BATANGAS STATE UNIVERSITY
The National Engineering University
Alangilan Campus

STOCK PRICE PREDICTION

Submitted by:

REYNALYN I. ASILO
Master of Science in Data Science
(23-01071)

Submitted to:

Ms. SANDHYA AVASTHI

May 2024

ABSTRACT

Stock price prediction is a critical area of research in financial markets, with profound implications for investors, traders, and policymakers. Anticipating the direction of stock markets can function as an advanced recommendation system for short-term investors and as an early alert mechanism for long-term shareholders regarding financial distress. The paramount consideration in selecting any forecasting method resides in forecasting accuracy. Over the past decade, research endeavors aimed at enhancing the precision of forecasting models have seen a notable increase. The task of identifying suitable stocks for investment presents a considerable challenge. Each investor's primary objective is to maximize profits while minimizing risks on their investments. This project endeavors to employ various methods and algorithms for forecasting stock prices, including ARIMA, K-Nearest Neighbours regression, and Prophet. Subsequently, the accuracies of these models are compared to ascertain the optimal approach. Utilizing Amazon Stock Price data spanning five years obtained from the Yahoo Finance API, the study aims to delve into a thorough analysis of stock prices and their predictive capabilities for future research initiatives. The project serves as a cornerstone for democratizing machine learning technologies, particularly in facilitating broader access to investment opportunities for the general public.

Table of Contents

List of Figures

List of Table

Chapter 1

Introduction

1.1 Overview

Stock prices represent the market value of a company's shares, reflecting investors' perceptions of its current and future performance. These prices are determined by supply and demand dynamics in the stock market, influenced by a multitude of factors including company earnings, industry trends, economic indicators, investor sentiment, and geopolitical events.

Investors, traders, and financial analysts need to understand stock prices because they offer important information about the state and future prospects of both specific companies and the market as a whole. Investment choices, portfolio management techniques, and market stability are all impacted by stock prices, which operate as a gauge of investor confidence and sentiment.

The technique of projecting future changes in stock prices based on market trends, historical data, and other pertinent variables is known as stock price prediction. Accurately predicting stock prices is difficult because financial markets are dynamic and complex. For investors looking to take advantage of market opportunities, control risks, and maximize investment returns, it is also a vital undertaking.

A wide range of approaches and strategies, from sophisticated machine learning algorithms to conventional statistical models, are used to predict stock prices. To determine a company's inherent value and development potential, fundamental analysis examines its financial statements, industry position, and growth prospects. Contrarily, technical analysis concentrates on examining historical price changes and trading patterns in order to spot trends and patterns that might portend future price changes.

In recent years, advancements in technology and data analytics have led to the proliferation of machine learning and artificial intelligence techniques for stock price prediction. Large volumes of historical market data are utilized by these sophisticated algorithms to find patterns, correlations, and anomalies that may escape the notice of conventional analysis techniques.

Despite the advancements in predictive modeling techniques, stock price prediction remains inherently uncertain due to the unpredictable nature of financial markets. Factors such as market sentiment, investor behavior, and unexpected events can influence stock prices in unpredictable ways, making accurate prediction challenging.

In conclusion, stock prices play a central role in financial markets, serving as a key determinant of investment decisions, market sentiment, and economic activity. Stock price prediction, while challenging, is a vital endeavor for investors seeking to navigate the complexities of financial markets and make informed investment decisions. Through the use of advanced analytical techniques and comprehensive market analysis, investors can enhance their ability to forecast stock prices and achieve their investment objectives.

Chapter 2

Methodologies

2.1 ARIMA Modeling

When working with time series data, a statistical analysis method called an autoregressive integrated moving average, or ARIMA, is used to forecast future trends or acquire deeper insights into the dataset. If a model uses historical data analysis to predict future values, it is said to be autoregressive. An ARIMA model might, for example, use historical performance data to estimate future stock prices or use historical data to project future profitability for a corporation.

In ARIMA, every element serves as a parameter with a standard notation. In order to identify the type of ARIMA model being used, a conventional notation for ARIMA models would be ARIMA with p, d, and q, where integer values are used in place of the parameters. One definition of the parameters is:

p: the lag order, or number of lag observations in the model.
d: the degree of differencing, or the number of times the raw observations are differed.
q: the moving average window size, sometimes referred to as the moving average order.

ARIMA, a method utilized for forecasting future outcomes from historical time series, relies on the statistical principle of serial correlation, wherein prior data points impact subsequent ones.[1] Functioning as a forecasting tool, the ARIMA model extrapolates future behavior based on past performance. Particularly employed in technical analysis, it aids in predicting an asset's future trajectory.

2.2 Prophet Forecasting

Prophet models time patterns using a somewhat straightforward but powerful methodology.[2] By incorporating factors such as seasonality, trend, and holiday effects, Prophet outperforms traditional forecasting algorithms in identifying the complex patterns present in time series data.

Prophet benefits from:

Simple syntax makes Prophet easy to use for people with varying degrees of data analysis experience.

Resilience to Missing Values: Capable of effectively managing time series missing values.

Effective Management of Outliers: Prophet is capable of managing data with extreme variations or outliers.

As a result, Prophet not only makes future predictions based on historical data, but it also makes typical data analysis issues simple to apply and handle.

2.3 KNN Regression Time Series Forecasting

Traditionally, time series forecasting relied on statistical methods like ARIMA models or exponential smoothing. However, recent decades have seen the emergence of computational intelligence techniques for this purpose. While artificial neural networks stand out as the primary machine learning method for time series forecasting, alternative approaches like Gaussian Process or KNN have also found application. KNN, a widely-used algorithm in classification and regression tasks, operates by storing a set of examples, each comprising a feature vector and its corresponding class (for classification) or numerical value (for prediction). Based on a distance measure like Euclidean distance, KNN finds its k nearest neighbors—the instances that are most similar to a given example—when it is given a new example. Next, based on its closest neighbors, it predicts the class as the majority class or, in the case of regression, as an accumulation of the target values connected to these neighbors. The tsfknn R package, intended for univariate time series forecasting using KNN regression, is presented in this paper.[3]

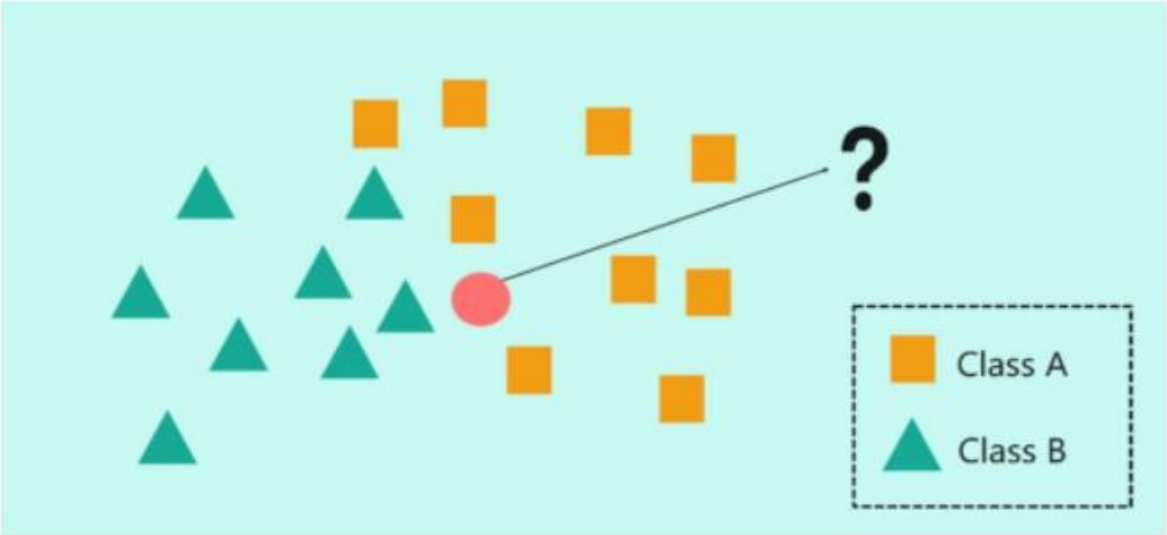

Figure 2.1: KNN model

Chapter 3

Analysis and Results

3.1 Initial Analysis

3.1.1 Visualization

The figure 3.1 shows the continuous line graph of the Amazon's Stock Price Trend. We see a sharp rise in the graph after July 2021 which indicates that the company blossomed after July 2021, since its share price increased.

Figure 3.1: AMAZON's Stock Price Trend 2019-2024

3.1.2 ACF-PACF Plots

Autocorrelation measures the correlation between a time series and its past values. In autoregressive (AR) models, the autocorrelation function (ACF) typically exhibits exponential damping. The ACF plot illustrates the correlation between data points up to a specified lag. While significant autocorrelations may be observed across multiple lags, it's possible that correlations at later lags are primarily influenced by those at earlier lags. To determine the order (p) of the AR model, we analyze the partial autocorrelation function (PACF) plot. Conversely, for moving average (MA) models, the ACF plot is utilized to identify the order (q), while the PACF plot demonstrates exponential damping.

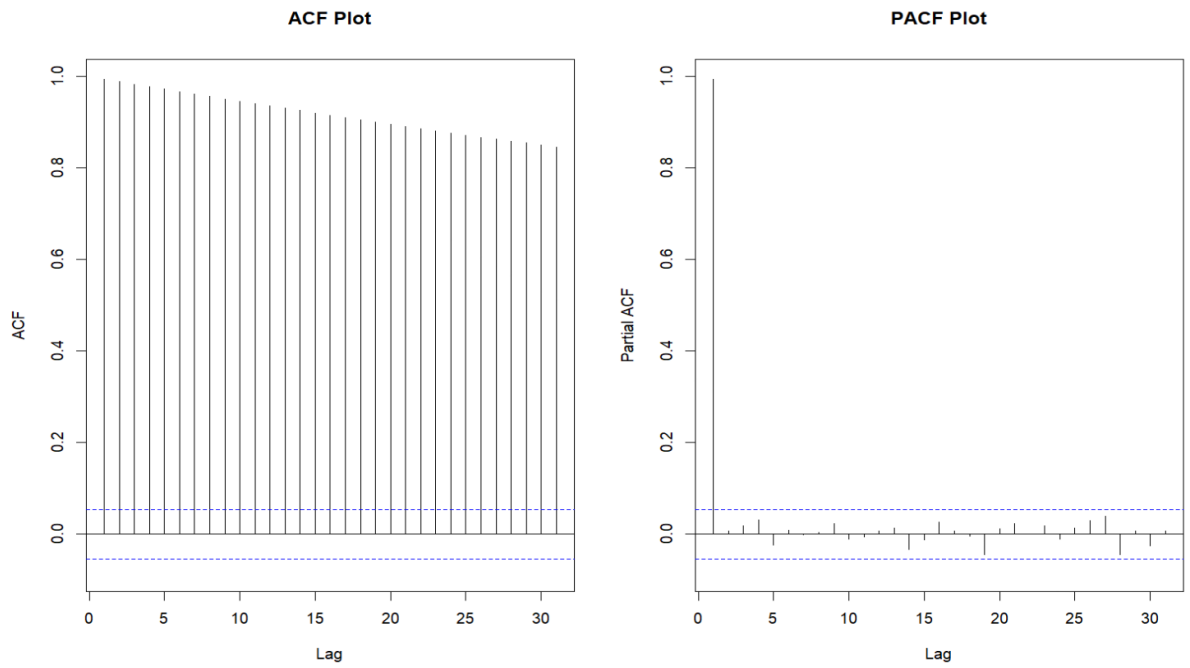

Figure 3.2: ACF-PACF Plots

3.1.3 Additive and Multiplicative Decomposition

A time series can exhibit either additive or multiplicative components. The multiplicative model is favored when the seasonal pattern's magnitude varies in proportion to the data values, increasing or decreasing along with them. Conversely, the additive model is preferred when the magnitude of the seasonal pattern remains consistent and does not vary with the data values.

The Additive Model : $T_t + S_t + X_t + Z_t$

The Multiplicative Model : $T_t * S_t * X_t * Z_t$

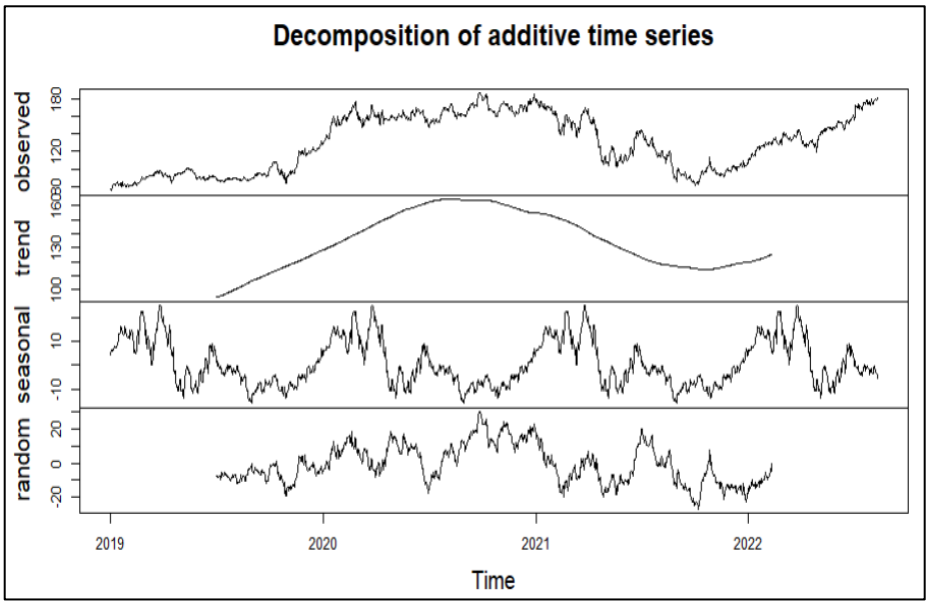

Figure 3.3: Additive Decomposition of Time Series

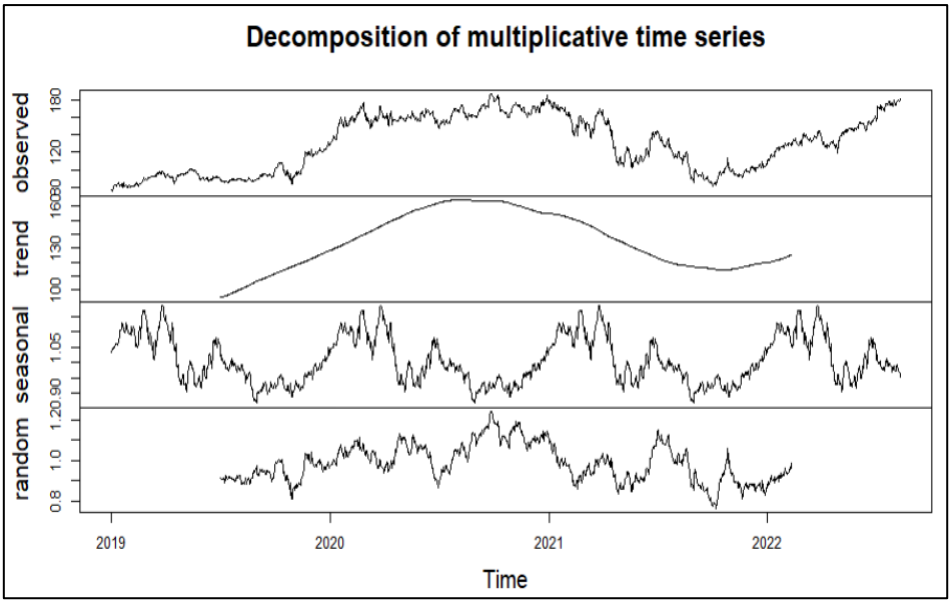

Figure 3.4: Decomposition of Time Series

3.1.4 Augmented Dickey Fuller Test

The Augmented Dickey Fuller Test (ADF) serves as a unit root test to assess stationarity. Unit roots may lead to unpredictable outcomes in time series analysis. The ADF test is effective in detecting serial correlation and is capable of handling more intricate models compared to the Dickey-Fuller test, making it a more powerful tool. The hypothesis for the test:

- The existence of a unit root is the test's null hypothesis. (The data's non-stationarity is the null hypothesis.)
- The alternative hypothesis varies little depending on the equation you apply. The time series being stationary is the fundamental alternative.

We perform the ADF test on our dataset and observe our p-value to be very high which denotes non-stationarity, so to make it stationary we perform a log transformation.

```
> print(adf.test(AMZN_CP))

        Augmented Dickey-Fuller Test

data:  AMZN_CP
Dickey-Fuller = -1.5671, Lag order = 10, p-value = 0.7616
alternative hypothesis: stationary
```

Figure 3.5: ADF before Log Transformation

```
> print(adf.test(logs))

        Augmented Dickey-Fuller Test

data:  logs
Dickey-Fuller = -11.335, Lag order = 10, p-value = 0.01
alternative hypothesis: stationary
```

Figure 3.6: ADF after Log Transformation

The p values should be made as low as possible. It is clearly observable that after taking the log of the dataset, the p value becomes significantly less (=0.01), hence denoting a stationary time series.

3.1.5 Log Transformation

As observed in the earlier ADF test, our dataset exhibits non-stationarity. To rectify this, we applied a log transformation, which effectively eliminates the non-stationary component. The following is graph 3.7 illustrating the plot of log returns.

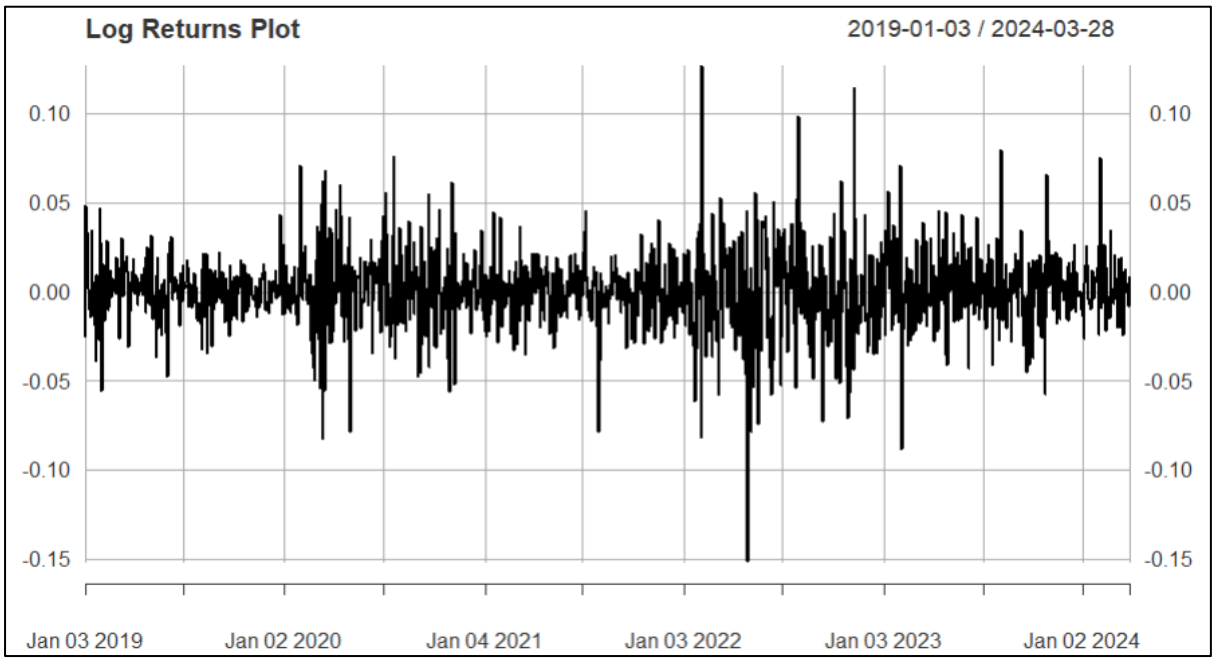

Figure 3.7: Log Returns Plot of Amazon Closing Price

3.1.6 Histogram and Empirical Distribution

Due to the substantial size of the five-year dataset, a histogram proves to be the most suitable tool. It aids in assessing the approximate normal distribution of the data. A bell-shaped histogram suggests a normal distribution pattern. The histogram representing the closing

prices of Amazon stock exhibits a unimodal shape, with symmetry around the mean, and conforms to a bell-shaped form, indicative of a normal distribution.

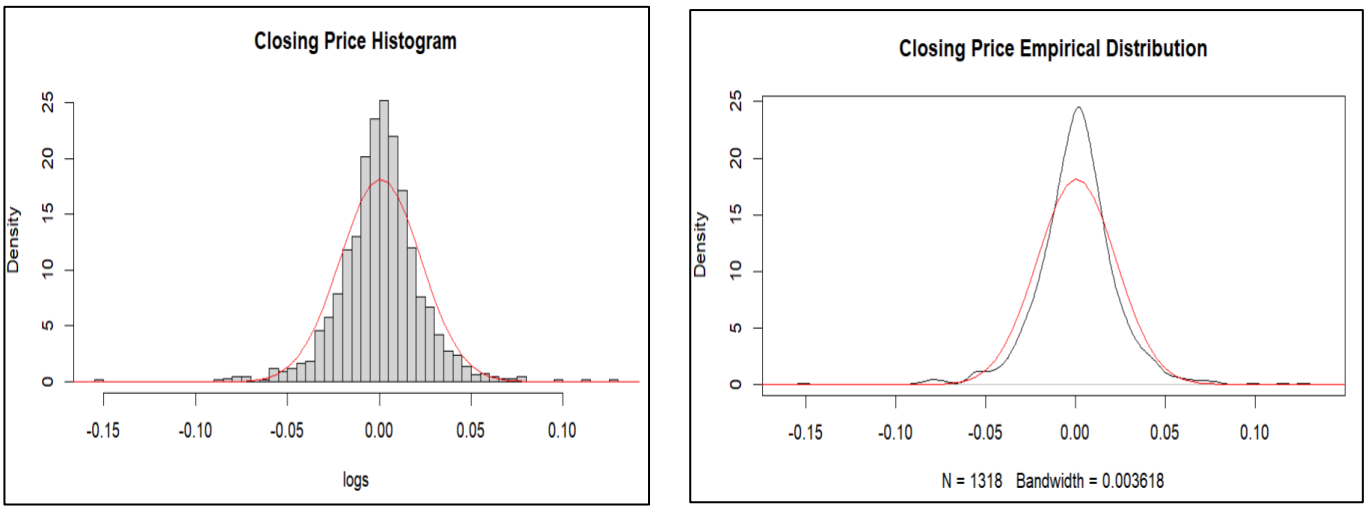

Figure 3.8: Histogram and Empirical Distribution

3.2 ARIMA Modeling

We utilized the AUTO-ARIMA function to determine the optimal approach for our dataset, bypassing the need for an in-depth analysis to identify model parameters. Our subsequent task involves forecasting future stock prices. We've conducted the forecasting using this model for the upcoming 30 days. The figure below demonstrates the accuracy of the ARIMA(1,1,1).

```
> # Auto ARIMA with seasonal = FALSE
> fit1<-auto.arima(AMZN_CP, seasonal=FALSE)
> tsdisplay(residuals(fit1), lag.max = 40, main='(1,1,1) Model Residuals')
> fcast1<-forecast(fit1, h=30)
> plot(fcast1)
> accuracy(fcast1)
                      ME      RMSE       MAE         MPE      MAPE      MASE         ACF1
Training set 0.07846889 2.824627 1.997025 0.04056302 1.564788 0.999271 -0.02146464
```

Figure 3.9: Accuracy of ARIMA(1,1,1)

We've applied another ARIMA model with the parameter lambda set to 'auto'. Our subsequent task involves forecasting. We've conducted the forecasting for the upcoming 30 days using this model. Below is the figure depicting the accuracy of the ARIMA(2,1,2) fitting described above.

```
> # Auto ARIMA with lambda = "auto"
> fit2<-auto.arima(AMZN_CP, lambda = "auto")
> tsdisplay(residuals(fit2), lag.max = 40, main='(2,1,2) Model Residuals')
> fcast2<-forecast(fit2, h=30)
> plot(fcast2)
> accuracy(fcast2)
                      ME      RMSE       MAE         MPE      MAPE      MASE         ACF1
Training set 0.07859248 2.824634 1.997148 0.04072362 1.564948 0.9993329 -0.02149591
```

Figure 3.10: Accuracy of ARIMA(2,1,2)

We further fine-tuned the ARIMA model, resulting in the optimal configuration of ARIMA(8,2,8). Below is the time series representation of model residuals obtained from this ARIMA model. Using this model, we forecast the next 30 days. We plot both the normal residuals and the standardized residuals. These residual plots help identify any outliers visually.

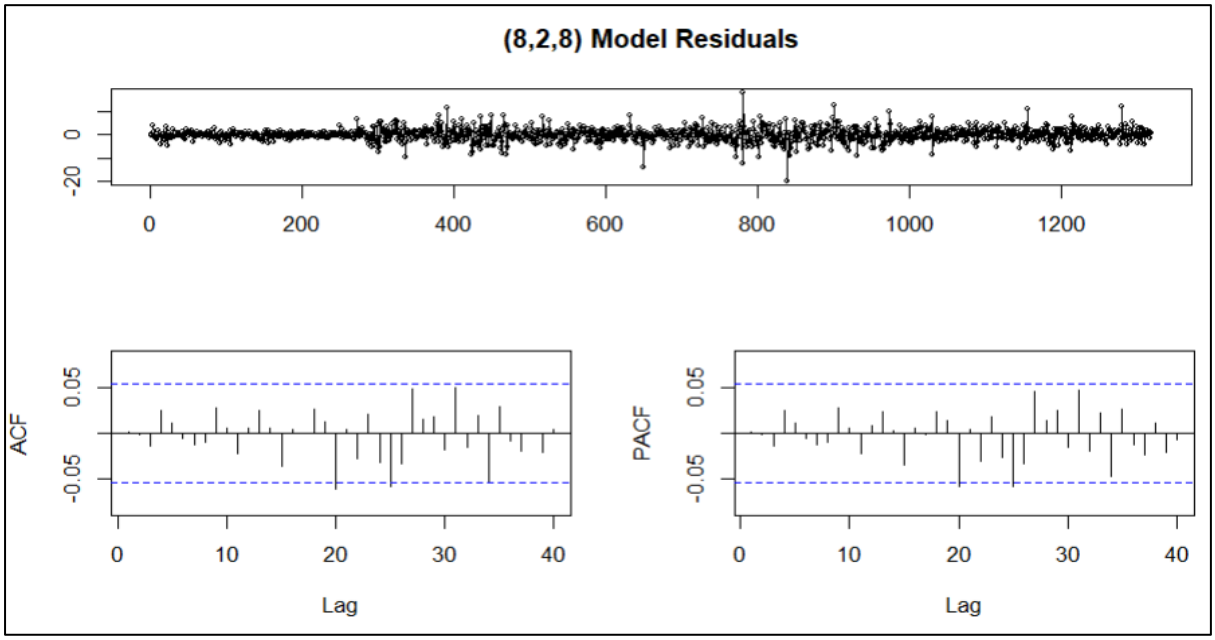

Figure 3.11: ARIMA (8,2,8) Residuals Plot

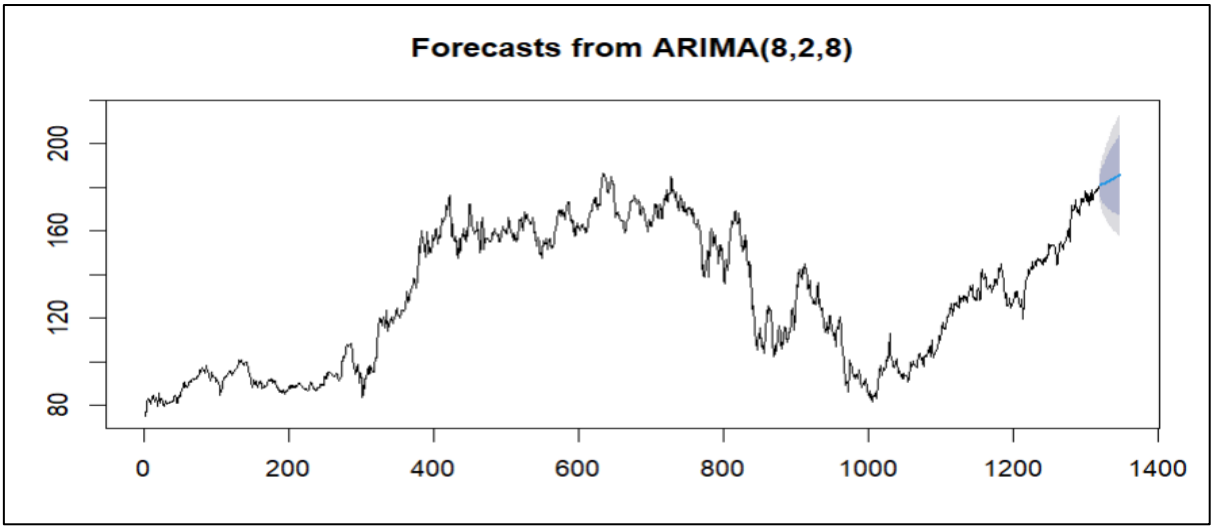

Figure 3.12: Forecasting Plot for ARIMA(8,2,8)

```
> accuracy(fcast3)
                     ME     RMSE      MAE          MPE     MAPE      MASE         ACF1
Training set 0.005884535 2.79802 1.973335 -0.01408278 1.548573 0.9874173 0.001869132
```

Figure 3.13: Accuracy of ARIMA(8,2,8) model

3.3 **KNN Regression Forecasting**

Our main objective is to use a KNN experimental approach to estimate future values of our stock price. The next 30 daily values for the close price dataset are forecasted after loading our package and converting it to a dataframe. For this prediction, we employ a value of k equal to 40, selected experimentally through a heuristic approach aimed at finding the optimal k value. While KNN algorithms typically require tuning experiments, our focus in this study is on demonstrating various forecasting tools rather than fine-tuning the model for maximum accuracy.

RMSE	MAE	MAPE
5.255123	4.335268	2.442462

Figure 3.14: KNN Accuracy

The model and related time series are used to evaluate the model's forecasting accuracy using the rolling origin function. After reviewing our model, we may use the following graph to visualize our predictions.

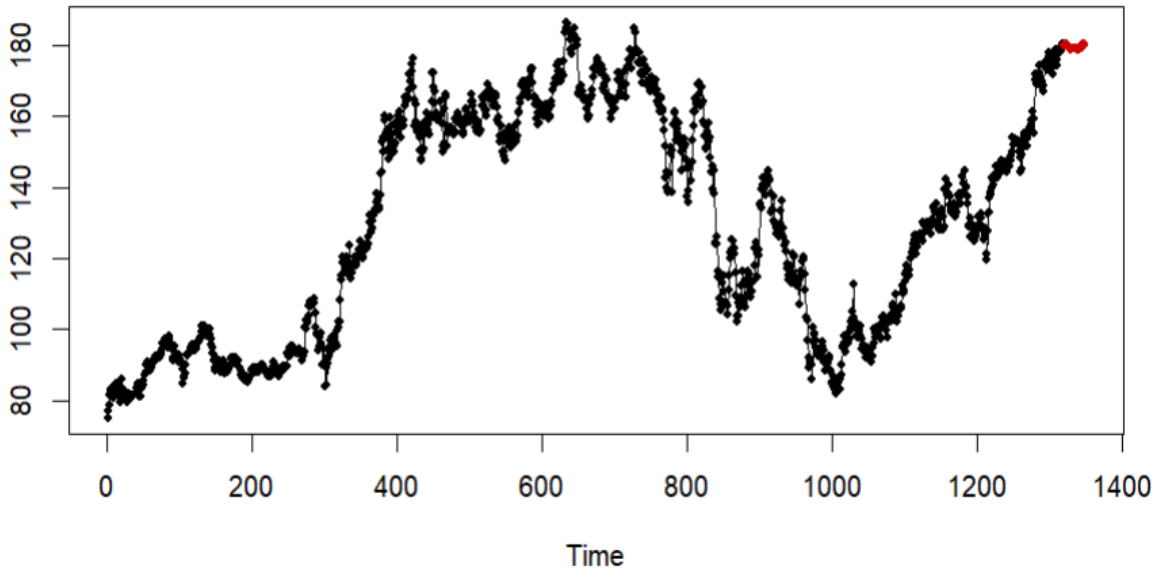

Figure 3.15: KNN Forecast

3.4 **Prophet Forecasting**

Formatting the dataset to fit Prophet's input requirements is the first step towards utilizing Prophet for predicting. After this conversion is finished, we can use the model to apply to the dataset and forecast values in the future.

```
MAPE for Prophet model: 2.100888 %
```

Figure 3.16: Prophet Forecasting Accuracy

Assessing the model's performance comes next, following the application of the model and the charting of the forecast. We will compare the real values with the training set's estimated values using the accuracy function as we are fresh to this model application. While setting up a cross-validation procedure to examine model performance indicators is the proper course of action in Prophet, our goal is to compare ARIMA with other models that employ the same methodology.

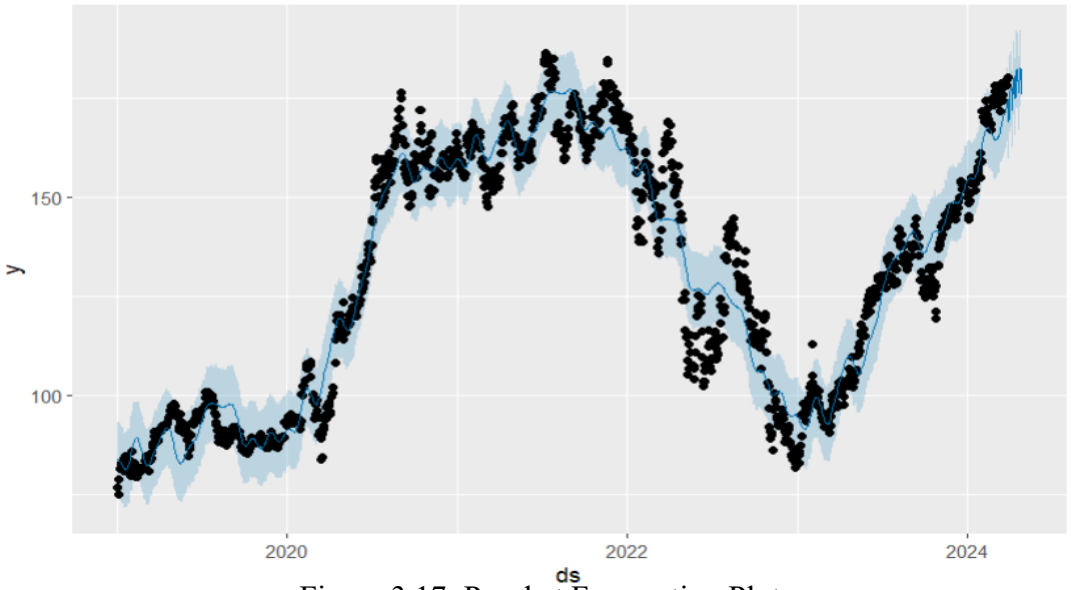

Figure 3.17: Prophet Forecasting Plot

Lastly, we may visualize our prophet components divided by a trend component, weekly seasonality, and yearly seasonality to gain a better understanding of the dataset.

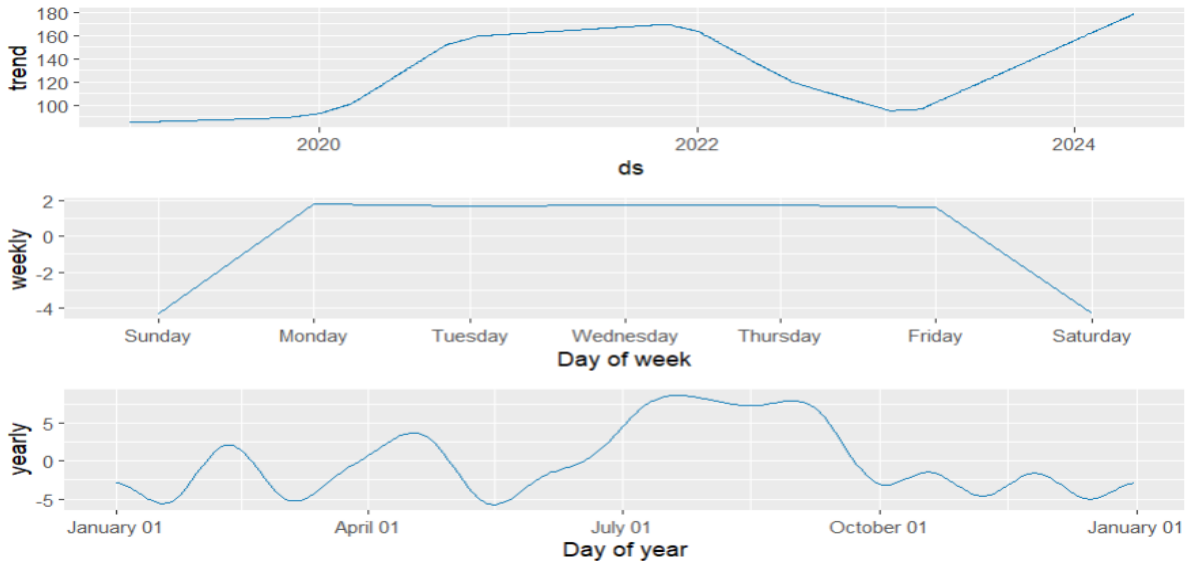

Figure 3.18: Prophet Forecasting Decomposition

Chapter 4

Results and Conclusions

Models	MAPE	Accuracy
ARIMA (Best)	1.3124	96.6875
KNN	2.4425	97.5575
Prophet	2.1009	97.8991

Table 4.1: Models with their respective MAPE and Accuracy

In this study, our focus was on the application of various models, aiming to understand their usage for forecasting future price values. Our results indicate that the models exhibited similar predictions for future tendencies, all projecting a higher price over the next 30 days. This suggests that ARIMA is the most suitable predictive model for this scenario. While the other models used did not perform as well as ARIMA according to our metrics, this could be attributed to the need for additional tuning phases and refinement of training and testing approaches. Alternatively, their effectiveness might be limited due to their primary application in classification rather than forecasting.

Chapter 5

Appendix

5.1 Code

The source code along with the data can be found here: (Stock-Price-Prediction/MSDS 511 - Stock Price Prediction in R.R at main · ReynAsilo/Stock-Price-Prediction (github.com))

References

[1] https://www.investopedia.com/terms/a/autoregressive-integrated-moving-average-arima.asp

[2] https://reyhannananta.medium.com/understanding-forecasting-model-using-prophet-a-comprehensive-guide-e05a76ecc5bb

[3]https://cran.r-project.org/web/packages/tsfknn/vignettes/tsfknn.html#:~:text=KNN%20is%20a%20very%20popular,numeric%20value%20(for%20prediction).