Credit Card Default Classification

MSDS 498: Capstone

Kevin Johnson

Table of Contents

I. Model Development Guide

1. Introduction

Through its lending business, banks seek to maximize profit by considering the return on each of its loans. To accomplish this, they must lend to clients at a high volume while also monitoring who is able to repay their loans. Two major trade-offs occur in bank lending: the tradeoff between risk and return and the tradeoff between profit and volume. Though banks must increase lending volume, the failure for clients to repay loans can result in substantial losses (Thomas, 2009, p. 12-13). For banks to be able to confidently lend to clients, they must have a reliable system in place for predicting defaults. One such system can be designed using predictive modeling. To predict client defaults, an analysis was performed on the *default of credit card clients* data set. Exploratory data analysis was performed, including feature engineering, variable based analysis, and computational analysis. Variables were then selected to build four different models: Random Forest, Gradient Boosting, Logistic Regression, and Naïve Bayes. The results of these four models were then analyzed and compared.

2. Data Description

The *default of credit card clients* data set was provided by I-Cheng Yeh and Che-hui Len of Chung-Hua University and Thompson Rivers University, respectively, for their 2009 paper *The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients*. The data set was provided by a Taiwan bank and contains credit card information of customers from April to September 2005. The data contain 30,000 observations and 24 variables. Of the 24 variables, 10 are factor variables and 14 are numerical. The full description of the data is shown in Table 1.

Table 1: Data Description

Name	Data Type	Measurement	Description
LIMIT_BAL	Numerical	NT dollar	Amount of the given credit
SEX	Factor	1 (Male), (Female)	Gender designation
EDUCATION	Factor	1 (Graduate School), 2 (University), 3 (High School), 4 (Other)	Education level of credit card holder
MARRIAGE	Factor	1 (Married), 2 (Single), 3 (Other)	Marital status of credit card holder
AGE	Numerical	Year	Age of credit card holder
PAY_0	Factor	-1 (Pay Duly) to 9 (Payment delay for nine or more months)	Monthly payment record September 2005
PAY_2	Factor	-1 (Pay Duly) to 9 (Payment delay for nine or more months)	Monthly payment record August 2005
PAY_3	Factor	-1 (Pay Duly) to 9 (Payment delay for nine or more months)	Monthly payment record July 2005
PAY_4	Factor	-1 (Pay Duly) to 9 (Payment delay for nine or more months)	Monthly payment record June 2005
PAY_5	Factor	-1 (Pay Duly) to 9 (Payment delay for nine or more months)	Monthly payment record May 2005
PAY_6	Factor	-1 (Pay Duly) to 9 (Payment delay for nine or more months)	Monthly payment record April 2005
BILL_AMT1	Numerical	NT dollar	Amount on Bill Statement September 2005
BILL_AMT2	Numerical	NT dollar	Amount on Bill Statement August 2005
BILL_AMT3	Numerical	NT dollar	Amount on Bill Statement July 2005
BILL_AMT4	Numerical	NT dollar	Amount on Bill Statement June 2005
BILL_AMT5	Numerical	NT dollar	Amount on Bill Statement May 2005
BILL_AMT6	Numerical	NT dollar	Amount on Bill Statement April 2005
PAY_AMT1	Numerical	NT dollar	Amount of payment September 2005
PAY_AMT2	Numerical	NT dollar	Amount of payment August 2005
PAY_AMT3	Numerical	NT dollar	Amount of payment July 2005
PAY_AMT4	Numerical	NT dollar	Amount of payment June 2005
PAY_AMT5	Numerical	NT dollar	Amount of payment May 2005
PAY_AMT6	Numerical	NT dollar	Amount of payment April 2005
DEFAULT	Factor	0 (No Default) , 1 (Default)	* Response variable - Default payment designation

The response variable for this analysis is Default for the following month. Default is indicated with a value of 1, while non-default is given a value of 0. In the entire data set, there are 6,636 instances of default, or 22.12% of the observations. The breakdown of the response variable is shown in Table 2.

Table 2: Response Variable

Default	Frequency	Percentage
0	23,364	77.88%
1	6,636	22.12%

2.1 Data Quality Check

A quick check of the data reveals a handful of issues. Most notably, the variables Education and Marriage contain values outside the data dictionary. While the dictionary for Education indicates values of 1,2,3, and 4, some observations also include values of 0, 5, and 6. Since these cannot be verified, they will instead be moved into the 'Other' category and be given a value of 4, as indicated in Table 3.

Table 3: Education Data Cleaning

Education (Dirty)			Education (Clean)		
Education	**Frequency**	**Percentage**	**Education**	**Frequency**	**Percentage**
0 (n/a)	14	0.05%	0 (n/a)	-	-
1 (Graduate School)	10,585	35.28%	1 (Graduate School)	10,585	35.28%
2 (University)	14,030	46.77%	2 (University)	14,030	46.77%
3 (High School)	4,917	16.39%	3 (High School)	4,917	16.39%
4 (Other)	123	0.41%	4 (Other)	468	1.56%
5 (n/a)	280	0.93%	5 (n/a)	-	-
6 (n/a)	51	0.17%	6 (n/a)	-	-

Similarly, while the dictionary shows values of 1, 2, and 3 for Marriage, some observations include a value of 0. These will also be moved to the 'Other' category and given a value of 3. The new values are shown in Table 4.

Table 4: Marriage Data Cleaning

Marriage (Dirty)			Marriage (Clean)		
Marriage	**Frequency**	**Percentage**	**Marriage**	**Frequency**	**Percentage**
0 (n/a)	54	0.18%	0 (n/a)	-	-
1 (Married)	13,659	45.53%	1 (Married)	13,659	45.53%
2 (Single)	15,964	53.21%	2 (Single)	15,964	53.21%
3 (Other)	323	1.08%	3 (Other)	377	1.26%

Next issue that needs to be addressed appears in the six Pay_n variables. The data dictionary mentions a value of -1 for paid duly, 1 for one month late on payments, and follows that format up to 9 for nine months late on payments. However, a large amount of observations contain values of -2 and 0. Further research indicates the definitions provided by the data dictionaries are not complete. The definitions for these values are shown in Table 5. In addition, the PAY_0 column appears to be mislabeled. This will be renamed from PAY_0 to PAY_1.

Table 5: Definitions for Pay_n variables

Value	Description
-2	No consumption
-1	Paid in full
0	Payment with use of revolving credit
1	Payment delay for one month
2	Payment delay for one month
3	Payment delay for two months
4	Payment delay for three months
5	Payment delay for five months
6	Payment delay for six months
7	Payment delay for seven months
8	Payment delay for eight months

The final piece of data cleaning to be performed is with respect to the BILL_AMT columns. In some observations, there are instances where the payment amount exceeds the balance amount, resulting in a negative balance. These negative balances run the risk of skewing the models. To fix this, the BILL_AMT columns will be rescaled so that 0 is the minimum value for each column. For one more task, the BILL_AMT columns will be renamed. In its original form, the BILL_AMT columns show the outstanding balance for the end of each month and the following month's PAY_AMT will pay for that balance. For example, BILL_AMT2 will be paid off by PAY_AMT1. To make the balance/payment a little easier to follow, the BILL_AMT columns will be moved up by 1 value, so BILL_AMT2 will now be BILL_AMT1, etc. The final balance will now be BILL_AMT0, and the next month's payment will be expected to pay off this amount.

2.2 Train/Test/Validation Split

Before performing further analysis on the data set, the data will be split into three different sets. The training set, where the analysis will be performed, will contain 50% of the observations, while the test set, where the models will be checked, will contain 25% of the observations. A validation set will also be created containing 25% of the observations. The validation set will be set aside for use after the models have been developed. A full breakout of the three datasets is shown in Table 6.

Table 6: Train/Test/Validation Data Split

Data Set	Observations	Percentage
Train	15,180	50.60%
Test	7,323	24.41%
Validate	7,497	24.99%

3. Feature Engineering

In its original form, the *default of credit card clients* data set may be too simple to create meaningful models. To remedy this, feature engineering will be performed by creating additional variables and performing weight-of-evidence binning.

3.1 Utilization Rate

The first variable that will be created is the utilization rate. The utilization rate measures the percentage of the client's credit limit that is currently used up with their balance. This rate is measured on a 0 to 100 scale. A client with no balance will have a utilization rate of 0, while a client whose balance is their entire credit limit will have a utilization rate of 100. In theory, clients with a lower utilization rate will likely have a better credit score.

Equation 1: Utilization Rate

$$UTL_n = \frac{BILL\_AMT_n}{LIMIT\_BAL} \text{ x } 100$$

3.2 Payment Rate

The second variable to be created is the payment rate. The payment rate measures the percentage of the balance a client pays off. This rate is measured on a 0 to 100 scale. A client that pays off their entire balance will have a payment rate of 100, while a client who does not make any payment will have a payment rate of 0. In the event a client has no balance and thus no payment, the payment rate is given a value of 100.

Equation 2: Payment Rate

$$PAY\_RAT_n = \frac{PAY\_AMT_n}{BILL\_AMT_n} \text{ x } 100$$

3.3 Moving Averages

Values in the dataset show the individual values for each month. To synthesize this and observe how each client behaves over the entire period, moving averages will be created for bill amount, payment amount, utilization rate, and pay rate. The simple moving average will take the average value over the entire period. In addition, a weighted moving average will also be calculated. The weighted moving average differs from the simple moving average in that it places a heavier emphasis on the more recent months. The theory behind the weighted moving average is that a clients' more recent behavior will have a stronger predictive value for how they will behave in the short term.

Equation 3: Simple Moving Averages

$$BILL\_AMT\_SMA = \frac{BILL\_AMT_0 + BILL\_AMT_1 + BILL\_AMT_2 + BILL\_AMT_3 + BILL\_AMT_4 + BILL\_AMT_5}{6}$$

$$PAY\_AMT\_SMA = \frac{PAY\_AMT_1 + PAY\_AMT_2 + PAY\_AMT_3 + PAY\_AMT_4 + PAY\_AMT_5}{5}$$

$$PAY\_RAT\_SMA = \frac{PAY\_RAT_1 + PAY\_RAT_2 + PAY\_RAT_3 + PAY\_RAT_4 + PAY\_RAT_5}{5}$$

$$UTL\_SMA = \frac{UTL_0 + UTL_1 + UTL_2 + UTL_3 + UTL_4 + UTL_5}{6}$$

Equation 4: Weighted Moving Averages

$$BILL\_AMT\_WMA = \frac{6 * BILL\_AMT_0 + 5 * BILL\_AMT_1 + 4 * BILL\_AMT_2 + 3 * BILL\_AMT_3 + 2 * BILL\_AMT_4 + 1 * BILL\_AMT_5}{6 + 5 + 4 + 3 + 2 + 1}$$

$$PAY\_AMT\_WMA = \frac{5 * PAY\_AMT_1 + 4 * PAY\_AMT_2 + 3 * PAY\_AMT_3 + 2 * PAY\_AMT_4 + 1 * PAY\_AMT_5}{5 + 4 + 3 + 2 + 1}$$

$$PAY\_RAT\_WMA = \frac{5 * PAY\_RAT_1 + 4 * PAY\_RAT_2 + 3 * PAY\_RAT_3 + 2 * PAY\_RAT_4 + 1 * PAY\_RAT_5}{5 + 4 + 3 + 2 + 1}$$

$$UTL\_WMA = \frac{6 * UTL_0 + 5 * UTL_1 + 4 * UTL_2 + 3 * UTL_3 + 2 * UTL_4 + 1 * UTL_5}{6+5+4+3+2+1}$$

3.4 Change over Previous Months

The final set of variables to be created will observe how the final month of values differ from previous months. The intention behind these variables is to observe whether changes in the final months' bill results in a higher probability of default. For analysis purposes, the changes will be compared to the previous month, simple moving average, and weighted moving average. One item to note, for the bill change and utilization rate change moving averages, the final month is not included in the moving average number. Instead, the current month will be compared to the averages of the previous five months.

Equation 5: Change over Previous Months

$$BILL\_CHANGE\_1M = \frac{BILL\_AMT_0}{BILL\_AMT_1}$$

$$BILL\_CHANGE\_SMA = \frac{BILL\_AMT_0}{BILL\_AMT\_SMA}$$

$$BILL\_CHANGE\_WMA = \frac{BILL\_AMT_0}{BILL\_AMT\_WMA}$$

$$UTL\_CHANGE\_1M = UTL_0 - UTL_1$$

$$UTL\_CHANGE\_SMA = UTL_0 - UTL\_SMA$$

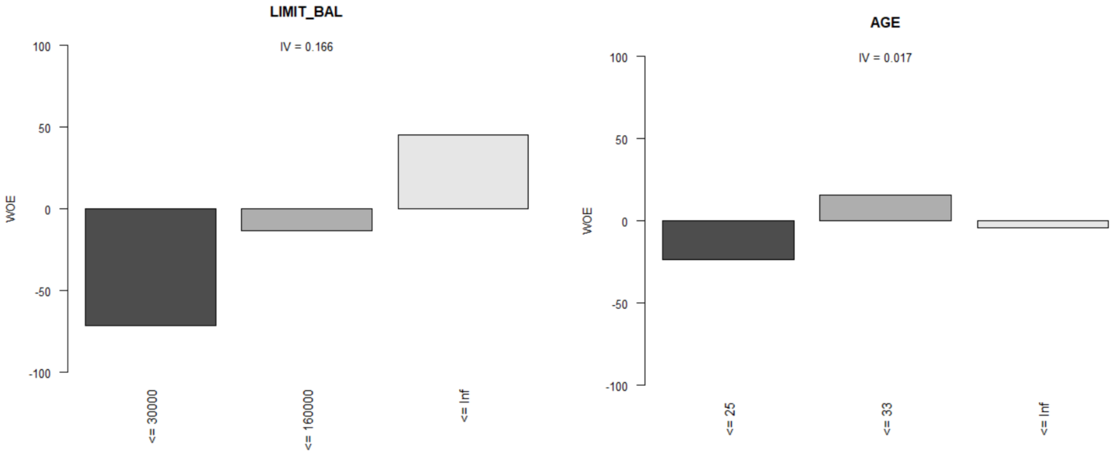
$$UTL\_CHANGE\_WMA = UTL_0 - UTL\_WMA$$

$$PAY\_RAT\_1M = PAY\_RAT_0 - PAY\_RAT_1$$

3.5 WOE Binning

Weight-of-evidence binning is used to transform a continuous variable into groups based on the similarity to a target variable. In terms of credit default, WOE binning is used to separate the paying customers and the defaulting customers. For this analysis, WOE binning is performed on both Age and Limit Balance. The results of each are in Figure 1.

Figure 1: WOE Binning



The usefulness of WOE binning is measured by an information value score. As a general rule, information values of .30 to .50 are indicative of strong predictive power, .10 to .30 indicate medium predictive power, and values less than .10 are viewed as having weak predictive power. The information value for Limit Balance is .166, while the value for Age is .017. For further analysis, both variables will be observed in their binned form.

3.6 Full Dataset

Following the completion of feature engineering, the data now contain 50 variables. A breakdown of the full data set is show in Table 7.

Table 7: Full Dataset

Name	Data Type	Measurement	Description
DEFAULT	Factor	0 (No Default) , 1 (Default)	* Response variable - Default payment designation
LIMIT_BAL.binned	Factor	NT dollar	Amount of the given credit
AGE.binned	Factor	Year	Age of credit card holder
SEX	Factor	1 (Male), (Female)	Gender designation
EDUCATION	Factor	1 (Grad School), 2 (University), 3 (High School), 4 (Other)	Education level of credit card holder
MARRIAGE	Factor	1 (Married), 2 (Single), 3 (Other)	Marital status of credit card holder
PAY_1	Factor	-2 (No Balance) to 8 (Payment delay for eight or more months)	Monthly payment record September 2005
PAY_2	Factor	-2 (No Balance) to 8 (Payment delay for eight or more months)	Monthly payment record August 2005
PAY_3	Factor	-2 (No Balance) to 8 (Payment delay for eight or more months)	Monthly payment record July 2005
PAY_4	Factor	-2 (No Balance) to 8 (Payment delay for eight or more months)	Monthly payment record June 2005
PAY_5	Factor	-2 (No Balance) to 8 (Payment delay for eight or more months)	Monthly payment record May 2005
PAY_6	Factor	-2 (No Balance) to 8 (Payment delay for eight or more months)	Monthly payment record April 2005
BILL_AMT0	Numerical	NT dollar	Amount on Bill Statement September 2005, EOM
BILL_AMT1	Numerical	NT dollar	Amount on Bill Statement August 2005, EOM
BILL_AMT2	Numerical	NT dollar	Amount on Bill Statement July 2005, EOM
BILL_AMT3	Numerical	NT dollar	Amount on Bill Statement June 2005, EOM
BILL_AMT4	Numerical	NT dollar	Amount on Bill Statement May 2005, EOM
BILL_AMT5	Numerical	NT dollar	Amount on Bill Statement April 2005, EOM
PAY_AMT1	Numerical	NT dollar	Amount of payment September 2005
PAY_AMT2	Numerical	NT dollar	Amount of payment August 2005
PAY_AMT3	Numerical	NT dollar	Amount of payment July 2005
PAY_AMT4	Numerical	NT dollar	Amount of payment June 2005
PAY_AMT5	Numerical	NT dollar	Amount of payment May 2005
PAY_AMT6	Numerical	NT dollar	Amount of payment April 2005
UTL0	Numerical	Percentage	Utilization Rate September 2005, EOM
UTL1	Numerical	Percentage	Utilization Rate August 2005, EOM
UTL2	Numerical	Percentage	Utilization Rate July 2005, EOM
UTL3	Numerical	Percentage	Utilization Rate June 2005, EOM
UTL4	Numerical	Percentage	Utilization Rate May 2005, EOM
UTL5	Numerical	Percentage	Utilization Rate April 2005, EOM
PAY_RAT1	Numerical	Percentage	Payment Rate September 2005
PAY_RAT2	Numerical	Percentage	Payment Rate August 2005
PAY_RAT3	Numerical	Percentage	Payment Rate July 2005
PAY_RAT4	Numerical	Percentage	Payment Rate June 2005
PAY_RAT5	Numerical	Percentage	Payment Rate May 2005
BILL_SMA	Numerical	NT dollar	Simple average of Bill Statement, April to August 2005
BILL_WMA	Numerical	NT dollar	Weighted average of Bill Statement, April to August 2005
BILL_CHANGE_1M	Numerical	Percentage	Percent change in Bill Statement between September and August 2005
BILL_CHANGE_SMA	Numerical	Percentage	Percent change in September 2005 Bill Statement over Simple Average of Bill Statement
BILL_CHANGE_WMA	Numerical	Percentage	Percent change in September 2005 Bill Statement over Weighted Average of Bill Statement
PAY_AMT_SMA	Numerical	NT dollar	Simple average of Payments, May to September 2005
PAY_AMT_WMA	Numerical	NT dollar	Weighted average of Payments, May to September 2005
UTL_SMA	Numerical	Percentage	Simple average of Utilization Rate, April to August 2005
UTL_WMA	Numerical	Percentage	Weighted average of Utilization Rate, April to August 2005
UTL_CHANGE_1M	Numerical	Change in Percentage	Difference in Utilization Rate between September and August 2005
UTL_CHANGE_SMA	Numerical	Change in Percentage	Difference in September 2005 Utilization Rate and Simple Average of Utilization Rate
UTL_CHANGE_WMA	Numerical	Change in Percentage	Difference in September 2005 Utilization Rate and Weighted Average of Utilization Rate
PAY_RAT_1M	Numerical	Change in Percentage	Difference in Payment Rate between September and August 2005
PAY_RAT_SMA	Numerical	Percentage	Simple average of Payment Rate, May to September
PAY_RAT_WMA	Numerical	Percentage	Weighted average of Payment Rate, May to September

4. Exploratory Data Analysis

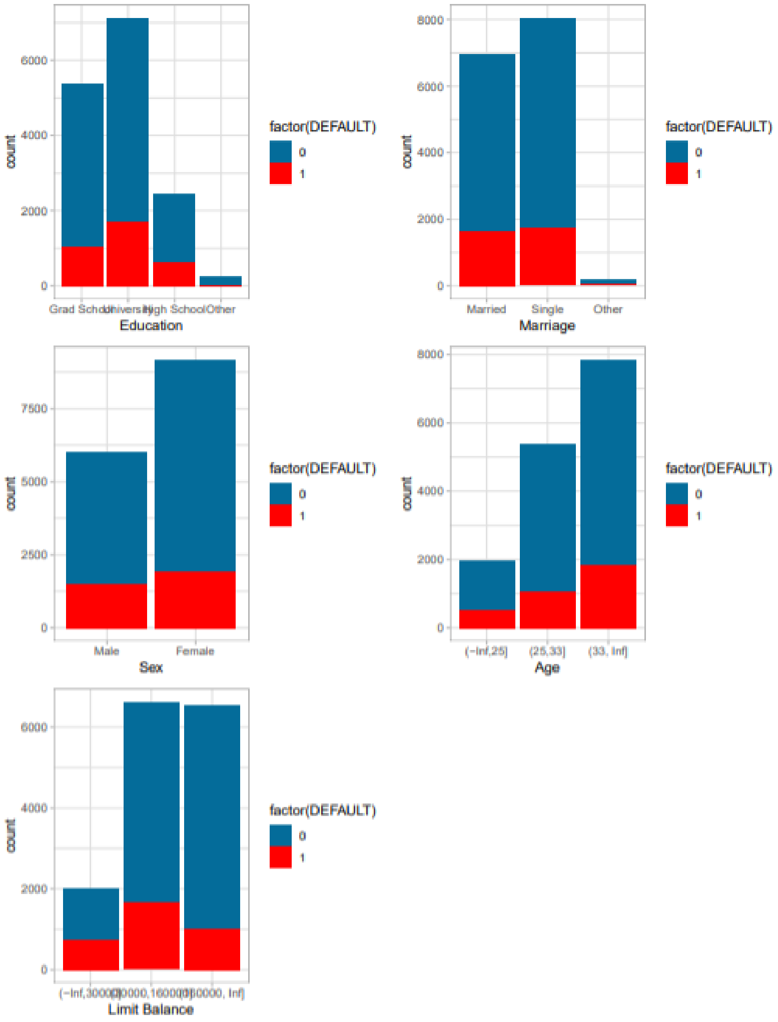
4.1 Variable Based EDA

4.1.1. Factor Variables

The five factor variables in the dataset are Education, Marriage, Sex, binned Age, and binned Limit Balance. Bar charts for each factor variable are shown in Figure 1. Some notable factors that have higher than normal default rates are people with ages from 21 to 25, who default at a rate of 26.88%. Meanwhile, those whose highest education is High School have a default rate of

26.07%. For comparison, the default rate of the training data set is 22.55%. The factor variable with the most significant default rate between each category is Limit Balance. Borrowers whose limit is 30,000 or under have a default rate of 37.23%. The borrowers who fall in the 31,000 to 160,000 range default at a 24.94% rate, which is slightly above average. Meanwhile, borrowers with a limit above 160,000 default at a rate of only 15.59%.

Figure 2: Factor Variables



Next, some analysis is done comparing how each factor variable interacts with each other. Across the board, those with a Limit Balance of 30,000 or less default at a much higher rate than the rest of the group. In addition, individuals in the 21-25 Age group who are also married have a larger default rate. The interaction default rates for each factor variable are shown in Table 8.

Table 8: Interaction of Factor Variables

	L: <30K	L: 30K-160K	L: 160K+	S: Male	S: Female	M: Married	M: Single	M: Other	E: Grad	E: College	E: High	E: Other
A: 21-25	34.35%	24.12%	14.94%	28.82%	26.08%	32.88%	26.21%	11.11%	23.33%	28.73%	29.33%	3.70%
A: 26-33	35.91%	23.52%	13.26%	20.94%	19.25%	22.73%	18.98%	12.50%	17.50%	21.26%	26.41%	9.33%
A: 33+	40.28%	26.36%	17.11%	26.61%	20.77%	23.90%	21.77%	24.54%	20.69%	24.45%	25.59%	9.16%
L: <30,000				38.82%	35.73%	41.96%	34.57%	33.33%	34.22%	37.36%	40.08%	13.64%
L: 30,000-160,000				27.26%	23.53%	26.98%	23.47%	21.43%	23.22%	25.72%	26.35%	13.27%
L: 160,000+				17.09%	14.65%	17.47%	13.66%	9.76%	15.80%	15.86%	15.64%	3.54%
S: Male						27.08%	25.33%	26.44%	20.87%	27.09%	28.98%	11.76%
S: Female						22.06%	20.14%	18.80%	18.62%	22.20%	24.11%	6.76%
M: Married									20.98%	24.72%	27.36%	7.69%
M: Single									18.77%	23.35%	24.79%	9.65%
M: Other									20.69%	26.14%	18.82%	0.00%

4.1.2. Numerical Variables

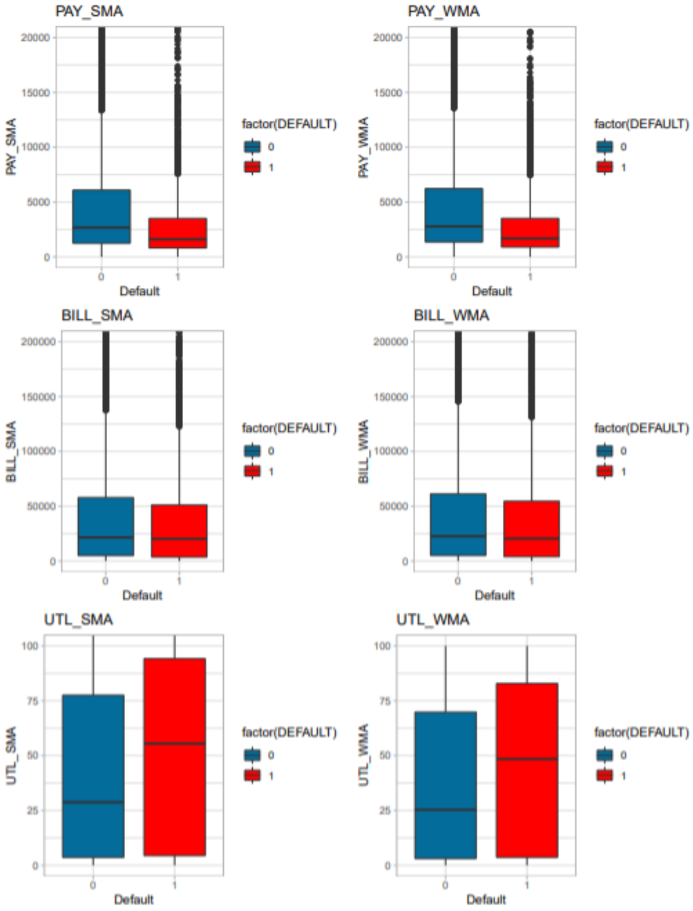
Due to the wide range in values for each numerical variable, traditional graphics for exploratory data analysis were not useful for every variable. Instead, a table of default rates for numerical variables was created. One interesting takeaway is that clients who experienced a positive change in their utilization rate had a smaller rate of default, while those who experienced no change had a much higher rate. Meanwhile, clients averaged higher payment amounts each month had a lower rate of default.

Table 9: Default Rates for Numerical Variables

	DEFAULT RATES		
Variable	**Neg. Change**	**No Change**	**Pos. Change**
UTL_CHANGE_1M	22.72%	32.59%	18.96%
UTL_CHANGE_SMA	24.87%	39.39%	18.93%
UTL_CHANGE_WMA	24.95%	38.54%	18.81%
PAY_RAT_1M	23.45%	19.41%	24.51%
	<= 3.5%	**3.5%<n<=25%**	**>25%**
PAY_RAT_SMA	47.30%	24.64%	17.75%
PAY_RAT_WMA	48.16%	24.48%	17.50%
	<= 2,000	**2,000<n<=11,500**	**>11,500**
PAY_AMT_SMA	29.71%	18.24%	10.18%
PAY_AMT_WMA	30.08%	18.25%	10.30%
	<= 1,000	**1,000<n<=10,000**	**>10,000**
BILL_SMA	28.66%	19.62%	22.43%
BILL_WMA	28.63%	19.96%	22.33%
	<= 1.0%	**1.0%<n<=50%**	**>50%**
UTL_SMA	25.25%	16.06%	28.35%
UTL_WMA	24.95%	16.69%	28.54%

One additional observation in observing the default rates of each numerical variable is the lower default rates for clients who fall in the middle range for payment amount, bill amount, and utilization rate. One possible explanation for this is that clients who fall in this range are largely more active users of their line of credit, and thus are likely to be paying closer attention to their balance each month.

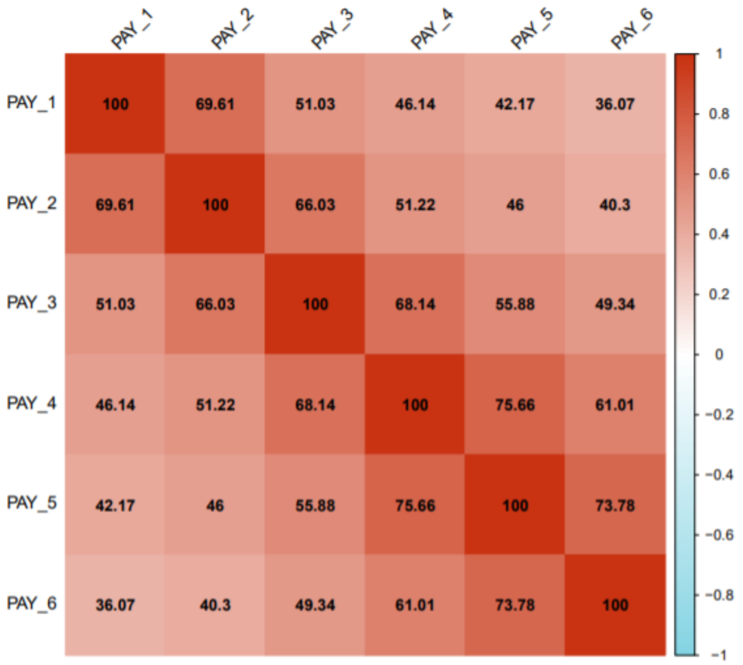
Figure 3: Numerical Variables



4.1.3. Principal Components Analysis

After some analysis, the six PAY_n variables have shown to be important predictors for client default. However, these six variables have high correlations among themselves. Multicollinearity within predictor variables can be a problem because it lowers the statistical significance of each variable. To remedy this, principal components analysis is performed.

Figure 4: PAY_n Correlation Plot



	PAY_1	PAY_2	PAY_3	PAY_4	PAY_5	PAY_6
PAY_1	100	69.61	51.03	46.14	42.17	36.07
PAY_2	69.61	100	66.03	51.22	46	40.3
PAY_3	51.03	66.03	100	68.14	55.88	49.34
PAY_4	46.14	51.22	68.14	100	75.66	61.01
PAY_5	42.17	46	55.88	75.66	100	73.78
PAY_6	36.07	40.3	49.34	61.01	73.78	100

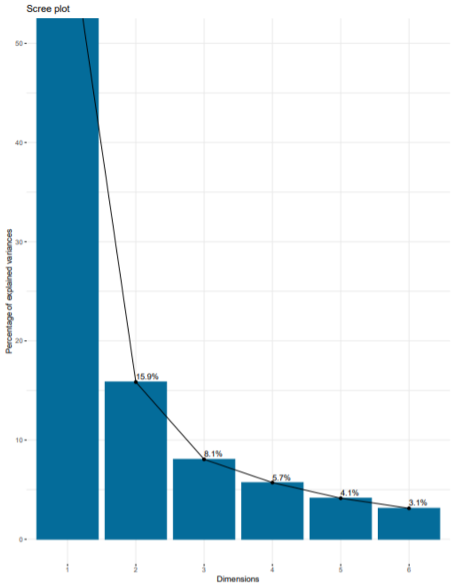
With highly correlated variables, principal components can be used for summarizing the set with a smaller number of variables that explain most of the variability within the original set. In other words, principal components analysis is a technique for transforming a set of variables into a new set of variables which are uncorrelated with one another. The usefulness of the transformed variables stems from their proportion of variance explained. The first principal component is the normalized linear combination of the features that has the largest variance. The second principal component is the linear combination of the features that has the maximal variance out of all the linear combinations that uncorrelated with the first principal component (Dunn & Everitt, 2001, p. 49).

Table 10: PCA Results

PC	Eigenvalue	Variance %	Cumulative Variance %
Dim1	3.7870	63.12%	63.12%
Dim2	0.9514	15.86%	78.97%
Dim3	0.4832	8.05%	87.03%
Dim4	0.3427	5.71%	92.74%
Dim5	0.2483	4.14%	96.88%
Dim6	0.1874	3.12%	100.00%

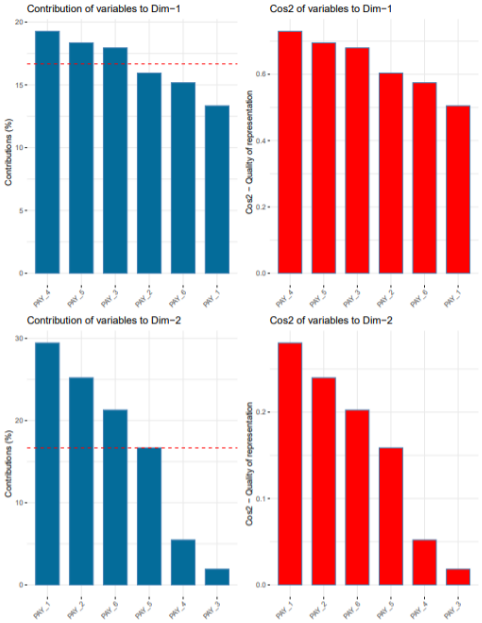
There are typically two different criteria that can be used when selecting principal components: eigenvalues or proportion of variance explained. Using eigenvalues, a value that is greater than 1 indicates that principal components accounts for more variance than accounted by one of the variables in the standardized data. Using proportion of variance explained, a predetermined percentage of variance is used as the cutoff point (Dunn & Everitt, 2001, p. 53). In this analysis, the first principal component has an eigenvalue of 3.7870 and accounts for 63.12% of the variance. The second component has an eigenvalue of 0.9514 and accounts for 15.86% of the variance. These two components combined account for 78.97% of the variance. Using a cutoff point of 70%, the PAY_n variables can be reduced from six to two variables while still accounting for over 70% of the variance. The percentage of variance explained can be more easily observed in the scree plot in Figure 5.

Figure 5: Scree Plot



In Figure 6, the contribution and quality of representation of each variable to the two dimensions is shown.

Figure 6: Contribution and cos2 of Dim 1 and Dim 2



4.2 Model Based EDA

4.2.1. One Rule Model

The first model to be run for exploratory data analysis is the One Rule Model. The One Rule Model is a simple model that determines one rule for each predictor variable and then selects the variable with the lowest error rate as its one rule. For this data, the *Dim 1* variable is shown to be the most useful with the rule that values of 0 and 1 result in no default, while values greater than 1 will default.

In addition to identifying the most useful variable, the One Rule Model can also provide a baseline score for future models. Applying the one rule for Dim 1 results in 12,142 correct predictions out of the 15,180 observations, or an error rate of 20.08%. However, since the model is meant to predict defaults correctly, the error rate may not be the most useful metric for evaluating a model. Instead, for evaluating the predictive accuracy on imbalanced data sets, metrics such as Sensitivity, False Positive Rate, and Balanced Accuracy will be more useful. The One Rule Model yields a Sensitivity of .36, False Positive Rate of .07 (Type I Error), and Balanced Accuracy of .645 ([TP+TN]/2). The full breakout of classification metrics is shown in Table 11. Equations for each metric can be found in Appendix A.

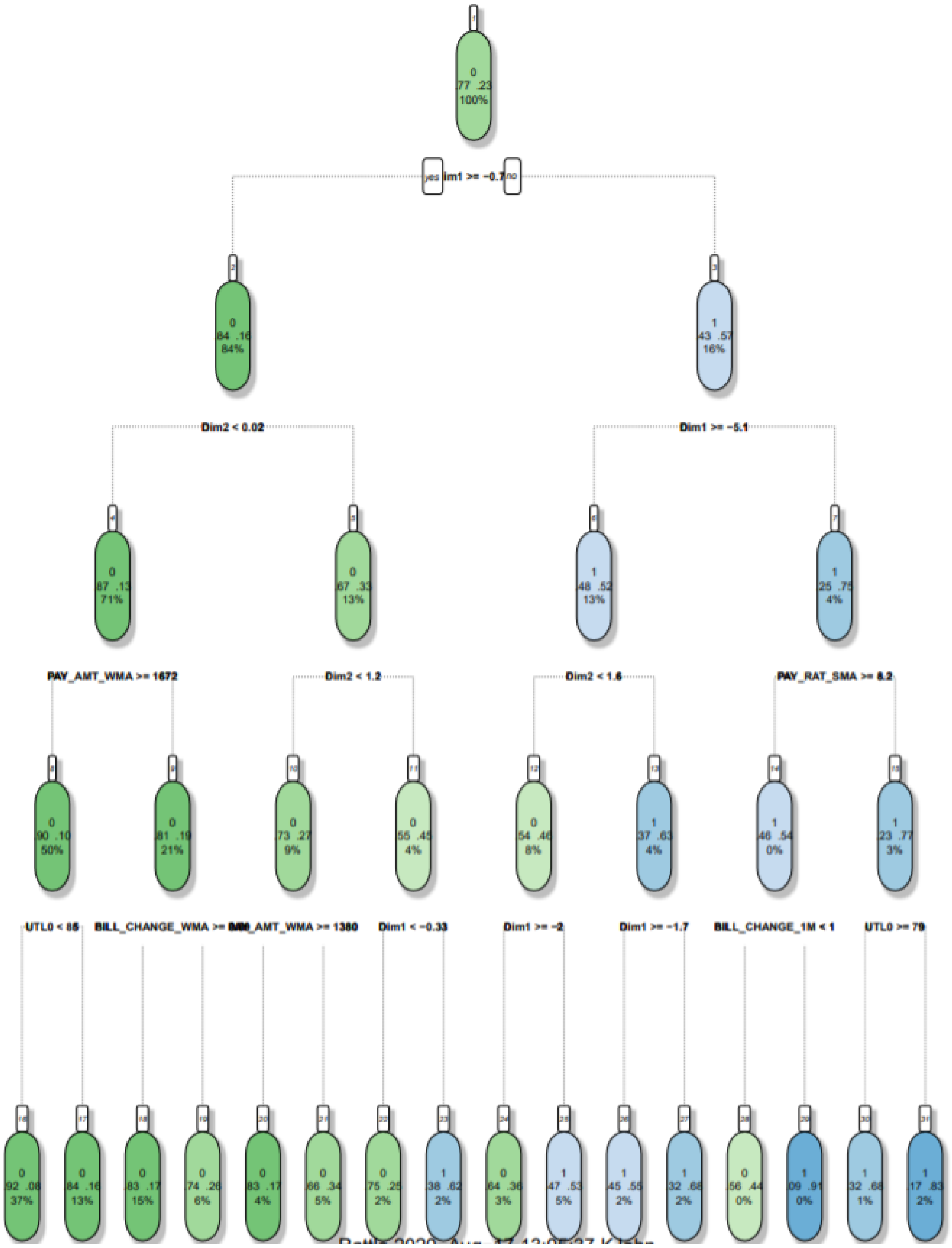
Table 11: One Rule Model

Computational EDA: One Rule Model													
Actual Class	Predicted Class		Totals		Actual Class	Predicted Class		TP	0.36	TP+TN	1.29	AUC	0.64
	0	1				0	1	TN	0.93	Precision	0.59	Sensitivity	0.36
0	10,907	850	11,757		0	0.93	0.07	Type I Error	0.07	Recall	0.36	Specificity	0.93
1	2,198	1,225	3,423		1	0.64	0.36	Type II Error	0.64	F1	0.50		

4.2.2. Decision Tree

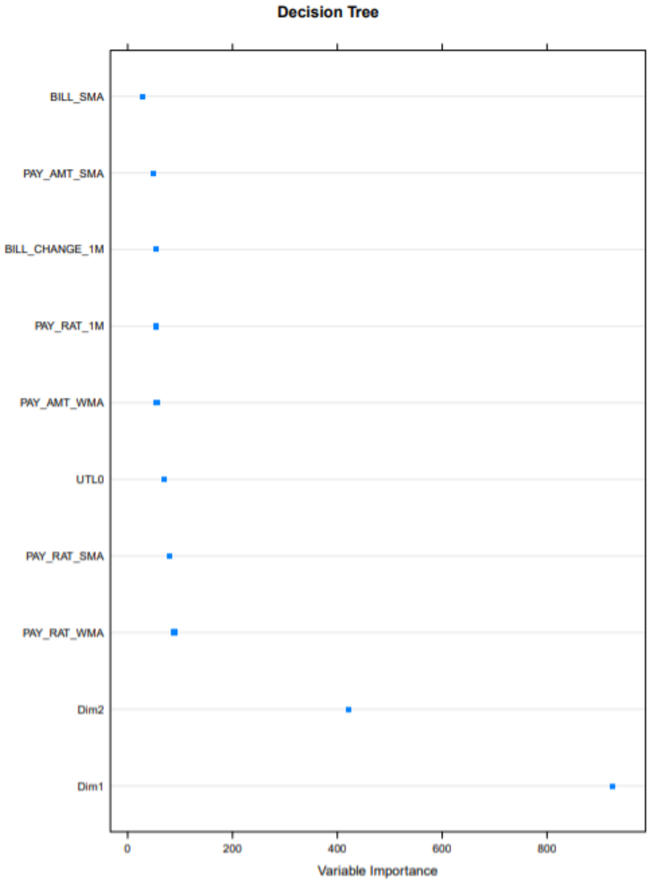
The second computational EDA model to be run is a decision tree. A decision tree is a greedy algorithm that works by iteratively splitting the data into distinct subsets. In classification, the splits in the data are selected to minimize the impurity of the resulting subsets. This purity score is measured using a value called the Gini index, which is defined by a measure of total variance across *K* classes (James et. al, 2013, p. 312). For exploratory data analysis, a decision tree can be used by observing where the splits in the data occur. The dendrogram for the decision tree is shown in Figure 7.

Figure 7: Decision Tree Dendrogram



In addition to observing the splits in each node, the variable importance for the decision tree can be observed. For this tree, Dim1 and Dim2 were by far the most important variables, while the moving averages for pay rate were also of importance. The variable importance plot can be observed in Figure 8.

Figure 8: Decision Tree Variable Importance

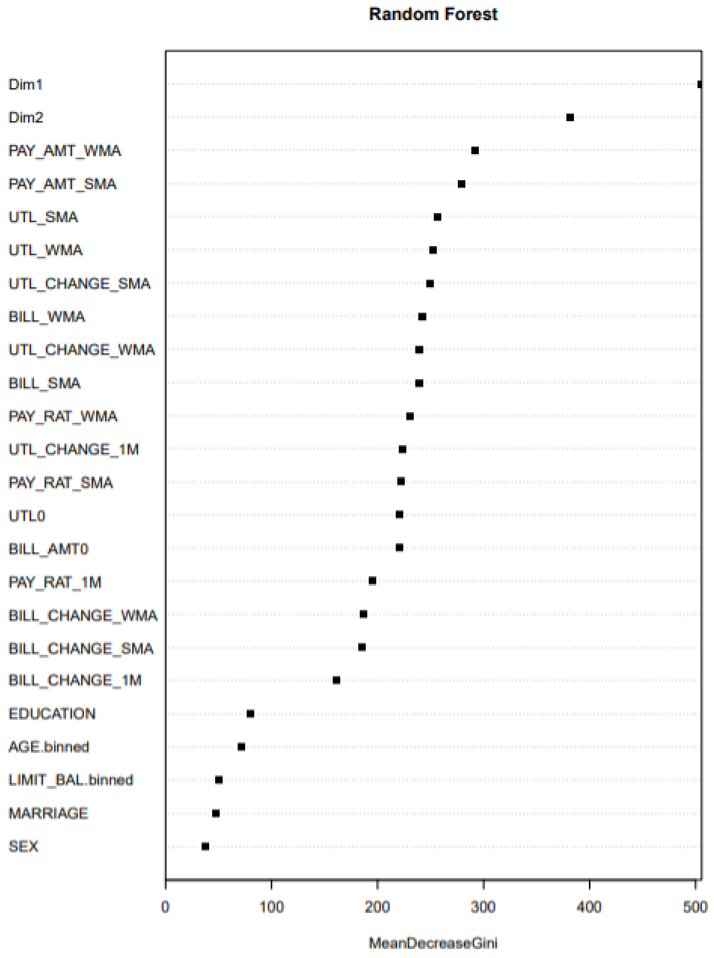


4.2.3. Random Forest

The third computational EDA model to be run is a Random Forest model. Random forest models are an ensemble method that creates many decision trees and then averages the results. The decision trees in a random forest are generated using bootstrapping, where sampling with replacement is performed on the data set. Bootstrapping the data set and then averaging the results improves upon the decision tree by searching for the best features among many random subsets of features. This results in a larger diversity of features and reduces the variance in the model (James et. al, 2013, p. 317).

Just like a decision tree, variance importance plots can be created for random forest models, which is shown in Figure 9. Once again, Dim 1 and Dim 2 are the most important variables, followed by the moving average for payment amount and utilization rate.

Figure 9: Random Forest Variable Importance



4.2.4. Backward Selection Logistic Regression

The final computational EDA model to be run is a backward selection logistic regression model. The backward selection logistic regression is an automated variable selection model that uses stepwise selection to select the best subset of predictor variables. Using backwards selection, the model begins with the full dataset and then works backwards removing the least useful predictor one at a time. The metric used for selecting which variables to remove is AIC, or Akaike information criterion. AIC measures the amount of information loss from simplifying the model. Using backward selection, the ideal model will be simplified without losing much performance. The summary of the full model along with the p-values of each variable is shown in Table 12.

Table 12: Backward Selection Logistic Regression

	Dependent variable:
	DEFAULT
SEXFemale	-0.019*** p = 0.004
EDUCATIONUniversity	-0.0004 p = 0.955
EDUCATIONHigh School	-0.004 p = 0.654
EDUCATIONOther	-0.093*** p = 0.0003
MARRIAGESingle	-0.024*** p = 0.001
MARRIAGEOther	-0.035 p = 0.199
BILL_AMT0	0.00000*** p = 0.0005
UTL0	1.800* p = 0.079
PAY_AMT_SMA	-0.00000*** p = 0.000
UTL_CHANGE_1M	-0.001** p = 0.046
UTL_SMA	-1.500* p = 0.079
UTL_CHANGE_SMA	-1.500 *p = 0.079
PAY_RAT_SMA	0.001*** p = 0.00000
PAY_RAT_1M	0.0002 p = 0.130
LIMIT_BAL.binned(30000,160000]	-0.052***p = 0.00001
LIMIT_BAL.binned(160000, Inf]	-0.106*** p = 0.000
AGE.binned(25,33]	-0.012 p = 0.255
AGE.binned(33, Inf]	0.003 p = 0.767
Dim1	-0.077*** p = 0.000
Dim2	0.056*** p = 0.000
Constant	0.281*** p = 0.000
Observations	15,180
Log Likelihood	-6,867.000
Akaike Inf. Crit.	13,775.000

4.3 Variable Selection

Before moving on to the model building section, the size of the data set has been decreased. Ultimately, 15 predictor variables were chosen for the final models. One item to note is that since weighted moving average and simple moving average have similar values, only one of these values were selected. In addition, all five factor variables were included in the model build. The full list of variables is in Table 13.

Table 13: Variable Selection

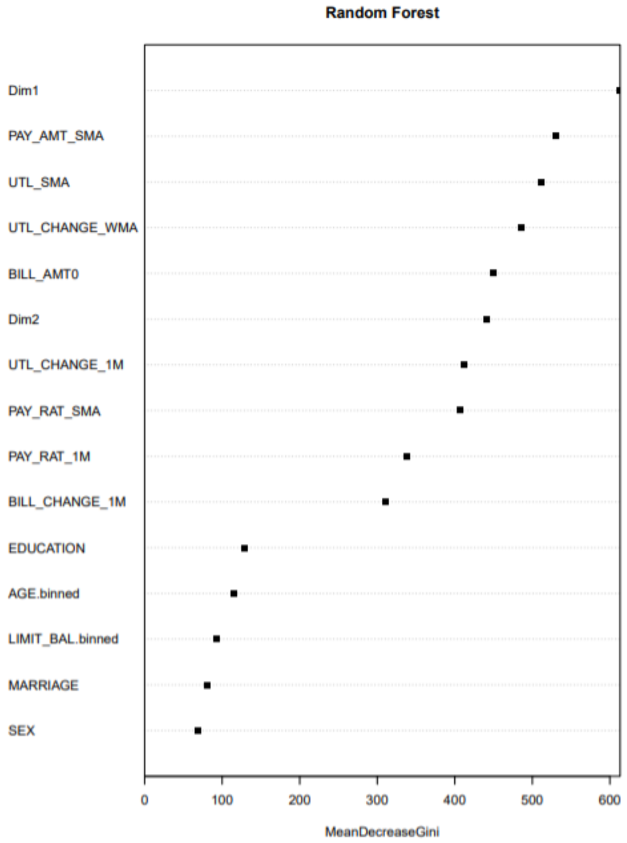
Variable Name
DEFAULT (response)
AGE.binned
BILL_AMT0
BILL CHANGE 1M
Dim1
Dim2
EDUCATION
LIMIT_BAL.binned
MARRIAGE
PAY_AMT_SMA
PAY_RAT_SMA
PAY_RAT_1M
SEX
UTL_CHANGE_1M
UTL_CHANGE_WMA
UTL_SMA

5. Model Building

5.1 Model 1: Random Forest

Model 1 is a Random Forest model. This model was run with the same parameters as the one in Section 4.2.3, with the one difference being the reduced data set with selected variables. As can be seen in the Variable Importance plot, Dim 1 is the most important predictor, with PAY_AMT_SMA, UTL_SMA, and UTL_CHANGE_WMA also of higher relative importance.

Figure 10: Model 1 Variable Importance



The Random Forest model performed similarly on both the training and test sets, with an AUC of .65 in both the training and test sets. Model 1 also had a Sensitivity of .36, False Positive Rate of .06, and Balanced Accuracy of .650 in the training set, compared to .37, .07, and .655 in the test sets. The full results for Model 1 are shown in Table 14.

Table 14: Model 1 Results

Model 1: Random Forest (Training)													
Actual Class	Predicted Class		Totals		Actual Class	Predicted Class		TP	0.36	TP+TN	1.30	AUC	0.65
	0	1				0	1	TN	0.94	Precision	0.64	Sensitivity	0.36
0	11,070	687	11,757		0	0.94	0.06	Type I Error	0.06	Recall	0.36	Specificity	0.94
1	2,188	1,235	3,423		1	0.64	0.36	Type II Error	0.64	F1	0.51		

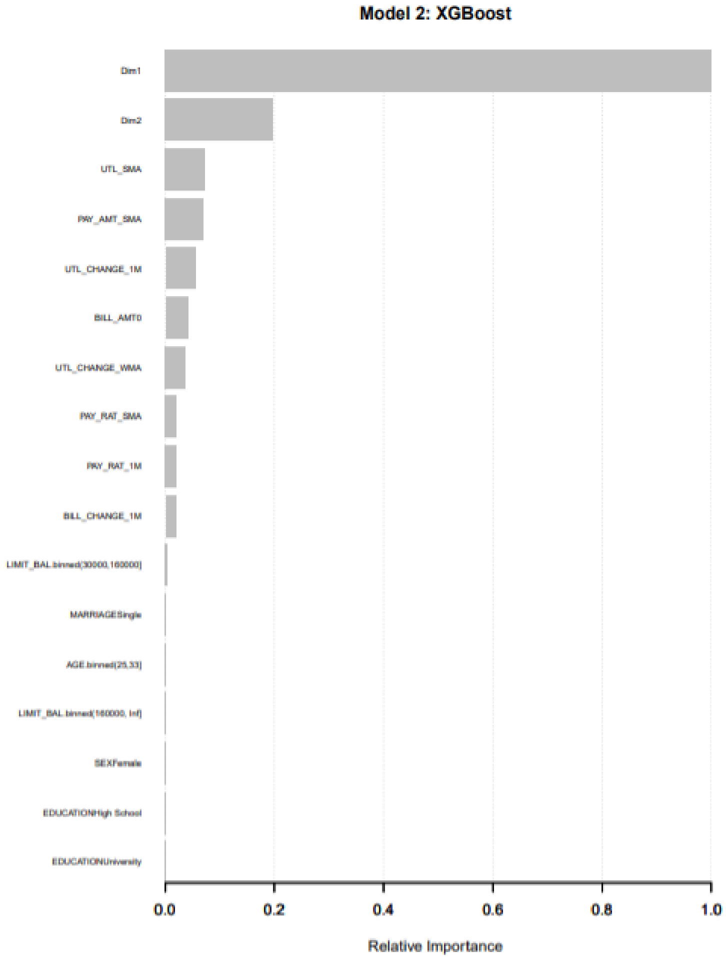
<table>
<tr><th colspan="14">Model 1: Random Forest (Test)</th></tr>
<tr><td rowspan="2">Actual Class</td><td colspan="2">Predicted Class</td><td rowspan="2">Totals</td><td rowspan="4"></td><td rowspan="2">Actual Class</td><td colspan="2">Predicted Class</td><td>TP</td><td>0.37</td><td>TP+TN</td><td>1.31</td><td>AUC</td><td>0.65</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>1</td><td>TN</td><td>0.93</td><td>Precision</td><td>0.61</td><td>Sensitivity</td><td>0.37</td></tr>
<tr><td>0</td><td>5,388</td><td>378</td><td>5,766</td><td>0</td><td>0.93</td><td>0.07</td><td>Type I Error</td><td>0.07</td><td>Recall</td><td>0.37</td><td>Specificity</td><td>0.93</td></tr>
<tr><td>1</td><td>977</td><td>580</td><td>1,557</td><td>1</td><td>0.63</td><td>0.37</td><td>Type II Error</td><td>0.63</td><td>F1</td><td>0.52</td><td></td><td></td></tr>
</table>

5.2 Gradient Boosting

Model 2 is a Gradient Boosting model. Gradient boosting models are similar to random forests, in that they combine decision trees into a single predictive model. The difference in gradient boosting is that the trees in a boosting model are grown sequentially, where each tree is grown using the information from previous trees. Boosting does not involve sampling but instead each tree is fit on a modified version of the dataset. By learning sequentially, a boosting model can be more beneficial than a bagging model due to it decreasing bias over time. However, it can also be more prone to overfitting. Boosting models have three tuning parameters: the shrinkage parameter (learning rate), the number of splits in each tree (max_depth), and the number of trees (rounds) (James et. al, 2013, p. 321-322). The gradient boosting method used for Model 2 is XGBoost. The parameters selected for this model were a learning rate of .001, max_depth of 6, and was run at 600 rounds.

Figure 16 shows the variable importance plot for Model 2. Dim 1 had the highest relative importance of every variable by a wide amount. Other more important variables for this model were Dim 2, UTL_SMA, PAY_AMT_SMA, and UTL_CHANGE_1M. Of note, the factor variables appeared to have little importance for the XGBoost model.

Figure 11: Model 2 Variable Importance



One important consideration for performing gradient boosting is that it can be easy to overfit the training data based on the parameters of the model. To help prevent this, hyperparameter tuning was performed using both grid search and some trial and error. Model 2 performed well on the both training set and test set, with an AUC of .81 and .78, respectively. On the training set, Model 2 had a Sensitivity of .65, False Positive Rate of .17, and Balanced Accuracy of .735. Full results for Model 2 are shown in Table 15.

Compared to the Random Forest model in Model 1, the Gradient Boosting model identified true negatives at nearly the double the rate. However, there appears to be a tradeoff, as the False Positive Rate more than doubled as well. As defaults are more costly to a bank, identifying true positives would be considered the higher priority for this analysis. A further study that may be necessary would be a cost-benefit analysis to try to identify if there is a point that rejecting too many false positives would begin to lose the bank money.

Table 15: Model 2 Results

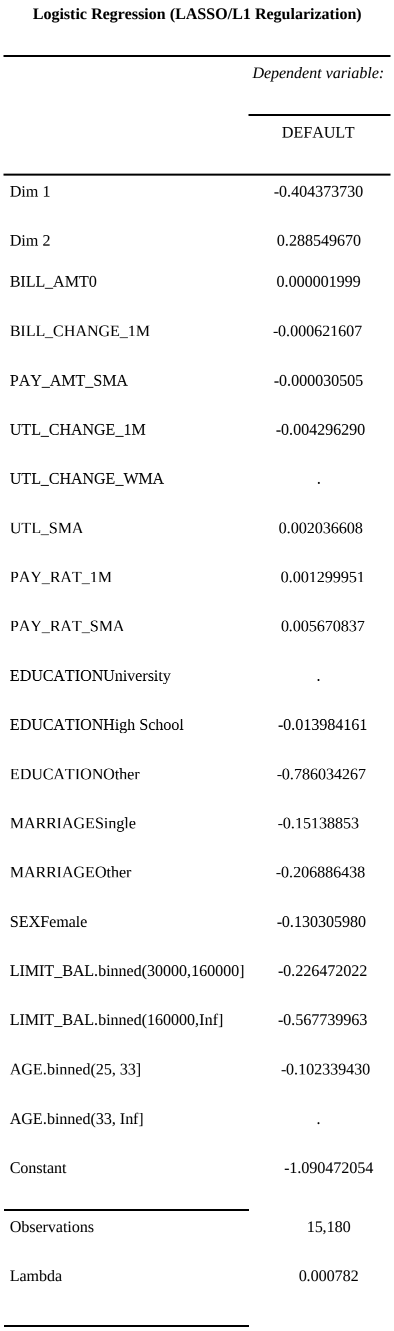
Model 2: Gradient Boosting (Training)													
Actual Class	Predicted Class		Totals		Actual Class	Predicted Class		TP	0.65	TP+TN	1.47	AUC	0.81
	0	1				0	1	TN	0.83	Precision	0.52	Sensitivity	0.65
0	9,742	2,015	11,757		0	0.83	0.17	Type I Error	0.17	Recall	0.65	Specificity	0.83
1	1,214	2,209	3,423		1	0.35	0.65	Type II Error	0.35	F1	0.71		

Model 2: Gradient Boosting (Test)													
Actual Class	Predicted Class		Totals		Actual Class	Predicted Class		TP	0.67	TP+TN	1.43	AUC	0.78
	0	1				0	1	TN	0.76	Precision	0.43	Sensitivity	0.67
0	4,381	1,385	5,766		0	0.76	0.24	Type I Error	0.24	Recall	0.67	Specificity	0.76
1	514	1,043	1,557		1	0.33	0.67	Type II Error	0.33	F1	0.70		

5.3 Logistic Regression

Model 3 builds an optimal Logistic Regression model by performing a LASSO regression. LASSO, or Least Absolute Shrinkage and Selection Operator, discourages large weights in the coefficients by setting a penalty on the absolute values of the predictor variables. By shrinking the coefficients, the L1 penalty helps prevent overfitting in the model. LASSO is beneficial for building an optimal model because in addition to performing regularization, when the tuning parameter is large enough some coefficients can shrink to 0. Thus, regularization and variable selection is performed simultaneously (James et. al, 2013, p. 219). The summary of the coefficients for the LASSO Logistic Regression is in Table 16.

Table 16: LASSO Logistic Regression Summary



Logistic Regression (LASSO/L1 Regularization)

	Dependent variable:
	DEFAULT
Dim 1	-0.404373730
Dim 2	0.288549670
BILL_AMT0	0.000001999
BILL_CHANGE_1M	-0.000621607
PAY_AMT_SMA	-0.000030505
UTL_CHANGE_1M	-0.004296290
UTL_CHANGE_WMA	.
UTL_SMA	0.002036608
PAY_RAT_1M	0.001299951
PAY_RAT_SMA	0.005670837
EDUCATIONUniversity	.
EDUCATIONHigh School	-0.013984161
EDUCATIONOther	-0.786034267
MARRIAGESingle	-0.15138853
MARRIAGEOther	-0.206886438
SEXFemale	-0.130305980
LIMIT_BAL.binned(30000,160000]	-0.226472022
LIMIT_BAL.binned(160000,Inf]	-0.567739963
AGE.binned(25, 33]	-0.102339430
AGE.binned(33, Inf]	.
Constant	-1.090472054
Observations	15,180
Lambda	0.000782

The Logistic Regression model performed only slightly below the Gradient Boosting model. Model 3 scored an AUC of on the training set and .77 on the test set. Meanwhile, the sensitivity score was .64 and .65 on the training and test sets, respectively. However, just as was observed in Model 2, there is a tradeoff in predicting true positives with higher false positives. The false positive rate for Model 3 was .23 on the training set and .24 on the test set.

Table 17: Model 3 Results

Model 3: Logistic Regression (Training)													
Actual Class	Predicted Class		Totals		Actual Class	Predicted Class		TP	0.64	TP+TN	1.41	AUC	0.76
	0	1				0	1	TN	0.77	Precision	0.45	Sensitivity	0.64
0	9,052	2,705	11,757		0	0.77	0.23	Type I Error	0.23	Recall	0.64	Specificity	0.77
1	1,228	2,195	3,423		1	0.36	0.64	Type II Error	0.36	F1	0.69		

Model 3: Logistic Regression (Test)													
Actual Class	Predicted Class		Totals		Actual Class	Predicted Class		TP	0.65	TP+TN	1.42	AUC	0.77
	0	1				0	1	TN	0.76	Precision	0.43	Sensitivity	0.65
0	4,408	1,358	5,766		0	0.76	0.24	Type I Error	0.24	Recall	0.65	Specificity	0.76
1	543	1,014	1,557		1	0.35	0.65	Type II Error	0.35	F1	0.69		

5.4 Naïve Bayes

Model 4 is a Naïve Bayes model. Naïve Bayes is a simple classification model based on Bayes' Theorem. One nuance that must be considered for Naïve Bayes is that it assumes independence among the predictors, which is rarely the case. As a result, the probability that is calculated is not the true probability. However, this assumption does allow for simplifying the estimation in the model (Friedman et. al, 2009, p. 211).

Table 18: Model 4 Results

Model 4: Naïve Bayes (Training)													
Actual Class	Predicted Class		Totals		Actual Class	Predicted Class		TP	0.47	TP+TN	1.35	AUC	0.67
	0	1				0	1	TN	0.88	Precision	0.53	Sensitivity	0.47
0	10,318	1,439	11,757		0	0.88	0.12	Type I Error	0.12	Recall	0.47	Specificity	0.88
1	1,820	1,603	3,423		1	0.53	0.47	Type II Error	0.53	F1	0.59		

Model 4: Naïve Bayes (Test)													
Actual Class	Predicted Class		Totals		Actual Class	Predicted Class		TP	0.47	TP+TN	1.35	AUC	0.67
	0	1				0	1	TN	0.87	Precision	0.50	Sensitivity	0.47
0	5,035	731	5,766		0	0.87	0.13	Type I Error	0.13	Recall	0.47	Specificity	0.87
1	820	737	1,557		1	0.53	0.47	Type II Error	0.53	F1	0.59		

The Naïve Bayes model was able to generalize well between the training and test sets, scoring an AUC of .71, Sensitivity of 0.47, and Balanced Accuracy of .675 for both sets. The False Positive Rate only slightly differed, with a score of .12 in the training set and .13 in the test set.

6. Model Comparison

The side-by-side comparison of all four models is shown in Table 19. Overall, the Gradient Boosting model performed best with an AUC of .81/.78, Balanced Accuracy of .74/.72, and F1 of .71/.70. The next best performing model was the LASSO Regression. Despite a simpler model, there was only a slight drop off in scores between Model 2 and Model 3. In addition, Model 3 generalizes better to the test set, as Model 2 is slightly overfit.

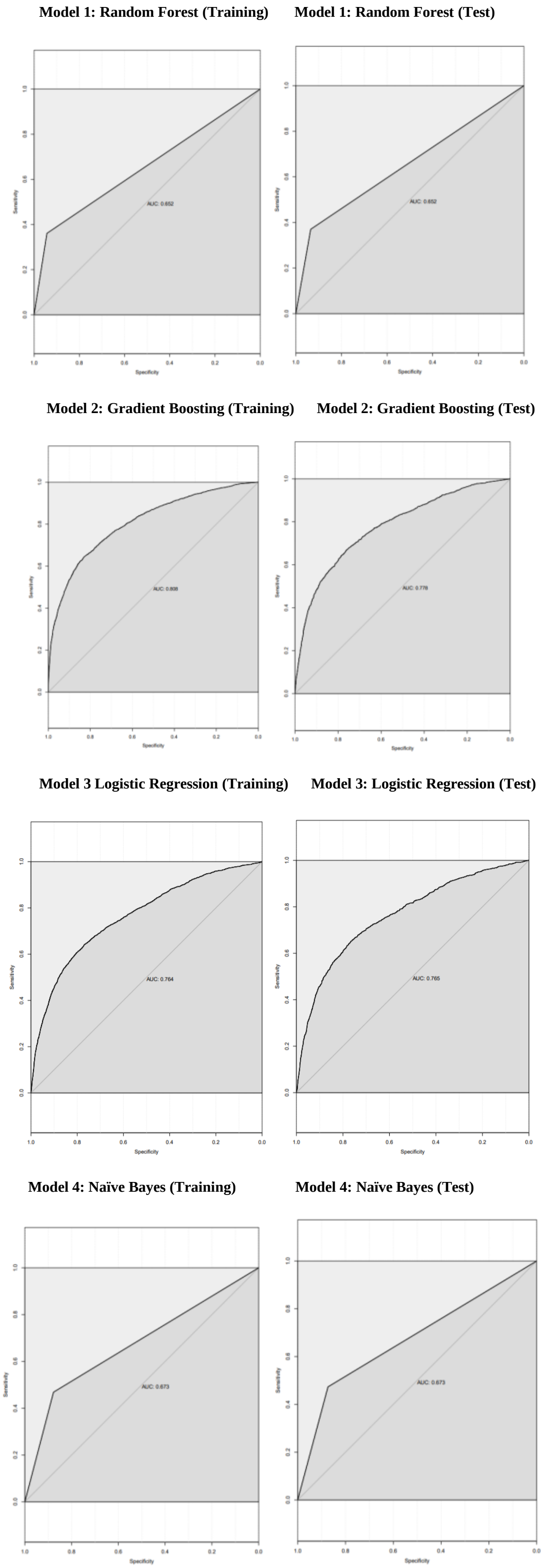
As mentioned in 5.2, there appears to be a tradeoff between higher Sensitivity scores and False Positive Rates. The Random Forest and Naïve Bayes models performed with the lowest False Positive Rates, with scores of .06/.07 and .12/.13, respectively. However, these models also performed poorly for predicting true positives.

Table 19: Model Comparison

	Model 1 Random Forest		**Model 2 Gradient Boosting**		**Model 3 LASSO Regression**		**Model 4 Naïve Bayes**	
Metric	**Training**	**Test**	**Training**	**Test**	**Training**	**Test**	**Training**	**Test**
AUC	0.65	0.65	0.81	0.78	0.76	0.77	0.67	0.67
Balanced Accuracy	0.65	0.66	0.74	0.72	0.71	0.71	0.68	0.68
Sensitivity	0.36	0.37	0.65	0.67	0.64	0.65	0.47	0.47
Specificity	0.94	0.93	0.83	0.76	0.77	0.76	0.88	0.87
False Positive Rate	0.06	0.07	0.17	0.24	0.23	0.24	0.12	0.13
F1	0.51	0.52	0.71	0.70	0.69	0.69	0.59	0.59

One final comparison between each model is done using the ROC curves. As can be seen in Figure n, the Gradient Boosting and LASSO Regression models performed the best. Meanwhile, despite the low false positive rates, the Random Forest and Naïve Bayes models performed poorly in predicting defaults.

Figure 12: Model Comparison ROC Curves



Model 1: Random Forest (Training)

Model 1: Random Forest (Test)

Model 2: Gradient Boosting (Training)

Model 2: Gradient Boosting (Test)

Model 3 Logistic Regression (Training)

Model 3: Logistic Regression (Test)

Model 4: Naïve Bayes (Training)

Model 4: Naïve Bayes (Test)

7. Conclusion

One of the biggest challenges in predicting credit card default is there is no clear set of red flags that can guarantee payment or non-payment. Two individuals with the exact same profiles could differ each month between defaulting or making payments on their card. One solution to this would be using a more robust dataset. For example, some possible variables that could improve upon the above models are occupation, number of dependents, and other outstanding loans. Of the variables analyzed for this data set, the most important predictor for default rates is the recent behavior of each client. Clients who consistently make their payments each month are likely to continue doing so. This example is best shown by the principal components of the PAY_n variables being the most important variable for each model.

One takeaway from this analysis is the tradeoff between predicting true positives and false positives. The more robust models, in this case the Gradient Boosting and Logistic Regression models, were able to correctly predict defaults at a much higher rate. However, the false positive rates were also higher for both models. Future analysis from a bank would need to evaluate the costs and benefits between acquiring new clients and the cost of defaults. As delinquent loans are costly to a bank, so to is the opportunity cost of rejecting potential clients. One way to alleviate this would be to bin clients based on their probability of default and charge different interest rates for each bin. The bank will also need to monitor the quality of their models. Details on monitoring performance of the models will be discussed in Section 2 of this analysis.

II. Performance Monitoring Guide

1. Production Model

The model to be used in production is the Logistic Regression model described in 5.3 of Section 1. This is a LASSO regression model. The LASSO regression shrinks the size of the coefficients through a process called L1 regularization. In L1 regularization, a penalty is set on the absolute values of the predictor values. Shrinking the coefficients reduces variance by ensuring no single variable is overweight in the model. In addition, coefficients that are not useful are shrunk down to 0. This combination of shrinking coefficients and variable selection ultimately reduces the chance of overfitting the model.

In the production model, Dim 1 and Dim 2 were given the heaviest weights. Meanwhile, UTL_CHANGE_WMA and the factor variables for Education (University) and Age (+ 33) were removed. Full summary of the model is shown in Table 20.

Table 20: Production Model Summary

Logistic Regression (LASSO/L1 Regularization)

Dependent variable:	DEFAULT
Dim 1	-0.404373730
Dim 2	0.288549670
BILL_AMT0	0.000001999
BILL_CHANGE_1M	-0.000621607
PAY_AMT_SMA	-0.000030505
UTL_CHANGE_1M	-0.004296290
UTL_SMA	0.002036608
PAY_RAT_1M	0.001299951
PAY_RAT_SMA	0.005670837
EDUCATIONHigh School	-0.013984161
EDUCATIONOther	-0.786034267
MARRIAGESingle	-0.15138853
MARRIAGEOther	-0.206886438
SEXFemale	-0.130305980
LIMIT_BAL.binned(30000,160000]	-0.226472022
LIMIT_BAL.binned(160000,Inf]	-0.567739963

AGE.binned(25, 33]	-0.102339430
Constant	-1.090472054
Observations	15,180
Lambda	0.000782

2. Model Development Performance

The performance metrics of the production model can be found in Table 2. On the training set, the model scored a Balanced Accuracy of .71, Sensitivity of .64, Specificity of .77, False Positive Rate of .23, and F1 score of .69. On the test set, the model scored a Balanced Accuracy of .71, Sensitivity of .65, Specificity of .64, False Positive Rate of .24, and F1 score of .69. The nearly identical scores on both data sets shows the model will be able to generalize well.

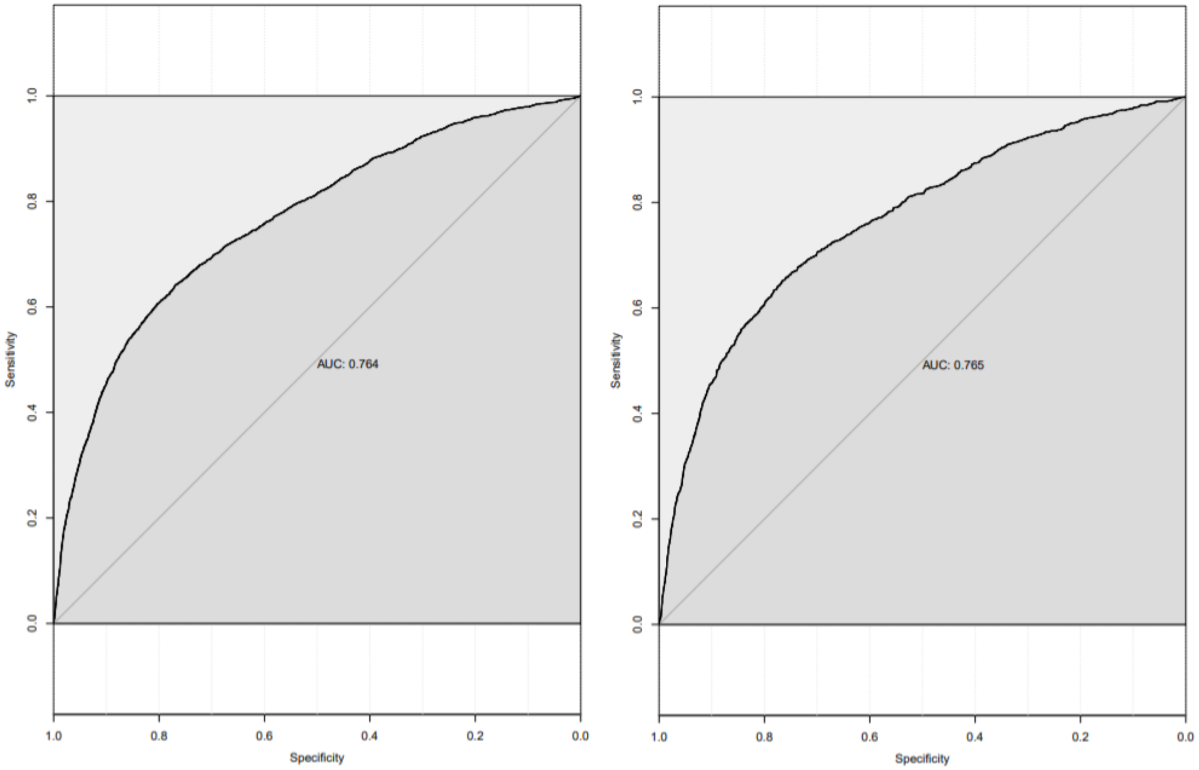
Table 21: Production Model Performance

Production Model: Logistic Regression (Training)													
Actual Class	Predicted Class		Totals		Actual Class	Predicted Class		TP	0.64	TP+TN	1.41	AUC	0.76
	0	1				0	1	TN	0.77	Precision	0.45	Sensitivity	0.64
0	9,052	2,705	11,757		0	0.77	0.23	Type I Error	0.23	Recall	0.64	Specificity	0.77
1	1,228	2,195	3,423		1	0.36	0.64	Type II Error	0.36	F1	0.69		

Production Model: Logistic Regression (Test)													
Actual Class	Predicted Class		Totals		Actual Class	Predicted Class		TP	0.65	TP+TN	1.42	AUC	0.77
	0	1				0	1	TN	0.76	Precision	0.43	Sensitivity	0.65
0	4,408	1,358	5,766		0	0.76	0.24	Type I Error	0.24	Recall	0.65	Specificity	0.76
1	543	1,014	1,557		1	0.35	0.65	Type II Error	0.35	F1	0.69		

In addition to evaluating the metrics for the mode, a ROC curve has been created for the production model. The ROC curve plots the sensitivity and specificity of the model and illustrates how well the model can distinguish between each of the classes. As mentioned, the model has a sensitivity of .64 on the training set and .65 on the test set, while scoring a specificity of .77 on the training set and .76 on the test set. Meanwhile, the AUC on the training set is .764 and .765 on the test set. The ROC curves are shown in Figure 1.

Figure 13: Production Model ROC Curves



Training Set

Test Set

One final evaluation to be performed on the production model is a Kolgomorov/Smirnov test, which will be referred to as a KS test. The KS test is a goodness-of-fit test used for determining the maximum difference between the true positive rate and true negative rate. Applied to credit default prediction, two distributions are made for good clients (non-defaulters) and bad clients (defaulters). The KS score is the maximum, across all credit default predictions, of the difference in the cumulative proportions of good and bads. A KS score of 0 indicates the model fails to differentiate between defaulters and non-defaulters, while a KS score of 100 indicates the model can perfectly differentiate between defaulters and non-defaulters.

Applying the KS test to the production model, the observations were divided into 20 groups or half-deciles. The highest scoring half-decile was group 6, where the model scored a 40.6 on the training set and 41.1 on the test set. Full results for the KS test are shown in Table 3.

Table 22: Kolgomorov/Smirnov Test

Kolmogorov/Smirnov Test

Production Model: Logistic Regression (Training)

Decile	Obs	Target (Y=1)	NonTarget (Y=0)	Target Density	NonTarget Density	Target CDF	NonTarget CDF	KS Stat
1	759	555	204	16.2%	1.7%	16.2%	1.7%	14.5%
2	759	434	325	12.7%	2.8%	28.9%	4.5%	24.4%
3	759	367	392	10.7%	3.3%	39.6%	7.8%	31.8%
4	759	320	439	9.3%	3.7%	9.3%	3.7%	5.6%
5	759	239	520	7.0%	4.4%	55.9%	16.0%	40.0%
6	759	188	571	5.5%	4.9%	61.4%	20.8%	**40.6%**
7	759	165	594	4.8%	5.1%	4.8%	5.1%	0.2%
8	759	144	615	4.2%	5.2%	70.5%	31.1%	39.3%
9	759	110	649	3.2%	5.5%	73.7%	36.7%	37.0%
10	759	115	644	3.4%	5.5%	3.4%	5.5%	2.1%
11	759	109	650	3.2%	5.5%	80.2%	47.7%	32.6%
12	759	107	652	3.1%	5.5%	83.3%	53.2%	30.1%
13	759	118	641	3.4%	5.5%	3.4%	5.5%	2.0%
14	759	91	668	2.7%	5.7%	89.5%	64.3%	25.1%
15	759	98	661	2.9%	5.6%	92.3%	70.0%	22.4%
16	759	78	681	2.3%	5.8%	2.3%	5.8%	3.5%
17	759	54	705	1.6%	6.0%	96.2%	81.8%	14.4%
18	759	49	710	1.4%	6.0%	97.6%	87.8%	9.8%
19	759	37	722	1.1%	6.1%	1.1%	6.1%	5.1%
20	758	45	713	1.3%	6.1%	100.0%	100.0%	0.0%
Totals	**15,179**	**3,423**	**11,756**	**100.0%**	**100.0%**			

Kolmogorov/Smirnov Test

Production Model: Logistic Regression (Test)

Decile	Obs	Target (Y=1)	NonTarget (Y=0)	Target Density	NonTarget Density	Target CDF	NonTarget CDF	KS Stat
1	367	250	117	16.1%	2.0%	16.1%	2.0%	14.0%
2	366	211	155	13.6%	2.7%	29.6%	4.7%	24.9%
3	366	176	190	11.3%	3.3%	40.9%	8.0%	32.9%
4	366	141	225	9.1%	3.9%	9.1%	3.9%	5.2%
5	366	108	258	6.9%	4.5%	56.9%	16.4%	40.5%
6	366	85	281	5.5%	4.9%	62.4%	21.3%	**41.1%**
7	366	77	289	4.9%	5.0%	4.9%	5.0%	0.1%
8	366	64	302	4.1%	5.2%	71.4%	31.5%	39.9%
9	366	50	316	3.2%	5.5%	74.6%	37.0%	37.6%
10	366	43	323	2.8%	5.6%	2.8%	5.6%	2.8%
11	367	60	307	3.9%	5.3%	81.2%	47.9%	33.3%
12	366	33	333	2.1%	5.8%	83.4%	53.7%	29.7%
13	366	52	314	3.3%	5.4%	3.3%	5.4%	2.1%
14	366	54	312	3.5%	5.4%	90.2%	64.6%	25.6%
15	366	33	333	2.1%	5.8%	92.3%	70.3%	22.0%
16	366	24	342	1.5%	5.9%	1.5%	5.9%	4.4%
17	366	35	331	2.2%	5.7%	96.1%	82.0%	14.1%
18	366	22	344	1.4%	6.0%	97.5%	88.0%	9.5%
19	366	21	345	1.3%	6.0%	1.3%	6.0%	4.6%
20	366	18	348	1.2%	6.0%	100.0%	100.0%	0.0%
Totals	**7,322**	**1,557**	**5,765**	**100.0%**	**100.0%**			

3. Performance Monitoring Plan

After deploying to production, the model must still be monitored for changes in performance. One of the main causes for changes in performance is that over time, change in consumer activity may shift the importance of certain variables. In addition, as new data comes in, the statistical properties of predictors may change. To track this model drift, a monitoring plan will be put in place for monitoring and evaluating. In production, the model will be revalidated at a standard interval of six months. In addition, metrics will be continually updated as new data comes in and will be tracked using the KS score.

The performance monitoring plan will be based on defining three different statuses of the model: Red, Amber, and Green. A model with a Red status will need immediate redevelopment. A model with an Amber status will need to be re-validated at the end of three months. A model with a Green status is performing as expected and will be revalidated at the standard interval of six months. The thresholds used will be percentages of the original KS score. A model performing within 2.5% of the original KS score will be coded as Green. Using 41.1 from the test set as the starting point, the model will continue to be labeled Green if the KS score stays above 40.0. If the model performance is between 2.5 and 5.0% of the original score, it will be coded as Amber. Starting with 41.1, a model is Amber status if the KS score is between 39.0 and 40.0. Finally, a model with a change in performance of more than 5.0% will be given a Red status. Definitions and thresholds for the performance monitoring plan are shown in Table 4.

Table 23: Performance Monitoring Plan

Status		Threshold	KS	Definition
Green	🟩	<= 2.5%	40.0	Model performs as expected. Will be revalidated at standard six-month interval.
Amber	🟧	2.5 - 5.0%	39.0	Model needs to be revalidated after three months.
Red	🟥	> 5.0%	< 39.0	Model needs redevelopment.

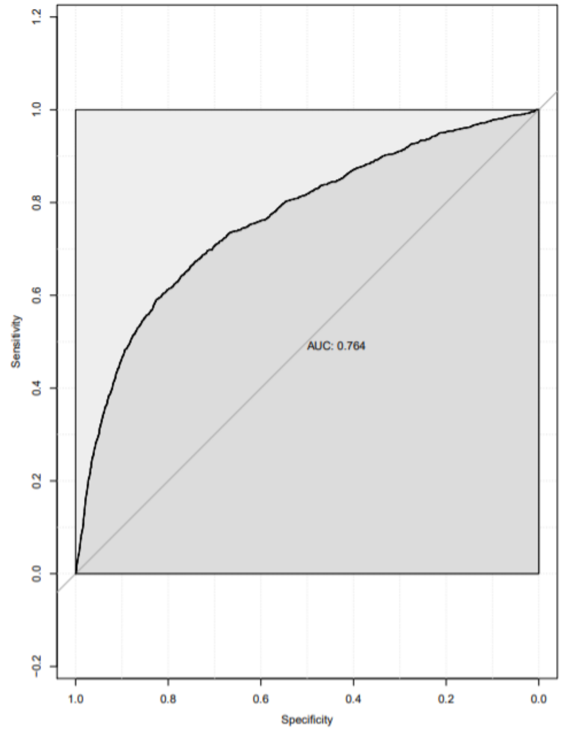
III. Performance Validation

The first round of validation to be performed on the production model will be run on the validation set. The validation set was 25% of the original data set that was originally split in Section 2.2 of Part 1. On the validation set, the production model had a Sensitivity of .83 and Specificity of .59. The model also had a False Positive Rate of .17 and F1 score of .67. The Balanced Accuracy was .71 and AUC was .764. Full results for the production model on the validation set are shown in Table 5 and Figure 2.

Table 24: Production Model Performance Validation

Production Model: Logistic Regression (Validation)													
Actual Class	Predicted Class		Totals		Actual Class	Predicted Class		TP	0.59	TP+TN	1.42	AUC	0.76
	0	1				0	1	TN	0.83	Precision	0.49	Sensitivity	0.59
0	4,824	1,017	5,841		0	0.83	0.17	Type I Error	0.17	Recall	0.59	Specificity	0.83
1	678	978	1,656		1	0.41	0.59	Type II Error	0.41	F1	0.67		

Figure 14: Validation ROC Curve



Finally, the KS test is run on the validation set, in accordance with the steps outlined in Section 3. For the performance validation, production model scored a KS of 41.1, which is identical to the test set. Since there is no change and the performance is within the margin outlined, the current status of the production model is Green. The full KS results are shown in Table 7.

Table 25: KS Test (Validation)

Kolmogorov/Smirnov Test

Production Model: Logistic Regression (Validation)

Decile	Obs	Target (Y=1)	NonTarget (Y=0)	Target Density	NonTarget Density	Target CDF	NonTarget CDF	KS Stat
1	375	253	122	15.3%	2.1%	15.3%	2.1%	13.2%
2	375	225	150	13.6%	2.6%	28.9%	4.7%	24.2%
3	375	186	189	11.2%	3.2%	40.1%	7.9%	32.2%
4	374	156	219	9.4%	3.8%	9.4%	3.8%	5.7%
5	375	114	260	6.9%	4.5%	56.4%	16.1%	40.3%
6	375	93	282	5.6%	4.8%	62.0%	20.9%	**41.1%**
7	375	87	288	5.3%	4.9%	5.3%	4.9%	0.3%
8	375	75	300	4.5%	5.1%	71.8%	31.0%	40.8%
9	375	47	328	2.8%	5.6%	74.6%	36.6%	38.0%
10	374	45	329	2.7%	5.6%	2.7%	5.6%	2.9%
11	375	59	316	3.6%	5.4%	80.9%	47.7%	33.3%
12	375	43	332	2.6%	5.7%	83.5%	53.3%	30.2%
13	375	47	328	2.8%	5.6%	2.8%	5.6%	2.8%
14	375	45	330	2.7%	5.7%	89.1%	64.6%	24.5%
15	374	36	338	2.2%	5.8%	91.2%	70.4%	20.9%
16	375	41	334	2.5%	5.7%	2.5%	5.7%	3.2%
17	375	32	343	1.9%	5.9%	95.7%	82.0%	13.7%
18	375	25	350	1.5%	6.0%	97.2%	88.0%	9.2%
19	375	26	349	1.6%	6.0%	1.6%	6.0%	4.4%
20	374	21	353	1.3%	6.0%	100.0%	100.0%	0.0%
Totals	**7,496**	**1656**	**5,840**	**100.0%**	**100.0%**			

Model Status: Green 🟩

Appendix A: Model Equations

Model Development Metrics

$\mathrm{AIC} = \ 2\mathrm{k} - 2\ln\,(\mathrm{L})$

- K = number of estimated parameters in the model
- L = Maximized likelihood function for the estimated model

Logistic Regression: $\log\left(\frac{p(X)}{1-p(X)}\right) = \ \beta_o + \beta_1 X_1 + \cdots + \beta_p X_p$

Naive Bayes: $\mathrm{p(c|x)} = \ \frac{p(\mathcal{X}|C)*p(c)}{p(x)}$

- c = class
- x = data

Model Evaluation Metrics

True Positive Rate $= \ \frac{True\ Positives}{Total\ Positives}$; also called Sensitivity and Recall

True Negative Rate $= \ \frac{True\ Negatives}{Total\ Negatives}$; also called Specificity

False Positive Rate $= \ \frac{False\ Positives}{Total\ Negatives}$; also called Type I Error

False Negative Rate $= \ \frac{False\ Negatives}{Total\ Positives}$; also called Type II Error

Balanced Accuracy $= \ \frac{True\ Negative\ Rate+True\ Positive\ Rate}{2}$

Precision $= \ \frac{True\ Positives}{True\ Positives+False\ Positives}$; also called Positive Predictive Value

$\mathrm{F1} = 2 * \ \frac{Precision*Recall}{Precision+Recall}$

Appendix B: Index

Tables

Figures

Equations

References

Everitt, B., & Dunn, G. (2001). *Applied multivariate data analysis*. London, UK: Arnold.

Hastie, T., Friedman, J., & Tisbshirani, R. (2009). *The Elements of statistical learning: Data mining, inference, and prediction*. New York, NY: Springer.

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An introduction to statistical learning: With applications in R*. New York, NY: Springer.

Thomas, L. C. (2009). *Consumer credit models: Pricing, profit, and portfolios*. Oxford, UK: Oxford University Press.

Yeh, I., & Lien, C. (2009). The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients. *Expert Systems with Applications, 36*(2), 2473-2480. doi:10.1016/j.eswa.2007.12.020