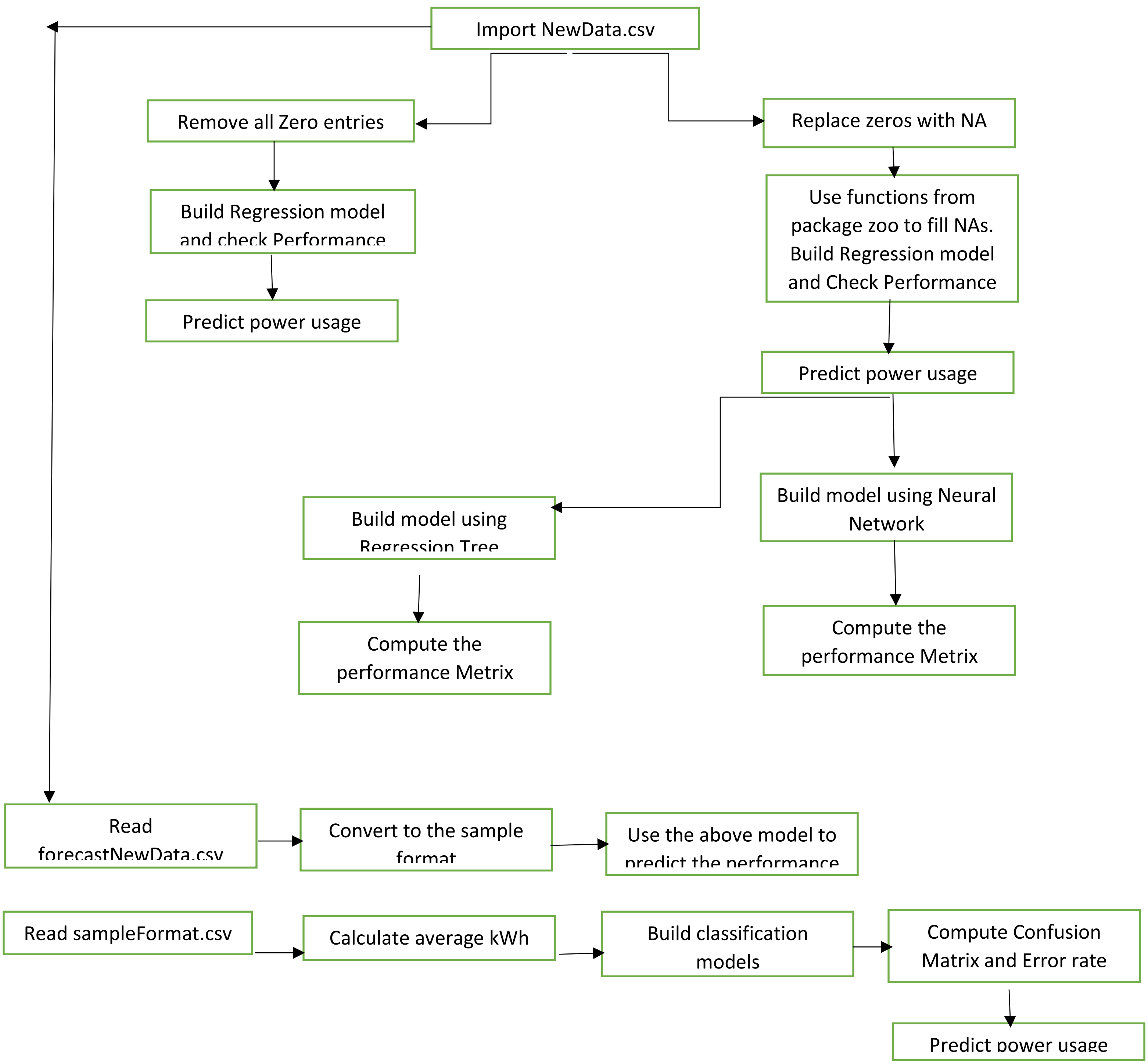
Assignment 2

Case 2: Energy Forecasting

FLOWCHART:-



Algorithm Implementation:-

Data wrangling and cleansing and Multiple linear regression :

1. Remove all the zero-entries and use only the non-zero entries for KWH to build a regression model

```
> acc
              Test set
ME        0.7367799085
RMSE     85.1428870865
MAE      64.4805899309
MPE   -4616.9470190566
MAPE   4662.0280100006
```

Model is as below:

	x
(Intercept)	-169676
Date	0.000122
month	-319.645
day	-10.9814
hour	-2.3304
PeakHour	84.73587
temperature	2.594253

b. Raw data including zero:

	Test set
ME	1.216494
RMSE	81.27946
MAE	58.6963
MPE	NA
MAPE	Inf

Model as below:

	x
(Intercept)	-231595
Date	0.000167
month	-435.408
day	-14.6289
hour	-1.89471
PeakHour	66.19491
temperature	2.422783

2. Use this model to predict power usage. Again build model on combined data. Check performance.
   a. Raw data including zero:

	Test set
ME	1.216494
RMSE	81.27946
MAE	58.6963
MPE	NA
MAPE	Inf

Model as below:

	x
(Intercept)	-231595
Date	0.000167
month	-435.408
day	-14.6289
hour	-1.89471
PeakHour	66.19491
temperature	2.422783

3. Using package zoo we replace zeros with NA and use functions na.approx, na.locf and na.fill to replace NAs.

Performance matrix after using functions na.approx or na.locf or na.fill to replace NAs

	Test set
ME	0.7951740765
RMSE	87.8805314670
MAE	60.3048688404
MPE	NaN
MAPE	Inf

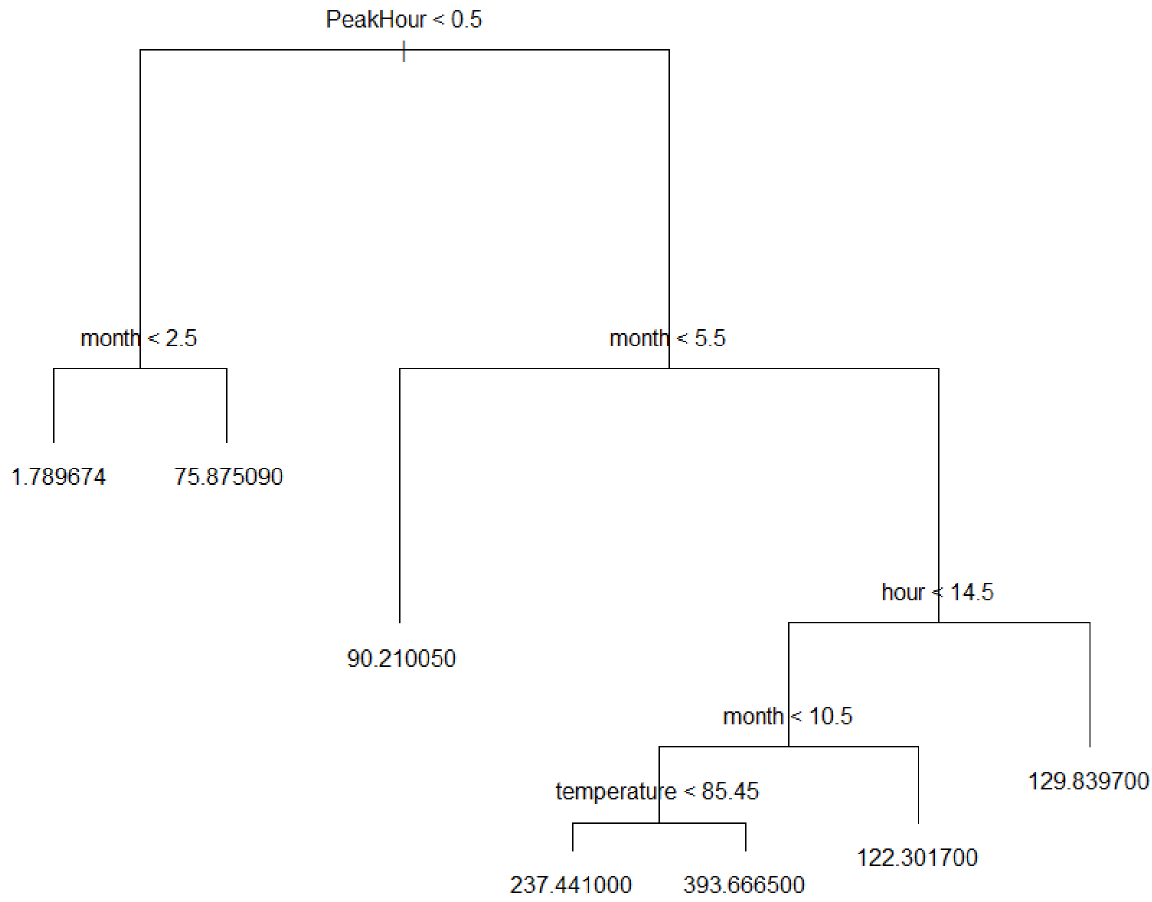
4. Selected the optimized model as combination of above 3 steps to get final data set for below models

Prediction:-

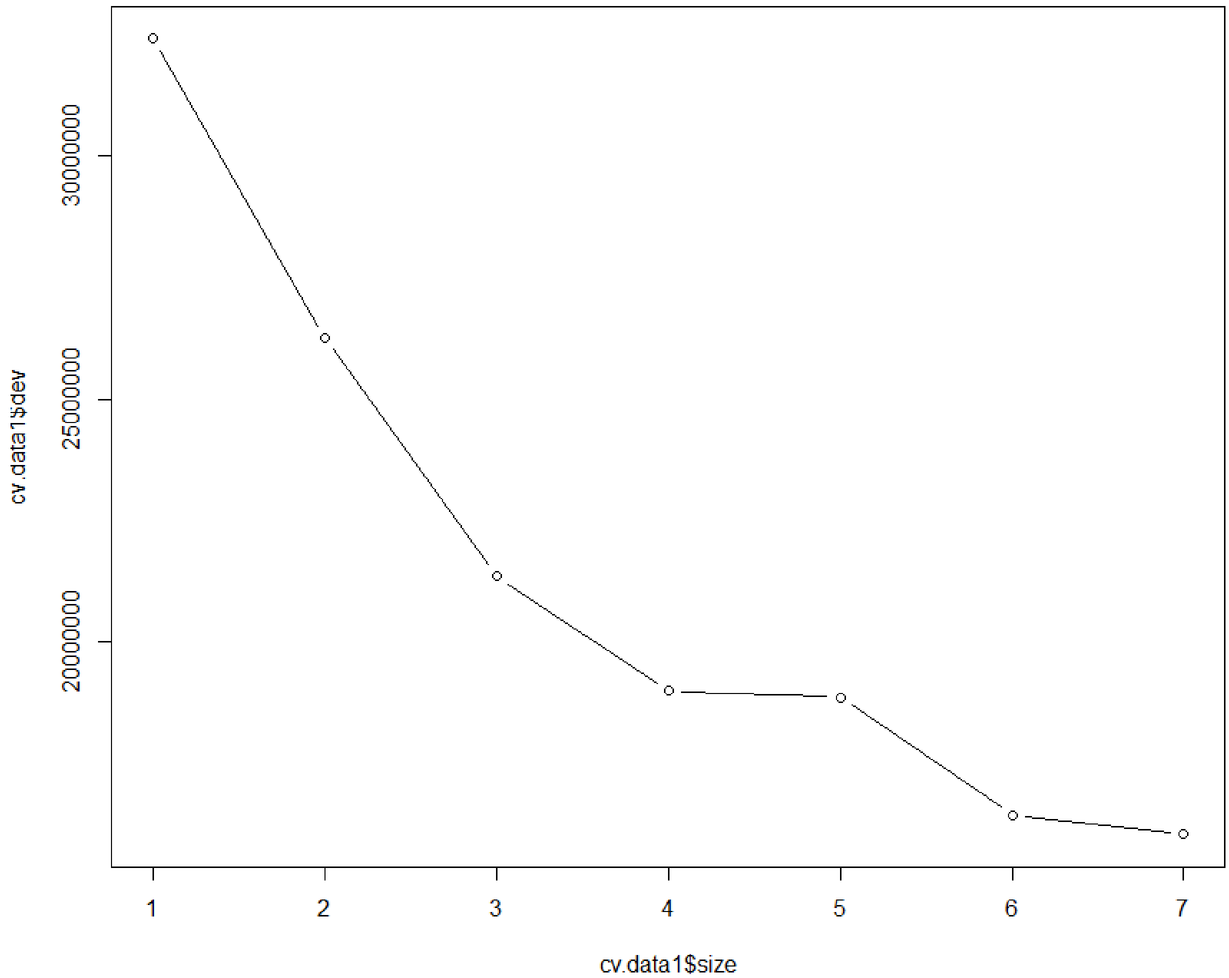
1. Regression Tree: Build a model using the tree() function and predict the power usage. Find performance value with the help of accuracy.

a. Tree model without pruning

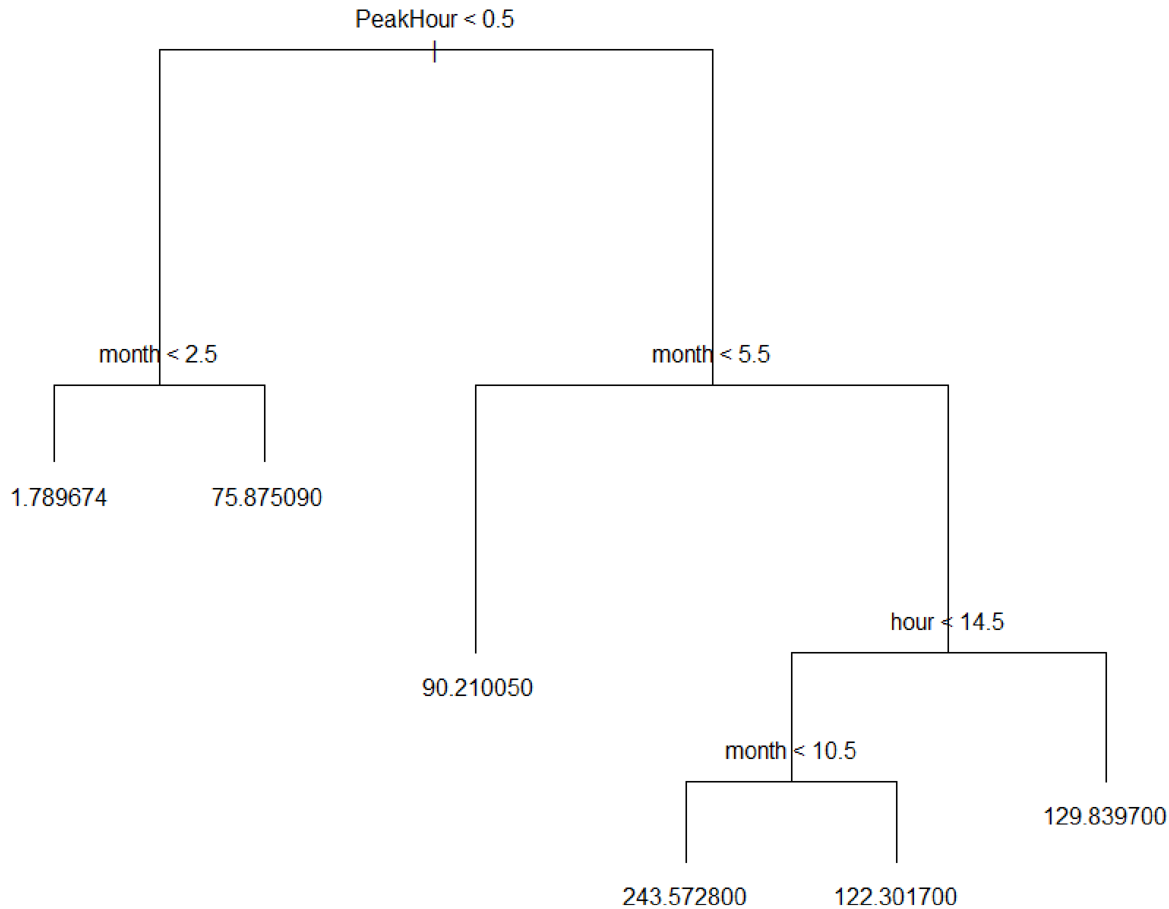
b.



b. Decide best size for prune tree:


cv.data1$dev
30000000
25000000
20000000
1
2
3
4
5
6
7
cv.data1$size

c. Prune tree:

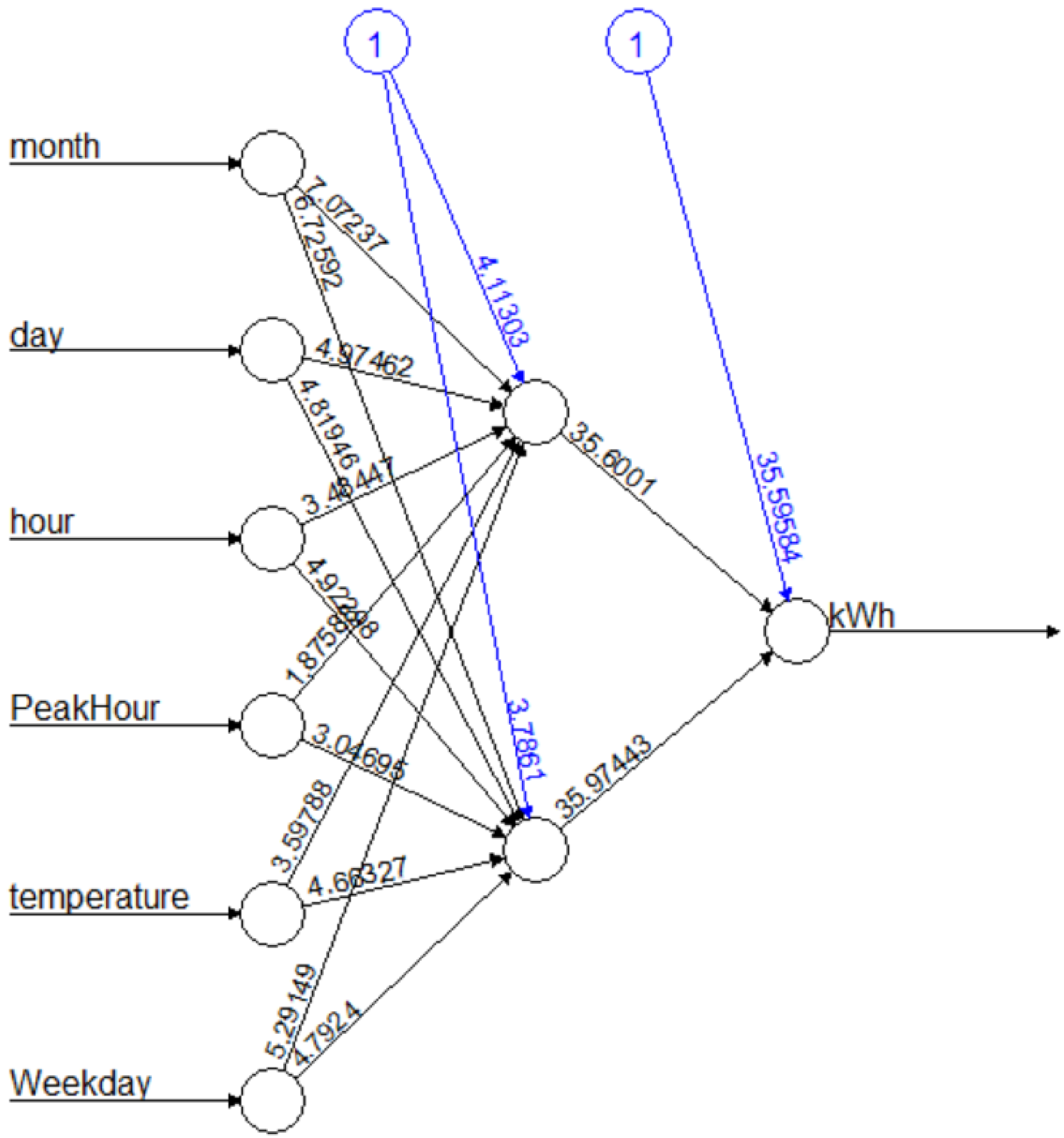


Tree	Test set
ME	1.89472185
RMSE	65.1483755
MAE	42.7838445
MPE	-3380.8634
MAPE	3417.8145

We have decided the tree based on performance matrix and we found that we get better result when we take the tree with prune of best 6

2. Neural Networks: Build a model using the neuralnet() function and predict the power usage. Find performance value with the help of accuracy.

a. Build neural network on our data, model is as below:



b. Performance matrix for above model:

	V1
Mean Error	4644.13401349
Root Mean Squared Error	92.39262125
Mean absolute Error	64.82760324
Mean Absolute Percentage Error	47.70612417
MPE	-4617.87269866

Forecast:-

1. Converted forecastNewData.csv and forecastNewData2.csv into forecastInput.csv format
2. Applied Tree model and Neural Network models generated in step 2 to forecast power
   a. a. After applying neural network model, below is power forecast for each hour for each hour for **forecastNewData.csv**:

	A	B	C	D	E	F	G	H	I	J	K	L	M
1		X	Date	hour	temperature	Year	month	day	Day.Of.Week	Weekday	PeakHour	kWh	
2	1	1	12/1/2014	0	51.98	2014	12	1	0	1	0	75.87508628	
3	2	2	12/1/2014	1	51.98	2014	12	1	0	1	0	75.87508628	
4	3	3	12/1/2014	2	51.08	2014	12	1	0	1	0	75.87508628	
5	4	4	12/1/2014	3	51.08	2014	12	1	0	1	0	75.87508628	
6	5	5	12/1/2014	4	51.08	2014	12	1	0	1	0	75.87508628	
7	6	6	12/1/2014	5	51.08	2014	12	1	0	1	0	75.87508628	
8	7	7	12/1/2014	6	50	2014	12	1	0	1	0	75.87508628	
9	8	8	12/1/2014	7	53.06	2014	12	1	0	1	1	122.3017428	
10	9	9	12/1/2014	8	55.94	2014	12	1	0	1	1	122.3017428	

   b. After applying tree model, below is power forecast for each hour for **forecastNewData.csv**:

	A	B	C	D	E	F	G	H	I	J	K	L	M
1		X	hour	temperature	Date	Year	month	day	Day.Of.Week	Weekday	PeakHour	kWh	
2	1	1	18	70	6/13/2016	2016	6	13	0	1	1	129.8397072	
3	2	2	19	70	6/13/2016	2016	6	13	0	1	1	129.8397072	
4	3	3	20	69	6/13/2016	2016	6	13	0	1	0	75.87508628	
5	4	4	21	67	6/13/2016	2016	6	13	0	1	0	75.87508628	
6	5	5	22	65	6/13/2016	2016	6	13	0	1	0	75.87508628	
7	6	6	23	63	6/13/2016	2016	6	13	0	1	0	75.87508628	
8	7	7	0	61	6/14/2016	2016	6	14	1	1	0	75.87508628	
9	8	8	1	60	6/14/2016	2016	6	14	1	1	0	75.87508628	
10	9	9	2	59	6/14/2016	2016	6	14	1	1	0	75.87508628	

   c. a. After applying neural network model, below is power forecast for each hour for each hour for **forecastNewData2.csv**:

	A	B	C	D	E	F	G	H	I	J	K	L	M
1		X	Date	hour	temperature	Year	month	day	Day.Of.Week	Weekday	PeakHour	kWh	
2	1	1	12/1/2014	0	51.98	2014	12	1	0	1	0	107.1704	
3	2	2	12/1/2014	1	51.98	2014	12	1	0	1	0	107.1704	
4	3	3	12/1/2014	2	51.08	2014	12	1	0	1	0	107.1704	
5	4	4	12/1/2014	3	51.08	2014	12	1	0	1	0	107.1704	
6	5	5	12/1/2014	4	51.08	2014	12	1	0	1	0	107.1704	
7	6	6	12/1/2014	5	51.08	2014	12	1	0	1	0	107.1704	
8	7	7	12/1/2014	6	50	2014	12	1	0	1	0	107.1704	
9	8	8	12/1/2014	7	53.06	2014	12	1	0	1	1	107.1704	
10	9	9	12/1/2014	8	55.94	2014	12	1	0	1	1	107.1704	
11	10	10	12/1/2014	9	60.08	2014	12	1	0	1	1	107.1704	
12	11	11	12/1/2014	10	60.08	2014	12	1	0	1	1	107.1704	
13	12	12	12/1/2014	11	62.06	2014	12	1	0	1	1	107.1704	

d. After applying tree model, below is power forecast for each hour for **forecastNewData2.csv**:

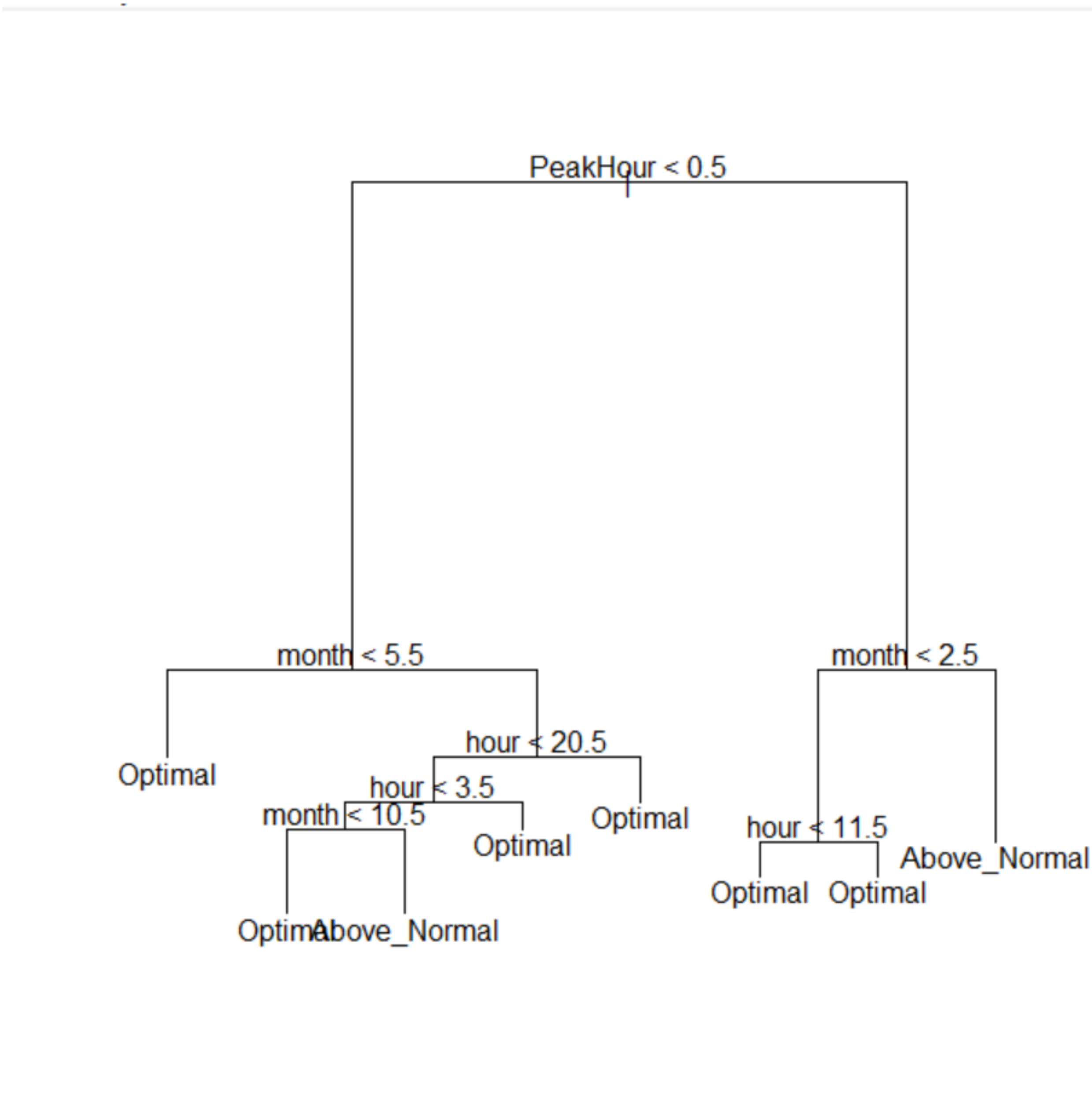
	A	B	C	D	E	F	G	H	I	J	K	L	M
1		X	Date	hour	temperatu	Year	month	day	Day.Of.W	Weekday	PeakHour	kWh	
2	1	1	12/1/2014	0	51.98	2014	12	1	0	1	0	75.87509	
3	2	2	12/1/2014	1	51.98	2014	12	1	0	1	0	75.87509	
4	3	3	12/1/2014	2	51.08	2014	12	1	0	1	0	75.87509	
5	4	4	12/1/2014	3	51.08	2014	12	1	0	1	0	75.87509	
6	5	5	12/1/2014	4	51.08	2014	12	1	0	1	0	75.87509	
7	6	6	12/1/2014	5	51.08	2014	12	1	0	1	0	75.87509	
8	7	7	12/1/2014	6	50	2014	12	1	0	1	0	75.87509	
9	8	8	12/1/2014	7	53.06	2014	12	1	0	1	1	122.3017	
10	9	9	12/1/2014	8	55.94	2014	12	1	0	1	1	122.3017	
11	10	10	12/1/2014	9	60.08	2014	12	1	0	1	1	122.3017	

Part 2: Classification:-

1. In the Hourly_filled_data.csv file, compute the average KWH and add a new column, KWH_Class.
2. If KWH > average KWH, KWH_Class = “Above_Normal” otherwise, KWH_Class = “Optimal”

i) Logistic Regression

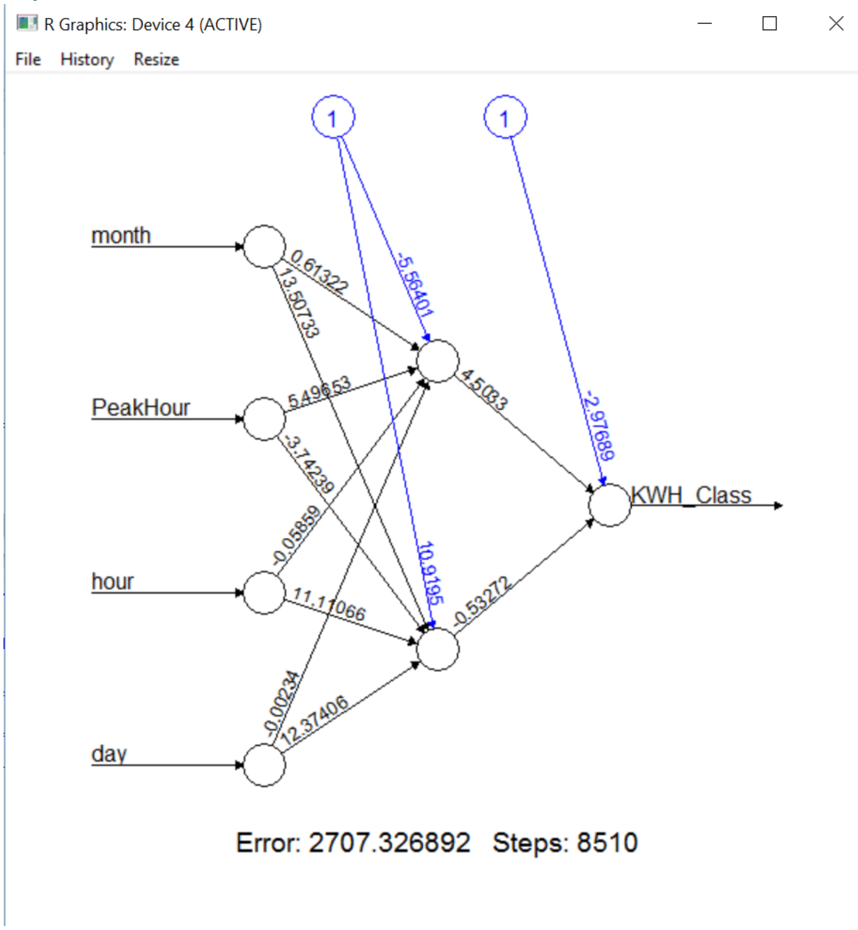
ii) Classification Tree




```
> Kwh<-ifelse(y>0.5,1,0)
> xx=table(Kwh,test.neural$KWH_Class)
> xx

Kwh     0    1
  0  1043  173
  1   244  544
>
```

iii) Neural Network



Neural net output:

```
> neuralnettt
Call: neuralnet(formula = KWH_Class ~ month + PeakHour + hour + day,     data = train.neural, hidden = 2, err.fct = "ce", linear.output =
 FALSE)

1 repetition was calculated.

        Error Reached Threshold Steps
1 2707.326892    0.007043119181   8510

> neuralnettt$result.matrix
                                       1
error                  2707.326892483754
reached.threshold         0.007043119181
steps                  8510.000000000000
Intercept.to.1layhid1    -5.564012030018
month.to.1layhid1         0.613223963113
PeakHour.to.1layhid1      5.496534265153
hour.to.1layhid1         -0.058585381711
day.to.1layhid1          -0.002339843055
Intercept.to.1layhid2    10.919503547549
month.to.1layhid2        13.507326365219
PeakHour.to.1layhid2     -3.742388076425
hour.to.1layhid2         11.110658239085
day.to.1layhid2          12.374063573123
Intercept.to.KWH_Class   -2.976889673148
1layhid.1.to.KWH_Class    4.503301866533
1layhid.2.to.KWH_Class   -0.532717470177
> 
```

Confusion Metrix for classification using regression tree and neural network:

	A	B	C	D
1		x		
2	1	Account No		
3	1	999999999		
4	x			
5	1	Classification Tree		
6				
7	Above_Normal	629	299	
8	Optimal	103	985	
9	x			
10	1	Neural Network		
11	Above_Normal	1043	173	
12	Optimal	244	544	
13				

3. **Classification Forecast:**
   a. **Forecast for Classification tree for forecastnewData.csv**

	A	B	C	D	E	F	G	H	I
1		X	hour	temperatu	Date	year	month	day	DayOfWeek
2	1	1	18	70	6/13/2016	2016	6	13	0
3	2	2	19	70	6/13/2016	2016	6	13	0
4	3	3	20	69	6/13/2016	2016	6	13	0
5	4	4	21	67	6/13/2016	2016	6	13	0
6	5	5	22	65	6/13/2016	2016	6	13	0
7	6	6	23	63	6/13/2016	2016	6	13	0
8	7	7	0	61	6/14/2016	2016	6	14	1
9	8	8	1	60	6/14/2016	2016	6	14	1
10	9	9	2	59	6/14/2016	2016	6	14	1
11	10	10	3	58	6/14/2016	2016	6	14	1
12	11	11	4	57	6/14/2016	2016	6	14	1
13	12	12	5	56	6/14/2016	2016	6	14	1
14	13	13	6	56	6/14/2016	2016	6	14	1
15	14	14	7	58	6/14/2016	2016	6	14	1

   b. **Forecast for classification tree for forecastnewData2.csv**

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1		X	hour	temperatu	Date	year	month	day	DayOfWeek	Weekday	PeakHour	Account	tree.pred	
2	1	1	18	70	6/13/2016	2016	6	13	0	1	1	222	Optimal	
3	2	2	19	70	6/13/2016	2016	6	13	0	1	1	222	Optimal	
4	3	3	20	69	6/13/2016	2016	6	13	0	1	0	222	Optimal	
5	4	4	21	67	6/13/2016	2016	6	13	0	1	0	222	Optimal	
6	5	5	22	65	6/13/2016	2016	6	13	0	1	0	222	Optimal	
7	6	6	23	63	6/13/2016	2016	6	13	0	1	0	222	Optimal	
8	7	7	0	61	6/14/2016	2016	6	14	1	1	0	222	Optimal	
9	8	8	1	60	6/14/2016	2016	6	14	1	1	0	222	Optimal	
10	9	9	2	59	6/14/2016	2016	6	14	1	1	0	222	Optimal	
11	10	10	3	58	6/14/2016	2016	6	14	1	1	0	222	Optimal	
12	11	11	4	57	6/14/2016	2016	6	14	1	1	0	222	Optimal	
13	12	12	5	56	6/14/2016	2016	6	14	1	1	0	222	Optimal	
14	13	13	6	56	6/14/2016	2016	6	14	1	1	0	222	Optimal	
15	14	14	7	58	6/14/2016	2016	6	14	1	1	1	222	Optimal	

c. Forecast for classification neural net for forecastnewData.csv

X	hour	temperatu	Date	year	month	day	DayOfWe	Weekday	PeakHour	Kwh
1	18	70	########	2016	6	13	0	1	1	Above_Normal
2	19	70	########	2016	6	13	0	1	1	Above_Normal
3	20	69	########	2016	6	13	0	1	0	Optimal
4	21	67	########	2016	6	13	0	1	0	Optimal
5	22	65	########	2016	6	13	0	1	0	Optimal
6	23	63	########	2016	6	13	0	1	0	Optimal
7	0	61	########	2016	6	14	1	1	0	Optimal
8	1	60	########	2016	6	14	1	1	0	Optimal
9	2	59	########	2016	6	14	1	1	0	Optimal
10	3	58	########	2016	6	14	1	1	0	Optimal
11	4	57	########	2016	6	14	1	1	0	Optimal
12	5	56	########	2016	6	14	1	1	0	Optimal
13	6	56	########	2016	6	14	1	1	0	Optimal
14	7	58	########	2016	6	14	1	1	1	Above_Normal
15	8	61	########	2016	6	14	1	1	1	Above_Normal
16	9	64	########	2016	6	14	1	1	1	Above_Normal
17	10	67	########	2016	6	14	1	1	1	Above_Normal
18	11	69	########	2016	6	14	1	1	1	Above_Normal
19	12	71	########	2016	6	14	1	1	1	Above_Normal
20	13	73	########	2016	6	14	1	1	1	Above_Normal
21	14	75	########	2016	6	14	1	1	1	Above_Normal

forecastOutput1_999999999_Neura

d. Forecast for classification neural net for forecastnewData2.csv

X	Date	hour	temperatu	year	month	day	DayOfWe	Weekday	PeakHour	Kwh
1	12/1/2014	0	51.98	2014	12	1	0	1	0	Above_Normal
2	12/1/2014	1	51.98	2014	12	1	0	1	0	Above_Normal
3	12/1/2014	2	51.08	2014	12	1	0	1	0	Above_Normal
4	12/1/2014	3	51.08	2014	12	1	0	1	0	Above_Normal
5	12/1/2014	4	51.08	2014	12	1	0	1	0	Above_Normal
6	12/1/2014	5	51.08	2014	12	1	0	1	0	Above_Normal
7	12/1/2014	6	50	2014	12	1	0	1	0	Above_Normal
8	12/1/2014	7	53.06	2014	12	1	0	1	1	Above_Normal
9	12/1/2014	8	55.94	2014	12	1	0	1	1	Above_Normal
10	12/1/2014	9	60.08	2014	12	1	0	1	1	Above_Normal
11	12/1/2014	10	60.08	2014	12	1	0	1	1	Above_Normal
12	12/1/2014	11	62.06	2014	12	1	0	1	1	Above_Normal
13	12/1/2014	12	62.96	2014	12	1	0	1	1	Above_Normal
14	12/1/2014	13	60.98	2014	12	1	0	1	1	Above_Normal
15	12/1/2014	14	57.92	2014	12	1	0	1	1	Above_Normal
16	12/1/2014	15	55.04	2014	12	1	0	1	1	Above_Normal
17	12/1/2014	16	53.06	2014	12	1	0	1	1	Above_Normal
18	12/1/2014	17	51.08	2014	12	1	0	1	1	Above_Normal
19	12/1/2014	18	48.92	2014	12	1	0	1	1	Above_Normal
20	12/1/2014	19	48.02	2014	12	1	0	1	1	Above_Normal
21	12/1/2014	20	46.94	2014	12	1	0	1	0	Optimal