

AN ANALYSIS OF THE PERFORMANCE AND INTERPRETABILITY OF MACHINE LEARNING CLASSIFICATION ALGORITHMS TO PREDICT LONG-TERM SHARE RETURNS ON THE JSE

BY JAMIE BOAKES

SUBMITTED IN PARTIAL FULFILMENT FOR THE REQUIREMENTS FOR THE DEGREE

MSC INFORMATION TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE

UNIVERSITY OF CAPE TOWN



Supervised By:

Associate Professor Deshendran Moodley

May 2024

The copyright of this thesis vests in the author. No quotation from it or information derived from it is to be published without full acknowledgement of the source. The thesis is to be used for private study or non-commercial research purposes only.

Published by the University of Cape Town (UCT) in terms of the non-exclusive license granted to UCT by the author.

Declaration

I the undersigned, Jamie Edward Russell Boakes, hereby declare that the work on which this dissertation is based, is my original work, except where acknowledgements indicate otherwise, and that neither the whole work nor any part of it has been, is being, or is to be submitted for another degree in this or any other university.

Abstract

The prediction of long-term share returns is an essential yet complex task in financial analysis and formulating investment strategy. Machine learning is a promising approach for improving the accuracy of these predictions. However, the outputs of machine learning models are not transparent or interpretable, which limits their usability for real-world decision making. There is a lack of research on the use of machine learning algorithms to predict long-term share returns on the Johannesburg Stock Exchange (JSE), with no studies that specifically examine the interpretability of machine learning algorithms. This study investigates the use of machine learning algorithms to predict long-term share returns on the JSE based on fundamental data and analyses the interpretability of the top performing algorithms. Based on a review of the literature, eight machine learning classification algorithms were selected and compared to predict tercile class 12-month share returns using fundamental data, spanning a period of two decades. The XGBoost, Random Forest, and GradBoost algorithms were found to outperform the Support Vector Classifier, Logistic Regression, Decision Tree, Artificial Neural Network, and AdaBoost algorithms. XGBoost and Random Forest were further investigated using SHAP (SHapley Additive exPlanations) global summary plots to identify the most influential input features and to analyse the interpretability of these algorithms. The study found that ensemble-based classification algorithms, i.e. XGBoost, Random Forest and GradBoost, outperformed the other algorithms. Further analysis of the results varied, with some sectors outperforming the overall market. An analysis of the input features identified the most important valuation and profitability ratios that contributed to prediction performance, and thus improves the transparency and interpretability of the models. This research is the first to comprehensively compare and analyse the interpretability of machine learning algorithms to predict long-term share returns on the JSE.

Contents

Declaration.....	i
Abstract.....	ii
List of Tables	viii
List of Figures.....	viii
Glossary of Terms	xi
Acknowledgements.....	xiii
1. Introduction	1
1.1 Research Aim and Objectives	2
1.2 Tools and Approach	2
1.2.1 Overall Approach.....	2
1.2.2 Dataset	3
1.2.3 Algorithms	3
1.2.4 Evaluation	3
1.2.5 Tools	3
1.3 Contributions	3
1.4 Structure of the Dissertation.....	4
2. Literature Review.....	5
2.1 The Share Market: Johannesburg Stock Exchange (JSE)	5
2.2 The Efficient Market Hypothesis	5
2.3 Modern Portfolio Theory.....	6
2.4 Techniques for Share Prediction: Fundamental, Technical and Sentiment Analysis.....	6
2.4.1 Fundamental Analysis	7
2.4.2 Technical Analysis.....	7
2.4.3 Sentiment Analysis	8
2.5 Machine Learning for Predicting Share Returns	8
2.5.1 Problem Formulation	8
2.6 Classification Algorithms for Share Return or Price Trend Prediction.....	9
2.6.1 Datasets: Fundamental versus Technical.....	9
2.6.2 Feature Engineering.....	9
2.6.3 Prediction Window	9
2.6.4 Classification Classes or Rankings	10

2.6.5	Rolling versus Recursive Time Windows	10
2.7	Applying Machine Learning to Fundamental Data	11
2.7.1	Fundamental Data	11
2.7.2	Technical and Fundamental Data	11
2.7.3	Related Work in the South African Domain	12
2.7.4	Related Work Internationally	12
2.7.5	Research Trends	15
2.8	Machine Learning Algorithms	15
2.8.1	Random Forests	15
2.8.2	GradBoost	16
2.8.3	XGBoost	17
2.8.4	AdaBoost	17
2.8.5	Decision Trees	18
2.8.6	Support Vector Machines	19
2.8.7	Neural Networks	20
2.8.8	Logistic Regression	22
2.9	Evaluation	23
2.9.1	Classification Metrics	23
2.9.2	Return Analysis	23
2.9.3	Baseline Approaches	23
2.10	Explainable AI (XAI)	24
2.10.1	Background	24
2.10.2	Global versus Local Explanations	25
2.10.3	Model-Agnostic and Model-Specific Explanations	25
2.10.4	Popular Methods	25
2.10.5	SHAP	27
2.10.6	Explainable AI and Finance – A Survey of Approaches and Current Research	30
2.11	Summary Findings	32
3.	Experimental Design and Execution	34
3.1	Dataset	34
3.1.1	Scope	34

3.1.2	Overview.....	34
3.1.3	Limitations.....	41
3.1.4	Biases.....	42
3.2	Experimental Process	43
3.2.1	Classification Prediction Task, Machine Learning Problem and Pipeline	43
3.2.2	Experiments	44
3.2.3	Implementation.....	47
3.3	Evaluation.....	48
3.3.1.	Data Partitioning and Validation Approach	48
3.3.2	Evaluation Metrics.....	50
3.4	Data Preprocessing.....	53
3.4.1	Data Preprocessing Steps.....	53
3.4.2	Feature Engineering.....	54
3.4.3	Managing Outliers	54
3.4.4	Feature Scaling	54
3.4.5	Label Encoding.....	55
3.4.6	One-Hot Encoding.....	55
3.5	Algorithms.....	55
3.5.1	Random Forest.....	55
3.5.2	Gradient Boosting.....	56
3.5.3	XGBoost	57
3.5.4	AdaBoost	58
3.5.5	Decision Tree Classifier	59
3.5.6	Support Vector Machines	60
3.5.7	Artificial and Deep Neural Network using a Multilayer Perceptron.....	61
3.5.8	Logistic Regression	63
3.6	Summary.....	63
4.	Results and Analysis.....	65
4.1	Experiment 1 – Comparison of Machine Learning Algorithms	65
4.1.1	Algorithm Performance	65
4.1.2	Top Performing Algorithms	67

4.1.3	Key Findings.....	72
4.2	Experiment 2 - Sector and Share-level Performance Analysis	73
4.2.1	Sector-level Performance.....	73
4.2.2	Top Three Performing Sectors and Their Respective Shares	77
4.2.3	Share-level Performance Over Time	83
4.2.4	Key Findings.....	85
4.3	Experiment 3 - SHAP Analysis.....	86
4.3.1	Feature Importance using SHAP	86
4.3.2	Class Specific Feature Importance Results using SHAP.....	88
4.3.3	Detailed Analysis of Feature Importance Results using SHAP.....	94
4.3.4	Key Findings.....	95
4.4	Discussion.....	95
4.4.1	Overall Performance.....	95
4.4.2	Sector and Share-level Performance.....	96
4.4.3	SHAP Analysis	96
4.4.4	Comparison with Other Studies.....	97
4.5	Summary.....	99
5.	Conclusion and Future Work.....	101
5.1	Summary of Key Findings.....	101
5.2	Limitations.....	102
5.3	Future Work.....	103
6.	References	104
Appendix A		111
A1.	Chapter 2: Summary of Research - An Examination of Classification Algorithms Applied to Primarily Technical Data	111
A2.	Chapter 2: Machine Learning Algorithms	118
	Random Forest: Mathematical Representation	118
	Random Forest Classifier: Parameters	118
	GradBoost: Mathematical Representation	119
	GradBoost Classifier: Parameters	121
	XGBoost: Mathematical Representation.....	122
	XGBoost: Parameters.....	123

AdaBoost: Mathematical Representation	124
AdaBoost: Parameters	125
Decision Tree: Mathematical Representation	125
Decision Tree Classifier: Parameters	127
Support Vector Classifier: Mathematical Representation	128
Support Vector Classifier: Parameters	129
Multilayer Perceptron: Mathematical Representation	130
Keras Neural Network API: Parameters	131
Logistic Regression: Mathematical Representation	133
Logistic Regression: Parameters	134
Appendix B	136
B1.Chapter 3: Shares Listed on the FTSE/JSE ALSI	136
B2.Chapter 3: Initial Data Heatmap	144
B3.Chapter 3: Intermediate Data Heatmap Excluding Rows with 20% Nulls	145
B4.Chapter 3: Intermediate Data Heatmap Excluding Columns with 10% Nulls	146
B5.Chapter 3: Final Data Heatmap	147
B6.Chapter 3: Features Before and After adjusting for Outliers using the clipRatios Function	148
B7.Chapter 3: Features Before and After Transformations adjusting for Standard Scaling	154

List of Tables

Table 2-1 - Summary of Research – An Examination of Classification Algorithms Applied to Primarily Fundamental Data	14
Table 2-2 - SHAP Formula Variables.....	28
Table 2-3 - Explainable AI and Financial Prediction.....	31
Table 3-1- Data Elements and Detailed Descriptions	36
Table 3-2 - High-level Sector Names	46
Table 3-3- Categories of Fundamental Ratios	46
Table 3-4 - Absolute Error and Mean Absolute Error Formulae Variables.....	50
Table 3-5 – Accuracy, Precision and Recall Formulae Variables	51
Table 3-6 - F1 Score Formula Variables.....	52
Table 3-7 - Summary of Dataset Before and After Preprocessing.....	54
Table 3-8 - Random Forest Grid Search Parameters	55
Table 3-9 - Random Forest Optimal Model.....	56
Table 3-10 - GradBoost Grid Search Hyperparameters.....	56
Table 3-11 - GradBoost Optimal Model	57
Table 3-12 - XGBoost Grid Search Parameters.....	57
Table 3-13 - XGBoost Optimal Model	58
Table 3-14 - AdaBoost Grid Search Parameters	58
Table 3-15 - AdaBoost Optimal Model	59
Table 3-16 - Decision Tree Grid Search Parameters	60
Table 3-17 - Decision Tree Optimal Model.....	60
Table 3-18 - SVC Grid Search Hyperparameters	61
Table 3-19 - SVC Optimal Model.....	61
Table 3-20 - Neural Network Grid Search Hyperparameters	62
Table 3-21 - ANN Optimal Model.....	62
Table 3-22 - Logistic Regression Grid Search Parameters	63
Table 3-23 - Logistic Regression Optimal Model	63
Table 4-1 - Combined Metrics for Comparison.....	67
Table 4-2 - Precision, Recall, F1 Score and Support Values	72
Table 4-3 - Summary of Top Three Algorithms’ Accuracy Scores per Sector	74
Table 4-4 - XGBoost Accuracy per Sector.....	75
Table 4-5 - GradBoost Accuracy per Sector.....	76
Table 4-6 - Random Forest Accuracy per Sector.....	77
Table 4-7 - Detailed SHAP Feature Importance	95

List of Figures

Figure 2-1 - Effectiveness vs Time for Fundamental and Technical Analysis (Saini and Sharma, 2022)	7
Figure 2-2 - Support Vector Machine Decision Boundary (Schütze et al., 2008).....	19

Figure 2-3 – Single Layer Multilayer Perceptron (Chen et al., 2023)	21
Figure 2-4 - Forward Pass and Backpropagation (LeCun et al., 2015).....	21
Figure 2-5 - Chronology of Explanatory Methods (Holzinger et al., 2022)	26
Figure 2-6 - Popular Explanatory Methods on GitHub (Holzinger et al., 2022)	27
Figure 3-1 - Summary of FTSE/JSE ALSI Shares per Sector	41
Figure 3-2 - Long-term Share Prediction and Evaluation Workflow	44
Figure 3-3 - Experimental Design Overview for Experiment 1	47
Figure 3-4 - Implementation Workflow for Experiments 1,2 and 3	48
Figure 3-5 – Five-fold Cross Validation using GridSearchCV	49
Figure 3-6 - Generic Confusion Matrix	51
Figure 4-1 - Comparison of Accuracy, Precision, Recall and F1 Score Metrics	66
Figure 4-2 - Comparison of Accuracy, Absolute Error and Mean Absolute Error	67
Figure 4-3 - XGBoost Confusion Matrix	68
Figure 4-4 - XGBoost Correct versus Incorrect	69
Figure 4-5 - GradBoost Confusion Matrix	70
Figure 4-6 - GradBoost Tercile Prediction Accuracy: Correct versus Incorrect Predictions	70
Figure 4-7 - Random Forest Confusion Matrix	71
Figure 4-8 - Random Forest Tercile Prediction Accuracy: Correct versus Incorrect Predictions	71
Figure 4-9 - Summary of Top Three Algorithms’ Accuracy Scores per Sector	73
Figure 4-10 - XGBoost Accuracy per Sector	75
Figure 4-11 - GradBoost Accuracy per Sector	76
Figure 4-12 - Random Forest Accuracy per Sector	77
Figure 4-13 - XGBoost Technology Sector	78
Figure 4-14 - XGBoost Telecommunications Sector	78
Figure 4-15 - XGBoost Health Care Sector	79
Figure 4-16 - GradBoost Health Care Sector	80
Figure 4-17 - GradBoost Technology Sector	80
Figure 4-18 - GradBoost Real Estate Sector	81
Figure 4-19 - Random Forest Health Care Sector	82
Figure 4-20 - Random Forest Consumer Discretionary Sector	82
Figure 4-21 - Random Forest Telecommunications Sector	83
Figure 4-22 - XGBoost Naspers Limited (NPN) Share	84
Figure 4-23 - GradBoost Netcare Limited (NTC) Share	84
Figure 4-24 - Random Forest Netcare Limited (NTC) Share	85
Figure 4-25 - XGBoost SHAP Global Feature Importance	87
Figure 4-26 - Random Forest SHAP Global Feature Importance	88
Figure 4-27 - XGBoost SHAP Feature Importance for Tercile 1	89
Figure 4-28 - XGBoost SHAP Feature Importance for Tercile 2	90
Figure 4-29 - XGBoost SHAP Feature Importance for Tercile 3	91
Figure 4-30 - Random Forest SHAP Feature Importance for Tercile 1	92

Figure 4-31 - Random Forest SHAP Feature Importance for Tercile 2.....	93
Figure 4-32 - Random Forest SHAP Feature Importance for Tercile 3.....	94

Glossary of Terms

Term	Meaning
Security or Share or Stock	A security more broadly includes several types of financial instruments including equities, fixed income and derivative instruments. In the context of equity, a share is most often a unit of ownership in a company which represents one of the equal parts into which a company's capital is divided and a portion of the company's value.
Portfolio	A collection of shares and/or other financial assets held by an individual or organization, which is constructed in accordance with investment objectives.
Portfolio Construction	The process of creating a portfolio based on a range of asset classes typically including equity, fixed income and derivative securities. The different asset classes are used to adjust for various risk exposures of the underlying asset classes.
Stock Exchange	The entity which facilitates buyers and sellers to negotiate prices and make trades on listed securities. Securities traded on a stock exchange include stock issued by listed companies (shares), unit trusts, derivatives, pooled investment products and bonds.
Share or Stock Market	A general term which refers to a place where shares are traded and refers to the network of exchanges where shares may be traded.
JSE (Johannesburg Stock Exchange)	The main exchange in South Africa and is the largest share exchange in Africa.
Return	The profit (or loss) from an investment over a period of time. This can be expressed as a percentage or a nominal amount.
Excess Return	Positive additional return relative to the broader market or the return of a benchmark over an equivalent period of time.
Prediction Window	Timeframe or period over which a prediction is made. This can vary depending on the objectives of the prediction and the data used.
Long-term Prediction	A prediction occurring more than 6 months into the future. This is also synonymous with the prediction or forecast window.
Share Price Performance	The increase (or decrease) in the market price of a share.
Share Evaluation	The evaluation of a particular share to determine the expected future share price performance.

Active Investment Strategy	An investment strategy that involves ongoing buying and selling of shares by an investor. The strategy involves continual monitoring of share activity to exploit profitable conditions.
Share Index	A share index, or stock market index, is an index that measures a stock market or a subset of the stock market and allows investors to compare all current price levels with past prices to calculate market performance. The share index is generally computed from the prices of the selected shares.
Developed Market	These represent countries or markets that are typified by high income and economic growth, higher standard of living, good infrastructure and sophisticated capital markets.
Developing Market	These are countries or markets that might have some aspects of a development market but do not fully meet the standard.

Acknowledgements

I would like to thank my supervisor Deshendran Moodley for his continuous support and guidance during the process of completing my dissertation.

1. Introduction

Predicting share returns is regarded as one of the most challenging tasks because of the dynamic, nonlinear and complex nature of the financial markets, along with their long-term fluctuations (Wang et al., 2020). Despite this, the drive to achieve excess share returns on the share markets continues unabated, motivated by the potential for monetary rewards by those who can achieve this goal (Huang, 2019). The expected return on a share remains a crucial aspect used for preliminary share selection and is critical for portfolio management (Wang et al., 2020).

Fundamental analysis is performed by investors who look at the intrinsic value of shares based on broader measures related to the performance of the company, industry, economy, political climate, macro-economic drivers and then try to decide whether to invest (Patel et al., 2015). According to Sloan (2019), comprehensive fundamental analysis is an essential aspect of portfolio management, particularly for predicting share returns.

The time horizon to predict returns is a key consideration given that, in the short term, irrationally high share prices will move back toward fundamental values (Fama, 1991). Dependable investment decisions should rely on long-term observations and patterns of behaviour of asset data rather than those in the short-term (Wang et al., 2020). As such, using fundamental data with its inherent ability to consider a broader set of indicators is likely to inform a better investment decision outcome in the long-term.

Numerous share return prediction techniques using machine learning have been developed over the years with varying success (Kumar et al., 2021). There is no single machine learning algorithm that gives the best result when applied to different datasets (Kolanovic and Krishnamachari, 2017). Selecting an algorithm can itself prove challenging, as an algorithm may yield vastly different results on various datasets. Therefore, extensive experiments are necessary to compare the performance of different algorithms and identify the best performing one for a given dataset (Gu et al., 2020).

In recent years eXplainable Artificial Intelligence (XAI) has grown as a field and represents one of the most crucial challenges being faced by AI (Ignatiev, 2020). It focuses on the study and design of approaches, algorithms, and tools to produce human-understandable explanations of AI-based decisions (Ahmed et al., 2022). Complex algorithms are difficult to interpret creating a

tension between accuracy and interpretability (Lundberg and Lee, 2017). The inability to explain predictions dilutes the potential value for decision making as well as limiting the possibility of improving the underlying performance of algorithms provided by these insights.

Various approaches exist to help explain predictions with SHapley Additive exPlanations (SHAP) being one of the most popular and relevant techniques within the domain of XAI (Holzinger et al., 2021). SHAP has a solid theoretical foundation for explaining predictions for any type of algorithm (Aas et al., 2021). SHAP is a powerful tool that can help with the understanding of a complex algorithm and is able to verify an algorithm's accuracy by examining the various input features that contribute to its prediction (Kannangara et al., 2022).

1.1 Research Aim and Objectives

The overall aim of this research is to compare and analyse the interpretability of state-of-the-art machine learning approaches for predicting long-term share returns on the Johannesburg Stock Exchange (JSE).

The specific objectives of the research:

- Review existing research related to share return prediction or price movement and identify state-of-the-art techniques using machine learning classification algorithms.
- Use selected classification algorithms to predict the tercile class of each share's 12-month return.
- Use XAI techniques, such as SHAP, to analyse the influence of each input feature on the predictions of the best performing algorithms.

1.2 Tools and Approach

1.2.1 Overall Approach

The existing research related to share return prediction or price movement was reviewed. The goal was to identify state-of-the-art techniques using machine learning classification algorithms. Eight of the most popular machine learning algorithms were selected. Hyperparameter optimisation was performed for each algorithm and their performance was analysed in predicting the tercile class of the 12-month share returns, based on historical fundamental ratio data for shares listed on the JSE. Their performance was evaluated against common metrics. Three of the top performing algorithms underwent additional analysis to assess their sector-level and share-level predictive accuracy. SHAP global summary plots was used to identify the most influential input features and analyse the interpretability of two of the top performing algorithms.

1.2.2 Dataset

The dataset was sourced from FactSet, a global leader in market data and investment decision making solutions. It included semi-annual, end-of-week financial data for the 143 shares listed on the FTSE/JSE All Share Index (ALSI) as of the 31 December 2021. This was a retrospective extract for the period 1st January 2000 to 31st December 2021, comprising 22 years and, at most, 44 semi-annual observations per share. The data includes share identifiers, dates, closing prices and fundamental ratios derived from the financial statements, which are a required disclosure item for a share to be listed on the JSE. The fundamental ratios are grouped into Valuation, Profitability and Liquidity ratios.

1.2.3 Algorithms

The literature review identified the following popular machine learning classification algorithms for use in the experiments: XGBoost, Random Forest, GradBoost, AdaBoost, Decision Trees, Support Vector Classifiers, Logistic Regression, and Artificial Neural Networks. Furthermore, SHAP was identified in the literature review as a popular and relevant technique to identify specific input features, aiding in the analysis of the interpretability of the top performing algorithms.

1.2.4 Evaluation

The following machine learning metrics were used to compare and analyse the predictive performance of the selected algorithms: Accuracy, Absolute Error, Mean Absolute Error, Confusion Matrix, Precision, Recall and F1 Score. These evaluation metrics were identified as the key metrics used in previous work in this area. Additionally, a sector-level and share-level analyses were performed on the top three performing algorithms. SHAP global summary plots were applied to the XGBoost and Random Forest algorithms to identify key input features. These results were ranked and analysed to determine which features the trained models deemed most influential on long-term share returns.

1.2.5 Tools

The Python programming language was used for machine learning algorithms from popular frameworks TensorFlow and scikit-learn, as well as Pandas and SHAP. The associated libraries were run as Jupyter Notebooks both on a Windows and Google Colab platforms.

1.3 Contributions

There is a lack of research on the use of machine learning algorithms to predict long-term share returns on the JSE. The results of this study can be used to help select the appropriate algorithms

to predict long-term share returns. By applying SHAP global summary plots, the analysis of the input features identified the most important fundamental data ratios for prediction performance and interpretability. As machine learning algorithms evolve, the results from this study can also be used as a benchmark for future studies.

The outcomes of the research include:

A detailed comparison of various machine learning algorithms for long-term share return prediction using classification and fundamental ratio data, noting outcomes and differences identifying:

- Hyperparameters for each of the optimised algorithms.
- Evaluation results for each approach using the same input features and target variable.
- Comparative ranking of each approach using common evaluation metrics.
- Comparison of the various algorithms, including a baseline algorithm that uses a Uniform Baseline Classifier, agnostic of the input features.
- Detailed analysis of the findings of the top three algorithms, including sector-level and share-level predictive accuracy.
- Understanding of the impact of the input features to aid interpretability and their relative importance related to each of the top performing algorithms along with tercile class-specific outcomes using SHAP global summary plots.

1.4 Structure of the Dissertation

The literature review begins with a discussion of the share market and relevant theories, then provides a comparison of related work on share prediction using machine learning techniques that employ classification algorithms with fundamental data. The review also explores XAI and discusses related work that uses XAI techniques in financial prediction. This is followed by a description of the experimental design and execution which includes the problem definition, dataset used, experimental process including experiments conducted and implementation details, evaluation metrics, data preprocessing aspects and algorithms employed. The results and analysis are presented, comparing various algorithms using standard evaluation metrics. The analysis delves into the top algorithms in more detail, including sector-specific accuracy and examining the input features of the two top-performing models for greater model interpretability. Lastly, the conclusion and discussion are presented based on the findings, along with limitations and potential future studies which are identified and included for debate and possible investigation.

2. Literature Review

The literature review begins with an exploration of the share market, relevant theories and key aspects of machine learning. It proceeds to compare various studies on share prediction, focusing on the use of classification algorithms in machine learning that use either fundamental or technical data, while also identifying various research trends. The review explores various machine learning algorithms in more detail. Additionally, it delves into XAI and examines related research in financial prediction.

2.1 The Share Market: Johannesburg Stock Exchange (JSE)

The Johannesburg Stock Exchange (JSE) is the premier African exchange and provides investors with an orderly marketplace for trading shares in listed companies (JSE Limited Listing Requirements, 2019). Each company that is listed on the JSE has a continuing obligation to disclose financial information derived from the preliminary and final published financial statements. This information is required to ensure that the JSE is operated in the public interest and that companies meet listing requirements (JSE Limited Listing Requirements, 2019).

Investors trade in shares using various strategies to achieve excess market returns. These strategies are based on exploring publicly available information about companies, which includes financial information and ratios that the JSE requires to be disclosed biannually.

2.2 The Efficient Market Hypothesis

The origins of the Efficient Market Hypothesis date back to the turn of the 20th century by a French mathematician Bachelier (Walter, 2000). It became popularised in the 1960s by, among others, Fama, who published a paper on the random walk hypothesis stating that share prices could not be predicted (Fama, 1965). Later, Fama (1991) argued that the Efficient Market Hypothesis was based on share prices fully reflecting all publicly available information and, more specifically, that returns are unpredictable from past returns or past variables.

The Efficient Market Hypothesis has been challenged because it fails to explain market events such as bubbles and crashes, or the success of Warren Buffett and George Soros (Moffitt, 2018). It is also unlikely that investors are rational and trade on information as the Efficient Market Hypothesis implies (Shleifer, 2000).

A study by Lobão (2018) examined various statistical anomalies in the pricing of shares in key African markets. The South African share market had a high prevalence of anomalies, suggesting

that it is inefficient and has a long way to go before comparing itself to the capital markets of more developed economies (Lobão, 2018). This implies that the South African share market may offer the potential for greater returns to astute investors who can predict returns based on these market inefficiencies.

The successful use of machine learning prediction algorithms contradicts the Efficient Market Hypothesis by challenging the assertion that examining past data cannot yield excess returns (Ballings et al., 2015). Furthermore, emerging markets, such as South Africa have shown better accuracy with machine learning algorithms than developed countries, likely due to the greater complexity of these markets. (Bustos and Pomares-Quimbaya, 2020).

2.3 Modern Portfolio Theory

Harry Markowitz laid the foundation of Modern Portfolio Theory in a paper he published in 1952. Markowitz (1952) hypothesised that an investor presented with portfolios of equal returns will either choose the one with the least risk or make the trade-off between risk and return for the individual components of the portfolio.

The typical process of selecting a portfolio starts with observation to determine a view of the future performance of various shares that would be considered for inclusion (Markowitz, 1952). This is followed by the choice of shares to include in a portfolio based on maximising future returns and diversifying for minimum variance and avoiding high covariance between these shares (Markowitz, 1952). According to Markowitz (1952), the process of selecting shares should include statistical techniques to arrive at a tentative selection and then use judgment to account for various nuances not included in the formal calculations.

2.4 Techniques for Share Prediction: Fundamental, Technical and Sentiment Analysis

Popular share return prediction techniques can be categorised as fundamental, technical and sentiment analysis, along with their associated data (Huang, 2019). Fundamental and technical datasets are two of the most favoured for share or index prediction studies (Sezer et al., 2020). According to Saini and Sharma (2022) technical analysis approaches are most effective for short to medium term, while fundamental analysis is better for long-term investment horizons. The time horizon to predict returns is a key consideration, given that, in the short term, irrationally high share prices will move back toward fundamental values (Fama, 1991). In recent years, sentiment

analysis has gained interest due to more published information being available online (Kumbure et al., 2022).

Figure 2-1 presents a symbolic representation of the relationship between time and effectiveness for Fundamental and Technical Analysis.

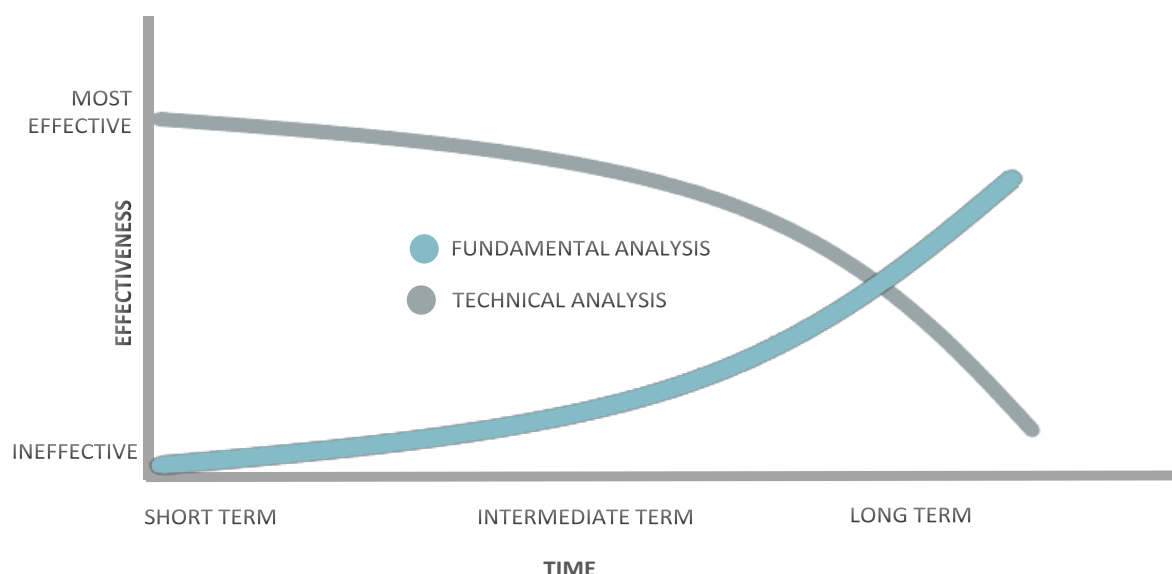


Figure 2-1 - Effectiveness vs Time for Fundamental and Technical Analysis (Saini and Sharma, 2022)

2.4.1 Fundamental Analysis

Fundamental analysis investigates the intrinsic value of shares using financial statements or financial ratios derived from these statements. It is also based on broader measures related to the performance of the industry and economy, political climate and macro-economic drivers (Patel et al., 2015). According to Sloan (2019) popular approaches to investing often overlook key information, specifically that comprehensive fundamental analysis is essential to investment decision making. This research focuses on using fundamental ratios, which are some of the most influential predictors of excess share returns (Drobetz and Otto, 2021).

2.4.2 Technical Analysis

Technical analysis is the evaluation of shares by studying statistics generated by market activity such as past prices to identify patterns and trends to predict future prices (Patel et al., 2015). Typically, this is done by charting various statistical indicators to assist with identifying trends and making investment decisions (Nti et al., 2020). Technical data that underlies the process of

technical analysis is used extensively with machine learning algorithms to predict the future return of shares, given that it is readily available (Nti et al., 2020).

2.4.3 Sentiment Analysis

Sentiment analysis utilises information typically obtained as text from sources such as published articles, social media, reports, and commentaries by independent or company officials related to shares (Huang, 2019). This type of data is considered hard to collect and process, but it is potentially useful given it can also surface additional information that is not available in the share market (Jiang, 2021). Sentiment analysis is considered effective over short-term time horizons for the prediction of share returns (Ma et al., 2021).

2.5 Machine Learning for Predicting Share Returns

This section describes the problem formulation from a machine learning perspective.

2.5.1 Problem Formulation

Share return prediction for machine learning purposes can be formulated as either a classification or a regression problem. If the prediction target is the category or class of return then it is considered as a classification problem (Jiang, 2021). A classification approach can predict whether to ‘buy, sell or hold’ a share based on its predicted excess returns. Alternatively, a quantile approach could predict the quantile of the predicted return. This approach also facilitates the construction of a mixed portfolio using a ‘long/short’ strategy, where the predicted top-performing quantile(s) are assigned to the long portfolio and the bottom-performing quantile(s) are assigned to the short portfolio. As such, an investor can maximise returns from both the market upsides and downsides.

In contrast, a regression problem involves predicting a specific value, such as a share return, which is a continuous variable (Jiang, 2021). A regression approach to portfolio construction predicts the future return of a share and derives the quantile ranking by comparing it to the future returns of other shares in the investible market. Again, a long/short portfolio could be constructed based on the quantile ranking. The majority of financial time series prediction is considered a regression type problem (Kaczmarek and Peres, 2021). The advantage of classification over regression is that it bypasses the need to predict specific share returns, which are then ranked for portfolio construction (Drobetz and Otto, 2021).

2.6 Classification Algorithms for Share Return or Price Trend Prediction

A comprehensive meta-analysis was conducted on numerous recent research papers that utilised machine learning classification algorithms for predicting share and share index returns or price trends. The analysis distilled the information from thirty-five research papers, aiming to identify both the most popular and state-of-the-art techniques currently in use. The summary of these findings is presented in Appendix A, Table A1 and in Table 2-1. Table 2-1 lists the research papers that primarily applied fundamental data in their analysis, while Appendix A, Table A1 includes those focusing mainly on technical data, along with either economic or fundamental data. Sentiment-based data has been excluded from this study, as it falls outside the scope of the current research, despite being recognised as a growing field of interest. The following discussion will highlight several pertinent aspects relevant to this study and the Literature Review.

2.6.1 Datasets: Fundamental versus Technical

Out of the nine research papers focused on fundamental data, which represent 26% of the total, six also include technical data. This supports the view that composite datasets may perform better (Bustos and Pomares-Quimbaya, 2020). The predominance of technical data in the current literature is confirmed by numerous other meta-analyses (Nti et al., 2020; Sezer et al., 2020; Jiang, 2021; Thakkar and Chaudhari, 2021).

Of the total, twenty-six research papers, representing 74%, primarily use technical data. This predominance may be attributed to the greater availability of technical data compared to fundamental data (Nti et al., 2020).

2.6.2 Feature Engineering

The primary goal of feature engineering is to increase the signal-to-noise ratio by correctly framing the problem and transforming the data to produce cleaner signals (Rasekhschaffe and Jones, 2019). In three research papers, feature engineering was applied to enhance the accuracy of predictions. The employed techniques included feature scaling and normalisation, dimension reduction using Principal Component Analysis and the removal of rows containing missing values (Zhong and Enke, 2019; Rasekhschaffe and Jones, 2019; Barak et al., 2017).

2.6.3 Prediction Window

According to Huang (2019), fundamental data is preferred for longer prediction horizons such as a quarter, a year, or longer (see Figure 2-1), and technical analysis is preferred for short-term prediction such as days, weeks, or less.

The prediction window for most of the research papers reviewed was short, ranging from intraday periods to days or weeks (Ghosh et al., 2021; Chen and Huang, 2021; Kamalov, 2020; Liu et al., 2021; Ntakaris et al. 2019; Peng et al., 2021; Haq et al., 2021; Yang et al., 2022; Yun et al., 2021; Wu et al., 2020). In contrast, longer prediction windows were more common among algorithms that utilised fundamental data as input features (Ballings et al., 2015; Drew, 2018; Drobetz and Otto, 2020; Huang, 2019; Kamble, 2017; Rasekhschaffe and Jones, 2019). According to Rasekhschaffe and Jones (2019), in their study, the AdaBoost algorithm, which employed fundamental data with a 12-month prediction window, was identified as the best base learning algorithm for predicting share returns. Kamble (2017) compared a long-term algorithm using fundamental data with an equivalent technical data algorithm featuring a short prediction window, finding that the long-term algorithm outperformed the short-term counterpart.

A major drawback for short-term prediction and high-frequency trading is the impact of frictional costs or trading commissions, which can erode potential returns (Huang, 2019). Typically, the best results are achieved when the prediction window aligns with the data frequency (Rasekhschaffe and Jones, 2019).

2.6.4 Classification Classes or Rankings

The most popular target variable among the various research papers was a simple directional or trend indicator, usually binary, such as ‘up’ and ‘down’ or ‘buy’ and ‘sell’. The most granular approach used was the decile ranking used with fundamental data (Drobetz and Otto, 2020; Liu et al., 2021; Rasekhschaffe and Jones, 2019). Other popular ranking methods included quintiles, quartiles, and terciles (Lohrmann and Luukka, 2019; Choi and Renelle, 2019; Barak et al., 2017; Abe and Nakayama, 2018; Drue, 2018). Drobetz and Otto (2020) noted that the more granular the class or ranking used for the classification task, the better the results achieved.

2.6.5 Rolling versus Recursive Time Windows

The most popular approach among the research papers in Appendix A, Table A1 and Table 2-1 was a rolling window approach. This approach is used to keep the training and test lengths the same while moving through time. In contrast, a recursive window will incorporate the entire history of data, resulting in more computationally intensive updates to a machine learning model (Drobetz and Otto, 2020).

Given that fundamental data is updated infrequently, consideration should be given to the timing of refitting an algorithm, as this can affect both accuracy and the computational efficiency (Drobetz and Otto, 2020).

2.7 Applying Machine Learning to Fundamental Data

This section reviews existing studies based on the fundamental data, a combination of fundamental and technical data, and related work in both South African and international domains, while also identifying research trends.

2.7.1 Fundamental Data

Only four studies representing 11% of the evaluated papers used fundamental data exclusively, as shown in Table 2-1. Most of the prior research has included a combination of fundamental and technical data, as shown in Appendix A, Table A1.

Random Forest was featured in three of the studies (Ballings et al., 2015; Barak et al., 2017; Huang et al., 2021), ranking as the top performing algorithm in two of them (Ballings et al., 2015; Huang et al., 2021), while Decision Trees appeared in two others (Barak et al., 2017; Cheng and Chen, 2007). Neural Networks were employed in three studies (Barak et al., 2017; Cheng and Chen, 2007; Huang et al., 2021). Support Vector Machines were mentioned in three studies (Ballings et al., 2015; Barak et al., 2017; Drue, 2018) and notably, they outperformed a factor-based algorithm that utilised multiple regression in one of these studies (Drue, 2018).

2.7.2 Technical and Fundamental Data

Neural Networks, encompassing various types, have been extensively utilised in most recent studies concerning share return or price trend prediction (Wu et al., 2020; Chen and Hang, 2021; Ghosh et al., 2021; Kamalov, 2020; Livieris et al., 2020; Peng et al., 2021; Yang et al., 2022). The most employed types include the Multilayer Perceptron, Long Short-Term Memory and Convolutional Neural Networks.

Support Vector Machines, ranking as the next most popular algorithm, have demonstrated superior performance over many Neural Networks models in several studies (Kim, 2003; Huang et al., 2005; Shynkevich et al., 2017; Dingli and Fournier, 2017). Lastly, Random Forest was the third most frequently used algorithm and has outperformed other algorithms in various studies (Lohrmann and Luukka, 2019; Patel et al., 2015).

2.7.3 Related Work in the South African Domain

In a study by Drue (2018), factor-based investing using multiple regression was compared with machine learning using a Support Vector Classifier (SVC) algorithm. The study was based on fundamental ratio data derived from financial statements. The target variable was the classification of the 3-month excess return for each share into either the underperforming or the outperforming category. Fourteen fundamental ratios, utilised both individually and collectively over a 15-year period for shares listed on the FTSE/JSE All Share Index, formed the basis of this study. The findings revealed that the SVC algorithm, employing all ratios within a recursive window, consistently surpassed the traditional factor-based investing approach. This performance was underscored by a 11.49% excess return over the full period and an accuracy of 61.78% (Drue, 2018).

2.7.4 Related Work Internationally

Ballings et al. (2015) benchmarked ensemble algorithms, including Random Forest, AdaBoost, and Kernel Factory, against Neural Networks, Logistic Regression, Support Vector Machines, and K-Nearest Neighbours, for predicting the long-term share price movement a year in advance as either a 'buy' or 'sell' signal. The dataset comprised 5,767 listed European shares over a 5-year period. Various fundamental ratios, along with six economic indicators, were utilised in the algorithms. The findings show that, on aggregate, the ensemble algorithms outperformed all the other types, with the Random Forest achieving a median AUC (Area Under Curve) score of 0.9037 from an evaluation metric perspective. This score significantly outperformed the next best score of 0.8395 median AUC achieved by the Support Vector Machine algorithm.

Barak et al. (2017) compared the outcomes from a fusion approach that generated models for predicting share returns into a quintile class. The dataset comprised records from the Tehran Stock Exchange between 2002 to 2012, totalling 1951 records across 400 companies. Eight fundamental ratios were used as input features for the share return prediction, including: return, market return, beta coefficient, return on assets, percentage of growth in EPS, EPS, predicted profit margin and EPS coverage percent.

The models were created using base classifier algorithms along with three diversity methods: Bagging, Boosting and AdaBoost. The employed base algorithms included Decision Tree, Multilayer Perceptron, Decision Tree Naïve Bayes, Logical Analysis of Data (LAD) Tree, Reducing Error Pruning (REP) Tree, K-Nearest Neighbour, Support Vector Machine and Random Forest. A Meta classifier was utilised to learn from the output of the base classifiers, aiming to

achieve the highest prediction accuracy. The Bagging method of diversification, employing Decision, LAD and REP Trees as base classifier algorithms, achieved the highest accuracy of 83.65% for predicting share returns.

Cheng and Chen (2007) conducted the earliest study among those reviewed, employing Decision Tree and Multilayer Perceptron algorithms, as well as Bayes Net and Rough Set techniques, to predict the tercile class revenue growth rate of shares listed on the Taiwan Stock Market. The dataset comprised 636 shares listed on Taiwan Stock Exchange between 2004 to 2005. Twelve input features were used for predicting the share revenue growth rate, including total fixed assets, gross sales, net sales (revenue), cost of goods sold, gross profit, operating expenses, operating income, total non-operating income, total non-operating expenses, total salary, total depreciation and number of employees. The Rough Set technique achieved the highest accuracy for the 2004 sub-dataset of 75.15% ($\pm 1.6\%$).

Huang et al. (2021) used 22 years of quarterly financial data from S&P 100 shares to investigate the predictive accuracy of three machine learning algorithms, using 21 input features for 70 shares, with 88 observations per share. The Multilayer Perceptron, Random Forest, and Adaptive Neural Fuzzy Inference System algorithms were used to predict 3-month relative returns as either a buy or sell signal. The primary evaluation metric, Portfolio Score, considered both portfolio performance and standard deviation over the validation period. The Random Forest algorithm exhibited the best performance. The 'buy' portfolio generated a 32.1% compound return with a Portfolio Score of 0.414, while the 'sell' portfolio had a -23.7% compound return and a Portfolio Score of -0.305.

The feature importance analysis for the Random Forest algorithm showed the following ranking of the top six input features:

1. Price to Book
2. Relative Return
3. Book Value
4. Price Earnings
5. Capital Expenditure
6. Liability

Three of the top four input features are valuation-based, while ‘Relative Return’ is the only profitability-based feature among them. The primary finding suggests that machine learning models can assist fundamental analysts with decision-making, even when working with a limited dataset.

Table 2-1 - Summary of Research – An Examination of Classification Algorithms Applied to Primarily Fundamental Data

No.	Paper	Data Set	Features	Algorithms	Target Variable and Class	Prediction Window	Performance Criteria	Result
1	Ballings et al., 2015	Listed Shares in Europe 2009 - 2014	Fundamental and Economic	LR, SVM, ANN, K-Nearest Neighbour (KNN), RF, AdaBoost.	Share Price Movement Binary (up and down)	1 year	AUC (Area Under the Curve)	RF (0.903) and SVM (0.839) are best performers when comparing median AUC results to predict price increases one year ahead.
2	Barak et al., 2017	Tehran Stock Exchange 2002-2012	Fundamental	LAD Tree, CART Decision Tree, DTNB Rule, Decision Table, Rep Tree, BF Tree, Part Rule, J48 Graph, MLP, Bayes, RF, SVM	Share Return - Quintiles (very low, low, normal, high and very high))	None provided	Accuracy	Bagging Fusion with Decision Tree, LAD Tree and Rep Tree achieved best result of 83.6% accuracy.
3	Cheng and Chen, 2007	Shares listed on Taiwan Stock Exchange. 2004-2005	Fundamental	Decision Tree (C4.5), ANN, Bayes Net, Rough Set	Revenue Growth Rate Terciles (negative, positive, and high positive)	None provided	Accuracy	Rough Set outperformed the other method with an average accuracy of 75.15% ($\pm 1.6\%$) for 2004.
4	Drue, 2018	Shares included in the All Share Index of the Johannesburg Stock Exchange 2003-2018	Fundamental	Factor-based Investment (FBI) using Linear Regression, SVM	Excess Share Return Binary (underperform and overperform).	3 months	Accuracy, Return	SVM outperformed the Factor-based Investment algorithm when comparing excess returns over various periods.
5	Huang et al., 2021	Shares included in the S&P 100 Index (Q1 1996 to Q4 2017)	Fundamental	ANN(MLP), RF and Adaptive Neural Fuzzy Inference System	Shares quarterly relative return Binary (buy or sell)	3 months	Portfolio Score, Compound Return, Standard Deviation	RF achieves best performance: buy portfolio achieving 32.1% compound return and a Portfolio Score of 0.414. Sell

No.	Paper	Data Set	Features	Algorithms	Target Variable and Class	Prediction Window	Performance Criteria	Result
								portfolio achieving -23.7% compound return and a Portfolio Score of -0.305.

2.7.5 Research Trends

An evaluation of the research papers suggests that Random Forests, Support Vector Machines, Neural Networks, Decision Trees were the most popular algorithms and should be included in any comparative study (Ballings et al., 2015; Barak et al., 2017; Drue, 2018; Huang et al., 2021). Random Forest achieved the best results in two of the studies, making it the most useful algorithm for prediction (Huang et al., 2021; Ballings et al., 2015).

Two of the most recent studies utilised Multilayer Perceptron Neural Networks for their prediction tasks (Huang et al., 2021; Barak et al., 2017). In the study by Huang et al. (2021), the results from the Neural Network were surpassed by the Random Forest ensemble technique.

2.8 Machine Learning Algorithms

While numerous algorithms are discussed in the broader machine learning literature in finance, this study will specifically focus on those identified in the Literature Review. These algorithms have consistently achieved superior results in classification approaches using fundamental and/or technical data to predict share returns or price movements (see Appendix A, Table A1 and Table 2-1). The algorithms under consideration include Random Forest, GradBoost, XGBoost, AdaBoost, Decision Tree, Support Vector Machines, Logistic Regression and Neural Networks (both Artificial Neural Networks and Deep Neural Networks).

2.8.1 Random Forests

Random Forests are an ensemble learning technique, specifically a type of bagging, frequently used in both regression and classification tasks (Bustos and Pomares-Quimbaya, 2020). The underlying mechanics of the algorithm make it highly suitable for the share market as it carefully considers all the features to predict the most likely outcome (Basak et al., 2019). Noted for their precision, Random Forests are robust to outliers and noise, efficient in execution, and lend themselves well to parallel computing (Breiman, 2001).

The Random Forest algorithm constructs an ensemble of decision trees through Binary Recursive Partitioning (BRP). In this process, each tree is grown using a random subset of candidate features and evaluates multiple potential splits within those features (Ballings et al., 2015). A bootstrap aggregation, or 'bagging', method is utilised, whereby numerous uncorrelated decision trees collectively contribute to the classification of new samples (Basak et al., 2019).

During classification, individual trees in the ensemble predict class labels, and the class with the highest vote count becomes the overall model prediction (Breiman, 2001). In regression tasks, the individual tree predictions are aggregated using the mean function to obtain the final model prediction (Breiman, 2001).

There are two key parameters in a Random Forest: the number of trees in the ensemble and the number of variables to try at each split (Ballings et al., 2015). The recommended practice is to use a substantial number of trees, with the number of variables to try at each split typically set to the square root of the total number of input features available (Ballings et al., 2015).

The mathematical representation of the Random Forest algorithm can be found in Appendix A2, titled 'Random Forest: Mathematical Representation'. The available parameters for the scikit-learn Random Forest Classifier module are provided in Appendix A2, titled 'Random Forest Classifier: Parameters'.

2.8.2 **GradBoost**

GradBoost, short for Gradient Boosting, is an ensemble method that utilises boosting techniques for both regression and classification tasks. This method begins with a simple initial model, often a decision tree, and then iteratively adds weak learners, typically shallow trees, to address the deficits of the existing learners in a stage-wise manner (Friedman, 2001). The algorithm aims to optimise a loss function, which is a measure of how well the model fits the training data. It employs gradient descent to minimise this loss, specifically by calculating the gradient of the loss function relative to the current model's predictions (Friedman, 2001). During the process, trees are added sequentially, without altering existing trees in the model. The parameters for each new tree are determined through a gradient descent step (Natekin and Knoll, 2013). Additionally, a learning rate or shrinkage factor moderates each tree's contribution, ensuring a gradual and controlled improvement of the ensemble model (Natekin and Knoll, 2013). This iterative process continues until the model converges or a predefined stopping criterion is met.

Gradient boosting is effective because it can handle various types of input features and accommodate missing data. It is differentiated from other boosting algorithms by its use of gradient descent to minimize the loss (Friedman, 2001).

The mathematical representation of the GradBoost algorithm can be found in Appendix A2, titled ‘GradBoost: Mathematical Representation’. The available parameters for the scikit-learn GradBoost Classifier module are provided in Appendix A2, titled ‘GradBoost Classifier: Parameters’.

2.8.3 XGBoost

XGBoost (eXtreme Gradient Boosting) is considered a recent ensemble method based on decision trees and employs the rules of boosting for weak learners and is considered a faster performer compared to other tree-based algorithms (Nabipour et al., 2020). It employs Lasso (L1) and Ridge (L2) Regression regularisation to prevent overfitting and has built in cross-validation capability (Nabipour et al., 2020). In contrast to traditional gradient boosting methods, XGBoost introduces a novel tree pruning technique, which involves growing trees to their maximum depth before pruning them back, thus optimising tree structure (Chen and Guestrin, 2016). XGBoost is designed for computational efficiency, with features like parallel computing and cache awareness, making it faster than many other gradient boosting implementations (Chen and Guestrin, 2016). XGBoost natively incorporates a missing value handling mechanism, eliminating the need for pre-imputation. This mechanism operates within the tree learning process, automatically accounting for missing data during node splits and leaf value calculations (Chen and Guestrin, 2016).

The mathematical representation of the XGBoost algorithm can be found in Appendix A2, titled ‘XGBoost: Mathematical Representation’. The available parameters for the scikit-learn XGBoost Classifier module are provided in Appendix A2, titled ‘XGBoost: Parameters’.

2.8.4 AdaBoost

AdaBoost (Adaptive Boosting) is widely used ensemble method for classification that employs a boosting algorithm. It is known for its simplicity in implementation and high adaptability (Wang and Feng, 2020). The AdaBoost algorithm sequentially trains weak learners, which are classifiers performing slightly better than random guessing. It progressively transforms these learners into a stronger composite model by focusing specifically on areas where previous learners' predictions were weak (Wang and Feng, 2020). After each iteration, the algorithm adapts by adjusting the

weights of misclassified instances, a feature that contributes to its adaptability (Freund and Schapire, 1997). AdaBoost is considered robust against overfitting but its performance degrades in the presence of noisy data (Schwenk and Bengio, 2000).

The mathematical representation of the AdaBoost algorithm can be found in Appendix A2, titled ‘AdaBoost: Mathematical Representation’. The available parameters for the scikit-learn AdaBoost module are provided in Appendix A2, titled ‘AdaBoost: Parameters’.

2.8.5 Decision Trees

Decision Trees can be used for both regression and classification. They create a tree-like structure of decision rules, which enables a systematic approach to predicting values and classifying instances (Kingsford and Salzberg, 2008). They are considered easy to implement and interpret, but are prone to overfitting with many input features, leading to poor performance on unseen data (Kotsiantis, 2013).

In a decision tree, each internal node represents an attribute, while each leaf node signifies a label or value. The edges connecting these nodes indicate the potential values that the attributes may assume. Attributes at each level of the tree are selected based on their efficacy in classifying or predicting the target variable (Quinlan, 1986).

Attribute selection in Decision Trees is critical, using metrics such as Information Gain, Gini Impurity, and Chi-Squared tests to identify attributes that optimally separate data. Information Gain assesses the effectiveness of an attribute in classifying training examples, with higher values indicating greater efficacy (Kotsiantis, 2013).

To mitigate overfitting, a common challenge in Decision Trees characterised by strong performance on training data but poor generalisation on new data, pruning methods are applied. These methods eliminate sections of the tree that contribute minimally to instance classification, thereby enhancing the model’s generalisability (Breiman et al., 1984). Decision Trees address missing values through methods such as surrogate splits. In this approach, the algorithm identifies an alternative attribute that effectively replicates the division the original attribute would have achieved, thus handling missing values efficiently (Kotsiantis, 2013).

The mathematical representation of the Decision Tree algorithm can be found in Appendix A2, titled ‘Decision Tree: Mathematical Representation’. The available parameters for the scikit-learn Decision Tree module are provided in Appendix A2, titled ‘Decision Tree: Parameters’.

2.8.6 Support Vector Machines

Support Vector Machines (SVMs) have emerged as a powerful tool for addressing both classification and regression challenges (Cortes and Vapnik, 1995). SVMs are considered to achieve high levels of prediction accuracy, due primarily to the small number of hyperparameters required for tuning and their relative simplicity of implementation (Bustos and Pomares-Quimbaya, 2020). SVMs, including their classification-specific variant SVC (Support Vector Classifier), are a popular tool for classification with good performance and have been widely adopted (Jiang, 2021).

Cortes and Vapnik (1995) describe an SVM as a technique where input features are non-linearly mapped to a high-dimensional feature space. In this space, a linear decision boundary, also known as a hyperplane, is constructed to distinguish between two classes of data (Figure 2-2). The dimensionality of the hyperplane is $p-1$, which is one dimension less than the dimensionality of the feature space, denoted as p . When there are two input features, the hyperplane manifests as a line, which is one-dimensional. Similarly, in a feature space described by three variables, the corresponding hyperplane is a plane, which is two-dimensional.

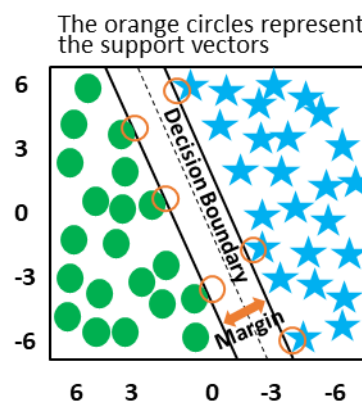


Figure 2-2 - Support Vector Machine Decision Boundary (Schütze et al., 2008)

Various kernels can be used to map the input training samples into a high-dimensional space (Jiang, 2021). Among these, the most frequently utilised are linear, polynomial, radial basis function (RBF), and sigmoid (Ballings et al., 2015).

The mathematical representation of the Support Vector Classifier algorithm and kernels can be found in Appendix A2, titled ‘Support Vector Classifier: Mathematical Representation’. The available parameters for the scikit-learn Support Vector Classifier module are provided in Appendix A2, titled ‘Support Vector Classifier: Parameters’.

2.8.7 Neural Networks

Neural Networks (NNs) encompass a wide range of computational architectures within machine learning, including both Artificial Neural Networks (ANNs) and Deep Neural Networks (DNNs). DNNs are characterised by their complex architectures, typically consisting of three or more layers in an Artificial Neural Network. There are several distinct types of Neural Networks, such as Deep Multilayer Perceptron, Convolutional Neural Networks, Recurrent Neural Networks, Long Short-Term Memory Networks, Deep Belief Networks, Restricted Boltzmann Machines, and Autoencoders (Ozbayoglu et al., 2020).

The primary advantage of NNs is their ability to automatically learn the most important features from input data using a general learning methodology, which has led to their application across a diverse range of fields (Sezer et al., 2020). DNNs are arguably the most powerful algorithms in machine learning and are the preferred approach for complex problems such as computer vision, natural language processing and automated game-playing (Gu et al., 2018). DNNs frequently outperform other machine learning techniques on very large and complex problems, an advantage that has been amplified by the proliferation of powerful graphics processors initially developed for the gaming industry (Géron, 2017).

A significant number of studies have used DNN classification algorithms for financial trend prediction (Sezer et al., 2020). DNN algorithms require more computational power and can suffer from overfitting that can reduce their generalisation ability (Zhong and Enke, 2019).

Multilayer Perceptron

It consists of three parts: an input layer, one or more hidden layers and an output layer as depicted in its simplest form in Figure 2-3. Deep Multilayer Perceptron (DMLP) differs from a Multilayer Perceptron in that it contains a greater number of hidden layers, which generally leads to improved performance (Ma et al., 2021).

The number of neurons in each layer and the total number of layers, constitute the hyperparameters of the network. These also include the input (x), weight (w) and bias (b) parameters (Sezer et al., 2020).

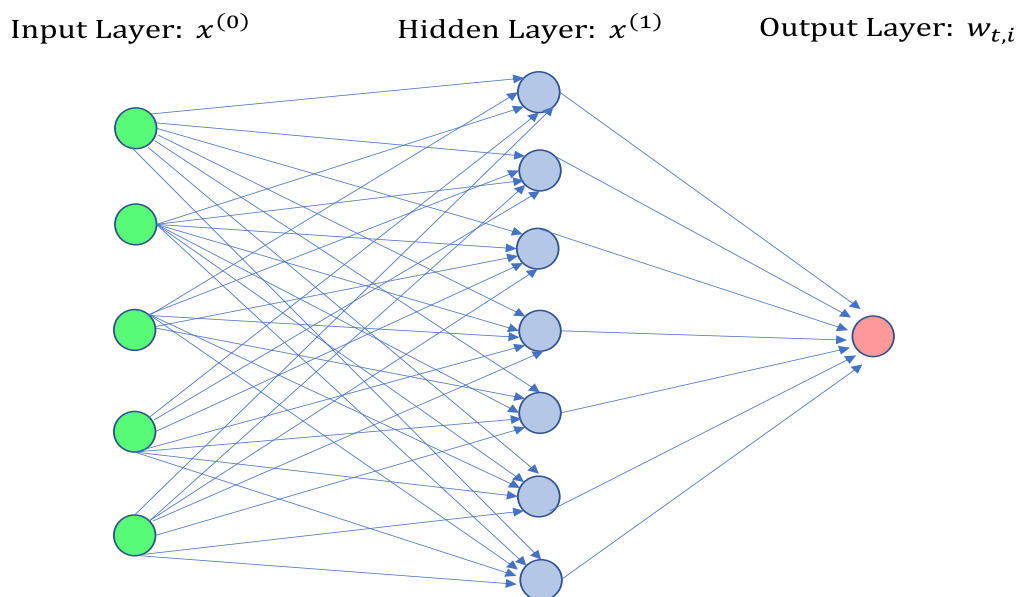


Figure 2-3 – Single Layer Multilayer Perceptron (Chen et al., 2023)

The learning process for Deep Multilayer Perceptrons is implemented through backpropagation (see Figure 2-4), where the output layer's neurons send feedback to the neurons in the previous layers (Sezer et al., 2020).

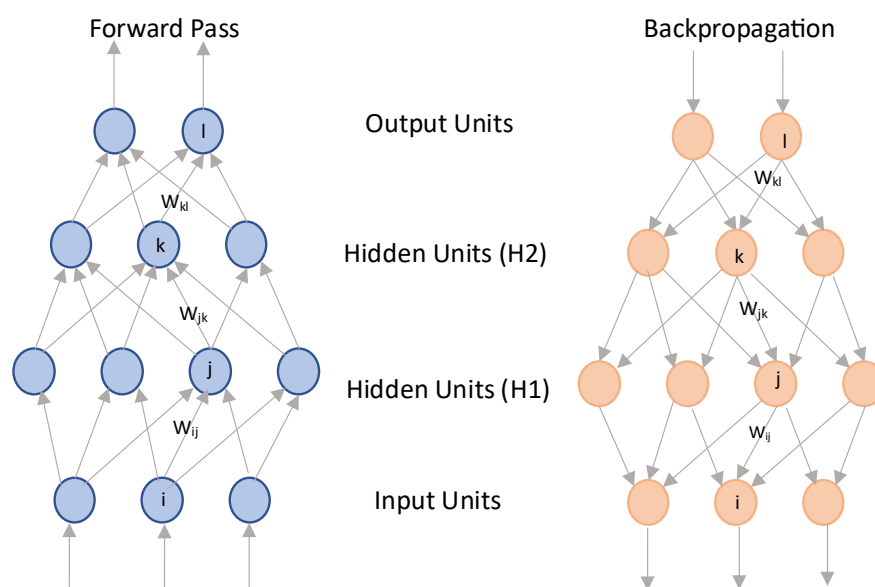


Figure 2-4 - Forward Pass and Backpropagation (LeCun et al., 2015)

Deep Multilayer Perceptrons have problems with computational performance as the number of input features increase due to the fully connected nature of the algorithm but generally outperform shallow Artificial Neural Networks in both classification and regression (Sezer et al., 2020).

Different optimisation algorithms have been developed to find the best possible parameters of each neural network that are used to update the weights of the connections between the layers (Sezer et al., 2020). The Adaptive Moment Estimation (ADAM) is an upgrade to Root Mean Square Propagation (RMSProp) and works well on online and non-stationary settings as well as with sparse gradients and is a popular choice (Ma et al., 2021; Sezer et al., 2020). Finding the best parameters for a Deep Multilayer Perceptron is critical as this affects both the architecture and performance (Sezer et al., 2020).

The mathematical representation of the Multilayer Perceptron algorithm and activation functions can be found in Appendix A2, titled ‘Multilayer Perceptron: Mathematical Representation’. A comprehensive overview of the parameters utilised for the Keras Neural Network API, which employed an MLP architecture in this study, is provided in Appendix A2 in the Table, titled Table ‘Keras Neural Network API: Parameters’.

2.8.8 Logistic Regression

Logistic regression is similar to linear regression, with the exception that it applies a sigmoid function (Nabipour et al., 2020). It can be used for multiclass classification. The algorithm employs a logistic function to constrain the output of a linear equation to a range between 0 and 1, using a threshold value of probabilities for different classes (Molnar, 2020). Although logistic regression may exhibit less predictive accuracy compared to some other algorithms in certain scenarios, and its interpretation can be less straightforward, it is still a widely used due to its efficiency and strong theoretical foundation (Molnar, 2020).

Logistic regression employs a logistic function to estimate the likelihood of a specific class or event. This function, characterized by an S-shaped curve, transforms any real-valued number into a probability value ranging from 0 to 1, without reaching these extremes. The algorithm's coefficients, known as β values, are determined from the training data, usually through the process of Maximum Likelihood Estimation (MLE) (Peng et al., 2002).

The mathematical representation of the Logistic Regression algorithm can be found in Appendix A2, titled ‘Logistic Regression: Mathematical Representation’. The available parameters for the

scikit-learn Logistic Regression module are provided in Appendix A2, titled ‘Logistic Regression: Parameters’.

2.9 Evaluation

This section covers the evaluation metrics identified in the Literature Review. This includes classification metrics, followed by return analysis and baseline approaches.

2.9.1 Classification Metrics

Classification metrics, used to measure the performance of an algorithm for either price trend or share return prediction, are extensively referenced in the research papers included in Appendix A, Table A1 and Table 2-1. The most widely used metric is accuracy, defined as the total number of correct predictions for directional change or quantile ranking compared to the actual number (Jiang, 2021). Another popular measure is the area under the receiver operating characteristic curve (AUC), which provides a single aggregate measure of performance across all possible classification thresholds. It indicates how well the model distinguishes between classes. AUC is considered a useful performance metric (Ballings et al., 2015; Kamalov, 2020; Yang et al., 2022). Other metrics include Precision, Recall, F1 Score, Hit Ratio, Matthews Correlations Coefficient (MCC) (Basak et al., 2019; Ul et al., 2021). Confusion matrix has also been used as a summary of prediction results (Guang et al., 2019). Mean Square Error (MSE) and Mean Absolute Error (MAE) have also been used in several research papers (Ma et al., 2021; Drobetz and Otto, 2020; Zhong and Enke, 2019).

2.9.2 Return Analysis

Return analysis evaluates whether a share’s predicted return can generate either an excess return over a period of time or a profit based on a theoretical portfolio over a holding window. Typically, such a portfolio is constructed with a long and short strategy where the highest ranked shares are bought and the lowest ranked shares are sold based on a quantile ranking (Abe and Nakayama, 2018). It can also be evaluated from the risk perspective. Sharpe Ratio, introducing the element of risk, is a comprehensive metric that considers both the return and risk. It is defined as the average return earned in excess of the risk-free rate, per unit of volatility. The risk-free rate is the return on an investment with zero risk, such as cash, whose return rate can be offset against the share or portfolio return.

2.9.3 Baseline Approaches

Baseline approaches, employed for comparative purposes in machine learning, can include linear algorithms (Ballings et al., 2015), as well as published equal weighted indices (Kaczmarek and

Perez, 2021). Although the Autoregressive Integrated Moving Average (ARIMA) was not mentioned in the research papers summarised, it remains a popular choice in the broader literature. ARIMA, an extension of Autoregressive Moving Average (ARMA), is designed specifically to handle non-stationary time series data (Jiang, 2021).

2.10 Explainable AI (XAI)

This section provides background on XAI, including the differences between global and local explanations, model-agnostic versus model-specific explanations and popular methods. It also includes a detailed review of SHAP and a survey of popular XAI approaches in finance.

2.10.1 Background

The term eXplainable AI (XAI) was initially coined by the United States' Defense Advanced Research Projects Agency (DARPA) during a programme it sponsored in 2017 (Gunning and Aha, 2019). XAI primarily focuses on the study and development of methodologies, models, and tools that produce explanations of AI-based systems, their information, and decisions that are comprehensible to humans. (Ahmed et al., 2022). This is critical given that many AI approaches are often termed 'black boxes' due to their opaque nature. The inability to explain their behaviour diminishes the trust and potential value that can be extracted from such models. Recently, interest in XAI has grown due to regulatory changes, along with an increase in investment by key stakeholders in the industry and government sectors (Dazeley et al., 2021).

Government regulators in both Europe and America are placing increased emphasis on the fairness and discrimination issues in decision making processes using machine learning algorithms, particularly in healthcare and finance (Albini et al., 2021). In America, regulations within the credit industry prescribe that automatic decisions must be explained in terms of key inputs that contribute to an adverse decision (Albini et al., 2021). Meanwhile, in the European Union, the General Data Protection Regulation (GDPR) mandates that autonomous systems must provide explanations of any decisions made (Dazeley et al., 2021). Given the trend towards further integration of autonomous systems into societal frameworks, the legislative demand for explanation is anticipated to grow (Dazeley et al., 2021).

The objective of XAI is to empower human experts to understand the explanatory aspects underlying an AI decision, thereby ensuring both an ethical and transparent outcome in machine learning decision support (Holzinger et al., 2022). XAI arguably represents one of the most crucial challenges being faced by the field of AI (Ignatiev, 2020).

Understanding a model and the reasons behind its predictions can be as crucial as the model's accuracy. There exists an inherent tension between accuracy and interpretability; complex models, while potentially more accurate, are harder to interpret (Lundberg and Lee, 2017). Frequently, methods that perform optimally, such as deep learning models are the least explainable, whereas the most explainable such as decision trees may not achieve the same level of accuracy (Gunning and Aha, 2019).

2.10.2 Global versus Local Explanations

Existing XAI models can be classified into two principal categories: global and local explanations (Aas et al., 2021). Global explanations attempt to describe the model as a whole, focusing on which input features most significantly influence the overall model. This is achieved through methods such as either permutation-based feature importance or by generating partial dependence plots (Aas et al., 2021).

Local explanations identify the influence of different input features on a specific prediction made by an algorithm and are commonly referred to as individual prediction explanation methods (Aas et al., 2021). These explanations are particularly valuable for complex algorithms where performance varies with different combinations of features, and where the local behaviour may not align with the global explanation (Aas et al., 2021).

2.10.3 Model-Agnostic and Model-Specific Explanations

Within local explanations, two further categories exist: model-specific and model-agnostic. These aim to provide simple, interpretable explanations for individual predictions made by the algorithm (Aas et al., 2021). Model-agnostic explanations, also referred to as black-box explanations, rely solely on accessing the algorithm's input and output (Aas et al., 2021).

2.10.4 Popular Methods

A variety of explanatory methods have been developed in recent years. Figure 2-5 illustrates the chronological progression of these explanatory methods.

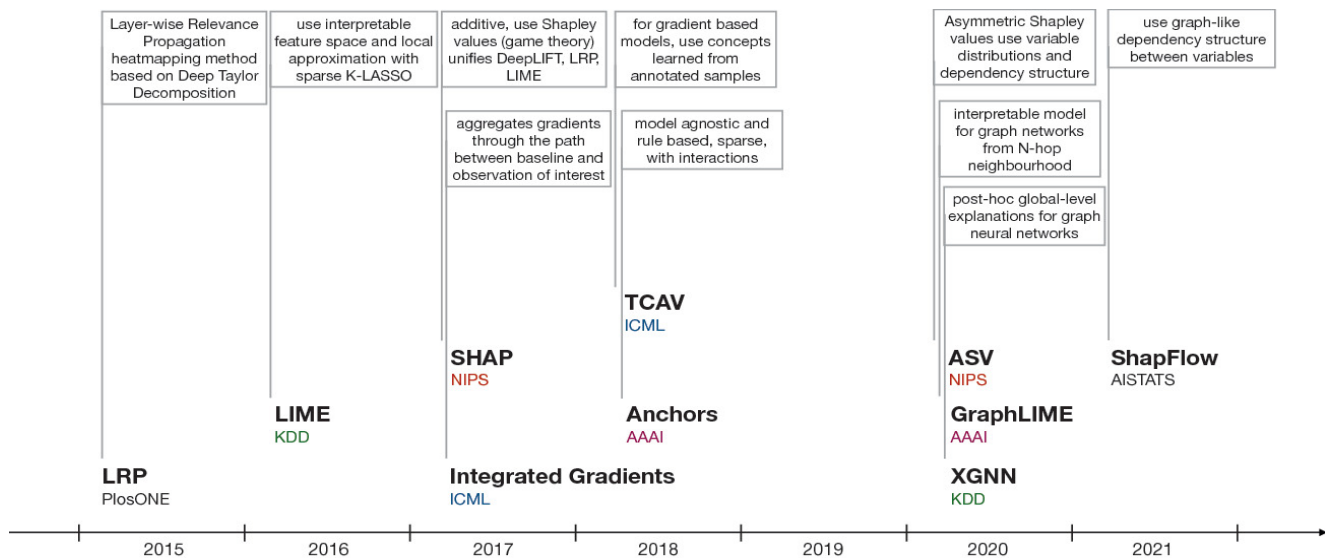


Figure 2-5 - Chronology of Explanatory Methods (Holzinger et al., 2022)

Figure 2-6 displays the number of star ratings on GitHub for the most popular methods, which includes Shapley values or SHAP (SHapley Additive exPlanations) as the marginal leader, followed by LIME (Local Interpretable Model-agnostic Explanations), and subsequently by newer methods that are increasing in popularity.

Both of these popular methods, SHAP and LIME, are model-agnostic as they rely solely on the model's inputs and outputs, with feature weights determined without access to the model's internal workings. The distinction between their approaches lies in the fact that SHAP explains the contribution of each feature to the difference between the actual prediction and the expected value (or global average prediction), while LIME provides explanations by approximating the model locally around the prediction (Aas et al., 2021).

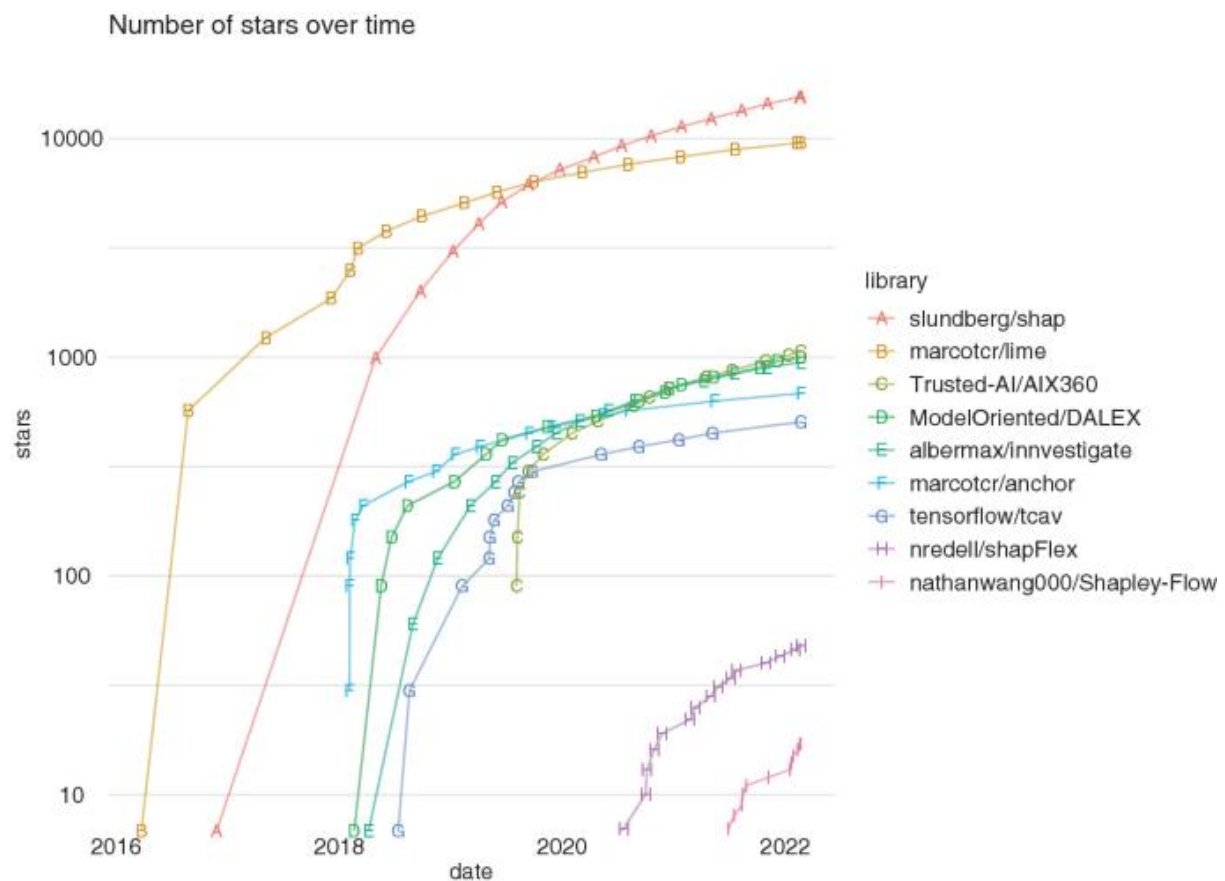


Figure 2-6 - Popular Explanatory Methods on GitHub (Holzinger et al., 2022)

2.10.5 SHAP

Overview

SHAP (SHapley Additive exPlanations) provides a robust theoretical framework for explaining the predictions of various algorithms (Aas et al., 2021). This method supports both local and global interpretations, maintaining its relevance and popularity, as evidenced by its consistent usage on GitHub. Developed as a Python library, SHAP was originally conceptualized by Lundberg and Lee in 2017. This framework surpasses its predecessors in terms of computational efficiency and exhibits a closer alignment with human intuition (Lundberg and Lee, 2017).

The methodology of SHAP is grounded in the concept of Shapley values, which are derived from classical cooperative game theory. In this context, each feature acts as a player in an m -person game, and the aim is to fairly distribute the contribution of each feature to the model's output (Albini et al., 2021). The characteristic function, $v: 2F \rightarrow \mathbb{R}$, assigns the payoff among a set of players $F = \{1, \dots, m\}$ (Albini et al., 2021). This allocation is achieved by comparing the model's

output with the inclusion of a particular feature to the output without that feature (Albini et al., 2021).

A distinctive attribute of Shapley values is their equitable distribution of the difference between the model's prediction and the average prediction among the contributing features or players. This equitable distribution sets Shapley values apart from alternative methods like LIME, which do not provide this level of precise allocation (Aas et al., 2021). Shapley values can yield local explanations, which can be aggregated to generate global interpretations, such as feature-dependence plots, assessments of feature importance, and interaction analysis (Holzinger et al., 2022).

SHAP is a valuable tool for unravelling the intricacies of complex machine learning models. It facilitates the interpretation of a model's predictions by quantifying the contribution of each input feature to the model's outputs, thereby aiding in verifying the model's predictive accuracy (Kannangara et al., 2022).

The SHAP value for a feature is the average of its marginal contributions across all possible combinations of features. Mathematically, the SHAP value for feature i can be defined as follows (Lundberg and Lee, 2017):

$$SHAP_i = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|! (|N| - |S| - 1)!}{|N|!} (f(S \cup \{i\}) - f(S)). \quad (1)$$

The variables for the formula are described in Table 2-2.

Table 2-2 - SHAP Formula Variables

Variable	Description
i	The feature for which the SHAP value is calculated.
N	Set of all features.
S	Subset of any combination of all features.
$S \subseteq N \setminus \{i\}$	Subset of features without feature i .
$f(S)$	The prediction for the subset of features S .

$ S $	The cardinality of set S , representing the number of elements or features in set S .
$ N $	The cardinality of set N , representing the number of elements or features in set N .

The factorial terms in the SHAP value formula make sure that all features are considered equally.

Advantages

Shapley values provide consistent outcomes in the analysis of key features across algorithms and across evaluations with access to only the algorithm inputs and outputs (Smith and Alvarez, 2021). Features that do not contribute to the prediction receive an attribution of zero, thereby clearly distinguishing these features, aiding with the simplicity of understanding and interpretation (Holzinger et al., 2022). Shapley values derived from local interpretations can be aggregated to form global interpretations of the model, providing insights into feature dependence plots, feature importance, and interaction analysis (Holzinger et al., 2022).

In comparison to other XAI methods, Shapley values can measure the contribution of each explanatory feature for each prediction made by a machine learning model (Giudici and Raffinetti, 2021). Moreover, tools based on Shapley values are not limited by specific model classes or data types, allowing for a broad range of applications and the personalisation of results (Giudici and Raffinetti, 2021).

Shapley values are readily interpretable by domain experts who may not be familiar with machine learning terminology, thereby enhancing their applicability and implementation in real-world scenarios (Smith and Alvarez, 2021). Juscafresa (2022) demonstrated that SHAP offered superior consistency and theoretical robustness compared to LIME, particularly when evaluating accuracy and reliability. This superiority stems from SHAP's comprehensive consideration of all predictions, in contrast to LIME's methodology, which focuses on a local sample instance and assigned weights based on its proximity to the instance under scrutiny. This was based on empirical analysis, which utilised a Random Forest model to assess diabetes data, underscoring the comparative advantages of these two XAI methods.

Disadvantages

The primary disadvantage of Shapley values is their computational complexity, requiring substantial computational resources to calculate (Holzinger et al., 2022). This complexity increases exponentially with the number of features (Aas et al., 2021). Such demanding computations become particularly costly in problems with numerous features, as is often the case with DNNs (Saeed and Omlin, 2021). However, there are algorithm-specific implementations designed to mitigate this computational burden (Holzinger et al., 2022). Additionally, the effectiveness of the results may be compromised if the features used in an algorithm are highly interdependent, a situation common in real-world datasets (Aas et al., 2021).

2.10.6 Explainable AI and Finance – A Survey of Approaches and Current Research

Table 2-3 includes recent research papers that use explainable AI techniques in financial prediction. LIME and SHAP are the most popular among these techniques.

LIME's explanatory capabilities were used to enhance transparency in decision-making processes (Bandi et al., 2021, Gite et al., 2021). Additionally, LIME has been instrumental in improving features and understanding feature importance, a key application in the research (Carta et al., 2022; Ghosh et al., 2023).

Fior et al. (2022) utilised SHAP for its effective visualisations to explain algorithm variability, specifically through a dashboard designed for a domain expert. SHAP plots were used to explain actions of a deep reinforcement algorithm (Kumar et al., 2022). SHAP's ability to provide insights into feature importance was another notable application (Fior et al., 2022; Ghosh et al., 2023).

Ghosh et al. (2023) utilised both SHAP and LIME in their research, employing SHAP to interpret the global impact of features on the algorithm, and LIME to understand the importance of features at a local level.

Table 2-3 - Explainable AI and Financial Prediction

No.	Paper	Application	XAI Approach	Intended Outcome	XAI Criteria	XAI Results
1	Bandi et al., 2021	XAI tool to support decision making for share market index movement prediction	LIME is used for the interpretation of the news sentiment aspect	LIME is used for explanation of ‘bullish, neutral or bearish’ sentiment derived from the Random Forest classification algorithm output using news text	Simplify process of analysis using an interactive dashboard. News is represented as red and green circles which indicates bullish or bearish market sentiment.	XAI is used to help make decisions by transparently displaying reasons behind the prediction using a dashboard with insights. This could result in higher accuracy in decision making and efficiency by reducing time.
2	Carta et al., 2022	Automatic feature selection to support financial prediction of next-day share returns	Mean decrease impurity (MDI), LIME Comparing Mean Square Error versus base regressor by removing one feature at a time.	Separate important features from unimportant ones to improve prediction performance.	MSE and ratio of improvements over base regressor when a feature is removed	Three strategies to improve prediction performance using an optimal threshold.
3	Fior et al., 2022	XAI tool for cryptocurrency prediction	SHAP	Visual interface from which domain experts can infer actionable dependencies based on ML predictions and input features	SHAP series and summary plots to show temporal variations of the mean Shapley values associated with ML predictions	Tool was able to explain algorithm variability through selected features, feature subcategories and categories across 21 cryptocurrencies and 3 reference periods.
4	Gite et al., 2021	Share price prediction from sentiment analysis of financial news articles using LSTM and XAI	Algorithm Agnostic using LIME	Global understanding of the algorithm.	Algorithm Transparency and Feature Importance	Explanatory prediction can justify why an algorithm has predicted a given output using XAI tools.
5	Ghosh et al., 2023	Predicting daily NFT and DeFi prices, use XAI to interpret feature contribution	SHAP and LIME	Analyse feature contribution for underlying algorithms	Use XAI tools to provide meaningful insights	The overall feature contribution is not different between NFT and DeFi tokens.
6	Kumar et al., 2022	Explainable reinforcement learning on financial share trading using SHAP	SHAP used to explain prediction of maximum cumulative rewards via waterfall plots	SHAP applied alongside deep reinforcement learning to explain an agent’s action at a given instance of financial share trading	SHAP waterfall plots are compared to daily returns predictions using a deep Q network	SHAP plots can explain agent trading actions (buy and sells) when applied to a deep Q network for specific trading events.
7	Wand et al., 2022	XAI to reduce features to identify sectors dominating market states of S&P 500	Clustering and Layer-wise Relevance Propagation. Local feature	Identify sector correlations to a daily market state	Feature relevance score for correlation between sectors to all instances	Certain trading days are dominated by only a few sector correlations

relevance is
aggregated to
create global
feature
relevance

(market state) for
each trading day

2.11 Summary Findings

Previous research indicates that fundamental data can significantly assist in predicting share returns, achieving accuracy levels that justify its inclusion in machine learning datasets in finance (¹ 1, 9, 14, 17; ² 1, 2, 5). However, it is less used in studies with deep learning algorithms, partly due to the low frequency of reporting. This might explain the relative scarcity of research on fundamental data compared to technical data in this field (¹ 2-5, 7, 8, 10-13, 19, 20, 22, 24-30; ² 5).

Additionally, most studies in this area tend to be short-term (¹ 5, 9, 10, 13, 17, 22, 26, 27, 29). Long-term predictions offer the advantage of lower frictional costs, related to trading, which can erode profit potential (Huang, 2019). The meta-analysis also suggests that mixed datasets can improve accuracy (¹ 1; 8, 9, 14, 17-20, 26, 30; ² 1).

Careful consideration should be given to selecting the correct prediction window, as evidence indicates that fundamental data is most effective for long-term prediction (¹ 14; ² 1, 5). This effectiveness is largely influenced by the frequency of financial reporting updates, which should align with the prediction window (Shynkevich et al., 2017). In an international context, this translates to quarterly updates. However, in South Africa, the JSE mandates biannual reporting, which may necessitate adjustments in the prediction window for accuracy.

The evaluation metrics should include a diverse mix of criteria. Accuracy is the most popular metric among the typical classification measures. It is frequently complemented by return analysis, which is used for evaluating financial models (¹ 2, 4, 5, 8, 11, 17, 18, 22, 24, 26, 28, 30; ² 2, 5). Additionally, using baseline algorithms as a point of comparison has proven useful (Huang, 2019).

Classification algorithms have proven not only useful, but also more accurate than regression-style approaches in predicting share return and price trend (Drobetz and Otto, 2020; Sezer et al., 2020). In the context of broader portfolio construction, these algorithms offer the advantage of

¹ Appendix A, Table A1. Refer to the corresponding paper number for details.

² Table 2-1. Refer to the corresponding paper number for details.

providing categorical rankings. This simplifies the portfolio creation process by eliminating an additional step typically required for sorting and ranking, thereby streamlining the problem statement and potential solution steps (Drobetz and Otto, 2020). The majority of research in this area has focused on simple categorical rankings, often binary or tercile. These classes are frequently labelled as ‘up and down’ or ‘buy, sell and hold’ (¹2-5, 8, 10, 11, 13, 19, 20, 22-24, 26-30; ²1, 5).

Popular algorithms for predicting share returns using classification and fundamental data include Multilayer Perceptron Neural Networks (as Artificial or Deep Neural Networks), AdaBoost, GradBoost, Random Forest, Decision Trees, Logistic Regression, Linear Regression, K-Nearest Neighbour and Support Vector Machines (¹1, 9, 12; ²1, 2, 4, 5). It is also evident that no single algorithm consistently yields the best results across similar datasets. Therefore, conducting comparative studies could provide deeper insights into this domain.

Machine Learning prediction models in emerging and developing markets often demonstrate greater accuracy than those in developed countries (Bustos and Pomares-Quimbaya, 2020). The inherent randomness and complexity of developed markets might limit the creation of accurate models (Bustos and Pomares-Quimbaya, 2020). In contrast, the larger inefficiencies in emerging market, such as South Africa, could lead to higher accuracy in prediction models that capitalise on these inefficiencies. This phenomenon might challenge the Efficient Market Hypothesis in emerging and developing markets, where high prediction accuracy suggests that not all available information is reflected in share prices.

XAI techniques can offer additional insights into the performance and understanding of algorithms (Ghosh et al., 2023). They aid in determining the relative importance of input features in an algorithm’s predictive outcomes and provide a consistent basis for comparison across different algorithms by evaluating their input and outputs. SHAP, in particular, is considered a robust technique for use in financial services (Ghosh et al., 2023).

Lastly, there appears to be a lack of published research conducting a comparative analysis of classification algorithms using fundamental data in the South African context. Undertaking such research could yield new insights and contribute significantly to this field, especially if state-of-the-art machine learning classification algorithms and XAI techniques are applied.

3. Experimental Design and Execution

This chapter presents the approach, design and implementation of the experiments performed. It starts by providing an overview of the dataset. The second section discusses the experimental process, encompassing the classification task, machine learning problem and pipeline, experiments conducted and the tools used during the implementation, including overviews and workflow diagrams. The third section delves into the evaluation metrics. The fourth section details the data preprocessing aspects. The fifth section examines the development of each of the eight selected algorithms. Finally, a summary of the contents is provided.

3.1 Dataset

This section introduces the dataset used for the experiments, including its scope, an overview of data, the sourcing methods and identified limitations and biases.

3.1.1 Scope

The scope is limited to shares listed on the Johannesburg Stock Exchange (JSE). The number of shares has been constrained to the FTSE/JSE All Share Index, the constituents of which change over time. The input features used consist of fundamental ratios derived from financial statements, such as the income statement, balance sheet and cash flow statement.

3.1.2 Overview

The data was sourced via FactSet, a provider of market data and investment decision-making solutions. This included semi-annual, end-of-week financial data for the 143 shares listed on the FTSE/JSE All Share Index (ALSI) as of the 31st December 2021. This dataset represents a retrospective extract for the period 1st January 2000 to 31st December 2021, comprising 22 years and at most 44 semi-annual observations per share. The data includes share identifiers, dates, closing prices and fundamental ratios derived from the financial statements, which are a required disclosure item for a share to be listed on the JSE. These fundamental ratios are part of the broader ‘FactSet Fundamentals’ dataset and are grouped into ‘Valuation, Profitability and Liquidity’. The number of fundamental ratios is three for liquidity, eleven for profitability, and seven for valuation, making a total of twenty-one ratios. A detailed description of each of these data elements is provided in Table 3-1. For a summary view of the dataset before and after preprocessing, see Table 3-7.

A computer equipped with FactSet Workstation software, including proprietary Microsoft Excel add-ins, facilitated programmatic extraction of the required data. This was achieved using Excel formulas and the FactSet ‘Quick Report’ feature, specifically focusing on data elements related to

the universe of shares limited to the FTSE/JSE ALSI. The data extracts, downloaded in batches, were compiled into a single .csv file using a custom-built Python function.

The historic data availability for the FTSE/JSE ALSI varied during this period, due to factors such as new listings, delistings or mergers. Appendix B, Table B1, presents the complete list of FTSE/JSE ALSI shares, including their full share name, sector and sub-sector (with numerical codes). Figure 3-1 offers a summary view of the shares, grouped per sector.

Table 3-1- Data Elements and Detailed Descriptions

No.	Element	Descriptions	Category
1	Share Identifier	Returns the ticker, SEDOL, CUSIP, ISIN or FactSet identifier for a share.	Reference Data
2	Date	The date used for retrieval of the closing price	Date
3	Close Price	The daily price as of the date(s) requested. By default, the price is in local trading currency, split adjusted and not spinoff adjusted. The price change between closing prices was used to calculate a rolling 12 month share return.	Pricing
4	Enterprise Value to EBIT	Represents Enterprise Value to Earnings before Income and Tax (EBIT) for the period and date(s) requested. Returns values only for Commercial and Other Financial companies. If data is requested on a quarterly or semi-annual frequency, the formula uses the last twelve months' value for EBIT (Operating Income)	Valuation
5	Enterprise Value to EBITDA	Represents Enterprise Value to Earnings before Interest, Tax, Depreciation and Amortisation (EBITDA) for the period and date(s) requested. Returns values only for Commercial and Other Financial companies. If data is requested on a quarterly or semi-annual frequency, the formula uses the last twelve months' value for EBITDA (Operating Income plus Depreciation and Amortisation).	Valuation
6	Enterprise Value to Sales	Represents enterprise value to sales for the period and date(s) requested. This is calculated as Enterprise Value divided by Net Sales or Revenue. If data requested is on a quarterly or semi-annual frequency, the formula will use the last twelve months' value of Net Sales.	Valuation

No.	Element	Descriptions	Category
7	Price to Earnings -	Represents price to earnings per share for the period and date(s) requested. This is calculated as Closing Price divided by Earnings Per Share for the Fiscal Period. For quarterly or semi-annual periodicities, this formula uses a 12-month Earnings Per Share value. If the denominator is negative, the formula will return an N/A.	Valuation
8	Price to Sales	Represents price to sales or revenue using price as of the last trading day for the period and date(s) requested in local currency by default. This is calculated as Closing Price divided by Sales Per Share. If the denominator is negative, the formula will return an N/A. If data is requested on a quarterly or semi-annual frequency, the formula uses the last twelve months' value for Net Sales or Revenue.	Valuation
9	Price to Book Value	Represents price to book value using price as of the last trading day for the period and date(s) requested in local currency by default. This is calculated as Closing Price divided by Book Value Per Share. If the denominator is negative, the formula will return an N/A.	Valuation
10	Dividend Yield	Represents the dividend yield for the period and date(s) requested in local currency by default. This is calculated as Dividends Per Share divided by Closing Price, multiplied by 100. If data is requested on a quarterly or semi-annual frequency, the formula uses the last twelve months' value for Dividends Per Share (FF_DPS).	Valuation

No.	Element	Descriptions	Category
11	Return on Average Total Equity	Represents return on average total equity for the period and date(s) requested in local currency by default. This is calculated as Net Income divided by the two fiscal period average of Total Shareholders' Equity, multiplied by 100. If data is requested on a quarterly or semi-annual frequency, the formula uses the last twelve months' value for Net Income. If the numerator is negative, the formula will return an N/A.	Profitability
12	Return on Average Assets	Represents return on average assets for the period and date(s) requested in local currency by default. This is calculated as Net Income divided by the two fiscal period average of Total Assets. If you request data on a quarterly or semi-annual frequency, the formula uses the last twelve months' value for Net Income. If the numerator is negative, the formula will return an N/A.	Profitability
13	EBITDA Margin	Returns the EBITDA margin for the period and date(s) requested. This is calculated as EBITDA divided by Net Sales, multiplied by 100. If the numerator is negative, the formula will return an N/A.	Profitability
14	EBIT Margin	Returns the EBIT margin for the period and date(s) requested. This is calculated as EBIT - Operating Income divided by Net Sales, multiplied by 100. If the denominator is negative, the formula will return an N/A.	Profitability
15	Gross Income Margin	Represents gross income margin for the period and date(s) requested. This is calculated as Gross Income divided by Net Sales or Revenue,	Profitability

No.	Element	Descriptions	Category
		multiplied by 100. If the numerator is negative, the formula will return an N/A.	
16	Net Income Margin	Represents the net income margin for the period and date(s) requested. This is calculated as Net Income divided by Net Sales or Revenue, multiplied by 100. If the numerator is negative, the formula will return an N/A.	Profitability
17	Pretax Margin	Represents pretax margin for the period and date(s) requested. This is calculated as Pretax Income divided by Net Sales or Revenue, multiplied by 100. If the numerator is negative, the formula will return an N/A.	Profitability
18	Dividend Payout Ratio	Represents the dividend payout ratio for the period and date(s) requested. This is calculated as Dividends Per Share divided by Earnings Per Share for the Fiscal Period, multiplied by 100. If the denominator is negative, the formula will return an N/A.	Profitability
19	Operating Margin	Represents operating margin for the period and date(s) requested. This is calculated as Operating Income divided by Net Sales or Revenue, multiplied by 100. If the numerator is negative, the formula will return an N/A.	Profitability
20	Return on Average Total Capital	Represents the return on average total capital for the period and date(s) requested. This is calculated as Operating Income divided by the sum of two fiscal period average of Total Capital Including Short-term Debt and two fiscal period average of Accumulated Minority Interest, multiplied by 100. If the numerator is negative, the formula will return an N/A. If you	Profitability

No.	Element	Descriptions	Category
		request data on a quarterly or semi-annual frequency, this formula uses the last twelve months' value for Operating Income.	
21	Sales Per Share	Represents sales per share for the period and date(s) requested in local currency by default. This is calculated as Net Sales or Revenue divided by Common Shares Used to Calculate Fully Diluted Earnings Per Share.	Profitability
22	Current Ratio	Represents the current ratio for the period and date(s) requested in local currency by default. This is calculated as Total Current Assets divided by Total Current Liabilities.	Liquidity
23	Quick Ratio	Represents the quick ratio for the period and date(s) requested. This is calculated as Current Assets - Total and Inventories, divided by Current Liabilities. If the denominator is negative, the formula will return an N/A.	Liquidity
24	Total Debt to Equity	Represents total debt as a percent of equity for the period and date(s) requested in local currency by default. This is calculated as Total Debt divided by Total Shareholders' Equity, multiplied by 100.	Liquidity

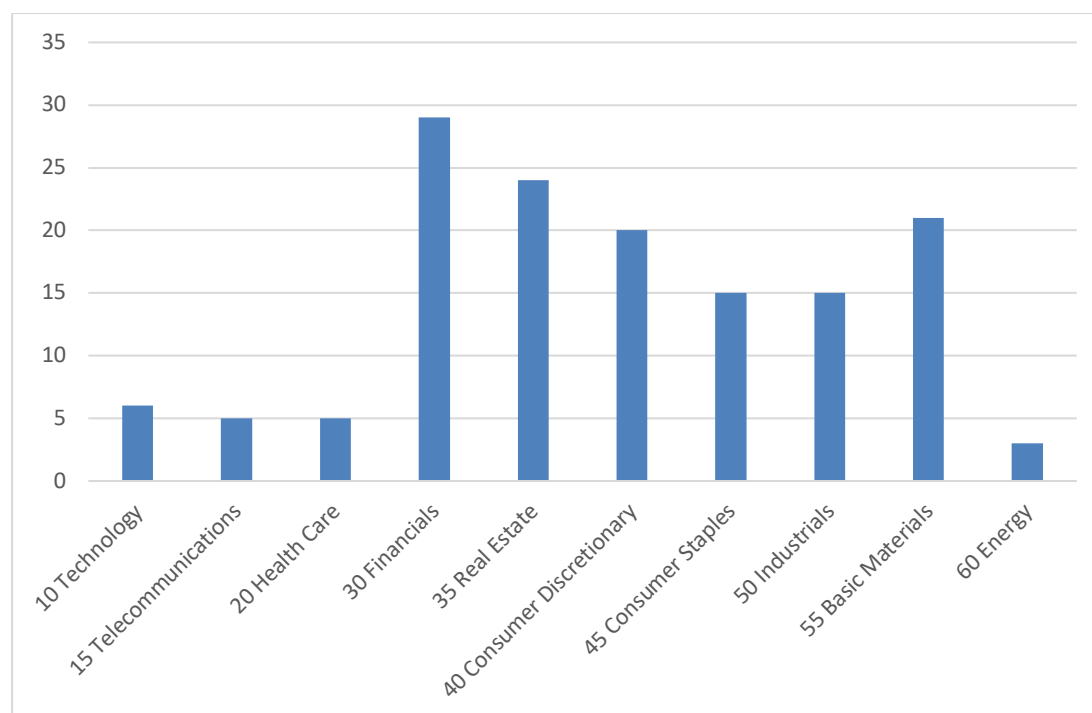


Figure 3-1 - Summary of FTSE/JSE ALSI Shares per Sector

3.1.3 Limitations

Every attempt was made to ensure a complete dataset, but this could be hampered by the availability of such data. The veracity of data can be affected by the following errors, given that a single provider was used as the source:

1. Capture or sourcing errors when compiling the dataset by the source provider.
2. Some metrics will be missing if the underlying financial statements do not support the capture of these values.
3. Several ratios are calculated using formulas that in the long-term could change with the changing needs of the investment market.
4. Changes in accounting standards might result in the reclassification of items, making comparison difficult or impossible.

The experiments performed differ from real-world scenarios in that they do not factor in trading costs. These costs include, but are not limited to, Securities Transfer Tax (STT), STRATE Trading Fees, Financial Services Board (FSB) Investor Protection Levies, Custodian Holding costs and VAT. Additional considerations include the feasibility of trading large volumes of shares at a single closing price, which is highly unlikely when rebalancing a portfolio, particularly for shares subject to varying liquidity constraints.

3.1.4 Biases

Two types of biases have been identified: Survivorship and Look-Ahead bias. Survivorship bias relates to shares that fail to remain in the dataset for the full duration of the experiment, as opposed to those that do. This will affect the outcome of the experiments since these shares are excluded for certain periods.

Look-Ahead bias pertains to the availability of financial information and its dissemination to the broader market. The JSE stipulates that financial statements should be made available within three months after the financial year-end. This data could be accessible to a selected few before the official release, potentially influencing the share price. However, such premature access is expressly prohibited and could result in sanctions against any individual or entity found guilty of this type of transgression.

3.2 Experimental Process

This section outlines the experimental process, including the classification prediction task, machine learning problem and pipeline, experiments performed and the implementation aspects.

3.2.1 Classification Prediction Task, Machine Learning Problem and Pipeline

The first experiment identifies the optimal hyperparameters using a Grid Search technique, which was employed to select the parameters for the eight algorithms, ensuring the best possible results. The performance of these algorithms is then compared and evaluated. In the second experiment, the top three performing algorithms are selected, and additional metrics are incorporated to further assess their performance. These algorithms are also analysed to identify any patterns or perturbations specific to sectors or shares. In the third experiment, SHAP Summary Plots are utilised to explore the interpretability of the selected models.

The study addresses the following specific questions:

1. Which classification algorithms are the most effective in predicting tercile class of 12-month share returns?
2. Did any sector-specific or share-specific patterns or perturbations provide additional insights into the performance of the algorithms?
3. How does feature importance impact the predictive models and how does this contribute to the interpretability of the models?
4. What insights into interpretability of the predicted tercile class can be gained from the SHAP Summary Plots?

3.2.1.1 Classification Prediction Task

The machine learning task involves predicting the tercile class of the 12-month share returns using fundamental ratio data. For this purpose, a tercile ranking system is employed, wherein each tercile in the dataset is assigned a numerical class from 1 to 3, with 1 being the highest ranking. For a detailed list of the input features, refer to Section 3.2.2.1 and the summary in Table 3-7. It includes sixteen input features and one target variable, labelled as either 1, 2, or 3.

The machine learning problem involves using selected classification algorithms to predict the tercile class of each share's 12-month return. This constitutes a supervised machine learning problem, for which eight classification-based algorithms are used. These algorithms are compared

based on their relative efficacy in predicting the target variable, with the evaluation metrics for this comparison described in Section 3.3.

3.2.1.2 Machine Learning Pipeline

Three experiments are conducted on the dataset, as depicted in Figure 3-2, which provides an abstract overview of the experimental process.

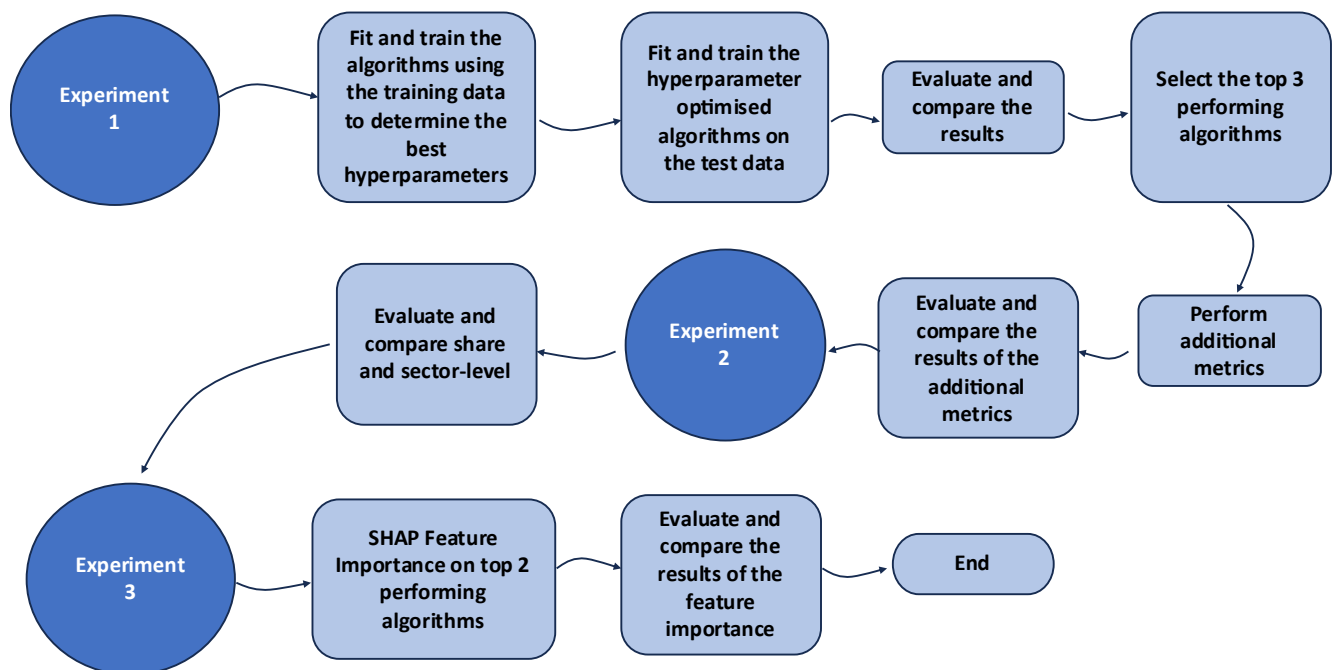


Figure 3-2 - Long-term Share Prediction and Evaluation Workflow

3.2.2 Experiments

This section details each of the three experiments performed, including their purpose and the intended outcomes for evaluation.

3.2.2.1 Experiment 1– Comparison of Machine Learning Algorithms

The purpose of this experiment was twofold. First, to identify the best hyperparameter settings for each algorithm, using the training dataset with 5-fold cross validation. Subsequently, the experiment compared each of the eight optimally configured algorithms based on the evaluation metrics. This comparison utilised the hyperparameters determined in the initial phase and was conducted against the test dataset. Additionally, a Uniform Baseline Classifier was included for comparative purposes. Furthermore, a detailed analysis of the top three performing algorithms was conducted. This analysis focused on investigating their predictive accuracy, alongside additional evaluation metrics including Confusion Matrix, precision, recall and F1 score per tercile class. The input features used were obtained after completing the data preprocessing steps described in Section 3.4.1.

The following input features and target variable were used for this experiment:

Input Features

1. Enterprise Value to EBIT
2. Enterprise Value to Sales
3. Price to Earnings
4. Price to Sales
5. Price to Book
6. Dividend Yield
7. Return on Average Total Equity
8. Return on Average Assets
9. EBIT Margin
10. Net Income Margin
11. Pre-tax Margin
12. Dividend Payout Ratio
13. Operating Margin
14. Return on Average Total Capital
15. Sales Per Share
16. Total Debt Percentage Equity

Target Variable

1. Tercile Class of 12-Month Share Return, labelled as either 1, 2, or 3.

3.2.2.2 Experiment 2 – Comparison of Sector and Share-level Performance

The purpose of this experiment to uncover nuances at the share and sector-level in the predictive accuracy of the top performing algorithms. The analysis involved predicting the tercile class for each of the three algorithms, using the same test dataset. The resulting datasets included share identifiers, which were then matched to their respective high-level sectors for further investigation. The ten sectors in Table 3-2 were matched to the underlying share identifiers for analysis. The detailed mapping of shares to sectors is provided in Appendix B, Table B1.

Table 3-2 - High-level Sector Names

Sector Description
10 Technology
15 Telecommunications
20 Health Care
30 Financials
35 Real Estate
40 Consumer Discretionary
45 Consumer Staples
50 Industrials
55 Basic Materials
60 Energy

3.2.2.3 Experiment 3 – SHAP and Feature Importance

SHAP was applied to two of the top performing algorithms at a global level. This approach aimed to explore feature importance using SHAP Summary Plots, thereby enhancing the interpretability of the algorithms and providing insights into the relative impact of the features on the tercile classifications.

The analysis encompassed the three categories of fundamental ratios: Valuation, Profitability and Liquidity, as detailed in Table 3-3. The analysis was conducted in the context of this information.

Table 3-3- Categories of Fundamental Ratios

Input Features/Fundamental Ratios	Category
Price to Book Dividend Yield Enterprise Value to EBIT Enterprise Value to Sales Price to Earnings Price to Sales	VALUATION
Dividend Payout Ratio EBIT Margin Net Income Margin Operating Margin Pretax Margin Return on Average Assets Return on Average Total Capital Return on Average Total Equity Sales Per Share	PROFITABILITY
Total Debt % Equity	LIQUIDITY

3.2.3 Implementation

This section describes the tools, technologies, process and workflow of the experiments that were performed.

3.2.3.1 Jupyter and Colab Notebooks

The initial data extraction was conducted using a desktop installation of Python version 3. All experiments were performed using Jupyter via Google's Cloud platform (Colab Notebooks using Python version 3) to facilitate shared learning and simplicity of understanding, exposing each step with accompanying descriptions. The use of cloud processing notably improved the runtimes for resource intensive tasks such as hyperparameter optimisation. Numerous libraries were employed for machine learning, data preprocessing, analysis and visualisation. These include, but are not limited to, the following: Scikit-learn, Keras (TensorFlow), Pandas and Seaborn.

The detailed data preprocessing steps have been extensively covered in Section 3.4.

The steps performed in Experiment 1 followed the sequence depicted in Figure 3-3. The detailed implementation workflow for both experiments is represented in Figure 3-4.

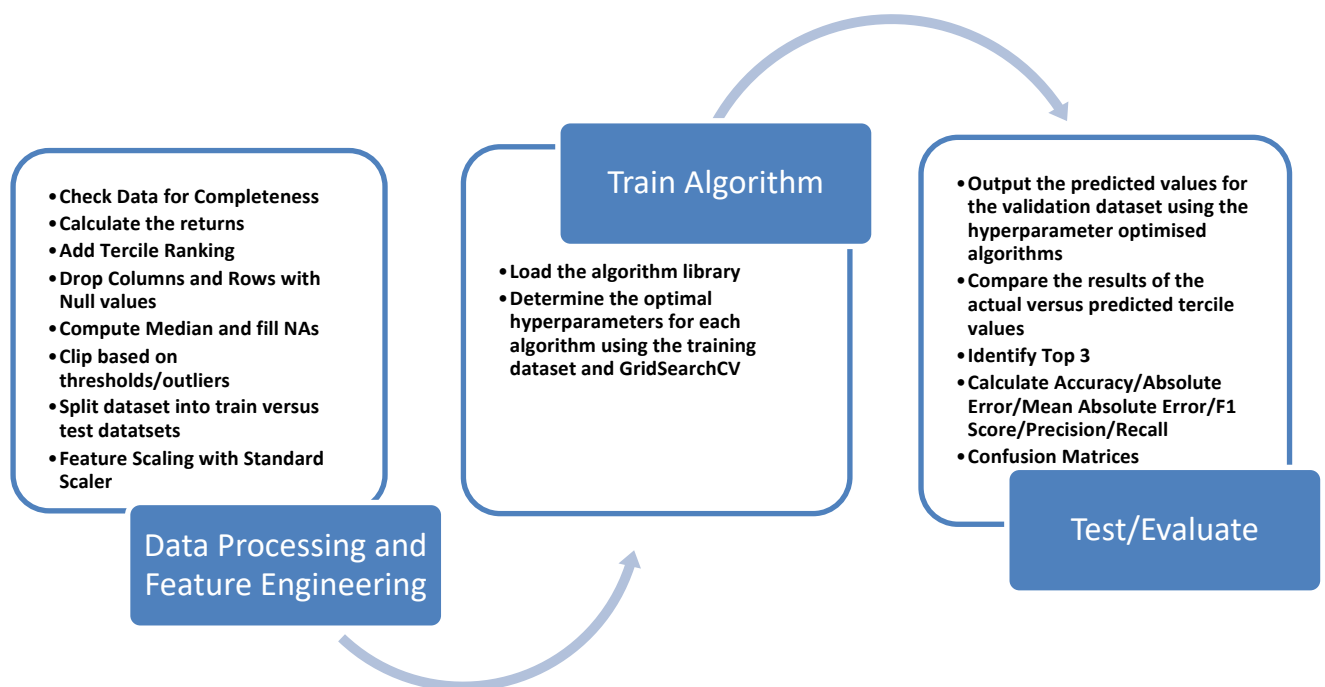


Figure 3-3 - Experimental Design Overview for Experiment 1

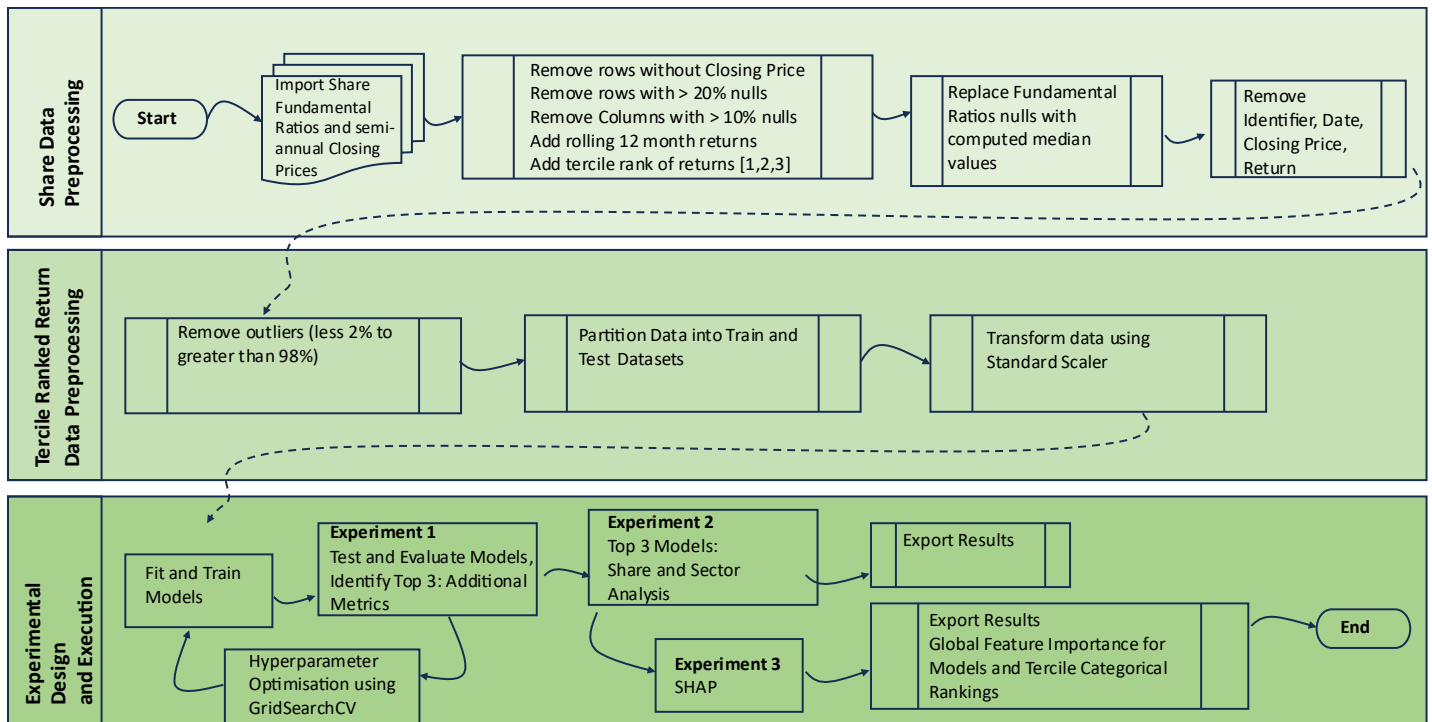


Figure 3-4 - Implementation Workflow for Experiments 1, 2 and 3

3.3 Evaluation

This section describes the data partitioning and validation approach, including the k-fold cross-validation approach. It covers the evaluation metrics used, such as accuracy, precision, recall, F1 score, mean absolute error and absolute error. Additionally, it discusses the use of a baseline classifier and model interpretability through feature importance analysis using SHAP Summary Plots.

3.3.1. Data Partitioning and Validation Approach

All algorithms were trained and tested against the same dataset, as described in Section 3.1 and Section 3.4. The dataset was divided into an 80% training set and a 20% test set. This split was achieved using a NumPy randomisation function, setting a consistent seed value of 42 to ensure repeatability. Once the training for each algorithm was complete and the best hyperparameters determined, the same test data was used to predict the tercile class of each share's 12-month return, with the results then being used to compare the performance of the various algorithms.

K-fold cross-validation was employed for hyperparameter optimisation of all algorithms, utilising the GridSearchCV object from scikit-learn for systematic tuning. Specifically, five-fold cross-validation was applied, setting the 'cv' parameter of GridSearchCV to five. This method entails dividing the entire training data into five equal-sized folds. In each iteration of the cross-

validation, one fold is used as the validation set, while the remaining folds serve as the training set. This process ensures that each fold is used once as the validation set across the five iterations. The performance of each algorithm is assessed in every iteration, with an average performance metric computed at the end. This comprehensive approach not only increases the training iterations for each algorithm, but also enhanced the accuracy determination during hyperparameter optimisation (Lee, 2020). The equal distribution of the training data across each fold is depicted in Figure 3-5, where the blue fold represents the validation dataset in each iteration.

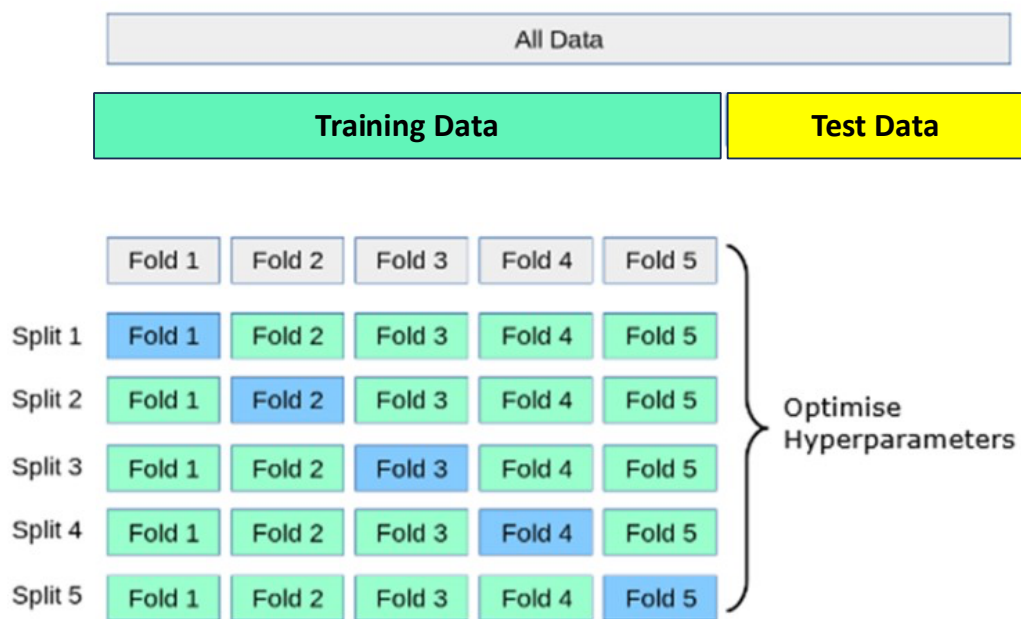


Figure 3-5 – Five-fold Cross Validation using GridSearchCV

The evaluation metrics, selected based on findings from the Literature Review, included accuracy, precision, recall, F1 score and mean absolute error. Additionally, absolute error was included as an extra metric to provide further insight. These metrics were applied to the test dataset to compare actual versus predicted tercile class for each share. The respective results were then recorded and analysed to evaluate and determine the performance of the algorithm. The algorithm with the best performance was determined based on its ability to predict the tercile class of the 12-month share return with the highest accuracy on the test dataset.

3.3.2 Evaluation Metrics

For this study, Absolute Error is the absolute value of the difference between actual and predicted tercile class. Although not used extensively in the literature, it provides a useful metric when comparing the different algorithms.

The mathematical formula for Absolute Error is as follows:

$$Absolute\ Error = \sum_{i=1}^n |Y_i - \hat{Y}_i|. \quad (2)$$

Table 3-4 - Absolute Error and Mean Absolute Error Formulae Variables

Variable	Description
n	Total number of observations or data points.
i	Index variable that iterates through each data point from 1 to n .
Y_i	Actual value for the i -th data point.
$\hat{Y}_i$	Predicted value for the i -th data point.

The Absolute Error was computed for each of the experiments on the test dataset using the scikit-learn Mean Absolute Error function and dividing this by the number of inputs (n) for each experiment.

When there are many outliers, the Mean Absolute Error (MAE), also known as the Average Absolute Deviation, is preferred (Géron, 2017). This method emphasizes larger errors by averaging the absolute differences between predicted and actual values, while smaller errors have less impact on the overall metric (Géron, 2017).

The mathematical formula for Mean Absolute Error is as follows:

$$Mean\ Absolute\ Error = \frac{1}{n} \sum_{i=1}^n |Y_i - \hat{Y}_i|. \quad (3)$$

The scikit-learn Mean Absoluter Error function was used for each of the experiments on the test dataset.

Accuracy is typically defined as the ratio of correct predictions to the total number of predictions and is commonly used for classification tasks (Géron, 2017). However, it may not be the preferred performance measure with skewed datasets (Géron, 2017). In this context, 12-month share returns are based on tercile ranking, which does not exhibit skewness. Nevertheless, accuracy does not reflect the relative degree of profitability. In this context, this could still result in profit from a portfolio construction perspective, i.e. shares ranked in the 1st and 2nd tercile class are likely to outperform those in the 3rd ranked tercile class.

Confusion Matrix		Actual Class	
		Class A	Not Class A
Predicted Class	Class A	True Positive	False Positive
	Not Class A	False Negative	True Negative

Figure 3-6 - Generic Confusion Matrix

The mathematical formula for Accuracy is as follows:

$$Accuracy = \left(\frac{True\ Positive + True\ Negative}{True\ Negative + True\ Positive + False\ Positive + False\ Negative} \right) \times 100. \quad (4)$$

The formula is based on the use of a confusion matrix, as seen in Figure 3-6.

Table 3-5 – Accuracy, Precision and Recall Formulae Variables

Variable	Description
<i>True Positive</i>	Data points correctly labelled as ‘Class A’.
<i>True Negative</i>	Data points that are correctly labelled as ‘Not Class A’.
<i>False Positive</i>	Data points incorrectly labelled as ‘Class A’.
<i>False Negative</i>	Data points incorrectly labelled as ‘Not Class A’.

Precision measures the accuracy of positive predictions and is calculated as the ratio of true positives to the sum of true positives and false positives (Nasa and Suman, 2012).

The mathematical formula for Precision is as follows:

$$Precision = \left(\frac{True\ Positive}{True\ Positive + False\ Positive} \right). \quad (5)$$

Recall, also known as sensitivity, measures the completeness of positive predictions. It is calculated as the ratio of true positives to the sum of true positives and false negatives (Nasa and Suman, 2012).

The mathematical formula for Recall is as follows:

$$Recall = \left(\frac{True\ Positive}{True\ Positive + False\ Negatives} \right). \quad (6)$$

The F1 Score measures a model's ability to accurately identify relevant information and is calculated as the harmonic mean of both precision and recall (Nasa and Suman, 2012).

The mathematical formula for F1 Score is as follows:

$$F1\ score = 2 \times \left(\frac{Precision \times Recall}{Precision + Recall} \right). \quad (7)$$

Table 3-6 - F1 Score Formula Variables

Variable	Description
<i>Precision (equation 5)</i>	Ratio of <i>True Positives</i> to the sum of <i>True Positives</i> and <i>False Positives</i> .
<i>Recall (equation 6)</i>	Ratio of <i>True Positives</i> to the sum of <i>True Positives</i> and <i>False Negatives</i> .

The literature reviewed in Section 2.9.3 (Ballings et al., 2015) identifies the usefulness of a baseline classifier for comparative purposes. Such a classifier represents each target class uniformly at random in the available dataset, disregarding all the input features. For this purpose, the scikit-learn DummyClassifier class was employed, utilising the ‘Uniform’ strategy and setting the random state to 42.

Feature importance refers to the relative influence of a given input feature to the predictions made by a model. It is crucial for understanding which input features significantly contribute to improving the model’s performance (Hooker et al., 2019). This aspect was analysed in more detail using SHAP, as discussed in the reviewed literature in Section 2.10.5.

3.4 Data Preprocessing

This section describes the data preprocessing steps performed, including aspects related to feature engineering, management of outliers, feature scaling, label encoding and one-hot encoding.

3.4.1 Data Preprocessing Steps

An initial completeness check of the data was performed using a heatmap to identify any missing values; see Appendix B, Figure B2. This showed that significant data was missing for the five ratios: Enterprise Value to EBITDA, Gross Income Margin, EBITDA Margin, Current Ratio and Quick Ratio. The rolling 12-month returns were calculated based on the semi-annual closing price data. All rows that did not have a closing price were removed from the dataset. Rows missing 20% or more of the required financial ratios were excluded from the final dataset, reducing the number of shares from the initial 143 down to 124. A corresponding heatmap of the dataset after this exclusion is presented in Appendix B, Figure B3. Furthermore, columns representing the financial ratios were filtered to only include those with a maximum of 10% missing data. The heatmap depicting this further exclusion is in Appendix B, Figure B4.

This filtering process reduced the available fundamental ratios down from 21 to 16 excluding the following:

1. Quick Ratio (Liquidity)
2. Current Ratio (Liquidity)
3. EBITDA Margin (Profitability)
4. Gross Income Margin (Profitability)
5. Enterprise Value to EBITDA (Valuation)

The median value for each financial ratio was computed and the null values were then subsequently replaced with these medians. The final heatmap, illustrating the culmination of preprocessing changes made to the underlying dataset, is presented in Appendix B, Figure B5. See Table 3-7 for a summary view of the dataset before and after data preprocessing.

The target variable for the experiments is the tercile class of the 12-month return for each share. Tercile ranking, a specific form of quantile ranking, divides the data into three equal parts. In this study's context, the ordered distribution of 12-month rolling returns has been segmented into three equal classes.

Table 3-7 - Summary of Dataset Before and After Preprocessing

	Raw Dataset	Clean Dataset	Training	Test
Rows	6291	2911	2328	583
Input Features/Fundamental Ratios	21	16	16	16
Number of Shares	143	124	123	113
Date - Start	1-Jan-00	31-Dec-00	31-Dec-00	30-Jun-02
Date - End	31-Dec-21	31-May-21	31-May-21	31-Dec-20
Actual Rows of Data per Share (maximum)	N/A	35	35	11
Theoretical Rows of Data per Share (maximum)	44	41	41	37

3.4.2 Feature Engineering

Feature engineering is broadly defined as the process of curating information and identifying input features that are likely to correlate well with share performance (Lee, 2020). It is considered a critical component to the success of any machine learning project (Geron, 2019).

In this study, the dates, share identifiers and the calculated values for 12-month returns have been removed from the dataset. This removal is intentional to align with the study's objective, which is to use fundamental ratio data for predicting long-term share returns.

3.4.3 Managing Outliers

Managing outliers is a crucial step in the process, as financial models can be highly sensitive to the presence of even a small number of outliers (de Prado, 2020). Each fundamental ratio was evaluated for outliers and adjustments were made using a custom Python function named 'clipRatios'. This function utilised the clipping functionality of the Pandas DataFrame, modifying the boundary values to ensure they remained within the threshold of 2nd and 98th percentile thresholds for each fundamental ratio. The impact of these percentile-based adjustments can be observed in Appendix B, Figure B6.

3.4.4 Feature Scaling

Feature Scaling is one of the most crucial transformations applied to data in experimentation, particularly due to the varying numerical scales of input data (Geron, 2019). In the experiments, feature scaling was performed by normalising the data using the Standard Scaler from scikit-learn. This step is a common requirement in machine learning to ensure the appropriate algorithm performance and is typically done after splitting the data into training and test sets. Appendix B, Figure B7, illustrates the adjustments made to accommodate the feature scaling transformations using the Standard Scaler.

3.4.5 Label Encoding

Label encoding is the process of converting text into numerical values, as most algorithms require numerical inputs for their calculations (Yadav, 2019). In this case, the tercile categories chosen do not require categorical encoding, as they are already represented as numerical labels.

3.4.6 One-Hot Encoding

This process involves converting integer categorical values into one-hot binary vectors. The objective is to ensure that neither nearby nor distant values are mistakenly interpreted as having ordinal significance (Geron, 2019). This transformation results in a matrix of ones and zeros, representing the various categories, thereby preventing the misrepresentation of numerical values by the algorithms as indicative of a hierarchy or order (Yadav, 2019). One-hot encoding was applied to the terciles as required by the relevant experiments.

The ‘dummy variable trap’ occurs when two or more dummy variables created through one-hot encoding are multicollinear, meaning one variable can be predicted from the others (Karabiber, 2021).

3.5 Algorithms

This section details the approach adopted for implementing the selected algorithms in this study. The algorithms - Random Forest, Gradient Boosting, XGBoost, AdaBoost, Decision Tree Classifier, Support Vector Machines, Neural Networks (including Artificial Neural Networks and Deep Neural Networks) and Logistic Regression - were identified in related studies in Section 2.7.

3.5.1 Random Forest

Random Forest is the first of the ensemble learning techniques used in this study. Appendix A2, titled ‘Random Forest Classifier: Parameters’, lists the full set of available parameters for this algorithm. The applied Grid Search parameters are as follows, with any missing parameters set to their defaults. Additionally, a 5-fold cross-validation was utilised on the training data:

Table 3-8 - Random Forest Grid Search Parameters

Grid Search Parameters	Value(s)
n_estimators	[50, 100, 500]
criterion	[gini, entropy]
max_depth	[2, 5, 10, 50, None]
min_samples_split	[2, 8, 16, 32]

min_samples_leaf	[1, 2, 4, 8, 16]
max_features	[auto, sqrt, log2]
random_state	[42]
scoring	[accuracy]

The total number of fits performed were 9000 using 5-folds for each of the 1800 candidates.

The Grid Search process identified the optimal model with the following hyperparameters.

Table 3-9 - Random Forest Optimal Model

Parameters	Value
n_estimators	[100]
criterion	[entropy]
max_depth	[50]
min_samples_split	[8]
min_samples_leaf	[1]
max_features	[auto]
random_state	[42]

3.5.2 Gradient Boosting

Appendix A2, titled ‘GradBoost Classifier: Parameters’, lists the full set of available parameters for this algorithm. The applied Grid Search parameters are as follows, with any missing parameters set to their defaults. Additionally, a 5-fold cross-validation was utilised on the training data:

Table 3-10 - GradBoost Grid Search Hyperparameters

Grid Search Parameters	Value(s)
loss	[deviance]
learning_rate	[0.1]
n_estimators	[100, 200]
criterion	[friedman_mse, mse]
max_depth	[3, 10]
max_features	[auto]

Grid Search Parameters	Value(s)
random_state	[42]
scoring	[accuracy]

The total number of fits performed were 40 using 5-folds for each of 8 candidates.

The Grid Search process identified the optimal model with the following hyperparameters.

Table 3-11 - GradBoost Optimal Model

Parameters	Value(s)
loss	[deviance]
learning_rate	[0.1]
n_estimators	[100]
criterion	[friedman_mse]
max_depth	[10]
max_features	[auto]
random_state	[42]
scoring	[accuracy]

3.5.3 XGBoost

Appendix A2, titled ‘XGBoost: Parameters’, lists the full set of available parameters for this algorithm. The applied Grid Search parameters are as follows, with any missing parameters set to their defaults. Additionally, a 5-fold cross-validation was utilised on the training data:

Table 3-12 - XGBoost Grid Search Parameters

Grid Search Parameters	Value(s)
Objective	[multi: softmax, multi: softprob]
learning_rate	[0.10, 0.01]
n_estimators	[100, 200]
Max_depth	[3, 50]
min_child_weight	[1, 5]
Gamma	[0.1, 0.4]
colsample_bytree	[0.3, 0.7]

Grid Search Parameters	Value(s)
eval_metric	[mae]
random_state	[42]
Scoring	[accuracy]

The total number of fits performed were 320 using 5-folds for each of 64 candidates.

The Grid Search process identified the optimal model with the following hyperparameters.

Table 3-13 - XGBoost Optimal Model

Parameters	Value
Objective	[multi: softprob]
learning_rate	[0.10]
n_estimators	[200]
Max_depth	[50]
min_child_weight	[1]
Gamma	[0.1]
colsample_bytree	[0.7]
eval_metric	[mae]
random_state	[42]
Scoring	[accuracy]

XGBoost has been chosen as the second ensemble technique otherwise referred to as Extreme Gradient Boosting.

3.5.4 AdaBoost

Appendix A2, titled ‘AdaBoost: Parameters’, lists the full set of available parameters. The applied Grid Search parameters are as follows, with any missing parameters set to their defaults.

Additionally, a 5-fold cross-validation was utilised on the training data:

Table 3-14 - AdaBoost Grid Search Parameters

Grid Search Parameters	Value(s)
base_estimator_max_depth	[1, 8, 16]

Grid Search Parameters	Value(s)
algorithm	[SAMME, SAMME.R]
learning_rate	[0.01,0.1,1,10]
n_estimators	[20, 50, 100, 500, 1000]
min_samples_split	[2, 8, 16, 32]
min_samples_leaf	[1, 2, 4, 8, 16]
max_features	[auto, sqrt, log2]
random_state	[42]
scoring	[accuracy]

The total number of fits performed were 600 using 5-folds for each of 120 candidates.

The Grid Search process identified the optimal model with the following hyperparameters.

Table 3-15 - AdaBoost Optimal Model

Parameters	Value
base_estimator_max_depth	[16]
algorithm	[SAMME]
learning_rate	[1]
n_estimators	[500]
min_samples_split	[2]
min_samples_leaf	[1]
max_features	[None]
random_state	[42]
scoring	[accuracy]

3.5.5 Decision Tree Classifier

Appendix A2, titled ‘Decision Tree Classifier: Parameters’, lists the full set of available parameters for this algorithm. The applied Grid Search parameters are as follows, with any missing parameters set to their defaults. Additionally, a 5-fold cross-validation was utilised on the training data:

Table 3-16 - Decision Tree Grid Search Parameters

Grid Search Parameters	Value(s)
criterion	[gini, entropy]
splitter	[best, random]
max_depth	[2, 5, 10, 20, 25, 30, 50]
min_samples_split	[2, 4, 8, 16, 32]
min_samples_leaf	[1, 2, 4, 8, 16]
max_feature	[auto, sqrt, log2]
random_state	[42]
scoring	[accuracy]

The total number of fits performed were 10,500 using 5-folds for each of 2,100 candidates.

The Grid Search process identified the optimal model with the following hyperparameters.

Table 3-17 - Decision Tree Optimal Model

Parameters	Value(s)
criterion	[entropy]
splitter	[random]
max_depth	[25]
max_samples_split	[16]
max_samples_leaf	[1]
max_feature	[auto]
random_state	[42]
scoring	[accuracy]

3.5.6 Support Vector Machines

The scikit-learn Support Vector Classifier algorithm was employed, as the experiment was framed as a classification task. Four different kernels were used: Polynomial, Radial Basis Function, Sigmoid and Linear Kernels. The Polynomial Kernel was set to degrees 2 and 3.

Appendix A2, titled ‘Support Vector Classifier: Parameters’, lists the full set of available parameters for this algorithm. The applied Grid Search parameters are as follows, with any

missing parameters set to their defaults. Additionally, a 5-fold cross-validation was utilised on the training data:

Table 3-18 - SVC Grid Search Hyperparameters

Grid Search Parameters	Value(s)
kernel	[linear, poly, sigmoid, rbf]
C	[0.1, 1, 10]
decision_function_shape	[ovr]
degree (for polynomial kernel)	[2, 3]
gamma	[scale, auto, 10,1,0.1]
random_state	[42]
scoring	[accuracy]

The total number of fits performed were 315 using 5-folds for each of 63 candidates.

The Grid Search process identified the optimal SVC model with the following hyperparameters.

Table 3-19 - SVC Optimal Model

Parameters	Value(s)
kernel	[rbf]
C	[10]
decision_function_shape	[ovr]
gamma	[scale]
random_state	[42]

3.5.7 Artificial and Deep Neural Network using a Multilayer Perceptron

A Multilayer Perceptron, which is a type of feed forward neural network, was identified in related studies, in Section 2.7.

Appendix A2, titled ‘Keras Neural Network API: Parameters’ lists the full set of available parameters in Keras. The applied Grid Search parameters are as follows, with any missing parameters set to their defaults. Additionally, a 5-fold cross-validation was utilised on the training data. The hyperparameter values used for the hidden layers would allow for either an ANN or

DNN based on the range extending from two to five. A dropout layer was added prior to the final output layer with a value of ‘0.3’ to cater for overfitting. The output layer was set to a ‘softmax’ activation function with an n value of ‘3’ representing the tercile classes. No random seeding was applied to the Grid Search.

Table 3-20 - Neural Network Grid Search Hyperparameters

Grid Search Parameters	Value(s)
input_shape	[16]
batch_size	[32, 64, 128]
epochs	[32, 64, 128]
unit	[8,32,64,12]
optimiser	[adam, rmsprop]
activation	[relu]
hidden_layers	[2,3,4,5]
Output Layer: activation	[softmax, 3]
loss	[categorical_crossentropy]
metrics	acc

The total number of fits performed were 1080 using 5-folds for each of 216 candidates.

The Grid Search process identified the optimal neural network model with the following hyperparameters.

Table 3-21 - ANN Optimal Model

Parameter	Value(s)
input_shape	[16]
batch_size	[64]
epochs	[64]
unit	[32]
optimiser	[adam]
activation	[relu]
hidden_layers	[2]
Output Layer: activation	[softmax, 3]

Parameter	Value(s)
loss	[categorical_crossentropy]

3.5.8 Logistic Regression

Appendix A2, titled ‘Logistic Regression Parameters’, lists the full set of available parameters for this algorithm. The applied Grid Search parameters are as follows, with any missing parameters set to their defaults. Additionally, a 5-fold cross-validation was utilised on the training data:

Table 3-22 - Logistic Regression Grid Search Parameters

GridSearch Parameters	Value(s)
penalty	[l1, l2, elasticnet, none]
solver	[newton-cg, lbfgs, liblinear, sag, saga]
multi_class	[auto, ovr, multinomial]
max_iter	[1000]
random_state	[42]
scoring	[accuracy]

The total number of fits performed were 175 using 5-folds for each of the 35 candidates.

The Grid Search process identified the optimal model with the following hyperparameters.

Table 3-23 - Logistic Regression Optimal Model

Parameters	Value
penalty	[l2]
solver	[newton-cg]
multi_class	[auto]
max_iter	[1000]
random_state	[42]
scoring	[accuracy]

3.6 Summary

The study focuses on the application of machine learning to predict long-term share returns using fundamental data from JSE-listed shares, particularly exploring the FTSE/JSE All Share Index

data from 2000 to 2021 utilising 22 years of data. Despite its comprehensive data coverage, the study acknowledges limitations such as potential data errors, inconsistencies in data frequency, and changes in accounting standards, as well as biases like survivorship and look-ahead. Data preprocessing included addressing missing data and refining fundamental ratios, supplemented by feature engineering, outlier management, scaling, and encoding.

In the practical implementation of this study, eight classification algorithms were employed within a supervised machine learning framework to predict 12-month share returns through a tercile ranking system. Key tools used include Jupyter and Colab Notebooks, along with Scikit-learn and Keras libraries. The algorithms, including Random Forest, Gradient Boosting, XGBoost, AdaBoost, Decision Tree Classifier, Support Vector Machines, Neural Networks (including Artificial Neural Networks and Deep Neural Networks), and Logistic Regression, underwent rigorous optimisation using Grid Search and 5-fold cross-validation. The study also incorporated evaluation metrics like accuracy, precision, recall, F1 score, mean absolute error, and absolute error and employed SHAP Summary Plots for feature importance analysis to enhance model interpretability.

4. Results and Analysis

In this chapter, the outcomes and analysis of the three experiments are presented. The first experiment compared the performance of various algorithms, including a detailed analysis of the top three performing algorithms. The second experiment focused on sector and share-specific findings for these top performers. The third and final experiment explored model interpretability, particularly for XGBoost and Random Forest models, using SHAP Summary Plots to assess feature importance. This analysis demonstrated the contribution of each feature to the model's predictions, both globally and for each tercile class.

4.1 Experiment 1 – Comparison of Machine Learning Algorithms

The results are based on the models optimised through hyperparameter tuning, as detailed in the 'Experimental Design and Execution' section. Specifically, the optimal models and their hyperparameters are presented as follows: Random Forest in Table 3-9, GradBoost in Table 3-11, XGBoost in Table 3-13, AdaBoost in Table 3-15, Decision Tree in Table 3-17, Support Vector Classifier with the Radial Basis Function in Table 3-19, Artificial Neural Network in Table 3-21 and Logistic Regression in Table 3-23.

4.1.1 Algorithm Performance

Metrics, including accuracy, precision, recall, F1 score, absolute error and mean absolute error, were employed for the evaluation of the various algorithms in Experiment 1, as detailed in Section 3.3. The uniform baseline classifier, representing the uniform random distribution of data, served as another form of evaluation. Figure 4-1 presents a line plot of the accuracy and bar plots for precision, recall and F1 score. Figure 4-2 presents a line plot of the mean absolute error, absolute error, along with a bar plot of accuracy results. Table 4-1 provides the detailed values upon which Figures 4-1 and 4-2 are based.

Accuracy, Precision, Recall and F1 Score

In Figure 4-1, higher scores indicate increased accuracy, precision, recall and F1 score results. The least effective algorithms based on accuracy were Logistic Regression, Decision Tree and Artificial Neural Network. Notably, XGBoost, which is listed first in Table 4-1, achieved the highest accuracy score of 51.80%.

XGBoost also achieved the highest scores for precision, recall and F1 score metrics, with a score of 0.52 for all three. GradBoost matched the XGBoost's precision score at 0.52. The worst performing algorithms were Logistic Regression, Decision Tree and Artificial Neural Network.

Mean Absolute Error and Absolute Error

In Figure 4-2, the lower the mean absolute error, the better the performance. GradBoost, which is listed second, has the best score of 0.63, followed by XGBoost and then Random Forest. The least effective algorithms were Logistic Regression, Decision Tree and Artificial Neural Network.

Similarly, lower absolute errors indicate better performance. GradBoost, listed second, has the best score of 365, followed by XGBoost and then Random Forest. The least effective algorithms were Logistic Regression, Decision Tree and Artificial Neural Network.

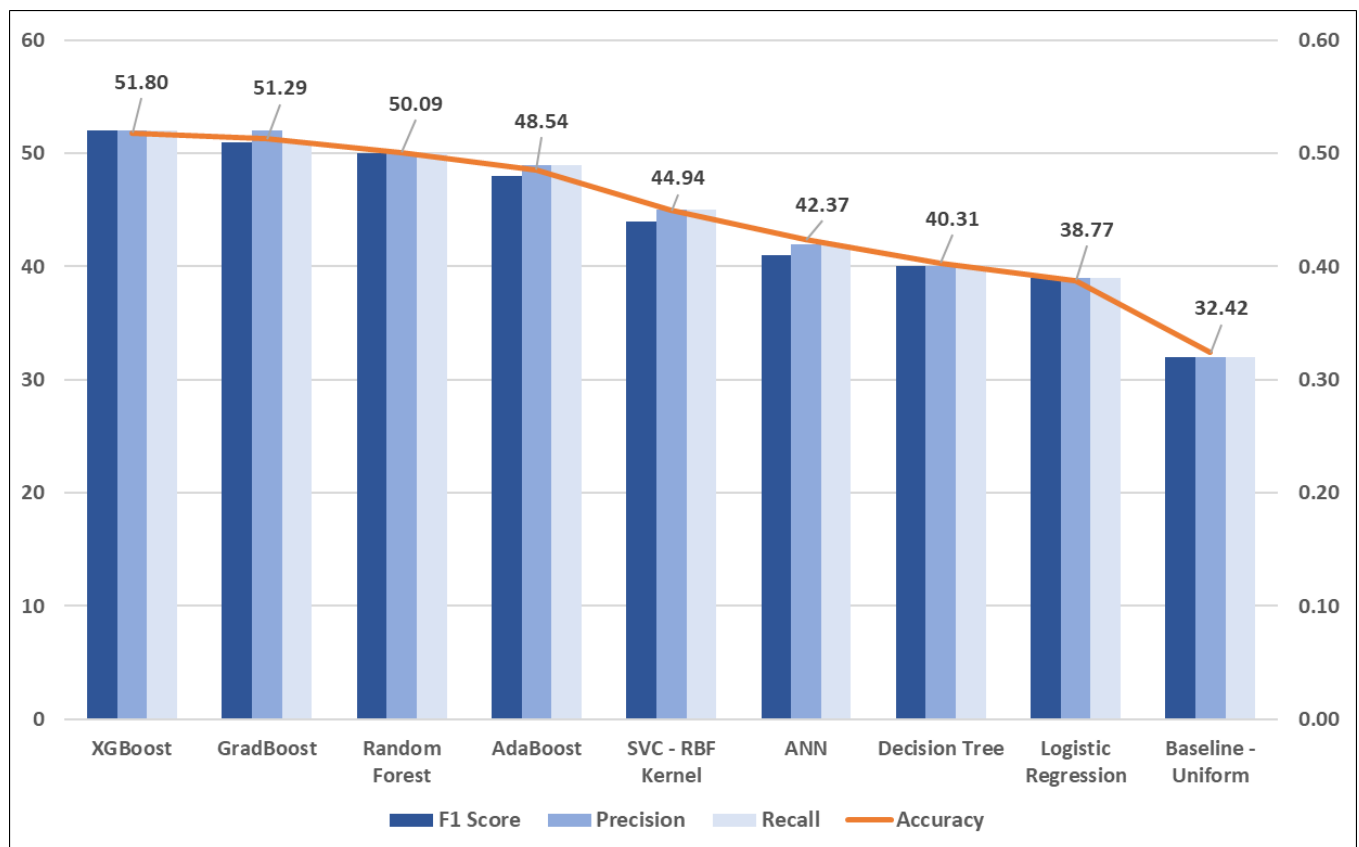


Figure 4-1 - Comparison of Accuracy, Precision, Recall and F1 Score Metrics

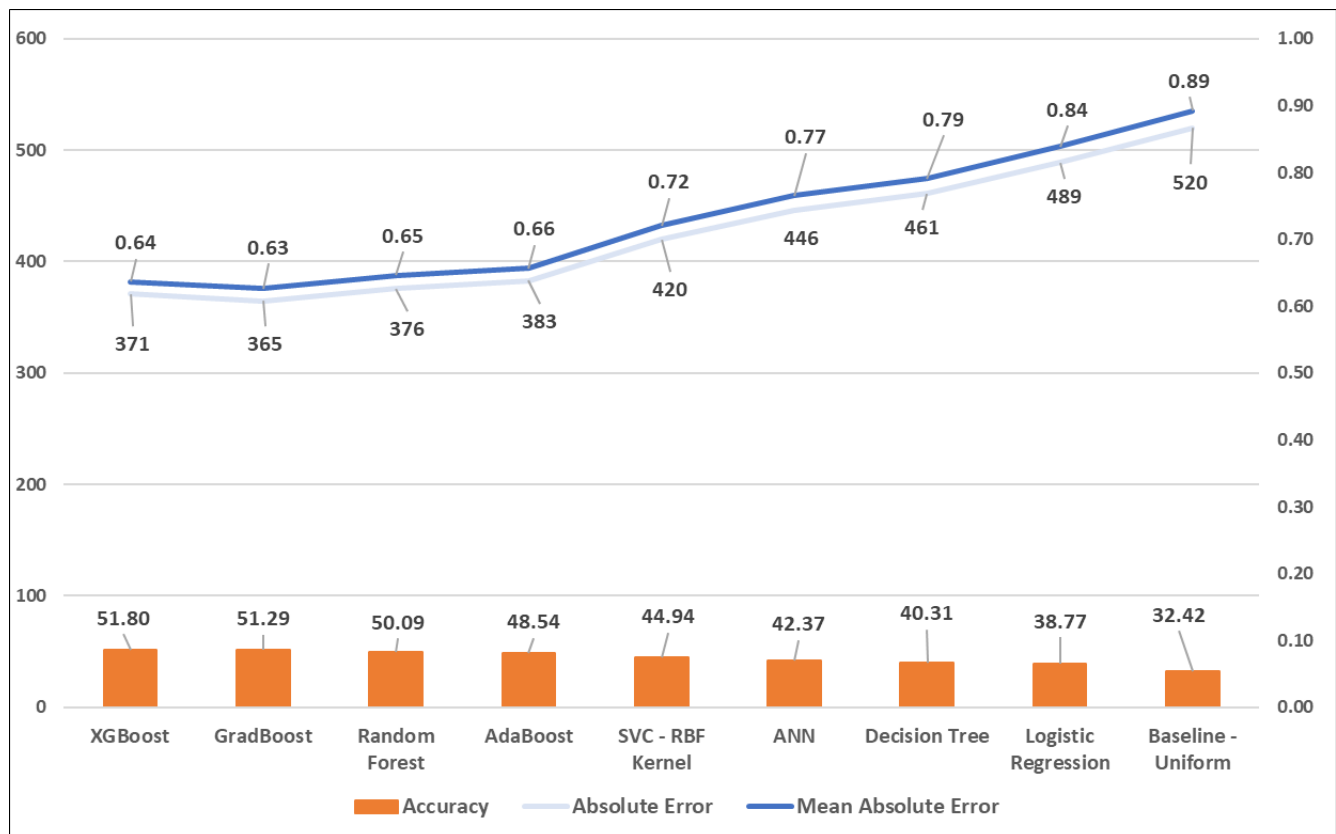


Figure 4-2 - Comparison of Accuracy, Absolute Error and Mean Absolute Error

Table 4-1 - Combined Metrics for Comparison

Algorithm	Accuracy	Precision	Recall	F1 Score	Absolute Error	Mean Absolute Error
XGBoost	51.80	0.52	0.52	0.52	371	0.64
GradBoost	51.29	0.52	0.51	0.51	365	0.63
Random Forest	50.09	0.50	0.50	0.50	376	0.65
AdaBoost	48.54	0.49	0.49	0.48	383	0.66
SVC - RBF Kernel	44.94	0.45	0.45	0.44	420	0.72
ANN	42.37	0.42	0.42	0.41	446	0.77
Decision Tree	40.31	0.40	0.40	0.40	461	0.79
Logistic Regression	38.77	0.39	0.39	0.39	489	0.84
Baseline - Uniform	32.42	0.32	0.32	0.32	520	0.89

4.1.2 Top Performing Algorithms

This section presents detailed results related to the three top performing algorithms. It includes confusion matrices with accuracy metrics, bar plots contrasting correct versus incorrect predictions and a table of metrics including precision, recall and F1 scores per tercile class.

These results, shown in Figures 4-4, 4-6 and 4-8, consistently demonstrate greater accuracy across all three algorithms in predicting the 1st and 3rd tercile classes. According to the confusion matrix data, the Random Forest algorithm exhibits the highest accuracy for predicting the 1st tercile class at 18.18%, followed by XGBoost at 17.84% and GradBoost at 16.30%. For the 3rd tercile class, the GradBoost algorithm achieved the highest accuracy of 19.55%, followed by the XGBoost at 18.55% and then the Random Forest at 17.67%. In predicting the 2nd tercile class, the GradBoost algorithm again leads with an accuracy of 15.44%, followed by XGBoost at 15.09% and the Random Forest algorithm at 14.24%. Notably, these scores for the 2nd tercile class are notably lower than those for either the 1st or 3rd tercile classes.

4.1.2.1 XGBoost

Confusion Matrix

The Confusion Matrix in Figure 4-3 presents the nominal count along with the relative percentage accuracy for each quadrant. The overall algorithm accuracy, indicated at the bottom of the matrix, is 51.80%. In the comparison of the top three algorithms, XGBoost achieved the second highest accuracy rates across all terciles: 17.84% for the 1st tercile, 15.09% for the 2nd tercile and 18.87% for the 3rd tercile.

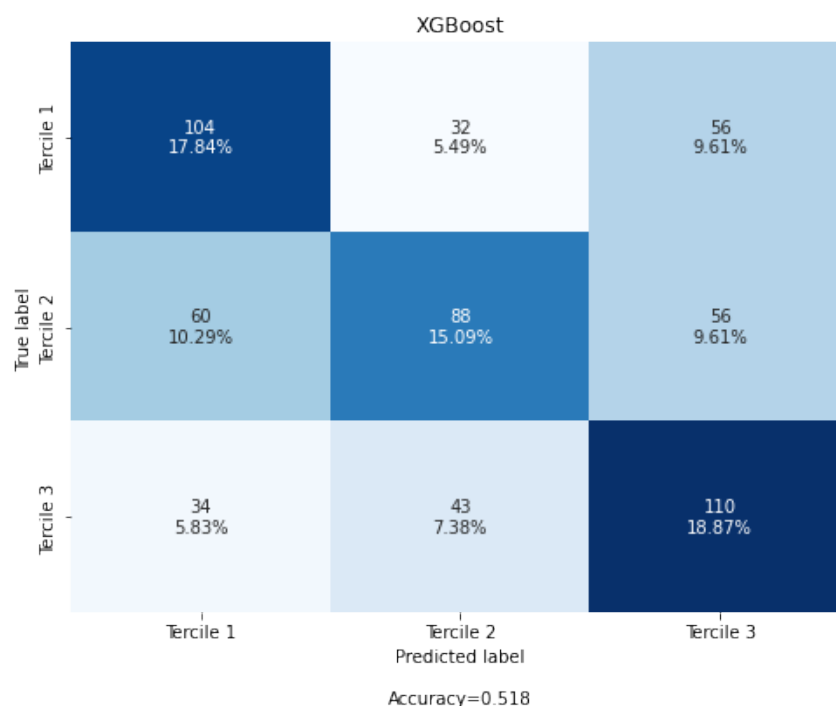


Figure 4-3 - XGBoost Confusion Matrix

Bar Plot of Prediction Accuracy per Tercile: Correct versus Incorrect

The bar plot in Figure 4-4 clearly demonstrates that the best accuracy was achieved for both the 1st and 3rd tercile classes.

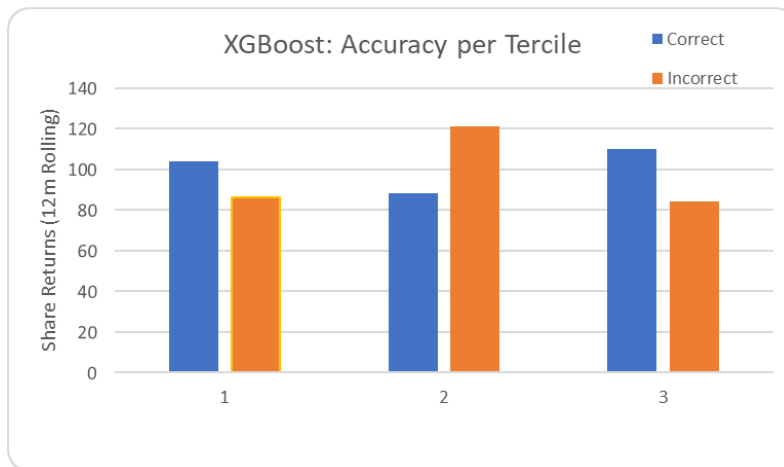


Figure 4-4 - XGBoost Correct versus Incorrect

4.1.2.2 GradBoost

Confusion Matrix

The Confusion Matrix in Figure 4-5 displays the nominal count and relative percentage accuracy for each quadrant. The overall algorithm accuracy, indicated at the bottom of the diagram, is 51.30%. This represents the rounded value of the previously disclosed 51.29% in the results section. Among the top three algorithms, the GradBoost algorithm achieved the highest accuracy for the 3rd tercile class, with a score of 19.55%. This compares to 18.87% for XGBoost and 17.67% for Random Forest within the same category. For the 2nd tercile class, GradBoost also led with a score of 15.44%, surpassing XGBoost's 15.09% and Random Forest's 14.24%.

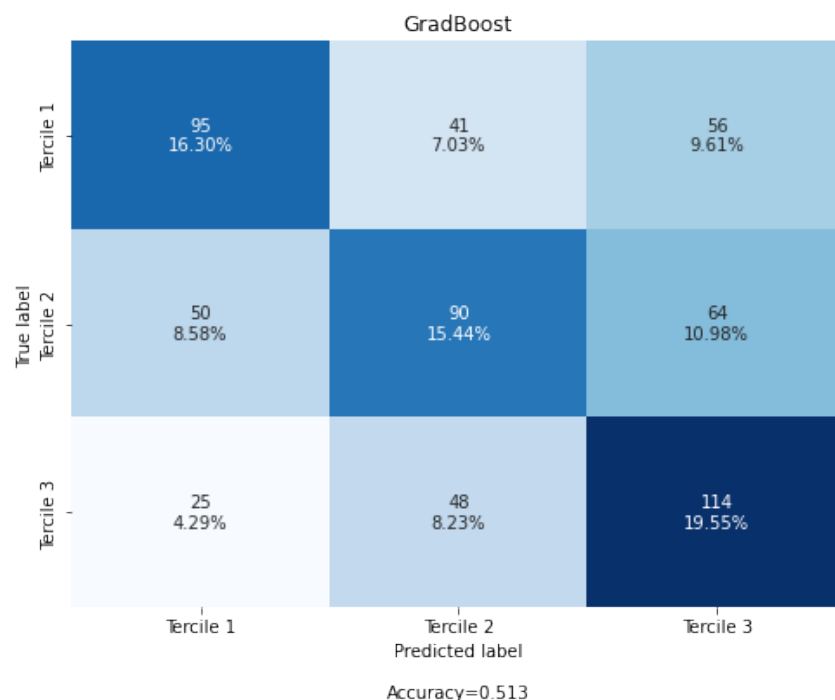


Figure 4-5 - GradBoost Confusion Matrix

Bar Plot Depicting Tercile Prediction Accuracy: Correct versus Incorrect Predictions

The bar plot in Figure 4-6 clearly indicates that the best accuracy was achieved for both the 1st and 3rd tercile classes. Notably, the highest accuracy was recorded for the 3rd tercile class.

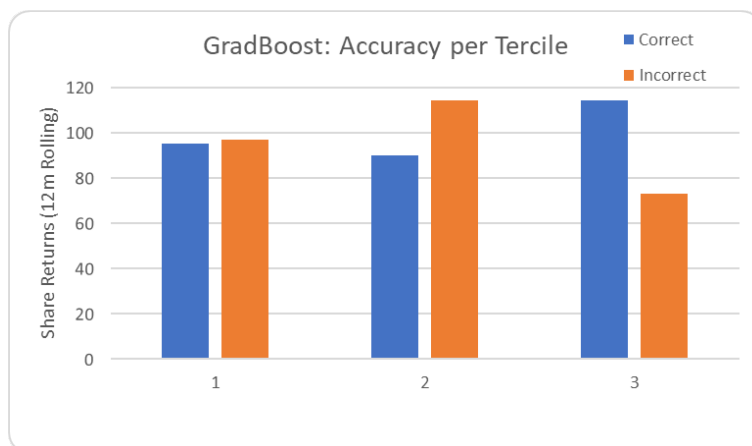


Figure 4-6 - GradBoost Tercile Prediction Accuracy: Correct versus Incorrect Predictions

4.1.2.3 Random Forest

Confusion Matrix

The Confusion Matrix in Figure 4-7 displays the nominal count and relative percentage accuracy for each quadrant. The overall algorithm accuracy, indicated at the bottom of the diagram, is 50.10%. This represents the rounded value of 50.09%, as disclosed earlier in the results section.

Among the top three algorithms, the Random Forest algorithm achieved the highest accuracy for the 1st tercile class, scoring 18.18%.

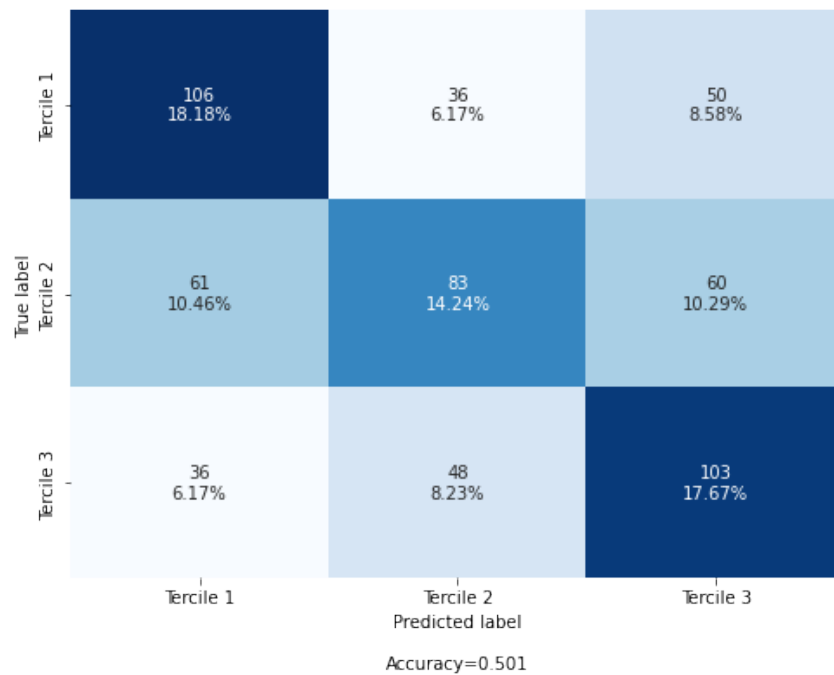


Figure 4-7 - Random Forest Confusion Matrix

Bar Plot Depicting Tercile Prediction Accuracy: Correct versus Incorrect Predictions

The bar plot in Figure 4-8 clearly indicates that the best accuracy was achieved for both the 1st and 3rd tercile classes. Notably, the highest accuracy was recorded for the 1st tercile class, which also ranked as the best among the top three performing algorithms.

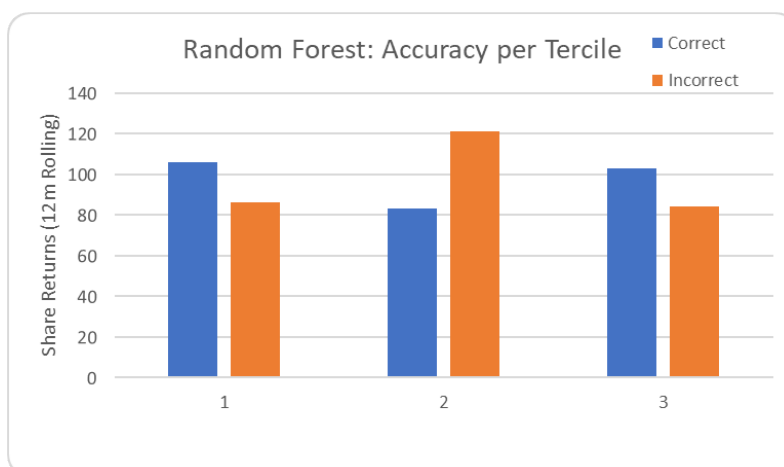


Figure 4-8 - Random Forest Tercile Prediction Accuracy: Correct versus Incorrect Predictions

Precision, Recall, F1 Score and Support per Class

Table 4-2 presents additional detailed metrics for each of the top three performing algorithms, providing further insight into their individual performances. The XGBoost algorithm achieved the highest scores for both the 2nd and 3rd tercile classes in terms of both precision and F1 score. Meanwhile, the GradBoost algorithm scored highest for the 1st and 3rd tercile classes in precision and F1 score, with its F1 score for the 3rd tercile class equalling that of the XGBoost algorithm. Additionally, GradBoost also attained the highest scores in recall for both the 2nd and 3rd tercile classes. For the 1st tercile class, the Random Forest algorithm recorded the highest values in both Recall and F1 score. The Support values represent the nominal counts for each tercile class. These counts are consistent with the experimental process for comparison purposes and the test dataset used.

Table 4-2 - Precision, Recall, F1 Score and Support Values

	XGBoost			GradBoost			Random Forest			ALL
	Precision	Recall	F1 Score	Precision	Recall	F1 Score	Precision	Recall	F1 Score	Support
Tercile 1	0.53	0.54	0.53	0.56	0.49	0.52	0.52	0.55	0.54	192
Tercile 2	0.54	0.43	0.48	0.50	0.44	0.47	0.50	0.41	0.45	204
Tercile 3	0.50	0.59	0.54	0.49	0.61	0.54	0.48	0.55	0.51	187
Accuracy			0.52			0.51			0.50	583
Macro Average	0.52	0.52	0.52	0.52	0.52	0.51	0.50	0.50	0.50	583
Weighted Average	0.52	0.52	0.52	0.52	0.51	0.51	0.50	0.50	0.50	583

4.1.3 Key Findings

In this comparison, XGBoost was identified as the top-performing algorithm with an accuracy of 51.80%, closely followed by GradBoost and Random Forest at 51.29% and 50.09%, respectively. These ensemble algorithms demonstrated superior accuracy performance compared to other types of algorithms. The least effective were the Artificial Neural Network (42.37%), Decision Tree (40.31%) and Logistic Regression (38.77%).

In terms of accuracy in predicting tercile classes, the three leading algorithms demonstrated greater accuracy, particularly in the 1st and 3rd tercile classes. Notably, the Random Forest algorithm achieved the highest accuracy in predicting the 1st tercile class at 18.18%, closely followed by XGBoost at 17.84% and GradBoost at 16.30%. In evaluating the 3rd tercile class, the

GradBoost algorithm was most accurate, registering 19.55%. This was closely followed by XGBoost with an accuracy of 18.55%, and then Random Forest, which 17.67%. For the 2nd tercile class, the best performing algorithm was GradBoost with a 15.44% accuracy, followed by XGBoost at 15.09% and Random Forest with 14.24%

4.2 Experiment 2 - Sector and Share-level Performance Analysis

This section presents a variety of diagrams, providing an in-depth analysis of both the share and sector-specific performance of the top three performing algorithms. The sector-specific performance is a comparison to the overall market, while the share-specific performance focuses on individual shares.

4.2.1 Sector-level Performance

The sector-level analysis, presented below, has been conducted from the ‘bottom-up’. This involved first classifying the shares into the relevant sectors and then calculating an overall accuracy score for each sector against the entire test dataset. This approach forms the basis of comparison and yields useful insights. Figure 4-9 represent a summary view of the performance of the top three algorithms across each of the sectors. Table 4-3 provides a summary view of the underlying data.

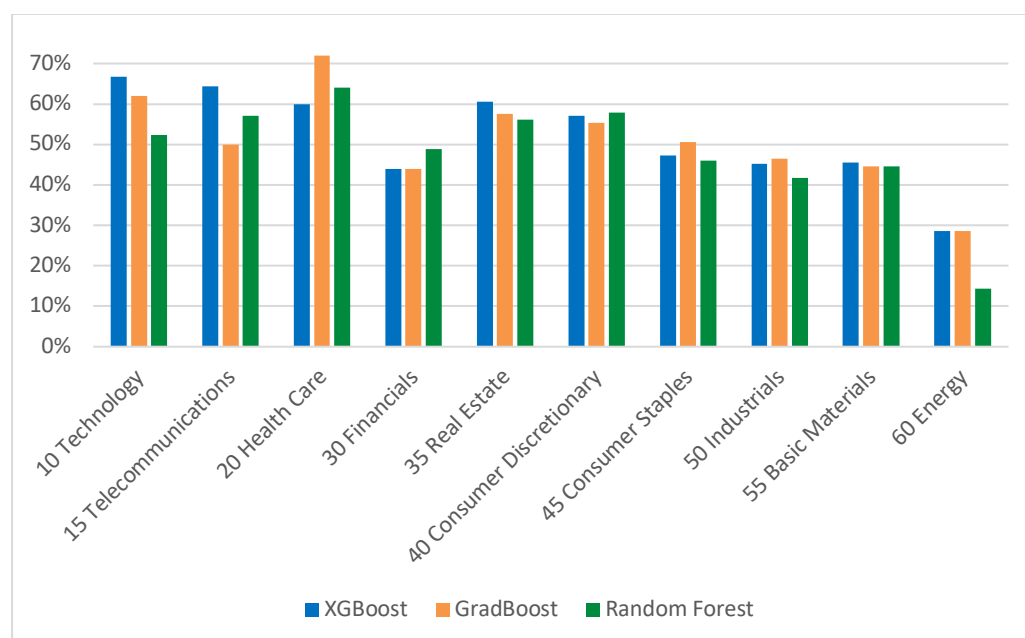


Figure 4-9 - Summary of Top Three Algorithms' Accuracy Scores per Sector

Table 4-3 - Summary of Top Three Algorithms' Accuracy Scores per Sector

Sectors/Description	Accuracy		
	XGBoost	GradBoost	Random Forest
Overall Market Accuracy	51.80%	51.29%	50.09%
10 Technology	66.67%	61.90%	52.38%
15 Telecommunications	64.29%	50.00%	57.14%
20 Health Care	60.00%	72.00%	64.00%
30 Financials	43.90%	43.90%	48.78%
35 Real Estate	60.61%	57.58%	56.06%
40 Consumer Discretionary	57.02%	55.37%	57.85%
45 Consumer Staples	47.19%	50.56%	46.07%
50 Industrials	45.24%	46.43%	41.67%
55 Basic Materials	45.54%	44.55%	44.55%
60 Energy	28.57%	28.57%	14.29%

The most accurate sector across the top three performing algorithms was the Health Care sector for the GradBoost algorithm, with an accuracy score of 72%, as indicated in Figure 4-9 and Table 4-3. This was followed by the Technology sector for the XGBoost algorithm, which achieved a score of 66.67%. The third best performing sector was the Telecommunications for the XGBoost algorithm, with a score of 64.29%.

Included in the algorithm-specific diagrams is the average accuracy for each algorithm, represented by the horizontal orange bar. This provides a useful reference for comparing the relative performance of each sector against the algorithm's average.

XGBoost

The highest scoring sector for the XGBoost algorithm is the Technology sector, with a score of 66.67%. This is followed by the Telecommunication and Real Estate sectors, with respective scores of 64.29% and 60.61%. The worst performing sector is Energy, with an accuracy score of 28.57%. For more details, refer to Table 4-4 and Figure 4-10.

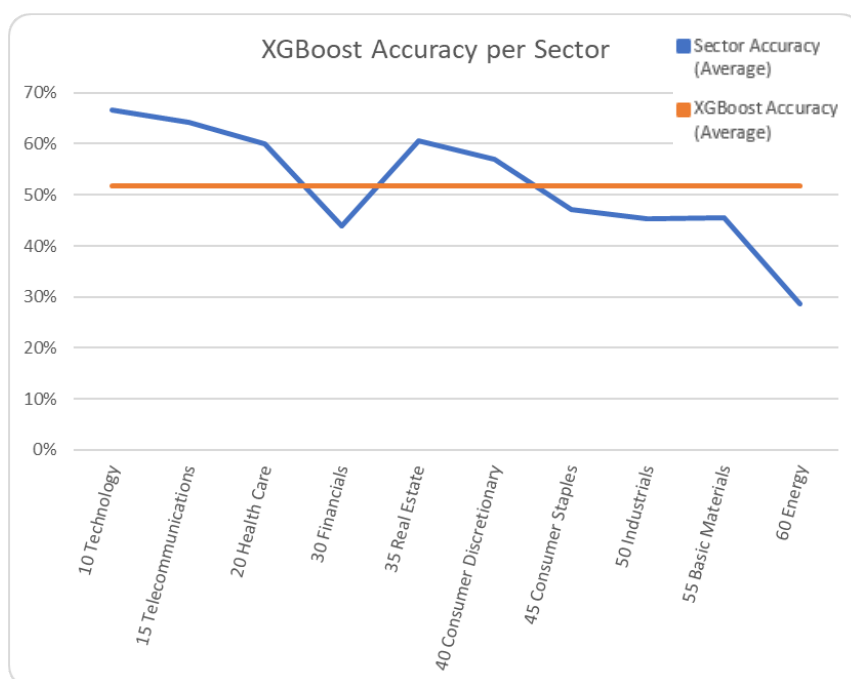


Figure 4-10 - XGBoost Accuracy per Sector

Table 4-4 - XGBoost Accuracy per Sector

XGBoost - Sectors	Accuracy
10 Technology	66.67%
15 Telecommunications	64.29%
20 Health Care	60.00%
30 Financials	43.90%
35 Real Estate	60.61%
40 Consumer Discretionary	57.02%
45 Consumer Staples	47.19%
50 Industrials	45.24%
55 Basic Materials	45.54%
60 Energy	28.57%

GradBoost

The highest scoring sector for the GradBoost algorithm is the Health Care sector, with a score of 72%. This is followed by the Technology and Real Estate sectors, with respective scores of 61.90% and 57.58%. The worst performing sectors are Basic Materials and Energy, with respective scores of 44.55% and 28.57%. For more details, refer to Table 4-5 and Figure 4-11.

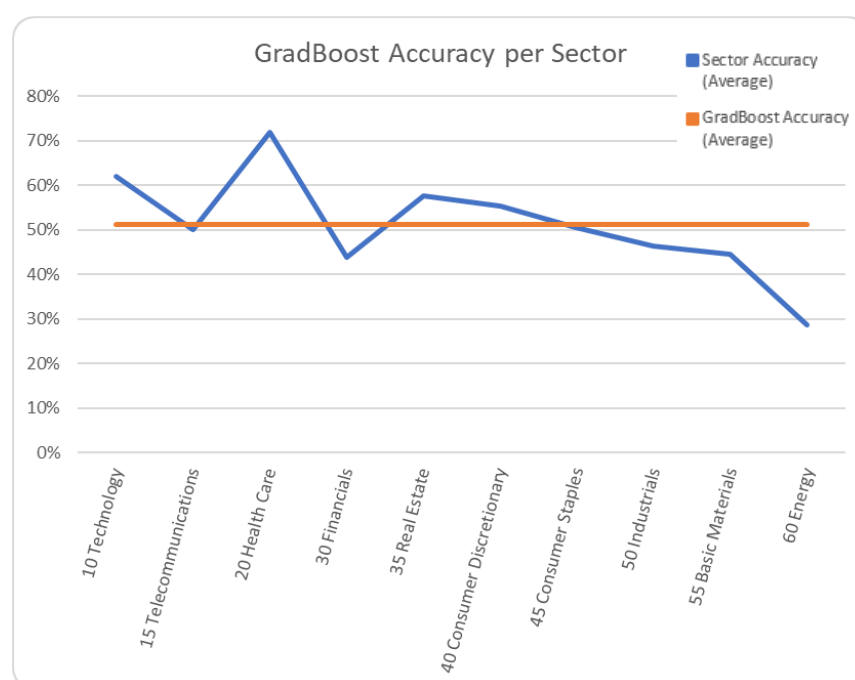


Figure 4-11 - GradBoost Accuracy per Sector

Table 4-5 - GradBoost Accuracy per Sector

GradBoost - Sectors	Accuracy
10 Technology	61.90%
15 Telecommunications	50.00%
20 Health Care	72.00%
30 Financials	43.90%
35 Real Estate	57.58%
40 Consumer Discretionary	55.37%
45 Consumer Staples	50.56%
50 Industrials	46.43%
55 Basic Materials	44.55%
60 Energy	28.57%

Random Forest

The highest scoring sector for the Random Forest algorithm is the Health Care sector, with a score of 64%. This is followed by the Consumer Discretionary and Telecommunications sectors, with respective scores of 57.85% and 57.14%. The worst performing sectors are Basic Materials and Energy, with respective scores of 44.55% and 14.29%. For more details, refer to Table 4-6 and Figure 4-12.

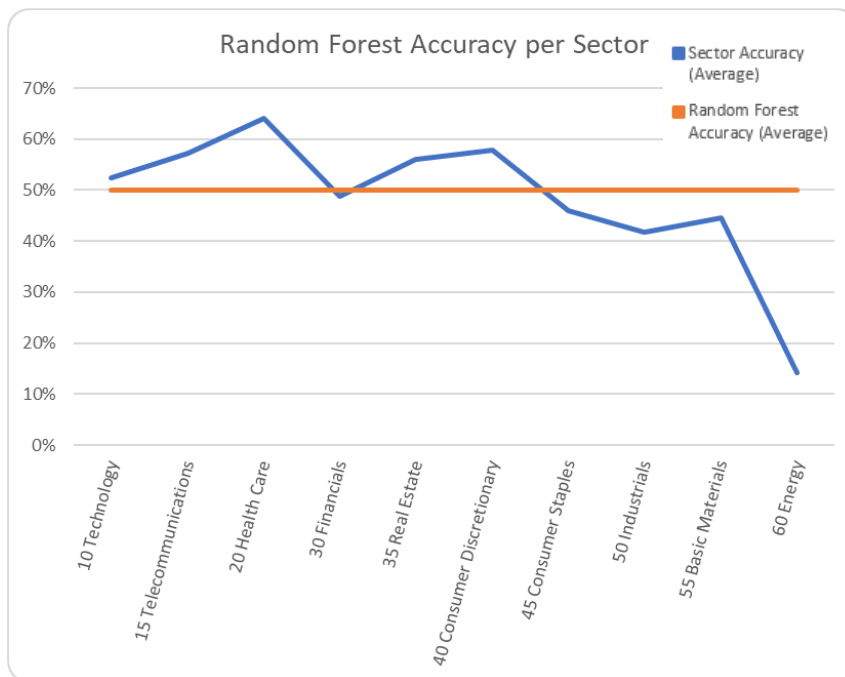


Figure 4-12 - Random Forest Accuracy per Sector

Table 4-6 - Random Forest Accuracy per Sector

Random Forest - Sectors	Accuracy
10 Technology	52.38%
15 Telecommunications	57.14%
20 Health Care	64.00%
30 Financials	48.78%
35 Real Estate	56.06%
40 Consumer Discretionary	57.85%
45 Consumer Staples	46.07%
50 Industrials	41.67%
55 Basic Materials	44.55%
60 Energy	14.29%

4.2.2 Top Three Performing Sectors and Their Respective Shares

This section features various diagrams that show the individual shares for each sector. Included in these diagrams is the average accuracy for each sector, represented by an orange bar. This provides a useful reference for comparing the relative performance of each share against the sector's average and the specific algorithm.

XGBoost

Listed below are the top three performing sectors, along with the underlying shares, for the XGBoost algorithm. These sectors are ordered by their relative accuracy: Technology, Telecommunications and Health Care.

Figure 4-13 illustrates the performance of the top performing Technology sector in relation to the underlying shares. The shares included in the test dataset are Naspers (NPN), Datatec (DTC) and Altron Limited (AEL). Among these, Datatec recorded the highest accuracy relative to the sector average.

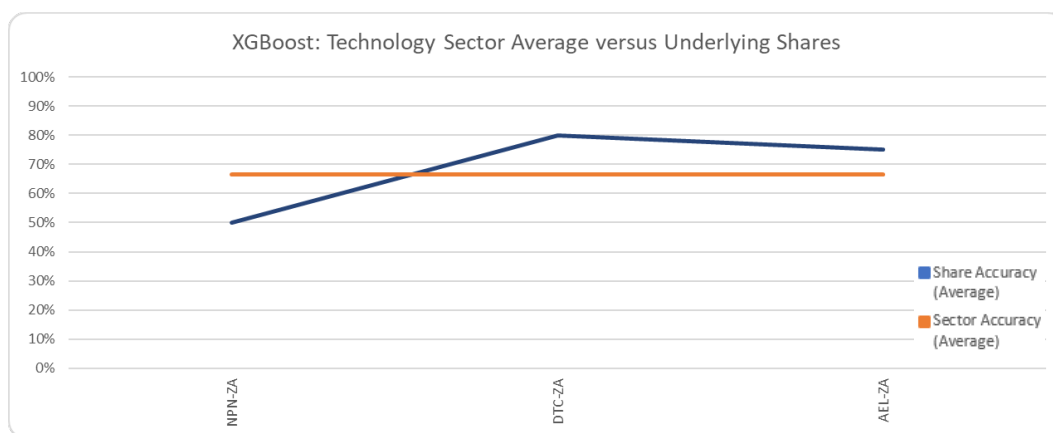


Figure 4-13 - XGBoost Technology Sector

Figure 4-14 displays the Telecommunications sector in relation to the underlying shares. The shares included in the test dataset are MultiChoice Group (MCG), Vodacom (VOD), Telkom SA (TKG), Blue Label Telecoms (BLU) and MTN Group (MTN). Vodacom achieved the highest relative accuracy based on the results from the test dataset.

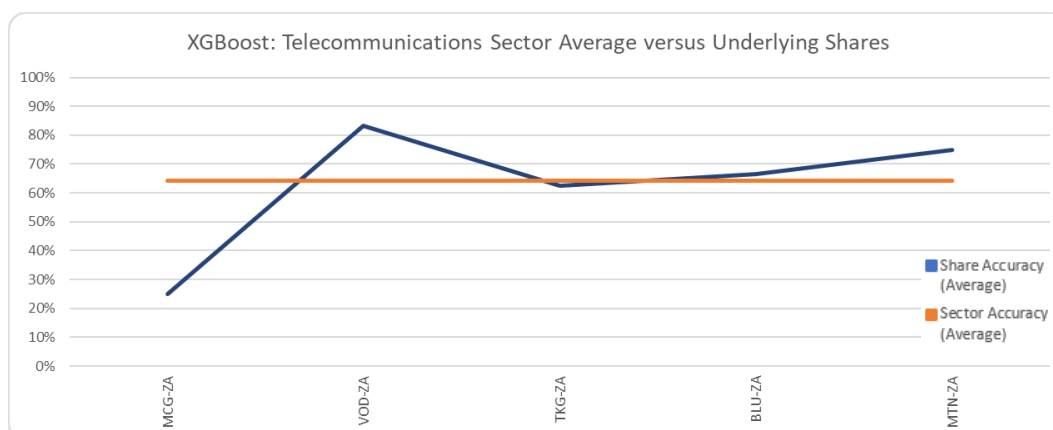


Figure 4-14 - XGBoost Telecommunications Sector

Figure 4-15 illustrates the Health Care sector in relation to the underlying shares. The shares included in the test dataset are Netcare Limited (NTC), Aspen Pharmacare Holdings Limited (APN), Adcock Ingram Holdings Limited (AIP), Life Healthcare Holdings Limited (LHC) and Mediclinic International PLC (MEI), all of which are plotted in the figure. Mediclinic International PLC achieve the highest relative accuracy based on the results from the test dataset, as seen in the diagram.

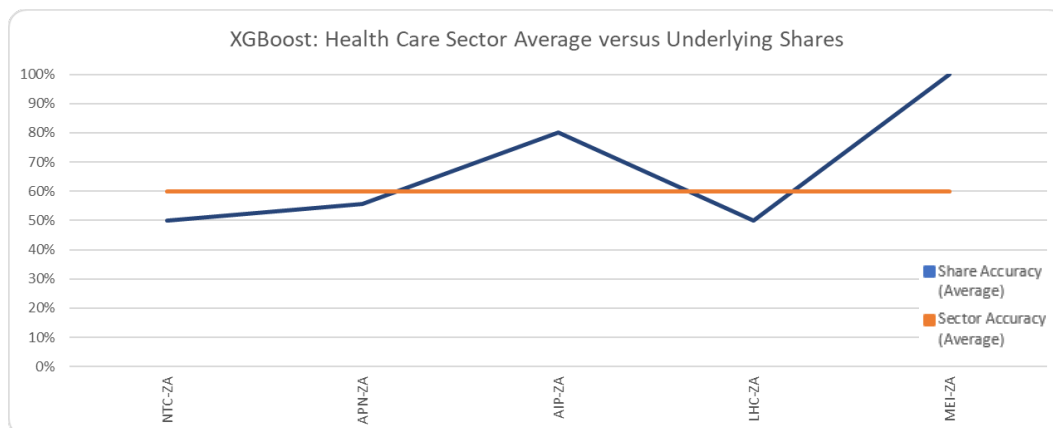


Figure 4-15 - XGBoost Health Care Sector

GradBoost

Listed below are the top three performing sectors, along with the underlying shares, for the GradBoost algorithm. These sectors are ranked in order of their relative accuracy: Health Care, Technology and Real Estate.

Figure 4-16 illustrates the performance of the top performing Health Care sector in relation to the underlying shares. The shares included in the test dataset are Netcare Limited (NTC), Aspen Pharmacare Holdings Limited (APN), Adcock Ingram Holdings Limited (AIP), Life Healthcare Holdings Limited (LHC) and Mediclinic International PLC (MEI). The highest accuracy was recorded for both Adcock Ingram Holdings Limited and Mediclinic International PLC, relative to the sector average.

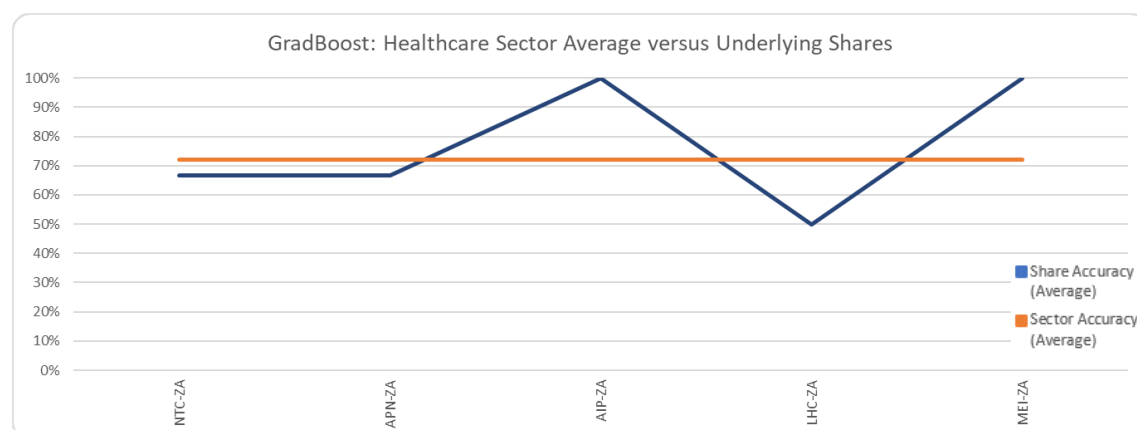


Figure 4-16 - GradBoost Health Care Sector

Figure 4-17 illustrates the performance of the top performing Technology sector in relation to the underlying shares. The shares included in the test data are Naspers (NPN), Datatec (DTC) and Altron Limited (AEL). The highest accuracy was recorded for Datatec, relative to the sector average.

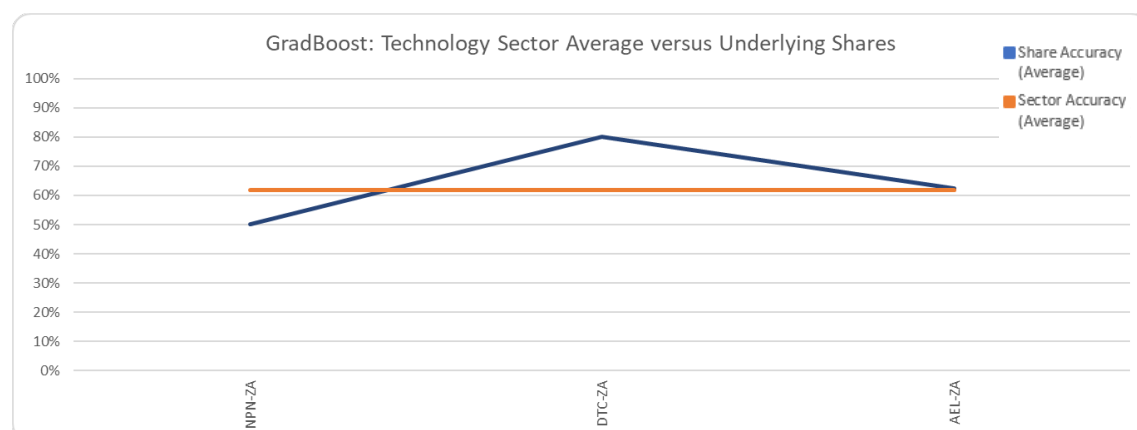


Figure 4-17 - GradBoost Technology Sector

Figure 4-18 illustrates the performance of the top performing Real Estate sector in relation to the underlying shares. The shares included in the test dataset are Redefine Properties Limited (RDF), Fortress REIT Limited B (FFB), Vukile Property Fund Limited (VKE), Hammerson PLC (HMN), Hyprop Investments Limited (HYP), SA Corporate Real Estate Fund Managers (SAC), Fortress REIT Limited A (FFA), Resilient REIT Limited (RES), MAS Real Estate Incorporated (MSP), Growthpoint Properties Limited (GRT), Capital and Counties Properties PLC (CCO), Equities Property Fund Limited (EQU), Emira Property Fund (EMI), Investec Property Fund Limited (IPF), Irongate Group (IAP), Attacq Limited (ATT), Sirius Real Estate Limited (SRE), Industrials REIT Limited (MLI) and Arrowhead Properties Limited B (AHB). The highest accuracy was

recorded for Hammerson PLC, Equities Property Fund Limited, Irongate Group, Sirius Real Estate Limited and Arrowhead Properties Limited B, relative to the sector average for this algorithm.

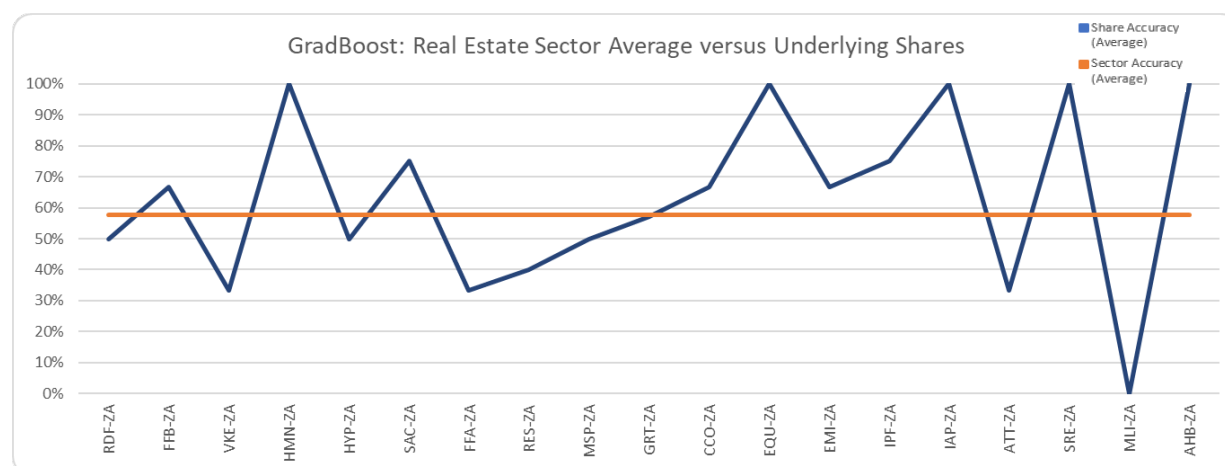


Figure 4-18 - GradBoost Real Estate Sector

Random Forest

Listed below are the top three performing sectors, along with the underlying shares, for the Random Forest algorithm. These sectors are ranked in order of their relative accuracy: Health Care, Consumer Discretionary and Telecommunications.

Figure 4-19 illustrates the performance of the top performing Health Care sector in relation to the underlying shares. The shares included in the test dataset are Netcare Limited (NTC), Aspen Pharmacare Holdings Limited (APN), Adcock Ingram Holdings Limited (AIP), Life Healthcare Holdings Limited (LHC) and Mediclinic International PLC (MEI). The highest accuracy was recorded for Mediclinic International PLC, relative to the sector average.

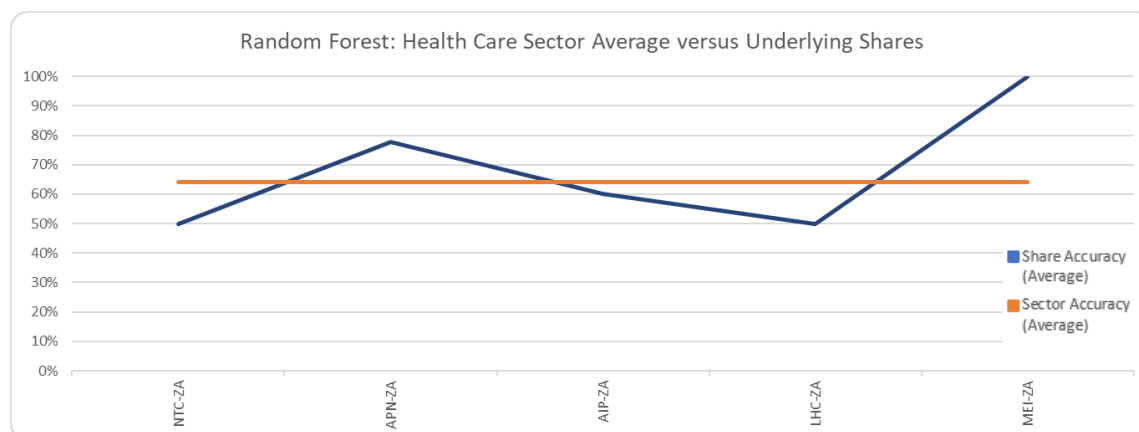


Figure 4-19 - Random Forest Health Care Sector

Figure 4-20 illustrates the performance of the top performing Consumer Discretionary sector in relation to the underlying shares. The shares included in the test dataset are Richemont GDR (CFR), Woolworth Holdings Limited (WHL), Mr Price Group (MRP), City Lodge Hotels (CLH), Metair Investments Limited (MTA), Steinhoff International Holdings NV (SNH), Tsogo Sun Hotels (TGO), Advtech Limited (ADH), Tsogo Sun Gaming Limited (TSG), Curro Holdings Limited (COH), Cashbuild Limited (CSB), Truworths International Limited (TRU), Italtile Limited (ITE), Massmart Holdings Limited (MSM), Pepkor Holdings Limited (PPH), Sun International Limited (SUI), Famous Brands Limited (FBR), Foschini Group Limited (TFG) and Long4Life Limited (L4L). The highest accuracies were recorded for Tsogo Sun Hotels (TGO), Pepkor Holdings Limited (PPH) and Long4Life Limited (L4L), relative to the sector average.

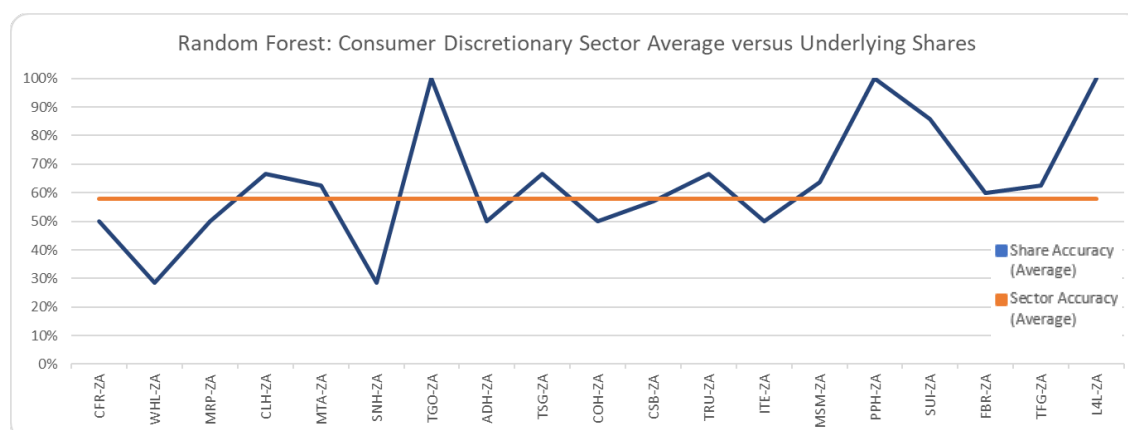


Figure 4-20 - Random Forest Consumer Discretionary Sector

Figure 4-21 illustrates the performance of the Telecommunications sector in relation to the underlying shares. The shares included in the test dataset are MultiChoice Group (MCG), Vodacom (VOD), Telkom SA (TKG), Blue Label Telecoms (BLU) and MTN Group (MTN).

Vodacom achieved the highest relative accuracy based on the results from the test dataset for the Random Forest algorithm.

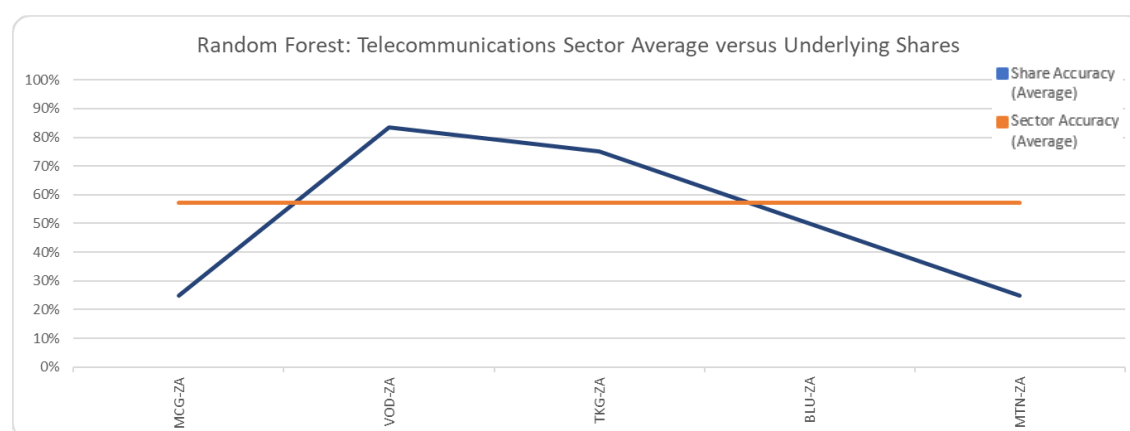


Figure 4-21 - Random Forest Telecommunications Sector

4.2.3 Share-level Performance Over Time

The analysis presented focuses on the performance variation of individual shares at different times for each of the top performing algorithms. The diagrams feature actual values (y_{test} , represented by blue circles) versus predicted values (y_{pred} , represented by orange pluses) for each tercile class. These tercile classes correspond to points in time on the x-axis and plot 12-month rolling share returns using a tercile ranking on the y-axis.

XGBoost

As shown in Figure 4-22, the share selected for this diagram is Naspers Limited (NPN), which is part of the Technology sector. This sector achieved the highest accuracy. The XGBoost algorithm correctly predicted one data point for the 3rd tercile class and two for the 1st tercile class (representing the best tercile or top quantile return), as well as one for the 2nd tercile class. The largest error occurred for an actual value in the 3rd tercile class, where the algorithm predicted the 1st tercile class. All the other errors were one tercile ranking above or below the actual values. The overall accuracy score for the individual data points is consistent with the algorithm's sector-level accuracy score, as detailed in Section 4.2.2.

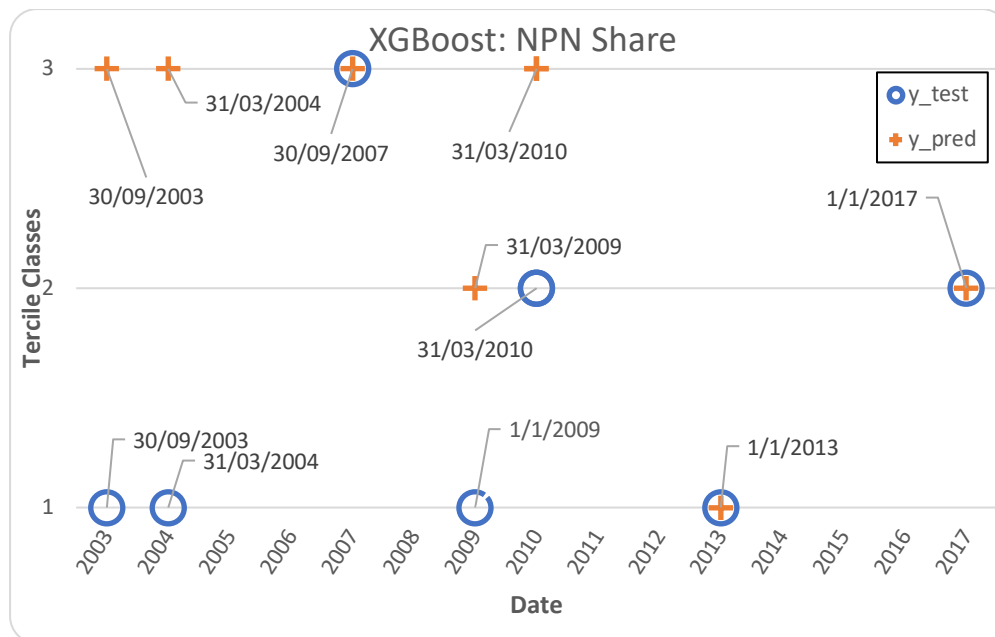


Figure 4-22 - XGBoost Naspers Limited (NPN) Share

GradBoost

As shown in Figure 4-23, the share chosen for this diagram is Netcare Limited (NTC), which is part of the Health Care sector. This sector achieved the highest accuracy for this algorithm. The GradBoost algorithm correctly predicted one data point for the 3rd tercile class, two for the 1st tercile class (representing the best tercile or top quantile return) and one for the 2nd tercile class. The errors made were one tercile ranking above or below the actual values. The overall score of accuracy for the individual data points is consistent with the algorithm's sector-level accuracy score, as detailed in Section 4.2.2.

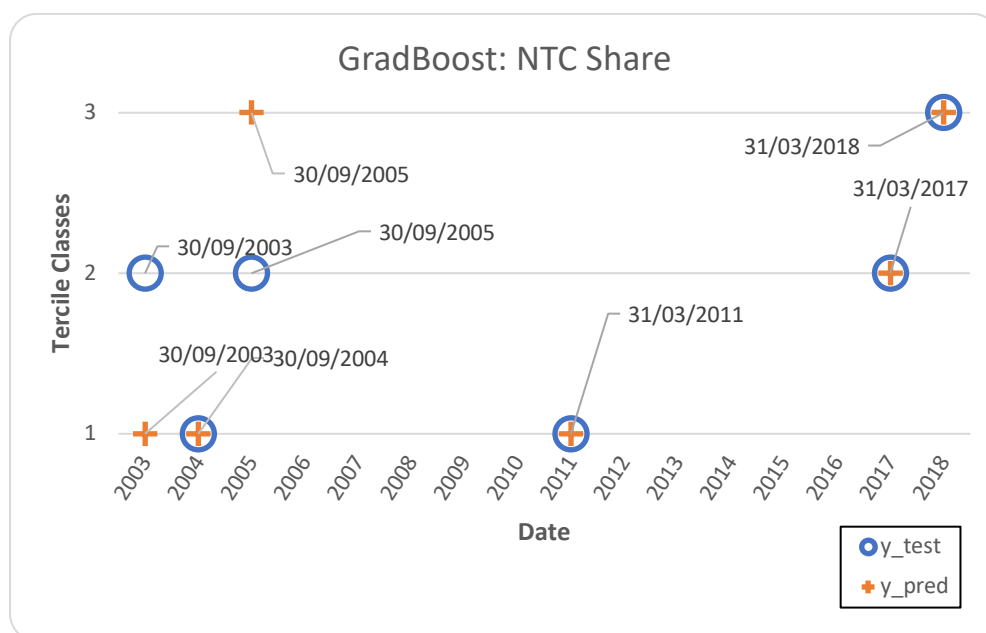


Figure 4-23 - GradBoost Netcare Limited (NTC) Share

Random Forest

As shown in Figure 4-24, the share chosen for this diagram is Netcare Limited (NTC), which is part of the Health Care sector. This sector achieved the highest accuracy for this algorithm. The Random Forest algorithm correctly predicted one for the 3rd tercile class and two for the 1st tercile class (representing the best tercile or top quantile return). Most errors occurred in the 3rd tercile class, with the errors being one tercile ranking above or below the actual values. The overall accuracy score for the individual data points is consistent with the algorithm's sector-level accuracy score, as detailed in Section 4.2.2.

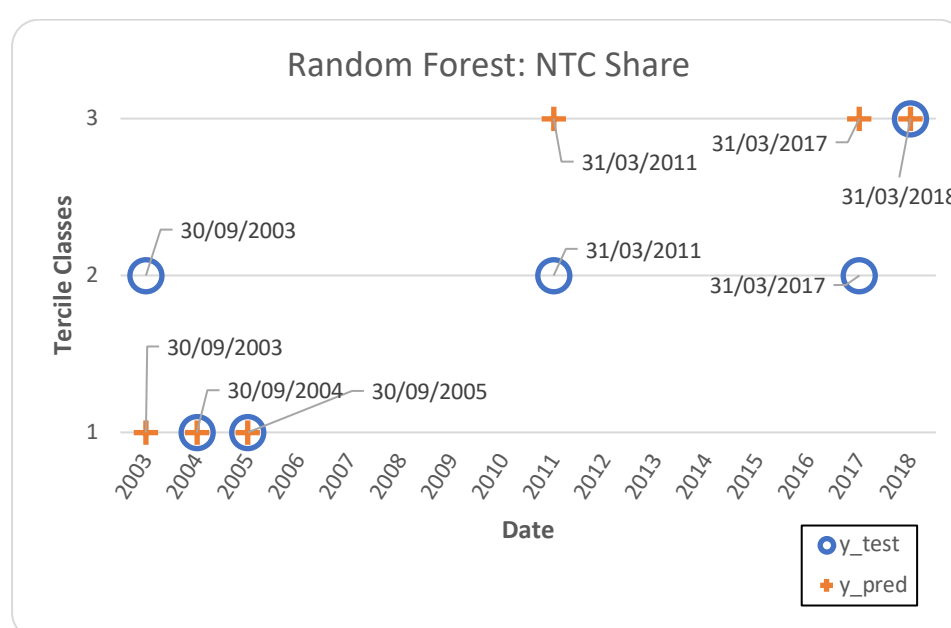


Figure 4-24 - Random Forest Netcare Limited (NTC) Share

4.2.4 Key Findings

The most accurate sector among the top three performing algorithms was Health Care sector for the GradBoost algorithm, with an accuracy score of 72%. This was followed by the Technology sector for the XGBoost algorithm, which achieved a score of 66.67%. The third best performing sector was the Telecommunications for the XGBoost algorithm, with a score of 64.29%. These sector-based predictions significantly surpassed the overall market accuracy of their respective models, as indicated in Table 4-3. The shared-based predictions corroborated the findings of the sector-based analysis, detailed above.

4.3 Experiment 3 - SHAP Analysis

The SHAP library is a valuable tool for gaining additional insight into the models and what drives their behaviour. In this experiment, it has been utilised specifically to understand feature importance in the context of the top performing algorithms. This approach offers a clearer understanding of how fundamental ratios influence predictive outcomes, thereby enhancing interpretability.

4.3.1 Feature Importance using SHAP

The input features in each model have been assessed for their mean relative importance, with a ranking applied to each feature after building and validating the models. This was done using SHAP Summary Plots on the dataset. This analysis helps determine which input features in the dataset are most important for prediction for each model. Additionally, the summary plot illustrates the relative impact of each feature on the tercile classes. The findings for each of the top three performing models are depicted in the figures and discussion below.

XGBoost

Price to Sales is shown to have the greatest importance of all features on the model, albeit by a slight relative margin, as depicted in Figure 4-25. The subsequent input features of importance, again with slight relative margins, are Sales Per Share, Price to Earnings, Dividend Payout Ratio and Enterprise Value to EBIT. The least impactful input features for the model are Operating Margin followed by EBIT Margin and Net Income Margin.

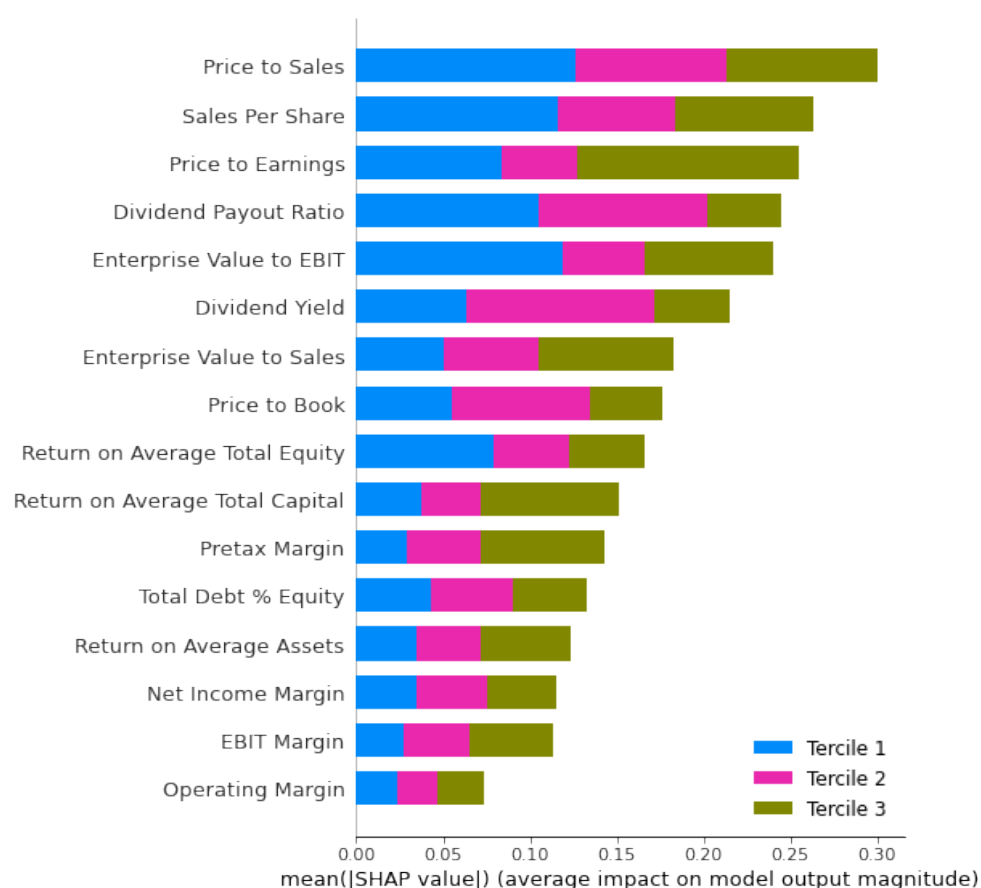


Figure 4-25 - XGBoost SHAP Global Feature Importance

GradBoost

The SHAP library is currently unavailable for use with the GradBoost models when handling more than two categorical classes.

Random Forest

Price to Sales is shown to have the greatest importance of all features on the model, albeit by a slight relative margin, as depicted in Figure 4-26. The subsequent input features of importance, again with slight relative margins, are Price to Earnings, Sales Per Share, Enterprise Value to EBIT and Enterprise Value to Sales. The least impactful input features for the model are Operating Margin followed by EBIT Margin and Net Income Margin.

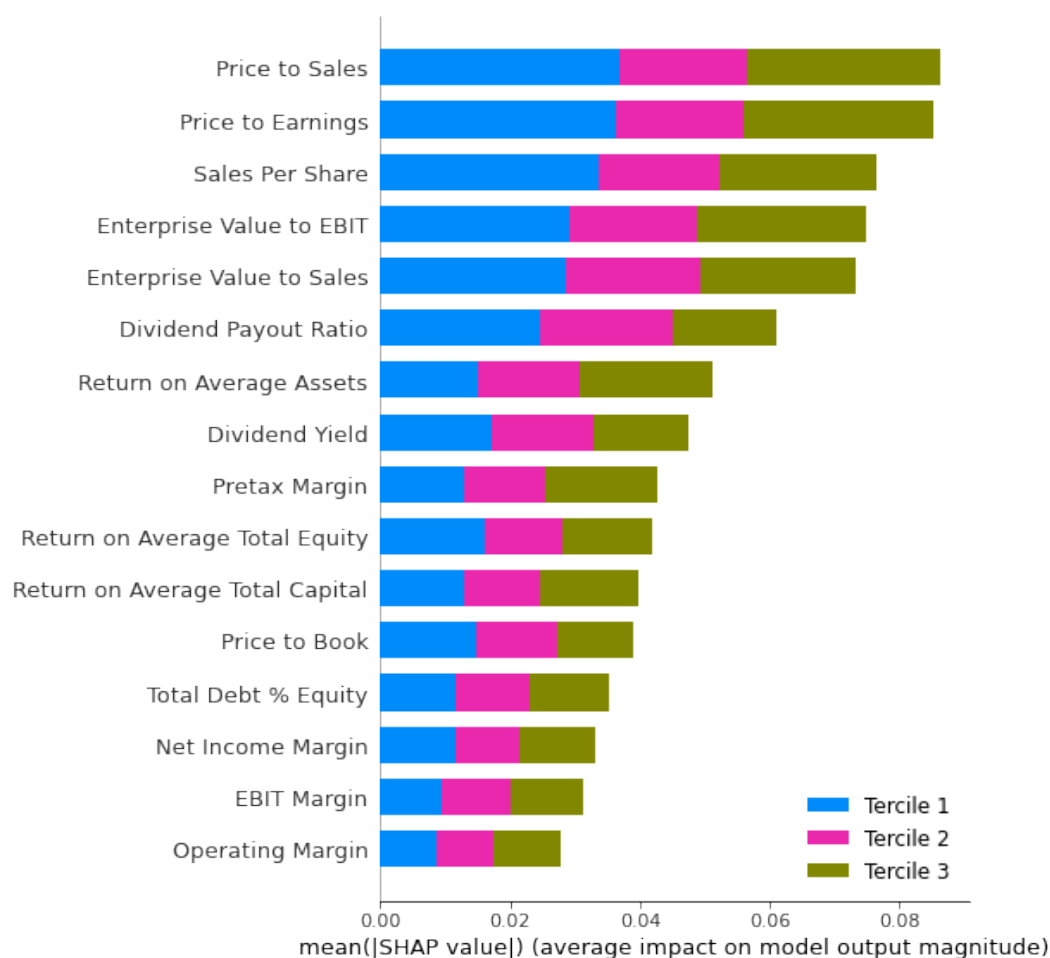


Figure 4-26 - Random Forest SHAP Global Feature Importance

4.3.2 Class Specific Feature Importance Results using SHAP

Figures 4-27 to 4-32 illustrate the three tercile classes making up the tercile rankings for the top performing models, along with the relative importance of the individual features.

XGBoost

The input feature Price to Sales has the largest relative mean impact on the 1st tercile class, which represents the highest ranked shares based on return, denoted as ‘Tercile 1’ in Figure 4-27. This is followed by Enterprise Value to EBIT, Sales Per Share, Dividend Payout Ratio and Price to Earnings.

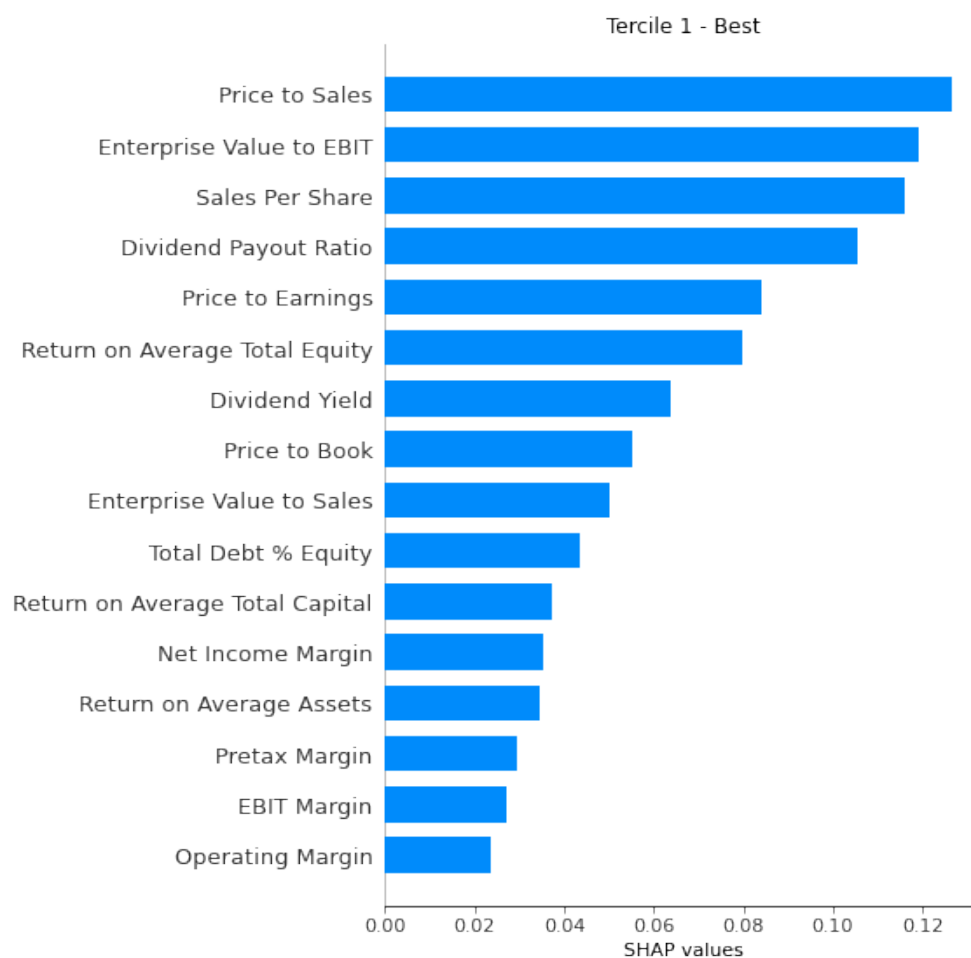


Figure 4-27 - XGBoost SHAP Feature Importance for Tercile 1

The input feature Dividend Yield has the largest relative mean impact on the 2nd tercile class, which represents the mid ranked shares based on return, denoted as ‘Tercile 2’ in Figure 4-28. This is followed by Dividend Payout Ratio, Price to Sales, Sales Per Share and Price to Book.

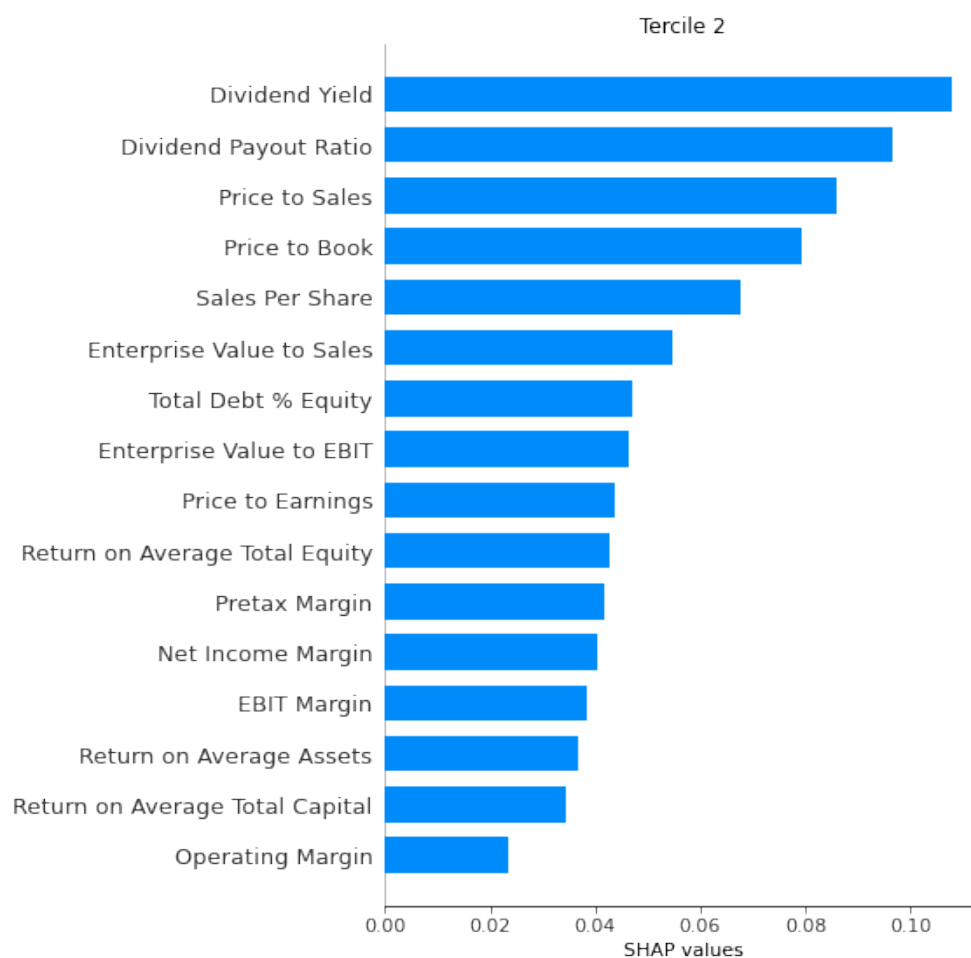


Figure 4-28 - XGBoost SHAP Feature Importance for Tercile 2

The input feature Price to Earnings has the largest relative mean impact on the 3rd tercile class, which represents the worst ranked shares based on return, denoted as ‘Tercile 3’ in Figure 4-29. This is followed by Price to Sales, Return on Average Total Capital, Sales Per Share and Enterprise Value to Sales.

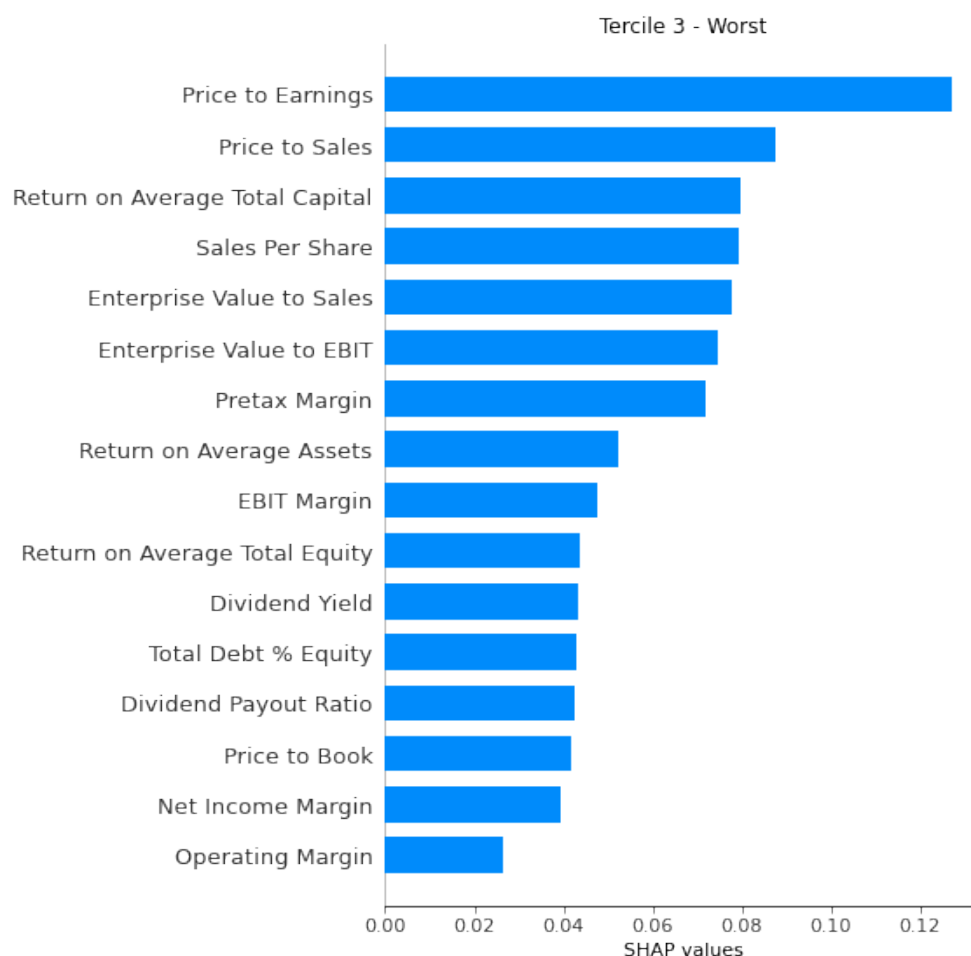


Figure 4-29 - XGBoost SHAP Feature Importance for Tercile 3

GradBoost

The SHAP library is currently unavailable for use with the GradBoost models when handling more than two categorical classes.

Random Forest

The input feature Price to Sales has the largest relative mean impact on the 1st tercile class, which represents the highest ranked shares based on return, denoted as ‘Tercile 1’ in Figure 4-30. This is followed by Price to Earnings, Sales Per Share, Enterprise Value to EBIT and Enterprise Value to Sales.

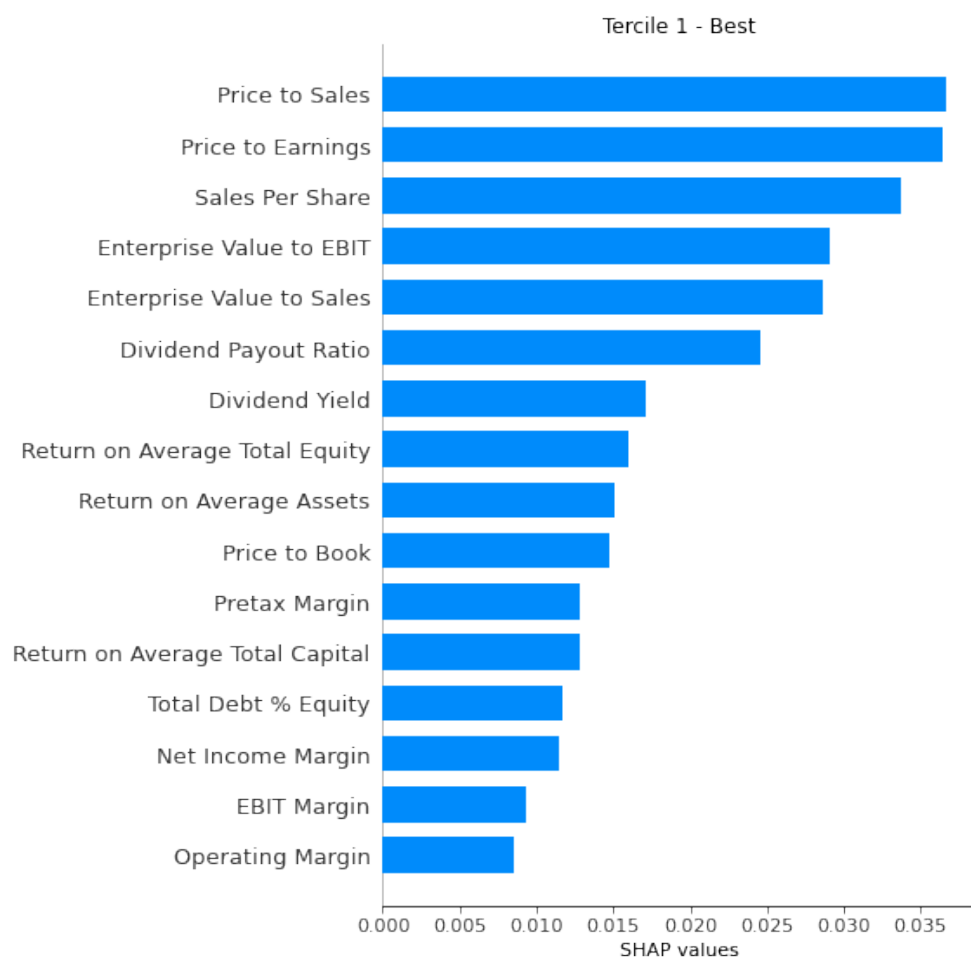


Figure 4-30 - Random Forest SHAP Feature Importance for Tercile 1

The input feature Enterprise Value to Sales has the largest relative mean impact on the 2nd tercile class, which represents the mid ranked shares based on return, denoted as ‘Tercile 2’ in Figure 4-31. This is followed by Dividend Payout Ratio, Price to Sales, Enterprise Value to EBIT and Price to Earnings.

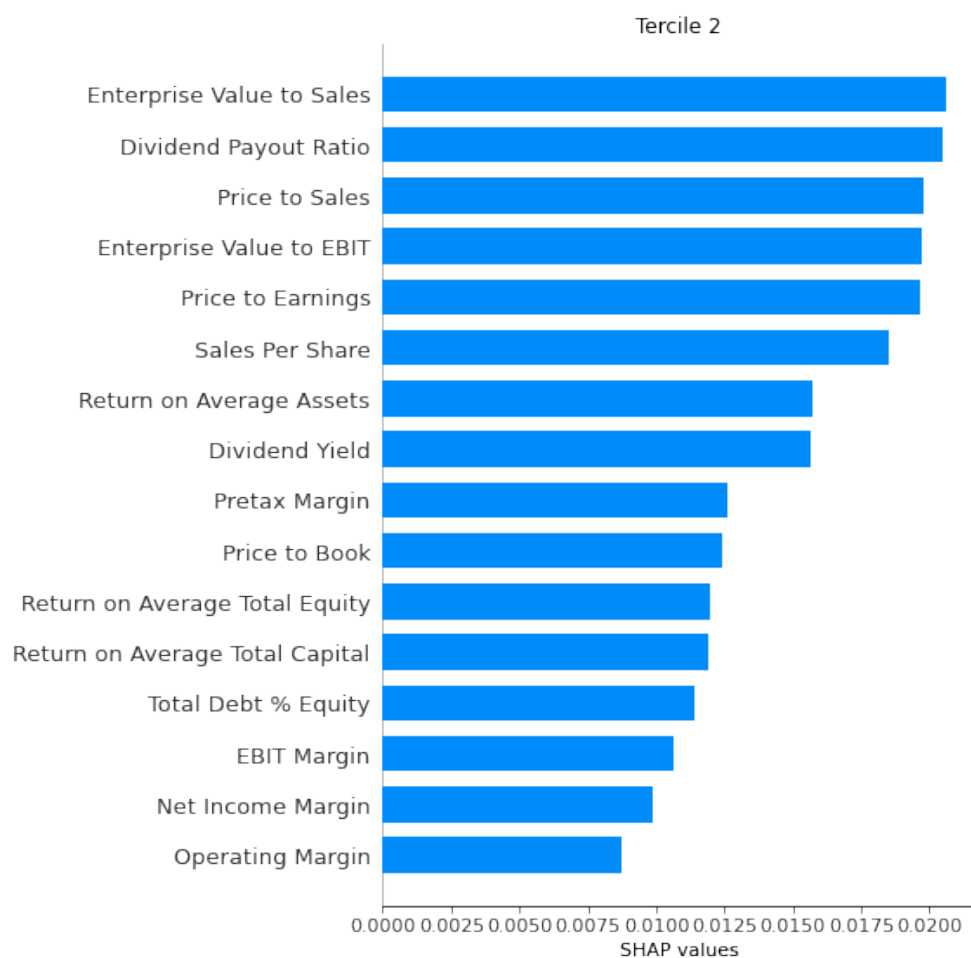


Figure 4-31 - Random Forest SHAP Feature Importance for Tercile 2

The input feature Price to Sales has the largest relative mean impact on the 3rd tercile class, which represents the worst ranked shares based on return, denoted as ‘Tercile 3’ in Figure 4-32. This is followed by Price to Earnings, Enterprise Value to EBIT, Sales Per Share and Enterprise Value to Sales.

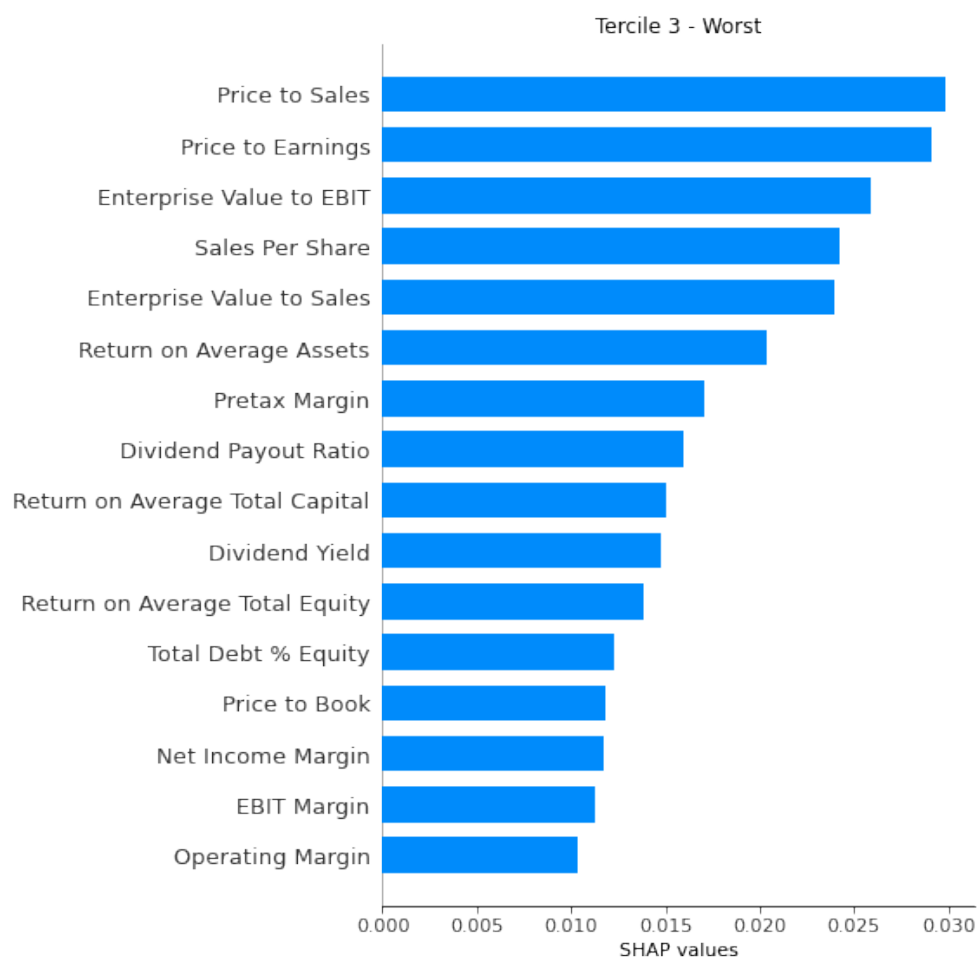


Figure 4-32 - Random Forest SHAP Feature Importance for Tercile 3

4.3.3 Detailed Analysis of Feature Importance Results using SHAP

Table 4-7 presents a detailed analysis of results based on the SHAP Summary Plots for both the global and individual tercile classes. The fundamental ratios contributing most significantly to both the XGBoost and Random Forest models included Price to Sales, Price to Earnings, Sales per Share, Enterprise Value to Earnings before Interest and Tax (EBIT). Additionally, two other fundamental ratios were prominent among the top five contributors: Enterprise Value to Sales and Dividend Payout Ratio for the Random Forest and XGBoost algorithms respectively. These fundamental ratios can be categorised into two main types: valuation and profitability. Notably, two of the key fundamental ratios, Price to Earnings and Price to Sales, are considered valuation-based ratios and Sales Per Share is a profitability-based fundamental ratio. These ratios were among the top three contributing features for the respective models, with valuation-based ratios being the strongest contributors. Conversely, the two least contributing features were Operating Margin and EBIT Margin, both of which are profitability-based ratios.

Table 4-7 - Detailed SHAP Feature Importance

Input Features/Fundamental Ratios	Category	XGBoost	RF	XGBoost			Random Forest		
		Global	Global	Tercile 1	Tercile 2	Tercile 3	Tercile 1	Tercile 2	Tercile 3
Price to Book	VALUATION	8	12	8	4	14	10	10	13
Dividend Yield		6	8	7	1	11	7	8	10
Enterprise Value to EBIT		5	4	2	8	6	4	4	3
Enterprise Value to Sales		7	5	9	6	5	5	1	5
Price to Earnings		3	2	5	9	1	2	5	2
Price to Sales		1	1	1	3	2	1	3	1
Dividend Payout Ratio	PROFITABILITY	4	6	4	2	13	6	2	8
EBIT Margin		15	15	15	13	9	15	14	15
Net Income Margin		14	14	12	12	15	14	15	14
Operating Margin		16	16	16	16	16	16	16	16
Pretax Margin		11	9	14	11	7	11	9	7
Return on Average Assets		13	7	13	14	8	9	7	6
Return on Average Total Capital		9	11	11	15	3	12	12	9
Return on Average Total Equity		10	10	6	10	10	8	11	11
Sales Per Share	LIQUIDITY	2	3	3	5	4	3	6	4
Total Debt % Equity		12	13	10	7	12	13	13	12

4.3.4 Key Findings

The top contributing fundamental ratios for both the XGBoost and Random Forest models were Price to Sales, Price to Earnings, Sales per Share and Enterprise Value to Earnings before Interest and Tax (EBIT). Furthermore, these can be categorised into two main types: valuation and profitability. Notably, Price to Earnings and Price to Sales, are considered valuation-based ratios, while Sales Per Share is a profitability-based fundamental ratio. Among the top three contributing input features for the respective models, the valuation-based ratios emerged as the strongest contributors to model performance.

4.4 Discussion

This section explores the analysis and key findings, starting with the overall performance of the algorithms, including sector and share-level performance, as well as SHAP analysis. Lastly, this study is compared to other studies both in South Africa and internationally.

4.4.1 Overall Performance

The ensemble algorithms, in aggregate, outperformed all the other algorithms based on the evaluation metrics detailed in Table 4-1 – ‘Combined Metrics for Comparison’. These results corroborate findings presented in Section 2.7 of the Literature Review, indicating that ensemble techniques perform well with sparse datasets (Ballings et al., 2015; Huang et al., 2021). The XGBoost algorithm achieved the best accuracy score of 51.80%, followed by the GradBoost at 51.29% and the Random Forest algorithm at 50.09%. Significantly, these algorithms outperformed the Uniform Baseline Classifier, which served as a benchmark approximating a

uniform random distribution of the tercile classes. This suggests that fundamental ratio data can be effective in predicting 12-month share returns. However, caution should be exercised, as the relative outperformance was somewhat muted. The worst performing algorithms were Logistic Regression, Decision Tree and Artificial Neural Network.

The most accurately predicted terciles were consistently the 1st and 3rd across the top performing algorithms, as evidenced by the supporting confusion matrices and bar plots depicting the accuracy of the correct versus incorrect tercile predictions (see Section 4.1.2). Typical portfolio selection is based on selecting the best tercile for long portfolios and worst for short portfolios, implying marginally better real-world performance.

The overall evaluation metrics achieved by the various algorithms are likely the consequence of the sparse input data, as discussed in the Data Preprocessing section (see Section 3.4). This is the case despite the data being sourced from a credible provider and the employment of 'best practices' in data preparation.

4.4.2 Sector and Share-level Performance

The sector-level analysis revealed that the top performing algorithms achieved better results for specific sectors compared to others. This suggests that a mixed approach of selecting specific algorithms for different sectors, could yield the best outcomes. Among the ten sectors, Health Care, Technology and Telecommunications were the most successful for the XGBoost, GradBoost and Random Forest algorithms, respectively. Health Care, in particular, achieved the best accuracy results for both the GradBoost and Random Forest algorithms. The consistently underperforming sectors across all three algorithms were Basic Materials and Energy. Additional charts depicting detailed share-specific outcomes for each sector support these findings. Furthermore, individual share predictions were also plotted against their actual values to incorporate further detailed analysis. These confirm the accuracy achieved for the 1st and 3rd tercile classes by these algorithms.

4.4.3 SHAP Analysis

The analysis of the SHAP Summary Plots revealed that the valuation-based input features used in this study were the top performers in terms of feature importance. Leading the group on aggregate was Price to Sales, followed by Price to Earnings. Sales per Share was the highest ranked feature within the profitability category, followed by Dividend Payout Ratio. The least effective input

features were Operating Margin, EBIT Margin and Net Income Margin, all of which fall within the profitability category.

The input features that have the largest relative mean impact on the accuracy of the predicted tercile class varied across each algorithm. Notably, Price to Sales had the greatest impact on tercile rankings across all models, with Price to Earnings having the second most significant impact.

Based on the above discussion, Price to Sales and Price to Earnings are key input features for any model using fundamental ratios, both belonging to the valuation category of financial ratios. This study's findings align with those of Drue (2018) and Huang et al. (2012), as identified in Section 2.7 of the Literature Review, in this regard. The next highest ranked input feature is Sales Per Share which is a part of the profitability category. XAI techniques can assist practitioners who are domain experts in their fields but have limited knowledge in machine learning, in implementing models. Furthermore, these techniques can aid in both understanding and enhancing the accuracy of financial models, thereby improving their practical application.

4.4.4 Comparison with Other Studies

Further, this analysis compares the obtained results against related work identified in the Literature Review, encompassing both South Africa and international domains. One gap identified in the literature review was the limited number of studies focused on the South African domain, as discussed in Section 2.7.3 of the Literature Review. Additionally, this study included additional algorithms for comparison, drawing on international studies identified in Section 2.7.4 of the Literature Review.

Only one other study, Drue (2018) used a comparable dataset. The dataset in this study differed, as it was sourced from FactSet instead of Refinitiv and considered the implications of multicollinearity. This particularly related to the fundamental ratios that fluctuated with daily and weekly price movements but were tied to static semi-annual figures in the published financial statements (interim and final). This aspect informed the prediction window of the current study, which was extended to a 12-month rolling return, surpassing the frequency of the semi-annual source data. The target variable was ostensibly similar, consisting of class-ranked share returns, albeit without using different time windows, thereby removing any time series aspect. The

emphasis was instead on predicting the next tercile-based share return using exclusively fundamental ratios as input features.

This study incorporated additional machine learning algorithms, aligning with related international work. Drue (2018) used a Support Vector Classifier and compared it with Multiple Linear Regression, employing a binary quantile with a 90-day prediction window. Drue's evaluation metrics were profitability and accuracy, achieving a 61.78% classification accuracy for all input features. In contrast, this study found that the Support Vector Classifier achieved a 44.94% accuracy for all input features and was ranked 5th, with the highest classification accuracy being 51.80% for the XGBoost algorithm (refer to Figure 4-1 and Table 4-1). The differences in quantile classes (binary versus tercile) used for share returns, prediction windows and underlying datasets limits this comparison. However, this study shows that the Support Vector Classifier was outperformed by all algorithms, except for the Artificial Neural Network, Decision Tree and Logistic Regression. Moreover, the sector-level accuracy achieved by the top performing algorithms exceeded that of Drue's overall accuracy of 61.78% for XGBoost (Technology – 66.67%, Telecommunications – 64.29%), as well as for GradBoost (Health Care – 72%, Technology – 61.90%) and Random Forest (Health Care – 64%).

Drue (2018) also evaluated and analysed input features, categorising them. The study found that valuation input features scored the highest accuracy, scoring 56.69%. This study utilised SHAP Summary Plots to assess model interpretability, focusing on feature importance in the XGBoost and Random Forest models. The findings indicated that the Price to Earnings and Price to Sales, both part of valuation, was the highest contributing category (see Table 4-7). This aligns with this aspect of Drue's findings regarding feature importance.

Relevant comparable international studies for long-term predictions using fundamental data for share price movement prediction, employing classification techniques, were limited to Huang et al. (2021), Barak et al. (2017) and Ballings et al. (2015). Ballings et al. (2015) utilised a mixed dataset, comprising both fundamental ratios and economic input features. In contrast, Huang et al. (2021) exclusively used fundamental ratios. The findings from all these studies indicated that ensemble techniques performed best, with each study reporting the highest accuracy achieved with Random Forest algorithms. Specifically, Ballings et al. (2015) achieved an AUC score of 0.903 and Huang et al. (2021) reported a 32.1% compound return for the buy portfolio and a -23.7% return for the sell portfolio. However, this study, demonstrates that Random Forest was

surpassed in accuracy by both the XGBoost and GradBoost algorithms. These findings corroborate the efficacy of ensemble techniques in using fundamental data for share prediction. Huang et al. (2021) also underscored the importance of valuation-based input features, followed by profitability-based ones, in predictive accuracy.

The results of the study do not appear to indicate that South Africa, as a developing market, exhibits greater prediction accuracy due to market inefficiencies, a concept identified by Bustos and Pomares-Quimbaya (2020) in Section 2.2 of the Literature Review. This is likely attributable to its complexity, efficiency and size, but warrants further investigation.

4.5 Summary

The selected classification algorithms successfully predicted the tercile class of each share's 12-month return. XGBoost emerged as the top performer, achieving an accuracy of 51.80%, narrowly surpassing GradBoost's 51.29% and Random Forest's 50.09%. These ensemble algorithms outperformed others, with AdaBoost at 48.54%, Support Vector Classifier with Radial Basis Function at 44.94%, Artificial Neural Network at 42.37%, Decision Tree at 40.31% and Logistic Regression at 38.77%, all proving less effective.

In predicting the tercile class, three algorithms demonstrated superior accuracy in the 1st and 3rd classes. The Random Forest achieved the highest accuracy in the 1st class with 18.18% accuracy, followed by XGBoost at 17.84% and GradBoost at 16.30%. In the 3rd class, GradBoost achieved the highest accuracy with 19.55%, with by XGBoost at 18.55% and Random Forest at 17.67%.

The Health Care sector was the most accurate for the GradBoost algorithm, achieving 72% accuracy. Next, the Technology sector scored 66.67% with XGBoost, followed by Telecommunications at 64.29% also with XGBoost. These sector-specific predictions outperformed overall market predictions for each model. Share-based predictions confirmed the sector-based results.

In analysing the interpretability of the XGBoost and Random Forest models, key input features impacting predictions were identified: Price to Sales, Price to Earnings, Sales per Share and Enterprise Value to EBIT. These findings suggest a strong correlation between these fundamental ratios and model accuracy, categorised into valuation and profitability. Notably, Price to Earnings

and Price to Sales are valuation ratios, while Sales Per Share is based on profitability. Among these, valuation ratios were the top contributors to model performance.

This study compares its results with related work, primarily focusing on the South African market. It expands the scope by incorporating additional algorithms, as highlighted in the Literature Review, Section 2.7. Notably, it differs from Drue (2018), the only comparable study, by using a different dataset and emphasizing fundamental ratios with a 12-month prediction window. Although, the study did not surpass Drue's 61.78% overall accuracy when compared with the best performing XGBoost accuracy of 51.80%, it did demonstrate higher sector-level accuracy, notably achieving 72% in the Health Care sector with the GradBoost algorithm.

Furthermore, this research aligns with international studies, such as those conducted by Huang et al. (2021) and Ballings et al. (2015), which favour ensemble techniques, particularly Random Forest. However, this study shows XGBoost and GradBoost outperforming Random Forest, confirming the effectiveness of ensemble methods in share prediction using fundamental data. Additionally, it reinforces the significance of valuation-based input features, such as Price to Earnings and Price to Sales, echoing Drue's findings and the trends observed in other international research (Huang et al., 2021).

5. Conclusion and Future Work

This chapter concludes the study by providing a summary of the key findings, along with identified limitations and potential avenues for future work.

5.1 Summary of Key Findings

This study aimed to evaluate the performance and interpretability of machine learning classification algorithms in predicting long-term share returns on the JSE. Eight popular machine learning algorithms were selected: XGBoost, Random Forest, GradBoost, AdaBoost, Decision Trees, Support Vector Classifiers, Logistic Regression and Artificial Neural Networks. The performance of these algorithms were assessed based on their ability to predict the tercile class of 12-month share returns. The prediction task utilised historical fundamental ratio data from shares listed on the JSE All Share Index, encompassing 22 years and up to 44 semi-annual observations per share.

The ensemble algorithms outperformed others on aggregate. XGBoost emerged as the best performing algorithm, closely followed by GradBoost and then Random Forest. This relative superiority of the ensemble algorithms could be attributed to the sparse dataset used in the study. Previous research has highlighted their known relative superiority under similar conditions (Ballings et al., 2015).

The detailed analysis of the XGBoost, GradBoost and Random Forest algorithms revealed insights into sector-specific accuracy, demonstrating that several sectors consistently outperformed the overall market. The top performing sectors for XGBoost were Technology, Telecommunications, Real Estate, Health Care and Consumer Discretionary. GradBoost excelled in Health Care, Technology, Real Estate and Consumer Discretionary, while Random Forest led in Health Care, Consumer Discretionary, Telecommunications, Real Estate and Technology. A notable overlap in high performing sectors was observed across all three algorithms, albeit with variations in ranking orders. However, Telecommunications, a strong performer in XGBoost and Random Forest, did not show similar results in GradBoost. In contrast, sectors like Consumer Staples, Basic Materials, Industrials, Financials and Energy consistently underperformed. These findings suggest that selecting algorithms based on predictive accuracy for specific sectors can enhance investment decisions, as evidenced by their significant outperformance compared to the market. Additionally, an ensemble of the three most accurate algorithms could potentially yield profitability in these specific sectors.

The results suggest that the highest and lowest terciles, typically referred to as ‘buy’ and ‘sell’ categories, exhibit the highest accuracy. A conventional long/short portfolio is constructed based on these terciles. This comparative accuracy implies the potential for a profitable portfolio.

SHAP Summary Plots identified several ratios, particularly Price to Sales, Sales per Share and Price to Earnings, as having a greater impact on the XGBoost and Random Forest models. Notably, two of these fundamental ratios, which belong to the valuation category, support the view that they have a significant influence in predicting long-term share returns. Additionally, Sales Per Share, the next ranked fundamental ratio, is categorised under the profitability category. These insights are particularly useful for interpreting the model’s behaviour. They could reduce the number of input features, diminish the computing resources required and lessen the costs associated with data acquisition and retention. Furthermore, these insights will assist with the practical application of machine learning in financial prediction and investment strategies.

In summary, this study contributes to the field of financial machine learning by analysing the performance of various machine learning algorithms in the prediction of long-term share returns. It also enhances the understanding of algorithm interpretability, a crucial aspect for practical applications in financial markets. This research represents the first comprehensive comparison and analysis of both the performance and interpretability of machine learning algorithms to predict long-term share returns on the JSE.

5.2 Limitations

A limitation of this study could be its exclusive reliance on fundamental data for predicting long-term share returns. Furthermore, the availability of fundamental ratio data based on financial statements in the South African context is sparse, with reporting occurring at most semi-annually, as mandated by the JSE. In contrast, overseas financial reporting typically occurs on a quarterly basis. Fundamental data, typically derived from financial statements, may not capture all the market dynamics or external factors that could influence share prices. These factors include macroeconomic indicators, market sentiment, geopolitical events and sector-specific trends.

Furthermore, the study’s use of SHAP Summary Plots for interpretability, while innovative, might not fully capture the complexity of machine learning model decision-making processes. The

interpretability of these models is a nuanced field, and different methods might yield varying insights into how these algorithms arrive at their predictions.

Lastly, the study's specific focus on the JSE might limit the generalisability of the findings to other markets, particularly those with different reporting standards, market dynamics, and economic contexts. This geographical and market-specific focus means the findings might not be directly applicable to other stock exchanges or financial markets.

5.3 Future Work

Future work could include investigations to determine which datasets and machine learning algorithms are most suitable for developed, developing and frontier markets. Currently, there are no comparative studies on share return prediction in these markets and the results of such research would be intriguing. Additionally, assessing whether market dissonance significantly affects share return predictions would be valuable. Future research might also explore the differences between developing and developed markets. This could include examining variations in prediction windows. These prediction windows may be shorter or longer, given that the frequency of published data is typically quarterly in most developed markets.

Further work could include creating an ensemble of the best-performing algorithms, specifically XGBoost, GradBoost and Random Forest. Barak et al. (2017) demonstrated that an ensemble approach achieved improved results compared to individual machine learning algorithms across the broader market.

Another consideration for future work involves sourcing data that complements the prediction window of fundamental ratios. This could include economic factors or indices that serve as proxies for broader market financial conditions, such as inflation or long-term bond yields. Intra-algorithm specifics, such as the relative outperformance of several sectors, could be more thoroughly explored in additional research, as suggested by the findings of this study. This exploration might also include investigating whether the feature importance of fundamentals ratios is sector specific. Without such insights, maintaining and sourcing a large compendium of input features could become a costly overhead, negatively impacting modelling timeframes.

6. References

1. Aas, K., Jullum, M. and Løland, A., 2021. Explaining individual predictions when features are dependent: More accurate approximations to Shapley values. *Artificial Intelligence*, 298, p.103502.
2. Abe, M. and Nakayama, H., 2018, June. Deep learning for forecasting stock returns in the cross-section. In *Pacific-Asia conference on knowledge discovery and data mining* (pp. 273-284). Springer, Cham.
3. Ahmed, I., G. Jeon & F. Piccialli. 2022. From Artificial Intelligence to Explainable Artificial Intelligence in Industry 4.0: A Survey on What, How, and Where. *IEEE Transactions on Industrial Informatics*, vol. 18, no. 8, pp. 5031-5042, Aug. 2022, doi: 10.1109/TII.2022.3146552.
4. Albin, E., Long, J., Dervovic, D. and Magazzeni, D., 2022, June. Counterfactual shapley additive explanations. In 2022 ACM Conference on Fairness, Accountability, and Transparency (pp. 1054-1070).
5. Ballings, M., Van den Poel, D., Hespeels, N. and Gryp, R., 2015. Evaluating multiple classifiers for stock price direction prediction. *Expert systems with Applications*, 42(20), pp.7046-7056.
6. Bandi, H., Joshi, S., Bhagat, S. and Ambawade, D., 2021, June. Integrated Technical and Sentiment Analysis Tool for Market Index Movement Prediction, comprehensible using XAI. In *2021 International Conference on Communication information and Computing Technology (ICCICT)* (pp. 1-8). IEEE.
7. Barak, S., Arjmand, A. and Ortobelli, S., 2017. Fusion of multiple diverse predictors in stock market. *Information Fusion*, 36, pp.90-102.
8. Basak, S., Kar, S., Saha, S., Khaidem, L. and Dey, S.R., 2019. Predicting the direction of stock market prices using tree-based classifiers. *The North American Journal of Economics and Finance*, 47, pp.552-567.
9. Borovkova, S. and Tsiamas, I., 2019. An ensemble of LSTM neural networks for high-frequency stock market classification. *Journal of Forecasting*, 38(6), pp.600-619.
10. Breiman, L., 2001. Random Forests. *Machine Learning*, 45, pp.5-32.
11. Breiman, L., Friedman, J., Stone, C.J., & Olshen, R.A. (1984). *Classification and Regression Trees*. Wadsworth & Brooks/Cole Advanced Books & Software.
12. Bustos, O. and Pomares-Quimbaya, A., 2020. Stock market movement forecast: A Systematic review. *Expert Systems with Applications*, 156, p.113464.
13. Carta, S., Podda, A.S., Reforgiato Recupero, D. and Stanciu, M.M., 2022, February. Explainable AI for financial forecasting. In *Machine Learning, Optimization, and Data Science: 7th International Conference, LOD 2021, Grasmere, UK, October 4–8, 2021, Revised Selected Papers, Part II* (pp. 51-69). Cham: Springer International Publishing.
14. Cheng, C.H. and Chen, Y.S., 2007, August. Fundamental analysis of stock trading systems using classification techniques. In *2007 International Conference on Machine Learning and Cybernetics* (Vol. 3, pp. 1377-1382). IEEE.

15. Chen, T. and Guestrin, C., 2016, August. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 785-794).
16. Chen, S. and He, H., 2018, October. Stock prediction using convolutional neural network. In *IOP Conference Series: Materials Science and Engineering* (Vol. 435, No. 1, p. 012026). IOP Publishing.
17. Chen, Y.C. and Huang, W.C., 2021. Constructing a stock-price forecast CNN model with gold and crude oil indicators. *Applied Soft Computing*, 112, p.107760.
18. Chen, A.S., Leung, M.T. and Daouk, H., 2003. Application of neural networks to an emerging financial market: forecasting and trading the Taiwan Stock Index. *Computers & Operations Research*, 30(6), pp.901-923.
19. Chen, L., Pelger, M. and Zhu, J., 2023. Deep learning in asset pricing. *Management Science*.
20. Choi, S. and Renelle, T., 2019, July. Deep learning price momentum in US equities. In *2019 International Joint Conference on Neural Networks (IJCNN)* (pp. 1-8). IEEE.
21. Cortes, C. and Vapnik, V., 1995. Support-vector networks. *Machine learning*, 20, pp.273-297.
22. de Prado, M.M.L., 2020. *Machine Learning for Asset Managers*. Cambridge University Press.
23. Dazeley, R., Vamplew, P., Foale, C., Young, C., Aryal, S. & Cruz, F. 2021. Levels of explainable artificial intelligence for human-aligned conversational explanations. *Artificial Intelligence*. 299:103525.
24. Dingli, A. and Fournier, K.S., 2017. Financial time series forecasting—a deep learning approach. *International Journal of Machine Learning and Computing*, 7(5), pp.118-122.
25. Drobetz, W. and Otto, T., 2021. Empirical asset pricing via machine learning: evidence from the European stock market. *Journal of Asset Management*, pp.1-32.
26. Drue, S., 2018. *A comparison of a factor-based investment strategy and machine learning for predicting excess returns on the JSE* (Master's thesis, Faculty of Science).
27. Fama, E.F., 1965. The behavior of stock-market prices. *The journal of Business*, 38(1), pp.34-105.
28. Fama, E.F., 1991. Efficient capital markets: II. *The journal of finance*, 46(5), pp.1575-1617.
29. Fior, J., Cagliero, L. and Garza, P., 2022. Leveraging Explainable AI to Support Cryptocurrency Investors. *Future Internet*, 14(9), p.251.
30. Freund, Y. and Schapire, R.E., 1997. A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. *Journal of Computer and System Sciences*, 55(1), pp.119-139.
31. Friedman, J.H., 2001. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pp.1189-1232.
32. Géron, A., 2019. *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems*. O'Reilly Media.
33. Ghosh, P., Neufeld, A. and Sahoo, J.K., 2021. Forecasting directional movements of stock prices for intraday trading using LSTM and random forests. *Finance Research Letters*, p.102280.
34. Ghosh, I., Alfaro-Cortés, E., Gámez, M. & García-Rubio, N. 2023. Prediction and interpretation of daily NFT and DeFi prices dynamics: Inspection through ensemble machine learning &

- XAI. *International Review of Financial Analysis*. 87:102558.
Available: <https://www.sciencedirect.com/science/article/pii/S1057521923000741>.
35. Gite, S., Khatavkar, H., Srivastava, S., Maheshwari, P. and Pandey, N., 2021. Stock prices prediction from financial news articles using LSTM and XAI. In *Proceedings of Second International Conference on Computing, Communications, and Cyber-Security: IC4S 2020* (pp. 153-161). Springer Singapore.
 36. Giudici, P. and Raffinetti, E., 2021. Shapley-Lorenz eXplainable artificial intelligence. *Expert Systems with Applications*, 167, p.114104.
 37. Gu, S., Kelly, B. and Xiu, D., 2020. Empirical asset pricing via machine learning. *The Review of Financial Studies*, 33(5), pp.2223-2273.
 38. Guang, L., Xiaojie, W. and Ruifan, L., 2019. Multi-scale RCNN model for financial time-series classification. *arXiv preprint arXiv:1911.09359*.
 39. Gunning, D. and Aha, D., 2019. DARPA's eXplainable Artificial Intelligence (XAI) program. *AI Magazine*, 40(2), pp.44-58.
 40. Hastie, T., Tibshirani, R., Friedman, J.H. and Friedman, J.H., 2009. *The Elements of Statistical Learning: Data mining, Inference, and Prediction* (Vol. 2, pp. 1-758). New York: Springer.
 41. Haq, A.U., Zeb, A., Lei, Z. and Zhang, D., 2021. Forecasting daily stock trend using multi-filter feature selection and deep learning. *Expert Systems with Applications*, 168, p.114444.
 42. Holzinger, A., Saranti, A., Molnar, C., Biecek, P. and Samek, W., 2022. Explainable AI methods-a brief overview. In *International Workshop on Extending Explainable AI Beyond Deep Models and Classifiers* (pp. 13-38). Springer, Cham.
 43. Hooker, S., Erhan, D., Kindermans, P.J. and Kim, B., 2019. A Benchmark for Interpretability Methods in Deep Neural Networks.
 44. Hosmer Jr, D.W., Lemeshow, S. and Sturdivant, R.X., 2013. *Applied logistic regression* (Vol. 398). John Wiley & Sons.
 45. Huang, Y., 2019. *Machine Learning for Stock Prediction Based on Fundamental Analysis* (Doctoral dissertation, The University of Western Ontario).
 46. Huang, Y., Capretz, L.F. and Ho, D., 2021, December. Machine learning for stock prediction based on fundamental analysis. In *2021 IEEE Symposium Series on Computational Intelligence (SSCI)* (pp. 01-10). IEEE.
 47. Huang, W., Nakamori, Y. and Wang, S.Y., 2005. Forecasting stock market movement direction with support vector machine. *Computers & operations research*, 32(10), pp.2513-2522.
 48. Ignatiev, A. 2020. Towards trustable explainable AI. In *Proceedings of the twenty-ninth international joint conference on artificial intelligence, IJCAI*. 5154–5158.
 49. Jiang, W., 2021. Applications of deep learning in stock market prediction: recent progress. *Expert Systems with Applications*, p.115537.
 50. *JSE Limited Listing Requirements*. 2019. JSE Limited. Available.
<https://www.jse.co.za/sites/default/files/media/documents/2019-04/JSE%20Listings%20Requirements.pdf> [Accessed 7 Dec. 2021].
 51. Juscafresa, A.N., 2022. An introduction to explainable artificial intelligence with LIME and SHAP.

52. Kaczmarek, T. and Perez, K., 2021. Building portfolios based on machine learning predictions. *Economic Research-Ekonomska Istraživanja*, pp.1-19.
53. Kamalov, F., 2020. Forecasting significant stock price changes using neural networks. *Neural Computing and Applications*, 32(23), pp.17655-17667.
54. Kamble, R.A., 2017, June. Short and long term stock trend prediction using decision tree. In *2017 International Conference on Intelligent Computing and Control Systems (ICICCS)* (pp. 1371-1375). IEEE.
55. Kannangara, K.P.M., Zhou, W., Ding, Z. and Hong, Z., 2022. Investigation of feature contribution to shield tunneling-induced settlement using Shapley additive explanations method. *Journal of Rock Mechanics and Geotechnical Engineering*.
56. Karabiber F, 2021-last update, Dummy Variable Trap.
Available: <https://www.learndatasci.com/glossary/dummy-variable-trap/> [Accessed 20 Jan, 2022].
57. Kim, K.J., 2003. Financial time series forecasting using support vector machines. *Neurocomputing*, 55(1-2), pp.307-319.
58. Kingsford, C. and Salzberg, S.L., 2008. What are decision trees?. *Nature biotechnology*, 26(9), pp.1011-1013.
59. Koehrsen W., Mar 3, 2018-last update, *Beyond Accuracy: Precision and Recall*.
Available: <https://towardsdatascience.com/beyond-accuracy-precision-and-recall-3da06bea9f6c> [Jan 10, 2022].
60. Kolanovic, M. and Krishnamachari, R., 2017. Big data and AI strategies: Machine learning and alternative data approach to investing. *JP Morgan Global Quantitative & Derivatives Strategy Report*, 25.
61. Kotsiantis, S.B., 2013. Decision trees: a recent overview. *Artificial Intelligence Review*, 39, pp.261-283.
62. Kumbure, M.M., Lohrmann, C., Luukka, P. and Porras, J., 2022. Machine learning techniques and data for stock market forecasting: A literature review. *Expert Systems with Applications*, 197, p.116659.
63. Kumar, I., Dogra, K., Utreja, C. and Yadav, P., 2018. A comparative study of supervised machine learning algorithms for stock market trend prediction. In *2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT)* (pp. 1003-1007). IEEE.
64. Kumar, D., Sarangi, P.K. and Verma, R., 2021. A systematic review of stock market prediction using machine learning and statistical techniques. *Materials Today: Proceedings*.
65. Kumar, S., Vishal, M. and Ravi, V., 2022. Explainable Reinforcement Learning on Financial Stock Trading using SHAP. arXiv preprint arXiv:2208.08790.
66. Kumbure, M.M., Lohrmann, C., Luukka, P. and Porras, J., 2022. Machine learning techniques and data for stock market forecasting: A literature review. *Expert Systems with Applications*, 197, p.116659.
67. Lee, D., 2020. *Build Your Own AI Investor: With Machine Learning and Python*.
68. Liu, Q., Tao, Z., Tse, Y. and Wang, C., 2021. Stock market prediction with deep learning: The case of China. *Finance Research Letters*, p.102209.

69. Livieris, I.E., Kotsilieris, T., Stavroyiannis, S. and Pintelas, P., 2020. Forecasting stock price index movement using a constrained deep neural network training algorithm. *Intelligent Decision Technologies*, (Preprint), pp.1-11.
70. Lobão, J., 2018. Are African stock markets inefficient? New evidence on seasonal anomalies. *Scientific Annals of Economics and Business*, 65(3), pp.283-301.
71. Lohrmann, C. and Luukka, P., 2019. Classification of intraday S&P500 returns with a Random Forest. *International Journal of Forecasting*, 35(1), pp.390-407.
72. Lundberg, S.M., Erion, G.G. and Lee, S.I., 2018. Consistent individualized feature attribution for tree ensembles. arXiv preprint arXiv:1802.03888.
73. Lundberg, S.M. and Lee, S.I., 2017, December. A unified approach to interpreting model predictions. In *Proceedings of the 31st international conference on neural information processing systems* (pp. 4768-4777).
74. Ma, Y., Han, R. and Wang, W., 2021. Portfolio optimization with return prediction using deep learning and machine learning. *Expert Systems with Applications*, 165, p.113973.
75. Markowitz, H., 1952. The utility of wealth. *Journal of political Economy*, 60(2), pp.151-158.
76. Moffitt, S.D., 2018. Why Markets are Inefficient: A Gambling" Theory" of Financial Markets For Practitioners and Theorists. *arXiv preprint arXiv:1801.01948*.
77. Molnar, C., 2020. *Interpretable Machine Learning*. Lulu. com.
78. Nabipour, M., Nayyeri, P., Jabani, H., Shahab, S. and Mosavi, A., 2020. Predicting stock market trends using machine learning and deep learning algorithms via continuous and binary data; a comparative analysis. *IEEE Access*, 8, pp.150199-150212.
79. Nasa, C. and Suman, S., 2012. Evaluation of Different Classification Techniques for Web Data. *International Journal of Computer Applications*, 52(9), pp.34-40.
80. Natekin, A. and Knoll, A., 2013. Gradient boosting machines, a tutorial. *Frontiers in Neurorobotics*, 7, p.21.
81. Ntakaris, A., Mirone, G., Kanninen, J., Gabbouj, M. and Iosifidis, A., 2019. Feature engineering for mid-price prediction with deep learning. *IEEE Access*, 7, pp.82390-82412.
82. Nti, I.K., Adekoya, A.F. and Weyori, B.A., 2020. A systematic review of fundamental and technical analysis of stock market predictions. *Artificial Intelligence Review*, 53(4), pp.3007-3057.
83. Ou, P. and Wang, H., 2009. Prediction of stock market index movement by ten data mining techniques. *Modern Applied Science*, 3(12), pp.28-42.
84. Ozbayoglu, A.M., Gudelek, M.U. and Sezer, O.B., 2020. Deep learning for financial applications: A survey. *Applied Soft Computing*, 93, p.106384.
85. Patel, J., Shah, S., Thakkar, P. and Kotecha, K., 2015. Predicting stock and stock price index movement using trend deterministic data preparation and machine learning techniques. *Expert systems with applications*, 42(1), pp.259-268.
86. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V. and Vanderplas, J., 2011. Scikit-learn: Machine Learning in Python. *The Journal of Machine Learning Research*, 12, pp.2825-2830. Retrieved from <https://scikit-learn.org/stable/>. Accessed 15 Mar. 2023.

87. Peng, Y., Albuquerque, P.H.M., Kimura, H. and Saavedra, C.A.P.B., 2021. Feature selection and deep neural networks for stock price direction forecasting using technical analysis indicators. *Machine Learning with Applications*, p.100060.
88. Peng, C.Y.J., Lee, K.L. and Ingersoll, G.M., 2002. An Introduction to Logistic Regression Analysis and Reporting. *The Journal of Educational Research*, 96(1), pp.3-14.
89. Qiu, M. and Song, Y., 2016. Predicting the direction of stock market index movement using an optimized artificial neural network model. *PloS one*, 11(5), p.e0155133.
90. Rasekhschaffe, K.C. and Jones, R.C., 2019. Machine learning for stock selection. *Financial Analysts Journal*, 75(3), pp.70-88.
91. Saeed, W. and Omlin, C., 2021. Explainable ai (xai): A systematic meta-survey of current challenges and future opportunities. arXiv preprint arXiv:2111.06420.
92. Saini, A. and Sharma, A., 2022. Predicting the unpredictable: an application of machine learning algorithms in Indian stock market. *Annals of Data Science*, 9(4), pp.791-799.
93. Schwenk, H. and Bengio, Y., 2000. Boosting neural networks. *Neural computation*, 12(8), pp.1869-1887.
94. Sezer, O.B., Gudelek, M.U. and Ozbayoglu, A.M., 2020. Financial time series forecasting with deep learning: A systematic literature review: 2005–2019. *Applied Soft Computing*, 90, p.106181.
95. Schütze, H., Manning, C.D. and Raghavan, P., 2008. *Introduction to information retrieval* (Vol. 39, pp. 234-265). Cambridge: Cambridge University Press.
96. Shleifer, A., 2000. *Inefficient markets: An introduction to behavioural finance*. Oup Oxford.
97. Shynkevich, Y., McGinnity, T.M., Coleman, S.A., Belatreche, A. and Li, Y., 2017. Forecasting price movements using technical indicators: Investigating the impact of varying input window length. *Neurocomputing*, 264, pp.71-88.
98. Sloan, R.G., 2019. Fundamental analysis redux. *The Accounting Review*, 94(2), pp.363-377.
99. Smith, M. and Alvarez, F., 2021. Identifying mortality factors from Machine Learning using Shapley values—a case of COVID19. *Expert Systems with Applications*, 176, p.114832.
100. Teixeira, L.A. and De Oliveira, A.L.I., 2010. A method for automatic stock trading combining technical analysis and nearest neighbor classification. *Expert systems with applications*, 37(10), pp.6885-6890.
101. Thakkar, A. and Chaudhari, K., 2021. A comprehensive survey on deep neural networks for stock market: The need, challenges, and future directions. *Expert Systems with Applications*, 177, p.114800.
102. Walter, C.P., 2000, November. The efficient market hypothesis, the Gaussian assumption, and the investment management industry. In *EFMA 2001 Lugano Meetings*.
103. Wand, T., Heßler, M. and Kamps, O., 2022. Identifying Dominant Industrial Sectors in Market States of the S&P 500 Financial Data. *arXiv preprint arXiv:2208.14106*.
104. Wang, Y. and Feng, L., 2020. Improved Adaboost algorithm for classification based on noise confidence degree and weighted feature selection. *IEEE Access*, 8, pp.153011-153026.
105. Wang, W., Li, W., Zhang, N. and Liu, K., 2020. Portfolio formation with preselection using deep learning from long-term financial data. *Expert Systems with Applications*, 143, p.113042.

106. Wu, J.M.T., Li, Z., Srivastava, G. and Lin, J.C.W., 2020. A Tool based on ML-driven Graphical Model for Stock Price Prediction by Leading Indicators. *IFAC-PapersOnLine*, 53(5), pp.692-697.
107. Yadav D, Dec 6, 2019-last update, *Categorical Encoding using Label-Encoding and One-Hot-Encoder*. Available: <https://towardsdatascience.com/categorical-encoding-using-label-encoding-and-one-hot-encoder-911ef77fb5bd> [Dec 3, 2020].
108. Yang, Y., Hu, X. and Jiang, H., 2022. Group penalized logistic regressions predict up and down trends for stock prices. *The North American Journal of Economics and Finance*, 59, p.101564.
109. Yun, K.K., Yoon, S.W. and Won, D., 2021. Prediction of stock price direction using a hybrid GA-XGBoost algorithm with a three-stage feature engineering process. *Expert Systems with Applications*, 186, p.115716.
110. Zhong, X. and Enke, D., 2019. Predicting the daily return direction of the stock market using hybrid machine learning algorithms. *Financial Innovation*, 5(1), pp.1-20.

Appendix A

A1. Chapter 2: Summary of Research - An Examination of Classification Algorithms Applied to Primarily Technical Data

Table A1: Summary of Research: An Examination of Classification Algorithms Applied to Primarily Technical Data

No.	Paper	Data Set	Features	Algorithms	Target Variable and Classes	Prediction Window	Performance Criteria	Result
1	Abe and Nakayama, 2018	Constituents of the MSCI Japan Index 1990-2016	Technical and Fundamental	NN, DNN, RF, SVR,	Share Return Tercile and Quintile	1 month	MSE, Return, Risk, Return divided by Risk	DNN8_4 had the marginally best directional accuracy for both tercile and quintile classes over RF. RF outperformed all the other algorithms with a Tercile Return percentage of 9.04%.
2	Basak et al., 2019	10 Listed shares on US Stock Market Date from listing to 2017	Technical	RF, XGBoost	Share Price Binary (increase and decrease)	3,5,10,15,30,60,90 days	Accuracy, Recall, Precision, F-Score, Brier Score, AUC	RF outperforms XGBoost with a 78% accuracy. 90-day prediction window was the most accurate.
3	Borovkov and Tsiamas, 2019	22 Listed Shares on US Large Cap Index, traded on NASDAQ or NYSE 2014-2014	Technical (high frequency)	Two stacked LSTMs, Lasso LR, Ridge LR	Share Price Binary (buy and sell)	5 minutes	AUC	Multiple (performance weighted ensemble) of LSTM algorithms outperformed both the Lasso and Ridge LR algorithms for intraday directional price predictions.
4	Chen and He, 2018	Listed Shares on Chinese Stock Market 2015-2017	Technical	CNN	Share Price Binary (up and down)	10 days	Accuracy, F1	CNN algorithm produces an accuracy of above 73.2% across all data source on whether a share price will go up or down after 10 days.
5	Chen and Huang, 2021	S&P 500 Index 2010-2019	Technical	LSTM, CNN	Share Price Index Binary (up and down) and tercile (buy, sell and hold)	10 and 20 days	Accuracy, Precision, Recall, F1, Return	CNN and LSTM equally accurate at predicting rise or fall of S&P 500 Index in next 10 to 20 days with 67% accuracy. CNN8 outperformed all other algorithms with a ROI of 13.23% with buy, hold and sell strategy.

No.	Paper	Data Set	Features	Algorithms	Target Variable and Classes	Prediction Window	Performance Criteria	Result
6	Chen et al., 2003	Taiwan Stock Index 1982-1992	Technical and Economic	Probabilistic Neural Network (PNN), Generalised Method of Moments (GMM)	Share Price Index Binary (negative and positive)	3, 6 and 12 months	Accuracy, Return	PNN demonstrated the strongest predictive power for direction of index returns.
7	Choi and Renelle, 2019	Listed Shares on the Russell 1000 Index 1996-2017	Technical	Convolutional Neural Network (CNN), Recurrent Neural Network (RNN).	Share Returns Tercile and quintile	1 month	Quintile Annualised Return	The RNN algorithm improves annual returns on conventional price momentum portfolio. Quintile portfolio outperforms the tercile portfolio's annualised return by 0.48%. Quintile portfolio achieves 15.92% annual return.
8	Dingli and Fournier, 2017	World Indices 2003-2016	Technical and Economic	CNN, Logistic Regression, SVM	Share Price Binary (up and down)	1 week and 1 month	Accuracy	SVM and LR outperformed the CNN algorithm. CNN achieves accuracy of 65% for next month price direction and 60% for the next week price direction.
9	Drobetz and Otto, 2020	Shares listed in 19 Eurozone countries 1990-2020	Technical and Fundamental	OLS, RF, Gradient Boosted Regression Trees, Artificial Neural Network (ANN), SVM (Support Vector Machine)	Share Return Deciles	1 month	Mean Squared Error (MSE)	SVM outperforms OLS equal weighted long portfolio algorithm by 43.28%. SVM outperforms ANN equal weighted long portfolio algorithm by 6.39%
10	Ghosh et al., 2021	Constituents of the S&P 500 1990-2018	Technical	LSTM, RF	Share Price Binary (outperform or underperform)	Intraday	Return	LSTM achieved a return of 69.67% and RF was 65.85% for positive returns based on whether share will outperform the cross-sectional median.

No.	Paper	Data Set	Features	Algorithms	Target Variable and Classes	Prediction Window	Performance Criteria	Result
11	Guang et al., 2019	Shanghai Stock Exchange Composite Index, Shenzhen Stock Exchange Small and Medium Index, Shenzhen ChiNext Price Index 2016-2016	Technical	Multi-Scale Temporal Dependent Recurrent Convolutional Neural Network (MSTD-RCNN), SVM, RF, MS-CNN, TreNet, FDNN	Share Price Terciles (downward, still and upward)	10,20,30,40 and 50 days	Accuracy, F1 and Confusion Matrix, Return	MSTD-RCNN achieved the highest profit on all three datasets. MSTD-RCNN had highest upward and downward classification accuracy. MSTD-RCNN achieved highest accuracy and F1 across all datasets along with the best return. The best accuracy was achieved with the 30-day prediction window
12	Huang et al., 2005	Nikkei 225 Index 1990-2002	Technical and Economic	SVM, Back Propagation Neural Network (BPM), Linear and Quadratic Discriminant Analysis (LDA, QDA)	Share Price Index Binary (1 and -1)	1 week	Hit ratio	SVM outperformed all other algorithms with a hit ratio of 73%. Ensemble algorithm achieved 75% hit ratio.
13	Kamalov, 2020	Four major US listed shares (Coca-Cola, Cisco Systems, Nike, and Goldman Sachs) Daily Share Prices. 2009-2019	Technical	CNN, LSTM, MLP, RF	Share Price Binary (positive and negative)	7,14,30 and 60 days	Relative Strength Index (RSI), AUC	LSTM algorithm outperformed other algorithms overall in predicting significant changes in price.
14	Kamble, 2017	Listed shares on NSE and BSE, 5 years of data.	Technical and Fundamental	RF, ID3 (J48)	Share Price Tercile (buy, sell and hold)	1 year	Out of Bag Error (OOB),	RF had 66.8% accuracy on Buy signal for short-term trend prediction. The long-term algorithm outperforms the short-term trend prediction algorithm.
15	Kim, 2003	Korean Composite Stock Price	Technical	SVM, Back Propagation Neural	Share Price Index	1 day	Hit Ratio	SVM outperformed the other methods for

No.	Paper	Data Set	Features	Algorithms	Target Variable and Classes	Prediction Window	Performance Criteria	Result
		Index (KOSPI) 1989-1998		Network (BPM), Case Based Reasoning (CBR)	Binary (higher or lower)			direction of the daily price of the KOSPI.
16	Kumar et al., 2018	4 Listed shares – Amazon, Bata, Bosch and Cipla. 5 to 10 years of data.	Technical	SVM, RF, KNN, NB and ANN	Share Price Trend (details not provided)	1 to 90 days	Accuracy and F-Score	RF was best for large datasets (Accuracy 72.36%, F-Score 80.55%) and NB for small datasets (Accuracy 70.80%, F-Score 60.42%). Reducing input features, reduces accuracies.
17	Liu et al., 2021	Listed Shares on Shanghai and Shenzhen Exchanges 2009-2020	Technical and Fundamental	Single-layer ANN, DLNN	Share Returns Decile	1 month	Accuracy	DLNN achieves a 55.46% accuracy beating the ANN.
18	Lohrmann and Luukka, 2019	S&P 500 Index, STOXX50 (Europe), Hang Seng (China), Nikkei 225 (Japan), FTSE100 (United Kingdom), DAX (Germany), US Commodities Index, Exchange Rates, Interest Rates and Bonds 2010-2018	Technical and Economic	RF, KNN, Naïve Bayes, Decision Tree	Share Index Return Quartile (strong positive, positive, negative, strong negative)	1 day	Accuracy, Mean Accuracy, Performance return against Buy-and-Hold	RF outperformed all other methods to predict S&P 500 closing return. Extreme classes ('strong positive' and 'strong negative') resulted in higher average returns.
19	Ntakaris et al., 2019	5 Nordic and 2 US Listed Shares US Shares: Sep 2015-Oct 2015	Technical and Economic	MLP, CNN, LSTM	1. Share Mid-price Binary (up and down). 2. Using regression: Number of events in the	Intraday	F1 for classification, RMSE for regression	MLP_4 outperforms CNN and LSTM algorithms for classification. LSTM_2 outperforms the MLP and CNN algorithms for regression.

No.	Paper	Data Set	Features	Algorithms	Target Variable and Classes	Prediction Window	Performance Criteria	Result
		Nordic: Jun 2010-Jun 2010			future when the movement will take place.			
20	Ou and Wang, 2007	Hong Kong Hang Seng Index, Currency Rates 2000-2006	Technical and Economic	Least Squares SVM, SVM, QDA, LDA, K-Nearest Neighbour, Naïve Bayes	Share Price Index Binary (next price is larger and next price is smaller)	1 day	Accuracy, Hit Rate, MSE	LS SVM and SVM outperformed the other algorithms. SVM best for out of sample modelling.
21	Patel et al., 2015	Listed Indian Shares and CNX Nifty and BSE Sensex on Bombay Stock Exchange 2003-2012	Technical	ANN, SVM, RF, Naïve Bayes	Share Price and Share Price Index Binary (up and down)	1 day	Accuracy, F1	RF overall outperforms the other algorithms. Results improve using trend deterministic data.
22	Peng et al., 2021	S&P 100, FTSE 100, CAC 40, DAX-30, NIKKEI 225 and Bovespa Indices and the Top 50 assets from China's SSE 180 Index. 2008-2019	Technical	DNN	Share Price Binary (increase and decrease)	1 day	Accuracy, Return	DNN was not able to outperform a 'Buy-and-Hold' strategy for share price movement prediction. Out-of-sample accuracy converged on either 50% or 65%.
23	Qiu and Song, 2016	Nikkei 225 Index 2007-2013	Technical	ANN, GA-ANN	Share Price Index: Two classes (up or down)	1 day	Hit Ratio	Best performing GA-ANN hybrid achieved hit ratio of 81.27% in predicting the daily direction of the Nikkei 225 Index.
24	Shynkevich et al., 2017	50 Randomly Selected Shares 2002-2012	Technical	SVM, ANN, KNN.	Share Price Binary (up and down) and terciles (up, no move and down)	1,3,5,7,10,15,20,25 and 30 days	Accuracy, Winning Rate, Return per Trade, Sharpe Ratio	SVN outperforms all the other algorithms when the input window length is equal to the prediction horizon/window. The best accuracy achieved:

No.	Paper	Data Set	Features	Algorithms	Target Variable and Classes	Prediction Window	Performance Criteria	Result
								SVM 20-day 2 class algorithm with 74.64% accuracy.
25	Teixeira and de Oliveira, 2010	15 Shares on Sao Paulo Stock Exchange (Bovespa) 1998-2009	Technical	KNN	Share Price Tercile (buy, keep and sell)	1 day	Accuracy, Return	KNN algorithm with stop loss and stop gain and the RSI filter outperformed the 'Buy-and-Hold' strategy by 58.10%
26	Haq et al., 2021	Listed shares on NASDAQ (88) 2014-2016	Technical and Tweets	SVM, RF, Logistic Regression, Generative Algorithm	Share Price Binary (rise or fall)	1 day	Classification Accuracy, Mathews Correlations Coefficient (MCC)	Combined multi-filter feature selection from LR, RF and SVM for input into GM. GM produced highest accuracy of 59.44%.
27	Wu et al., 2020	5 American and Taiwanese Shares	Technical	Stock Sequence Array Convolutional Neural Network (SSACNN), CNN, SVM	Share Price Tercile (increasing, decreasing and within range)	30 days	Accuracy	SSACNN outperforms both the CNN and SVM algorithms.
28	Yang et al., 2022	BAC, Amazon and Citibank shares from Yahoo Finance 2010-2019	Technical	Smooth Clipped Absolute Deviation /Minmax Concave Penalty (SCAD/MCP) Penalised Regression.	Share Price Binary (up and down)	1 day	Accuracy, AUC	Group SCAD/MCP penalized logistic regression accuracies exceeded 71% and AUC exceeded 0.78 for next day 'up or down' trend prediction.
29	Yun et al., 2021	Korean Composite Stock Price Index (KOSPI) 2000-2016	Technical	GA-XGBoost	Share Price Index Binary (up or down)	1 day	Accuracy, Precision, Recall, F1, AUC	GA-XGBoost achieved an accuracy of 93.28% in predicting next day 'up or down' movement.
30	Zhong and Enke, 2019	SPDR S&P 500 ETF, Treasury Bills 2003- 2013	Technical and Economic	DNN, Multilayer Feed-Forward Neural	Share Index Return Binary (up or down)	1 day	Accuracy, MSE, Return, Sharpe Ratio	PCA-DNN with 10 hidden layers highest classification accuracy, daily return of 0.7%, Sharpe Ratio of 0.105

No.	Paper	Data Set	Features	Algorithms	Target Variable and Classes	Prediction Window	Performance Criteria	Result
				Network, PCA				

A2. Chapter 2: Machine Learning Algorithms

Random Forest: Mathematical Representation

The mathematical representation of the Random Forest algorithm is as follows (Breiman, 2001):

A Random Forest algorithm constructs an ensemble of decision trees to make predictions for a given dataset $X = \{x_1, x_2, \dots, x_n\}$ with corresponding labels $Y = \{y_1, y_2, \dots, y_n\}$. A series of decision trees are built $\{T_1, T_2, \dots, T_B\}$. Each decision tree T_b is trained on a different bootstrap sample S_b drawn from X . Each decision tree T_b is constructed using the following algorithm:

For $b = 1$ to B :

Construct a bootstrap sample S_b by selecting N instances with replacement.

Build a decision tree T_b on S_b .

For each node in the decision tree:

Randomly choose m variables out of all p variables.

Identify the optimal split using the selected m variables.

Generate two child nodes from the current node.

The final Random Forest model combines the predictions from all trees. For classification tasks, the predictions $\{T_1(x'), T_2(x'), \dots, T_B(x')\}$ is taken as the final prediction for each input x' . For regression tasks, the average prediction is used as the final prediction for x' .

Random Forest Classifier: Parameters

The available parameters for the sci-kit learn Random Forest Classifier module are provided in the Table titled 'Random Forest Classifier Parameters'.

Table Random Forest Classifier Parameters (Pedregosa et al., 2011)

Parameter	Description
<i>n_estimator</i>	The number of trees in the forest.
<i>criterion</i>	The function to measure the quality of a split.
<i>max_depth</i>	The maximum depth of the tree.
<i>min_samples_split</i>	The minimum number of samples required to split an internal node.

Parameter	Description
<i>min_samples_leaf</i>	The minimum number of samples required to be at a leaf node.
<i>min_weight_fraction_leaf</i>	The minimum number of samples required to be at a leaf node.
<i>max_features</i>	The number of features to consider when looking for the best split.
<i>max_leaf_nodes</i>	Grow trees with <i>max_leaf_nodes</i> in best-first fashion.
<i>min_impurity_decrease</i>	A node will be split if this split induces a decrease of the impurity greater than or equal to this value.
<i>bootstrap</i>	Whether bootstrap samples are used when building trees.
<i>oob_score</i>	Whether to use out-of-bag samples to estimate the generalization score.
<i>n_jobs</i>	The number of jobs to run in parallel.
<i>random_state</i>	Controls both the randomness of the bootstrapping of the samples used when building trees.
<i>verbose</i>	Controls the verbosity when fitting and predicting.
<i>warm_start</i>	When set to True, reuse the solution of the previous call to fit and add more estimators to the ensemble, otherwise, just fit a whole new forest.
<i>class_weight</i>	Weights associated with classes.
<i>ccp_alpha</i>	Complexity parameter used for Minimal Cost-Complexity Pruning.
<i>max_samples</i>	If bootstrap is True, the number of samples to draw from X to train each base estimator.

GradBoost: Mathematical Representation

The mathematical representation of the GradBoost algorithm is as follows (Friedman, 2001):

1. Initialise the model with a constant value:

$$F_0(x) = \operatorname{argmin}_y \sum_{i=1}^N L(y_i, \gamma). \quad (8)$$

2. For $m=1$ to M :
 - a. Compute the negative gradient:

$$r_{im} = - \left[\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)} \right]_{F(x)=F_{m-1}(x)}. \quad (9)$$

for $i = 1, \dots, N$.

- b. Fit base learner to the decision tree negative gradient to get $h_m(x)$.
 - c. Compute the multiplier γ that minimises the loss when added to the previous models:

$$\gamma_m = \operatorname{argmin}_\gamma \sum_{i=1}^N L(y_i, F_{m-1}(x_i) + \gamma h_m(x_i)). \quad (10)$$

- d. Update the model:

$$F_m(x) = F_{m-1}(x) + \gamma_m h_m(x). \quad (11)$$

3. Final model:

$$F_M(x) = F_0(x) + \sum_{m=1}^M \gamma_m h_m(x). \quad (12)$$

Where:

The differentiable loss function is L .

The model after m boosting rounds is $F_m(x)$.

The output of the m -th learner designed to improve the model by addressing the residual errors from the previous ensemble model is $h_m(x)$.

The coefficient or weight assigned to the m -th learner is γ_m .

The total number of models is M where each m corresponds to a specific step or stage in the ensemble building process.

The input data on which the prediction is made is x .

GradBoost Classifier: Parameters

The available parameters for the sci-kit learn GradBoost Classifier module are provided in the Table titled ‘GradBoost Classifier Parameters’.

Table GradBoost Classifier Parameters (Pedregosa et al., 2011)

Parameter	Description
<i>loss</i>	The loss function that needs to be optimized. Log loss is used for classification.
<i>learning_rate</i>	Learning rate is used to shrink the contribution of each tree.
<i>n_estimators</i>	The number of boosting stages to perform.
<i>subsample</i>	The fraction of samples to be used for fitting the individual base learners.
<i>criterion</i>	The function to measure the quality of a split.
<i>min_sample_split</i>	The minimum number of samples required to split an internal node.
<i>min_sample_leaf</i>	The minimum number of samples required to be at a leaf node.
<i>min_weight_fraction_leaf</i>	The minimum weighted fraction of the sum total of weights (of all the input samples) required to be at a leaf node.
<i>max_depth</i>	Maximum depth of the individual regression estimators. The maximum depth limits the number of nodes in the tree.
<i>min_impurity_decrease</i>	A node will be split if this split induces a decrease of the impurity greater than or equal to this value.
<i>initestimator</i>	An estimator object that is used to compute the initial predictions.
<i>random_state</i>	Controls the random seed given to each Tree estimator at each boosting iteration.
<i>max_features</i>	The number of features to consider when looking for the best split.
<i>verbose</i>	Enable verbose output.
<i>max_leaf_nodes</i>	Grow trees with <i>max_leaf_nodes</i> in best-first fashion. Best nodes are defined as relative reduction in impurity.
<i>warm_start</i>	When set to True, reuse the solution of the previous call to fit and add more estimators to the ensemble, otherwise, just erase the previous solution.
<i>validation_fraction</i>	The proportion of training data to set aside as validation set for early stopping.

<i>n_iter_no_change</i>	Used to decide if early stopping will be used to terminate training when validation score is not improving.
<i>tol</i>	Tolerance for the early stopping.
<i>ccp_alpha</i>	Complexity parameter used for Minimal Cost-Complexity Pruning.

XGBoost: Mathematical Representation

The mathematical representation of the objective function in the XGBoost algorithm is as follows (Chen and Guestrin, 2016):

$$\text{XGBoost obj}(\theta) = \sum_{i=1}^n l(y_i, \hat{y}_i^{(t)}) + \sum_{i=1}^t \Omega(f_i). \quad (13)$$

The variables for the formulae are described below in the Table titled ‘XGBoost Formula Variables’.

Table XGBoost Formula Variables

Variable	Description
$l(y_i, \hat{y}_i^{(t)})$	The loss function, comparing the true label y_i and the prediction $\hat{y}_i^{(t)}$.
y_i	True label.
$\hat{y}_i^{(t)}$	The prediction.
$\Omega(f)$	The regularisation term.
t	The number of leaves in the tree.
$\Omega(f) = \gamma T + \frac{1}{2} \lambda \sum_{j=1}^T w_j^2$ represents the regularisation term, including:	
T	The number of trees.
w_j	The score on the j-th leaf.
γ	Regularisation parameter.
λ	Regularisation parameter.

XGBoost: Parameters

The available parameters for the XGBoost module are provided in the Table titled ‘XGBoost Parameters’.

Table XGBoost Parameters (Chen and Guestrin, 2016)

Parameter	Description
<i>booster</i>	The booster to use e.g. gbtrees, gblinear or dart.
<i>validate_parameters</i>	When set to True, XGBoost will perform validation of input parameters to check whether a parameter is used or not
<i>nthread</i>	Number of parallel threads used to run XGBoost.
<i>disable_default_eval_metric</i>	Flag to disable default metric.
<i>learning_rate</i> (alias <i>eta</i>)	Step size shrinkage used in update to prevents overfitting.
<i>min_split_loss</i> (alias <i>gamma</i>)	Minimum loss reduction required to make a further partition on a leaf node of the tree.
<i>max_depth</i>	Maximum depth of a tree. Increasing this value will make the model more complex and more likely to overfit.
<i>min_child_weight</i>	Minimum sum of instance weight (hessian) needed in a child. If the tree partition step results in a leaf node with the sum of instance weight less than <i>min_child_weight</i> , then the building process will give up further partitioning.
<i>max_delta_step</i>	Maximum delta step we allow each leaf output to be.
<i>subsample</i>	Subsample ratio of the training instances.
<i>sampling_method</i>	The method to use to sample the training instances.
<i>colsample_bytree</i> , <i>colsample_bylevel</i> , <i>colsample_bynode</i>	This is a family of parameters for subsampling of columns.
<i>tree_method</i>	The tree construction algorithm used in XGBoost.

Parameter	Description
<i>scale_pos_weight</i>	Control the balance of positive and negative weights, useful for unbalanced classes.
<i>updater</i>	A comma separated string defining the sequence of tree updaters to run, providing a modular way to construct and to modify the trees.
<i>refresh_leaf</i>	This is a parameter of the refresh updater.
<i>process_type</i>	A type of boosting process to run.
<i>grow_policy</i>	Controls the way new nodes are added to the tree.
<i>max_leaves</i>	Maximum number of nodes to be added.
<i>max_bin</i>	Maximum number of discrete bins to bucket continuous features.
<i>num_parallel_tree</i>	Number of parallel trees constructed during each iteration.
<i>monotone_constraints</i>	Constraint of variable monotonicity
<i>interaction_constraints</i>	Constraints for interaction representing permitted interactions.

AdaBoost: Mathematical Representation

The mathematical representation of the AdaBoost algorithm is as follows (Hastie et al., 2009):

Input for each dataset with N number of samples, the weight of each data point is $w^i = \frac{1}{N}$

For $m = 1$ to M :

- i. Sample the dataset using weights $w_i^{(m)}$ to obtain the training samples x_i .
- ii. Fit a classifier K_m using all the training samples x_i .
- iii. Compute:

$$\epsilon = \frac{\sum_{y_i \neq K_m(x_i)} w_i^{(m)}}{\sum_{y_i} w_i^{(m)}}. \quad (14)$$

where y_i is the ground truth value of the target variable and $w_i^{(m)}$ is the weight of the sample i at iteration m .

- iv. Compute:

$$\alpha_m = \frac{1}{2} \ln \frac{1 - \epsilon}{\epsilon}. \quad (15)$$

v. Update all weights:

$$w_i^{(m+1)} = w_i^{(m)} e^{-\alpha_m y K_m(x)}. \quad (16)$$

vi. New predictions are computed by:

$$K(x) = \text{sign} \left[\sum_{m=1}^M \alpha_m K_m(x) \right]. \quad (17)$$

AdaBoost: Parameters

The available parameters for the sci-kit learn AdaBoost Classifier are provided in the Table titled ‘AdaBoost Classifier Parameters’.

Table AdaBoost Classifier Parameters (Pedregosa et al., 2011)

Parameter	Description
<i>estimator</i>	The base estimator from which the boosted ensemble is built.
<i>n_estimators</i>	The maximum number of estimators at which boosting is terminated.
<i>learning_rate</i>	Weight applied to each classifier at each boosting iteration. A higher learning rate increases the contribution of each classifier.
<i>algorithm</i>	‘SAMME or SAMMER.R.’ algorithm. The SAMME.R algorithm typically converges faster than SAMME, achieving a lower test error with fewer boosting iterations.
<i>random_state</i>	Controls the random seed given to each Tree estimator at each boosting iteration.

Decision Tree: Mathematical Representation

The mathematical representation of the Decision Tree algorithm, which involves repeatedly splitting the data based on input features, is as follows (James et al., 2013):

A Decision Tree algorithm makes predictions for a given input features $X = \{x_1, x_2, \dots, x_n\}$ with the target variable to predict $Y = \{y_1, y_2, \dots, y_n\}$.

Each internal N_i in the decision tree acts as a decision point, such as $x_j < \theta$ where x_j is a feature and θ is a threshold. This process of splitting the data based on conditions at internal nodes is known as ‘branching’ and it forms the core of the decision tree algorithm.

The decision tree algorithm aims to identify the most informative feature and its corresponding threshold value that effectively splits the data into subsets. This process is guided by specific criteria, such as Gini impurity or entropy for classification tasks and variance reduction for regression.

In classification tasks, the Gini impurity at node N can be described mathematically as:

$$G(N) = 1 - \sum_{i=1}^k p_i^2. \quad (10)$$

Where p represents the proportion of samples in node N that belong to class i . The variable k denotes the total number of classes present in the data.

In regression tasks, the decision tree typically aims to minimise the MSE at each split. The MSE quantifies the average squared difference between the predicted values and the actual values. For node N , the MSE can be described mathematically as:

$$\text{MSE}(N) = \frac{1}{|N|} \sum_{i \in N} (y_i - \bar{y}_N)^2. \quad (11)$$

Where $|N|$ represents the total number of samples within node N . The variable y_i denotes the value of the target variable for sample i . The symbol $\bar{y}_N$ is the mean value of Y for samples N .

The decision tree learning process involves a recursive top-down approach:

1. Begin the decision tree construction by selecting the most discriminating feature and its corresponding threshold value at the root node that will split the data into two child nodes.
2. Assign each subset of the data to its corresponding child node.
3. Recursively repeat the feature selection and splitting procedure for each child node until one of the pre-defined stopping criteria is reached.

The final decision tree model M can be represented as a series of decision rules that can be applied to an input vector x' to make a prediction about its corresponding output value. This can be described mathematically as:

$$M(x') = y'. \quad (12)$$

The predicted value, represented by y' , for a given input data point x' , is determined by traversing the decision tree from the root node to a leaf node based on the decision rules specified at each node.

Decision Tree Classifier: Parameters

The available parameters for the sci-kit learn Decision Tree Classifier module are provided in the Table titled 'Decision Tree Classifier Parameters'.

Table Decision Tree Classifier Parameters (Pedregosa et al., 2011)

Parameter	Description
<i>criterion</i>	The function to measure the quality of a split.
<i>splitter</i>	The strategy used to choose the split at each node.
<i>max_depth</i>	The maximum depth of the tree.
<i>min_samples_split</i>	The minimum number of samples required to split an internal node.
<i>min_samples_leaf</i>	The minimum number of samples required to be at a leaf node.
<i>min_weight_fraction_leaf</i>	The minimum weighted fraction of the sum total of weights (of all the input samples) required to be at a leaf node.
<i>max_features</i>	The number of features to consider when looking for the best split.
<i>random_state</i>	Controls the randomness of the estimator.
<i>max_leaf_nodes</i>	Grow a tree with <i>max_leaf_nodes</i> in best-first fashion.
<i>min_impurity_decrease</i>	A node will be split if this split induces a decrease of the impurity greater than or equal to this value.
<i>class_weight</i>	Weights associated with classes.
<i>ccp_alpha</i>	Complexity parameter used for Minimal Cost-Complexity Pruning.

Support Vector Classifier: Mathematical Representation

The mathematical representation of a p-dimensional hyperplane is as follows (Cortes and Vapnik, 1995):

$$b_0 + \sum_{j=1}^p b_j x_j = 0. \quad (13)$$

The mathematical representation of the SVM classification function is as follows:

$$f(x) = \sum_{i=1}^N \alpha_i y_i K(x, x_i) - b. \quad (14)$$

The mathematical representation of the Linear kernel is as follows:

$$K(x, x_i) = (x, x'). \quad (15)$$

The mathematical representation of the Polynomial kernel is as follows:

$$K(x, x_i) = e^{\frac{-|x - x_i|^2}{2\beta^2}}. \quad (16)$$

The mathematical representation of the Sigmoid kernel is as follows:

$$K(x, x_i) = \left((x, x_i) + 1 \right)^a. \quad (17)$$

The mathematical representation of the Radial Basis Function is as follows:

$$K(x, x_i) = S[(x, x_i)]. \quad (18)$$

The variables for the formulae are described below in the Table titled ‘SVM Classification Formula Variables’.

Table SVM Classification Formula Variables

Variable	Description
x	Feature vector consisting of N input variables
x_i	All historical feature vectors of the training set
N	The number of samples used to train the SVM and fit parameters
a_i	A scalar tuning parameter with a value between 0 and C
y_i	Accompanies x_i by indicating the classification into either the positive or negative class.
b	A scalar value obtained in the training process that is used to shift the output of the SVM function by a constant
$K(x, x_i)$	The kernel function of the SVM. The kernel function transforms the non-separable linear data to a separable higher dimensional space and is critical for the functioning of the SVM.

Support Vector Classifier: Parameters

The available parameters for the sci-kit learn Support Vector Classification module are provided in the Table titled ‘Support Vector Classifier Parameters’.

Table Support Vector Classifier Parameters (Pedregosa et al., 2011)

Parameter	Description
C	C is a regularisation parameter that controls the strength of regularisation. Lower values of C correspond to stronger regularisation.
<i>kernel</i>	This option specifies the type of kernel function to use in the algorithm.
<i>degree</i>	The degree of specifically the polynomial kernel function and must be non-negative.
<i>gamma</i>	Kernel coefficient for 'rbf', 'poly' and 'sigmoid'.
<i>Coef</i>	The independent term in a kernel function and is only significant in polynomial and sigmoid kernels.
<i>shrinking</i>	This parameter is used for a shrinking heuristic.
<i>probability</i>	Used to enable probability estimates.
<i>Tol</i>	Specifies the tolerance for the stopping criterion.
<i>cache_size</i>	Specify the size of the kernel cache
<i>class_weight</i>	The parameter C of class i is set to $\text{class_weight}[i]C$. If class_weight is not given, all classes are assigned a weight of one.
<i>random_state</i>	Controls the pseudo random number generation for shuffling the data for probability estimates.
<i>verbose</i>	Enable verbose output.
<i>max_iter</i>	Hard limit on iterations within the solver.
<i>decision_function_shape</i>	Decision function shape: one-vs-rest (ovr) or one-vs-one (ovo).
<i>break_ties</i>	If the decision function shape is set to "ovr" and the number of classes is greater than 2, the predict function will break ties according to the confidence values of the decision function. Otherwise, the predict function will return the first class among the tied classes.

Multilayer Perceptron: Mathematical Representation

The mathematical formula for a single neural network can be shown as follows (Sezer et al., 2020):

$$y_i = \sigma \left(\sum_i w_i x_i + b_i \right). \quad (19)$$

Various non-linear activation functions can be applied to a Deep Multilayer Perceptron as represented mathematically in the Table titled ‘Activation Functions for DMLPs’.

Activation Functions for DMLPs

Activation Function	Formula
Sigmoid	$\sigma(z) = \frac{1}{1 + e^{-z}}$
Hyperbolic Tangent	$\tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$
Rectified Linear Unit (ReLU)	$R(z) = \max(0, z)$
Leaky-ReLU	$R(z) = 1(x < 0)(\alpha x) + 1(x \geq 0)(x)$
Swish	$f(x) = x\sigma(\beta x)$
Softmax (used for classification)	$softmax(z_i) = \frac{expz_i}{\sum_j expz_i}$

Keras Neural Network API: Parameters

A comprehensive overview of the available parameters for the Keras Neural Network API are provided in the Table titled ‘Keras API Parameters’.

Table Keras API Parameters (Chollet, 2015)

Parameter	Description
<i>Layers Parameters</i>	
<i>units</i>	Number of neurons in a layer.
<i>activation</i>	Activation function to use.

Parameter	Description
<i>use_bias</i>	Whether to include a bias term.
<i>kernel_initializer</i>	Initialiser for the kernel weights matrix.
<i>bias_initializer</i>	Initialiser for the bias vector.
<i>kernel_regularizer</i>	Regularizer function applied to the kernel weights matrix.
<i>bias_regularizer</i>	Regularizer function applied to the bias vector.
<i>activity_regularizer</i>	Regularizer function applied to the output of the layer.
Model Compilation Parameters	
<i>optimizer</i>	The optimization algorithm to use.
<i>Loss</i>	Objective function to minimize.
<i>metrics</i>	List of metrics to be evaluated by the model during training and testing such as accuracy.
Training Parameters	
<i>batch_size</i>	Number of samples per gradient update.
<i>epochs</i>	Number of epochs to train the model.
<i>validation_split</i>	Fraction of the training data to be used as validation data.
<i>shuffle</i>	Whether to shuffle the training data before each epoch.
Callback Parameters	
<i>callbacks</i>	List of callback functions to apply during training.
Regularization Parameters	
<i>dropout</i>	Fraction of the input units to drop to prevent overfitting.
<i>kernel_constraint</i>	Constraint function applied to the kernel weights matrix.
<i>bias_constraint</i>	Constraint function applied to the bias vector.
Input and Output Parameters	
<i>input_shape</i>	Shape of the input data.
<i>target_tensors</i>	The model's target tensors.

Logistic Regression: Mathematical Representation

The logistic regression function can be represented mathematically as follows (Hosmer et al., 2013):

$$\log \left(\frac{p(X)}{1 - p(X)} \right) = \beta_0 + \beta_1 X. \quad (20)$$

The variables for the formulae are described in the Table titled ‘Logistic Regression Formula Variables’.

Table Logistic Regression Formula Variables

Variable	Description
$p(X)$	Probability of the target variable equal to a certain value, given that it is in the 'success' category.
$1 - p(X)$	This is the probability of the outcome failing, given that it is not a success.
$\log\left(\frac{p(X)}{1 - p(X)}\right)$	This is the natural logarithm of the odds ratio of success to failure.
β_0	This is the intercept term in the logistic regression equation, which represents the log odds of the outcome when all predictors are zero.
β_1	This is the logistic regression coefficient for the predictor, which quantifies the effect of the predictor on the probability of the outcome occurring.
X	This is the predictor variable, also known as an input feature.

Logistic Regression: Parameters

The available parameters for the sci-kit learn Logistic Regression Classifier module are provided in the Table titled 'Logistic Regression: Parameters'.

Table Logistic Regression Classifier Parameters (Pedregosa et al., 2011)

Parameter	Description
C	C is a regularisation parameter that controls the strength of regularisation. Lower values of C correspond to stronger regularisation.
<i>penalty</i>	This option specifies the type of penalty normalisation used.
<i>dual</i>	Dual (constrained) or primal (regularised) formulation.
<i>tol</i>	Tolerance for stopping criteria.
<i>fit_intercept</i>	Used to add a bias constant to the decision function.
<i>intercept_scaling</i>	Only applicable when using the 'liblinear' solver and setting the 'fit_intercept' parameter to True.

Parameter	Description
<i>class_weight</i>	Option to define the weights associated with each class. If not provided, all classes are considered to have equal weight.
<i>solver</i>	Algorithm to use in the optimization problem and depends on the penalty chosen.
<i>random_state</i>	Controls the pseudo random number generation for shuffling the data for probability estimates.
<i>verbose</i>	Enable verbose output.
<i>max_iter</i>	Hard limit on iterations within the solver.
<i>multi_class</i>	Determines the strategy used to handle multiclass classification problems.
<i>warm_start</i>	Enabling this parameter will preserve the solution obtained from the preceding fitting operation and employ it as the basis for the current fitting process.
<i>n_jobs</i>	Number of processor cores utilized for parallel processing across classes when the <i>multi_class</i> parameter is set to 'ovr'.
<i>l1_ratio</i>	The Elastic-Net regularisation parameter.

Appendix B

B1.Chapter 3: Shares Listed on the FTSE/JSE ALSI

Table B1: Shares Listed on the FTSE/JSE ALSI

Identifier	Full Name	Sector	Sub-Sector
KRO-ZA	Karoo0000	10 Technology	101010 Software and Computer Services
BYI-ZA	Bytes Technology Group	10 Technology	101010 Software and Computer Services
PRX-ZA	Prosus	10 Technology	101010 Software and Computer Services
AEL-ZA	Altron Limited A	10 Technology	101010 Software and Computer Services
NPN-ZA	Naspers	10 Technology	101010 Software and Computer Services
DTC-ZA	Datatec	10 Technology	101010 Software and Computer Services
MCG-ZA	MultiChoice Group	15 Telecommunications	151020 Telecommunications Service Providers
BLU-ZA	Blue Label Telecoms Ltd.	15 Telecommunications	151020 Telecommunications Service Providers
TKG-ZA	Telkom SA SOC	15 Telecommunications	151020 Telecommunications Service Providers
MTN-ZA	MTN Group	15 Telecommunications	151020 Telecommunications Service Providers
VOD-ZA	Vodacom Group	15 Telecommunications	151020 Telecommunications Service Providers
MEI-ZA	Mediclinic International plc	20 Health Care	201010 Health Care Providers
LHC-ZA	Life Healthcare Group Holdings	20 Health Care	201010 Health Care Providers
NTC-ZA	Netcare	20 Health Care	201010 Health Care Providers
AIP-ZA	Adcock Ingram Holdings	20 Health Care	201030 Pharmaceuticals and Biotechnology
APN-ZA	Aspen Pharmacare Holdings	20 Health Care	201030 Pharmaceuticals and Biotechnology

Identifier	Full Name	Sector	Sub-Sector
INP-ZA	Investec PLC	30 Financials	301010 Banks
SBK-ZA	Standard Bank Group	30 Financials	301010 Banks
CPI-ZA	Capitec Bank Hldgs Ltd	30 Financials	301010 Banks
ABG-ZA	Absa Group Limited	30 Financials	301010 Banks
INL-ZA	Investec Ltd	30 Financials	301010 Banks
FSR-ZA	Firststrand Limited	30 Financials	301010 Banks
NED-ZA	Nedbank Group	30 Financials	301010 Banks
TCP-ZA	Transaction Capital	30 Financials	302010 Finance and Credit Services
BAT-ZA	Brait PLC	30 Financials	302020 Investment Banking and Brokerage Services
PSG-ZA	PSG Group	30 Financials	302020 Investment Banking and Brokerage Services
HCI-ZA	Hosken Cons Invest	30 Financials	302020 Investment Banking and Brokerage Services
RMI-ZA	Rand Merchant Investment Holdings Ltd	30 Financials	302020 Investment Banking and Brokerage Services
AFH-ZA	Alexander Forbes Equity Holdings Proprietary	30 Financials	302020 Investment Banking and Brokerage Services
JSE-ZA	JSE	30 Financials	302020 Investment Banking and Brokerage Services
REM-ZA	Remgro	30 Financials	302020 Investment Banking and Brokerage Services
RNI-ZA	Reinet Investments	30 Financials	302020 Investment Banking and Brokerage Services
CML-ZA	Coronation Fund Managers	30 Financials	302020 Investment Banking and Brokerage Services
KST-ZA	PSG Konsult Ltd	30 Financials	302020 Investment Banking and Brokerage Services
QLT-ZA	Quilter	30 Financials	302020 Investment Banking and Brokerage Services
NY1-ZA	Ninety One Ltd	30 Financials	302020 Investment Banking and Brokerage Services
N91-ZA	Ninety One Plc	30 Financials	302020 Investment Banking and Brokerage Services

Identifier	Full Name	Sector	Sub-Sector
ZED-ZA	Zeder Investments	30 Financials	302020 Investment Banking and Brokerage Services
AIL-ZA	African Rainbow Capital Investments Ltd	30 Financials	302040 Closed End Investments
SLM-ZA	Sanlam	30 Financials	303010 Life Insurance
LBH-ZA	Liberty Holdings	30 Financials	303010 Life Insurance
OMU-ZA	Old Mutual Ltd	30 Financials	303010 Life Insurance
DSY-ZA	Discovery Ltd	30 Financials	303010 Life Insurance
MTM-ZA	Momentum Metropolitan Holdings	30 Financials	303010 Life Insurance
SNT-ZA	Santam	30 Financials	303020 Non-life Insurance
NRP-ZA	NEPI Rockcastle PLC	35 Real Estate	351010 Real Estate Investment and Services
SRE-ZA	Sirius Real Estate Ltd	35 Real Estate	351010 Real Estate Investment and Services
MSP-ZA	MAS PLC	35 Real Estate	351010 Real Estate Investment and Services
LTE-ZA	Lighthouse Capital Ltd	35 Real Estate	351010 Real Estate Investment and Services
VKE-ZA	Vukile Property Fund	35 Real Estate	351020 Real Estate Investment Trusts
MLI-ZA	Industrials REIT Limited	35 Real Estate	351020 Real Estate Investment Trusts
IPF-ZA	Investec Property Fund Ltd	35 Real Estate	351020 Real Estate Investment Trusts
IAP-ZA	Irongate Group	35 Real Estate	351020 Real Estate Investment Trusts
SAC-ZA	SA Corporate Real Estate Fund	35 Real Estate	351020 Real Estate Investment Trusts
GRT-ZA	Growthpoint Prop Ltd	35 Real Estate	351020 Real Estate Investment Trusts
ATT-ZA	Attacq Limited	35 Real Estate	351020 Real Estate Investment Trusts
CCO-ZA	Capital & Counties Properties PLC	35 Real Estate	351020 Real Estate Investment Trusts

Identifier	Full Name	Sector	Sub-Sector
EQU-ZA	Equites Property Fund	35 Real Estate	351020 Real Estate Investment Trusts
EMI-ZA	Emira Property Fund	35 Real Estate	351020 Real Estate Investment Trusts
HYP-ZA	Hyprop Investments Ltd	35 Real Estate	351020 Real Estate Investment Trusts
RES-ZA	Resilient REIT Ltd	35 Real Estate	351020 Real Estate Investment Trusts
RDF-ZA	Redefine Properties	35 Real Estate	351020 Real Estate Investment Trusts
FFB-ZA	Fortress REIT Ltd (B)	35 Real Estate	351020 Real Estate Investment Trusts
FFA-ZA	Fortress REIT Ltd (A)	35 Real Estate	351020 Real Estate Investment Trusts
AHB-ZA	Arrowhead Properties B	35 Real Estate	351020 Real Estate Investment Trusts
L2D-ZA	Liberty Two Degrees	35 Real Estate	351020 Real Estate Investment Trusts
HMN-ZA	Hammerson Plc	35 Real Estate	351020 Real Estate Investment Trusts
SSS-ZA	Stor-Age Property REIT	35 Real Estate	351020 Real Estate Investment Trusts
EPP-ZA	EPP N.V.	35 Real Estate	351020 Real Estate Investment Trusts
MTA-ZA	Metair Investments Ord	40 Consumer Discretionary	401010 Automobiles and Parts
ADH-ZA	Advtech	40 Consumer Discretionary	402010 Consumer Services
COH-ZA	Curro Holdings	40 Consumer Discretionary	402010 Consumer Services
L4L-ZA	Long4life	40 Consumer Discretionary	402030 Leisure Goods
CFR-ZA	Compagnie Financiere Richemont AG	40 Consumer Discretionary	402040 Personal Goods

Identifier	Full Name	Sector	Sub-Sector
WHL-ZA	Woolworths Holdings	40 Consumer Discretionary	404010 Retailers
MTH-ZA	Motus Holdings	40 Consumer Discretionary	404010 Retailers
TRU-ZA	Truworths International	40 Consumer Discretionary	404010 Retailers
SNH-ZA	Steinhoff International Holdings N.V.	40 Consumer Discretionary	404010 Retailers
PPH-ZA	Pepkor Holdings Ltd	40 Consumer Discretionary	404010 Retailers
CSB-ZA	Cashbuild Ltd	40 Consumer Discretionary	404010 Retailers
TFG-ZA	The Foschini Group Ltd	40 Consumer Discretionary	404010 Retailers
MSM-ZA	Massmart Holdings	40 Consumer Discretionary	404010 Retailers
MRP-ZA	Mr Price Group	40 Consumer Discretionary	404010 Retailers
ITE-ZA	Italtile Ltd	40 Consumer Discretionary	404010 Retailers
CLH-ZA	City Lodge Hotels	40 Consumer Discretionary	405010 Travel and Leisure
SUI-ZA	Sun International Ltd	40 Consumer Discretionary	405010 Travel and Leisure
TSG-ZA	Tsogo Sun Gaming	40 Consumer Discretionary	405010 Travel and Leisure
FBR-ZA	Famous Brands	40 Consumer Discretionary	405010 Travel and Leisure
TGO-ZA	Tsogo Sun Hotels	40 Consumer Discretionary	405010 Travel and Leisure
ANH-ZA	Anheuser-Busch InBev	45 Consumer Staples	451010 Beverages
DGH-ZA	Distell Group Holdings	45 Consumer Staples	451010 Beverages
LBR-ZA	Libstar Holdings	45 Consumer Staples	451020 Food Producers
AVI-ZA	AVI	45 Consumer Staples	451020 Food Producers
ARL-ZA	Astral Foods Ltd	45 Consumer Staples	451020 Food Producers

Identifier	Full Name	Sector	Sub-Sector
OCE-ZA	Oceana Group	45 Consumer Staples	451020 Food Producers
RFG-ZA	RFG Holdings	45 Consumer Staples	451020 Food Producers
TBS-ZA	Tiger Brands	45 Consumer Staples	451020 Food Producers
BTI-ZA	British American Tobacco PLC	45 Consumer Staples	451030 Tobacco
SPP-ZA	The Spar Group	45 Consumer Staples	452010 Personal Care, Drug and Grocery Stores
PIK-ZA	Pick N Pay Stores	45 Consumer Staples	452010 Personal Care, Drug and Grocery Stores
CLS-ZA	Clicks Group Ltd	45 Consumer Staples	452010 Personal Care, Drug and Grocery Stores
DCP-ZA	Dis-Chem Pharmacies	45 Consumer Staples	452010 Personal Care, Drug and Grocery Stores
SHP-ZA	Shoprite	45 Consumer Staples	452010 Personal Care, Drug and Grocery Stores
BID-ZA	BidCorp Ltd	45 Consumer Staples	452010 Personal Care, Drug and Grocery Stores
WBO-ZA	Wilson Bayly Holmes-Ovcon	50 Industrials	501010 Construction and Materials
PPC-ZA	PPC Ltd	50 Industrials	501010 Construction and Materials
AFT-ZA	Afrimat	50 Industrials	501010 Construction and Materials
RBX-ZA	Raubex Group	50 Industrials	501010 Construction and Materials
MUR-ZA	Murray & Roberts	50 Industrials	501010 Construction and Materials
RLO-ZA	Reunert	50 Industrials	502020 Electronic and Electrical Equipment
KAP-ZA	KAP Industrial Holdings Ltd	50 Industrials	502030 General Industrials
BAW-ZA	Barloworld	50 Industrials	502030 General Industrials
BVT-ZA	Bidvest Group	50 Industrials	502030 General Industrials
MNP-ZA	Mondi Plc	50 Industrials	502030 General Industrials

Identifier	Full Name	Sector	Sub-Sector
HDC-ZA	Hudaco Industries	50 Industrials	502050 Industrial Support Services
TXT-ZA	Textainer Group Holdings	50 Industrials	502060 Industrial Transportation
IPL-ZA	Imperial Logistics Ltd	50 Industrials	502060 Industrial Transportation
SPG-ZA	Super Group	50 Industrials	502060 Industrial Transportation
GND-ZA	Grindrod	50 Industrials	502060 Industrial Transportation
SAP-ZA	Sappi	55 Basic Materials	551010 Industrial Materials
THA-ZA	Tharisa PLC	55 Basic Materials	551020 Industrial Metals and Mining
BHP-ZA	BHP Group Plc	55 Basic Materials	551020 Industrial Metals and Mining
KIO-ZA	Kumba Iron Ore	55 Basic Materials	551020 Industrial Metals and Mining
ARI-ZA	African Rainbow Minerals Ltd	55 Basic Materials	551020 Industrial Metals and Mining
GLN-ZA	Glencore	55 Basic Materials	551020 Industrial Metals and Mining
AGL-ZA	Anglo American	55 Basic Materials	551020 Industrial Metals and Mining
ACL-ZA	ArcelorMittal South Africa Ltd	55 Basic Materials	551020 Industrial Metals and Mining
SSW-ZA	Sibanye Stillwater	55 Basic Materials	551030 Precious Metals and Mining
RBP-ZA	Royal Bafokeng Platinum	55 Basic Materials	551030 Precious Metals and Mining
ANG-ZA	Anglogold Ashanti	55 Basic Materials	551030 Precious Metals and Mining
DRD-ZA	DRD Gold	55 Basic Materials	551030 Precious Metals and Mining
GFI-ZA	Gold Fields	55 Basic Materials	551030 Precious Metals and Mining

Identifier	Full Name	Sector	Sub-Sector
NPH-ZA	Northam Platinum Holdings	55 Basic Materials	551030 Precious Metals and Mining
HAR-ZA	Harmony	55 Basic Materials	551030 Precious Metals and Mining
IMP-ZA	Impala Platinum Holdings Ltd	55 Basic Materials	551030 Precious Metals and Mining
PAN-ZA	Pan African Resources	55 Basic Materials	551030 Precious Metals and Mining
AMS-ZA	Anglo American Platinum	55 Basic Materials	551030 Precious Metals and Mining
OMN-ZA	Omnia Holdings Ltd	55 Basic Materials	552010 Chemicals
AFE-ZA	AECI	55 Basic Materials	552010 Chemicals
SOL-ZA	Sasol	55 Basic Materials	552010 Chemicals
EXX-ZA	Exxaro Resources	60 Energy	601010 Oil, Gas and Coal
TGA-ZA	Thungela Resources	60 Energy	601010 Oil, Gas and Coal
MKR-ZA	Montauk Renewables Inc	60 Energy	601020 Alternative Energy

B2.Chapter 3: Initial Data Heatmap

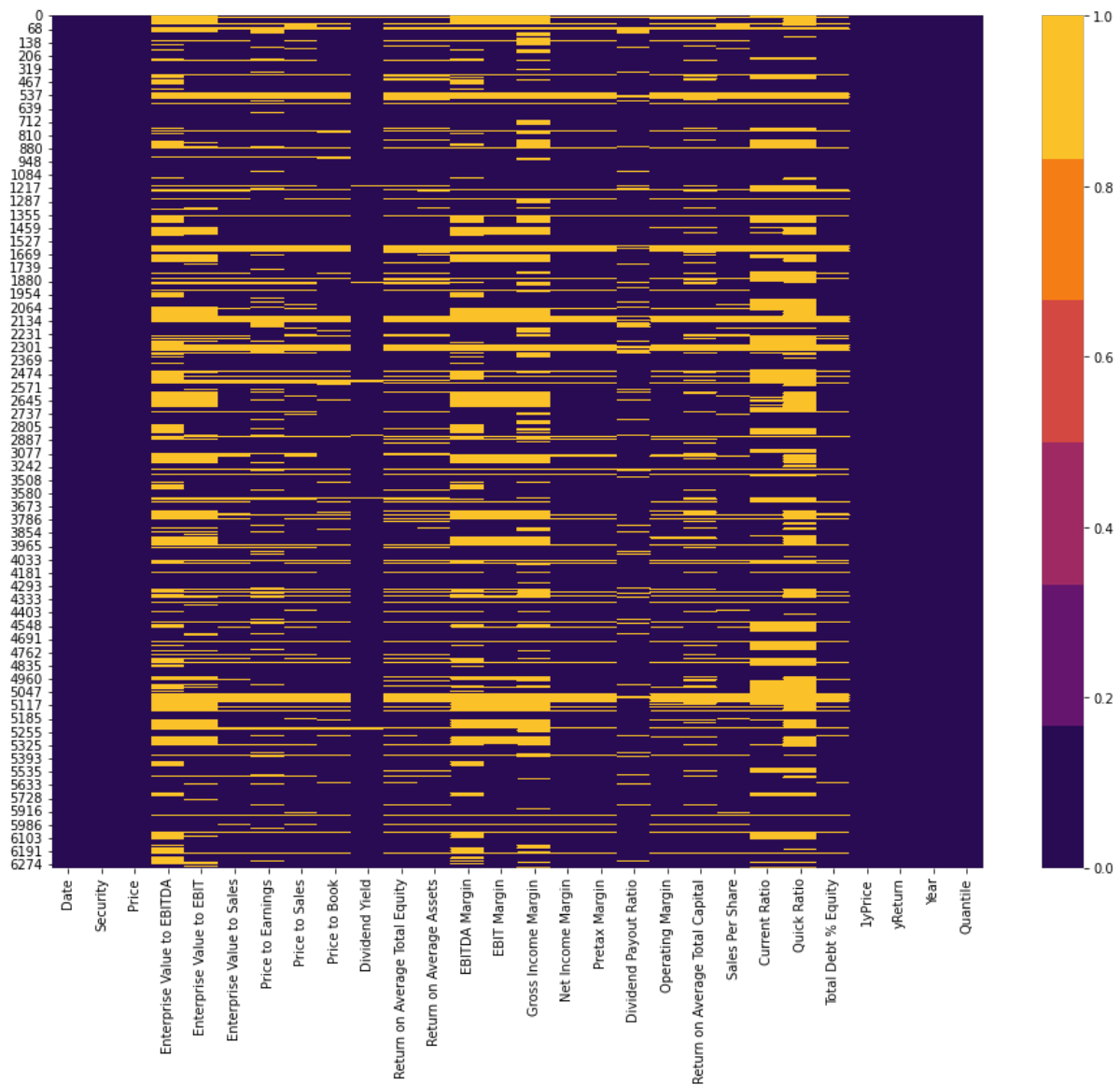


Figure B2. Initial Data Heatmap

B3.Chapter 3: Intermediate Data Heatmap Excluding Rows with 20% Nulls

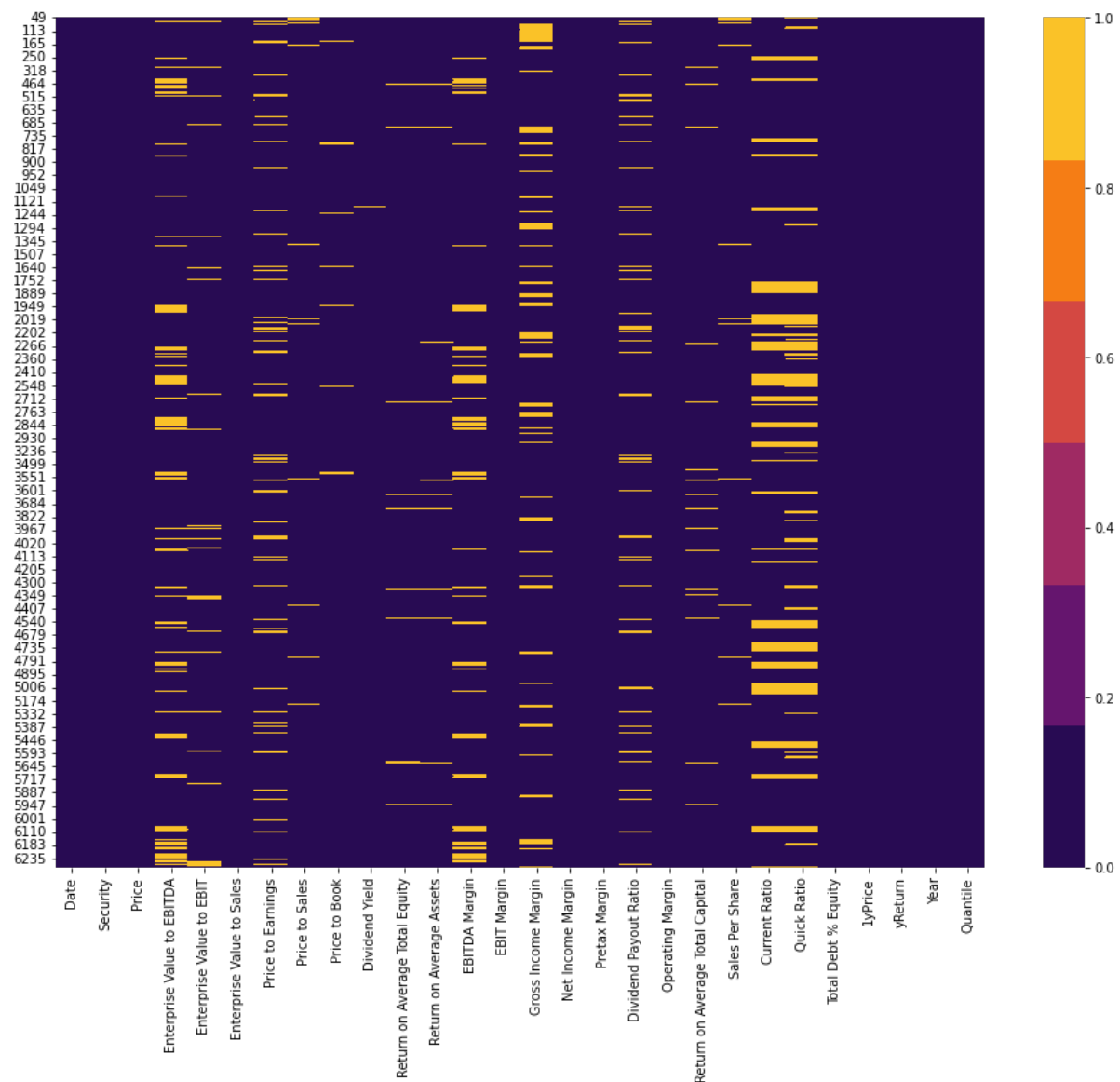


Figure B3: Intermediate Data Heatmap Excluding Rows with 20% Nulls

B4.Chapter 3: Intermediate Data Heatmap Excluding Columns with 10% Nulls

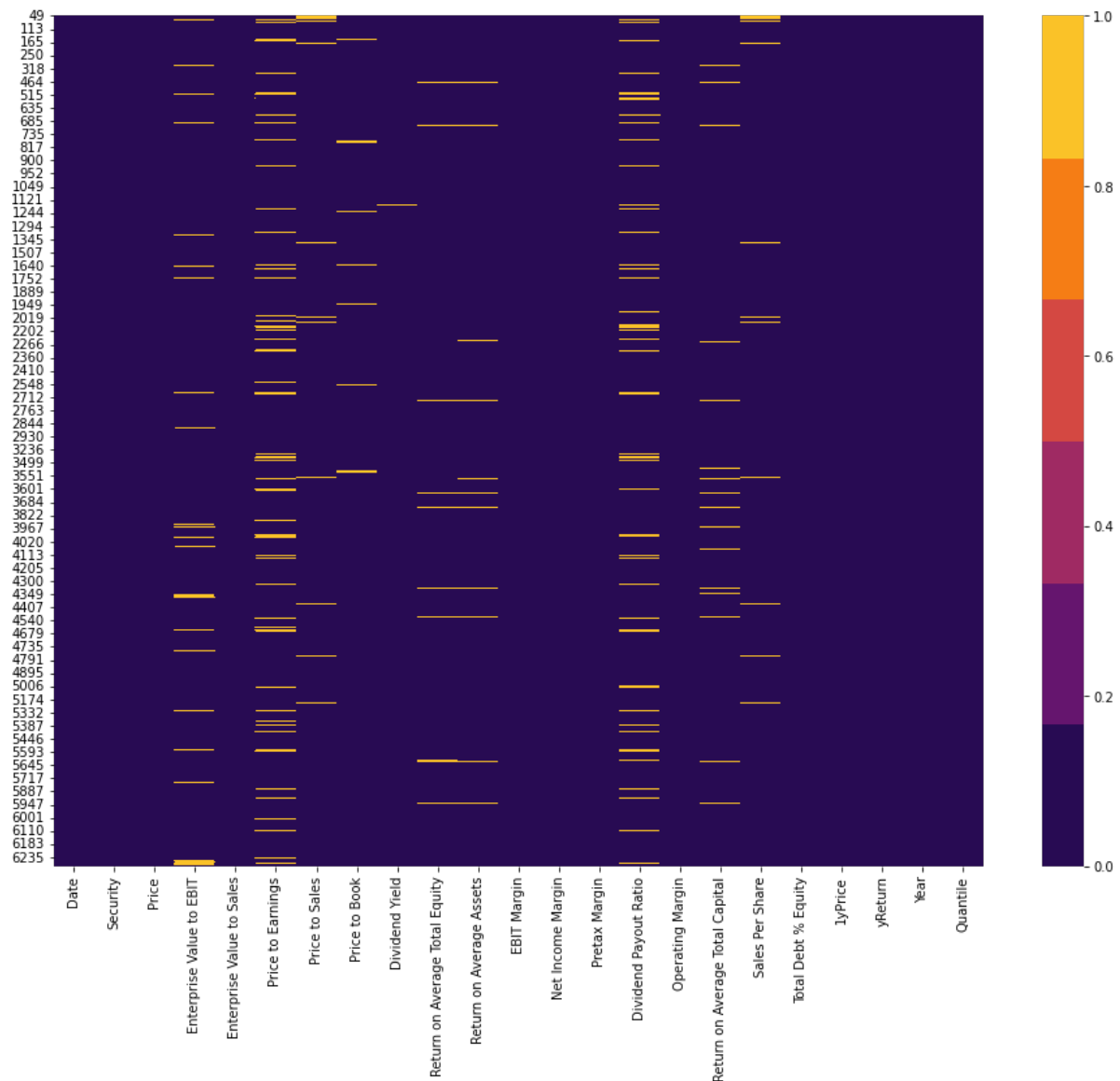


Figure B4: Intermediate Data Heatmap Excluding Columns with 10% Nulls

B5.Chapter 3: Final Data Heatmap

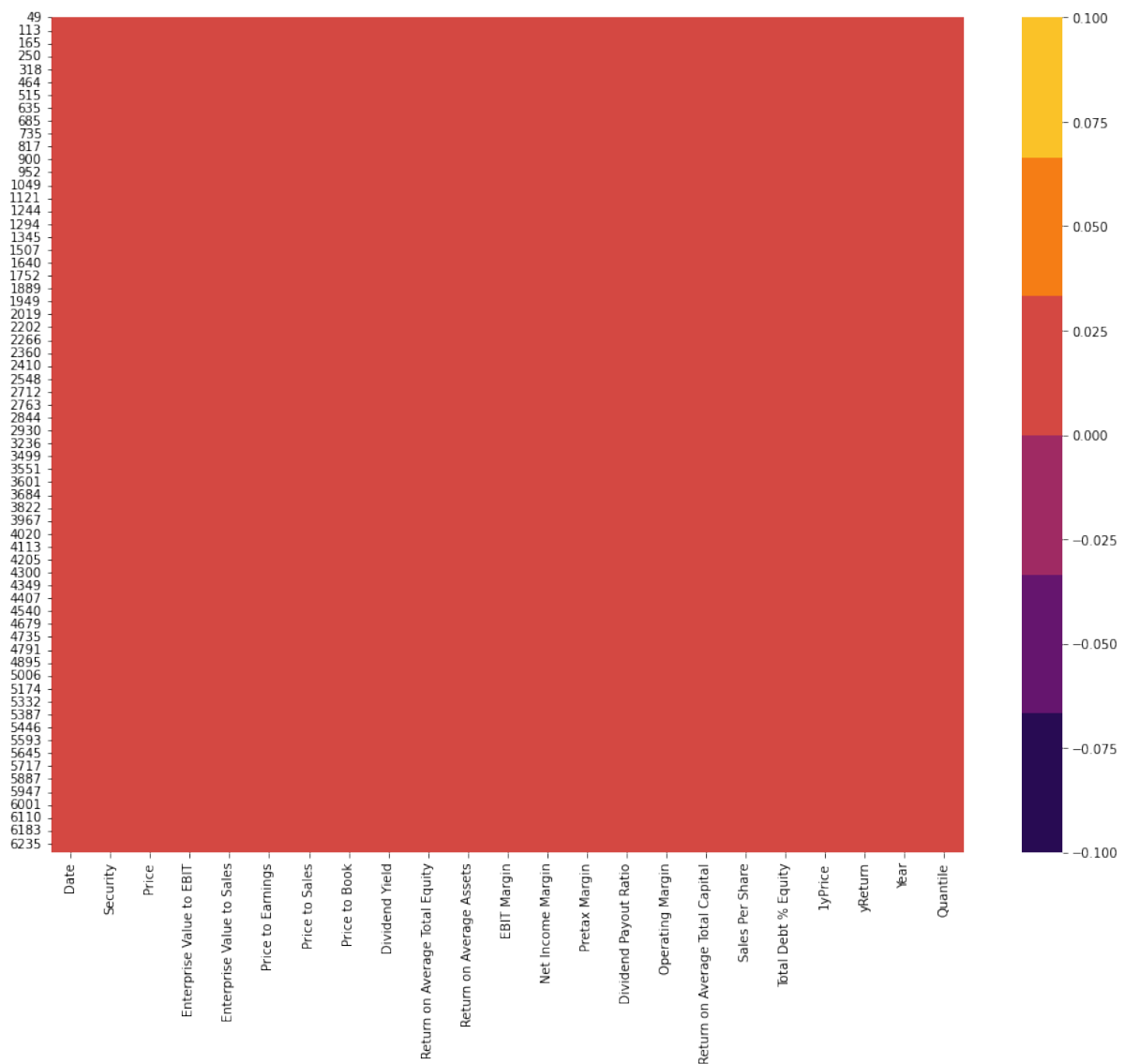
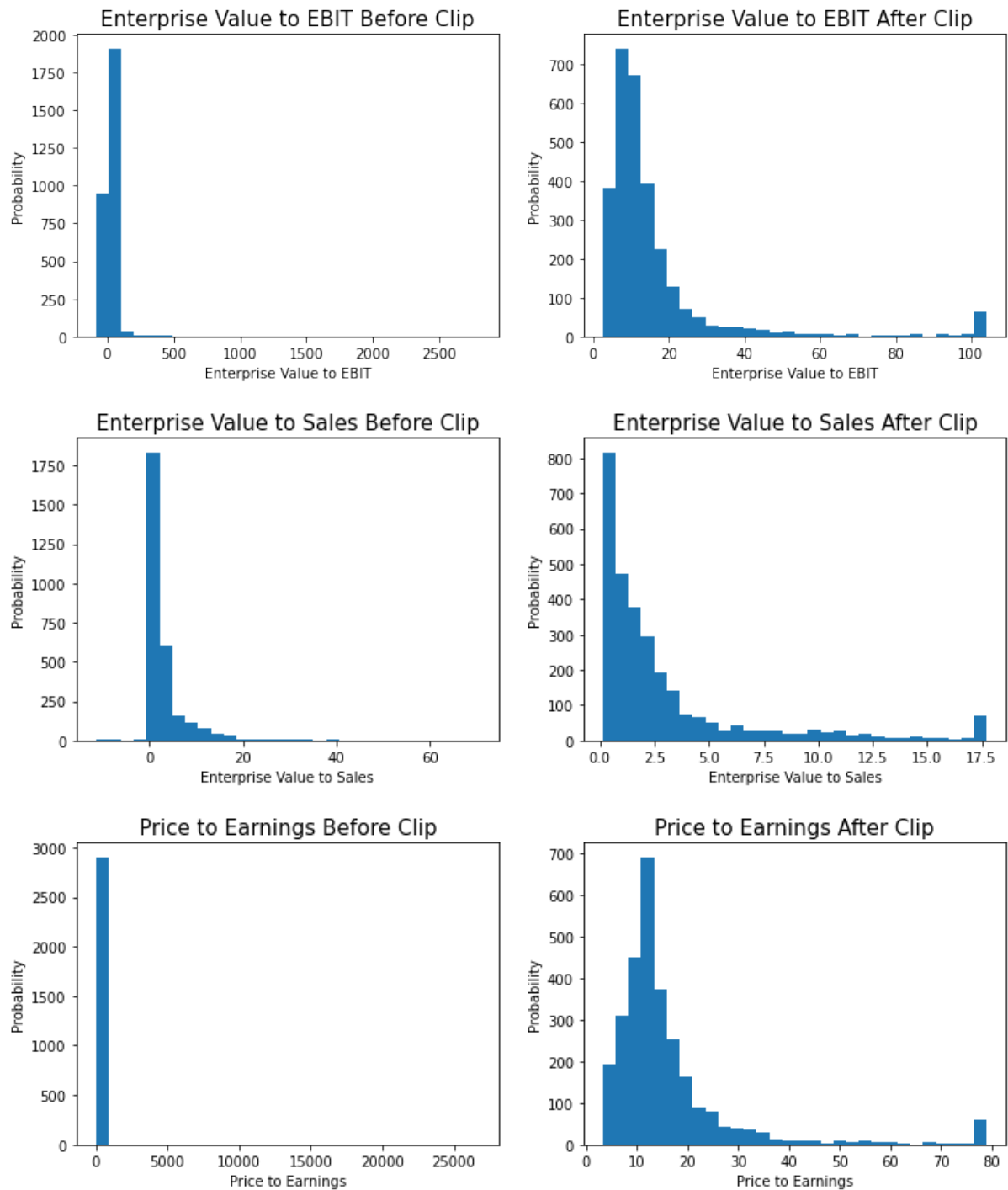
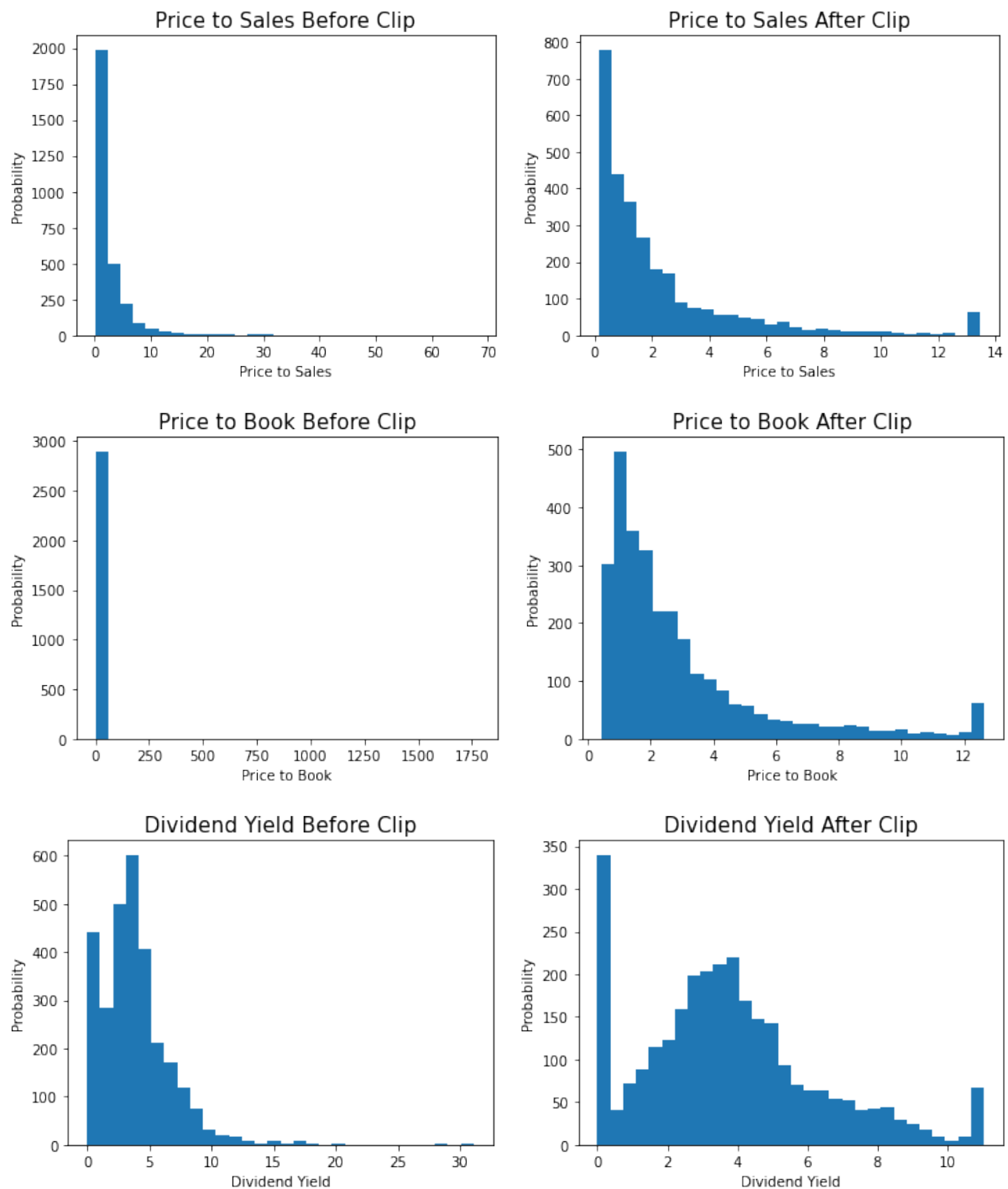
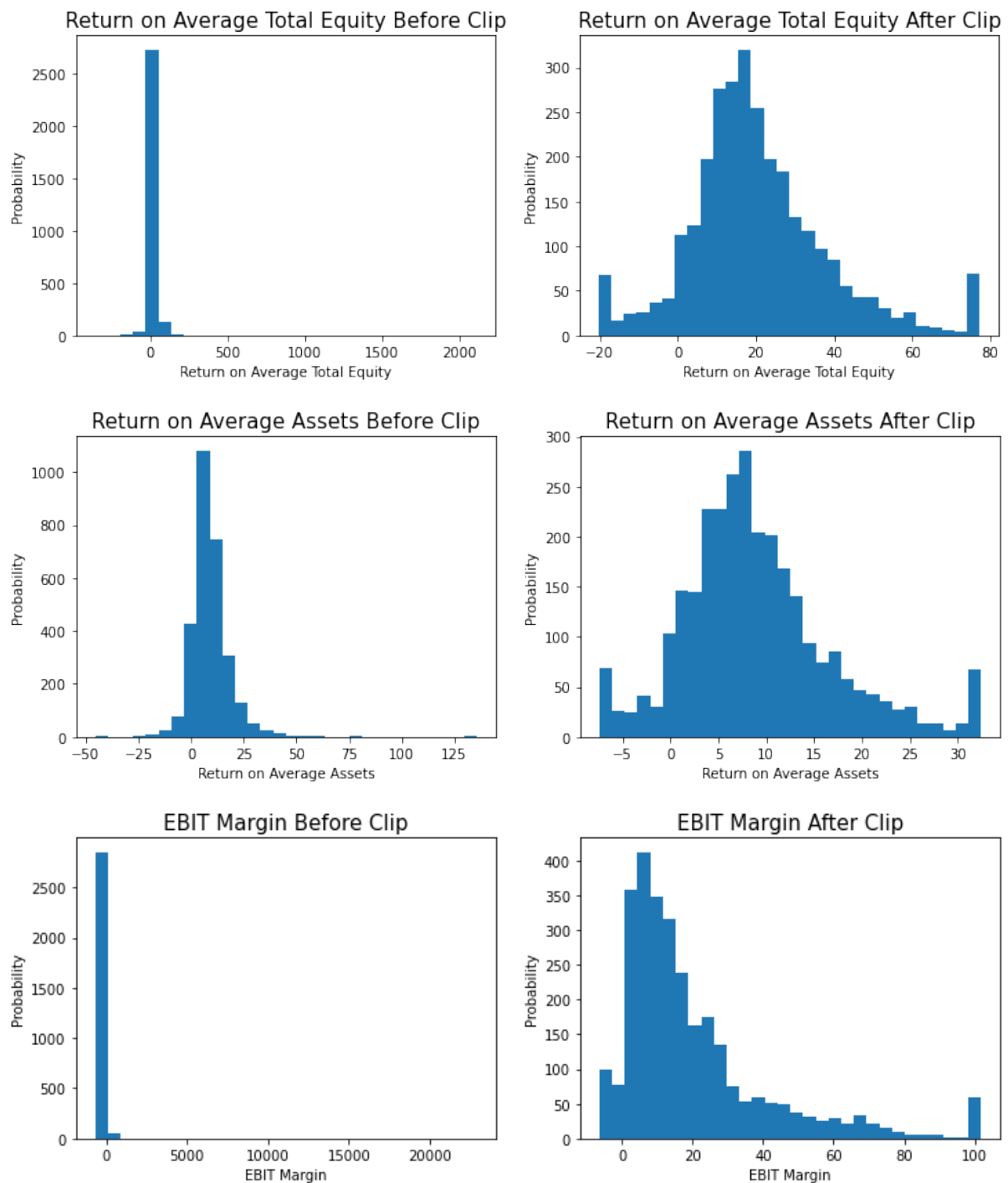


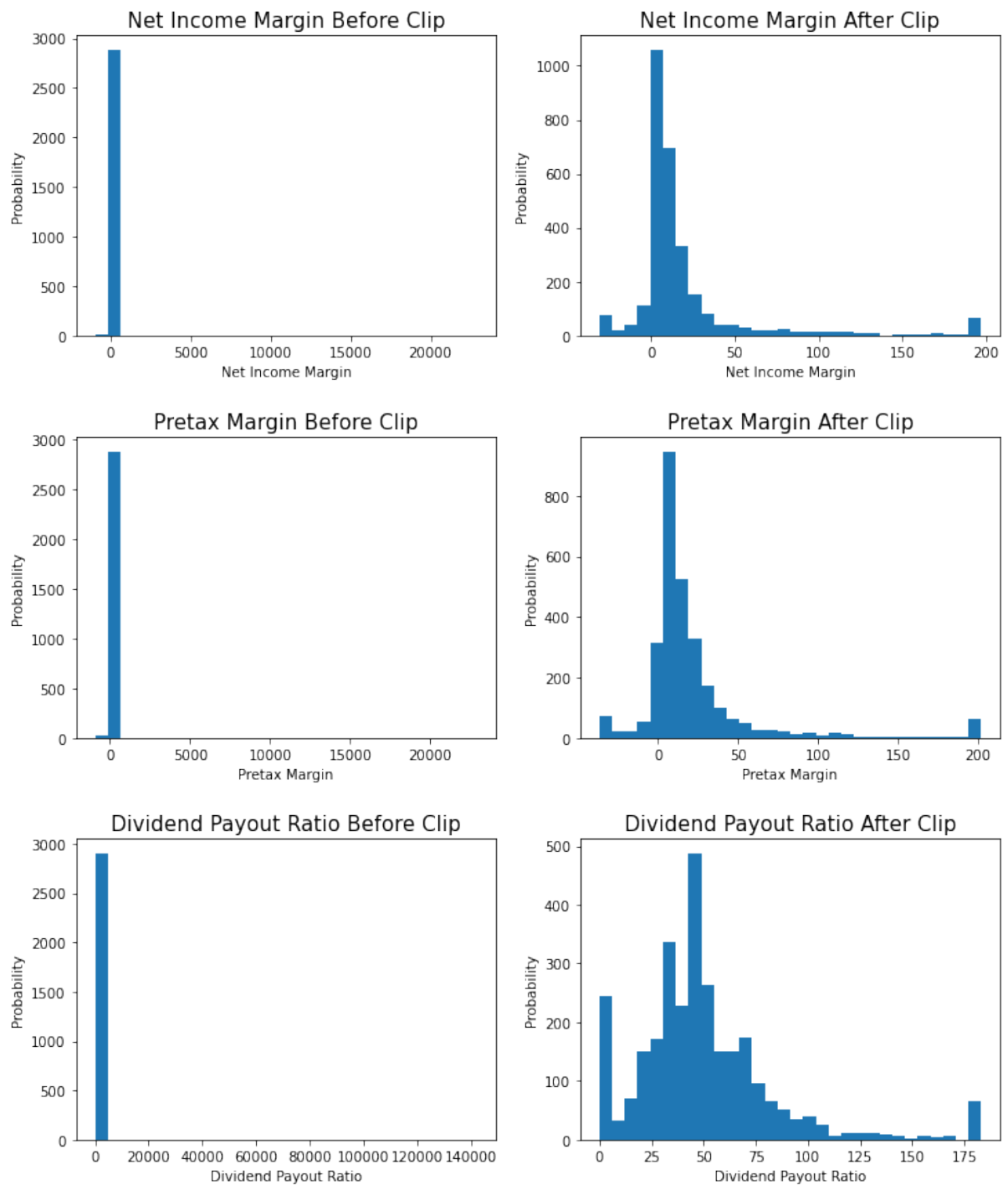
Figure B5: Final Data Heatmap

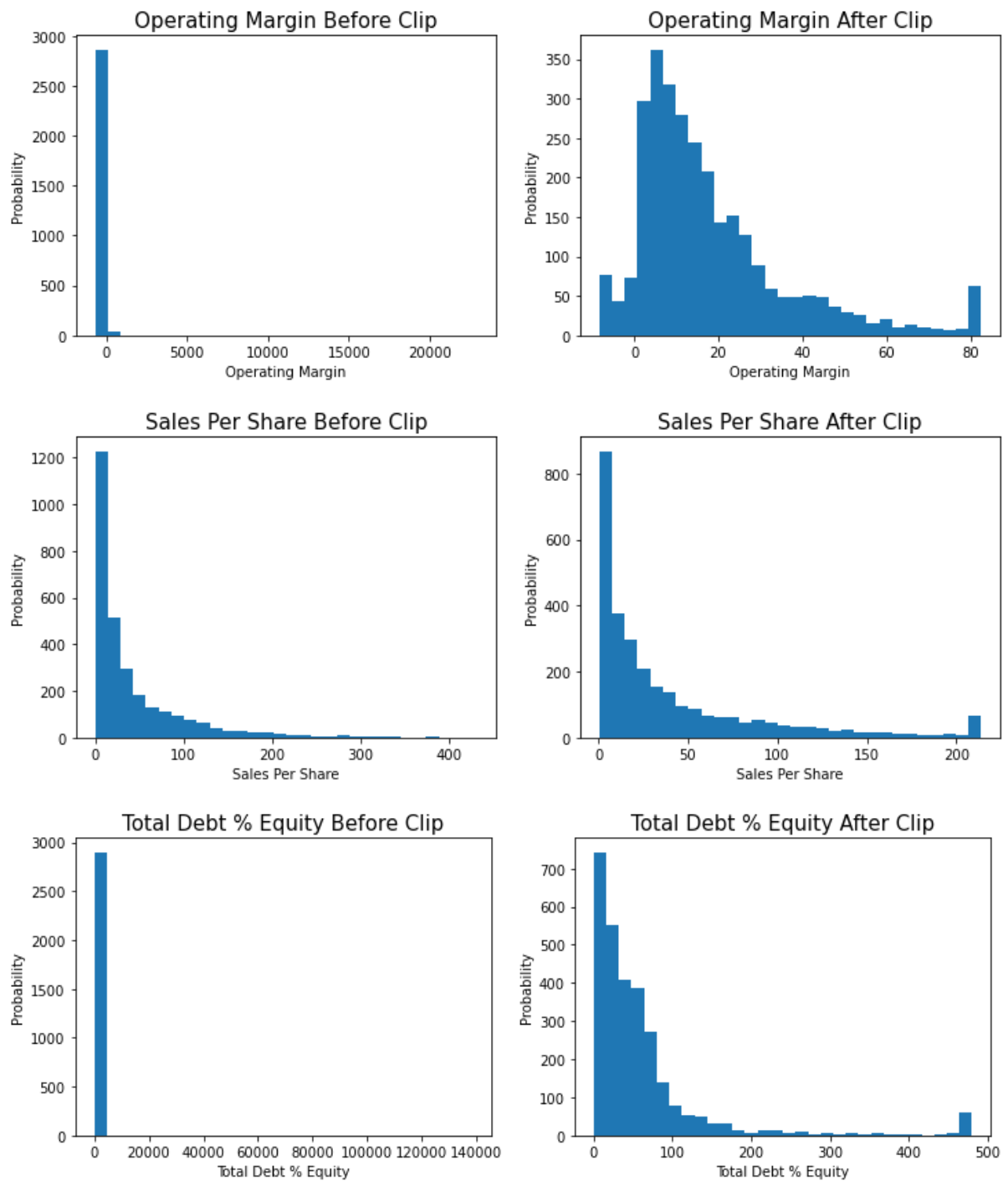
B6.Chapter 3: Features Before and After adjusting for Outliers using the clipRatios Function











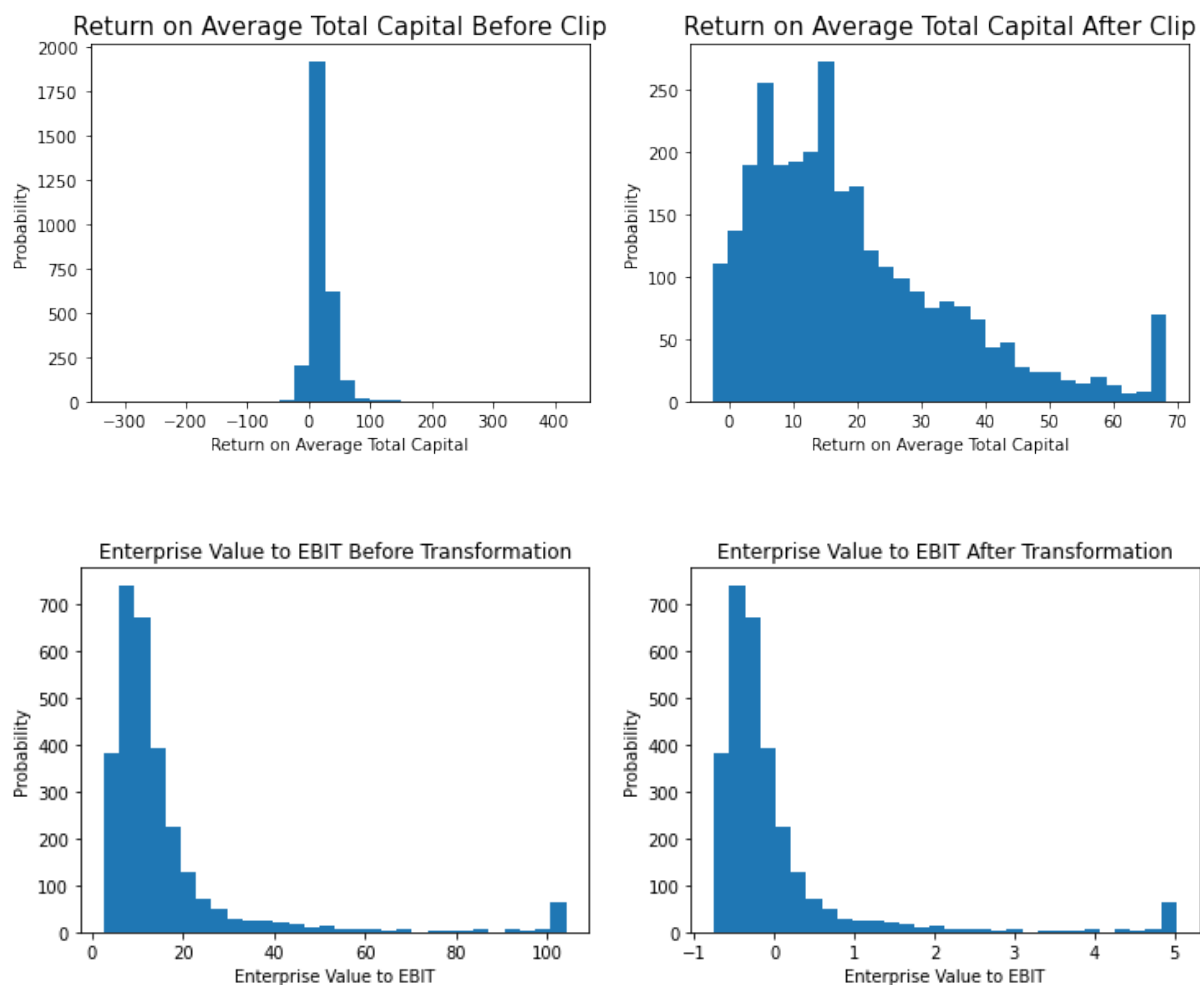
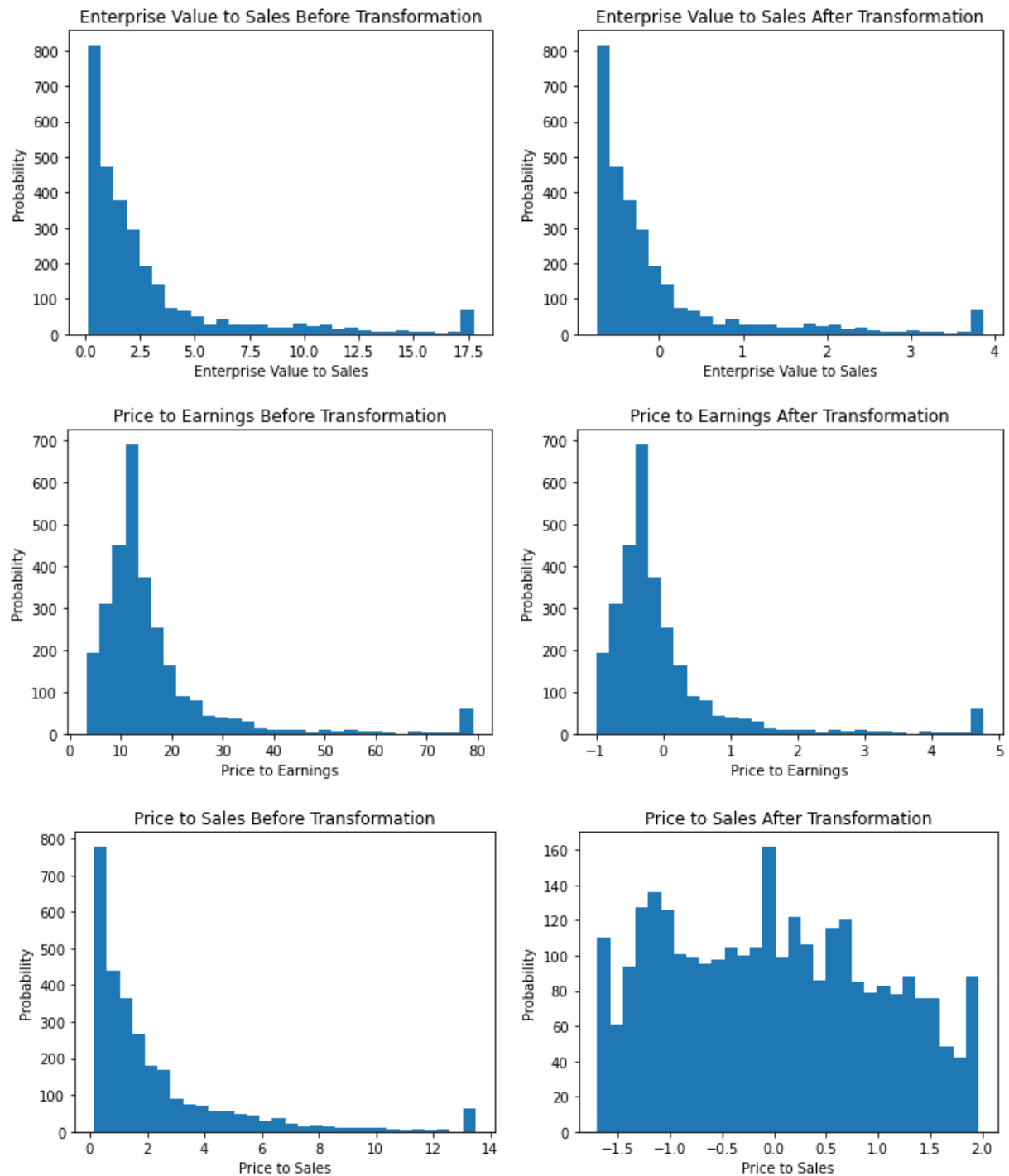
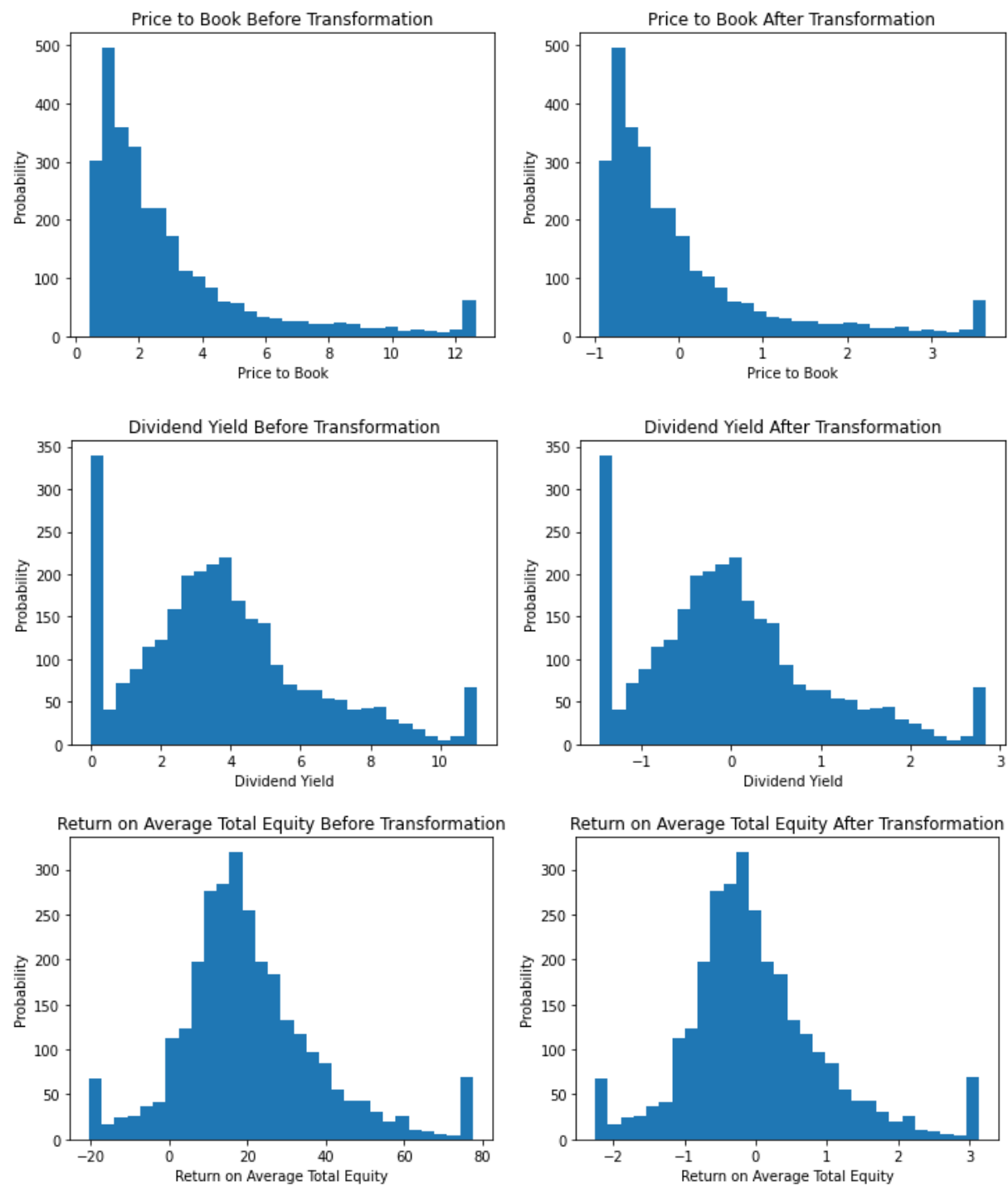
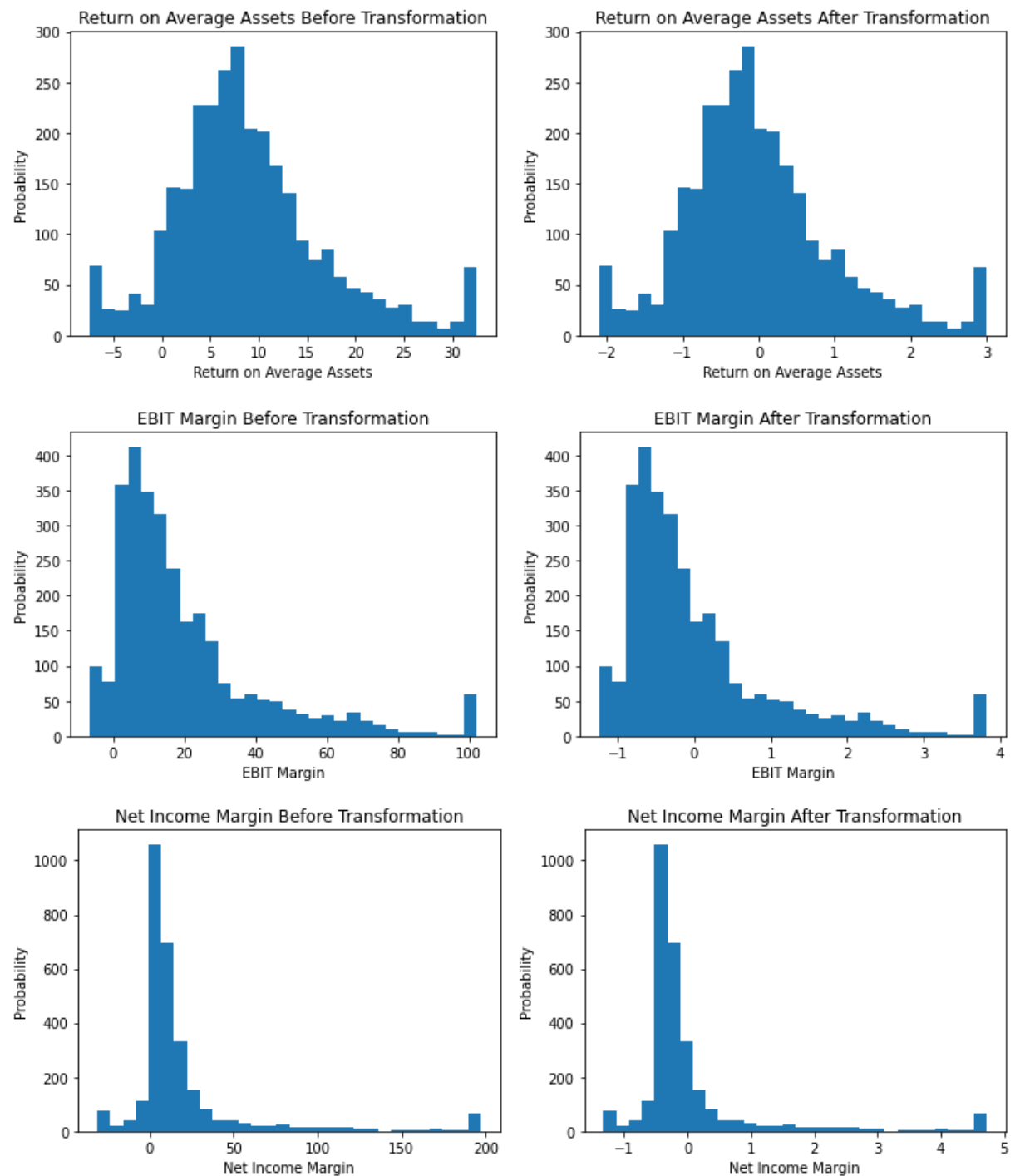


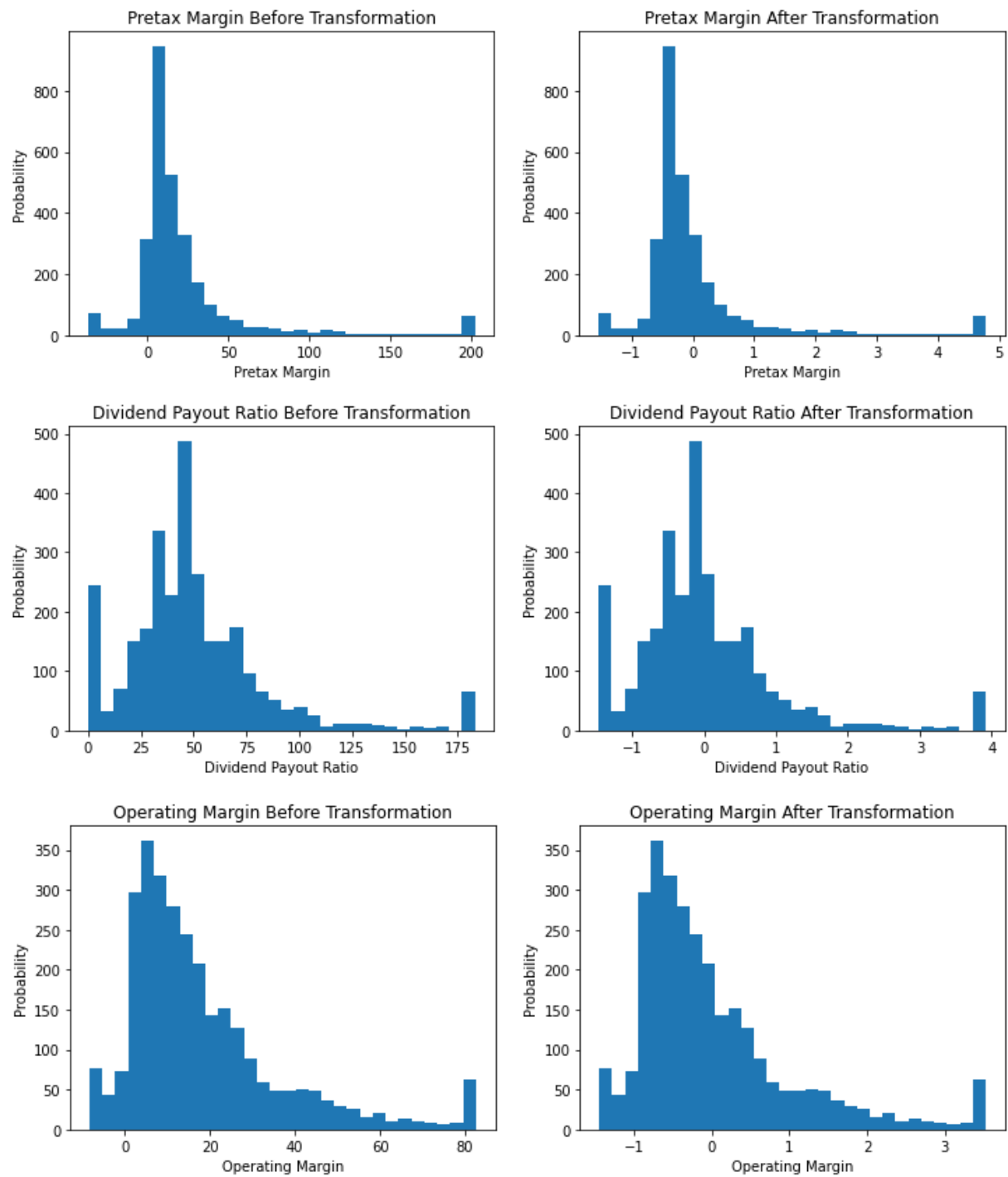
Figure B6 – Features Before and After adjusting for Outliers using the `clipRatios` Function

B7.Chapter 3: Features Before and After Transformations adjusting for Standard Scaling









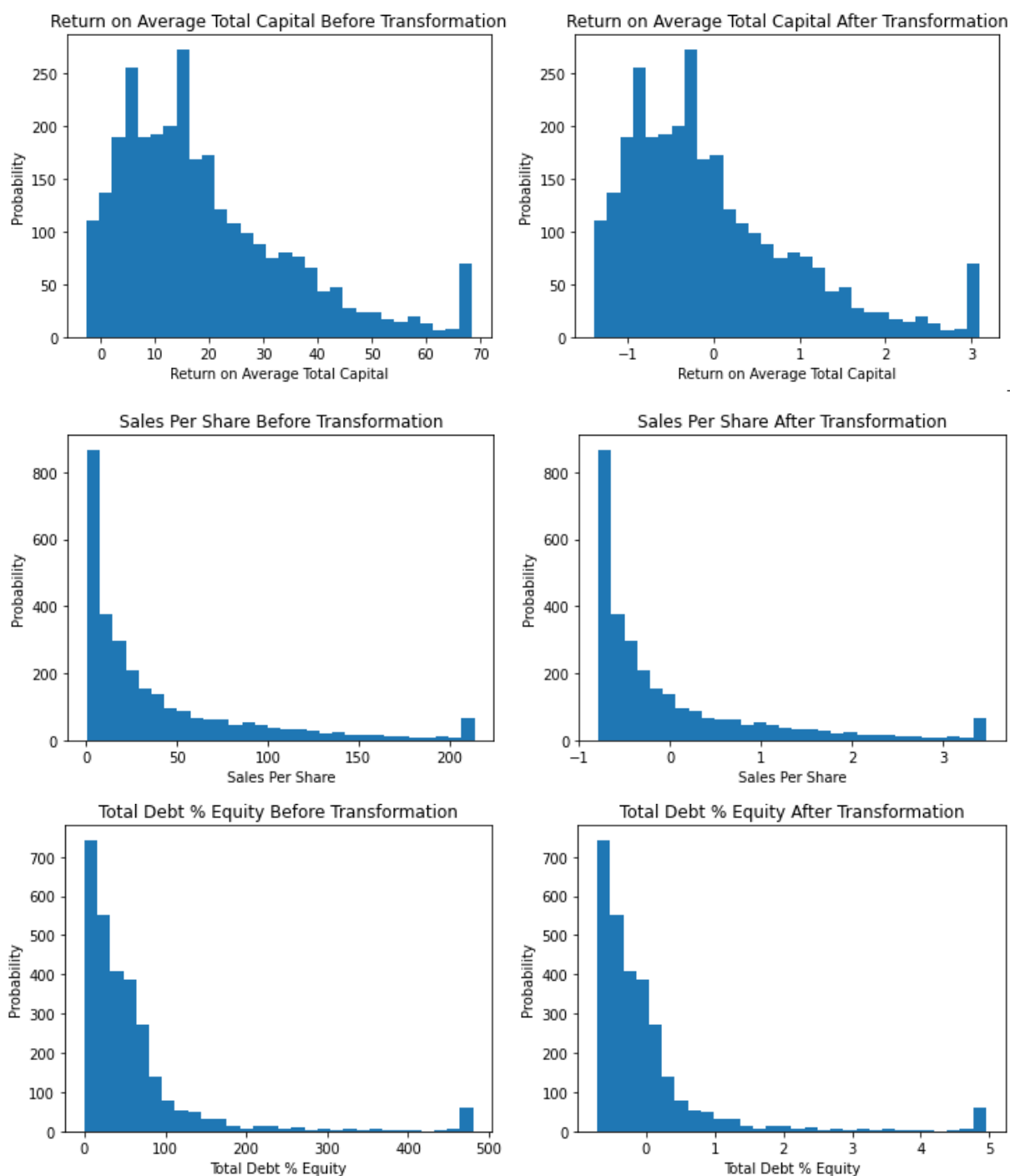


Figure B7 – Features Before and After Transformations adjusting for Standard Scaling