Predicting Interest Rates For Loans

(IS 621 Final Project Report)

Conor Buckley, Nathan Groom, Xingjia Wu

Abstract:

Our project focuses on the Lending Club, a peer to peer lending startup in San Francisco. We hope to determine the attributes most strongly linked to interest rates paid on personal loans, which may subsequently be used to advise any individual or family themselves looking to borrow money.

Our dataset consists of 39,787 rows, each representing a loan given to a borrower. It includes 51 columns, 50 of which represent predictor variables and the remaining variable, 'Interest Rate', representing our target variable.

After data cleaning and variable pre-selection, several regression models were applied on this dataset, including multiple linear regression with stepwise selection, regularized linear regression, and tree-based models such as random forest and gradient boosting machine (GBM), using packages built in the R statistical programming language. Selection of models was based on mean squared error (MSE) of prediction. Our lowest MSE was found with GBM models. The most relative influence of variables for these models were 'revol_util' (the revolving line utilization rate, the amount of credit the borrower is using relative to all available revolving credit) and 'term' (the number of payments on the loan).

Key Words:

Interest rates, loan, borrower, credit, term

Introduction:

Lending Club is a San Francisco based startup, which is described on their website as "the world's largest online marketplace connecting borrowers and investors." Lending Club offers personal loans of up to $40,000 and claims to distinguish itself by connecting borrowers with investors, thereby bypassing the need for a bank or a more traditional loan agency. The purpose of this report is to investigate the peer to peer lending industry. More specifically, we intend to determine which predictive features of a potential borrower have the strongest effects on the interest rate that borrower ultimately pays on his or her loan.

We recognize that taking out a loan, even with a maximum dollar amount of $40,000, can be a major decision in someone's life and comes with an inherent amount of risk. This project aims to alleviate as much of that risk and anxiety as possible through the use of predictive analytics, helping the borrower in question better understand exactly what taking out a loan with the Lending Club entails for his or her specific financial situation.

Literature review:

One of the key roles played by Lending Club in their peer-to-peer marketplace is to screen potential borrowers. This allows lenders to engage in transactions with more confidence about the risk they would be taking on. In their paper, 'Mitigating adverse selection in P2P lending: Empirical evidence from Prosper.com' (1), the authors find that this screening function is very important in mitigating adverse selection in peer-to-peer lending.

For one part of their analysis they employed an OLS regression on interest rates in order to establish borrower characteristics that had a significant influence on the interest rate. They found that the key significant characteristics were credit rating and a dummy variable indicating if the borrower was retired. The authors also detected both multicollinearity and heteroscedasticity in the data, so two additional regressions were created that excluded certain variables and used the White estimator.

Along with the screening service provided by Lending Club, it is also beneficial for the lenders to have an understanding about the factors that contribute to a borrower defaulting in a peer-to-peer marketplace. In their paper, 'Determinants of Default in P2P Lending' (2), the authors attempt to explain the factors that contribute to a borrower defaulting in peer-to-peer lending. They used data containing loan information from the Lending Club market place. In order to predict defaults from the data, the authors created a logistic regression. The findings from this regression were that factors such as loan purpose, annual income, current housing situation, credit history and indebtedness contributed to the explanation of default.

Methodology:

Data cleaning

The problem we are attempting to address largely relates to the ability of lower end borrowers (I.e. those with lower credit scores) to secure an affordable loan. As the maximum loan amount provided by the Lending Club is $40,000, the loans we are analyzing are more likely to pertain to expenses such as student debt or medical costs rather than much bigger loans to purchase a home or start a business.

The dataset upon which we are performing our analysis comes directly from the Lending Club's website. We found this dataset relatively easy to work with aside from a few challenges. With respect to the dataset's 50 predictor variables, we immediately eliminated 27, leaving us with 23 to use in our regression analysis. Most of the eliminated variables related to the payment plan after the loan had been approved. For example, "total payment received", "total principal received", "total late fee", "most recent payment", and "next payment" all pertain to an active loan were therefore not of use in predicting the interest rate, which would have already been assigned at this point in the lending process. Other variables were eliminated due to either being impractical to analyze or for simply offering no insight. The "job title" variable was eliminated as it would have been a factor variable containing thousands of levels. Conversely, the "initial list status" variable contained only one value for all 39,000+ rows.

The variables we kept were those we judged to be attributes of the borrower (or the borrower's financial history) that could potentially be related either directly or indirectly to what interest rate might be paid on a small to medium sized loan. Some of these included the date of the borrower's first credit line, whether or not that borrower was a homeowner, how long the borrower had been with his or her current employer, annual income, whether the borrower had missed payments on a previous loan, and the borrower's zip code, which was reduced to a factor variable with 10 levels (0-9, each representing the first zip code digit).

In terms of cleaning the data, one variable proved moderately difficult to format so as to be of use for analysis. This variable "earliest_cr_line" signified the date at which the borrower in question initially took out his or her first credit line. The problem arose due to the formatting of this variable, in which half of the values took on a "day-month" format and the other half took on a "month-year" format. Useless in this format, and therefore nearly eliminated before model building, we were able to reformat this column in Excel to display for all rows a "MM/DD/YY" format. Beyond this relatively minor setback, we did not encounter any major impediments to data cleaning. Some variables required a quick reformatting from a 'char' to 'str' value but this was addressed without difficulty.

Before building regression models, we further prepared the data by addressing missing values. After choosing which variables to keep and otherwise cleaning the data, we made the decision to eliminate all rows that did not contain complete cases. Fortunately, this amounted to eliminating only 1,500 of our approximate 40,000 rows, hardly enough to significantly affect our regression analysis. Next, we made a new variable representing the length of time between the time a borrower took out his or her first credit line and the time that borrower took out the Lending Club loan in question. This gave some quantitative significance to the "earliest_cr_line" variable, and made it numeric rather than in date form. Finally, before we ran any regression, we split the data into training (60%), validation (20%) and evaluation (20%) sets.

Models

As our target variable is a continuous value, we naturally started our model building with a multiple linear regression, starting with a saturated model before reducing the number of variables with both forward and backward stepwise selection and finally moving to a regularized regression model with Elasticnet. Additionally, tree-based models were used including Random Forest and Gradient Boosting Machine (GBM).

- **Multiple linear regression and Multicollinearity**

  A standard multiple linear regression model with the following form was used:

  $$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_k X_k + \varepsilon.$$

  where Y is the response variable (interest rate) and X represents the predictor variables.

  A variance inflation factor (VIF) was used to identify multicollinearity issues within the multiple linear regression model. The variance inflation factor for each X is calculated as follows:

  Step 1 – run an OLS regression that has X as a function of all the other predictor variables.

  Step 2 – calculate the VIF, where the VIF = $1/(1-R^2)$.

Step 3 – compare the VIF against a threshold. The threshold used for the analysis was 2.5, which corresponds to an R^2 of 0.6 between X and the other predictor variables.

For the variables with a VIF above the threshold of 2.5 the following process was used to remove them from the model:

Step 1 – calculate the average correlation between the each of the offending variables and the other predictor variables.

Step 2 – rank the variables by average correlation and remove the variable with the largest average correlation from the multiple linear regression model.

Step 3 – re-run the model and calculate the variance inflation factors.

Step 4 – repeat the above steps until all of the VIFs are below the threshold.

- **Stepwise**

Forward selection on AIC was used to reduce the number of variables in the multiple linear regression model. Forward selection starts with no potential predictor variables and at each step it adds the predictor variable, such that the resulting model has the best AIC value. This continues until all the predictor variables have been added to the model or the AIC value starts getting worse.

Backward elimination on AIC was used to reduce the number of variables in the multiple linear regression model. Backward elimination starts with all of the potential predictor variables and at each step it deletes the predictor variable, such that the resulting model has the worst AIC value. This continues until all the predictor variables have been deleted from the model or the AIC value starts increasing.

- **Regularized linear regression**

For regularized linear regression, the regularization is computed by elasticnet.

$$\min_{\beta_0,\beta} \frac{1}{N} \sum_{i=1}^{N} w_i l(y_i, \beta_0 + \beta^T x_i) + \lambda \left[(1-\alpha)||\beta||_2^2/2 + \alpha||\beta||_1\right]$$

In which, α controls the balance between lasso and ridge and λ controls the overall strength of the penalty. The regularized linear regression was performed using caret package calling the glmnet method. The parameters α and λ were optimized by 5-fold cross-validation. The important variables were plot using varImp() function.

- **Random Forest**

Random Forest is tree-based model. It's an ensemble technique compared to single trees model. Each model in the ensemble is used to generate a prediction for new sample and these predictions are averaged to give the forest's prediction. One of random forests' tuning parameters is the number of randomly selected predictors *mtry*, which is usually suggested to be one-third of the number of predictors. The other parameter is *ntree*, which is the number of

bootstrap samples. The default value for ntree is 500. However, at least 1000 is usually suggested.
The random forest model is performed using randomForest package using default value for mtry (6 in current case) and 1000 for ntree. The variable importance was plotted using varImpPlot() function.

- **Gradient Boosting Machine**
  Gradient Boosting Machine is a boosting model, which was originally developed for classification problems. Given a loss function (e.g., squared error for regression) and a weak learner (e.g., regression trees), the algorithm of gradient boosting seeks to find an additive model that minimizes the loss function. The algorithm is typically initialized with the best guess of the response (e.g., the mean of the response in regression). The gradient (e.g., residual) is calculated, and a model is then fit to the residuals to minimize the loss function. The current model is added to the previous model, and the procedure continues for a user-specified number of iterations (3). The parameters for gbm model was tuned using caret package calling gbm method. The tune grid was interaction depth seq(1, 7, 2), ntrees seq(100, 1000, 50) and shrinkage (0.01 or 0.1). Then the best parameters based on RMSE was chosen for final model. The relative influence of variables were plotted using summary().

Results:

- **Multiple linear regression**

  The initial multiple linear regression model contained all 24 variables. When a variance inflation factor (vif) was applied it appeared that there were four aliased coefficients, which arises when variables are linearly dependent on other variables. Therefore, the four variables ('grade', 'subgrade', 'issue_d' & 'cr_line_age') were removed and the variance inflation factor was re-applied to the remaining variables.

  Using a variance inflation factor threshold of 2.5, four variables looked to be problematic. One of the variables, 'zip_code', was a categorical variable, which could be safely ignored. For the remaining three variables ('loan_amnt', 'funded_amnt_inv' and 'installment') the average correlation was calculated. These variables were then ranked and the variable with the largest average correlation was removed from the model and the variance inflation factor was applied to the updated model. This process was repeated until all variables were under the 2.5 threshold. This resulted in the removal of the 'loan_amnt' and 'funded_amnt_inv' variables

  This resulted in a multiple linear regression model, corrected for multicollinearity, that contained the following 18 predictor variables:

  'term'; 'installment'; 'emp_length'; 'home_ownership'; 'annual_inc'; 'is_inc_v'; 'purpose'; 'zip_code'; 'addr_state'; 'dti'; 'delinq_2yrs'; 'inq_last_6mths'; 'open_acc'; 'pub_rec'; 'revol_bal'; 'revol_util'; 'total_acc'; 'cr_line_age';

```
Residual standard error: 0.02489 on 23106 degrees of freedom
Multiple R-squared:  0.5534,    Adjusted R-squared:  0.5516
F-statistic: 318.1 on 90 and 23106 DF,  p-value: < 2.2e-16
```

The model has an R^2 of 0.5534. Some notable predictor variables positively correlated with the interest rate are 'delinq_2yrs', 'pub_rec', 'revol_util'. This is consistent with expectations. The 'annual_inc' variable was also positively correlated with the interest rate. This was a little puzzling, but the coefficient was very small and the variable was not significant. Predictor variables negatively correlated with the interest rate are 'emp_length', 'dti' and 'cr_line_age'. Again these are consistent with expectations.

- **Backward and forward selection**

  Applying both forward selection and backward elimination resulted in the same multiple linear regression model that contained the following 16 variables:

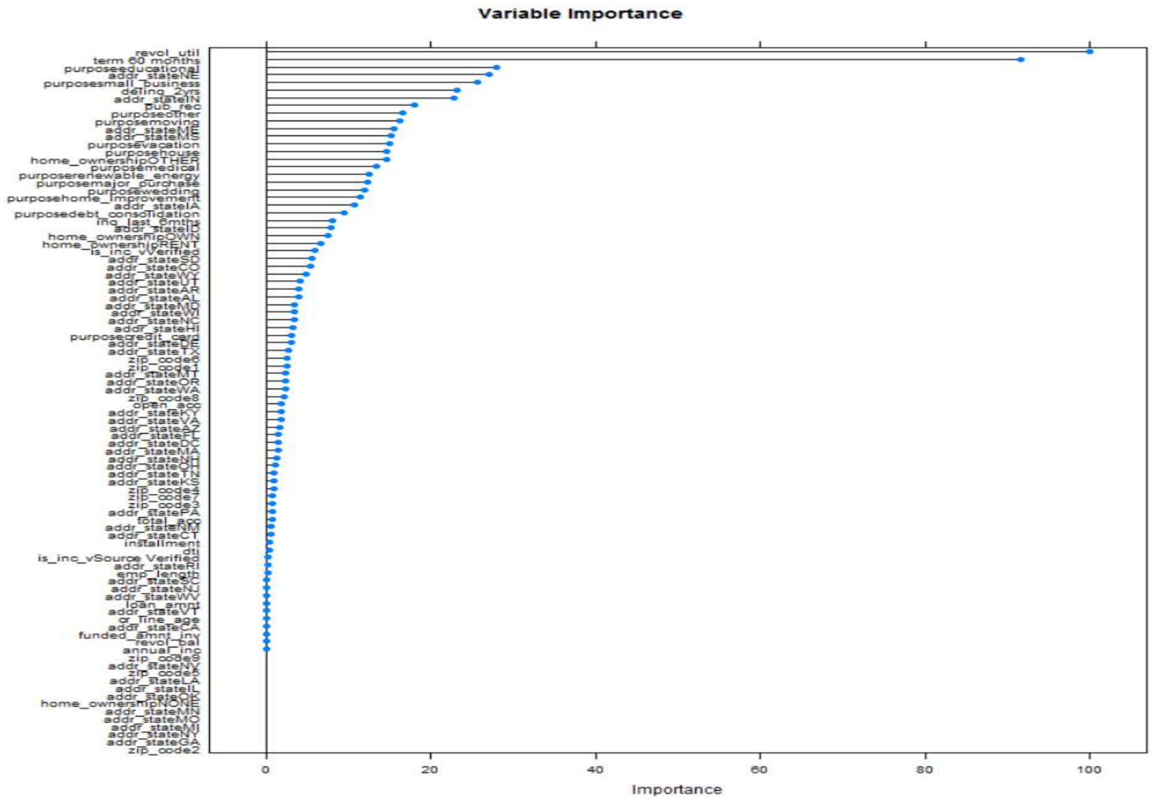
  'term'; 'installment'; 'emp_length'; 'home_ownership'; 'is_inc_v'; 'purpose'; 'zip_code'; 'dti'; 'delinq_2yrs'; 'inq_last_6mths'; 'open_acc'; 'pub_rec'; 'revol_bal'; 'revol_util'; 'total_acc'; 'cr_line_age';

```
Residual standard error: 0.02489 on 23156 degrees of freedom
Multiple R-squared:  0.5524,    Adjusted R-squared:  0.5517
F-statistic: 714.5 on 40 and 23156 DF,  p-value: < 2.2e-16
```

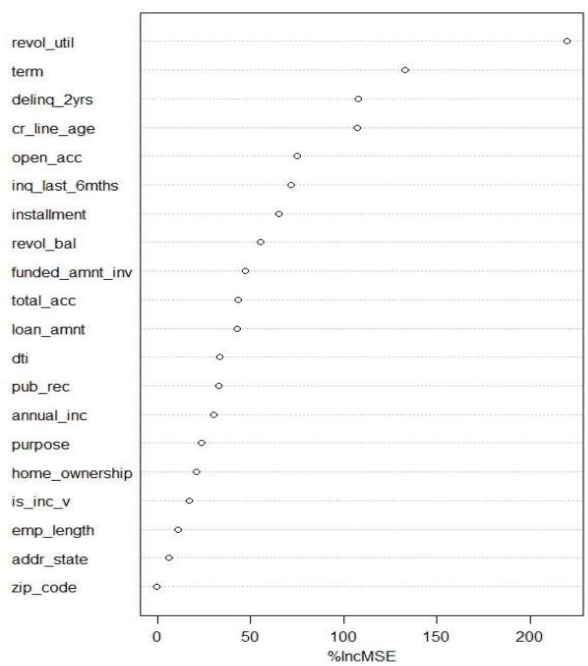
  The model has an R^2 of 0.5524 and is more parsimonious than the above multiple linear regression model. Also the same notable predictor variables have coefficients with a direction that is consistent with expectations. The full summary output from both multiple linear regression models is presented in Appendix A.

- **Regularized linear regression**

  By 5-fold cross-validation, the parameters α and λ were optimized as 0.55 and 3.466×10^{-5} respectively. The most important variables for this model were revol_util (the revolving line utilization rate, the amount of credit the borrower is using relative to all available revolving credit) and the term, which is the number of payments on the loan, the value is either 36 or 60 months.

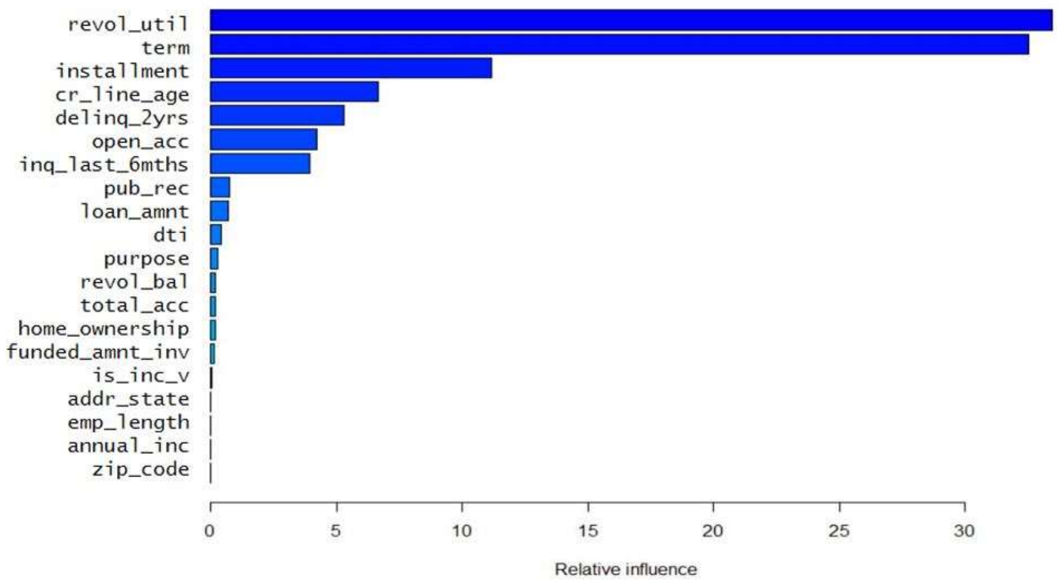


- **Random Forest**
  Using default value for mtry (6 in current case) and 1000 for ntrees, the top important variables for this model were revol_util and term, which were the same for regularized linear regression model.



- **Gradient Boosting Machine**

  The parameters of gbm were optimized as 7 for interaction depth, 1000 for ntrees and 0.1 for shrinkage. The most relative influence for this models were also revol_util and term.



- **Model comparison**

  Using validation dataset, the MSE and SE were computed for full multiple linear regression, backward/forward selected model, regularized linear regression, random forest and GBM models. The GBM had the lowest MSE (1.93 X 10^{-4}) and SD (3.90X10^{-6}). Therefore, the gbm model was chosen as our best model for prediction on evaluation dataset.

	Full Model	Backward/Forward	Regularized	Random Forest	GBM
MSE	6.207E-04	6.203E-04	5.827E-04	4.54E-04	1.93E-04
SD	1.068E-05	1.066E-05	1.007E-05	7.64E-06	3.90E-06

- **Prediction**

  Using gbm model, the interest rates were predicted as numeric number, range from 10.35% to 15.06% with mean 12.7%. Although the mean of predicted values was closed to real value (12.11%), the range of predicted values is narrower than real values (range from 5.42% to 24.11%).

Discussions and Conclusions:

This analysis was meant to serve as a way to better understand how a borrower is assigned an interest rate for a loan. We ultimately concluded that our dataset contained about 10-14 significant variables, the most significant of which being 'revol_util', which represents the percentage of the loan the borrower is using, and 'term', which signifies either a 36 month or 60 month term.

Our models included two multiple linear regressions, a regularized model, random forest and gradient boosting machine. Side by side comparison of our models suggests that the GBM is the best fit for our data in terms of adjusted MSE and standard deviation. As a tree based model, the GBM tells us which variables are the most significant but does not provide any coefficients and therefore does not tell us how a variable affects the response. However, if we use one of our other models, such as a multiple linear regression model, we can put the variables into such a real world context. With respect to the 'term' variable, we can conclude with a linear model that the move from a 36 month to a 60 month term entails a significantly higher interest rate, and similarly, that the move to a higher 'revol_util' percentage leads to a higher interest rate as well.

We recognize some limitations, necessary due to the dataset we are employing. Firstly, peer-to-peer lending services, while growing, still represent a niche in the financial services market. Secondly, with a limit of $40,000 for the loans we are examining, we cannot include most home or business loans. Therefore we are unable to conclude with absolute certainty that our conclusions can be extrapolated to more traditional loans or loans of a larger size. Having acknowledged this, we have no reason to believe our conclusions would not translate to other kinds of loans or lending agencies.

However, any future works could and should address these limitations. The next step would be to collect data for loans with higher dollar amounts, and from a number disparate lending sources, including larger banks. Furthermore, in this analysis, we had only one response variable, interest rate. In the future, we may consider another target variable or variables. For example, how long will it take for the borrower to pay off this loan at his or her current income or even taking into account future earning potential? This may be something some borrowers could be interested in knowing. Indeed, taking out a loan, especially a major loan for a home or business could be the most significant financial decision most individuals or families will ever make. Being able to apply predictive analytics to ensure a smooth process and avoid major pitfalls could be immensely useful for any potential borrower.

References:

(1) Mitigating adverse selection in P2P lending: Empirical evidence from Prosper.com. Weiss GN, Pelger K, Horsch A. July 2010. Available at SSRN: http://ssrn.com/abstract=1650774 or http://dx.doi.org/10.2139/ssrn.1650774.

(2) Determinants of Default in P2P Lending. C. Serrano-Cinca, B. Gutiérrez-Nieto and L. López-Palacios. Public Library of Science, October 2015.

(3) Applied Predictive Modeling. Max Kuhn and Kjell Johnson. P173-220. Chapter 8 Regression Trees and Ruled-Based Models. 2013.

Appendix A - Additional R Output

18 variable multiple linear regression model summary output

```
Call:
lm(formula = int_rate ~ . - loan_amnt - funded_amnt_inv, data = loan_train)


Residuals:
      Min        1Q    Median        3Q       Max 
-0.092456 -0.017590 -0.001503  0.016198  0.125571 


Coefficients:
                           Estimate Std. Error t value Pr(>|t|)    
(Intercept)               7.723e-02  2.054e-02   3.760 0.000170 ***
term 60 months            3.495e-02  3.894e-04  89.762  < 2e-16 ***
installment               4.626e-05  9.371e-07  49.370  < 2e-16 ***
emp_length               -9.326e-05  4.981e-05  -1.873 0.061143 .  
home_ownershipNONE        1.786e-03  2.492e-02   0.072 0.942876    
home_ownershipOTHER       9.638e-03  3.379e-03   2.852 0.004344 ** 
home_ownershipOWN         4.926e-03  6.582e-04   7.485 7.42e-14 ***
home_ownershipRENT        4.223e-03  4.023e-04  10.497  < 2e-16 ***
annual_inc                3.220e-09  2.578e-09   1.249 0.211661    
is_inc_vSource Verified  -7.893e-04  4.170e-04  -1.893 0.058383 .  
is_inc_vVerified          1.232e-03  4.281e-04   2.877 0.004015 ** 
purposecredit_card        6.409e-04  9.724e-04   0.659 0.509870    
purposedebt_consolidation 4.054e-03  8.902e-04   4.554 5.29e-06 ***
purposeeducational        1.533e-02  2.016e-03   7.603 3.00e-14 ***
purposehome_improvement   5.410e-03  1.041e-03   5.198 2.03e-07 ***
purposehouse              7.118e-03  1.862e-03   3.822 0.000133 ***
purposemajor_purchase     6.405e-03  1.094e-03   5.853 4.88e-09 ***
purposemedical            7.336e-03  1.509e-03   4.862 1.17e-06 ***
purposemoving             8.413e-03  1.631e-03   5.158 2.52e-07 ***
purposeother              8.851e-03  9.839e-04   8.996  < 2e-16 ***
purposerenewable_energy   7.883e-03  3.275e-03   2.407 0.016077 *  
purposesmall_business     1.414e-02  1.139e-03  12.417  < 2e-16 ***
purposevacation           8.379e-03  1.915e-03   4.375 1.22e-05 ***
purposewedding            6.510e-03  1.338e-03   4.865 1.15e-06 ***
zip_code1                -1.331e-02  1.052e-02  -1.265 0.205891    
zip_code2                -1.260e-02  1.659e-02  -0.759 0.447721    
zip_code3                -8.048e-03  1.854e-02  -0.434 0.664219    
zip_code4                -1.713e-02  1.907e-02  -0.899 0.368830    
zip_code5                -6.331e-03  2.094e-02  -0.302 0.762366    
zip_code6                -1.558e-02  1.980e-02  -0.787 0.431453    
zip_code7                -4.365e-03  1.866e-02  -0.234 0.815064    
zip_code8                 3.197e-04  1.985e-02   0.016 0.987151    
zip_code9                -1.092e-02  2.040e-02  -0.535 0.592551    
addr_stateAL             -1.414e-03  1.119e-02  -0.126 0.899461    
addr_stateAR             -9.891e-03  1.303e-02  -0.759 0.447994    
addr_stateAZ             -9.974e-03  1.125e-02  -0.887 0.375190    
addr_stateCA             -1.224e-03  3.503e-03  -0.349 0.726759    
addr_stateCO             -1.468e-02  1.125e-02  -1.305 0.191879    
addr_stateCT             -1.244e-02  2.055e-02  -0.606 0.544755    
addr_stateDC             -1.006e-03  1.478e-02  -0.068 0.945741    
addr_stateDE              1.519e-03  1.966e-02   0.077 0.938421    
addr_stateFL             -2.701e-03  1.109e-02  -0.243 0.807625    
addr_stateGA             -3.786e-03  1.111e-02  -0.341 0.733328    
addr_stateHI             -2.710e-03  4.266e-03  -0.635 0.525258    
addr_stateIA             -9.134e-03  1.899e-02  -0.481 0.630561    
addr_stateID             -5.235e-03  1.675e-02  -0.313 0.754656    
addr_stateIL              1.909e-03  1.376e-02   0.139 0.889708    
addr_stateIN              3.104e-03  1.763e-02   0.176 0.860262    
addr_stateKS              1.096e-03  1.392e-02   0.079 0.937258    
addr_stateKY              5.876e-03  1.255e-02   0.468 0.639595    
addr_stateLA             -7.495e-03  1.299e-02  -0.577 0.563966    
addr_stateMA             -1.292e-02  2.053e-02  -0.629 0.529236    
addr_stateMD              3.239e-03  1.464e-02   0.221 0.824942    
addr_stateME             -2.368e-02  3.226e-02  -0.734 0.462852    
addr_stateMI              4.706e-03  1.254e-02   0.375 0.707506    
addr_stateMN             -5.663e-03  1.238e-02  -0.457 0.647397    
addr_stateMO              1.966e-03  1.379e-02   0.143 0.886670    
addr_stateMS              6.738e-03  1.306e-02   0.516 0.605818    
addr_stateMT             -4.138e-03  1.297e-02  -0.319 0.749750    
addr_stateNC             -1.582e-03  1.465e-02  -0.108 0.913982    
addr_stateNE              2.452e-02  2.235e-02   1.097 0.272648    
addr_stateNH             -1.351e-02  2.069e-02  -0.653 0.513801    
addr_stateNJ             -1.244e-02  2.053e-02  -0.606 0.544543    
addr_stateNM             -1.123e-02  1.145e-02  -0.981 0.326781    
addr_stateNV             -1.086e-02  1.129e-02  -0.962 0.336150    
addr_stateNY             -2.563e-04  1.940e-02  -0.013 0.989459    
```

```
addr_stateOH              4.091e-03  1.250e-02   0.327 0.743479
addr_stateOK             -7.453e-03  1.304e-02  -0.571 0.567781
addr_stateOR             -2.902e-03  3.818e-03  -0.760 0.447188
addr_statePA             -7.991e-04  1.942e-02  -0.041 0.967183
addr_stateRI             -1.117e-02  2.065e-02  -0.541 0.588539
addr_stateSC              1.270e-03  1.470e-02   0.086 0.931125
addr_stateSD             -8.435e-03  1.310e-02  -0.644 0.519548
addr_stateTN             -9.111e-03  1.338e-02  -0.681 0.495970
addr_stateTX             -5.719e-03  1.291e-02  -0.443 0.657871
addr_stateUT             -1.393e-02  1.136e-02  -1.226 0.220118
addr_stateVA              1.622e-03  1.464e-02   0.111 0.911747
addr_stateVT             -1.011e-02  2.064e-02  -0.490 0.624373
addr_stateWA              7.456e-04  3.650e-03   0.204 0.838164
addr_stateWI             -7.775e-03  1.250e-02  -0.622 0.533972
addr_stateWV              7.505e-04  1.481e-02   0.051 0.959590
addr_stateWY             -8.382e-03  1.169e-02  -0.717 0.473371
dti                      -1.721e-04  2.844e-05  -6.051 1.46e-09 ***
delinq_2yrs               1.424e-02  3.373e-04  42.203  < 2e-16 ***
inq_last_6mths            4.908e-03  1.570e-04  31.260  < 2e-16 ***
open_acc                  1.147e-03  5.356e-05  21.421  < 2e-16 ***
pub_rec                   1.110e-02  6.988e-04  15.887  < 2e-16 ***
revol_bal                -1.628e-07  1.255e-08 -12.972  < 2e-16 ***
revol_util                6.056e-02  6.697e-04  90.429  < 2e-16 ***
total_acc                -4.324e-04  2.201e-05 -19.644  < 2e-16 ***
cr_line_age              -1.951e-06  7.829e-08 -24.920  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1


Residual standard error: 0.02489 on 23106 degrees of freedom
Multiple R-squared:  0.5534, Adjusted R-squared:  0.5516
F-statistic: 318.1 on 90 and 23106 DF,  p-value: < 2.2e-16
```

16 variable multiple linear regression model summary output

```
Call:
lm(formula = int_rate ~ term + installment + emp_length + home_ownership +
    is_inc_v + purpose + zip_code + dti + delinq_2yrs + inq_last_6mths +
    open_acc + pub_rec + revol_bal + revol_util + total_acc +
    cr_line_age, data = loan_train)

Residuals:
     Min       1Q   Median       3Q      Max
-0.09271 -0.01760 -0.00146  0.01627  0.12286

Coefficients:
                            Estimate Std. Error t value Pr(>|t|)
(Intercept)                6.483e-02  1.176e-03  55.134  < 2e-16 ***
term 60 months             3.491e-02  3.889e-04  89.763  < 2e-16 ***
installment                4.646e-05  9.261e-07  50.165  < 2e-16 ***
emp_length                -9.222e-05  4.966e-05  -1.857 0.063337 .
home_ownershipNONE         1.648e-03  2.490e-02   0.066 0.947248
home_ownershipOTHER        9.513e-03  3.372e-03   2.821 0.004794 **
home_ownershipOWN          4.936e-03  6.558e-04   7.526 5.42e-14 ***
home_ownershipRENT         4.217e-03  3.974e-04  10.612  < 2e-16 ***
is_inc_vSource Verified   -7.424e-04  4.162e-04  -1.784 0.074456 .
is_inc_vVerified           1.249e-03  4.273e-04   2.923 0.003470 **
purposecredit_card         6.034e-04  9.712e-04   0.621 0.534428
purposedebt_consolidation  4.017e-03  8.893e-04   4.516 6.32e-06 ***
purposeeducational         1.531e-02  2.014e-03   7.602 3.02e-14 ***
purposehome_improvement    5.490e-03  1.039e-03   5.284 1.28e-07 ***
purposehouse               7.060e-03  1.861e-03   3.794 0.000149 ***
purposemajor_purchase      6.440e-03  1.093e-03   5.889 3.93e-09 ***
purposemedical             7.284e-03  1.507e-03   4.834 1.35e-06 ***
purposemoving              8.401e-03  1.629e-03   5.156 2.54e-07 ***
purposeother               8.887e-03  9.830e-04   9.041  < 2e-16 ***
purposerenewable_energy    7.942e-03  3.273e-03   2.427 0.015249 *
purposesmall_business      1.412e-02  1.137e-03  12.416  < 2e-16 ***
purposevacation            8.350e-03  1.913e-03   4.364 1.28e-05 ***
purposewedding             6.560e-03  1.337e-03   4.908 9.28e-07 ***
zip_code1                 -1.172e-03  6.628e-04  -1.768 0.077156 .
zip_code2                  1.116e-03  7.159e-04   1.559 0.119017
zip_code3                  1.578e-03  6.847e-04   2.305 0.021180 *
zip_code4                 -1.305e-04  8.443e-04  -0.155 0.877139
zip_code5                 -3.129e-04  1.069e-03  -0.293 0.769817
zip_code6                 -1.198e-03  8.279e-04  -1.447 0.147801
zip_code7                  1.801e-03  7.339e-04   2.453 0.014161 *
zip_code8                  9.114e-04  7.984e-04   1.142 0.253669
zip_code9                  4.775e-04  6.069e-04   0.787 0.431474
dti                       -1.797e-04  2.785e-05  -6.453 1.12e-10 ***
```

```
delinq_2yrs              1.425e-02  3.370e-04  42.295  < 2e-16 ***
inq_last_6mths           4.906e-03  1.563e-04  31.382  < 2e-16 ***
open_acc                 1.147e-03  5.338e-05  21.491  < 2e-16 ***
pub_rec                  1.116e-02  6.975e-04  16.000  < 2e-16 ***
revol_bal               -1.594e-07  1.235e-08 -12.909  < 2e-16 ***
revol_util               6.061e-02  6.685e-04  90.662  < 2e-16 ***
total_acc               -4.295e-04  2.188e-05 -19.627  < 2e-16 ***
cr_line_age             -1.949e-06  7.815e-08 -24.935  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.02489 on 23156 degrees of freedom
Multiple R-squared:  0.5524, Adjusted R-squared:  0.5517
F-statistic: 714.5 on 40 and 23156 DF,  p-value: < 2.2e-16
```

Appendix B – R Code

```
# Load packages
library(ggplot2)
library(lubridate)
library(leaps)
library(car)
library(caret)
library(randomForest)
library(gbm)

#read in the csv file and clean the data
loan <- read.csv("LoanStats3a_finalVersion.csv", header = T, stringsAsFactors = FALSE)
choice <- c(3, 5, 6, 7, 8, 9, 10, 12, 13, 14, 15, 16, 20, 22, 23, 24, 25, 26, 27, 30, 31, 32, 33, 34)
loan1 <- loan[ ,choice]

######data clean#####

# term
loan1$term <- as.factor(loan1$term)
# int_rate
loan1$int_rate <- as.numeric(sub("%", "", loan1$int_rate))/100
# grade
loan1$grade <- as.factor(loan1$grade)
# subgrade
loan1$sub_grade <- as.factor(loan1$sub_grade)
# emp_length
loan1[loan1$emp_length == "< 1 year", "emp_length"] <- 0
loan1[loan1$emp_length == "1 year", "emp_length"] <- 1
loan1[loan1$emp_length == "2 years", "emp_length"] <- 2
loan1[loan1$emp_length == "3 years", "emp_length"] <- 3
loan1[loan1$emp_length == "4 years", "emp_length"] <- 4
loan1[loan1$emp_length == "5 years", "emp_length"] <- 5
loan1[loan1$emp_length == "6 years", "emp_length"] <- 6
loan1[loan1$emp_length == "7 years", "emp_length"] <- 7
loan1[loan1$emp_length == "8 years", "emp_length"] <- 8
```

```
loan1[loan1$emp_length == "9 years", "emp_length"] <- 9
loan1[loan1$emp_length == "10+ years", "emp_length"] <- 10
loan1[loan1$emp_length == "n/a", "emp_length"] <- NA
loan1$emp_length <- as.integer(loan1$emp_length)
# home_ownership
loan1$home_ownership <- as.factor(loan1$home_ownership)
# is_inc_v
loan1$is_inc_v <- as.factor(loan1$is_inc_v)
# issue_d
loan1$issue_d<-as.Date(mdy(loan1$issue_d))
# purpose
loan1$purpose <- as.factor(loan1$purpose)
# zip code
loan1$zip_code <- as.integer(substr(loan1$zip_code, 1, 1))
# addr_state
loan1$addr_state <- as.factor(loan1$addr_state)
# earliest_cr_line
loan1$earliest_cr_line<-as.Date(mdy(loan1$earliest_cr_line))
# revol_util
loan1$revol_util <- as.numeric(sub("%", "", loan1$revol_util ))/100
# create  cr_line_age variable
loan1$cr_line_age <- as.numeric(loan1$issue_d-loan1$earliest_cr_line)

#save the data frame into an rdata file in the local directory
save(loan1, file="loan.rdata")

#load the rdata file from the local directory
loan <-load("loan.rdata")

#remove the rows with NAs
#goes from 39,786 to 38,661 rows (1,125 rows removed)
loan1 <- loan1[complete.cases(loan1),]

#print out the 25 variable names
names(loan1)
# [1] "loan_amnt"       "funded_amnt_inv"  "term"           "int_rate"        "installment"
# [6] "grade"           "sub_grade"       "emp_length"      "home_ownership"   "annual_inc"
#[11] "is_inc_v"        "issue_d"         "purpose"         "zip_code"        "addr_state"
#[16] "dti"             "delinq_2yrs"     "earliest_cr_line" "inq_last_6mths"   "open_acc"    #[21] "pub_rec"
"revol_bal"       "revol_util"      "total_acc"       "cr_line_age"

#plot the distribution of the variables
ggplot(loan1,
    aes(x=loan_amnt))+geom_density()+ggtitle("loan_amnt Distribution")
ggplot(loan1,
    aes(x=funded_amnt_inv))+geom_density()+ggtitle("funded_amnt_inv Distribution")
ggplot(loan1,
    aes(x=term))+geom_density()+ggtitle("term Distribution")
```

```
ggplot(loan1,
    aes(x=int_rate))+geom_density()+ggtitle("int_rate Distribution")
ggplot(loan1,
    aes(x=installment))+geom_density()+ggtitle("installment Distribution")
ggplot(loan1,
    aes(x=grade))+geom_density()+ggtitle("grade Distribution")
ggplot(loan1,
    aes(x=sub_grade))+geom_density()+ggtitle("sub_grade Distribution")
ggplot(loan1,
    aes(x=emp_length))+geom_density()+ggtitle("emp_length Distribution")
ggplot(loan1,
    aes(x=home_ownership))+geom_density()+ggtitle("home_ownership Distribution")
ggplot(loan1,
    aes(x=annual_inc))+geom_density()+ggtitle("annual_inc Distribution")
ggplot(loan1,
    aes(x=is_inc_v))+geom_density()+ggtitle("is_inc_v Distribution")
ggplot(loan1,
    aes(x=issue_d))+geom_density()+ggtitle("issue_d Distribution")
ggplot(loan1,
    aes(x=purpose))+geom_density()+ggtitle("purpose Distribution")
ggplot(loan1,
    aes(x=zip_code))+geom_density()+ggtitle("zip_code Distribution")
ggplot(loan1,
    aes(x=addr_state))+geom_density()+ggtitle("addr_state Distribution")
ggplot(loan1,
    aes(x=dti))+geom_density()+ggtitle("dti Distribution")
ggplot(loan1,
    aes(x=delinq_2yrs))+geom_density()+ggtitle("delinq_2yrs Distribution")
ggplot(loan1,
    aes(x=earliest_cr_line))+geom_density()+ggtitle("earliest_cr_line Distribution")
ggplot(loan1,
    aes(x=inq_last_6mths))+geom_density()+ggtitle("inq_last_6mths Distribution")
ggplot(loan1,
    aes(x=open_acc))+geom_density()+ggtitle("open_acc Distribution")
ggplot(loan1,
    aes(x=pub_rec))+geom_density()+ggtitle("pub_rec Distribution")
ggplot(loan1,
    aes(x=revol_bal))+geom_density()+ggtitle("revol_bal Distribution")
ggplot(loan1,
    aes(x=revol_util))+geom_density()+ggtitle("revol_util Distribution")
ggplot(loan1,
    aes(x=total_acc))+geom_density()+ggtitle("total_acc Distribution")
ggplot(loan1,
    aes(x=cr_line_age))+geom_density()+ggtitle("cr_line_age Distribution")

#correlation plots
ggplot(loan1,
    aes(x=loan_amnt,
```

```
    y=int_rate))+geom_point()+geom_smooth(method='lm')+ggtitle("loan_amnt")
ggplot(loan1,
  aes(x=funded_amnt_inv,
    y=int_rate))+geom_point()+geom_smooth(method='lm')+ggtitle("funded_amnt_inv")
ggplot(loan1,
  aes(x=term,
    y=int_rate,
    group=1))+geom_point()+geom_smooth(method='lm')+ggtitle("term")
ggplot(loan1,
  aes(x=installment,
    y=int_rate))+geom_point()+geom_smooth(method='lm')+ggtitle("installment")
ggplot(loan1,
  aes(x=grade,
    y=int_rate,
    group=1))+geom_point()+geom_smooth(method='lm')+ggtitle("grade")
ggplot(loan1,
  aes(x=sub_grade,
    y=int_rate,
    group=1))+geom_point()+geom_smooth(method='lm')+ggtitle("sub_grade")
ggplot(loan1,
  aes(x=emp_length,
    y=int_rate))+geom_point()+geom_smooth(method='lm')+ggtitle("emp_length")
ggplot(loan1,
  aes(x=home_ownership,
    y=int_rate,
    group=1))+geom_point()+geom_smooth(method='lm')+ggtitle("home_ownership")
ggplot(loan1,
  aes(x=annual_inc,
    y=int_rate))+geom_point()+geom_smooth(method='lm')+ggtitle("annual_inc")
ggplot(loan1,
  aes(x=is_inc_v,
    y=int_rate,
    group=1))+geom_point()+geom_smooth(method='lm')+ggtitle("is_inc_v")
ggplot(loan1,
  aes(x=issue_d,
    y=int_rate))+geom_point()+geom_smooth(method='lm')+ggtitle("issue_d")
ggplot(loan1,
  aes(x=purpose,
    y=int_rate,
    group=1))+geom_point()+geom_smooth(method='lm')+ggtitle("purpose")
ggplot(loan1,
  aes(x=zip_code,
    y=int_rate))+geom_point()+geom_smooth(method='lm')+ggtitle("zip_code")
ggplot(loan1,
  aes(x=addr_state,
    y=int_rate,
    group=1))+geom_point()+geom_smooth(method='lm')+ggtitle("addr_state")
ggplot(loan1,
```

```
    aes(x=dti,
      y=int_rate))+geom_point()+geom_smooth(method='lm')+ggtitle("dti")
ggplot(loan1,
    aes(x=delinq_2yrs,
      y=int_rate))+geom_point()+geom_smooth(method='lm')+ggtitle("delinq_2yrs")
ggplot(loan1,
    aes(x=earliest_cr_line,
      y=int_rate))+geom_point()+geom_smooth(method='lm')+ggtitle("earliest_cr_line")
ggplot(loan1,
    aes(x=inq_last_6mths,
      y=int_rate))+geom_point()+geom_smooth(method='lm')+ggtitle("inq_last_6mths")
ggplot(loan1,
    aes(x=open_acc,
      y=int_rate))+geom_point()+geom_smooth(method='lm')+ggtitle("open_acc")
ggplot(loan1,
    aes(x=pub_rec,
      y=int_rate))+geom_point()+geom_smooth(method='lm')+ggtitle("pub_rec")
ggplot(loan1,
    aes(x=revol_bal,
      y=int_rate))+geom_point()+geom_smooth(method='lm')+ggtitle("revol_bal")
ggplot(loan1,
    aes(x=revol_util,
      y=int_rate))+geom_point()+geom_smooth(method='lm')+ggtitle("revol_util")
ggplot(loan1,
    aes(x=total_acc,
      y=int_rate))+geom_point()+geom_smooth(method='lm')+ggtitle("total_acc")
ggplot(loan1,
    aes(x=cr_line_age,
      y=int_rate))+geom_point()+geom_smooth(method='lm')+ggtitle("cr_line_age")

#correlation matrix
#create a data frame without the factors & dates for the correlation matrix
loan1_cor <- loan1[ , -which(names(loan1) %in%
c("term","grade","sub_grade","home_ownership","is_inc_v","purpose","addr_state","issue_d","earliest
_cr_line"))]
#view the correlation matrix
View(round(cor(loan1_cor),2))
cor(loan1_cor)

##### Model building #####

#load the rdata file from the local directory
loan <-load("loan.rdata")
loan1$zip_code <- as.factor(loan1$zip_code) # convert zipcode to factor

#remove the rows with NAs
#goes from 39,786 to 38,661 rows (1,125 rows removed)
loan1 <- loan1[complete.cases(loan1),]
```

```
# Remove grade, subgrade, issue_d, earliest_cr_line
ex <- c(6, 7, 12, 18)
loan <- loan1[, -ex]


#####dataset#####
#partition the data into Training (60%), Test (20%) & Validation (20%) groups

# sample size = 38,661
n <- dim(loan)[1]
# set random number generator seed
set.seed(1125)
# randomly sample 20% test
test <- sample(n, round(n/5))
# define the test dataset (7,732 rows)
loan_test <- loan[test,]
loan_left <- loan[-test,]
#sample size = 30,929
nn <- dim(loan_left)[1]
# set random number generator seed
set.seed(1125)
# randomly sample 20% validation (25% of the remaining data)
valid <- sample(nn, round(nn/4))
# define the validation dataset (7,732 rows)
loan_valid <- loan_left[valid,]
# define the training dataset (23,197 rows)
loan_train <- loan_left[-valid,]

##### Linear regression and Multicollinearity #####
loan_full_back <- lm(int_rate~., data=loan_train)
# check vif
vif(loan_full_back) #loan_amnt, funded_amnt_inv, and installment higher than 2.50

# Create correlation matrix between predictors (numeric variable only)
loan.num <- loan[ , -which(names(loan) %in%
                        c("int_rate", "term", "home_ownership", "is_inc_v", "purpose", "zip_code",
"addr_state"))]
cortable <- cor(loan.num)

(sum(cortable["loan_amnt",])-1)/15 # average correlation for loan_amnt 0.221002
(sum(cortable[c("funded_amnt_inv"),])-1)/15 # average correlation for funded_amnt_inv 0.21452
(sum(cortable[c("installment"),])-1)/15 # average correlation for installment 0.21432

# Remove the higher average correlation, loan_amnt
loan_full_back1 <- lm(int_rate~.-loan_amnt, data=loan_train)
vif(loan_full_back1) # funded_amnt_inv and installment still higher than 2.50
```

```
# Remove funded_amnt_inv
loan_full_back2 <- lm(int_rate~.-loan_amnt-funded_amnt_inv, data=loan_train) # Final full model
vif(loan_full_back2) # OK
summary(loan_full_back2)

##### Stepwise #####
#Stepwise Backward Selection
loan_back <- step(loan_full_back2, data=loan_train, direction="backward")
summary(loan_back)
formula(loan_back)
# int_rate ~ term + installment + emp_length + home_ownership +
#   is_inc_v + purpose + zip_code + dti + delinq_2yrs + inq_last_6mths +
#   open_acc + pub_rec + revol_bal + revol_util + total_acc +
#   cr_line_age

#Stepwise Forward Selection
loan_empty_for <- lm(int_rate~1,data=loan_train)
loan_for <- step(loan_empty_for,
            scope=list(lower=formula(loan_empty_for),
                  upper=formula(loan_full_back2)),
            data=loan_train, direction="forward")
formula(loan_for)
# int_rate ~ revol_util + term + installment + delinq_2yrs + cr_line_age +
#   inq_last_6mths + purpose + pub_rec + home_ownership + revol_bal +
#   open_acc + total_acc + dti + zip_code + is_inc_v + emp_length

summary(loan_for)

# Residual plot
par(mfrow = c(2, 2))
plot(loan_back)

##### Regularization#####
# Regularized regression elasticnet

set.seed(1125)
control <- trainControl(method="cv", number=5)
fit.glmnet <- train(int_rate~., data=loan_train, method="glmnet", metric = "RMSE",trControl=control)
fit.glmnet$bestTune
#   alpha      lambda
# 4  0.55 3.466128e-05

# With center and scale
fit.glmnet1 <- train(int_rate~., data=loan_train, method="glmnet", metric = "RMSE",
preProc=c("center","scale"), trControl=control)
fit.glmnet1 # no differece from the model without pre-process

# Plot variable importance
```

```
plot(varImp(fit.glmnet, lambda = fit.glmnet$bestTune$lambda), main = "Variable Importance")

##### Random Forest #####
fit.rf <- randomForest(int_rate~., data=loan_train, importance = TRUE, ntrees = 1000)
# mtry equals to 6.
# default mtry (number of predictors) is the number of predictors divided by 3.

# Plot variable importance
varImpPlot(fit.rf)

##### Gradient Boosting Machine #####
# Tune gbm model
gbmGrid <- expand.grid(.interaction.depth = seq(1, 7, by = 2),
                .n.trees = seq(100, 1000, by = 50),
                .shrinkage = c(0.01, 0.1),
                .n.minobsinnode = 10)
set.seed(1125)
gbmTune <- train(int_rate~., data=loan_train, method = "gbm", tuneGrid = gbmGrid, verbose = FALSE)
gbmTune

# Tuning parameter 'n.minobsinnode' was held constant at a value of 10
# RMSE was used to select the optimal model using  the smallest value.
# The final values used for the model were n.trees = 1000, interaction.depth = 7, shrinkage = 0.1
# and n.minobsinnode = 10.

# gbm model with tuned parameters
fit.gbm <- gbm(int_rate~., data=loan_train,
          distribution = "gaussian",
          n.trees = 1000,
          interaction.depth = 7,
          shrinkage = 0.1)

# plot the relative influence
summary(fit.gbm)

##### Compare MSE between models #####

# Full stack 2
#MSE
mean((loan_test$int_rate - predict(loan_full_back2, loan_test))^2)
# 0.0006209339
sd((loan_test$int_rate - predict(loan_full_back2, loan_test))^2)/sqrt(nrow(loan_test))
# 1.06945e-05

# Loan_back
#MSE
mean((loan_test$int_rate - predict(loan_back, loan_test))^2)
# 0.0006189477
```

```
sd((loan_test$int_rate - predict(loan_back, loan_test))^2)/sqrt(nrow(loan_test))
# 1.0655e-05

# Loan_for
#MSE
mean((loan_test$int_rate - predict(loan_for, loan_test))^2)
# 0.0006189477
sd((loan_test$int_rate - predict(loan_for, loan_test))^2)/sqrt(nrow(loan_test))
# 1.0655e-05

# fit.glmnet
#MSE
mean((loan_test$int_rate - predict(fit.glmnet, loan_test))^2)
# 0.0005826064
sd((loan_test$int_rate - predict(fit.glmnet, loan_test))^2)/sqrt(nrow(loan_test))
# 1.007136e-05

# fit.glmnet1
#MSE
mean((loan_test$int_rate - predict(fit.glmnet1, loan_test))^2)
# 0.0005826064
sd((loan_test$int_rate - predict(fit.glmnet1, loan_test))^2)/sqrt(nrow(loan_test))
# 1.007136e-05

# fit.rf
#MSE
mean((loan_test$int_rate - predict(fit.rf, loan_test))^2)
# 0.000453902
sd((loan_test$int_rate - predict(fit.rf, loan_test))^2)/sqrt(nrow(loan_test))
# 7.635157e-06

# fit.gbm
#MSE
mean((loan_test$int_rate - predict(fit.gbm, loan_test, n.trees = 1000))^2)
# 0.0001929875
sd((loan_test$int_rate - predict(fit.gbm, loan_test, n.trees = 1000))^2)/sqrt(nrow(loan_test))
# 3.903445e-06


##### Prediction #####

int_rate <- predict(fit.gbm, loan_valid, n.trees = 1000)
```