INTERNATIONAL HELLENIC UNIVERSITY

DEPARTMENT OF COMPUTER, INFORMATICS AND TELECOMMUNICATIONS ENGINEERING

MASTER IN APPLIED INFORMATICS

Machine And Deep Learning Techniques for Stock Price Prediction

Praxitelis-Nikolaos Kouroupetroglou

UID: 230

Supervisor: Prof. Alkiviadis Tsimpiris

Submission Date: June 27th 2024

Disclaimer

I confirm that this thesis type is my own work and I have documented all sources and material used.

Serres, Greece 27th of June 2024 Praxitelis-Nikolaos Kouroupetroglou

Acknowledgments

First and foremost, I would like to express my deepest gratitude to my supervisor, Prof. Mr. Alkiviadis Tsimpiris, for his invaluable guidance, support, and encouragement throughout the course of this research. His insights and expertise have been instrumental in shaping this thesis.

I am also profoundly grateful to Prof. Mr. Dimitrios Varsamis, whose comprehensive Python courses provided me with the essential programming skills needed for this project. His patience and dedication in teaching have greatly enhanced my understanding and proficiency in Python.

Additionally, I extend my heartfelt thanks to Prof. Mr. Charalampos Strouthopoulos for his enlightening lessons in machine learning. His thorough instruction and willingness to share his extensive knowledge have been crucial in developing the machine learning models explored in this thesis.

Lastly, I would like to thank my family and friends for their unwavering support and encouragement throughout this journey. This accomplishment would not have been possible without their understanding and encouragement.

Thank you all.

Abstract

Artificial Intelligence (AI) has emerged as a transformative force and has the ability to accurately predict short-term stock prices which is a highly sought-after goal in the field of financial markets. This thesis explores the application of advanced machine learning methods to achieve precise short-term stock price forecasts. The study encompasses a detailed examination of various machine learning techniques, including regression models, classification models, neural networks, and ensemble methods, with a particular focus on time series analysis and deep learning architectures like Long Short-Term Memory (LSTM) networks.

A comprehensive dataset, comprising historical stock prices from the KRIKRI milk industry, and technical indicators, is employed to train and test the predictive models. The performance of these models is evaluated using metrics such as Mean Absolute Percent Error (MAPE) and root Mean Squared Error (RMSE).

Additionally, this thesis introduces a financial stock dashboard designed to leverage these machine learning models, providing real-time predictive insights and aiding investors in making informed decisions.

The findings reveal that Random Forest models and ARIMA generally offer superior accuracy in capturing price movements. The developed dashboard demonstrates the practical applicability of these models, integrating various analytical tools and visualizations to enhance user experience and decision-making capabilities. This research contributes to the growing body of knowledge in financial analytics and highlights the potential of machine learning in improving short-term stock price forecasting.

The code of my thesis can be found at https://github.com/praxitelisk/FinancialDashboard and the financial dashboard is placed at Streamlit https://financevue.streamlit.app/

Keywords: Artificial Intelligence, Machine Learning, Stock Price Prediction, Time Series Analysis, Regression Models, Deep Learning, Long Short-Term Memory (LSTM), Support Vector Machines (SVM), Random Forests, Gradient Boosting, Ensemble Methods, Feature Engineering, Technical Indicators, Financial Time Series, Autoregressive Integrated Moving Average (ARIMA), Data Preprocessing. Model Evaluation, Hyperparameter Tuning

Contents

1 Introduction

The stock market's movement of shares participating in it constitutes a speculative phenomenon, as it is influenced by many conditions and continuous changes, while characterized by increased fluctuations and non-linearity. The large number of factors affecting share prices can depend on each company's external and internal factors, such as changes in management or production. The development of technology, methods, and computational capabilities that have taken place since the early 21st century provides the possibility of deeper observation and prediction of these movements using complex mathematical models capable of accepting many parameters Researcher (2022). Artificial Intelligence allows for a new way of studying and predicting developments in the financial sector. This thesis aims to comprehensively study and analyze some of the most well-known and widely-used machine learning methods for the short-term prediction of share prices. The primary objectives is to evaluate their performance. Additionally, this research seeks to develop an interactive and user-friendly financial stock dashboard that leverages these machine learning techniques to provide real-time insights and predictions to investors. The dashboard will integrate various data sources, visualizations, and analytical tools to aid users in making informed investment decisions. Through this dual approach of theoretical study and practical application, the thesis aims to contribute valuable knowledge to the field of financial analytics and provide a useful tool for market participants.

1.1 Artificial Intelligence

Artificial Intelligence (AI) represents a term "umbrella" that covers various categories such as Machine Learning, Deep Learning (DL), and Artificial Neural Networks (ANNs). Often, the terms machine learning and deep learning tend to overlap because the techniques for approaching problems and the models used follow similar logic (IBM (a)).

1.2 Machine Learning

The term Machine Learning refers to the development and use of computing systems capable of "learning" and adapting without predefined instructions. This is achieved by using algorithms and statistical models capable of analyzing data and extracting information and patterns from them, resembling how humans learn. Through these models, it is possible to achieve. (IBM (b))

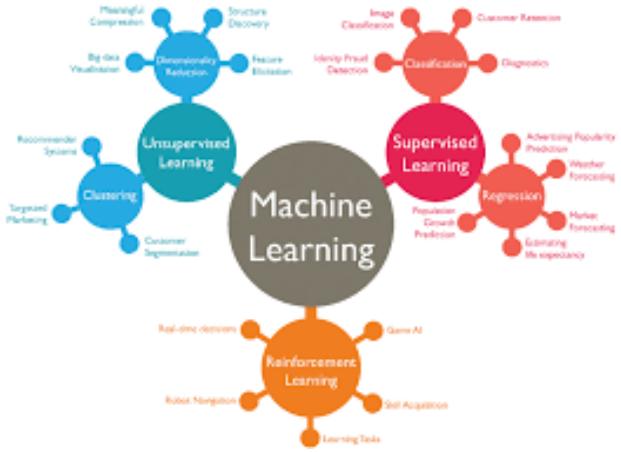

Figure 1: Machine Learning map

1.3 Thesis Goals

The primary goal of this thesis is to explore and evaluate various models for predicting the next-day closing prices of stocks, with a focus on improving the accuracy and reliability of these predictions. The research involves a comprehensive comparison of different predictive models, utilizing performance metrics such as Mean Absolute Percentage Error (MAPE) and Root Mean Square Error (RMSE) to assess their effectiveness. By systematically analyzing these models, the thesis aims to identify the best approaches for stock price prediction and highlight their respective strengths and weaknesses.

In addition to the predictive analysis, another significant objective of the thesis is to develop a web-based cloud service using Streamlit. This service integrates stock price prediction with a financial dashboard that provides real-time stock news and insights. The dashboard is designed to be user-friendly, offering investors and financial analysts a powerful tool to make informed decisions based on the latest market trends and predictive analytics. This integration of prediction models and financial news aims to enhance the overall utility of the service, providing a comprehensive resource for market analysis.

Ultimately, the thesis seeks to contribute to the field of financial forecasting by not only advancing predictive techniques but also by making these techniques accessible and actionable through an innovative cloud-based platform. The dual focus on model evaluation and practical application ensures that the research findings have both theoretical and real-world relevance, addressing the needs of both academic researchers and market practitioners.

2 Stock Price Prediction Methods - Theoretical Background

Primarily, four methods can be used to predict the course of a stock.

2.1 Technical Analysis

Technical analysis is a method used to evaluate and forecast the price movements of financial instruments such as stocks, commodities, currencies, and other tradable assets based on historical

market data. Unlike fundamental analysis, which focuses on evaluating a company's intrinsic value based on financial statements, economic indicators, and qualitative factors, technical analysis is grounded in the study of price and volume data. The primary assumption underlying technical analysis is that all relevant information is already reflected in the asset's price, and therefore, analyzing historical price movements can help predict future price trends (Achelis (2001)).

2.1.1 Core Principles of Technical Analysis

Technical analysis is built upon several core principles that guide its methodologies and tools firstly the price discounts everything. This principle asserts that all known and unknown information, including fundamentals, market psychology, and external factors, is already reflected in the asset's price. Therefore, analyzing price movements alone is sufficient to predict future price trends (Achelis (2001)).

Another principle is that price movements are not random. Technical analysts believe price movements are not random but follow identifiable and repeatable patterns over time. These patterns emerge due to collective market behaviour and recurring investor psychology. Moreover, the history tends to repeat itself. Market participants tend to react similarly to similar events over time, creating repetitive price patterns and trends. By studying historical price data, analysts can identify these patterns and make informed predictions about future price movements (Achelis (2001))

Finally, the trends persist until they reverse. trends, once established, are likely to continue until a clear reversal signal is identified. This principle emphasizes the importance of identifying and following trends, as they can persist for extended periods (Bulkowski (2020)).

2.1.2 Types of Charts Used in Technical Analysis

Technical analysts use various types of charts to visualize historical price data and identify patterns. The most commonly used chart types include:

- Line Charts:

Line charts connect closing prices over a specific period with a continuous line, providing a simple visual representation of price movements. They are useful for identifying overall trends but lack detailed information about intraday price fluctuations.

- Bar Charts:

Bar charts display a vertical line representing the high and low prices for each time period, with horizontal ticks indicating the opening and closing prices. They provide more detailed information about price movements within each period.

- Candlestick Charts:

Candlestick charts are similar to bar charts but use a rectangular body to represent the opening and closing prices, with wicks or shadows indicating the high and low prices. The body is filled or colored based on whether the closing price is higher or lower than the opening price, providing a visually intuitive representation of price movements.

- Point and Figure Charts:

Point and figure charts plot price movements without considering time. They use columns of Xs and Os to represent rising and falling prices, respectively, focusing solely on significant price changes and filtering out minor fluctuations (Investopedia (b)).

2.1.3 Key Concepts in Technical Analysis

Several key concepts underpin the methodologies and tools used in technical analysis:

- Trends

Trends are the general direction in which prices move over a specific period. They can be classified into three types: Uptrends are characterized by higher highs and higher lows, indicating a sustained increase in prices. Downtrends are characterized by lower highs and lower lows, indicating a sustained decrease in prices. Sideways Trends (Range-Bound): Occur when prices move within a horizontal range, indicating indecision or consolidation in the market (strike.money (b)).

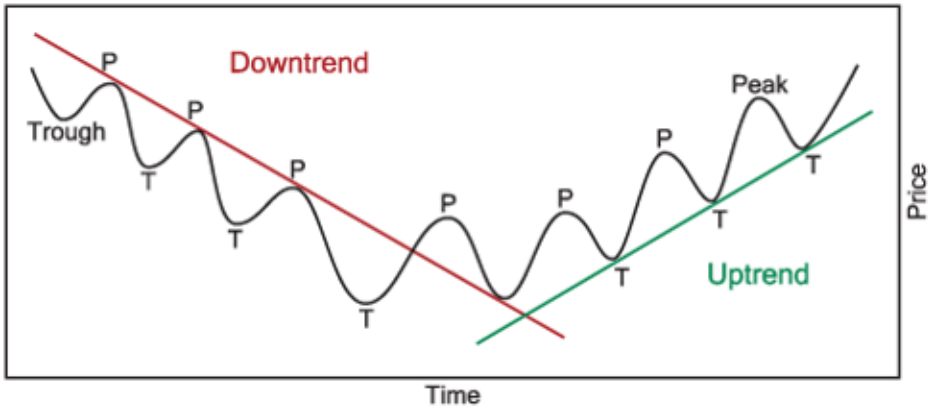

Figure 2: Trends

- Support and Resistance

Support and resistance levels are critical price levels where buying or selling pressure is expected to be strong enough to prevent further price movement. Support is the price level where a downtrend is expected to pause due to a concentration of buying interest. Resistance is the price level where an uptrend is expected to pause due to a concentration of selling interest. These levels are often identified through historical price patterns and can act as psychological barriers (strike.money (a)

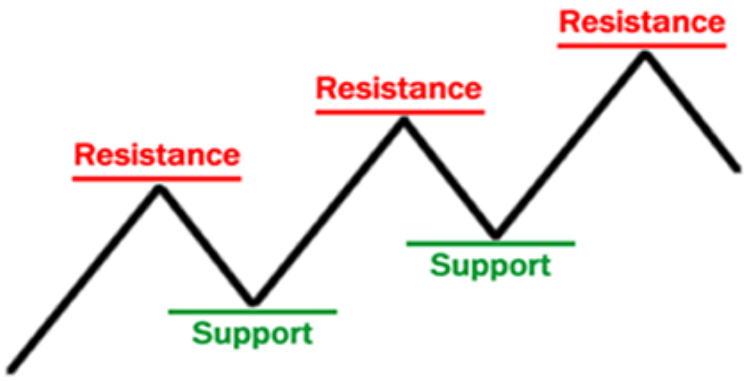

Figure 3: support and resistance

- Volume

Volume refers to the number of shares or contracts traded within a specific period. It is a crucial indicator of market activity and helps confirm the strength of price movements. High volume during price increases or decreases indicates strong market interest and can validate the trend strike.money (c).

- Momentum

Momentum measures the speed and strength of a price movement. Indicators such as the Relative Strength Index (RSI) and Moving Average Convergence Divergence (MACD) are used to assess momentum and identify overbought or oversold conditions Investopedia (d).

Figure 4: Momentum

- Volatility

Volatility measures the degree of variation in price movements over time. High volatility indicates significant price fluctuations, while low volatility suggests stable prices. Volatility indicators, such as Bollinger Bands, help assess market conditions and potential price breakouts Investopedia (c)

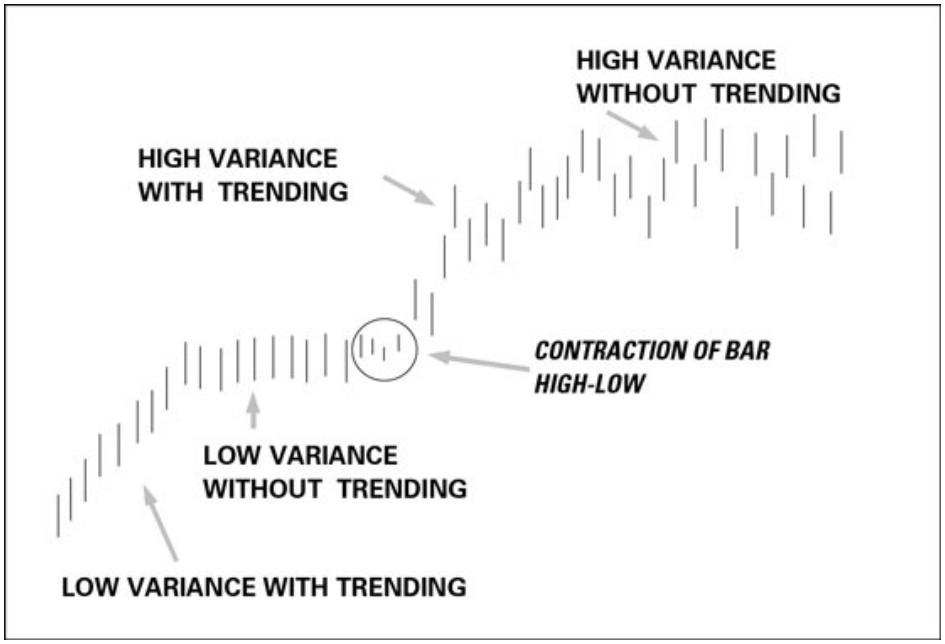

Figure 5: Volatility

In technical analysis, both technical indicators and patterns are studied to predict future price movements and make trading decisions

2.1.4 Patterns and Technical Indicators

In technical analysis, both technical indicators and patterns are studied to predict future price movements and make trading decisions.

- Technical Indicators

Technical indicators are mathematical calculations based on the price, volume, or open interest of a security. They are used to forecast future price movements and identify trends, momentum, volatility, and other aspects of the market. Common types of technical indicators include:

1. Trend Indicators: Moving Averages (MA): Simple Moving Average (SMA), Exponential Moving Average (EMA), Moving Average Convergence Divergence (MACD): Indicates the relationship between two moving averages of a security's price. Bollinger Bands: Uses moving averages and standard deviation to define high and low price levels (Anychart).

Indicator	**Formula**
Simple Moving Average (SMA)	$$\text{SMA} = \frac{1}{n}\sum_{i=1}^{n} P_i$$ where P_i is the price at the i-th period and n is the number of periods.
Exponential Moving Average (EMA)	$$\text{EMA} = P_t \cdot \frac{2}{n+1} + \text{EMA}_y \cdot \left(1 - \frac{2}{n+1}\right)$$ where P_t is the price at time t, n is the number of periods, and EMA_y is the previous EMA value.
Moving Average Convergence Divergence (MACD)	$$\text{MACD} = \text{EMA}_{12} - \text{EMA}_{26}$$ where EMA_{12} and EMA_{26} are the 12-day and 26-day exponential moving averages, respectively.
Bollinger Bands	$$\text{Upper Band} = \text{SMA} + k \cdot \sigma$$ $$\text{Lower Band} = \text{SMA} - k \cdot \sigma$$ where σ is the standard deviation, and k is a constant (typically 2).

Table 1: Technical Indicators of Trend and Their Formulas

2. Momentum Indicators: Relative Strength Index (RSI): Measures the speed and change of price movements, identifying overbought or oversold conditions. Stochastic Oscillator: Compares a particular closing price to a range of prices over a certain period, used to identify overbought or oversold conditions. Average Directional Index (ADX): Measures the strength of a trend (Anychart).

Indicator	**Formula**
Relative Strength Index (RSI)	$$\text{RSI} = 100 - \left(\frac{100}{1+RS}\right)$$ where $RS = \frac{\text{Average Gain}}{\text{Average Loss}}$.
Stochastic Oscillator	$$\%K = \frac{C - L_{14}}{H_{14} - L_{14}} \times 100$$ $$\%D = \text{SMA}_3(\%K)$$ where C is the most recent closing price, L_{14} is the lowest price over the last 14 periods, and H_{14} is the highest price over the last 14 periods.
Average Directional Index (ADX)	$$\text{ADX} = 100 \times \text{SMA}_n\left(\frac{\lvert DMI^+ - DMI^- \rvert}{DMI^+ + DMI^-}\right)$$ where DMI^+ and DMI^- are the positive and negative directional movement indicators, respectively.

Table 2: Technical Indicators of Momentum and Their Formulas

3. Volume Indicators: On-Balance Volume (OBV): Uses volume flow to predict changes in stock price. Volume Price Trend (VPT): Combines price and volume to confirm price trends. Chaikin Money Flow (CMF): Measures the volume-weighted average of accumulation and distribution over a specified period (Anychart).

Indicator	**Formula**
On-Balance Volume (OBV)	$\text{OBV} = \text{OBV}_{\text{previous}} + \begin{cases} V & \text{if } P_t > P_{t-1} \\ -V & \text{if } P_t < P_{t-1} \\ 0 & \text{if } P_t = P_{t-1} \end{cases}$ where V is the volume at time t.
Volume Price Trend (VPT)	$\text{VPT} = \text{VPT}_{\text{previous}} + V\left(\frac{P_t - P_{t-1}}{P_{t-1}}\right)$ where V is the volume at time t.
Chaikin Money Flow (CMF)	$\text{CMF} = \frac{\sum_{i=1}^{n}\left(\frac{(C_i - L_i) - (H_i - C_i)}{H_i - L_i} \times V_i\right)}{\sum_{i=1}^{n} V_i}$ where H_i, L_i, C_i, and V_i are the high, low, close, and volume for the i-th period, respectively.

Table 3: Technical Indicators of Volume and Their Formulas

4. Volatility Indicators: Average True Range (ATR): Measures market volatility. Standard Deviation: Measures the dispersion of price from its average. Volatility Index (VIX): Measures the market's expectation of 30-day volatility (Anychart).

Indicator	**Formula**
Average True Range (ATR)	$$\text{ATR} = \frac{1}{n}\sum_{i=1}^{n} TR_i$$ where TR_i is the true range for the i-th period, calculated as $$TR_i = \max(H_i - L_i, \lvert H_i - C_{i-1}\rvert, \lvert L_i - C_{i-1}\rvert)$$ where H_i, L_i, and C_{i-1} are the high, low, and close of the previous period, respectively.
Standard Deviation	$$\sigma = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(P_i - \mu)^2}$$ where P_i is the price at the i-th period and μ is the mean price over n periods.

Table 4: Technical Indicators and Their Formulas

- Patterns

Technical patterns are formations created by the price movements of a security on a chart, and they are used to predict future price movements based on historical data. These patterns can be broadly categorized into two types: chart patterns and candlestick patterns.

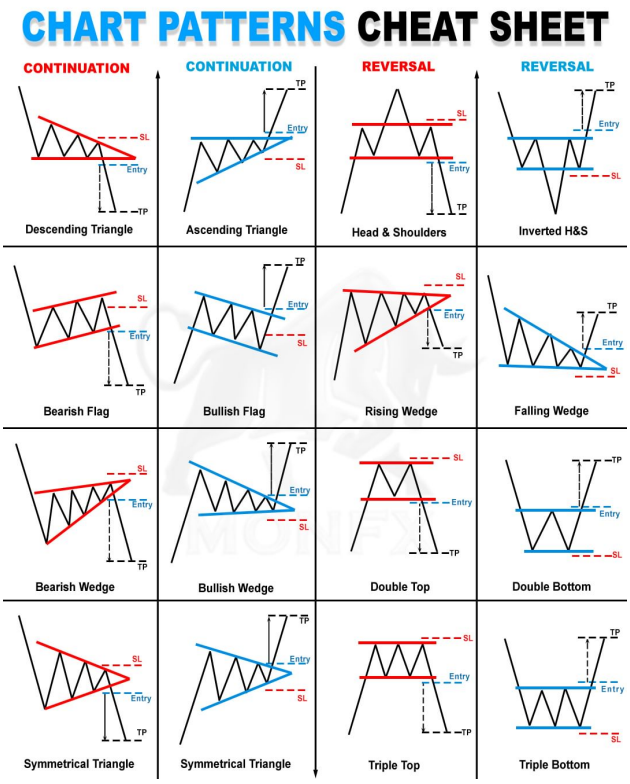

Figure 6: Chart Patterns

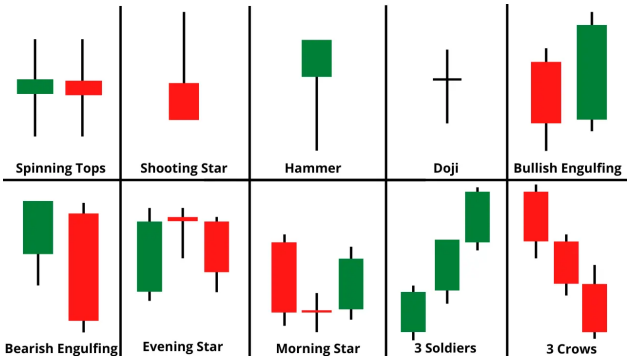

Figure 7: Candlestick Patterns

In technical analysis, both technical indicators and patterns play crucial roles in predicting future price movements. Technical indicators provide quantitative measures of market conditions and trends, while patterns offer visual cues based on historical price action. Together, these tools help traders make informed decisions by analyzing past market behavior and anticipating future movements.

2.1.5 Applications of Technical Analysis

Technical analysis is widely used by traders and investors in various financial markets, including stocks, commodities, currencies, and cryptocurrencies. Its applications help to identify the direction

of the market trend, allowing traders to align their trades with the prevailing trend. By following trends, traders can increase their chances of profitable trades. Moreover, technical analysis provides tools and indicators to identify optimal entry and exit points for trades. By analyzing support and resistance levels, chart patterns, and technical indicators, traders can make informed decisions about when to enter or exit a position. Also, technical analysis provides insights into market timing, allowing traders to enter and exit trades at the most opportune moments. By analyzing price patterns, volume, and technical indicators, traders can time their trades more effectively. The drawback of this method is that many technical indicators use lagging features, meaning they provide signals after the price movement has occurred. However, the technical analysis can produce false signals, leading to incorrect predictions and potential losses. Also, when we overemphasise then we rely on historical patterns that may not always accurately predict future price movements. And finally, we cannot predict market anomalies with sudden market changes or external events that can impact prices.

2.2 Fundamental Analysis

Fundamental analysis is a method of evaluating a stock to determine its intrinsic value by examining related economic, financial, and other qualitative and quantitative factors. Unlike technical analysis, which focuses on historical price movements and trading volumes, fundamental analysis delves into the financial health and performance of a company to predict future stock price movements. The goal of fundamental analysis is to determine whether a stock is overvalued or undervalued based on its current price relative to its intrinsic value Investopedia (a).

2.2.1 Key Components of Fundamental Analysis

Fundamental analysis involves a comprehensive examination of a company's financial health and intrinsic value by scrutinizing various data points from the company's financial statements. This includes the balance sheet, which provides a snapshot of the company's assets, liabilities, and shareholders' equity at a specific point in time, offering insights into its financial stability and capital structure. Additionally, fundamental analysis delves into the company's income statement to assess its revenue, expenses, and profitability over a reporting period, helping analysts understand its operational efficiency and earnings potential. Furthermore, the cash flow statement is analyzed to evaluate the company's liquidity and cash generation capabilities, ensuring it has sufficient cash to meet its obligations, reinvest in its operations, and return value to shareholders. By integrating these financial documents, fundamental analysts aim to determine the true value of a company's stock, identify potential for growth, and make informed investment decisions Investopedia (a).

Fundamental analysis uses various factors that can impact a company's performance and, consequently, its stock price. These factors can be broadly categorized into two types: quantitative and qualitative Investopedia (a).

2.2.2 Quantitative Factors

Quantitative factors encompass elements that can be numerically measured, such as a company's assets, liabilities, cash flow, revenue, and the price-to-earnings ratio. The primary aim of fundamental

analysis is to generate a numerical value that investors can use to compare with the current market price of a security, thereby assessing whether it is undervalued or overvalued Investopedia (a).

2.2.3 Qualitative Factors

Qualitative factors are non-numerical elements that influence a company's performance. Various factors such as Management and Leadership; The experience, reputation, and track record of the company's management team. The company's governance practices and board composition. Also, the Industry's current conditions which is the overall health and growth prospects of the industry in which the company operates. Market trends, competitive landscape, and regulatory environment Investopedia (a).

Steps in Fundamental Examination 1. Economic Analysis: Assess the broader economic environment to understand the macroeconomic factors that can impact the stock market and individual companies. This includes analyzing economic indicators, fiscal and monetary policies, and global economic trends.

2. Industry Analysis: Evaluate the industry in which the company operates. This involves examining industry trends, competitive dynamics, barriers to entry, regulatory factors, and the industry's growth potential.

3. Company Analysis: Conduct a thorough analysis of the company's financial statements to assess its financial health and performance. This includes calculating and interpreting various financial ratios to gain insights into profitability, liquidity, leverage, and efficiency. Analyze qualitative factors such as management quality, competitive advantages, product offerings, and strategic initiatives Investopedia (a).

2.2.4 Valuation Methods

- Discounted Cash Flow (DCF) Model

The DCF model estimates the intrinsic value of a stock based on the present value of its expected future cash flows. This involves forecasting the company's free cash flows and discounting them back to their present value using an appropriate discount rate Wafi et al. (2015).

- Price-to-Earnings (P/E) Ratio

The P/E ratio compares a company's current stock price to its earnings per share (EPS). It provides insights into how much investors are willing to pay for each dollar of earnings. A high P/E ratio may indicate that the stock is overvalued, while a low P/E ratio may suggest that it is undervalued Bajajfinserv.

- Price-to-Book (P/B) Ratio

The P/B ratio compares a company's market value to its book value (net asset value). It helps assess whether a stock is trading at a premium or discount relative to its book value Bajajfinserv.

- Dividend Discount Model (DDM)

The DDM estimates the intrinsic value of a stock based on the present value of its expected future dividends. This model is particularly useful for valuing dividend-paying stocks Wafi et al. (2015).

- Residual Income Valuation Model

The Residual Income Valuation Model is a method for valuing a company's stock that focuses on the residual income, which is the net income generated by the company after accounting for the cost of equity. Essentially, it calculates the value of a company by adding the book value of equity to the present value of future residual incomes. This model is useful because it adjusts for accounting distortions and focuses on economic profit rather than accounting profit Wafi et al. (2015).

2.3 Time Series Analysis

With time series analysis the prediction is based on the historical data of each stock and the analysis of these as a time series. Methods such as linear interpolation and AutoRegressive Integrated Moving Average (ARIMA) models are included in this category Hyndman & Athanasopoulos (2018).

2.3.1 Time Series Analysis in Stock Price Prediction

Time series analysis is a statistical technique that deals with analyzing time-ordered data points to extract meaningful statistics and identify patterns that can aid in forecasting future values. This method is particularly relevant for stock price prediction as stock prices are sequential data points recorded at regular intervals Hyndman & Athanasopoulos (2018).

2.3.2 Key Concepts in Time Series Analysis

Key concepts in time series analysis include stationarity, which refers to a time series whose statistical properties, such as mean and variance, are constant over time. Another important concept is autocorrelation, which measures the degree of similarity between a given time series and a lagged version of itself over successive time intervals. Seasonality refers to regular, repeating patterns or cycles of behavior over a specific period, such as daily, monthly, or annually, driven by seasonal factors. A trend, on the other hand, is the long-term movement or direction in the data over a prolonged period, indicating an upward, downward, or constant trajectory in the time series Hyndman & Athanasopoulos (2018).

- Stationarity

A time series is considered stationary if its statistical properties, such as mean, variance, and autocorrelation, remain constant over time. Many time series analysis techniques assume stationarity, making it a crucial concept. Non-stationary series often need to be transformed to stationary through differencing, logging, or detrending Science (2019).

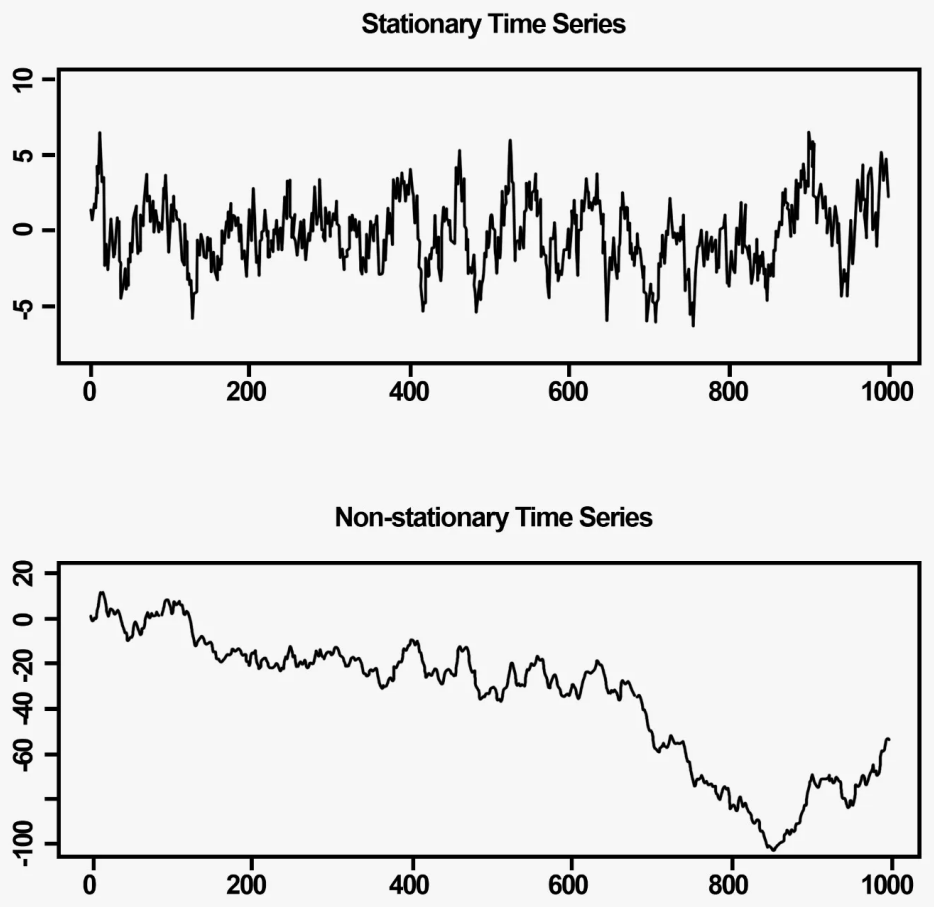

Figure 8: Time Series Stationarity

- Autocorrelation

Autocorrelation measures the correlation between time series values at different lags. It helps identify patterns such as seasonality and trend. The Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF) are tools used to detect the extent of correlation at various lags Frost (2023).

- Seasonality

Seasonality refers to periodic fluctuations in a time series that occur at regular intervals, such as daily, monthly, or yearly patterns. Identifying and modeling seasonality is essential for accurate forecasting Arpal (2020).

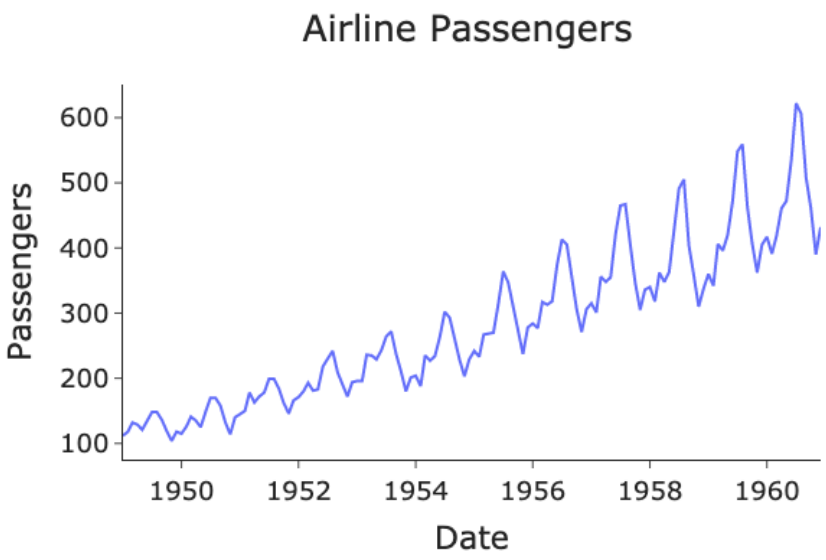

Figure 9: Time Series Seasonality

- Trend

Trend refers to the long-term progression of a time series, which can be upward, downward, or flat. It indicates the general direction of the series over a longer period Arpal (2020).

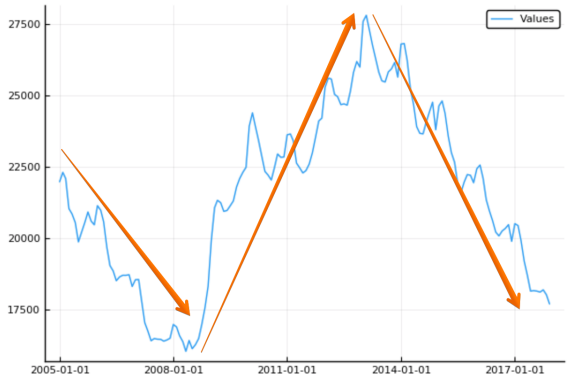

Figure 10: Time Series Trend

2.3.3 Methods of Time Series Analysis

- Decomposition

Decomposition involves breaking down a time series into its constituent components: trend, seasonality, and residual (irregular) components. This helps in understanding and modeling each component separately. Additive and multiplicative decomposition are two common approaches Hyndman & Athanasopoulos (2018).

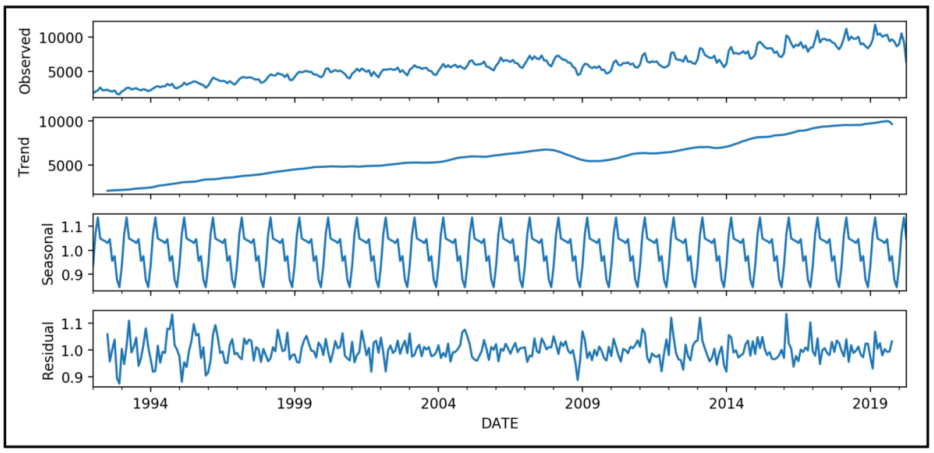

Figure 11: Time Series Decomposition

- Smoothing

Smoothing techniques like Moving Averages and Exponential Smoothing are used to reduce noise and highlight the underlying trend and seasonality in the data Hyndman & Athanasopoulos (2018).

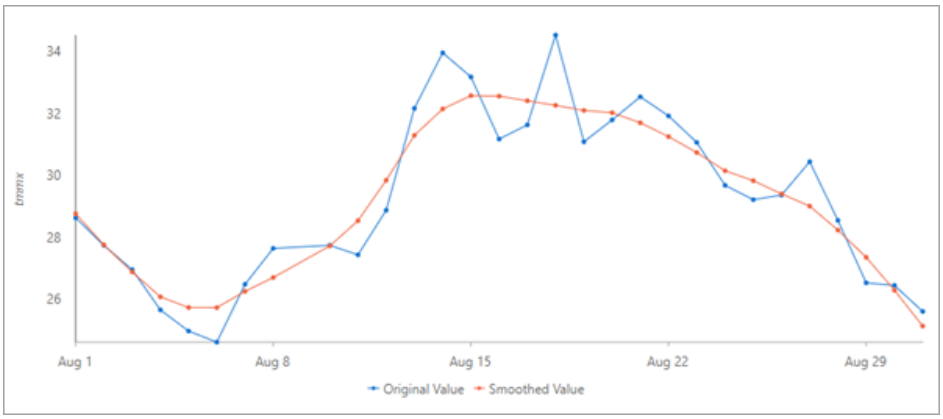

Figure 12: Time Series Smoothing

- Autoregressive Integrated Moving Average (ARIMA)

ARIMA is a widely used statistical model for time series forecasting. It combines autoregressive (AR) terms, differencing (I), and moving average (MA) terms. The model is denoted as ARIMA(p, d, q), where: - p: Number of lag observations in the model (AR terms). - d: Degree of differencing to make the series stationary. - q: Size of the moving average window Hyndman & Athanasopoulos (2018).

- Seasonal ARIMA (SARIMA)

SARIMA extends ARIMA to handle seasonality in the data. It includes seasonal autoregressive (SAR), seasonal differencing (SD), and seasonal moving average (SMA) terms Hyndman & Athanasopoulos (2018).

- Exponential Smoothing State Space Model (ETS)

ETS models, including Holt-Winters methods, are used for forecasting time series data with trend and seasonal components. They apply exponential smoothing to capture the level, trend, and seasonality Hyndman & Athanasopoulos (2018).

- Vector Autoregression (VAR)

VAR models capture the linear dependencies among multiple time series. It is useful when forecasting systems where multiple time series influence each other Hyndman & Athanasopoulos (2018).

2.3.4 Arima - Time Series

A time series is a series of observations collected at specific time points or periods that are equidistant. Here, the reference time chosen is not the physical but the economic trading time, assuming a constant step.

Some well-known time series problems include weather forecasting, rainfall prediction in a specific period, and many operational research problems such as product and service demand/supply. It is evident that time series use can provide an initial approach and be used in combination with other methods to yield better results in problems affected by exogenous factors.

The ARIMA method is considered one of the best-known time series prediction methods and relies on describing the autocorrelation that exists among the data.

Autoregressive Integrated Moving Average (ARIMA) models are a class of models used for forecasting time series data. They are particularly useful for non-stationary data, which can be transformed into stationary data by differencing. The ARIMA model is composed of three key components: Autoregressive (AR), Integrated (I), and Moving Average (MA).

1. Autoregressive (AR) Component The AR component of an ARIMA model specifies that the output variable depends linearly on its own previous values. This can be written as:

$$Y_t = c + \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + \ldots + \phi_p Y_{t-p} + \varepsilon_t$$

where: - Y_t is the value at time t, - c is a constant, - $\phi_1, \phi_2, \ldots, \phi_p$ are the parameters of the autoregressive part, - p is the number of lag observations included in the model (i.e., the order of the AR model), - ε_t is white noise.

2. Integrated (I) Component The integrated part of the ARIMA model is the differencing of raw observations to make the time series stationary. Differencing involves subtracting the previous observation from the current observation:

$$Y_t' = Y_t - Y_{t-1}$$

where Y_t' is the differenced series. This can be done d times if needed, where d is the order of differencing.

3. Moving Average (MA) Component The MA component models the error of the time series as a linear combination of error terms occurring contemporaneously and at various times in the past. This can be written as:

$$Y_t = \mu + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \ldots + \theta_q \varepsilon_{t-q}$$

where: - μ is the mean of the series, - $\theta_1, \theta_2, \ldots, \theta_q$ are the parameters of the moving average part, - q is the order of the MA model, - ε_t is white noise.

ARIMA Model Specification An ARIMA model is generally denoted as ARIMA(p, d, q), where: - p is the number of lag observations in the AR model, - d is the number of times that the raw observations are differenced, - q is the size of the moving average window.

The model combines these components as follows:

$$\text{ARIMA}(p,d,q) : Y'_t = c + \phi_1 Y'_{t-1} + \ldots + \phi_p Y'_{t-p} + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \ldots + \theta_q \varepsilon_{t-q}$$

where Y'_t represents the differenced series if d is greater than 0.

Assumptions of ARIMA Models 1. Stationarity: The time series should be stationary (mean, variance, and covariance should be constant over time). Differencing is applied to achieve stationarity. 2. No Autocorrelation in Residuals: The residuals (errors) from the model should be white noise, meaning they should not show any autocorrelation. 3. Linearity: ARIMA models assume a linear relationship between past values and the present value.

2.3.5 Advantages and Limitations of Time Series Analysis

Time series analysis offers several advantages and limitations. One major advantage is its ability to recognize patterns such as trends and seasonality, which can significantly aid in forecasting. It also provides precise numerical predictions and can model dependencies within a series or between multiple time series. However, there are limitations: many models require the series to be stationary, necessitating preprocessing steps. Additionally, time series models can be complex to implement and interpret, particularly for non-experts, and they are often more suitable for short- to medium-term forecasting, with reduced accuracy for long-term predictions Team.

2.4 Machine Learning

With machine learning the stock price prediction is based on the use of Artificial Intelligence (AI) and Machine Learning models which are trained through data feeding and analysis. Known methods include Regression Trees, Support Vector Machines (SVM), k-Nearest Neighbors, Ensemble techniques such as Random Forest and XGBoost which are supervised machine learning algorithms. Long short-term memory networks (LSTM) are a special category of neural networks using specific assumptions about the data. Input data may consist of historical data, sentiment analysis, etc. Another path of Machine learning called unsupervised learning requires recognizing patterns without providing pre-classified data. The process is performed by identifying correlations among the provided data without human supervision.

2.4.1 Supervised Learning

Supervised learning models include all models that perform the learning process based on pre-existing input-output pairs, understanding the classification methods of these. These models, often with modifications of existing classification models, also offer solutions to regression problems. This

process begins with the split of the data into two categories: training data and test data. Generally, the split is around 70%-30%, with the larger percentage corresponding to training data. A typical example of supervised learning is the recognition of different types of vehicles based on images.

Various supervised learning techniques have been used to determine stock trends, with notable results obtained from Support Vector Machine techniques, one of the best-known methods for solving classification problems. In this specific case, Support Vector Regression (SVR) algorithms are used.

Another technique studied in this scenario is eXtreme Gradient Boosting (XGBoost). This algorithm uses historical data related to the stock's opening price, volume, and possibly other features like market demand in the specific sector. This method shows impressive results, especially for long-term trend predictions, with an accuracy of 87-99%.

A subcategory of RNN, Echo State Networks (ESN), as per the "Financial Market Time Series Prediction with Recurrent Neural Networks. State College: Citeseer" study, shows significant improvement compared to the optimal estimator technique used in the Kalman filter.

The Long Short-Term Memory (LSTM) network belongs to the RNN category and studies data in time series form. The first complete proposal was presented in 2017 by David M. Q. Nelson and his collaborators Nelson et al. (2017a).

2.4.2 SVR

Support Vector Machines (SVMs) are a fundamental category of machine learning methods used for classification problems, particularly when data are not linearly separable. SVM models are widely used in the scientific literature to predict the stock price movements Joseph (2019), Madge & Bhatt (2015). SVMs achieve the classification of nonlinear data using kernels, which linearize the data by projecting them into higher dimensions. Based on this principle, SVMs are employed for prediction tasks involving real-valued data. Likewise, Support Vector Regression (SVR) is a method used to predict continuous outcomes like stock prices. SVR works by finding the best line (or hyperplane) that predicts the target variable (e.g., closing price) within a margin of error.

To address the issue of the infinite range of real-valued predictions, an error tolerance hyperparameter ε is used to calculate the maximum acceptable error from the predicted value. The other several key hyperparameters in SVR which are the kernel, the C and Gamma as described below.

The use of kernels can be implemented in various ways (e.g., linear, polynomial, sigmoid), with the optimal choice for stock prediction often being the radial basis function (rbf), summarized by the following equation:

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|^2\right) \tag{1}$$

where $\gamma > 0$ is a parameter that defines the spread of the kernel, and $\|\mathbf{x}_i - \mathbf{x}_j\|$ is the Euclidean distance between the two feature vectors $\mathbf{x}_i$ and $\mathbf{x}_j$.

The use of kernels can be implemented in various ways (e.g., linear, polynomial, sigmoid), with the optimal choice for stock prediction often being the radial basis function (RBF), summarized by the following equation:

$$K(x, x') = \exp\left(-\gamma \|x - x'\|^2\right)$$

The hyperparameter γ defines the inverse of the standard deviation of the Radial Basis Function

(RBF) and represents the similarity between two points. Smaller γ values lead to RBFs with high variance, causing distant data points to be grouped together due to their strong influence on neighboring values. Conversely, larger γ values result in a lower influence of data points on neighboring regions, leading to more precise classification based on closer data points. Balances between fitting the training data well and keeping the model simple. High C means fewer training errors but can overfit; low C allows more errors but is simpler.

The hyperparameter C (regularization parameter) determines the number of predictions allowed to have an error greater than the defined ε. A larger C results in fewer misclassified predictions by reducing the margin width, whereas a smaller C leads to a larger margin width and a higher likelihood of misclassification. The hyperparameter C is described by the equation:

$$C = \frac{1}{2}\sum_{i=1}^{n}(y_i - f(x_i))^2 + \frac{\lambda}{2}\|w\|^2$$

ε - insensitive Loss Function: Only penalizes errors larger than ε, ignoring small errors to avoid over-fitting.

The figure below depicts how SVR are being trained from continuous-valued data.

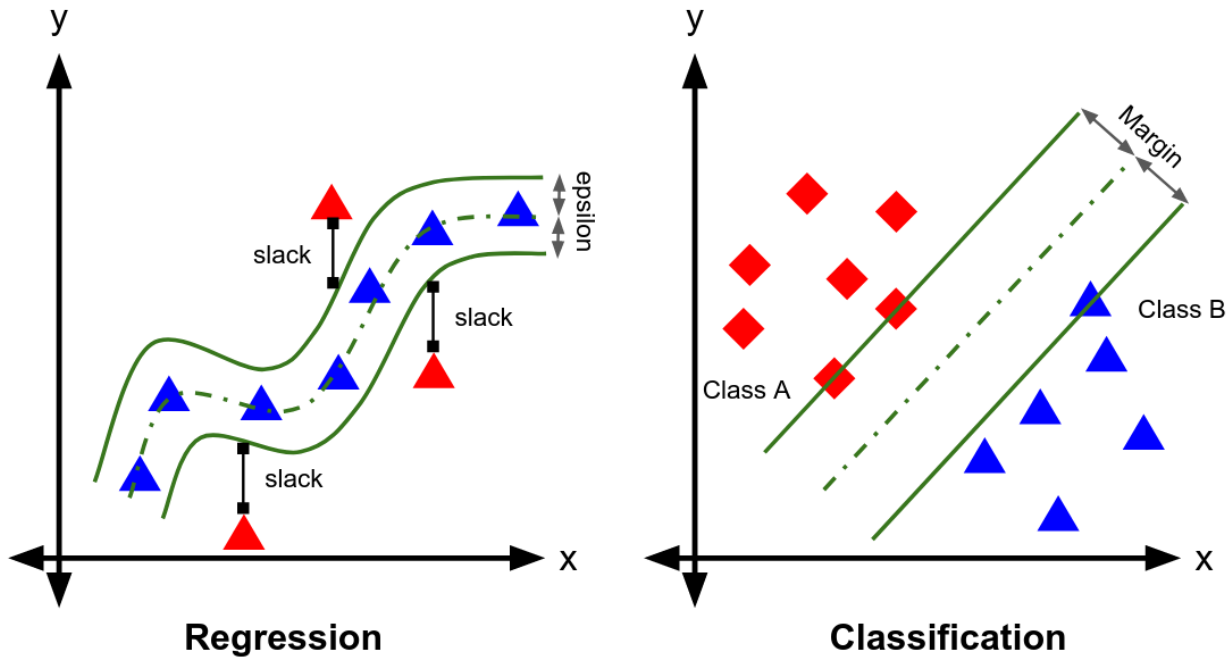

Figure 13: SVR

2.4.3 CART - Classification and Regression trees

Classification and Regression Trees (CART) are a type of decision tree used for predicting continuous outcomes. In regression tasks, CART trees model the relationship between input features and a continuous target variable by recursively splitting the data into subsets.

Key Concepts

1. **Tree Structure**: A CART regression tree consists of internal nodes, branches, and leaf nodes. Internal nodes represent feature-based decisions, branches represent the outcomes of those decisions, and leaf nodes represent the predicted values.

2. **Recursive Binary Splitting**: The tree is built using a process called recursive binary splitting, which divides the data into two subsets at each node based on a feature and a split point. The goal is to minimize the sum of squared residuals (SSR), which measures the difference between the observed and predicted values.

$$SSR = \sum_{i=1}^{N} (y_i - \hat{y}_i)^2$$

3. **Splitting Criterion**: The splitting criterion for regression trees is usually based on minimizing the variance within each subset. For a given split, the variance reduction is calculated as:

$$\Delta\text{Var} = \text{Var}(T) - \left(\frac{N_L}{N} \text{Var}(T_L) + \frac{N_R}{N} \text{Var}(T_R) \right)$$

where T is the parent node, T_L and T_R are the left and right child nodes, and N is the number of instances.

4. **Pruning**: To prevent overfitting, CART trees are often pruned by removing nodes that provide little predictive power. This process can be done using cost-complexity pruning, which balances the tree's complexity with its predictive accuracy.

$$C_\alpha(T) = \text{Err}(T) + \alpha \cdot |\text{leaves}(T)|$$

where $\text{Err}(T)$ is the error of the tree T, $|\text{leaves}(T)|$ is the number of leaves, and α is a regularization parameter.

Applying CART Trees to Stock Price Prediction

In the context of predicting stock prices, CART trees are beneficial because they can handle complex, nonlinear relationships between input features (such as historical prices and trading volumes) and the target variable (next day's closing price). By recursively partitioning the data, CART trees create a model that can adapt to different market conditions and provide interpretable predictions.

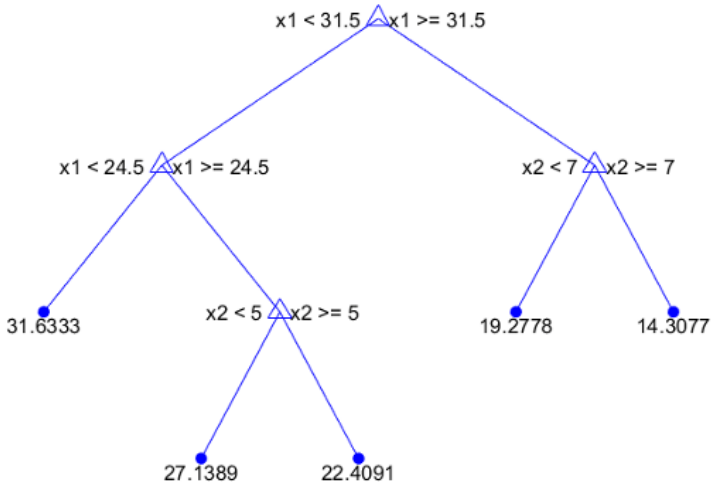

Figure 14: Regression Trees

2.4.4 XGBoost

Since CART models were explained above, we can now introduce an ensemble technique called XGBoost. The Extreme Gradient Boosting (XGBoost) method is a technique categorized under supervised learning, used for both classification and regression problems. XGBoost is an ensemble learner. The model relies on the use of distinct weak learners - trees, which are models that make predictions with limited accuracy, employed in an iterative manner. It is considered on of the top data mining techniques Wu et al. (2008).

Xgboost consists of Weak learners. Weak learners are characterized by the use of machine learning algorithms that behave in a very simplistic manner (naive algorithms). This category includes linear regression algorithms.

Gradient boosting involves minimizing the loss function by adding weak learners (decision trees) sequentially. Each tree is built to minimize the residual errors of the previous trees. XGBoost incorporates regularization to prevent overfitting and improve generalization.

Objective Function: The objective function in XGBoost includes a loss function and a regularization term:

$$\text{Obj} = \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k)$$

where l is the loss function (e.g., mean squared error), Ω is the regularization term, y_i is the actual value, $\hat{y}_i$ is the predicted value, and f_k represents the k-th tree.

Tree Structure: Each decision tree is represented as $f(x)$, and the prediction for a given input is the sum of predictions from all trees:

$$\hat{y}_i = \sum_{k=1}^{K} f_k(x_i)$$

Regularization Term: The regularization term for a tree includes both the complexity of the tree and the leaf weights:

$$\Omega(f) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2$$

where T is the number of leaves, γ is a complexity parameter, λ is a regularization parameter, and w_j are the weights of the leaves.

Gradient and Hessian: The optimization in XGBoost uses both the gradient and the Hessian of the loss function:

$$g_i = \frac{\partial l(y_i, \hat{y}_i)}{\partial \hat{y}_i}, \quad h_i = \frac{\partial^2 l(y_i, \hat{y}_i)}{\partial \hat{y}_i^2}$$

Tree Building: For each leaf node, the score is calculated as:

$$w_j = -\frac{\sum_{i \in I_j} g_i}{\sum_{i \in I_j} h_i + \lambda}$$

where I_j represents the instances in leaf j.

For predicting stock prices, XGBoost is effective because it can handle large datasets, manage missing values, and prevent overfitting through regularization. By using gradient boosting, XGBoost iteratively improves prediction accuracy, making it a powerful tool for financial forecasting and aiding investment decisions.

The following figure, on the next page, provides a schematic representation of the XGBoost method's functionality.

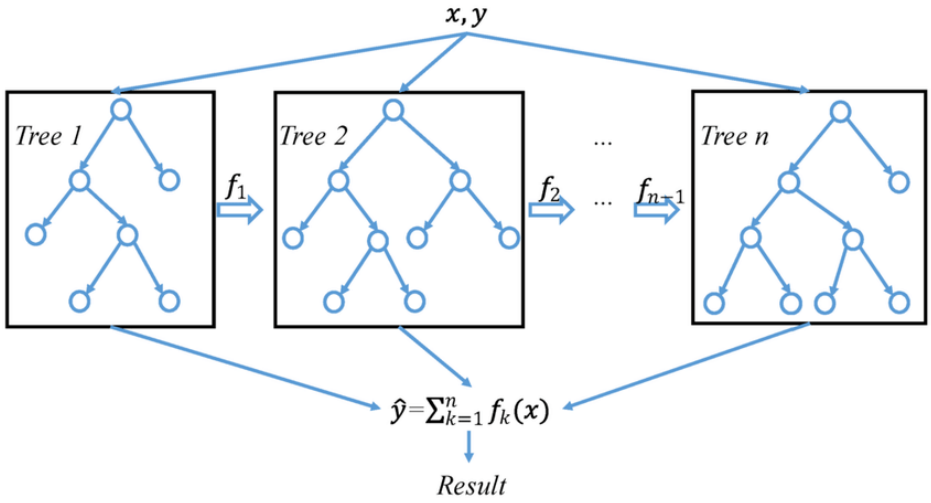

Figure 15: Enter Caption

2.4.5 Random Forest for Regression

Random Forest is a popular ensemble learning method used for both classification and regression tasks. It operates by constructing multiple decision trees during training and outputting either the

mode of the classes (classification) or the mean prediction (regression) of the individual trees Amit & Geman (1997).

Random Forest employs a technique called bootstrap aggregating, or bagging, which helps to reduce the variance of the model. In this process, multiple subsets of the training data are sampled with replacement. For each subset, a decision tree is constructed. The algorithm for bagging is as follows:

1. Select B bootstrap samples from the original dataset. 2. Train a decision tree f_b on each bootstrap sample. 3. For a new data point, make predictions by averaging the predictions from all the B trees (for regression) or taking a majority vote (for classification).

$$\hat{f}(x) = \frac{1}{B} \sum_{b=1}^{B} f_b(x)$$

In addition to bagging, Random Forest introduces random feature selection. At each split in the decision tree, a random subset of the features is selected, and the best split is found only within this subset. This process helps to reduce the correlation between the individual trees.

Let m be the number of features in the dataset. For each split in the tree, a number m_{subset} of features is randomly chosen, where $m_{\text{subset}} < m$. This results in the following algorithm:

1. For each node in the tree, randomly select m_{subset} features.
2. Find the best split among these features.
3. Split the node into two child nodes based on this split criterion.

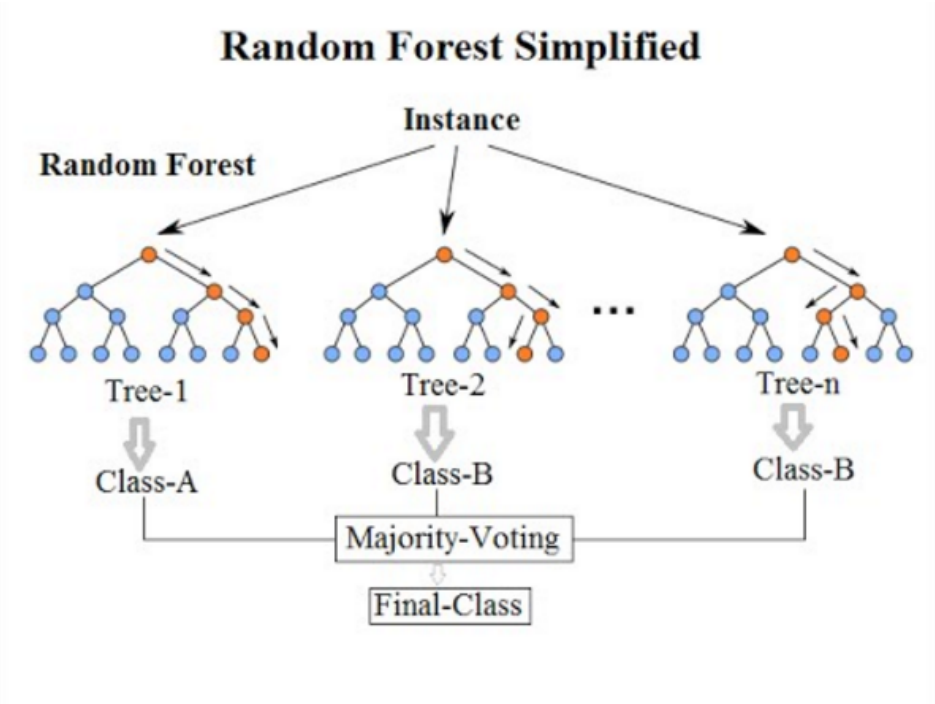

Figure 16: Random Forest trees

2.4.6 k-Nearest Neighbors

Theoretical Background of k-Nearest Neighbors (k-NN)

The k-Nearest Neighbors (k-NN) algorithm is a simple, intuitive, and powerful machine learning method used for both classification and regression tasks. It is a non-parametric, instance-based learning technique that does not make any assumptions about the underlying data distribution. Here, we will delve into the theoretical foundation of k-NN, covering its basic principles, distance metrics, parameter selection, strengths, and limitations.

1. Basic Principles

The k-NN algorithm operates on the principle of similarity. It predicts the output for a given input by analyzing the outputs of the k nearest data points in the training set. The algorithm can be broken down into the following steps:

1. Store the Training Data: - The algorithm stores all the training data points. Unlike other algorithms, k-NN does not create a model during the training phase. Instead, it memorizes the dataset.

2. Distance Calculation: - For a given test point, the distance between the test point and all training points is calculated using a chosen distance metric.

3. Identification of Nearest Neighbors: - The algorithm identifies the k training points that are closest to the test point. These k points are referred to as the ”nearest neighbors.”

4. Prediction: - Classification: The algorithm assigns the class label most frequent among the k nearest neighbors to the test point. - Regression: The algorithm predicts the value as the average (or weighted average) of the values of the k nearest neighbors.

2. Distance Metrics

The choice of distance metric is crucial for the performance of the k-NN algorithm. Common distance metrics include:

1. Euclidean Distance: - The most widely used distance metric in k-NN, Euclidean distance measures the straight-line distance between two points in a multidimensional space.

$$d(p,q) = \sqrt{\sum_{i=1}^{n}(p_i - q_i)^2}$$

2. Manhattan Distance: - Also known as the L1 distance or city block distance, it measures the sum of the absolute differences between the coordinates of the points.

$$d(p,q) = \sum_{i=1}^{n}|p_i - q_i|$$

3. Minkowski Distance: - A generalization of both Euclidean and Manhattan distances, Minkowski distance adds a parameter p to control the type of distance.

$$d(p,q) = \left(\sum_{i=1}^{n}|p_i - q_i|^p\right)^{1/p}$$

- For $p = 1$, it becomes Manhattan distance; for $p = 2$, it becomes Euclidean distance.

4. Cosine Similarity: - Measures the cosine of the angle between two non-zero vectors,

focusing on the orientation rather than the magnitude.

$$\text{similarity}(A,B) = \frac{A \cdot B}{\|A\|\|B\|}$$

- Often used for text data and high-dimensional spaces.

5. Hamming Distance: - Used for categorical data, Hamming distance counts the number of positions at which corresponding symbols are different.

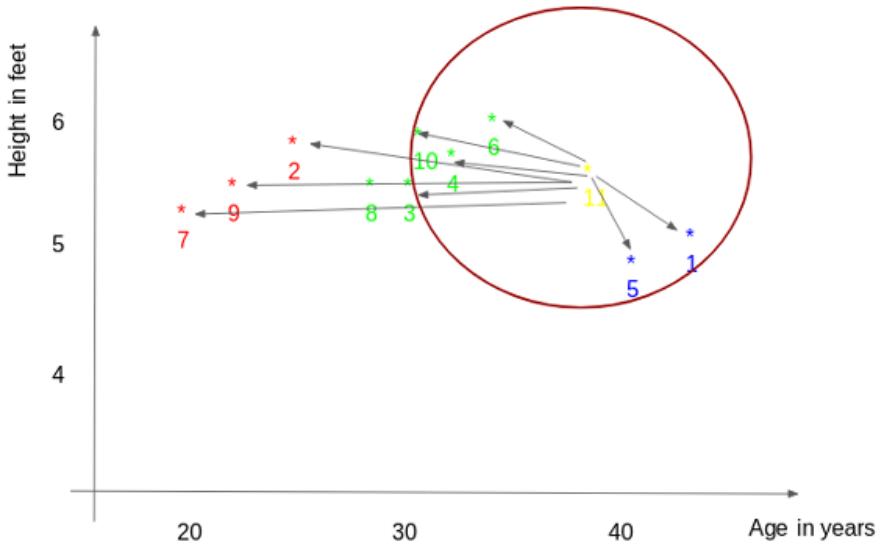

Figure 17: kNN for Regression

2.4.7 LSTM

Long Short-Term Memory (LSTM) networks are a special kind of recurrent Neural Network (RNN) designed to model sequences and long-term dependencies more effectively than traditional RNNs. They were introduced by Hochreiter and Schmidhuber in 1997 to address the vanishing and exploding gradient problems that can occur during the training of standard RNNs. This detailed exploration will cover the fundamental concepts, structure, functionality, and advantages of LSTMs Hochreiter & Schmidhuber (1997).

Recurrent Neural Networks (RNNs) are a class of neural networks that are particularly well-suited for sequence data, such as time series, text, and speech. Unlike feedforward neural networks, RNNs have connections that form cycles, allowing them to maintain a state that can capture information about previous inputs. This makes them powerful for tasks where context and order matter Zhu (2020).

However, standard RNNs struggle with learning long-term dependencies due to the vanishing gradient problem, where gradients of the loss function with respect to the parameters diminish exponentially during backpropagation through time, making it difficult for the network to learn and update weights effectively for earlier layers Karmiani et al. (2019).

- The Structure of LSTM Networks

LSTM networks extend RNNs with a more complex structure that includes special units called memory cells. These cells are capable of maintaining information for long periods, thereby mitigating the vanishing gradient problem. The key components of an LSTM cell are the following based on the scientific literature Varsamopoulos et al. (2018).

- Cell State (C_t):

- The cell state acts as a memory that carries information across different time steps. It can be thought of as a conveyor belt that runs through the entire LSTM cell, with only minor linear interactions.

- Forget Gate (f_t):

- The forget gate decides what information to discard from the cell state. It takes the current input (x_t) and the previous hidden state (h_{t-1}) as inputs and passes them through a sigmoid activation function to produce a value between 0 and 1.

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$$

- Input Gate (i_t) and Input Modulation Gate ($\tilde{C}_t$):

The input gate controls how much new information to add to the cell state. The input modulation gate creates a vector of new candidate values ($\tilde{C}_t$), which are added to the cell state.

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$

- Output Gate (o_t):

- The output gate determines what information to output based on the cell state. It produces the hidden state for the next time step.

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$$

- Cell State Update:

The cell state is updated by combining the old cell state (modulated by the forget gate) and the new candidate values (modulated by the input gate).

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t$$

- Hidden State Update:

The hidden state is updated using the output gate and the updated cell state.

$$h_t = o_t \cdot \tanh(C_t)$$

These gates and states work together to control the flow of information and maintain the cell's ability to remember important information over long periods.

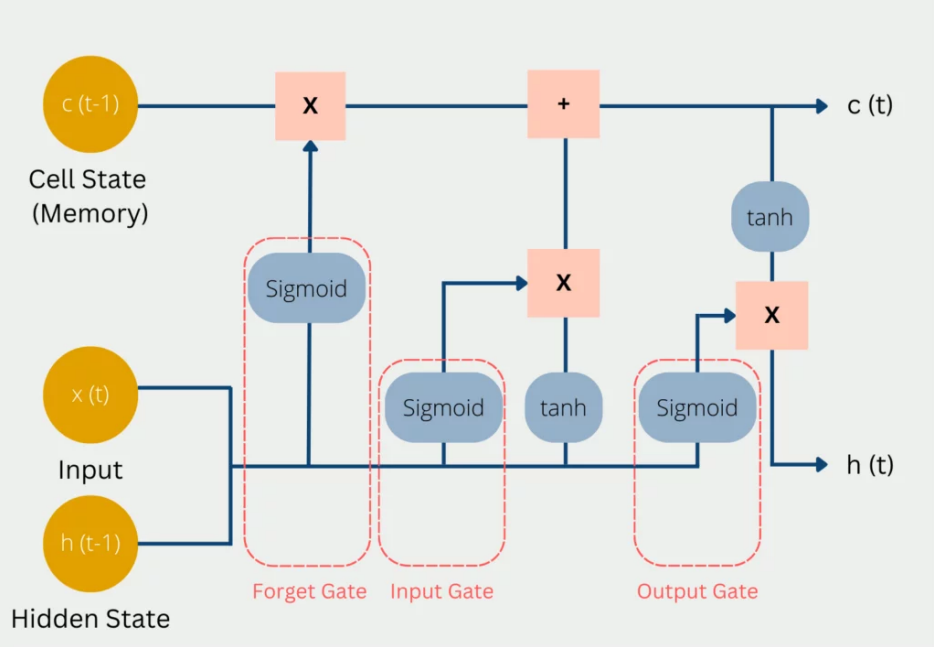

Figure 18: LSTM network

LSTM networks are specifically designed to remember information for long periods, making them effective at capturing long-term dependencies in sequence data.

Long Short-Term Memory (LSTM) networks are a powerful extension of traditional RNNs, designed to effectively model long-term dependencies in sequential data. By incorporating memory cells and gating mechanisms, LSTMs address the vanishing gradient problem and enable the learning of complex temporal patterns. Despite their computational complexity and the challenges associated with hyperparameter tuning, LSTMs are widely used in various applications, from time series forecasting to natural language processing, due to their robustness and ability to handle sequences of arbitrary length Medium (2023).

3 Research methodology and experiments of the thesis

This section outlines the methodology used to predict the next day closing price of the KRIKRI milk industry stock using various machine learning models. Four different experiments were conducted to evaluate the performance of these models based on their root mean square error (RMSE) and mean absolute percentage error (MAPE).

3.1 Data Collection and Preprocessing

The dataset used in this study includes historical stock prices of the KRIKRI milk industry. The data was collected from [source], spanning from [start date] to [end date]. The primary features used in the analysis were: - Open price - High price - Low price - Close price - Volume

Additionally, date and time features were extracted to incorporate temporal aspects into the models.

Software Used for the Research

The research methodology involved utilizing various software tools and libraries essential for data analysis, machine learning, and stock price prediction. The key tools and libraries used are:

- **Python 3.7**: The primary programming language.
- **Pandas**: Data manipulation and analysis.
- **NumPy**: Numerical computing.
- **scikit-learn (sklearn)**: Machine learning.
- **Matplotlib** and **Seaborn**: Data visualization.
- **yfinance**: Accessing financial data.
- **pmdarima**: Time series analysis.
- **pandas-ta**: Technical analysis.
- **TensorFlow-GPU**: Deep learning.
- **CUDA Toolkit** and **cuDNN**: GPU acceleration.
- **JupyterLab**: Interactive coding environment.
- **Visual Studio Code (VS Code)**: Integrated development environment.

Data Collection Process

For the research, historical stock data for KRIKRI was obtained using the `yfinance` library in Python. This library provides a convenient way to download financial data from Yahoo Finance, allowing access to a wide range of stock market information. The data collection involved fetching the daily Open, High, Low, Close prices, and trading Volume for KRIKRI. This dataset served as the foundation for performing feature engineering and developing predictive models for stock closing price.

3.2 Feature representation

In this case study, we utilize various financial indicators to predict the next day's closing price of a stock. The features include the open price, high price, low price, close price, and volume of trades for a given day. These features are used to forecast the target variable, which is the closing price of the stock on the following day.

Feature Name	Description
Open Price	The price at which the stock opens for trading on a particular day.
High Price	The highest price reached by the stock during the trading day.
Low Price	The lowest price reached by the stock during the trading day.
Close Price	The price at which the stock closes at the end of the trading day.
Volume	The number of shares traded during the day.
Target	The closing price of the stock on the next trading day.

Table 5: Features and Target Variable

This structure helps in understanding how different market variables influence future stock prices, thereby aiding in making informed investment decisions through predictive modeling in machine learning.

3.3 Definitions and Error Metrics

3.3.1 Stock

The term stock, which is often referred to as capital stock, refers to each of the equal pieces into which the capital of a corporation is divided. The price of a stock depends on the last transaction that took place, i.e., the last sale and purchase of a stock of the respective financial enterprise.

There are two main ways to buy and sell stocks: either from another shareholder who is interested in buying or selling the stocks they had acquired in the past, or directly from the company in the case of issuing stocks, a process primarily aimed at raising capital.

A multitude of reasons can lead to changes in the price of a stock, such as the company's investments, internal changes, and social interventions. A characteristic and recent example is the stock market crisis due to COVID-19 in 2020. The price of a stock, i.e., the monetary amount corresponding to one piece at a given time, is subject to continuous changes. For better monitoring of price developments, specific indices and their changes are studied within each 24-hour period.

3.3.2 Opening - Closing Prices

The most important indices on which future predictions can be based involve the price at which the stock stood at the opening of each day (Open) and the price at the closing of the day (Close or Adj. Close), which most models for predicting the course of each stock rely on.

It should be noted that the closing price is calculated from the average of the prices as they evolved in the last thirty minutes before the end of the trading process for each day.

3.3.3 High - Low Prices

In addition to the opening and closing prices, the high and low prices of a stock during a trading day are crucial for understanding its volatility and intraday price movements.

The high price (High) represents the maximum price at which the stock was traded during the day. This metric is significant as it indicates the highest level of buyer willingness to pay for the stock within that trading period.

The low price (Low), on the other hand, represents the minimum price at which the stock was traded during the day. This metric highlights the lowest level at which sellers were willing to sell the stock.

Both the high and low prices provide valuable insights into the stock's daily price range and can be used in various technical analysis methods to forecast future price movements. They help traders and analysts understand the market sentiment and the stock's volatility, which are essential factors for making informed investment decisions.

3.3.4 Last Price

The last price (Last) refers to the price at which the last transaction of the specific stock took place. Consequently, if transactions for a stock occurred in thirty minutes before the end of the trading day, the last price may not coincide with the closing price as described above.

3.3.5 Volume

The stock volume of a company refers to the number of shares that were exchanged within a specific time frame. It constitutes the total number of pieces that were exchanged during a specific duration of time.

3.4 Error and Accuracy Metrics

In order to provide a comprehensive presentation and comparison of the results of each method examined in this study, specific error metrics were selected for use.

The choice of metrics was made with the aim of allowing comparisons both between different forecasting methods and between the different stocks that will be used for comparison, as presented in the corresponding section, following the literature.

3.4.1 Root Mean Square Error

The Root Mean Square Error (RMSE) is an error metric in which the weight of the error is presented in the largest deviation of the forecast from the observation and not in its sign, which is achieved by raising the difference of the two values to the square.

The square root of the product of the reciprocal of the number of forecasts made and the sum of the squared errors gives a more understandable result as calculated by the following mathematical formula:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

where:

- n is the number of observations,
- y_i is the actual value,
- $\hat{y}_i$ is the predicted value.

3.4.2 Mean Absolute Percentage Error

The Mean Absolute Percentage Error (MAPE) is an error metric widely used in the literature because it offers easier understanding. MAPE is calculated as the product of the reciprocal of the number of forecasts and the sum of the absolute errors divided by the actual value to be forecasted.

Care must be taken when using MAPE if the data contains zero values and in the case of a small total number of observations \(N \); a small number of forecasts with large errors can significantly affect the MAPE, which is calculated by the following mathematical formula:

$$\text{MAPE} = \frac{100\%}{n} \sum_{i=1}^{n} \left| \frac{y_i - \hat{y}_i}{y_i} \right|$$

where:

- n is the number of observations,
- y_i is the actual value,
- $\hat{y}_i$ is the predicted value.

The MAPE expresses the error as a percentage, making it easier to interpret. Lower MAPE values indicate better model accuracy.

3.4.3 MAPE vs RMSE metrics, key differences

RMSE has the same units as the data, while MAPE is a percentage. RMSE is more sensitive to outliers due to the squaring of errors, whereas MAPE is less sensitive but can be skewed by small actual values. RMSE gives an absolute measure of prediction accuracy, whereas MAPE provides a relative measure. In practice, the choice between RMSE and MAPE depends on the context and the specific needs of the analysis. RMSE is often preferred when dealing with continuous data and when it's important to penalize larger errors more heavily. MAPE is useful for comparing predictive accuracy across different scales and for understanding relative error.

3.5 Data Cleaning

- Missing values were handled using forward filling. - Outliers were identified and treated using the IQR method. - Data was normalized to ensure uniform scaling.

3.6 Train - Test Split

In this study, the dataset was divided into a training set and a test set to evaluate the performance of the model. The training set comprised data from January 1, 2020, to April 30, 2024. This period was used to train the model, allowing it to learn the patterns and relationships within the data. The test set included data from May 1, 2024, to May 31, 2024. This separate time period was reserved to test the model's predictions and assess its generalizability to new, unseen data.

3.7 Feature Engineering

To enhance the predictive accuracy of my stock closing price model, I conducted extensive feature engineering using a range of technical indicators. These indicators include the Simple Moving Average (SMA) with a length of 2, and the Weighted Moving Average (WMA) with a length of 2, which smooth out price data to highlight trends. Additionally, I incorporated Momentum to measure the rate of price change, and the Stochastic Oscillator to determine the position of a closing price relative to its price range over a given period. The Relative Strength Index (RSI) was used to identify overbought or oversold conditions, while the Moving Average Convergence Divergence (MACD) helped identify changes in the strength, direction, momentum, and duration of a trend. Moreover, I included William's %R with a length of 7 to further assess overbought and oversold levels. The Accumulation/Distribution (A/D) Oscillator was utilized to measure the cumulative flow of money into and out of a security. Lastly, the Commodity Channel Index (CCI) was included to identify cyclical trends in the stock price. By integrating these diverse technical indicators, my model benefits from a comprehensive set of features that capture various market dynamics, ultimately improving the robustness and accuracy of stock closing price predictions.

3.8 Models and Techniques Used

- AutoRegressive Integrated Moving Average (ARIMA)

ARIMA is a statistical model used for time series forecasting, which combines autoregression, differencing, and moving average components.

- Random Forest (RF)

Random Forest is an ensemble learning method that constructs multiple decision trees during training and outputs the mean prediction of the individual trees to reduce overfitting.

- Extreme Gradient Boosting (XGBoost)

XGBoost is an optimized gradient boosting algorithm designed to be highly efficient, flexible, and portable.

- k-Nearest Neighbors (KNN)

KNN is a non-parametric method used for classification and regression, where the input consists of the k closest training examples in the feature space.

- Long Short-Term Memory (LSTM)

LSTM is a type of recurrent neural network (RNN) capable of learning long-term dependencies, particularly suitable for sequential data like time series.

3.9 Experimental Design

Experiment 1: Default Models In the first experiment, the following machine-learning models were used with their default hyperparameters: - ARIMA (1,1,1) - Random Forest (RF) - Extreme Gradient Boosting (XGBoost) - k-Nearest Neighbors (KNN) - Long Short-Term Memory (LSTM) - Tuned AutoRegressive Integrated Moving Average (ARIMA)

The performance of these models was evaluated based on RMSE and MAPE.

3.10 Experiment 2: Tuned Models

In the second experiment, hyperparameter tuning was conducted for each model using grid search or random search methods. The tuned models included: - ARIMA (1,1,1) - Random Forest (RF_grid) - XGBoost (XGB_grid) - k-Nearest Neighbors (KNN_grid) - Support Vector Machine (SVM_grid) - Long Short-Term Memory (LSTM_grid)

3.11 Experiment 3: Feature Engineering with Default Models

In the third experiment, the default models from Experiment 1 were retrained using the engineered features to evaluate the impact of feature engineering. The models that were trained are the following: - ARIMA (1,1,1) - Random Forest (RF_fe) - XGBoost (XGB_fe) - k-Nearest Neighbors (KNN_fe) - Support Vector Machine (SVM_fe) - Long Short-Term Memory (LSTM_fe)

where the abbrevation fe, means incorporating new engineered features.

3.12 Experiment 4: Feature Engineering with Tuned Models

In the fourth experiment, the tuned models from Experiment 2 were retrained using the engineered features to evaluate the combined effect of hyperparameter tuning and feature engineering.

3.13 Hyperparameter Tuning

Hyperparameter tuning was performed using grid search or random search to find the optimal parameters for each model. Key parameters tuned included: - ARIMA (1,1,1) - For RF: Number of trees, maximum depth, minimum samples split - For XGBoost: Learning rate, maximum depth, number of estimators - For KNN: Number of neighbors, distance metric - For SVM: Kernel type, regularization parameter - For LSTM: Number of layers, number of units per layer, learning rate

3.14 Evaluation Metrics

The performance of each model was evaluated using two metrics: - Root Mean Square Error (RMSE):

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

where y_i is the actual value and $\hat{y}_i$ is the predicted value. - Mean Absolute Percentage Error (MAPE):

$$MAPE = \frac{100\%}{n} \sum_{i=1}^{n} \left| \frac{y_i - \hat{y}_i}{y_i} \right|$$

3.15 Results and Discussion

The results of the experiments are summarized below:

Experiment 1: Default Models

Model	RMSE	MAPE
ARIMA	0.150457	1.057122
Random Forest (RF)	0.137974	1.040942
XGBoost (XGB)	0.200465	1.443823
Support Vector Machine (SVM)	0.144565	1.075336
k-Nearest Neighbors (KNN)	0.181033	1.419206
Long Short-Term Memory (LSTM)	0.220316	1.322311

Experiment 2: Tuned Models

Model	RMSE	MAPE
ARIMA	0.150457	1.057122
Random Forest (RF_grid)	0.130005	1.075838
XGBoost (XGB_grid)	0.174309	1.241328
k-Nearest Neighbors (KNN_grid)	0.181033	1.419206
Support Vector Machine (SVM_grid)	0.144565	1.075336
Long Short-Term Memory (LSTM_grid)	0.136895	1.347298

Experiment 3: Feature Engineering with Default Models

Model	RMSE	MAPE
ARIMA	0.150457	1.057122
Random Forest (RF_fe)	0.140336	0.978169
XGBoost (XGB_fe)	0.185596	1.241250
Support Vector Machine (SVM_fe)	0.758735	6.516719
k-Nearest Neighbors (KNN_fe)	0.298057	2.293710
Long Short-Term Memory (LSTM_fe)	0.363042	3.101163

Experiment 4: Feature Engineering with Tuned Models

Model	RMSE	MAPE
ARIMA	0.150457	1.057122
Random Forest (RF_grid_fe)	0.146349	1.041161
XGBoost (XGB_grid_fe)	0.182745	1.313748
k-Nearest Neighbors (KNN_grid_fe)	0.298057	2.293710
Support Vector Machine (SVM_grid_fe)	0.758735	6.516719
Long Short-Term Memory (LSTM_grid_fe)	0.267982	1.137688

3.16 Discussion

The experiments revealed several insights: The Random Forest model, both default and tuned, consistently performed well across all experiments, indicating its robustness in stock price prediction. Feature engineering significantly improved the performance of some models but had a detrimental effect on others, such as SVM and KNN. Hyperparameter tuning generally improved the model performance, as observed in the RF_grid and LSTM_grid results. LSTM models, while powerful, require careful tuning and feature selection to outperform simpler models like Random Forest.

3.17 Conclusion

This study demonstrates the effectiveness of various machine learning models in predicting the next day closing price of KRIKRI milk industry stock. Random Forest emerged as the most reliable model, particularly when tuned and combined with feature engineering. Future work could explore additional feature engineering techniques, alternative model architectures, and the integration of external data sources to further enhance prediction accuracy.

This methodology section provides a comprehensive overview of the experiments conducted, the models used, the data preprocessing steps, and the results obtained. Adjust and expand the content as needed to meet the specific requirements of your thesis and to reach the desired word count.

4 Financial Dashboard

The financial dashboard created for this thesis offers a comprehensive tool for analyzing stock market data, integrated into a web-based platform accessible at FinanceVue (https://financevue.streamlit.app/). The dashboard is divided into five key sections, each serving a unique purpose in financial analysis and forecasting.

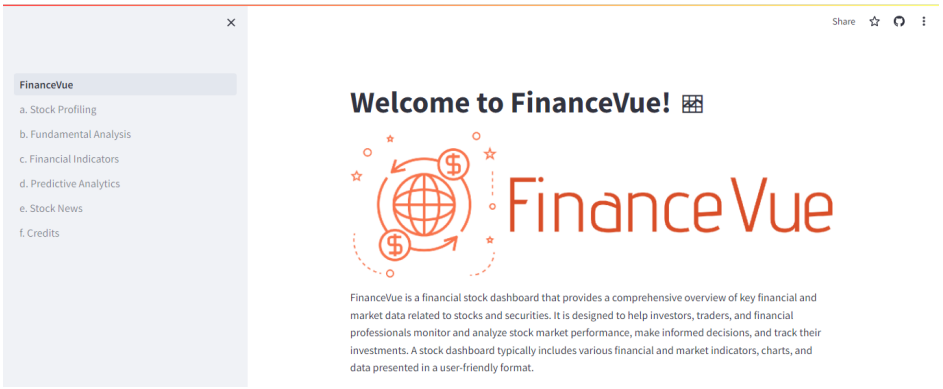

Figure 19: FinanceVue dashboard

4.1 Page 1 - Examining the Stock Historical Data

This section of the financial dashboard provides a detailed examination of a stock's historical data. It includes essential company information such as industry, address, and website. Users can visualize the stock's closing price history over time through various charts and plots. The tools available include candlestick plots, which show daily price movements; annual evolution charts, displaying year-by-year trends; histograms of the most common stock prices; and violin plots, which combine box plots with kernel density estimation to show the distribution of prices.

Additionally, this section offers scatterplots and candlestick plots for open, high, low prices, and trading volume over time. These visualizations help users understand how these variables fluctuate and interact. The candlestick plots are particularly useful for identifying patterns and trends in the stock's price movements.

Furthermore, the page includes bivariate analysis for open, high, low, close, adjusted close prices, and volume using Kernel Density Estimation (KDE) plots, scatterplots, and correlation coefficients. These tools allow users to explore the relationships between different price metrics and trading volume, providing insights into how various factors influence stock price behavior. By examining these relationships, users can make more informed decisions about potential future movements of the stock.

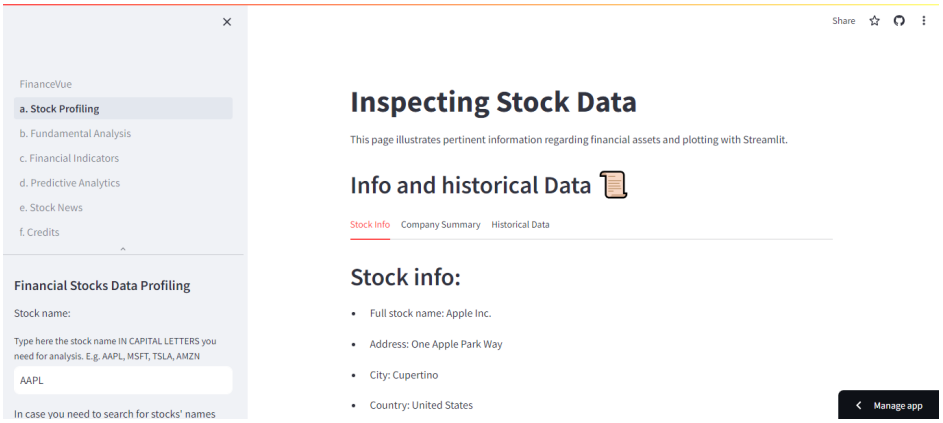

Figure 20: Stock Profiling Page

4.2 Page 2 - Fundamental Analysis

The second section, "Fundamental Analysis," focuses on the intrinsic value of stocks by examining key financial metrics and company fundamentals. Users can access detailed information on financial statements, including balance sheets, income statements, and cash flow statements. This page provides ratios and metrics such as the price-to-earnings ratio (P/E), earnings per share (EPS), and return on equity (ROE), which are essential for evaluating a company's financial health and growth potential.

By integrating these fundamental metrics, the dashboard helps users assess whether a stock is undervalued or overvalued relative to its true worth. This analysis is vital for long-term investors who base their decisions on the fundamental strength of companies rather than short-term market movements. The inclusion of fundamental analysis tools allows users to perform a thorough evaluation of stocks, supporting more strategic investment decisions.

The fundamental analysis indexes and indicators used are the following:

Category	**Ratio**	**Formula**
Profitability Ratios	Earnings Per Share (EPS)	$\text{EPS} = \frac{\text{Net Income} - \text{Preferred Dividends}}{\text{Average Outstanding Shares}}$
	Price-to-Earnings (P/E) Ratio	$\text{P/E Ratio} = \frac{\text{Market Value per Share}}{\text{EPS}}$
	Return on Equity (ROE)	$\text{ROE} = \frac{\text{Net Income}}{\text{Shareholders' Equity}}$
	Return on Assets (ROA)	$\text{ROA} = \frac{\text{Net Income}}{\text{Total Assets}}$
	Gross Profit Margin	$\text{Gross Profit Margin} = \frac{\text{Gross Profit}}{\text{Revenue}} \times 100$
	Operating Margin	$\text{Operating Margin} = \frac{\text{Operating Income}}{\text{Revenue}} \times 100$
Liquidity Ratios	Current Ratio	$\text{Current Ratio} = \frac{\text{Current Assets}}{\text{Current Liabilities}}$
	Quick Ratio	$\text{Quick Ratio} = \frac{\text{Current Assets} - \text{Inventories}}{\text{Current Liabilities}}$
Leverage Ratios	Debt-to-Equity Ratio	$\text{Debt-to-Equity Ratio} = \frac{\text{Total Liabilities}}{\text{Shareholders' Equity}}$
	Interest Coverage Ratio	$\text{Interest Coverage Ratio} = \frac{\text{EBIT}}{\text{Interest Expense}}$
Efficiency Ratios	Asset Turnover Ratio	$\text{Asset Turnover Ratio} = \frac{\text{Net Sales}}{\text{Total Assets}}$
	Inventory Turnover Ratio	$\text{Inventory Turnover Ratio} = \frac{\text{COGS}}{\text{Average Inventory}}$
Valuation Ratios	Price-to-Book (P/B) Ratio	$\text{P/B Ratio} = \frac{\text{Market Price per Share}}{\text{Book Value per Share}}$
	Dividend Yield	$\text{Dividend Yield} = \frac{\text{Annual Dividends per Share}}{\text{Price per Share}} \times 100$
Growth Ratios	Revenue Growth Rate	$\text{Revenue Growth Rate} = \frac{\text{Current Year Revenue} - \text{Previous Year Reve}}{\text{Previous Year Revenue}}$
	Earnings Growth Rate	$\text{Earnings Growth Rate} = \frac{\text{Current Year Earnings} - \text{Previous Year Earn}}{\text{Previous Year Earnings}}$
Cash Flow Ratios	Free Cash Flow (FCF)	$\text{FCF} = \text{Operating Cash Flow} - \text{Capital Expenditures}$
	Operating Cash Flow Ratio	$\text{Operating Cash Flow Ratio} = \frac{\text{Operating Cash Flow}}{\text{Current Liabilities}}$

Table 6: Financial Ratios

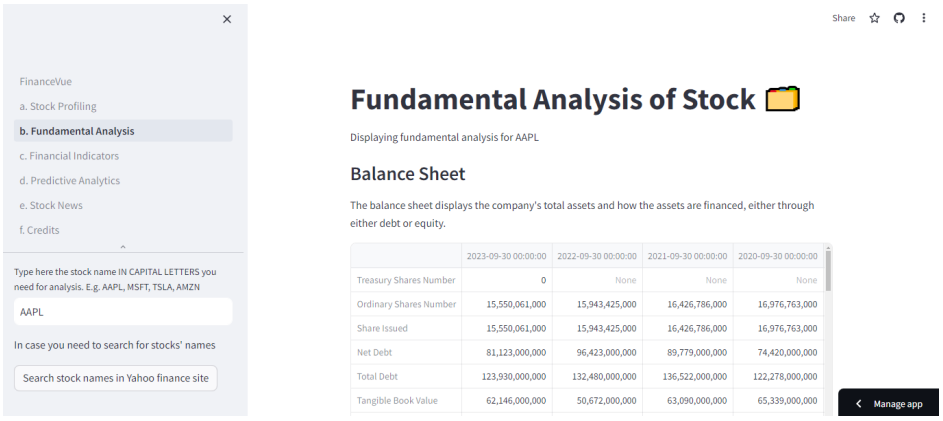

Figure 21: Fundamental Analysis page

4.3 Page 3 - Technical Analysis

The ”Technical Analysis” section provides tools for analyzing price patterns and market trends using various technical indicators. This page includes popular indicators such as moving averages, relative strength index (RSI), moving average convergence divergence (MACD), and Bollinger Bands. These tools help users identify trends, momentum, and potential reversal points in stock prices, which are crucial for making timely buy or sell decisions.

Interactive charts in this section allow users to overlay multiple technical indicators on stock price charts, offering a holistic view of market conditions. This detailed technical analysis enables users to spot trading opportunities and manage risks effectively. By combining multiple indicators, users can gain a more comprehensive understanding of market dynamics and enhance their trading strategies.

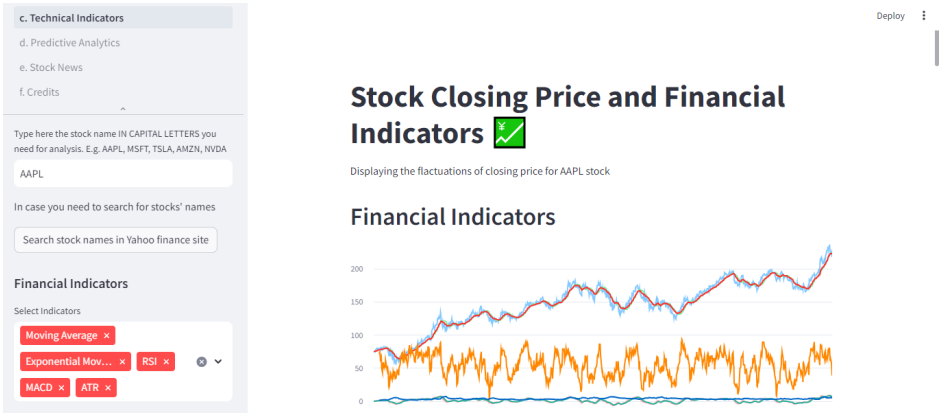

Figure 22: Technical Indicators page

4.4 Page 4 - Forecasting Future Prices

In the ”Forecasting Future Prices” section, the dashboard leverages advanced machine learning models to predict future stock prices. This page provides users with predictive analytics based on historical data, incorporating models such as Support Vector Regression (SVR) and XGBoost. These predictions include visualizations of expected price movements and confidence intervals, helping users make informed predictions about future stock performance.

The forecasting tools in this section are designed to be user-friendly, offering clear and concise predictions that can be easily interpreted. By providing accurate and reliable forecasts, the dashboard empowers users to make proactive decisions and optimize their investment strategies. This section highlights the practical application of advanced analytics in financial forecasting.

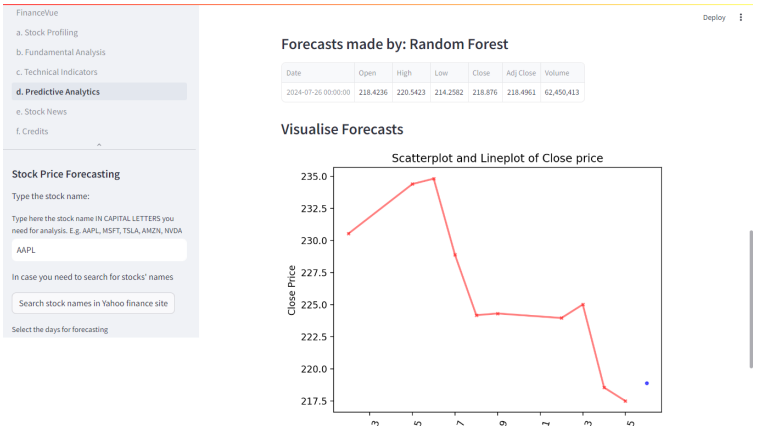

Figure 23: Forecasting stock prices page

4.5 Page 5 - Stock News

The final section, ”Stock News,” aggregates the latest news and updates related to stocks from various financial news sources. This page provides real-time news feeds, ensuring that users stay informed about the latest developments that might impact stock prices. By offering a centralized location for relevant news, the dashboard helps users keep track of important events and trends that could influence their investment decisions.

In addition to news feeds, this section may include sentiment analysis, which evaluates the overall market sentiment based on news articles. This analysis helps users gauge the market mood and potential reactions to recent events. By combining news updates with sentiment analysis, the dashboard offers a comprehensive view of the market, supporting more informed and timely investment decisions.

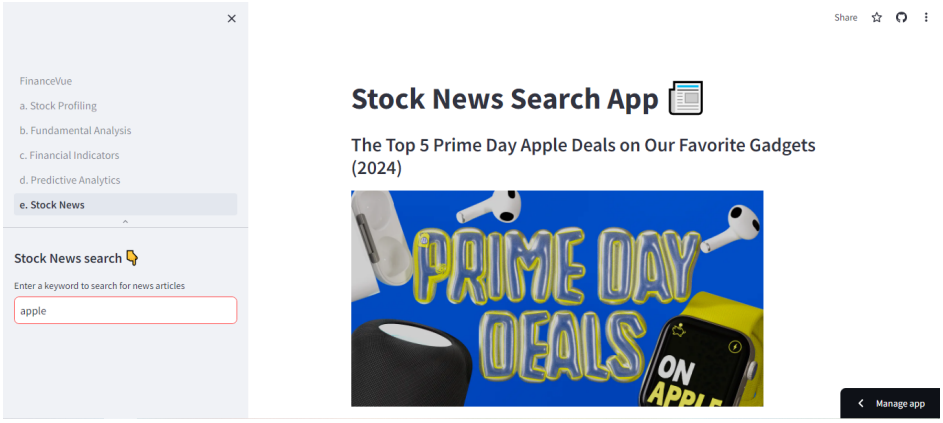

Figure 24: Stock News Page

5 Conclusions - Suggestions for future research

Future research can investigate the application of other advanced machine learning techniques, such as Transformer models and Generative Adversarial Networks (GANs), to further enhance the accuracy of short-term stock price predictions Selvin et al. (2017).

Expanding the dataset to include alternative data sources such as social media sentiment, news articles, economic indicators, and macroeconomic variables can provide a more comprehensive view of market dynamics and improve predictive performance Gite et al. (2021).

Developing adaptive models that can update themselves in real-time based on incoming data can help maintain accuracy in the face of rapidly changing market conditions. Techniques such as online learning and reinforcement learning could be explored for this purpose Agrawal et al. (2019).

Further work can be directed towards sophisticated feature engineering techniques, including the creation of new technical indicators and the application of feature selection algorithms, to improve the input quality for predictive models.

Another area for future research is improving the interpretability of machine learning models. This could involve developing methods to better understand the decision-making processes of complex models such as deep learning networks, making them more transparent and trustworthy for financial analysts and investors Nelson et al. (2017b).

Future research should also consider the ethical and regulatory implications of using machine learning for stock price prediction, including the potential for market manipulation and the need for transparency in algorithmic trading practices Vellaiparambill & Natchimuthu (2022).

List of Figures

List of Tables

References

2023. Understanding stock market behavior: A multifactorial analysis. *Finance Review* 12.45–67. URL https://financejournal.com/article123.

ACHELIS, STEVEN B. 2001. *Technical analysis from a to z*. McGraw-Hill. Accessed: 2024-07-17. URL https://www.bettertraderacademy.com/resources/Technical-Analysis-from-A-to-Z.pdf.

AGRAWAL, MANISH; ASIF KHAN; und PIYUSH SHUKLA. 2019. Stock indices price prediction based on technical indicators using deep learning model. *International Journal on Emerging Technologies* 10.

AMIT, YALI, und DONALD GEMAN. 1997. Shape Quantization and Recognition with Randomized Trees. *Neural Computation* 9.1545–1588. URL https://doi.org/10.1162/neco.1997.9.7.1545.

ANYCHART. technical indicators and their formulas. URL https://docs.anychart.com/Stock_Charts/Technical_Indicators/Mathematical_Description.

ARPAL, SHREYAS. 2020. Time series forecasting. URL https://medium.com/@shreyasarpal26/time-series-forecasting-55771b2fa401.

BAJAJFINSERV. Fundamental analysis. URL https://www.bajajfinserv.in/fundamental-analysis.

BULKOWSKI, THOMAS N. 2020. *Chart patterns*. self-published. Accessed: 2024-07-17. URL https://thepatternsite.com/CCP.pdf.

DANIEL, FABRICE. 2019. Financial time series data processing for machine learning. URL https://arxiv.org/pdf/1907.03010.

FROST, JIM. 2023. Autocorrelation and partial autocorrelation in time series data. URL https://statisticsbyjim.com/time-series/autocorrelation-partial-autocorrelation/.

GITE, SHILPA; HRITUJA KHATAVKAR; KETAN KOTECHA; SHILPI SRIVASTAVA; PRIYAM MAHESHWARI; und NEERAV PANDEY. 2021. Explainable stock prices prediction from financial news articles using sentiment analysis. *PeerJ Computer Science* 7.e340.

HOCHREITER, SEPP, und JÜRGEN SCHMIDHUBER. 1997. Long short-term memory. *Neural computation* 9.1735–80.

HYNDMAN, ROB J, und GEORGE ATHANASOPOULOS. 2018. *Forecasting: Principles and practice*. 2nd Ed. OTexts. URL https://otexts.com/fpp2/.

IBM. a. What is ai? URL https://www.ibm.com/topics/artificial-intelligence.

IBM. b. What is ml? URL https://www.ibm.com/topics/machine-learning.

INVESTOPEDIA. a. How do i take qualitative factors into consideration when using fundamental analysis? URL https://www.investopedia.com/ask/answers/qualitative-factors-when-using-fundamental-analysis/.

INVESTOPEDIA. b. Technical analysis. URL https://www.investopedia.com/terms/t/technicalanalysis.asp.

INVESTOPEDIA. c. Volatility: Meaning in finance and how it works with stocks. URL https://www.investopedia.com/terms/v/volatility.asp.

INVESTOPEDIA. d. What is momentum? definition in trading, tools, and risks. URL https://www.investopedia.com/terms/m/momentum.asp.

JOSEPH, ELIJAH. 2019. Forecast on close stock market prediction using support vector machine (svm). *International Journal of Engineering Research and* V8.

KARMIANI, DIVIT; RUMAN KAZI; AMEYA NAMBISAN; AASTHA SHAH; und VIJAYA KAMBLE. 2019. Comparison of predictive algorithms: Backpropagation, svm, lstm and kalman filter for stock market. 228–234.

MADGE, SAAHIL, und SWATI BHATT. 2015. Predicting stock price direction using support vector machines. URL https://api.semanticscholar.org/CorpusID:43966173.

MEDIUM. 2023. Understanding lstm: Architecture, pros and cons, and implementation. URL https://medium.com/@anishnama20/understanding-lstm-architecture-pros-and-cons-and-implementation-3e0cca194094.

NELSON, DAVID; ADRIANO PEREIRA; und RENATO DE OLIVEIRA. 2017a. Stock market's price movement prediction with lstm neural networks. 1419–1426.

NELSON, DAVID; ADRIANO PEREIRA; und RENATO DE OLIVEIRA. 2017b. Stock market's price movement prediction with lstm neural networks. 1419–1426.

RESEARCHER, FINANCIAL. 2022. Machine learning in financial forecasting: A u.s. review: Exploring the advancements, challenges, and implications of ai-driven predictions in financial markets. *Journal of Finance and Machine Learning* 5.89–112. URL https://www.researchgate.net/publication/378567755_Machine_learning_in_financial_forecasting_A_US_review_Exploring_the_advancements_challenges_and_implications_of_AI-driven_predictions_in_financial_markets#:~:text=Key%20findings%20indicate%20that%20AI,reinforcement%20learning%2C%20and%20hybrid%20models.

SCIENCE, TOWARDS DATA. 2019. Stationarity in time series analysis. URL https://towardsdatascience.com/stationarity-in-time-series-analysis-90c94f27322.

SELVIN, SREELEKSHMY; VINAYAKUMAR RAVI; E. A GOPALAKRISHNAN; VIJAY MENON; und SOMAN KP. 2017. Stock price prediction using lstm, rnn and cnn-sliding window model. 1643–1647.

STATISTICIAN, FINANCIAL. 2020. Correlation analysis of company characteristics and stock price predictions. *Financial Analytics* 15.123–145. URL https://fianalytics.com/article012.

STRIKE.MONEY. a. Technical analysis, support-resistance. URL https://www.strike.money/technical-analysis#Support_Resistance.

STRIKE.MONEY. b. Technical analysis, trends. URL https://www.strike.money/technical-analysis#Trends.

STRIKE.MONEY. c. Technical analysis, volume. URL https://www.strike.money/technical-analysis#Volume.

TEAM, EDUCATIONAL WAVE. Pros and cons of time series analysis. URL https://www.educationalwave.com/pros-and-cons-of-time-series-analysis/.

VARSAMOPOULOS, SAVVAS; KOEN BERTELS; und CARMEN ALMUDEVER. 2018. Designing neural network based decoders for surface codes.

VELLAIPARAMBILL, ALAN GEORGE, und NATCHIMUTHU NATCHIMUTHU. 2022. Ethical tenets of stock price prediction using machine learning techniques: A sustainable approach. *ECS Transactions* 107.137–149.

WAFI, AHMED. S.; HASSAN HASSAN; und ADEL MABROUK. 2015. Fundamental analysis models in financial markets – review study. *Procedia Economics and Finance* 30.939–947, IISES 3rd and 4th Economics and Finance Conference. URL https://www.sciencedirect.com/science/article/pii/S2212567115013441.

WU, XINDONG; VIPIN KUMAR; J ROSS QUINLAN; JOYDEEP GHOSH; QIANG YANG; HIROSHI MOTODA; GEOFFREY J MCLACHLAN; ANGUS NG; BING LIU; PHILIP S YU; ZHI-HUA ZHOU; MICHAEL STEINBACH; DAVID J HAND; und DAN STEINBERG. 2008. Top 10 algorithms in data mining. *Knowledge and Information Systems* 14.1–37.

ZHU, YONGQIONG. 2020. Stock price prediction using the rnn model. *Journal of Physics: Conference Series* 1650.032103.