National Economics University

College of Technology

FINAL PROJECT

Predicting Credit Card Defaults:

Leveraging Machine Learning Models for Effective Credit Risk Analysis

Course Instructor: Dr. Thien Van Luong

Group 7 - DSEB 64A

Team Members:

Thu Hang Le - 11222080
Thu Tra Nguyen - 11226306
Anh Tuan Do - 11226687

November 29, 2024

Abstract

Background. Due to an increasing number of credit card defaulters, companies are now taking greater precautions when approving credit applications. When a customer meets certain requirements, credit card firms typically use their experience to decide whether to grant them a credit card. Additionally, a few machine learning methods have been applied to support the final decision.

Objectives. The aim of this study is to evaluate the performance of various machine learning algorithms, including XGBoost (XGB), Neural Network (NN), Ensemble, Random Forest Classifier (RFC), Decision Tree Classifier (DTC), and Logistic Regression Classifier (LRC), in predicting credit card approval.

Methods. This study applies machine learning to predict credit card defaults using the Kaggle "American Express - Default Prediction" dataset. The process includes data collection, preprocessing, feature engineering, and addressing class imbalance with SMOTE. Various algorithms like XGBoost, Neural Networks, Random Forest, Decision Trees, and Logistic Regression are tested, with ensemble learning (stacking) used to improve performance. Models are evaluated using metrics such as accuracy, precision, recall, F1-score, AUC, and Gini Index to identify the best predictor for credit card defaults.

Results. Among the selected machine learning algorithms, both XGBoost and Ensemble models performed the best with an AUC of 0.96 and a Gini index of 0.92. These models outperformed the others, with Random Forest following closely behind with an AUC of 0.95 and a Gini index of 0.90.

Conclusions: This research highlights the effectiveness of machine learning algorithms, such as XGBoost, Ensemble methods, and Random Forest, in predicting credit card defaults. The models outperformed traditional scoring methods, with XGBoost achieving the highest AUC, accuracy, and F1-score. While Random Forest and Decision Trees also performed well, Neural Networks showed the highest recall, essential for detecting defaults. The findings emphasize the potential of machine learning to improve credit risk management, offering more accurate and fair assessments, enhancing financial outcomes, and fostering greater inclusivity in credit decision-making.

Keywords: Machine Learning, Credit Default Prediction, AMEX.

Acknowledgments

We would like to extend our profound gratitude to **Dr.Thien Van Luong**, our course advisor and the Head of the Business AI Lab, whose unwavering support and expert guidance have been instrumental throughout this project. Dr.Thien's profound knowledge and expertise in the field of Machine Learning, combined with his genuine commitment to fostering academic growth, have been a constant source of inspiration. His thoughtful feedback, constructive suggestions, and encouragement have significantly shaped our research, ensuring its successful completion. His mentorship has not only elevated the quality of this project but has also enriched our understanding of the transformative potential of machine learning.

We are equally grateful to **Mr.Bach Xuan Mai**, a distinguished member of the DSEB 62 cohort, for his invaluable advice and motivational support. His practical insights into the real-world application of machine learning techniques have added immense value to our project, enabling us to approach challenges with a clearer and more pragmatic perspective. Mr. Bach's willingness to share his knowledge and experiences has greatly enhanced our learning journey, helping us bridge the gap between theoretical concepts and their practical implementation.

Our heartfelt appreciation also goes to **our classmates from DSEB 64A**, whose camaraderie and support have been a cornerstone of our journey. The collaborative atmosphere within our class created an environment that was both intellectually stimulating and personally rewarding. The exchange of ideas, constructive discussions, and shared encouragement among our peers have been pivotal in enriching the depth and breadth of our work.

Lastly, we would like to express our deepest thanks to **the members of our project team** for their remarkable dedication, perseverance, and teamwork. Every team member played a crucial role in the success of this project, contributing to various aspects such as data collection, analysis, model development, and the presentation of our findings. Their relentless efforts, creativity, and unwavering commitment transformed challenges into opportunities and ideas into impactful outcomes. It has been an absolute privilege to collaborate with such a talented and passionate group of individuals.

This project represents not only the culmination of our efforts but also the collective support and inspiration from our mentors and teammates. We are sincerely grateful to everyone who has contributed to this meaningful journey.

Table of Contents

List of Figures

List of Tables

List of Acronyms

1. LRC: Logistic Regression Classifier
2. DTC: Decision Tree Classifier
3. RFC: Random Forest Classifier
4. ML: Machine Learning
5. XGB: XGBoost
6. XBC: XGBoost Classifier
7. AMEX: American Express

Chapter 1

Introduction

1.1 Overview of Credit Default Prediction

The ability to predict credit defaults has become an indispensable aspect of risk management in the financial services industry, particularly within the credit card lending sector. As credit cards have grown to become a staple of modern consumer lifestyles and financial transactions, accurately forecasting defaults is critical for minimizing financial losses, preserving profitability, and maintaining a competitive edge in the market. In recent years, the rising rates of credit card defaults, coupled with heightened competition among financial institutions, have further underscored the need for precise and proactive credit risk prediction methods. These efforts are not only vital for safeguarding institutional stability but also for fostering customer trust and long-term loyalty.

The unique characteristics of the credit card industry pose distinct challenges and opportunities for credit default prediction. Unlike other forms of credit that may involve secured loans with tangible collateral, credit cards typically operate on unsecured credit. This means that in cases where cardholders fail to meet their repayment obligations, banks are left with no assets to recover their losses. This absence of collateral magnifies the importance of establishing robust systems for predicting and mitigating credit risk. Furthermore, while individual credit card accounts often have relatively small credit limits, the scale of credit card issuance—covering millions of customers—creates an extensive pool of data ripe for analysis. This data includes transactional histories, spending behaviors, repayment patterns, and credit histories, offering a comprehensive foundation for developing sophisticated predictive models.

Traditionally, credit risk assessment relied heavily on manual evaluations by analysts using frameworks like the 5Cs model—Character, Capital, Collateral, Capacity, and Conditions. Analysts would carefully review loan applications and make decisions based on these five factors, using their judgment to approve or reject credit requests. While effective to an extent, this approach was inherently subjective and prone to inconsistencies, as decisions could vary depending on the individual analyst's interpretation. Moreover, the manual nature of this method limited its scalability, making it less suited for handling the vast datasets generated by the modern credit card industry.

In response to these limitations, the financial services industry has increasingly turned to advanced technological solutions, particularly machine learning models, to enhance the accuracy and efficiency of credit default predictions. Machine learning leverages vast amounts of data and sophisticated algorithms to uncover patterns and correlations that traditional methods may overlook. Unlike human-driven evaluations, machine learning models are data-driven, objective, and capable of analyzing millions of data points simultaneously. This allows financial institutions to detect potential default risks earlier, respond to changes in customer behavior more effectively, and optimize credit decision-making processes.

One of the key advantages of machine learning in credit default prediction is its ability to incorporate diverse and unconventional data sources into its analysis. For instance, in addition to traditional credit metrics such as repayment histories and credit scores, machine learning models can analyze behavioral data, including customer spending habits, transaction frequencies, and even social and online behavioral indicators. This holistic approach enables these models to generate insights that are not only more accurate but also more comprehensive, providing financial institutions with a deeper understanding of their customers and the broader market environment.

The practical impact of machine learning adoption in credit risk management is already evident in the operations of leading financial institutions. For example, JPMorgan Chase, one of the largest banks in the United States, has integrated machine learning technology across various aspects of its credit analysis processes. By utilizing these advanced models, the bank has been able to predict defaults with greater precision and uncover hidden patterns in credit behaviors. According to a 2019 report by the institution, the use of machine learning models reduced default rates by an estimated 5-10% compared to traditional assessment methods. Such results not only highlight the effectiveness of machine learning but also demonstrate its potential to drive substantial cost savings and improve operational efficiency in the financial services sector.

The importance of predictive credit default modeling extends beyond risk mitigation. It also plays a pivotal role in enabling financial institutions to navigate increasingly dynamic and uncertain market conditions. For example, during periods of economic volatility, such as the global financial crisis or the COVID-19 pandemic, traditional risk assessment methods may struggle to adapt to rapidly changing circumstances. Machine learning models, on the other hand, are designed to quickly process new data and adjust predictions accordingly, ensuring that financial institutions remain agile and responsive in the face of economic shocks.

In conclusion, the evolution of credit default prediction methodologies reflects the growing need for more accurate, scalable, and data-driven approaches in the financial services industry. Machine learning models represent a significant leap forward in addressing these requirements, offering unparalleled capabilities for risk assessment and decision-making. As the credit card sector continues to expand and diversify, leveraging these advanced tools will be essential for financial institutions to maintain their competitiveness and secure their position in an ever-changing economic landscape.

1.2 Research Motivation

The motivation for this research lies in the growing need to enhance credit risk analysis by leveraging machine learning models to address challenges in financial decision-making, risk management, and customer experience. Credit cards play a critical role in the financial lives of millions, acting as a bridge for consumers between their needs and financial stability. However, the increasing prevalence of defaults poses risks not only to financial institutions but also to the broader economy. Accurate credit default prediction is essential for minimizing losses, building trust, and fostering long-term customer relationships.

A major driving force for this study is the imperative to improve financial decision-making processes. Traditional methods of assessing creditworthiness often rely on static models and limited data, which may fail to capture the complexities of consumer behavior or adapt to rapidly changing market dynamics. Machine learning models, with their ability to process vast amounts of data and uncover nuanced patterns, offer the potential to revolutionize credit evaluation practices by making them more accurate, adaptive, and equitable.

Effective risk management is another key motivation for this research. The financial services sector is grappling with rising credit card default rates amid economic uncertainty and the inherent risk of unsecured lending. Traditional risk assessment frameworks often fall short in providing timely and precise insights, leaving institutions vulnerable to unanticipated losses. This research aims to bridge that gap by introducing advanced predictive models that can identify high-risk borrowers early, optimize resource allocation, and strengthen the resilience of financial systems in an ever-changing economic landscape.

Moreover, this study is driven by a desire to enhance the customer experience. Conventional credit evaluation often applies rigid criteria, leading to credit denial or unfavorable terms for individuals who might otherwise demonstrate repayment capability under alternative considerations. By utilizing machine learning techniques, financial institutions can offer more personalized and fair credit solutions, enabling better access to financial services for a broader range of customers. This not only benefits consumers but also contributes to greater inclusivity and fairness within the financial ecosystem.

In addressing these challenges, this research seeks to contribute to a more sustainable, resilient, and customer-centric approach to credit card risk management.

1.3 Research Objectives

The primary objective of this study is to develop a robust predictive model capable of identifying credit card default events with high accuracy. By utilizing advanced machine learning techniques, this research aims to significantly enhance the effectiveness of credit risk management processes at American Express. Accurate default prediction is critical for financial institutions, as it enables timely and informed decisions regarding credit approvals, limits, and collection strategies. Such improvements can lead to reduced financial losses, improved customer segmentation, and more efficient allocation of resources. The outcomes of this study will contribute to refining the credit decision-making process, fostering both business growth and customer trust.

The first specific goal of this study is to evaluate and compare the performance of various machine learning algorithms to identify the most suitable model for predicting credit card defaults. This evaluation will include traditional techniques such as Logistic Regression, along with more complex models like Logistic Regression, Random Forests, XG Boost, Ensemble Model and Neural Networks. Each model will be assessed based on key performance metrics, including accuracy, precision, recall, AUC, Gini index and F1-score, ensuring a comprehensive understanding of their strengths and weaknesses. By comparing these algorithms, the research aims to establish a benchmark for predictive performance and determine which approaches provide the best balance between complexity and interpretability.

A second objective involves optimizing the chosen machine learning models to maximize their predictive capabilities. Optimization techniques such as hyperparameter tuning, feature selection, and cross-validation will be employed to enhance model accuracy and generalizability. Feature selection will help reduce noise in the data and highlight the most relevant factors influencing credit card defaults, while cross-validation will ensure that the models perform well across diverse subsets of the dataset. These steps will help refine the predictive models and ensure their robustness in practical applications.

Machine learning models offer substantial advantages over traditional credit scoring methods due to their ability to detect complex and nonlinear relationships within data. While traditional models like linear regression are limited by their assumption of linearity and independence among predictors, machine learning algorithms can capture intricate interactions between features. For example, Random Forests are well-suited for handling high-dimensional datasets and identifying feature importance, while Neural Networks excel at modeling nonlinear dependencies and capturing subtle patterns in large volumes of data. These capabilities enable machine learning models to provide a more nuanced and accurate assessment of credit risk, which is crucial in today's dynamic financial environment.

By harnessing the power of machine learning, this research aspires to enhance the accuracy and reliability of customer classification into ”low-risk” and ”high-risk” categories. Such improvements can help American Express reduce the likelihood of defaults, optimize its credit allocation strategies, and ultimately maintain a competitive edge in the financial industry. Beyond immediate financial benefits, the successful implementation of these models has the potential to foster a more data-driven, customer-centric approach to credit management, benefiting both the company and its clients.

1.4 Research questions

RQ. Which among the selected machine learning algorithms (XGBoost, Neural Networks, Random Forest Classifier, Decision Tree Classifier, and Logistic Regression Classifier) is the most efficient model in predicting credit card default events?

Chapter 2

Background

2.1 Machine Learning

Machine Learning is a branch of artificial intelligence that enables computers to identify patterns in data and make predictions based on the patterns [30]. It is also referred to as predictive analytic or statistical learning. This includes developing statistical models and algorithms that can evaluate data, learn from it, and use what they learn to predict patterns in new data. Healthcare, finance, retail, and manufacturing are just a few of the areas where Machine Learning is making a big impact [31]. The three categories of these Machine Learning models are semi-supervised learning models, unsupervised learning models, and supervised learning models.

Supervised Learning: Supervised learning is an intuitive approach to machine learning. Just as people learn from their experiences and mistakes to improve their skills, machines also need to be provided with relevant data and criteria to evaluate their performance. This is the basis of supervised learning, where the machine is 'supervised' by providing examples of what constitutes success or failure for the task at hand [39].

In supervised learning, input and output variables are mapped using mathematical methods, and this mapping is used to predict outputs on data that the machine has not yet seen. The accuracy of this mapping determines the success of the algorithm. If the mapping is inaccurate, appropriate corrections are applied to ensure that learning can continue.

Supervised learning can be applied in various use cases, but it expects two basic forms of output. As a result, there are two types of supervised learning based on the expected output type, which are:

- **Classification:** is a technique that involves predicting categorical class labels for new data points based on patterns learned from historical observations. The main objective of classification is to accurately assign class labels to unseen data instances. This enables a wide range of tasks such as spam detection, sentiment analysis, and medical diagnosis. By using labeled datasets containing input features and corresponding class labels, classification algorithms aim to discern patterns and relationships within the data to make accurate predictions.

- **Regression:** is a technique that focuses on predicting continuous numerical values rather than categorical labels. The primary aim of regression is to develop models that accurately estimate the relationship between input features and continuous target variables. Regression models are constructed based on datasets containing input features and corresponding continuous target values. These models aim to predict numerical outcomes for new data points, facilitating tasks such as stock price prediction, demand forecasting, and temperature modeling.

Unsupervised Learning: Unsupervised Machine Learning is the process of training algorithms on unbalanced data. There are no stated goal labels to predict in unsupervised learning, and the algorithm must identify hidden patterns and structures in the data on its own. The purpose of unsupervised learning is to detect patterns or structures in the data by identifying similarities and differences in the input data and grouping comparable data points. Clustering, dimensionality reduction, and association rule learning are examples of unsupervised learning approaches.

Unsupervised algorithms are pivotal in machine learning for handling unlabeled datasets [33], which lack explicit output variables. Their primary role is to uncover hidden patterns and structures within unstructured data, enabling businesses to gain deeper insights. Key functions include clustering similar data points and reducing dimensionality to simplify complex datasets. By facilitating exploratory data analysis, unsupervised algorithms empower organizations to make informed decisions and drive innovation. There are two main categories of unsupervised learning that are mentioned below:

- **Clustering:** Clustering is a method of organizing data points into groups based on their similarity. This technique is helpful in identifying patterns and relationships in data without requiring labeled examples.

- **Association:** Association rule learning is a technique used to discover relationships between items in a dataset. It identifiesA rules that indicate the presence of one item implies the presence of another item with a specific probability.

Semi-supervised Learning: Supervised learning is computationally expensive and is applicable only to a limited range of domains. On the other hand, unsupervised learning has a broader range of applications but is not as effective in solving problems. Semi-supervised learning is a technique that overcomes these learning constraints by using both labeled and unlabeled data. In this approach, there are several sets of unlabeled data and only a few sets of labeled data. Unsupervised learning methods are used to group similar data, and the labeled dataset is used to convert the unlabeled data into labeled data. It is further categorized into:

- **Pure semi-supervised:** It is an open-world assumption and datasets are considered not as test samples.

- **Transductive learning:** It is a closed-world assumption in which the datasets are test samples on which performance is optimized.

Reinforcement Learning: Reinforcement learning is a crucial concept in the field of machine learning. It encompasses a problem, a class of solution methods, and a vast area of study. At its core, the idea of reinforcement learning is to learn to maximize cumulative rewards by interacting with an uncertain environment through trial and error. Unlike supervised learning, where labeled examples guide the learning process, and unsupervised learning, which focuses on discovering hidden structures within data, reinforcement learning navigates a unique terrain where actions impact not only immediate rewards but also future situations and rewards. This interplay between exploration and exploitation is intrinsic to reinforcement learning, posing a significant challenge that requires a delicate balance to achieve optimal decision-making. Furthermore, reinforcement learning emphasizes the importance of considering the larger context and uncertainty inherent in real-world environments.

2.2 Machine Learning Algorithms

The Machine Learning algorithms utilized in this thesis will be explained in this section.

2.2.1 Logistic regression classifier (LRC)

The Logistic regression model is a statistical method, generally employed when we are in the presence of a classification problem. Similarly to linear regression, we want to understand the relationship between a dependent variable and one or more independent variables. However, the main difference lies in the fact that in this case, we want to predict a categorical output variable y, instead of a continuous output variable.

The logistic model is given by the following formula:

$$f(x) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x)}} \tag{1}$$

Where $f(x)$ represents the probability of an output variable, which in this context is 0 or 1 (two classes: no default or default). β_0 can interpreted as the value of the interception and β_1 can be interpreted as the regression coefficient, which is multiplied by the value of the predictor.

Equation (1) represents a modification of the linear regression, a monotonic modification. This function allows the outputs to take binary values (zero and one) but at the same time, it enables us to preserve linearity. The sigmoid function (logit function) is able to map the values resulting from a linear regression into a value between 0 and 1. The relationship between linear and logistic regression can be depicted by equation (2) below, which can also know as a logit function (log of odds).

$$\frac{f(x)}{1 - f(x)} = e^{(\beta_0 + \beta_1 x)} \tag{2}$$

In the context of the present thesis we are dealing with binary logistic regression, in this approach, the dependent variable has a dichotomous nature, i.e. it has only two possible outcomes (default or no default). In this case, if a client is predicted to default the output value should be equal to one, if a client is predicted to not default, then the output value should equal zero. This is represented below in the form of a logistic equation.

$$P = P(\text{loan status is 1}) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x + \ldots + \beta_k x_k)}}$$

where k is the number of features (independent variables). The logit function is given below:

$$\frac{f(x)}{1-f(x)} = e^{(\beta_0+\beta_1 x+\ldots+\beta_k x_k)}$$

where $f(x)$ is the probability of the loan being of the nature default. Thus, it is possible to conclude that $1-P$ is the formula for the loans that are of nature non-default.

2.2.2 Decision Tree Classifier

A decision tree is an algorithm that can be seen as a flow chart, and as the name suggests, that flow chart has the structure of a tree. Several elements can be identified in a decision tree model. The first element that is possible to detect is the top node. The top node in a tree is called the root node and from there, a path is traced until a leaf node is reached. The second element is the internal nodes, also referred to as non-leaf nodes. These nodes represent the test on an attribute, for instance, "Is x higher than 0.64?", which can be seen in the tree below. And finally, the last element is the terminal nodes, also known as leaf nodes, the leaves hold a class label (the outcome of a test). The leaf nodes allow us to evaluate the discriminatory power of the tree. The more homogeneous the leaves are, the better the model performs. On the contrary, if the leaves tend to be heterogeneous, that means that the model doesn't separate the output classes well. An example of a decision tree with a two-dimensional split feature space can be seen in Figure 1. The terminal nodes in the tree below are called leaves and correspond to the predicted outcomes.

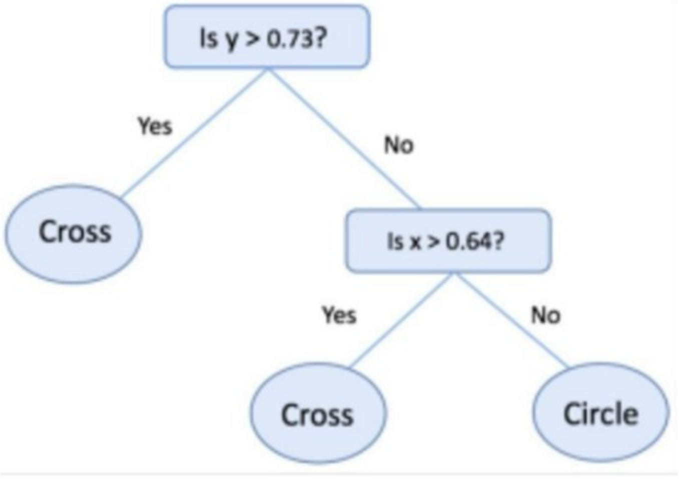


Figure 2.1: Decision tree with two classes

As mentioned above the internal nodes detone the test on an attribute, but the choice of this attribute is a key factor when constructing a decision tree model because it will affect its performance. If the dataset is composed of n attributes, deciding which attributes to select for each split should be done carefully and not just by random selection. Entropy, Information Gain, Gini index, and Gain Ratio are among some of the solutions that have been proposed by researchers to tackle this problem. These methods will calculate a value for every single one of the possible attributes. The values are sorted, and attributes are placed according to the correct order i.e., in case of information gain the attribute with the highest value is selected for the first split and is placed at the root.

2.2.3 Neural Network

A neural network (NN) allows non-linear modeling by setting up a flexible structure of connected layers of neurons. The time-lagged features (input layer) are passed to one (or more) so-called hidden layer(s) that non-linearly transform the input and feed the derived features onto an output layer. The schematic representation of an Neural Network architecture compared to Logistic Regression is displayed in Figure 2.2 (see, e.g., Hastie et al. (2009)). Panel A shows the relation between the P D and the input feature vector x of length F for LR, where x enters linearly. The same input is used in the NN (Panel B), but now the input layer is connected to the output layer via two hidden layers $h(1)$ and $h(2)$ with n and m neurons, respectively, allowing non-linear transformations in the hidden layers. For default prediction, typically a logistic output function is used in analogy to logistic regression.

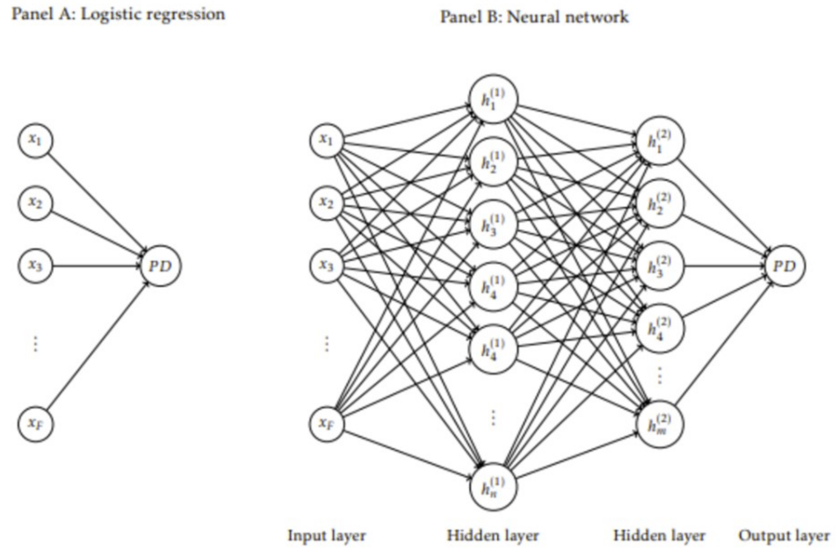


Figure 2.2: Neural Network architecture compared to Logistic Regression

We can now discuss the training process of a neural network. It is organised into epochs, each consisting of forward and backward passes of the entire training set through the network. In order to make the process more efficient, the data are typically split into smaller batches.

Forward Pass: During this phase, the parameters of the network are fixed. Each batch is thus simply forwarded through the network. That is, the output yˆ of the model is computed based on these inputs. It is compared to the expected output y in a loss function L(ˆy, y) which must be minimized. In the case of our own binary classification model, discussed later, we will use the binary cross-entropy (BCE) loss, combined with a regularisation term. The BCE is expressed as:

$$l(\hat{y}, y)_i = -\left[y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)\right]$$

Note that here, we have set our two classes to correspond to the values 0 and 1. This means that exactly one term among y_i and $(1 - y_i)$ will be equal to 1, and the other one to 0. If the prediction yˆi corresponds perfectly to y_i, then $l(\hat{y}, y)_i = 0$. In the opposite case, where $\hat{y}$ predicts the opposite class, $l(\hat{y}, y)_i$ tends to $+\infty$. Intermediate values could also be obtained if the model assigns a non-zero probability for the input to belong to each of the two classes.

As for the regularisation term, it is used to prevent the model from becoming too complex. Let Θ denote the set of all the parameters in the model. Adding the regularisation term, the loss function can finally be expressed as

$$L(\hat{y}, y) = l(\hat{y}, y) + \lambda \|\Theta\|,$$

where λ is a meta-parameter, and we use $\|\Theta\|$ as a general expression of the complexity of the model. For example, we could choose $\|\Theta\| = P2$. By including such a regularisation term in the loss function that we are minimizing, we can force some parameters to zero, hence favoring simpler models. The λ meta-parameter then helps to find a balance between simplicity and expressiveness.

Backward Pass: Once the loss $l(\hat{y}, y)$ based on the current batch has been extracted at the end of the network, it is used to update the parameters of the model. While many variants exist, they mostly revolve around the concept of gradient descent, which ultimately aims at minimizing the loss, and works as follows. In order to update any given parameter θ, which was used to compute the prediction $\hat{y}$, we compute $\partial\partial\theta L(\hat{y}, y)$, the partial derivative of the loss with respect to θ 2 . This parameter is then updated in the following way:

$$\theta \leftarrow \theta - \eta \frac{\partial}{\partial\theta} L(\hat{y}, y),$$

where η denotes the learning rate. A more advanced update rule is discussed later in this section. But first, we take a look at how to compute $\partial\partial\theta L(\hat{y}, y)$.

The computation of the gradient of the loss with respect to the output(s) of the network is immediate from the definition of the loss function. Afterward, we rely on the principle of backpropagation, which is based on the well-known chain rule. Indeed, neural networks intrinsically have a hierarchical structure, with the outputs of one layer serving as inputs for the next one. Therefore, the gradient of the loss with respect to the parameters and inputs of a layer can be computed from those of the next layer, through simple multiplication by the gradient of the function that relates them. This way, the gradient of the loss performs a backward pass through the network, updating every parameter of the model on its way by using the above rule.

We finish our presentation of the training process by discussing another update rule for the parameters, namely using the Adam optimizer which is the one that will be used in our own model. This optimizer operates in time steps, looping until convergence. At each time step t, the vector of parameters is updated as follows:

$$\Theta_t \leftarrow \Theta_{t-1} - \eta \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon}.$$

In this expression, ϵ is a fixed parameter (typically $\epsilon = 10 - 8$), while $\hat{m}_t$ and $\hat{v}_t$ are computed from gt, the gradient at time t. They are respectively estimates of the mean and uncentered variance of gt, and have both been corrected to not be biased towards zero. The Adam optimizer is widely used for its efficiency and scalability

2.2.4 Ensemble Learning

Ensemble learning is an ML technique that combines two or more different ML models. It is a typical ML method that may be applied to both supervised and unsupervised learning models, which are utilized in a wide range of applications. There are three different types of ensemble learning models:

- **Bagging (Bootstrap Aggregating):** In bagging, multiple models are trained independently on random subsets of the training data, with replacement. The models are then combined by averaging their predictions or using a voting mechanism.
- **Boosting:** In boosting, multiple models are trained sequentially, with each subsequent model attempting to correct the errors of the previous model. The final prediction is a weighted combination of the predictions of all the models.
- **Stacking:** Stacking involves training multiple models and using their predictions as input features for a meta-model that produces the final prediction. This approach combines the strengths of various models to produce accurate predictions.

2.2.5 XGBoost Classifier (XBC)

As this thesis focuses on spline models but also includes a comparison with XGBoost, this section provides a brief introduction to the theory behind it. Short for **eXtreme Gradient Boosting**, XGBoost is a powerful and efficient implementation of gradient tree boosting tailored for both regression and classification problems, developed by Chen and Guestrin (2016). XGBoost is an ensemble method that combines the predictions of multiple regression trees to make a final prediction. Each tree in the ensemble contributes an additive function $f_k(x)$ to the final prediction $\hat{y}_i$. The prediction for a given input x_i is:

$$\hat{y}_i = \eta \sum_{k=1}^{K} f_k(x_i)$$

where η is the learning rate, reducing the impact of each individual tree, and K is the number of trees (M. Zou et al., 2022). To ensure the model generalizes well and avoids overfitting, XGBoost incorporates a regularized objective function, $L(\phi)$, defined as:

$$L(\phi) = \sum_{i=1}^{n} l(\hat{y}_i, y_i) + \sum_{k=1}^{K} \Omega(f_k)$$

This objective function combines a loss term, $l(\hat{y}_i, y_i)$, which measures the error between the predicted and actual values — typically the Mean Squared Error (MSE) for regression tasks — with a regularization term that penalizes the complexity of the model. The regularization term, $\Omega(f_k)$, is applied to each tree f_k to control its complexity, and is defined as:

$$\Omega(f) = \gamma T + \frac{1}{2}\lambda \|w\|^2$$

Here, γ penalizes the total number of leaves in the tree, T, and λ controls the L2 regularization on the leaf weights (w). This regularization helps to keep the model simple, preventing overfitting by penalizing complexity and the influence of individual data points.

XGBoost trains the model additively. Starting with an initial prediction, it iteratively adds new trees to improve the prediction. At each iteration t, a new tree f_t is added to minimize the objective:

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

where $\hat{y}_i^{(t-1)}$ is the prediction from the previous iteration, and f_t is the new tree being added.

To optimize this objective, XGBoost uses a second order Taylor approximation, simplifying the optimization process and allowing for the calculation of optimal weights for the tree leaves. For a fixed tree structure, the optimal weight of leaf j is given by:

$$w_j = -\frac{\sum_{i \in I_j} g_i}{\sum_{i \in I_j} h_i + \lambda}$$

where I_j represents the set of data points that fall into leaf j, g_i is the gradient and h_i is the hessian of the loss function with respect to the prediction from the previous iteration $\hat{y}_i^{(t-1)}$. The respective optimal value of the regularized objective is:

$$L^{(t)} = -\frac{1}{2} \sum_{j=1}^{T} \frac{(\sum_{i \in I_j} g_i)^2}{\sum_{i \in I_j} h_i + \lambda} + \gamma T$$

This value serves as an evaluation metric for the decision tree. It combines the fit of the tree to the data, reflected by the sum of gradients and hessians of the loss function, with the complexity of the tree, moderated by the regularization parameters λ and γ. A lower score indicates a better tradeoff between accuracy and simplicity, signifying a more optimal tree structure. A greedy algorithm is typically used to build the tree, starting from a single leaf and iteratively adding branches to minimize the loss function while considering the regularization term. Specifically, each potential split is evaluated based on the reduction in the loss function, or gain, defined as:

$$\text{Gain} = \frac{1}{2} \left[\frac{(\sum_{i \in I_L} g_i)^2}{\sum_{i \in I_L} h_i + \lambda} + \frac{(\sum_{i \in I_R} g_i)^2}{\sum_{i \in I_R} h_i + \lambda} - \frac{(\sum_{i \in I} g_i)^2}{\sum_{i \in I} h_i + \lambda} \right] - \gamma$$

where I_L and I_R are the subsets of data points for the left and right leaves after the split (children), and $I = I_L \cup I_R$ represents the parent leaf. Thus, if the gain of the split is lower than γ, the branch is not created. For further details of the XGBoost algorithm, see Chen and Guestrin (2016).

To use XGBoost, certain hyperparameters need to be set by the user:

- **Learning rate (η)**: Controls the step size at each iteration while moving towards a minimum of the loss function. A smaller value makes the model more robust but requires more boosting rounds.

- **Max depth**: The maximum depth of a tree. Increasing this value makes the model more complex and more likely to overfit.

- **Number of estimators (K)**: The number of boosting rounds. More boosting rounds improve the model's performance but also increase computation time and may lead to overfitting.
- **Gamma (γ)**: The minimum loss reduction required to make a further partition on a leaf node of the tree. The larger gamma is, the more conservative the algorithm will be.
- **Min child weight**: The minimum sum of instance weight (hessian) needed in a child. Ensures that any node being split contains a sufficient number of data points. A larger value prevents overfitting.
- **Lambda (λ)**: L2 regularization term on weights, helping to reduce overfitting.
- **Alpha (α)**: L1 regularization term on weights.

These hyperparameters enable the user to tune the model according to the complexity and size of their data, to find a balance between bias and variance. One popular method for finding the optimal hyperparameters is **GridSearchCV** (M. Zou et al., 2022). By defining a range for each parameter, training a model for each configuration, and then evaluating each combination using cross-validation, the optimal set of hyperparameters can be determined.

2.3 Loss Function and Optimization Techniques

2.3.1 Loss Function

The Cross Entropy Loss function is utilized as our loss metric. Cross Entropy Loss is particularly suited for multiclass classification tasks, as it measures the performance of a classification model whose output is a probability value between 0 and 1. It effectively captures the distance between the model's predicted probability distribution and the true distribution, with a lower loss indicating a model that's more accurate.

This loss function is advantageous because it penalizes incorrect classifications more severely when the model is confident about its incorrect predictions, thus driving the model towards more accurate and confident classifications. Cross Entropy loss for multiclass classification is calculated as follows:

$$CE = -\sum_{c=1}^{M} y_{o,c} \log(p_{o,c})$$

where M is the total number of classes. The summation runs over all classes, where $y_{o,c}$ is a binary indicator (0 or 1) that is 1 if class label c is the correct classification for observation o, and $p_{o,c}$ is the predicted probability that observation o belongs to class c.

2.3.2 Optimizer

For optimization, we selected the Adam optimizer, which stands for Adaptive Moment Estimation. Adam is renowned for its efficiency in handling sparse gradients and its adaptability to the problem's scale, thanks to its moment-based approach [66]. It combines the advantages of two other extensions of stochastic gradient descent: AdaGrad, which works well with sparse gradients, and RMSProp, which works well in online and non-stationary settings. Adam achieves this by maintaining a learning rate for each parameter, which it adjusts as learning progresses. This optimizer starts with an initial learning rate of 0.001, a commonly used value that strikes a balance between training speed and stability, allowing the model to converge efficiently without overshooting optimal solutions.

2.4 Hyperparameter Tuning

In our study, we used GridSearchCV to automate the process of finding the best parameters to improve the performance of our machine-learning models. GridSearchCV works with the basic concepts of cross-validation and parameter grid exploration. GridSearchCV works by defining a grid, which is a range of possible values for each hyperparameter to be adjusted. The method then evaluates all combinations of the defined values using cross-validation techniques, such as k-fold cross-validation, to determine the combination that gives the best performance on the training data. The result is an optimized model with the most suitable hyperparameters for a particular dataset. GridSearchCV helps automate the process of finding the best parameters to maximize the performance of machine learning models. GridSearchCV works with the basic concepts of cross-validation and parameter grid exploration. Some mathematical elements underlie the way GridSearchCV works. Suppose there are n parameters to be tuned, and each parameter has multiple value options (Eq. (4)).

Where p_i is the number of possible values for the i^{th} parameter.

$$\text{Total Combinations} = P_1 \times P_2 \times P_3 \cdots \times P_n \tag{4}$$

The data is divided into k folds. For each parameter combination, GridSearchCV performs k iterations where the model is trained on $k-1$ folds and tested on the remaining folds. Then, the equation for GridSearchCV is as follows (Eq. (5)):

$$\text{Average Score} = \frac{1}{k}\sum_{i=1}^{k} \text{Score}_i \tag{5}$$

Average Score is the average value of all evaluation scores obtained from each fold during the training and testing process. This average is used to get an overall picture of how well the model performs with a particular parameter combination. Where $\frac{1}{k}$ is a scaling factor that divides the total score by the number of folds k to obtain the average. $\sum_{i=1}^{k} \text{Score}_i$ is the sum of all evaluation scores (Score_i) obtained from each fold from 1 to k. Each (Score_i) represents the evaluation result of the model on the i^{th} fold.

2.5 Classification Evaluation Metrics

This task is a binary classification problem using an imbalanced dataset. For the evaluation metrics, we adopted the ROC-AUC, which is a standard evaluation metric for binary classification problems. Additionally, the Gini Index was considered alongside ROC-AUC, as it is a commonly used metric in credit scoring problems due to its ability to measure the discriminatory power of a model. The F1 (macro), F1, accuracy, precision, and recall values were also recorded. We prioritized ROC-AUC and Gini Index first, then F1 (macro), and recall, due to the imbalance in the dataset.

- **Accuracy:** is the ratio of correctly predicted instances to the total number of instances. It provides a general sense of how well the model is performing but can be misleading with imbalanced datasets. [3]

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

- **Precision:** is the proportion of true positives to the total number of predicted positives. It measures how often positive predictions are correct, which is crucial when the cost of false positives is high. [19]

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

- **Recall:** is the proportion of true positives to the total number of actual positives. It indicates how often actual positive instances are correctly identified, which is important when the cost of false negatives is high. [19]

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

- **F1 score:** The F1 Score is the harmonic mean of precision and recall. It balances the two metrics and is useful when both false positives and false negatives are important. [6]

$$F1\text{-score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

- **ROC Curve and AUC:** The Receiver Operating Characteristic (ROC) curve plots the true positive rate against the false positive rate at various classification thresholds. The Area Under the Curve (AUC) measures the overall performance, where a higher value indicates better discrimination ability [45].

  True Positive Rate (TPR) is a synonym for recall and is therefore defined as follows:

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

  False Positive Rate (FPR) is defined as follows:

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

- **AUC:** The Area Under the Receiver Operating Characteristic Curve (AUC) is a threshold-independent measure of the discriminatory ability of a classifier. A value closer to 1 means that the classifier can successfully distinguish between classes. A higher value also means higher robustness for different thresholds. It is calculated as the area under the Receiver Operating Characteristic curve, which is generated by sorting the predictions in decreasing predicted probability and calculating the sensitivity and specificity, or mathematically

$$\text{AUC} = \int_0^1 \text{TPR}(\text{FPR})d\text{FPR}$$

where TPR and FPR stand for respectively true positive rate and false positive rate

- **Partial Gini Index:** The partial Gini index is another widely utilized performance measure in credit scoring. It is based on a graph with all observations on the x-axis, ordered by the probability of default decreasing, and on the y-axis the fraction of positive observations (defaults). A perfect Gini curve can be generated: a partly diagonal, partly horizontal line, which would be the case when the model perfectly captures all default cases within a fraction of the population equal to the sample default rate (all defaults at the lower end of the x-axis). For the actual model's Gini curve, observations are ordered by the predicted probabilities of default. This curve will be less steep as not all defaults will correspond to the fraction of equal size with the highest predicted probabilities. A third curve is the random Gini curve: a straight diagonal line corresponding to a model where the probabilities of default are given random predicted 15 values from a uniform distribution. The Gini coefficient is calculated by:

$$\frac{\text{area between the model's and random Gini}}{\text{area between the perfect and random Gini}}$$

The partial Gini coefficient only focuses on a specific (crucial) part of the curve. Here we choose to calculate the PG on cases with the probability of default smaller than 0.4. This area is crucial because the cost of accepting bad loans is generally considered higher than that of rejecting good loans (e.g. Lessmann et al., 2015; Thomas et al., 2002; West, 2000, among others). In practice, the PG is calculated as

$$PG_{0.4} = 2AUC_{0.4} - 1$$

where $AUC_{0.4}$ is the AUC calculated on 40% of the observations with the lowest predicted probabilities.

2.6 Working Environment

This experiment was carried out using Google Colab, a platform for cloud-based computing. The CPU and RAM of the local system will not be overworked by doing this experiment at Google Colab, which offers significant computing power compared to most laptops. This thesis uses several Python libraries, including:

- **Pandas**
  Pandas is an open-source library in Python. It is widely used for data cleaning, manipulation, preprocessing data, and data analysis. Pandas can read data from various sources such as CSV, Excel, SQL, and JSON files and handle large datasets. Pandas provides two main data structures: Series and DataFrame. The Series data structure is a one-dimensional labeled array capable of holding any data type. The DataFrame data structure is a two-dimensional labeled data structure with columns of potentially different types. It is also compatible with other popular libraries in Python such as NumPy, Matplotlib, and Scikit-learn. We used the Pandas library with version 1.3.5 [27].

- **Scikit-learn (sklearn)**
  Scikit-learn, also known as sklearn, is a popular open-source library in Python. It is widely used for data analysis, data preprocessing, and machine learning algorithms. Scikit-learn is built on top of NumPy, SciPy, and Matplotlib. It focuses on performance and scalability. It can handle large datasets and complex machine-learning problems. Additionally, Scikit-learn provides tools for feature selection, feature extraction, and model evaluation. We used Scikit-learn with version 1.0.2.

- **Matplotlib**
  Matplotlib is a popular open-source library in Python. It was created by John D. Hunter in 2003 as a tool for scientific and technical computing. It is widely used for creating high-quality visualizations, including line plots, scatter plots, bar plots, and histograms. It is also compatible with other popular Python libraries, including NumPy and Pandas. We used Matplotlib with version 3.5.3 [38].

- **NumPy**
  Numerical Python, also known as NumPy, is a popular open-source library in Python. It provides support for multidimensional arrays, matrices, and mathematical functions. It is widely used for data analysis, data preprocessing, and numerical operations. One of the advantages of NumPy is its performance, as it is built on top of the C programming language. It is also compatible with other popular libraries in Python such as SciPy and Matplotlib. We used NumPy with version 1.21.6 [27].

Chapter 3

Literatue Review

3.1 Credit Card Default Factors

A preliminary review of the literature indicates that previous studies have identified various factors considered when predicting the risk of default. Dominguez identified that credit limit, gender, education level, marital status, age, previous payment history, the number of bills in the last 6 months, and the amount paid in the last 6 months are important factors when predicting default [24]. The factors identified in this study are crucial because they allow researchers to compare differences between individuals with high and low credit limits, gender, and education levels, thereby narrowing down the key predictors of default. Identifying individual characteristics and differences will help credit card companies easily deny high-risk customers based on specific factors. Another study also suggested that the most important factors when predicting payment default are gender, education level, age, marital status, and credit limit (the amount of credit granted) [35]. Additionally, other important indicators of credit card default include the number of late payments (based on payment status) and the average payment amount (based on payment amounts) [35].

Credit Limit, Education Level, and Age

Through Pearson correlation analysis, the study found that "credit limit" and "education level" are negatively correlated with default [35]. An individual with a higher credit limit and education level is less likely to default, and vice versa; an individual with a lower credit limit and education level is more likely to default. Meanwhile, age was found to have a positive correlation with default [35]. As an individual gets older, their likelihood of defaulting also increases. Therefore, older individuals with lower education levels and lower credit limits tend to default more than younger individuals with higher education levels and lower credit limits.

Gender

Many studies have highlighted the important role of gender in predicting default. For example, Li (2018) observed that women often have lower credit scores due to difficulties in making payments [11]. However, this study has certain limitations, as many women are single, non-white, under 30 years old, and have higher education levels than men. As mentioned earlier, marital status, age, and education level are important predictors of default. Therefore, Li (2018) concluded that single women tend to have slightly lower credit scores than single men with the same demographic characteristics. A recent study by Dunn and Mirzaie (2022) also yielded similar results, showing that women face more challenges in repaying debts and have higher credit utilization rates [20]. Furthermore, the study also indicated that women have a debt stress index approximately 30% higher than men. Therefore, it can be concluded that women tend to carry debt longer, miss payments more often, and have a higher debt stress index, making them more likely to default compared to men.

Marital Status

As discussed earlier, marital status is an important characteristic when predicting credit card default. According to Stolba (2020), married couples typically have twice the amount of debt compared to single individuals [34]. However, single individuals are more likely to have overdue accounts. The reason is that married couples usually pay their credit card bills on time, helping their accounts avoid being recorded as late payments. Therefore, statistically, unmarried individuals are more likely to default on credit card payments.

Amount Paid and Payment Status

The factors of the amount paid and payment status were found to be negatively correlated with default status [35]. This means that an individual with a higher ability to repay is less likely to default on their credit card payments. Additionally, an individual's payment status also affects the likelihood of default; an individual with payment status issues (e.g., pending transactions or declined transactions) is more likely to default. Jain and Jayabalan (2022) further emphasized that this is an expected result, as a person with a better payment history will reduce their risk of default [35].

In conclusion, these factors are considered significant in predicting credit card default. Credit card users typically need to pay their bills on time to avoid late fees and prevent their accounts from being recorded as overdue. However, due to various circumstances, many people are unable to pay their debts on time, which is more common among older women, single individuals, those with below-average education levels, and lower credit limits. Furthermore, an individual's ability to repay debt directly impacts their likelihood of default. While these listed factors are the most common characteristics among credit card customers, they do not apply to every individual with those specific characteristics. Similarly, even though others without these traits may have a lower likelihood of default, it does not mean they will never default. Therefore, using various machine learning techniques to predict credit card default and determining which technique has the highest accuracy is extremely important.

3.2 Machine Learning Techniques Used to Predict Credit Card Payment Defaults

In 2021, Kibria and Mehmet [25] compared the performance of deep learning with logistic regression and support vector machines for predicting the approval of credit cards in their research. The evaluation metrics used in this research are accuracy, precision, recall, and F1 score. The accuracy of deep learning is the highest, at 87.10%. However, the accuracy of the other two classifiers is the same at 86.23%. In the results, it is concluded that the deep learning model performed slightly better than the other two machine learning techniques.

In 2022, Yiran Zhao [42] examined the performance of various machine learning classifiers for predicting credit card approval. The classifiers used in the research are LRC, a linear SVM, and a naive Bayes classifier. The performance of each algorithm was analyzed using balanced accuracy. The balanced accuracy for LRC is around 88.4%, linear SVM is around 89.0%, and the naive Bayes classifier is around 83.4%. The results show that the linear SVM has the best prediction performance among the models.

In 2019, Peiris [26] examined the various machine-learning techniques in their research "Credit Card Approval Prediction using Machine Learning Techniques". They used a credit card approval prediction dataset, which contains 438,510 records. The techniques used in this research are ANN, linear SVM, and nonlinear SVM. The performance of each algorithm is evaluated by the ROC curve and confusion matrix. The confusion matrix is used to find the accuracy, precision, recall, and F1 score. The accuracy for ANN is 0.78%, linear is 0.71% and nonlinear SVM is 0.88%. In the results, it is concluded that nonlinear SVM performed better than ANN and linear SVM.

In 2018, Ipin, Umi, and Bibit [36] performed a comparison of data mining algorithms to predict the approval of credit cards. The algorithms used in their research are artificial neural networks, support vector machines, and logistic regression. In this study, the authors used principal component analysis (PCA) and particle swarm optimization (PSO) to see better results. This shows that the ANN produces an accuracy of 82.6%, the SVM produces an accuracy of 78.7%, and the LRC produces an accuracy of 81.6%. The results show that the ANN performed well when compared to the other machine learning algorithms.

In 2022, Rowell, Lysa, Celinne, and Maricel [9] compared the performance of various machine learning classifiers. In this research, they used an open credit card approval dataset containing 19 variables and 304,356 instances. After data cleaning and dimensional reduction using relief-based feature selection, it is revealed that 12 attributes and 132,492 instances are still present in the dataset. They used 10-fold cross-validation for testing and training data. The algorithms used are RF, KNN, and NN. The results are evaluated using a confusion matrix and a ROC curve. The accuracy of RF is 95.76%, followed by KNN with 94.37%, and finally, NN has 71.56%. The results show that among the three classifiers, the RF has the highest accuracy, precision, recall, specificity, and AUC.

Authors	**Selected Algorithms**	**Efficient Algorithm**	**Accuracy**
Kibria Md and Şevkli Mehmet	Logistic regression, deep learning and support vector machines	Deep learning	87.10%
Yiran Zhao	Linear SVM, LRC and naive Bayes classifier	Linear SVM	89.0%
M. P. C. Peiris	ANN, Linear SVM, and Non-linear SVM	Nonlinear SVM	88.0%
Ipin Sugiyarto, Bibit Sudarsono, and Umi Faddillah	Artificial neural networks, support vector machines, logistic regression	Artificial neural networks	82.6%
Leonard, Rowell, Lysa, Celine, and Maricel	RF, KNN, NN	RF	95.76%
Pathipati Yasasvi and S. Magesh Kumar	XGBoost classifier and Decision Tree	XGBoost classifier	87.96%

Table 3.1: Related works summary

Previously, several works have been done on credit card approval prediction based on comparison within traditional statistical models and deep learning but we did not find any works related to ensemble learning techniques done on this type of data. So by considering this as an issue, we are going to compare whether there is any change in accuracy when the selected techniques are tested with the same dataset.

Upon reviewing the previous studies, it was evident that all of them employed comparable parameters, such as age, gender, marital status, debt, industry, and employment. status, credit score, and income, for their predictions. It was ensured that these identical parameters were also incorporated into their own research for making predictions.

However, traditional statistical models such as LRC, RFC, and SVC have limitations in predicting complex patterns in credit card approval data. Therefore, this thesis aims to compare the accuracy of these models with the ensemble learning bagging technique, which has shown good results in predicting complex patterns in data. By comparing the accuracy of these models, this thesis will provide insights into the accurate model for predicting credit card approval. The findings of this study can have a significant impact on financial institutions and individuals by improving the accuracy of credit card approval prediction.

Chapter 4

Methodology

In this thesis, the methodology is designed to systematically explore and identify the most effective machine learning techniques for predicting credit card approval. The complexity of financial data, combined with the challenge of predicting defaults, requires careful analysis and model evaluation. This investigation utilizes a dataset obtained from the Kaggle competition ”American Express - Default Prediction”, provided by American Express (AMEX), to determine the most accurate classification techniques. The dataset contains various features related to credit card applicants, including customer spending habits, payment histories, and delinquency status.

The methodology follows a general experimentation approach, where selected classification algorithms are trained using training data, validated using a validation set, and their performance evaluated on a testing dataset using metrics such as AUC, Gini Index, accuracy, precision, recall, F1 score, and ROC curve analysis. In this study, ensemble learning techniques, particularly the bagging method, are applied to combine the predictions of multiple models through majority voting, creating an ensemble model. The performance of the ensemble model is then evaluated against the individual models, with the aim of identifying the most accurate classification method for credit card approval prediction.

4.1 Research Design

The research design of this study is a case study. This study reviewed previous research on machine learning methods, factors predicting credit card defaults, and predicting credit card defaults using machine learning methods. Based on this understanding, a case study approach was developed to analyze this real-world situation, predict credit card defaults using available data, understand previous research efforts, and address the issues identified in earlier studies. Based on the scope of the research, one of the previously mentioned machine learning methods was found to be the most accurate when used to predict credit card defaults. In this study, research on machine learning methods was conducted to ensure a thorough understanding of the program. Finally, a theoretical framework was developed to provide solutions to the previous issues related to predicting credit card defaults using machine learning methods. This study was conducted from September 2024 to November 2024.

4.2 Data Collection

The data in this study was obtained from the American Express - Default Prediction competition on the Kaggle platform, a well-known platform that provides open datasets and organizes machine learning prediction competitions. The dataset was provided by American Express (AMEX), a leading financial organization, with the goal of addressing the problem of predicting credit risk. The dataset consists of a total of 190 feature columns and is organized into variable groups as follows:

- D_*: Variables related to payment delinquency (Delinquency variables).
- S_*: Variables describing customer spending behavior (Spend variables).
- P_*: Variables related to payment history and behavior (Payment variables).
- B_*: Variables describing credit balances and credit limit utilization (Balance variables).
- R_*: Variables related to credit risk levels (Risk variables).

Categories of Credit Card Debtors
Delinquency
Spend
Payment
Balance
Risk

Figure 4.1: Data Framework

Some columns in the dataset are defined as categorical variables, including: ['B_30', 'B_38', 'D_114', 'D_116', 'D_117', 'D_120', 'D_126', 'D_63', 'D_64', 'D_66', 'D_68']. The variables in the dataset are continuous, binary, and categorical. The categorical variables are transformed into binary using dummy variables. The response variable is binary and takes the value of 1 if a customer defaulted in the following year and 0 otherwise. Since the observations in the dataset correspond to accepted customers of AMEX, the number of defaults is very low, less than 1%. This results in a highly imbalanced dataset.

4.3 Statistical Analysis

The first step is to perform statistical analysis of the dataset to gain an overview of the variables' statistics, including the number of missing data points and the balance of the dataset. The statistical analysis shows that most data points are missing values, and the dataset is highly imbalanced, meaning the number of defaults accounts for only a small portion of the total observations. To address this issue, the SMOTE method was chosen to oversample the minority class to 50% of the majority class size. This is a common technique used to improve the predictive ability of models when dealing with imbalanced data. At the same time, we also perform further analysis on the variables to handle missing values, making the data more complete and suitable for model building.

Moreover, the dataset contained an extensive amount of variables, some being duplicates and identification variables, e.g. dates of events for customers. An initial manual selection of what variables to include was therefore conducted. Remaining data management followed the flow visualized in Figure 4.2 and will be discussed in more detail below.

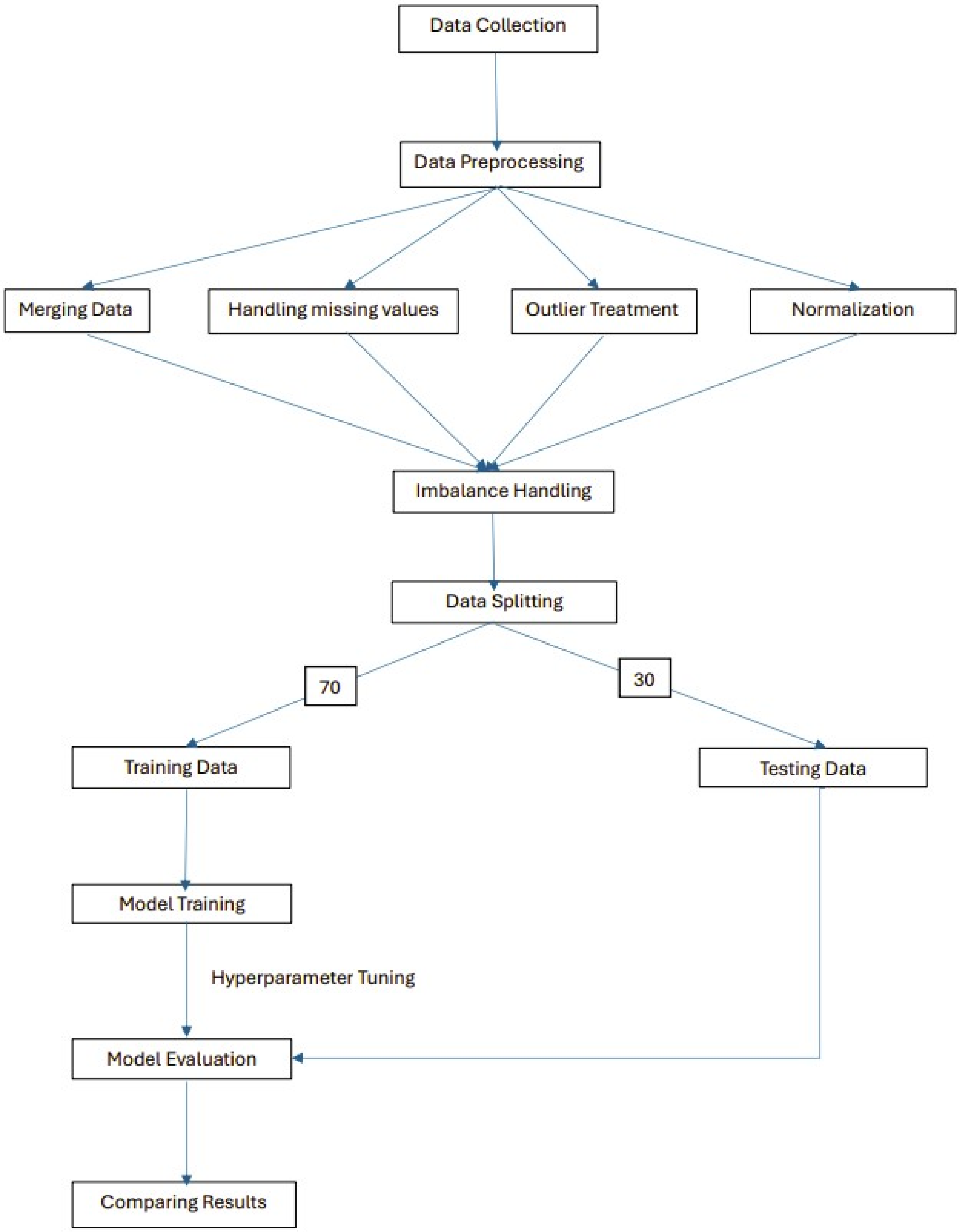


Figure 4.2: Experimentation procedure.

4.4 Data Preprocessing

The first step in the data preprocessing pipeline involves merging various data sources and performing necessary preprocessing steps to prepare the data for subsequent analysis. This includes:

4.4.1 Merging data

The transaction data and default labels are merged using the customer_ID, which serves as the common key between the two datasets. This results in a new dataset with 1,106,348 rows and 191 columns, each row representing a transaction and containing both the target variable and the associated features.

4.4.2 Handling missing values

Missing values are handled using the SimpleImputer method. For numeric columns, missing values are imputed with the mean value of the respective column, while for categorical columns, the most frequent value is used for imputation.

4.4.3 Feature Engineering

After handling the missing values, we proceed to create new features from the original data to improve the model's predictive capability. First, the continuous data columns are identified and converted to the Float64 type to ensure accurate calculations. Next, aggregate features for the continuous data are created by calculating statistical metrics such as the mean, minimum, and maximum values over the past 6 months and 12 months relative to the reference date for each customer.

For categorical data, we calculate the response rate and the number of positive responses for each categorical variable, creating corresponding aggregate features. After the aggregate features for both continuous and categorical data are generated, we combine them into a single DataFrame by joining them based on the customer_ID. Categorical variables are then converted into dummy variables for use in machine learning models.

4.4.4 Imbalance Handling

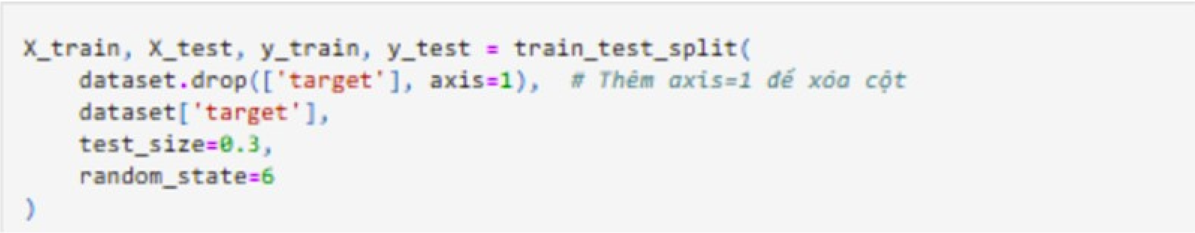
Imbalanced classes are a common problem in many classification projects. They occur when the number of observations representing one class is much lower than the number of observations representing the other classes. Imbalances pose a major issue when the ”cost” of misclassifying the minority class - the class with far fewer observations - outweighs the cost of misclassifying the majority class (or classes), for example, the classification of cancerous cells in medical images [?].

In the case of this project, the cost of misclassifying an applicant that is likely to default on their loan outweighs the cost of misclassifying an applicant that is likely to repay their loan. If an applicant defaults on a loan, the micro-finance company granting the loan loses the loan amount lent and the potential interest that would have been gained on the loan (minus any repayments made). While, if an applicant is simply not granted a loan, then the company will only lose the potential interest that would have been gained if the applicant repaid their loan.

Given the high class imbalance, the SMOTE (Synthetic Minority Over-sampling Technique) method is applied to balance the dataset by generating synthetic samples for the minority class, increasing its size to 50% of the majority class. This helps improve the model's ability to predict the minority class.

4.5 Data Splitting

The dataset was divided into training and testing sets using the train_test_split function from the sklearn.model_selection module. This function randomly splits the data into training and testing subsets based on the specified test size. The training set is used to train the models, while the testing set evaluates the final performance of the selected model. The parameters used in the code are as follows:



```
X_train, X_test, y_train, y_test = train_test_split(
    dataset.drop(['target'], axis=1),  # Thêm axis=1 để xóa cột
    dataset['target'],
    test_size=0.3,
    random_state=6
)
```

Figure 4.3: Data Splitting

The parameters used in the above code are:

- X_train: training features (70% of the data)
- X_test: testing features (30% of the data)
- y_train: training target variable (70% of the data)
- y_test: testing target variable (30% of the data)
- test_size: Determines the proportion of the dataset allocated for testing, set to 0.3 in this case.
- random_state: This parameter is used to set the seed for the random number generator. It ensures that the data is split in the same way every time the code is run.

4.6 Model Training

4.6.1 XGBoost

The XGBoost model was trained with default parameters to determine the importance of the features. We use the feature_importances_ attribute to extract a list of important features. Features with an importance greater than 0.005 (0.5%) are selected for further use in subsequent analysis steps.

To find the best parameters for the XGBoost model, we use GridSearch. This process tests all combinations of parameters and selects the set of parameters that performs best based on evaluation metrics such as AUC (Area Under the Curve). We define a grid of parameter values to test, including:

- n_estimators: The number of trees in the model, testing values like 50, 100, 300.
- learning_rate: The learning rate of the model, testing values like 0.01 and 0.1.
- subsample: The subsample ratio, testing values like 0.5 and 0.8.
- colsample_bytree: The subsample ratio for each tree, testing values like 0.5 and 1.0.
- scale_pos_weight: The weight of the classes in the case of imbalanced data, testing values like 1, 5, and 10.

After selecting the model with the optimal parameter set, we evaluate the model on the test set. Evaluation metrics include classification reports, AUC, and Gini Index, which help measure the model's accuracy, class discrimination ability, and overall performance on the test data.

Through this process, we can select the optimal XGBoost model with the most suitable parameters, enhancing prediction performance for the classification problem.

Chapter 5

Results and Analysis

In this chapter, the results of the dataset trained and tested on selected algorithms, including XGB, NN, Ensemble, RFC, DTC, and LRC, will be explained. Performance metrics such as ROC-AUC, Gini Index, accuracy, precision, recall, and F1 score have been evaluated on the built models to identify the optimal model for predicting credit card approval. All the results have been derived from the test dataset. The results have been tabulated to evaluate the performance of the selected algorithms, and the tabulated results are presented below.

5.1 AUC

The results show that XGBoost (XGB) and the Ensemble models performed the best with an AUC score of 0.96, indicating strong discrimination between classes. The Random Forest Classifier (RFC) followed closely with a score of 0.95. Decision Tree Classifier (DTC) and Neural Network (NN) had scores of 0.91 and 0.90, respectively, showing reasonable performance but slightly lower than the ensemble models. Logistic Regression Classifier (LRC) had the lowest score at 0.89, indicating less effectiveness in handling the imbalanced dataset. Overall, XGB and Ensemble models were the most effective for the classification problem.

Algorithms	**AUC Score**
XGB	**0.96**
NN	0.90
Ensemble	**0.96**
RFC	0.95
DTC	0.91
LRC	0.89

Table 5.1: AUC Scores of Selected Algorithms

5.2 Gini Index

The results show that the XGBoost (XGB) and Ensemble methods perform the best, both achieving a Gini Index of 0.92, indicating strong predictive power. Random Forest Classifier (RFC) follows with a Gini Index of 0.90, also performing well. Decision Tree Classifier (DTC) has a Gini Index of 0.82, suggesting it struggles with complexity. Neural Network (NN) and Logistic Regression Classifier (LRC) show lower performance, with Gini Index values of 0.80 and 0.78, respectively.

Algorithms	**Gini Index**
XGB	**0.92**
NN	0.80
Ensemble	**0.92**
RFC	0.90
DTC	0.82
LRC	0.78

Table 5.2: Gini Index of Selected Algorithms

5.3 Accuracy

The results show that the XGBoost (XGB) model achieved the highest accuracy at 95%, indicating its strong performance in correctly classifying both classes. The Ensemble model followed closely with an accuracy of 88%, showcasing good performance as well. Random Forest Classifier (RFC) also performed well with an accuracy of 87%. The Decision Tree Classifier (DTC) and Logistic Regression Classifier (LRC) had accuracy scores of 85% and 83%, respectively, while the Neural Network (NN) performed the least, with an accuracy of 82%.

Algorithms	**Accuracy**
XGB	**0.95**
NN	0.82
Ensemble	0.88
RFC	0.87
DTC	0.85
LRC	0.83

Table 5.3: Accuracy of Selected Algorithms

5.4 F1-Score

The F1-Score combines both precision and recall, providing a balanced measure of a model's performance. XGBoost (XGB) and Random Forest Classifier (RFC) both achieved the highest F1-Score of 0.88, indicating that they were very effective in balancing precision and recall. The Neural Network (NN) followed with an F1-Score of 0.84, while the Ensemble and Decision Tree Classifier (DTC) both achieved scores of 0.84 and 0.85, respectively. Logistic Regression Classifier (LRC) had the lowest F1-Score at 0.83, suggesting it was less balanced in terms of precision and recall.

Algorithms	**Accuracy**
XGB	**0.95**
NN	0.82
Ensemble	0.88
RFC	**0.87**
DTC	0.85
LRC	0.83

Table 5.4: Accuracy of Selected Algorithms

5.5 Recall

Neural Network (NN) achieved the highest recall score of 0.96, which suggests it was the best at identifying positive cases (default). XGBoost (XGB) and Random Forest Classifier (RFC) both had recall scores of 0.93, indicating strong performance in detecting defaults. The Ensemble model followed closely with a recall score of 0.89, while the Decision Tree Classifier (DTC) and Logistic Regression Classifier (LRC) had recall scores of 0.87 and 0.86, respectively.

Algorithms	**Recall**
XGB	0.93
NN	**0.96**
Ensemble	0.89
RFC	0.93
DTC	0.87
LRC	0.86

Table 5.5: Recall of Selected Algorithms

5.6 Precision

Precision measures how many of the positively classified instances are actually relevant. Ensemble models achieved the highest precision of 0.88, followed by XGBoost (XGB) and Decision Tree Classifier (DTC), both with a precision of 0.83. Logistic Regression Classifier (LRC) had a precision of 0.81, while Neural Network (NN) had the lowest precision score of 0.75, indicating that NN misclassified a higher number of negative instances as positive compared to the other models.

Algorithms	**Precision**
XGB	0.83
NN	0.75
Ensemble	**0.88**
RFC	0.83
DTC	0.84
LRC	0.81

Table 5.6: Precision of Selected Algorithms

Chapter 6

Comparison

6.1 Comparison with Papers

Authors	**Selected Algorithms**	**Efficient Algorithms**	**Metrics (M)/AUC**
Mengran Zhu et al, 2024	GRU, Transformer, Tabtransformer, NN, XGB, LightGBM, CatBoost, Ensemble	Ensemble	0.80128
Kangshuai Guo et al, 2023	LightGBM, XGBoost, Local Ensemble	Ensemble	0.80872
Zhiren Gan et al, 2023	XGBoost, Lasso, Catboost, LightGBM	LightGBM	0.801
Yujin PAN et al, 2024	XGBoost, LightGBM, DNN, Hybrid Model	XGBoost	0.798
Saeid Bakhtiary et al, 2023	LightGBM, LiteMORT, Simple Average, Weighted Average	Weighted Average	AUC = 0.952
Zongqi Hu & Chai Kiat Yeo, 2024	GBDT Ensemble, DART Ensemble, LightGBM, Transformer based NN, Transformer Only, FE Only.	GBDT Ensemble	0.8013
Ali & Ayub, 2023	LM, XGB, DTC, NB, SVM, LG, ADA, SVM-GS, DenseNet, CapsNet, ResNet, RXT-J, BERT	RXT-J	0.987
Our Best Model	-	-	0.96

Table 6.1: Comparison of Algorithms and AUC from Various Studies

6.2 Comparison with Papers

Top	**Selected Algorithms**	**AUC Score**	**Gini Index**
Top 1	LightGBM	0.9617	0.92
Top 2	XGBoost	0.9611	0.92
Top 3	XGBoost	0.96099	0.92
Our Best Model	-	0.9621	0.92

Table 6.2: Comparison with submissions on Kaggle

Chapter 7

Discussion

Four ML algorithms are chosen for predicting if a credit card will be approved, and the most accurate ML algorithm is found using a Kaggle data set. These algorithms are compared based on evaluation metrics accuracy, precision, recall, F1 score, and ROC curve for predicting the approval of credit cards. Our project intended to help and have a significant impact on financial institutions and individuals by improving the accuracy of credit card approval prediction leading to more informed decisionmaking and ultimately reducing the risk of credit default. The RQ answers were described in this section.

7.1 Reflection on the Results of RQ

RQ: Which among the selected machine learning algorithms (XGBoost, Neural Networks, Random Forest Classifier, Decision Tree Classifier, and Logistic Regression Classifier) offers the highest accuracy in predicting credit card default events?

Motivation for RQ:

The primary motivation behind this research question (RQ) is the increasing need for effective and reliable machine-learning models in the field of credit card fraud detection and default prediction. Credit card companies face substantial financial losses due to default events, making it crucial to develop models that can predict defaults with high accuracy. By comparing various machine learning algorithms, such as XGBoost, Neural Networks (NN), Random Forest Classifier (RFC), Decision Tree Classifier (DTC), and Logistic Regression Classifier (LRC), this study aims to identify the most effective model in terms of accuracy, which is a key performance metric for assessing the model's ability to predict defaults correctly.

Furthermore, accurately predicting credit card defaults is not only beneficial to the financial sector but also to customers, as it helps in improving credit management, minimizing risks, and optimizing decision-making processes. Therefore, choosing the optimal algorithm to predict these events can contribute to enhanced risk assessment strategies, better financial planning, and the implementation of more accurate credit approval processes. This RQ was designed to identify which algorithm performs best under real-world conditions, using metrics like accuracy to ensure that the results can be trusted in practical applications.

The comparison of algorithms with the performance metrics is discussed below.

Comparison Based on AUC Score:

float

The comparison of selected ML algorithms based on AUC is shown in the figure. The AUC obtained by XGB was 96%, NN with 90%, Ensemble with 96%, RFC with 95%, DTC with 91%, and LRC with 89% on the testing dataset. From the figure, among the selected ML algorithms, XGB and Ensemble both achieved the highest AUC Score of 96% on the testing dataset. The second-best algorithm is RFC with an AUC Score of 95%.

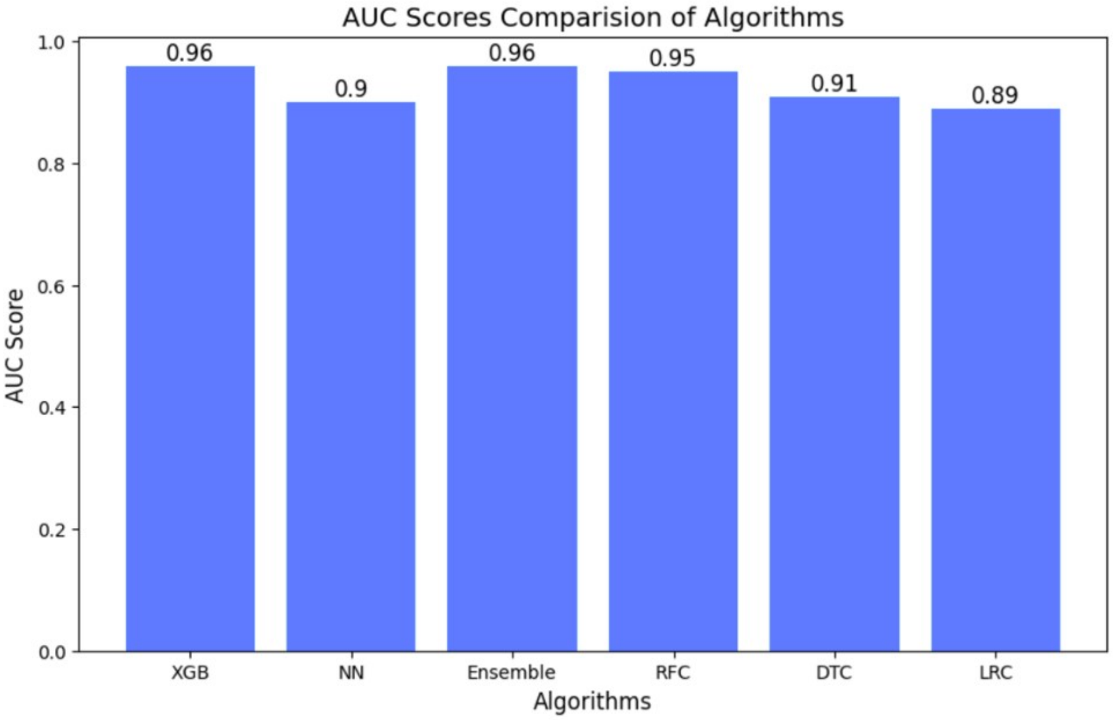


Figure 7.1: Comparison base on AUC

The comparison of selected ML algorithms based on the Gini Index is shown in the figure. The Gini Index obtained by XGB was 0.92, Ensemble with 0.92, RFC with 0.90, DTC with 0.82, NN with 0.80, and LRC with 0.78. From the figure, among the selected ML algorithms, XGB and Ensemble both achieved the highest Gini Index of 0.92. The second-best algorithm is RFC with a Gini Index of 0.90.

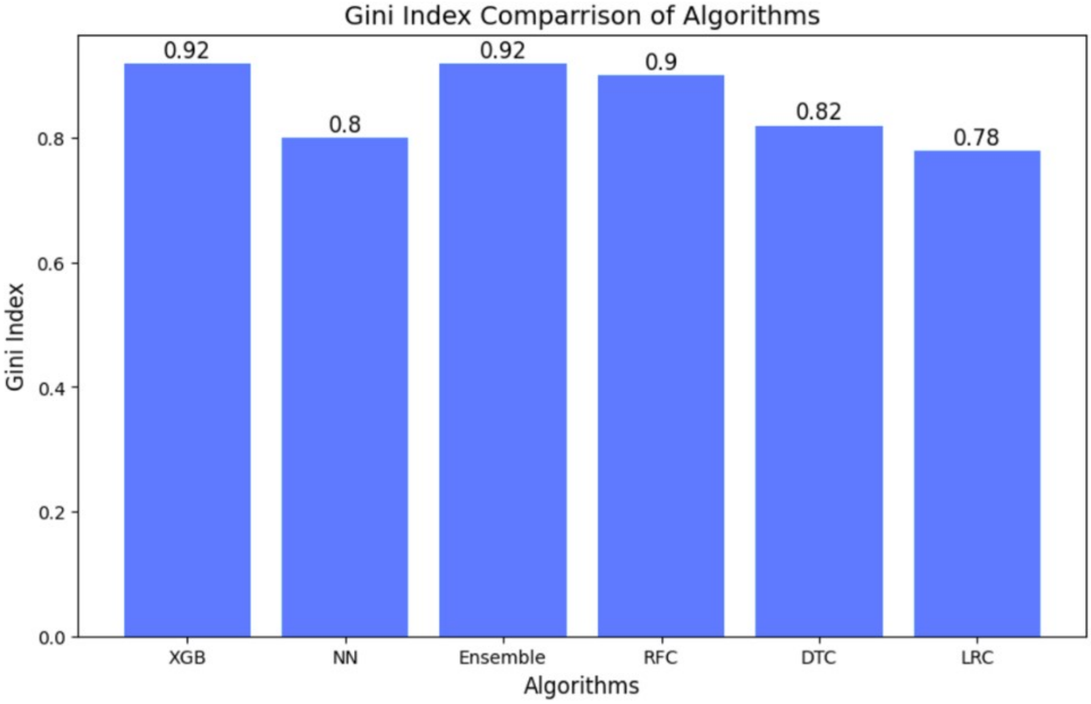


Figure 7.2: Comparison base on Gini Index.

The comparison of selected ML algorithms based on accuracy is shown in the figure. The accuracy obtained by XGB was 95%, NN with 82%, Ensemble with 88%, RFC with 87%, DTC with 85%, and LRC with 83% on the testing dataset. From the figure, among the selected ML algorithms, XGB achieved the highest accuracy of 95% on the testing dataset. The second-best algorithm is Ensemble with an accuracy of 88%.

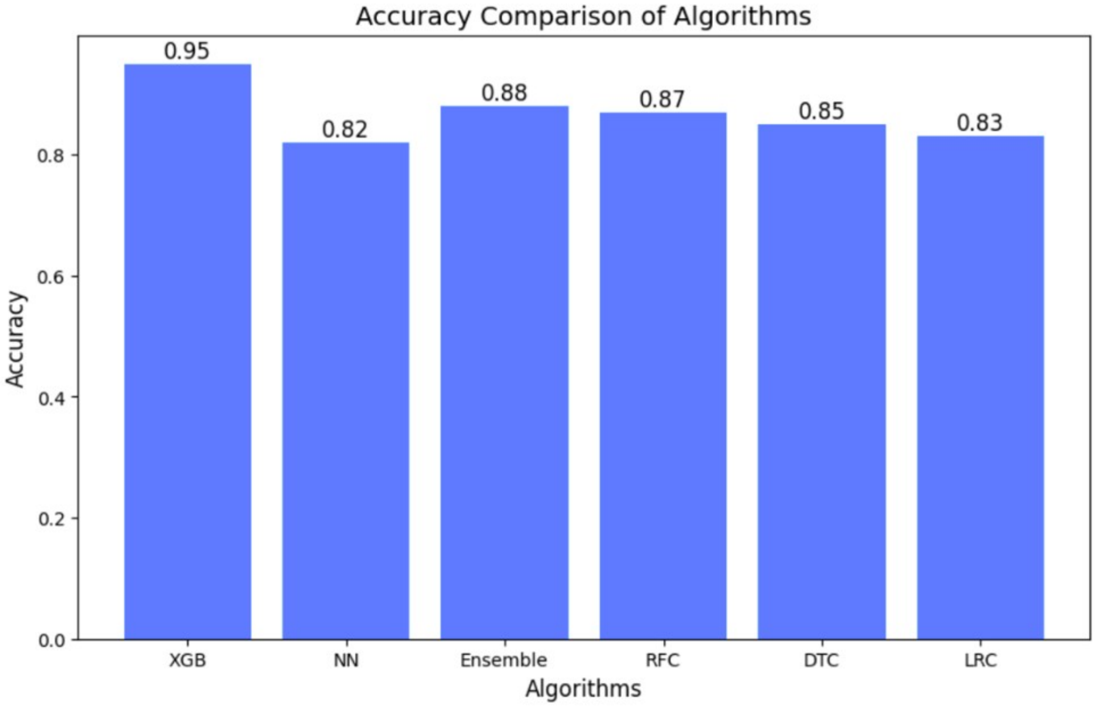


Figure 7.3: Comparison base on Accuracy.

The comparison of selected ML algorithms based on F1-Score is shown in the figure. The F1-Score obtained by XGB was 0.88, NN with 0.84, Ensemble with 0.84, RFC with 0.88, DTC with 0.85, and LRC with 0.83. From the figure, among the selected ML algorithms, XGB and RFC both achieved the highest F1-Score of 0.88. The second best algorithm is DTC with an F1-Score of 0.85, followed by NN and Ensemble, both with an F1-Score of 0.84. LRC obtained the lowest F1-Score of 0.83. These results show that XGB and RFC performed equally well in terms of balancing precision and recall, while other algorithms like NN and Ensemble also demonstrated strong performance but slightly lower F1 scores.

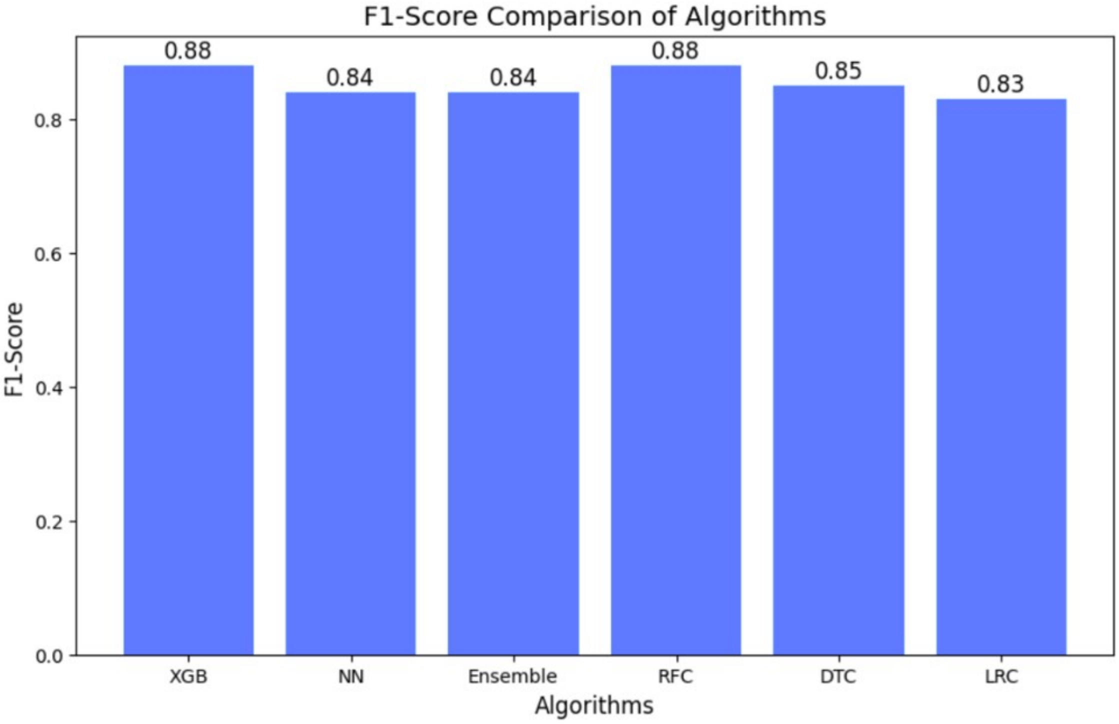


Figure 7.4: Comparison base on F1-Score.

The comparison of selected ML algorithms based on Recall is shown in the figure. The Recall obtained by XGB was 0.93, NN with 0.96, Ensemble with 0.89, RFC with 0.93, DTC with 0.87, and LRC with 0.86. From the figure, among the selected ML algorithms, NN achieved the highest Recall of 0.96 on the testing dataset, followed by XGB and RFC, both with a Recall of 0.93. The third best algorithm is Ensemble with a Recall of 0.89, while DTC and LRC showed lower recall values of 0.87 and 0.86, respectively. These results highlight that NN performed exceptionally well in identifying true positives, while XGB and RFC also performed strongly.

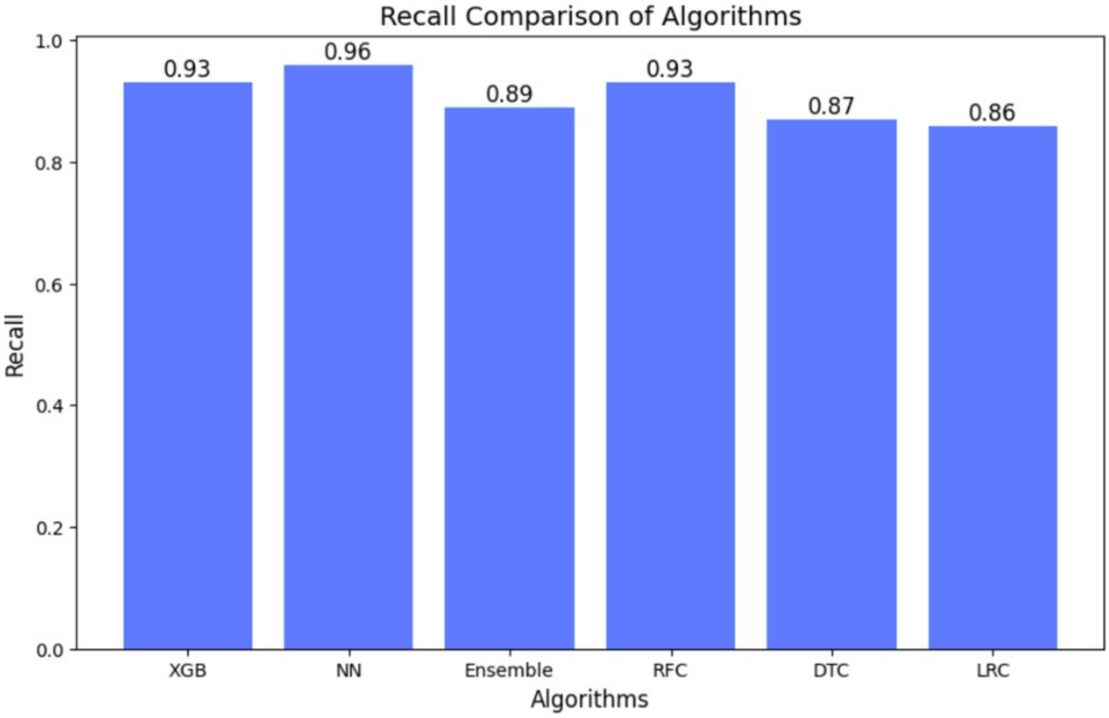


Figure 7.5: Comparison base on Recall.

The comparison of selected ML algorithms based on Precision is shown in the figure. The Precision obtained by XGB was 0.83, NN with 0.75, Ensemble with 0.88, RFC with 0.83, DTC with 0.84, and LRC with 0.81. From the figure, among the selected ML algorithms, Ensemble achieved the highest Precision of 0.88 on the testing dataset, followed by DTC with a Precision of 0.84. XGB and RFC both obtained a Precision of 0.83, while LRC achieved a Precision of 0.81. NN had the lowest Precision with 0.75. These results show that Ensemble performed best in terms of Precision, while NN struggled to balance precision, despite having the highest Recall.

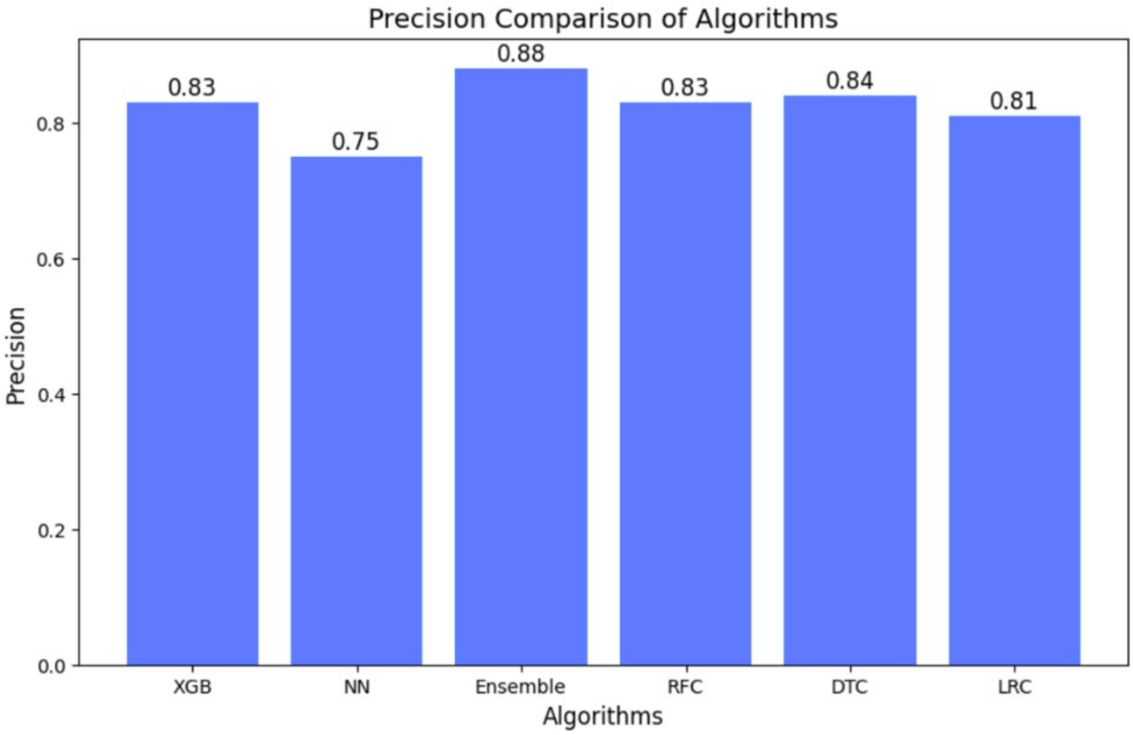


Figure 7.6: Comparison base on Precision.

7.2 Summary

Based on the evaluation metrics considered in this study, XGBoost (XGB) and Ensemble methods emerged as the top performers in predicting credit card default events. Both models achieved high AUC and Gini Index scores, indicating their strong discriminatory power in identifying defaults. Additionally, XGB achieved the highest accuracy of 95%, demonstrating its effectiveness in making correct predictions. Random Forest Classifier (RFC), although slightly lower in accuracy, performed similarly to XGB in terms of F1-Score and Recall, making it a strong contender for this task.

Neural Networks (NN) exhibited the highest Recall, making it particularly effective at detecting actual default cases. However, its lower precision and accuracy suggest that it may not be as reliable in practice, as it misclassified a significant number of non-default cases as defaults. Decision Tree Classifier (DTC) and Logistic Regression Classifier (LRC) demonstrated relatively lower performance across all metrics, with DTC showing slightly better balance in terms of F1-Score and Recall.

While XGB and RFC proved to be the most balanced and effective models overall, Ensemble techniques excelled in Precision, which is valuable when minimizing false positives—important in situations where the cost of incorrectly predicting a default is high (e.g., denying credit). However, despite performing well, Ensemble models were not as effective as XGB and RFC in terms of accuracy and F1-Score.

In conclusion, XGBoost and Random Forest Classifier are the optimal models for predicting credit card default in this study, as they provide the best balance between accuracy, Recall, and F1-Score. These models, particularly XGB, are recommended for deployment in real-world systems to predict defaults, as they perform well across all key evaluation metrics. Future work could focus on improving these models or integrating them with other techniques to further enhance prediction accuracy, particularly in scenarios where minimizing the risk of missed defaults (false negatives) is crucial.

Chapter 8

Conclusions and Future Work

8.1 Conclusion

This research has demonstrated the significant potential of machine learning algorithms in enhancing the accuracy and efficiency of credit card default prediction. By leveraging advanced models such as XGBoost (XGB), Neural Networks (NN), Ensemble methods, Random Forest Classifiers (RFC), Decision Trees (DTC), and Logistic Regression (LRC), we have provided valuable insights into the performance capabilities of these techniques in predicting credit card defaults.

The results of the evaluation, based on key performance metrics such as AUC, Gini Index, accuracy, precision, recall, and F1-score, revealed that XGBoost and Ensemble models consistently provided superior performance. XGBoost achieved the highest AUC, accuracy, and F1-score, indicating its strong ability to discriminate between classes and accurately classify credit card defaults. Random Forest Classifier and Decision Tree also performed well, showcasing their robustness in handling complex datasets. On the other hand, Neural Networks, despite being slightly less accurate in some aspects, demonstrated the highest recall, which is crucial for identifying potential defaults.

The findings of this research emphasize the clear advantage of machine learning over traditional, rule-based credit scoring methods. Machine learning models, with their ability to analyze vast amounts of data and uncover hidden patterns, allow financial institutions to better predict credit defaults and make more informed decisions. This approach is particularly valuable in an era where credit data is growing exponentially, and traditional methods may not be sufficient to keep up with the complexity of modern financial transactions.

Furthermore, the study highlights the potential for machine learning to improve not only the financial outcomes for institutions but also the customer experience. By offering more accurate and fairer assessments, financial institutions can ensure that their credit decisions are data-driven, reducing the chances of misclassifying borrowers. This not only minimizes financial risk but also promotes greater financial inclusivity, offering opportunities for customers who may have been overlooked by traditional scoring models.

In conclusion, this research underscores the transformative role that machine learning can play in the field of credit risk management. By embracing these advanced models, financial institutions can enhance their ability to predict defaults, optimize credit allocations, and ultimately drive more efficient, customer-centric operations. The results demonstrate the immense potential of machine learning to revolutionize credit card default prediction and improve the stability and growth of financial systems, ensuring that they remain competitive in an increasingly complex and data-driven world.

8.2 Future Work

- **Explore Other Ensemble Methods:** Future research can investigate other ensemble techniques like Boosting (e.g., AdaBoost, Gradient Boosting) and Stacking to potentially enhance the prediction accuracy.
- **Better Handling of Missing Data:** More advanced strategies for handling missing data, such as K-Nearest Neighbors (KNN) imputation or Multiple Imputation, could be explored.
- **Feature Creation and Selection:** Apply Principal Component Analysis (PCA) for dimensionality reduction or use feature importance methods like Random Forest or XGBoost.
- **Hyperparameter Optimization:** Employing more advanced hyperparameter tuning techniques, such as Bayesian Optimization or RandomizedSearchCV, could further improve model performance beyond basic GridSearchCV.
- **Use Larger Datasets:** Using a larger dataset could help increase the robustness and accuracy of the model, especially for real-world applications.

References

[1] Almazroi, A. A., Ayub, N. (2023). Online Payment Fraud Detection Model Using Machine Learning Techniques. IEEE Access, 11, 137188-137203.

[2] Bakhtiari, S., Nasiri, Z., Vahidi, J. (2023). Credit card fraud detection using ensemble data mining methods. Multimedia Tools and Applications, 82(19), 29057-29075.

[3] Bodepudi, H. (2021). Credit card fraud detection using unsupervised machine learning algorithms. Int J Comput Trends Technol, 69, 1-13.

[4] Chaovalit, P., Zhou, L. (2005, January). Movie review mining: A comparison between supervised and unsupervised classification approaches. In Proceedings of the 38th annual Hawaii international conference on system sciences (pp. 112c-112c). IEEE.

[5] Chen, T., Guestrin, C. (2016, August). Xgboost: A scalable tree boosting system. In Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining (pp. 785-794).

[6] Chicco, D., Jurman, G. (2020). The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. BMC genomics, 21, 1-13.

[7] Cunningham, P., Cord, M., Delany, S. J. (2008). Supervised learning. In Machine learning techniques for multimedia: case studies on organization and retrieval (pp. 21-49). Berlin, Heidelberg: Springer Berlin Heidelberg.

[8] D. P. Kingma and J. Ba, Adam: A method for stochastic optimization, 2017. arXiv: 1412.6980 [cs.LG].

[9] Flores, L., Hernandez, R. M., Tolentino, L. C., Mendez, C. A., Fernando, M. G. Z. (2022, August). A Classification Approach in the Probability of Credit Card Approval using Relief-Based Feature Selection. In 2022 2nd Asian Conference on Innovation in Technology (ASIANCON) (pp. 1-7). IEEE.

[10] Gan, Z., Qiu, J., Li, F., Liang, Q. (2023, August). A LightGBM Based Default Prediction Method for American Express. In Proceedings of the 2nd International Conference on Information Economy, Data Modeling and Cloud Computing, ICIDC 2023, June 2–4, 2023, Nanchang, China.

[11] G. LI, (2018), Gender-related differences in credit use and credit scores, The Fed - Gender-Related Differences in Credit Use and Credit Scores.

[12] Guo, K., Luo, S., Liang, M., Zhang, Z., Yang, H., Wang, Y., Zhou, Y. (2023, June). Credit Default Prediction on Time-Series Behavioral Data Using Ensemble

Models. In 2023 International Joint Conference on Neural Networks (IJCNN) (pp. 01-09). IEEE.

[13] H. H. van Engelen, J.E., in A survey on semi-supervised learning, 2022, pp. 373–440.

[14] HASTIE, T., FRIEDMAN, J.H. TIBSHIRANI, R. 2009 The Elements of Statistical Learning: Data Mining, Inference, and Prediction, vol. 2. Springer.

[15] Hossin, M., Sulaiman, M. N. (2015). A review on evaluation metrics for data classification evaluations. International journal of data mining knowledge management process, 5(2), 1.

[16] Hu, Z., Yeo, C. K. (2024, June). A Lightweight Neural Network with Transformer to Predict Credit Default. In 2024 IEEE Conference on Artificial Intelligence (CAI) (pp. 29-30). IEEE.

[17] J. E. Van Engelen and H. H. Hoos, "A survey on semi-supervised learning," Machine learning, vol. 109, no. 2, pp. 373–440, 2020, publisher: Springer.

[18] Kadhim, A. I. (2019). Survey on supervised machine learning techniques for automatic text classification. Artificial intelligence review, 52(1), 273-292.

[19] Kynkäänniemi, T., Karras, T., Laine, S., Lehtinen, J., Aila, T. (2019). Improved precision and recall metric for assessing generative models. Advances in neural information processing systems, 32.

[20] L. F. DUNN, I. A. MIRZAIE, (2022), Gender Differences in Consumer Debt Stress: Impacts on Job Performance, Family Life and Health, J Fam Econ Iss. https://doi.org/10.1007/s10834-022-09862-z

[21] Lessmann, S., Baesens, B., Seow, H. V., Thomas, L. C. (2015). Benchmarking state-of-the-art classification algorithms for credit scoring: An update of research. European Journal of Operational Research, 247(1), 124-136.

[22] Liu, B., Liu, B. (2011). Supervised learning. Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data, 63-132.

[23] M. A. ULLAH, M. M. ALAM, S. SULTANA, R. S. TOMA, (2018), Predicting Default Payment of Credit Card Users: Applying Data Mining Techniques, 2018 International Conference on Innovations in Science, Engineering and Technology (ICISET), 355-360.

[24] M. DOMINGUEZ, (2021), Predicting credit card defaults with machine learning, Medium: The Startup.

[25] M. Kibria and M. Şevkli, "Application of Deep Learning for Credit Card Ap proval: A Comparison with Two Machine Learning Techniques," vol. 11, pp. 286–290, Jun. 2021.

[26] M. P. C. Peiris, "Credit Card Approval Prediction by Using Machine Learning Techniques."

[27] McKinney, W. (2012). Python for data analysis: Data wrangling with Pandas, NumPy, and IPython. " O'Reilly Media, Inc.".

[28] Narkhede, S. (2018). Understanding auc-roc curve. Towards data science, 26(1), 220-227.

[29] Pan, Y., Zhang, F., Xu, T., Cai, Y. (2024). A Novel Default Prediction Model Based on DNN and LightGBM. In Artificial Intelligence Technologies and Applications (pp. 316-322). IOS Press.

[30] Q. Bi, K. E. Goodman, J. Kaminsky, and J. Lessler, “What is Machine Learning? A Primer for the Epidemiologist,” American Journal of Epidemiology, vol. 188, no. 12, pp. 2222–2239, Dec. 2019. [Online]. Available: https://doi.org/10.1093/aje/kwz189

[31] Rajula, H. S. R., Verlato, G., Manchia, M., Antonucci, N., Fanos, V. (2020). Comparison of conventional statistical methods with machine learning in medicine: diagnosis, drug development, and treatment. Medicina, 56(9), 455.

[32] R. S. Sutton and A. G. Barto, Reinforcement Learning, second edition: An Introduction. MIT Press, Nov. 2018, google-Books-ID: uWV0DwAAQBAJ.

[33] Sibyan, H., Suharso, W., Suharto, E., Manuhutu, M. A., Windarto, A. P. (2021, February). Optimization of Unsupervised Learning in Machine Learning. In Journal of Physics: Conference Series (Vol. 1783, No. 1, p. 012034). IOP Publishing.

[34] S. L. STOLBA, (2020), Married consumers have higher credit scores and debt than single adults, In Experian.

[35] S. V. JAIN, M. JAYABALAN, (2022), Applying Machine Learning Methods for Credit Card Payment Default Prediction With Cost Savings, Biomedical and Busi- ness Applications Using Artificial Neural Networks and Machine Learning: 285- 305.

[36] Sugiyarto, I., Sudarsono, B., Faddillah, U. (2019). Performance comparison of data mining algorithm to predict approval of credit card. Sinkron: jurnal dan penelitian teknik informatika, 4(1), 149-157.

[37] Thomas, L., Crook, J., Edelman, D. (2017). Credit scoring and its applications. Society for industrial and Applied Mathematics.

[38] Tosi, S. (2009). Matplotlib for Python developers. Packt Publishing Ltd.

[39] V. Nasteski, “An overview of the supervised machine learning methods,” Horizons. b, vol. 4, pp. 51–62, 2017.

[40] West, D. (2000). Neural network credit scoring models. Computers operations research, 27(11-12), 1131-1152.

[41] Z.-H. Zhou, Machine learning. Springer Nature, 2021.

[42] Zhao, Y. (2022, February). Credit card approval predictions using logistic regression, linear SVM and Naïve Bayes classifier. In 2022 International Conference on Machine Learning and Knowledge Engineering (MLKE) (pp. 207-211). IEEE.

[43] Zhu, M., Zhang, Y., Gong, Y., Xing, K., Yan, X., Song, J. (2024). Ensemble methodology: Innovations in credit default prediction using lightgbm, xgboost, and localensemble. arXiv preprint arXiv:2402.17979.

[44] Zou, M., Jiang, W. G., Qin, Q. H., Liu, Y. C., Li, M. L. (2022). Optimized XGBoost model with small dataset for predicting relative density of Ti-6Al-4V parts manufactured by selective laser melting. Materials, 15(15), 5298.

[45] S. Narkhede, “Understanding AUC- ROC Curve.”