# Executive Summary – Regression – Boston Housing data

Goal and Background

In this problem, we use various modeling techniques such a linear regression, regression trees, and advanced tree methods like bagging, boosting and random forest, and General Additive model and Neural networks to predict the response. We use Boston housing data for the analysis. Our response variable the median value of owner-occupied homes which is continuous. Our goal is to evaluate different tree models and advance machine learning algorithms evaluate model performance.

Approach

To identify which variables best predict housing prices in Boston, we used seven modeling processes:

1. Dataset was split into training (75%) and testing (25%) by sampling from original dataset
2. Linear Regression: Stepwise variable selection method used to choose best model with highly significant variables and low average sum squared error.
3. Regression Tree: Regression tree performed with all variable. Pruning based on cp plot to come up with low prediction MSE and hence better fit.
4. Bagging: Picked optimal number of bags to reduce prediction error
5. Random Forest: Random Forest model performed to identify most important variables in predicting the response variable.
6. Boosting: Boosting performed to identify most important variables in predicting the response variable and see individual predictor impact on increase in response.
7. General Additive Model: General Additive model was performed to fit non-linear predictors by using smooth functions of predictors.
8. Neural Network: Very advanced machine learning method neural network was performed to obtain accurate predictions of medv.

Major Findings

We used Average sum squared error to find in-sample training error and out of sample prediction error. These values were used to compare different modeling techniques. Table 1 shows a summary of the in sample and out of sample average sum squared errors for each model type. We can conclude from this table that neural network gave us the lowest in sample error and boosting gives the lowest prediction error. In general, in sample error is lower than out of sample error, indicating the possibility of overfitting. The in-sample error of boosting is very small, however, we cannot interpret this error in case of boosting tree method.

|  |  |  |
| --- | --- | --- |
| **Method** | **Training Error (75%)** | **Prediction Error (25%)** |
| **Linear Regression** | 20.57 | 30.3 |
| **Regression Tree** | 14.83 | 29.32 |
| **Bagging** | 15.97 | 22.72 |
| **Random Forest** | 10.42 | 17.61 |
| **Boosting** | 0.0069 | 14.76 |
| **GAM** | 6.674571 | 18.38538 |
| **Neural Network** | 5.002204 | 17.29139 |

**Table 1**: Average Sum Squared Error for in sample prediction vs out of sample prediction error

# Executive Summary – Classification – German Credit scoring data

Goal and Background

In this report, we use various modeling techniques such a logistic regression, classification trees, and advanced tree methods like bagging, boosting and random forest, and General Additive model and Neural networks to predict the response. We use German housing data for the analysis. Our response variable is categorical – good or bad customer. Our goal is to evaluate different machine learning algorithms to evaluate model performance by using costs and misclassification rate.

Approach

1. Dataset was split into training (75%) and testing (25%) by sampling from original dataset
2. Logistic Regression: Stepwise variable selection method used to choose best model with highly significant variables and low AIC values.
3. Classification Tree: Classification tree performed with all variables and pcut of 1/6
4. Bagging: Picked optimal number of bags to reduce prediction error
5. Random Forest: Random Forest model performed to identify most important variables in predicting the response variable
6. Boosting: Boosting performed to identify most important variables in predicting the response variable and see individual predictor impact on increase in response.
7. General Additive Model: General Additive model was performed to fit non-linear predictors by using smooth functions of predictors.
8. Neural Network: Very advanced machine learning method neural network was performed to obtain accurate predictions.

Major Findings

We used Misclassification cost which was defined using asymmetric cost weights 5:1. These values were used to compare different modeling techniques. Table 2 shows a summary of the in sample and out of sample misclassification cost for each model type. We can conclude from this table that pruned classification tree gave us the lowest in sample error and General Additive Model gives the lowest prediction error. In general, in sample error is lower than out of sample error, indicating the possibility of overfitting.

|  |  |  |
| --- | --- | --- |
| **Method** | **In sample Missclassification Cost (75%)** | **Out of sample Missclassification Cost (25%)** |
| **Logistic Regression** | 0.465 | 0.54 |
| **Classification Tree** | 0.33 | 0.768 |
| **GAM** | 0.48 | 0.536 |
| **Neural Network** | 0.708 | 0.676 |

**Table 2**: Misclassification cost for in sample prediction vs out of sample prediction error

# Regression – Boston Housing Data

Data Description

We used Boston data from library MASS to perform analysis to predict factors impacting housing prices in Boston. The dataset has the following variables:

|  |  |  |
| --- | --- | --- |
| **Variable Name** | **Description** | **Role in modeling** |
| medv | median value of owner-occupied homes in \$1000s. | Response |
| lstat | lower status of the population (percent). | Predictor |
| black | 1000(Bk - 0.63)^2 where *Bk* is the proportion of blacks by town. | Predictor |
| ptratio | pupil-teacher ratio by town. | Predictor |
| tax | full-value property-tax rate per \$10,000. | Predictor |
| rad | index of accessibility to radial highways. | Predictor |
| dis | weighted mean of distances to five Boston employment centers. | Predictor |
| age | proportion of owner-occupied units built prior to 1940. | Predictor |
| rm | average number of rooms per dwelling. | Predictor |
| nox | nitrogen oxides concentration (parts per 10 million). | Predictor |
| chas | Charles River dummy variable (= 1 if tract bounds river; 0 otherwise). | Predictor |
| indus | proportion of non-retail business acres per town. | Predictor |
| zn | proportion of residential land zoned for lots over 25,000 sq.ft | Predictor |
| crim | per capita crime rate by town | Predictor |

**Table 3**: Variable description

## Linear Regression

First, I ran a linear regression model by using all predictors. The resultant model had a BIC of 2292.15. I used stepwise variable selection method by using BIC as a criteria as BIC penalizes model for complexity. The resultant model had a BIC of 2286.9 and adjusted R squared of 75.14 %. The following variables were shown as significant in the resulting model after variable selection.

|  |  |  |  |
| --- | --- | --- | --- |
| **Variable** | **Estimate** | **Pr(>|t|)** | **Significance** |
| lstat | -0.57852 | < 2e-16 | \*\*\* |
| rm | 3.70396 | 8.76E-16 | \*\*\* |
| ptratio | -0.875076 | 3.77E-11 | \*\*\* |
| dis | -1.432574 | 4.06E-11 | \*\*\* |
| nox | -13.934955 | 0.000194 | \*\*\* |
| black | 0.00853 | 0.004951 | \*\* |
| chas | 3.196521 | 0.002735 | \*\* |
| zn | 0.041792 | 0.003577 | \*\* |
| crim | -0.07345 | 0.015022 | \* |

**Table 4**: Variables selected in the final linear regression model when predicting medv.

The final model according to the training data is: **Medv ~ lstat + rm + ptratio + dis + nox + black + chas + zn + crim.**

In sample Average Sum Squared Error: 20.57. Prediction error on out of sample data: 30.30

Regression Tree

The following regression tree is made using training data from the boston dataset when medv was regressed on all the predictors (all variables included in model).

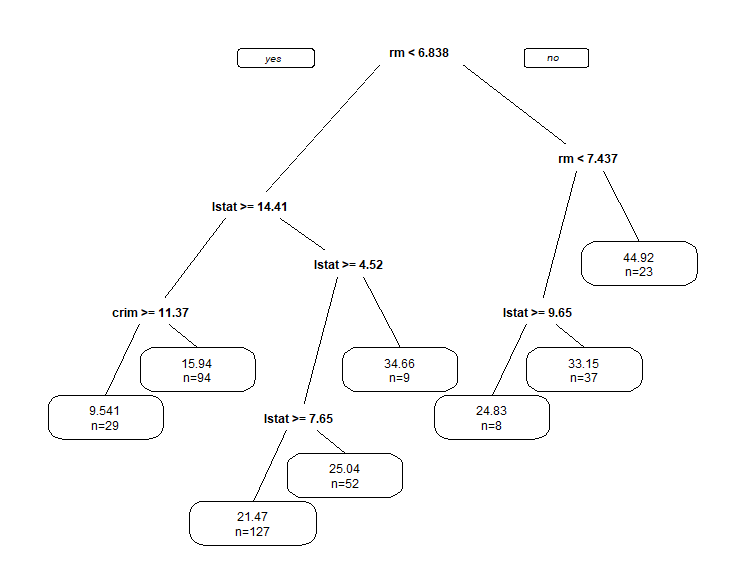


Figure 1: Regression Tree

When we tried to create a large tree with cp=0.001, we get the following cp plot. According to this cp plot, we identify that optimal cp to be used for tree pruning is 0.00147.

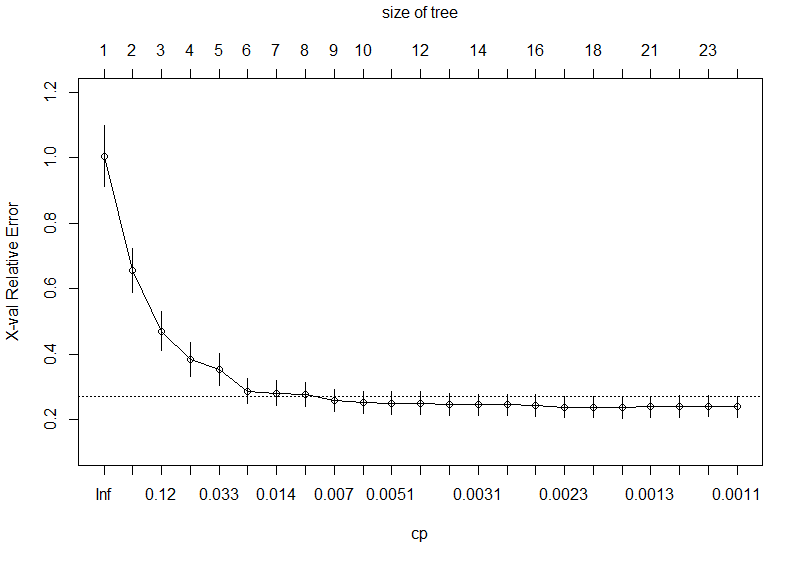


Figure 2: cp plot

In sample Average Sum Squared Error: 14.83. Prediction error on out of sample data: 29.32

## Bagging

I performed bagging using the training dataset with number of bags as 100. The ASE of the model with training data came out to be 15.97 whereas the prediction error was 22.72. when we plot ntree vs prediction error, we can see that after 80 number of bags, if we increase number of bags, the prediction error does not change much. Hence 80 can be optimal nbagg we need to select for bagging.

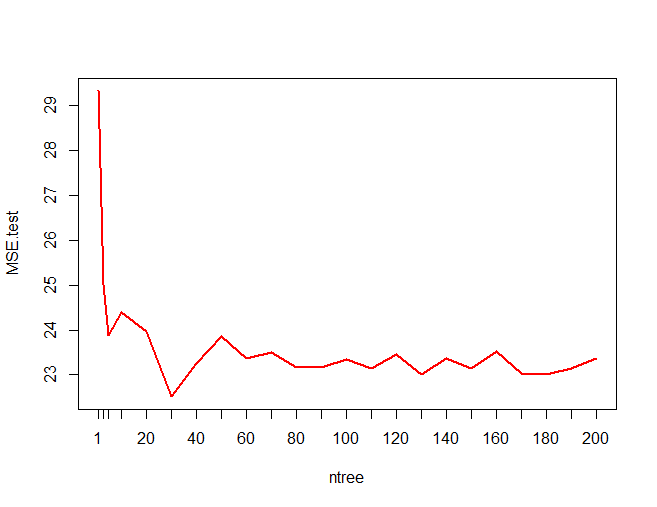


Figure 3: prediction error vs number of bags

In sample Average Sum Squared Error: 15.97. Prediction error on out of sample data: 22.72

## Random Forest

When I ran a random forest model on training dataset, I got the following result for importance of variables. According to this result, we can say the most important variables in predicting medv are rm and lstat, followed by ptratio and crim, as they have the largest increase in node purity.

|  |  |  |
| --- | --- | --- |
| **Variable** | **%incMSE** | **IncNodePurity** |
| rm | 33.815 | 9574.988 |
| lstat | 49.760 | 9281.054 |
| ptratio | 7.732 | 2533.159 |
| crim | 7.477 | 2188.065 |
| indus | 4.915 | 1704.980 |
| nox | 6.841 | 1558.093 |
| dis | 5.277 | 1409.114 |
| tax | 3.938 | 1133.069 |
| age | 3.376 | 909.386 |
| black | 1.652 | 594.014 |
| rad | 1.215 | 287.656 |
| chas | 0.476 | 194.646 |
| zn | 0.827 | 181.081 |

**Table 5**: Variable importance in predicting medv by random forest ensemble method.

The following figure shows us the MSE for testing dataset vs when we use OOB random forest method and calculate OOB error. Mtry indicates the number of variables selected by the model. We can see that, in general OOB has lower MSE as compares to testing dataset. When mtry =12, both the errors are closest to each other.

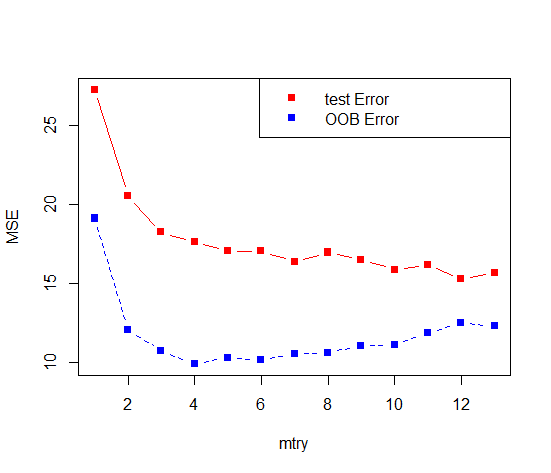


Figure 4: MSE for testing vs OOB error

In sample Average Sum Squared Error: 10.42. Prediction error on out of sample data:17.61

## Boosting

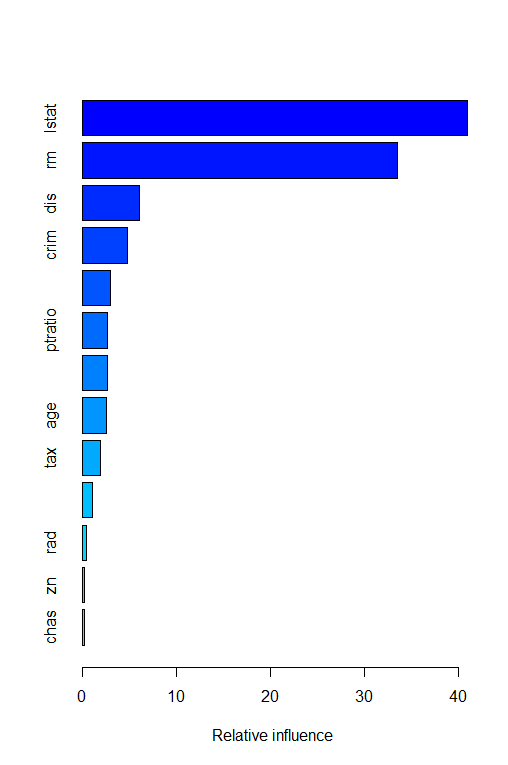
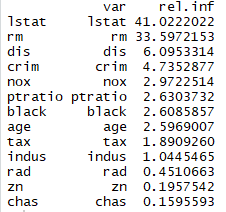
Upon performing boosting on the training dataset and specifying 10000 trees, we get the following variable importance (from most important to least important).

Figure 5: variable importance



The following two graphs show the impact of individual predictors lstat and rm on response (medv). We can see how response changes for different values of response.

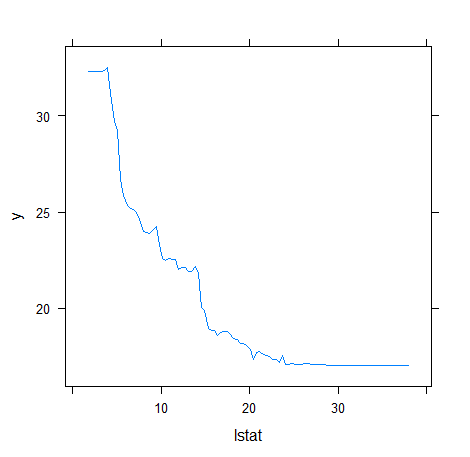
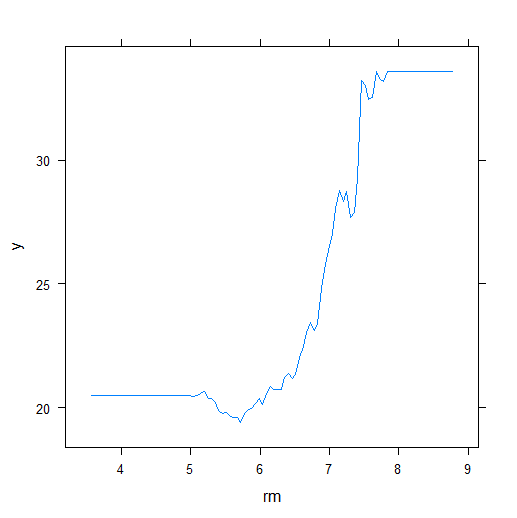
 

Figure 6: predictor impact on response

The following graph shows us how prediction error changes with increasing number of trees in the boosting model

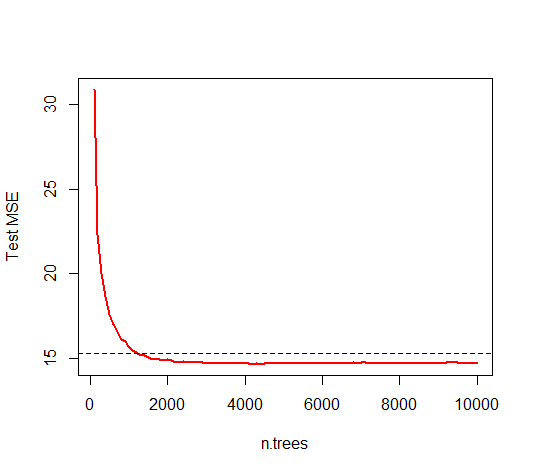


Figure 7: prediction error vs number of trees

In sample Average Sum Squared Error: 0.0069

Prediction error on out of sample data: 14.76

## GAM – General Additive Model

I looked at the scatterplots of medv vs all the predictors to visually see if there is any non-linear relationship. All the non-linear variables were added into the model with a smoothing function, and rad and chas were added normally as linear predictors(as observed from scatterplots). The initial model showed that edf of zn was 1. So we remove it from the smooth function and add it in as a linear variable. The following figure the variables with significant smooth terms that have been included in the model. The overall model adj R squared is 90.6%.

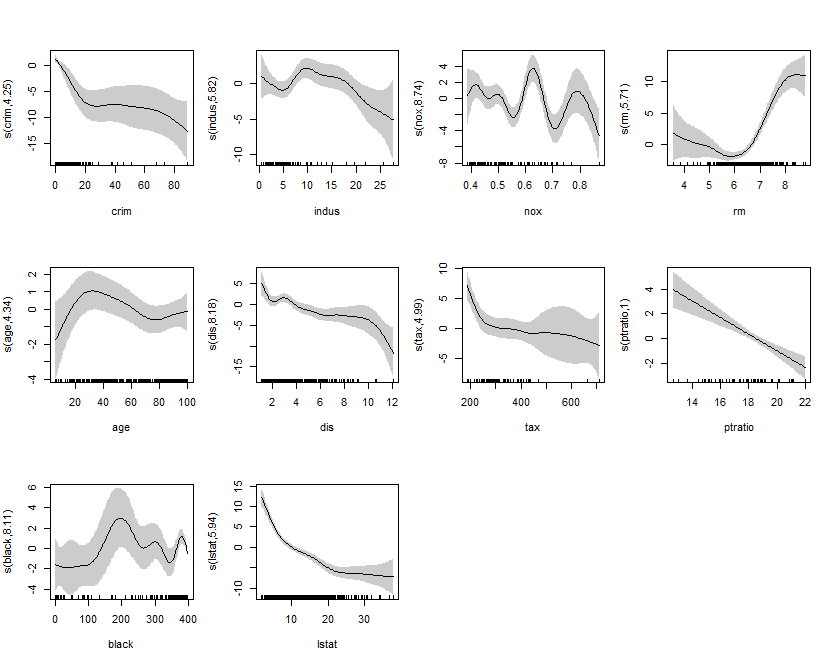


Figure 8: Smoothed non linear variables

In sample Average Sum Squared Error: 6.67

Prediction error on out of sample data: 18.38

## Neural Network

Artificial neural networks are computing systems inspired by the biological neural networks constitute animal brains. The neural network itself is not an algorithm, but rather a framework for many different ML algorithms to work together and process complex data inputs. A neural network assigns weights to the input variables established and runs multiple iterations to come up with assigned numbers of layers which consist of a combination of inputs and corresponding weights. We do not know what happens at these nodes. The result gives us accurate predictions.

To scale our response variable, we standardize it my using max and min. The network looks like the following.

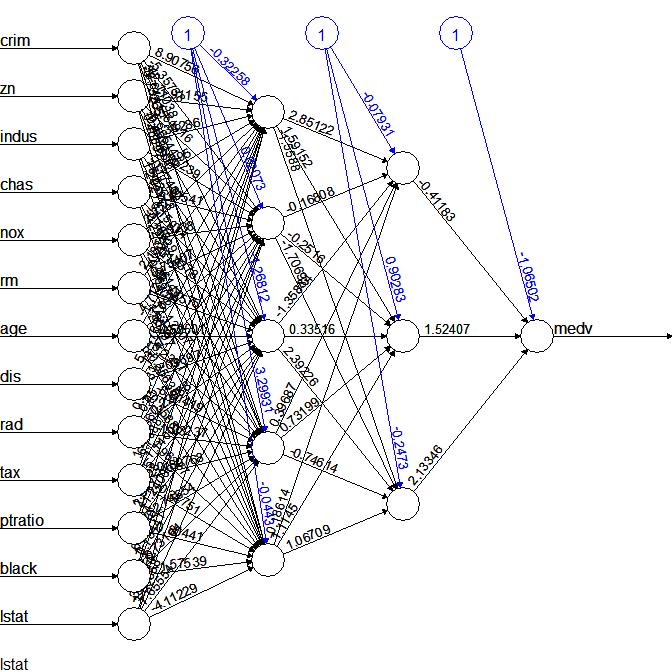


Figure 9: Neural Network for Boston Housing dataset

In sample Average Sum Squared Error: 5.002204

Prediction error on out of sample data: 17.29

# Classification - German Credit Score Data

Data Description

We used German credit score data to perform analysis to predict factors impacting whether the response variable is good(0) or bad(1). The dataset contains 21 variables and 1000 observations. The EDA of this dataset has already been completed in data mining I class. The data has 700 0 (good) and 300 1 (bad)

## Logistic Regression

I ran logistic regression model with all the variables in the beginning which gave poor fit and very few significant variables. Then I ran a stepwise model selection criteria on the logistic regression m odell to come up with the final variable selected model using BIC as a criteria. The final model is

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

**amount + saving\_acct + present\_emp + installment\_rate + sex +**

**property + age + other\_install + n\_credits**

The pcutoff for this dataset is 1/6 = .166. Based on this we calculate our misclassification rate.

Area Under the ROC curve is coming out to be 0.83 for the training dataset. However, AUC for testing dataset is 0.776

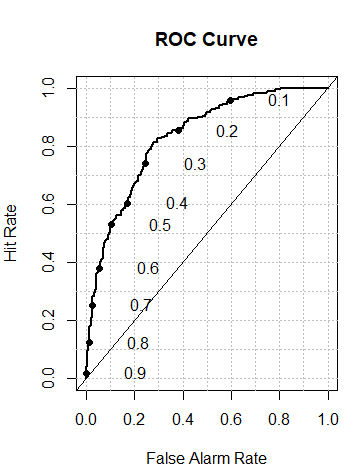


Figure 10 : Training ROC Curve

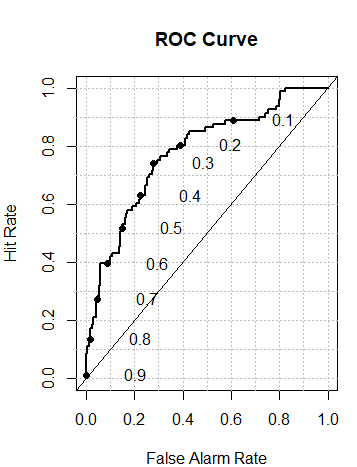


Figure 11: Testing ROC Curve

We define a cost function by using the assymetric costs by using ratio 5:! and use it to calculate the misclassification cost of each model.

In sample Misclassification Cost : 0.465, Out of sample Missclassification Cost: 0.54

## Classification Trees

We ran classification tree model by using 1/6 pcut with all predictors. The tree looks like the following.

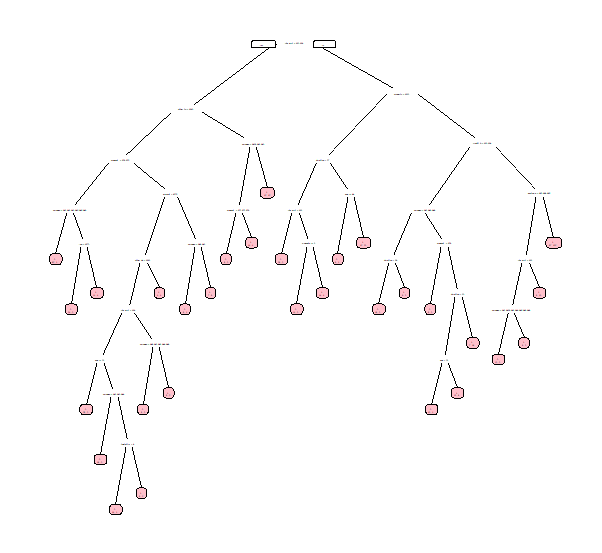


Figure 12: Classification Tree before pruning

Then we looked at the cp plot to identify the cp to prune the tree. Accorfing to the plot below, we choose 0.007 as our cp to prune the tree.

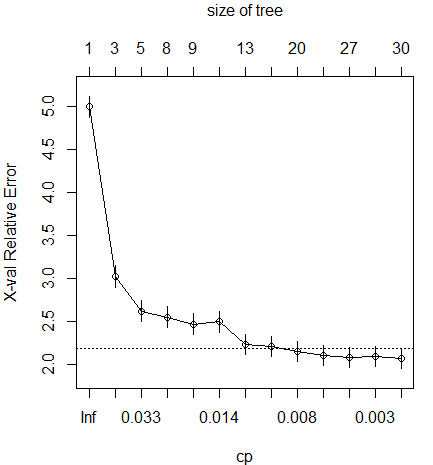


Figure 13: cp plot

Our pruned tree looks like the following

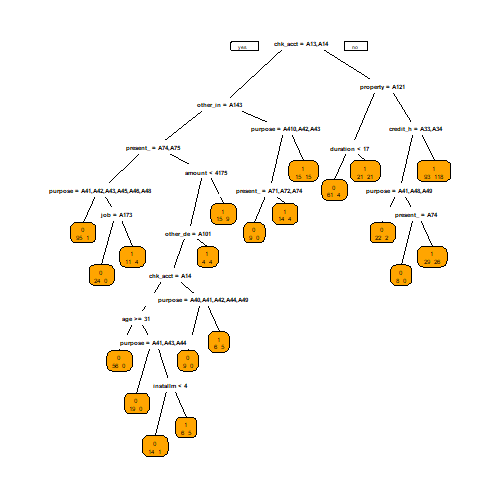


Figure 14: Classification tree after pruning

In sample misclassification Cost = 0.33

Out of sample misclassification cost = 0.768

The ROC Curves for training and testing prediction look like the following with AUC of 0.82 and 0.71 respectively.

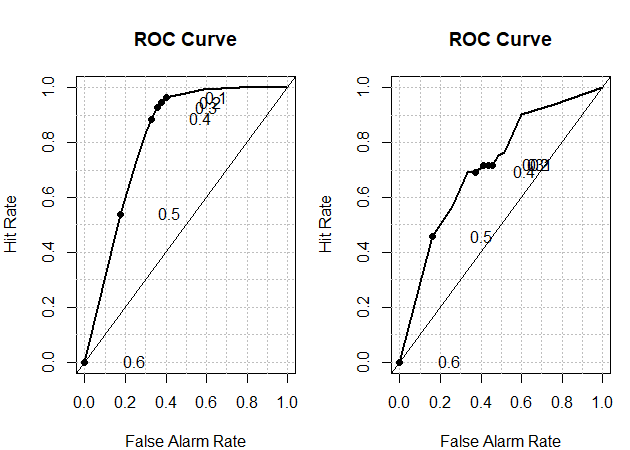


Figure 15: ROC curve for training and testing prediction

## General Additive Model

I fit a general additive model to the training dataset to account for non linear predictors while predicting good or bad response. In my initial model, I chose duration, amount and age to be smoothed with the smooth function while keeping all other terms linear. My initial model results showed that age has an edf of1. Hence I moved age into a linear term in the model while applying smooth function to only duration and amount. The resulting model looks like this:

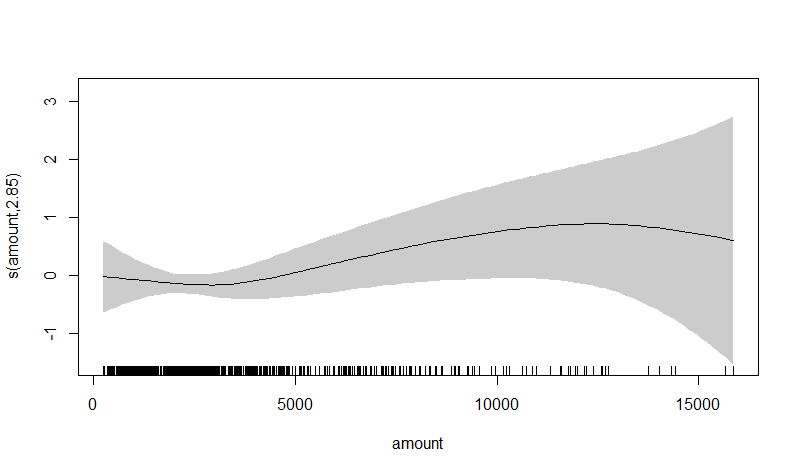
**response ~ chk\_acct + s(duration) + credit\_his + purpose +**

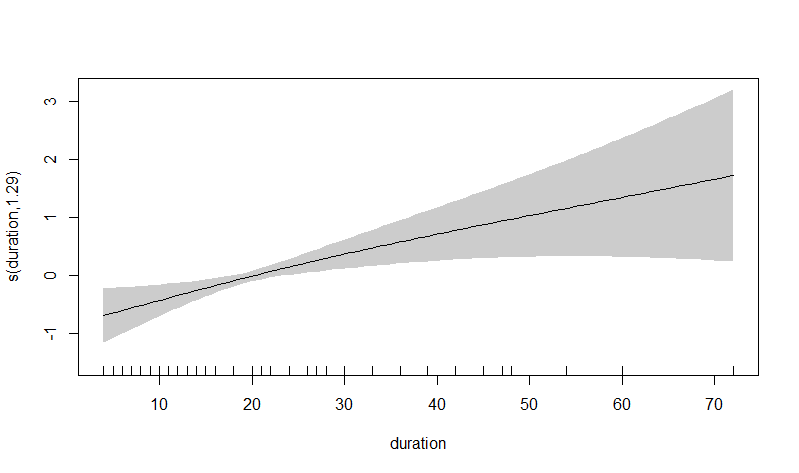
**s(amount) + saving\_acct + present\_emp + installment\_rate +**

**sex + other\_debtor + present\_resid + property + age + other\_install +**

**housing + n\_credits + telephone + foreign**

The smoothed variables look like the following:

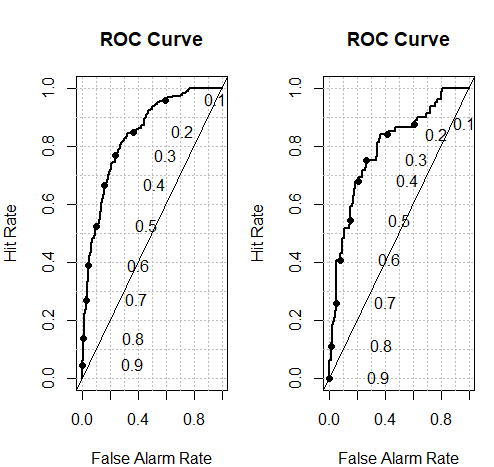




The Misclassification Cost for in sample data = 0.48

Misclassification Cost of out of sample data = 0.536

ROC curves for in sample and out of sample predictions look like the following with AUC of 0.83 and 0.79 respectively.



## Neural Network

The neural network for the training dataset looks like the following. The in sample missclassification cost is 0.708 and out of sample missclassification cost is also 0.676.

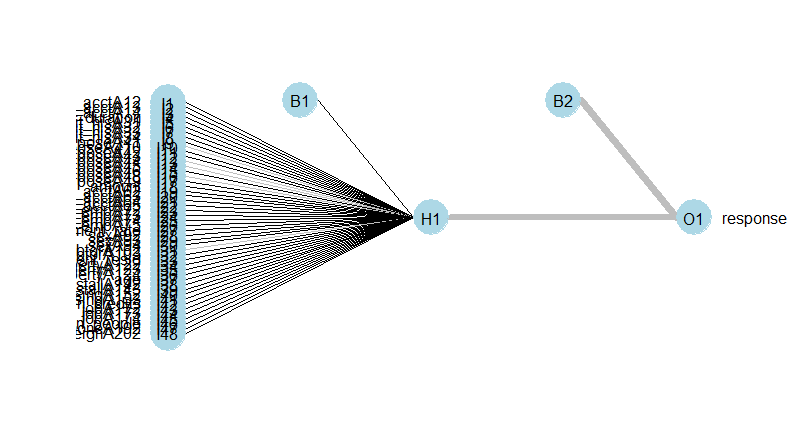


Figure 16: Neural Network