JOURNAL OF STATISTICAL MODELING & ANALYTICS (JOSMA)
(ISSN: 2180-3102)
UNIVERSITI MALAYA

Comparative Analysis of Machine Learning Models for Vintage-Based Credit Scoring

Tan Yong Seng[1*] and Soo Huei Ching[2]

[1,2] School of Mathematical and Computer Sciences, Heriot-Watt University Malaysia, 62200 Putrajaya, Malaysia

[*]Corresponding author: yt2033@hw.ac.uk

RESEARCH ARTICLE

Received 6 August 2025
Accepted 18 Nov 2025
Published 26 Dec 2025

Abstract

Accurate credit risk assessment is crucial for financial institutions to minimise loan defaults. This study proposes a vintage-based credit scoring framework that integrates individual repayment behaviour with vintage analysis and evaluates five machine learning models, including logistic regression, random forest, XGBoost, stacking ensemble, and multilayer perceptron (MLP), for binary credit risk classification. Results show that ensemble methods, particularly random forest, achieve superior predictive performance with the highest F1-score (0.81), precision (0.87) and accuracy (0.96), while logistic regression exhibits high recall but low precision. The MLP shows good recall (0.79) and a competitive F1-score (0.77), making it suitable for prioritising high-risk borrower detection, although it lacks interpretability. Overall, the study highlights the trade-offs between predictive performance and interpretability, emphasising the potential of vintage-based approaches and ensemble learning for practical credit scoring applications.

Keywords: Binary Classification, Credit Risk Scoring, Ensemble Learning, Machine Learning, Vintage Analysis

1. Introduction

Financial institutions rely on credit assessments to evaluate borrowers' creditworthiness and minimise potential lending losses. Credit card balances, as a form of revolving credit, provide valuable information on repayment behaviour, which can be used to assess delinquency risk.

A recent survey by a Malaysia private university, UCSI, indicates that 73% of 1,077 Malaysians between the ages of 18 and 40 are in debt, primarily due to car loans (Soo, 2023). This statistic underscores the prevalence of loan applications and highlights the urgent need for accurate credit assessment systems that can reliably evaluate borrowers' repayment capacity. As the lending environment becomes increasingly complex with the rise of big data, the development of efficient and accurate credit scoring methods is critical for the financial sector. The ability to differentiate between low-risk and high-risk borrowers is fundamental to responsible lending and effective risk management.

In response to these challenges, financial institutions have increasingly adopted machine learning techniques for credit risk assessment. According to Nallakaruppan et al. (2023), logistic regression remains the most widely used model, followed by decision trees, neural networks, and other machine learning techniques. While these models offer valuable insights, each has its limitations: logistic regression struggles to capture non-linear relationships; decision trees are prone to overfitting; and neural networks though powerful, often lack interpretability due to their black-box nature.

This study explores ensemble learning techniques, which combine multiple estimators to improve generalisation and prediction accuracy and compares their performance with logistic regression and neural networks. These ensemble techniques are evaluated against traditional logistic regression model in terms of prediction performance, model complexity, and training cost.

Supervised machine learning relies on labelled datasets that include both the target variable and explanatory features. However, in credit scoring contexts, such datasets are not always readily available, particularly due to the lack of a universally accepted definition of a "bad" borrower. In this study, the only observable indicator of borrower behaviour is loan repayment status, which is used as a proxy for creditworthiness. Therefore, a systematic approach is needed to classify borrowers based on their repayment performance, enabling the effective training of supervised models.

To address this, vintage analysis, a method originally used to evaluate wine quality over time, is adapted in this study to monitor loan portfolio performance (Stanimir, 2011). By grouping loans into cohorts, referred to as vintages, based on their origination period, this technique facilitates the tracking of repayment behaviour over time. It enables the identification of repayment trends, assessment of credit risk, and systematic labelling of borrowers as "good" or "bad".

This study aims to develop a robust framework for labelling borrowers based on repayment behaviour and to evaluate the performance of several machine learning models for binary classification. The specific objectives are:

i. To conduct vintage analysis of loan repayment status to gain insights into each vintage's performance.
ii. To develop a method for labelling borrowers as "good" or "bad" using vintage analysis and classify them using machine learning models.
iii. To compare model performance and identify the most suitable method for credit risk assessment.

By achieving these objectives, the study provides practical tools and analytical insights to support financial institutions in making informed lending decisions while minimising credit risk.

The structure of this paper is as follows. Section 2 reviews relevant literature to establish the foundation of the study. Section 3 describes the data source. Section 4 presents the methodology for constructing the vintage-based credit scoring system and the architecture of the machine learning models. Section 5 outlines the experimental setup, encompassing data preprocessing, partitioning strategies, model validation procedures, and performance evaluation metrics. Section 6 reports the results and discussion. Section 7 concludes the study and Section 8 highlights potential directions for future research.

2. Literature Review

2.1 Anjani's Credit Card Approval Framework and Evaluation

Anjani et al. (2023) proposed a framework for building a credit card approval system, including data preparation, model development, and performance evaluation. Analysing the cumulative distribution of account statuses revealed time-dependent credit risk patterns. Tree-based models, especially XGBoost, achieved 90.06% accuracy. However, accuracy alone can be misleading in the presence of class imbalance. Despite high accuracy, the model's AUC was poor, highlighting the limitations of accuracy when the majority class dominates. This underscores the importance of appropriate evaluation metrics for reliable model assessment.

2.2 Insights into Credit Score Construction

Defining an appropriate target variable is critical for training supervised machine learning models. However, this study faces a challenge due to the absence of a clearly defined outcome label in the dataset. One of the most widely used credit scoring systems is FICO, developed by the Fair Isaac Corporation. It serves as a standard risk indicator to support lenders in making loan approval decisions. FICO scores are used by 90% of top lenders, as they summarise a borrower's credit history using a scientifically derived score that supports informed decision-making. According to Gorman (2025), the FICO score is computed based on five key components of a borrower's credit report:

i. **Payment History (35%)**
This is the most significant factor, assessing whether the borrower has consistently made payments on time.

ii. **Amount Owed (30%)**
This measures the ratio of total outstanding debt to available credit limits. A lower ratio indicates responsible credit use and contributes positively to the score.

iii. **Length of Credit History (15%)**
This evaluates how long each credit account has been active. A longer credit history generally contributes positively to the score.

iv. **Credit Mix (10%)**
This factor evaluates the variety of credit types in the borrower's profile. A diverse mix of credit accounts is typically viewed positively.

v. **New Credit (10%)**
This examines the number of newly opened credit accounts. Frequent applications for new credit may signal increased credit risk.

2.3 Methodology for Constructing the Target Variable

A key limitation in the dataset used in this study is that only two variables, repayment history and length of credit history, are available to evaluate borrower performance. Consequently, an alternative approach is required to address the constraint imposed by limited data availability. This study adopts vintage analysis as a practical and robust method for managing credit risk under such conditions. The term *vintage*, originally from the wine industry where it denotes the year grapes are harvested, it refers to loan age in finance (Stanimir, 2011). In this context, vintage analysis leverages the borrower's days past due (DPD) to classify credit risk. This approach aligns repayment behaviour with risk segmentation, providing a structured and analytical foundation for effectively labelling borrowers.

2.4 Review of Ensemble Learning

Ensemble learning is a technique combines multiple weak learners to create a robust model with enhanced predictive performance (Hastie et al., 2009). This approach has proven effective in various machine learning competitions, including classification and regression tasks. A notable example is the Kaggle *Home Credit Default Risk* competition, where the winning team implemented a three-level stacking ensemble model (Tunguz, 2018). The first layer comprised numerous base models; the second layer used stackers such as a neural network, extra trees, and a hill climber to combine those predictions; and the final blending layer (level 3) employed the same algorithms to generate the final output. This approach achieved an impressive AUC of 0.8057, demonstrating the effectiveness of ensemble learning.

2.5 Review of Multilayer Perceptron (MLP)

Alsmadi et al. (2009) discussed the theoretical foundations of multilayer perceptron (MLP) algorithms in deep learning and provided practical guidance for constructing neural networks. The MLP extends the basic perceptron model by incorporating at least three layers (input, hidden, and output) and activation functions, enabling them to capture complex relationships in data. The backpropagation algorithm is highly effective for training MLPs, iteratively adjusting weights to minimise prediction error. However, the effectiveness of an MLP depends on sufficient training data and an appropriately configured hidden layer. An undersized network may fail to detect patterns, while an oversized one increases the risk of overfitting.

2.6 Summary of the Literature Review

The reviewed literature addresses various aspects of credit scoring and machine learning techniques relevant to this study. Anjani's (2023) credit card approval framework outlines the procedures for data preparation, model development, and performance evaluation, highlighting the strengths of tree-based models such as extreme gradient boosting (XGBoost). In terms of credit score construction, FICO scores are widely adopted by lenders due to their systematic incorporation of repayment history, credit utilisation, and credit mix. In situations where detailed credit data is unavailable, vintage analysis offers a practical alternative by classifying borrower risk based on repayment patterns over time. Ensemble learning methods, including random forest and XGBoost, consistently demonstrate superior predictive performance by combining multiple weak learners into a more robust model. Similarly, multilayer perceptron (MLP) can capture complex non-linear relationships in data. However, their effectiveness depends heavily on adequate training data and well-designed network architecture to avoid underfitting or overfitting. Overall, these insights guide the development of the vintage-based credit scoring framework in this study.

3. Data Source

The dataset used in this study was obtained from Kaggle's *Credit Card Approval Prediction* (2020) dataset. Although originally intended for approval prediction, it contains sufficient temporal and behavioural information to support vintage-based credit risk scoring through feature engineering and outcome labelling. Given that credit card repayments operate as short-term revolving loans, analysing repayment behaviour aligns with the evaluation of loan repayment risk. The dataset comprises two main files: application_record.csv and credit_record.csv.

- application_record.csv provides demographic and socio-economic attributes for each applicant, including ID, gender, car ownership, property ownership, number of children, annual income, income category, education level, marital status, housing type, date of birth, employment duration, mobile phone ownership, phone presence, email availability, occupation type, and family size.
- credit_record.csv captures the monthly repayment behaviour for each borrower, including ID, MONTHS_BALANCE (a relative time index of the repayment records), and STATUS (repayment status codes ranging from on-time to severe delinquency).

The two files were merged using the unique identifier ID, resulting in a combined dataset containing 25,129 observations with 16 features. Borrowers were labelled as "good" (0) or "bad" (1) based on their repayment patterns, as described in Section 4. The final dataset exhibits class imbalance, with 89.57% of borrowers classified as "good" and 10.43% as "bad". This imbalance was addressed during model training, as discussed in Section 5. Missing values were identified, and any records with critical missing data were either imputed or removed to ensure data quality.

3.1 Data Preprocessing

Preprocessing was conducted to ensure that the dataset was clean, structured, and suitable for modelling. The steps applied to each dataset are outlined below.

i. Preprocessing for application_record.csv

This file contains personal and demographic variables for each borrower. The variable types and corresponding preprocessing methods are summarised in Table 1.

Table 1. Overview of variable types and preprocessing methods for application_record.csv.

Variable Type	Variables	Preprocessing Method
Binary	Gender, car ownership, property ownership, mobile phone ownership, telephone ownership, email availability	Encoded as 0 or 1
Nominal	Income category, marital status, housing type, occupation type	One-hot encoding
Ordinal	Education level	Label encoding
Numerical (transformed)	Date of birth (converted to age), days employed (converted to years of employment)	Numerical transformation

ii. Preprocessing for credit_record.csv

This file records the monthly repayment behaviour of each borrower. Several features were engineered to support model training. These refinements are summarised in Table 2.

Table 2. Feature engineering and refinement for credit_record.csv.

Feature	Description
MONTHS_BALANCE	The values were adjusted to span from 0 to 60 months ago, representing the relative timeline of repayment records. The earliest three months (0, 1, 2) were excluded to reduce noise and enhance data quality.
GOOD_MONTHS	A newly derived feature indicating the number of months with good repayment status (number of STATUS C and 0) for each borrower.
ORIGINATION_MONTHS	Estimated using MONTHS_BALANCE and ID to approximate the loan origination period for each borrower.

4. Research Methodology

4.1 Defining the Target Variable

Vintage analysis is recognised as a valuable tool for measuring credit risk. It tracks the performance of a loan portfolio over time (Stanimir, 2011), using the number of days past due (DPD) as the primary indicator of risk. This approach enables monitoring of a portfolio’s credit quality and helps identify patterns in borrower behaviour, offering a summary of potential risks and performance trends within the dataset. An illustrative example of the credit history for a borrower whose loan originated in MONTH 3 is shown in Table 3.

Table 3. Recorded loan performance of a borrower.

STATUS	C	C	0	C	C	0	1	2	C	X
MONTH	3	4	5	6	7	8	9	10	11	12

MONTH represents the time frame over which the loan's performance is tracked, beginning from MONTH 3. The STATUS variable indicates the repayment condition of the loan, coded according to the number of days the repayment is overdue. There are eight unique STATUS codes in the dataset, as summarised in Table 4.

Table 4. Definitions of STATUS code levels.

STATUS	C	X	0	1	2	3	4	5
Description	Paid off on time	No active loan	1-29 DPD	30-59 DPD	60-89 DPD	90-119 DPD	120-149 DPD	≥ 150 DPD

As shown in Figure 1, clear trends emerge in the distribution of loan statuses over time. Specifically, the proportion of STATUS 0 (indicating slightly late payments) declines, while STATUS C (representing on-time payments) increases. The remaining STATUS categories do not show significant change throughout the observation period. This trend suggests a reduction in credit risk over time, as evidenced by the decreasing frequency of STATUS 0 and the concurrent rise in STATUS C. These temporal patterns form the basis for developing a vintage-based credit scoring framework.

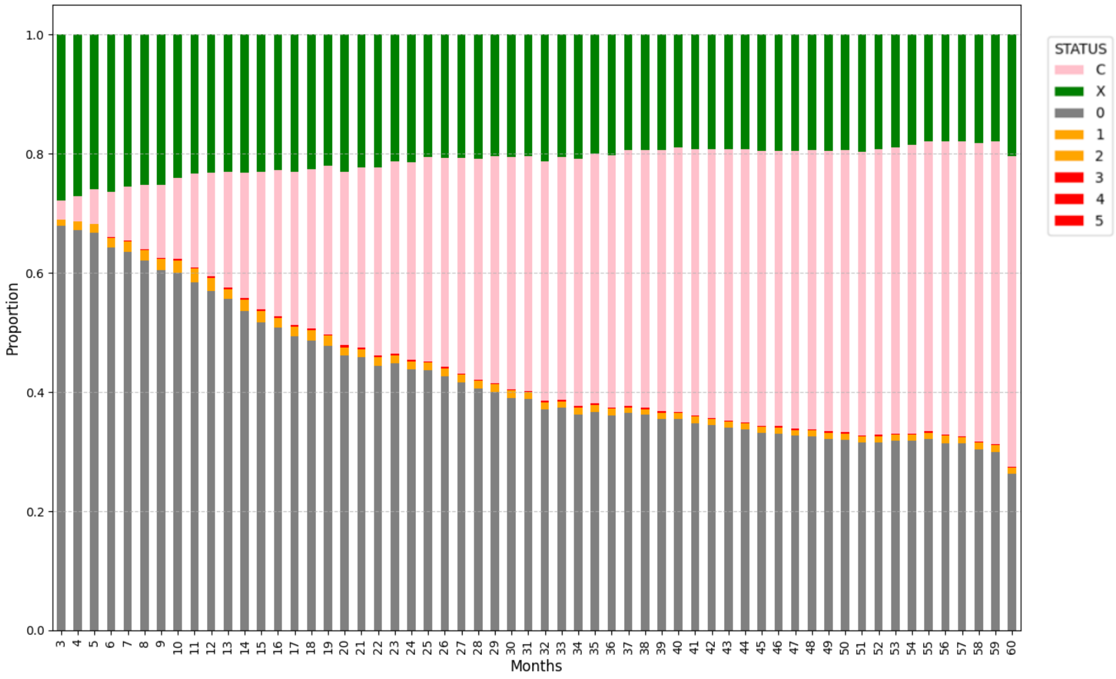

Figure 1. Monthly distribution of loan STATUS codes from MONTH 3 to Month 60.

4.1 Vintage-Based Credit Risk Evaluation

As observed in Figure 1, earlier loan vintages are generally associated with higher credit risk. Loans originated in earlier periods (i.e., with lower MONTH values) exhibit a higher proportion of STATUS 0, which corresponds to minor delays in repayment. In contrast, more recent vintages tend to show an increasing frequency of STATUS C, indicative of consistent and timely repayment behaviour. This temporal pattern suggests the tendency for risk to diminish as loan portfolios mature.

Assuming the proportion of STATUS 0 in each vintage serves as the risk indicator, its declining trend over time can be well approximated by an exponential decay model of the form:

$$f(x) = a \cdot e^{-bx} + c \tag{1}$$

where

a denotes the scale factor,
b is the decay rate,
c represents the baseline offset, and
x corresponds to the MONTH variable.

The parameters of Eq. (1) were estimated using non-linear least squares fitting via the scipy.optimize.curve_fit function, based on the observed frequency of STATUS 0 across different vintages. The resulting fitted curve is shown in Figure 2, with the equation: $f(x) = 0.498e^{-0.044x} + 0.263$.

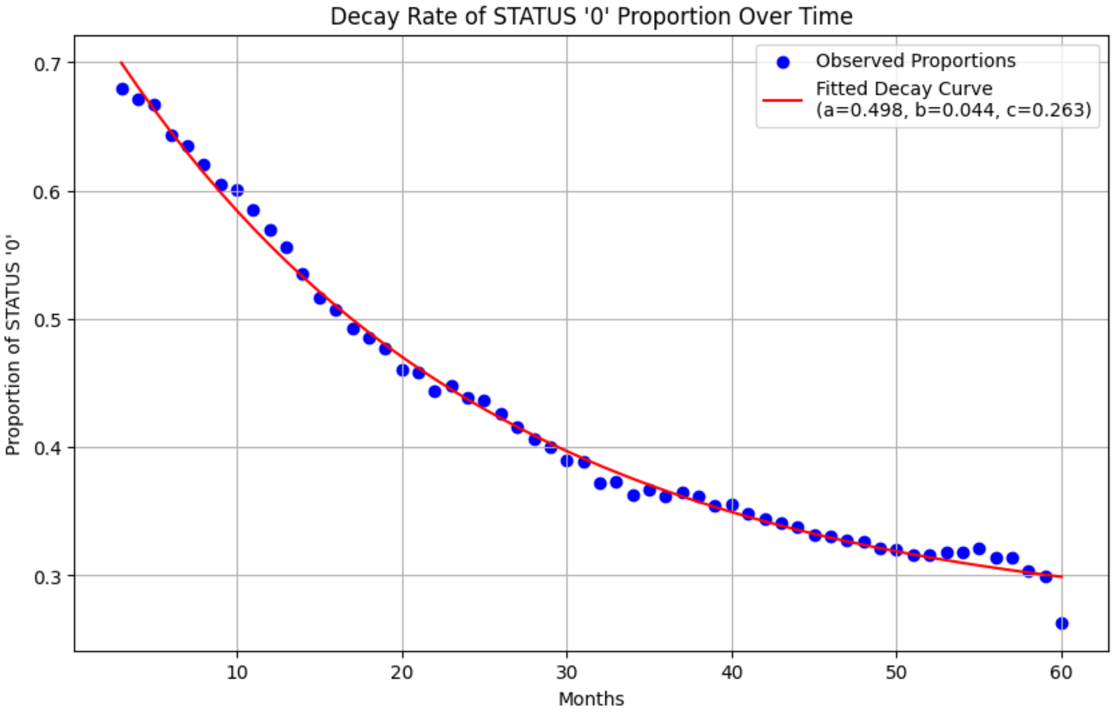

Figure 2. Exponential decay curve fitted to the frequency of STATUS 0 across vintages.

To quantify credit risk across vintages, the **Vintage Risk** at MONTH x, denoted $R(x)$, is defined by scaling the fitted decay function as follows:

$$R(x) = F \cdot f(x) = 43 \cdot (0.498e^{-0.044x} + 0.263) \tag{2}$$

Here, factor F = 43 is chosen such that $R(3)$ = 30, aligning with the study's target of achieving a 10% "bad" borrower ratio. It is important to note that MONTH = 3 marks the beginning of the available loan performance records in the dataset. As x increases, $R(x)$ decreases, indicating that newer loans are associated with lower credit risk.

4.2 Credit Risk Implications by DPD Classification

Understanding the consequences of late repayments is essential in developing a credit scoring system based on actual loan performance. When borrowers fail to meet payment obligations, their creditworthiness declines, resulting in adverse outcomes such as lower credit scores, increased borrowing costs, stricter loan conditions, and greater difficulty securing future credit.

According to Experian (Axelton, 2025), the consequences associated with different days past due (DPD) categories reflected in the STATUS codes are as follows:

1. **1 to 29 DPD (STATUS 0)**
   This level of delinquency generally does not affect the borrower's credit score, as lenders typically do not report payments that are only slightly late. If repayment occurs within this period, penalties to the credit score can usually be avoided.
2. **30 to 89 DPD (STATUS 1 and 2)**
   At this stage, the account is considered delinquent, and lenders begin reporting the missed payments in 30-day increments. This leads to a gradual but measurable decline in credit scores.
3. **90 to 119 DPD (STATUS 3)**
   The account is regarded as being in default, reflecting a more serious breach of the repayment agreement and a significant deterioration in the borrower's credit profile.
4. **120 or more DPD (STATUS 4 and 5)**
   This stage indicates severe delinquency. The lender may charge off the account, classifying the debt as uncollectible and potentially selling it to a third-party debt collector. Borrowers in this category are usually flagged with a serious derogatory mark on their credit report.

4.2 Development of a Vintage-Based Credit Scoring System

To develop an effective credit scoring system, this study incorporates temporal repayment behaviour using vintage analysis, in which different late-payment statuses are assigned distinct weights. Drawing insights from observed trends in loan performance over time and the associated consequences of various delinquency levels, a vintage-based credit risk model is proposed. The framework is underpinned by the following assumptions:

- Although the dataset captures credit card repayments, it is treated as representative of general revolving loan performance for credit risk assessment.
- Loans originated earlier (i.e., with lower MONTH values) are assumed to carry a higher inherent risk of default.
- STATUS 0 (slightly late payment) is viewed as an early sign of financial instability and a potential precursor to default.
- Accounts in STATUS 4 or 5 are considered to represent permanent default, with minimal likelihood of repayment recovery.

4.2.1 Borrower Classification Criteria

Each borrower is assigned a **Vintage Risk** based on the month in which their loan originated. All borrowers begin with an **initial score of 100**, providing a consistent reference point for scoring. This initial score allows for clear differentiation between:

- Borrowers from earlier vintages, who inherently carry higher risk; and
- Borrowers with no repayment activity, reflecting growing uncertainty over time.

Points are then added or deducted from this baseline according to the frequency of each repayment STATUS in the borrower's loan history. Each repayment STATUS is mapped to a specific score that contributes to the borrower's final credit risk score and shown in Table 5.

Table 5. Repayment status and corresponding scores.

Repayment status	**Description**	**Assigned score**
C	On-time payment	+10
X	No active loan for the month	0
0	1-29 days past due	+5
1	30-59 days past due	-15
2	60-89 days past due	-15
3	90-119 days past due	-50
4	120-149 days past due	-1000
5	150 days past due or more	-1000

The final score for each borrower is computed using the following formula:

$$\text{Final Score} = \text{Initial Score} - \text{Vintage Risk} + \sum_x^n \text{Status Score}_x \qquad (3)$$

where Status Score_x represents the score assigned based on the borrower's repayment status in month x, and n is the total number of observed months. The rationale for the assigned scores is discussed in Section 4.2.4.

4.2.2 Score Calculation Example

Table 6 illustrates a borrower's STATUS history and the corresponding assigned scores.

Table 6. Example of a borrower's performance record with assigned scores.

STATUS	C	C	0	C	C	0	1	2	C	X
MONTH, x	3	4	5	6	7	8	9	10	11	12
Score	10	10	5	10	10	5	-15	-15	10	0

The total Status Score for this borrower is:

$$\sum_x^n Status\ Score_x = 10 + 10 + 5 + 10 + 10 + 5 - 15 - 15 + 10 + 0 = 30$$

Assuming the loan originated in MONTH = 3, the Vintage Risk as defined in Eq. (2), is:

$$R(3) = 43 \times (0.498e^{-0.044 \times 3} + 0.263) = 30$$

Using Eq. (3), the final score is:

$$Final\ Score = 100 - 30 + 30 = 100$$

4.2.3 Borrower Classification Threshold

A classification threshold is set to categorise borrowers as “good” or “bad”. A lower threshold results in small proportion of “bad” borrowers, while a higher threshold value classifies more borrowers as “bad”, enabling stricter risk management. In this study, the threshold is set at 87, such that the bottom 10% of borrowers are classified as “bad”:

- **Final Score > 87** → “good” borrower
- **Final Score ≤ 87** → “bad” borrower

In the example above, the final score is 100, which exceeds the threshold. Hence, the borrower is classified as a “good” borrower.

4.2.4 Status Scoring Methodology

The reasoning behind the scores assigned to each STATUS code is summarised as follows:

- **C (On-Time Payment)**
  Indicates timely repayment, a responsible borrowing behaviour. A positive score of **+10** is awarded to reward punctual payments.
- **X (No Loan)**
  Indicates no active loan during the month; therefore, no points are added or deducted.
- **0 (Slightly Late Payment)**
  Reflects a delay of fewer than 30 days. While such delays are generally not reported to credit bureaus, a small positive score of **+5** is given to acknowledge partial risk while recognising eventual repayment.
- **1 and 2 (Delinquent Payments)**
  Represent payments overdue by 30–89 days. A penalty of **–15** points is applied per month to indicate increasing risk exposure.
- **3 (Default)**
  Signals severe delinquency typically considered a default. A penalty of **–50** points is applied to reflect the borrower’s deteriorating credit profile.
- **4 and 5 (Charge-Off)**
  Indicate critical delinquency where the loan is unlikely to be recovered. A severe penalty of **–1000** points is imposed to reflect presumed loss.

Clarification on Status 0 Scoring

While most repayment statuses directly reflect the severity of delinquency, Status 0 (1–29 days past due) requires clarification. Although it may indicate early signs of payment difficulty, a small positive score (+5) is assigned to maintain a balanced risk–reward trade-off.

Empirical analysis shows that treating Status 0 as mildly positive improves classification accuracy by better separating “good” and “bad” borrowers. In contrast, assigning a zero or negative score increases overlap among borderline cases, thereby reducing predictive accuracy.

As a vintage-based risk penalty is already applied to all borrowers, the small positive score accommodates brief delays. Fully penalising minor delays could exclude reliable borrowers, potentially lowering loan approval rates and revenue. The +5 allocation therefore balances predictive performance, business objectives, and risk management.

4.2.5 Advantages of the Vintage-Based Scoring Framework

This credit scoring system provides an alternative way for classifying borrowers based solely on observed repayment behaviour. It is particularly effective when only limited data are available. The key advantages include:

1. **Risk-Adjusted Penalties**
   The model penalises borrowers based on the origination date of their loans, incorporating the inherent risk associated with earlier vintages.
2. **Tolerance for Minor Delinquencies**
   Borrowers with a strong history of on-time payments can maintain a good score despite isolated delinquencies, offering a more balanced and fair assessment.
3. **Customisable Classification Threshold**
   Lenders can adjust the threshold value based on their risk tolerance or strategic goals, enabling flexible credit decision-making.

4.3 Logistic Regression

Logistic regression is a widely adopted statistical technique for modelling binary outcome variables. Its primary objective is to characterise the relationship between a dichotomous dependent variable and a set of explanatory variables using the logistic (sigmoid) function. Unlike linear regression, which predicts continuous outcomes, logistic regression estimates the probability that a given observation belongs to a particular class (Class 0 or Class 1 in this study). Due to its simplicity, interpretability, and effectiveness, logistic regression frequently serves as a baseline model for evaluating the performance of more complex classification algorithms.

The model assumes that the log-odds of the probability p_i, representing the likelihood of observation *i* belonging to Class 1 (i.e. $p_i = P(Y = 1|x_i)$), can be expressed as a linear combination of the predictors, as shown in Eq. (4):

$$\text{logit}\ (p_i) = \ln\left(\frac{p_i}{1-p_i}\right) = \beta_0 + \sum_{j=1}^{k} \beta_j\, x_{ij} \tag{4}$$

where x_{ij} is the value of the j-th explanatory variable for the i-th observation, β_j is the corresponding coefficient, and β_0 is the intercept term.

Figure 3 shows the sigmoid function, which corresponds to the inverse of the logit transformation. Applying this inverse yields the probability p_i as follows:

$$p_i = \frac{1}{1 + \exp\left(-\left[\beta_0 + \sum_{j=1}^{k} \beta_j x_{ij}\right]\right)} \tag{5}$$

The sigmoid function maps any real-valued input into the interval [0, 1], allowing for a probabilistic interpretation of class membership. Logistic regression was implemented as a baseline model under its standard assumptions of linearity in the log-odds, independence of observations, and low multicollinearity among predictors. A default classification threshold of 0.5 was applied to the predicted probability p_i to distinguish between good (Class 0) and bad (Class 1) borrowers, although this threshold may be adjusted to align with specific risk management objectives.

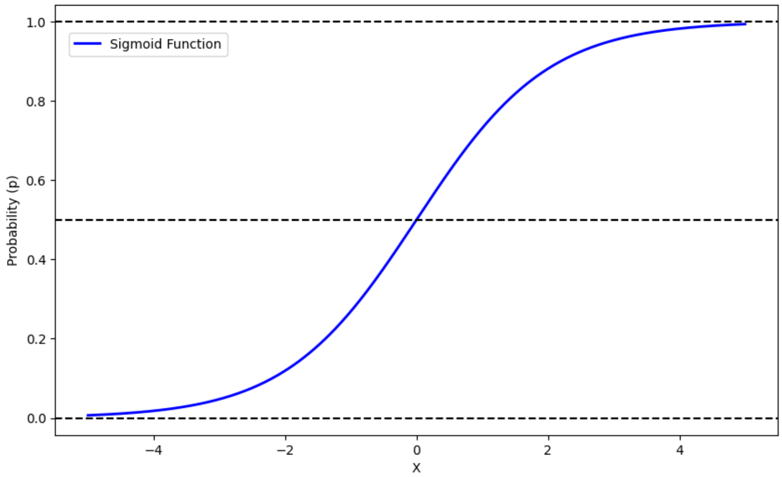

Figure 3. The sigmoid function.

4.4 Decision Tree

Decision tree is a fundamental tree-based model commonly employed in classification tasks due to its simplicity, interpretability, and ability to handle both numerical and categorical variables. In binary classification, decision trees recursively partition the input space, assigning each region to one of the two classes based on feature values. A decision tree model consists of a root node, internal nodes, and leaf nodes. The root node represents the initial split, internal nodes further divide the data, and leaf nodes assign a class label to observations.

Figure 4 illustrates a simplified binary decision tree diagram, highlighting the hierarchical structure of root, internal, and leaf nodes. Each split is determined by a threshold that aims to increase node purity, guiding the classification process toward the final decision outcomes at the leaf nodes. The algorithm only considers features that contribute to reducing impurity, thereby ensuring that parent nodes are split into more homogeneous (purer) child nodes (Song & Lu, 2015).

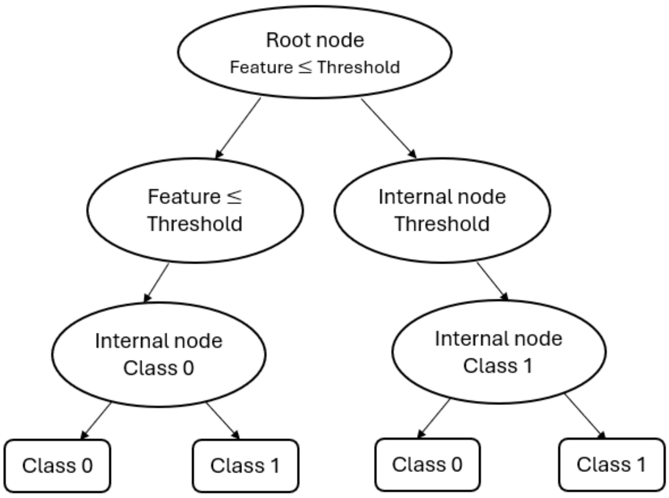

Figure 4. Structure of a binary decision tree for classification.

4.4.1 Splitting criterion

At each internal node, the decision tree algorithm selects the optimal feature and corresponding threshold that best partition the dataset, with the objective of maximising the *purity* of the resulting child nodes. One of the most commonly used impurity measures is the **Gini impurity** (Yuan, Wu, & Zhang, 2021), defined as:

$$\text{Gini} = 1 - \sum_{i=1}^{K} {p_i}^2 \tag{6}$$

where

p_i is the proportion of samples in the node that belong to class i, and

K is the total number of classes.

The Gini impurity reaches its minimum value of 0 when the node is pure (i.e., all samples belong to a single class). Its maximum value, $1 - \frac{1}{K}$, occurs when classes are equally represented within the node. In binary classification, the maximum impurity is 0.5. A lower Gini impurity therefore indicates a purer node. During tree construction, the algorithm evaluates all possible splits and selects the one that yields the greatest reduction in impurity. This ensures that each subsequent division improves the homogeneity of the resulting nodes and enhances the model's ability to discriminate between classes.

4.4.2 Overfitting and Mitigation Techniques

While decision trees are powerful and flexible, they are prone to overfitting, particularly when the tree grows excessively deep or complex. Overfitting occurs when the model captures noise in the training data rather than underlying patterns, resulting in poor generalisation on new data.

To mitigate this, *pruning techniques* are used to remove branches that contribute minimally to predictive accuracy, thereby simplifying the model. Another common strategy is to set a *maximum depth* for the tree, limiting the number of splits. These regularisation methods help maintain a balance between model complexity and generalisation, enhancing performance on unseen data.

4.5 Ensemble Machine Learning

In recent decades, ensemble machine learning has demonstrated outstanding effectiveness in solving classification problems. This approach has gained substantial attention from the research community, as ensemble methods frequently achieve top rankings in major machine learning competitions. For example, in the Netflix Prize competition, the winning team significantly improved prediction accuracy by blending multiple models (Koren, 2009). Similarly, the winning solution in the 2014 Kaggle Large Scale Hierarchical Text Classification (LSHTC) competition employed an ensemble of sparse generative models (Puurula, Read, & Bifet, 2014). These achievements highlight the strength of ensemble methods in achieving high accuracy in classification tasks.

To provide a clear understanding of ensemble learning and its practical applications, this section introduces three widely used techniques: **bagging**, **boosting**, and **stacking**, along with the Random Forest (RF) and Extreme Gradient Boosting (XGBoost) algorithms as representative implementations.

4.5.1 Bagging

Bagging (short for *Bootstrap Aggregating*) is a parallel ensemble learning technique designed to improve model robustness and accuracy. The key steps are illustrated in Figure 5 and described below:

Bootstrapping

Let *D* be a dataset with *M* rows. To introduce diversity into the training process, multiple bootstrapped datasets D_i (where *i* = 1, 2, …, *N*) are generate by randomly sampling *M* rows *with replacement*. As a result, some rows may appear multiple times, while others may be excluded entirely. Each base model

h_i is then trained on its corresponding D_i, allowing models to learn different aspects of the data. This diversity reduces overfitting and increases generalisation.

Aggregating

Once all base models h_i (typically decision trees) are trained, their individual predictions $\hat{y}_i$ are aggregated to generate the final ensemble prediction $\hat{y}_e$. For classification tasks, this aggregation is typically carried out using *majority voting*, where the most frequently predicted class is selected as the final output. This aggregation reduces variance and improves robustness to data noise, thereby enhancing generalisation performance.

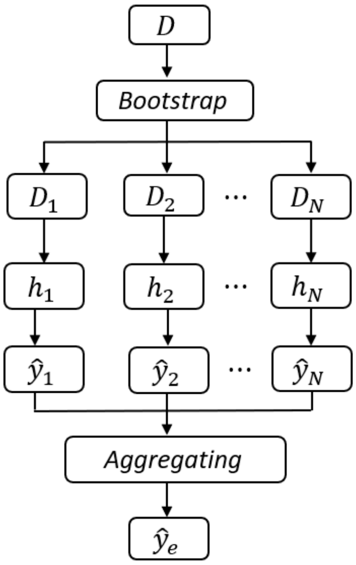

Figure 5. Illustration of the bagging technique.

4.5.2 Random Forest Classifier (RF)

The random forest (RF) classifier, introduced by Breiman (2001), extends the principles of bagging and decision trees to achieve high predictive accuracy and better generalisation to unseen data. It retains the interpretability of decision trees while improving performance through bootstrapped sampling and random feature selection (Qi, 2012).

A key limitation of a single decision tree is its tendency to overfit the training data, particularly when the tree becomes too deep or complex, resulting to a high-variance model with poor generalisation on unseen data. RF addresses this limitation by constructing multiple decision trees, each trained on a bootstrapped sample and a random subset of features, which reduces variance and improves generalisation. Nevertheless, RF models can be computationally intensive due to the large number of trees required, and their ensemble nature makes them less interpretable than individual trees.

Definition of Random Forest

According to Breiman, a random forest classifier $h(x, \Theta_N)$ consists of an ensemble of N decision trees. For each tree, a random vector Θ_N is generated independently and identically distributed (i.i.d.) across trees. Each tree is trained on a bootstrapped dataset D_N, and node splits are determined using a randomly selected subset of features. During prediction, the input x is passed through all N trees, with each tree casting one vote for its predicted class. The final output is determined by majority voting.

Hyperparameter tuning

The key hyperparameters influencing RF performance include (Probst, Wright, & Boulesteix, 2019):

- n_estimators: number of trees in the forest
- max_depth: maximum depth for each tree
- min_samples_split: minimum number of samples required to split a node
- min_sample_leaf: minimum number of samples required at a leaf node
- max_features: number of features considered for the best split

4.5.3 Boosting Methods

Boosting is a sequential ensemble learning technique in which models are trained iteratively. Each successive model is constructed to correct the errors made by the ensemble of previous models. Unlike bagging, where models are trained independently, boosting adaptively concentrates on observations that were previously misclassified, directing learning toward the most challenging cases.

In each iteration, a modified version of the original dataset is generated by incorporating residual information from the current model (James et al., 2023). The process begins by initialising predictions, such as using the mean of the target variable for regression or uniform class probabilities for classification and computing the residuals. A new decision tree is subsequently trained to model these residuals, and its outputs are added to the cumulative predictions to incrementally reduce overall error. This process continues until a predefined number of iterations is reached or the residuals become sufficiently small. The boosting workflow is illustrated in Figure 6.

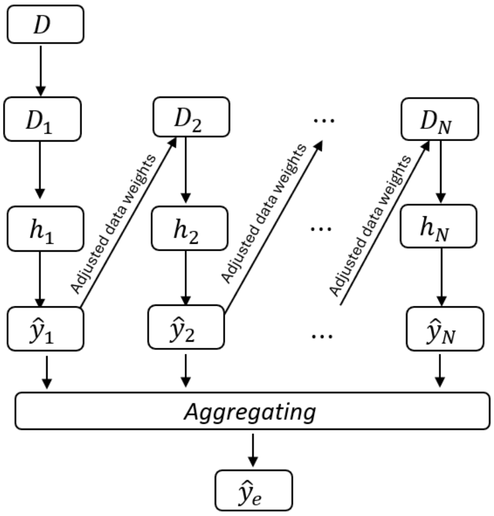

Figure 6. Iterative flow of the boosting algorithm.

4.5.3.1 Extreme Gradient Boosting (XGBoost)

XGBoost, proposed by Chen and Guestrin (2016), is an efficient and scalable implementation of gradient boosting. It has gained widespread adoption due to its strong predictive performance, computational efficiency, and suitability for large-scale datasets.

Objective Function and Regularisation

XGBoost optimises a regularised objective function, which simultaneously optimises predictive accuracy and controls model complexity:

$$L(\phi) = \sum_i l(\hat{y}_i, y_i) + \sum_k \Omega(f_k) \tag{7}$$

where $l(\hat{y}_i, y_i)$ is the logistic loss function that measuring the difference between the predicted value $\hat{y}_i$ and the true label y_i, and $\Omega(f_k)$ is the regularisation term for the k-th tree f_k, defined as:

$$\Omega(f_k) = \gamma T + \frac{1}{2}\lambda\|w_k\|^2 \tag{8}$$

Here, T denotes the number of leaves, w_k is the vector of leaf weights, γ penalises the number of leaves to encourage simpler tree structures, and λ penalises large leaf weights to reduce overfitting. This regularisation framework enhances model generalisation by discouraging excessive complexity.

Hyperparameter Tuning in XGBoost

Effective deployment of XGBoost requires careful tuning of its hyperparameters to achieve optimal performance (Banerjee, P, 2020). Key hyperparameters include:

- n_estimators: number of boosting iterations (trees)
- max_depth: maximum depth of each tree
- learning_rate: step size shrinkage to prevent overfitting
- subsample: fraction of training samples used for fitting individual trees
- colsample_bytree: fraction of features randomly sampled for each tree
- gamma: minimum loss reduction required to create a further partition
- min_child_weight: minimum sum of instance weights required in a child node
- reg_alpha: L1 regularisation term on leaf weights
- reg_lambda: L2 regularisation term on leaf weights

Careful calibration of these hyperparameters can significantly improves predictive accuracy, particularly in high-dimensional or imbalanced data scenarios.

4.5.4 Stacking Ensemble Method

Stacking is an advanced ensemble learning technique that combines the predictive outputs of multiple base models using a secondary model, known as a **meta-learner** (Arthur, 2018). Rather than relying on a single best-performing model, stacking leverages the complementary strengths of diverse learners by using their predictions as input features for the meta-learner. This layered architecture enables the meta-learner to identify optimal combinations of base model outputs, thereby improving overall predictive performance and robustness.

In this study, a stacking ensemble is constructed using four heterogeneous machine learning models: (i) logistic regression, (ii) random forest, (iii) XGBoost and (iv) multi-layer perceptron (MLP). Each base learner independently generates predictions for the classification task, and the meta-learner is trained to integrate these outputs, enhancing predictive accuracy, reducing overfitting, and improving model stability. The structure of the stacking ensemble employed in this study is illustrated in Figure 7.

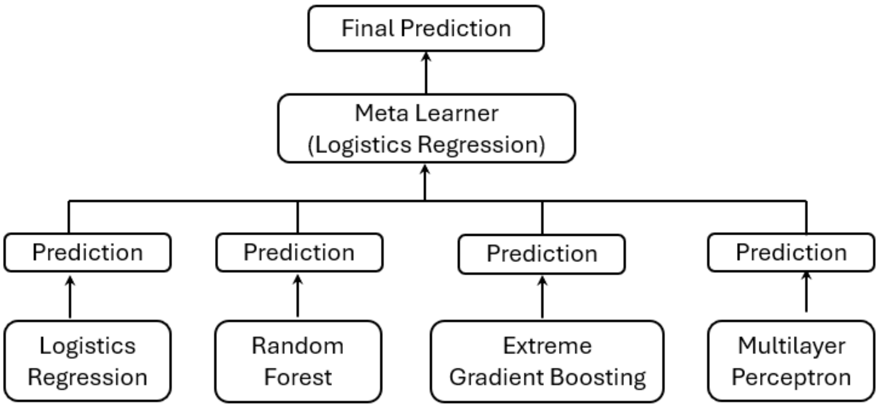

Figure 7. Architecture of the stacking ensemble model.

4.6 Multilayer Perceptron Classifier (MLP)

The perceptron, introduced by Frank Rosenblatt in 1958, laid the foundation for the development of artificial neural networks. As a single perceptron functions only as a linear classifier, multiple perceptrons are combined to form more complex models capable of capturing intricate patterns and non-linear relationships within data. This architecture is known as a multilayer perceptron (MLP). MLPs enable machines to mimic the neural structure of the human brain and perform tasks by processing large volumes of data (Jeatrakul & Wong, 2009). A typical MLP comprises an input layer, one or more hidden layers, and an output layer, as illustrated in Figure 8.

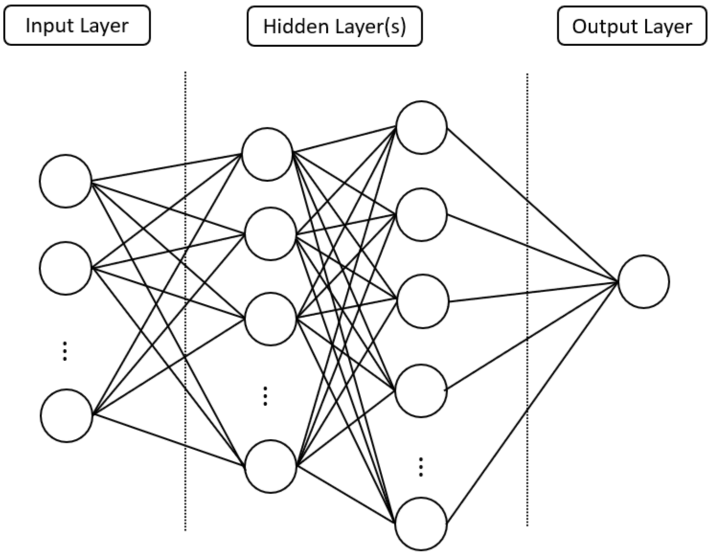

Figure 8. Structure of a multilayer perceptron (MLP).

A widely adopted variant of the MLP is the *Feedforward Backpropagation Network* (FBPN), developed in the early 1970s. FBPNs are among the most widely used and powerful models for constructing complex neural networks. Although no universal rule exists for determining the optimal number of hidden layers or neurons, several heuristic guidelines are commonly applied in FBPN architecture design (Alsmadi et al., 2009).

4.6.1 Guidelines for Choosing Hidden Layer Architectures

i. Increase the number of hidden layers for more complex patterns
   Additional hidden layers allow the model to learn and represent increasingly complex patterns within the data.
ii. Use multiple hidden layers for multi-stage transformations
   If the process being modelled involves several stages of transformation, additional hidden layers may improve performance. However, for processes that are not inherently multi-stage, increasing layers may lead to overfitting rather than better generalisation.
iii. Limit the number of neurons based on dataset size
   The number of neurons in the hidden layers should be constrained based on the size of the training dataset. A commonly rule of thumb is:

$$\text{Maximum neurons} = \frac{\text{No. of training examples}}{(\text{No. of input nodes} + \text{No. of output nodes}) \times \text{scaling factor}} \tag{9}$$

The scaling factor typically ranges from 5 to 10. For datasets with high noise levels, a larger scaling factor (e.g., 20 to 50) is recommended to mitigate overfitting. In this study, the MLP model is trained on 20,103 examples with 43 input variables and one output node. Using a scaling factor of 5, the estimated upper bound for the number of neurons per hidden layer is approximately:

$$\frac{20103}{(43+1)\times 5} \approx 91$$

A grid search is conducted to explore multiple hidden layer configurations, with each layer containing up to 91 neurons, in order to identify the optimal model architecture.

4.6.2 Learning Process of Feedforward Backpropagation Networks

The FBPN learns by iteratively adjusting connection weights to minimise prediction error (Alsmadi et al., 2009). The key steps are:

1. Error calculation
   The model compares predicted outputs with actual targets to compute the prediction error.
2. Backpropagation of error
   The error is propagated backward from the output layer to the hidden layers. Each neuron updates its weights based on its contribution to the error, while neurons with no contribution remain unchanged.
3. Iterative refinement
   The process is repeated over multiple training iterations, progressively updating the weights to improve predictive accuracy.

5. Experimental Setup

5.1 Data Partitioning and Scaling

The dataset is partitioned into 80% training data and 20% testing data. The training set is used for model construction and validation, while the testing set remains unseen during training and is employed to evaluate the model’s generalisation performance.

Prior to model training, feature scaling is applied to standardise the input variables. This preprocessing step ensures that all variables are on a comparable scale, thereby enhancing model convergence and predictive performance. Standardisation is performed using the **Standard Scaler**, which transforms each as follows:

$$z = \frac{x - m}{s} \tag{10}$$

where z is the standardised value, x is the original feature value, m is the mean of the feature, and s is the standard deviation of the feature.

5.2 Stratified *k*-Fold Cross-Validation

Stratified k–fold cross-validation is used to evaluate model performance while preserving the class distribution in each fold. The training dataset is divided into k equal subsets (folds). In each iteration, one fold is used as the validation set, and the model is trained on the remaining $k - 1$ folds. The final performance metric is computed by averaging the evaluation results across all folds.

This method provides a more robust estimate of model performance and mitigates overfitting to specific data partitions. By stratifying the data, each fold approximately maintains the same class proportions, allowing minority classes to be adequately represented in every fold.

5.3 Hyperparameter Optimisation

Hyperparameter tuning is conducted to improve model performance by identifying the optimal combination of predefined parameter values. Two primary search strategies are applied:

i. **Grid Search**: A deterministic and exhaustive approach that evaluates all possible combinations of predefined hyperparameters within a specified search space. Although computationally intensive, it ensures comprehensive coverage of the parameter space.

ii. **Random Search**: A stochastic approach that randomly samples parameter combinations from the predefined search space. This approach is more computationally efficient, especially when the hyperparameter space is large.

In this study, the multilayer perceptron (MLP) model is tuned using grid search due to its relatively small number of hyperparameters, allowing a comprehensive search over hidden layer configurations. In contrast, the random forest and XGBoost models are tuned using random search, which is more computationally efficient for their extensive hyperparameter spaces.

5.4 Model Evaluation Metrics

To assess predictive performance, several evaluation metrics derived from the *confusion matrix* are employed:

- True Positive (TP): Instances correctly classified as positive.
- True Negative (TN): Instances correctly classified as negative.
- False Positive (FP): Instances incorrectly classified as positive.
- False Negative (FN): Instances incorrectly classified as negative.

The primary metric used in this study is the **F1–score**, which combines precision and recall into a single measure, making it especially suitable for imbalanced datasets. Accuracy is also reported as an overall measure of model correctness. The metrics are defined as follows:

- **Accuracy:** Overall correctness of the model's predictions

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{11}$$

- **Recall:** Proportion of actual positives correctly identified

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{12}$$

- **Precision:** Proportion of true positives among all predicted positives

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{13}$$

- **F1–score:** Harmonic mean of precision and recall

$$\text{F1} = \frac{2 \times \text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}} \tag{14}$$

The F1–score is particularly effective for imbalanced datasets, as it provides a balanced measure that accounts for both false positives and false negatives, which may otherwise be overlooked when relying solely on accuracy.

6. Results and Discussion

Table 7 and Figure 9 present a comparative analysis of the performance of the five machine learning models investigated in this study. The decision tree model is excluded from direct comparison, as it serves as a base estimator within the ensemble frameworks. Overall, the results indicate that ensemble models, particularly random forest, outperform traditional logistic regression model in credit risk classification tasks. Although all ensemble methods (random forest, XGBoost, and stacking ensemble) achieved comparable performance, random forest attained the highest F1–score (0.81) and precision (0.87), demonstrating a strong ability to capture complex, non-linear patterns in the data. This capability is especially valuable in credit scoring, where misclassifying risky borrowers as creditworthy can lead to significant financial losses.

The logistic regression model, constrained by its linear nature, demonstrated high recall (0.93) but poor precision (0.33). This implies that while it successfully identifies most of the risky borrowers, it also generates many false positives, potentially misclassifying creditworthy applicants as high risk. In comparison, the multilayer perceptron (MLP) achieved competitive performance, with an F1–score of 0.77 and recall of 0.79. Although slightly behind the ensemble models in precision, MLP may be preferred in applications prioritising recall, i.e., capturing all potential defaulters. However, its extended training time and lack of interpretability (often regarded as a “black box”) may constrain its practical use.

Table 7. Evaluation metrics of the five classification models.

Models		**F1–score**	**Accuracy**	**Recall**	**Precision**
⬛	Logistic Regression	0.49	0.80	0.93	0.33
🟩	**Random Forest**	**0.81**	**0.96**	**0.77**	**0.87**
🟨	Extreme Gradient Boosting (XGBoost)	0.81	0.96	0.77	0.86
🟧	Stacking Ensemble Method	0.81	0.96	0.77	0.86
🟫	Multilayer Perceptron (MLP)	0.77	0.95	0.79	0.76

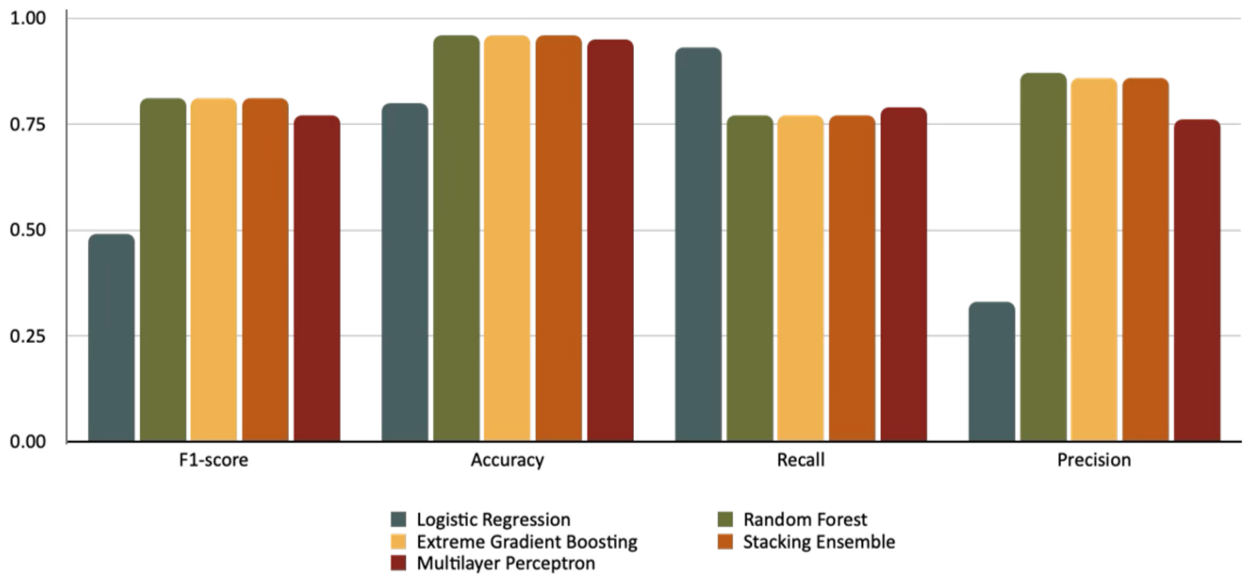

Figure 9. Comparison of evaluation metrics across the five classification models.

Interestingly, the stacking ensemble method did not outperform its individual base models, suggesting a lack of model diversity within the ensemble. For instance, the architecture of XGBoost and RF are tree-based model which may produce similar result when performing classification. Incorporating additional heterogeneous models, such as naïve Bayes, support vector machines, linear discriminant analysis, or MLP with varied internal architectures, could enable the meta-learner to capture novel patterns in the data. This result highlights the importance of including more heterogeneous base learners to fully exploit the strengths of stacking.

7. Conclusions

This study proposed a vintage-based credit scoring framework that integrates repayment behaviour with vintage analysis, adjusting scores for delinquency while allowing flexible risk thresholds. Logistic regression performed poorly, highlighting the limitations of linear models, whereas the multilayer perceptron (MLP) achieved strong recall (0.79) with acceptable precision (0.76), making it suitable when minimising false negatives is critical, although its longer training time and lower interpretability may limit practical application. Among ensemble methods, random forest achieved the highest precision (0.87) and F1-score (0.81), demonstrating robust predictive performance and generalisability. Overall, random forest is the most promising model for deployment, emphasising the effectiveness of ensemble learning in credit risk classification.

8. Future Work

This study shows that ensemble methods achieve superior performance in binary classification tasks. However, the advanced stacking model did not surpass its base learners, likely due to limited model diversity. For instance, RF and XGBoost are both tree-based ensembles, which may produce correlated errors, limiting the meta-learner's ability to capture new patterns. Future work could enhance stacking by combining heterogeneous base models (e.g., decision trees, neural networks, support vector machines) or training them on distinct feature subsets and preprocessing strategies, enabling the ensemble to capture a broader range of patterns. While ensemble learning enhances predictive accuracy, it also increases complexity and reduces interpretability. Incorporating model-agnostic tools such as LIME and SHAP (Nallakaruppan et al., 2023) could provide clearer insights into model behaviour and highlight key contributing features, thereby improving stakeholder trust. The dataset used has a time-series structure, although temporal information was not utilized in training or evaluation. Future research could adopt temporal splits, include time as a feature, or explore time-aware models such as ARIMA, LSTM, or RNNs to further enhance robustness and practical utility.

9. Acknowledgements

I sincerely appreciate the professional perspective provided by Dr. Soo throughout my paper writing process. Her invaluable support has been instrumental in my continuous efforts to enhance my work, and I am deeply grateful for the opportunity to have her as my supervisor. I would also like to thank all the reviewers and JOSMA team for their time, effort, and support in facilitating the publication of this paper.

10. References

Anjani, S. D. D., Bhargavi, A., Aishwarya, S. M. L., Karthik, S. P. P., Vamsi, T., & Rishi, N. M. S. K. (2023). Credit card approval prediction using machine learning. *Journal of Information Technology (JIT)*, 12(3): 39.

Arthur, T. (2018). Introduction to ensembling/stacking in Python. *Kaggle Notebook*. https://www.kaggle.com/code/arthurtok/introduction-to-ensembling-stacking-in-python

Axelton, K. (2025). What happens if you don't pay back a personal loan? *Experian*. https://www.experian.com/blogs/ask-experian/what-happens-if-you-dont-pay-back-personal-loan/

Banerjee, P. (2020). A guide on XGBoost hyperparameters tuning. *Kaggle*. https://www.kaggle.com/code/prashant111/a-guide-on-xgboost-hyperparameters-tuning

Breiman, L. (2001). Random Forest. *Machine Learning*, 45(1): 5–32.

Chen, T. & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785–794.

Credit card approval prediction. (2020). *Kaggle*. https://www.kaggle.com/datasets/rikdifos/credit-card-approval-prediction

Gorman, J. (2025). What is a FICO score? *Bankrate*. https://www.bankrate.com/personal-finance/credit/what-is-a-fico-score/

Hastie, T., Tibshirani, R. & Friedman, J. (2009). *The elements of statistical learning* (2nd ed.). Springer.

James, G., Witten, D., Hastie, T., Tibshirani, R. & Taylor, J. (2023). *An introduction to statistical learning: With applications in Python*. Springer.

Jeatrakul, P. & Wong, K. W. (2009). Comparing the performance of different neural networks for binary classification problems. *Proceedings of the Eighth International Symposium on Natural Language Processing*, 111–115.

Koren, Y. (2009). The BellKor solution to the Netflix Grand Prize. *Netflix Prize Documentation*, 81: 1–10.

Nallakaruppan, M. K., Balusamy, B., Shri, M. L., Malathi, V., & Bhattacharyya, S. (2023). An explainable AI framework for credit evaluation and analysis. *Applied Soft Computing*, 153: Article 111307.

Probst, P., Wright, M. N. & Boulesteix, A. (2019). Hyperparameters and tuning strategies for random forest. *WIREs Data Mining and Knowledge Discovery*, 9(3): e1301.

Puurula, A., Read, J., & Bifet, A. (2014). Kaggle LSHTC4 winning solution. *arXiv preprint arXiv:1405.0274*.

Qi, Y. (2012). Random forest for bioinformatics. In Zhang, C. & Ma, Y. (Eds.), *Ensemble Machine Learning*, 307–323. Springer.

Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6): 386–408.

Alsmadi, M. K. S, Omar, K. B., & Noah, S. M. (2009). Back propagation algorithm: The best algorithm among the multi-layer perceptron algorithm. *International Journal of Computer Science and Network Security (IJCSNS)*, 9(4): 378–383.

Singh, A. (2021). What machine learning approaches have won most Kaggle competitions? *Kaggle*. https://www.kaggle.com/discussions/general/248068

Song, Y.-Y. & Lu, Y. (2015). Decision tree methods: Applications for classification and prediction. *Shanghai Archives of Psychiatry*, 27(2): 130–135.

Soo, W. J. (2023). Malaysian youths in debt, mostly for car loans. *Malay Mail*. https://www.malaymail.com/news/malaysia/2023/01/31/ucsi-survey-finds-seven-in-10-malaysian-youths-in-debt-mostly-for-car-loans/52723

Stanimir, A. (2011). Vintage analysis as a basic tool for monitoring credit risk. *Mathematical Economics*, 7(14): 214–228.

Tunguz, B. (2018). Home Credit Default Risk Competition. *Kaggle*.

Yuan, Y., Wu, L., & Zhang, X. (2021). Gini-impurity index analysis. *IEEE Transactions on Information Forensics and Security*, 16(1): 3154–3169.