Loan Default Modeling

Wilson Phurwo (wphurwo2)

May 18, 2020

Executive summary

Prediction of loan default probability is best achieved by penalized regression, lasso in particular with over 99% test accuracy and 100% specificity. The AIC stepwise variable selection also obtains the most contributing factors toward the likelihood of loan default. These variables include borrower's home ownership, loan purpose, and many indicators of borrower's historical credit performance. These models are still under the assumption of logistic regression. However, k-nearest neighbor performs poorly in predicting the occurrence of default although it costs substantial computational resources. It is also learned that borrowers who opt with short-term loans when long-term option is available are less likely to end up defaulting than when long-term is not an option. The study was originally conducted by [3], and this report emphasizes the finding with logistic regression.

1 Introduction

L&S Consultancy is honored to deliver this report as the foundation of a series of models that are highly accurate to predicting default probability. Many thanks to the Lending Club as the publication of their data enables companies like L&S Consultancy to conduct such project. This report delivers the uses and results of various models, such as the logistic regression and k-nearest neighbor, of their predictive power to classify the characteristics of loans and borrowers toward the likelihood of default.

High-accuracy predictive models are achieved by utilizing the penalized and stepwise regression. Both are still built under the logistic regression assumption. Through these methods, many variables are penalized / eliminated. These models achieve over 95% test accuracy. Some of them even obtained perfect specificity, that is zero false negative.

However, the k-nearest neighbor poorly predicts the probability of default. Although its sensitivity is as high as 99%, but it often overestimates the non-defaulted cases, resulting in low specificity. Therefore, it is not recommended to proceed with this model. Another major drawback of the k-nearest neighbor is that it costs abundant computational resource.

Home ownership appears to be an adequate predictor for most models. Whether a borrower rents, owns, or pays mortgages usually contributes to the likelihood of default. The purpose of the loan, which should be stated when the borrower applies for the loan, also

makes difference in the odds of default occurrences. Other than that, most adequate predictors are generally related to the borrower's credit performance, such as the number of derogatory public records, revolving line utilization rate, and the number of open credit lines.

This report begins with the introduction of models that are used intensively. This section offers the theories of all models used. Next, there is a series of exploratory data analysis where distribution of variables are included in form of plots and tables. This section is to provide preliminary insights about the data in general. The most content section is the implementation part where the data slicing and model fitting are thoroughly discussed. Finally, the last section compares these models with a difference-in-differences model used in a sophisticated empirical study [3].

2 The models

There are two appropriate and commonly used models to predict whether a loan is at high risk of defaulting, the logistic regression and the k-nearest neighbors. This section is dedicated to provide overview of these models. For readers whose background is not mathematics-rigorous, examples are also written. Readers who wish to be directed to the implementation may go to the Section 3.

Before proceeding, a brief introduction to address the nature of the problem shall be given. Unlike regression problem, for instance a stock's return, loan default prediction is defined by classification problem, in which the response variable is categorical. In this case, a loan either defaults or does not. Suppose loan status is denoted by Y, then $Y \in \{\text{Default}, \text{No Default}\}$. The explanatory variables are denoted by $\boldsymbol{X} = X_1, X_2, ..., X_k$. The goal is to predict Y given that $\boldsymbol{X}$ is fixed. A model is favorable when the misclassification rate is lower, that is higher accuracy.

2.1 Logistic Regression

Suppose Y is a Bernoulli response, that is it can only take 0 and 1, and one explanatory variable is given, denoted as X. It is common to denote

$$P(Y = 1 \mid X = x) = \pi(x) \in (0, 1) \text{ for } x \in \mathbb{R}$$

That is, $\pi(x)$ is the probability, taking value between 0 and 1, given a fixed value of x such that x can be all real numbers. The distribution of $\pi(x)$ is described as

$$\pi(x) = \frac{e^{\alpha+\beta x}}{1 + e^{\alpha+\beta x}}$$

The logit function of x is defined as $\ln \frac{x}{1-x}$, so

$$\text{logit}(\pi(x)) = \ln \left(\frac{\pi(x)}{1 - \pi(x)}\right) = \alpha + \beta x \tag{1}$$

To find α and β, data must be fitted to this model so these coefficients can have values. The later example will make this more clear. Now, the interpretation of these coefficients is as follows. β governs the relationship of X and Y. α by itself is the log-odds when $x = 0$.

- $\beta > 0$: the probability that $Y = 1$ increases as x increases
- $\beta = 0$: x no relationship between X and Y
- $\beta < 0$: the probability that $Y = 1$ increases as x decreases

Example 1 The original horseshoe crab data originates from [2]. In this data, the number of male satellites that are attached to a female crab is denoted as satell. For simplicity, the variable y is added to indicate whether a female crab brings at least one satellite. y shall be the response variable here, and x is the width.

```
    color spine width satell weight y
1       3     3  28.3       8   3050 1
2       4     3  22.5       0   1550 0
3       2     1  26.0       9   2300 1
4       4     3  24.8       0   2100 0
5       4     3  26.0       4   2600 1
6       3     3  23.8       0   2100 0
```

```
Call:
glm(formula = y ~ width, family = binomial, data = horseshoe)

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -12.3508     2.6287  -4.698 2.62e-06 ***
width         0.4972     0.1017   4.887 1.02e-06 ***
```

It appears that whether a female crab is attached by at least one male satellite can be described by the width of that specimen (the female). As a predictor, the width is significant at any reasonable significance level. Figure 1 shows how logistic regression fits to the data. Note that the y-axis does not indicate the number of satellites, it instead only denotes the probability of having at least one satellite. For example, given that the width of a female crab is 25, there is 50% chance that it has at least one satellite.

2.2 K-Nearest Neighbors

The k-nn algorithm essentially predicts the outcome by taking into account the k nearest neighbors to x. The mathematical setup for this model is rather simple as k-nn algorithm is one of the most straightforward supervised learning algorithm. One must first choose a favorable distance measurement, of which many people commonly choose the Euclidian distance.

$$d(x, x^*) = \sqrt{(x_1 - x_1^*)^2 + (x_2 - x_2^*)^2 + ... + (x_n - x_n^*)^2}$$

for an n-dimensional space. After choosing a value for x, the chronology of the algorithm is shown as below.

1. Compute all $d(x, x^*)$ between x and x^* where x^* is an observed point. If there are 100 points in the training set, then the product of this step is 100 distances.

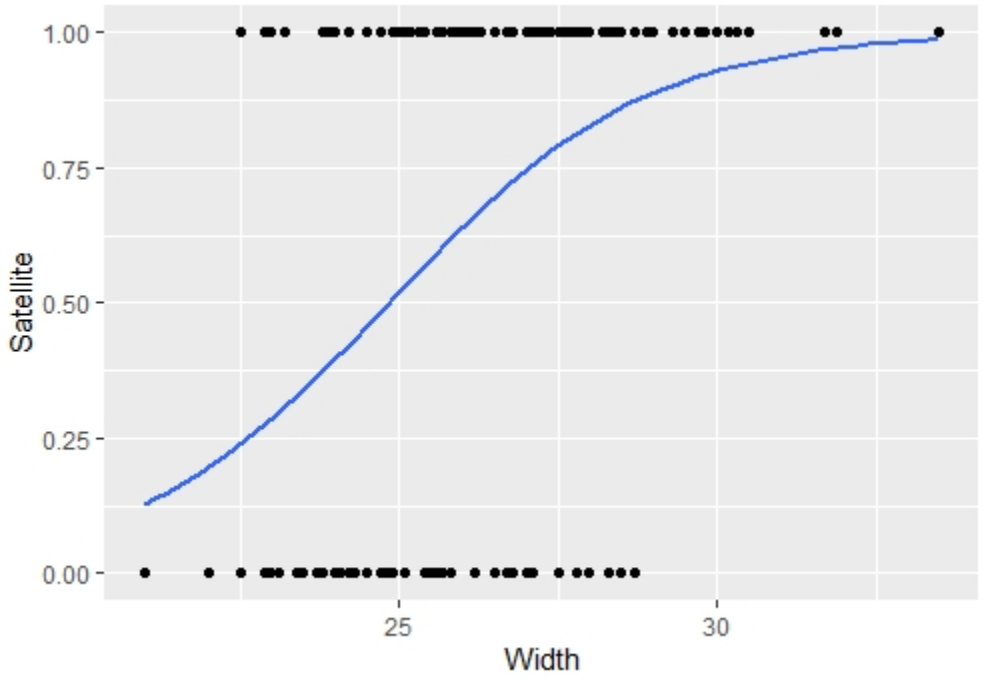


Figure 1: Visual representation of logistic regression

2. Choose k points of which the distance is the smallest, i.e. closest to x, and this set is called $\mathcal{A}$.

3. Compute Equation (2) for all possible values of Y, i.e. for all classes.

4. The class with the largest probability is chosen as the prediction for that x.

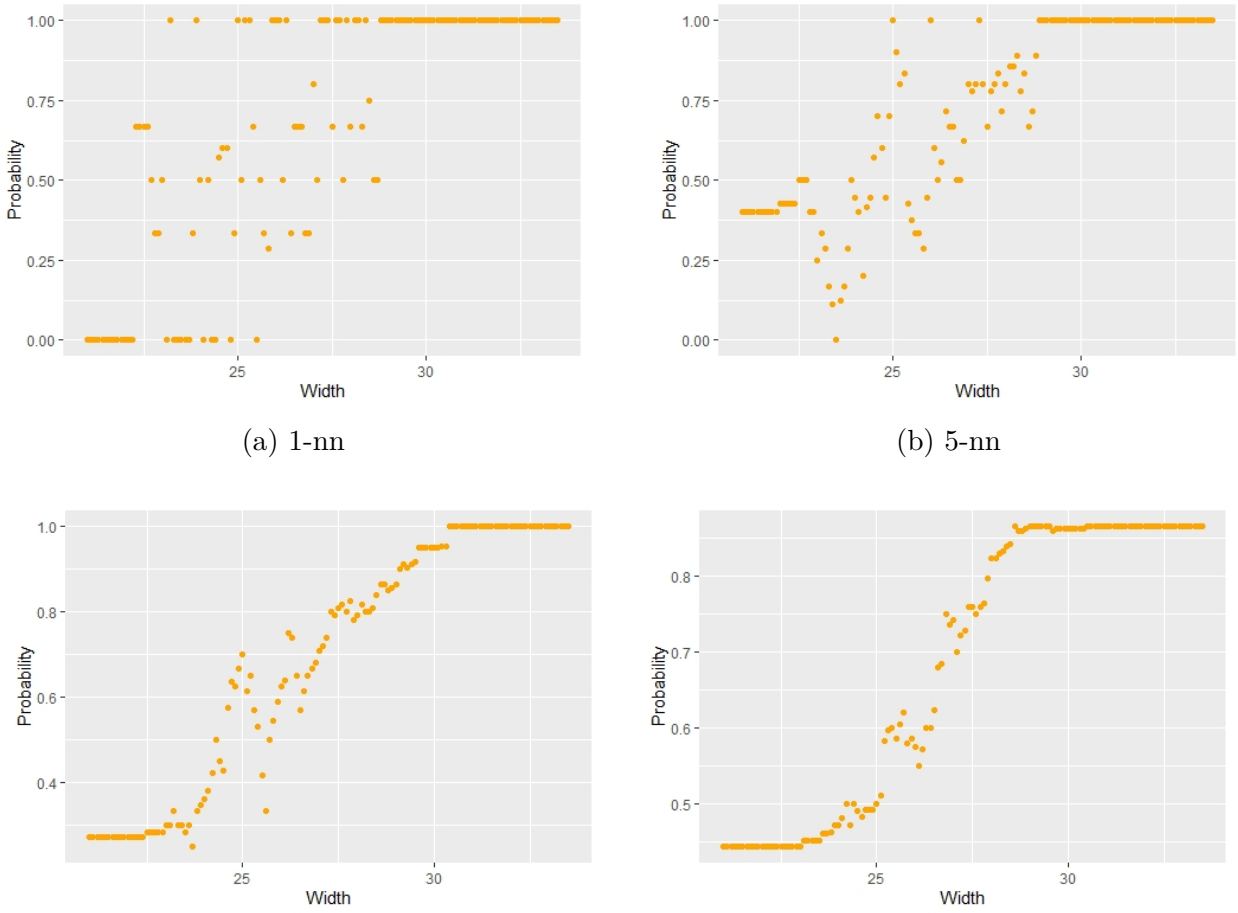
$$P(Y = j \mid X = x) = \frac{1}{K} \sum_{i \in \mathcal{A}} I(Y^{(i)} = j) \tag{2}$$

Example 2 The horseshoe crab data is used again in this example, so readers can examine how this model performs compared to logistic regression. There are many ways to conduct k-nn in R. This example uses *knn3* function in the *caret* package.

```
> library(caret)
> horseshoe.5nn = knn3(y ~ width, data = horseshoe, k = 5)
```

A 5-nn model is fitted to the data.

```
> predict(horseshoe.5nn, horseshoe, type = c("prob"))
              0         1
  [1,] 0.1111111 0.8888889
  [2,] 0.5000000 0.5000000
  [3,] 0.0000000 1.0000000
  [4,] 0.5555556 0.4444444
  [5,] 0.0000000 1.0000000
  [6,] 0.7142857 0.2857143
> head(horseshoe$width, n = 6)
[1] 28.3 22.5 26.0 24.8 26.0 23.8
```



(a) 1-nn

(b) 5-nn

(c) 20-nn

(d) 50-nn

Figure 2: The k-nearest neighbor models with k = {1, 5, 20, 50}

According to the 5-nn model, a female crab whose width is 28.3 has 89% chance of having at least one satellite. Hence, it is assumed that this female crab has at least one satellite. One can visualize the k-nn model and directly compares with the logistic regression. In this example, there are four models with different k for each, that is 1, 5, 20, and 50. Figure 2 shows the prediction of these models. Note that the 50-nn model shown in Figure 2d resembles the logistic regression.

Since k is a tuning parameter, a question that typically arises is which k value one should choose. In general, large k puts the model at a risk of underfitting. An underfitted model is bias at prediction. Conversely, small k nearest neighbor model usually overfits train data, hence it performs poorly on new data. The scientific method is to split the data into train and test set. Multiple models with k ranging from a small to a large value are fitted to the train set. Then, these models are tested using the test set. The model that predicts most accurately should be chosen.

2.3 Model Selection

Dealing with high dimension data may lead to redundancy and irrelevance. There are many methods to eliminate predictors that are not relevant to the response variable as well as those that behave similarly as another predictor.

2.3.1 Akaike Information Criterion

The AIC is defined as

$$AIC = 2k - 2\ln \hat{L} \tag{3}$$

such that k is the number of estimated parameters in the model and $\hat{L}$ is the maximum value of the likelihood function of the model. AIC is deemed better to have a smaller value. Hence, the first term is to penalize model complexity, and the second term is to reward for goodness-of-fit.

2.3.2 Lasso Regression

This method's main goal is to minimize a particular loss function that is defined as follows.

$$L_{\text{lasso}}(\beta) = SS(\beta) + \lambda_1 \sum_{j=2}^{k} \mid \beta_j \mid \tag{4}$$

The first term is sum of squares which is the sum of distance between observation y_i and prediction $\hat{y}$ squared. The second term is the penalty of complexity. λ_1 is the penalty parameter, which is the target variable that can be achieved using the *glmnet* package. The lasso regression specifically uses the scaled sum of absolute values of β_j where $j = 2, 3, ..., k$ as $j = 1$ is the intercept parameter.

Also, the following formal mathematical definition of $\hat{\beta}_{\text{lasso}}$ is important.

$$\hat{\beta}_{\text{lasso}} = \underset{\beta}{\operatorname{argmin}}\, L_{\text{lasso}}(\beta)$$

2.3.3 Ridge Regression

The loss function to be minimized is as follows.

$$L_{\text{ridge}}(\beta) = SS(\beta) + \lambda_2 \sum_{j=2}^{k} \beta_j^2 \tag{5}$$

2.3.4 Elastic Net Regression

This combines the penalty terms of lasso and ridge regressions.

$$L_{\text{elastic net}}(\beta) = SS(\beta) + \lambda_1 \sum_{j=2}^{k} \mid \beta_j \mid + \lambda_2 \sum_{j=2}^{k} \beta_j^2 \tag{6}$$

3 Exploratory Data Analysis

This section is dedicated to data wrangling. The data initially consists of 2,260,668 rows and 96 variables. There are 12 categorical variables and 84 numerical variables.

3.1 Missing Values

Figure 3 shows 20 variables with highest number of missing values. The trade-off of eliminating the observations with missing values is the loss of possibly important information held by other columns / variables. For example, the missing values of the *mths_since_last_record* variable accounts for more than 84% of the whole dataset. Removing rows with missing value of this variable would cause greater loss of information held by the *int_rate*, which may be more important than the former.

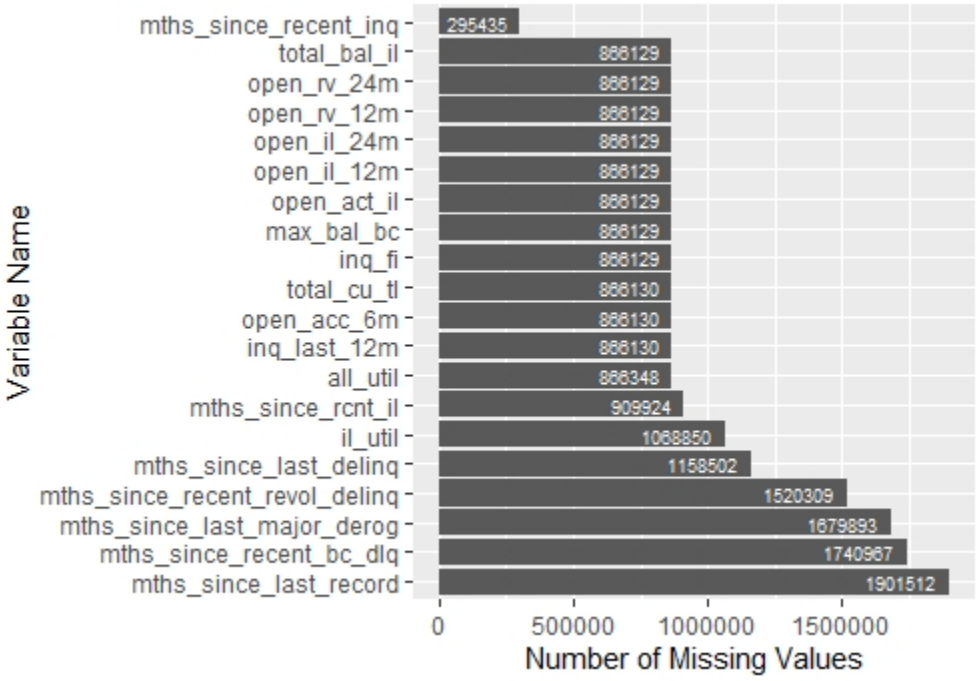


Figure 3: Variables with the highest number of missing values

Variables with more than 800,000 missing values are eliminated from the analysis. The observations with the remaining missing values are omitted from the data. The cleaned data now consists of 1,674,260 rows and 77 columns, of which 12 are categorical and 65 are numerical.

3.2 Loan-descriptive Variables

There are certain variables that describe any given loan, such as interest rate, installment amount, loan amount, etc. This subsection provides visualization of these variables so that readers acknowledge the distribution and characteristics of the loan.

3.2.1 Loan Status

Loan status is the response variable for the rest of this paper. It appears that loans whose status does not meet the credit policy were eliminated during the cleaning process. This might indicate the reason of missing values on some variables. Hence, the loan status went from 9 to 7 variations. Figure 4 shows the distribution.

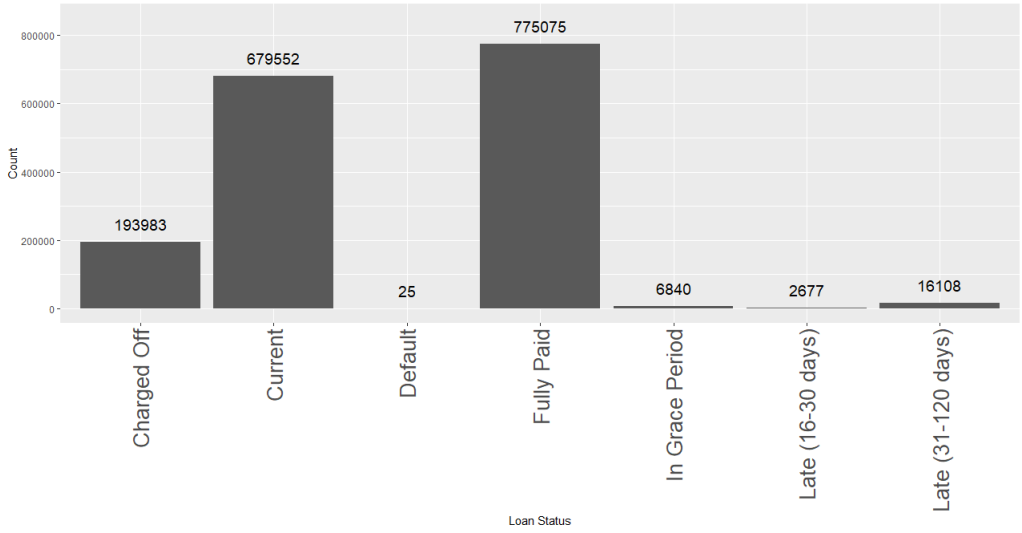


Figure 4: Distribution of loan status

The logistic regression is only adequate for binary response, that is either defaulted or non-defaulted in this case. Hence, defaulted loans consist of charged-off, default, and late loans, while the remainders are deemed as non-defaulted loans. This can easily be done using the following R command.

```
> data.clean$loan_status.b = ifelse(data.clean$loan_status %in% c("Current", "
    Fully Paid", "In Grace Period"), "Non-Defaulted", "Defaulted")
```

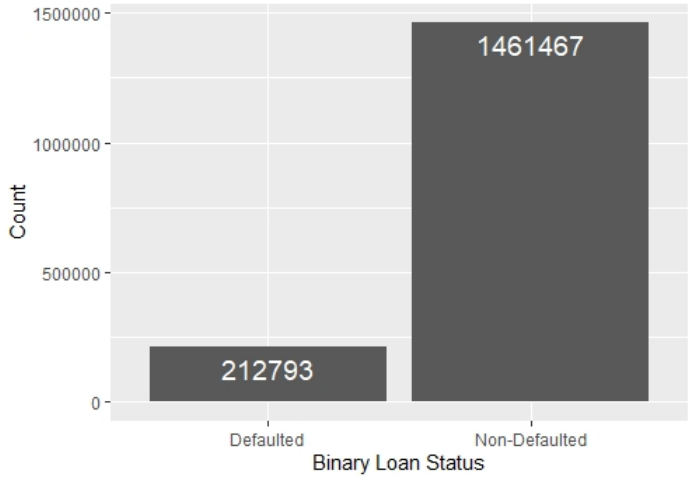


Figure 5: Distribution of binary loan status

Shown in Figure 5, this is the new distribution.

3.2.2 Loan Amount

Figure 6 shows the distribution of the loan amount. The mean of loan amount is $15,387 with median of $13,650.

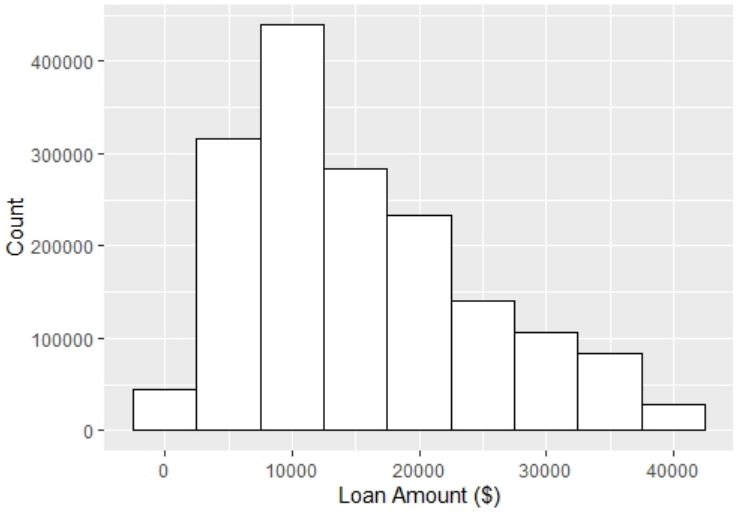


Figure 6: Loan amount distribution

3.2.3 Interest Rate

Higher interest rate is usually associated with higher risk of default. Understanding its distribution may be necessary as shown in Figure 7. The mean is 13.20% and the median is 12.69%.

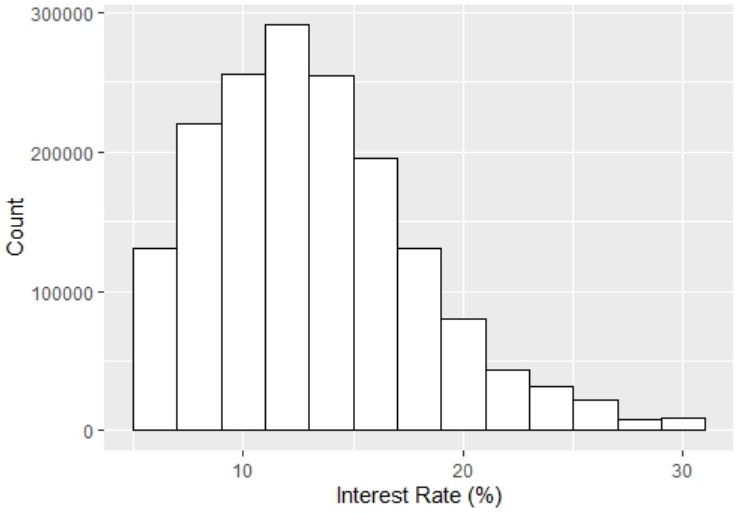


Figure 7: Interest rate distribution

3.2.4 Loan Grade

The company, Lending Club, assigns grades to their outstanding loans. A is the best, and G is the worst, in term of the likelihood of being fully repaid. As shown in Figure 8, the distribution resembles with that of the interest rate. This strengthen the previous argument that higher interest rate is correlated with higher default risk. Also, the relationship between loan grade and interest rate is shown by the scatter-plot of all loans in Figure 9.

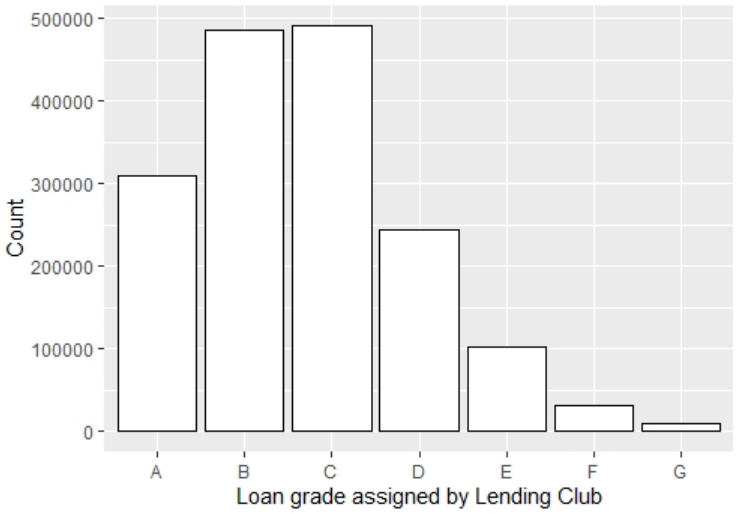


Figure 8: Loan grade distribution

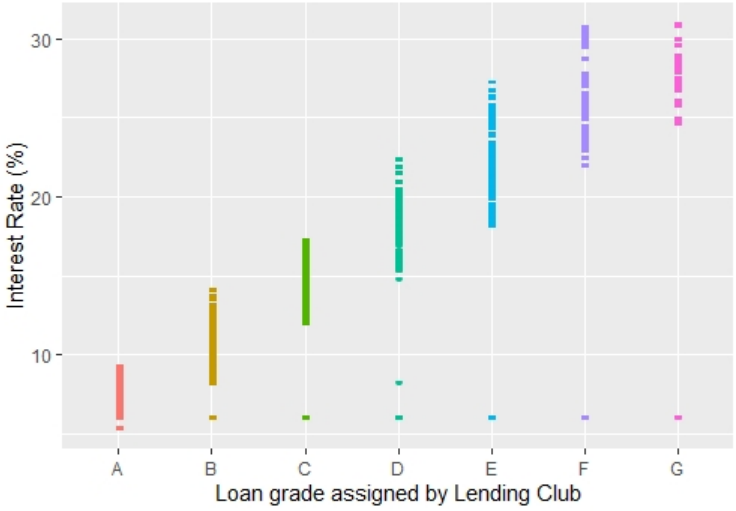


Figure 9: Relationship between grade and interest rate

Loans with higher grades are much less likely to default than those assigned low grades. The distribution of loan grade by loan status is shown in Table 1. Table 2 also shows that mean of interest rate does not differ significantly between loan status.

Loan Grade	Defaulted (%)	Non-Defaulted (%)
A	4.85	20.44
B	18.87	30.52
C	32.37	28.85
D	22.84	13.34
E	13.71	5.05
F	5.55	1.40
G	1.81	0.39

Table 1: The distribution of grade by loan status

Loan Grade	Defaulted (%)	Non-Defaulted (%)
A	7.39	7.08
B	10.77	10.66
C	14.11	14.16
D	17.88	18.25
E	21.31	22.11
F	25.28	25.80
G	28.16	28.52

Table 2: Mean of interest rate by loan grade and status

3.2.5 Purpose

The purpose of loans also plays an important role in determining whether a loan is likely to be defaulted. Figure 10 shows the distribution of loan purpose.

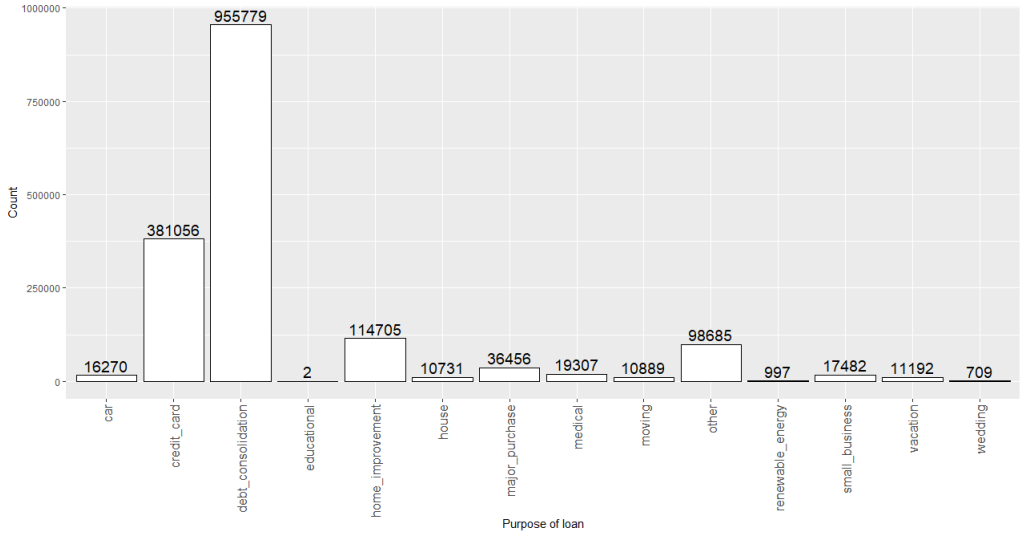


Figure 10: Distribution of loan purpose

Shown in Table 3, the defaulted loans are comprised of 62% debt consolidation loans while it is only 56% for non-defaulted loans. Larger percentage are held by non-defaulted

Purpose	Defaulted (%)	Non-Defaulted (%)
car	0.71	1.01
credit_card	18.47	23.38
debt_consolidation	62.03	56.37
educational	0.00	0.00
home_improvement	5.92	6.99
house	0.61	0.64
major_purchase	2.04	2.20
medical	1.18	1.15
moving	0.78	0.63
other	5.90	5.89
renewable_energy	0.07	0.06
small_business	1.58	0.97
vacation	0.64	0.67
wedding	0.05	0.04

Table 3: The distribution of loan purpose by loan status

loans for credit card and home improvement payments. Only two borrowers are applying for educational purpose, hence these rows will be omitted.

3.2.6 Verification Status

This variable indicates if income was verified by Lending Club, not verified, or if the income source was verified. The distribution is shown in Figure 11. Table 4 shows that non-verified defaulted loans accounts for smaller percentage than non-verified non-defaulted loans. On the contrary, there are more verified defaulted loans than verified non-defaulted, in term of the share of the corresponding loan status.

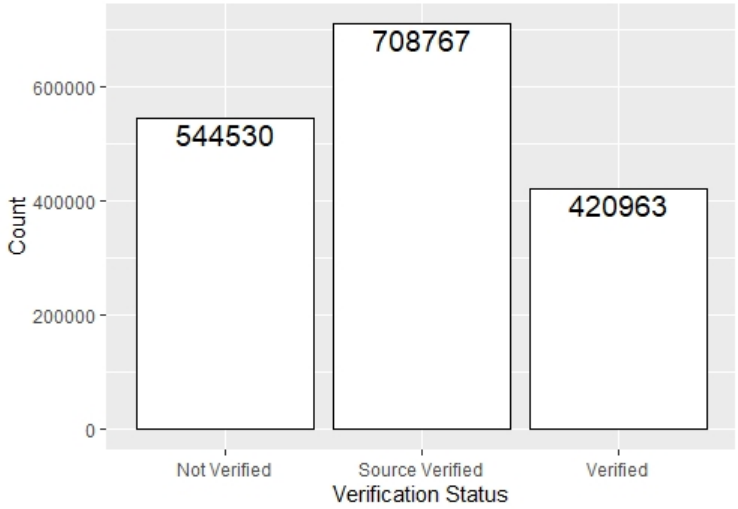


Figure 11: Distribution of loan's verification status

Verif. Status	Defaulted (%)	Non-Defaulted (%)
Not Verified	21.93	34.07
Source Verified	44.87	41.96
Verified	33.20	23.97

Table 4: The distribution of verification status by loan status

3.2.7 Application Type

This variable indicates whether the loan is an individual application or a joint application with two co-borrowers. There are 1,595,848 individual and 78,412 joint applications in the cleaned data. Table 5 shows the distribution of loan status by application type. Note that this table is based on the row sums. The finding is that there is larger tendency that individual application ends up defaulted than joint application. This may make sense because joint application is backed by two source of income.

App. Type	Defaulted (%)	Non-Defaulted (%)
Individual	13.01	86.99
Joint App	6.64	93.36

Table 5: Distribution of loan status by application type

Figure 12 and 13 are very helpful for examining the difference between the two application types. Joint application tends to request for larger amount of funding, hence the interest rate being charged is also higher than individual applications. Hence, it may be more beneficial for lending companies to fund joint application as they also tend to pay back their liabilities.

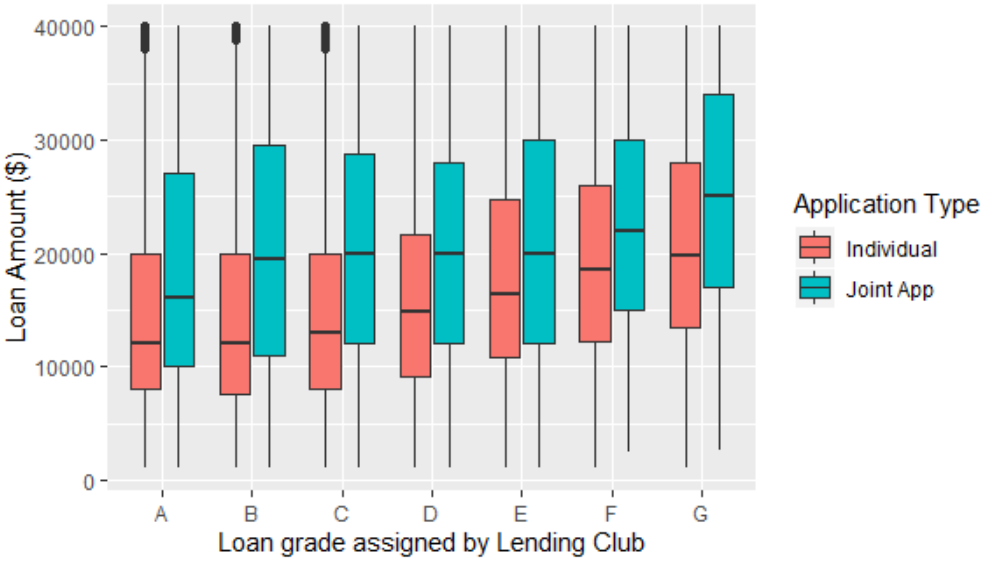


Figure 12: Distribution of loan amount versus loan grade by application type

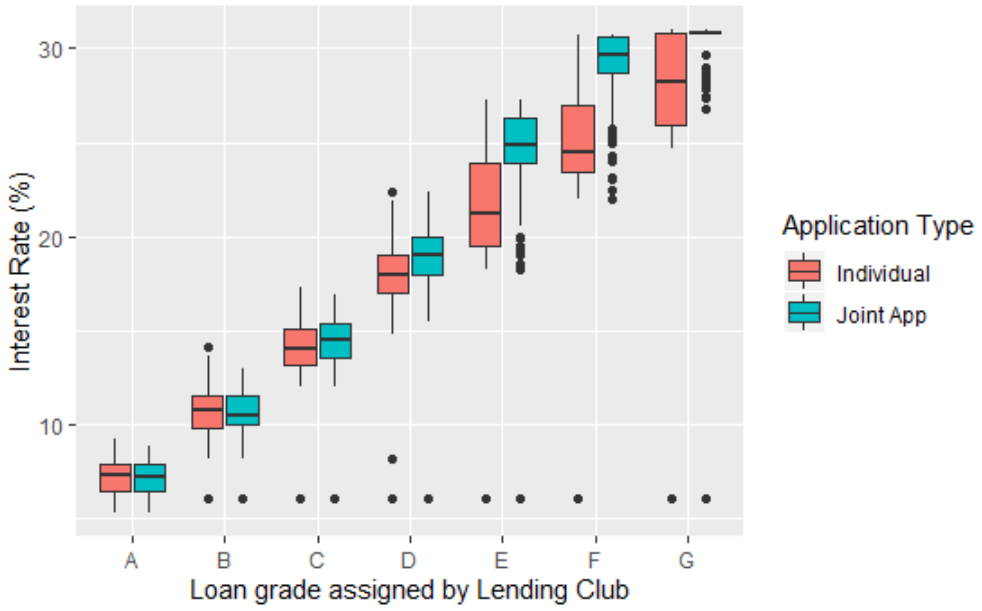


Figure 13: Distribution of interest rate versus loan grade by application type

3.2.8 Employment Length

The distribution of loan status by employment length is shown in Table 6. The finding is that employment length seems not to have significant relationship with loan status. The default rate among different employment length averages at about 13 percent.

Employment Length	Defaulted (%)	Non-Defaulted (%)
< 1 year	12.90	87.10
1 year	13.40	86.60
2 years	12.90	87.10
3 years	13.00	87.00
4 years	12.60	87.40
5 years	12.80	87.20
6 years	12.80	87.20
7 years	13.40	86.60
8 years	14.20	85.80
9 years	13.70	86.30
10+ years	12.00	88.00

Table 6: Distribution of loan status by employment length

3.2.9 Term

Lending Club offers two options for the loan term, 36-month and 60-month, which are commonly regarded as short-term and long-term loans, respectively. There are 1,171,522 short-term and 502,738 long-term loans in the cleaned data. Table 7 shows the distribution

of loan status by loan term. About 1 in 10 short-term loans ends up defaulting, while it is about 1 in 6 for long-term loans. It is very apparent that loan term plays an important role in predicting the probability of default.

Loan Term	Defaulted (%)	Non-Defaulted (%)
36 months	10.61	89.39
60 months	17.61	82.39

Table 7: Distribution of loan status by loan term

3.2.10 Home Ownership

In the cleaned data, 51.35% borrowers pay for mortgage, 38.13% rent, 10.47% own, and the rest, less than 0.05%, are ambiguous.

Home Ownership	Defaulted (%)	Non-Defaulted (%)
Mortgage	44.24	52.38
Own	10.49	10.47
Rent	45.23	37.10

Table 8: Distribution of home ownership by loan status

Borrowers who mortgage tend to repay their loans, while those who rent have smaller tendency to do so. The distribution is shown in Table 8.

4 Implementation

This section is dedicated to provide the step-by-step process done to obtain the concluded models.

4.1 Data Slicing

The dataset provided is substantial as it contains over 2 millions of observations with 96 variables. Using household computers, it is nearly impossible to fit any model to the whole data. Hence, partitioning the data to much smaller sets is necessary. Using the *caret* package, one can use the *createDataPartition* function to easily make smaller partitions. The arguments that one can adjust with this function are how small the partition is desired and how many times the partitioning should be repeated. In this example, due to computational limitation, the smallest partition used to fit the preliminary models consists of only 1,172 rows.

```
> library(caret)

> set.seed(5396) #for reproducibility
> train_idx <- createDataPartition(y = data.clean.d$loan_status.b, p = 0.7, list
    = FALSE)
> train <- data.clean.d[train_idx,]
```

```
> test <- data.clean.d[-train_idx,]

#as train data still consists of 1,171,456 rows, we need to partition more so we
    can compute faster
> train_idx_more <- createDataPartition(y = train$loan_status.b, p = 0.001, times
     = 10)
```

How does one know that these partitions are fair? One way to ensure that this representation of the whole data is fair is to check the mean of the response variable.

```
> train_more1 <- train[train_idx_more[[1]],]
> x1 <- model.matrix(loan_status.b ~ ., train_more1)[,-1]
> y1 <- train_more1$loan_status.b
> summary(y1)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 0.0000  0.0000  0.0000  0.1331  0.0000  1.0000
```

The mean of default rate is still about 13%, which should provide evidence that this chunk of the whole data is sufficient. Also, there are numerous near-zero-variance predictors and these should be removed to avoid collinearity, which can easily be done using *nearZeroVar* function in *caret* package.

4.2 Predictive Power of Various Models

Due to the high-dimensional nature of the dataset, eliminating redundant and irrelevant features is necessary as more features costs more computational resources. Even after trimming and cleaning, there are 93 variables. Chance is that many of them are not necessary to be included.

4.2.1 Penalized Regression

The penalized estimations include lasso, ridge, and elastic net regressions. These methods penalized redundant and irrelevant variables. Using *glmnet* library, penalized regression can be easily done even with the built-in cross-validation feature.

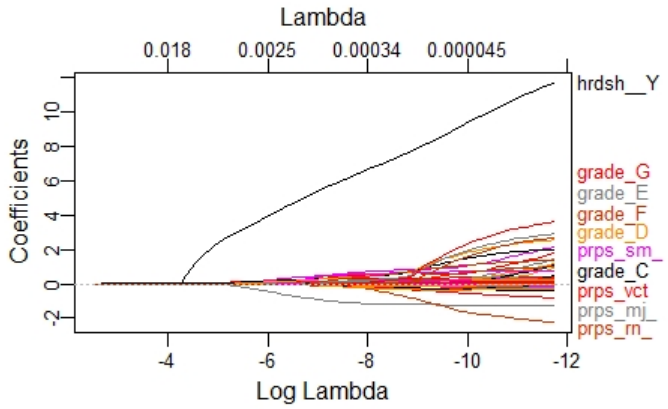


Figure 14: Penalization using Lasso Estimation

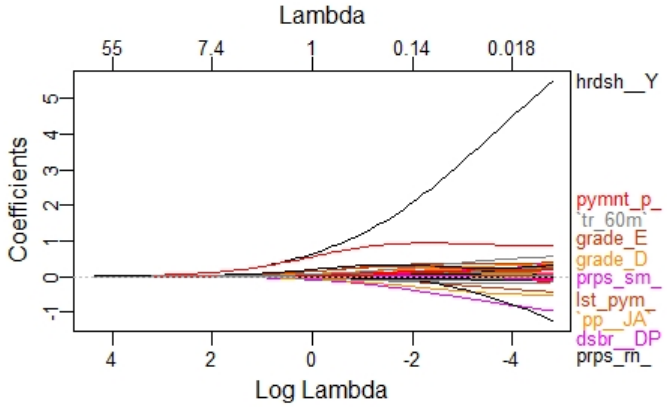


Figure 15: Penalization using Ridge Estimation

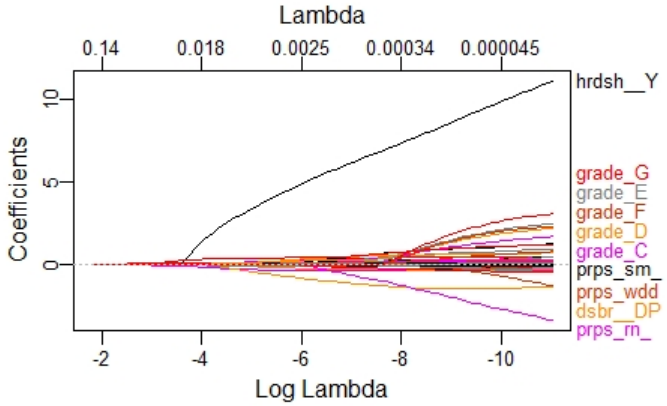


Figure 16: Penalization using Elastic Net Estimation

Figure 14, 15, and 16 show that hardship flag, loan grades, and purposes are vital in predicting the probability of default. To ensure that one chooses the right value for λ, cross-validation needs to be done. The λ achieved after cross-validation is then used to fit a penalized regression by supplying this value as an argument. Then, the test accuracy achieved by these three penalized regressions is shown in Table 9.

4.2.2 AIC Stepwise Regression

This method initially considers the model with all variables as the full model and the model with only intercept as the null model. An argument that can be adjusted is whether to do perform stepwise selection backward, forward, or both. After running the algorithm that takes a while for a train set of over 11,000 observations, there are two chosen models. The

Metrics	Lasso Regression	Ridge Regression	Elastic Net Regression
Test Accuracy	98.69	96.43	98.65
95% CI	(98.33, 98.98)	(95.88, 96.93)	(98.29, 98.95)
Sensitivity	98.57	96.22	98.53
Specificity	100	100	100

Table 9: Test accuracy of penalized regressions

diagnostic and accuracy of these models are shown in Table 10.

Table 10: Model diagnostics and test accuracy for stepwise regression models

Metrics	Stepwise with *purpose*	Stepwise without *purpose*
Test Accuracy	98.65	98.65
95% CI	(98.29, 98.95)	(98.29, 98.95)
Sensitivity	98.54	98.54
Specificity	99.76	99.76
Deviance (df)	1063.8 (11691)	1054.4 (11679)
AIC	1111.8	1126.4

The takeaway here is that the stepwise regression performs as excellent as the penalized regressions. The variable *purpose* does not make much difference although it lowers the AIC, but the test metrics are exactly the same. All in all, these methods that have been attempted by far are adequate at predicting defaulted and non-defaulted cases.

4.2.3 KNN

For KNN, the same features as selected by the stepwise regression are used as using the full model is an overkill.

Table 11: Model diagnostics and test accuracy for KNN models

Metrics	KNN with *purpose*	KNN without *purpose*
Test Accuracy	92.85	92.17
95% CI	(92.10, 93.55)	(91.40, 92.90)
Sensitivity	99.82	99.98
Specificity	27.93	19.51
K	17	21

Both KNN models perform similarly, except the presence of the variable *purpose* increases the specificity significantly. However, these models are less accurate than penalized and stepwise regressions. This model should not be used.

4.3 Future Direction and Recommendations

Previously, the methods used to select features are rather technical, such as AIC and penalized estimation. These methods quantify each variable, assign scores, and choose accordingly. Nonetheless, another way to approach the main problem is to understand the data and the nature of each variable. For example, categorizing subsets of variables as the following list may be helpful.

- Assigned by Lending Club
  - Loan grade
  - Interest rate
- Assigned by borrowers
  - Loan amount
  - Term (36-month or 60-month)
  - Purpose (debt consolidation, credit card payment, etc.)
  - Application type (individual or joint)
- Information about the borrowers
  - Income
  - Employment length
  - Home ownership (own, mortgage, or rent)
  - Debt-to-income ratio
- Information about the borrowers' credit
  - Number of currently active revolving trades
  - Number of derogatory public records
  - Revolving line utilization rate
  - Number of inquiries in past 6 months

Such variables categorization may be more helpful to predict the likelihood of probability. For example, is the information assigned by Lending Club sufficient to predict default probability? How about the other categories? What if some of these categories are combined together? Furthermore, including FICO score may be helpful since it contains information about borrowers' tendency to pay back their liabilities on time.

Furthermore, the analysis done for this paper was performed in a four-core 8GB-memory laptop. Such technical analysis should be done in a computer with much more power in order to achieve the most accurate model.

5 Screening on Loan Terms

After penalized estimation and AIC elimination, loan term seems to be an important variable to predict the likelihood of default. This section is dedicated to provide more extensive information whether loan term is a drive for defaulted cases. This part of the paper is referenced to Hertzberg, Liberman and Paravisini (2018) [3].

5.1 Empirical Strategy and Result

Lending Club made long-term loans available for lower amount, that is between $10,000 and $16,000, in 2013. Previously, low amount loans are only lent with the 36-month option. The main objective of this study is to compare the outcomes (defaulted or non-defaulted) of borrowers who applied for the short-term loans before and after the expansion within the same risk category (same grade). Table 12 shows how the study groups loans.

Table 12: Grouping of loans within the sample period (Dec 2012 - Oct 2013)

Loan Amount ($)	**5,000 - 10,000**	**10,000 - 12,000**	**12,000 - 16,000**	**16,000 - 20,000**
Control / Treatment	Control	Control	Treatment	Control
Long-term Availability	Never	> July 2013	> March 2013	Always
Short-term Availability	Always	Always	Always	Always
Impact	Unaffected	Affected	Affected	Unaffected

The study also made many assumptions which were supported by quantitative arguments. They assume that changes in economic environment, options outside Lending Club, and how Lending Club assigns grades to borrowers do not affect borrowers opting to take long-term loans when they are available for certain amounts. These arguments were emphasized to assure that the strategy only relies on comparison of the amount of short-term loans before and after the expansion.

The study uses difference-in-differences regression by collapsing the original data to count data. N_{jkt} denotes the number of loans originated at month t as the risk category (grade) j and loan amount bin k where loan amount is categorized starting from $10,000 with increment of $1,000. Then a dummy variable is created to indicate short-term loans after expansion dates (depending on the amount) and denoted as D_{kt}. The regression function to predict N_{jkt} is

$$\log(N_{jkt}) = \beta_k^{'} + \delta_{jt}^{'} + \gamma^{'} D_{kt} + \epsilon_{jkt}$$

The result is that the number of short-term loans is 14.51% lower when the expansion occurred. Then, based on this finding, the study is one step closer to the main goal. Instead of regression, the next step is to perform difference-in-differences classification which is aimed to predict the outcome.

$$\text{Default}_i = \beta_i^{1000\text{bin}} + \delta_i^{jt} + \gamma D_i + X_i + \epsilon_i$$

γ is the coefficient of interest as it measures the change of default rate of 36-month loans after the expansion relative to the change of default rate of unaffected loans. The result is that borrowers who take the short-term loans when long-term loans are available for the same amount are 0.8% less likely to default than short-term loans when there is no long-term option.

These significant results do not perfectly align with the methodologies done in the previous sections. The study initially claims the loan amount bin as a predictor, but cross-validated penalized regressions penalize the continuous loan amount variable quite highly and its coefficients are nearly zero for lasso, ridge, and elastic net regressions. Also, for the stepwise regression, loan amount is eliminated. The risk category, which is denoted as j in the study, is used in the penalized regression, but not in the stepwise regression. The issued date information was omitted in the data provided to perform the analysis in the previous sections.

5.2 Implementation of logistic regression with the same approach

Since this paper does not primarily focus on difference-in-differences methods, this subsection is to provide with results of implementing the previous methodologies using the same strategy. For this part, the issued date data is given, but the FICO score is not given. As a substitute for FICO, the grade is assumed to contain information that the FICO score is based on.

Following the same scheme described by the study, the logistic regression attempted to be fit to the train data is as follows.

$$Y_i = \beta_1 x_i^{grade} x_i^{\text{month of origination}} + \beta_2 x_i^{bin} + \beta_3 D_i$$

The summary of this model shows that β_3, which is the coefficient of interest is -0.07133. This means that holding other variables constant, the odds that borrowers who take short-term loans for the amounts that offer long-term option to default is $e^{-0.07133} = 0.9312$ times relative to those who opt for short-term loans without long-term option. Although the estimate is not significant, but it indicates some resemblance with the study's finding.

Table 13: The rest of the estimates are omitted for brevity

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	-2.2848	1.0620	-2.15	0.0315
D	-0.0713	0.0526	-1.36	0.1749
bin2	0.1539	0.0736	2.09	0.0365
bin3	0.2643	0.0616	4.29	0.0000
bin4	0.1931	0.0706	2.74	0.0062

However, this model is definitely not adequate to predict default probability as the test accuracy is only 43.55% with 95% confidence interval of (42.84%, 44.26%). Specificity is as low as 15.48%, and sensitivity is 92.31%, which is biased due to the large number of non-defaulted cases.

6 Conclusion

The original data has more than 2 million observations with 96 variables. The first phase to process this data should be to explore the distribution of many variables. Variables can be categorized as borrower's characteristics, their historical credit performance, the loan's characteristics, and variables assigned by Lending Club. Using all 96 variables to predict the probability of default is an overkill and not recommended. It takes exponentially large computational resource and may deliver inaccurate results.

The methods used in this paper to reduce dimensions is penalized regression and stepwise variable selection based on the Akaike Information Criterion. Another classification algorithm that is common to predict binary outcomes is the k-nearest neighbor.F For the first two, the full model is initially considered, and the former penalized redundant and irrelevant variables while the latter computes the AIC and choose the model with the lower AIC iteratively.

Three penalized regression methods are used, which are lasso, ridge, and elastic net regression. Each of these yield similar predictive power with over 95% test accuracy and sensitivity and amazingly 100% specificity. Stepwise regressions rank the second as their specificity cannot achieve 100%, but they perform quite similarly as the penalized regressions.

While for the non-parametric k-nearest neighbor, methods to choose which variable to use are beyond the scope of this paper. It was decided to use the same variables as the stepwise regression, and the result is not satisfactory. Although the test accuracy reaches 92%, the specificity is about 28% at best. Plus, one of the drawbacks of fitting KNN to such large data is it takes substantial resources.

This paper also refers to a study done by Hertzberg, Liberman and Paravisini (2018) [3]. The study claims that borrowers who choose short-term loans when a long-term option for that amount is available tend to be less likely to default relative to those who opt for short-term loans with no long-term option. The study relies on difference-in-differences regression. This paper reproduces the approach using logistic regression and finds that the odds of the former group are indeed less likely to default than the latter.

References

[1] Agresti, A. (2012). An Introduction to Categorical Data Analysis, 3rd edition. Wiley: New Jersey.

[2] Brockmann, H. J. (1996). Satellite male groups in horseshoe crabs. Limulus polyphemus. Ethology, 102: 1-21.

[3] Hertzberg, A., Liberman, A., and Paravisini, D. (2018) Screening on Loan Terms: Evidence fromMaturity Choice in Consumer Credit.

[4] Linders, D. (2019) Predictive Analytics. Topic 7: Penalized Estimation, Course slides for ASRM 552.

Appendices

A Cross-Validation for Penalized Estimation

Table 14: Coefficients from training partition with 11,000 observations

Variables	Lasso	Ridge	Elastic Net
(Intercept)	-5.611	-2.549	-4.409
total_il_high_credit_limit	0.000	-0.000	0.000
total_bc_limit	0.000	-0.000	0.000
total_bal_ex_mort	-0.000	-0.000	-0.000
tot_hi_cred_lim	-0.000	-0.000	-0.000
pub_rec_bankruptcies	-0.385	-0.143	-0.336
percent_bc_gt_75	-0.001	0.002	0.000
num_tl_op_past_12m	0.091	0.009	0.073
num_sats	-0.037	-0.003	-0.013
num_rev_tl_bal_gt_0	-0.055	0.007	0.000
num_rev_accts	-0.031	-0.001	-0.039
num_op_rev_tl	0.028	-0.001	0.021
num_il_tl	0.031	0.008	0.019
num_bc_tl	0.060	0.016	0.067
num_bc_sats	0.051	0.017	0.069
num_actv_rev_tl	0.091	0.007	0.064
num_actv_bc_tl	-0.164	-0.008	-0.216
num_accts_ever_120_pd	-0.038	0.006	-0.049
mths_since_recent_inq	0.015	-0.010	0.008
mths_since_recent_bc	0.002	-0.000	0.001
mort_acc	0.043	0.001	0.001
mo_sin_rcnt_tl	0.005	-0.001	0.007
mo_sin_rcnt_rev_tl_op	-0.002	-0.003	-0.002
mo_sin_old_rev_tl_op	-0.003	-0.000	-0.003
mo_sin_old_il_acct	0.001	0.000	0.001
bc_util	-0.000	0.001	0.002
bc_open_to_buy	-0.000	-0.000	-0.000
avg_cur_bal	-0.000	-0.000	-0.000
acc_open_past_24mths	0.019	0.035	0.027
total_rev_hi_lim	0.000	-0.000	0.000
tot_cur_bal	0.000	-0.000	0.000
last_credit_pull_d	0.383	0.204	0.456
last_pymnt_amnt	-0.001	-0.000	-0.001
last_pymnt_d	-0.338	-0.430	-0.328
total_rec_int	0.000	0.000	0.001
total_rec_prncp	-0.008	-0.000	-0.002

Continued on next page

Table 14 – *Continued from previous page*

Variables	Lasso	Ridge	Elastic Net
total_pymnt_inv	0.000	-0.000	-0.001
total_pymnt	0.000	-0.000	-0.001
out_prncp_inv	-0.007	-0.000	-0.002
out_prncp	0.000	-0.000	-0.001
total_acc	-0.013	0.002	0.000
revol_util	0.004	0.003	0.003
pub_rec	0.134	0.103	0.167
open_acc	0.000	-0.006	-0.030
inq_last_6mths	0.068	0.019	0.079
earliest_cr_line	0.039	-0.005	0.028
delinq_2yrs	0.061	0.063	0.094
dti	0.010	0.017	0.013
annual_inc	0.000	0.000	-0.000
installment	0.010	0.002	0.010
int_rate	-0.156	0.000	-0.144
funded_amnt_inv	0.007	0.000	0.001
funded_amnt	0.001	0.000	0.001
loan_amnt	0.000	0.000	0.001
disbursement_method_DirectPay	-1.237	-0.966	-1.364
hardship_flag_Y	11.706	5.494	11.086
‘application_type_Joint App‘	-0.231	-0.511	-0.293
initial_list_status_w	0.111	-0.165	0.064
purpose_credit_card	1.083	-0.049	0.215
purpose_debt_consolidation	1.077	0.074	0.255
purpose_educational	0.000	0.000	0.000
purpose_home_improvement	1.151	0.082	0.476
purpose_house	0.896	0.184	0.238
purpose_major_purchase	1.412	0.230	0.474
purpose_medical	0.490	-0.072	-0.082
purpose_moving	1.057	0.080	0.432
purpose_other	1.081	-0.081	0.270
purpose_renewable_energy	-2.238	-1.253	-3.370
purpose_small_business	2.147	0.335	1.255
purpose_vacation	1.757	0.058	0.928
purpose_wedding	-0.030	0.120	-1.283
pymnt_plan_y	0.000	0.861	0.000
‘verification_status_Source Verified‘	0.164	0.129	0.113
verification_status_Verified	0.302	0.181	0.240
home_ownership_OWN	0.280	0.025	0.167
home_ownership_RENT	0.094	0.045	0.089
‘emp_length_1 year‘	-0.802	-0.118	-0.444
‘emp_length_2 years‘	-0.225	0.062	-0.041

Continued on next page

Table 14 – *Continued from previous page*

Variables	**Lasso**	**Ridge**	**Elastic Net**
‘emp_length_3 years‘	0.225	0.094	0.255
‘emp_length_4 years‘	0.035	-0.043	-0.011
‘emp_length_5 years‘	0.243	0.112	0.102
‘emp_length_6 years‘	-0.127	-0.012	0.022
‘emp_length_7 years‘	-0.059	0.015	-0.202
‘emp_length_8 years‘	0.769	0.292	0.694
‘emp_length_9 years‘	-0.326	-0.112	-0.410
‘emp_length_10+ years‘	0.146	-0.022	0.089
grade_B	0.985	0.042	0.778
grade_C	2.006	0.318	1.678
grade_D	2.560	0.379	2.177
grade_E	2.905	0.398	2.460
grade_F	2.693	0.232	2.290
grade_G	3.595	0.321	3.052
‘term_60 months‘	1.373	0.557	1.193

B R Source Code

```
setwd("C:/Users/wilso/Documents/2. UIUC/7. 2020 Spring/ASRM 499/Final Project")
load("LC.rda")

#########################################################
####################MODELS###############################
#########################################################

#logistic regression explanation
horseshoe = read.table("horseshoe.txt", header = TRUE)
summary(glm(y ~ width, family = "binomial", data = horseshoe))
library(ggplot2)
ggplot(horseshoe, aes(x = width, y = y)) +
        geom_point() +
        labs(x = "Width", y = "Satellite") +
        stat_smooth(method = "glm", method.args = list(family = "binomial"), se =
             FALSE)

#knn explanation
library(caret)
horseshoe.1nn = knn3(y ~ width, data = horseshoe, k = 1)
horseshoe.5nn = knn3(y ~ width, data = horseshoe, k = 5)
horseshoe.20nn = knn3(y ~ width, data = horseshoe, k = 20)
horseshoe.50nn = knn3(y ~ width, data = horseshoe, k = 50)
horseshoe.sat.prob.1nn = predict(horseshoe.1nn, data.frame(width = seq(min(
    horseshoe$width), max(horseshoe$width), by = 0.1)), type = c("prob"))[,2]
horseshoe.sat.prob.5nn = predict(horseshoe.5nn, data.frame(width = seq(min(
    horseshoe$width), max(horseshoe$width), by = 0.1)), type = c("prob"))[,2]
horseshoe.sat.prob.20nn = predict(horseshoe.20nn, data.frame(width = seq(min(
    horseshoe$width), max(horseshoe$width), by = 0.1)), type = c("prob"))[,2]
horseshoe.sat.prob.50nn = predict(horseshoe.50nn, data.frame(width = seq(min(
    horseshoe$width), max(horseshoe$width), by = 0.1)), type = c("prob"))[,2]
horseshoe.knn.pred = data.frame(width = seq(min(horseshoe$width), max(horseshoe$
    width), by = 0.1), pred.1nn = horseshoe.sat.prob.1nn, pred.5nn = horseshoe.sat
    .prob.5nn, pred.20nn = horseshoe.sat.prob.20nn, pred.50nn = horseshoe.sat.prob
    .50nn)

ggplot(horseshoe.knn.pred, aes(x = width, y = pred.1nn)) + geom_point(color = "
    orange") + labs(x = "Width", y = "Probability")
ggplot(horseshoe.knn.pred, aes(x = width, y = pred.5nn)) + geom_point(color = "
    orange") + labs(x = "Width", y = "Probability")
ggplot(horseshoe.knn.pred, aes(x = width, y = pred.20nn)) + geom_point(color = "
    orange") + labs(x = "Width", y = "Probability")
ggplot(horseshoe.knn.pred, aes(x = width, y = pred.50nn)) + geom_point(color = "
    orange") + labs(x = "Width", y = "Probability")

#checking which variables are numerical and categorical
vars = unlist(lapply(data, class))
table(vars)
vars

#########################################################
#######################EDA###############################
#########################################################

#checking missing values
vars.na.count = data.frame(na.count = sort(colSums(is.na(data)), decreasing =
    TRUE)[1:20])
ggplot(data = vars.na.count, aes(x = reorder(rownames(vars.na.count), -na.count),
     y = na.count)) +
        geom_bar(stat = "identity") +
        coord_flip() +
        geom_text(aes(label = na.count), hjust = 1.2, color = "white", size =
            2.3) +
        labs(x = "Variable Name", y = "Number of Missing Values")
```

```
#removing variables with more than 800,000 missing values
data.trim = subset(data, select = -c(mths_since_last_record, mths_since_recent_bc
    _dlq, mths_since_last_major_derog, mths_since_recent_revol_delinq, mths_since_
    last_delinq, il_util, mths_since_rcnt_il, all_util, inq_last_12m, total_cu_tl,
     open_acc_6m, inq_fi, max_bal_bc, open_rv_24m, open_rv_12m, total_bal_il, open
    _il_24m, open_il_12m, open_act_il))

options(scipen = 999)
vars.trim.na.count = data.frame(na.count = sort(colSums(is.na(data.trim)),
    decreasing = TRUE)[1:20])
ggplot(data = vars.trim.na.count, aes(x = reorder(rownames(vars.trim.na.count), -
    na.count), y = na.count)) +
        geom_bar(stat = "identity") +
        coord_flip() +
        geom_text(aes(label = na.count), hjust = 1.2, color = "white", size =
            2.3) +
        labs(x = "Variable Name", y = "Number of Missing Values")

#removing all rows with missing value in at least one variable
data.clean = data.trim[complete.cases(data.trim),]
vars.clean = unlist(lapply(data.clean, class))
table(vars.clean)

#loan status
ggplot(data.clean, aes(x = loan_status)) +
        geom_bar(aes(y = ..count..)) +
        geom_text(stat = "count", aes(label = ..count..), vjust = -1, size = 5) +
        theme(axis.text.x = element_text(angle = 90, hjust = 1, vjust = 0.5, size
            = 20)) +
        labs(x = "Loan Status", y = "Count") + ylim(0, 850000)

data.clean$loan_status.b = ifelse(data.clean$loan_status %in% c("Current", "Fully
     Paid", "In Grace Period"), "Non-Defaulted", "Defaulted")

ggplot(data.clean, aes(x = loan_status.b)) +
        geom_bar(aes(y = ..count..)) +
        geom_text(stat = "count", aes(label = ..count..), vjust = 1.5, size = 5,
            color = "white") +
        labs(x = "Binary Loan Status", y = "Count")

#loan amount
ggplot(data.clean, aes(x = loan_amnt)) +
        geom_histogram(binwidth = 5000, colour = "black", fill = "white") +
        labs(x = "Loan Amount ($)", y = "Count")
summary(data.clean$loan_amnt)

#interest rate
ggplot(data.clean, aes(x = int_rate)) +
        geom_histogram(binwidth = 2, colour = "black", fill = "white") +
        labs(x = "Interest Rate (%)", y = "Count")
summary(data.clean$int_rate)

#loan grade
ggplot(data.clean, aes(x = grade)) +
        geom_histogram(stat = "count", colour = "black", fill = "white") +
        labs(x = "Loan grade assigned by Lending Club", y = "Count")

library(scattermore)
ggplot(data.clean) +
        geom_scattermore(aes(x = grade, y = int_rate, color = grade), pointsize =
             2) +
        labs(x = "Loan grade assigned by Lending Club", y = "Interest Rate (%)")
            +
        theme(legend.position = "none")
table2 = table(data.clean$grade, data.clean$loan_status.b)
round(prop.table(table2, 2) * 100, 2)
table3 = aggregate(int_rate ~ loan_status.b + grade, data = data.clean, mean)
xtabs(int_rate ~ grade + loan_status.b, data = table3)
```

```
#purpose
ggplot(data.clean, aes(x = purpose)) +
        geom_histogram(stat = "count", colour = "black", fill = "white") +
        geom_text(stat = "count", aes(label = ..count..), vjust = -0.3, size = 5)
            +
        theme(axis.text.x = element_text(angle = 90, hjust = 1, vjust = 0.5, size
            = 12)) +
        labs(x = "Purpose of loan", y = "Count")
table1 = table(data.clean$purpose, data.clean$loan_status.b)
round(prop.table(table1, 2) * 100, 2)

#verification status
ggplot(data.clean, aes(x = verification_status)) +
        geom_histogram(stat = "count", colour = "black", fill = "white") +
        geom_text(stat = "count", aes(label = ..count..), vjust = 1.2, size = 5)
            +
        labs(x = "Verification Status", y = "Count")
table4 = table(data.clean$verification_status, data.clean$loan_status.b)
round(prop.table(table4, 2) * 100, 2)

#application type
ggplot(data.clean, aes(x = application_type)) +
        geom_histogram(stat = "count", colour = "black", fill = "white") +
        geom_text(stat = "count", aes(label = ..count..), vjust = 1.2, size = 5)
            +
        labs(x = "Application Type", y = "Count")
table5 = table(data.clean$application_type, data.clean$loan_status.b)
round(prop.table(table5, 1) * 100, 2)

ggplot(data.clean, aes(x = grade, y = loan_amnt, fill = application_type)) +
        geom_boxplot() +
        labs(x = "Loan grade assigned by Lending Club", y = "Loan Amount ($)",
            fill = "Application Type")
ggplot(data.clean, aes(x = grade, y = int_rate, fill = application_type)) +
        geom_boxplot() +
        labs(x = "Loan grade assigned by Lending Club", y = "Interest Rate (%)",
            fill = "Application Type")

#employment length
table6 = table(data.clean$emp_length, data.clean$loan_status.b)
round(prop.table(table6, 2) * 100, 2)

#term
table7 = table(data.clean$term, data.clean$loan_status.b)
round(prop.table(table7, 1) * 100, 2)

#home ownership
table8 = table(data.clean$home_ownership, data.clean$loan_status.b)
round(prop.table(table8, 2) * 100, 2)

#########################################################
###################Relevel################################
#########################################################

data.clean = data.clean[!(data.clean$home_ownership == "ANY"),]
data.clean = data.clean[!(data.clean$home_ownership == "NONE"),]
data.clean = data.clean[!(data.clean$home_ownership == "OTHER"),]

data.clean$disbursement_method = factor(data.clean$disbursement_method)
data.clean$hardship_flag = factor(data.clean$hardship_flag)
data.clean$application_type = factor(data.clean$application_type)
data.clean$initial_list_status = factor(data.clean$initial_list_status)
data.clean$purpose = factor(data.clean$purpose)
data.clean$pymnt_plan = factor(data.clean$pymnt_plan)
data.clean$verification_status = factor(data.clean$verification_status)
data.clean$home_ownership = factor(data.clean$home_ownership)
data.clean$emp_length = factor(data.clean$emp_length, levels = c("< 1 year", "1
```

```
    year", "2 years", "3 years", "4 years", "5 years", "6 years", "7 years", "8
    years", "9 years", "10+ years"))
data.clean$grade = factor(data.clean$grade)
data.clean$term = factor(data.clean$term)
data.clean$loan_status.b = factor(data.clean$loan_status.b, levels = c("Non-
    Defaulted", "Defaulted"))

#########################################################
############Convert Categorical to Dummies#############
#########################################################

data.clean.d = fastDummies::dummy_cols(data.clean, select_columns = c("
    disbursement_method", "hardship_flag", "application_type", "initial_list_
    status", "purpose", "pymnt_plan", "verification_status", "home_ownership", "
    emp_length", "grade", "term"), remove_first_dummy = TRUE)
data.clean.d = subset(data.clean.d, select = -c(disbursement_method, hardship_
    flag, application_type, initial_list_status, purpose, pymnt_plan, verification
    _status, home_ownership, emp_length, grade, term, loan_status))
data.clean.d$loan_status.b = ifelse(data.clean.d$loan_status.b == "Non-Defaulted"
    , 0, 1)

#########################################################
############Near Zero Var###############################
#########################################################

library(caret)

##### removing predictors that have near-zero variance
nearZeroVar(data.clean.d)
data.clean.d = data.clean.d[,-c(5,8,10,11,12,30,31,38,39,40,44,53)]


#########################################################
############Data Slicing################################
#########################################################

library(caret)

set.seed(5396) #for reproducibility
train_idx <- createDataPartition(y = data.clean.d$loan_status.b, p = 0.7, list =
    FALSE)
train <- data.clean.d[train_idx,]
test <- data.clean.d[-train_idx,]

#as train data still consists of 1,171,456 rows, we need to partition more so
    computes faster
train_idx_more <- createDataPartition(y = train$loan_status.b, p = 0.01, times =
    10)
test_idx_more <- createDataPartition(y = test$loan_status.b, p = 0.01)

#########################################################
############Penalized Est Implementation###############
#########################################################

library(glmnet)
library(plotmo)

train_more1 <- train[train_idx_more[[1]],]
test_more1 <- train[test_idx_more[[1]],]

x1 <- model.matrix(loan_status.b ~ ., train_more1)[,-1]
y1 <- train_more1$loan_status.b

x1.test <- model.matrix(loan_status.b ~ ., test_more1)[,-1]
y1.test <- test_more1$loan_status.b

model1.1<-glmnet(x=x1,y=y1,family="binomial", alpha=1)
model1.2<-glmnet(x=x1,y=y1,family="binomial", alpha=0)
```

```
model1.3<-glmnet(x=x1,y=y1,family="binomial", alpha=0.5)
print(model1.1)
print(model1.2)
print(model1.3)

options(scipen = 999)
cvmodel1.1<-cv.glmnet(x=x1,y=y1,family="binomial", alpha=1)
cvmodel1.2<-cv.glmnet(x=x1,y=y1,family="binomial", alpha=0)
cvmodel1.3<-cv.glmnet(x=x1,y=y1,family="binomial", alpha=0.5)

as.data.frame(as.matrix(cbind(coef(cvmodel1.1, s='lambda.min'), coef(cvmodel1.2,
    s='lambda.min'), coef(cvmodel1.3, s='lambda.min'))))

plot_glmnet(model1.1)
plot_glmnet(model1.2)
plot_glmnet(model1.3)

model1.1<-glmnet(x=x1,y=y1,family="binomial", alpha=1, lambda=cvmodel1.1$lambda.
    min)
model1.2<-glmnet(x=x1,y=y1,family="binomial", alpha=0, lambda=cvmodel1.2$lambda.
    min)
model1.3<-glmnet(x=x1,y=y1,family="binomial", alpha=0.5, lambda=cvmodel1.3$lambda
    .min)

model1.1.pred <- ifelse(predict(model1.1, x1.test, type = "response") > 0.5, 1,
    0)
model1.2.pred <- ifelse(predict(model1.2, x1.test, type = "response") > 0.5, 1,
    0)
model1.3.pred <- ifelse(predict(model1.3, x1.test, type = "response") > 0.5, 1,
    0)

confusionMatrix(factor(y1.test), factor(model1.1.pred))
confusionMatrix(factor(y1.test), factor(model1.2.pred))
confusionMatrix(factor(y1.test), factor(model1.3.pred))

#########################################################
#################AIC elimination#########################
#########################################################

library(MASS)
library(dplyr)

model2.1 <- glm(loan_status.b ~ ., data = train_more1, family = "binomial") %>%
    stepAIC(trace = FALSE)

model2.2 <- glm(loan_status.b ~ total_bc_limit + pub_rec_bankruptcies +
                        percent_bc_gt_75 + num_sats + num_accts_ever_120_pd +
                            mths_since_recent_inq +
                        mo_sin_rcnt_tl + bc_open_to_buy + last_credit_pull_d +
                            last_pymnt_amnt +
                        total_rec_int + total_rec_prncp + total_pymnt + out_prncp
                            _inv +
                        revol_util + pub_rec + open_acc + installment + int_rate
                            +
                        funded_amnt_inv + hardship_flag_Y + home_ownership_OWN +
                            home_ownership_RENT,
                        data = train_more1, family = "binomial")
model2.3 <- glm(loan_status.b ~ total_bc_limit + pub_rec_bankruptcies +
                        percent_bc_gt_75 + num_sats + num_accts_ever_120_pd +
                            mths_since_recent_inq +
                        mo_sin_rcnt_tl + bc_open_to_buy + last_credit_pull_d +
                            last_pymnt_amnt +
                        total_rec_int + total_rec_prncp + total_pymnt + out_prncp
                            _inv +
                        revol_util + pub_rec + open_acc + installment + int_rate
                            +
                        funded_amnt_inv + hardship_flag_Y + purpose_credit_card +
                        purpose_debt_consolidation + purpose_educational +
```

```
                            purpose_home_improvement +
                          purpose_house + purpose_major_purchase + purpose_medical
                              +
                          purpose_moving + purpose_other + purpose_renewable_energy
                               +
                          purpose_small_business + purpose_vacation + purpose_
                              wedding +
                          home_ownership_OWN + home_ownership_RENT,
                          data = train_more1, family = "binomial")

model2.2.pred <- ifelse(predict(model2.2, test_more1, type = "response") > 0.5,
    1, 0)
model2.3.pred <- ifelse(predict(model2.3, test_more1, type = "response") > 0.5,
    1, 0)

confusionMatrix(factor(test_more1$loan_status.b), factor(model2.2.pred))
confusionMatrix(factor(test_more1$loan_status.b), factor(model2.3.pred))

#########################################################
#######################KNN###############################
#########################################################

library(caret)

set.seed(5396)
trCtrl <- train(method = "repeatedcv", number = 1, repeats = 1)

model3.1 <- train(factor(loan_status.b) ~ total_bc_limit + pub_rec_bankruptcies +
                  percent_bc_gt_75 + num_sats + num_accts_ever_120_pd + mths_since_
                      recent_inq +
                  mo_sin_rcnt_tl + bc_open_to_buy + last_credit_pull_d + last_pymnt
                      _amnt +
                  total_rec_int + total_rec_prncp + total_pymnt + out_prncp_inv +
                  revol_util + pub_rec + open_acc + installment + int_rate +
                  funded_amnt_inv + hardship_flag_Y + home_ownership_OWN + home_
                      ownership_RENT,
                  data = train_more1, method = "knn", preProcess = c("center", "
                      scale"),
                  tuneLength = 30)
model3.2 <- train(loan_status.b ~ total_bc_limit + pub_rec_bankruptcies +
                  percent_bc_gt_75 + num_sats + num_accts_ever_120_pd + mths_since_
                      recent_inq +
                  mo_sin_rcnt_tl + bc_open_to_buy + last_credit_pull_d + last_pymnt
                      _amnt +
                  total_rec_int + total_rec_prncp + total_pymnt + out_prncp_inv +
                  revol_util + pub_rec + open_acc + installment + int_rate +
                  funded_amnt_inv + hardship_flag_Y + purpose_credit_card +
                  purpose_debt_consolidation + purpose_educational + purpose_home_
                      improvement +
                  purpose_house + purpose_major_purchase + purpose_medical +
                  purpose_moving + purpose_other + purpose_renewable_energy +
                  purpose_small_business + purpose_vacation + purpose_wedding +
                  home_ownership_OWN + home_ownership_RENT,
                  data = train_more1, method = "knn", preProcess = c("center", "
                      scale"),
                  tuneLength = 30)

model3.1.pred <- predict(model3.1, x1.test)
model3.2.pred <- ifelse(predict(model3.2, x1.test) > 0.5, 1, 0)

confusionMatrix(factor(model3.1.pred), factor(y1.test))
confusionMatrix(factor(model3.2.pred), factor(y1.test))

#########################################################
#######################Question 2########################
#########################################################

setwd("C:/Users/wilso/Documents/2. UIUC/7. 2020 Spring/ASRM 499/Final Project")
```

```
load("LCwithissuedate.rda")

datanew$issue_m <- format(as.yearmon(as.Date("2018-12-01")) - round(datanew$issue
    _d * 12) * (1/12),
                          "%Y-%m")

accepted_m <- c("2012-12","2013-01","2013-02","2013-03","2013-04","2013-05",
                "2013-06","2013-07","2013-08","2013-09","2013-10")

datanew.a <- datanew[which(datanew$issue_m %in% accepted_m),]
datanew.a$loan_status.b <- factor(ifelse(datanew.a$loan_status %in% c("Fully Paid
    ", "Current",
                                                                      "In Grade Period",
                                                                           "Late (16-30
                                                                          days)"),
                                         0,1), levels = c(0, 1))

datanew.b <- subset(datanew.a, select = c(loan_status.b, loan_amnt, issue_m,
    grade, term))

datanew.c <- datanew.b[which(datanew.b$loan_amnt >= 5000 & datanew.b$loan_amnt <=
     20000),]

datanew.d <- datanew.c[which(datanew.c$term %in% "36 months"),]

datanew.d$D <- ifelse(datanew.d$loan_amnt >= 12000 & datanew.d$loan_amnt < 16000
    & as.yearmon(datanew.d$issue_m) >= as.yearmon("2013-03"), 1,
                      ifelse(datanew.d$loan_amnt >= 10000 & datanew.d$loan_amnt <
                          12000 & as.yearmon(datanew.d$issue_m) >= as.yearmon("
                          2013-07"), 1, 0))

datanew.d$bin <- cut(datanew.d$loan_amnt, breaks = 1000*c(5:20), labels = c(1:15)
    , include.lowest = TRUE)

set.seed(5396) #for reproducibility
train_idx <- createDataPartition(y = datanew.d$loan_status.b, p = 0.7, list =
    FALSE)
train <- datanew.d[train_idx,]
test <- datanew.d[-train_idx,]

model4.1 <- glm(loan_status.b ~ factor(grade):factor(issue_m) + D + bin, family =
     "binomial", data = train)

test <- test[which(test$grade != "G"),]
model4.1.pred <- ifelse(predict(model4.1, test, type = "response") > 0.1, 1, 0)

confusionMatrix(factor(test$loan_status.b), factor(model4.1.pred))
```