ANALYSING THE EXPLAINABILITY OF CREDIT SCORING MACHINE LEARNING MODELS USING SHAPLEY ADDITIVE EXPLANATIONS APPROACH

Ramakgahlele Merriam Thoka

A research report submitted in partial fulfillment for the degree of

Masters in eScience

in the

Department of Computer Science

Faculty of Science & Agriculture

University of Limpopo

Supervisor: Prof TI Modipa

May, 2025

DEDICATION

I dedicate this dissertation to my devoted family, whose unfailing encouragement and support have served as my beacon of hope along this academic journey. Their belief in my abilities and the compromises made along the way have been essential to my success. I would also like to extend my heartfelt appreciation to my friends who provided a stimulating and collaborative environment, fostering intellectual discussions and shared learning experiences. I am indebted to the research participants who generously volunteered their time and shared their knowledge, without whom this study would not have been possible. Lastly, I want to dedicate this work to everyone who aspires to truth, knowledge, and constructive change. May our collective efforts contribute to a better understanding and advancement in our respective fields. This dissertation is a testament to the unwavering support, love, and inspiration from all those mentioned and the countless others who have left an indelible mark on my academic and personal journey.

DECLARATION

I Ramakgahlele Merriam Thoka declare that **Analysing the Explainability of Credit Scoring Machine Learning Models Using Shapley Additive Explanations Approach** is my own original work and all information extracted from other sources is acknowledged as such by means of complete references. I further affirm that I have not submitted this work to any other university for any other degree or examination.

Signature:  Date: 30/07/2025

Acknowledgements

In order to complete this dissertation, I would like to sincerely thank everyone who helped and encouraged me.

Above all, I want to express my sincere gratitude to Dr. TI Modipa, my supervisor, for his leadership, knowledge, and steadfast support during this study process. His invaluable advice, kind criticism, and support have greatly influenced the development of my work. I extend my sincere appreciation to the members of The Speech Technology Research Group in the Department of Computer Science committee for their valuable input, feedback, and expertise. Their thoughtful suggestions and helpful critiques have substantially improved the caliber of this study.

I am thankful to the participants for giving freely of their time and expertise, enabling me to collect the necessary data for this study. Their contributions have been fundamental to the findings and conclusions presented.

Abstract

In recent years, machine learning models have gained popularity in credit scoring applications due to their ability to handle large volumes of data and capture complex patterns. However, the lack of transparency and interpretability in these models raises concerns regarding their trustworthiness and fairness. This study aims to address this matter by employing the Shapley Additive Explanations (SHAP) approach to analyse the explainability of credit scoring machine learning models. The lending club dataset, a comprehensive collection of loan applications and associated attributes, is utilized for this analysis. The methodology involves training and evaluating various credit scoring models, including Random Forest, XGBoost, and CatBoost, and generating SHAP values to quantify the importance of input features in the prediction process. The results reveal valuable insights into the factors influencing credit scoring decisions and provide a holistic understanding of the models' behaviour. By utilizing SHAP explanations, we gain interpretability and can identify features that significantly impact the credit scoring outcomes. This knowledge can help stakeholders, including lenders and regulators, make informed decisions and improve the transparency and accountability of credit scoring systems. The discoveries of this study advance the expanding field of explainable artificial intelligence(AI) and its application in the domain of credit risk management. By enhancing the explainability of credit scoring models, we aim to increase trust, fairness, and accountability in the lending process, ultimately shaping a more inclusive and responsible financial ecosystem.

Keywords – Explainability, Credit scoring, Machine learning models, Shapley Additive Explanations.

Contents

List of Tables

List of Figures

Chapter1: Introduction

In this chapter, we introduce the research topic of analysing the explainability of credit scoring machine learning models using the SHAP approach. The chapter sets the stage by providing an overview of the problem statement, motivation, aim and objectives, proposed solution, and dissertation structure.

1.1 Problem Statement

Credit scoring is a critical process in the financial sector, used by lenders to assess the creditworthiness of borrowers before granting loans or extending credit [3]. The increasing adoption of complex machine learning models in credit scoring has improved predictive accuracy but has also brought forth challenges related to model explainability. The lack of transparency and interpretability in these models hinders stakeholders' ability to understand how credit decisions are made, leading to concerns regarding potential biases, discrimination, and the need for accountability in the credit assessment process. Traditional credit scoring models, such as logistic regression and decision trees, provide clear explanations for their decisions, making them easier to trust and validate. However, as financial institutions embrace more advanced machine learning methods, including ensemble techniques and deep learning, the interpretability of these models diminishes significantly [6][39]. The lack of explainability in credit scoring machine learning models raises questions such as: What are the key factors influencing credit scoring decisions? How do specific input features contribute to individual credit assessments? Can we identify and address potential biases in these models? How can we ensure transparency and accountability in credit scoring systems without compromising predictive accuracy? To

address these challenges, this study aims to analyse the explainability of credit scoring machine learning models using the SHAP approach [29]. By applying SHAP to popular credit scoring algorithms such as Random Forest, XGBoost, and CatBoost, we seek to gain insights into the factors driving credit decisions and enhance the transparency and interpretability of these models. The analysis of credit scoring machine learning models' explainability using SHAP values will provide valuable insights to lenders, regulators, and borrowers alike. Understanding the factors influencing credit decisions can enable lenders to identify key drivers of credit risk and make more informed lending decisions. For borrowers, it offers transparency into credit assessment, empowering them to understand and potentially improve their credit scores. Ultimately, this research aims to contribute to a fairer and more accountable credit scoring system, fostering trust and inclusivity in the financial ecosystem. By addressing the problem of model explainability, we seek to bridge the gap between model accuracy and interpretability, ensuring that credit scoring decisions are transparent, fair, and free from potential biases or discrimination

1.2 Motivation

The motivation behind conducting this study lies in the growing significance of credit scoring machine learning models in the financial sector and the increasing demand for transparency and accountability in their decision making process [22][12]. As machine learning models become more prevalent in credit assessment, there is a pressing need to ensure that these models are not just accurate but also interpretable, fair, and free of biases. The lack of transparency in complex machine learning models has raised concerns among lenders, regulators, and borrowers about the potential risks associated with opaque decision making [11]. Understanding how credit scoring models arrive at their predictions is critical for stakeholders to validate the model's reliability, identify potential biases, and comply with regulatory requirements. The SHAP approach provides a promising solution to this issue, offering a unified and intuitive framework to explain the factors driving individual predictions. By quantifying the contribution of each input feature, SHAP values offer valuable insights into the model's decision-making process, thereby

increasing model transparency and trust. The motivation to analyse the explainability of credit scoring machine learning models using the SHAP approach is to bridge the gap between model complexity and interpretability [21]. By applying SHAP to popular credit scoring algorithms, we aim to shed light on the critical features influencing credit decisions, empowering lenders to make more informed and responsible lending choices. Moreover, this research seeks to empower borrowers by providing transparent insights into the credit assessment process [20]. Understanding the factors affecting creditworthiness can enable borrowers to take proactive steps to improve their credit scores and financial standing [8]. Ultimately, the motivation behind this study is to contribute to a fairer and more accountable credit scoring system. By addressing the problem of model explainability, we aim to enhance the transparency and fairness of credit scoring decisions, fostering trust among stakeholders, and ensuring inclusive access to credit for individuals and businesses alike [37]. Through this research, we aspire to promote responsible lending practices, mitigate biases, and pave the way for a more equitable financial ecosystem [27].

1.3 Aim and Objectives

The aim of this study is to analyse explanations of credit scoring tree-based machine learning models using Shapley Additive Explanations approach. This goal can be broken down into the following objectives:

- Perform data preparation and feature selection on the Lending Club dataset.
- Train Extreme Gradient boost, Catboost and Random Forest tree-based machine learning models.
- Evaluate the performance of these tree-based machine learning models.
- Provide model explanations of the implemented machine learning models through Shapley Additive Explanations.

1.4 Proposed Solution

Our proposed solution tackles the explainability challenge head-on by leveraging the power of SHAP. We will apply the TreeExplainer variant specifically designed for tree-based models like Random Forest, XGBoost, and CatBoost. For each creditworthiness prediction, SHAP will quantify the marginal contribution of every feature, unravelling their individual and collective influence on the outcome. This granular insight allows us to pinpoint key drivers of credit decisions, identify potential biases within the model, and analyse how different features interact to impact the final score. We will leverage SHAP visualizations to present these explanations in a clear and intuitive manner, enabling lenders to understand the rationale behind each credit assessment. This approach addresses all our research objectives: it involves data preparation and feature selection to ensure model accuracy, trains diverse tree-based models for a comprehensive comparison. It also evaluates performance to validate their effectiveness, and finally uses SHAP explanations to demystify the black box, fostering trust and accountability in credit scoring decisions.

1.5 Dissertation Structure

The format of this report is as follows:

- Chapter 2: offers an in-depth review of existing methodologies in the literature, establishing foundational concepts crucial for subsequent project phases
- Chapter 3: a detailed exposition of the system's workflow is presented, along with elucidation of implementation procedures for tree-based models and the SHAP XAI technique.
- Chapter 4: offers an elaborate account of various conducted experiments, including evaluation of tree-based models and evaluation of the effectiveness of the implemented credit scoring system's interpretability
- Chapter 5: engages in discussions on limitations, explores potential enhancements,

and presents conclusive findings of the study.

Chapter2: Literature Review

2.1 Introduction

The explainability of machine learning models, especially in credit-related applications, has garnered significant attention in recent years. Researchers have explored distinct methods and techniques to improve the interpretability of these models and address the concerns surrounding their black-box nature. In this chapter, we review some of the key works in the field of explainable credit scoring and the use of SHAP in model interpretability.

2.2 Credit Scoring

Credit scoring models serve as critical tools in assessing the creditworthiness of individuals or entities. These models leverage various features, such as credit history, income, and debt, to predict the likelihood of default or credit risk. The overarching goal is to provide lenders with accurate and reliable insights for making informed decisions on loan approvals or credit extensions. The landscape of credit scoring has witnessed advancements in machine learning techniques, leading to the adoption of sophisticated algorithms like Random Forest(RF), Extreme Gradient Boosting (XGBoost), and CatBoost in this study. Credit scoring has its origins in the late 1960s when bankers assessed loan applicants based on subjective judgments related to their character, capital, capacity, and collateral, collectively known as the 4 'Cs [1]. Several years later, there was a shift towards employing univariate accounting-based credit scoring systems. In these systems, borrowers' essential accounting ratios were compared against established standards. Roughly

a decade afterward, financial institutions began utilizing multivariate accounting-based credit scoring systems, which represented an enhanced approach that assigns weight to key accounting variables of borrowers to generate a credit risk score, commonly referred to as the default probability (PD) [15]. The PD score is then evaluated against the vital threshold, influencing the decisions regarding loan approval or denial. Essentially, there exist four primary categories of multivariate accounting-based credit scoring systems, encompassing the linear discriminant analysis (LDA) model, the logit model, the probit model, and the linear probability model [1]. The linear probability model utilizes borrowers' characteristics and historical data, presuming a linear connection between the PD and the variables. In contrast, the logit model employs a designated set of variables for PD prediction, assuming a logistic distribution for the probability. On the other hand, the probit model assumes a normal distribution for the probability. A linear function of variables that successfully distinguishes between two loan application categories (Good and Bad) is found by the discriminant analysis model, which then groups borrowers into these groups. Subsequently, Neural Networks (NN) also entered this field, using supervised learning to discover the connection between borrowers' attributes and PD without imposing specific assumptions [33].

2.3 Machine learning algorithms used in credit scoring

Lately, a range of diverse methods and approaches have been introduced. David West conducted experiments involving five different types of NN to identify the network achieving the greatest accuracy rating [48]. The findings showed that Multilayer Perceptron (MLP) performed the least accurately for credit scoring, while Radial Basis Function (RBF) and Mixture-of-Experts were the most successful NN models. In comparison to MLP, Li et al. statistically showed that the Support Vector Machine (SVM) performed better in classification, mainly because of its superior global optimization and generalization capabilities [25]. In a similar vein, Martens, David, et al constructed a credit scoring model

using SVMs and incorporated rule extraction techniques to enhance SVM interpretability [32]. The results highlighted that rule extraction outperformed sensitivity analysis and inverse classification in providing explanations. Additionally, when combined with methods like Trepan and Genetic Rule Extraction, SVM only experienced a minor decrease in classification performance.

Additionally, decision trees have been used in the field of credit scoring. In a study by Zurada, Jozef, three distinct DT methods were utilized: Entropy reduction, Gini reduction, and chi-squared [49]. This research demonstrated that decision trees perform exceptionally well in credit scoring tasks and offer a high degree of transparency due to their ability to generate straightforward IF-THEN rules. Another approach, presented by Malekipirbazari, Milad, and Vural Aksakalli introduced a classification system for predicting credit risk based on RF [30]. According to their analysis, which was carried out using a subset of the Lending Club (LC) dataset, their RF model achieved an astounding 78% classification accuracy, outperforming SVM, Logistic Regression (LR), k-Nearest Neighbor (k-NN), Fair Isaac Corporation (FICO) scores, and LC grades. Additionally, Florez-Lopez, Raquel, and Juan Manuel Ramon-Jeronimo introduced the CorrelatedAdjusted Decision Forest (CADF), an ensemble technique that combines multiple decision trees [18].

CADF comprises five base DT classifiers to enhance diversity in the results. These results are then combined using a correlation-adjusted weighted voting scheme, and the approach yields a rule-based structure that is easily comprehensible to humans. Marceau, Louis, et al conducted a comparison of various well-known machine learning models, including DNN, SVM, XGBoost, and RF [31]. Their findings indicated that RF and XGBoost algorithms demonstrated superior performance, particularly in handling imbalanced data. Interestingly, they found that when handling extremely unbalanced datasets, deep learning models did not outperform decision tree-based techniques. This aligns with the results reported by Salvaire, Pierre Antony Jean Marie by and Bussmann, Niklas, et al, who also implemented XGBoost and verified its effectiveness, surpassing conventional models of credit scoring such as LR and Scorecards [43][7].

2.4 Shapley Additive Explanations

The use of SHAP values in interpreting credit scoring models has gained significant attention in recent research. Du Toit et al. applied SHAP to an RF model and demonstrated its potential in providing individual feature contributions that explain model decisions [17]. The findings emphasized the importance of interpretability in credit assessment processes, promoting trust and understanding in lending decisions. Building on this work Lundberg Scott, and Su-In Lee introduced the SHAP framework as a unified method for elucidating any machine learning model, showcasing its application in finance and healthcare domains [29]. The study highlighted the effectiveness of SHAP in improving the interpretability of credit scoring models. Ariza-Garzón, Miller Janny, et al proposed an interpretable credit scoring framework by combining SHAP with a rule-based credit scoring algorithm, resulting in transparent and intuitive credit scores [2] . The study highlighted the effectiveness of SHAP in improving the interpretability of credit scoring models. Ribeiro et al. introduced Anchors, an approach for explaining individual predictions, and demonstrated its application in credit scoring, providing interpretable rules for credit decisions [40].

In the pursuit of transparent and interpretable credit scoring models, researchers have extensively explored the application of SHAP values. Misheva, Branka Hadji, et al demonstrated the practical use of SHAP in understanding the decision-making process of a random forest model, highlighting the value of individual feature contributions in credit assessments [34] . Lundberg Scott and Su-In Lee extended the application of SHAP across various domains, including finance, stressing the significance of model transparency to ensure ethical and accountable decisions [29]. With SHAP's growing prominence, the research community recognizes its potential in enhancing credit scoring models, fostering trust, and offering meaningful insights to both lenders and borrowers. These studies collectively underscore the importance of interpretability in credit assessment processes and advocate for the adoption of SHAP as a powerful tool for understanding and improving credit scoring models. Future research in this area will undoubtedly contribute to ad-

vancing transparency and fairness in credit evaluation, aligning the industry with ethical and accountable practices.

2.5 Tree-based Credit Scoring Models

In this section, we discuss the various credit scoring models used in our study, focusing on three key algorithms: RF, XGBoost, and CatBoost. These algorithms are quite important in evaluating credit risk by analyzing historical data to predict the chance of borrower default. We will begin with an overview of the RF algorithm, which employs an ensemble of decision trees to enhance predictive accuracy. Next, we will explore the XGBoost algorithm, recognized for its efficiency in gradient boosting and handling large datasets. Finally, we will examine the CatBoost algorithm, which excels in managing categorical variables and offers strong performance across various scenarios. For each algorithm, we will outline the underlying mechanisms and relevant functions, as well as discuss their disadvantages.

2.5.1 Random Forest

Random Forest is a potent ensemble learning technique that falls under the category of tree-based machine learning models. During training, the model constructs multiple decision trees and generates predictions by averaging outputs for regression tasks or selecting the most frequent class for classification [5] [26]. The prediction formula for Random Forest is:

$$\hat{y} = \frac{1}{M} \sum_{k=1}^{M} h_k(x) \tag{2.1}$$

where $\hat{y}$ represents the predicted outcome, M denotes the entire number of trees in the forest, and $h_k(x)$ indicates the prediction generated by the k-th tree. What distinguishes Random Forest is its inherent ability to improve generalization and reduce overfitting by combining the predictive strength of multiple trees [14]. Each tree in the forest is built on a subset of the dataset and is trained independently, introducing diversity in the predictions [24]. The "random" in Random Forest refers to the random selection of

features for each split in the trees, further promoting diversity. This diversity contributes to the model's robustness and resilience to noisy data [28]. Random Forest excels in various applications due to its simplicity, scalability, and high predictive accuracy [4]. However, it can be computationally intensive and may require substantial memory. It's particularly well-suited for tasks where interpretability is not the primary concern and focuses on achieving accurate and stable predictions [35]. Random Forest is considered in this study to compare its interpretability with other tree-based algorithms.

2.5.2 Extreme Gradient Boosting

Extreme Gradient Boosting (XGBoost) is a sophisticated and effective gradient boosting framework implementation that is intended for speed and performance [9]. As a tree-based ensemble model, XGBoost builds a sequence of decision trees iteratively, where each subsequent tree fixes the oversights of the one before it [19]. The prediction formula for XGBoost is:

$$\hat{y} = \sum_{t=1}^{T} f_t(x) \tag{2.2}$$

where predicted value, denoted as $\hat{y}$, T represents the total number of trees, and $f_t(x)$ denotes the prediction generated by t-th tree.

What sets XGBoost apart is its optimization of both computational efficiency and predictive accuracy. It incorporates regularization techniques to control overfitting and handles missing data seamlessly [38]. However, without proper regularization, the model remains susceptible to overfitting. One of its key features is its ability to capture complex relationships within the data and the flexibility to handle various types of features [36]. XGBoost leverages a gradient boosting algorithm, which optimizes the loss function by minimizing the gradient, enhancing model precision [24]. Its versatility and quickness make it a preferred option for a variety of applications, such as classification, regression, and ranking tasks [9]. With its versatility and state-of-the-art performance, XGBoost has become a staple in machine learning competitions and real-world applications [9]. Additionally, while it provides feature importance scores, the model's overall interpretability can be challenging due to its ensemble nature.

2.5.3 CatBoost

CatBoost is a potent gradient boosting technique created specifically for dealing with categorical data with ease [38]. Developed by Yandex, CatBoost is known for its robustness and ability to deliver high-quality predictions with minimal parameter tuning [16]. What makes CatBoost distinctive is its capacity to process categorical data without the need for explicit encoding, reducing preprocessing efforts significantly [38]. It employs an ordered boosting technique, introducing an innovative way of handling categorical variables during the tree-building process [38] . CatBoost incorporates efficient strategies for dealing with overfitting, such as the implementation of random permutations during training. This makes it less prone to overfitting and contributes to its generalization capabilities [38]. CatBoost is suitable for various machine learning tasks, ranging from classification to regression, and performs well even with large datasets [23]. Its out-of-the-box support for categorical features, coupled with competitive performance, has made CatBoost an appealing choice for practitioners dealing with diverse datasets in both research and industry [16]. The formula used to predict outcomes with CatBoost is given by:

$$\hat{y} = \sum_{j=1}^{J} g_j(x) \tag{2.3}$$

where $\hat{y}$ denotes the predicted value, J represents the total number of trees in the model, and $g_j(x)$ signifies the j-th tree in the ensemble.

2.6 Explainable AI Approaches

A key area of machine learning is Explainable Artificial Intelligence (XAI), which aims to comprehend and interpret the methods used by sophisticated AI models to make decisions [47]. This transparency is crucial for establishing trust, ensuring fairness, debugging models, and fostering responsible AI development [45]. While numerous XAI techniques exist, this research delves into two prominent approaches, which are SHAP and LIME, each offering unique perspectives on model explainability. Key motivations for XAI extend

beyond mere curiosity. At its core, XAI fosters trustworthiness in AI systems by demystifying their decision-making processes [42]. This transparency allows users to evaluate the models' reliability, identify potential biases, and understand how they arrived at specific outputs [13]. XAI also plays a crucial role in ensuring fairness. By illuminating inherent biases within models, it empowers developers to mitigate them and ensure equitable outcomes for everyone, regardless of individual or group characteristics [45]. Furthermore, XAI acts as a valuable tool for debugging. By providing insights into model behavior, it facilitates the identification and correction of errors and inconsistencies, ultimately leading to better performance and more reliable results [29]. Finally, with growing regulatory frameworks for AI demanding transparency and accountability, XAI emerges as a pivotal tool for ensuring responsible and ethical development and deployment of AI systems [10]. SHAP and LIME, as two prominent XAI techniques, offer distinct approaches to understanding model behavior. SHAP focuses on global model interpretations, providing insights into how features contribute to overall predictions [29][41]. LIME (Local Interpretable Model-agnostic Explanations) specializes in explaining individual predictions, generating local insights into model behavior for specific instances [40].

2.6.1 SHAP in Action

SHAP emerges as a robust framework designed for unraveling the inner workings of machine learning models. This method, rooted in cooperative game theory, assigns values to features, elucidating their impact on a model's predictions. What makes SHAP so brilliant is its capacity to offer both global and local explanations. Globally, it aids in understanding feature importance across the entire dataset, while locally, it delves into explaining individual predictions. SHAP values bring a level of consistency and interpretability by satisfying crucial properties, making them reliable for understanding complex models. Its applications span across model interpretation and feature importance analysis, facilitating a deeper comprehension of the factors influencing a model's decisions. SHAP is versatile, capable of being applied to various machine learning models, from tree-based models like Random Forest to gradient boosting models like XGBoost. While SHAP enhances model

interpretability, users should remain mindful of the potential complexities associated with interpreting models with numerous features. In essence, SHAP serves as a valuable tool for demystifying the "why" behind a model's predictions, fostering trust and informed decision-making. In more elaborate terms, Shapley values ascertain, for every prediction and feature, the extent to which the prediction alters when that particular feature is excluded from the model. The process involves simulating the impact of removing the feature by substituting it with the feature value from a randomly chosen data point. A new prediction is then generated using this hypothetical data point. The disparity between this probability and the anticipated value across the entire dataset is utilized as an estimate of the contribution of that particular feature. The accuracy of these estimations is refined through the repetition of the sampling step and the averaging of results. This methodology aligns with coalition game theory, conceptualizing each feature value as a player evaluated based on its contribution to the total score, which is the difference between the predicted probability and the overall anticipated value of the probability. Notably, SHAP represents Shapley values through a linear model:

$$g(z') = \phi_0 + \sum_{j=1}^{n} \phi_j z'_j \tag{2.4}$$

In this equation:

- $g(z')$ represents the explanation model.
- ϕ_0 is a constant term.
- ϕ_j represents the Shapley value for feature j.
- z'_j represents the value of feature j in the modified data point.

This linear model offers a clear and interpretable method to comprehend how each feature affects the model's predictions.

While SHAP provides a unified and theoretically grounded framework for interpreting model predictions, there are important considerations regarding its application. One key implication is that SHAP enables consistent comparison of feature contributions across

different machine learning models, offering transparency even in complex algorithms like XGBoost or Random Forest[29]. However, the values generated by SHAP can vary depending on the model architecture and the nature of feature interactions. For instance, the same feature may have different importance levels when assessed across models due to differences in how each model captures patterns in the data [35]. Additionally, interpreting SHAP values in practice can be computationally intensive, especially with large datasets or ensemble models, which may limit its scalability and applicability[29]. Within the broader interpretability framework, SHAP is highly effective for post-hoc explanation but should be used alongside other interpretability tools or model transparency strategies to build trust and ensure a comprehensive understanding of model behavior.

2.6.2 Local Interpretable Model-agnostic Explanations (LIME)

LIME emerged as a pivotal tool in the realm of interpretable machine learning, addressing the opacity of complex models. Developed by [40] LIME is designed to generate locally faithful explanations for individual predictions, irrespective of the underlying black-box model. The motivation behind LIME is rooted in the need for model-agnostic interpretability, enabling users to comprehend and trust the decision-making process of even the most intricate machine learning models. LIME operates on the premise of creating simple, interpretable surrogate models around specific instances of interest. The workflow involves selecting a data point for explanation, introducing perturbations to create a perturbed dataset, acquiring predictions from the black-box model for these perturbed instances, and finally training an interpretable, local surrogate model on this perturbed data. The coefficients of this surrogate model then serve as explanations for the prediction of the complex model for the chosen instance. To implement LIME, one can use the lime library in Python. After selecting an instance for explanation, perturbing the data, and obtaining black-box model predictions, a local interpretable model, such as a linear regression model, is trained using the perturbed data. This model estimates the behavior of the complex model in the vicinity of the chosen instance, providing human-understandable insights into the decision-making process. While LIME does not have a

specific mathematical formula, its essence lies in fitting a local interpretable model to estimate the behavior of the complex black-box model for a particular instance. The coefficients of this local model are determined through standard regression techniques. The surrogate model aims to capture the relationship between the perturbed feature values and the predictions, offering a simplified representation of the complex model in the local context.

2.6.3 Global and Local explanations

Global explanations refer to those that offer an overarching comprehension of how the classification model functions as a whole. These explanations interpret the general reasoning behind the model's logic when making predictions. Regulators and data scientists typically prefer global explanations, as they are more interested in the general comprehension of the behavior of the credit scoring algorithm than in the particular logic underlying each forecast. Their responsibility is to guarantee the accuracy, equity, and compliance of the model's predictions. However, local explanations concentrate on a specific model prediction, explaining the reasoning behind the model's choice for a certain data instance. Loan applicants, in particular, are more likely to prefer local explanations, as they want to understand the reasons behind the approval or rejection of their loan application.

2.7 Conclusion

This chapter traced the evolution of credit scoring using modern machine learning techniques. Noteworthy models like Random Forest and XGBoost demonstrated high accuracy in credit risk prediction. The integration of SHAP values enhanced interpretability, offering insights into feature contributions. Descriptions of Random Forest, XGBoost, and CatBoost highlighted their strengths. SHAP, rooted in game theory, emerged as a crucial tool for comprehending model predictions globally and locally. This sets the stage for the application of these models and SHAP in credit scoring, emphasizing transparency and fairness. In this study, SHAP, not LIME, was chosen for its efficiency with tree-based

models used in credit scoring analysis, providing both global and local interpretability crucial for understanding credit scoring decisions.

Chapter3: Methodology

3.1 Introduction

In this chapter, we present the methodology used to analyze the explainability of credit scoring machine learning models using the SHAP approach. The chapter outlines the data collection process, detailing how the data was acquired and prepared for analysis. It then introduces the model development process, where the selected machine learning models are discussed, along with the evaluation metrics used to assess their performance. Finally, the application of SHAP is presented to interpret and explain the models' predictions.

3.2 Dataset and Preprocessing

In this study, we used the Lending club[1] dataset obtained from Kaggle. The selection of the dataset was based on its ease of use,importance and abundance of variables that are critical to model explainability and credit scoring. As a peer-to-peer lending platform, lending club provides real-world loan application data, including borrower financial profiles, loan terms, and repayment outcomes. This makes it highly representative of actual credit decision scenarios. The dataset covers the years 2007 to 2020. There are more than 145 fields in different formats and more than 2.3 million records. Performing analyses using all the variables and data can frequently be challenging due to the complexity and potential for overfitting, as well as the increased computational resources required. The dataset was cleaned, and only relevant continuous variables and two categorical factors were selected as predictor variables, while the rest were excluded. The target variable

[1]Kaggle Lending Club Dataset

selected was the loan status, as it explicitly indicates whether a loan was fully repaid, had late payments, or resulted in default. Since this column contained text values, they were converted into numerical representations before training the models. The dataset exhibits a clear class imbalance, with fewer loans (23.13%) having been charged off compared to the majority that have been fully paid off (76.87%). This imbalance is a common issue in binary classification problems. During training, models tend to develop a bias towards predicting the majority class, which in this case are the fully paid-off loans, while rarely identifying instances of the minority class, i.e., charged-off loans. As the imbalance increases, the model's bias becomes more pronounced, reducing its ability to effectively classify the minority class.

To address this, string features were converted to numerical values (float and int) to enable the model to appropriately process the data. A ratio of 80/20 was then used to divide the dataset into training and testing sets. Various sampling techniques were used on the training data to address class imbalance, including oversampling the minority class and undersampling the majority class (refer to Section 3.2.6 for details on these strategies). These approaches were aimed at ensuring that the models could learn from a more balanced dataset and improve their performance on the minority class. Consequently, a balanced dataset is produced for each technique. We deleted outliers, refer to Table 3.1 which shows such features and what makes them to be outliers. Additionally, unnecessary white spaces were eliminated from string attributes, date attributes were converted into appropriate formats to enable date-based computations, and missing values were imputed using the mean of the respective attribute. However, certain attributes were excluded from the analysis after it was observed that more than half of their values were missing. Imputing these with the mean would have significantly altered their distribution, potentially introducing bias into the model.

Categorical variables were transformed utilizing one-hot encoding, ensuring that each unique category value was represented as a separate field in the dataset. For each record, a '1' was assigned to the relevant category's column, while the remaining columns were filled with '0.' This transformation helped remove unnecessary textual data that could

hinder model performance and enabled the model to effectively differentiate between category values.

3.2.1 Data distribution and feature analysis

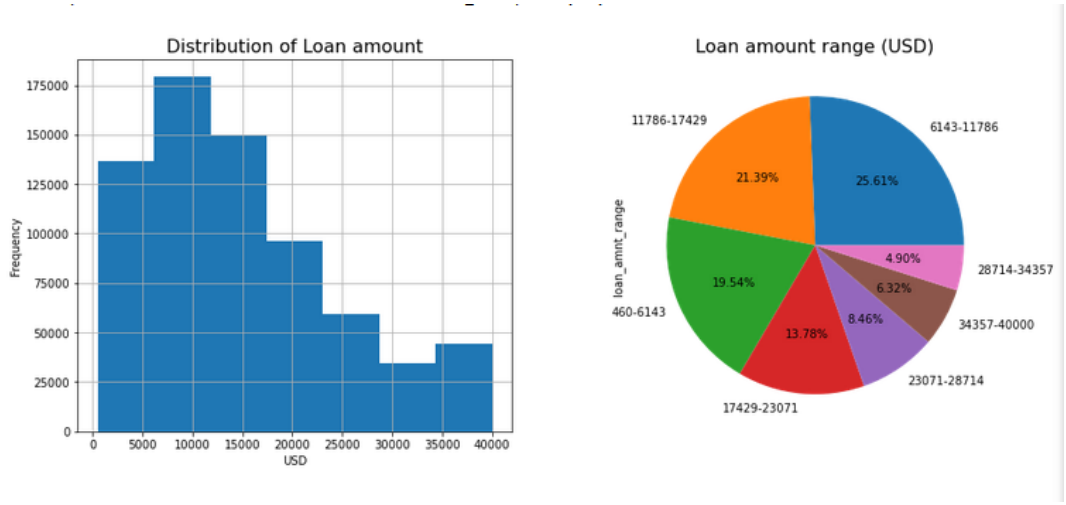


Figure 3.1: Histogram and pie chart showing loan distribution and loan range

The histogram and the pie chart in Figure 3.1 above are generated with an identical number of bins, revealing that the predominant range for loan amounts falls between 6143 and 11786 USD, constituting slightly over 25% of the overall distribution. The loan amounts span from 460 to 40000 USD, showcasing a distinct left-skewness in the data. Notably, the majority of loan applicants sought amounts ranging from 460 to 17429 USD, with a pronounced decline in the number of applications for loan amounts exceeding 17429 USD. Visually, no outliers are evident, signifying the absence of exceptionally low or high loan amounts in the dataset.

Figure 3.2 the bar graph depicts the data distribution concerning the purposes of obtaining a loan. Predominantly, the most common reason for borrowers seeking a loan is "debt consolidation," accounting for approximately 55% of the instances. Following this, "credit_cards" represents a secondary significant reason, albeit at a lower frequency, constituting around 25%. Conversely, loan applications for purposes such as education, renewable energy, or weddings are infrequent. The subsequent pie chart illustrates the distribution of loan terms, representing the duration over which applicants will make payments, with options of either 36 or 60 months. Notably, the majority of loans (approximately 72%) have a term of 36 months, surpassing the number of 60-month loans by double, which accounts for about 28%.

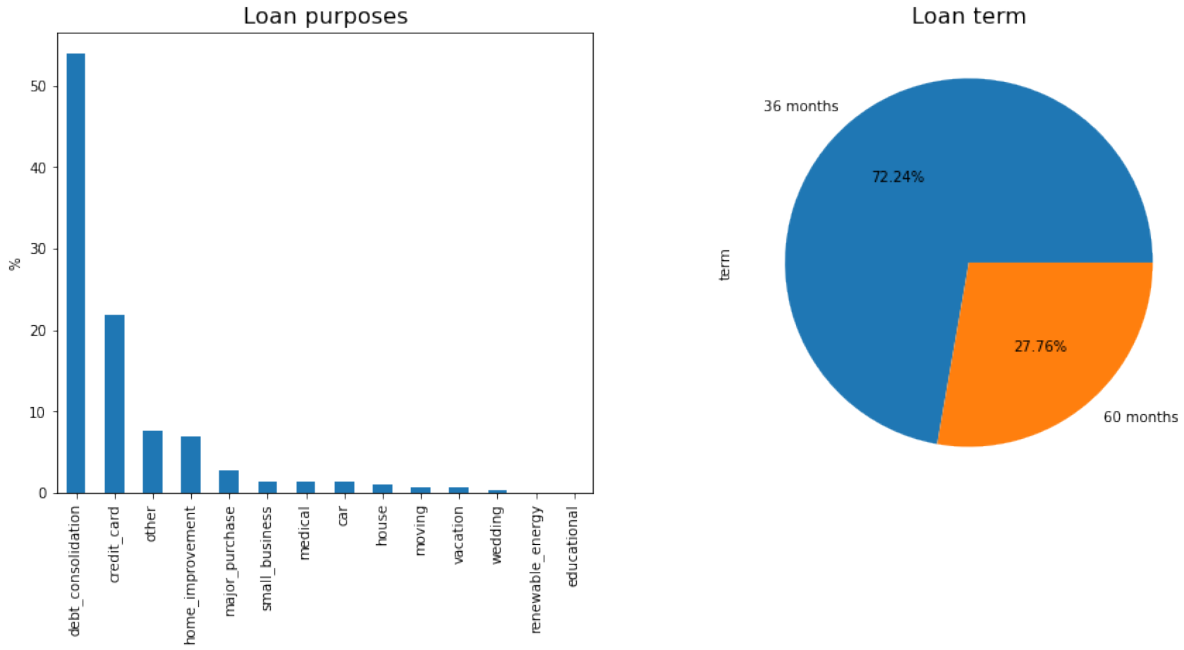


Figure 3.2: Bar graph and pie chart illustrating loan purpose and loan term.

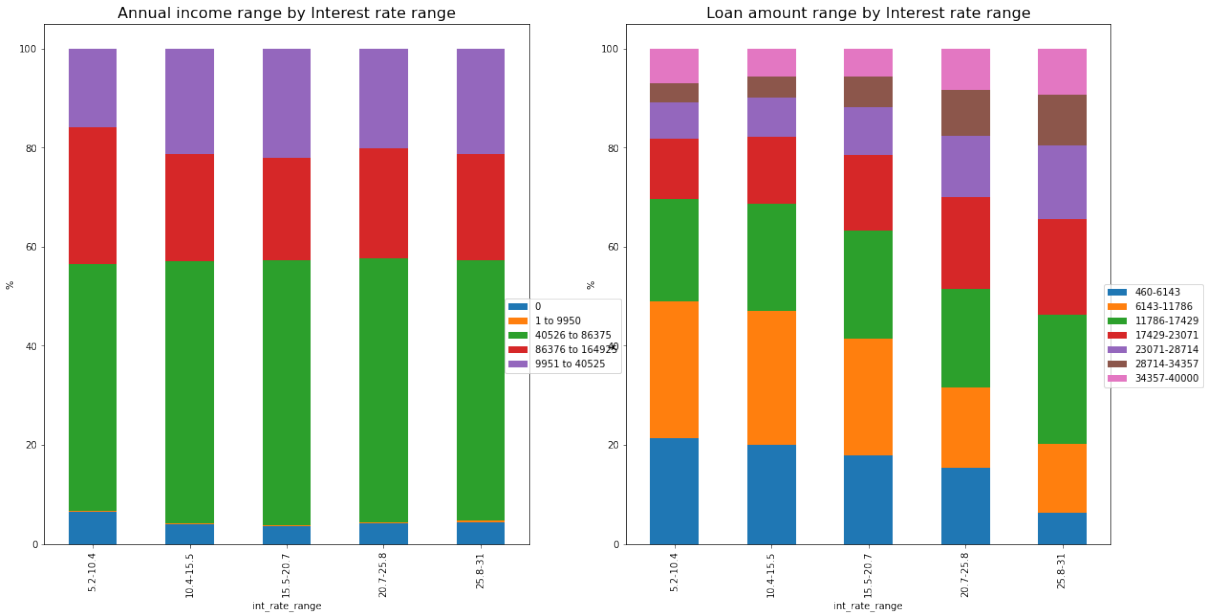


Figure 3.3: Bar graph showing annual income, loan amount vs interest rate.

In Figure 3.3, the first graph indicates that people whose income ranges from 40526 to 86375 USD account for around 50% of the people in each interest range. We can observe a similar pattern in each category of interest range. It is a lack of evidence to conclude that a person with a higher income will have a specific trend to apply more for a higher or lower-risk loan. The graph on the right shows that people who use low loan amounts (from 460-6143 USD) are less likely to accept high interest rate loans from 25.8%-31%,

which makes sense in practice. The people who apply for loan amounts in the range of 6143-11786 USD are mostly in the low interest rate range. It is evident that the proportion of these people is high in the interest range of 5.2%-10.4% but decreases when the interest rate range increases.

Table 3.1: Summary Statistics

Names of the features	**Min**	**25th Quantile**	**Median**	**75th Quantile**	**Max**
loan_amnt	500	8000	12000	20000	40000
int_rate	5.31	9.75	12.79	16.02	30.99
Installment	4.93	249.52	375.54	581.58	1719.83
annual_inc	430	45823	65000	90000	10999200
Dti	-1	11.80	17.63	24.07	999
delinq_2yrs	0	0	0	0	39
inq_last_6mths	0	0	0	1	8
mths_since_last_delinq	0	16	31	50	226

Prior to resampling or normalization, the data distribution of the dataset is examined to find and address any inconsistencies or missing values. Table 3.1 presents key statistical measures such as the minimum, 25th percentile, median, 75th percentile, and maximum values for the first eight numerical features of our dataset. The results indicate that some features, like 'annual income,' contain extreme values, as evidenced by a maximum value significantly higher than the 75th percentile. These outliers could negatively influence classification accuracy, making it necessary to handle them appropriately, as discussed in Section 3.2.2

Figure 3.4 highlights that individuals with annual incomes ranging from 40,526 to 86,375 USD make up the largest proportion across different loan statuses, averaging around 50%. This trend reflects their prevalence among applicants in the dataset. However, it cannot be conclusively stated that individuals with higher incomes are less likely to fall into categories such as 'Charged Off,' 'Default,' or 'Late'.

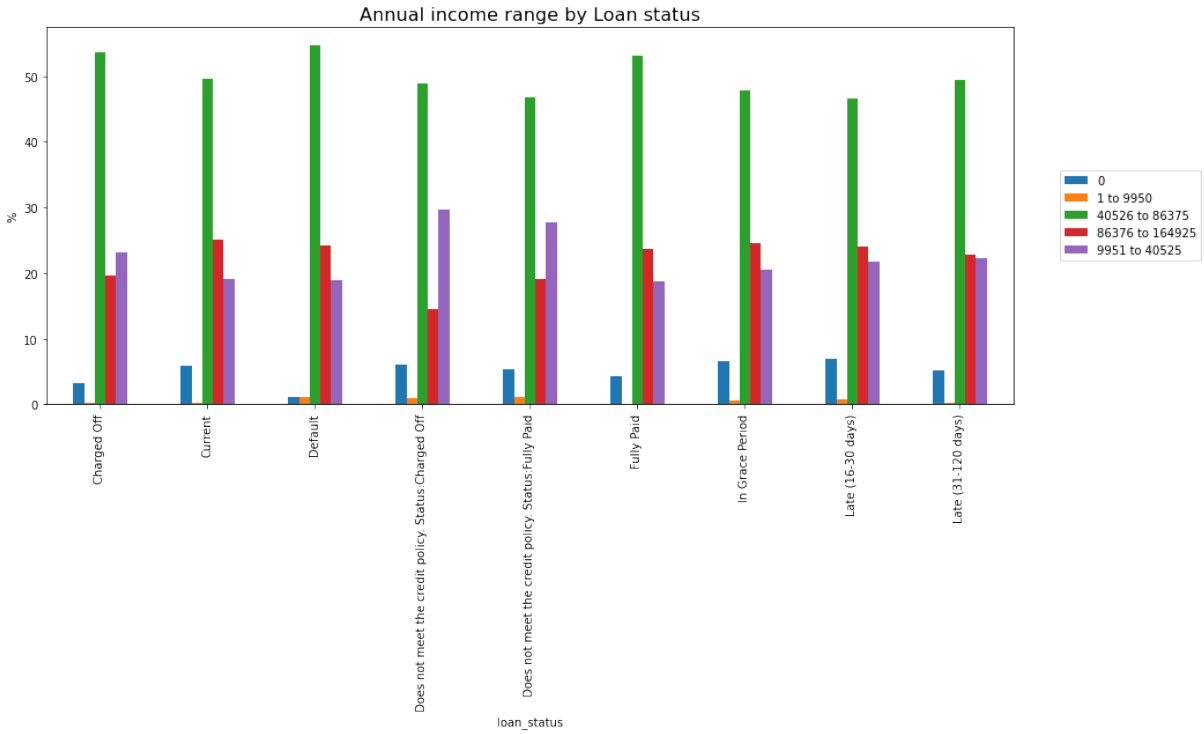


Figure 3.4: Annual income vs loan status

3.2.2 Data Cleaning

Since the `loan_status` variable indicates the loan's condition and determines whether it is classified as 'bad' or 'good,' we utilized it as the dependent variable. The two classes are assigned binary labels as follows:

- Bad = 0
- Good = 1

The initial step involves data preprocessing, where records with statuses such as 'Current,' 'In Grace Period,' or 'Late (16-30 days)' are excluded since these loans are still active and cannot yet be classified as good or bad. Loans marked as 'Fully Paid' are assigned a target label of 1 (Good), while those categorized as 'Charged Off,' 'Default,' or 'Late (31-120 days)' are labeled as 0 (Bad). Although loans in the 'Late (31-120 days)' category have not yet defaulted, they are considered high risk due to the significant delay in payments, indicating potential default. Refer to Figure 3.6 for the data distribution among the loan statuses used before data cleaning and Figure 3.7 for the data distribution after data cleaning.

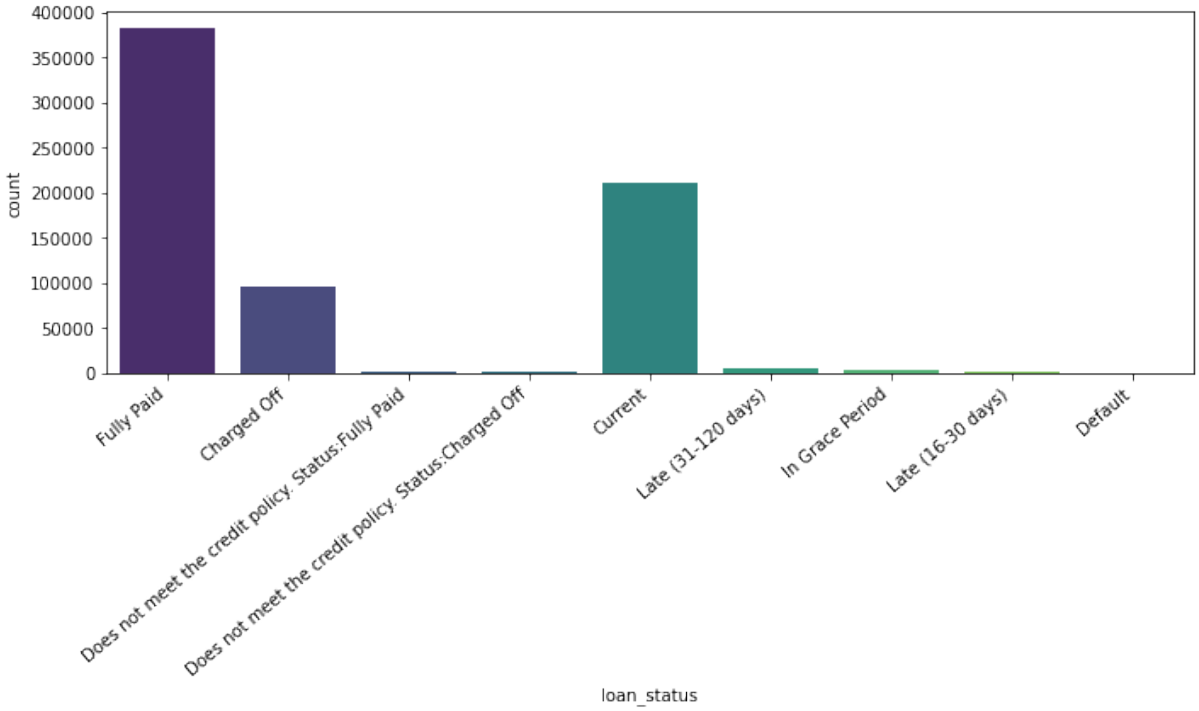


Figure 3.5: Distribution of loan statuses before data preprocessing

Figure 3.5 illustrates the distribution of loan statuses before data cleaning. It provides insights into the prevalence of different loan statuses among the dataset's records. This visualization helps to understand the initial composition of the data, including the proportions of various loan statuses such as 'Current', 'Fully Paid', 'Charged Off', 'Default', 'In Grace Period', and etc. Data cleaning continues further with steps such as eliminating

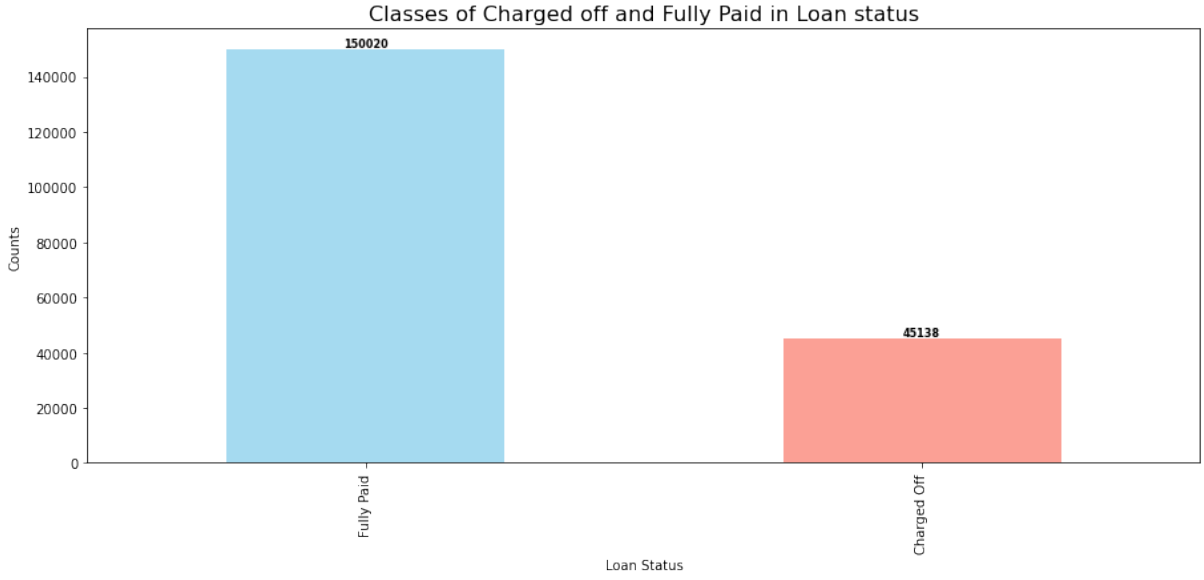


Figure 3.6: Data distribution amongst target classes after data cleaning.

outliers, deleting extra white spaces from string variables, changing date variables to the proper type so that data computations may be performed, and using the attribute's mean to impute empty values. Since we observed that over 50% of the mths_since_last_delinq feature's values were empty and since imputing the occurrences with the mean would drastically alter the variable's distribution and perhaps bias the classifier, we decided not to use it. We eliminated columns with less useful information related to the purpose of this project. Specifically, we removed the "inq_last_6mths" column, which recorded the number of credit inquiries made within the last six months. While credit inquiries are relevant in credit assessment, their direct connection to loan characteristics and borrower behaviour is minimal. Therefore, retaining this data could introduce unnecessary noise and potentially hinder the analysis. One-hot encoding is applied to convert categorical variables, creating a separate column for each distinct category in the dataframe. Every record has a "1" in the column that corresponds to the appropriate category value, and a "0" is allocated to every other column. This process removes the textual nature of categorical variables, which can complicate classifier performance, and aids the model in distinguishing between different category values. For the data on loans issued between 2012 - 2020, the cleaned dataset comprises 1,284,403 instances, with 76.87% classified as good applications and 23.13% as bad applications, as shown in figure 3.10. This indicates that the dataset is imbalanced, with one class being significantly more prevalent, as elaborated in Section 3.2.6. Figure 3.7 effectively utilizes a yellow-purple colour scheme to guide the attention to missing values. Features such as "loan_amnt" display no missing values, as indicated by the consistent purple colour. In contrast, features like "emp_length" reveal the presence of missing values through the presence of yellow regions.

The visual representation below (Figure 3.8) showcases the refined dataset, showcasing its composition after the strategic exclusion of columns deemed excessively incomplete (containing over 50% missing values) to enhance data quality and analysis reliability. Features such as hardship_status, hardship_amount, hardship_length were removed.

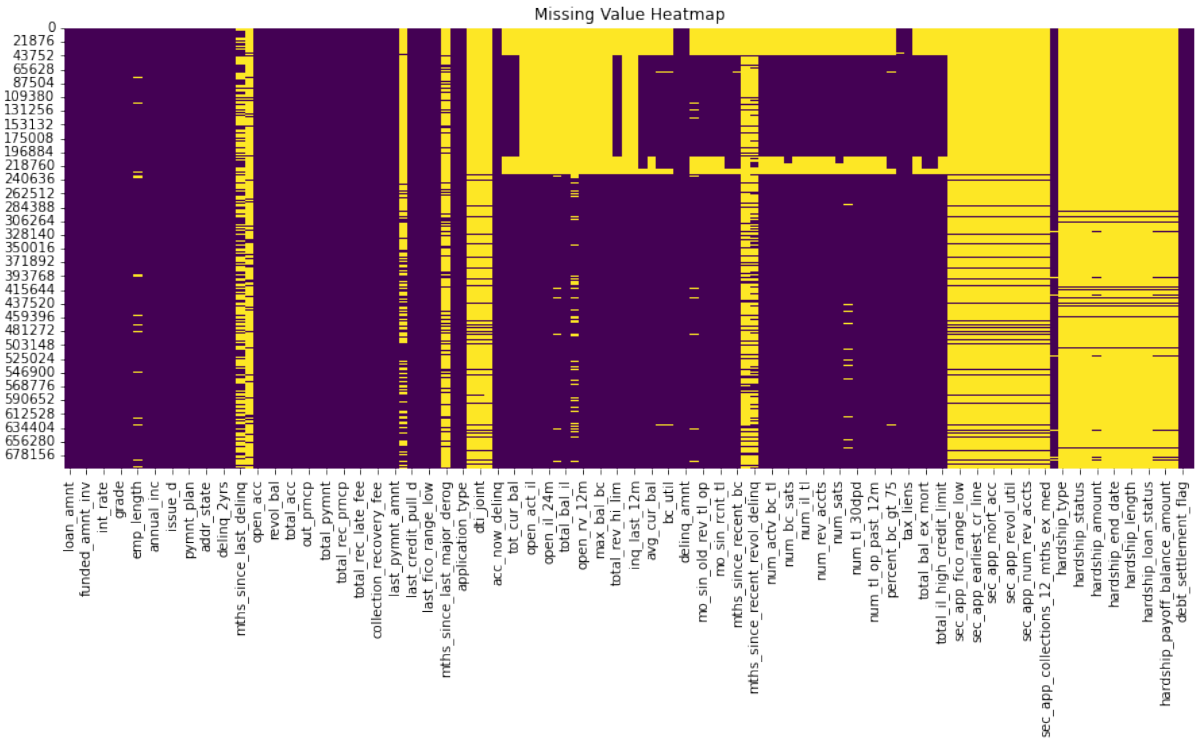


Figure 3.7: Heat map showing missing values.

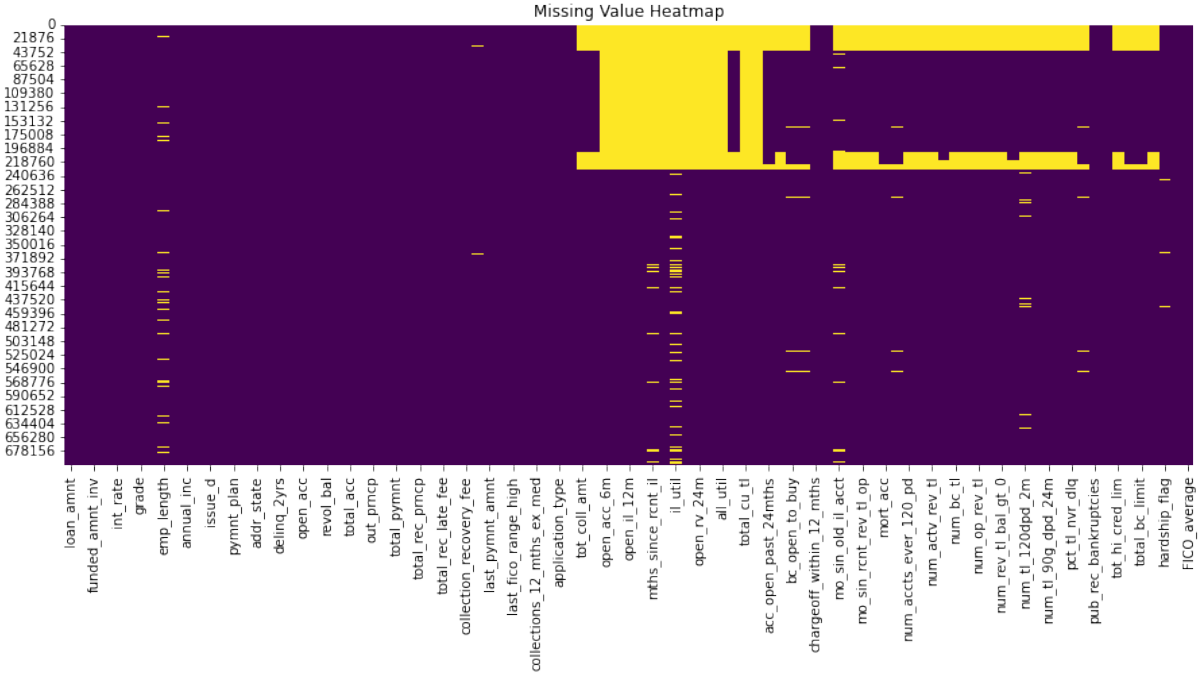


Figure 3.8: Heatmap of null values after excluding features with more than 50% missing data.

3.2.3 Feature Generation

Motivated by [44] and [30], additional variables are introduced to enhance model performance. Research suggests that using ratios and differences between variables can be particularly beneficial for deep learning classifiers, as they establish relationships between attributes and yield more meaningful representations. In this study, four new variables are computed:

- **Credit age**: The period of time between the loan issue date and the borrower's earliest recorded credit line.
- **Loan amount to annual income**: The proportion between the borrower's yearly income and the proposed loan amount.
- **Monthly installments to monthly income**: The proportion of the borrower's monthly payment obligations to their monthly income.
- **Revolving balance to monthly income**: The proportion between the borrower's monthly income and the amount of their revolving credit.

As highlighted in [30], the significance of the installments to income ratio lies in its ability to contextualize the borrower's financial burden. For example, a €500 monthly payment has a vastly different impact on someone earning €1,000 per month compared to an individual with a €10,000 monthly income. The other generated features offer similar advantages by providing a more nuanced understanding of financial capacity.

3.2.4 Feature selection

Feature selection plays a vital role in data preprocessing, as it helps eliminate unnecessary variables that could negatively impact model performance rather than enhance it. A correlation matrix, which presents correlation coefficients between variable pairs, serves as a valuable tool in this process. As shown in figure 3.9 , the correlation matrix reveals that open_acc and num_actv_rev_tl exhibit a strong correlation, with a Pearson coefficient of 0.67, indicating potential redundancy if both are retained. Conversely, variables such

as dti and mort_acc display a low correlation of 0.005, suggesting that they contribute distinct information to the predictive model.

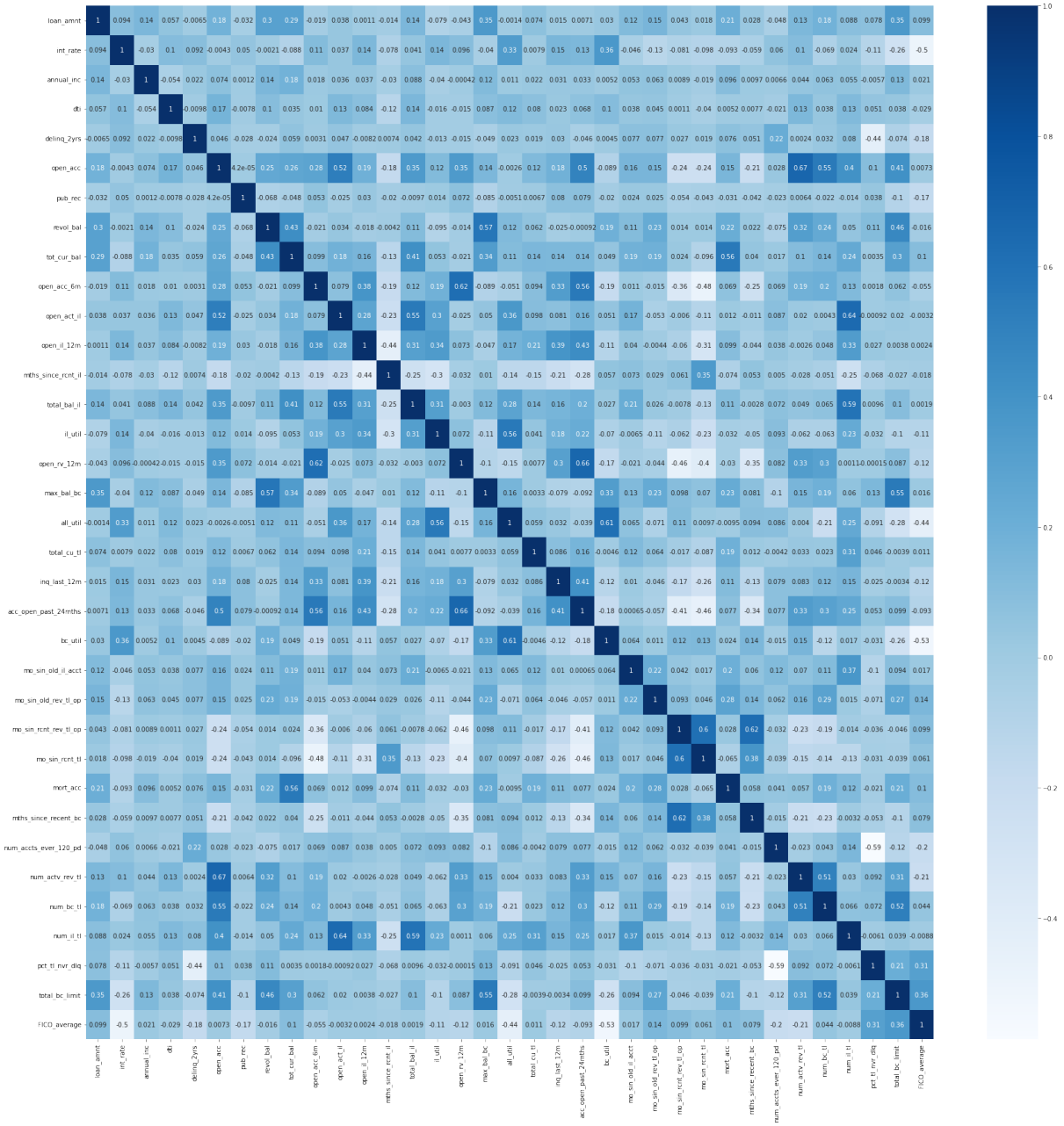


Figure 3.9: Correlation matrix.

3.2.5 Cross Validation

Cross-validation is an effective method that ensures a model's generalizability to unseen data by strategically utilizing all available data. It entails splitting the data into several folds, training the model on various folds repeatedly, then assessing how well it performs

on an unknown test set. K-fold cross-validation, a popular variant, averages model performance across multiple iterations, effectively combating overfitting and giving reliable assessment of performance in the real world. This rigorous approach is crucial for building robust and trustworthy AI systems that effectively generalizes to new data [46]. The predictive performance of the implemented models was rigorously evaluated through a 5-fold cross-validation scheme, implemented using the scikit-learn library. This technique ensured robust and unbiased model assessment by iteratively dividing the data into five mutually exclusive folds, with each fold serving as the testing set for one iteration while the remaining folds were used for training. Stratified sampling was employed to preserve the original class proportions within each fold, ensuring that the test sets accurately reflected the real-world distribution of positive and negative samples (23% bad loan applicants). In each iteration, 80% of the data was used for training, while the remaining 20% served as the unseen test set. The average accuracy score across all five folds was then utilized as the primary performance metric, providing a robust and generalizable measure of model effectiveness.

3.2.6 Imbalanced Data Handling

An imbalanced dataset poses significant challenges in supervised learning classification, as models tend to favor the majority class—often the less critical one—while overlooking the minority class. In credit scoring, accurately identifying bad applications is crucial, as misclassification can result in substantial financial losses for the institution. Consequently, addressing class imbalance is essential in this domain. Two primary data-level methods employed to mitigate this issue are undersampling and oversampling. Figure 3.10 displays the distribution of the data before data balancing.

Undersampling - is a technique employed to address class imbalance by reducing the majority class through the random removal of instances, ensuring a more balanced class distribution. This method is particularly useful when one class has significantly more instances than the other, as it helps prevent biased model performance that favors the majority class. While random undersampling effectively mitigates the issue of class im-

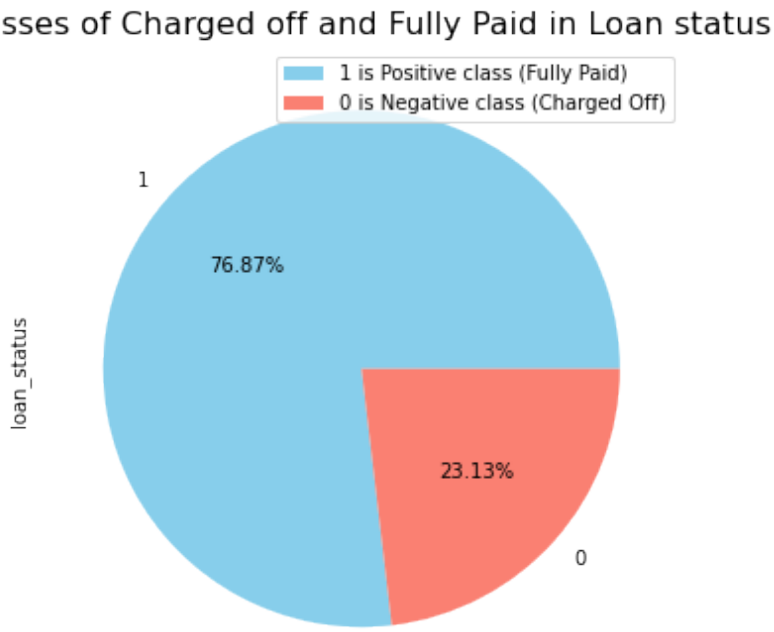


Figure 3.10: Data distribution amongst target classes.

balance, it also poses the risk of discarding potentially valuable data that could contribute to the model's learning process.

In addition to the standard approach of randomly selecting instances for removal, undersampling can be refined by targeting specific instances within the majority class, such as those identified as noise or borderline examples. These instances are often less informative for the learning algorithm, as they may not represent the underlying patterns of the data effectively. By focusing on eliminating such instances, it is possible to retain more relevant data, thereby enhancing the model's ability to generalize from the training set.

In this study, we implemented random undersampling using the XAI library, which is specifically designed to support explainable artificial intelligence methods. Our approach involved preserving all instances of the 'Bad' class, which represents the minority class in the dataset. This is crucial, as retaining all negative examples ensures that the model has a comprehensive understanding of the characteristics associated with undesirable outcomes. Simultaneously, we randomly selected a specified percentage of instances from the 'Good' class, the majority class, to create a more balanced dataset. This strategy not only addresses the issue of class imbalance but also maintains the integrity of the minority

class, which is essential for developing a robust credit scoring model.

By leveraging the capabilities of the XAI library, we ensured that the undersampling process is aligned to the best practices in machine learning, promoting both model performance and interpretability. This careful consideration of class distribution is vital in the context of credit scoring, where the accurate identification of 'charged off' instances can have significant implications for risk assessment and decision-making.

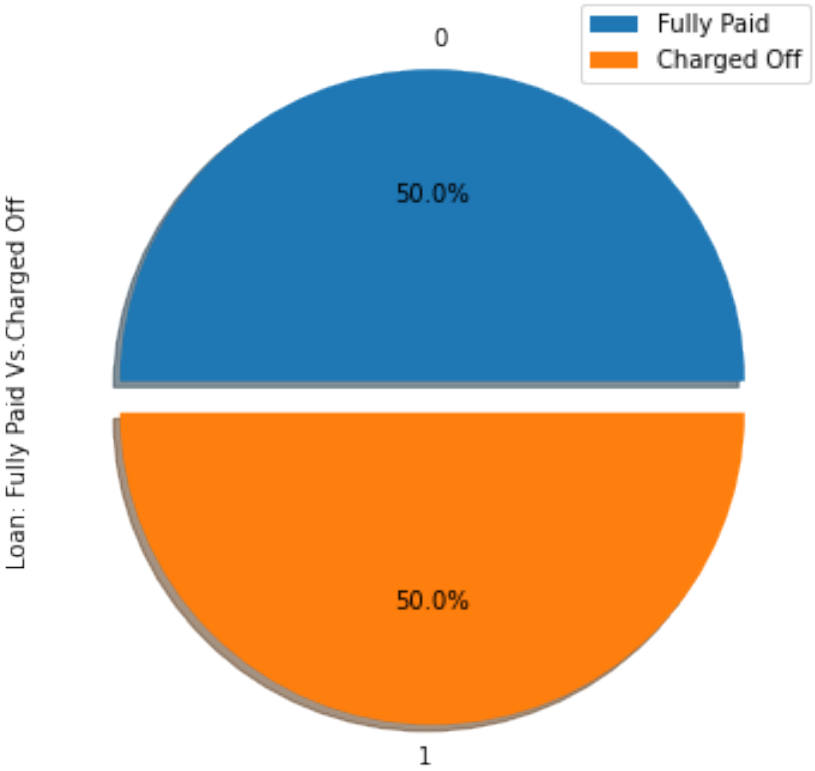


Figure 3.11: Class distribution of undersampling.

Synthetic Minority Over-sampling Technique (SMOTE)-is a widely recognized technique for handling class imbalance by creating synthetic instances of the minority class. Unlike traditional undersampling methods that eliminate instances from the majority class, SMOTE enhances the minority class by creating new synthetic examples through interpolation between existing minority class instances. This approach effectively balances the class distribution while preserving all available data from the majority class, thus minimizing the risk of losing potentially valuable information.

In this study, we implemented SMOTE using the imblearn library, which provides a robust implementation of the technique. We configured the sampling strategy parameter

to 'minority', indicating that the oversampling would focus solely on the minority class. This configuration ensures that the number of instances in the minority class is increased to match that of the majority class, thereby creating a balanced dataset. Specifically, our implementation resulted in a dataset where 50% of the instances belonged to the 'Good' class (the majority class) and 50% to the 'Bad' class (the minority class). The choice of SMOTE as a resampling technique is particularly relevant in the context of credit scoring, where the minority class often represents instances of higher risk. By generating synthetic samples, we aim to provide the model with a richer representation of the minority class, which can lead to improved predictive performance and more accurate risk assessments. Additionally, the ability of SMOTE to create new instances rather than merely duplicating existing ones helps to reduce the likelihood of overfitting, a common concern when handling an imbalanced datasets. These techniques were chosen because random undersampling helps to reduce the influence of the majority class by removing some of its examples, making the dataset more balanced, while SMOTE enhances the presence of the minority class by creating synthetic examples based on existing data. This combination allows for improved class representation without sacrificing valuable information, making the model more sensitive to rare but important outcomes like loan defaults In summary, the application of SMOTE in this study serves to enhance the balancing of the dataset, enhancing the effectiveness of the credit scoring models. The synthetic instances generated through this technique enable the model to learn more effectively from the minority class, ultimately contributing to better classification outcomes and more reliable predictions.

3.3 Development of Credit Scoring Models

In this study, three experiments were conducted to develop and evaluate credit scoring models using different machine learning algorithms. The first model was developed using the Random Forest algorithm, which is known for its robustness and effectiveness in handling classification tasks. Following this, the second model utilized the XGBoost algorithm, a powerful gradient boosting framework that excels in predictive accuracy and computational efficiency. Finally, the third model employed the CatBoost algorithm,

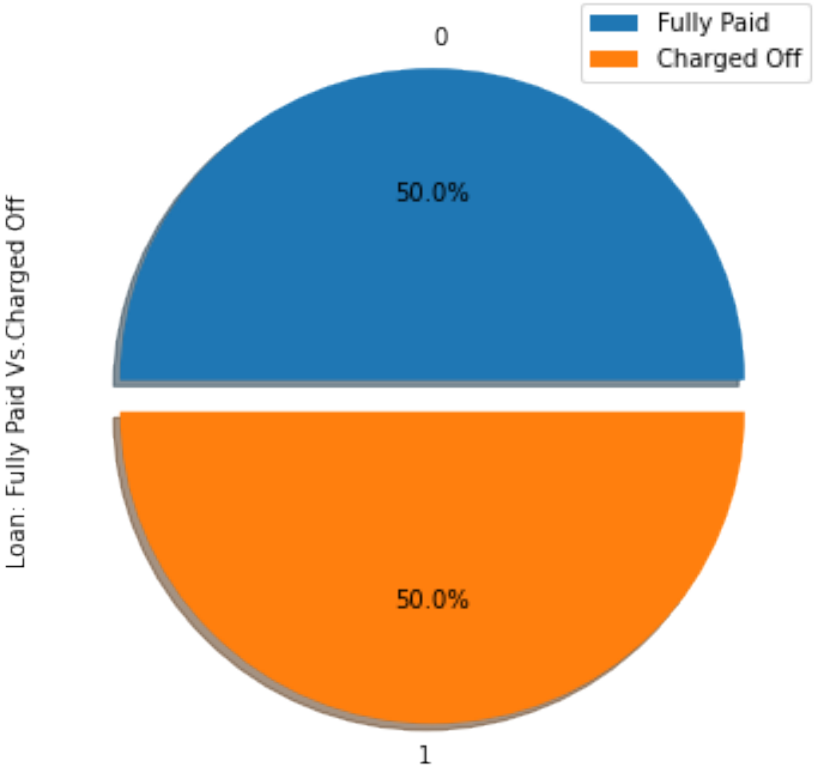


Figure 3.12: Distribution of classes post SMOTE application.

which is particularly adept at managing categorical features and has demonstrated superior performance in various applications. Each model was carefully tuned and evaluated to assess its predictive capabilities in the context of credit scoring, providing insightful information about the pros and cons of these methods.

3.3.1 Random Forest

To develop the Random Forest (RF) model, the `RandomForestClassifier` from the `sklearn` library was utilized, known for its robustness and efficiency in classification tasks. The first step involved setting the hyperparameters, which are crucial for optimizing model performance.

The parameter `n_estimators=100` was chosen, indicating that the ensemble would consist of 100 decision trees. This choice, being the default in `sklearn`, balances model accuracy and computational efficiency. Generally, a higher number of trees improves the model's ability to generalize and lowers the possibility of overfitting, though it also increases

computational time.

The parameter `max_depth=10` was set to limit the depth of each tree. This is important as deeper trees can identify more intricate patterns, but may also lead to overfitting if the depth is too great. Preliminary experiments indicated that a depth of 10 provided a suitable trade-off between capturing complexity and maintaining generalization.

To ensure reproducibility, `random_state=42` was included. This parameter controls the randomness in the bootstrapping process used to create the trees, allowing for consistent results across different runs.

After configuring the hyperparameters, the model was fitted to the training data using the `fit()` method. This step involved training the ensemble of decision trees on the input features and their corresponding labels. For classification tasks, the model typically uses majority voting, aggregating predictions from all trees.

The model's performance was evaluated using metrics including recall, accuracy, and precision to ensure that it met the desired criteria of effectiveness. The careful selection and tuning of hyperparameters, particularly the number of estimators and tree depth, were instrumental in developing a model that was both accurate and efficient, highlighting the importance of meticulous hyperparameter tuning in machine learning model development.

3.3.2 Extreme Gradient Boost

To develop the XGBoost model, the `XGBClassifier` from the `xgboost` library was selected, a powerful and widely used gradient boosting framework known for its efficiency and performance in classification tasks. The initial step involved configuring several key hyperparameters to optimize model performance.

The parameter `n_estimators=100` was set, defining the number of boosting rounds. This choice was based on the principle that a higher number of estimators generally leads to improved model accuracy by allowing the model to learn from the data iteratively. However, computational cost was also considered, with 100 trees providing a balance between performance and efficiency.

To regulate the step size for each boosting iteration, a `learning_rate` of 0.1 was selected.

Although it takes longer boosting rounds for the model to converge, a lower learning rate can improve performance by enabling the model to learn more gradually. The value of 0.1 was selected as it mitigates overfitting while allowing effective learning within a reasonable number of iterations.

Additionally, the parameter `max_depth=6` was set to limit the maximum depth of each tree. This decision was informed by preliminary tests indicating that a depth of 6 provided a suitable balance between identifying intricate patterns in the data and avoiding overfitting. More complex trees can become excessively intricate and may fit noise in the training data; thus, constraining the depth helps maintain generalization.

To ensure reproducibility, `random_state=42` was included. This parameter fixes the seed for the random number generator used in the model, which is crucial for obtaining consistent results across different runs, allowing for reliable comparisons and validations.

Once the hyperparameters were configured, the model was fitted to the training data using the `fit()` method. During this step, the XGBoost algorithm constructs the ensemble of trees successively, each new tree intended to rectify the errors of its predecessor. This iterative learning process is a hallmark of boosting algorithms and is key to their effectiveness.

To make sure the model satisfied the required standards for effectiveness, its performance was evaluated after training using measures including accuracy, precision, and recall. Careful tuning of hyperparameters, particularly the number of estimators, learning rate, and tree depth, was instrumental in achieving a model that was both accurate and efficient, underscoring the importance of thoughtful hyperparameter tuning in machine learning model development.

3.3.3 Catboost

The development of the CatBoost model involved selecting the `CatBoostClassifier` from the CatBoost library, a framework specifically designed to handle categorical features effectively and renowned for its high performance in various machine learning tasks. The initial step in the development process was to configure several key hyperparameters to

optimize model performance.

The parameter `n_estimators=100` was set to specify the number of boosting rounds, allowing the model to build an ensemble of 100 trees. This choice was based on the principle that an adequate number of trees enhances the model's ability to learn complex patterns in the data while balancing computational efficiency.

The parameter `learning_rate=0.1` was selected to control the step size during boosting iterations. A learning rate of 0.1 is used as it strikes a balance between convergence speed and model accuracy. While a lower learning rate can improve performance by allowing the model to learn more gradually, it requires more boosting iterations to achieve convergence. This value was chosen to mitigate the risk of overfitting while ensuring effective learning within a reasonable number of rounds.

Additionally, the `depth=6` parameter was set to limit the maximum depth of each tree. Initial evaluations indicated that a depth of 6 provided a suitable trade-off between capturing data complexity and preventing overfitting. Deeper trees may fit noise in the training data, so constraining the depth helps maintain the model's generalization ability.

To ensure reproducibility, `random_seed=42` was specified. This parameter fixes the seed for the random number generator used in the model, which is crucial for obtaining consistent results across different runs, allowing for reliable comparisons and validations.

Once the hyperparameters were established, the model was fitted to the training data using the `fit()` method. During this step, the CatBoost algorithm constructs the ensemble of trees while automatically handling categorical features and applying techniques such as ordered boosting to enhance performance. This approach aids in learning from the data efficiently and effectively.

To make sure the model satisfied the required standards for effectiveness, its performance was evaluated after training using metrics such as accuracy, precision, and recall to ensure it met the desired criteria for effectiveness. The careful selection of hyperparameters, particularly the number of estimators, learning rate, and tree depth, was instrumental in achieving a model that was both accurate and efficient. This process emphasizes how crucial careful hyperparameter tuning is when creating machine learning models,

particularly when using sophisticated algorithms like CatBoost.

3.4 Evaluation Metric

The model's performance is assessed using common metrics such as precision, recall, F1 score, and accuracy. Each metric offers a distinct perspective on the predictive capabilities of an algorithms:

- **Accuracy**: Represents the overall proportion of correct classifications, calculated as:

$$\text{Accuracy} = \frac{TP + TN}{N}$$

  While widely used, accuracy can be deceptive in imbalanced datasets where the majority class dominates.

- **Recall (True Positive Rate)**: Measures the algorithm's ability to correctly identify positive cases (charged-off loans) among all actual positive cases, calculated as:

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

- **Specificity (True Negative Rate)**: Assesses the algorithm's accuracy in correctly classifying negative cases (fully paid loans) among all actual negative cases, calculated as:

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

- **Precision**: Captures the proportion of predicted positive cases that are truly positive, calculated as:

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

- **F1-score**: Provides a balanced measure of precision and recall, calculated as:

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

These four metrics will be used to assess the model's performance, utilizing a binary classification system with the following definitions: positive example (Positive), negative example (Negative), correct prediction (True), and incorrect prediction (False). The results will be summarized in a confusion matrix, which categorizes true and false positives and negatives. Refer to Table 3.2.

- **True Positives (TP)**: Loans correctly predicted as charged-off, aligning with their actual status.
- **False Negatives (FN)**: Loans incorrectly predicted as fully paid, despite their actual charged-off status.
- **False Positives (FP)**: Loans incorrectly predicted as charged-off, despite their actual fully paid status.
- **True Negatives (TN)**: Loans correctly predicted as fully paid, aligning with their actual status.

	Predicted Charged-off	**Predicted Fully Paid**
Observed Charged-off	True Positives (TP)	False Negatives (FN)
Observed Fully Paid	False Positives (FP)	True Negatives (TN)

Table 3.2: Confusion Matrix

3.5 Implementation of Shapley Additive explanations

To implement SHAP (Shapley Additive Explanations) for the trained models and generate the SHAP values, a structured approach was followed. Initially, the test set was prepared for SHAP value computation, which included addressing any preprocessing needs such as handling missing values and encoding categorical features. SHAP explainer objects were then created for each model. The `shap.TreeExplainer` was utilized for both the RF and XGBoost models, while the `shap.CatBoostExplainer` was employed for the CatBoost model.

After establishing the explainers, SHAP values for the test set were computed. Specifically, the commands `shap_values_rf = explainer.shap_values(X_test)` and `shap_values_xgb = explainer.shap_values(X_test)` were executed for the Random Forest and XGBoost models, respectively. For the CatBoost model, SHAP values were generated using `shap_values_cb = explainer.get_shap_values(X_test)`.

Various visualizations were employed to analyze the SHAP values. Summary plots illustrated feature importance across all predictions, dependence plots showed the relationship between features and their corresponding SHAP values, and force plots provided a detailed view of how specific features influenced model outputs. These visualizations are discussed in more detail in Section 4.3. The SHAP values across the three models were compared to assess consistency in feature importance and overall model interpretability. This comparison involved analyzing the distribution of SHAP values for key features and identifying any discrepancies in feature impact among the models. The insights gained from this analysis enhanced the understanding of model behavior.

3.6 Conclusion

The methodology involved careful preprocessing of the dataset, including handling missing values and encoding features. Data distribution was analyzed to address imbalances. Credit scoring models, including Random Forest, XGBoost, and CatBoost, were developed and evaluated using metrics like accuracy and F1 score. Shapley Additive Explanations (SHAP) were implemented to enhance model interpretability, providing insights into feature importance and model predictions

Chapter4: Experimental Results and Analysis

4.1 Introduction

This chapter delves into the outcomes of machine learning models, encompassing their predictive accuracy and effectiveness. The algorithms evaluated include Random Forest, Extreme Gradient Boosting, and CatBoost. Furthermore, it scrutinizes the interpretability aspects, shedding light on how well the models can be understood and trusted. Accuracy, recall, precision, and F1 score are the main assessment measures used to gauge the models' performance. In addition, SHAP is used to analyze the interpretability of the models.

4.2 Model performance and analysis

In Table 4.1, when analyzing the performance of Random Forest, XGBoost, and CatBoost, several key insights emerge. Random Forest shows moderate performance with 76% accuracy, 62% recall, 61% precision, and a 64% F1 score. This suggests that RF, despite its ensemble nature and ability to handle complex data, struggles somewhat with the class imbalance. Its relatively balanced precision and recall indicate that it is not heavily biased towards the majority class. XGBoost and CatBoost both achieve slightly higher accuracy at 78%, but their recall (56% for XGB, 56% for CatBoost) and F1 scores (56%, 67% for XGB and CatBoost respectively) are lower than Random Forest's. This pattern

indicates that while these gradient boosting models are adept at overall prediction, they are more affected by the class imbalance, likely prioritizing the majority class. The higher accuracy coupled with lower recall suggests that they might be overlooking some of the minority class instances, a common issue in imbalanced datasets. CatBoost's marginally higher precision (56% vs 68% for XGBoost) hints at a slightly more conservative approach in predicting the positive class. The similar performance of XGBoost and CatBoost is not surprising given their related gradient-boosting foundations.

Table 4.1: Performance of the models before balancing the dataset

Models	Accuracy	Recall	Precision	F1 score
RF	76%	62%	61%	64%
XGBoost	78%	56%	68%	56%
Catboost	78%	56%	56%	67%

Table 4.2 illustrates the impact of undersampling on model performance. RF, XGB, and CatBoost models after undersampling the Lending Club dataset, reveals intriguing insights into how each model's properties interact with the undersampling technique. RF shows a notable improvement, achieving 84% in all parameters (F1 score, accuracy, recall, and precision). This exceptional performance can be associated to RF's ensemble nature, which integrates several decision trees trained on diverse bootstrap samples. The undersampling process aligns well with RF's ability to handle varied data distributions and its feature importance mechanism, allowing it to adapt effectively to the balanced dataset. In contrast, both XGB and CatBoost show a decrease in accuracy to 72% but achieve balanced performance across all metrics. This outcome reflects the impact of undersampling on these gradient-boosting models. Although undersampling addresses class imbalance, potentially improving their ability to identify minority class instances, it also leads to information loss that affects overall accuracy. The sequential learning process and specific optimization techniques of XGB and CatBoost make them more sensitive to changes in data distribution introduced by undersampling. The equalized metrics in accuracy, recall, precision and F1 score for all models indicate that undersampling has successfully balanced the dataset, forcing the models to give equal importance to both classes.

Table 4.2: Performance of the models after performing undersampling.

Models	Accuracy	Recall	Precision	F1 score
RF	84%	84%	84%	84%
XGBoost	73%	73%	73%	73%
Catboost	72%	72%	72%	72%

In Table 4.3, applying Synthetic Minority Over-sampling Technique (SMOTE) to the dataset, the performance metrics of RF, XGBoost, and CatBoost demonstrate varying levels of effectiveness. After implementing SMOTE, the RF model maintained an accuracy of 84%, with the recall, precision, and F1 score all at 84%. This consistent performance across all metrics indicates that RF effectively learned from the augmented dataset, which balanced the class distribution by generating synthetic examples of the minority class. RF's ensemble nature, which blends several decision trees, permits it to generalize well and capture the underlying patterns in the data, making it robust to the variations introduced by SMOTE.

In contrast, both the XGBoost and CatBoost models exhibited lower performance metrics, with accuracy, recall, precision, and F1 score all at 72% for both models. This performance drop compared to the RF model suggests that while SMOTE helped balance the data set, it may not have been as beneficial for these gradient-boosting algorithms. XGBoost and CatBoost are designed to optimize for specific loss functions and may be more sensitive to the noise introduced by synthetic samples, which may cause overfitting if the model fails to differentiate between genuine and artificially created data points. However, the Random Forest model demonstrated robust performance, maintaining high metrics throughout the dataset, indicating that overfitting did not occur. The ensemble approach of RF, which integrates predictions from several decision trees, enables it to generalize effectively and manage the additional noise from synthetic data. This ability to maintain performance illustrates that, in this case, the application of SMOTE kept the model's capacity to generalize to unseen data unaffected.

Interestingly, when comparing the performance of SMOTE with undersampling, all three models RF, XGBoost, and CatBoost—exhibit similar performance. This observation suggests that these models, despite their differing architectures, are resilient to both reduced

and augmented data. For RF, its ensemble approach, which aggregates predictions from multiple decision trees, allows it to generalize effectively to different data distributions without significant performance loss. The consistent accuracy, recall, precision, and F1 score indicate that RF captures essential patterns in the data, whether it has been undersampled or augmented using SMOTE. The robustness of the RF to noisy and synthetic data ensures that the balance between majority and minority classes, regardless of the method used, plays a key role in maintaining high performance.

For XGBoost and CatBoost, the performance stability across both SMOTE and undersampling stems from their gradient boosting frameworks, which handle class imbalance through iterative boosting processes. These models consistently identify relationships in the data, whether synthetic instances are introduced or the dataset size is reduced. Their use of regularization techniques prevents them from overfitting to noise from SMOTE or reducing data from undersampling. Although more sensitive to hyperparameter tuning than RF, the ability of XGBoost and CatBoost to maintain similar performance across sampling methods highlights their ability to balance between underfitting and overfitting. This emphasizes that when properly tuned, both SMOTE and undersampling can result in stable, comparable outcomes for these models, even in imbalanced datasets.

Table 4.3: Performance of the models after performing SMOTE.

Models	Accuracy	Recall	Precision	F1 score
RF	84%	84%	84%	84%
XGBoost	72%	72%	73%	72%
Catboost	72%	72%	72%	72%

4.3 Model Explainability analysis

In this subsection, we dive into the interpretations provided by the SHAP explainer for the outcomes of the RF, XGBoost, and CatBoost models. The SHAP explainer offers a unified measure of feature importance, which enhances the interpretability of complex models. Our analysis will expand on these interpretations and scrutinize each model's outcomes to determine their alignment with established financial logic.

4.3.1 SHAP Feature Importance Plot

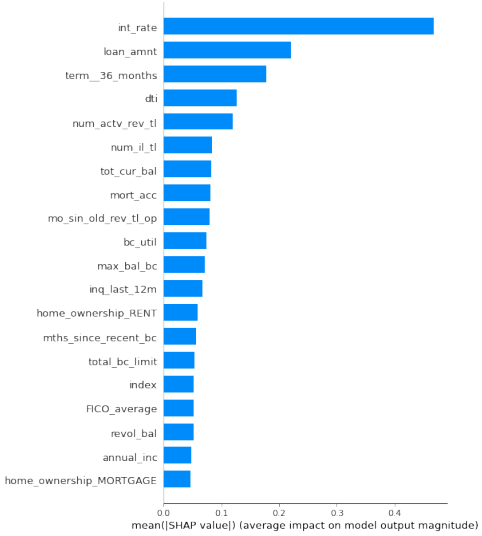
It is a valuable visualization that summarizes the significance of features across all predictions constructed using machine learning model. Features are shown on the y-axis, arranged by their mean absolute SHAP values, which measure the average impact of each feature on the model's predictions. The values on the x-axis shows the magnitude of these SHAP values, allowing for a clear comparison of feature importance. Longer bars indicate features that have a greater impact on the model's output, while shorter bars represent less influential features. This plot not only highlights the most critical factors driving predictions but also enhances model interpretability by offering insights into the interactions between various features and the model. By visualizing feature importance in this manner, stakeholders can make well-informed decisions by considering the relevant factors that significantly influence outcomes, thereby improving trust and transparency in the model's predictions. The most influential features observed from the SHAP feature importance plots, vary across the models.

For the XGBoost model refer to figure 4.1a, the top three features that significantly impact the loan default predictions are int_rate, loan_amount, and term_36_months, each aligning with fundamental credit risk concepts. A higher interest rate typically reflects a greater perceived lending risk, as borrowers with weaker credit profiles are often assigned higher rates, which in turn increase repayment costs and the likelihood of default. The loan amount's significance suggests that larger loans may place more financial strain on borrowers, raising the chance of missed payments and posing greater loss potential for lenders. The prominence of the 36-month loan term indicates that repayment structure also plays a critical role; shorter terms often involve higher monthly payments, which may be difficult for some borrowers to manage, despite potentially lower interest over time.

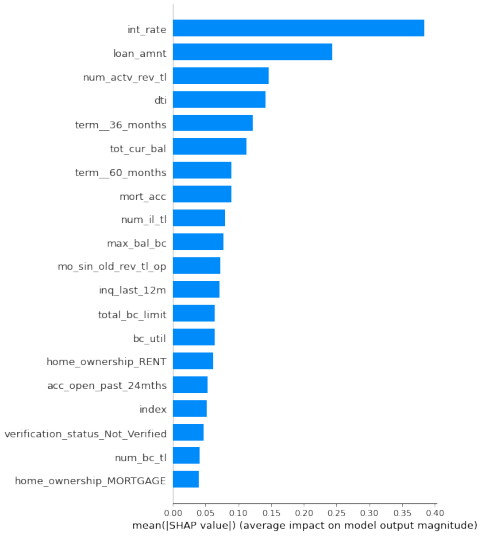
In the case of the CatBoost model refer to figure 4.1b, the most influential features in predicting loan defaylt are int_rate, loan_amount, and num_actv_rev_tl.The prominence of interest rate supports the idea that borrowers facing higher borrowing costs are more likely to default, as elevated interest rates increase repayment pressure and often correspond to higher credit risk profiles. Loan amount also ranked highly, reinforcing the notion that

larger borrowing amounts may strain a borrower's financial capacity, thereby increasing the potential for missed repayments. The number of active revolving credit lines reflects the borrower's existing credit exposure; a higher number may indicate overextension or greater reliance on revolving debt, both of which can signal higher default risk. These findings align with common credit assessment practices, highlighting how financial stress and indebtedness are captured by machine learning models and made interpretable through SHAP values.

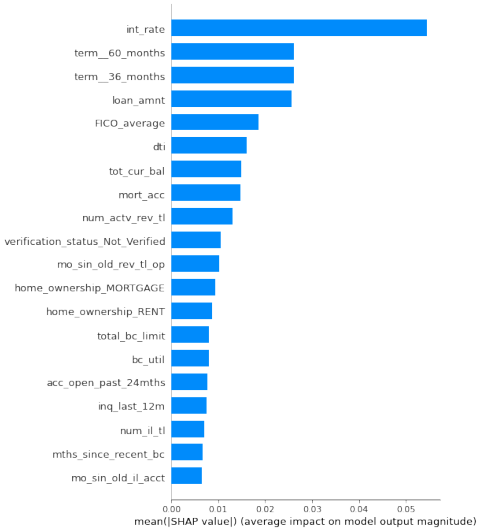
Similarly, for the RF model refer to figure 4.1c, the most influential features in predicting loan default f are int_rate, term_36_months, and term_60_months. The interest rate remains a key indicator of borrower risk, as higher rates typically reflect lenders' response to lower creditworthiness and increase the repayment burden on borrowers, thereby raising default probability. The inclusion of both 36-month and 60-month terms among the top features suggests that loan duration plays a significant role in assessing risk. Shorter terms often come with higher monthly instalments, which can strain borrowers' cash flow, while longer terms, although reducing monthly payments, may be associated with increased overall interest costs and long-term financial exposure. The SHAP values therefore reinforce that repayment structure and cost of borrowing are central factors in credit risk assessment, supporting the interpretability of the RF model's predictions within established financial logic. Across all models, the feature int_rate consistently ranks as the most critical in influencing predictions, indicating its substantial impact on the models' decision-making process.



(a) SHAP feature importance-XGBoost.



(b) SHAP feature importance-catboost.



(c) SHAP feature importance-Random forest.

Figure 4.1: SHAP Feature Importance for each model

4.3.2 SHAP summary plot

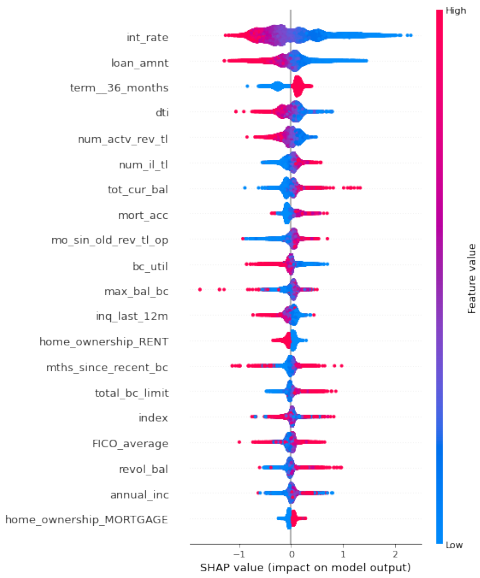
It integrates feature importance with feature effects to evaluate the global influence of features on the model. The y-axis shows features ranked from most to least important, while the x-axis represents the corresponding Shapley values for individual predictions. Each point on the plot signifies a specific Shapley value, with colors ranging from blue (low impact) to red (high impact) to illustrate the effect on model output. To prevent overlap and improve visualization, points are distributed along the y-axis, providing in-

sight into the distribution of Shapley values for each feature.

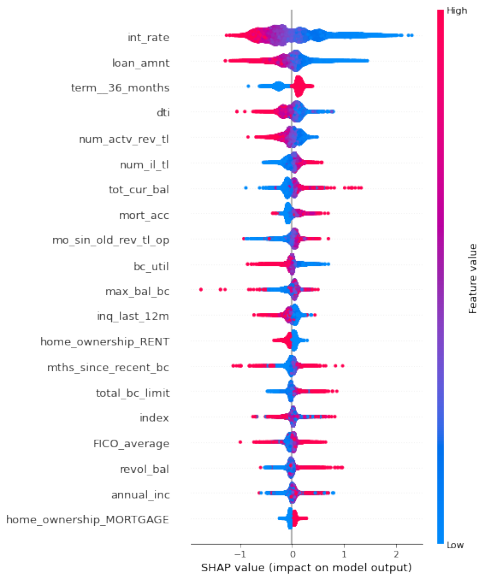
Figure 4.2a illustrates summary plots for XGB models. The most significant features include int_rate (interest rate) and loan_amnt (loan amount), where higher interest rates negatively impact the probability of a loan being classified as "Fully Paid," indicating higher risk and lower creditworthiness. Additionally, the term_30_months (loan term of 30 months) and FICO_average (average FICO score) are emphasized, showing that XGBoost takes a comprehensive view of creditworthiness, considering both loan duration and borrower credit profiles. The broader distribution of SHAP values in XGB suggests its capability to capture intricate relationships between features, making it a robust model for predicting loan outcomes.

Figure 4.2b illustrates summary plots for catboost models. CatBoost is known for its efficiency in handling categorical data, which is evident in its feature importance distribution. The model highlights features such as dti (debt-to-income ratio) and tot_cur_bal (total current balance) alongside the consistently significant int_rate and loan_amnt. Higher values of dti negatively impact predictions, indicating that borrowers with higher debt relative to their income are seen as higher risk, affecting their creditworthiness. Conversely, higher tot_cur_bal values positively impact predictions, reflecting better financial stability and higher creditworthiness. CatBoost's focus on these financial balance indicators underscores its strength in capturing the borrower's overall financial health and obligations, providing a distinct perspective on loan risk assessment.

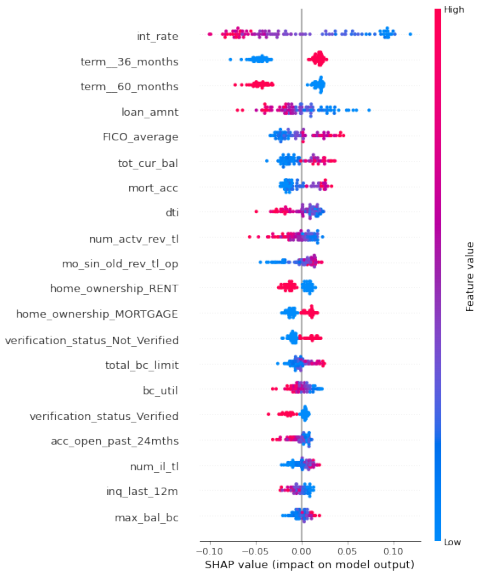
Figure 4.2c presents the SHAP summary plot for the RF model, highlighting the significance and impact of various features on loan repayment predictions. The int_rate feature emerges as the most influential, with higher interest rates correlating with a decreased likelihood of a loan being classified as "Fully Paid," reflecting the association between higher interest rates and increased risk. Similarly, larger loan amounts negatively impact repayment probability, indicating that higher loan amounts may pose greater risk. Regarding loan terms, the 36-month term positively influences repayment likelihood, suggesting that shorter-term loans are perceived as less risky, while the 60-month term exhibits mixed effects, indicating a more complex relationship with repayment outcomes.



(a) Summary plot for-XGBoost.



(b) Summary plot for-catboost.



(c) Summary plot-Random forest.

Figure 4.2: Summary plot for each model

Comparing the SHAP summary plots with the feature importance plots, it is observed that the ranking of features is generally consistent across both methods, with int_rate being the most influential feature across all models. However, slight differences in the ranking and inclusion of certain features, such as term_60_months in Random Forest, suggest that SHAP values provide additional insights into feature contributions that may not be fully captured by feature importance methods.

4.3.3 SHAP force plot

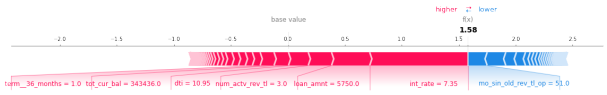
The SHAP force plot shows how each feature impacts the model's output, moving it from the base value prediction (the average predicted outcome across the training set) to the final prediction for the target class. It is centered along the x-axis, offering an explanation of how features are expected to influence the target class. Refer to Figures 4.3a, 4.3b, and 4.3c, features that positively impact the prediction are represented in red, while those with a negative influence appear in blue. The direction of influence depends on the output value, f(x), relative to the base value. If f(x) exceeds the base value, red-colored features push the prediction toward a higher range, whereas blue-colored features drive it downward. Additionally, the most significant feature effects are displayed at the bottom of the plot.

Figure 4.3a illustrates various features and their respective contributions to influencing the model output for the class 'Fully Paid' relative to the base value. Since the output value (f(x)) is 1.58, which is higher than the base value of 0.0, the features represented in red shift the prediction towards higher values. For instance, "int_rate" and "term_36_month" positively impact the model's prediction, as indicated by their presence in the red region, pushing the output to the right. Conversely, feature such as "dti" and "mort_acc" appear in the blue region, signifying a negative impact that drives the prediction toward lower values. However, the overall force driving the prediction upward is stronger, as the cumulative impact of features in the red region outweighs that of the features in the blue region.

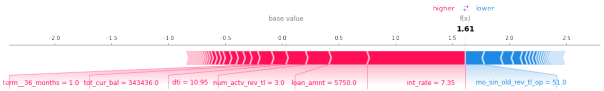
Figure 4.3b highlights various features and their individual contributions to shifting the model output from the base value. Given that the output value, (f(x)) is 1.61, which is significantly higher than the base value of almost 0.0, the features represented in red pushes the prediction towards higher values. For instance, "int_rate" positively influences the model's prediction, as indicated by its presence in the red region, driving the output to the right. In contrast, the feature "addr_state_FL" appears in the blue region, signifying a negative impact that pulls the model's prediction toward lower values.

In Figure 4.3c, the various features are depicted alongside their respective contributions

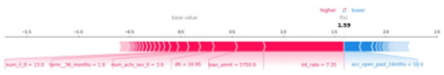
to influencing the model output relative to the base value. With an output value (f(x)) of 1.59, exceeding the base value of 0.0, features highlighted in red are shifted towards higher values. For instance, "int_rate" exhibits a positive impact on the model's prediction, evident in the red area, thus influencing the model's prediction towards higher values. Conversely, the feature "acc_open_past_24months," depicted in the blue area, negatively impacts the model's prediction, shifting it towards lower values. This visualization offers insights into the directional influence of individual features on the model's predictions, with red denoting positive contributions and blue denoting negative contributions.



(a) SHAP force plot-XGBoost.



(b) SHAP force plot-catboost.



(c) SHAP force plot-Random forest.

Figure 4.3: SHAP force plot for each model

4.4 Conclusion

RF demonstrated the most balanced performance across accuracy, recall, precision, and F1 score, both before and after data balancing techniques. XGBoost achieved higher accuracy before balancing but exhibited a trade-off with lower recall and precision. CatBoost achieved consistent recall and precision, with a slightly higher F1 score than XGBoost. Undersampling significantly improved Random Forest's performance, while both XGBoost and CatBoost showed slightly lower accuracy after balancing. SMOTE had minimal impact on Random Forest's performance but further decreased the accuracy of XGBoost and CatBoost. These results highlight the importance of choosing the appropriate balancing technique for each model, as some may be more susceptible to its impact than others.

Chapter5: Conclusion, Summary,Future work and Limitations

5.1 Summary of results

This study proposes an interpretable credit scoring model that includes the XGBoost, RF, Catboost algorithms and SHAP to improve prediction accuracy and interpretability. The XGBoost and Catboost algorithms effectively predict loan status, while SHAP analyses the contribution of each input characteristic or parameter to the performance of the models. The model accurately reflects the impact of various parameters on the prediction performance of the credit model and provides a basis for formulating credit scoring strategies. This interpretable model helps to accurately analyse the causes of bad customers, allowing credit service personnel to make informed decisions, avoid blind decision-making, and improve work quality and efficiency. It also investigated the performance of three models. The evaluation was conducted before and after applying dataset balancing techniques, specifically undersampling and SMOTE. Before balancing, the models demonstrated varying degrees of effectiveness, with Random Forest demonstrating balanced performance, XGBoost displaying higher accuracy but with trade-offs in recall and precision, and CatBoost demonstrating balanced recall and precision with a higher F1 score. Undersampling significantly improved the RF model's performance, yielding consistently high values across accuracy, recall, precision, and F1 score metrics. However, XGBoost and CatBoost exhibited a decline in accuracy post-undersampling, with consistent performance in recall, precision, and F1 score. The application of SMOTE resulted in

sustained high performance for Random Forest, maintaining an accuracy of 0.84 across all metrics. In contrast, XGBoost and CatBoost experienced a modest decrease in accuracy, demonstrating consistent recall, precision, and F1 scores of 0.72. In addition, optimizing the parameters with cross validation further improves the prediction accuracy.

5.2 Conclusion

In conclusion, this study demonstrates the importance of evaluating both model performance and interpretability when making data-driven decisions. While Random Forest achieved the most balanced performance in this specific case, its interpretability may not be as crucial for applications where black-box models are acceptable. Conversely, XGBoost and CatBoost, although achieving slightly lower accuracy, offer valuable insights into their decision-making process through SHAP values, potentially enhancing trust and understanding in loan assessment applications. Using the SHAP exploration model, the study identifies the importance of features, decodes complex relationships between input variables, and explains the impact of input features on the data model. This analysis improves the interpretability of the credit scoring model and provides valuable insights for risk assessment in financial lending.

5.3 Future work and Limitations

All three models proposed in this dissertation have limitations. RF faces challenges when visualizing SHAP values, resulting in prolonged execution times. The intricacy of the ensemble model, comprised of numerous decision trees, contributes to computational inefficiencies, especially when interpreting and visualizing individual feature contributions. While the Random Forest excels in predictive performance, the inherent complexity poses challenges in the interpretability of the model through SHAP values, impacting the efficiency of real-time visualizations. In comparison, SHAP visualizations for XGBoost and CatBoost appeared more concise and interpretable, while the Random Forest model's SHAP outputs were more complex and harder to analyze. This complexity raises con-

cerns about the interpretability and computational demands of using Random Forest with SHAP, especially in high-stakes applications like credit scoring where clear feature attribution is essential. Addressing this limitation is crucial for enhancing the model's usability and trustworthiness, necessitating future research to optimize the SHAP visualization process for Random Forest models.

While explainability in credit scoring has seen growing attention, several important research avenues remain open. One key area is the need for deeper comparisons between existing interpretability techniques such as SHAP, LIME, and counterfactual approaches to assess which are most suitable for financial applications where clarity and regulatory transparency are critical. Future studies could also focus on embedding financial expertise into explainability tools to make the outputs more meaningful and actionable for practitioners. Additionally, explainability in models dealing with complex or sequential financial data, such as transaction histories, is still relatively underdeveloped and deserves more exploration. There is also value in examining how different users like risk managers or customers understand and respond to explanations, ensuring that interpretability supports decision-making across all levels. Advancing in these areas would strengthen the real-world impact of explainable credit scoring systems and promote trust in machine learning-driven financial services.

Bibliography

[1] Altman, E. I., and Saunders, A. Credit risk measurement: Developments over the last 20 years. *Journal of banking & finance 21*, 11-12 (1997), 1721–1742.

[2] Ariza-Garzón, M. J., Arroyo, J., Caparrini, A., and Segovia-Vargas, M.-J. Explainability of a machine learning granting scoring model in peer-to-peer lending. *Ieee Access 8* (2020), 64873–64890.

[3] Bekhet, H. A., and Eletter, S. F. K. Credit risk assessment model for jordanian commercial banks: Neural scoring approach. *Review of Development Finance 4*, 1 (2014), 20–28.

[4] Biau, G., and Scornet, E. A random forest guided tour. *Test 25* (2016), 197–227.

[5] Breiman, L. Random forests. *Machine learning 45* (2001), 5–32.

[6] Bücker, M., Szepannek, G., Gosiewska, A., and Biecek, P. Transparency, auditability and explainability of machine learning models in credit scoring. *arXiv preprint arXiv:2009.13384* (2020).

[7] Bussmann, N., Giudici, P., Marinelli, D., and Papenbrock, J. Explainable ai in fintech risk management. *Frontiers in Artificial Intelligence 3* (2020), 26.

[8] Chaplinska, A. Evaluation of the borrower's creditworthiness as an important condition for enhancing the effectiveness of lending operations. In *SHS Web of Conferences* (2012), vol. 2, EDP Sciences, p. 00009.

[9] Chen, T., and Guestrin, C. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining* (2016), pp. 785–794.

[10] Cheong, B. C. Transparency and accountability in ai systems: safeguarding well-being in the age of algorithmic decision-making. *Frontiers in Human Dynamics 6* (2024), 1421273.

[11] Chesterman, S. Through a glass, darkly: artificial intelligence and the problem of opacity. *The American Journal of Comparative Law 69*, 2 (2021), 271–294.

[12] Cizmic, J., and Boban, M. Impact of new eu general data protection regulation 2016/679 (gdpr) on the protection of personal data in the republic of croatia. *Zb. Prav. Fak. Sveuc. Rij. 39* (2018), 377.

[13] Coussement, K., Abedin, M. Z., Kraus, M., Maldonado, S., and Topuz, K. Explainable ai for enhanced decision-making, 2024.

[14] Cutler, A., Cutler, D. R., and Stevens, J. R. Random forests. *Ensemble machine learning: Methods and applications* (2012), 157–175.

[15] Dastile, X., Celik, T., and Potsane, M. Statistical and machine learning models in credit scoring: A systematic literature survey. *Applied Soft Computing 91* (2020), 106263.

[16] Dorogush, A. V., Ershov, V., and Gulin, A. Catboost: gradient boosting with categorical features support. *arXiv preprint arXiv:1810.11363* (2018).

[17] Du Toit, H., Schutte, W. D., and Raubenheimer, H. Shapley values as an interpretability technique in credit scoring. *Journal of Risk Model Validation 17*, 4 (2023).

[18] Florez-Lopez, R., and Ramon-Jeronimo, J. M. Enhancing accuracy and interpretability of ensemble strategies in credit risk assessment. a correlated-adjusted decision forest proposal. *Expert Systems with Applications 42*, 13 (2015), 5737–5753.

[19] Friedman, J. H. Greedy function approximation: a gradient boosting machine. *Annals of statistics* (2001), 1189–1232.

[20] Gaiha, R., and Kulkarni, V. S. Credit, microfinance and empowerment. In *Expert Group Meeting: Policies and Strategies to Promote Empowerment of People in Achieving Poverty Eradication, Social Integration and Full Employment and Decent Work for All. United Nations* (2013).

[21] Galli, A., Piscitelli, M. S., Moscato, V., and Capozzoli, A. Bridging the gap between complexity and interpretability of a data analytics-based process for benchmarking energy performance of buildings. *Expert Systems with Applications 206* (2022), 117649.

[22] Gollapudi, S. *Practical machine learning*. Packt Publishing Ltd, 2016.

[23] Hancock, J. T., and Khoshgoftaar, T. M. Catboost for big data: an interdisciplinary review. *Journal of big data 7*, 1 (2020), 94.

[24] Hastie, T., Tibshirani, R., Friedman, J. H., and Friedman, J. H. *The elements of statistical learning: data mining, inference, and prediction*, vol. 2. Springer, 2009.

[25] Li, S.-T., Shiue, W., and Huang, M.-H. The evaluation of consumer loans using support vector machines. *Expert Systems with Applications 30*, 4 (2006), 772–782.

[26] Liaw, A., Wiener, M., et al. Classification and regression by randomforest. *R news 2*, 3 (2002), 18–22.

[27] Liefgreen, A., Weinstein, N., Wachter, S., and Mittelstadt, B. Beyond ideals: why the (medical) ai industry needs to motivate behavioural change in line with fairness and transparency values, and how it can do it. *AI & SOCIETY* (2023), 1–17.

[28] Louppe, G. Understanding random forests: From theory to practice. *arXiv preprint arXiv:1407.7502* (2014).

[29] Lundberg, S. M., and Lee, S.-I. A unified approach to interpreting model predictions. *Advances in neural information processing systems 30* (2017).

[30] Malekipirbazari, M., and Aksakalli, V. Risk assessment in social lending via random forests. *Expert Systems with Applications 42*, 10 (2015), 4621–4631.

[31] Marceau, L., Qiu, L., Vandewiele, N., and Charton, E. A comparison of deep learning performances with other machine learning algorithms on credit scoring unbalanced data. *arXiv preprint arXiv:1907.12363* (2019).

[32] Martens, D., Baesens, B., Van Gestel, T., and Vanthienen, J. Comprehensible credit scoring models using rule extraction from support vector machines. *European journal of operational research 183*, 3 (2007), 1466–1476.

[33] Mester, L. J., et al. What's the point of credit scoring. *Business review 3*, Sep/Oct (1997), 3–16.

[34] Misheva, B. H., Osterrieder, J., Hirsa, A., Kulkarni, O., and Lin, S. F. Explainable ai in credit risk management. *arXiv preprint arXiv:2103.00949* (2021).

[35] Molnar, C. *Interpretable machine learning*. Lulu. com, 2020.

[36] Nielsen, D. Tree boosting with xgboost-why does xgboost win" every" machine learning competition? Master's thesis, NTNU, 2016.

[37] O'Connor, R. A., Nel, J. L., Roux, D. J., Lim-Camacho, L., van Kerkhoff, L., and Leach, J. Principles for evaluating knowledge co-production in natural resource management: Incorporating decision-maker values. *Journal of Environmental Management 249* (2019), 109392.

[38] Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A. V., and Gulin, A. Catboost: unbiased boosting with categorical features. *Advances in neural information processing systems 31* (2018).

[39] Ramteke, A. K., Wadhwa, P., and Yan, M. Interpretability of machine learning versus statistical credit risk models. *Journal of Financial Data Science 4*, 2 (2022).

[40] Ribeiro, M. T., Singh, S., and Guestrin, C. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining* (2016), pp. 1135–1144.

[41] Rudin, C. Please stop explaining black box models for high stakes decisions. *Stat 1050*, 26 (2018), 457.

[42] Salih, A. A perspective on explainable artificial intelligence methods: Shap and lime. *arXiv preprint arXiv:2305.02012* (2023). Accessed: 2024-07-23.

[43] Salvaire, P. A. J. M. Explaining the prdictions of a boosted tree algorithm.

[44] Serrano-Cinca, C., Gutiérrez-Nieto, B., and López-Palacios, L. Determinants of default in p2p lending. *PloS one 10*, 10 (2015), e0139427.

[45] Steinmeyer, K. Xai: Transparency and fairness in ai. *LinkedIn* (2024). Accessed: 2024-07-23.

[46] Stone, M. An asymptotic equivalence of choice of model by cross-validation and akaike's criterion. *Journal of the Royal Statistical Society: Series B (Methodological) 39*, 1 (1977), 44–47.

[47] Tiwari, R. Explainable ai (xai) and its applications in building trust and understanding in ai decision making. *International J. Sci. Res. Eng. Manag 7* (2023), 1–13.

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

[49] Zurada, J. Could decision trees improve the classification accuracy and interpretability of loan granting decisions? In *2010 43rd Hawaii International Conference on System Sciences* (2010), IEEE, pp. 1–9.