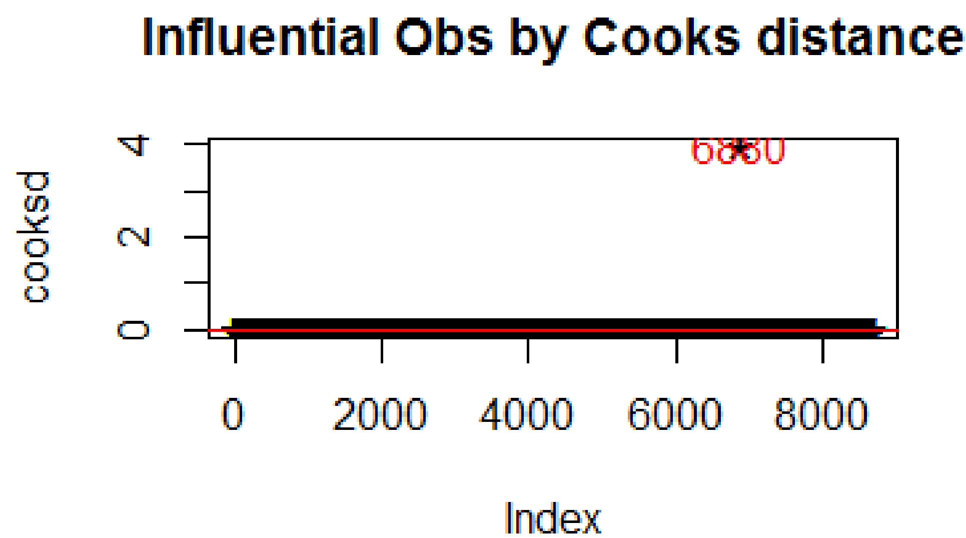
1. The aim of this project is to cleanse the data given to us and remove the zeroes in the data set such that the integrity of data is maintained. We need to implement various methods to remove Zeroes and predict the value of kwh which fits best instead of Zero.
2. Later using this sheet we have to implement Regression tree and neural networks algorithms to predict the value of Kwh. Forecast the per hour consumption of energy to the given data set on the basis of the model created in above step.
3. We also have to build classification model using Logistic Regression, Neural Networks and Classification trees. Evaluate different trends and compare the sensitivity, Specification and plot confusion matrix for all the classification method.

Data Wrangling & Cleansing and Multiple linear Regression



Data is cleansed and temperature is merged using R into final file sampleformat.csv as we did in last assignment.

1. We cleaned the data using Cook's distance to remove Outliners.



Cooks distance gives the data points which are mostly influencing the model in a negative way i.e. we can set the threshold after looking at the data as in what part of the data we need to remove so as to build an effective model.

This plot showed the outliers and we removed those so as to cleanse the data.

2. The major part of data cleaning in this assignment was removing Zeroes from data set. To solve this we used 3 approaches and predicted the "kwh" value for all the 3 approach and build a model based on multiple linear regression and also calculated the MAPE, RMS, MAE coefficients for all the scenario.

- **<u>REMOVE ALL THE ZEROES FROM DATA</u>**
  Under this method we removed all the zeroes and created a new data set. We removed outliners using Cook's method and then build a regression model to get best Adjusted R square value.
  We predicted the value of kwh against Non Zero dataset and Raw data set to get the following values.

```
lm.fit=lm(kwh~.-Date-Account-Year+I(temp^2)+I(Hour^2)-Day+I(Month^2), data=model3)
summary(lm.fit)

smp_size <- floor(0.85 * nrow(model3))
set.seed(123)
train_ind <- sample(seq_len(nrow(model3)), size = smp_size)

train <- model3[train_ind, ]
test <- model3[-train_ind, ]

lm.fit=lm(kwh~.-Account-Year+I(temp^2)+I(Hour^2)-Day+I(Month^2), data=train)
summary(lm.fit)
#str(forecastInputTest)
```

Accuracy when compared to Non Zero Data set

```
> library(forecast)
> pred = predict(lm.fit, test)
> accuracy(pred, test$kwh)
                 ME     RMSE      MAE       MPE     MAPE
Test set -0.4106077 56.0068 42.17829 -2269.257 2809.686
```

Accuracy when compared to Non Zero Data set

```
> accuracy(pred, testModel$kwh)
                ME     RMSE      MAE MPE MAPE
Test set -89.81056 128.9979 109.2586 NaN  Inf
```

- **<u>BUILD A MODEL TO REPLACE ZEROS</u>**

  Under this method we build a model without temperature first and predicted the values of "kwh" and later, we removed the zeroes with the predicted value.

  We removed outliners using Cook's method and then build a regression model to get best Adjusted R square value.

  We predicted the value of kwh against Non Zero dataset and Raw data set to get the following values.

```
lm.fit=lm(kwh~.-Date-temp-Account-Year+I(Hour^2)-Day, data=model3)
summary(lm.fit)

smp_size <- floor(0.85 * nrow(model3))
set.seed(123)
train_ind <- sample(seq_len(nrow(model3)), size = smp_size)

train <- model3[train_ind, ]
test <- model3[-train_ind, ]

lm.fit=lm(kwh~.-Account-Year-temp+I(Hour^2)-Day, data=train)
summary(lm.fit)
```

Accuracy when compared to Non Zero Data set

```
> library(forecast)
> pred = predict(lm.fit, test)
> accuracy(pred, test$kwh)
                 ME     RMSE      MAE       MPE     MAPE
Test set -0.1257138 49.52136 35.91116 -2477.557 2506.357
```

Accuracy when compared to Raw Data set

```
> library(forecast)
> pred = predict(lm.fit, testModel)
> accuracy(pred, testModel$kwh)
                 ME     RMSE      MAE MPE MAPE
Test set -50.78408 3084.717 96.11133 NaN  Inf
```

- **<u>USE ZOO PACKAGE TO REPLACE ZEROS</u>**

  Under this method we used Zoo package and replaced the values of zero using "locf" function which replaces the value with last non zero value. To do this we first replace it with NA and then apply na.locf function.

```
model1 <- na.omit(read.csv("sampleformat_both_tempa.csv",stringsAsFactors = FALSE))
model1$Date <- NULL
#zoo package
library(zoo)
model1$kwh[model1$kwh == 0] <- NA

cz <- zoo(model1$kwh)
model1$kwh <- na.locf(cz)
write.csv(model1,"sampleformat_zerosfilled_temp3a.csv",row.names = FALSE)
model1 <- na.omit(read.csv("sampleformat_zerosfilled_temp3a.csv",stringsAsFactors = FALSE))
```

```
lm.fit=lm(kwh~.-Account-Year+I(temp^2)+I(Hour^2)+I(DayOfWeek^2), data=model3)
summary(lm.fit)

# lm.fit=regsubsets (kwh~+DayOfWeek+Weekday+PeakHour+temp+I(temp^2), data=model2,nvmax=
# reg.summary =summary (lm.fit)
# 9-par(mfrow=c(1, 2))
# plot(reg.summary$rss ,xlab="Number of Variables ",ylab="RSS", type="l")
# plot(reg.summary$adjr2 ,xlab="Number of Variables ", ylab="Adjusted RSq",type="l")

smp_size <- floor(0.75 * nrow(model1))
set.seed(123)
train_ind <- sample(seq_len(nrow(model1)), size = smp_size)

train <- model1[train_ind, ]
test <- model1[-train_ind, ]

lm.fit=lm(kwh~.-DayOfWeek-Account-Year-Month-Day-Weekday, data=train)
summary(lm.fit)
```

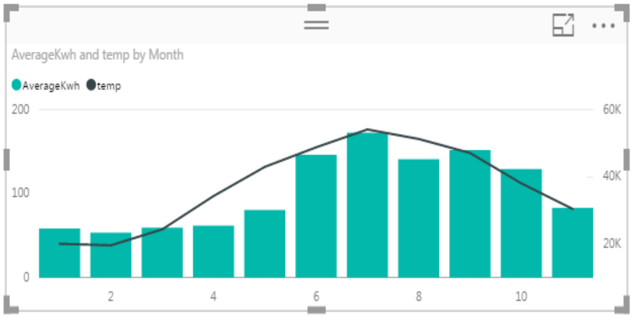
Accuracy when compared to Non Zero Data set

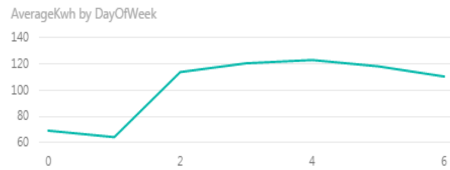
```
> library(forecast)
> pred = predict(lm.fit, test)
> accuracy(pred, test$kwh)
                ME     RMSE      MAE       MPE     MAPE
Test set 0.8787223 91.23902 64.94695 -27192.75 27225.49
```

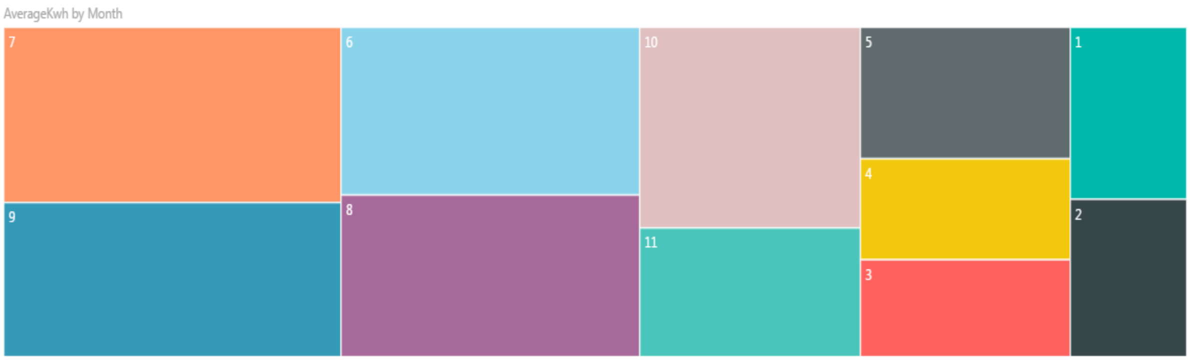
Accuracy when compared to Non Zero Data set

```
> library(forecast)
> pred = predict(lm.fit, testModel)
> accuracy(pred, testModel$kwh)
                ME     RMSE      MAE  MPE MAPE
Test set -3.567264 93.43065 67.55703 -Inf  Inf
```

DATA VISUALIZATION OF BOSTON ENERGY FORECAST DATA SET

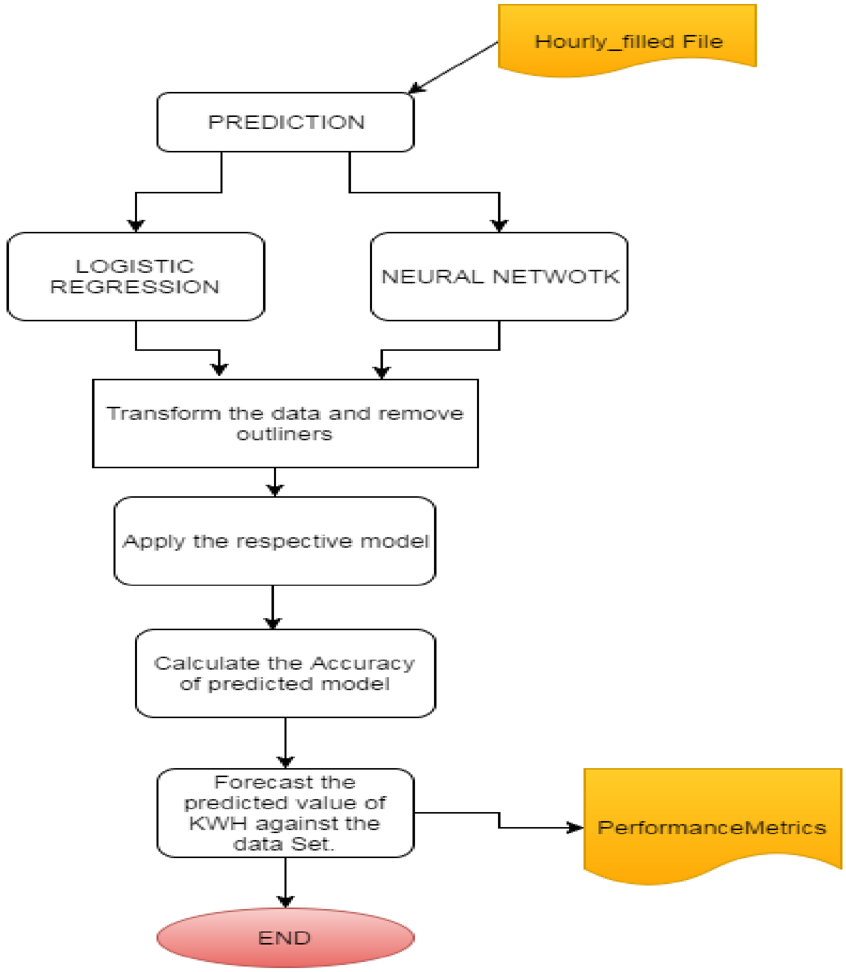






We can see the Average of Energy Consumption has a trend with Month and Day of week. On Average the 7th month has highest energy consumption and as Temperature decreases the energy consumption increases can be depicted from graph 1

Prediction

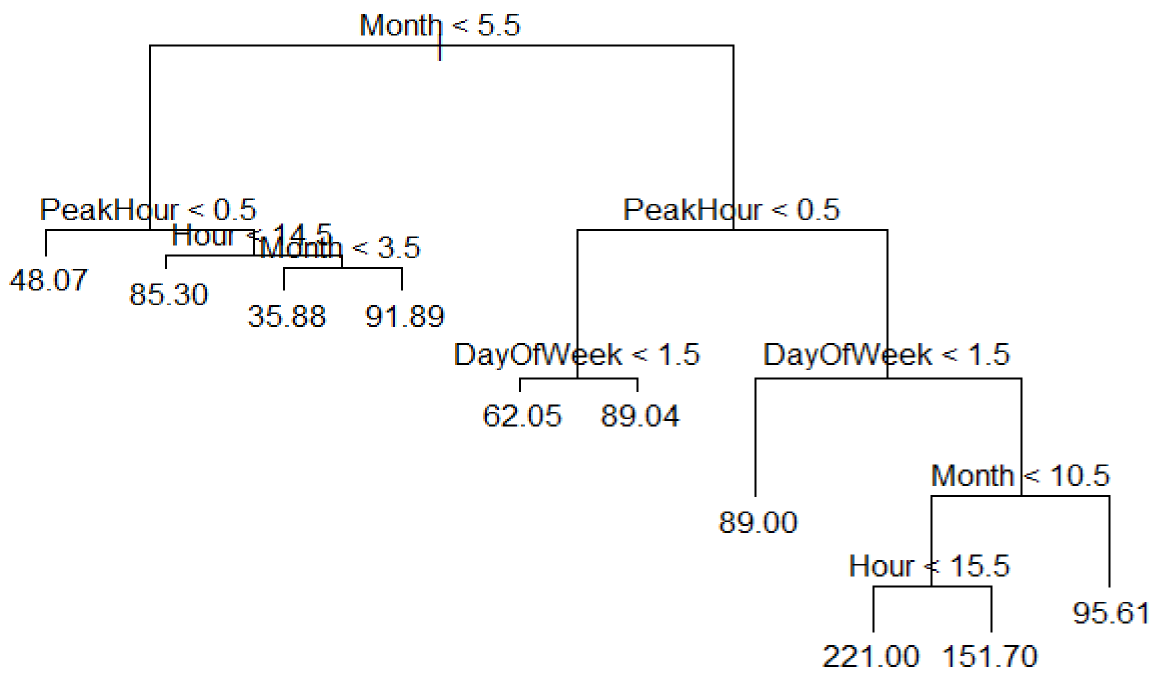


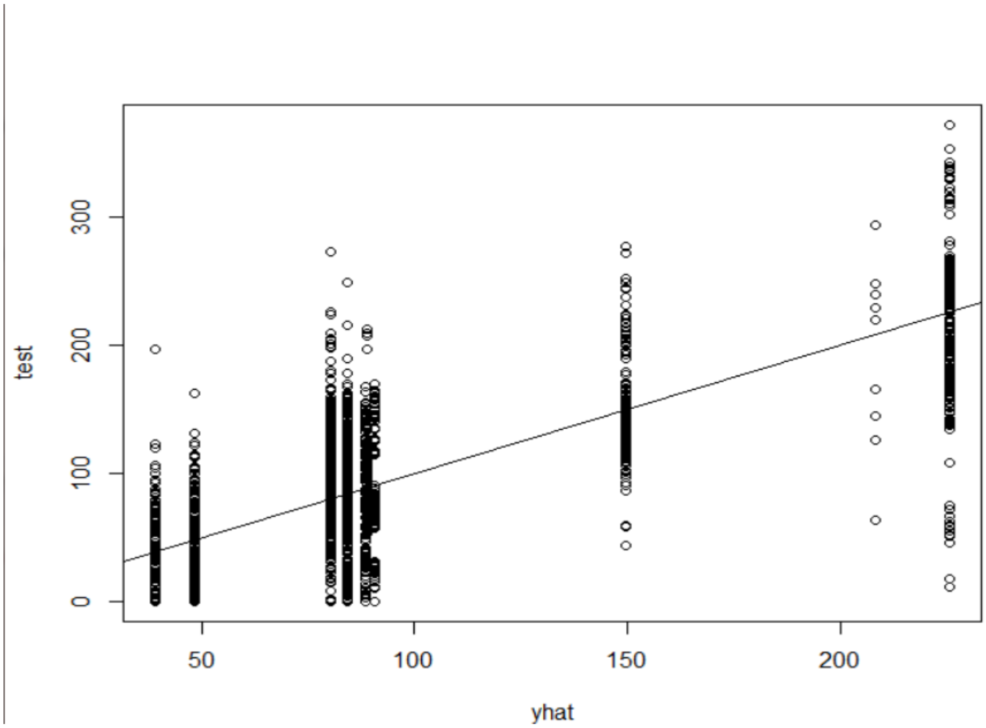
1. ***Regression Tree***

-The outliers were removed from the dataset and Regression tree model was built on dataset.

-The data was sampled into test and train and Train data was modelled with the Regression tree model

```
#Sampling Data
train = sample (1:nrow(model3), nrow(model3)/2)
tree.model3 = tree(kwh~.-Date-Account-Year-kwh,model3,subset=train)
summary (tree.model3)
plot (tree.model3)
text (tree.model3, pretty = 0)
cv.model3 = cv.tree (tree.model3)
plot (cv.model3$size, cv.model3$dev, type='b')
```


Month < 5.5
PeakHour < 0.5
Hour < 14.5
Month < 3.5
48.07
85.30
35.88
91.89
PeakHour < 0.5
DayOfWeek < 1.5
DayOfWeek < 1.5
62.05
89.04
Month < 10.5
89.00
Hour < 15.5
95.61
221.00
151.70


test
0
100
200
300
50
100
150
200
yhat

The Accuracy for the Regression Tree

```
> accuracy(yhat, test)
                 ME    RMSE      MAE       MPE     MAPE
Test set -0.2948791 35.98482 25.98622 -2511.839 2533.324
>
```

2. ***Neural Network***

- Train the neural network

- Going to have 10 hidden layers

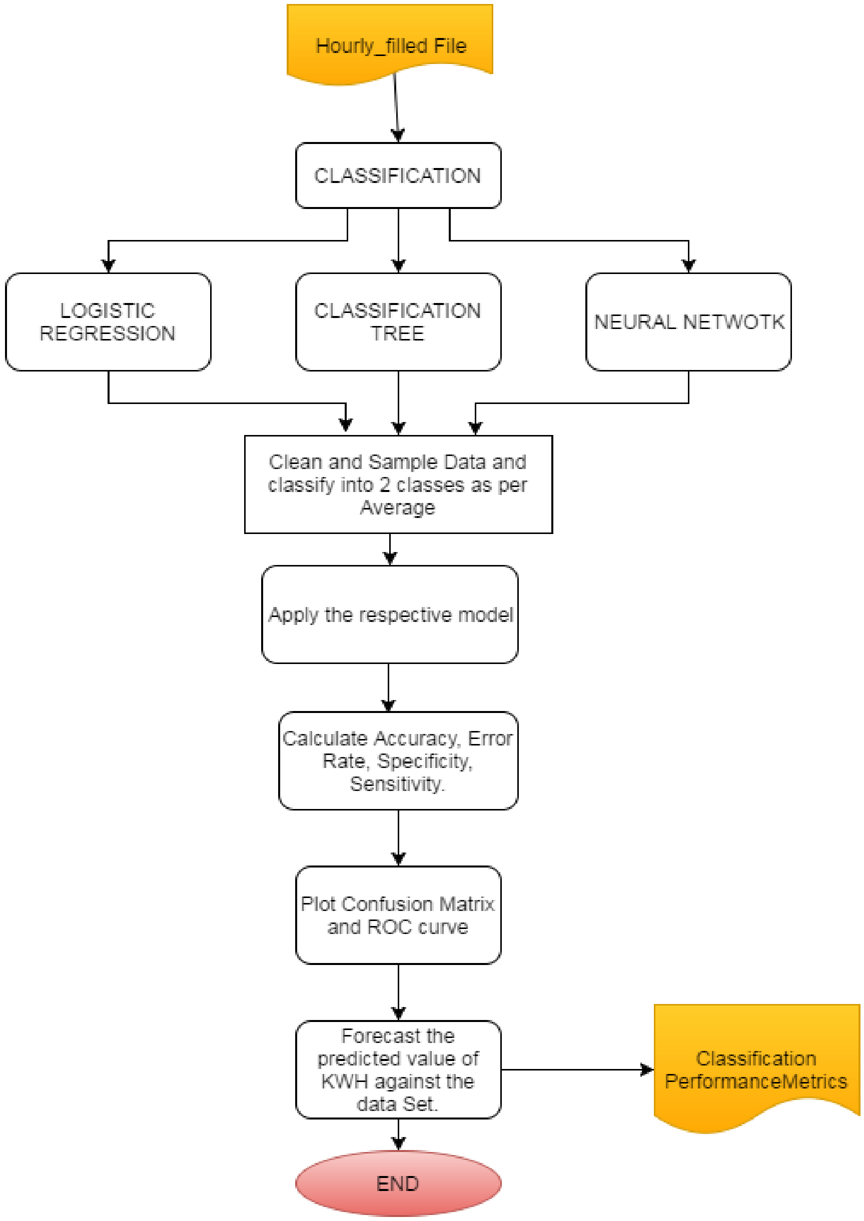
- Threshold is a numeric value specifying the threshold for the partial

- Derivatives of the error function as stopping criteria.

```
index <- sample(1:nrow(model1),round(0.75*nrow(model1)))
trainingdata <- model1[index,]
testdata <- model1[-index,]


library(nnet)
fml<- as.formula("trainingdata$kwh ~ +DayOfWeek+Weekday+PeakHour+temp+Month");
res <- nnet(fml, data=trainingdata,size=10, linout=TRUE, skip=TRUE, MaxNWts=10000, trace=FALSE, maxit=100)
pred<-predict(res, newdata=testdata)

#table(testdata$kwh,predict(res,newdata=testdata,type="class"))

prestige.rmse <- sqrt(mean((pred- testdata$kwh)^2))
```

```
> pred<-predict(res, newdata=testdata)
> #table(testdata$kwh,predict(res,newdata=testdata,type="class"))
> prestige.rmse <- sqrt(mean((pred- testdata$kwh)^2))
> # table(testdata$kwh,pred)
> prestige.rmse
[1] 68.63377186
>
```

Classification

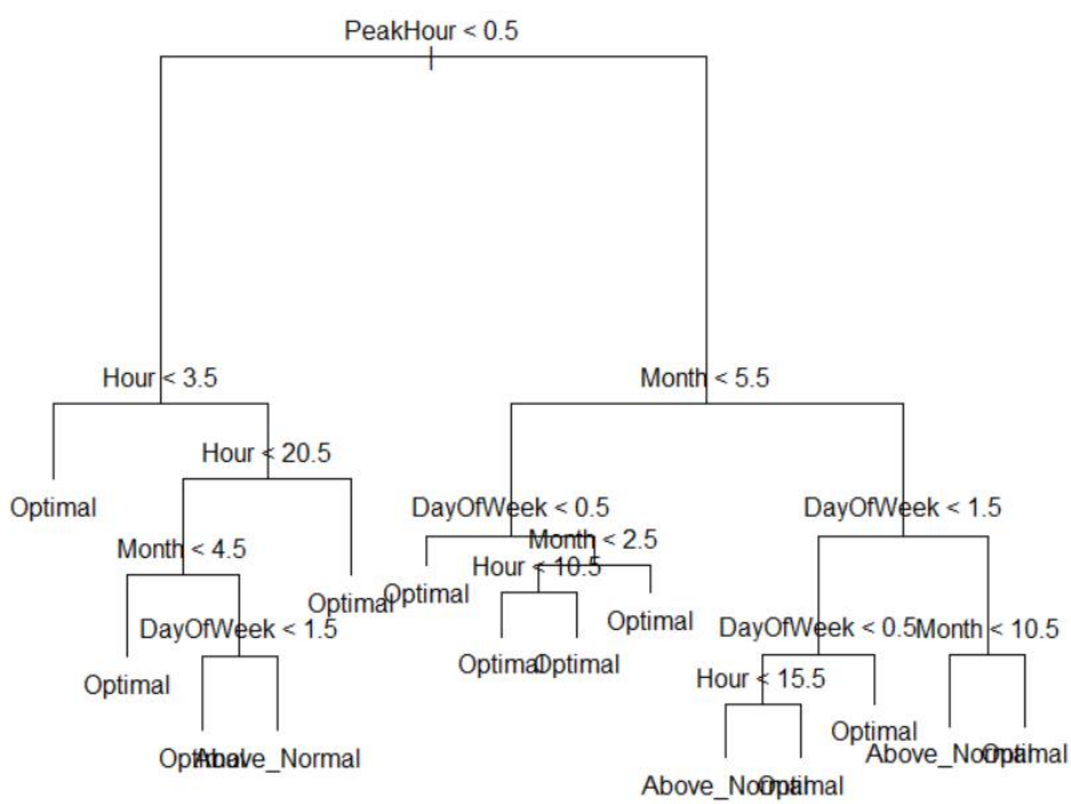


1. ***Classification Tree***

The data set was first classified into "optimal" and "Above Average" based on "Average of kwh" and then classification model was applied

```
#Transform kwh into a dichotomous factor for classification
averagekwh <- mean(model1$kwh)

model1$KWH_Class[model1$kwh > averagekwh] <- 1
model1$KWH_Class[model1$kwh <= averagekwh] <- 0
model1$KWH_Class <- factor(model1$KWH_Class,
                           levels=c(0,1),
                           labels=c("Optimal","Above_Normal"))
```

```
#Use variables to fit a classification tree
library(tree)
tree.train = tree(train$KWH_Class ~+PeakHour+Hour+Month+DayOfWeek+Weekday, data=train
summary(tree.train)

#Display the tree structure and node labels
plot(tree.train)
text(tree.train, pretty =0) #Pretty=0 includes the category names

forecastTestInput <- na.omit(read.csv("forecastInput1.csv",stringsAsFactors = FALSE))
tree.pred = predict(tree.train, forecastTestInput, type = "class")
table(tree.pred, test$KWH_Class)
```



Confusion Matrix :

```
Confusion Matrix and Statistics

              Reference
Prediction     Optimal Above_Normal
  Optimal         1204          165
  Above_Normal      94          361

               Accuracy : 0.8580044
                 95% CI : (0.8411366, 0.8737114)
    No Information Rate : 0.7116228
    P-Value [Acc > NIR] : < 0.00000000000000022204

                  Kappa : 0.6395653
 Mcnemar's Test P-Value : 0.00001363933

            Sensitivity : 0.9275809
            Specificity : 0.6863118
         Pos Pred Value : 0.8794741
         Neg Pred Value : 0.7934066
             Prevalence : 0.7116228
         Detection Rate : 0.6600877
   Detection Prevalence : 0.7505482
      Balanced Accuracy : 0.8069463

       'Positive' Class : Optimal
```

2. ***Logistic Regression***

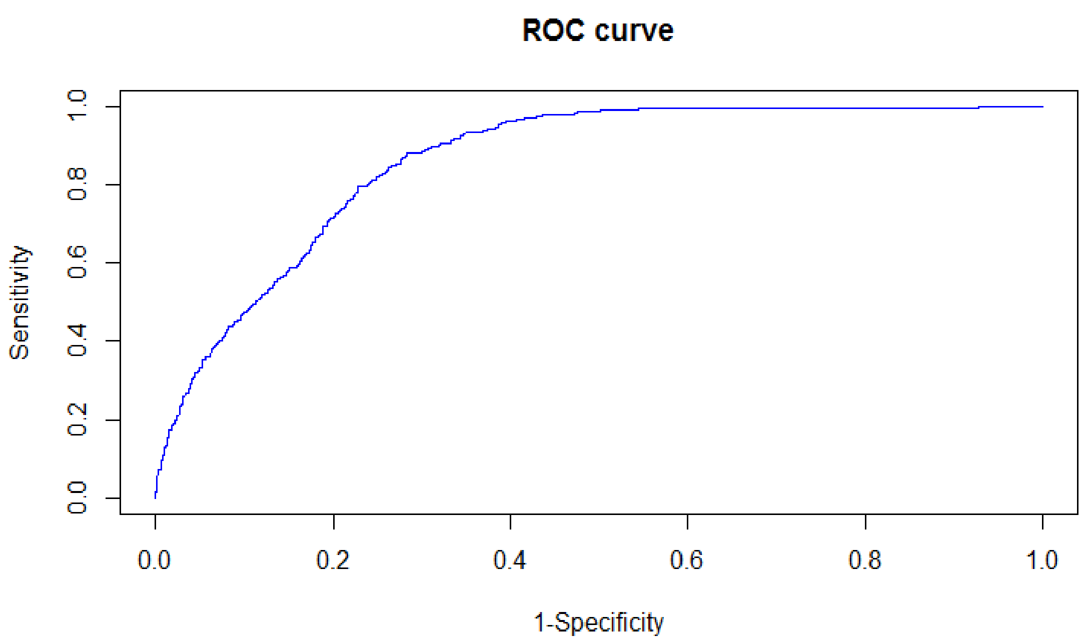
The data set was first classified into “optimal” and “Above Average” based on “Average of kwh” and then classification model was applied.

The outliners were removed and logistic regression was built on dataset with GLM model.

```
#Build Logistic Regression
fit1 <- glm(KWH_Class~+DayOfWeek+I(temp^2)+I(Hour^2)+Hour+Weekday,data=train, family=binomial(link="l
summary(fit1)

forecastTestInput <- na.omit(read.csv("forecastInput1.csv",stringsAsFactors = FALSE))
prob <- predict(fit1, newdata=forecastTestInput, type="response")
pred <- rep("Optimal",length(prob))

#Set the cutoff value =0.5
pred[prob>=0.5] <- "Above_Normal"

#MergeData
dataToMerge <- na.omit(read.csv("forecastNewData1.csv",stringsAsFactors = FALSE))
predictionResults <- data.frame(dataToMerge, KWH_Class = pred)
write.csv(predictionResults,"forecastOutput_26435791004_regressionTree1.csv",row.names = FALSE)
```

Confusion matrix and ROC curve was plotted.

```
Confusion Matrix and Statistics

              Reference
Prediction     Optimal Above_Normal
  Optimal         1159          269
  Above_Normal     139          257

               Accuracy : 0.7763158
                 95% CI : (0.7564817, 0.7952571)
    No Information Rate : 0.7116228
    P-Value [Acc > NIR] : 0.0000000002428433

                  Kappa : 0.4117692
 Mcnemar's Test P-Value : 0.0000000001697791

            Sensitivity : 0.8929122
            Specificity : 0.4885932
         Pos Pred Value : 0.8116246
         Neg Pred Value : 0.6489899
             Prevalence : 0.7116228
         Detection Rate : 0.6354167
   Detection Prevalence : 0.7828947
      Balanced Accuracy : 0.6907527

       'Positive' Class : Optimal
```

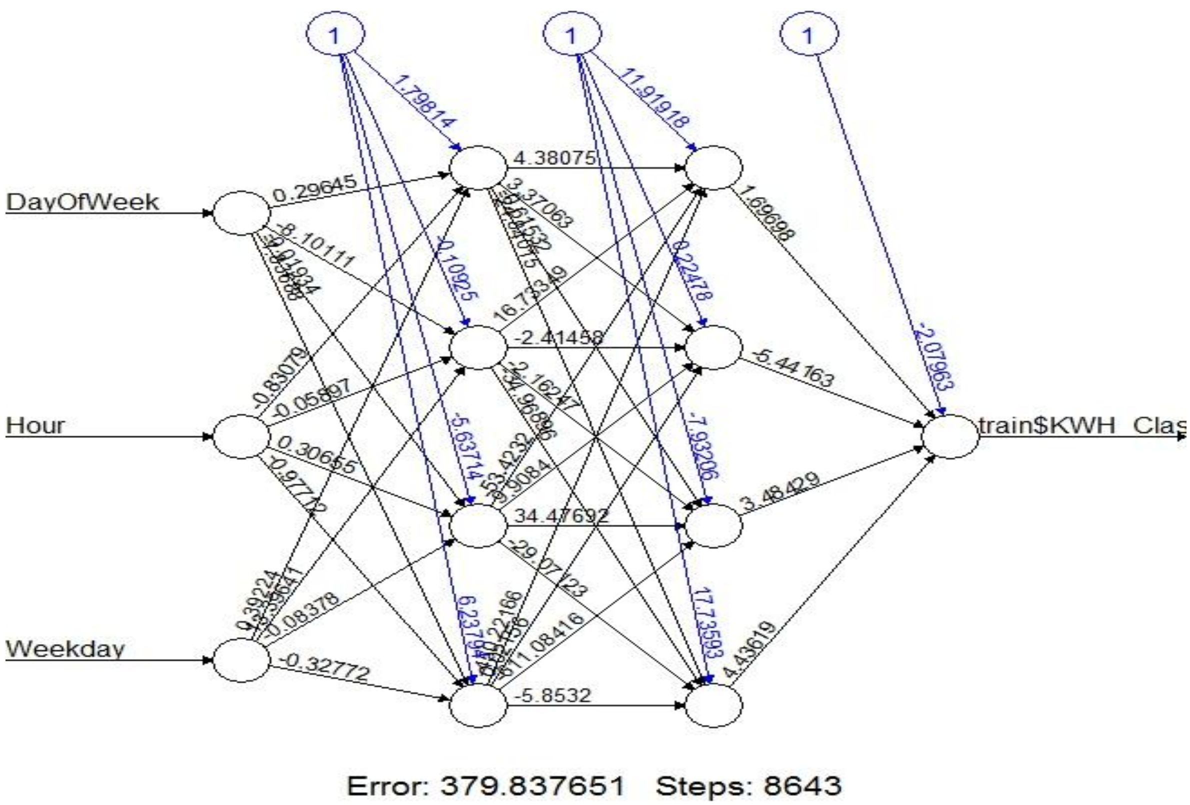


3. ***Neural Network Classification***

- Data set was cleaned and sample was created.

- Neural Network model was applied to predict kwh value. The network has 4 hidden layers
- Confusion matrix and plotted the neural network and the optimum or Above normal status of dataset was classified.

```
neuralnet <- neuralnet(train$KWH_Class ~ DayOfWeek+Hour+Weekday, data=train,
                        hidden=c(4,4), err.fct="sse", linear.output=FALSE,threshold = 0.1,lifesign = "minimal")
##plot network
neuralnet$result.matrix
plot(neuralnet)

##Predict
temp_test <- subset(test, select = c("DayOfWeek", "Hour","Weekday"))
test.results <- compute(neuralnet, temp_test)
#test.results <- round(test.results$net.result)

results <- data.frame(actual = test$KWH_Class, prediction = test.results$net.result)
results$prediction <- round(results$prediction)
str(results)

results$prediction <- factor(results$prediction,
                       levels=c(0,1),
                       labels=c("Optimal","Above_Normal"))

results$actual <- factor(results$actual,
                       levels=c(0,1),
                       labels=c("Optimal","Above_Normal"))

confusionMatrix(results$prediction,results$actual)
```



```
> confusionMatrix(results$prediction,results$actual)
Confusion Matrix and Statistics

              Reference
Prediction     Optimal Above_Normal
  Optimal         1073          216
  Above_Normal     225          310

               Accuracy : 0.7582237
                 95% CI : (0.7378945, 0.777719)
    No Information Rate : 0.7116228
    P-Value [Acc > NIR] : 0.000004490579

                  Kappa : 0.413904
 Mcnemar's Test P-Value : 0.7032386

            Sensitivity : 0.8266564
            Specificity : 0.5893536
         Pos Pred Value : 0.8324282
         Neg Pred Value : 0.5794393
             Prevalence : 0.7116228
         Detection Rate : 0.5882675
   Detection Prevalence : 0.7066886
      Balanced Accuracy : 0.7080050

       'Positive' Class : Optimal
```

****