Group Assignment – Data Mining

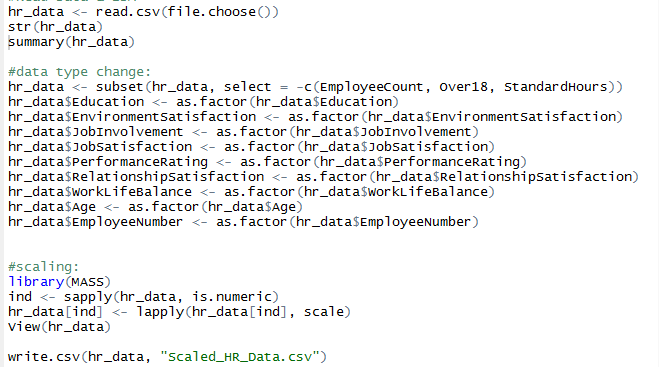
Comparison of CART & Neural Net Models for HR Attrition Data

Avinash Balamurali , Divyalakshme Arangasubramanian, Subashree Bhaskaran, Senthilraj, Obli Narasimharajan

**Changing the Data Type**

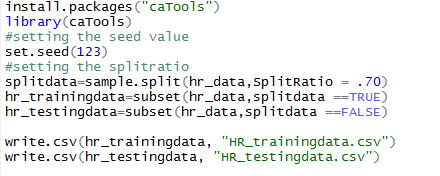
The Employee Attrition Data given is loaded into R for analysis. But the Data has Ordinal Ratings That have been picked by R as Integers.

So Changing the DATA Type of these Variables and scaling them as follows.



**Splitting the Data**

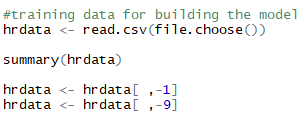
Data after splitting into Training and test Data Sets are stored as HR\_trainingdata.csv , HR\_testingdata.csv files separately to be used in the CART and Neural net Models Later.



**Building the CART Model**

CART is "Classification and Regression Tree”. The basic principle for working of decision trees is to split each parent node in as distinct Binary nodes as possible. CART has only one type criteria splitting, which is impurity-based splitting or Gini Gain.

Before building the model, it is imperative to remove the serial column and employee number attribute from the training data and it is done in R as follows:



**Model Control Parameters**

We then must choose a **Minsplit, Minbucket ,CP and Xval** values for the CART model control.

**Minsplit value** ranges from 1-4% of the total number of observations. Since the total number of observations in our training data is 2022, we take the

Minsplit = 40 [approximately 2% of the total records].

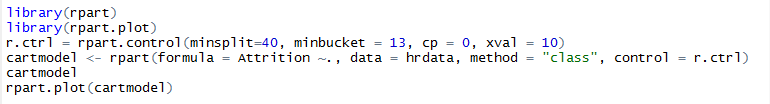
**Minbucket value** is usually one third of the Minsplit value.

Minbucket = 13.

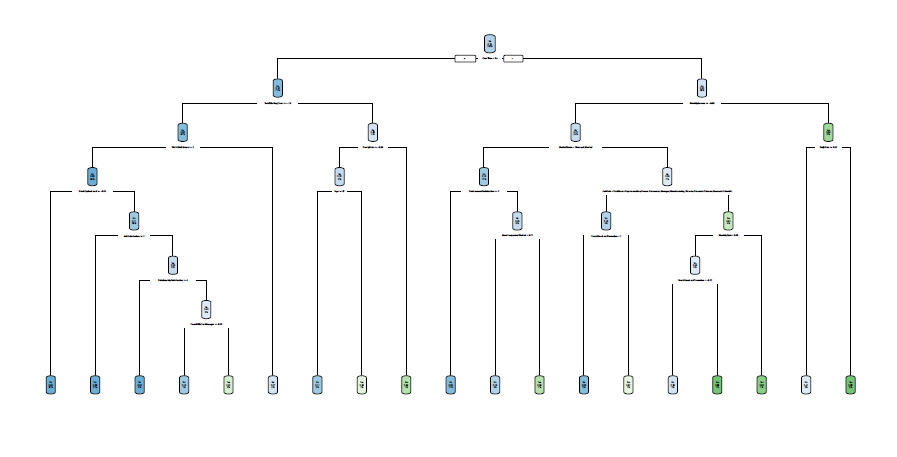
**Xval value** is needed cross validation to take care of the greediness in the algorithm and generally 2, 3, 5 or 10 are chosen as Xval values. Therefore, **Xval value** =10

The Cost Complexity parameter is yet to be identified so., plotting the entire tree to be pruned later.

We build the model in R, using our training set and the control values we chose, as shown below:

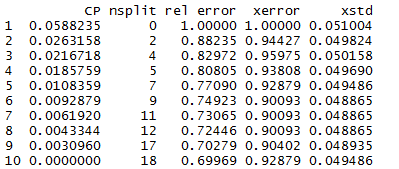


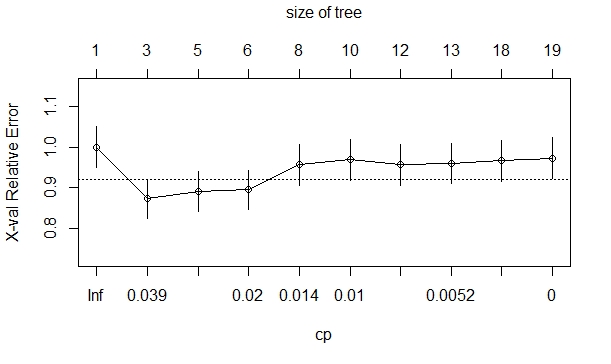
and we get the following tree output:



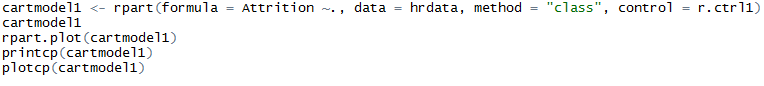
**Identifying the Right CP**

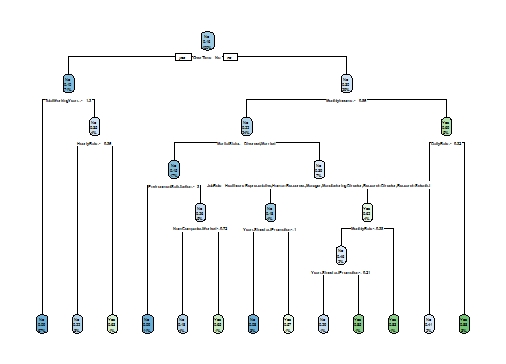




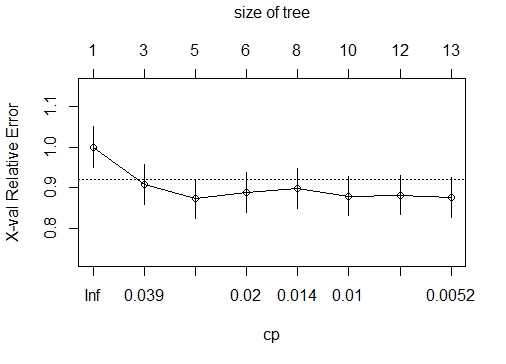


The Point of Inflection seems to be at **CP=0.00433.** Running the Model Again For CP=0.00433

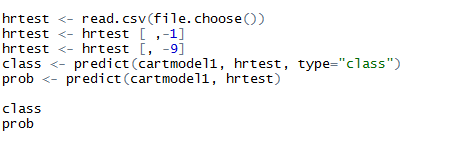


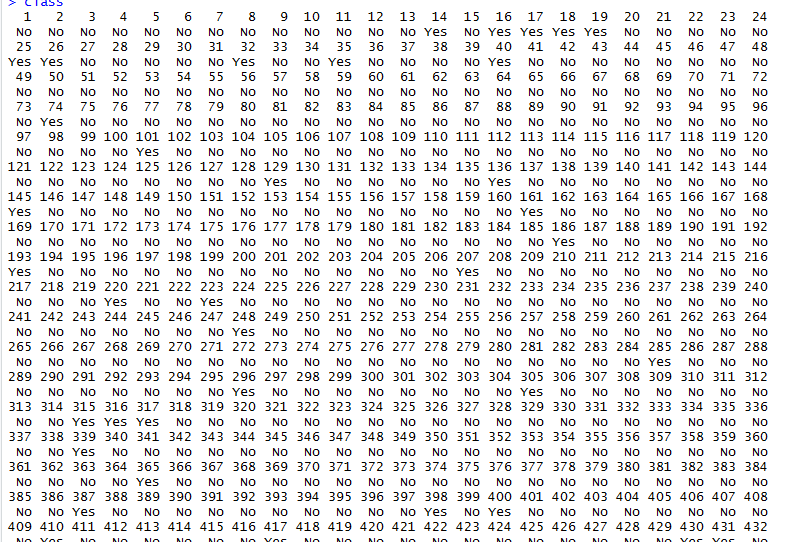


The No of Splits has been reduced due to the Pruning of the branches

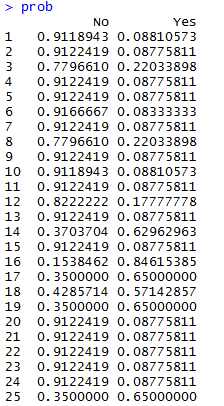


Validating the Model Using the Test Data.





The Probabilities are



**Confusion Matrix**

The confusion matrix and the other performance indices are as follows:

Confusion Matrix and Statistics

Reference

Prediction No Yes

No 748 19

Yes 95 56

Accuracy : 0.8758

95% CI : (0.8527, 0.8965)

No Information Rate : 0.9183

P-Value [Acc > NIR] : 1

Kappa : 0.4338

Mcnemar's Test P-Value : 2.15e-12

Sensitivity : 0.8873

Specificity : 0.7467

Pos Pred Value : 0.9752

Neg Pred Value : 0.3709

Precision : 0.9752

Recall : 0.8873

F1 : 0.9292

Prevalence : 0.9183

Detection Rate : 0.8148

Detection Prevalence : 0.8355

Balanced Accuracy : 0.8170

'Positive' Class : No

From the above performance values, we can see that the model performance for the current business problem is considerably good. But to choose the best model for the problem at hand, we need to compare these values with other model performances.

**Building Neural Network Model**

Before running the Neural network, model there are few variables in training data set which needs to be converted form categorical to numerical

Number of rows, response rate for both training and test data are checked

The below variables are removed from both training and test data set since the correlation is less for these variables.

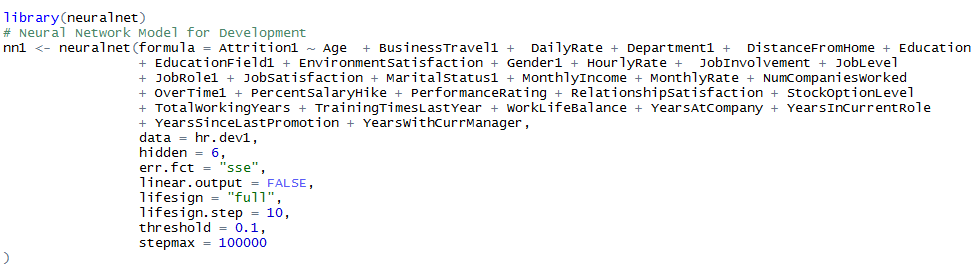
EmployeeCount

EmployeeNumber

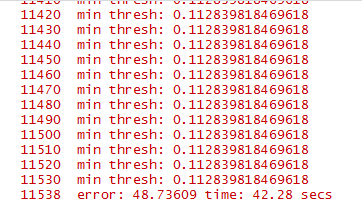
Over18

StandardHours

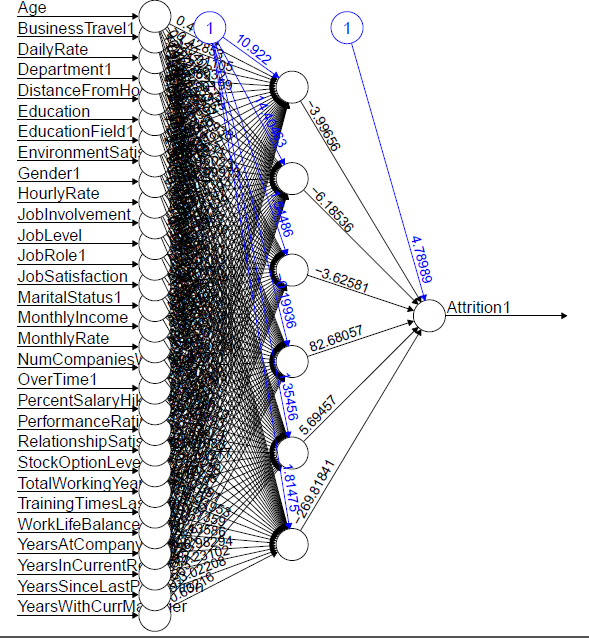
Neural Network with target variable as attrition



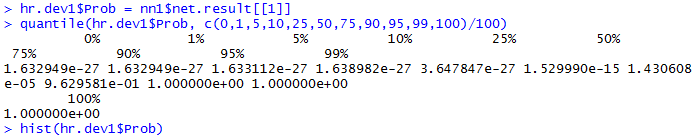
The NN Model Converges at 11538th Iteration as mentioned below



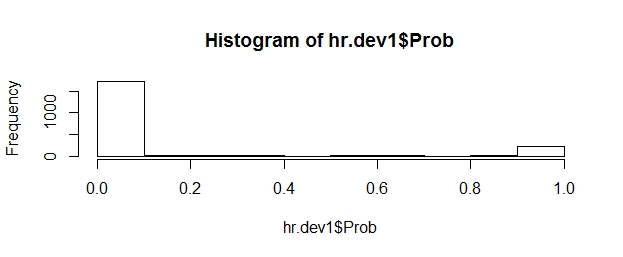
The NN Plot for the dev data Set is



**Calculating Probabilities**

****

The Histogram of the Probabilities are



Declining and the Rank Order Output is as Follows

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| S.no | deciles | cnt | cnt\_resp | cnt\_non\_resp | rrate | cum\_resp | cum\_non\_resp | cum\_rel\_resp | cum\_rel\_non\_resp | ks |
| 1 | 10 | 203 | 193 | 10 | 95.00% | 193 | 10 | 60.00% | 1% | 0.59 |
| 2 | 9 | 202 | 57 | 145 | 28.00% | 250 | 155 | 77% | 9.00% | 0.68 |
| 3 | 8 | 202 | 5 | 197 | 2.00% | 255 | 352 | 79% | 21.00% | 0.58 |
| 4 | 7 | 202 | 14 | 188 | 7.00% | 269 | 540 | 83% | 32.00% | 0.51 |
| 5 | 6 | 202 | 11 | 191 | 5.00% | 280 | 731 | 87% | 43.00% | 0.44 |
| 6 | 5 | 203 | 7 | 196 | 3.00% | 287 | 927 | 89% | 55.00% | 0.34 |
| 7 | 4 | 202 | 11 | 191 | 5.00% | 298 | 1118 | 92% | 66.00% | 0.26 |
| 8 | 3 | 201 | 10 | 191 | 5.00% | 308 | 1309 | 95% | 77.00% | 0.18 |
| 9 | 2 | 203 | 7 | 196 | 3.00% | 315 | 1505 | 98% | 89.00% | 0.09 |
| 10 | 1 | 202 | 8 | 194 | 4.00% | 323 | 1699 | 100.00% | 100.00% | 0 |

**Confusion Matrix**

hr.dev1$Class = ifelse(hr.dev1$Prob>0.21,1,0)

with( hr.dev1, table(Attrition1, as.factor(Class) ))

Attrition1 0 1

0 1574 125

1 58 265

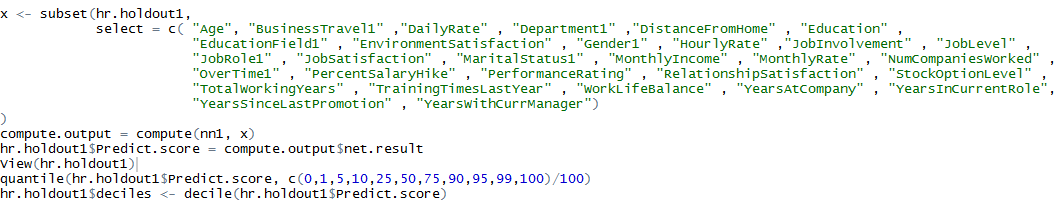
Accuracy = 91%

Precision= 82%

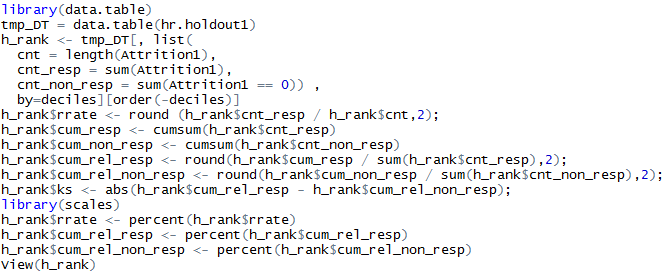
Recall=67.95%

F1 score=74%

**Testing the Model on Hold-out Data**

****

Computing the Rank Order Matrix



|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| S.no | deciles | cnt | cnt\_resp | cnt\_non\_resp | rrate | cum\_resp | cum\_non\_resp | cum\_rel\_resp | cum\_rel\_non\_resp | ks |
| 1 | 10 | 92 | 87 | 5 | 95.00% | 87 | 5 | 58.00% | 1.00% | 0.57 |
| 2 | 9 | 92 | 30 | 62 | 33.00% | 117 | 67 | 77.00% | 9.00% | 0.68 |
| 3 | 8 | 92 | 1 | 91 | 1.00% | 118 | 158 | 78.00% | 21.00% | 0.57 |
| 4 | 7 | 91 | 10 | 81 | 11.00% | 128 | 239 | 85.00% | 31.00% | 0.54 |
| 5 | 6 | 92 | 5 | 87 | 5.00% | 133 | 326 | 88.00% | 43.00% | 0.45 |
| 6 | 5 | 92 | 6 | 86 | 7.00% | 139 | 412 | 92.00% | 54.00% | 0.38 |
| 7 | 4 | 91 | 4 | 87 | 4.00% | 143 | 499 | 95.00% | 65.00% | 0.3 |
| 8 | 3 | 92 | 2 | 90 | 2.00% | 145 | 589 | 96.00% | 77.00% | 0.19 |
| 9 | 2 | 92 | 1 | 91 | 1.00% | 146 | 680 | 97.00% | 89.00% | 0.08 |
| 10 | 1 | 92 | 5 | 87 | 5.00% | 151 | 767 | 100.00% | 100.00% | 0 |

The Training & Test data Rank order Matrix appear to be consistent , Now Computing the Confusion Matrix for the Performance Measures

**The Confusion Matrix** of the Test Data is

Attrition1 0 1

0 708 29

1 59 122

The Performance Measures of the NN Model is

Accuracy = 90.4 %

Precision= 67%

Recall=81%

F1 score=73%

**Inference & Conclusion :**

Comparing the Accuracy of the CART & Neural Network Models., Neural Network Appears to be the most Accurate.

But for the current problem statement of attrition rate, analysing sensitivity is important as having a lower type II error ( false negative) is critical. That is, attrition should not be predicted as 'No' when the employee attrition occurs.

Hence, Recall & F1 Score seems to be the appropriate performance measures.

By this, the CART Model Outperforms the Neural Network Model and will be the reliable Model for application.

**----------------------------------------------------------------------------------------------------------------------------------**