For this report, I will be using a modified version of the German Credit Data, which comprises information on 1000 customers, described by 20 decision attributes and a class attribute. But in the data, there are missing values. I removed rows that contains missing values. Now the data comprises information on 812 customers. Each instance is classified as ‘Good credit risk’ (GCR) or ‘Bad credit risk’ (BCR). The proportion of ‘Good credit risk’ in the data is 70% and the proportion of ‘Bad credit risk’ in the data is 30%.

Upon analysing the numerical variables of GCR data and BCR data separately using descriptive statistics, I found that on average GCR data has a lower duration (19.26 months) as compared to BCR data (24.81 months). Duration refers to the time taken to pay back the amount. Since mean is susceptible to outliers, I looked at the median. Similar pattern holds for median as well. Therefore, we can conclude that the longer the duration, the higher the chance of customer having BCR. This indeed makes sense, because the longer the duration, the higher the chance of defaulting which is why it is classified under ‘Bad credit risk’.

I further analysed categorical variables and felt that it would be interesting to analyse Duration against Employment. Employment refers to the number of years each customer is employed.

In figure 1, we can tell that, customers who are employed for more than 4 years (A74 and A75) have a higher median as compared to those who are employed for less than 4 years (A71, A72 and A73). This shows that customers who are employed for more than 4 years take longer to pay back as compared to those who are employed for less than 4 years or unemployed. This is indeed interesting because those who are employed for more than 4 years will be able to pay back at a shorter period of time given that they have a stable income from their employment. But in figure 1 it says otherwise.

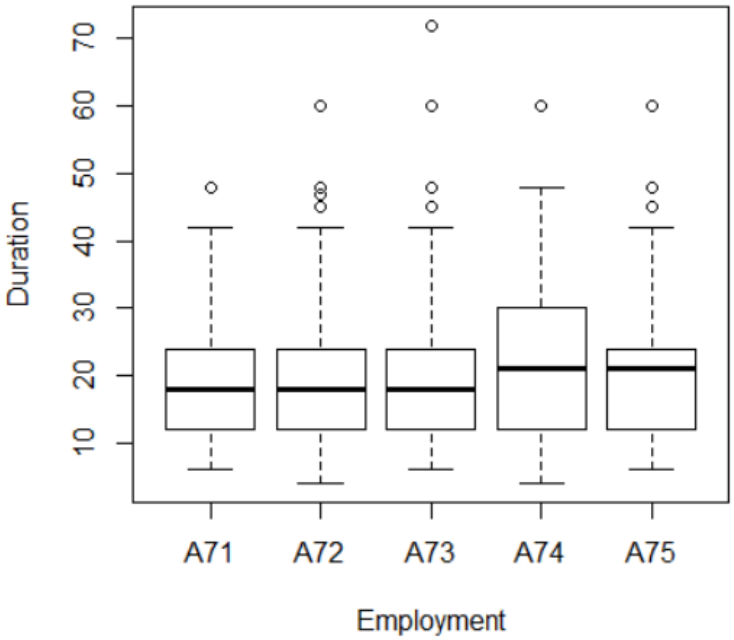

Figure 1:

Therefore, we can conclude that customers who are employed for more than 4 years tend to classified under Bad credit risk and customers who are employed for less than 4 years or are unemployed tend to be classified under Good credit risk.

We want to obtain a model that can be used to predict whether a new customer is at risk of defaulting a borrowed loan. In order to be able to do this, we divide our data into a 70% training set and 30% test set randomly. We will be running Decision trees, Naïve Bayes, Bagging, Boosting and Random Forest and deciding on the best model from these.

Firstly, I will be running Decision tree. After plotting the decision tree as shown in Figure 2, I evaluate the performance of all the classifiers using accuracy. Accuracy can be calculated

using $\frac{TP+TN}{TP+TN+FP+FN}$ for each classifier as shown below and the screenshots of confusion matrix for each classifier is provided as well.

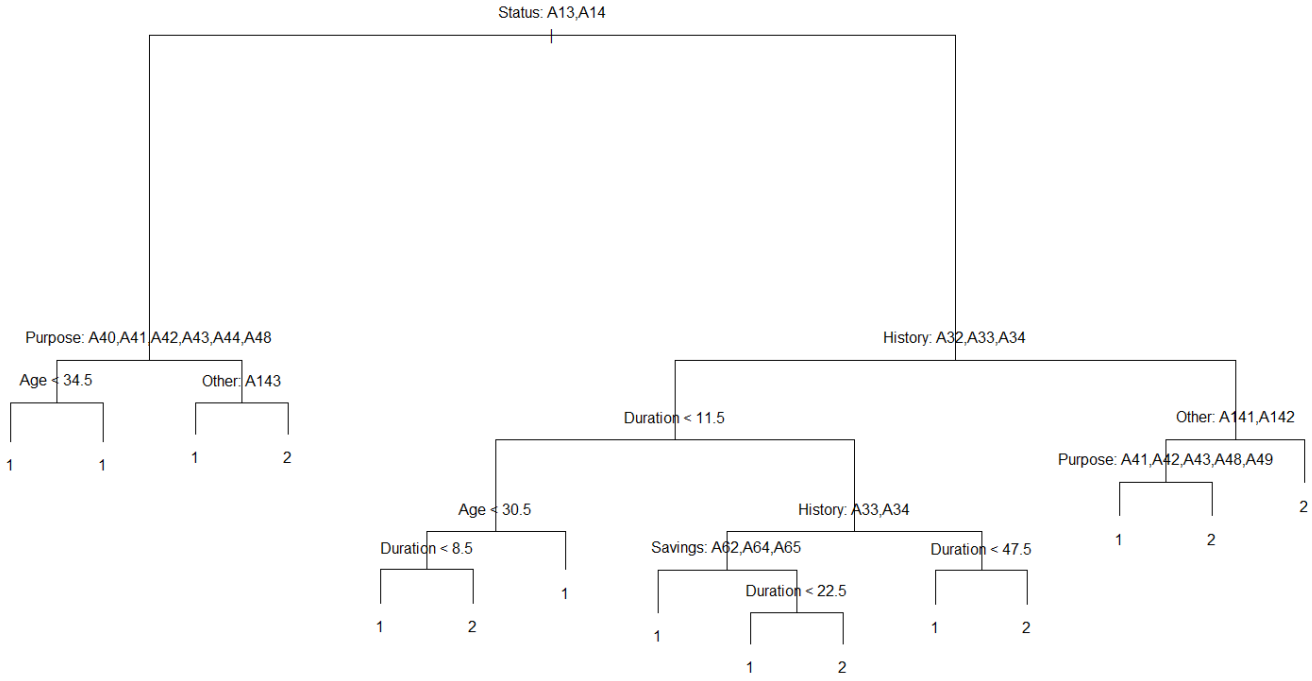

Figure 2: Decision Tree

```
#Decision Tree Confusion
> print(t1)
                Actual_Class
Predicted_Class   1   2
              1 155  48
              2  11  30
```

$$\text{Accuracy (DT)} = \frac{TP+TN}{TP+TN+FP+FN} = \frac{155+30}{155+30+11+48} = 0.7582 = 75.82\%$$

```
#Naive Bayes Confusion
> print(t2)
                Actual_Class
Predicted_Class   1   2
              1 146  32
              2  20  46
```

$$\text{Accuracy (Naive Bayes)} = \frac{TP+TN}{TP+TN+FP+FN} = \frac{146+46}{146+46+20+32} = 0.7869 = 78.69\%$$

```
Bagging Confusion
> print(GCD.predbag$confusion)
                Observed Class
Predicted Class   1   2
              1 154  49
              2  12  29
```

$$\text{Accuracy (Bagging)} = \frac{TP+TN}{TP+TN+FP+FN} = \frac{154+29}{154+29+12+49} = 0.75 = 75.00\ \%$$

```
Boosting Confusion
> print(GCDpred.boost$confusion)
                Observed Class
Predicted Class   1   2
              1 149  36
              2  17  42
```

$$\text{Accuracy (Boosting)} = \frac{TP+TN}{TP+TN+FP+FN}$$
$$= \frac{149+42}{149+42+17+36}$$
$$= 0.7828$$
$$= 78.28\ \%$$

```
Random Forest Confusion
> print(t3)
                 Actual_Class
Predicted_Class   1   2
              1 158  42
              2   8  36
```

$$\text{Accuracy (Random Forest)} = \frac{TP+TN}{TP+TN+FP+FN}$$
$$= \frac{158+36}{158+36+8+42}$$
$$= 0.7951$$
$$= 79.51\%$$

Following that, I calculated the confidence of predicting a 'Good credit risk' for each case and constructed a ROC curve for each classifier as shown in Figure 3. Each classifier is represented in different colours as shown in the legend in Figure 3.

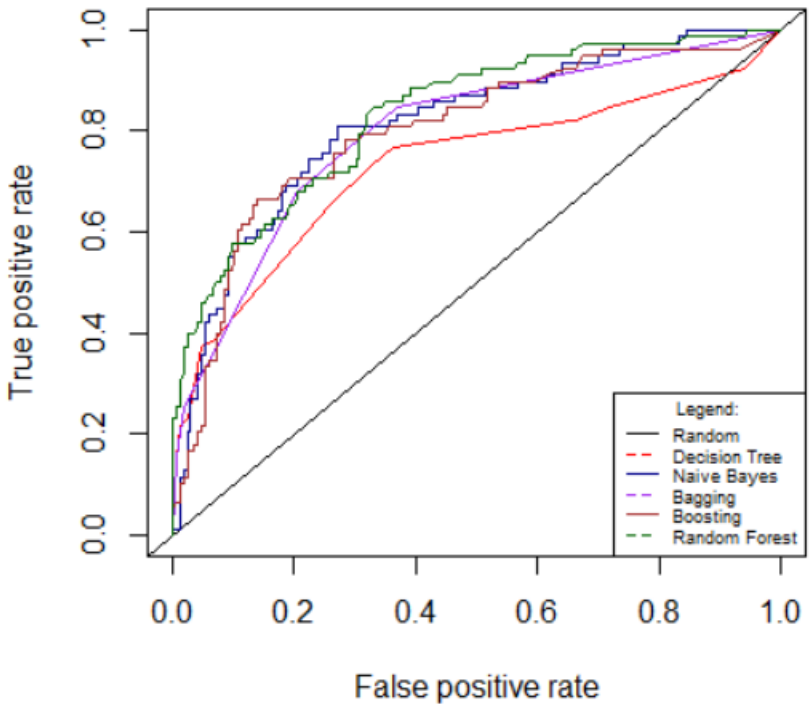

Figure 3: ROC curve for each classifier

The AUC for Decision tree is 73.13%

The AUC for Naïve Bayes is 81.35%

The AUC for Bagging is 79.76%

The AUC for Boosting is 79.92%

The AUC for Random Forest is 83.23%

The Accuracy and AUC for each classifier is represented in tabular format as shown in Figure 4.

	Decision Tree	**Naïve Bayes**	**Bagging**	**Boosting**	**Random Forest**
Accuracy	75.82%	78.69%	75%	78.28%	79.51%
AUC	73.13%	81.35%	79.76%	79.92%	83.23%

Figure 4: Accuracy and AUC for each classifier

From Figure 4, I can conclude that Random Forest is a best classifier because it has the highest accuracy (79.51%) and highest AUC (83.23%) as compared to other classifiers.

In each model, there will be certain variables that will be significant in terms of predicting whether or not an applicant is a good or bad credit risk. I will be determining the most important variables for each classifiers and necessary screenshots will be provided below.

```
 Decision Tree Attribute Importance
> print(summary(GCD.tree))

Classification tree:
tree(formula = Class ~ ., data = GCD.train)
Variables actually used in tree construction:
[1] "Status"   "Purpose"  "Age"      "Other"    "History"
[6] "Duration" "Savings"
Number of terminal nodes:  15
Residual mean deviance:  0.8603 = 475.7 / 553
Misclassification error rate: 0.2218 = 126 / 568
> |
```

For Decision tree, “Status”, “Purpose”, “Age”, “Other”, “History”, “Duration” and “Savings” are the most important attributes. These variables are used in building the decision tree as shown in Figure 2, therefore, it is logical to consider these variables when predicting whether or not an applicant is a good or bad credit risk. We can omit other attributes from the data as they have little effect on performance.

```
Bagging Attribute Importance
> print(GCD.bag$importance)
       Age     Credit    Debtors   Duration Employment
    5.3803     8.5417     1.1368    17.1751     1.3517
  Existing    Foreign    History    Housing        Job
    1.1349     0.0000     8.8736     0.0000     0.0000
    Liable      Other   Personal   Property    Purpose
    0.0000     0.9358     3.2169     4.4645    10.6225
      Rate  Residence    Savings     Status  Telephone
    2.9989     1.8648     6.9425    24.0030     1.3568
```

For Bagging, “Status”, “Duration” and “Purpose” are the most important attributes as they have a weighting greater than 10% as compared to other variables therefore they have a higher influence in predicting whether or not an applicant is a good or bad credit risk. We can omit other attributes from the data as they have little effect on performance.

```
Boosting Attribute Importance
> print(GCD.boost$importance)
       Age     Credit    Debtors   Duration Employment
    7.2886    15.5656     0.8177     9.5756     7.3281
  Existing    Foreign    History    Housing        Job
    2.5209     0.0000     8.3556     0.2219     1.7063
    Liable      Other   Personal   Property    Purpose
    0.0000     1.5150     3.5726     4.7553    13.1668
      Rate  Residence    Savings     Status  Telephone
    3.6346     2.1875     4.2240    12.5249     1.0389
```

For Boosting, “Credit”, “Purpose” and “Status” are the most important attributes as they have a weighting greater than 10% as compared to other variables therefore they have a higher influence in predicting whether or not an applicant is a good or bad credit risk. We can omit other attributes from the data as they have little effect on performance.

```
Random Forest Attribute Importance
> print(GCD.randomF$importance)
           MeanDecreaseGini
Status              26.1486
Duration            21.8235
History             18.5499
Purpose             20.1405
Credit              26.1323
Savings             11.9350
Employment          14.2180
Rate                 9.5599
Personal             8.5403
Debtors              3.9706
Residence            7.8469
Property            10.6016
Age                 21.4222
Other                6.5277
Housing              4.9906
Existing             5.0297
Job                  6.4803
Liable               3.1325
Telephone            4.4658
Foreign              0.5311
```

For Random Forest, “Status”, “Duration”, “History”, “Purpose”, “Credit”, “Savings”, “Employment”, “Property” and “Age” are the most important attributes as they have a weighting greater than 10% as compared to other variables therefore they have a higher influence in predicting whether or not an applicant is a good or bad credit risk. We can omit other attributes from the data as they have little effect on performance.

For Naïve Bayes, importance attributes don’t apply here because all the attributes are given equal importance.

I will be experimenting with parameter settings for Decision tree by doing pruning and cross validation.

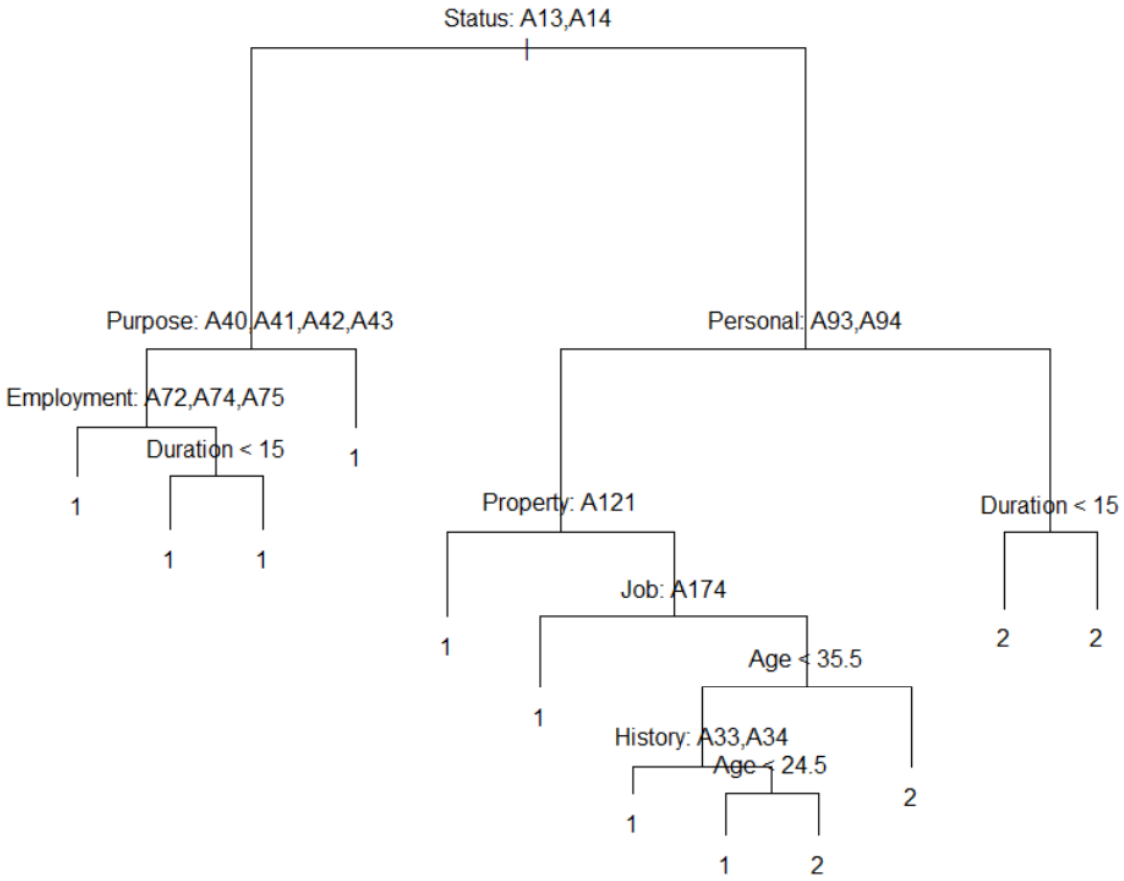

Figure 5: Decision Tree after Pruning

After Pruning the Decision tree as shown in Figure 5, I evaluate the performance by using accuracy and the screenshot of confusion matrix for Decision tree is provided as well.

```
#Pruned Decision Tree w/o CV Confusion
> print(t5)
                Actual_Class
Predicted_Class   1   2
              1 129  32
              2  37  46
```

$$\text{Accuracy (DT)} = \frac{\text{TP+TN}}{\text{TP+TN+FP+FN}}$$

$$= \frac{129+46}{129+46+37+32}$$

$$= 0.7172$$

$$= 71.72\%$$

The accuracy is worse than expected because previously the Decision tree before pruning is 75.82% and the Decision tree after pruning is 71.72%. Therefore, I will perform cross validation.

```
$size
[1] 12  8  5  3  1

$dev
[1] 26 25 23 23 38
```

The best is 5 leaves with standard deviation of 23. Therefore, I plotted the Decision tree after doing cross validation as shown in Figure 6.

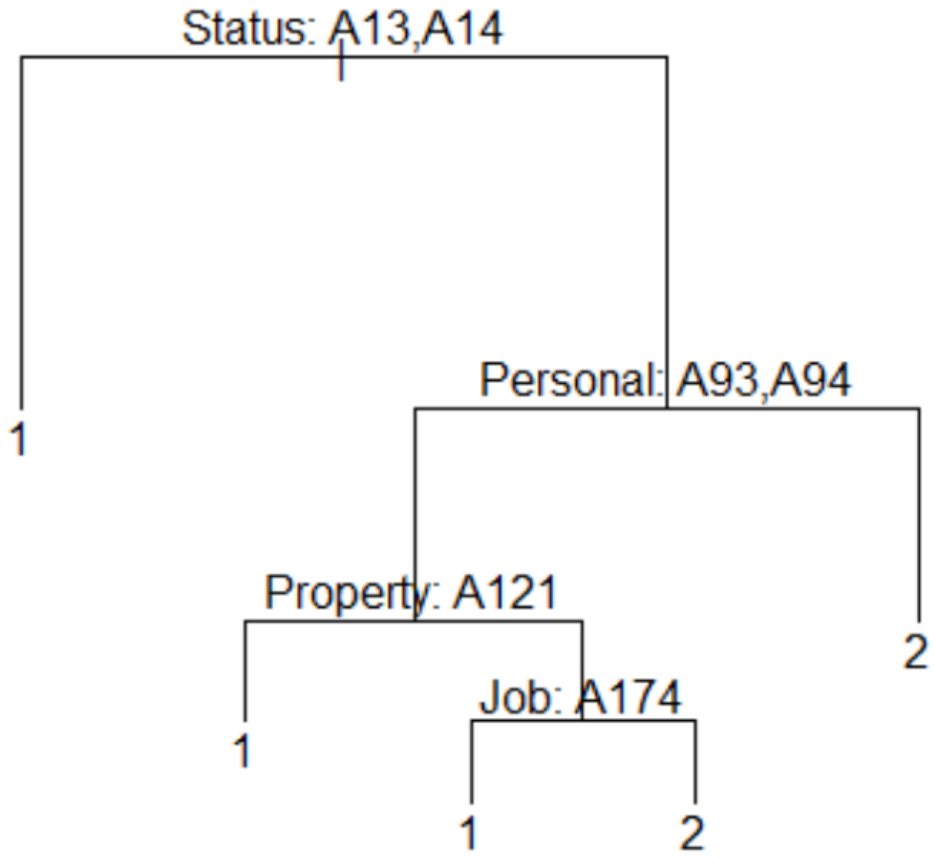

Figure 6: Decision tree after cross validation

Following which, I evaluate the performance by using accuracy and the screenshot of confusion matrix for Decision tree is provided as well.

```
#Pruned Decision Tree with CV Confusion
> print(t6)
                Actual_Class
Predicted_Class   1   2
              1 119  24
              2  47  54
```

$$\text{Accuracy (DT)} = \frac{\text{TP+TN}}{\text{TP+TN+FP+FN}}$$

$$= \frac{119+54}{119+54+47+24}$$

$$= 0.7090$$

$$= 70.90\%$$

Once again, the accuracy is worst than expected. Previously the accuracy of Decision tree before pruning and cross validation is 75.82% and the accuracy of Decision tree after pruning and before cross validation is 71.72% and the accuracy of Decision tree after pruning and cross validation is 70.90%. After which I plotted the ROC curve of the Decision tree after pruning and cross validation as shown in Figure 7.

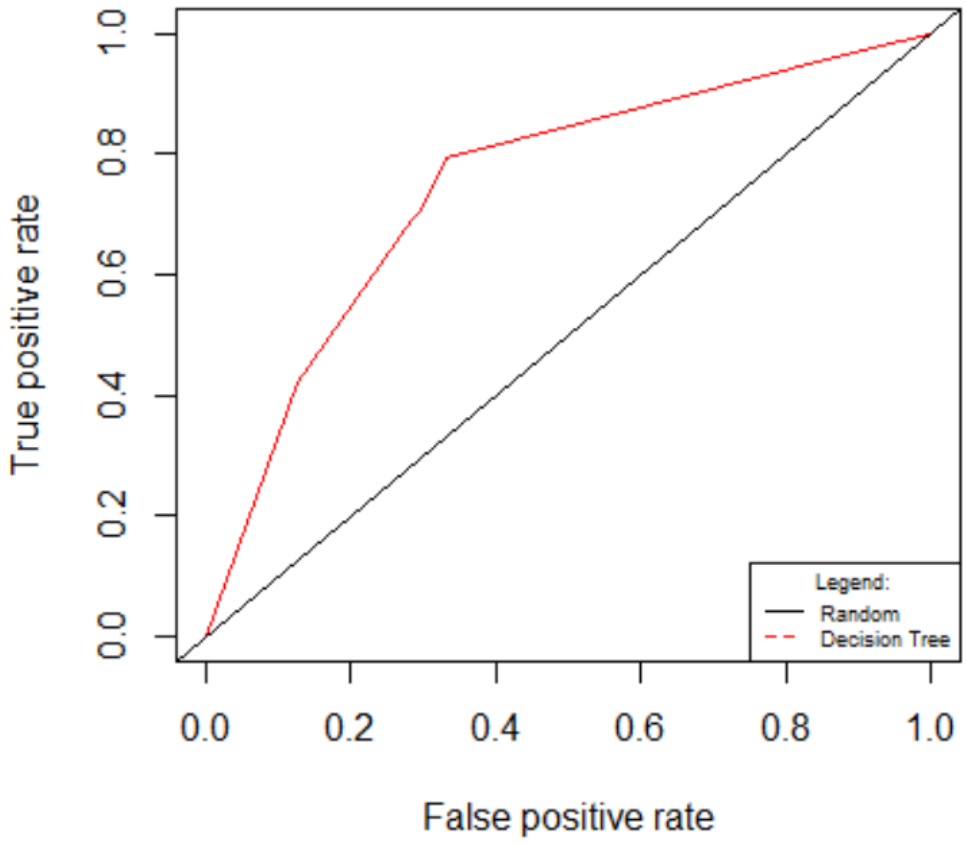

Figure 7: ROC curve for DT after pruning and CV

The AUC for Decision tree after pruning and cross validation is 75.10% which is higher than the AUC for Decision tree before pruning and cross validation which is 73.13%.

In conclusion, I can say Decision tree after pruning and cross validation is a better classifier as compared to Decision tree before pruning and cross validation. This is because, the AUC (75.10%) is higher even though the Accuracy (70.90%) is lower.

Before fitting the ANN, I need to prepare my data. For that I had to do some data pre-processing such as removing rows that contains missing (NA) values, dividing my data into 70% training set and 30% test set and then only I fit my neural network and tested accuracy. I didn't have to recode the output (i.e. 'Class') as numeric because it is already numeric.

In implementing the Artificial Neuron Network (ANN) with 3 hidden layers as shown below in Figure 8, the attributes I used are "Duration", "Age" and "Credit". Because they are the most important variables when predicting whether or not an applicant is a good or bad credit risk for at least one of the classifiers as mentioned in Question 7. Moreover, these attributes are numerical.

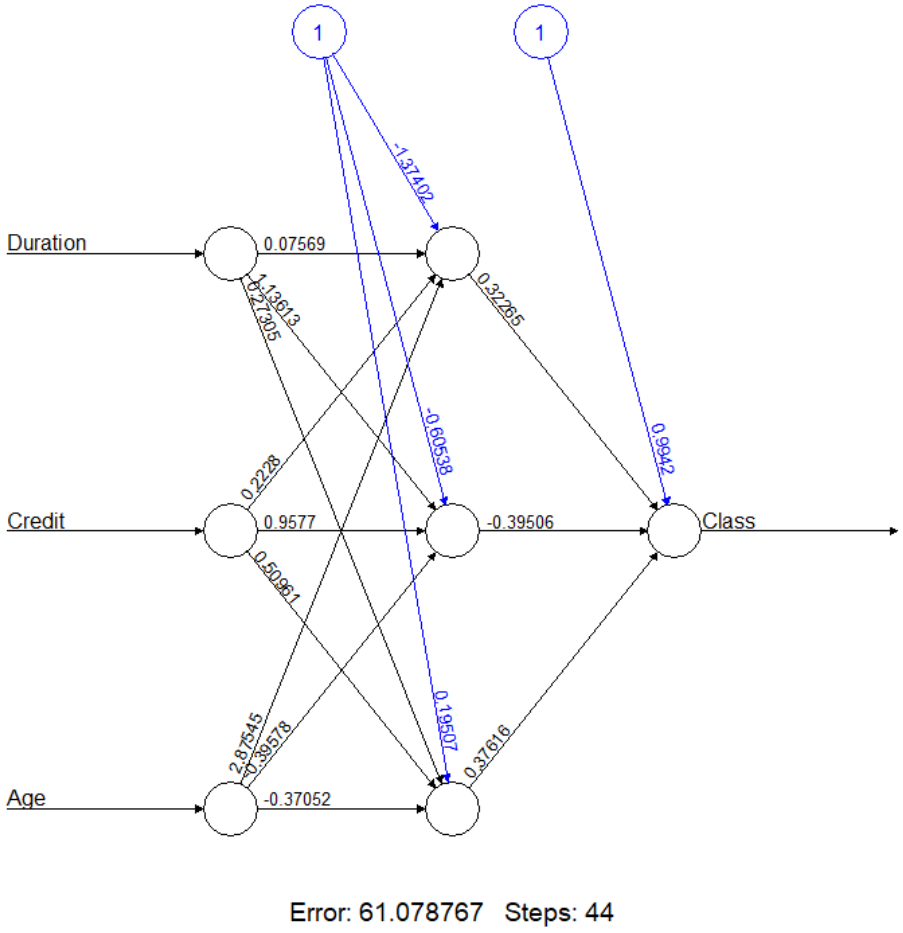

Figure 8

Following which I wanted to evaluate the performance of ANN and the screenshot is provided below.

```
#ANN Confusion
> table(observed=GCDDtest$Class,predicted=GCDD.pred$V1)
        predicted
observed   1   2
       1 157   1
       2  70   0
```

The accuracy of ANN is $\frac{157+0}{157+0+70+1} = 0.6886 = 68.86\%$

In terms of accuracy, it is quite bad (68.86%) as compared to other classifiers. This could be due to the lower number of hidden layers which is why accuracy might be low. Increasing the hidden layers will possibly increase the accuracy as well. In this case, we are using 3 hidden layers, increasing it to 4 hidden layers might help in increasing the accuracy. But we can't guarantee that accuracy will increase because the accuracy might be constant or even fall when we add an extra layer.