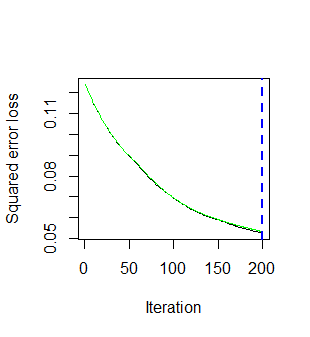
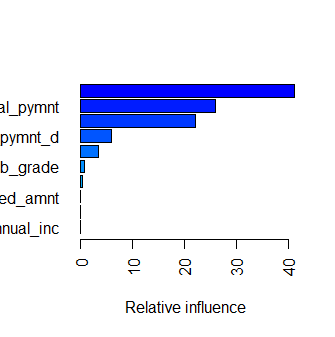
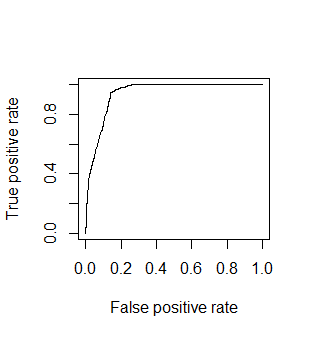
|  |  |
| --- | --- |
| Image result for data mining  Data Mining - 572  Assignment 2 | Ashok Bhatraju – 670248723  shourya narayan – 651193827  vivek kumar - 670460685 |

1. **(a1) Develop gradient boosted models to predict loan\_status. Experiment with different parameter values and identify which gives ‘best’ performance. How do you determine ‘best’ performance?**

**(a2) For the gbm model, what is the loss function, and gradient in the method you use?**

**GBM (uisng train data):**





**aucPerf\_gbm\_model@y.values**

**[[1]]**

**[1] 0.9401151**

**sqrt(min(gbm\_model$cv.error))**

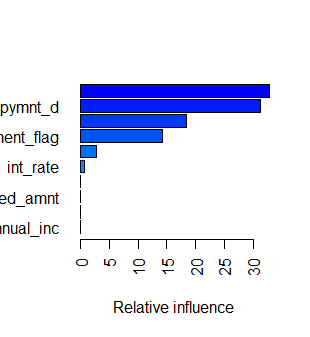
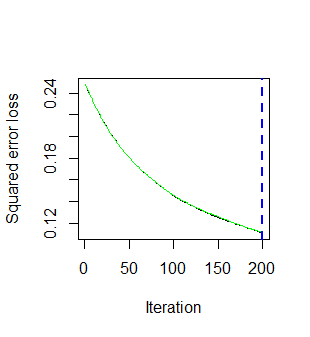
**[1] 0.2303919**

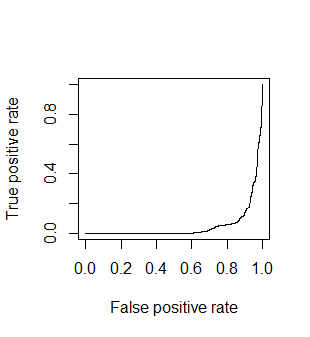
AUC of 0.94 clearly suggests that GBM is the best model among the models used in the following questions.

0.23 sqrt suggests that loan status is 0.23 off from the actual loan status

Iteration vs squared error loss graph suggests that our model needs more number of trees.

**GBM (using over sampling):**





**aucPerf\_gbm\_model1@y.values**

**[[1]]**

**[1] 0.2571706**

**sqrt(min(gbm\_model1$cv.error))**

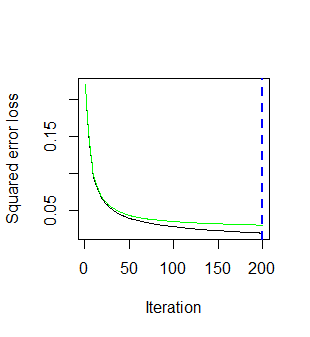
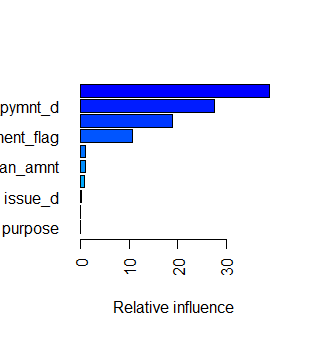
**[1] 0.4564674**

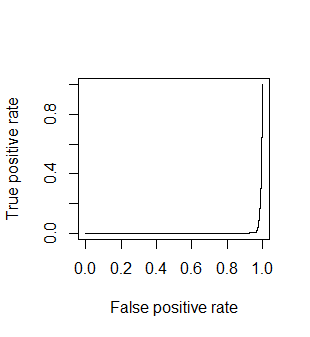
Over sampling gave us the least AUC for GBM which is 0.25.

0.45 sqrt suggests that loan status is 0.45 off from the actual loan status

Iteration vs squared error loss graph suggests that our model needs more number of trees.

**GBM (using Under sampling):**





**aucPerf\_gbm\_model2@y.values**

**[[1]]**

**[1] 0.2303179**

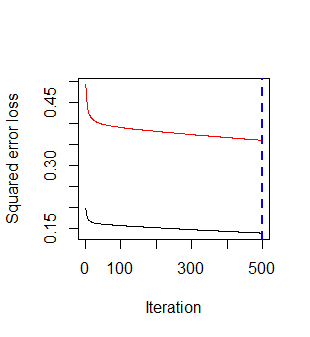
Under sampling gave us the least AUC for GBM which is 0.23.

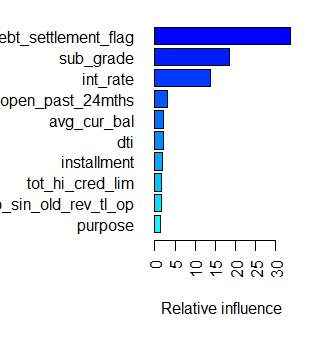
Iteration vs squared error loss graph suggests that our model needs more number of trees.

**GBM Grid results:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **treeDepth** | **minNodesize** | **bagFraction** | **shrinkage** | **bestTree** | **minRMSE** |
| 1 | 2 | 10 | 0.5 | 0.001 | 500 | 0.675905 |
| 2 | 5 | 10 | 0.5 | 0.001 | 500 | 0.671361 |
| 3 | 2 | 30 | 0.5 | 0.001 | 500 | 0.675973 |
| 4 | 5 | 30 | 0.5 | 0.001 | 500 | 0.671383 |
| 5 | 2 | 10 | 0.8 | 0.001 | 500 | 0.676011 |
| 6 | 5 | 10 | 0.8 | 0.001 | 500 | 0.671474 |
| 7 | 2 | 30 | 0.8 | 0.001 | 500 | 0.675995 |
| 8 | 5 | 30 | 0.8 | 0.001 | 500 | 0.671447 |
| 9 | 2 | 10 | 1 | 0.001 | 500 | 0.676062 |
| 10 | 5 | 10 | 1 | 0.001 | 500 | 0.671509 |
| 11 | 2 | 30 | 1 | 0.001 | 500 | 0.676062 |
| 12 | 5 | 30 | 1 | 0.001 | 500 | 0.671509 |
| 13 | 2 | 10 | 0.5 | 0.01 | 500 | 0.636929 |
| 14 | 5 | 10 | 0.5 | 0.01 | 499 | 0.630466 |
| 15 | 2 | 30 | 0.5 | 0.01 | 500 | 0.636984 |
| 16 | 5 | 30 | 0.5 | 0.01 | 500 | 0.63059 |
| 17 | 2 | 10 | 0.8 | 0.01 | 500 | 0.637537 |
| 18 | 5 | 10 | 0.8 | 0.01 | 500 | 0.631327 |
| 19 | 2 | 30 | 0.8 | 0.01 | 500 | 0.637543 |
| 20 | 5 | 30 | 0.8 | 0.01 | 500 | 0.631325 |
| 21 | 2 | 10 | 1 | 0.01 | 500 | 0.63817 |
| 22 | 5 | 10 | 1 | 0.01 | 500 | 0.632161 |
| 23 | 2 | 30 | 1 | 0.01 | 500 | 0.63817 |
| 24 | 5 | 30 | 1 | 0.01 | 500 | 0.632134 |
| 25 | 2 | 10 | 0.5 | 0.1 | 493 | 0.619706 |
| 26 | 5 | 10 | 0.5 | 0.1 | 500 | 0.599443 |
| 27 | 2 | 30 | 0.5 | 0.1 | 478 | 0.620861 |
| 28 | 5 | 30 | 0.5 | 0.1 | 500 | 0.600144 |
| 29 | 2 | 10 | 0.8 | 0.1 | 500 | 0.619903 |
| 30 | 5 | 10 | 0.8 | 0.1 | 500 | 0.599258 |
| 31 | 2 | 30 | 0.8 | 0.1 | 500 | 0.620598 |
| 32 | 5 | 30 | 0.8 | 0.1 | 500 | 0.599871 |
| 33 | 2 | 10 | 1 | 0.1 | 500 | 0.621681 |
| 34 | 5 | 10 | 1 | 0.1 | 500 | 0.600718 |
| 35 | 2 | 30 | 1 | 0.1 | 498 | 0.621288 |
| 36 | 5 | 30 | 1 | 0.1 | 500 | 0.600295 |

**0.68 is the highest error and 0.59 is the least error which suggests that loan\_status is off from actual data about 59% to 68%**





The loss function and gradient used for this model are:

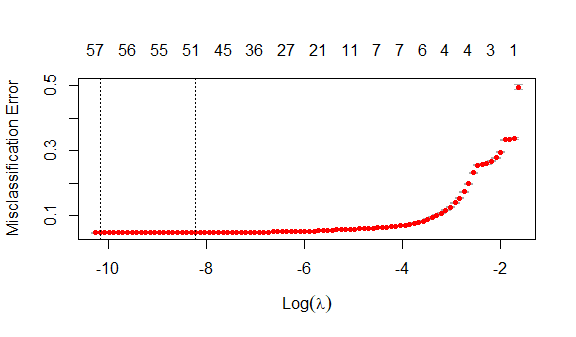
**Loss Function: L( y, F(x) ) = (y – F(x) )2/2**

**Gradient: It is the derivative of loss function, which upon simplification comes out to F(xi) - yi**

**(b1) Develop linear (glm) models to predict loan\_status. Experiment with different parameter values and identify which gives ‘best’ performance. How do you determine ‘best’ performance ? How do you handle variable selection? Experiment with Ridge and Lasso, and show how you vary these parameters, and what performance is observed.**

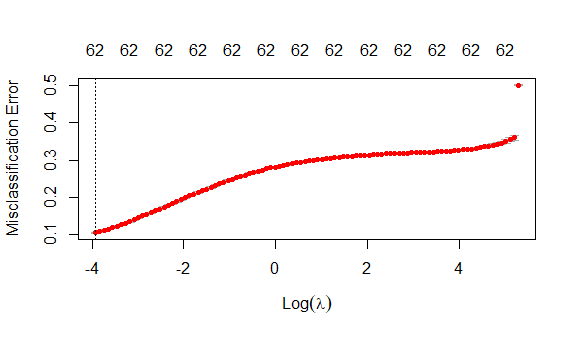
**(b2) For the linear model, what is the loss function, and link function you use ?**

**Glm with alpha=1: (Lasso)**



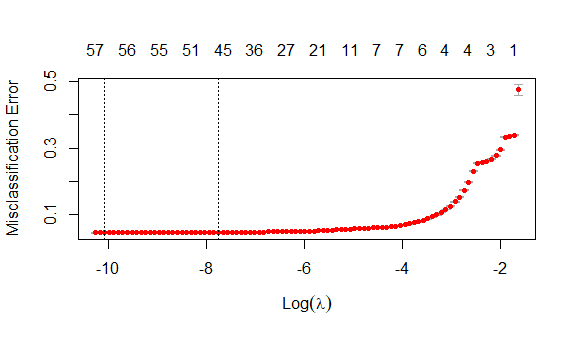
**AUC = 0.29**

**Glm with alpha=0: (Ridge regression)**



**AUC: 0.78**

**Glm without alpha:**



**AUC: 0.33**

**For the GLM Model, the loss function used is:**

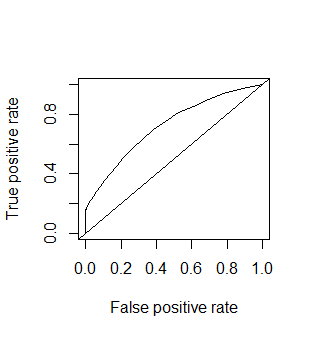
J(w) = ∑(xiwi -yi)2

And link function is:

ln(p/(1-p)) ; p is the probability of the linear function

**(c) Compare performance of models with that of random forests**

**Random:**



**curve@y.values**

**[[1]]**

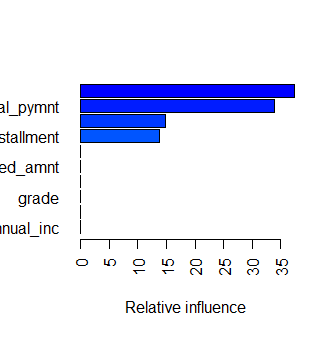
**[1] 0.7211528**

**AUC:0.72**

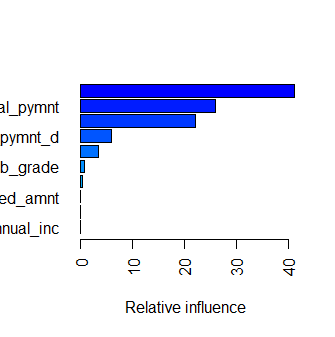
|  |
| --- |
| **Reference** |
| Prediction Charged Off Fully Paid |
| Charged Off 680 28 |
| Fully Paid 3479 23600 |
|  |
| Accuracy : 0.8738 |
| 95% CI : (0.8698, 0.8777) |
| No Information Rate : 0.8503 |
| P-Value [Acc > NIR] : < 2.2e-16 |
|  |
| Kappa : 0.2466 |
|  |
| Mcnemar's Test P-Value : < 2.2e-16 |
|  |
| Sensitivity : 0.16350 |
| Specificity : 0.99881 |
| Pos Pred Value : 0.96045 |
| Neg Pred Value : 0.87152 |
| Prevalence : 0.14967 |
| Detection Rate : 0.02447 |
| Detection Prevalence : 0.02548 |
| Balanced Accuracy : 0.58116 |
|  |
| 'Positive' Class : Charged Off |

Our Random forest model gave us the accuracy of 0.75 for normal model and when we adjusted the threshold to 0.2 gave us accuracy of 0.87S

**(d) Examine which variables are found to be important by the best models from the different methods, and comment on similarities, difference. What do you conclude?**



**GBM** model results suggests that total\_pymnt, instalments and debt settlement flag are among the most important variables



**GLM** model results suggests that sub grade, interest rate are among the important variables of this model

**(e) In developing models above, do you find larger training samples to give better models ? Do you find balancing the training data examples across classes to give better models?**

**Bigger samples performance:**

aucPerf\_gbm\_modela@y.values

[[1]]

[1] 0.7461099

aucPerf\_gbm\_model1a@y.values

[[1]]

[1] 0.7426346

We did run the models with different train and test data size’s (0.6,0.7 and 0.8) and the results doesn’t vary much. This suggests that larger training samples doesn’t increase the model performance.

Balancing training data gave us the results which are consistent in nature.

***2. Develop models to identify loans which provide the best returns. Explain how you define returns? Does it include Lending Club’s service costs?***

***Develop glm, rf, gbm models for this. Show how you systematically experiment with different parameters to find the best models. Compare model performance. Do you find larger training sets to give better models?***

We have used actual\_return across different models to identify loans which will provide the best returns. We have developed Random Forest, GLM and GBM models for this data (after over sampling) on actual return and got the least root mean square error lowest for Random Forest (0.8%), a bit higher for GBM model (1.8%) and highest for GLM model (6.8% for **Vanilla**, 6.8% for **Lasso** and 7.14% for **Ridge** model). With lowest RMSE, **Random Forest** is the best model to identify loans which provide the best returns

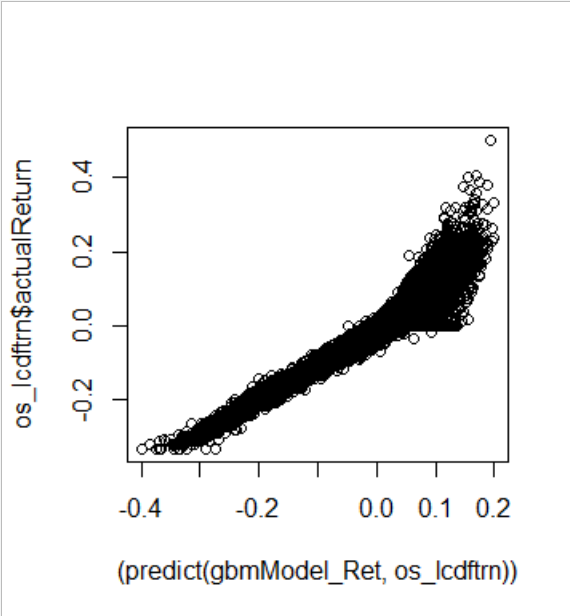
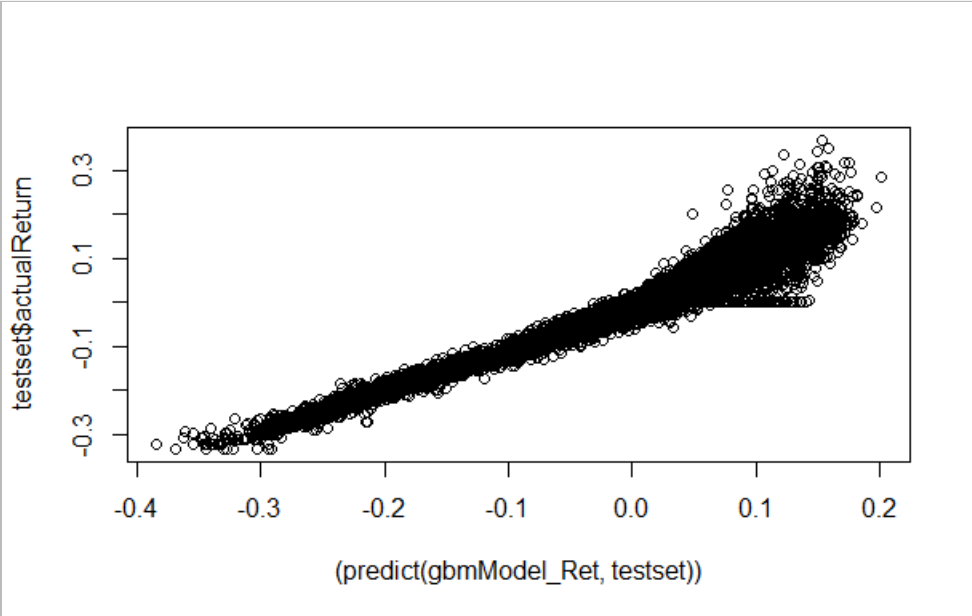
The ratio of difference between Total Payment and Funded Amount over Funded Amount gives us the measure of actual return over the total term. This value when divided by the actual term gives the **actual return**.

The model doesn’t include Lending Club service cost, as the required parameter ‘serviceFeeRate’ is not included in the dataset.

To analyse the variation of accuracy or (lower RMSE) on the size of data, we varied the size of data for our Random Forest model and observed that as we increase the size of training data set from 60% to 70% and 80%, **we were getting better models (lower RMSE) for larger size training data**.

Below are the plots obtained by running RF, GLM and GBM models on actual\_return for the given data after oversampling.

GBM Model on Actual Return for Train and Test Data:

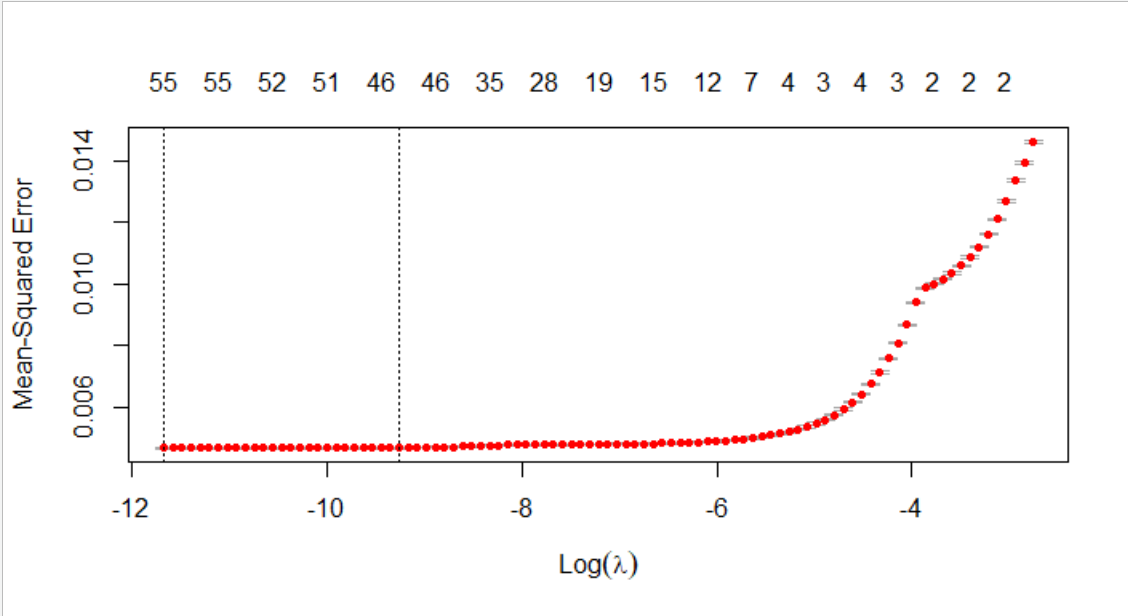
RMSE for Train Data: 1.86%

RMSE for Test Data: 1.96%

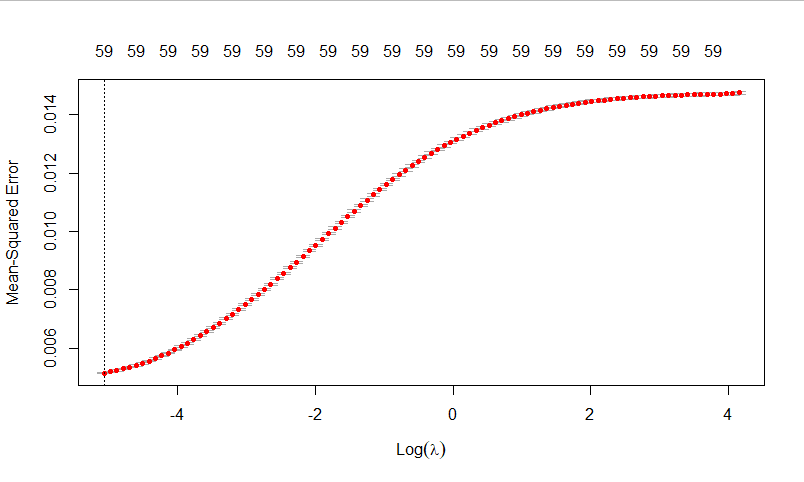
**GBM Grid results:**

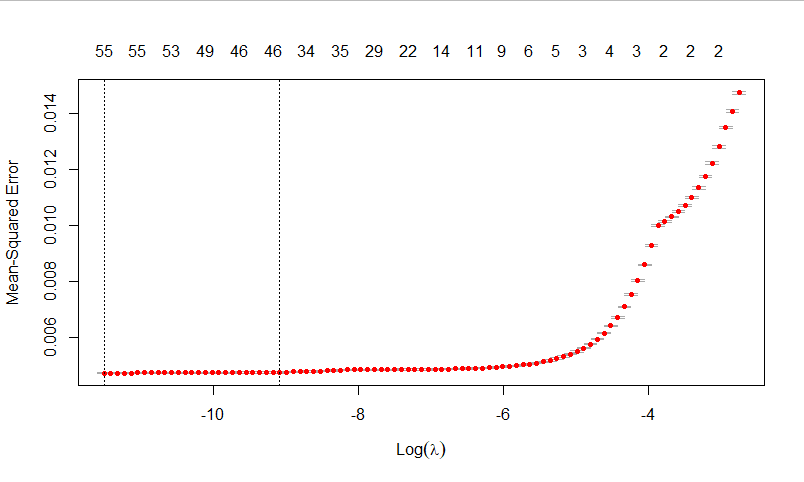
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **treeDepth** | **minNodeSize** | **bagFraction** | **shrinkage** | **bestTree** | **minRMSE** |
| 1 | 2 | 10 | 0.5 | 0.001 | 500 | 0.074374 |
| 2 | 5 | 10 | 0.5 | 0.001 | 500 | 0.069843 |
| 3 | 2 | 30 | 0.5 | 0.001 | 500 | 0.074328 |
| 4 | 5 | 30 | 0.5 | 0.001 | 500 | 0.069798 |
| 5 | 2 | 10 | 0.8 | 0.001 | 500 | 0.074373 |
| 6 | 5 | 10 | 0.8 | 0.001 | 500 | 0.070106 |
| 7 | 2 | 30 | 0.8 | 0.001 | 500 | 0.074344 |
| 8 | 5 | 30 | 0.8 | 0.001 | 500 | 0.070069 |
| 9 | 2 | 10 | 1 | 0.001 | 500 | 0.074365 |
| 10 | 5 | 10 | 1 | 0.001 | 500 | 0.069898 |
| 11 | 2 | 30 | 1 | 0.001 | 500 | 0.074365 |
| 12 | 5 | 30 | 1 | 0.001 | 500 | 0.069898 |
| 13 | 2 | 10 | 0.5 | 0.01 | 500 | 0.047476 |
| 14 | 5 | 10 | 0.5 | 0.01 | 500 | 0.03103 |
| 15 | 2 | 30 | 0.5 | 0.01 | 500 | 0.047591 |
| 16 | 5 | 30 | 0.5 | 0.01 | 500 | 0.030929 |
| 17 | 2 | 10 | 0.8 | 0.01 | 500 | 0.047235 |
| 18 | 5 | 10 | 0.8 | 0.01 | 500 | 0.03147 |
| 19 | 2 | 30 | 0.8 | 0.01 | 500 | 0.047404 |
| 20 | 5 | 30 | 0.8 | 0.01 | 500 | 0.031543 |
| 21 | 2 | 10 | 1 | 0.01 | 500 | 0.046599 |
| 22 | 5 | 10 | 1 | 0.01 | 500 | 0.031871 |
| 23 | 2 | 30 | 1 | 0.01 | 500 | 0.046599 |
| 24 | 5 | 30 | 1 | 0.01 | 500 | 0.031871 |
| 25 | 2 | 10 | 0.5 | 0.1 | 500 | 0.02069 |
| 26 | 5 | 10 | 0.5 | 0.1 | 499 | 0.013809 |
| 27 | 2 | 30 | 0.5 | 0.1 | 500 | 0.021 |
| 28 | 5 | 30 | 0.5 | 0.1 | 500 | 0.014918 |
| 29 | 2 | 10 | 0.8 | 0.1 | 500 | 0.020848 |
| 30 | 5 | 10 | 0.8 | 0.1 | 493 | 0.013416 |
| 31 | 2 | 30 | 0.8 | 0.1 | 500 | 0.021175 |
| 32 | 5 | 30 | 0.8 | 0.1 | 500 | 0.014488 |
| 33 | 2 | 10 | 1 | 0.1 | 500 | 0.021224 |
| 34 | 5 | 10 | 1 | 0.1 | 500 | 0.013644 |
| 35 | 2 | 30 | 1 | 0.1 | 500 | 0.021577 |
| 36 | 5 | 30 | 1 | 0.1 | 500 | 0.014315 |

**0.07 is the highest error and 0.01 is the least error which suggests that loan\_status is off from actual data about 1% to 7%**

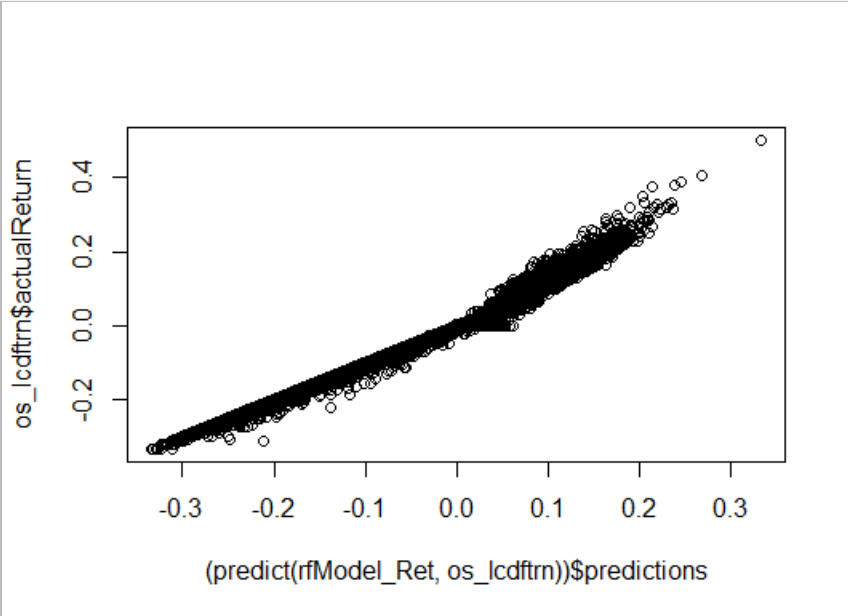
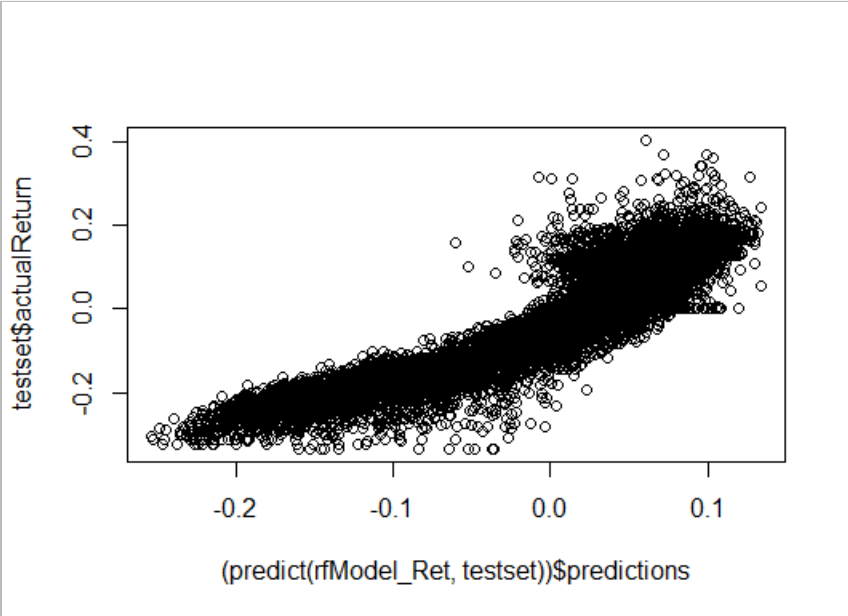
**GLM** for Actual Return (**Vanilla**)  
\*Plot for test data is almost same due to similar RMSE values for both the sets

RMSE for Training Data: 6.81%  
RMSE for Test Data: 6.91%

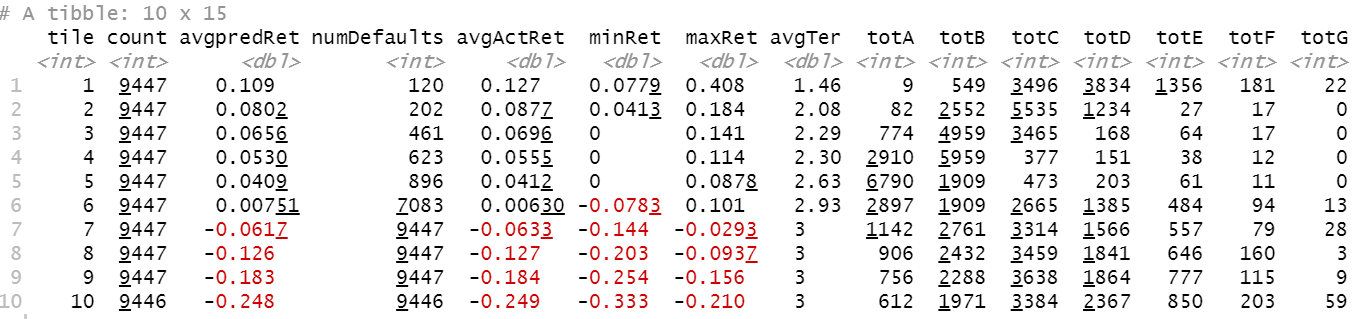
**GLM** for Actual Return (**Lasso**)   
  
RMSE for Training Data: 6.81%  
RMSE for Test Data: 6.91%

**GLM** for Actual Return (**Ridge**)   
  
RMSE for Training Data: 7.14%  
RMSE for Test Data: 7.54%

|  |  |
| --- | --- |
|  |  |

**Random Forest** for Train and Test Data respectively: 

RMSE for Train: 0.8%  
RMSE for Test: 4.1%



The above decile table for random forest model on actual return can yield some effective investment approach, like **average actual return is highest for decile 1 and grade D** has maximum number of loans for decile 1, so this lot becomes the best pick for higher returns. In addition to that, loans in deciles 7 and lesser have negative average predicted rate as well as average actual return rate, so it will not be wise to invest in loans from these deciles.

**Best model based on RMSE:**

|  |  |  |
| --- | --- | --- |
|  | **RMSE for Training** | **RMSE for Testing** |
| **GBM** | 1.86 | 1.96 |
| **GLM(Lasso)** | 6.81 | 6.91 |
| **GLM(Ridge)** | 7.14 | 7.54 |
| **Random Forest** | 0.8 | 4.1 |

RMSE is an absolute measure of fit. As a square root of variance, RMSE can be interpreted as standard deviation of the unexplained variance and has the useful property of being in the same units as the response variable. RMSE lower values indicate better fit. RMSE is a good measure of how accurately the model predicts the response, and it is the important measure for fit if the main purpose of the model is prediction.

Based on the above values, Random forest is the best if we consider the lowest RMSE values. If we consider the better fit based on RMSE values for both training and testing **GBM** is the better one as the difference of values is just 0.1

**3. Considering results from Questions 1 and 2 above, how would you select loans for investment? Describe your approach and show performance.**

We used GBM,GLM and Random forest in order to determine the best investment strategy. We used loan status and actual return to reach at the best investment strategy.

**Above these models, we can see that random forest has the least error of 0.8%. we are going to use the data from random forest to determine the best investment strategy for us.**

From the table above we only took Decile 1 to 6 as they give us positive returns, whereas the rest gives us negative returns.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Average actual return** | **Defaults** | **A** | **B** | **C** | **D** |
| **1** | **0.13** | **120** | **9** | **549** | **3496** | **3834** |
| **2** | **0.09** | **202** | **82** | **2552** | **5535** | **1234** |
| **3** | **0.07** | **461** | **774** | **4959** | **3465** | **168** |
| **4** | **0.06** | **623** | **2910** | **5959** | **377** | **151** |
| **5** | **0.04** | **896** | **6790** | **1909** | **473** | **203** |
| **6** | **0.01** | **7083** | **2897** | **1909** | **2665** | **1385** |
|  | **Average return\*Grade wise number** |  | **508.93** | **1082** | **1236** | **638** |

Decile 6 has very high number of defaults, its wise to eliminate that too.

Decile 1 and Grade D looks like better combination to invest, but we can also see that higher numbers of grade D falls into Decile 7 and lower, so its wise not to invest unless you are high risk averse.

From weighted average calculation between Average actual return and grade wise number, we can see that Grade C gives the best average return followed by B and A.

**Conclusion:**

**Grade C-B-A :** for those who wants best average return

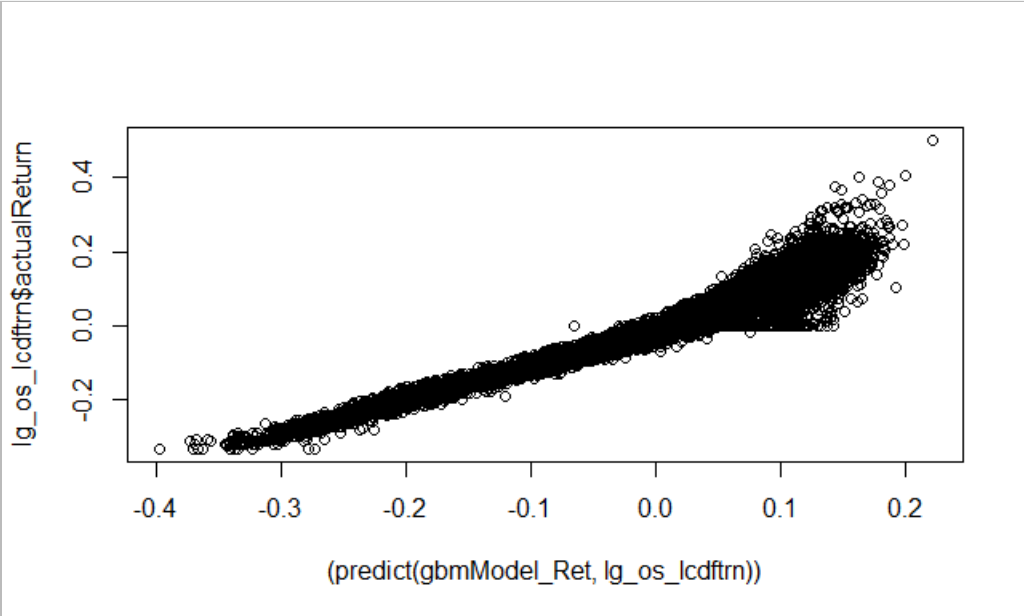
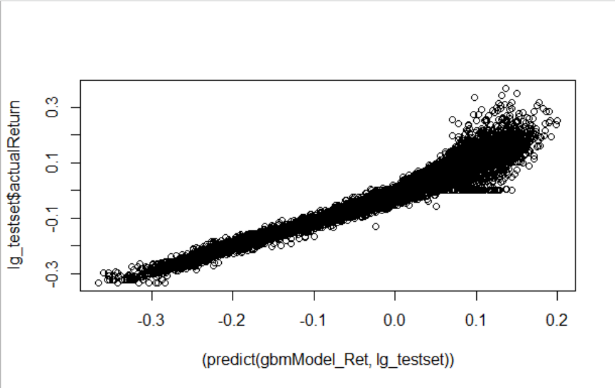
**Grade D**: Decile 1 and this grade gives the best return but poses the risk of high default chance

**4. As seen in data summaries and your work in the first assignment, higher grade loans are less likely to default, but also carry lower interest rates; many lower grad loans are fully paid, and these can yield higher returns. One approach may be to focus on lower grade loans (C and below) and try to identify those which are likely to be paid off. Develop models from the data on lower grade loans, and check if this can provide an effective investment approach. Compare performance of models from different methods (glm, gbm, rf).**

**Can this provide a useful approach for investment? Compare performance with that in Question 3.**

We have created another data set excluding loan grades A and B from the original data set. Identification of loans which are likely to be paid off, can be given by measure of actual return. Hence we have used Random Forest and GBM Model on actual return for loan grades C to G and basis our observation, Random Forest model appears to be the best model for the new data set.

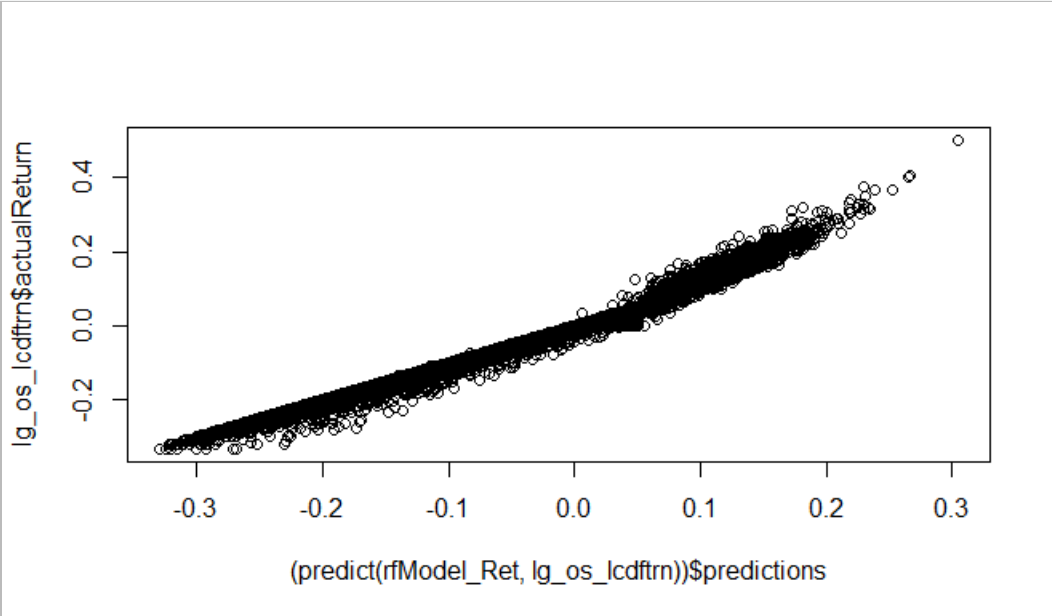
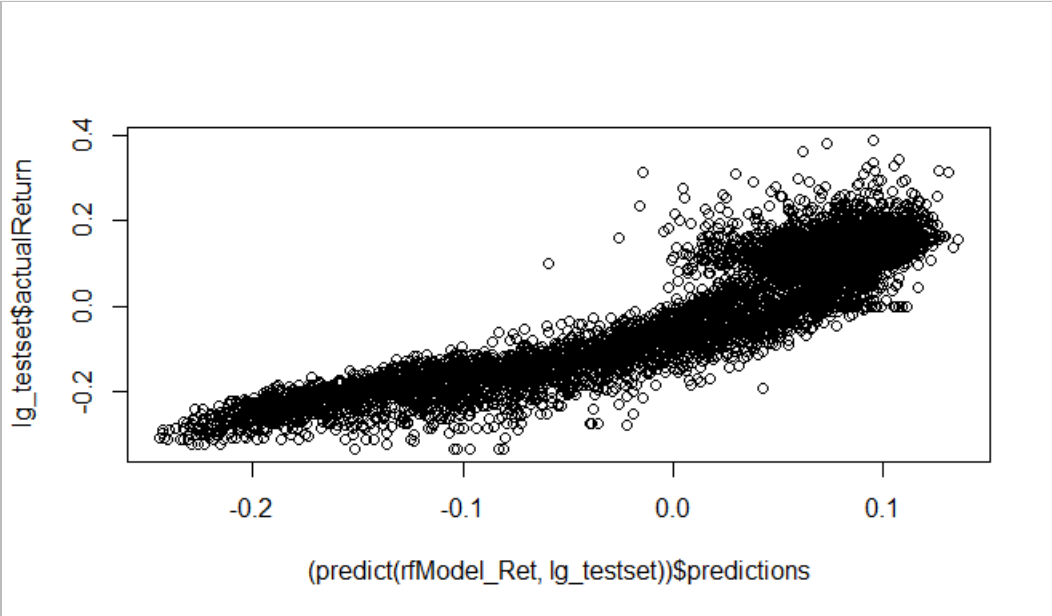
* GBM Plot of Actual\_Return on Train and Test Data for loan grades C to G



RMSE for Train Data: 1.89%

RMSE for Test Data: 1.98%

* Random Forest Plot of Actual\_Return on Train and Test Data for grades C to G

RMSE for Train Data: 0.89%

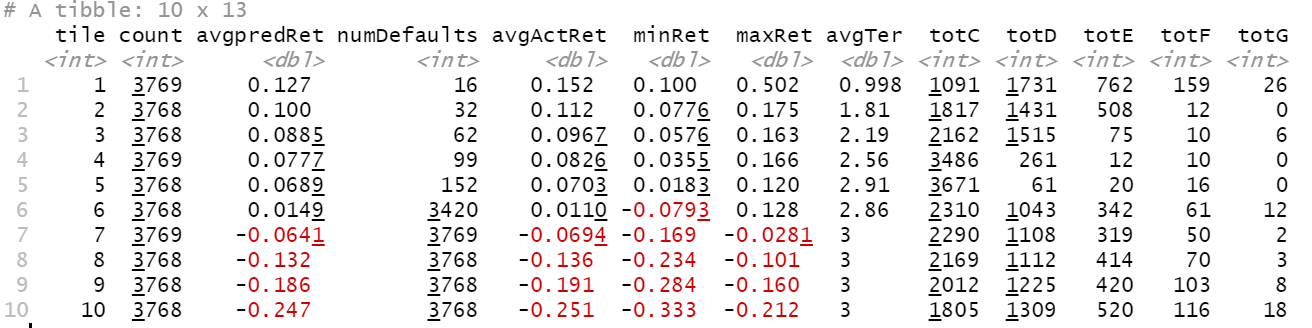
RMSE for Test Data: 4.02%

The decile plot for lower grade loans can help us formulate strategies for an effective investment

approach. For example when we look at the decile plot below, then we can draw the following

inferences which can be used for an effective investment approach:

1. The average actual return is highest for decile 1 and loan grade D has maximum count for  
   that decile, so it would be a wise to invest for these loans and get better returns
2. The average actual return is second highest for decile 2 and loan grade C has maximum  
   count for that decile, so this chunk becomes the 2nd most preferred category amongst  
   lower grade loans
3. Average predicted return and average actual return is negative beyond 6th decile, so it will not be wise to invest in 7th to 10th decile loans



1. General Observation:
   1. The average predicted return and average actual return decreases as we move from 1st to 10th decile
   2. The number of defaults per decile increases as we move from 1st to 10th decile

**Best model based on RMSE:**

|  |  |  |
| --- | --- | --- |
|  | **RMSE for Training** | **RMSE for Testing** |
| **GBM** | 1.89 | 1.98 |
| **Random Forest** | 0.89 | 4.02 |

Based on the above values, Random forest is the best if we consider the lowest RMSE values. If we consider the better fit based on RMSE values for both training and testing **GBM** is the better one as the difference of values is just 0.09