**Assignment 2 Report~ Team1**

1. **Data Cleansing:**
2. Regression model with the “raw dataset” including zeros

Building a Regression model with the “raw dataset” including zeros –

A linear regression model was implemented with the following attributes: Peak hour, Temperature, Weekday, Day of Week, Hour, Day and Month.

On fitting the linear regression model, the measures of predictive accuracy are as follows:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | ME | RMSE | MAE | MPE | MAPE |
| Test set | -1.563999 | 116.31718 | 88.09954472 | NaN | Inf |

ADJUSTED R SQUARE= 0.4075647

1. Regression model with the “non zero” dataset

Building a Regression model with the “raw dataset” excluding zeros by cleansing the data in Python–

A linear regression model was implemented in R with the following attributes: Peak hour, Temperature, Weekday, Day of Week, Hour, Day and Month.

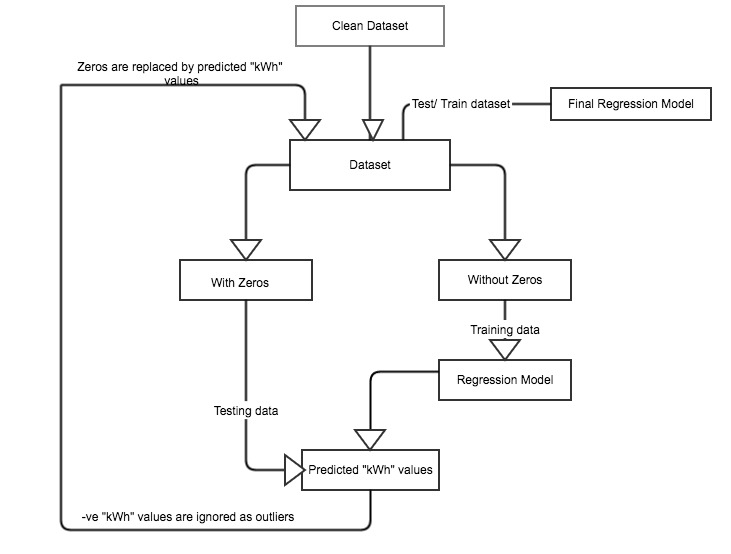
On fitting the linear regression model, the measures of predictive accuracy are as follows:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | ME | RMSE | MAE | MPE | MAPE |
| Test set | -4.376781 | 119.477846 | 92.09790606 | -6609.404 | 6648.9947 |

ADJUSTED R SQUARE= 0.3484754

1. Regression model with replaced zeros with predicted values

Building a Regression model with the “raw dataset” excluding zeros with predicted values. The regression model is first trained on a dataset with doesn’t have any zeros then the part of the dataset with have zeros are predicted and finally the regression model is run on the whole dataset with had previously “kWh” values and the also the predicted “kWh”



On fitting the linear regression model, the measures of predictive accuracy are as follows:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | ME | RMSE | MAE | MPE | MAPE |
| Test set | -0.281915 | 108.027354 | 79.95937254 | -4195.617 | 4243.9364 |

ADJUSTED R SQUARE= 0.4191468

1. Regression model with replaced zeros by zoo package

Using “**locf function**” of the zoo package, first replaced the zeros in the test data with “NA” and then further used the na.locf function to replace the “NA” values of kWh.

On fitting the linear regression model, the measures of predictive accuracy are as follows:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | ME | RMSE | MAE | MPE | MAPE |
| Test set | -1.570623 | 113.21921 | 85.15383016 | -39107.76 | 41102.4972 |

ADJUSTED R SQUARE= 0.3973992

Using “**approx. function**” of the zoo package, first replaced the zeros in the test data with “NA” and then further used the na.approx function to replace the “NA” values of kWh.

On fitting the linear regression model, the measures of predictive accuracy are as follows:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | ME | RMSE | MAE | MPE | MAPE |
| Test set | -1.183909 | 111.65126 | 83.69516355 | -5257.033 | 5482.67805 |

ADJUSTED R SQUARE= 0.3909442

Using “**fill.function**” of the zoo package, first replaced the zeros in the test data with “NA” and then further used the na.approx function to replace the “NA” values of kWh and then use na.fill function to replaced the trailing NA values.

On fitting the linear regression model, the measures of predictive accuracy are as follows:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | ME | RMSE | MAE | MPE | MAPE |
| Test set | -1.579120 | 111.67315 | 83.7589262 | -6356.035 | 6756.73950 |

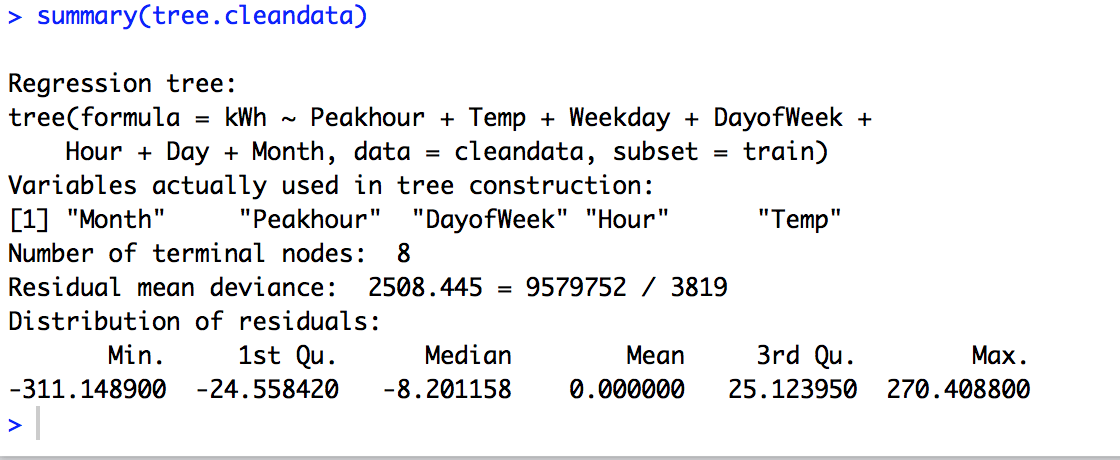
ADJUSTED R SQUARE= 0.3917891

The technique that works the best is using predicted values of kWh. This technique uses the dataset without zeros to train the model and then further predicts values for the zero values. Finally, the dataset with new predicted values of kWh and the non-zero values of kWh are used to create the final regression model. This model gives around 300 rows of negative kWh values, we have decided to ignore the negatives values as treating them as outliers as kWh can never be negative and 300 rows is around 3.8% of the whole dataset.

The adjusted R square value = 0.4075647, which is the highest in all the methods.

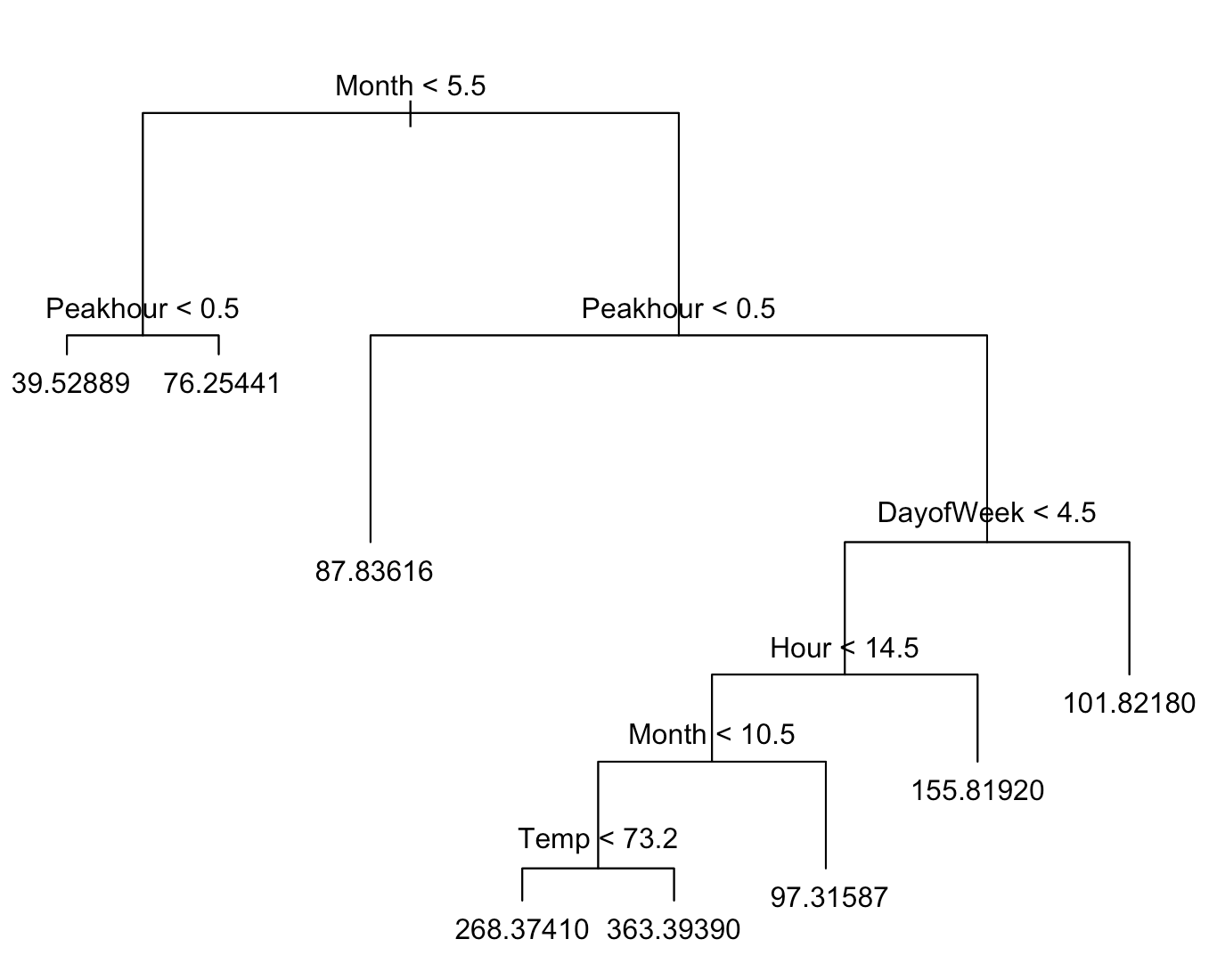
1. **Prediction & Forecast:**
2. Regression trees-

Fitting the Regression Tree Model on to the cleansed dataset without zeros or negative “kWh” values.

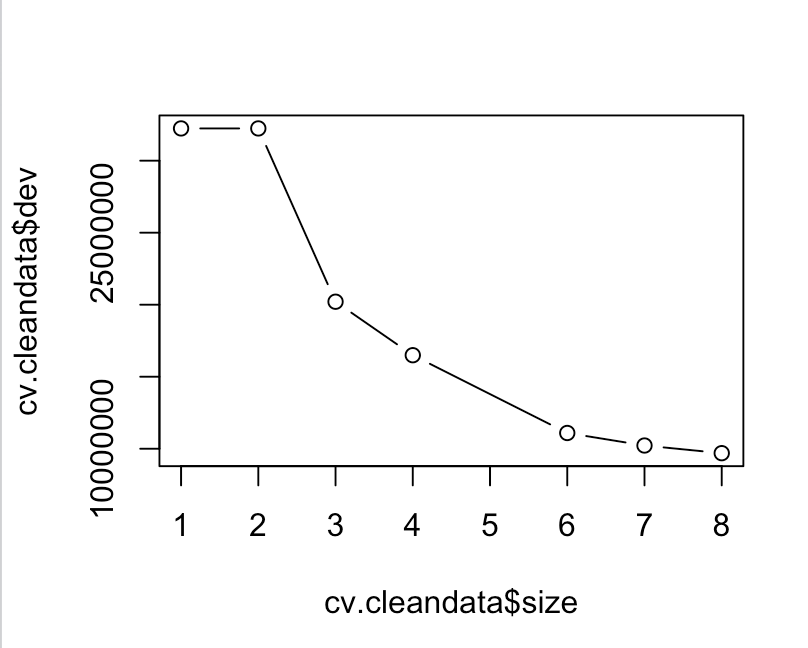


The model picks only 5 variables namely: Month, Peak hour, Day of the week, Hour, Temperature as these are the variables according to the model that effect the model the most.

On plotting the tree:



Then the tree goes though Cross-Validation to check whether pruning the tree further would improve the performance of the tree model.



After Pruning the tree according to the $dev i.e. deviance produced we use best=7 and grow a tree.

MSE for tree = 2603.957827

MSE for pruned tree = 2762.506921

Therefore the tree without pruning in a better model.

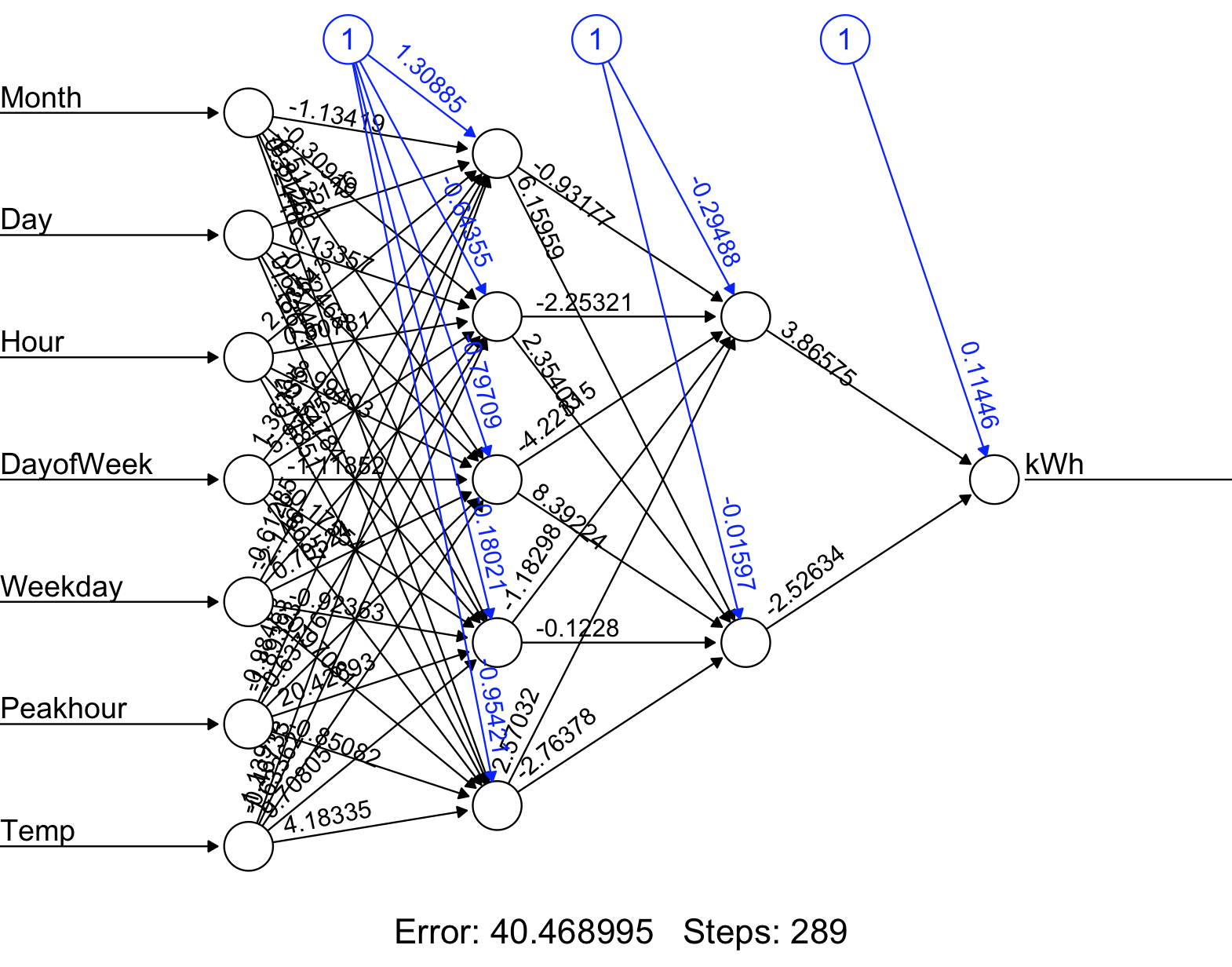
To forecast values for “kWh” first the files forecastNewData.csv and forecastNewData2.csv are cleansed and converted to the sample format. The cleansing of data is done in Python (scripts are provided).

For forecasting the tree model is trained on the dataset hourly\_filled\_data.csv and the testing dataset is forecastInput.csv. Predicted values of “kWh” is stored in a separated csv file called “forecastNewdataOutput\_Tree2” (csv file is provided)

1. Neural Network-

Fitting the Neural Network Model on to the cleansed dataset without zeros or negative “kWh” values. The dataset is divided into train and test, it is further converted to numeric

After scaling and plotting the neural network the network is as follows:



The accuracy measures are as follows:

RMSE = 66.68735933

MSE = 4447.203895

MAE = 45.57783088

MAPE = 43.97612948

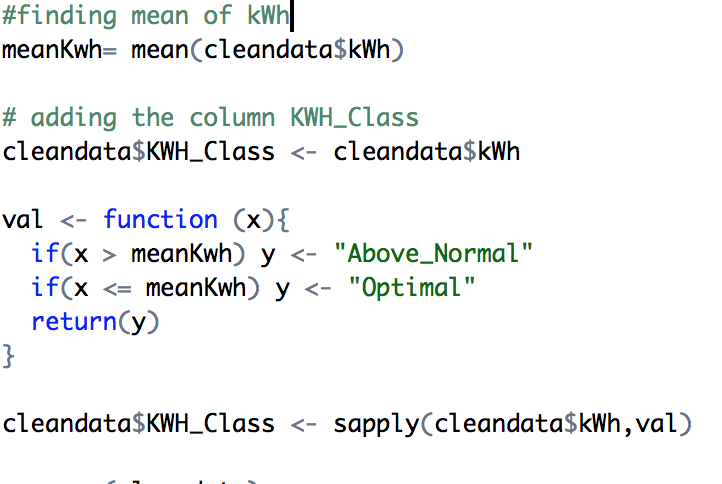
For forecasting the neural network model is trained on the dataset hourly\_filled\_data.csv and the testing dataset is forecastInput.csv. Predicted values of “kWh” is stored in a separated csv file called “NeuralNetwork\_ForcastOutput” (csv file is provided)

1. **Classification:**
2. Logistic Regression-

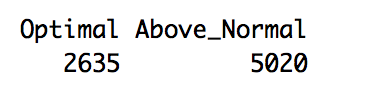
Fitting the Logistic Regression Model on to the cleansed dataset without zeros or negative “kWh” values.

For the purpose of classification a new column is created called “KWH\_Class” and the values in this column are either “Optimal” or “Above\_Normal”

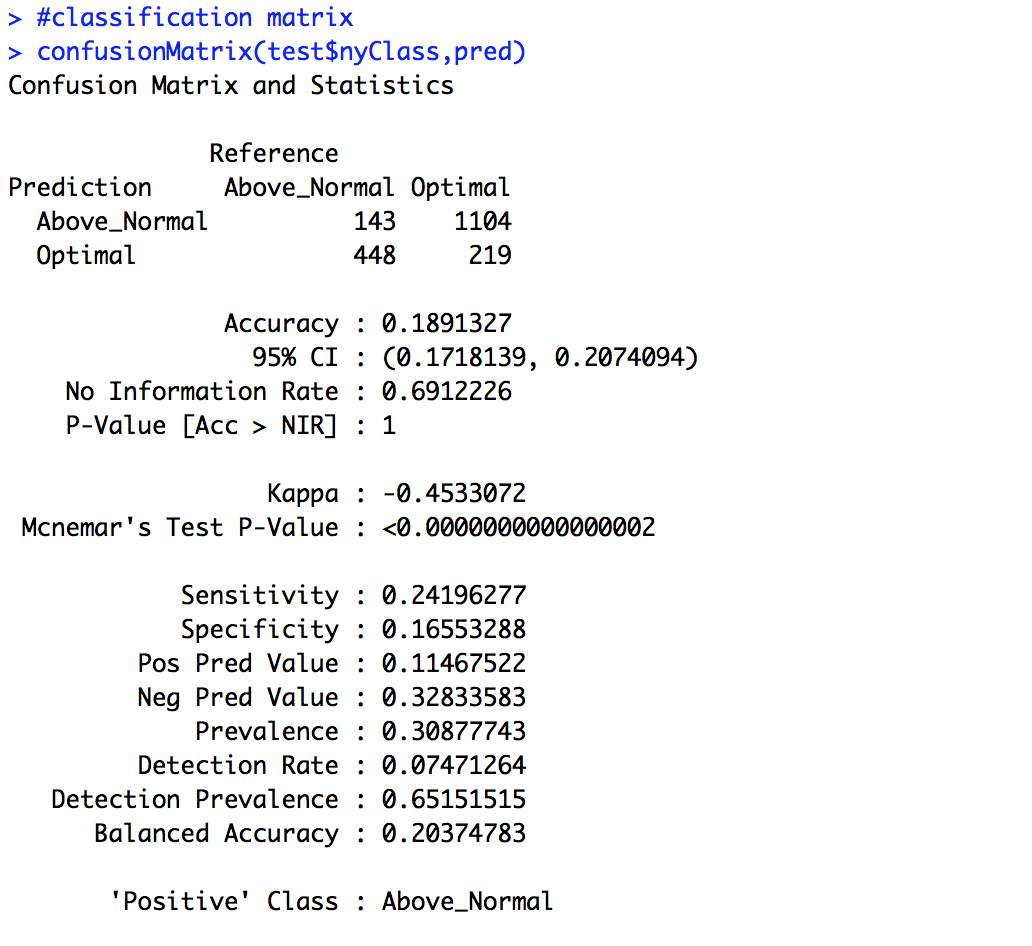
Condition (code snippet in R):



On Transforming “KWH\_Class” into binary (0,1)



After splitting the “hourly\_filled\_data.csv” into train and test data and running logistic regression the Classification matrix obtained is as follows:



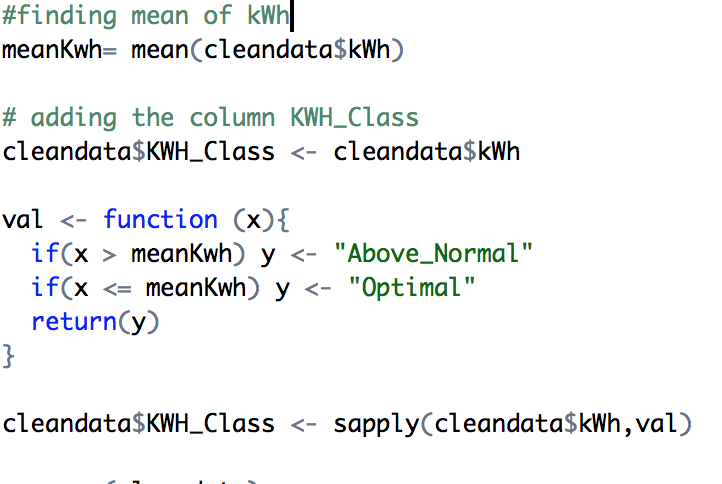
1. Classification Tree-

Fitting the Classification Tree Model on to the cleansed dataset without zeros or negative “kWh” values.

A new column is added to the dataset called “KWH\_Class”.

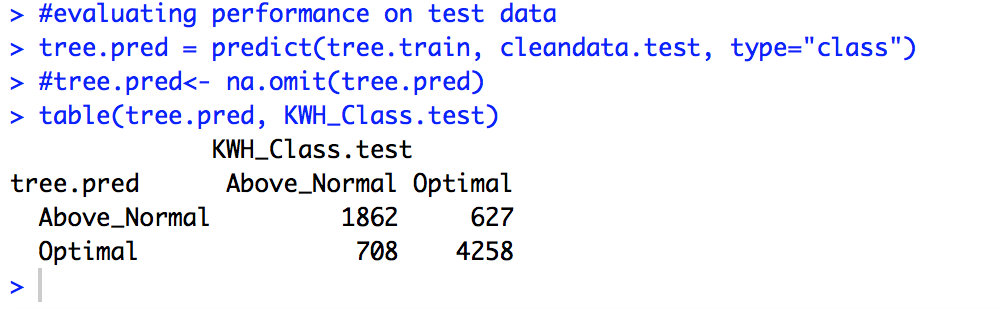
This column contains two values, “Optimal” and “Above\_Normal”.

Condition:



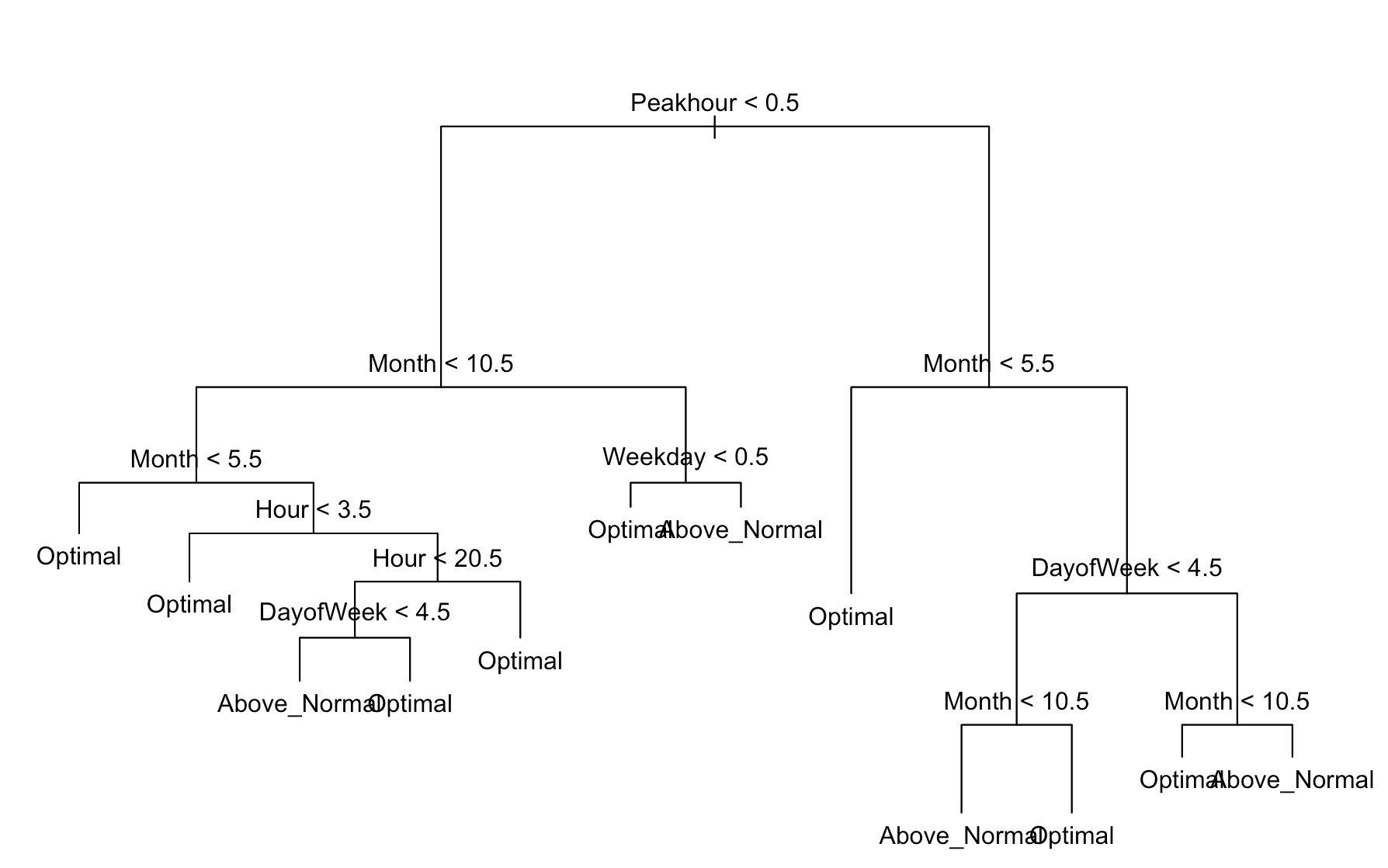
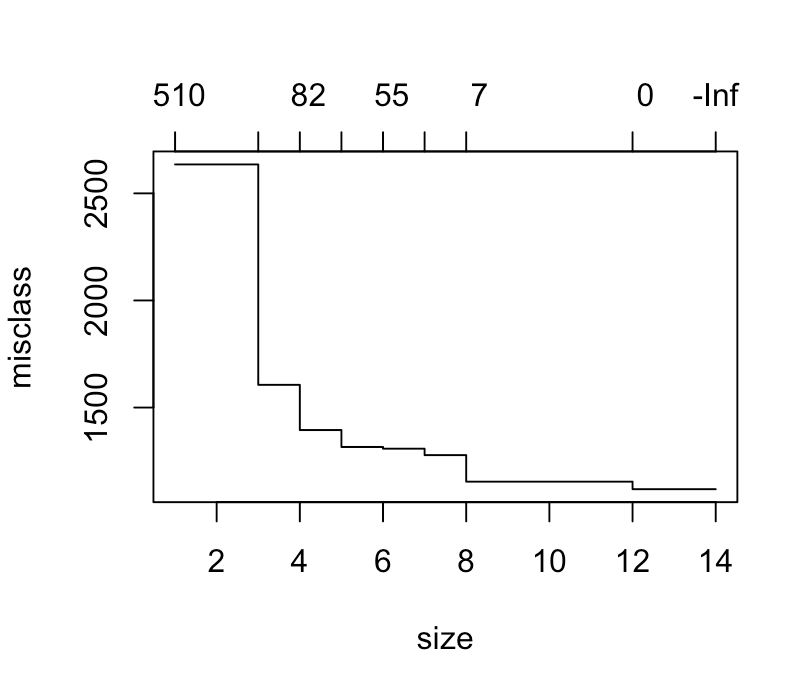
To predict the value of “KWH\_Class” the tree() function is used.

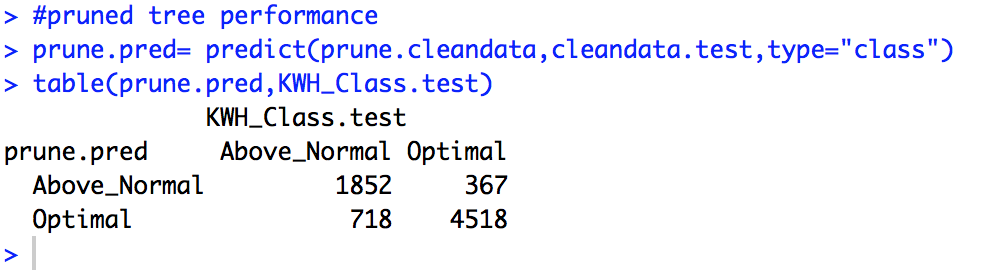
On splitting the dataset into test and train the classification matrix is as follows:



Error rate= (627+708)/(1862+4258)= 21%

On cross validation of the tree and pruning the tree with best=12



The classification matrix of the pruned tree is as follows:

Error rate= (718+367)/(1852+4518)=17.03 %

Therefore, The pruned model is to be selected, as the error rate is lesser.

For forecasting the tree model is trained on the dataset hourly\_filled\_data.csv and the testing dataset is forecastInput.csv. Predicted values of “kWh” is stored in a separated csv file called “forecastOutput\_ClassificationTree.csv” (csv file is provided)

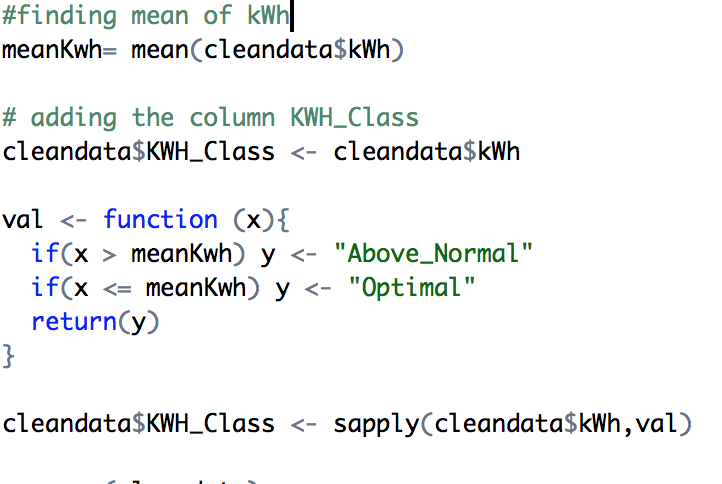
1. Neural Network-

Fitting the Neural Network Classification Model on to the cleansed dataset without zeros or negative “kWh” values.

A new column is added to the dataset called “KWH\_Class”.

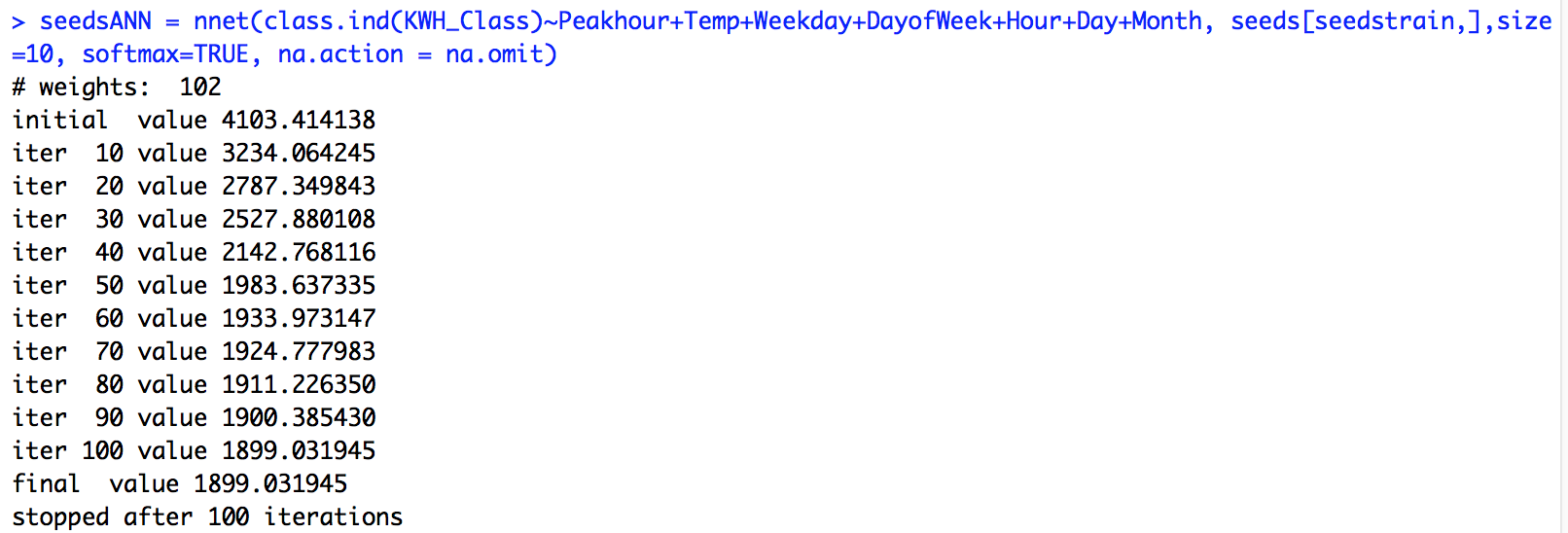
This column contains two values, “Optimal” and “Above\_Normal”.

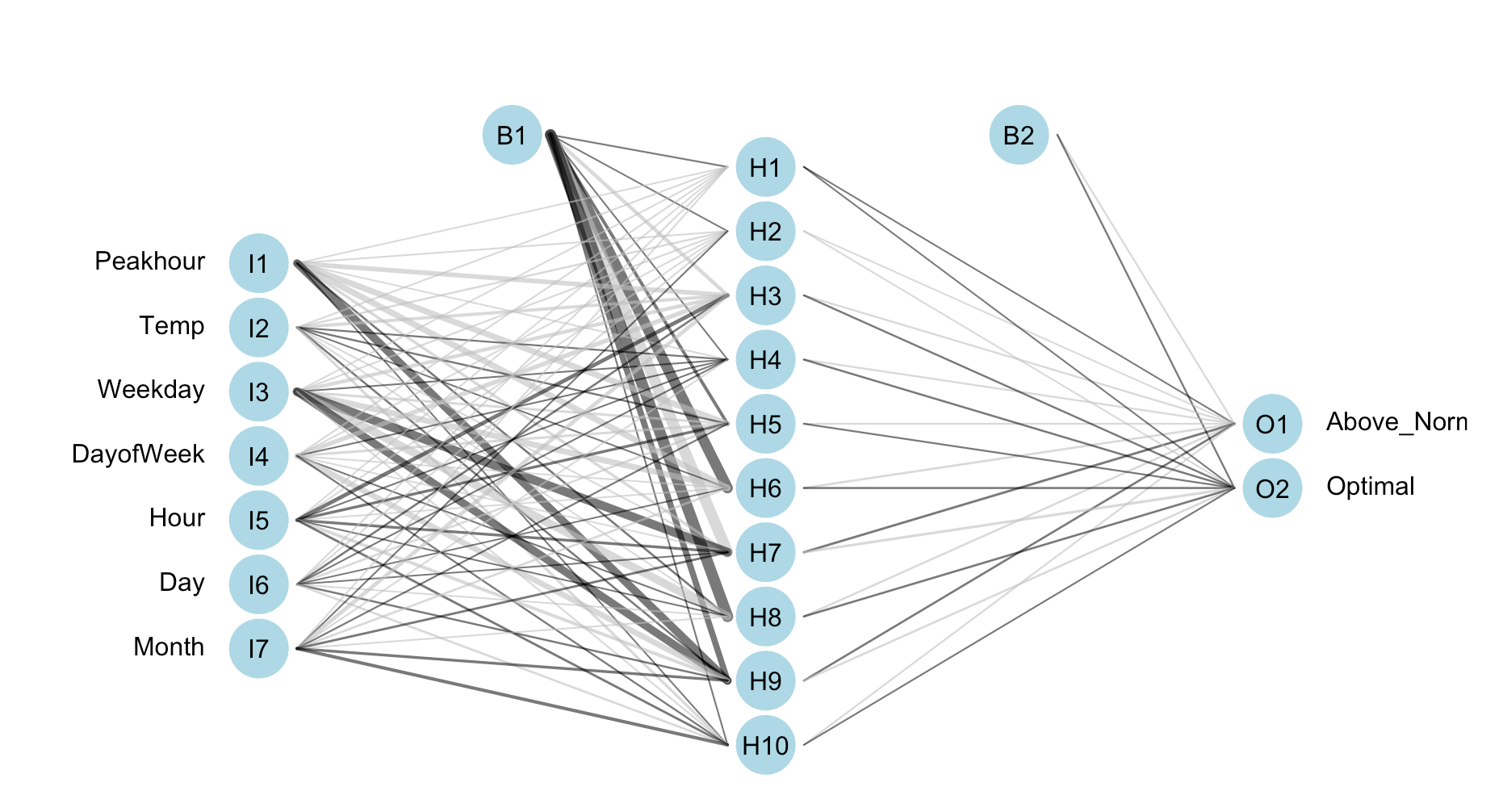
Condition:



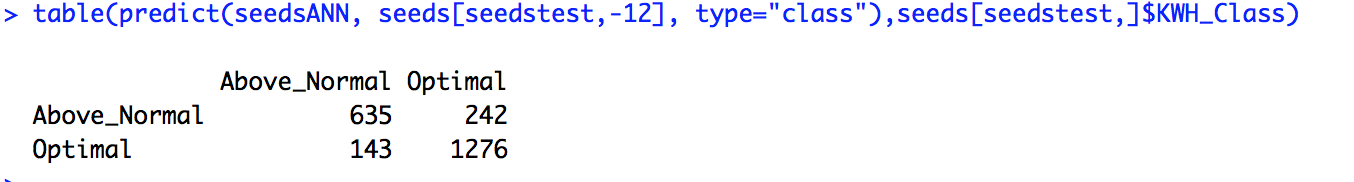
On splitting the dataset into test and train data and removing the “NA” values using na.omit and using the variables:

Peak hour, Temperature, Weekday, Day of Week, Hour, Day and Month. The neural network that was generated is as follows along with its iterations:





The classification Matrix that was generated is as follows:



Error Rate = (143+242)/(635+1276)=20.14%

For forecasting the neural network model is trained on the dataset hourly\_filled\_data.csv and the testing dataset is forecastInput.csv. Predicted values of “kWh” is stored in a separated csv file called “forcastNewdataOuput\_ClassificationNeural.csv” (csv file is provided)