NATIONAL TECHNICAL UNIVERSITY OF ATHENS
SCHOOL OF ELECTRICAL AND COMPUTER ENGINEERING

UNIVERSITY OF PIRAEUS
DEPARTMENT OF INDUSTRIAL MANAGEMENT AND TECHNOLOGY

POSTGRADUATE PROGRAM
"ENGINEERING – ECONOMIC SYSTEMS"

Evaluation of Machine Learning Methods for Loan Default Prediction: A Case Study Using Peer-to-Peer Lending Data

MASTER THESIS

Marios Theodoridis

Supervisor: George Matsopoulos, Professor NTUA

Athens, October 2024

[This page is intentionally left blank]

INTERUNIVERSITY POSTGRADUATE PROGRAM
"ENGINEERING – ECONOMIC SYSTEMS"

Evaluation of Machine Learning Methods for Loan Default Prediction: A Case Study Using Peer-to-Peer Lending Data

MASTER THESIS

Marios Theodoridis

Supervisor: George Matsopoulos, Professor NTUA

This postgraduate diploma thesis was approved by the three-member examination committee on October 7, 2024.

1st member	2nd member	3rd member
George Matsopoulos	Athanasios Panagopoulos	Symeon Papavassiliou
Professor,	Professor,	Professor,
School of Electrical and Computer Engineering, NTUA	School of Electrical and Computer Engineering, NTUA	School of Electrical and Computer Engineering, NTUA

Athens, October 2024

……………………………

Marios Theodoridis
Graduate of the Interuniversity Postgraduate Program "Engineering-Economic Systems", School of Electrical and Computer Engineering, National Technical University of Athens

The ideas and conclusions presented in this paper are the author's and do not necessarily reflect the official views of the National Technical University of Athens.

Abstract

This diploma thesis investigates the transformative impact of digital technologies on the financial industry, particularly focusing on the role of data science and machine learning in banking. The study aims to determine how emerging technologies such as blockchain, big data analytics, artificial intelligence (AI), and machine learning (ML) are reshaping financial services. It highlights the fintech revolution and its disruptive influence on traditional banking models, with a particular focus on digital banks, neobanks, and the integration of technology in lending, payments, and investment management.

A comprehensive analysis is provided on advanced methods for credit risk assessment and fraud detection, leveraging machine learning algorithms to enhance predictive accuracy and operational efficiency. A significant portion of the study is dedicated to developing predictive modeling techniques for loan default prediction. Various machine learning algorithms, including logistic regression, decision tree, gradient boosting, random forest, and neural networks, are employed to evaluate and predict loan defaults. The research underscores the importance of model evaluation metrics such as accuracy, precision, recall, and especially F1 score in optimizing model performance.

The findings demonstrate that machine learning, especially models using ensemble learning, can effectively predict loan defaults, thereby aiding financial institutions in mitigating risk and improving decision-making processes. The study concludes with recommendations for future research, including exploring advanced neural network architectures and integrating alternative data sources like social media activity for enhanced predictive power.

Keywords

machine learning, financial technology, banking, loan default prediction, predictive analytics, classification algorithms, supervised learning, exploratory data analysis, peer-to-peer lending

Περίληψη

Η διπλωματική εργασία ερευνά τον αντίκτυπο που έχει η ψηφιακή τεχνολογία στον μετασχηματισμό του χρηματοοικονομικού κλάδου, εστιάζοντας στον ρόλο της επιστήμης των δεδομένων και της μηχανικής μάθησης στον τραπεζικό τομέα. Η μελέτη στοχεύει να αναλύσει πώς αναδυόμενες τεχνολογίες, όπως το blockchain, η ανάλυση δεδομένων, η τεχνητή νοημοσύνη (AI) και η μηχανική μάθηση (ML), αναδιαμορφώνουν τις σύγχρονες χρηματοοικονομικές υπηρεσίες. Περιγράφεται ο τρόπος με τον οποίο η εφαρμογή καινοτομιών στον κλάδο της χρηματοοικονομικής τεχνολογίας επιδρά ριζικά στα παραδοσιακά τραπεζικά μοντέλα και ανατρέπει τον μέχρι τώρα τρόπο λειτουργίας τους. Ιδιαίτερη έμφαση δίνεται στις ψηφιακές τράπεζες και στην ενσωμάτωση τεχνολογικών λύσεων στις υπηρεσίες δανεισμού, πληρωμών και διαχείρισης επενδύσεων.

Πραγματοποιείται εκτενής ανάλυση των προηγμένων μεθόδων και αλγορίθμων μηχανικής μάθησης που εφαρμόζονται στον τραπεζικό τομέα, όπως συμβαίνει για την αξιολόγηση του πιστωτικού κινδύνου και την ανίχνευση απάτης στις συναλλαγές. Ένα σημαντικό μέρος της μελέτης αφιερώνεται στην ανάπτυξη μοντέλων μηχανικής μάθησης για την πρόβλεψη της αθέτησης υποχρεώσεων σε δάνεια από πιθανούς δανειολήπτες. Διάφοροι αλγόριθμοι μηχανικής μάθησης, όπως η λογιστική παλινδρόμηση, το δέντρο απόφασης, οι αλγόριθμοι ενίσχυσης κλίσης, τα τυχαία δάση και τα πολυεπίπεδα νευρωνικά δίκτυα, χρησιμοποιούνται για την αξιολόγηση και πρόβλεψη της αθέτησης υποχρεώσεων της δανειακής σύμβασης από πιθανούς δανειολήπτες. Η έρευνα υπογραμμίζει την σημασία του υπολογισμού διαφορετικών δεικτών για την αξιολόγηση των μοντέλων και την βελτιστοποίηση της απόδοσης τους.

Τα αποτελέσματα δείχνουν ότι η μηχανική μάθηση, και ειδικά τα μοντέλα που χρησιμοποιούν τις λεγόμενες τεχνικές ensemble learning, μπορούν να προβλέψουν αποτελεσματικά τις αθετήσεις δανείων, βοηθώντας με αυτόν τον τρόπο τα χρηματοπιστωτικά ιδρύματα να μειώσουν τον επιχειρηματικό τους κίνδυνο και να βελτιώσουν τις διαδικασίες λήψης αποφάσεων. Η μελέτη καταλήγει σε προτάσεις για μελλοντική έρευνα, συμπεριλαμβανομένης της ανάπτυξης προηγμένων μοντέλων νευρωνικών δικτύων και της ενσωμάτωσης εναλλακτικών πηγών δεδομένων στην ανάλυση.

Λέξεις κλειδιά

μηχανική μάθηση, χρηματοοικονομική τεχνολογία, τραπεζικός τομέας, πρόβλεψη αθέτησης δανείων, προγνωστική ανάλυση, αλγόριθμοι ταξινόμησης, επιβλεπόμενη μάθηση, διερευνητική ανάλυση δεδομένων, δανεισμός

Acknowledgements

I would like to express my sincere gratitude to everyone who supported me throughout this master's journey. Firstly, I am deeply thankful to my supervisor and professor, George Matsopoulos, for giving me the opportunity to work on this fascinating topic, which significantly enriched my academic experience. His invaluable guidance, feedback, and support were crucial throughout my master's thesis.

A heartfelt thanks to my family and friends for their constant encouragement, understanding, and patience. Their emotional support provided me with the strength and motivation needed to persevere, and it was instrumental in the completion of this thesis.

Εκτεταμένη Περίληψη

Η διπλωματική εργασία ερευνά τον αντίκτυπο που έχει η ψηφιακή τεχνολογία στον μετασχηματισμό του χρηματοοικονομικού κλάδου, εστιάζοντας στον ρόλο της επιστήμης των δεδομένων και της μηχανικής μάθησης στον τραπεζικό τομέα. Η μελέτη αναλύει πώς αναδυόμενες τεχνολογίες, όπως το blockchain, η ανάλυση δεδομένων, η τεχνητή νοημοσύνη (AI) και η μηχανική μάθηση (ML), αναδιαμορφώνουν τις σύγχρονες χρηματοοικονομικές υπηρεσίες. Στο εισαγωγικό κεφάλαιο περιγράφεται ο τρόπος με τον οποίο η εφαρμογή καινοτομιών στον κλάδο της χρηματοοικονομικής τεχνολογίας (fintech) επιδρά ριζικά στα παραδοσιακά τραπεζικά μοντέλα και ανατρέπει τον μέχρι τώρα τρόπο λειτουργίας τους. Ιδιαίτερη έμφαση δίνεται στις ψηφιακές τράπεζες και στην ενσωμάτωση τεχνολογικών λύσεων στις υπηρεσίες δανεισμού, πληρωμών και διαχείρισης επενδύσεων.

Στο τρίτο κεφάλαιο, πραγματοποιείται εκτενής ανάλυση των προηγμένων μεθόδων και αλγορίθμων μηχανικής μάθησης που εφαρμόζονται στον τραπεζικό τομέα, όπως συμβαίνει για την αξιολόγηση του πιστωτικού κινδύνου και την ανίχνευση απάτης στις συναλλαγές. Ένα σημαντικό μέρος της μελέτης, αφιερώνεται στην περιγραφή και ανάπτυξη μοντέλων μηχανικής μάθησης για την πρόβλεψη της αθέτησης υποχρεώσεων σε δάνεια από πιθανούς δανειολήπτες. Στο τέταρτο κεφάλαιο, περιγράφονται εκτενώς διάφοροι αλγόριθμοι μηχανικής μάθησης, όπως η λογιστική παλινδρόμηση (logistic regression), το δέντρο απόφασης (decision tree), το μοντέλο ενίσχυσης κλίσης (gradient boosting machines), το τυχαίο δάσος (random forest) και τα πολυεπίπεδα νευρωνικά δίκτυα (neural networks), που είναι κατάλληλοι για την αξιολόγηση και πρόβλεψη της αθέτησης υποχρεώσεων της δανειακής σύμβασης από πιθανούς δανειολήπτες. Η έρευνα υπογραμμίζει την σημασία του υπολογισμού διαφορετικών δεικτών για την αξιολόγηση των μοντέλων και την βελτιστοποίηση της απόδοσης τους.

Αναλυτικότερα, το σύνολο των δεδομένων που αναλύεται στο πέμπτο κεφάλαιο αποτελείται από στοιχεία για δάνεια που εκδόθηκαν από το 2007 έως το 2018 από πλατφόρμα δανεισμού Peer-To-Peer (P2P). Η διαδικτυακή αυτή πλατφόρμα, συνδέει απευθείας τους δανειολήπτες με τους δανειστές, παρακάμπτοντας τα παραδοσιακά χρηματοπιστωτικά ιδρύματα. Η μεθοδολογία ανάπτυξης του μοντέλου ξεκινάει με την επιλογή κατάλληλων μεταβλητών από το σύνολο των δεδομένων που σχετίζονται με το δάνειο, τον δανειολήπτη και τα στοιχεία του λογαριασμού του, καθώς και την τρέχουσα κατάσταση του δανείου. Το ποσό δανείου, το επιτόκιο, ο λόγος χρέους προς εισόδημα, ο αριθμός των ανοιχτών τραπεζικών λογαριασμών και η διάρκεια

επαγγελματικής απασχόλησης, θεωρούνται σημαντικοί παράγοντες πρόβλεψης της αθέτησης δανείου και εντάσσονται στην διαδικασία εκπαίδευσης του μοντέλου. Στην συνέχεια, μέσω διερευνητικής ανάλυσης δεδομένων, πραγματοποιείται επεξεργασία των διαθέσιμων δεδομένων, διαχείριση των κενών και των ακραίων τιμών και εφαρμόζονται κατάλληλες τεχνικές κωδικοποίησης, με σκοπό την διευκόλυνση της ανάπτυξης και την βελτίωση της προβλεπτικής ικανότητας του μοντέλου. Η διαδικασία διαχείρισης των δεδομένων διασφαλίζει ότι τα μοντέλα εκπαιδεύονται σε κατάλληλα παρελθοντικά δεδομένα, ώστε μελλοντικά να προβλέπουν με ακρίβεια τις περιπτώσεις μη έγκαιρης αποπληρωμής δανείου. Έτσι, οι δανειολήπτες και τα χρηματοπιστωτικά ιδρύματα λαμβάνουν ενημερωμένες αποφάσεις σχετικά με την έγκριση ή απόρριψη μίας αίτησης δανείου και το ύψος του επιτοκίου δανεισμού.

Τα διάφορα μοντέλα μηχανικής μάθησης κατατάσσονται κατά φθίνουσα σειρά, πρωτίστως με βάση τον δείκτη F1 Score και δευτερευόντως με βάση τον δείκτη Recall. Το μοντέλο του τυχαίου δάσους (random forest) επικράτησε, εμφανίζοντας την υψηλότερη βαθμολογία στον δείκτη F1, υποδηλώνοντας την αποτελεσματικότητά του στο να ανιχνεύει με ακρίβεια την αθέτηση δανείων. Το μοντέλο ενίσχυσης κλίσης (gradient boosting), παρουσίασε αντίστοιχα καλή προβλεπτική ικανότητα στον προσδιορισμό της αθέτησης δανείων. Στην κατάταξη ακολουθούν με φθίνουσα σειρά απόδοσης ως προς τον δείκτη F1, η λογιστική παλινδρόμηση (logistic regression), τα πολυεπίπεδα νευρωνικά δίκτυα (multi-layer perceptron neural networks) και το μοντέλο του δέντρου απόφασης (decision tree). Συμπερασματικά, τα αποτελέσματα δείχνουν ότι η μηχανική μάθηση και ειδικά τα μοντέλα που χρησιμοποιούν τις λεγόμενες τεχνικές ensemble learning (random forest και gradient boosting machines), μπορούν να προβλέψουν αποτελεσματικά τις αθετήσεις δανείων, βοηθώντας με αυτόν τον τρόπο τα χρηματοπιστωτικά ιδρύματα να μειώσουν τον επιχειρηματικό τους κίνδυνο και να βελτιώσουν τις διαδικασίες λήψης αποφάσεων. Η μελέτη καταλήγει στο έβδομο κεφάλαιο, με προτάσεις για μελλοντική έρευνα, συμπεριλαμβανομένης της ανάπτυξης προηγμένων μοντέλων νευρωνικών δικτύων και της ενσωμάτωσης εναλλακτικών πηγών δεδομένων στην ανάλυση.

Table of Contents

1. Software Packages

1.1 Python

Python is a well-known programming language, that can be implemented in a vast number of applications, domains, and projects. Python is categorized into the interpreted programming languages type, because commands are executed without the need for previous compilation. Consequently, it can be easily adapted to any new requirements or changes, enabling rapid and straightforward software development processes and approaches. Multiple software engineering paradigms can be built in python, such as object-oriented and procedural programming. This study concentrates on the applications of data science and machine learning within financial technology. In this context, Python's main data science libraries such as Pandas, NumPy (Numerical Python), SciPy (Scientific Python), and Matplotlib are employed for data manipulation and visualization, numerical calculations and the application of machine learning models or neural networks [1].

1.2 Pandas

Pandas is a python library utilized for importing and analyzing structured datasets in various formats. The Dataframe, a two-dimensional table comprising rows (records) and columns (features) constitutes the fundamental data structure in pandas, facilitating the manipulation of data in a tabular format. The key operations performed with pandas include importing or exporting files, handling missing data, data inspection, reshaping, editing, and plotting [2]. In the subsequent chapters, the pandas library is employed for conducting exploratory data analysis and preprocessing data for machine learning models.

1.3 NumPy

NumPy is a remarkable high-level package developed to enhance mathematical operations and numerical calculations within Python. Its core data structure, the nd array (n-dimensional array), facilitates the storage of diverse numerical data in multi-dimensional arrays, enabling the execution of sophisticated scientific computations. Key functionalities of NumPy encompass comprehensive functions for fundamental arithmetic operations and robust statistical computations, highlighting its use in computing applications and scientific research [3].

1.4 SciPy

SciPy expands on NumPy's foundational capabilities by offering a variety of additional modules and functionalities specifically tailored for advanced engineering and mathematical computations. SciPy goes beyond elementary array operations and basic linear algebra by providing functionalities such as solving differential equations, optimizing algorithms and implementing specialized numerical functions.

1.5 Scikit-learn

A prominent machine learning library in Python, Scikit-learn was designed to enable the widespread adoption and straightforward implementation of both supervised and unsupervised learning techniques. It offers an extensive collection of machine learning models that can be instantiated and trained using its comprehensive suite of functionalities. In addition, the library supports customization of default parameters and provides extensive tools for evaluating performance across different models. Essential components include feature and model selection methods, as well as hyperparameter tuning techniques, which are crucial aspects enhancing its utility and appeal among data scientists and researchers.

1.6 Matplotlib & Seaborn

Matplotlib plotting package in Python offers diverse capabilities for creating a wide range of plots and conducting data visualization tasks during data exploration. Similarly, Seaborn, which is built on top of Matplotlib, strengthens these capabilities by providing more attractive and detailed graphics. It focuses on high-level aesthetics and the extraction of actionable insights and meaningful information, making it valuable for both exploratory data analysis and statistical analysis. Comprehending relationships within data and effectively communicating results are critical tasks, rendering the aforementioned visualization libraries indispensable throughout the data science pipeline.

1.7 Jupyter Notebook

Jupyter Notebook offers a web-based integrated and interactive computational environment for developing, executing code and viewing the results. This computational notebook, accessible via a web application, features multiple input cells where users can write and execute code (e.g., Python) either collectively or independently across different cells. The outcome of each input cell is incorporated into the notebook and displayed in the corresponding output cell. Following this structured way, the readability and comprehension of the executed code is enhanced. In the field of data science, Jupyter platform is broadly adopted for its facilitation of fundamental software engineering practices throughout the entire data science pipeline. Its flexibility allows for quick adaptation to changes, effortless code re-execution, and seamless code sharing with stakeholders [4]. In this study, Jupyter serves as the primary tool for conducting tasks ranging from data preprocessing to model training and evaluation.

2. Introduction

2.1 Digital Transformation in the Financial industry

Technology has been rapidly integrated into all aspects of human life, affecting both daily activities and the corporate environment across various industries throughout the previous few decades. This integration has led to disruptive changes. Technological advancements have reshaped the technical and institutional environments in which businesses function. We are progressively observing shifts in business processes and models, alongside the development of novel methods for recording, updating, validating, and sharing digital information [5].

The process of leveraging contemporary digital technologies to significantly improve business processes and minimize operational costs is known as digital transformation. This is a continuous process, and to remain viable and competitive, it must be adopted by most companies. Traditional financial institutions are no exception to this transformation [6]. The evolution of technological innovation and emerging trends in financial services are impacting business processes within the financial industry and create opportunities for a more efficient and resilient financial system, as described in the following paragraphs.

2.2 Role of New Technologies in Financial services

Emerging technologies such as mobile technologies, internet of things (IoT), cloud computing, blockchain technology, big data analytics, artificial intelligence (AI), and machine learning (ML) are disrupting the financial environment and creating new opportunities for enhancing workflows, automating and streamlining processes. Financial technology companies (fintechs) and regulatory technology companies (regtechs), leveraging innovative technology and products, are significantly influencing the financial ecosystem by advancing digitalization of services and introducing novel business models and products. This technological revolution is benefiting a wide spectrum of financial institutions beyond commercial and investment banks, including insurance companies, asset management firms, brokerage firms, hedge funds, and private equity firms. These institutions actively engage in providing a range of financial services regarding transaction processing, investment management, saving accounts, financial management, lending, portfolio management, and risk management [6]. Each of these services can be transformed through the application of technological advancements, thereby fostering opportunities for global financial innovation, new product offerings and enhanced customer experience.

2.3 Fintech Revolutionizing the Banking industry

2.3.1 Historical overview of Digital Transformation in Banking

Banking constitutes a substantial segment of the financial services sector, traditionally defined by physical branches and manual operations. However, the advent of digital technology has transformed the delivery of banking services. The initial phase of the Fintech revolution, spanning from the mid-20th century to the early 21st century, marked the introduction of ATMs, online banking services, electronic payments, and clearing systems. These innovations enabled customers to access financial services through online platforms and mobile banking applications, significantly enhancing convenience and accessibility. Internet banking facilitates 24/7 functionalities such as checking account balances, bill payments and money transfers. Similarly, mobile payment solutions have become prevalent for conducting transactions [7]. These advancements indicate just the beginning of the widespread adoption and development of digital tools in banking, marking the onset of a significant revolution, poised to integrate banking innovation and offer non-intermediate services to customers in the years ahead.

2.3.2 Disruption of Traditional Banking models through Fintech

Fintech, abbreviation for Financial Technology, is defined as the integration of technology with finance. This field encompasses innovations that empower non-financial institutions, including fintech start-ups and tech companies, to offer financial services. These innovations influence various financial sectors. Revolutionary advancements are evident in lending, with the advent of platforms simplifying loan application and approval process, and new lending models. Similarly, in the payments sector, traditional methods are altered through digital currencies and digital wallets using fiat currency for mobile payments. Furthermore, investment and wealth management have seen innovations such as real-time trading, along with the utilization of robo-advisors for portfolio management and diversification. The insurance industry has also experienced transformation with the rise of insurtech companies that provide individualized solutions and leverage big data analytics for enhanced risk-based pricing models [5]. The subsequent sections offer a comprehensive overview of the innovations in lending, payment, and investment/wealth management.

2.3.2.1 Lending

Lending is undergoing significant transformations through fintech advancements that simplify the loan application and assessment processes. These innovations facilitate lending decisions and introduce new methods for rapid, flexible, and affordable lending via online lending services and platforms.

Platforms Accelerating Loan Application and Approval Processes

Technological advancements in banking have enabled the development of user-friendly online interfaces for loan applications. Through these online platforms, the loan application process is streamlined and accelerated, providing applicants with instant approval or rejection decisions. These platforms benefit both lenders and borrowers. Lenders can make accurate lending decisions based on real-time data, while small companies and individuals can quickly receive capital without the bureaucracy of traditional banks [5]. This efficiency bypasses traditional banking constraints, offering a more efficient and responsive lending experience.

Peer-to-peer (P2P) lending

Innovative P2P lending platforms are significantly affecting the future of banking, by challenging traditional revenue models and the banks’ role in the lending process. New intermediaries have emerged in the lending landscape through P2P lending platforms, such as Lending Club, providing capital access to entities otherwise constrained from borrowing through conventional banking institutions.

These platforms establish direct connections between potential borrowers and a diverse array of lenders, including non-conventional financial institutions and investors such as hedge funds. Thus, they enhance lending flexibility, eliminate intermediary costs, and facilitate the distribution of funds from multiple sources to diverse borrowers. Additionally, these platforms employ machine learning algorithms, including loan default prediction models, to evaluate loans online, thereby expediting lending decisions [5]. This technological approach results in lower and more attractive interest rates for the borrowers and provides lenders with higher returns.

Crowdfunding & Initial Coin Offerings (ICOs)

Apart from the development of P2P lending platforms, there is an emerging trend in the use of digital financing platforms for lending both fiat and digital currencies. Crowdfunding is a notable example, where entrepreneurs and organizations can remotely raise capital through online platforms from a diverse group of individuals, who receive equity stakes or product services in return. Kickstarter is a prominent digital platform facilitating this type of financing. Additionally, Kiva exemplifies a crowdfunding platform that provides small, short-term loans, known as microloans, enabling entrepreneurs, even from remote areas, to borrow small amounts of money from individuals as a form of donation or investment. This innovation allows startups to secure initial funding for growth and expansion, even if they lack the financial establishment and collateral required as warranty for the loan repayment in traditional loans [5].

By eliminating the need for physical presence in bank branches, these lending platforms significantly transform the banking industry, facilitating mainly the provision of unsecured loans to businesses and individuals who face constraints in accessing traditional lending. Similarly, Initial Coin Offerings (ICOs) represent the cryptocurrency version of crowdfunding, enabling startups to access external financing by issuing digital tokens. These tokens, often in the form of cryptocurrencies, can be used by potential investors to purchase services, further enhancing the flexibility and reach of digital financing.

2.3.2.2 Payments

Fintech innovations in payments hold significant disruptive potential and constitute a substantial segment of the fintech industry. Key advancements in this domain include the proliferation of mobile payment applications, the widespread adoption of internet-based payments (e-payments), and the introduction of contactless payment methods. These innovations enhance the speed, efficiency, and security of both person-to-person and person-to-merchant transactions. Moreover, the advent of cryptocurrencies and digital currencies has facilitated the implementation of decentralized payment systems and the development of decentralized finance (DeFi) platforms. Digital wallets that support fiat currencies enable electronic transactions and peer-to-peer (P2P) money transfers, further transforming the financial landscape.

Internet-Based Payments (e-Payments)

With the rapid expansion of e-commerce, online payments and mobile banking have become increasingly prevalent. Customers can utilize mobile banking applications provided by their banks to conduct payments and purchases directly through their smartphones. Additionally, various platforms linked to their bank accounts facilitate secure and convenient money transfers. E-commerce providers, such as Amazon, have also developed proprietary applications to ensure secure transactions, thereby enhancing the online shopping experience for their users.

Digital Wallets with Fiat currency

Digital wallets have substantially transformed the payments landscape by allowing consumers to conduct transactions electronically. These wallets facilitate peer-to-peer (P2P) fiat currency transactions and support a variety of payment methods, such as bank transfers and credit card payments. PayPal, the global leader in digital wallet services, processed an annual mobile payment volume of $227 billion in 2018 and continues to grow, with over 400 million active users and support for 25 currencies across more than 200 countries with data as of August 2021 [5]. These digital wallets act as secure bridges between merchants and customers, encrypting payment information and facilitating the transfer of funds through approved payment processes.

Decentralized Payments and Digital Currencies

A notable fintech development in the payments sector is the expansion of numerous digital assets, including cryptocurrencies and digital currencies. These assets, issued in their own denominations, challenge traditional payment methods. However, unlike traditional currencies, they are distinct due to the lack of redeemability. Cryptocurrencies such as Bitcoin, Ethereum, and Ripple utilize distributed ledger technology, specifically blockchain, to simplify peer-to-peer electronic cash transactions. Blockchain technology is used by decentralized payment systems to ensure safe and transparent transactions without the need for intermediaries within the financial system. Bitcoin, for instance, enables direct online payments between individuals, circumventing financial institutions. These digital currencies provide low transaction costs and convenience, positioning them as leading alternatives to conventional fiat currencies. Additionally, certain fintech companies have launched Bitcoin payment services, enabling customers to perform swift and secure transactions [5], [8].

2.3.2.3 Investment & Wealth Management

Technological advancements are disrupting the investment and wealth management domains by enabling the provision of automated advisory services to assist clients. Firstly, technologies such as artificial intelligence (AI) and machine learning (ML) are leveraged to develop sophisticated investment management services and products. For example, robo-advisors, with their ability to preprocess and analyze high-volume and high-velocity market data, utilize advanced machine learning models to provide investment recommendations, customize and diversify portfolios, optimize the distribution of assets, and continuously monitor portfolio performance and mitigate associated risks, based on investors' individual financial targets and risk tolerance. Real-time online trading represents another domain where robo-advisors leverage data science algorithms to provide insights and recommendations. By continuously analyzing time-series data to identify trends in the online stock market and applying advanced forecasting methods, robo-advisors make predictions to inform investment decisions and execute buy or sell orders automatically. Additionally, wealth management is enhanced through digital tools, supporting comprehensive financial and tax planning. These technological advancements in investment and wealth management create opportunities for businesses and individuals to achieve their financial objectives at minimized costs, as these services are typically more affordable due to the reduced involvement of human advisors [5].

2.3.3 Digital Banks & Neobanks

With the ongoing digital transformation of the banking industry, an increasing number of services are being provided online. Consequently, traditional banks are embracing this change by restructuring their organizational frameworks and updating their technological infrastructure to support the delivery of digitalized services. Digital banks have emerged from the digitalization of conventional banking and financial institutions, extending the provision of traditional banking services to the online realm. These digital banks benefit from the established security, licensing, and trust associated with traditional banks, yet they still maintain physical branches. In contrast, neobanks are primarily technology startups that are not typically created or supported by conventional banks [8]. This independence allows neobanks to develop an online cross-border presence. Their objective is to offer innovative, specialized products with diverse features at lower or no cost due to the absence of conventional operational expenses, thereby outperforming the traditional services provided by digital banks [9]. Although neobanks must adhere to regulatory compliance and sometimes face skepticism, they possess significant potential. A notable example of their success is Revolut, a financial technology company which allows customers to hold multi-currency accounts and perform international money transfers at a low cost via a mobile application.

3. Data Science and Machine Learning in Banking

3.1 Credit Risk Assessment in Lending

Through the pervasive digital transformation across all industries and particularly in banking, coupled with the widespread adoption of information and communication technology, a broad range of data points is now being systematically collected, stored and readily available for processing. The interconnectedness of information systems facilitates access to extensive historical personal and transactional data from diverse sources. Over time, the accumulation of these data points has significantly enhanced the efficient assessment of an applicant's creditworthiness. Alternative data sources can now be leveraged, including borrower's spending patterns and volume of transactions, educational background, employment status, residential details, bill payment records or even social media engagement, in addition to traditional metrics such as credit score and financial situation. This broader spectrum enables a holistic evaluation of each applicant, providing lenders with more complete insights for making informed lending decisions. Such practices help mitigate the loan default risk and increase economic activity.

3.1.1 Advanced methods for evaluating Credit Risk

Both statistical and machine learning techniques are utilized for credit risk assessment. For instance, a widely used method in both academic literature and practical applications is logistic regression, which evaluates attributes like credit rating, debt-to-income ratio, transaction patterns, and consumer behavior to forecast the likelihood of loan default. Furthermore, another effective approach is the decision tree method, which segments loan applicants based on various data points to assess the risk of defaulting in loan obligations. Additionally, ensemble techniques such as random forests and gradient boosting machines combine the results of multiple decision trees and lead to better predictions. Moreover, advanced algorithms like neural networks demonstrate strong performance and find widespread application in the real-world, despite their substantial computational demands [10]. Therefore, potential borrowers are efficiently classified as either defaulters or non-defaulters, aiding banks and other lending organizations in making decisions regarding lending, capital allocation and interest rates.

3.2 Fraud Detection in Payments

Despite the implementation of proactive safety measures designed to authenticate customers and verify transactions, security issues remain unresolved. With the widespread adoption of online transactions and e-payments, the incidence of fraudulent activities has increased, leading to financial losses and reputational damage. These illegal activities occur by circumventing the authentication and security processes established by financial institutions. The primary objective of such malicious behavior is to steal money and generate financial profit, either through the unauthorized use of credit card details for purchases or by gaining access to bank accounts for illicit withdrawals or transfers.

This issue is amplified by the presence of customers on the internet, particularly when using e-commerce platforms and email interfaces. Through phishing emails, fake online banking applications, or commercial websites that mimic legitimate ones, individuals can be deceived into downloading malware or providing personal information to conduct payments or log into their banking accounts [11]. Consequently, it is imperative for banking and fintech organizations to implement robust fraud prevention and detection models to ensure the security of financial transactions for their customers.

3.2.1 Advanced approaches for Detecting Fraud in Payments

In the same context as loan default prediction, fraud detection involves categorizing transactions as either fraudulent or legitimate. This task in supervised machine learning is known as binary classification, where the goal is to classify financial transactions into one of two distinct classes. Therefore, similar classification algorithms, such as logistic regression, decision trees, random forests, support vector machines, gradient boosting machines, and neural networks, can be employed to detect suspicious transactions, akin to classifying a loan as default. These models are primarily trained on datasets comprising historical transaction data, encompassing customer attributes (e.g., customer ID and demographics), customer behavior (e.g., login frequency, device, and location), spending patterns, and transactional attributes (e.g., transaction time, location, amount, type, and periodicity). For these machine learning models to effectively classify new instances, labeled historical data are required during the training phase.

Patterns that are irregular and inconsistent may also imply fraudulent activity. Anomaly detection algorithms identify significant deviations from the customer's normal transactional behavior, enabling the detection of fraud through the presence of outliers in real-time transactional data. These algorithms do not require labeled data and can identify fraud without prior knowledge of the characteristics of fraudulent behavior, hence they are categorized as unsupervised learning. For instance, a significant kilometric discrepancy between the customer’s declared location and the transaction’s geolocation will be flagged by an anomaly detection algorithm as potential fraud. Similarly, factors such as unusually high transaction amounts, increased transaction frequency, and irregular transactions occurring late at night (during typical sleeping hours) are common indicators used to mark a transaction as fraudulent and potentially revoke it [11].

3.3 Investment & Wealth Management solutions

In the financial ecosystem, both historical and real-time data present opportunities for developing technological solutions aimed at predicting future indicators and outcomes. Firstly, the accumulation of historical numerical and time-series data, typically stored in databases or cloud environments, facilitates the training and development of various forecasting algorithms. Such data includes historical market information such as stock prices, trends, trading volumes, and a range of fundamental economic indicators (e.g., gross domestic product, inflation rate, interest rates), alongside technical indicators like moving averages, and company-specific metrics, such as earnings per share, price-earnings ratio and revenue growth. These algorithms, including supervised machine learning and time-series regression models, facilitate effective portfolio management and investment allocation, thereby reducing exposure to risks and meeting the investors' investment goals.

Moreover, the integration of real-time data, including intraday market updates, financial news, economic announcements affecting key economic indicators, social media sentiment scores, and other relevant metrics, facilitates accurate short-term predictions. This capability is crucial for algorithmic trading strategies designed to generate profit from short-term price movements in the stock market, enhancing capabilities in intraday trading and high-frequency trading. Advanced robo-advisors leverage automation to combine algorithmic trading strategies with automated investment advisory services, further optimizing decision-making processes in investment management [12].

3.3.1 Algorithmic approaches for Investment Management

As discussed in the previous section, machine learning algorithms for regression including linear regression, random forest regression, gradient boosting regression, support vector regression, and neural networks, can be leveraged for predicting stock prices and guiding long-term investment or trading decisions. Similarly, time-series regression models such as autoregressive integrated moving average (ARIMA) and long short-term memory networks (LSTM) are employed to discern trends in the online stock exchange and forecast future prices or returns based on historical time-series data. Robo-advisors implement advanced machine learning models like decision trees, random forests, and neural networks to analyze intricate complex data, detect patterns, and provide investment recommendations, thus improving the efficiency of investment and wealth management [12].

4. Predictive Modeling Techniques in Loan Default Prediction

Machine learning methods are primarily classified into supervised and unsupervised learning. Supervised learning involves algorithms that develop predictive models using both input features and corresponding output labels, which can be further divided into classification and regression techniques. In contrast, unsupervised learning algorithms, such as clustering and association, organize and interpret data solely based on input features without pre-existing labels. The key distinction lies in the use of labeled data in supervised learning, enabling the model to learn the relationship between inputs and outputs.

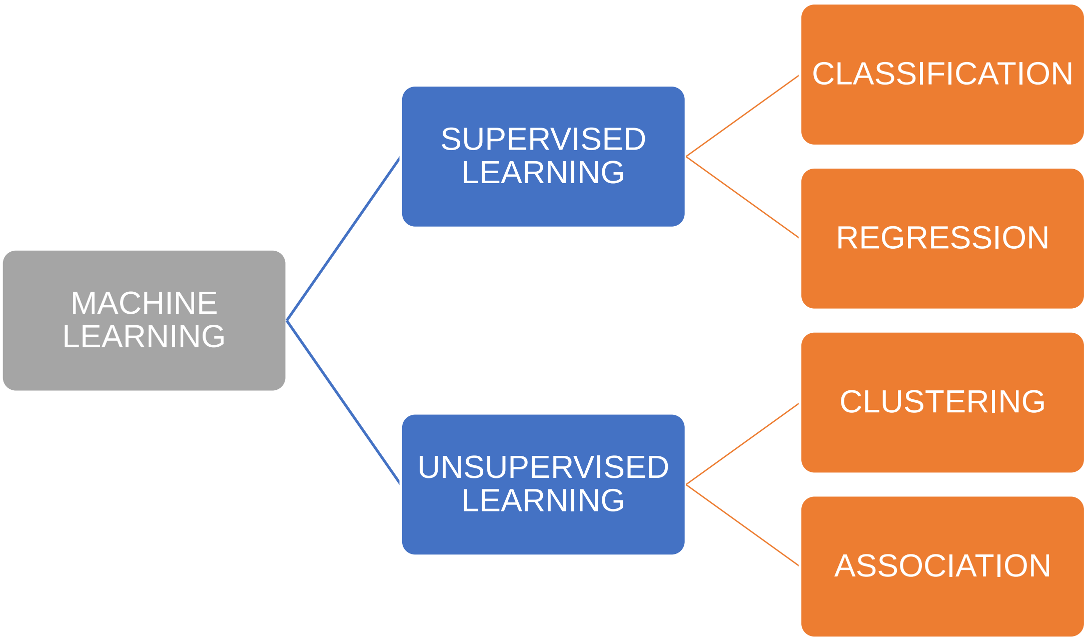


Figure 1. Supervised and unsupervised machine learning

4.1 Supervised Learning Algorithms

In supervised learning, the algorithm is trained using a labeled dataset, where each data point is associated with a correct response or classification. The input data are linked with their corresponding output data. Through this training process, often referred to as fitting, the model learns the relationship between the input variables (features or 'x variables') and the output variable (target or 'y variable'). This learned relationship enables the model to predict the correct output for new, unlabeled input data. Supervised learning algorithms can be categorized into two types: classification and regression.

4.1.1 Classification

Classification algorithms are designed to predict discrete outcomes. When the target variable is limited to two possible categories (such as yes/no or default/no default), it is referred to as binary classification. Conversely, if the target variable can have more than two categories, it is known as multiclass classification. During training, the model uses labeled data to learn and subsequently predict the category, class, or outcome of new, unseen instances based on their features. Some widely used machine learning algorithms for classification include logistic regression, decision tree classifier, random forest classifier, support vector machine (SVM), k-nearest neighbor (KNN) classifier, and naïve Bayes.

4.1.1.1 Logistic Regression

Logistic regression is a statistical method typically employed for binary classification tasks, where the dependent variable is dichotomous, meaning it can only take one of two possible values (e.g., yes/no, default/no default). The logistic function, also known as the sigmoid function, produces an output between 0 (negative infinity) and 1 (positive infinity), which makes it particularly suitable for binary categorization problems. This function is characterized by an S-shaped curve that transforms any real-valued number into a probability value within this range. When the output of the logistic function exceeds a specified default threshold (typically 0.5), the instance is classified into the positive class (e.g., default). Conversely, if the output is below this threshold, the instance is classified into the negative class (e.g., no default). Consequently, logistic regression estimates the likelihood of a binary event occurring [13].

The probability of the positive class, denoted as $P(y = 1 \mid X)$, is modeled using the logistic function (or sigmoid function). The general form of the probability equation for logistic regression is:

$$P(y = 1|X) = \frac{1}{1 + e^{-(\beta_0+\beta_1 X_1+\beta_2 X_2+\cdots+\beta_n X_n)}}$$

where:

- $P(y = 1 \mid X)$ is the probability that the outcome is 1 (e.g., loan default)
- y is the independent variable
- β_0 is the intercept (bias term)
- $\beta_1, \beta_2, \ldots, \beta_n$ are the regression coefficients (weights) for the input features
- $X_1, X_2, \ldots, X_n$ are the input features (e.g., credit score, loan amount, debt-to-income ratio)
- e is the base of the natural logarithm, approximately equal to 2.718

The Sigmoid function $\sigma(z)$ is defined as:

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

The method can be used to assess the data of each applicant and generate a probability score indicating the likelihood of default. Applicants who receive high probability scores (e.g., exceeding 0.5) are either declined the loan or subjected to further evaluation, whereas those with lower probability scores are expedited through the approval process.

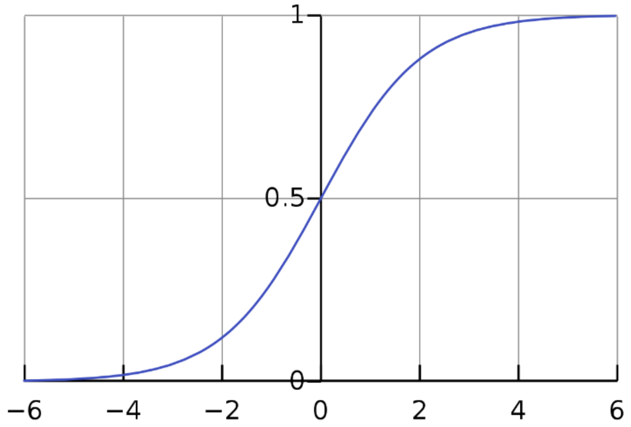


Figure 2. Sigmoid function

4.1.1.2 Decision Tree Classifier

The Decision Tree Classifier is a method employed to predict categorical target variables. The algorithm traverses the tree from the root node to a leaf node that corresponds to the features of a new data point, determining the predicted class as the majority class within that leaf node. Generally, decision trees are applicable to both classification and regression tasks. The decision tree is structured as a flowchart with a root node (starting point) representing the complete dataset, branches representing decision rules, internal decision nodes representing choices based on input features or attributes, and terminal leaf nodes representing outcomes (class labels or numerical values).

The Classification and Regression Tree (CART) algorithm is utilized to construct the decision tree, containing both categorical and numerical data. The construction process begins with feature selection, which is performed using Attribute Selection Measures (ASM) to suitably split the records. The objective of ASM is to identify the most significant attributes for prediction by evaluating the homogeneity of the data subsets post-split (impurity metric). Consequently, the top nodes in the tree are the most influential features. The selected feature becomes a decision or internal node, and the dataset is partitioned into smaller subsets. This recursive partitioning process continues on each derived subset until no further classification is possible, completing the tree construction [14].

To enhance the performance of decision trees and mitigate overfitting on the training set, several practices are commonly adopted. One such practice is limiting the depth of the tree, which refers to the maximum distance between the root and any leaf. Another practice is setting a minimum threshold for the number of samples in a leaf node, ensuring that no leaf contains fewer samples than this threshold.

In Decision Tree Classifier, mathematical concepts such as Information Gain, Entropy and Gini Index play critical roles in determining the best features for splitting the data at each node. Below is an explanation of these concepts with their corresponding mathematical equations.

Entropy

Entropy $\mathrm{H}(S)$ is a measure of impurity or uncertainty in the data. It measures the uncertainty in predicting the class label of a randomly selected instance. The Entropy of a dataset S with a set of classes (for instance, binary classification with two classes) is defined as:

$$\mathrm{H(S)} = -\sum_{\mathrm{i}=1}^{\mathrm{k}} \mathrm{p_i} \log_2(\mathrm{p_i})$$

where:

- $\mathrm{H}(S)$ is the Entropy of the dataset and ranges between 0 (pure dataset) and 1 (maximum impurity)
- k is the number of distinct classes in the dataset
- $\mathrm{p_i}$ is the probability of class i in the dataset

Information Gain

Information Gain $\mathrm{IG}(S{,}A)$ measures the reduction in entropy after splitting a dataset S on an attribute A. The attribute that results in the highest Information Gain is chosen for splitting. It is mathematically defined as:

$$\mathrm{IG(S,A)} = \mathrm{H(S)} - \sum_{\upsilon \in \mathrm{V(A)}} \frac{|\mathrm{S}_\upsilon|}{|\mathrm{S}|} \mathrm{H}(\mathrm{S}_\upsilon)$$

where:

- $\mathrm{IG}(S{,}A)$ is the Information Gain of attribute A
- $\mathrm{H}(S)$ is the entropy of the original dataset S
- $\mathrm{V}(A)$ is the set of possible values for attribute A
- S_υ is the subset of S for which attribute $A=\upsilon$
- $\mathrm{H}(\mathrm{S}_\upsilon)$ is the entropy of the subset S_υ

Gini Index

The Gini Index is another measure of impurity used for splitting in decision trees, particularly in classification tasks. It measures the probability that a randomly chosen element from the dataset would be incorrectly classified. A lower Gini Index indicates a better split. Unlike entropy, which uses logarithms, the Gini Index uses squared probabilities. The Gini Index for a dataset S is defined as:

$$\mathrm{Gini(S)} = 1 - \sum_{\mathrm{i}=1}^{\mathrm{k}} \mathrm{p_i^2}$$

where:

- $\mathrm{p_i}$ is the probability of class i in the dataset
- k is the number of distinct classes

Decision Tree Classifiers are extensively utilized in the domain of loan default prediction and credit risk assessment. By evaluating a range of borrower characteristics and financial metrics, these models enable financial institutions to make well-informed lending decisions. Specifically, the decision tree algorithm examines features such as credit score, income, and loan amount to determine whether a loan application should be approved or rejected.

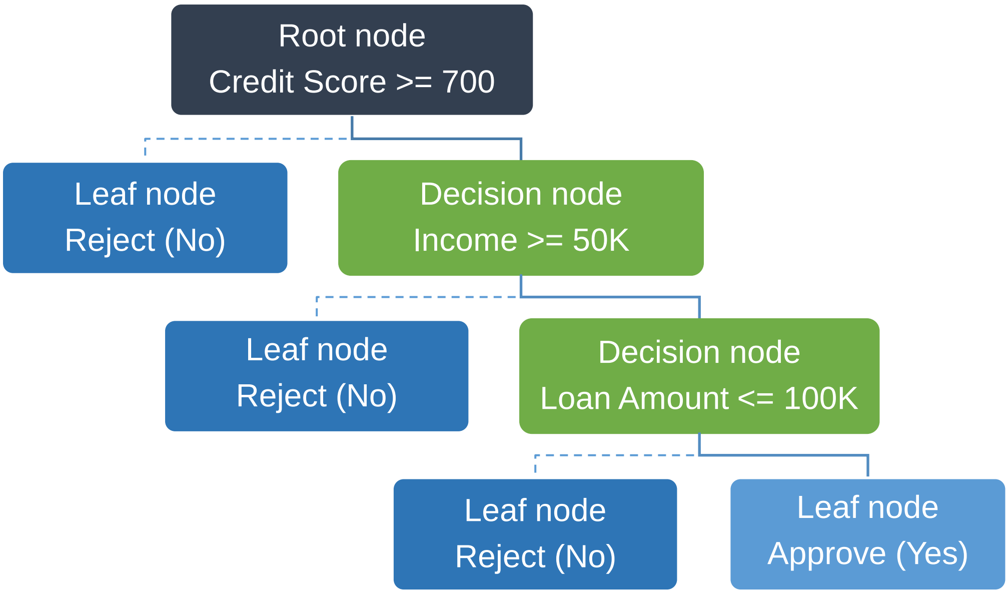


Figure 3. Decision Tree Classifier

4.1.1.3 Random Forest Classifier

The Random Forest Classifier is a sophisticated machine learning algorithm employed for the prediction of categorical variables. This method enhances prediction accuracy and mitigates overfitting by combining (aggregating) the outputs of multiple decision tree models, a process known as ensemble learning. In this approach, the algorithm constructs multiple decision trees and each decision tree is trained on a distinct subset of the training data. By combining numerous decision trees in parallel, the overall variance is reduced, thereby improving the robustness of the model. The Random Forest algorithm can be applied to both classification and regression tasks. For classification problems, the construction of each tree involves measuring the impurity of a node using metrics such as entropy or the Gini index. The final classification output is determined by the majority vote of the individual trees (majority voting classifier) [15].

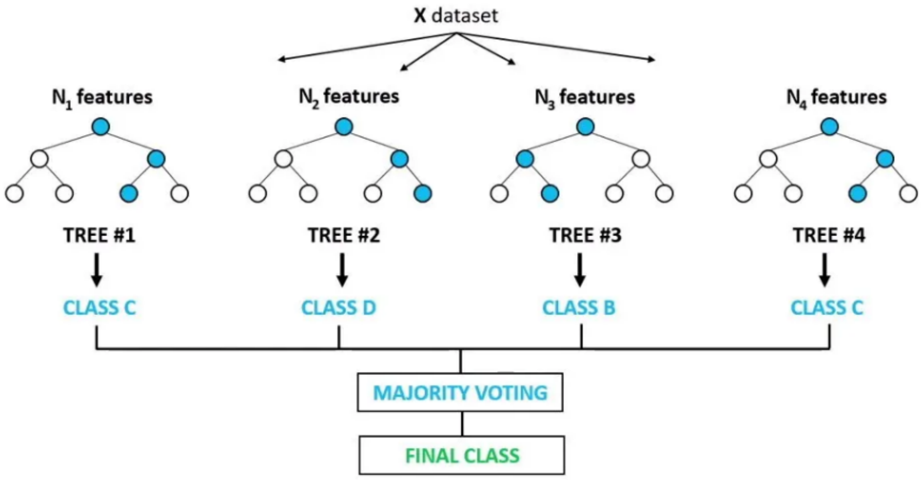


Figure 4. Working principle of Random Forest Classifier [15]

Accordingly, for a new loan applicant, the trained model evaluates their risk of default by aggregating the predictions from all the constructed decision trees. If most of the trees predict a default, the loan application might be rejected or flagged for further review.

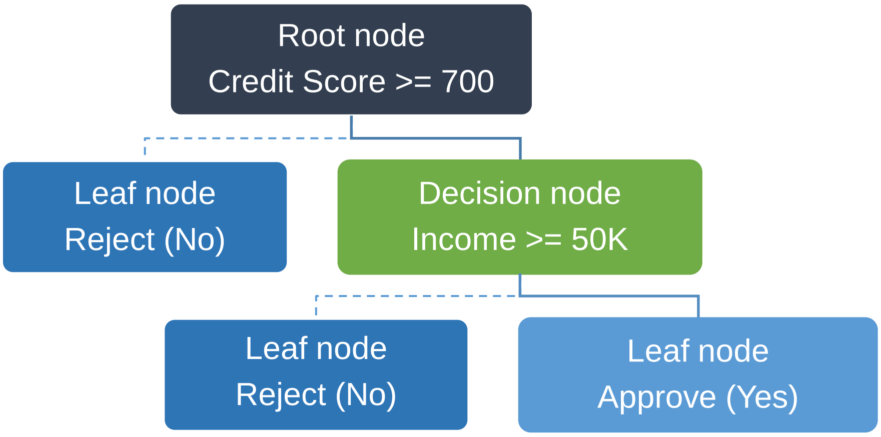


Figure 5. Tree 1 - Random Forest Classifier

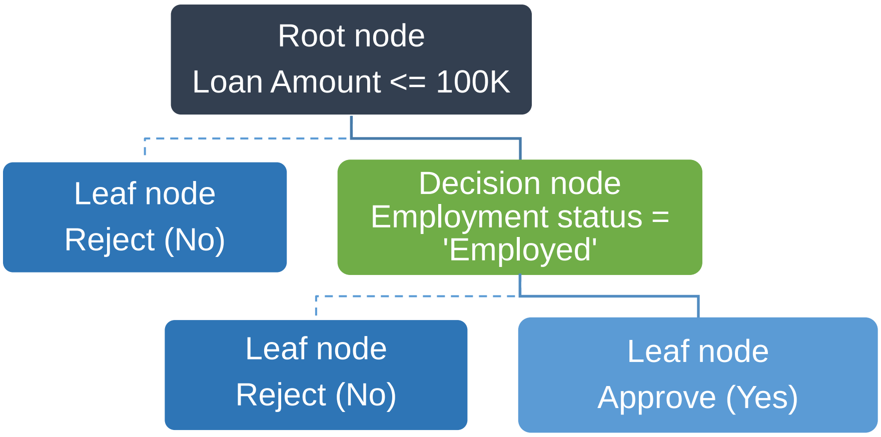


Figure 6. Tree 2 - Random Forest Classifier

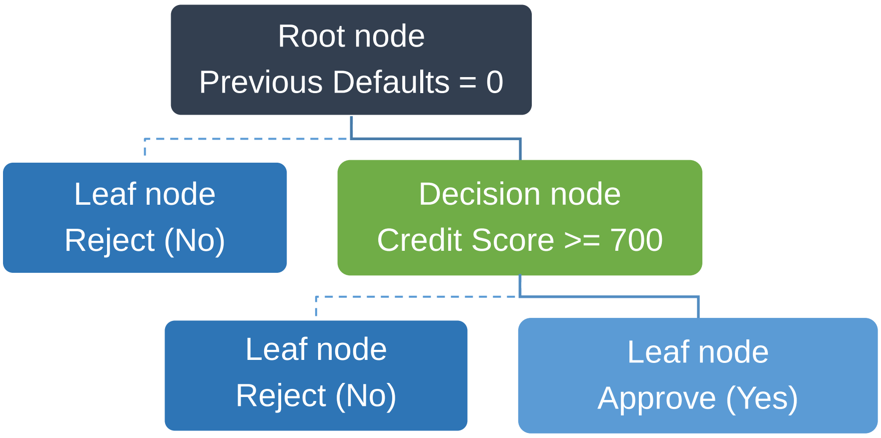


Figure 7. Tree 3 - Random Forest Classifier

4.1.1.4 Support Vector Machine (SVM)

Support Vector Machine (SVM) is a robust machine learning algorithm widely utilized for classification, regression, and outlier detection tasks. The algorithm's primary objective is to identify the optimal hyperplane (decision boundary) in the n-dimensional feature space that effectively separates data points of different classes. The dimensionality of the hyperplane is determined by the number of input features: for two input features, the hyperplane is a straight line; for three input features, it becomes a two-dimensional plane; and so forth. In scenarios where data points are not linearly separable in the input space, SVM employs a kernel function (such as linear, polynomial, or radial basis function) to transform the input data into higher-dimensional feature spaces, thereby simplifying the detection of a separating hyperplane. The closest data points to the hyperplane are called support vectors, and the distance between these support vectors and the hyperplane is referred to as the margin [16]. The SVM algorithm aims to maximize this margin on both sides to ensure the largest possible separation between the classes, thereby increasing classification performance.

During training, the SVM algorithm identifies the optimal hyperplane that best separates the borrowers who defaulted from those who did not. For a new applicant, the SVM calculates the decision function based on the support vectors and determines on which side of the hyperplane the applicant lies. For instance, if the decision function indicates that the applicant is more likely to default, then the model predicts that the new applicant is likely to default.

4.1.1.5 K-Nearest Neighbor (KNN) Classifier

K-Nearest Neighbor (KNN) is a non-parametric statistical method used to predict the class or value of a new data point based on the classes or values of its k-nearest neighbors within the training dataset. Specifically, KNN identifies the k-nearest neighbors (i.e., the closest data points or groups) to a given new data point utilizing a distance metric such as Euclidean, Manhattan, or Minkowski distance. The class or value of the new data point is then determined by the majority vote or the average of these k-neighbors, respectively. KNN is applicable to both classification and regression tasks, as it can handle categorical as well as numerical data.

The parameter k in the KNN algorithm dictates the number of nearest neighbors to be considered and must be selected appropriately based on the input data. For instance, a higher value of k tends to perform better with datasets containing outliers or noise. While there is no definitive method to determine the optimal value of k, a commonly used heuristic is to set k to the square root of the total number of samples in the dataset. To avoid ties during the decision-making process, k is typically chosen as an odd number [17].

K-Nearest Neighbor (KNN) is particularly useful in loan default prediction due to its simplicity and effectiveness in handling both categorical and numerical data. In the context of loan default prediction, KNN can classify borrowers into categories such as likely to default or not likely to default based on the characteristics of their nearest neighbors in the training dataset. For each new loan applicant, the KNN algorithm finds the k-nearest neighbors among the historical data points. The class (default or no default) of the new applicant is determined by the majority vote among these neighbors. For instance, the model identifies the k-nearest neighbors (e.g., k=5). Suppose the 5 nearest neighbors have the following default statuses: [no default, default, no default, no default, default]. The majority vote among the 5 neighbors is "no default" and hence the model predicts that the new applicant is not likely to default.

4.1.1.6 Naïve Bayes Classifier

The Naïve Bayes algorithm is a probabilistic classifier that is established on Bayes' theorem. Bayes' theorem calculates the probability of an event taking place given the probability of another event that has already occurred, which is known as conditional probability:

$$\mathrm{P}(\mathrm{A}|\mathrm{B}) = \frac{\mathrm{P}(\mathrm{B}|\mathrm{A}) \cdot \mathrm{P}(\mathrm{A})}{\mathrm{P}(\mathrm{B})}$$

The Naïve Bayes algorithm predicts the likelihood of an instance belonging to a particular class based on a given set of feature values. Although the algorithm assumes that the features are independent - a condition that is often violated in practice - this assumption typically does not detract from its effectiveness. This presumed independence of feature occurrences is what lends the algorithm its designation as naïve [18].

In the context of loan default prediction, the Naïve Bayes algorithm can classify borrowers as likely to default or not likely to default based on their financial and demographic attributes. The goal is to calculate the posterior probability $\mathrm{P}(\mathrm{D} \mid \mathrm{X})$ which is the probability of default given the features. Similarly, for the posterior probability of the no-default class. To make a prediction, the posterior probabilities for both classes (default and no default) are compared and the class with the higher probability is assigned.

4.1.1.7 Gradient Boosting Classifier

Gradient Boosting Machine (GBM) is an advanced ensemble machine learning method used for both regression and classification problems. In GBM, weights are assigned to the data, and a learning procedure is employed to train the model by combining a set of weak learners. Hence, the improvement in model performance is achieved through the construction of new base learners.

The Gradient Boosting Classifier, a powerful algorithm used for classification tasks, typically consists of an ensemble of Decision Tree models. These trees are trained sequentially, with each subsequent tree attempting to correct the errors of its predecessors. Specifically, Gradient Boosting constructs a series of decision trees in a step-by-step manner, each one addressing the residual errors from the previous trees. The algorithm minimizes a loss function - often the logistic loss for binary classification - by fitting new trees to these residuals. This process, known as gradient descent, aims to reduce the gradient of the loss function.

Loss Function

The loss function quantifies how well the model predicts the target variable. For binary classification, a common choice is the Logistic Loss, also known as binary cross-entropy:

$$L(y, \hat{y}) = -(y \log(\hat{y}) + (1 - y) \log(1 - \hat{y}))$$

where:

- y is the actual label
- $\hat{y}$ is the predicted probability

Model Prediction

The prediction of the model at any iteration m is given by:

$$\hat{y}_m(x) = \sum_{i=1}^{m} f_i(x)$$

where:

- $f_i(x)$ represents the i-th weak learner, typically a decision tree
- m is the total number of trees

Gradient Descent

Gradient Boosting employs gradient descent to minimize the loss function. The update rule for the prediction after adding the new weak learner is:

$$\hat{y}_{m+1}(x) = \hat{y}_m(x) + \gamma f_{m+1}(x)$$

where:

- γ is the learning rate that controls the contribution of the new learner

Residuals

To fit a new learner, the algorithm calculates the residuals of the loss function with respect to the current predictions, which provides the direction in which to improve the model:

$$r_i = -\frac{\partial L(y_i, \hat{y}_m(x_i))}{\partial \hat{y}_m(x_i)}$$

Fitting the Weak Learner

The next weak learner f_{m+1} is fit to the residuals r_i instead of the original target variable, which ensures that each new learner addresses the mistakes made by the ensemble thus far:

$$f_{m+1}(x) = \text{argmin}_f \sum_{i=1}^{N} (r_i - f(x_i))^2$$

Learning Rate

The learning rate η ($0<\eta\leq1$) is introduced to control how much the new learner f_{m+1} influences the overall prediction:

$$\hat{y}_{m+1}(x) = \hat{y}_m(x) + \eta f_{m+1}(x)$$

Unlike common ensemble techniques such as random forests, which rely on simple averaging of models in the ensemble, the boosting approach - similar to bagging - combines base learners by weighted voting. In boosting, new models are added to the ensemble sequentially. During each iteration, a new weak base-learner model is trained based on the errors of the entire ensemble learned so far. In GBMs, the learning procedure consecutively fits new models to provide a more accurate estimate of the response variable. Each tree in the sequence is trained on the residuals of the previous tree, gradually enhancing the model's accuracy. The contribution of each tree is controlled by the learning rate, which determines how much each new tree influences the final prediction. Smaller learning rates generally require more trees to achieve the same error reduction but often result in more robust models.

The high flexibility of GBMs makes them highly customizable for specific data-driven tasks, introducing significant freedom in model design. This flexibility necessitates the careful selection of the most appropriate loss function, often through trial and error. Despite this complexity, boosting algorithms are relatively straightforward to implement, allowing for experimentation with various model designs [19]. In load default prediction, the Gradient Boosting Classifier is trained on the training dataset. During training, the model builds an ensemble of decision trees, each trained to correct the errors of the previous trees. The learning rate and number of trees are hyperparameters that need to be tuned for optimal performance. For a new applicant, the Gradient Boosting Classifier predicts a high or low probability of default based on the input features.

4.1.2 Regression

While classification models are typically employed in loan default prediction, regression models can also offer valuable insights by predicting continuous risk scores or probabilities of default. Regression is a machine learning technique where the model is trained to predict real or continuous values, such as stock prices, temperature, or sales. Specifically, the regression algorithm aims to capture the relationship between independent variables (features or ‘x variables’) and the dependent variable (target or ‘y variable’) by fitting a mathematical model to the data. This mathematical relationship can then be utilized to make precise predictions on new datasets. The primary types of regression techniques include linear regression, polynomial regression, decision tree regression, random forest regression, support vector regression (SVR), and Bayesian linear regression. These methods are not utilized in this study but are included for reasons of completeness.

4.1.2.1 Linear Regression

Linear regression is a statistical method that assumes a linear relationship between one (Simple Linear Regression) or more (Multiple Linear Regression) independent variables (features) and a single dependent variable (target). The algorithm calculates the coefficients of the linear equation y=a+bx that best fits the data, where y is the target variable, x the input feature, b is the slope, and a is the intercept. The linear regression algorithm makes predictions for continuous numeric variables (such as risk score) by calculating how the value of the dependent variable changes according to variations in the independent variable

[20]. The main distinction between linear regression and logistic regression is that linear regression is used to predict continuous dependent variables, whereas logistic regression is employed to predict categorical dependent variables.

In credit risk assessment, the linear regression model is trained using the historical dataset. The model calculates the coefficients a and b of the linear equation that best fits the data. Consequently, the trained model is used to predict the continuous risk score for new loan applicants based on their input features. For example, a higher score may indicate a higher risk of default, whereas a lower score may indicate a lower risk.

4.1.2.2 Polynomial Regression

Polynomial regression is employed when nonlinear relationships exist between the independent variables and the dependent variable. In particular, the polynomial regression model incorporates higher-degree polynomial terms into the linear equation, such as x^2 and x^3, which enable capturing the nonlinear relationship between the features and the target by fitting a polynomial equation to the data. The polynomial regression equation of degree n has the form of $y=a+b_1x+b_2x^2+...+b_nx^n$, where y is the target variable, x is the input feature, $a,b_1,b_2,\dots,b_n$ are the coefficients, and n is the polynomial degree. The selection of the polynomial degree n is important, as a higher-degree model fits the data more closely but may result in overfitting. The model is trained on training data to determine the coefficients and capture the non-linear relationship. Subsequently, it is used to make predictions on new, unseen data [20].

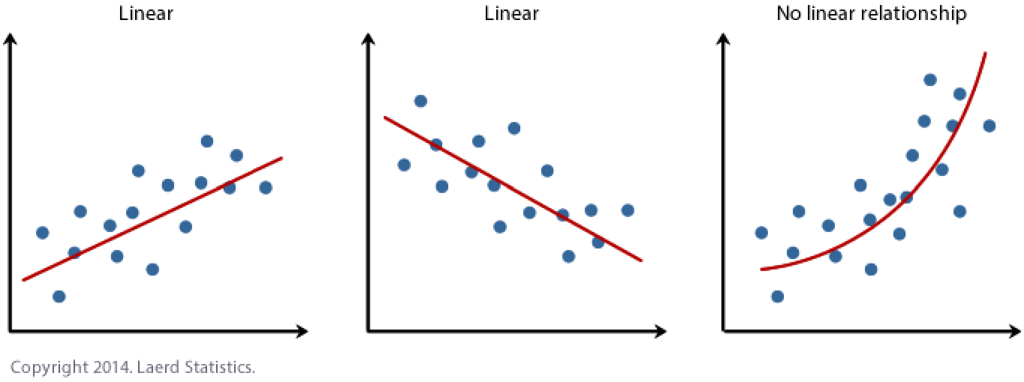


Figure 8. Relationship between variables displayed in scatterplot [21]

When the relationship between the independent variables (features: x) and the dependent variable (risk of default: y) is nonlinear, polynomial regression effectively captures the complexities of these relationships among various borrower characteristics and their likelihood of defaulting on a loan by fitting a polynomial equation to the data. The trained model is subsequently utilized to predict continuous risk scores or probabilities of default for new loan applicants based on their input features.

4.1.2.3 Decision Tree Regressor

A Decision Tree Regressor is employed to predict continuous target variables. The algorithm traverses the tree from the root node to the leaf node that best matches the features of a new data point and calculates the predicted value as the mean of the target variable for all data points within that leaf node. The characteristics and functionality of the decision tree algorithm are comprehensively detailed in section 4.1.1.2. In addition to the primary distinction concerning the type of predicted variable - continuous for the regressor and categorical for the classifier - another significant difference lies in the impurity metric utilized for data splitting. Typically, the Decision Tree Regressor employs the mean squared error (MSE) as its impurity metric, whereas the Decision Tree Classifier uses entropy or the Gini index [22].

In credit risk assessment, during the prediction phase, the new applicant's data is processed through the tree. The algorithm initiates at the root node and follows the branches based on the values of the applicant's features, moving through various decision nodes until it reaches a leaf node. At this leaf node, the algorithm determines the predicted risk score by calculating the mean value of the target variable (risk of default) for all training data points contained within that leaf node. For instance, if the leaf node contains historical data points with risk scores r_1, r_2, r_3, and r_4, the predicted risk score for the new applicant is the arithmetic mean of these values. This score serves as an estimate of the applicant's likelihood of default.

4.1.2.4 Random Forest Regressor

The Random Forest Regressor is a machine learning algorithm used for predicting numerical values. Generally, the random forest approach is an ensemble technique that enhances performance by combining the predictions of multiple decision trees, as discussed in section 4.1.1.3. The construction of decision trees within the ensemble involves selecting the most optimal attributes for splitting at each node. For regression problems, node impurity is assessed using mean squared error (MSE), and the final prediction output is computed as the average of all individual tree outputs [22]. The predictions from multiple trees are aggregated using a technique known as bootstrap aggregating (bagging), wherein the individual model predictions are combined by averaging.

To conduct credit risk assessment, the Random Forest Regressor is trained on historical data to understand the relationship between various features and the risk of default. Each tree within the forest is trained on a random subset of the data and features, enhancing the model's robustness. During the prediction phase, each tree independently provides a prediction based on the applicant's features. The final risk score is obtained by averaging the predictions from all the trees. For instance, if the following risk scores are predicted by five different trees in the forest: Tree 1: 2.5, Tree 2: 2.7, Tree 3: 2.9, Tree 4: 3.0, and Tree 5: 2.6, the final predicted risk score for the new applicant would be the mean of these values.

4.1.2.5 Support Vector Regression (SVR)

Support Vector Regression (SVR) is a variant of the Support Vector Machine (SVM) algorithm, designed for predicting continuous numeric values. While SVM is primarily utilized for classification tasks, SVR is specifically implemented for regression tasks. SVR identifies a function that predicts a continuous output value corresponding to an input value, effectively handling both linear and nonlinear data through the use of kernel functions such as linear and radial basis function (RBF) kernels. In SVR, the hyperplane represents the regression line that best fits the data, along with a margin of tolerance. This margin is controlled by two hyperparameters: epsilon (ε), which defines the width of the margin, and C, which regulates the trade-off between maximizing the margin and minimizing the prediction error [23]. The objective of SVR is to find a hyperplane that maximally separates different output values while maintaining the specified tolerance margin.

In the context of credit risk assessment, the Support Vector Regression (SVR) model is trained using a historical dataset. During the training phase, the model identifies a function that predicts continuous risk scores based on input features, employing kernel functions to manage both linear and nonlinear relationships. Once trained, the model is utilized to predict continuous risk scores or probabilities of default for new loan applicants based on their respective input features.

4.1.2.6 Bayesian Linear Regression

Bayesian Regression is a form of linear regression that utilizes Bayes' theorem to estimate the unknown parameters of the model. The objective of Bayesian regression is to derive the best estimates for the parameters of a linear model that describes the relationship between the dependent variables and the independent variables. While traditional linear regression assumes that data follows a Gaussian or normal distribution, Bayesian regression assumes a probability distribution on the parameters [24]. The goal is to determine the distribution of the model parameters, rather than identifying a single best value for these parameters.

4.2 Unsupervised Learning Algorithms

In the realm of unsupervised learning, models are designed to work with data that do not have pre-existing labels or categories, commonly referred to as unlabeled data. Unsupervised machine learning algorithms analyze and group this unstructured data based on inherent patterns, similarities, or differences, without any form of external guidance or supervision. These algorithms are primarily categorized into two types: clustering, which aims to partition the data into distinct groups based on feature similarity, and association, which seeks to identify interesting relationships or associations between variables within the dataset. These methods are not implemented in this study but are mentioned for reasons of completeness.

4.2.1 Clustering

Cluster analysis is a fundamental technique in unsupervised learning, with the objective of grouping similar data points from a heterogeneous dataset, such as those used in fraud detection, loan default prediction, or customer segmentation. This process involves creating groups of homogeneous data points by evaluating their similarity based on a specific metric, referred to as a similarity measure (e.g., Euclidean distance, Manhattan distance, Cosine similarity). Data points with the highest similarity are clustered together, resulting in the dataset being divided into distinct clusters that represent collections of similar data. Common clustering algorithms include centroid-based clustering (partitioning methods) such as k-means clustering, density-based clustering (model-based methods) exemplified by DBSCAN (Density-Based Spatial Clustering of Applications with Noise), connectivity-based clustering (hierarchical clustering), and distribution-based clustering, such as the Gaussian Mixture Model (GMM) [25].

Financial institutions collect historical data on borrowers, which is then analyzed using clustering techniques to identify groups of borrowers with similar characteristics. For instance, clustering can aid in distinguishing between high-risk and low-risk borrowers based on their financial behavior and credit history. Features that significantly influence the likelihood of loan default are selected as input variables for the clustering algorithm. Algorithms such as k-means are applied to the dataset to group similar data points into clusters. These resulting clusters are analyzed to understand the unique characteristics of each group [26]. For example, one cluster may represent borrowers with high credit scores and low debt-to-income ratios, indicating a low risk of default, while another cluster may represent borrowers with low credit scores and high debt-to-income ratios, suggesting a high risk of default.

4.2.2 Association

Association rule learning in the context of unsupervised learning and data mining is utilized to identify correlations and relationships among variables within large datasets, thereby enabling predictions based on these discovered patterns. This method is especially valuable for analyzing customer behavior (such as market basket analysis and customer segmentation), detecting fraudulent activities, performing social network analysis and developing recommendation systems. Association rule learning uncovers rules that describe the connections between different items. These rules follow an "if-then" structure, where the "if" part, known as the antecedent, represents the condition being evaluated, and the "then" part, known as the consequent, indicates the outcome that occurs when the condition is satisfied. By identifying these associations, the rules clarify how different variables may be interrelated [27]. Association rule learning encompasses three main types of algorithms: Apriori, Eclat, and FP-Growth.

Specifically, the derived rules can assist in identifying combinations of borrower attributes that are closely linked to higher or lower risks of default. An association rule learning algorithm can be applied on the dataset to uncover rules that reveal the relationships between various borrower characteristics and the likelihood of default. These resultant association rules are then examined to identify the primary factors contributing to loan defaults. For instance, a rule might indicate that borrowers with a credit score under 600 and a debt-to-income ratio exceeding 40% are very likely to default on their loans [28]. This insight can be leveraged by financial institutions to evaluate the credit risk of new applicants and make more informed lending decisions.

5. Loan Default Prediction

5.1 Loan Agreements and Default Conditions

A loan is defined as a sum of money borrowed by an individual or business (borrower) from a bank, corporation, government, or another entity (lender) for a specific purpose. This borrowed amount is provided with the expectation of repayment, which includes the loan principal and additional charges such as interest. The borrower is obligated to repay both the principal and the accrued interest. To assess the borrower's repayment capability, the lender typically requires detailed information, including the borrower's financial history, income, credit score, and debt-to-income ratio. Based on this evaluation, the lender either approves or denies the loan application, depending on the applicant's creditworthiness. If the loan is approved, both parties sign a contract that outlines the terms and conditions of the loan agreement. Occasionally, borrowers may fail to adhere to the agreed-upon payment schedule. After a grace period, usually ranging from 30 to 90 days, a loan is considered in default if the borrower continues to miss the required payments.

5.2 Business Case: Loan Default Prediction and Financial Risk

To mitigate financial risk, decision makers must identify the factors influencing loan repayment. Various algorithms have been utilized to predict a borrower's default probability. Loan default prediction presents two possible outcomes: either the applicant fully repays the loan (non-defaulter) or defaults (defaulter). Consequently, classification algorithms such as logistic regression, decision tree classifier, random forest classifier, gradient boosting classifier and KNN classifier are well-suited for predicting these binary outcomes.

Each model must identify the factors most significantly affecting loan default and prioritize minimizing false positives and false negatives. In this study, false negatives occur when the model fails to detect loans that will default, while false positives occur when the model incorrectly rejects a good candidate's application as a potential loan default. This issue poses a significant concern for lending financial institutions, as it leads to financial and business losses that increase the institution's exposure to financial risk and jeopardize its stability.

The main risks include both the business loss due to not approving good candidates or rejecting too many of them (false positive) and the financial loss from approving candidates who default and do not repay the loan (false negative). To deal with these concerns, models are evaluated and optimized using accuracy, precision, and recall metrics.

5.3 Model Evaluation Metrics

In this loan default prediction scenario, the positive class (1) represents an individual who charges off and does not repay the loan, while the negative class (0) represents an individual who fully repays the loan. This convention aligns with the common practice of labeling the class of interest or the rarer event as 1.

A true positive (TP) occurs when an individual is predicted to default (1) and indeed defaults (1). A false positive (FP) occurs when an individual is predicted to default (1) but actually repays the loan in full (0). Conversely, a true negative (TN) refers to an individual predicted to repay the loan in full (0) who indeed repays the loan in full (0). A false negative (FN) refers to an individual predicted to fully pay the loan (0) but who actually defaults (1).

Table 1. Confusion matrix - Loan Default Prediction

		Predicted	
		0: Fully Paid	1: Charged Off
Actual	0: Fully Paid	True Negatives	False Positives
	1: Charged Off	False Negatives	True Positives

5.3.1 Accuracy

Accuracy indicates the frequency with which the model correctly predicts outcomes, defined as the percentage of correct predictions over the total number of predictions. However, in the case of highly imbalanced datasets - where one category occurs much more frequently than the other (e.g., a majority to minority class ratio of 80:20 or 90:10) - the accuracy metric may be misleading due to the model's successful performance on the majority class. This occurs because accuracy treats all classes as equally important and counts all correct predictions without differentiating between the significance of predicting the majority or minority class correctly.

$$\text{Accuracy} = \frac{\text{True Positives} + \text{True Negatives}}{\text{True Positives} + \text{True Negatives} + \text{False Positives} + \text{False Negatives}}$$

5.3.2 Precision

To comprehensively evaluate the model's performance and its effectiveness in predicting true positives, additional metrics such as precision and recall are calculated. Precision measures the accuracy of the model when predicting the positive class and is calculated as the number of correct positive predictions (TP) divided by the total number of predicted positive instances (TP + FP). A perfect precision score of 1.0 is achieved when the model always correctly predicts the positive class.

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} = \frac{\text{True Positives}}{\text{Total Predicted Positives}}$$

Precision is particularly important when the cost of a false positive is high. A high precision score indicates that the model makes few false positive predictions, applying a stringent approach to classifying a sample as positive. However, precision does not account for false negatives, necessitating the use of the recall metric to provide a more balanced evaluation.

5.3.3 Recall

Recall measures how frequently the model accurately detects the positive class from all actual positive instances. It is calculated as the number of true positives (TP) divided by the total number of actual positive instances (TP + FN). A perfect recall score of 1.0 is achieved when the model identifies all positive instances in the dataset.

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} = \frac{\text{True Positives}}{\text{Total Actual Positives}}$$

Recall is particularly important when the cost of false negatives is high, and the focus is on identifying all the instances of positive class, even if this may result in an increase in false positives. This approach entails a more lenient criterion for categorizing a sample as positive.

Ideally, the model should be evaluated using multiple metrics simultaneously to define the right balance between precision and recall. If false positive errors are more critical and costly than false negatives then precision should be considered and optimized, and vice versa, if false negatives are more important and costly then recall should be prioritized and maximized.

A higher precision score involves a stricter classification approach, which may result in a reduced recall score by doubting the actual positive samples from the dataset. On the other hand, a higher recall score implies a lax classification approach, which may reduce the precision score by allowing borderline negative samples to be classified as positive. Ideally, both precision and recall should be maximized to achieve a perfect classifier.

5.3.4 F1 Score

The trade-off between precision and recall is managed by the F1 score, a metric that combines precision and recall, defined mathematically as their harmonic mean. The F1 score increases when precision and recall are similar and decreases when there is a significant deviation between the two. The highest possible F1 score is 1.0, indicating perfect precision and recall, while the lowest is 0.

$$\text{F1 Score} = \frac{2}{\frac{1}{\text{Precision}} + \frac{1}{\text{Recall}}} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

Maximizing the F1 score means simultaneously maximizing both precision and recall. As the F1 score incorporates both precision and recall into a single metric, it is well-suited for evaluating the model alongside accuracy. Additionally, it aids in optimizing the balance between revenue and default loss, thereby maximizing profit for the business or lending institution.

Table 2. Model Evaluation Metrics for binary classification

	Model Prediction		**Total**	**Formula**
Actual	TN	FP	Total Actual Negative cases	
	FN	TP	Total Actual Positive cases	**Recall** = TP/(TP+FN)
Total	Total Predicted Negative cases	Total Predicted Positive cases	Total Cases	
Formula		**Precision** = TP/(TP+FP)	**Accuracy** = (TP+TN)/(TP+FP+FN+TN)	**F1 Score** = (2xPxR)/(P+R)

5.3.5 ROC curve

The ROC (Receiver Operating Characteristic) curve is a graphical tool used to evaluate the performance of a binary classification model. It is particularly valuable for assessing binary classifiers, as it plots the True Positive Rate (TPR or Recall) against the False Positive Rate (FPR) at various threshold settings. TPR measures the proportion of actual positives correctly identified, while FPR measures the proportion of actual negatives incorrectly classified as positives.

$$\text{TPR} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

$$\text{FPR} = \frac{\text{False Positives}}{\text{False Positives} + \text{True Negatives}}$$

The ROC curve is instrumental in assessing the model's ability to discriminate between positive and negative classes. The Area Under the ROC Curve (AUC-ROC) provides a single scalar value that summarizes the model's overall performance. Hence, this visualization is especially useful for comparing multiple models; a model with a curve closer to the top-left corner generally indicates superior performance, as indicated in the corresponding figure.

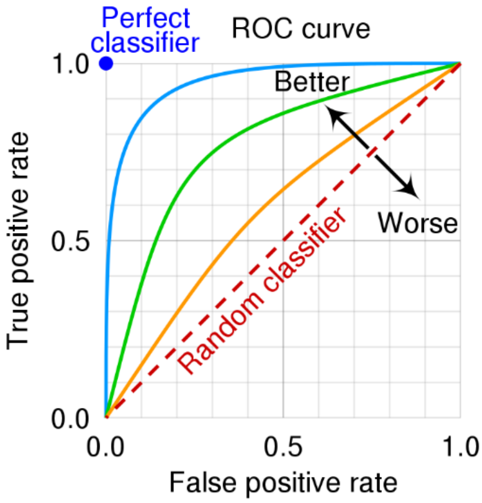


Figure 9. ROC Curve for multiple classifiers [29]

In situations involving imbalanced datasets, where metrics like accuracy may be misleading, the ROC curve offers a more balanced evaluation by concentrating on TPR and FPR. Furthermore, the ROC curve can assist in selecting the optimal decision threshold for a classifier. Although the default threshold is generally set to 0.5, analyzing the curve enables adjustment of this threshold to better balance the trade-off between TPR and FPR based on the specific requirements of the application. This decision threshold is a critical factor in binary classification, determining when the classifier transitions from predicting the negative class to the positive class.

5.4 Peer-to-peer Lending Dataset

5.4.1 Raw Dataset

The raw dataset comprises data on issued loans from 2007 to 2018 by the P2P lending platform, including the current loan status (Current, Fully Paid, Charged Off). It encompasses various details about previous loans and borrowers, such as annual income, years of employment, loan amount, debt-to-income ratio (dti), spending behavior, number of accounts, and as previously referred the current status of the loan. The raw dataset contains over 2 million records (observations) and 151 columns (features), representing a substantial volume of data that requires significant computational resources and processing time. Therefore, the raw data must undergo preparation, cleaning, and transformation prior to processing and analysis, which is achieved through Exploratory Data Analysis (EDA).

5.4.2 Exploratory Data Analysis

Exploratory Data Analysis (EDA) is utilized to examine and investigate the raw dataset, summarizing its key characteristics. The fundamental steps of EDA include understanding the problem and the data, importing and inspecting the dataset, handling missing values, exploring data properties, performing data transformations, visualizing data relationships, managing outliers, and communicating findings and insights. EDA ensures data quality and relevance, aiding in the selection of the most appropriate and valuable features for subsequent analysis.

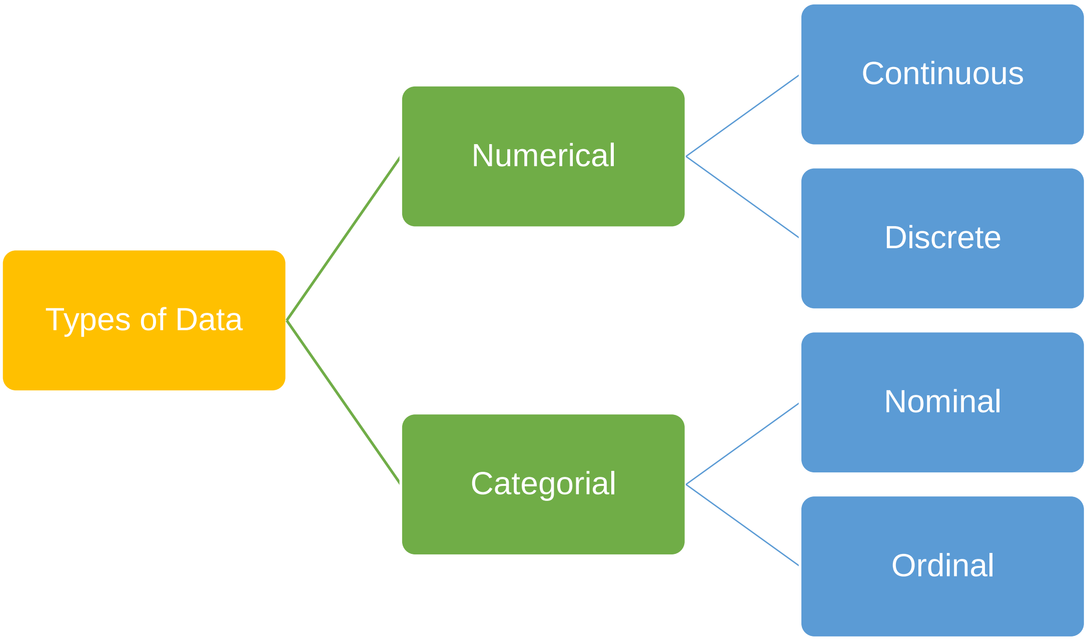


Figure 10. Types of Data

5.4.2.1 Feature selection

In loan default prediction, the primary objective is to identify patterns that can detect candidates likely to default, with the positive class (1) representing this outcome. The model's results are utilized to approve or reject loan applications and to determine the loan amount and interest rate. For instance, identifying a high-risk borrower may lead to a reduction in the loan amount and an increase in the interest rate. Thus, it is crucial to select the most relevant and informative features from the dataset for analysis and modeling. The selected features, presented in the table below, encompass information about the loan, the borrower's employment and income, the borrower's profile and demographics, credit behavior, and details of accounts and credit lines.

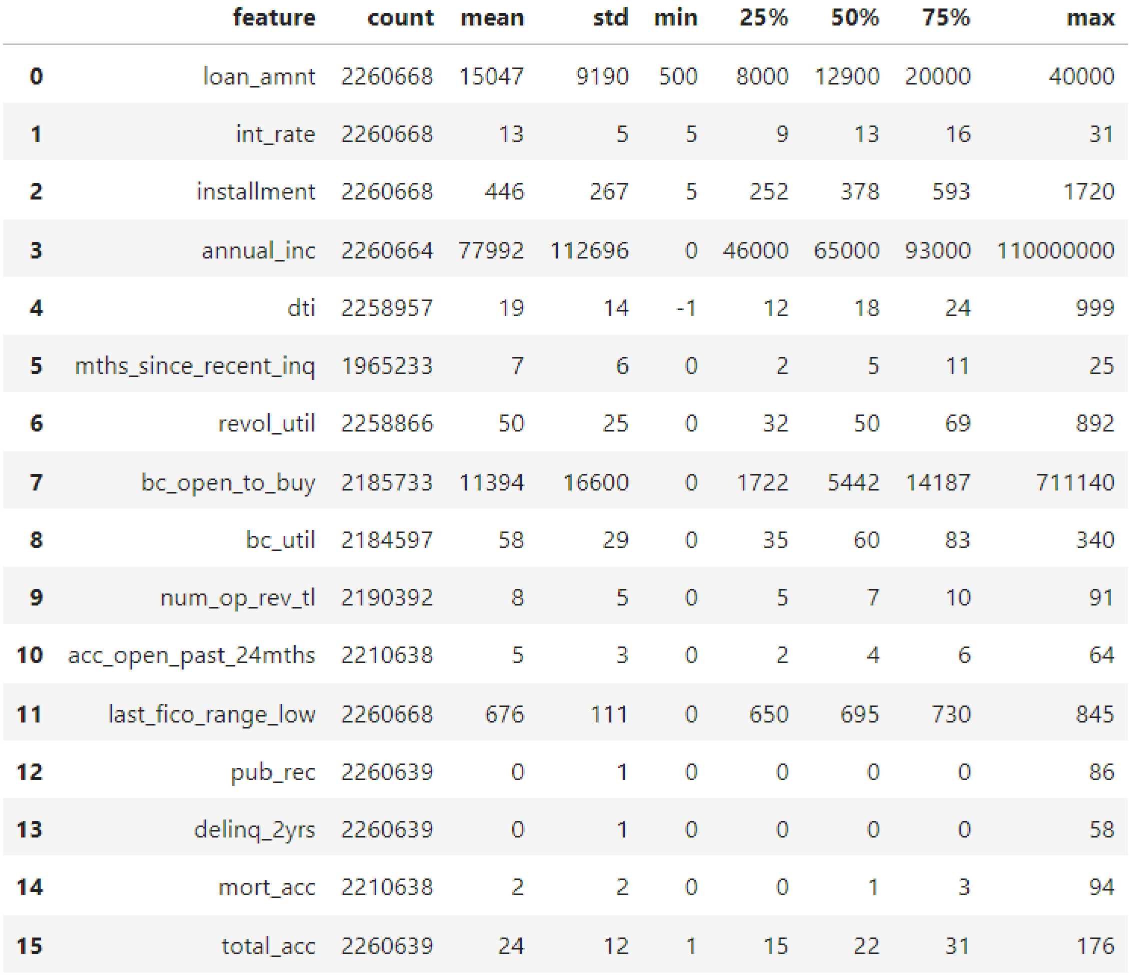
Table 3. Feature Selection from the Raw dataset

	Feature	Decription	Data type	Column name
1.	Loan amount	Loan amount requested by the borrower	numerical (float64)	'loan_amnt'
2.	Repayment schedule	The length of time for which the loan is taken, expressed in months (range of 36 to 60 months)	categorical (string)	'term'
3.	Interest rate	The interest rate charged on the loan	numerical (float64)	'int_rate'
4.	Installment	The fixed periodic payment amount that the borrower must pay to the lender to repay the loan over a specified period	numerical (float64)	'installment'

5.	Subgrade	Loan subgrade score determined by the credit history of the borrower, providing additional granularity (e.g., A1, A2, B1)	categorical (string)	'sub_grade'
6.	Employment title	The job title of the borrower	categorical (string)	'emp_title'
7.	Employment length	Years of employment for the borrower	categorical (string)	'emp_length'
8.	Homeownership status	The status of the borrower's home ownership, such as Rent, Own, or Mortgage	categorical (string)	'home_ownership'
9.	Annual income	The annual income of the borrower	numerical (float64)	'annual_inc'
10.	State address	The state declared by the applicant in the loan application	categorical (string)	'addr_state'
11.	Debt-to-income ratio	Debt-to-income ratio, calculated as the ratio of the borrower's total monthly debt payments to their gross monthly income	numerical (float64)	'dti'
12.	Delinquency	The number of times the borrower had been delinquent for 30+ days in the past 2 years	numerical (float64)	'delinq_2yrs'
13.	Derogatory public records	The number of derogatory public records (bankruptcies, tax liens, or judgments)	numerical (float64)	'pub_rec'
14.	Revolving line utilization rate	Revolving line utilization rate, calculated as the ratio of the borrower's credit being used to the total amount of available credit	numerical (float64)	'revol_util'
15.	Total number of credit lines	The total number of credit lines in the borrower's credit history	numerical (float64)	'total_acc'
16.	FICO score	The lower bound of the most recent FICO score range of the borrower	numerical (float64)	'last_fico_range_low'
17.	Number of accounts in the past 24 months	The number of accounts/trades the borrower has opened in the past 24 months	numerical (float64)	'acc_open_past_24mths'

18.	Open-to-buy amount on bankcards	The total open-to-buy amount on the borrower's revolving bankcard accounts (i.e., the remaining credit limit)	numerical (float64)	'bc_open_to_buy'
19.	Bankcard utilization	Bankcard utilization, calculated as the ratio of the total current balance to the maximum credit limit on bankcard accounts	numerical (float64)	'bc_util'
20.	Number of mortgage accounts	The number of mortgage accounts the borrower has	numerical (float64)	'mort_acc'
21.	Months since most recent inquiry	The number of months since the borrower's most recent credit inquiry	numerical (float64)	'mths_since_recent_inq'
22.	Number of open revolving accounts	The number of open revolving accounts the borrower has	numerical (float64)	'num_op_rev_tl'
23.	Current loan status (target variable)	The current status of the loan, such as Fully Paid, Charged Off, or Current, which is the label that the model will predict	categorical (string)	'loan_status'

Table 4. Descriptive statistics for numerical features in Dataset



	feature	count	mean	std	min	25%	50%	75%	max
0	loan_amnt	2260668	15047	9190	500	8000	12900	20000	40000
1	int_rate	2260668	13	5	5	9	13	16	31
2	installment	2260668	446	267	5	252	378	593	1720
3	annual_inc	2260664	77992	112696	0	46000	65000	93000	110000000
4	dti	2258957	19	14	-1	12	18	24	999
5	mths_since_recent_inq	1965233	7	6	0	2	5	11	25
6	revol_util	2258866	50	25	0	32	50	69	892
7	bc_open_to_buy	2185733	11394	16600	0	1722	5442	14187	711140
8	bc_util	2184597	58	29	0	35	60	83	340
9	num_op_rev_tl	2190392	8	5	0	5	7	10	91
10	acc_open_past_24mths	2210638	5	3	0	2	4	6	64
11	last_fico_range_low	2260668	676	111	0	650	695	730	845
12	pub_rec	2260639	0	1	0	0	0	0	86
13	delinq_2yrs	2260639	0	1	0	0	0	0	58
14	mort_acc	2210638	2	2	0	0	1	3	94
15	total_acc	2260639	24	12	1	15	22	31	176

5.4.2.2 Missing values

Typically, datasets contain missing values where no data (null) is recorded in one or more features of a record. Missing data can result in biased estimates, loss of valuable information, decreased statistical power, and diminished generalizability of findings. The proportion of missing data directly impacts the quality of statistical inferences. A common simplistic approach for handling missing values is to delete the affected observations by dropping rows with null values. However, instead of discarding valuable data and creating gaps in the analysis, missing values can be addressed through data imputation.

Data imputation is a method that retains most of the data and information by substituting missing values with alternative values from the dataset. For continuous variables, the simplest imputation methods involve replacing missing values with the mean or median of the dataset or neighboring values or using similar summary statistics. For categorical features, null values can be replaced by the most frequent category. While more complex techniques for predicting missing values exist, they are not implemented in this study.

Despite the fact that filling in missing values might seem insignificant, it can potentially lead to biased and misleading results. For example, if the data follows a Gaussian probability distribution, imputing values with the average of neighboring data points will likely increase the concentration of values around the mean and reduce the distribution's tails, thus decreasing data variability and lowering the standard deviation. Using the minimum value for imputation can create a negatively skewed distribution (increased number of low values), whereas using the maximum value can result in a positively skewed distribution (increased number of high values). Despite these risks, it is often necessary to fill in missing values, but this should be done cautiously, employing best practices for imputation. Although there is no universally accepted percentage of missing data in a dataset, a missing rate of 5% or less is generally considered inconsequential, while a missing rate of 10% or more is likely to introduce bias [30]. Therefore, as there is no absolute cutoff for the number of missing values to be imputed, this study adopts the common practice of filling in no more than 5-10% of the null values in the dataset.

In this specific case, as the dataset contains a substantial number of records and the missing rate is close to the aforementioned margins, the study will proceed by dropping the missing values. Additionally, the dataset was reviewed to determine whether the missing data is Missing Completely at Random (MCAR), Missing at Random (MAR), or Missing Not at Random (MNAR) [31]. It appears that there is no relationship between the missingness of the data and any observed or missing values, indicating that the missing data points are a random subset of the data (MCAR).

5.4.2.3 Outliers (Numerical features)

An outlier is an element that significantly deviates from the distributional behavior of the other elements in a statistical sample [32]. In other words, an outlier is a data point that differs notably from the rest of the data points and behaves differently. Outliers can result from measurement or execution errors and are generally classified into three types: a) global or point outliers, which are individual data points that significantly deviate from the overall dataset distribution; b) collective outliers, which consist of groups of data points that together deviate significantly from the overall dataset distribution; and c) contextual or conditional outliers, which are data points that significantly deviate from expected behavior within a specific context or subgroup [33].

Detecting outliers is essential for maintaining the quality and accuracy of machine learning models, as outliers can significantly impact their performance. For instance, certain models like linear regression are particularly sensitive to outliers due to their reliance on minimizing the sum of squared errors. Outliers with large residuals can skew the model's estimates, leading to biased predictions. In a model that predicts a target variable using multiple features, an outlier with an exceptionally high value can heavily influence the regression line, resulting in an overestimated prediction for the target variable. On the other hand, models such as decision trees and random forests exhibit greater robustness to outliers because they base their decisions on splits that are less affected by individual data points. Nonetheless, outliers can still influence the tree-building process by affecting the purity of splits or altering the tree structure.

Effectively identifying, removing, or handling outliers can enhance the performance of a model. Consequently, addressing outliers is a crucial step in the data preprocessing phase of this study. Several common techniques can be used to detect outliers, depending on the specific dataset, the analysis or modeling technique used, and the underlying domain knowledge. These techniques include: a) visualization tools such as box plots, which visually display data distribution and highlight potential outliers, and scatter plots, where outliers can be identified as data points that substantially deviate from the overall pattern; b) distance-based metrics like Z-scores, which measure how far a data point is from the dataset mean in terms of standard deviations; c) various statistical tests, where outliers are identified as data points that significantly deviate from the expected distribution or variable relationships [31]; and d) clustering algorithms that automatically group similar data points, isolating outliers into separate clusters [33].

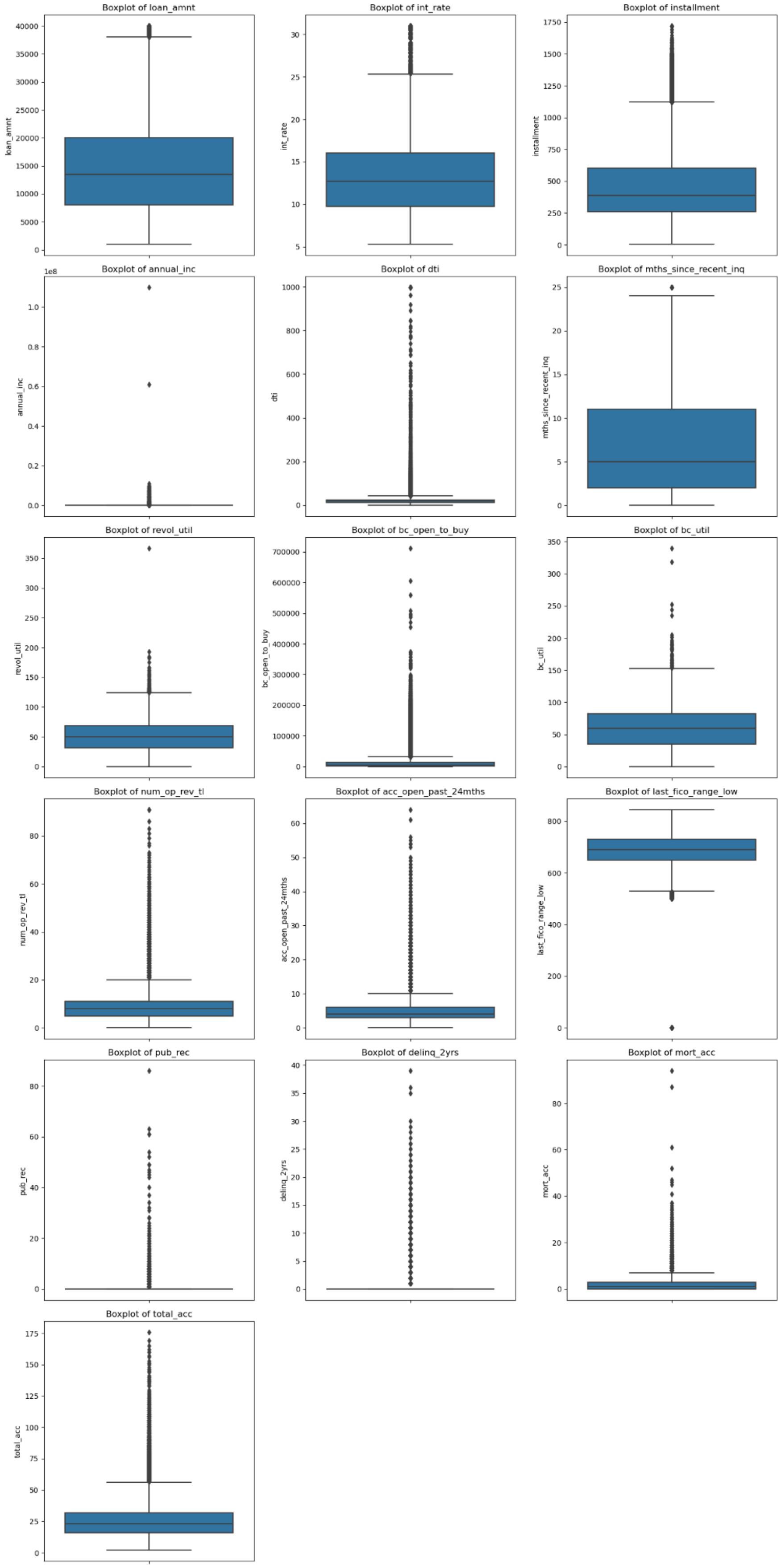


Figure 11. Numerical feature distribution before removing outliers (boxplots)

As shown in the boxplots, 'annual_inc', 'dti', 'revol_util', 'bc_open_to_buy', 'bc_util', 'num_op_rev_ti' have outliers defined as the individual points beyond the whiskers of the boxplot that differ significantly from the rest of the dataset.

From a business perspective, the objective is to determine whether an outlier should be replaced or retained and to understand the reason for its existence. Depending on the nature of the data and the modeling approach employed, various strategies can be used to handle outliers [34]. There are three common methods for managing outliers: a) imputation, which involves substituting or filling in outlier values with estimated or imputed values; b) clipping, which involves capping or truncating extreme values at a specified threshold, such that values above a certain percentile (e.g., the 95th percentile) are set to the value of that percentile, and values below a certain percentile (e.g., the 5th percentile) are set to the value of that percentile. A percentile is a value below which a certain percentage of the data falls (e.g., k% of the data falls below the kth percentile); and c) removing outliers based on an outlier formula, which identifies outliers using upper and lower boundaries (cutoff points) [31].

In this study, the last approach is adopted, utilizing the Interquartile Range (IQR) method. The data are divided into four quantiles, with three quartiles (Q_1, Q_2, Q_3) splitting the data into four equal parts. Quartiles are a type of percentile that divide sorted data into equal sections. Any value that exceeds the third quartile by 1.5 times the IQR (Q_3 + 1.5xIQR) is classified as an outlier, and any value that is less than the first quartile by 1.5 times the IQR (Q_1 - 1.5xIQR) is also classified as an outlier. The interquartile range (IQR) is the difference between the first quartile (Q_1) and the third quartile (Q_3).

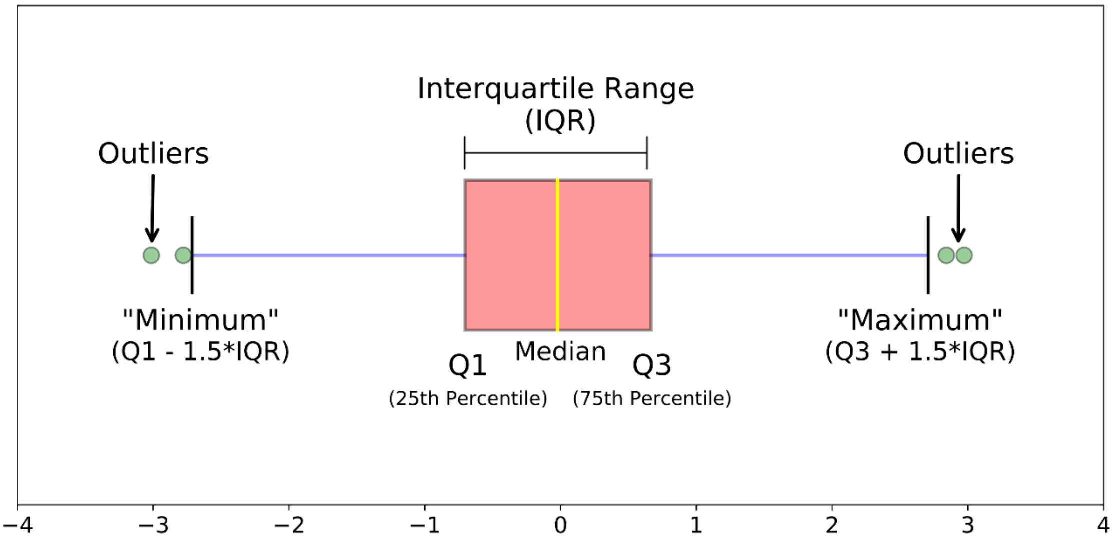


Figure 12. Characteristics of a Boxplot [35]

The outliers are eliminated by using the outlier formula described, that uses upper and lower boundaries (IQR method). The boxplots created after removing outliers are shown below.

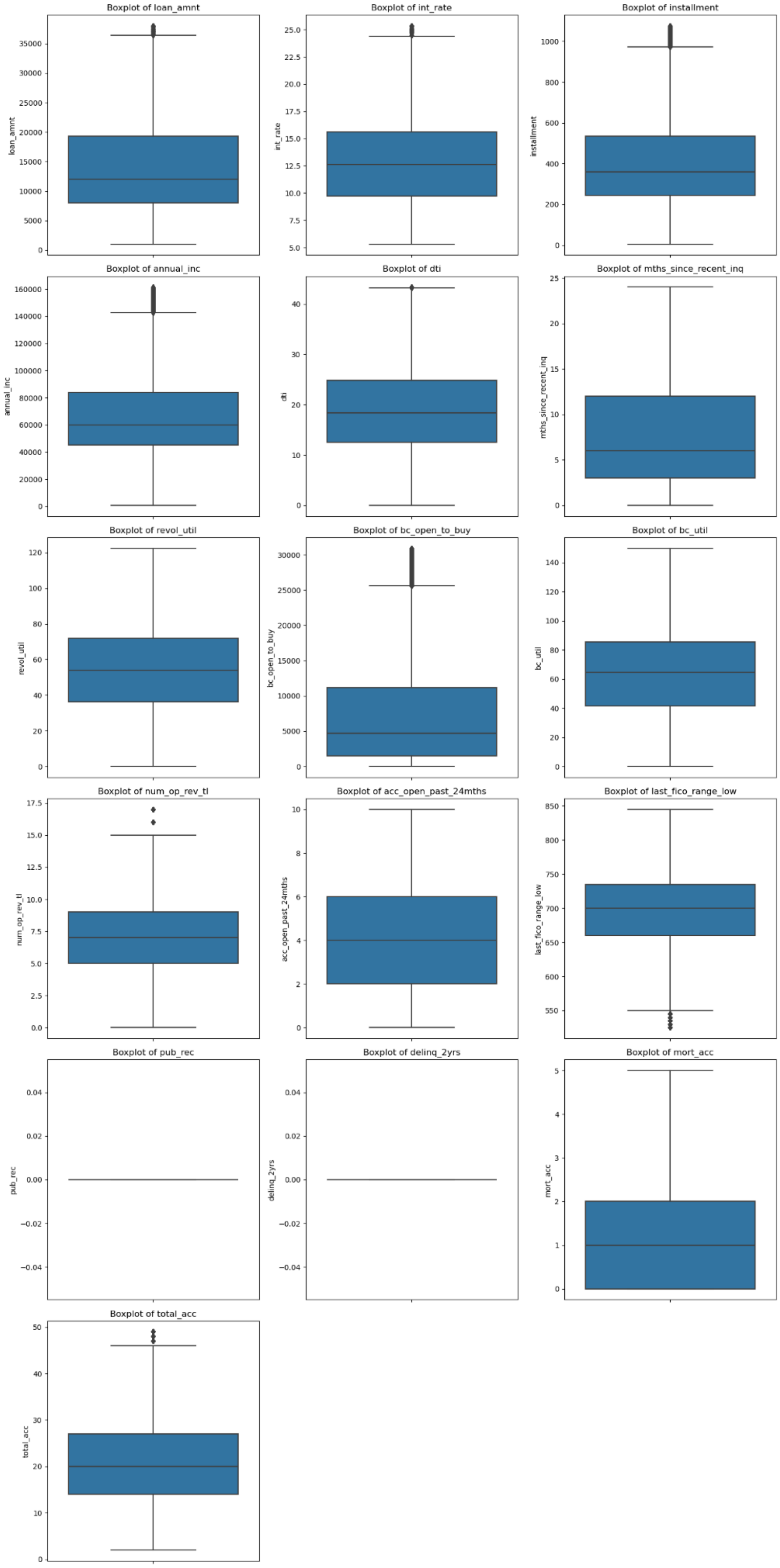


Figure 13. Boxplots after removing outliers in the initial dataset

5.4.2.4 Categorical features

Encoding is essential for handling categorical variables in machine learning and data analysis, as models cannot process textual data. By encoding, categorical variables are converted into numerical values that algorithms can comprehend and interpret. This process involves transforming data in the form of strings or object data types into integer format, making it suitable for use in various models. Two widely used techniques for encoding categorical data are one-hot encoding and ordinal encoding.

One-hot encoding

This method converts categorical features into a matrix where each category is represented by a new separate column. Each instance is marked with a value of 1 in the respective column and 0 in all other columns (binary values of 1 or 0). This technique is applied to handle nominal variables, which are categorical variables with two or more categories that cannot be ordered or ranked (e.g., color). One-hot encoding is useful for data where categories have no inherent relationship to each other. Machine learning algorithms often interpret the order of numbers as significant, potentially viewing higher numbers as better or more important than lower ones. In this study, one-hot encoding is used for the nominal features 'home_ownership' and 'addr_state' in the raw dataset.

Table 5. One-hot Encoding for 'home_ownership' feature

Data	MORTG	RENT	OWN
MORTGAGE	1	0	0
RENT	0	1	0
OWN	0	0	1

Ordinal encoding

Ordinal encoding assigns a distinct integer value to each category based on their natural order, preserving the inherent ranking while converting ordinal data into numerical form. This technique is used for categorical features where one category is greater or lesser than another. For example, in the sequence very dissatisfied < dissatisfied < neutral < satisfied < very satisfied, the category very dissatisfied would be assigned the value 1, while very satisfied would be assigned the highest value of 5. In this study, ordinal encoding is utilized for the ordinal features 'term', 'sub_grade', and 'emp_length', as these have a clear hierarchical order in the raw dataset.

Table 6. Ordinal Encoding - mapping to integer values

Ordinal feature	Original data	Encoded data
'term'	36 months	1
	60 months	2
'sub_grade'	A1-A5	1-5
	B1-B5	11-15
	C1-C5	21-25
	D1-D5	31-35
	E1-E5	41-45
	F1-F5	51-55
	G1-G5	61-65

'emp_length'	<1 year	0
	1 year	1
	2 years	2
	3 years	3
	…	
	9 years	9
	≥10 years	10

5.4.2.5 Target variable

The target variable in this analysis is the loan status. In the raw dataset, 'loan_status' includes the following values: Fully Paid, Current, Charged Off, Late (31-120 days), In Grace Period, Late (16-30 days), and Default. The goal in loan default prediction is to identify patterns that can detect candidates likely to charge off. To streamline the model development process and reduce the number of outcomes in the dataset, only the two most significant and relevant outcomes are retained: Fully Paid and Charged Off. These outcomes are then converted to numerical values, with Fully Paid represented by 0 and Charged Off by 1.

Table 7. Target variable Encoding

Target variable	Data	Replaced data
'loan_status'	Fully Paid	0
	Charged Off	1

5.4.2.6 Multicollinearity

The raw dataset comprises numerous features and data points, which enhances the potential for modeling the relationship between independent variables and the target variable. However, this also raises the risk of skewed or misleading results due to multicollinearity. Multicollinearity occurs when there is an approximate linear relationship between two or more independent variables, indicating that these variables are highly correlated. This condition suggests that collinear independent variables are not genuinely independent [36]. Therefore, it is essential to assess the correlation between the independent variables in the raw dataset. By selecting different types of features instead of multiple features of the same type, we can avoid multicollinearity, which may result in less reliable outcomes. One common method to address multicollinearity is to first identify the collinear independent predictors through the Correlation matrix and subsequently remove one or more of them.

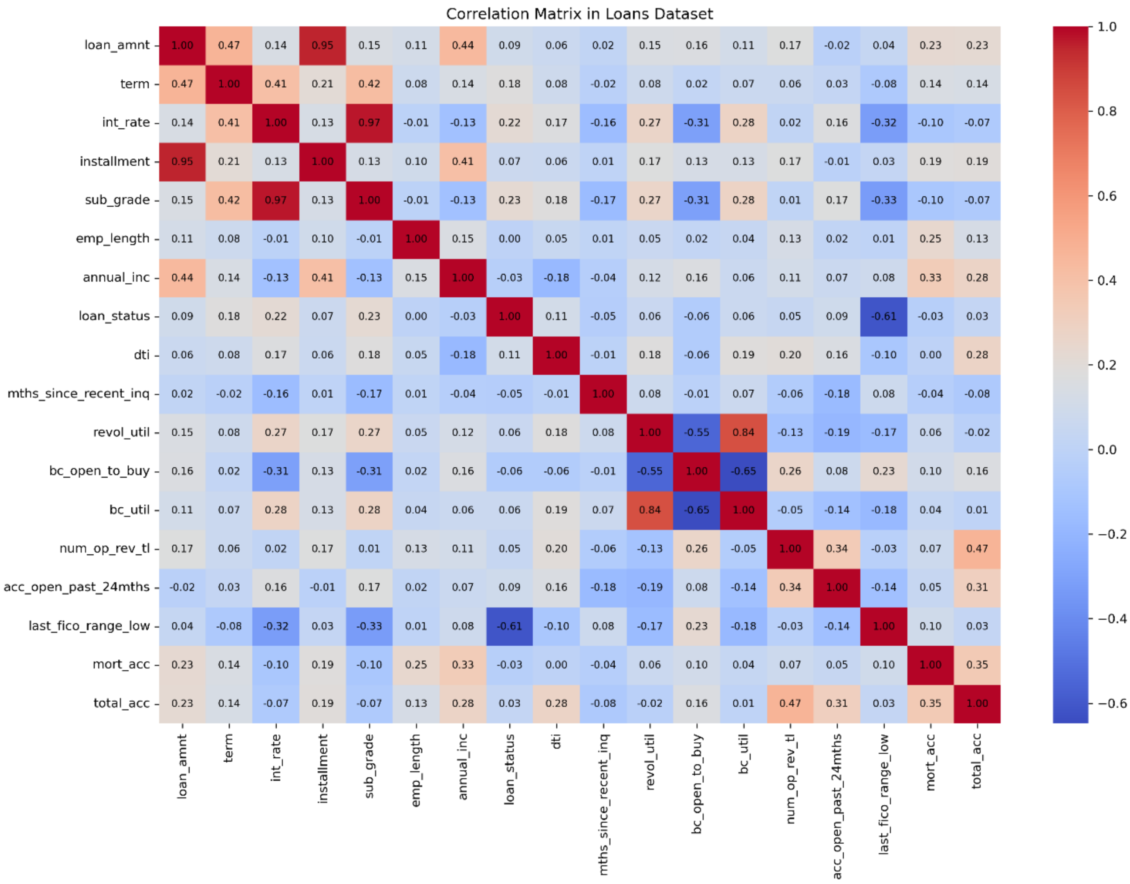


Figure 14. Correlation Matrix - Dataset

In the correlation matrix, 'revol_util' and 'bc_util' features have a correlation coefficient of +0.84 which indicates a strong positive relationship between the two variables. In other words, as one variable increases, the other variable tends to increase also. The correlation coefficient of -0.55 between 'bc_open_to_buy' and 'revol_util' indicates a moderate negative relationship between the variables. This means that as one variable increases, the other variable tends to decrease. Correspondingly, the correlation coefficient of -0.65 between 'bc_open_to_buy' and 'bc_util' denotes a strong negative relationship. Hence, these three features are highly correlated, probably because they contain information related to revolving accounts and bankcards. To avoid multicollinearity issues, 'bc_open_to_buy' and 'bc_util' features are removed from the dataset. Accordingly, the correlation coefficient of +0.97 between 'int_rate' and 'sub_grade' represents an extremely strong positive relationship, because interest rate evidently derives from 'sub_grade', and therefore subgrade is removed from the dataset. For the same reason, installment is removed as it is highly correlated with 'loan_amnt'.

5.5 Model Development

5.5.1 Class Imbalance

The dataset exhibits class imbalance, a condition where the distribution of classes is significantly skewed, with one class vastly outnumbering the other. The class distribution analysis of the 'loan_status' variable reveals that there are 399,511 instances labeled as class 0 (representing non-defaulted loans: Fully Paid) and 70,474 instances labeled as class 1 (representing defaulted loans: Charged Off). This disparity indicates a remarkable class imbalance, with the majority class (non-defaulted loans) accounting for approximately 85% of the dataset, while the minority class (defaulted loans) constitutes only about 15%.

Generally, assessing class imbalance is crucial in machine learning because models trained on datasets with uneven class distributions may exhibit bias toward the majority class, resulting in poor performance in predicting minority classes. In particular, imbalance should be addressed to ensure that the model accurately predicts both classes [10]. In this study, the Synthetic Minority Over-sampling Technique (SMOTE operator) was applied to the training set to increase the representation of the minority class [10], [37]. Additionally, under-sampling the majority class was considered and yielded similar results. Finally, for the evaluation of model performance, metrics such as precision, recall, F1-score and ROC AUC score were used and prioritized over accuracy, providing a comprehensive assessment across both classes [38].

As discussed in the previous chapter, the objective is to use a model that accurately predicts the applicants who will not be able to fully pay their loan back. Each model aims to correctly predict these cases, although this may also result in instances where the model incorrectly predicts a default, even though the borrower fully repays the loan. Thus, each model is primarily evaluated based on the F1 score and subsequently on the recall metric for class 1.0 given that the cost of false negatives is deemed higher in comparison with the cost of false positives [10]. In other words, the models are designed and assessed relying on the assumption that the economic loss of the lending organization due to false negatives prevails over the loss of potential customers (business loss) due to false positives.

5.5.2 Dataset Split

The final step before proceeding with the application of the different models, is to separate the features ('X') from the target variable ('y') in the dataset and then split the dataset into a training set and a testing set. In this study, 70% of the data is allocated for training ('X_train', 'y_train'), while the remaining 30% is reserved for testing the model ('X_test', 'y_test'). The testing set ('y_test') remains an unseen portion of the dataset, which is used for evaluating the model predictions ('y_pred') at the end of the process.

5.5.3 Logistic Regression model

The Logistic Regression model builds a predictive model that establishes a relationship between the independent variables and the target variable using a logistic function. The machine learning model was initialized, trained, and evaluated using the corresponding sklearn.linear_model module in the scikit-learn library in Python with the default parameters. The evaluation metrics are denoted in the tables below.

Table 8. Logistic Regression Default Parameters

Parameter	Default Value	Description
penalty	l2	Regularization term
c	1.0	Inverse of regularization strength
solver	lbfgs	Algorithm to use in the optimization
max_iter	100	Maximum number of iterations

Table 9. Evaluation metrics - Logistic Regression

Metric	Value
Accuracy	0.8945
Precision	0.6006
Recall	0.8837
F1 Score	0.7151
ROC AUC Score	0.8900

The model correctly predicts 89.45% of all instances (accuracy). A percentage of 60.06% of all the positive predictions made by the model are true positives (precision). Additionally, the model correctly identifies 88.37% of the actual positives (recall). The harmonic mean of precision and recall is 71.51% (F1 score) and the model has an area under the ROC curve of 0.89.

Table 10. Classification report - Logistic Regression

Class	Precision	Recall	F1-Score	Support
0.0	0.98	0.90	0.94	119,856
1.0	0.60	0.88	0.72	21,140
Overall Metrics				
Accuracy			0.89	140,996
Macro Avg	0.79	0.89	0.83	140,996
Weighted Avg	0.92	0.89	0.90	140,996

The logistic regression model achieves a high overall accuracy of 89.45%. While the precision for predicting defaults (class 1.0) is moderate at 60.06%, the recall is quite high at 88.37%. This indicates that the model is effective at identifying actual defaults, which is the main objective, though it has a moderate rate of false positives. The F1 score and ROC AUC score further highlight the balanced performance and good discriminative power of the model respectively.

The confusion matrix depicted below provides detailed insights into the predictive capabilities of Logistic Regression across the two classes. There are 2,459 missed detections (false negatives) where the loan was actually charged off but was predicted as non-default. Additionally, there are 12,423 false positives, where the model incorrectly predicted a default for instances where the loan did not actually default. Eventually, these signify that while the Logistic Regression model can effectively identify a significant portion of defaults, the rate of false positives could be optimized more.

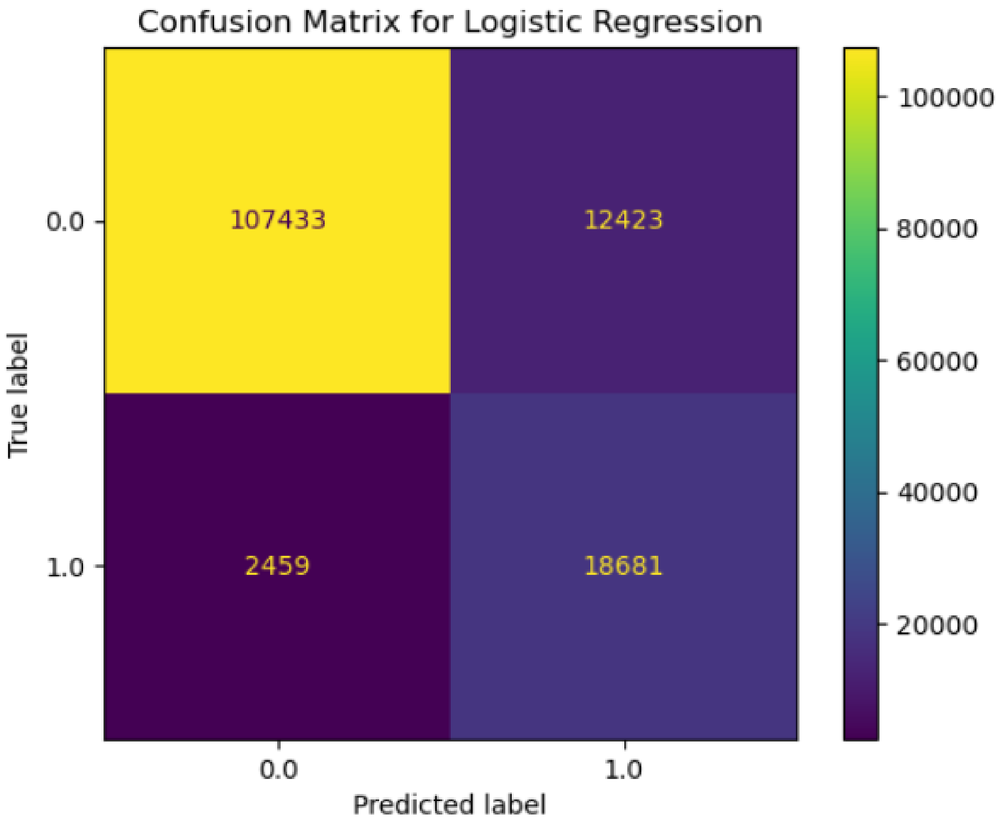


Figure 15. Confusion Matrix - Logistic Regression

5.5.4 Decision Tree model

The Decision Tree classifier builds a model that splits the dataset into subsets based on the value of the features, creating branches until it reaches a leaf node that represents a prediction. The machine learning model provided by the sklearn.tree module in the scikit-learn library in Python was initialized, trained, and evaluated with the default parameters. The assessment metrics are presented in the tables below.

Table 11. Decision Tree Classifier Default Parameters

Parameter	Default Value	Description
criterion	gini	Function to measure the quality of a split
max_depth	none	Maximum depth of the tree
min_samples_split	2	Minimum number of samples to split an internal node
min_samples_leaf	1	Minimum number of samples at a leaf node
max_features	none	Number of features to consider for the best split

Table 12. Evaluation Metrics - Decision Tree

Metric	Value
Accuracy	0.8772
Precision	0.5859
Recall	0.6163
F1 Score	0.6007
ROC AUC Score	0.7697

The model achieves an overall accuracy of 87.72%, correctly predicting 87.72% of all instances. Of the instances predicted as positive, 58.59% are true positives, as indicated by the precision metric. Additionally, the model successfully identifies 61.63% of actual positive cases, as reflected in the recall metric. The harmonic mean of precision and recall, represented by the F1 score, is 60.07%. Furthermore, the model demonstrates an area under the ROC curve (AUC-ROC) of 0.7697.

Table 13. Classification Report - Decision Tree

Class	Precision	Recall	F1-Score	Support
0.0	0.93	0.92	0.93	119,856
1.0	0.59	0.62	0.60	21,140
Overall Metrics				
Accuracy			0.88	140,996
Macro Avg	0.76	0.77	0.76	140,996
Weighted Avg	0.88	0.88	0.88	140,996

Ultimately, the Decision Tree model demonstrates moderate overall performance, correctly predicting 87.72% of all instances. The precision metric indicates that, when the model predicts a default, it is accurate 58.59% of the time, reflecting a moderate rate of false positives. The recall metric for the positive class reveals that the model correctly identifies 61.63% of actual defaults, corresponding to a moderate rate of false negatives. Therefore, the model maintains a balance between identifying defaults and non-defaults, as evidenced by the F1 score.

As illustrated in the confusion matrix below, the model failed to identify 8,112 instances (false negatives) where the loan actually defaulted but was predicted as non-default, and it incorrectly classified 9,207 instances as defaults (false positives). In conclusion, the metrics indicate that the Decision Tree model does not detect a high percentage of loan defaults and tends to overpredict positive cases.

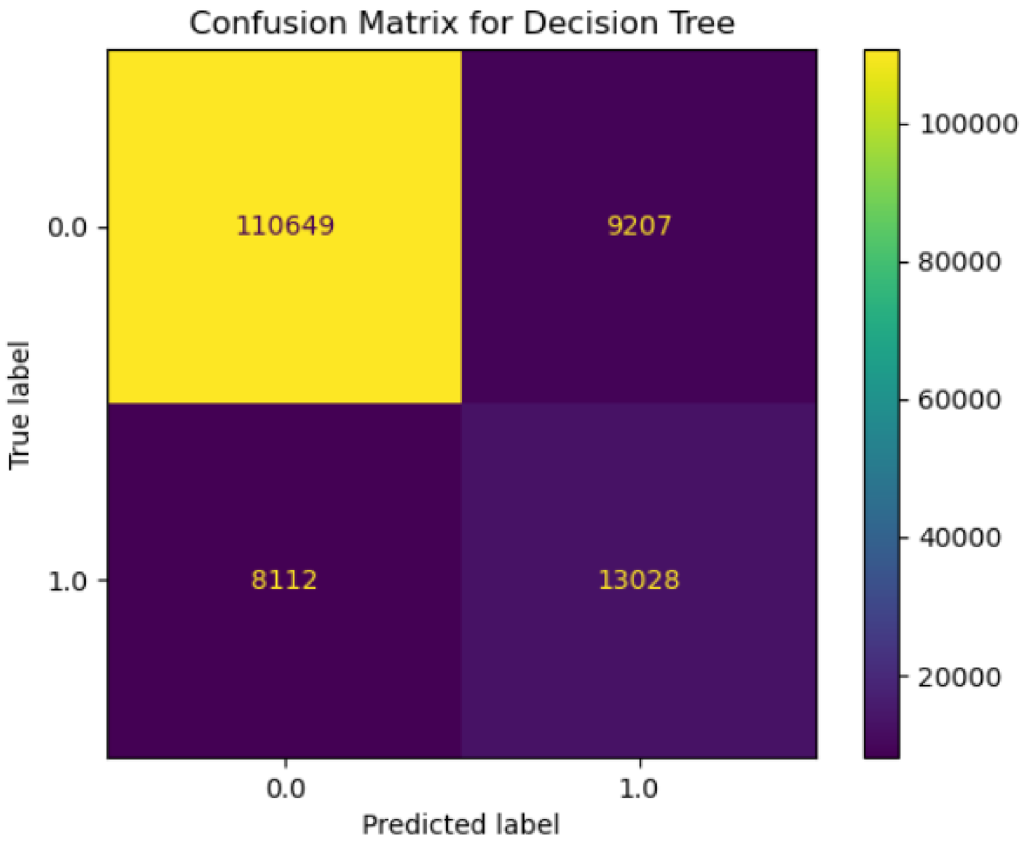


Figure 16. Confusion Matrix - Decision Tree

5.5.5 Gradient Boosting model

The Gradient Boosting classifier (boosting technique) builds an ensemble model of weak learners (decision trees) to improve predictive performance. The machine learning model provided by the sklearn.ensemble module in the scikit-learn library in Python was initialized, trained, and evaluated with the default parameters. The assessment metrics are presented in the tables below.

Table 14. Gradient Boosting Classifier Default Parameters

Parameter	Default Value	Description
loss	deviance	Loss function to be optimized
learning_rate	0.1	Shrinks the contribution of each tree
n_estimators	100	Number of boosting stages
subsample	1.0	Fraction of samples used for fitting the base learners
max_depth	3	Maximum depth of the individual estimators
min_samples_split	2	Minimum number of samples to split an internal node
min_samples_leaf	1	Minimum number of samples at a leaf node

Table 15. Evaluation metrics - Gradient Boosting

Metric	Value
Accuracy	0.9007
Precision	0.6239
Recall	0.8505
F1 Score	0.7198
ROC AUC Score	0.8801

The model correctly predicts 90.07% of all instances (accuracy). Out of all instances predicted as positive, 62.39% are true positives (precision). Moreover, the model correctly identifies 85.05% of the actual positives (recall). The harmonic mean of precision and recall is 71.98% (F1 score) and the model possesses an area under the ROC curve of 0.8801.

Table 16. Classification report - Gradient Boosting

Class	Precision	Recall	F1-Score	Support
0.0	0.97	0.91	0.94	119,856
1.0	0.62	0.85	0.72	21,140
Overall Metrics				
Accuracy			0.90	140,996
Macro Avg	0.80	0.88	0.83	140,996
Weighted Avg	0.92	0.90	0.91	140,996

Conclusively, the Gradient Boosting model performs well overall, making correct predictions for 90.07% of all instances. Additionally, the precision metric indicates that when the model predicts a default, it is correct 62.39% of the time (moderate rate of false positives). The high recall metric for the positive class demonstrates the effectiveness of the model at identifying 85.05% of actual defaults (low rate of false negatives). Hence the model shows a good balance between identifying defaults and non-defaults, also confirmed by the high F1 score.

As shown in the confusion matrix below, the model missed 3,160 instances (false negatives) where the loan actually defaulted but was predicted as non-default and incorrectly predicted default for 10,837 instances (false positives). In conclusion, the metrics suggest that the Gradient Boosting model detects a substantial percentage of loan defaults, but it tends to overpredict such positive cases.

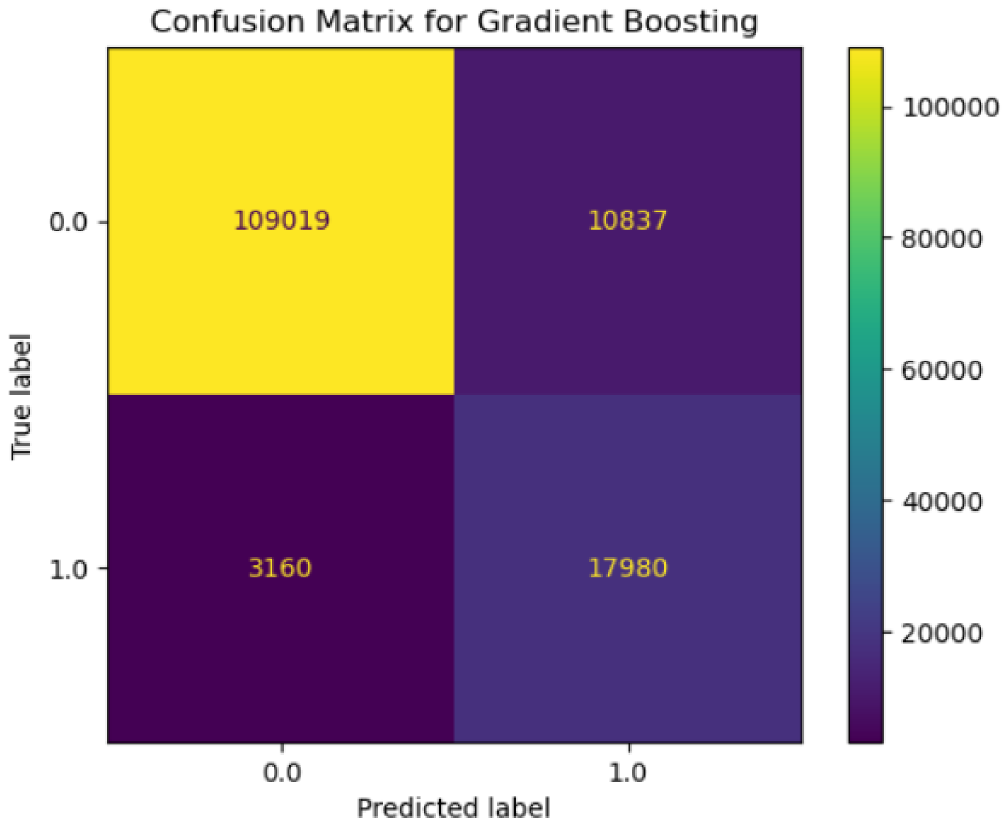


Figure 17. Confusion Matrix - Gradient Boosting

5.5.6 Random Forest model

The Random Forest classifier (bagging technique) constructs an ensemble model by combining multiple decision trees to enhance predictive performance. The machine learning model provided by the sklearn.ensemble module in the scikit-learn library in Python was initialized, trained, and evaluated with the default parameters. The evaluation metrics are presented in the tables below.

Table 17. Random Forest Classifier Default Parameters

Parameter	Default Value	Description
n_estimators	100	Number of trees in the forest
criterion	gini	Function to measure the quality of a split
max_depth	none	Maximum depth of the tree
min_samples_split	2	Minimum number of samples to split an internal node
min_samples_leaf	1	Minimum number of samples at a leaf node
max_features	auto	Number of features to consider for the best split
bootstrap	true	Whether bootstrap samples are used when building trees

Table 18. Evaluation metrics - Random Forest

Metric	Value
Accuracy	0.9056
Precision	0.6460
Recall	0.8199
F1 Score	0.7226
ROC AUC Score	0.8703

The model achieves an overall accuracy of 90.56%, correctly predicting 90.56% of all instances. Of the instances predicted as positive, a percentage of 64.60% are true positives, as indicated by the precision metric. Additionally, the model successfully identifies 81.99% of actual positive cases, as reflected in the recall metric. The harmonic mean of precision and recall, represented by the F1 score, is 72.26%. Furthermore, the model demonstrates an area under the ROC curve of 0.8703.

Table 19. Classification report - Random Forest

Class	Precision	Recall	F1-Score	Support
0.0	0.97	0.92	0.94	119,856
1.0	0.65	0.82	0.72	21,140
Overall Metrics				
Accuracy			0.91	140,996
Macro Avg	0.81	0.87	0.83	140,996
Weighted Avg	0.92	0.91	0.91	140,996

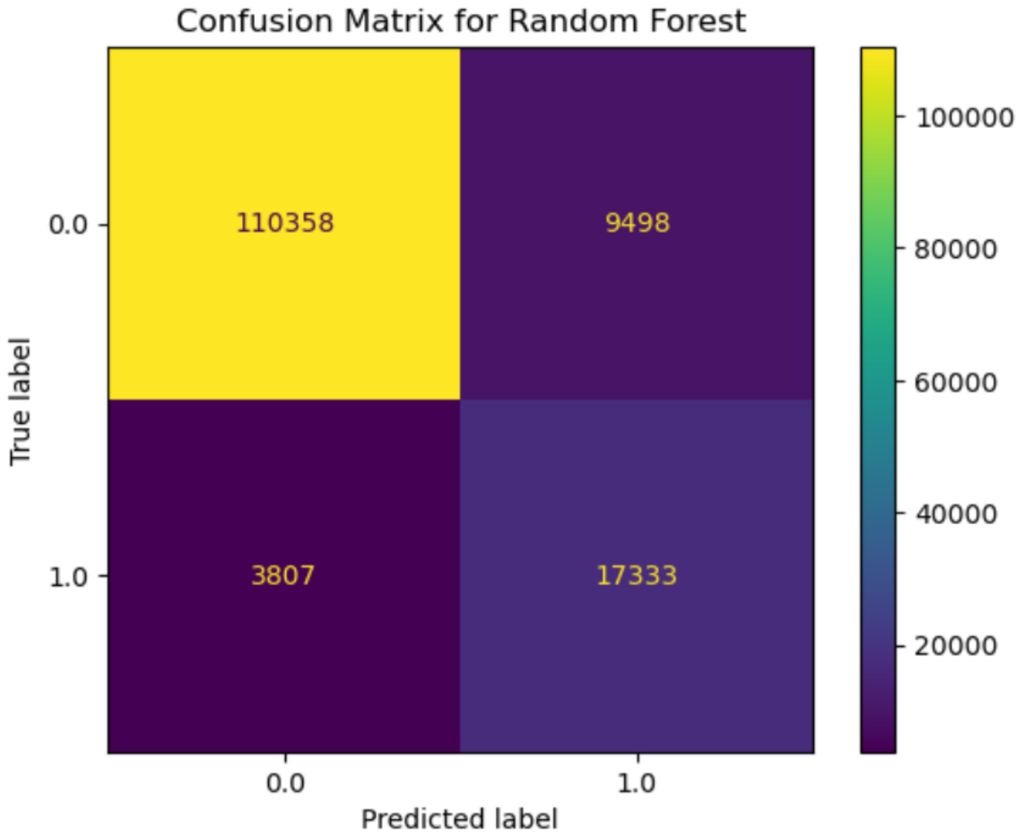


Figure 18. Confusion Matrix - Random Forest

Finally, the evaluation metrics for Random Forest show its robust performance, as it correctly predicts 90.56% of all instances. Additionally, the model demonstrates moderate (64.60%) correctness of positive predictions and high (81.99%) completeness of positive predictions. The performance between precision and recall is balanced as denoted by the F1 score. The ROC AUC score highlights the great ability of the model to effectively distinguish between defaults and non-defaults.

An analysis of the confusion matrix reveals that the model failed to identify 3,807 instances where the loan actually defaulted (false negatives). Furthermore, the model incorrectly predicted defaults for 9,498 instances (false positives). Ultimately, these inaccuracies suggest that the model could be enhanced more to differentiate between default and non-default cases more effectively.

5.5.7 Neural Networks model (MLP Classifier)

The Multi-Layer Perceptron (MLP) neural network, designed to enhance predictive performance through multiple layers of neurons, was initialized, trained, and evaluated using the MLPClassifier from the sklearn.neural_network module in the scikit-learn library in Python with the default parameters. The assessment metrics are presented in the tables below.

Table 20. MLP Classifier Default Parameters

Parameter	Default Value	Description
hidden_layer_sizes	(100,)	Number of neurons in the ith hidden layer
activation	relu	Activation function for the hidden layer
solver	adam	Solver for weight optimization
alpha	0.0001	L2 penalty (regularization term) parameter
learning_rate	constant	Learning rate schedule for weight updates
max_iter	200	Maximum number of iterations

Table 21. Evaluation metrics - MLP

Metric	Value
Accuracy	0.8813
Precision	0.5686
Recall	0.8636
F1 Score	0.6857
ROC AUC Score	0.8740

The model correctly predicts 88.13% of all instances (accuracy). Out of the instances predicted as positive, 56.86% are true positives (precision). Moreover, the model correctly identifies 86.36% of the actual positives (recall). The harmonic mean of precision and recall is 68.57% (F1 score) and the model has an area under the ROC curve of 0.8740.

Table 22. Classification report - MLP

Class	Precision	Recall	F1-Score	Support
0.0	0.97	0.88	0.93	119,856
1.0	0.57	0.86	0.69	21,140
Overall Metrics				
Accuracy			0.88	140,996
Macro Avg	0.77	0.87	0.81	140,996
Weighted Avg	0.91	0.88	0.89	140,996

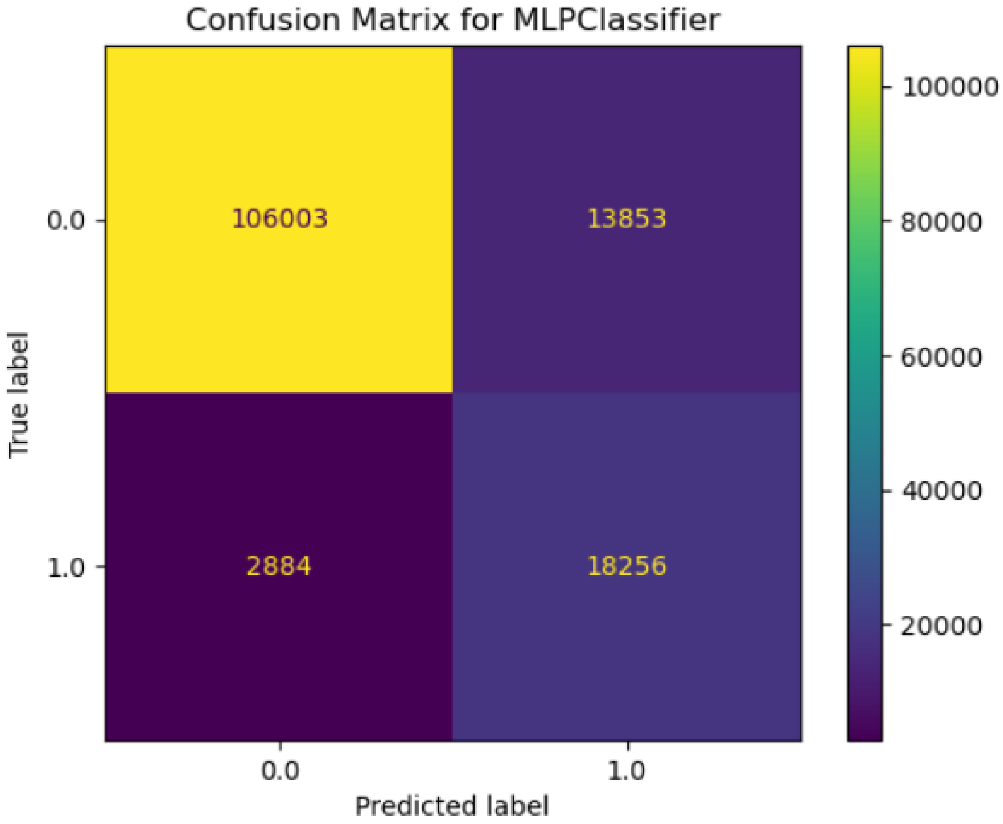


Figure 19. Confusion Matrix - MLP

The MLP neural network model demonstrates a solid performance overall, achieving an accuracy of 88.13%, meaning it correctly classifies 88.13% of all instances. The precision score of 56.86% indicates that when the model predicts a default, it is correct slightly more than half the time, demonstrating a moderate rate of false positives. The recall metric shows that the model identifies 86.36% of actual defaults, which implies a low rate of false negatives. The F1 score confirms the balanced performance between precision and recall.

Examining the confusion matrix reveals that the model missed 2,884 instances (false negatives) where the loan actually defaulted, and incorrectly predicted default for 13,853 instances (false positives). In conclusion, these highlight that while the MLP model effectively identifies a substantial percentage of defaults, there is room for improvement especially in reducing false positives for even better performance.

5.5.8 Comparison of the ML models

The different ML models are ranked in descending order, primarily based on the F1 score and secondarily on the recall metric. The Random Forest model excelled by demonstrating the highest F1 score, emphasizing its robustness in accurately detecting charged off loans, while at the same time maintaining precision at acceptable levels. Gradient Boosting closely followed, indicating vigorous predictive ability in identifying defaulted loans. Logistic Regression ranked next, consistently performing well across F1 score and recall. The neural network model (MLP) placed fourth, showing acceptable results, while the Decision Tree model ranked last, demonstrating moderate performance. These findings underscore that Random Forest and Gradient Boosting are particularly effective for loan default prediction, emphasizing their ability to balance between precisely identifying defaults and minimizing false positives.

Table 23. Evaluation metrics of various ML models

	Model	F1 score	Recall	Accuracy	Precision	ROC AUC score
1.	Random Forest	0.7226	0.8199	0.9056	0.6460	0.8703
2.	Gradient Boosting	0.7198	0.8505	0.9007	0.6239	0.8801
3.	Logistic Regression	0.7151	0.8837	0.8945	0.6006	0.8900
4.	MLP (Neural Network)	0.6857	0.8636	0.8813	0.5686	0.8740
5.	Decision Tree	0.6007	0.6163	0.8772	0.5859	0.7697

6. Conclusions

This study addresses the problem of loan default prediction, a significant challenge in the banking sector. The raw dataset for this research was preprocessed by choosing appropriate features, handling missing values, and managing outliers to ensure the data's integrity and suitability for model application.

Effective preprocessing, including feature selection and handling missing values and outliers, played a crucial role in improving model performance. Key features such as loan amount, interest rate, debt-to-income ratio, credit score, number of accounts opened, revolving utilization rate and employment length were identified as significant predictors of loan default. Techniques like SMOTE were crucial in addressing class imbalance, which is a common issue in loan default datasets. This ensured that the models were better equipped to predict defaults accurately.

Multiple machine learning methods for classification were applied to predict loan defaults, including logistic regression, decision tree, gradient boosting, random forest, and neural networks. These models were evaluated using several metrics, such as accuracy, precision, recall, F1 score, and ROC curve. While accuracy is a commonly used metric, this study emphasized the importance of F1 score and recall, due to class imbalance and significance of false negatives. This comprehensive evaluation provided deeper insights into each model’s performance. Given the higher cost of false negatives compared to false positives, each model was primarily evaluated based on the F1 score and subsequently on the recall metric. The results indicate that classification algorithms can significantly enhance the detection of loan defaults. Specifically, the random forest classifier showed the highest F1 score, indicating a balanced performance in predicting loan defaults while minimizing false negatives. Logistic regression, while simpler, also provided strong predictive power, making it a viable option for institutions with limited computational resources.

7. Recommendations for Future Research

Outlined below are several options for future research building upon the information and results presented in this study:

i. Excessive Feature Engineering: A future study may consider applying more extensive feature engineering techniques, such as Recursive Feature Elimination (RFE), Lasso regularization, and Principal Component Analysis (PCA). This will help identify any discrepancies in the features selected through Exploratory Data Analysis (EDA) in this study and potentially improve model performance.

ii. Hyperparameter Tuning: To further enhance model performance, hyperparameter tuning using grid search, random search, or Bayesian optimization can be conducted. These techniques can optimize model parameters and provide a quantitative measure of performance improvement over the default parameters used in this study.

iii. Weighted Ensemble Models: Developing a weighted ensemble model by combining existing models from this study with corresponding weights can enhance predictive performance. Algorithms such as stacking, blending, and voting classifiers can be utilized to achieve this.

iv. Testing on Multiple Datasets: Evaluating the various models on multiple datasets, including more complex datasets, will provide a better understanding of their robustness and generalizability.

v. Advanced Neural Networks and Deep Learning: Future research could explore more advanced neural network architectures and deep learning models using frameworks like TensorFlow and PyTorch. Due to high computational power and time constraints, these models were not explored in this study but hold significant potential for improving predictive accuracy.

vi. Integration of Alternative Data Sources: Investigating the integration of alternative data sources, such as social media activity, mobile phone usage, and other non-traditional data, could provide additional predictive power for loan default prediction.

8. References

[1] M. Ranjan, K. Barot, V. Khairnar, V. Rawal, A. Pimpalgaonkar, S. Saxena and A. Sattar, "Python: Empowering Data Science Applications and Research," *Journal of Operating Systems Development & Trends,* 2023.

[2] W. Mckinney, "Pandas: a Foundational Python Library for Data Analysis and Statistics," *Python High Performance Science Computer,* 2011.

[3] G. Mahalaxmi and A. Srinivas, "A Short Review of Python Libraries and Data Science Tools," 2023.

[4] M. Choetkiertikul, A. Hoonlor, C. Ragkhitwetsagul, S. Pongpaichet, T. Sunetnanta, T. Settewong, V. Jiravatvanich and U. Kaewpichai, "Mining the Characteristics of Jupyter Notebooks in Data Science Projects," 2023.

[5] H. Han and R. Jarvis, "Fintech Innovation: Review and Future Research Directions.," *International Journal of Banking, Finance and Insurance Technologies,* 2021.

[6] T. Riasanow, R. Floetgen, D. Soto Setzke, M. Böhm and H. Krcmar, "The Generic Ecosystem and Innovation Patterns of the Digital Transformation in the Financial Industry", 2018.

[7] P. Tran, T. Le and N. Phan, "Digital Transformation of the Banking Industry in Developing Countries Article history: Digital Transformation of the Banking Industry in Developing Countries," *International Journal of Professional Business Review,* 2023.

[8] H. P. Josyula, "The Role of Fintech in Shaping the Future of Banking Services," *The International Journal of Interdisciplinary Organizational Studies,* 2021.

[9] A. Ziouache and M. Bouteraa, "Descriptive Approach of Neo-Banking System: Conception, Challenges and Global Practices," *International Journal of Business and Technology Management,* 2023.

[10] P. Gashi, "Loan Default Prediction Model," 2023.

[11] R. Achary and C. Shelke, "Fraud Detection in Banking Transactions Using Machine Learning," in *2023 International Conference on Intelligent and Innovative Technologies in Computing, Electrical and Electronics (IITCEE)*, 2023.

[12] R. Tao, C.-W. Su, Y. Xiao, K. Dai and F. Khalid, "Robo advisors, algorithmic trading and investment management: Wonders of fourth industrial revolution in financial markets," *Technological Forecasting and Social Change,* 2020.

[13] J. Peng, K. Lee and G. Ingersoll, "An Introduction to Logistic Regression Analysis and Reporting. Journal of Educational Research," *The Journal of Educational Research,* 2002.

[14] B. Jijo and A. Abdulazeez, "Classification Based on Decision Tree Algorithm for Machine Learning," *Journal of Applied Science and Technology Trends,* 2021.

[15] M. Habib, A. Majumder, R. Nandi, F. Ahmed and M. S. Uddin, "A Comparative Study of Classifiers in the Context of Papaya Disease Recognition," 2020.

[16] Y. Tian, Y. Shi and X. Liu, "Recent advances on support vector machines research," *Technological and Economic Development of Economy,* 2012.

[17] K. Taunk, S. De, S. Verma and A. Swetapadma, "A Brief Review of Nearest Neighbor Algorithm for Learning and Classification," in *2019 International Conference on Intelligent Computing and Control Systems (ICCS)*, 2019.

[18] A. Wibawa, A. Kurniawan, D. Murti, R. P. Adiperkasa, S. Putra, S. Kurniawan and Y. Nugraha, "Naïve Bayes Classifier for Journal Quartile Classification," *International Journal of Recent Contributions from Engineering, Science & IT (iJES),* 2019.

[19] A. Natekin and A. Knoll, "Gradient boosting machines, a tutorial," *Frontiers in neurorobotics,* 2013.

[20] D. Maulud and A. Abdulazeez, "A Review on Linear Regression Comprehensive in Machine Learning. Journal of Applied Science and Technology Trends.," *Journal of Applied Science and Technology Trends,* 2020.

[21] "Laerd Statistics," [Online]. Available: https://statistics.laerd.com/spss-tutorials/linear-regression-using-spss-statistics.php.

[22] A. Dasgupta, Y. Sun, I. König, J. Bailey-Wilson and J. Malley, "Brief review of regression-based and machine learning methods in genetic epidemiology: the Genetic Analysis Workshop 17 experience," *Genet Epidemiol.,* 2011.

[23] M. Awad and R. Khanna, Support Vector Regression, Efficient Learning Machines, 2015.

[24] B. Wundervald, "Bayesian Linear Regression," National University of Ireland, Maynooth, 2019.

[25] A. Munusamy and D. Sridharan, "Analysis of Clustering Algorithms in Machine Learning for Healthcare Data," *Springer,* 2020.

[26] M. Bakoben, A. Bellotti and N. Adams, "Identification of Credit Risk Based on Cluster Analysis of Account Behaviours," *Journal of the Operational Research Society,* 2017.

[27] C. Győrödi, R. Gyorodi, P. Dr, S. Ing and H. Stefan, "A Comparative Study of Association Rules Mining Algorithms," *SACI 2004, 1st Romanian-Hungarian Joint Symposium on Applied Computational Intelligence,* 2004.

[28] T. Tanatorn and P. Loetwiphut, "Association rule mining framework for financial credit-risk analysis in peer-to-peer lending platforms," *Science, Engineering and Health studies,* 2023.

[29] "Receiver operating characteristic curve (WIKIPEDIA)," [Online]. Available: https://en.wikipedia.org/wiki/Receiver_operating_characteristic.

[30] Y. Dong and J. Peng, "Principled missing data methods for researchers," *SpringerPlus,* 2013.

[31] S. Kwak and J. Kim, "Statistical data preparation: Management of missing values and outliers," *Korean Journal of Anesthesiology,* 2017.

[32] D. Monhor and S. Takemoto, "Understanding the concept of outlier and its relevance to the assessment of data quality: Probabilistic background theory," 2005.

[33] D. Divya and D. Sasidhar Babu, "Methods to detect different types of outliers," in *2016 International Conference on Data Mining and Advanced Computing*, 2016.

[34] R. Gottfredson and H. Joo, "Best-Practice Recommendations for Defining, Identifying, and Handling Outliers," 2013.

[35] M. Galarnyk, "Understanding Boxplots," [Online]. Available: https://www.kdnuggets.com/2019/11/understanding-boxplots.html.

[36] J. Chan, S. Leow, K. Bea, W. K. Cheng, S. W. Phoong, Z.-W. Hong and Y.-L. Chen, "Mitigating the Multicollinearity Problem and Its Machine Learning Approach: A Review," *Mathematics,* 2022.

[37] A. Ali, S. M. Shamsuddin and A. Ralescu, "Classification with class imbalance problem: A review," 2015.

[38] A. Luque, A. Carrasco, A. Martín Gómez and A. de Las Heras, "The impact of class imbalance in classification performance metrics based on the binary confusion matrix," 2019.

9. Appendices

9.1 Appendix 1: “Raw dataset preparation” code snippet

The code used for the Raw dataset preparation in this thesis can be found in the file 'LoanDefaultRawDatasetPrep.ipynb' in the following GitHub repository:
https://github.com/MariosTheodoridis/Loan-Default-Prediction.git

9.2 Appendix 2: “Loan Default model development” code snippet

The code used for developing the various Loan Default Models in this thesis can be found in the file 'LoanDefaultModelDev.ipynb' in the following GitHub repository:
https://github.com/MariosTheodoridis/Loan-Default-Prediction.git