Predicting Credit Card Defaults: A Comparative Analysis of Machine Learning Models

Marc Leon Jackwitz

Dissertation written under the supervision of Francesco Rotondi

Dissertation submitted in partial fulfilment of requirements for the MSc in International Finance, at Universidade Católica Portuguesa and for the MSc in Accounting, Financial Management and Control at Bocconi University, 07.09.2024.

Predicting Credit Card Defaults: A Comparative Analysis of Machine Learning Models

Marc Leon Jackwitz

Abstract

In a time of rising interest rates and economic difficulties, the accurate prediction of credit card defaults is essential for financial institutions. In this master thesis, different machine learning models are evaluated to improve the predictive performance compared to traditional methods. Models such as XGBoost, Random Forest, K-Nearest Neighbor, Neural Networks and Support Vector Machines are developed and compared to the Logistic Regression. Performances are evaluated based on accuracy, precision, recall and F1-score and compared to their computation time as a cost factor. The learning curves are then analyzed to identify possible over- or underfitting. Finally, the importance of each feature is reviewed to gain an understanding of which data banks should include in their analysis. The results show that all advanced machine learning models outperform the Logistic Regression, with XGBoost coming out on top. However, KNN lags behind the other models. Key features identified include payment status and closeness to credit limit. Therefore, this study underlines the potential of advanced machine learning models to improve the prediction of credit card defaults, but also highlights the need for institution specific analysis due to data variability and computational complexity.

Key words: Credit Card Defaults; Machine Learning; Predictive Modeling; Financial Risk Analysis

Previsão de incumprimentos de cartões de crédito: Uma análise comparativa dos modelos de aprendizagem automática

Marc Leon Jackwitz

Resumo

Em um período de aumento das taxas de juros e dificuldades econômicas, a previsão precisa de inadimplências de cartões de crédito é essencial para as instituições financeiras. Nesta tese de mestrado, diferentes modelos de machine learning são avaliados para melhorar o desempenho preditivo em comparação com métodos tradicionais. Modelos como XGBoost, Random Forest, K-Nearest Neighbor, Neural Networks e Support Vector Machines são desenvolvidos e comparados à Regressão Logística. Os desempenhos são avaliados com base em acurácia, precisão, recall e F1-score, e comparados com o tempo de computação como um fator de custo. As curvas de aprendizado são então analisadas para identificar possíveis overfitting ou underfitting. Finalmente, a importância de cada feature é revisada para entender quais dados os bancos devem incluir em suas análises. Os resultados mostram que todos os modelos avançados de machine learning superam a Regressão Logística, com o XGBoost se destacando. No entanto, o KNN fica atrás dos outros modelos. As principais features identificadas incluem status de pagamento e proximidade do limite de crédito. Portanto, este estudo sublinha o potencial dos modelos avançados de machine learning para melhorar a previsão de inadimplências de cartões de crédito, mas também destaca a necessidade de análises específicas para cada instituição devido à variabilidade dos dados e à complexidade computacional.

Palavras-chave: Incumprimentos de Cartões de Crédito; Machine learning; Modelação Preditiva; Análise de Risco Financeiro

Content table

Acknowledgements

This thesis marks the final step towards the completion of my Double Degree Master at the Católica Lisbon School of Business & Economics and the Bocconi University. Therefore, I would like to express my deepest gratitude to everyone who has supported me during the development of this thesis.

I would like to sincerely thank my supervisor, Francesco Rotondi, for his valuable guidance, patience and constructive advice. His expertise has contributed significantly to the success of this thesis.

Furthermore, I would like to thank my girlfriend, family and friends for their unconditional support during all years of my studies. Without the most important people in my life, none of this would have been manageable for me. Therefore, all of you deserve my sincere thanks.

List of tables

List of figures

List of abbreviations

Extreme Gradient Boost	XGBoost
Support Vector Machine	SVM
K-Nearest Neighbor	KNN
New Taiwan Dollar	TWD
United States Dollar	USD
Synthetic Minority Oversampling Technique	SMOTE
Adaptive Moment Estimation	Adam
Radial Basis Function	RBF

1. Introduction

In a world where uncertainty is ever-present, the accurate prediction of credit card defaults plays a crucial role for financial institutions. The ability to accurately predict defaults can significantly impact the success or failure of credit card institutions (Ray, 2012). Recently, machine learning has emerged as a powerful tool in improving these predictions (Barboza et al., 2017). Machine learning, which intersects with both cybernetics and computer science, has garnered overwhelming interest from professionals and the public alike. Advances in computer science have significantly improved computational performance, facilitating the handling of big data (Fradkov, 2020). This growing importance is also evident from the increasing frequency with which CEOs mention artificial intelligence (AI) in their earnings calls, a trend that has been rising since 2015 (Manjaly et al., 2021).

The first modern credit card was introduced by American Express in 1958, and the number of cards in use has been growing ever since (Johnson, 2024). Traditionally, basic statistical models have been used to evaluate the creditworthiness of a client and predict the probability of default. Nevertheless, the limitations of these models in regards of precision and adaptability make them less suitable for credit card institutions (Greene, 1992). This is where machine learning models demonstrate their advantages, as they can process large datasets and recognize complex patterns more effectively (Zhou et al., 2017).

Despite the potential benefits, implementing these technologies in credit risk management still presents several challenges. These include feature selection, dataset imbalance, and the varying characteristics of each machine learning model (Noriega et al., 2023).

To address these issues, this master thesis evaluates different machine learning models and compares them to a benchmark to analyze their predictive power for credit card defaults. The research questions are:

1. How do advanced machine learning models perform compared to a traditional statistical model in predicting credit card defaults when evaluated across various performance metrics?
2. What are the key features, based on the dataset used, that significantly influence the predictive performance of advanced machine learning models in credit risk assessment?

To answer these questions, an analysis of existing research is conducted to select the models for this study. After acquiring a dataset with sufficient instances of credit card clients, various machine learning models are developed, including Extreme Gradient Boost (XGBoost), Random Forest, K-Nearest Neighbor (KNN), Neural Networks, and Support Vector Machines (SVM). These models are selected to represent the most prominent areas of machine learning for classification problems and are frequently utilized in recent research due to their strong performance (Maxwell et al., 2018).

As benchmark, this research uses Logistic Regression with and without penalty terms. Logistic Regression is a suitable benchmark model because of its simplicity, interpretability, and widespread use in binary classification problems. It provides a clear baseline to compare the performance of more complex machine learning models (Landwehr et al., 2003). The performance of the advanced machine learning models is evaluated against the benchmark using various metrics, such as accuracy, precision, recall, F1-score, and computational time as a cost measure. Additionally, learning curves are analyzed to evaluate model quality and detect overfitting or underfitting. Feature importance is introduced to understand the relevance of each feature.

This thesis contributes to the research field of machine learning by applying it to the crucial area of credit card risk. Furthermore, it provides valuable insights for credit card institutions and banks on improving and automating their risk management processes. Notably, this research uniquely considers a cost factor and examines feature importance, which has not been done with this dataset before.

The master thesis is structured as follows: Chapter 2 provides an overview of the theoretical basis and current research in the field of machine learning in finance and credit card default prediction. Chapter 3 details the dataset used in this analysis. Chapter 4 explains the research methodology and the functionality of each model. Chapter 5 presents the research results and compares them with existing research. Finally, chapter 6 answers the research questions and discusses limitations and future research directions.

2. Literature review

2.1 Basic knowledge about machine learning

2.1.1 Definition of machine learning

One of the first definitions of machine learning comes from Samuel (1959), in which he says that machine learning is the field that gives computers the ability to learn without being explicitly programmed.

A newer definition is set by Mitchell (1997) and defines machine learning as "a computer program said to learn from experience E with the respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E" (p. 2).

Similar to other current research (Muggleton et al., 2018), this thesis is based on the machine learning definition of Mitchell (1997).

2.1.2 Application fields

Machine learning has become essential in several areas. In healthcare, for example, deep learning techniques can improve disease diagnosis, give personalized treatment recommendations and do medical image analysis (Lenert et al., 2018).

In addition, machine learning plays an important role in our daily lives, which is particularly evident in activities such as listening to music or streaming videos on the internet. Recommendation engines like Facebooks' ResNeXt WSL models excel in finding underlying characteristics to give recommendations on human's tastes and preferences (Orhan, 2019).

Moreover, machine learning can be used to perform sentiment analysis, which is also known as opinion mining. This involves extracting and analyzing people's opinions, sentiments, attitudes, and perceptions toward different entities, such as topics, products, and services (Birjali et al., 2021).

In addition, language models, such as ChatGPT, have the primary function of generating human-like text by taking advantage of the patterns and structures present in the large amount of text data on which they have been trained (Radford et al., 2019).

2.1.3 Most important models

Machine learning can be divided into different types, each of which is suitable for different problems and data. In supervised learning, a model is trained on a labelled dataset where the input data are matched with the correct output and is used for tasks such as classification and regression, for example, for spam detection in emails or predicting property prices. In contrast, unsupervised learning trains a model on data without labels and focuses on discovering hidden patterns or intrinsic structures. It is used for clustering, anomaly detection, and association tasks such as client segmentation. Semi-supervised learning uses both labeled and unlabeled data for training and is useful in situations where labeling data are expensive or time-consuming, such as image and speech recognition. Reinforcement learning trains models to make a series of decisions by rewarding desirable and punishing undesirable behaviors. Self-supervised learning is a form of unsupervised learning which the data itself does the monitoring. It is typically used in scenarios, like language modeling, and has applications in natural language processing, such as language translation and sentiment analysis (Libbrecht & Noble, 2015).

Machine learning includes various models developed to solve different problems. Linear regression, a supervised learning algorithm, predicts continuous outcomes by fitting a linear equation to the data, for example to determine house prices (Kumari & Yadav, 2018). In contract, Logistic Regression, another supervised technique, is used for binary classification tasks. Both regression have regularization methods such as Lasso, Ridge, and Elastic Net to improve their performance and prevent overfitting. Regressions are typically used in applications such as spam detection and medical diagnosis (Berkson, 1944; Tibshirani, 1996; Hoerl & Kennard, 1970; Zou & Hastie, 2005).

Decision tree based models, which are also supervised, create a tree-like structure of decisions to solve a predefined tasks, such as client segmentation and risk assessment (Quinlan, 1986). SVMs are robust supervised models used for classification and regression by finding the best separating hyperplane that is effective in high-dimensional spaces for applications such as image and text classification (Mammone et al., 2008). Neural Networks inspired by the human brain are powerful supervised models that can capture complex patterns, and are widely used in image and speech recognition and natural language processing (Goldberg, 2015; Brush, 1967).

KNN is a simple supervised algorithm that classifies samples based on their nearest neighbors and is used for pattern recognition (Fix & Hodges, 1951). K-Means Clustering, an unsupervised algorithm, divides data into clusters based on feature similarity, which is useful for market segmentation and exploratory data analysis (Likas et al., 2003).

2.2 Credit card risk measurement before machine learning

A crucial aspect for banks is the effective implementation of credit card default prediction models, which can lead to significant cost savings. Even before the advent of advanced machine learning techniques, assessing default risk was important for banks given the long history of credit cards as a payment method (Johnson, 2024). However, in the absence of modern computing capabilities, empirical research has been significantly limited in its ability to build prediction models.

Initially, conceptual frameworks such as asymmetric information and adverse selection were employed to enhance the comprehension of defaults (Jaffee & Russell, 1976). The surge in credit card defaults during the early 1990s increased rapidly, leading to an interest in their underlying causes by identifying correlations within the data (Ausubel, 1997).

Subsequently, a more targeted analysis focused on client-specific data and utilized regression models to identify variables that could explain the default patterns. Consequently, the focus is on independent variables that can capture the most important behaviors of credit card holders to facilitate default prediction (Dunn & Kim, 1999).

2.3 Machine learning applications in finance

In contemporary finance, machine learning has applications across various fields. Aziz et al. (2024) categorizes these applications into three primary areas: (1) investment analysis topics, (2) asset modeling and forecasting topics, and (3) risk management topics.

2.3.1 Investment analysis topics

The first application is to understand market sentiment by extracting information from a large sample of news and images and translating them into investor sentiment. These insights can be used to predict market return reversals (Obaid & Pukthuanthong, 2022). Despite

general market sentiment, machine learning models demonstrate remarkable accuracy in predicting financial crises (Samitas et al., 2020).

Machine learning extends its benefits beyond individual securities for fund selections. By exploiting fund characteristics, machine learning models can effectively differentiate high- from low-performing mutual funds, both before and after fees, while also identifying skilled managers (DeMiguel et al. 2023; Kaniel et al. 2023).

2.3.2 Asset modeling and forecasting topics

Nonlinear machine learning models, such as XGBoost with and without dropout, Random Forests, and Feed-Forward Neural Networks, represent a promising path in option return prediction. These models can generate statistically and economically sizable profits in long-short portfolios of equity options, even after accounting for transaction costs (Bali et al. 2023; Benth et al., 2024).

The significant contribution of machine learning techniques, specifically extreme trees and Neural Networks, can also be highlighted by the predictability of bond returns. The findings indicate that Neural Network forecasts that leverage macroeconomic and yield data outperform forecasts based solely on yields, resulting in greater economic gains (Bianchi et al., 2020).

Similar success can be achieved in terms of measuring equity risk premiums, with trees and Neural Networks playing crucial roles (Gu et al., 2020; Routledge, 2019). This predictive power extends beyond the US market, and a high predictability of large stocks and state-owned enterprises could be achieved in the Chinese market using Neural Networks (Leippold et al., 2022).

In addition to return forecasting, machine learning models are used to predict realized and implied volatility. Nevertheless, the combination of traditional and machine learning models yields better results than relying solely on new machine learning techniques (Gunnarsson et al., 2024). These models can also be used in the cryptocurrency market, outperforming the GARCH model (Y. Wang et al., 2023).

Less favorable results are obtained when using linear earnings models to forecast firms' earnings, defined as the variance between analysts' expectations and an unbiased machine learning benchmark. Analysts have, on average, an upward bias, but the commonly used linear earnings models do not work out-of-sample and are still inferior to analysts'

predictions (van Binsbergen et al., 2022). Consequently, the out-of-sample forecast does not generate a more accurate forecast than analysts' consensus forecasts (Cao & You, 2024).

Despite the prediction of stocks, machine learning techniques, particularly XGBoost, can effectively forecast inflation, surpassing the performance of traditional forecasting models (Mirza et al., 2024).

2.3.3 Risk management topics

In risk management, classification models are commonly used to predict default. One such example is the Logistic Regression model. Combined with mixed-data sampling, it significantly improved the prediction of US bank failures and even included some correctly classified bank failures that have been misclassified without mixed-data sampling (Audrino et al., 2019).

Similarly, in the UK, a broader approach to predicting firm failures has concluded that Random Forest models exhibit superior accuracy compared to other models (Sermpinis et al., 2023).

Regarding fraud detection, remarkable improvements over alternative methods could have been made by using dynamic time warping models in order to detect illegal trading behavior in high-frequency trading data (R. James et al., 2023). Additionally, dynamic ensemble selection models outperform several other machine learning models, and should be considered for other applications in finance (Achakzai & Peng, 2023).

Another approach to using KNN, SVM, XGBoost, and Neural Networks is to predict systematic risk for the stock market. These models seem especially effective in covering the interaction between stock market volatility and other drivers (G.-J. Wang et al., 2024). In addition, when focusing on geopolitical risks, the prediction of machine-learning models is superior to other forecasting methods (Z. Niu et al., 2023).

2.4 Machine learning for credit card risk

2.4.1 Research using different datasets

Machine learning models play a crucial role in credit card operations and can be used to predict default risk and model profits for card underwriters.

In the area of online peer-to-peer lending in China, the combination of Random Forest with hyperparameter optimization proved superior to logistic models in predicting default probabilities (Y. Liu et al., 2022).

The economic impact of using machine learning models such as XGBoost, instead of traditional Logistic Regression models, is substantial. This change results in significant savings in regulatory capital for banks, from 12.4% to 17%, underlining the importance of adopting advanced machine learning models (Alonso-Robisco & Carbó, 2022).

While various studies compare different datasets and models, some researchers come to the conclusion that boosted models bring the best results (Bačová & Babič, 2021) whereas others find Neural Networks to bring the best accuracy (Chishti & Awan, 2019). Conversely, KNN seems to lack, in most cases, the predictive power of Neural Networks (Chen & Zhang, 2021).

For a dataset from Taiwan, Random Forest, SVM, and Logistic Regression algorithms were tested, with SVM showing a slightly better accuracy (Arora et al., 2022). Similar findings are observed across various datasets and variables (Lawi and Aziz, 2018; Moula et al., 2017).

Nevertheless, other studies have confirmed that Random Forest is a superior model for predicting credit defaults (Subasi & Cankurt, 2019; Sayjadah et al., 2018). In some cases, they are superior to classic FICO models and decision trees (Yu, 2020).

2.4.2 Research using the same dataset

In recent years, various studies used different machine learning models to analyze the same dataset, providing valuable insights into their comparative performance. Therefore, this section summarizes the key findings of previous research.

Faraj et al., (2021) conduct a comprehensive comparison of machine learning models for both imbalanced and balanced datasets. Their study shows the superiority of Neural Networks over linear models, with XGBoost proving to be the best performing model. They

use the F1-score as a performance measure and find that balancing the dataset significantly improves the performance of all models.

Soui et al. (2018) study various models including Genetic Programming, Logistic Regression, KNN, Random Tree and Naïve Bayes to predict defaults. They conclude that Genetic Programming achieves the best results, using precision, recall and accuracy as evaluation metrics.

Islam et al., (2018) combine the Taiwan dataset with a similar dataset from Spain. Their findings reveal that Extremely Random Trees outperforms all other models, with Random Forests coming in a close second. They note that other models lag significantly behind these two top performers, using the F1-score and other metrics for evaluation.

Singh and Sivasankar (2019) utilize KNN, Naïve Bayes, Decision Tree, and SVM as base models. They apply bagging, boosting and Random Forest techniques for prediction and find that boosting achieves the highest accuracy.

Teng and Lee (2019) test models such as KNN, Decision Trees, Boosting, SVM and Neural Networks. Their study highlights that decision trees perform best among the evaluated models.

R. Liu (2018) evaluate SVM, KNN, Decision Trees, Random Forest, Neural Network, and long short-term memory models. He addresses the issue with dataset imbalance in order to improve model performance and reports that the Neural Network performs best.

Mei (2016) compares the Lasso model with the Random Forest model and comes to very similar results, with Random Forest performing slightly better.

Neema and Soibam (2017) examine models such as Neural Network, Random Forest, Decision Tree, Naïve Bayes, KNN, Ridge Regression and Logistic Regression. Their results show that Random Forest is the best performing model. They handle the imbalance of the dataset with SMOTE, which further improves the performance of the model.

Finally, Yeh and Lien (2009) compare different models and conclude that Neural Network is the best performing model. Their study includes KNN, Logistic Regression, Discriminant Analysis, Naïve Bayes, Neural Networks and Classification Trees.

3. Data

3.1 Origin of the data

The dataset used in this study is from the UCI Machine Learning Repository and includes data from 30,000 Taiwanese residents from October 2005. It contains detailed information on their credit card transactions over the previous six months, starting from May 2005. This period coincides with a crucial period for credit card issuers in Taiwan, characterized by problems in managing cash flow and credit card debt (Chou, 2006). There are two main factors contributed to the occurrence of this crisis. First, the financial supervisory authority failed to effectively supervise the banks' credit reporting system. Secondly, the situation was worsened by the indiscriminate issuing of cards by banks combined with the lack of strict credit checks (Chang, 2022).

3.2 Structure of the dataset

In this study, the dataset comprises 30,000 instances, each characterized by 23 features. Apart from the binary character of the label, all data types in the dataset are integers. However, it should be noted that some features have categorical characteristics, although this is not explicitly indicated as they are not classified as objects. Table 1 provides a comprehensive list of all features contained in the dataset, which allows a detailed understanding of its structural composition. The cleanliness of the dataset should also be highlighted, as there are no missing values. Therefore, it is no need for imputation strategies. However, it should be noted that the characteristic "education" comprises three undocumented categories and the feature "marital status" one. Although this does not affect the overall reliability of the data set, it does require careful consideration when analyzing the data.

Variable Name	Description	Role	Type	Missing Values
ID	Client Identification number	ID	Integer	No
credit_amount	Amount of given credit in TWD (includes individual and family/supplementary credit limit)	Feature	Integer	No
gender	Gender (1 = male, 2 = female)	Feature	Integer	No
education	Education (1 = graduate school, 2 = university, 3 = high school, 0,4,5,6 = others)	Feature	Integer	No
marital_status	Marital status (0,3 = others, 1 = married, 2 = single)	Feature	Integer	No
age	Age in years	Feature	Integer	No
Scale for pay_0 to pay_6 : (-2 = No consumption, -1 = paid in full, 0 = use of revolving credit (paid minimum only), 1 = payment delay for one month, 2 = payment delay for two months, ... 8 = payment delay for eight months)				
pay_1	Repayment status in September, 2005 (scale same as above)	Feature	Integer	No
pay_2	Repayment status in August, 2005 (scale same as above)	Feature	Integer	No
pay_3	Repayment status in July, 2005 (scale same as above)	Feature	Integer	No
pay_4	Repayment status in June, 2005 (scale same as above)	Feature	Integer	No
pay_5	Repayment status in May, 2005 (scale same as above)	Feature	Integer	No
pay_6	Repayment status in April, 2005 (scale same as above)	Feature	Integer	No
bill_amt_1	Amount of bill statement in September, 2005 (TWD)	Feature	Integer	No
bill_amt_2	Amount of bill statement in August, 2005 (TWD)	Feature	Integer	No
bill_amt_3	Amount of bill statement in July, 2005 (TWD)	Feature	Integer	No
bill_amt_4	Amount of bill statement in June, 2005 (TWD)	Feature	Integer	No
bill_amt_5	Amount of bill statement in May, 2005 (TWD)	Feature	Integer	No
bill_amt_6	Amount of bill statement in April, 2005 (TWD)	Feature	Integer	No
pay_amt_1	Amount of previous payment in September, 2005 (TWD)	Feature	Integer	No
pay_amt_2	Amount of previous payment in August, 2005 (TWD)	Feature	Integer	No
pay_amt_3	Amount of previous payment in July, 2005 (TWD)	Feature	Integer	No
pay_amt_4	Amount of previous payment in June, 2005 (TWD)	Feature	Integer	No
pay_amt_5	Amount of previous payment in May, 2005 (TWD)	Feature	Integer	No
pay_amt_6	Amount of previous payment in April, 2005 (TWD)	Feature	Integer	No
label	Default payment next month (1=yes, 0=no)	Target	Binary	No

Table 1: List of all features

This table gives a brief description about all features and their types. It includes also the possible values for categorical features and the meaning behind each number.

To provide a deeper understanding of the interconnections between the features “pay”, “bill_amt”, and “pay_amt”, an equation is formulated to show their relationships. Here, the bill amount (X) represents the expenditure made by the client in a given month, while the previous payment amount (Y) denotes the repayment made by the client in the preceding month. The repayment status indicates whether the amount paid is sufficient or if there remains an outstanding credit amount. Notably, the bill amount for the subsequent month accumulates the leftover amount from the previous month. This relationship is expressed through the following equation:

$$X_0 = \sum_{t=1}^{5} X_t - Y_t \quad (1)$$

3.3 Descriptive statistics of the features and the label

The label analysis shows that of the 30,000 instances in the dataset, 6,636 are categorized as default. This corresponds to a default rate of 22.12%. However, this also shows a great imbalance within the dataset, as a significant majority of the instances belong to the non-error class. The imbalance of the data set represents a challenge for modeling, as it could potentially lead to biased model predictions that favor the majority class. Therefore, it is essential to address this issue to ensure the development of fair and accurate prediction models.

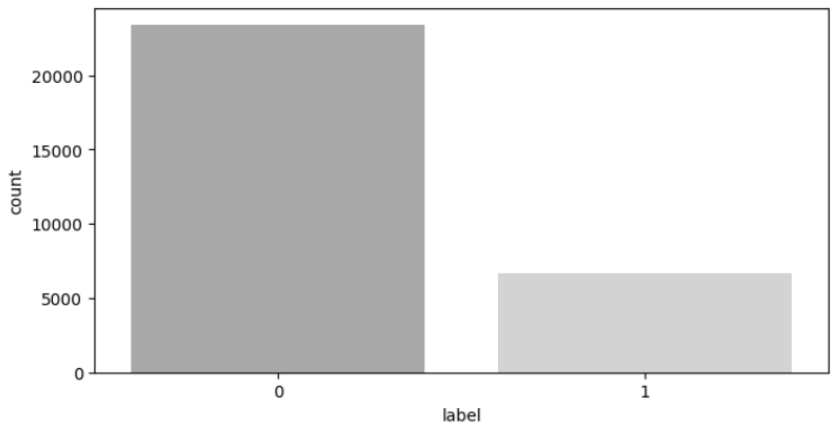


Figure 1: Share of loans defaulted

The figure illustrates the difference between non-defaulted (0) and defaulted (1) loans of clients in the whole dataset.

The dataset shows a wide range of values for the feature "credit amount", which describes the maximum credit amount, one client is allowed to have. The average credit amount is New Taiwan Dollar (TWD) 167,484, which corresponds to approximately United States Dollar (USD) 5,139, with a minimum of TWD 10,000 (~USD 206) and a maximum of TWD 1,000,000 (~USD 30,684). Figure 1 shows a right-skewed distribution, with the most common credit amounts being TWD 50,000 (USD 1,534), TWD 20,000 (USD 613) and TWD 30,000 (USD 920). Credits above TWD 500,000 are particularly rare, with the majority of loans below TWD 210,000.

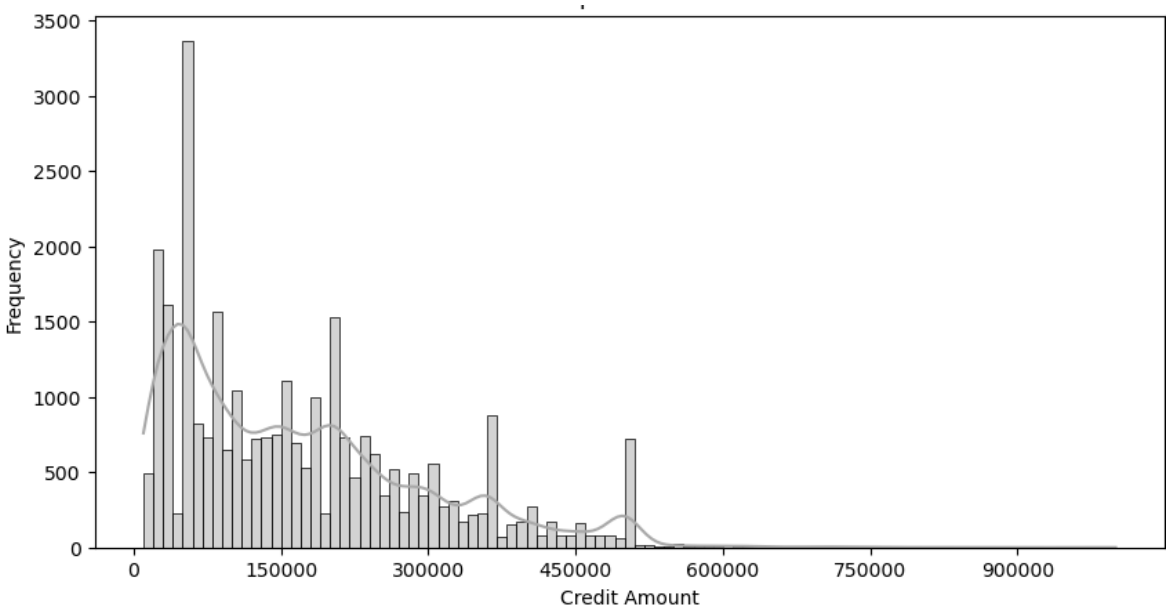


Figure 2: Distribution plot of maximum credit amount

The distribution plot is showing the maximum credit amount in million TWD for each client.

As far as the feature "gender" is concerned, the data shows that men make up around 39.6% of the dataset. The analysis shows a relatively even distribution of credit amount limits between the genders, as shown in figure 2. While men have slightly lower values for the second quartile, but higher values for the third and fourth quantile, women have a higher outlier maximum value, which reaches 1 million TWD. In addition, the average credit card limit for men is lower. Nevertheless, it is worth observing that a higher proportion of loans defaulted by men (24.17%) compared to women (20.78%), which could suggests a higher probability of default among male borrowers.

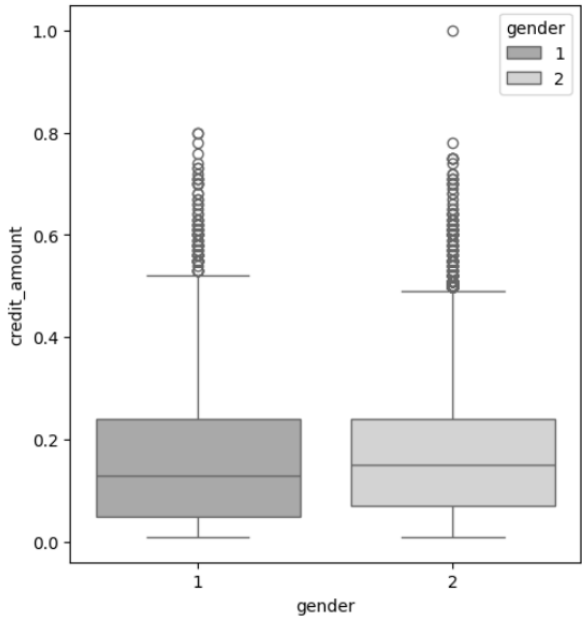


Figure 3: Boxplot with outliers by genders

The distribution of the credit amount limit appears relatively equitable across genders. Males exhibit a slightly smaller Q2 but larger Q3 and Q4, accompanied by a lower mean. Conversely, females display a higher outlier maximum value, reaching 1 million TWD.

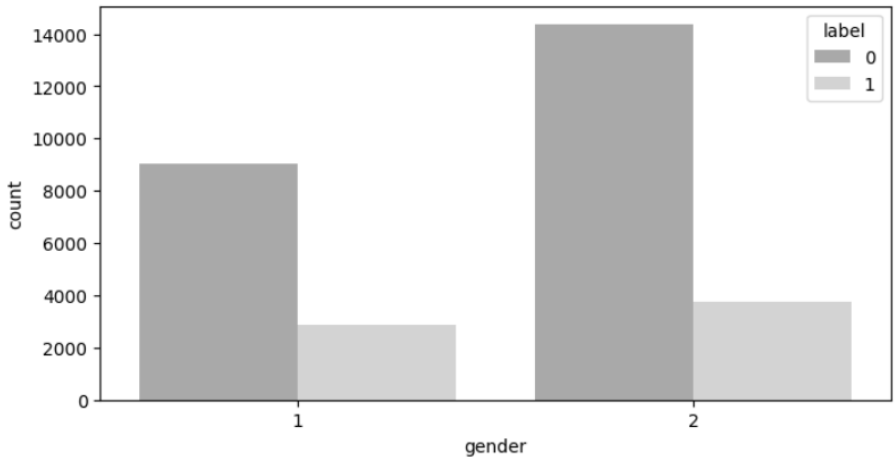


Figure 4: Distribution of defaults by men and women

In addition to the overall higher number of women included in the dataset, the share of defaulted loans is different for men (1) and woman (2).

Furthermore, there appears to be a correlation between the level of education and the default rate of credits. This trend suggests that higher levels of education are associated with lower default rates. Therefore, credits from clients with a master's degree only defaulted in 19.23% of cases, while bachelor's and high school graduates had default rates of 23.73% and 25.16%, respectively. In addition, it can be observed that clients tend to take higher credit amounts as

their level of education increases. This may indicate a higher income level as education level increases, although this correlation cannot be definitively confirmed without additional income data. Nevertheless, the distribution of clients is quite diverse, with most people having a university degree, followed by high school graduates and clients who only have a high school diploma.

For the feature "marital status", the dataset shows a slightly higher proportion of married clients compared to single clients. Interestingly, single clients have a higher average credit limit but also show higher outliers in their borrowing behavior. Nevertheless, the default rate of married people is 23.47% and therefore higher compared to 20.93% for singles.

With regard to the feature "age", the average age of the clients in the dataset is around 35 years, with a minimum age of 21 and a maximum age of 79. The age distribution within the dataset appears to be right-skewed, with the most common age being 29, 27 and 28.

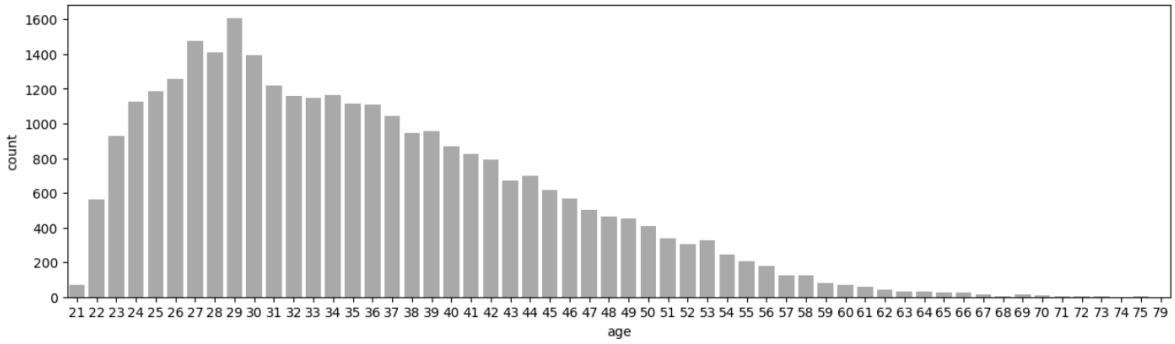


Figure 5: Distribution of age

In this case, the data show a right-skewed distribution with the most people being in the age around 30.

When categorizing clients into age intervals of five years, a clear trend emerges indicating that both younger and older people are more likely to default on their credits. Interestingly, the 30-34 age group has the lowest number of credit defaults, indicating a possible peak in financial stability or sense of responsibility in this age group.

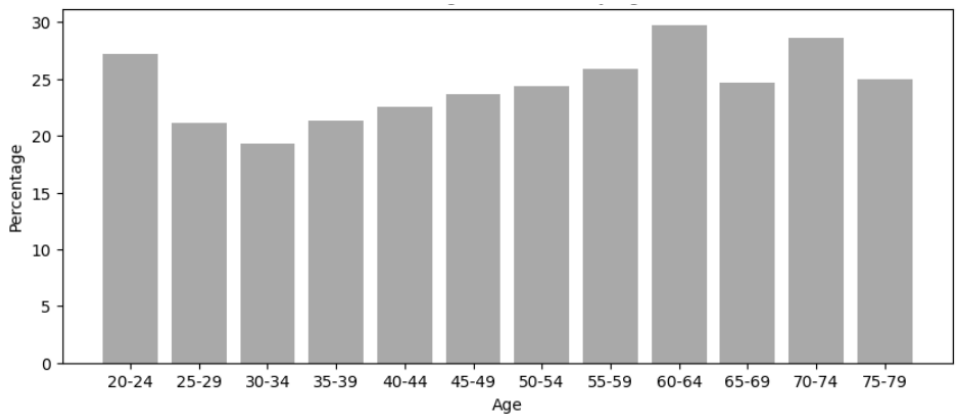


Figure 6: Percentage of defaults by age groups

At first, the percentage of defaults decreases by the increasing age of the group. After the turning point at age 30-34, the percentage of defaults increases again.

The repayment status feature demonstrates a clear trend where the likelihood of credit default correlates with a deterioration in repayment status over time. Initially, in April, a substantial majority of clients, approximately 90%, exhibited a favorable repayment status (denoted by values of -2, -1, or 0). However, by September, this proportion had declined to 77%, suggesting a decline in the financial health of borrowers as the loan maturity progresses.

The analysis of the "bill amount" feature indicates a similar picture, as the average bill amount increased over the past six months by 22.3%. Compared to a rather stable repayment amount, it may indicate, that clients are not capable of paying back their credits or their spending habits increased a lot in those six months.

The correlation analysis shows insights into the relationships among the features within the dataset. Notably, the correlation between "gender", "education", "marital status", and "age" appears to be relatively low, indicating limited linear dependence. However, a slightly higher correlation is observed between "age" and "marital status". Stronger correlations exist among the other features "bill amounts", "repayment amount", and "repayment status", which is expected given their direct relationship as described by equation 1 in section 3.2. Regarding correlation with the label, the findings suggest that most individual features have correlations lower than 0.1, despite "repayment status" and "credit amount" showing a stronger correlation.

	Label	Gender	Education	Marital status	Age	Credit amount
Label	1					
Gender	0.04	1				
Education	0.028	0.0142	1			
Marital status	0.0243	0.0314	0.1435	1		
Age	0.0139	0.0909	0.1751	-0.4142	1	
Credit amount	0.1535	0.0248	-0.2192	-0.1081	0.1447	1

Table 2: Correlation matrix for a subset of features

The correlation matrix is showing the relationship between the label and a few features. This helps to understand the dataset and should be kept in mind for the building of the models.

4. Methodology

4.1 Research Design

This research aims to compare different models for predicting credit card defaults. As such, the selected models require a dataset comprising diverse observations. Each observation represents therefore one client. Given the supervised nature of the learning task, each observation consists of various features and a corresponding label indicating whether the client's credit defaulted or not. The features include thereby information about the client and their spending behavior.

As this is a binary categorization, classification machine learning models are used to solve the prediction problem. These machine learning models aim to discover patterns in the data in order to accurately classify each observation.

To establish a baseline for comparison, the Logistic Regression model without a penalty term is chosen as benchmark. This model is known for good performance in classification task but is rather easy to implement and has a low need for computational resources. In addition, further models are selected based on previous research results and the diversity of machine learning models. Lasso, Ridge and Elastic Net models are implemented to directly compare them with standard Logistic Regression and evaluate the impact of adding regularization terms on the prediction performance (Landwehr et al., 2003).

Random Forest and XGBoost are chosen to represent decision tree based algorithms, leveraging their ability to handle complex relationships and interactions in the data. These models offer robustness and scalability, making them suitable models for comparison (González et al., 2020).

For other non-linear models, Neural Networks, SVM, and KNN are considered based on their demonstrated effectiveness in past research (Bianchi et al., 2020). These models are known for their ability to capture intricate patterns and relationships in the data (X. Niu & Suen, 2012).

To evaluate model performance, unseen data is utilized, aligning with the objective of predicting new defaults from clients rather than optimizing performance on the training data.

4.2 Data processing and feature engineering

4.2.1 Data processing

The dataset used in this study comes from the UCI Machine Learning Repository, which can be accessed via the provided application programming interface. This dataset includes 23 features as well as a target variable indicating a binary outcome.

During the initial review, it is found that the feature names in the dataset are not directly interpretable. To improve the clarity and interpretability of the dataset, a pre-processing step is performed. In particular, the features received a name closer to their description, as shown in table 1.

In addition to renaming features, certain categorical variables, such as education and marital status, contain multiple categories denoting unknown. To address this issue and streamline the dataset, these different categories are grouped into a single unified category.

As no missing data was found, there is no need for any input-algorithm.

4.2.2. Feature engineering

In the subsequent stage, a new variable is introduced to assess the extent to which clients have utilized their credit limits. In essence, these variable measures the closeness of clients' current debt to their permissible credit limit. This evaluation is performed for each month included in the dataset. The newly introduced variable, termed “closeness”, is the percentage of credit amount already used by the client and is calculated as follows:

$$Closeness = \frac{credit\ amount - bill\ amount_X}{credit\ amount} \quad (2)$$

The rationale behind this feature engineering lies in the idea that as clients approach their credit limits, the likelihood of default should increase.

4.2.3 Transforming dataset

Following the feature engineering step, the dataset was split into two groups of features: categorical and numerical.

The categorical features include gender, education, and marital status. To ensure consistent treatment and avoid erroneous ranking amount the categories, these features are transformed

using the OneHotEncoder by scikit learn, resulting in one dummy variable for each category of the features in the final data frame.

The numerical features are scaled using the StandardScaler technique. This process standardizes the data to exhibit a mean of zero and a standard deviation of one, a crucial step for machine learning models that rely on distance-based metrics. By facilitating the recognition of meaningful patterns and the resulting improvement in prediction accuracy, this scaling method has also increased the model's resistance to outliers.

Addressing concerns related to data imbalance, various techniques can be employed. This study uses the Synthetic Minority Oversampling Technique (SMOTE) algorithm as it does not lead to a significant loss of observations, unlike the under-sampling or over-sampling methods. This algorithm generates synthetic data for the minority class and is labeling the generated data by the KNN algorithm. Therefore, the SMOTE function rebalances the data to achieve the same number of defaulted and non-defaulted credits. This is important as otherwise the models can be biased to the majority class (Chawla et al., 2002). The result of using the SMOTE algorithm therefore improves the performance of the machine learning models at a 1% significant level (Alam et al., 2020).This positive effect is also evident in the dataset used in this study (Faraj et al., 2021). The new distribution of the dataset used in this study can be seen in figure 7, where both classes have the same number of observations.

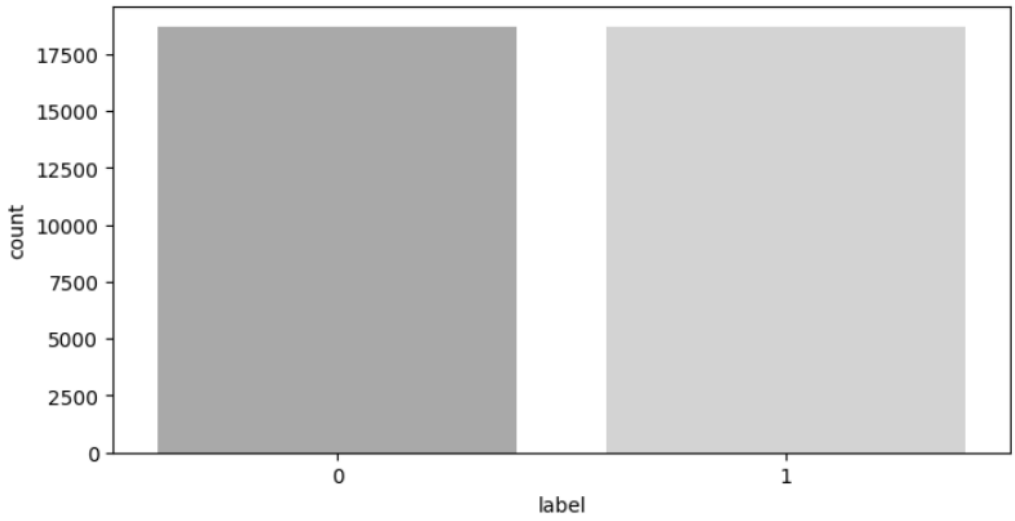


Figure 7: SMOTE dataset

This graph shows the classification of observations after the rebalancing by the SMOTE algorithm. Both classes have the same number of observations, which is important for the training of the model, in order to prevent a bias in direction of non-defaulting loans.

Following the rebalancing step, the dataset is split into separate training and testing samples. To establish a robust foundation for model training, 80% of the available data is allocated to the training dataset. Given the absence of temporal distinctions among data observations, the selection is made randomly with a random state of 42.

4.3 Description of models

The following section presents an overview of the models used in this study, whereas the information are based on the book of G. James et al. (2021).

4.3.1 Logistic Regression

A Logistic Regression is a statistical method used for binary classification tasks. It models the probability that an observation belongs to a particular class based on one or more features. It works by transforming the linear combination of these features into a probability by using the sigmoid function:

$$P(y = 1 \,|X) = \frac{1}{1 + e^{-\beta^T X}} \tag{3}$$

Where:

- $P(y = 1 \,|X)$ is the probability that the outcome variable Y equals 1
- X is a vector of features
- β is a vector of coefficients associated with each feature

By training the model, the model estimates the coefficients, which then can be used to predict the probability that a new observation belongs to the positive class. If this predicted probability is greater than 0.5, the observation is classified as positive (coded as 1). Otherwise, the classification belongs to the negative class (coded as 0).

The Logistic Regression without penalty term is a very computational efficient model but can be vulnerable to overfitting and be sensitive to outliers. Those problems can be reduced by using different regularization terms.

4.3.2 Lasso

By incorporating a Lasso regularization into the Logistic Regression, the objective function contains an additional term that penalizes the absolute values of the coefficient. Thereby, some coefficients can become exactly zero and the model performs a feature selection. The term added to the objective function of the Logistic Regression, also called L1 regularization and can be written as follows:

$$\|\beta\|_1 = \sum_i |\beta_i| \tag{4}$$

$$min \sum_{i=1}^{P} \|\beta^T x_i - y_i\|^2 + \alpha\|\beta\|_1 \tag{5}$$

Where:

- $\|\beta\|_1$ represents the L1 norm of the coefficient vector β
- $\beta^T x_i$ is the predicted outcome for the observation based on the coefficients and features of the model
- $\|\beta^T x_i - y_i\|^2$ is the squared loss between the predicted outcome and the true outcome
- α controls the magnitude of the penalty term

The new objective function finds a balance between fitting the data well and keeping the magnitude of the coefficient small. This reduces overfitting but can also lead to a selection bias if important predictors are incorrectly excluded from the model.

4.3.3 Ridge

In comparison to the Lasso regularization, the Ridge regularization, also known as L2 regularization, cannot shrink features exactly to zero. Therefore, it has no selection bias and still can reduces the chances of overfitting. Nevertheless, the fact that it cannot exclude features from the model, no feature selection is done, and also unnecessary features are included in the model.

Incorporating Ridge regularization into Logistic Regression involves the addition of a penalty term, typically represented as follows:

$$\|\beta\|_2 = \sum_i \beta_i{}^2 \tag{6}$$

(7) $$min \sum_{i=1}^{P} \|\beta^T x_i - y_i\|^2 + \alpha\|\beta\|_2^2$$

Where:

- $\|\beta\|_2$ represents the L1 norm of the coefficient vector β

4.3.4 Elastic Net

Elastic Net regularization is a combination of the two regularization Lasso and Ridge, offering a balance between feature selection and coefficient shrinkage. Nevertheless, by adding both regularization terms, the complexity and therefore the computational cost increases. The objective function when adding both penalty terms is the following:

(8) $$min \sum_{i=1}^{P} \|\beta^T x_i - y_i\|^2 + a\eta\|\beta\|_1 + a(1-\eta)\|\beta\|_2^2$$

Where:

- η is a mixing parameter that determines the balance between the L1 and L2 regularization

Figure 8 demonstrates the three regularizations in comparison. While Lasso shrinks the coefficient β_1 to zero, Ridge and Elastic Net only reduces its value. Additionally, the form of the Elastic Net regularization is a combination of the Lasso and Ridge regularization.

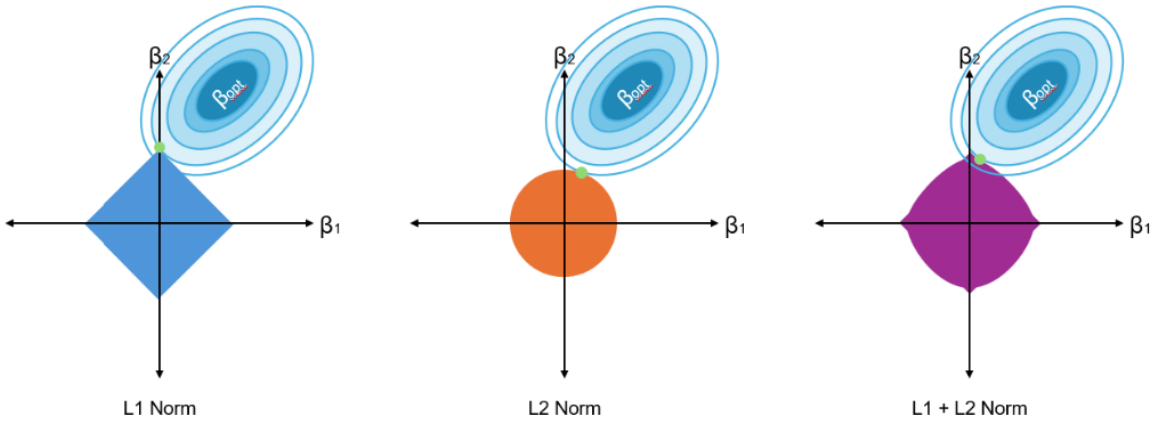


Figure 8: Logistic Regression regularizations

Each of the three illustration shows a different regularization of the Logistic Regression. Starting with the L1 regularization which shows a case of feature selection, as is shrinks a coefficient to zero. In comparison, the l2 regularization only shrinks the coefficient but not until zero. The Elastic Net illustration can be visualized by the combination of the L1 and L2 regularization.

4.3.5 XGBoost

Extreme Gradient Boosting is an ensemble learning method that combines the prediction of multiple decision trees, to build a strong predictive model. It is only one type of many boosting algorithm existing but is known as one of the best performing. In comparison to the Logistic Regression models, XGBoost is capable of capturing nonlinear relationships between the features and the label. In this study the loss function is the typical negative log-likelihood function of the Logistic Regression model. By including it in the objective function of the XGBoost, the function can be defined as following:

$$\mathcal{L}(y_i, \hat{p}_i) = -[y_i \, log(\hat{p}_i) + (1 - y_i) \, log(1 - \hat{p}_i)] \tag{9}$$

$$min \sum_{i=1}^{N} \mathcal{L}(y_i, \hat{p}_i) + \sum_{k=1}^{K} \Omega(f_k) \tag{10}$$

Where:

- N is the number of observations
- y_i is the true label
- $\hat{p}_i$ is the predicted probability that the observation belongs to the positive class
- $\mathcal{L}$ is the loss function, which measures the difference between the true labels and the predicted probabilities
- K is the number of decision trees
- $\Omega(f_k)$ is the regularization term applied to each tree f_k which penalizes model complexity to prevent overfitting.

Figure 9 demonstrates the idea of XGBoost, as each weak learner learns from the result of the learner before. This iterative process allows the model to gradually improve its performance by focusing on the areas where the previous learner performed poorly.

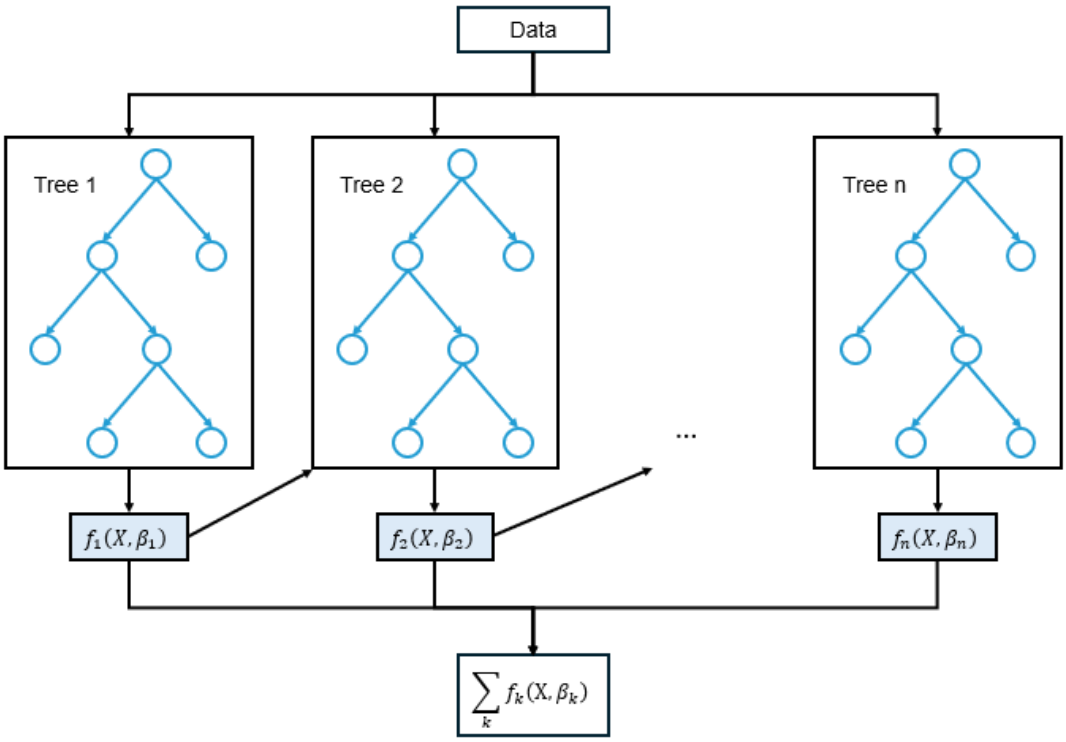


Figure 9: Extreme Gradient Boosting

This figure shows the structure of XGBoost algorithm, where each new decision tree is based on the result of the tree before. Therefore, the trees can learn from the errors of the previous tree.

4.3.6 Random Forest

Random Forest is a non-linear ensemble learning method that builds multiple decision trees during the training phase. Each individual tree results in a classification of 0 or 1, whereas the average of above 0.5, indicates that the whole model classifies the observation as one and vice versa below. Each tree is thereby trained independently on a random subset of the features. This helps the model to decorrelate the trees and reduces overfitting.

This process is illustrated in the figure 10. At first, the input data consists of the training set, which lets the model learn each individual tree. Because of that, those trees are not exactly the same, but can differ. Therefore, the Random Forest algorithm does not need an objective function to minimize.

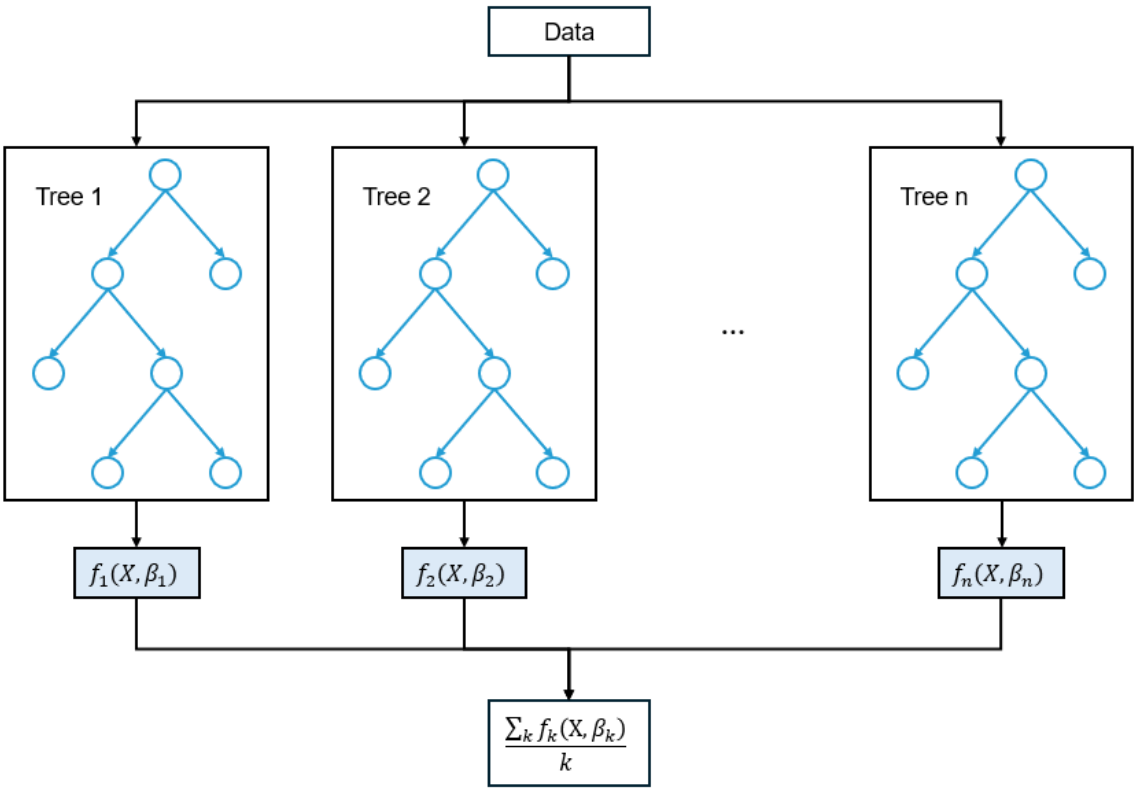


Figure 10: Random Forest

The Random Forest is not learning from the previous tree, but instead generates independent trees. The average result of all trees indicates than the final classification.

In comparison to the XGBoost algorithm, Random Forest builds decision trees parallel and not sequentially. For that reason, individual trees cannot learn from the errors of the previous trees but is also less prone to overfitting.

4.3.7 K-Nearest Neighbors

The main idea behind KNN is, that similar instances are likely to have similar outcomes. In particular, the algorithm does not explicitly learn a model, instead memorizes the training dataset, and makes predictions for new instances based on their similarity to existing data points.

Therefore, the majority of the values of its k nearest neighbors decide, what the classification of the new observation will be, as illustrated in figure 11. The parameter k can be selected based on the complexity of the problem and the characteristics of the dataset.

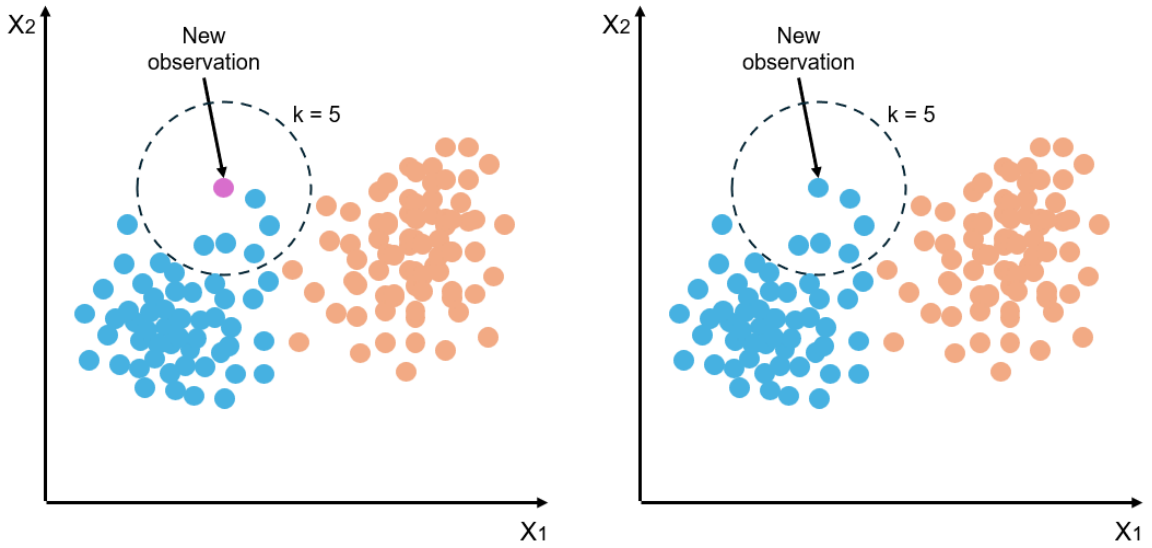


Figure 11: K-Nearest Neighbors

The figure is showing the process of predicting a new observation, based on the data already reviewed by the model. The classification is made by the nearest neighbors of the data point, which therefore seem to be the best indicator for the new classification.

Additionally, KNN offers the flexibility to give each observation a weight for the final classification. Therefore, the algorithm is weighting the observations inversely proportional to the distance from the query point, so that closer neighbors have a greater influence. The opposite is an equal weighting for all data points. Both methods are evaluated in the hyperparameter tuning in this study.

An advantage of the KNN model is, that it can manage new input data easily, as is does not need a training phase but just adds the new or changing observations in the memory. In addition, it makes no assumption about the overall distribution of the dataset. Nevertheless, for big datasets the model can be computationally expensive and sensitive to outliers, and it is not the best model to capture non-linear datasets.

4.3.8 Neural Network

Neural Networks are a class of machine learning algorithms inspired by the structure and function of the human brain. Therefore, a Neural Network consists of interconnected neurons organized into layers. The first layer, also called input layer, receives the raw data. In the second step, the model creates hidden layers to process the data in a series of non-linear transformations to produce the output in the last layer. In order to create each neuron during the feedforward process, the model uses the weighted sums of the inputs followed by the activation function. These weights are written as β_{10}^{2}, where the two at the top symbols the

starting layer and the numbers at the bottom the connected neurons. For a clearer visualization not all weights are included in the figure 13. The activation function then transforms the result $z_1^2$to a new neuron a_1^2, as shown in figure 12. Again, the number at the top indicates the layer and the number at the bottom shows the number of the neuron.

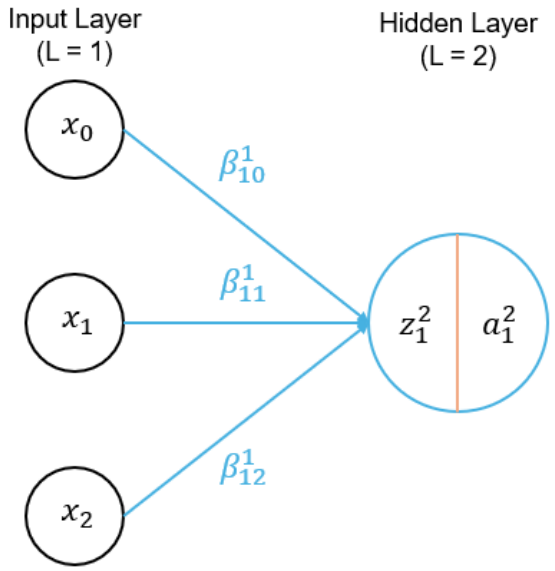


Figure 12: Neuron with activation function

The figure is showing the actual process of the creation of a new neuron. First, the input data is used to get a result, which than gets transformed into a new neuron.

For this research, the rectified linear unit function (11) is used as activation function, as it is known to solve classification tasks efficiently and helps to reduce the vanishing gradient problem.

(11) $$f(x) = max(0, z)$$

Where:

- z is weighted sum of the outputs of neurons from the previous layer

After the feedforward pass, the prediction of the network is compared to the true labels of the training set. Hence, the model uses a cost function to validate the accuracy of the predictions. In this research the Adaptive Moment Estimation (Adam) is used for that, as it is particularly well-suited for high-dimensional parameter spaces. Additionally, the learning rate is set to be adaptive, and the maximum iteration is limited to 1000.

During training, the backpropagation process employs to adjust the parameters of the network. Backpropagation involves calculating the gradient of the loss function with respect

to each weight by applying the chain rule of calculus, then updating the weights in the direction that minimizes the loss. This iterative process continues until the network's performance on the training data is optimized. The end results is then visible in figure 13, where the value of each weight is visualized as the strength of each arrow.

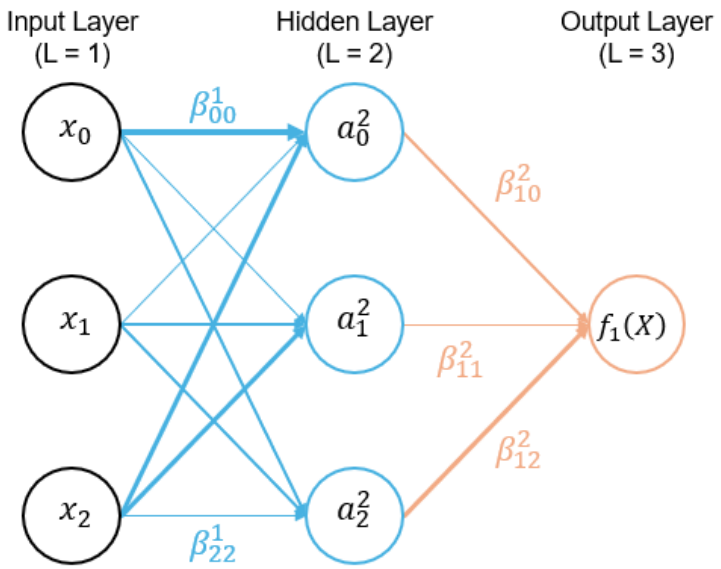


Figure 13: Neural Network

This illustration shows an example of a full Neural Network with one hidden layer. The value of the weights is demonstrated by the strength of the arrows.

On the one hand, the advantages of a Neural Network are their ability to learn complex patterns and their flexibility. Neural Networks can learn highly non-linear relationships making them suitable for a wide range of tasks. And because of the different settings of layers and neurons, Neural Networks can explicitly be adjusted to suit different problem domains. On the other hand, this complexity makes them difficult to interpret or compare and computationally intensive.

4.3.9 Support Vector Machine

SVM are versatile machine learning models that can be used for both classification and regression tasks. The objective of SVM is to find a hyperplane in an n-dimensional space that uniquely classifies data points, where “n” represents the number of features in the dataset. This optimization process aims to maximize the margin between the data points and the hyperplane, by maximizing the following equations:

(12) $$\max_{\omega,b} \|\omega\|^{-2}$$

(13) $$\omega^T x + b \geq 1 \quad \forall x \in C_1$$

(14) $$\omega^T x + b \leq -1 \quad \forall x \in C_2$$

Where:

- ω is the weight vector of the hyperplane
- b is the bias term
- x is the feature vector
- C_i is the set of data points belonging to the i-class

The hyperplane is therefore defined by a set of weights that determine its orientation and position in the feature space. Figure 14 shows an example of a dataset with two features and a corresponding one-dimensional hyperplane, which illustrates the basic concept of SVM classification.

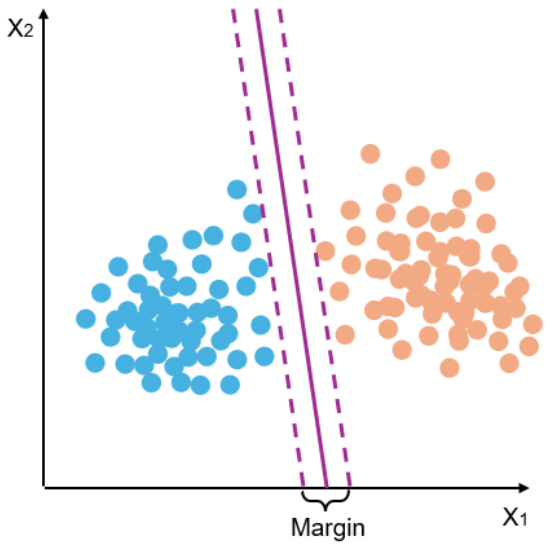


Figure 14: Support Vector Machine

This figure shows the mechanism of the SVM model. The algorithm finds a hyperplane that clearly separates the two classifications points and is choosing the hyperplane with the highest margin to the data nearest data points.

SVM has also the ability to manage non-linear problems by mapping the input features into a higher dimensional space using a kernel function. This allows the data to be separated by a hyperplane in the higher dimensional space, even if they are not separable in the original feature space. In this research, the Radial Basis Function (RBF) (15) kernel is used as it is suitable for non-linear problems.

(15) $$K(x_i, x_j) = exp\left(-\gamma \cdot \|x_i - x_j\|^2\right)$$

Where:

- x_i and x_j are the input vectors
- γ is a parameter that determines the width of the RBF kernel ($\gamma = \frac{1}{2\sigma^2}$)
- $\|x_i - x_j\|^2$ denotes the squared Euclidean distance between the input vectors

SVMs also contain a regularization parameter (C) to strike a balance between maximizing the margin and minimizing the classification error. Therefore, this parameter helps to control the trade-off between bias and variance, making the model more robust against outliers and prevent overfitting.

The advantages of SVMs are a relatively easy implementation and understandability, making them suitable for various applications, even with small datasets. However, it should be noted that SVMs can be computationally intensive, especially with large datasets or complex kernels. In addition, SVMs are sensitive to the parameters chosen, such as the choice of kernel and regularization parameters.

4.3.10 Hyperparameter tuning and cross validation

Hyperparameter tuning is a crucial aspect of machine learning optimization, as it enables the exploration of different configurations of model parameters. With many parameters available, manual tuning can be impractical and may not result in the best model fit. Therefore, hyperparameter tuning automates this process by searching for the optimal hyperparameter values within a predefined space.

Two common techniques for hyperparameters tuning are the grid search and the random search. In the grid search, the algorithm systematically evaluates all possible combinations of hyperparameters within the predefined space. This exhaustive approach guarantees that the best combination is found. In contrast random search, selects the hyperparameters at random from the predefined space, which is more efficient and requires less computational effort. However, it does not guarantee that the best combination will be found.

For this study, grid search is chosen because it ensures the optimal selection of hyperparameters within the predefined space. The hyperparameter space is therefore defined on the basis of previous research using the same or similar datasets (Faraj et al., 2021; Islam et al., 2018; Soui et al., 2018).

To provide a comprehensive overview of the hyperparameter tuning process and its results, table 3 summarizes the hyperparameter spaces investigated in this study and the best parameters for the machine learning models found.

	Hyperparameter tuning	Best hyperparameters
Lasso	$C = 10^{(-3+i\frac{5}{29})}, i \in [0,29]$ *	$C = 0.3857$
Ridge	$C = 10^{(-3+i\frac{5}{29})}, i \in [0,29]$ *	$C = 13.7382$
Elastic Net	$C = 10^{(-3+i\frac{5}{29})}, i \in [0,29]$ *	$C = 0.3857$
XGBoost	1. $Learning\ rate = [0.01, 0.05, 0.1]$ 2. $Num\ of\ trees = [50, 100, 150]$ 3. $Max\ tree\ depth = [None, 3, 6, 9]$	1. $Learning\ rate = 0.05$ 2. $Num\ of\ trees = 100$ 3. $Max\ tree\ depth = 3$
Random Forest	1. $Num\ of\ trees = [100, 200, 300]$ 2. $Max\ tree\ depth = [None, 6, 9, 12]$	1. $Num\ of\ trees = 300$ 2. $Max\ tree\ depth = 9$
KNN	1. $k \in [1,100]$ 2. $weights \in \{uniform, distance\}$	1. $k = 45$ 2. $weight = uniform$
Neural Network	1. $Hidden\ layer\ size = [5, 10, 15]$ 2. $\alpha\_term = [0.0001, 0.001, 0.01]$	1. $Hidden\ layer\ size = 5$ 2. $\alpha\_term = 0.001$
SVM	1. $C = [0.1, 1, 10]$ 2. $Polynomial\ terms = [2, 3, 4]$	1. $C = [0.1, 1, 10]$ 2. $Polynomial\ term = 2$

Table 3: Hyperparameter tuning

This table shows the hyperparameter space for each machine learning model used in the study and the best hyperparameter found by using GridSearchCV.

** Giving 30 numbers between* 10^{-3} *and* 10^{2}

To further improve the performance of the machine learning model, the function GridSearchCV not only includes the option of hyperparameter tuning but also the option of cross validation. Cross validation divides the training dataset into serval subsets to iteratively train the model on one part and validate it on the remaining part. An illustration of this process for k = 4 can be found in figure 15, whereas the dataset is not only split into one training and test set, but numerous subsets.

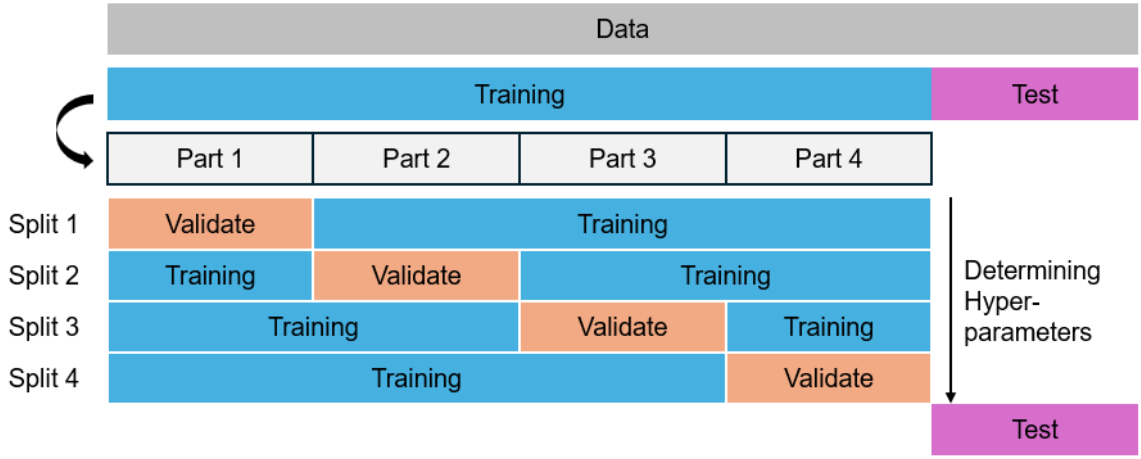


Figure 15: Cross validation

In comparison to the standard process, where the data is only split into a training and testing set, cross validation is further splitting the dataset into a validation part. This validation part is than used for every data part of the dataset.

This process ensures that each data point is used for both training and validation, reducing the risk of overfitting and providing a more reliable estimate of model performance.

The choice of the number of iterations, k, depends on factors such as the size of the dataset, computational resources, and the desired level of accuracy. For this study, ten iterations (k = 10) is chosen, which is commonly used in other research (Marcot & Hanea, 2020).

4.4 Evaluation of the models

In order to evaluate the performance of the different models and determine which model provides the best results, it is important to use meaningful metrics. While many studies use the accuracy of a machine learning model as the primary metric, it is important to note that accuracy alone does not provide a complete picture of prediction quality (Yacouby & Axman, 2020).

To gain a deeper understanding of the performance of a machine learning model, a confusion matrix is calculated, as shown in figure 16. The confusion matrix shows the number of correctly classified labels in relation to the incorrectly predicted labels. Based on this confusion matrix, key performance metrics such as accuracy, precision, recall and F1-score are calculated.

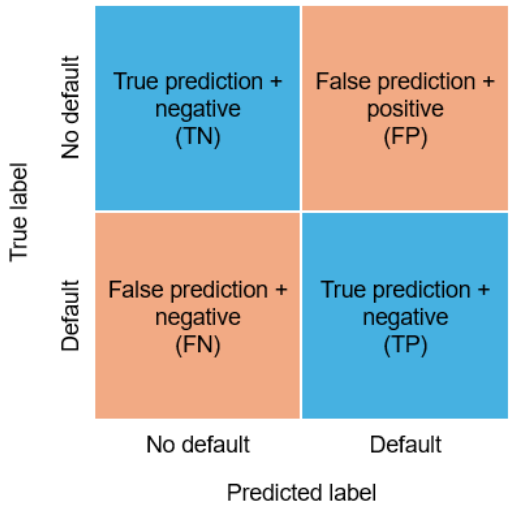


Figure 16: Evaluation confusion matrix

This figure shows the layout of the confusion matrix calculated for each machine learning model. Each cell is therefore the sum of the observations in regard to their true and predicted label.

In addition, the models are evaluated against unseen data, as shown in figure 15. This ensures that the performance evaluation is based on data on which the models have not been trained, allowing an assessment of their applicability in practice.

4.4.1 Accuracy

Accuracy is one of the simplest metrics for evaluating classification models. It measures the proportion of correctly classified instances out of the number of all instances. It is calculated as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{16}$$

This means, for example, if the model correctly classifies 90 out of 100 instances, the accuracy is 90%. A high accuracy therefore indicates that the models is making correct predictions. Nevertheless, it may not be the best metric alone, if the classes are imbalanced (Yacouby & Axman, 2020).

In terms of credit card default prediction, a model with high accuracy would make good predictions if a new observation is correctly classified as default or non-default. However, while a high accuracy means a large proportion of correctly classified instances, it is not necessarily only excellent at accurately classifying defaults. Therefore, it is important to also consider other metrics to capture that the ability to predict defaults alone correctly.

4.4.2 Precision

Precision measures the proportion of true positive among all positive predictions of the model. It indicates how many instances predicted as positive are actually positive. Mathematically, precision is calculated as follows:

$$Precision = \frac{TP}{TP + FP} \quad (17)$$

In simple terms, of all the instances the model labeled as positive, how many are actually positive (Yacouby & Axman, 2020).

A high precision indicates that the model is making few false positive predictions. This can be important if the cost of false positive is high. In terms of credit card default prediction, which would mean, a high precision model is likely to predict a credit card default correctly.

4.4.3 Recall

Recall, also known as sensitivity or true positive rate, measures the proportion of true positive instances correctly classified by the model out of all actual positive instances.

$$Recall = \frac{TP}{TP + FN} \quad (18)$$

In simple terms, of all the positive instances in the dataset, how many did the model correctly identify (Yacouby & Axman, 2020).

High recall means that the models capture most of the positive instances. This is important when it is crucial to capture all positive instances, even if that means capturing some false positive one. This is especially important for credit card default prediction, as it indicates that the model correctly captures instances that default.

4.4.4. F1-Score

The F1-score is the harmonic mean of precision and recall. It combines both into a single value and is often used as the most important metric to evaluate the performance of a classification model. The F1-score can be calculated as follows:

$$F1 - score = 2 * \frac{precision * recall}{precision + recall} \quad (19)$$

Therefore, a high F1-score indicates a good balance between precision and recall, meaning that the model strikes a good balance between accurately identifying defaults and minimizing false default predictions. This is important for real world scenarios whereas the

credit card institutes need to minimize the risk of default of their clients, but still need to accept as many clients as possible to increase their profits.

4.4.5 Further evaluation methods

The next step is not only to compare metrics to evaluate the models, but also to understand what the individual models are based on when making their decisions. This is done using feature importance, which can be used to determine the weight of each feature in the predictive model that led to the predictions. The way this feature works depends on the type of model used. In the case of linear models, the feature displays the coefficients β after the training process. In this way, the feature importance can provide a valuable insight into the underlying factors that determine the default behavior and helps to identify the most important predictors.

Considering the learning curve of a machine learning model is important because it shows how well the model learns from the training data and how its performance changes when applied to unseen data. A learning curve represents the error rates during the training and validation process as a function of the size of the training dataset. The learning curve can be used to identify problems such as overfitting and underfitting. Overfitting occurs when the model performs well on the training data but poorly on the validation data, while underfitting occurs when the model performs poorly on both the training and validation data.

Finally, the computation time is used as a measure of cost when evaluating each model. This approach complements the cost-benefit analysis by considering the practical implications of model performance. A machine learning model with exceptional performance may be impractical if its training time hinders scalability or significantly increases resource costs. To ensure consistency and reproducibility for other researchers, the computation time is measured during the training and fitting of the models using the Google Colab environment.

5. Results & Discussion

5.1 Benchmark

5.1.1 Metrics

The benchmark model, Logistic Regression, perform moderately well in predicting credit card defaults, achieving an F1-score of 0.35 and a recall of 0.24. However, the expected

disparity between the Logistic Regression without regularization term and those with L1 and L2 regularization is not as pronounced as anticipated.

Nevertheless, the models strike a reasonable balance between accurately identifying defaults and minimizing false default predictions, although there is still potential for improvement. The precision analysis shows that 0.69 of the predicted defaults are actual defaults, indicating a satisfactory performance in this aspect. Additionally, the Logistic Regression achieves an 0.81 accuracy rate in predicting correct classified instances overall within the dataset.

However, considering the crucial role of the recall rate in credit card default models, Logistic Regression may not be enough for credit card institutions, as it only correctly predicts 24% of all credit card defaults.

	Logistic Regression			
	No penalty	Lasso	Ridge	Elastic Net
F1-score	0.3524	0.3526	0.3515	0.3521
Accuracy	0.8095	0.8097	0.8093	0.8098
Precision	0.6881	0.6896	0.6874	0.692
Recall	0.2369	0.2369	0.2361	0.2361

Table 4: Logistic Regression results

This table includes the most important metrics for the Logistic Regression with and without penalty terms. It clearly shows the similarity between all models, making the Logistic Regression a robust benchmark for the advanced machine learning models.

The confusion matrices highlight the similarities in the model performance, showing that regularization term minimally affect the results. Therefore, Lasso and Ridge models classify only one instance differently from the Logistic Regression without penalty, while the Elastic Net diverges in only four instances.

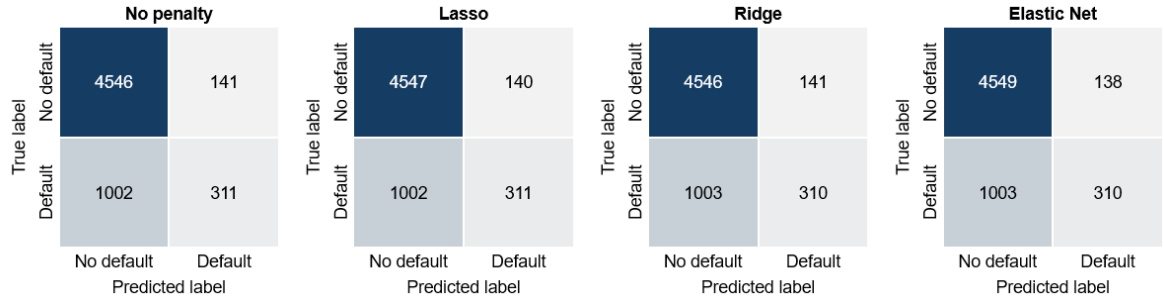


Figure 17: Logistic Regression confusion matrix

The confusion matrices of all Logistic Regression show the very similar classification of all instances in the dataset.

These results underline the robustness of the results, as changes in the regularization terms only minimally affect the model results. In addition, the similarity of these results to those reported in previous Logistic Regression studies underscores the validity of using the model without regularization as a benchmark (Faraj et al., 2021).

5.1.2 Feature importances

The reason for the similarity of the results becomes clear when examining the feature importance of the individual models. Lasso, as described in the methodology, can shrink certain features to zero, resulting in feature selection. In particular, Lasso discards all features related to bill amounts and retains only "bill_amt_3". This is due to the high correlation observed between these features.

Nevertheless, the top five features remain the same across all regressions. An interesting deviation, however, is that the Ridge regression assigns a higher weight to the feature "bill_amt_3" compared to other models, although Lasso eliminates all other bill amount features.

The importance of "pay_1" as the most important feature is in line with expectation, as it provides insight into the payment status one month before the due date. In addition, it is noteworthy that the newly created feature "closeness" has a consistently high importance in all models, making it a valuable addition to the models. In summary, all models have the same 5 most important features, which are shown in figure 18.

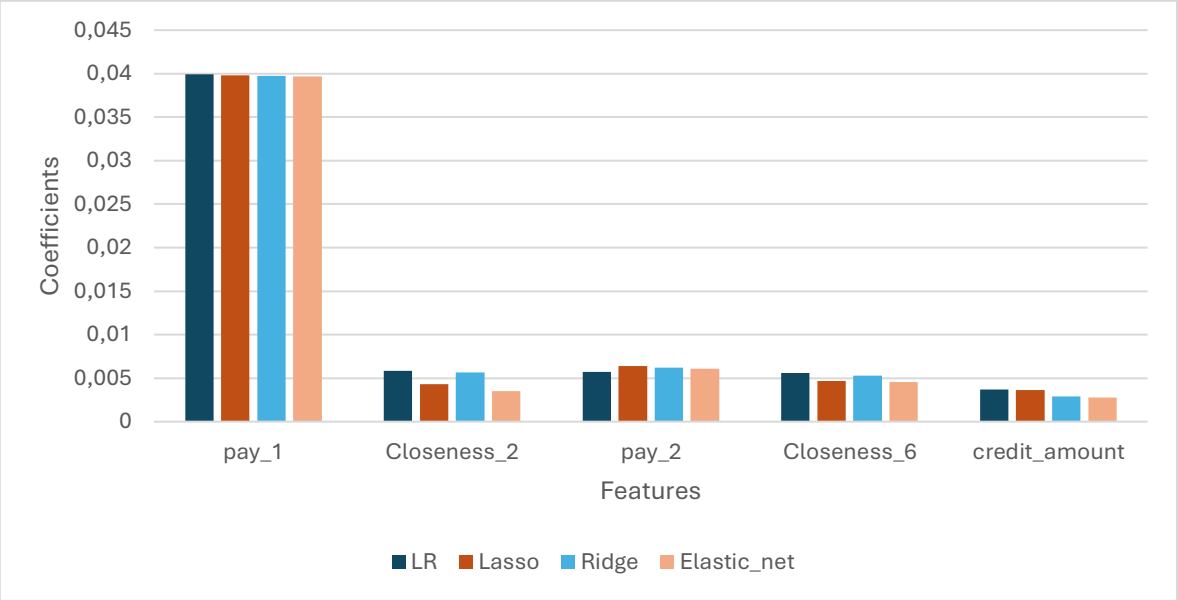


Figure 18: Feature importance for Logistic Regression

The feature "pay_1" is the most important feature for all Logistic Regression models. This makes sense, as it the status of the payments one month prior the deadline.

5.1.3 Benchmark learning curves

The rapid convergence and the intersection of the training and validation curves show that the Logistic Regression model is well trained. The uniformity observed in all four diagrams indicates that the Logistic Regression model without regularization serve as a good benchmark, which shows no signs of overfitting or underfitting.

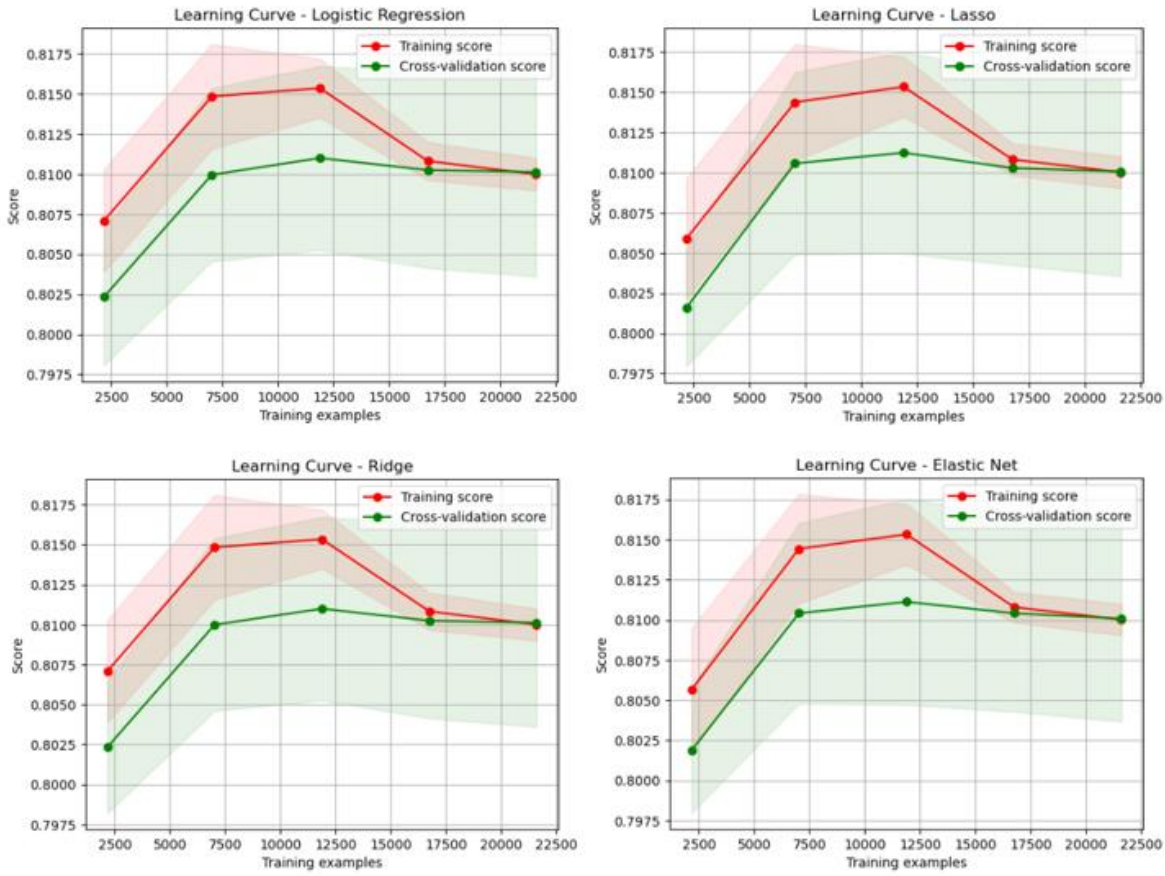


Figure 19: Logistic Regression learning curve

The training and validation curve are converting to each other, which is a sign of no over- or underfitting. Additionally, all curves look very similar, which shows again the robustness of the Logistic Regression model.

5.1.4 Benchmark computational cost

The computational costs for the Logistic Regression models vary significantly. The Logistic Regression model without a regularization term takes approximately one minute to train and fit. In contrast, the Lasso regularization increases the computational time to around ten minutes, which is even two minutes longer than the Elastic Net regression. Nevertheless, the Ridge regularization requires only one minute, making it comparable to the regression model without a penalty term.

5.2 Advanced machine learning models

5.2.1 XGBoost

The overall performance is significantly better compared to the Logistic Regression model. The F1-score, which reflects the balance between the accurate detection of payment defaults

and the minimization of incorrect default predictions, increased by 30%. In contrast, there is only a slight improvement in overall accuracy and precision. However, the crucial measure recall demonstrates that XGBoost correctly identifies 11% more defaults correctly than the Logistic Regression. This difference becomes clear by comparing the confusion matrix of both models (figures 17 and 20).

	XGBoost	Logistic Regression
F1-score	0.4599	0.3524
Accuracy	0.8203	0.8095
Precision	0.672	0.6881
Recall	0.3496	0.2369

Table 5: XGBoost results

The metrics for the XGBoost show a significant better result than the Logistic Regression. This is especially because of the improvement of the recall, where the XGBoost model can identify 11% more defaults correctly than benchmark model.

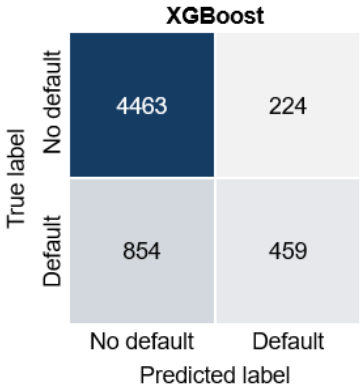


Figure 20: XGBoost confusion matrix

The confusion matrix illustrates the distribution of correct classified and misclassified instances. It is clear to see, that only 459 of 854 instances are correctly detected as default.

In regard to the importance of each feature, the XGBoost model places the highest focus on "pay_1", surpassing all other models and especially the Logistic Regression. In addition, it includes education, a feature that is not playing an important role in any other model.

The learning curve shows that the XGBoost model is remarkable robust. The training curve decreases steadily over time, indicating continuous improvement with additional data. As both curves are close to each other and the training curve is not increasing in the end, it indicates that the XGBoost model is neither underfitted nor overfitted.

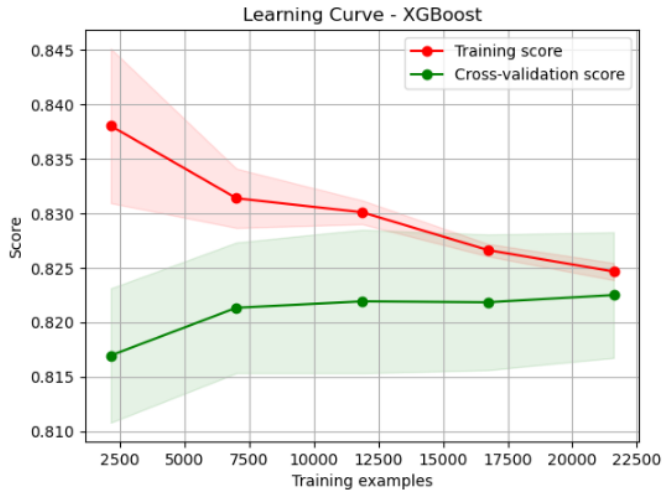


Figure 21: XGBoost learning curve

The learning curve of the XGBoost is a very good example of how a learning curve should look like, so that the model is neither over- nor underfitted. Both graphs are converting to each other and the gap between both is small.

Lastly, the computational time for the XGBoost model with the hyperparameter settings shown in section 4.3.10 is around 6 minutes. It is therefore even lower than the Logistic Regression with L1 regularization and as the Elastic Net regression. Nevertheless, in general the XGBoost is a very computationally intensive model and different hyperparameter can lead to a different result.

5.2.2 Random Forest

Once again, the performance of the Random Forest model surpasses that of the benchmark model by far. Again, especially the recall of the model is significantly better than for the Logistic Regression. The reason for that is, that the Random Forest model excels in capturing highly non-linear dataset, which is an advantage in this scenario. In comparison to the XGBoost, the difference in performance is small and may depend on subtle factors. This similarity is reflected in the confusion matrices of the two models, which show striking similarities (figure 20 and 22).

	Random Forest	Logistic Regression
F1-score	0.4589	0.3524
Accuracy	0.8188	0.8095
Precision	0.6624	0.6881
Recall	0.3511	0.2369

Table 6: Random Forest results

The Random Forest is showing significantly better results than the benchmark model, especially in terms of F1-score and recall. In contrast, the accuracy is only slightly better and the precision even worst.

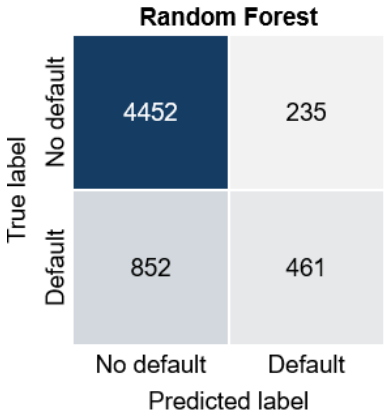


Figure 22: Random Forest confusion matrix

The confusion matrix of the Random Forest is quite similar to the XGBoost. It even shows a slightly higher number of correct classified defaults. In comparison to the benchmark, the difference is significantly higher, and the Random Forest is capable of classifying more defaults correctly.

Although the Random Forest model places less importance on "pay_1" than XGBoost, it still rands it as the modes important feature by a significant margin. Consequently, the model assigns greater weight to "pay_2" and "pay_3" and includes "pay" as only model in the top five features.

In contrast to the XGBoost model and the Logistic Regression models, the learning curve of the Random Forest model is not optimal. First, the scale of the y-axis is larger, indicating that the training and validation curve do not converge as closely together. This indicates that either the dataset is not ideal for the Random Forest or that there is slight overfitting.

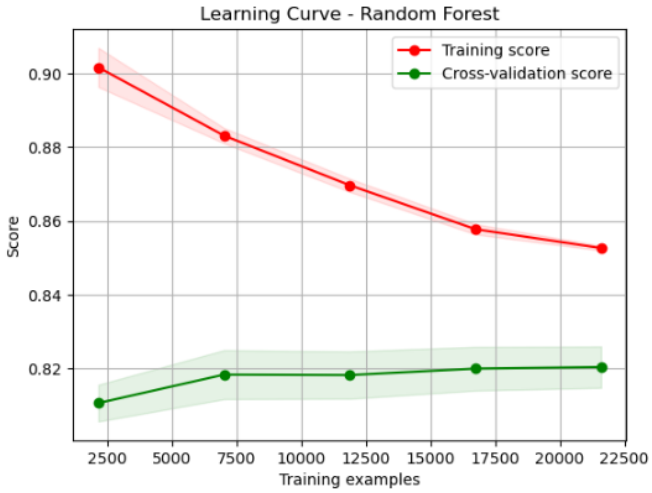


Figure 23: Random Forest learning curve

For the Random Forest, the difference between the training and the validation curve is significantly larger compared to the other models.

In terms of computational cost, fitting the Random Forest model requires approximately 20 minutes. This is over three times longer than the fitting time for the XGBoost model and significantly exceeds that of the benchmark model. The primary factor contributing to this discrepancy are the hyperparameter settings. The Random Forest model operates with wider parameter ranges compared to XGBoost. These expanded ranges are intended to offset the absence of a learning effect between each tree in the Random Forest model.

5.2.3 K-Nearest Neighbors

The KNN algorithm performs better than Logistic Regression but is significantly outperformed by decision tree-based models in terms of F1-score and recall. KNN is sensitive to scaling and imbalanced datasets. Nevertheless, these issues have been addressed during the data processing stage. Additionally, various k parameters were explored during hyperparameter tuning. A plausible explanation for KNN's lower performance is the high-dimensional nature of the dataset, which poses challenges for the algorithm.

	K-Nearest Neighbors	Logistic Regression
F1-score	0.3951	0.3524
Accuracy	0.8117	0.8095
Precision	0.6649	0.6881
Recall	0.281	0.2369

Table 7: KNN results

The KNN model outperforms the benchmark model in terms of F1-score and recall. Nevertheless, it shows a worse result when comparing to the other advanced machine learning models.

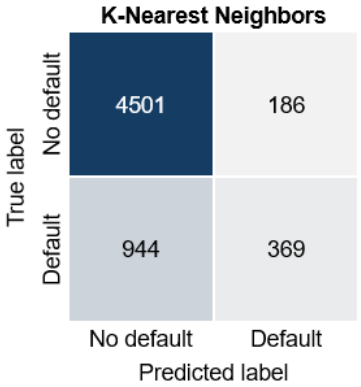


Figure 24: KNN confusion matrix

In the confusion matrix the difference between the other advanced machine learning models and the KNN model is visible. The KNN model is capturing less correct defaults.

When examining the importance of features, it is noticeable that the weighting of “pay_1” is significantly lower compared to other models. This discrepancy may contribute to the reduced performance of the model. However, the model compensates for this by giving greater importance to other payment statuses. In addition, "closeness" is one of the five most important features.

The learning curve confirms these observations and reflects a similar pattern to the performance metrics. Both the training and validation curves show minimal fluctuations, indicating possible inadequacies in capturing the inherent data patterns.

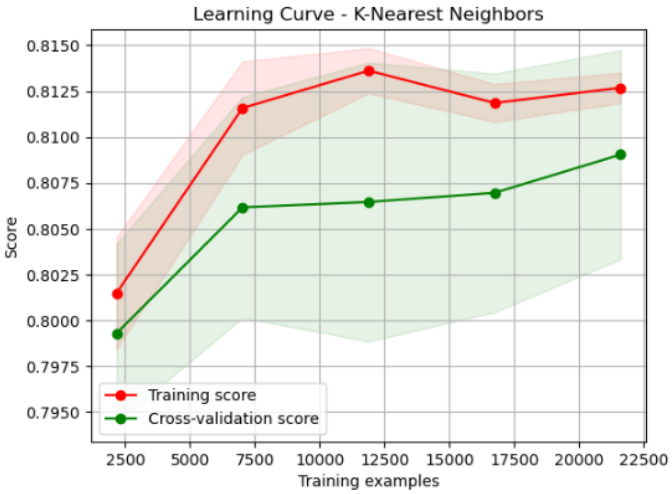


Figure 25: KNN learning curve

On the one hand, the validation score is showing a good curve in which it increases over time, which means that the performance of the model increases. On the other hand, the training curve is first increasing and then

decreasing, which is usually not the case. Nevertheless, both curves are close to each other and no underfitting or overfitting could be detected.

Also, in terms of costs, the KNN model performs on the lower end of this comparison. Consuming 15 minutes of computational time, it ranks as the third slowest model. Consequently, it neither receives an outstanding performance nor proves to be particularly cost-efficient.

5.2.4 Neural Network

The Neural Network shows significantly better results than Logistic Regression and is close on the decision tree models performance, although with slightly reduced performance. This discrepancy is primarily due to the lower ability to accurately classify default instances, as shown in figure 26.

	Neural Network	Logistic Regression
F1-score	0.4467	0.3524
Accuracy	0.8175	0.8095
Precision	0.6637	0.6881
Recall	0.3366	0.2369

Table 8: Neural Network results

The Neural Network model outperforms the benchmark model, especially in terms of recall. It shows a similar picture as the decision tree models but slightly worse.

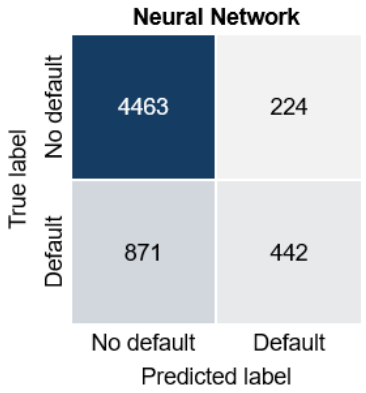


Figure 26: Neural Network confusion matrix

The matrix shows the difference to the benchmark model, in which less defaults are correctly identified.

In terms of feature importance, the importance of the newly introduced feature "closeness" stands out, remarkably ranking second in importance. In addition, the inclusion of a variety of features in the top five places highlights a nuanced representation that differs from other models examined.

When analyzing the validation curve, a slight deterioration over time can be observed, indicating a possible mismatch between the complexity of the model and the inherent structure of the dataset. However, the convergence of the validation and training curves indicates a favorable balance, suggesting neither overfitting nor underfitting take place and thus supporting the reliability of the model.

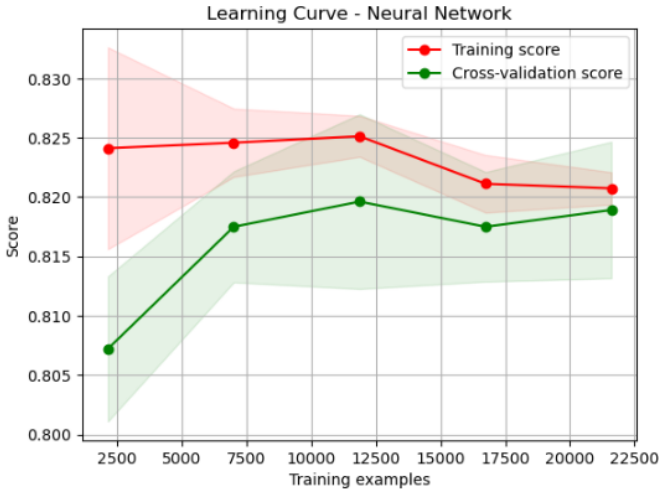


Figure 27: Neural Network learning curve

While the validation curve is increasing over time and only dips once, the training curve is not really moving. Nevertheless, the curves are almost touching each other at the end, which indicates a well trained model and no underfitting or overfitting takes place.

Contrary to initial expectations, the Neural Network used in this study has a relatively fast computation time. It takes only 9 minutes to compute, making it the second most efficient model, only behind XGBoost.

5.2.5 Support Vector Machine

The SVM performs well, but lags in detecting actual defaults, especially when compared to the decision tree models. Therefore, it can only detect 32% of all defaults which is 10ppt less than the Random Forest or the XGBoost.

	SVM	Logistic Regression
F1-score	0.4443	0.3524
Accuracy	0.8195	0.8095
Precision	0.6808	0.6881
Recall	0.3298	0.2369

Table 9: SVM results

The SVM model outperforms the benchmark model in all metrics despite the precision. Again, the main difference comes from the ability to better predict correct defaults, which is visible in recall.

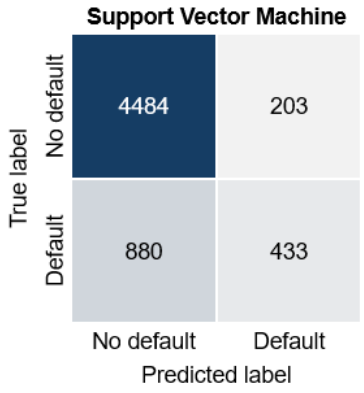


Figure 28: SVM confusion matrix

The confusion matrix of the Random Forest is quite like the Neural Network. Despite, it shows even a slightly lower number of correct classified defaults.

In the context of feature importance, it is noteworthy that the newly introduced feature "closeness" is not one of the five most important features of the SVM model, but only ranks seventh. The five most important features are made up exclusively of different payment statuses.

The analysis of the learning curves shows a deterioration in the score towards the end for both the validation curve and the training curve. This pattern indicates a possible problem of slight overfitting, which means that the model may be capturing noise in the training data rather than generalizable patterns.

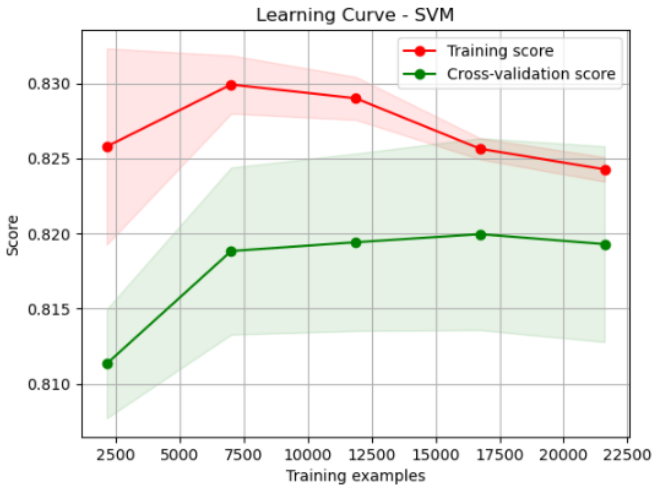


Figure 29: SVM learning curve

In comparison to the other models, the end of both curves are not extremely close. Additionally, the validation curve decreases at the last spot, which could indicate a small overfitting of the model.

The theory of overfitting is also supported by the computation time of the SVM model. With a computation time of 48 minutes, the SVM model is by far the slowest and consequently the least cost-efficient of the models evaluated.

5.2.6 Summary of advanced models

A comparison of the machine learning models used in this study shows that all advanced machine learning models outperform the benchmark Logistic Regression. However, the KNN model consistently lags behind the other advanced models. This difference is likely due to the fact that the dataset is not linear and other models can better capture its complexity.

When analyzing the feature importance in the different models, "pay_1" consistently emerges as the most important feature, with "closeness" almost always ranking in the top

five. This indicates that despite the differences in model architecture, the most important predictive features remain similar.

Despite the overall superiority of the advanced machine learning models, their performance does not quite meet expectations. The models detect less than 40% of all payment defaults, which is a somewhat disappointing result. However, the overall accuracy, which is consistently above 0.8, is a positive aspect and indicates in general a good predictive ability.

In terms of real-world applications, reducing the cost of credit card defaults by over 50% while increasing the cost of computing by a factor of six could be a worthwhile trade-off. However, this only works if the computational costs are proportionally much lower than the losses from defaults. This balance between cost savings and increased computational effort is a critical factor for credit card institutions considering the implementation of these advanced machine learning models. This implementation therefore has a direct impact on their financial statements and consequently on their profitability.

	Bench-mark	XGBoost	Random Forest	KNN	Neural Network	SVM
F1-score	0.3524	0.4599	0.4589	0.3951	0.4467	0.4443
Accuracy	0.8095	0.8203	0.8188	0.8117	0.8175	0.8195
Precision	0.6881	0.672	0.6624	0.6649	0.6637	0.6808
Recall	0.2369	0.3496	0.3511	0.281	0.3366	0.3298
Time [min]	1	6	20	15	9	48

Table 10: Comparison of all models

This comparison of all models shows clearly that all advanced machine learning models perform better than the benchmark model in terms of F1-score and recall. Nevertheless, the computational time also increases a lot, which illustrates the trade-off for a credit card institution.

5.3 Comparison to existing research

5.3.1 Machine learning in finance

Even in comparison to other research fields of finance, there are some interesting general findings that could be observed. One notable observation is that more complex models tend to perform better across various finance-related research fields. Bianchi et al. (2020), Gu et al. (2020) and Mirza et al. (2024) demonstrate that in general more complex machine learning models perform better than simpler models, supporting the findings in this thesis.

Additionally, the results in this study indicate that model performance depends on a few features. For instance, the payment status emerges as the most significant predictor of credit card defaults. This pattern is consistent with other research where the predictive power is concentrated in a few key features (Bali et al., 2023; Leippold et al., 2022; van Binsbergen et al., 2022).

Overall, there is a consensus in various areas of finance that advanced machine learning models can be very effective and profitable. They consistently outperform traditional statistical approaches (Benth et al., 2024; DeMiguel et al., 2023; Routledge, 2019).

When evaluating the specific performance metrics, the accuracy of predicting financial crises, is notably high at 0.98, which surpasses the accuracy levels achieved in this study (Samitas et al., 2020). Conversely, the accuracy of predicting option prices, which stands at 0.68 is lower than the accuracy observed in this study (Bali et al., 2023). This comparison highlights the variability in prediction performance across different financial tasks, indicating that credit card default prediction sits somewhere in the middle in terms of predictability.

5.3.2 Machine learning for credit card default risk

In line with the findings of this study, past research also highlights that decision tree-based models often achieve the best results (Alonso-Robisco & Carbó, 2022; Bačová & Babič, 2021; Y. Liu et al., 2022). Followed closely by Neural Network models, which in general also perform well (Chishti & Awan, 2019). This indicates the robust performance of complex machine learning models in this area. Interestingly, KNN models consistently underperform in credit card default prediction across various datasets. Y. Chen and Zhang (2021) finds similar results, suggesting that KNN may not be suitable for this specific predictive task, which aligns with the findings of this thesis.

Similar to this research, other studies also report accuracies of around 0.80 in their best models (Arora et al., 2022; Sayjadah et al., 2018). However, there are differences in the F1-scores. While this study achieves a F1-score of 0.46, some researchers have reported scores up to 0.9 (Subasi & Cankurt, 2019). These researchers attribute their higher scores to the use of the SMOTE function to address dataset imbalance. Given that this study also employs SMOTE, the reason for this difference can be questioned. Nevertheless, it suggests that with

different data, both the F1-score and recall can be improved, which is encouraging for credit card institutions seeking to enhance their predictive models.

The analysis of feature importance reveals varying results. Similar to this study Bačová and Babič (2021) find that only a few features significantly influence the predictive outcomes in their datasets. In contrast, other researchers observe a more even distribution of feature importance across their dataset (Chen & Zhang, 2021). This variation indicates that the relevance of specific features may differ depending on the dataset, but it also highlights the flexibility of machine learning models to adapt to different data characteristics.

In summary, the results of this study align well with other research in the field, suggesting that advanced machine learning models perform well in predicting credit card defaults. This consistency across different datasets and studies enhances the robustness of the findings. Additionally, Alonso-Robisco and Carbó (2022) demonstrate that the superior performance of machine learning predictions could potentially reduce regulatory capital requirements, further emphasizing the practical benefits of these models.

5.3.3 Machine learning model using the same data

Even when using the same dataset, comparing results across different studies can be challenging due to several factors. Feature engineering, for instance, is often considered as art, and each researcher approaches it differently. This study introduced a new feature, "closeness", which played an important role in every model. This feature is missing in other researchers' models, potentially affecting their results. Additionally, machine learning models have numerous settings, and even with hyperparameter optimization, differences in these settings can still have a significant impact on results. Many researchers do not disclose their final settings, making replication difficult. Furthermore, many researchers either do not address or do not mention how they handle the dataset imbalance. The use of the SMOTE function in this study is a critical step that may not be considered in other works.

Despite these differences, the findings of this study align with other research using similar datasets, particularly in identifying decision tree algorithms as the best-performing models. For example, Islam et al. (2018) combine the dataset with another dataset from Spain and achieve a F1-score of 0.9 for the extremely randomized tree, although they do not test XGBoost or Neural Networks. Their extremely randomized trees also achieve a recall of 0.86 and an accuracy of 0.96. On the other hand, the other models have F1-scores below 0.5,

recall below 0.4, and accuracy around 0.8, which is similar to the results of this research. Faraj et al., (2021) report an F1-score of 0.67 for the XGBoost model, with Neural Networks close behind at 0.65. Both models have a recall around 0.6, but their accuracy is significantly lower at 0.65, likely due to their method of handling data imbalance by down sampling, which reduces the dataset by over 50%.

Singh and Sivasankar (2019) find boosting trees to be the best performing model, additionally to Teng and Lee (2019), who report an accuracy of 0.8 for their XGBoost model. Mei (2016) results in having the Random Forests as best model and achieves an F1-score of 0.5, whereas Neema and Soibam (2017) report a similar accuracy of 0.82 of their Random Forests model. In contrast, R. Liu (2018) reaches the best F1-score of 0.44 with their Neural Network model.

In summary, despite the differences in methodologies and specific model settings, other researchers using the same or similar datasets find results that generally align with those of this study. Decision tree-based algorithms most often emerge as the top performers, followed by Neural Networks. However, the lack of detailed information about feature importance and model settings in other studies makes direct comparison and interpretation challenging. This highlights the importance of transparency and comprehensive reporting in machine learning research to enhance the reproducibility and value of findings. Additionally, no researcher writes about the costs associated with the models. However, as shown in this study, there are significant differences between each model, which are relevant for banks and credit card institutions.

6. Conclusion

The results of this study reveal that all advanced machine learning models outperform the benchmark Logistic Regression. However, the KNN model lags behind compared to the other models. Among the models evaluated, XGBoost emerges as the best performer in terms of metrics, model fit, and computational time. Despite this, its recall of 0.35 suggests that the performance may not be sufficient for a credit card institution, as 65% of defaults are not detected. However, other studies indicate that these results can be improved with different data, achieving up to 0.96 accuracy and detecting 86% of all defaults.

Addressing the research question, the findings indicate that advanced machine learning models do offer an improvement over traditional statistical methods. Nevertheless, as the increase of computational time is immense and the data is different for each credit card institution, no final evaluation can be done. In general, if the computational costs are lower in proportion to the costs of default, the implementation of advanced machine learning models can be favorable. Regarding the second question, the most important features identified are the payment status and the closeness to the credit limit. These are especially important for the credit card institutions as socio-demographic factors seem to play subordinate role.

In a broader context, this study aligns with existing research and demonstrates that advanced machine learning models can significantly improve predictive performance. Therefore, the results are consistent with findings from other areas in finance, showing that more complex models tend to perform better, and that the importance of few key features is a common theme. However, it is also highlighted that variability in results due to differences in data, feature engineering, and model settings is vast.

Nevertheless, the results do not come without limitations. They are highly dependent on the specific dataset and the feature engineering process implemented in this study. Additionally, the dataset covers only a short time frame of six month, which may be realistic for banks with many new clients but limits the generalizability of the findings. Therefore, each bank or credit card institution needs to conduct similar analyses with their own data available to obtain relevant insights. Thus, this research serves as a guideline for identifying potentially effective models rather than providing definitive solutions.

Looking ahead for future research, several studies can build on these findings. Applying the models to different datasets, including more recent data, could provide further validation of the results. Additionally, investigating the impact of different features on model performance would enhance understanding of the key factors for credit card defaults. Conducting long term studies to assess the models over longer time periods would help in evaluating their robustness and adaptability. Additionally, the evaluation of the cost-effectiveness, when implementing these models in real-world credit risk management systems, would be crucial for practical application in the financial industry. Therefore, cost forecasts for those implementation projects need to be made and put into comparison to the saving potential of reduced credit card defaults.

Bibliography

Achakzai, M. A. K., & Peng, J. (2023). Detecting financial statement fraud using dynamic ensemble machine learning. *International Review of Financial Analysis*, *89*, 102827. https://doi.org/10.1016/j.irfa.2023.102827

Alam, T. M., Shaukat, K., Hameed, I. A., Luo, S., Sarwar, M. U., Shabbir, S., Li, J., & Khushi, M. (2020). An Investigation of Credit Card Default Prediction in the Imbalanced Datasets. *IEEE Access*, *8*, 201173–201198. https://doi.org/10.1109/ACCESS.2020.3033784

Alonso-Robisco, A., & Carbó, J. M. (2022). Can machine learning models save capital for banks? Evidence from a Spanish credit portfolio. *International Review of Financial Analysis*, *84*, 102372. https://doi.org/10.1016/j.irfa.2022.102372

Arora, S., Bindra, S., Singh, S., & Nassa, V. K. (2022). Prediction of credit card defaults through data analysis and machine learning techniques. *Materials Today: Proceedings*, *51*, 110–117. https://doi.org/10.1016/j.matpr.2021.04.588

Audrino, F., Kostrov, A., & Ortega, J.-P. (2019). Predicting U.S. Bank Failures with MIDAS Logit Models. *The Journal of Financial and Quantitative Analysis*, *54*(6), 2575–2603.

Ausubel, L. M. (1997). Credit Card Defaults, Credit Card Profits, and Bankruptcy. *American Bankruptcy Law Journal*, 249–270.

Aziz, S., Dowling, M., Hammami, H., & Piepenbrink, A. (2024). *Machine learning in finance: A topic modeling approach*.

Bačová, A., & Babič, F. (2021). Predictive Analytics for Default of Credit Card Clients. *2021 IEEE 19th World Symposium on Applied Machine Intelligence and Informatics (SAMI)*, 000329–000334. https://doi.org/10.1109/SAMI50585.2021.9378671

Bali, T. G., Beckmeyer, H., Mörke, M., & Weigert, F. (2023). Option Return Predictability with Machine Learning and Big Data. *The Review of Financial Studies*, *36*(9), 3548–3602. https://doi.org/10.1093/rfs/hhad017

Barboza, F., Kimura, H., & Altman, E. (2017). Machine learning models and bankruptcy prediction. *Expert Syst. Appl.*, *83*, 405–417. https://doi.org/10.1016/j.eswa.2017.04.006

Benth, F. E., Detering, N., & Galimberti, L. (2024). Pricing options on flow forwards by neural networks in a Hilbert space. *Finance and Stochastics*, *28*(1), 81–121. https://doi.org/10.1007/s00780-023-00520-2

Berkson, J. (1944). Application of the Logistic Function to Bio-Assay. *Journal of the American Statistical Association*, *39*(227), 357–365.

Bianchi, D., Büchner, M., & Tamoni, A. (2020). Bond Risk Premiums with Machine Learning. *The Review of Financial Studies*, *34*(2), 1046–1089. https://doi.org/10.1093/rfs/hhaa062

Birjali, M., Kasri, M., & Beni-Hssane, A. (2021). A comprehensive survey on sentiment analysis: Approaches, challenges and trends. *Knowledge-Based Systems*, *226*, 107134. https://doi.org/10.1016/j.knosys.2021.107134

Brush, S. G. (1967). History of the Lenz-Ising Model. *Rev. Mod. Phys.*, *39*(4), 883–893. https://doi.org/10.1103/RevModPhys.39.883

Cao, K., & You, H. (2024). Fundamental Analysis via Machine Learning. *Financial Analysts Journal*, *0*(0), 1–25. https://doi.org/10.1080/0015198X.2024.2313692

Chang, C.-H. (2022). Information Asymmetry and Card Debt Crisis in Taiwan. *Bulletin of Applied Economics*, 123–145. https://doi.org/10.47260/bae/927

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, *16*, 321–357.

Chen, Y., & Zhang, R. (2021). Research on Credit Card Default Prediction Based on k-Means SMOTE and BP Neural Network. *Complexity*, *2021*, 6618841. https://doi.org/10.1155/2021/6618841

Chishti, W. A., & Awan, S. M. (2019). Deep Neural Network a Step by Step Approach to Classify Credit Card Default Customer. *2019 International Conference on Innovative Computing (ICIC)*, 1–8. https://doi.org/10.1109/ICIC48496.2019.8966723

Chou, M. (2006). Cash and credit card crisis in Taiwan. *Business Weekly*, 24–27.

DeMiguel, V., Gil-Bazo, J., Nogales, F. J., & Santos, A. A. P. (2023). Machine learning and fund characteristics help to select mutual funds with positive alpha. *Journal of Financial Economics*, *150*(3), 103737. https://doi.org/10.1016/j.jfineco.2023.103737

Dunn, L., & Kim, T. (1999). *An Empirical Investigation of Credit Card Default*.

Faraj, A. A., Mahmud, D. A., & Rashid, B. N. (2021). Comparison of Different Ensemble Methods in Credit Card Default Prediction. *UHD Journal of Science and Technology*, *5*(2), 20–25. https://doi.org/10.21928/uhdjst.v5n2y2021.pp20-25

Fix, E., & Hodges, J. L. (1951). *Discriminatory Analysis: Nonparametric Discrimination: Consistency Properties*. USAF School of Aviation Medicine. https://books.google.it/books?id=4XwytAEACAAJ

Fradkov, A. L. (2020). Early History of Machine Learning. *IFAC-PapersOnLine*, *53*(2), 1385–1390. https://doi.org/10.1016/j.ifacol.2020.12.1888

Goldberg, Y. (2015). A Primer on Neural Network Models for Natural Language Processing. *ArXiv*, *abs/1510.00726*. https://doi.org/10.1613/jair.4992

González, S., García, S., Ser, J., Rokach, L., & Herrera, F. (2020). A practical tutorial on bagging and boosting based ensembles for machine learning: Algorithms, software tools, performance study, practical perspectives and opportunities. *Inf. Fusion*, *64*, 205–237. https://doi.org/10.1016/j.inffus.2020.07.007

Gu, S., Kelly, B., & Xiu, D. (2020). Empirical Asset Pricing via Machine Learning. *The Review of Financial Studies*, *33*(5), 2223–2273. https://doi.org/10.1093/rfs/hhaa009

Gunnarsson, E. S., Isern, H. R., Kaloudis, A., Risstad, M., Vigdel, B., & Westgaard, S. (2024). Prediction of realized volatility and implied volatility indices using AI and machine learning: A review. *International Review of Financial Analysis*, *93*, 103221. https://doi.org/10.1016/j.irfa.2024.103221

Hoerl, A. E., & Kennard, R. W. (1970). Ridge Regression: Biased Estimation for Nonorthogonal Problems. *Technometrics*, *12*(1), 55–67.

Islam, S. R., Eberle, W., & Ghafoor, S. K. (2018). *Credit Default Mining Using Combined Machine Learning and Heuristic Approach*.

Jaffee, D. M., & Russell, T. (1976). Imperfect Information, Uncertainty, and Credit Rationing*. *The Quarterly Journal of Economics*, *90*(4), 651–666. https://doi.org/10.2307/1885327

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2021). *An Introduction to Statistical Learning: With Applications in R*. Springer US. https://doi.org/10.1007/978-1-0716-1418-1

James, R., Leung, H., & Prokhorov, A. (2023). A machine learning attack on illegal trading. *Journal of Banking & Finance*, *148*, 106735. https://doi.org/10.1016/j.jbankfin.2022.106735

Johnson, H. (2024, January 8). History of Credit Cards: A Brief Overview. *TIME*. https://time.com/personal-finance/article/history-of-credit-cards/

Kaniel, R., Lin, Z., Pelger, M., & Nieuwerburgh, S. V. (2023). Machine-learning the skill of mutual fund managers. *Journal of Financial Economics*, *150*(1), 94–138. https://doi.org/10.1016/j.jfineco.2023.07.004

Kumari, K., & Yadav, S. (2018). Linear regression analysis study. *Journal of the Practice of Cardiovascular Sciences*, *4*, 33–36. https://doi.org/10.4103/JPCS.JPCS_8_18

Landwehr, N., Hall, M., & Frank, E. (2003). Logistic Model Trees. *Machine Learning*, *59*, 161–205. https://doi.org/10.1007/s10994-005-0466-3

Lawi, A., & Aziz, F. (2018). Classification of Credit Card Default Clients Using LS-SVM Ensemble. *2018 Third International Conference on Informatics and Computing (ICIC)*, 1–4. https://doi.org/10.1109/IAC.2018.8780427

Leippold, M., Wang, Q., & Zhou, W. (2022). Machine learning in the Chinese stock market. *Journal of Financial Economics*, *145*(2, Part A), 64–82. https://doi.org/10.1016/j.jfineco.2021.08.017

Lenert, M. C., Mize, D. E., & Walsh, C. G. (2018). X Marks the Spot: Mapping Similarity Between Clinical Trial Cohorts and US Counties. *AMIA Annu Symp Proc*, 1110–1119.

Libbrecht, M. W., & Noble, W. S. (2015). Machine learning applications in genetics and genomics. *Nature Reviews Genetics*, *16*, 321–332. https://doi.org/10.1038/nrg3920

Likas, A., Vlassis, N., & Verbeek, J. (2003). The global k-means clustering algorithm. *Pattern Recognit.*, *36*, 451–461. https://doi.org/10.1016/S0031-3203(02)00060-2

Liu, R. (2018). Machine Learning Approaches to Predict Default of Credit Card Clients. *Modern Economy*, *09*(11), 1828–1838. https://doi.org/10.4236/me.2018.911115

Liu, Y., Yang, M., Wang, Y., Li, Y., Xiong, T., & Li, A. (2022). Applying machine learning algorithms to predict default probability in the online credit market: Evidence from China. *International Review of Financial Analysis*, *79*, 101971. https://doi.org/10.1016/j.irfa.2021.101971

Mammone, A., Turchi, M., & Cristianini, N. (2008). Support vector machines. *Wiley Interdisciplinary Reviews: Computational Statistics*. https://doi.org/10.1002/WICS.49

Manjaly, J., Varghese, R., & Varughese, P. (2021). Artificial Intelligence in the Banking Sector – A Critical Analysis. *Management Science*, *8*, 210–216. https://doi.org/10.34293/MANAGEMENT.V8IS1-FEB.3778

Marcot, B., & Hanea, A. (2020). What is an optimal value of k in k-fold cross-validation in discrete Bayesian network analysis? *Computational Statistics*, 1–23. https://doi.org/10.1007/s00180-020-00999-9

Maxwell, A., Warner, T., & Fang, F. (2018). Implementation of machine-learning classification in remote sensing: An applied review. *International Journal of Remote Sensing*, *39*, 2784–2817. https://doi.org/10.1080/01431161.2018.1433343

Mei, R. (2016). Study on Analysis and Influence Factors of Credit Card Default Prediction Model. *Statistics and Application*, *05*, 263–275. https://doi.org/10.12677/SA.2016.53026

Mirza, N., Rizvi, S. K. A., Naqvi, B., & Umar, M. (2024). Inflation prediction in emerging economies: Machine learning and FX reserves integration for enhanced forecasting. *International Review of Financial Analysis*, 103238. https://doi.org/10.1016/j.irfa.2024.103238

Mitchell, T. M. (1997). *Machine Learning*. McGraw-Hill.

Moula, F. E., Guotai, C., & Abedin, M. Z. (2017). Credit default prediction modeling: An application of support vector machine. *Risk Management*, *19*(2), 158–187. https://doi.org/10.1057/s41283-017-0016-x

Muggleton, S., Schmid, U., Zeller, C., Tamaddoni-Nezhad, A., & Besold, T. R. (2018). Ultra-Strong Machine Learning: Comprehensibility of programs learned with ILP. *Machine Learning*, *107*, 1119–1140. https://doi.org/10.1007/s10994-018-5707-3

Neema, S., & Soibam, B. (2017). *The comparison of machine learning methods to achieve most cost-effective prediction for credit card default*. https://api.semanticscholar.org/CorpusID:199425780

Niu, X., & Suen, C. (2012). A novel hybrid CNN-SVM classifier for recognizing handwritten digits. *Pattern Recognit.*, *45*, 1318–1325. https://doi.org/10.1016/j.patcog.2011.09.021

Niu, Z., Wang, C., & Zhang, H. (2023). Forecasting stock market volatility with various geopolitical risks categories: New evidence from machine learning models. *International Review of Financial Analysis*, *89*, 102738. https://doi.org/10.1016/j.irfa.2023.102738

Noriega, J. P., Rivera, L. A., & Herrera, J. A. (2023). Machine Learning for Credit Risk Prediction: A Systematic Literature Review. *Data*. https://doi.org/10.3390/data8110169

Obaid, K., & Pukthuanthong, K. (2022). A picture is worth a thousand words: Measuring investor sentiment by combining machine learning and photos from news. *Journal of Financial Economics*, *144*(1), 273–297. https://doi.org/10.1016/j.jfineco.2021.06.002

Orhan, A. E. (2019). *Robustness properties of Facebook's ResNeXt WSL models*. https://doi.org/10.48550/arXiv.1907.07640

Quinlan, J. R. (1986). Induction of Decision Trees. *Machine Learning*, *1*, 81–106. https://doi.org/10.1007/BF00116251

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., & others. (2019). Language models are unsupervised multitask learners. *OpenAI Blog*, *1*(8), 9.

Ray, R. (2012). Managing Financial Risk Via Prediction Markets. *The Journal of Investing*, *21*, 76–80. https://doi.org/10.3905/joi.2012.21.2.076

Routledge, B. R. (2019). Machine learning and asset allocation. *Financial Management*, *48*(4), 1069–1094. https://doi.org/10.1111/fima.12303

Samitas, A., Kampouris, E., & Kenourgios, D. (2020). Machine learning as an early warning system to predict financial crisis. *International Review of Financial Analysis*, *71*, 101507. https://doi.org/10.1016/j.irfa.2020.101507

Samuel, A. L. (1959). Some Studies in Machine Learning Using the Game of Checkers. *IBM Journal of Research and Development*, *3*(3), 210–229. https://doi.org/10.1147/rd.33.0210

Sayjadah, Y., Hashem, I. A. T., Alotaibi, F., & Kasmiran, K. A. (2018). Credit Card Default Prediction using Machine Learning Techniques. *2018 Fourth International Conference on Advances in Computing, Communication & Automation (ICACCA)*, 1–4. https://doi.org/10.1109/ICACCAF.2018.8776802

Sermpinis, G., Tsoukas, S., & Zhang, Y. (2023). Modelling failure rates with machine-learning models: Evidence from a panel of UK firms. *European Financial Management*, *29*(3), 734–763. https://doi.org/10.1111/eufm.12369

Singh, B. E. R., & Sivasankar, E. (2019). Enhancing Prediction Accuracy of Default of Credit Using Ensemble Techniques. In R. S. Bapi, K. S. Rao, & M. V. N. K. Prasad

(Eds.), *First International Conference on Artificial Intelligence and Cognitive Computing* (pp. 427–436). Springer Singapore.

Soui, M., Smiti, S., Bribech, S., & Gasmi, I. (2018). Credit Card Default Prediction as a Classification Problem. In M. Mouhoub, S. Sadaoui, O. Ait Mohamed, & M. Ali (Eds.), *Recent Trends and Future Technology in Applied Intelligence* (pp. 88–100). Springer International Publishing.

Subasi, A., & Cankurt, S. (2019). Prediction of default payment of credit card clients using Data Mining Techniques. *2019 International Engineering Conference (IEC)*, 115–120. https://doi.org/10.1109/IEC47844.2019.8950597

Teng, H.-W., & Lee, M. (2019). Estimation Procedures of Using Five Alternative Machine Learning Methods for Predicting Credit Card Default. *Review of Pacific Basin Financial Markets and Policies*, *22*(03), 1950021. https://doi.org/10.1142/S0219091519500218

Tibshirani, R. (1996). Regression Shrinkage and Selection via the Lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, *58*(1), 267–288.

van Binsbergen, J. H., Han, X., & Lopez-Lira, A. (2022). Man versus Machine Learning: The Term Structure of Earnings Expectations and Conditional Biases. *The Review of Financial Studies*, *36*(6), 2361–2396. https://doi.org/10.1093/rfs/hhac085

Wang, G.-J., Chen, Y., Zhu, Y., & Xie, C. (2024). Systemic risk prediction using machine learning: Does network connectedness help prediction? *International Review of Financial Analysis*, *93*, 103147. https://doi.org/10.1016/j.irfa.2024.103147

Wang, Y., Andreeva, G., & Martin-Barragan, B. (2023). Machine learning approaches to forecasting cryptocurrency volatility: Considering internal and external determinants. *International Review of Financial Analysis*, *90*, 102914. https://doi.org/10.1016/j.irfa.2023.102914

Yacouby, R., & Axman, D. (2020). Probabilistic Extension of Precision, Recall, and F1 Score for More Thorough Evaluation of Classification Models. *Proceedings of the First Workshop on Evaluation and Comparison of NLP Systems*. https://doi.org/10.18653/v1/2020.eval4nlp-1.9

Yeh, I.-C., & Lien, C. (2009). The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients. *Expert Systems with Applications*, *36*(2, Part 1), 2473–2480. https://doi.org/10.1016/j.eswa.2007.12.020

Yu, Y. (2020). The Application of Machine Learning Algorithms in Credit Card Default Prediction. *2020 International Conference on Computing and Data Science (CDS)*, 212–218. https://doi.org/10.1109/CDS49703.2020.00050

Zhou, L., Pan, S., Wang, J., & Vasilakos, A. (2017). Machine learning on big data: Opportunities and challenges. *Neurocomputing*, *237*, 350–361. https://doi.org/10.1016/j.neucom.2017.01.026

Zou, H., & Hastie, T. (2005). Regularization and Variable Selection Via the Elastic Net. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, *67*(2), 301–320. https://doi.org/10.1111/j.1467-9868.2005.00503.x

Appendix

Code

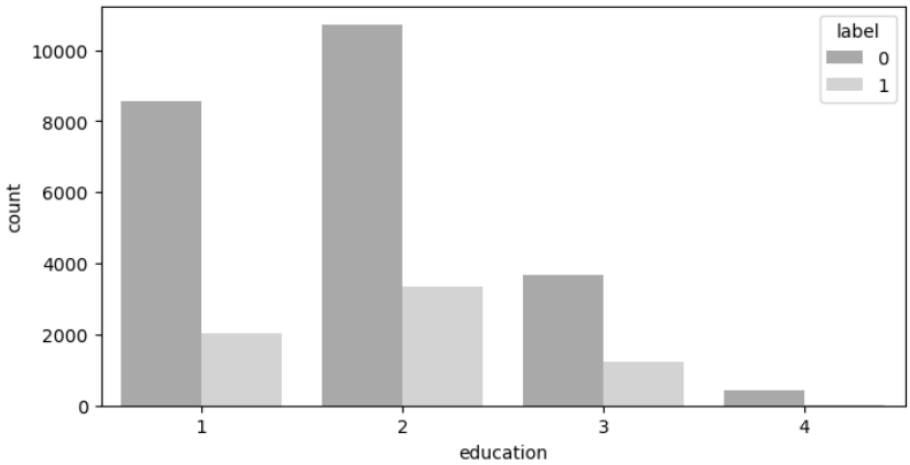
https://colab.research.google.com/drive/1KZ0K8b83hO-qwm2UTQ8AmnZTGiRIwUsJ?usp=sharing

Data

https://archive.ics.uci.edu/dataset/350/default+of+credit+card+clients

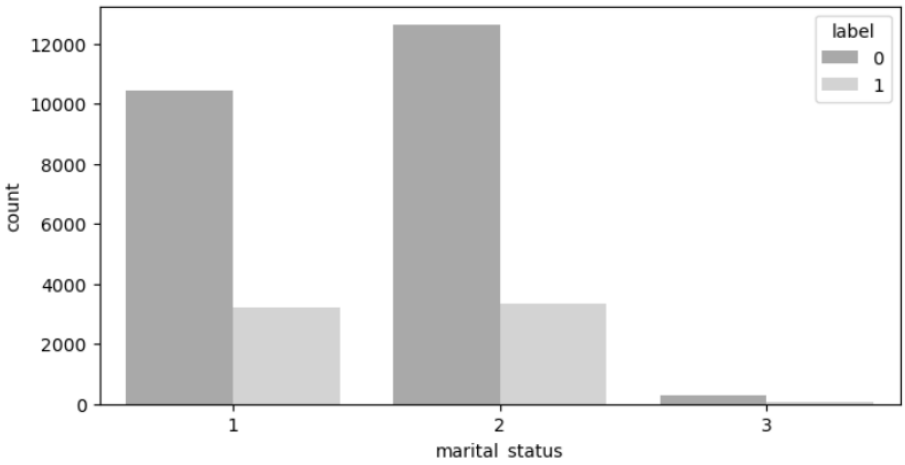
Descriptive statistics

Distribution of the education level in the data



(1 = graduate school, 2 = university, 3 = high school, 4 = others)

Distribution of the marital status in the data



(1 = married, 2 = single, 3 = others)

Correlation matrix

	credit_ amount	gender	educa- tion	marital_ status	age	pay_1	pay_2	pay_3	pay_4	pay_5	pay_6	bill_amt _1	bill_amt _2	bill_amt _3	bill_amt _4	bill_amt _5	bill_amt _6	pay_ amt_1	pay_ amt_2	pay_ amt_3	pay_ amt_4	pay_ amt_5	pay_ amt_6	label
credit_amount	1																							
gender	0.025	1																						
education	-0.231	0.014	1																					
marital_status	-0.111	-0.029	-0.137	1																				
age	0.145	-0.091	0.182	-0.412	1																			
pay_1	-0.271	-0.058	0.113	0.019	-0.039	1																		
pay_2	-0.296	-0.071	0.13	0.024	-0.05	0.672	1																	
pay_3	-0.286	-0.066	0.122	0.032	-0.053	0.574	0.767	1																
pay_4	-0.267	-0.06	0.117	0.032	-0.05	0.539	0.662	0.777	1															
pay_5	-0.249	-0.055	0.104	0.034	-0.054	0.509	0.623	0.687	0.82	1														
pay_6	-0.235	-0.044	0.089	0.033	-0.049	0.475	0.576	0.633	0.716	0.817	1													
bill_amt_1	0.285	-0.034	0.017	-0.028	0.056	0.187	0.235	0.208	0.203	0.207	0.207	1												
bill_amt_2	0.278	-0.031	0.012	-0.025	0.054	0.19	0.235	0.237	0.226	0.227	0.227	0.951	1											
bill_amt_3	0.283	-0.025	0.007	-0.029	0.054	0.18	0.224	0.227	0.245	0.243	0.241	0.892	0.928	1										
bill_amt_4	0.294	-0.022	-0.006	-0.027	0.051	0.179	0.222	0.227	0.246	0.272	0.266	0.86	0.892	0.924	1									
bill_amt_5	0.296	-0.017	-0.012	-0.029	0.049	0.181	0.221	0.225	0.243	0.27	0.291	0.83	0.86	0.884	0.94	1								
bill_amt_6	0.29	-0.017	-0.013	-0.025	0.048	0.177	0.219	0.222	0.239	0.263	0.285	0.803	0.832	0.853	0.901	0.946	1							
pay_amt_1	0.195	0	-0.041	-0.005	0.026	-0.079	-0.081	0.001	-0.009	-0.006	-0.001	0.14	0.28	0.244	0.233	0.217	0.2	1						
pay_amt_2	0.178	-0.001	-0.033	-0.01	0.022	-0.07	-0.059	-0.067	-0.002	-0.003	-0.005	0.099	0.101	0.317	0.208	0.181	0.173	0.286	1					
pay_amt_3	0.21	-0.009	-0.044	-0.004	0.029	-0.071	-0.056	-0.053	-0.069	0.009	0.006	0.157	0.151	0.13	0.3	0.252	0.234	0.252	0.245	1				
pay_amt_4	0.203	-0.002	-0.041	-0.014	0.021	-0.064	-0.047	-0.046	-0.043	-0.058	0.019	0.158	0.147	0.143	0.13	0.293	0.25	0.2	0.18	0.216	1			
pay_amt_5	0.217	-0.002	-0.045	-0.003	0.023	-0.058	-0.037	-0.036	-0.034	-0.033	-0.046	0.167	0.158	0.18	0.16	0.142	0.308	0.148	0.181	0.159	0.152	1		
pay_amt_6	0.22	-0.003	-0.044	-0.008	0.019	-0.059	-0.037	-0.036	-0.027	-0.023	-0.025	0.179	0.174	0.182	0.178	0.164	0.115	0.186	0.158	0.163	0.158	0.155	1	
label	-0.154	-0.04	0.034	-0.028	0.014	0.325	0.264	0.235	0.217	0.204	0.187	-0.02	-0.014	-0.014	-0.01	-0.007	-0.005	-0.073	-0.059	-0.056	-0.057	-0.055	-0.053	1

Results

Feature importance

	Logistic Regression	Lasso	Ridge	Elastic Net	XGBoost	Random Forest	K-Nearest Neighbors	Neural Network	SVM
credit_amount	0.00367	0.00362	0.00289	0.00278	0.00031	0.00375	0.00076	0.00617	0.00077
gender	-0.00041	-0.00019	-0.00015	-0.00001	0.00013	0.001	0.00101	0.00013	0.00036
education	0.0006	0.00057	0.00073	0.00038	0.00126	0.00173	0.00042	0.00173	0.00067
marital_status	0.00011	-0.00041	0.00009	0.00066	0.00036	0.00129	0.00059	0.0015	0.00033
age	-0.00045	-0.00042	-0.00007	-0.00038	0.00015	0.00239	0.00085	0.00043	0.00038
pay_1	0.03994	0.03981	0.03971	0.03969	0.07429	0.05862	0.01874	0.07428	0.06627
pay_2	0.00569	0.00641	0.00619	0.00607	0.00152	0.00819	0.00528	0.00561	0.00459
pay_3	0.00139	0.00223	0.00155	0.00189	0.00052	0.00619	0.00203	0.00171	0.00375
pay_4	0.00089	0.00139	0.00101	0.0013	0.00005	0.00399	0.00111	0.00035	0.00154
pay_5	0.00027	0.00041	0.00025	0.00056	0.00009	0.00387	0.0019	-0.00018	0.00187
pay_6	0.00005	0.00008	0.00005	0.00008	0.00045	0.00339	0.0011	0.00267	0.00175
bill_amt_1	0.00013	0	0.00003	-0.00019	0.00065	0.00292	-0.00014	0.00111	0.00096
bill_amt_2	-0.00048	0	-0.0007	0.00027	0.00005	0.00252	-0.00001	0.00043	0.00091
bill_amt_3	0.00351	0.00078	0.00356	0.00097	0.00011	0.00265	0.00006	0.00153	0.00098
bill_amt_4	-0.00074	0	-0.00094	-0.00013	-0.00003	0.00215	0.00005	0.00044	0.00072
bill_amt_5	0.00129	0	0.00137	0.00013	0.00006	0.00245	0.00024	0.00605	0.00091
bill_amt_6	-0.00143	-0.00097	-0.00139	-0.00115	0.00005	0.00224	0.00031	0.00197	0.00075
pay_amt_1	0.00093	0.00091	0.00059	0.00117	0.00023	0.00569	0.00023	0.00068	0.00041
pay_amt_2	0.00047	0.00054	0.00023	0.00043	0.00043	0.00434	0.00015	0.00157	0.00035
pay_amt_3	0.00018	0.00025	0.00011	0.00017	0.00013	0.00383	0.00037	0.00016	0.00034
pay_amt_4	-0.00003	-0.00001	-0.00001	-0.0002	0.00015	0.00351	0.00018	0.00003	0.00024
pay_amt_5	0.00003	0.00001	-0.00004	-0.00004	-0.00004	0.00337	0.00014	0.00099	0.00045
pay_amt_6	0.00023	0.00038	0.00026	0.00021	0.00006	0.00355	0.00031	0.00027	0.0008
Closeness_6	0.00558	0.00467	0.00531	0.00453	0.00041	0.00393	0.0007	0.00088	0.00055
Closeness_5	-0.00232	-0.00081	-0.00236	-0.00095	-0.00001	0.00279	0.00067	0.00032	0.0002
Closeness_4	0.00113	0	0.00121	0.00001	-0.00004	0.00311	0.00099	0.00309	0.00056
Closeness_3	-0.00225	-0.0004	-0.00185	-0.00039	0.00023	0.00459	0.00068	0.00023	0.00029
Closeness_2	0.00582	0.00432	0.00565	0.0035	0.00021	0.0039	0.00072	0.00736	0.00063
Closeness_1	0.00124	0.00034	0.00082	0.00032	0.00063	0.00409	0.00125	0.00471	0.00127

Metrics

	Benchmark	Lasso	Ridge	Elastic Net	XGBoost	Random Forest	KNN	Nerual Network	SVM
F1_score	0.3524	0.3526	0.3515	0.3521	0.4599	0.4589	0.3951	0.4466	0.4443
Accuracy	0.8095	0.8097	0.8093	0.8098	0.8203	0.8188	0.8117	0.8195	0.8195
Precision	0.6881	0.6896	0.6874	0.692	0.672	0.6624	0.6649	0.6786	0.6808
Recall	0.2369	0.2369	0.2361	0.2361	0.3496	0.3511	0.281	0.3328	0.3298
Time [min]	1	10	1	8	6	20	15	9	48