Portfolio Optimization using Credit Risk Analysis

Indian Institute of Technology Ropar

B.Tech Project (CP-303)

Submitted by

Mohit Kumar (2020MCB1242)
Ankit (2020MCB1227)
Samual (2020MCB1249)

Under the Guidance of

Dr. Arun Kumar

9th May 2024

ACKNOWLEDGMENT

We have made efforts in this project. However, it would not have been possible without the kind support and help of many individuals. We would like to extend our sincere thanks to all of them. We are highly indebted to Dr. Arun Kumar for his guidance and constant supervision, for providing necessary information regarding the project, and for his support in completing the project. We would like to express our gratitude to the Department of Mathematics for their kind cooperation and encouragement, which has helped us complete this project. Our thanks and appreciation also go to our colleagues who helped develop the project and the people who have willingly helped us out with their abilities.

ABSTRACT

This report outlines a comprehensive analysis of credit risk and portfolio optimization using machine learning and neural network techniques. Through the utilization of various machine learning and neural network models, we conducted feature selection and compared model performances. Additionally, we explored portfolio optimization strategies, including Markowitz mean-variance optimization and the Black Litterman allocation model. The findings highlight the importance of features such as MScore and the effectiveness of Long Short Term Memory in modeling credit risk. Furthermore, the report presents a practical application of the optimization models through a user-friendly website interface.

Contents

CHAPTER 1

INTRODUCTION

1.1 CONTEXT

In the world of finance, understanding and managing risks is crucial. One significant risk that financial institutions face is known as "credit risk." Credit risk revolves around the uncertainty of whether borrowers will repay the money they borrowed. This uncertainty can have significant implications for banks, lenders, and borrowers alike. For lenders, accurately assessing credit risk is essential to ensure the soundness of their lending practices and the stability of their financial health. Understanding this context is vital as it sets the stage for exploring the methods and tools used to mitigate credit risk effectively.

1.2 MOTIVATION

The motivation behind our project is rooted in the desire to empower financial institutions and investors with tools and strategies to make informed decisions and mitigate risks effectively. By delving into the realm of credit risk analysis and portfolio optimization, we aim to contribute to the financial industry's ongoing quest for stability and growth. Improving our understanding of credit risk enables lenders to make more accurate lending decisions, safeguarding their assets and promoting responsible borrowing practices. Similarly, optimizing investment portfolios empowers investors to achieve their financial objectives while minimizing exposure to unnecessary risks. This motivation serves as the driving force behind our endeavor to develop innovative solutions to complex financial challenges.

1.3 GOALS

Our project encompasses several key goals aimed at addressing the challenges posed by credit risk and portfolio optimization:

- Advanced Analysis Techniques: Utilizing machine learning to delve deep into credit risk assessment. By leveraging these advanced analytical techniques, we aim to enhance the accuracy and reliability of credit risk evaluations, enabling lenders to make more informed decisions.
- Optimization Strategies: Exploring innovative strategies for optimizing investment portfolios to achieve optimal returns while mitigating risks. This involves delving into methodologies such as mean-variance optimization and the Black Litterman allocation model to develop robust portfolio management strategies.

- Accessible Solutions: Creating user-friendly platforms or tools to democratize access to sophisticated financial analysis techniques. By making these tools accessible and intuitive, we aim to empower a broader audience, including individuals and smaller financial institutions, to harness the power of advanced financial analytics for better decision-making.

Chapter 2

Credit Risk

Credit risk is a fundamental concept in the world of finance, serving as a crucial indicator of potential financial loss resulting from borrowers' failure to fulfill their repayment obligations. At its core, credit risk reflects the uncertainty associated with lending money, highlighting the possibility that borrowers may default on their loans, leading to financial repercussions for lenders and investors alike.

This risk extends its impact beyond financial institutions, affecting individuals and industries on various levels. For individuals, credit risk influences borrowing costs, interest rates, and access to credit, shaping their financial decisions and opportunities. In industries, credit risk can influence investment decisions, corporate financing strategies, and overall economic stability, impacting growth prospects and business operations.

Understanding and calculating credit risk is essential for several reasons. Firstly, it enables lenders to assess the creditworthiness of borrowers accurately, ensuring responsible lending practices and safeguarding financial stability. Additionally, effective credit risk analysis provides insights into potential losses, guiding risk management strategies and portfolio diversification efforts. Moreover, in the context of investment management, evaluating credit risk informs asset allocation decisions, helping investors optimize their portfolios and achieve their financial objectives.

Traditionally, credit risk assessment relied on fundamental financial ratios and qualitative evaluations to gauge borrowers' creditworthiness. However, with advancements in technology and data analytics, particularly in the field of machine learning (ML), there has been a paradigm shift towards data-driven approaches. These approaches leverage historical data, complex algorithms, and predictive modeling techniques to forecast credit risk with greater accuracy and efficiency.

In our project, we leverage various ML models to predict credit risks and assess their performance in accurately identifying potential default risks. By harnessing the power of ML algorithms, we aim to enhance the effectiveness of credit risk assessment, ultimately contributing to more informed decision-making processes and robust portfolio management practices in the financial domain.

Chapter 3

Machine Learning Models

3.1 DataSet and Feature Selection

We took a dataset of different companies and their turnovers, EBITS, PLtaxs, MScores, ROEs, Leverages, and TAssets over the previous 4 years. It also includes Region, country, Sector1, Sector2, and NACE code. Now, we want to understand which features are most helpful in improving predictive accuracy. We used Logistic regression on many combinations of the above features to find which can predict credit risk most accurately. MScore 2018 predicted MScore 2019 using logistic regression with 79.4% accuracy, and other combinations of features did not improve the accuracy. We found that MScore is the best feature for predicting credit risk. MScore of all past years predicted MScore 2019 using logistic regression with 82.4% accuracy. Then, we used MScore from 2015 to 2018 to predict the MScore of 2019 using different machine learning models such as Logistic Regression, Decision Tree, Random Forest, Support Vector Machine, Convolution Neural Network, and Long Short-Term Memory.

3.2 Logistic Regression

Logistic regression is a supervised machine learning algorithm that accomplishes binary classification tasks by predicting the probability of an outcome, event, or observation. The model delivers a binary outcome limited to two possible outcomes: yes/no, 0/1, or true/false.

The logistic regression formula is given by:

$$y = \frac{e^{-(\beta_0+\beta_1 x_1)}}{1+e^{-(\beta_0+\beta_1 x_1)}} \tag{3.1}$$

Where:

- x is input value.
- y is the predicted output.
- β_0 is bias or intercept term.
- β_1 is coefficient for input (x).

Complexity Analysis of Logistic Regression

- Train time complexity: $O(n \cdot d)$

- Train space complexity: $O(n \cdot d)$
- Test time complexity: $O(d)$
- Test space complexity: $O(d)$
- d: number of features
- n: number of data points

MODEL PERFORMANCE OF LOGISTIC REGRESSION

- Logistic Regression delivered consistent performance with a training accuracy of 81.5% and a test accuracy of 82.4%.
- While demonstrating strong precision of 89%, the model exhibits a higher recall of 73% and F1-score of 81%.

3.3 DECISION TREE

A decision tree is a simple yet powerful supervised learning algorithm used for both classification and regression tasks. It has a hierarchical tree structure, which consists of a root node, branches, internal nodes, and leaf nodes. It works by recursively partitioning the feature space into regions that are homogeneous with respect to the target variable. Each internal node of the tree represents a decision based on a feature, leading to successive splits until reaching leaf nodes with final predictions.

The Decision formula is given by:

$$G = 1 - \sum_{i=1}^{c} p_i^2 \tag{3.2}$$

where:

G is the Gini index,
c is the number of classes,
p_i is the probability of an instance belonging to class i.

COMPLEXITY ANALYSIS OF DECISION TREE

- Train time complexity: $O(n \cdot logn \cdot d)$
- Train space complexity: $O(nodes)$
- Test time complexity: $O(depth)$
- Test space complexity: $O(nodes)$
- d: number of features
- n: number of data points

MODEL PERFORMANCE OF DECISION TREE

- The decision tree classifier achieved a training accuracy of 81.8% and a test accuracy of 82.2%.
- While performing well of precision: 87%, recall: 75%, F1-score: 81%.

3.4 RANDOM FOREST

Random forest is a machine learning algorithm that creates an ensemble of multiple decision trees to reach a singular, more accurate prediction or result. Each tree is constructed using a random subset of the data set to measure a random subset of features in each partition. This randomness introduces variability among individual trees, reducing the risk of overfitting and improving overall prediction performance. In prediction, the algorithm aggregates the results of all trees, either by voting (for classification tasks) or by averaging (for regression tasks). This collaborative decision-making process, supported by multiple trees with their insights, provides an example of stable and precise results.

COMPLEXITY ANALYSIS OF RANDOM FOREST

- Train time complexity: $O(n \cdot logn \cdot d \cdot k)$
- Train space complexity: $O(n \cdot k)$
- Test time complexity: $O(depth \cdot k)$
- Test space complexity: $O(depth \cdot k)$
- d: number of features
- n: number of data points
- k: number of decision trees

MODEL PERFORMANCE OF RANDOM FOREST

- The Random Forest Classifier delivered an impressive performance, achieving training and test accuracies of 81.8% and 82.2%, respectively.
- The model has performed well for precision: 88%, recall: 75%, F1-score: 81%.

3.5 SUPPORT VECTOR MACHINE

Support Vector Machine (SVM) is a supervised machine learning algorithm used for both classification and regression. Though we say regression problems are best suited for classification, The main objective of the SVM algorithm is to find the optimal hyperplane in an N-dimensional space that can separate the data points in different classes in the feature space. The hyperplane tries that the margin between the closest points of different classes should be as maximum as possible.

The equation for the linear hyperplane can be written as:

$$\vec{w} \cdot \vec{x} + b = 0$$

where $\vec{w}$ is the weight vector, $\vec{x}$ is the input vector, and b is the bias

The optimization problem for the hard margin linear SVM classifier can be formulated as follows:

$$\text{minimize} \quad \frac{1}{2}||\vec{w}||^2$$

subject to the constraints:

$$y_i(\vec{w} \cdot \vec{x}_i + b) \geq 1 \quad \text{for } i = 1, 2, ..., n$$

where $\vec{w}$ is the weight vector, b is the bias term, and $(\vec{x}_i, y_i)$ are the training samples with $\vec{x}_i$ as input and y_i as their corresponding class labels.

The optimization problem for the soft margin linear SVM classifier can be formulated as follows:

$$\text{minimize} \quad \frac{1}{2}||\vec{w}||^2 + C\sum_{i=1}^{n}\xi_i$$

subject to the constraints:

$$y_i(\vec{w} \cdot \vec{x}_i + b) \geq 1 - \xi_i \quad \text{for } i = 1, 2, ..., n$$

$$\xi_i \geq 0 \quad \text{for } i = 1, 2, ..., n$$

where $\vec{w}$ is the weight vector, b is the bias term, C is the regularization parameter, and $(\vec{x}_i, y_i)$ are the training samples with $\vec{x}_i$ as input and y_i as their corresponding class labels. ξ_i are slack variables.

COMPLEXITY ANALYSIS OF SUPPORT VECTOR MACHINE

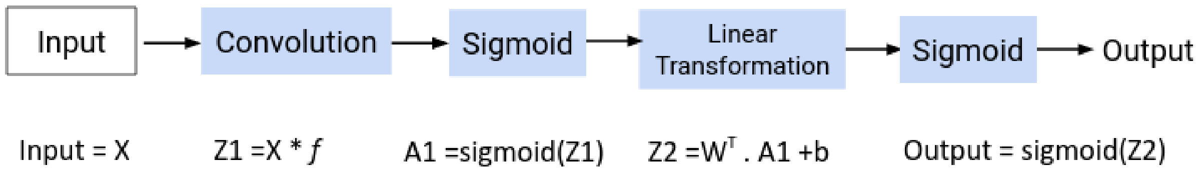
- Train Time Complexity: $O(n \cdot d^2)$
- Train Space Complexity: $O(n \cdot d)$
- Test Time Complexity: $O(n \cdot k)$
- Test Space Complexity: $O(n \cdot d)$
- d: number of features
- n: number of data points
- k: number of support vectors

MODEL PERFORMANCE OF SUPPORT VECTOR MACHINE

- The Support Vector Classifier stood out with impressive training and test accuracies of 81.8% and 82.2%, respectively.
- SVM model demonstrated a strong overall performance with precision: 87%, recall: 76%, F1-score: 81%.

3.6 CONVOLUTION NEURAL NETWORK

Convolutional Neural Network is the extended version of artificial neural networks, which is predominantly used to extract the feature from the grid-like matrix dataset. It is a specialized type of deep learning algorithm mainly designed for tasks that necessitate object recognition, including image classification, detection, and segmentation. It is made up of multiple layers, including convolutional layers, pooling layers, and fully connected layers. The Convolutional layer applies filters to the input image to extract features, the Pooling layer downsamples the image to reduce computation, and the fully connected layer makes the final prediction. The network learns the optimal filters through backpropagation and gradient descent.



MODEL PERFORMANCE OF CONVOLUTION NEURAL NETWORK

- The CNN stood out with impressive training and test accuracies of 83% and 84.1%, respectively.
- CNN model demonstrated a strong overall performance with precision: 89%, recall: 76%, F1-score: 81%.

3.7 LONG SHORT-TERM MEMORY

Long Short-Term Memory Networks is a deep learning, sequential neural network that allows information to persist. It is a special type of Recurrent Neural Network(RNN) that is capable of handling the vanishing gradient problem faced by RNN. LSTMs address this problem by introducing a memory cell, which is a container that can hold information for an extended period.

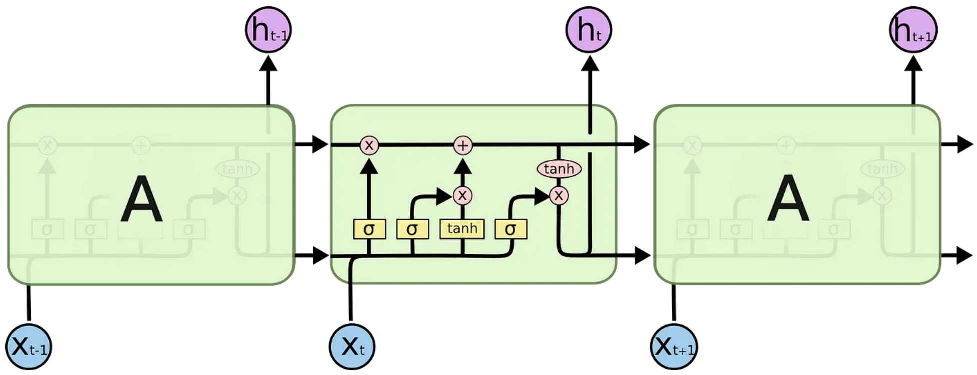


FIGURE 3.1
LSTM Model. Source

$$
\begin{aligned}
f_t &= \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)\\
i_t &= \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)\\
\tilde{C}_t &= \tanh(W_c \cdot [h_{t-1}, x_t] + b_c)\\
C_t &= f_t * C_{t-1} + i_t * \tilde{C}_t\\
o_t &= \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)\\
h_t &= o_t * \tanh(C_t)
\end{aligned}
$$

where:

f_t is the forget gate,
σ is the sigmoid activation function,
x_t is the input at time step t,
h_{t-1} is the previous hidden state,
i_t is the input gate,
$\tilde{C}_t$ is the cell state,
C_t is the new cell state,
o_t is the output gate,
h_t is the new hidden state,

MODEL PERFORMANCE OF LONG SHORT-TERM MEMORY MODEL

- The LSTM stood out with impressive training and test accuracies of 83.6% and 84.6%, respectively.
- LSTM model demonstrated a strong overall performance with precision: 90%, recall: 76%, F1-score: 82%.

That's mainly because RNN has less feature compatibility and the ability to take arbitrary output/input lengths, which can affect the total computational time and efficiency. On the other hand, CNN takes fixed input and gives a fixed output, which allows it to compute the results at a faster pace.

CHAPTER 4

STOCKS ANALYSIS

4.1 RETURNS

DAILY RETURNS

Daily returns are relevant for investors because they provide a quick way to check the performance of a stock over a short period.

CUMULATIVE RETURNS

Cumulative returns are calculated by taking the difference between the Stock's final price and Initial Price for a specified period, adding any dividends or other income received, and dividing the result by the initial price.
It's important to note that cumulative return takes into account the effects of compounding, meaning that any gains from a previous period are reinvested and contribute to additional gains in future periods, which can result in a larger cumulative return than the simple average of the individual returns over the specified period.

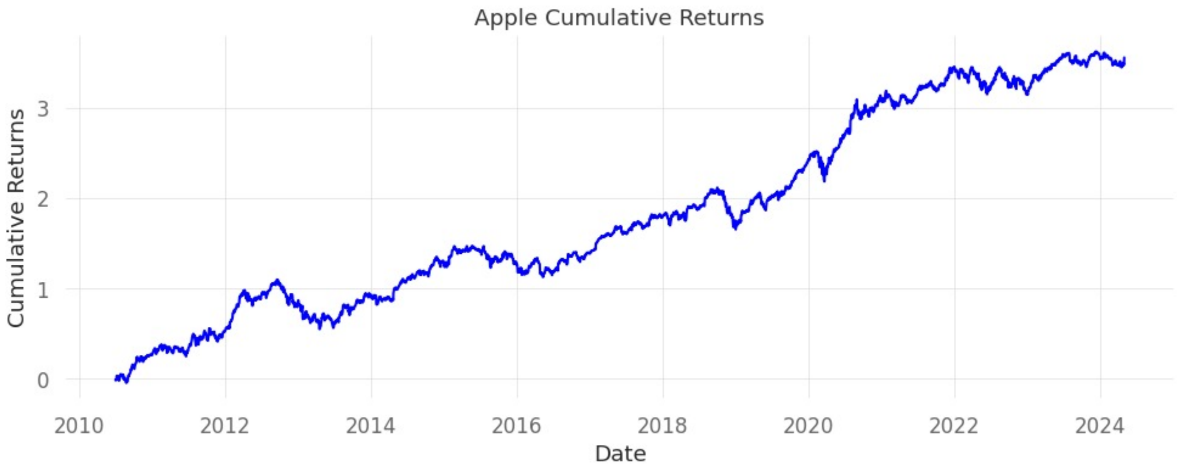


FIGURE 4.1
Returns of Apple over the period of 2010-2024

HISTOGRAM

Histograms are a graphical representation of the distribution of values, displaying how frequent they are in a dataset. It helps us to identify patterns, such as the range of daily returns of an asset over a certain period, indicating its level of stability and volatility.
Through the analysis of the histograms, we can observe that most daily returns are close to zero in the center of the distribution. However, it's easy to see some extreme values that are distant from the mean, which is the case of AMD, with daily returns of around 50%, indicating the presence of outliers in the positive range of the distribution, in contrast with the negative field where it seems to limit at about -20%.

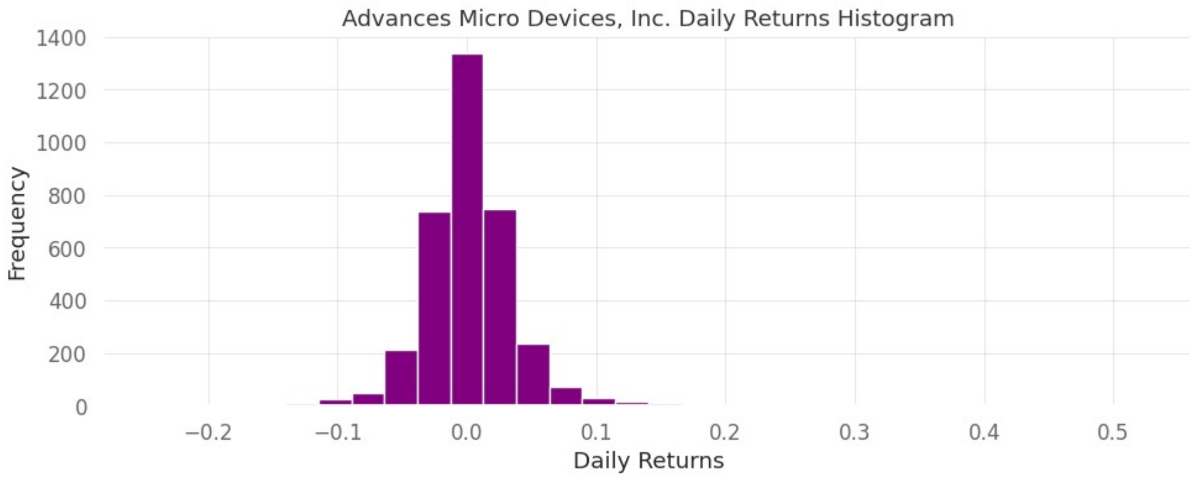


FIGURE 4.2
AMD Histogram

4.2 KURTOSIS

A high kurtosis value for daily returns may indicate frequent fluctuations in price that deviate significantly from the average returns of that investment, which can lead to increased volatility and risk associated with the stock.

$$\text{Kurtosis} = \frac{\sum_{i=1}^{n} \left(\frac{x_i - \bar{x}}{\sigma}\right)^4}{n} \tag{4.1}$$

The Kurtosis values we got for the stocks are:

Apple	5.34
Tesla	0.4.91
Google	8.592
Meta	21.1
AMD	16.587

TABLE 4.1
Kurtosis Value

4.3 SKEWNESS

Skewness is a metric that quantifies the asymmetry of returns. It reflects the shape of the distribution and determines if it is symmetrical, skewed to the left, or skewed to the right.

$$\text{Skewness} = \frac{\sum_{i=1}^{n} \left(\frac{x_i - \bar{x}}{\sigma}\right)^3}{n} \tag{4.2}$$

The skewness value we got for the stocks are:

Apple	-0.05
Tesla	0.33
Google	0.44
Meta	0.412
AMD	1.003

TABLE 4.2
Skewness

4.4 STANDARD DEVIATION

A stock exhibiting high daily return volatility, characterized by a high standard deviation, is considered riskier when compared to one with low daily return volatility, represented by a low standard deviation.
The Standard Deviation values we got for the stocks are:

Apple	0.018
Tesla	0.036
Google	0.017
Meta	0.025
AMD	0.036

TABLE 4.3
Standard Deviation from 2010 to 2024

4.5 CORRELATION MATRIX

Correlation analysis in the stock market allows us to interesting investment strategies like Long-Short, which is the act of buying shares of a company while selling shares of another company, believing that both assets will have opposite directions in the market. Correlation analysis is also crucial to avoid systemic risk, which is described as the risk of the breakdown of an entire system rather than simply the failure of individual parts.
We can see a significant correlation between Apple and Google Stocks.

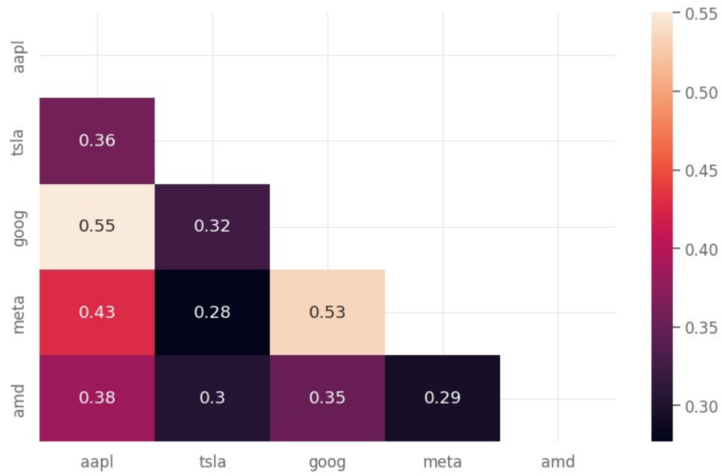


FIGURE 4.3
Correlation Matrix

4.6 SHARPE RATIO

The Sharpe ratio is a measure of the risk-adjusted return of an investment. It is calculated by dividing the average excess return of the investment over the standard deviation of the returns, as shown by the following equation:

$$\text{Sharpe Ratio} = \frac{r_p - r_f}{\sigma_p}$$

where:

$$r_p : \text{Portfolio return}$$
$$r_f : \text{Risk-free rate}$$
$$\sigma_p : \text{Portfolio standard deviation}$$

The Sharpe Ratio values we got for the stocks are:

Apple	0.92
Tesla	0.89
Google	0.86
Meta	0.71
AMD	0.67

TABLE 4.4
Sharpe Ratio

A higher Sharpe ratio indicates that an investment provides higher returns for a given level of risk. Apple and Tesla have the highest Sharpe ratios among the stocks analyzed, 0.92 and 0.89, respectively, indicating that these investments offer a better risk-return relationship.

Chapter 5

Portfolio Optimization

A portfolio in financial markets is a collection of financial assets. The weights in a portfolio refer to the percentage of the total value allocated to each individual asset. Allocating weights is a critical aspect of portfolio building because it determines the level of risk and return characteristics of the portfolio. A report is very useful for comparing the portfolio's performance and its level of risk with a benchmark, which, in our case, is the SP500.

5.1 Optimizing Portfolio

- Portfolio optimization is the process of selecting the optimal combination of assets and weights to maximize returns and minimize risk. This process involves selecting the most appropriate weights for each asset by taking into account the historical performance of the assets, their correlations with each other, and other relevant factors such as market conditions and economic outlook.

5.1.1 Markowitz mean-variance optimization model

- It is a widely-used framework for constructing portfolios with the best risk-return relationship. It is based on the idea that investors should maximize the expected return of a portfolio while minimizing its risk.
- There are two key requirements for mean-variance optimization:
  - First, we need to have expected returns for each of the assets in the portfolio. We calculated the expected returns for the assets by computing the arithmetic mean of their daily percentage changes.
  - Secondly, we need to choose a risk model that quantifies the level of risk in each asset. The most commonly used risk model is the covariance matrix, which describes the volatilities of assets and the degree to which they are co-dependent.

-	Apple	Tesla	Google	Meta	AMD
Apple	0.078242	0.056900	0.042235	0.048849	0.060719
Tesla	0.056900	0.323816	0.050379	0.063244	0.097666
Google	0.042235	0.050379	0.075212	0.058964	0.054299
Meta	0.048849	0.063244	0.058964	0.163034	0.068253
AMD	0.060719	0.097666	0.054299	0.068253	0.317881

TABLE 5.1
Covariance Matrix

- We take the covariance matrix and expected returns as inputs. The weights variable stores the optimized weights for each asset based on the maximization of the Sharpe ratio.
- The Percentage of the total value for each company is as follows:

Apple	0.52349
Tesla	0.18953
Google	0.27286
Meta	0.01412
AMD	0.0

TABLE 5.2

- Based on the report with new weights, the optimized portfolio appears to have performed better than the original portfolio in which the weights were equal.

5.1.2 BLACK LITTERMAN ALLOCATION MODEL

- It takes a Bayesian approach to asset allocation and combines a prior estimate of returns with the investor's particular views on the expected returns to generate an optimal allocation.
- The Black-Litterman formula calculates a weighted average between the prior estimate of returns and the views, with the weighting determined by the level of confidence for each view.
- After taking the prior expected returns and providing our views, as well as our confidence levels, we have an optimized portfolio with the following weights.

Apple	0.07798
Tesla	0.0
Google	0.45696
Meta	0.38499
AMD	0.08007

TABLE 5.3

- The result shows that our optimized portfolio outperformed in several key metrics, such as higher cumulative return and Sharpe, as well as lower annual volatility.
- Both the Markowitz Mean-Variance and Black-Litterman Allocation Models effectively enhanced the performance and reduced the risk of our original portfolio by optimizing the allocation weights of the stocks.

CHAPTER 6

RESULTS

- Our analysis revealed that out of all the the features of data, MScore emerged as the most influential feature in credit risk assessment. All Mscores of previous years predict credit risk with the most accuracy.
- Long Short-Term Memory demonstrated superior performance among all the learning models evaluated. Support Vector Machine performed best in the machine learning models.

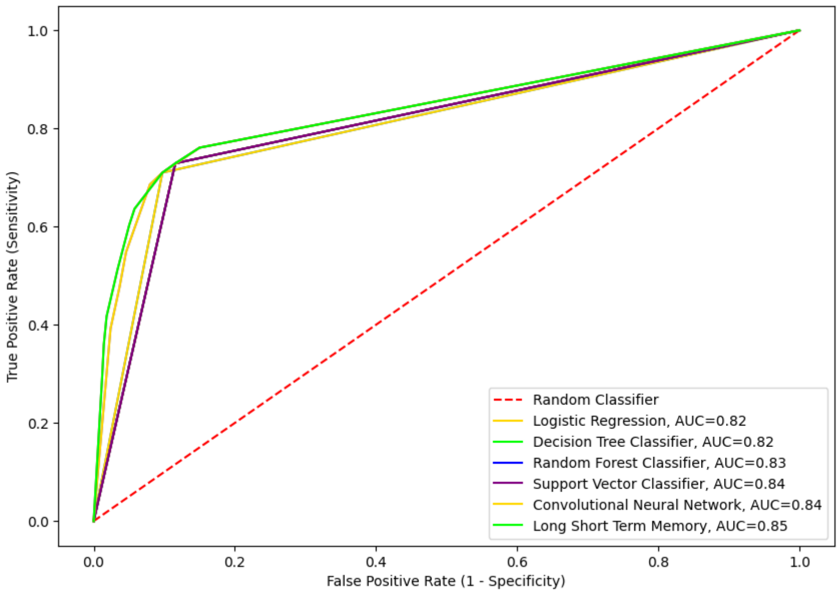


FIGURE 6.1
Area Under Curve of Different ML Models

- Portfolio optimization using Markowitz mean-variance optimization and the Black Litterman allocation model yielded promising results, enabling effective allocation of assets as compared to equally weighted portfolios. Markowitz's mean-variance optimization model uses stats like historical data and covariance matrix to determine the optimized weights of the stocks, whereas the Black Litterman Allocation Model is more dependent on individual views, current market situations, and the level of confidence.

Chapter 7

Conclusion

The integration of machine learning techniques with conventional quantitative finance methods for portfolio optimization has advanced significantly with this study. Through the utilization of sophisticated predictive algorithms and historical data, the study has illustrated the effectiveness of machine learning techniques in predicting credit risk and refining asset allocation plans. The results emphasize how crucial careful feature selection and model assessment are to getting precise predictions. Furthermore, the suggested methodologies' practical applicability in finance professionals' decision-making processes is highlighted by the validation of those approaches using actual portfolio data. In order to improve predictive accuracy, future research could concentrate on improving modeling techniques, investigating ensemble methods, and adding more data sources. In the end, practitioners can harness the power of machine learning to navigate the complexity of contemporary financial systems by adopting data-driven approaches.

Chapter 8

Future Work

While the current study has made significant strides in optimizing portfolios using credit risk, there are several avenues for future research and development that could enhance the effectiveness and scope of the project. Some potential areas for further exploration include:

Incorporating Sentiment Analysis

Expanding the optimization framework to incorporate sentiment analysis could provide valuable insights into market sentiment and its impact on portfolio performance. By integrating sentiment data from various sources, such as news articles, social media, and financial reports, we could develop more comprehensive models that capture both quantitative and qualitative factors influencing asset prices.

Enhancing Website Functionality and Visualization

Improving the visual appeal and functionality of the website can enhance user experience and engagement. Adding interactive features, customizable portfolio simulations, and real-time data updates could make the platform more dynamic and user-friendly. Additionally, integrating advanced visualization tools and dashboards can help users analyze portfolio performance and make informed investment decisions more effectively.

Deploying the portfolio optimization platform online would make it accessible to a broader audience of investors and financial professionals. We can ensure high availability and performance by hosting the website on a reliable and scalable cloud infrastructure, such as Amazon Web Services or Google Cloud Platform, allowing users to access the platform from anywhere with an internet connection.

Exploring Multi-Objective Optimization

Investigating multi-objective optimization techniques could enable us to optimize portfolios based on multiple criteria simultaneously, such as risk-adjusted returns, liquidity, and environmental, social, and governance (ESG) factors. By incorporating diverse investment objectives and constraints, we can develop more robust and personalized portfolio optimization solutions that align with investors' preferences and objectives.

References

- Isaac, P. (2022). Prediction of Credit Risk using Machine Learning Models. Uppsala University.
- Shi, S., Tse, R., Luo, W. et al. Machine learning-driven credit risk: a systemic review. Neural Comput & Applic 34, 14327–14339 (2022).
- Bessler, W., Opfer, H., & Wolff, D. (2017). Multi-asset portfolio optimization and out-of-sample performance: an evaluation of Black–Litterman, mean-variance, and naïve diversification approaches. The European Journal of Finance, 23(1).
- "Credit Risk Analysis with Machine Learning" (n.d.). Towards Data Science. Retrieved from Towards DataScience.
- Jorion, P. (1992). Portfolio Optimization in Practice. Financial Analysts Journal, 48(1), 68–74.