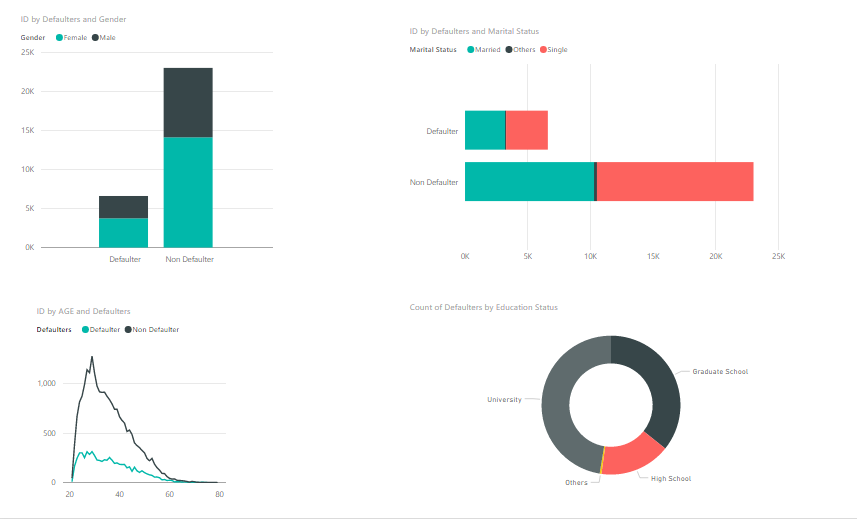
**Midterm Report**

**Problem 1**

**Tasks:**

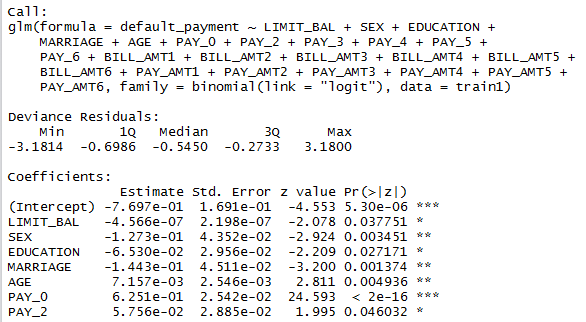
1. **Use POWER BI to explore the data. Summarize your observations**



1. **Clean and pre-process the data if needed** 
   1. Columns such as education and marriage has anomalies. Those values are cleansed and processed in power BI
   2. Sanity checks for null and 0 values are performed
   3. Reading the Excel file
   4. Performing sanitization (checking 0) using sqldf
   5. Performing sanitization (checking NULL) using sqldf
   6. Changing the column values accordingly
2. **Use Logistic regression, Neural Network and Classification trees to build classification models**

**Logistic Regression:**

1. Taking the sample
2. Set the seed to make your partition reproducible
3. Split the data into training and testing
4. Construct a logistic regression model using the variables
5. Summary of the fit
6. Measures of prediction
7. Find the overall error rate and confusion matrix
8. Plot the ROC as well as the lift curve



Data with max number of \*’s is an important factor for prediction

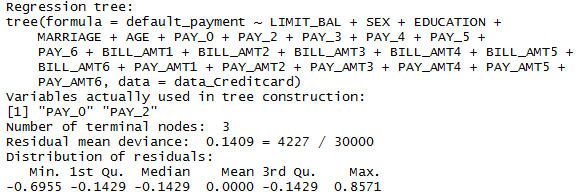
**Neural Network:**

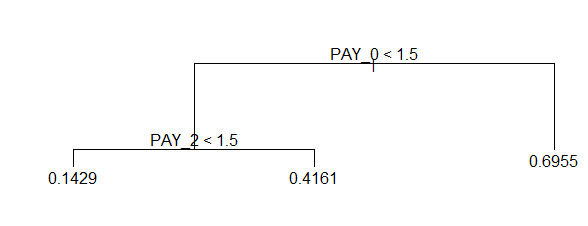
1. Repeat the steps similar to logistic regressions
2. Find the overall error rate and confusion matrix
3. Plot the ROC and lift curves



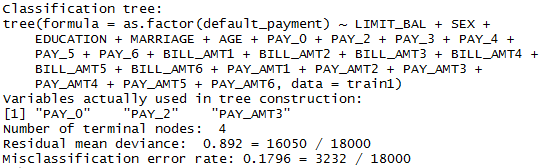
**Classification Trees:**

1. Repeat the steps similar to logistic regressions
2. Plot the tree





1. Split the dataset into a training set and a test set
2. Build the tree based on the training set
3. Evaluate its performance on the test data



1. Plot the ROC curve and lift curve

#ROC curve

install.packages("ROCR")

library(ROCR)

prediction <- prediction(tree.pred, payment.test)

performance <- performance(prediction, measure = "tpr", x.measure = "fpr")

plot(performance, main="ROC curve", xlab="1-Specificity", ylab="Sensitivity")

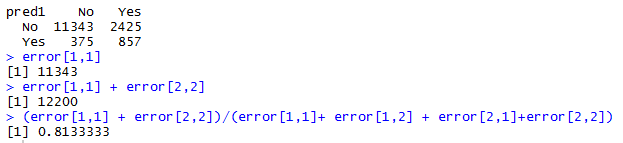
#Lift curve

perf <- performance(prediction,"lift","rpp")

plot(perf, main="lift curve", colorize=T)

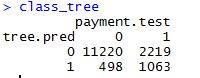
1. **Summarize the performance metrics**
   1. **Overall Error**

**Logistic Regression**



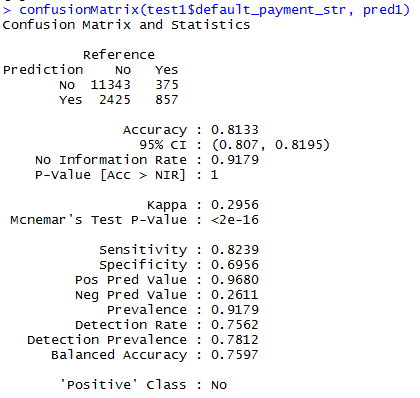
**Neural Networks**

**Classification Trees**

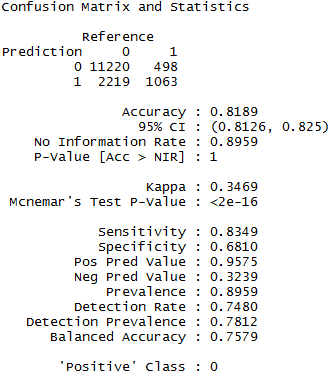


* 1. **Confusion matrix**

**Logistic Regression**



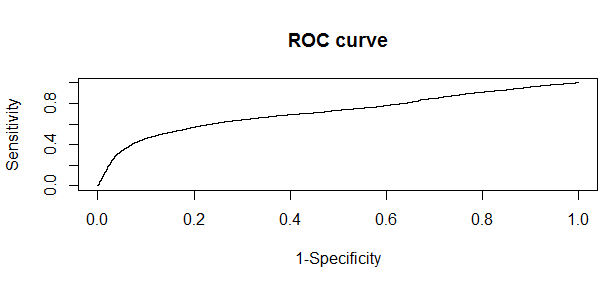
**Classification Trees**



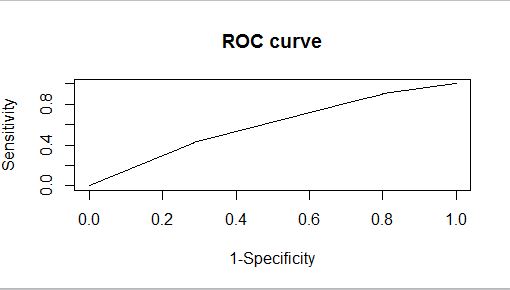
* 1. **ROC curves**

**Logistic Regression**

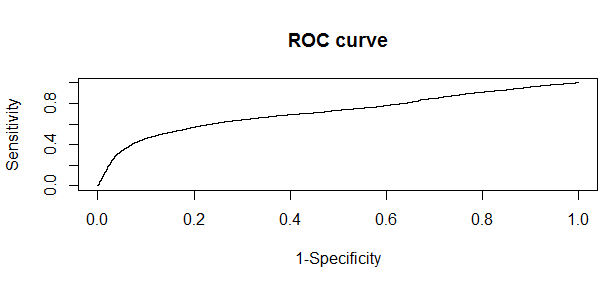
plot(performance, main="ROC curve", xlab="1-Specificity", ylab="Sensitivity")



**Neural Networks**

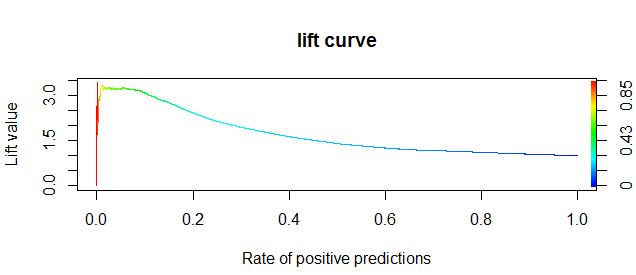


**Classification Trees**



* 1. **Lift charts**

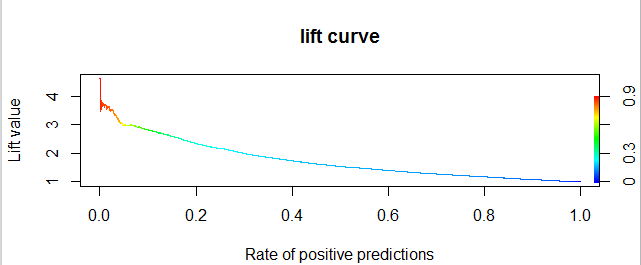
**Logistic Regression**



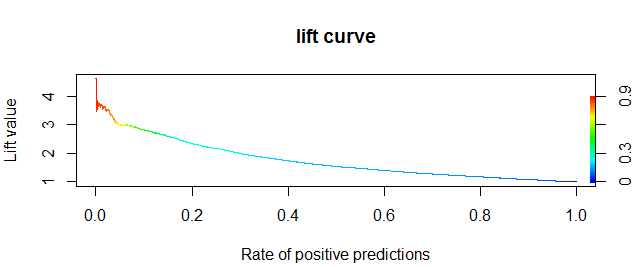
perf <- performance(prediction,"lift","rpp")

plot(perf, main="lift curve", colorize=T)

**Neural Networks**



**Classification Trees**



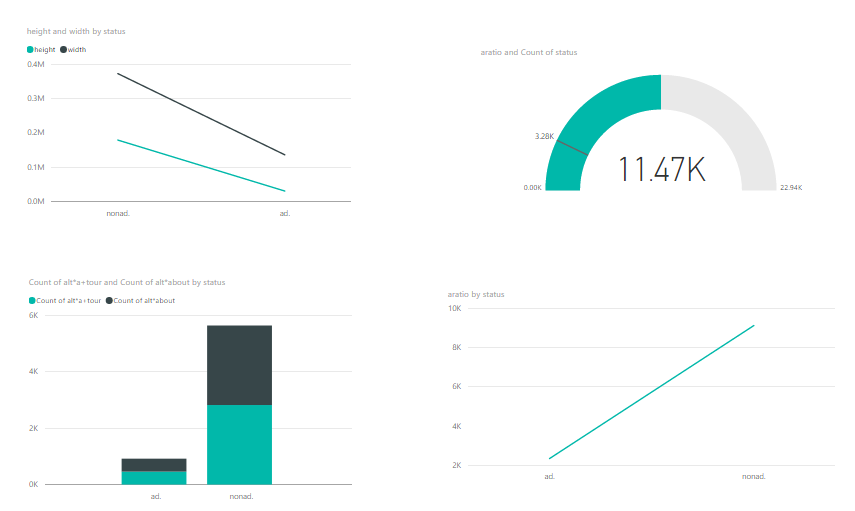
1. **Which model would you choose? Discuss**

From our analysis, we felt that classification model of this problem is the best predictive model as it has higher accuracy, less error rate and higher specificity.

**Problem 2**

**Tasks:**

1. **Use POWER BI to explore the data. Summarize your observations**

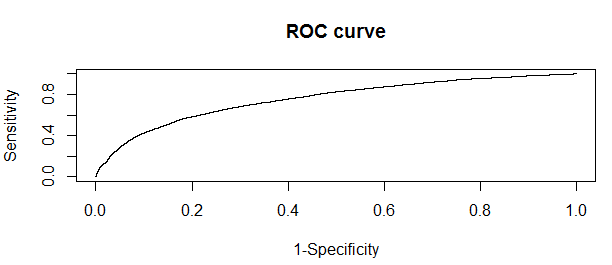


1. **Clean and pre-process the data. (Note: 28% of the continuous data is missing). Don’t just delete records. Think of a strategy to fill data**
2. Read the ad.data file
3. Read the ad.names file
4. Removing 1st 2 rows
5. Removing the unwanted texts that starts with "|"
6. Binding the columns
7. replace ? with NA
8. separating heightcolumn and converting to numeric for mean
9. Find the mean
10. Replace NA with mean of column
11. separating width and converting to numeric for mean
12. Replace NA with mean of column (width)
13. Remove local column NA's
14. **Use Logistic regression, Neural Network and Classification trees to build classification models**

**Logistic Regression**

1. Taking the sample data
2. Set the seed to make your partition reproductible
3. Split the data into training and testing
4. Fitting data into model
5. Predicting values
6. Error and Confusion Matrix
   1. **ROC curves**

**Logistic Regression**



1. **Which model would you choose? Discuss**

From our analysis, we felt that Neural Network tree model of this problem is the best predictive model as it has higher accuracy, less error rate and higher specificity.

**Problem 3**

**Goals:**

Following are the steps for removing zero entries:

1. We are performing date manipulation in the dataset.
2. First, we find that the first column contains the date and hour data together. This needs to be separated and converted to date format. We use substring to separate the date and hour data and create a new column with date and a column with hour.
3. ("select date,hors,u,v,ws,wd,dt,hr,

case when (hors+hr)==24 then '1.00'

when (hors+hr)==0 then '0.00

when (hors+hr)==23 then '0.23

when (hors+hr)<23 and (hors+hr)%23!=0 then (case when

substr(substr('0.'||(hors+hr),1,4),4,1)=='.' then substr('0.'||(hors+hr),1,3)

else substr('0.'||(hors+hr),1,4) end)

when (hors+hr)==48 then '2.00'

when (hors+hr)==47 then '1.23'

when ((hors+hr)>=25 and (hors+hr)<=46) then (case when substr(substr('1.'||((hors+hr)%24),1,4),4,1)=='.' then substr('1.'||((hors+hr)%24),1,3)

else substr('1.'||((hors+hr)%24),1,4) end)

when (hors+hr)==71 then '2.23'

when (hors+hr)==72 then '3.00'

when ((hors+hr)>=49 and (hors+hr)<=70) then (case when substr(substr('2.'||((hors+hr)%24),1,4),4,1)=='.' then substr('2.'||((hors+hr)%24),1,3)

else substr('2.'||((hors+hr)%24),1,4) end) end as tot\_hr from wf1")

1. We need to perform date manipulation by adding the hour data which was appended in the date with the hour data in the hour column. For this we do the following:

* When this total hour value is equal to 24, we add one day to the date. For this a new column is created which stores the value as 1.00
* When the total hour value is equal to 23, we add 23 hours to the date. For this a new column is created which stores the value as 0.23
* When the number of hours is less than twenty three. For this, new column has a value of 0, appended by substring value(23)
* When the total number of hours is equal to 48, the new column takes the value of 2.00
* When the total number of hours is equal to 47, we give a value of 1.23(one day and 23 hours)
* When we have a value between 25 and 46, we give a value of one appended by (total hours)%24
* When number of hours is 71, we assign a value of 2.23 (Two days and 23 hours)
* When number of hours is 72,the value assigned is 3.00 (three days)
* When hours between 49 and 70, we assign 2 appended by (totalhours)/24.

1. sqldf("select date,hors,u,v,ws,wd,dt,hr,tot\_hr,

substr(tot\_hr,1,1) as tt,substr(tot\_hr,3,2) as hh,hors+hr as net\_hr from wf2")

We then separate the hour and day value from the column we created in the earlier step.

6) We add the day value to the date value (to get the new date)

7) We then append the hour value to the date value to get the date as in the format required. Here, if the length of the hour column is one, we append a zero to the number before appending to the date column.

8) merging<-merge(x = wf3[,c("upd\_dt","hors","u","v","ws","wd","tot\_hr")], y = wind1[ , c("date", "wp1")], by.x ='upd\_dt',by.y = 'date', all.x=TRUE)

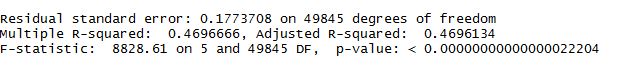
A left join is performed here to join all the columns in the data frame created earlier to the wp1 value in the train data file.

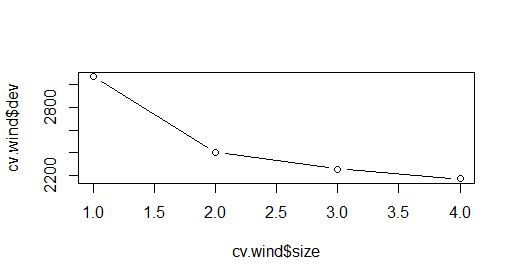
9) We next fill the NA values in the wp1 column (joined to original file) with na.locf.

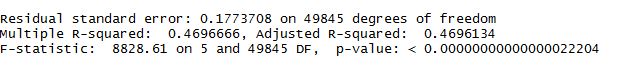
10) We use this dataset and the columns u,v,ws,wd and wp1 to predict the power values. Hence, we are using the earlier generated wind power and the environmental factors to predict the future wind power generation values.

Linear Regression Outputs:

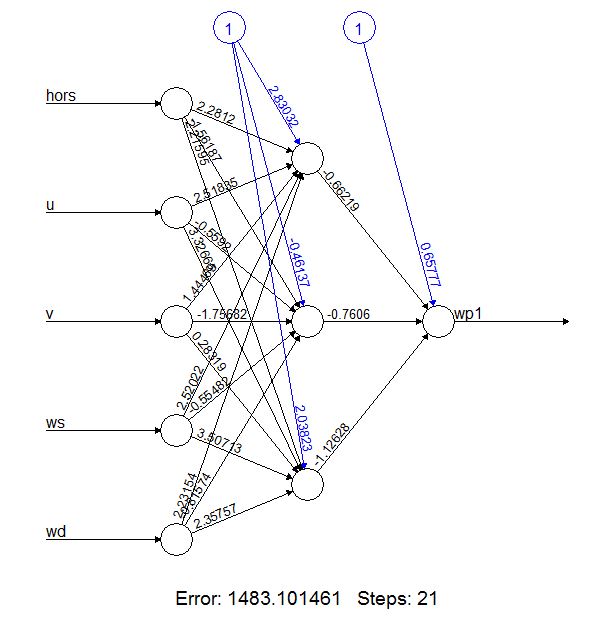
C:\Users\harikrishna\Downloads\log_regression.JPG



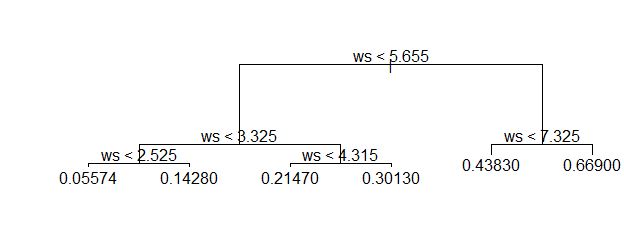
Deviance produced by cross validation



Neural Network:



**Classification Tree:**



**Tree Accuracy:**

