NOVA IMS Information Management School

MDSAA

Master Degree Program in

Data Science and Advanced Analytics

Machine Learning applied to credit risk assessment: Prediction of loan defaults

Sofia Beatriz Santos Simão

Dissertation

presented as partial requirement for obtaining the Master Degree Program in Data Science and Advanced Analytics

NOVA Information Management School

Instituto Superior de Estatística e Gestão de Informação

Universidade Nova de Lisboa

NOVA Information Management School

Instituto Superior de Estatística e Gestão de Informação

Universidade Nova de Lisboa

MACHINE LEARNING APPLIED TO CREDIT RISK ASSESSMENT: PREDICTION OF LOAN DEFAULTS

by

Sofia Beatriz Santos Simão

Dissertation presented as partial requirement for obtaining the Master's degree in Data Science and Advanced Analytics, with a Specialization in Data Science

Supervisor : Professor *Dr.* Mauro Castelli

November 2022

STATEMENT OF INTEGRITY

I hereby declare having conducted this academic work with integrity. I confirm that I have not used plagiarism or any form of undue use of information or falsification of results along the process leading to its elaboration. I further declare that I have fully acknowledge the Rules of Conduct and Code of Honor from the NOVA Information Management School.

Lisboa, Novembro 2022

ACKNOWLEDGEMENTS

First I would like to thank everyone that helped and accompanied me not only during this last year of writing the thesis but throughout the entire master's degree. I would especially like to express my gratitude to Professor Doctor Mauro Castelli, not only for accepting to be my supervisor but also for his feedback and support. I would also like to thank my parents for being supportive and for all the efforts they made so it would be possible for me to complete my education. Finally, I want to thank my friends, in particular Joana Calisto and Miguel Martins, who were a great support. Thank you for all the patience, motivation and words of advice.

ABSTRACT

Due to the recent financial crisis and regulatory concerns of Basel II, credit risk assessment is becoming a very important topic in the field of financial risk management. Financial institutions need to take great care when dealing with consumer loans in order to avoid losses and costs of opportunity. For this matter, credit scoring systems have been used to make informed decisions on whether or not to grant credit to clients who apply to them. Until now several credit scoring models have been proposed, from statistical models, to more complex artificial intelligence techniques. However, most of previous work is focused on employing single classifiers. Ensemble learning is a powerful machine learning paradigm which has proven to be of great value in solving a variety of problems. This study compares the performance of the industry standard, logistic regression, to four ensemble methods, i.e. AdaBoost, Gradient Boosting, Random Forest and Stacking in identifying potential loan defaults. All the models were built with a real world dataset with over one million customers from Lending Club, a financial institution based in the United States. The performance of the models was compared by using the Hold-out method as the evaluation design and accuracy, AUC, type I error and type II error as evaluation metrics. Experimental results reveal that the ensemble classifiers were able to outperform logistic regression on three key indicators, i.e. accuracy, type I error and type II error. AdaBoost performed better than the remaining classifiers considering a trade off between all the metrics evaluated. The main contribution of this thesis is an experimental addition to the literature on the preferred models for predicting potential loan defaulters.

KEYWORDS

Credit Risk ; Machine Learning ; Logistic Regression ; Ensemble Methods ; Loan Defaults

INDEX

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS AND ACRONYMS

AI	Artificial Intelligence
ANN	Artificial Neural Network
AUC	Area Under the Roc Curve
BCBS	Basel Committee on Banking Supervision
CART	Classification And Regression Tree
EAD	Exposure At Default
EDA	Exploratory Data Analysis
FNR	False Negative Rate
FPR	False Positive Rate
GA	Genetic Algorithm
IT	Information Technology
KNN	K-Nearest Neighbors
LD	Lending Club
LDA	Linear Discriminant Analysis
LGD	Loss given Default
LR	Logistic Regression
MARS	Multivariate Adaptive Regression Spline
MDA	Multiple Discriminant Analysis
ML	Machine Learning
NN	Neural Network
PCC	Percentage Correctly Classified
PD	Probability of Default
PLTR	Penalized Logistic Tree Regression
RF	Random Forest
SVM	Support Vector Machines

1. INTRODUCTION

Loan default has a central role in both the traditional banking industry and Internet financial industry, loan default will cause damage to banks supporting a country's economy. What is worse, may even result in an economic crisis. Therefore, it is necessary to establish and perfect the credit lending risk management system, and predict loan default to reduce default risk in the meantime.

Because of the primordial importance of money, banks hold a pivotal role in the economy and society as a whole, promoting economic vitality. Borrowing and lending are the two key pillars of a well functioning banking system, banks act as an intermediary by lending the surplus of money avalaible in the economy (from investemnets and deposits) to people who need it. In other words, they satisfy the need of many businesses and individuals for funds to cover cash needs. Credit has became an important aspect in meny people's life as most of the world population has access to banking services. For example, for many people, a loan may be the only way to afford a house or a car. And for many companies, taking out a loan could be the only solution to support the growth of new businesses and jobs.

Sheikh, Goel and Kumar (2020) stated that even though banking institutions can provide a variety of services their main source of income comes from the loans they grant. Because for every loan an interest rate is applied, which translates into profits for the bank. However, credit lending is a great source of risk. The risk associated with a loan is coupled to the difficulty in distinguishing between creditworthy applicants, which are the ones who will not default, from the unworthy applicants, the ones that are unable to honor the contract and pay back the loan.

As the most recent financial crisis resulted in huge losses world wide, financial institutions increased the attention paid to credit risk prediction, and more resources are allocated to credit risk management in order to lend to creditworthy borrowers and protect themselves from potential losses resulting from customer defaults. Banks are now conscious of the need to adopt accurate credit processes in their systems when granting loans to a company or individual. This industry is of considerable economic importance as billions of euros are borrowed every year, for instance, only in Portugal throughout the year 2021 the amount of credit conceded exceeds 20.000 million Euros, so undoubtedly even a of 1% decrease in the number of bad applicants could translate into a great reduction of losses.

For this matter, credit scoring is one of the main techniques applied for evaluating credit risk and for making informed decisions on whether to grant credit to consumers who submit an application. The main purpose of a credit scoring model is to classify loan applicants into one of two categories: "good" customers (those who are expected to fulfill their obligation and pay back the loan in full) and "bad" customers (those predicted to default, thus failing to pay back the money that was lent) (Lee et al, 2002).

Human screening was the first major method to predict loan defaults. Common practice was to use a method called the 5Cs (the person's character, the capital, the collateral, the capacity to pay the loan and the condition). The analystis would read the application and provide a positive or negative answer based on those five elements.

However, this method was highly subjective, and the same loan could be approved or denied depending on the analyst that was examining it. Later, with the rise in the number of credit applications and the advances in computing power other techniques started to emerge, which allowed the automation of the lending decision. This change in the paradigm lead organizations to realize that the use of credit scoring was a much more efficient than the previous methods employed. In fact, it allowed a decrease of the default rates by more than 50% (Li & Zhong, 2012).

Consequently, traditional credit scoring methodology evolved to the use of statistical techniques such as linear discriminant analysis and linear and logistic regression. These new techniques brought several advantages to financial institutions, as referred by Marqués, García and Sanchez (2012):

1. Time saving;
2. Reduced the probability of accepting a customer that would default in the future;
3. Reduced the overall cost inherent to the credit evaluation process;
4. Decisions could be based on objective and ccurate information; Eliminate the subjectivity associated with human decisions;
5. The performance of the models could be adjusted and perfected at any point in time, and in agreement to the business needs;

More recently, with the breakthrough of artificial intelligence, more sophisticated methods have also been employed to predict loan default, such as neural networks, support vector machines, genetic programming, Naïve Bayes and ensemble methods. According to Malhotra and Malhotra (2003) these models can achieve the same or better results when compared to statistical models, because AI techniques do not need certain distribution assumptions to be true in order to work or provide good results. They are able to learn relatioships between variables and extract the necessary knowledge from the training examples. Meanwhile traditional statistical methods can require statistical assumptions to be met, otherwise it is possible that they do not work correctly, which in the end will affect the accuracy of predictions.

Despite the developments of machine learning and the spreading of it's techniques, the benchmark model of the credit industry is still a statistical method, namely, logistic regression. The reason being is that it's easy to implement and interpret. Which in the indutry if a key factor, considering that banks should always be transparent and are obliged to provide an explanation on why a certain credit application is denied. And logistic regression provides this explanability need (Dumitrescu et al., 2022).

Machine learning is a category of AI that provides computers knowledge through real world interactions which ultimately allows the computer to adapt to new settings. It is one of the most efficient methods to provide analysts with more productive insights and is taking the world by storm as it makes more and more contributions to aspects of the modern world. Banking and finance is one such aspect. Tremendous work is being done to incorporate machine learning techniques in the banking industr, for instance detection of scams, frauds or defaulters are a few explamples of the application of such technology.

According to the World Bank Group financial institutions are progressively becoming more interested in applying sophisticated methods in their credit risk management tasks. The complexity that

emanates from the assessment of credit risk has opened the door for banks to try to find more precise and efficient methods, than the ones being employed nowadays. So it is not surprising that the use of these ML approaches to model credit risk is a growing phenomenon.

To conclude, the world's economic system is in constant change, and the study of credit risk management is of great importance not only to financial institutions but to society as a whole. Thus, the process of credit risk management should be progressive and contínuos. Nowadays, with the revolution of big data, and the requirements of the Basel Committee on Banking Supervision, demanding complex and rigorous credit scoring procedures, machine learning and its unparalleled predictive power and speed can be of great importance in the development of accurate and robust default prediction models.

1.1. BACKGROUND

Credit risk is the economic loss that results from the failure of a counterparty to fulfill its contractual obligations. To mitigate this critical risk, credit scoring has been regarded as a core appraisal tool by many institutions.

According to Huang, Chen and Wang (2007) credit scoring has become one of the primary ways for financial institutions to assess credit risk, improve cash flow, reduce possible risks and make managerial decisions. Its ultimate goal is to assess credit worthiness and discriminate between “good” and “bad” debts, depending on how likely the clients are to default.

Despite the very long history of credit, the history of credit scoring is only some decades old. Expert scoring was the first method employed in the evaluation of credit applications, and it was a purely judgmental process based on the view of what was called the 5 C’s:

1. **The character of the person** – Conduct and character of the applicant;
2. **The capacity-** income;
3. **The Capital-** the loan amount;
4. **The condition-** market conditions;
5. **The collateral-** Guarantee in case of default.

However, this method was highly dependent on the knowledge and experience of the expert, which made it a time consuming process as well as highly susceptible to errors.

One of the earliest research in the field of credit risk goes back to 1932 when Paul Fitzpatrick, tried to predict company insolvency based on financial data. Later, in 1936 Ronald Fisher published an article where he analyzed several types of Íris flowers proposing the technique of Linear Discriminant Analysis (LDA). Although his work was focused on biology, it provided a basis for predictive statistics and a scientific background for modern credit scoring. In 1941, Durand realized that the same approach could be applied to the area of credit risk, he analyzed 7.200 credits and applied Fisher's approach to distinguish them between good and bad loans, using variables such age, gender, job stability, bank account and patrimony (Johnson, 2004).

Soon after, the groundbreaking work of Altman (1968), who proposed an extension of the discriminant analysis, credit risk became a very researched subject. This translated into a rapidly increase in the development of models that were able to output predictions and perform classification tasks. The good results provided by new proposed methods, allowed for a quick replacement of the old methods, which were were less accurate and mainly judmental based. In the 1960s with the appearance of credit cards banks and other credit card issuers began to employ credit scoring, since it was impossible to assess such a high number of credit applications without some level of automation. By the 1980s, there was a complete acceptance of credit scoring and a new method to assess credit worthiness was introduced, namely, logistic regression. Gradually but at a slow pace, linear discriminant analysis and logistic regression, among a few others, started to disseminate and to establish their position as the prefered credit scoring methods, and were able to mantaining their status up to the present time. However both have been highly criticized, due to the fact they need the variables to have a linear relationship in order for the results to be plausible. In other words, they both lack credit scoring accuracy (Ping & Yongheng, 2011).

As stated by Ince and Aktan (2009) in addition to these classical methodologies, several AI techniques have also been employed to assess credit risk. Neural networks, k-nearest neighbors, decision trees, ensemble methods, genetic programming, support vector machines are a few examples of the most recent research methods in this field, which can be an alternative to discriminant analysis and logistic regression, which are still the baseline in the industry. According to Wang, Hao, Ma and Jiang (2011), previous studies have demonstrated that AI techniques can provide better results than statistical techniques when dealing with credit scoring problems, particularly for nonlinear pattern classification.

The innovations in technology have inevitably impacted everyone in different aspects of life, in particular over last decades, AI is na example of such innovations (Rohan Pothumsetty, 2020). In recent years many businesses have started to incorporate AI in their processes, some of the fields where AI is being applied include not only finance but also healthcare, human resources, law, education, robotics among many others. John McCarthy, recognized by many as the father of Artificial Intelligence, said that AI is what allow us to have intelligent machines.

Nonetheless, AI is a broad field where Machine Learning can be viewed as a subcategory. As stated before, one business function where AI incorporation, in particular machine learning, is taking place is the financial services industry. Core functions such as risk assessment, stock trading and fraud detection, have been the center of attention of several researchers and practitioners. The evidence is that credit risk management can be significantly improved through AI and machine learning, thus it is not surprising that the use of these technologies to model credit risk is a growing phenomenon.

The main idea behind ensemble methods is to generate multiple single models and then combine them to produce outputs. The goal is to obtain better predictions than the ones that a single model would be able to provide. Ensemble modeling has quickly proven itself to be a valuable addition, particularly in credit risk. In addition, (Breeden, 2020) explains that they can be diferentiated between homogeneous methods (multiple models of the same type are combined) and heterogenous methods (different types of models are aggregated). It is possible to identify three main classes of ensemble learning methods, which are: stacking, boosting and bagging, which will be discussed further.

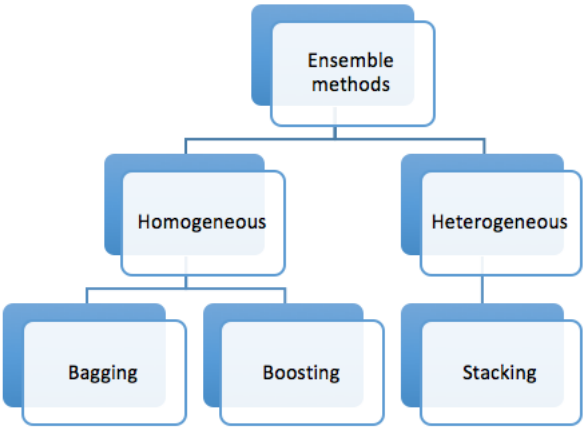


Figure 1: Types of ensemble methods

Source: Authors preparation

It is also important to note that the implementation of machine learning doesn't mean that this technology will completely take over the jobs. The goal is to take advantage of its usefulness and help professionals focus on the important and stategic aspects of the business and spend less on tasks that are repetitive and could more easily be automated.

1.2. STUDY OBJECTIVES

For banks or any financial institution, credit lending activities are a key element. Generally speaking, bad lending decisions can lead to great losses, while good lending decisions can lead to high profits. Therefore, loan providers should have a rigorous credit evaluation process and select only those applicants who have the lowest chance of defaulting. For this matter machine learning can be a great contribution.

The main purpose of the present thesis is to investigate which supervised machine learning classifiers perform the best at predicting customer loan defaults. The following research question will be addressed:

- Can ensemble methods perform better than logistic regression, which is the baseline in the industry, at predicting customer loan default?

The chosen classification methods are logistic regression (traditional scoring technique) and ensemble methods. AdaBoost, GradientBoosting, Random Forest and Stacking are chosen as advanced techniques. In the case of the stacking ensemble, a support vector machine, neural network and logistic regression will be employed as base classifiers and a decision tree will be employed as the meta learner. The algorithms will be trained on a real world credit data set and further evaluated by four metrics. Accuracy and area under the roc curve (AUC), are chosen because they are some of the most popular in credit score evaluation. Type I and type II errors will also be used to assess the performance of the models, in order to take misclassification costs into account.

The final goal is to obtain a robust model, with high discriminatory power that is capable of accurately predict worthy and unworthy credit applicants and aid institutions in their credit lending process.

1.3. MOTIVATION

According to the Bank of International Settlements, the BCBS was created with the purpose of strengthening financial stability worldwide, and work as a forum for regular cooperation between its members on banking supervisory matters. For that purpose, The Basel Committee has given a framework comprising international standards that should regulate the banks activities. These international standards are known as Basel I, Basel II and Basel III.

Basel I (the Basel Capital Accord) published in 1988 advised that the risk weighted assests ratio should not be greater than 8%.

Later, Basel II (the new capital Framework) was released. The new framework was created to improve the first basel accord.This accord revised the proposed guidelines taking into consideration the financial and technological innovations that had occured in the years following the publishing of Basel I, so that the requirements would more accuratly reflect the risks financial institutuions were exposed to.

It is important to mention that the housing boom in the 2000s decade together with the low interest rates verified at the time, caused lenders to give loans to individuals with relatively low capacicty to support a loan. Long after, interest rates rose, which led to a great number borrowers to default on their subprime mortgages, because they were unable to meet the monthy payments. This event led to the most severe recession in decades, known as the 2008 subprime crisis. The effects of the crisis weighed heavily on economic growth, financial stability and bank performance. As a response, Basel III was published. The agreement addressed the main vulnerabilities presented by the banking sector during the crisis, revised and strengthened the three pillars established previously by Basel II, and it also extended it in several areas.

The fourth industrial revolution is progressing significantly. The digital revolution is reshaping the processes of many businesses as well as the way individuals conduct their lifes. In the 21st century, the world is witnessing a drastic increase in data volume, speed and variety as well as the appearance of much more powerful computational tools. This change in paradigm is forcing companies to modify and modernize their processes in order to adapt, gain strategic advantages and ultimately be profitable, and for financial institutions the reality is no different.

In recent years, many studies have demonstrated that artificial intelligence techniques can provide better results when compared to the traditional statistical models, which are still the standard approach in the industry, in particular, as stated by Li and Chen (2020) ensemble methods have demonstrated to be superior to many other machine learning approaches, being considered by many a state of the art solution.

The Basel Accords requesting all banks to have rigorous risk discipline with complex credit scoring systems, the arrival of Industry 4.0 as well as the great results machine learning algorithms have been

demonstrating, are three strong pillars that incentivize the use of AI technologies. Machine learning can play a vital and influential role in the field of finance, in particular in credit risk modeling.

The most recent financial crisis serves as a useful reminder of the importance of a robust risk management culture. It is of paramount importance for a healthy economy, that lending institutions have rigorous and robust credit evalution processes incorporated in their systems. Because if credit risk can be predict well ahead of time, the appropriate actions to prevent loan default can be carried out, and at large avoid a new catastrophic recession.

1.4. STUDY RELEVANCE

The major contribution of this study lies in the advantage of using machine learning tools for credit risk management, specifically, in the implementation of ensemble learning methods to predict loan default.

Machine Learning is a field that is increasingly being explored in the area of financial risk management. However, the application of ensemble methods is still very embrionary. And given the fact that financial institutions are exposed to transactions that often amount to billions of euros, even a marginal improvement may result in a significant decrease in future losses and in a considerable increase in profitability.

The application of ensemble learners could provide three key advantages:

- Assist credit granting activities, promoting agility, support and security in decision making;
- Mitigate risks. Since Machine Learning can provide efficiency and effectiveness to processes ;
- Reduce default rates. The purpose of employing ML is to improve the quality of the credit risk analysis, obtaining more assertive and accurate predictions.

As stated by the World Bank credit scoring can be a great contribution to economic growth. It is important that banks and financial institutions, as well as regulators and governments are able to work together and cooprate in order to explore and develop the positive aspects of innovation.

2. LITERATURE REVIEW

This section provides necessary background information regarding credit risk management, machine learning as well as an overview of previous studies conducted in the field of credit scoring, where machine learning techniques were implemented.

2.1. CREDIT RISK MANAGEMENT

2.1.1. Concept of Credit

Firstly, the word “credit” means in a simplified definition, the possibility to buy now and pay later.

In the banking sector, the Bank of Portugal defined credit as follows: A credit agreement is an contract where a credit institution (creditor or lender) makes money available to a customer (debtor or borrower), who is obliged to repay that amount over an agreed period, plus interest and other costs.

2.1.2. Concept of Risk

According to the Oxford dictionary, risk can be defined as the likelihood that something bad will happen in the future, or a situation that could have a negative or menacing outcome. From an economic perspective, Soares, Moreira, Pinho, and Couto (2008) define risk as the probability of a future cash flow not occurring or occurring in a different amount than expected.

It is also important to make a distinction between risk and uncertainty. There is risk when the probabilities of an event can be objectively estimated, while uncertainty is based only on subjective probabilities. For example, we are in the presence of uncertainty when we are aware that a particular future cash flow is not certain but the probability of its occurrence is unknown. If an estimate of this probability is formulated, the notion of risk is being dealt with.

The Bank of Portugal through a document called MAR (Risk Assessment Model), identifies nine categories of risk, inserted into two distinct groups:

- **Financial risk**: We are in the presence of financial risk, when the risk is directly related to the institution's assets and liabilities; In this case, four types of risk can be identified: **credit risk, market risk, interest rate risk** and **currency risk;**

- **Non-financial risk**: We are in the presence of non financial risk when the risk results from circumstances that are external to the institution (economic, political or social phenomena) or internal to the institution (procedures ,human resources, technologies, and others). In this case five types of risk can be identified, namely: **operational, reputation, information systems, strategy, and compliance.**

The Basel II Capital Accord highlights the importance of two main sources of risk besides credit risk, which are market risk and operational risk, both referenced previously.

Market Risk: Is the probability that a negative event will result in losses. This risk is mainly realated to financial instruments that are affected by the conditions of the market. For example, risk associated with stocks, bonds, interest rates, exchance rates among a few others (Banco de Portugal, 2007).

Operational Risk: This risk is mainly related to losses resulting from failures in the internal processes, insufficient human resources, inadequate conduct from the employees, changes in regulations, activities affected by outsourcing, and unpredictable external events that can affect infrastructures (Banco de Portugal, 2007). For instance, robbery, damage to physical assets, hacking damage, failure of a computer system, natural disasters such as an earthquake are examples of events that constitute operational risk.

2.1.3. Concept of Credit Risk

According to Banco de Portugal (2007) credit risk can be simply defined as the probability of capital losses resulting from the fact the counterparty of a credit contract was unable to refund the total debt amount. The inability of a borrower to fulfill its financial commitments could negatively impact the results presented by banks and financial institutuions.

Credit risk is typically represented by means of three factors, which need to be estimated by requirement of the Basel II Capital Accord:

- **Default risk:** The default risk is the probability that a default event occurs. It is referred to as probability of default (PD) and its values range from 0 to 1. According to the BCBS when there is a payment delay of at least 3 months it can be considered default. There can be several reasons for default, e.g. counterparts in a fragile financial situation, debt to income ratio becames to high, loss of income, among many others, bur normally the counrterparty is in a fragile and financialy stressfull situation. In the eventualy of default the actual loss depends on two factors: LGD (loss given default) and EAD (exposure at default). These values are discussed below.

- **Exposure at Default**: EAD is the predicted amount of loss a bank may be exposed to when a client defaults on a loan. Being that default occurs at an undetermined date in the future, this loss is contingent upon the amount to which the bank was exposed at the time of default. For some products, for example a straight loan, the amount is fixed. However, for other products, e.g. credit cards the amount varies with the liquidity needs of the borrower.

- **Loss Given Default:** LGD represents the actual loss in case of default. When the loss equals the full exposure amount, the LGD is 100%, in the case of no loss, the LGD is equal to zero. A negative value can indicate that there is a profit , this could happen for instance, because of penalty fees. In other, the values of the LGD can be superior to 100%, e.g., due to almost no recovery from the defaulted counterpart, or even because of the litigation costs. The loss in

the case of default depends on the percentage that can be recovered from the defaulted counterpart and the total exposure to the counterpart. In practice, the LGD values can vary quite a lot and depend upon the type of default and its resolution:

Cure: In this case, the counterpart continues to fulfill its contractual obligations and there is no significant loss for the bank.

Restructuring: The customer overcomes the default sitation e.g., partial debt forgiveness and debt renegotiations.

Liquidation: A liquidation process is necessary and the financial institution proceeds to seize the collateral

2.1.4. Concept of Credit Risk Management

Banks need to manage a broad spectrum of risks, and the relationships between them. Manageing them in an effective way in essencial to survival and long term success of any banking or credit lending organization.

Credit risk management is undoubtedly among the most crucial issues in the field of financial risk management. Identify, measure, analyze and control risks are the focal points of appropriate credit risk management. For this purpose, the Basel Committee established the following four main principles that should be the core of an adequate credit risk management:

1. Promote a healthy credit risk environment;
2. Operate according to fair and transparent credit lending decisions;
3. Ensure that the measurement and monotoring procedures are adequate and appropriate;
4. Ensure that credit risk is managed with appropriate controls.

Although credit risk management processes may differ from institution to institution, depending on the complexity of their services, a comprehensive and detailed credit risk management system will address those four practices.

Credit risk management is of paramount importance and represents a great challenge for institutions, as failure in this front can inevitably lead to bankruptcy. The most recent subprime crisis is the maximum exponent of the consequences and of the significance of credit risk. On that account, institutions must privilege an eficiente, rigorous and agile credit risk management, allowing them to adapt to changes in the economy and in the population's standard of living.

2.1.5. Concept of Credit Scoring

Credit scoring is one of the most widely used credit risk analysis tools and according to the World Bank its main advantage is that it is effective and swift, way to decide on an applicant's suitability for a loan. And theoretically, several definitions of credit scoring have been provided by many researchers in the field.

One of the definitions is that credit scoring comprises a set of methods and techniques that should be used by credit granting institutions when deciding wheter or not to grant credit to an applicant. The main goal is to estimate the probability that that applicant will default. If that probability is low the applicant should be considered a "good" customer, since it is more likely that he will pay back the loan. If the probability is high he shoud be considered a "bad" custumor since it is likely that he will shouw unwanted behaviour in the future.

This process includes gathering, analyzing and classifying different credit elements and variables to assess the credit decisions. It has been regarded as a core assessment mechanism by different institutions during the last few decades. In particular, since the start of the twenty first century, the evolution of technology and the development of powerful computational tools have allowed the study and introduction of more advanced techniques in this field, namely, AI techniques.

Other than credit scoring, behavioral scoring has also become established as a major tool in forecasting financial risk. As stated by (Thomas, 2000) behavioral scoring deals with existing customers. And helps to deal and understand what actions should be taken when there is a problem ou doubt regarding credit matters of existing customers. For instance, if a customer fails to pay a monthly payment, what should be done? Or if a cliente wants to increase it's credit card limit, should the firm allow it? While behavioral scoring is a dynamic process credit scoring can be seen as a static process which deals with only new applicants.

2.2. MACHINE LEARNING

2.2.1. What is Machine Learning

Frequently, machine learning is viewed as an application of artificial intelligence. It is a research field in the intersection of statistics, artificial intelligence and computer science. In a broad definition we can say that machine learning is a concept that enables machines to learn from real world observations and behave like human beings, improving their ability to learn and perform using data given as inputs. Machine Learning is about creating algorithms (set of rules) that extract knowledge from data.

It is also known as predictive analytics. We can say that predictive data analytics is the art of building and using models that make predictions based on patterns extracted from historical data. Nonetheless, in predictive data analytics a broad definition of the word prediction is used. In everyday usage, the word prediction has a temporal aspect (we predict what is going to happen in the future). However, in data analytics a prediction is the assignment of a value to any unkown variable. For example, we can predict the price of something that will be sold in the future, or we can predict (classify) a type of document, e.g. spam or not spam, so in some cases, prediction has a temporal aspect but not in all.

In recent years, ML has gained the interest of researchers as they are trying to implement models and algorithms to perform various important tasks and facilitate everyday life. Currently, machine learning is the IT domain that contributes the most to business forecasting problems and according to Ngai, Wong, Chen and Sun (2011) is ideal to:

- Understand the pattern of banking transactions;
- Identify customers through data;
- Distinguish between a normal action and a fraud.

Traditional Programming vs Machine Learning**:** In traditional programming both data and a program are run on a computer to produce an output. As for machine learning both data and output are run on a computer to create a program that can then be used in traditional programming. In contrast with traditional programming, machine learning is more automated and less of a manual process.

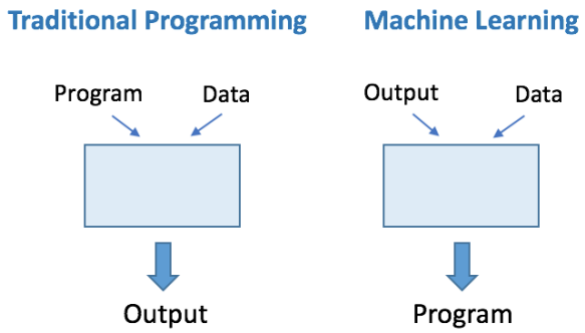


Figure 2: Traditional Programming vs Machine Learning

Source: Authors preparation

2.2.2. Types of Learning

At a fundamental level, machine learning can be divided in four categories: Supervised Learning, Unsupervised Learning, Semi-supervised learning and Reinforcement learning.

In supervised learning, the algorithm is developed using data that contains a target feature (label) and independent variables (features) and automatically learns a model of the relationship between them. This model can then be used to make predictions for new instances. Within supervised learning, several algorithms can be employed, including: Linear and logistic regression, decision trees, svm, neural networks, random forest among many others.

Two types of problems are at the centre of supervised learning:

- **Classification:** Classification algorithms are used to solve problems in which the output is a discreet target variable (categorical). One common real world example of a classification problem is spam filtering, where the algorithm is trained to distinguish a spam email from a non spam email.

- **Regression**: Regression algorithms are used to solve problems in which the output is a contínuos variable (numerical). Price prediction is a popular application of regression problems.

In unsupervised learning the ultimate goal is to describe, to provide a summary of a dataset. In contrast to supervised learning, in unsupervised learning the data doesn't contain labels. The algorithms are designed to exploit patterns in the data and try to identify similarities. In other words, these algorithms, try to understand and explore characteristics of the data instead of focusing on making predictions.

In regards to semi supervised learning, it can be said that it uses labeled and unlabeld data during the training stage. Which means that it has similarities with the both types of learning mentioned above.

Reinforcement learning is the training of ML models to make a sequence of decisions, and depending on those decisions the model gets either rewards or penalties, taking into account that its main goal is to maximize the total reward.

2.2.3. Credit Scoring as a Classification Problem

Classification is one of the main problem in machine learning, in particular, in supervised machine learning. Given a data set, a set of labeled observations, algorithms make use of the knowledge from the independent features and the target label relationship in order to predict the class label of an unlabeled instance. Even though many learning algorithms have been proposed, none of them has proven to perform better than all the others for all classification problems. From a practical point of view, credit scoring can be considered a binary classification problem. In the scope of the thesis, a new input sample (customer) must be categorized into one of the two predefined classes (default or no default).

The input consists of a variety of information that describes characteristics of both the loan and the borrower (e.g. age, number of dependents, interest rate, occupation, loan purpose, number of years employed, home ownership, annual income, etc...). The output consists of two classes: "no default"(representing those who are able to fulfill their financial obligation) and "default" (representing those who should be denied credit because they will likely not pay the loan). The algorithm is then trained on that data in order to learn a decision criteria that will be used to assign new credit applicants to one of the two mentioned classes. Formally, given a dataset of *n* customers, $S = \{(\chi_1, \gamma_1), \ldots, (\chi_n, \gamma_n)\}$, where each customer $\chi_i = (\chi_{i1}, \chi_{i2}, \ldots, \chi_{iD})$ is characterized by D variables defined on an input feature space X^D, and $\gamma_i \in \{no\ default, default\}$.

2.3. PREVIOUS RELATED STUDIES

The concept of credit scoring is relatively new, dating back to the 1940's when Durant (1941) with the purpose of assessing credit risk employed the technique of Linear Discriminant Analysis (Reichert, Cho, & Wagner, 1983) (Karels & Prakash, 1987).

Since then many other developments have been proposed in the literature, in particular, since the influential paper by Altman (1968). Logistic Regression and Survival Analysis are two of the first and most studied methods (Mungsai & Odhiambo, 2019), (Glennon & Nigro, 2005), (Cao, Vilar & Devia, 2009), (Stepanova & Thomas, 2002), (Dirick, Claeskens & Beasens, 2016), (Baesens et al., 2015).

Even though the concept of machine learning in finance is relatively new, much research has been done (Baesens, et al., 2003), (Atish & Jerrold, 2004), (Lee & Chen, 2006) (West, et al., 2005), (Sinha & Zhao, 2008), (Yu, et al., 2008), (Hsieh & Hung, 2010), (Zhou, et al., 2010) (Brown & Mues, 2012) (Abellán & Mantas, 2014) all employed machine learning in predicting loan default.

There is a plethora of research in this field and the following studies are a few examples. Huang, Chen and Wang (2007) investigated three strategies to construct a hybrid SVM-based credit scoring model and then benchmarked their performance against genetic programming, C4.5 and neural network . Their experimental results showed that the SVM approach was a promising addition to the literature, since less input features were necessary in order to obtain a similar accuracy.

Ensemble methods have also gained some attention West, Dellana and Qian (2004) proposed two ensemble strategies, bagging and boosting, where a multilayer perceptron neural network was employed as a base classifier. Their work showed that the ensemble strategies employed reduced the generalization errors by 3-5% in all datasets. And even if a reduction of 3 to 5% seems small , it is important to notice that the credit industry can have transactions of billions of dollars. Similar conclusions arise from the work of Akindaini (2017) who applied five machine learning methods, logistic regression, multinomial-multiclass logistic regression, naive bayes classifier, random forest model and KNN classifier in the prediction of mortgage loan default. The ensemble method (random forest) presented the highest accuracy among all classifiers, with a value of 95.68%. and Naive Bayes classifier provided the lowest accuracy of 70.74%.

A study involving two different credit problems was proposed by Chen and Huang (2003) who solved them by applying neural networks and genetic algorithms. The first problem was constructing a NN-based credit scoring model, which assessed credit worthiness and classified the applicants as "good" (accepted) or "bad" (rejected). The second part of the problem was trying to understand the reason behind the rejected credit application as well as trying to reassign them to the preferred clase. This second part of the problem was explored using a technique based on genetic algorithms.

Baesens et al., 2015 provided a comprehensive view of the state-of-the-art in predictive modeling as they performed a benchmark of 41 classification algorithms across eight credit scoring data sets. The classifiers were split into three categories: individual classifiers (e.g. Naive Bayes, KNN, LDA, LR, J4.8, CART), homogeneous ensemble classifiers (e.g. random forest, rotation forest and Bagged MLP) and heterogeneous ensembles (e.g. Stacking, GASEN, Kappa pruning and k-nearest oracle). The

performance of the classifiers was then assessed considering six indicators:PCC,H-measure, Kolmogorov–Smirnov statistic, Brier Score, AUC. The results suggested that heterogeneous ensembles classifiers perform very well, since the top ten classifiers all belonged to this category. In addition, several classifiers predicted credit risk significantly more accurately than logistic regression, which is the standard in the industry.

Marqués, García and Sánchez (2012) conducted a study to determine which base classifiers are most appropriate to be used in ensemble models. The results show that decision trees are the prime solution for the majority of ensemble methods, in particular the C4.5 decision tree. The Logistic regression and MLP presented a good performance as well. On the contrary, the naive Bayes and nearest neighbor classifiers appeared to have significantly worse results.

Malhotra and Malhotra (2003) performed a study that compared the performance of neural networks and multiple discriminant analysis (MDA) in identifying potential loan default. In their research they conclude that neural network models consistently perform better than the MDA models and are more successful in minimizing the Type I error. In addition, the author considers the neural network to be a better technique because unlike MDA it doesn't require normality assumptions to be met

In a similar line of thought Granstrom and Abrahamsson (2019) examined several machine learning models in order to undersand which one(s) were capable of better predicting defaults.The investigated techniques were Logistic Regression, Random Forest, Decision Tree, AdaBoost, XGBoost, Artificial Neural Network and Support Vector Machine. On pair with the previously mentioned studies, an ensemble model obtained the overall best performance. In the results presented by the authors XGBoost performed better than the remaining algorithms. However, the highest precision was shown by ANN while AdaBoost was able to obtain the highest value for sensitivity. Another conclusion was that, generally, tree-based models , on average, can perform better than ANN's and are also more stable.

Fitzpatrick and Mues (2015) in their research tried to understand if approaches from the statistical/machine learning literature provided better predictive performance for mortgage credit risk than logistic regression. For that purpose the authors implemented two techniques with roots in machine learning: Boosted Regression Trees and Random Forests and a statistical model: semi-parametric Generalized Additive Models. These models were applied to four datasets regarding more than 300,000 mortgages. The results suggest that the boosted regression trees have a significantly better performance than logistic regression.

All the previous studies used real world datasets, some with a reduced number of input features and only a few thousands of instances and others with millions of observations and more then twenty features, however they all aimed at predicting loan default. Some variables seemed to be important for many of the authors and were present across several studies, such as income, loan amount, loan purpose, employment status, and home ownership.

Many techniques have been proposed from statistical models to artificial intelligence methods, as well as performance evaluation criteria. The following table presents a brief comparison of previous works in terms of models and metrics applied.

Year	Author	Models	Performance Metric
2009	Ince & Akten	Linear Discriminant Analysis Logistic Regression Neural Network CART	Type I error Type II error
2011	Ping & Yongheng	Linear Discriminant Analysis Logistic Regression Neural Network Proposed SVM based hybrid	Accuracy
2011	Wang et al.	Stacking Boosting Support Vector Machine Neural Network Bagging Logistic Regression Decision Tree	Type I error Type II error Accuracy
2016	Hamid & Ahmed	J48 Naive Bayes Bayesian Network	Accuracy
2016	Ala'raj & Abbod	Logistic Regression MARS Proposed ensemble method	Accuracy AUC Brier Score H-measure
2018	Tokpavi et al.	PLTR Random Forest MARS Non-Linear Logistics Regression Linear Logistic Regression	AUC PCC Brier Score Partial Gini Index Kolmogorov-Smirnov
2019	Bayraci & Susuz	Support Vector Machine Deep Neural Network J48 Logistic Regression Naive Bayes	Type I error Type II error
2020	Maheswari & Narayana	Random Forest Logistic Regression K- Nearest Neighbor	Accuracy Precision Recall

2020	Li & Chen	Support Vector Machine Neural Network Decision Tree Logistic Regression Naive Bayes Random Forest AdaBoost XGBoost LightGBM Stacking	AUC Brie Score Kolmogorov-Smirnov Accuracy

Table 1: Comparison of previous related studies

Source*:*Authors preparation

3. METHODOLOGY

The present thesis follows the CRISP-DM line of work, which is presented below:

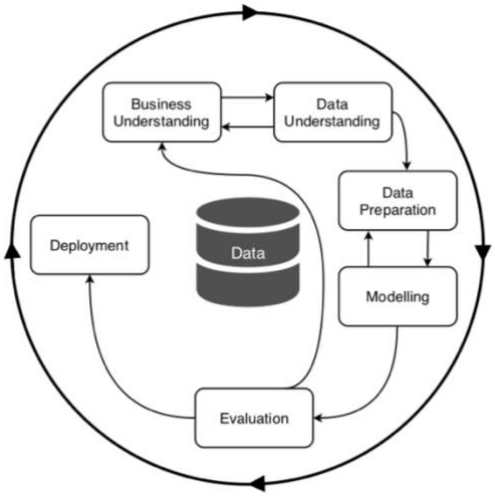


Figure 3: The CRISP-DM Process

Source: *CRISP-DM Twenty Years Later: From Data Mining Processes to Data* Science Trajectories (Plumed et al., 2020)

This initial step, business understanding consists of identifying and understanding the project requirements and objectives, gaining domain knowledge regarding the necessary fields and establishing a preparatory plan. This is mainly explicit in the literature review and background.

The data understanding step begins with obtaining the dataset and proceeds to an exploratory analysis to better understand the data collected. The purpose of this phase is to identify data quality problems, discover first insights and overall get familiar with the data. For this matter it is necessary to use statistical measures, to find the meaningful patterns, as well as visual representations of the data, resorting to graphs, histograms, scatter plots and other useful visualization tools.

The data preparation covers the activities that are necessary to transform the inicial raw data into a final dataset that can be used by the algorithms. This is the most time consuming step in the entire process and includes different tasks, namely, data cleaning, feature selection and data transformation.

The following phase, modeling consists in implementing the proposed algorithms and calibrating their parameters to optimal values.

One of the last steps consists of evaluating the models obtained in the previous phase and reviewing the steps executed to construct them. The performance of the models is assessed based on several appropriate measures.

The final step is the deployment of the model. In this context the construction of the models has only an academic purpose, it will not be presented to a customer as it could happen in a real life situation. In this case deployment will only consist of an analysis and discussion regarding the results obtained, and ultimately reach a conclusion concerning the research question.

4. MODELS PRESENTATION

The following section will provide a brief explanation of the proposed models.

4.1. LOGISTIC REGRESSION

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

The logistic model is given by the following formula:

$$f(x) = \frac{1}{1 + e^{-(\beta 0 + \beta 1 x)}} \quad (1)$$

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

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

$$\frac{f(x)}{1 - f(x)} = e^{(\beta 0 + \beta 1 x)} \quad (2)$$

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

$$P = P(loan\ status\ is\ 1) = \frac{1}{1 + e^{-(\beta 0 + \beta 1 x + \cdots + \beta k x k)}}$$

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

$$\frac{f(x)}{1 - f(x)} = e^{(\beta 0 + \beta 1 x + \cdots + \beta k x k)}$$

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

4.2. DECISION TREE

A decision tree is an algorithm that can be seen as a flow chart, and as the name suggests, that flow chart has the structure of a tree. Several elements can be identified in a decision tree model. The first element that is possible to detect is the the top node. The top node in a tree is called root node and from there a path is traced until a leaf node is reached. The second element, are the internal nodes, also refered to as non leaf nodes. This nodes represent the test on an attribute, for instance, “is x higher than 0.64?”, which can be seen in the tree below. And finally, the last element are the terminal nodes, also known as leaf nodes, the leafs hold a class label (the outcome of a test). The leaf nodes allow us to evaluate the discriminatory power of the tree. The more homoegeneou the leafs are the better the model performs. On the contrary if the leafs tend to be heterogeneous, that means that the model doesn’t separate well the output classes. An example of a decision tree with a two-dimensional split feature space can be seen in Figure 4. The terminal nodes in the tree bellow are called leaves and correspond to the predicted outcomes.

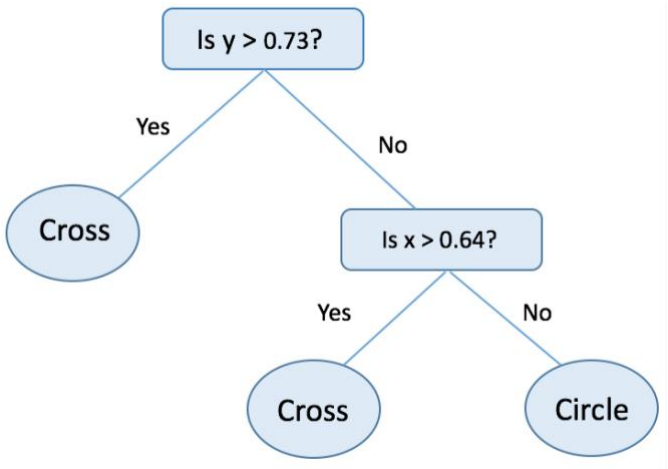


Figure 4: Decision tree with two classes

Source: Authors preparation

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

4.3. SVM – SUPPORT VECTOR MACHINE

A Support Vector Machine (SVM) is an algorithm that can be employed to solve supervised machine learning problems such as classification and regression tasks. SVMs are based on the idea of finding a hyper plane that divides a dataset into two classes, while maximizing its margin (smallest distance between the hyper-plane and the observations). This optimal hyperplane is obtained based on support vectors (data points that are the closest to hyperplane). Support vectors are considered critical, because they define the position of the hyperplane, if we were to alter them the posititon of the hyperplane would change.

Intuitively, the further from the hyper-plane the higher the probability of a data point being correctly classified. Therefore, an optimal hyper plane should maximize the margin, so that the data point can be as far away as possible, while still being on the correct side. So when new data is added, the class assigned to it will depend on the side of the hyperplane it belongs to.

Assuming an input vector (X_1,Y_1), (X_2,Y_2),…, (X_n,X_n) (in a d-dimensional feature space), and a $Y_i \in \{0, 1\}$ class label. What the SVM tries to do is find an hyperplane that separates the data points correctly according to the possible output classes, in this case, it tries to find the optimal hyperplane that separates the default observations from the non default observations. If an optimal hyperplane is found then the margin width should be largest possible. It is also important to note that the SVM is capable of using kernel functions to transform the data in higher dimensional spaces, this is particularly important in cases the the data is not linear.

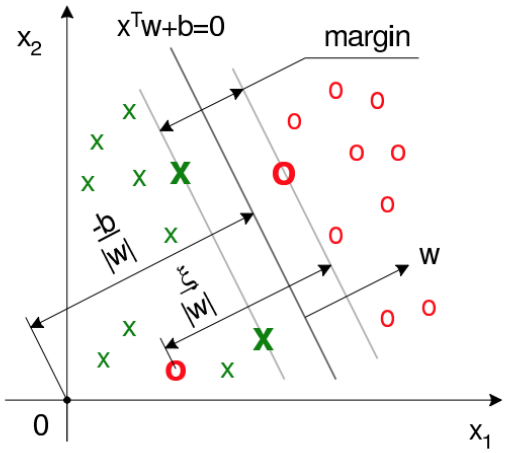


Figure 5: SVM Margin Maximization

Source: (Moro, 2006)

Figure 5 presents the schematic of the svm in a non separable case.

4.4. NEURAL NETWORK

Artificial neural networks take inspiration from the structing and functioning of the human brain functions and ultimately try to replicate it. In general terms, a neural network structure is composed of several units that are connected. In that structure it is possible to distinguish three layers: input layers, hidden layers, and output layers. The neural structure, the model of the neuron and the learning algorithm are the key factors to the implementation of a neural network.

Several netwok structures are available, nonetheless the most popular one is the multilayer feed forward network, largely because it is easy to train and to comprehend. In this architecture, there is a first layer, an input layer, which receives input feature vectors. Usually, to each neuran corresponds one value of the feature vector, so the number of features in a dataset is the same as the number of input vectors).There is also an output layer which outputs labels, (where each neuron usually corresponds to a possible class, in the case of the present thesis the output is binary so there is only need for one neuron). Between the input and output layers are the called hidden layers. The hidden neurons and the output neurons are functional units, which are activated by an activation function, ReLu and sigmoid functions are among some of the most popular (Zhou, 2012; Abdou & Pointon, 2011).

A Neuron can also be called a unit, which is the basic component in a NN. One of the most popular models is the perceptron. Each neuron receives a set of weighted inputs from the neurons of the previous layer. All these inputs are added together and to a bias (each neuron corresponds to a bias) in order to obtain an activation value. The activation value is then “squashed” by an activation funtion ir order to determine the ouput value. This output will act as the input value for the neurons in the next layer, this process is repeated until the last layer in the network is reached.

When training a neural network the main goal is to determine the values of the mentioned weights as well as the biases of the neurons, because they are the ones that will ultimately affect and determine the final performance of the the algorithm. There are many neural network learning algorithms but the most successful algorithm is Back- Propagation. The idea behind it is that at first, the inputs are feed-forwarded from the input layer to the output layer, through the hidden layer, at which the error is calculated by comparing the ouput provided by the neural network (prediction) to the ground truth (the actual real value).Then the error will go the opposite way (it is back propagated) through that path the weights and biases are adjusted to reduce the error as much as possible. This process is repeated several times until the training error is minimized or the training process is terminated to avoid overfitting (Zhou, 2012).

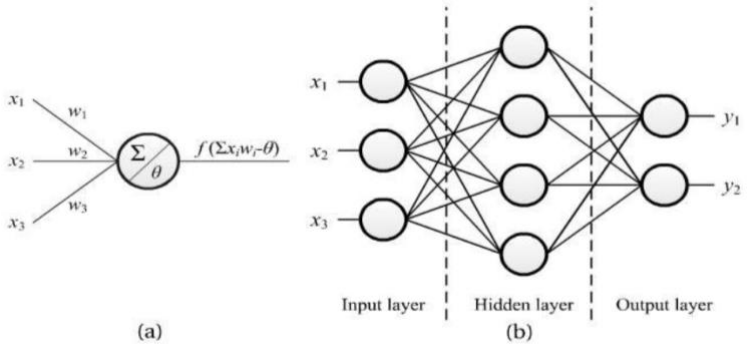


Figure 6: Ilustration of a neuron (a) and a neural network (b)

Source: Ensemble Methods Foundations and Algorithms (Zhou, Z.-H., 2012)

4.5. ENSEMBLE METHODS

As mentioned previously, machine learning models can generally subdivided into individual models and ensemble models. Ensemble methods are the application of multiple models to obtain better performance than it would be possible to obtain from a single model. They are considered to be a state-of-the-art solution for many tasks.

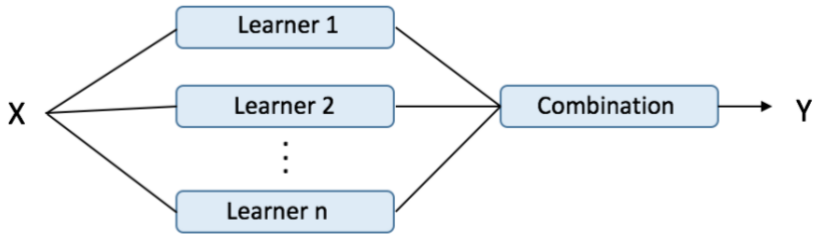


Figure 7: Common ensemble architecture

Source: Authors preparation

Generally, ensemble learning can be separated into two categories: homogenous (combines classifiers of the same kind) and heterogeneous (combines different kinds of classifiers). Boosting and Bagging belong to the homogenous integration while, Stacking belongs to the heterogeneous.

According to the base learner generation process, ensemble learning can be roughly divided into two methods:

- **Parallel ensemble methods:** Base learners are created all at the same time. The main motivation for applying this method is to use the independence between learners.
- **Sequential ensemble methods:** Base learners are generated consecutively. The motivation is to use the dependence between learners. The learners are constructed in sequence so the error of the previous learners can be avoided, thus improving the aggregated performance.

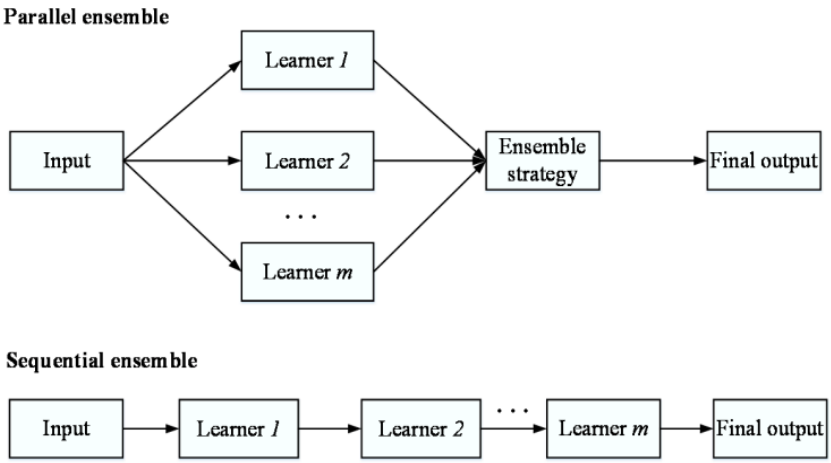


Figure 8: Flowchart of parallel and sequential ensemble

Source: Ensemble Methods Foundations and Algorithms (Zhou, Z.-H., 2012)

In terms of ensemble prediction, majority voting or one of its variants (e.g., weighted voting) are commonly employed, mainly because of their simplicity and easy implemention. In the case of majority voting, each base learner contributes an equal amount to the final prediction. In a classification problem the class label is predicted using the mode of the members predictions. In the context of weighted voting, the contribution of each learner to the final prediction is weighted based on the performance of that model. The better a model performs the higher the weight assigned to it, and therefore the more important they are in final prediction. Other methods can also be employed, e.g., soft voting, sum rule among others.

The ability of an ensemble model to generalize is usually much stronger the the ability of the single base learners.In fact, base learners are also referred to as weak learners. Because they can perform just slightly better than random guess, the ensemble methods are able to boost them to strong learners which can make accurate predictions.

4.5.1. Bagging

As stated previously Bagging is a representative of parallel ensemble methods. The simple idea behind it is that the ensemble is made of base learners built on bootstrap replicates of the training set and afterwards the outputs are combined by majority vote.

The algorithm of Bagging is given as follows:

Input: Data set D = {(x_1 , y_1), (x_2, y_2),...., (x_m , y_m)};
Base learning algorithm $\mathcal{L}$
Number of learning rounds T.
Process:
1. **for** t=1,...., T:
2. h_t= $\mathcal{L}$ (D, $\mathcal{D}_{bs}$); % $\mathcal{D}_{bs}$ is the bootstrap distribution
3. **end**
Output: $H(x)$ = arg max $\sum_{t=1}^{T} \mathbb{I}\ (h_t(x)= y)$
$y \epsilon y$

Figure 9: The Bagging Algorithm

Source: Authors preparation

Two key elements of bagging are bootstrap sampling and model aggregation. The dataset used to train the base learners is obtained by boostrap sampling. So that the dataset used by the single models is a subset of the original set of data.In detail, given a training data set containing m instances, a sample of m instances or less will be generated, this sampling is done with replacement. For this reason some original examples can appear in the same dataset several times, while in other datasets some observations may not be present. By applying the process T times, T samples of m or less instances are obtained.

In addition, the majority voting is usually employed as the aggregation method. The final classification result is the one with most occurrences among the classification results of the base learners. In the presence of a regression problem the final output is obtained by calculating the average value of the outputs of the base learners.

4.5.2. Boosting

The main idea behind boosting is that we have an algorithm with a weak performance, it performs just slighly better than random chance, however it can be "boosted", as the name suggests, into a stronger algorithm, with a much more powerfull prediction capacity.

Differently from bagging, the base learners are generated consecutively and not simultaneously. This method is mainly focused on reducing bias, so the base models often considered for boosting are models with low variance and high bias. The general boosting procedure is quite simple. First, it is necessary to train the weak classifiers, using the training set. Once the models are trained they produce the outputs for each one of the records present on the dataset they were trained with. And according to their level of correctness, a weight is assigned. In the case the sample is correctly classified, the weight assigned should be relatively small, or relatively large if the sample is wrongly classified.

Then, the construction of the second weak classifier is done based on the weighted samples of the previous classifier, in order to make the samples with larger weights be accurately classified. Hence, as the wrongly classified instances have higher weights the second classifier will consider them to be more important. By repeating this process, several weak classifiers will be built in order to achieve better classification performance. Hence, the learners with better performance will have a bigger importance in the final prediction.

Input: Sample distribution $\mathcal{D}$;
Base learning algorithm $\mathcal{L}$
Number of learning rounds T.
Process:
1. $\mathcal{D}_1 = \mathcal{D}$. % Initialize distribution
2. **for** t=1,...., T:
3. $h_t = \mathcal{L}(\mathcal{D}_t)$; %Train a weak learner from distribution $\mathcal{D}_t$
4. $\epsilon_t = \mathrm{P}_{x \sim \mathcal{D}_t}(h_t(x) \neq f(x))$; % Evaluate the error of h_t
5. $\mathcal{D}_{t+1}$ = Adjust Distribution ($\mathcal{D}_t$, ϵ_t)
6. **end**
Output: $H(x)$ = Combine Outputs ($\{h_1(x),...., h_t(x)\}$)

Figure 10: The Boosting Algorithm

Source: Authors preparation

4.5.3. Stacking

Stacking or stacked generalization, unlike bagging and boosting, uses heterogeneous weak learners (different kinds of base models are combined), which are built in parallel and introduces the concept of meta learner. This meta learner combine the individual learners and output the final prediction instead of voting to combine predictions of base learners. In simple terms, the idea of stacking is to build several classifiers in the first layer as base level learners (level 0 models), the models that fit on the training data and whose predictions are assembled. Along with building a meta-model (level-1 model) that learns how to best combine the predictions of the base models and outputs a final prediction.

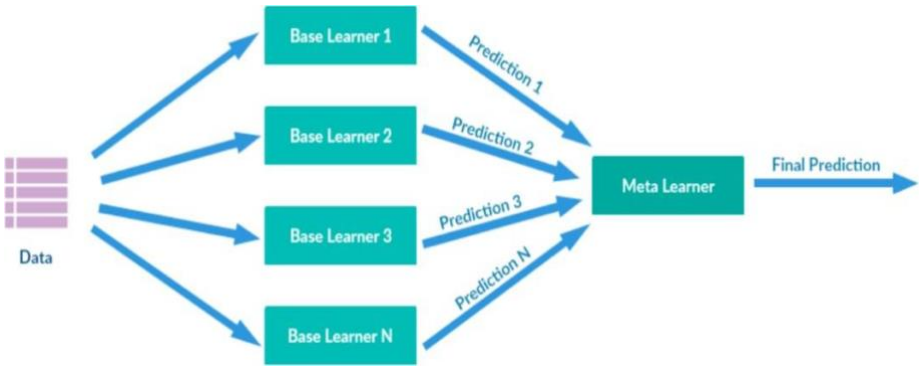


Figure 11: Schematic of a stacking classifier framework

Source: ResearcheGate

First, the several base learners are trained in parallel, on the original dataset. Secondly, they will generate the predictions for each one of the observations on the original dataset. These outputs generated by the first level classifiers will constitute the new dataset that is used for the learning stage of the meta learner. It is also important to note, that the meta learner and the base learners should not be trained using the exact same data, because if that happens there is a high probability that the model will tend to overfit.

Input: Dataset $D = \{(x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m)\}$;
First-level learning algorithm $\mathcal{L}_1, \ldots, \mathcal{L}_T$;
Second-level learning algorithm $\mathcal{L}$;
Process:
For $t = 1,2, \ldots, T$:
$h_t = \mathcal{L}_t(D)$ % Training a first-level individual learner h_t
end; % Learning an algorithm L_t to the original dataset D
$D' = \emptyset$; % Generate a new dataset
For $t = 1,2, \ldots, m$:
For $t = 1,2, \ldots, T$:
$z_{it} = h_t(x_i)$ % Use h_t to classify the training example x_i
end;
$D' = D' \cup \{((z_{i1}, z_{i2}, \ldots, z_{iT}), y_i)\}$
end;
$h' = \mathcal{L}(D')$ % Training the second-level learner h' by applying the second-level
% Learning algorithm $\mathcal{L}$ to the new dataset D'
Output: $F(x) = h'(h_1(x), h_2(x), \ldots, h_T(x))$

Figure 12: The stacking algorithm

Source: (Li & Chen, 2020)

4.5.4. Random Forest

Random Forest models are considered to state-of-the-art, and similarly to the other mentioned ensemble models they can be employed for solving both classification and regression problems. In this section the main focus is on random forest for classification problems. The basic general idea behind this method is that is based on the construction of various decision tress. Further, the output of each of those trees is considered to make a final prediction based on majority voting, the final output is the one that is most occuring amonst all the predicted classes.

The random forest algorithm is an extension of the bagging method, , nonetheless with one major difference. Random forest introduced the concept of feature randomness. If we were to construct several trees using the exact same data, we would get the same results and that is not the main goal, considering that we need diversity amongs the different trees and not models that are highly similar. For this matter, the random forest algorithm introduces randomness in two different ways: each tree in the ensemble is built on what is usually called the bootstrap sample (the data record is drawn from the original data with replacement), and another instance of randomness is then added as each tree is built using a randomly selected subset of features. Feature randomness, ensures that that is diversity amonst the various trees and hence, low correlation. This is a key contras between random forests and decision trees. While random forests only select a subset of features to perform the splits on each node, decision trees take into account the existent variables.

Based on the description of Andy Liaw and Matthew Wiener (2002) the random forest algorithm can be explained as follows:

1. Draw n instances from the original set (boostrastap sample), with replacement, to create new datasets. The number of datasets should be the same as the number of base decision trees.

2. Creat an unpruned classification tree that should be trained in one of the bootstrap samples (datasets). When creating the trees, introduce feature randomness. Instead of choosing the best split considering all features present in the dataset, consider only a randomly choosen subset of m predictors and select the best one.

3. Predict the final output by aggregation the predictions of the base learners (i.e., majority votes for classification).

To sum up, random forest combines flexibility with the simplicity ineherent of decision trees resulting in a vast improvement in accuracy. Other advantages are the fact that it is easy to understant the importance of each feature to obtain the final classification, it runs efficiently on large datasets, processes missing data and is capable of handling many independet feature without the need for feature selection.

4.5.5. AdaBoost

AdaBoost was first proposed by Freund and Schapire, and is still up to this day one of the most widely used ensemble approaches, being implemented in numerous areas. This current state is mainly caused by the fact that the algorithm provides a fast performance and low costs of implementation alied to a

great generalization capability, making it an effective and well liked solution to solve a variety of problems.

According to (Schapire, 2013) the main idea behind this approach it to combine many learners that are relatively weak, with inaccurate rules, this combination should provide an accurate prediction rule and thus a classifier with great performance.

Once again, AdaBoost algorithms can be used as a solution for both classification and regression problems. The most common algorithm used with AdaBoost are decision stumps. These stumps are considered to be the most simple trees that is possible to obtain, since they only have one split (decision tree with one level), and are considered to be weak learners, as they perform just slightly better than random chance. Simply, what this algorithm does is that, it starts with the unweighted training sample, then the AdaBoost builds a classifier, in this case, a classification tree (stump), that outputs the class labels for the given data set. If a data point is not correctly classified, the weight that is assigned to that training data point is increased (boosted). On the contrary, if the sample is correctly classified the weight is reduced. Further, a second classifier is built using the new weights, (generated from the previous classifier) which are no longer equal. Considering that the weights are no longer uniform, the new classifier will give more importance to the records that were wrongly classified, considering that they are associated with a bigger weight. After the second classifier produces the predictions, the process is repeated again, until the final prediction of the ensemble model is obtained.

Given: $(x_1, y_1), \ldots, (x_m, y_m)$ where $x_i \in \mathcal{X}$, $y_i \in \{-1, +1\}$.
Initialize: $D_1(i) = 1/m$ for $i = 1, \ldots, m$.
For $t = 1, \ldots, T$:

- Train weak learner using distribution D_t.
- Get weak hypothesis $h_t : \mathcal{X} \to \{-1, +1\}$.
- Aim: select h_t with low weighted error:

$$\varepsilon_t = \Pr_{i \sim D_t}\left[h_t(x_i) \neq y_i\right].$$

- Choose $\alpha_t = \frac{1}{2} \ln\left(\frac{1-\varepsilon_t}{\varepsilon_t}\right)$.
- Update, for $i = 1, \ldots, m$:

$$D_{t+1}(i) = \frac{D_t(i) \exp(-\alpha_t y_i h_t(x_i))}{Z_t}$$

where Z_t is a normalization factor (chosen so that D_{t+1} will be a distribution).

Output the final hypothesis:

$$H(x) = \operatorname{sign}\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right).$$

Figure 13: The AdaBoost Algorithm

Source: (Schapire, 2013)

One of the main advantages of AdaBoost is that, in some ocasions, it can be less susceptible to overfitting than other algorithms. However, in order for the algorithm to achieve a good performance it needs quality data, for instance, outliers and noise should be avoied since they could heavily influence the model, this disadvandged imposes a need for an adequate preprocessing step.

4.5.6 Gradient Boosting

Similarly to the ensemble methods reffered previously gradient boosting is also based on the idea that multiple weak models can be combined to get a better performance as a whole. The Gradient boosting algorithm can also be used for predicting not only continuous target variables but also categorical targets and is a method that stands out for its accuracy and speed, particularly with large and data sets, which is an additional help when trying to minimize the bias error of the model. Similarly to AdaBoost, the base models are also built in a sequencial scheme, using their depence to try to lower the errors made by the the previous model.

In this ensemble approach the main goal is to minimize the loss function, which should be specified along with the base classifiers, which in this case, are decision trees. Several different loss functions could be used, allowing a certain degree of flexibility. For instance, logaritmic loss is popular function typically used in classification problems.

Typically, random guess is used to initialize the algorithm, and the gradient descent is calculated. Further, a new weak learner is trained to fit the residuals of the previous model in the sequence, and in that way contribute to minimize the loss function and improve the overall model. The process is repeated until the specified number of iteration is reached. The final model should aggregate the result of the base classifiers and thus provide a stronger performance.

One potential problem regarding gradient boosting is that could overfit the data, since at each iteration, the base models select the optimal solution. One possible and effective solution to overcome this problem is to use some sort of regularization method, that would allow only a selection of the models to be added to the additive model. Nonetheless, this solution has the disadvantage of increasing the computational needs, as typically there is a need to increment the number of iterations.

To sum up gradient boosting involves three elements: A loss function that should be minimized, weak learners that are trained to make predictions and an additive model. The loss function used should take into account the problem that is being solved, for example, regression could use a squared error and classification may use logarithmic loss. Usually decision trees are used as the weak learners and these trees are added one at a time.

5. EVALUATION METRICS

Area under the Roc Curve and accuracy are considered standard evaluation metrics for the evaluation of credit scoring models, which are the most commonly applied in the literature. However, each measure has its positive and negative aspectes. In this case, these metrics ignore the costs of both error types (bad applicants being predicted as good, or vice versa). As said by (Baesens et al., 2003), in real world situations, the cost of a Type II error is much higher than the cost of a Type I error. Because for a credit lending institution accepting a client that ultimately defaults leads to real losses, and rejecting a customer that would not default only constitutes a cost of opportunity. So its possible to conclude that is much more dangerous to falsely classify a customer as non defaulter, than it is to falsely classify as non defaulter. This is the reason why it becomes especially interesting to use the type I and type II errors. Therefore, the present thesis employs all four metrics to obtain conclusions regarding the performance of the final models. Accuracy and AUC given their popularity in the field and type I and type II errors to take misclassification costs into account.

For a binary classification problem, most of these metrics can be easily derived from a confusion matrix as that given in table 2. Through this matrix is possible to determine the number of observations that were correctly and incorrectly predicted, providing a more detailed interpretation of the results for each of the classes.

	Predicted Negative	Predicted Positive
Actually Negative	**True Negative (TN)**	**False Positive (FP)**
Actually Positive	**False Negative (FN)**	**True Positive (TP)**

Table2: Confusion matrix

Source*:*Authors preparation

The above matrix can be interpreted as follows:

- **True Negative**- Predicted as non-default and it is actually non-default;
- **False Positive** – Predicted as default when it is actually non default;
- **False Negative** - Predicted as non default when it is actually default;
- **True positive** - Predicted as default and it is actually default;

Accuracy

The accuracy metric measures the ratio of events correctly classified (positive and negative) over the total number of events evaluated. Accuracy can be calculated as follows:

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

Specificity (True negative rate)

This evaluation metric is used to measure proportion of events identified as negative, on all negative events. The proportion of negative events that are correctly classified. In other words, specificity represents the proportion of non defaults (negative predictions) that were actually correct.

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

Recall (sensitivity/True positive rate)

Recall measures how many positive instances (default) are correctly predicted amongst all positive samples (predicted and actual). In other words, recall answers the question of what proportion of positive predictions (default) were actually correct.

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

Type I error (false positive rate)

Defines the "good" applicants that are predicted to be "bad". In the presence of a type I error, the misclassified good applicants are denied credit and therefore, there is an opportunity cost of revenues for the firm, caused by the loss of potential good customers. This error can be calculated as 1-Specificity, or by the following formula:

$$Type\ I\ error = \frac{FP}{FP + TN}$$

Type II error (false negative rate)

Defines the "bad" applicants that are predicted to be "good". In the presence of a type II error, a default applicant is misclassified as non default. As stated previously this error is considered to be more critical as it represents an actual real loss to the institution, and therefore, we want to minimize it as much as possible. It can be calculated as 1- Recall or by the the formula given as follows:

$$Type\ II\ error = \frac{FN}{TP + FN}$$

Area Under the Roc Curve (AUC)

The receiver operating characteristic (ROC) curve is a widely used method to evalute the performance of a classifier, and it is particularly usefull because of its visual representation. It is a two dimensional plot of the true positive rate (vertical axis) versus the false positive rate (horizontal axis).

The best possible classifier would give a data point in the most upper left corner of the roc space, wich would represent 100% specificity (no false positives) and 100% sensitivity (no false negatives). A point along the diagonal would represent a completely random guess. The diagonal divides the ROC space, so points below that line represent poor results (worse than random), while data points above the diagonal line represent good classification results (better than random chance).

To compare different classifiers, it is common practice to calculate the area under the ROC curve, commonly known as AUC. This metric measures of the ability of a classifier to distinguish between classes (positive and negative) and it is a useful complement to the plot of the roc curve, since it summarizes the classifiers performance into a single value.

When AUC equals 1, the classifier is able to differentiate between the positive and the negative class points perfectly. However, if the AUC is equal to zero, then the classifier would be predicting all instances wrong, all positives would be predicted as negatives and all negatives would be predicted as positives. A value of 0.5 for AUC indicates that the ROC curve will match the diagonal and hence suggests that the classifier has no discriminatory ability, performing no better than random choice.

When the AUC ranges between 0.5 and 1, the chance that the classifier will be able to separate the negative class from the positive class is high. This happens because the classifier is able to predict a higher number of true negatives and true positives than false negatives and false positives.

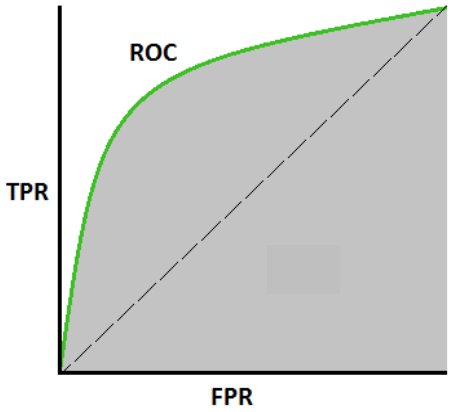


Figure 14: Area under the Roc Curve (AUC)

Source: Medium

6. DATA UNDERSTANDING AND DATA PREPROCESSING

In this section, a complete exploratory analysis of the data and the pre-processing steps are presented. Both were developed through Python, an interpreted, object-oriented programming language.

6.1. EXPLORATORY ANALYSIS

An exploratory analysis refers to the initial investigation on the data. With the intent to discover patterns, spot anomalies, summarize the main characteristics and understand what variables are being dealt with. It is good practice to first explore and try to gather as many insights as possible. These insights will be essential to perform both pre-processing and modeling properly.

The first step consisted of analyzing the data set and its properties. In this thesis, a publicly available open-source credit risk data from Lending Club is used to model credit risk. All loans present in the data set were conceded between 2012 and 2016. Lending Club is a digital marketplace that offers unsecured loans ranging between $1.000 and $40.000. Customers can choose from two loan terms—36 or 60 months, in contrast to many lenders that provide a wide range of loan terms, up to seven years or more.

The data set contains information on 1.485.575 clients and a total of 17 features regarding several characteristics of both the loan and the borrower. A short description of the explanatory variables, can be seen in table 2.

Variable	Description	Type
Id	A unique assigned ID for the customer	Numerical
addr_state	The borrower's state of residence	Categorical
term_months	Loan term. Values are in months and can be either: 36 (3 years) or 60 (5 years) months	Numerical
home_ownership	The home ownership status provided by the borrower. The values can be: RENT, OWN, MORTGAGE or OTHER	Categorical
purpose	Is the aim of the loan. A category provided by the borrower for the loan request	Categorical
verification_status	Indicates if LC verified the income or not	Categorical
application_type	Indicates if it is a joint application with two co-borrowers or is an individual application	Categorical
emp_length	The number of years the applicant has been employed. The values are between 0 and 10 where 0 means less than one year and 10 means ten or more years	Numerical
loan_amnt	The listed amount of the loan. The values are presented in dollars	Numerical

dti	Debt to income ratio. Represents the total monthly debt payments divided by the borrower's monthly income.	Numerical
revol_util	Revolving line utilization rate	Numerical
issue_year	The year in which the loan was conceded	Numerical
annual_inc	The annual income declared by the borrower during the registration process	Numerical
total_acc	Number of credit lines under the borrower's name	Numerical
int_rate	Interest rate	Numerical
delinq_2yrs	The number of incidences of delinquency in the borrower's credit file for the past 2 years	Numerical
default_loan	Target variable (0: non- default, 1: default)	Categorical

Table 3: Data set variables description

Source: Authors preparation

After analyzing the metadata, the variables "Id", "Issue year", and "addr_state" were removed before any further steps were taken, because they did not present any type of relevant information. "Id" was only an identifier, it did not represent a characteristic of the borrower or the loan. The variable "addr_state" was removed because it only contained the state in which the borrower lives and it is not the intent to make an analysis in terms of location. And the "Issue year" only provided an indication of when the loan was made, which once again is not relevant for the modeling problem. After this first step the dataset was reduced to 14 variables , one target variable and thirtheen independent features.

In the present EDA the variables were analyzed one by one, in order to try to understand what values can the variable assume and if any of them were missing, check the existence of outliers/strange values and duplicate observations, understand the distribution of the data as well as some statistical insights, e.g mean, maximum and minimum values.

For these matter it was easier to divide the variables between metric features and non metric features and apply methods of data visualization. The metric features were investigated using a histogram (see appendix A1) and for the non metric features and the discrete metric features bar charts were plotted (see appendix A2). Pandas profiling was also used as a resource in order to facilitate some of the analysis.

At first it was possible to get the following general insights: The data set was unbalanced, the number of customers (79.22%) that didn't default was much higher than the number of customers who defaulted (20.78%). This was expected, given the nature of the problem, since repaying a loan is more likely than defaulting. Most of the borrowers had been employed for ten years or more (33%), when they applied for a loan followed by 2 and 3 years of employment. Most of the loans were to be paid in three years (76%). Only a very small fraction of the applications were done with two co-borrowers, mainly the loans were individual. Over one third of the applicants were paying a mortgage, and the

main purposes for applying for a loan were debt consolidation and credit cards, which represent around 82% of the observations.

The borrowers earn on average 75.598 dollars per year, and the average debt to income ratio is 18.38%. In terms of loan amount the minimum and maximum values are inside the limits allowed by Lending Club and the average amount conceded is 14.492 dollars. In terms of interest rate, the average value is 13.18% .

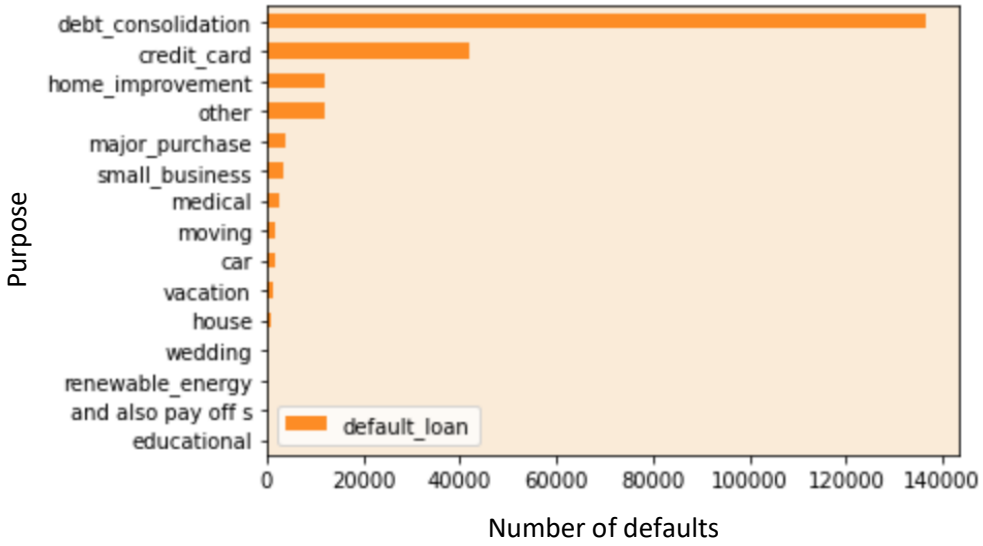
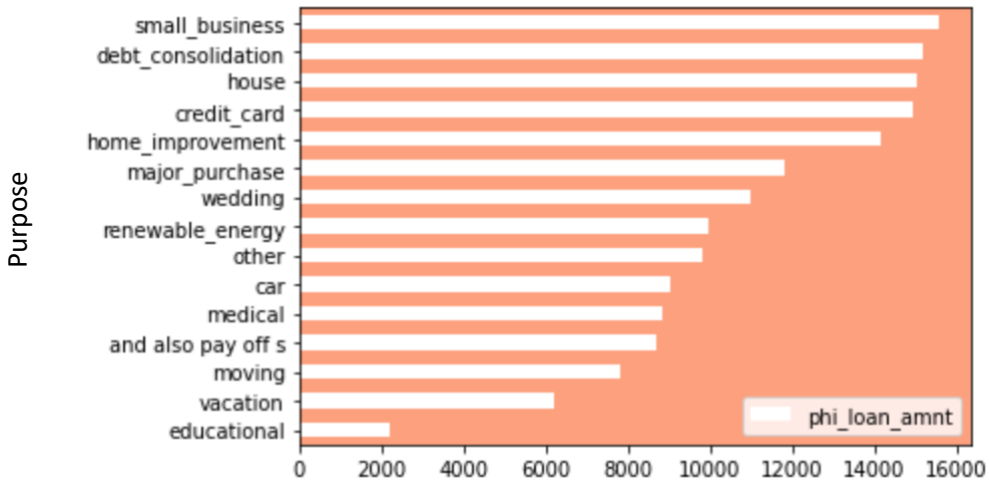


Figure 15 : *Number of defaults per loan purpose*

Source: Authors preparation

Debt consolidation constitutes the loan purpose where most defaults are verified, close to 140000, followed by credit card, and home improvement, with approximately 6000 and 2000 cases of default, respectively. Considering the fact that the two categories where most defaults are verified are also the two most common purposes for loan application in the data set, these conclusions were expected and it is not possible to understand if there is a purpose that stands out from the remaining.

With the intent to understand if these categories were the ones where the larger loans were conceded, which could translate into the bigger losses for the company, the following chart was constructed:



Loan amount (average value)

Figure 16: *Average loan amount per loan purpose*

Source: Authors preparation

Loans provided to small businesses are on average larger than the ones provided to the remaining clients, close to 16000$. Closely followed by debt consolidation, which represents the category where most loan defaults occur, house and credit card, where on average clients borrow 15000$. Loans taken for educational purposes are on average smaller than the remaining loans, with a value close to 2000$.

Regarding missing values it was possible to identify them in five variables: ”emp_lenght”, which had around 5.6% of values missing. In ”dti” and ”revol_util” less than 1% of the data was missing, and in the case of the variable ”application_type” only one observation was not present. Regarding “home_ownership”, it presented values that are not allowed categories, namely: “Any” and “None” and therefore were considered to be missing values as well.

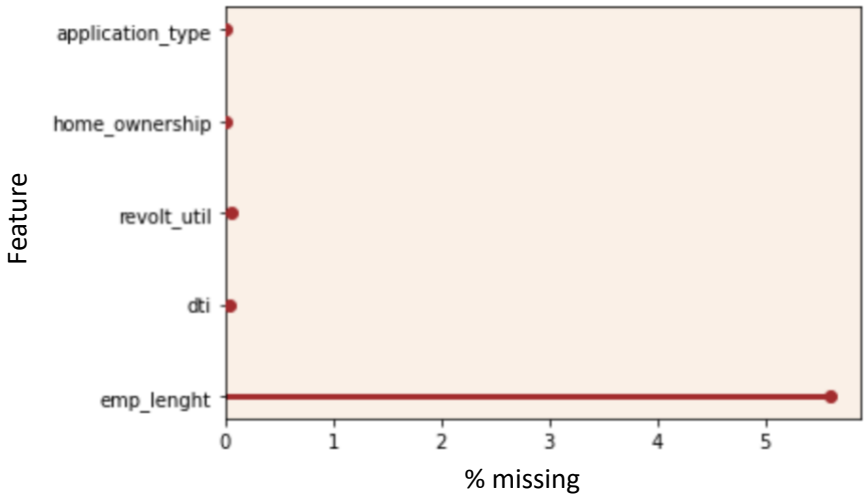


Figure 17: % of missing values

Source: Authors preparation

Spotting outliers in the dataset is another crucial step in EDA. For this matter the table of statistics (see appendix B1), the boxplots (see appendix A3) and the distribution plots (see appendix A4) were used as tools to auxiliate the analysis. Some of the metric features presented a few extreme values that were very distant from the mean and for that reason were considered to be outliers. For instance, in the variable that indicated the annual income of the client, the minimum income detected was of one dollar and the maximum was of 9.5 million dollars, these values are extremely far away from the average value, in fact, an income of one dollar is completely unrealistic . In the case of the variable that indicated the debt to income ratio of the client, it was possible to spot outliers, values that were 40% or above, because LD only allows borrowers with a dti inferior to 40%. In the metric “revol_util” outliers were also spotted, since the value could not be inferior to 0% or superior to 100%.

As for the categorical variables, the analysis of the bar chart bellow allowed to identify outliers in the variable “purpose”, the variable presented several categories assigned to less than 1000 observations, in particular, two categories (“educational” and “and also pay off”) were assigned to only one observation, therefore were considered outliers as well.

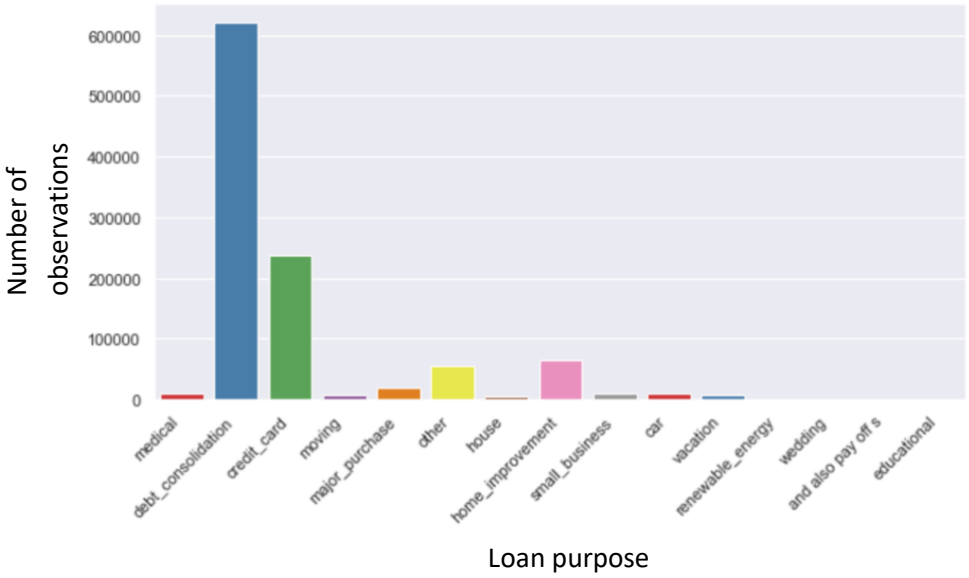


Figure 18 : *Number of loans conceded per purpose*

Source: Authors preparation

Correlation and Pairwise relationship

Performing an individual analysis of each variable during the exploratory data analysis is not enough, it is also necessary to understand how the variables in a dataset interact with respect to each other. The last step of the EDA was to try to understand if there were any type of relationship between variables, e.g linear or monotonic relations. There are several different ways to analyze these relationships visually, but one of the most common and practical methods is to scatter plots. However, through this first approach it wasn’t possible to detect any particular relationship. The images (see appendix A5) illustrate what the relationships look like at different degrees of strength.

In order to try to obtain a deeper insight regarding the relationship between the varibles, a correlation matrix was ploted, wich can be seen below in figure 19. This correlation analysis seeks to identify (by a single number) the degree to which there is a relation between two variables. The chosen method was the Spearman rank correlation, which depicts monotonic relationships between two continuous or ordinal variables. In the heatmap below, pink colors represent positive relations, (when one variable increases the other increases), while blue colors represent negative relationships (when one variable increases the other decreases). The more intense the color is, the stronger the relationship. Different suggestion have been made in order to translate the interpretation of the correlation values into labes like “weak”,“moderate,” or “strong” relationship, howeverthere is no general consensus. Even so, considering a general rule of thumb, all the relationships can be interpreted as weak or negligible.

Nonetheless, one exception is represented by a moderate positive correlation that can be detected between the variables annual income and loan amount (0.5). This dependency is expected and justifiably: the higher the income, the more likely it is that this client can apply for a bigger loan. On the contrary, the smaller the income the less money that person is capable of borrowing.

Figure 19: Correlation Matrix (heatmap)

Source: Authors preparation

6.2. DATA PREPROCESSING

Data preprocessing is normally mandatory and constitutes a critical step when developing classification models. In machine learning is by far one of the most time consuming and important stages out of the all process. Essentially, it refers to the task of transforming the initial raw data to make it suitable and understandable for the algorithm that will be used in the modeling process. If the data is not correcty prepared, then its possible that the algorithm will not be able to train, or even if it does it will report errors. In the best scenario the algorithm will work, but the results will not be considerd accurate.

Unfortunately, real world data is likely to be affected by several factors, such as, outliers, noise, inconsistant data, missing values among other problems. Thus, considering that low quality data will lead to models with poor performance and consequently inadequate results. Therefore, data

preprocessing will aim at overcoming these problems and help the models achieve better predictive power.

Pre-processing was divided in three main tasks:

- Data cleaning
- Data transformation
- Data reduction

The first task, data cleaning, includes operations that detect, correct, filter and remove errors and inconsistencies present in the data to improve its quality. Specifically, it will consist of handling missing values, removing duplicate observations, fixing structural errors, detection and removal of outliers.

6.2.1. Imputation of Missing Values

As mentioned by Acuña & Rodriguez (2004) the performance of a classifier can be highly affected by the presence of missing values. So it's critical to handle this situation.

The variable “application_type” had only one missing record. This observation was removed from the dataset as it was almost insignificant, considering that there are over one million records. It represented much less than 1% of the data, which is generally considered trivial.

The variable “home_ownership” presented 107 values missing which were imputed using a measure of central tendency, namely, the mode represented by the category “mortgage”.

For the variables “dti” and “revolt_util”, also less than 1% of the observations were missing. For the first metric the values were imputed using the median. The reason behind it was the fact that it is less affected by outliers and the distributions were skewed, therefore, instead of the mean, the median would be a better representation of the central tendency. For the “revolt_util” metric as it followed a normal distribution, the mean was chosen as an imputation measure.

Finally, the variable regarding employment length, it contained 58536 missing values which corresponds to 5.6% of the records. Considering that this percentage was a little high a more sophisticated method was required and a KNN imputer was employed to overcome this situation. KNN imputer is an instance-based algorithm. Every time a missing value is found in a current instance, the algorithm computes the K-nearest neighbors and a value is imputed. Considering that the algorithm was only applied to the metric features, the prediction was taken as the average of the k most similar samples. For this algorithm, the number of neighbors chosen was 3 (K=3) and the distance measure used to calculate the neighbors was the euclidean distance. After performing this step of data cleaning from the total 59273 values missing, only one was removed and the remaining were imputed.

6.2.2. Dealing with outliers

Outlier detection, it is the process of finding data points whose behavior is very different from the expected. An outlier is an individual point of data that is distant from the remaining points in the data. This anomaly in the dataset may be caused by a range of errors in capturing, manipulating or

processing data, but in many cases outliers occur naturally. Outliers can skew overall data trends, so its detection is an important step of data preparation.There are few commonly used outlier detection methods in machine learning. In this case, IQR, Isolation Forest and visualization methods, boxplot, distribution plot, and bar chart were the methods employed.

In the case of the categorical features, only the variable “Purpose” was considered to have outliers. Two categories were assigned to only one observation, considering a dataset of over one million observations, these categories were removed. The category energy was also removed as it represented less than 1000 observations.

Regarding the variable “dti” every value that was equal or superior to 40% was considered an outlier, since LC only accepts clientes with a debt to income ratio inferior to 40%. As for the metric “revol_util”, only values between 0 and 100 were acceptable, therefore any value outside this interval was also considered an outlier.

The outliers in the variable “annual income” and “loan amount” were detected and removed through the IQR method. For this matter the q3 (75th percentile) and q1 (25th percentile) were computed in order to obtain the inter quartile range (q3- q1), then it was possible to calculate the lower and upper bound:

Lower Bound = q1–1.5*IQR

Upper Bound = q3+1.5*IQR

However, considering the multiplication of IQR by 1.5 it would lead to the removal of a big proportion of data, so this value was adapted to 7, in order to remove a smaller number of observations. Any value below the lower bound and above the upper bound are considered to be outliers. This allowed extreme values, very distant from the average, including for instance, an unrealistic annual income of 1$, to be removed.

Isolation Forest was the method applied to remove outliers from the remaining metric features. Isolation Forest is an algorithm that provides an anomaly score to the data points. It does this by repeatedly spliting a data point by random attributes until it is isolated. Outlier data will generally need less partitions to achieve isolation, because they are a drift from other data points. The repeated partitioning can be seen as a tree structure and hence the name Isolation Forest. The hyperparameter contamination was adjusted in order to remove a smaller percentage of the data points, since the default value would lead to an elimination of more than 5% of the dataset which is not desirable. After all the methods employed a total of 4.1% of the data was removed.

The second task, is data transformation and In the present thesis three tasks are performed: Feature engineering, data normalization and encoding of categorical variables.

6.2.3. Feature Engineering

Feature engineering facilitates the machine learning process and is sometimes able to increase the predictive power of the algorithms by creating features from raw data. For that reason a new feature containing the amount of interest paid by the borrower was created.

When a client applies for a loan, an additional amount has to be paid, besides the loan itself. “Interest is the price a borrower pays for the use of money they borrow from a lender/financial institution” (Crowley, 2007 cited by Njoki, 2014). The level of interest depends on the interest rate, the loan amount and the duration of the loan. In some cases the portion of interest paid can be considerably high, particularly, if the interest rate is high enough the amount of interest can be superior to the amount of the loan, which can affect the capacity of the borrower to fulfill its obligations.

Lending Club charges simple interest, (and not compound) and there’s no prepayment penalty for borrowers who want to save money by paying off their loans early. So when analyzing the dataset it was clear that this variable was not present but it could be created. Considering the simple interest depends on the principal balance, interest rate, and time period, the amount of interest the borrower would have to support besides the loan amount. is given by the formula below:

$$I = P * r * t$$

- I is the total interest amount
- P is loan amount
- r is the interest rate
- t is the loan term

6.2.4. Encoding categorical variables

Encoding categorical data is another crucial step, most machine learning models work with numerical data, for that matter variables containing strings need to be encoded in order to be given to the model. In a general definition encoding is the process of converting categorical features into a numerical (integer) format so that the data with converted values can be provided to the models. Once again several methods have been proposed in the literature to solve this problem. In the case of our datasat, the categorical features were not ordinal, they were nominal. Which means that their values were not ordered, one value was not considered to be more important than the other. For this reason, one hot encoder was the method chosen to convert the categorical features.This is a frequently used approach that creates a binary column for each category, where 0 indicates non-existent while 1 indicates existent. For instance, in the variable “home_ownership”, four categories were allowed, namely, mortgage, rent, own and other. As stated this is a categorical variable and string values cannot be given

to the algorithm, so one hot encoder will create four new variables (dummy variables), each one of those variables represents a category mentioned previously. For each observation, only two possible values are allowed: 0 if that category is not present, and 1 if that category is present. Though this approach eliminates the hierarchy/order issues it has the disadvantage that adds more columns to the data set.

The following table presents the new columns added to the dataset after the implementation of one-hot encoder:

New varible	Previous variable
x0_car	purpose
x0_credit_card	purpose
x0_debt_consolidation	purpose
x0_home_improvement	purpose
x0_house	purpose
x0_major_purchase	purpose
x0_medical	purpose
x0_moving	purpose
x0_other	purpose
x0_small_business	purpose
x0_vacation	purpose
x0_wedding	purpose
x1_Not Verified	verification_status
x1_Verified	verification_status
x2_Individual	application_type
x2_Joint App	application_type
x3_MORTGAGE	home_ownership
x3_OTHER	home_ownership
x3_OWN	home_ownership
x3_RENT	home_ownership

Table 4: One- hot enconder variables

Source: Authors preparation

6.2.5. Normalization

Data normalization is a pre processing task that involves resclaling the values that a given feature can assume. This is necessary because features with great numeric values could dominate features with small numeric values. If this happens, the model creates a bias towards the feature with greater values, making their contribution to the final output more important. Through normalization, this problem is

minimized because, feature values are transformed to a common range, this allows each feature to make a uniform contribution. This is particularly importante when the relative importance of each feature is not known.

The significance of the data normalization for building accurate predictive models has been explored for various machine learning algorithms such as Artificial Neural Networks and Support Vector Machines which will be employed in the modeling phase. Many studies have highlighted the importance of normalization. This step when well performed can improve the quality of the data and hence the quality of the model. Min–Max Normalization (MMN) is one of the most popular methods to normalize data. The method scales the un-normalized data to a predefined lower and upper bounds linearly. By doing so, all the features will be transformed into the range [0,1] meaning that the minimum value of a feature/variable will be 0 and the maximum value will be 1.
The normalization equation is given as follows:

$$Zi = \frac{xi - \min(x)}{\max(x) - \min(x)}$$

where min denotes the minimum value observed for that variable and max denotes the maximum value of that feature.

Variables that are not measured with the same scale can contribute to a biased model, as referred previously. This can happen for instance, between the variable regarding employment length which has values between 0 and 10, in contrast to the variable annual income that has values between 30.000 and 300.000, without normalization the algorithm could give higher importance to the latter, given the fact that the values are much higher. This scaling method is useful when the data does not follow a normal distribution and when the data set does not contain outliers, which in this case, the outliers were removed previously.

6.2.6 Data Imbalanced

As stated previously during the explanatory analysis it was possible to detect that the data set was unbalanced, which means that the target variable has an uneven distribution of observations. We are in the presence of an unbalanced dataset when one or more target classes represents a much higher number of records (majority class) than the remaining classes (minority class). According to (Chawla et al., 2002) it's very common that real world data is afected by this phenomenon. In this case, the target variable has more observations of the class "non-default" (79.88%) than of the class default (20.12%). Which is expectable given the nature of the problem.

Unbalanced data constitutes a problem. This is because, due to the disparity of classes the classifiers tend to be biased towards the majority class and consequently perform poorly on the minority class. Thus, before fitting the model over the training dataset and forecast classes over the testing dataset, it is necessary to balance the data. Until now several solutions have been proposed in order to try to overcome this problem. Some of them aim at resampling the data, such as undersampling, which removes records of the majority class, or oversampling, wich adds records from the minority class. The

main goal of the proposed solutions is to ensure that all target classes are represented in the same proportion.

SMOTE- *Synthetic Minority Oversampling Technique* is an improved method of dealing with imbalanced data in classification problems. This algorithm is capable of overcoming class inbalancement in a more robust way than just by simply performing oversampling or undersampling. Because in this case, it is not necessary to remove observations from the majority class (undersampling) which would imply losing information. Thus, is also not necessary to create duplicate observations of the minority class (oversampling) which could later translate into a problem with overfitting. This algorithm is able to generate artifical samples based on the linear interpolation of both classes

According to Hu & Li (2013) the functioning of the algorithm can be described in a simple way. For each observation of the minority sample the algorithm finds its nearest neighbors (from the minority class). Following this step, the algorithm chooses one of the selected neighbors and generates the new artificial record by interpolating between both observations, as pictured in appendix A6. For these reasons, SMOTE was the considered algorithm to balance the data. Allowing the construction of 587.764 observations of the minority class and the final data set to have the same proportion of classes in the target variable.

BEFORE SMOTE

Target Class	Number of observations	%
Non-default (0)	802.599	79.88
Default (1)	202.182	20.12
	1.004.781	**100**

AFTER SMOTE

Target Class	Number of observations	%
Non-default (0)	802.599	50
Default (1)	802.599	50
	1.605.198	**100**

Table 5: Comparison in the number of observations of the target class before and after SMOTE

Source: Authors preparation

In the last task, data reduction, as the name suggests, the data is reduced. During this phase both the independent features and the number of records can be reduced, but in the present thesis, only feature selection was conducted.

6.2.7. Feature Selection

When developing a machine learning predictive model it is importante to take into consideration that variable selection is a core concept, whose goal is to lower the number of features of a given dataset. Liu et al. (2010) highlighted that the benefits of feature selection are three-fold: Build simpler models which provide a better understanding of the underlying process that generated the data, Improve model performance, and reduce the computational cost of modeling.

In general, feature selection methods can belong to one of three families: filter-based methods, wrapper based methods and embedded methods. Regarding filter based methods they are as expected applied before the learning algorithm and the main goal is to rank the features in order of importance, at the end select only those with the highest scores. Wrapper methods also assign scores to the features, but in this case by using the same algorithm that is employed in the modelling phase. Regarding embedded methods they combine feature selection with the learning algorithm.

In this thesis, a wrapper selection technique called Recursive Feature Elimination (RFE) is utilized.
RFE is regarded as a backward selection technique for the predictors. The basic idea behind this method is that it starts by building a model on the entire dataset, and simultaneously assigns feature importance scores to each of the attributes. Based on those scores, the ones that are considered to be least important predictor are removed in a recursive way. The model is rebuilt, and feature importance's scores are recomputed again. That procedure is recursively repeated until a desired number of attributes is reached.

An important hyperparameter for the RFE algorithm was the number of features to select. In practice, we don't know what is the best number of features to select, instead, different values between 1 and the total number of features were tested until the best score was obtained. In this case, for each model the feature selection method was applied with the corresponding algorithm. And for each model 25 variables were used (6 were removed). All the algorithms gave less importance to the features related with the purpose of the loan, the type of application and the duration of the loan, which were the most commonly removed variables among all algorithms.

7. IMPLEMENTATION OF THE MODELS

The modelling phase consists of training the machine learning models, fine tuning its hyperparameters and testing it on a validation set.

As stated before, the present thesis deals with a binary classification problem. Where the target feature assumes the value one, in case of default, or the value zero , in case of non-default. For this reason this part of the project aims at constructing five predictive models, which are capable of accurately predicting the target variable. The first model implemented is logistic regression, which is the baseline in the industry. Logistic regression will be compared to four ensemble classifiers: Bagging represented by random forest, Boosting represented by Adaboost and Gradient Boosting and Stacking, represented by a model with LR, NN and SVM as base learners and DT as a meta learner. All the models were constructed in Python.

7.1. HOLD- OUT METHOD

As well as being required to select appropriate performance measures to evaluate trained models, it is also necessary to ensure that an appropriate evaluation design is used. In this thesis, the evaluation design chosen was the hold-out method. A schematic of this method can be seen in appendix A7. Instead of using the entire dataset for training and evaluation, different sets called test set and validation set were set aside/separated (and, thus, hold-out name) from the entire dataset and the model is trained only on the training dataset, as the name suggests.

The hold-out method is used for both model selection and model evaluation and was chosen given the fact that the data set was large, containing over one million of records, and is much less computational expensive than cross validation.

In this approach the data was split in three different datasets, namely:

- Training dataset, with 60% of the data;
- Validation dataset, with 20% of the data;
- Test dataset, with 20% of the data.

Training set: The training data set contains 60% of the data, and will be used to build and train the model, that is to fit the parameters of the classifier. It is important to mention that the splitting process of the data into train and validation must be executed carefully, in a way that the proportion of examples in each class observed in the original dataset is preserved. Otherwise, the data given to the model would once again be unbalanced. For this to be accomplished, we use stratified sampling, which is defined at the moment of the split.

Validation set: The validation set represents a small portion of the data set (about 20%), and like the test set is also held back from training the model. Its purpose is to fine tune the hyperparameters of the classifiers as well as check the existence of overfitting. If an AI algorithm is well trained it should present a good performance on the training and on the validation set, which means that the algorithm is able generalize well on unseed data (validation set). If the model performs well on the training set but then fails to generalize on the validation set, it means that the model is overfitting. This can also me seen as memorizing, because the algorithm only memorizes the outputs instead of actually learning the distribution of the data. On the contrary, if a model is unable to perform well on both the training and validation set it can be said that the model is underfitting.

Test set: Preferably, the model should be evaluated on examples that were not used during the learning or fine tuning stage, so that the metrics provide a realistic and unbiased view of the performance. If the performance of the model is evaluated on the same data it was trained with we will have "peeking". Because the model has already seen that data, it's probable that it will perform very well. New data should be used, therefore the need for a test set, that was kept completely separate. This test set contains the observations that the model has never seen and will be used only for the model evaluation, in order to obtain an unbiased estimate.

7.2. FINE TUNE HYPERPARAMETERS

The architecture of a machine learning model can be defined by what are called the hyperparameters. Thus, the process of uncovering the optimal architecture can be referred to as hyperparameter tuning. Hyperparameters are set before training, however, oftentimes, it is not possible to immediately know what the ideal architecture should be for a given model, thus it's necessary to explore a range of possibilities in order to improve the performance of the classifiers as much as possible.

The more hyperparameters we try to tune, the slower and more computationally expensive the process is, so trying to find all the best hyperparameters is almost impossible. Therefore, only a subset of one to two hyperparameters per algorithm are chosen. According to (Bergstra & Bengio, 2012) manual search and grid search are the most commonly applied strategies to fine tune hyperparameters and both are employed to search the optimal values of the proposed classifiers.

Manual Search as the name indicates is done manually and without the use of an algorithm.The idea is to first take big jumps in order to understand what is the best value and then take small jumps to focus around that specific value. This method was applied on the ensemble techniques considering that using grid search was very computationally expensive.

Grid search is a simple algorithm and was used to fine tune the hyperparameters of logistic regression. Basically, it divides the domain of the hyperparameters (which are set manually) into a discrete grid. Then, it tries every combination of values on the grid, and determines which combination of values provides the highest accuracy, using cross validation.

Table 6 summarizes the searching space of the learners. Firstly, for the LR model, two hyperparameters were tested, namely, the solver and the penalty. The penalty can sometimes be helpful and it helps to regulate and reduce overfitting, while reducing the generalization error. The solver is the algorithm

used in the optimization problem and depends on the penalty, as some penalties do not work with some solvers. In AdaBoost 100 stumps are constructed and the learning rate is set to 0.8. There is a trade off between these two parameters, so when changing the default number of trees, the learning rate had to also be adjusted. A lower learning rate decreases the contribution of each regressor. In relation to GradientBoosting 150 trees are built and max_depth is tested for the values 2,3 and 4, which indicates the deph of the tree. The deeper the tree, the more it splits it has and the more information about the data is able to capture. Regarding random forest the number of estimators was tested for 10, 100 and 1000 trees where 100 trees was considered the optimal number. max_features was also tested, and "sqrt" was chosen has the optimal solution . This hyperparameter represents the size of the random subsets of features to consider when splitting a node, which in this case will be the square root of the total number of features. Regarding the last ensemble, stacking, the hyperparameters of the base learners were left with default values, and the meta learner, was only tested for different values of the "criterion" which represents the function used to measure the quality of a split.

Classifiers	hyperparameters	Optimal combination
Logistic Regression	solver ∈ {lbfgs, sag, saga} penalty ∈ { none, l2 }	solver = lbfgs penaty = none
Adaboost	n_estimators ∈ {30,50,100} learning_rate ∈ {0.8 ,1, 1.5}	n_estimators= 100 learning rate=0.8
GradientBoost	max_depth ∈ {2,3,4} n_estimators ∈ {80,150,200}	max_depth=4 n_estimators=150
Random Forest	n_estimators ∈ {10, 100, 500} *max_features* ∈ {sqrt, log2 }	n_estimators =100 max_features= sqrt
Stacking	criterion ∈ {gini,entropy,log_loss }	criterion= gini

Table 6: Search space of hyperparameters settings

Source: Authors preparation

7.3. CHECKING THE EXISTENCE OF OVERFITTING

As said previously the validation set allowed not only the tuning of the hyperparameters but it was also used to check the existence of overfitting. Overfitting is a criticial issue when dealing with supervised machine learning techniques. This problematic can inhibit the model from generalizing well on data that it has never seen before (Ying, 2019). A model that suffers from overfitting has difficulty dealing with the information on the testing set, which is different from the one it was trained with, this causes the model to perform perfectly on the training set, while fitting poorly on the validation set.

Overfitting can be identified by checking different validation metrics. When the model is affected by overfitting, these metrics normally increase until a point where they start declining or stagnate. In this case the classification report available on sklearn is used to compare the performance of the model on the training set and on the validation set. This report compares the models based on the following metrics: precision, recall and F1 score. From the classification reports (see appendix B2 to B6) it is possible to see that there is no significant difference in the performance of the model on each set, therefore concluding that there was no overfitting.

8. RESULTS AND DISCUSSION

In this chapter, the obtained results from the previously be presented and discussed.

8.1. PERFORMANCE OF THE OVERALL MODELS

In this section a comparative analysis will be made between the supervised learning models based on the performance metric discussed previously. The results will be discussed in two different approaches, first a general overview of the performance is presented. Further a comparison is made based on the 4 metrics, in order to obtain conclusions regarding the research question.

	Accuracy	Type I error	Type II error	AUC
Logistic Regression	0.9041	0.0032	0.3695	0.9932
AdaBoost	0.9069	0.0064	0.3407	0.8800
Gradient Boosting	0.8666	0.0789	0.2892	0.8695
Random Forest	0.9065	0.0019	0.3551	0.8800
Stacking	0.7439	0.2614	0.2411	0.6987

Table 7: Model performance summary

Source: Authors preparation

In terms of accuracy, the proportion of correctly classified loans (good and bad), which measures the predictive power of the model, AdaBoost presents the best performance with an accuracy of 0.9069, Random Forest and Logistic regression come to a close second and third place, with 0.9065 and 0.9041, respectively. Stacking presented the lowest value with only 0.7439.

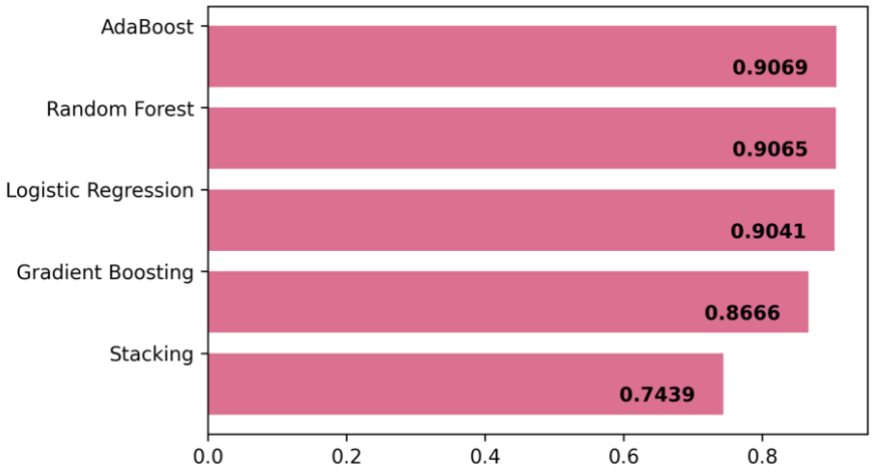


Figure 20: Models' accuracy

Source: Authors preparation

Regarding Type I error (good customers misclassified as bad), Random Forest was able to obtain the best performance 0.0019, Logistic regression presented an error of 0.0032 followed by AdaBoost with 0.0064. Gradient Boosting and Stacking presented higher values of Type I error, with 0.0789 and 0.2614 respectively.

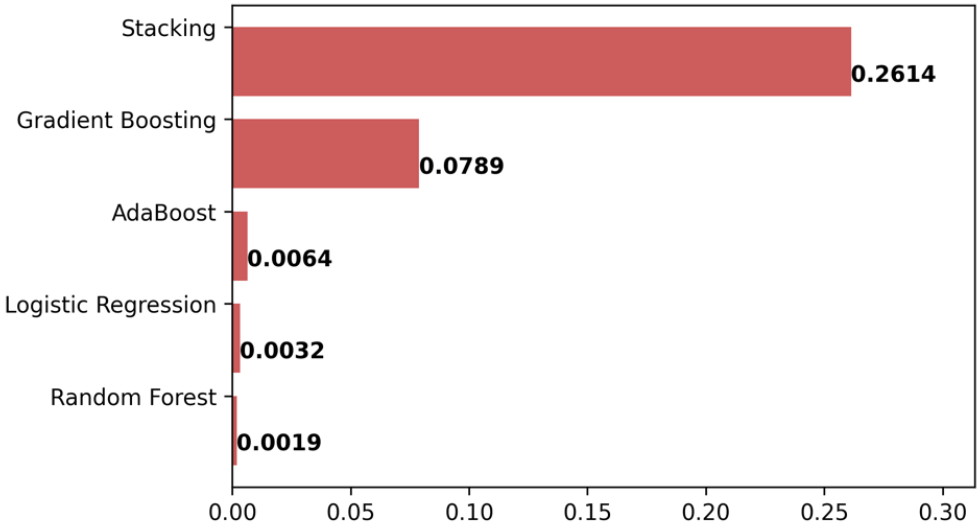


Figure 21: Models' type I error

Source: Authors preparation

For the metric Type II error (bad customers misclassified as good), stacking presented the lowest error with 0.2411, followed by Gradient Boosting. In contrast with the previous metrics, AdaBoost and Random Forest presented higher errors, with 0.3407, 0.3551 respectively. Logistic regression had the worst performance, predicting the highest number of false negatives amongst all classifiers (0.3695).

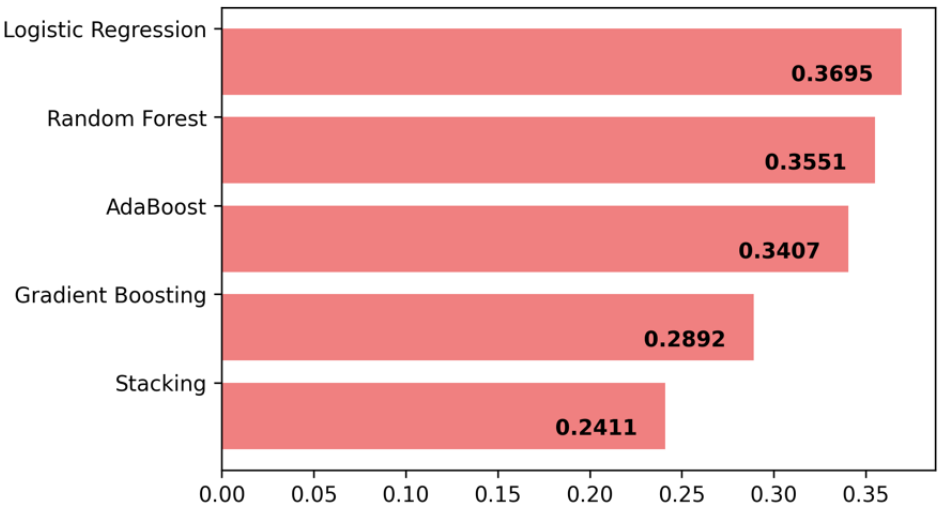


Figure 22: Models' type II error

Source: Authors preparation

Logistic regression presented a very good result in the metric AUC with a value of 0.9932, once again followed by AdaBoost and Random Forest as close challengers presenting a value of 0.8800 each. Gradient boosting scored 0.8695, and stacking provided the worst performance in terms of AUC with only 0.6987. For a visual representation of the performance of each model, their ROC curves were ploted, allowing the support of the mentioned analysis (see appendix A8 to A12).

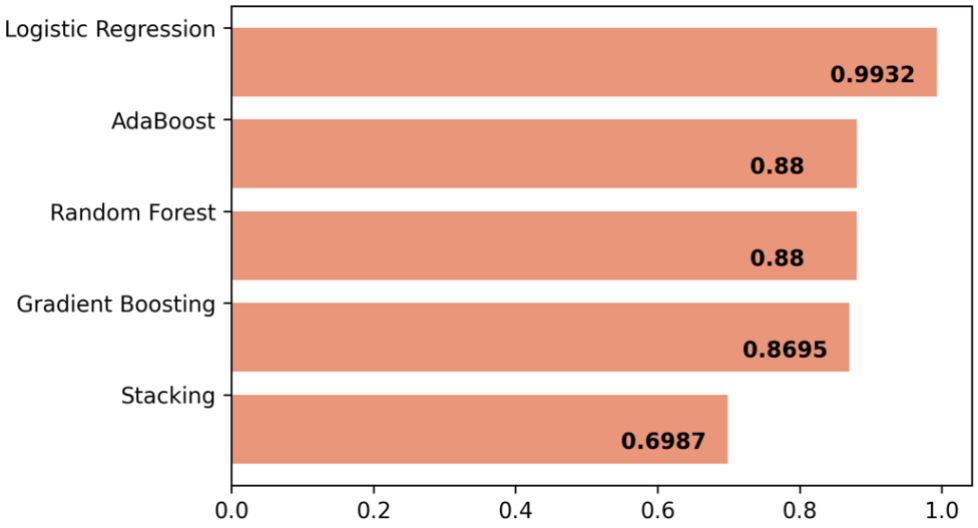


Figure 23: Models' AUC

Source: Authors preparation

8.2. DISCUSSION

Our objective in this study is to make a comparative performance evaluation among ensemble learners, i.e., AdaBoost, Gradient Boosting, Random Forest, Stacking, and the industry standard, Logistic Regression. The metrics chosen to evaluate the models play an important role, as they influence how the performance of machine learning algorithms can be compared and measured. To validate our models and to reach a reliable and robust conclusion, four performance indicator measures were chosen, namely, accuracy, AUC, type I and type II errors.

Logistic regression presented a very good performance in terms of accuracy, type I error and AUC, however it performed very poorly in terms of type II error, predicting too many defaulters as non defaulters. Which is not desirable considering that this is a very critical error.

AdaBoost and Random Forest presented a very stable performance, with high values of accuracy and AUC and low type I error, even so with a type II error relatively high. Gradient Boosting presented a stable performance as well, although slightly worse.

Stacking presented the biggest discrepancies with relatively poor performance in terms of accuracy, AUC and type I error when compared to the remaining classifiers, nonetheless it presented the lowest type II error, which is considered to be the most important as the bank loses some or all of not only the interest but also the repayment of principal.

So if the goal is to choose a model based on the number of correct predictions, and the importance lies on having a model with a good discriminatory ability logistic regression, AdaBoost and Random Forest performed relatively well.

Nonetheless, It is clear that the misclassification costs associated with type II errors are much more significant then the one associated with type I errors. In fact, According to West (2000), type II errors are five times more critical than type I errors. As the results show in figure 22 the stacking model has the lowest Type II error when compared to the other approaches. Therefore, we can conclude that this ensemble method can successfully reduce the possible risks of extra losses due to misclassification costs associated with Type II errors when compared to the remaining learners. So, if the goal is to minimize type II error, and predict the least possible number of false negatives, then stacking would be the right choice, considering however that we would be losing accuracy.

To conclude, a balance is necessary between the four metrics considering that no model performed better across all of them. In terms of trade off between metrics AdaBoost proposes as a good candidate, as it is in the top 3 across all metrics. In particular, it delivered the highest accuracy amongst all classifiers. So if the goal is to obtain accurate probabilities of default, considering however the costs of misclassification then AdaBoost would be the method of choice.

The results presented above reaffirm the importance of using different metrics when comparing credit scoring models. For example, choosing a classifier based solely on one metric, for instance AUC, the best method would be Logistic Regression, however we would ignore the high cost of the type II error. And a high FNR presents an even bigger problem than a high FPR because it is more dangerous to falsely classify a customer as non defaulter. The same thing would happen if we were to choose a model only based on Type II error, then stacking would be a better choice, ignoring however that it presented the lowest number of total correct predictions.

8.3. FEATURE IMPORTANCE

The use of machine learning techniques are often criticized because of their lack of interpretability, and they can at times be called a “black box”. That is one of the reasons why models like logistic regression are favoured. Because the coefficients of the logistic regression can provide explicit information about the statistical relationships. Understanding not only which variables have an impact on the final decision, but also how it is making the prediction is very important. In order to minimize this gap and try to understand what are the main features that affect the output of the model. For that reason a global importance plot was produced and is displayed below for the model AdaBoost.

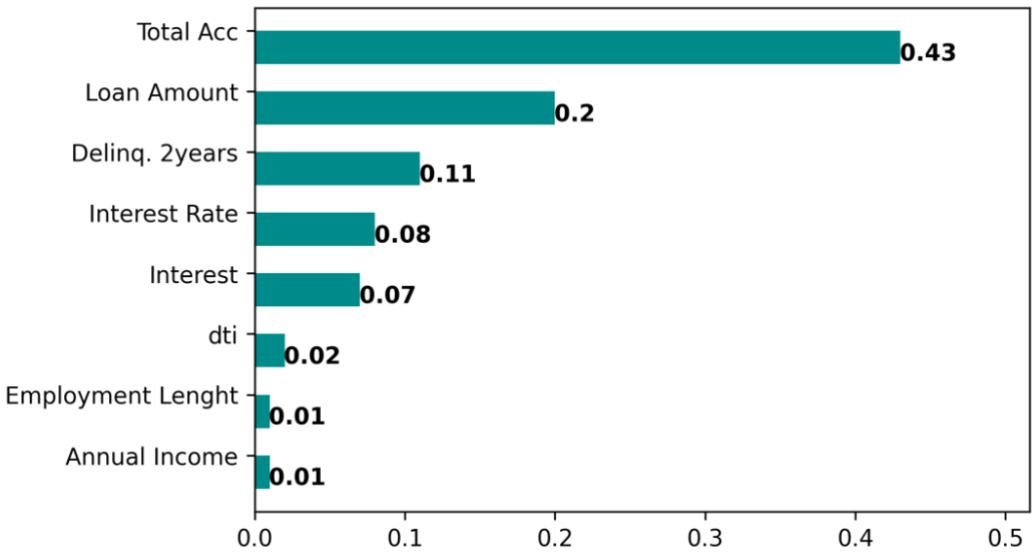


Figure 24: Global feature importance

Source: Authors preparation

The plot was built with the property “feature_importances_” available on sklearn and allows us to have a notion of what features are more important to the final decision. Considering that the higher the value, the more important they are. The importance of a feature is computed using the the Gini importance.

To predict future loan defaults, the most important feature to be considered is “total_acc” (the total number of credit lines currently in the borrower's credit file), followed by “loan_amount” and “delinq_2yrs”(the number of 30+ days past-due incidences of delinquency in the borrower's credit file for the past 2 years). In contrast, variables regarding the purpose of the loan, its duration, application type and the verification of the income seem to be irrelevant.

9. CONCLUSION

Because of the most recent global financial crisis as well as the guidelines of the Basel accords, credit risk has a key topic in the field of financial risk management.

Financial institutions have to make informed decisions on whether or not to grant credit to individuals who submit an application. Consequently, credit scoring has gained serious attention over the past decades as financial institutions are seeking better strategies to deal with this task. Until now, many credit scoring models have been developed based on traditional statistical techniques or AI techniques. The benchmark model in the industry is logistic regression, which by design leads to conclusions that are easy to understand and hence interpretable by both regulators and credit risk managers.

The purpose of this study is to explore the performance of credit scoring using logistic regression as the benchmark model and compare it to the performance of ensemble methods, namely, AdaBoost, Gradient Boosting, Random Forest and Stacking. The process starts with an exploratory analysis and data processing, and only then it was possible to build and train the models. After the models were trained they were evaluated based on four metrics, accuracy, AUC , type I and type II errors.

Taking into consideration the experimental findings, it is possible to make the following conclusions :

- None of the models obtained the best performance on all metrics;
- Considering a trade off between the four metrics AdaBoost provided the best performance, followed by random forest;
- Ensemble methods outperformed LR on all performance metrics except for the AUC. However it is possible that logistic regression achieves a very high AUC at the cost of a high type II error (or a high number of false negatives), which could mean that the classifier is slightly biased.
- Stacking presented the lowest type II error which is considered to be more concerning than type I error, however it provided the poorest performance on the remaining metrics, and it is also much more computational expensive and time consuming when compared to the remaining classifiers.
- Most of the time, those applicants who apply for a lower loan amount, didn't have incidences of past delinquency , and have higher income are more likely to get approved. Other characteristics like purpose of the loan, home ownership, or application type seem not to be taken into consideration.

Regarding the research question: Can ensemble methods perform better than logistic regression, which is the baseline in the industry, at predicting customer loan default?

It was possible to perceive that that well trained machine learning models like AdaBoost and Random Forest were able to outperform logistic regression in several key metrics (for instance, accuracy, type I error , and type II error). Hence, it is possible to conclude that a machine learning based credit model can yield better results than the industry standard.

Despite this fact non of the models presented an outstanding performance when compared to logistic regression, and are also less easy to interpret. Furthermore, it is unlikely that AdaBoost would replace Logistic Regression leading to a change in paradigm.

10. LIMITATIONS AND RECOMMENDATIONS FOR FUTURE WORKS

In the present thesis the main limitation faced was related to computational needs. Due to the size of the data set many of the processes were really time consuming and computationally expensive. In particular, fine tuning the parameters, KNN imputer to deal with missing values, feature selection and modelling the stacking ensemble presented the most challenges.

The gathering of the data set also proposed a challenge, as most of the information is considered sensible/confidencial so its access is very limited, if not impossible to people outside of the financial institution. For this reason the data set had to be chosen from a very limited number of public available data sets.

Hyper-parameter tuning is an important procedure during the construction of the classifiers. However, in this study only one or two hyperparameters per algorithm were tested, due to the computational limitations. Therefore, tuning a higher number of hyperparameters will be introduced in the future research.

The models were only trained and tested against the standard default threshold which is 0.5 (probabilities above or equal to 0.5 mean default, values below are considered non-default). Future work could involve threshold moving in order to try to reduce the false negative rate (type II error).

Future research could also address the exploration of new data sources, for instance, in a rapidly changing social environment, the emergence of the big data era, the digital information recorded on social networks and mobile applications can also be used for consumer credit risk related research, particularly, from a behavioral perspective.

The incorporation of economic effects would also constitute an interesting future research. Even if personal characteristics are the most important factors in determining if a borrower is capable of completing future payment, the external economic environment has an impact on performance as well. It is easier to pay the monthly debt obligations if jobs are stable and salaries are rising. The reverse is true in an economic recession. Even the most trustworthy customers may have trouble repaying the loan if they have the misfortune of becoming unemployed. As a result, the integration of features containing detailed economic information, for instance unemployment rate or house prices, in the credit scoring modeling process could possibly improve accuracy and interpretation.

The last future task is regarding model explainability. According to the world bank the decisions made on the basis of credit scoring should be explainable, transparent, and operate within equal opportunity laws. The variable-importance scores of each predictor presented in the results section provide some limited insight. So it would be important to explore this in depth. Python framework provides, for example, methods like LIME which aim at explaining and interpreting the decisions of a predictive model. Another example is fuzzy logic which has also been explored to improve the explanatory capability of neural networks.

11. REFERENCES

Abdou, H.A., & Pointon, J. (2011). CREDIT SCORING, STATISTICAL TECHNIQUES AND EVALUATION CRITERIA: A REVIEW OF THE LITERATURE. *Intelligent systems in Accounting, Finance and Management, 18*, 59-88. Retrieved from https://doi.org/10.1002

Ackermann, J. (2008). The subprime crisis and its consequences. *Journal of Financial Stability, 4 (04*), 329-337. Retrieved from https://doi.org/10.1016/j.jfs.2008.09.002

Acuna, E., & Rodriguez, C. (2004). The treatement of missing values and its effect on classifier accuracy. *Classification, clustering, and data mining applications,* 639-647. Retrieved from: https://doi.org/10.1007/978-3-642-17103-1_60

Akindaini, B. (2017). MACHINE LEARNING APPLICATIONS IN MORTGAGE DEFAULT PREDICTION. Master's thesis, University of Tampere. Retrieved from https://trepo.tuni.fi/handle/10024/102533

Ala'raj, M., & Abbod, M. (2016). Classifiers consensus system approach for credit scoring. *Knowledge-Based Systems, 104*, 89-105. Retrieved from https://doi.org/10.1016/j.knosys.2016.04.013

Ali, P. J. M., & Faraj, R. H.(2014).Data Normalization and Standardization: A Technical Report. *Machine Learning Technical Reports, 1(1)*, 1-6. Retrieved from: http://doi.org/10.13140/RG.2.2.28948.04489

Altman, E.I.(1968). Financial Ratios, Discriminant Analysis and the Prediction of Corporate Bankruptcy. *Journal of Finance, 23*, 589–609. Retrieved from https://doi.org/10.1111/j.1540-6261

Azevedo, Ana., & Santos, M. (2008). KDD, semma and CRISP-DM: A parallel overview. *IADIS European Conference on Data Mining*, 182-185. Retrieved from https://recipp.ipp.pt/bitstream/10400.22/136/3/KDD-CRISP-SEMMA.pdf

Banco de Portugal. (2007). MAR. Retrieved from https://www.bportugal.pt/sites/default/files/anexos/documentos-relacionados/consulta_bp_2_07_mar.pdf

Banco de Portugal. (2022). O que são e tipos de credito. Retrieved from https://clientebancario.bportugal.pt/pt-pt/o-que-sao-e-tipos-de-credito

Bayraci, J. S., & Susuz, O. (2019). A Deep Neural Network (DNN) based classification model in application to loan default prediction. *Theoretical and Applied Economics, XXVI,* 75-84*.* Retrieved from http://store.ectap.ro/articole/1421.pdf

Bergstra, J., & Bengio, Y. (2012). Random search for hyper-parameter optimization. *Journal of machine learning research*, *13*(2). Retrieved from: https://www.jmlr.org/papers/volume13/bergstra12a/bergstra12a.pdf

Bessis, J.(2002). *Risk Management in Banking*. Retrieved from http://insurance institute.ru/library/zothers/bessis.pdf

BIS.(1999 November 30). PRINCIPLES FOR THE MANAGEMENT OF CREDIT RISK. Retrieved from https://www.bis.org/publ/bcbs54.pdf

BIS.(2018 September 25). Counterparty credit risk in Basel III – Executive Summary. Retrieved from https://www.bis.org/fsi/fsisummaries/ccr_in_b3.htm

Blum, A., Kalai, A.T., & Langford, J. (1999). Beating the hold-out: bounds for K-fold and progressive cross-validation. *COLT '99*, 203-205. Retrieved fom https://doi.org/10.1145/307400.307439

Breeden, L.(2020). Survey of Machine Learning in Credit Risk.Retrieved from http://doi.org/10.13140/RG.2.2.14520.37121

Breiman, L. (2001).Random Forests. *Machine Learning 45*, 5–32. Retrieved from: https://doi.org/10.1023/A:1010933404324

Cao, R. Vilar, J., Devia R., & Andres, A. (2009). Modelling consumer credit risk via survival analysis. *Statistics and Operations Research Transactions, 33*(1), 3-30. Retrieved from https://www.researchgate.net/publication/28314957_Modelling_consumer_credit_risk_via_survival_analysis

Chen, M., & Huang, S. (2003).Credit scoring and rejected instances reassigning through evolutionary computation techniques.*Expert Systems with Applications, 24(4),*433-441. Retrieved from https://doi.org/10.1016/S0957-4174(02)00191-4

Chuang, C., & Lin, R. (2009). Constructing a reassigning credit scoring model. *Expert Systems with Applications, 36(2),* 1685-1694. Retrieved from https://doi.org/10.1016/j.eswa.2007.11.067

Crook, N. J., Edelman, B. D., & Thomas, L.C. (2007). Recent developments in consumer credit risk assessment. *European Journal of Operational Research, 183(3*), 1447-1465. Retrieved from https://doi.org/10.1016/j.ejor.2006.09.100

Dirick, L., Claeskens, G., & Baesens, B. (2016). Time to default in credit scoring using survival analysis: A benchmark study. *Journal of the Operational Research Society,68,* 653-665. Retrieved from https://doi.org/10.1057/s41274-016-0128-9

Dumitrescu, E., Hué, S., Hurlin, C.,& Tokpavi, S. (2022). Machine Learning for Credit Scoring: Improving Logistic Regression with Non Linear Decision Tree Effects. *European Journal of Operational Research, 297*(3), 1178-1192. Retrieved from https://doi.org/10.1016/j.ejor.2021.06.053

Ereiz, Z. (2019). Predicting Default Loans Using Machine Learning (OptiML). *27th Telecommunications Forum (TELFOR)*,1-4. Retrieved from https://doi.org/10.1109/TELFOR48224.2019.8971110

Fitzpatrick,T., & Mues, C. (2016). An empirical comparison of classification algorithms for mortgage default prediction: Evidence from a distressed mortgage market. *European Journal of Operational Research, 249(2)*, 427-439. Retrieved from https://doi.org/10.1016/j.eswa.2012.04.050

Gestel, G. V. T., & Baesens, B. (2009). *Credit Risk Management*. Retrieved from http://196.190.117.157:8080/xmlui/bitstream/handle/123456789/30418/143.Bart%20Baesens,Tony%20van%20Gestel.pdf?sequence=1

Grastrom, D., & Abrahamsson, J. (2019). *Loan Default Prediction using Supervised Machine Learning Algorithms*.Master's work project, KTH Royal Institute of Technology, Sweden. Retrieved from https://www.divaportal.org/smash/get/diva2:1319711/FULLTEXT02.pdf

Gönen,G., Gönen, M., & Gürgen, F.(2012). Probabilistic and discriminative group-wise feature selection methods for credit risk analysis. *Expert Systems with Applications, 39(14)*, 11709-11717. Retrieved from https://doi.org/10.1016/j.eswa.2012.04.050

Hamid, A., & Ahmed, T. (2016). Developing Prediction Model of Loan Risk in Banks Using Data Mining. *Machine Learning and Applications: An International Journal, 3(1),* 1-9. Retrieved from https://doi.org/10.5121/mlaij.2016.3101

Hu, F., & Li, H.(2013). A Novel Boundary Oversampling Algorithm Based on Neighborhood Rough Set Model: NRSBoundary-SMOTE. *Mathematical Problems in Engineering*, 2013. Retrieved from: http://doi.org/10.1155/2013/694809

Model: NRSBoundary-SMOTE. *Mathematical Problems in Engineering*, 2013. Retrieved from: http://doi.org/10.1155/2013/694809

Huang, C., Chen, M., & Wang, C. (2007). Credit scoring with a data mining approach based on support vector machines. *Expert Systems with Applications, 33(4),* 847-856. Retrieve from https://doi.org/10.1016/j.eswa.2006.07.007

Kruppa, J., Schwarz, A., Arminger, G., & Ziegler, A. (2013). Consumer credit risk: Individual probability estimates using machine learning. *Expert Systems with Applications, 40 (13),* 5125-5131. Retrieved from https://doi.org/10.1016/j.eswa.2013.03.019

Lee, T., Chiu, C., Lu, C., & Chen, I. (2002). Credit scoring using the hybrid neural discriminant technique. *Expert Systems with Applications, 23(3)*, 245-254. Retrieved from http://doi.org/10.1016/S0957-4174(02)00044-1

Lee, T.-S., Chiu, C.-C., Chou, Y.C., & Lu, C.-J. (2006). Mining the customer using classification and

regression tree and multivariate adaptative regression splines. *Computational Statistics & Data Analysis 50,* 1113-1130. Retrieved from https://cursa.ihmc.us/rid=1MYWPQZ77-25QC337-30R3/Lee_CART_MARS.pdf

Lessmann, S., Baesens, B., Seow, H. V., & Thomas, L. C. (2015). Benchmarking state-of-the-art classification algorithms for credit scoring: An update of research. *European Journal of Operational Research*, *247*(1), 124–136. Retrieved from https://doi.org/10.1016/j.ejor.2015.05.030

Li, X-L., & Zhong, Yu. (2012). An Overview of Personal Credit Scoring: Techniques and Future Work. *International Journal of Intelligence Science, 02 (04),* 181-189. Retrieved from https://doi.org/10.4236/ijis.2012.224024

Liashchynskyi, P., & Liashchynskyi, P. (2019). Grid Search, Random Search, Genetic Algorithm: A Big Comparison for NAS. Retrieved from: https://doi.org/10.48550/arXiv.1912.06059

Liaw, A., & Wiener, M. (2002). Classification and regression by randomForest. *R news, 2(3),* 18-22. Retrieved from: http://doi.org/10.1021/ci0304160g

Lynn, T., Mooney, G.J., Rosati, P., & Cummins, M. (2020). *Palgrave Studies in Digital Business & Enabling Technologies*. Retrieved from https://link.springer.com/book/10.1007/978-3-030-546601

Maheswari, P., & Narayana, V. C. (2020). Predictions of Loan Defaulter - A Data Science Perspective. *5th International Conference on Computing, Communication and Security (ICCCS)*, 1-4. Retrieved from https://doi.org/10.1109/ICCCS49678.2020.9277458

Malhotra, R., & Malhotra, K. D.(2003). Evaluating consumer loans using neural networks. *Omega, 31(2),* 83-96. Retrieved from https://doi.org/10.1016/S0305-0483(03)00016-1

Marqués, A.I., García, V., & Sánchez, J.S. (2012).Exploring the behaviour of base classifiers in credit scoring ensembles. *Expert Systems with Applications, 39 (11),* Retrieved from https://doi.org/10.1016/j.eswa.2012.02.092

Monett, D., Lewis, P. W. C., & Thórisson, R. K. (2020). John McCarthy's Definition of Intelligence. *Journal of Artificial General Intelligence, 11(2)* ,66-67. Retrieved from http://www.incompleteideas.net/papers/Sutton-JAGI-2020.pdf

Moro, A. R. (2006). *Estimating Probabilities of Default With Support Vector Machines.*Master's thesis, Humboldt University of Berlim. Retrieved from https://edoc.hu-berlin.de/bitstream/handle/18452/14718/moro.pdf?sequence=1

Mungasi, S., & Odhiambo, C.(2019). Comparison of Survival Analysis Approaches to Modelling Credit Risks. *American Journal of Theoretical and Applied Statistics*, *8(2),* 39-46. Retrieved from:http://doi.org/10.11648/j.ajtas.20190802.11

Ngai, T.W.E., Hu, Y., Wong, Y.H., Chen, Y., & Sun, X.(2011).The application of data mining

techniques in financial fraud detection: A classification framework and an academic review of literature. *Decision Support Systems, 50(3),* 559-569. Retrieved from https://doi.org/10.1016/j.dss.2010.08.006

Nigro, P., & Glennon, D. (2005). Measuring the Default Risk of Small Business Loans: A Survival Analysis Approach. *Journal of Money, Credit and Banking,37*, 923-47. Retrieved from http://doi.org/10.1353/mcb.2005.0051

Ping, Y., & Yongheng, L. (2011) Neighborhood rough set and SVM based hybrid credit scoring classifier. *Expert Systems with Applications, 38(9*), 11300-11304. Retrieved from https://doi.org/10.1016/j.eswa.2011.02.179

Plumed, M.T., Orchando, C. L., Ferri, C., Orallo, H. J., Laniche N. K., Quintana, R.J.M., & Flack, P. (2019). CRISP-DM Twenty Years Later: From Data Mining Processes to Data Science Trajectories. *IEEE Transactions on Knowledge and Data Engineering.* Retrieved from https://ec.europa.eu/jrc/communities/sites/default/files/publ046_tkde_2020_paper_earlyaccess.pdf

Pothumsetty, R. (2020). Implementation of Artifical Intelligence and Machine learning in Financial services. *International Research Journal of Engineering and Technology, 7(03),* 3186-3193. Retrieved from https://www.irjet.net/archives/V7/i3/IRJET-V7I3639.pdf

Schober, P., Boer, C., & Schwarte L. A.(2018). Correlation Coefficients: Appropriate Use and Interpretation. Anesthesia & Analgesia, 126(5), 1763-1768. Retrieved from: 10.1213/ANE.0000000000002864

Schröer,C., Kruse, F., & Gómez, M. J.(2021).A Systematic Literature Review on Applying CRISP-DM Process Model. *Procedia Computer Science, 181*, 526-534. Retrieved from https://doi.org/10.1016/j.procs.2021.01.199

Schapire, R.E. (2013). Explaining AdaBoost. *Empirical Inference*, 37-52. Retrieved from: https://doi.org/10.1016/j.procs.2021.01.199

Sheikh, A.M., Goel, K. A., & Kumar, T. (2020). An Approach for Prediction of Loan Approval using Machine Learning Algorithm. *2020 International Conference on Electronics and Sustainable Communication Systems (ICESC)*, 490-494. Retrieved from https://doi.org/10.1109/ICESC48915.2020.9155614

Singh, Dalwinder., & Singh, Birmohan. (2020). Investigating the impact of data normalization on classification performance. *Applied Soft Computing*. *97*, 105524. Retrieved from http://doi.org/10.1016/j.asoc.2019.105524

Soares, I., Moreira, J., Pinho, C., & Couto, J. (2015). *Decisões de Investimento análise* financeira de *projetos*. Retrieved from https://static.fnac-static.com/multimedia/PT/pdf/9789726188063.pdf

Stepanova, M. & Thomas, L. (2002). Survival Analysis Methods for Personal Loan Data. *Operations Research, 50(2*), 277-289. Retrieved from http://doi.org/10.1287/opre.50.2.277.426

Thomas, C. L., Edelman, B. D., & Crook, N. J. (2002). *Credit Scoring and its Applications.* Retrieved from https://books.google.pt/books?id=GMWcWuBDJZUC&printsec=frontcover&hl=pt-PT&source=gbs_ge_summary_r&cad=0#v=onepage&q&f=false

Tsai, C., & Wu, J.(2008). Using neural network ensembles for bankruptcy prediction and credit scoring. *Expert Systems with Applications, 34(4)*, 2639-2649. Retrieved from https://doi.org/10.1016/j.eswa.2007.05.019

Visalakshi, S., & Radha, V. (2015). A literature review of feature selection techniques and applications: Review of feature selection in data mining. *2014 IEEE International Conference on Computational Intelligence and Computing Research*, 1-6. Retrieved from: https://doi.org/10.1109/ICCIC.2014.7238499

Wang, G., Hao, J., Ma, J., & Jiang, H. (2011).A comparative assessment of ensemble learning for credit scoring. *Expert Systems with Applications, 38(1)*, 223-230. Retrieved from https://doi.org/10.1016/j.eswa.2010.06.048

West, D., Dellana, S., & Qian, J. (2005).Neural network ensemble strategies for financial decision applications. *Computers & Operations Research, 32(10),* 2543-2559. Retrieved from https://doi.org/10.1016/j.cor.2004.03.017

West, D. (2000). Neural network credit scoring models. *Computers & Operations Research, 27 (11-12)*, 1131-1152. Retrieved from https://doi.org/10.1016/S0305-0548(99)00149-5

World Bank Group.(2019). CREDIT SCORING APPROACHES GUIDELINES. Retrieved from https://thedocs.worldbank.org/en/doc/935891585869698451-0130022020/original/CREDITSCORINGAPPROACHESGUIDELINESFINALWEB.pdf

Yadav, S., & Shukla, S.(2016). Analysis of k-Fold Cross-Validation over Hold-Out Validation on Colossal Datasets for Quality Classification. *IEEE 6th International Conference on Advanced Computing (IACC),* 78-83. Retrieved from http://doi.org/10.1109/IACC.2016.25

Ying., X.(2019). An Overview of Overfitting and its Solutions. *Journal of Physics: Conference Series*, *1168(2)*, 022022. Retrieved from: http://doi.org/10.48550/arXiv.1912.06059

Zhou, Z.-H. (2012). *Ensemble Methods: Foundations and Algorithms*. Retrieved from https://doi.org/doi:10.1201/b12207-2

12. APPENDIX

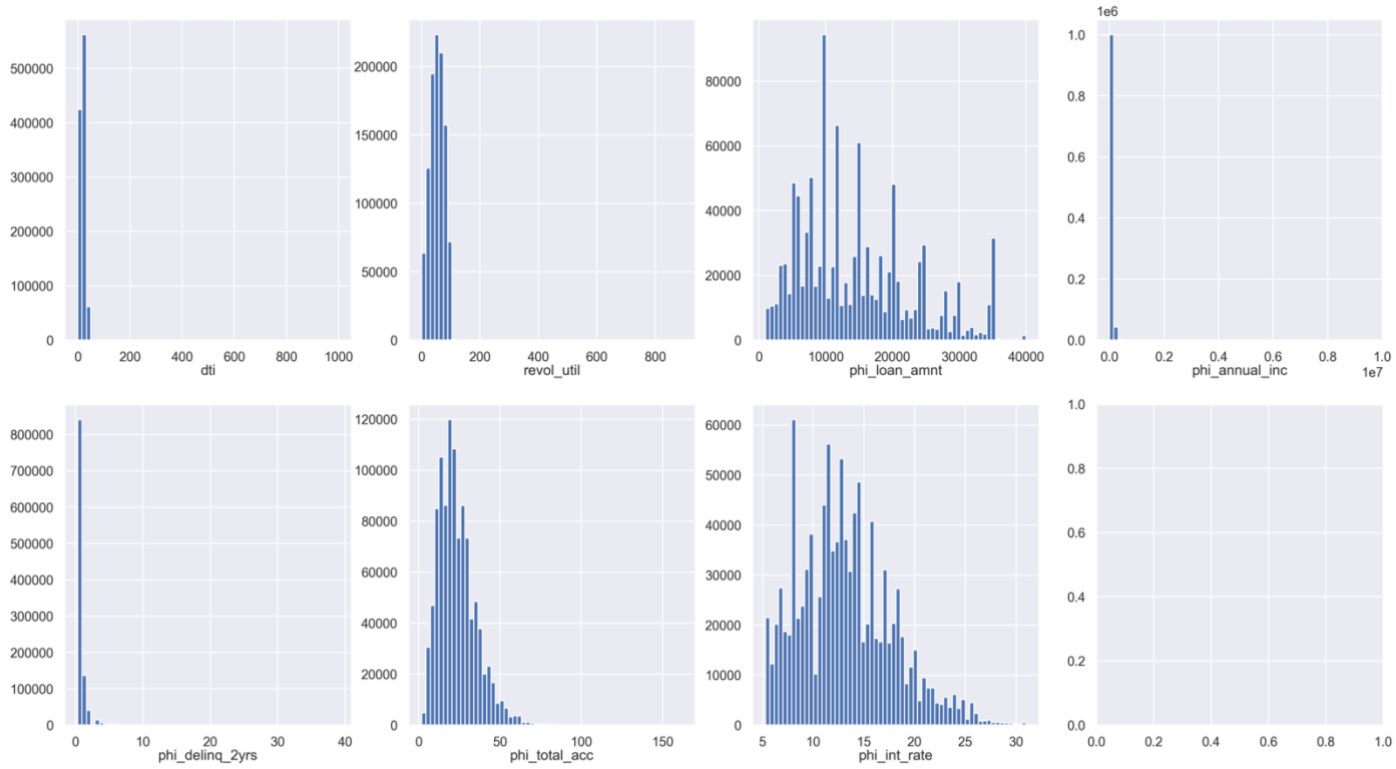


Figure A1: Numeric Variable's histograms

Source: Authors preparation

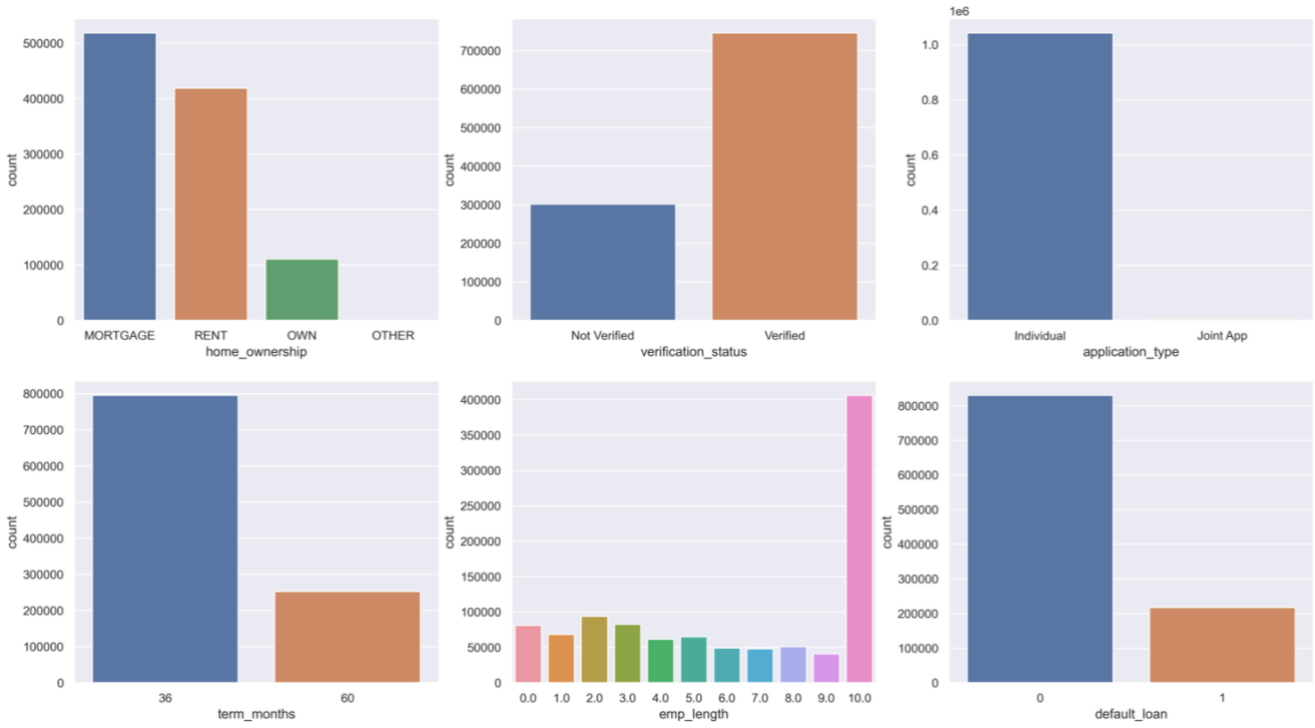


Figure A2: Countplot of categorical and discrete metric features

Source: Authors preparation

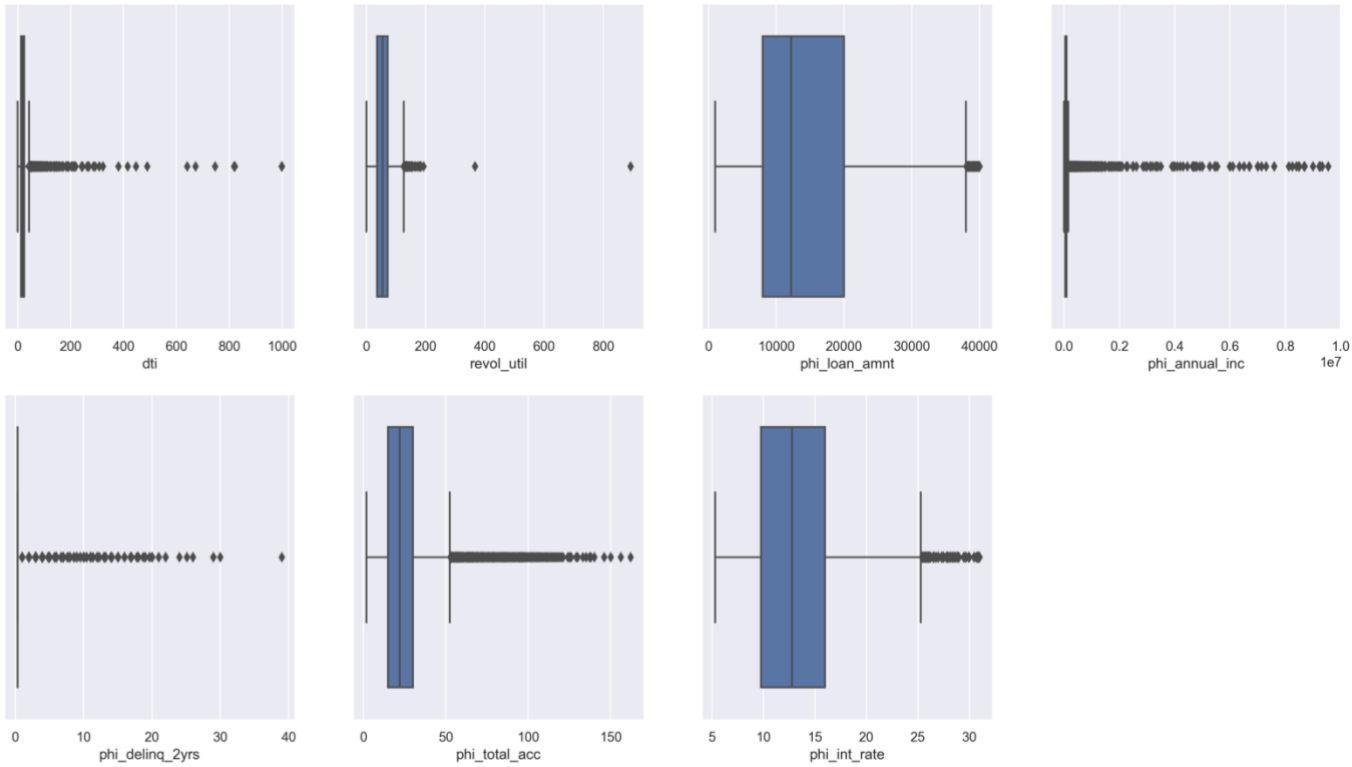


Figure A3: Numeric variable's Box Plots

Source: Authors preparation

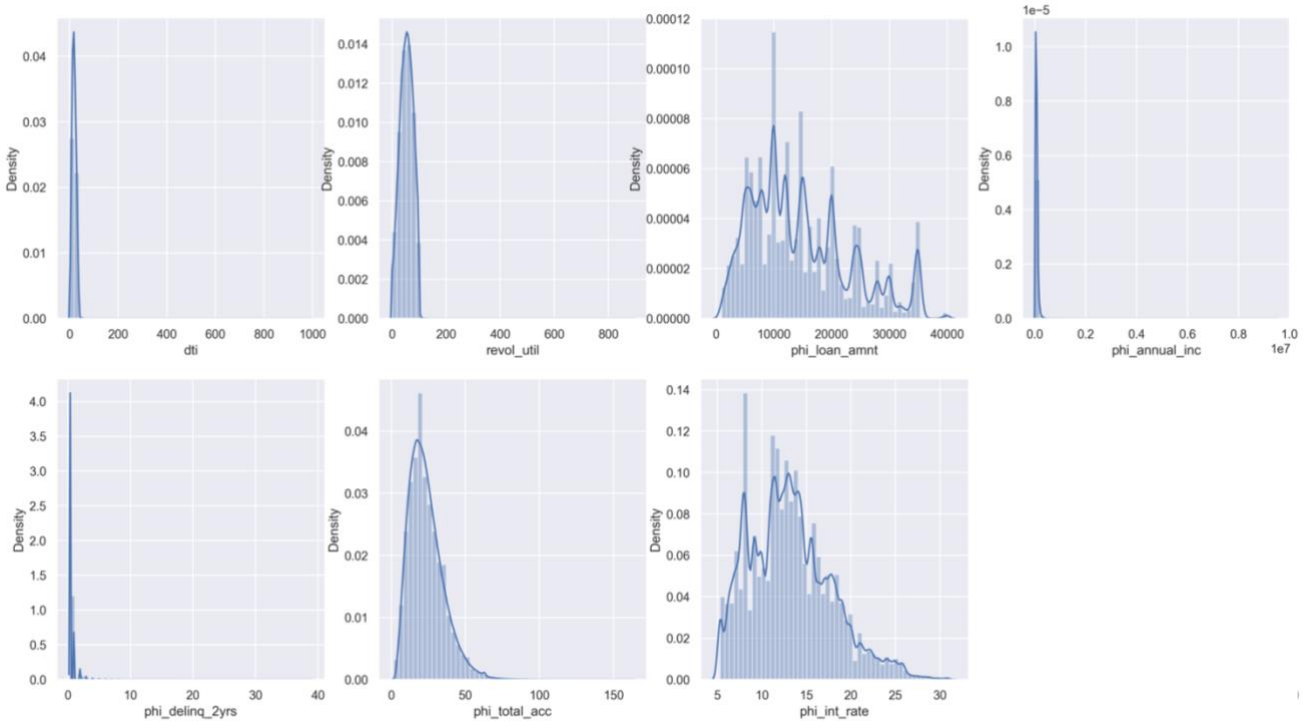


Figure A4: Numeric variable's Distribution Plots

Source: Authors preparation

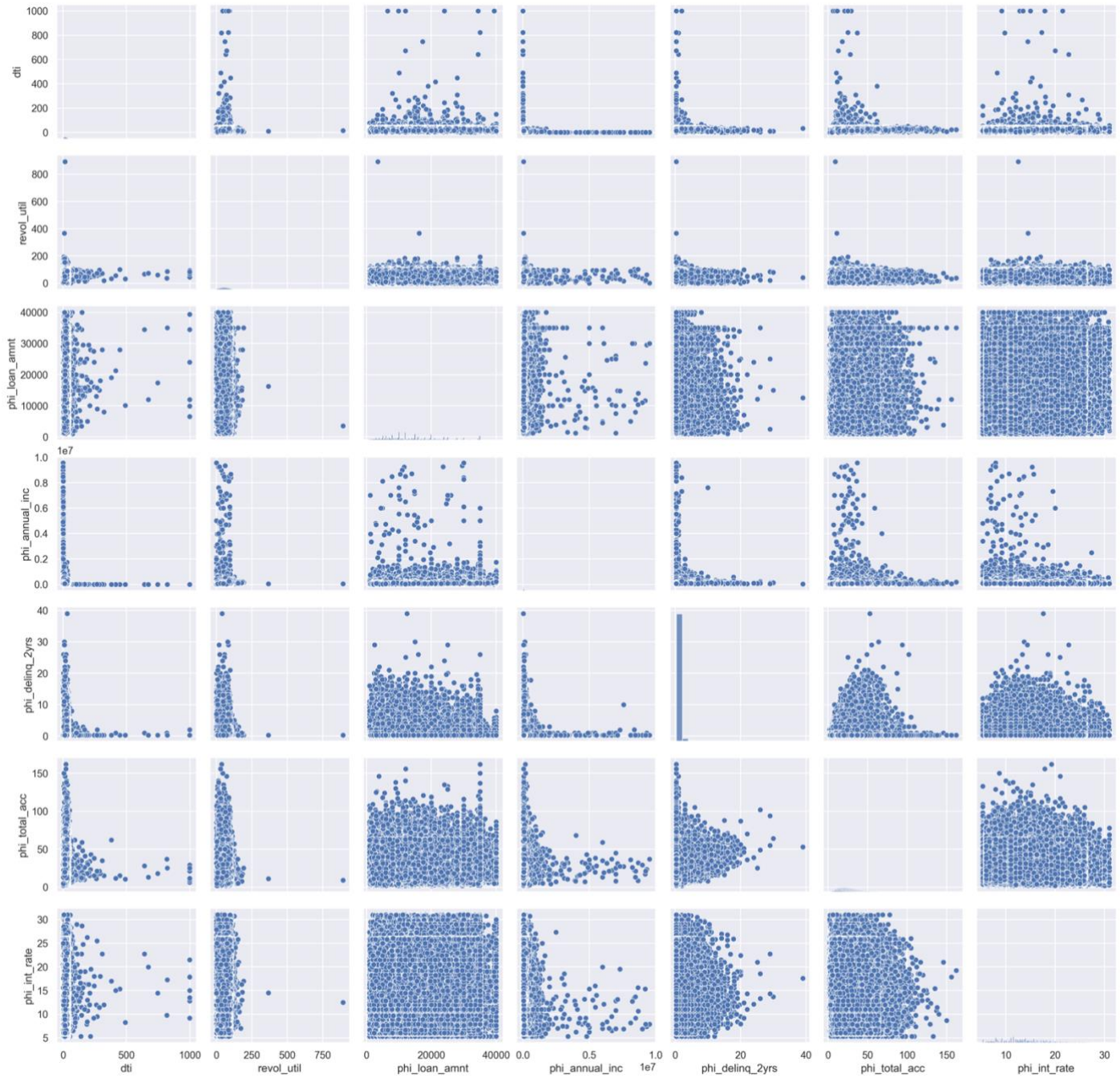


Figure A5: Pairwise Relationship of Numerical Variables

Source: Authors preparation

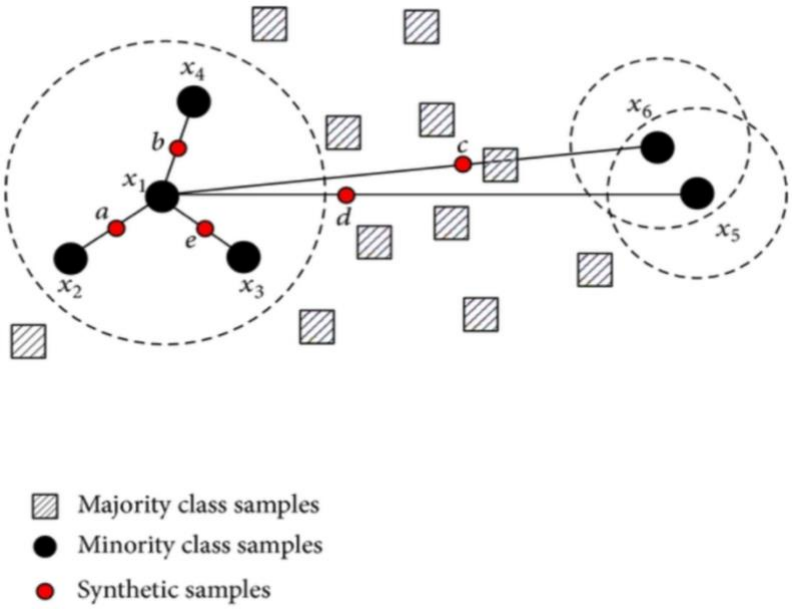


Figure A6: Schematic of SMOTE

Source: Hu, F., & Li, H. (2013)

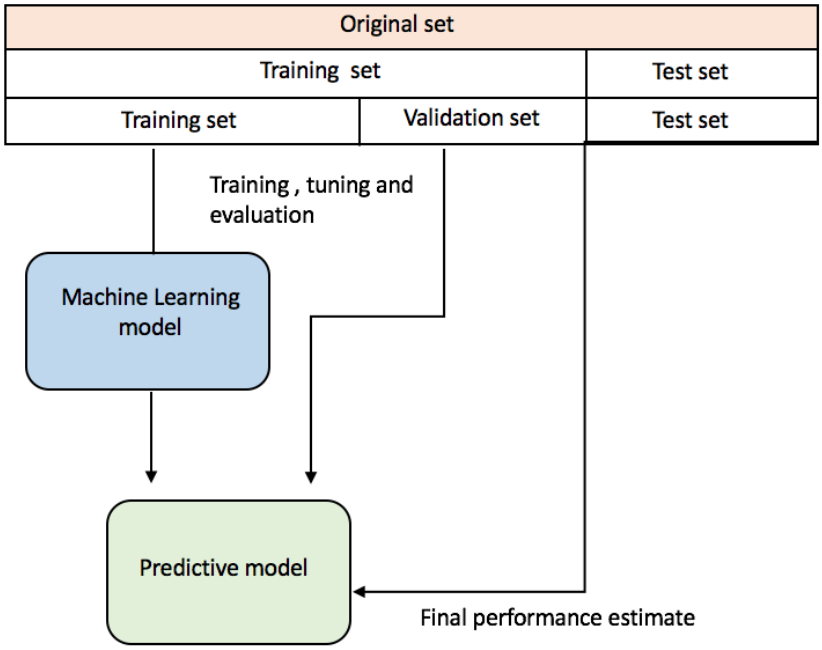


Figure A7: Illustration of the Hold-out method

Source: Authors preparation

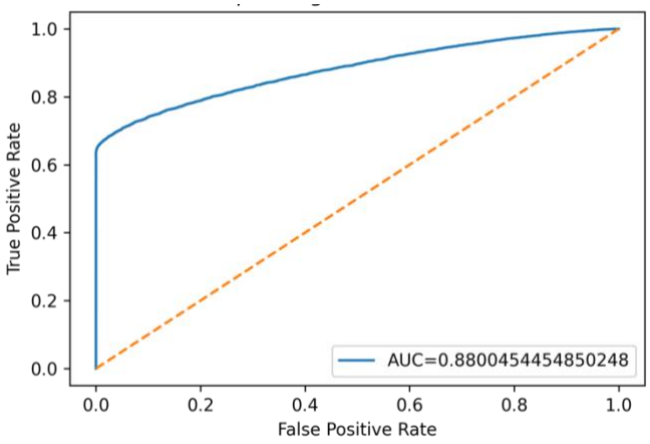


Figure A8: Receiver Operanting Characteristic- AdaBoost

Source: Authors preparation

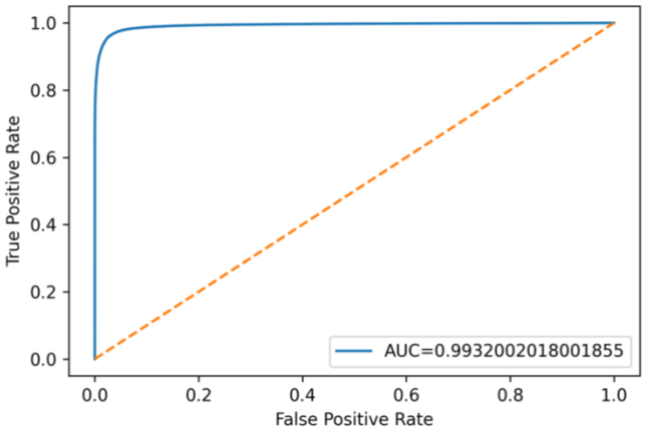


Figure A9: Receiver Operanting Characteristic- Logistic Regression

Source: Authors preparation

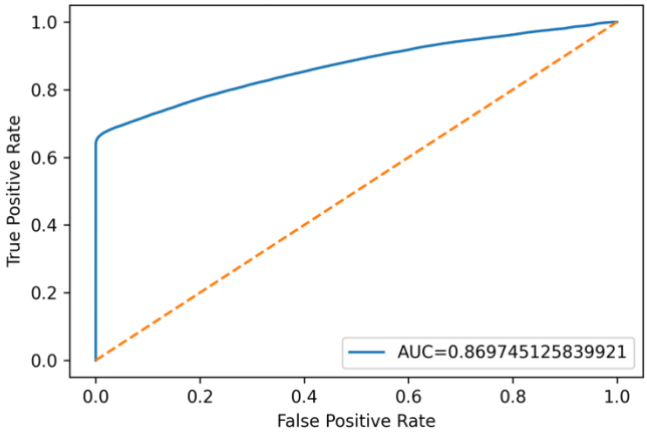


Figure A10: Receiver Operanting Characteristic- Gradient Boosting

Source: Authors preparation

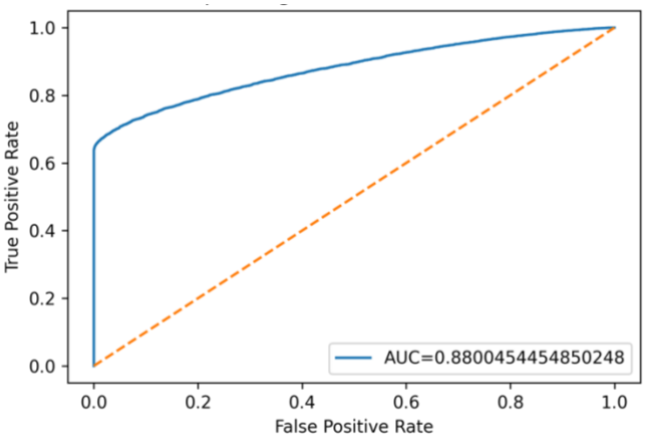


Figure A11: Receiver Operanting Characteristic- Random Forest

Source: Authors preparation

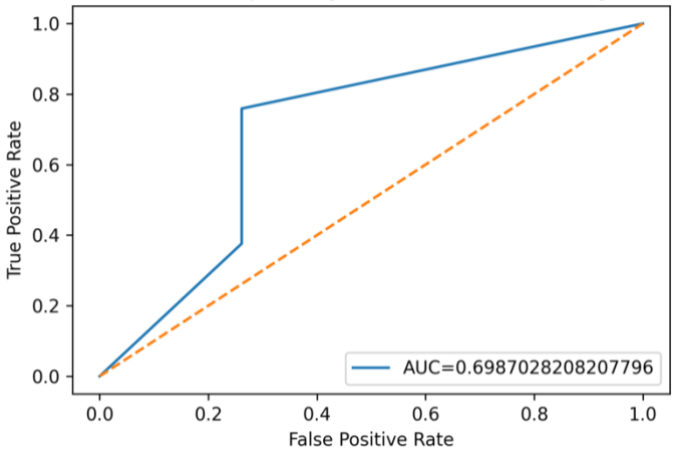


Figure A12: Receiver Operanting Characteristic- Stacking

Source: Authors preparation

B TABLES

	count	mean	std	min	25%	50%	75%	max
term_months	1048575.0	41.778471	10.261222	36.000000	36.000000	36.000000	36.000000	60.00
dti	1048536.0	18.377943	9.068765	-1.000000	12.040000	17.840000	24.270000	999.00
revol_util	1048012.0	53.467644	23.925333	0.000000	35.700000	54.000000	71.900000	892.30
default_loan	1048575.0	0.207834	0.405758	0.000000	0.000000	0.000000	0.000000	1.00
phi_loan_amnt	1048575.0	14491.850199	8502.968297	989.174936	8000.000000	12200.000000	20000.000000	40000.00
phi_annual_inc	1048575.0	75598.085442	68801.040049	1.000000	45203.802701	64819.563099	90000.000000	9550000.00
phi_delinq_2yrs	1048575.0	0.582482	0.794285	0.319032	0.319032	0.319032	0.319032	39.00
phi_total_acc	1048575.0	23.612509	11.572079	1.848795	15.000000	22.000000	30.000000	162.00
phi_int_rate	1048575.0	13.186467	4.605620	5.319019	9.747550	12.790000	15.985108	30.99

Table B1: Descriptive statistics of metric features

Source: Authors preparation

TRAIN

	precision	recall	f1-score	support
0.0	1.00	1.00	1.00	642079
1.0	1.00	1.00	1.00	161745
accuracy			1.00	803824
macro avg	1.00	1.00	1.00	803824
weighted avg	1.00	1.00	1.00	803824

```
[[641683    396]
 [     0 161745]]
```

VALIDATION

	precision	recall	f1-score	support
0.0	0.99	0.99	0.99	160520
1.0	0.98	0.98	0.98	40437
accuracy			0.99	200957
macro avg	0.99	0.99	0.99	200957
weighted avg	0.99	0.99	0.99	200957

```
[[159669    851]
 [   835  39602]]
```

Table B2:Stacking classification report

Source: Authors preparation

TRAIN

	precision	recall	f1-score	support
0.0	1.00	1.00	1.00	642079
1.0	1.00	1.00	1.00	642079
accuracy			1.00	1284158
macro avg	1.00	1.00	1.00	1284158
weighted avg	1.00	1.00	1.00	1284158

```
[[642079      0]
 [     0 642079]]
```

VALIDATION

	precision	recall	f1-score	support
0.0	0.97	1.00	0.99	160520
1.0	1.00	0.97	0.98	160520
accuracy			0.98	321040
macro avg	0.99	0.98	0.98	321040
weighted avg	0.99	0.98	0.98	321040

```
[[160053    467]
 [  4374 156146]]
```

Table B3: Random Forest classification report

Source: Authors preparation

TRAIN

	precision	recall	f1-score	support
0.0	0.94	0.98	0.96	642079
1.0	0.97	0.94	0.96	642079
accuracy			0.96	1284158
macro avg	0.96	0.96	0.96	1284158
weighted avg	0.96	0.96	0.96	1284158

```
[[626075  16004]
 [ 38301 603778]]
```

VALIDATION

	precision	recall	f1-score	support
0.0	0.94	0.97	0.96	160520
1.0	0.97	0.94	0.96	160520
accuracy			0.96	321040
macro avg	0.96	0.96	0.96	321040
weighted avg	0.96	0.96	0.96	321040

```
[[156471   4049]
 [  9522 150998]]
```

Table B4: Logistic Regression classification report

Source: Authors preparation

TRAIN

	precision	recall	f1-score	support
0.0	0.99	1.00	0.99	637302
1.0	1.00	0.99	0.99	637302
accuracy			0.99	1274604
macro avg	0.99	0.99	0.99	1274604
weighted avg	0.99	0.99	0.99	1274604

```
[[637248     54]
 [  8395 628907]]
```

VALIDATION

	precision	recall	f1-score	support
0.0	0.99	1.00	0.99	159326
1.0	1.00	0.99	0.99	159326
accuracy			0.99	318652
macro avg	0.99	0.99	0.99	318652
weighted avg	0.99	0.99	0.99	318652

```
[[159307     19]
 [  2240 157086]]
```

Table B5:GradientBoosting classification report

Source: Authors preparation

TRAIN

	precision	recall	f1-score	support
0.0	0.96	0.99	0.98	637302
1.0	0.99	0.96	0.98	637302
accuracy			0.98	1274604
macro avg	0.98	0.98	0.98	1274604
weighted avg	0.98	0.98	0.98	1274604

```
[[632310   4992]
 [ 26050 611252]]
```

VALIDATION

	precision	recall	f1-score	support
0.0	0.96	0.99	0.98	159326
1.0	0.99	0.96	0.98	159326
accuracy			0.98	318652
macro avg	0.98	0.98	0.98	318652
weighted avg	0.98	0.98	0.98	318652

```
[[158120   1206]
 [  6593 152733]]
```

Table B6: AdaBoost classification report

Source: Authors preparation

NOVA Information Management School
Instituto Superior de Estatística e Gestão de Informação

Universidade Nova de Lisboa