Model #101: Credit Card Default Model

Model Development Guide

Daniel Macdonald, MILR, MHCS
2019 SP_MSDS_498-DL_Sec58
May 26, 2019

Description:

Model Development Guide for MSDS capstone project in credit modeling. Project includes overview of the source data, feature engineering, exploratory data analysis, and exploration of four predictive modeling approaches: Random Forest, Gradient Boosting, Logistic Regression, and Principal Component Analysis using a Neural Network.

The objective of the modeling exercise is to accurately predict client default on credit accounts using a supervised data set. Original data includes demographic detail of the clients as well loan billing and payment history over six months. Results for each model are compared using performance measures: Sensitivity (percent 'True Positives'), Specificity (percent 'True Negatives') and Accuracy (True Positive + True Negative/Total). Using these measures, there is a final recommendation to model approach that most accurately predicts client default using the data set provided.

Table of contents:

1. Introduction

1.1 Problem Description & Scope

For most people, applying for a credit card or bank loan involves the completion of an application and a credit check. This is a fundamental part of the application process. It may be obvious what banks hope to achieve through this process – to reduce the risk of lending money to people who are not likely to pay it back. The more complex aspect of this same question is how the application process works. How can we estimate the likelihood of loan default? If we approve all loan applications regardless of risk, then the bank may stand to lose a lot of money. However, the same might be true if we are overly selective of which clients to offer credit.

In the case study used for this project, we take a look at the loan approval and payment history for a bank in Taiwan over a six month period in 2005. In the example, the bank we are researching has over 5 billion (NTD) at risk in terms of approved loans to 30,000 clients during the observed period, which is roughly $158 Million U.S. Dollar. In September, 2005 the bank reports that its clients owe ~1.5 billion (NTD) and one in five loans are considered to be in 'default', meaning they show no signs of paying their loans back. This default population represents ~322 million (NTD), or $10 Million USD that the bank may need to write off.

This is clearly a huge operational risk that needs to be reduced. For every 1% reduction in default costs, the bank averages ~3.2 million (NTD) in savings, or $100k USD. However, if we go to the the opposite extreme the bank would also risk a financial loss. Four out of five clients pay back their loans responsibly with interest, so it makes sense to factor in some amount of expected loss. But what is a reasonable amount that results in maximum profit?

That question perhaps goes beyond the limits of this analysis, but we will start with a question of how we can most accurately predict the likelihood of default prior to the loan being approved.

1.2 Methodology and approach

The practical guide for banks to establish a credit lending framework is the "Credit Card Lending Comptrollers Handbook". (OCC 2017) This guide is used to enable oversight of the financial lending industry, and establishes the standard of best practice. Along with guidance on risk associated with lending practices and examination procedures, the handbook gives direct guidance on credit scoring models and methodology:

> "Most banks use credit scorecards to some degree in their credit card operations. Credit scorecards, also referred to as models, are risk-ranking tools that attempt to differentiate between accounts that will exhibit "good" behavior and those that will not…The use of models also can pose risk to the bank specifically, the risk that the bank will suffer losses because the bank's lending strategies are based on poor or failed models."
>
> ***(OCC 2017 - Office of Comptroller of the Currency)***

In order to address the acknowledged risk of data modeling to differentiate between accounts, the OCC recommends a robust approach to model development based on the standards of the data science industry. These standards are described as:

OCC – recommended model development process

- A **clear statement of purpose** to ensure model is developed properly.
- 'Sound design' meaning robust **data exploration and variable selection**.
- Industry standards to apply theory and logic to **model development**.
- Robust model development and **testing** to ensure quality of prediction.

In line with these recommendations the following are the steps for this analysis:

- **Overview of objectives**

  As outlined section 1 – the objective of this model is to determine the best balance between identification of likely cases of default while minimizing 'false positive' results. For example, a model might have a high rate of sensitivity but sacrifice in the measure of specificity (true negatives) and overall accuracy. The desired approach would be to have an optimal rate across all three measures.

- **Data Quality Check & Transformation**

  Section 2 of this paper is a review of data from the Taiwan bank with a deep dive into each variable. We review inconsistencies in the actual data against the data dictionary, and in Section 3 describe transformation required to either summarize highly correlated data points or split continuous variables into optimal 'bins'.

- **Variable Selection for Modeling**

  Section 4 of this paper is an Exploratory Data Analysis (EDA) of all the data points, their relative correlation to each other, and their correlation to the response variable. Data points that are highly correlated to each other may be reason for either further data transformation or elimination.

- **Model Development and Measurement**

  The recommended approach will look at four modeling techniques
  (2) Decision tree based models: Random Forest & Gradient Boosting
  (2) Linear based models: Logistic Regression & Principal Component Analysis
  Model performance will be measured by Sensitivity, Specificity & Accuracy

- **Assessment of 'Best Fit' Model & approach**

  Section 6 & 7 take use through a comparison of the model measures and a recommendation for the approach.

1.3 Some discoveries in the analysis

Without giving away the ending, this journey of discovery is an interesting one. At least it was for me. There are many variables that appear they would have a significant influence on the outcome during initial exploration, but they do not carry as much predictive weight as perceived. Education is one of these variables, which has a high correlation to the response variable, but in the end will not play a significant role in the chosen model. There are also some other interesting twists, such as the multi-factor influence of age, gender, and marital status, which will be a factor in a client's likelihood to default. This will pose an interesting challenge in addressing bias through our models.

2. Data Overview and Quality Check

The data provided for this capstone project are the 'Default of Credit Card Clients Data Set' posted on the UCI Machine Learning Repository.[1] Although the data are available via download from the site, there is a special revised version that is used by our class to ensure consistent application of train, test and validate split. Therefore, in the data dictionary below are five additional fields to facilitate consistent analysis, which are marked.

The description on the website suggests that the original data are a measure of client payment history and additional customer information (e.g. gender, age, marital status, bill history and balance) in relation to customer default. The data were provided by a bank in Taiwan in 2016 for the purposes of this study and consists of 30,000 individual observations with 30 attributes. A description of the attributes is provided below. Each observation summarizes an individuals' available credit, billing and payment over a six-month period from April to September of 2005. The data are to be used in predictive modeling development to predict an individual's risk of defaulting on their payments prior to providing them the credit.

2.1 Data Dictionary

Below is a summary of the original data. Field title 'PAY_0', the most current pay period is replaced with 'PAY_1' to make consistent comparison to BILL_AMT1 & PAY_AMT1.

Table 2.1.1 Variable Names & Descriptions (Original Data)

Variable Name		Description
ID		Unique Identifier by row (1 to 30,000)
LIMIT_BAL		Amount of the given credit (NT dollar)
SEX		Gender - (1 = male; 2 = female)
EDUCATION		Completed - (1 = grad; 2 = univ; 3 = HS; 4 = others)
MARRIAGE		Status - (1 = married; 2 = single; 3 = others)

[1] https://archive.ics.uci.edu/ml/datasets/default+of+credit+card+clients#

AGE	Age in years
PAY_1 : 6	repayment status from September (PAY_1) : April (PAY_6) * (-2:8) -2:0 = 'on time'; positive #'s = late payment in months
BILL_AMT1 : 6	Bill statement (NTD) - September (BILL_AMT1) : April (BILL_AMT6)
PAY_AMT_1 : 6	Payment against bill – September (PAY_AMT1) : April (PAY_AMT6)
DEFAULT	Classification – response variable; (1 = True; 0 = False)
**added fields for consistent train/test/validate for purposes of project:*	
u	Random Sort for Train/Test/Validate Split
train	Dummy variable - 1 = Train data set
test	Dummy variable - 1 = Test data set
validate	Dummy variable - 1 = Validate data set
data group	Train/Test or Validate Data (1 = Train, 2=Test, 3 = Validate)

2.2 Observations of original data

All original data are an integer data type, and the data that has been included for the purposes of a train, test, and validate split are a number data type. A table of the summary statistics of the data is found in appendix X.1. I have grouped together data fields by topic. Here are some high-level observations of each data group:

Table 2.2.1 Initial observations of the data

Data Group & Field Title	**Notes:**
Identification and Sort	
ID, u, train, test, validate, data.group	Data are originally sorted by **ID**, which is 1:30,000. **'u'** is a randomly generated number for the purpose of distributing observations. **'train','test', & 'validate'** are dummy variables where '1' indicates inclusion to group. **'data.group'** is a categorical variable - '1' = train, '2' = test, & '3' = validate.
Limit Balance	
Limit_Bal	Integer variable with minimum of 10,000 and maximum of 1,000,000 Median value is 140,000 with 25% and 75% Quantiles of 50,000:240,000 Histogram of distribution provided in Appendix X.2.1
Demographic Information	
Sex, Education, Marriage, Age	Histogram of distributions provided in Appendix X.2.2 **Sex** – binary value. Suggestion to change to value of '1' if Female & '0' if not. **Education** – Four categories in data description, but observation of additional categories (0,5&6) which = 'others'. Correlation analysis shown in Appendix X.2.2 suggest we bin to 2 categories (Grad & other; High School & University) **Marriage** – Four categorical values (0:3) in the data set. Values 0 & 3 would classify as 'other', but analysis (table X.2.2) suggest binary treatment (Y/N)
Pay Categories (On Time/Delayed)	
PAY_1 : PAY_6	Histogram of distribution provided in Appendix X.2.3
Monthly Bill Amounts	
BILL_AMT_1 : BILL_AMT_6	Histogram of distribution provided in Appendix X.2.4

Monthy Payment Amounts	
PAY_AMT_1 : PAY_AMT_6	Histogram of distribution provided in Appendix X.2.5
Response Variable	
DEFAULT	Histogram of distribution provided in Appendix X.2.6

Below is the observations counts by 'data.group' for the Train, Test & Validate splits:

Table 2.2.2: Train, Test, & Validate Splits

Data group	group	Freq	%
1	Train	15,180	50%
2	Test	7,323	25%
3	Split	7,497	25%

2.3 Notes on data discrepancies

Initial review of the data shows some inconsistencies in the actual observations and the definitions in the data dictionary. Below is a summary of these observations and suggestions of how they may be managed in the data exploration & transformation stages.

- **EDUCATION**

  In the data description, this field should have four categories: (1 = grad; 2 = univ; 3 = HS; 4 = others). However, the observation is that there are additional un-defined categories (0,5,&6). The frequency is small, and the assumption is these are 'Other'

- **MARRIAGE**

  In the data description, this field should have three categories: ((1 = married; 2 = single; 3 = others). However, the observation is that there are additional un-defined categories (0). The frequency is small, and the assumption is these are 'Other'

- **PAY_1:PAY_6**

  In the data description, this field should not have negative numbers. But the observation is that there are un-defined negative numbers (-1 & -2). The assumption is these may be pre-payment clients, but we may choose to treat as (0) = 'on time'.

- **PAY_AMT compared to BILL_AMT & DEFAULT**

  There are some examples of discrepancies in the DEFAULT response measure relative to the reported delinquency in PAY_1, BILL_AMT2 & PAY_AMT1. For example, in the most recent month there are ~ 90 examples of where the bill for the prior month appears to be fully paid, but the case is reported as DEFAULT.

Table 2.3.1: Examples of response discrepancies

ID	PAY_1 *(Sept)*	BILL_AMT2 *(August)*	PAY_AMT1 *(Sept)*	DEFAULT (?)
371	2	0	0	1
10656	1	221	899	1
12206	1	-1868	0	1
12496	2	-368	0	1
14179	1	-2364	0	1
28202	1	779	781	1
28592	2	0	0	1
29109	2	0	0	1

In this circumstance, the recommendation is to leave the data as it is. It is not clear why this may be happening, but we do not want to over engineer the data, assuming that some ‘noise’ is a natural occurrence and should be a part of the model anyway. However, it does call into question the quality of the reporting which ultimately impacts model performance. A complete analysis of each variable and recommendations for transformation can be found in Appendix 8.2 & 8.3.

3. Feature Engineering

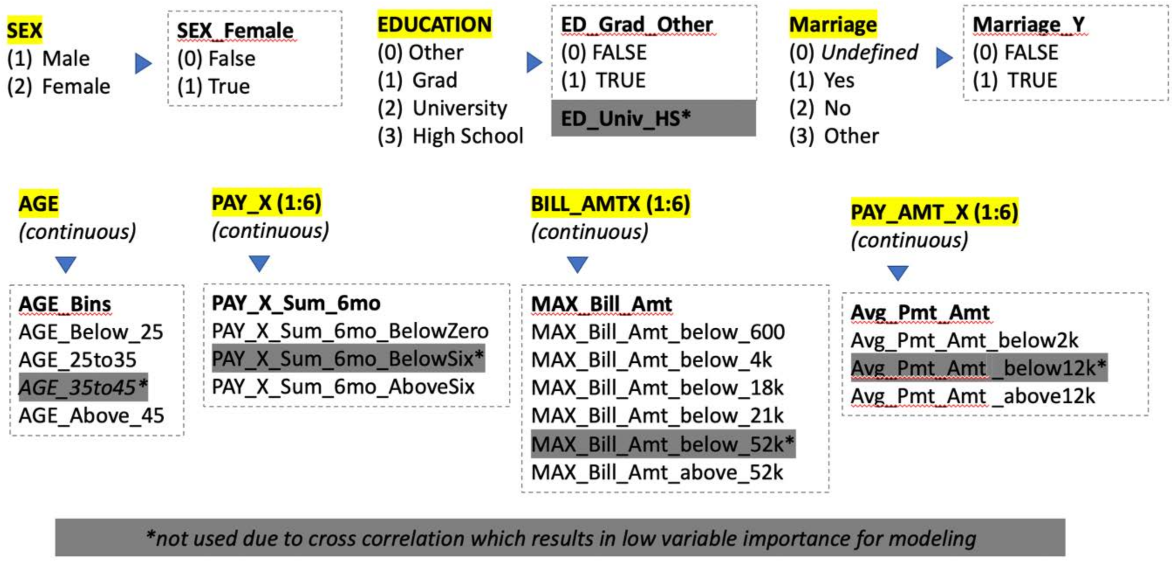
Within this section we consider how to transform the original data and develop new descriptive variables that may reduce cross correlation and provide more insight into the likelihood of default. In section 3.1, we consider the original data points and how they may be transformed. 3.2 looks at ‘new’ variables that are calculated from the original data set, such as ‘Balance Growth’ over the observed period or ‘Average Utilization’, which is the percent of the limit balance used.

Finally, in section 3.3 there is a recommendation to consider two data sets for modeling due to the nature of how the model techniques work. For example, tree analysis techniques work best with fewer variables that are continuous, while logistic regression will perform best if continuous variables are binned. Recommended binning of each variable is considered to maximize the variance in correlation to the response variable.

3.1 Transformation of original data

Below is an overview of the recommended data transformations and the resulting data fields from feature engineering. See visual of the mapping to new variable below and an example of the methodology. A full description of the data transformations and bin analysis to derive maximum variance in correlation can be found in Appendix 8.3.1. The resulting data dictionaries are provided in section 3.3.

Figure 3.1 – recommended variable transformation



**not used due to cross correlation which results in low variable importance for modeling*

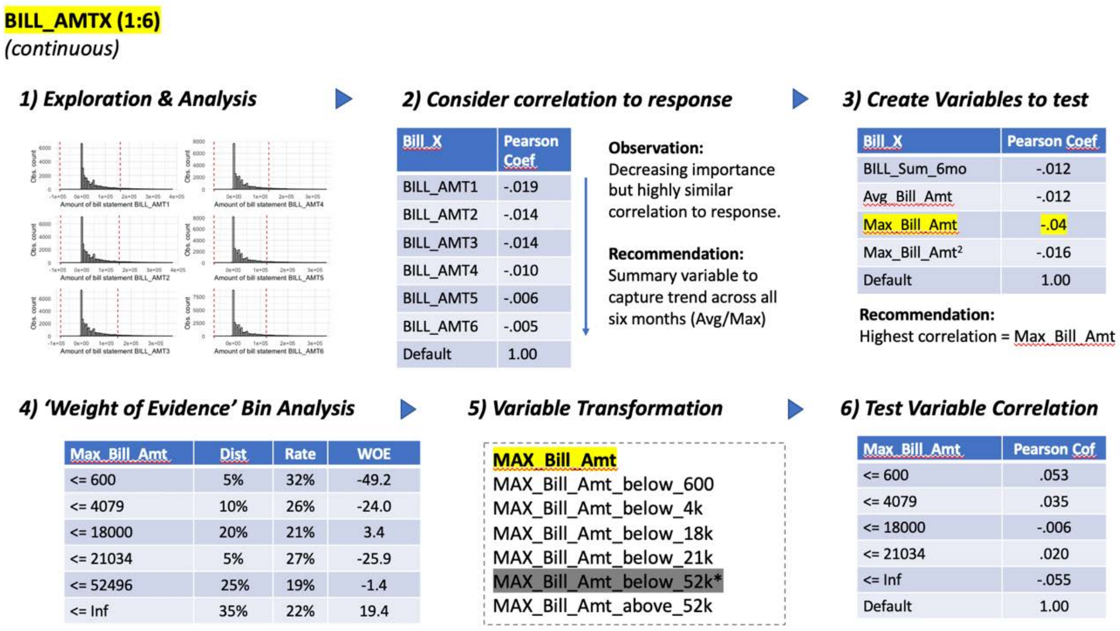
3.1.2 Approach to variable transformation

To demonstrate the logic behind the data transformation, we will review a specific example of one of the variables - 'BILL_AMTX'. The process below first examines the distribution and correlation of the variables to one another and to the response variable. The next step explores variable alternatives that maintain the correlation to the response variable while still capturing the trend across the 6 variables (BILL_AMT1 : BILL_AMT6).

In step 2, it is clear that the correlation across the variables are similar, and then in step 3 we explore alternatives, such as using the sum of the 6 variables, the average value across the 6 months, the maximum amount in the six months, and the max amount squared. Looking at the correlation to the response variable, it seems that the maximum amount is the best approach.

The next step uses a weight of evidence analysis to determine the optimal bin distribution relative to the response indicator. The continuous variable 'Max_Bill_Amt' is then transformed to align with that recommended approach. The resulting correlation analysis indicates that individuals with a Max_Bill_Amt < 600 are least likely to default, while individuals with a Max_Bill_Amt > 21k are the most likely candidates to default.

Figure 3.1.2 – example variable analysis and recommendation logic



BILL_AMTX (1:6)
(continuous)

1) Exploration & Analysis

Obs. count
Amount of bill statement BILL_AMT1
Amount of bill statement BILL_AMT2
Amount of bill statement BILL_AMT3
Amount of bill statement BILL_AMT4
Amount of bill statement BILL_AMT5
Amount of bill statement BILL_AMT6
-1e+05
0e+00
1e+05
2e+05
3e+05
4e+05

2) Consider correlation to response

Bill_X	Pearson Coef
BILL_AMT1	-.019
BILL_AMT2	-.014
BILL_AMT3	-.014
BILL_AMT4	-.010
BILL_AMT5	-.006
BILL_AMT6	-.005
Default	1.00

Observation:
Decreasing importance but highly similar correlation to response.

Recommendation:
Summary variable to capture trend across all six months (Avg/Max)

3) Create Variables to test

Bill_X	Pearson Coef
BILL_Sum_6mo	-.012
Avg_Bill_Amt	-.012
Max_Bill_Amt	-.04
Max_Bill_Amt^2	-.016
Default	1.00

Recommendation:
Highest correlation = Max_Bill_Amt

4) 'Weight of Evidence' Bin Analysis

Max_Bill_Amt	Dist	Rate	WOE
<= 600	5%	32%	-49.2
<= 4079	10%	26%	-24.0
<= 18000	20%	21%	3.4
<= 21034	5%	27%	-25.9
<= 52496	25%	19%	-1.4
<= Inf	35%	22%	19.4

5) Variable Transformation

MAX_Bill_Amt
MAX_Bill_Amt_below_600
MAX_Bill_Amt_below_4k
MAX_Bill_Amt_below_18k
MAX_Bill_Amt_below_21k
MAX_Bill_Amt_below_52k*
MAX_Bill_Amt_above_52k

6) Test Variable Correlation

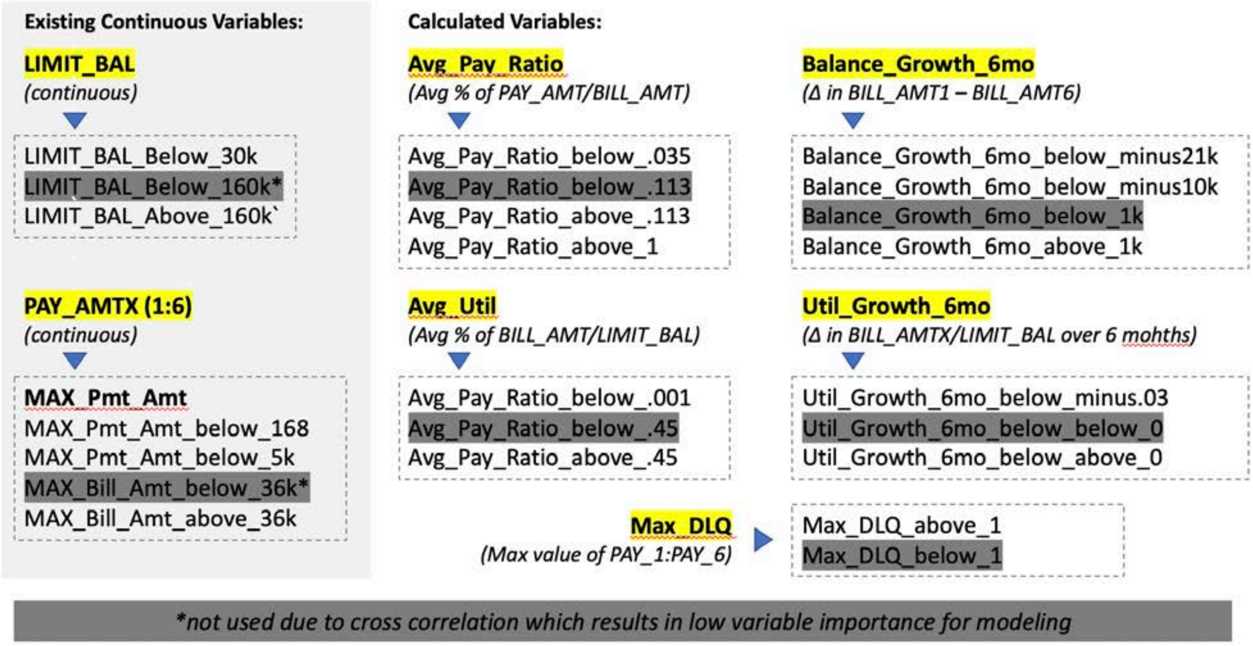
Max_Bill_Amt	Pearson Cof
<= 600	.053
<= 4079	.035
<= 18000	-.006
<= 21034	.020
<= Inf	-.055
Default	1.00

****A full outline of the transformation of each variable can be found in Appendix X.3.1.***

3.2 Development of new predictive variables

Below is an overview of the recommended steps to develop new predictive variables from the original data set. See visual of the mapping to new variable below and an example of the methodology. A full description of the data transformations and bin analysis to derive maximum variance in correlation can be found in Appendix 8.3.2. The resulting data dictionaries are provided in section 3.3.

Figure 3.2.1 – example variable analysis and recommendation logic



Existing Continuous Variables:

LIMIT_BAL
(continuous)

LIMIT_BAL_Below_30k
LIMIT_BAL_Below_160k*
LIMIT_BAL_Above_160k`

PAY_AMTX (1:6)
(continuous)

MAX_Pmt_Amt
MAX_Pmt_Amt_below_168
MAX_Pmt_Amt_below_5k
MAX_Bill_Amt_below_36k*
MAX_Bill_Amt_above_36k

Calculated Variables:

Avg_Pay_Ratio
(Avg % of PAY_AMT/BILL_AMT)

Avg_Pay_Ratio_below_.035
Avg_Pay_Ratio_below_.113
Avg_Pay_Ratio_above_.113
Avg_Pay_Ratio_above_1

Balance_Growth_6mo
(Δ in BILL_AMT1 – BILL_AMT6)

Balance_Growth_6mo_below_minus21k
Balance_Growth_6mo_below_minus10k
Balance_Growth_6mo_below_1k
Balance_Growth_6mo_above_1k

Avg_Util
(Avg % of BILL_AMT/LIMIT_BAL)

Avg_Pay_Ratio_below_.001
Avg_Pay_Ratio_below_.45
Avg_Pay_Ratio_above_.45

Util_Growth_6mo
(Δ in BILL_AMTX/LIMIT_BAL over 6 mohths)

Util_Growth_6mo_below_minus.03
Util_Growth_6mo_below_below_0
Util_Growth_6mo_below_above_0

Max_DLQ
(Max value of PAY_1:PAY_6)

Max_DLQ_above_1
Max_DLQ_below_1

**not used due to cross correlation which results in low variable importance for modeling*

3.3 Transformed data for model analysis

The recommendation is to consider two data sets. **The first data set** (data1) keeps the variables in a continuous format for the purposes of a tree analysis. This is due to the fact that the nature of Random Forest or Gradient Boosting work best using continuous variables, which will naturally split above and below the decision point. The **second data set** (data2) will be used for logisitic regression and principal component analysis due to the natural scaling of the data through the use of dummy indicators. These recommended data fields are all to be included in variable selection, which will inevitably result in variable elimination due to cross correlation.

Table 3.3.1 Recommended Data Sets for Model Development

Data1 (Random Forest & Gradient Boosting)	**Data2** (Logistic Regression & Principal Components Analysis)	
LIMIT_BAL	LIMIT_BAL_below_30k	Avg_Pmt_Amt_below2k
SEX_Female	LIMIT_BAL_above_160k	Avg_Pmt_Amt_above12k
EDUCATION	SEX_Female	Avg_Util_below_.001
MARRIAGE_Y	MARRIAGE_Y	Avg_Util_above_.45
AGE	ED_Grad_Other	PAY_X_Sum_6mo_belowZero
PAY_X_Sum_6mo	AGE_Below_25	PAY_X_Sum_6mo_aboveSix
Avg_Pmt Amt	AGE_25to35	Balance_Growth_6mo < -21k
Avg_Util	AGE_above45	Balance_Growth_6mo < -10k
Avg_Pay_Ratio	Avg_Pay_Ratio_below_.035	Balance_Growtth_6mo > 1k
Balance_Growth_6mo	Avg_Pay_Ratio_above_.113	Util_Growth_6mo < -.03
Util_Growth_6mo	Avg_Pay_Ratio_above_1	Util_Growth_6mo > 0
Max_DLQ	Max_DLQ_above1	
Target (DEFAULT)	Target (DEFAULT)	

4. Exploratory Data Analysis

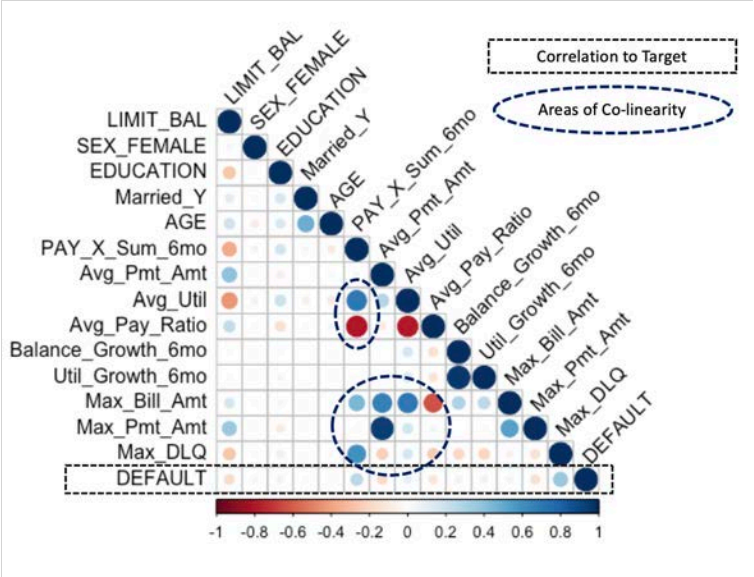
Following data transformation and binning, we can take a deeper look at the variables, their correlation with each other, and their predictive correlation to the response variable.

4.1 Exploratory Data Analysis – Engineered Features

Initial correlation analysis below shows that the highest predictive indicators of default are LIMIT_BAL (-), Avg_Pmt_Amt (-), Max_DLQ (+) & PAY_X_Sum_6mo(+). In the figure and table, variables with relatively high cross correlation values are highlighted.

Figure 4.1.1

Correlation matrix of Variables



Recommendation to remove due to cross correlation:

Max_Pmt_Amt, Util_Growth_6mo, Avg_Pay_Ratio, & Avg_Util

Table 4.1.1

Coefficient values to response

Variables	**DEFAULT**
LIMIT_BAL	-0.15
Avg_Pmt_Amt	-0.10
Max_Pmt_Amt	-0.08
Max_Bill_Amt	-0.04
SEX_FEMALE	-0.04
Balance_Growth_6mo	-0.03
Util_Growth_6mo	-0.02
Avg_Pay_Ratio	-0.01
AGE	0.01
Married_Y	0.03
EDUCATION	0.07
Avg_Util	0.12
PAY_X_Sum_6mo	0.28
Max_DLQ	0.37
DEFAULT	**1**

When we look at the some of the more influential binary variables in a mosaic format, the suggestion is that Max_DLQ or a combination of Marriage_Y and SEX_Female (i.e. married vs. un-married females) may be important variables in modeling.

Figure 4.1.2 –

Mosaic - Max_DLQ to target

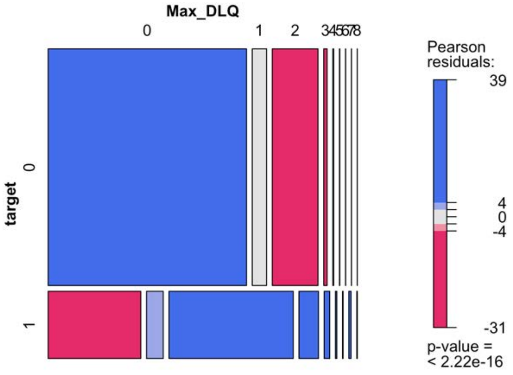
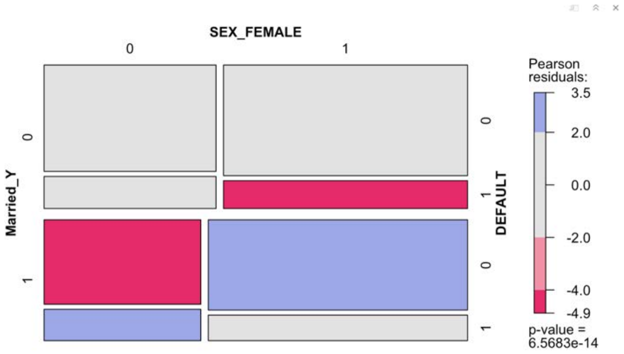


Figure 4.1.3 –

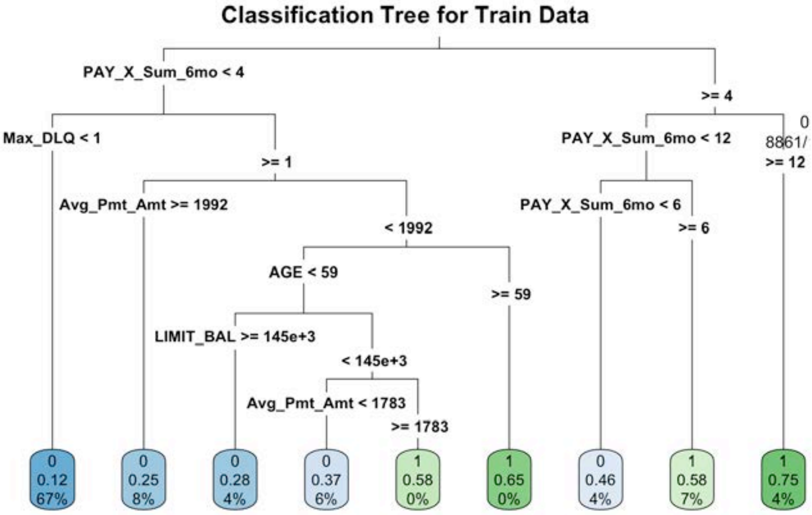
Mosaic – Female default correlation if single



4.2 Exploratory Data Analysis part 2 – Model Based EDA

Initial GLM model and decision tree analysis show the most relevant variables beginning with PAY_X_Sum_6mo, then Max_DLQ, Avg_Pmt_Amt, Age, LIMIT_BAL, and Avg_Pmt_Amt.

Figure 4.2.1 – Exploratory Tree Analysis using rpart



WOE Binning analysis and transformation is shown in appendix 8.2 & 8.3.

5. Predictive Modeling: Methods and Results

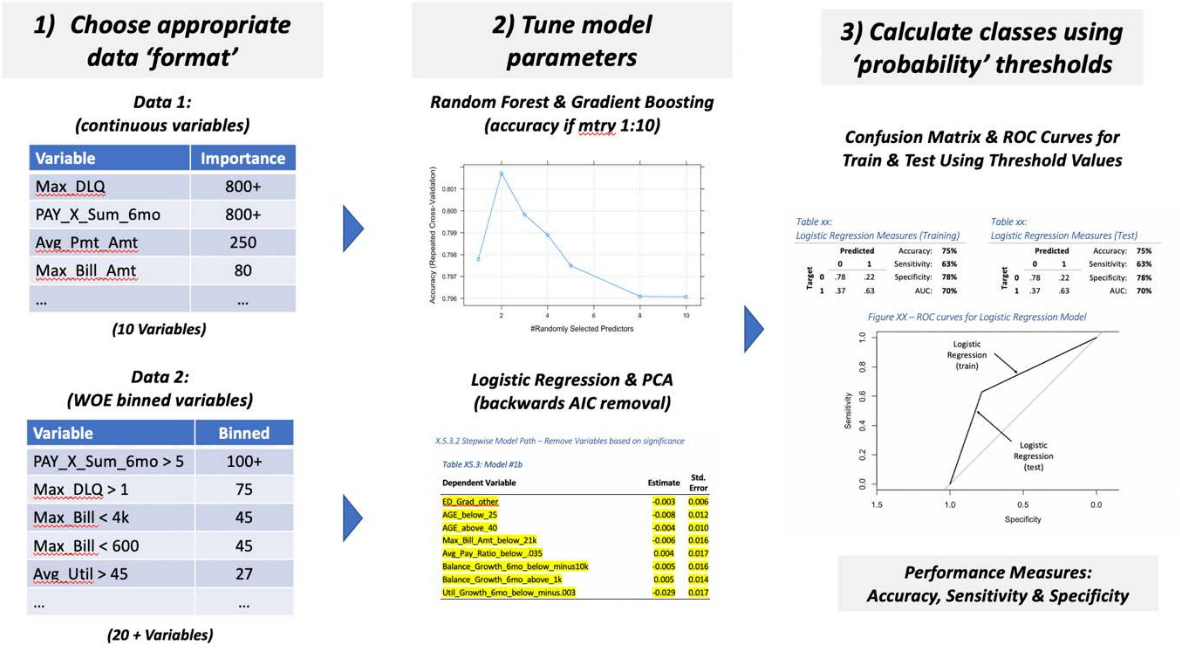
As part of this analysis, we explore four modeling techniques. Two are tree based, and two are a form of regression analysis. Each model will be developed using the 'train' set of data, which has 15,180 observations, and then the model will be tested on the 'test' data set, which as 7,323 observations. Below is the proposed comparison matrix for analysis.

	Type 1Tree Based Approach						Type 2: Regression Models					
	Model 1			Model 2			Model 3			Model 4		
	Random Forest			Gradient Boosting			Logistic Regression			PCANNet		
	Train	Test	Δ	Train	Test	Δ	Train	Test	Δ	Train	Test	Δ
Accuracy	..	..	…	..	..	…	..	..	…	..	..	…
Sensitivity	..	..	…	..	..	…	..	..	…	..	..	…
Specificity	..	..	…	..	..	…	..	..	…	..	..	…
AUC	..	..	…	..	..	…	..	..	…	..	..	…

The following factors will be considered in the model selection**:**

- Reasonable variation between model performance for both ‘train’ & ‘test’
- Model Accuracy and Sensitivity (True Positives) outweigh Specificity (True Neg)
- Consideration for the number of variables included to the model (simplicity)

Here is a visual outline of the model approach that will be used:



1) Choose appropriate data ‘format’

Data 1:
(continuous variables)

Variable	Importance
Max_DLQ	800+
PAY_X_Sum_6mo	800+
Avg_Pmt_Amt	250
Max_Bill_Amt	80
…	…

(10 Variables)

Data 2:
(WOE binned variables)

Variable	Binned
PAY_X_Sum_6mo > 5	100+
Max_DLQ > 1	75
Max_Bill < 4k	45
Max_Bill < 600	45
Avg_Util > 45	27
…	…

(20 + Variables)

2) Tune model parameters

Random Forest & Gradient Boosting
(accuracy if mtry 1:10)

Accuracy (Repeated Cross-Validation)
0.801
0.800
0.799
0.798
0.797
0.796
2
4
6
8
10
#Randomly Selected Predictors

Logistic Regression & PCA
(backwards AIC removal)

X.5.3.2 Stepwise Model Path – Remove Variables based on significance

Table X5.3: Model #1b

Dependent Variable	Estimate	Std. Error
ED_Grad_other	-0.003	0.006
AGE_below_25	-0.008	0.012
AGE_above_40	-0.004	0.010
Max_Bill_Amt_below_21k	-0.006	0.016
Avg_Pay_Ratio_below_.035	0.004	0.017
Balance_Growth_6mo_below_minus10k	-0.005	0.016
Balance_Growth_6mo_above_1k	0.005	0.014
Util_Growth_6mo_below_minus.003	-0.029	0.017

3) Calculate classes using ‘probability’ thresholds

Confusion Matrix & ROC Curves for
Train & Test Using Threshold Values

Table xx:
Logistic Regression Measures (Training)

Target		Predicted 0	Predicted 1		
	0	.78	.22	Accuracy:	75%
				Sensitivity:	63%
				Specificity:	78%
	1	.37	.63	AUC:	70%

Table xx:
Logistic Regression Measures (Test)

Target		Predicted 0	Predicted 1		
	0	.78	.22	Accuracy:	75%
				Sensitivity:	63%
				Specificity:	78%
	1	.37	.63	AUC:	70%

Figure XX – ROC curves for Logistic Regression Model

Performance Measures:
Accuracy, Sensitivity & Specificity

As explained in section 3.3, step one is a transformation of specific data sets for the model types based on the nature of the analysis – tree vs. regression. For the tree-based models, we will use fewer variables in a continuous format, while for regression, we break the continuous variables into bins based on our weight of evidence analysis in the exploration stage.

In the second stage we explore how to tune the training model parameters. In the case of the tree models, we look at variable importance, and run tuning formula to plot the comparative accuracy using the ‘mtry’ parameter, which limits the number of random variables the tree will try to split at any new branch. In the case of the two regression models, we will first run a variable selection analysis to remove any variables that do not have statistical significance, and in the final model we reduce the number of variables further, only keeping the ones with the most 'importance'.

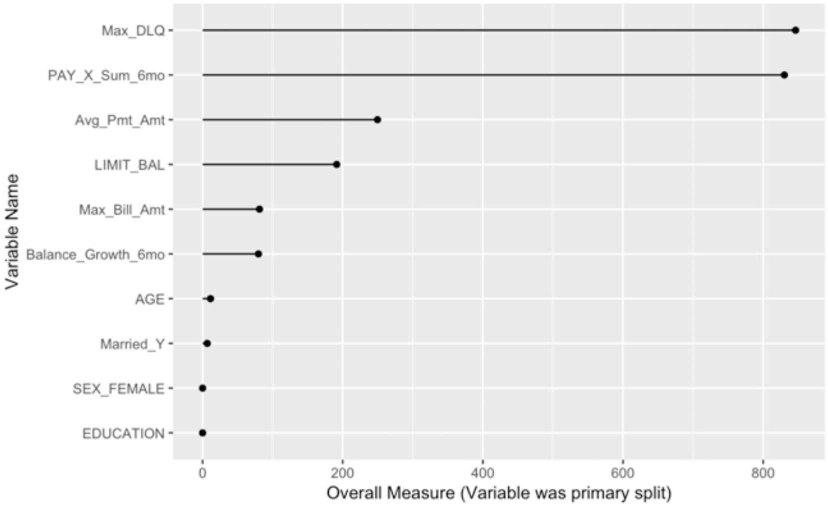
5.1 Random Forest Model (Model 1)

The development of the Random Forest model uses the caret package in R with the method = 'RF'. As described above, the model development covers three steps: first, the review of variable importance from the initial tree analysis in the prior section, then 'tuning' the model to select the right number of random variables selected for each branch (mtry), and finally calculating the 'class' variables using the roc threshold and likelihood of default. Details of the R code used to train and predict default 'class' variables is in Appendix 8.5.1.

5.1.1: review variable importance from initial tree EDA (section 4.2):

Step one of the model consideration looks at the importance of each variable (the number of times it is the primary factor in a decision branch). As suggested in figure 4.2.1, the tree models will lean heavily on a few select variables for prediction of the outcome. Plotting the importance levels shows that the tree approach will primarily consider the following variables: Max_DLQ, PAY_Amt_Sum_6mo, Limit_Bal, Max_Bill_Amt.

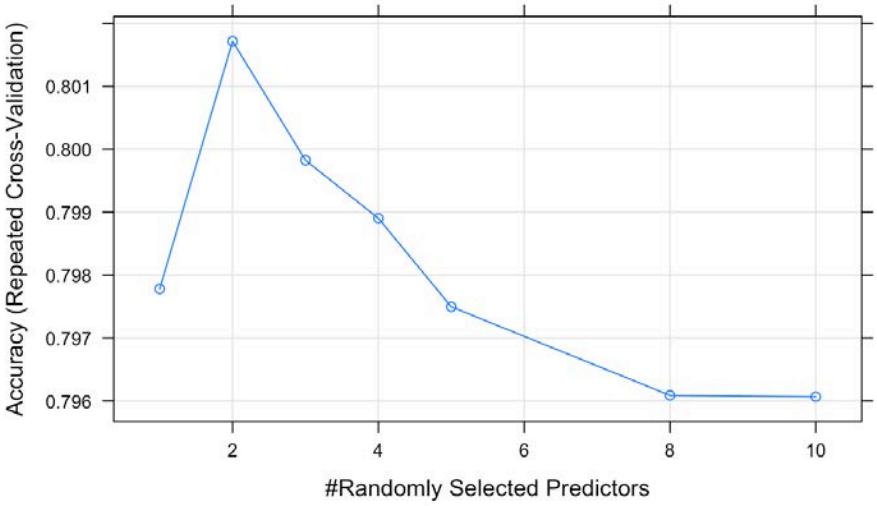
Figure 5.1.1 – Importance of Variables in Model



5.1.2: tuning the training model for Accuracy based on parameter 'mtry'

The second step of the process is to consider how many variables the model will select when testing the next set of predictors in each branch. As we will see, the tendency for the random forest model is to over-fit the data to the training set, so it is important to limit the scope of fit. In this case, we use the parameter 'mtry', which translates into the 'tuneLength' variable when we call the model. When we plot the accuracy, the recommended length will be 2, as we can clearly see a dip in performance as the number of potential variables increases:

Figure 5.1.2 – Model tuning – Mtry Accuracy



5.1.3: Predicting 'Class' using model threshold & measuring performance

The final stage in the process is to use the model threshold to predict the target 'class' using the probability values of the prediction. Below we can see the confusion matrix which shows the 'actual' target values on the Y axis, and the model predictions on the X axis. As we can see, the training set is highly accurate (98.5%), while the test set is much less accurate (77%). This is a classic example of over-fitting.

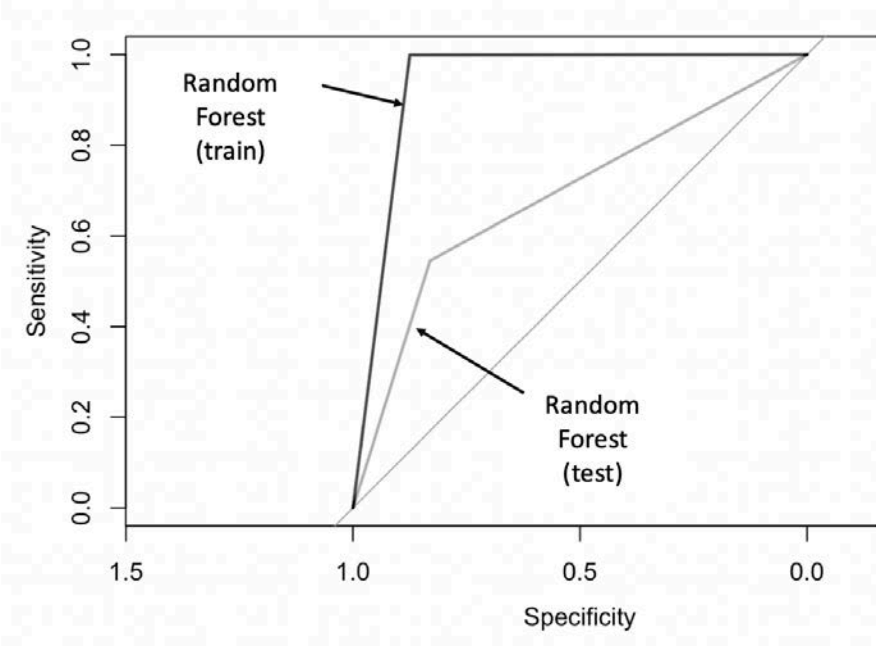
Table 5.1.3.1a:
Random Forest Measures (Training)

		Predicted		Accuracy:	**98.5%**
		0	**1**	Sensitivity:	**99%**
Target	**0**	.98	.02	Specificity:	**98%**
	1	.01	.99	AUC:	**94%**

Table 5.1.3.1b:
Random Forest Measures (Test)

		Predicted		Accuracy:	**77%**
		0	**1**	Sensitivity:	**55%**
Target	**0**	.83	.17	Specificity:	**83%**
	1	.45	.55	AUC:	**68%**

Figure 5.1.3.1 – ROC curves for Random Forest Model



In this example, we see a high amount of variation between the training ROC, which is almost a perfect square, and the test ROC. The 'Area Under Curve' (AUC) metric drops by almost a third. In this case, it appears that the Random Forest approach in and of itself will not be appropriate. In the next section, we consider Gradient Boosting as an alternative.

5.2 Gradient Boosting (Model 2)

One way to normalize the Random Forest approach is to assign a cost function to the addition of the new variables, and constantly search for the approach with the least amount of loss. In this respect, the initial approach to the model development is very similar to the Random Forest approach. It uses the same data set with continuous variables instead of binned ones, and it assumes the same starting point when it comes to variable importance and parameters such as ‘tuneLength’.

5.2.1: Variable Importance, parameter tuning & class prediction

In respect to development of the Gradient Boosting model, we will use the same assumptions as the Random Forest model. The primary variables for consideration are: Max_DLQ, PAY_Amt_Sum_6mo, Limit_Bal, Max_Bill_Amt. Also, the tuning analysis that was done in respect to Model 1 will also apply. The recommended ‘tuneLength’ variable is 2, meaning that the model will test two random variables at each branch split. The final stage in the process is to fit the model using the parameters above and predict the target ‘class’ using the probability values of the fitted model. The R code for this stage can be found in Appendix 8.5.2.

5.2.2: Model 2: Gradient Boosting - measuring performance

Below we can see the confusion matrix which shows the ‘actual’ target values on the Y axis, and the model predictions on the X axis. As we can see, the training set accuracy (75%) is lower in respect to the first model. This suggests that it is performing better than the Random Forest approach. Model 2 is not showing signs of over-fitting the data as in the previous model, but the predictions on the test set are slightly less accurate (77%).

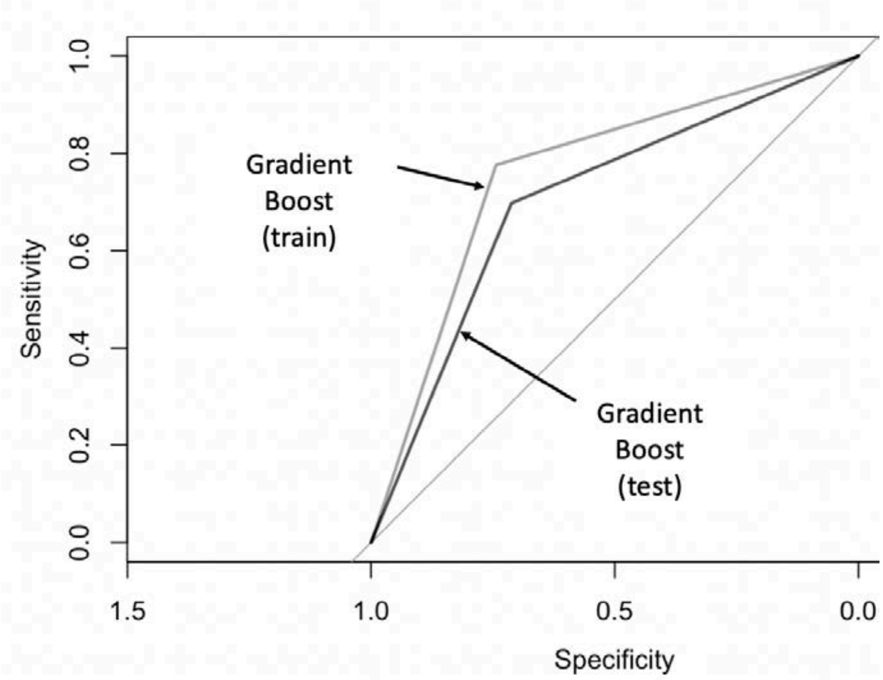
Table 5.2.2.1a:
Gradient Boosting Measures (Training)

		Predicted		Accuracy:	**75%**
		0	**1**	Sensitivity:	**77%**
Target	**0**	.74	.26	Specificity:	**74%**
	1	.23	.78	AUC:	**76%**

Table 5.2.2.1b:
Gradient Boosting Measures (Test)

		Predicted		Accuracy:	**71%**
		0	**1**	Sensitivity:	**70%**
Target	**0**	.71	.29	Specificity:	**71%**
	1	.30	.70	AUC:	**70%**

Figure 5.2.2.2 – ROC curves for Gradient Boosting Model



In respect to the Random Forest model, the Gradient Boosting approach is a great improvement. The Sensitivity metric will be the best out of the four model techniques, and one could argue that in the case of banks predicting actual defaults, sensitivity is the most important measure. However, there are some other factors to consider, such as accuracy, model complexity and specificity.

We can see in the ROC curve, there is still some variation between the training and test performance, resulting in a loss of AUC of about 6%. Another thing to consider as we look at the performance compared to regression modeling techniques is how to reproduce these models in practice. It is often thought that tree-based approaches are more complex than traditional regression, and in this case the model performance is roughly in line with the prediction capabilities of Models 3 & 4.

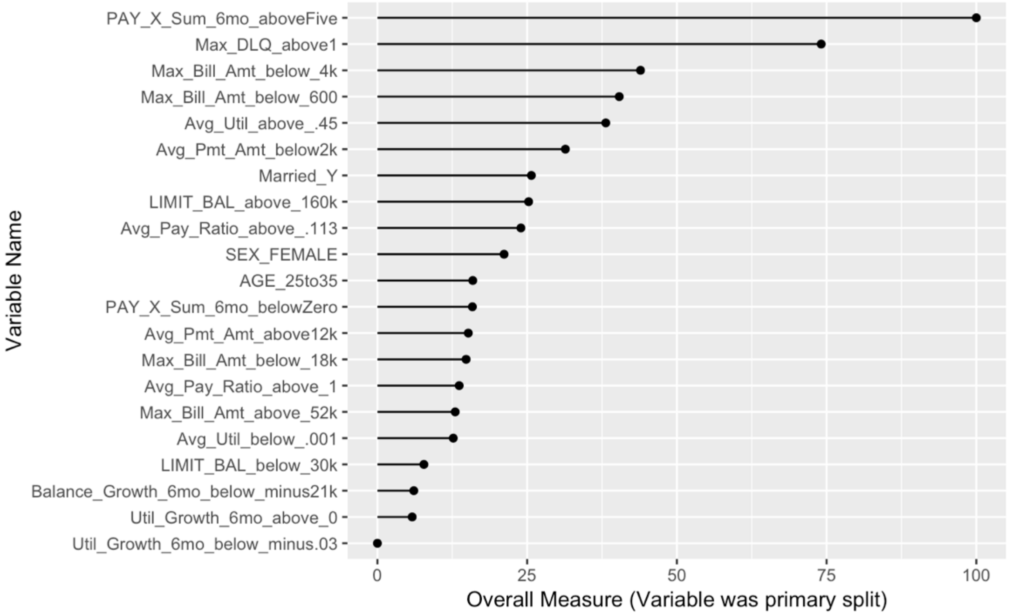
5.3 Logistic Regression with Variable Selection (Model 3)

As outlined in the exploratory data analysis and transformation, the recommended data set for the regression models is different than the data used for the trees. At the core, they are the same, but instead of continuous variables, each variable that is not binary is split into a dummy 'bin'. The specifics of the recommended binning can be found in the sections 2 & 3 – Data Overview & Feature Engineering and related appendices 8.2 & 8.3.

5.3.1: Variable importance and selection

Initial review of the variable importance relative to the response shows that it is still similar variables that have the most significance. Appendix 8.5.3 has additional detail on the entire data set and the statistical significance in correlation to the response variable.

Figure 5.3.1 – Importance of Variables in Model



Ultimately, this will be too many variables for the model, so the next step is to reduce them using a backwards AIC analysis *as outlined in Table 8.5.3.2.*

5.3.2: Training the model and using prediction scores to assign 'class'

Using the resulting data set after removing the recommended variables, we will train the model using caret method = "glm" and a train control function that was used prior: TrainControl(method="repeatedcv", number=10, repeats=2, search="random").

Appendix 8.5.3 shows the detailed code for the prediction of class using the probability of the fitted model, and then the creation of the confusion matrices and accuracy measures below. The fitted glm model is used to assign a probability and class variable for the test set as well. The results are compared below.

5.3.3: Model 3: Logistic Regression - measuring performance

In this case, it appears that the variation between the training data set and the test data set has been minimized relative to the tree methods. The two seem to perform very similarly. This is one of our important measurement criteria and will be taken into account during the model selection.

The sensitivity measure for the test data set is relatively improved in comparison to the Random Forest model without too much loss in accuracy. There is some loss in specificity, but in the case of the bank, it could be argued that the sensitivity measure plays more of a role in model selection due to the importance of predicting accurately cases of potential default over cases of non-default.

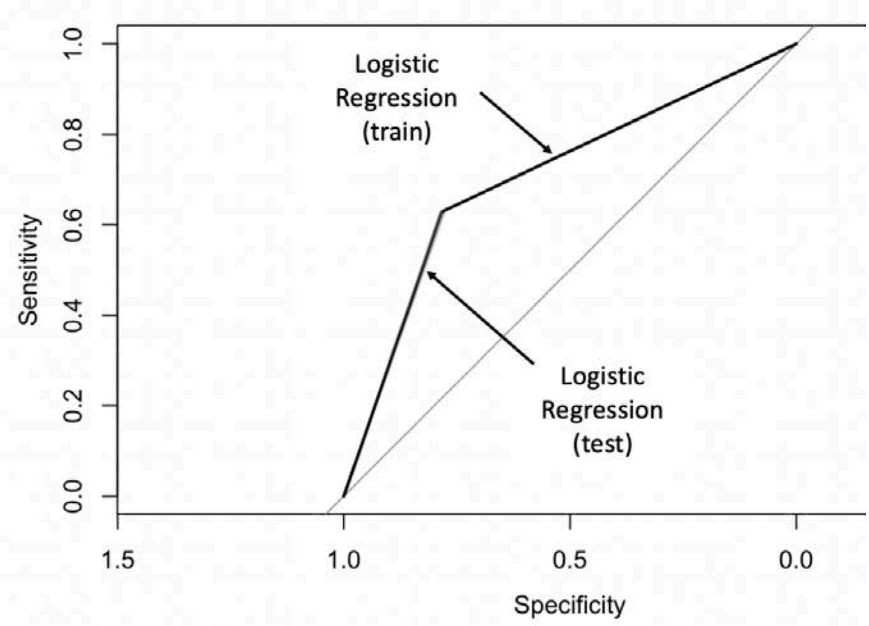
Table 5.3.3.1a:
Logistic Regression Measures (Training)

		Predicted		Accuracy:	**75%**
		0	**1**	Sensitivity:	**63%**
Target	**0**	.78	.22	Specificity:	**78%**
	1	.37	.63	AUC:	**70%**

Table 5.3.3.1b:
Logistic Regression Measures (Test)

		Predicted		Accuracy:	**75%**
		0	**1**	Sensitivity:	**63%**
Target	**0**	.78	.22	Specificity:	**78%**
	1	.37	.63	AUC:	**70%**

Figure 5.3.3.1 – ROC curves for Logistic Regression Model



As we see in the ROC curve, the models perform consistently, and there is no differentiation between train performance and test performance. This suggests that the model is a good fit to the data and responds well to uncertainty. In the final model, we will look into the opportunity to reduce the variable count even further, from 21 to 15 using Principal Component Analysis and Neural Network modeling. (PCANNet)

5.4 Principal Component Analysis & Neural Network (Model 4)

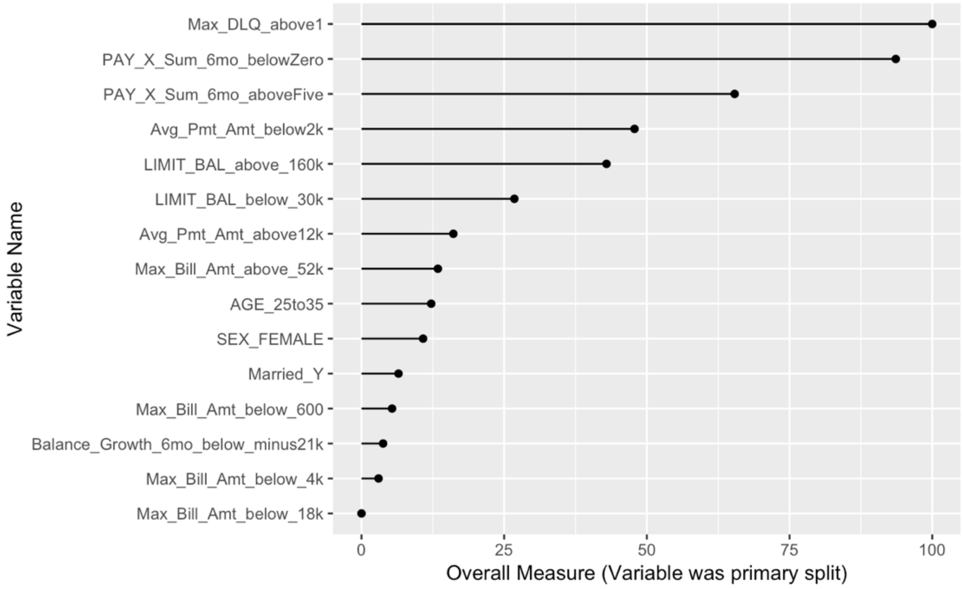
In the final model we will consider dimension reduction using Principal Component Analysis (PCA). PCA calculates the primary variance across the variables, and then calculates how many components are required to keep the variance in the model. Once the model has determined the correct variance between the elements, it then uses the principal components in a neural network model. The result is a model with reduced parameters that retains the predictability of the original model. As the data set for the logistic regression is already scaled (it only has binary variables of 0 or 1), then the data does not need to be transformed as it would if the data were continuous variables not scaled to one another.

5.4.1: Variable importance and selection

In the fourth model, we start with a reduced data set to begin with. As model simplicity is a factor, we should push down the number of variables used. We will take a chance and remove an additional 6 variables from the logistic regression model to see what happens.

The way we select which to remove is by reviewing which variables demonstrate lowest importance in table 5.3.1 as well as variables with high cross correlation, such as Avg_Util & Max_Pay_Amt. In the end, we are left with 16 variables compared to the original 29 prior to the first wave of variable selection. (table 8.5.3)

Figure 5.4.1 – Importance of Variables selected (Model 4 - PCANNet)



***Detail on the R code for model development of Model 4 can be found in Appendix 8.5.4**

5.4.2: Model 4: PCANNet - measuring performance

For the final iteration it appears that the variation between the training data set and the test data is still kept to a minimum despite removing ~28% of the variables used. Almost every performance measure is the same as the prior model, but the model has been simplified by using the principal components analysis. The sensitivity measure (78%) for the test data set has not been impacted, and the accuracy is the same.

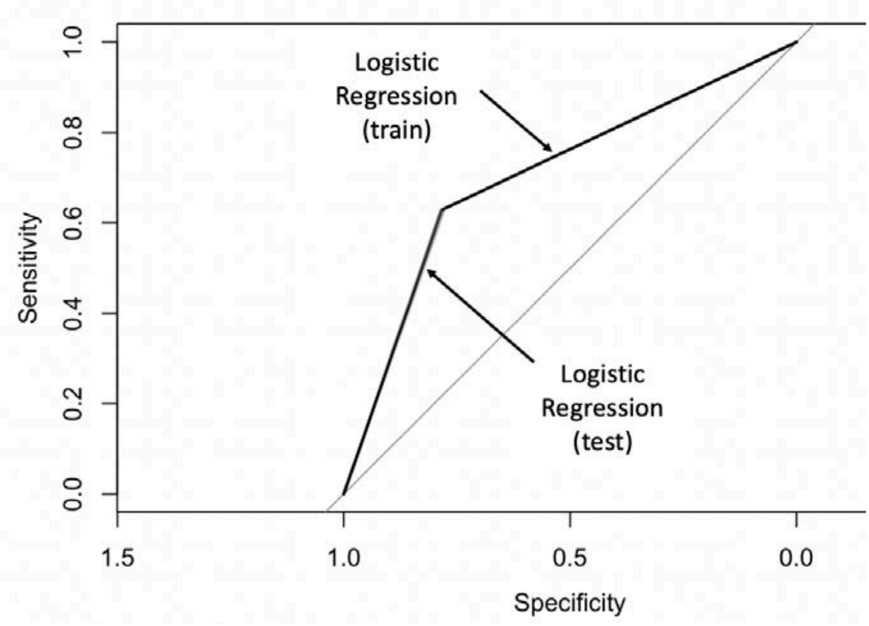
Table 5.4.2.1a:
PCANNet Measures (Training)

		Predicted		Accuracy:	**75%**
		0	**1**	Sensitivity:	**64%**
Target	**0**	.77	.23	Specificity:	**77%**
	1	.36	.64	AUC:	**71%**

Table 5.4.2.1b:
PCANNet Measures (Test)

		Predicted		Accuracy:	**75%**
		0	**1**	Sensitivity:	**63%**
Target	**0**	.77	.23	Specificity:	**78%**
	1	.36	.64	AUC:	**71%**

Figure 5.4.2.1 – ROC curves for Logistic Regression Model



Considering that this model has fewer variables than the logistic regression model, this will be an important factor in the final decision and recommendation. The two perform almost exactly the same, but the relative variable count for Model 4 (15) is significantly less than Model 3 (21). Considering that every other performance measure is the same, it seems that the most logical recommendation would be to use model 4 – PCANNet as it would be easier to put into practice and it has fewer variables to over-fit.

6. Comparison of Results

The table below is a summary of the comparative performance measures from Model 1 to Model 4. In section 5, we outlined the the most important to consider:

- Reasonable variation between model performance for both ‘train’ & ‘test’
- Model Accuracy and Sensitivity (True Positives) outweigh Specificity (True Neg)
- Consideration for the number of variables included to the model (simplicity)

In consideration of these factors, it appears that Model 4 meets the criteria best relative to the others. We began with the Random Forest (Model 1), which tended to over-fit the training data, and did not respond as well in a test environment. The Gradient Boost (Model 2) approach was an improvement, but the accuracy levels still fell short relative to the regression approach in Model 3 & 4.

Table 6.1.1 – Comparison of performance metrics Model 1:4

	Model 1			**Model 2**			**Model 3**			***Model 4**		
	Random Forest			Gradient Boosting			Logistic Regression			**PCANNet**		
	Train	**Test**	**Δ**	**Train**	**Test**	**Δ**	**Train**	**Test**	**Δ**	**Train**	**Test**	**Δ**
Accuracy	98%	77%	(22%)	75%	71%	(4%)	75%	75%	-	75%	75%	-
Sensitivity	99%	55%	(44%)	77%	70%	(7%)	63%	63%	-	64%	63%	(1%)
Specificity	98%	83%	(15%)	74%	71%	(3%)	78%	78%	-	77%	78%	1%
AUC	94%	68%	(36%)	76%	70%	(6%)	70%	70%	-	71%	71%	-

In comparison to the tree models, Logistic Regression (Model 3) seemed to perform very well. This model was the first one to use a different approach to the data preparation, which considered pre-binned binary classifications instead of continuous variables. The measured predictions on the training data are equal to those on the test data, which is evidence of a well fit model. Model 3 has lower performance in the Sensitivity metric (-7%)

in comparison to Model 2, but it is made up in the Specificity measure (+7%), resulting in an improved accuracy measure.

As we discussed in the problem overview, high sensitivity is an important measure for banking. It seems more profitable for a bank to accurately predict a default scenario than a non-default one. Therefore sensitivity is an important element to consider, but it is not the only one. When we look across the model performance, it seems that there is an argument to be made for lower sensitivity for the improved model performance in terms of accuracy, specificity and simplicity.

With respect to these considerations, the fourth model, which uses a pairing of Principal Component Analysis and Neural Networking, appears to be the optimal solution. It has a relatively high sensitivity, specificity and accuracy and it performs with equal performance measures as the Logistic Regression approach (Model 3). In addition, it uses a third less variables in the model, meaning that it will be easier to implement and have less reason for over-fitting due to variables that do not statistically influence the outcome.

7. Conclusions

If we return to the beginning of this journey, the problem we are trying to solve is to save banks money and risk through early identification of clients who may potentially default on a credit loan. We had calculated through the example data set that a 1% reduction in customer default was the equivalent of ~3.2M NTD ($100k USD). Through data exploration and predictive modeling, we were able to determine some of the predictive indicators of high likelihood of default: consistent behavior to pay bills on time, paying a bill in full each month, keeping credit limits in check, and demographic considerations such as age, marital status, and gender.

To develop the highest performing model, we explored a number of different modeling techniques. The first two models were ‘tree’ based approach, and the last two used linear regression. The first model, Random Forest, tended toward over fitting, while Gradient Boosting technique improved on the approach. Ultimately, the regression models using variable selection seemed to outperform the tree ones.

Each model used data developed through exploration and transformation, but the tree approach kept variables in a continuous variable format, while the regression approach used ‘one hot’ encoding and weight of evidence binning for each variable. Gradually through variable elimination and improved model tuning, we were able to achieve a model that performed relatively well with almost half as many variables as we began with.

The results of the modeling seem fair even though we never achieved an accuracy above 80%. This would likely require more time and expertise tuning the parameters of the training models: the number of trees, the number of branches, the number of variables and folds. In addition to further variable development and reduction, these are all elements that could be further explored to push the model performance even further.

One last consideration in terms of future development would be the original data itself. It is partially a problem of data accuracy, and more holistically a problem of model bias. Through exploration outlined in Section 2.3, it appeared there were some outlier instances when the default variable and the actual observed payments did not agree. Data capturing methods may be a factor in model accuracy.

In addition, if we consider the data set origins and the influence of variables such as age, gender and marital status, there may be a concern with the objectivity of the overall approach. To start with, the data set is only a representation of individuals who were approved for credit applications, not those who were rejected for a loan. This is a biased observation to begin with. If the data included applications that were rejected, and their default history following initial application then there may be a more robust picture.

Finally, the role of demographic information in the model process is a concern. If we consider the influence of variables such as age, gender and marital status, the outcome of the resulting model may result in wider, un-favorable social consequences. Should only younger males who are married be approved for loans? If so, wouldn't that further feed into the model that only the 'right' people should be approved? These types of social considerations are a slippery slope as we approach a world of data models that make decisions like who receives credit approval over who does not.

Overall, the data recommended data models as a result of this study present a realistic way to predict the likelihood of credit default and might be used to save a lot of lost revenue, as has been demonstrated. We simply need to take into consideration some of the wider implications of the modeling process, and if there is a way we can counteract the tendency for such models to systematically exclude entire groups of people, which is especially important when we consider the role that access to credit plays in overall health and well-being of society.

8. Appendices

8.1 Summary Statistics for Credit Card Default Original Data

Below are the summary statistics for the original data with highlighted discrepancies relative to the proposed data description. Detailed description is in sections 2.2 & 2.3.

Table 8.1.1: Summary Statistics for Credit Card Default

Statistic	N	Mean	St. Dev.	Min	Pctl(25)	Median	Pctl(75)	Max
ID	30,000	15,000.50	8,660.40	1	7,500.8	15,000.5	22,500.2	30,000
LIMIT_BAL	30,000	167,484.30	129,747.70	10,000	50,000	140,000	240,000	1,000,000
SEX	30,000	1.60	0.49	1	1	2	2	2
EDUCATION	30,000	1.85	0.79	0	1	2	2	6
MARRIAGE	30,000	1.55	0.52	0	1	2	2	3
AGE	30,000	35.49	9.22	21	28	34	41	79
PAY_1	30,000	-0.02	1.12	-2	-1	0	0	8
PAY_2	30,000	-0.13	1.20	-2	-1	0	0	8
PAY_3	30,000	-0.17	1.20	-2	-1	0	0	8
PAY_4	30,000	-0.22	1.17	-2	-1	0	0	8
PAY_5	30,000	-0.27	1.13	-2	-1	0	0	8
PAY_6	30,000	-0.29	1.15	-2	-1	0	0	8
BILL_AMT1	30,000	51,223.33	73,635.86	-165,580	3,558.8	22,381.5	67,091	964,511
BILL_AMT2	30,000	49,179.08	71,173.77	-69,777	2,984.8	21,200	64,006.2	983,931
BILL_AMT3	30,000	47,013.15	69,349.39	-157,264	2,666.2	20,088.5	60,164.8	1,664,089
BILL_AMT4	30,000	43,262.95	64,332.86	-170,000	2,326.8	19,052	54,506	891,586
BILL_AMT5	30,000	40,311.40	60,797.16	-81,334	1,763	18,104.5	50,190.5	927,171
BILL_AMT6	30,000	38,871.76	59,554.11	-339,603	1,256	17,071	49,198.2	961,664
PAY_AMT1	30,000	5,663.58	16,563.28	0	1,000	2,100	5,006	873,552
PAY_AMT2	30,000	5,921.16	23,040.87	0	833	2,009	5,000	1,684,259
PAY_AMT3	30,000	5,225.68	17,606.96	0	390	1,800	4,505	896,040
PAY_AMT4	30,000	4,826.08	15,666.16	0	296	1,500	4,013.2	621,000
PAY_AMT5	30,000	4,799.39	15,278.31	0	252.5	1,500	4,031.5	426,529
PAY_AMT6	30,000	5,215.50	17,777.47	0	117.8	1,500	4,000	528,666
DEFAULT	30,000	0.22	0.42	0	0	0	0	1
u	30,000	0.50	0.29	0.0000	0.25	0.49	0.75	1.00
train	30,000	0.51	0.50	0	0	1	1	1
test	30,000	0.24	0.43	0	0	0	0	1
validate	30,000	0.25	0.43	0	0	0	0	1
data.group	30,000	1.74	0.83	1	1	1	2	3

8.2 Original Data – Quality, Distributions & Recommended Binning

This section is the plots and tables used to support the 'Data Overview and Quality Check' in section 2 as well as recommended binning for continuous variables. Appendix 9.3 provides specific detail on variable transformation based on these observations and subsequent Exploratory Data Analysis (EDA) in section 3 of this paper.

8.2.1 LIMIT_BAL Variable

Figure 9.2.1.1 Histogram of LIMIT_BAL

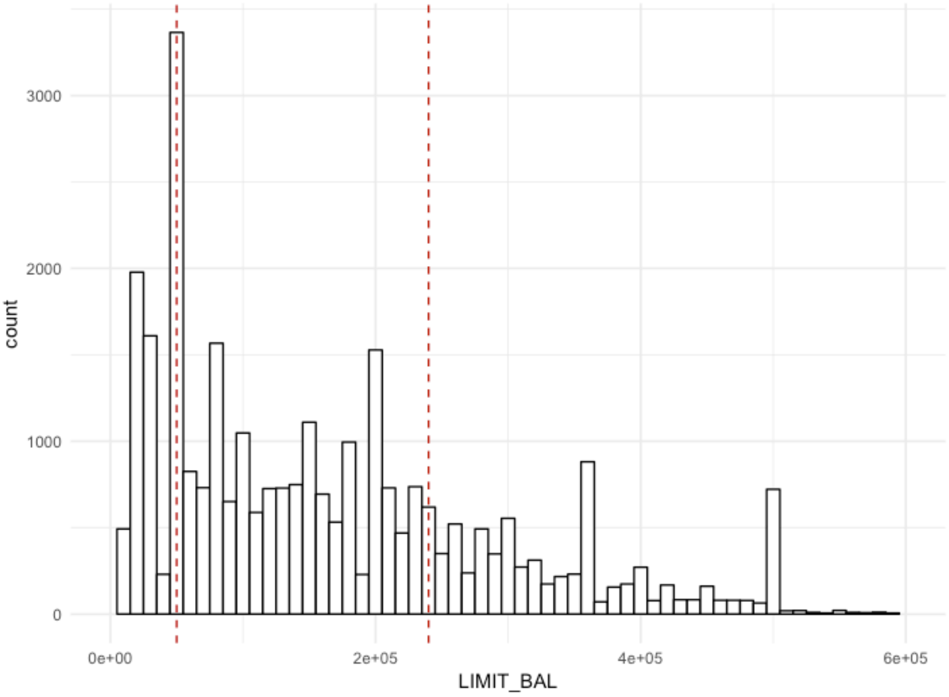


Table 8.2.1.1: Summary Statistics of LIMIT_BAL

Statistic	N	Mean	St. Dev.	Min	Pctl(25)	Median	Pctl(75)	Max
LIMIT_BAL	30,000	167,484.30	129,747.70	10,000	50,000	140,000	240,000	1,000,000

Table 8.2.2.2: WOE Analysis of LIMIT_BAL

Final Bin	Total Count	Total Distr.	1.Rate	WOE	IV
< = 30000	4,081	13.6%	35.8%	-67.7	0.073
< = 160000	13,013	43.4%	24.5%	-13.1	0.008
< = Inf	12,906	43.0%	15.4%	44.3	0.074
Total	**30,000**	**100.0%**	**22.1%**	**NA**	**0.155**

8.2.2 Demographic data: SEX, EDUCATION, MARRIAGE, & AGE

Figure 8.2.1 Histogram of Demographic Variables

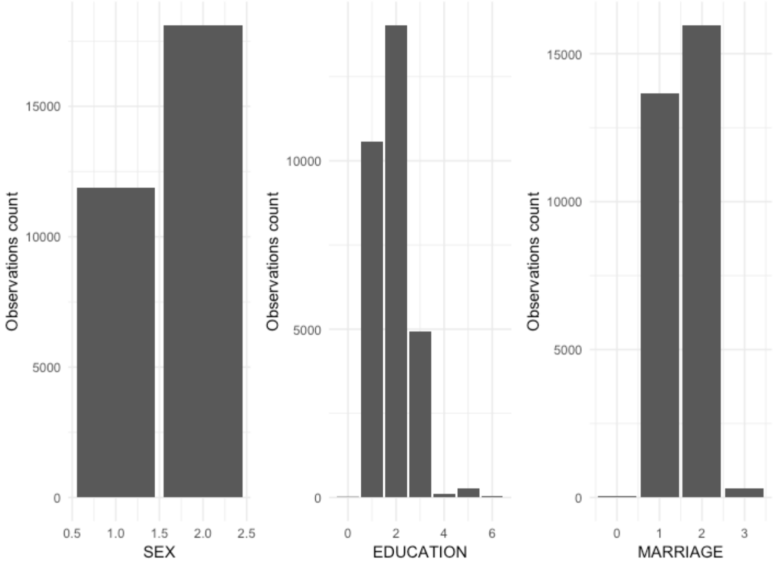


Table 8.2.2.1: Summary Statistics of Demographic Variables: SEX, EDUCATION & MARRIAGE

Statistic	N	Mean	St. Dev.	Min	Pctl(25)	Median	Pctl(75)	Max
SEX	30,000	1.60	0.49	1	1	2	2	2
EDUCATION	30,000	1.85	0.79	0	1	2	2	6
MARRIAGE	30,000	1.55	0.52	0	1	2	2	3

Analysis of EDUCATION categories - dimension reduction

Table 8.2.2.2a:

Summary of EDUCATION

EDUCATION	Freq	PCT
0	14	0%
1	10,585	35.3%
2	14,030	46.8%
3	4,917	16.4%
4	123	0.4%
5	280	0.9%
6	51	0.2%

* group 0,4:6 as ‘Other’ = 4

Table 8.2.2.2b:

Education Correlation

Group	DEFAULT
grad	-0.05
univ	0.04
high	0.03
other	-0.05

Frequency counts of education variable show 3 un-defined categories. Recommenation is to group them all as '0' – Other. Correlation analysis suggests binning for Grad/Other and one for High School/University. Weight of Evidence (WOE) analysis below supports this idea (in table below 'other' is changed to value of 0 instead of 4 as originally defined)

Table 8.2.2.3: Recommended Education Bins

	Final.Bin	Total.Count	Total.Distr.	0.Count	1.Count	0.Distr.	1.Distr.	1.Rate	WOE	IV
1	< = 1	11,053	36.8%	8,984	2,069	38.5%	31.2%	18.7%	21.0	0.015
2	< = Inf	18,947	63.2%	14,380	4,567	61.5%	68.8%	24.1%	-11.2	0.008
4	Total	30,000	100.0%	23,364	6,636	100.0%	100.0%	22.1%	NA	0.023

Analysis of MARRIAGE categories - dimension reduction

Initially, any '0' values are changed to '3' = Other. But following analysis of correlation to the response variable, the differentiation of this class seems insignificant. Looking at Table X.2.3b, recommendation is to group 'Other' into 'Married_N'.

Table 8.2.2.4a: Correlation Matrix
(Married Y/N/Other)

	DEFAULT
Married_Y	0.0298
Married_N	-0.0306
Married_Other	0.0040

Table 8.2.2.4b: Correlation Matrix
(Married Y/N – drop 'other')

	DEFAULT
Married_Y	0.0298
Married_N	-0.0298

Analysis of AGE variable and consideration of dimension reduction

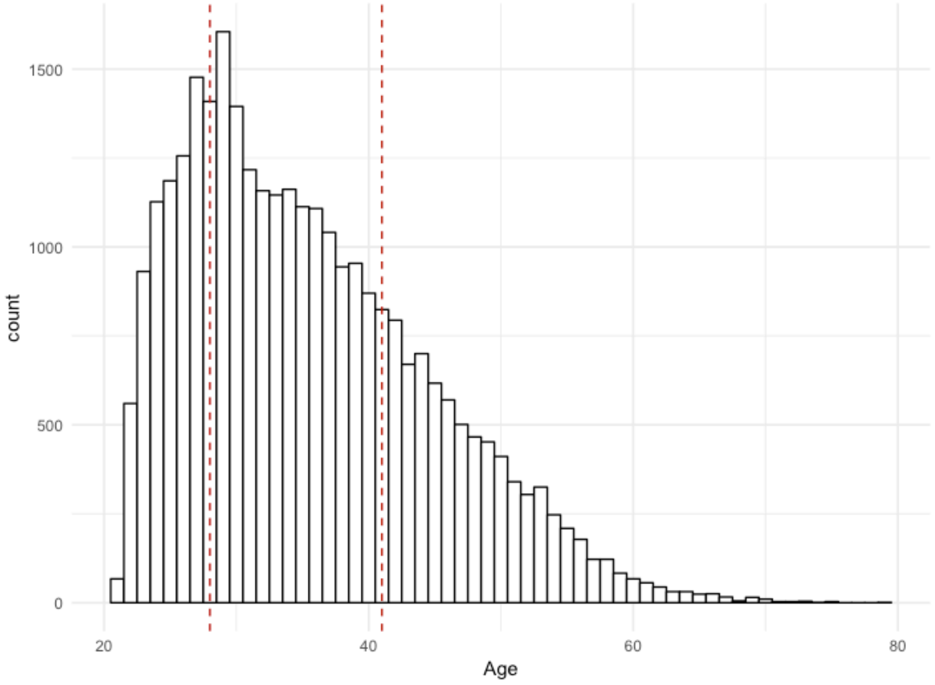


Figure 8.2.2.2 Histogram of AGE

Table 8.2.2.5: Summary Statistics of AGE

Statistic	N	Mean	St. Dev.	Min	Pctl(25)	Median	Pctl(75)	Max
AGE	30,000	35.49	9.22	21	28	34	41	79

Table 8.2.2.6:
Correlation of age bins

	DEFAULT
AGE_below_25	0.04
AGE_25to35	-0.05
AGE_35to45	-0.004
AGE_above_45	0.03

Below is WOE analysis of AGE variable, which suggests grouping the variable into four groups: <=25, 25:35, 35:45, and over 45, which will be the recommended bins for transformation. In the table to the left is an analysis of the correlation to the response variable by group.

Table 8.2.2.7: WOE analysis of Age Bins

AGE Bins	Total Count	Total Dist	Count (0)	Count (1)	Age (1) Rate	WOE	IV
< = 25	3,871	12.9%	2,839	1,032	26.7%	-24.7	0.008
< = 35	12,938	43.1%	10,373	2,565	19.8%	13.9	0.008
< = 45	8,522	28.4%	6,661	1,861	21.8%	1.6	0.000
< = Inf	4,669	15.6%	3,491	1,178	25.2%	-17.2	0.005
Total	**30,000**	**100.0%**	**23,364**	**6,636**	**22.1%**	**NA**	**0.021**

9.2.3 PAY_X Distribution and WOE Binning Analysis

Figure 8.2.3.1 Histogram of PAY_1 to PAY_6

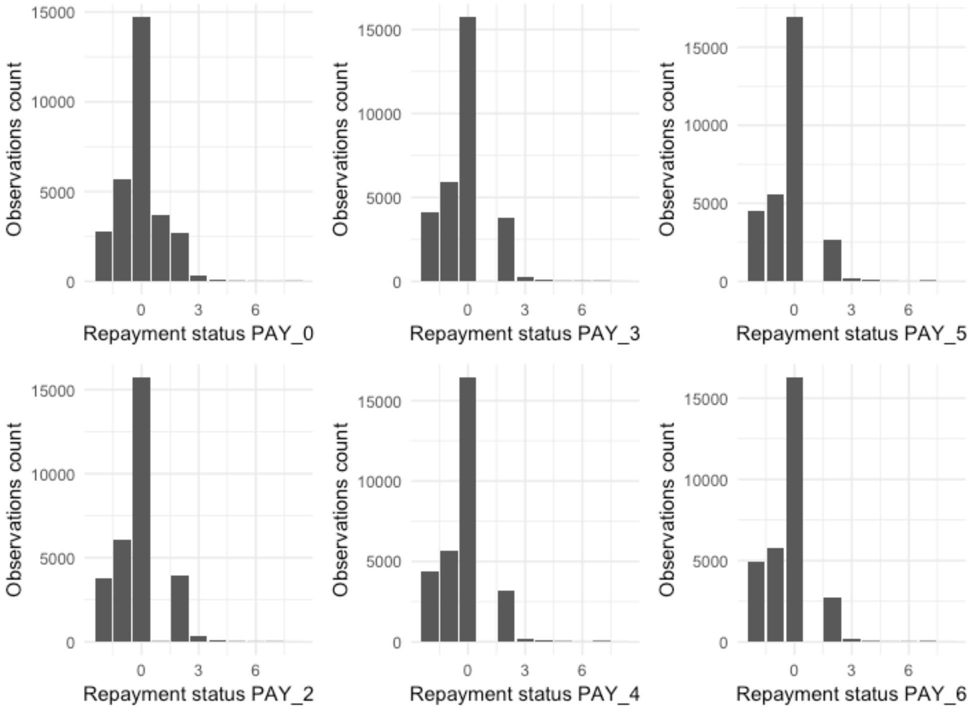


Table 8.2.3.1:
Correlation: PAY_X

	DEFAULT
PAY_1	0.32
PAY_2	0.26
PAY_3	0.23
PAY_4	0.21
PAY_5	0.20
PAY_6	0.18

PAY_X variables show high levels of cross correlation. The recommendation is to create a variable that is the sum of all PAY_X variables (PAY_X_Sum_6mo), and then bin the result into 3 groups as per the optimal distribution against response variable (Table 9.2.3.2)

Table 8.2.3.2: WOE Analysis of PAY_X_Sum_6mo Binning

Final Bin	**Total Count**	**Total Distr.**	**Rate (1)**	**WOE**	**IV**
< = 0	22,867	76.2%	13.8%	57.2	0.210
< = 5	3,950	13.2%	37.1%	-73.1	0.128
< = Inf	3,183	10.6%	63.2%	-179.9	0.396
Total	**30,000**	**100.0%**	**22.1%**	**NA**	**0.735**

8.2.4 BILL_AMTX – Distribution & WOE Binning Analysis

Figure 8.2.4.1 Histogram of BILL_AMT1 to BILL_AMT6

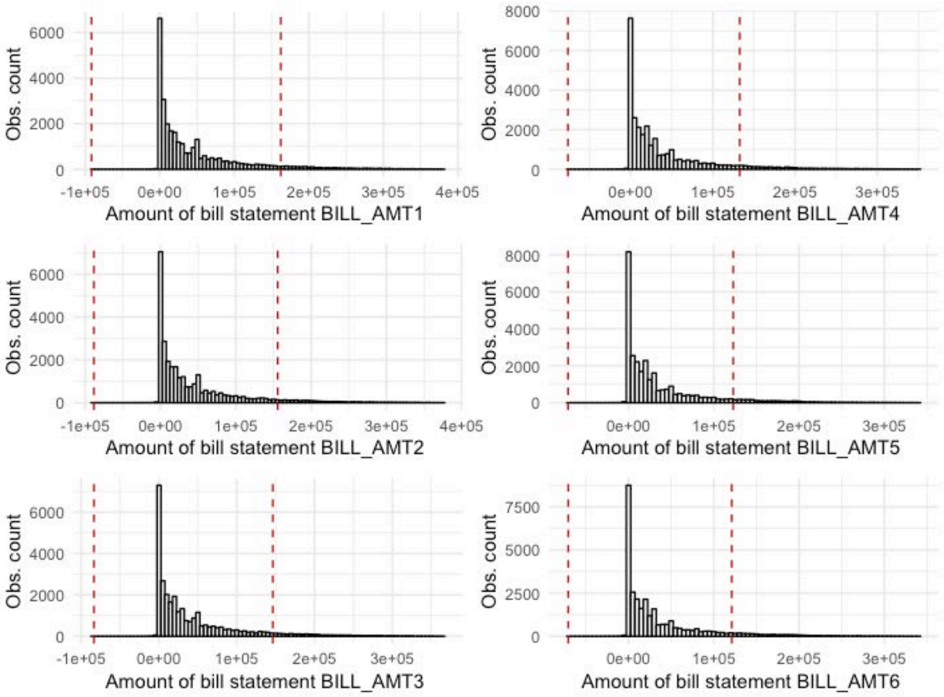


Table 8.2.4.1:
Correlation: BILL_AMTX

	DEFAULT
BILL_AMT1	-0.019
BILL_AMT2	-0.014
BILL_AMT3	-0.014
BILL_AMT4	-0.010
BILL_AMT5	-0.006
BILL_AMT6	-0.005

BILL_AMTX variables show high cross correlation. The recommendation is variable that captures the best relationship with response variable. Options considered were: BILL_SUM, Avg_Bill_Amt, Max_Bill_Amt & Max_Bill_Amt^2.

Correlation of these variable transformations suggested that 'Max_Bill_Amt' is the highest predictive indicator of default, so that is the recommended variable for consideration.

Table 8.2.4.2: WOE Analysis of Max_Bill_Amt Binning

Final Bin	**Total Count**	**Total Distr.**	**1.Rate**	**WOE**	**IV**
< = 600	1,507	5.0%	31.7%	-49.2	0.014
< = 4079	2,994	10.0%	26.5%	-24.0	0.006
< = 18400.65	5,999	20.0%	21.5%	3.4	0.000
< = 21034	1,502	5.0%	26.9%	-25.9	0.004
< = 52496.15	7,498	25.0%	22.4%	-1.4	0.000
< = Inf	10,500	35.0%	19.0%	19.4	0.012
Total	**30,000**	**100.0%**	**22.1%**	**NA**	**0.036**

8.2.5 PAY_AMTX – Distribution & WOE Binning Analysis

Figure 8.2.5.1 Histogram of PAY_AMT1 to PAY_AMT6

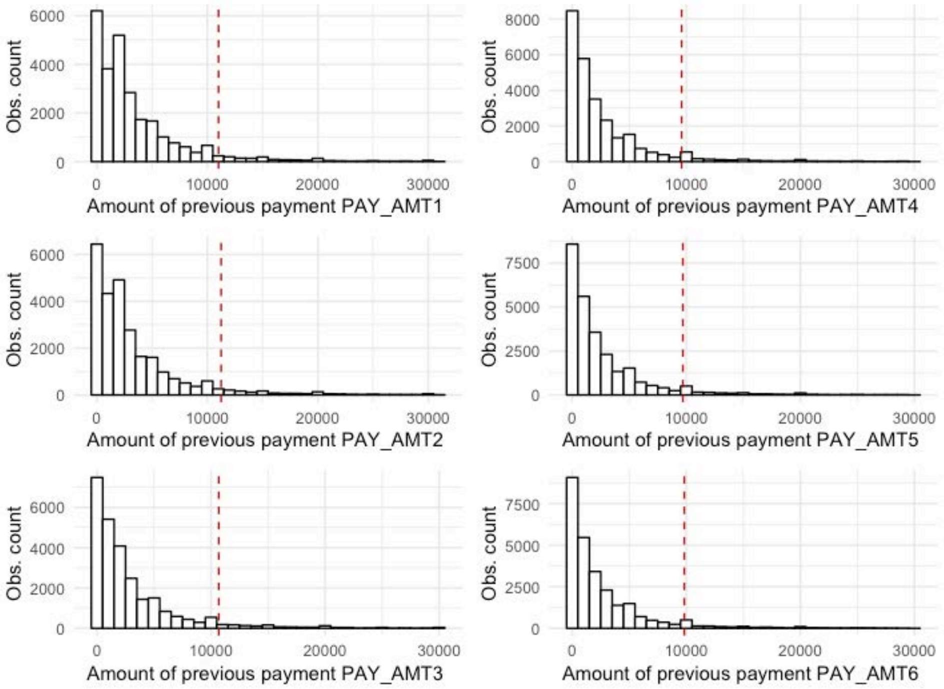


Table 8.2.5.1:
Correlation: PAY_AMTX

	DEFAULT
PAY_AMT1	-0.072
PAY_AMT2	-0.058
PAY_AMT3	-0.056
PAY_AMT4	-0.056
PAY_AMT5	-0.055
PAY_AMT6	-0.053

PAY_AMTX variables show high cross correlation. The recommendation is variable that captures the best relationship with response variable. Options considered were: Pmt_SUM, Avg_Pmt_Amt, Max_Pmt_Amt & Avg_PAY_Ratio.

Correlation of these variable transformations wilth response variable suggest that ‘Avg_Pmt_Amt’ is the highest predictive indicator, so that is the recommended variable.

Table 8.2.5.2: WOE Analysis of Avg_Pmt_Amt Binning

Final Bin	**Total Count**	**Total Distr.**	**1.Rate**	**WOE**	**IV**
< = 2045	13,500	45.0%	29.3%	-37.7	0.071
< = 12000	13,500	45.0%	17.7%	27.6	0.032
< = Inf	3,000	10.0%	9.6%	98.0	0.071
Total	**30,000**	**100.0%**	**22.1%**	**NA**	**0.173**

8.3 Transformed Data – Exploration, Distributions & Recommended Binning

8.3.1 Avg_Util – Creation, Distribution & WOE Binning Analysis

Definition: average amount of balance limit used <- (sum(BILL_AMTX/LIMIT_BAL)/6)

Figure 8.3.1.1
Histogram of Avg_Util

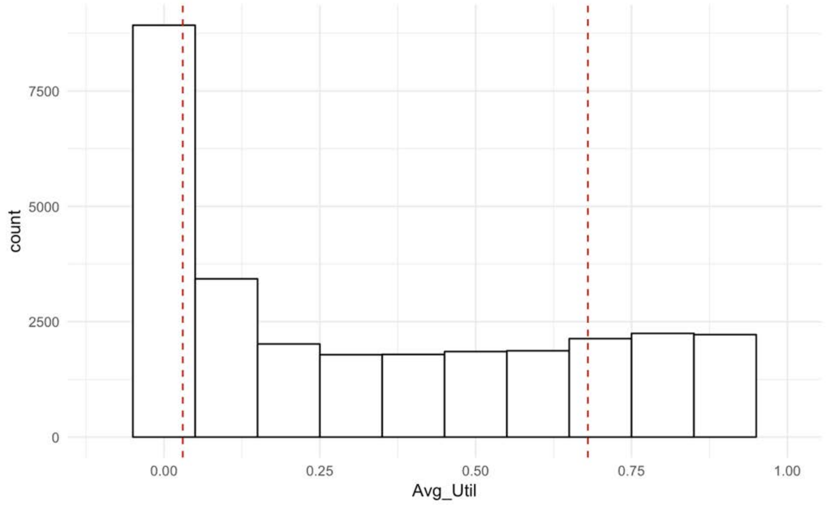


Table 8.3.1.1: WOE Analysis of Avg_Util Binning

Final Bin	Total Count	Total Distr.	1.Rate	WOE	IV
< = 0.0010	1,500	5.0%	30.0%	-41.1	0.009
< = 0.4517	16,500	55.0%	17.1%	31.8	0.051
< = Inf	12,000	40.0%	28.0%	-31.4	0.043
Total	30,000	100.0%	22.1%	NA	0.103

8.3.2 Avg_Pay_Ratio – Creation, Distribution & WOE Binning Analysis

Definition: average amount of Pmt against Bill <- (sum(BILL_AMTX+1/PAY_AmtX)/5)

Figure 8.3.1.1
Histogram of
Avg_Pay_Ratio

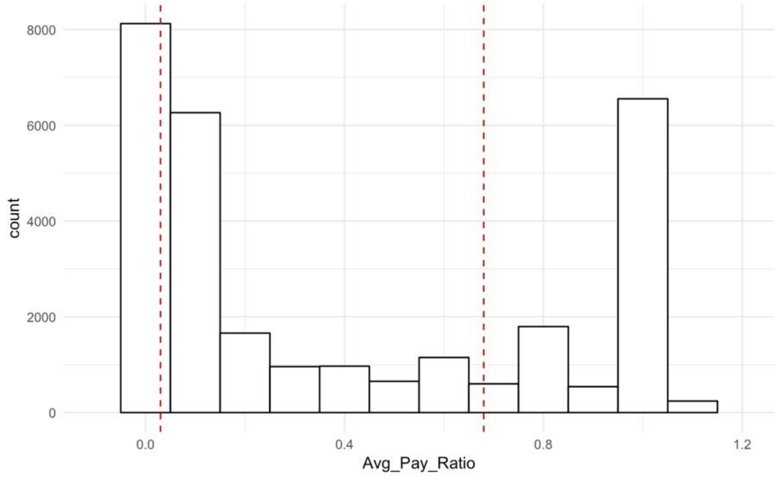


Table 8.3.2.1: WOE Analysis of Avg_Pay_Ratio Binning

Final Bin	Total Count	Total Distr.	1.Rate	WOE	IV
< = 0.0352	1,500	5.0%	46.6%	-112.2	0.080
< = 0.1134	12,000	40.0%	25.3%	-17.6	0.013
< = 1	13,395	44.6%	19.0%	18.9	0.015
< = Inf	3,105	10.3%	11.3%	80.1	0.052
Total	**30,000**	**100.0%**	**22.1%**	**NA**	**0.160**

8.3.3 Max_DLQ – Creation, Frequency & WOE Binning Analysis

Definition: max value in PAY_X variables across six months <- pmax (PAY_1 : PAY_6)

Table 9.3.3.1:
Summary of Max_DLQ

EDUCATION	Freq	PCT
0	19,931	66.4%
1	1,689	5.6%
2	7187	24%
3	789	2.6%
4	218	0.7%
5	69	0.2%
6	25	0.1%
7	67	0.2%
8	25	0.1%

Less than 30 percent of all clients have a MAX_DLQ above 1, but they represent almost half of all cases that default. This is demonstrated in the Weight of Evidence (WOE) analysis below. It turns out that this will be one of the highest predictive indicators of default in the model stage.

Table 8.3.3.2: WOE Analysis of Max_DLQ Binning

Final Bin	Total Count	Total Distr.	1.Rate	WOE	IV
< = 1	21,620	72.1%	12.7%	66.5	0.261
< = Inf	8,380	27.9%	46.3%	-111.0	0.435
Total	30,000	100.0%	22.1%	NA	0.696

8.3.4 Balance_Growth_6mo – Creation, Distribution & WOE Binning Analysis

Definition: Δ in balance <- ((LIMIT_BAL – BILL_AMT6) – (LIMIT_BAL – BILL_AMT1))

Figure 8.3.4.1 Histogram of Balance_Growth_6mo

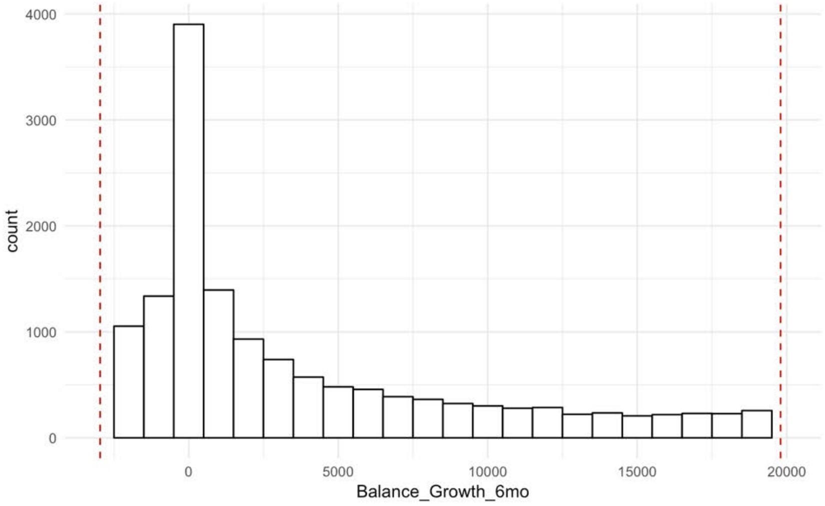


Table 8.3.4.1: WOE Analysis of Balance_Growth_6mo

Final Bin	Total Count	Total Distr.	1.Rate	WOE	IV
< = -21881.5	1,500	5.0%	12.7%	66.6	0.018
< = -10172.8	1,500	5.0%	19.7%	14.9	0.001
< = 923	12,002	40.0%	29.0%	-36.3	0.058
< = Inf	14,998	50.0%	17.8%	27.2	0.034
Total	**30,000**	**100.0%**	**22.1%**	**NA**	**0.111**

8.3.5 Util_Growth_6mo – Creation, Distribution & WOE Binning Analysis

Definition: Δ in utilization <- ((BILL_AMT1/LIMIT_BAL) – (BILL_AMT6/LIMIT_BAL))

Figure 8.3.5.1 Histogram of Util_Growth_6mo

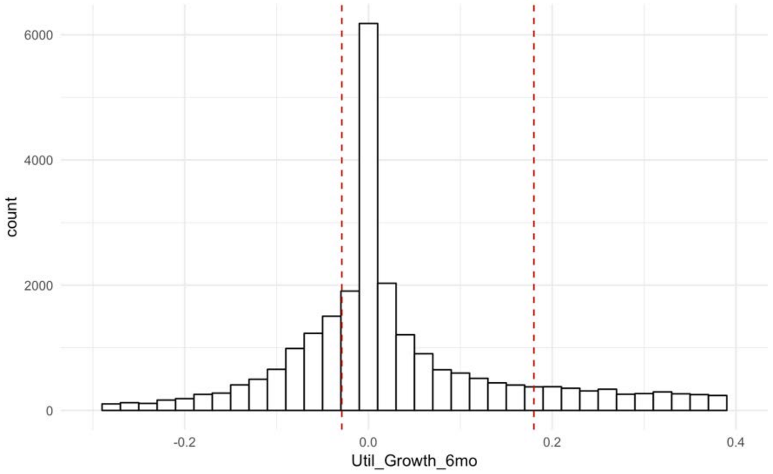


Table 8.3.5.1: WOE Analysis of Util_Growth_6mo

Final Bin	Total Count	Total Distr.	1.Rate	WOE	IV
< = -0.02909	7,500	25.0%	29.2%	-37.4	0.038
< = -0.00301	3,001	10.0%	19.3%	17.0	0.003
< = 0	2,726	9.1%	28.5%	-34.1	0.012
< = Inf	16,773	55.9%	18.4%	23.0	0.028
Total	**30,000**	**100.0%**	**22.1%**	**NA**	**0.081**

8.4 Recommended Variable Transformation Detail

Recommendations for transformation are according to the two data sets to be developed for modeling. In the case of tree models, continuous variables are left as is unless there is discrepancies to fix (e.g. 'EDUCATION > 'Other'). However, for regression models, the recommendation is to bin the data. Bins are according to the weight of evidence (WOE) analysis done in Section 2. For each binned variable, one bin is left out of the transformation to minimize cross correlation.

Table 8.4 Recommended Data Sets for Model Development

Data1 (Random Forest & Gradient Boosting)	**Data2** (Logistic Regression & Principal Components Analysis)	
LIMIT_BAL	LIMIT_BAL_below_30k	Avg_Pmt_Amt_below2k
SEX_Female	LIMIT_BAL_above_160k	Avg_Pmt_Amt_above12k
EDUCATION	SEX_Female	Avg_Util_below_.001
MARRIAGE_Y	MARRIAGE_Y	Avg_Util_above_.45
AGE	ED_Grad_Other	PAY_X_Sum_6mo_belowZero
PAY_X_Sum_6mo	AGE_Below_25	PAY_X_Sum_6mo_aboveSix
Avg_Pmt Amt	AGE_25to35	Balance_Growth_6mo < -21k
Avg_Util	AGE_above45	Balance_Growth_6mo < -10k
Avg_Pay_Ratio	Avg_Pay_Ratio_below_.035	Balance_Growtth_6mo > 1k
Balance_Growth_6mo	Avg_Pay_Ratio_above_.113	Util_Growth_6mo < -.03
Util_Growth_6mo	Avg_Pay_Ratio_above_1	Util_Growth_6mo > 0
Max_DLQ	Max_DLQ_above1	
Target (DEFAULT)	Target (DEFAULT)	

8.4.1 LIMIT_BAL – Recommended Transformation

Data1: No recommended Changes

Data 2: split LIMIT_BAL into 2 groups: <= 30k & > 160k

```
raw.data$LIMIT_BAL_below_30k <- ifelse(raw.data$LIMIT_BAL <= 30000,1,0)
raw.data$LIMIT_BAL_above_160k <- ifelse(raw.data$LIMIT_BAL > 160000,1,0)
```

8.4.2 Demographic Variables: (SEX, MARRIAGE, EDUCATION & AGE)

SEX <- Data1 & Data 2: If SEX = Female (2) then 1

*variable transformation of demographic data**

```
#transform SEX variable to SEX_FEMALE (1=true)
raw.data$SEX_FEMALE <- ifelse(raw.data$SEX == 2,1,0)
```

MARRIAGE <- Data1 & Data 2: If MARRIAGE = 1 (TRUE) then 1, ifelse = 0

```
#transform MARRIED variable to Married_Y (1=true)
raw.data$Married_Y <- ifelse(raw.data$MARRIAGE == 1,1,0)
```

```
#transform EDUCATION variable so all above 3 are (0 = Other)
raw.data$EDUCATION[raw.data$EDUCATION > 3] <- 0
```

ED_Grad_Other

```
#transform EDUCATION as per optimal binning in Exploratory Data Analysis (EDA)
raw.data$ED_Grad_other <- ifelse((raw.data$EDUCATION < 1) |
                    (raw.data$EDUCATION > 3) |(raw.data$EDUCATION == 1),1,0)
```

AGE Bins <- Data1 (no change) **& Data 2**: bin as per WOE Analysis (EDA)

```
#transform AGE as per optimal binning in Exploratory Data Analysis (EDA)
raw.data$AGE_below_25 <- ifelse(raw.data$AGE <= 25,1,0)
raw.data$AGE_25to35 <- ifelse((raw.data$AGE > 25) &
                    (raw.data$AGE <=35),1,0)
raw.data$AGE_above_40 <- ifelse(raw.data$AGE > 40,1,0)
```

8.4.3 PAY_X, BILL_AMTX & PAY_AMTX variable reduction

PAY_X <- Data1: Sum of PAY_X variables across six months/**Data2**: Binned as per WOE

PAY_X_Sum_6mo

```
#create sum variable of PAY_1 : PAY_6 variables
raw.data$PAY_X_Sum_6mo <- rowSums(cbind(raw.data$PAY_1,raw.data$PAY_2,
                    raw.data$PAY_3,raw.data$PAY_4,
                    raw.data$PAY_5,raw.data$PAY_6))
```

PAY_X_Sum_6mo_Bins

```
# bin the PAY_X_Sum_6mo as per optimal binning in Exploratory Data Analysis (EDA)
raw.data$PAY_X_Sum_6mo_belowZero <- ifelse(raw.data$PAY_X_Sum_6mo <= 0,1,0)
raw.data$PAY_X_Sum_6mo_aboveFive <- ifelse(raw.data$PAY_X_Sum_6mo > 5,1,0)
```

BILL_AMTX <- Data1: Sum of BILL_AMTX variables/**Data2**: Binned as per WOE

Max_Bill_Amt

```
#create variable of max value of BILL_AMT1 : BILL_AMT6
raw.data$Max_Bill_Amt <- pmax(raw.data$BILL_AMT1,raw.data$BILL_AMT2,
                        raw.data$BILL_AMT3,raw.data$BILL_AMT4,
                        raw.data$BILL_AMT5,raw.data$BILL_AMT6)
```

Max_Bill_Amt_Bins

```
# bin Max_Bill_Amt as per optimal binning in Exploratory Data Analysis (EDA)
raw.data$Max_Bill_Amt_below_600 <- ifelse(raw.data$Max_Bill_Amt <= 600,1,0)
raw.data$Max_Bill_Amt_below_4k <- ifelse(raw.data$Max_Bill_Amt > 600 &
                                    raw.data$Max_Bill_Amt <= 4000,1,0)
raw.data$Max_Bill_Amt_below_18k <- ifelse(raw.data$Max_Bill_Amt > 4000 &
                                    raw.data$Max_Bill_Amt <=18400,1,0)
raw.data$Max_Bill_Amt_below_21k <- ifelse(raw.data$Max_Bill_Amt > 18400 &
                                    raw.data$Max_Bill_Amt <=21000,1,0)
raw.data$Max_Bill_Amt_above_52k <- ifelse(raw.data$Max_Bill_Amt > 52000,1,0)
```

Avg_Pmt_Amt <- Data1: Avg of PAY_AMTX variables/**Data2**: Binned as per WOE

Avg_Pmt_Amt

```
#create variable of sum value of PAY_AMT1 : PAY_AMT6
raw.data$PMT_SUM <- rowSums(cbind(raw.data$PAY_AMT1,raw.data$PAY_AMT2,
                          raw.data$PAY_AMT3,raw.data$PAY_AMT4,
                          raw.data$PAY_AMT5,raw.data$PAY_AMT6))

#create variable of average of PAY_AMT1 : PAY_AMT6
raw.data$Avg_Pmt_Amt <- raw.data$PMT_SUM/6
```

Avg_Pmt_Amt_Bins

```
# bin Avg_Pmt_Amt as per optimal binning in Exploratory Data Analysis (EDA)
raw.data$Avg_Pmt_Amt_below2k <- ifelse(raw.data$Avg_Pmt_Amt <= 2045,1,0)
raw.data$Avg_Pmt_Amt_above12k <- ifelse(raw.data$Avg_Pmt_Amt > 12000,1,0)
```

8.4.4 Variable Creation: Avg_Pay_Ratio, Avg_Util & Max_DLQ

Avg_Pay_Ratio - variable creation (Pay_Ratio = BILL_AMTX/PAY_AMTX-1)

```
## NOTE: PAY_AMT1 is lagging payment on BILL_AMT2 (only 5 measures) ##
raw.data$Pay_Ratio_1 <- ifelse(raw.data$BILL_AMT2 > 0,
                       (raw.data$PAY_AMT1 / raw.data$BILL_AMT2),1)
raw.data$Pay_Ratio_2 <- ifelse(raw.data$BILL_AMT3 > 0,
                       (raw.data$PAY_AMT2 / raw.data$BILL_AMT3),1)
raw.data$Pay_Ratio_3 <- ifelse(raw.data$BILL_AMT4 > 0,
                       (raw.data$PAY_AMT3 / raw.data$BILL_AMT4),1)
raw.data$Pay_Ratio_4 <- ifelse(raw.data$BILL_AMT5 > 0,
                       (raw.data$PAY_AMT4 / raw.data$BILL_AMT5),1)
raw.data$Pay_Ratio_5 <- ifelse(raw.data$BILL_AMT6 > 0,
                       (raw.data$PAY_AMT5 / raw.data$BILL_AMT6),1)

raw.data$Ratio_SUM = rowSums(cbind(raw.data$Pay_Ratio_1,raw.data$Pay_Ratio_2,
                      raw.data$Pay_Ratio_3,raw.data$Pay_Ratio_4,
                      raw.data$Pay_Ratio_5))

raw.data$Avg_Pay_Ratio <- raw.data$Ratio_SUM/5
```

Avg_Pay_Ratio_Bins

```
## split Avg_Pay_Ratio as per optimal binning in Exploratory Data Analysis (EDA)
raw.data$Avg_Pay_Ratio_below_.035 <- ifelse(raw.data$Avg_Pay_Ratio <= .035,1,0)
raw.data$Avg_Pay_Ratio_above_.113 <- ifelse(raw.data$Avg_Pay_Ratio > .035 &
                                     raw.data$Avg_Pay_Ratio <= .113,1,0)
raw.data$Avg_Pay_Ratio_above_1 <- ifelse(raw.data$Avg_Pay_Ratio > 1,1,0)
```

AVG_Util - variable creation (Utilization = BILL_AMT/LIMIT_BAL)

```
#find utilization rate of each billing cycle (Utilization = BILL_AMT/LIMIT_BAL)
raw.data$Util_Bill_1 <- raw.data$BILL_AMT1 / raw.data$LIMIT_BAL
raw.data$Util_Bill_2 <- raw.data$BILL_AMT2 / raw.data$LIMIT_BAL
raw.data$Util_Bill_3 <- raw.data$BILL_AMT3 / raw.data$LIMIT_BAL
raw.data$Util_Bill_4 <- raw.data$BILL_AMT4 / raw.data$LIMIT_BAL
raw.data$Util_Bill_5 <- raw.data$BILL_AMT5 / raw.data$LIMIT_BAL
raw.data$Util_Bill_6 <- raw.data$BILL_AMT6 / raw.data$LIMIT_BAL

#create variable of sum values of utilization rates Util_Bill_1 : Util_Bill_6
raw.data$Util_SUM = rowSums(cbind(raw.data$Util_Bill_1,raw.data$Util_Bill_2,
                    raw.data$Util_Bill_3,raw.data$Util_Bill_4,
                    raw.data$Util_Bill_5,raw.data$Util_Bill_6))

#take the average Utilization rate from Util_Bill_1 : Util_Bill_6
raw.data$Avg_Util <- raw.data$Util_SUM/6
```

AVG_Util_Bins

```
## split Avg_Util as per optimal binning in Exploratory Data Analysis (EDA)
raw.data$Avg_Util_below_.001 <- ifelse(raw.data$Avg_Util <= .001,1,0)
raw.data$Avg_Util_above_.45 <- ifelse(raw.data$Avg_Util > .45,1,0)
```

Max_DLQ - variable creation (max value of PAY_1 : PAY_6)

```
#find max value of variables PAY_1 : PAY_6
raw.data$Max_DLQa <- pmax(raw.data$PAY_1,raw.data$PAY_2,raw.data$PAY_3,
                 raw.data$PAY_4,raw.data$PAY_5,raw.data$PAY_6)
```

Max_DLQ_Bins

```
#if Max_DLQa is below zero, set to zero, else max value of Max_DLQa
raw.data$Max_DLQ <- ifelse(raw.data$Max_DLQa <= 0,0,raw.data$Max_DLQa)

## split Max_DLQ as per optimal binning in Exploratory Data Analysis (EDA)
raw.data$Max_DLQ_above1 <- ifelse(raw.data$Max_DLQ > 1,1,0)
```

8.4.5 Variable Creation: Balance_Growth_6mo & Util_Growth_6mo

Balance_Growth_6mo - (Δ in difference from LIMIT_BAL to BILL_AMT over time)

Balance_Growth_6mo - variable creation

```
raw.data$Balance_Growth_6mo <- (raw.data$LIMIT_BAL-raw.data$BILL_AMT6)-
                 (raw.data$LIMIT_BAL-raw.data$BILL_AMT1)
```

Balance_Growth_6mo_Bins

```
## split Balance_Growth_6mo as per optimal binning in Exploratory Data Analysis (EDA)
raw.data$Balance_Growth_6mo_below_minus21k <- ifelse(
                raw.data$Balance_Growth_6mo <= -21800,1,0)
raw.data$Balance_Growth_6mo_below_minus10k <- ifelse(
                raw.data$Balance_Growth_6mo > -21800
                & raw.data$Balance_Growth_6mo <= -10000,1,0)
raw.data$Balance_Growth_6mo_above_1k <- ifelse(
                raw.data$Balance_Growth_6mo >= 1000,1,0)
```

Util_Growth_6mo - (Δ in utilization Util_Bill_1 - Util_Bill_6)

Util_Growth_6mo – Variable Creation

```
raw.data$Util_Growth_6mo <- raw.data$Util_Bill_1 - raw.data$Util_Bill_6
```

Util_Growth_6mo_Bins

```
# split Util_Growth_6mo as per optimal binning in Exploratory Data Analysis (EDA)
raw.data$Util_Growth_6mo_below_minus.03 <- ifelse(
                 raw.data$Util_Growth_6mo <= -.03,1,0)
raw.data$Util_Growth_6mo_below_minus.003 <- ifelse(
                 raw.data$Util_Growth_6mo > -.03 &
                 raw.data$Util_Growth_6mo <= -.003,1,0)
raw.data$Util_Growth_6mo_above_0 <- ifelse(
                 raw.data$Util_Growth_6mo > 0,1,0)
```

target <- transform DEFAULT variable to a factor for classification models

```
raw.data$target <- as.factor(raw.data$DEFAULT
```

8.5 Model Development – R code for Train/Test

Below is the R code used for model development as outlined in section 5 of this project in order of model development: Random Forest (Model 1), Gradient Boosting (Model 2), Logistic Regression (Model 4), & PCANNet (Model 4)

8.5.1 Random Forest (Model 1)

```
# create train/test/validation data set
sub_list_RFc <- c("LIMIT_BAL","SEX_FEMALE","EDUCATION",
          "Married_Y","AGE","PAY_X_Sum_6mo","Avg_Pmt_Amt",
          "Balance_Growth_6mo","Max_Bill_Amt","Max_DLQ","target")

xtrain_RF2 <- subset(raw.data, select = sub_list_RFc, data.group == 1)
xtest_RF2 <- subset(raw.data, select = sub_list_RFc, data.group == 2)
validate_RF2 <- subset(raw.data, select = sub_list_RFc, data.group == 3)

# tune model using variation of mtry
control <- trainControl(method="repeatedcv", number=10,
                         repeats=2, search="random")
set.seed(7)
metric = "Accuracy"
mtry <- 5
rf_random <- train(target ~., data=xtrain_RF2, method="rf",
            metric=metric, tuneLength=10,trControl=control)
print(rf_random)
plot(rf_random)

# run train model using recommended mtry (tuneLength) ‘2’
set.seed(7)
fit.rf2 <- train(target ~ .,data = xtrain_RF2,method = "rf",
                 tuneLength=2,trControl=control)

# predict probability of default = 1 on the train data set
# Predicting probability of survival using predict type 'prob'
predRF2_prob <- predict(fit.rf2, newdata = xtrain_RF2, type = "prob")

# create column in the train data set with probability
xtrain_RF2$predRF2_prob <- abs(as.numeric(predRF2_prob$'1'))

# create binary ‘class’ value based on threshold values
xtrain_RF2$classes <- ifelse(xtrain_RF2$predRF2_prob >.3,1,0)
```

```
# create confusion matrix
t_RF2_train = table(xtrain_RF2$target,xtrain_RF2$classes)
r_RF2_train <- apply(t_RF2_train,MARGIN=1,FUN=sum);

# normalize confusion matrix to rates
matrix_RF2_train <- t_RF2_train/r_RF2_train
matrix_RF2_train

# check accuracy

accuracy.RF2_train <- (t_RF2_train[1,1]+t_RF2_train[2,2])/(t_RF2_train[1,1]+
                        t_RF2_train[1,2]+t_RF2_train[2,1]+t_RF2_train[2,2])

cat('RF2_train accuracy:',accuracy.RF2_train)

# plot roc curve & print area under curve
rf.roc2 <-roc(xtrain_RF2$target,xtrain_RF2$classes)
plot(rf.roc2)
auc(rf.roc2)
```

8.5.2 Gradient Boosting (Model 2)

```
# create train/test/validation data set
sub_list_GBM <- c("LIMIT_BAL","SEX_FEMALE","EDUCATION",
              "Married_Y","AGE","PAY_X_Sum_6mo","Avg_Pmt_Amt",
              "Balance_Growth_6mo","Max_Bill_Amt","Max_DLQ","target")

xtrain_GBM <- subset(raw.data, select = sub_list_GBM, data.group == 1)
xtest_GBM <- subset(raw.data, select = sub_list_GBM, data.group == 2)
validate_GBM <- subset(raw.data, select = sub_list_GBM, data.group == 3)

# predict probability of default = 1 on the train data set
# Predicting probability of survival using predict type 'prob'
predGBM_prob <- predict(fit.GBM, newdata = xtrain_GBM, type = "prob")

# create column in the train data set with probability
xtrain_GBM$predGBM_prob <- abs(as.numeric(predGBM_prob$'1'))

# create binary ‘class’ value based on threshold values
xtrain_GBM$classes <- ifelse(xtrain_GBM$predGBM_prob >.3,1,0)

# create confusion matrix
t_GBM_train = table(xtrain_GBM$target,xtrain_GBM$classes)
r_GBM_train <- apply(t_GBM_train,MARGIN=1,FUN=sum);

# normalize confusion matrix to rates
```

```
matrix_GBM_train <- t_GBM_train/r_GBM_train
matrix_GBM_train
# check accuracy
accuracy.GBM_train <- (t_GBM_train[1,1]+t_GBM_train[2,2])/(t_GBM_train[1,1]+
                    t_GBM_train[1,2]+t_GBM_train[2,1]+t_GBM_train[2,2])

cat('GBM_train accuracy:',accuracy.GBM_train)
# plot roc curve & print area under curve
GBM.roc2 <-roc(xtrain_GBM$target,xtrain_GBM$classes)
plot(GBM.roc2)
auc(GBM.roc2)
```

8.5.3 Logistic Regression (Model 3)

The first step in the development of the linear models is to consider relative variable importance for final selection. Below is the R code used to develop the model.

```
# Initial data set for consideration of linear model and variable importance
sub_list_GLM <-c(
          "LIMIT_BAL_below_30k","LIMIT_BAL_above_160k",
          "SEX_FEMALE","ED_Grad_other","Married_Y","AGE_below_25",
          "AGE_25to35","AGE_above_40","PAY_X_Sum_6mo_belowZero",
          "PAY_X_Sum_6mo_aboveFive","Max_Bill_Amt_below_600",
          "Max_Bill_Amt_below_4k","Max_Bill_Amt_below_18k",
          "Max_Bill_Amt_below_21k","Max_Bill_Amt_above_52k",
          "Avg_Pmt_Amt_below2k","Avg_Pmt_Amt_above12k",
          "Avg_Util_below_.001","Avg_Util_above_.45",
          "Avg_Pay_Ratio_below_.035","Avg_Pay_Ratio_above_.113",
          "Avg_Pay_Ratio_above_1","Max_DLQ_above1",
          "Balance_Growth_6mo_below_minus21k",
          "Balance_Growth_6mo_below_minus10k",
          "Balance_Growth_6mo_above_1k","Util_Growth_6mo_below_minus.03",
          "Util_Growth_6mo_below_minus.003","Util_Growth_6mo_above_0","DEFAULT")

xtrain_GLM <- subset(raw.data, select = sub_list_GLM, data.group == 1)
xtest_GLM <- subset(raw.data, select = sub_list_GLM, data.group == 2)
validate_GLM <- subset(raw.data, select = sub_list_GLM, data.group == 3)

# Linear Model using all variables
full_glm <- glm(DEFAULT ~.,data=xtrain_GLM)
```

The first table is summary of the variable estimates and standard error prior to selection. The variables highlighted in yellow show a low significance to the target variable.

Table 8.5.3.1: Regression Summary using all potential variables

Dependent Variable	Estimate	Std. Error
Constant	0.187***	0.027
LIMIT_BAL_below_30k	0.028*	0.011
LIMIT_BAL_above_160k	-0.02**	0.007
SEX_FEMALE	-0.020**	0.006
ED_Grad_other	-0.003	0.006
Married_Y	0.023**	0.007
AGE_below_25	-0.008	0.012
AGE_25to35	-0.019*	0.009
AGE_above_40	-0.004	0.010
PAY_X_Sum_6mo_belowZero	-.051**	0.018
PAY_X_Sum_6mo_aboveFive	0.257***	0.130
Max_Bill_Amt_below_600	0.141***	0.028
Max_Bill_Amt_below_4k	0.078***	0.015
Max_Bill_Amt_below_18k	0.017	0.010
Max_Bill_Amt_below_21k	-0.006	0.016
Max_Bill_Amt_above_52k	0.018	0.009
Avg_Pmt_Amt_below2k	0.039***	0.008
Avg_Pmt_Amt_above12k	-0.02	0.012
Avg_Util_below_.001	-0.044	0.024
Avg_Util_above_.45	-0.046***	0.009
Avg_Pay_Ratio_below_.035	0.004	0.017
Avg_Pay_Ratio_above_.113	-0.03**	0.010
Avg_Pay_Ratio_above_1	-0.018	0.011
Max_DLQ_above1	0.17***	0.016
Balance_Growth_6mo_below_minus21k	-0.025	0.018
Balance_Growth_6mo_below_minus10k	-0.005	0.016
Balance_Growth_6mo_above_1k	0.005	0.014
Util_Growth_6mo_below_minus.03	-0.015	0.017
Util_Growth_6mo_below_minus.003	-0.029	0.017
Util_Growth_6mo_above_0	-0.034*	0.017
Akaike Inf. Crit.	13,561	
Note:	$^{*}p<0.1$; $^{**}p<0.05$; $^{***}p<0.01$	

*highlighted variables indicate low statistical significance and recommendation to remove following stepwiseAIC analysis

```
# use backward AIC process to remove variables with low significance
backward_glm <- stepAIC(full_glm,direction="backward",trace=FALSE)
backward_glm$anova
```

Table 8.5.3.2: Stepwise AIC analysis – variables to remove

Dependent Variable	Estimate	Std. Error
ED_Grad_other	-0.003	0.006
AGE_below_25	-0.008	0.012
AGE_above_40	-0.004	0.010
Max_Bill_Amt_below_21k	-0.006	0.016
Avg_Pay_Ratio_below_.035	0.004	0.017
Balance_Growth_6mo_below_minus10k	-0.005	0.016
Balance_Growth_6mo_above_1k	0.005	0.014
Util_Growth_6mo_below_minus.003	-0.029	0.017

```
# use updated train data set and model method ‘glm’ to fit model
control <- trainControl(method="repeatedcv", number=10, repeats=2, search="random")

fit.GLM2 <- train(target ~ .,data = xtrain_GLM2,family="binomial",
method = "glm",trControl=control)

# create train/test/validation data set
sub_list_GLM2 <- c("LIMIT_BAL","SEX_FEMALE","EDUCATION",
            "Married_Y","AGE","PAY_X_Sum_6mo","Avg_Pmt_Amt",
            "Balance_Growth_6mo","Max_Bill_Amt","Max_DLQ","target")

xtrain_GLM2 <- subset(raw.data, select = sub_list_GLM2, data.group == 1)
xtest_GLM2 <- subset(raw.data, select = sub_list_GLM2, data.group == 2)
validate_GLM2 <- subset(raw.data, select = sub_list_GLM2, data.group == 3)

# predict probability of default = 1 on the train data set
# Predicting probability of survival using predict type 'prob'
predGLM2_prob <- predict(fit.GLM2, newdata = xtrain_GLM2, type = "prob")

# create column in the train data set with probability
xtrain_GLM2$predGLM2_prob <- abs(as.numeric(predGLM2_prob$'1'))

# create binary ‘class’ value based on threshold values
xtrain_GLM2$classes <- ifelse(xtrain_GLM2$predGLM2_prob >.3,1,0)

# create confusion matrix
t_GLM2_train = table(xtrain_GLM2$target,xtrain_GLM2$classes)
r_GLM2_train <- apply(t_GLM2_train,MARGIN=1,FUN=sum);
```

```
# normalize confusion matrix to rates
matrix_GLM2_train <- t_GLM2_train/r_GLM2_train
matrix_GLM2_train
# check accuracy
accuracy.GLM2_train <- (t_GLM2_train[1,1]+t_GLM2_train[2,2])/(t_GLM2_train[1,1]+
                    t_GLM2_train[1,2]+t_GLM2_train[2,1]+t_GLM2_train[2,2])

cat('GLM2_train accuracy:',accuracy.GLM2_train)
# plot roc curve & print area under curve
GLM2.roc2 <-roc(xtrain_GLM2$target,xtrain_GLM2$classes)
plot(GLM2.roc2)
auc(GLM2.roc2)
```

8.5.4 PCANNet (Model 4)

The first step in the development of the PCA model is to consider further reduction of the variables. Taking into consideration the importance mapping of variables in section 5.3, as well as cross correlation in Exploratory Data Analysis (EDA).

Table 8.5.4.1:
Further variables to remove for PCA

Avg_Util_below_.001
Avg_Util_above_.45
Avg_Pay_Ratio_above_.113
Avg_Pay_Ratio_above_1
Util_Growth_6mo_below_minus.03
Util_Growth_6mo_above_0

Below is the R code used to develop the model.

```
# Initial data set for consideration of linear model and variable importance
sub_list_PCA <- c("LIMIT_BAL_below_30k","LIMIT_BAL_above_160k","SEX_FEMALE",
"Married_Y","AGE_25to35","PAY_X_Sum_6mo_belowZero","PAY_X_Sum_6mo_aboveFive",
         "Max_Bill_Amt_below_600","Max_Bill_Amt_below_4k","Max_Bill_Amt_below_18k",
          "Max_Bill_Amt_above_52k","Avg_Pmt_Amt_below2k","Avg_Pmt_Amt_above12k",
          "Max_DLQ_above1","Balance_Growth_6mo_below_minus21k","target")

xtrain_PCA <- subset(raw.data, select = sub_list_PCA, data.group == 1)
xtest_PCA <- subset(raw.data, select = sub_list_PCA, data.group == 2)
validate_PCA <- subset(raw.data, select = sub_list_PCA, data.group == 3)
```

```
# train the PCANNet model using TrainControl from previous models
control <- trainControl(method="repeatedcv", number=10, repeats=2, search="random")
set.seed(7)
fit.PCA <- train(target ~ .,data = xtrain_PCA,family="binomial",method = "pcaNNet",
                              trControl=control,verbose=FALSE)


# Predicting probability of survival using predict type ‘prob’
predPCA_train <- predict(fit.PCA, newdata = xtrain_PCA, type = "prob")

#create column with likelihood factor
xtrain_PCA$predPCA_train <- abs(as.numeric(predPCA_train$'1'))
summary(xtrain_PCA$predPCA_train)
#create binary classifier based on threshold values
xtrain_PCA$classes <- ifelse(xtrain_PCA$predPCA_train >.25,1,0)

# Checking classification accuracy
PCA_train = table(xtrain_PCA$target,xtrain_PCA$classes)
PCA_train

# Checking classification accuracy
t_PCA_train = table(xtrain_PCA$target,xtrain_PCA$classes)
t_PCA_train

accuracy.PCA_train <- (t_PCA_train[1,1]+t_PCA_train[2,2])/(t_PCA_train[1,1]+
                  t_PCA_train[1,2]+t_PCA_train[2,1]+t_PCA_train[2,2])

# Compute row totals;
r_PCA_train <- apply(t_PCA_train,MARGIN=1,FUN=sum);
# Normalize confusion matrix to rates;
matrix_PCA_train <- t_PCA_train/r_PCA_train
matrix_PCA_train

cat('PCA_train accuracy:',accuracy.PCA_train)
```

Supporting R Code & documentation can be found in my GitHub Repository:
https://github.com/talentrics/MSDS_Capstone_Project

Thank you for your interest in this project! If you have feedback, please leave me a note on any social media feed .Daniel Macdonald @talentrics