Department of *Corporate Finance*
Chair of *Digital Finance*

Exploring the Role of AI in Banking Risk Assessment

SUPERVISOR

Prof. Paolo Bonolis

CO-SUPERVISOR

Prof. Gaetano Casertano

CANDIDATE

Matteo Mascherucci (771061)

ACADEMIC YEAR
2023-2024

List of Figures

List of Abbreviations

AI:	Artificial Intelligence
ANN:	Artificial Neural Networks
BoW:	Bag of Words
CFPB:	Consumer Financial Protection Bureau
DL:	Deep Learning
FDIC:	Federal Deposit Insurance Corporation
FTE:	Full-Time or Equivalent
GAN:	Generative Adversarial Network
KDE:	Kernel Density Estimation
KNN:	K-Nearest Neighbors
LDA:	Linear Discriminant Analysis
LLM:	Large Language Models
LSTM:	Long Short-Term Memory Network
ML:	Machine Learning
MLP:	Multi-Layer Perceptron
NLP:	Natural Language Processing
PDF:	Probability Density Function
RAROC:	Risk Adjusted Return on Capital
RFE:	Recursive Feature Elimination
RL:	Reinforcement Learning
RNN:	Recurrent Neural Network
RPA:	Robotic Process Automation
SMOTE:	Synthetic Minority Over-sampling Technique
SVC:	Support Vector Classifier
SVM:	Support Vector Machine
TF-IDF:	Term Frequency-Inverse Document Frequency
VaR:	Value at Risk
XAI:	Explainable AI
XGB:	Extreme Gradient Boosting

Introduction

As the financial institutions navigate an increasingly complex regulatory and economic environment, traditional mechanisms of risk management are called into debate. The subject of this thesis is a discussion on how Artificial Intelligence can be applied by financial institutions in their different processes for risk management. The study in question will be conducted through deep analysis of the development made within the field of AI technologies, studying the use of these methods in risk evaluation and comparing such superior methodologies with traditional credit-scoring models.

The thesis follows a structure that, to begin, gives a conceptual overview of what are considered the major types of risk, such as market risk, credit risk, liquidity risk, and operational risk; and how the traditional models usually manage these risks. In this context it will firstly be treated the general processes of risk management before focusing more narrowly on their specific applications within the banking sector. After setting this informational foundation, the paper will proceed to introduce the concepts of Artificial Intelligence and Machine Learning, explaining the differences between them and including a short historical background of their development.

The second section of the thesis describes the development of risk management in banking, aiming to illustrate how these practices have evolved through time.
Subsequently, the attention will be focused on the detailed examination of ML algorithms, covering the spectrum of existing types of algorithm.
These include: Supervised Learning, Unsupervised Learning, and Deep Learning, among others. The discussion will extend to the programming aspects of these algorithms, providing a technical backdrop to their functional applications.
The following section develop what concrete benefits ML can bring to financial institutions. Great emphasis will be placed on the application of ML in the management of several kinds of financial risks, focusing on credit risk. Indeed, ML algorithms have played a truly significant role in this respect. Specifically, it will be highlighted how this technology is employed in credit scoring, considering the three key models of Logistic Regression, Decision Tree, and Random Forest. Each model's methodology and effectiveness will be properly addressed, as well as what each separately brings to credit risk assessment. Finally, this section will be concluded by looking at different AI technologies being put into the processes of risk management, such as Expert Systems, Natural Language Processing, and Generative AI. A clear description will be given about how inventions of this nature contribute to augmenting the analytical capabilities and decision-making processes within the industry.

The third part of this essay will focus on analyzing a dataset of 1,000 individuals and 20 different variables to determine the most suitable machine learning model for predicting creditworthiness. A detailed examination of the dataset’s variables will be conducted to understand their impact on credit risk assessment. Following this, preprocessing steps such as feature scaling, handling missing values, and balancing the dataset will be applied to ensure a fair evaluation of the models.

Thirteen different models will then be tested, including Logistic Regression, Decision Tree, Random Forest, and their variations, along with more advanced models. The implementation and testing of these algorithms will be carried out using Python. Each model will be assessed based on key performance metrics to evaluate its ability to distinguish between creditworthy and non-creditworthy individuals.
By systematically comparing these models, this study aims to identify the most effective approach for credit scoring, striking a balance between predictive power, interpretability, and efficiency.

Lastly, the fourth and last section will focus on analyzing the current level of integration of Artificial Intelligence in banking, specifically in risk management and compliance. It will examine the current state of AI adoption across financial institutions, highlighting differences between banks, fintech firms, and other sectors in leveraging AI for regulatory compliance and fraud detection. A key aspect will be the role of data quality and governance, as inconsistencies and fragmentation remain major barriers to effective AI implementation. The chapter will also explore how AI is currently used in data analysis, automation, and predictive risk modeling, assessing its impact on efficiency and decision-making. While AI offers significant advantages, concerns around data privacy, transparency, and over-reliance on automation persist. The discussion will address these challenges and the necessary measures, such as AI governance frameworks and fairness testing, to ensure responsible adoption.

Chapter 1: Comprehensive Overview on Risk and Artificial Intelligence

1.1 Different Types of Risk

Risk, in its most basic form is "*the possibility that something unpleasant or unwelcome will happen*"[1].

In finance, the concept of risk is primarily associated with the uncertainty and the potential for financial loss that come with investment decisions. In general, the higher is the level of risk taken, the greater should be the expected return in order to justify the increased probability of losses. This relationship reflects the fundamental trade-off between risk and return that guides investor behavior and market dynamics. Investors typically demand higher returns as compensation for the higher uncertainty and increased likelihood of financial loss associated with riskier investments.**(1)**

To quantify this risk, finance professionals often analyze the historical volatility of asset returns, utilizing standard deviation as a key metric. Historical volatility measures how much asset returns deviate from their average over a certain period, providing a clear picture of how much the value of the asset fluctuates. A higher standard deviation indicates greater variability, which not only signifies a higher risk level but also suggests the potential for higher returns. This relationship between high risk and high potential returns is fundamental in finance, helping investors understand the risks associated with different investment options and guiding them in making informed decisions that align with their risk tolerance and investment goals. **(2)**

Turning the attention to the banking sector, it is important to focus on the distinct types of risk banks face:

1. **Market Risk:** Refers to the potential for financial losses due to adverse movements in market prices and rates, impacting the values of positions held by financial institutions. It arises from fluctuations in various market parameters known as "risk factors," which include interest rates, equity indexes, foreign exchange rates, commodity prices, inflation indexes, and credit spreads. The extent of market risk depends on the time required to liquidate assets; longer periods generally see wider price movements, particularly for less liquid assets traded over-the-counter. Market risk is predominantly associated with a bank's trading book, which contains financial assets held for trading purposes rather than long-term investment. **(3,4,5)**
2. **Credit Risk:** Refers to the potential losses a bank faces if borrowers fail to meet their obligations. This risk primarily arises from the bank's lending and treasury activities, where there's a risk of borrowers defaulting on their loans or the value of the bank's investments diminishing. Credit risk also includes concerns such as rating downgrades, as well as settlement and pre-settlement risks. Settlement risk refers to the failure of a counterparty to meet their obligations at the time of transaction completion. Pre-settlement risk, on the other hand, concerns the

[1] "Risk," Oxford English Dictionary Online, Oxford University Press, 2024, www.oed.com/view/Entry/164805.

possibility of a counterparty defaulting in the period from the start of a transaction until its settlement. **(5,6)**

3. **Liquidity Risk:** Refers to the potential loss that arises when a bank cannot fulfill its payment obligations on time due to a mismatch in the maturities of its assets and liabilities. This risk can manifest in two main ways: funding liquidity risk, which is the difficulty in obtaining new financing, and market liquidity risk, which involves challenges in converting assets into cash without significant losses.
   Liquidity issues are often linked to other types of financial risks, such as significant market or credit losses, which can undermine a bank's creditworthiness and lead to decreased lending or even rapid withdrawals by depositors. **(5,7)**
4. **Operational Risk:** Defined as the risk of loss due to failures in internal processes, people, and systems, or from external events. Can arise from issues like malfunctions in information and reporting systems, or failures in internal monitoring and corrective procedures. This can include legal risks or reputational risks linked to the bank’s operations. Unlike credit, market, and liquidity risks, operational risk is less understood and is the most challenging to measure, manage, and monitor effectively. **(5,8)**

1.2 Traditional Methods of Risk Management

Risk management is a fundamental practice for any organization aiming to safeguard its operations and enhance its market position. Effective risk management is crucial for business success and stability, as it helps companies strike a balance between potential gains and possible losses. Businesses inherently take risks to generate profits, but excessive risk can lead to failure. Modern risk management must also adapt to new challenges brought on by globalization, digital technology advancements, and environmental changes like climate change, which is considered a "threat multiplier" by experts. By effectively managing risks, companies can seize growth opportunities while safeguarding against potential threats. **(5)**

There are five basic techniques of effective risk management:

1. **Avoidance:** While not always possible, avoiding risk is a primary tactic. For instance, delaying vehicle use during severe weather can prevent accidents, and avoiding storage in flood-prone areas can reduce water damage claims.
2. **Retention:** In certain cases, it may be cheaper for an organization to retain the risk, rather than transferring it, especially when the cost of mitigating it exceeds the potential loss. For example, small businesses might choose to self-insure against health issues of their employees rather than purchasing very expensive health insurance packages from insurance companies. This approach allows the business to save on high insurance premiums while retaining the risk of having to pay out significant medical costs in the event of less frequent, more serious health issues among employees.
3. **Spreading**: To minimize impacts, risks can be spread across different areas. In the realm of investments, a common practice for spreading risk is through diversification of the investment portfolio. Rather than investing a large amount of capital into a single stock or sector, a company or individual investor might

distribute their investments across various asset classes such as stocks, bonds, real estate, and international markets. This way, the potential portfolio impact of a downturn in a single market is reduced.

4. **Loss Prevention and Reduction:** This involves taking specific actions to decrease the likelihood and mitigate the impact of risks. An example can be found in the manufacturing industry, where companies often implement regular safety training and equipment maintenance. This approach helps to prevent accidents and machinery breakdowns, reducing the risk of production interruptions and enhancing workplace safety.
5. **Transferring:** Involves shifting the potential loss to another party, typically through insurance or contracts. A common example of risk transfer is when investment firms buy put options to hedge against stock market declines. By purchasing these options, the firm limits its potential losses, effectively shifting the risk to the seller of the put options in exchange for a premium. **(11)**

Narrowing the focus on the banking industry, it's important to consider that the various risks that banks face can have a significant impact on many people's life. For this reason, it's crucial to implement preventive measures through a structured risk management framework. This framework, involving personnel, methodologies, and technology, is essential for aligning organizational goals with risk tolerance and values. An effective risk management strategy addresses legal, contractual, internal, and ethical standards and stays current with technology-related regulations. By focusing on risk management and dedicating the necessary resources, banks can protect themselves from uncertainties, reduce costs, and enhance their chances of ongoing operational success.
This process in banking is typically composed of six steps: **(9)**

1. **Identification:** This involves recognizing the types of financial risks the bank faces, understanding their sources, and why they are potentially harmful to the institution.
2. **Assessment and Analysis:** Here, the bank assesses the likelihood and severity of each identified risk. This analysis helps in prioritizing risks based on their potential impact, guiding the bank on where to focus its risk management efforts.
3. **Mitigation:** This step includes developing and applying specific policies and procedures aimed at reducing the likelihood of risks materializing into actual threats, as well as limiting the damage if these risks were to materialize.
4. **Monitoring:** Continuous monitoring is essential to assess the effectiveness of the risk management strategies put in place. This includes tracking the ongoing performance of risk controls and staying informed about new and emerging risks. Monitoring ensures that the bank's risk management strategies are current and effective.
5. **Cooperation:** Risk management requires a coordinated approach across various departments within the bank. This cooperation helps ensure that risk management strategies are comprehensive and integrated throughout the organization, enhancing the collective response to potential threats.
6. **Reporting:** Regular documentation and reporting are critical for evaluating the effectiveness of the risk management process. Reports help keep track of the bank's risk management activities and provide insights into how the bank's risk profile is evolving over time **(10)**

1.3 Introduction to AI and Machine Learning

Artificial Intelligence (AI) and Machine Learning (ML) are two linked branches of computer science that however have quite different applications.
Artificial Intelligence is a wide set of different technologies that enable computers to behave and "think" as a human, enabling them to execute tasks that normally only humans could perform including understanding natural languages, patterns, images, and making decisions. This represent a new frontier in the development of smart machines. In fact, AI is now built into many everyday technologies, from smart appliances to voice-activated assistants like Siri or Alexa, enhancing user interactions and automating routine processes.
Business AI applications exploit advanced techniques like natural language processing (NLP), which is the ability to recognize and understand human language as it's spoken or written, and computer vision, that enables computer to identify and understand objects and people in images and videos. These technologies enable companies to automate tasks, streamline decision-making, and facilitate interactions with customers using chat-bots.
Machine Learning is a subdomain of Artificial Intelligence which is primarily concerned with the development of algorithms and statistical models that allow systems to learn from data and make decisions improving their performance overtime.
ML is one of the components through which Artificial Intelligence achieves its capabilities. This type of models automatically acquires knowledge and detects the pattern from the abundant data to make decisions more accurately. The most advanced ML algorithms involve deep learning, which uses neural networks that simulates the human brain functions to analyze data, recognizes complex patterns and makes predictions all independently. **(12)**

1.4 Historical Evolution of AI

AI Early Days (1950s-1970s):
The journey began in 1950, when Alan Turing published "*Computing machinery and intelligence*", presenting the concept of machines thinking and solving problems like humans and also introducing the Turing test as a measure of machine intelligence. This test assumes that a machine could be considered intelligent if it is able to imitate human conversation indistinguishably.
The formal initiation of AI as a scientific discipline occurred at the Dartmouth Conference in 1956, where John McCarthy coined the term "artificial intelligence". This conference set the mission and scope for AI research, asserting that every aspect of learning or intelligence could potentially be so precisely described that a machine could simulate it. The following years saw AI progress in waves, initially flourishing with advancements like the Logic Theorist program and later experiencing setbacks during periods known as "AI winters," where funding and interest waned due to unmet expectations.

Symbolic AI (1980s-1990s):

In the 1980s, the AI research saw a significant funding boost from Japan's Fifth Generation Computer Project. Also, the focus shifted towards symbolic AI and expert systems, which aimed to encapsulate human expert knowledge into software that could be distributed across personal computers. These usually had two major components: a knowledge base that held facts, rules, and relationships concerning some particular subject area, and an inference engine which is a software that applies logical rules to the knowledge base to deduct new informations. An example of that can be found in IBM's Deep Blue machine that for the first time beat chess champion Garry Kasparov, showcasing the achieved levels of capabilities of this technology and also renewing interest and funding to AI research.

AI Current Days (2000s-2020s):

Entering the 21st century, the explosion of data and advancements in computational power enabled AI to integrate deeply into various sectors such as healthcare, finance, and entertainment. The advent of "big data"[2] has allowed AI to operate on an unprecedented scale, learning through massive datasets rather than solely algorithmic innovation.

Today, the integration of AI in everyday life only continues to increase, with technologies such as automated customer service and improvements in natural language processing leading to even more sophisticated applications like real-time translation and autonomous vehicles. Increased integration of AI in everyday life is what encompasses the future of AI, bringing out important ethical and policy considerations that have to be overcome if AI's full potential is ever going to be harnessed responsibly. **(13,14)**

[2] Big data: extremely large data sets that may be analyzed computationally to reveal patterns, trends, and associations, especially relating to human behavior and interactions. Definition adapted from Oxford English Dictionary Online, Oxford University Press, 2024, www.oed.com/view/Entry/164805..

Chapter 2: AI Techniques Used in Risk Assessment

As discussed in the previous chapter, risk assessment is crucial in the banking activity. It involves evaluating the potential risks associated with bank operations, from issuing loans to managing investments.

Over time risk management has evolved from methods based on personal knowledge to sophisticated systems that utilize advanced technologies subject to strict regulatory frameworks.

The history of risk management began with the early days of banking, when operations were very small and geographically localized, and decisions in large part were based upon a banker's intuitive sense about his clients' creditworthiness.

However, with industrialization, banks began to grow in number, and things became quite complicated, bringing to the realization of more formalized ways of risk management. It resulted in the development of credit risk management techniques that based themselves on financial positions and collateral to mitigate the risk of non-repayment.

For example, the Banking Act of 1933[3], enacted during the Great Depression, leading to the beginning of regulatory oversight of banking.

Then, in successive decades, ideas like the risk-adjusted return on capital (RAROC) in the 1970s and Basel Accords[4], beginning in 1988, laid down international standards around capital adequacy and risk management. The global financial crisis in 2008 has in itself become a very significant junction. It exposed many weaknesses in the current practices of risk management and brought to many reforms to enhance the transparency and stability in the banking system.

---

[3] **The Banking Act of 1933** is also known as the Glass-Steagall Act. It was an act whose purpose was to rescue the financial order during the Great Depression. Its key provisions were: (1) Separation of commercial and investment banking to prevent high-risk activities connected with stock market speculations; (2) Establishment of the Federal Deposit Insurance Corporation, or FDIC, which will provide federal insurance for depositors' accounts in order to prevent bank runs; and (3) Several reforms on banking with an aim to be more transparent and produce the banking industry in a more responsible way. These were fundamental steps intended to institute public confidence in the financial system and aid in framing the future path of modern finance regulation.

[4] **The Basel Accords**, developed by the Basel Committee on Banking Supervision, consist of four separate frameworks known as Basel I, II, III and IV, each building on the last to address emerging financial challenges. Basel I, introduced in 1988, focused on capital adequacy by setting minimum capital requirements for banks. Basel II, implemented in 2004, expanded on this by introducing refined risk and capital management requirements, focusing on three pillars: minimum capital requirements, supervisory review, and market discipline.

This includes the Dodd-Frank Act[5] in the United States and the Basel III[6] and Basel IV[7] frameworks.
Parallel to these developments, technological progress has changed the face of risk management dramatically. In fact, it is due to breakthroughs in data analytics, artificial intelligence, and machine learning that it is now possible for banks to manage their risks very accurately by providing tools to predict potential pitfalls and automate highly complex decision-making processes. The chapter develops how these technologies are not simply improving existing capabilities but also paving the way for new methods and strategies that may define the future of banking. **(17)**

2.1 Machine Learning Models

As mentioned before, Machine Learning is a subset of Artificial Intelligence that focuses on the development of models that enable computers to learn from data without explicit programming. In fact, unlike traditional programming that relies on strict rules and decision-making structures (like IF-THEN commands), Machine Learning uses algorithms to analyze data, learn from it, and decide or predict outcomes autonomously. Essentially, ML systems improve over time by optimizing their performance as they process more data. By exposing the Machine Learning model to vast amounts of labeled training data, the system learns how to independently identify and classify information. This capability makes ML particularly useful in the fields like data mining, automating customer service, and and improving various business processes through insights gained from extensive data analysis.
To get a good understanding of how Machine Learning works, it is important to explore the different machine learning methods and algorithms, which are basically sets of rules that machines use to make decisions. These are:

- Supervised Learning;
- Unsupervised Learning
- Semi-Supervised Learning

---

[5] Enacted in July 2010, the **Dodd-Frank Wall Street Reform** and Consumer Protection Act aimed to enhance financial stability, increase transparency, and prevent future financial crises. It established crucial oversight bodies like the Consumer Financial Protection Bureau (CFPB) and introduced measures to limit risky practices in financial institutions.

[6] **Basel III**, developed in response to the financial crisis of 2008, further tightened capital requirements, introduced new regulatory requirements for bank liquidity and leverage, and emphasized the need for banks to maintain sufficient capital buffers to absorb financial shocks.

[7] **Basel IV**, adopted in January 2023, is a set of revisions aimed at strengthen the existing Basel III framework. Key enhancements include a standardized approach for credit risk, a 72.5% output floor to maintain minimum capital levels, simplified operational risk calculations, and revised market risk guidelines. Basel IV also introduces a leverage ratio buffer for major banks to bolster financial stability.

Supervised Learning:
This is the most used method in ML. Here the algorithm is presented with a set of training data where every training sample has an input and a desired output. This teaches the algorithm to make “educated guesses” based on learned patterns whenever presented with unseen data.
This approach is called "supervised" because the data must be manually associated with the correct answers in order to guide the algorithm in identifying and understanding patterns and relationships within the data.
There are two tasks that can be performed by a supervised learning algorithm: Regression and Classification.
In **Regression tasks**, the output value is a continuous number, meaning that the outcome is a number within a certain range. An example of this can be a model trained in the prediction of house prices. Here, the algorithm is trained on a dataset with features like location, size, and number of rooms, along with historical selling prices. The goal is for the model to learn from these features and predict the price of a new house based on its characteristics.
The **Classification tasks** the output value is a category with a finite number of options.
For instance, to create an automatic spam detection system, it’s necessary to present the algorithm with different examples of spam mails and with some examples of mails that are certainly not spam in order to educate it to efficiently recognize the junk mails. In this case the output options are “spam” or “non-spam”.

Unsupervised Learning:
Here, the algorithms analyze unlabeled data to find patterns and insights without any pre-determined labels or outcomes. This type of models can recognize structures and patterns autonomously using techniques like clustering, which is the grouping of similar data together. Unsupervised learning is particularly useful for exploratory data analysis. For example in the marketing field this type of algorithms are able to sort different costumers based on their age or spending and then propose different promotions or products to each group.

Semi-Supervised Learning:
This type of learning, as the name suggests, mixes Supervised and Unsupervised learning. In fact, the training data pool is formed by a majority of unlabeled data and a small portion of labeled data. The latter are used to guide the prediction of the algorithm on the unlabeled data, often offering a better accuracy than the normal Supervised models. This method is a great choice for businesses that have to deal with a large amount of data but limited in resources for extensive labeling. This efficiency in managing data, in fact, not only speeds up the machine learning process but also significantly reduces operational costs. An example of Semi-Supervised learning can be observed in the organization of digital photo libraries. Here, a small subset of photos is manually labeled with specific tags such as "beach" or “urban". Then, the model uses the labeled photos to learn what features are associated with each category. Subsequently, it applies this knowledge to classify a larger set of unlabeled photos, efficiently categorizing them without the need for extensive manual tagging.

In addition to these basic models there are two more types of learning that have unique characteristics associated with them. The first one is the Reinforcement Learning and the other one is Deep Learning.

Reinforcement Learning:
RL is different from the previously discussed models because it doesn't rely on a data set to work. Instead, it focuses on "teaching" a computer program, or "agent", how to make the best possible decisions in a given setting, through a process of trial and error. This means that as the agent interacts with its environment, it learns from its actions by experiencing the consequences of those actions, gradually identifying which ones yield the greatest rewards. This method is particularly useful in the fields of robotics of gaming where the connection between actions and outcomes is explicit. For instance, in video games, the game's scoring system provides immediate feedback on the effectiveness of different strategies, making it an ideal testing ground for refining RL algorithms. This continuous interaction allows the algorithm to improve its decision-making over time, enhancing its ability to perform tasks or achieve objectives autonomously.

Deep Learning:
Deep Learning models can be supervised, semi-supervised and also unsupervised (or any combination of these). They are very sophisticated algorithms that power extensive systems for tech giants like Amazon, Google or Microsoft, enabling impressive features, from self-driving cars to intelligent virtual assistants. Deep Learning is built using Artificial Neural Networks (ANN), which are systems created to operate in the same way as the human brain, as the name suggests.
These networks are composed of multiple layers of interconnected neurons that work together in harmony, allowing for complex, layered processing of information.
When deep learning models are fed input data, which could be anything from images and text to video and audio, they process this data layer by layer. This structure enables the models to gradually enhance their learning, much like a human brain that grows and learns from new experiences over time. Deep learning is especially prevalent in applications such as image and speech recognition, as well as Natural Language Processing (NLP), where it often surpasses traditional machine learning techniques. These models excel in solving complicated problems and handling large volumes of data. However, they typically need vast amounts of training data and considerable training time to perform effectively, reflecting the scale and complexity of their capabilities. **(15)**

2.2 How Machine Learning Algorithms Are Developed

The code implementation of a Machine Learning Algorithm follows five distinct phases:

1. **Data Collection:** This is the foundational step in the creation of a ML algorithm. This step involves gathering the necessary data that the algorithm will learn from. The quality and quantity of data collected have a direct impact onto the performance of the resulting model. Such data may be obtained from public data sets, company databases, data generated by sensors, or data scraping from the internet.
2. **Data Processing:** Once the data have been collected they usually aren't ready to use immediately, they need to be refined to transform the row data into a clear and usable data set. This process includes several sub-steps such as:
   - *Clearing:* Implies removing or correcting any missing or corrupt data;
   - *Transformation:* Normalizing or scaling data to a specific range or format, making it easier for the model to learn;
   - *Feature Selection:* In this part, the most relevant features (variables) to use in the model are chosen. This step is done to reduce complexity and improve performance;
   - *Encoding:* That means converting all the categorical data (age, sex, income, education level) into numeric format in order to be processed by the ML algorithm.
3. **Model Training:** Once the data set is ready, a ML model is selected according to the specific problem to solve (the most used models are Linear Regression, Logistic Regression, or Decision Trees). When the right model has been selected, the proper training begins by feeding the algorithm the prepared training data set. Here the model will learn to map inputs (features) to outputs (targets) by adjusting its parameters.
4. **Model Evaluation:** After the model has been trained, it is evaluated using another data set that has not been used during the training phase. This is a vey important step in the development because it is possible to verify the performance of the algorithm. Performance is assessed using different metrics such as accuracy, precision, recall, F1 Score which is a metric that balances precision and recall by calculating their harmonic mean, making it particularly useful in situations with imbalanced datasets.
5. **Model Deployment:** At this stage the model is ready to be used to solve real-world problems. **(16)**

2.3 Machine Learning Applications in the Risk Management Process

This section will provide an analysis of actual applications of Machine Learning in the risk management process. Especially in the fields of Credit and Market risk, where ML based models find the most widespread application.

2.3.1 Application to Credit Risk

Credit risk refers to the potential potential economic loss when a counterparty fails to meet its contractual duties, such as on-time payments of interest or principal, or from an escalated risk of default during the duration of the transaction. Historically, financial institutions have utilized traditional regression techniques like linear, logit, and probit models to assess credit risk. However, there has been a shift towards incorporating machine learning into these practices, driven by the limitations observed in conventional methods. In fact, ML models, with their proficiency in processing unstructured data, offer significant enhancements.

The need for advanced machine learning techniques becomes more apparent in complex financial environments, such as the credit default swap market, which involves numerous uncertainties in predicting defaults and estimating potential losses. **(18)**

Traditional credit scoring frameworks depend on historical credit data and set rules, rejecting applications that don't align with established criteria.

Conversely, machine learning (ML) credit scoring models utilize both standard data (like overall credit scores) and non-traditional data (such as rent and mobile payments) to discern patterns in borrower behavior. These patterns help predict different credit risks.

ML-based credit scoring models offer a comprehensive view of an applicant's financial habits, revealing details that traditional methods may overlook. These models also reduce biases related to an applicant's age, gender, profession, employment status, or ethnicity.

There are several differences between traditional and ML-based credit scoring models:

1. **Data Sources:** Traditional models mainly rely on the data provided by major reporting agencies, such as Experian or Equifax. These models, including FICO Score and VantageScore, use this historical data to create a tailored score for each borrower, which banks use to make lending decisions.

   In regions where credit scoring is less common, banks typically employ rule-based systems for loan origination, using a limited array of data sources such as transaction data, lending history, and employment records, typically with 10 to 20 assessment criteria.

   ML-based models incorporate a broader array of data types and sources, including:
   - Rent and utility payments
   - Cash flow trends
   - Checking account details
   - Mobile data and mobile payments
   - Telecom and internet data

   Using these alternative data types and sources ML models can achieve a greater level of precision.
2. **Decision Speed:** Traditional lending processes are manual and time-consuming, with home loans taking 35 to 40 days on average to close. By contrast, FinTech lenders use automation and predictive analytics to process applications about 20% faster than traditional methods. They leverage Open Banking and financial APIs for quick data access and verification, streamlining the loan origination process and enhancing decision-making efficiency.

3. **Default Rates:** Banks set specific targets for non-performing loans (NPLs) relative to total assets, typically aiming for a ratio around 4%. During economic downturns, many banks halt lending to credit-thin consumers.
   Machine learning allows for more precise loss predictions on a per-case basis. For example, Chinese digital banks like WeBank, MYBank, and XWbank manage to keep their NPLs at about 1% despite issuing millions of loans each year.
4. **Analysis Methods:** Traditional credit scoring is based on statistical, rule-based evaluations heavily reliant on credit history data and often influenced by subjective judgments during applicant interviews. This has led to discriminatory practices, with minority groups in the U.S. facing higher interest rates and greater loan rejection rates.
   In contrast, ML models analyze a wider spectrum of data points using algorithms that deliver objective, data-driven decisions. For example, Argentine business lender Mercado Libre uses around 2,400 behavioral variables to score each applicant, with different factors weighted according to their relevance.

The main Machine Learning models used in credit scoring are:

1. **Logistic Regression:** Among various statistical tools used in credit scoring, machine learning logistic regression models surely occupy one of the leading positions as predictive models for binary outcomes, such as if a borrower will default on a loan. This model type has many applications to credit scoring because it can provide the probabilities of a default of a given applicant, which is crucial in assessing the risk of granting loans.
   Logistic regression models take a set of diversified predictor variables to estimate these probabilities. These may range from factors relating to credit history, to the income levels, employment status, debt-to-income ratios, and everything else that would be relevant data. Each factor enters with its weight in the model; these weights, or coefficients, are refined during the training of the model. Historical data is used for training, where the outcome, which is already known as “default” or “non-default”, allows the model to tweak its parameters through techniques like maximum likelihood estimation to reduce predictive errors. Once trained, the model can apply these parameters to new data yielding a probability score between 0 and 1 at the output for each new loan application. A decision threshold is applied to these scores in order to arrive at lending decisions. Normally, a score above 0.5 may mean default and thus suggests a rejection, though the threshold may be adjusted depending of the type of financial institution granting the loan.
   Logistic regression has some very important advantages in credit scoring because it is an interpretable model. It is quantifiable with regard to the contribution of each of the predictor variables, and this can be very much related directly to how that particular variable influences the probability of a borrower defaulting. This helps the lenders justify their decisions in accordance with financial regulations.
2. **Decision Tree:** Credit scoring decision trees represent a unique and intuitive method of classification of borrowers as credit worthy or not. This model divides the dataset into branches to make predictions, creating a tree-like structure of decisions. Each node in the tree represents a decision point, and the paths from these nodes are the values of the predictor variables, that can include aspects like

income, credit history, and employment status. Decision trees classify borrowers by splitting the data at each node based on simple rule-based decisions, which are determined during the tree's construction. For example, a node might split borrowers into those with incomes above or below a certain threshold. This process continues recursively, creating a comprehensive map of decisions that lead to either a default or non-default outcome. Generally speaking, training a decision tree means deciding on which variables to split and at what threshold. This is usually done in a manner that maximizes the homogeneity of resultant nodes, meaning that each branch of the tree is as pure as possible with respect to the outcome variable. The end result is a model that partitions the data into leaf nodes, each representing a specific segment of the dataset with a high uniformity of outcomes, either default or non-default. Decision trees are particularly valued in credit scoring for their straightforward interpretability. Each decision in the tree provides clear and actionable insights, which can be easily communicated and understood, not just by the analysts who build the model but also by those who implement and rely on it for making lending decisions. Moreover, the visual aspect of decision trees helps in quickly understanding the key variables and how different combinations of inputs lead to a decision. However decision trees do suffer from overfitting, especially complex ones that have a very large number of branches and leaves. Overfitting is a situation in which a model would be too closely fitted to the limited data on which it is trained, where its performance would degrade on new data. To counter this, different techniques include pruning, that is removing parts of the tree that don't contribute to added power, or setting a minimum number of samples per leaf are used. These approaches simplify the tree and improve its generalization.

3. **Random Forest:** This type of model increases the predictive accuracy and stability beyond what can be provided by a single decision tree. An ensemble approach is used in this model, where the outputs from multiple decision trees are aggregated to provide a single, more robust and accurate prediction. Each of the trees in the forest decides based on a slightly different subset of data and takes a random subset of features at each split, hence diversifying the decision paths and predictions. Random forests begin by creating multiple decision trees during the preparation phase, growing each tree on a random sample of the data drawn with replacement (bootstrapping). In addition, at each node of each tree that is built, only a random subset of the features is considered for the split, which reinforces the generalization capability of the model by penalizing the dependencies between each tree in the ensemble. Once the forest is constructed, it makes predictions by having each tree in the ensemble vote on the outcome, and the most common outcome (again, default or non-default) among all the trees is chosen as the final prediction. This voting mechanism inherently reduces the risk of overfitting, a common problem in single decision trees, making random forests a more reliable and accurate model for predicting complex outcomes like creditworthiness. The strength in credit scoring with random forests is the ability of the algorithm to handle high-dimensional data for input variables and their interactions without the need for extensive preprocessing or feature selection. This aspect of the technique renders it very useful in cases where the variable interdependencies are not very transparent. It follows that the random forests also offer measures of feature importance that can

be used to extract knowledge about which variables are most predictive of defaults, thus helping financial organizations enhance their credit scoring processes. Despite their many advantages, random forests can be computationally intensive to train, especially for huge datasets, and their predictions are not as interpretable as those of a single decision tree due to the inherent complexity of the ensemble. However, their superior performance in terms of accuracy and their robustness against overfitting often outweigh these drawbacks, making them a popular choice among data scientists for developing sophisticated credit scoring systems. **(19)**

2.3.2 Application to Market Risk

Traditional risk management models, such as Value at Risk (VaR), have been longstanding staples in the industry. These models provide a static view of the condition prevailing and it assumes the market conditions to be constant over time. However, the reality of the financial market is somewhat different from stability, it is consistently the play of a host of factors that includes geopolitical developments, shifts in the economic indicators, and changes in investor sentiment. Despite their widespread use, many traditional risk models have difficulty capturing the involved nature of current financial instruments, such as complex derivatives. These instruments feature complex relationships and dynamics that require more adaptive analytical approaches, using data-driven insights. Machine learning methods are turning into strong tools by which these challenges are tackled and that further extend the capabilities of market risk models. Machine learning, especially through its advanced implementations such as deep learning neural networks, offers significant advantages. These can handle huge volumes of data and disclose complex patterns and dependencies obscure for both human analytics and conventional statistical modeling. For instance, machine learning-based models, such as recurrent neural networks (RNNs), are very performant in analyzing time series. They can identify patterns in historical market volatility and are able to use this information to predict, in very close approximation, what market behaviors are going to be exhibited shortly. This is in contrast to the conventional models that often rely on historical data in a very static fashion and rely on set assumptions. Another important benefit is the ability of machine learning models to work in real time. Traditional risk models typically rely on old data and cannot adjust quickly to new information, so they are usually inefficient in functioning with fast-changing markets. In contrast, ML models operate by constantly analyzing incoming streams of data and adjusting their predictions and risk assessments in real time. This is especially important when trading in environments with high stakes, such as in the derivative markets, where the small inter-millisecond reaction to market changes can mean the difference between significant profit or loss. Moreover, machine learning algorithms are very efficient in ensuring that many data sources are continuously monitored. They cover market sentiment analyses, real-time news updates, enormous amounts of trading data, thereby providing an awfully comprehensive and real-time view of the market conditions and associated risks to the trader and risk manager. However, the integration of machine learning into market risk assessment also poses a number of challenges. These are very advanced models that require huge piles of data

during the learning process and require a lot of computation. In other words, they need powerful computing infrastructure and a well-trained team of data scientists. Another concerning problem associated with the interpretability of machine learning models is that sometimes the complex calculations of the model may result in the so-called "black box" effect, meaning it is not clear how inputs get transformed into outputs. This can result in lack of transparency, which is what is usually required to be complied with by regulatory bodies, in which the comprehension and explanation of decisions is of the essence. To overcome these challenges, financial institutions are increasingly investing in data infrastructure and talent acquisition. They are also exploring advanced techniques-like explainable AI, (XAI) which will make the inner work of machine learning models more transparent and understandable. **(20)**

2.4 Other Artificial Intelligence Models

Other than traditional ML models, there are many other AI tools that can help financial institutions to identify, assess and mitigate risk. Among there there are: Expert Systems; Natural Language Processing (NPL) and Generative AI.

2.4.1 Expert Systems

An expert system is a rather complex computer program that employs artificial intelligence to simulate some of the human experts' decision-making abilities, who are currently well informed in a given field. The systems do not try to replace human experts but only increase the user's ability. The concept of expert systems dates back to the 1970s, first formulated by the Stanford University professor and creator of the Stanford Knowledge Systems Laboratory, Edward Feigenbaum. During that period, he envisioned the transition from simple data processing to “knowledge processing”, a concept highlighting the potential of computers to tackle complex problems due to advancements in processor technology and computer architectures. Expert systems function by integrating machine learning and AI to simulate the judgment and behaviors of domain experts, enhancing their ability to resolve issues as they gain more experience. They consolidate and utilize knowledge in a database, integrating it with an inference or rules engine that applies this stored knowledge to real-world applications. These systems employ forward chaining, where facts are examined and future events are predicted, and backward chaining, where causes of certain events are inferred, for example, diagnosing diseases from symptoms. Expert system development and maintenance, most commonly is referred to as “knowledge engineering”, include the processes of making sure the system has been adequately informed to provide an effective solution to a problem. This process involves the use of several methodologies for knowledge representation to maintain the information within the system.
The main components of an expert system include: the knowledge base, which is the source of information and includes the module for acquiring external knowledge; the inference engine, which extracts and applies information from the knowledge base to solve problems based on a set of rules; and the user interface, allowing the user to

interact with the system to solve their queries. Together, these components make expert systems invaluable in fields where specialized knowledge must be applied and providing an essential tool for decision-making and problem-solving.
Expert systems significantly enhance the risk management process within financial institutions by automating the analysis of complex data sets to identify, assess, and mitigate potential financial risks. These systems are good at synthesizing diverse information from market trends, credit histories, transaction records, and customer behaviors to highlight anomalies that may indicate fraud, credit risk, or market manipulation. Expert systems enable financial firms to have much more granular insight into their risk factors and proactively address those issues before they cause significant financial losses. As an example, during trading operations, these can analyze the trading patterns from the past and present market conditions, warning officers among traders and risk managers of future market risk scenarios. Moreover, these systems ensure adherence to regulatory compliance by continuously monitoring financial activities against a backdrop of changing regulations. They can flag transactions that deviate from established norms, thereby helping institutions avoid penalties and reputational damage. Overall, expert systems not only streamline risk management processes in financial institutions but also reinforce the decision-making framework, resulting in more robust financial operations. **(21)**

2.4.2 Natural Language Processing (NLP)

NLP is an important field of artificial intelligence that that involves the development of methods to improve the ability of computers to interact effectively with human language. It thus focuses on the study and development of operational procedures that enable computers to understand, interpret, and generate human language in useful ways, covering all activities from text data analysis and processing to the derivation and extraction of meaningful insights. NLP is applied in many different ways in the areas of text mining, speech recognition, machine translation, sentiment analysis, and many more. The preprocessing of text itself starts by converting raw text into a more structured form that machines can analyze. In this, the text is broken down into smaller units, such as words or tokens; unnecessary elements like tags and special characters are removed to reduce the noise of the text; and stemming or lemmatization techniques are performed to consolidate words to their base forms. After preprocessing, feature extraction changes the text into numerical representations. To perform this operation, the most commonly used techniques include the “Bag of Words” (BoW), which represents text in terms of word frequency vectors, and Term Frequency-Inverse Document Frequency (TF-IDF), which refines BoW, where word frequencies are normalized with respect to the document-wide importance of words, hence amplifying the important words but shrinking the frequent ones
With the foundational data prepared, various machine learning algorithms are employed for further analysis. Similarity algorithms help in implementing clustering or comparing text, classification algorithms analyze sentiment and categorize, while advanced techniques like Recurrent Neural Networks and Transformers execute sequential data processing tasks to understand long-distance dependencies.

Financial institutions, use NLP to strengthen predictive analytics in fraud detection and, by extension, simplify the process of finding out potential threats and enhancing their security. NLP analyzes enormous volumes of unstructured textual data such as customer e-mails, transactional data, social media posts, news articles, and many more for trends and patterns indicative of fraudulent behavior and potential risk exposure. For fraud detection, NLP techniques are used to scrutinize communication and transactional records for anomalies or irregular patterns that deviate from the norm, flagging suspicious activities for further investigation. These could include unusual financial transactions or atypical changes in customer communication that may suggest identity theft or account takeover attempts. Moreover, NLP-driven systems assess risk by monitoring public sentiment and market reactions through news outlets and financial reports, providing insights that inform risk mitigation strategies. These systems can dynamically adapt risk models based on real-time data, allowing financial institutions to respond more agilely to potential threats. This integration of NLP not only bolsters the security frameworks but also enhances compliance with regulatory requirements by ensuring continuous monitoring and reporting. **(22)**

2.4.3 Generative AI

Generative AI is revolutionary technology that creates a whole new way to create content in text form, image, and voice. It uses the power of deep learning models combined with large-scale language models to generate original content. This makes generative AI a transformative force across many industries.

Large Language Models work by analyzing the probability distributions of word sequences with the aim of predicting subsequent words in a sentence. This prediction is not hinged on strict grammatical objections but rather on mimicking the way that humans construct sentences. This method allows for the generation of fluent, contextually appropriate language.

Deep Learning on the other hand, utilizes a series of artificial neural networks to analyze a vast amount of data, allowing the system to interpret complex patterns and make decisions accordingly. This significantly enhances the system's capability to generate high-quality, realistic content.

The most common types of generative AI model are:

1. **Generative Adversarial Networks:** This involves two neural networks (the generator and discriminator), both of which are engaged in a competitive process. The generator produces data while the discriminator assesses its authenticity, progressively compelling the generator to generate realistic output. GANs have wide-ranging applications in image generation and the creation of artistic content.
2. **Variational Auto-encoders:** These operate on the principle of probabilistic modeling to find and learn latent distributions of data for tasks such as image generation or compression.
3. **Recurrent Neural Networks and Long Short-Term Memory Networks:** RNNs were designed for sequential data, such as text and time-series analysis, but usually do not solve long-term dependencies within the data very well. LSTMs are an advanced form of RNNs and hence overcome these issues; empirical studies have

shown the efficiency of LSTMs in challenging natural language processing applications requiring extended context.

4. **Generative Pre-trained Transformers:** These are more recent developments in generative AI and draw on the pre-training of the transformer architecture on large text databases. The GPT models are good at generating coherent and contextually relevant text and find their places integrally within a range of applications that comprise automated chatbots, sophisticated content generation, and translation. **(23)**

In the context of risk assessment, generative AI technologies, can analyze large volumes of unstructured data from different sources for patterns and relationships that may elude traditional observation methods. This capability allows financial institutions for early recognition of upcoming risks that could otherwise go unnoticed until they pose significant threats.

Moreover, generative AI can simulate potential risk scenarios derived from real-world data inputs. This facilitates a more robust and comprehensive analysis, providing deeper insights into potential vulnerabilities that might impact the banking sector. The dynamic monitoring capabilities of generative AI stand out as it continuously learns from new data, enabling banks to keep their risk assessments up-to-date in real-time. This is crucial in quickly adapting to changes in risk exposure and allows for preemptive measures to mitigate risks effectively. Additionally, generative AI supports the development and implementation of risk mitigation strategies by offering insights into potential process improvements and pinpointing gaps in control mechanisms.

From an operational perspective, embedding generative AI into the risk management framework will automate these labor-intensive tasks and by doing so it enhances efficiency while reducing costs. This way this technology ensures that risk management practices are not only reactive but also proactive, adapting swiftly to new regulations and changing market conditions. This adaptability is critical for maintaining compliance and securing a competitive edge in the fast-evolving financial landscape.

However, the integration of generative AI comes with challenges. It requires a high level of data quality and volume for effective training of AI models; it needs complex model interpretability for transparency in decision making, equaling robust measures that would validate and ensure the reliability of AI applications. Ethical considerations and privacy concerns, more particularly on sensitive data, should be managed radically in order to uphold the in-trust and regulatory standards. **(24)**

Chapter 3: How to Develop a Machine Learning Algorithm for Credit Scoring

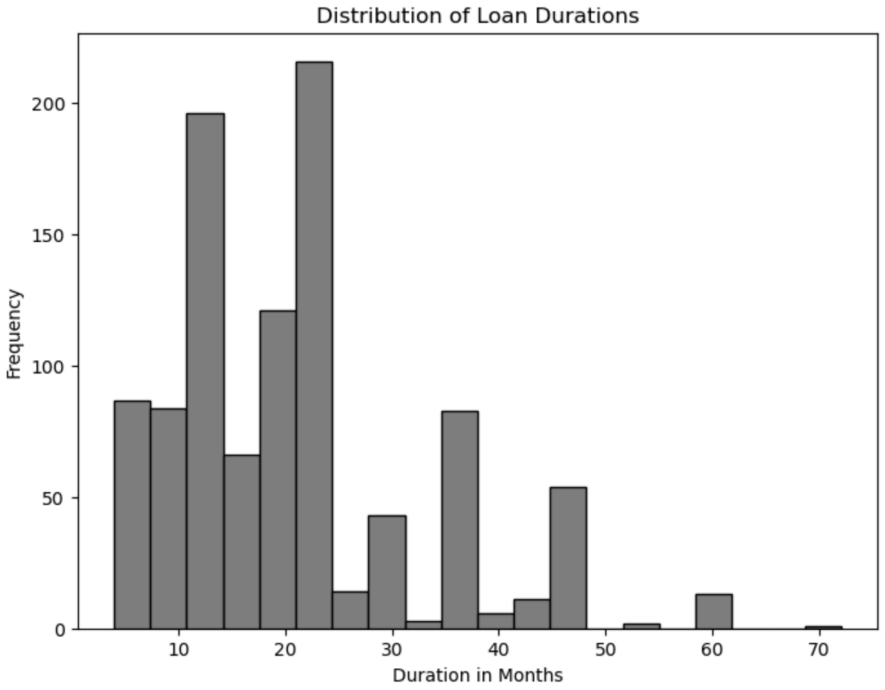
This chapter presents a step-by-step methodology for developing different types machine learning algorithms with credit-scoring applications. This is done by training and testing many different models typically used for this purpose, including Logistic Regression, Decision Tree, Random Forest, and their variations, along with more advanced models. Each will be discussed in detail to illustrate its special capabilities and suitability with regard to credit rating evaluation. Accordingly, the development process is strictly divided into four major phases of operation: data collection, data processing, model training, and model evaluation. To ensure comparability between the models, the first two phases, involving the collection and the refining of the data, are standardized across all the algorithmic approaches.

3.1 Data Set

The fist step in the creation of a Machine Learning algorithm is the collection of the data used for the training and testing of the model.
The data set , sourced from Kaggle, was created by Hans Hofmann in 2024 and consists of 1,000 subjects with 21 different attributes considered. Kaggle is an online platform that provides datasets, code and notebooks for data science and machine learning. The link to the dataset can be found in the references.
There are 20 independent variables used for the training of the algorithms and one target variable which is the variable that the models will attempt to predict based on the input features.
Among the 20 independent variables there are 8 numerical variables and 12 categorical variables.
The numerical are: Duration in Months, Credit amount, Account Balance, Installment rate in percentage of disposable income, Number of years in the current residence, Age, Number of existing credits at the bank, Number of people being liable to provide maintenance for.
The categorical are: Credit history, Purpose of the credit, Status of savings account/ bonds, Present employment(years), Personal status, Presence of co-applicants or guarantors, Property', 'Other installment plans, Housing, Job, Possessing a telephone, Home or foreign worker, Account Balance.
Let’s now discuss all the single variables in detail and how they interact with each other.

3.1.1 Detailed Variable Analysis

1. **Duration in Months:** This is a numerical indicator indicating the number of months in which the borrower will have to repay the loan.
   The loan duration spans from as short as 4 months to as long as 72 months. The average (mean) duration across all loans is approximately 20.9 months, with a standard deviation of about 12.06 months, indicating a wide variance in loan terms. The most typical (median) loan duration is 18 months. A detailed inspection reveals that 25% of the loans have durations of 12 months or fewer and 75% have durations of 24 months or fewer.



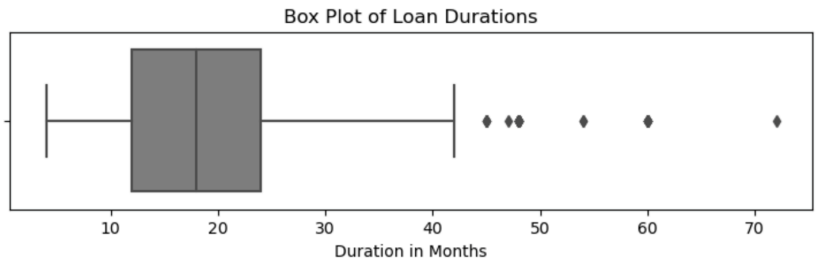


Figure 1. Illustrates the distribution of the loan duration in the dataset ranging from 4 to 72 months.

Figure 2. Illustrates the median loan duration, the interquartile range (IQR), and the overall spread of durations from the minimum to the maximum.

2. **Credit History:** This is a qualitative indicator that categorizes an individual's past dealings with credit, giving insight into their reliability as a borrower. There are 5 categories within this indicator:
   - No credits taken/ all credits paid back duly (293 applicants): applicants who have either never taken a loan or have successfully paid back all their credits in full and on time. Individuals in this category might be new to credit or highly responsible borrowers with a history of fulfilling their financial commitments. This can be seen as a positive indicator, although the lack of a credit history could affect the depth of data available for risk assessment.
   - All credits at this bank paid back duly (293 applicants): applicants in this category have taken loans only from the institution in question and have repaid these loans punctually and in full. This demonstrates a reliable track record with the specific bank, which might encourage the bank to view these individuals as lower-risk borrowers. This history suggests a well-established relationship with the bank.
   - Existing credits paid back duly till now (88 applicants): this signifies that the applicant currently has ongoing credits and has been making payments on time up to the present. This ongoing compliance indicates good financial management, suggesting that the borrower is less risky in terms of potential default.
   - Delay in paying off in the past (49 applicants): this category is for those who have had instances of delayed payments in their credit history. Delays can be indicative of financial distress, mismanagement, or changes in the borrower's financial situation.This could be a red flag for potential lenders, as past payment issues may predict future credit risks.
   - Critical account/ other credits existing (not at this bank) (40 applicants): **t**his includes individuals with accounts that are deemed critical, which typically means they have had serious credit issues such as defaults or accounts that have been handed over to collections. It also covers borrowers who have other outstanding credits not held at the bank in question. This is the most concerning category for lenders, indicating high risk and potential financial instability.

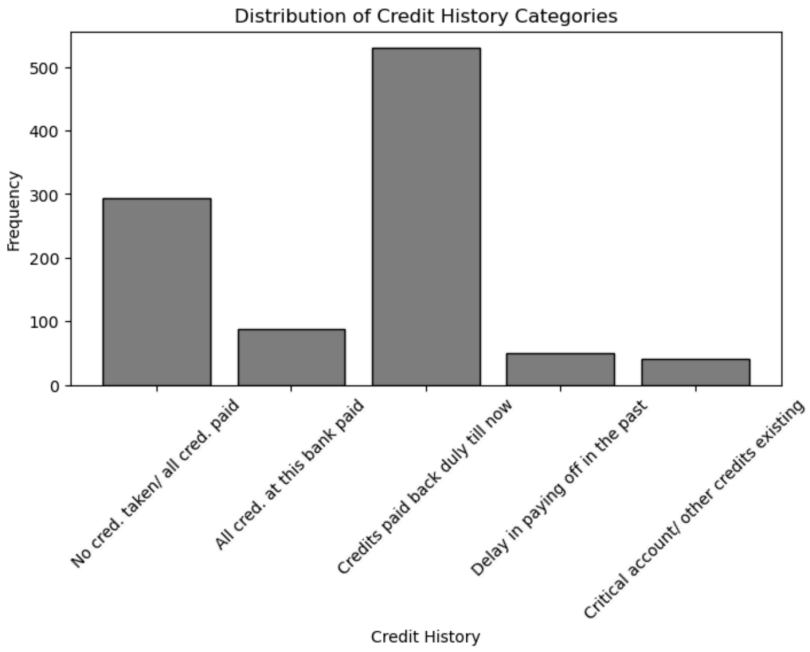


Figure 3. Illustrates the distribution of the applicants in the five Credit history categories

3. **Purpose of the Credit:** categorizes applicants according to their reasons applicants have for seeking credit. In the dataset, the distribution of loan purposes among applicants is categorized into ten distinct areas, reflecting a diverse range of financial needs. The categories include radio/TV, with 280 applications; new Car, with 234; furniture/equipment, with 181; used Car, with 103; business, with 97; education, with 50; repairs, with 22; domestic appliance, with 12; other, also with 12; and retraining, with 9 applications.

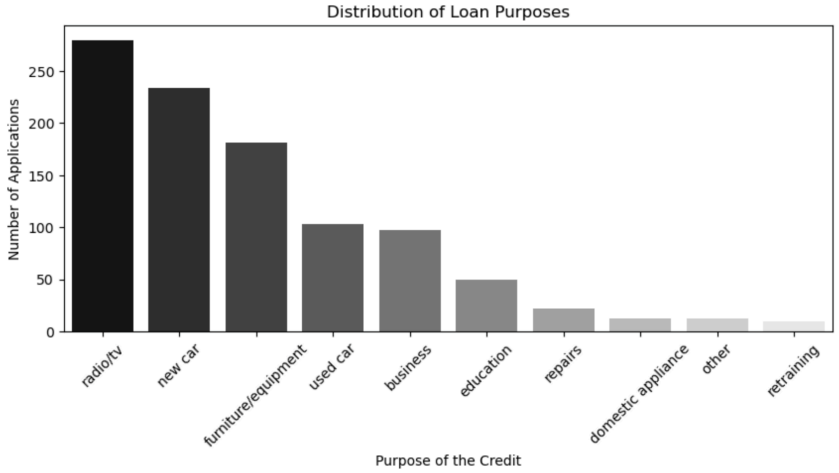


Figure 4. Illustrates the distribution of the applicants in the ten loan purpose categories

4. **Status of Saving Account/Bonds:** this variable categorizes applicants based on their financial reserves, providing insights into their savings and investment levels. There are five categories in the dataset:
   - <100: most common category with 603 applicants, indicating minimal savings.
   - No known savings: Includes 183 applicants without identifiable savings.
   - 100<=X<500: represents 103 applicants with moderate savings.
   - 500<=X<1000: comprises 63 applicants, indicating higher savings.
   - >=1000: smallest group with 48 applicants, showing substantial financial reserves.

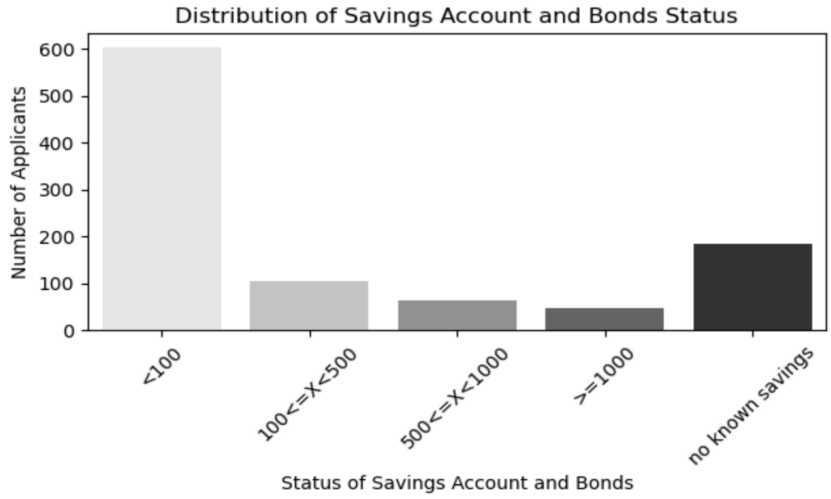


Figure 5. Illustrates the distribution of the savings amount of the applicants

5. **Credit Amount:** Numerical variable that represents the total monetary value of the loans sought by the borrowers. This variable can range from small sums, reflecting minor or short-term financial needs, to substantial amounts indicative of major purchases or investments, such as buying a car or starting a business. The loan amount spans from € 250.00 to € 18,424.00. The average (mean) loan amount is approximately € 3,271.26, with a standard deviation of about € 2,822.74. A detailed inspection reveals that 25% of the loans consist of amounts of € 1,365.50 or lower, 50% consist of amounts of € 2,319.5 or lower and 75% consist of amounts of € 3972.25 or lower.

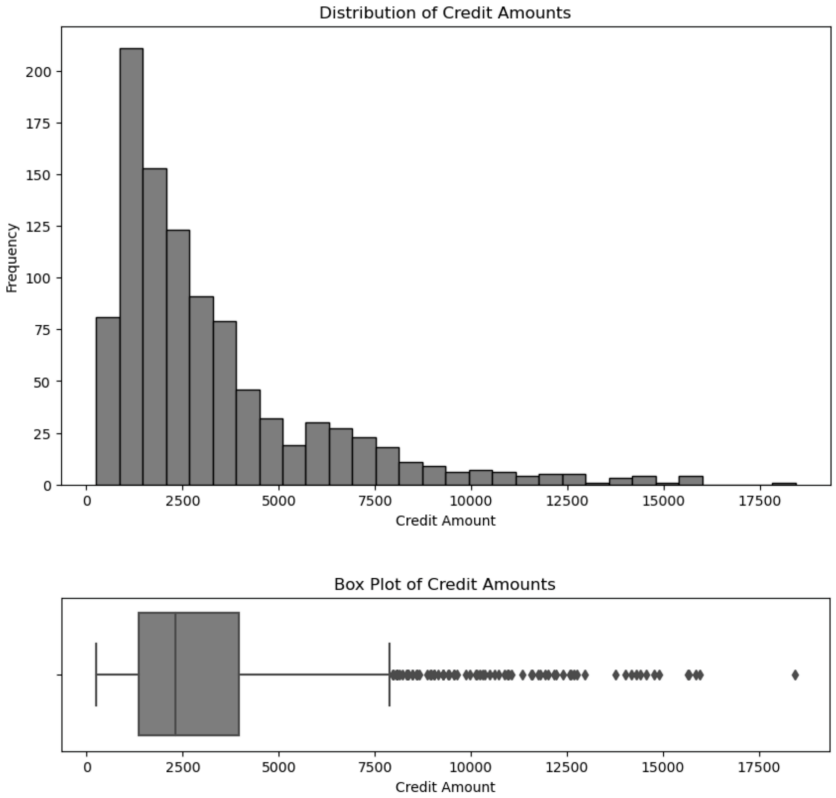


Figure 6. Illustrates the distribution of the loan amount in the dataset ranging from € 250.00 to € 18,424.00.

Figure 7. Illustrates the median loan amount, the interquartile range (IQR), and the overall spread in amounts from the minimum to the maximum.

6. **Present Employment (Years):** reflects the length of time applicants have been in their current job, offering insights into their employment stability and potential financial reliability. The employment durations in the dataset are defined as follows:
   - <1 year: includes 172 applicants, indicating relatively new employment.
   - 1<=X<4 years: the most common category with 339 applicants, suggesting a moderate level of job stability.
   - 4<=X<7 years: has 174 applicants, pointing to established employment relations.
   - >=7 years: comprises 253 applicants, highlighting long-term job stability.
   - Unemployed: accounts for 62 applicants, representing those currently without employment.

Figure 8. Illustrates the distribution of present employment duration across five categories

7. **Personal Status:** categorizes applicants based on their marital status and gender. Providing insight into the applicants' demographic backgrounds, which may influence their financial stability and loan repayment capabilities.
   There are four distinct categories: single male which is the largest group with 548 applicants; female div/dep/mar that includes 310 applicants who are either divorced, dependent, or married women; male mar/wid that comprises 92 applicants, denoting married or widowed men, typically associated with more stable family units and male div/sep which consists of 50 applicants who are divorced or separated men, potentially facing financial and personal challenges.

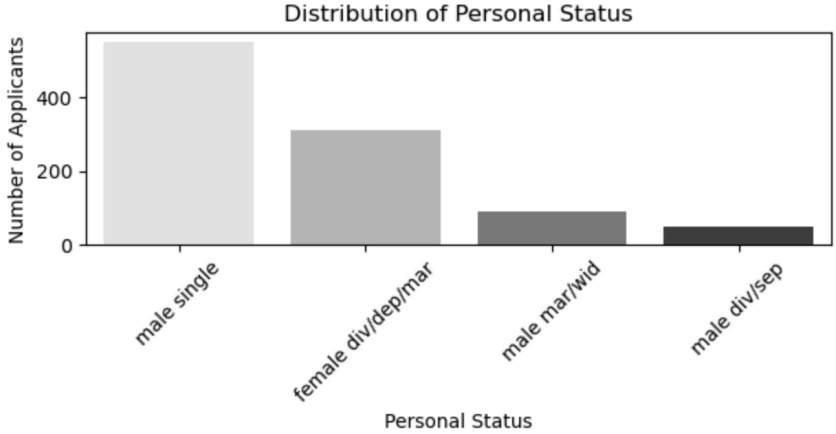


Figure 9. Illustrates the distribution of personal status of the applicants within four categories

8. **Installment Rate in Percentage of Disposable Income:** This variable is critical for understanding how much of a borrower's disposable income is allocated towards loan repayments. This rate provides insights into the financial burden that loan repayments impose on the borrowers, directly impacting their ability to sustain other financial obligations and lifestyle needs. The installment rate as a percentage of disposable income spans from 1% to 4%. The average (mean) loan amount is approximately 3%, with a standard deviation of about 1.12. Although this is a numerical variable, the rates are distributed only over four values: 1, 2, 3 and 4%.

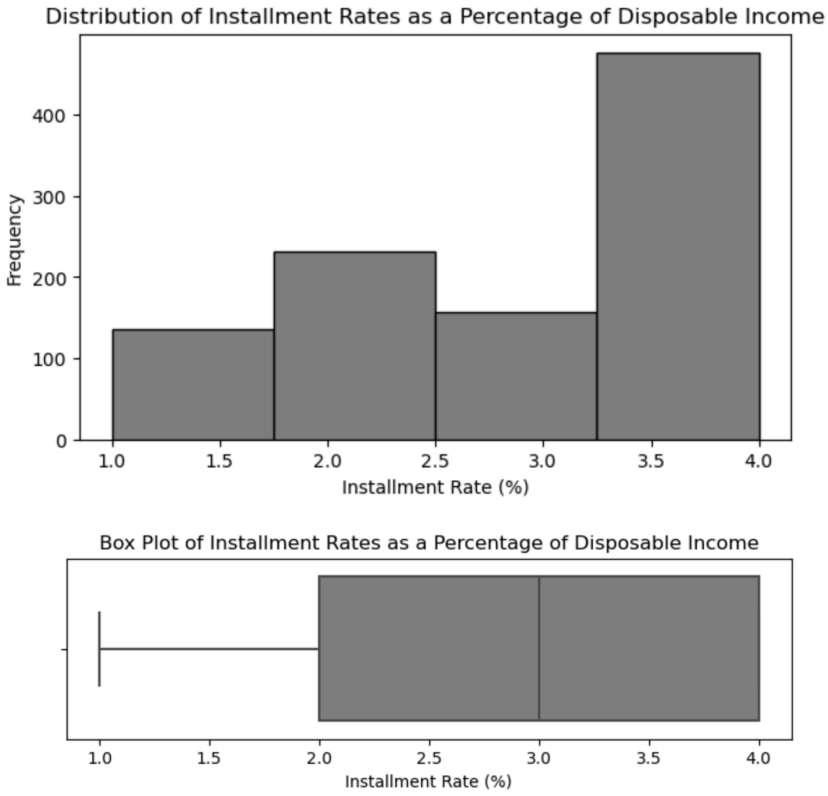


Figure 10. Illustrates the distribution of the percentage allocation of disposable income into the repayment of the loan.

Figure 11. Illustrates the median percentage of disposable income allocated to the repayment of the loan, the interquartile range (IQR), and the overall spread in amounts from the minimum to the maximum.

9. **Presence of Co-Applicants or Guarantors:** This variable evaluates the additional security that might be available for a loan, by categorizing the involvement of co-applicants or guarantors. The most predominant category is “none” with 907 applicants indicating no additional debtors or guarantors are involved in the loan. Then there is a smaller group of 52 applicants where a guarantor is present to back the loan. Finally the last group, formed by 41 applicants, includes the loans which are jointly applied for with another individual sharing the repayment responsibility.

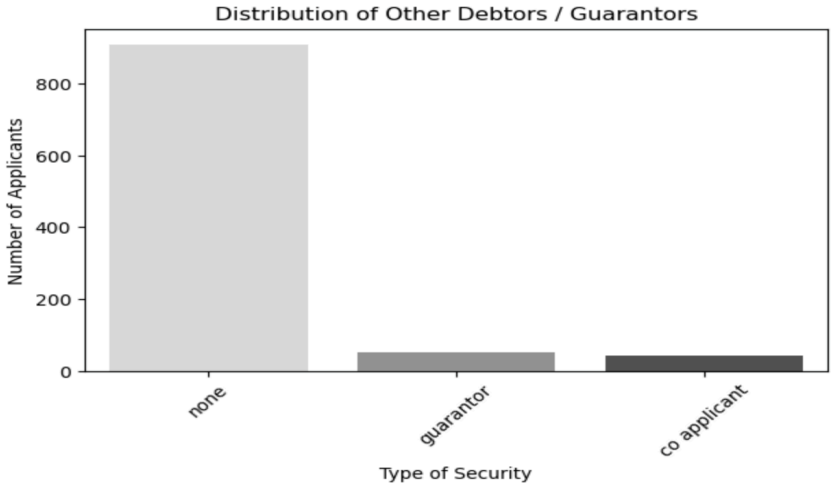


Figure 12. Illustrates the distribution across the three types of security of the applicants

10. **Number of Years in the Current Residence:** This variable shows for how long applicants have lived at their current addresses, providing a measure of residential stability. There are four categories for this variable: 4 years, which is the most common duration with 413 applicants, indicating a substantial group with relatively stable living conditions; 2 years, the next largest group, including 308 applicants, suggests a moderate level of stability; 3 years, which represents 149 applicants, which shows less frequency but still considerable residence time; and 1 year, the least common with 130 applicants, pointing to those who have recently moved or potentially have a more transient lifestyle

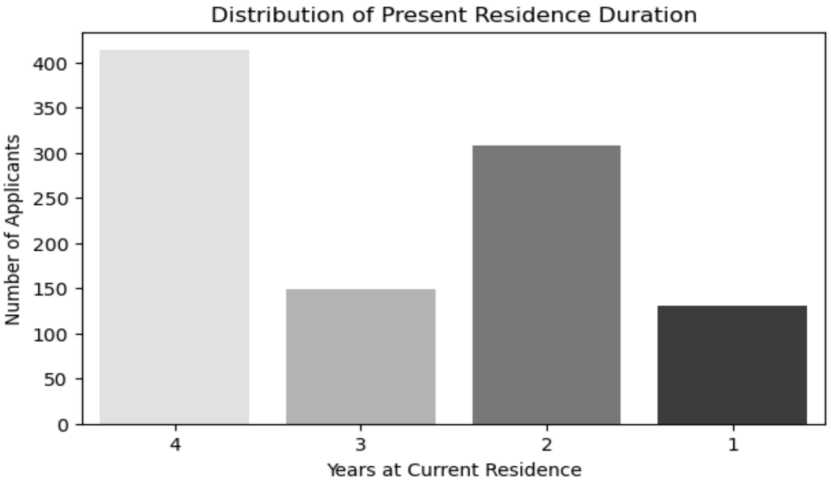


Figure 13. Illustrates the different number of years each applicant has been in their current residence

11. **Property:** this variable details the types of assets owned by applicants, serving as a significant factor in assessing their creditworthiness and loan security. There are four categories within this variable: car, the most common property type with 332 applicants, which can serve as collateral for personal loans or auto loans; real estate, owned by 282 applicants, represents a substantial financial asset, offering considerable security for lenders due to its typically high value and stability; life insurance, including 232 applicants, who have life insurance policies with a cash value that can be borrowed against, providing an additional form of financial security; and finally 154 applicants have no known property, potentially representing a higher risk group for lenders.

Figure 14. Illustrates the types of property owned by each applicant

12. **Other Installment Plans:** This variable shows any additional financial commitments of loan applicants, important for assessing their overall debt burden and repayment capacity. The majority of applicants, totaling 814, have no other installment plans, indicating a lower level of external financial commitments; 139 applicants have existing installment obligations with banks. This suggests a significant level of formal financial commitments that might affect their loan repayment capabilities. Finally 47 applicants have installment plans from retail stores, typically for consumer goods, which could reflect discretionary spending affecting their budget.

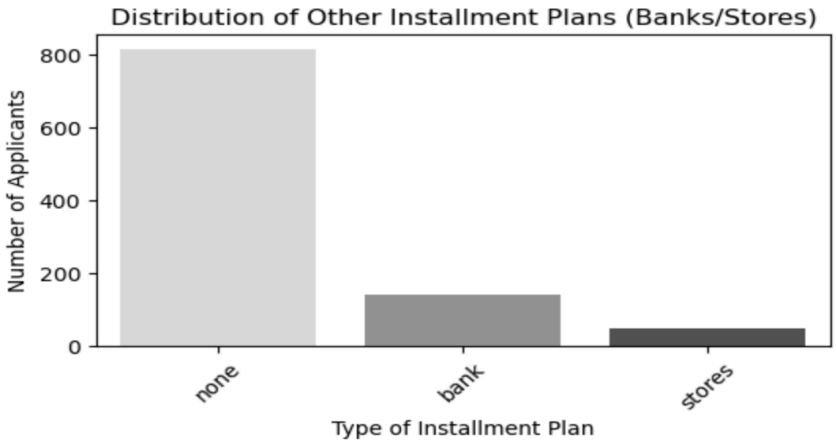


Figure 15. Illustrates the other types of debt each applicant has

13. **Age:** provides the distribution of the ages of all the applicants, ranging from from 19 to 75 years. The average age of applicants is approximately 35.54 years with a standard deviation of 11.35, suggesting a relatively young borrower pool but with a wide range of ages and possibly diverse financial needs and risks. 25% of applicants are 27 years old or younger, the median age is 33 years and 75% of applicants are 42 years or younger.

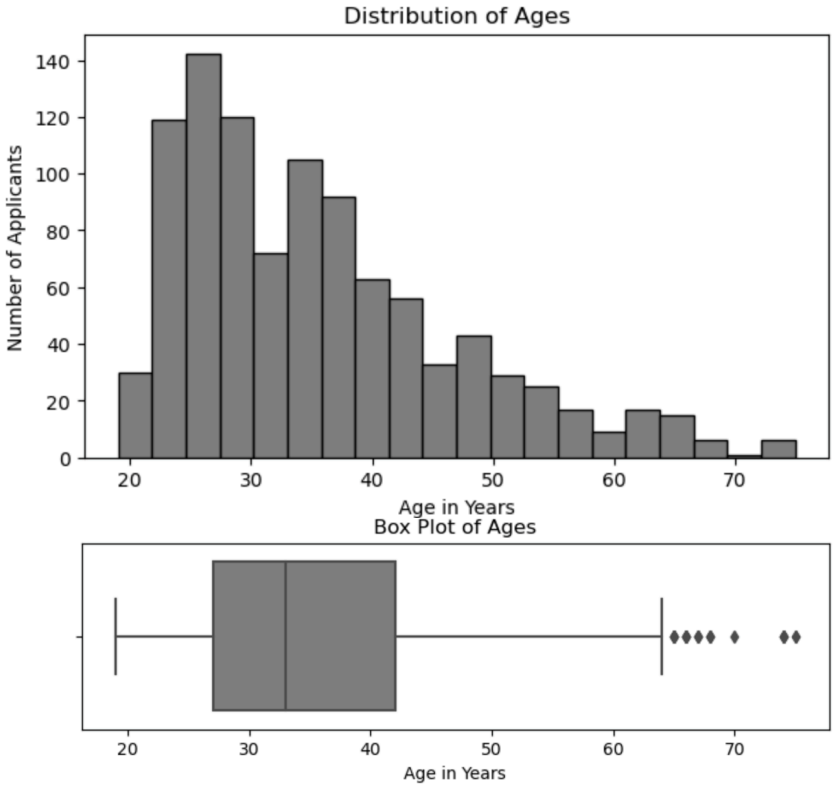


Figure 16. Illustrates the distribution of the age of the borrowers.

Figure 17. Illustrates the median age of the borrowers, the interquartile range (IQR), and the overall spread in ages from the minimum to the maximum.

14. **Housing:** This variable provides insights into the living arrangements of the loan applicants. There are 714 applicants who own the house they live in, which usually indicates financial stability and a greater ability to manage financial commitments effectively. Then there are 179 applicants who rent their residences. Renting can indicate more variable monthly expenses and potentially less financial stability than homeownership. Finally 107 applicants live in accommodations provided free of charge. This situation often reduces living expenses significantly, which could affect disposable income and financial decision-making.

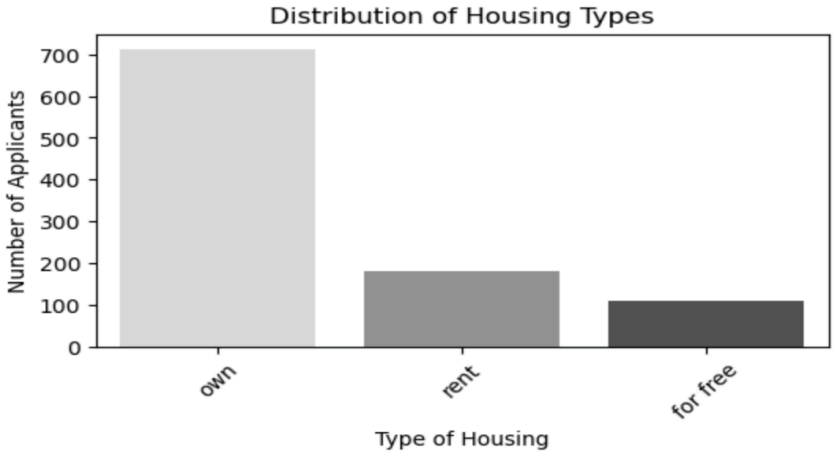


Figure 18. Illustrates the distribution of the applicants across three different types of housing

15. **Number of Existing Credits at the Bank:** quantifies how many loans or credit lines an applicant currently has with, or is applying to the bank. In the dataset, 633 applicants have 1 credit with the bank. 333, have 2 credits; 28 applicants have 3 credits, and only 6 applicants have 4 credits with the bank. The more credits the applicants have with the bank, the deeper financial relationship or higher trust they have with the bank. However an higher number of existing credit lines or loans might also be an indicator of possible financial problems.

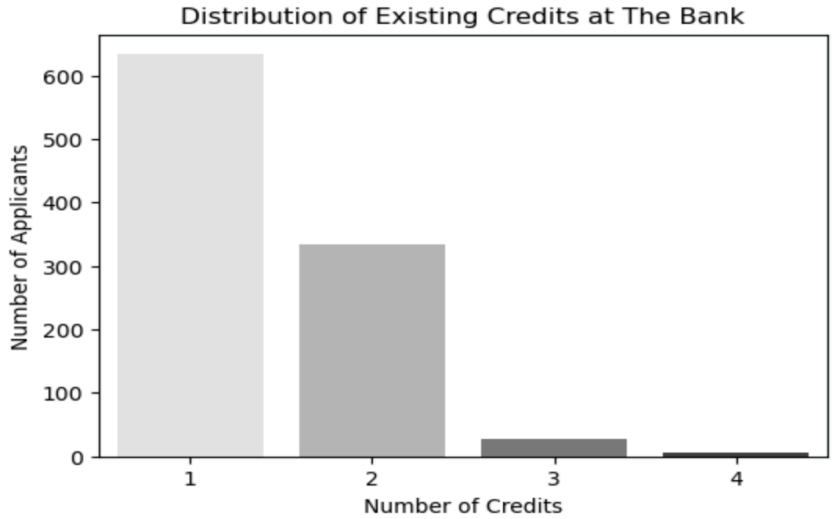


Figure 19. Illustrates the distribution of existing credits with the bank issuing the loan

16. **Job:** this variable categorizes the employment status and type of job held by loan applicants. “Skilled is the most prevalent category with 630 applicants, indicating those who possess specific skills likely associated with stable and possibly higher income levels. Then, there are 200 applicants in the “Unskilled” category, referring to those in unskilled jobs who reside locally. This group might face lower income levels and potentially less financial stability. Highly qualified/self-employed/ management includes 148 applicants in high-level professional or managerial positions, or who are self-employed and benefit from higher salaries and greater financial stability. Finally there are 22 applicants who are Unemployed or are unskilled workers working abroad. This category has the highest potential risk due to unemployment or non-resident status combined with unskilled job qualifications.

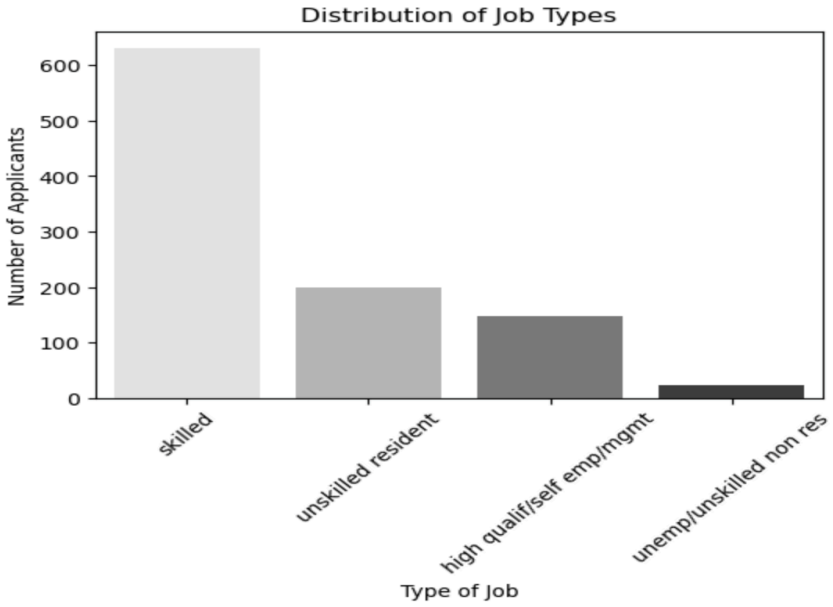


Figure 20. Illustrates the distribution of Job type of the applicants

17. **Number of People Being Liable to Provide Maintenance For:** quantifies the number of dependents an applicant is financially responsible for. In the dataset there are only two options provided. The first and most common, with 845 applicants, is having 1 dependent. The second one, with 155 applicants, is having 2 dependents.

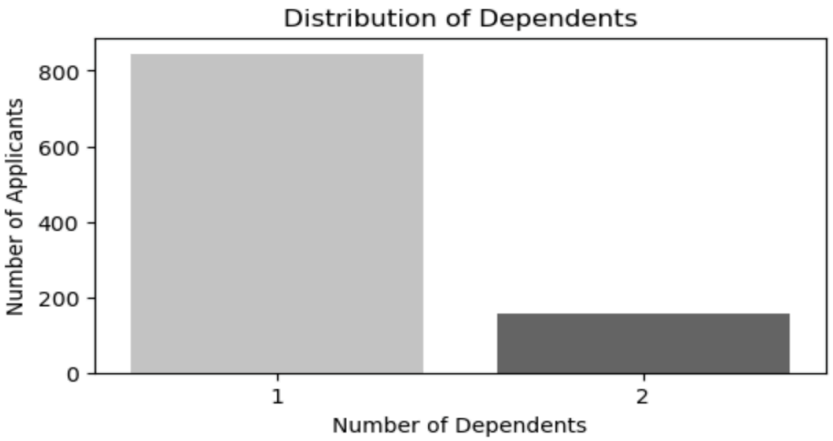


Figure 21. Illustrates the number of dependents each applicant has to provide for

18. **Possessing a Telephone: c**ategorizes applicants based on whether they have a telephone registered in their name. In the dataset the majority of applicants (596) do not possess a phone, while the remaining 404 have a phone.

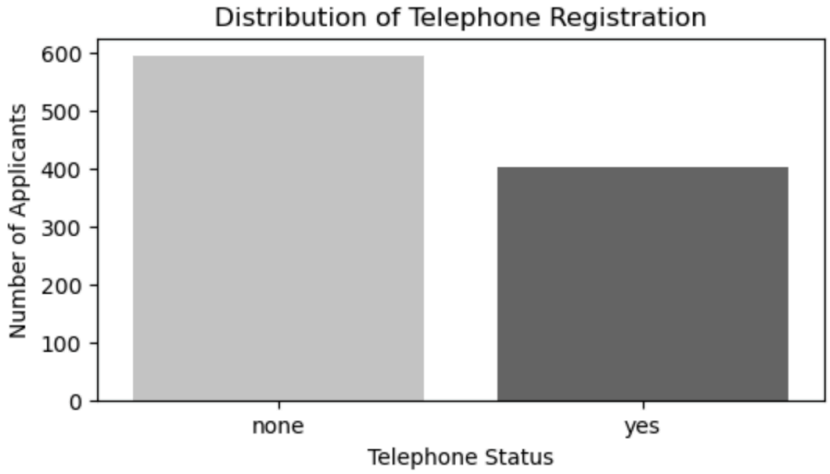


Figure 22. Illustrates the number of applicant possessing a telephone

19. **Home or Foreign Worker:** this variable categorizes the amount of applicants who work in the home country or in a foreign country. In the dataset there are 963 people working in the home state and 37 working abroad.

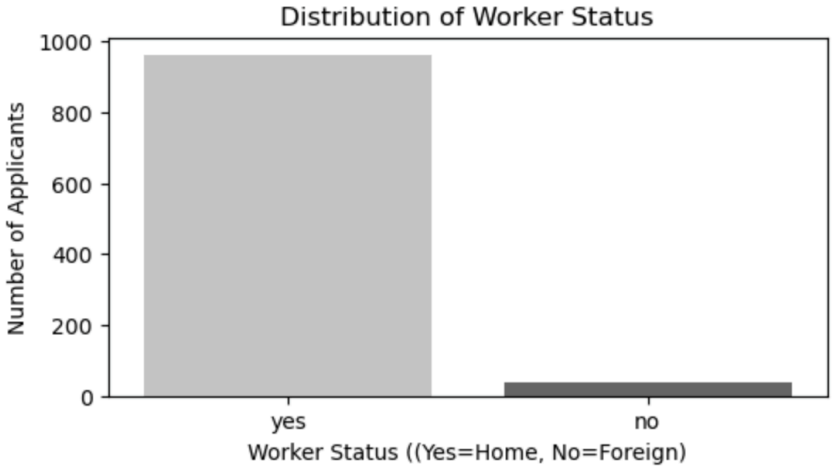


Figure 23. Illustrates the number of applicants working in the home state or abroad

20. **Account Balance: t**his variable categorizes an individual based on his available funds, into four levels. A low balance or empty account has a total count of 310, which means the individual has little to no funds available, and thus, there is a higher financial risk and potential inability to repay loans. A positive balance but not high was noted in 269 cases, reflecting a more stable financial situation that does not yet provide complete financial security. A good or stable balance was recorded in 63 cases, indicating that the person maintains a healthy financial status, thus being more likely to be creditworthy. A high or very positive balance is the most frequent category, with 358 occurrences, which means a good financial status with large available funds and is usually linked to low credit risk.

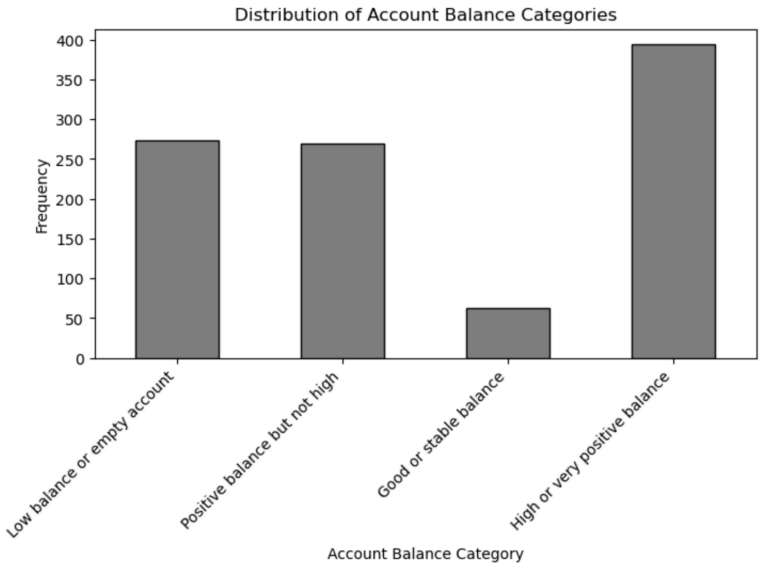


Figure 24. Illustrates the distribution of account balance categories of the applicants

21. **Credit Score**: The last variable is the credit score given by the bank to each applicant. In this dataset the majority of applicants (700) obtained a good credit score while the remaining 300 obtained a bad credit score. This means that 70% of the applicants are deemed creditworthy, indicating a high probability of loan repayment without issues. On the other hand, the remaining 30% of applicants are considered at a higher risk for defaulting on a loan.

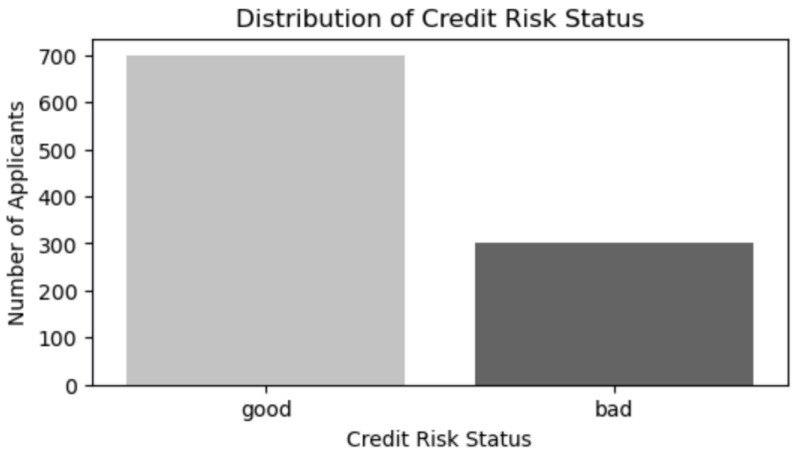


Figure 25. Illustrates the distribution of the credit scores of the applicants

3.1.2 Comprehensive Dataset Analysis

Looking at the dataset as a whole, it's important to observe how the variables are correlated. This can be done with the aid of the correlation matrix. This matrix enables the user to observe the linear relation shared by pairs of variables in a given dataset. Traditionally, such a matrix comes out as a tabular format where the rows and columns stand for various variables under study, while each cell says something about the correlation coefficient for the pair of variables in question. These range from -1 to +1, where +1 signifies a perfect positive linear relationship, -1 indicates a perfect negative linear relationship, and 0 suggests no linear relationship.

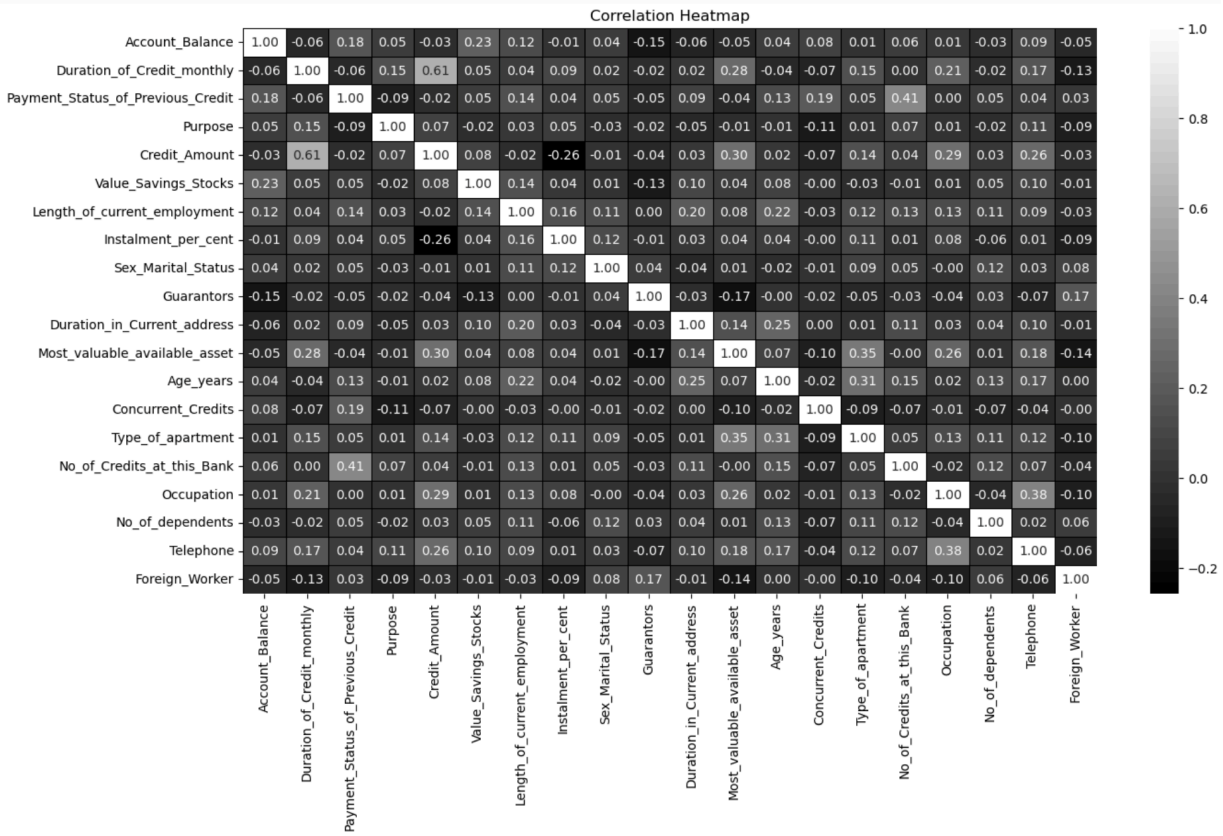


Figure 26. Illustrates the correlation matrix showing the multicollinearity between variables

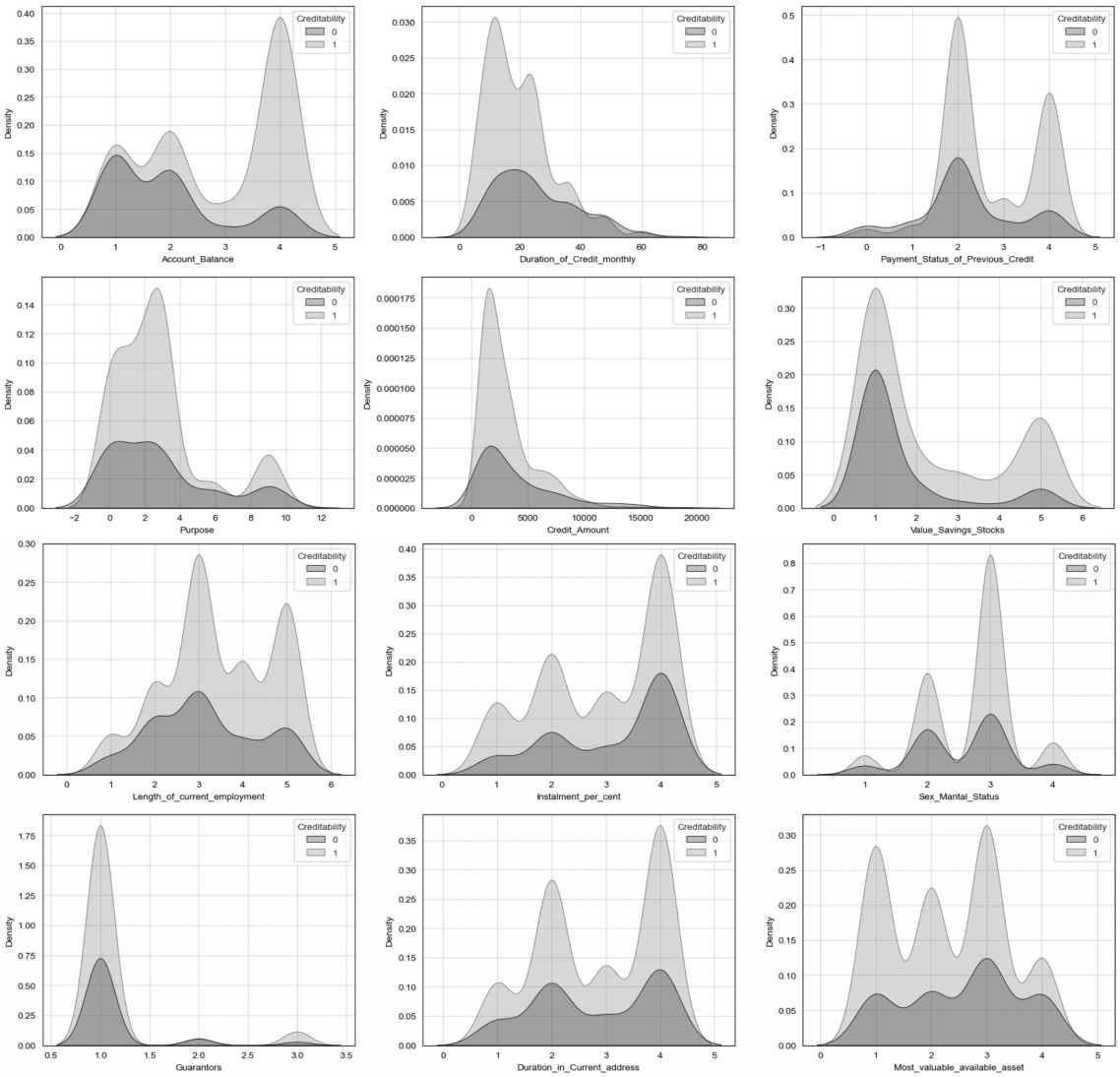
The variables in the matrix are all the numerical independent variable without the target variable. The matrix enables multicollinearity detection, that is, when two or more predictors are highly interrelated. This may pose problems in interpreting results from the predictive model since it will be hard to tell or measure the contribution of each independent variable on a dependent variable. For instance, the highest correlation of 0.61 between 'Duration in months' and 'Credit amount' does indicate that these two variables share a good amount of variance and could potentially cause issues with multicollinearity for some types of models.

On the other hand, the general level of correlation among the variables is low. This suggests a complex landscape of relationships that are not readily apparent through linear measures. This scenario implies that the influences on credit scoring are subtle and possibly non-linear, challenging the efficacy of a traditional linear model like linear regression.

Another useful way to visualize the data in showing the correlation between the numerical variables and the target variable is using the Kernel Density Estimation (KDE) plots, also known as density plots. A kernel density plot is a non-parametric technique to estimate the probability density function (PDF) of a continuous variable. Unlike histograms, which rely on an a priori selection of bin size, KDE provides a smoother description of the data distribution. To build the KDE curve, one places a kernel—a typically Gaussian function—at each data point and sums these local contributions to construct a continuous probability density function.

The series of kernel density plots displayed below explore the variables' distributions across the two categories of credit scores, or creditability, that are 1 and 0. In this case '1' stands for "Good credit score" and '2' stands for "Bad credit score".

The curve height at any value is the relative probability density of the value occurring in the data set. Compared to a histogram that plots frequency counts for discrete ranges, KDE allows for the plotting of a continuous curve, thus making it easier to interpret underlying trends and variability.



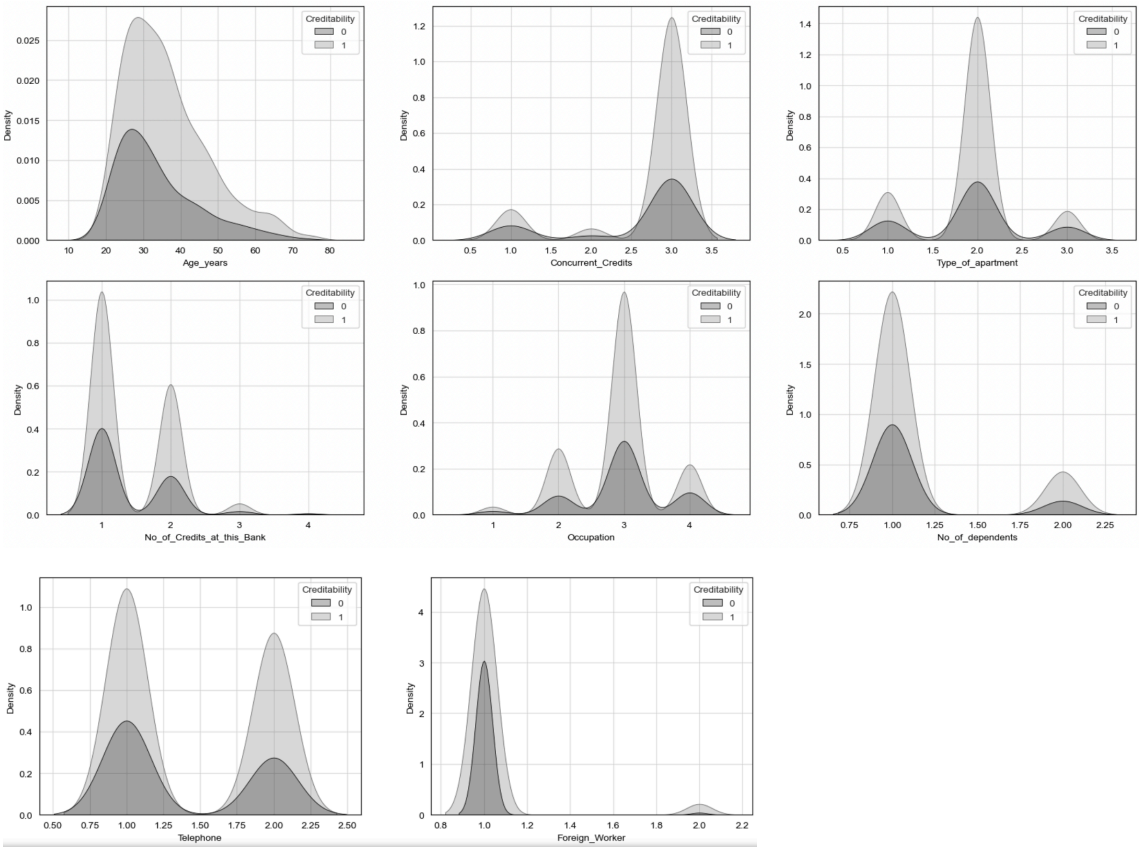


Figure 27. Illustrates trough KDE plots how the variables are distributed in relation to the target variable.

Account Balance: Individuals with a higher account balance (4) are more likely to be creditworthy, while those with a low balance (1-2) tend to have a lower probability of good credit standing.

Duration in Months: Shorter credit durations are more associated with creditworthy individuals, while longer durations show a higher density for non-creditworthy individuals.

Credit History: Here, there is a significant peak for creditworthy individuals around category 4 (no credit taken or all credits paid). Conversely, non-creditworthy individuals exhibit a wider spread across lower values, reflecting a more inconsistent repayment history.

Purpose: This shows that creditworthy individuals tend to cluster around specific credit purposes like category 4 (used cars), whereas non-creditworthy individuals have a more evenly distributed presence across different categories. This suggests that certain loan purposes may be associated with a higher risk of default.

Credit Amount: Highlights that non-creditworthy individuals are more likely to have higher loan amounts, as their distribution is more spread out towards larger values. In contrast, creditworthy individuals show a denser peak at lower credit amounts, indicating that smaller loan requests tend to be less risky.

Status of Saving Account/Bonds: Creditworthy individuals exhibit strong peaks, particularly in category 5, which includes people with no known savings. This might seem like a contradiction; however, this category also includes individuals who may possess other types of non-liquid assets, which could make them creditworthy in other

ways. On the other hand, non-creditworthy individuals are concentrated in lower asset value categories, reinforcing the idea that having fewer savings is a significant risk factor.

Present Employment: Creditworthy individuals show strong peaks around categories 3, 4, and 5, indicating that longer employment durations are associated with higher creditworthiness. Non-creditworthy individuals have a more evenly spread distribution, with a notable presence in lower employment duration categories. It is important to specify that the category "unemployed" corresponds to value "1" in the graph above.

Installment Rate in Percentage of Disposable Income: This graph shows that both non-creditworthy and creditworthy individuals are distributed in a similar manner, with peaks occurring at the same categories. This suggests that the proportion of income allocated to installment payments does not significantly differentiate between the two groups.

Personal Status: Here, the peak at category 3 (married male) is significantly higher for creditworthy individuals compared to non-creditworthy ones. Non-creditworthy individuals are more evenly distributed across the other categories.

Presence of co-Applicants or Guarantors: The majority of individuals fall into category 1, which corresponds to those without a guarantor. This suggests that having no guarantor is the most common scenario, regardless of creditworthiness. However, in category 3, which represents individuals with a co-applicant, there is a noticeable peak for creditworthy individuals, despite the lower overall frequency. This suggests that while having a co-applicant is relatively uncommon, it is associated with a higher likelihood of being creditworthy.

Number of Years in Current Residence: This graph indicates that creditworthy individuals have strong peaks at categories 2, 3, and 4, implying that longer residence stability is linked to better creditworthiness. Non-creditworthy individuals are more dispersed, with a higher density at lower categories, suggesting that frequent changes in residence may be a risk factor.

Property: Creditworthy individuals show higher peaks in category 1 (car ownership) and category 2 (real estate ownership), indicating that owning these assets correlates with an high financial stability. In contrast, category 4 (no assets) has the highest density of non-creditworthy individuals, meaning that that the lack of properties is a major risk factor for credit default.

Age: This graph shows that most of the creditworthy individuals are concentrated between 30 and 40 years old, while non-creditworthy individuals have a broader distribution, including older ages, although many are concentrated between the ages of 20 and 30 years old.

Other Installment Plans: Both groups have a peak at category 3 (that includes people with no other installment plans), indicating that having no concurrent credits is common among both creditworthy and non-creditworthy individuals.

Housing: This graph highlights that most creditworthy individuals fall into category 2 (which in this case corresponds to owning a house). In contrast, non-creditworthy individuals are more prevalent in the other two categories indicating that those who live in a rented house or in a house they do not own but without paying rent are more likely to be at risk of default.

Number of Credits at the Bank: Creditworthy individuals are most concentrated at 1 and 2, indicating a preference for having fewer loans. Also non-creditworthy individuals are present in these categories but in lower densities.
Job: This graph shows that creditworthy individuals peak in category 3 and 4 (skilled workers and highly qualified jobs respectively), indicating that stable jobs correlate with better creditworthiness. Non-creditworthy individuals, on the other hand, are more concentrated in category 4 (unskilled workers) and category 1 (unemployed), reinforcing their higher risk of default.
Number of Dependents: Both groups peak at 1, but non-creditworthy individuals have a slightly higher density at 2. This suggests that having more dependents may slightly increase financial strain, potentially influencing credit risk.
Telephone: Here, both creditworthy and non-creditworthy individuals are fairly evenly distributed across categories 1 and 2, suggesting that having a registered telephone may not be a strong differentiator for creditworthiness.
Home or Foreign Worker: As for the previous variable, the regular distribution of values suggests that being a home or foreign worker is not a major factor in determining creditworthiness since both creditworthy and non-creditworthy individuals follow a similar pattern**.**

3.2 Data Processing

In this phase of the study, it is necessary to process the data for adequately prepare the dataset for further modeling. Python was chosen for this purpose because of its effectiveness in the manipulation of large amounts of data.
In order to use Python it is necessary to import the following libraries into the program:

```
import numpy as np
import pandas as pd
pd.set_option("display.max_columns", None)

import seaborn as sns
import matplotlib.pyplot as plt
import matplotlib.ticker as ticker

from statsmodels.graphics.gofplots import qqplot
from scipy.stats import shapiro, norm
from scipy.stats import boxcox

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler, MinMaxScaler, LabelEncoder
from sklearn.model_selection import cross_val_score
from sklearn.compose import ColumnTransformer
from imblearn.over_sampling import RandomOverSampler, SMOTE
from imblearn.under_sampling import RandomUnderSampler
from sklearn.impute import KNNImputer
from sklearn.feature_selection import RFE
from sklearn.preprocessing import PowerTransformer

from imblearn.pipeline import Pipeline, make_pipeline
from xgboost import XGBClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import StratifiedKFold
from catboost import CatBoostClassifier, Pool, cv
from sklearn.ensemble import
RandomForestClassifier,
GradientBoostingClassifier,
ExtraTreesClassifier,
VotingClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.svm import SVC
from sklearn import svm
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import AdaBoostClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn.neural_network import MLPClassifier
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis
from sklearn.model_selection import GridSearchCV

from sklearn.metrics import
accuracy_score,
classification_report,
ConfusionMatrixDisplay,
balanced_accuracy_score,
recall_score,
fbeta_score,
f1_score

from typing import Dict, List

import time

import shap

import warnings
warnings.filterwarnings("ignore")
```

Among the most notable libraries there are:

- For basic data handling:
  Numerical Python (Numpy): A common library used to handle big matrixes and multi-dimensional arrays, providing an extensive collection of mathematical functions to facilitate these operations. NumPy serves as one of the fundamental libraries on which other scientific libraries exist.
  Pandas: This is designed for data manipulation and analysis, offering high-level data structures along with a comprehensive array of analysis tools. It simplifies tasks such as data sorting, re-indexing, concatenation, and visualization, along with advanced data manipulation and cleaning techniques.

- For data visualization, including the creation of all the graphs in sections 3.1.1. and 3.1.2., the following libraries are used:
  Matplotlib, which is a foundation library for plotting in Python and forms the basis for most other visualization libraries. This library provides very detailed control over plots, including but not limited to labeling axes and setting figure sizes.
  This is further extended by the ticker module in Matplotlib, which provides even finer control over axis formatting, enabling precise adjustment of tick locations, label formatting, and spacing.
  Seaborn, that is a high-level visualization library based on Matplotlib. It is particularly useful in statistical graphics and makes creating informative and attractive plots much easier. It interfaces nicely with Pandas and is very suitable for exploratory data analysis. Seaborn provides some specialized visualization tools: violin plots, box plots, and heatmaps that are very useful in visualizing distributions, trends, and correlations within structured data. One of the key strengths is the automatic inclusion of statistical elements, such as confidence intervals in regression plots, thereby making the data more interpretable.

- For data processing:
  Scikit-Learn: This is probably the most popular library in machine learning, which possesses many tools for both supervised and unsupervised learning. It contains the implementations of most important algorithms of classification, regression, clustering, and dimensionality reduction. Scikit-learn is built upon NumPy and SciPy, making structured data processing very efficient. In the project, several essential components have been used from Scikit-learn:
  The train_test_split essentially helps divide the dataset into training and test sets to ensure models are tested on unseen data to avoid overfitting.
  StandardScaler and MinMaxScaler are used in data preprocessing to normalize features of numeric values either by standardizing the values at mean 0 and variance of 1, or scaling features within a given range from 0 to 1. This helps in modeling the algorithms of support vector machines, K-nearest neighbors and neural networks.
  OneHotEncoder transforms categorical variables into binary vectors, therefore making them machine learning model-compatible, which require numerical inputs. ColumnTransformer enables the preprocessing of numerical and categorical data in one pipeline and takes care that the different transformations are applied correctly. Furthermore, LabelEncoder is used to encode target labels into numbers, especially

for classification problems.
Recursive Feature Elimination (RFE) selects features by iteratively eliminating the less important ones to enhance model efficiency. For handling missing values in the dataset, KNNImputer will impute missing entries in a dataset based on the values of nearest neighbors. Finally, PowerTransformer performs transformations such as Box-Cox and Yeo-Johnson for making data more normally distributed, which would help in enhancing the effectiveness of machine learning models.
The training will be done using several classifiers from Scikit-learn's ensemble, tree, svm, and linear_model modules. These include Random Forest, Gradient Boosting, Extra Trees, Voting Classifier, Decision Trees, Support Vector Classifier (SVC), Logistic Regression, AdaBoost, Naïve Bayes, Multi-layer Perceptron (MLP), and Linear Discriminant Analysis (LDA).
Hyperparameter tuning will be done with GridSearchCV and StratifiedKFold to allow systematic optimization and robust cross-validation of the models.
Imb-learn: The problem of imbalanced datasets is very common in machine learning, where it deals with a classification problem in which one class significantly outweighs another. The imbalanced-learn library is built on top of Scikit-learn and provides specialized techniques to deal with such datasets to improve model performance without bias toward one majority class. To handle class imbalance, oversampling and under-sampling are performed. RandomOverSampler performs the balancing by duplicating samples of the minority class. SMOTE makes new data points to balance a dataset, using Synthetic Minority Over-sampling Technique. Meanwhile, RandomUnderSampler does the opposite by removing random samples from the majority class to reduce class disparity and improve the model's generalization across classes. Those resampling methods are highly relevant in the training of a classifier, enabling generalization from both the majority and the minority class distributions so that a bias for one dominating class does not occur in the model. This is particularly useful when combined with sensitive ensemble models such as Random Forest, XGBoost, and Gradient Boosting, as they tend to perform poorly against imbalanced distributions.
XGBoost: This is one of the most powerful machine learning algorithms for structured data. It is the optimized implementation of gradient boosting that is much faster and scalable than traditional boosting methods, with better performance in a variety of tasks. Because of its efficiency on large datasets, it finds a wide range of applications in predictive modeling, financial risk analysis, and competitions in machine learning. XGBoost regularization methods include both L1 and L2 for preventing overfitting of regular decision trees, and intrinsically, they support parallel and GPU processing of the code; therefore, XGBoost results in fast processing. This puts it in front of many existing models. Built-in support has been made with a range of advanced techniques available, from feature importance to automatically handling missing values, and in this way can early stop optimized model performance. In this project, XGBClassifier is used, which is a classification model based on extreme gradient boosting that improves the accuracy of prediction. For tuning the best hyperparameters that yield the best generalization performance, GridSearchCV and StratifiedKFold are used. **(25)**

After importing all the libraries, the next step is to ensure that the dataset is structured correctly. This involves organizing the features and the target variable in a consistent format, positioning the target (Creditability) as the last column of the dataset. This simplifies data manipulation and ensures compatibility with the various machine learning models tested below.
To achieve this, the code is:

```
a = df.pop('Creditability')
df['Creditability'] = a
```

Once the dataset is properly structured, the next step is to split the dataset into training and testing subsets, allowing the models to be trained on one portion of the data while being tested on unseen data to assess its generalization performance.
Here, the code used is:

```
def split_data(df, test_size=0.25, random_state=42):
        X = df.iloc[:, :-1]
        y = df.iloc[:, -1]
        X_train, X_test, y_train, y_test =
        train_test_split(X, y, test_size=test_size, random_state=random_state)
        df_train = pd.concat([X_train, y_train], axis=1)
        df_test = pd.concat([X_test, y_test], axis=1)
        print('split_data is ready')
        return df_train, X_test, y_test
```

Where X represents the independent variables (features), while y is the dependent variable (target). The dataset is then split into training (75%) and testing (25%) subsets, ensuring that the model is trained on a majority portion of the data while being evaluated on the remaining portion. The random_state parameter ensures that the split remains consistent and reproducible.

At this point, the process continues with the features processing. This is important for achieving maximum model performance by making the features optimally scaled, transformed, and balanced prior to their utilization in classification models.
One of the most important preprocessing techniques employed is feature scaling, which makes all numeric features lie in the same range. For this purpose, Min-Max Scaling is performed using the following function:

```
X_train = helper.min_max_scale_columns(X_train)
X_test = helper.min_max_scale_columns(X_test)
```

This scaling normalizes all feature values to be between 0 and 1, which prevents variables with large magnitudes from dominating the model.
Besides scaling, Power Transformation also works to normalize distributions of data that are skewed. Most machine learning algorithms, particularly linear models and neural networks, presume that the input features are normally distributed. In real-world applications, data tends to exhibit skewness that can adversely affect model performance. The Yeo-Johnson Power Transformation (26) is applied to make feature distributions more Gaussian-like by reducing skewness and variance differences among features **(26)**:

```
power = PowerTransformer(method='yeo-johnson')
X_train_p = power.fit_transform(X_train)
power = PowerTransformer(method='yeo-johnson')
X_test_p = power.fit_transform(X_test)
```

One of the essential aspects of data preprocessing includes the handling of class imbalance. In classification problems, particularly those related to fraud detection, medical diagnosis, or credit risk assessment, datasets often witness an imbalanced distribution of target classes. This imbalance can lead to biased models that overwhelmingly prefer the majority class and have compromised performance on the minority class. To counter this issue, the Synthetic Minority Over-sampling Technique (SMOTE) is employed:

```
X_train_s, y_train_s = helper.balance_data_smote(X_train, y_train)
X_train_p_s, y_train_p_s = helper.balance_data_smote(X_train_p, y_train)
```

SMOTE creates synthetic instances of the minority class, thereby boosting its representation in the training set. By this method, the model can learn patterns from both classes more efficiently, thereby enhancing its generalization capability to unseen data.

3.3 Models training and Evaluation

With the data preprocessing phase completed, the focus shifts to testing various machine learning algorithms to determine which model achieves the highest predictive accuracy. The evaluation begins with Linear Regression-based models, such as Logistic Regression and other variants. It then moves to Decision Tree-based models, including Decision Trees, Random Forest and other variants. Finally, a selection of non-tree-based models, such as K-Nearest Neighbors (KNN), Support Vector Classifier (SVC), Gaussian Naïve Bayes, and Multi-Layer Perceptron (MLP).

Each model will be evaluated on different classification measures to provide an examination of its predictive ability. The accuracy score, representing the ratio of correctly predicted instances to the total number of predictions, is a fundamental measure of general correctness. For unbalanced datasets, when one class is much more common than others, accuracy can be deceptive; a model can have a high accuracy rate simply by preferring the majority class. To prevent this problem, balanced accuracy is employed, which computes the arithmetic mean of recall for all classes so that each class has a proportionate impact on the ultimate measure independent of its frequency. This renders it highly suitable for identifying underrepresented classes and reducing class imbalance bias. A second important measure is the F1-score, which is a harmonic mean of precision and recall.

Precision is a measure of the ratio of correctly identified positive cases to all positively predicted cases, and hence a measure of the success of the model in avoiding false positives. Recall, on the contrary, calculates the ratio of correctly predicted positive cases to the total number of true positive cases, thereby representing the model's capacity to predict true positives. The F1-score compromises between the two, and hence it is particularly important in scenarios where false positives and false negatives have unequal costs, as is commonly encountered in applications such as medical diagnosis, fraud detection, or credit risk analysis. In a bid to investigate further the performance of the model, a confusion matrix will be established. The matrix outlines the components of true positives, true negatives, false positives, and false negatives, hence providing a graphical illustration of classification errors.

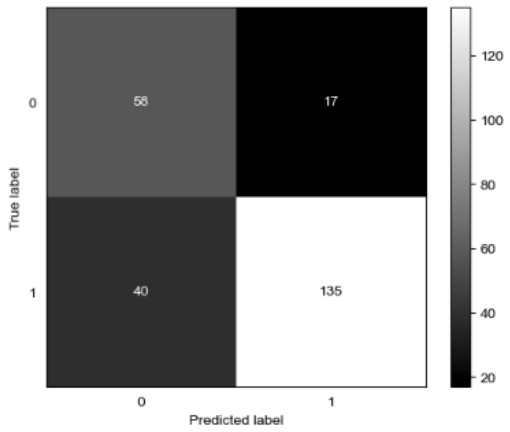
From the confusion matrix, one can determine particular weaknesses in the model, for example, its propensity to confuse certain classes with greater regularity or its inability to handle borderline instances.

3.3.1 Linear Regression-Based Models

The first model to be tested is the most popular and widely used ML algorithm, the **Logistic Regression** model, where the relationship between input variables and the target is defined through a linear equation, with coefficients estimated using maximum likelihood estimation.
The code to train and evaluate the model, also showing the results, is the following:

```
etc = LogisticRegression(random_state=42).fit(X_train_s, y_train_s)
y_pred = etc.predict(X_test)

ac = accuracy_score(y_test, y_pred)
g = balanced_accuracy_score(y_test, y_pred)
f1 = f1_score(y_test, y_pred)

print('Accuracy Score: %.2f' % ac)
print('F1-score: %.2f' % f1)
print('G-Mean: %.2f' % g)
print(classification_report(y_pred, y_test))
cm = confusion_matrix(y_test, y_pred)
disp = ConfusionMatrixDisplay(confusion_matrix=cm)
disp.plot(cmap='gray')
plt.grid(False)
plt.show()
```

```
Accuracy Score: 0.77
F1-score: 0.83
G-Mean: 0.77
              precision    recall  f1-score   support

           0       0.77      0.59      0.67        98
           1       0.77      0.89      0.83       152

    accuracy                           0.77       250
   macro avg       0.77      0.74      0.75       250
weighted avg       0.77      0.77      0.76       250
```



The results show an overall accuracy of 0.77, indicating that the model correctly classifies approximately 77% of the instances in the test set.
The precision for class 0 is 0.77, meaning that when the model predicts a negative class, it is correct 77% of the time. However, the recall for class 0 is 0.59, indicating that only 59% of actual negative cases are correctly identified. This suggests that the model struggles to detect a significant portion of negative instances, potentially leading to a higher false negative rate. On the other hand, class 1 exhibits a higher recall (0.89), implying that the model is highly effective at identifying positive cases but at the cost of some misclassification errors.

The F1-score, which balances precision and recall, is 0.67 for class 0 and 0.83 for class 1, confirming that the model performs substantially better in predicting the positive class. This disparity indicates that the model may be biased towards predicting positive cases more confidently while being more uncertain about negative cases.

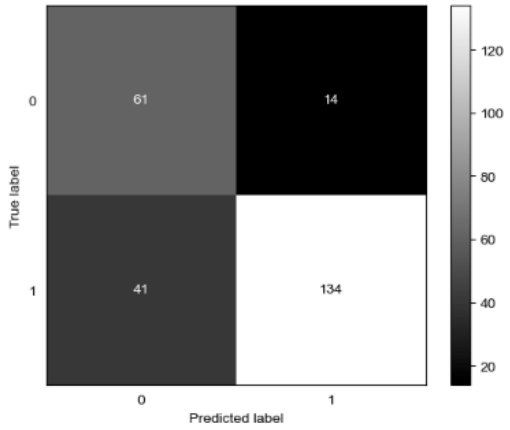
The confusion matrix further highlights this imbalance. Among the 98 actual instances of class 0, the model correctly classifies 58, while misclassifying 40 as class 1. Conversely, out of 152 actual instances of class 1, the model correctly identifies 135, misclassifying only 17 as class 0. This imbalance suggests that the model prioritizes capturing positive cases, possibly due to an underlying class distribution or the influence of resampling techniques.

The balanced accuracy score of 0.77 provides a fairer evaluation in the presence of class imbalance, confirming that the model's predictive capability is reasonably consistent across both classes. This metric suggests that, despite a tendency to favor positive predictions, the model still maintains a relatively stable ability to distinguish between the two classes.

The second Linear Regression based model to be tested is the **Linear Discriminant Analysis (LDA)**. The LDA is a classification technique that finds a linear combination of features that best separates two or more classes. Unlike Logistic Regression, which models the probability of class membership, LDA assumes that each class follows a Gaussian distribution with equal covariance matrices. It projects the data onto a lower-dimensional space that maximizes the separation between classes, making it particularly effective for classification tasks with well-separated groups. LDA works by computing discriminant functions that define decision boundaries, helping classify new observations based on their feature values. **(27)**
Again, the code and the results are shown below:

```
etc = LinearDiscriminantAnalysis().fit(X_train_s, y_train_s)
y_pred = etc.predict(X_test)

ac = accuracy_score(y_test, y_pred)
g = balanced_accuracy_score(y_test, y_pred)
f1 = f1_score(y_test, y_pred)

print('Accuracy Score: %.2f' % ac)
print('F1-score: %.2f' % f1)
print('G-Mean: %.2f' % g)
print(classification_report(y_pred, y_test))
cm = confusion_matrix(y_test, y_pred)
disp = ConfusionMatrixDisplay(confusion_matrix=cm)
disp.plot(cmap='gray')
plt.grid(False)
plt.show()
```

```
Accuracy Score: 0.78
F1-score: 0.83
G-Mean: 0.79
              precision    recall  f1-score   support

           0       0.81      0.60      0.69       102
           1       0.77      0.91      0.83       148

    accuracy                           0.78       250
   macro avg       0.79      0.75      0.76       250
weighted avg       0.79      0.78      0.77       250
```



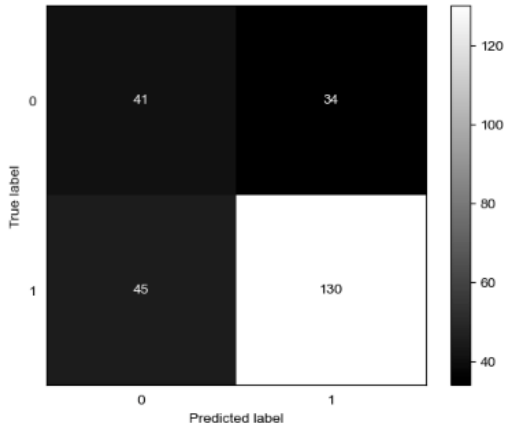
In this case, the LDA model achieves an accuracy of 0.78, slightly improving over Logistic Regression. The F1-score remains at 0.83, indicating strong performance, especially in classifying positive instances. The balanced accuracy of 0.79 shows a slight enhancement in handling class imbalance.
Examining class-specific metrics, class 0 has precision of 0.81 but a recall of only 0.60, meaning that a considerable number of actual negatives are misclassified. On the other hand, class 1 demonstrates higher recall (0.91), ensuring most positive instances are detected. This pattern is confirmed by the confusion matrix, where class 0 has 41 false negatives, while class 1 has only 14 false positives.
The results suggest that LDA improves class separation compared to Logistic Regression, particularly in precision for class 0, but still struggles with false negatives. Further optimization, such as adjusting decision thresholds or incorporating additional feature engineering, could enhance the balance between the two classes.

3.3.2 Decision Tree Based Models

Moving on to decision tree-based models, we start with the **Decision Tree Classifier**. Decision trees work by recursively splitting the data into branches based on feature values, ultimately leading to a decision node. The model creates a series of "if-then" rules, making it highly interpretable and flexible. The results are shown below:

```
etc = DecisionTreeClassifier(random_state=42).fit(X_train_s, y_train_s)
y_pred = etc.predict(X_test)

ac = accuracy_score(y_test, y_pred)
g = balanced_accuracy_score(y_test, y_pred)
f1 = f1_score(y_test, y_pred)

print('Accuracy Score: %.2f' % ac)
print('F1-score: %.2f' % f1)
print('G-Mean: %.2f' % g)
print(classification_report(y_pred, y_test))
cm = confusion_matrix(y_test, y_pred)
disp = ConfusionMatrixDisplay(confusion_matrix=cm)
disp.plot(cmap='gray')
plt.grid(False)
plt.show()
```

```
Accuracy Score: 0.68
F1-score: 0.77
G-Mean: 0.64
              precision    recall  f1-score   support

           0       0.55      0.48      0.51        86
           1       0.74      0.79      0.77       164

    accuracy                           0.68       250
   macro avg       0.64      0.63      0.64       250
weighted avg       0.68      0.68      0.68       250
```



In this case, the Decision Tree Classifier achieves an accuracy of 0.68, which is significantly lower than the previous models. The F1-score is 0.77, and the balanced accuracy is 0.64, indicating weaker generalization, particularly in handling class imbalance.

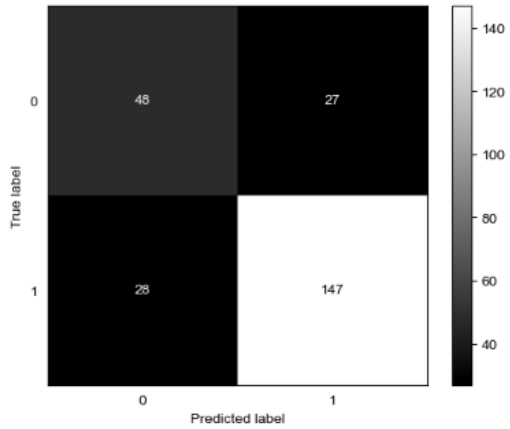
Analyzing class-specific performance, class 0 has a precision of 0.55 and recall of only 0.48, meaning that nearly half of the actual negative instances are misclassified. Conversely, class 1 performs slightly better, with a recall of 0.79, ensuring most positives are captured but at the cost of some false positives. This is reflected in the confusion matrix, where class 0 has 41 false negatives and 34 false positives, while class 1 has 45 false negatives.

The results suggest that the Decision Tree classifier struggles to achieve a balanced performance. The presence of a high number of misclassifications indicates overfitting to training data, leading to poor generalization.

Moving forward with **Random Forest Classifier**, this model builds upon the decision tree concept but addresses its limitations by combining multiple decision trees into a single ensemble model. Each tree in a random forest is trained on a random subset of the data, and when predicting, the model aggregates the outputs from all the trees. This "bagging" technique reduces overfitting, improves generalization, and enhances accuracy compared to a single decision tree. Results are shown below:

```
etc = RandomForestClassifier(random_state=42).fit(X_train_s, y_train_s)
y_pred = etc.predict(X_test)

ac = accuracy_score(y_test, y_pred)
g = balanced_accuracy_score(y_test, y_pred)
f1 = f1_score(y_test, y_pred)

print('Accuracy Score: %.2f' % ac)
print('F1-score: %.2f' % f1)
print('G-Mean: %.2f' % g)
print(classification_report(y_pred, y_test))
cm = confusion_matrix(y_test, y_pred)
disp = ConfusionMatrixDisplay(confusion_matrix=cm)
disp.plot(cmap='gray')
plt.grid(False)
plt.show()
```

```
Accuracy Score: 0.78
F1-score: 0.84
G-Mean: 0.74
              precision    recall  f1-score   support

           0       0.64      0.63      0.64        76
           1       0.84      0.84      0.84       174

    accuracy                           0.78       250
   macro avg       0.74      0.74      0.74       250
weighted avg       0.78      0.78      0.78       250
```



In this case, the Random Forest Classifier achieves an accuracy score of 0.78, showing a substantial improvement compared to the previous decision tree model. The F1-score of 0.84 and G-Mean of 0.74 indicate a well-balanced performance, with a good trade-off between precision and recall, especially for the minority class.

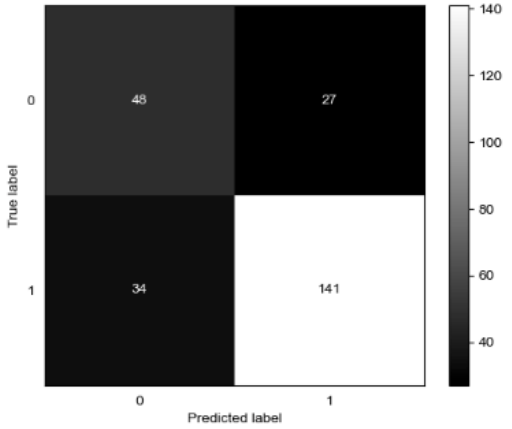
Specifically, class 1, with a precision of 0.84 and recall of 0.84, demonstrates strong performance, accurately identifying positive cases. On the other hand, class 0 shows precision of 0.64 and recall of 0.63, indicating that the model struggles slightly with the negative class but still performs better than the Decision Tree.

The confusion matrix illustrates that the model misclassified 48 true negatives as false positives and 28 false negatives as true positives, which, while non-optimal, is a marked improvement over the Decision Tree. This indicates that the Random Forest model provides better robustness and more reliable predictions, making it a more suitable choice for classification tasks in this dataset.

The next model to be evaluated is the **Gradient Boosting Classifier.** This is an advanced tree-based model that sequentially builds an ensemble of weak learners, typically decision trees, to improve performance. Unlike Random Forest, which trains trees independently, Gradient Boosting trains trees sequentially, with each new tree correcting the errors of the previous ones. This boosting technique allows the model to focus more on difficult-to-classify samples, improving predictive accuracy while maintaining flexibility in handling non-linear relationships. **(28)**
The code and the results are shown below:

```
etc = GradientBoostingClassifier(random_state=42).fit(X_train_s, y_train_s)
y_pred = etc.predict(X_test)

ac = accuracy_score(y_test, y_pred)
g = balanced_accuracy_score(y_test, y_pred)
f1 = f1_score(y_test, y_pred)

print('Accuracy Score: %.2f' % ac)
print('F1-score: %.2f' % f1)
print('G-Mean: %.2f' % g)
print(classification_report(y_pred, y_test))
cm = confusion_matrix(y_test, y_pred)
disp = ConfusionMatrixDisplay(confusion_matrix=cm)
disp.plot(cmap='gray')
plt.grid(False)
plt.show()
```

```
Accuracy Score: 0.76
F1-score: 0.82
G-Mean: 0.72
              precision    recall  f1-score   support

           0       0.64      0.59      0.61        82
           1       0.81      0.84      0.82       168

    accuracy                           0.76       250
   macro avg       0.72      0.71      0.72       250
weighted avg       0.75      0.76      0.75       250
```



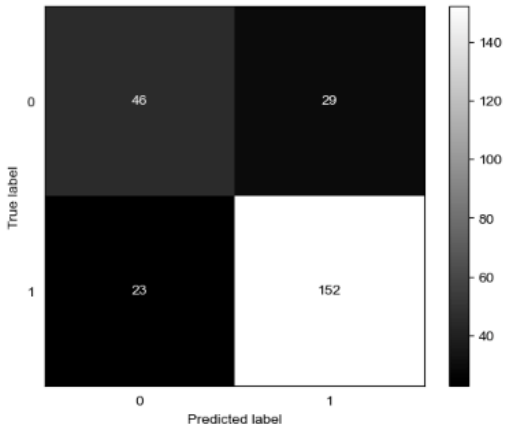
In this case, the model achieves an accuracy score of 0.76, slightly lower than Random Forest but still competitive. The F1-score of 0.82 and G-Mean of 0.72 indicate strong class balance, with a particularly high recall of 0.84 for class 1, showing that the model effectively identifies positive cases.
Looking at precision and recall, class 1 maintains precision of 0.81 and recall of 0.84, highlighting good predictive power for identifying creditworthy individuals. However, class 0 shows precision of 0.64 and recall of 0.59, meaning the model struggles more with detecting non-creditworthy cases, leading to a higher number of false negatives.
The confusion matrix reveals that 48 true negatives were misclassified as false positives, and 34 false negatives were identified as positives. While the performance is slightly less balanced than Random Forest, Gradient Boosting's adaptive learning mechanism still makes it a powerful model.

The last Decision Tree based model is the **Extra Trees Classifier (Extremely Randomized Trees)**, which is an ensemble learning method similar to Random Forest but introduces additional randomness in tree construction. Unlike Random Forest, which selects the best split based on information gain or Gini impurity, Extra Trees randomly selects split points. This approach reduces variance and enhances generalization. **(29)**

The code and the results for this model are shown below:

```
etc = ExtraTreesClassifier(random_state=42).fit(X_train_s, y_train_s)
y_pred = etc.predict(X_test)

ac = accuracy_score(y_test, y_pred)
g = balanced_accuracy_score(y_test, y_pred)
f1 = f1_score(y_test, y_pred)

print('Accuracy Score: %.2f' % ac)
print('F1-score: %.2f' % f1)
print('G-Mean: %.2f' % g)
print(classification_report(y_pred, y_test))
cm = confusion_matrix(y_test, y_pred)
disp = ConfusionMatrixDisplay(confusion_matrix=cm)
disp.plot(cmap='gray')
plt.grid(False)
plt.show()
```

```
Accuracy Score: 0.79
F1-score: 0.85
G-Mean: 0.74
              precision    recall  f1-score   support

           0       0.61      0.67      0.64        69
           1       0.87      0.84      0.85       181

    accuracy                           0.79       250
   macro avg       0.74      0.75      0.75       250
weighted avg       0.80      0.79      0.79       250
```



In this case, the model achieves an accuracy score of 0.79, making it one of the best-performing models so far. The F1-score of 0.85 and G-Mean of 0.74 indicate strong predictive power. Notably, class 1 (creditworthy individuals) shows an impressive recall of 0.84 and precision of 0.87, demonstrating the model's ability to correctly identify most positive cases.

However, class 0 (non-creditworthy individuals) has precision of 0.61 and recall of 0.67, suggesting some difficulty in identifying negative cases. The confusion matrix highlights this issue, with 29 false positives and 23 false negatives, meaning that while the model excels in identifying creditworthy individuals, it still misclassifies a significant number of non-creditworthy ones.

Overall, Extra Trees provides strong results, balancing precision and recall effectively.

3.3.3 Other Models

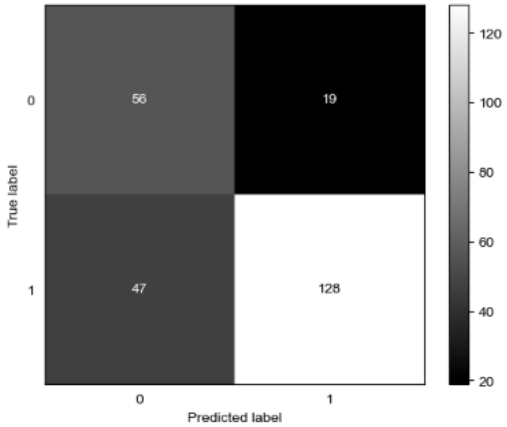
After analyzing both Linear Tree-based models, the next step is to explore classification algorithms that follow different underlying principles. These models employ distinct strategies for decision-making, ranging from probabilistic methods to distance-based approaches and neural networks.

The first one to be analyzed is the **Adaptive Boosting (AdaBoost) Classifier**, which is an ensemble learning method designed to improve the predictive accuracy of weak classifiers by combining multiple iterations of simple base learners. The algorithm assigns different weights to training instances, focusing on those that were previously misclassified. With each iteration, the model updates these weights, prioritizing difficult cases and refining the decision boundary. Typically, AdaBoost uses decision trees as base classifiers, though it can work with various weak learners. **(30)**

The code and the results are shown below:

```
etc = AdaBoostClassifier(random_state=42).fit(X_train_s, y_train_s)
y_pred = etc.predict(X_test)

ac = accuracy_score(y_test, y_pred)
g = balanced_accuracy_score(y_test, y_pred)
f1 = f1_score(y_test, y_pred)

print('Accuracy Score: %.2f' % ac)
print('F1-score: %.2f' % f1)
print('G-Mean: %.2f' % g)
print(classification_report(y_pred, y_test))
cm = confusion_matrix(y_test, y_pred)
disp = ConfusionMatrixDisplay(confusion_matrix=cm)
disp.plot(cmap='gray')
plt.grid(False)
plt.show()
```

```
Accuracy Score: 0.74
F1-score: 0.80
G-Mean: 0.74
              precision    recall  f1-score   support

           0       0.75      0.54      0.63       103
           1       0.73      0.87      0.80       147

    accuracy                           0.74       250
   macro avg       0.74      0.71      0.71       250
weighted avg       0.74      0.74      0.73       250
```

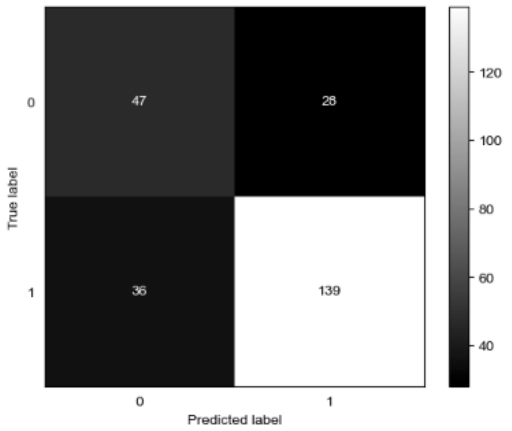


The evaluation of the AdaBoost Classifier on this dataset yielded an accuracy score of 0.74, an F1-score of 0.80, and a balanced accuracy of 0.74. The model demonstrated strong recall for class 1 (0.87), suggesting it effectively captures positive cases. However, its recall for class 0 (0.54) indicates difficulty in correctly identifying negative instances, leading to a noticeable class imbalance. This can be observed in the confusion matrix, where misclassifications are primarily concentrated in false negatives. Compared to previous models, AdaBoost maintains competitive overall performance but may struggle with datasets that exhibit significant class asymmetry.

Moving forward, the next model is the **Extreme Gradient Boosting (XGBoost) Classifier**. This is a powerful, optimized gradient boosting algorithm known for its efficiency and accuracy in classification and regression tasks. Unlike traditional boosting models, XGBoost incorporates regularization techniques, such as L1 (Lasso) and L2 (Ridge) penalties, to prevent overfitting. It builds decision trees sequentially, with each new tree correcting errors made by the previous ones. **(31)**
Below there are the code and the results:

```
etc = XGBClassifier(random_state=42).fit(X_train_s, y_train_s)
y_pred = etc.predict(X_test)

ac = accuracy_score(y_test, y_pred)
g = balanced_accuracy_score(y_test, y_pred)
f1 = f1_score(y_test, y_pred)

print('Accuracy Score: %.2f' % ac)
print('F1-score: %.2f' % f1)
print('G-Mean: %.2f' % g)
print(classification_report(y_pred, y_test))
cm = confusion_matrix(y_test, y_pred)
disp = ConfusionMatrixDisplay(confusion_matrix=cm)
disp.plot(cmap='gray')
plt.grid(False)
plt.show()
```

```
Accuracy Score: 0.74
F1-score: 0.81
G-Mean: 0.71
              precision    recall  f1-score   support

           0       0.63      0.57      0.59        83
           1       0.79      0.83      0.81       167

    accuracy                           0.74       250
   macro avg       0.71      0.70      0.70       250
weighted avg       0.74      0.74      0.74       250
```

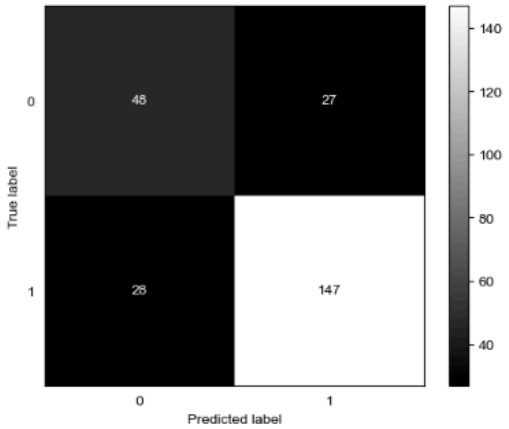


The evaluation of the XGBoost classifier resulted in an accuracy score of 0.74, an F1-score of 0.81, and a balanced accuracy of 0.71. The recall for class 1 (0.83) suggests strong predictive power for positive cases, while recall for class 0 (0.57) indicates some difficulty in correctly identifying negative instances. This is reflected in the confusion matrix, where false negatives remain relatively high. Compared to AdaBoost, XGBoost demonstrates a slight improvement in handling class imbalances, but its overall performance remains similar.

The third model is the **Categorical Boosting (CatBoost) Classifier**, that is a gradient boosting algorithm designed to handle categorical features efficiently without requiring extensive preprocessing. Unlike other boosting models, CatBoost implements ordered boosting, which reduces overfitting, and employs symmetric trees, leading to faster training and inference times. Its ability to handle categorical variables directly makes it particularly well-suited for structured data applications, often outperforming other boosting algorithms such as XGBoost in certain tasks. **(32)**
The results are shown below:

```
etc = CatBoostClassifier(random_state=42).fit(X_train_s, y_train_s)
y_pred = etc.predict(X_test)

ac = accuracy_score(y_test, y_pred)
g = balanced_accuracy_score(y_test, y_pred)
f1 = f1_score(y_test, y_pred)

print('Accuracy Score: %.2f' % ac)
print('F1-score: %.2f' % f1)
print('G-Mean: %.2f' % g)
print(classification_report(y_pred, y_test))
cm = confusion_matrix(y_test, y_pred)
disp = ConfusionMatrixDisplay(confusion_matrix=cm)
disp.plot(cmap='gray')
plt.grid(False)
plt.show()
```

```
Accuracy Score: 0.78
F1-score: 0.84
G-Mean: 0.74
              precision    recall  f1-score   support

           0       0.64      0.63      0.64        76
           1       0.84      0.84      0.84       174

    accuracy                           0.78       250
   macro avg       0.74      0.74      0.74       250
weighted avg       0.78      0.78      0.78       250
```

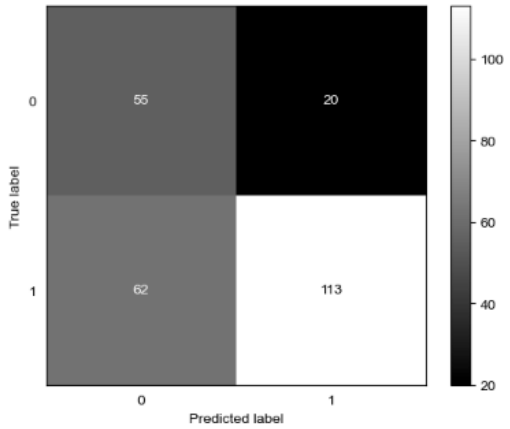


The results obtained from the CatBoost classifier indicate an accuracy score of 0.78, an F1-score of 0.84, and a balanced accuracy of 0.74. The recall for class 1 (0.84) is significantly high, confirming the model's effectiveness in identifying positive cases. The recall for class 0 (0.63) is moderate, indicating that while the model correctly classifies most positive instances, it still struggles with a portion of the negative cases. The confusion matrix highlights that false negatives are relatively lower compared to other boosting models, suggesting better balance in classification.
Compared to XGBoost and AdaBoost, CatBoost achieves a slightly higher overall accuracy and a strong F1-score, making it a competitive option.

Moving on there is the **K-Nearest Neighbors (KNN)** algorithm is an instance-based learning model that classifies data points based on their proximity to labeled examples in the training set. It determines the class of a new observation by calculating the majority class among its k-nearest neighbors. KNN is a non-parametric model, meaning it does not make assumptions about the data distribution, making it flexible but computationally expensive for large datasets. **(33)**
The results are shown below:

```
etc = KNeighborsClassifier().fit(X_train_s, y_train_s)
y_pred = etc.predict(X_test)

ac = accuracy_score(y_test, y_pred)
g = balanced_accuracy_score(y_test, y_pred)
f1 = f1_score(y_test, y_pred)

print('Accuracy Score: %.2f' % ac)
print('F1-score: %.2f' % f1)
print('G-Mean: %.2f' % g)
print(classification_report(y_pred, y_test))
cm = confusion_matrix(y_test, y_pred)
disp = ConfusionMatrixDisplay(confusion_matrix=cm)
disp.plot(cmap='gray')
plt.grid(False)
plt.show()
```

```
Accuracy Score: 0.67
F1-score: 0.73
G-Mean: 0.69
              precision    recall  f1-score   support

           0       0.73      0.47      0.57       117
           1       0.65      0.85      0.73       133

    accuracy                           0.67       250
   macro avg       0.69      0.66      0.65       250
weighted avg       0.69      0.67      0.66       250
```

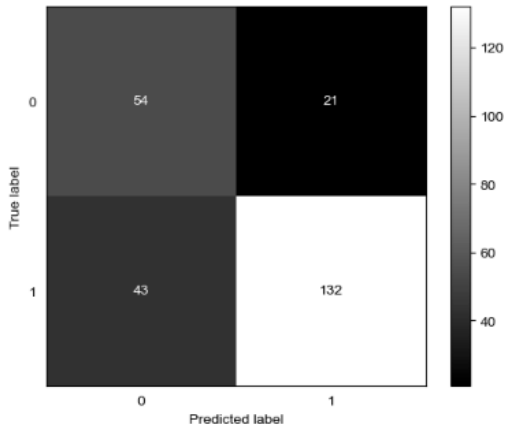


The results from the KNN Classifier indicate an accuracy score of 0.67, an F1-score of 0.73, and a balanced accuracy of 0.69. The recall for class 1 (0.85) is relatively high, suggesting that the model is capable of identifying positive cases effectively. However, the recall for class 0 (0.47) is significantly lower, indicating a tendency to misclassify negative instances as positive. The confusion matrix confirms this imbalance, showing a high number of false negatives, which can be problematic in scenarios where correctly identifying negative cases is crucial.
Compared to previous models, KNN has one of the lowest accuracy scores, likely due to its sensitivity to noise and high-dimensional data. Its reliance on distance-based classification can be affected by variations in feature scaling and the curse of dimensionality. While KNN may be useful for small, well-separated datasets, it appears to struggle with more complex decision boundaries.

The next model on the line is the **Support Vector Classifier (SVC)**, which is a ML algorithm based on Support Vector Machines (SVMs), designed to find an optimal hyperplane that maximizes the margin between different classes. It is particularly effective for binary classification tasks, especially when the data is not perfectly linearly separable. SVC uses kernel functions to project data into higher-dimensional spaces, enabling it to capture complex decision boundaries. **(34)**
The results are shown below:

```
etc = SVC(random_state=42).fit(X_train_s, y_train_s)
y_pred = etc.predict(X_test)

ac = accuracy_score(y_test, y_pred)
g = balanced_accuracy_score(y_test, y_pred)
f1 = f1_score(y_test, y_pred)

print('Accuracy Score: %.2f' % ac)
print('F1-score: %.2f' % f1)
print('G-Mean: %.2f' % g)
print(classification_report(y_pred, y_test))
cm = confusion_matrix(y_test, y_pred)
disp = ConfusionMatrixDisplay(confusion_matrix=cm)
disp.plot(cmap='gray')
plt.grid(False)
plt.show()
```

```
Accuracy Score: 0.74
F1-score: 0.80
G-Mean: 0.74
              precision    recall  f1-score   support

           0       0.72      0.56      0.63        97
           1       0.75      0.86      0.80       153

    accuracy                           0.74       250
   macro avg       0.74      0.71      0.72       250
weighted avg       0.74      0.74      0.74       250
```

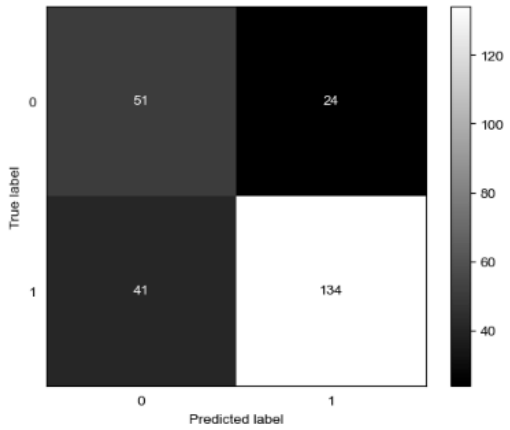


The results of the SVC model reveal an accuracy score of 0.74, an F1-score of 0.80, and a balanced accuracy of 0.74. The recall for class 1 (0.86) is significantly higher than for class 0 (0.56), suggesting that the model is better at identifying positive instances while struggling with negative cases. The confusion matrix shows that false negatives are more prevalent than false positives, meaning that many class 0 instances were misclassified as class 1.
Compared to simpler models like Decision Trees and KNN, SVC provides a more balanced performance, benefiting from its ability to handle complex decision boundaries. However, it does not outperform ensemble methods like Random Forest or Gradient Boosting, which leverage multiple weak learners for more stable predictions.

The next model is the **Multi-Layer Perceptron (MLPClassifier)** is a neural network-based model designed for classification tasks. It consists of multiple layers of perceptrons, using an activation function to learn non-linear patterns in the data. Unlike traditional machine learning models, MLP uses backpropagation and gradient descent optimization to adjust weights and improve classification accuracy over multiple iterations. It is well-suited for complex problems where relationships between features are not easily captured by linear or tree-based models. **(35)**
The results and the code for this model are shown below:

```
etc = MLPClassifier(random_state=42).fit(X_train_s, y_train_s)
y_pred = etc.predict(X_test)

ac = accuracy_score(y_test, y_pred)
g = balanced_accuracy_score(y_test, y_pred)
f1 = f1_score(y_test, y_pred)

print('Accuracy Score: %.2f' % ac)
print('F1-score: %.2f' % f1)
print('G-Mean: %.2f' % g)
print(classification_report(y_pred, y_test))
cm = confusion_matrix(y_test, y_pred)
disp = ConfusionMatrixDisplay(confusion_matrix=cm)
disp.plot(cmap='gray')
plt.grid(False)
plt.show()
```

```
Accuracy Score: 0.74
F1-score: 0.80
G-Mean: 0.72
              precision    recall  f1-score   support

           0       0.68      0.55      0.61        92
           1       0.77      0.85      0.80       158

    accuracy                           0.74       250
   macro avg       0.72      0.70      0.71       250
weighted avg       0.73      0.74      0.73       250
```

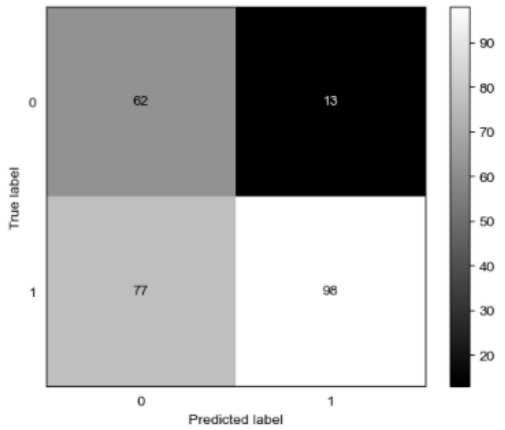


In this evaluation, the MLP model achieved an accuracy score of 0.74, with an F1-score of 0.80 and a balanced accuracy of 0.72. While these results are comparable to SVC and ensemble methods, the model exhibits imbalances in class-specific recall. Class 1 shows a recall of 0.85, indicating strong performance in identifying positive instances, while class 0 has a recall of 0.55, suggesting misclassification of negative cases. This pattern is consistent with models that struggle with class imbalances.
The confusion matrix highlights the presence of false negatives, which means the model often misclassifies actual negative instances as positive.

The last model to be evaluated is the **Gaussian Naïve Bayes (GaussianNB)** classifier is a probabilistic model based on Bayes' theorem, assuming that features follow a normal (Gaussian) distribution. It is particularly effective for high-dimensional data and is computationally efficient due to its independence assumption between features. **(36)**
Below are the results:

```
etc = GaussianNB().fit(X_train_s, y_train_s)
y_pred = etc.predict(X_test)

ac = accuracy_score(y_test, y_pred)
g = balanced_accuracy_score(y_test, y_pred)
f1 = f1_score(y_test, y_pred)

print('Accuracy Score: %.2f' % ac)
print('F1-score: %.2f' % f1)
print('G-Mean: %.2f' % g)
print(classification_report(y_pred, y_test))
cm = confusion_matrix(y_test, y_pred)
disp = ConfusionMatrixDisplay(confusion_matrix=cm)
disp.plot(cmap='gray')
plt.grid(False)
plt.show()
```

```
Accuracy Score: 0.64
F1-score: 0.69
G-Mean: 0.69
              precision    recall  f1-score   support

           0       0.83      0.45      0.58       139
           1       0.56      0.88      0.69       111

    accuracy                           0.64       250
   macro avg       0.69      0.66      0.63       250
weighted avg       0.71      0.64      0.63       250
```



In this evaluation, GaussianNB achieved an accuracy score of 0.64, with an F1-score of 0.69 and a balanced accuracy of 0.69. These results indicate weaker overall performance compared to more complex models such as Random Forest, Gradient Boosting, or SVC. The precision for class 0 (0.83) is relatively high, indicating that when the model predicts a negative instance, it is often correct. However, the recall for class 0 is only 0.45, meaning that a significant portion of actual negative cases are misclassified. Conversely, class 1 has a recall of 0.88, showing the model's ability to correctly identify positive instances but at the cost of misclassifying many negative ones.
The confusion matrix further highlights this imbalance, as 77 false negatives indicate that the model is struggling with distinguishing between the two classes effectively. GaussianNB tends to work best when features are truly independent and normally distributed, which is not always the case in real-world applications.

3.4 Performance Comparison

Below there are the classification results of the tested models, ranked in descending order based on their accuracy scores.

	Classifier	Accuracy Score	G-Mean	F1-Score
2	ExtraTreesClassifier	0.792	0.740952	0.853933
0	RandomForestClassifier	0.780	0.740000	0.842407
12	CatBoostClassifier	0.780	0.740000	0.842407
11	LinearDiscriminantAnalysis	0.780	0.789524	0.829721
3	LogisticRegression	0.772	0.772381	0.825688
1	GradientBoostingClassifier	0.756	0.722857	0.822157
4	XGBClassifier	0.744	0.710476	0.812865
6	SVC	0.744	0.737143	0.804878
10	MLPClassifier	0.740	0.722857	0.804805
8	AdaBoostClassifier	0.736	0.739048	0.795031
7	DecisionTreeClassifier	0.684	0.644762	0.766962
5	KNeighborsClassifier	0.672	0.689524	0.733766
9	GaussianNB	0.640	0.693333	0.685315

The results of the model evaluation reveal clear patterns in the performance of different classification algorithms. Tree-based ensemble methods, such as ExtraTreesClassifier, RandomForestClassifier, and CatBoostClassifier, emerged as the top performers, demonstrating the highest accuracy and F1-scores. The success of these models lies in capturing nonlinear relationships, handling imbalanced data, and offering feature importance insights, making them ideal for credit risk assessment. These models excel at learning complex interactions between financial and demographic variables, which are key determinants of creditworthiness. ExtraTreesClassifier, in particular, outperformed the other models, likely due to its ability to reduce variance by randomly selecting splits, making it more resilient to noise in the data.

Linear models such as Logistic Regression and Linear Discriminant Analysis also delivered strong results, suggesting a degree of linear separability in the dataset. LDA slightly outperformed Logistic Regression by better differentiating between creditworthy and non-creditworthy clients. However, these models can struggle with complex interactions found in real-world credit data, where financial behavior is often influenced by multiple interdependent factors.

Boosting models like Gradient Boosting, AdaBoost, and XGBoost showed competitive performance, benefiting from iterative learning and the ability to correct misclassifications. However, their effectiveness is highly dependent on hyperparameter tuning and handling noisy features, which may explain their slightly lower ranking compared to bagging methods. While they can achieve high accuracy, their complexity and increased computational requirements may pose challenges in large-scale credit scoring applications.

Among other algorithms, SVC, MLPClassifier, and KNeighborsClassifier had mixed results. SVC performed reasonably well but may have struggled with overlapping class distributions, which are common in credit risk datasets. MLPClassifier, a neural network-based model, required deeper tuning for optimal performance and may have been hindered by the dataset size. KNeighborsClassifier, sensitive to feature scaling and categorical variables, ranked lower due to inefficiency in high-dimensional spaces, making it less suitable for credit scoring tasks.

Finally, Gaussian Naïve Bayes ranked lowest, likely due to its assumption of independent features, which does not hold in credit scoring data where financial and demographic variables are highly correlated. This limitation makes it an unsuitable choice for modeling credit risk, where relationships between variables play a crucial role in predicting default probability.

Overall, Extra Trees and Random Forest emerged as the most reliable models, offering strong predictive power while maintaining interpretability, a key factor in financial decision-making. These models provide a balance between performance and transparency, which is essential in credit scoring, where regulatory compliance and explainability are as important as accuracy.

It is also important to consider that, while the highest accuracy score achieved in this study was 0.792, this should not necessarily be viewed as low in absolute terms. The dataset used was relatively limited in scope, containing only 1,000 individuals and 20 variables, whereas real-world financial institutions have access to datasets with tens or even hundreds of thousands of clients and a far broader range of financial, behavioral, and transactional variables. With a larger dataset and richer features, machine learning models can achieve significantly higher predictive power, further improving their ability to assess creditworthiness accurately. Nonetheless, the results presented here provide valuable insights into model selection for credit scoring and demonstrate how machine learning can effectively support risk evaluation in financial decision-making.

Chapter 4: Advantages and Issues in the Implementation of AI in the Banking Risk Assessment Process

Artificial Intelligence is becoming increasingly important in the banking industry, revolutionizing traditional processes and enhancing the efficiency and accuracy of operations. Today, AI's capabilities are being harnessed across various domains within banking, from customer service enhancements with chatbots to sophisticated fraud detection systems and intricate risk management solutions. The technology's ability to process vast amounts of data and learn from patterns makes it ideally suited to address the complex requirements of modern banking.

As of 2024, it is still early days in AI's diffusion into the world of risk and compliance. In fact, only 30% of banks and other financial institutions are actively using or trialing AI in compliance and risk management, with 9% being active users and 21% currently in the trial or pilot phase. Meanwhile, just under half of the firms are considering its adoption, and about 21% have not pursued AI integration.

As shown in the graph below, the major adopters of AI technologies are Banks with 40% either using or experimenting with them, followed by Fintech companies at 36%.

In contrast, the insurance, asset, and wealth management sectors are lagging slightly behind in AI adoption.

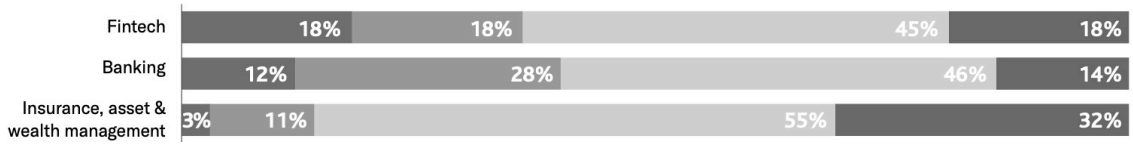


Figure 28. Illustrate the current level of implementation of ai for the purpose of compliance or risk management across different types of financial companies.
Source: Berry, K. (2024). Navigating the AI Landscape: Insights from Compliance and Risk Management Leaders. Moody's Analytics, pp. 07.

Notably, larger companies are significantly more engaged in using or trialing AI, with 42% compared to only 23% of smaller companies.

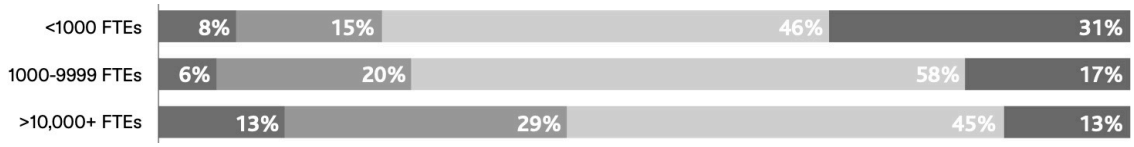


Figure 29. Illustrate the current level of implementation of ai for the purpose of compliance or risk management across companies with different sizes. FTE stands for "Full-Time Equivalent employees of a company.
Source: Berry, K. (2024). Navigating the AI Landscape: Insights from Compliance and Risk Management Leaders. Moody's Analytics, pp. 07.

This disparity suggests that organizations with larger head-counts and more substantial budgets are leveraging their resources to facilitate a shift towards AI, aiming for efficiency gains, performance standardization, and reductions in headcount.

This strategic move underscores the transformative impact AI is poised to have on the financial services industry, reshaping how institutions manage risk and compliance in an increasingly digital world.

In the evolving landscape of artificial intelligence adoption within corporate environments, the quality of data greatly affects a company's ability to use artificial intelligence (AI) effectively. Good data organization and quality are essential for successfully implementing AI. However, in this context, only a minority of financial institutions, approximately 2%, report having highly refined data systems that integrate seamlessly into decision-making processes, offering extensive detail and broad applicability.
A further 12% of institutions have managed to maintain clean, well-organized data, regularly monitored for quality, though the breadth and depth of this data are only moderately extensive. Another 19% have their data in good order, with routine checks in place, although the depth of the data remains somewhat restricted.
However, the predominant data maturity levels identified paint a less ideal picture. A substantial 44% of responses indicate data is "Inconsistent"—while structured, such data suffers from frequent irregularities that demand regular manual correction, thus limiting its utility and scope. Even more concerning is the "Fragmented" category, comprising 23% of the responses, where data is poorly organized and significantly disjointed, necessitating extensive efforts to cleanse and make it serviceable.
This sheds light on a fundamental barrier to the broader adoption of AI in risk and compliance functions. The prevalence of suboptimal data maturity levels emphasizes the critical need for firms to enhance their data governance strategies. By improving data management practices, businesses can better position themselves to unlock the transformative potential of artificial intelligence, thereby optimizing their operational efficiencies and strategic capabilities in the digital age.

As the integration of artificial intelligence within corporate risk and compliance sectors continues to evolve, understanding its current deployment offers crucial insights into its potential transformative effects. With data volumes growing exponentially, it's unsurprising that 63% of organizations, whether actively using or experimenting with AI, leverage it primarily for data analysis and interpretation. This function is particularly vital in banking for risk management and fraud prevention. Concurrently, applications such as automation, screening, and regulatory compliance are expanding as AI adoption becomes more widespread.
The diversity of AI models employed across organizations mirrors the varied requirements these tools meet, ranging from statistical or stochastic models designed for anomaly detection and forecasting, to traditional language models suited for textual analysis tasks like sentiment analysis and named entity recognition. Machine learning (ML) models are utilized to handle structured data for predictions and clustering, while generative language models produce contextually relevant, extended text passages. The average organization engages with approximately 1.8 different AI models, underscoring the multifaceted nature of AI technology. A noticeable trend is the shift toward exploring newer models like traditional language models and generative models, likely facilitated by recent technological advancements enhancing accessibility.

The goals behind AI implementation and the resulting outcomes shed light on its expansive potential within risk and compliance frameworks. Organizations are not only seeking to boost efficiency but also to enhance quality, with 91% of AI adopters reporting significant or moderate improvements in their operations.

The detailed benefits include:

- Efficiency Improvements: Automation of routine tasks, such as anti-money laundering operations, has significantly reduced workloads, freeing teams to focus on more strategic endeavors.
- Enhanced Risk Identification: AI has refined the accuracy and timeliness of risk detection processes, thereby informing more effective decision-making.
- Tighter Fraud Detection:Enhanced capabilities in fraud detection significantly bolster cybersecurity measures.
- Cost Savings and Error Reduction:AI's role in minimizing errors and inefficiencies contributes to lower operational costs and reduced need for extensive physical infrastructure.
- Data Processing and Quality Enhancements: The optimization of data collection and analysis through AI offers deeper and more comprehensive insights.

Despite the tangible benefits AI brings to risk management and compliance, the broader corporate perspective reflects a mix of caution and optimism. Around one in four organizations views Large Language Models (LLMs) like ChatGPT favorably, yet an equal percentage actively resists their adoption due to application concerns. Interestingly, the largest portion of organizations has not yet decided on a policy regarding these tools, indicating the emerging nature of this technology.

Sector-specific responses vary, with fintechs displaying a markedly higher openness to LLMs compared to more conservative banking sectors, which are particularly sensitive to reputational risks and data privacy. Larger organizations, especially those with workforces exceeding 10,000, are more likely to have developed explicit AI policies compared to smaller entities.

The interest in developing bespoke "co-pilot" LLMs, customized for proprietary data to sidestep privacy and security issues, spans organizations of all sizes. Despite some reservations, the overarching sentiment toward AI in risk and compliance is overwhelmingly positive, with 82% of participants confident in AI's significant future benefits. This positive outlook is consistent across all types of organizations and is reinforced by the recognition of efficiency and speed as primary benefits derived from AI use.

This complex yet optimistic view across the professional landscape suggests a cautious but inevitable trajectory toward deeper AI integration, promising profound impacts on risk and compliance practices. As familiarity with AI's capabilities grows and technologies become more embedded, more nuanced benefits like reduced false positives and enhanced accuracy are expected to become more apparent, reshaping risk and compliance operations fundamentally.

Artificial Intelligence (AI) is ideally situated to confront the escalating challenges that many risk and compliance teams face today. In an era where efficiency is paramount and teams are continuously being streamlined, many professionals feel overwhelmed by the burden of rapidly expanding datasets and constantly evolving regulatory demands. AI offers a promising solution to the complex problem of doing more with less, as reflected in the first chart, which indicates significant perceived advantages such as improved efficiency and speed noted by 72% of respondents, with 25% considering efficiency the top advantage.

However, alongside the recognized benefits, there are substantial concerns associated with the deployment of AI in risk and compliance contexts. As depicted in the second chart, data privacy and transparency in decision-making are leading worries, each cited by 55% of respondents. These concerns are not just theoretical but are practical issues for those currently using or testing AI technologies. The fear that organizations might rely too heavily on AI, potentially at the expense of human judgment, is a significant concern, with over-reliance on AI marked as the primary worry by 17% of respondents. This anxiety highlights the need for balance, ensuring that AI supports rather than replaces human decision-making.

Addressing these concerns requires concrete measures to foster trust and confidence in AI applications within risk and compliance frameworks. The third chart illustrates the necessary safeguards, with 51% of professionals emphasizing the need for transparency and explainability in AI decisions. Furthermore, 48% advocate for a comprehensive AI governance framework, and 45% see regular testing for bias and fairness as critical.

These findings suggest a path forward that involves careful integration of AI with robust oversight mechanisms to ensure that AI tools enhance rather than complicate the regulatory and risk management landscapes. It is clear that while AI has the potential to significantly aid risk and compliance functions, the transition must be managed thoughtfully to mitigate risks and maximize the technology's benefits. Ensuring that AI remains a tool under human control, rather than a decision-maker, will be crucial in maintaining the effectiveness and integrity of risk and compliance strategies in the future. **(37)**

Conclusion

Artificial Intelligence is reshaping the landscape of financial risk assessment, introducing new paradigms that challenge traditional methodologies. This thesis has highlighted the profound impact of AI-driven models in banking, not just as tools for automation but as transformative forces capable of redefining decision-making processes. By leveraging vast amounts of data and detecting intricate patterns beyond human capability, AI has emerged as an essential asset in credit scoring, fraud detection, and risk mitigation. However, its integration into financial services is not without obstacles, and its adoption raises fundamental questions about transparency, governance, and trust.

The practical application of machine learning models in this study underscored both the promise and the complexity of AI in banking. The results revealed the superiority of tree-based ensemble methods, particularly Extra Trees and Random Forest, in capturing the non-linear and interdependent nature of financial variables. At the same time, traditional statistical models, such as Logistic Regression and LDA, while reliable, demonstrated clear limitations in dealing with the complexity of real-world financial behavior. The contrast between these approaches is not merely a technical distinction but a reflection of a broader shift in financial risk assessment—one that moves away from rigid, rule-based frameworks toward adaptive, data-driven intelligence.

Yet, despite AI's capacity to enhance efficiency and accuracy, this study also highlighted the fragility of its foundations. The quality and structure of financial data remain a major bottleneck, with many institutions struggling with fragmented, inconsistent, or biased datasets that hinder AI's full potential. Furthermore, while machine learning models can outperform traditional techniques, their lack of explainability presents a significant challenge in a regulatory environment where financial decisions must be transparent and justifiable. This paradox, between AI's predictive power and the necessity for human interpretability, remains an unresolved tension that will shape the future of AI adoption in banking.

Looking ahead, AI's role in financial risk management will continue to expand, but its success will depend not only on technological advances but also on how institutions navigate the delicate balance between automation and accountability. The promise of AI lies not just in its ability to optimize decision-making but in its potential to redefine the very nature of risk itself. The challenge is no longer whether AI will be integrated into banking (it already is) but whether financial institutions can do so responsibly, ensuring that efficiency does not come at the cost of fairness and trust.

As AI continues to evolve, it forces us to rethink fundamental assumptions about how financial risks are evaluated, who makes those decisions, and to what extent we are willing to rely on algorithmic judgment in matters of economic stability. This thesis does not offer definitive answers to these questions but instead underscores their urgency. The future of banking will not be determined by algorithms alone, but by the decisions we make about how, when, and why to use them.

References

Chapter 1:

(1) *What is Risk?* | Investor.gov. (n.d.). https://www.investor.gov/introduction-investing/investing-basics/what-risk

(2) Chen, J. (2024, May 16). *Risk: What it means in investing, how to measure and manage it*. Investopedia. https://www.investopedia.com/terms/r/risk.asp

(3) "*Market Risk*" | Council of Europe Development Bank. https://coebank.org/en/investor-relations/risk-management/market-risk

(4) *What is Financial Risk & How to Assess It* | Allianz Trade US. (n.d.). Corporate. https://www.allianz-trade.com/en_US/insights/how-to-assess-financial-risk.html

(5) Penza, D. (2023, October 22). | *Risk and Risk Management* [PowerPoint slides]. LUISS Guido Carli.

(6) "*Credit Risk*" | Council of Europe Development Bank. https://coebank.org/en/investor-relations/risk-management/credit-risk

(7) "*Liquidity Risk*" | Council of Europe Development Bank. https://coebank.org/en/investor-relations/risk-management/liquidity-risk

(8) "*Operational Risk*" | Council of Europe Development Bank. https://coebank.org/en/investor-relations/risk-management/operational-risk

(9) *What is Risk Management?* | IBM. (n.d.). https://www.ibm.com/topics/risk-management

(10) Unit21. (2024, July 23). | Risk Management in Banking 2024: *Types + Best Practices for Financial Institution Mitigation.* https://www.unit21.ai/blog/risk-management-in-banking

(11) *What are the Essential Techniques of Risk Management - Human Resources, Diversity and Inclusion* | CSUF. (n.d.). https://hr.fullerton.edu/risk-management/information-and-document-requests/information-management/essential-techniques-of-risk-management.php

(12) Columbia AI. (2023, October 3). CU-CAI. | *Artificial Intelligence (AI) vs. Machine Learning* https://ai.engineering.columbia.edu/ai-vs-machine-learning/

(13) Blagoj, D., Chrysi, T., & Uros, K. (2020). | *Historical evolution of artificial intelligence: Analysis of the three main paradigm shifts in AI. Joint Research Centre*, 7,9,11,12. https://doi.org/10.2760/801580

(14) "*History of Artificial Intelligence.*" | Science in the News, Harvard University. https://sitn.hms.harvard.edu/flash/2017/history-artificial-intelligence/.

Chapter 2:

(15) Medallia. (n.d.). *Real Time Text Analytics Software - Medallia*. Medallia. https://monkeylearn.com/machine-learning/

(16) GeeksforGeeks. (2024, February 8). *How does Machine Learning Works?* GeeksforGeeks. https://www.geeksforgeeks.org/how-does-machine-learning-works/

(17) Ramakrishnan, J. (2023, October 26). *The evolution of risk management in banking*. https://www.linkedin.com/pulse/evolution-risk-management-banking-jambunathan-ramakrishnan-nuhif

(18) Aziz, S., & Dowling, M. (2018). Machine learning and AI for risk management. In *Palgrave studies in digital business & enabling technologies* (pp. 33–50). https://doi.org/10.1007/978-3-030-02330-0_3

(19) Inc, S. S. (2024, March 14). Machine learning for credit scoring: Benefits, models, and implementation challenges. *Svitla Systems, Inc*. https://svitla.com/blog/machine-learning-for-credit-scoring

(20) R, Ali. (2023, October 5). *Enhancing Market Risk Models with Machine Learning Techniques*. https://www.linkedin.com/pulse/enhancing-market-risk-models-machine-learning-techniques-ali-h-rizvi

(21) Lutkevich, B. (2024, August 26). *expert system*. Enterprise AI. https://www.techtarget.com/searchenterpriseai/definition/expert-system#:~:text=An%20expert%20system%20is%20a,experience%20in%20a%20particular%20field.

(22) Wang, L., Cheng, Y., Xiang, A., Zhang, J., & Yang, H. (2024b, June 14). *Application of natural language processing in financial risk detection*. arXiv.org. https://arxiv.org/abs/2406.09765

(23) purpleSlate. (2023, November 12). The Comprehensive Guide to Understanding Generative AI. *Medium*. https://medium.com/@social_65128/the-comprehensive-guide-to-understanding-generative-ai-c06bbf259786

(24) Wang, Yanqing, Generative AI in Operational Risk Management: Harnessing the Future of Finance (May 17, 2023). Available at SSRN: https://ssrn.com/abstract=4452504 or http://dx.doi.org/10.2139/ssrn.4452504

Chapter 3:

DATASET: Hans Hofmann. (2024). Credit Risk Dataset [Data set]. Kaggle. https://doi.org/10.34740/KAGGLE/DSV/9017171

(25) GeeksforGeeks. (2024, August 1). *Libraries in Python*. GeeksforGeeks. https://www.geeksforgeeks.org/libraries-in-python/

(26) Yeo, I. K., & Johnson, R. A. (2000). *A new family of power transformations to improve normality or symmetry*.

(27) IBM. (2023). *Linear Discriminant Analysis (LDA)*. IBM Think. https://www.ibm.com/think/topics/linear-discriminant-analysis.

(28) GeeksforGeeks. (2024). *ML - Gradient Boosting*. GeeksforGeeks. https://www.geeksforgeeks.org/ml-gradient-boosting/.

(29) GeeksforGeeks. (2024). *ML - Extra Tree Classifier for Feature Selection*. GeeksforGeeks. https://www.geeksforgeeks.org/ml-extra-tree-classifier-for-feature-selection/.

(30) DataCamp. (2024). *AdaBoost Classifier in Python*. DataCamp. https://www.datacamp.com/tutorial/adaboost-classifier-python.

(31) XGBoost Developers. (n.d.). *XGBoost Documentation*. https://xgboost.readthedocs.io/en/stable/.

(32) CatBoost Documentation. (n.d.). CatBoost: Gradient Boosting on Decision Trees. https://catboost.ai/.

(33) IBM. (2023). *K-Nearest Neighbors*. https://www.ibm.com/it-it/topics/knn

(34) GeeksforGeeks. (2024). *Support Vector Machine (SVM) Algorithm*. GeeksforGeeks. https://www.geeksforgeeks.org/support-vector-machine-algorithm/

(35) DataCamp. (2024). *Multilayer Perceptrons in Machine Learning*. DataCamp. https://www.datacamp.com/tutorial/multilayer-perceptrons-in-machine-learning.

(36) GeeksforGeeks. (2024). *Gaussian Naïve Bayes*. GeeksforGeeks. https://www.geeksforgeeks.org/gaussian-naive-bayes/.

Chapter 4:

(37) Berry, K. (2024). *Navigating the AI Landscape: Insights from Compliance and Risk Management Leaders. Moody's Analytics.* https://www.moodys.com/web/en/us/site-assets/ma-kyc-navigating-the-ai-landscape-report.pdf