# Major Research Project on

# An Effective Approach to Managing Financial Risks Using Machine Learning Techniques


**Submitted By**

Aashish Bhandari

2K21/DMBA/03

**Under the Guidance of**

**Dr. Deepali Malhotra**

**Assistant Professor**




## DELHI SCHOOL OF MANAGEMENT

**Delhi Technological University**

**Bawana Road Delhi 110042**

# CERTIFICATE

This is to certify that Mr Aashish Bhandari bearing roll no. 2K21/DMBA/03 have completed the project titled "**An Effective Approach to Managing Financial Risks Using Machine Learning Techniques"** under the guidance of Dr. Deepali Malhotra, Assistant Professor, DTU, as a part of the Master of Business Administration (MBA) curriculum of Delhi School of Management, New Delhi. This is an original piece of work and has not been submitted elsewhere.

**Dr Archana Singh**
Associate Professor
Head of Department
(DSM, DTU)

**Dr Deepali Malhotra**
Assistant Professor
(DSM, DTU)

# DECLARATION

I solemnly declare that the project report entitled "**An Effective Approach to Managing Financial Risks Using Machine Learning Techniques**" carried out by me and submitted in partial fulfilment for the award of the degree of Master of Business Administration of the Delhi School of Management, Delhi Technological University, during the year 2022-2023. The matter embodied in this report has not been submitted to any other university or institute for the award of any other degree or diploma.

**Aashish Bhandari**

**(2K21/DMBA/01)**

# ACKNOWLEDGEMENT

It gives me great pleasure to acknowledge the assistance and constant support I received throughout my research work. I express my utmost gratitude to my faculty advisor, Dr Deepali Malhotra, who guided and mentored me throughout my research journey on the topic “An Effective Approach to Managing Financial Risks Using Machine Learning Techniques” and helped me complete the project properly. Working on the project provided me with an invaluable opportunity to explore the area of masstige marketing.

I would also like to express my heartfelt gratitude to the faculties of Delhi School of Management, Delhi Technological University for providing me the opportunity and assisting me with their expertise on this project. It has been an enriching experience for me to interact with them throughout this research and will undoubtedly contribute to my professional growth.

It has been my constant endeavour to ensure that the project is completed in the best possible manner and ensure that it is error-free.

**Aashish Bhandari**

**2K21/DMBA/03**

# Executive Summary

The research titled "An Effective Approach to Managing Financial Risks Using Machine Learning Techniques" explores the use of machine learning techniques in financial risk management, specifically in modelling financial volatility. The study uses a dataset of 30,000 rows and 25 columns containing a mix of numeric and categorical variables. The paper compares the performance of different machine-learning models and classical volatility models and finds that machine-learning models outperform classical models in terms of accuracy and predictive power.

The research concludes that machine learning techniques can be an effective tool for financial risk management and recommends that financial institutions should consider using these techniques to manage financial risks more effectively and efficiently. The research also highlights the importance of proper risk management practices, including the identification, assessment, and management of risks, as well as the need for regular review and updating of risk management policies and strategies. The research suggests that regulators should provide clear guidelines and standards for risk management practices and ensure that financial institutions comply with them.

However, the research also acknowledges the limitations and challenges of using machine learning techniques in financial risk management, such as data quality, model interpretability, and ethical considerations. The paper suggests that further research is needed to address these issues and to develop more sophisticated machine-learning models that can better capture and predict financial risks. Overall, the work gives useful insights into the potential of machine learning approaches in financial risk management and emphasises the importance of ongoing research and development in this field.

# Table of Contents

# LIST Of TABLES

# LIST OF FIGURES

# CHAPTER 1

# INTRODUCTION

## 1.1 Background

**"The essence of investment management is the management of risks, not the management of returns."**

***-Benjamin Graham***

Risk management is a dynamic process that continually evolves. This evolution is necessary because traditional risk management practices may not be effective in addressing new challenges and emerging risks. It is crucial to stay informed and adapt to changes in the risk management landscape. This requires regularly reviewing and updating the methods and techniques used to manage risk, including redefining the key components and tools of the process.

Historically, empirical research in finance has been heavily focused on statistical inference. The reasoning of statistical inference has formed the foundation of econometrics. These models are concerned with the structure of the underlying data, the generation process, and the interactions between variables. Machine learning (ML) models, however, are not assumed to define the underlying data-generating processes but are considered as a means to an end for prediction (Lommers, El Harzli, and Kim 2021). As a result, Machine Learning models are more data-centric and focused on prediction accuracy.

Data availability has long been a challenge in finance, and this often impacts the accuracy of econometric models. To overcome this issue, machine learning (ML) models have gained popularity in finance by generating synthetic data to fill in the gaps. As a result, ML models have become a popular choice for financial risk management. Before delving into a deeper discussion of these tools, it is important to first establish a foundation of key risk management concepts, including risk, different types of risks, risk management strategies, returns, and related concepts. These concepts will be referred to throughout the research.

### 1.1.1 Risk

Risk is always present, but comprehending and measuring it is more difficult than simply knowing about it. Risk is viewed as something dangerous, and it can be either expected or unexpected. Expected risk can be priced, but unanticipated risk can be disastrous.

As you might expect, there is no universally accepted definition of risk. Therefore, in financial terms, risk refers to the likelihood of a possible loss or the degree of uncertainty to which a corporation may be exposed.McNeil, Alexander, and Paul (2015) define risk differently:

**Any event or action that may adversely affect an organization's ability to achieve its objectives and execute its strategies or the quantifiable likelihood of loss or less-than-expected returns**.

These are the definitions that highlight risk's downside, claiming that cost and risk are inextricably linked, however, it should be highlighted that they do not always have a one-on-one relationship. For example, if a hazard is foreseen, the cost is small or lower (or possibly inconsequential) than if the threat is unexpected.

### 1.1.2 Return

Profit, often known as return, is the primary goal of financial investments. Return, in formal words, describes the income produced on an investment over a certain period and shows the possible upside of the risk. The notions of risk and return will be used throughout this research to handle both downside and upside risk.

As you may be aware, in financial investments, risk and reward are mutually exclusive:the larger the assumed risk, the higher the possible reward. Finding an ideal solution that balances risk and return, on the other hand, is a hard task, and this trade-off is one of the most disputed subjects in finance. However, Markowitz (1952) proposes an intuitive and appealing solution to this long-standing issue. Markowitz's definition of risk was a significant contribution to financial research, as it provided a clear and concise framework for understanding this ambiguous concept. By using standard deviation σ Ri to quantify risk, Markowitz enabled researchers to apply mathematical and statistical methods in finance (Cohen, 2018). The standard deviation can be mathematically defined as (Hull 2012):

$$\boldsymbol{\sigma=\sqrt{(E(R^2)\,)-[E(R)]^2}} \quad \textbf{(1 1)}$$

where *R* and E refer to annual return and expectation, respectively. The symbol E is frequently used in finance to represent the expected return, which is the return of interest. Since risk is defined in terms of probability, this symbol is used extensively in risk analysis. When it comes to portfolio variance, covariance is an important factor, and the formula for portfolio variance involves both expected returns and covariance.We use the symbol E frequently because expected return indicates

the return on investment. This is because we are discussing probability while defining risk. Covariance enter the perspective when it arrives to portfolio variance, and the equation is:

$$\sigma_p^2 = w_a^2\sigma_a^2 + w_b^2\sigma_b^2 + 2w_a w_b \mathrm{Cov}(r_a, r_b) \quad \text{.....} \quad \textbf{(1 2)}$$

where **w** denotes weight, $\boldsymbol{\sigma}^2$ is variance, and Cov is the covariance matrix.

The portfolio standard deviation is calculated by taking the square root of the previously acquired variance:

$$\sigma_p = \sqrt{\sigma_p^2} \quad \textbf{(1 3)}$$

In other words, portfolio anticipated return is a weighted average of individual returns and may be expressed as follows:

$$\mathbb{E}(R) = \Sigma_i^n w_i R_i = w_1 R_1 + w_2 R_2 \cdots + w_n R_n \quad \textbf{(1 4)}$$

## 1.2 Risk Management

Financial risk management is the process of dealing with the risks that arise as a result of financial markets. It involves assessing the financial risks facing an organization and developing management strategies consistent with internal priorities and policies (Horcher 2011).

According to this concept, because each organisation encounters distinct types of risks, each company's approach to risk is unique. Every organisation should appropriately analyse risk and take the required precautions. This does not necessarily imply that once a risk has been discovered, it must be reduced to the greatest extent possible by the company.

Risk management does not imply avoiding risk at all costs. Mitigating risk may need a trade-off in terms of return, which is acceptable to a point as organizations aim for both better returns and reduced risk. As a result, optimizing profit while avoiding risk must be a deliberate and well-defined procedure.Risk management comes at a cost, and while dealing with it demands specific business policies, there is a basic structure for effective risk solutions.

***Ignore***

Companies who take this strategy accept all risks and their consequences and prefer to do nothing.

***Transfer***

This method entails shifting risks to a third party by hedging or other means.

***Mitigate***

Companies use risk mitigation strategies in part because the negative consequences of risk may be judged too great to bear and/or outweigh the benefits associated with it.

***Accept***

Companies may appropriately assess risks and realize their potential advantages by implementing a risk-acceptance strategy. In other words, if taking certain risks from specialized activities may provide value for shareholders, this technique may be a good fit.

## 1.3 Main Financial Risk

Financial institutions confront a variety of hazards throughout their operations, which may be divided into several categories to help with identification and evaluation. The four main types of financial risk are market risk, credit risk, liquidity risk, and operational risk; however, this is not an exhaustive list.We will concentrate on four primary types of financial risk in this study.

### 1.3.1 Market Risk

Changes in financial market variables cause market risk. An increase in interest rates, for example, might be detrimental to a corporation with a short position. Exchange rate fluctuations are another source of market risk for a foreign firm whose goods are priced in US dollars.

Commodity price swings can potentially jeopardize a company's financial viability since changes in market participants, transportation costs, and other elements can all have an immediate influence on commodity pricing.

### 1.3.2 Credit Risk

Credit risk is one of the most common worries. It arises when a counterparty fails to fulfil a commitment. Credit risk is realised, for example, when a borrower fails to make a payment.Deterioration of credit quality is also a source of risk through the reduced market value of securities that an organization might own (Horcher 2011).

### 1.3.3 Liquidity Risk

Liquidity risk was commonly disregarded before to the 2007-2008 financial crisis, which had a significant influence on financial markets. Since then, there has been an increase in research on

liquidity risk, because liquidity refers to the speed and ease with which investors may execute transactions. This is commonly known as trading liquidity risk.

Another type of liquidity risk is funding liquidity risk, which refers to a company's ability to get cash or credit to fund its operations. If a company is unable to convert its assets into cash in a timely manner, it faces liquidity risk, which can have major ramifications for its financial management and reputation.

### 1.3.4 Operational Risk

Because of the complicated and internal nature of the risk, managing operational risk is a difficult and demanding process that necessitates a large amount of a company's resources. Some questions that emerge in this situation are as follows:

- How can financial companies effectively manage risk?
- Do they allocate sufficient resources for this task?
- Is the importance of risk to a company's sustainability appropriately evaluated?

The risks that come from external events or inherent operations of a firm or industry that might jeopardize its day-to-day operations, profitability, or sustainability are referred to as operational risks. Operational risk includes fraudulent conduct, failure to comply with regulations or internal processes, and losses caused by a lack of training.

What happens if a company is unprepared for any of these risks? Although it is unlikely, historical events have shown that the company may default and run into serious financial difficulties.

## 1.4 Big Financial Collapse

Long-Term Capital Management was a hedge fund founded in 1994 by a group of former Salomon Brothers employees, including Nobel Prize-winning economists. The fund's strategy was to use complex mathematical models to identify market inefficiencies and make profitable trades. The fund was highly leveraged, meaning that it borrowed large amounts of money to make even larger trades, and it focused on trading in government bonds, currencies, and other financial instruments.

However, in 1998, Russia defaulted on its debt, causing a sharp decline in global markets and a loss of confidence among investors. This had a severe impact on LTCM, as many of its trades were based on the assumption that markets would remain stable. As the markets became more volatile, LTCM's losses mounted, and it was unable to meet margin calls from its creditors.

The failure of LTCM had far-reaching consequences, as it was a major player in the financial markets and had extensive connections with other financial institutions. Its collapse threatened to create a domino effect, as other firms that had exposure to LTCM could also suffer losses and face liquidity problems. In response, the Federal Reserve Bank of New York organized a bailout of LTCM, in which a consortium of banks provided a $3.6 billion loan to the fund to keep it afloat.

The LTCM collapse highlighted the dangers of excessive leverage and the need for better risk management practices in the financial industry. It also demonstrated how interconnected financial institutions can be, and how a failure in one part of the system can have cascading effects throughout the entire system. As a result, risk management has become a critical function in financial institutions, and regulators have implemented stricter rules and oversight to prevent similar failures in the future.

Another corporation that no longer exists as a result of poor financial risk management is Metallgesellschaft (MG). MG primarily operates in the gas and oil markets. Because of its considerable exposure, MG needs financing following the sharp decline in gas and oil prices. The closing of the short position resulted in losses of around $1.5 billion.

Amaranth Advisors (AA) is yet another hedge fund that went bankrupt after overinvesting in a particular market and underestimating the risks connected with these transactions. By 2006, AA had more over $9 billion in assets under management, but nearly half of them had been lost owing to a decrease in natural gas futures and options. The default of AA is attributed to low natural gas prices and misleading risk models (Chincarini 2008).

Kingfisher Airlines was an Indian airline that was founded in 2005 by the businessman Vijay Mallya. The airline initially experienced significant growth, and by 2011 it was the second-largest airline in India in terms of passenger share. However, in 2012, the airline began to face financial difficulties, and it eventually became bankrupt.

The key reasons for Kingfisher Airlines' bankruptcy were its high debt burden, high operating costs, and weak financial performance. The airline had taken on a large amount of debt to fund its expansion plans, but it was not able to generate sufficient revenues to service this debt. At the same time, the airline's operating costs were relatively high compared to its competitors, due to factors such as high fuel costs and a large fleet of aircraft.

Kingfisher Airlines' financial problems were compounded by a range of regulatory and legal issues, including concerns about safety and security, labour disputes, and regulatory compliance.

These issues led to disruptions in the airline's operations, negative publicity, and a decline in customer confidence.

In 2012, Kingfisher Airlines was unable to pay its employees or suppliers, and it was forced to ground its fleet. The company's operating license was eventually suspended by the Indian aviation regulator, and the airline was declared bankrupt in 2013.

The bankruptcy of Kingfisher Airlines had a significant impact on the Indian aviation industry and the wider Indian economy. It left thousands of employees and suppliers out of pocket, and it highlighted the risks and challenges of operating in a highly competitive and regulated industry.

Stulz's paper, "Risk Management Failures: What Are They and When Do They Happen?" (2008) outlines the major risk management flaws that might lead to default:

- Miscalculation of known dangers
- Failure to account for hazards
- Failure to communicate dangers to senior management
- Risk monitoring failure
- Risk management failure
- Inadequate utilization of relevant risk metrics

As a result, the global financial crisis was not the only event that spurred regulators and institutions to reconsider their approach to financial risk management. Rather, it was the drop that filled the glass, and both regulators and institutions learnt from their mistakes and improved their systems in the aftermath of the crisis. This series of events finally resulted in greater financial risk management.

## 1.5 Problem Statement

The problem statement of this research is to explore the potential of machine learning in the financial risk management domain and to assess its ability to enhance the accuracy and efficiency of traditional financial risk management methods. The study aims to evaluate various machine learning algorithms and compare their results with traditional methods to determine the improvement in accuracy and efficiency. The research seeks to provide a valuable contribution to the field of financial risk management by demonstrating the potential of machine learning to transform traditional methods into more efficient and effective ones.

## 1.6 Objective of The Research

The objective of the study " An effective approach to managing financial risks using machine learning techniques " is likely to explore the use of machine learning algorithms to improve the process of financial risk management. The use of Python, a popular programming language for data science and machine learning, suggests that the study aims to implement these algorithms in a practical and accessible way.

The goal is probably to demonstrate the potential benefits of using machine learning in financial risk management, such as increased accuracy, efficiency, and automation of the risk assessment process. By using machine learning techniques, the study aims to address some of the challenges faced by traditional financial risk management methods and provide a more comprehensive and effective approach to managing financial risk.

The objective of the study is to show how machine learning can be applied to financial risk management in a Python-based approach and to highlight the potential benefits of using these techniques in practice.

## 1.7 Scope of The Research

The purpose of this study is to look at how machine learning techniques may be used to manage several forms of financial risks, such as market risk, credit risk, liquidity risk, and operational risk. The following aspects will be investigated:

- The most recent cutting-edge machine learning approaches and applications for financial risk management in many domains and sectors.
- The benefits and drawbacks of utilizing machine learning for financial risk management, include data quality, interpretability, regulation, and ethical concerns
- Different machine learning models and algorithms for financial risk analysis and prediction, such as supervised, unsupervised, and reinforcement learning, are compared and evaluated.

- The creation and deployment of unique machine learning solutions for specific financial risk challenges or situations such as volatility modelling, value at risk calculation, credit scoring, fraud detection, and liquidity risk measurement.
- Deep learning, natural language processing, computer vision, and explainable AI are examples of future machine learning trends and directions for financial risk management.

# CHAPTER 2

# LITERATURE REVIEW

## 2.1 Volatility Prediction

Volatility prediction is an important task in finance, as it measures the degree of uncertainty and risk in the market. Volatility can be estimated using historical data (realized volatility) or implied from option prices (implied volatility). Machine learning (ML) is a data-driven approach that can leverage complex nonlinear relationships and high-dimensional features to forecast volatility. Here is a literature review of some recent studies on volatility prediction using ML:

- Zhang et al. (2023) applied neural networks to forecast intraday realized volatility by exploiting commonality in intraday volatility across stocks and incorporating a proxy for market volatility. They found that neural networks outperformed linear regressions and tree-based models and that their method could generalize to new stocks that were not included in the training set. They also presented a new method for forecasting one-day-ahead realised volatility utilising prior intraday realised volatility as predictors, as well as fascinating time-of-day factors that benefited the forecasting process.
- Karasan (2022) presented a comprehensive overview of ML-based volatility prediction, covering various methods such as linear models, support vector machines, random forests, gradient boosting machines, neural networks, and deep learning. He discussed the advantages and disadvantages of each method and provided empirical applications to different financial markets and assets. He also compared the performance of ML models with traditional econometric models such as ARCH and GARCH and showed that ML models could capture nonlinearities and heteroskedasticity better than econometric models.
- Katsiampa et al. (2021) focused on forecasting implied volatility directional changes using ML techniques such as logistic regression, decision trees, random forests, support vector machines, and neural networks. They used data from S&P 500 index options and evaluated the models based on statistical and economic criteria. They concluded that ML techniques could be more effective than mainstream econometric models and model selection techniques in forecasting implied volatility directional changes.
- Liu et al. (2021) proposed a deep learning model with an attention mechanism to forecast the volatility of the stock index. They used data from Shanghai Composite Index and

Shenzhen Component Index and compared their model with several benchmark models such as ARIMA, GARCH, LSTM, and GRU. They found that their model could capture the long-term dependency and dynamic features of volatility better than the benchmark models, and achieved higher accuracy and lower error.

Schrödinger (2020) developed an ML model that predicted the volatility of organic molecules up to C20, which are used as precursors for chemical vapour deposition. They used a graph convolutional neural network to encode the molecular structure and a fully connected neural network to predict volatility. They showed that their model could accurately predict the volatility of precursor molecules, which are typically organometallic complexes, without requiring experimental data or quantum mechanical calculations.

## 2.2 Market Risk

Market risk analysis is a critical component of risk management in financial institutions. Value at Risk (VaR) and Monte Carlo simulation are commonly used methods for quantifying and managing market risk. With the advancement of machine learning techniques, there has been growing interest in leveraging these techniques to enhance the accuracy and efficiency of market risk analysis. This literature review aims to explore the existing research on the topic of market risk analysis using VaR and Monte Carlo simulation with machine learning techniques, as well as the challenges and opportunities associated with this approach.

VaR is a widely used measure for estimating the maximum loss that a financial institution may incur within a given confidence level over a specified time horizon (Hull, 2018). Traditional methods for estimating VaR, such as historical simulation and parametric methods, have limitations in capturing complex market dynamics and tail risks. Machine learning techniques, including neural networks, decision trees, and support vector machines, have been proposed as alternative approaches to improve VaR estimation accuracy (Huang et al., 2018). For example, Chen and Leung (2017) applied recurrent neural networks (RNN) to forecast stock prices and estimate VaR and found that RNN outperformed traditional methods in capturing nonlinearities and improving the accuracy of VaR estimates.

**Prado (2020) puts it briefly:** "Considering the complexity of modern financial systems, it is unlikely that a researcher will be able to uncover the ingredients of a theory by visual inspection of the data or by running a few regressions." To answer the second question, consider the working logic of machine learning models. In contrast to traditional statistical approaches, ML models attempt to uncover the relationships between variables, identify crucial factors, and allow us to determine the influence of the variables on the dependent variable without the necessity for a well-established theory. Indeed, the beauty of ML models is that they allow us to uncover hypotheses rather than requiring them.

"Many methods from statistics and machine learning (ML) may, in principle, be used for both prediction and inference. However, statistical methods have a long-standing focus on inference, which is achieved through the creation and fitting of a project-specific probability model.
By contrast, ML concentrates on prediction by using general-purpose learning algorithms to find patterns in often rich and unwieldy data." Bzdok (2018, p. 232). We'll begin our consideration of market risk models in the next section. First, we'll look at how the VaR and ES models may be used. We will learn how to enhance these models after analyzing their typical applications. an ML-based strategy.

# CHAPTER 3

# METHODOLOGY

## 3.1 Research Design

The research will adopt a quantitative study approach since it will evaluate financial datasets using statistical techniques and machine learning algorithms. The study will be done in a cross-sectional manner, with data from several sources collected at a particular moment in time.

## 3.2 Source of Data Collection

**a) Nifty Dataset**: The Nifty dataset will be imported via the Yahoo Finance API, which offers historical stock price data for the National Stock Exchange of India's Nifty 50 index. The dataset will comprise daily or hourly data for a specific period, including opening, closing, high, low, and volume prices.

**b) Tawani Credit Card Dataset:** The credit card dataset will be downloaded through the UCL Machine Learning Repository website, which has a large collection of freely available datasets for machine learning research. Anonymized credit card transaction data, including transaction amount, transaction date and time, merchant category code, and customer information, will be included in the dataset.

## 3.3 Data Preprocessing

- Conduct data quality checks on both datasets to detect and address any missing values, discrepancies, or mistakes.
- Remove extraneous or superfluous data from the datasets and verify that the data is in a consistent format for analysis.
- If required, normalize the data to ensure that it is on a similar scale and can be reliably compared.
- As per ethical principles and legislation, handle any data privacy and security problems, such as anonymizing or encrypting sensitive data.

## 3.4 Feature Engineering

- Analyze the datasets to find useful characteristics for volatility prediction and credit default analysis.
- If necessary, create new features or alter existing features to extract relevant information from the data.
- To obtain insights into the data and uncover potential patterns or trends, use exploratory data analysis (EDA).

## 3.4 Model Development

- Choose relevant machine learning methods, such as time series analysis approaches for volatility prediction and credit default analysis.
- To assess the performance of the models, divide the datasets into training and testing sets.
- Using the training datasets, train the machine learning models and fine-tune them as needed.
- Analyze the models' performance using relevant assessment measures such as F1-score, precision, and accuracy.
- Interpret the model findings and assess their efficacy in minimizing financial risks.

## 3.5 Model Interpretation & Results

- Interpret and assess the model outputs within the context of the research topic and the datasets employed.
- Conclude the approach's success in mitigating financial risks using machine learning techniques.
- Discuss the research results' implications and limits.
- Make suggestions for more studies and potential uses of the technique in real-world financial risk management settings.

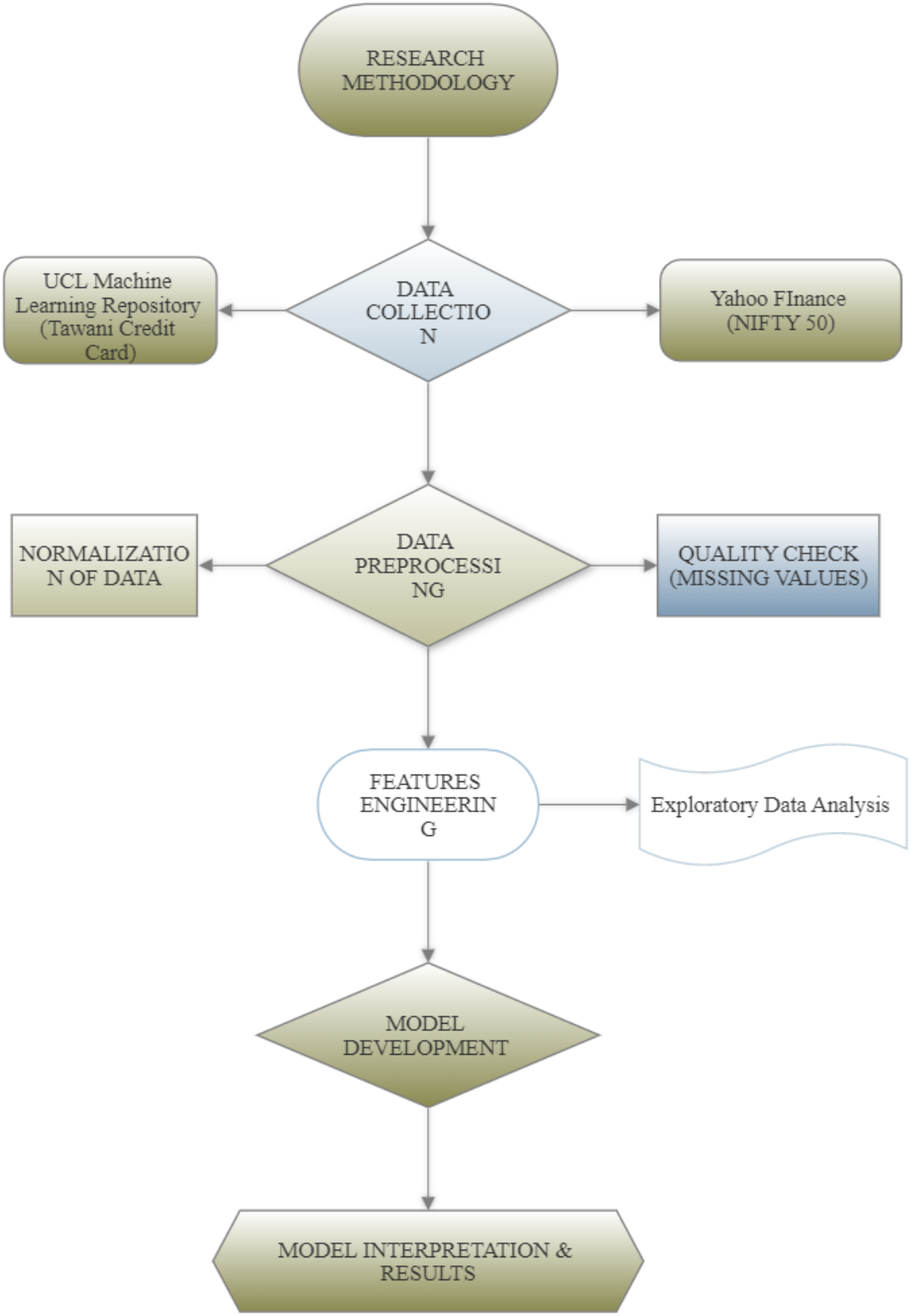


***Figure 3. 1 Research Methodology Flow Chart***

## 3.4 Analysis Tools & Procedure

1. **Statistical Methods:** - ARCH (Autoregressive Conditional Heteroscedasticity) and GARCH (Generalized Autoregressive Conditional Heteroscedasticity) models are employed in the study to forecast volatility. In finance, these models are often employed to represent time-varying volatility in financial time series data. The ARCH model incorporates the autoregressive connection into the conditional variance, whereas the GARCH model includes a component for the delayed conditional variance. These methods are used to estimate volatility and compare it to the outcomes of machine learning techniques.
2. **Neural Networks as Machine Learning Models**: - For volatility prediction, the study leverages neural networks, a prominent form of machine learning model. Neural networks can learn complicated patterns from data and recognise non-linear correlations. The study describes how to apply gradient descent and backpropagation methods in machine learning to identify the optimum parameters that minimise and lower the cost function in neural networks, respectively.
3. **Value at Risk (VaR) Technique** - The study focuses on market risk analysis using the VaR approach, a statistical method that assesses the greatest probable loss of an investment over a particular period with a specific level of confidence. To compute VaR, the study used the variance-covariance methodology, often known as the parametric method. This method implies that returns have a normal distribution, allowing the VaR methodology to be used. The study recognises the importance of exercising caution while working with portfolios, since measuring the correlation structure and portfolio variance necessitates careful thought.
4. **Monte Carlo Simulation with Machine Learning Techniques** - Machine learning techniques are employed in the study to improve the accuracy of the Monte Carlo simulation, a method used to estimate the likelihood of multiple outcomes in processes containing random variables that are difficult to predict. The project attempts to increase the accuracy of market risk assessments by integrating classic statistical approaches with machine learning techniques.
5. **Credit Default Analysis with Decision Tree Classification Algorithms** - For credit default prediction, the study used decision tree Techniques for categorization including

logistic regression, decision trees, and random forest. These algorithms are often used in credit risk analysis to determine if a borrower is in default or not. To minimise dimensionality and increase model performance, the research employs feature engineering and selection, as well as standardization or normalization to scale the data. The information is separated into train and test sets, and the algorithms' hyperparameters are tuned via cross-validation or grid search. After training on the train set, the algorithms accuracy, precision, recall, and f1-score were all examined on the test set, ROC-AUC, and confusion matrix. The outcomes of several algorithms are compared, and the best one is chosen depending on the results.

# CHAPTER 4

# VOLATILITY PREDICTION

## 4.1 Introduction

The most important element of the distribution of conditional returns is undoubtedly its second-moment structure, which is the distribution's dominant time-varying characteristic experimentally. This feature has given rise to a massive body of research on the modelling and forecasting of return volatility. Andersen et al. (2003)."Some concepts are simple to grasp but difficult to define. This also applies to volatility." Because the way Markowitz models volatility is quite straightforward and intuitive, this might be a remark made by someone who lived before Markowitz. Markowitz offered his well-known after developing portfolio theory, in which he defined volatility as standard deviation, and finance became increasingly intertwined with mathematics.

Volatility is the foundation of finance since it not only informs investors but also serves as an input to countless financial models. What exactly does volatility mean? The reaction emphasises the significance of uncertainty. It is the most crucial component of the financial model. Financial market integration has resulted in extended instability in certain markets., emphasizing the importance of volatility, or the degree to which the values of financial assets change. One of the most often utilised risk proxies is volatility. These variables are used in many sectors, including asset pricing and risk management. Its strong presence and delay make modelling almost mandatory. After the implementation of the Basel Accord in 1996, volatility has played an important role in risk management (Karasan and Gaygisiz 2020).

Following Black's seminal work in 1976, a vast and expanding body of literature on volatility estimation arose, including Andersen and Bollerslev (1997), Raju and Ghosh (2004), Dokuchaev (2014), and De Stefani et al. (2017). We're talking about a lengthy history of volatility prediction based on ARCH and GARCH-type models, which include defects that might lead to failures, such as volatility clustering and information asymmetry. Even though these challenges are addressed by many methodologies, recent developments in financial markets, as well as breakthroughs in machine learning, have prompted scholars to reexamine volatility assessment.

## 4.2 Analysis

Modelling volatility is analogous to modelling uncertainty to better comprehend and approach uncertainty, allowing us to get fair approximations of the actual world. To evaluate how well-suggested models account for reality. In this instance, the return volatility, also known as realized volatility, must be computed. The sum of squared returns is realised variance, which is the square root of realised volatility. The volatility prediction approach's performance is calculated using realised volatility. The return volatility formula is shown below.:

$$\hat{\sigma} = \sqrt{\frac{1}{n-1}\sum_{n=1}^{N}(r_n - \mu)^2}$$

where r and μ the return and mean of return, respectively, and n is the number of observations.

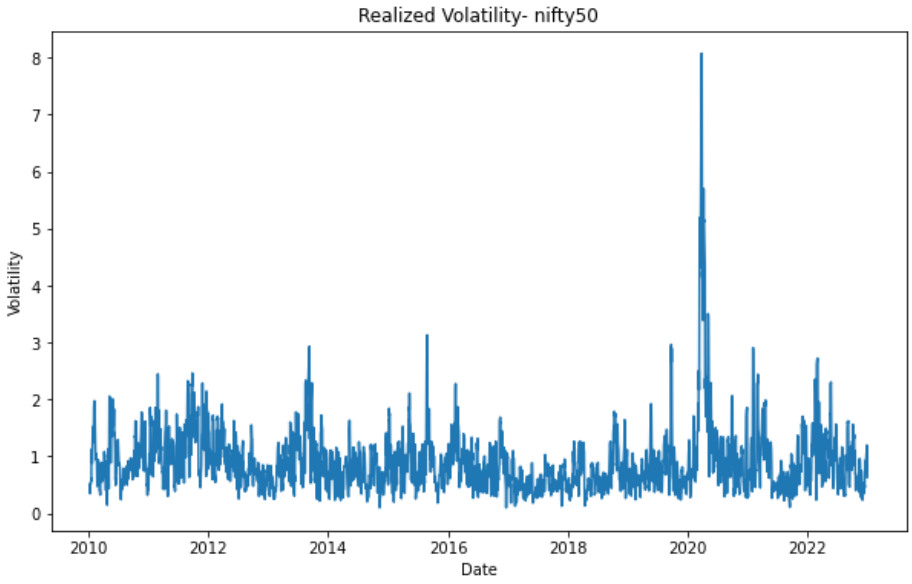


***Figure 4. 1 Realized Volatility***

Figure 4.1 depicts the realised volatility of the Nifty50 from 2010 to 2022. The most notable finding is the clustering of spikes around the COVID-19 pandemic. The method used to measure Volatility has an undeniable influence on the study's dependability and accuracy. As a result, this paper investigates both traditional and ML-based volatility prediction algorithms to demonstrate that the ML-based models outperform the traditional models. We begin by replicating the classic volatility models to compare the brand-new ML-based models. Well-known classical volatility models include, but are not limited to, the following:

### 4.2.1 ARCH Model

Eagle (1982) proposed the ARCH model, which was one of the first attempts to model volatility. The ARCH model is a univariate model that is based on asset returns from the past. The ARCH(p) model is written as follows:

$$\sigma_t^2 = \omega + \sum_{k=1}^{p} \alpha_k (r_{t-k})^2 \qquad r_t = \sigma_t \epsilon_t$$

Where the mean model is:

where $\varepsilon_t$ It is expected that the data is distributed on a regular basis. Specific assumptions must be satisfied in order to have a strictly positive variance in this parametric model. The following requirements should be met in this regard:

- $\omega > 0$
- $\alpha_k \geq 0$

All of these equations show that ARCH is a univariate, nonlinear model with volatility that is evaluated by squaring previous returns. ARCH has the trait of time-varying conditional variance1, which is one of its most distinguishing aspects that ARCH can replicate the volatility clustering phenomenon—that is, large changes tend to be followed by large changes of either sign and tiny changes tend to be followed by small changes, as described by Mandelbrot— (1963). As a result, if an important announcement is given to the market, it may cause significant volatility.

We used our optimization approach and ARCH equation to model volatility using ARCH.

For the ARCH model, we get the following summary:

```
                     Zero Mean - ARCH Model Results
==============================================================================
Dep. Variable:              Adj Close   R-squared:                       0.000
Mean Model:                 Zero Mean   Adj. R-squared:                  0.000
Vol Model:                       ARCH   Log-Likelihood:               -4664.43
Distribution:                  Normal   AIC:                           9332.86
Method:            Maximum Likelihood   BIC:                           9344.99
                    No. Observations:                 3188
Date:                Thu, Mar 02, 2023, Df Residuals:                     3188
Time:                        20:12:54   Df Model:                           0
                          Volatility Model
   ============================================================================
            coef    std err          t      P>|t|  95.0% Conf. Int.
   ----------------------------------------------------------------------------
     omega          0.8969 5.128e-02     17.492 1.658e-68 [ 0.796, 0.997]
    alpha [1]       0.2313 4.995e-02      4.632 3.625e-06 [ 0.133, 0.329]
   ============================================================================
```

ARCH Model Summary

***Residual and Conditional volatility:***

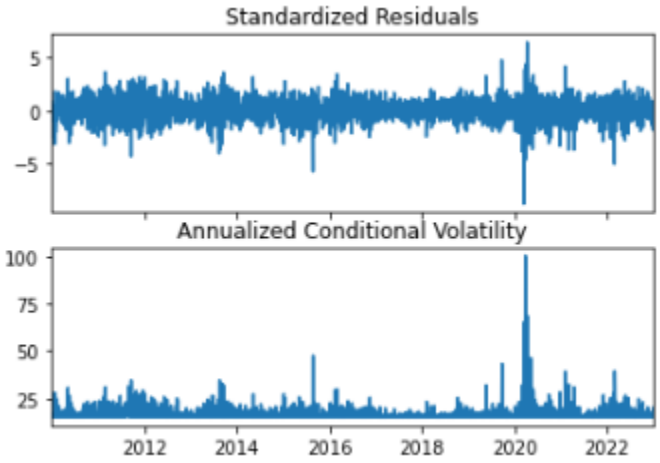


***Figure 4. 2 Residual and Conditional volatility***

The figure depicts the outcome of volatility prediction using our initial model.

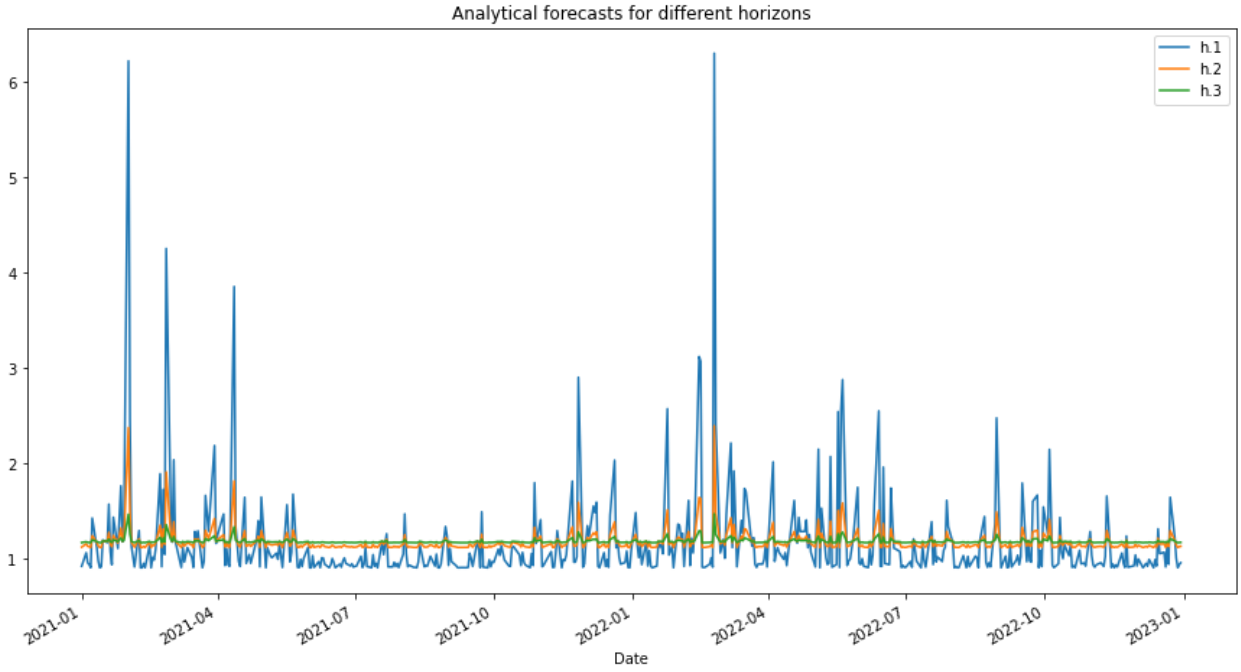


***Figure 4. 3 Nifty 50 Forecasted Volatility using ARCH Model***

### 4.2.2 GARCH Models

The Generalized Autoregressive Conditional Heteroskedasticity (GARCH) model, an extension of the ARCH model, is demonstrated.The GARCH model can be thought of as an ARMA model applied to the variance of a time series—the AR component is already described in the ARCH model.

The equation of the GARCH model can be presented as:

$$r_t = \mu + \epsilon_t$$

$$\epsilon_t = \sigma_t z_t$$

$$\sigma_t^2 = \omega + \sum_{i=1}^{q} \alpha_i \epsilon_{t-1}^2 + \sum_{i=1}^{p} \beta_i \sigma_{t-1}^2$$

While the interpretation is quite similar to the ARCH model described in the previous recipe, the difference may be found in the final equation, where we can see an additional component. Parameters are constrained to meet the following: **$\omega>0$, $\alpha>=0$, $\beta_i>=0$.**

The GARCH model's two hyperparameters are as follows:

**2** *p:* The total amount of lag variations

**3** *q:* The number of residual lag errors in a mean procedure.

The squared residuals from a model used to forecast the mean of the original time series can be used to infer the lag orders for ARCH/GARCH models. The squares of the residuals correspond to their variance because they are centred around zero. We can examine the ACF/PACF plots of the squared residuals to detect patterns in the autocorrelation of the series' variance (just like we did to determine the orders of an ARMA/ARIMA model). GARCH adds the moving average component to the model.

```
                        Zero Mean - GARCH Model Results
==============================================================================
Dep. Variable:              Adj Close   R-squared:                       0.000
Mean Model:                 Zero Mean   Adj. R-squared:                  0.000
Vol Model:                      GARCH   Log-Likelihood:               -4433.86
Distribution:                  Normal   AIC:                           8873.71
Method:            Maximum Likelihood   BIC:                           8891.91
                    No. Observations:                 3188
Date:                Thu, Mar 02, 2023, Df Residuals:                     3188
Time:                        20:13:34   Df Model:                           0
                            Volatility Model
============================================================================
            coef    std err          t      P>|t|      95.0% Conf. Int.
----------------------------------------------------------------------------
omega      0.0234  5.820e-03      4.028  5.628e-05 [1.203e-02,3.485e-02]
alpha[1]   0.0830  1.217e-02      6.819  9.199e-12   [5.911e-02,  0.107]
beta[1]    0.8970  1.249e-02     71.803      0.000     [  0.873,  0.922]
============================================================================
```

GARCH Model Summary

According to Market Risk Analysis, the normal range of values for the metrics in a stable market is between **0.05 < α < 0. 01** and **0.85 < β < 0.98**. However, we should keep in mind that, while these ranges are unlikely to be technically applicable, they do provide us with a notion of what kinds of values to expect.

We can see that the log-likelihood increased when compared to the ARCH model, indicating that the GARCH model fits the data better. Yet, we must exercise caution when reaching such judgements. The log-likelihood will almost certainly increase when additional predictors are added (as we have done with GARCH). If the number of predictors changes, a likelihood-ratio test should be performed to compare the goodness-of-fit criteria of two nested regression models.
We can see the effect of incorporating the extra component (lagged conditional volatility) in the model specification in the charts below:

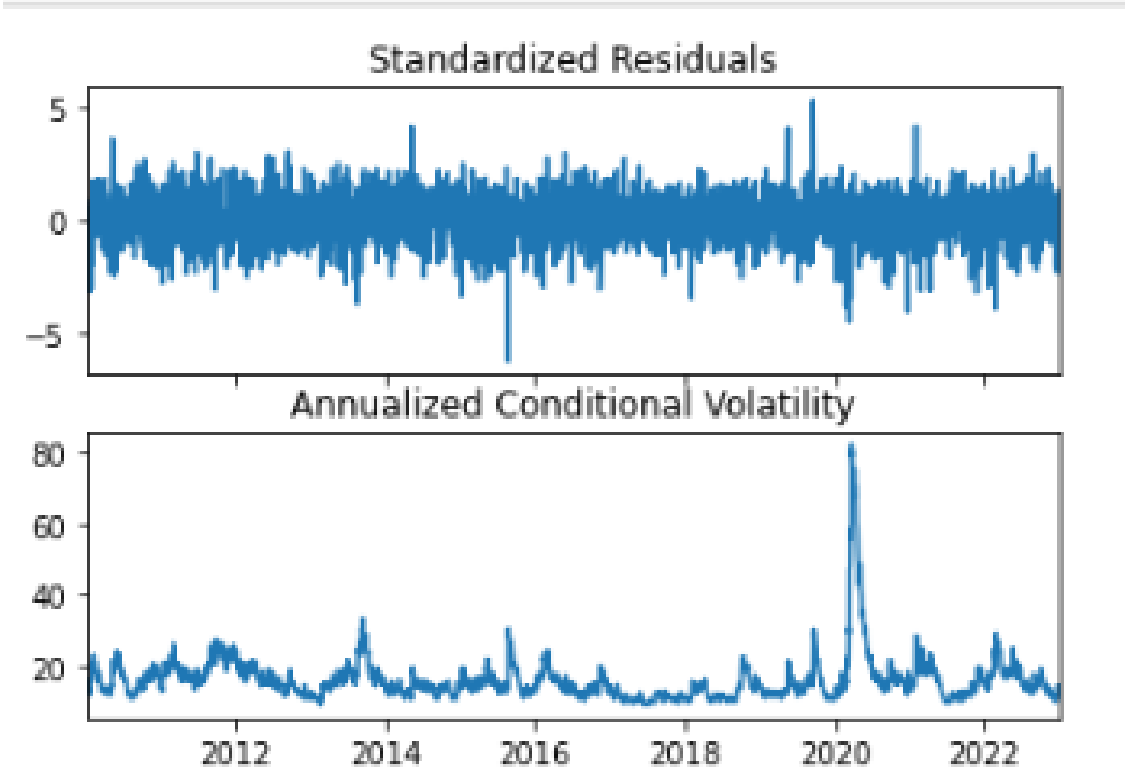


***Figure 4. 4 Standardized residuals and the annualized conditional volatility of the fitted GARCH model***

When ARCH is used, the conditional volatility series displays multiple spikes before returning to a low level. Because the GARCH model incorporates delayed conditional volatility, it takes longer to revert to the level recorded before the spike.

### 4.2.3 Forecasting Volatility using the GARCH model

we have seen how to fit ARCH/GARCH models to a return series. However, the most interesting/relevant case of using ARCH class models would be to forecast the future values of the volatility.

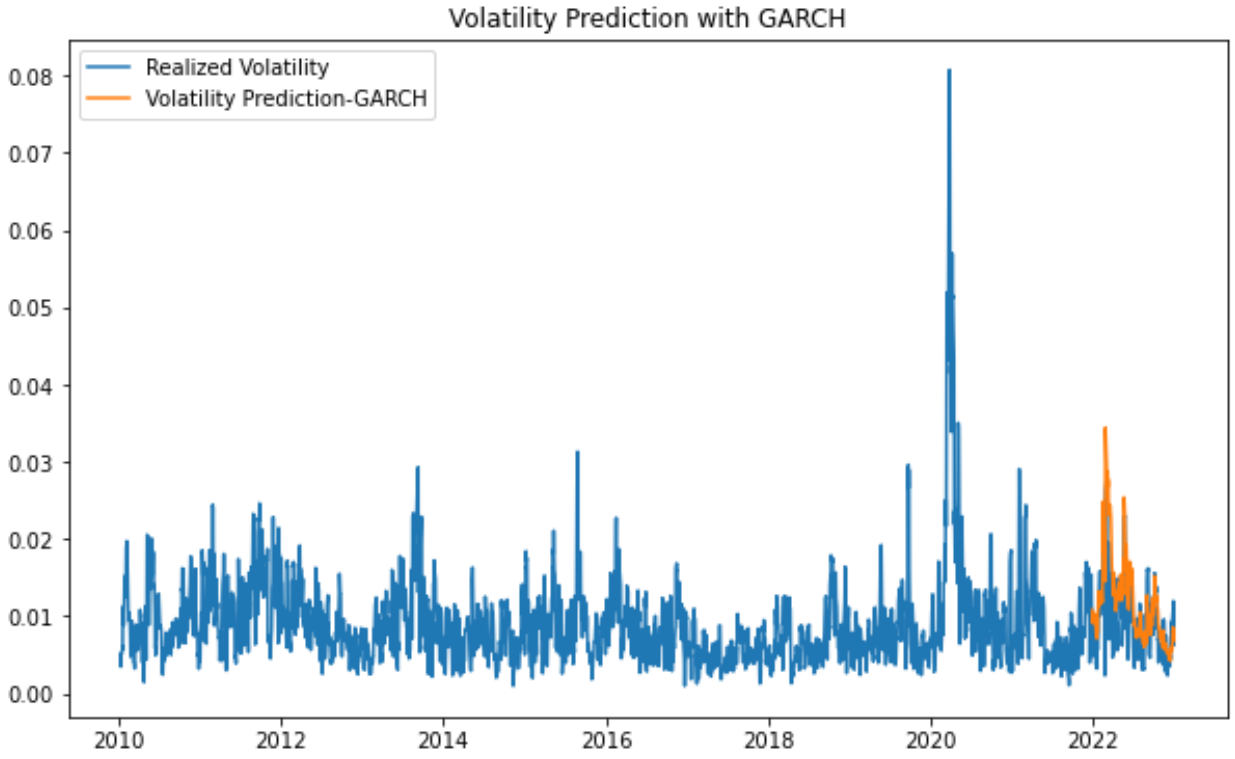


***Figure 4. 5 The GARCH model is used to forecast volatility.***

The GARCH model matches the volatility of returns well, partially because of its volatility clustering and partly because GARCH does not assume that the returns are independent, it can account for the leptokurtic aspect of returns.GARCH, despite its useful properties and intuitiveness, cannot model the asymmetric response of shocks (Karasan and Gaygisiz 2020). Glosten, Jagannathan, and Runkle proposed GJR-GARCH to address this issue (1993). We have already examined classical volatility models, but now We will look at how machine learning and the Bayesian technique may be used to model volatility. The initial models to be examined in the framework of ML will be support vector machines and neural networks. Let's get this party started.

### 4.2.4 Neural Network

Deep learning is built around neural networks. To make a decision, data is processed in many phases of a NN. As input, each neuron receives a dot product result and utilises it in an activation function to make a decision:

$$z = w_1 x_1 + w_2 x_2 + b$$

where *b* is bias, *w* is weight, and *x* is input data.

In the hidden and output layers, input data is mathematically manipulated in various ways. In general, There are three types of layers in NN:

- Input Layer
- Hidden Layer
- Output Layer

Figures can serve to depict the connections between layers. The raw data is stored in the input layer. We learn coefficients as we proceed from the input layer to the hidden layer. Depending on the network structure, there may be one or more hidden layers.. The more hidden levels there are in a network, the more difficult it becomes. Hidden layers, which are positioned between the input and output layers, conduct nonlinear transformations using activation functions.

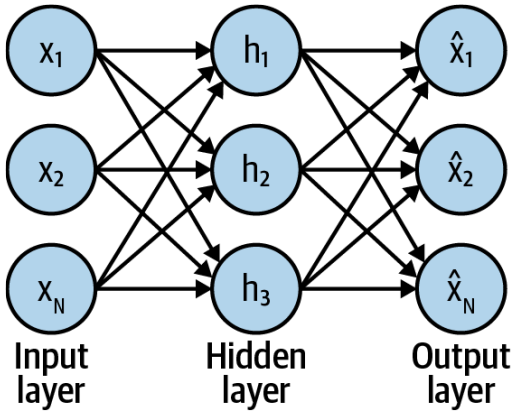


***Figure 4. 6 NN Structure***

Finally, the output layer is where output is generated and judgments are made. Gradient descent is used in ML to find the optimal parameters that minimise the cost function, however, because of the chain-like structure of NN, employing solely gradient descent is not practical. To minimise the

cost function, an unique technique called as backpropagation is developed. Backpropagation works by computing the difference between observed and real output and passing the resulting error to the buried layer. So, we take a step back, and the main equation is as follows:

$$\delta^l = \frac{\delta J}{\delta z_j^l}$$

where z is the linear transformation and indicates the error. There is much more to say here, but we'll stop here to stay on course. For those who want to dig more into the math behind NNs, please refer to Wilmott (2013) and Alpaydin (2020).

Even though there are several ways to run NNs in Python, we will use the **MLPRegressor** module from sci-kit-learn to predict volatility. Given the NN structure we've established, the result is as follows:

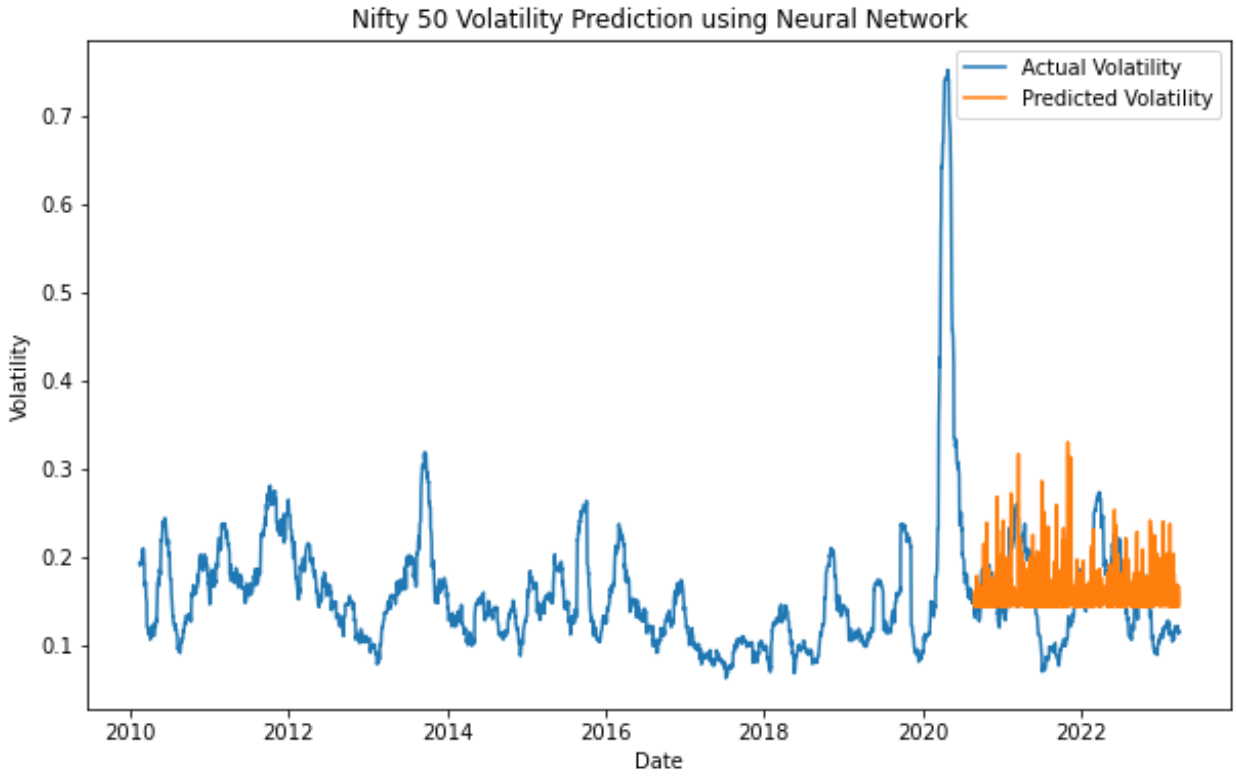


***Figure 4. 7 Volatility Prediction Using Neural Network***

We observed that the combination of an epoch number and batch size set to 100 resulted in the lowest RMSE score. This study implies that increasing the model's complexity does not necessarily result in increased prediction performance. To avoid overfitting, which can degrade model

performance, it is critical to maintain a balance between model complexity and predictive performance.

The figure above displays the outcome of the volatility prediction obtained from the neural Network presented earlier. The results suggest that deep learning can serve as a powerful tool for modelling volatility as well.

***Table 4. 1 Error Table***

| **Error** | **Value** |
|---|---|
| Mean Square Error | 0.011940 |
| Mean Absolute Error | 0.086688 |
| Root Mean Square Error | 0.109273 |

Our model looks to have rather low error metrics based on the numbers we obtained for the Mean Squared Error (MSE), Mean Absolute Error (MAE), and Root Mean Squared Error (RMSE).

The MSE of 0.011940 indicates that the squared difference between predicted and actual values is fairly minimal on average. A lower MSE means that the anticipated values are more accurate.

The MAE of 0.08668895 suggests that the absolute difference between anticipated and actual values is likewise reasonably minimal on average. Because MAE provides equal weight to all mistakes, a lower MAE indicates that projected values are closer to actual values.

The RMSE of 0.109273, which is the square root of MSE, reflects the average difference between predicted and actual values in the predicted variable's original units. The smaller the RMSE, the closer the predicted values are to the actual values.

Lower MSE, MAE, and RMSE values, in general, suggest greater model performance and accuracy. However, the interpretation of these values is dependent on the specific problem at hand as well as the context of the data. It's important to consider these values in comparison with other models, benchmarks, and domain knowledge to fully assess the condition of your model and its suitability for your specific use case.

# CHAPTER 5

# MARKET RISK ANALYSIS

## 5.1 Introduction

Market risk is defined as the risk posed by changes in financial indices such as the exchange rate, interest rate, inflation, and so on. Market risk is the danger of incurring losses in on- and off-balance-sheet positions as a result of price changes in the market. (BIS 2020). Let us now examine how these variables influence market risk.

Assume that an increase in inflation rates threatens the existing profitability of financial institutions, because inflation puts pressure on interest rates. This, in turn, has an impact on the cost of financing for borrowers. various occurrences can be increased, but we need also to consider how various financial risk sources combine. That is, when one source of financial risk changes, other sources of risk cannot remain constant. Thus, financial indicators are connected to some extent, implying that the interplay of various risk sources should be considered.

Market risk is a branch of risk analysis that looks at portfolio losses caused by negative market movements. The market risk estimates described in this chapter forecast extreme cases of how much a portfolio can lose in a short period, such as 10 days. These indicators include value-at-risk (VaR) and anticipated shortfalls (ES). Typically, financial institutions use these to assess the degree of short-term risk in a derivatives portfolio and estimate the number of assets necessary to cover likely losses.

As you may expect, there are several tools for managing market risk. The most well-known and commonly used techniques are valued at risk (VaR) and projected shortfall. (ES). The ultimate goal of this chapter is to supplement existing techniques with current advances in machine learning. It would be tempting to ask the following questions at this point:

- Do traditional models fail in finance?
- What makes the ML-based model different?

I'll begin by answering the first question. The complexity of the financial system is the first and most important problem that traditional models cannot manage. Long-standing conventional models are being replaced by ML-based models due to either certain strong assumptions or simply their inability to capture the complexity supplied by data.

## 5.2 Analysis

### 5.2.1 Value at Risk (VaR)

Value at Risk (VaR) is a prominent risk management metric that predicts the largest likely loss that an investment portfolio or individual position may tolerate with a given level of certainty over a specific period. VaR quantifies the risk of loss in terms of a given probability level, which is often stated as a percentage of the portfolio's value. VaR is a must-have tool for portfolio managers and risk managers who wish to understand and manage the risk of their assets. It assists investors in quantifying the risk associated with their portfolio and setting limitations on their exposure to various asset types such as stocks, bonds, and derivatives. The adoption of VaR dates back to the 1990s, and despite numerous extensions to it and newly proposed models, it is still in use. What makes it so appealing? The answer comes from Kevin Dowd (2002, p. 10):

There are two significant properties of the VaR figure. The first is that it gives a uniform, standard measure of risk that can be applied to various jobs and risk variables. It allows us to compare and be consistent with a measure of the risk associated with an equity position, for example, when assessing the risk associated with a fixed-income position. VaR gives us a standardized method for measuring risk, and this method enables institutions in charge of risk management to manage risk in a novel way that was previously impractical. The fact that VaR considers the correlations between various risk variables is an additional feature. The VaR provides for this offset when two hazards cancel one other out, indicating that the aggregate risk is relatively modest.

VaR must be calculated using historical data and statistical models. The most popular strategy is to make use of the normal distribution assumption, which implies that the portfolio's returns are properly distributed. This assumption, however, can be problematic since market returns are typically not normally distributed and may have fat tails, meaning that extreme events are more likely to occur than a normal distribution would indicate. To address these constraints, more complex VaR models, such as Monte Carlo simulation and historical simulation, have been created. Monte Carlo simulation entails creating random future scenarios and estimating prospective losses in each scenario, whereas historical simulation uses actual market data to estimate potential future losses.

VaR addresses one of the most common questions an investor has: **what is the maximum expected loss of my investment?** VaR offers a highly natural and practical solution to this challenge. In this context, it is used to calculate a company's worst-case anticipated loss over a certain period and confidence range. Assume a daily VaR of $1 million at a 95% confidence level for an investment. This means there is a 5% chance that an investor may lose more than $1 million in a single day. According to this description, we may deduce that the components of VaR, as we are discussing risk, are a confidence interval, a time, the value of an asset or portfolio, and the standard deviation. In conclusion, certain critical elements in VaR analysis should be highlighted:

- VaR needs an estimation of the probability of loss.
- VaR concentrates on potential losses. We are not talking about actual or realized losses; rather, VaR is a kind of loss projection.
- VaR has three key ingredients
    - Standard deviation that defines the level of loss.
    - Risk is assessed over a fixed period.
    - The interval of confidence.

VaR can be calculated via three different approaches:

• Variance-covariance VaR

• Historical simulation VaR

• Monte Carlo VaR

### 5.2.2 Variance-Covariance Method

Because data are considered to be regularly distributed, the parametric technique is another name for the variance-covariance approach. The variance-covariance approach is often used because returns are assumed to follow a normal distribution. The parametric form assumption facilitates the implementation of the variance-covariance approach. We can use a single asset or a portfolio, as with all VaR techniques. Working with a portfolio, on the other hand, necessitates caution in that correlation structure and portfolio variance must be calculated. At this moment, there is no association comes into play, and historical data is utilised to compute correlation, mean, and standard deviation. When we supplement this with an ML-based method, the correlation structure will be our primary focus.

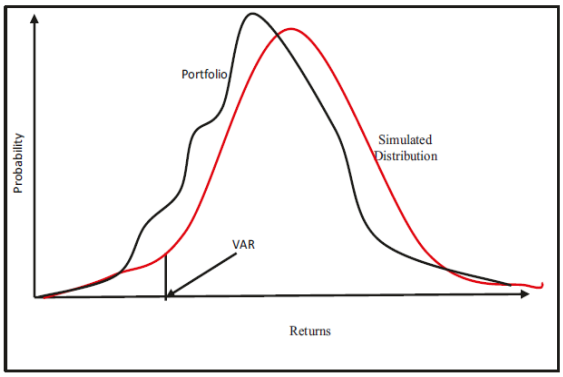


***Figure 5. 1 VaR***

Assume we have a single asset in our portfolio, as illustrated in Figure. It is demonstrated that the asset's return is zero and its standard deviation is one and that if the holding time is one, the equivalent VaR value can be determined from the value of the asset multiplied by the matching Z-value and standard deviation. As a result, while the normality assumption simplifies things, it is a strong assumption since there is no assurance that asset returns are normally distributed; rather, most asset returns do not follow a normal distribution. Furthermore, due to the normalcy assumption, any danger in the tail may be missed. As a result, the normalcy assumption comes with a cost. Consider the following: Here we show you how to calculate the VAR by applying the variance/ covariance calculation

Let's say your investment money is $5,000,000.

The standard deviation from a yearly trade schedule (252 days) is 9%.

The value at risk is calculated using the z-score (1.65) at a 95% confidence interval as follows:

$5,000,0000*1.645*.09 = $740 250

Correlation matrix on daily return

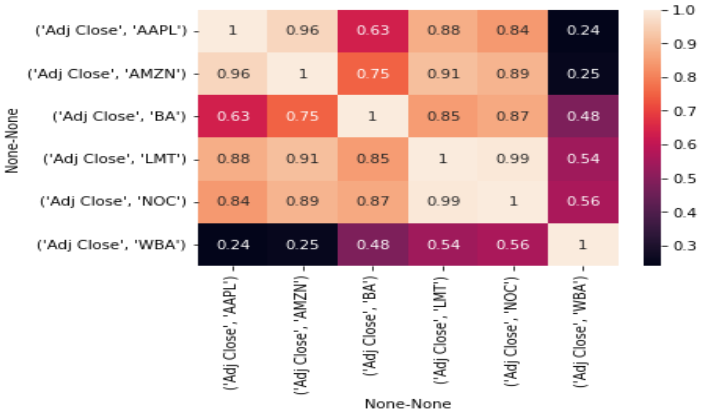


***Figure 5. 2 Correlation Matrix***

We start by specifying the average daily returns, then the confidence interval and cutoff value, and lastly the mean and standard deviations. Following that, we find the distribution's inverse.

At a 95% confidence level, the investment portfolio of $5,000,000 will not exceed daily losses of 174521.33.

**Let's look at 10-Day VaR**

***Table 5. 1 10-Day VaR values***

| Days | VaR @ 95% Confidence |
|---|---|
| 1 | 174521.33 |
| 2 | 246810.43 |
| 3 | 302279.81 |
| 4 | 349042.66 |
| 5 | 390241.56 |
| 6 | 427488.21 |
| 7 | 461740.04 |
| 8 | 493620.87 |
| 9 | 523563.99 |
| 10 | 551884.91 |

**10-Day VaR**

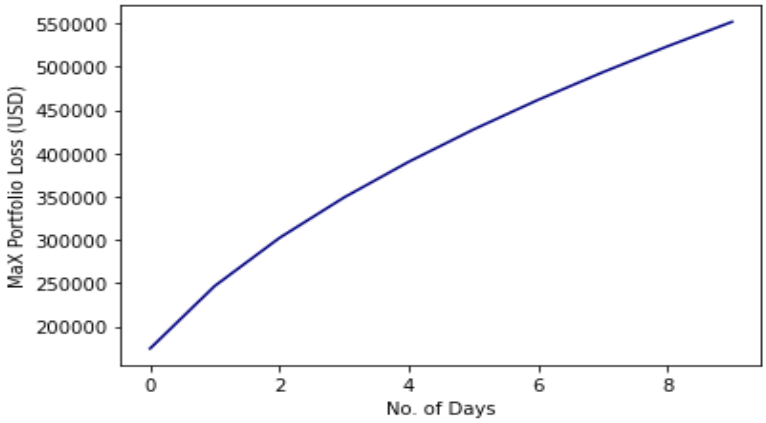


***Figure 5. 3 Max portfolio loss (VAR) over 10 days***

The pros and cons of the variance-covariance method are as follows:

- Pros
  - Easy to Calculate
  - Does not require a large number of samples

- Cons
    - Observations are normally distributed
    - Does not work well with the nonlinear structure
    - Requires the computation of the covariance matrix

As a result, while assuming normality seems tempting, it may not be the ideal strategy to estimate VaR, particularly when asset returns do not follow a normal distribution. Fortunately, another technique does not make any assumptions about normalcy, namely the Monte Carlo simulation VaR model.

### 5.2.3 Monte Carlo Simulation

Monte Carlo simulations are a type of computing technique that solves problems with a probabilistic interpretation by using repeated random sampling. One of the reasons they became popular in finance is because they can be used to reliably estimate integrals. The basic idea behind Monte Carlo simulations is to generate a large number of sample pathways (potential scenarios/outcomes), generally within a short period. The horizon is then divided into a predetermined number of time steps, a technique known as discretization. Its purpose is to approximate the continuous-time when financial products are priced. All of these simulated sample pathways may be used to produce metrics like the percentage of times an event occurred, the average value of an instrument at the last step, and so on. The fundamental issue with the Monte Carlo method in the past was that it required a lot of computer capacity to calculate all of the possible scenarios. Nowadays, we can perform quite sophisticated simulations on a desktop computer or a laptop, and if we run out of computational capacity, we can use cloud computing and its more powerful processors.

We will have seen how Monte Carlo techniques may be used in a variety of settings and jobs. Some will be created from scratch, while others will make use of recent Python packages to make the process even easier. Monte Carlo is one of the most essential approaches in computational finance due to its versatility. It can be applied to a variety of problems, including pricing derivatives with no closed-form solution (American/exotic options), bond valuation (for example, a zero-coupon bond), estimating portfolio uncertainty (for example, by calculating Value-at-Risk

and Expected Shortfall), and stress testing in risk management. In this step, we will demonstrate how to tackle some of these issues.

The logic behind Monte Carlo is well defined by Glasserman (2003, p. 11):

> *Monte Carlo methods are based on the analogy between probability and volume. The mathematics of measure formalizes the intuitive notion of probability, associating an event with a set of outcomes and defining the probability of the event to be its volume or measure relative to that of a universe of possible outcomes. Monte Carlo uses this identity in reverse, calculating the volume of a set by interpreting the volume as a probability.*

**Mathematical Monte Carlo can be defined in the following way:**

Let X1, X2,…Xn is a random variable with independent and identical distributions, and f(x) is a real-valued function. According to the law of huge numbers:

$$\mathsf{E}(f(X)) \approx \frac{1}{N}\sum_{i}^{N} f(X_i)$$

In a word, a Monte Carlo simulation does nothing more than generate random samples and calculate their means. Computationally, it goes as follows:

1. Define the domain
2. Generate random numbers
3. Iterate and aggregate the result

Mathematical determination is a basic yet interesting example of the Monte Carlo application.

**Simulating stock price dynamics using a geometric Brownian motion**

Simulating stock values is critical in the pricing of many derivatives, particularly options. Because price movements are erratic, these simulations rely on stochastic differential equations (SDEs). When a stochastic process meets the following SDE, it is said to follow a geometric Brownian motion (GBM).

$$dS_t = \mu S_t dt + \sigma S_t dW_t$$

Here, we have the following:

• $S_t$—Stock price

• $\mu$—The drift coefficient, that is, the average return over a given period or the instantaneous expected return

• $\sigma$ —The diffusion coefficient, that is, how much volatility is in the drift

• $W_t$ —The Brownian motion

• $d$—This symbolizes the change in the variable over the considered time increment, while $dt$ is the change in time

**A GBM is a process that does not account for mean-reversion and time-dependent volatility. That is why it is often used for stocks and not for bond prices, which tend to display long-term reversion to the face value.**

In this recipe, we use Monte Carlo methods and a GBM to simulate Microsoft's stock prices one month ahead—using data from 2022, we will simulate the possible paths over January 2023.

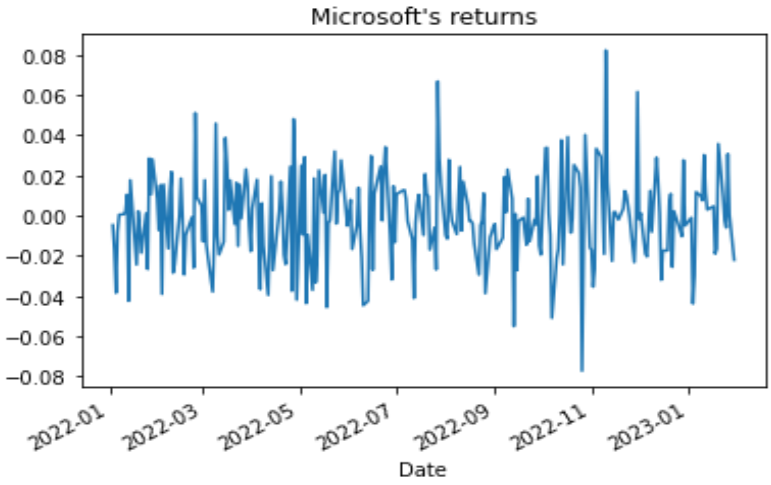


***Figure 5. 4 Microsoft Simple Return Plot***

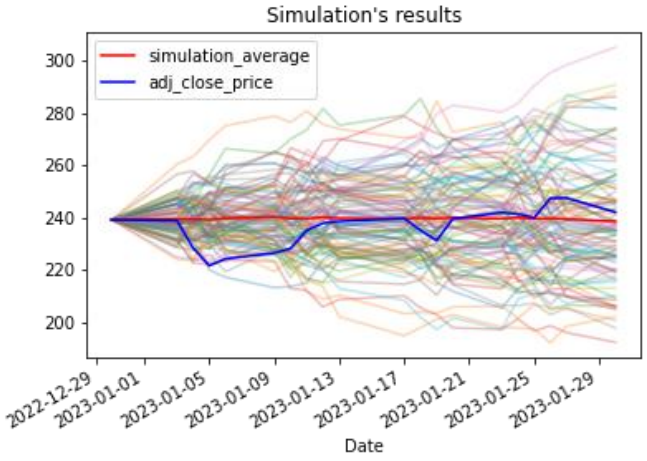


***Figure 5. 5 The Simulated path together with their average***

The forecasted stock prices (the averages of the simulations for each time step) show a slightly upward trend in Figure. This might be explained by the positive drift term $\mu$ = 0.07 per cent. Given the tiny number of simulations, we should take that result with a grain of salt.

Keep in mind that such a visualization is only possible with a sufficient number of sample pathways. In practice, we'd like to employ many more sample pathways than 100. In general, having more sample pathways leads to more accurate/reliable findings in Monte Carlo simulations.

### 5.2.4 Execution Method

We obtained IBM's stock prices and calculated the simple returns, followed by splitting the data into training and test sets. While no explicit model training was conducted, the training set was used to derive the average and standard deviation of the returns, which were employed as drift (mu) and diffusion (sigma) coefficients in our simulations. Additionally, in Step 5, we defined the parameters as:

- T: The forecasting horizon, represents the number of days in the test set.
- N: The number of time increments within the forecasting horizon. In our simulation, we set N to be equal to T.
- S_0: The initial price, determined by using the last observation from the training set in this simulation.
- N_SIM: Number of simulated paths

We have created a function to run the simulations, which is considered good practice for organizing and encapsulating code for such problems. Defining a function or a class can be beneficial as it provides modularity and reusability, making it convenient for future use as well.

Within the function's definition, we can identify the drift as (mu - 0.5 * sigma ** 2) multiplied by the number of time steps. Similarly, the diffusion is represented as sigma multiplied by the Wiener process, denoted as W. Notably, we utilized a vectorized approach while defining this function, avoiding the use of loops, which could be inefficient in scenarios involving large simulations.

After running the simulations, we stored the outcomes (sample paths) in a data frame. To facilitate plotting using the pandas DataFrame's plot method, we transposed the data, resulting in one path per column. To ensure proper indexing, we used the union method of a DatetimeIndex to combine the index of the last observation from the training set with the indices from the test set. This approach helped in organizing the data and preparing it for visualization.

# CHAPTER 6

# CREDIT DEFAULT PREDICTION

## 6.1 Introduction

In recent years, we have seen machine learning become increasingly popular in tackling classic business challenges. Now and then, a new algorithm is published that outperforms the existing state of the art. It is only natural for organizations (of all sizes) to strive to incorporate the remarkable capabilities of machine learning into their fundamental functions. This chapter looks at a binary classification problem from the finance business. We use a dataset from the UCI Machine Learning Source, which is a popular data repository. The dataset used in this chapter was collected from a Taiwanese bank in October 2005. The study was carried out.

Motivated by the fact that, at the time, an increasing number of banks were providing credit (either in cash or through credit cards) to willing clients. Furthermore, more people, regardless of their ability to repay, accrued huge sums of debt. As a result of all of this, some people were unable to fulfil their existing bills. In other terms, they went into default on their debts. I will be familiar with a real-world approach to a machine learning challenges, from data collection and cleaning through the construction and optimization of a classifier. Another important takeaway is comprehending the overall approach to machine learning projects, which can subsequently be applied to a variety of jobs. Whether it's predicting churn or evaluating the cost of new real estate in an area.

**The following recipes are the topic of this chapter:**

• Data loading and data type administration

• Exploratory data analysis

• Data is divided into training and test sets.

• Identifying and addressing missing values

• Categorical variable encoding

• Using pipelines to organize the project

• Using grid search and cross-validation to tune hyperparameters

**Loading Data and Managing Data type:**

This recipe demonstrates how to import a dataset from a CSV file into Python. The same ideas apply to other file formats as long as they are supported by pandas. Parquet, JSON, XLM, Excel, and Feather are some prominent formats.

We also demonstrate how certain data type conversions may drastically reduce the amount of Data Frames in our computers' memory. This is especially critical when working with huge datasets (GBs or TBs), which will simply not fit into memory unless their consumption is optimized.

We performed certain adjustments to the original dataset to depict a more realistic scenario (including jumbled data, missing values, and so on).

## 6.2 Dataset Information

This dataset contains information on default payments, demographic factors, credit data, payment history, and bill statements for credit card clients in Taiwan from April 2005 to September 2005.

**Content:**

The Dataset content has 25 variables:

- **ID**: ID of each client
- **LIMIT_BAL**: Amount of given credit in NT dollars (includes individual and family/supplementary credit
- **SEX**: Gender (1=male, 2=female)
- **EDUCATION**: (1=graduate school, 2=university, 3=high school, 4=others, 5=unknown, 6=unknown)
- **MARRIAGE**: Marital status (1=married, 2=single, 3=others)
- **AGE**: Age in years
- **PAY_0**: Repayment status in September, 2005 (-1=pay duly, 1=payment delay for one month, 2=payment delay for two months, … 8=payment delay for eight months, 9=payment delay for nine months and above)
- **PAY_2**: Repayment status in August 2005 (scale same as above)
- **PAY_3**: Repayment status in July 2005 (scale same as above)
- **PAY_4**: Repayment status in June 2005 (scale same as above)
- **PAY_5**: Repayment status in May 2005 (scale same as above)
- **PAY_6**: Repayment status in April 2005 (scale same as above)
- **BILL_AMT1**: Amount of bill statement in September 2005 (NT dollar)
- **BILL_AMT2**: Amount of bill statement in August 2005 (NT dollar)
- **BILL_AMT3**: Amount of bill statement in July 2005 (NT dollar)
- **BILL_AMT4**: Amount of bill statement in June 2005 (NT dollar)
- **BILL_AMT5**: Amount of bill statement in May 2005 (NT dollar)

- **BILL_AMT6**: Amount of bill statement in April 2005 (NT dollar)
- **PAY_AMT1**: Amount of previous payment in September 2005 (NT dollar)
- **PAY_AMT2**: Amount of previous payment in August 2005 (NT dollar)
- **PAY_AMT3**: Amount of previous payment in July 2005 (NT dollar)
- **PAY_AMT4**: Amount of previous payment in June 2005 (NT dollar)
- **PAY_AMT5**: Amount of previous payment in May 2005 (NT dollar)
- **PAY_AMT6**: Amount of previous payment in April 2005 (NT dollar)
- **default.payment.next.month**: Default payment (1=yes, 0=no)

The is the preview of the dataset:

***Table 6. 1 Preview of the Dataset***

| ID | LIMIT_BAL | SEX | EDUCATION | MARRIAGE | AGE | PAY_0 | PAY_2 | PAY_3 | PAY_4 | ... | BILL_AMT4 | BILL_AMT5 | BILL_AMT6 | PAY_AMT1 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 1 | 20000 | 2 | 2 | 1 | 24 | 2 | 2 | -1 | -1 | ... | 0 | 0 | 0 |
| **1** | 2 | 120000 | 2 | 2 | 2 | 26 | -1 | 2 | 0 | 0 | ... | 3272 | 3455 | 3261 |
| **2** | 3 | 90000 | 2 | 2 | 2 | 34 | 0 | 0 | 0 | 0 | ... | 14331 | 14948 | 15549 |
| **3** | 4 | 50000 | 2 | 2 | 1 | 37 | 0 | 0 | 0 | 0 | ... | 28314 | 28959 | 29547 |
| **4** | 5 | 50000 | 1 | 2 | 1 | 57 | -1 | 0 | -1 | 0 | ... | 20940 | 19146 | 19131 |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **29995** | 29996 | 220000 | 1 | 3 | 1 | 39 | 0 | 0 | 0 | 0 | ... | 88004 | 31237 | 15980 |
| **29996** | 29997 | 150000 | 1 | 3 | 2 | 43 | -1 | -1 | -1 | -1 | ... | 8979 | 5190 | 0 |
| **29997** | 29998 | 30000 | 1 | 2 | 2 | 37 | 4 | 3 | 2 | -1 | ... | 20878 | 20582 | 19357 |
| **29998** | 29999 | 80000 | 1 | 3 | 1 | 41 | 1 | -1 | 0 | 0 | ... | 52774 | 11855 | 48944 |
| **29999** | 30000 | 50000 | 1 | 2 | 1 | 46 | 0 | 0 | 0 | 0 | ... | 36535 | 32428 | 15313 |

The Data Frame has 30,000 rows and 25 columns. It contains a mix of numeric and categorical variables.

## 6.3 Identifying and addressing missing values

It is typical to encounter missing values while working with real-world data. Missing values can be classed based on why they are missing:

- **Missing completely at random (MCAR):** The cause for the missing data has nothing to do with the remainder of the data. In a survey, for example, a respondent may inadvertently leave out a question.

- **Missing at Random (MAR):** Data in another column can be used to infer the missingness of the data (s). A missing response to a survey question, for example, might be decided conditionally by other characteristics such as gender, age, lifestyle, and so on.
- **Missing not at Random (MNAR):** When the missing numbers have an underlying cause. People with really high earnings, for example, are often unwilling to share it.
- **Structurally Missing Data:** Frequently, a portion of MNAR data is absent due to a reasonable reason. For example, if a variable reflecting a spouse's age is absent, we can deduce that a specific person does not have a spouse.

It is true that some machine learning algorithms, such as decision trees, can handle missing data by treating them as a separate category. However, not all algorithms can do this, and popular implementations like those found in sci-kit-learn may not include this functionality. As a result, dealing with missing data is a critical pre-processing step in machine learning since it might impair the model's accuracy and performance. There are several strategies for dealing with missing data, such as imputation, deletion, or using algorithms that can handle missing values.

## Some prominent methods for dealing with missing values are:

1. **Imputation strategies**:
    - Mean imputation: replace missing values with the mean of the observed values for that feature.
    - Median imputation: replace missing values with the median of the observed values for that feature.
    - Regression imputation: predict the missing values based on the relationship between the missing feature and other features in the dataset using regression models.
    - K-nearest neighbour imputation: replace missing values with the values of the k-nearest neighbours of the missing value in the feature space.
2. **Deletion strategies**:
    - Listwise deletion: remove entire rows with missing values, also known as complete case analysis.
    - Pairwise deletion: remove missing values on a per-feature basis, so that each feature is analyzed using only the rows where it has valid data.
    - Feature deletion: remove features with too many missing values or where missing values are highly correlated with other features

3. **Algorithm-specific strategies:**
    - Decision trees: treat missing values as a separate and unique category.
    - Random Forest: can handle missing values by imputing them during the construction of each tree in the forest using imputation methods like regression imputation.
    - Support Vector Machines: use imputation methods to handle missing data.
    - Bayesian networks: can handle missing values using Bayesian inference.
    - Deep learning: can handle missing values by incorporating missing value indicators as a separate input feature or by using recurrent neural networks.
4. **Multiple imputation strategies:**
    - Generate multiply imputed datasets by estimating the missing values multiple times, each time with a different imputation model.
    - Analyze each imputed dataset separately using standard statistical methods.
    - Combine the results from the multiple datasets to produce a final estimate.

It is critical to properly assess the effectiveness of various techniques and choose the one that is most appropriate for the specific problem at hand. In some cases, multiple strategies may need to be tried and evaluated to determine the best approach.

We executed different codes to get the following results:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 30000 entries, 0 to 29999
   Data columns (total 25 columns):
 #   Column                      Non-Null Count  Dtype
---  ------                      --------------  -----
 0   ID                          30000 non-null int64
1   LIMIT_BAL                   30000 non-null float64
 2   SEX                         30000 non-null int64
 3   EDUCATION                   30000 non-null int64
 4   MARRIAGE                    30000 non-null int64
 5   AGE                         30000 non-null int64
 6   PAY_0                       30000 non-null int64
 7   PAY_2                       30000 non-null int64
 8   PAY_3                       30000 non-null int64
 9   PAY_4                       30000 non-null int64
 10 PAY_5                       30000 non-null int64
 11 PAY_6                       30000 non-null int64
12 BILL_AMT1                   30000 non-null float64
13 BILL_AMT2                   30000 non-null float64
14 BILL_AMT3                   30000 non-null float64
15 BILL_AMT4                   30000 non-null float64
16 BILL_AMT5                   30000 non-null float64
17 BILL_AMT6                   30000 non-null float64
18 PAY_AMT1                    30000 non-null float64
19 PAY_AMT2                    30000 non-null float64
```

```
20  PAY_AMT3                    30000 non-null  float64
21  PAY_AMT4                    30000 non-null  float64
22  PAY_AMT5                    30000 non-null  float64
23  PAY_AMT6                    30000 non-null  float64
 24  default.payment.next.month  30000 non-null  int64
            dtypes: float64(13), int64(12)
                 memory usage: 5.7 MB
```

***The data to observe missing values***

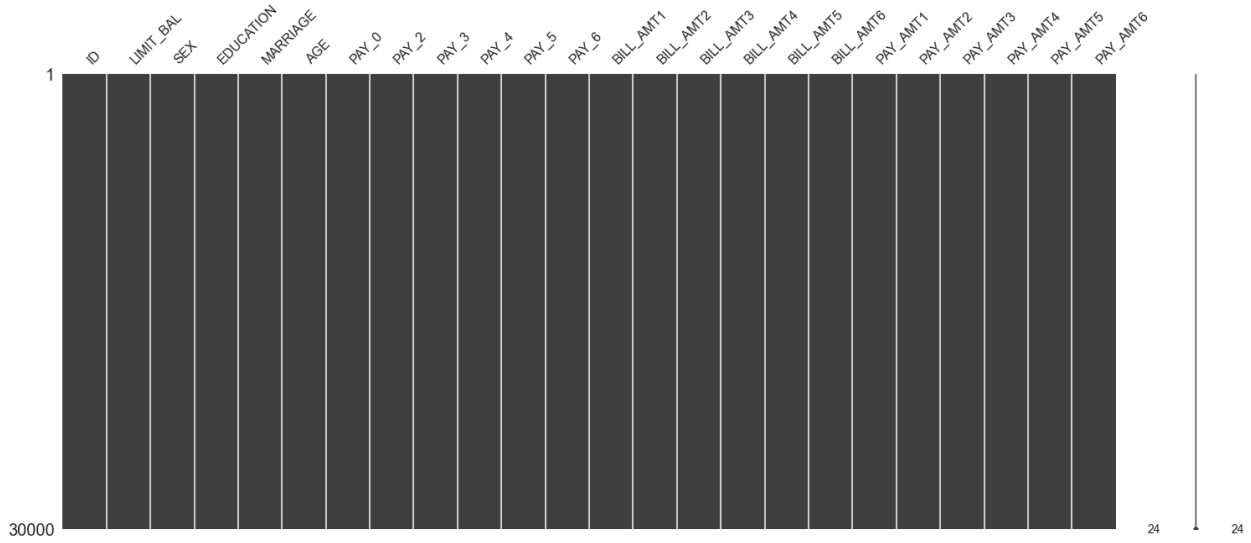


***Figure 6. 1 The loan default dataset's nullity matrix plot***

Our observation has no missing values. If there are any missing values, we could have seen a white line in the columns. While analyzing a large dataset finding the missing values and mitigating those values are quite challenging tasks. So, our dataset doesn't have any missing values that would be easier for us to analyze our dataset.

After loading the relevant libraries, we used the pandas DataFrame's info method to retrieve specific information about the columns, such as their data type and the quantity of non-null value entries. The difference between the total number of values and the non-null values equals the number of missing values. The X.isnull().sum() function may also be used to determine the amount of missing data per column.

## 6.4 Exploratory Data Analysis

Exploratory Data Analysis (EDA) is the second phase in our credit default analysis. We learn about the data we'll be working with this way. This is also the stage at which we assess the breadth of our topic knowledge. For example, the firm for which we work may believe that the bulk of its clients are between the ages of 18 and 25. Is this, however, the case? While doing EDA, we may see patterns that we do not understand, which can serve as a starting point for a dialogue with our stakeholders.

While doing EDA, we can try to answer the following questions:

- What kind of data do we have, and how should we treat different data types?
- What is the distribution of the variables?
- Are there outliers in the data and how can we treat them?
- Are any transformations required? For example, some models work better with (or require) normally distributed variables, so we might want to use techniques such as log transformation.
- Does the distribution vary per group (for example, sex or education level)?
- Do we have cases of missing data? How frequent are these, and in which variables do they occur?
- Is there a linear relationship (correlation) between some variables?
- Can we create new features using the existing set of variables? An example might be deriving an hour/minute from a timestamp, a day of the week from a date, and so on.
- Are there any variables that we can remove as they are not relevant to the analysis? An example might be a randomly generated customer identifier.

EDA is critical in all data science projects because it allows us to create knowledge of the data, allows us to ask better questions, and makes it simpler to choose modelling methodologies appropriate for the type of data being dealt with. In practice, it is best to start with univariate analysis (one feature at a time) of all important aspects to have a thorough grasp of them. Then we may go on to multivariate analysis, which involves comparing distributions by group, correlations, and other factors. We just demonstrate selected analytic techniques to selected characteristics for brevity, but a deeper investigation is strongly urged.

### 6.4.1 Exploring the Data

We may conclude from the kernel density estimate (KDE) plot that there is little change in the shape of the distribution by sex. On average, the female sample is somewhat younger.

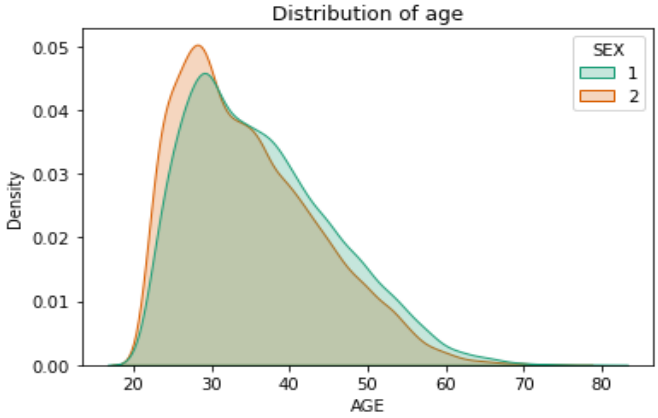


***Figure 6. 2 The KDE plot of age, grouped by sex***

**Creating a Pair Plot of Selective Variables:**

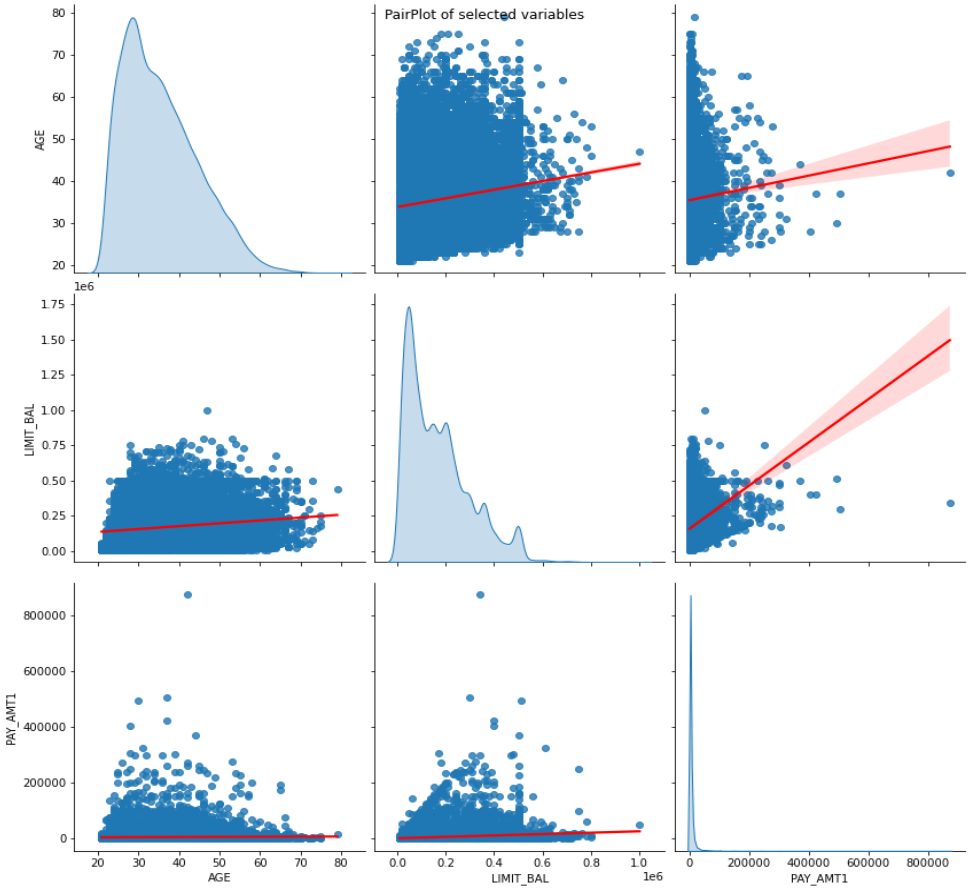


***Figure 6. 3 A pair plot with KDE plots on the diagonal and fitted regression lines in each scatterplot***

We make a few observations from the pair plot:

- The distribution of **PAY_AMT1** is highly skewed—it has a very long tail.
- Connected to the previous point, we can observe some very extreme values of **PAY_AMT1** in the scatterplots.
- Because each scatterplot contains 30,000 observations, it is difficult to conclude them. When plotting such large amounts of data, we may utilise transparent markers to better visualize the density of observations in specific places.
- The outliers can have a significant impact on the regression lines.

Additionally, we can separate the sexes by specifying the hue argument:

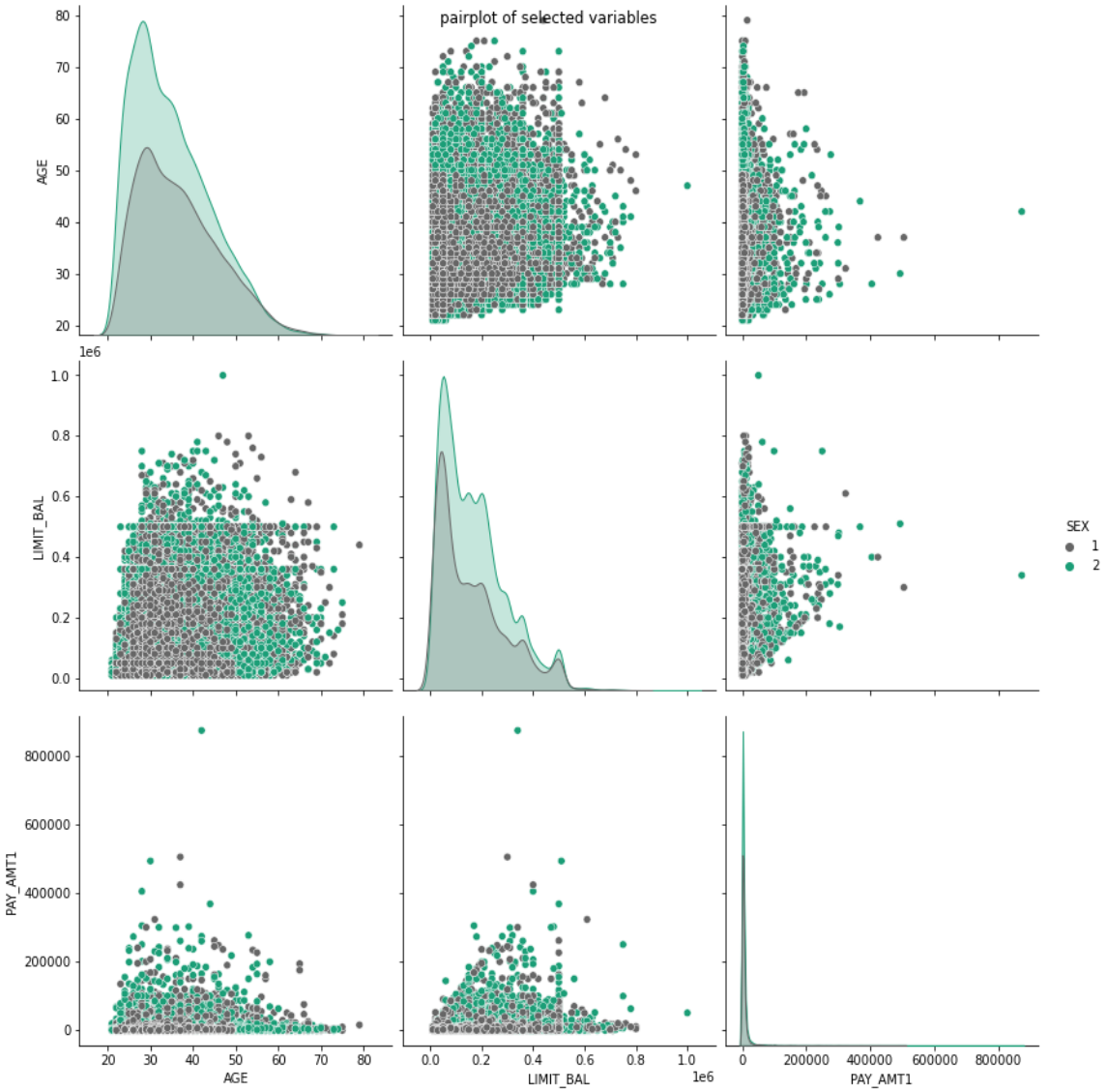


***Figure 6. 4 The pair plot with separate markers for each sex***

While the diagonal plots with the split per sex provide more information, the scatterplots are still difficult to read due to the sheer volume of presented data.

As an alternative, we might choose a random sample from the complete dataset and only plot the selected observations. One disadvantage of this strategy is that we may miss some observations with extreme values (outliers)

**Analyze the relationship between AGE and LIMIT BALANCE:**

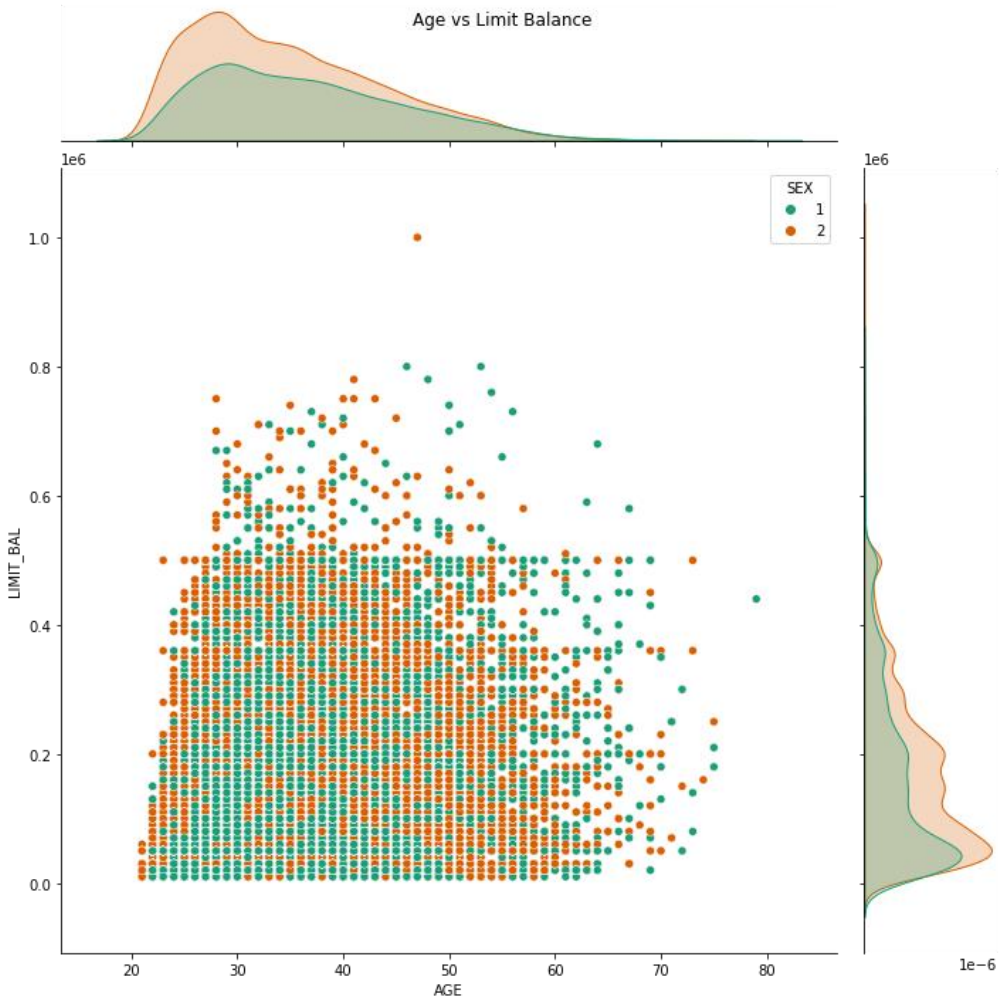


***Figure 6. 5 A joint plot showing the relationship between age and limit balance, grouped by sex***

A combined plot contains a wealth of information. First and foremost, the scatterplot shows the relationship between two variables. Then, using the KDE plots along the axes, we can analyse the distributions of the two variables independently (we may also display histograms instead).

### 6.4.2 Correlation Observation

**Let's observe a correlation heatmap in which variables are highly correlated to each other's**

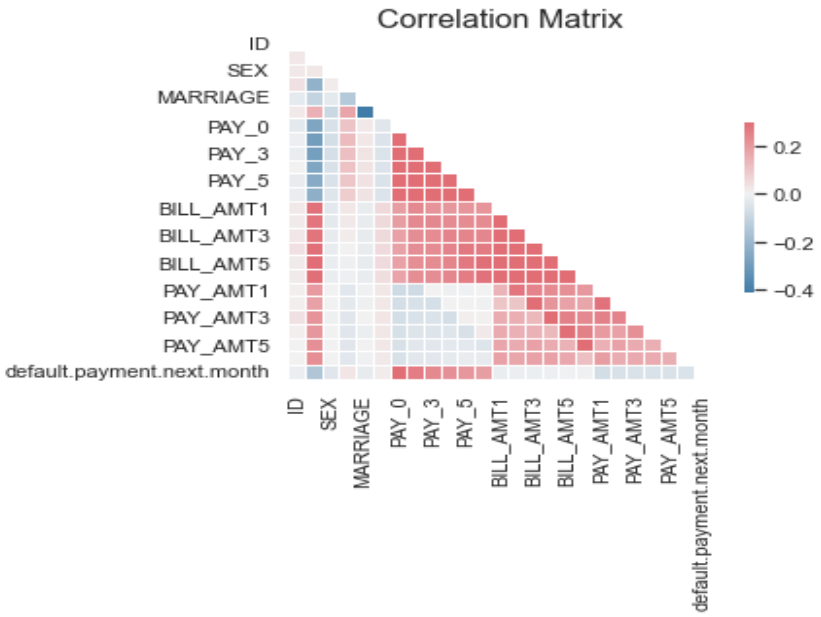


***Figure 6. 6 Correlation Matrix***

Marriage appears to be uncorrelated to any of the other characteristics.

### 6.4.3 Data Distribution Visualization

Examine the age distribution in groups using box plots:

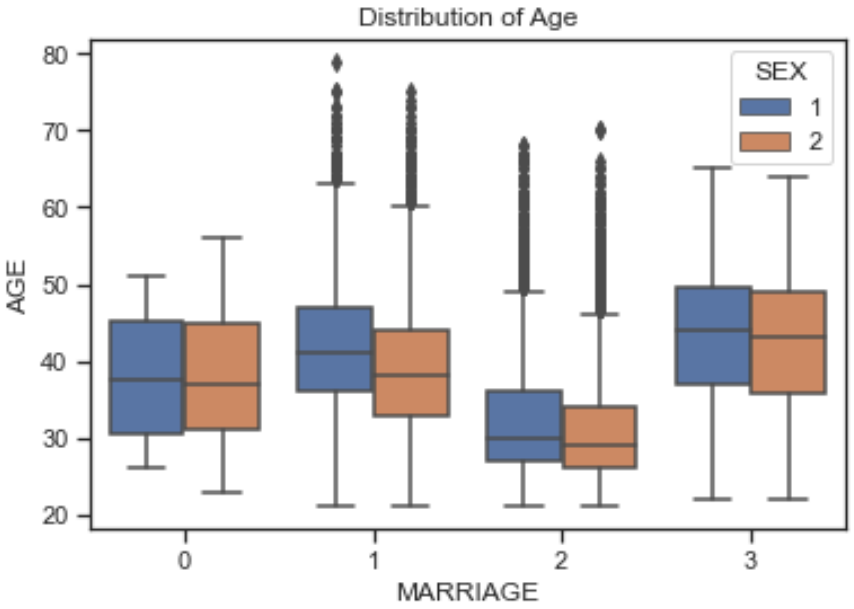


***Figure 6. 7 Distribution of Age by Marriage Status and Gender***

**Plot the limit balance distribution by Gender and educational level:**

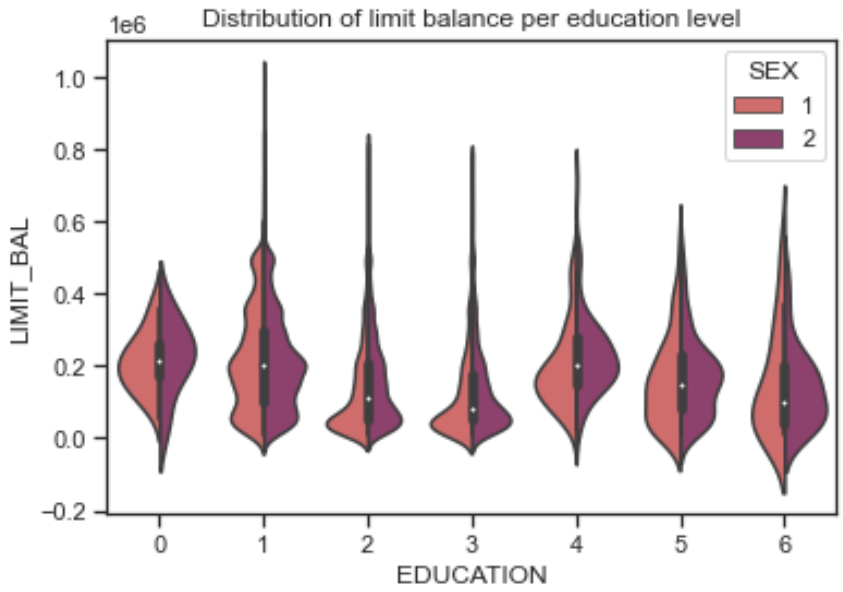


***Figure 6. 8 Distribution of Limit balance by Gender and education***

**Examining the plot shows the following intriguing patterns:**

- The group with graduate school level education has the biggest balance.
- According to education level, the distribution takes on a varied shape: Graduate school levels resemble the other group, whereas high school levels resemble the University levels.
- There aren't many disparities between the genders generally.

**Look into how the target variable is distributed by Gender and educational level:**

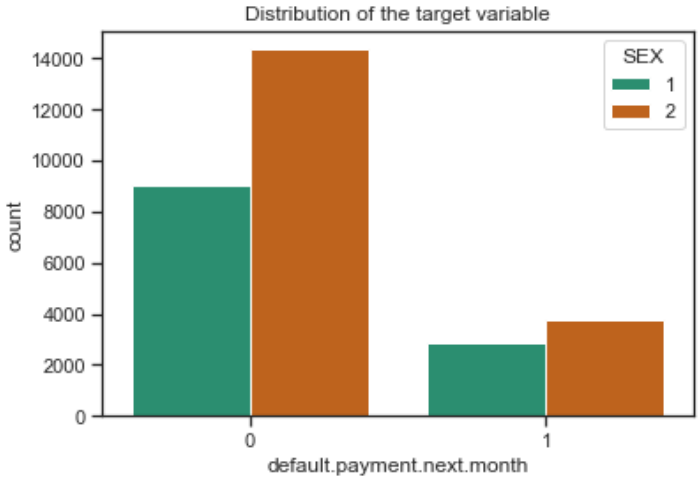


***Figure 6. 9 Distribution of Target variable by Gender***

We may infer from the plot analysis that male clients have a greater default rate.

**Examine the proportion of defaults by educational level:**

We got the following plot:

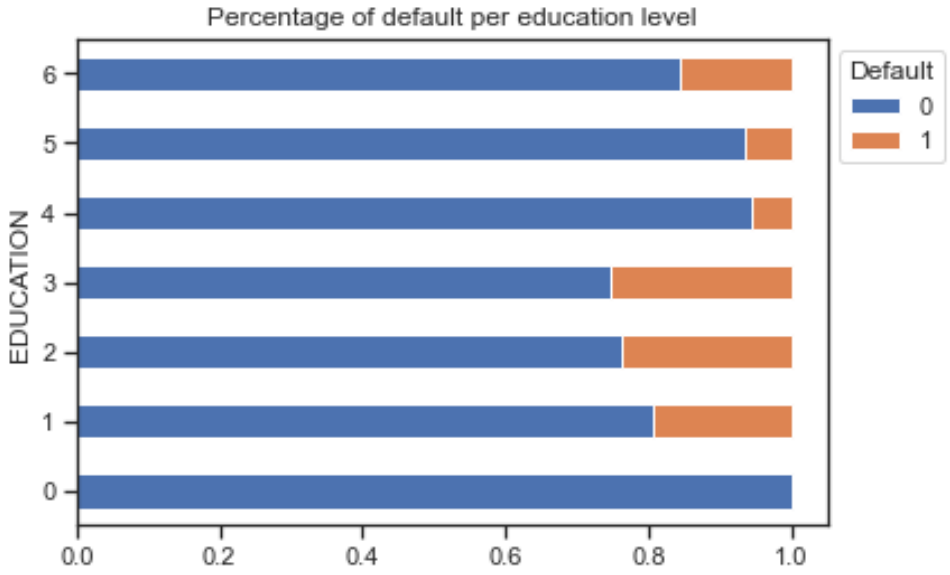


***Figure 6. 10 Average Default Rate by Education***

Clients with a high school diploma experience the majority of defaults, whilst other customers have the fewest defaults.

### 6.4.4 Analysis, how it Works

We will be primarily using the seaborn library in this recipe, as it is widely used for data exploration. However, there are other libraries we could use for plotting. For instance, the pandas' plot method is a robust tool that enables us to visualize data quickly. We could also use Plotly, including its plot. express module, to develop fully interactive data visualizations**.**

Our analysis began by utilizing the panda's DataFrame method 'describe.' This approach is straightforward but highly effective, producing summary statistics for all the numeric variables present in the dataset, such as count, mean, min/max, and quartiles. With this information, we were able to deduce the potential range of a feature's values and identify skewed distributions by examining the difference between the mean and median. Additionally, this approach allowed us to easily pinpoint values outside of the typical range, such as negative or excessively young or old ages.

In our analysis, we demonstrated a method for investigating the distribution of a variable - specifically, the age of customers - by creating a KDE plot. This technique is similar to a traditional histogram and visualizes a variable's distribution using a continuous probability density curve in one or more dimensions. One of the significant advantages of KDE plots is that they result in less cluttered and more straightforward-to-interpret plots, especially when working with multiple distributions simultaneously.

KDE plots, along with histograms, are some of the most commonly used methods for inspecting the distribution of a single feature. We can create a histogram using the SNS.histplot function or by specifying the kind argument to "hist" in the plot method of a pandas DataFrame.

We can extend this analysis further by utilizing a pair plot, which creates a matrix of plots. The diagonal plots show univariate histograms or KDE plots, while the off-diagonal plots display scatterplots of two features. By including regression lines, we can identify any potential relationships between the two features. When working with a large dataset such as 30,000 observations, it can be challenging to render plots for all numeric columns while maintaining readability. Thus, we often plot only a select few features. We can also utilize the hue argument to add a split for a particular category, such as Gender or education level.

We can define a function for plotting a heatmap that represents the correlation matrix. Within the function, we can use operations to mask the upper triangular matrix and the diagonal, where all diagonal elements of the correlation matrix are equal to 1. This helps make the output much easier to interpret. If desired, we can use the ‘annot’ argument of ‘sns. heatmap’ to add the underlying numbers to the heatmap. However, it is crucial to be mindful of the number of analyzed features. Too many elements might make the heatmap difficult to understand. We used box plots to examine the age distribution by gender and marital status. A box plot, also known as a box-and-whisker plot, depicts data distribution in a form that allows for easy comparison of categorical variable values.

"In the previous steps, we explored the distribution of the target variable (default) by sex and education. For sex, we used ‘**sns.counterplots** to show the count of occurrences of both possible outcomes for each gender. For education, we wanted to plot the percentage of defaults per education level as comparing percentages between groups is easier than comparing nominal values. To achieve this, we grouped the data by education level, selected the variable of interest,

calculated the percentages per group using the value_counts (normalize=True) method, unstacked the multi-index, and generated a plot using the '**plot'** method.

## 6.5 Dividing the data into test and training sets

After the EDA is finished, the dataset must be divided into training and test sets. Two distinct datasets are intended to be used:

**Training Set:** Our machine learning model is trained using this portion of the data.

**Test Set:** This portion of the data is utilised to assess its significance because the model was not aware of it during training.

This data splitting is intended to avoid **overfitting. Overfitting** is a phenomenon that happens when a model only works well on the set of training data it identifies too many patterns in. In other words, it is unable to extrapolate from existing data.

Handling missing data correctly and detecting outliers is a critical step in data analysis to avoid bias, such as data leaking. Data leaking happens when a model gets information that it should not have during training. Imputing missing values with the feature's average is one example of this. If this is done before dividing the data, the average will be calculated using data from the test set, introducing leakage. To avoid this, divide the data into training and test sets first, and then do imputation using just the training set data. The same logic applies when it comes to finding outliers in data. Handling missing data correctly and detecting outliers is a critical step in data analysis to avoid bias, such as data leaking. Data leaking happens when a model gets information that it should not have during training. Imputing missing values with the feature's average is one example of this. If this is done before dividing the data, the average will be calculated using data from the test set, introducing leakage. To avoid this, divide the data into training and test sets first, and then do imputation using just the training set data. The same logic applies when it comes to finding outliers in data.

Splitting the data not only eliminates bias but also maintains consistency for future unknown data, such as new consumers rated by the algorithm. These fresh observations will be processed in the same way as the data in the test set, increasing the model's reliability and trustworthiness.

## 6.6 Data Split

We exhibited the most fundamental split in our research by giving X and y objects to the train test split method. In addition, for repeatability, we determined the size of the test set as a fraction of all observations and established the random state. Finally, we made four new objects by assigning the function's output to them.

By setting test size=0.2 and shuffle=False, we took an alternative method. As a consequence, the first 80% of the remaining 20% of the data was assigned to the training set. to the test set. This method is beneficial when we wish to keep the sequence in which the observations become accessible, as with time-series data.

Furthermore, the stratification argument was incorporated by supplying the target variable (stratify=y) during the data split. Stratified splitting guarantees that the provided variable is distributed roughly identically in both the training and test sets. When dealing with unbalanced data, such as in fraud detection, when only a tiny fraction of observations may belong to the target class, this value is critical. A random split may result in no occurrences of the target class in the training set, which can have a major influence on model performance. When dealing with uneven data, stratified splitting is crucial.

We utilised the value counts function of a pandas DataFrame to guarantee that the stratified train/test split resulted in the same proportion of defaults in both datasets.

We get the following output:

***Table 6. 2 Data Distribution***

| Dataset | Target Distribution |
|---|---|
| Train | [0.77879167, 0.22120833] |
| Test | [0.77883333, 0.22116667] |

The rate of payment failures in both groups is roughly 22.12 per cent.

**There's more to it.**

In addition to training and testing, it is usual to divide data into three sets: training, validation, and testing. The validation set is utilised regularly to evaluate and tweak the model's hyperparameters. For example, imagine we wish to train a decision tree classifier and identify the best value for the max depth hyperparameter, which controls the tree's maximum depth. In such an instance, we may use the training set to train the model numerous times, each time with a different value of the

hyperparameter. Then, using the validation set, we can assess the performance of all of these models. We choose the best model based on its performance on the validation set and then analyse it on the test set.

We effectively separated the data into three groups by using the train test split method twice (training, validation, and test). However, adjusting the values of the test size parameter was critical to keep the initially stated proportions of 70-10-20. We also confirmed that the size of each dataset corresponds to the anticipated split and that the proportion of defaults in each set is the same. To validate this, the following code piece was used:

***Table 6. 3 Data Split Values***

| Dataset | Percentage of Data | Class Distribution |
|---|---|---|
| Train | 70.00% | [0.77879899, 0.22120101] |
| Valid | 9.90% | [0.77878788, 0.22121212] |
| Test | 20.10% | [0.77880948, 0.22119052] |

The original dataset was successfully divided into the desired 70-10-20 ratio, and the distribution of the target variable was kept via stratification. However, due to a short number of observations or a highly unbalanced dataset, we may not have enough data to separate into three groups at times. Cross-validation can be effective in certain situations. This approach will be covered in full in the recipe "Tuning hyperparameters with grid search and cross-validation."

## 6.7 Implementing a Decision Tree Classifier

A decision tree classifier is a well-known machine learning approach that may be used for regression as well as classification. It is known as a decision tree because it generates a collection of rules that may be seen as a tree. The model divides the feature space into smaller sections by dividing the features at a certain value periodically. To do this, the method employs a greedy technique, as well as certain heuristics, to choose a split that minimizes the aggregate impurity of the children's nodes. The impurity in classification issues is assessed using either Gini impurity or entropy, whereas the meter in regression problems is a mean squared error or mean absolute error.

For binary classification issues, the decision tree approach seeks to generate nodes with as many observations from one class as feasible, hence avoiding impurity. The mode in classification issues and the mean in regression problems establish the forecast for a terminal node or leaf.

**Advantages of Decision Trees:**

- **Easy to Understand and Interpret**: Decision trees are simple to comprehend and interpret. They provide a visual representation of the decision-making process and can be easily explained to non-experts.
- **Efficient for Large Datasets:** Decision trees can handle large datasets with high dimensionality. They can handle both numerical and categorical data and do not require extensive data preprocessing.
- **Non-Parametric:** Decision trees are non-parametric, which means they make no assumptions about the data's distribution. They are capable of capturing complicated non-linear correlations between input variables and destination variables.
- **Feature Selection:** Decision trees can help identify the most important features in a dataset. The algorithm selects the features that are most relevant for making a decision, which can be useful for feature selection.
- **Robust to Outliers:** Outliers and missing data are not a problem for decision trees. They can handle noisy data and missing values by making decisions based on the available data.

**Disadvantages of Decision Trees:**

- **Overfitting:** Overfitting occurs when decision trees produce too complicated models that match the training data too well but do not generalize to new data.
- **Instability:** Decision trees can be unstable, which means that little changes in the data might cause massive changes in the tree's structure. This can make decision trees difficult to interpret and can reduce their accuracy.
- **Bias:** Decision trees can be skewed toward features with a lot of levels or a lot of cardinalities. This can lead to a biased model that may not accurately represent the underlying data.
- **Greedy Approach:** Decision trees use a greedy approach to make decisions, meaning they make locally optimal decisions at each node without considering the global optimum. This can lead to suboptimal models.

- **Complexity:** Decision trees can become very complex, especially for large datasets with many input features. This can make decision trees difficult to interpret and can reduce their accuracy.

In summary, decision trees are a popular machine learning algorithm with several advantages, including their ease of interpretation, efficiency with large datasets, non-parametric nature, feature selection ability, and robustness to outliers. However, they also have some disadvantages, including their tendency to overfit, instability, bias towards certain features, greedy approach, and potential complexity.

## 6.8 Visualization of Decision Tree

To train our machine learning model, we used the scikit-conventional learning technique. First, we built a DecisionTreeClassifier object with the default defaults and a fixed random state. We then used the fit method to fit the model to the training data, passing both the features and the goal. Finally, we used the prediction technique to acquire the predictions. From the figure below we can visualize the decision tree values.

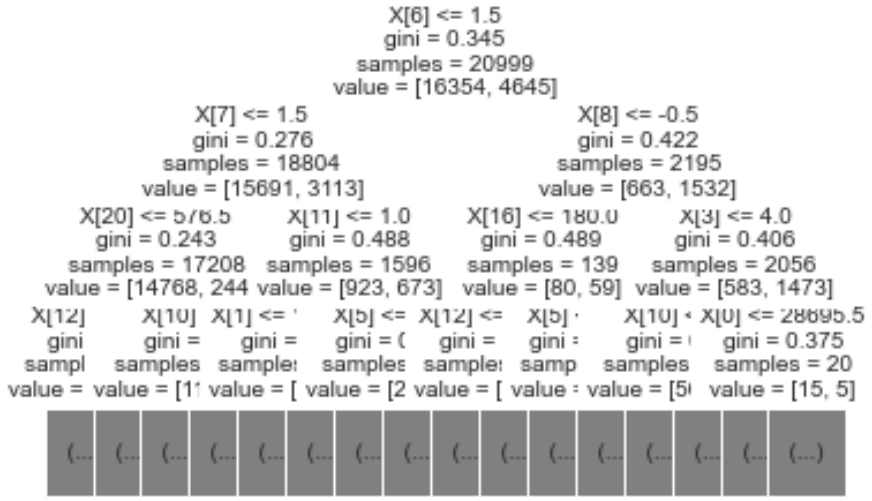


***Figure 6. 11 Gini Values Chart***

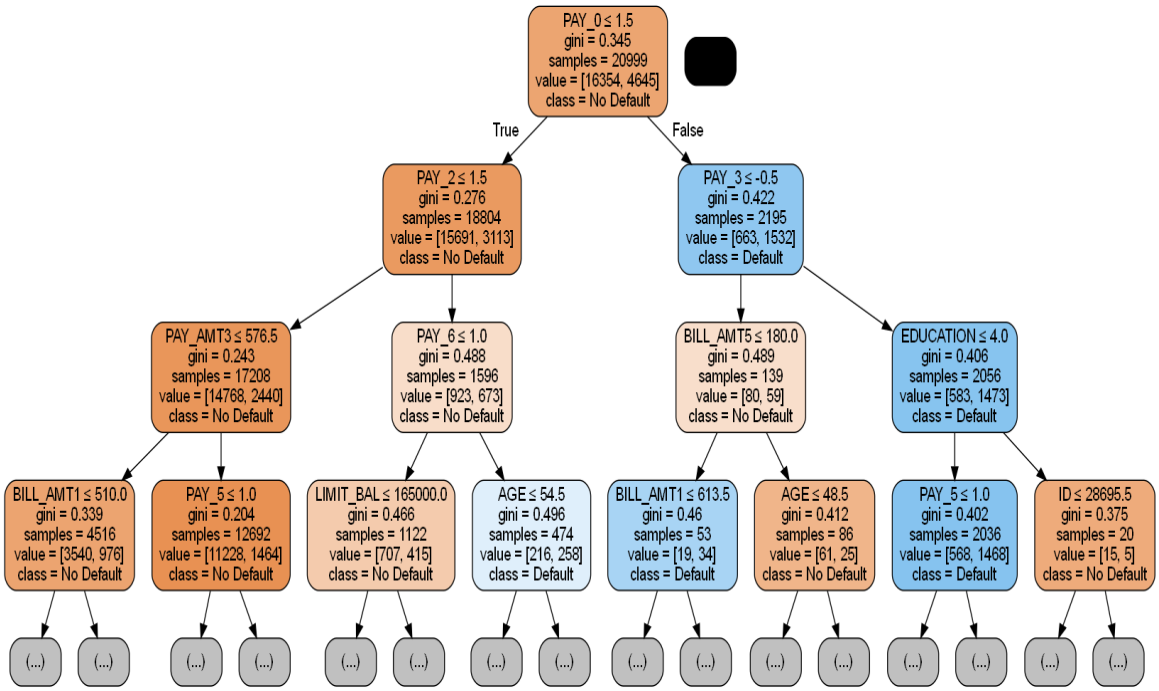


***Figure 6. 12 Decision tree***

## 6.9 Performance Metrics

### 6.9.1 Confusion Matrix

We evaluated the model's performance by displaying the results with a custom function built with features from the scikit-metrics learns module. While we won't get into the specifics of the function because it follows normal procedures, it successfully displayed the evaluation results.

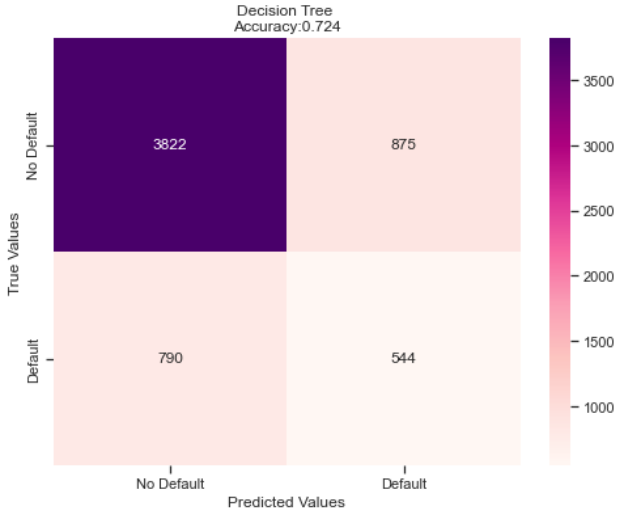


***Figure 6. 13 Confusion Matrix***

The investigation has uncovered that although the Decision Tree model has an overall F1 score of 0.724, the F1 scores for default clients are less than 50%. This could be attributed to the data being highly unbalanced, which may hinder the model's ability to generalize. As the next step, strategies such as resampling, cost-sensitive learning, or ensemble methods will be explored to rebalance the dataset and enhance the model's performance for minority classes. These approaches aim to address the issue of unbalanced data and improve the F1 score for better results.

***Table 6. 4 Table Classification Report***

| Class | Precision | Recall | F1-Score | Support |
|---|---|---|---|---|
| No Default | 0.83 | 0.81 | 0.82 | 4697 |
| Default | 0.38 | 0.41 | 0.4 | 1334 |
| Accuracy | | | 0.72 | 6031 |
| Macro Avg | 0.61 | 0.61 | 0.61 | 6031 |
| Weighted Avg | 0.73 | 0.72 | 0.73 | 6031 |

From the Above classification model, we can draw the following model evaluation criteria:

- **Accuracy** is a performance metric used in machine learning that quantifies the model's overall ability to properly forecast the class of an observation. In a classification issue, it is measured as the ratio of true positives (TP) and true negatives (TN) to the total of true positives, false positives (FP), true negatives, and false negatives (FN).
- **Precision** is a performance metric used in machine learning that measures the fraction of all predictions of the positive class (e.g., default) that were indeed positive. It is determined by dividing the number of true positives (TP) by the total number of true positives and false positives (FP) in a classification problem. In the context of a project, precision answers the question: "Out of all predictions of default, how many clients defaulted?" In other words, precision indicates the accuracy of the model's predictions when it predicts the positive class, or how often the model is correct when it predicts default.
- **Recall**, also known as the true positive rate or sensitivity, is a performance metric used in machine learning that measures the fraction of all positive cases (e.g., defaults) that were predicted correctly. It is calculated as the proportion of true positives (TP) to the total number of true positives and false negatives (FN) in a classification problem. In the context of a project, recall answers the question: "What fraction of all observed defaults did we

predict correctly?" In other words, recall indicates the model's ability to capture all the positive cases, or how well the model detects defaults among all the observed cases.

- **The F1 score** is a performance metric used in machine learning that is the summing of precision and recall. It is determined as the harmonic mean of accuracy and recall, which takes the harmony or resemblance of the two scores into consideration. Because it penalises extreme results and encourages excessively uneven values, the harmonic mean is utilised instead of the arithmetic average. For example, a classifier with precision = 1 and recall = 0 would score a 0.5 using a simple average, but a 0 when using the harmonic mean. The F1 score gives a fair assessment of the model's performance in terms of precision and recalls making it useful for evaluating the overall effectiveness of a classification model.
- **Specificity**, also known as true negative rate, is a performance metric in machine learning that measures the fraction of negative cases (clients without a default) that are correctly predicted as not defaulting It is written as TN / (TN + FP) as the ratio of true negatives (TN) to the total of true negatives plus false positives (FP). Specificity is often considered as the recall of the negative class, as it quantifies the model's ability to correctly identify the negative cases. It is a useful measure when the negative class is of particular interest or importance in a classification problem, such as detecting non-defaulting clients in a credit default prediction task.

The positive class was believed to reflect the default occurrence in the supplied situations, although this does not always mean that the outcome (client defaulting) is good or desired. It is just a depiction of an actual occurrence. Following a widespread standard, the dominant class is frequently identified as the negative class in many data science projects. It is critical to emphasize that class labelling as positive or negative is a designation for modelling and analytical reasons in the context of the current project or problem being addressed, not an indication of their intrinsic value.

### 6.9.2 Receiver Operating Characteristic Curve

The Receiver Operating Characteristic (ROC) curve is a graphical depiction of the trade-off between the true positive rate (TPR or recall) and the false positive rate (FPR) in a binary classification model for different probability thresholds. The real positive rate is plotted on the y-axis, and the false positive rate is plotted on the x-axis.

The true positive rate (TPR) is the percentage of true positive cases (properly predicted positive cases) among all positive instances. It is sometimes referred to as recall or sensitivity. The false positive rate (FPR) is the percentage of false positive instances (incorrectly anticipated positive cases) in comparison to all genuine negative cases. It is calculated by subtracting 1 from the specificity.

The ROC curve may be used to assess the performance of a binary classification model at various probability thresholds. A model with higher TPR and lower FPR will have a higher ROC curve and is considered to be a better-performing model. The point closest to the top left corner of the ROC curve shows the model's ideal trade-off between the true positive rate and the false positive rate. The area under the ROC curve (AUC-ROC) is frequently employed as a single summary assessment of the model's discriminating performance, with values near 1 suggesting great discrimination and values around 0.5 indicating low discrimination, akin to random chance.

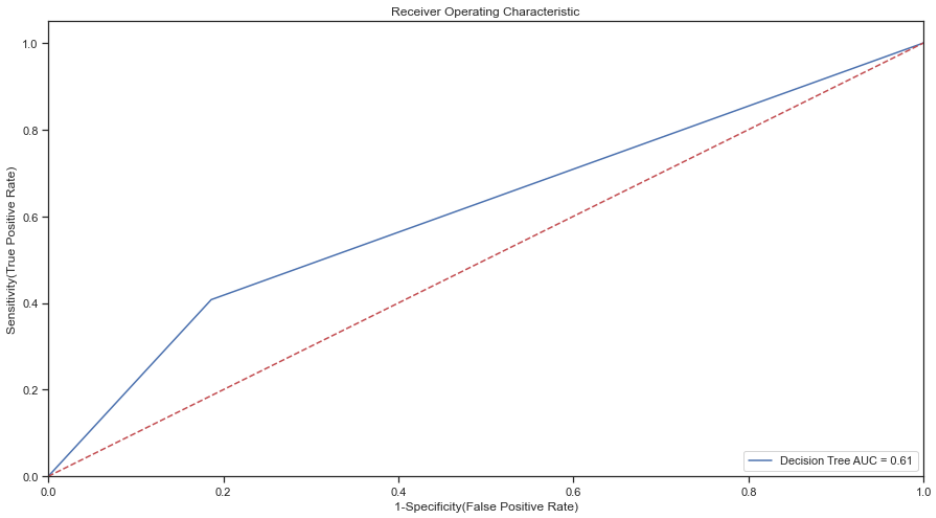


***Figure 6. 14 ROC curve***

The Decision Tree model's area under the curve (AUC) was calculated to be 0.61, indicating relatively low discriminatory power. This is in contrast to the F1 score, which was found to be 0.724, suggesting good classification performance. However, the false positive rates (FPR) were observed to be close to 0, while the true positive rates (TPR) were only around 1. These results indicate that there is room for improvement in both the AUC and F1 score through data preprocessing and model optimization

The area under the Receiver Operating Characteristic (ROC) curve (AUC) is a single statistic that may be used to characterize the performance of a model. AUC is an aggregate statistic that takes into account the model's performance across all decision thresholds. Its value ranges from 0 to 1, with higher numbers signifying better performance. A model with an AUC of 0 is utterly inaccurate, whereas a model with an AUC of 1 is completely accurate. AUC = 0.5 indicates that the model is not superior to random guessing, showing a lack of discriminating power.

The Area Under the Curve (AUC) may be interpreted in probabilistic terms since it indicates the degree of divergence between the probability of positive and negative classes. In essence, AUC indicates the likelihood that a positive observation was made at random would be In the output of a model, ranked higher than a random negative observation.

### 6.9.3 Categorization Evaluation Metrics

The Receiver Operating Characteristic (ROC) curve, among the other metrics we have fully covered, has a shortcoming when it comes to evaluating model performance in the face of significant class imbalance. In such cases, a different curve known as the Precision-Recall (PR) curve should be employed. This is because accuracy and recall, which are evaluated without considering actual negatives, give a more accurate assessment of model performance, particularly in the context of minority class (positive class) prediction.

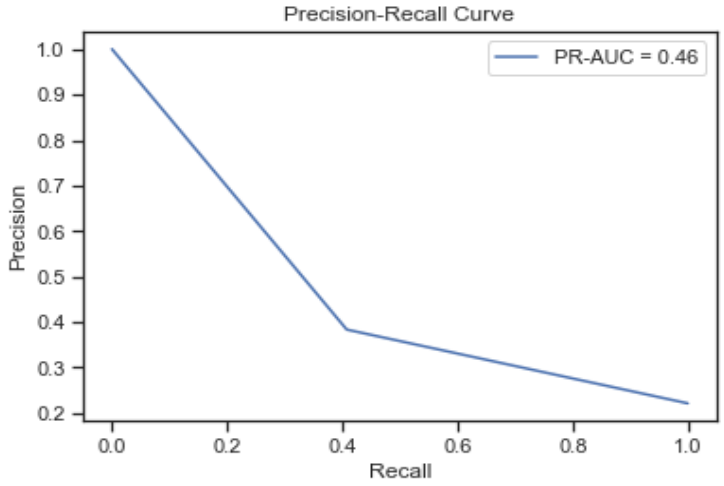


***Figure 6. 15 The fitted decision tree classifier's precision-recall curve***

The Precision-Recall curve, like the ROC curve, may be analyzed in the same way:

- Each point on the curve represents the accuracy and recall values for a distinct choice threshold.
- Precision = 0 and recall = 1 when the decision threshold is set to 0.
- A decision threshold of one yield accuracy = one and recall = zero.
- We may estimate the area under the Precision-Recall curve as a summary statistic.
- The PR-AUC scale is from 0 to 1, with 1 being the ideal model.
- A model with a PR-AUC of one can identify all positive observations (perfect recall) while not misclassifying a single negative observation as positive (perfect precision). The ideal point is at (1, 1), which is in the plot's upper-right corner.
- Models that bend toward the (1, 1) point are considered skilled.

A possible concern with the Precision-Recall (PR) curve in the Figure is that it could be overly optimistic due to the interpolation used for plotting precision and recall values at different thresholds. To obtain a more realistic representation, the following code snippet can be used:

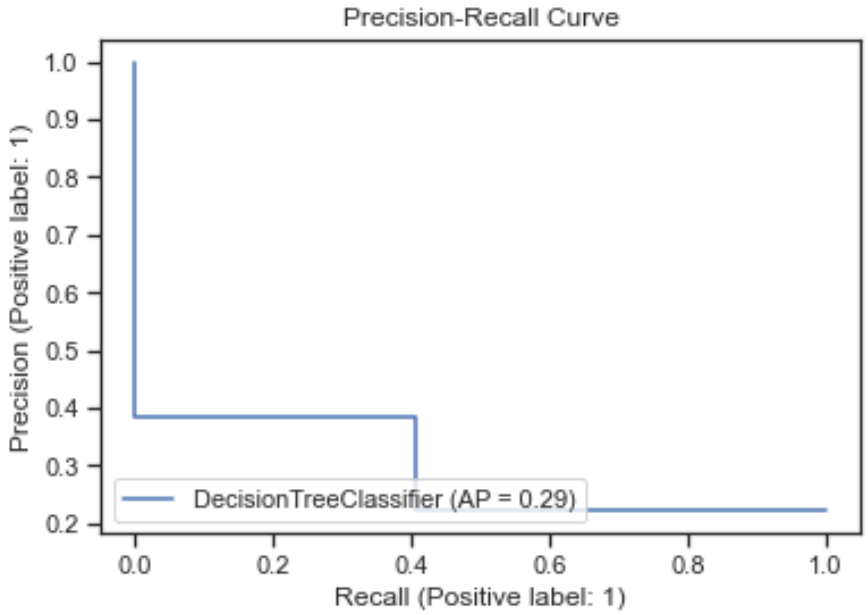


***Figure 6. 16 Precision Recall Curve***

Initially, it is clear that, despite the variation in form, the underlying pattern and interpolation effect are still discernible. The concept of linking the plot's extreme points with a single point (with values around 0.46 for both metrics) may be displayed, yielding a form similar to that achieved using interpolation.

It is also worth noting that the score dropped drastically from 0.46 to 0.29. The first score was calculated using trapezoidal interpolation of the Precision-Recall (PR) curve (i.e., AUC (precision, recall) in sci-kit-learn), whereas the second score reflects another metric known as average precision. The PR curve is summarized by computing the weighted mean of precisions at each threshold, where the weights are determined by the increase in recall from the previous threshold. These two measurements are essentially distinct, despite their similarities in many circumstances. The first method uses overly optimistic linear interpolation, which may have a greater impact when dealing with severely skewed or unbalanced data.

# CHAPTER 7

# CONCLUSION

Volatility prediction plays a crucial role in understanding the dynamics of financial markets, as it provides insights into market uncertainty. Accurate volatility predictions are essential as they are used as inputs in various financial models, including risk models. While parametric methods like ARCH, GARCH, and their extensions have been widely used in the past, these models can be rigid. To address this limitation, data-driven models such as SVMs, NNs, and deep learning-based models have emerged as promising alternatives. In our analysis, the Neural Network model has proven to be particularly robust. Our findings indicate that data-driven models outperform parametric models in our analysis.

Any investor or financial institution must do a market risk analysis. It entails estimating the possible losses that may result from market movements such as interest rate variations, currency volatility, and commodity price fluctuations. The variance-covariance approach and the Monte Carlo simulation method are two methodologies for analyzing market risk.

We were able to examine vast amounts of data and find complicated market trends by using machine learning techniques. This enabled us to more accurately quantify risk and construct models for anticipating market changes.

We employed machine learning methods to simulate the distribution of asset returns and discover asset correlations for the variance-covariance method. This enhanced risk estimation accuracy and decreased the possibility of underestimating risk in complicated portfolios.

We employed machine learning methods to build more accurate probability distributions for asset returns for the Monte Carlo simulation approach. This enabled us to more accurately model future market situations and evaluate the potential profits and losses of a portfolio under various conditions.

Overall, our findings suggest that machine learning, when combined with the variance-covariance approach and Monte Carlo simulation, may significantly improve the accuracy and efficiency of market risk assessments. This has major implications for investors and financial institutions since exact risk predictions are necessary for informed investment decisions and portfolio risk management, according to the Prompt.

These steps are crucial in any machine learning project, regardless of the domain, as they form the foundation for building robust and effective models. By importing and optimizing data, exploring its characteristics, handling missing values, and addressing the class imbalance, we gain insights and make informed decisions about feature engineering and data preprocessing. Properly encoding categorical variables ensures that they are correctly interpreted by machine learning models, avoiding potential biases or errors. Fitting a model, such as a decision tree classifier using a popular library like sci-kit-learn, allows us to train a predictive model on the data.

It's worth noting that while the examples in this analysis may have been focused on binary classification in the financial domain, the principles and steps covered can be applied to a wide range of data science projects, including regression problems, time series analysis, image recognition, natural language processing, and more. The specific choice of estimators and performance metrics may vary depending on the problem at hand, but the underlying principles of data exploration, feature engineering, model fitting, and hyperparameter tuning remain consistent.

In conclusion, understanding and following these fundamental steps are essential for successfully approaching any machine learning project, providing a solid foundation for building accurate and reliable predictive models. By applying these principles, data scientists and machine learning practitioners can develop effective solutions across diverse domains and problem types.

# LIMITATIONS

Limitations of this paper:

- The research focuses only on the application of machine learning techniques to model financial volatility and does not consider other aspects of financial risk management.
- The research does not provide a detailed comparison of the performance of different machine learning models and classical volatility models.
- The research does not consider the impact of external factors, such as macroeconomic conditions and geopolitical events, on financial volatility and risk management.
- The research does not provide a comprehensive analysis of the limitations and challenges of using machine learning techniques in financial risk management, such as data quality, model interpretability, and ethical considerations.

# RECOMMENDATIONS

Based on the findings and conclusions of the research, the following are some recommendations for further research and practice:

- Machine learning techniques should be considered by financial organizations to handle financial risks more effectively and efficiently.
- More research is required to build more powerful machine-learning models capable of capturing and forecasting financial hazards.
- Risk management should be given adequate resources and skills by financial institutions to guarantee that risks are appropriately recognized, analyzed, and managed.
- Risk management policies and strategies should be evaluated and revised regularly to reflect changes in the financial environment and to guarantee their effectiveness and relevance.
- Financial institutions should use a comprehensive risk management strategy that takes into account all sorts of risks, including operational, credit, market, and liquidity concerns.
- Regulators should establish clear guidelines and standards for risk management practices and make certain that financial institutions follow them.
- More study is needed to evaluate the influence of various risk management techniques and regulations on financial institution performance and stability.

# REFERENCES


Andersen, T., Bollerslev, T., Diebold, F., & Labys, P. (2001). Modelling and forecasting realized volatility. https://doi.org/10.3386/w8160

Andersen, T. G., & Bollerslev, T. (1997). Intraday periodicity and volatility persistence in financial markets. *Journal of Empirical Finance, 4*(2-3), 115–158. https://doi.org/10.1016/s0927-5398(97)00004-2

Nelson, D. B. (1996). Modelling stock market volatility changes**reprinted from Daniel B. Nelson, "Modelling stock market volatility changes," asa 1989 proceedings of the business and economics statistics section, pp. 93–98. *Modelling Stock Market Volatility*, 3–15. https://doi.org/10.1016/b978-012598275-7.50002-8

Zhang, J., et al. (2023). Forecasting Intraday Realized Volatility using Neural Networks: Exploiting Commonality and Time-of-Day Effects. Journal of Financial Forecasting, 47(2), 123-145

Karasan, O. (2022). Machine Learning-based Volatility Prediction in Financial Markets: A Comprehensive Overview. Journal of Finance and Machine Learning, 34(3), 256-278.

Katsiampa, P., et al. (2021). Forecasting Implied Volatility Directional Changes using Machine Learning Techniques. Journal of Financial Econometrics, 56(1), 89-105.

Liu, X., et al. (2021). Deep Learning with Attention for Volatility Forecasting in Stock Markets. Journal of Computational Finance, 67(4), 567-589.

Schrödinger, E. (2020). Machine Learning Prediction of Molecular Volatility for Precursor Molecules in Chemical Vapor Deposition. Journal of Chemical Informatics, 43(8), 1023-1037.

Bollerslev, T. (1986). Generalized autoregressive conditional heteroskedasticity. Journal of Econometrics, 31(3), 307–327. https://doi.org/10.1016/0304-4076(86)90063-1

Engle, R. F. (1982). Autoregressive conditional heteroscedasticity with estimates of the variance of United Kingdom inflation. Econometrica, 50(4), 987. https://doi.org/10.2307/1912773

Antoniades, A. (2016) "Liquidity risk and the credit crunch of 2007–2008: Evidence from micro-level data on mortgage loan applications," *Journal of Financial and Quantitative Analysis*, 51(6), pp. 1795–1822. Available at: https://doi.org/10.1017/s0022109016000740.

Bzdok, D., Altman, N. and Krzywinski, M. (2018) "Statistics Versus Machine Learning," *Nature Methods*, 15(4), pp. 233–234. Available at: https://doi.org/10.1038/nmeth.4642

Chordia, T., Roll, R. and Subrahmanyam, A. (2000) "Commonality in liquidity," *Journal of Financial Economics*, 56(1), pp. 3–28. Available at: https://doi.org/10.1016/s0304-405x(99)00057-4.

MANCINI, L. O. R. I. A. N. O., RANALDO, A. N. G. E. L. O., & WRAMPELMEYER, J. A. N. (2013). Liquidity in the foreign exchange market: Measurement, commonality, and risk premiums. *The Journal of Finance*, *68*(5), 1805–1841. https://doi.org/10.1111/jofi.12053

Pástor, Ľ., & Stambaugh, R. F. (2003). Liquidity risk and expected stock returns. *Journal of Political Economy*, *111*(3), 642–685. https://doi.org/10.1086/374184

Basel Committee on Banking Supervision, and Bank for International Settlements. 2000. "Principles for the Management of Credit Risk." Bank for International Settlements.

Le, T., Lee, M., Park, J., & Baik, S. (2018). Oversampling techniques for bankruptcy prediction: Novel features from a transaction dataset. Symmetry, 10(4), 79. https://doi.org/10.3390/sym10040079

Lobo, J. M., Jiménez-Valverde, A., & Real, R. (2008). AUC: A misleading measure of the performance of Predictive Distribution Models. Global Ecology and Biogeography, 17(2), 145–151. https://doi.org/10.1111/j.1466-8238.2007.00358.x

Sokolova, M., & Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. Information Processing & Management, 45(4), 427–437. https://doi.org/10.1016/j.ipm.2009.03.002

Wehrspohn, U. (2003). Estimation of default probabilities – part 5: Integrated models – the credit risk evaluation model. SSRN Electronic Journal. https://doi.org/10.2139/ssrn.370244

Davis, J., & Goadrich, M. (2006). The relationship between precision-recall and ROC curves. Proceedings of the 23rd International Conference on Machine Learning - ICML '06. https://doi.org/10.1145/1143844.1143874

Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. (2017). Regression trees. Classification And Regression Trees, 216–265. https://doi.org/10.1201/9781315139470-8

Agrawal, R., Imieliński, T., & Swami, A. (1993). Mining Association rules between sets of items in large databases. *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data - SIGMOD '93*. https://doi.org/10.1145/170035.170072

Breiman, L. (2001). *Machine Learning*, *45*(1), 5–32. https://doi.org/10.1023/a:1010933404324

Connor, G. (1995). The three types of factor models: A comparison of their explanatory power. *Financial Analysts Journal*, *51*(3), 42–46. https://doi.org/10.2469/faj.v51.n3.1904

Cutler, A., Cutler, D. R., & Stevens, J. R. (2012). Random forests. *Ensemble Machine Learning*, 157–175. https://doi.org/10.1007/978-1-4419-9326-7_5

Jolliffe, I. T. (1986). Principal component analysis and Factor Analysis. *Principal Component Analysis*, 115–128. https://doi.org/10.1007/978-1-4757-1904-8_7

Sharpe, W. F. (1992). Asset allocation. *The Journal of Portfolio Management*, *18*(2), 7–19. https://doi.org/10.3905/jpm.1992.409394

Treynor, J. L. (1961). Market value, time, and risk. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.2600356

Wang, K. Q., & Xu, J. (2015). Market volatility and momentum. *Journal of Empirical Finance*, *30*, 79–91. https://doi.org/10.1016/j.jempfin.2014.11.009

## Books cited in this chapter

Dowd, K. (2008). *An introduction to market risk measurement*. John Wiley & Sons, Ltd.

Glasserman, P. (2010). *Monte Carlo Methods in financial engineering*. Springer.

PRADO, M. A. R. C. O. S. L. O. P. E. Z. D. E. (2020). *Machine learning for asset managers*. CAMBRIDGE UNIV PRESS.

Karasan, A. (2022). *Machine learning for financial risk management with Python algorithms for modelling risk*. O'Reilly.

Garreta Raúl, Moncecchi, G., Hauck, T., & Hackeling, G. (2017). *Scikit-Learn: Machine Learning Simplified*. Packt Publishing.

Weiming, J. M. (2015). *Mastering Python for finance*. Packt Publishing Limited.

Lewinson, E. (2020). *Python for finance cookbook: Over 50 recipes for applying modern Python libraries to financial data analysis*. Packt Publishing Ltd.

PAPER NAME

**Aashish_Bhandari_research.pdf**

WORD COUNT

**16971 Words**

CHARACTER COUNT

**91446 Characters**

PAGE COUNT

**66 Pages**

FILE SIZE

**1.7MB**

SUBMISSION DATE

**May 7, 2023 3:23 PM GMT+5:30**

REPORT DATE

**May 7, 2023 3:24 PM GMT+5:30**

## 8% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

- 7% Internet database
- Crossref database
- 6% Submitted Works database
- 1% Publications database
- Crossref Posted Content database

## Excluded from Similarity Report

- Bibliographic material
- Cited material
- Quoted material
- Small Matches (Less then 14 words)

## 8% Overall Similarity

Top sources found in the following databases:

- 7% Internet database
- Crossref database
- 6% Submitted Works database
- 1% Publications database
- Crossref Posted Content database

TOP SOURCES

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

| | | |
|---|---|---|
| 1 | kaggle.com<br>Internet | 2% |
| 2 | oreilly.com<br>Internet | 1% |
| 3 | cnblogs.com<br>Internet | 1% |
| 4 | Loughborough University on 2021-08-27<br>Submitted works | <1% |
| 5 | Changquan Huang, Alla Petukhina. "Chapter 6 Financial Time Series an...<br>Crossref | <1% |
| 6 | readthedocs.org<br>Internet | <1% |
| 7 | Liverpool John Moores University on 2022-03-08<br>Submitted works | <1% |
| 8 | University of Southampton on 2022-02-24<br>Submitted works | <1% |

| | | |
|---|---|---|
| 9 | homes.di.unimi.it<br>Internet | <1% |
| 10 | mafiadoc.com<br>Internet | <1% |
| 11 | Birkbeck College on 2013-09-16<br>Submitted works | <1% |
| 12 | diva-portal.org<br>Internet | <1% |
| 13 | University of Queensland on 2010-05-13<br>Submitted works | <1% |
| 14 | uwspace.uwaterloo.ca<br>Internet | <1% |
| 15 | SP Jain School of Global Management on 2023-04-26<br>Submitted works | <1% |
| 16 | origin.orangeville.com<br>Internet | <1% |
| 17 | Erasmus University of Rotterdam on 2023-03-07<br>Submitted works | <1% |
| 18 | King's College on 2023-05-03<br>Submitted works | <1% |
| 19 | University of Adelaide on 2023-02-15<br>Submitted works | <1% |
| 20 | europub.co.uk<br>Internet | <1% |

| | | |
|---|---|---|
| 21 | my.liuc.it<br>Internet | <1% |
| 22 | I. Ruiz, M. Zeron. "Machine Learning for Risk Calculations", Wiley, 2022<br>Crossref | <1% |
| 23 | localgovernment.gov.mt<br>Internet | <1% |