ZHANG_JING_02083604

by Jing Zhang

Submission date: 05-Sep-2022 05:00PM (UTC+0100)
Submission ID: 185614588
File name: ZHANG_JING_02083604.pdf (7.79M)
Word count: 16333
Character count: 85096

Imperial College London

Imperial College London

Department of Mathematics

Machine Learning in Credit Risk

Author: Jing Zhang (CID: 02083604)

A thesis submitted for the degree of

MSc in Mathematics and Finance, 2021-2022

Declaration

The work contained in this thesis is my own work unless otherwise stated.

Acknowledgements

Throughout the time leading to the completion of this thesis, I have received tremendous support from the Mazars Quantitative Teams. I would like to express my sincere gratitude to my manager Mariem Bouchaala for her invaluable advice, continuous support and patience, who supervised the project all over the summer. I would also like to thank Didier Verchiere, who provided me with new ideas, gave me frequent feedback, and shared his expertise on the advancement of the project over the entire summer; Hamza Saber, who guided me through some model structures; Andiren Palayret and Joseph Burrin, who helped with my data collection; and Xavier Larrieu who offered essential guidance through the term. My gratitude also extends to all other quantitative team members who provided me with treasured support and made me feel welcomed in the team. Besides, my sincere gratitude goes to Professor Brigo Damiano, my thesis supervisor from Imperial College London. This thesis would have never been accomplished without his assistance and dedicated involvement in every step throughout the process. Finally, I must express my very profound gratitude to my parents for providing me with unfailing support and tremendous encouragement throughout my life.

Abstract

Credit risk modelling is a field with access to a large amount of diverse data where machine learning methods, a powerful tool in computing, can be deployed to add great analytical value. In this paper, we assessed the benchmark algorithm Logistic Regression and different machine learning algorithms, including Decision Tree, Random Forest, Extreme Gradient Boosting (XGBoost), K-nearest Neighbourhood (KNN), Support Vector Machine (SVM), and Artificial Neural Network (ANN) on the performance of Probability of Default (PD) prediction for home loans. The overall best model is ANN, reaching an AUC score of 0.7863, followed by SVM (AUC = 0.775) and Random Forest (AUC = 0.7737). The benchmark model logistic regression also has a high statistic value with an AUC score of 0.7641. These results indicate that machine learning techniques can indeed improve the performance of PD prediction, but meanwhile, the traditional model is proved to be a reliable method, and it explains why logistic regression has been applied extensively in previous literature. Moreover, this study compares the model performance trained with several features selected through feature selection as well as trained with all the features. The models achieve similar results using the two different training sets, while the former has a lower computational cost. It indicates that feature selection is important when having a high-dimensional dataset, because it saves computational power while incorporating the most useful information from the dataset.

Contents

List of Figures

List of Tables

Introduction

A definition of credit risk is provided by the Basel Committee on Banking Supervision:" A counterparty fails to meet its obligations according to the agreed terms(BCBS, 2000). A classic example is a borrower's failure to repay a loan. Beyond that, credit risk is also a significant risk factor to be considered in government lending, corporate debt instruments, retail credit products, and all financial transactions. In the past 20 years, a series of financial crises have manifested the threat of credit risk and the significance of managing it, such as the Argentine default (2001), the subprime mortgage crisis triggering Lehman Brothers' collapse (2007-2010), the European sovereign debt crisis (2010-2012) and the bankruptcies of small and medium-sized companies following the recent pandemic (2020) [1]

The regulators have taken a series of actions to cope with these challenges. New absolute measures of credit risk, Probability of Default (PD), Loss Given Default (LGD), and Exposure at Default (EAD) were introduced in Basel II by the Basel Committee along with the new internal rating based approaches (IRB) in 2006. But soon, the global financial crisis in 2007 and 2008 challenged the three main credit risk components. The regulators were forced to pivot and consider the systemic conditions and overall portfolio default rates and introduced stress tests. After IAS 39 which performed an easier modelling task of incurred loss, IFRS 9 was introduced as an expected loss approach to model the loan staging that might respond in a crisis. [2]

In the first chapter of this paper, we define the credit risk, demonstrate the motivations for studying it in the current economic environment, and assess the credit risk measurement under the IFRS 9 regulation and the IRB regulation. In the second chapter, we present a literature review of the studies conducted on credit risk using machine learning. Then, the third chapter of this paper describes the theoretical implications of machine learning models, feature selection methods, and performance measures. In the fourth chapter, we describe the data source and the data preprocessing process for conducting a numerical studies on PD prediction. In the fifth chapter, we evaluate the models and discuss the results. Finally, the last chapter concludes our studies, points out the main contributions of our work, states the limitations, and proposes some future work that can be conducted in further studies.

Chapter 1

Credit Risk and the Background

1.1 The Importance of Credit Risk Modelling

1.1.1 The Definition of Credit Risk

A definition of credit risk is provided by the Basel Committee on Banking Supervision:" A counterparty fails to meet its obligations according to the agreed terms(BCBS, 2000)." Therefore, credit institutions that offer loans to their clients have to take into account the credit risk when the interest rate on a loan is calculated. The subprimes crisis has shown that the implemented regulation failed in measuring credit risk, because many defaults that occurred were from a mismanagement of the accounting of financial instruments and a majority of those were related to counterparties that were supposed to be safe and performing. To measure the credit risk, several important risk measures were introduced, including Probability of Default (PD), Loss Given Default (LGD), Exposure at Default (EAD), and Significant Increase in Credit Risk (SICR) by the regulating institutions.

1.1.2 The Regulation of Credit Risk after the Financial Crisis

The global financial crisis of 2007 and 2008 showed that Basel II underestimated the risks involved in current banking practices and that the financial system was overleveraged and undercapitalized, despite Basel II's requirements. Therefore, Basel III was introduced with a set of reforms designed to mitigate risk within the international banking sector by requiring banks to maintain specific leverage ratios and keep certain levels of reserve capital on hand. Meanwhile, the politician noticed that the financial crises do not only concern the financial markets but also have a significant impact on the global economic situation. During the G20 of 2009, the international leaders have pushed the international financial institutions to develop new regulations incorporating the current economic and financial situations. The resulting new financial instrument accounting is the IFRS9 norm developed by the International Accounting Standards Board (IASB). [3]

1.2 The Regulatory Standards for Credit Risk Measure

1.2.1 Internal Rating-Based Model (IRB)

The IRB approach was introduced by Basel Committee on Banking Supervision. The main idea is that under the Basel II guidelines, banks are allowed to use their own estimated risk parameters for the purpose of calculating regulatory capital, subject to supervisory approval. Also, only banks meeting certain minimum conditions, disclosure requirements and approval from their national supervisor are allowed to use this approach in estimating capital for various exposures. There are two broad approaches that a bank can follow [4] :

- Foundational IRB approach: banks calculate their own PD parameter while the other risk parameters are provided by the bank's national supervisor
- Advanced IRB approach: banks calculate their own risk parameters, PD, EAD, and LGD, subject to meeting some minimum guidelines

In Basel II, the three pillars of sound regulation are described, the first of which is minimum capital requirements (Basel, 2005). The banks may choose for themselves which method to follow to calculate minimum capital requirements, so-called regulatory capital. In the standardized approach, a fixed percentage of outstanding loans is set aside. This percentage varies for different asset classes. It may be seen as the most straightforward and most primitive approach, but it may be very expensive for the banks in that they hold much capital when it is not needed, and the opposite. In the Internal Ratings Based (IRB) approach, the bank chooses the percentage of total exposure in each asset class to set aside. Expected and unexpected losses are to be calculated, where the second is of much greater importance. Indeed, regulatory capital is only concerned with unexpected losses. As part of the methodology to calculate unexpected losses, PD models are built. The implication is that when reading this thesis, one should note that the models described are not used by the firm to score customers prior to issuing a loan. The models are used by the firm to set aside enough regulatory capital.

1.2.2 International Financial Reporting Standard 9 (IFRS 9) Model

IFRS 9 is an International Financial Reporting Standard (IFRS) published by the International Accounting Standards Board (IASB). It contains three main topics: classification and measurement of financial instruments, impairment of financial assets and hedge accounting. The complete version of the norm IFRS 9 was published in July 2014, and it came into force on 1st January 2018, replacing the earlier IFRS for financial instruments IAS 39. [5] The main changes regard the classification and the measurement of financial assets and do not affect financial liabilities. Based on the official document of the IASB, we summarize the main points from IFRS 9 that are related to the work:

- Stage 1 comprises instruments that have not suffered any significant deterioration of credit quality since initial recognition (performing assets). To measure the expected loss (EL), banks have to calculate the 12-month expected credit loss (ECL).
- Stage 2 comprises instruments that have suffered significant deterioration since initial recognition (under-performing assets). This is the main change of the new IFRS 9 norm. The corresponding EL is a Lifetime ECL which takes into account a deterioration in the credit quality of the asset. The Incurred But Not Reported (IBNR) assets are subject to " a significant increase in credit risk since initial recognition," according to the new norm. However, no actual information gives a material indication about this increase.
- Stage 3 concerns all instruments where a default has occurred (Non-performing assets). In this case, the lifetime ECL is the measurement that has to be implemented.

In all stages, to compute the 12-month or lifetime ECL, PD has to be modelled. The standard requires the modelling to include "historical information as well as Forward-Looking macroeconomic information."

1.2.3 The Comparison between IRB and IFRS 9

IRB is the regulatory capital based on risk-weighted assets and leverage ratios. It sets the limit on the total size of the business for a bank. The regulatory capital covers both unexpected losses and expected losses which have been recognized by loan loss provisioning. On the other hand, IFRS 9 is a provision which is based on the expected loss, and so the purpose is to absorb expected losses, required to be the best estimate of loss that is not conservative.

The table summarizes the major differences between the two approaches: [6]

Aspect	Internal Ratings-Based Model	IFRS 9 Model
Default Definition	Specific definition based on a combination of days past due and unlikely to pay.	Consistent with Credit Risk Management practice plus rebuttable presumption that default does not occur later than 90 days past due.
Lifetime v.s. 12-month Horizon	Credit Rating System and associated PDs are based on a 12-month horizon	Stage 1 Assets allowances are based on a 12-month horizon. Stage 2 and stage 3 allowances are based on lifetime expected losses.
Point-in-time (PIT) vs. Through-the-cycle (TTC)	Models are generally developed using a hybrid approach (considering both cyclical and non-cyclical variables) which determines the ratings, which are then calibrated to a PD which may be somewhere between PIT and TTC.	Expected losses should reflect current conditions. This may require a PIT adjustment over historically based estimates.
Quantitative Floors	The regulatory PD has a floor at 0.03% for all exposures except sovereign counterparties.	No floor on the PD.
LGD Estimates	Conservative estimate (Downturn LGD).	Unbiased, PIT estimate.
Frequency of Estimates	Annual	Continuous basis (at least, every time Financial Statements are prepared).
Auditing of Figures	Bank supervisors.	Auditors and market supervisors.

Table 1.1: Differences between IRB Model and IFRS9 Model

1.3 Non-performing Loans

1.3.1 Definition of Non-performing Loans

A non-performing loan (NPL) is defined as a loan that is in default due to the inability of the borrower to make the scheduled payments for a specified period. The specified period varies depending on the industry and the type of loan. In general, the period is 90 days or 180 days. In order to evaluate risk exposures, several international financial authorities offer specific guidelines for determining non-performing loans. For example, the European Central Bank (ECB) specifies multiple criteria for defining an NPL, that loans are non-performing if they are [7] :

- 90 days past due, without the borrower paying the agreed installments or interest
- impaired with respect to the accounting specifics for U.S. GAAP and IFRS banks
- in default according to the Capital Requirements Regulation

In this study, the first criterion is adopted that when the payment due date exceeds 90 days, the loan is defined as default.

1.3.2 Definition of Probability of Default

The Probability of Default (PD) is the likelihood that a borrower will fail to pay back a debt. It is estimated using historical data and statistical techniques. For individuals, PD can be reflected in a Fair Isaac Corporation (FICO) score, while the PD of a business tends to be reflected in credit ratings. Evaluating PD has important implications. For example, when the estimated PD is higher, a higher interest rate needs to be charged from the borrower to compensate for the risk the lender takes. Besides, calculating PD is necessary for default prediction. When the PD exceeds a certain threshold, then the loan is labelled as default. In this study, the optimal threshold is calculated for each model.

Chapter 2

Literature Review of Credit Risk Modelling

2.1 Probability Default and Recovery Rate Estimation and Prediction

Since the financial crisis, NPLs have been the focus of European regulators for years, as many banks still face difficulties in disposing of those materialized on their balance sheets during the crisis. [1] In order to predict NPLs in advance, many studies have been focused on PD prediction. According to Guidance to Banks on Non-performing Loans from ECB [8] , once NPLs have occurred, the regulators tend to recommend banks to pool the NPLs and sell them to specialized investors, such as debt collection agencies. One of the most important variables governing the price of NPLs portfolios is the recovery rate, that is the percentage of exposure that can be recovered from each borrower through the debt collection process. The relationship between PD and RR have been investigated for decades. Table 2.1 summarized by Altman, Resti, and Sironi [9] are presented below.

In this thesis, the reduced-form models are assumed that RR is independent of PD. Also, PD is the main focus in the analysis and RR will not be further discussed due to data availability.

Model Type	Related Studies	Model Explanation	Relationship between RR and PD
First generation structural-form models	Merton (1974), Black and Cox (1976), Geske (1977), Vasicek (1984), Crouhy and Galai (1994), Mason and Rosenfeld (1984).	PD and RR are a function of the structural characteristics of the firm. RR is therefore an endogenous variable.	PD and RR are inversely related
Second generation structural-form models	Kim, Ramaswamy e Sundaresan (1993), Nielsen, Saà-Requejo, Santa Clara (1993), Hull and White (1995), Longstaff and Schwartz (1995).	RR is exogenous and independent from the firm's asset value.	RR is generally defined as a fixed ratio of the outstanding debt value and is therefore independent from PD.
Reduced-form models	Litterman and Iben (1991), Madan and Unal (1995), Jarrow and Turnbull (1995), Jarrow, Lando and Turnbull (1997), Lando (1998), Duffie and Singleton (1999), Duffie (1998) and Duffee (1999).	Reduced-form models assume an exogenous RR that is either a constant or a stochastic variable independent from PD.	Reduced-form models introduce separate assumptions on the dynamic of PD and RR, which are modeled independently from the structural features of the firm.
Latest contributions on the PD-RR relationship	Frye (2000a and 2000b), Jarrow (2001), Carey and Gordy (2003), Altman, Brady, Resti and Sironi (2001 and 2004).	Both PD and RR are stochastic variables which depend on a common systematic risk factor (the state of the economy).	PD and RR are negatively correlated. In the "macroeconomic approach" this derives from the common dependence on one single systematic factor. In the "microeconomic approach" it derives from the supply and demand of defaulted securities.

Table 2.1: The Treatment of PD and RR within Different Credit Risk Models

2.2 The Evolution of Modelling Techniques

One of the earliest models is the linear discriminant analysis (LDA) model, and Wiginton (1980) first applied a logistic regression (LR) model to evaluate credit risk and found that this model had high classification accuracy and strong practicability. Subsequently, logistic regression analyses became a standard method used to evaluate credit risk in many studies such as Van Gestel, Tony, Bart, Garcia, Dijcke (2003), Avery, et al. (2004), and Huang, Chen, Wang (2007). [10]

These models however, suffer from their apparent inability to capture non-linear dynamics, which are prevalent in financial ratio data pointed out by Petr and Gurný (2013). [11] Therefore, machine learning methods are considered to overcome the limitations of standard models such as logit and probit models, as they can detect non-linear interactions between input variables presented in the studies such as Van Gestel et al. (2003), Khandani et al. (2010), Crook, Edelman, Thomas (2007), Brown Mues (2012), and Kruppa et al. (2013). [10]

However, the researchers did not reach a universal conclusion on which machine learning method is decisively better than the others, while it largely depends on the problem and the datasets. For example, Huang et al. (2004) studied corporate credit rating models, suggesting support vector machines (SVM) achieves better explanatory power compared to the benchmark artificial neural network (ANN). [11] On the other hand, Yeh and Lien (2009) applied different methods such as k-nearest neighbors (KNN), logistic regression, discriminant analysis, Naive Bayes, ANN and classification trees on a data set of customers' credit default in Taiwan and found ANN outperforms

other techniques for obtaining reliable estimates of default probability. [12] Moreover, Arora and Kaur (2020) showed that Random Forest (RF) is superior to SVM and KNN in predicting the probability of borrower default. [10] Similarly, Anastasios et al. (2018) compared Extreme Gradient Boosting (XGBoost) against other machine learning methods and concluded it achieves better performance in default classification.

Motivated by all the aforementioned research endeavours, we revisit the issue of credit risk modelling. Among all these techniques, we choose LR, SVM, KNN, RF, ANN, and XGBoost for our modelling tasks, which have been proven to be computationally fast, ease to implement and show good performance.

2.3 Approaches for Feature Selection and Extraction

With the rapid increase in data dimensionality, selecting necessary information from the data to process becomes crucial, as it can reduce the overfitting risk, mitigate the curse of dimensionality, increase computational efficiency, and decrease memory requirements, as proven in the studies by Khadlid et al. (2014) and Zhaowen Wang, et al. (2016). The feature selection is to choose features from the original datasets based on the feature importance. It selects the top ones by ranking their importance and disregards the rest. On the other hand, feature extraction creates a whole new set of features on top of the original ones, mapping higher dimensional space into a lower-dimensional coordinate. [13]

One of the most popular approaches for feature extraction is Principal Component Analysis (PCA), invented by Pearson (1901) and Hotelling (1933). The main idea is to project the original high-dimensional feature space in the direction of the greatest variance according to coordinate ranking, so that the most crucial information is extracted while the data size is compressed. Since then, some variants of PCA have been developed. For example, Sd' Aspremont et al. proposed sparse principal component analysis (SPCA) that modifies components with zero loadings to improve the sparsity of the financial asset trading data. Another popular feature extraction method is independent component analysis (ICA). It undertakes a linear transfer of the original feature space to a new coordinate system, making the components of the new feature space mutually independent. Hyvarinen A. et al. (2000) concluded that ICA could efficiently capture the essential structure of non-Gaussian data in many applications. Moreover, discriminant analysis (DA) is known for the ability to find boundaries around clusters of classes and projects data points into a new feature space to maximize class separability. The study by Yang J. (2003) pointed out that the combination of PCA and DA achieves better results than employing PCA or DA alone. [13]

There are three main methods for feature selection: filtered methods, wrapper methods, and embedded methods. Filtered methods select variables regardless of the model. They are based only on general features like the correlation with the variable to predict. Filter methods suppress the least interesting variables and the other variables will be used for classification or regression models. Phuong (2005) suggests these methods are particularly effective in computation time and robust to overfitting. Wrapper methods evaluate subsets of variables which allows, unlike filter approaches, to detect the possible interactions amongst variables, such as Recursive Feature Elimination used by Zongrui Dai (2021) studying bank credit rating prediction. [14] Embedded methods have been recently proposed that try to combine the advantages of both previous methods. A learning algorithm takes advantage of its own variable selection process and performs feature selection and classification simultaneously, such as Random Forest used in LGD estimation by Elina Velka (2020).

2.4 Techniques for Imbalanced Datasets

Apart from selecting algorithms, a notable challenge for credit risk modelling is the imbalanced datasets. In general, the loan is granted after a careful evaluation of the client, so the probability of default is usually low. This phenomenon can be reflected in the datasets by comparing the ratio of the target labels. The number of "No default" labels largely exceeds the number of "default" labels. Thus, some researches regarding such imbalanced datasets have been conducted. Brown

and Mues (2012) applied different classifiers to credit scoring data, and they found that ensemble methods such as XGBoost and RF achieve relatively good performances when encountering severe class imbalances. They also discovered the traditional methods such as LDA and LR can compete with the more complex methods, but SVM performs poorly with imbalanced data. [12]

Common techniques to handle imbalanced data include sampling the training datasets, generating synthetic data, and cost-sensitive training. Yap Bee Wah et al. (2016) applied oversampling and undersampling to the imbalanced datasets with the majority class occupying 95.8% while the minority class only consists of 4.2%. The idea of oversampling is to duplicate minority class observations until the number of observations balances the majority class. In contrast, downsampling is to reduce the majority class observation to balance the minority class. The authors concluded that the models perform significantly better on the resampled datasets. [15]

Anna Stelzer (2019) adopted cost-sensitive learning methods. It makes use of a cost matrix which contains a class misjudged penalizing coefficient in order to raise the misjudgement cost weight of the defaulted samples. In this case, the loss is greater when misclassifying the minority class. So it becomes an optimization problem to minimize the cost of misclassification. The usefulness of this technique is also supported by the study of Hand and Vinviotti (2003). [12]

Choosing proper evaluation metrics is another important decision for imbalanced datasets. In traditional classification problems, accuracy is used, which is simply the percentage of correct predictions. However, in the presence of imbalance, the algorithm can achieve high accuracy by simply predicting every sample to be the majority class, which is not meaningful. Therefore, it is essential to use metrics that can provide insights into the model performance. Lessman et al. (2015) used three metrics: accuracy, Area under Receiver Operating Characteristics (ROC) Curve (AUC), and Brier Score to evaluate the model performance. [12] They found that the advanced models achieve a better score on AUC and Brier Score compared to the traditional linear models, while they do not exhibit significant differences when comparing accuracy scores. Thus, AUC and Brier Score provide a more informative evaluation in terms of model selection.

2.5 The Benefits and Challenges of Machine Learning Approaches

In this era, financial databases are increasing rapidly with large datasets, which require more robust and efficient data mining processes and financial statistical modelling to support more informed decision-making. Conventional econometric methods fail to efficiently capture the information in the full spectrum of the datasets, as big financial datasets are usually characterized by increased noise, heavy-tailed distributions, nonlinear patterns and temporal dependencies, which pose significant statistical challenges, since conventional statistical methods are based on the assumption that observations follow a normal curve. [10]

Machine learning techniques are known for the superior capability of handling large datasets, which can review large volumes of data and discover specific trends and patterns that would not be apparent to humans. They can also be used to reduce dimensionality and increase accuracy in predicting the future behaviour of corporate loans to facilitate a more effective micro and macro supervision of credit risk for financial institutions. Also, machine learning is algorithm-based and requires minimum human intervention, so it greatly saves labour resources and makes the outputs more objective.

Despite the numerous advantages, some challenges still exist. First, machine learning algorithms require massive datasets to train on, but due to confidentiality, it may become difficult to receive enough training data to produce an unbiased model. Also, machine learning algorithms are often criticized for the lack of transparency which makes them controversial, particularly in the financial industry, because regulators have strict requirements concerning the interpretability of the credit risk models. This creates a burden on the financial institutions to provide evidence of model interpretability. To overcome this, it can be considered to use some machine learning models with high interpretability such as tree models. [10]

Another challenge is the tendency to overfit. This often occurs when the model captures excessive information in the data, making it performs exceptionally well on the training data while under-performs on other datasets. Hopefully, many approaches to deal with overfitting have been developed, such as data splitting, early-stopping, drop-out and so on. Overall, the advantages of machine learning models outweigh these drawbacks and make them promising in credit risk modelling.

Chapter 3

Methods

3.1 Mathematical Formulation

The prediction problem of whether a loan will default is modelled using a binary variable Y as

$$Y = \begin{cases} 1, \text{ default} \\ 0, \text{ no default} \end{cases}$$

and a feature vector $X = (X_1, \ldots, X_m)$ where m is the number of features plus a error term ε to capture idiosyncratic error in the data.

So the mathematical formula can be written as:

$$Y = f(X) + \varepsilon \tag{3.1.1}$$

The function $f\ (X)$ represents the default probability of the loan.

$$f(X) = E(Y \mid X = x) = \Pr(Y = 1 \mid X = x) \tag{3.1.2}$$

The goal is then to find a good estimate of $f(X)$, denoted by $\widehat{f}(X)$. For any $\widehat{f}(X)$, the loss $L\{f(x), \widehat{f}(X)\}$ is computed. In this study, cross-entropy is used as it is a standard loss measurement for the binary classification problems, written as

$$L = -(y \ln p + (1 - y) \ln(1 - p)) \tag{3.1.3}$$

where y is the true label and p is the predicted probability. Thus, the learning problem becomes an optimizing problem which is to minimize $L\{f(x), \widehat{f}(X)\}$ by finding the optimal $\widehat{f}(X)$ denoted as $\widehat{f^{\star}}(X)$. One thing to be noted is that when using machine learning, $f(X)$ is only a mathematical expression, but may not have an exact formula to give rise to the expression, because machine learning models do not make assumptions about the model structure and parameters. This is a root difference between traditional mathematical models and machine learning models. Thus, when predicting default probabilities, the study will focus on evaluations of different metrics rather than giving detailed explanations of the process that happened inside the models.

3.2 Statistical and Machine Learning Models

3.2.1 Logistic Regression Model

Logistic regression is commonly used in many studies for classification problems and is the only statistical model used in this study to serve as a benchmark for the machine learning models. It uses a logit-link function to estimate the relationship between features and labels:

$$\text{logit}\,(p_i) = \log\left(\frac{p_i}{1 - p_i}\right) = \beta_0 + \beta_1 X_1 + \ldots + \beta_m X_m \tag{3.2.1}$$

$$p_i = p(y_i = 1 \mid x) = \frac{1}{1 + \exp\left(-\left(\beta_0 + \sum_{j=1}^{m} \beta_j x_j\right)\right)} \tag{3.2.2}$$

where p_i is the probability of default, β are the regression coefficients, and m is the total number of features.

From Figure 3.1 [16], it can be seen that the output is bounded between 0 and 1 which satisfies the range of the probability. Here, $\phi(z)$ is equivalent to p_i and z is equivalent to $\beta_0 + \sum_{j=1}^{m} \beta_j x_j$.

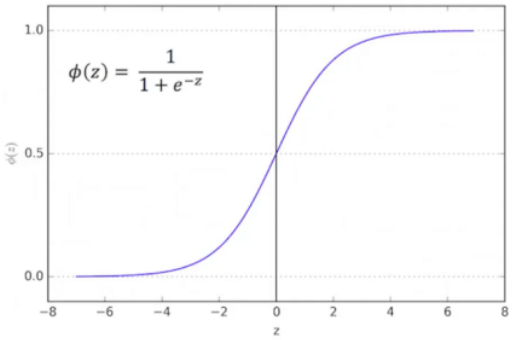

Figure 3.1: Logistic Regression

This can be derived by considering the equivalent formulation of a latent variable model. For each sample i, there is a continuous latent variable y_i^* that is distributed as

$$y_i^* = \beta_0 + \boldsymbol{\beta} \cdot \mathbf{X_i} + \epsilon_i \tag{3.2.3}$$

where $\epsilon_i \sim$ Logistic(0,1) that the random error variable has a standard logistic distribution. Then y_i can be viewed as an indicator for whether this latent variable is positive:

$$y_i = \begin{cases} 1 & \text{if } y_i^* > 0 \text{ i.e. } -\varepsilon_i < \beta_0 + \boldsymbol{\beta} \cdot \mathbf{X}_i \\ 0 & \text{otherwise} \end{cases} \tag{3.2.4}$$

Then using the fact that the cumulative distribution function of the standard logistic distribution is the inverse of the logit function,

$$\Pr(\varepsilon_i < x) = \text{logit}^{-1}(x) \tag{3.2.5}$$

so the following derivation is obtained:

$$\begin{aligned} \Pr(y_i = 1 \mid \mathbf{X}_i) &= \Pr(y_i^* > 0 \mid \mathbf{X}_i) \\ &= \Pr(\beta_0 + \boldsymbol{\beta} \cdot \mathbf{X}_i + \varepsilon_i > 0) \\ &= \Pr(\varepsilon_i > -\beta_0 - \boldsymbol{\beta} \cdot \mathbf{X}_i) \\ &= \Pr(\varepsilon_i < \beta_0 + \boldsymbol{\beta} \cdot \mathbf{X}_i) \\ &= \text{logit}^{-1}(\beta_0 + \boldsymbol{\beta} \cdot \mathbf{X}_i) \\ &= p_i \end{aligned} \tag{3.2.6}$$

Unlike Linear Regression, the coefficients of Logistic Regression cannot be derived by an analytical approach. Instead, it starts with random parameters and uses an iterative process to maximize the likelihood function. It determines the model parameters β_0 and $\boldsymbol{\beta} = \beta_1, \ldots \beta_k$ through minimizing the negative log-likelihood function:

$$\min_{(\beta_0, \boldsymbol{\beta})} l = -\sum_{i=1}^{n} (y_i \ln(p_i) + (1 - y_i) \ln(1 - p_i)) \tag{3.2.7}$$

For modelling PD, Logistic Regression does not require the researcher to decide on hyperparameters before estimating the model. Thus, it is easy to implement and reproduce. [17]

3.2.2 Decision Trees

Decision Trees (DT) is a non-parametric supervised learning method that can perform both classification and regression. The mechanism behind tree-based algorithm is simpler than other ML methods. DT split the data into partitions with operations at each branch. Each branch consists of a root node at the top and two child nodes underneath. The inputs of DT include the width and the depth. The width determines the number of leaves at the bottom of the tree which are mostly pure and the depth is the number of levels of the tree.

There are several splitting rules that can be applied to split the dataset. A common approach is to measure information gain (IG) of the split, formulated as

$$IG(parent, children(left, right)) = E(parent) - [w(left) \times E(left) + w(right) \times E(right)] \quad (3.2.8)$$

where w is the weight of the sample in a split and E represents the entropy calculated as

$$E = -\sum_{i}^{N} p_i \log_2 p_i \quad (3.2.9)$$

where p_i is the probability of randomly picking an element of class i. In this study, $N = 2$ as there are only two classes: default and no default. As can be seen, IG measures the difference between the entropy of the parent and the weighted sum of the entropy of the children, the higher the IG, the better the split.

An alternative measure is the Gini impurity index, ranging from 0 to 0.5, and is written as:

$$\text{Gini}(t) = 1 - \sum_{i=1}^{N} p(i \mid t)^2 \quad (3.2.10)$$

where N represents the number of classes in the label and the p represents the ratio of classes at the i^{th} node. Same as before, in this study, $N = 2$. In this study, it can be expressed as

$$\text{Gini}(X) = 1 - [p(y = 0 \mid X)^2 + p(y = 1 \mid X)^2] \quad (3.2.11)$$

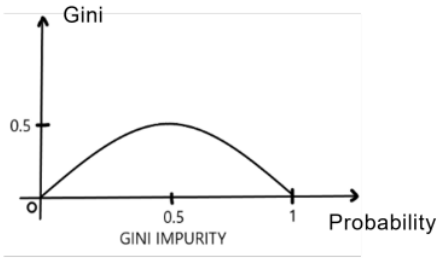

Figure 3.2: The Gini Index

Then the decision tree undergoes a pruning process to establish individual decision trees. The lower the Gini impurity is, the purer so the better the split is. Finally, it outputs a decision $y = 0$ or 1 based on the final split result.

Figure 3.3 gives a simple example of the pruning process:

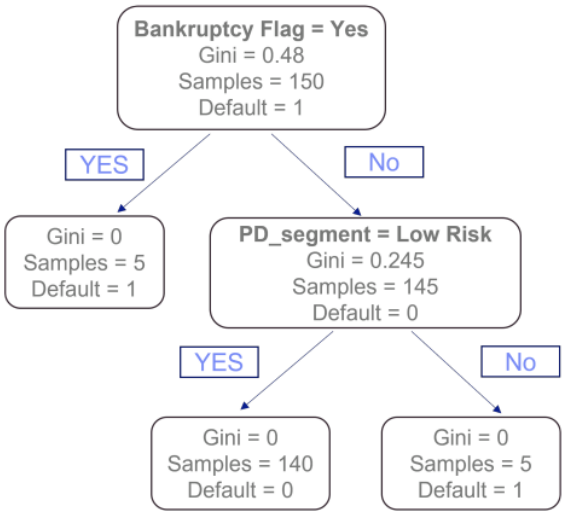

Figure 3.3: A simple example of a decision tree

The DT algorithm has the advantage that the model is intuitive and easy to interpret, but it has the limitation that the over-fitting problem is likely to occur in the process of dividing the feature space or producing branches. The prediction accuracy is reduced as a result.

3.2.3 Random Forest

Random Forest (RF) is an extended version of decision trees by utilizing an ensemble learning technique. It is composed of multiple decision trees, and there is no correlation among different decision trees. Each decision tree DT_i makes its own classification independently, and the results of the random forest are given by the votes of multiple decision trees (e.g. n trees), formulated as

$$f(x) = \underset{\theta}{\operatorname{argmax}} \sum_{i=1}^{n} I\left(DT_i(\theta) = y\right) \tag{3.2.12}$$

In a random forest model, the features of X are randomly subsetted and classification trees are drawn for every sample. The algorithm can be seen as an extension of the bagging methodology since it takes advantage of bagging as well as feature randomness to construct an uncorrelated ensemble of decision trees.

The structure of a random forest is shown in Figure 3.4 [18].

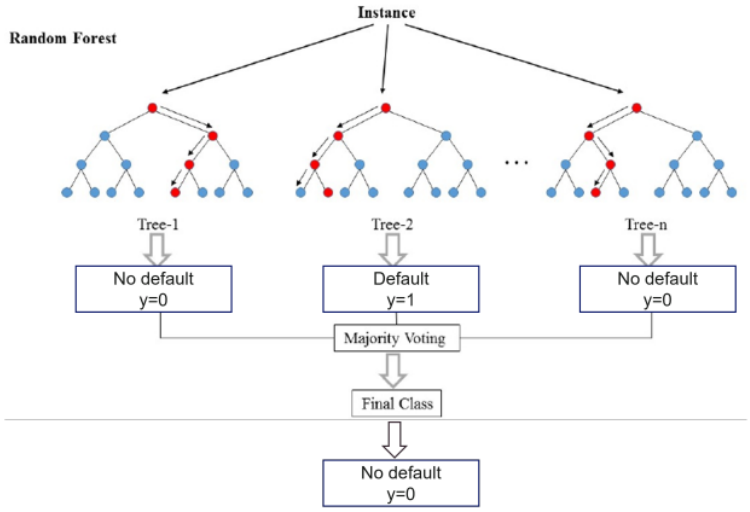

Figure 3.4: Random Forest

The "randomness" of Random Forest comes from the feature selection process, because not all features are used during the training process, but the variables are chosen in each node by random instead of using discriminatory power. This "randomness" feature reduces the calculation cost while performing better. [10]

The random forest algorithm overcomes the main challenge faced by a single decision tree, which reduces the risk of overfitting by increasing the number of trees in a forest, making it more accurate in generalization. The algorithm is very stable that a new data point introduced in the dataset would not impact the overall model much. One drawback of random forest models is the complexity. Since it is more complex than a single decision tree, it requires more datasets and takes more time to train than other comparable models.

3.2.4 Extreme Gradient Boosting

Extreme Gradient Boosting (XGBoost) is another ensemble decision tree algorithm based on the gradient boosting methodology. Gradient boosting seeks to approximate a function of weights on weaker classifiers to minimize the loss function. The algorithm starts with arbitrary weights and trains the model sequentially. For example, a decision tree is a typical weak classifier. Each decision tree is created using a greedy search procedure to select split points that best minimize an objective function. This can result in trees that use the same attributes and even the same split points again and again. New decision trees are added to the model to correct the residual error of the existing model. On the top of the classic gradient boosting tree, extreme gradient boosting adds a penalty term to the cost function. The objective function is formulated as

$$L(\phi) = \sum_i l\left(\hat{y}_i, y_i\right) + \sum_t \Omega\left(f_t\right) \tag{3.2.13}$$

where the regularization term $\Omega(f) = \gamma T + \frac{1}{2}\lambda\|w\|^2$ The term w adds a penalty to the model complexity while the other regularization parameters smooth the final trained weights to avoid overfitting.

Let $y_i^{(t)} = \hat{y}_i^{(t-1)} + f_t$ be the i^{th} predicted instance at time t. The additive training process can be formulated as

$$L^{(t)} = \sum_{i=1}^{n} l\left(y_i, \hat{y}_i^{(t-1)} + f_t\left(x_i\right)\right) + \Omega\left(f_t\right) \tag{3.2.14}$$

where n is the total number of trees.

The f_t obtained from the new decision tree is greedily added with the function in order to optimize the objective in Equation 3.2.14. Finally, the objective function is optimized using a second-order approximation to find the optimal weights and leaf sizes in the model. [17]

The structure of an extreme boosting tree is shown in Figure 3.5. Different from the random forest where the result of each decision tree is independent, in XGBoost, the result of the subsequent tree depends on the result of the previous tree.

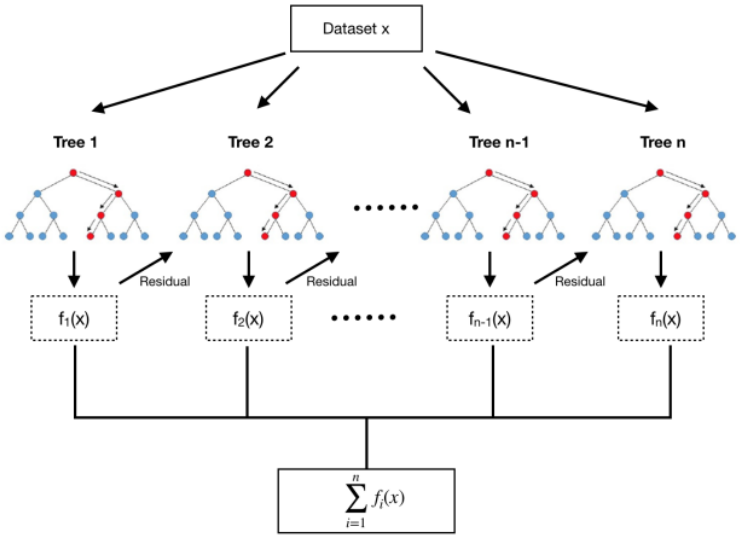

Figure 3.5: Extreme Gradient Boosting Tree

The advantages of XGBoost include computational efficiency and handling missing data. The efficiency is demonstrated in two aspects. First, XGBoost is based on an approximate greedy algorithm which uses weighted quantiles when looking for the best node split instead of evaluating every possible split. Second, XGBoost supports parallel learning by splitting up the data into smaller datasets to run processes in parallel. Moreover, when encountering missing data, XGBoost calculates Information Gain by putting observations with missing values into the left leaf and right leaf respectively and then chooses the scenario which produces the higher Gain. A major drawback of XGBoost is that it is sensitive to the outliers, since every classifier is forced to fix the errors in the predecessor learners before proceeding to the next step.

3.2.5 K-nearest Neighbour

The K-nearest Neighbour (KNN) algorithm is a supervised classifier to find the nearest neighbour for new observations. The observation x is mapped to k observations in a training set that are closest in input features to x from $\hat{Y}$. The prediction for x then simply becomes the average of these k observations.

$$\hat{Y}(x) = \frac{1}{k} \sum_{x_i \in N_k(x)} y_i \tag{3.2.15}$$

where N_k denotes the k closest neighbours of x with respect to the Euclidean distance in the p-dimensional space. For instance, if k = 3, when the loan x_i from the test set is mapped to the k closest loans in the training set, the corresponding labels of the three closets loans are (1,0,0), then the prediction of x_i takes the average of the three labels which is $\frac{1}{3}$. The most important hyperparameter of KNN is the number of neighbours k. It can be chosen by grid search which optimally discriminates between the classes.

Figure 3.6 is a visual representation of the example.

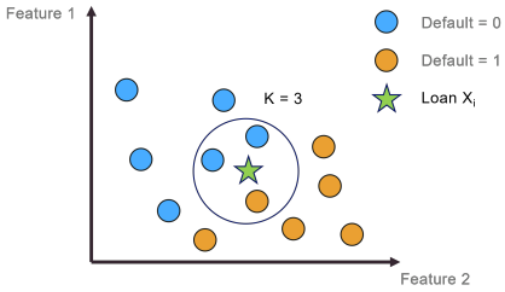

Figure 3.6: K-nearest Neighbour with K = 3

A major characteristic that differentiates KNN from other supervised learning algorithms is that it is a lazy learner which performs instance-based learning, meaning there is no training period. It does not derive any discriminative function from the training data. The algorithm stores the training dataset and learns from it only at the time of making predictions for the test data. Owe to this characteristic, new data can be added seamlessly, which will not impact the algorithm accuracy. Another advantage is that it is easy to implement, because it only requires two parameters, the number of neighbours k and the distance function, such as Euclidean or Manhattan, etc.

However, KNN has several disadvantages that need to be noted. First, KNN does not work well with large datasets and high dimensions. When the sample number is large, the cost of calculating the distance between the new point and the existing point is expensive. Similarly, if the dimension is high, it makes the algorithm difficult to calculate the distance in each dimension. Also, since KNN makes predictions based on the absolute value of distance, feature scaling becomes an essential step in data preprocessing. Finally, KNN is sensitive to noise in the dataset, so it requires the outliers to be removed and missing values to be imputed. [19]

3.2.6 Support Vector Machines

Support Vector Machines (SVM) classify samples by finding a hyperplane in the feature space to divide the samples that maximizes the minimum interval between two different samples. In the case of two classes, there are many possible hyperplanes that can be chosen. The objective is to find a hyperplane that has the maximum margin, which is the maximum distance between data points of both classes. The reason for maximizing the margin is that it enhances the confidence level for classifying future data. A common loss function used to maximize the margin is hinge loss, defined as

$$c(x, y, f(x)) = \begin{cases} 0, & \text{if } y \cdot f(x) \geq 1 \\ 1 - y \cdot f(x), & \text{else} \end{cases} \tag{3.2.16}$$

$$c(x, y, f(x)) = (1 - y \cdot f(x))^{+} \tag{3.2.17}$$

The loss is 0 if the sign of the predicted value equals the actual value; otherwise, the loss is given by the difference between 1 and the dot product of predicted and true values. Often, a regularization parameter is added to the objective function to reduce overfitting, written as

$$c(x, y, f(x)) = \min_{w} \lambda \|w\|^2 + \sum_{i=1}^{n} (1 - y_i \langle x_i, w \rangle)^{+} \tag{3.2.18}$$

To find the optimal weights w, one can take the partial derivative with respect to w and update w by gradient descent.

The original maximum-margin hyperplane algorithm constructs a linear classifier, but later the ability to be a non-linear classifier makes SVM a more powerful tool. The resulting algorithm is

similar but simply replaces the dot product of predicted and true values with a non-linear kernel function. [20] The kernel function transforms the original non-linear feature space into a linear feature space, so that the original SVM algorithm can be applied. Let the transformed data points be $\varphi(\mathbf{x}_i)$ and a kernel function k satisfy

$$k(\mathbf{x}_i, \mathbf{x}_j) = \varphi(\mathbf{x}_i) \cdot \varphi(\mathbf{x}_j). \tag{3.2.19}$$

The classification vector $\mathbf{w}$ in the transformed space satisfies

$$\mathbf{w} = \sum_{i=1}^{n} c_i y_i \varphi(\mathbf{x}_i) \tag{3.2.20}$$

where the c_i are obtained by solving the optimization problem

$$\begin{aligned} \text{maximize } f(c_1 \dots c_n) &= \sum_{i=1}^{n} c_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} y_i c_i \left(\varphi(\mathbf{x}_i) \cdot \varphi(\mathbf{x}_j)\right) y_j c_j \\ &= \sum_{i=1}^{n} c_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} y_i c_i k(\mathbf{x}_i, \mathbf{x}_j) y_j c_j \end{aligned} \tag{3.2.21}$$

subject to $\sum_{i=1}^{n} c_i y_i = 0$, and $0 \leq c_i \leq \frac{1}{2n\lambda}$ for all i. The coefficients c_i can be solved using quadratic programming and some index i can be found such that $0 < c_i < (2n\lambda)^{-1}$, so that $\varphi(\mathbf{x}_i)$ lies on the boundary of the margin in the transformed space, and then solve

$$\begin{aligned} b = \mathbf{w}^T \varphi(\mathbf{x}_i) - y_i &= \left[\sum_{j=1}^{n} c_j y_j \varphi(\mathbf{x}_j) \cdot \varphi(\mathbf{x}_i) \right] - y_i \\ &= \left[\sum_{j=1}^{n} c_j y_j k(\mathbf{x}_j, \mathbf{x}_i) \right] - y_i \end{aligned} \tag{3.2.22}$$

Finally,

$$\mathbf{z} \mapsto \operatorname{sgn}\left(\mathbf{w}^T \varphi(\mathbf{z}) - b\right) = \operatorname{sgn}\left(\left[\sum_{i=1}^{n} c_i y_i k(\mathbf{x}_i, \mathbf{z}) \right] - b \right). \tag{3.2.23}$$

Figure 3.7 [21] shows the hyperplane in 2D and 3D feature spaces.

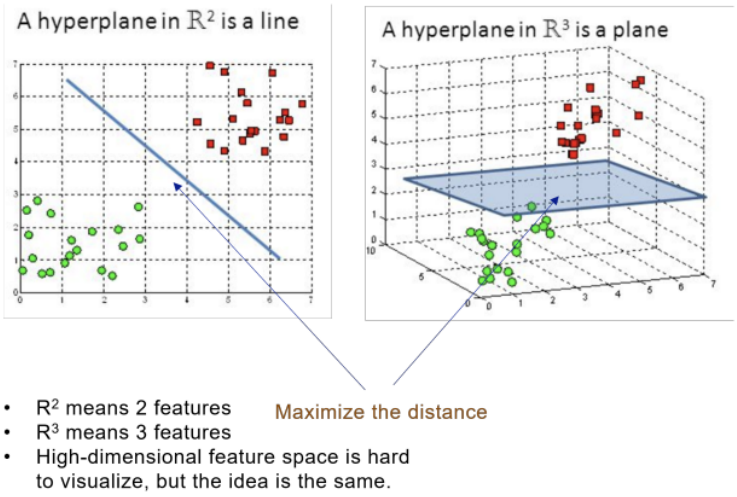

Figure 3.7: Support Vector Machine in two and three dimensional spaces

The advantages of SVM include the good performance for data with a clear margin of separation, the ability of scaling high dimensional data, good generalization ability with lower overfitting risk, and the capability of handling non-linear data with different kernel functions. On the other hand, a good choice of kernel function is challenging which depends on the data attributes.[22] The training time is quite long for large datasets. Moreover, the final model cannot be visualized which makes it difficult to understand and interpret the results and calibrate the model. This is a major drawback for financial data, since the regulators require model interpretations for modelling problems.

3.2.7 Artificial Neural Networks

An Artificial Neural Network (ANN) is a series of algorithms that endeavours to recognize underlying relationships in a set of data through a process that mimics the way the human brain operates. [23] It has an input layer, one or multiple hidden layers, and an output layer. The nodes are connected between each layer by certain weights and thresholds. If the output value of a node exceeds the threshold, then it is activated so that its value can be passed to the next layer.

The leftmost layer is the input layer with a set of neurons $x_1, x_2, ...x_n$, which are n input features. In the hidden layer, every neuron transforms the values of the layer to the left by a weighted linear summation shown in Equation 3.2.7 and subsequently inputted into an activation function σ. Lastly, the output receives the values and transforms them into a final prediction. The weights of the network are updated in an iterative manner by backpropagation and a pre-defined cost function.

To give a more formal definition, let I (the dimension of input features), O (the dimension of output values), r (the number of layers besides the input layer in the network) $\in \mathbb{N}$. A function $\boldsymbol{f} : \mathbb{R}^I \to \mathbb{R}^O$ is a feedforward neural network (FNN) with $r-1 \in \{0, 1, \ldots\}$ hidden layers, where there are $d_i \in \mathbb{N}$ units in the i-th hidden layer for any $i = 1, \ldots, r-1$, and activation functions $\sigma_i : \mathbb{R}^{d_i} \to \mathbb{R}^{d_i}, i = 1, \ldots, r$, where $d_r := O$, if

$$\boldsymbol{f} = \boldsymbol{\sigma}_r \circ \boldsymbol{Z}_r \circ \cdots \circ \boldsymbol{\sigma}_1 \circ \boldsymbol{Z}_1, \tag{3.2.24}$$

where $\mathbf{Z}_i : \mathbb{R}^{d_{i-1}} \to \mathbb{R}^{d_i}$, for any $i = 1, \ldots, r$, is an affine function

$$\boldsymbol{Z}_i(\boldsymbol{x}) := W^i \boldsymbol{x} + \boldsymbol{b}^i, \quad \boldsymbol{x} \in \mathbb{R}^{d_{i-1}}, \tag{3.2.25}$$

parameterised by weight matrix $W^i = \left[W^i_{j,k}\right]_{j=1,\ldots,d_i, k=1,\ldots,d_{i-1}} \in \mathbb{R}^{d_i \times d_{i-1}}$ and bias vector $\boldsymbol{b}^i = \left(b^i_1, \ldots, b^i_{d_i}\right) \in \mathbb{R}^{d_i}$, with $d_0 := I$.

A single layer neural network is shown in Figure 5.10 [24].

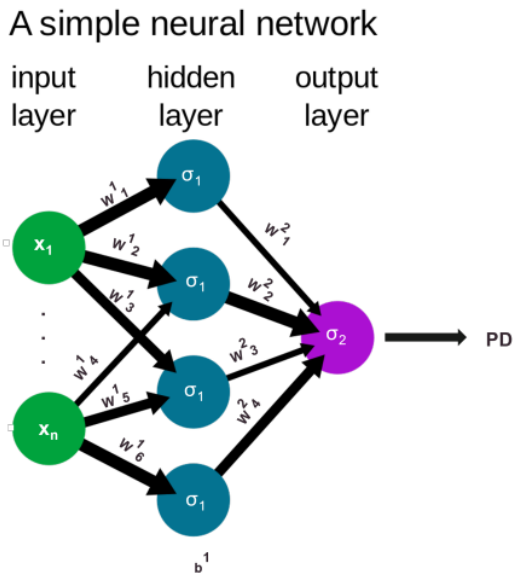

Figure 3.8: Artificial neural networks

In this study, since it is a binary classification problem, cross-entropy introduced in Section 3.1 is chosen to be the loss function and the sigmoid function shown in Figure 3.9 [25] is selected to be the activation function for the output layer as it can squeeze the value between 0 and 1 which is a proper interval for probability. The rectified linear Unit (ReLU) function shown in Figure 3.9 [25] is chosen to be the activation function in the hidden layer, because it performs better gradient propagation as there are fewer vanishing gradient problems compared to sigmoidal activation functions that saturate in both directions. Also, it is efficient in computation since it only requires comparison, addition and multiplication.

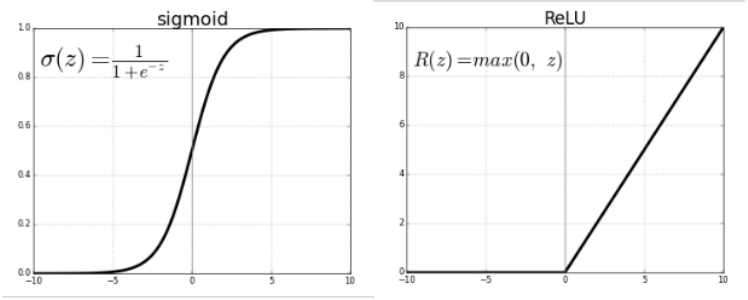

Figure 3.9: Activation Functions

ANNs have gained popularity over the decades due to various advantages. First, they can learn and model non-linear and complex relationships, that are difficult to discover at the first sight. Second, ANNs are good at generalization that after sufficient training, they can predict unseen data well. Third, they do not impose any restrictions on the put variables, unlike other statistical techniques. Many studies have shown that ANNs can better model heteroskedasticity, that is data with high volatility and non-constant variance, given its ability to learn hidden relationships in the data without imposing any fixed relationships in the data. [26]

Several disadvantages still exist. The biggest concern of ANN in the financial industry is the interpretability of the model. Since the procedure is done through "a black box", it is difficult to explain how the result is achieved and there is no way to visualize the model. Besides, the model requires a large amount datasets to train in order to well generalize the problem. There is no definite rule for choosing hyperparameters, so it requires significant trials and error.

3.3 Feature Transformation and Selection Methods

3.3.1 Weight of Evidence

Weight of Evidence (WoE) is a typical feature transformation method combined with logistic regression to reduce overfitting. WoE is used to assess the relative risk of the attributes within a feature, telling the predictive power of a single feature concerning its independent feature. The formula of WoE for each attribute in a feature can be written as

$$\begin{aligned} WoE_{attribute} &= \ln \frac{P_{\text{non-events}}}{P_{\text{events}}} \\ &= \ln \frac{N_{\text{attribute non-events}}/N_{\text{total non-events}}}{N_{\text{attribute events}}/N_{\text{total events}}} \end{aligned} \tag{3.3.1}$$

where

- $P_{\text{non-events}}$ is the percentage of non-event observations that exhibit the attribute
- P_{events} is the percentage of event observations that exhibit the attribute
- $N_{\text{attribute non-events}}$ is the number of non-event observations that exhibit the attribute
- $N_{\text{total non-events}}$ is the total number of non-event observations
- $N_{\text{attribute events}}$ is the number of event observations that exhibit the attribute
- $N_{\text{total events}}$ is the total number of event observations

A higher WoE indicates the attribute is more useful to separate events and non-events. WoE can transform both continuous and categorical variables. For continuous variables, we create bins based on certain intervals and WoE is calculated for each bin. A typical rule is that each bin should have at least 5% of the observations. For categorical variables, WoE can be directly calculated for each category. Then the original values in the dataset can be replaced by the values of WoE. WoE transformation is widely applied due to several benefits. First, it makes it possible to capture non-linear relationships between the predictors and the dependent variable. It also explicitly handles the outliers and missing values by either grouping them in an existing bin or creating a new bin. [3] Moreover, it handles categorical variables without the need of creating dummy variables.

3.3.2 Information Value

Information Value (IV), as one of the most useful techniques for feature selection in a predictive model, is calculated based on WoE:

$$\begin{aligned} IV &= \sum_{i=1}^{m} (P_{\text{non-events}} - P_{\text{events}}) \times WoE_i \\ &= \sum_{i=1}^{m} (\frac{N_{\text{attribute non-events}}}{N_{\text{total non-events}}} - \frac{N_{\text{attribute events}}}{N_{\text{total events}}}) \times WoE_i \end{aligned} \tag{3.3.2}$$

where WoE_i is the WoE for the i^{th} attribute in the feature and m is the total number of attributes in the feature. According to Siddiqi (2006) [27] , the IV Statistic in credit scoring follows the rule of thumb:

IV	Varaible Predictiveness	Notes
< than 0.02	Not useful for prediction	Unable to separate default and no default based on the feature
0.02-0.1	Weak predictive power	the feature has a weak relationship to the default/no default odds ratio
0.1-0.3	Medium predictive power	the feature has a medium strength relationship to the default/no default odds ratio
0.3-0.5	Strong predictive power	the feature has a strong relationship to the default/no default odds ratio
> 0.5	Suspicious predictive power	the feature has a suspicious relationship to the default/no default odds ratio and needs to double-check

Table 3.1: IV Metric Chart

The advantages of IV are that it is easy to be calculated and interpreted. Meanwhile, it is important to note that IV increases with the number of bins or categories, so a high IV associated with a large number of bins requires a second check. Besides, IV works well for logistic regression as conditional log odds are highly related to the calculation of WoE, but it is not necessary for other classification models such as random forest and SVM, because these algorithms have a good capability of detecting non-linear relationship.

3.3.3 Stepwise Regression

Stepwise regression is the step-by-step iterative construction of a regression model that involves the selection of independent variables to be used in a final model. It involves adding or removing potential explanatory variables in succession and testing for statistical significance after each iteration. [28] The underlying mechanism is that through a series of tests (e.g. F-tests, t-tests) to find a set of independent variables that are statistically significant to predict the dependent variable.

There are three main approaches: [29]

1. Forward selection begins with no explanatory variables, tests each variable, and then adds variables one by one, based on the statistical significance, until there are no remaining statistically significant variables.
2. Backward elimination starts with all possible explanatory variables and discards them one at a time. Then it tests to see if the removed variable is statistically significant. The elimination stops when each variable remaining in the equation is statistically significant. One challenge is that it does not work when the number of candidate variables exceeds the number of observations.
3. Bidirectional stepwise procedure is a combination of forward selection and backward elimination. The procedure starts with no variables and adds variables using a pre-specified criterion. Then at every step, the procedure also considers the statistical consequences of dropping variables that were previously included. So, a variable might be added in Step 2, dropped in Step 5, and added again in Step 9.

Stepwise regression is efficient to choose a relatively small number of explanatory variables from a vast array of possibilities. However, a limitation of the approach pointed out by many studies is that it may create a false confidence interval in the final model when standard statistical tests assume a single test of a pre-specified model and are not appropriate when a sequence of steps is used to choose the explanatory variables. [29]

3.3.4 Random Forest with Recursive Feature Elimination

Aside from being a classifier, random forest is also widely applied as a feature selection tool. Each tree of the random forest performs random extraction of observations and features and then cal-

culates the importance of a feature according to its ability to increase the pureness of the leaves. The higher the increment in leaf purity, the higher the importance of the feature. The final output is the average of all the trees in the forest. The pureness is measured through either the Gini impurity or the information gain/entropy introduced in Section 3.2.2.

The feature importance is based on the impurity reduction achieved by splitting on the features, that is how much this feature contributes to decrease the Gini impurity in a classification problem. For a given binary node m with left and right child nodes, the impurity reduction called Gain_m is calculated as

$$\text{Gain}_m = \text{impurity}_m - \left(\text{weight}_{left} \cdot \text{impurity}_{\text{left}} + \text{weight}_{\text{right}} \cdot \text{impurity}_{\text{right}} \right) \quad (3.3.3)$$

The weights is defined as the share of the parents examples in a child node, formulated as

$$weight_{\text{left}} = N_{\text{left}} / N_m \quad (3.3.4)$$

where N is the number of examples in a node or leaf. To derive the total impurity reduction of a given feature f in the tree t, Gain_m needs to be summed across all nodes $m \in M_f^{(t)}$, which perform a split on that feature f and divide it by the total impurity reduction number of all nodes of that tree:

$$Importance_f^{(t)} = \frac{\sum_{m \in M_f^{(t)}} Gain_m}{\sum_f \sum_{m \in M_f^{(t)}} Gain_m} \quad (3.3.5)$$

With this normalization, the feature importances can sum up to 1. Finally, the total importance of a feature f is calculated across all trees t in the random forest with a total number of trees T :

$$\text{Importance}_f = \frac{1}{T} \sum_{t=1}^{T} \text{Importance}_f^{(t)}$$

Once the feature importance is obtained, feature selection can be performed with a procedure called Recursive Feature Elimination. The concept behind it is to first fit the model with all features except the least relevant feature and calculate the performance metric. Then the model is fitted again after removing the second least important feature and the performance metric is calculated again. The procedure is repeated until there is no feature left. The set of features that maximizes the performance metric is the set of features to be selected. The entire procedure needs to work with the same values for the hyperparameters.

Selecting features by using tree-derived feature importance is straightforward, fast and generally accurate way for machine learning. A drawback of this approach is rooted in multicollinearity. If there are highly correlated predictors in a training set that are useful for predicting the outcome, then which predictor is chosen for partitioning the samples is essentially a random selection. When there is a set of highly redundant and useful predictors in the splits across the ensemble of trees, the predictive performance of the ensemble of trees is unaffected by highly correlated, useful features. However, the redundancy of the features dilutes the importance scores. [30]

3.4 Performance Measures

3.4.1 Mean Squared Logarithmic Error

Mean squared logarithmic error (MSLE) can be interpreted as a measure of the ratio between the true and predicted values. It is a variation of Mean Squared Error (MSE). The loss is the mean over the seen data of the squared differences between the log-transformed true and predicted values, written as:

$$L(y, \hat{y}) = \frac{1}{N} \sum_{i=0}^{N} \left(\log\left(y_i + 1\right) - \log\left(\hat{y}_i + 1\right) \right)^2 \quad (3.4.1)$$

where $\hat{y}$ is the predicted value. This loss can be interpreted as a measure of the ratio between the true and predicted values, since:

$$\log(y_i + 1) - \log(\hat{y}_i + 1) = \log\left(\frac{y_i + 1}{\hat{y}_i + 1}\right) \tag{3.4.2}$$

The reason ' 1 ' is added to both y and $\bar{y}$ is for mathematical convenience since $\log(0)$ is not defined but both y or $\hat{y}$ can be 0.

One characteristic of MSLE is its robustness to the effect of the outliers. When using MSE, the presence of outliers can explode the error term to a large magnitude, but in the case of MSLE, the outliers are drastically scaled down by the logarithmic term. Another characteristic inherent in MSLE is the biased penalty. It incurs a larger penalty for the underestimation of the actual variable than the overestimation. This is especially useful for business cases where the underestimation of the target variable is not acceptable, but overestimation can be tolerated. In default flag prediction, it is more important to detect the presence of default. If the default probability is underestimated, then the potential default cannot be captured, which can lead to a huge loss.

3.4.2 Averaged Log Loss

Log-loss, sometimes called cross-entropy, indicates how close the prediction probability is to the corresponding actual/true value which is 0 or 1 in case of binary classification. The more the predicted probability diverges from the actual value, the higher the log-loss value is. To calculate the averaged log loss, it simply takes the average of the log loss of individual samples shown in Equation 3.1.3, formulated as

$$L = -\frac{1}{N}\sum_{i=1}^{N} y_i \ln(p_i) + (1 - y_i)\ln(1 - p_i) \tag{3.4.3}$$

where y_i is the actual target, p_i is the predicted probability, N is the number of samples. The benefit of using averaged log loss is that it penalizes more when the prediction deviates more from the actual target.

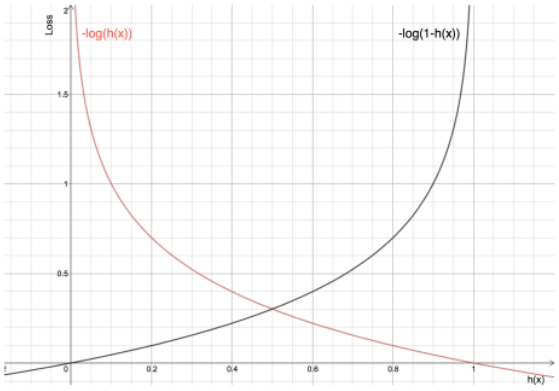

Figure 3.10: Averaged Log Loss

In the graph [31] , the y-axis represents loss and the x-axis represents predicted probability. The red line shows the loss behaviour for the class with label 1. When the predicted probability approaches 1, the loss approaches 0; on the other hand, when the predicted probability approaches 0, the loss approaches infinity. Similarly, the black line shows the behaviour for the class with the label 0. If the loss is small when the prediction is close to 0 and becomes larger and larger when the prediction approaches 1.

3.4.3 Accuracy

Accuracy is the most popular metric for classification problems to evaluate the model performance. It is formulated as

$$Accuracy = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Samples}} \tag{3.4.4}$$

A higher accuracy is achieved when the model makes more correct predictions. The accuracy ranges from 0 to 1, which corresponds to zero correct prediction and perfect predictions. Besides, the accuracy is often expressed with percentages, multiplying the accuracy with 100.

Despite its popularity, there are some major drawbacks. First, accuracy is not capable of utilizing the output of probability but only based on the class label. For example, two models can achieve the same accuracy, whereas one has low confidence and the other has high confidence. In this case, accuracy cannot differentiate which model has a better performance. Moreover, when encountering imbalanced datasets, it is easy to achieve high accuracy by simply making all predictions to be the majority class. However, the classifier is in fact meaningless and there is no way to tell by only looking at the accuracy.

3.4.4 Brier Score

The Brier Score is a strictly proper score function that measures the accuracy of probabilistic predictions. For unidimensional predictions, it is strictly equivalent to the mean squared error as applied to predicted probabilities. The formula according to Brier (1950) is that

$$\text{Brier Score} = \frac{1}{n}\sum_{i}^{n}(p_i - y_i)^2 \tag{3.4.5}$$

where p_i is the estimated probability, y_i is the actual target, and n is the number of observations in the dataset.

The benefit of the Brier Score is that the concept is easy to understand and implement. It provides a way to assess the predictive performance of different models where a lower score indicates a superior performance of the model when the other metrics such as accuracy are the same. [17] For example, for a non-default sample with the true label to be 0, when the threshold for default is at 0.5, a model predicts the PD to be 0.4 whereas another model predicts the PD to be 0.2. Although both models make correct predictions, the latter model performs better in terms of Brier Score, because the predicted PD, 0.2, is closer to 0, compared to 0.4.

3.4.5 Confusion Matrix

Confusion matrix is used to measure the performance of the classification model, because checking the model performance by accuracy is misleading when having imbalanced data. The matrix is a table with four components. [32]

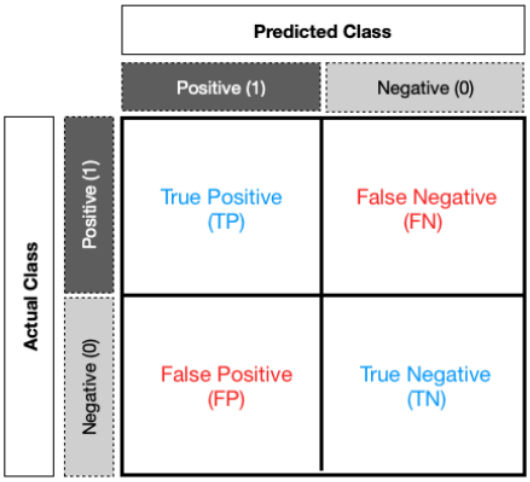

Figure 3.11: Confusion Matrix

where

- Predicted Values: the returned values predicted by the model
- Actual Values: the true label provided by the dataset
- True Positive (TP): the values that are actually positive and predicted as positive. In this study, it is the default samples predicted as default.
- False Positive (FP): the values that are actually negative and predicted as positive. In this study, it is the no default samples predicted as default.
- False Negative (FN): the values that are actually positive but predicted as negative. In this study, it is the default samples predicted as no default.
- True Negative (TN): the values that are actually negative and predicted as negative. In this study, it is the no default samples predicted as no default.

The confusion matrix can also be represented in the rate form for better interpretability:

1. True Positive Rate (TPR)/ Sensitivity/Recall

$$Sensitivity = \frac{TP}{TP + FN}$$

This indicates what proportion of the positive class got correctly classified. In this study, it indicates the percentage of default samples that are correctly classified. A higher TPR is desired.

2. False Negative Rate (FNR)

$$FNR = \frac{FN}{TP + FN} = 1 - Sensitivity$$

This calculates what proportion of the positive class got incorrectly classified by the classifier. A lower FNR is desired.

3. True Negative Rate (TNR)/ Specificity

$$Specificity = \frac{TN}{TN + FP}$$

Specificity indicates what proportion of the negative class got correctly classified. In this study, it indicates the percentage of no-default samples that are correctly classified. A higher TNR is desired.

4. False Positive Rate (FPR)

$$FPR = \frac{FP}{TN + FP} = 1 - Specificity$$

This calculates what proportion of the negative class got incorrectly classified by the classifier. A lower FPR is desired.

3.4.6 Area under Receiver Operating Characteristics Curve

The Receiver Operator Characteristic (ROC) curve is an evaluation metric for binary classification problems. It is a probability curve that plots the True Positive Rate (TPR) against the False Positive Rate (FPR) at various threshold values. Here, TPR and FPR come from the concept of confusion matrix described in the previous Section 3.4.5

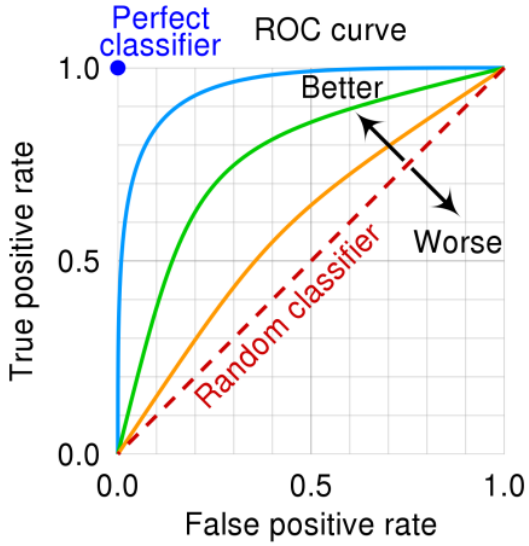

Figure 3.12: ROC Curve

Figure 3.12 [33] displays several ROC curves. The blue curve represents the best classifier followed by the green and orange curves. The red dashed line is a random classifier, meaning there is a fifty-to-fifty chance of predicting 0 and 1. Thus, only curves above the dashed line have an adequate classifying ability.

In general, an adequate model should output high sensitivity and specificity with low FNR and FPR. Another important concept when discussing AUC is the probability threshold. A standard threshold is 0.5, meaning if the probability is greater than 0.5, then the sample is classified as positive and classified as negative otherwise. But one can also set the threshold manually, because different thresholds may change the sensitivity and specificity of the model. The threshold giving the best result is chosen to be the optimal threshold.

The Area Under the Curve (AUC) is the measure of the ability of a classifier to distinguish between classes and is used as a summary of the ROC curve. The higher the AUC, the better the performance of the model at distinguishing between the positive and negative classes. In a ROC curve, a higher X-axis value indicates a higher number of False positives than True negatives, while a higher Y-axis value indicates a higher number of True positives than False negatives. So, the choice of the threshold depends on the ability to balance False positives and False negatives. The table summarizes the interpretation of AUC. [34]

AUC	Model Performance	Explanation
0	Poor	make completely wrong predictions which predict all Negatives as Positives and all Positives as Negatives
0 - 0.5	Inadequate	make weak predictions which predict most samples incorrectly and when AUC equals to 0.5, it makes random guess
0.5 - 1.0	Adequate	there is a high chance that the classifier will be able to distinguish the positive class values from the negative class values
1	Perfect	perfectly distinguish between all the Positive and the Negative class points correctly

Table 3.2: AUC Metric Chart

ROC curve can be visualized to see the tradeoff between sensitivity and specificity for all possible thresholds rather than simply using the default setting. Also, AUC is extremely useful for imbalanced datasets. It overcomes the weakness of accuracy by balancing the class sizes when computing the score.

Chapter 4

Data Description

4.1 Data Source and Anonymity

The dataset has been collected from a financial services firm. It was provided by the firm by extracting the information from the firm's data warehouse. There are 25530 observations, which consist of various information for home loans for six months from 2018 to 2019. For confidential reasons, the financial services firm shall remain nameless throughout the thesis and will be referred to as "the firm" henceforth.

4.2 Candidate Variables

There are 51 candidate variables which are previously identified through studies of the quantitative solution team of the firm. The variables are related to risk analytics that the firm has deemed potentially important for the loan evaluations. The full list of variables is summarized in the table that can be found in Appendix A.1. Among all these variables, 24 variables are removed as they have either no impact on PD modelling or are not applicable to PD modelling, listed in Appendix A.1.

4.3 Target

The goal of this study is to predict the default flag of 2019. For each loan, there are five months of data available during 2019, from August to December. The target is determined by choosing the maximum default flag during these five months. In other words, if the loan defaults during any of these months, then the default flag is set to 1; otherwise, the default flag is set to 0.

4.4 Data Preprocessing

To predict the default flag of 2019, only performing loans of 2018 are considered, so the loans with the default flags of 2018 which equal "Yes" are dropped. Then the 2018 December data is inner joined with 2019 December data based on the unique Asset ID. With the merged data frame, three new variables are created, called "Month in Book", "Maturity", and "Month Maturity Change", which represent the difference between "Effective Date" and "Origination Date", the difference between "Contractual Maturity" and "Origination Date", and the difference between "Effective Date" and "Latest Maturity Changed Date" respectively. Then, the origination value is adjusted by monthly compounding with inflation rates of the UK from 1997 to 2018 to calculate the present value as of December 2018. Finally, 24 variables that are not used are dropped, left with 30 used variables. The list of 30 used variables is available in Appendix A.1.

Among the variables, there is a mixture of continuous variables and categorical variables. In order to have numerical values to be the input of the models, all the variables undergo a weight of evidence transformation introduced in Section 3.3.1. The table summarizes the statistics of the dataset:

Number of Samples	Number of No Default	Number of Default	Default Rate
23413	22542	871	3.72%

Table 4.1: The statistics of dataset

Figure 4.1 demonstrates the entire process of data preprocessing which also includes the feature selection that will be introduced in the following section.

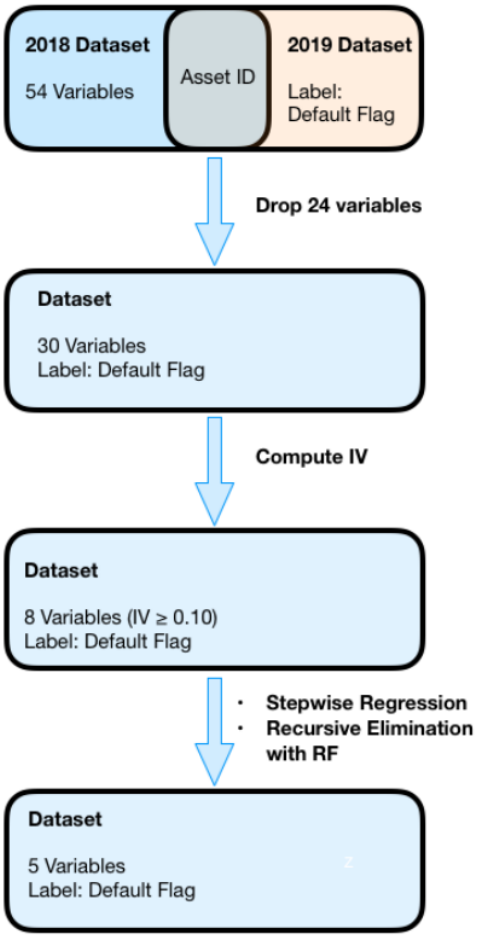

Figure 4.1: Data preprocessing flowchart

Chapter 5

Results and Discussion

5.1 Feature Selection

In order to avoid overfitting and save computational cost, it is important to only incorporate necessary features in the model. The feature selection process consists of two parts. First, all variables are included in a univariate analysis to find variables with high discriminatory power, using the Information Value (IV) introduced in Section 3.3.2 as the criteria. This provides a quick way to filter our variables that do not have explanatory power. The cutoff of IV is set to be 0.10, meaning a variable with an IV lower than 0.10 will not be included for further analysis, because it suggests this variable has weak or no predictive power explained in Section 3.3.2. The results are shown in the plots below. After the first round selection, it is left with 8 variables.

The variables are sorted based on IV:

Index	Variable	IV
1	**PaymentMethod**	**0.927379**
2	**CurrentPoolID**	**0.555402**
3	**PD_Segment**	**0.504481**
4	**ProductGrade**	**0.473292**
5	**AssetSegment**	**0.456299**
6	**month_in_book**	**0.371195**
7	**PropertyType**	**0.254173**
8	**CSOFlag**	**0.182347**
9	AMCGroupEntity	0.0797
10	PropertyRegion	0.0337
11	LGD_Segment	0.0273
12	Bankruptcy_Flag	0.0215
13	PropertyType2	0.0204
14	EIR	0.0194
15	Capitalisation	0.0149
16	Collateral_Value_AccountID_Level	0.0135
17	Collateral_Value	0.0124
18	DWPPayer	0.0114
19	PropertyRegion2	0.0092
20	BalanceAtTerm	0.0071
21	AdvanceAmount	0.0065
22	maturity	0.0058
23	OriginalValuation	0.0045
24	TempIOSwitch	0.0041
25	Outstanding_Balance	0.0019
26	IndividualVoluntaryArrangementFla	0.0019
27	Repayment_Type	0.0016
28	TermExtension	0.0001
29	Possession	0.0000
30	month_maturity_change	0.0000

Figure 5.1: Sorted variables based on Information Value

The second part is to choose the variables that can exert significant impacts on model results. By applying Stepwise Regression, it can be seen from Figure 5.2, the optimal result is obtained when

five variables are incorporated in the model, as the AUC score reaches the highest point. The best combination is 'CSO Flag', 'PD Segment', 'Payment Method', 'Current Pool ID', and 'Month in Book'. Meanwhile, the selection methods also indicate that the rest variables do not improve the model performance on the prediction results.

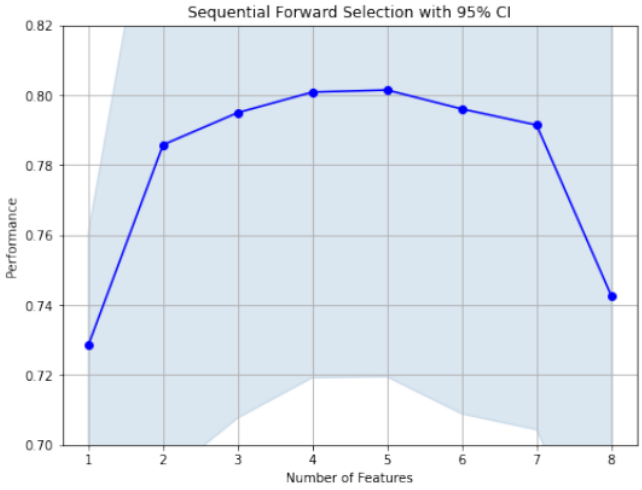

Figure 5.2: Feature Importance by Stepwise Regression

The selection method Recursive Feature Elimination with Random Forest is also performed based on the calculation described in Section 3.3.4. Figure 5.3 shows the importance value of all the features. Then, the features need to be eliminated recursively. First, the least important feature, 'Property Type', is removed and we calculate the AUC score followed by fitting the model. Then, the model is fitted and the AUC score is calculated after removing 'Product Grade' which is the second least important feature. The procedure is repeated until we finish evaluating the model that only incorporates 'Payment Method'. By comparing the AUC score, we find that the score reaches the highest when the top five variables are incorporated which are the identical set of variables obtained from Stepwise Regression. Thus, these five variables are used for the final analysis.

The correlations among the variables are also computed, which can be found in Appendix A.2, to ensure there is no multicollinearity. Multicollinearity is the occurrence of high intercorrelations among two or more independent variables in a multiple regression model. It can result in wider confidence intervals with larger standard errors which lower the statistical significance of regression coefficients and thus the regression model becomes less reliable.

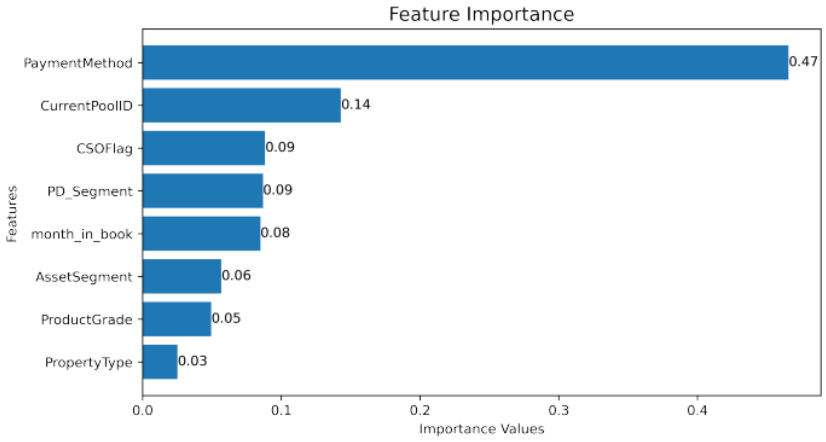

Figure 5.3: Feature Importance by Random Forest

The bar charts display the Default Rate versus these five variables:

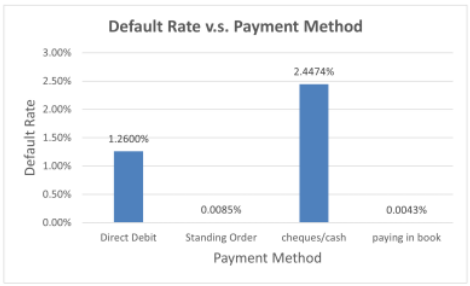

Figure 5.4: Default Rate v.s Payment Method

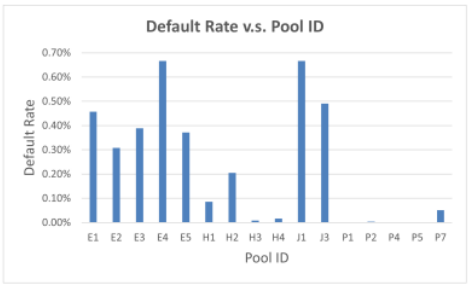

Figure 5.5: Default Rate v.s Pool ID

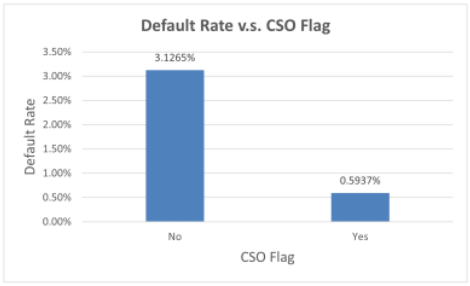

Figure 5.6: Default Rate v.s CSO Flag

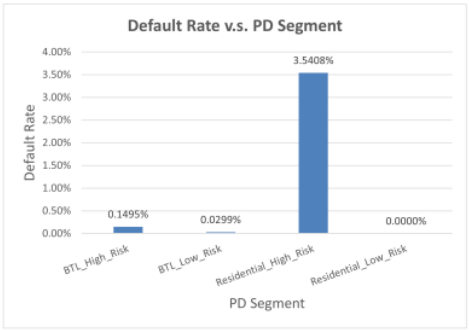

Figure 5.7: Default Rate v.s PD Segment

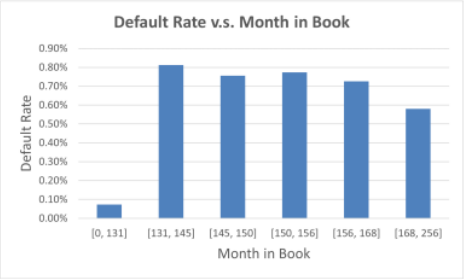

Figure 5.8: Default Rate v.s Month in Book

From the bar charts of variables, the relationship between the variable attributes and the default rate can be identified. For different payment methods, cheque and cash have the highest default rate which is much riskier compared to the direct debit, whereas for the rest two payments, it is difficult to determine the risk level as there are not enough observations. From different pool IDs, it can be found the loans held by Entity E and J have relatively high default rates. Also, the Credit Services Organizations (CSO) loans are much less likely to be defaulted compared with the rest, as CSOs are licensed companies that assist consumers in obtaining loans from unaffiliated third-party lenders. Moreover, the high risk segments also have higher default rates as expected. Finally, the chart suggests that the new loans issued within 131 months have the least probability to go defaulted.

5.2 Model Evaluation

5.2.1 Data Splitting and Handling Imbalance

In order to evaluate the model, the data are split into train and test sets with a 7:3 ratio. The statistics are summarized in the table:

Data set	Number of Samples	Number of No Default	Number of Default	Default Rate
Training set	16389	15797	592	3.61%
Test set	7024	6745	279	3.97%

Table 5.1: The statistics of dataset

From Table 5.1, it can be found the dataset is extremely imbalanced with much more no default samples. In order to handle the imbalance, the cost-sensitive learning method is adopted as introduced in Section 2.4. A higher weight is imposed on the training cost or loss for the default samples based on the default and no default ratio, meaning if a default sample is misclassified as a no default sample, the penalty is larger. The weight is the reversed ratio of the two classes. For example, in the training set, the ratio of default and no default is 3.61:100. Then, the weight gives to the default class when computing the loss function is 100, whereas the weight for the no default class is 3.61.

5.2.2 Hyperparameter Tuning

An important question in machine learning models is hyperparameter tuning, because different hyperparameters can produce significant differences in results. Since there are a large number of combinations of hyperparameters for each model, it is computationally expensive to implement manual trials. Thus, a grid search optimizer is conducted to select the optimal hyperparameters. The idea is to perform an exhaustive search on a set of different values for the hyperparameters, to find which set of hyperparameter values can optimize the objective function of the model. The criterion for determining the optimal value is AUC score, that is to check which candidate hyperparameter achieves the highest score.

Model	Hyperparameters	Search Space	Optimal Value
Decision Tree	Minimum samples in a leaf	1,2,3,4	2
	Minimum samples to split	[2, 10]	3
	Maximum depth	[2, 10]	2
Random Forest	Maximum number of trees	[100, 500]	200
	Minimum samples in a leaf	1,2,3,4	2
	Minimum samples to split	[2, 10]	3
	Maximum depth	[2, 10]	2
XGBoost	Maximum number of trees	[100, 500]	200
	Maximum depth	[2, 10]	2
	eta	[0.1, 0.5]	0.5
	gamma	[0, 0.2]	0
	regularization term	[0, 1.5]	1
KNN	Number of neighbors	[2, 30]	26
SVM	kernel	'linear', 'poly', 'rbf', 'sigmoid'	'rbf'
	gamma	'scale', 'auto'	'auto'
	regularization term	[0, 1.5]	1
ANN	Number of layers	1,2,3	2
	Number of neurons (1st layer)	[2, 10]	6
	Number of neurons (2nd layer)	[2, 10]	4
	Number of Epochs	[10, 30]	20
	Batch size	10, 30, 50, 100	30

Table 5.2: Overview of Search Space for Hyperparameter Tuning

Figure 5.9 demonstrates the entire process of default prediction.

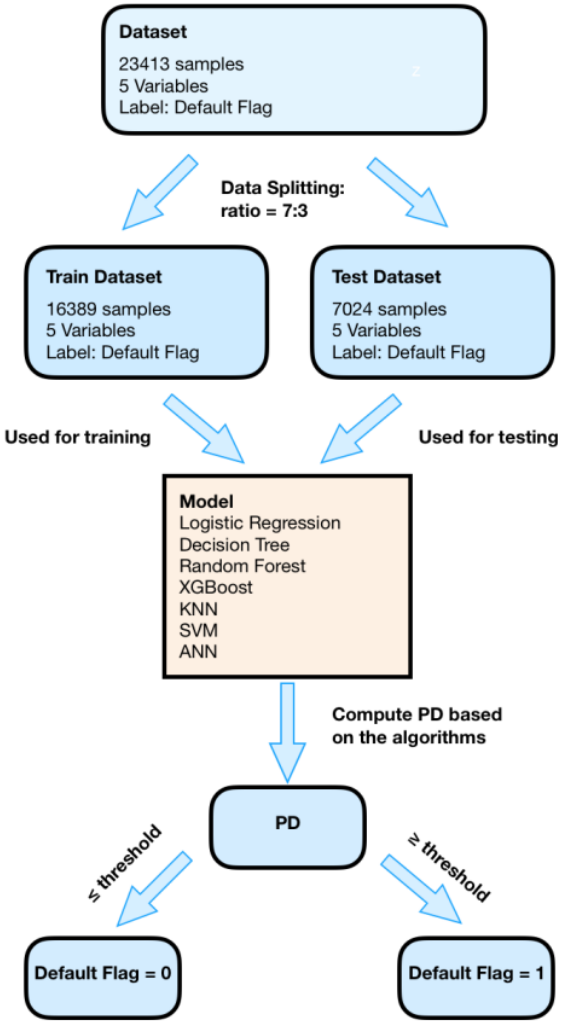

Figure 5.9: Default prediction flowchart

5.2.3 Model Output

The logistic regression model output is summarized in Table 5.2.3

Variable	Coef	Mean	Std	97.5% CI	p-Value
Intercept	-0.0108	-0.0243	0.0199	(-0.0308,0.0091)	0.0281
PaymentMethod	-0.8651	-0.8638	0.0373	(-0.9024, -0.8277)	1.2E-05
CurrentPoolID	-0.6721	-0.5659	0.1644	(-0.8366,-0.5076)	2.2E-05
CSOFlag	-0.5334	-0.5753	0.0785	(-0.6120,-0.4548)	5.9E-05
PD_Segment	-0.0813	-0.1267	0.0970	(-0.1784,0.0156)	0.0081
month_in_book	-0.1309	-0.1774	0.0858	(-0.2167,0.0451)	0.0335

Table 5.3: The outputs of logistic regression model

The values are obtained by running logistic regression model ten times, where

- Coef: the coefficient of the variable
- Mean: the coefficient value averaged over all the results by running ten times
- Std: the standard deviations of the results
- 97.5% CI: 97.5% confidence interval
- p-Value: the probability of obtaining the observed results, assuming that the null hypothesis is true, where the null hypothesis is 0 for all variables

Here, the p-values are all below 0.05 which is typically the threshold of significance, so we can conclude that the null hypotheses of the variables can be rejected.

To train the neural network, the training set is split into a ratio of 8:2 with 80% as the training set and 20% as the validation set. Figure 5.10 demonstrates the metric and loss during the training process.

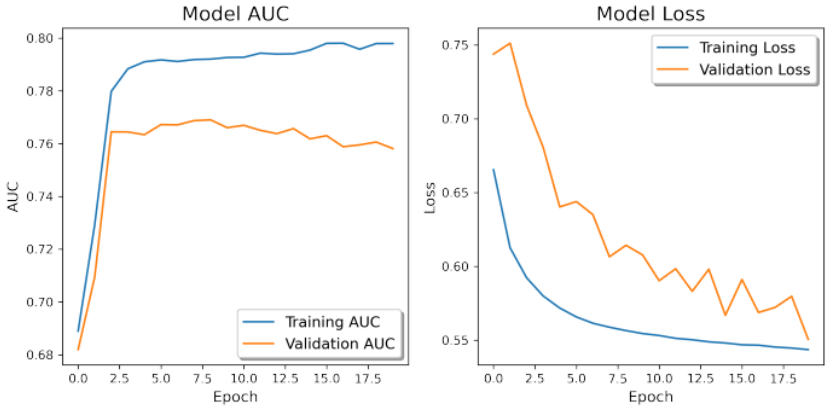

Figure 5.10: The values of AUC and loss during the training process

It can be seen that the AUC score for the training set keeps increasing with the increase number of epochs, whereas for the validation set, the AUC score increases at the beginning, reaching the highest at the eighth epoch, and then decreases afterwards. It suggests that overfitting can happen as the training progresses and early stopping is a strategy to prevent this.

For the rest machine learning algorithms, since they are non-parametric, there are no coefficients or weights to be trained, we will evaluate these models based on the metrics in the following section.

5.2.4 Metric Evaluation

All the machine learning algorithms are written based on scikit-learn, the machine learning library for the Python programming language.[35] The table summarizes the model performance with the selected variables based on the metrics introduced in Section 3.4

Model	MSLE	Avg LL	TPR	FPR	AUC	Brier Score	Time (s)
Logistic Regression	0.5559	0.1153	0.7011	0.2102	0.7641	0.1765	0.3150
Decision Tree	0.5629	0.1221	0.6882	0.2176	0.7585	0.1887	0.0834
Random Forest	0.5274	0.1022	0.7018	0.2004	0.7737	0.1654	2.1808
XGBoost	0.5621	0.1298	0.5806	0.2335	0.6990	0.1938	3.3348
KNN	0.5893	0.1347	0.5663	0.1764	0.7059	0.1902	1.2513
SVM	0.5025	0.0901	0.7155	0.2038	0.7755	0.1553	613.4508
ANN	0.4578	0.0712	0.7258	0.1885	0.7863	0.1257	20.9574

Table 5.4: The metrics for different models with selected variables

The table summarizes the model performance with the selected variables based on the metrics introduced in Section 3.4

Model	MSLE	Avg LL	TPR	FPR	AUC	Brier Score	Time (s)
Logistic Regression	0.5535	0.1102	0.7091	0.2012	0.7732	0.1789	1.2590
Decision Tree	0.5611	0.1211	0.6902	0.2136	0.7685	0.1802	0.1198
Random Forest	0.5221	0.1009	0.7118	0.2001	0.7782	0.1658	3.5746
XGBoost	0.5608	0.1254	0.5906	0.2275	0.6801	0.2012	5.4599
KNN	0.5811	0.1315	0.5763	0.1684	0.6962	0.1954	2.2775
SVM	0.5022	0.0876	0.7167	0.2018	0.7801	0.1523	700.0089
ANN	0.4513	0.0703	0.7306	0.1817	0.7923	0.1203	28.0257

Table 5.5: The metrics for different models with all variables

From both tables, it can be found ANN achieves the highest AUC score, followed by SVM, RF, LR, DT, KNN, and XGBoost. This performance ranking is also reflected in all other metrics except Time. The best two models however take longer time, especially SVM, whereas the time taken by ANN is relatively acceptable.

By comparing the two tables, the AUC scores are slightly higher, about 1% for the models trained with all variables compared to the models trained with the selected 5 variables by feature selection. However, the Brier Score is not necessarily lower, which suggests there might be a tendency of overfitting if all the variables are incorporated in the model. On the other hand, the time increases significantly with the increase of the number of variables.

The ability of the classifiers to discriminate between default and non-default samples is evaluated with the ROC chart. The chart is set up such that the False Positive Rate (FPR) and the True Positive Rate (TPR) are plotted against each other, with different probability thresholds. The dots on the curve represent the TPR and FPR with the optimal threshold for each model. The ROC chart for all classifiers is illustrated in Figure 5.11. Since the classifiers perform roughly equally well, with most classifiers achieving a ROC AUC between 0.76 and 0.79, it is difficult to clearly see the difference on the graph. Still, one can observe that the deep neural network (purple) consistently has a higher TPR for a given FPR than the other classifiers. Also, it is clear from the figure that the XGBoost and the KNN perform worst. One reason can account for this is that the samples may contain outliers, since both algorithms, XGBoost and KNN, are sensitive to outliers which leads to poor performance as discussed in Section 3.2.14 and 3.2.5.

Figure 5.11 shows the ROC curves obtained from the test dataset with all the models trained and tested with the selected five variables after WoE transformation:

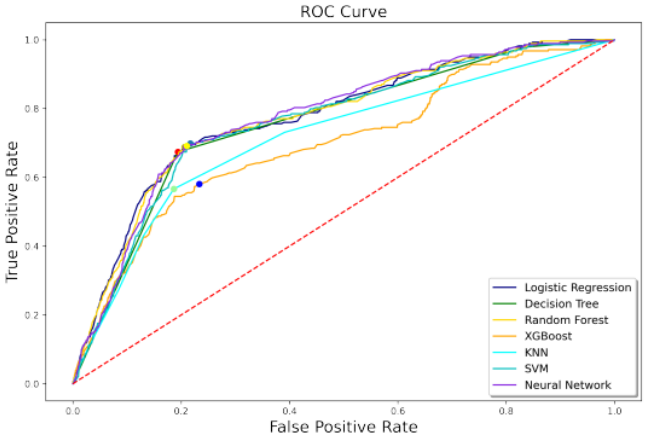

Figure 5.11: ROC curves of all models with selected variables

Figure 5.12 shows the ROC curves obtained from the test dataset with all the models trained and tested with all the variables after WoE transformation:

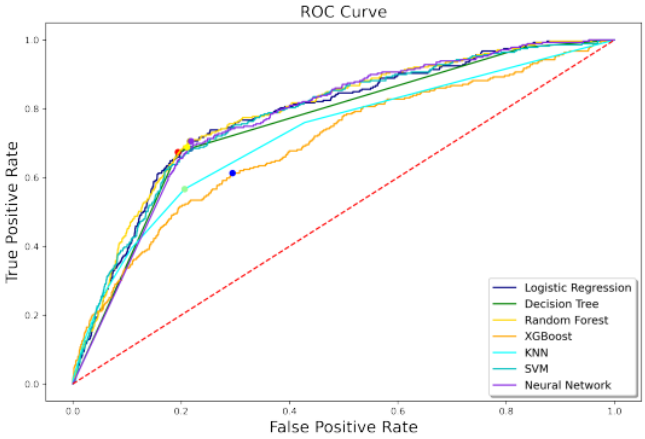

Figure 5.12: ROC curves of all models with all the variables

From the above two figures, it can be observed that there is no large discrepancy, which means the variables chosen through feature selection capture the majority of information of the dataset, so that other variables provide little information for prediction and there is no need to incorporate them. It suggests that feature selection for high-dimensional datasets is essential for saving computation power and avoiding potential overfitting.

The dots on the ROC curves represent the corresponding TPRs and FPRs with the optimal thresholds. The threshold is optimized based on the geometric mean of sensitivity (TPR) and specificity (TNR), also called G-Mean in short. The formula is

$$\begin{aligned} G\text{-}Mean &= \sqrt{TPR * TNR} \\ &= \sqrt{TPR * (1 - FPR)} \end{aligned} \tag{5.2.1}$$

The optimal probability threshold is the one which yields the highest G-Mean. The table summarizes the optimal threshold for each model.

Model	Optimal Threshold
Logistic Regression	0.6043
Decision Tree	0.7654
Random Forest	0.5596
XGBoost	0.4754
KNN	0.3069
SVM	0.3945
ANN	0.4762

Table 5.6: The optimal threshold for different models with selected variables

Figure 5.13 displays the confusion matrix for all the algorithms trained and tested with selected variables. The values are calculated based on the optimal threshold probability for each model. As it can be seen, Neural Network achieves the highest True Positive Rate and True Negative Rate, which means it can make the most number of correct predictions for both default and no default samples. In default prediction, a low FNR is desired, because when a loan actually defaults but is predicted as no default, a huge loss would occur. Thus, when evaluating the performance of RF and LR, although they only have a 0.07% difference in TPR, the FNR of RF is 0.98% higher than LR, making RF slightly more reliable.

On the other hand, although LR is less precise than ANN, RF, and SVM, the high interpretability remains a remarkable advantage of LR. It has a clear coefficient for each variable so that one can know how much influence each variable exerts on the final result. Also, a p-value test can be conducted on each variable to see if it is statistically significant. On the contrary, such interpretability is hard to achieve for the machine learning algorithms. For SVM, the plane separation cannot be visualized in high-dimensional space when there are more than three variables. For RF, it is difficult to plot all the trees and explain how the final node separations are achieved. Finally, for ANN, the interpretation of "Black Box" is still an unresolved topic in academia. Although ANN has achieved many practical applications in various industries, the drawback of low interpretability limits its usage in the financial industry which emphasizes the clear interpretation of the models. Hopefully, with more research done in this area, the "Black Box" will be unrevealed in the future.

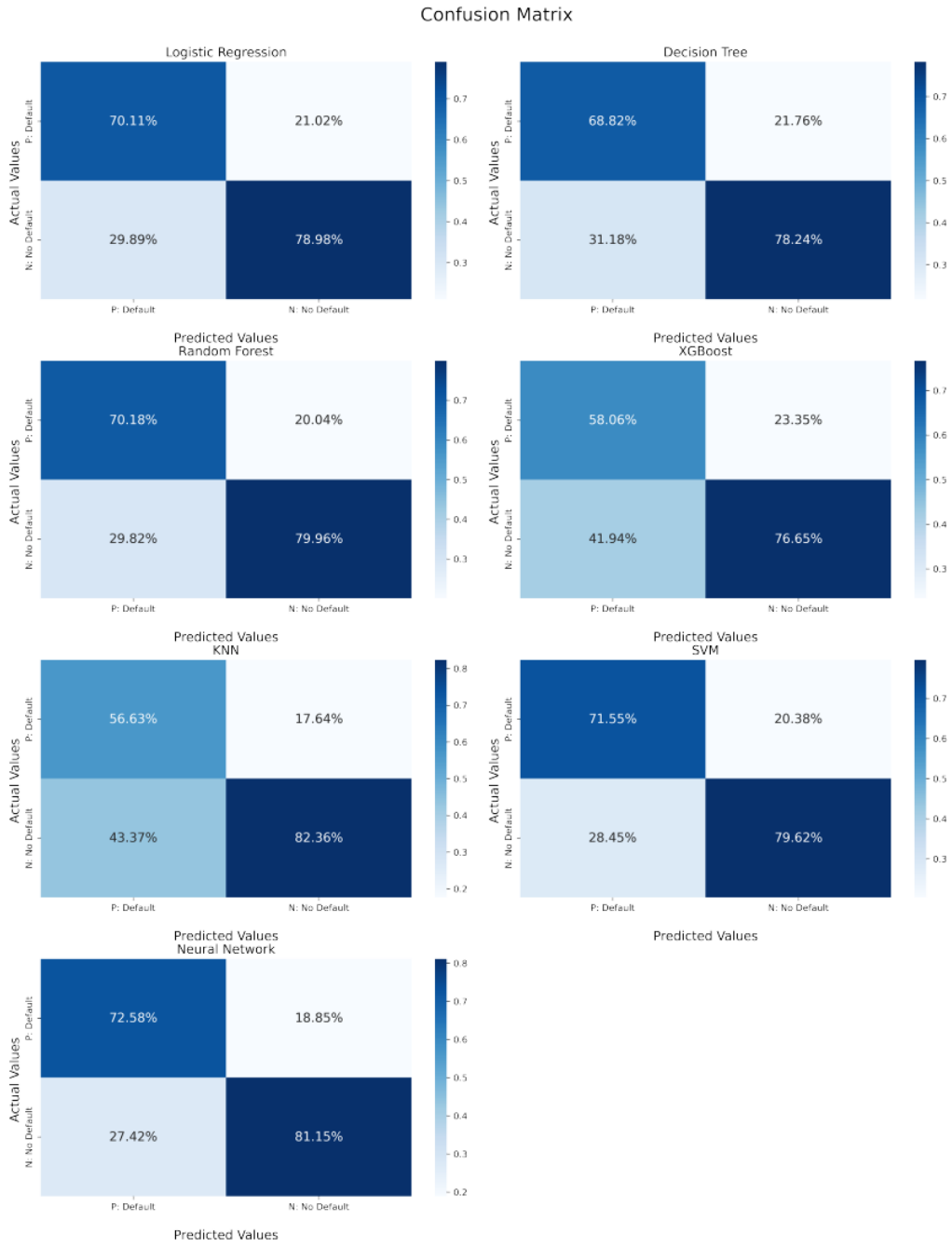

Figure 5.13: The confusion matrices of different models

Chapter 6

Conclusion

6.1 Contributions

In this study, it evaluates the incremental power of five machine learning techniques over Logistic Regression on a set of home loan datasets to determine PD and therefore the default flag based on the optimal threshold for each model. The study follows the guideline of IRB, where the institutions are allowed to use self-developed methods to calculate risk parameters. For all classifiers, the variables undergo the same preprocessing process, including variable transformation by Weight of Evidence and a two-step feature selection process. By this mean, the results become more comparable between the models. The AUC for the best classifier is Artificial Neural Network which is 0.7863, followed by Support Vector Machine and Random Forest. The results of the analysis suggest machine learning techniques have great potential in credit risk modelling. Meanwhile, Logistic Regression still remains as a robust method in PD estimation, that the AUC is only 2% lower than Neural Network. It is not surprising since this model has been applied in the industry for decades, which manifests the strong discriminant power it possesses. However, the machine learning techniques still perform slightly better than LR. These algorithms are experiencing a fast pace development and are particularly useful for handling high-dimensional datasets.

6.2 Limitations and Future Work

Despite the contributions discussed above made by this study, there are still several limitations that exist due to time constraints and data availability. Based on the limitations, some future research directions are proposed.

First, since the goal is to predict 1-year PD, it is better to have monthly or quarterly data for the entire year. However, only six months of data is available for this study, which may not be able to represent the whole picture of one year. Thus, once the data is available, the study needs to be re-conducted on a full-year dataset to verify the results, so that it can meet the requirement outlined by IRB and IFRS9 for calculating 1-year PD.

Second, to deal with the imbalanced data, only one approach is considered and applied to all the models, which is cost-sensitive learning. This is to add more weight to minority data during the training process. However, the models may exhibit different behaviours if other balancing techniques are applied such as oversampling or downsampling. In the future, these techniques can be applied to the data to evaluate the model performances.

Third, the home loan datasets are the sole data used for training and testing the models. Without testing on other datasets, it is difficult to comment on the generalization of the models. Hence, different datasets from the firm or similar datasets from other firms should be used to test how well the models can generalize.

Fourth, the interpretability of some machine learning models, especially the so-called "Black Box" Artificial Neural Networks, remains low which can be a major concern of the regulators. Unlike the traditional models such as the logistic regression which can exhibit a clear relationship between

each explanatory variable and the final output, the machine learning models can hardly build a single link between one variable to the result, but all the variables are taken into account together for the output, as the samples mapped in a high-dimensional feature space. This is an important research topic in academia to uncover the underlying mechanisms of the machine learning models so that they can be more widely applied in practice.

Lastly, due to the data availability and time constraint, we only study the PD prediction problem in this paper. In the future, we can also apply these models to calculate other risk parameters, such as EAD, LGD, SICR, and RR to evaluate the results in the existing literature as well as propose new findings by employing machine learning techniques in the field of credit risk.

Appendix A

Data Description

A.1 Variables

This table lists all the variables contained in the dataset:

Index	Variable Name	Index	Variable Name
1	ExtractDate	28	IndividualVoluntaryArrangementFlag
2	EffectiveDate	29	IVARegisteredDate
3	Asset_ID	30	TermExtension
4	Account_ID	31	LatestMaturityChangedDate
5	OriginationDate	32	TempIOSwitch
6	AdvanceAmount	33	Capitalisation
7	CurrentPoolID	34	Possession
8	CurrentPoolDescription	35	PaymentMethod
9	LegalEntityName	36	InterestOnlyRepVehicle
10	PD_Segment	37	DWPPayer
11	CCF_Segment	38	PropertyRegion
12	Contractual_Maturity	39	PropertyRegion2
13	Default_Flag	40	ProductGrade
14	Default_Date	41	PropertyType
15	ArrearsCounter	42	PropertyType2
16	Collateral_Value	43	CSOFlag
17	OriginalValuation	44	CSODate
18	OriginalValuationDate	45	LGD_Segment
19	Outstanding_Balance	46	Counterparty
20	BalanceAtTerm	47	Origination_Rating
21	Currency	48	Current_Rating
22	Repayment_Type	49	Collateral_Value_AccountID_Level
23	EIR	50	AssetSegment
24	Frequency	51	GroupEntity
25	Limit	52	Maturity
26	Bankruptcy_Flag	53	Months_in_book
27	BankruptcyRegisteredDate	54	Month_maturity_change

Figure A.1: All variables contained in the dataset

This table lists the variables that are removed from the dataset:

Index	Variable Name	Reason for Exclusion
1	ExtractDate	Same for all loans - 01/01/2019
2	CurrentPoolDescription	Not related to default prediction
3	LegalEntityName	Not related to default prediction
4	CCF_Segment	Same for all loans - All
5	Default_Date	Not use default information to predict default
6	Currency	Same for all loans - GBP
7	Frequency	Same for all loans - monthly
8	Limit	Same for all loans - 0
9	Counterparty	Same for all loans - All
10	Origination_Rating	Same for all loans - All
11	Current_Rating	Same for all loans - All
12	OriginalValuationDate	Not related to default prediction
13	BankruptcyRegisteredDate	Use a flag to indicate bankruptcy
14	IVARegisteredDate	Use a flag to indicate Individual Voluntary Arrangement
15	LatestMaturityChangedDate	Not related to default prediction
16	CSODate	Use a flag to indicate CSO
17	InterestOnlyRepVehicle	Has null and blank value
18	Account_ID	Not related to default prediction
19	Contractual_Maturity	Will convert to maturity
20	OriginationDate	Not related to default prediction
21	Asset_ID	Used for identifying loans only
22	ArrearsCounter	Same as default flag, if >= 3 months, default
23	Default_Flag	Not use default information to predict default
24	EffectiveDate	Same for all loans - 31/12/2018

Figure A.2: Dropped variables and the reasons

This table lists the variables that are used during data analysis after removing unnecessary variables and indicates the variable type:

Index	Variable	Type	Description
1	AdvanceAmount	Numerical	The maximum loan amount that a lender is willing to extend.
2	PD_Segment	Categorical	The category based on the default risk level.
3	Collateral_Value	Numerical	The value of the collateral for this loan.
4	OriginalValuation	Numerical	The value of the loan at the origination date.
5	Outstanding_Balance	Numerical	The amount of the loan that has not been paid.
6	BalanceAtTerm	Numerical	The balance at the maturity.
7	Repayment_Type	Categorical	The type for making the payment.
8	EIR	Numerical	Effective interest Rate.
9	Bankruptcy_Flag	Categorical	The indicator of whether the loan holder gets bankrupt.
10	IndividualVoluntaryArrangementFlag	Categorical	The indicator of an agreement with the creditors to pay all or part of the debts.
11	TermExtension	Categorical	The indicator of whether the loan has been extended.
12	TempIOSwitch	Categorical	The indicator of whether the loan is interest-only.
13	Capitalisation	Categorical	The indicator of whether the interest is capitalised on the loan.
14	Possession	Categorical	The indicator of whether the buyer takes ownership of a property after signing closing documents.
15	PaymentMethod	Categorical	The ways of making the payment.
16	DWPPayer	Categorical	The indicator of whether the loan is from the the Department of Work and Pensions.
17	PropertyRegion	Categorical	The region of the property located based on the location.
18	PropertyRegion2	Categorical	The region of the property located based on the city.
19	ProductGrade	Categorical	The grade of the home loan.
20	PropertyType	Categorical	The type of the property based on the use.
21	PropertyType2	Categorical	The type of the property based on the building type.
22	CSOFlag	Categorical	The indicator of Credit Services Organizations loans, which are installment loans originated by independent third party lenders.
23	LGD_Segment	Categorical	The category based on different LGD value.
24	Collateral_Value_AccountID_Level	Numerical	The value of the collateral for loans in the same account.
25	AssetSegment	Categorical	The category based on the asset.
26	GroupEntity	Categorical	The entity which sells the loan.
27	Maturity	Numerical	The time difference in month between the effective date and the maturity date.
28	Months_in_book	Numerical	The time difference in month between the effective date and the origination date.
29	CurrentPoolID	Categorical	The ID of the entities that buy the mortgage pool which is a group of loans bundled for selling.
30	Month_maturity_change	Numerical	The time difference in month between the effective date and the maturity change date.

Figure A.3: Used variables with their types and descriptions

The histograms show the distributions of all numerical variables:

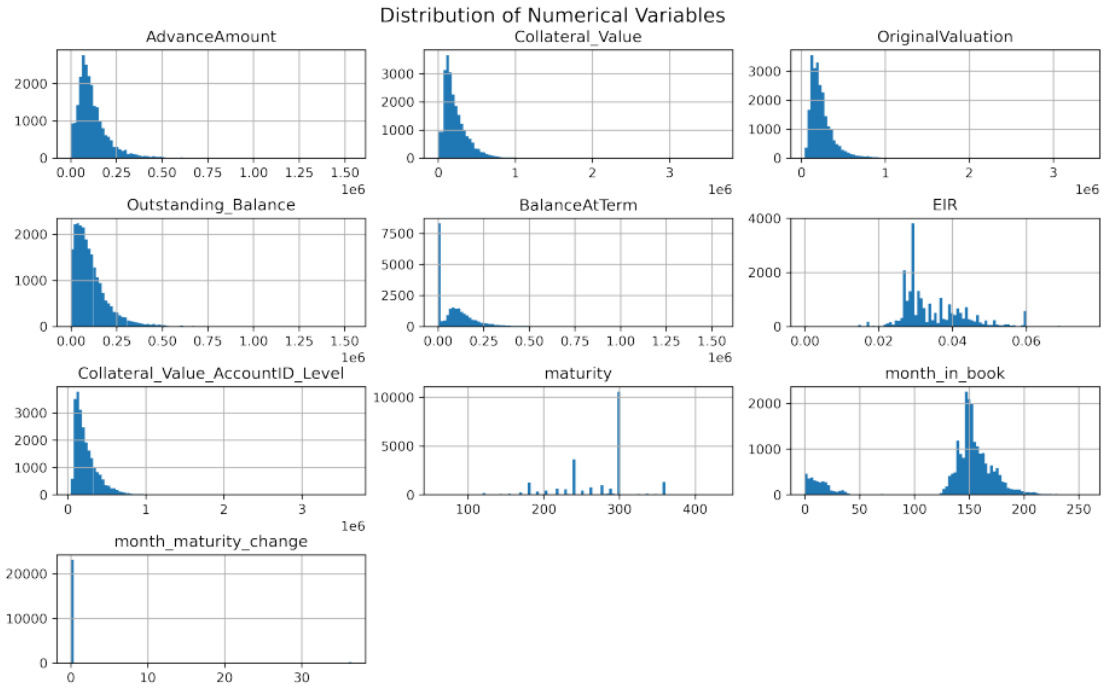

Figure A.4: Distributions of all numerical variables

The histograms show the distributions of all categorical variables:

Figure A.5: Distributions of all categorical variables

A.2 Correlation

The correlations among all the variables are plotted below:

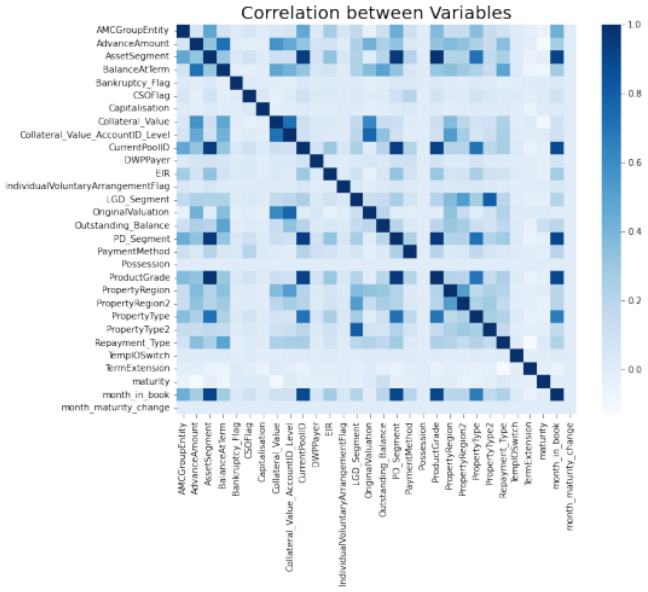

Figure A.6: Correlations among all the variables

The correlations among the variables with IV greater than 0.1 are plotted below:

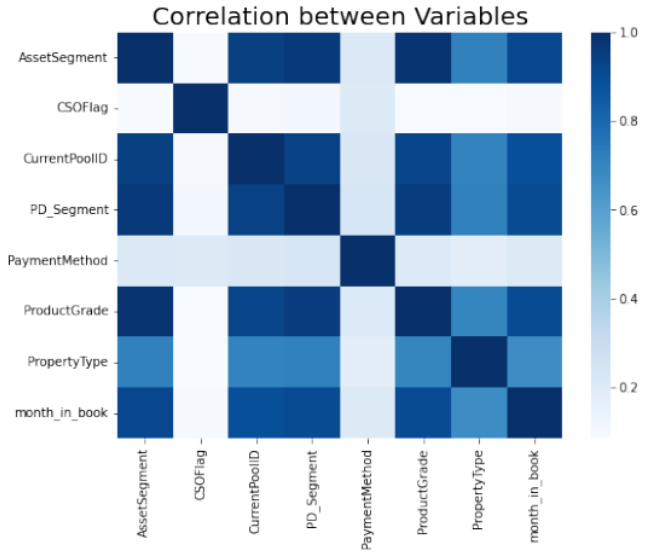

Figure A.7: Correlations among the variables with IV greater than 0.1

Bibliography

[1] A. Bellotti, D. Brigo, P. Gambetti, and F. Vrins, "Forecasting recovery rates on non-performing loans with machine learning," *International Journal of Forecasting*, vol. 37, p. 428–444, Aug 2019.

[2] J. Mongelard, "A brief history in credit risk in banking."

[3] A. Matre, *Machine Learning in Default Prediction*. PhD thesis, 2019.

[4] *Financial soundness indicators: Compilation guide*. International Monetary Fund, 2006.

[5] "Iasb completes reform of financial instruments accounting."

[6] J. Suarez, C. Laux, and S. Lind. 2017.

[7] T. Segal, "Nonperforming loan (npl)," May 2022.

[8] "Guidance to banks on non-performing loans - europa," Mar 2017.

[9] E. Altman, A. Resti, and A. Sironi, "Default recovery rates in credit risk modelling: A review of the literature and empirical evidence," *Economic Notes*, vol. 33, no. 2, p. 183–208, 2004.

[10] Y. Liu, M. Yang, Y. Wang, Y. Li, T. Xiong, and A. Li, "Applying machine learning algorithms to predict default probability in the online credit market: Evidence from china," *International Review of Financial Analysis*, vol. 79, p. 101971, 2022.

[11] A. Petropoulos, V. Siakoulis, E. Stavroulakis, and A. Klamargias. 2018.

[12] A. Stelzer, "Predicting credit default probabilities using machine learning techniques in the face of unequal class distributions," *Econometrics*, Jul 2019.

[13] S. Cheng, *A Machine Learning Based Bond Rating Prediction System Facilitating Investment Decision Making*. PhD thesis, 2020.

[14] Z. Dai, Z. Yuchen, A. Li, and G. Qian, "The application of machine learning in bank credit rating prediction and risk assessment," *2021 IEEE 2nd International Conference on Big Data, Artificial Intelligence and Internet of Things Engineering (ICBAIE)*, 2021.

[15] Y. B. Wah, H. A. Rahman, H. He, and A. Bulgiba, "Handling imbalanced dataset using svm and k-nn approach," *AIP Conference Proceedings*, 2016.

[16] S. Remanan, "Logistic regression: A simplified approach using python," Sep 2018.

[17] Y. Yang, *ESTIMATING AND EVALUATING THE PROBABILITY OF DEFAULT - A MACHINE LEARNING APPROACH*. PhD thesis, 2021.

[18] J. Kim, "Overview random forest," Oct 2018.

[19] N. Kumar, "Advantages and disadvantages of knn algorithm in machine learning."

[20] "Support-vector machine," Aug 2022.

[21] R. Gandhi, "Support vector machine - introduction to machine learning algorithms," Jul 2018.

[22] S. Theodoridis and K. Koutroumbas, p. 203. Academic Press, 2009.

[23] J. Chen, "Neural network definition," Jun 2022.

[24] "Neural network," Aug 2022.

[25] K. Sarkar, "Relu: Not a differentiable function: Why used in gradient based optimization," May 2018.

[26] J. Mahanta, "Introduction to neural networks, advantages and applications," Jul 2017.

[27] N. Siddiqi, *Credit risk scorecards: Developing and Implementing Intelligent Credit scoring*. Wiley, 2006.

[28] A. Hayes, "Stepwise regression," Jun 2022.

[29] G. Smith, "Step away from stepwise," *Journal of Big Data*, vol. 5, no. 1, 2018.

[30] M. Kuhn and K. Johnson, *11.3 Recursive Feature Elimination*. Chapman amp; Hall/CRC, 2021.

[31] M. Setia, *Log Loss Function*. Analytics Vidhya, Nov 2020.

[32] N. Shrivastav, "Confusion matric(tpr,fpr,fnr,tnr), precision, recall, f1-score," Feb 2021.

[33] "Receiver operating characteristic," Jul 2022.

[34] A. Bhandari, "Auc-roc curve in machine learning clearly explained," Jun 2022.

[35] "scikit-learn," Aug 2022.