Individual Case I

Last Name: **Ramakrishnan**   
First Name: **Siva**  
UCID: **M10669365**

# 1. Boston Housing data

Background:

The dataset reports the median value(Medv) of owner-occupied homes in the suburbs of Boston area. The dataset also contains several predictor variables which can be used to predict Medv.

Goal:

The aim of this case study and contrast different models in predicting the median value. Since the response variable is continuous, the techniques used will be Generalized Linear Model, Generalized Additive Model, Regression Tree and Neural Networks.

Approach:

The Analysis of the different techniques was performed in R. The following steps were followed.

1. Load the data into a Dataset and perform EDA to gain insights into the data.
2. Divide the dataset into training and test sets in the ratio 75:25
3. Use the Training dataset to fit different the 4 models mentioned above
4. Record the Average sum squared error for both the in-sample and out-sample data
5. Analyze the recorded values for the 4 models and determine the best model.

Results:

Given below is the consolidated table with the average SSE for the 4 models

|  |  |  |
| --- | --- | --- |
|  | Averaged Sum Squared Error | |
|  | In-Sample | Out-Sample |
| Linear Regression | 21.89 | 22.82 |
| Regression Tree | 14.45 | 28.12 |
| GAM | 7.36 | 15.82 |
| Neural Network | 3.35 | 30.32 |

### Major findings:

1. The Best model among the four is the Generalized Additive model. It gives us the least Averaged SSE for test data. This shows that Generalized Additive models are usually better than Generalized Linear Models in terms of flexibility and the ability to model non-linear relationships.

2. Although Neural Network gives the least in-sample Avg. SSE, the out-Sample value is the highest. This shows that Neural Networks tend to generalize poorly and tend to overfit the data.

# 2. German Credit Data

Background:

The Loan Application process involves classifying people into good credit risk or bad credit risk. The German credit scoring dataset reports 1000 such observations of good/bad credit risks along with several predictor variables which can be used in the prediction.

Goal:

The aim of this case study and contrast different models in classifying good/bad credit risks. Since the response variable is binary, the techniques used will be Logistic Regression, Classification trees, Generalized Additive Models and Linear Discriminant Analysis.

Approach:

The Analysis of the different techniques was performed in R. The following steps were followed.

1. Load the data into a Dataset and perform EDA to gain insights into the data.
2. Divide the dataset into training and test sets in the ratio 75:25
3. Use the Training dataset to fit different the 4 models mentioned above
4. Record the misclassification rates and AUC of each of the model
5. Analyze the recorded values for the 4 models and determine the best model.

Results:

Given below is the consolidated table with the average SSE for the 4 models

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Asymmetric Misclassification Rate | | AUC | |
|  | In-Sample | Out-Sample | In-Sample | Out-Sample |
| Logistic Regression | 0.42 | 0.62 | 0.846 | 0.762 |
| Classification Tree | 0.52 | 0.62 | 0.836 | 0.682 |
| GAM | 0.42 | 0.62 | 0.854 | 0.767 |
| LDA | 0.42 | 0.65 | 0.849 | 0.763 |

### Major findings:

1. The performance of all the four statistical models are very similar. There is no one clear model which results in much better prediction.

2. When considering only the Out-Sample AUC the Classification tree is performing poorly when compared to others.

## Boston Housing

#### Dataset Summary

Given below the summary of the data in the Boston housing.

* Number of Observations: 506
* Number of Attributes: 13

#### Attribute Information:

* CRIM: per capita crime rate by town
* ZN: proportion of residential land zoned for lots over 25,000 sq.ft.
* INDUS: proportion of non-retail business acres per town
* CHAS: Charles River dummy variable (= 1 if tract bounds river; 0 otherwise)
* NOX: nitric oxides concentration (parts per 10 million)
* RM: average number of rooms per dwelling
* AGE: proportion of owner-occupied units built prior to 1940
* DIS: weighted distances to five Boston employment centers
* RAD: index of accessibility to radial highways
* TAX: full-value property-tax rate per $10,000
* PTRATIO: pupil-teacher ratio by town
* B: 1000(Bk - 0.63) ^2 where Bk is the proportion of blacks by town
* LSTAT: % lower status of the population

MEDV (Median value of owner-occupied homes in $1000's) is the response variable.

#### First Steps:

* Loaded the MASS library which contains the Boston dataset.
* Spit the data into training and testing datasets in the ratio 75:25 using M-number as seed.

#### Linear Regression:

Performed Variable selection using Bidirectional Stepwise procedure with AIC criterion. The resulting model obtained was

medv ~ lstat + rm + ptratio + chas + black + dis + nox + zn + crim + rad + tax

Model statistics are mentioned below:

|  |  |
| --- | --- |
| AIC | 2271.18 |
| BIC | 2322.36 |
| Model MSE | 22.60 |

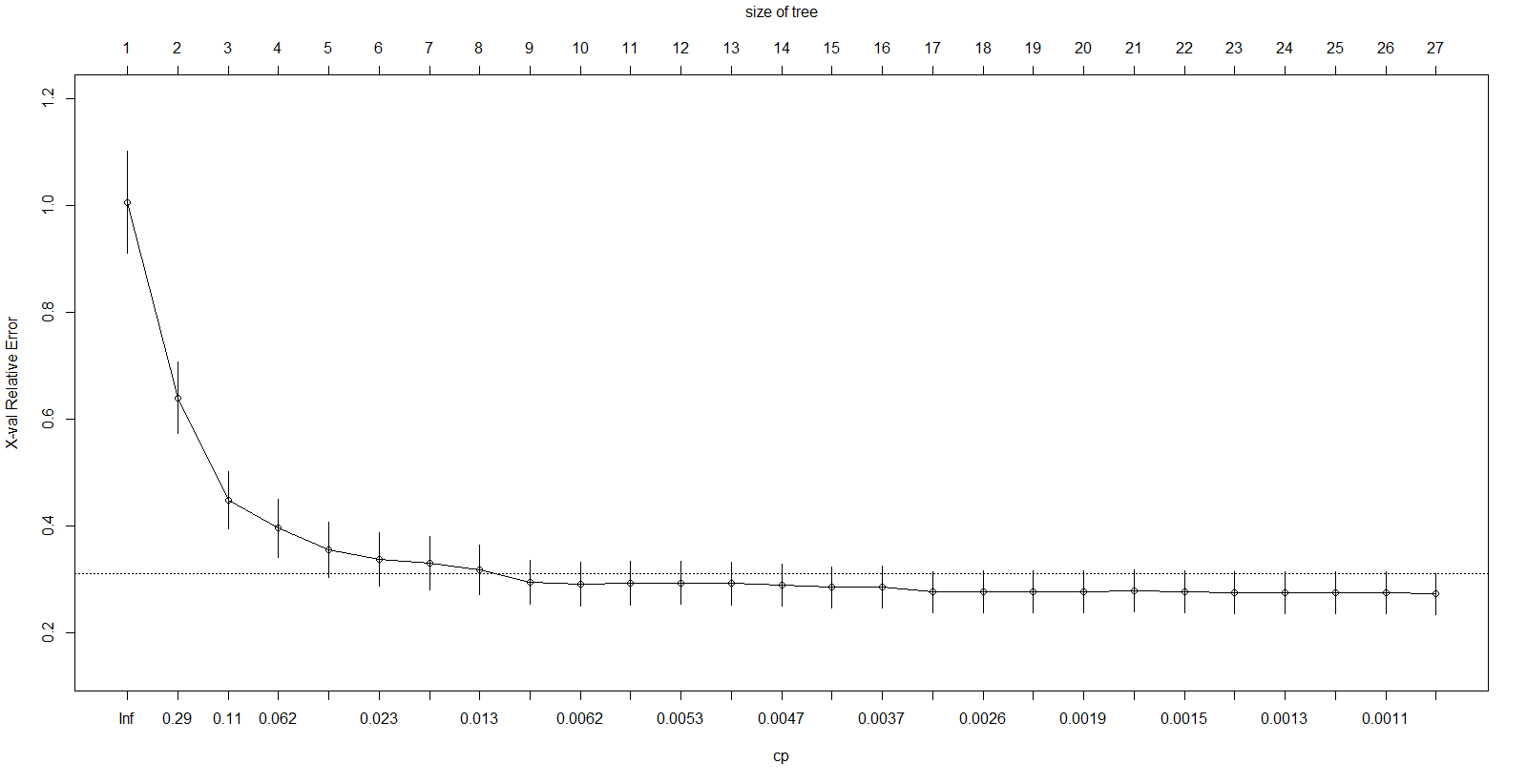
The 3-fold cross validation of the linear regression model using the entire boston dataset resulted in MSE value of 23.68

|  |  |
| --- | --- |
|  | Average Sample Squared error |
| In-Sample | 21.89 |
| out-Sample | 22.82 |

#### Regression Tree

Used the rpart package to build the regression tree.

Regression Trees tend to be overly complex and overfit the data. In order to reduce the complexity of the tree pruning the tree using plotcp() function is performed. The function gives the relationship between 10-fold cross-validation error in the training set and size of tree. The optimum value of Cp obtained was 0.011 and size of the tree as 9.



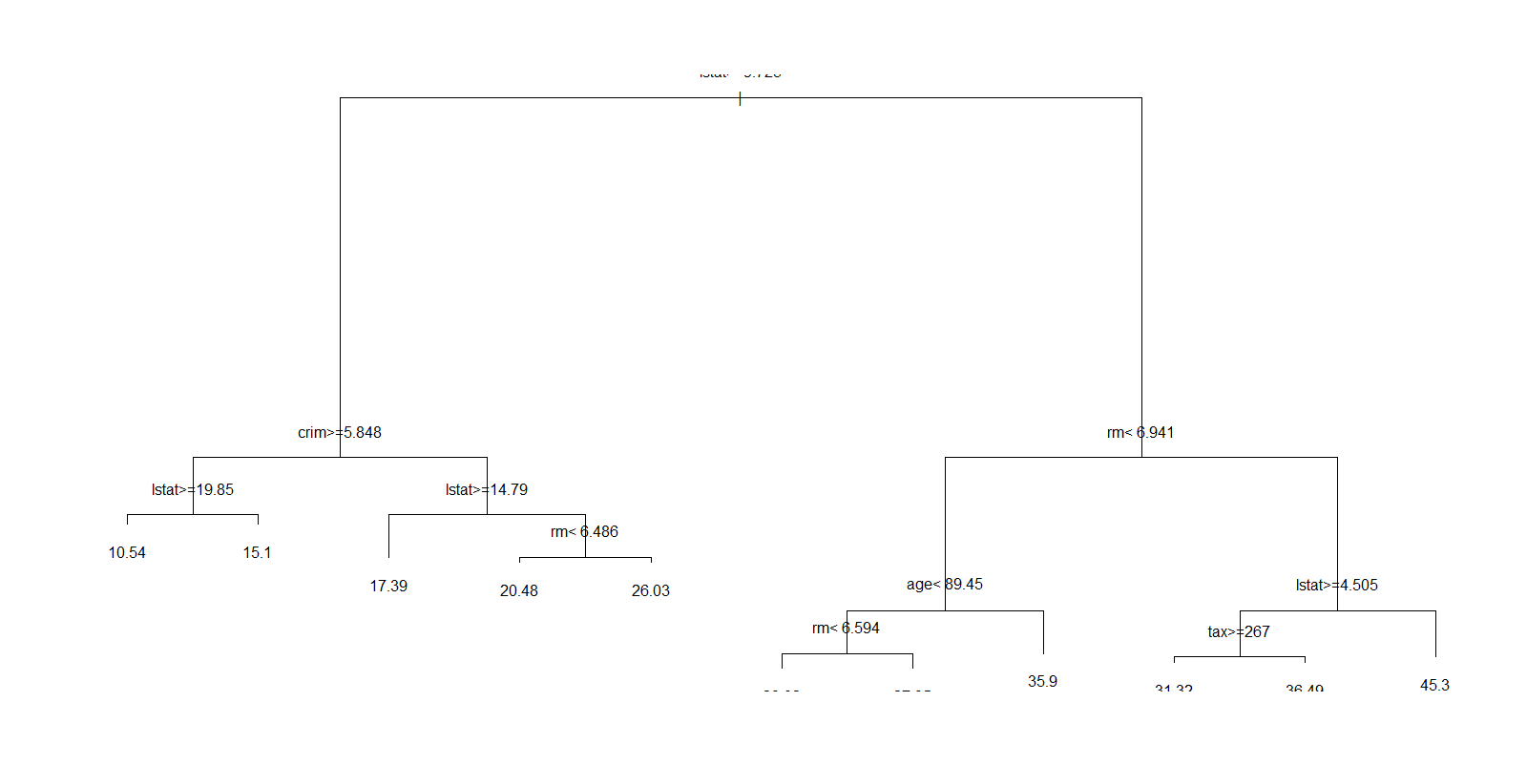


Figure 1:Pruned tree

Average Sample squared errors of training and test sets are given below

|  |  |
| --- | --- |
|  | Average Sample Squared error |
| In-Sample | 14.45 |
| out-Sample | 28.12 |

#### Generalized Additive Model

GAM is a generalized linear model in which the we predict using sum of smooth functions of the predictors and conventional linear predictors.

Used the implementation of GAM in the mgcv library to build a GAM Model. The smooth terms are specified in a gam formula using the s() function.

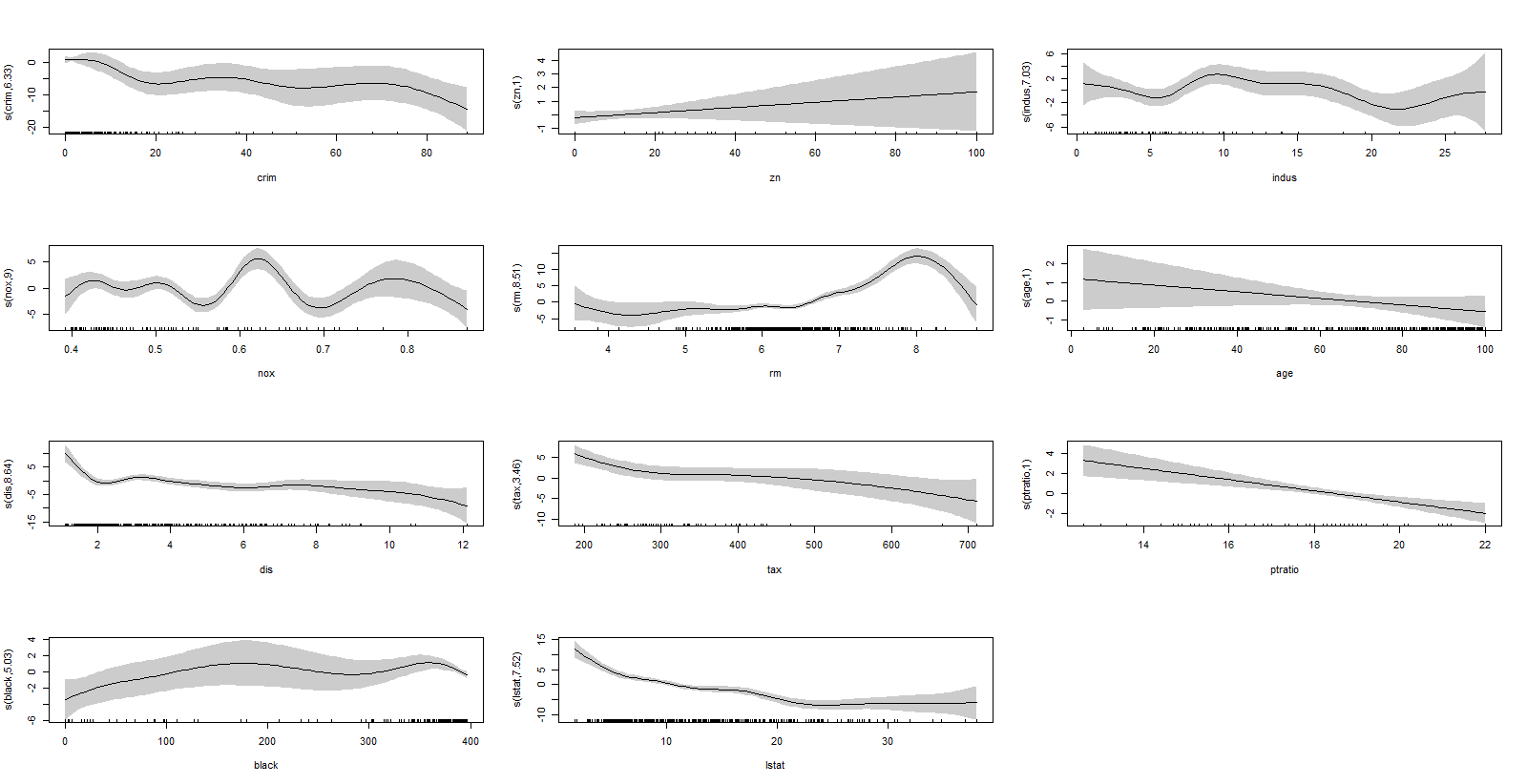
Initially ran the GAM model with all continuous predictors with smooth function.

The summary of Spline terms is given below

|  |
| --- |
| Approximate significance of smooth terms: |
| edf Ref.df F p-value |
| s(crim) 6.330 7.381 8.277 1.92e-09 \*\*\* |
| s(zn) 1.000 1.000 1.387 0.239793 |
| s(indus) 7.031 8.003 3.773 0.000299 \*\*\* |
| s(nox) 9.000 9.000 13.504 < 2e-16 \*\*\* |
| s(rm) 8.510 8.917 20.821 < 2e-16 \*\*\* |
| s(age) 1.000 1.000 2.088 0.149450 |
| s(dis) 8.637 8.958 7.145 2.10e-09 \*\*\* |
| s(tax) 3.457 4.163 7.697 6.32e-06 \*\*\* |
| s(ptratio) 1.000 1.000 17.887 3.05e-05 \*\*\* |
| s(black) 5.026 6.061 2.911 0.008500 \*\* |
| s(lstat) 7.517 8.439 16.943 < 2e-16 \*\*\* |

From the above table, we can observe that the spline terms for predictors zn, age and ptratio the effective degrees of freedom is 1 suggesting that these terms wouldn’t require spline smoothing.

It can also be observed that the p-values of zn and age are not significant suggesting that these terms can be removed entirely from the model. But since these p-values are not overly high, I decided to keep them in the final model.



The above figure plots the predictors vs their spline terms. If the plot is a straight line, it suggests that the spline terms can be dropped from the model

The Final model selected based on the above analysis is:

medv ~ s(crim)+zn+s(indus)+chas+s(nox)+s(rm)+age+s(dis)+rad+s(tax)+ptratio+s(black)+s(lstat)

Model performance statistics are mentioned below:

|  |  |
| --- | --- |
| AIC | 1957.2 |
| BIC | 2203.32 |
| Adj- Rsquared | .89 |
| Deviance explained | 91.1% |

We are getting a good fit using the GAM model has the Adj- Rsquared and Deviance explained by the model is high.

Average Sample squared errors of training and test sets are given below

|  |  |
| --- | --- |
|  | Average Sample Squared error |
| In-Sample | 7.36 |
| out-Sample | 15.82 |

#### Neural Network

Neural Networks are black box models for prediction

Used the NNet package for implementing the Neural network in R. The Nnet Package implements the single-Hidden-layer Neural network.

There are two basic parameters that are to be fine-tuned for the Neural network. They are

* Size: The size parameters indicates the number of neuron in a hidden layer.
* decay – This is a penalty factor which cause decay to the weights.

Approach:

To find the optimal combination of Size and decay, we follow the below steps:

1. Divide the training data again into 2 sets (training and testing) in the ratio 75:25
2. Build the neural network for different size values
3. Plot the Neural network size vs average squared error for both the datasets which were created above
4. Select a subset of the neuron sizes where the errors are low
5. Build neural networks for each of the size determined above for various decay values between 0 and 0.0001
6. Plot the size vs average squared error for each of the decay values
7. Select the combination of decay and size for which the error is least.

Plot of the average squared error vs the layer size for both the test and training data is shown below

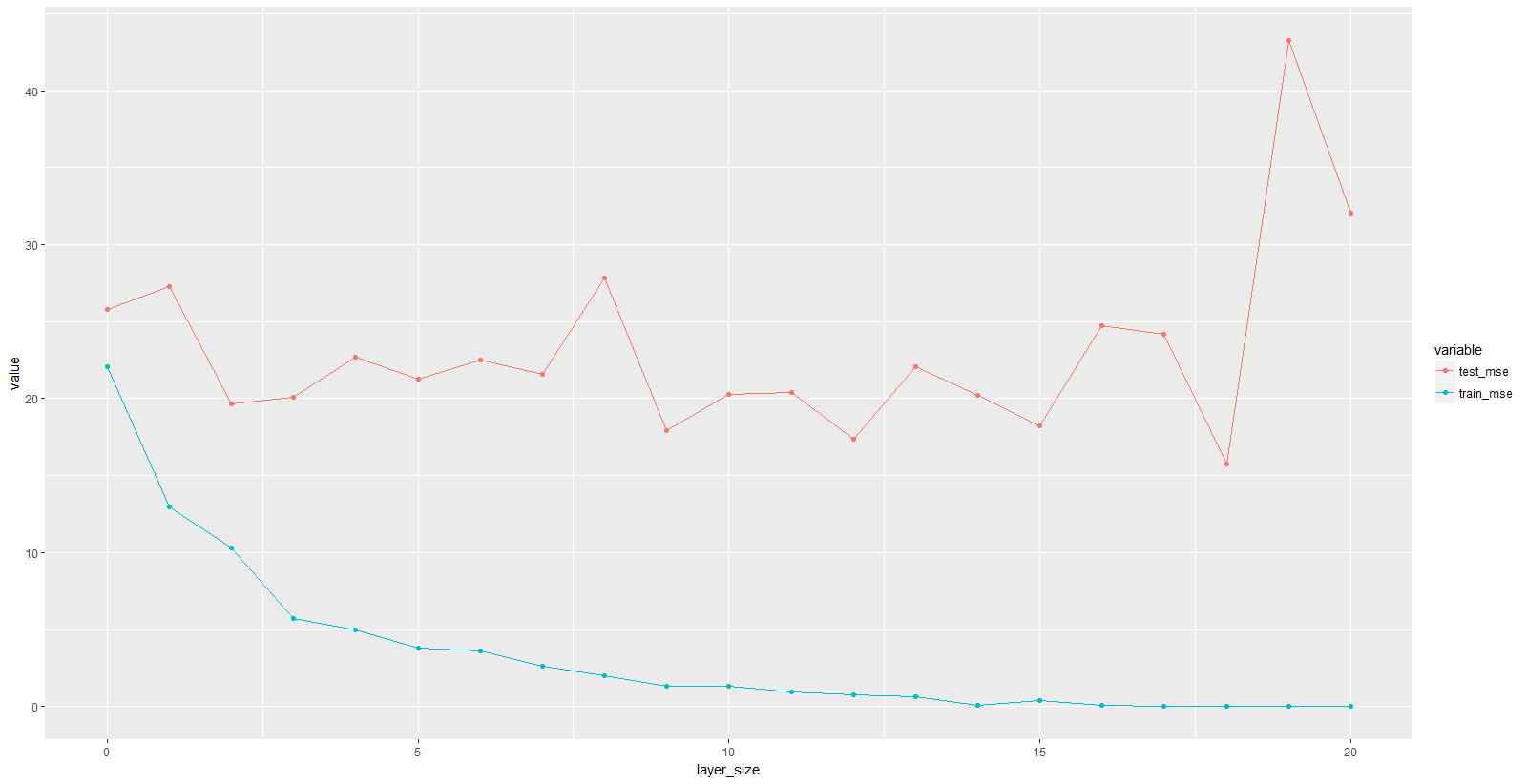
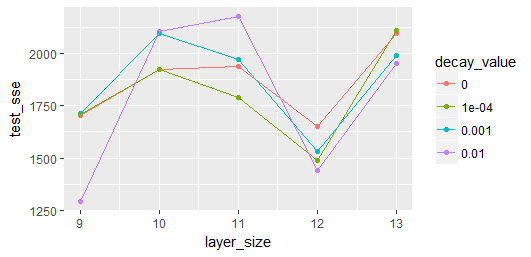


Figure 2: MSE vs layer size

From the above plot it looks like errors are low for size in the range of 9 to 13 for the test dataset. Next step would be to try different combinations of these layer sizes and decay.

Plot of the average squared error in the test set vs the layer size various decay values is shown below



The combination of size=9 and decay value of 0.01 is giving us the least error. So we use these values to build the final neural network

Average Sample squared errors of training and test sets using the final model is given below are given below

|  |  |
| --- | --- |
|  | Average Sample Squared error |
| In-Sample | 3.35 |
| out-Sample | 30.32 |

## German Credit

#### Dataset Summary

The dataset contain data about people being classified as good or bad credit risks according to the set of attributes.

Given below the summary of the data in the German Credit

* Number of Observations: 1000
* Number of predictor Attributes in the dataset: 20 (7 numerical, 13 categorical)

The response variable is binary. It is coded as 0 when the person is classified as good credit risk and 1 for bad credit risk.

#### First Steps:

* Created a data frame of the German Credit dataset from the UCI website.
* Split the data into training and testing datasets in the ratio 75:25 using M-number as seed.
* A cut off probability of 1/6 is chosen as a cost of 5:1 is mentioned.

#### Logistic Regression:

In logistic regression the probabilities or odds of the response are modeled as a linear combination of predictors

Performed Variable selection using Bidirectional Stepwise procedure with AIC criterion. Logit link was used as the Link function. Of the 20 variables 7 were selected by this procedure. The resulting model obtained was(categrorical variables in the below model were dummy coded)

response ~ chk\_acct + duration + purpose + credit\_his + saving\_acct +

other\_debtor + housing + other\_install + foreign + installment\_rate +

amount + sex + telephone

Model statistics are mentioned below:

|  |  |
| --- | --- |
| AIC | 720.84 |
| BIC | 882.54 |

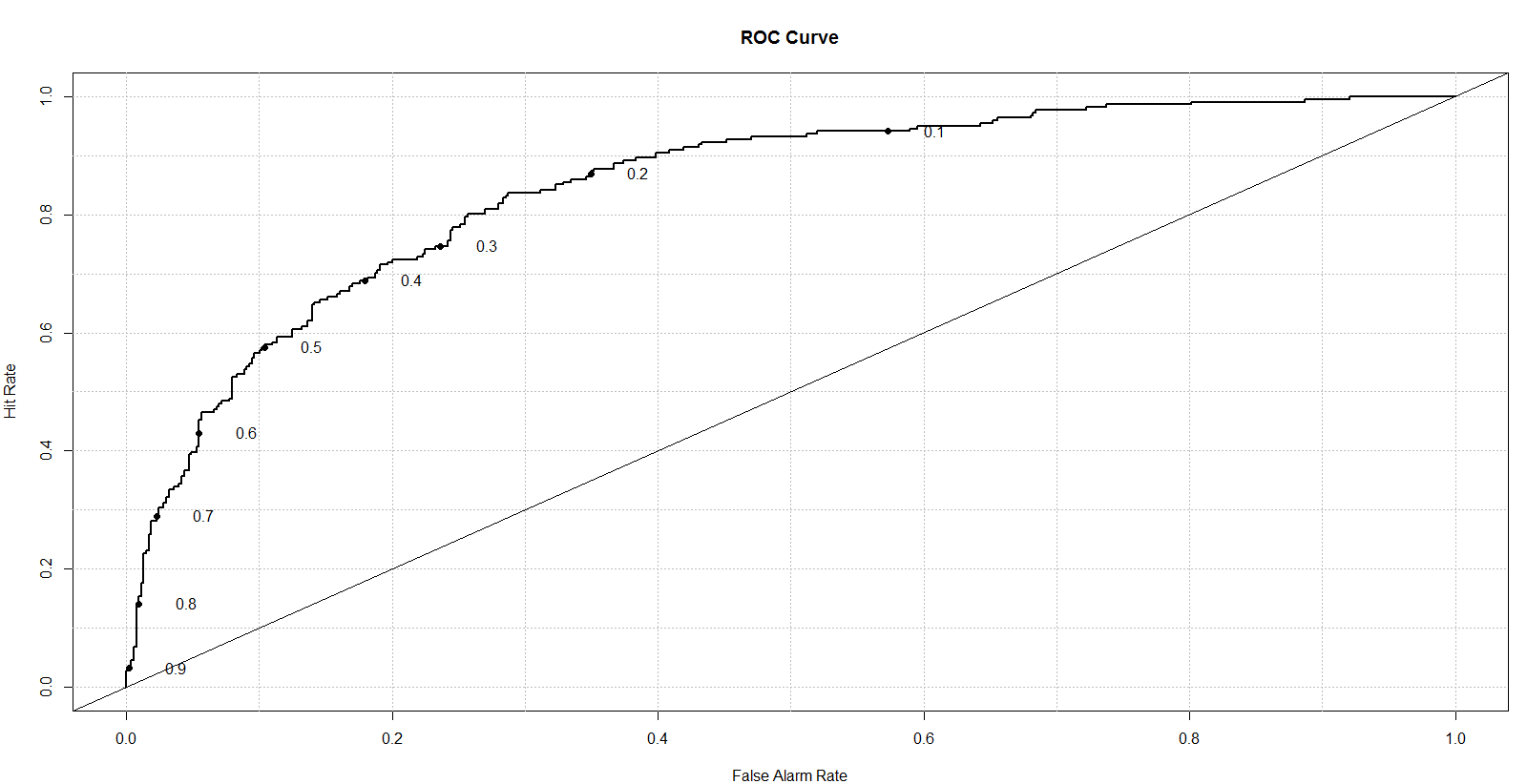
Below table gives us key in-sample and out-sample output values:

|  |  |  |  |
| --- | --- | --- | --- |
|  | Asymmetric Cost | Error Rate | AUC |
| In-sample | 0.42 | 0.31 | 0.846 |
| Out-Sample | 0.62 | 0.34 | 0.762 |

Confusion matrix on the training data with a cutoff value of (1/6) is given below:

|  |  |  |
| --- | --- | --- |
| Truth\predicted | 0 | 1 |
| 0 | 311 | 218 |
| 1 | 20 | 201 |

In-Sample ROC Curve (AUC=0.846)



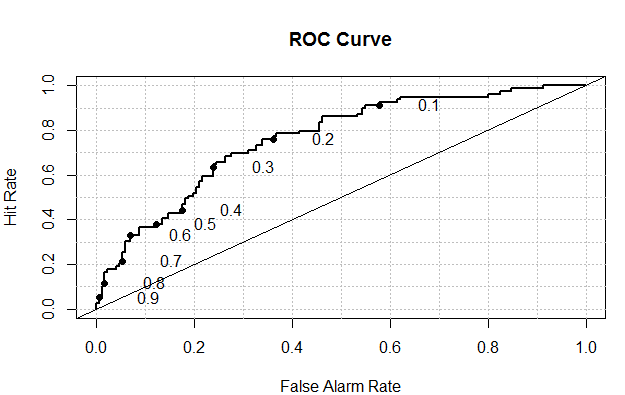
Confusion matrix on the test data with a cutoff value of (1/6) is given below:

|  |  |  |
| --- | --- | --- |
| Truth\predicted | 0 | 1 |
| 0 | 101 | 70 |
| 1 | 17 | 62 |

1= Bad Credit Risk, 0 = Good Credit Risk

As desired the number of bad credit risks misclassified as good is very low since they have a high cost.

Out-sample ROC curve (AUC =.762)



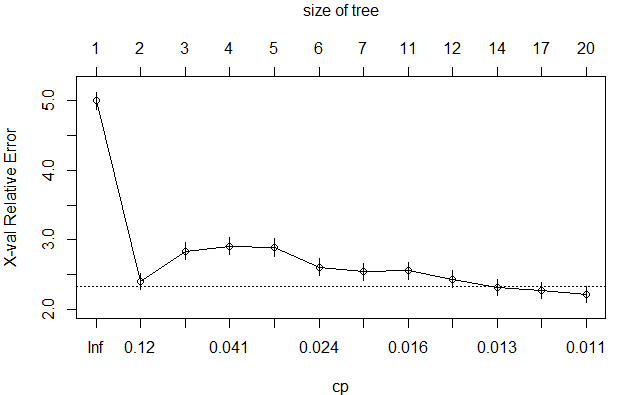
The Asymmetric cost obtained by 10 fold cross validation of the data is 0.53

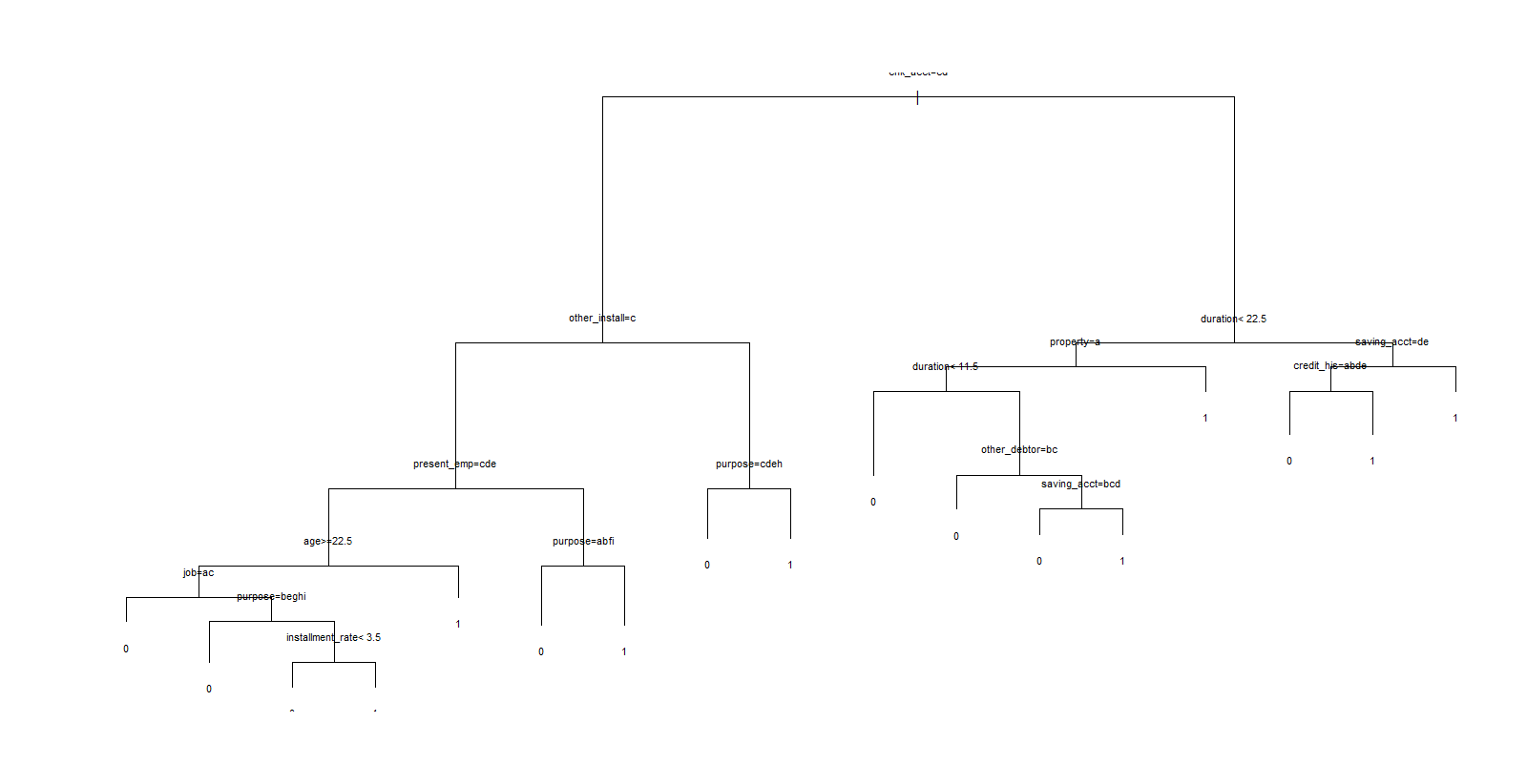
#### Classification Tree

Used the rpart package to build the classification tree.

To reduce the complexity of the tree pruning the tree is performed using plotcp() function.

The optimum value of Cp we obtained was 0.012 and size of the tree as 17.





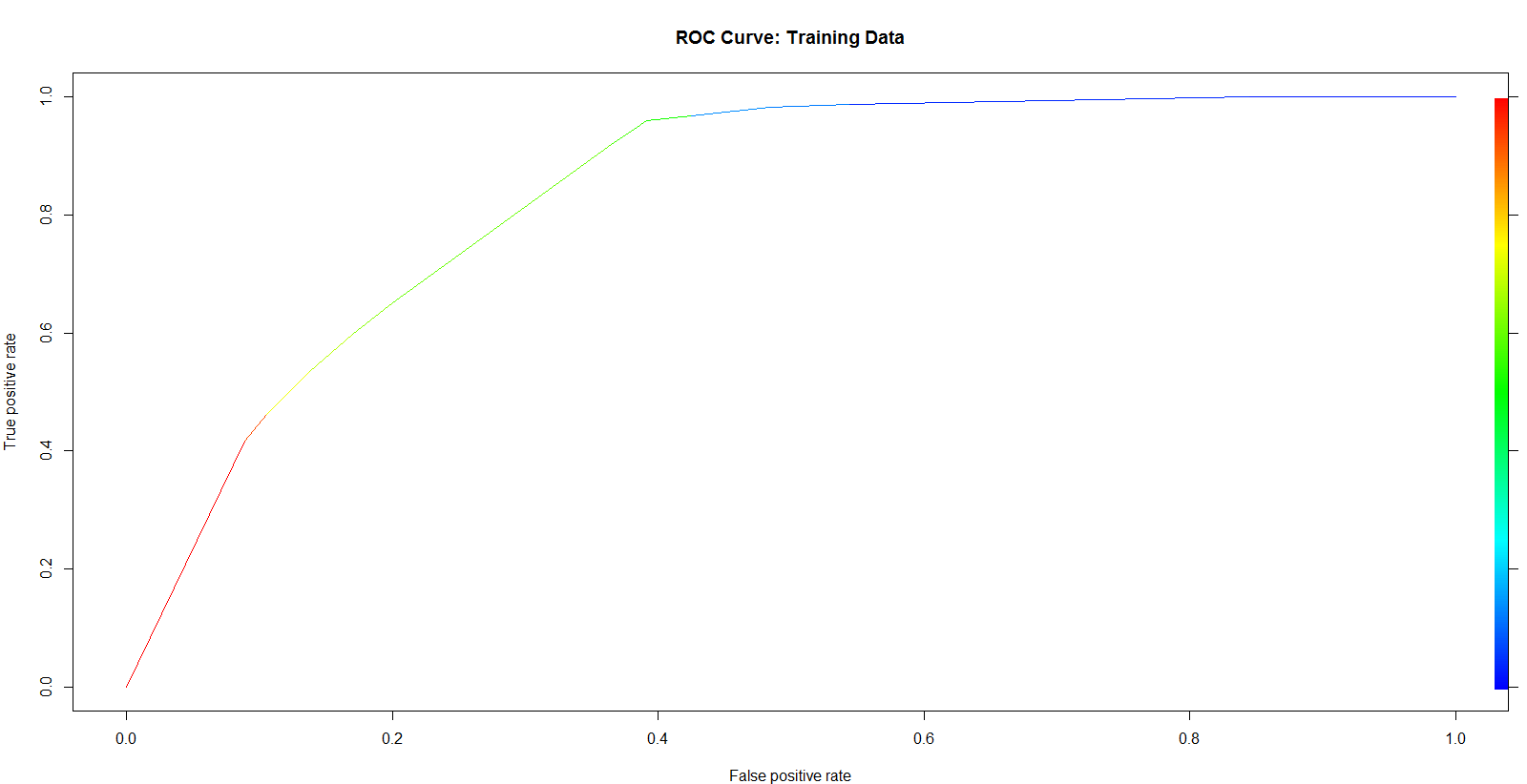
Below table gives us key in-sample and out-sample output values:

|  |  |  |  |
| --- | --- | --- | --- |
|  | Asymmetric Cost | Error Rate | AUC |
| In-sample | 0.52 | 0.28 | 0.836 |
| Out-Sample | 0.62 | 0.34 | 0.682 |

Confusion matrix on the training data with a cutoff value of (1/6) is given below:

|  |  |  |
| --- | --- | --- |
| Truth\predicted | 0 | 1 |
| 0 | 322 | 207 |
| 1 | 9 | 212 |

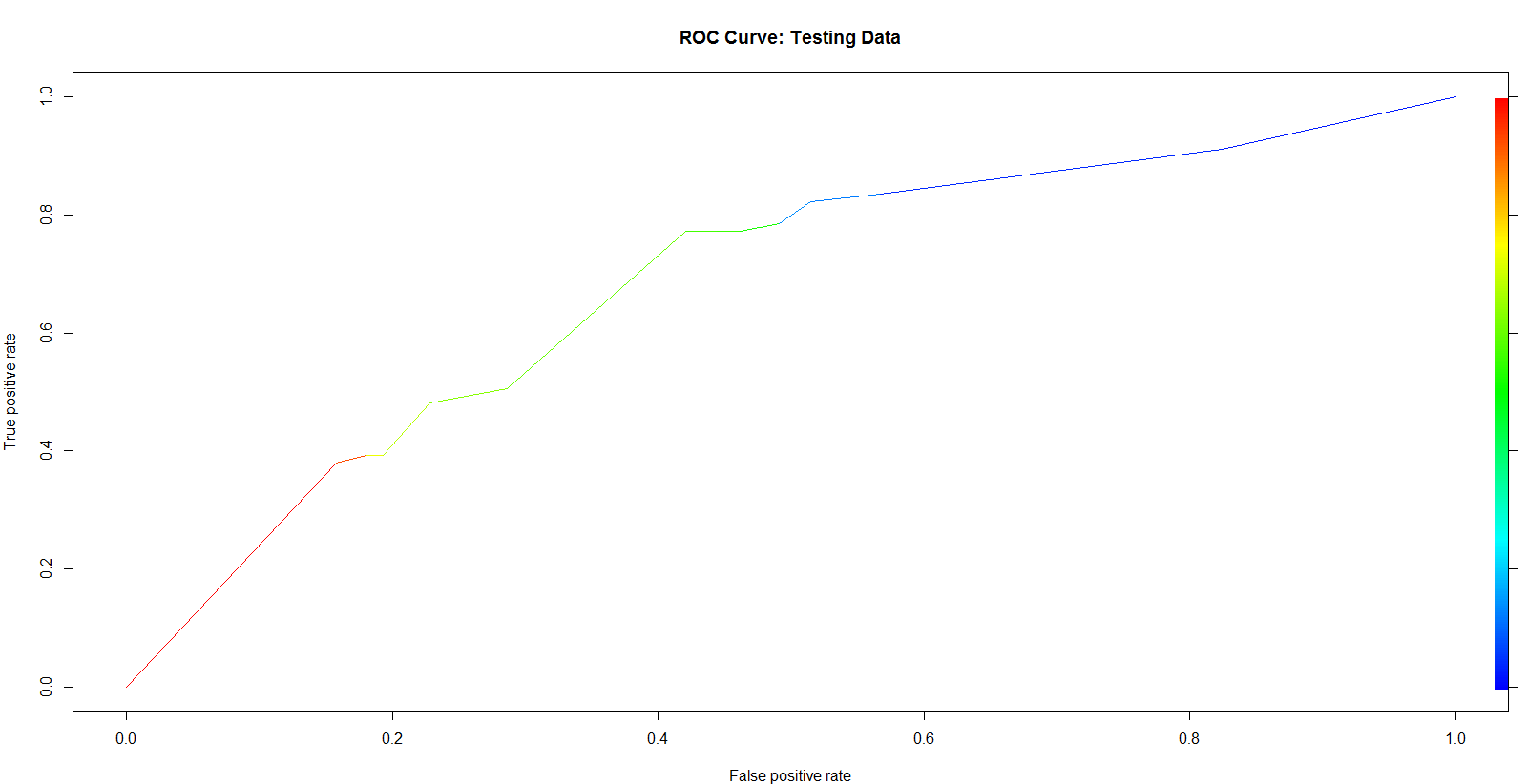
In-Sample ROC Curve (AUC=0.836)



Confusion matrix on the test data with a cutoff value of (1/6) is given below:

|  |  |  |
| --- | --- | --- |
| Truth\predicted | 0 | 1 |
| 0 | 92 | 79 |
| 1 | 18 | 61 |

out-Sample ROC Curve (AUC=0.682)



#### GAM Model

GAM is a generalized linear model in which the we predict using sum of smooth functions of the predictors and conventional linear predictors

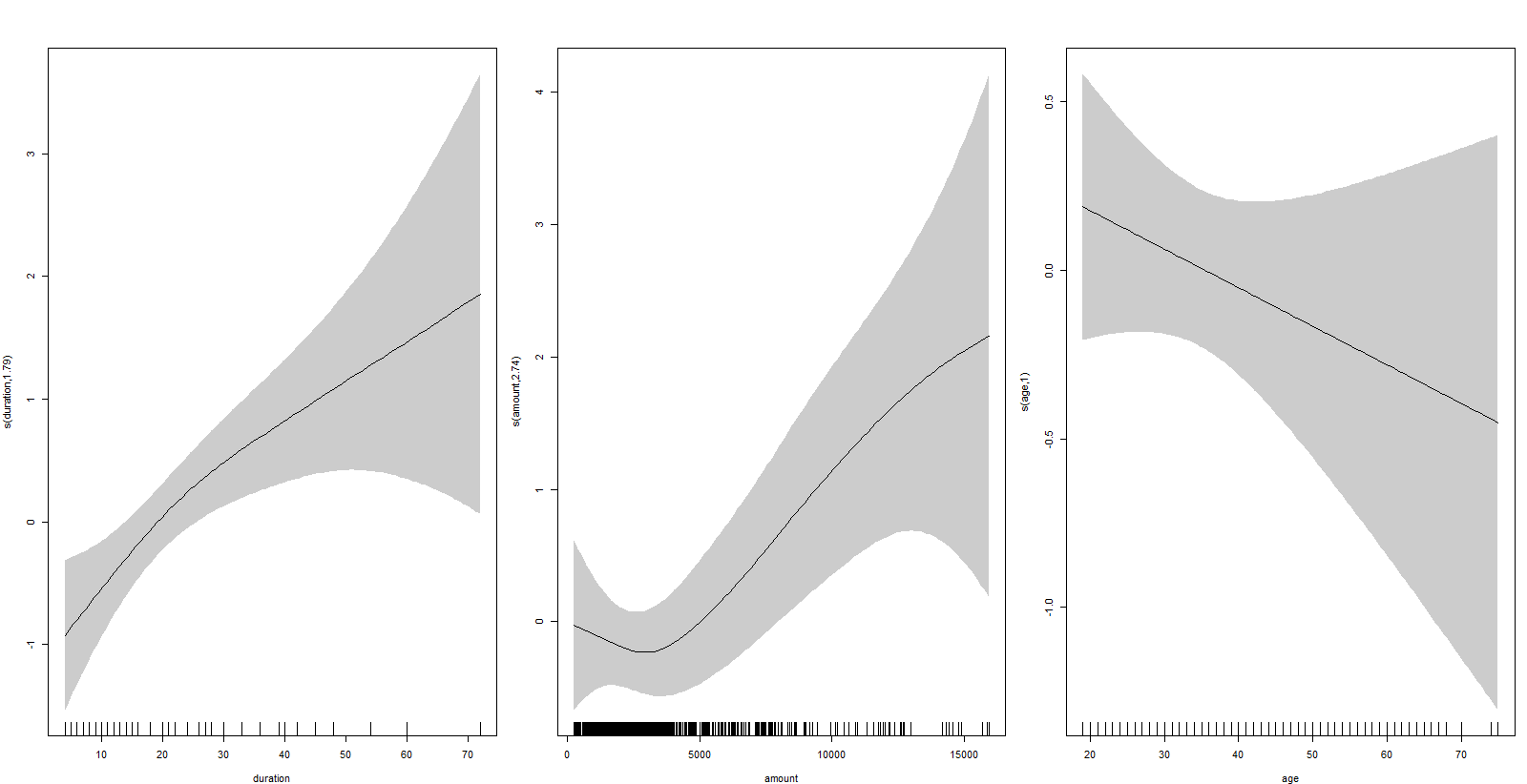
Used the implementation of GAM in the mgcv library to build a GAM Model. The smooth terms are specified in a gam formula using the s() function.

Initially ran the GAM model with all continuous predictors with smooth function.

The summary of Spline terms is given below

|  |
| --- |
| edf Ref.df Chi.sq p-value |
| s(duration) 1.502 1.859 13.305 0.000868 \*\*\* |
| s(amount) 2.611 3.293 14.860 0.002791 \*\* |
| s(age) 1.000 1.001 0.751 0.386248 |

From the above table we can observe that for the spline terms for predictors age the effective degrees of freedom is 1 suggesting that the term wouldn’t require spline smoothing.



The above figure plots the predictors vs their spline terms. If the plot is a straight line, it suggests that the spline terms can be dropped from the model

The Final model selected based on the above analysis is:

response ~ chk\_acct + s(duration) + credit\_his + purpose + s(amount) +

saving\_acct + present\_emp + (installment\_rate) + sex + other\_debtor +

(present\_resid) + property + (age) + other\_install + housing +

(n\_credits) + job + (n\_people) + telephone + foreign

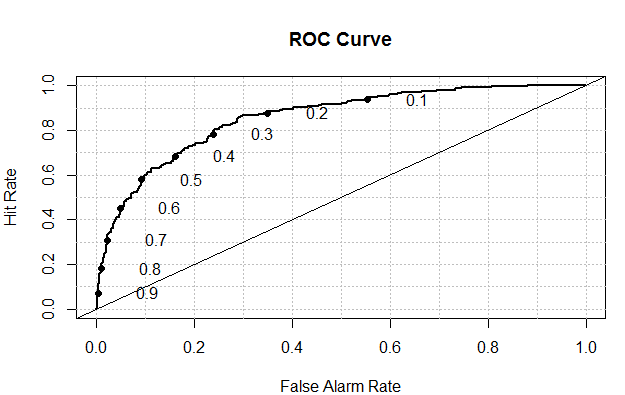
|  |  |  |  |
| --- | --- | --- | --- |
|  | Asymmetric Cost | Error Rate | AUC |
| In-sample | 0.42 | 0.30 | 0.854 |
| Out-Sample | 0.62 | 0.33 | 0.767 |

Confusion matrix on the training data with a cutoff value of (1/6) is given below:

Table 1: Training Data

|  |  |  |
| --- | --- | --- |
| Truth\predicted | 0 | 1 |
| 0 | 323 | 206 |
| 1 | 23 | 198 |

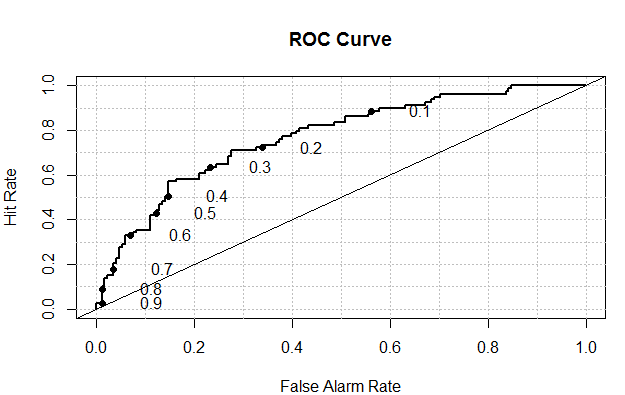
In-Sample ROC Curve (AUC=0.854)



Confusion matrix on the test data with a cutoff value of (1/6) is given below:

|  |  |  |
| --- | --- | --- |
| Truth\predicted | 0 | 1 |
| 0 | 106 | 65 |
| 1 | 18 | 61 |

Out-sample ROC curve (AUC =.767)



#### Linear Discriminant Analysis

Linear discriminant analysis (LDA) is a method used in statistics to find a linear combination of features that separates two or more classes of objects.

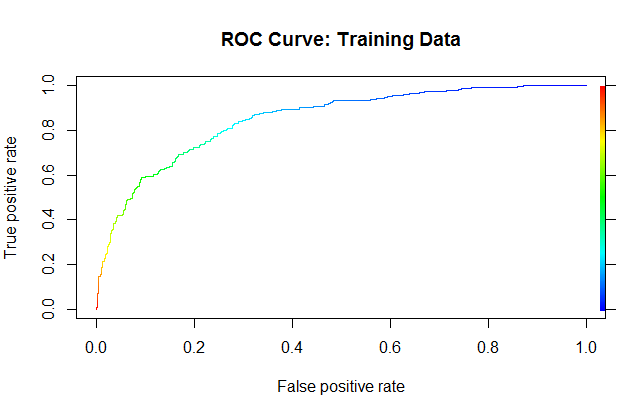
Below table gives us key in-sample and out-sample output values:

|  |  |  |  |
| --- | --- | --- | --- |
|  | Asymmetric Cost | Error Rate | AUC |
| In-sample | 0.42 | 0.30 | 0.849 |
| Out-Sample | 0.65 | 0.35 | 0.763 |

Confusion matrix on the training data with a cutoff value of (1/6) is given below:

|  |  |  |
| --- | --- | --- |
| Truth\predicted | 0 | 1 |
| 0 | 323 | 206 |
| 1 | 23 | 198 |

In-Sample ROC Curve (AUC=0.849)



Confusion matrix on the test data with a cutoff value of (1/6) is given below:

|  |  |  |
| --- | --- | --- |
| Truth\predicted | 0 | 1 |
| 0 | 102 | 69 |
| 1 | 19 | 60 |

1= Bad Credit Risk, 0 = Good Credit Risk

As desired the number of bad credit risks misclassified as good is very low since they have a high cost.

Out-sample ROC curve (AUC =.763)

