# CREDIT RISK PREDICTION USING CALIBRATION METHOD: AN APPLICATION IN FINANCIAL SCORECARD

LEE CHOON YI

MASTER OF SCIENCE

LEE KONG CHIAN FACULTY OF ENGINEERING AND SCIENCE
UNIVERSITI TUNKU ABDUL RAHMAN
DECEMBER 2021

# CREDIT RISK PREDICTION USING CALIBRATION METHOD: AN APPLICATION IN FINANCIAL SCORECARD

By

**LEE CHOON YI**

A thesis submitted to the Department of Mathematical and Actuarial Sciences,
Lee Kong Chian Faculty of Engineering and Science,
Universiti Tunku Abdul Rahman,
in partial fulfillment of the requirements for the degree of
Master of Science
December 2021

# ABSTRACT

## CREDIT RISK PREDICTION USING CALIBRATION METHOD: AN APPLICATION IN FINANCIAL SCORECARD

**Lee Choon Yi**

Machine Learning models have been extensively researched in the area of credit scoring. Banks have put in substantial resources into improving the credit risk model performance as improvement in accuracy by a fraction could translate into significant future savings. Given the lack of interpretability in machine learning models, it is often not used for capital provisioning in banks. This paper uses the Taiwan Credit Card dataset and illustrates the use of machine learning techniques to improve assessment of credit worthiness using credit scoring models. In factor transformation for a credit scorecard construction, Decision Tree technique showed the ability to produce quick and predictive transformation rule. Besides, model comparison result showed that Artificial Neural Network and Gradient Boosting Approach have great predictive power compared to traditional logistic regression scorecard. Credit underwriting decision could be improved by implementing a better discriminatory power scorecard, as more good customers are likely to be better than score cut-off and thus accepted by banks. Probability of Default (PD) Calibration maps model scores to output PD that reflects portfolio underlying performance. This paper illustrates approach to perform PD calibration for machine learning models that can be used to align with banks internal application scorecard strategy.

Calibration Plot and Binomial Test assessment showed that traditional scorecard approach performed better with least risk of underestimation of actual PD. Both tests suggested the use of traditional scorecard approach for capital estimation purpose.

# ACKNOWLEDGEMENTS

The author would like to thank Moshe Lichman for publicly providing the Taiwan Credit Card dataset at the UCI Machine Learning Repository. The author would also like to thank Kaggle and its community for description of the dataset as well as interesting topics discussed on the dataset.

## DECLARATION

I hereby declare that the dissertation is based on my original work except for quotations and citations which have been duly acknowledged. I also declare that it has not been previously or concurrently submitted for any other degree at UTAR or other institutions.

Name ______Lee Choon Yi___________

Date ______1 December 2021__ _______

**FACULTY OF ENGINEERING AND SCIENCE**

**UNIVERSITI TUNKU ABDUL RAHMAN**

Date: ____1 December 2021________

**SUBMISSION OF FINAL YEAR PROJECT /DISSERTATION/THESIS**

It is hereby certified that ______***Lee Choon Yi***__________________________ (ID No: __***1508089***___ ) has completed this final year thesis entitled "*Credit risk prediction using calibration method: an application in financial scorecard*" under the supervision of Dr Koh Siew Khew (Supervisor) from the Department of Mathematical and Actuarial Sciences, Lee Kong Chian Faculty of Engineering and Science, and Dr Pan Wei Yeing (Co-Supervisor) from the Department of Mathematical and Actuarial Sciences, Lee Kong Chian Faculty of Engineering and Science.

I understand that University will upload softcopy of my final year thesis in pdf format into UTAR Institutional Repository, which may be made accessible to UTAR community and public.

Yours truly,

_Lee Choon Yi_______
(*Student Name*)

*Delete whichever not applicable

## APPROVAL SHEET

This dissertation/thesis entitled "**CREDIT RISK PREDICTION USING CALIBRATION METHOD: AN APPLICATION IN FINANCIAL SCORECARD**" was prepared by LEE CHOON YI and submitted as partial fulfillment of the requirements for the degree of Master of Science at Universiti Tunku Abdul Rahman.

Approved by:

Koh

(Dr Koh Siew Khew) Date: 2 December 2021

Assistant Professor/Supervisor
Department of Mathematical and Actuarial Sciences
Lee Kong Chian Faculty of Engineering and Sciences
Universiti Tunku Abdul Rahman

(Dr Pan Wei Yeing) Date: 5 December 2021

Assistant Professor/Co-supervisor
Department of Mathematical and Actuarial Sciences
Lee Kong Chian Faculty of Engineering and Sciences
Universiti Tunku Abdul Rahman

## LIST OF TABLES

## LIST OF FIGURES

# LIST OF ABBREVIATIONS

| Abbreviation | Description |
|---|---|
| AR | Accuracy Ratio |
| BCBS | Basel Committee on Banking Supervision |
| B-score | Behavioral scorecard |
| BU | Business Unit |
| CC | Credit Card |
| CI | Confidence Interval |
| DPD | Days past due |
| EAD | Exposure at Default |
| EL | Expected Loss |
| GINI | Gini Coefficient |
| IV | Information Value |
| KS | Kolmogorov-Smirnov |
| LGD | Loss Given Default |
| MFA | Multi-factor Analysis |
| MIA | Month in Arrears |
| MOB | Months on Book |
| NA | Not Applicable |
| PD | Probability of Default |
| PiT | Point-in-time |
| SFA | Single Factor Analysis |
| TTC | Through-the-cycle |
| WoE | Weight of Evidence |

## TABLE OF CONTENTS

# CHAPTER 1

# INTRODUCTION

## *1.1 Background*

Credit lending has been one of the main driving forces behind the economies of most leading industrial countries. According to Thomas (2009), the founding of Bank of England in 1694 was one of the first signs of the financial revolution which would allow mass lending. Over the years, bank began lending to the noble, and slowly lending begun to be offered by manufacturer in the form of hire purchase, where they would sell machines to client in the form of credit lending. Unsecured lending started in 1920s and later, credit card facility was introduced in 1950s and 1960s, where consumer can enjoy the benefit of purchasing variety of products ranging from food to air flight tickets using this facility. Today, banks and financial institutions offer credit facilities range from Sovereign Bond, Corporate Lending, specific project financing, to consumer credit such as mortgages, hire purchase, personal loan, credit cards, overdrafts and many other financial products. While the demand for credit is worldwide and continuing to grow at extreme high rate, it is important to put in place a sophisticated Credit Risk Management Framework that manages credit risk and allows lending decision to be made under a systematic and automated way.

Today, the world has entered into dawn of the fourth industrial evolution, which differs in speed, scale, complexity, and transformative power compared to previous revolutions. Xu et al. (2018) examined the opportunities and challenges that are likely to arise as a result of the revolution. Rapid development of Machine Learning tools in the recent year have solved challenges in many areas, including the banking and finance industry. Gan et al. (2020) attempted to predict Asian option prices using deep learning model, and showed that the speed of the trained deep learning model is extremely fast, with high accuracy. Other than finance industry, Wang et al. (2020) used machine learning methods to forecast binary New Product Development (NPD) strategy for Chinese automotive industry, which is crucial for decision making to ensure the scarce resources are allocated effectively.

Credit losses refers to loss that arise in the event credit borrowers failed to fulfil obligation in repayment. It is crucial for banking supervisors to ensure credit lenders maintain enough capital with adequate loss absorption capacity. In June 2004, the Basel Committee on Banking Supervision issued a revised framework on International Convergence of Capital Measurement and Capital Standards. The standard will allow "internal ratings-based" (IRB) banking institutions with the use of internal measures for key drivers of credit risk as primary inputs to their minimum regulatory capital calculation (Basel Committee on Banking Supervision, 2004). A Research Task Force formed a subgroup in 2002, to review and develop research on the validation of credit rating systems that would be useful to banks and supervisors as they consider options for implementing Basel II. Basel working Paper 14 (Basel Committee on Banking Supervision, 2005) outlined validation guidelines for the 3 key risk components (PD: Probability of Default, LGD: Loss Given Default, EAD: Exposure at Default). Banks can internally develop their own credit risk models for calculating expected loss. Expected Loss $= PD \times LGD \times EAD$. Explicit requirements in the revised Framework underline the need for Banks to validate internal rating systems, and must demonstrate to their regulatory supervisor that they can assess the performance of their internal ratings and their risk estimation systems consistently and meaningfully.

Machine Learning in Banking Risk Management has gained significant amount of attention from academia and industry. Tang et al. (2019) studied the application of random forest algorithm to assess credit risk of the energy industry in China and found that the algorithm produces prediction with high accuracy, and more capable of dealing with multicollinearity issue. They also used the mean decrease accuracy and mean decrease GINI method to rank order the importance of all variables to provide insight into which variable is most predictive of the outcome. Digitalization of risk processes in bank and financial institutions have become increasingly important, for example, conduct risk. By combining machine learning and transaction data, financial institutions are able to automate conduct monitoring for mortgage underwriting (Oliver Wyman, 2017). Leo et al. (2019) and Abdou & Pointon (2011) had reviewed a number of available literature and evaluated machine-learning techniques that have been researched in the context of banking risk management, and identified problems in risk management that have been inadequately explored and are potential areas for further research. Many researches were focusing on the scoring accuracy of credit decision, since an improvement in accuracy of even a fraction of a percent translates into significant future savings (West, 2000). West (2000) investigated 5 different neural network architectures, which are multilayer-perceptron (MLP), mixture of experts (MOE), radial basis function (RBF), learning vector quantization (LVQ), and fuzzy adaptive resonance (FAR), against other credit scoring models on 2 real world datasets. He found that while the MLP architecture is the more commonly used neural network model, the MOE model is slightly more accurate than the other credit scoring models. He also demonstrated the explanatory ability of neural network models by using variable importance of the input variables. This is useful for explaining the denial of credit decision by looking at the most important variables of the model.

Machine Learning is viewed as the intersection of computer science, engineering and statistics. Awad & Khanna (2015) highlighted machine learning as a tool that can be applied to various problems, especially in fields that require data to be interpreted and acted upon. Machine learning tools are also driving advances in other areas such as search engines, and self-driving cars. Manufacturing sector are also increasingly adopting machine learning for potential opportunities for cost reduction, improved productivity, and improved risk management. To financial institutions, machine learning is capable of impacting every aspects of business model, not only risk management but also aspects such as fraud detection, cross-selling of products according to customer preferences, and etc.

### *1.2 Problem Statements*

Given the capabilities to impact business significantly, however, many banks have not completely adopted machine learning due to many techniques are falling short of providing an explanation for the analysed relationship. This created complexities around model development and evaluation, as some argued, they are more “black box” in nature, with results at times being difficult to interpret. Besides, it is also argued that machine learning models are sensitive to outliers, and prone to overfitting, and result in counterintuitive predictions.

Adopting machine learning has become a challenge faced by banks and financial institutions, especially with the validation of rating system requirements given by the Basel Accord, and ability to demonstrate consistency and accuracy of model over time, albeit the fact that the machine learning algorithms have shown good uplift to accuracy of the model as compared to traditional scoring approach.

In this paper, a few problems will be studied. Firstly, can machine learning techniques be applied to improve efficiency of credit scorecard development process for banks, and result in a hybrid scoring approach? Secondly, can machine learning models be applied to improve credit underwriting process, that is, improve approval rate while preserving a good quality loan portfolio? Thirdly, what are the potential advantages and pitfalls when utilizing machine learning models in production?

### *1.3 Outline of Dissertation*

In this research, a few machine learning techniques that offer improvement in efficiency of traditional credit scoring framework will be presented, and the proposed methodology to produce PD estimate using calibration method will be introduced. Traditional credit scoring framework used in this paper refers to methodology outlined in Credit Risk Scorecards: Developing and Implementing Intelligent Credit Scoring (Siddiqi, 2005). Firstly, Decision Tree technique will be applied for feature selection and variable transformation process in credit scorecard development. Secondly, several machine learning models will be built to compare the discriminatory power of the models with the traditional credit scorecard model, which is done through Logistic Regression. There are many machine learning algorithms available and

6 common approaches are selected which have been greatly studied by other researchers, namely, Conditional Inference Tree, Gradient Boosting, Random Forest, Neural Network, Support Vector Machine and K-Nearest Neighbors. R language is chosen since extensive libraries are available, and leveraged on the great model tuning capability of the CARET package. Thirdly, a probability of default (PD) calibration method will be proposed for the usage in credit application scorecard. For the purpose of capital and provisioning, the internal scoring models from traditional approach is recommended to be maintained, as it offers great interpretability, allows for effective scorecard monitoring strategy and ease of recalibration to ensure no underestimation in capital required. Fourthly, Calibration Plot test and Binomial test on the PD are performed to assess the quality of calibration for usage in application scorecard. Adjustments to the PD calibration will also be proposed to produce Long Run Average PD, for the estimation to reflect PD through the economic cycle condition. Finally, various advantages that could be reaped from machine learning models when applied to credit risk management in banks will be discussed in concluding remarks.

### *1.4 Objectives*

This paper focus on objectives below:

- To study the effectiveness of machine learning technique in variable transformation, in terms of speed and accuracy as compared to traditional approaches.
- To compare scoring model performance between traditional scorecard approach and machine learning approach.

- To improve the banks existing application scorecard, in terms of accepting more good customers, have a “sharpened” cut-off strategy empowered by higher discriminatory power machine learning models, and calibrated to output PD that represents portfolio performance.
- To assess the quality of calibrated PD estimate output among various approaches in terms of accuracy and conservatism.

# CHAPTER 2

# LITERATURE REVIEW

## *2.1 Credit Scoring*

Increased competition in the financial lending sector have led banks and financial institutions to search for more effective ways to attract creditworthy customers. Optimizing scorecard's approval decision to ensure credible customers are offered with credit facilities while keeping capital charge low has become more challenging to most banks and financial institutions. On July 24, 2014, the IASB published the complete version of IFRS9, have introduced new impairment model, which results in earlier recognition of losses (PricewaterhouseCoopers, 2014). The change in impairment model approach has resulted in significant impact to banks and financial institutions. Major difference between old and new impairment model (term as Expected Credit Loss model) is that, banks need to provide increased provisions (lifetime ECL) to customers who are delinquent in payment prior to default. Under the new standard, the banks with poorer book quality with more customers who are delinquent on payment, will be impacted the most with the increased in provision charge. Besides, the paper from PwC also highlighted that, besides payment delinquent as criteria, banks also need to provide increased provision (lifetime ECL) to financial assets which have "significant increase in credit risk". Besides providing adequate provision as part of the lending business, banks and financial institutions must also maintain their capital above a minimum capital adequacy ratio threshold as prescribed by regulator to ensure having enough liquidity, especially during adverse economic scenario. An explanation on the capital requirement and risk weight formula under the IRB framework (Basel

Committee on Banking Supervision, 2005) highlighted that while the Expected Losses should be covered in the provision as part of the cost of lending business, the banks must hold enough capital to account for Unexpected Losses of a loan. Excessive lending to risky customers would result in very high provision and capital charge, which would affect the profitability of the business. In general, banks should focus on improving the book quality through better credit underwriting as well as portfolio review process. Application scorecard refers to the credit scoring model that rates customer upon the credit facility application. Behavioural scorecard refers to the credit scoring model that tracks customer payment, delinquent behaviour in order to estimate PD as of reporting date. It is crucial for banks to have high performance Application and Behavioural scorecards that are able to accurately assess the risk of customers. Some banks and financial institutions have developed Collection Scorecard, by which customers who are more likely to repay the delinquent payment would be given priority to work on for collection staff, as they have limited resources to perform debt collection from delinquent and defaulted customer. Banks and regulatory supervisors thus focus on both predictive and also interpretability of credit scoring models to ensure there is adequate capital provisioning. Traditional credit scoring approach has offered great interpretability. Siddiqi (2005) highlighted that while there are various mathematical techniques available to build prediction scorecards, the most appropriate technique to be used can depend on various issues. For example, data quality, target variable type, sample size, implementation platforms, interpretability of results, and legal compliance on methodology as usually required to be transparent and explainable. Besides, the ability to track and diagnose scorecard performance is also key to selecting the most suitable technique. Siddiqi (2005) also outlined steps and methodology in traditional scorecard development, including exploring data, identifying missing values and outliers, correlation, initial characteristic analysis, multiple factor analysis, preliminary scorecard, reject inference, final scorecard production, scorecard scaling, and scorecard validation. The methodology for traditional scorecard approach presented in this paper is based on similar methodology, which is greatly practice by many banks and financial institutions for scorecard development. Sun and Wang (2005) highlighted that the validity of a rating model should be discriminative, homogeneous, and stable. They also proposed Kolmogorov-Smirnov Test (K-S test), Gini Coefficient and Receiver Operating Characteristic (ROC) as possible ways to validate credit rating model.

The financial crisis in year 2007 had caused substantial damage to global economic systems, and was due to credit fraud problem. To avoid credit fraud, social credit has become an increasingly important criterion for the evaluation of economic agent activity and guaranteeing the development of a market economy with minimal supervision costs (Yu et al., 2015). Yu et al. (2015) researched on a number of social credit literature to provide review in terms of theoretical foundation, scoring methods, and regulatory mechanism. In response to the credit crisis, Basel Committee had introduced a comprehensive set of reform measures to strengthen the regulation and supervision on banking sector. These measures are also known as “Basel III” Framework (Basel Committee on Banking Supervision, 2011).

Hand and Henley (1997) highlighted some other areas of credit scoring and credit control which also present interesting statistical challenge. Loan

Servicing and Review functions are important area that also involve credit scoring. For example, using customer behavioural score to determine credit limits, risk-based pricing, with interest rate charged according to estimated risk, fraudulent use of credit card scorecard, profitability scoring, collection scorecard on delinquent loan, and marketing scorecard for customers who require credit facilities.

### *2.2 Data mining*

Sharma (2009) presented a useful Guide to credit scoring in R that uses the German Credit dataset to demonstrate traditional credit scoring using logistic regression, and also cutting edge techniques available in R. Yap et al. (2011) compared traditional credit scorecard approach (similar to the approach taken in this paper), with Decision Tree and Logistic Regression approaches. They found that the final selected model is credit scorecard approach and that Decision Tree approach has shown lower misclassification rate in Training Dataset, but higher misclassification rate in Validation Dataset. Wang et al. (2012) proposed use of ensemble techniques bagging and random subspace to improve accuracy rate of Decision Tree model, by reducing the influence of the noise data and redundant attributes. Barboza et al. (2017) compared machine learning approaches against the traditional approaches which are Multivariate Discriminate Analysis (MDA) and Logistic Regression, and found that machine learning models show improved bankruptcy prediction accuracy over traditional models. Increasing number of available data mining techniques also attract significant interest from researchers to apply in credit scoring, as highlighted by Louzada et al. (2016) that growing number of credit scoring papers published is increasing by years. Besides, they also reviewed and compared the classification performance from various researches done on the Australian and German credit datasets.

Zhao et al. (2014) compared the Average Random Choosing method to Pure Random Choosing method when sampling Training, Validation, and Test set. They found that Average Random Choosing method has positive impact towards performance of the machine learning model. Based on their result, Pure Random Choosing method will not be applied in the research for optimal model performance. The Average Random Choosing method algorithm is similar to the stratified random sampling applied in this research (refer to modelling data sampling methodology). This algorithm will maintain a similar event rate across training, validation and test data set. They also tested multi-layer perceptron neural network algorithm, where they trained 34 models that tune across different number of neurons from 6 to 39, with 1 hidden layer. They found that the optimal number of neurons is 9. They also reported under severe class imbalance dataset, for example 99% vs 1% event rate, the Average Random Choosing method does not improve the performance of model, and approach such as oversampling is more preferred. They also highlighted MLP neural network computation time increases with the increase number of neurons, and such scenario is also observed in this research. Khashman (2011) applied an input normalization technique to transform all input variables into range between 0 and 1, by dividing them against the maximum value of each variable, before training neural network models. Kuhn & Johnson (2013) highlighted that to improve the effectiveness of neural networks model, various data transformation methods were evaluated. They found that the spatial sign transformation method on variable showed significant improvement on the performance of neural networks model. They also highlighted in classification model that the predicted class probability needs to be well-calibrated, and they suggested the use of calibration plot to assess the quality of the class probability. In their study, the quadratic discriminant analysis (QDA) was compared to a random forest model. The calibration plot showed that QDA class probabilities did not perform as well compared to the random forest model. Calibration plot shows sigmoidal pattern such that QDA model tend to underestimate the probability when the actual event probability is moderately high or low. They proposed that an additional model could be built to adjust for this pattern. Platt (2000) described methods of post-processing the prediction output of the Support Vector Machine model to estimate class probability, by using the logistic regression model equation. After post processing with the calibration equation, the result shows improved calibration with the same data. In this paper, the post-processing technique is studied for it's effectiveness to produce accurate PD estimate.

Öğüt et al. (2012) used Support Vector Machine and Artificial Neural Network to compare against traditional approaches (Multiple Discriminant Analysis and Ordered Logistic Regression) in predicting the financial strength rating of Turkish Banks. They found that both Multiple Discriminant Analysis and Support Vector Machine achieve the highest accuracy rate when pre-transformed variables are used as input variables. Whereas Ordered Logistic Regression performed the best when transformed factors scores are used as input variables. Desai et al. (1996) compared the performance of neural networks such as multilayer perceptron and modular neural networks, as well as some traditional techniques such as linear discriminant analysis and logistic regression. The finding reported that neural networks offers good improvement in percentage of bad loans correctly classified (sensitivity). However, on the measure of accuracy rate, logistic regression models are comparable to neural networks approach. Min and Lee (2008) used data envelopment analysis (DEA) to predict the bankruptcy of manufacturing firms. They used financial ratios of externally audited firms and found that the DEA based approach might be a good alternative as it requires only ex post information to calculate credit scores which have good accuracy rate. The results do not differ significantly from that obtained by Discriminant Analysis. Oreski et al. (2012) proposed a feature selection method by using combination of genetic algorithm with neural networks to improve accuracy rate of neural network classifier. They found that the approach is better than other techniques such as Forward selection, Information gain, Gain ratio, Gini index, and Correlation. Between the bank's internal behavioral scoring model and the external credit bureau scoring model, Chi and Hsu (2012) found that combining the two models is more predictive than by looking only at one of the model alone.

Yeh and Lien (2009) examined six major classification methods – Artificial Neural Networks (ANN), K-nearest neighbor, Logistic regression, Discriminant Analysis, Naïve Bayesian, and Classification trees. From their finding, ANN performs the best among all the other methods in terms of Accuracy Ratio. They also suggested the use of Accuracy Ratio to compare the model performance, instead of error rate. This is because in the credit card dataset used (similar dataset is used in this research), most records are non-risky, therefore the error rate is insensitive to classification accuracy of models. The

ANN is also compared to the other five classification methods for default probability produced, and it is reported that ANN performs the best in presenting real probability of default. Li et al. (2020) studied the use of machine learning techniques in the credit ratings prediction, and found that the prediction precision was at its maximum for the random forest algorithm. The precision remained consistent when predictions were done for different rating classes, from Investment Grade Ratings, to Speculative Grade Ratings and Default Ratings.

# CHAPTER 3

# PROPOSED METHODOLOGY

## *3.1 Overview*

This paper presents machine learning techniques that can be used to improve traditional scorecard construction, as well as compare machine learning model performance to traditional scorecard approach. Finally, the machine learning model scores were calibrated to produce an accurate PD estimate. In general, development of a robust scorecard is a multi-step process that involves not just statistical analysis but also expert judgement. The process is similar between traditional scorecard construction and machine learning models.

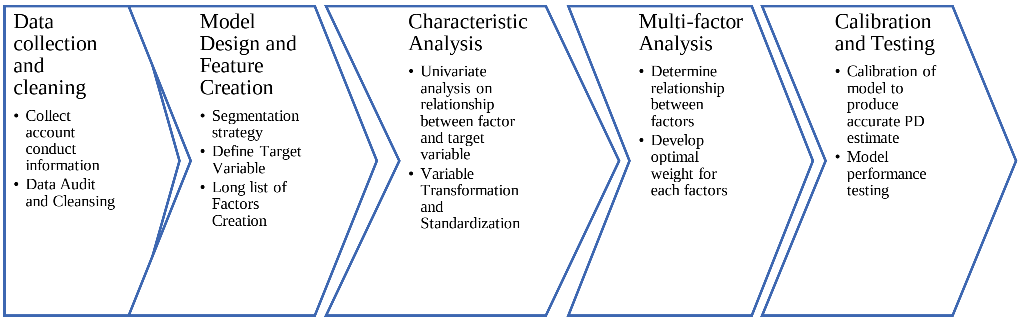
Data collection and cleaning
• Collect account conduct information
• Data Audit and Cleansing
Model Design and Feature Creation
• Segmentation strategy
• Define Target Variable
• Long list of Factors Creation
Characteristic Analysis
• Univariate analysis on relationship between factor and target variable
• Variable Transformation and Standardization
Multi-factor Analysis
• Determine relationship between factors
• Develop optimal weight for each factors
Calibration and Testing
• Calibration of model to produce accurate PD estimate
• Model performance testing

**Figure 1: Overview of Modelling Process**

## *3.2 Data Collection and Data Cleaning*

This research uses the Default of Credit Card Clients dataset. The dataset contains information on default payments, demographic factors, credit data, history of payment, and bill statements of credit card clients in Taiwan from April 2005 to September 2005. The dataset source from a public repository, namely UCI Machine Learning Repository. We have downloaded the dataset from Kaggle website in UCI Machine Learning Repository. A brief description of the dataset is tabulated below.

**Table 1: Description of dataset**

| No | Name | Data Type | Description |
|---|---|---|---|
| 1 | ID | Numeric | ID of each client |
| 2 | LIMIT_BAL | Numeric | Amount of given credit in NT dollars |
| 3 | SEX | Categorical | Gender |
| 4 | EDUCATION | Categorical | Education status |
| 5 | MARRIAGE | Categorical | Marital status |
| 6 | AGE | Numeric | Age in years |
| 7 | PAY_0 | Numeric | Repayment status in September, 2005 |
| 8 | PAY_2 | Numeric | Repayment status in August, 2005 |
| 9 | PAY_3 | Numeric | Repayment status in July, 2005 |
| 10 | PAY_4 | Numeric | Repayment status in June, 2005 |
| 11 | PAY_5 | Numeric | Repayment status in May, 2005 |
| 12 | PAY_6 | Numeric | Repayment status in April, 2005 |
| 13 | BILL_AMT1 | Numeric | Bill statement in September, 2005 |
| 14 | BILL_AMT2 | Numeric | Bill statement in August, 2005 |
| 15 | BILL_AMT3 | Numeric | Bill statement in July, 2005 |
| 16 | BILL_AMT4 | Numeric | Bill statement in June, 2005 |
| 17 | BILL_AMT5 | Numeric | Bill statement in May, 2005 |
| 18 | BILL_AMT6 | Numeric | Bill statement in April, 2005 |
| 19 | PAY_AMT1 | Numeric | Previous payment in September, 2005 |
| 20 | PAY_AMT2 | Numeric | Previous payment in August, 2005 |
| 21 | PAY_AMT3 | Numeric | Previous payment in July, 2005 |
| 22 | PAY_AMT4 | Numeric | Previous payment in June, 2005 |
| 23 | PAY_AMT5 | Numeric | Previous payment in May, 2005 |
| 24 | PAY_AMT6 | Numeric | Previous payment in April, 2005 |
| 25 | default | Numeric | Default payment in next month |

The dataset consists of 30,000 observations and 25 variables. The dataset is examined for potential issue such as missing values, outliers or any inconsistencies. No missing values is found in the dataset. However, two exclusions were applied:

**Table 2: Sample Exclusion**

| No | Exclusion | Count |
|---|---|---|
| | Initial Dataset | 30,000 |
| 1 | Repayment status in September, 2005 is delinquent, with PAY_0 > 0, but however Amount of bill statement in September, 2005 (NT dollar) is <= 0. As the repayment status (delinquent) is not consistent with the amount of bill statement. | 1,689 |
| 2 | Amount of bill statement in September, 2005 (NT dollar) <= 0, but however default payment next month. As the default event is not consistent with the amount owed. | 184 |
| | Final Modeling Sample | 28,127 |

As shown in the table above, the excluded 1,873 observations accounts for 6% of the population, the remaining 28,127 (94%) observations will be used to perform scoring models analysis.

### *3.3 Model Design*

In the process of credit scorecard construction, banks typically perform segmentation analysis to identify groups of homogeneous risk populations. The identified populations will be treated separately and applied different scorecard due to the unique relationships between customer's characteristics and target variable. The segmentation is generally driven by both expert input and statistical analysis. In this research, such analysis was not performed as the sample only has 28,127 count, and breaking the sample into smaller groups for scoring purpose might affect the performance of the scorecard due to over-granularity. Only a single scorecard will be constructed and used to compare against the machine learning models.

From the description of dataset above, our target variable is default payment next month, which can be modelled by inputs such as payment, usage, utilisation and some demographic factors. Typically, a credit scorecard should model the default within a performance period. Bad is defined as defaulted within the performance period. Defaulted in Basel Accord means delinquent for more than 90 days. The choice of performance period varies depending on the specific loan portfolio, where the chosen period should be long enough to cover enough observed Bads or risk of the portfolio, while not too long for the predictive information to be outdated and no longer relevant to the modelled bad. In this research, our target variable is delinquent in the next month, which might capture not only bad but also good customers who forget to make payment before due date. Besides, we do not have the performance of customers across different observation months to revise the definition according to performance period. Hence, we are only able to model the Bads using the field given in the dataset, that is, default payment next month. In other words, the performance window is only 1 month. As such, the model performance could be impacted, and we recommend revision to the definition if there is available data.

### *3.4 Modelling Data Sampling*

To ensure the constructed scorecard is predictive and could generalize well on new sample, a portion of the modelling sample should be separated out from model training. This approach is also used in machine learning model training to prevent overfitting of models. Modelling data sampling is the process of partitioning the modelling sample into training sample and holdout or testing sample. In this research, Stratified Random Sampling is performed to split modelling data into Train and Test sample. With Strata variable set as our target variable, we ensure the event rate across training and test dataset is similar. If the sample size is too small, it will be more appropriate to use the entire sample for model training and then perform bootstrap validation. Typically, banks would choose 70% vs 30% or 80% vs 20% as the splitting portion. In this research, 80% Train vs 20% Test sample is used. Oversampling is not required as there is 21% event rate in our modelling sample, with sufficient bad. During the development of machine learning models, the training dataset will be further split into Train and Validate sample to identify the most optimal tuning parameters. Repeated cross validation technique will be applied, where n = 10 folds and repeat = 5 times. In essence, each machine learning algorithm will be trained 50 times, to pick the best tuning parameters. In this technique, n (10) equal parts are obtained from training data initially. Then, n – 1 (9) parts are used in order to develop model and the remaining part (Validate sample) is used in order to test the model. This process is repeated until all parts of data are used as test data for the model developed by the remaining part of training data. Once all parts of data are tested, repeat the process 5 times using other random seed. Once all parts of data are tested, the tuning parameter with the highest average Gini is chosen as the optimal parameter and the performance of the machine learning model is validated with this best tuned parameter. The best tuned machine learning models will then be compared on the 20% Test sample.

### *3.5 Long List Factor Creation*

This process is also known as Feature Creation. In the process of constructing a robust credit scorecard, a list of factors is typically created to ensure the designed scorecard will fit well to the business strategy. The list of factors created can be driven by expert input or statistical analysis of the initial input field. In general, banks will maintain this list and further enhance the list when there is a change in business model or risk management strategy. For example, banks would like to explore business opportunity with potential customers coming from area that are underbanked. Existing list of risk factors might not be sufficient and new factors could be added through scoring model analysis. As the banks progress in more advanced credit risk management framework, the list become more complex and focus on specific future business needs. In this research, a list of 65 factors is created from the original 23 input fields using common industry approach for behavioral scoring model. The list of factors created is shown in Table 3.

**Table 3: Summary of the Long List of Factors Created**

| No | Name | Data Type | Description |
|---|---|---|---|
| 1 | UTIL1 | Numeric | Utilization Current Month (Sep 2005) |
| 2 | UTIL2 | Numeric | Utilization last 1 Month (Aug 2005) |
| 3 | UTIL3 | Numeric | Utilization last 2 Month (July 2005) |
| 4 | UTIL4 | Numeric | Utilization last 3 Month (June 2005) |
| 5 | UTIL5 | Numeric | Utilization last 4 Month (May 2005) |
| 6 | UTIL6 | Numeric | Utilization last 5 Month (April 2005) |
| 7 | AVG_UTIL_6m | Numeric | Average Utilization in last 6 month |
| 8 | AVG_UTIL_5m | Numeric | Average Utilization in last 5 month |
| 9 | AVG_UTIL_4m | Numeric | Average Utilization in last 4 month |
| 10 | AVG_UTIL_3m | Numeric | Average Utilization in last 3 month |
| 11 | AVG_UTIL_2m | Numeric | Average Utilization in last 2 month |
| 12 | MAX_UTIL_6m | Numeric | Maximum Utilization in last 6 month |
| 13 | MAX_UTIL_5m | Numeric | Maximum Utilization in last 5 month |
| 14 | MAX_UTIL_4m | Numeric | Maximum Utilization in last 4 month |
| 15 | MAX_UTIL_3m | Numeric | Maximum Utilization in last 3 month |

| No | Name | Data Type | Description |
|---|---|---|---|
| 16 | MAX_UTIL_2m | Numeric | Maximum Utilization in last 2 month |
| 17 | MAX_BY_AVG_UTIL_6m | Numeric | Maximum Utilization div Average Utilization in last 6 month |
| 18 | MAX_BY_AVG_UTIL_5m | Numeric | Maximum Utilization div Average Utilization in last 5 month |
| 19 | MAX_BY_AVG_UTIL_4m | Numeric | Maximum Utilization div Average Utilization in last 4 month |
| 20 | MAX_BY_AVG_UTIL_3m | Numeric | Maximum Utilization div Average Utilization in last 3 month |
| 21 | MAX_BY_AVG_UTIL_2m | Numeric | Maximum Utilization div Average Utilization in last 2 month |
| 22 | Curr_bill_perc_max_bill_6m | Numeric | Current bill statement as % of maximum bill statement in last 6 month |
| 23 | Curr_bill_perc_max_bill_5m | Numeric | Current bill statement as % of maximum bill statement in last 5 month |
| 24 | Curr_bill_perc_max_bill_4m | Numeric | Current bill statement as % of maximum bill statement in last 4 month |
| 25 | Curr_bill_perc_max_bill_3m | Numeric | Current bill statement as % of maximum bill statement in last 3 month |
| 26 | Curr_bill_perc_max_bill_2m | Numeric | Current bill statement as % of maximum bill statement in last 2 month |
| 27 | Worst_Status_L6M | Numeric | Worst Repayment Status in last 6 month |
| 28 | Worst_Status_L5M | Numeric | Worst Repayment Status in last 5 month |
| 29 | Worst_Status_L4M | Numeric | Worst Repayment Status in last 4 month |
| 30 | Worst_Status_L3M | Numeric | Worst Repayment Status in last 3 month |
| 31 | Worst_Status_L2M | Numeric | Worst Repayment Status in last 2 month |
| 32 | Count_Status_GT0_L6M | Numeric | Number of times Repayment Status Greater Than 0 in last 6 month |
| 33 | Count_Status_GT0_L5M | Numeric | Number of times Repayment Status Greater Than 0 in last 5 month |

| No | Name | Data Type | Description |
|---|---|---|---|
| 34 | Count_Status_GT0_L4M | Numeric | Number of times Repayment Status Greater Than 0 in last 4 month |
| 35 | Count_Status_GT0_L3M | Numeric | Number of times Repayment Status Greater Than 0 in last 3 month |
| 36 | Count_Status_GT0_L2M | Numeric | Number of times Repayment Status Greater Than 0 in last 2 month |
| 37 | Count_Status_GT1_L6M | Numeric | Number of times Repayment Status Greater Than 1 in last 6 month |
| 38 | Count_Status_GT1_L5M | Numeric | Number of times Repayment Status Greater Than 1 in last 5 month |
| 39 | Count_Status_GT1_L4M | Numeric | Number of times Repayment Status Greater Than 1 in last 4 month |
| 40 | Count_Status_GT1_L3M | Numeric | Number of times Repayment Status Greater Than 1 in last 3 month |
| 41 | Count_Status_GT1_L2M | Numeric | Number of times Repayment Status Greater Than 1 in last 2 month |
| 42 | Count_Status_GT2_L6M | Numeric | Number of times Repayment Status Greater Than 2 in last 6 month |
| 43 | Count_Status_GT2_L5M | Numeric | Number of times Repayment Status Greater Than 2 in last 5 month |
| 44 | Count_Status_GT2_L4M | Numeric | Number of times Repayment Status Greater Than 2 in last 4 month |
| 45 | Count_Status_GT2_L3M | Numeric | Number of times Repayment Status Greater Than 2 in last 3 month |
| 46 | Count_Status_GT2_L2M | Numeric | Number of times Repayment Status Greater Than 2 in last 2 month |
| 47 | Mths_since_status_GT0 | Numeric | Months Since last Repayment Status Greater Than 0 |
| 48 | Mths_since_status_GT1 | Numeric | Months Since last Repayment Status Greater Than 1 |
| 49 | Mths_since_status_GT2 | Numeric | Months Since last Repayment Status Greater Than 2 |
| 50 | avg_pmt_as_perc_bill_L5m | Numeric | Average Payment as a percent of bill statement in last 5 month |
| 51 | avg_pmt_as_perc_bill_L4m | Numeric | Average Payment as a percent of bill statement in last 4 month |

| No | Name | Data Type | Description |
|---|---|---|---|
| 52 | avg_pmt_as_perc_bill_L3m | Numeric | Average Payment as a percent of bill statement in last 3 month |
| 53 | avg_pmt_as_perc_bill_L2m | Numeric | Average Payment as a percent of bill statement in last 2 month |
| 54 | avg_pmt_as_perc_bill_L1m | Numeric | Average Payment as a percent of bill statement in last month |
| 55 | Cnt_Mth_With_pmt_L6M | Numeric | Number of times Payment > 0 in last 6 month |
| 56 | Cnt_Mth_With_pmt_L5M | Numeric | Number of times Payment > 0 in last 5 month |
| 57 | Cnt_Mth_With_pmt_L4M | Numeric | Number of times Payment > 0 in last 4 month |
| 58 | Cnt_Mth_With_pmt_L3M | Numeric | Number of times Payment > 0 in last 3 month |
| 59 | Cnt_Mth_With_pmt_L2M | Numeric | Number of times Payment > 0 in last 2 month |
| 60 | Cnt_Mth_With_pmt_L1M | Numeric | Number of times Payment > 0 in last month |
| 61 | Count_Pmt_GE_BAL_L5M | Numeric | Number of times Payment >= bill amount in last 5 month |
| 62 | Count_Pmt_GE_BAL_L4M | Numeric | Number of times Payment >= bill amount in last 4 month |
| 63 | Count_Pmt_GE_BAL_L3M | Numeric | Number of times Payment >= bill amount in last 3 month |
| 64 | Count_Pmt_GE_BAL_L2M | Numeric | Number of times Payment >= bill amount in last 2 month |
| 65 | Count_Pmt_GE_BAL_L1M | Numeric | Number of times Payment >= bill amount in last month |

From the table above, the list of factors can be categorized into broad categories such as payment, delinquent, utilisation information of the customers. The list covers few dimensions below:

1. Payment frequency, recency
2. Delinquent frequency, recency, severity
3. Utilisation frequency, recency

On top of the original 23 input variables given in the dataset, the above list of 65 variables result in a total 88 input variables for scoring analysis. Typically, banks have more customers data covering other dimensions. For example, transactional data such as customers purchase.

### *3.6 Characteristic Analysis*

Characteristic Analysis, also known as Single Factor Analysis, is a process by which a long list of potential factors is univariately analysed to arrive at a shorter list of candidate factors for inclusion in the credit scorecard. Sometimes it is also termed as Feature Selection process. The outcome of this process is mainly to identify and remove low discriminatory power, too concentrated or poor distributed, unstable and redundant variables. During the process, variable transformation, standardization, variable shortlist decision was made for further analysis on the variables.

To assess the predictive power of a variable, we will use the Accuracy Ratio (AR), also known as Gini coefficient as the primary test for discriminatory power, and Information Value as secondary test. AR is defined as the ratio of $a_R$ to $a_P$, that is:

$$AR = \frac{a_R}{a_P}$$

where

- $a_R$ is the area between the Cumulative Accuracy Profile (CAP) of the rating model being validated and the CAP of the random model; and
- $a_P$ is the area between the CAP of the perfect rating model and the CAP of the random model.

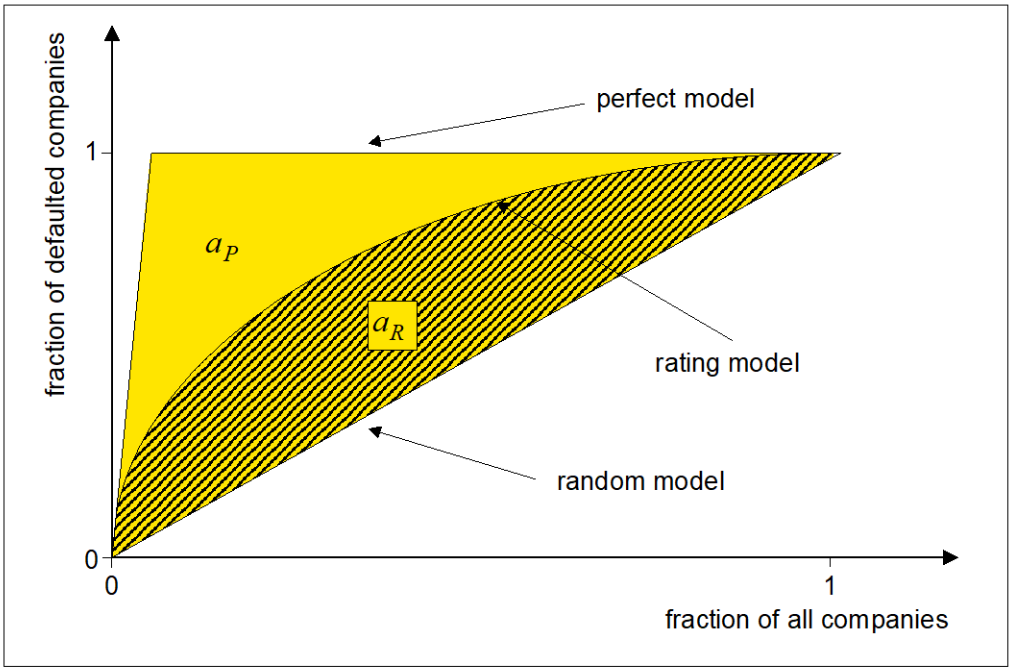

**Figure 2: Cumulative Accuracy Profiles and Accuracy Ratio**

The CAP is also known as the GINI curve, Power curve, or Lorenz curve. Gini coefficient ranges between 0 to 1, when it is equal to 1, it means the model output is fully able to differentiate non-defaulters and defaulters. When it is equal to 0, the rating model cannot discriminate between non-defaulters and defaulters. In reality, CAP curve of a rating model would run between the perfect curve and random model curve. Information Value (IV), or total strength of the characteristic comes from information theory, and is measured using the formula:

$$IV = \sum_{i=1}^{n} (Distr\ Good_i - \ Distr\ Bad_i) * \ln(\frac{Distr\ Good_i}{Distr\ Bad_i})$$

where,

- $Distr\ Good_i$ is the Distribution of Good observation in group *i*

$$Distr\ Good_i = \frac{Count\ Good_i}{Total\ Count\ of\ Good}$$

- $Distr\ Bad_i$ is the Distribution of Bad observation in group *i*

$$Distr\ Bad_i = \frac{Count\ Bad_i}{Total\ Count\ of\ Bad}$$

High IV indicates high predictive power, and vice versa. In this research, Gini >= 10% was used as shortlisting criteria. IV was only for reference but not used as shortlisting criteria. 69 variables were shortlisted for multi-factor analysis.

Variable transformation was performed to improve performance of the scorecard. This approach is also common in credit scorecard construction. For machine learning models, both the transformed and raw variables are analyzed for the multi-factor analysis and it is observed that there is no material difference in the performance of the model. In practice, there are various methodology in performing variable transformation. In this paper, Weight of Evidence (WOE) transformation was applied as per the methodology outlined in Siddiqi (2005). The transformation involves Binning or grouping of identical risk subpopulation (guideline refer to **Equation 3: Good/Bad Index equation**), and assign the WOE measure as the score for the subpopulation. WOE is based on the log of odds calculation:

**Equation 1: Weight of Evidence**

$$WoE = ln\left(\frac{\%\ Distribution\ of\ Goods}{\%\ Distribution\ of\ Bads}\right)$$

where,

- % Distribution of Goods represents percentage of good customers in a particular group; and
- % Distribution of Bads represents percentage of bad customers in a particular group.

The WOE measures the strength of each attribute class in discriminating good and bad accounts. It is a measure of the difference between the proportion of good and bad accounts in each attribute class. Positive number implies that the particular attribute class is isolating a higher proportion of good than bad, and vice versa. A higher WOE value implies lower risk while lower WOE value implies higher risk in that attribute class. WOE transformation offers advantages below:

- Handle outliers and missing values without imputation.
- Grouping allows modeler to understand relationships between the factor and the target variable, it allows modeler to explain the nature of this relationship in addition to the strength and predictive power of the factor.
- Allows comparison of the strength of the continuous and categorical variables without creating dummy variables.
- Allows control over the development process, by shaping the groups, one shapes the final composition of the scorecard. A factor can be grouped to align with existing decision strategy, such as loan-to-value ratio and debt-service-ratio to match with the banks policy limit.

Variable standardization was applied in this stage. All scores were normalized to mean, $\mu = 0$ and standard deviation, $\sigma = 1$. An average customer would receive a WOE score close to 0, and negative value implies higher than average risk, and vice versa. Modelling variables of similar scale could improve performance of machine learning models. For logistic regression approach, it would allow for meaningful comparison of the regression coefficient across all variables.

**Equation 2: Scores normalization equation**

$$Standardized\ Score = \frac{(score - \mu)}{\sigma}$$

Binning, or sometimes refers to classing, involves a combination of statistical analysis and expert input to arrive at final binning. The common guideline for interval and categorical variables is shown below:

For interval variable,

- Factors are first being fine-classed into 20 bands, where each band consists of approximately 5% of the total population.
- To combine groups with similar bad rate or based on business intuition. One common approach is to combine group within 20 Good/Bad index, where Good/Bad index is defined as below:

**Equation 3: Good/Bad Index equation**

$$Good/Bad\ Index = \frac{Number\ of\ Goods}{Number\ of\ Bads}$$

- Ensure there is sufficient observation (>= 5% of population).
- Fine classed result may not produce monotonic risk trend across bands, thus further combine bands to produce monotonic (increasing/decreasing) risk trend.

For nominal variables,

- Start by combining attribute with small sample size into group “Others”. Generally, there is no sample size number that is used to define “small”, however minimum number of bads (for example, > 30 or 1% bads) in each attribute is more commonly used to ensure there is meaningful result in score assignment.
- For the remaining attribute, group similar bad rate attribute.

- Further group until it has met guideline similar to those set upon interval variables.

The common approach above has been applied and shows good performance on the transformed variable, but it requires a lot of time due to the need to combine groups that may or may not result in monotonic risk trend. Further, some of the characteristic, for example delinquent, is by nature do not comprise 5% of total population in each attribute, but having significantly different risk than no delinquent. Hence the 5% threshold should be relaxed, and ensure that there is sufficient bad count. The common approach that bins according to 5% of the population might result in the substantially higher risk group included into the lower risk group. The minor tweak and fine tuning on this grouping could be performed by first splitting the higher risk population into sub-group before fine classing the rest, this could be time consuming considering there are many variables to be analysed in real world banking dataset. Besides, for nominal variable, it might be time consuming if there are too many attributes.

In this research, the effectiveness of Decision Tree approach in automating the above process will be explored. Splitting observation using Decision Tree could produce binning that is optimal as the predictive power of the factor is maximized through the training process. Conceptually, the decision tree binning should provide best predictive power of the factor, while benefit from all the grouping advantages highlighted above. However, time should still be spent on further grouping the decision tree binning result to meet requirement such as business operations consideration, monotonicity of the risk trend. For non-monotonic risk trend binning, the result could be applied, provided that it is aligned with the business expectation. The idea of improving the modelling process in this research, is to test whether the algorithm is effective and able to reduce time needed to achieve binning that has good predictive power and suit other business requirements. Especially for nominal variable, decision tree binning result will produce grouping that combine similar risk attribute together, and less time will be spent for further grouping. Conditional inference tree, ctree algorithm was used to perform binning on variables. At times, decision tree might not produce a monotonically risk ranking result, and we might want to consider alternative scheme, monotonic binning scheme works as follows:

- Factors are first being fine-classed into $n$ = 20 bands, where each band consists of approximately 5% of the total population.
- Examine the monotonicity of the risk across bands, if there is a break, repeat step 1 with n = $n$ – 1 = 19.
- Break if there are only 2 bands left.

To demonstrate the proposed methodology on characteristic analysis process, the first variable, credit limit, LIMIT_BAL is chosen with the result given as follows. Firstly, generate high level overview of the relationship between variable and the dependent variable.

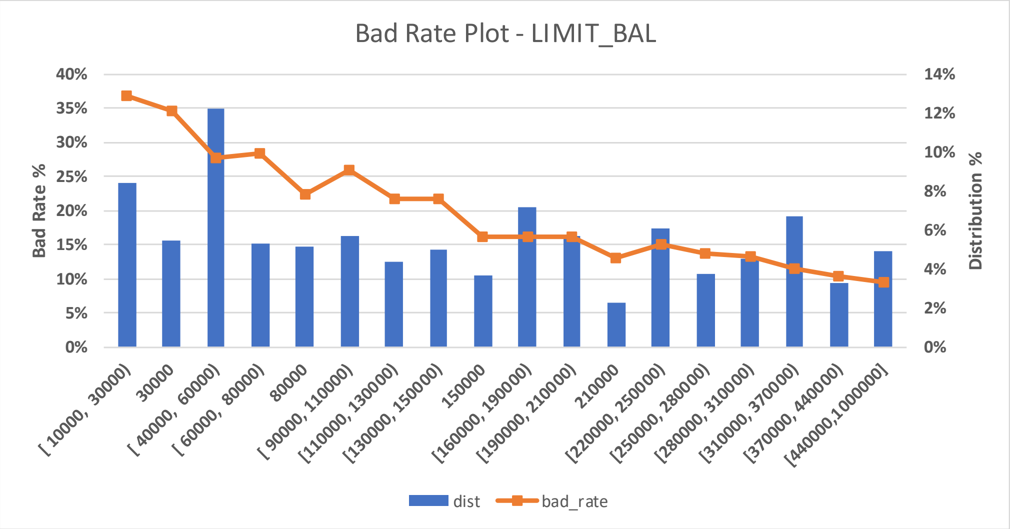

**Figure 3: Factor Bad Rate Plot**

Generally, the bad rate decreases as the credit limit increases, indicating that customers who are having bigger limit have lower risk than the average customers. Similarly, a box plot is made to confirm the observation.

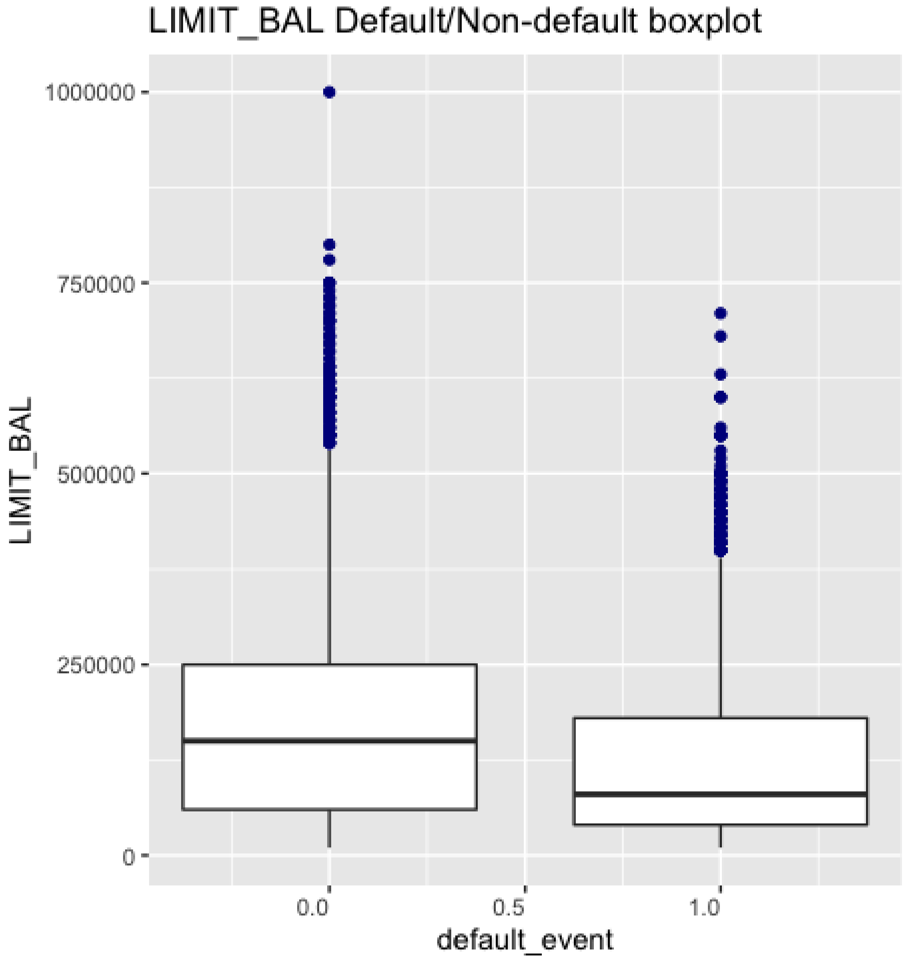

**Figure 4: Factor Box Plot**

Figure 4 shows that the defaulter population generally have lower credit limit than the average customers. We also observed the distribution of credit limit is skewed and some customers have substantially larger credit limit, the maximum observed value is 1 million NTD. Next, we generate the proposed binning schemes.

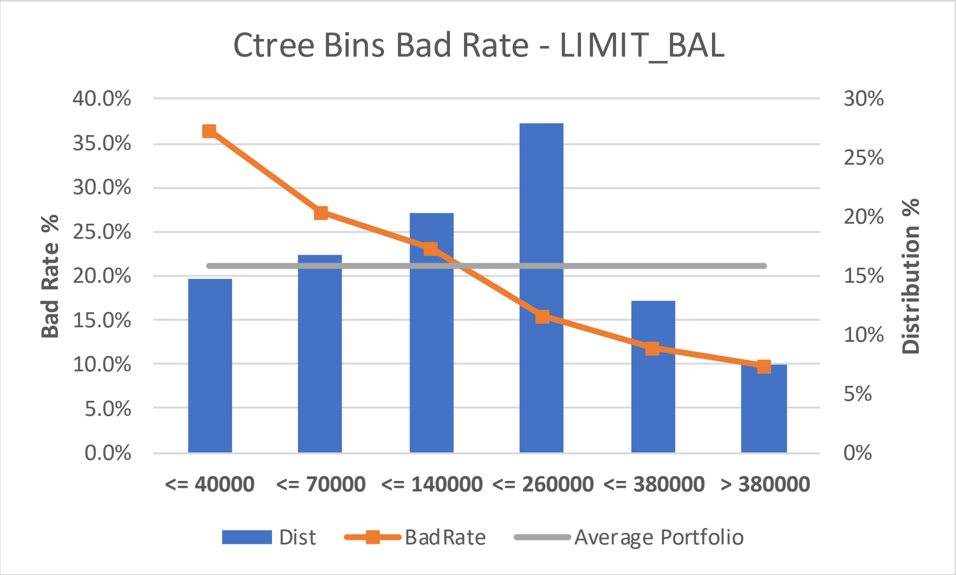

**Figure 5: Conditional Inference Tree Binning Scheme Bad Rate Plot**

The above ctree binning implies that customers who have credit limit <= NTD 140,000 should receive WOE scores that are < 0 since it is higher risk compared to average population.

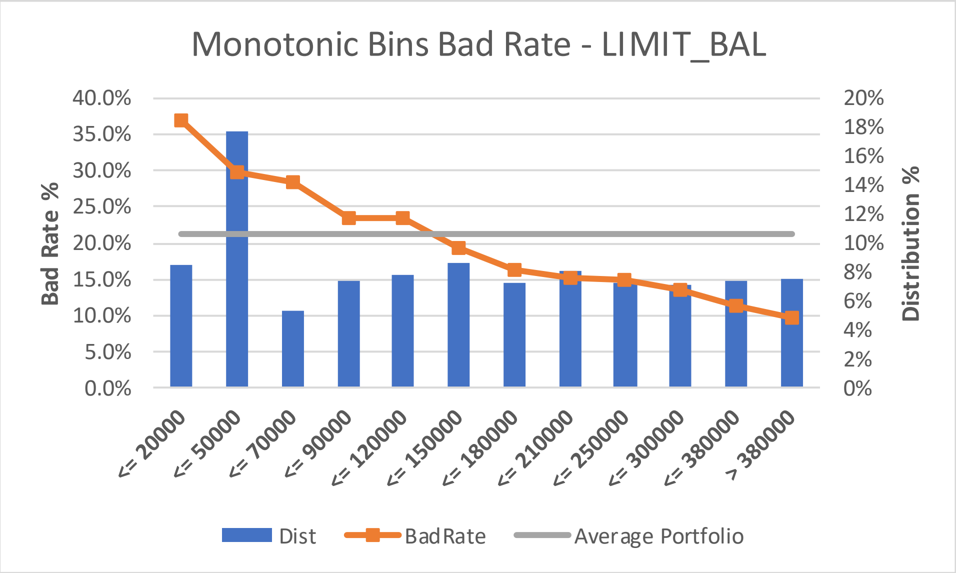

**Figure 6: Monotonic Binning Scheme Bad Rate Plot**

Figure 6 shows the result of performing monotonic binning strategy on credit limit. It can be observed that the number of resulting bins is more compared to decision tree binning and even though the risk remained its monotonicity across bins, however, some bins appear to have closer bad rate. We also compared the predictive strength of both the binning strategies, and observed that decision tree binning produced higher Gini result, i.e 27.9% vs 27.6% in monotonic binning result. Hence, the decision tree binning will be used as the final classing result for the variable and assign WOE scores to the population. The WOE scores is inversely related to risk, i.e higher score implies lower risk. In the multi-factor analysis, final scores will be assigned according to regression output and scaled to have inverse relationship with the risk of default as well. The assignment of WOE scores is shown in Table 4:

**Table 4: WOE scores assignment**

| Bin | Cutpoint | CntGood | CntBad | WOE |
|---|---|---|---|---|
| 1 | <= 40000 | 2111 | 1207 | -0.7527 |
| 2 | <= 70000 | 2744 | 1028 | -0.3299 |
| 3 | <= 140000 | 3507 | 1053 | -0.1086 |
| 4 | <= 260000 | 5305 | 977 | 0.3802 |
| 5 | <= 380000 | 2543 | 346 | 0.683 |
| 6 | > 380000 | 1517 | 164 | 0.9129 |

where,

- CntGood is the Count of Good observation in the bin; and
- CntBad is the Count of Bad observation in the bin.

Next, the assigned WOE scores will be standardized to have mean 0 and standard deviation 1, as given in **Equation 2** above.

The above process is repeated for all variables. We observe that the decision tree and monotonic binning scheme could be generated automatically with algorithm. Modeler typically needs to make final choice of combining the binning result to arrive at final binning scheme, and this makes the process much more efficient as the optimal power scheme has already been given by Decision Tree binning result. The final checking would only need to ensure the result matches underlying economic hypothesis and have intuitive connection with the dependent variable. Table 5 shows the list of variables shortlisting decision using the algorithm described above. Drop/Keep decision was made based on GINI on Train sample, whether it is >= 10%. Information value (IV) on Train sample and GINI on Test sample is also shown in Table 5. We observed the predictive power of the transformed variable to be consistent between Train and Test sample.

**Table 5: Summary of Variables Shortlisted in SFA**

| No | Name | IV | GINI.<br>Train | GINI.<br>Test | Drop/<br>Keep |
|---|---|---|---|---|---|
| 1 | LIMIT_BAL | 25.4% | 27.9% | 25.7% | KEEP |
| 2 | SEX | 1.0% | 5.0% | 5.3% | DROP |
| 3 | EDUCATION | 2.3% | 7.1% | 8.9% | DROP |
| 4 | MARRIAGE | 0.8% | 4.1% | 3.3% | DROP |
| 5 | AGE | 0.1% | 1.2% | 1.4% | DROP |
| 6 | PAY_0 | 95.3% | 40.6% | 43.5% | KEEP |
| 7 | PAY_2 | 68.4% | 35.8% | 36.4% | KEEP |
| 8 | PAY_3 | 52.8% | 32.9% | 30.9% | KEEP |
| 9 | PAY_4 | 44.8% | 29.2% | 26.5% | KEEP |
| 10 | PAY_5 | 40.4% | 26.7% | 25.7% | KEEP |
| 11 | PAY_6 | 34.2% | 24.9% | 23.3% | KEEP |
| 12 | BILL_AMT1 | 0.0% | 0.2% | 2.7% | DROP |
| 13 | BILL_AMT2 | 0.1% | 1.4% | 1.8% | DROP |
| 14 | BILL_AMT3 | 0.2% | 2.0% | 0.3% | DROP |
| 15 | BILL_AMT4 | 0.3% | 2.7% | 0.4% | DROP |
| 16 | BILL_AMT5 | 0.4% | 3.3% | 1.8% | DROP |
| 17 | BILL_AMT6 | 0.8% | 4.5% | 3.5% | DROP |
| 18 | PAY_AMT1 | 15.9% | 20.1% | 20.6% | KEEP |
| 19 | PAY_AMT2 | 16.0% | 12.6% | 12.2% | KEEP |
| 20 | PAY_AMT3 | 10.5% | 17.0% | 16.5% | KEEP |
| 21 | PAY_AMT4 | 7.0% | 14.1% | 17.7% | KEEP |
| 22 | PAY_AMT5 | 7.8% | 14.1% | 13.1% | KEEP |
| 23 | PAY_AMT6 | 7.6% | 14.2% | 15.0% | KEEP |
| 24 | UTIL1 | 10.3% | 16.8% | 13.2% | KEEP |
| 25 | UTIL2 | 12.1% | 18.4% | 15.6% | KEEP |
| 26 | UTIL3 | 13.0% | 19.5% | 17.2% | KEEP |
| 27 | UTIL4 | 14.3% | 20.2% | 18.4% | KEEP |
| 28 | UTIL5 | 14.3% | 20.2% | 19.7% | KEEP |
| 29 | UTIL6 | 14.3% | 19.9% | 19.9% | KEEP |
| 30 | AVG_UTIL_6m | 14.6% | 20.8% | 19.1% | KEEP |
| 31 | AVG_UTIL_5m | 14.1% | 20.3% | 17.3% | KEEP |
| 32 | AVG_UTIL_4m | 13.2% | 19.8% | 16.5% | KEEP |
| 33 | AVG_UTIL_3m | 12.2% | 19.0% | 15.2% | KEEP |
| 34 | AVG_UTIL_2m | 11.5% | 18.0% | 14.9% | KEEP |

| No | Name | IV | GINI. Train | GINI. Test | Drop/ Keep |
|---|---|---|---|---|---|
| 35 | MAX_UTIL_6m | 10.7% | 17.3% | 14.3% | KEEP |
| 36 | MAX_UTIL_5m | 10.3% | 16.5% | 13.5% | KEEP |
| 37 | MAX_UTIL_4m | 10.4% | 17.1% | 13.8% | KEEP |
| 38 | MAX_UTIL_3m | 10.0% | 16.3% | 13.2% | KEEP |
| 39 | MAX_UTIL_2m | 10.8% | 17.3% | 13.8% | KEEP |
| 40 | MAX_BY_AVG_UTIL_6m | 18.6% | 23.0% | 22.4% | KEEP |
| 41 | MAX_BY_AVG_UTIL_5m | 18.3% | 21.7% | 21.6% | KEEP |
| 42 | MAX_BY_AVG_UTIL_4m | 18.4% | 21.0% | 19.0% | KEEP |
| 43 | MAX_BY_AVG_UTIL_3m | 18.6% | 20.6% | 18.7% | KEEP |
| 44 | MAX_BY_AVG_UTIL_2m | 15.0% | 15.5% | 15.2% | KEEP |
| 45 | Curr_bill_perc_max_bill_6m | 0.3% | 2.7% | 3.2% | DROP |
| 46 | Curr_bill_perc_max_bill_5m | 0.3% | 2.9% | 2.6% | DROP |
| 47 | Curr_bill_perc_max_bill_4m | 0.4% | 3.0% | 3.0% | DROP |
| 48 | Curr_bill_perc_max_bill_3m | 0.5% | 3.6% | 3.2% | DROP |
| 49 | Curr_bill_perc_max_bill_2m | 5.9% | 6.1% | 6.1% | DROP |
| 50 | Worst_Status_L6M | 85.8% | 44.4% | 46.6% | KEEP |
| 51 | Worst_Status_L5M | 88.6% | 44.6% | 45.9% | KEEP |
| 52 | Worst_Status_L4M | 89.8% | 44.5% | 45.4% | KEEP |
| 53 | Worst_Status_L3M | 89.5% | 43.4% | 45.2% | KEEP |
| 54 | Worst_Status_L2M | 89.1% | 41.2% | 43.4% | KEEP |
| 55 | Count_Status_GT0_L6M | 96.7% | 47.2% | 48.9% | KEEP |
| 56 | Count_Status_GT0_L5M | 98.1% | 47.0% | 47.9% | KEEP |
| 57 | Count_Status_GT0_L4M | 96.5% | 46.2% | 47.0% | KEEP |
| 58 | Count_Status_GT0_L3M | 94.5% | 44.6% | 46.0% | KEEP |
| 59 | Count_Status_GT0_L2M | 88.8% | 41.2% | 43.2% | KEEP |
| 60 | Count_Status_GT1_L6M | 99.1% | 47.5% | 49.3% | KEEP |
| 61 | Count_Status_GT1_L5M | 99.8% | 47.2% | 48.0% | KEEP |
| 62 | Count_Status_GT1_L4M | 99.9% | 46.7% | 47.4% | KEEP |
| 63 | Count_Status_GT1_L3M | 100.7% | 45.4% | 47.0% | KEEP |
| 64 | Count_Status_GT1_L2M | 98.2% | 42.5% | 44.7% | KEEP |
| 65 | Count_Status_GT2_L6M | 19.8% | 10.3% | 10.8% | KEEP |
| 66 | Count_Status_GT2_L5M | 18.5% | 9.7% | 10.4% | DROP |
| 67 | Count_Status_GT2_L4M | 16.9% | 8.9% | 9.9% | DROP |
| 68 | Count_Status_GT2_L3M | 16.4% | 8.4% | 9.2% | DROP |
| 69 | Count_Status_GT2_L2M | 13.7% | 6.9% | 7.4% | DROP |
| 70 | Mths_since_status_GT0 | 99.5% | 47.3% | 49.3% | KEEP |
| 71 | Mths_since_status_GT1 | 110.8% | 48.9% | 51.0% | KEEP |
| 72 | Mths_since_status_GT2 | 20.5% | 10.4% | 10.8% | KEEP |
| 73 | avg_pmt_as_perc_bill_L5m | 22.6% | 23.0% | 20.4% | KEEP |
| 74 | avg_pmt_as_perc_bill_L4m | 22.6% | 22.8% | 21.1% | KEEP |
| 75 | avg_pmt_as_perc_bill_L3m | 15.4% | 21.4% | 18.6% | KEEP |
| 76 | avg_pmt_as_perc_bill_L2m | 15.9% | 21.2% | 17.6% | KEEP |
| 77 | avg_pmt_as_perc_bill_L1m | 14.0% | 19.8% | 18.8% | KEEP |
| 78 | Cnt_Mth_With_pmt_L6M | 23.4% | 23.9% | 24.9% | KEEP |
| 79 | Cnt_Mth_With_pmt_L5M | 23.6% | 24.0% | 24.7% | KEEP |
| 80 | Cnt_Mth_With_pmt_L4M | 23.5% | 23.8% | 24.5% | KEEP |
| 81 | Cnt_Mth_With_pmt_L3M | 21.7% | 22.2% | 22.5% | KEEP |
| 82 | Cnt_Mth_With_pmt_L2M | 16.7% | 18.2% | 18.0% | KEEP |
| 83 | Cnt_Mth_With_pmt_L1M | 9.3% | 11.4% | 12.3% | KEEP |
| 84 | Count_Pmt_GE_BAL_L5M | 13.5% | 17.6% | 15.0% | KEEP |
| 85 | Count_Pmt_GE_BAL_L4M | 13.0% | 16.9% | 15.4% | KEEP |

| No | Name | IV | GINI.<br>Train | GINI.<br>Test | Drop/<br>Keep |
|---|---|---|---|---|---|
| 86 | Count_Pmt_GE_BAL_L3M | 13.3% | 16.6% | 14.7% | KEEP |
| 87 | Count_Pmt_GE_BAL_L2M | 12.9% | 16.0% | 14.0% | KEEP |
| 88 | Count_Pmt_GE_BAL_L1M | 11.8% | 14.6% | 13.7% | KEEP |

From Table 5, we managed to short list 69 factors covering utilisation factor, delinquency factor, and payment factors. Intuitively, those are the factors which will likely be the key driver that predicts default of customer. In the next process, multifactor analysis will be performed. At the end of this process, two datasets will be produced, which are variables shortlisted before transformation, and variables shortlisted after transformation. As mentioned earlier, the purpose is to test the performance of transformation to the final machine learning models. In credit scorecard model approach, transformed datasets will be used to perform scorecard construction.

### *3.7 Multi Factor Analysis*

Multi-factor Analysis (MFA) has two objectives, which are to select the most predictive combination of factors from the short list, and to determine the most appropriate weighting between these factors in final scorecard. In this stage, various machine learning approaches will be compared with the traditional credit scorecard approach. MFA can be summarized as per the equation below:

**Equation 4: Multi-factor analysis functional form**

$$PD = F(w_0, w_1 f_1, w_2 f_2, \dots, w_n f_n)$$

where $f_x$ denotes the factor, and $F$ is the chosen function. In traditional scorecard approach, logistic regression function is chosen. While in machine learning approach, other functions are chosen. In this function, various weights

$w_x$ is applied to the factors to achieve the best fit. In machine learning model such as Decision Tree, rule-based transformation on the factors will be applied to split the factors to achieve best fit. In this research, we will compare the functions given in Table 6.

**Table 6: List of MFA model type**

| No | Model Type |
|---|---|
| 1 | Credit Scorecard |
| 2 | Decision Tree |
| 3 | Gradient Boosting |
| 4 | Random Forest |
| 5 | Neural Network |
| 6 | Support Vector Machine |
| 7 | K-Nearest Neighbors |

Prior to that, correlation analysis and variable clustering were performed to understand the pairwise correlation between predictors, as some of the technique are susceptible to multicollinearity issue. Collinearity refers to the situation where a pair of predictor variables have high correlation with each other. At times, there could also be relationships between multiple predictors at once, which is known as multicollinearity. The issue is also common in many credit scoring problem where a list of customer payment and delinquent information in past months are analysed, variables which indicate high delinquent in past 2 month is likely to have high delinquent in past 3 month. In general, we should avoid modelling data with highly correlated predictors. Including highly correlated predictors in techniques such as regression analysis might produce highly unstable models, numerical errors, and degraded predictive performance (Kuhn & Johnson, 2013).

Variable Clustering is used for assessing multicollinearity, redundancy, and for separating variables into clusters that can be scored as a single variable, thus resulting in data reduction. Variable Clustering divides the inputs in a predictive modelling data set into disjoint clusters. The inputs included in a cluster are strongly inter-correlated. The algorithm starts with all variables in one single cluster and successively divides it into smaller and smaller clusters. This technique however, should be noted that sometimes variables belonging to different clusters may be correlated. In practice, variable clustering not only offer data reduction, but also allow the modeler to understand relationship between variables, as well as to serve as starting point, for discussion to choose between variables within the cluster that best suits business requirement to form the final scorecard. In this research, we performed clustering analysis to form disjoint clusters. Within the cluster, we further ranked the variables in descending order of GINI on Train sample, then selected the highest GINI variable with Rank = 1. Table 7 shows the variable clustering result on the shortlisted 69 variables.

**Table 7: Variable clustering selection result**

| Cluster | WOE_variable | GINI | IV | Rank | Selected |
|---|---|---|---|---|---|
| 1 | woe_LIMIT_BAL | 28% | 25% | 1 | Y |
| 2 | woe_PAY_0 | 41% | 95% | 1 | Y |
| 3 | woe_PAY_2 | 36% | 68% | 1 | Y |
| 4 | woe_PAY_3 | 33% | 53% | 1 | Y |
| 5 | woe_PAY_4 | 29% | 45% | 1 | Y |
| 6 | woe_PAY_5 | 27% | 40% | 1 | Y |
| 7 | woe_PAY_6 | 25% | 34% | 1 | Y |
| 8 | woe_PAY_AMT1 | 20% | 16% | 1 | Y |
| 9 | woe_PAY_AMT2 | 13% | 16% | 1 | Y |
| 10 | woe_PAY_AMT3 | 17% | 10% | 1 | Y |
| 11 | woe_PAY_AMT4 | 14% | 7% | 1 | Y |
| 12 | woe_PAY_AMT5 | 14% | 8% | 1 | Y |
| 13 | woe_PAY_AMT6 | 14% | 8% | 1 | Y |

| Cluster | WOE_variable | GINI | IV | Rank | Selected |
|---|---|---|---|---|---|
| 14 | woe_UTIL1 | 17% | 10% | 1 | Y |
| 15 | woe_UTIL2 | 18% | 12% | 1 | Y |
| 16 | woe_UTIL3 | 20% | 13% | 1 | Y |
| 17 | woe_UTIL4 | 20% | 14% | 1 | Y |
| 18 | woe_UTIL5 | 20% | 14% | 1 | Y |
| 19 | woe_UTIL6 | 20% | 14% | 1 | Y |
| 20 | woe_AVG_UTIL_5m | 20% | 14% | 2 | N |
| 20 | woe_AVG_UTIL_6m | 21% | 15% | 1 | Y |
| 21 | woe_AVG_UTIL_3m | 19% | 12% | 2 | N |
| 21 | woe_AVG_UTIL_4m | 20% | 13% | 1 | Y |
| 22 | woe_AVG_UTIL_2m | 18% | 11% | 1 | Y |
| 22 | woe_MAX_UTIL_2m | 17% | 11% | 2 | N |
| 23 | woe_MAX_UTIL_5m | 17% | 10% | 2 | N |
| 23 | woe_MAX_UTIL_6m | 17% | 11% | 1 | Y |
| 24 | woe_MAX_UTIL_3m | 16% | 10% | 2 | N |
| 24 | woe_MAX_UTIL_4m | 17% | 10% | 1 | Y |
| 25 | woe_MAX_BY_AVG_UTIL_6m | 23% | 19% | 1 | Y |
| 26 | woe_MAX_BY_AVG_UTIL_5m | 22% | 18% | 1 | Y |
| 27 | woe_MAX_BY_AVG_UTIL_4m | 21% | 18% | 1 | Y |
| 28 | woe_MAX_BY_AVG_UTIL_3m | 21% | 19% | 1 | Y |
| 29 | woe_MAX_BY_AVG_UTIL_2m | 16% | 15% | 1 | Y |
| 30 | woe_Worst_Status_L6M | 44% | 86% | 1 | Y |
| 31 | woe_Worst_Status_L4M | 44% | 90% | 2 | N |
| 31 | woe_Worst_Status_L5M | 45% | 89% | 1 | Y |
| 32 | woe_Count_Status_GT0_L3M | 45% | 95% | 2 | N |
| 32 | woe_Count_Status_GT1_L3M | 45% | 101% | 1 | Y |
| 32 | woe_Worst_Status_L3M | 43% | 90% | 3 | N |
| 33 | woe_Count_Status_GT0_L2M | 41% | 89% | 1 | Y |
| 33 | woe_Worst_Status_L2M | 41% | 89% | 2 | N |
| 34 | woe_Count_Status_GT0_L6M | 47% | 97% | 2 | N |
| 34 | woe_Count_Status_GT1_L6M | 48% | 99% | 1 | Y |
| 35 | woe_Count_Status_GT0_L4M | 46% | 96% | 4 | N |
| 35 | woe_Count_Status_GT0_L5M | 47% | 98% | 2 | N |
| 35 | woe_Count_Status_GT1_L4M | 47% | 100% | 3 | N |
| 35 | woe_Count_Status_GT1_L5M | 47% | 100% | 1 | Y |
| 36 | woe_Count_Status_GT1_L2M | 43% | 98% | 1 | Y |
| 37 | woe_Count_Status_GT2_L6M | 10% | 20% | 2 | N |
| 37 | woe_Mths_since_status_GT2 | 10% | 20% | 1 | Y |
| 38 | woe_Mths_since_status_GT0 | 47% | 99% | 2 | N |
| 38 | woe_Mths_since_status_GT1 | 49% | 111% | 1 | Y |
| 39 | woe_avg_pmt_as_perc_bill_L5m | 23% | 23% | 1 | Y |

| Cluster | WOE_variable | GINI | IV | Rank | Selected |
|---|---|---|---|---|---|
| 40 | woe_avg_pmt_as_perc_bill_L4m | 23% | 23% | 1 | Y |
| 41 | woe_avg_pmt_as_perc_bill_L3m | 21% | 15% | 1 | Y |
| 42 | woe_avg_pmt_as_perc_bill_L2m | 21% | 16% | 1 | Y |
| 43 | woe_avg_pmt_as_perc_bill_L1m | 20% | 14% | 1 | Y |
| 44 | woe_Cnt_Mth_With_pmt_L6M | 24% | 23% | 1 | Y |
| 45 | woe_Cnt_Mth_With_pmt_L5M | 24% | 24% | 1 | Y |
| 46 | woe_Cnt_Mth_With_pmt_L4M | 24% | 23% | 1 | Y |
| 47 | woe_Cnt_Mth_With_pmt_L3M | 22% | 22% | 1 | Y |
| 48 | woe_Cnt_Mth_With_pmt_L2M | 18% | 17% | 1 | Y |
| 49 | woe_Cnt_Mth_With_pmt_L1M | 11% | 9% | 1 | Y |
| 50 | woe_Count_Pmt_GE_BAL_L4M | 17% | 13% | 2 | N |
| 50 | woe_Count_Pmt_GE_BAL_L5M | 18% | 13% | 1 | Y |
| 51 | woe_Count_Pmt_GE_BAL_L2M | 16% | 13% | 2 | N |
| 51 | woe_Count_Pmt_GE_BAL_L3M | 17% | 13% | 1 | Y |
| 52 | woe_Count_Pmt_GE_BAL_L1M | 15% | 12% | 1 | Y |

From the table above, there are 52 clusters and we have chosen the highest Gini among the variables within the same cluster. It can be observed that the selection result from clustering might still have high pairwise correlated variables, which will be further handled using correlated variable selection algorithm. Nonetheless, it provides good alternatives to constructing various candidate scorecards as the selection could be modified to pick the second alternative variable from the same clusters.

Correlated Variable Selection is an algorithm we create to deal with multicollinearity issue. It is improvised with the ability to remove only the variable with less predictive power instead of all correlated variable. Similarly, we use Gini as the predictive strength measure. Steps to implement the algorithm as per below:

1. Compute the correlation matrix of the predictors.

2. Sort the matrix according to Gini from highest to lowest. Determine the pairwise correlation, starting from highest power variable, compare whether it is above threshold with other predictors, if yes add them to drop list.
3. For the second highest power variable, check whether it has been added into drop list, if yes skip to next variable, else Do step 2.
4. Iterate through all variables.
5. Final keep list should be all variables excluding the drop list variables.

In this research, both variable clustering and "correlated variable selection" are compared to study the effectiveness of each method towards the performance of traditional credit scorecard approach. Next multiple modelling methodologies will be tested, which include Credit Scorecard Approach, Basic Classification and Regression Tree Approach, Boosting Approach, Random Forest Approach, Neural Network approach, Support Vector Machine Approach, and K-Nearest Neighbors Approach. In this research, correlated variable selection is applied and the correlation threshold is set as 0.85, which resulted in 31 variables shortlisted for machine learning models training. For traditional scorecard approach, variable clustering, correlated variable selection and stepwise selection are compared to produce the best performance scorecard.

### *3.8 Credit Scorecard*

Traditional Credit Scorecard is generally developed through multivariate regression model using a variety of methods. Methods such as Linear Regression and Logistic Regression are most commonly used. In this research we used Logistic Regression with the WOE transformed variables because the target variable has binary outcome. We performed stepwise selection method with Akaike Information Criterion, AIC as criteria to choose variables entering final scorecard. In terms of variable reduction, we have performed 3 variable selection methods to compare the resulting scorecard and performance. Table 8 shows the comparison of the 3 methods.

**Table 8: Comparison of variable reduction techniques in Credit Scorecard**

| Model | Approach | GINI | |
|---|---|---|---|
| | | Train | Test |
| 1a | Only stepwise selection | 56.35% | 57.50% |
| 1b | Variable Clustering and stepwise selection | 56.37% | 57.43% |
| 1c | Correlated variable selection and stepwise selection | 56.28% | 58.05% |

In model 1a, we used all the transformed variables and performed stepwise selection to result in final model. Throughout the iteration, we further add a criteria to remove variable that counter intuition, that is, coefficients >= 0. Intuitively, the WOE transformed variable should result in negative relationship with the target variable (default). Thus in each iteration the largest coefficient with >= 0 variable will be removed, then proceed to next iteration, until all the coefficients are < 0.

In model 1b, we used variable clustering to group similar variables into clusters, and select the highest predict power factor from each cluster. Then perform the stepwise selection and intuition check iteration. It is noted that the final variable in the scorecard is similar to that in model 1a, except the last variable, Count Months with Payment in past 4 month.

In model 1c, we used the correlated variable selection algorithm to shortlist variables that have pairwise correlation below threshold of 0.5, then perform stepwise selection and intuition check iteration. It is noted that the selection result in final variable is similar to that in model 1a and 1b, except for utilisation factor and payment factor.

Comparison of results between the 3 selection methods shows that the performance is similar, with the variable clustering being slightly favoured in training dataset. Based on the comparison above, there is no material difference in the performance of different approaches as the best approach only perform slightly better by 0.09%. Based on our model selection strategy, model 1b is chosen for the credit scorecard approach as it shows the best performance in training sample. We observed that on the Test sample, this approach does not give the best result but the difference is also negligible at only 0.6%.

In practice, business requirement and implementation plays a greater role when choosing the final scorecard. Banks would typically choose the scorecard that best aligns with their business strategy moving forward. One observation worth noting is the time taken to take the different approaches to arrive at final scorecard. Both the model 1b and 1c has taken significantly less time for iteration due to variable list has been shortened by variable clustering and correlated variable algorithm. Model 1a is much more time consuming as all scorecard variables are required to go through stepwise selection. In practice, more combination of variables will be tested through permutation algorithm, however in this research we only focus on the 3 iterations above. Final score is assigned as per equation below.

**Equation 5: Final Score Assignment**

$$Final\ Score_i = round(WOE_i \times \ \beta_i \times -100)$$

where,

- $\beta_i$ denotes the coefficient of regression for the variable;
- $WOE_i$ denotes the WOE for the attribute; and
- Multiplication by –100 is the selected choice of scaling.

The final score assignment under model 1b is given in Table 9.

**Table 9: Credit Scorecard Scores Assignment**

| Variable | Attribute | WOE | Standardized Score | Coefficient of regression | Final Score |
|---|---|---|---|---|---|
| woe_LIMIT_BAL | <= 40000 | -0.7527 | -1.6169 | -0.12406 | -20 |
| | <= 70000 | -0.3299 | -0.7897 | -0.12406 | -10 |
| | <= 140000 | -0.1086 | -0.3567 | -0.12406 | -4 |
| | <= 260000 | 0.3802 | 0.5996 | -0.12406 | 7 |
| | <= 380000 | 0.683 | 1.1920 | -0.12406 | 15 |
| | > 380000 | 0.9129 | 1.6418 | -0.12406 | 20 |
| woe_PAY_6 | <= -2 | 0.5642 | 0.9345 | -0.09586 | 9 |
| | <= -1 | 0.3478 | 0.5253 | -0.09586 | 5 |
| | <= 0 | 0.1353 | 0.1234 | -0.09586 | 1 |
| | > 0 | -1.3858 | -2.7534 | -0.09586 | -26 |
| woe_PAY_AMT3 | <= 0 | -0.4625 | -1.4748 | -0.05782 | -9 |
| | <= 2901 | -0.0788 | -0.3298 | -0.05782 | -2 |
| | <= 3912 | 0.1048 | 0.2181 | -0.05782 | 1 |
| | <= 15587 | 0.3566 | 0.9695 | -0.05782 | 6 |
| | > 15587 | 0.8617 | 2.4768 | -0.05782 | 14 |
| woe_PAY_AMT4 | <= 396 | -0.2604 | -1.0389 | -0.07416 | -8 |
| | <= 1668 | -0.1809 | -0.7454 | -0.07416 | -6 |
| | <= 4300 | 0.0955 | 0.2748 | -0.07416 | 2 |
| | > 4300 | 0.4314 | 1.5148 | -0.07416 | 11 |
| woe_PAY_AMT5 | <= 0 | -0.3227 | -1.1729 | -0.05639 | -7 |
| | <= 2927 | -0.0904 | -0.3887 | -0.05639 | -2 |
| | <= 14100 | 0.2508 | 0.7633 | -0.05639 | 4 |
| | > 14100 | 0.8606 | 2.8220 | -0.05639 | 16 |
| woe_PAY_AMT6 | <= 910 | -0.2452 | -0.9305 | -0.06904 | -6 |
| | <= 2304 | -0.0979 | -0.4209 | -0.06904 | -3 |
| | <= 9794 | 0.2221 | 0.6861 | -0.06904 | 5 |
| | > 9794 | 0.6679 | 2.2284 | -0.06904 | 15 |
| woe_MAX_UTIL_6m | <= 0.4711 | 0.3408 | 0.9542 | -0.06346 | 6 |
| | <= 0.6099 | -0.0578 | -0.2707 | -0.06346 | -2 |
| | <= 1.019 | -0.2472 | -0.8527 | -0.06346 | -5 |
| | > 1.019 | -0.5861 | -1.8942 | -0.06346 | -12 |
| woe_MAX_BY_AVG_ UTIL_3m | 01 <= 1.1071 | -0.343 | -0.8067 | -0.16056 | -13 |
| | 02 <= 1.5385 | 0.2346 | 0.3352 | -0.16056 | 5 |
| | 03 > 1.5385 | 0.5587 | 0.976 | -0.16056 | 16 |
| | 04 Utilisation <= 0 in last 3 month | 2.7401 | 5.2886 | -0.16056 | 85 |
| woe_Mths_since_status _GT1 | 01 <= 0 | -2.0962 | -2.3706 | -0.81227 | -193 |
| | 02 <= 1 | -0.9778 | -1.2288 | -0.81227 | -100 |

| **Variable** | **Attribute** | **WOE** | **Standardized Score** | **Coefficient of regression** | **Final Score** |
|---|---|---|---|---|---|
| | 03 <= 3 | -0.3609 | -0.5989 | -0.81227 | -49 |
| | 04 > 3 | 0.1667 | -0.0603 | -0.81227 | -5 |
| | 05 All status <= 1 in last 6 month | 0.7862 | 0.5722 | -0.81227 | 46 |
| woe_Cnt_Mth_With_pmt_L4M | <= 3 | -0.5643 | -1.3228 | -0.07905 | -10 |
| | > 3 | 0.4245 | 0.7559 | -0.07905 | 6 |

### *3.9 Decision Tree*

Basic Classification and Regression Tree Approach is a machine learning technique that strives to partition the data into smaller groups, such that the resulting groups become more homogenous with respect to the response. Various considerations are examined in training model of this type. Firstly, the tree will determine the feature or predictor to split on and value of the split. In this paper, the value of the split is referred to as cutpoint. Secondly, the depth or complexity of the tree. Depth and complexity of the tree are typically specified as tuning parameters for model training. Higher depth parameter allows for a more complex model, however, it may be prone to overfitting. Thirdly, the prediction equation in the terminal nodes. There are a variety of methods for constructing regression trees. In this paper, we will compare and select the best from 2 methods, which are the classification and regression tree (CART) methodology of (Breiman, Friedman, Stone, & Olshen, 1984), and conditional inference tree methodology of (Hothorn, Hornik, & Zeileis, 2006). Beginning with entire dataset, the model searches every distinct value of every predictor to identify the predictor and cutpoint that partitions the data into two groups. The overall sum of squares error (difference between actual and group mean) in the two groups will be minimized in the process. Then, the process is repeated for the resulting output from the previous process, until the number of instances in the resulting sample fall below a certain threshold. One disadvantage of the CART, as highlighted by (Loh & Shih, 1997) is that predictors with higher number of distinct values tend to be favored over the others for splitting the top nodes of the tree. Hothorn, Hornik, and Zeileis (2006) proposed a unified framework for unbiased tree-based models for regression and classification models training. The framework is also known as Conditional Inference Tree approach that uses an unbiased selection of variable at each split. In the process of constructing conditional inference tree, statistical hypothesis tests will be used to do exhaustive search across the list of predictors and their possible split points. When a potential split is being considered, a statistical test will be conducted to test the difference between the means of the two resulting groups created by the split. A p-value will be computed for the test. Correction to the raw p-value will be applied to reduce the selection bias caused by large number of split candidates. As more split candidates will produce higher number of false-positive test results.

In this research, we compare the basic classification and regression tree as well as conditional inference tree, and observe that the conditional inference tree produces a slightly better result, with Gini on train sample at 58.1%. Below is the tuning performed on the conditional inference tree model training process.

**Table 10: Tuning Grid for Conditional Inference Tree algorithm**

| mincriterion |
|---|
| 0.05 |
| 0.10 |
| 0.15 |
| 0.20 |
| 0.25 |
| 0.30 |
| 0.35 |
| 0.40 |
| 0.45 |
| 0.50 |
| 0.55 |
| 0.60 |
| 0.65 |
| 0.70 |
| 0.75 |
| 0.80 |
| 0.85 |

| |
|---|
| 0.90 |
| 0.95 |

From the table above, the approach was tuned with one parameter, mincriterion. It is the value of the test statistic or 1– p-value that must be exceeded in order to implement a split. Based on highest GINI criteria, the final selected best tuned parameter is mincriterion = 0.90.

### *3.10 Gradient Boosting*

Boosting is an ensemble technique where many classification models were combined into a final model. The classification models to be combined are also known as weak classifiers, which predicts marginally better than no model. In this approach, new models are added to correct the errors made by existing models, and models are added sequentially until no further improvements can be made. Gradient boosting approach begins with weak prediction model which is initial guess of the target variable, typically mean of the target variable. Then, given a loss function, for example defined by squared error, the approach seeks to find an additive model that fit to the existing model residuals to minimize the loss function. It is called gradient boosting because it uses a gradient descent algorithm to minimize the loss when adding new models. At each iteration, the algorithm will look for the best prediction model based on the current sample weights. Samples that have been incorrectly classified in the current iteration will receive more weight in the next iteration. On the other hand, samples that have been correctly classified will receive less weight in the subsequent iteration. In essence, each iteration of the algorithm is tasked to learn a different aspect of the data not modelled by the previous iterations. In order to prevent overfitting, regularization strategy and learning rate can be tuned to constrain the learning process. The regularization strategy works as such, at the end of each iteration, supposed a simple gradient boosting algorithm will update the predicted value by adding predicted value from previous iteration to predicted value of current iteration. By constraining the learning rate, the process adds only a fraction (denoted as eta in Table 11) of current iteration's predicted value to previous iteration's predicted value. This regularization works similar to L2 regularization. A small value of eta would cause the training process to take longer computation time. In this paper, we fixed the learning rate at 0.10.

There are various gradient boosting algorithms. In this paper, we compare between Stochastic Gradient Boosting and Extreme Gradient Boosting algorithm. It is observed that the Extreme Gradient Boosting have slightly better performance, with Gini on train sample at 62.8%. Below is the tuning performed on the extreme gradient boosting model training process.

**Table 11: Tuning Grid for Extreme Gradient Boosting algorithm**

| nrounds | max_depth | colsample_bytree | eta | gamma | min_child_weight | subsample |
|---|---|---|---|---|---|---|
| 100 | 3 | 0.5 | 0.1 | 0 | 1 | 1 |
| 200 | 3 | 0.5 | 0.1 | 0 | 1 | 1 |
| 100 | 5 | 0.5 | 0.1 | 0 | 1 | 1 |
| 200 | 5 | 0.5 | 0.1 | 0 | 1 | 1 |
| 100 | 10 | 0.5 | 0.1 | 0 | 1 | 1 |
| 200 | 10 | 0.5 | 0.1 | 0 | 1 | 1 |
| 100 | 3 | 0.6 | 0.1 | 0 | 1 | 1 |
| 200 | 3 | 0.6 | 0.1 | 0 | 1 | 1 |
| 100 | 5 | 0.6 | 0.1 | 0 | 1 | 1 |
| 200 | 5 | 0.6 | 0.1 | 0 | 1 | 1 |
| 100 | 10 | 0.6 | 0.1 | 0 | 1 | 1 |
| 200 | 10 | 0.6 | 0.1 | 0 | 1 | 1 |
| 100 | 3 | 0.7 | 0.1 | 0 | 1 | 1 |
| 200 | 3 | 0.7 | 0.1 | 0 | 1 | 1 |
| 100 | 5 | 0.7 | 0.1 | 0 | 1 | 1 |
| 200 | 5 | 0.7 | 0.1 | 0 | 1 | 1 |
| 100 | 10 | 0.7 | 0.1 | 0 | 1 | 1 |
| 200 | 10 | 0.7 | 0.1 | 0 | 1 | 1 |
| 100 | 3 | 0.8 | 0.1 | 0 | 1 | 1 |

| nrounds | max_depth | colsample_bytree | eta | gamma | min_child_weight | subsample |
|---|---|---|---|---|---|---|
| 200 | 3 | 0.8 | 0.1 | 0 | 1 | 1 |
| 100 | 5 | 0.8 | 0.1 | 0 | 1 | 1 |
| 200 | 5 | 0.8 | 0.1 | 0 | 1 | 1 |
| 100 | 10 | 0.8 | 0.1 | 0 | 1 | 1 |
| 200 | 10 | 0.8 | 0.1 | 0 | 1 | 1 |
| 100 | 3 | 0.9 | 0.1 | 0 | 1 | 1 |
| 200 | 3 | 0.9 | 0.1 | 0 | 1 | 1 |
| 100 | 5 | 0.9 | 0.1 | 0 | 1 | 1 |
| 200 | 5 | 0.9 | 0.1 | 0 | 1 | 1 |
| 100 | 10 | 0.9 | 0.1 | 0 | 1 | 1 |
| 200 | 10 | 0.9 | 0.1 | 0 | 1 | 1 |

From the table above, the approach was tuned with three parameters, which are nrounds, max_depth, and colsample_bytree. Description of the three and other parameters are below:

- nrounds is the maximum number of iterations.
- max_depth is the maximum depth of the tree.
- colsample_bytree is the percentage of features supplied to a tree.
- eta is the learning rate. After every round, it shrinks the feature weight to reach the best optimum.
- gamma is the control for regularization.
- min_child_weight is the minimum number of instance weight in child node to implement a split.
- subsample is the percentage of observations supplied to a tree.

Based on highest GINI criteria, the final selected best tuned parameter is nrounds = 100, max_depth = 3, eta = 0.1, gamma = 0, colsample_bytree = 0.5, min_child_weight = 1, subsample = 1.

### *3.11 Random Forest*

Random Forest is an ensemble technique that build *m* number of trees or models and aggregate the result to form the final prediction. Each tree in the "forest" casts a vote and the average of these predictions will be the forest's prediction. One similarity between Random Forest and tree bagging is that during each tree training, a bootstrap sample will be generated for model training, and the result is aggregation of all tree's prediction. However, one difference is that in Random Forest approach, during iteration *k* number of random predictors will be supplied to the model to consider at each split. The model then selects the best predictor among the *k* predictors for partitioning the data. The resulting models will be used to generate prediction on new sample and the aggregated result will be the final model prediction. In this paper, we studied the Random Forest approach and not the bagging approach. The tuning parameter *k* is also referred to as mtry. Below is the tuning performed on the random forest model training process.

**Table 12: Tuning Grid for Random Forest algorithm**

| mtry |
|---|
| 5 |
| 10 |
| 15 |
| 20 |
| 25 |
| 30 |
| 35 |
| 40 |
| 45 |
| 50 |

From the table above, the approach was tuned using mtry parameter. It is the number of features supplied to a tree during the iteration. Based on highest GINI criteria, the final selected best tuned parameter is mtry = 5.

### *3.12 Neural Network*

In recent years, Neural Network have been discussed extensively as an alternative to the parametric models as given in traditional scorecard approach. They offer a more flexible design to represent the connections between independent and dependent variables (Hayden & Porath, 2011). Neural network is a complex nonlinear modelling algorithm that is inspired by theories about how human brain works. A neural network model is represented by a number of layers, each layer containing computing elements known as *neurons* or *units*. Each unit in a layer takes inputs from the preceding layer and computes outputs. Then, the outputs from the neurons in one layer become inputs to the next layer in the sequence of layers. The first layer is the input layer, and the last layer is the output layer. In between the input and output layers there can be a number of hidden layers. The units in a hidden layer is also known as *hidden units*. The *hidden units* perform intermediate calculations and pass the results to the next layer. The calculations involve combining the inputs they receive from the previous layer and performing a mathematical transformation on the combined values. The transformation is also known as activation function, which is typically the linear or logistic function. In this paper, the logistic function is chosen as the target variable is binary. Neural Network models are often adversely affected by multicollinearity issue. In this research, variable selection is performed using correlated variable selection algorithm prior to modelling with neural network. There are various types of neural network architecture, we choose the best between the single hidden layer network, model averaged neural network, and multi-layer perceptron neural network. Single hidden layer network is the simplest network with only one hidden layer and varying number of neurons. Model average neural network will aggregate result from several neural network models. Multi-layer perceptron neural network is built with 3 hidden layers and different number of neurons. In this research, we have compared the performance between Artificial Neural Network (single hidden layer), Model Averaged Neural Network, and Multilayer Perceptron Neural Network. It is found that the Artificial Neural Network performed slightly better, with Gini on train sample at 61.0%. Below is the tuning performed on the artificial neural network model training process.

**Table 13: Tuning Grid for Artificial Neural Network algorithm**

| size | decay |
|---|---|
| 5 | 0.01 |
| 10 | 0.01 |
| 25 | 0.01 |
| 30 | 0.01 |
| 5 | 0.1 |
| 10 | 0.1 |
| 25 | 0.1 |
| 30 | 0.1 |
| 5 | 0.5 |
| 10 | 0.5 |
| 25 | 0.5 |
| 30 | 0.5 |
| 5 | 1 |
| 10 | 1 |
| 25 | 1 |
| 30 | 1 |

From the table above, the approach was tuned with the size parameter and decay parameter. Size parameter is the number of units (neurons) in hidden layer, and decay is the regularization parameter. Based on highest GINI criteria, the final selected best tuned parameter is size = 25, decay = 1.

### *3.13 Support Vector Machine*

Support Vector Machines (SVM) modelling is a machine learning method that is “model free” method which does not require assumptions of distribution and interdependency of predictor variables. The theory behind SVM was originally developed in the context of classification models and later extended to regression models. According to Gunn (1998), SVM can also be applied to regression problems by the introduction of an alternative loss function. In SVM, each data point is represented as a *n*-dimensional vector. The algorithm creates an *n*-1-dimensional separating hyperplane to discriminate two classes which will maximize the distance between the hyperplane and data points on each side. A separating hyperplane is called optimum if it can classify without error and distance between adjacent vectors is maximal, where the closest vector with hyperplane called support vector. Non-linear functions, also known as kernel function can be used to transform data into multidimensional space. The use of non-linear kernel to transform input data into higher dimension space allows for nonlinear classification boundaries. There are various types of kernel functions available to transform the inputs. In this research, the Gaussian Radial Basis Function (RBF) will be used for our SVM approach. Below is the tuning performed on the SVM model training process.

**Table 14: Tuning Grid for Support Vector Machine algorithm**

| sigma | C |
|---|---|
| 0.001 | 0.015625 |
| 0.003 | 0.015625 |
| 0.005 | 0.015625 |
| 0.001 | 0.03125 |
| 0.003 | 0.03125 |
| 0.005 | 0.03125 |
| 0.001 | 0.0625 |
| 0.003 | 0.0625 |
| 0.005 | 0.0625 |
| 0.001 | 0.125 |
| 0.003 | 0.125 |
| 0.005 | 0.125 |
| 0.001 | 0.25 |
| 0.003 | 0.25 |
| 0.005 | 0.25 |
| 0.001 | 0.5 |
| 0.003 | 0.5 |
| 0.005 | 0.5 |
| 0.001 | 1 |
| 0.003 | 1 |
| 0.005 | 1 |
| 0.001 | 2 |
| 0.003 | 2 |
| 0.005 | 2 |
| 0.001 | 4 |
| 0.003 | 4 |
| 0.005 | 4 |

From Table 14, the approach was tuned with the sigma and C parameters. Sigma is a kernel parameter that defines how much influence a single training sample has. C is the cost of error parameter for decision surface simplicity, which mainly controls the complexity of the model to prevent over-fitting. Based on highest GINI criteria, the final selected best tuned parameter is sigma = 0.001, C = 0.25.

### *3.14 K-Nearest Neighbors*

*K*-Nearest Neighbors (KNN) is a non-parametric approach that can be used in classification and regression. KNN predicts new sample classification by using a sample's geographic neighbourhood. For each data point, the algorithm finds the *k* closest observations in the training dataset, then makes prediction based on the majority of the *k* closest observations. "Closeness" is determined by a distance metric, typically the Euclidean distance or Minkowski distance (a generalization of Euclidean distance). The choice of metric depends on the characteristics of the predictors. Regardless which distance metric is chosen, it should be noted that the measurement scales of the predictors affect the resulting distance calculations. Hence if the set of predictors are on different scales, the distance value between samples will be biased towards predictors with larger scales. In this research, variables were standardized to have the same scales, with mean 0 and standard deviation 1. Class probability is estimated on new sample by calculating the proportion of training set neighbors in each class. Predicted class of new sample will be the class with highest probability estimate. Sometimes, the predicted classes with highest probability estimate will have a tie situation, especially when *k* equals to even number. When this happens, the tie will be either broken at random, or by looking ahead to the *K* + 1 closest neighbors. In this paper, second approach is taken to break the ties. Below is the tuning performed on the KNN model training process.

**Table 15: Tuning Grid for K-Nearest Neighbors algorithm**

| *k* |
|---|
| 100 |
| 120 |
| 140 |
| 160 |

| *k* |
|---|
| 180 |
| 200 |
| 220 |
| 240 |
| 260 |
| 280 |
| 300 |
| 320 |
| 340 |
| 360 |
| 380 |
| 400 |

From the table above, the approach was tuned using *k* parameter. It is the number of neighbors in training dataset to be used in predicting new sample. Based on highest GINI criteria, the final selected best tuned parameter is *k* = 220.

### *3.15 Performance of Scoring Approaches*

After trained and obtained various best tuned machine learning models, below are the final result in comparison to the traditional logistic regression credit scorecard approach.

**Table 16: Performance of various scoring approaches**

| Model | Approach | GINI | | GINI Standard Deviation |
|---|---|---|---|---|
| | | Train | Test | Cross Validation |
| 1 | Credit Scorecard | 56.37% | 57.43% | NA |
| 2 | Conditional Inference Tree | 58.13% | 58.44% | 0.022 |
| 3 | Gradient Boosting | 62.81% | 60.38% | 0.022 |
| 4 | Random Forest | 56.81% | 57.09% | 0.021 |
| 5 | Neural Network | 61.02% | 59.15% | 0.022 |
| 6 | Support Vector Machine | 51.80% | 54.45% | 0.023 |

|  |  |  |  |  |
|---|---|---|---|---|
| 7 | K-Nearest Neighbors | 59.30% | 58.36% | 0.022 |

From the result shown in Table 16, we observe that extreme gradient boosting approach has the best discriminatory power. It is also noted that the artificial neural network model has second best performance compared to other approaches. On the Test sample, we see that the traditional credit scorecard approach has comparable performance 57% vs 60% from machine learning technique. We also observe that the standard deviation of GINI performance from each machine learning model on cross validation’s sample are similar at approximately 0.02. It is expected that the models to perform with GINI within ±2% (that is, 1 standard deviation) when scoring on new sample. It might be beneficial to apply machine learning model to improve the credit underwriting performance scorecard. It is recommended for banks to adopt the dual scoring approach, whereby traditional scorecard and machine learning scorecard are both used to score applicants, and the cut-off score to accept customers can be improvised by accepting customers who are scored under “Refer” risk grades in traditional scorecard and scored under “Accept” risk grades in machine learning scorecards. To achieve that, it is important to ensure that the machine learning scorecards produce accurate and conservative PD outcome. Any adjustment done to the PD estimation in traditional scorecard should also be done to the machine learning models. In the next section, we demonstrate the proposed PD calibration methodology on machine learning models and evaluate the performance of the PD models.

### *3.16 Probability of Default Calibration*

The output of the scorecards score produced by various modelling methodology as described in the above sections are used to rank order the customers by risk. However, to ensure the predicted probability is reflective of the actual underlying probability, the scores need to be calibrated. Calibration process aims to define a mathematical relationship between score and PD. In this research, the class predictions output and the target variable values (default/non-default) from the entire dataset (combine train and test datasets) will be used in post-processing the probability estimate based on the following equation (Platt, 2000),

**Equation 6: Calibration Function**

$$PD = \frac{exp^{\beta_0 + \beta_1 * s(x)}}{1 + exp^{\beta_0 + \beta_1 * s(x)}}$$

where *s*(*x*) is the model score output by various modelling techniques. The $\beta_0$ and $\beta_1$ are the parameters estimated by predicting the true default outcome as a function of the uncalibrated model scores or probabilities, *s*(*x*).

Default experience and borrowers' behavioural score of a loan portfolio fluctuates with changes in the macro-economic environment due to scorecard cyclicality. The average of these observed default rates should represent the Central Tendency of default for the portfolio, which is the long run average PD through the anticipated credit cycle, and therefore will partly incorporate the impact of cyclical downturn in future. Under the Basel II Framework, the predicted PD used for capital calculation should represent the portfolio long run average PD throughout the economic cycle. The capital requirement, *K* for credit card portfolio is given as per below (Basel Committee on Banking Supervision, 2005):

$$K = LGD \times \mathrm{N}\left(\sqrt{\frac{1}{(1-\rho)}}\, N^{-1}(PD) + \sqrt{\frac{\rho}{1-\rho}}\, N^{-1}(0.999)\right) - PD \times LGD$$

and the risk weight, RWA equation below:

$$RWA = K \times 12.5 \times EAD$$

where

*LGD*: Loss Given Default during adverse economic scenario,

*EAD*: Exposure at Default during adverse economic scenario,

*N* is cumulative standard Normal Distribution,

$N^{-1}$ is the inverse cumulative standard normal distribution and

$\rho$ is the asset correlation, for Qualifying Revolving Retail Exposure is assumed to be 4%.

The capital requirement formula consists of converting the average PD into a conditional PD using the Merton's single asset model to credit portfolios. The capital requirement formula above ensures that adequate capital is held in reserve for downturn. This would also avoid excessive origination during economic boom time and vice versa. Further details could be found on the Explanatory Note on the Basel II IRB Risk Weight Functions (Basel Committee on Banking Supervision, 2005). When constructing a scorecard that does not have sample which covers an economic cycle, it is necessary to make adjustment such that the predicted PD represents the portfolio long run average default experience.

In this research, we propose to use the **Equation 6** to produce the PD estimate using machine learning scores that can be comparable to a Basel compliant credit underwriting scorecard. Through the calibration function in

**Equation 6**, we could run weighted logistic regression, by which the sample's class weight is adjusted to match the portfolio target PD before the calibration. However, if the equation does not produce a well calibrated PD, we could perform bucketing analysis on the machine learning scores, and obtain empirical default rate from risk buckets to be used as the predicted PD for the risk bucket.

### *3.17 PD model Assessment*

For the validation of PD model, it could be differentiated into 2 stages: validation of the discriminatory power of a rating system, and validation of the accuracy of the PD quantification (Basel Committee on Banking Supervision, 2005). In this research, our PD models has the same discriminatory power measured by Gini, as the underlying scoring models. Two tests are proposed to assess the PD model performance across all machine learning models and traditional scorecard. First test is called the Calibration Plot and second test is called the Binomial Test. In practice, there are more tests to be performed, however in this paper, these 2 tests are presented to gauge the effectiveness of the machine learning models when applied to credit underwriting scorecard.

#### *3.17.1 Calibration Plot*

Calibration plot works as follow, firstly, perform scoring on the samples with known outcome values using the prediction models. In this research, the prediction models would be credit scorecard as well as the 6 machine learning models. Next, bin the data into 10 groups according to their predicted class probability. Then, create set of bins [0, 10 %], (10 %, 20 %], . . ., (90 %, 100 %] which represents the 10 groups. For each resulting group or bin, determine the observed default rate, also called actual PD rate % in this paper. For example, suppose that 100 observations belong to the first bin, [0, 10 %] for predicted PD less than equal 10%. Suppose there were three default events in the bin, then the actual PD rate % would be 3%. The midpoint of the bin [0, 10 %] is 5%. The calibration plot displays the midpoint of each bin on the horizontal axis and the actual PD rate % on the vertical axis. If the points fall along a 45 ◦ line, this implies that the prediction model has produced well-calibrated probability. Assessment is done visually from the closeness of actual default rate along the 45 ◦ line. However, the actual mid PD of the bins might be different. We propose an additional assessment using the actual mid PD of the bin, which is the sum-of-square, SSE of all the bins.

$$SSE = \sum_{i=1}^{10} (y_i - \bar{y}_i)^2$$

Where $\bar{y}_i$ is the actual mid PD of bin *i*,

And $y_i$ is the actual default rate in bin *i*

The lower the SSE, the closer is the calibrated PD to the actual default rate.

Result of Calibration Plot test for credit scorecard and machine learning models are shown in figures below.

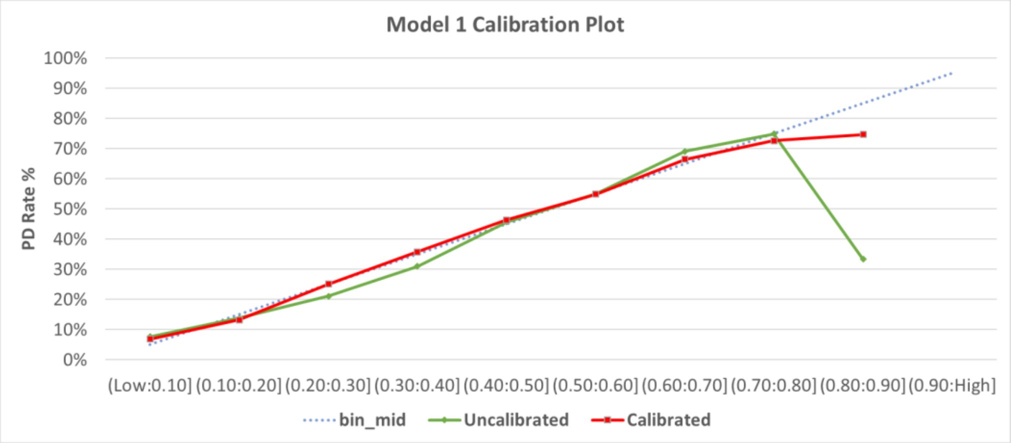

**Figure 7: Credit Scorecard Model Calibration Plot**

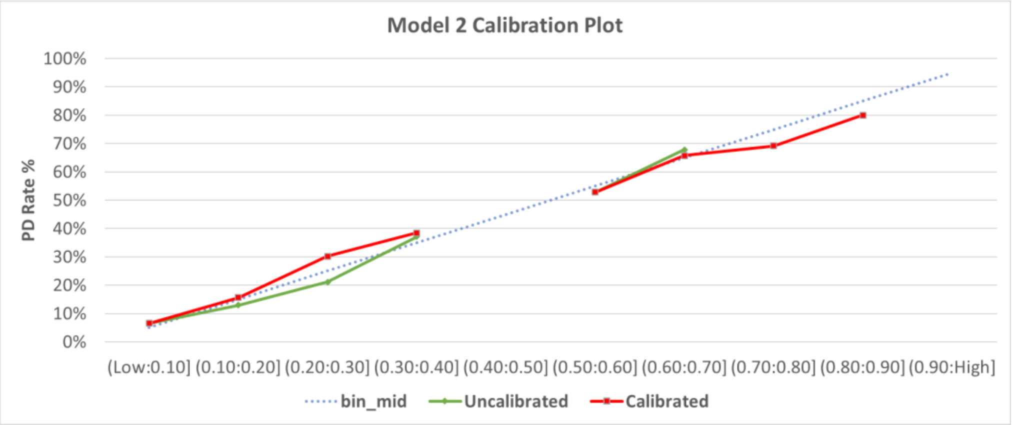

**Figure 8: Decision Tree Model Calibration Plot**

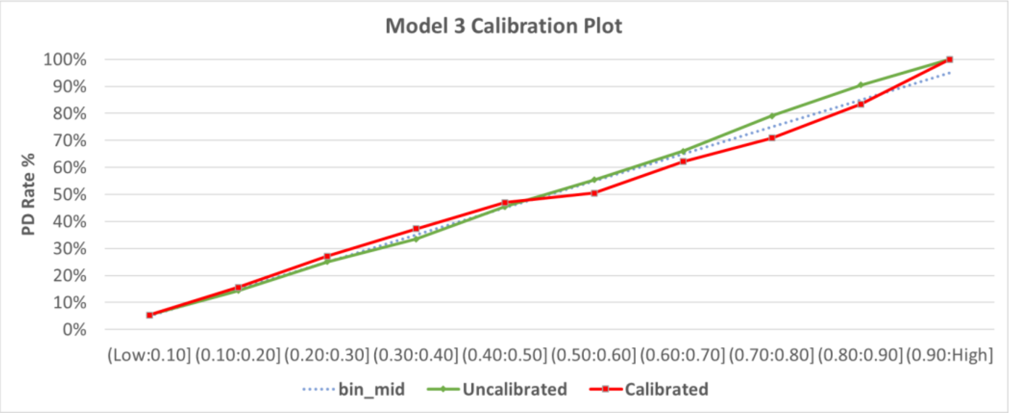

**Figure 9: Gradient Boosting Model Calibration Plot**

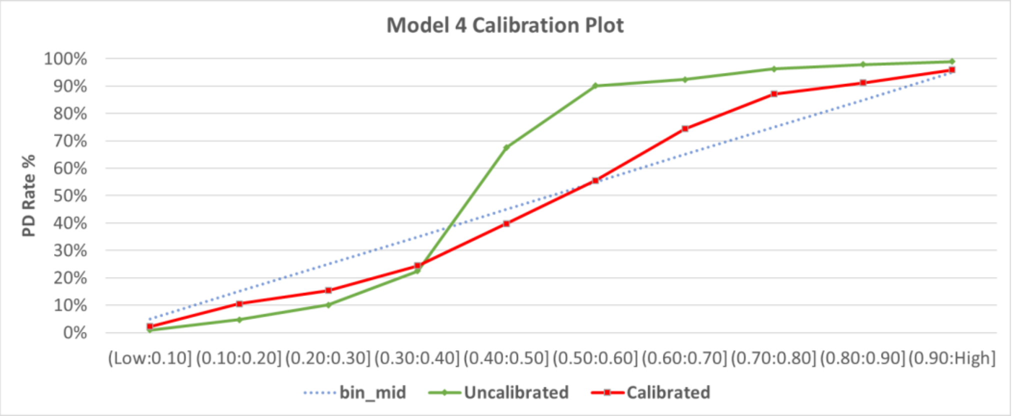

**Figure 10: Random Forest Model Calibration Plot**

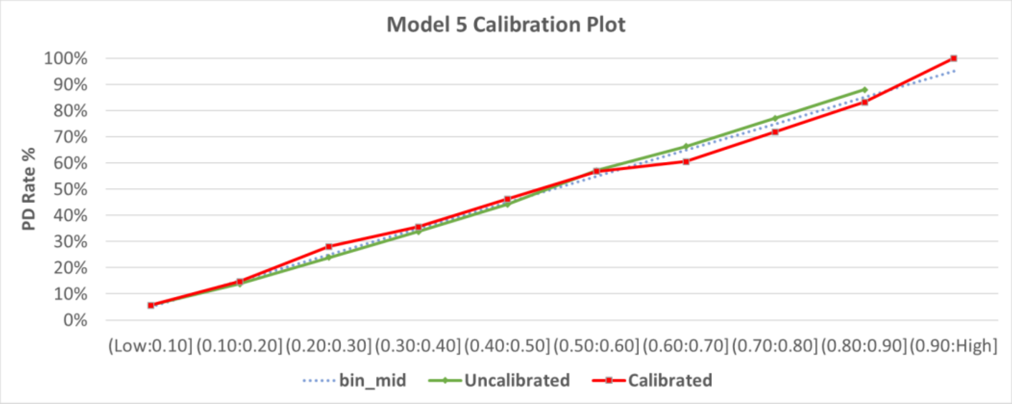

**Figure 11: Neural Network Model Calibration Plot**

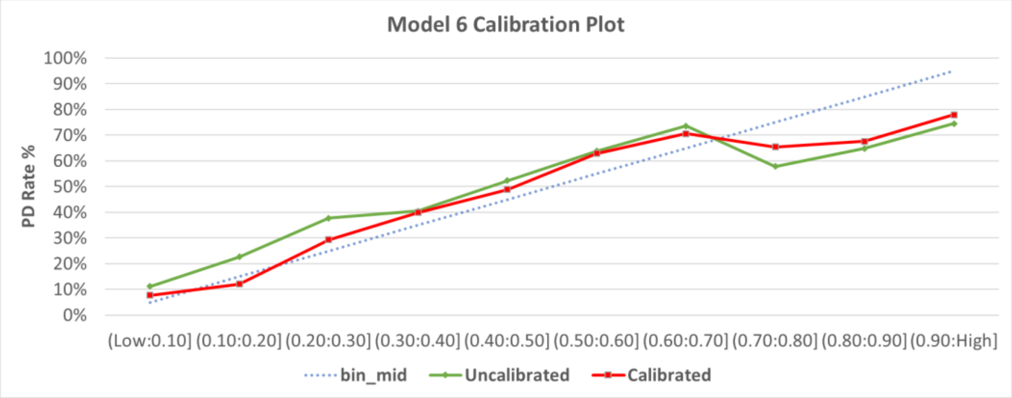

**Figure 12: Support Vector Machine Model Calibration Plot**

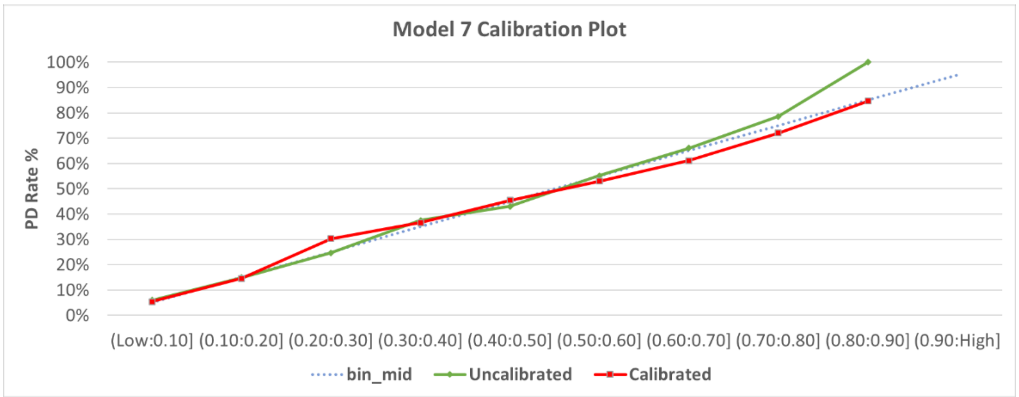

**Figure 13: K-Nearest Neighbors Model Calibration Plot**

From the figures above, the use of **Equation 6** to post process the scores had resulted in better PD estimate. SSE measure of each models are given in table below.

**Table 17: Comparison of various PD model in SSE test**

| Model | Approach | SSE |
|---|---|---|
| 1 | Credit Scorecard | 0.006 |
| 2 | Conditional Inference Tree | 0.007 |
| 3 | Gradient Boosting | 0.017 |
| 4 | Random Forest | 0.049 |
| 5 | Neural Network | 0.017 |
| 6 | Support Vector Machine | 0.076 |
| 7 | K-Nearest Neighbors | 0.010 |

From Table 17, traditional credit scorecard approach gives the lowest SSE, other approaches do not differ much, except for Random Forest and Support Vector machine models. It can be seen from the calibration plot that the Random Forest model tends to underestimate PD for low risk customer, and overestimate PD for high risk customers. Support Vector Machine approach gives the highest SSE, possibly due to the low discriminatory power in comparison to other approaches.

### *3.17.2 Binomial Test*

Binomial test is a common test performed on the PD estimates of a credit scorecard. In this paper, the test was performed on the 7 models. In the Calibration of PD, binomial test provides a conservative indicator (Sun & Wang, 2005). Common practice for banks is to use discrete PD estimates represented by a master scale of rating grades. Binomial test for rating grades works as such, consider one single rating grade over a single time period, usually 1 year. A certain rating grade, $k \in \{1, \ldots, K\}$ is chosen, and additionally, assume independence of default events between all credits within the chosen rating grade $k$. This assumption implies that the number of defaults in rating grade $k \in \{1, \ldots, K\}$ can be modelled as a binomially distributed random variable, $X$ with size parameter $N_k$ and "success" probability $PD_k$. Hypothesis test is shown below.

$\boldsymbol{H_0}$: The estimated PD of the rating category is conservative enough, i.e. the actual default rate is less than or equal to the forecasted default rate,

$\boldsymbol{H_1}$: The estimated PD of the rating category is less than the actual default rate.

The null hypothesis $\boldsymbol{H_0}$ is rejected at a confidence level $\alpha$ whenever the number of observed defaults in this rating grade is greater than or equal to the critical value. The critical value, $d_\alpha$ is computed as shown in the equation below.

**Equation 7: Binomial Test's critical value for number of defaulters**

$$d_\alpha \ = \min \{d : \sum_{j=d}^{N_k} \binom{N_k}{j} PD_k^j \left(1 - \ PD_k^j\right)^{N_k - j} \ \leq 1 - \ \alpha\}$$

In light that binomial test ignores the effects of economic fluctuation and asset correlation, it generally underestimates $d_\alpha$. The Basel II framework assumes Credit Card, which is classified as Qualifying Revolving Retail Exposures, to have asset correlation at 4%.

In this paper, Binomial test was performed using the bins formed in Calibration Plot. Each bin or pool was tested whether actual observed bad is greater or equal to the critical value, if null hypothesis is rejected, test result was denoted as "Failed", indicating the prediction for the bin is not conservative. Total number of "Failed" pool divided by total number of pools tested, result in % Failed which indicates whether the overall rating system is conservative enough. List below shows the terminology used in the test:

- Bin: PD pool formed using Calibration Plot test
- Cnt: number of observations in the bin
- Cntgood: number of good observations in the pool
- Cntbad: number of bad observations in the pool
- midPD: Predicted PD of the pool
- *k**: Binomial test's critical value for number of defaulters

Result of Binomial Test for credit scorecard and machine learning models are shown in tables below.

**Table 18: Credit Scorecard Model Binomial Test Result**

| bin | cnt | cntgood | cntbad | midPD | *k** | Binomial Test |
|---|---|---|---|---|---|---|
| B01: (Low - 0.1] | 6604 | 6150 | 454 | 9% | 650 | PASSED |
| B02: (0.1 - 0.2] | 14229 | 12345 | 1884 | 12% | 1786 | FAILED |
| B03: (0.2 - 0.3] | 1599 | 1202 | 397 | 25% | 426 | PASSED |
| B04: (0.3 - 0.4] | 1180 | 755 | 425 | 35% | 436 | PASSED |
| B05: (0.4 - 0.5] | 927 | 497 | 430 | 45% | 444 | PASSED |
| B06: (0.5 - 0.6] | 825 | 372 | 453 | 55% | 477 | PASSED |
| B07: (0.6 - 0.7] | 1001 | 336 | 665 | 65% | 679 | PASSED |
| B08: (0.7 - 0.8] | 1501 | 411 | 1090 | 75% | 1151 | PASSED |
| B09: (0.8 - 0.9] | 261 | 66 | 195 | 81% | 223 | PASSED |
| | | | | | # Failed | 1 |
| | | | | | # Pool | 9 |
| | | | | | % Failed | 11% |

**Table 19: Decision Tree Model Binomial Test Result**

| bin | cnt | cntgood | cntbad | midPD | *k** | Binomial Test |
|---|---|---|---|---|---|---|
| B01: (Low - 0.1] | 8963 | 8379 | 584 | 9% | 841 | PASSED |
| B02: (0.1 - 0.2] | 13242 | 11173 | 2069 | 14% | 1914 | FAILED |
| B03: (0.2 - 0.3] | 580 | 405 | 175 | 25% | 160 | FAILED |
| B04: (0.3 - 0.4] | 1160 | 714 | 446 | 36% | 440 | FAILED |
| B06: (0.5 - 0.6] | 1472 | 694 | 778 | 53% | 814 | PASSED |
| B07: (0.6 - 0.7] | 745 | 255 | 490 | 69% | 533 | PASSED |
| B08: (0.7 - 0.8] | 1115 | 344 | 771 | 72% | 824 | PASSED |
| B09: (0.8 - 0.9] | 850 | 170 | 680 | 83% | 724 | PASSED |
| N/A | N/A | N/A | N/A | N/A | N/A | N/A |
| | | | | | # Failed | 3 |
| | | | | | # Pool | 8 |
| | | | | | % Failed | 38% |

### Table 20: Gradient Boosting Model Binomial Test Result

| bin | cnt | cntgood | cntbad | midPD | *k** | Binomial Test |
|---|---|---|---|---|---|---|
| B01: (Low - 0.1] | 9385 | 8883 | 502 | 8% | 834 | PASSED |
| B02: (0.1 - 0.2] | 11792 | 9949 | 1843 | 13% | 1593 | FAILED |
| B03: (0.2 - 0.3] | 1534 | 1118 | 416 | 24% | 399 | FAILED |
| B04: (0.3 - 0.4] | 1065 | 668 | 397 | 35% | 395 | FAILED |
| B05: (0.4 - 0.5] | 674 | 357 | 317 | 45% | 322 | PASSED |
| B06: (0.5 - 0.6] | 653 | 323 | 330 | 55% | 382 | PASSED |
| B07: (0.6 - 0.7] | 784 | 297 | 487 | 66% | 539 | PASSED |
| B08: (0.7 - 0.8] | 1365 | 397 | 968 | 75% | 1056 | PASSED |
| B09: (0.8 - 0.9] | 857 | 142 | 715 | 83% | 731 | PASSED |
| B10: (0.9 - High) | 18 | 0 | 18 | 91% | N/A | N/A |
| | | | | | # Failed | 3 |
| | | | | | # Pool | 9 |
| | | | | | % Failed | 33% |

### Table 21: Random Forest Model Binomial Test Result

| bin | cnt | cntgood | cntbad | midPD | *k** | Binomial Test |
|---|---|---|---|---|---|---|
| B01: (Low - 0.1] | 20448 | 19990 | 458 | 2% | 455 | FAILED |
| B02: (0.1 - 0.2] | 980 | 877 | 103 | 14% | 160 | PASSED |
| B03: (0.2 - 0.3] | 443 | 375 | 68 | 25% | 125 | PASSED |
| B04: (0.3 - 0.4] | 296 | 224 | 72 | 35% | 118 | PASSED |
| B05: (0.4 - 0.5] | 261 | 157 | 104 | 45% | 132 | PASSED |
| B06: (0.5 - 0.6] | 281 | 125 | 156 | 55% | 170 | PASSED |
| B07: (0.6 - 0.7] | 340 | 87 | 253 | 66% | 238 | FAILED |
| B08: (0.7 - 0.8] | 557 | 72 | 485 | 75% | 437 | FAILED |
| B09: (0.8 - 0.9] | 872 | 77 | 795 | 85% | 762 | FAILED |
| B10: (0.9 - High) | 3649 | 150 | 3,499 | 98% | 3577 | PASSED |
| | | | | | # Failed | 4 |
| | | | | | # Pool | 10 |
| | | | | | % Failed | 40% |

### Table 22: Neural Network Model Binomial Test Result

| bin | cnt | cntgood | cntbad | midPD | *k** | Binomial Test |
|---|---|---|---|---|---|---|
| B01: (Low - 0.1] | 8142 | 7684 | 458 | 8% | 731 | PASSED |
| B02: (0.1 - 0.2] | 13294 | 11347 | 1947 | 13% | 1775 | FAILED |
| B03: (0.2 - 0.3] | 1435 | 1032 | 403 | 24% | 374 | FAILED |
| B04: (0.3 - 0.4] | 873 | 562 | 311 | 35% | 330 | PASSED |
| B05: (0.4 - 0.5] | 705 | 379 | 326 | 45% | 341 | PASSED |
| B06: (0.5 - 0.6] | 651 | 281 | 370 | 55% | 380 | PASSED |
| B07: (0.6 - 0.7] | 883 | 348 | 535 | 66% | 604 | PASSED |
| B08: (0.7 - 0.8] | 1241 | 350 | 891 | 75% | 955 | PASSED |
| B09: (0.8 - 0.9] | 902 | 151 | 751 | 84% | 776 | PASSED |
| B10: (0.9 - High) | 1 | 0 | 1 | 90% | N/A | N/A |
| | | | | | # Failed | 2 |
| | | | | | # Pool | 9 |
| | | | | | % Failed | 22% |

### Table 23: Support Vector Machine Model Binomial Test Result

| bin | cnt | cntgood | cntbad | midPD | *k** | Binomial Test |
|---|---|---|---|---|---|---|
| B01: (Low - 0.1] | 1129 | 1042 | 87 | 10% | 126 | PASSED |
| B02: (0.1 - 0.2] | 20641 | 18150 | 2491 | 13% | 2689 | PASSED |
| B03: (0.2 - 0.3] | 814 | 575 | 239 | 23% | 205 | FAILED |
| B04: (0.3 - 0.4] | 664 | 399 | 265 | 38% | 275 | PASSED |
| B05: (0.4 - 0.5] | 2012 | 1030 | 982 | 43% | 911 | FAILED |
| B06: (0.5 - 0.6] | 1031 | 383 | 648 | 55% | 596 | FAILED |
| B07: (0.6 - 0.7] | 822 | 242 | 580 | 67% | 571 | FAILED |
| B08: (0.7 - 0.8] | 321 | 111 | 210 | 74% | 253 | PASSED |
| B09: (0.8 - 0.9] | 475 | 154 | 321 | 86% | 423 | PASSED |
| B10: (0.9 - High) | 218 | 48 | 170 | 91% | 206 | PASSED |
| | | | | | # Failed | 4 |
| | | | | | # Pool | 10 |
| | | | | | % Failed | 40% |

### Table 24: K-Nearest Neighbors Model Binomial Test Result

| bin | cnt | cntgood | cntbad | midPD | *k** | Binomial Test |
|---|---|---|---|---|---|---|
| B01: (Low - 0.1] | 7125 | 6735 | 390 | 9% | 659 | PASSED |
| B02: (0.1 - 0.2] | 14433 | 12324 | 2109 | 13% | 1949 | FAILED |
| B03: (0.2 - 0.3] | 1255 | 875 | 380 | 24% | 326 | FAILED |
| B04: (0.3 - 0.4] | 791 | 501 | 290 | 35% | 301 | PASSED |
| B05: (0.4 - 0.5] | 829 | 452 | 377 | 44% | 393 | PASSED |
| B06: (0.5 - 0.6] | 613 | 288 | 325 | 55% | 358 | PASSED |
| B07: (0.6 - 0.7] | 1145 | 445 | 700 | 66% | 784 | PASSED |
| B08: (0.7 - 0.8] | 1727 | 482 | 1245 | 74% | 1314 | PASSED |
| B09: (0.8 - 0.9] | 209 | 32 | 177 | 81% | 180 | PASSED |
| N/A | N/A | N/A | N/A | N/A | N/A | N/A |
| | | | | | # Failed | 2 |
| | | | | | # Pool | 9 |
| | | | | | % Failed | 22% |

From the tables above, traditional scorecard model approach has the lowest number of failed bins, which means the approach should have the most conservative PD estimate.

# CHAPTER 4

# CONCLUDING REMARKS

In conclusion, the scoring approach which has the highest discriminatory power is gradient boosting, followed by artificial neural network. Decision tree splitting technique was very effective in factor transformation for traditional credit scorecard. The transformation using decision tree algorithm was quick and resulted in good predictive power variable. Performance of the traditional credit scorecard model is also improved by the transformation process. PD model assessment shows that traditional credit scorecard produced the most accurate and conservative PD estimate. To ensure there is no underestimation in capital and provisioning, it is recommended to use the traditional credit scorecard approach which gives lowest % Failed in Binomial Test. Besides, for the ease of credit rating model monitoring and capital provisioning, traditional credit scorecard using logistic regression is much more preferred, as each factor is additive to the final model score, every component can be monitored and adjusted for extra conservatism. In order to improve performance of the application scorecard, it is recommended for banks to use the dual scoring approach, where customers should be accepted when the machine learning scorecards scored the customer above cut-off. PD assessment shows that both the models provide accurate and conservative PD estimate for the low risk customers, that is, B01: (Low - 0.1], which is crucial for underwriting purpose. In the long run, a higher performance application scorecard used for underwriting should benefit the bank due to its better discriminatory power. However, due to the complex design of the algorithm, the stability of the model output should always be closely monitored.

Generally, there is no correct or incorrect decision in choosing a particular tuning parameter for machine learning models. The appropriate tuning parameter could only be found through searching on a tuning grid of potential parameters. Besides, other than the given tuning parameters used on the training process done in this research, there are many more other parameters which could also be tuned for better performance. For example, in the Random Forest algorithm we have used the default number of trees, ntree which is 500 for model training. Further improvement could be made by tuning over this parameter.

In general, a bank's credit underwriting scorecard should have cut-off strategy that has predicted PD lower than the portfolio average PD. This is because higher PD rate would affect profitability as the expected losses will be too high. To ensure the rating system performs accurately for underwriting, the Calibration Plot and Binomial Test could be performed on lower risk buckets only. The tests could be performed on rating grades below cutoff, with more granular PD boundaries. Extremely high PD rating grades could be ignored as those rating grades will not be accepted by the bank. In this paper, we only have dataset with bad rate at 22%, which is not suitable to demonstrate the adjustment towards lower PD boundaries. Nonetheless, such adjustment is possible by just tweaking the lower PD bins to increase granularity.

We also observed limitation in the research dataset, whereby the data is taken only for one snapshot (September 2005), the limited snapshot data could cause model instability due to the effect of seasonality. Some of the model factors such as credit card utilization might have different distribution when other snapshot months are considered. In practice, more data points covering different snapshots are used to construct a robust credit scorecard. Further improvement of the model could be explored through having more snapshot months in the dataset.

Finally, we observed two important highlights for implementing machine learning scorecards. Firstly, it is important to have well architecture database that allows for less time spent on data preparation and more time spent tuning the machine learning model. Secondly, it is computationally expensive to train models such as the gradient boosting, neural network, and support vector machines algorithm. In this research, five cloud machines from DigitalOcean were used, with the spec 32 CPUs, 64gb RAM, CPU optimized, and with parallel training of models, took approximately 31 hours to finish training all the models. While improvement to the models could be done through adding on to the tuning grid to fine tune our models, the computational time will increase according to the length of tuning grids. In practice, banks which are capable to apply Internal-Ratings-based (IRB) approach for capital requirement estimation should have well architecture database and extensive customers recoveries data. Key challenges are to design the database such that processing time to extract data is efficient in merging various source systems databases, and also keeping the data consistencies in check. Also, challenges arise as tuning model in complex algorithms does cost much more computing resources than the traditional approach, and the improvement in model performance should be justifiable to the increase in cost.

## Bibliography

Abdou, H. A., & Pointon, J., April, 2011. Credit Scoring, Statistical Techniques and Evaluation Criteria: A Review of the Literature. *Intelligent Systems in Accounting Finance And Management,* 18(2-3), pp. 59–88.

Awad, M., & Khanna, R., 2015. Machine Learning in Action: Examples. In *Efficient Learning Machines*, pp. 209 - 240. Apress, Berkeley, CA.

Barboza, F., Kimura, H., & Altman, E., 2017. Machine learning models and bankruptcy prediction. *Expert System With Applications*, 83, pp. 405–417.

Basel Committee on Banking Supervision, 2004. Basel II: International Convergence of Capital Measurement and Capital Standards: A Revised Framework. Retrieved from https://www.bis.org/publ/bcbs107.htm

Basel Committee on Banking Supervision, July, 2005. An Explanatory Note on the Basel II IRB Risk Weight Functions. Retrieved from https://www.bis.org/bcbs/irbriskweight.htm

Basel Committee on Banking Supervision, 2005. Working Paper No. 14: Studies on the Validation of Internal Rating Systems. Retrieved from https://www.bis.org/publ/bcbs_wp14.htm

Basel Committee on Banking Supervision, June, 2011. *Basel III: A global regulatory framework for more resilient banks and banking systems.* Retrieved from https://www.bis.org/publ/bcbs189.pdf

Breiman, L., Friedman, J., Stone, C. J., & Olshen, R. A., 1984. *Classification and Regression Trees.* Chapman and Hall, New York.

Chi, B. W., & Hsu, C. C., 2012. A hybrid approach to integrate genetic algorithm into dual scoring model in enhancing the performance of credit scoring model. *Expert Systems with Applications*, 39(3), pp. 2650–2661.

Desai, V. S., Crook, J. N., & Overstreet Jr., G. A., 1996. A comparison of neural networks and linear scoring models in the credit union environment. *European Journal of Operational Research*, 95(1), pp. 24–37.

Gan, L., Wang, H., & Yang, Z., 2020. Machine Learning solutions to challenges in finance: An application to the pricing of financial products. *Technological Forecasting & Social Change*, 153, 119928.

Gunn, S., 1998. *Support Vector Machines for Classification and Regression.* Image Speech & Intelligent Systems Group.

Hand, D. J., & Henley, W. E., 1997. Statistical Classffication Methods in Consumer Credit Scoring: A Review. *Journal of the Royal Statistical Society. Series A*, 160(3), pp. 523–541.

Hayden, E., & Porath, D., 2011. *The Basel II Risk Parameters: Estimation, Validation, Stress Testing - with Applications to Loan Risk Management.* (B. Engelmann, & R. Rauhmeier, Eds.) Springer-Verlag Berlin Heidelberg.

Hothorn, T., Hornik, K., & Zeileis, A., 2006. Unbiased Recursive Partitioning: A Conditional Inference Framework. *Journal of Computational and Graphical Statistics*, 15(3), pp. 651–674.

Khashman, A., 2011. Credit risk evaluation using neural networks: Emotional versus conventional models. *Applied Soft Computing*, 11(8), pp. 5477–5484.

Kuhn, M., & Johnson, K., 2013. *Applied Predictive Modeling.* Springer New York Heidelberg Dordrecht London.

Leo, M., Sharma, S., & Maddulety, K., 2019. Machine Learning in Banking Risk Management: A Literature Review. *Risks*, 7(1), pp. 29.

Li, J. P., Mirza, N., Rahat, B., & Xiong, D., 2020. Machine learning and credit ratings prediction in the age of fourth industrial revolution. *Technological Forecasting & Social Change*, 161, 120309.

Loh, W. Y., & Shih, Y. S., 1997. Split Selection Methods for Classification Trees. *Statistica Sinica*, 7, pp. 815–840.

Louzada, F., Ara, A., & Fernandes, G. B., 2016. Classification methods applied to credit scoring: Systematic review and overall comparison. *Surveys in Operations Research and Management Science*, 21(2), pp. 117–134.

Min, J. H., & Lee, Y. C., 2008. A practical approach to credit scoring. *Expert Systems with Applications*, 35(4), pp.1762–1770.

Öğüt, H., Doğanay, M. M., Ceylan, N. B., & Aktaş, R., 2012. Prediction of bank financial strength ratings: The case of Turkey. *Economic Modelling*, 29(3), pp. 632–640.

Oliver Wyman., 2017. *Next Generation Risk Management.* Retrieved from https://www.oliverwyman.com/content/dam/oliver-wyman/v2/publications/2017/aug/Next_Generation_Risk_Management_Targeting_A-Technology_Dividend.pdf

Oreski, S., Oreski, D., & Oreski, G., 2012. Hybrid system with genetic algorithm and artificial neural networks and its application to retail credit risk assessment. *Expert Systems with Applications*, 39(16), pp. 12605–12617.

Platt, J. C., 2000. Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods.

PricewaterhouseCoopers., 2014. *IFRS 9: Expected credit losses.* Retrieved from https://www.pwc.com/gx/en/audit-services/ifrs/publications/ifrs-9/ifrs-in-depth-expected-credit-losses.pdf

Sharma, D., 2009. *Guide to Credit Scoring in R.* Retrieved from https://cran.r-project.org/doc/contrib/Sharma-CreditScoring.pdf

Siddiqi, N., 2005. *Credit Risk Scorecards: Developing and Implementing Intelligent Credit Scoring.* John Wiley & Sons, Inc., Hoboken, New Jersey.

Sun, M. Y., & Wang, S. F., 2005. Validation of Credit Rating Models - A Preliminary Look at Methodology and Literature Review. *Review of Financial Risk Management*, pp. 1–15.

Tang, L., Cai, F., & Ouyang, Y., 2019. Applying a nonparametric random forest algorithm to assess the credit risk of the energy industry in China. *Technological Forecasting & Social Change*, 144, pp. 563–572.

Thomas, L. C., 2009. *Consumer Credit Models: Pricing, Profit and Portfolios.* Oxford University Press.

Wang, G., Ma, J., Huang, L., & Xu, K., 2012. Two credit scoring models based on dual strategy ensemble trees. *Knowledge-Based Systems*, 26, pp. 61–68.

Wang, X., Zeng, D., Dai, H., & Zhu, Y., 2020. Making the right business decision: Forecasting the binary NPD strategy in Chinese automotive industry with machine learning methods. *Technological Forecasting & Social Change*, 155, pp. 120032.

West, D., 2000. Neural network credit scoring models. *Computers & Operations Research*, 27(11-12), pp. 1131–1152.

Xu, M., David, J. M., & Kim, S. H., 2018. The Fourth Industrial Revolution: Opportunities and Challenges. *International Journal of Financial Research*, 9, 2.

Yap, B. W., Ong, S. H., & Nor Huselina, M. H., 2011. Using data mining to improve assessment of credit worthiness via credit scoring models. *Expert System with Applications*, 38(10), pp. 13274-13283.

Yeh, I. C., & Lien, C. H., 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), pp. 2473-2480.

Yu, L., Li, X., Tang, L., Zhang, Z., & Kou, G., 2015. Social credit: a comprehensive literature review. *Financial Innovation*, 1, 6.

Zhao, Z., Xu, S., Kang, B. H., Mir Md, J. K., Liu, Y., & Wasinger, R., 2014. Investigation and improvement of multi-layer perceptron neural networks for credit scoring. *Expert Systems with Applications*, 42(7), pp. 3508-3516.