Comparison of Different Methods for predicting Loss Given Default

Group 14Balan, Mahesh  
Boppidi, Venkat Kanishka  
Reddy Dulay, Kuldip  
Subramanian, Rajarajan  
Ramakrishnan, Siva

Contents

[Objective: 3](#_Toc480746578)

[Introduction: 3](#_Toc480746579)

[Data Set: 4](#_Toc480746580)

[Initial Data Exploration: 5](#_Toc480746581)

[LGD distributions across year, collateral & industry: 8](#_Toc480746582)

[Relationship between continuous variables: 9](#_Toc480746583)

[Linear Regression 10](#_Toc480746584)

[Fractional Response Regression(FRR) 12](#_Toc480746585)

[Regression Tree 13](#_Toc480746586)

[Neural Networks 17](#_Toc480746587)

[Random Forest 21](#_Toc480746588)

[Conclusion: 23](#_Toc480746589)

# Objective:

The primary objective of the project is to analyze different ways of predicting Loss Given Default(LGD) and compare the results among the various methodologies. LGD is one of the two crucial components in estimating the expected credit loss rate on any pool of debt observed over a period. Expected Loss is defined as a function of Probability of Default(PD) and Loss Given Default(LGD) The LGD takes on continuous values between 0 and 1. Therefore, the challenge is to perform techniques that bounds the continuous outcome within a range. Methods to be tested:

* Fractional Response Regression
* Regression Tree
* Neural Network
* Random Forest

# Introduction:

The data set used to perform the study is from the Moody’s Ultimate recovery database. The data contains US corporate default events with more than $50 million in debt. The general accepted methodologies for modelling LGD are

* Settlement Method
* Trading Price Method
* Liquidity Method

The settlement method is the most preferred and is used in performing the study.

# Data Set:

The data set contains about 29 variables.

**Key Variables Definition**

Seniority Index: It is a variable that is used in the Moody’s loss calculations. Seniority Index is calculated using the percent above variable.

Distance to Default: This variable indicates the number of Standard Deviations, the instrument is away from default.

Ind: The Ind variable indicates the different sectors/industries that are analyzed in the study. They are consumer non-durables, consumer durables, manufacturing, energy, chemical, business equipment, telecommunications, utilities, shops, health care, financial institutions. Any industry that doesn’t fall in the above mentioned 11 industries fall under the others category.

Lgd: The response variable in the study, is defined as the proportion of loss expected given a default has already occurred.

Default\_rate: The variable defines the default rate observed over the last 12 months.

Mktret: The variable indicates the returns expected on investing in the instrument.

Collateral: Variables such as Capital stock, Equipment, Guarantees, Intellectual, Intercompany debt, Inventory Cash debt, most assets, other assets, unsecured, second lien, third line all fall under the collateral category. A total of 11 categories of collaterals are available.

Instrument type: Variables such as Senior Secure, Junior, revolver, Term loan, senior unsecure, senior subordinated, subordinated fall under the instrument type.

# Initial Data Exploration:

Default counts across different instruments:

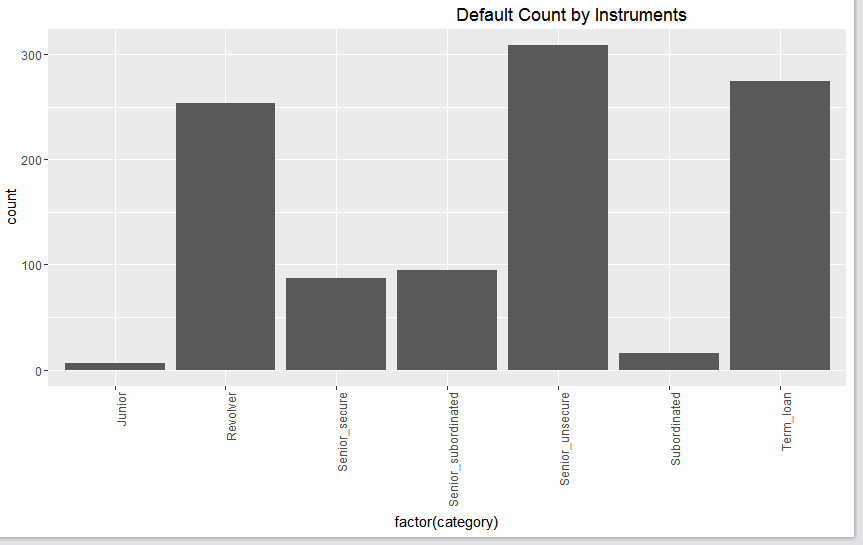


Figure Default counts across different instruments

As it is evident from the bar graph above, the senior unsecured instrument has the maximum number of default in the study, which is closely followed by the term loan and revolver product.

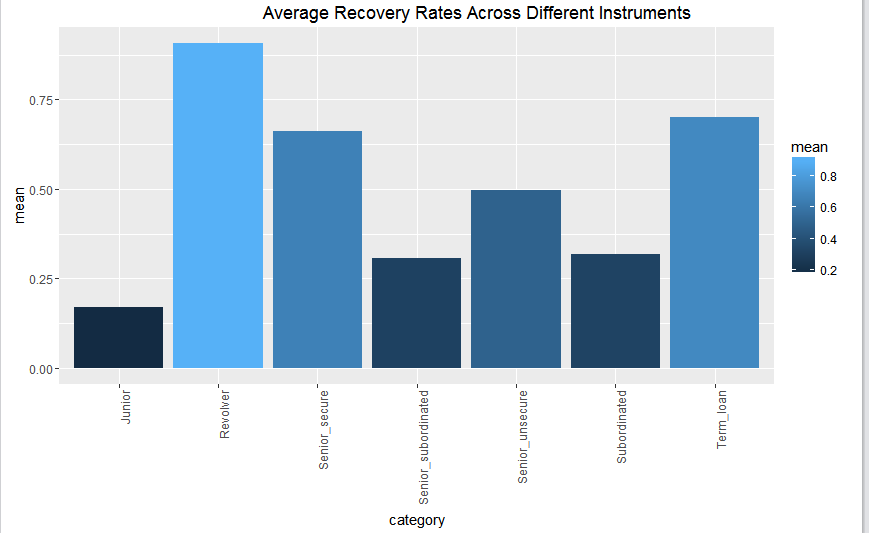


Figure Average recovery rate

The average recovery rates are the highest for revolver, followed by senior secured and term loan. The least recovery is for junior and the unsecured/subordinated instruments.

Collateral: The Unsecured ones are the most default followed by the most assets segment.

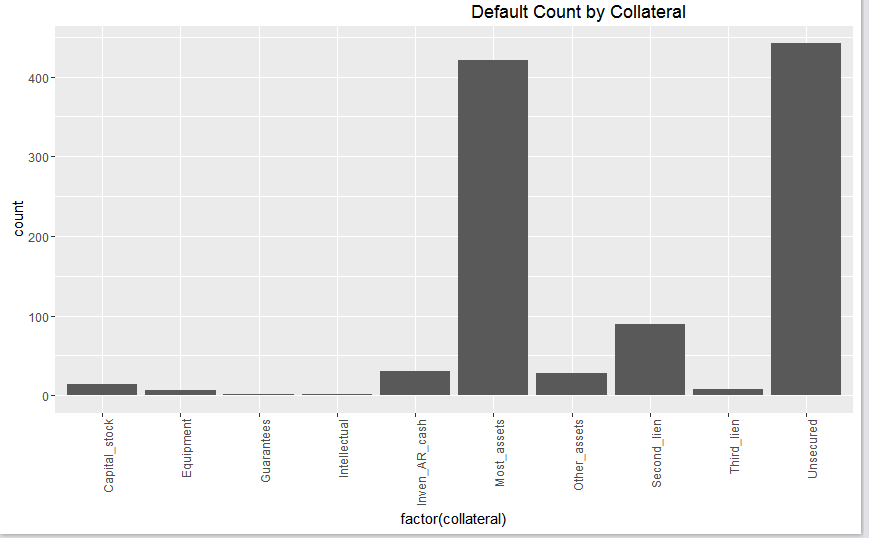


Figure Collateral

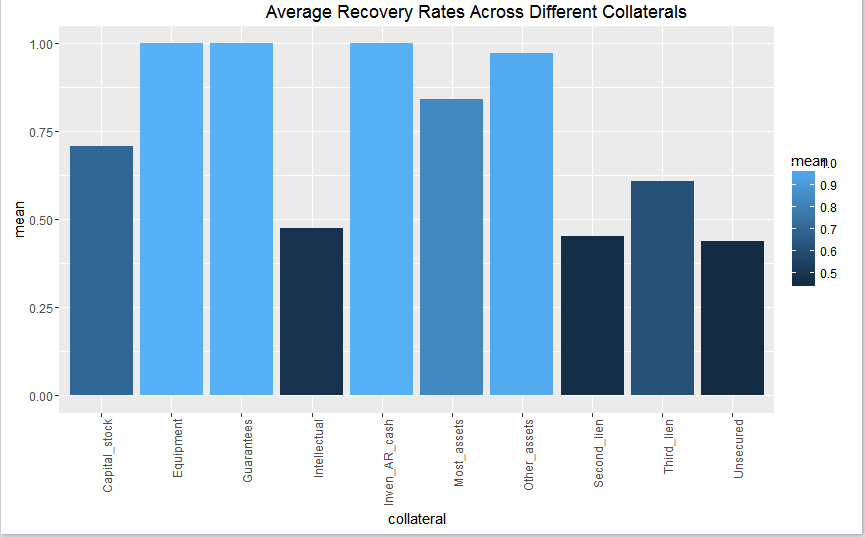


Figure Average Recovery rates

The least recovery is for the unsecured collateral and the maximum recovery is found for the equipment and guarantee. It is intuitive because the chances to recover with no or less collateral is always low compared to other types such as equipment, guarantees and inventory cash et al.

Industry: The category ‘others’ has maximum defaults. The second and third industry with the maximum defaults are Manufacturing, and consumer non-durable goods.

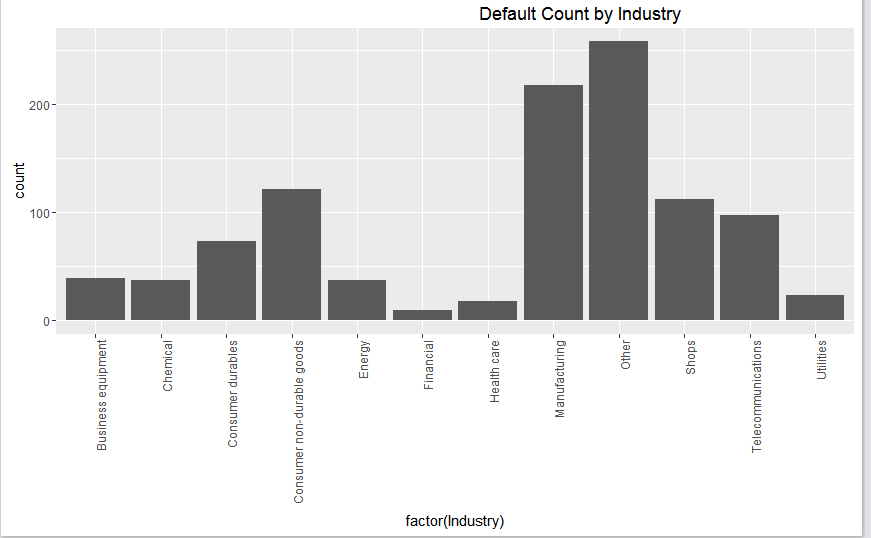


Figure Defaults by industry

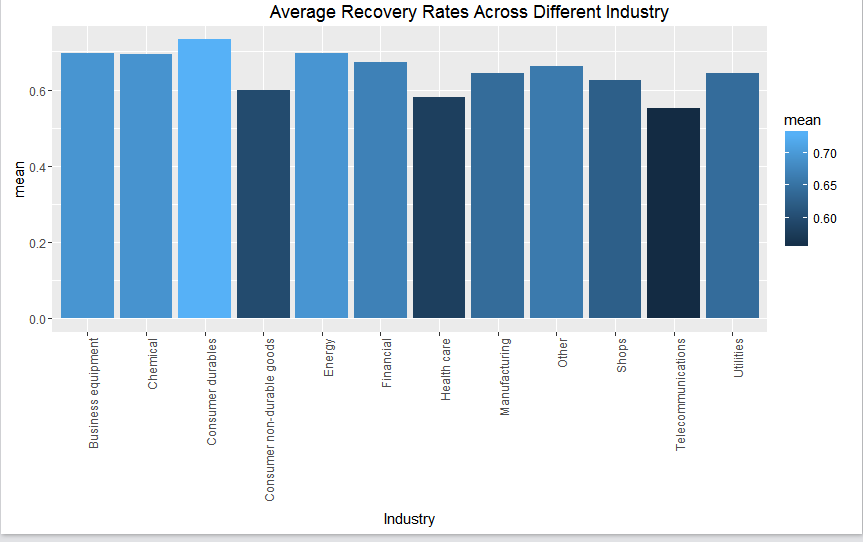


Figure Average recovery rates across industries

The recovery across industries is almost consistent barring consumer non-durable goods, healthcare, and telecommunications.

### LGD distributions across year, collateral & industry:

**Year wise:** The below diagram shows the distribution of LGDs across different years.

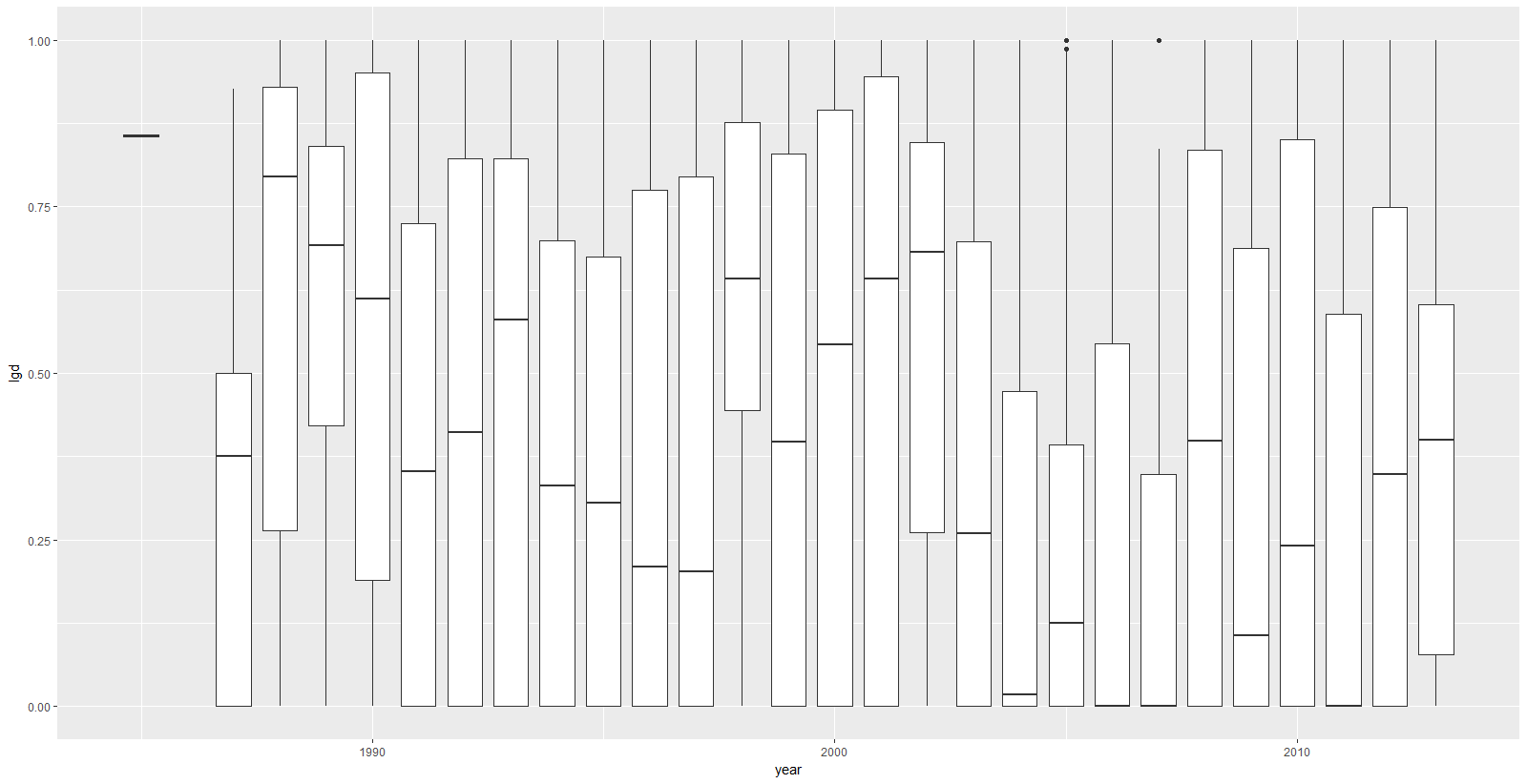


Figure 7 LGDs year wise

The LGD distribution is almost even for values between 0 and 1 for post 2008 cases. The distribution at values 1 and 0 is high in both pre-and post-2008 cases.

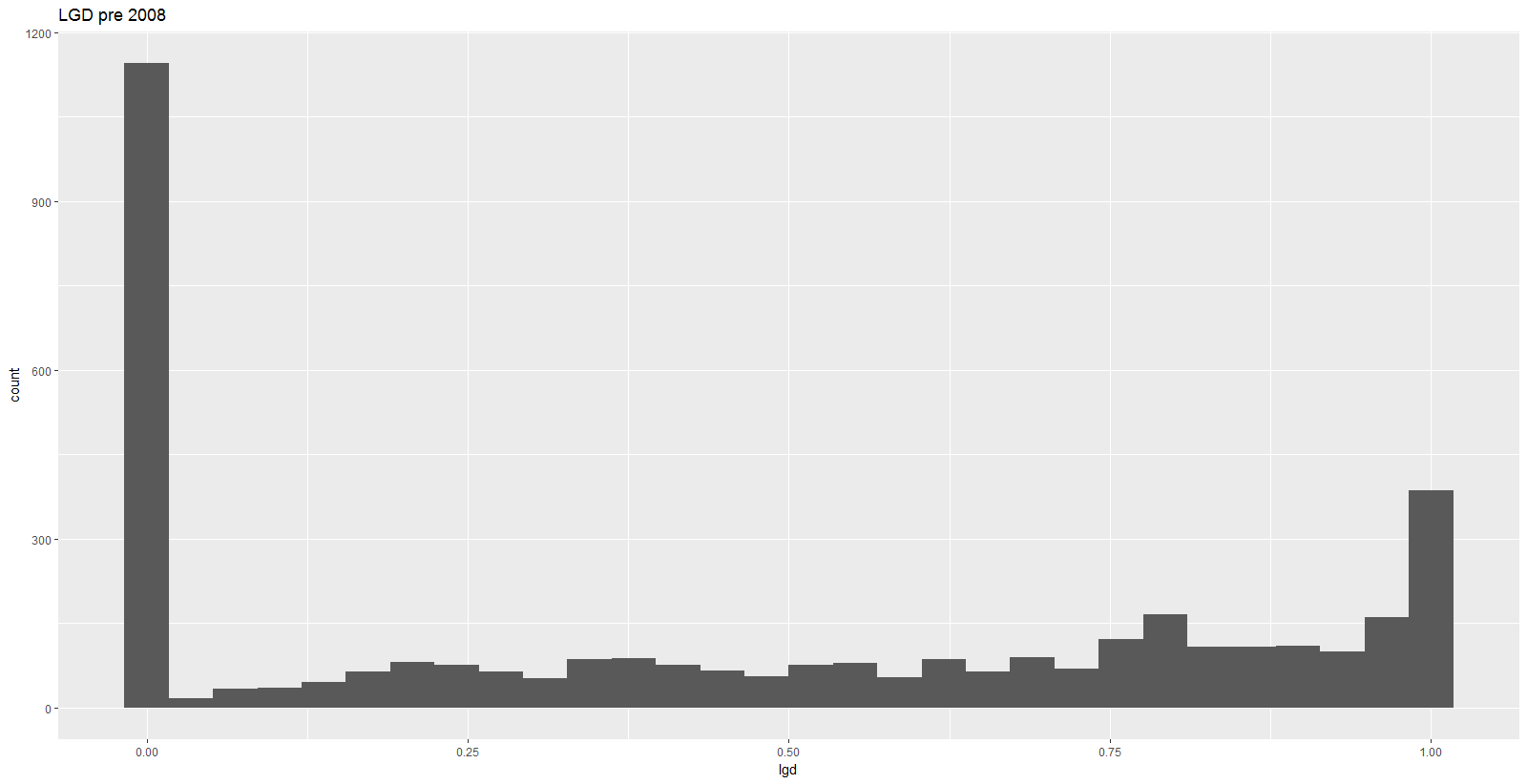
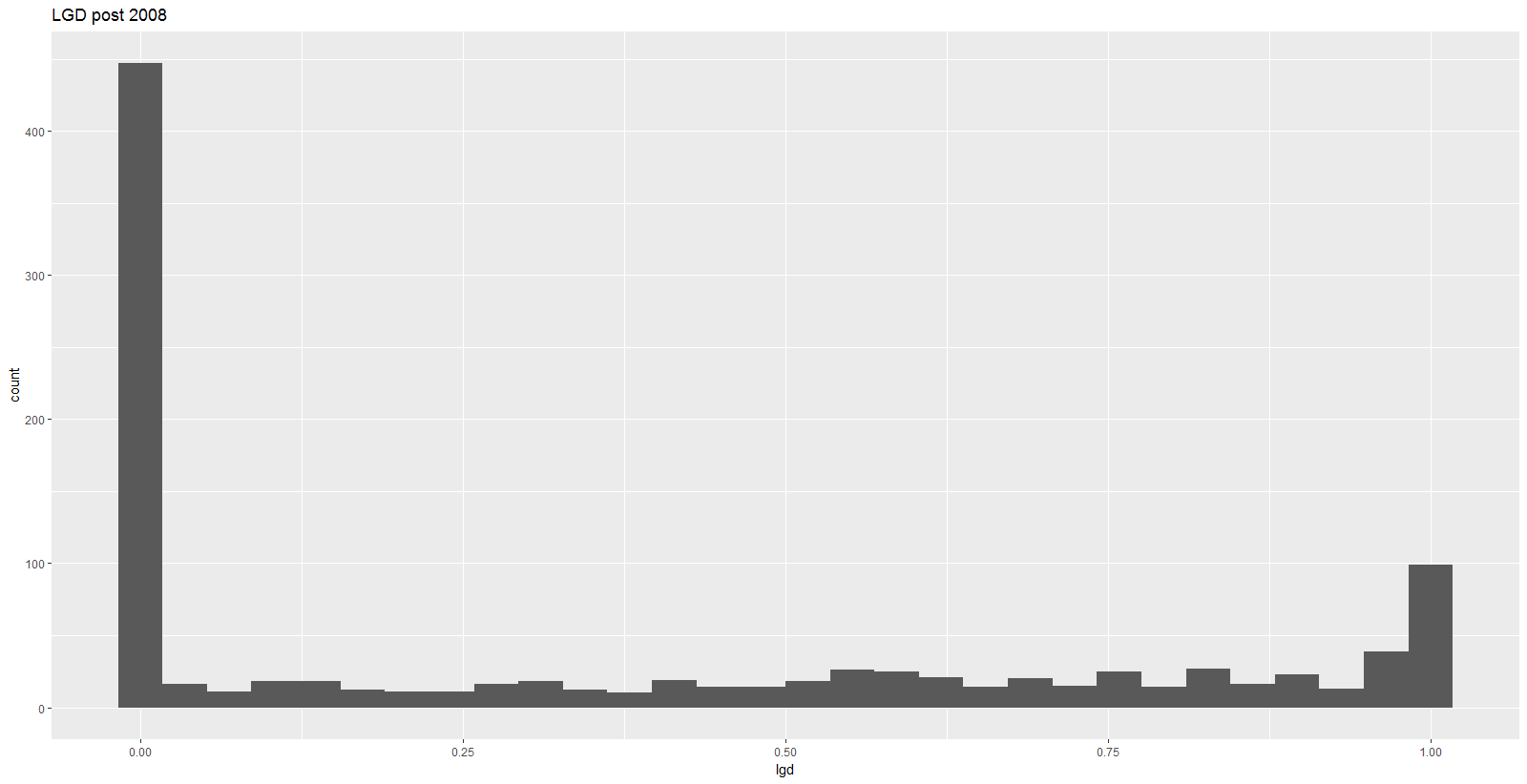
 

Figure LGD pre and post 2008

**Industry wise & Guarantee type**

The mean and median LGD values are greater in pre-2008 for almost all industries except utilities.

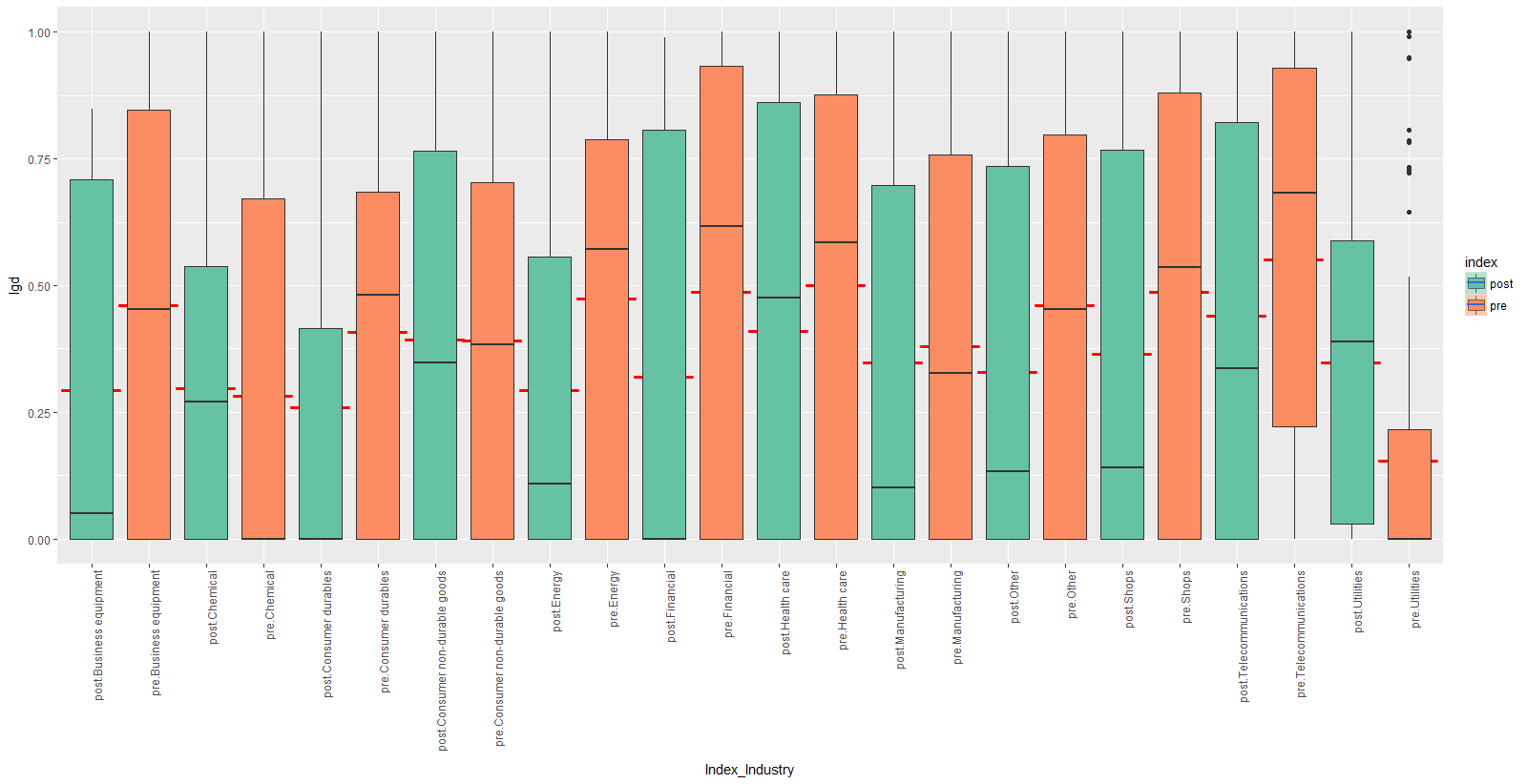


Figure Index and Industry

In second lien cases mean and median LGDs are higher in post 2008 cases.

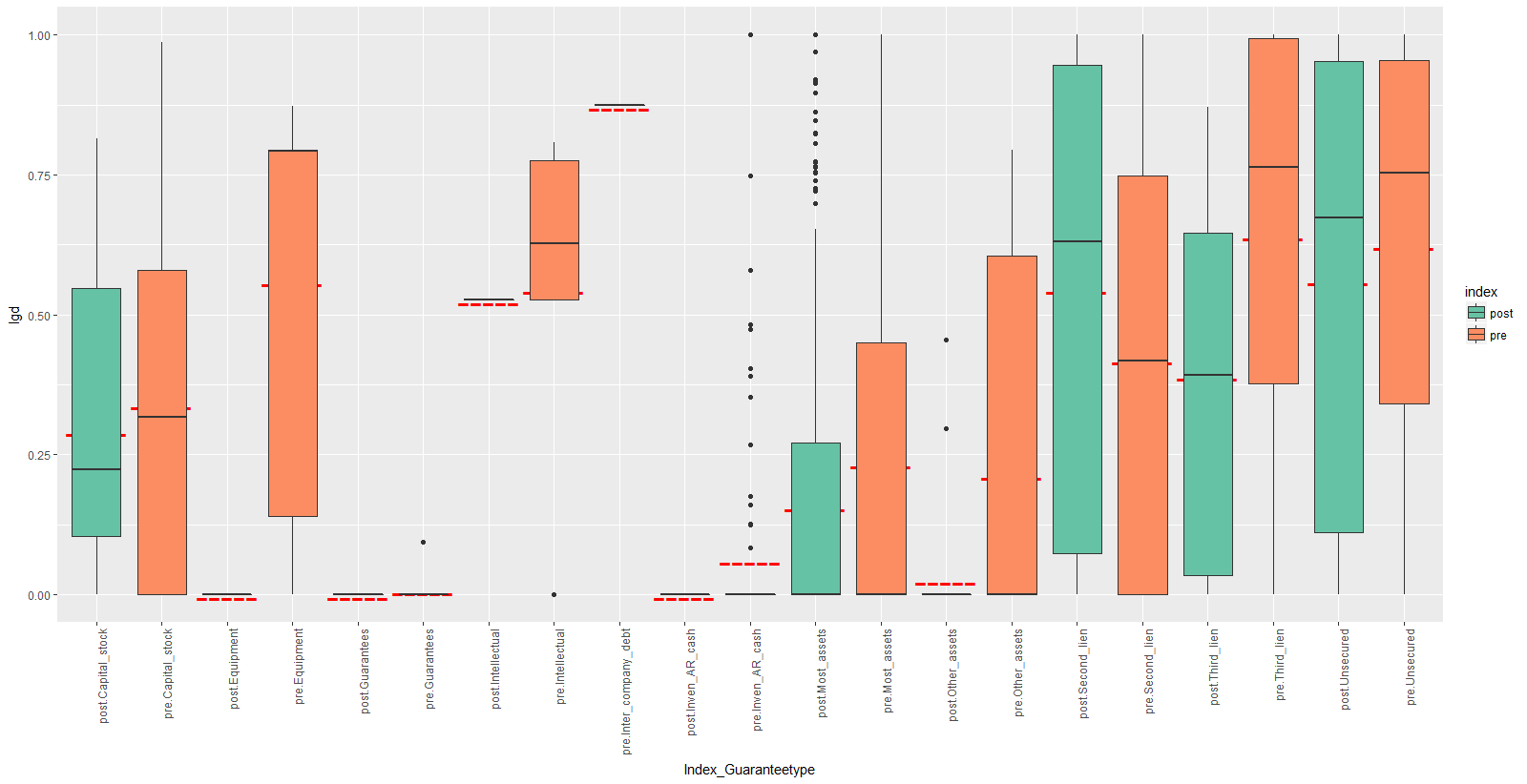


Figure 10 Guarantee Type

## Relationship between continuous variables:

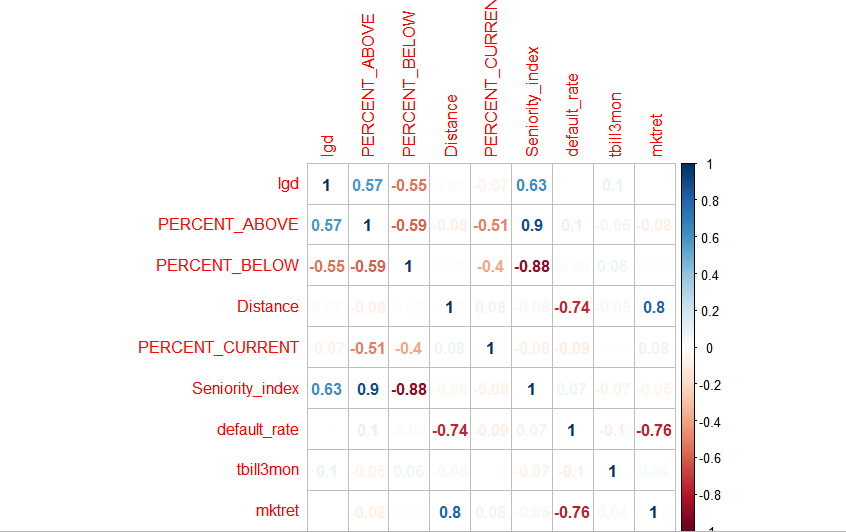


Figure Correlation matrix

The takeaways from the correlation matrix of the continuous variables are

1. The key variables that has the potential impact to estimate the LGD are seniority index, percent above and percent below.
2. There seems to be high correlation between seniority and percent above and below
3. The distance to default is negatively correlated with the default rate, which is very intuitive because the distance to default signifies the number of standard deviations away from the actual default
4. The market return and the default rates are inversely related showing a greater market return for the industry signifies a lesser chance to default
5. LGD is negatively correlated with percent below as the chances for losses goes down drastically with a higher proportion in the percent below as opposed to the percent above

# Linear Regression

The main objective is to build a linear regression model for three datasets- entire dataset containing observations before and after year 2008 (4793 observations), existing ‘pre’ dataset with 3751 observations and new ‘post’ dataset 1042 observations. Sampling is done for training and testing datasets in the ratio 75:25. That is the dataset is split into 75% training and 25% testing sets using a fixed seed value (14).

**Results:**

Regression was performed taking into consideration all of the predictor variables to begin with. After this, stepwise variable selection was executed to select the best set of predictor variables. Model selection was based on AIC and BIC criteria. The model with the lowest AIC and BIC values was selected.

**Final Model Complete Dataset**

lgd ~ Seniority\_index + ind + Unsecured + tbill + mktret + Senior\_unsecure

Percent\_Above + Revolver + Utility\_dummy + Second\_lien + Equipment +

Third\_lien + Inven\_AR\_cash + Guarantees + Intellectual + Term\_loan +

Subordinated

**Final Model (Pre-2008) Dataset**

lgd ~ Seniority\_index + ind + Unsecured + mktret + Senior\_unsecure +

Percent\_Above + Revolver + tbill + Utility\_dummy + Third\_lien +

Inven\_AR\_cash + Guarantees + Term\_loan + Equipment + Intellectual +

Subordinated

**Final Model (Post-2008) Dataset**

lgd ~ Seniority\_index + ind + Senior\_unsecure + tbill + Intellectual +

Third\_lien + mktret + Equipment + Revolver

**Model Performance:** To gauge the performance of the model the average sums of squares error was calculated for in-sample, out-sample datasets.

Table 1: Model Performance: Linear Regression

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | Average SSE | |  |
| Dataset | **#observations** | **In Sample** | **Out of Sample** | **Adjusted R-Sq** |
| Complete Dataset | 4793 | 0.081 | 0.086 | 0.44 |
| Pre(Old) Dataset | 3751 | 0.077 | 0.086 | 0.46 |
| Post(New) Dataset | 1042 | 0.103 | 0.0825 | 0.44 |

**Inferences:**

* The model performed better for pre-2008 data as compared to post 2008 & full dataset.
* The model for post-2008 dataset has fewer significant predictor variables
* Macro-economic variables like market return, treasury bill rate are found significant in all the models

**SAS Enterprise Miner Results:**

The same process is followed and models are built for all three datasets. The results are in line with results observed from running R scripts.

The dataset is again divided into training and testing datasets in the ration 3:1. Model is built using the training dataset and tested on the testing dataset.

**SAS Enterprise Diagram:**

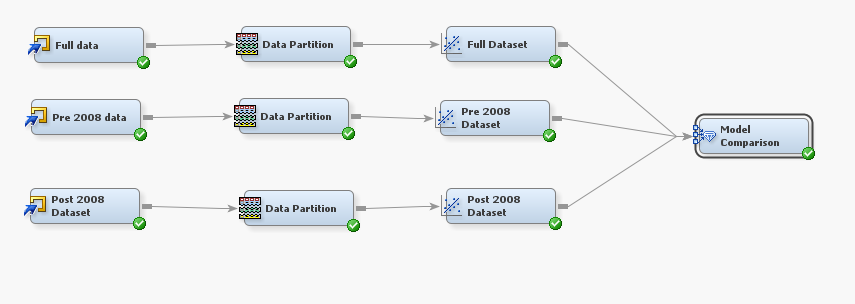
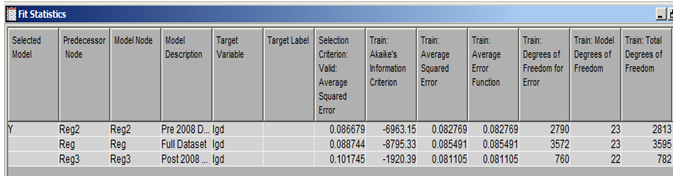


Figure 12 Modelling diagram for linear regression

**Model Comparison:**

Table 2 Model comparison for linear regression

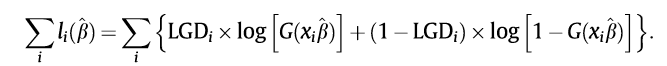


**Inferences:**

* Observed values for average SSE for in sample and out sample datasets are very similar to the values obtained by running R scripts
* The model performs the best for pre-2008 dataset

# Fractional Response Regression(FRR)

As the objective is to model the LGD between a range of 0 and 1, a Fractional Response Regression is used. The FRR is a quasi- likelihood function (squashing function) that is used to limit the values between 0 and 1.



The equation above is used to estimate the co-efficient of the FRR equation. The estimates thus obtained are consistent and asymptotically normal.

**Key Variables in the model:**

Apart from Junior, Senior secure and Sub-ordinated all the variables are significant in both the pre-2008, post-2008 and combined data set.

**Model results:**

Table 3: Model Results

|  |  |  |
| --- | --- | --- |
| FRR | Insample -Avg.SSE | R-Squared |
| Pre | 0.082 | 0.429 |
| Post | 0.086 | 0.414 |
| Full | 0.084 | 0.433 |

**Inference:**

The model variables chosen and estimated using the FRR is consistent with the regression and other models deployed to predict the LGD. The advantage of using FRR over other models is the ability of the model to strict predict values between 0 and 1.

# Regression Tree

The main objective is to build regression tree for three datasets- entire dataset containing observations before and after year 2008 (4793 observations), existing ‘pre’ dataset with 3751 observations and new ‘post’ dataset 1042 observations. Same sampling (75:25) is followed.

***Complete Dataset:***

Complete dataset, containing 4793 observations is split into training and testing sets. Using ‘rpart’ function, a regression tree is built using the training dataset. Variables ‘index’, ‘id’ and ‘year’ are excluded while building the tree. Using ‘plotcp’ function, ideal number of nodes is found to be 9.



Figure 13: Plotcp- Original Dataset Figure 14 Prunned Regression Tree- Complete Dataset

Resulting tree indicated that ‘Seniority\_index’ is the most important variable, followed by ‘Senior\_unsecure’, ‘Utility\_dummy’ and ‘tbill’. Using ‘predict’ function, both in-sample (training) and out-of- sample (testing) predictions are determined for the dependent variable, i.e. ‘lgd’. Prediction performance is determined by calculating average sums of squared error.

**‘Pre’ (Old) Dataset:**

Similar to entire dataset, regression tree is built for observations corresponding to ‘pre’ or existing old dataset containing 3751 observations. Using ‘plotcp’ function, ideal number of nodes is found to be 9.



Figure 15:Plotcp- ‘Pre’- Existing Dataset Figure 16 Pruned Regression Tree- ‘Pre’- Existing Dataset

**‘Post’ New Dataset:**

A regression tree is also built for observations corresponding ‘post’ new dataset containing 1042 observations. Using ‘plotcp’ function, ideal number of nodes is found to be 7.

Figure 17 Plotcp- ‘Post’- New Dataset Figure 18:Prunned Regression Tree- ‘Post’- New Dataset

To determine how well the regression tree predicts the dependent variable, ‘lgd’, average sums of squared error or Average SSE metric is used. All three regression trees are compared using the same metric. Following table provides average sums of squared error value obtained in both in-sample and out-of-sample predictions for all three regression tress.

Table 4 Comparison- Average SSE

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | Avg SSE | |
| Dataset | **#observations** | **In Sample** | **Out of Sample** |
| Complete Dataset | 4793 | 0.0848 | 0.0960 |
| Pre(Old) Dataset | 3751 | 0.0805 | 0.0917 |
| Post(New) Dataset | 1042 | 0.0725 | 0.0825 |

Important inferences are: All regression trees have a comparatively poor prediction performance in out-of-sample predictions than in-sample. Regression tree built on new/ ‘post’ dataset possessed better prediction performance than other two trees. Regression tree built on complete dataset possessed worse prediction performance than other two trees

**SAS Enterprise Miner:** All three regressions tree built in R, are also built using SAS Enterprise Miner. All three datasets- complete, ‘pre’ and ‘post’ datasets are imported individually into SAS Enterprise Miner using ‘File Import’ module under ‘Sample’. All three datasets are split into training and validation sets in 75:25 ratio- 75% of every dataset is set as its training and 25% of every dataset is set as its validation set. A fixed seed (14) is used for simple random sampling. This is implemented using ‘Data Partition’ module under ‘Sample’. A total of 3 such modules are used- one for each dataset. To build a regression tree in SAS EM, ‘Decision Tree’ module can be used which is available under ‘Model’. Three such decision trees are used one for each dataset. The final diagram would consist of three sets of ‘File Import’, ‘Data Partition’ and ‘Decision Tree’ connected- one set of each of the three datasets. All three regression trees can be compared using ‘Model Comparison’ module present under ‘Assess’ which provides a comprehensive list of values of various parameters including average sums of squared error, root mean squared error.

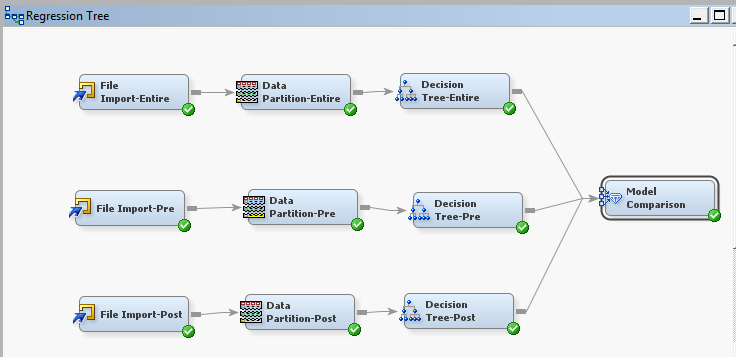
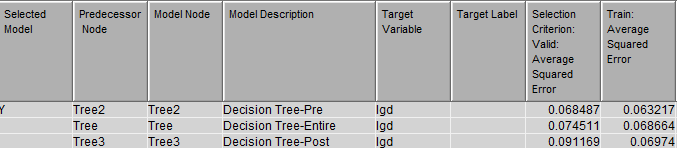


Figure 19 Regression Tree- SAS EM

Table 5 Average SSE- SAS EM



From the following table, we can observe that in-sample average SSE’s are almost similar among all the tree with low values. Regression tree for ‘post’ dataset performed the worst and that for ‘pre’ dataset performed better for both of which we obtained vice-versa inference when built using R codes.

# Neural Networks

Neural Networks are black box models for prediction. We 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.

* 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 Full Dataset into train and test sets in the ratio 75:25
2. Divide the training data again into 2 sets (training and testing) in the ratio 75:25.
3. Build the neural network for different size values and 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 and Build neural networks for each of the size determined for various decay values between 0 and 0.0001
5. Plot the size vs average squared error for each of the decay values
6. Select the combination of decay and size for which the error is least
7. Build neural networks with above selected decay and size values and different seed values. This is done in order to avoid local minima
8. Select the best seed value for which Average Sum squared errors is low

Plot of the average squared error vs the layer size for both the test and training data and Plot of average squared error in the test set vs the layer size various decay values is shown below

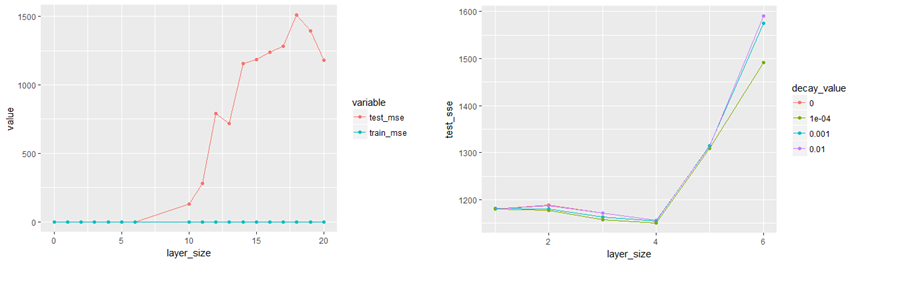


Figure 20 Neural Network tuning

From the first plot it looks like errors are low for size below 6 for the test dataset. From the second plot we can infer that a layer size of 4 and decay value of 0.0001 is giving us the least avg. sum squared error.

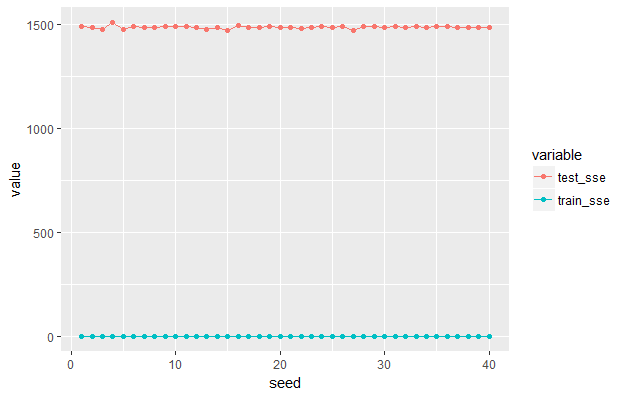


Figure 21 Tuning for the best seed value

Plot of Avg SSE for different seed values is shown above. Although there isn’t a big change in the Avg. SSE values a seed of 15 resulted in the least Avg. SSE

**Final Model**

set.seed(15)

net<-nnet(lgd~.,size = 4, data=lgd\_train, rang = 0.00001,

linout = TRUE, maxit = 10000, decay = 0.0001, skip = TRUE);

Similar procedure was followed for Pre and Post datasets separately to build Neural networks on them.

**SAS Enterprise Miner**

The Auto Neural Module in SAS enterprise miner was used to build the neural network model. Three models were built one each for complete data set, pre 2008 and post 2008 data

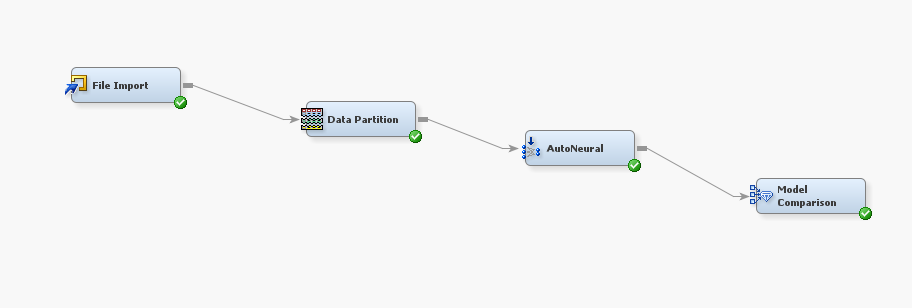


Figure 22 SAS EM

**Model Results:**

Table 6 Model Results

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | Full Dataset | Pre 2008 Data | Post 2008 Data |
| R | Train Averaged Sum Squared Error | 0.0614998 | 0.05288881 | 0.04553702 |
| Test Averaged Sum Squared Error | 0.0785984 | 0.07505478 | 0.08465286 |
| SAS EM | Train Averaged Sum Squared Error | 0.0733111 | 0.069111556 | 0.059517288 |
| Test Averaged Sum Squared Error | 0.0760485 | 0.076549236 | 0.103306254 |

**Inferences:**

Neural Network model built and tuned in R performed better than SAS EM.

The test Avg. SSE of the neural network models was around 0.07 which is similar to linear regression models. So, Neural network model didn’t offer any advantages over LR model.

# Random Forest

Random forests work by constructing multitude of decision trees and outputting the mean value of the individual trees is outputted. For building each tree a subset of available features is used. We built the Random Forest using the “RandomForest” Package in r. Below is the Plot of avg. SSE vs number of trees used in the Random forest. We see that the error Stabilizes around 200.

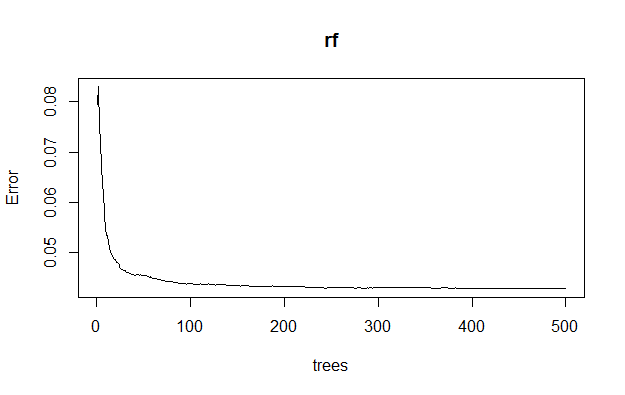


Figure 23 Random Forest Number of trees vs error

One advantage of using Random Forest is that we get the variable importance plot which plots the importance of variables based on their contribution to reduction in Avg. SSE of LGD.

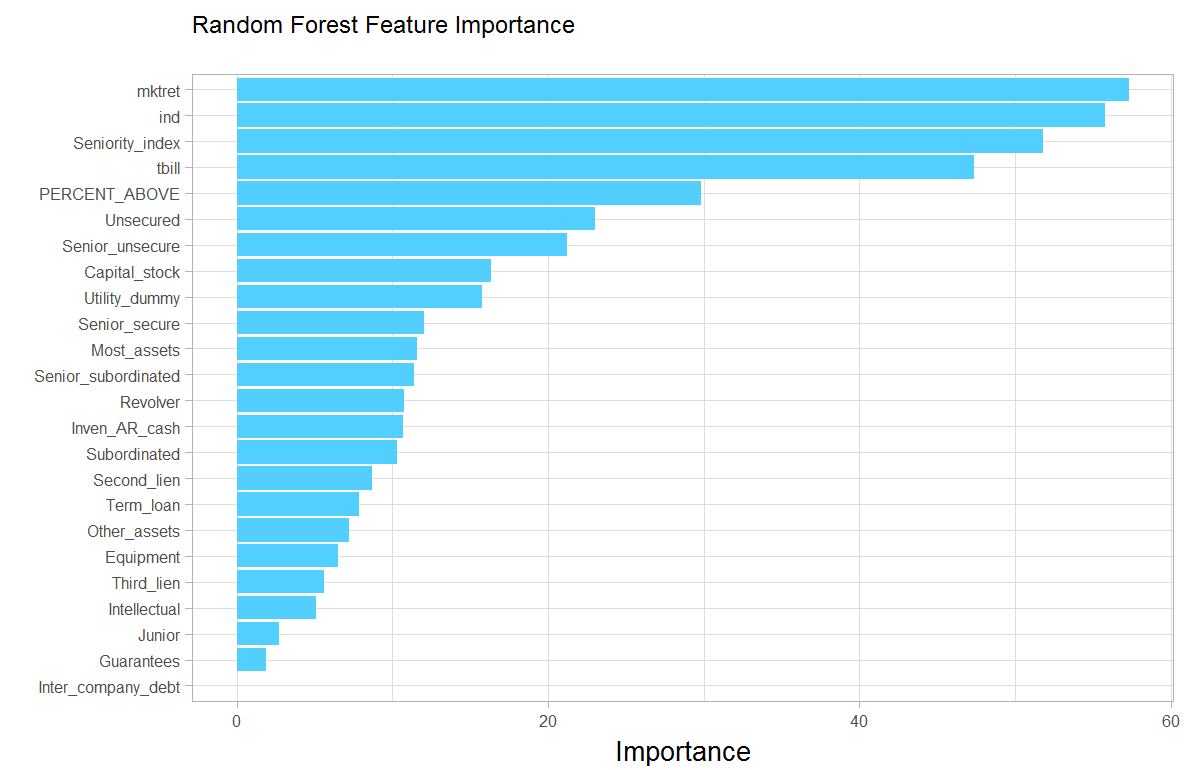


Figure 24: Variable importance

Based on the plot we can infer that most important variables are mktret, ind, Seniority\_index and tbill. The interesting fact that we observe is that the tbill rate which we used were the yearly averages for each year. If we had granular level data about the date of defaults in the observations we could have used the monthly tbill rates for better predictions.

We built models on all the three datasets using R and SAS EM. For SAS Enterprise miner we used the High performance Forest module.

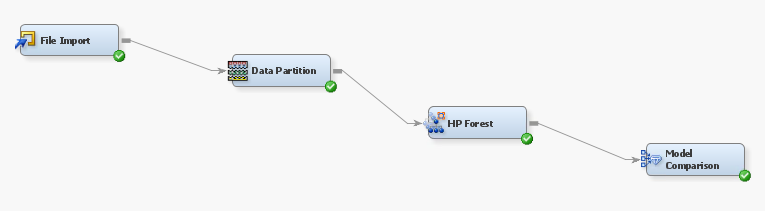


Figure SAS EM

**Model Results:**

Table 7 Model Results

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | Full Dataset | Pre 2008 Data | Post 2008 Data |
| R | Train Averaged Sum Squared Error | 0.01370 | 0.012212 | 0.01727103 |
| Test Averaged Sum Squared Error | 0.04545 | 0.04479158 | 0.06212173 |
| SAS EM | Train Averaged Sum Squared Error | 0.076145 | 0.073863 | 0.076407 |
| Test Averaged Sum Squared Error | 0.079798 | 0.077733 | 0.098357 |

**Inference:**

* The random forest built using R where much better than models built on SAS EM.
* The Avg. SSE among obtained in the full dataset is 0.045 which is the best among all the models tried.

We also plotted a migration matrix between the actual and predicted values.

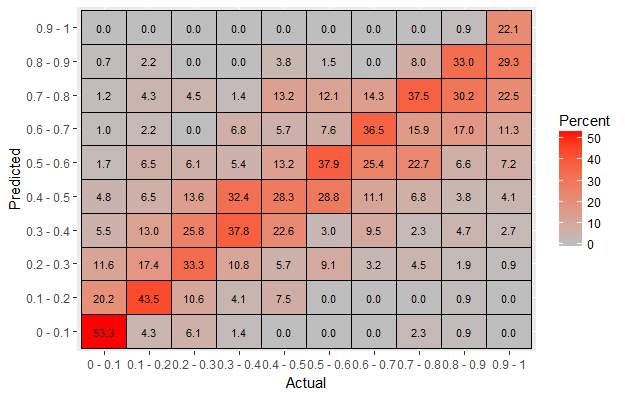


Figure 26: Migration Matrix

For the plot, we divided the actual and predicted lgd values into 10 bins and found out the percentage of correct predictions within each bin. The Matrix shows that we get the highest percentage of 53.3% correct in the less than 0.1 bin i.e. for 53.3% of observations with lgd values less than 0.1 had a predicted value of less than 0.1. The prediction poor in the “0.9-1” or greater than 0.9 bin where we get only 22.1% of the predictions in the same bin.

# Conclusion:

On comparing the various models, Random Forest had the least Avg. SSE value in the all the three datasets. Hence, we conclude that it is the best technique to make LGD predictions.