Department of Computer Science


Submitted in part fulfilment for the degree of BEng.

# Credit Default Prediction Using Machine Learning

Daniz Hajizada

2025-April

Supervisor: Xinwei Fang

This work is dedicated to my family, whose support has made this journey possible.
It is also dedicated to my late grandparents, Zakiyya and Gurban, whose values and memory continue to shape the person I am today. As this project marks the conclusion of my degree, I honour their lasting influence on my character.


## Acknowledgements

I would like to thank my project supervisor, Mr. Xinwei Fang, for all invaluable guidance, support and trust throughout the course of this research.
I am also grateful to project allocator, Mr. Simon Foster, for providing me with the opportunity to take this project. Additionally, I would like to extend my thanks to my personal academic supervisor, Mr. Antonio Garcia-Dominguez, for his constant support and encouragement during the writing process and throughout my studies.
Finally, I would like to express my appreciation to Jo Enderby, the department receptionist, for all her support and assistance in preparing essential documents for my master's application, helping me take the next step in continuing my studies.

# Contents

**Executive Summary** **vi**

**Statement of Ethics** **1**

**1 Introduction** **1**

**2 Literature Review** **4**

**3 Methodology** **7**
3.1 Models Overview . . . . . 7
3.2 Data Overview . . . . . 8
3.3 Feature Engineering Process . . . . . 10
3.4 Logistic Regression . . . . . 10
3.5 Tree-Based and Boosted Models . . . . . 16
3.6 Data Split for Models Above . . . . . 16
3.7 Artificial Neural Network . . . . . 16

**4 Results** **18**
4.1 General results . . . . . 18
4.2 Model Stacking - Ensemble Learning . . . . . 20
4.3 Feature Importance . . . . . 21

**5 Conclusion** **24**

# List of Figures

3.1 Boxplot of `Delay_Days` before outlier treatment . . . . . 14
3.2 Boxplot of `Delay_Days` after outlier treatment . . . . . 14
3.3 ANN architecture. . . . . 17

4.1 Strongest SHAP interaction . . . . . 21
4.2 Detailed view of feature importance bar chart. . . . . 22

# List of Tables

3.1 Comparison of Machine Learning Models for Credit Risk Prediction . . . . . 8
3.2 Categorisation of Features in the Credit Card Default Dataset 9
3.3 Engineered Features Derived from Repayment Status Variables . . . . . 11
3.4 Engineered Features Derived from Billing and Payment History Columns . . . . . 12
3.5 The dataset for Logistic Regression up to this point . . . . . 14

4.1 Final Model Performances Based on Gini Coefficient . . . . 20

# Executive Summary

Predicting bank failures plays a crucial role in preventing catastrophic consequences such as bankruptcy and damage to the economy. As known, nowadays banks prioritise customer satisfaction, meaning a significant portion of risk services is tied to user loans. Therefore, accurately predicting customer loan/credit defaults is crucial for sustaining good service and healthy banking. This leads to the Probability of Default (PD) problem, which is a critical issue in financial risk management that involves determining the likelihood of customers failing to meet their loan/credit obligations. This paper aims to identify the most effective machine learning models for Taiwanese credit card clients [1]. The motivation for this research stems from both the increasing need for banks to manage credit risk efficiently. Additionally, the popularity of advanced machine learning models continues to grow, yet many academic papers do not clearly differentiate which models perform better for credit default prediction. To address this, 12 popular machine learning models were implemented — including Logistic Regression, Decision Trees, Random Forest, LightGBM, XGBoost, CatBoost, and an Artificial Neural Network — as well as their optimised versions using the Optuna hyperparameter tuning framework. A model stacking approach was also explored. The evaluation was based on three key success criteria: high test Gini coefficient, minimal overfitting (small gap between train and test scores), and practical applicability. The best-performing model, a combination of Logistic Regression, LightGBM, and CatBoost within a stacking ensemble, achieved a test Gini of 0.5356 (=53.56%), passing the commonly accepted industry threshold of 40% for credit scoring models [2]. This indicates a possibility for real-world deployment, particularly in credit risk departments. Tree-based models showed satisfactory results and handled imbalanced data and multicollinearity well. These findings imply that the combination of traditional and modern machine learning techniques could be the most suitable for financial use. Future work could explore more complex ensemble strategies, integration of real-time banking data with more powerful devices, or application to different loan types to further validate generalisability. Feature importance and SHAP values analyses revealed that variables related to delayed payment history and credit limits were the most influential predictors of defaulters.

**Keywords:** banking; probability of default; financial risk management.

# Statement of Ethics

No significant legal, social, or professional concerns were identified in the conduct of this study. However, an ethical consideration arose from working with human data and using an external dataset.

**Avoidance of Harm**: The dataset used in the paper includes sensitive financial information such as credit limits and payment history; however, it is fully anonymised and de-identified. The project did not involve any surveys or interviews with individuals, and no procedures were carried out that could cause physical, psychological or reputational harm.

**Informed Consent**: The data was collected and publicly released by the third party platform. The dataset is distributed under the 'Public Domain' license, meaning that it can be copied, modified, distributed without asking permission. No new data was collected for the purposes of this project.

**Data Protection**: The dataset includes human data; however, it does not contain any personally identifiable features and was fully anonymised before publication. In addition, the data was published in a well known database and with appropriate consent, which confirms that ethical standards were followed during and after publication. The data will be deleted following the completion of the project in accordance with university data handling guidelines.

**Ethical Approval**: A Fast-Track Application Form was submitted for ethical review. The use of human data from an external dataset was reviewed and approved by the Department of Computer Science Department Ethics Team.

# 1 Introduction

As mentioned briefly in the executive summary, the relevance of this study lies in the importance of predicting the PD in financial risk management, particularly in the context of modern credit and banking systems. Much research has explored this topic, where models like Logistic Regression remain dominant due to its simplistic nature and consistent reliability throughout many years; however, the capabilities of such traditional methods may no longer be sufficient in capturing the complex nature of modern credit systems.

While reading about machine learning in risk prediction, several studies stood out that used advanced boosting models such as LightGBM and XGBoost in domains like healthcare and fraud detection. Research success proved that models like these could work for predicting credit defaults because of overlapping characteristics with previously mentioned domains. This observation motivated the exploration of machine learning techniques for the PD problem, aiming to assess their effectiveness and adaptability in default risk modelling.

Datasets play a massive role in studies of this nature, as their quality directly influences the validity of both the study's results and the performance of models proposed. This research uses a well-structured dataset of 30,000 Taiwanese credit card clients, which includes a range of variables such as demographic details, credit limits, payment histories, and billing and payment amounts across six months. This provides a realistic and rich representation of credit behaviours, enabling the practice of both traditional and advanced machine learning methods in PD problem.

Even though a more in-depth discussion of both dataset overview and methodology process are discussed later in this paper, a key consideration in working with any dataset is the preprocessing stage. Common preprocessing steps include:

1. Handling missing values, including invalid entries. Having missing values and invalid entries in the dataset not only reduce the performance of machine learning models, but may also introduce bias or errors during training and evaluation. As the simple principle of "garbage in, garbage out" (GIGO) suggests [3], models are only as good as the

data they are trained on—if the input is incomplete or incorrect, the output will be unreliable.

2. Encoding categorical variables. This means converting non-numerical data, such as education levels, into numerical formats for compatibility with machine learning models. However, some models such as CatBoost do not require this step.

3. Descriptive statistics help us understand the overall structure of the dataset by summarising key characteristics such as mean, median, standard deviation, etc.

4. Feature distribution analysis, where the Kolmogorov–Smirnov (K–S) test was applied to each numerical feature to evaluate its distribution. This helps to determine whether the data is normally distributed across the database, which could potentially influence and sabotage results.

5. Correlation plays a massive role in determining the extent to which variables change together. This is useful for feature selection and understanding the dependencies in data.

6. Feature engineering. Generating more features expands the model's perception and gives it more coverage, hence improving the model's understanding.

By following these traditional preprocessing steps, the dataset achieves completeness and consistency.
The relevance of this topic is further highlighted by the evolving demands of credit risk management, particularly as many successful businesses aim to integrate predictive modelling into their risk assessment systems. This requires precise attention to the preparation, analysis and evaluation of methodologies used.

The **objective** of this research is systematically compare traditional and advanced modelling techniques for PD prediction, focusing on their performance under optimised conditions.

**The point of investigation** - the influence of hyperparameter tuning, through frameworks such as Optuna, on predictive machine learning models. This includes evaluating how optimised parameters, such as learning rate, tree depth, and split criteria, influence model accuracy.

In the line with the main objective, the following **tasks** are set: to follow a step-by-step approach in conducting the methodology of the study; to perform feature engineering to refine the dataset and maximise its pre-

dictive potential; to develop a table of variables with explanations of their roles and significance in predicting defaults; and to establish key variables that influence prediction accuracy the most.

The **methodology section** of this research relies on a structured approach to data preparation, analysis, and model evaluation. Preprocessing steps include distribution checks, outlier treatment, different transformations, and multicollinearity reduction. Both classical models and modern machine learning techniques were applied and optimised using Optuna framework for hyperparameter tuning. Feature importance and SHAP values analysis were carried out at the end, highlighting the interpretability topic. The logical flow in the methodology section creates practical value when used for credit default modelling applications.

The **structure** of this research is organised as follows: introduction, literature review section, methodology section, which will include multiple subsections, results section, and general conclusions in the last section of the study.

# 2 Literature Review

The aim of this literature review is to build a foundational understanding of the modern banking sector and its role in society, while exploring the evolution of predictive methods in this domain. This chapter begins with traditional risk assessment approaches, such as the Z-score for bankruptcy prediction, then shifts focus to recent advancements in credit default modelling using machine learning techniques. It concludes with the introduction of the concept of hyperparameter tuning, which plays a key role in optimising these models for better performance.

The banking sector operates in a competitive environment where customer satisfaction is a backbone for long-time success. Banks must continuously improve their service quality to maintain existing customers, sustain profitability and engage new customers [4]. However, aside from customer service and profitability, banks play a huge role in the economy. As financial intermediaries, they facilitate efficient capital allocation by channelling savings into productive investments, supporting economic growth and stability [5]. Additionally, state-owned banks contribute to overall economic development by providing financial support to government projects and offering low-interest rates, especially in economically weak regions [6]. Poorly functioning banks are considered as obstacles to economic progress and aggravate poverty. The 2008-2009 financial crisis demonstrated how instability in the banking sector can lead to economic downturns, as banking distress results in more prolonged recessions compared to other financial shocks [7]. Even though the possibility of a similar crisis remains, modern banking regulations — such as Basel III — aims to prevent excessive lending and risky financial practices, significantly reducing such risks [8].

Given the sector's important ties to economic health, it is crucial to monitor banks' financial stability. To this day, the Z-score model [9] remains a very present and common tool for evaluating the financial health of banks. According to Rashid, Khan, and Qureshi [10] the application of this model scopes to multiple different areas, such as banking sector, service industry, financial institutions, and public stock market analysis. In each of these sectors, it has demonstrated significant value in predicting bankruptcy scenarios. They suggest that the Z-score is not a one-size model and heavily depends on the industry it is applied in. In other words, the proposed Z-score thresholds differ, leading to changes to Safe, Gray,

and Distress zone borders. The interpretation in banking specific sector follows a different approach compared to any other industry mentioned previously. While a Z-score above 2.6 or 2.9 is considered a "Safe Zone" for public and private enterprises, banking institutions require much higher Z-scores to indicate financial stability. In banking applications, the Z-score is more of a relative measure of risk rather than a strict classification model [11].

Despite the success and popularity of this model, relying solely on it presents several challenges. The study by Rashid, Khan, and Qureshi [10] has also suggested that the model had to go through changes multiple times because its past versions became less effective due to changes in financial environments. This can be also interpreted as if the model was fully adaptable to changing conditions, there would not be a need for constant modifications. This suggests that the model relies on historical financial data and assumptions that may not always apply to present or future economic conditions. In response to these limitations, researchers have proposed machine learning techniques as a much more effective alternative. In a study by Tran et al. [12], previous studies have shown that a Support Vector Machine (SVM) model with less restrictive constraints than modified Z-score model, has better exploit rates and proved to be more effective. In other words, the predictive capabilities of modern model concepts are better and more effective than traditional methods. Given the fact that the probability of default naturally is connected to the financial health of a bank - a concept that traditionally evaluated risk management through models like Z score - it is therefore relevant and important to explore this topic.

Various research papers have explored this matter using machine learning techniques, each using different methodologies in order to catch complex relationships and make an accurate prediction. One research has explored the application of Artificial Neural Network (ANNs) in predicting credit risk and probability of default. Nazari and Alidadi [13] applied this methodology in order to classify bank customers as good or bad borrowers based on financial and demographic data. This research highlighted the practical use of ANNs in the banking sector, reinforcing the argument that machine learning models can outperform traditional statistical approaches in credit risk assessment. More recent papers are further focusing on the effectives of machine learning models. Zhu et al. [14] compared the predictive performance of Logistic Regression, Decision Trees, XGBoost, and LightGBM in predicting loan defaults. Their findings suggest that loan term, loan grade, and credit score play crucial roles in determining default risk. They have also achieved higher accuracy and precision, and found that XGBoost and LightGBM significantly outperformed other discussed models in the study.

However, the effectiveness of these models heavily depends on how they

are configured. Many machine learning methods provide powerful predictive capabilities that are heavily dependent on their inner configurations. Most of them have their algorithms configured to default parameters, which is not always optimal as the operation might not reach its full potential. Hyperparameter tuning plays a huge part in model optimisation, allowing to manipulate the configuration that could significantly improve accuracy performance. It is worth mentioning the fact that wrong decisions in this area can affect either the quality of the resulting model or the speed of convergence of the tuning procedure [15].

Overall, the review revealed that traditional models like the Z-score offer a foundation for risk assessment; however, they lack adaptability in evolving financial environments. This suggests that traditional methods in machine learning, such as Logistic Regression, may eventually face similar limitations in the context of credit default prediction. Recent studies show that machine learning models — particularly boosting techniques like LightGBM and XGBoost — outperform traditional approaches. These models identified variables such as loan term, loan grade, and credit score as crucial predictors of default risk.

These findings highlighted the importance of historical behaviour attributes such as loan term, loan grade and credit score, which directly influenced the feature engineering process in the study, where multiple history-based variables were created to reflect delayed payments, repayment patterns and total debt. Additionally, careful attention was paid to preprocessing steps for Logistic Regression such as outlier treatment, multicollinearity, and Weight of Evidence transformation to maximise its performance to get the best possible results.

# 3 Methodology

In this section, the methodology steps used to preprocess the dataset, feature engineer, train models and optimise their hyperparameters are discussed. The goal of this section involves testing various models and their performance metrics under certain conditions. The basecode used and any other aspect of the project details can be found on github repository of the author - https://github.com/danizhajizada/FinalYearProject.

## 3.1 Models Overview

In this study, 12 machine learning models will be used to evaluate and compare predictive performance. These include Logistic Regression, LightGBM, XGBoost, CatBoost, Random Forest, and an Artificial Neural Network (ANN). As previously promised, optimised versions of LightGBM, XGBoost, CatBoost, and Random Forest will be implemented using the Optuna framework for automated hyperparameter tuning. However, only the optimised version of the Artificial Neural Network was developed, as the manual configuration of the network would require specifying weight initialisations, learning rates, layer dimensions, and activation functions individually.

The decision to evaluate 12 machine learning models was made to ensure a diverse comparison across different modelling families and domains. As shown in Table 3.1, these include: (1) a traditional statistical model (Logistic Regression), (2) classical tree-based models (Decision Tree, Random Forest), (3) advanced boosting models (LightGBM, XGBoost, CatBoost), (4) a deep learning model (Artificial Neural Network), and (5) optimised versions of selected models using the Optuna framework. These models represent best practice approaches commonly used in the banking area. The reasoning behind the use of such a wide range of models lies in the diversity of the design, as the most optimal and best performing results can only be identified through comparison of alternative approaches. In addition, numerous real-world applications exist in which specific models from the list above, such as Logistic Regression, are used as baseline standards. Finally, the literature review identified two studies that also employed several of these models, reinforcing their relevance and popularity

in financial forecasting.

| Machine Learning Models for Default Prediction | | |
|---|---|---|
| **Model Type** | **Models Used** | **Complexity** |
| Statistical Model | Logistic Regression | Low (Baseline) |
| Tree-Based Models | Decision Tree<br>Random Forest | Medium<br>Medium |
| Boosted Models | LightGBM<br>XGBoost<br>CatBoost | High<br>High<br>High |
| Neural Networks | Artificial Neural Networks (ANN) | Very High |

Table 3.1: Comparison of Machine Learning Models for Credit Risk Prediction

The machine learning models were implemented using Python 3.12.1, along with key libraries including NumPy 1.24.3, Pandas 2.1.1, Scikit-learn 1.5.2, Seaborn 0.13.2, Matplot for visual representation of results and TensorFlow 2.18.0 for ANN.

## 3.2 Data Overview

Despite mentioning the database previously in the introduction section of this paper, a more detailed discussion of its structure, content, and relevant decisions is provided here. The dataset used in this study is the Default of Credit Card Clients Dataset, made publicly available by the UCI Machine Learning Repository. It contains demographic information on 30.000 credit card holders in Taiwan. The relevance was the deciding factor for dataset selection, as its structure and origin from a real Taiwanese bank provide practical realism to the study. This is the closest publicly available source that reflect the nature of data used in banking environments [1].

The dataset is designed to support binary classification tasks related to credit risk. The dataset includes 25 input features and a binary target variable, which implies that its primary objective is to predict whether a client will default on their payment in the following month, which is also illustrated by the target variable's name - "**default**". This variable takes the value of 1 if a default is expected, and 0 otherwise. Approximately 22.1% of the clients in the dataset are "defaulters", which implies that the data is imbalanced. In any other database, particularly those used in academic studies, it would be necessary to carefully address the data imbalance using oversampling/undersampling methods, such as SMOTE

| Feature Group | Features |
|---|---|
| Demographic Information | SEX (Gender), EDUCATION, MARRIAGE, AGE |
| Financial Data | LIMIT_BAL (Amount of given credit), BILL_AMT1 to BILL_AMT6 (Bill statement amounts over the past six months) |
| Historical Repayment Records | PAY_0 to PAY_6 (Repayment status for each of the past six months), PAY_AMT1 to PAY_AMT6 (Actual amounts paid during those months) |

Table 3.2: Categorisation of Features in the Credit Card Default Dataset

or RandomOverSampler, to guarantee that minority classes are properly represented during training; however, in real-world practical scenarios, such artificial interferences are intentionally avoided. This is due to the fact that working with naturally imbalanced data reflect the true operational conditions of banks in the real world.

As shown in Table 3.2, the dataset consists of 25 input variables which can be mainly grouped into three categories.

Before applying any transformations to the data, for data processing and perception, a *"value_counts()"* operation was performed to understand the distribution of demographic information. For example, the column *"Education"* included "not educated" records. Similarly, another column *"Marriage"* had a very rare "others" group. This observation confirms the depth and inclusivity of the dataset used.

The dataset contains no missing values, which makes it ideal for direct application of desired machine learning techniques. The column for unique identifier "ID" was excluded from all modelling tasks to prevent further inconvenience.

## 3.3 Feature Engineering Process

As briefly mentioned in "Introduction" section, feature engineering is a key consideration to capture more complex and hidden patterns to improve model's predictive performance. This project takes advantage on this matter and some new features were constructed. Table 3.3 illustrates the engineered features derived from repayment status. In addition to these features, new variables were created using the billing and payment columns (`BILL_AMT1` to `BILL_AMT6`, and `PAY_AMT1` to `PAY_AMT6`). Each debt value was calculated using a simple formula:

$$\texttt{BILL\_AMTx} - \texttt{PAY\_AMTx} = \texttt{Debt\_x}$$

Using these new variables, new features were created. Table 3.4 shows the engineered features based on billing and payment history columns. A straightforward scoring method was developed to evaluate client payment performance during each month. A payment score was assigned to a client each month, 0 being fully paid or overpaid, 1 being partial payment and 2 being no payment at all. This produced six new features: `Month_x_Payment_Score`, x being a number from 1 to 6.

The feature richness received reinforcement through the addition of statistical features derived from essential features including `LIMIT_BAL`, `Max_Delay`, `Delay_Status` and `Delay_Category`. Selected numerical features received calculations for mean, sum, minimum and maximum statistics before returning the results to the original dataset. After feature engineering the decision was reached to eliminate intermediate variables from the analysis. This means that the model would process only the most relevant and meaningful variables. The transformation phase led to the development of 40 new input variables with the exception of the single dropped variable ("ID").

## 3.4 Logistic Regression

Logistic Regression is a sensitive linear model, which is sensitive to multicollinearity, variable distribution, and feature scale.

The last step before training and testing stage is to assign the working data to the machine learning models. Since there are no missing values, the data is already ready for models like Random Forest, LightGBM and others. However, for models like Logistic Regression more careful data

| Feature Name | Description |
|---|---|
| `Delay_Status` | Binary variable indicating whether the client had any history of delayed payments. Takes value 1 if any delay has occured, 0 otherwise. |
| `Delay_Count`, `Max_Delay` | Numerical variables capturing the total number of delayed months and the longest delay experienced. |
| `Delay_Category` | Categorical feature that groups clients into: 0 = no delay, 1 = moderate delay (max delay $\leq 3$), 2 = severe delay (max delay $> 3$). |
| `Delay_Days` | Approximate number of overdue days, calculated as `Delay_Count` multiplied by 30. |
| `Sequential_Delay_Status` | Binary feature indicating whether the client experienced two or more consecutive months of payment delays. |
| `Has_Early_Payment` | Binary indicator for whether any repayment value was negative (i.e., early payment). |

Table 3.3: Engineered Features Derived from Repayment Status Variables

| Feature Name | Description |
|---|---|
| `Avg_Bill_Amt`, `Total_Bill_Amt` | Represent the client's average and total billed amounts over the six-month period. |
| `Avg_Pay_Amt`, `Total_Pay_Amt` | Represent the client's average and total amounts paid over the same six-month period, giving insight into repayment capacity and consistency. |
| `Dif_between_Total_pay_and_Total_Bill` | The difference between total payments and total billed amounts. |
| `Total_Bill_Amt_Quantile` | A categorical variable placing clients into quartiles based on their total billed amount. |

Table 3.4: Engineered Features Derived from Billing and Payment History Columns

handling should be applied, especially to the data distribution of the input features. To address this issue, Kolmogorov-Smirnov (K-S) was applied to each numerical feature in the dataset to determine whether a set of samples follows a certain distribution - in this case, normal distribution. If the difference between the samples is greater than 0.05, the data may be normally distributed, in any other case - not. The formula to calculate this metric is straightforward:

$$D = \max_x |F_1(x) - F_2(x)|$$

Here, $F_1(x)$ and $F_2(x)$ represent the empirical cumulative distribution functions of the two samples. $F_2(x)$ is replaced with the cumulative distribution function (CDF) of the reference. The calculated $D$ value gets checked against critical values obtained from the K–S distribution table that requires sample size and significance level of 0.05. The conclusion for distribution difference determination emerges when the test statistic $D$ surpasses its critical value [16].

The results from this test suggest that none of the numeric columns are normally distributed.

For further understanding and analysis of the data, target correlation and inner correlation checks were made. It is worth mentioning that as the main method of identifying the correlation, the decision was made to use the Spearman's method. Spearman's correlation coefficient measures

the strength and direction of association between two ranked variables. It determines the strength and direction of the monotonic relationship between them, meaning as one variable increases, the other tends to increase (or decrease) consistently. Spearman's method proves more suitable than Pearson's method because the K-S test results indicate the dataset does not adhere to normal distribution. There are two approaches to calculate Spearman's correlation depending on whether there are no tied ranks, which is considered an ideal case, and there are tied ranks, which is more common in real-world data [17].

When there are no tied ranks:

$$\rho = 1 - \frac{6 \sum d_i^2}{n(n^2 - 1)}$$

where $d_i$ is the difference in paired ranks and $n$ is the number of cases.

When there are tied ranks:

$$\rho = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_i (x_i - \bar{x})^2 \sum_i (y_i - \bar{y})^2}}$$

where $i$ is the paired score.

A threshold of 0.3 was chosen to select features that have at least a moderate relationship with the target. The observation shows that `Delay_Status`, `Delay_Count` and `Max_Delay` were among the most correlated features with the target variable. However, when it comes to intercorrelation, a threshold of 0.6 was chosen in attempt to exclude features that are too similar to each other. This could happen especially when intensive feature engineering process was applied. This statement is further confirmed by the results - `Delay_Category`, `Sequantial_Delay_Status` and score-based variables like `Delay_Total_Score_sum_by_Delay_Status` and `Total_Score_mean_by_Delay_Status` were constantly involved in tight correlations with each other. Without any interference, this problem could affect the regression results.

To tackle this issue, the Variance Inflation Factor (VIF) was calculated for a selected subset of features. VIF is a measure of the amount of multicollinearity in regression analysis. High VIF values indicate that a variable is highly collinear with one or more other variables in the model [18]. By multiple trial and error, the decision was made to keep informative features such as `Max_Delay`, `Delay_Days`, and `Sequantial_Delay_Status`. Other features such as `Delay_Count`, `Delay_Category`, `Total_Score_mean_by_Delay_Status` and other score-based variables were abandoned due to high VIF coefficient.

Table 3.5: The dataset for Logistic Regression up to this point

| **Variables** |
|---|
| SEX |
| EDUCATION |
| MARRIAGE |
| Delay_Days |
| Sequantial_Delay_Status |
| Max_Delay |
| Delay_Status_sum_by_Max_Delay |
| Total_Score_sum_by_Delay_Category |
| default |

The dataset's look for Logistic Regression is shown in Table 3.5.

To avoid the influence of extreme values in the dataset, an outlier treatment was performed, using the Interquartile Range (IQR) method. It is a descriptive statistics formula, which informs the spread of the data by measuring the range within the central half of values. The formula goes as follows:

$$IQR = Q3 - Q1$$

where: $IQR$ = Interquartile Range, $Q3$ = 3rd quartile or 75th percentile, and $Q1$ = 1st quartile or 25th percentile.

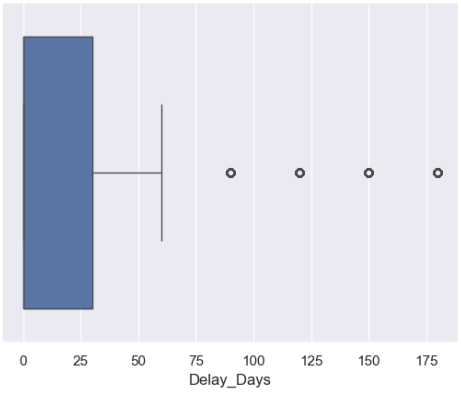


Figure 3.1: Boxplot of `Delay_Days` before outlier treatment

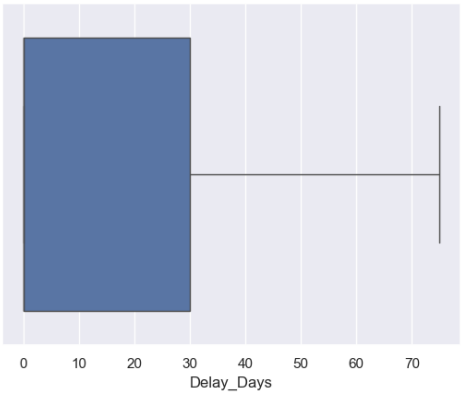


Figure 3.2: Boxplot of `Delay_Days` after outlier treatment

As it is illustrated on Figures 3.1 and 3.2, the boxplots for `Delay_Days` clearly show the presence of extreme values prior to outlier treatment. Specifically, there are multiple instances being beyond the upper hand. However, after applying the IQR-based outlier treatment, the distribution

becomes more compact and within the acceptable range of values. This treatment has been applied to every numerical column, which reduces the overall influence of outlier affecting the model's predictive capability.

In order to improve the model's understanding of the data, a Weight of Evidence (WOE) transformation will be applied. The purpose of the Weight of Evidence module is to provide flexible tools to recode the values in continuous and categorical predictor variables into discrete categories automatically, and to assign to each category a unique Weight-of-Evidence value [19]. In other words, an input instance would be refined into a unique value close to the target. As noted by Raymaekers et al., WOE is commonly used in credit risk modelling due to its strong balance between interpretability and predictive performance, and its compatibility with logistic regression and similar models [20]. This is due to the fact that the unique WOE value is used to measure the strength of grouping for separating good and bad default risk, while also providing. The formula does as follows:

$$\text{WOE} = \log\left(\frac{\text{Proportion of Goods}}{\text{Proportion of Bads}}\right)$$

or equivalently:

$$\text{WOE}_i = \log\left(\frac{n_i^{\text{good}} / N^{\text{good}}}{n_i^{\text{bad}} / N^{\text{bad}}}\right)$$

where:

- $n_i^{\text{good}}$: Number of non-default (good) observations in group $i$
- $n_i^{\text{bad}}$: Number of default (bad) observations in group $i$
- $N^{\text{good}}$: Total number of non-default observations
- $N^{\text{bad}}$: Total number of default observations

The logit transformation used in logistic regression is simply the logarithm of the odds. Therefore, by using WOE-coded predictors, all variables are transformed onto the same scale, which aligns with the logit function [19].

## 3.5 Tree-Based and Boosted Models

As mentioned in previous subsection of this study, models such as Decision Tree, Random Forest, LightGBM and other models on the list do not require strict control as Logistic Regression. According to Chowdhury et al. [21], tree-based models such as Decision Trees and Random Forests are not negatively impacted by the multicollinearity and non-normality, making them an ideal choice for datasets with skewed distributions and multicollinear features, which is the case for our dataset.

The input data requirements for boosted models do not include normal distribution nor the absence of multicollinearity. This is because they are based on tree structures and inherent the immunity to skewed distribution and outliers.

## 3.6 Data Split for Models Above

After the preprocessing and feature engineering steps were completed, the dataset was split into input and output variables for model training. Logistic Regression, tree-based and boost models received the target column `default` for use as their output variable. The data received a fixed random seed value of 42 during its process of dividing training and testing subsets according to a ratio of 70/30.

However, in the case of CatBoost, a copied version of the dataset was used in order to take advantage of its built-in ability to handle categorical features directly, without the need for any modifications. This special ability was mentioned in the Introduction section of the paper, where the topic of handling categorical variables was discussed.

## 3.7 Artificial Neural Network

The same dataset was used for the Artificial Neural Network. Input and outputs variables were divided from the start, as the dataset is already ready for utilisation. Given the sensitivity of neural network to feature scaling, all inputs were scaled using `StandardScaler`. Standardisation can be sensitive to outliers, as extreme values may influence the scaling of features [22]. However, the existing outlier treatment applied to the dataset previously did not affect the process due to the approach already being

completed. The dataset was then split into training and testing split by 80/20 ratio, with random seed of 42.

The model was configured using a Sequential architecture, which is suitable when layers are arranged in a straight line. The network includes two hidden layers, with the number of neurons (units) in each layer chosen through hyperparameter tuning. Both layers use the ReLu (Rectified Linear Unit) activation function. As noted in online article by Bharath Krishnamurthy, ReLu is one of the most popular activation functions and also known for avoiding issues like vanishing gradients, where the network struggles to learn because the updates during training become too small. It also makes the model efficient by turning off some neurons [23]. In other words, the network focuses only on important signals. The output layer has one neuron with a sigmoid activation function, which produces a probability between 0 and 1. This represents the likelihood that a given input belongs to the default class. Both learning rate and optimiser are suggested by the hyperparameter tuning framework Optuna. For optimisers, four commonly used options were selected - Adam, SGD, RMSprop, and Adagrad. On the other hand, for learning rate parameter the range was from 1e-5 to 1e-2. The performance is evaluated using the AUC (Area Under the Curve) metric. This decision was due to a better indication of how well the model differentiate default from non-default outcomes.

Apart from performance enhancement Optuna selects the values for epochs and batch size. A range of 50 but no fewer than 10 epochs was considered alongside selection of batch sizes from 16 to 64. AUC optimization served as the main goal of Optuna todiscover the best hyperparameter combinations from 10 experimental trials. The trial showing the highest AUC score from the conducted runs became the decided final model configuration.

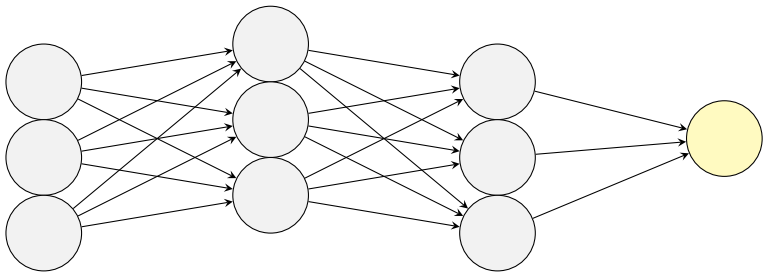


Figure 3.3: ANN architecture.

Note: The number of input neurons shown is illustrative. The actual network includes all preprocessed input variables.

# 4 Results

The findings directly support the main objective of this study - to evaluate the practical performance of traditional and modern machine learning models for credit default prediction in real-world banking context. All the tasks set prior to the methodology section were completed and discussed.

## 4.1 General results

The model assessment included the use of ROC AUC, Gini coefficient, log loss, accuracy, and the confusion matrix as classification metrics. The main evaluation standards consisted of (1) achieving high test Gini performance and (2) minimising overfitting by maintaining small Gini differences between training and test results and (3) ensuring practical deployment capabilities. It is important to highlight that the dataset used in this study is imbalanced, as discussed in the Data Overview section of the methodology. This imbalance in the target variable distribution means that accuracy cannot be reliably used as the primary evaluation metric, because it may provide misleading results. As demonstrated by Akosa [24], accuracy often fails to reflect model performance in such settings. Similarly, A. Cherif and others [25] also reference Akosa [24], emphasizing the importance of choosing performance metrics that are robust to class imbalance, for example, noting the Adjusted F-1 Measure (AGF), which is an improvement of the F-Measure that suits imbalanced data better. In this context, the Gini coefficient was chosen as a key performance metric. For Artificial Neural Network (ANN) only AUC and Gini coefficient were chosen. The Gini coefficient is closely related to AUC and is widely used in credit scoring. As discussed by E.Schechtman and G.Schechtman [26], it is mostly defined as twice the area between the ROC curve and the diagonal line, leading to the formula:

$$\texttt{Gini} = 2 \times \texttt{AUC} - 1$$

In practice, the interpretation of Gini values helps determine whether a model is ready for production use. Credit scorecards typically have Gini

values between 40-60%, while behavior scorecards have higher values, ranging from 70-80% [2]. A Gini value around 50% or above is therefore considered acceptable for credit risk prediction models used in real-world banking environments. This benchmark provides useful context for evaluating the results obtained in this study.

Among the tested models, LightGBM (Optuna-tuned) and Random Forest (Optuna-tuned) showed the best results with test Gini approximately 0.534 and moderate training Gini values (0.58 and 0.68). This suggests they react reasonably well to unseen data and benefit from hyperparameter tuning, despite a minor sign of overfitting in the case of optimised Random Forest. CatBoost Optuna and CatBoost Custom, which is CatBoost using its built-in ability to handle categorical data, follow with similar test scores.

Logistic Regression performed less than average among the other models, with test Gini of 0.462, which is expected given its linear nature and the complexity of the data. This could be used as a useful benchmark in evaluating other performances.

While XGBoost (non-optimised) achieved a high training Gini of 0.835, its test Gini dropped to 0.495, suggesting possible overfitting. The perfect Gini score of 0.999 reached by XGBoost Optuna validated the issue of overfitting when compared against its lowest Gini score of 0.411. Similarly, Random Forest (non-optimised) reached a training Gini of 0.999, yet its test score fell to 0.493, again suggesting overfitting,

The optimised ANN achieved a training Gini of 0.5541 (=55.41%) and a test Gini of 0.5273 (=52.73%). These values indicate a balance between fitting the training data and adapting to unseen data and making this model competitive with the best-performing models in this study.

These results reveal some interesting patterns. In most cases, the optimised models not only performed better on the test set — indicated by an increase in test Gini — but also showed reduced training Gini scores, indicating a step away from overfitting. However, XGBoost presents an exception: while the non-optimised version already showed signs of overfitting, the optimised version (XGBoost Optuna) took this issue further by achieving near-perfect training Gini (0.999) but the lowest test Gini (0.411) among all models. This could suggest that Optuna may have selected overly complex hyperparameters, causing the model to overfit during training. Alternatively, it may reflect that XGBoost's architecture is not suited with the patterns in this dataset.

## 4.2 Model Stacking - Ensemble Learning

As an additional attempt to obtain the best and final model — and out of academic curiosity — a stacking ensemble was created using the `StackingCVClassifier` from the `mlxtend` library. This method combines multiple base classifiers' predictions as inputs and makes a final prediction based on those inputs.

The system includes LightGBM Optuna and Logistic Regression as base learners together with CatBoost Optuna functioning as the meta-device. The decision to choose LightGBM Optuna makes sense because this model demonstrated outstanding performance with low overfitting on new data and it earned selection over simpler Logistic Regression primarily because it relies on a distinct linear model structure. The system selected CatBoost Optuna to function as the meta-classifier for its balanced operational behavior. The main factor behind the model selection focused on both stability and effectiveness.

Among all models studied this approach demonstrated a Gini value of 0.536 which became the best Gini score regarding model performance. The proposed method achieved an 80% overall accuracy level through its confusion matrix which indicated successful non-defaulter identification with precision at 0.83 and recall at 0.94. The model performed poorly in detecting default cases since it produced a default class evaluation score of 0.33. TheClassification is a problem in this application because data distribution between classes is unequal. The minority class F1-score reached 0.43 while the macro average F1-score measured 0.65 even though the default class accuracy stood at 0.33.

Table 4.1: Final Model Performances Based on Gini Coefficient

| **Model** | **Train Gini** | **Test Gini** |
|---|---|---|
| Stacking Model | 0.5854 | 0.5356 |
| LightGBM Optuna | 0.5812 | 0.5339 |
| Random Forest Optuna | 0.6775 | 0.5337 |
| CatBoost Optuna | 0.5760 | 0.5336 |
| CatBoost Custom | 0.6948 | 0.5303 |
| CatBoost | 0.7234 | 0.5294 |
| ANN Optuna | 0.5540 (= 55.4090%) | 0.5272 (= 52.7269%) |
| LightGBM | 0.7224 | 0.5228 |
| XGBoost | 0.8345 | 0.4951 |
| Random Forest | 0.9999 | 0.4927 |
| Logistic Regression | 0.4775 | 0.4620 |
| XGBoost Optuna | 0.9999 | 0.4112 |

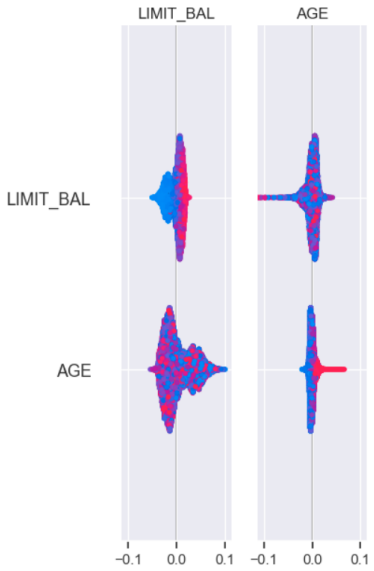


Figure 4.1: Strongest SHAP interaction

## 4.3 Feature Importance

As one of the main tasks of this investigation, feature importance analysis was conducted to identify the variables that contribute most to predicting default. The analysis used tree-based feature importance and SHAP (SHapley Additive exPlanations) values as its main approaches. Machine learning models can be explained through these methods by determining the influence each feature has on the prediction outcome.

SHAP interaction plot particularly focused on features such as `AGE` and `LIMIT_BAL`, implying the strong bond between the two features. This can be interpreted as: older clients are more likely to be assigned higher credit limits due to their longer credit histories and financial stability, whereas younger clients are granted higher limits; therefore, the model could identify this as an increased risk of default. The model analyses the relationship between default prediction outcomes and target variable `LIMIT_BAL` when applied to different client age groups. Financial maturity assessment by the bank determines the credit limits it assigns to its clients. Younger individuals granted high credit limits may represent high risk due to limited financial history. This is shown in Figure 4.1: Strongest SHAP prediction.

The bar chart (Figure 4.2) illustrates normalised feature importances from tree-based models. As expected, features such as `Delay_Count`, `Max_Delay`, and `Delay_Days` are shown as the most influential predictors. These features capture the client's past behaviour in terms of delays, meaning clients with a history of missed or late payments are statistically more likely to default again. The dominance of historic variables such as

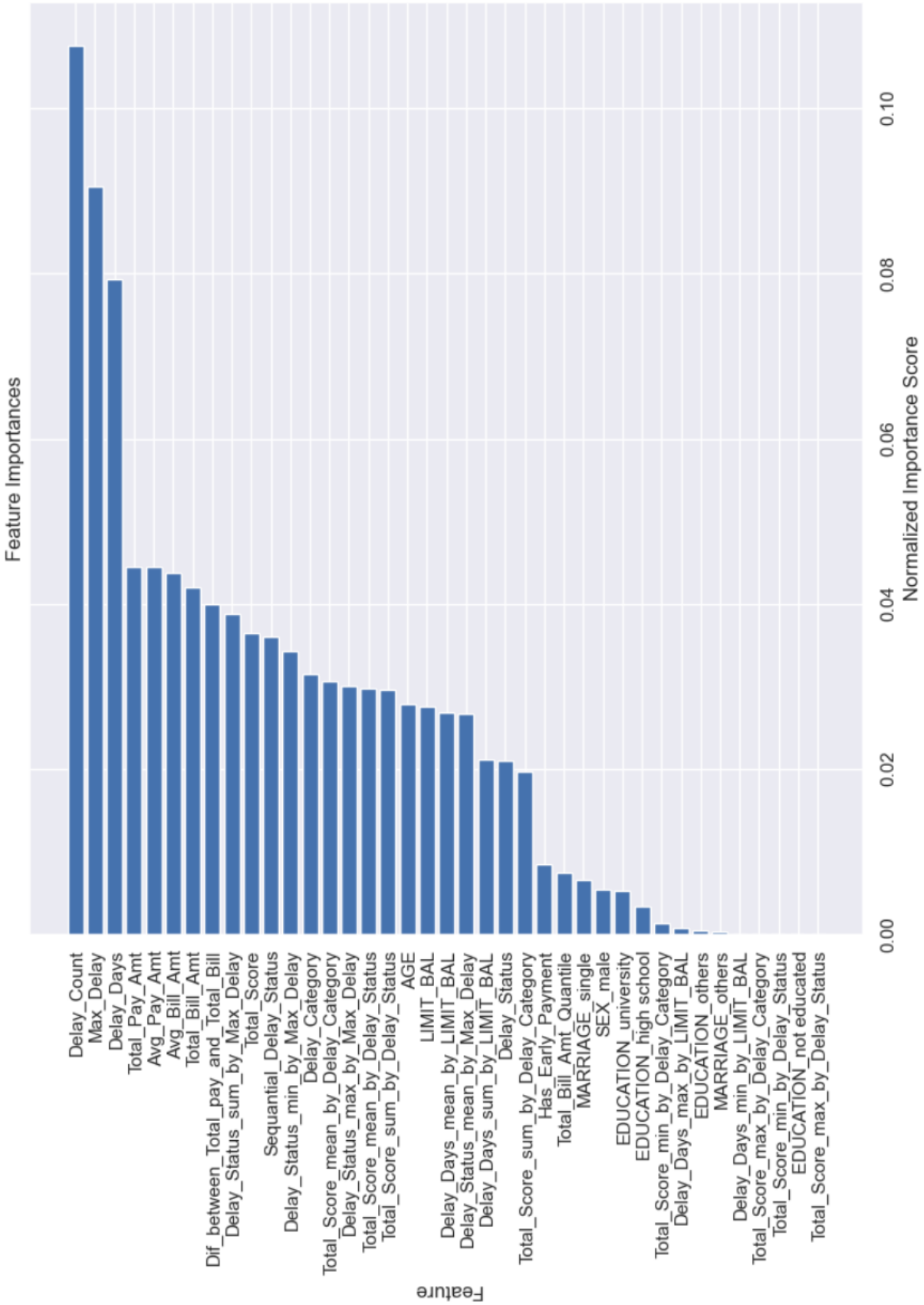


Figure 4.2: Detailed view of feature importance bar chart.

`Delay_Count`, `Max_Delay`, and `Delay_Days` is economically intuitive. Clients with a history of missed or late payments are statistically more likely to default again, aligning with standard credit risk practices, as Zhu et al. [14] especially mentioned in the literature review, where past behaviour is often the strongest predictor of future outcomes.

It is important to note that feature importance measures derived from tree-based models can sometimes be biased toward variables with more levels. In simpler terms, if a feature has a variety of unique values, the model can consider that specific feature very important. The assessment requires understanding that `Delay_Count`, `Max_Delay`, and `Delay_Days` emerge as dominant predictors but their effects should be studied in connection with the model architecture.

# 5 Conclusion

This study explored the effectiveness of various machine learning algorithms for credit default prediction, placing a strong emphasis on model interpretability and performance evaluation. It also involved the systematic steps that were used, starting from data exploration, feature engineering and model tuning to final performance evaluation.

All the models evaluated in this study successfully passed the commonly accepted Gini benchmark of 40%, which is considered the lower threshold for deployment in practical applications. This suggests that depending on data, these models could be considered suitable for real-world use in banking environment.

The final results show that ensemble and tree-based models consistently outperformed simpler statistical approaches. In specific, Stacking Model, LightGBM Optuna and Random Forest Optuna delivered the best results, even though the next following couple of models did approximately the same. The conclusion can be made that a hybrid of Logistic Regression, LightGBM Optuna and CatBoost Optuna is the best-performing classifier, highlighting the value, capability and potential of combining different model architectures.

In addition, this study underlined the importance of model interpretability by using feature importance scores to share the influence of specific variables. Another conclusion was made that in default prediction problem, client's past history of interactions with the bank plays a massive role, while also highlighting the relationship between age and credit limits.

Hyperparameter optimisation through Optuna proved vital for different performance results since it enhanced generalisation skills and minimised overfitting to a certain extent.

The study also properly addressed the sensitivity of Logistic Regression statistical models while delving into several concepts such as data preprocessing and feature scaling and multicollinearity.

Ultimately, this study not only identifies the gap between simple and advanced models, but also contributes to the field by demonstrating how to build a predictive pipeline that is practically applicable in the real banking

sector. By thoroughly evaluating 12 machine learning models under optimised conditions, this project delivers an additional framework that can inform real-world model selection for credit risk assessment. The stacking model, which combined Logistic Regression, LightGBM and CatBoost, achieved the most effective result, outperforming other models. This contribution is valuable in light of the original motivation: banks today must manage credit risk in increasingly complex environments, and this research provides insight into which tools perform best in that context. However, the study is limited by the use of a single dataset without real-time financial data, which could impact the validity of the results.

Future work may focus on expanding the feature space or exploring deep ensemble learning techniques.

# Bibliography


[1] U. M. L. Repository, *Default of credit card clients dataset*, https://www.kaggle.com/datasets/uciml/default-of-credit-card-clients-dataset, Last accessed: 06-11-2024, 2017.

[2] D. Ribeiro, *Understanding the normilized gini coefficient and dedault rate*, https://diogoribeiro7.github.io/statistics/gini_coefficiente/#typical-values, Last accessed: 11-04-2025, 2024.

[3] F. Brown, *What is 'garbage in, garbage out,' and why is it [still] a problem?* https://profisee.com/blog/garbage-in-garbage-out/, Last accessed: 11-04-2025, 2023.

[4] A. Khashman, 'Customer satisfaction quality in banking sector,' *International Journal of Business and Management*, vol. 18, no. 2, pp. 15–25, 2023. DOI: 10.5539/ijbm.v18n2p15.

[5] I. Drigă, 'Financial crisis and bank profitability – the case of romania,' *Annals of the University of Craiova, Economic Sciences Series*, 2013.

[6] P. Sapienza, 'The effects of government ownership on bank lending,' *Journal of Financial Economics*, vol. 72, no. 2, pp. 357–384, 2004. DOI: 10.1016/j.jfineco.2002.10.002.

[7] I. M. Fund, *World Economic Outlook: Financial Stress, Downturns, and Recoveries*. International Monetary Fund, 2008, October edition. [Online]. Available: https://www.imf.org/en/Publications/WEO/Issues/2016/12/31/World-Economic-Outlook-October-2008-Financial-Stress-Downturns-and-Recoveries-22028.

[8] D. Anginer, A. C. Bertay, R. Cull, A. Demirgüç-Kunt and D. S. Mare, 'Bank capital regulation and risk after the global financial crisis,' *Journal of Financial Stability*, vol. 54, p. 100 891, 2021. DOI: 10.1016/j.jfs.2021.100891.

[9] E. I. Altman, 'Financial ratios, discriminant analysis and the prediction of corporate bankruptcy,' *The Journal of Finance*, vol. 23, no. 4, pp. 589–609, 1968. DOI: 10.2307/2978933.

[10] F. Rashid, R. A. Khan and I. H. Qureshi, 'A comprehensive review of the altman z-score model across industries,' *The Business Review*, vol. 27, no. 2, pp. 35–42, 2023. [Online]. Available: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=5044057.

[11] L. Laeven and R. Levine, 'Bank governance, regulation, and risk taking,' *NBER Working Paper No. 14113*, 2008, Available at: https://www.nber.org/papers/w14113. DOI: 10.3386/w14113.

[12] T. Tran, M. Nguyen, Q. Pham *et al.*, 'Explainable machine learning for financial distress prediction: Evidence from vietnam,' *Data*, vol. 7, no. 160, 2022. DOI: 10.3390/data7110160.

[13] M. Nazari and M. Alidadi, 'Measuring credit risk of bank customers using artificial neural network,' *Journal of Management Research*, vol. 5, no. 2, pp. 17–27, 2013. DOI: 10.5296/jmr.v5i2.2899.

[14] X. Zhu, L. Peng, L. Wang, B. Wu and Y. Zeng, 'Explainable prediction of loan default based on machine learning models,' *Data Science and Management*, vol. 6, pp. 123–133, 2023. DOI: 10.1016/j.dsm.2023.04.003.

[15] P. Probst, A.-L. Boulesteix and B. Bischl, 'Tunability: Importance of hyperparameters of machine learning algorithms,' vol. 20, pp. 1–32, 2019. [Online]. Available: http://jmlr.org/papers/v20/18-444.html.

[16] StudySmarter, *Kolmogorov-smirnov test*, https://www.studysmarter.co.uk/explanations/math/probability-and-statistics/kolmogorov-smirnov-test/, Last accessed: 18-03-2025, 2023.

[17] Laerd Statistics, *Spearman's rank-order correlation*, https://statistics.laerd.com/statistical-guides/spearmans-rank-order-correlation-statistical-guide.php, Last accessed: 04-04-2025.

[18] Investopedia, *Variance inflation factor (vif)*, https://www.investopedia.com/terms/v/variance-inflation-factor.asp, Last accessed: 04-04-2025, 2024.

[19] TIBCO Product Documentaton, *Data science documentation*, https://docs.tibco.com/data-science/GUID-44739B00-E85F-4CE7-8404-24F9B775ADE8.html, Last accessed: 04-04-2025.

[20] J. Raymaekers, W. Verbeke and T. Verdonck, 'Weight-of-evidence through shrinkage and spline binning for interpretable nonlinear classification,' *arXiv preprint arXiv:2101.01494*, 2021. [Online]. Available: https://arxiv.org/abs/2101.01494.

[21] S. Chowdhury, Y. Lin, B. Liaw and L. Kerby, 'Evaluation of tree based regression over multiple linear regression for non-normally distributed data in battery performance,' *Proceedings of the 2022 International Conference on Intelligent Data Science Technologies and Applications (IDSTA)*, pp. 17–25, 2022. DOI: 10.1109/IDSTA55301.2022.9923169.

[22] S.-l. developers, *Standardscaler — scikit-learn 1.6.1 documentation*, Last accessed: 04-04-2025, Scikit-learn developers. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html.

[23] B. Krishnamurthy and B. Whitfield. ‘An introduction to the relu activation function.’ Last accessed: 08-04-2025. (2024), [Online]. Available: https://builtin.com/machine-learning/relu-activation-function.

[24] J. S. Akosa, ‘Predictive accuracy: A misleading performance measure for highly imbalanced data,’ in *Proceedings of the SAS Global Forum*, Paper 0942-2017, 2017, pp. 1–12.

[25] A. Cherif, A. Badhib, H. Ammar, S. Alshehri, M. Kalkatawi and A. Imine, ‘Credit card fraud detection in the era of disruptive technologies: A systematic review,’ *Journal of King Saud University – Computer and Information Sciences*, vol. 35, pp. 145–174, 2023. DOI: 10.1016/j.jksuci.2022.11.008. [Online]. Available: https://doi.org/10.1016/j.jksuci.2022.11.008.

[26] E. Schechtman and G. Schechtman, ‘The relationship between gini terminology and the roc curve,’ *Metron*, vol. 77, no. 2, pp. 133–143, 2019. DOI: 10.1007/s40300-019-00160-7. [Online]. Available: https://link.springer.com/article/10.1007/s40300-019-00160-7.