



INSTITUTO  
UNIVERSITÁRIO  
DE LISBOA

---

Credit Scoring: A comparison between statistical and machine learning techniques for probability of default estimation

João D. Sousa Dias

Master in Monetary and Financial Economics

Supervisor:

Prof. Paulo Viegas de Carvalho, Invited Assistant Professor,  
ISCTE Business School

September, 2024

Department of Political Economy

Credit Scoring: A comparison between statistical and machine learning techniques for probability of default estimation

João D. Sousa Dias

Master in Monetary and Financial Economics

Supervisor:

Prof. Paulo Viegas de Carvalho, Invited Assistant Professor,  
ISCTE Business School

September, 2024

## **Acknowledgements**

This work would not have been possible without the help, support and guidance of different people. I would like to thank them for their impact on my life and subsequently, on this work.

First of all, I would like to express my gratitude to Prof. Paulo Viegas de Carvalho. His expertise, attention and advice were crucial for me, helping me surpassing many roadblocks found through all stages of this work.

To my whole family, more particularly to my grandfather José António for being an academia figure for me. His constant motivation and support helped me becoming a learning enthusiast, trait that I always find to be useful.

To Diana, mainly for her love and patience, but also for showing me another perspective on different things in life. Watching different plays and going into museums loosened my mind when this work most needed.

To all my friends, that along this journey were always there for me to help me distracting and laughing, the latter being the thing that I most enjoy.

To ISCTE, for receiving me as their student, helping me embracing my goals and providing me the tools to learn, this being the second thing most enjoyable in my life.

Once again, from the bottom of my heart, thank you all!



## **Resumo**

Risco de crédito é definido pelo Comité de Basileia como a probabilidade de um devedor entrar em incumprimento para com as suas obrigações creditícias, sendo que é necessária uma gestão efetiva do mesmo para otimizar rendibilidades ajustadas ao risco. Esta dissertação pretende ser um estudo sobre um conjunto de dados de empréstimos concedidos, publicamente disponível no repositório de Machine Learning da Universidade da Califórnia, Irvine (UCI), onde uma comparação é efetuada entre modelos estatísticos e modelos baseados em machine learning. Esta análise comparativa evidencia os vários pontos fortes e limitações respetivos a cada tipo de modelo, pelo aprofundamento das suas características e resultados na estimação da probabilidade de incumprimento. As conclusões apontam para a importância de um tratamento de dados robusto, da seleção do melhor modelo e na utilização de técnicas de interpretabilidade, destacando a complexidade dos vários fatores que influenciam o risco de crédito.



## **Abstract**

Credit risk, defined by the Basel Committee as the potential for a borrower to default on obligations, necessitates effective management to optimize risk-adjusted returns. This work intends to be a study on a publicly available loan default dataset from the University of California, Irvine (UCI) Machine Learning Repository, where a comparison is conducted between statistical and machine learning models. The comparative analysis of these models highlights their strengths and limitations, offering insights into their application in credit risk assessment. The findings underscore the importance of robust data preprocessing, model selection, and interpretability techniques in predicting credit defaults, highlighting the complex interplay of various factors influencing credit risk.





# INDEX

Acknowledgements	i
Resumo	iii
Abstract	v
Chapter 1: Introduction	13
Chapter 2: Literature review	17
Chapter 3: Methodology	21
3.1 Statistical Models	21
3.1.1 Logistic Regression	21
3.2 Machine Learning Models	22
3.2.1 Decision Trees	22
3.2.2 Random Forests	23
3.2.3 Artificial Neural Networks	24
3.2.4 Support Vector Machines	27
3.2.5 XGBoost	29
3.2.6 LightGBM	30
3.2.7 AdaBoost	32
3.3 Interpretability techniques	33
3.3.1 Shapley Additive Explanations	33
3.3.2 LIME	34
3.4 Performance metrics	35
Chapter 4: Exploratory Data Analysis	37
4.1 Dataset details and statistics	37
4.2 Data preprocessing	40
4.3 Modelling	43
Chapter 5: Results and explanations	47
5.1 Shapley values	51
5.2 LIME	52
Chapter 6: Conclusions and recommendations for future work	55
References	56
Annex	60



## List of tables

Table 1: Summary of recent applications regarding credit risk modelling.	19
Table 2: Confusion matrix.	35
Table 3: Performance measures.	36
Table 4: Categorical variables mapping.	41
Table 5: Recursive Feature Elimination results.	42
Table 6: ANOVA test results.	42
Table 7: Different prepared datasets.	43
Table 8: Time spent on each model's training.	44
Table 9: Normal dataset grid search results for each model.	44
Table 10: Performance metrics for ANOVA dataset.	47
Table 11: DeLong et al. (1988) test to compare AUCs.	49
Table 12: DeLong et al. (1988) test to compare AUCs.	50



## List of figures

Figure 1: Number of studies on Machine Learning for Credit Scoring.	18
Figure 2: Sigmoid function.	22
Figure 3: An example of a decision tree based on credit data.	23
Figure 4: Bagging concept visual example.	24
Figure 5: Neuron signal representation.	25
Figure 6: One-layer artificial neural network representation.	25
Figure 7: Gradient descent visual representation.	27
Figure 8: Support Vector Machine visual representation.	28
Figure 9: Gradient one-side sampling example.	31
Figure 10: EFB example.	32
Figure 11: AdaBoost algorithm visual representation.	32
Figure 12: Age variable distribution by class.	37
Figure 13: Credit Amount variable distribution by class.	38
Figure 14: Correlations between different variables.	39
Figure 15: Gender distribution by class.	39
Figure 16: Education distribution by class.	40
Figure 17: Marital Status distribution by class.	40
Figure 18: Data pipeline.	43
Figure 19: Cross-validation method.	45
Figure 20: ROC curves for ANOVA dataset training phase.	48
Figure 21: ROC curves for ANOVA dataset testing phase.	49
Figure 22: LightGBM-ANOVA confusion matrix on test data.	50
Figure 23: LightGBM-ANOVA accuracy for different thresholds.	51
Figure 24: SHAP explanation for a non-default prediction.	51
Figure 25: SHAP explanation for a default prediction.	52
Figure 26: LIME explanation for a default prediction.	52
Figure 27: LIME explanation for a non-default prediction.	53



## CHAPTER 1

# Introduction

The financial crises that occurred in the first two decades of the twenty-first century led financial institutions worldwide to pay increased attention to risk management, particularly credit risk. High non-performing loan (NPL) ratios that currently exist in a significant number of Member States of the European Union (EU) (European Council, 2017) may change market perceptions of the European banking industry as a whole and increase the risk of cross-border spillovers into the EU's economy and financial system.

According to the Basel Committee, credit risk can be defined as “the potential that a bank borrower or counterparty will fail to meet its obligations in accordance with agreed terms” (Basel Committee on Banking Supervision, 2000, p. 1). For financial institutions to optimize their risk-adjusted rate of return and keep their risk exposure within reasonable bounds, they must effectively manage the credit risk in their banking book. Therefore, the credit default likelihood of borrowers needs to be determined or estimated through the evaluation and analysis of loan applications. Determining what constitutes a “good” loan and what characteristics set it apart from a bad loan is key (Basel Committee on Banking Supervision, 2000).

An approach to estimating a debtor’s likelihood of credit default events lies in credit scoring. The use of this technique was intensified and stimulated by the widespread use of credit cards, as client databases grew significantly due to economic pressures brought on by a rise in loan demand, increased commercial competition, and the progress of new computer technologies (Hand et al., 1997). In this context, financial institutions began to integrate or replace subjective judgment-based credit-granting decisions with statistical models (Goh et al. 2019).

The Basel II accord was published in 2004. Under this accord, a bank may use the Internal ratings-based (IRB) approach after successful validation by the supervisor (Article 143 of Regulation (EU) No 575/2013. This is one approach to calculating the minimum capital requirement for credit risk, rather than using a more rigid set of rules as laid out in the Standardized approach. This marked a profound change in the credit scoring industry, with efforts to create complex models being deeply studied. Benefitting from the swift advancement of computer technology, creating more complex and sophisticated models becomes viable (Goh et al. 2019).

During the first half of the twentieth century, before the development of credit scoring models, credit decisions relied solely on the personal judgment of credit analysts. With the pioneering work of Altman (1968), resorting to multiple discriminant analysis (MDA), an effective determination of credit scoring to forecast business bankruptcy was achieved. With MDA, a set of financial ratios is modelled as input variables, with the resulting output (z-score) seeming like a reliable predictor and helpful for analysts

when making credit investment advice. Ohlson (1980) later presented a probabilistic method for estimating a company's creditworthiness, using an alternative approach: logistic regression.

The credit risk literature has studied two types of automatic credit scoring approaches: statistical techniques and artificial intelligence (AI) (Khatir et al., 2022). The industry standard statistical techniques used to produce scorecards are mostly composed of logistic regression, discriminant analysis, and linear regression (Hand et al., 1997).

More recently, several studies have shown that machine learning techniques—including support vector machines, decision trees, ensemble models, random forests, and artificial neural networks—are useful tools for assessing credit risk (see, e.g., Bensic, 2005; Brown, 2012; Huang, 2007; Gedela et al., 2022; and Ponsam et al., 2021). Machine Learning techniques, in contrast to statistical approaches, automatically extract information from training samples. According to earlier research, AI frequently performs better than statistical techniques when handling credit scoring problems, particularly in the face of nonlinear classification patterns (Khatir et al., 2022).

The application of machine learning algorithms in credit scoring may contribute to increased access to credit while enabling a potentially more accurate, segmented assessment of creditworthiness. In certain markets, traditional credit scoring methods require that those seeking loans are related to a substantial quantity of past credit data to be labelled "scorable." A credit score cannot be produced without this data, therefore a borrower who may be creditworthy may find it difficult to get credit without a credit history. Lenders may be able to make credit choices that were previously unattainable by using different data sources and applying machine learning algorithms to help create an evaluation of ability and willingness to repay (FSB, 2017).

From the second Basel Accord in 2004 onwards, financial institutions were allowed to employ internal ratings-based (IRB) credit scoring models to determine the regulatory capital requirements for credit risk. Nowadays the models used do not materially differ from those applied at that time. Taking, as an example, Logistic Regression usage for credit scoring, since Ohlson (1980), many authors employed different variants of this approach, but the underlying method remains the same. Nevertheless, scientific development in credit risk modelling has been evolving until recently (EBA, 2021a). Machine Learning models are frequently less "transparent" and more sophisticated than traditional approaches like regression analysis or simple decision trees.

According to the European Banking Association (EBA), "the main pivotal challenge comes from their complexity which leads, at least for the more complex ones, to challenges in interpreting their results, ensuring their adequate understanding by the management functions and justifying their results to supervisors" (EBA 2021, p. 5). It is known that machine learning models can yield additional benefits; however, to meet the requirements of the Capital Requirements Regulation (CRR), these models must also be interpreted, and the relevant stakeholders must have a level of understanding of how the model works that is at least in line with their involvement and legal compliance. If not, there's a chance that "black box" models will exist. To permit the use of machine learning (ML) models for regulatory



purposes, it must be essential that institutions and all their levels of management activities and bodies have a sufficient understanding of their IRB models (EBA, 2021b).

Some examples of ML techniques used in IRB modelling context that are compliant with CRR guidelines are outlined below (EBA, 2021b):

- Model validation: Alternative algorithms underlying different modelling approaches, considered model challengers are developed using ML models and are meant to act as a benchmark against which the standard model used to calculate capital requirements is measured. Both challengers and standard models are contrasted, so that the quality of results is maximized. For example, the probability of default (PD) modelling can be risk-differentiated when ML models are used as a module. This module may allow upgrades and downgrades to the PD grade previously estimated by the "traditional" PD model.
- Variable selection: Within a big dataset, ML may be utilized to find explanatory variables and their combinations with significant predictive power.

The main contribution of this dissertation lies in comparing the performance of statistical and machine learning models for probability of default estimation. An overview on these different models' specifications tries to clarify more complex techniques to the reader. Performance metrics like receiver operating characteristic curve and area under each curve (AUC) were used to decide on the best performing model. Being the winner a machine learning model, interpretability techniques are employed to overcome the existing interpretability barrier. With this, the objective is to present a model development framework that includes these, in vogue, artificial intelligence-based models, in order to produce more accurate probability of default estimations.

The remainder of this dissertation is organized as follows. The second chapter provides an overview of traditional and machine learning models, credit risk, and regulatory guidelines for IRB modelling. A summary table of recent research on this topic concludes Chapter 3, which summarizes the numerous contributions to the credit risk literature. Chapter 4 discusses the selected models, followed by a detailed examination of the most often used interpretability techniques and the performance metrics applied for evaluation. Chapter 5 presents a thorough analysis of the data, a summary of the preprocessing techniques, and an overview of each model's training. The results of each classifier are shown in Chapter 6, along with a comparison of the top-performing machine learning model versus Logistic Regression, and the conclusions are drawn from the application of interpretability approaches. Chapter 7 concludes.



## CHAPTER 2

### **Literature review**

The initial phases of credit scoring research during the second half of the 20th century utilized the statistical techniques known at the time applied to data sources of just a few tens or hundreds of cases. The seminal work of Beaver (1966) showed that a bank's lending decision could be viewed as a dichotomous choice of accepting or rejecting a loan application having as an object the analysis of financial ratios as cash flows to total assets, net income to total assets, total debt to total assets and working capital to total assets. The possibility of default could be evaluated using these financial ratios. However, this approach applied only a univariate analysis focusing on one ratio at a time.

Since then, several methods accounting for both conventional statistical methods and more advanced modelling techniques have been developed to aid decision-makers and financial analysts in predicting default (Ashofteh & Bravo, 2021). The statistical or traditional techniques have some noteworthy studies, including logistic regression models (e.g., Martin, 1977; Ohlson, 1980; Zavgren, 1985). These remain popular because they meet a few specific requirements in the context of credit risk modelling (Bücker, 2022):

the models are subject to regulation and auditors are typically familiar with interpreting logistic regression models due to their linear nature;

regulation further requires recurrent monitoring, which can be easily interpreted on a variable level for logistic regression models;

as customers have the right to an explanation of individual decisions, answers as to why a credit application has been rejected can be easily broken down since the score computed from a logistic regression model is the sum of the variables' effects.

Altman (2018) reviewed credit scoring evolution for the probability of default capture since his famous Z-score model. Despite his model being focused on corporate indicators and ratios, fifty years after it was developed it still shows impressive resilience, notwithstanding massive growth in the size and complexity of financial data. He states his skepticism towards machine learning models, as practitioners may not accept black-box methods even with a substantial improvement in prediction accuracy (Barboza et al., 2017).

Meanwhile, as computer technology improved, better and more modern machine learning models to evaluate credit risk were created thanks to developments in credit risk modelling. Credit risk analysis and pattern recognition problems are similar enough that algorithms can be utilized for assessing counterparties' creditworthiness (Barboza et al., 2017). Observing the number of related studies available on Google Scholar we observe a clear increase in recent years. This reflects the existing recent hype with Machine Learning and its application on Credit Scoring.

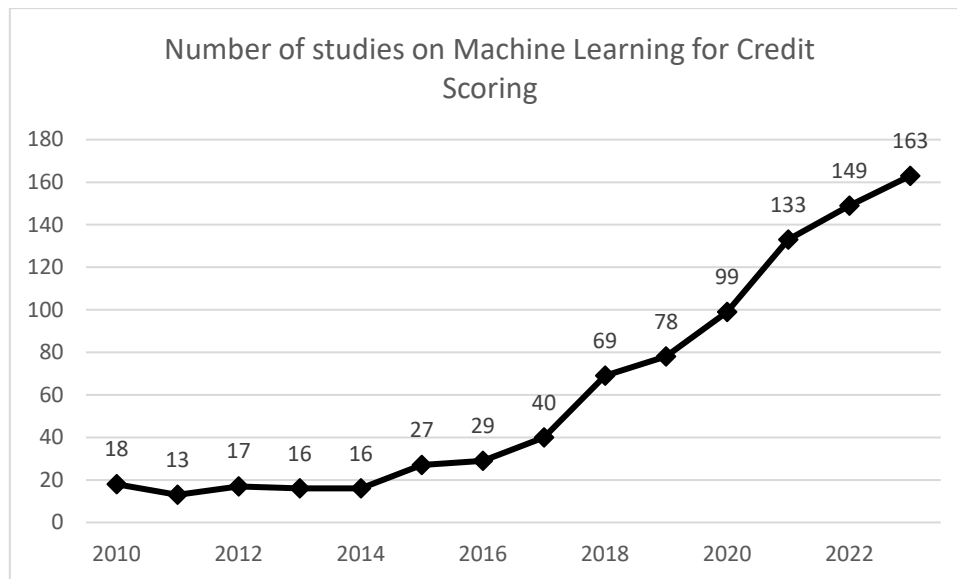


Figure 1: Number of studies on Machine Learning for Credit Scoring.  
Source: Author's preparation

The World Bank (2019) wrote a guideline for credit scoring approaches, including traditional and machine learning techniques. The usefulness behind the usage of alternative sources of data, such as social media, mobile applications, online transactions, and behavioural data was not yet proven. The existing Big Data combined with continuous innovation of computing power can lead to great opportunities.

Bensic (2005) used logistic regression, neural networks, and decision trees to model a Croatian dataset from a savings and loan association specialized in financing small and medium-sized enterprises. Brown (2012) concluded that gradient boosting and random forest classifiers perform well in dealing with sample where a large class imbalance was present. Huang (2007) applied a Support Vector Machine (SVM) to a German dataset. According to his study, SVM can achieve a good classification performance when dealing with datasets with low dimensionality; however, one should avoid overfitting the training data. Gedela et al. (2022) compared four different models: logistic regression, neural network, decision tree, and AdaBoost. The Area under the receiver operating characteristics (ROC) curve (AUROC) was computed to compare the different models. To generate the curves, the authors calculated the True Positive Rate and the False Positive Rate of each model. Other metrics were used such as accuracy, specificity, sensitivity, precision and F-Score. Li et al. (2020) investigated the application of the Extreme Gradient Boosting (XGBoost) method to the credit scoring problem. This model has become increasingly prevalent in the credit scoring domain over recent years due to its superior predictive performance (Chen et al. 2024). Ponsam et al. (2021) employed the Light Gradient Boosting Machine (LightGBM), an open-source framework developed by Microsoft. According to his study, ensemble model's result predictions tend to be less noisy and more stable, since they are aggregations from multiple models.

Given that all the above models showed great results in predicting credit defaults, a comparison will be conducted between them and traditional models, to confirm the extent to which there is a relevant performance difference between them. As such, the first hypothesis is defined as:

**H1: Machine Learning models perform better than statistical models at predicting credit default**

Regarding changes to the data that could improve a model's performance, Darst et al. (2018) employed a Recursive Feature Elimination (RFE) algorithm to mitigate the effects of high-dimensional correlated data. Ding et al. (2014) applied an Analysis of Variance (ANOVA) that showed an efficient predictive performance improvement. To better understand if the effects of these techniques have a positive impact on a model's performance, the second hypothesis is defined below:

**H2: Feature selection techniques improve the performance of each model.**

A selection of the more recent research on the use of machine learning models in credit scoring applications is shown in Table 1. The year of publication, authors' names, the data used in each publication, and the applied models are listed in the table.

Year	Authors	Data	Models
2023	Zhu, Chu, Song, Hu, Peng	Tianchi Competition	Logistic Regression; XGBoost; LightGBM; Decision Tree
2023	Barbaglia, Manzan, Tosetti	European Datawarehouse	Logistic Regression; Gradient Tree Boosting; XGBoost
2022	Khatir, Almustfa, Bee	German Credit Dataset	Random Forest; K-Nearest Neighbor; Decision Tree; Gaussian Naive Bayes; Neural Network; Logistic Regression
2021	Bucker, Szepannek, Gosiewska, Biecek	Home equity line of credit	Support Vector Machines, Gradient Boosting
2021	Oskarsdottir, Bravo	Agricultural lending	XGBoost
2021	Khanh, Duong, Quang-Linh, Ân, Nguyen, Nguyen	Kalapa Credit Score	LightGBM, CatBoost Random Forest
2020	Ariza-Garzon, Arroyo, Caparrini, Segovia-Vargas	P2P lending	XGBoost; Random Forest
2019	Bracke, Datta, Jung, Sem	UK regulated mortgages	Logistic Regression; Gradient Tree Boosting
2019	Kim & Cho	Lending Club	Convolutional Neural Network, Deep Learning

*Table 1: Summary of recent applications regarding credit risk modelling.  
Source: Author's preparation*

Regulators and supra-national entities have published various papers about credit risk and its modelling using traditional and machine learning techniques. The EU Council (2017) emphasized that banks must not only restructure their business models and promptly resolve their non-performing loan (NPL) issues, but also take steps to prevent NPLs from arising in the first place in the future. EBA (2021) recognizes that machine learning presents both potential and challenges when used for internal ratings-based (IRB) models to determine regulatory capital for credit risk. The difficulties are in: (i) interpreting their findings; (ii) making sure management functions fully comprehend them; and (iii)

defending their findings to supervisors. On the other hand, the opportunities are improved risk differentiation, risk quantification, and data collection and preparation.

A paper on post-hoc explainers for black-box models was released by Deloitte in 2023. These explainers may be utilized to understand how these machine learning algorithms make decisions. These explainers are techniques to examine dynamics derived from the model output and are independent of the model. These can be global, aiming to provide transparency into the model's decisions, or local, focusing on the reasons behind the model's result for a particular observation.

Lundberg et al. (2017) developed a global interpretation technique called Shapley Additive Explanations (SHAP) that assigns each feature an importance value for a particular prediction. By presenting several different estimation methods for SHAP values, along with proofs and experiments, they show that these values are useful for model interpretation.

Ribeiro (2016) developed Local Interpretable Model-agnostic Explanations (LIME), an algorithm to explain the predictions of any classifier or regressor in a faithful way, by approximating it locally with an interpretable model. His experiments demonstrate that explanations are useful for many models in trust-related tasks in the text and image domains, with both expert and non-expert users: deciding between models, assessing trust, improving untrustworthy models, and getting insights from predictions.

Some studies even compiled all these techniques into a methodology to better aid future research on how to proceed regarding data preparation, modelling, and explaining the predictions. Moscato (2021) proposed an approach to help structure this work's methodology. He divides it into three main modules: ingestion, classification, and explanation. The ingestion module aims for data cleaning, like removing features with a relevant number of missing or null values or converting categorical features to numeric ones. An exploratory data analysis is also performed, by computing a correlation analysis to better understand the data and its attributes. Classification is where the author chooses the different models or classifiers to be tested against each other. Lastly, the explanation module compares different explainable artificial intelligence (XAI) techniques for explaining the obtained results.

Bücker (2022) suggests a structured framework called Transparency, Auditability, and eXplanability for Credit Scoring model (TAX4CS) for model-level and instance-level exploration, starting with general measures of model performance (or accuracy of single predictions, respectively). It also provides tools to assess the suitability of models but also to compare them. It covers techniques like SHAP, LIME, and ROC curve.

## CHAPTER 3

### Methodology

#### 3.1 Statistical Models

Logistic regression (LR), Multivariate discriminant analysis (MDA), and linear discriminant analysis (LDA) are examples of traditional statistical models. Statistical models identify the best combination of explanatory input variables that can be used to analyse, predict, and model the risk of credit default. Because of some rigid assumptions like linear separability, multivariate normality, the independence of the predictive variables, and the existence of a preexisting functional form, they frequently struggle to recognize the complexity, boundaries, and interrelationships of the financial variables (Chen et al. 2016).

##### 3.1.1 Logistic Regression

One of the most significant models for categorical response data is the logistic regression or logit model. This is an example of a general model whose primary function is to calculate the likelihood of a binary response given a set of predictor variables (Al-Aradi, 2014). Due to its simplicity in development, confirmation, calibration, and interpretation, the logit model has gained popularity as an approach to assessing the probability of default (The World Bank Group, 2019). It is nowadays a commonly used and recognized technique for binary outcome variable analysis. This popularity is a result of the readily interpretable findings of the fitted model, which can be used to estimate odds ratios or probabilities, as well as the readily available software in both computer and microcomputer packages (Al-Aradi, 2014).

The logistic function is an S-shaped or sigmoid curve which is approximately linear in the middle but curved at either end, as  $X$  approaches low or very high values (De Maris, 1995). The sigmoid curve is given by the following function:

$$f(x) = \frac{1}{1 + e^{-x}} \quad (1)$$

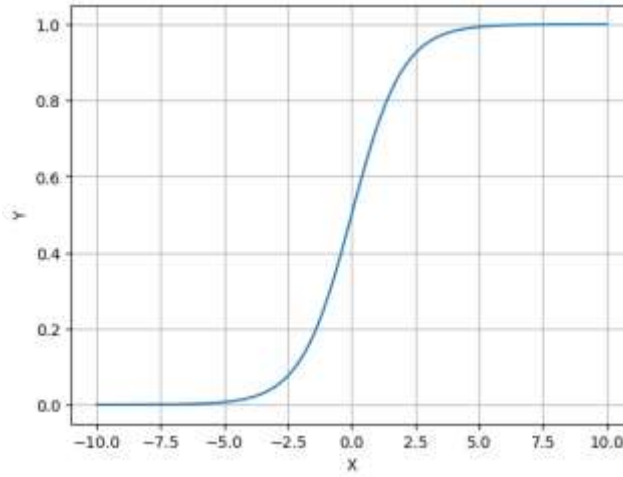


Figure 2: Sigmoid function.  
Source: Author's preparation

The logistic regression is then given by the equation below:

$$P(y = 1 | x_1, \dots, x_n) = \frac{1}{1 + e^{-(\beta_0 + \sum_{i=1}^n \beta_i x_i + \varepsilon)}} \quad (2)$$

where  $p$  is the probability of the outcome of interest or odds,  $\beta_0$  is the intercept term,  $\beta_i$  is the respective coefficient of  $X_i$ , that represents the independent features and  $\varepsilon$  is the error term. The outcome variable in a binary logistic regression model is a dummy variable, which sets it apart from a linear regression model (Hosmer et al., 2013).

## 3.2 Machine Learning Models

EBA takes ISO/IEC 38505-1:20175 definition of Machine Learning, a “process using algorithms rather than procedural coding that enables learning from existing data to predict future outcomes” (2021, p. 9). In general, it is a branch of computer science that works with models’ creation whose parameters are determined, mostly without human input, automatically from data (EBA, 2021b).

### 3.2.1 Decision Trees

Decision trees are graphs that resemble trees and are used for classification. They consist of decision branches and various outcomes represented by leaves. The goal of this technique is to maximize the selected performance measure concerning the target variable. Despite having a "white box" aspect that makes it relatively simple to grasp, the decision tree approach is limited in how it can handle many variables and extrapolate outcomes (Stroie, 2013).

The outcome of a decision tree is made at the leaves of the tree, which are nodes that represent decisions depending on features. Since the decisions require a condition defined by a set of attributes as



input, they approximate if-then rules. The decision tree performs more analysis if it is not satisfied, returning an outcome that is the estimated value (Khatir et al. 2022).

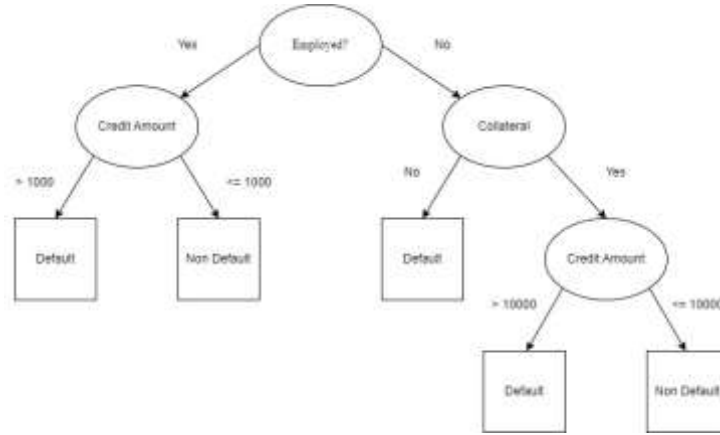


Figure 3: An example of a decision tree based on credit data.  
Source: Author's preparation

At each node, choose the attribute that maximizes the information gain. This depends on entropy, a measure of disorder or impurity in a given dataset. For a dataset that has  $C$  classes and the probability of randomly choosing data from class  $i$  is  $p_i$ , then entropy can be mathematically represented as:

$$Entropy = - \sum_{i=1}^c p_i \log_2 p_i \quad (4)$$

The information gain will quantify the quality of a split rule. It's calculated for a split by subtracting the weighted entropies of each branch from the parent entropy. Given  $w_i$  as the weight of class  $i$  after a split, the quality of the split is determined by weighting the entropy of each branch by how many elements it has, where  $\sum_{i=1}^n w_i = 1$ .

$$Information\ Gain = Entropy_{parent} - \sum_{i=1}^n w_i \cdot Entropy_i \quad (5)$$

The best split is chosen by maximizing Information Gain.

### 3.2.2 Random Forests

A random forest is essentially an ensemble of decision trees. Since multiple decision trees were generated using bootstrapped samples taken from the initial sample, random forests are a generalization of decision trees (Khatir et al., 2022). In addition, a subset of randomly selected features is used in the development of each tree. Since every decision tree outputs a predicted class, a random forest uses the majority vote criterion to predict the class overall, accounting for the output of all decision trees.

The bagging concept comes from "bootstrap aggregation", the name given to this voting procedure. While predicting a result, the aggregate averages over the many versions; while predicting a class, it uses a plurality vote (Breiman, 1996). By generating bootstrap duplicates of the learning set and

utilizing them as new learning sets, many versions are created. Bagging can increase accuracy if changes in the learning set result in appreciable improvements to the predictor created (Liu et al., 2012).

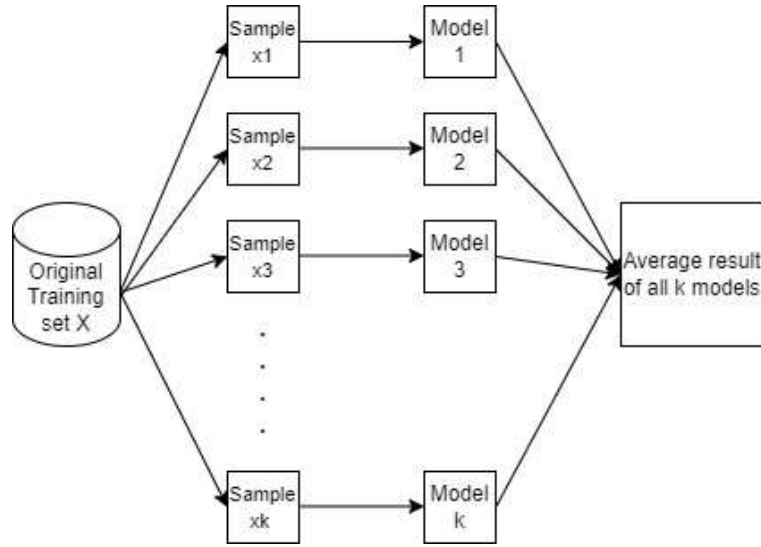


Figure 4: Bagging concept visual example.  
Source: Author's preparation

Another indicator used to estimate the quality of each decision tree's split is the Impurity Index, where  $p_i$  is the frequency of class  $i$  at a node and  $C$  is the number of unique classes.

$$\text{Impurity Index} = \sum_{i=1}^c p_i \cdot (1 - p_i) \quad (6)$$

As an ensemble model, a merger between predictions from multiple models occurs, that aims to increase accuracy while offering more robust forecasts. Averaging accounts with all the individual model's predictions for a given sample point  $x$ , with  $y_n$  being the estimated PD for each model and  $N$  the number of models.

$$\hat{y} = \frac{\sum_{n=1}^N y_n}{N} \quad (7)$$

### 3.2.3 Artificial Neural Networks

Artificial Neural networks (ANN) aim to build networks that function in a similar way to the brain. Thus, the brain's structure serves as the inspiration for the concept of neural networks. In the human brain, a neuron receives electrical impulses from a vast array of dendrites. These dendrites then convert the signals into electrical pulses, which are subsequently sent by an axon to numerous synapses, which relay concepts or information to the dendrites of other neurons (Abdou et al., 2008).

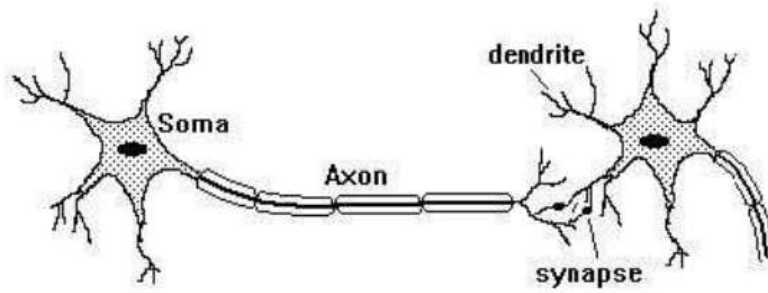


Figure 5: Neuron signal representation.

Source: Cilimkovic, M. (2015). *Neural networks and back propagation algorithm*. Institute of Technology Blanchardstown, Blanchardstown Road North Dublin, 15(1).

ANNs are composed of several basic nodes connected through one or more layers. The basic neuron components used in neural nets vary depending on the type of net utilized. Every neuron in the network completes a part of the computations. First, it receives specific values as inputs, processes them through a simple calculation, and outputs the result. Except for neurons, that provide the network's ultimate output values, a neuron's output value is sent to another neuron as one of its inputs (Abdou et al., 2008).

A simple neural network is composed of an input, some hidden layers, and an output. Each connection carries some weight in general. The raw data in the network is represented by the input layer. The data from the input layer is then received by the hidden layer, which applies a weight value to modify the input data. The new value is then sent to the output layer, where it is further adjusted by a weight from the hidden-to-output layer link. If the objective is to reach non-linearity, we must add an activation function between layers (Cilimkovic, 2015).

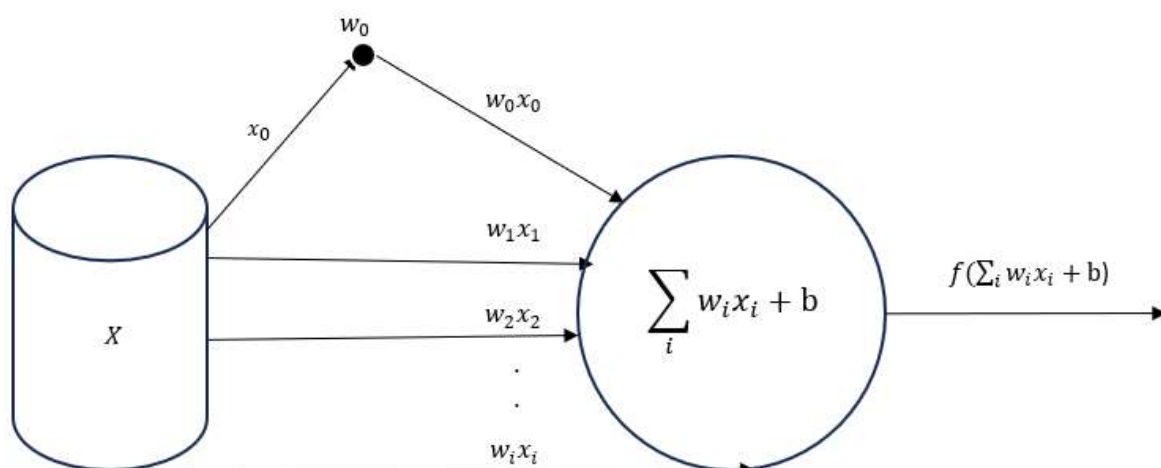


Figure 6: One-layer artificial neural network representation.

Source: Author's preparation

$x \cdot w$  are the dot product between input and weight vectors, and  $b$  denotes the bias also known as the offset that moves the entire activation function.

$$z = x \cdot w + b \quad (8)$$

To introduce non-linearity into the output of the neurons,  $z$  must be passed into a non-linear activation function such as sigmoid (Eq. 1) or rectified linear unit (ReLU) (Eq. 9) functions.

$$f(z) = \max(0, z) \quad (9)$$

For a neural network to learn, we need to codify the correct or incorrect outputs, and how far an incorrect output is from the correct one. For a classification problem, a cross-entropy cost function  $C$  can be used. It serves as a measure of how far away a particular solution is from an optimal solution.

$$C = -\frac{1}{n} \sum_x [y \cdot \ln \hat{y} + (1 - y) \cdot \ln(1 - \hat{y})] \quad (10)$$

The different weights ( $w$ ) are then iteratively updated by applying a Taylor approximation to the Cost Function (Eq. 10):

$$\Delta C \approx \frac{\partial C}{\partial w} \cdot \Delta w + \frac{\partial C}{\partial b} \cdot \Delta b \quad (11)$$

from here we can extract the gradient and the change in both weights and bias, from a vectorized form:

$$\Delta C = \begin{bmatrix} \frac{\partial C}{\partial w} & \frac{\partial C}{\partial b} \end{bmatrix} \times \begin{bmatrix} \Delta w \\ \Delta b \end{bmatrix} = \nabla C \cdot \Delta v \quad (11.1)$$

The gradient applied to the cost function gives the two gradients which are telling in which direction the cost increases the most. The objective is to make changes to the variables leading to the opposite direction, and that is essentially what gradient descent does.

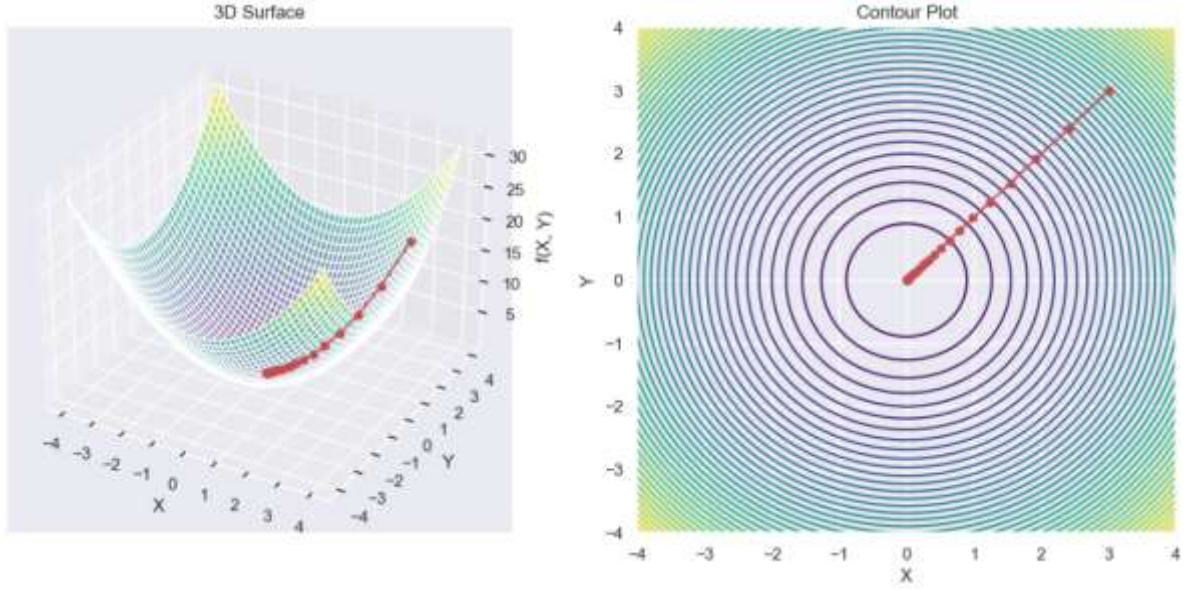


Figure 7: Gradient descent visual representation.  
Source: Author's preparation

So, if we let  $\Delta v = -\nabla C$  then we end up with  $\Delta C = -\tau \|\nabla C\|^2$ , if we add the learning rate  $\tau$ , a parameter that controls how fast the movements are towards the minimum gradient. Then, all weights and bias should be updated in each iteration as below:

$$w \rightarrow w' = w - \tau \cdot \frac{\partial C}{\partial w} \quad (12)$$

$$b \rightarrow b' = b - \tau \cdot \frac{\partial C}{\partial b} \quad (13)$$

### 3.2.4 Support Vector Machines

Vapnik et al. (1995) first introduced Support Vector Machines (SVM) under statistical learning theory. Pattern recognition has seen numerous successful SVM applications, showing that SVM is a competitive classifier. To distinguish between the two different classes, SVM looks for an ideal hyperplane with a maximum margin that serves as the decision boundary.

Although LDA and SVMs compute optimal hyperplanes concerning their respective goals, there are several differences between them. Only when the covariance matrices for each class are the same is the hyperplane calculated by LDA optimal, a condition that is frequently broken in real-world scenarios. In contrast, SVMs do not assume anything while computing ideal hyperplanes for margin maximization (Gokcen et al. 2002).

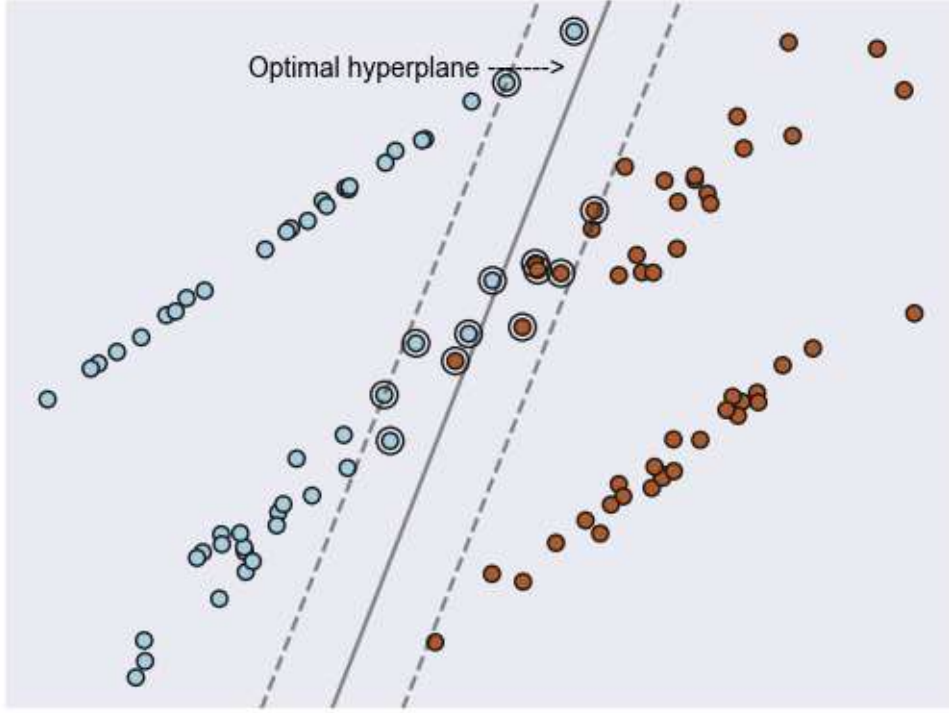


Figure 8: Support Vector Machine visual representation.  
Source: Author's preparation

This hyperplane can be defined, given an  $n$  dimensional feature vector  $x = (X_1, \dots, X_n)$ , as below (Vapnik, 2015):

$$\beta_0 + \sum_{i=1}^n \beta_i X_i = 0 \quad (14)$$

in a binary classification problem, represented by two labels  $y \in \{-1, 1\}$  we get the following mathematical separating property:

$$y = \begin{cases} 1, & \text{if } \beta_0 + \sum_{i=1}^n \beta_i X_i > 0 \\ -1, & \text{if } \beta_0 + \sum_{i=1}^n \beta_i X_i < 0 \end{cases} \quad (15)$$

To overcome a linear boundary limitation, Kernel functions are introduced to transform a linear problem into a higher dimensional one, where you can classify linearly to get a non-linear solution. This Kernel trick allows SVMs to learn nonlinear functions. By replacing inner products with this kernel function, a higher dimension can be used to create a nonlinear decision boundary in the original  $R^N$ , which corresponds to a linear decision boundary in  $R^M$  (where  $M > N$ ).

$$\begin{aligned}
& x_i, x_j \in R^N, K(x_i, x_j) = \langle \Phi(x_i), \Phi(x_j) \rangle_M \\
& \text{where } \langle \Phi(x_i), \Phi(x_j) \rangle_M \text{ is an inner product of } R^M \\
& \text{and } \Phi(x) \text{ transforms } x \text{ to } R^M
\end{aligned} \tag{16}$$

A popular kernel function is the polynomial that provides a significantly more flexible decision boundary and essentially amounts to fitting a SVM in a higher-dimensional feature space involving  $d$  - degree polynomial of the features.

$$\left(1 + \sum_{j=1}^p x_{ij}x_{kj}\right)^d \tag{17}$$

Error variables  $\varepsilon_1 \dots \varepsilon_n$  can be added when there isn't a perfect discrimination between both classes. Their sum must be kept below a pre-defined budget  $B$  (Vapnik et al., 2015).

### 3.2.5 XGBoost

An effective technique for regression analysis and classification is gradient boosting. Using adaptively reweighted versions of the original training data, a weak classification algorithm is applied sequentially in this method (Barbaglia et al., 2023). The weighting system is adaptive since it gives a higher weight to the data that the classifier misclassified in the previous phase and a lower weight to the observations that were properly categorized. As a result, the boosting classification algorithm prioritizes harder-to-classify observations over those that have previously been accurately identified as iterations go on (Barbaglia et al., 2023).

XGBoost is an enhanced gradient tree boosting model developed by Chen & Guestrin (2016). An initial prediction is created before utilizing XGBoost to fit a training dataset. The predicted value and the observed values are used to compute residuals. A decision tree is generated using the residuals applying a similarity score for residuals. The process is repeated until the residuals stop decreasing, or after a predetermined number of repetitions. Unlike Random Forest, each succeeding tree gains knowledge from previous ones and is not given the same weight.

This has similarities to the gradient boosting algorithm, which minimizes the loss function to assess how well the model matches the available data by using an additive form of weak base learners. During a given number of standard gradient boosting iterations, the base learner is found by minimizing the objective function (Chen et al., 2023).

$$obj = \sum_i L(y_i, \hat{y}_i) + \sum_k \Omega(f_k) \quad (18)$$

where  $L(y_i, \hat{y}_i)$  can be any loss function that measures the difference between the prediction and true label for a given training instance.  $\Omega(f_k)$  defines the complexity of a tree  $f_k$ , as follows (Mitchell et al., 2017):

$$\Omega(f_k) = \gamma T + \frac{1}{2} \lambda w^2 \quad (19)$$

$T$  is the number of leaves of the tree  $f_k$  and  $w$  is the predicted weights stored at the leaf nodes. Adding this complexity factor, the optimization will force a less complex tree contributing to overfit reduction.  $\gamma T$  provides a penalty for each additional tree leaf and  $\lambda w^2$  penalises extreme weights (Mitchell et al., 2017).

### 3.2.6 LightGBM

With Gradient-based One-Side Sampling (GOSS) and Exclusive Feature Bundling (EFB), LightGBM is another gradient boosting decision tree-based algorithm. Several tests on publicly available datasets demonstrate that LightGBM can achieve almost the same accuracy while up to 20 times faster training times (Ke et al., 2017).

Based on the theory behind GOSS, data instances with various gradients have different purposes in the information gain computation. As per the concept of information gain, instances that are under-trained would contribute more to the information gain. Therefore, to maintain accuracy, it is preferable to retain instances with large gradients and discard instances with small gradients exclusively. When the amount of information gain has a wide range, this type of treatment can produce a more accurate gain estimation than uniformly random sampling at the same target sample rate. LightGBM buckets data into bins using a histogram-based approach. To split the data, compute the gain, and iterate, bins are utilized rather than each data point. It is also possible to optimize this strategy for a sparse dataset (Ke et al., 2017).

By sorting the data instances according to their absolute gradient or residual values in descending order and selecting the top  $a\%$  instances. Then, it randomly samples  $b\%$  instances from the remaining data. Finally, in order not to alter the distribution of the data, a constant multiplier of  $\frac{1-a}{b}$  is applied to the instances with smaller residuals when calculating the information gain.



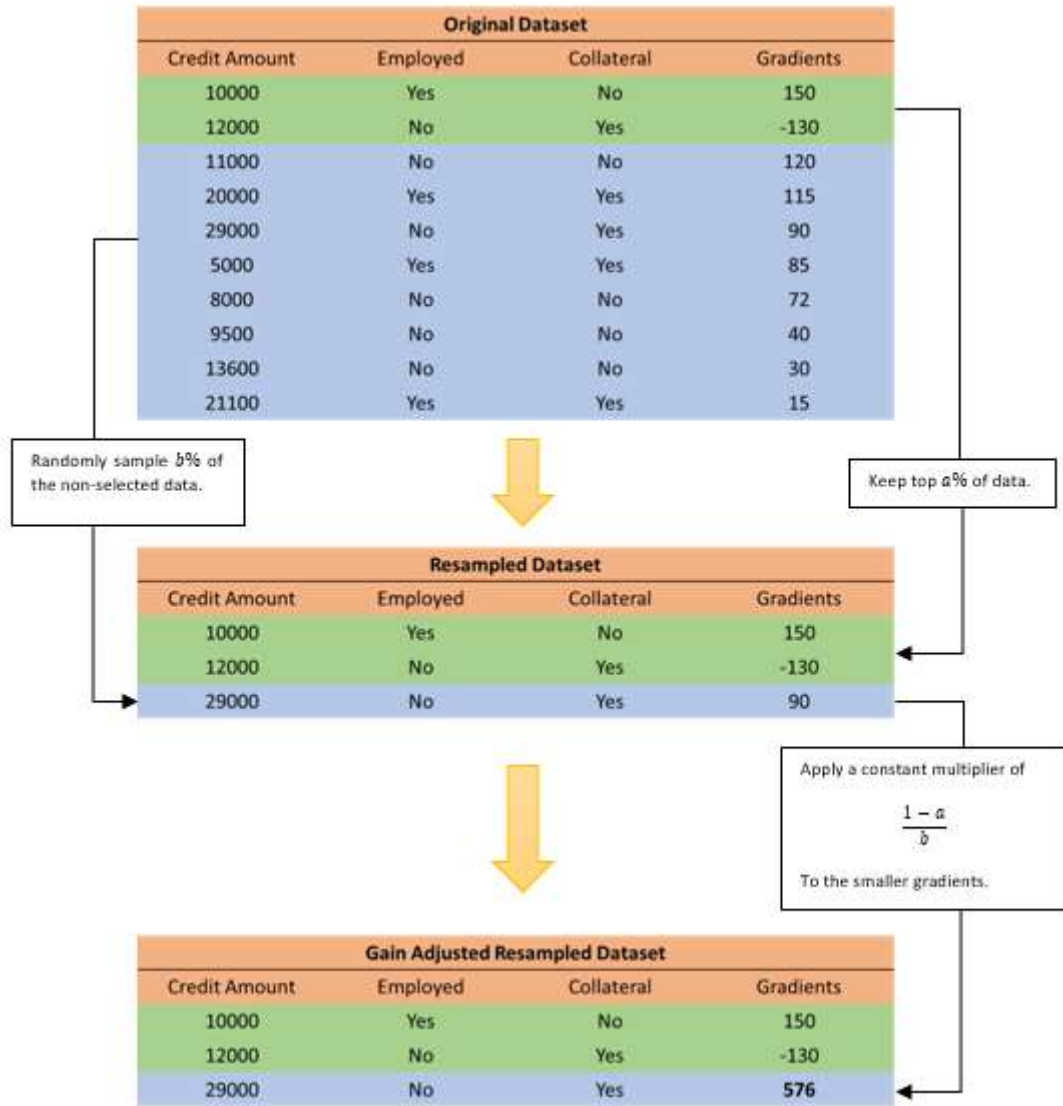


Figure 9: Gradient one-side sampling example.  
Source: Author's preparation

EFB which lowers dimensionality and increases speed and efficiency by combining unique features, is another aspect of this model (Ke et al., 2017). The goal is to ensure that even though the features are merged, the original values of each feature can still be retrieved from this new bundled variable. Adding offsets to the variable's original values is a workaround, for example: "Number of delinquencies" has values ranging from  $[0, 10[$  and "Number of loans" from  $[0, 20[$ . Offsetting the "Number of loans" variable by 10 would mean that both variables are no longer in conflict, since the "Number of loans" range is now  $[10, 30[$ . With this, a merger that allows each value original feature is possible.

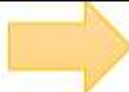
Original Dataset			Feature Bundled Dataset
Number of delinquencies	Number of loans		New feature
N/A	10	<div>Added an offset of 10 to "Number of loans"</div> 	20
N/A	5		15
4	N/A		4
8	N/A		8
3	N/A		3
N/A	8		18
N/A	4		14
N/A	2		12
1	N/A		1
N/A	9		19

Figure 10: EFB example.  
Source: Author's preparation

### 3.2.7 AdaBoost

The AdaBoost algorithm employs the boosting method Adaptive Boosting as an ensemble approach in machine learning, with the designation being a combination of both words (Gedela et al., 2022).

Every instance is assigned an initial set of weights, with incorrectly classified instances getting larger weights. A given number of decision trees are produced during the data training procedure. When building the first decision tree or model, preference is given to the record erroneously classified in the prior model (Gedela et al., 2022). We continue until a predetermined number of beginning trees remain for us to develop. When the random forest approach is used,  $n$  trees are produced. It creates just-right trees with many leaves and a root node. The depth of a random forest is completely dependent on chance, even though some trees may be larger than others. Conversely, AdaBoost only produces a stump or a node with two leaves.

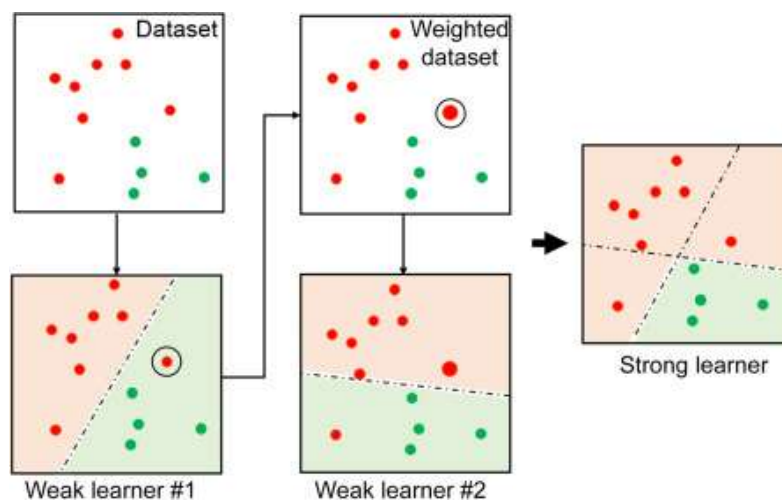


Figure 11: AdaBoost algorithm visual representation.

Source: Misra, S., Li, H., & He, J. (2020). Noninvasive fracture characterization based on the classification of sonic wave travel times. *Machine learning for subsurface characterization*, 4, 243-287.

With  $H = \{h_1, h_2, \dots, h_n\}$  as a set where  $h_1, h_2, \dots, h_n$  represent individual weak learners, the algorithm initializes weights  $w_i$  that satisfy  $\sum_{i=1}^n w_i = 1$ . Let  $\varepsilon_j = \sum_{i=1}^n w_i(t) I[y_i \neq h_j(x_i)]$  be the sum of weighted classification errors for the weak classifier  $h_j$  where:

$$I[y_i \neq h_j(x_i)] = \begin{cases} 1, & \text{if True} \\ 0, & \text{if False} \end{cases} \quad (20)$$

then, after choosing the weak classifier that minimizes the sum of weighted classification error, all weights are updated to the next iteration by:

$$w_i(t+1) = \frac{w_i(t) \exp(-\alpha(t) y_i h(t)(x_i))}{Z(t)} \quad (21)$$

where  $\alpha(t) = \frac{1}{2} \log \left( \frac{1-\varepsilon(t)}{\varepsilon(t)} \right)$  and  $Z(t) = \sum_{k=1}^n w_k(t) \exp(-\alpha(t) y_i h(t)(x_i))$  as a normalization factor (Hu et al., 2008).

### 3.3 Interpretability techniques

Machine learning models have the potential to quickly develop into "black boxes," opaque systems whose internal decisions are hard to comprehend. As a result, it can be challenging to understand (and validate) how a model reached a certain conclusion or prediction (EBA, 2020). Depending on the learning mode and underlying model complexity, an ML solution's opaqueness can change. For instance, decision trees' internal workings are easier for humans to understand, whereas neural networks' characteristics are less transparent because of the inherent complexity of the underlying algorithm. The opposing idea of explainability is deeply related to this technological opaqueness (EBA, 2020).

The degree of understanding of the model outputs determines how successful human participation is, hence explainability is a crucial component of both representativeness and accuracy in the models (EBA, 2020).

#### 3.3.1 Shapley Additive Explanations

The Shapley value, an idea from game theory that determines a player's fair reward based on how much they contributed to the overall benefit after coalitions are accounted for, is the foundation of the Shapley Additive Explanations (SHAP) analysis. Coalitions are potential feature subsets, players are specific feature values, and the fair payout denotes the contribution of a particular feature value to the prediction in machine learning. Therefore, in our case, averaging the prediction differences produced between the model with and without all feasible feature subsets and considering all feature groups will yield the Shapley value of a feature value in the target (Chen et al., 2023).

Considered an additive feature attribution method this class explains a model's output as a sum of real values attributed to each independent variable. This method is a linear function of binary variables (Lundberg et al., 2018):

$$g(z') = \phi_0 + \sum_{i=1}^M \phi_i z'_i \quad (22)$$

where  $z' \in \{0,1\}^M$ ,  $M$  is the number of independent variables, and  $\phi_i \in \mathbb{R}$ .

This method requires retraining the model on all feature subsets  $S \subseteq F$ , where  $F$  is the set of all variables (Lundberg et al., 2017). To compute each feature attribution ( $\phi_i$ ), model  $f_{S \cup \{i\}}$  is developed with feature  $i$  included, and another model  $f_S$  is trained without that feature. Afterwards, the predictions from both models are compared  $f_{S \cup \{i\}}(x_{S \cup \{i\}}) - f_S(x_S)$ , where  $x_S$  holds the values of the input features in the set  $S$ . This is done for all possible subsets inside  $S \subseteq F \setminus \{i\}$ , with a weighted average of all possible differences as a result (Lundberg et al., 2017).

$$\phi_i = \sum_{S \subseteq F \setminus \{i\}} \frac{|S|! (|F| - |S| - 1)!}{|F|!} [f_{S \cup \{i\}}(x_{S \cup \{i\}}) - f_S(x_S)] \quad (23)$$

### 3.3.2 LIME

By using a local interpretable model (LIME), where a family of potential interpretable models, as presented by Ribeiro et al. (2016), seeks to interpret the machine learning model prediction of a particular target by appropriating the "black box" machine learning model.

LIME creates a new dataset by randomly modifying features from the target and receiving the appropriate predictions from the "black box" model to fit a local surrogate centred on the target. The new dataset, weighted by the distances between the modified samples and the target, is then used to train the interpretable model. As a result, the trained interpretable model guarantees local fidelity, which implies that while it may not assure a good global approximation, it should be a good local approximation of the "black box" model predictions locally (Chen et al., 2024).

Let  $f$  be the "black-box" model, where  $f(x)$  is the probability that  $x$  belongs to a certain class (default or non-default), and  $g$  a local interpretable model. Additionally, to measure the proximity between the real ( $x$ ) and perturbed ( $z$ ) samples,  $L(f, g, \pi_x)$  is used as a loss function, where  $\pi_x$  is a proximity measure between  $z$  to  $x$  (Ribeiro et al., 2016)

$$\operatorname{argmin}_{g \in G} L(f, g, \pi_x) + \Omega(g) \quad (24)$$

with  $G$  is a set of potentially interpretable models and  $\Omega(g)$  a measure of complexity.

A locally weighted square loss function ( $L$ ), with  $\pi_x(z) = \exp(-D(x, z)^2 / \sigma^2)$  as an exponential kernel defined on some distance function  $D$ , like Euclidian (Ribeiro et al. 2016), is

$$L(f, g, \pi_x) = \sum_{z, z' \in Z} \pi_x(z) (f(z) - g(z'))^2 \quad (25)$$

where  $z'$  denotes the simplified inputs (Chen et al., 2024).

### 3.4 Performance metrics

Every machine learning exercise includes performance metrics. Performance metrics track and evaluate a model's performance throughout training and testing, regardless of whether it is dealing with a regression or classification problem.

A confusion matrix is essentially used for evaluating classification problems. In a binary class problem, the matrix is squared, 2x2. The row is the real value of the class label, and the column is the classifier prediction. Typically, in an unbalanced credit scoring dataset, default class observations are labelled as one, whereas non-default class observations are labelled as zero (Bekkar et al., 2014).

	Predicted Non-Default	Predicted Default
Actual Non-Default	TN	FP
Actual Default	FN	TP

*Table 2: Confusion matrix.  
Source: Author's preparation*

The following is what the confusion matrix cells' abbreviations TP, FN, FP, and TN stand for:

- TP = true positive, the number of default cases that are correctly identified as default;
- FN = false negative, the number of default cases that are misclassified as non-default;
- FP = false positive, the number of non-default cases that are incorrectly identified as default;
- TN = true negative, the number of non-default cases that are correctly identified as non-default.

Next, seven distinct metrics are calculated using those four values:

Measure	Formula
Accuracy	$\frac{TP + TN}{TP + TN + FP + FN}$
Error rate	$\frac{FP + FN}{TP + TN + FP + FN}$
Sensitivity (or True Positive Rate)	$\frac{TP}{TP + FN}$
Specificity	$\frac{TN}{TN + FP}$
Precision	$\frac{TP}{TP + FP}$
F1-Score	$\frac{2TP}{2TP + FP + FN}$
False Positive Rate	$\frac{FP}{FP + TN}$

Table 3: Performance measures  
Source: Author's preparation

The most used metric for evaluating classifiers is accuracy, which evaluates the algorithm's efficacy for a given cut-off point, by calculating the likelihood that the class label is correct. Precision assesses the model's ability to capture default, or how many of the default-labeled examples are indeed defaults. Recall is a metric for how accurate or comprehensive positive examples are; that is, how many default examples were properly classified (Bekkar et al., 2014). F1-Score is the harmonic mean of precision and recall. It provides a fair assessment of the model's effectiveness that considers both true positive and false positive rates (Lipton et al., 2014).

A Receiver Operating Characteristic (ROC) curve is a graph that displays a classification model's performance overall classification thresholds. This graph plots the True Positive Rate versus the False Positive Rate. Reductions in the classification threshold result in a higher number of positive classifications, which raises the number of True Positives and False Positives. The diagonal line is where the false positive rate and the true positive rate are equal, corresponding to a model where discrimination is entirely random (Hoo et al., 2017).

One of the most used ranking-type metrics is the AUC. In addition to comparing learning algorithms, it was utilized to build a winning learning model. In contrast to threshold and probability measurements, the AUC value indicates a classifier's overall ranking performance (Bekkar et al., 2014). This statistic provides an overall assessment of a test's capacity to distinguish between default and non-default cases. A model with no discriminating ability has an AUC of 0.5, whereas a test with perfect discrimination has an AUC of 1.0 (Hoo et al., 2017).

## CHAPTER 4

# Exploratory Data Analysis

### 4.1 Dataset details and statistics

The dataset used in this dissertation belongs to the UCI Machine Learning repository, a collection of databases for the empirical analysis of machine learning algorithms. It includes payment information from an influential Taiwanese bank from October 2005; the targets were the bank's credit card holders. A limitation while using this dataset is that the bank's default policy is unknown, so this will only be useful to predict next month's default state for each borrower. This collection comprised a binary variable – default payment (Yes = 1, No = 0) and twenty-three potential explanatory variables, concentrating on two distinct domains: credit details and personal information. For an explanation of every variable, a description can be found in Annex A.

From 30,000 total observations, 6,636 (22.1%) belong to cardholders with default payments. We can observe an existing class imbalance that is not extreme. Regarding the numeric type variables, Annex B shows different statistics for each variable, such as the mean, standard deviation, minimum, maximum, and the different quartiles (25%, 50%, and 75%). The age varies between 21 and 79 years old, with 75% of the population being up to 41 years old.

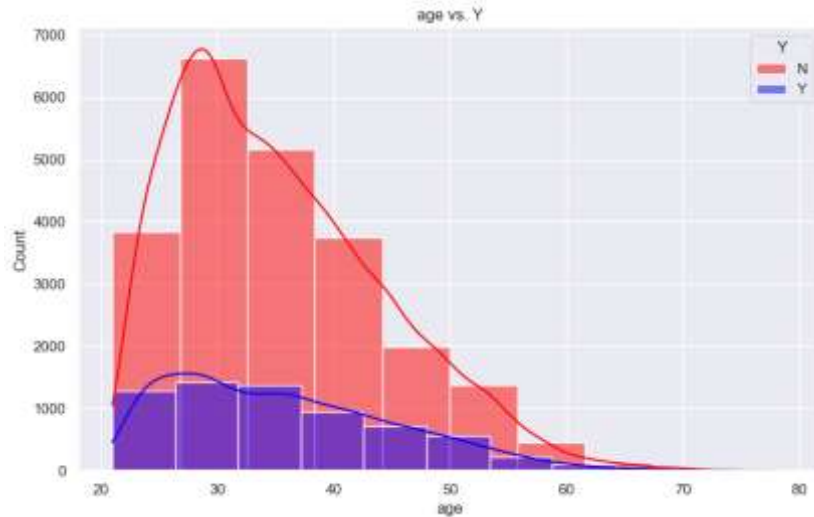


Figure 12: Age variable distribution by class.  
Source: Author's preparation

The credit amount is on average 167,484.32 Taiwanese Dollars with most credits having 240,000 or less nominal amount.

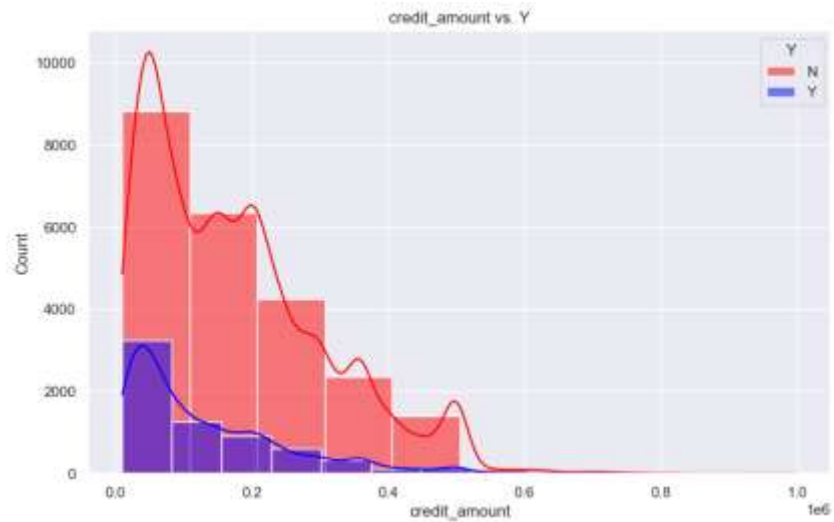


Figure 13: Credit Amount variable distribution by class.  
Source: Author's preparation

The different bill states range from a negative value, meaning an overpaid credit, and a positive value meaning accumulated debt. Finally, “amount\_payed”, as the designation indicates, shows the payment amount done at a specific time (from t-1 to t-6).

The different correlations between the variables indicate a high relationship between the bill-state features, with a minimum correlation of 0.80 between “bill\_state\_t-1” and “bill\_state\_t-6” and a maximum correlation between “bill\_state\_t-1” and “bill\_state\_t-2”. This could lead to unstable predictions due to the interdependence between these variables causing high multicollinearity level. It can be difficult to determine a variable's marginal impact when there are linear correlations between two or more variables (Chan et al., 2022). The remaining variables seem to have a low and non-significant correlation.



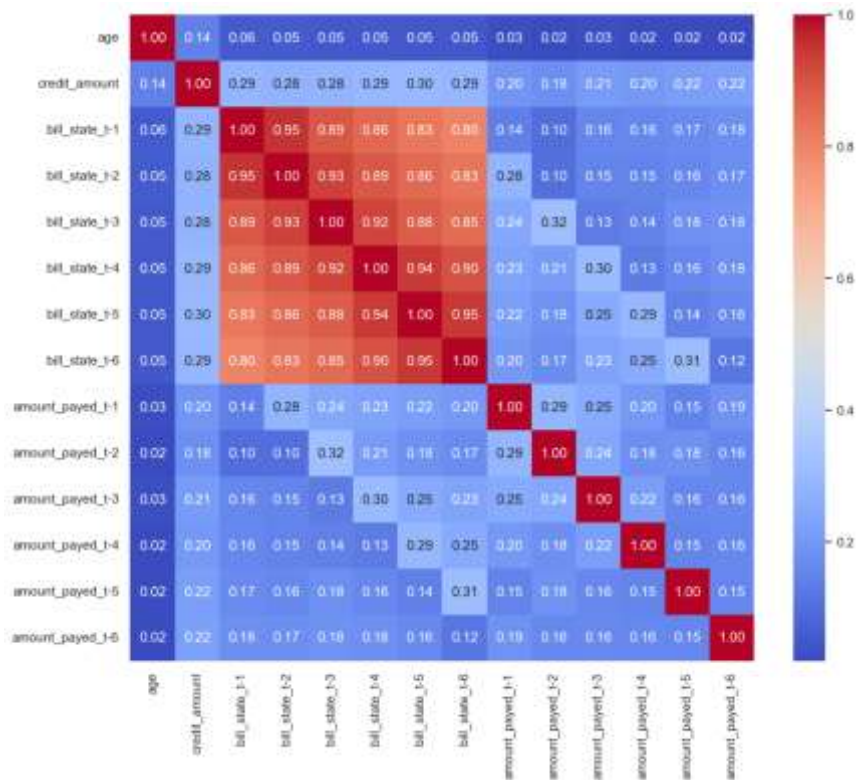


Figure 14: Correlations between different variables.  
Source: Author's preparation

For the categorical columns, different types of analysis were performed. The predominant gender is female (60.37%), but males default slightly more (24.17% versus 20.78%).

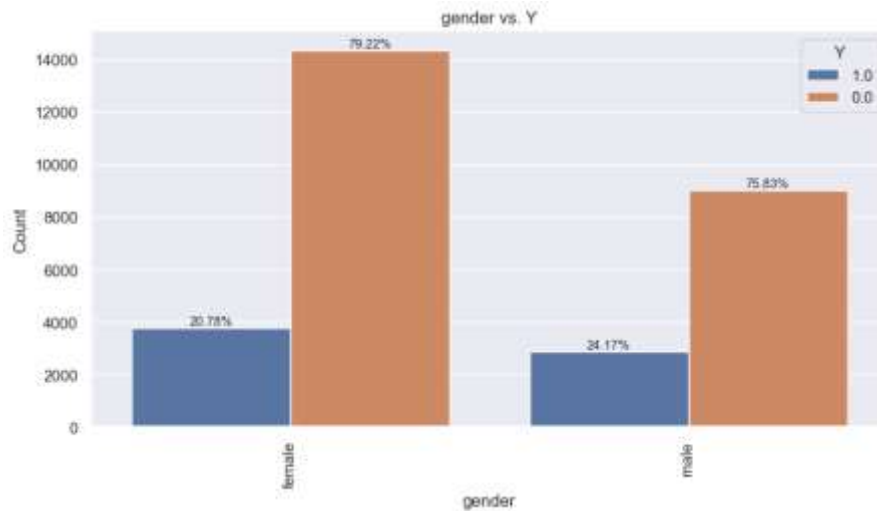


Figure 15: Gender distribution by class.  
Source: Author's preparation

82.05% of borrowers went into university, but those with high school level defaulted more in percentage terms (25.16%).

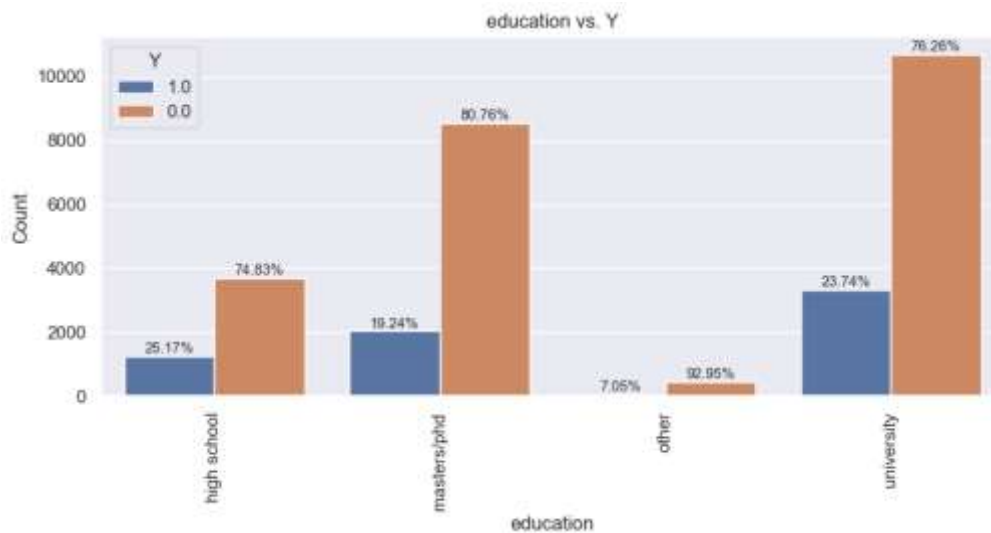


Figure 16: Education distribution by class.  
Source: Author's preparation

The marital status is balanced, with 45.54% being married and 53.21 being single. The largest default rate sits at 23.47% for married borrowers.

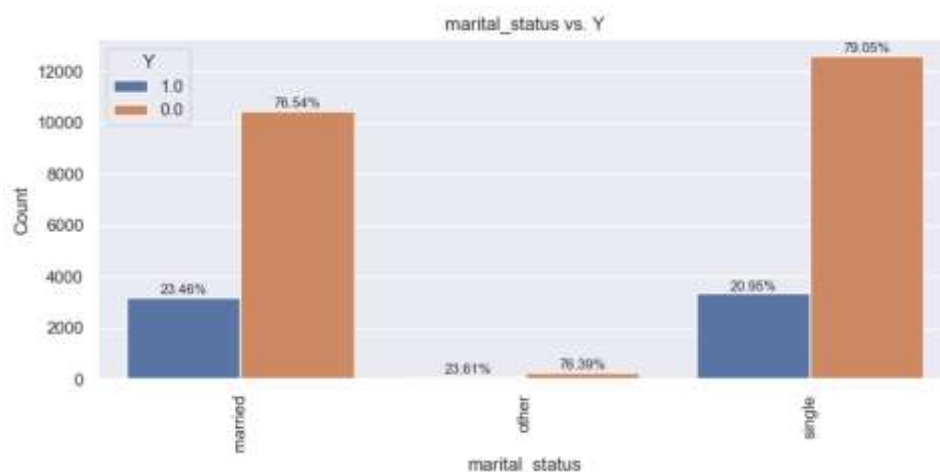


Figure 17: Marital Status distribution by class.  
Source: Author's preparation

The different repayment status indicates a very similar distribution, where most non-defaults are non-consumers, fully paid or only use revolving credit, meaning that they didn't have any delayed payments pending.

## 4.2 Data preprocessing

This phase includes data transformations needed to start modelling. Data cleaning fixes or removes incorrect, corrupted, incorrectly formatted, duplicate, or incomplete data within a dataset. Removing duplicates, filtering unwanted outliers, and handling missing data are examples of data-cleaning processes.

The dataset contained 35 duplicates that were excluded. There were no missing values or unrealistic outliers.

Since all models in use only accept numeric values as input, encoding must be done since some categorical variables were filled with text, such as gender, which could be either "female" or "male". Gender and marital status are nominal features, binary encoding was used for them. The process involves first converting the feature into an ordinal scale, then those integers into binary code, and finally dividing the digits from that binary string into distinct features (Potdar et al., 2017). Each category is given an integer, and the data is not expanded to include more columns.

Variable	Type	Mapping
gender	Nominal	"female": New binary column "gender_female" "male": New binary column "gender_male"
marital_status	Nominal	"married": New binary column "marital_status_married" "other": New binary column "marital_status_other" "single": New binary column "marital_status_single"
education	Ordinal	"masters/phd": 4 "university": 3 "high school": 2 "other": 1

Table 4: Categorical variables mapping.  
Source: Author's preparation

Numerical features were standardized using a scalar. Data scaling transforms values until they are within a specific range. This ensures that no single feature dominates the distance calculations in an algorithm and helps to improve the algorithms' performance and convergence. Standardization is a method where the mean of each feature  $x_i$  becomes 0 and the standard deviation becomes 1 (Pedregosa et al., 2011).

$$x'_i = \frac{x_i - \bar{x}_i}{\sigma_i} \quad (26)$$

Four different datasets were prepared, using distinct techniques to understand their effectiveness in improving the performance of each model. Feature selection methods can be used to reduce the size of the dataset or adapt it to achieve a more efficient analysis (Jović et al., 2015). These can be wrapper methods such as Recursive Feature Elimination (RFE), or filter methods such as the Analysis of Variance (ANOVA) test. The main difference between both types is that for filter methods features are selected based on an individual performance measure, and wrapper methods consider the performance of a subset of variables (Jović et al., 2015). The Recursive Feature Elimination asks for a given external estimator that assigns weights to features, where the selected estimator is Logistic Regression. For the

ANOVA test results, a p-value equal to or higher than 0.05 was considered to exclude features. The selected features that resulted from each method are highlighted in bold, below:

Features	Ranking
<b>credit_amount</b>	1
<b>repay_status_t-3</b>	1
<b>repay_status_t-2</b>	1
<b>repay_status_t-1</b>	1
<b>marital_status_married</b>	1
<b>gender_female</b>	1
<b>education_num</b>	1
<b>amount_payed_t-2</b>	1
<b>amount_payed_t-5</b>	1
<b>bill_state_t-6</b>	1
<b>bill_state_t-1</b>	1
<b>bill_state_t-2</b>	1
<b>amount_payed_t-1</b>	1
<b>bill_state_t-4</b>	1
<b>bill_state_t-3</b>	1
amount_payed_t-4	2
amount_payed_t-3	3
repay_status_t-5	4
marital_status_single	5
marital_status_other	6
age	7
gender_male	8
bill_state_t-5	9
amount_payed_t-6	10
repay_status_t-4	11
repay_status_t-6	12

Table 5: Recursive Feature Elimination results.  
Source: Author's preparation

Features	p_value
<b>credit_amount</b>	0.00000
<b>age</b>	0.04990
<b>bill_state_t-1</b>	0.00110
<b>bill_state_t-2</b>	0.01342
<b>bill_state_t-3</b>	0.01422
<b>bill_state_t-4</b>	0.04912
bill_state_t-5	0.27298
bill_state_t-6	0.43766
<b>amount_payed_t-1</b>	0.00000
<b>amount_payed_t-2</b>	0.00000
<b>amount_payed_t-3</b>	0.00000
<b>amount_payed_t-4</b>	0.00000
<b>amount_payed_t-5</b>	0.00000
<b>amount_payed_t-6</b>	0.00000
<b>education_num</b>	0.00000
<b>gender_female</b>	0.00000
<b>gender_male</b>	0.00000
<b>marital_status_married</b>	0.00000
<b>marital_status_single</b>	0.00000
marital_status_other	0.91085
<b>repay_status_t-1</b>	0.00000
<b>repay_status_t-2</b>	0.00000
<b>repay_status_t-3</b>	0.00000
<b>repay_status_t-4</b>	0.00000
<b>repay_status_t-5</b>	0.00000
<b>repay_status_t-6</b>	0.00000

Table 6: ANOVA test results.  
Source: Author's preparation

Finally, the previous correlation analysis between the independent variables excluded highly correlated variables, to prevent multicollinearity. The removed features were bill\_state\_t-2, bill\_state\_t-3, bill\_state\_t-4, bill\_state\_t-5 and bill\_state\_t-6. A table that compiles each dataset's information can be found below:

Datasets	Number of Features	Number of samples in training set
Normal	26	23972
ANOVA	23	23972
RFE	15	23972
Multicollinearity	21	23972

Table 7: Different prepared datasets.  
Source: Author's preparation

After these transformations, a split was made to divide the dataset into two sets: a training set, that will be used for model training, and a test set, to evaluate the model's performance and estimate all metrics. The used ratio was 80% for training and 20% for test. To compile this whole data processing phase, a pipeline must be defined to ensure that all the input data follows the same process, to avoid data leakage.

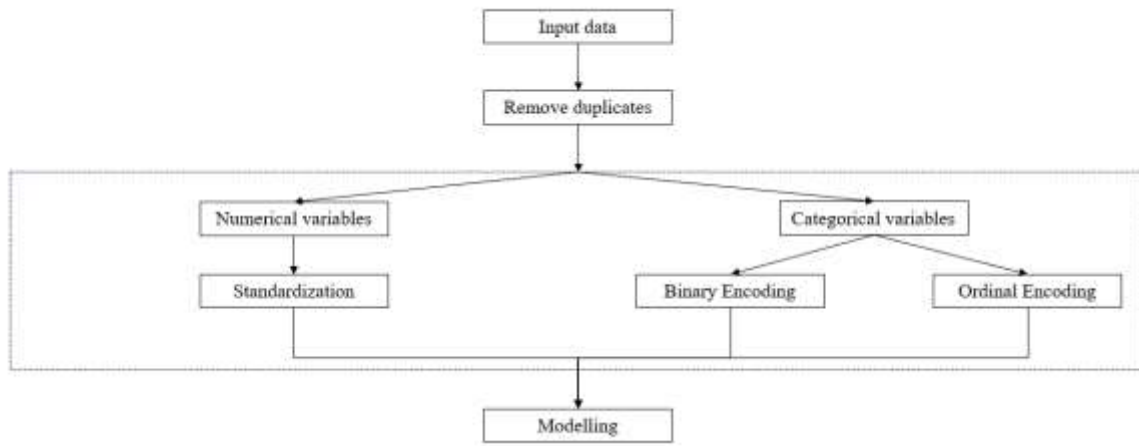


Figure 18: Data pipeline.  
Source: Author's preparation

### 4.3 Modelling

The modelling phase can be divided into two stages: hyperparameter tuning and model evaluation. Hyperparameters are external configuration variables used to control, and model structure, function, and performance. Tuning them allows users to tweak model performance for optimal results. The time spent to train and tweak each model for the five different datasets can be found below. Logistic Regression was the model that took less time to train, including hyperparameter tuning. SVM even with a limited maximum number of iterations (100000), was the model that needed more time to complete all the grid searches.

	Time spent on training (in minutes)			
	Normal	Multicollinearity	ANOVA	RFE
<b>Logistic Regression</b>	00:42	00:31	00:44	00:25
<b>Decision Tree</b>	10:57	05:53	18:33	07:26
<b>Random Forest</b>	29:10	15:37	21:42	20:59
<b>Artificial Neural Network</b>	51:56	51:16	53:04	51:17
<b>XGBoost</b>	39:35	26:05	33:48	28:07
<b>AdaBoost</b>	14:23	08:07	11:41	09:22
<b>LightGBM</b>	02:48	02:28	02:38	02:22
<b>SVM</b>	67:16	55:47	56:56	49:35

Table 8: Time spent on each model's training.  
Source: Author's preparation

There are different techniques for hyperparameter tuning as Bayesian optimization, Grid Search and Random Search. The used one was Grid Search which essentially works through all possible combinations to determine the best model. These combinations result from a list of hyperparameters specified by the user. Annex C has each hyperparameter description.

Model	Hyperparameters	Values	Grid search result
Logistic Regression	C	[1, 0.1, 0.01, 0.001, 0.0001, 0.00001, 0.000001, 0.0000001, 0.00000001]	1
	tol	[1, 0.1, 0.01, 0.001, 0.0001, 0.00001]	0.1
	max_iter	[50000]	50000
Decision Tree	criterion	['gini', 'entropy']	entropy
	max_depth	[13, 14, 15, 20, 30]	13
	min_samples_split	[1, 2, 3]	2
	min_samples_leaf	[1, 2, 3]	2
Random Forest	min_samples_split	[100, 200, 300, 400]	100
	criterion	['gini', 'entropy', 'log_loss']	entropy
Artificial Neural Network	hidden_layer_sizes	[(26,1), (52,1)]	(52,1)
	activation	['relu', 'sigmoid']	Input Layer: sigmoid Hidden Layer: relu Output Layer: sigmoid
XGBoost	max_depth	[2, 3, 4, 5, 6, 7, 8, 9]	3
	n_estimators	[60, 100, 140, 180, 220, 260]	180
	learning_rate	[0.1, 0.05, 0.01]	0.05
AdaBoost	estimator	[DecisionTreeClassifier]	DecisionTreeClassifier
	n_estimators	[50, 75, 85, 100]	50
LightGBM	boosting_type	['gbdt', 'dart']	gbdt
	objective	['binary']	binary
	n_estimators	[60, 100, 140, 180, 220, 260]	260
	learning_rate	[0.1, 0.05, 0.01]	0.01
SVM	kernel	['linear', 'poly', 'sigmoid']	poly
	tol	[0.001, 0.0001, 0.00001]	0.001
	max_iter	[100000]	100000

Table 9: Normal dataset grid search results for each model.  
Source: Author's preparation

To improve the model selection robustness, k-fold cross-validation was used to assess a model's ability to generalize to new data instances that were not considered or "seen" during training (Soper, 2021). Since it can identify issues with selection bias and overfitting, in addition to being a useful estimator of generalization performance, it has emerged as the main technique employed by ML practitioners to assess candidate models. (Soper, 2021). This technique is a crossing-over training and validation stage in successive rounds. The idea behind cross-validation is that each sample in our dataset

is tested. The number of iterations or k-folds was 10 for most models, excluding only the Artificial Neural Network.

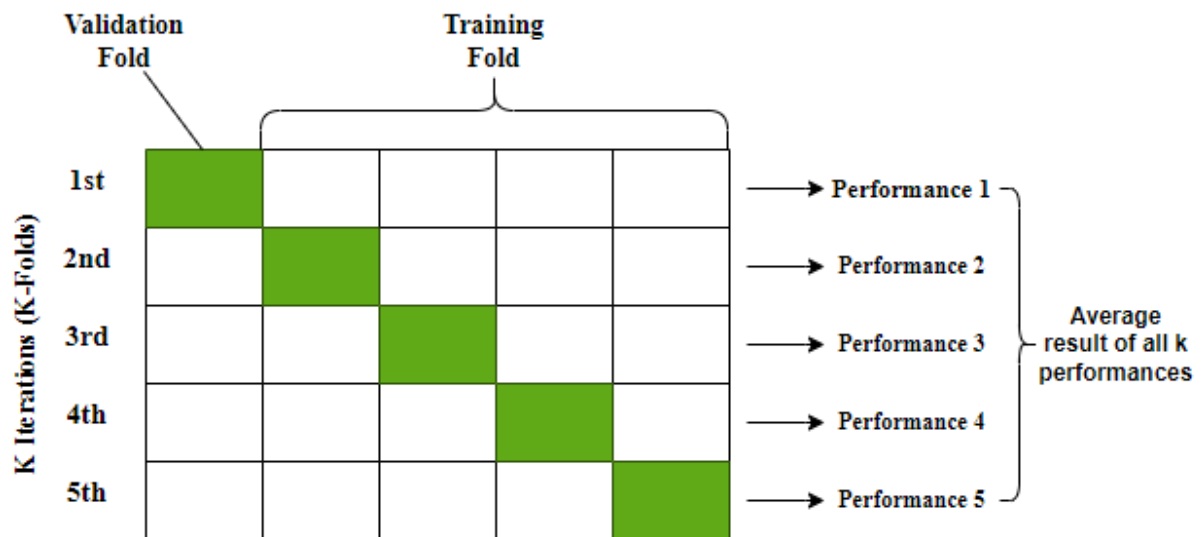


Figure 19: Cross-validation method.  
Source: Author's preparation





## CHAPTER 5

### Results and explanations

This chapter presents the results from the different techniques used for each model and the subsequent comparison between each one. After choosing the best-performing model, we apply explainable artificial intelligence (XAI) techniques to better understand the model outputs.

Model	ANOVA Dataset - Performance Metrics									
	Accuracy		Precision		Recall		F1-Score		AUC	
	Training	Testing	Training	Testing	Training	Testing	Training	Testing	Training	Testing
Logistic Regression	0.8094	0.8160	0.7148	<b>0.7079</b>	0.2407	0.2446	0.3602	0.3635	0.7218	0.7310
Linear Discriminant Analysis	0.8101	0.8173	0.7028	0.6982	0.2559	0.2640	0.3752	0.3831	0.7172	0.7239
Decision Tree	0.8649	0.8026	0.8043	0.5617	0.5202	0.3711	0.6318	0.4469	0.8824	0.6997
Random Forest	0.8315	<b>0.8298</b>	0.7270	0.6872	0.3903	0.3820	0.5079	<b>0.4910</b>	0.8868	0.7894
Artificial Neural Network	0.8482	0.8024	0.7501	0.5607	0.4777	0.3727	0.5837	0.4478	0.8554	0.7456
XGBoost	0.8258	0.8286	0.7107	0.6905	0.3684	0.3672	0.4853	0.4795	0.8024	0.7885
AdaBoost	<b>0.9994</b>	0.7422	<b>1.0000</b>	0.4006	<b>0.9974</b>	<b>0.4022</b>	<b>0.9987</b>	0.4014	<b>1.0000</b>	0.6779
LightGBM	0.8244	0.8283	0.7132	0.6977	0.3547	0.3548	0.4738	0.4704	0.8211	<b>0.7917</b>
SVM	0.8230	0.8253	0.7222	0.6972	0.3343	0.3307	0.4571	0.4487	0.7327	0.7127

Table 10: Performance metrics for ANOVA dataset.  
Source: Author's preparation

Above we can observe metrics for both the training and testing stages for each model applied to the ANOVA-adjusted dataset since it was where the best Area under the Curve (AUC) value was found. The remaining results can be found on the Annex D, E, F, for Normal, RFE and Multicollinearity datasets, respectively. Logistic Regression obtained one winner metric in the testing phase, with a precision of 70.79%. Linear Discriminant Analysis didn't achieve any winner metric for both phases. Using AUC to compare all models, SVM, Decision Tree, LDA, and AdaBoost lose to the best performing traditional model, Logistic Regression. Although AdaBoost achieved the best metrics for the training phase, his performance wasn't quite consistent with the testing phase, showing an overfit to the training data. Analysing the training stage, AdaBoost achieved the best accuracy (99.94%), precision (100%), recall (99.74%), F1-Score (99.87%), and AUC (100%).

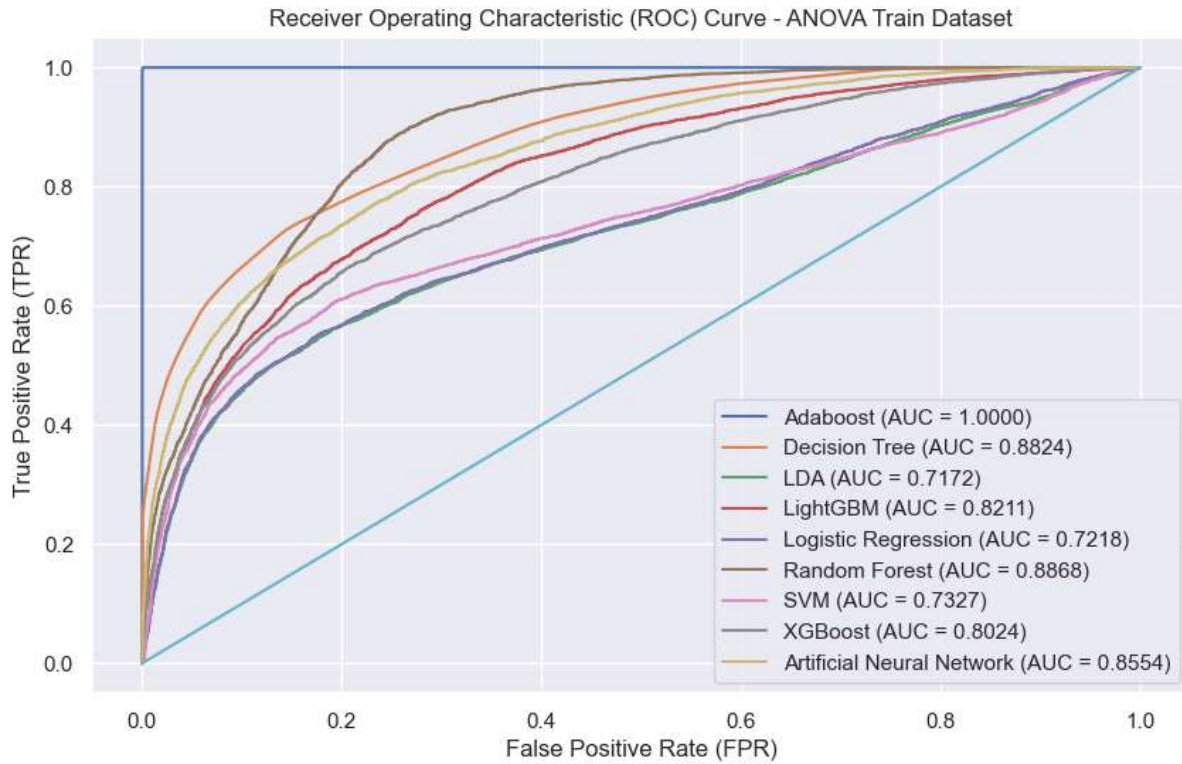


Figure 20: ROC curves for ANOVA dataset training phase.  
Source: Author's preparation

Nevertheless, the research will focus more on the performance metrics that resulted from applying these models to out-of-sample entries. The developed LightGBM despite losing to Random Forest on Accuracy and F1-Score, to Logistic Regression on Precision, to AdaBoost on Recall, wins on AUC with the highest score of 79.17%.

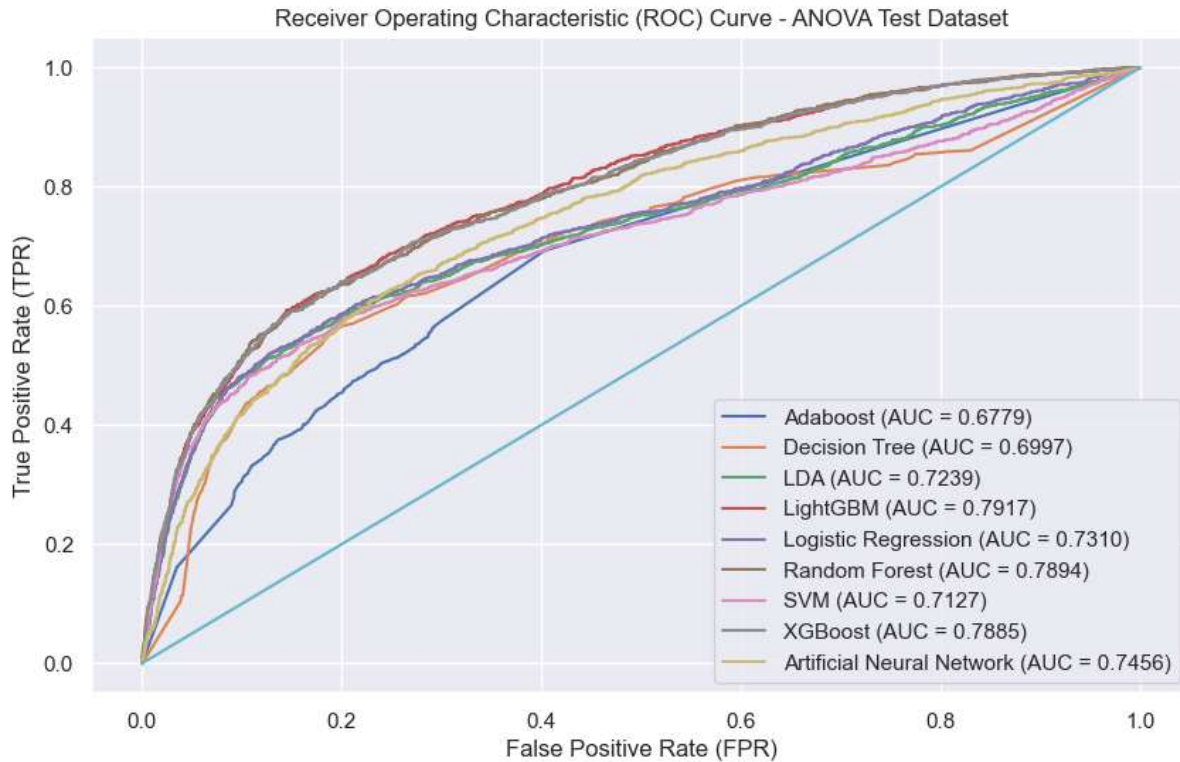


Figure 21: ROC curves for ANOVA dataset testing phase.  
Source: Author's preparation

In addition, a DeLong test was conducted to confirm if the differences between AUCs are statistically significant, under the null hypothesis that both areas are equal.

Comparison of AUCs (Logistic Regression vs)	z-score	p-value
<b>AdaBoost</b>	6.09656	0.00000
<b>Decision Tree</b>	2.40889	0.01600
<b>LightGBM</b>	-9.23158	0.00000
<b>Random Forest</b>	-8.97982	0.00000
<b>SVM</b>	2.14919	0.03162
<b>XGBoost</b>	-8.94304	0.00000
<b>Artificial Neural Network</b>	-0.67457	0.49995

Table 11: DeLong et al. (1988) test to compare AUCs.  
Source: Author's preparation

Being a two-tailed test, there are two types of differences. If the value is on the positive tail both areas are significantly different, but Logistic Regression is better. If the value is on the negative tail both areas are significantly different, but the ML model is better. LightGBM still shows a better performance compared with Logistic Regression, with XGBoost and Random Forest being the two next-best models. On the other hand, AdaBoost is the worse model, with Decision Tree and SVM being considerably better, but still all three are worse than Logistic Regression. Artificial Neural Network

wasn't significantly different from the Logistic Regression. Overall, the test corroborates the results given in Figure 21.

Another DeLong test was used to compare two models, one with all variables and another with only a subset of variables (a nested model). The purpose is to evaluate if the different feature selection techniques improved the model's performance compared to a full model.

Comparison of AUCs (Normal dataset vs)	Logistic Regression		LightGBM	
	z-score	p-value	z-score	p-value
<b>ANOVA</b>	-1.34618	0.17825	-0.95259	0.34080
<b>Multicollinearity</b>	1.87347	0.06100	0.05230	0.95829
<b>RFE</b>	0.13169	0.89523	2.99613	0.00273

Table 12: DeLong et al. (1988) test to compare AUCs.  
Source: Author's preparation

As in the previous test, the results seem to confirm the results from the different cross-validation processes, when comparing the AUC values in Annex D with Table 10, Annex E, and Annex F. Nevertheless, most of the differences don't seem statistically significant, except for LightGBM-RFE, with a decrease in performance when compared with the LightGBM-Normal model. With LightGBM-ANOVA being the best-performing model, next an analysis of this model's structure and outputs is done. For a 0.5 threshold, comparing the results with the actual default status of each borrower, we can generate the below confusion matrix. The model predicted 655 defaults and 5,338 non-defaults versus 1,288 observed defaults and 4,705 observed non-defaults. It seems that many defaults were wrongly labelled as non-defaults (831), meaning that a considerable amount of bad-quality loans would have been granted (13.87%).

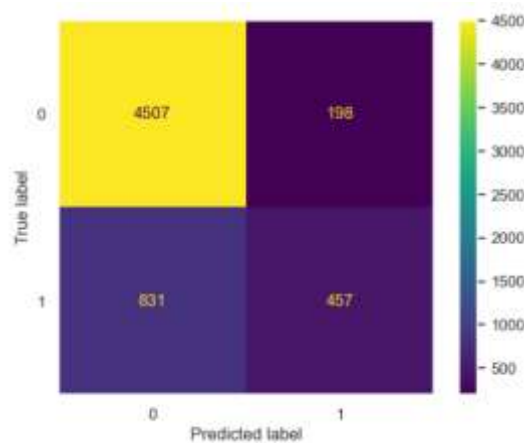


Figure 22: LightGBM-ANOVA confusion matrix on test data.  
Source: Author's preparation

Analysing the accuracy for different thresholds, a maximum value is reached at a 0.47 cut-off.

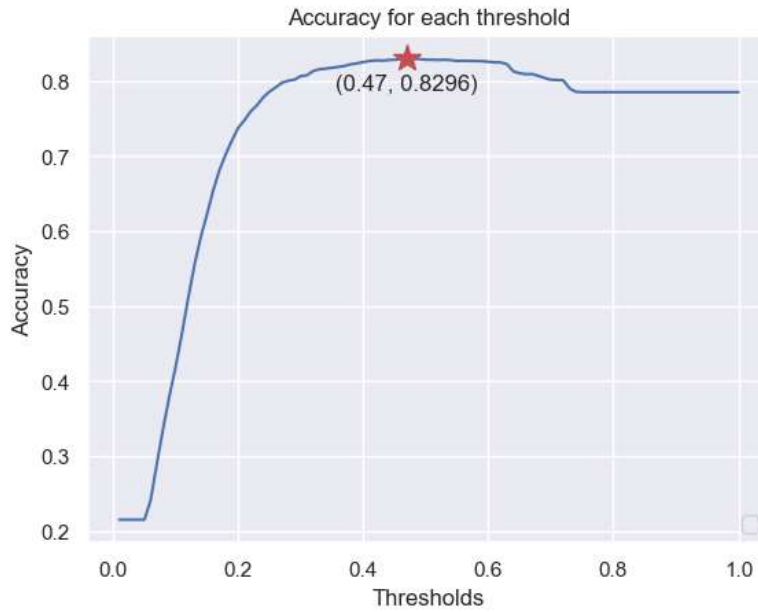


Figure 23: LightGBM-ANOVA accuracy for different thresholds.  
Source: Author's preparation

## 5.1 Shapley values

SHAP values show how each feature affects every final prediction, the significance of each feature compared to others, and the model's reliance on the interaction between features. Features with positive SHAP values positively impact the prediction or, in other words, contribute more to default. On the other hand, features with negative SHAP values contribute more to a non-default.

Below we can find a non-default predicted by the model. Since the data was scaled, the values diverged from the real ones. Three different "repay\_status" with -1 means that the borrower fully paid all the contracted credit from the past three months. Reinforcing this idea are the "amount\_paid" variables that with a positive value show the good quality of this borrower.

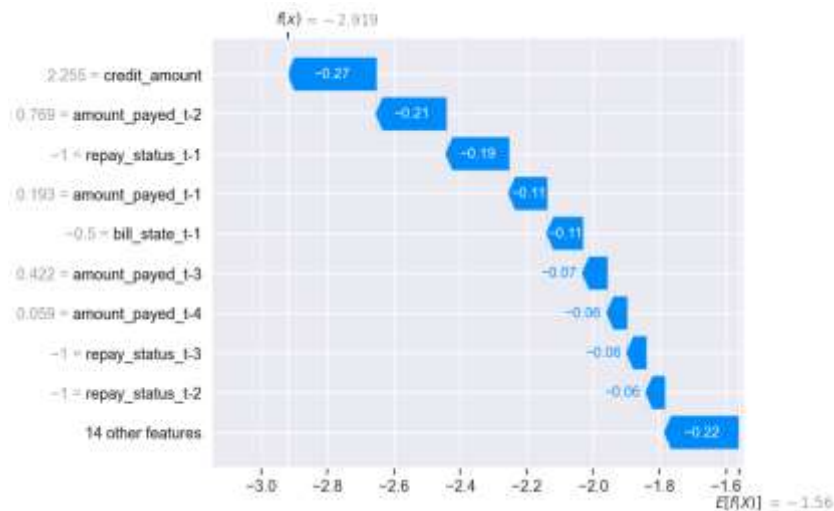


Figure 24: SHAP explanation for a non-default prediction.  
Source: Author's preparation

On the other hand, below is a default example predicted by the model. Six “repay\_status” that show a structural delay on due payments. For 5 straight months (t-6 to t-2), this borrower maintained a 2-month due payment, but on t-1 another payment was left overdue, and it defaulted on the next month.

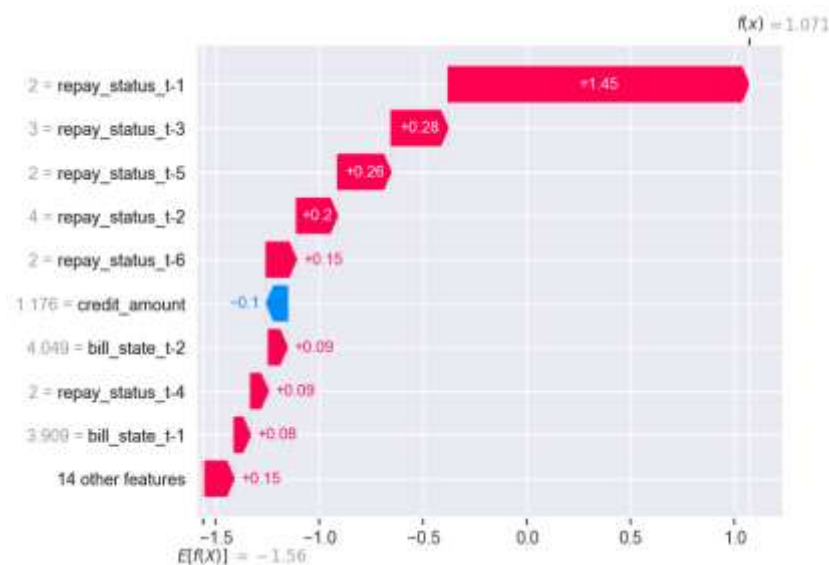


Figure 25: SHAP explanation for a default prediction.  
Source: Author's preparation

## 5.2 LIME

The LIME technique further explains and interprets the prediction of an instance, for an individual borrower.

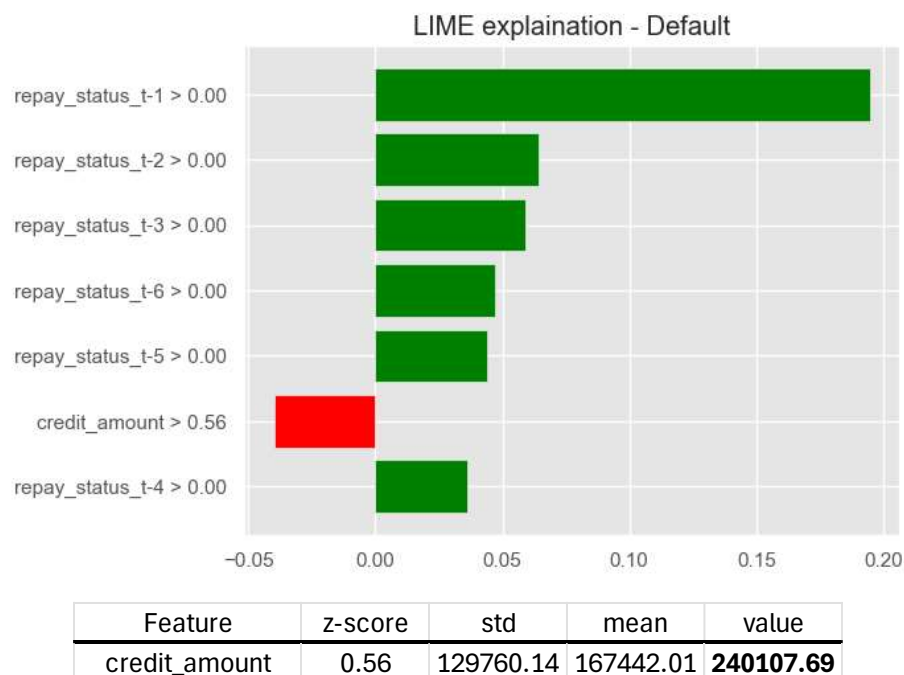


Figure 26: LIME explanation for a default prediction.  
Source: Author's preparation

The above figure shows the same default example chosen for the SHAP analysis but gives us additional insights into that prediction. From the LIME explanation generated, we can observe that “repay\_status” that is higher than 0 and a credit\_amount lower or equal to 240,107.70 are characteristics of a default example. Taking the same non-default example, the LIME technique gives us the results below:

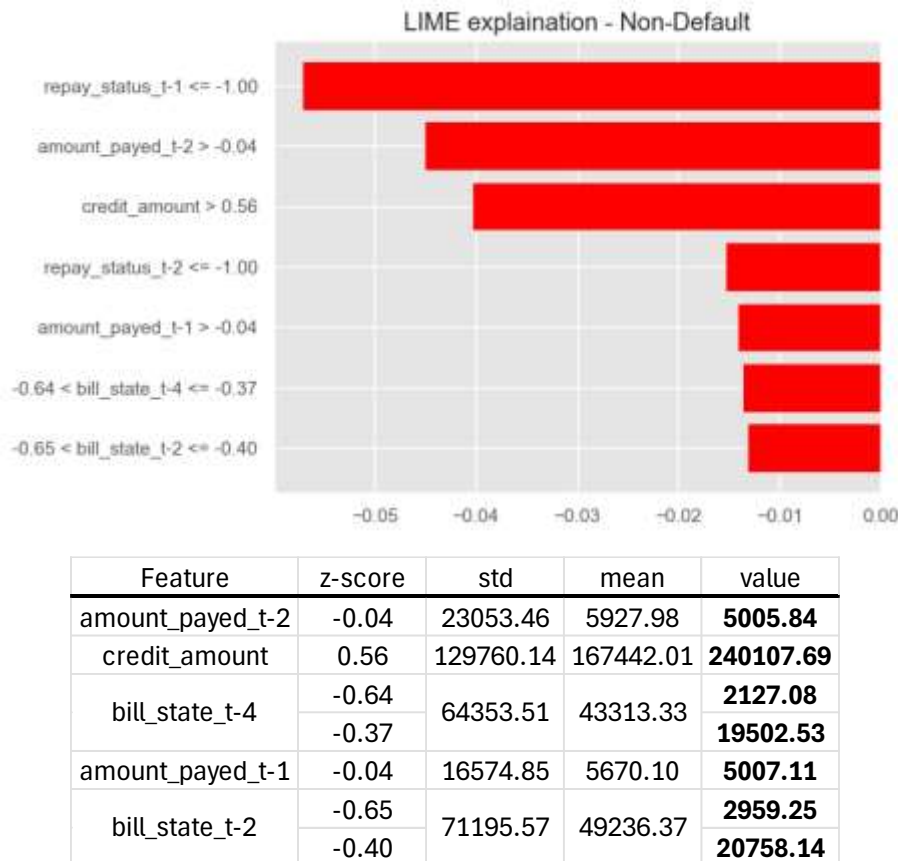


Figure 27: LIME explanation for a non-default prediction.  
Source: Author's preparation

We can observe that a “repay\_status” variable value below or equal to -1 contributed to a non-default. Observing the remaining continuous variables, both “amount\_paid” variables' values are positive and near mid-distribution by analysing their z-score. A higher “credit\_amount” than 240,107.69 seems to be related to non-default reinforcing the previous SHAP analysis that lower “credit\_amount” borrowers were more related to defaults. Finally, “bill\_state\_t-2” and “bill\_state\_t-4” with a value between 2,959.25 and 20,758.14, and 2,127.08 and 19,502.53, respectively, contributed to the non-default prediction.





## CHAPTER 6

### **Conclusions and recommendations for future work**

Credit scoring techniques are a risk management tool that allows a lender to assess an applicant's capacity to repay a loan on schedule. The many developments over the years aided financial institutions by handling loan applications far more quickly, with lower costs. More traditional techniques are still broadly used, including Logistic Regression, for their prevalence on this field. Nevertheless, the latest developments in data science, such as machine learning techniques on feature selection and modelling can deliver better results with a cost on interpretability. Regulators often release papers and recommendations on the usage of these artificial intelligence methods for IRB modelling to alert for different cautions to have when applying them, mainly on model knowledge and results interpretation. Luckily, different Machine Learning enthusiasts have developed techniques that help with this interpretability issue such as LIME and SHAP.

The main objective of this work is to compare these two techniques for the probability of default estimation. Since the bank's default policy is unknown, it was difficult to perform a more in-depth analysis on each class. The conducted data exploration was mainly useful to select features that would result on a model's performance improvement, but not to conclude on any economic/financial aspect of each variable. The implementation of many transformations, including handling missing data, encoding categorical variables, and appropriate scaling, was made easier by the use of machine learning pipelines. Furthermore, cross-validation, hyperparameter tuning, and statistical tests to AUC were conducted to make sure the classifiers were robust. The results show that LightGBM model is comparatively superior in performance to all other models, including Logistic Regression. However, when using LightGBM some considerations need to be made, when comparing with Logistic Regression: reduction of model interpretability and increase in computation time. To overcome this model interpretability issue, two techniques were employed to explain the model's results globally and locally. Pairing with the unknown default policy, the used hardware can be pointed out as another limitation of this work. While training models using cross-validation grid searches, a better computation power will lead to faster modelling and a better hyperparameter selection. This would result in more accurate models, that could extract more insights from data.

For upcoming studies, applying this same methodology to a different credit scoring dataset could help to improve the robustness of this work's findings. Due to the class imbalance commonly present in the credit scoring domain, Chen et al. (2024) analysed this effect on SHAP and LIME methodologies with different stability measures such as sequential rank agreement, coefficient of variance, variables stability index, and coefficient stability index. Employing their methodology may result in more robust explanations.

## References

- Abdou, H., Pointon, J., & El-Masry, A. (2008). Neural nets versus conventional techniques in credit scoring in Egyptian banking. *Expert systems with applications*, 35(3), 1275-1292.
- Al-Aradi, A. (2014). Credit Scoring via Logistic Regression. Department of Statistical Sciences, University of Toronto, Toronto, Canada.
- Altman, E. I. (1968). Financial ratios, discriminant analysis and the prediction of corporate bankruptcy. *The journal of finance*, 23(4), 589-609.
- Altman, E. I. (2018). A fifty-year retrospective on credit risk models, the Altman Z-score family of models and their applications to financial markets and managerial strategies. *Journal of Credit Risk*, 14(4).
- Ashofteh, A., & Bravo, J. M. (2021). A conservative approach for online credit scoring. *Expert Systems with Applications*, 176, 114835.
- Barbaglia, L., Manzan, S., & Tosetti, E. (2023). Forecasting loan default in Europe with machine learning. *Journal of Financial Econometrics*, 21(2), 569-596.
- Barboza, F., Kimura, H., & Altman, E. (2017). Machine learning models and bankruptcy prediction. *Expert Systems with Applications*, 83, 405-417.
- Basel Committee on Banking Supervision (2000). Principles for the management of credit risk. <https://www.bis.org/publ/bcbssc125.pdf>
- Beaver, W. H. (1966). Financial ratios as predictors of failure. *Journal of accounting research*, 71-111.
- Bekkar, M., Djemaa, H. K., & Alitouche, T. A. (2013). Evaluation measures for models assessment over imbalanced data sets. *Journal of Information Engineering and Applications*, 3(10).
- Bensic, M., Sarlija, N., & Zekic-Susac, M. (2005). Modelling small-business credit scoring by using logistic regression, neural networks and decision trees. *Intelligent Systems in Accounting, Finance & Management: International Journal*, 13(3), 133-150.
- Breiman, L. (1996). Bagging predictors. *Machine learning*, 24, 123-140.
- Brown, I., & Mues, C. (2012). An experimental comparison of classification algorithms for imbalanced credit scoring data sets. *Expert systems with applications*, 39(3), 3446-3453.
- Bücker, M., Szepannek, G., Gosiewska, A., & Biecek, P. (2022). Transparency, auditability, and explainability of machine learning models in credit scoring. *Journal of the Operational Research Society*, 73(1), 70-90.
- Chan, J. Y. L., Leow, S. M. H., Bea, K. T., Cheng, W. K., Phoong, S. W., Hong, Z. W., & Chen, Y. L. (2022). Mitigating the multicollinearity problem and its machine learning approach: a review. *Mathematics*, 10(8), 1283.
- Chen, N., Ribeiro, B., & Chen, A. (2016). Financial credit risk assessment: a recent review. *Artificial Intelligence Review*, 45, 1-23.
- Chen, T., & Guestrin, C. (2016, August). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining* (pp. 785-794).
- Chen, Y., Calabrese, R., & Martin-Barragan, B. (2024). Interpretable machine learning for imbalanced credit scoring datasets. *European Journal of Operational Research*, 312(1), 357-372.
- Chen, Z., Xiao, F., Guo, F., & Yan, J. (2023). Interpretable machine learning for building energy management: A state-of-the-art review. *Advances in Applied Energy*, 9, 100123.

- Cilimkovic, M. (2015). Neural networks and back propagation algorithm. Institute of Technology Blanchardstown, Blanchardstown Road North Dublin, 15(1).
- Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine learning*, 20, 273-297.
- Darst, B. F., Malecki, K. C., & Engelman, C. D. (2018). Using recursive feature elimination in random forest to account for correlated variables in high dimensional data. *BMC genetics*, 19, 1-6.
- DeMaris, A. (1995). A tutorial in logistic regression. *Journal of Marriage and the Family*, 956-968.
- Ding, H., Feng, P. M., Chen, W., & Lin, H. (2014). Identification of bacteriophage virion proteins by the ANOVA feature selection and analysis. *Molecular BioSystems*, 10(8), 2229-2235.
- European Banking Authority (2020). Report on Big Data and advanced analytics. [https://www.eba.europa.eu/sites/default/files/document\\_library/Final%20Report%20on%20Big%20Data%20and%20Advanced%20Analytics.pdf](https://www.eba.europa.eu/sites/default/files/document_library/Final%20Report%20on%20Big%20Data%20and%20Advanced%20Analytics.pdf)
- European Banking Authority (2021). Analysis of Regtech in the EU financial sector. [https://www.eba.europa.eu/sites/default/files/document\\_library/Publications/Reports/2021/1015484/EBA%20analysis%20of%20RegTech%20in%20the%20EU%20financial%20sector.pdf](https://www.eba.europa.eu/sites/default/files/document_library/Publications/Reports/2021/1015484/EBA%20analysis%20of%20RegTech%20in%20the%20EU%20financial%20sector.pdf)
- European Banking Authority (2021). Discussion paper on machine learning for IRB models. [https://www.eba.europa.eu/sites/default/files/document\\_library/Publications/Discussions/2022/Discussion%20on%20machine%20learning%20for%20IRB%20models/1023883/Discussion%20paper%20on%20machine%20learning%20for%20IRB%20models.pdf](https://www.eba.europa.eu/sites/default/files/document_library/Publications/Discussions/2022/Discussion%20on%20machine%20learning%20for%20IRB%20models/1023883/Discussion%20paper%20on%20machine%20learning%20for%20IRB%20models.pdf)
- European Council (2017). Council conclusions on Action plan to tackle non-performing loans in Europe. <https://www.consilium.europa.eu/en/press/press-releases/2017/07/11/conclusions-non-performing-loans/pdf>.
- Financial Stability Board (2017). Artificial intelligence and machine learning in financial services - Market developments and financial stability implications. <https://www.fsb.org/wp-content/uploads/P011117.pdf>
- Folpmers, M., Kool, R., Goos, A., Prins, C., Hottenhuis, W., Meulenbelt, I. (2023). The application of machine learning and challenger models in IRB Credit Risk modelling - The use in model estimation. <https://www2.deloitte.com/content/dam/Deloitte/nl/Documents/risk/deloitte-nl-risk-challenger-models-model-estimation.pdf>
- Gedela, B., & Karthikeyan, P. R. (2022, February). Credit card fraud detection using AdaBoost algorithm in comparison with various machine learning algorithms to measure accuracy, sensitivity, specificity, precision and F-score. In 2022 International Conference on Business Analytics for Technology and Security (ICBATS) (pp. 1-6). IEEE.
- Goh, R. Y., & Lee, L. S. (2019). Credit scoring: a review on support vector machines and metaheuristic approaches. *Advances in Operations Research*, 2019.
- Gokcen, I., & Peng, J. (2002, October). Comparing linear discriminant analysis and support vector machines. In International Conference on advances in information systems (pp. 104-113). Berlin, Heidelberg: Springer Berlin Heidelberg.
- Hand, D. J., & Henley, W. E. (1997). Statistical classification methods in consumer credit scoring: a review. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 160(3), 523-541.
- Hoo, Z. H., Candlish, J., & Teare, D. (2017). What is an ROC curve?. *Emergency Medicine Journal*, 34(6), 357-359.

- Hosmer Jr, D. W., Lemeshow, S., & Sturdivant, R. X. (2013). *Applied logistic regression*. John Wiley & Sons.
- Hu, W., Hu, W., & Maybank, S. (2008). Adaboost-based algorithm for network intrusion detection. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 38(2), 577-583.
- Huang, C. L., Chen, M. C., & Wang, C. J. (2007). Credit scoring with a data mining approach based on support vector machines. *Expert systems with applications*, 33(4), 847-856.
- Hussin Adam Khatir, A. A., & Bee, M. (2022). Machine learning models and data-balancing techniques for credit scoring: What is the best combination?. *Risks*, 10(9), 169.
- Jović, A., Brkić, K., & Bogunović, N. (2015, May). A review of feature selection methods with applications. In *2015 38th international convention on information and communication technology, electronics and microelectronics (MIPRO)* (pp. 1200-1205).
- Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., ... & Liu, T. Y. (2017). Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems*, 30.
- Li, H., Cao, Y., Li, S., Zhao, J., & Sun, Y. (2020). XGBoost model and its application to personal credit evaluation. *IEEE Intelligent Systems*, 35(3), 52-61.
- Lipton, Z. C., Elkan, C., & Naryanaswamy, B. (2014). Optimal thresholding of classifiers to maximize F1 measure. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2014, Nancy, France, September 15-19, 2014. Proceedings, Part II 14* (pp. 225-239). Springer Berlin Heidelberg.
- Liu, Y., Wang, Y., & Zhang, J. (2012). New machine learning algorithm: Random forest. In *Information Computing and Applications: Third International Conference, ICICA 2012, Chengde, China, September 14-16, 2012. Proceedings 3* (pp. 246-252). Springer Berlin Heidelberg.
- Lundberg, S. M., & Lee, S. I. (2017). A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30.
- Madge, S., & Bhatt, S. (2015). Predicting stock price direction using support vector machines. Retrieved from [https://www.cs.princeton.edu/sites/default/files/uploads/saahil\\_madge.pdf](https://www.cs.princeton.edu/sites/default/files/uploads/saahil_madge.pdf)
- Martin, D. (1977). Early warning of bank failure: A logit regression approach. *Journal of Banking & Finance*, 1(3), 249-276.
- Mitchell, R., & Frank, E. (2017). Accelerating the XGBoost algorithm using GPU computing. *PeerJ Computer Science*, 3, e127.
- Moscato, V., Picariello, A., & Sperlí, G. (2021). A benchmark of machine learning approaches for credit score prediction. *Expert Systems with Applications*, 165, 113986.
- Ohlson, J. A. (1980). Financial ratios and the probabilistic prediction of bankruptcy. *Journal of accounting research*, 109-131.
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, É. (2011). Scikit-learn: Machine learning in Python. *the Journal of machine Learning research*, 12, 2825-2830.
- Ponsam, J. G., Gracia, S. J. B., Geetha, G., Karpaservi, S., & Nimala, K. (2021, December). Credit Risk Analysis using LightGBM and a comparative study of popular algorithms. In *2021 4th International Conference on Computing and Communications Technologies (ICCCT)* (pp. 634-641). IEEE.
- Potdar, K., Pardawala, T. S., & Pai, C. D. (2017). A comparative study of categorical variable encoding techniques for neural network classifiers. *International journal of computer applications*, 175(4), 7-9.

- Ribeiro, M. T., Singh, S., & Guestrin, C. (2016, August). " Why should i trust you?" Explaining the predictions of any classifier. In Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining (pp. 1135-1144).
- Soper, D. S. (2021). Greed is good: Rapid hyperparameter optimization and model selection using greedy k-fold cross validation. *Electronics*, 10(16), 1973.
- Stroie, L. M. B. (2013). Techniques for customer behaviour prediction: A case study for credit risk assessment. In Proceedings of International Conference on New Techniques and Technologies for Statistics (pp. 685-694).
- The World Bank Group (2019). Credit Scoring approaches guidelines. <https://thedocs.worldbank.org/en/doc/935891585869698451-0130022020/original/CREDITSCORINGAPPROACHESGUIDELINESFINALWEB.pdf>
- Zavgren, C.V. (1985). Assessing the vulnerability to failure of American industrial firms: A logistic analysis. *Journal of Business Finance & Accounting*, 12, 19-45."

## Annex

### Annex A

Variable	Description
credit_amount	Amount of the given credit
gender	Borrower's gender: Male or Female
education	Borrower's habilitation: Masters/phd, University, High School, Others
age	Borrower's age
repay_status_t-1 to t-6	History of past payments for each month -2: No consumption; -1: Paid in full; 0: The use of revolving credit; 1-9: payment delay (months)
bill_state_t-1 to t-6	Amount of bill statement
amount_payed_t-1 to t-6	Amount of previous payment

## Annex B

	Features													
Statistic	age	credit_amount	bill_state_t-1	bill_state_t-2	bill_state_t-3	bill_state_t-4	bill_state_t-5	bill_state_t-6	amount_payed_t-1	amount_payed_t-2	amount_payed_t-3	amount_payed_t-4	amount_payed_t-5	amount_payed_t-6
count	29965	29965	29965	29965	29965	29965	29965	29965	29965	29965	29965	29965	29965	29965
mean	35.49	167442.01	51283.01	49236.37	47067.92	43313.33	40358.33	38917.01	5670.10	5927.98	5231.69	4831.62	4804.90	5221.5
std	9.22	129760.14	73658.13	71195.57	69371.35	64353.51	60817.13	60817.13	16574,85	23053.46	17616.36	15674.46	15286.37	17786.98
min	21	10000	-165580	-69777	-157264	-170000	-81334	-81334	0	0	0	0	0	0
25%	28	50000	3595	3010	2711	2360	1787	1787	1000	850	390	300	261	131
50%	34	140000	22438	21295	20135	19081	18130	18130	2102	2010	1804	1500	1500	1500
75%	41	240000	67260	64109	60201	54601	50247	50247	5008	5000	4512	4016	4042	4000
max	79	1000000	964511	983931	1664089	891586	927171	927171	873552	1684259	896040	621000	426529	528666

## Annex C

Hyperparameter	Definition
<b>C</b>	Represents the inverse of regularization strength. Used to prevent overfitting by adding a penalty term to the model's loss function. Smaller value means a simpler model, and a large value suggests a more complex model.
<b>max_iter</b>	Used to control the maximum number of iterations for the optimization algorithm
<b>solver</b>	Specifies the method used to compute the shrinkage regularization, which is used to estimate the covariance matrix of the data.
<b>criterion</b>	Specifies the criterion used to select the dimensionality of the reduced space.
<b>max_depth</b>	Determines the maximum depth of the tree. The depth of a tree is the length of the longest path from the root node to a leaf node. Limiting the maximum depth of a decision tree helps to prevent overfitting.
<b>min_samples_split</b>	The minimum number of samples required to split an internal node. When the number of samples in a node is less than this value, the node will not be split, and it will become a leaf node.
<b>min_samples_leaf</b>	Helps to prevent overfitting by controlling the minimum size of the leaf nodes.
<b>hidden_layer_sizes</b>	Defines the neural network structure (excludes input layer).
<b>activation</b>	Defines the output of a neuron based on its input by applying this function
<b>n_estimators</b>	Controls the maximum number of weak learners to train.
<b>learning_rate</b>	Control the trade-off between bias and variance and prevent overfitting. Lower values typically require more trees to reach the same level of performance, but they generalize better. Higher values can lead to faster learning but make the model more prone to overfitting
<b>estimator</b>	The model used as weak learner
<b>boosting_type</b>	Specifies the type of boosting algorithm to be used to build the ensemble model.
<b>objective</b>	Specifies the loss function that the model will optimize during training.
<b>kernel</b>	Specifies the kernel function used to transform the input features into a higher-dimensional space, where data may be more easily separated by a hyperplane.



## Annex D

Model	Normal Dataset - Performance Metrics									
	Accuracy		Precision		Recall		F1-Score		AUC	
	Training	Testing	Training	Testing	Training	Testing	Training	Testing	Training	Testing
Logistic Regression	0.8097	0.8155	0.7165	0.7059	0.2413	0.2422	0.3610	0.3607	0.7218	0.7304
Decision Tree	0.8649	0.8014	0.8028	0.5565	0.5219	0.3750	0.6326	0.4481	0.8824	0.7066
Random Forest	0.8336	0.8285	0.7354	0.6831	0.3954	0.3766	0.5142	0.4855	0.8982	0.7917
Artificial Neural Network	0.8551	0.7963	0.7834	0.5401	0.4835	0.3502	0.5980	0.4249	0.8710	0.7359
XGBoost	0.8239	0.8285	0.6997	0.6852	0.3673	0.3734	0.4817	0.4834	0.8008	0.7877
AdaBoost	0.9994	0.7329	1.0000	0.3865	0.9974	0.4138	0.9987	0.3997	1.0000	0.6731
LightGBM	0.8248	0.8283	0.7156	0.6989	0.3551	0.3533	0.4747	0.4693	0.8219	0.7913
SVM	0.8227	0.8251	0.7217	0.6993	0.3325	0.3269	0.4552	0.4455	0.7365	0.7166

## Annex E

Model	Multicollinearity Dataset - Performance Metrics									
	Accuracy		Precision		Recall		F1-Score		AUC	
	Training	Testing	Training	Testing	Training	Testing	Training	Testing	Training	Testing
Logistic Regression	0.8092	0.8143	0.7164	0.7002	0.2383	0.2376	0.3576	0.3548	0.7210	0.7290
Decision Tree	0.8586	0.8049	0.7952	0.5726	0.4921	0.3641	0.6080	0.4452	0.8756	0.7124
Random Forest	0.8253	0.8301	0.7074	0.6945	0.3680	0.3742	0.4842	0.4864	0.8464	0.7892
Artificial Neural Network	0.8433	0.8113	0.7367	0.5942	0.4620	0.3843	0.5679	0.4668	0.8436	0.7533
XGBoost	0.8253	0.8285	0.7082	0.6906	0.3671	0.3657	0.4835	0.4782	0.8000	0.7885
AdaBoost	0.9994	0.7342	1.0000	0.3873	0.9974	0.4068	0.9987	0.3968	1.0000	0.6673
LightGBM	0.8248	0.8283	0.7150	0.6977	0.3555	0.3548	0.4749	0.4704	0.8185	0.7913
SVM	0.8225	0.8250	0.7180	0.6900	0.3351	0.3370	0.4569	0.4528	0.7327	0.7122

## Annex F

Model	RFE Dataset - Performance Metrics									
	Accuracy		Precision		Recall		F1-Score		AUC	
	Training	Testing	Training	Testing	Training	Testing	Training	Testing	Training	Testing
Logistic Regression	0.8089	0.8136	0.7121	0.6939	0.2389	0.2376	0.3577	0.3540	0.7217	0.7302
Decision Tree	0.8524	0.8109	0.7460	0.5856	0.5118	0.4115	0.6071	0.4834	0.8671	0.7195
Random Forest	0.8251	0.8295	0.7163	0.6991	0.3564	0.3626	0.4760	0.4775	0.8497	0.7845
Artificial Neural Network	0.8292	0.8176	0.7172	0.6352	0.3854	0.3556	0.5014	0.4559	0.8107	0.7722
XGBoost	0.8230	0.8280	0.7053	0.6909	0.3531	0.3610	0.4706	0.4742	0.7911	0.7826
AdaBoost	0.9946	0.7369	0.9960	0.3901	0.9798	0.3983	0.9878	0.3942	0.9999	0.6708
LightGBM	0.8257	0.8276	0.7140	0.6872	0.3635	0.3634	0.4818	0.4754	0.8146	0.7840
SVM	0.8175	0.8213	0.7097	0.6900	0.3066	0.3059	0.4282	0.4239	0.7049	0.7061