Significant Factors in Credit Card Default Prediction

Zhuo Feng Lei

May 12, 2020

Abstract

To avoid significant financial losses, banks must be able to recognize whether clients will default on their credit card payments. Statistical machine learning methods and techniques are utilized to analyze significant factors in predicting default payments and find an accurate model that best classify whether clients will default on credit card payments.

Contents

1 Introduction

Credit cards are flexible tools issued by banks to consumers. It enables the cardholder to pay a merchant for goods and services based on the cardholder's promise to the card issuer to pay them for the amounts plus other agreed charges by the due date listed on the credit card statement. When a customer fails to pay back the loan by the due date, the credit card will be defaulted. If the bank is certain that they will not get their money back, the bank will try to sell the loan. If the bank is unable to sell off the loan to debt collectors, the bank will write off the loan (charge-off). As a result, the customer's credit rating tanks while the bank suffers significant financial losses.

Risk prediction is a powerful tool that is utilized by banks and other financial service providers. Risk predictions is essential for predicting customer's credit risks or business performance. The ability to predict which customers are most probable to default on payments will allow banks to reduce damages and uncertainty when issuing credit cards to customers. By detecting the potential default early on, they may take early preventative actions. However, analyzing millions of transactions and client information to make a prediction is resource consuming, takes a long time, and prone to errors. Therefore, it is more efficient to use machines to compute and analyze the data.

Financial service providers and credit card companies can minimize risk and losses by analyzing the significant factors that causes clients to default on their payments. Various machine learning techniques and methods are used to classify payments based on the client's demographical and financial information. By comparing the model's performance and summary statistics, common significant features can be found. These significant features can be used to help explain why clients tend to default on their credit card payments and alert financial companies of these potential 'red flags'.

1.1 Literature Review

Machine learning is commonly used in the finance industry to assess risk and predict default. The research work of [1] combines both machine learning and heuristic approach to predict default on the same Taiwan credit card default data set. They first use supervised machine learning algorithms to determine the best algorithm for predicting default. Using the best algorithm, they compute a credit default probability score from archived transaction history. Next, they continuously update the probability score as new transactions occur and apply a heuristic to compute a risk score. The risk score and the archived score is used to calculate the client's overall risk probability. Their study involves finding a good model for analyzing the clients' risk. My study only utilizes machine learning algorithms, but I focus more on feature selection and default prediction accuracy. I also use some additional models that [1] have not cover in their study such as neural networks.

The research work of [2] also covers a similar topic on the same data set. In the work of [2], they used AdaBoost, an adaptive boosting-based online learning algorithm, to predict whether clients are at risk of defaulting. After receiving new data, a re-weighting technique stabilized the dynamic and stochastic process changes. The performance of AdaBoost was compared to various state-of-art classification methods such as random forest, support vector machines, and naïve Bayesian models. While [2] updates their model with real time data, I only consider offline data. However, I applied a different set of algorithms and approaches to find significant predictors.

My project has some similarities with the work of [1][2]. Some of the machine algorithms that I used such as random forest and KNN had appeared in the works mentioned previously. In addition, both [1] and [2] updated their models with real time data. For my project, I did not update my models with real time data. However, I do implement new machine learning algorithms and approaches that did not appear in [1] and [2] such as neural network. In

addition, my project focuses more on analyzing the significant predictors that affect whether payments default or not while [1] focus on finding the best model to assess overall risk probability and [2] focuses on finding a model that best classify the default payments correctly.

2 Data

The data was accessed via Kaggle [3]. The data set contains information on default payments, demographic factors, credit data, history of payment, and bill statements of credit card clients in Taiwan from April 2005 to September 2005. The dataset has 30,000 observations and 25 variables. The response variable is ***default.payment.mext.month***, whether the payment will default next month.

The input variables and explanations as to why they might be significant is stated as follows:

- **LIMIT_BAL**: Amount of given credit in NT dollars (includes individual and family/supplementary credit

  Clients with a good credit score usually have higher limit balances since they are deemed more reliable and trustworthy.

- **SEX**: Gender of the client (1 = male, 2 = female)

  Gender can be a decent predictor since there are gender wage gaps. Men may earn more money than women in some workplaces. Having high wages usually indicates that clients can pay back what they owe.

- **EDUCATION**: (1 = graduate school, 2 = university, 3 = high school, 4 = others, 5 = unknown

  Higher education usually means more job opportunities which may affect the client's ability to pay their credit card payments. Student loans may also affect payments.

- **MARRIAGE**: Marital status (1 = married, 2 = single, 3 = others

  Marriage can lead to having kids (expensive to raise) or can bring dual income to a household. This may affect whether credit card payments default or not.

- **AGE**: Age in years

  Younger adults may still be in school (no income or very little income). Working young adults also tend to start out in entry level jobs that doesn't pay as much as senior positions.

- **PAY_0**: Repayment status in September 2005 (-1 = pay duly, 1 = payment delay for one month, 2 = payment delay for 2 months, … 8 = payment delay for eight months, 9 = payment delay for 9 months and above)
- **PAY_2**: Repayment status in August 2005 (factors same as above)
- **PAY_3**: Repayment status in July 2005 (factors same as above)
- **PAY_4**: Repayment status in June 2005 (factors same as above)
- **PAY_5**: Repayment status in May 2005 (factors same as above)
- **PAY_6**: Repayment status in April 2005 (factors same as above)

Repayment status helps determine how clients are doing financially. Delayed payments indicate that clients are having difficulties paying back the money that they owe.

- **BILL_AMT1**: Amount of bill statement in September, 2005 (NT dollar)
- **BILL_AMT2**: Amount of bill statement in August, 2005 (NT dollar)
- **BILL_AMT3**: Amount of bill statement in July, 2005 (NT dollar)
- **BILL_AMT4**: Amount of bill statement in June, 2005 (NT dollar)
- **BILL_AMT5**: Amount of bill statement in May, 2005 (NT dollar)
- **BILL_AMT6**: Amount of bill statement in April, 2005 (NT dollar)

A high bill amount may indicate bad spending habits. Lower billing amounts are also easier to pay off.

- **PAY_AMT1**: Amount of previous payment in September, 2005(NT dollar)
- **PAY_AMT2**: Amount of previous payment in August, 2005(NT dollar)
- **PAY_AMT3**: Amount of previous payment in July, 2005(NT dollar)
- **PAY_AMT4**: Amount of previous payment in June, 2005(NT dollar)
- **PAY_AMT5**: Amount of previous payment in May, 2005(NT dollar)
- **PAY_AMT6**: Amount of previous payment in April, 2005(NT dollar)

Payment amount is like repayment status variable except that it is in a numeric variable. It also helps determine if clients are having financial difficulties.

2.1 Data Split

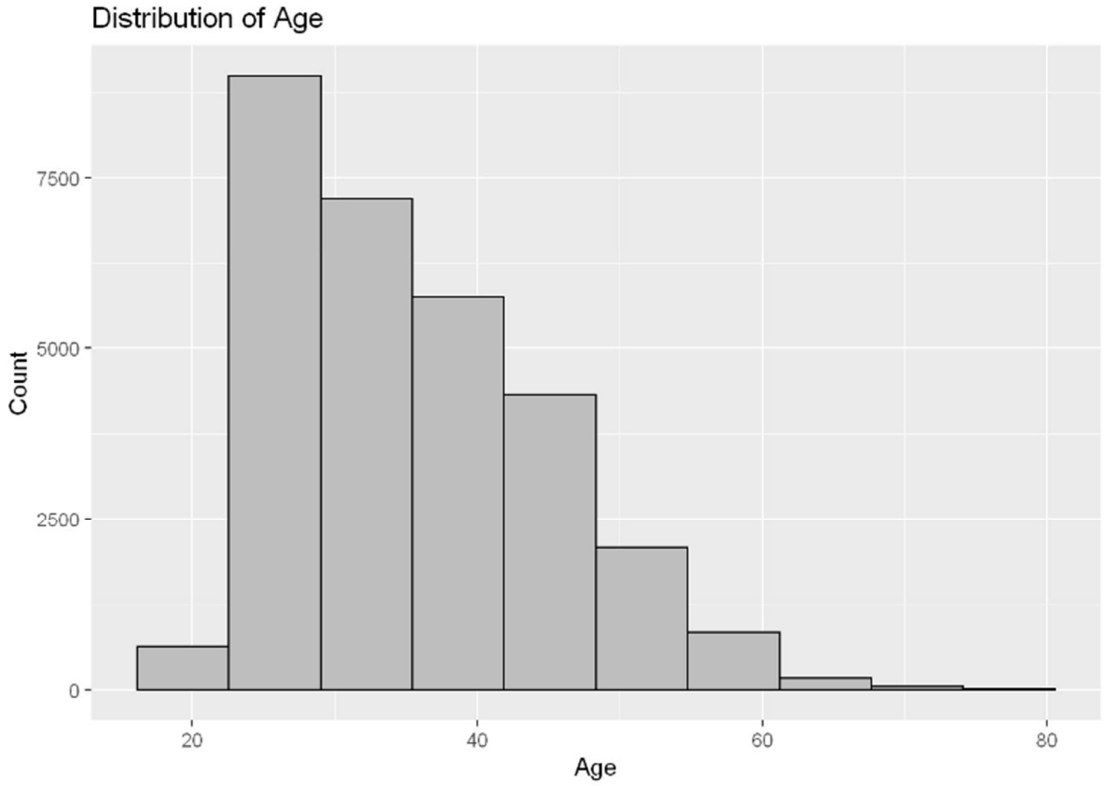
The data set will be split into a training and testing set. 70% of the observations will be randomly sampled to create a training data set and the remaining will be used to create the testing data set. Models will be trained using the training set, and performance will be assessed using the testing set. Accuracy is used as a metric to determine how well the model can classify whether payments will default or not.

2.2 Summary Statistics

Before I begin to apply machine learning algorithms to the data, it is helpful to look at some summary statistics to help gain insights about the data.

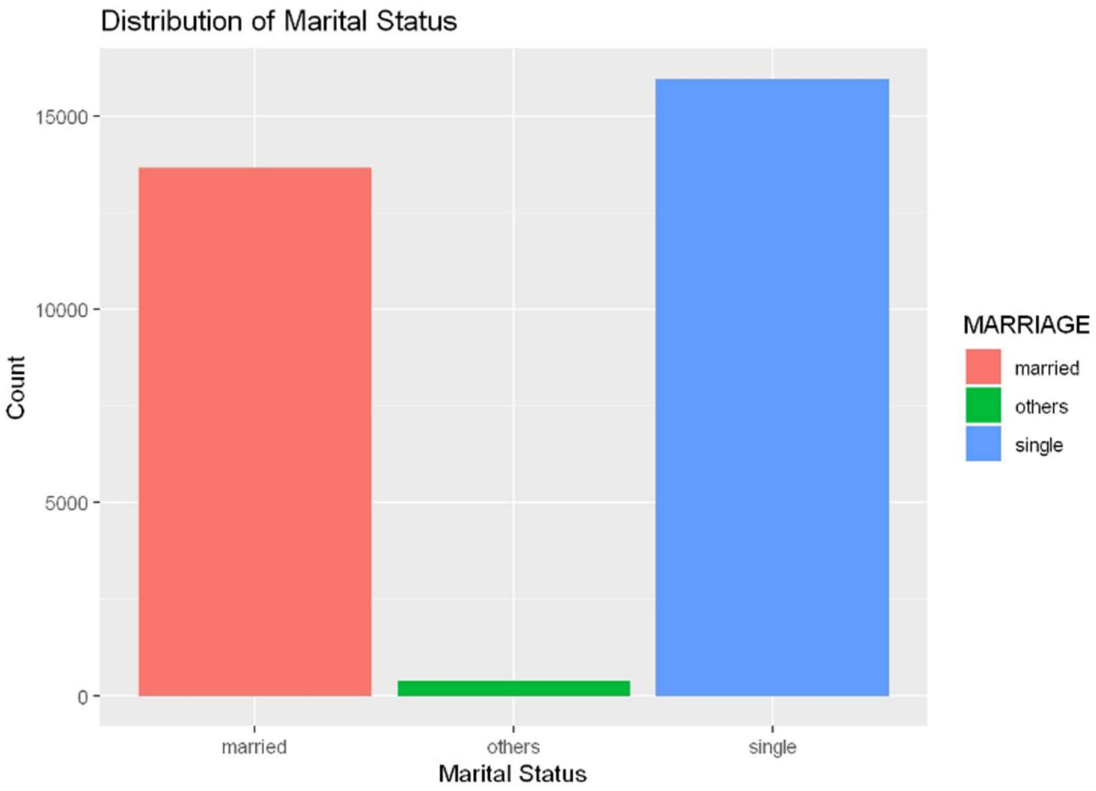
2.2.1 Univariate Analysis

Figure A. Distribution of Age



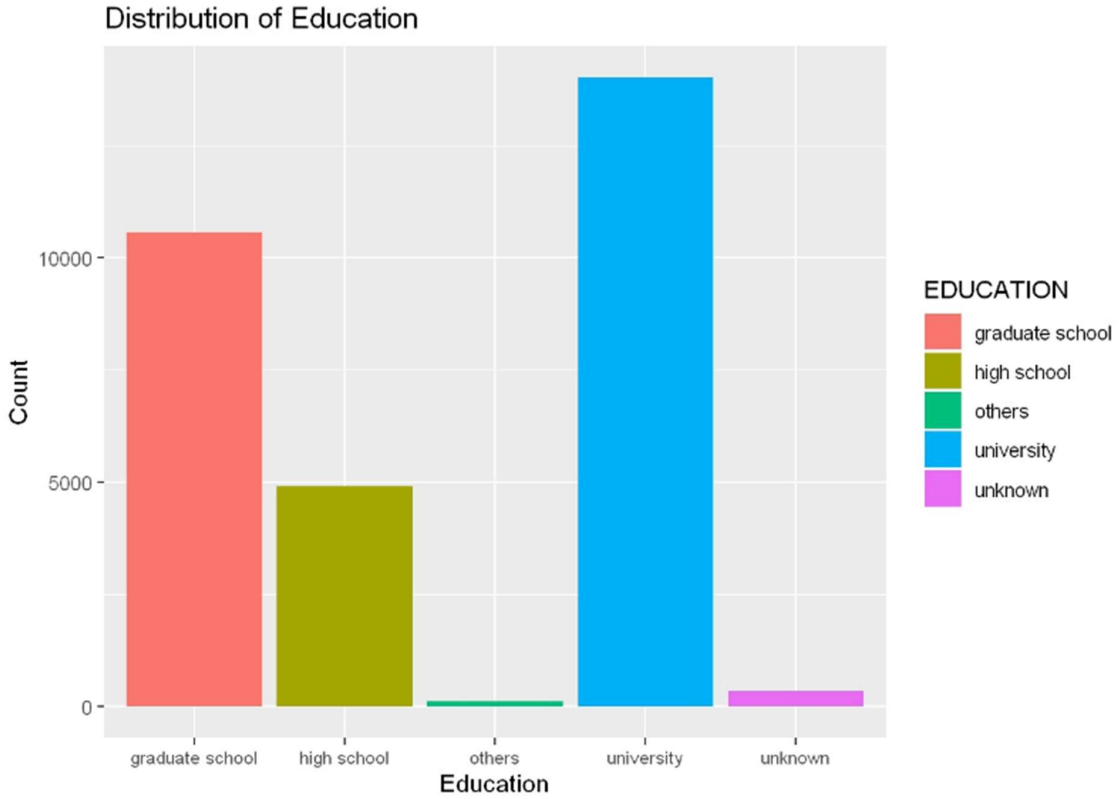
Most of the clients are in high 20s and 30s. There are very little clients in the 20s. This is probably due to low credit scores in young adults.

Figure B. Distribution of Marital Status



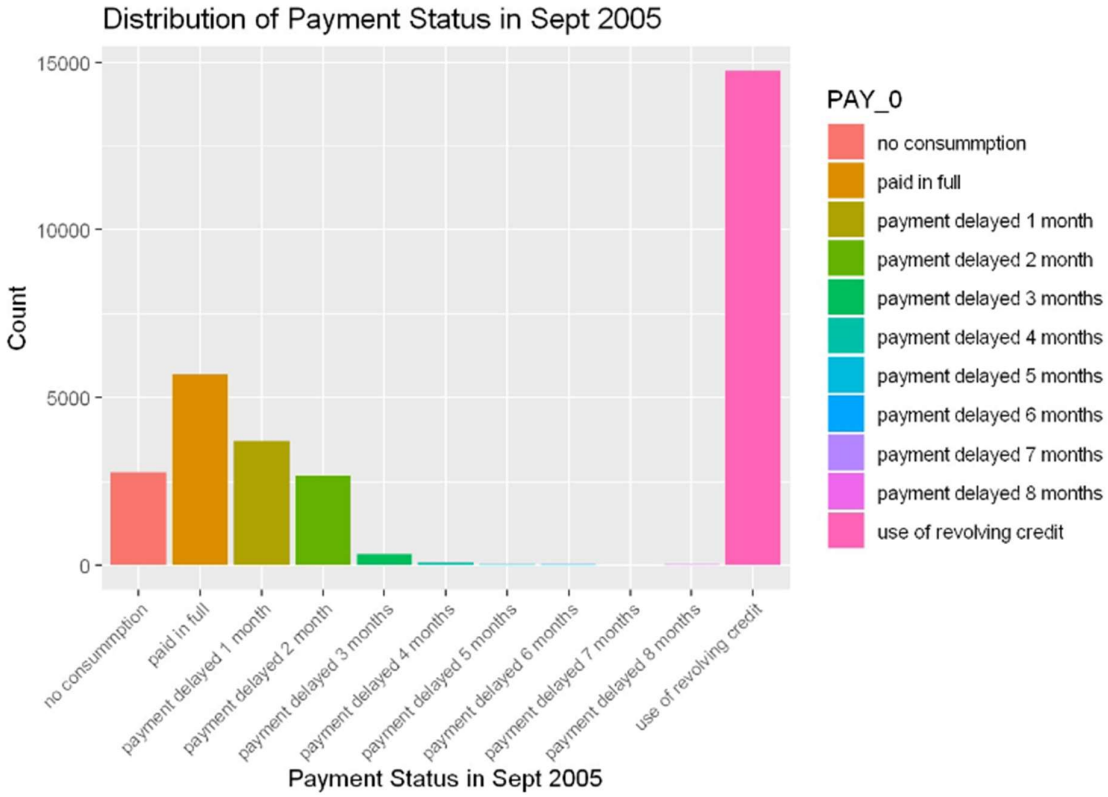
The sample distribution of marital status is roughly even.

Figure C. Distribution of Education



Most of the client's either have graduate school or university as their highest form of education. The third most group is high school.

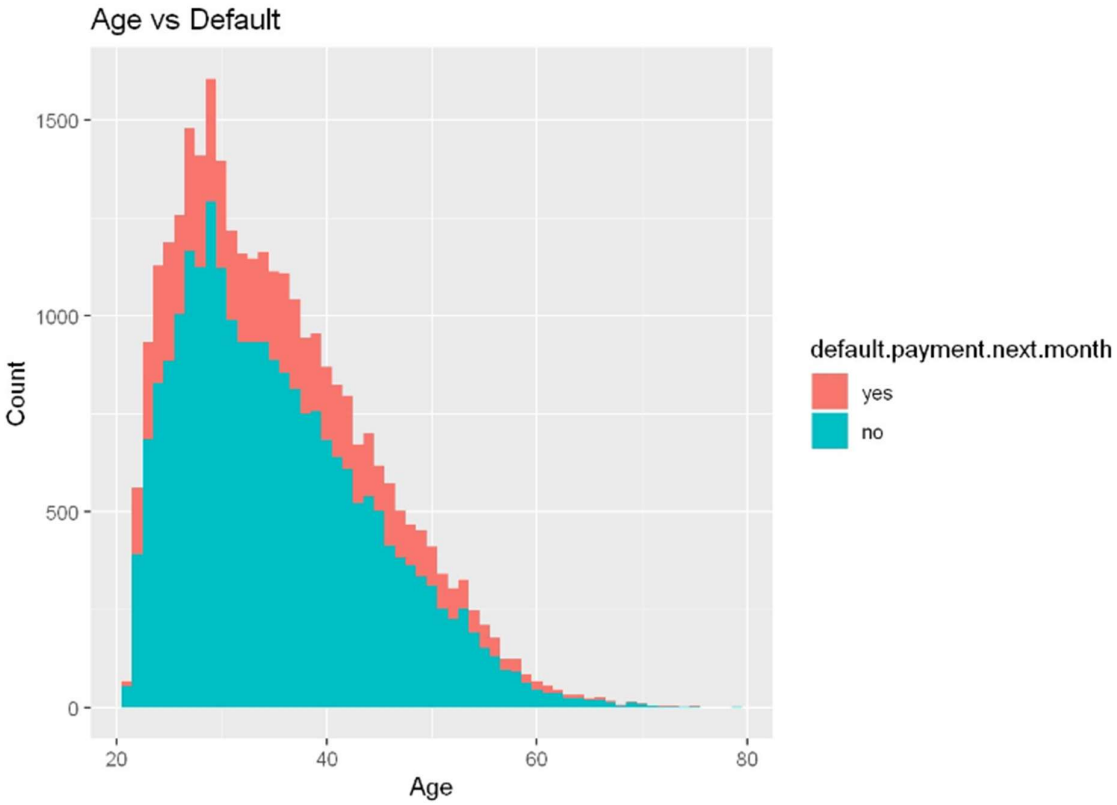
Figure D. Distribution of Payment Status in September 2015



It appears that clients mostly use revolving credit for their payments in Sept 2005. Stagnant wages coupled with the rising cost of living may cause people to live paycheck to paycheck. As a result, many people may not be able to pay off all their credit card bill and use revolving credit. The second most frequent status is clients paying off all their balance.

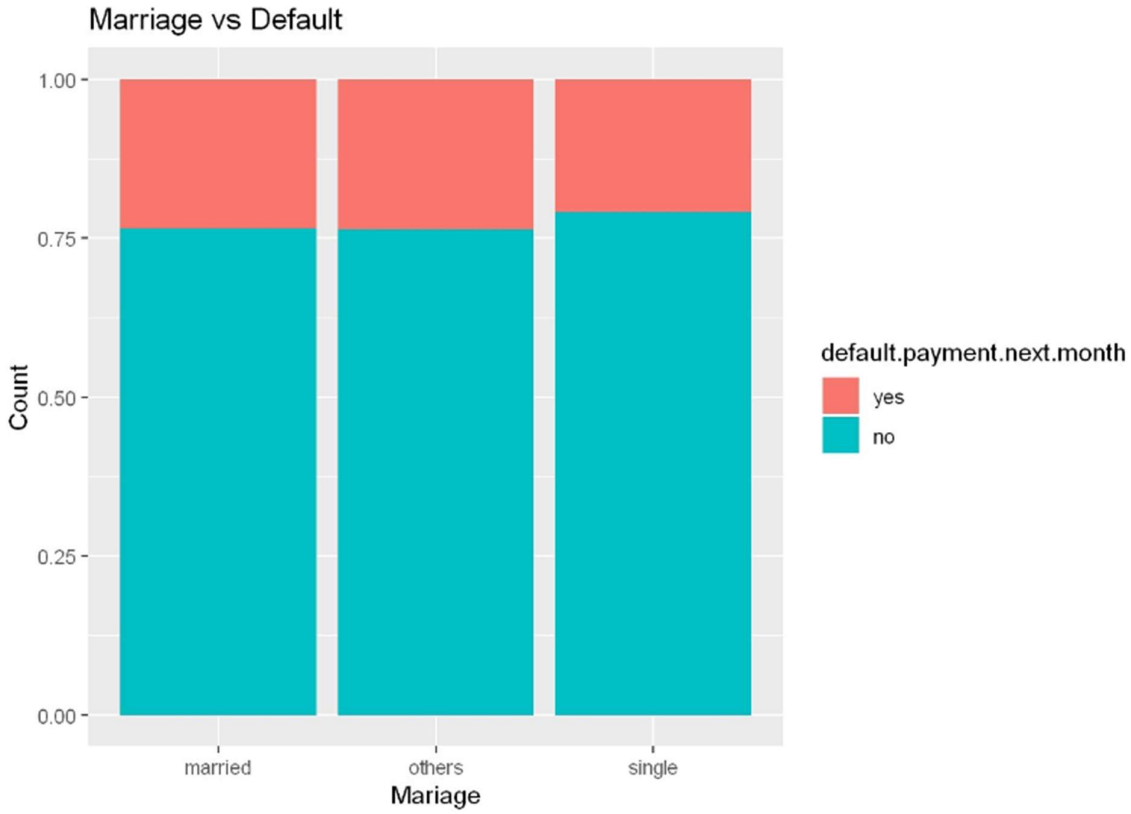
2.2.2 Bivariate Analysis

Figure E. Distribution of Age vs Default



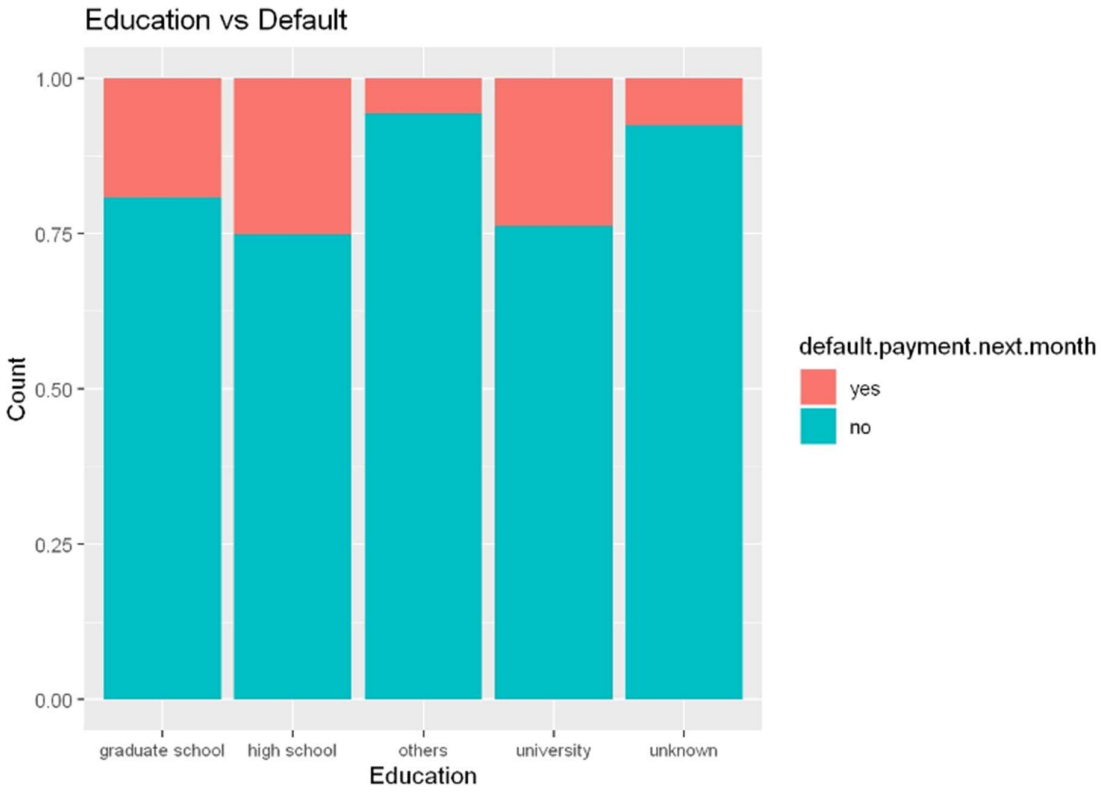
The age distribution for default payments are more skewed than the age for non-default payments. Many people that default in their next month payments are in their 20s to 40s. As age increases, the number of default payments seem to decrease. It appears that age have a significant effect on whether payments will default, but additional analysis is needed to determine if this is truly the case.

Figure F. Distribution of Marriage vs Default



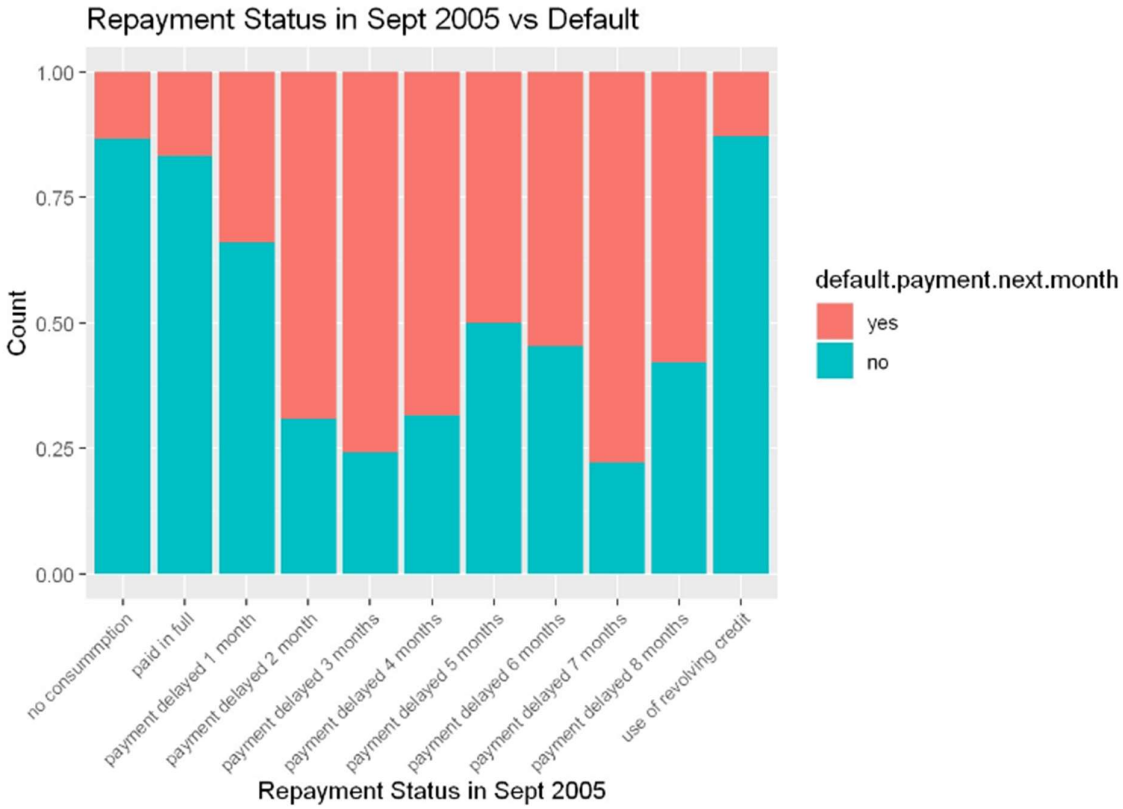
It appears that marital status does not have significantly affect whether payments will default since all three groups have roughly the same ratio of default payments in their respective groups. Additional analysis is needed to determine whether marital status have any significant effects on default payments.

Figure G. Distribution of Education vs Default



Clients whose highest education was university or high school have the largest ratio of default payments. In the case of university, this may be due to student loans or field of studies. In the case of high school, clients may not have high income due to the lack of higher education. Additional analysis is needed to determine if education have any significant effect on whether payments will default.

Figure H. Distribution of Repayment Status in Sept 2005 vs Default



It appears that the payments that are delayed for 2,3,4 or 7 months have the highest ratio of default payments in their respective populations. Additional analysis is needed to determine if repayment status has any significant effects on the outcome.

3 Modeling

3.1 Logistic Regression

Logistic regression studies the association between a categorical dependent variable and a set of independent variables. Using the input features, logistic regression models the probability that Y belongs to one of the categories. In this case, logistic regression is used to estimate the probability of whether payments default or not. The logistic function is as follows:

$$p(X) = \frac{e^{\beta_0+\beta_1 X}}{1 + e^{\beta_0+\beta_1 X}}$$

Table 1. Logistic Regression Models and Input Features

Model	Input Features
Logistic Regression 1	All input features mentioned in the data section
Logistic Regression 2	All input features except **PAY_0**, **PAY_2**, **PAY_3**, **AGE**, and **MARRIAGE**
Logistic Regression 3	**LIMIT_BAL**, **AGE**, **EDUCATION**, **PAY_0**, **PAY_2**
Logistic Regression 4	**LIMIT_BAL**, **EDUCATION**, **MARRIAGE**, **AGE**, **PAY_0**, **BILL_AMT1**
Logistic Regression 5	**LIMIT_BAL**, **EDUCATION**, **MARRIAGE**, **AGE**, **PAY_0**, **PAY_2**, **PAY_3**, **PAY_4**, **PAY_5**, **PAY_6**

Table 2. Performance of Logistic Regression Models on Testing Data Set

Model	Accuracy
Logistic Regression 1	**0.8214776**
Logistic Regression 2	**0.7971429**
Logistic Regression 3	**0.8200486**
Logistic Regression 4	**0.8210481**
Logistic Regression 5	**0.8210489**

The best logistic model is the one with all the input features. It has the highest accuracy rate out of all 5 models. However, logistic regression model 4 and 5 performs similarly to logistic regression 1, but their error rate is slightly higher. Independent features with p-value less than .05 have a significant effect on the dependent feature (whether payments will default or not in the next month), and their coefficients are significantly different from zero. Features with lower p-values have a larger effect on the outcome. By viewing the coefficients of the logistic regression model, I can identify the features the model considers to be significant in predicting default payments.

Figure 1. Coefficients of Significant Input Features in Logistic Regression Model 1

```
##                      LIMIT_BAL                          SEXmale
##                   1.852798e-06                    -1.883685e-01
##          `EDUCATIONhigh school`              EDUCATIONuniversity
##                   6.561628e-02                    -1.418319e-02
##                  MARRIAGEothers                              AGE
##                   8.667289e-02                    -3.416999e-03
##  `PAY_0payment delayed 1 month`  `PAY_0payment delayed 2 month`
##                   -8.810109e-01                    -2.123550e+00
## `PAY_0payment delayed 3 months` `PAY_0payment delayed 4 months`
##                   -2.143407e+00                    -1.726956e+00
## `PAY_0payment delayed 5 months`  `PAY_2payment delayed 1 month`
##                   -6.706212e-01                     3.719730e-01
## `PAY_3payment delayed 3 months` `PAY_3payment delayed 4 months`
##                   -5.319631e-01                     8.325471e-02
## `PAY_4payment delayed 5 months` `PAY_6payment delayed 4 months`
##                   4.563092e-01                     3.679199e-02
##                      BILL_AMT1                         PAY_AMT2
##                   1.258920e-06                    9.862795e-06
##                       PAY_AMT3
##                   4.270652e-07
```

According to the coefficients of the logistic regression model with all the input features, some of the significant features are the balance limit on the account, the male gender, unknown education status, how much the client paid in the first and second month, and repayment status in the first month. Some of the coefficients from the logistic regression model seems illogical in a real-life scenario. I agree that if clients owe more to the card companies, the odds of defaulting increases. People with high payment amounts are usually bad with money management and live a lifestyle they cannot sustain. However, the model also says that delayed payments reduce the chance of defaulting. It will make more sense if

delayed payments increase the chance of defaulting since it indicates that the client cannot financially pay back the amount owed. It is also interesting to note that being single increases the chances of defaulting. This may be caused by people usually wait until they are financially stable before starting a family (raising kids cost a lot). Being male also lowers the chances of defaulting. This may be caused by the gender wage gaps. Additional modeling and research are needed to verify whether these relationships are true or not.

Figure 2: Logistic Regression Model 1 Confusion Matrix

```
## Cross-Validated (10 fold) Confusion Matrix
##
## (entries are percentual average cell counts across resamples)
##
##           Reference
## Prediction  yes   no
##        yes  8.0  3.7
##        no  14.2 74.1
##
##  Accuracy (average) : 0.8215
```

To summarize, logistic regression model 1 accurately classified ~82.15% of the data. The true positive rate is 8% while the false positive rate is 3.7%. Out of 22.2% clients that defaulted, the model only accurately classified 8% of them. The logistic regression model should not be used for classifying whether payments will default because it missed 14.2% of the clients that defaulted. In addition, some of the coefficients appears to be illogical and does not reflect actual client behavior or circumstances very well. In a business context, this would mean significant financial losses as the model failed to catch a lot of the clients that will default on their card payments. However, the model does provide some insight on the significant factors that may cause payments to default.

3.2 Probit Regression

The probit model is very similar to the logit model. It also studies the association between a categorical dependent variable and a set of independent variables. The difference between the

logistic model and the probit model is the nonlinear function that the model fits to the data. The probit model's function is as follows:

$$p(X) = \phi(\beta_0 + \beta_1 X)$$

Since the probit regression is quite like the logit regression, I expect both models to perform similarly on the testing data. After running the probit regression with all 24 input features, I discovered that probit regression has a slightly higher accuracy than the logistic regression. The logistic regression model accurately classified ~82.15% of the testing data while the probit regression correctly classified ~82.18%.

Let's look at the coefficients of the significant features in the probit model. Both models have similar significant features. Like the logistic regression model, some of the probit regression results also does not make sense in a business context. It is illogical that delayed payments decrease the chances of defaulting.

Figure 3. Coefficients of Significant Input Features in Probit Regression Model

```
##                      LIMIT_BAL                          SEXmale
##                   1.048554e-06                    -1.054614e-01
##          `EDUCATIONhigh school`              EDUCATIONuniversity
##                   2.953039e-02                    -1.394896e-02
##                  MARRIAGEothers                              AGE
##                   4.691508e-02                    -1.975981e-03
##  `PAY_0payment delayed 1 month`  `PAY_0payment delayed 2 month`
##                   -5.044629e-01                    -1.264947e+00
## `PAY_0payment delayed 3 months` `PAY_0payment delayed 4 months`
##                   -1.288808e+00                    -1.020902e+00
## `PAY_0payment delayed 5 months`  `PAY_2payment delayed 1 month`
##                   -3.505612e-01                     2.616484e-01
## `PAY_3payment delayed 3 months` `PAY_3payment delayed 4 months`
##                   -3.158608e-01                     5.221333e-02
## `PAY_6payment delayed 4 months`                        BILL_AMT1
##                   1.703023e-02                     2.184918e-07
##                       PAY_AMT2                         PAY_AMT3
##                   4.032940e-06                     4.830022e-07
```

Both logistic and probit regression have similar results and performance on the data. Both models returned the same significant features with slightly varying coefficients. However, some of the coefficients of the significant also do not make sense in realistic applications. Like the logistic

regression, the probit regression provides some insight on significant variables but should not be used to classify default payments for businesses.

3.3 K-Nearest Neighbors (KNN)

K-nearest neighbor is an unsupervised machine learning algorithm. The main idea is to define K, the number of nearest neighbors that we want to consider for each observation. After choosing K, the algorithm finds K neighbors for each observation. Finally, the neighbors in each group are counted. The objective function of KNN is as follows:

$$Pr(Y = j|X = x_0) = \frac{1}{K}\sum_{i \in N_0} I(y_i = j)$$

KNN performs well if the actual relations in the data set in non-linear. However, KNN suffers from the curse of dimensionality. It does not perform well when the number of parameters increase. The data set contains many predictors. The input feature array contains 24 elements. This may result in observations that have no nearby neighbors. Base on this phenomenon, it can be expected that KNN will not perform that well on the Taiwan credit card payment default data.

Figure 4. KNN Model Summary

```
## k-Nearest Neighbors
##
## 21000 samples
##    24 predictor
##     2 classes: 'yes', 'no'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 18899, 18900, 18900, 18899, 18901, 18899, ...
## Resampling results across tuning parameters:
##
##   k  Accuracy   Kappa
##   5  0.7540948  0.1237021
##   7  0.7593810  0.1082840
##   9  0.7649999  0.1099473
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was k = 9.
```

I tuned the model to find the best with the lowest error rate. The algorithm misclassified the least observations in the testing data with 9 nearest neighbors. It accurately predicted ~76.5% of the test data. The data set have high dimensions which prevents the KNN model from accurately classifying default credit payments. It is also important to note that the KNN model does not have any feature selection. It is hard to tell which features are significant at predicting default payments.

3.4 Ridge Regression

To select significant features, it is useful to use subset selection methods to find a subset of p regressors that will output the best model. Shrinkage methods are useful since they use all p of the regressors but uses a technique that shrinks the estimates. Ridge regression is one of these shrinkage methods. The model performs better than least squares when the least square estimates have high variance. Ridge regressions tend to have lower variance than least squares in exchange of a higher bias. The functional form of Ridge Regression is :

$$\mathcal{L} = \sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1} p\beta_j x_{ij}\right)^2 + \lambda\sum_{j=1}^{p}\beta_j^2$$

Using all the regressors, I fit a ridge regression model to the training data. I tuned lambda in the ridge regression model to get the lowest testing error rate. The tuned ridge regression model accurately classified ~81.66% of the testing data. Ridge regression does not perform variable selection since it does not set any of the coefficient estimates to exactly be 0 (only towards 0). However, I can still draw insight about the significant variables that predict default payment from the coefficient estimates.

Figure 5. Coefficients of Ridge Model:

LIMIT_BAL	SEXmale	EDUCATIONhigh school	EDUCATIONothers	EDUCATIONuniversity	
1.661062e-06	-1.361582e-01	1.756536e-03	8.742573e-01	-4.037247e-02	
EDUCATIONunknown	MARRIAGEothers	MARRIAGEsingle	AGE	PAY_0paid in full	
8.606922e-01	6.456639e-02	1.286576e-01	-3.507986e-03	-1.630770e-01	
PAY_0payment delayed 1 month	PAY_0payment delayed 2 month	PAY_0payment delayed 3 months	PAY_0payment delayed 4 months	PAY_0payment delayed 5 months	
-4.969004e-01	-1.628793e+00	-1.577554e+00	-1.261570e+00	-9.069509e-01	
PAY_0payment delayed 6 months	PAY_0payment delayed 7 months	PAY_0payment delayed 8 months	PAY_0use of revolving credit	PAY_2paid in full	
-2.232731e-01	-1.147306e+00	1.055774e-01	4.274054e-01	6.560073e-02	
PAY_2payment delayed 1 month	PAY_2payment delayed 2 month	PAY_2payment delayed 3 months	PAY_2payment delayed 4 months	PAY_2payment delayed 5 months	
5.365009e-01	-2.116594e-01	-2.624525e-01	4.183582e-01	-9.474963e-01	
PAY_2payment delayed 6 months	PAY_2payment delayed 7 months	PAY_2payment delayed 8 months	PAY_2use of revolving credit	PAY_3paid in full	
-1.066636e+00	-7.485074e-01	1.431973e+00	-5.506756e-02	6.232664e-02	
PAY_3payment delayed 1 month	PAY_3payment delayed 2 month	PAY_3payment delayed 3 months	PAY_3payment delayed 4 months	PAY_3payment delayed 5 months	
9.411059e-01	-3.189696e-01	-3.378665e-01	1.185642e-01	6.307373e-01	
PAY_3payment delayed 6 months	PAY_3payment delayed 7 months	PAY_3payment delayed 8 months	PAY_3use of revolving credit	PAY_4paid in full	
-1.164618e+00	-2.122372e-01	1.226202e+00	-9.955529e-03	7.372505e-02	
PAY_4payment delayed 1 month	PAY_4payment delayed 2 month	PAY_4payment delayed 3 months	PAY_4payment delayed 4 months	PAY_4payment delayed 5 months	
-2.685573e+00	-2.598629e-01	-1.439501e-01	-4.033678e-01	8.453430e-01	
PAY_4payment delayed 6 months	PAY_4payment delayed 7 months	PAY_4payment delayed 8 months	PAY_4use of revolving credit	PAY_5paid in full	
2.641790e+00	-3.274304e-02	3.564582e+00	2.586693e-02	9.995637e-02	
PAY_5payment delayed 2 month	PAY_5payment delayed 3 months	PAY_5payment delayed 4 months	PAY_5payment delayed 5 months	PAY_5payment delayed 6 months	
-2.389425e-01	-6.122996e-02	1.919235e-01	-5.895302e-01	-1.206964e+00	
PAY_5payment delayed 7 months	PAY_5payment delayed 8 months	PAY_5use of revolving credit	PAY_6paid in full	PAY_6payment delayed 2 month	
-6.188929e-01	-2.585703e+00	3.933022e-02	3.059653e-02	-1.813728e-01	
PAY_6payment delayed 3 months	PAY_6payment delayed 4 months	PAY_6payment delayed 5 months	PAY_6payment delayed 6 months	PAY_6payment delayed 7 months	
-6.517594e-01	-1.591981e-01	1.383168e-01	-1.028214e+00	-5.470686e-02	
PAY_6payment delayed 8 months	PAY_6use of revolving credit	BILL_AMT1	BILL_AMT2	BILL_AMT3	
-4.602558e+00	1.848355e-01	-3.037793e-07	-8.418378e-07	-6.464863e-07	
BILL_AMT4	BILL_AMT5	BILL_AMT6	PAY_AMT1	PAY_AMT2	
-2.187557e-07	1.389749e-08	1.533201e-07	8.129814e-06	5.500552e-06	
PAY_AMT3	PAY_AMT4	PAY_AMT5	PAY_AMT6		
2.300467e-06	2.823587e-06	3.132620e-06	2.750934e-06		

Independent variables with coefficients close to zero have less importance or effect on the dependent variable. According to the ridge model, features like repayment status, education, and marital status have higher coefficients compared to other variables like payment amount and billing amount, therefore they are more important for predicting on the dependent variable. The coefficient estimates seem to make sense in real life scenarios. The ridge models indicate that single clients are less likely to default. People with families might find it harder to pay off their card balances since they also have a family to support. Repayment status helps determine the financial risks. Clients with delayed payments are often deemed as less trustworthy and involves more risk. Education is correlated with income. People with higher education can find better jobs and have more job opportunities. The estimates of the ridge model make sense from an economic perspective.

The ridge regression model accurately classified~81.66% of the testing data. The model has a slightly higher error rate when compared to the logit and probit regression but outperforms the KNN model.

3.5 Lasso Regression

Lasso regression is another shrinkage method. Lasso also uses a penalty to shrink the coefficient estimates. However, lasso performs variable selection whereas ridge does not. When $\lambda = 0$, the coefficients are OLS regression coefficients. When λ is large enough, all coefficients are set to 0. Lasso performs better when only a smaller subset of variables is relevant for the model whereas Ridge performs better when many of the predictors have predictive power. The functional form of lasso is as follows:

$$\mathcal{L} = \sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1} p\beta_j x_{ij}\right)^2 + \lambda\sum_{j=1}^{p}|\beta_j|$$

Lasso regressions shrink the coefficients of non-significant features to equal to zero. This results in smaller models that are easier to interpret. Based on the estimates in figure 6, 20 of the 80 coefficient estimates are exactly zero. This means that those variables have no effect on whether credit card payments will default. According to the lasso model, variables like payment amount, repayment status, education, and age all affect whether client's default on their credit card payments. This makes sense in a business context. Younger adults tend to have entry level jobs, so they do not make as much money as their seniors. Having a higher education allows clients to have more job opportunities. Repayment status and payment amount helps determine the credit of the client. Paying very little and delaying payments are usually indicators of financial trouble. However, the lasso model is also a bit illogical. The lasso model only found bill amount in certain months to be useful in predicting whether client's default or not instead of finding all the months important. This issue also applies to repayment status in certain months as well.

Figure 6. Coefficients of Lasso Model

LIMIT_BAL	SEXmale	EDUCATIONhigh school	EDUCATIONothers	EDUCATIONuniversity	
1.833262e-06	-1.317952e-01	0.000000e+00	7.413715e-01	-2.144408e-02	
EDUCATIONunknown	MARRIAGEothers	MARRIAGEsingle	AGE	PAY_0paid in full	
8.786389e-01	0.000000e+00	1.287915e-01	-2.965077e-03	-2.430045e-01	
PAY_0payment delayed 1 month	PAY_0payment delayed 2 month	PAY_0payment delayed 3 months	PAY_0payment delayed 4 months	PAY_0payment delayed 5 months	
-6.328563e-01	-1.860616e+00	-1.828286e+00	-1.346042e+00	-7.638013e-01	
PAY_0payment delayed 6 months	PAY_0payment delayed 7 months	PAY_0payment delayed 8 months	PAY_0use of revolving credit	PAY_2paid in full	
-2.897274e-01	-1.003132e+00	0.000000e+00	3.398330e-01	9.599316e-02	
PAY_2payment delayed 1 month	PAY_2payment delayed 2 month	PAY_2payment delayed 3 months	PAY_2payment delayed 4 months	PAY_2payment delayed 5 months	
2.962033e-01	-8.232628e-02	-1.134548e-01	3.009446e-01	-5.942216e-01	
PAY_2payment delayed 6 months	PAY_2payment delayed 7 months	PAY_2payment delayed 8 months	PAY_2use of revolving credit	PAY_3paid in full	
-5.451947e-01	-7.221375e-01	1.413121e-01	0.000000e+00	5.198394e-02	
PAY_3payment delayed 1 month	PAY_3payment delayed 2 month	PAY_3payment delayed 3 months	PAY_3payment delayed 4 months	PAY_3payment delayed 5 months	
0.000000e+00	-3.307121e-01	-2.605054e-01	0.000000e+00	1.286879e-02	
PAY_3payment delayed 6 months	PAY_3payment delayed 7 months	PAY_3payment delayed 8 months	PAY_3use of revolving credit	PAY_4paid in full	
-3.155071e-01	-5.862294e-02	0.000000e+00	0.000000e+00	6.409409e-02	
PAY_4payment delayed 1 month	PAY_4payment delayed 2 month	PAY_4payment delayed 3 months	PAY_4payment delayed 4 months	PAY_4payment delayed 5 months	
-7.185503e-01	-2.700061e-01	-1.067376e-01	-2.494520e-01	2.305335e-01	
PAY_4payment delayed 6 months	PAY_4payment delayed 7 months	PAY_4payment delayed 8 months	PAY_4use of revolving credit	PAY_5paid in full	
4.396583e-01	-2.762189e-02	1.054769e+00	0.000000e+00	7.997094e-02	
PAY_5payment delayed 2 month	PAY_5payment delayed 3 months	PAY_5payment delayed 4 months	PAY_5payment delayed 5 months	PAY_5payment delayed 6 months	
-2.749492e-01	-5.816983e-02	0.000000e+00	0.000000e+00	-7.204482e-02	
PAY_5payment delayed 7 months	PAY_5payment delayed 8 months	PAY_5use of revolving credit	PAY_6paid in full	PAY_6payment delayed 2 month	
-5.126269e-01	0.000000e+00	0.000000e+00	4.931109e-02	-1.164090e-01	
PAY_6payment delayed 3 months	PAY_6payment delayed 4 months	PAY_6payment delayed 5 months	PAY_6payment delayed 6 months	PAY_6payment delayed 7 months	
-4.914677e-01	0.000000e+00	0.000000e+00	-5.608047e-01	0.000000e+00	
PAY_6payment delayed 8 months	PAY_6use of revolving credit	BILL_AMT1	BILL_AMT2	BILL_AMT3	
-1.914082e+00	2.357446e-01	0.000000e+00	-1.373813e-06	-4.907284e-07	
BILL_AMT4	BILL_AMT5	BILL_AMT6	PAY_AMT1	PAY_AMT2	
0.000000e+00	0.000000e+00	0.000000e+00	9.530053e-06	5.986068e-06	
PAY_AMT3	PAY_AMT4	PAY_AMT5	PAY_AMT6		
1.241629e-06	1.772342e-06	2.451019e-06	1.890882e-06		

Both the lasso and ridge models have similar performance. The lasso model is slightly outperformed by the ridge model. The ridge model correctly classified .03% more data than the lasso model. The lasso model also has a higher error rate than the logistic and probit models. On the other hand, the lasso model outperforms the KNN model due to the curse of high dimensionality.

3.6 Decision Trees

Tree-based methods are commonly used for classification problems. These methods involve stratifying or segmenting the predictor space into a number of simple regions. Decision trees are made up of leaves and nodes. The leaf nodes represent the classes, the internal nodes define tests on certain attributes, and each branch represents an outcome of the test. The algorithm divides the predictor space into mutually exclusive regions, It predicts the outcome for all the observations that fall into the region by calculating the average of the outcome for the training observations in that region. Classification trees have feature selection and is easy to explain. However, they tend to be very sensitive to the choice of the

samples. In my classification tree model, I will be using the classification error rate as the criterion to make the binary splits:

$$E = 1 - max_k(\hat{p}_{mk})$$

Figure 6 shows a summary of the pruned classification tree. After pruning the classification tree, only one variable is left. The pruned classification tree model considers PAY_0 to be the only feature needed to predict whether credit card payments will default. It differentiates whether clients had no consumption, paid in full, used revolving credits, or delayed the payment for 1-8 months. The pruned classification tree only has one branch. If the clients had no consumption, paid all their balances, delayed their payments for 1 or 5 months, or used revolving credit in September 2005, then they are classified to not default on their credit card payments. On the other hand, if the clients had their payments delayed for 2,3,4,6,7, or 8 months, then the clients are classified to default on their card payments. This result does not really make sense in a business scenario. Clients that delayed their payments for 5 months are classified to not default on their payments while clients that delayed their payments for 2 months are. It would make more sense if clients that delayed their payments for 5 months are classified to default on their payments.

Figure 6: Summary of the Prune Classification Tree

```
## node), split, n, deviance, yval, (yprob)
##       * denotes terminal node
##
## 1) root 21000 22250 no ( 0.2222 0.7778 )
##   2) PAY_0: payment delayed 2 month,payment delayed 3 months,payment delayed 4 mont
hs,payment delayed 6 months,payment delayed 7 months,payment delayed 8 months 2206  26
78 yes ( 0.7044 0.2956 ) *
##   3) PAY_0: no consummption,paid in full,payment delayed 1 month,payment delayed 5
months,use of revolving credit 18794 16870 no ( 0.1656 0.8344 ) *
```

The pruned classification tree seems illogical in a business context. Based on prior and background knowledge, payment amount cannot be the sole indicator of whether credit card payments will default or not. There should be more factors that affect whether the dependent variable, but the classification tree is not considering those. The classification tree performed better than the lasso and ridge models. The lasso model accurately predicted ~81.63% of the

data while the ridge model accurately classified ~81.66% of the data. However, the classification tree performed worse than the logistic model and the probit model. The logistic model with all covariates correctly classified ~82.15% of the data and the probit model correctly classified ~81.18% of the data. The classification tree outperforms the KNN model, but that is to be expected since KNN performs badly with high dimensionality data

3.7 Bootstrapping

As mentioned previously, classification trees can be very non-robust and sensitive to the choice of sample. A small change in the data will result in a very different tree. One way to address this issue is by bootstrapping. Bootstrap can help improve the performance of classification trees by finding the average outcome of multiple trees. Averaging over multiple trees helps reduce variance. As shown in figure 7, the bootstrapped model accurately classified ~81.69% of the data with a std error of .00038 and a bias of -7.78 e^-06. The bootstrapped model has a less bias and variance. The bootstrapped model has an identical performance to the classification tree. In this case, bootstrapping did not improve the performance of the classification tree significantly. The bootstrapped model is also not very useful to help address our issue since it does not have an importance matrix feature to find significant features for predicting default credit payments.

Figure 7. Summary of Bootstrapped Classification Tree

```
##
## ORDINARY NONPARAMETRIC BOOTSTRAP
##
##
## Call:
## boot(data = cc_trn, statistic = boot_fn, R = 100, formula = default.payment.next.mo
nth ~
##     .)
##
##
## Bootstrap Statistics :
##      original        bias     std. error
## t1* 0.8168889 -7.777778e-06 0.0003810759
```

3.8 Bagging

Another method to reduce variance in classification trees is bagging. Bagging is the process of averaging over multiple bootstrap trees:

$$\hat{f}_{bag} = \frac{1}{B}\sum_{b=1}^{B} \hat{f}^{*b}(X)$$

Bagging works a little differently from other machine learning algorithms. Instead of using cross-validation, out-of-bag sampling is used for the model instead. By using the out of bag observations, the model can predict the response for the ith observation using each of the bootstrap trees. Out of bag error is the average of these B/3 predictions and is a pretty good estimate of the test error. By using the out of bag error, the model avoids fitting multiple bagging models.

The bagging model can report significant variables through the importance matrix. The model records the overall reduction in the Gini index that is due to split over a given predictor to measure the importance of a variable. The importance matrix allows the user to show the important variables and interpret the results.

Figure 8. Bagging Model Importance Matrix

```
##                  yes         no MeanDecreaseAccuracy MeanDecreaseGini
## LIMIT_BAL  19.182476  21.769037            28.891229        450.64390
## SEX        -1.924070   5.047149             3.407947         59.22184
## EDUCATION  -2.097800   7.978057             5.730147        140.95399
## MARRIAGE   -5.607034  18.567454            14.650514         76.74884
## AGE         2.049547  29.033931            27.742907        566.06804
## PAY_0     122.649122 107.575994           177.452561       1210.79138
## PAY_2      10.393669  52.093478            63.562702        264.66679
## PAY_3      11.914752  31.755646            38.836298         86.64990
## PAY_4       5.611669  32.139219            37.862323         70.66421
## PAY_5       5.933912  30.614477            37.092647         72.63727
## PAY_6      23.664795  32.777273            48.764134         92.60930
## BILL_AMT1  22.324313  19.110551            29.591640        460.98722
## BILL_AMT2 -14.815029  45.457815            48.229967        327.06956
## BILL_AMT3 -14.115849  45.819416            48.287510        314.37423
## BILL_AMT4 -18.045527  50.522630            53.044367        304.95185
## BILL_AMT5  -5.916044  41.264169            46.513121        299.52241
## BILL_AMT6   8.218083  30.039235            39.090316        326.92237
## PAY_AMT1   -7.561595  38.189996            40.256303        341.45676
## PAY_AMT2    2.565895  30.990050            35.132920        385.78308
## PAY_AMT3   10.279756  30.654742            39.223096        369.05215
## PAY_AMT4    9.426616  29.278779            36.016264        316.62259
## PAY_AMT5    3.076627  32.911603            38.254631        327.03450
## PAY_AMT6   14.792542  31.013857            40.094221        375.98113
```

The variable Pay_0 have the largest magnitude value by far. This indicates that the most recent payment amount made by the client is the most important variable at determining whether the payment will default or not. This makes sense. If a client does not pay or pays a small fraction of the amount owed, it is very likely that the client is unable to pay back what they owed. This makes the client highly at risk for defaulting. All the other variables also have some form of effect on whether payments will default as their importance matrix coefficients are not 0. Some examples of other notable predictors are the most recent bill amount, the balance limit on the account, and the first payment amount. These variables have a magnitude in the 20s, the balance limit, payment amount, and bill amount are all reasonable predictors for predicting whether payments will default. Higher bill amounts are harder to pay off, a low balance limit indicates lower income or credit score, and a low payment amount may indicate financial inability to pay back what was owed.

Bagging model accurately classified ~81.39% of the testing data. Figure 9 shows that out of 1970 clients that defaulted on their credit card payment, the bagging model only caught 723 cases. The bagging model has a false negative rate of ~63.3%. This is extremely high. Utilizing this model in real life would cause credit card companies to suffer a lot of losses as the model is bad at predicting positive cases. On the other hand, the model has a false positive rate of ~6.1%. The model performs decently at predicting negative cases. This means that the model will occasionally turn away good clients for credit card applications. The bagging model have a higher error rate than the lasso/ridge model. The ridge model misclassified 18.34% of the testing data while the lasso model misclassified 18.36% of the testing data. The bagging model misclassified 18.61% of the data, slightly higher than the error rate of the lasso and ridge models. The bagging model also have a higher error rate than the logistic model, the probit model, the classification tree model, and the bootstrap tree model. However, it did manage to perform better than the KNN model which had an error rate of 23.5%.

Figure 9. Performance of Bagging Model

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  yes   no
##        yes  723  428
##        no  1247 6602
##
##                Accuracy : 0.8139
##                  95% CI : (0.8057, 0.8219)
##     No Information Rate : 0.7811
##     P-Value [Acc > NIR] : 1.001e-14
##
##                   Kappa : 0.36
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.36701
##             Specificity : 0.93912
##          Pos Pred Value : 0.62815
##          Neg Pred Value : 0.84113
##              Prevalence : 0.21889
##          Detection Rate : 0.08033
##    Detection Prevalence : 0.12789
##       Balanced Accuracy : 0.65306
##
##        'Positive' Class : yes
##
```

3.8 Random Forest

Random forest is very similar to bagging. However, random forest selects a random subset of predictors instead of all predictors each time the model splits the tree. Random forest addresses some issues with the bagging model. Bagging does not work very well when there are strong predictors in the model along with moderate predictors. The strong predictors will be on top of its split in every tree and averaging over correlated trees does not reduce variance by much. Random forest avoids this issue by forcing each split to use a subset of predictors to decorrelate trees.

Using the random forest model, I tuned the number of predictors considered at each split to find the model with the lowest out of bag error rate. As seen in figure 10, the Random Forest model and the bagging model have similar importance matrices. Both consider the

variable Pay_0 to be the most important since it have the largest magnitude value. If a client does not pay or pays a small fraction of the amount owed, it is very likely that the client is unable to pay back what they owed. This makes the client highly at risk for defaulting. All the other variables also have some effect on whether payments will default as their importance matrix coefficients are not 0. Similarly, both models found variables like bill amount, balance limit, and payment amount to be significant at predicting whether credit card payments will default.

Figure 10. Random Forest Importance Matrix

```
##                 yes        no MeanDecreaseAccuracy MeanDecreaseGini
## LIMIT_BAL  19.048573 18.164629            25.133451        377.32416
## SEX         1.063040  5.551208             5.457726         66.07302
## EDUCATION  -1.808091  4.206595             2.733439        134.53591
## MARRIAGE   -2.785210 13.964114            10.914620         80.22669
## AGE         5.392862 19.735666            20.683092        425.49711
## PAY_0     103.912997 75.587913           129.723502        782.09900
## PAY_2      23.150866 28.130808            44.014582        317.19962
## PAY_3      15.870843 28.249104            38.955594        214.21010
## PAY_4      11.963017 26.702784            34.309408        166.77636
## PAY_5      10.712658 25.823804            34.345785        151.18392
## PAY_6      17.454556 26.402356            37.122580        139.00681
## BILL_AMT1  20.778633 21.220066            35.643596        433.79039
## BILL_AMT2  -5.388692 40.888934            45.442897        381.48234
## BILL_AMT3  -4.033535 40.897509            45.546185        367.16818
## BILL_AMT4  -7.299518 42.303178            49.360381        355.13977
## BILL_AMT5   1.722944 38.177107            44.981289        354.15409
## BILL_AMT6   5.070563 30.441358            38.552123        351.32450
## PAY_AMT1    5.623695 28.896515            33.309876        359.07839
## PAY_AMT2    7.386301 29.902001            36.570264        350.33819
## PAY_AMT3    7.818866 31.791731            40.467335        332.42509
## PAY_AMT4    6.025205 32.049914            38.446073        308.65597
## PAY_AMT5    4.250880 27.320017            32.390339        307.50921
## PAY_AMT6   11.655749 26.209279            32.302279        333.23011
```

As shown in figure 11, the random forest model accurately classified ~81.73% of the testing data. Out of 1970 clients that defaulted on their credit card payment, the Random Forest model only caught 724 cases. The Random Forest model has a false negative rate of ~63.25%. This is also extremely high. Like the bagging model, utilizing the Random Forest model in real life would cause credit card companies to suffer huge losses as the model is bad at accurately predicting positive cases. On the other hand, the model has a false positive rate of ~5.7%. The model performs decently at predicting negative cases. This means that the

model will occasionally turn away good clients for credit card applications. Like the bagging model, the Random Forest is bad at predicting positive cases accurately but predicts the negative cases quite well.

The Random Forest model performed better than Lasso, Ridge, and Bagging models. The Ridge model accurately classified ~81.66 percent of the test data while the lasso accurately classified ~81.63%. The bagging model accurately classified ~81.39% of the testing data. The Random Forest model accurately classified ~81.73 percent of the testing data which is higher than the correct classification of the models listed previously. However, the Random Forest model's performance does not exceed the logistic and probit regression models which correctly classified ~82% of the testing data.

Figure 11. Summary of Random Forest Model

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  yes   no
##        yes  725  408
##        no  1245 6622
##
##                Accuracy : 0.8163
##                  95% CI : (0.8082, 0.8243)
##     No Information Rate : 0.7811
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.3659
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.36802
##             Specificity : 0.94196
##          Pos Pred Value : 0.63989
##          Neg Pred Value : 0.84174
##              Prevalence : 0.21889
##          Detection Rate : 0.08056
##    Detection Prevalence : 0.12589
##       Balanced Accuracy : 0.65499
##
##        'Positive' Class : yes
##
```

3.9 Boosting

Boosting is an ensemble method that helps improve model predictions and converts weak learners to strong learners. Boosting can be applied to tree-based methods as well. Unlike bagging and random forest, trees are grown sequentially in boosting and does not involve bootstrap sampling. Boosting models are computationally intensive since it learns slowly. In each period, the model slightly improves the previous estimation. The functional form of boosting with slow learning is as follows:

$$Y = r_0$$

$$Y = \lambda \hat{f}_1(x) + r_1$$

$$Y = \lambda \hat{f}_1(x) + \lambda \hat{f}_2(x) + r_2$$

.

.

.

$$Y = \lambda \hat{f}_1(x) + \lambda \hat{f}_2(x) + \lambda \hat{f}_B(x) + r_B$$

The boosting model also have an importance matrix as shown in figure 12. Like the bagging and random forest model, the boosting model also considers PAY_0 to be the most important variable. PAY_0 have the highest relative influence out of all the predictors. In addition, the boosting model also consider all other predictors to influence whether credit card payments will default. However, some variables have less of an effect when compared to others. An example of this would be the variables MARRIAGE with a relative influence of 0.7498291 and SEX with a relative influence of 0.4501672 . The relative influence of these variables is close to 0 which indicates that the effect of these independent variables does not have a strong impact on the dependent variable. This seems reasonable. As society gets more and more progressive, we see gender equality in more places. More and more women are paid about the same as their male counterparts. Marriage also does not have a strong impact on defaulting. This may be due to different tax brackets for singles and couples or other

factors. However, that is another issue. For the most part, variables like payment amount, bill amount, limit balance, age have similar relative influences. All of these are useful for predicting whether clients will default on their credit card payments. As stated previously, these variables do make sense in real life scenarios.

Figure 12. Bagging Model Importance Matrix

	var <fctr>	rel.inf <dbl>
PAY_0	PAY_0	52.7461878
PAY_2	PAY_2	8.3637947
PAY_3	PAY_3	4.1688143
LIMIT_BAL	LIMIT_BAL	3.7268911
PAY_5	PAY_5	3.5658320
BILL_AMT1	BILL_AMT1	3.1099675
PAY_6	PAY_6	2.8997372
PAY_AMT3	PAY_AMT3	2.4803079
PAY_AMT1	PAY_AMT1	2.3649345
PAY_4	PAY_4	2.3277529
1-10 of 23 rows	Previous 1 2 3 Next	

Figure 13 shows that the boosting model accurately classified ~82.26% of the testing data. Out of 1065 clients that defaulted on their credit card payment, the boosting model missed 346 cases. The boosting model has a false negative rate of ~32.49%. While this is not as high as the bagging/Random Forest model, it would still induce losses from the defaulting clients that the model missed. On the other hand, the model has a false positive rate of ~15.77%. Unlike the random forest and bagging models, the boosting model is slightly worse at predicting negative cases. This means that the model will sometimes turn away good clients for credit card applications. When compared to the Random Forest model, it appears that the boosting model is slightly better at predicting positive cases but slightly worse at predicting negative cases. The boosting model have the lowest error rate out of all the models ran so far. The boosting model accurately classified 82.26% of the data. The logistic model

accurately predicted ~82.15% and probit model accurately predicted ~82.18%. The lasso model and ridge model both accurately predicted ~81.6% of the data. The random forest correctly classified ~81.73% of the testing data and the bagging model correctly classified ~81.33%. The boosting model classified most testing cases correctly and have the lowest error rate thus far.

Figure 13. Performance of Boosting Model

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   no  yes
##        no  6689  341
##        yes 1252  718
##
##               Accuracy : 0.823
##                 95% CI : (0.815, 0.8308)
##     No Information Rate : 0.8823
##     P-Value [Acc > NIR] : 1
##
##                  Kappa : 0.379
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.67800
##             Specificity : 0.84234
##          Pos Pred Value : 0.36447
##          Neg Pred Value : 0.95149
##              Prevalence : 0.11767
##          Detection Rate : 0.07978
##    Detection Prevalence : 0.21889
##       Balanced Accuracy : 0.76017
##
##        'Positive' Class : yes
##
```

3.10 XGBoost

XGBoost is a decision tree based ensemble machine learning algorithm that uses a gradient boosting framework. It have very good computational speed and prediction performance on structured or tabular data. The objective function of XGBoost is as follows:

$$obj = \sum_{i=1}^{n} l\left(y_i, \hat{y}_{i_1}^{t} + f_t(x_t)\right) + rT + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2$$

The higher the gain, the more effect the predictor has on the dependent variable. Like the previous models, the XGBoost model also found the variable PAY_0 to be the most important variable for predicting whether credit card payments will default or not. Variables with relative importance close to 0 does not impact the dependent variable significantly. The XGBoost model found the older payment amount variables and older bill amounts to be not as important as the more recent counterparts. This seems reasonable since recent information are more up to date and reflects the client's situation better than information in the past. According to the XGBoost model, recent payment amounts, bill amounts, limit balance, and age are good indicators of credit card payment defaults.

Figure 13. Coefficient Estimates of XGBoost Model

Feature <chr>	Gain <dbl>	Cover <dbl>
PAY_0payment delayed 2 month	0.1963003233	3.076860e-02
BILL_AMT1	0.0581180306	6.146263e-02
PAY_AMT2	0.0544573373	5.675712e-02
PAY_AMT1	0.0540706557	7.011576e-02
PAY_2payment delayed 2 month	0.0480671988	8.614280e-03
LIMIT_BAL	0.0452079397	3.589075e-02
PAY_AMT3	0.0447751944	5.441775e-02
BILL_AMT2	0.0407288235	8.821172e-02
AGE	0.0380825701	1.949846e-02
BILL_AMT3	0.0352121531	5.987515e-02

1-10 of 53 rows

XGBoost model accurately classified ~81.13% of the testing data. Out of 1970 clients that defaulted on their credit card payment, the bagging model only caught 689 cases. The XGBoost model has a false negative rate of ~65%. This is also extremely high. Utilizing this model in real life would cause credit card companies to suffer huge losses as the model is bad at predicting positive cases. On the other hand, the model has a false positive rate of ~5.9%. The model performs decently at predicting negative cases. This means that the model will occasionally turn away good clients for credit card applications. The XGBoost have a higher error rate than the Lasso, Ridge, Bagging, boosting, and random forest models. The bagging and XGBoost models both have similar test error rates. Both correctly classified ~81.13% of the data. The boosting model accurately classified 82.26% of the data. The logistic model accurately predicted ~82.15% and probit model accurately predicted ~82.18%. The lasso

model and ridge model both accurately predicted ~81.6% of the data. The random forest correctly classified ~81.73% of the testing data.

Figure 14. Performanceof XGBoost Model

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0  689  417
##          1 1281 6613
##
##                Accuracy : 0.8113
##                  95% CI : (0.8031, 0.8194)
##     No Information Rate : 0.7811
##     P-Value [Acc > NIR] : 9.506e-13
##
##                   Kappa : 0.3449
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.34975
##             Specificity : 0.94068
##          Pos Pred Value : 0.62297
##          Neg Pred Value : 0.83772
##              Prevalence : 0.21889
##          Detection Rate : 0.07656
##    Detection Prevalence : 0.12289
##       Balanced Accuracy : 0.64521
##
##        'Positive' Class : 0
##
```

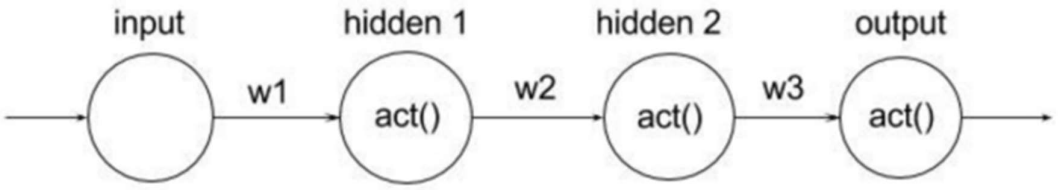
3.11 Neural Network

Neural networks are a set of algorithms designed to recognize patterns, modeled loosely after the human brain. It interprets sensory data through a perceptron,labeling, or clustering raw input. Neural networks can help cluster and classify data by recognizing numerical patterns.The data goes through the network layer by layer until it produces an output. Each neuron in the neural network produces an output or activation.. The neural network trains the model by adjusting the weights that connect the neurons. The purpose of my neural network is to find

$$Y = f\left(g\left(\sigma(\ldots(X))\right)\right)$$

where $g(.), \sigma(.)$ can be nonlinear or linear functions of their inputs. The downside of neural networks is that they do not have feature selection. The results from a neural network are very hard to interpret in 'human' terms.

Figure 15. Diagram of a Neural Network



There are two hidden dense layers in my neural network. Each hidden dense layer has 64 nodes. Since the problem is a classification problem, I used the softmax activation function in all my layers. The softmax takes a input vector of K real numbers and normalizes it into a probability distribution consisting of K probabilities, proportional to the exponentials of the input numbers. After applying the softmax function, each component will be in the interval (0,1) and all the components will add up to 1. This is similar to the objective function of the logistic regression. The softmax function is as follows:

$$g_k(T) = \frac{e^{T_k}}{1 + e^{T_k}}$$

The neural network fits the model by finding unknown parameters known as weights. In order to find the values of these weights, I will be using the categorical cross-entropy function whichfinds the deviance. The categorical cross-entropy function is as follows:

$$R(\theta) = -\sum_{k=1}^{K}\sum_{i=1}^{N} y_{ik}\, log f_k(x_i)$$

In order to optimize the model, I will be using root mean square propagation. The function form of RMSprop is as follows:

$$E[g^2]_t = \beta E[g^2]_{t-1} + (1-\beta)\left(\frac{\delta C}{\delta w}\right)^2$$

$$w_t = w_{t-1} - \frac{n}{\sqrt{E[g^2]_t}}\frac{\delta C}{\delta w}$$

My neural network misclassified ~22.2% of the testing data. The testing dataset was randomly sampled. Due to the imbalanced proportion of classes in the whole dataset, the testing dataset have no positive cases for the model to predict on. Therefore, we do not know how well the model can predict on positive cases. Since the misclassification rate of positive cases is unknown, we should not use this model in real life. Since this dataset is imbalanced, there are more negative cases than positive cases. As a result, the model is not given enough positive samples to be trained to accurately detect whether clients will default on their credit card payments. The neural network model slightly outperformed the KNN model. The KNN model suffers from the curse of high dimensionality and is still the worse performing model by far. The KNN model accurately classified ~76.5% of the data while the neural network only classified ~77.8% of the testing data. However, the other models from the previous section except for KNN all outperformed the neural network model and have a higher accuracy than the neural network model.

Figure 16. Neural Network Confusion Matrix

```
##    test_class
##         0
##   0 7010
##   1 1991
```

Due to not having feature selection, the neural network does not provide any insights on the significant factors for predicting default payments. The downside of neural networks is that it is hard to interpret in meaningful ways. Because of this, I only know how well the neural network classify default payments for this data set.

3.12 Results

Table 3. Performance of All Models

Model	Best Tune	Accuracy
Logistic Regression	NA	0.8214776
Probit Regression	NA	0.8217621
KNN	K = 9	0.7649999
Ridge Regression	λ = 0.0149514301573017	0.8165556
Lasso Regression	λ = 0.0014271864520944	0.8163333
Classification Tree	Terminal Nodes = 2	0.8168889
Bootstrapped Tree	NA	0.8168889
Bagging	mtry = 24	0.8138889
Random Forest	mtry = 5	0.8163333
Boosting	number of trees = 100, and interaction depth = 4	0.8230000
Tuned Boosting	shrinkage = .05, number of trees = 80, interaction depth = 4	0.8182222
XGBoost	Number of rounds = 100	0.8113333
Neural Network	number of epochs = 21	0.7788023

4 Conclusion

The goal of this project is to find significant factors for predicting default credit card payments. Based off the models I ran so far, the logistic regression model classified most of the testing data correctly. Probit regression also performed similarly to the logistic regression. On the other hand, K's nearest neighbor performed very badly on the data. This may be due to the large number of features in the dataset. Both logistic regression and probit regression model found features that had a significant effect on the dependent variable. Such variables include payment statuses during certain months, payment amount in the first few months, gender, education status, and marriage status. However, some of the results are illogical. The models claim that clients owe more to the card companies, the odds of defaulting increases. This makes sense since people with high payment amounts are usually bad with money management and live a lifestyle they cannot sustain. However, the model also says that delayed payments reduce the chance of defaulting. It will make more sense if delayed payments increase the chance of defaulting since it indicates that the client cannot financially pay back the amount owed. Different models should be trained to find better relationships between X's and Y.

After running classification trees, ridge regression, lasso regression, and training a neural network, the KNN model still have the highest error rate. The KNN model is very likely to suffer from the curse of high dimensionality. The neural network slightly outperforms the KNN model. The neural network model classified 77.9% of the testing data accurately while the KNN model only accurately classified ~76.5% of the test data. It appears that there is not enough positive cases in the dataset to train the neural network better. The performance of the neural network on positive cases is also unknown. In addition, the neural network model also suffers from a lack of feature detection. The downside of neural networks is that is hard to comprehend in human languages. Humans do not know how the network is classifying the

data while other models have some sort of feature selection or variable coefficients. Therefore, the neural network and the KNN model both does not seem to be a good fit for our dataset. On the other hand, both ridge and lasso models have identical performance with ridge classifying slightly more data correctly. The classification tree classified have a lower error rate than the ridge and lasso model. However, pruning and bootstrapping the tree did not yield a lower error rate. These new models can still not beat the logistic and probit regression model which managed to correctly classify ~82% of the data. While these models have the lowest error rate, it is important to note that some of the coefficient estimates of the models are illogical in a real-world scenario. On the other hand, the coefficient estimates from the ridge and lasso model seems to be more reasonable and provides a better context in the real world. In addition, the bagging, random forest, XGBoost, and the tuned boosting model also have similar error rates. These models managed to classify ~81% of the testing data correctly. However, the untuned boosting model with 100 trees and 4 interaction depth managed to slightly outperform the logistic and probit regression models. The boosting model accurately predicted ~82.39% of the data. The boosting model classified the most cases correctly by far.

Boosting, the best performing model, found PAY_O, the most recent payment made by the client, to be the most important variable at predicting whether credit card payments will default. The model also found other recent payments and bill amount to have a significant effect on the dependent variable. The model did not find older payments and bill amounts to have a big impact on defaulting. This makes sense because financial information from the past does not provide as much relevant information from financial information in the present especially when things can change over time. The model also found limit balance to be significant in predicting default payments. Clients with a good credit score or a high income tend to have higher balance limits in their credit accounts. Out of the demographic variables (age, sex, marriage, and education), the model only found age and education to have a strong

impact on default payments. This also makes sense. Gender wage gap is slowly decreasing, and more women are working today. Therefore, both genders have access to well-paying jobs. Younger people generally have more entry level positions which pays less than the senior positions or younger people might have student loans. Therefore, age does influence whether credit card payments will default or not. As for education, people that have graduate degrees tend to have more opportunities for higher paying jobs. The importance matrix of the boosting model seems reasonable and makes sense in real world context.

It would also be useful to compare the results to other regions. Demographics, cultures, and behaviors is very likely to be different in other regions. Therefore, it is important to see if these factors are significant in other places as well. Future research may involve the usage of different data sets originating from different countries from different continents.

5 Reference

Islam, Sheikh Rabiul., William Eberle, and Sheikh Khaled Ghafoor. “Credit Default Mining Using Combined Machine Learning and Heuristic Approach.” arXiv preprint arXiv:1807.01176(2018).

Lu, Hongya, Haifeng Wang, and Sang Won Yoon. “Real Time Credit Card Default Classification Using Adaptive Boosting-Based Online Learning Algorithms.” IIE Annual Conference. Proceedings. Institute of Industrial and Systems Engineers (IISE), 2017.

“Default of Credit Card Clients Dataset | Kaggle”. [online]. Available: https://www.kaggle.com/uciml/default-of-credit-card-clients-dataset. Accessed: February 10, 2020.