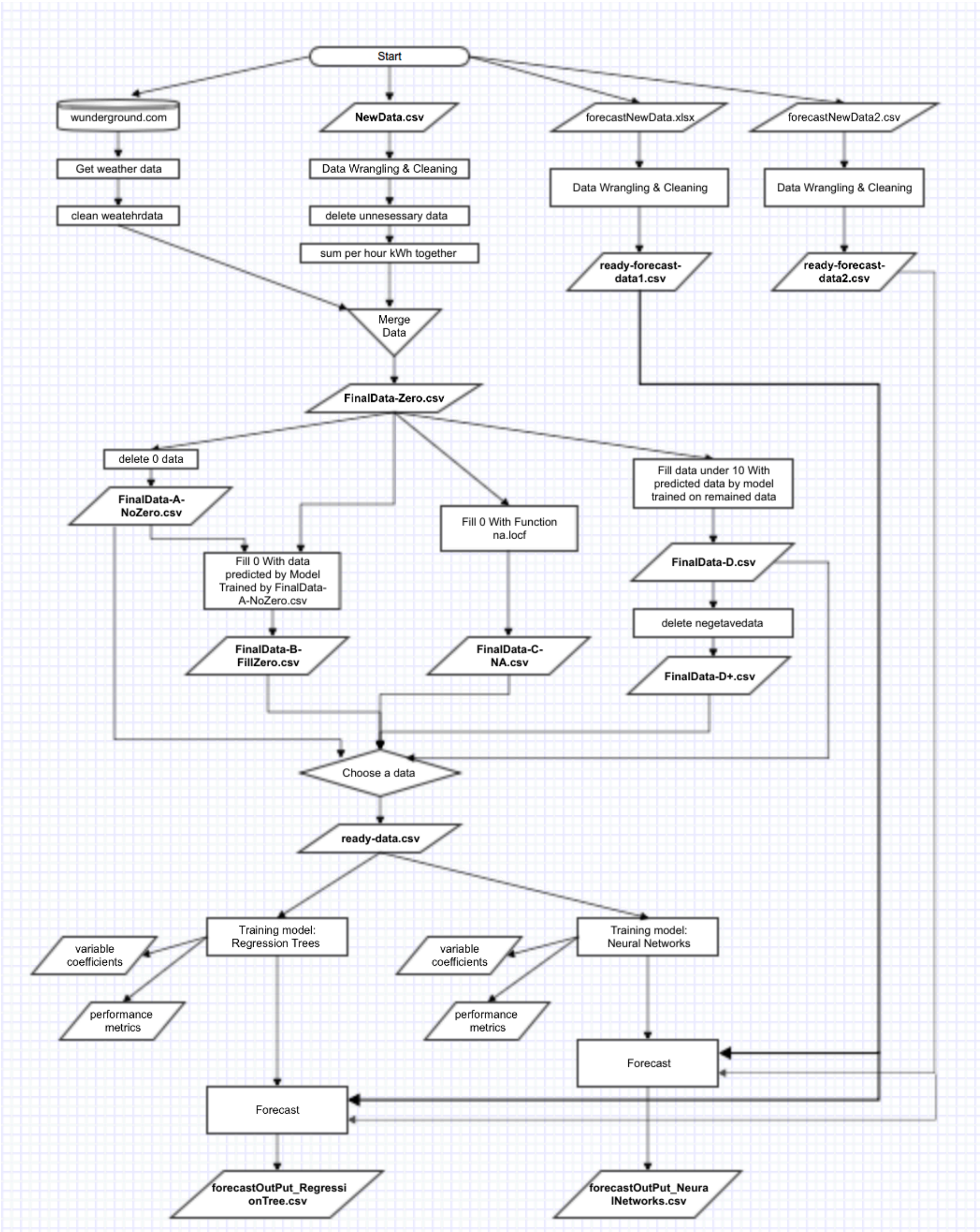
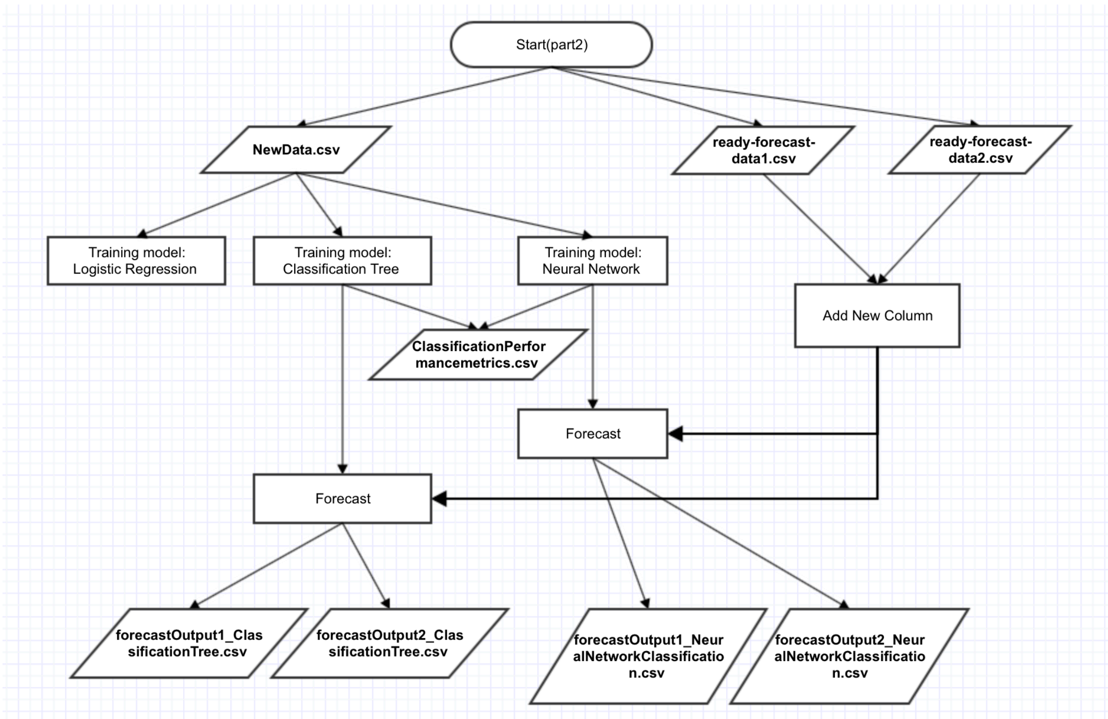
Report

1. Flow chart

A. Part 1



B. Part 2



2. Data wrangling and cleansing

A. Handle NewData.csv

a. Not handle the zeroes

We build model with all the entries. Here are the accuracy measures of the model it delivers: RMSE is 77.427, MAE is 57.091 and the MAPE is very large.

b. Remove all the zero-entries

We remove all the zero entries and use only the non-zero data. By this way, the model's RMSE is 82.83, MAE is 63.348, and MAPE is 5219.579.

c. Fill all the zero-entries

We build a model using the non-zero data, and then predict the zero-entries. By this way, RMSE is 65.05, MAE is 39.071, MAPE is 3219.261.

d. Replace the zeroes with NA

We use the function na.approx, na.fill, na.locf to fill 0. In this way, RMSE is 57.491, MAE is 41.156, and MAPE is very large.

e. Our further exploration and final choice

Among the above models, **c** has the best results.

However, when we look into the column “kWh” in the “NewData.csv”, we notice that many entries should also be handled though they are not 0. For example, in 4/22 17:05-18:00, the kWh data is 0.06, 0, 0, 0.09, 0, 0, 0, 0.09, 0.165, 0.12, 0, 0. This is obviously abnormal. So we think that we should clean these data too.

Our decision is to consider all the kWh entries that are smaller than 10 per hour also as abnormal. Then we delete all the abnormal data, use the remaining data to build a new model which are used to make predictions to fill all the abnormal entries. (A combination of method **b** and **c**)

Then we find that there are all still some abnormal (predicted) entries which are smaller than zero. Obviously it is impossible for kWh be negative. So we decide to delete them (method b).
Finally, we build a model using such data whose accuracy measures are: RMSE = 66, MAE = 41, MAPE = 49. This one is the best and thus our final choice.

rowname	Zero	NoZero	FillZero	NA	D	D+
RMSE	77.427	82.83	65.05	57.491	62.329	66.373
MAE	57.091	63.348	39.071	41.156	36.539	41.434
MAPE	Inf	5219.579	3219.261	Inf	42.987	48.746

We name the finally-cleaned file “FinalData-D-NonNegative-final/ReadyData”.

B. Handle forecastNewData.xlsx and forecastNewData2.csv

We first change forecastNewData2.xlsx to a csv file, and modify the two csv files to be of the correct format. Our cleaned files are ready-forecast-data1.csv and ready-forecast-data2.csv, which are used to do the predictions.

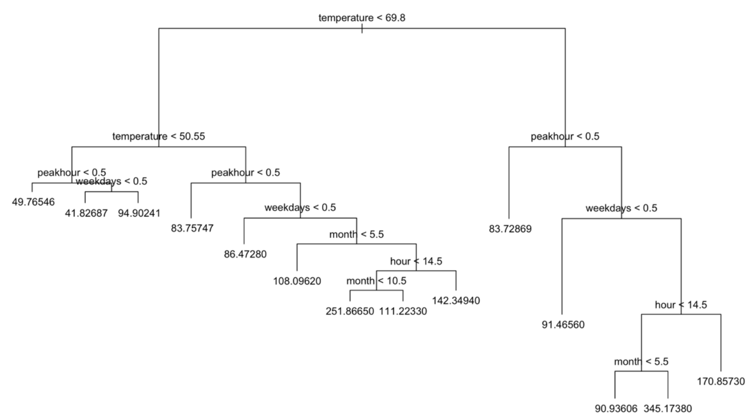
3. Multiple linear regression

A. Regression tree

```
set.seed(1)

training = sample(1:nrow(mydata),nrow(mydata)/2)

testing = mydata[-training,]

tree.mydata = tree(kWh~month + day + hour + DayofWeek + weekdays + peakhour + temperature,mydata,subset = training)
```

```
Variables actually used in tree construction:
[1] "temperature" "peakhour"    "weekdays"    "month"       "hour"
Number of terminal nodes:  14
Residual mean deviance:  2586.406 = 9104148 / 3520
Distribution of residuals:
       Min.     1st Qu.      Median        Mean     3rd Qu.        Max.
-292.928800  -23.533650   -6.206048    0.000000   22.985560  306.962500
```

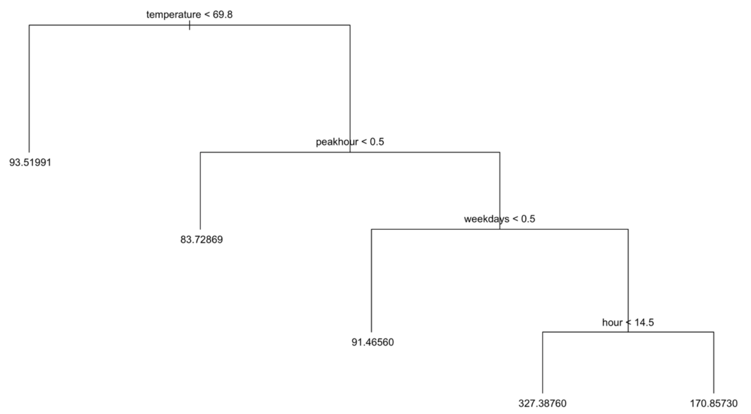
```
$size
 [1] 14 12 11 10  8  7  6  5  4  2  1

$dev
 [1]  9184312.381 10155461.242 10272740.639 10970323.654 14033028.566 14034564.963 16729856.864 16729856.864 18869124.675 25203793.105
[11] 29642476.649

$k
 [1]         -Inf   372637.8395   404644.0572   735669.8096   994820.8646 1021953.6313 1323676.4666 1362820.8202 2146545.2526 3158706.4989
[11] 4468924.4077

$method
[1] "deviance"

attr(,"class")
[1] "prune"         "tree.sequence"
```



Assignment 2_Team 4

tree.mydata = prune.tree(tree.mydata,best = 5)

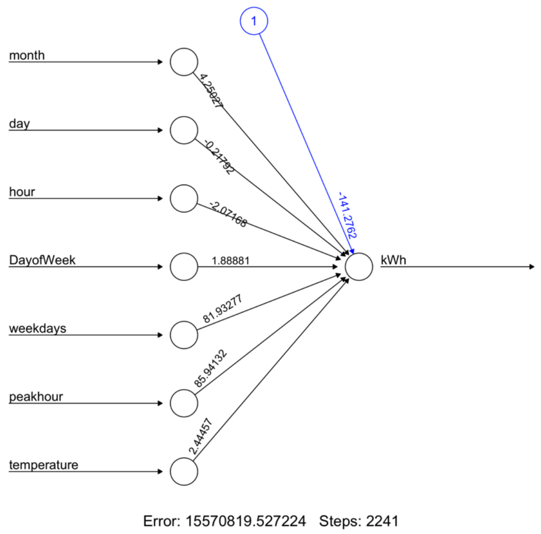
```
Number of terminal nodes:  5
Residual mean deviance:  4728.77 = 16687830 / 3529
Distribution of residuals:
      Min.    1st Qu.     Median       Mean    3rd Qu.       Max.
-306.11760  -34.65844  -15.13299    0.00000   26.97962  355.97010
```



B. Neural network

net.mydata <- neuralnet(kWh~month + day + hour + DayofWeek + weekdays + peakhour + temperature,mydata,hidden = 0)

```
1 repetition was calculated.

        Error Reached Threshold Steps
1 15570819.53    0.006794698634  2241
```

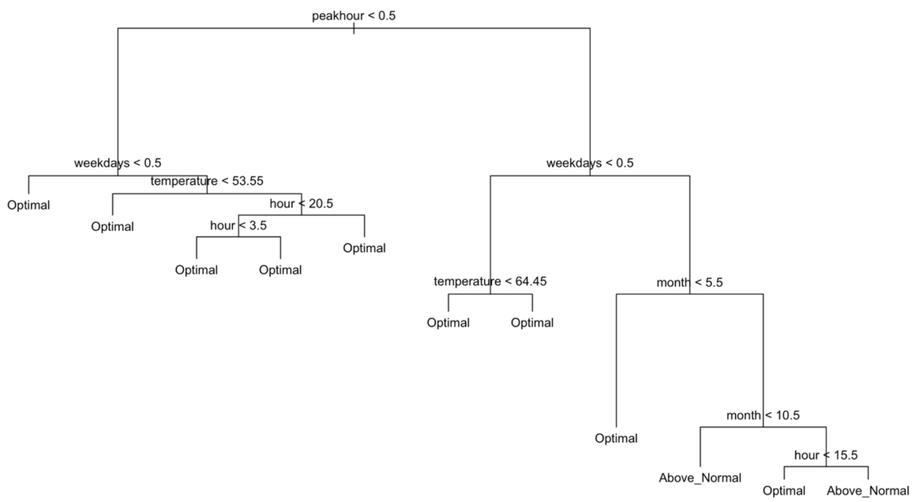


Assignment 2_Team 4

4. Classification

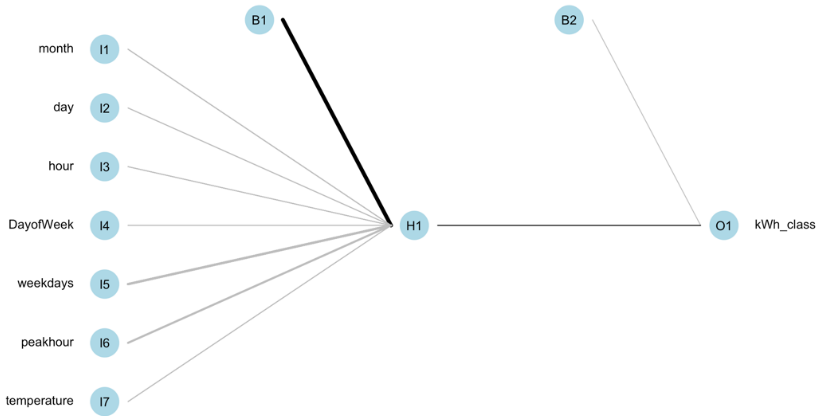
A. Classification tree

```
attach(mydata)

mkWh = mean(kWh)

kWh_class = ifelse(kWh>mkWh,"Above_Normal","Optimal")

mydata = data.frame(mydata,kWh_class)

mytree = tree(kWh_class~month + day + hour + DayofWeek + weekdays + peakhour + temperature,mydata)
```



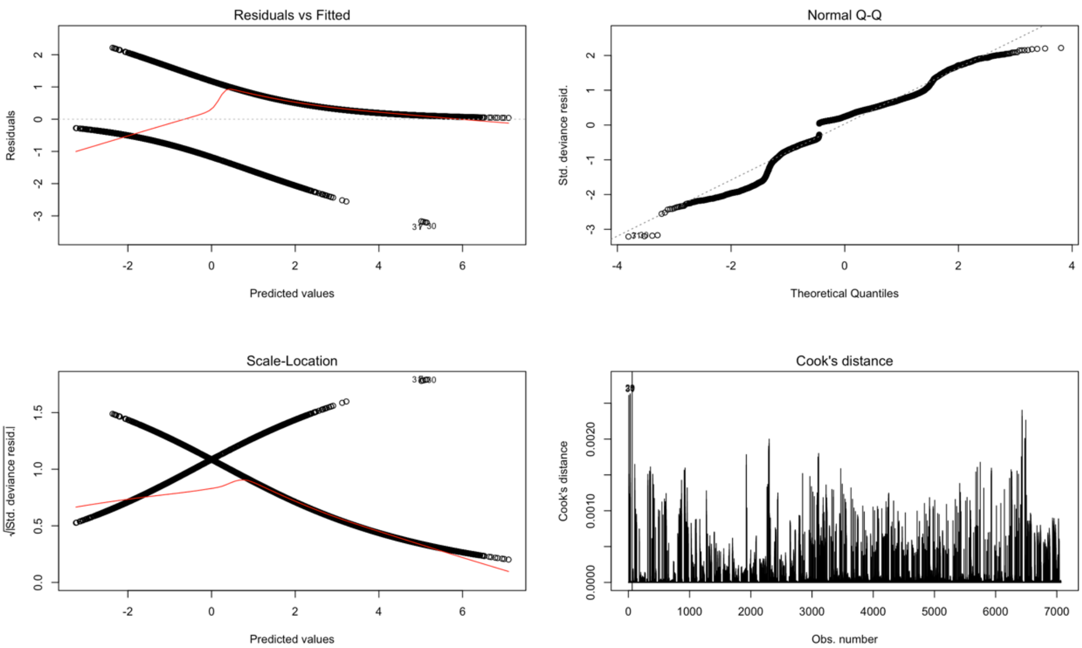
B. Neural network

```
nn = nnet(kWh_class~month + day + hour + DayofWeek + weekdays + peakhour + temperature,mydata,size=1)
```



C. Logistic regression

lg = glm(kWh_~ month + day + hour + DayofWeek + weekdays + peakhour + temperature, family=binomial(link='logit'), mydata)



5. Evaluation

A. Prediction performance metrics

Mean absolute error (MAE)

The MAE measures the average magnitude of the errors in a set of forecasts, without considering their direction. It measures accuracy for continuous variables. MAPE is the percentage of MAE

Mean absolute percentage error(MAPE)

One problem with the MAE is that the relative size of the error is not always obvious. Mean Absolute Percentage Error (MAPE) allows us to compare forecasts of different series in different scales.

Root mean squared error (RMSE)

The RMSE is a quadratic scoring rule which measures the average magnitude of the error. It gives a relatively high weight to large errors, which means the RMSE is most useful when large errors are particularly undesirable.

The MAE and the RMSE can be used together to diagnose the variation in the errors in a set of forecasts. The greater difference between them, the greater the variance in the individual errors in the sample. If the RMSE=MAE, then all the errors are of the same magnitude. Both the MAE and RMSE can range from 0 to ∞. They are negatively-oriented scores: lower values are better.

Neural Network	
MAE	41.43
MAPE	72.38 %
RMSE	66.37

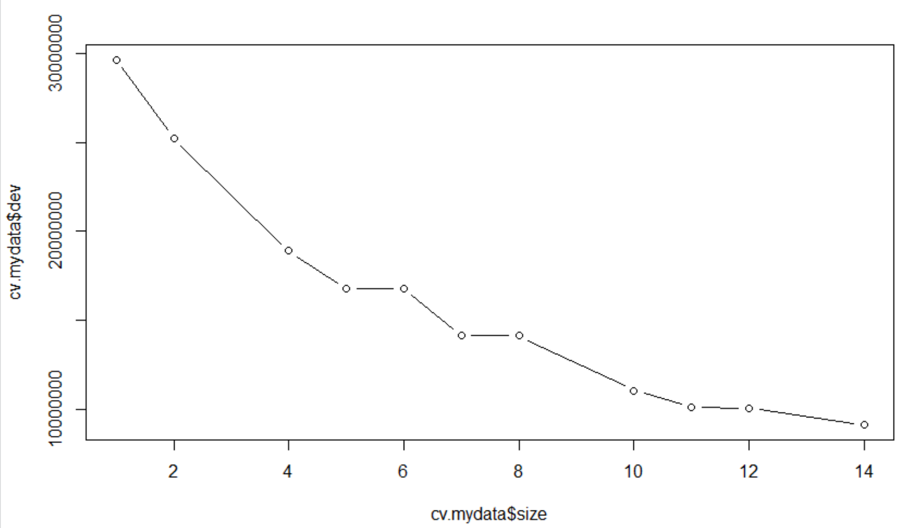
To build the neural network model, we use “neuralnet()”. We delete all the irrelative columns in the model including “Account”, “Data”, “kWh” which are what we need to predict and “Year” which is same for all records. We set “hidden value” as 0, which might increase error rate but if we change it into c(4,4) or another value, there will be runtime errors which we can’t figure out why.
We use the function “accuracy()” to compare the real values and predict values. As shown above, the RMSE is higher than MAE, which means there is variation in the errors. MAPE = 72% means differences between the predicted value and real values is

72% of real values, which is not very good. One reason for it is that the value of the argument “hidden value” is not that prefect.

To conclude, the neural network model can make predictions promisingly and delivers better results than the regression tree model.

Regression Tree	
MAE	49.5
MAPE	46.3 %
RMSE	70.89

We use “tree()” to build the regression tree model. After comparing with the real values, we noticed that the MAPE is pretty large due to over fitted, so we decide to do pruning for the tree model. Firstly, we use “cv.tree()” to check whether we need to prune, the result is below.



It should work after pruning, so we use “prune.tree()” to select best five variables in our model. Then the MAPE becomes smaller, which is 46%.

```
> prune.tree(tree.mydata,best = 5)
node), split, n, deviance, yval
      * denotes terminal node

 1) root 3534 29620000 111.90
   2) temperature < 69.8 2787 12410000  93.52 *
   3) temperature > 69.8 747 12740000 180.60
     6) peakhour < 0.5 208   243200  83.73 *
     7) peakhour > 0.5 539  9792000 218.00
      14) weekdays < 0.5 159   138900  91.47 *
      15) weekdays > 0.5 380  6042000 271.00
        30) hour < 14.5 243  3536000 327.40 *
        31) hour > 14.5 137   359200 170.90 *
> accuracy(testing$kWh,tree.pred)
                ME     RMSE      MAE       MPE     MAPE
Test set -1.238209 70.88582 49.50029 -1.191346 46.29618
```

B. Classification performance metrics

Classification Tree		
Overall Error	17.27 %	
	Above Normal	Optimal
Above Normal	1482	405
Optimal	781	4201

Neural Network		
Overall Error	14.75 %	
	Above Normal	Optimal
Above Normal	1566	300
Optimal	743	4460

The predicted values are separated into two classes: above normal and optimal. For these two matrixes, the second column of first row and the first column of second row are the number of error values, we can get overall error rate by calculating the number of error divided by all the records. Neural Network is around 15%, Classification Tree is 17%, which are acceptable.