StockSight: Time Series Forecasting

Sachin Dimri (23N0063)

Guided by: Prof Neeraj Joshi
Indian Institute of Technology Delhi

July 9, 2025

Dedication

I dedicate this report with heartfelt appreciation and deep respect to my mentor, **Prof. Neeraj Joshi**, whose unwavering support, exceptional guidance, and constant encouragement have been pivotal in shaping this project. His insightful feedback and patient mentorship guided me at every critical juncture—from conceptualisation to execution—ensuring that I not only met our academic objectives but also deepened my understanding of data science and Time Series Forecasting. His mentorship inspired me to think beyond conventional approaches, engage deeply with the problem, and strive for both clarity and innovation in my work.

I, **Sachin Dimri**, also take this opportunity to recognize my own journey throughout this project. The countless hours of brainstorming, the sleepless nights, and the many moments of learning and resilience have made this experience truly rewarding. This project is not just an academic milestone—it is a reflection of personal dedication, perseverance, and growth.

This report represents more than just an project submission—it reflects the dedicated effort, continuous learning, and the invaluable support of an exceptional mentor. It serves as a reminder that with the right guidance and collective commitment, even the most complex challenges can be trans- formed into meaningful achievements. I am truly grateful for his guidance and for the opportunity to undertake this enriching journey under his supervision.

Declaration

I hereby declare that this written submission is the result of my own work and represents my ideas in my own words. Wherever the ideas or words of others have been used, appropriate acknowledgment has been given through proper citation and referencing.

I affirm that I have upheld the principles of academic honesty and integrity, and that this submission does not contain any misrepresentation, fabrication, or falsification of data, facts, or sources. I understand that any violation of the above principles may lead to disciplinary action by the Institute and may also attract legal consequences from the original sources if permissions have not been duly obtained or citations not properly provided.

Sachin Dimri
24N0083

Date: 13/08/2025
Indian Institute of Technology Bombay
Mumbai-400076, India

Abstract

This project presents a comprehensive approach to stock price forecasting using advanced time series analysis techniques, with a focus on evaluating and comparing the performance of different models across companies from diverse economic sectors.

The study begins with an in-depth analysis of Axis Bank stock data, applying foundational time series methods such as decomposition, stationarity testing, model fitting, and accuracy evaluation. After establishing a robust modeling framework, the methodology was extended to additional stocks representing various sectors, including banking (ICICI Bank), information technology (TCS), infrastructure (L&T), and conglomerates (Reliance Pvt. Ltd.).

The primary objective is to capture sector-specific patterns in stock price movements and assess the generalizability of models such as ARIMA, SARIMA, XGBoost, and Prophet—both in their default configurations and with hyperparameter tuning. The best-performing model was further deployed using a Streamlit web interface, enabling intuitive, user-friendly forecasting on new datasets.

This study underscores the value of combining classical time series models with modern tools such as Optuna and Prophet to develop scalable, automated forecasting solutions tailored to the Indian stock market.

Contents

Chapter 1
Introduction

Stock price forecasting is a fundamental task in financial analytics, guiding investment strategies and risk management. Accurate predictions of future stock prices are crucial for investors, financial analysts, and policy makers. This project aims to explore and compare various time series forecasting techniques for modeling and predicting stock prices across different sectors.

The focus of this study is on assessing the performance of core time series models such as Decomposition Models, ARIMA, SARIMA, and Facebook's Prophet, along with a few machine learning approaches. Among these, Prophet played a particularly central role due to its robustness in handling volatility and automatic tuning capabilities.

We began the analysis using stock price data of Axis Bank to build a foundational understanding of each model's performance. Once this baseline was established, the study was extended to include datasets from other companies representing diverse sec- tors: ICICI Bank (Banking), TCS (Consulting), and Larsen & Toubro Limited (Engineer- ing). All stock price data was obtained from the National Stock Exchange (NSE) of India, covering the period from January 2023 to December 2024. The data consisted solely of business days and required minimal preprocessing, as there were no significant missing values.

The primary objective of the project is to evaluate the ability of different forecasting models to generalize across sectors with distinct patterns and volatility. While traditional models like ARIMA and SARIMA are effective for relatively stable time series, we observed that they struggle when applied to highly volatile stock data. In contrast, Prophet demonstrated strong performance in capturing irregular patterns when appropriately tuned, making it a promising tool for generalized forecasting tasks in the financial domain.

This report details the methodology, modeling techniques, comparative analysis, and key insights obtained through the forecasting of stock prices across sectors. It also highlights the limitations and challenges encountered during the project, providing directions for future enhancement.

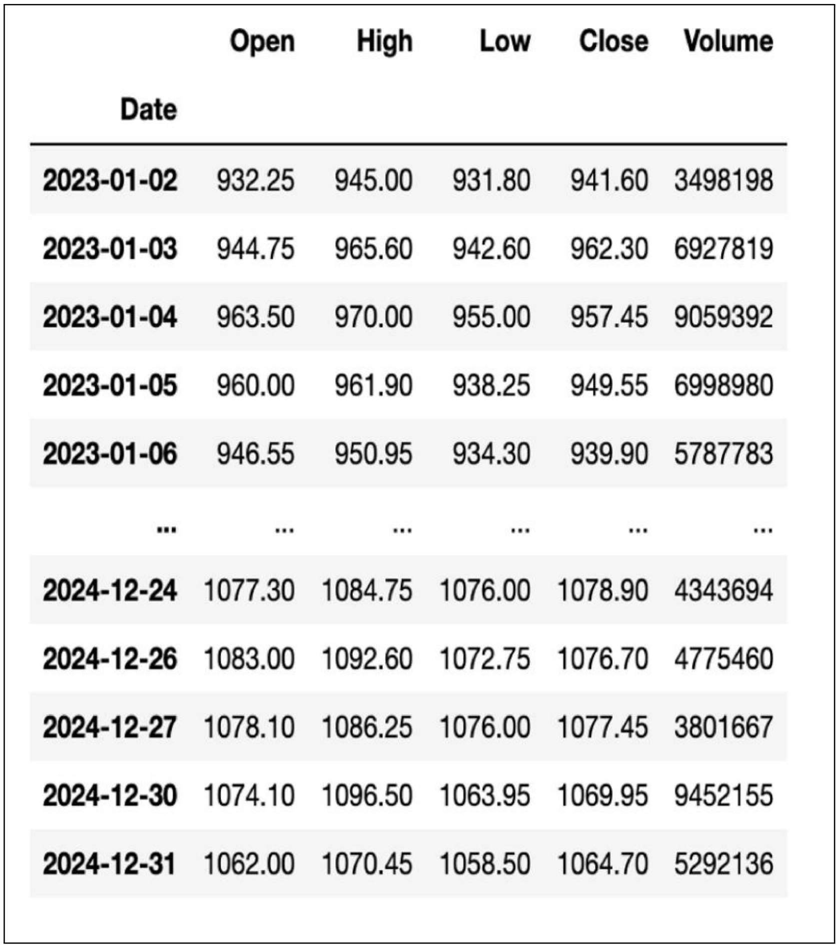
1.1 Data Description

- Dataset Source: NSE market data.
- Duration: January 2023 to December 2024.
- Frequency: Daily stock prices (over 400 trading days).
- Initial stock: Axis Bank.
- Data Preparation: Cleaning, handling missing values, formatting timestamps.

Chapter 2
Exploratory Data Analysis

This section explores the dataset to understand its key characteristics and identify patterns or trends that could inform model development. We begin with a snapshot of the raw data, followed by a summary of descriptive statistics, and finally, we perform some visual analyses to capture any notable trends or anomalies.

2.1 Data Snapshot



	Open	High	Low	Close	Volume
Date					
2023-01-02	932.25	945.00	931.80	941.60	3498198
2023-01-03	944.75	965.60	942.60	962.30	6927819
2023-01-04	963.50	970.00	955.00	957.45	9059392
2023-01-05	960.00	961.90	938.25	949.55	6998980
2023-01-06	946.55	950.95	934.30	939.90	5787783
...	...	...	...	...	...
2024-12-24	1077.30	1084.75	1076.00	1078.90	4343694
2024-12-26	1083.00	1092.60	1072.75	1076.70	4775460
2024-12-27	1078.10	1086.25	1076.00	1077.45	3801667
2024-12-30	1074.10	1096.50	1063.95	1069.95	9452155
2024-12-31	1062.00	1070.45	1058.50	1064.70	5292136

Figure 1: Snapshot of Axis Bank closing prices from January 2023 to December 2024

Figure 1 shows the time series of Axis Bank's closing stock prices from January 2023 to December 2024. The dataset captures daily stock prices, with visible trends and fluctuations over time, including periods of volatility. This initial view of the data highlights the need for careful modeling to account for these variations.

2.2 Descriptive Statistics

To better understand the behavior of the stock prices, we summarize the descriptive statistics of the opening and closing prices.

Statistic	Close Price	Open Price
Count	495	495
Mean	1050.63	1051.04
Std	120.82	120.82
Min	824.05	821.50
10%	868.57	871.91
25%	959.33	960.00
50% (Median)	1059.10	1056.25
75%	1147.70	1150.00
90%	1187.12	1191.30
99%	1292.61	1300.06
Max	1317.30	1316.70

Table 1: Descriptive statistics of Axis Bank stock prices (Opening and Closing)

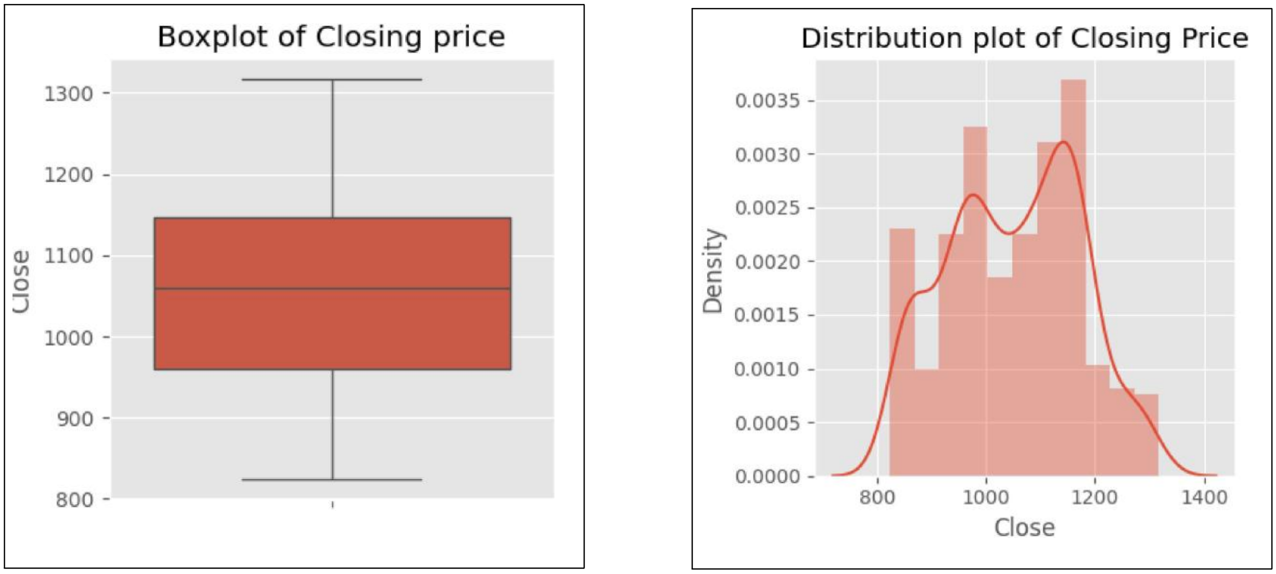
Table 1 presents the key descriptive statistics for Axis Bank's opening and closing stock prices. These include measures like the mean, standard deviation, skewness, and kurtosis, which help characterize the data's central tendency and dispersion.

Key Observations:

- The means of opening and closing prices are nearly identical, indicating pricing consistency throughout the day.
- The standard deviation of 120 suggests significant volatility in the stock's price.
- The price range for opening and closing prices spans 800 to 1320, with occasional sharp peaks above 1300.
- Skewness values close to zero (Close: -0.02, Open: 0.01) indicate nearly symmetric distributions.
- The kurtosis values (Close: -0.91, Open: -0.91) suggest a relatively flat distribution with fewer extreme outliers.

2.3 Time Series Visualization

We begin our visual analysis with basic descriptive plots that provide insight into the distribution and characteristics of Axis Bank's closing prices. These preliminary visualizations help us identify potential outliers and examine the overall behavior of the stock price over the study period.



(a) Boxplot of closing prices

(b) Distribution of closing prices

Figure 2: Boxplot and Distribution plot of closing prices

Insights:

- The boxplot indicates no significant outliers, suggesting relatively consistent price behavior.
- The distribution plot reveals a bimodal pattern, possibly indicating a regime change or the influence of multiple market forces affecting the price dynamics during the observed period.

Next, we take a step further by visualizing the time series of Axis Bank's closing stock prices over the entire period of study, from January 2023 to December 2024. This original time series plot offers a clearer view of the data's trends and fluctuations, helping us to identify longer-term patterns and short-term volatility.

Original Time Series Plot:

The time series plot below illustrates the daily closing stock prices of Axis Bank over the specified period. By observing the plot, we can explore the overall trend of the stock and any visible patterns, such as seasonality or sudden changes in price.

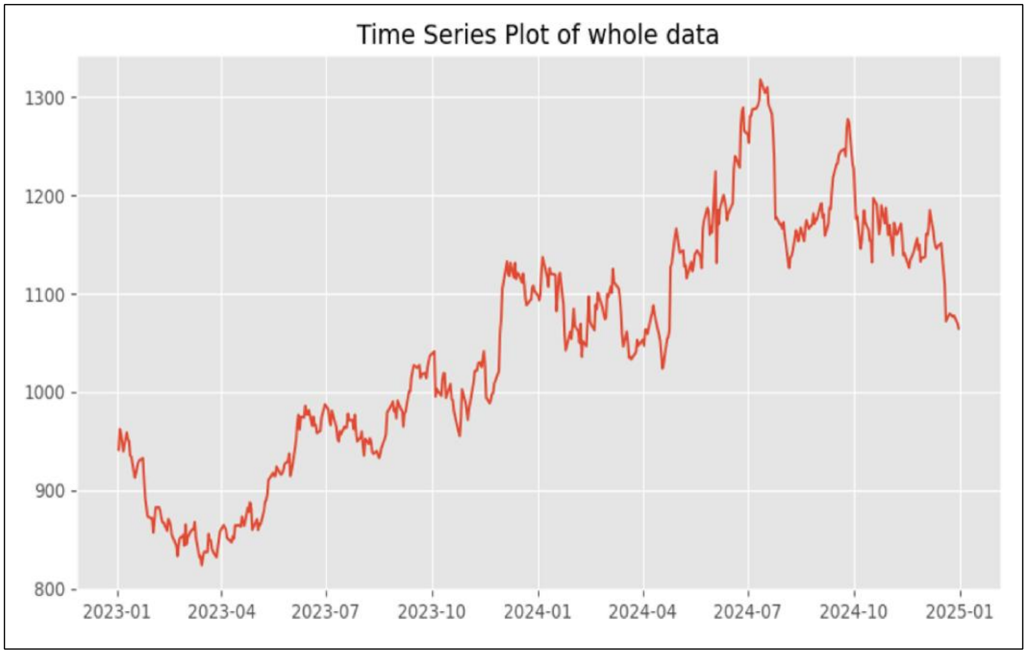


Figure 3: Original Time Series Plot of Axis Bank Closing Prices

Insights:

- The time series plot reveals clear trends in the stock price, with observable fluctuations occurring throughout the observed period.
- There are periods of upward or downward movement that may reflect broader market conditions or specific company-related news or events.
- Some sharp fluctuations and high volatility are evident, indicating potential market responses to news or broader economic changes.
- The overall trend suggests the influence of both external market forces and company-specific events.

Next Steps: From this plot, we can raise several important questions that will guide us towards the next phase of analysis:

- Can we observe any repetitive patterns or seasonality in the data?
- How can we quantify these trends and patterns to better forecast future price movements?

These questions highlight the need to explore the underlying components of the time series data. To answer them, we will decompose the time series into its trend, seasonality, and residual components. The next section will delve into this decomposition to better understand the stock price dynamics and identify the key factors driving the observed price behavior.

2.4 Decomposition of Time Series

Building upon our previous visualizations, we now decompose the time series data of Axis Bank's closing stock prices into its constituent components: trend, seasonality, and residuals. Time series decomposition helps us better understand the underlying patterns and trends in the data, which is crucial for accurate forecasting.

Time series decomposition is typically done using statistical methods like additive or multiplicative decomposition. In this analysis, we employ an additive decomposition model, assuming that the components (trend, seasonality, and residuals) are added together to form the observed time series.

Decomposition Plot: The decomposition of the original time series reveals the distinct components contributing to the overall price behavior. Below, we present the decomposition of Axis Bank's closing prices.

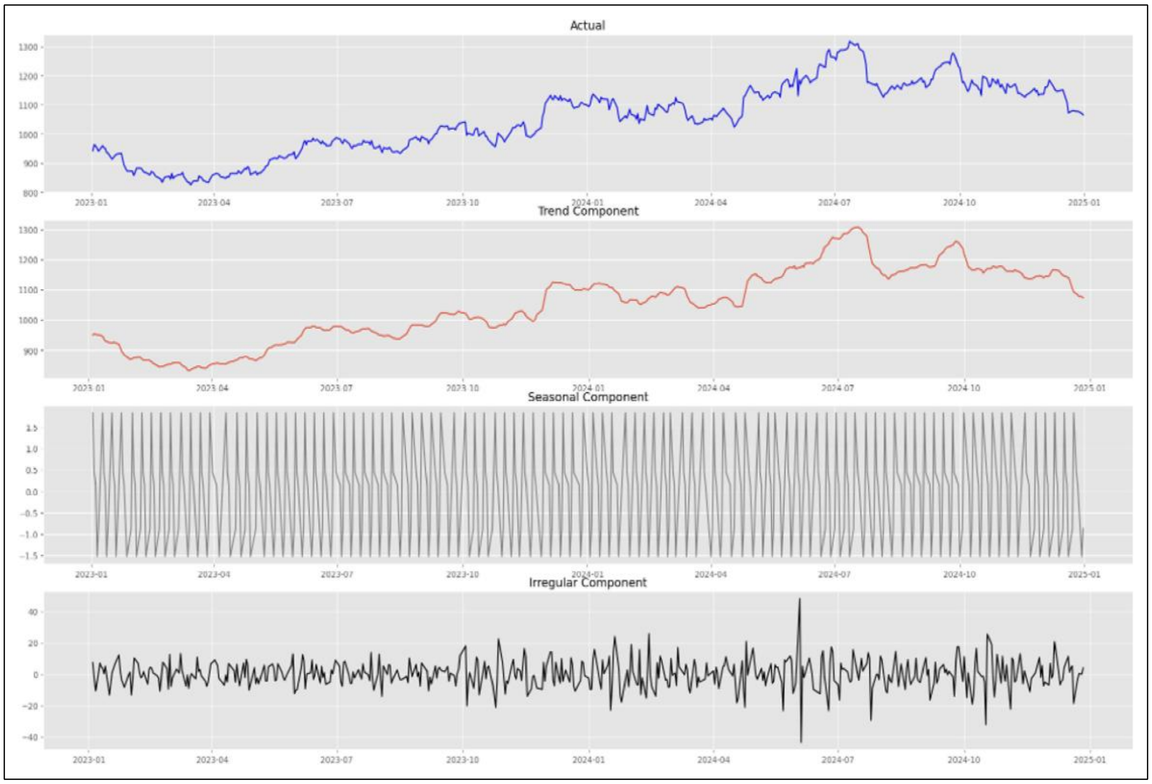


Figure 4: Decomposition of Time Series: Trend, Seasonality, and Residuals

Insights:

- **Trend Component:** The trend plot shows the long-term movement of the stock price. It illustrates whether the price is generally increasing, decreasing, or remaining stable over time. In this case, the trend seems to reflect a gradual upward movement with some fluctuations.

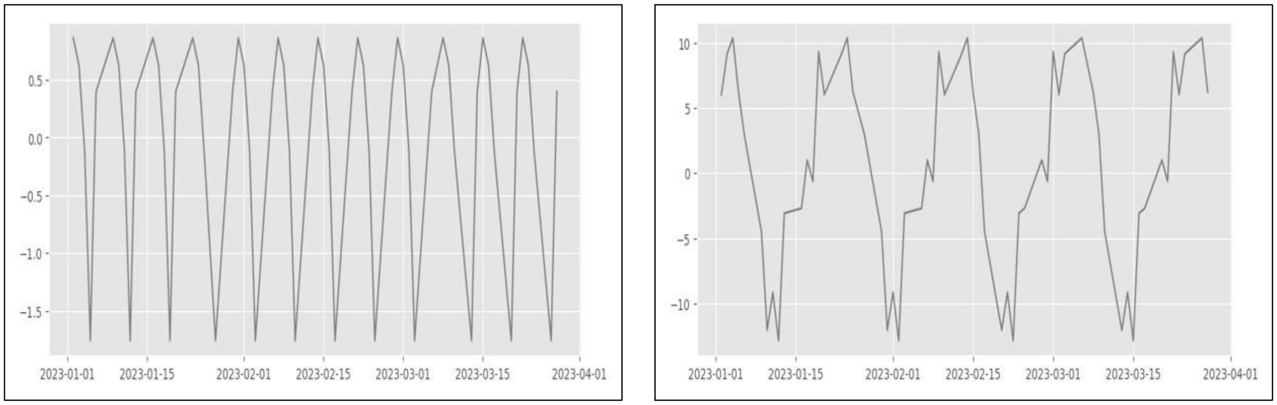
- **Seasonal Component:** The seasonal plot reveals any periodic fluctuations in the stock price, such as regular up-and-down movements over specific time intervals.

These patterns may be related to factors like quarterly earnings reports, market sentiment, or broader economic cycles. We have choosen period 5 to see weekly seasonality and can see consistent periodic movement over the time.

- **Residual Component:** The residuals plot captures the remaining variations in the data after accounting for trend and seasonality. It represents the "noise" in the time series and highlights any random fluctuations or outliers that were not explained by the trend and seasonality.

By decomposing the time series, we can now better understand the contributions of each component. This allows us to make more informed predictions about future price movements, as we can separate the underlying trends and seasonality from random fluctuations.

Next, we zoom in specifically on the seasonality component by experimenting with different periodicities to identify the best-fitting seasonal cycle.



(a) Seasonality Component (Period = 5) (b) Seasonality Component (Period = 14)

Figure 5: Comparison of Seasonality Components at Different Periods

Insights:

- With a period of 5, the seasonality component exhibits consistent cyclical behavior. However, the magnitude of seasonal variation is quite low, ranging approximately between - 1.5 and -0.5.
- In contrast, with a period of 14, the amplitude of the seasonal component is significantly higher, indicating stronger and more impactful seasonal patterns in the stock prices.
- This suggests that a longer periodic cycle may be more appropriate for capturing the recurring behavior in Axis Bank's stock prices.

Next Steps: With the decomposition completed, we now have a clearer understanding of the underlying structure of Axis Bank's stock prices. Before moving on to forecasting, we will delve deeper into the time series by examining additional insights such as the rolling mean and rolling standard deviation. These measures will help us assess the stability and evolving nature of the stock's behavior over time. Following that, we will explore further analytical visualizations to enhance our understanding of the data.

2.5 Rolling Mean and Standard Deviation

To further understand the stability and stationarity of the time series, we now plot the rolling mean and rolling standard deviation along with the original closing price data. This helps identify whether the statistical properties of the series change over time, which is crucial for selecting appropriate forecasting models.

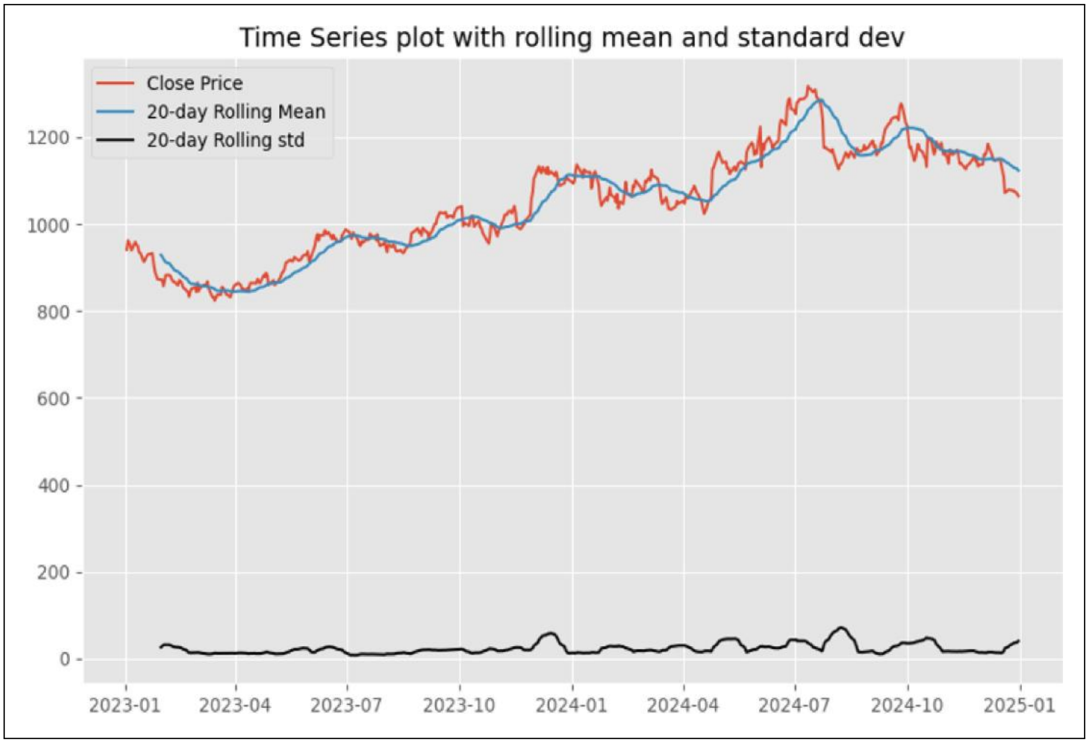


Figure 6: Rolling Mean and Standard Deviation of Axis Bank Closing Prices

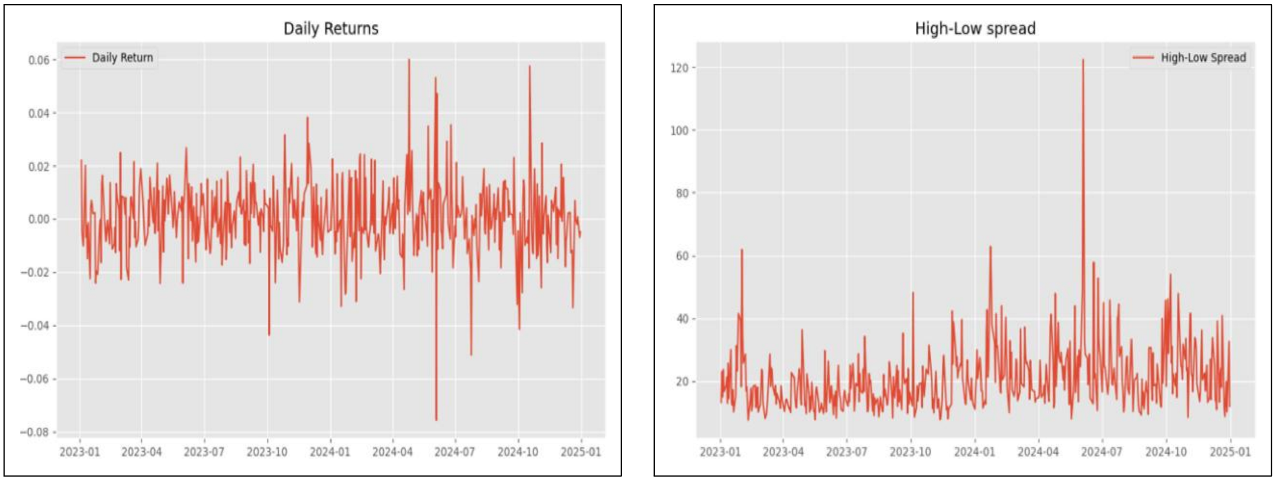
Insights:

- The rolling mean follows a smooth, wave-like pattern-resembling a sine or cosine curve-indicating the **presence of seasonality** in the data.
- The rolling mean shows **smooth trends** in the overall movement of prices, offering a clearer view of the underlying direction by filtering out short-term fluctuations.
- The rolling standard deviation helps assess **volatility over time;** it captures how spread out the price movements are.

- The black line (rolling standard deviation) shows **multiple spikes,** indicating periods of **high volatility.** For example, notable peaks around **December 2023** and **July 2024** may reflect **external shocks or company-specific events.**

2.6 Additional Insights from Daily Returns and Volatility Spread

While rolling statistics help us capture overall trends and volatility, deeper insights can be gained by analyzing **daily returns** and the **high-low spread.** These plots help us understand how the stock behaves on a day-to-day basis, revealing its risk profile and price momentum characteristics.



(a) Daily Returns of Axis Bank Stock

(b) Daily High-Low Price Spread

Figure 7: Volatility and Return-Based Insights

Insights:

- The daily returns plot highlights the *percentage change in closing price from one day to the next,* showing both stable periods and sudden market reactions.
- Returns fluctuate between -0.08 to +0.06, indicating *moderate volatility* in daily price changes.
- Returns oscillate around zero, which is *typical for stable stocks* with no sustained upward or downward drift.
- Frequent crossings of the zero line imply *no long-term directional bias* in daily performance.
- The high-low spread plot visualizes *intraday volatility,* i.e., the difference between the highest and lowest price on each trading day.
- Larger spreads indicate *uncertainty or speculative activity,* while tighter spreads sug- gest *calm and predictable market behavior.*

Chapter 3
Time Series Analysis

After completing the exploratory data analysis (EDA), the next step is to delve deeper into the properties of the time series data. This phase focuses on ensuring the data is suit- able for modeling by checking its stationarity and analyzing the autocorrelation structure. Based on these analyses, we can select the appropriate time series model.

3.1 Test for Stationarity (ADF Test)

Before proceeding with time series modeling, it is important to test whether the time series is stationary. A stationary time series has constant mean and variance over time and does not exhibit seasonal effects or long-term trends. Many time series models, such as ARIMA, assume stationarity as a prerequisite.

The **Augmented Dickey-Fuller (ADF)** test is a statistical test used to determine if a time series has a unit root, which would make it non-stationary. The presence of a unit root indicates that the time series is non-stationary, whereas the absence of a unit root suggests that the series is stationary.

The null hypothesis (Ho) and alternative hypothesis (H1) for the ADF test are as follows:

- **Null Hypothesis (H):** The time series has a unit root, meaning it is non-stationary.
- **Alternative Hypothesis (H):** The time series does not have a unit root, meaning it is stationary.

If the p-value obtained from the ADF test is less than a chosen significance level (typically 0.05), we reject the null hypothesis, indicating that the time series is stationary. If the p-value is greater than the significance level, we fail to reject the null hypothesis, implying that the time series is non-stationary.

The ADF test also provides critical values at different significance levels (1

Interpretation of ADF Test Results:

For our dataset, the results of the ADF test are as follows:

- **ADF Statistic:** -1.455649
- **p-value:** 0.555326
- **Critical Value** (1%): -3.443739
- **Critical Value (5%):** -2.867444
- **Critical Value (10%):** -2.569915

The null hypothesis states that the time series has a unit root (i.e., it is non-stationary). The ADF statistic is -1.456, which is greater than the critical values at all significance levels (1%, 5%, and 10%). Additionally, the p-value is 0.555, which is much greater than the commonly used significance level of 0.05.

Since the p-value is high and the ADF statistic is greater than the critical values, we fail to reject the null hypothesis. This suggests that the time series is non-stationary and may need to be transformed (e.g., differenced) to achieve stationarity before proceeding with further time series modeling.

Conclusion:

Based on the ADF test results, we conclude that the time series is non-stationary. Further steps, such as differencing, may be required to transform the series into a stationary form before applying time series models like ARIMA.

3.2 ACF and PACF Analysis

Once stationarity is established (or addressed if needed), the next step is to analyze the autocorrelation (ACF) and partial autocorrelation (PACF) functions. These plots are cru- cial for identifying the structure of the time series and for selecting appropriate lags for autoregressive (AR) and moving average (MA) components in models such as ARIMA. **ACF Plot:** The **Autocorrelation Function (ACF)** measures the correlation between the series and its lagged versions. By examining the ACF plot, we can identify the number of MA terms (q) that should be included in the model. **PACF Plot:** The **Partial Autocorrelation Function** (PACF) shows the correlation between the series and its lagged versions after removing the effect of intermediate lags. The PACF plot helps identify the number of AR terms (p) that should be included in the model.

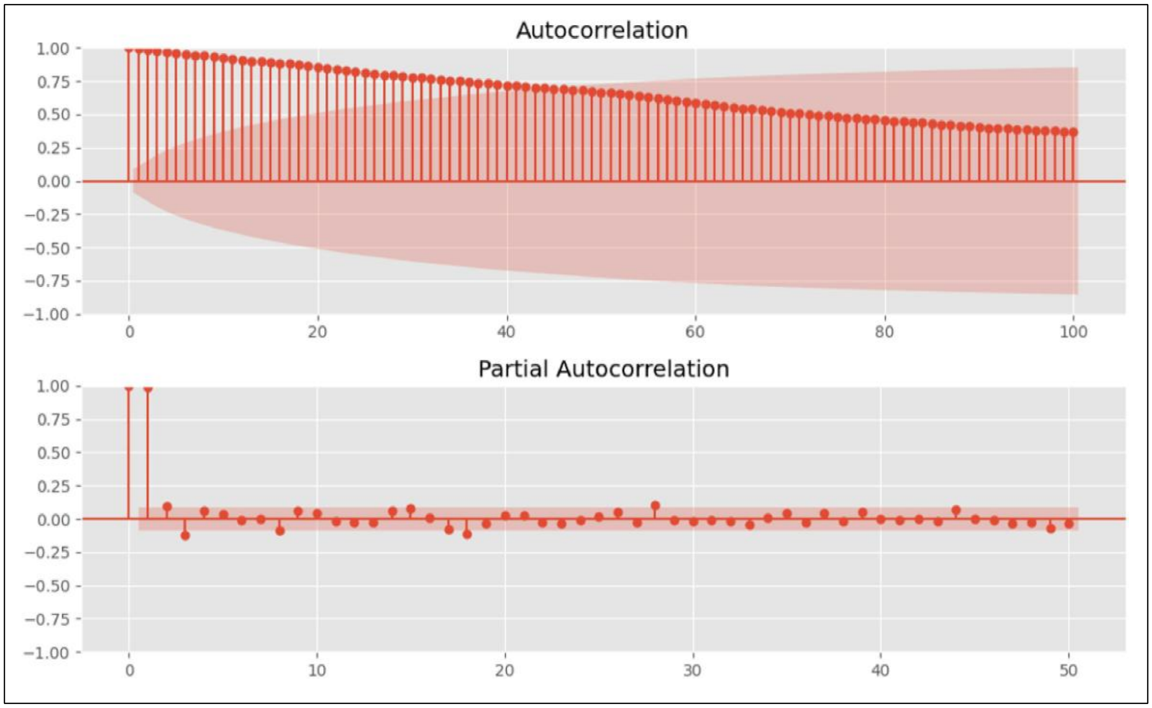


Figure 8: ACF Plot of Axis Bank Closing Prices

Figure 8 shows ACF and PACF plot in which we observe that the autocorrelation slowly decays, indicating that there is a significant correlation between the observations at different lags. This suggests that the time series may exhibit long-term dependencies. On the other hand, the PACF plot shows a sharp spike at lag 1, followed by values close to zero at subsequent lags. This pattern is characteristic of an AR(l) process, meaning that the time series is likely to be best modeled by a first-order autoregressive model (AR(l)).

Based on these plots, we can hypothesize that an AR(l) model might be appropriate for the time series data, where the price at time *t* depends linearly on the price at time *t* -1.

Next Steps: With the stationarity test and ACF /PACF analysis complete, we are now ready to proceed with model selection. These analyses will guide us in choosing the appropriate time series model, such as ARIMA or SARIMA, for forecasting Axis Bank's stock prices. We will use the insights from the ACF and PACF plots to help determine the optimal values for the autoregressive (AR) and moving average (MA) components, ensuring an effective forecasting model.

Chapter4
Time Series Modeling

After exploring and analyzing the data thoroughly, we now shift our focus to modeling the time series in order to forecast future stock prices. Before selecting appropriate models, it's crucial to recall two important observations from our previous analysis:

- The Augmented Dickey-Fuller (ADF) test indicated non-stationarity in the data, primarily due to the presence of a trend.
- Seasonal decomposition revealed evident seasonal components within the series.

These findings guide us away from using basic models such as Autoregressive (AR), Moving Average (MA), or ARMA, which rely on the assumption of stationarity. In- stead, we move forward with more sophisticated models that can account for both non-stationarity and seasonality.

Modeling Strategy

To ensure a structured and effective approach, we adopt the following strategy:

- **Model Selection:** We implement the ARIMA model, which handles non-stationarity by differencing the data. Additionally, to capture the seasonal structure identified earlier, we extend this to the SARIMA (Seasonal ARIMA) model.
- **Model Tuning:** We use a combination of ACF/PACF plots and automated hyperparameter search techniques to identify optimal model parameters.
- **Evaluation Metrics:** Model performance is assessed using:
  - Root Mean Squared Error (RMSE)
  - Mean Absolute Error (MAE)
  - Mean Absolute Percentage Error (MAPE)
- **Model Validation:** Beyond performance metrics, we also evaluate the reliability and consistency of our models through:
  - **Back-testing:** Repeatedly training the model on past data and forecasting forward to verify robustness over different time windows.
  - **Cross-validation:** Measuring forecast accuracy across multiple hold-out sets to ensure that performance is not dependent on a specific split of data.

This comprehensive approach enables us to select the most suitable model not only based on accuracy but also on its ability to generalize across time and different stock scenarios.

Overview of ARIMA and SARIMA Models

ARIMA (Autoregressive Integrated Moving Average) is a generalization of ARMA that includes differencing (the 'I' component) to handle non-stationary time series. It is defined by three parameters:

- p: the number of autoregressive terms (AR)
- d: the number of times the data needs to be differenced to achieve stationarity
- q: the number of lagged forecast errors in the prediction equation (MA)

The ARIMA model can be expressed as:

$$y_t = c + \phi_1 y_{t-1} + \cdots + \phi_p y_{t-p} + \theta_1 \varepsilon_{t-1} + \cdots + \theta_q \varepsilon_{t-q} + \varepsilon_t$$

after differencing d times.

SARIMA (Seasonal ARIMA) extends ARIMA by incorporating seasonal components into the model. These seasonal terms capture repeated patterns over fixed periods (e.g., weekly, monthly, quarterly). The full SARIMA model is denoted as:

$$\text{SARIMA}(p,d,q)(P,D,Q)m$$

where:

- P, D, Q: seasonal autoregressive, differencing, and moving average orders
- m: the number of periods in each season (e.g., 12 for monthly data with yearly seasonality)

This allows SARIMA to model both short-term dynamics and recurring seasonal effects within the data-an important consideration for financial time series like stock prices.

4.1 ARIMA Model

The ARIMA model (AutoRegressive Integrated Moving Average) is a widely used time series forecasting model. We first examine the performance of the ARIMA(l,1,0) model, followed by a more refined ARIMA model with hyperparameter tuning.

4.1.1 ARIMA(l,1,0)

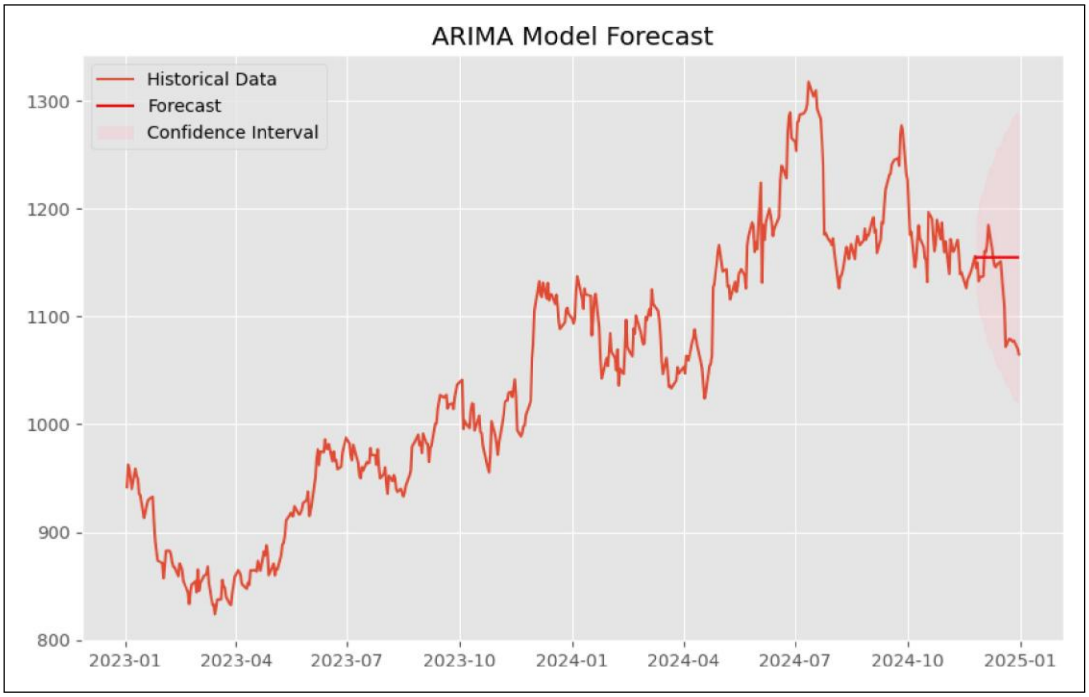


Figure 9: Forecast using ARIMA(l,1,0) Untuned Model

Observation: The ARIMA(l,1,0) model suggested by ACF and PACF plots, yields poor forecasts. The model fails to capture the structure and volatility of the stock prices, likely due to the complexity and semi-random nature of financial time series. Since the model was not fine-tuned, its performance is limited.

4.1.2 Hyperparameter-Tuned ARIMA(4,1,2)

Before arriving at the optimal ARIMA model, we conducted an iterative hyperparameter tuning process, testing various combinations of the parameters *(p,d,q)* and evaluating performance based on RMSE, MAE, and MAPE.

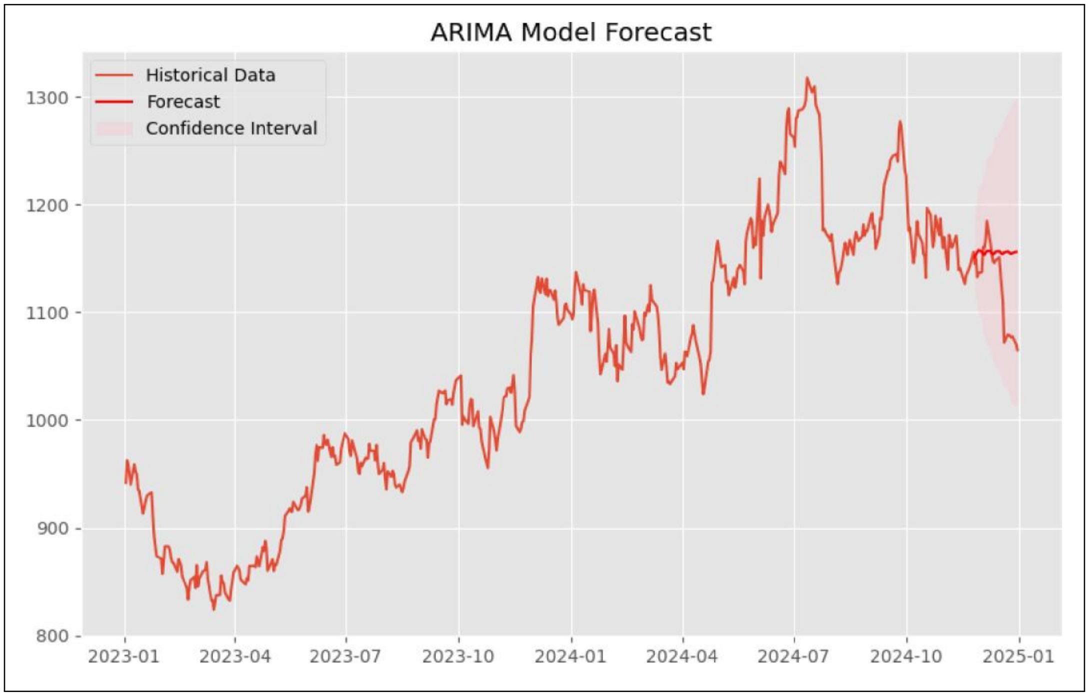


Figure 10: Forecast using tuned ARIMA(4,1,2)

Performance:

- Mean Absolute Error (MAE): 33.07
- Root Mean Squared Error (RMSE): 45.78
- Mean Absolute Percentage Error (MAPE): 3.02

With appropriate tuning, the ARIMA model performs significantly better. However, it still lacks the ability to model the seasonal patterns observed earlier.

4.2 SARIMA Model

Given the seasonal behavior identified during decomposition, we extend the ARIMA model to SARIMA by incorporating seasonal components. The seasonal parameters (P, D, Q, m) are selected based on the decomposition analysis, and the non-seasonal parameters (p, d, q) are inherited from the tuned ARIMA model.

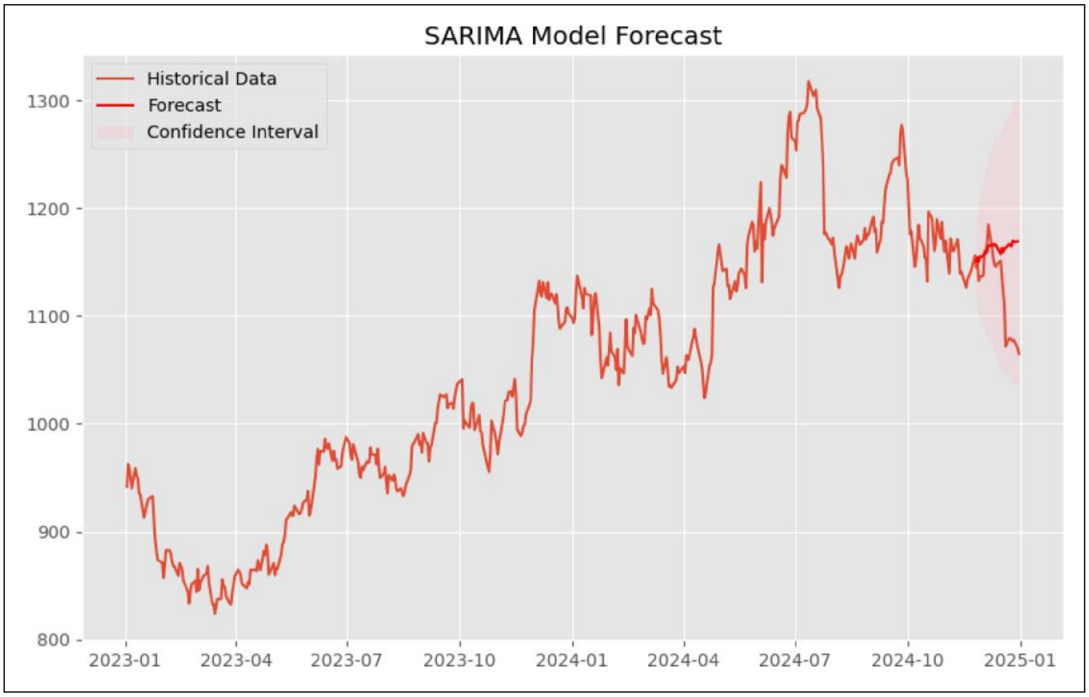


Figure 11: Forecast using SARIMA(4,1,2)(1,1,1)[14]

SARIMA Model Parameters:

- Non-seasonal parameters: $(p = 4, d = 1, q = 2)$ from the tuned ARIMA model.
- Seasonal parameters: $(P = 1, D = 1, Q = 1, m = 14)$ based on seasonal behavior observed in the decomposition analysis.

Performance:

- MAE: 37.38
- RMSE: 52.05
- MAPE: 3.42

Conclusion: The SARIMA model performs better at capturing the seasonal patterns present in the data, whereas the ARIMA model tends to forecast a nearly linear trend. Metric-wise, the tuned ARIMA model slightly outperforms SARIMA in terms of MAE (33.07 vs. 37.38). However, both models show suboptimal performance, suggesting that classical time series methods alone might not be sufficient to fully capture the complexity and volatility of stock price data.

Chapter 5
Advanced Modeling Approaches

While classical time series models like ARIMA and SARIMA offer interpretable solutions, their capacity to handle complex, nonlinear patterns-especially those present in stock price data-is limited. Therefore, we now shift our focus to advanced modeling techniques that leverage machine learning and flexible forecasting frameworks.

5.1 XGBoost: Gradient Boosting for Time Series

Extreme Gradient Boosting (XGBoost) is a high-performance, scalable implementation of gradient boosting that has become one of the most popular machine learning algorithms. It is particularly effective in capturing non-linear relationships and interactions among variables-making it a strong candidate for time series modeling when used with lagged features, rolling statistics, and engineered covariates.

Why XGBoost?

- Handles non-linear patterns and noise better than linear models.
- Offers regularization to prevent overfitting.
- Supports automated hyperparameter tuning and efficient computation.
- Can incorporate external regressors and engineered features.

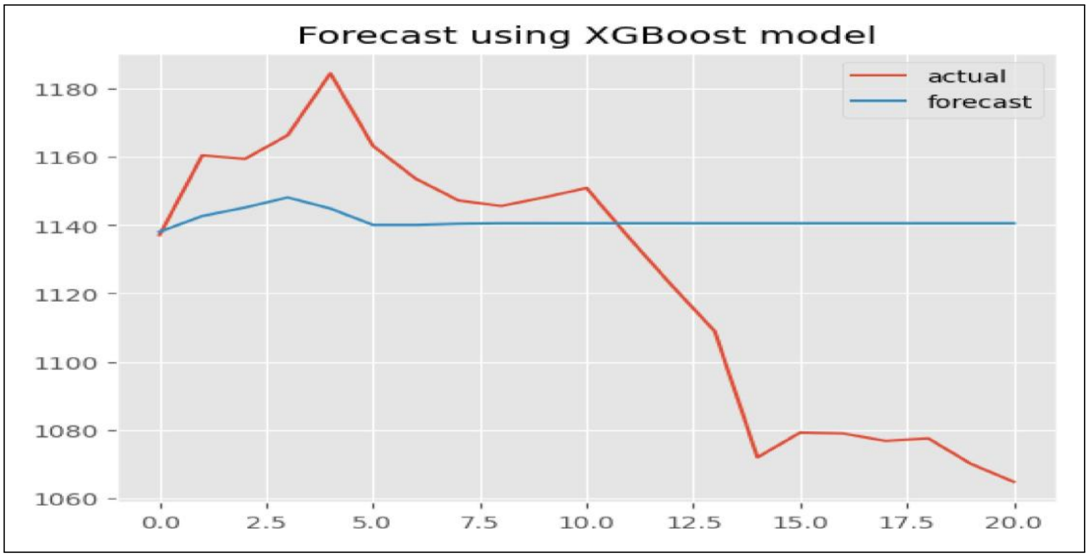


Figure 12: Forecast using tuned XGBoost model

Observation: Despite its theoretical strengths, the XGBoost model in our context did not yield favorable results. Even with hyperparameter tuning via Grid Search Cross-Validation, the model struggled to capture meaningful patterns in the time series, as evident from the forecast plot. This outcome likely stems from the lack of strong exogenous features or additional engineered signals that XGBoost typically relies on in time series forecasting tasks.

5.2 Facebook Prophet: A Flexible Time Series Forecasting Framework

Facebook Prophet is an open-source procedure developed by Meta for producing high-quality forecasts for time series data with strong seasonal effects and several missing values. It is based on an additive model where non-linear trends are fit with yearly, weekly, and daily seasonality, along with holiday effects.

Why Prophet?

- Automatically detects trends, seasonality, and changepoints.
- Intuitive parameter tuning without needing deep statistical expertise.
- Robust to missing data and outliers.
- Designed for business time series applications such as retail or finance.

Key Parameters:

- changepoint_prior _scale: Controls the flexibility of the trend component. A higher value allows more rapid trend changes.
- seasonality_prior_scale: Regulates the strength of seasonality. Higher values allow stronger seasonal patterns.
- holidays_prior_scale: Determines the influence of holidays on the forecast.
- seasonality _mode: Specifies whether the seasonal effect is additive or multiplicative.
- n_changepoints: Number of potential changepoints in the trend.
- yearly _seasonality, weekly _seasonality, daily_seasonality: Boolean flags that enable corresponding seasonal patterns.

5.2.1 Prophet Model with Default Parameters

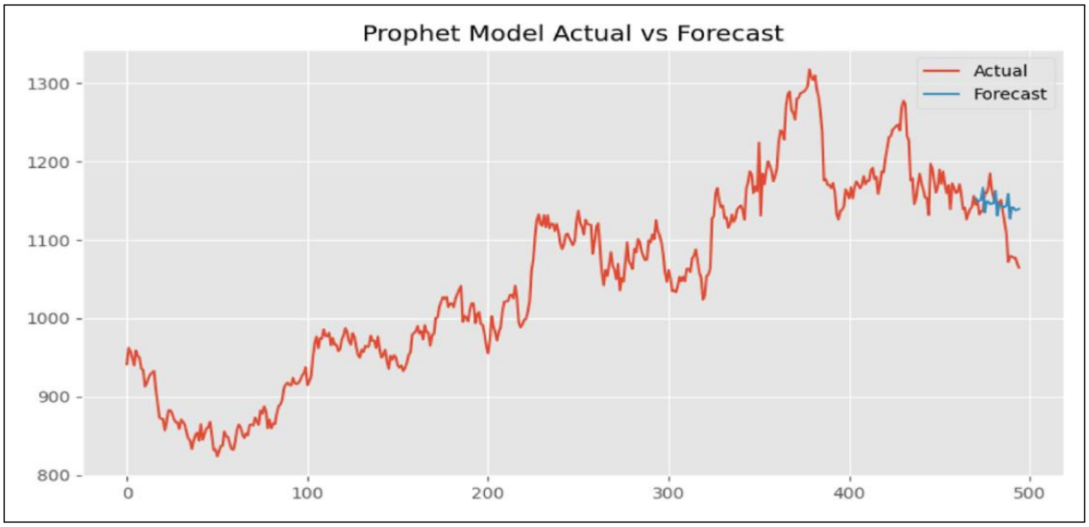


Figure 13: Forecast using Prophet with default parameters

Performance:

- MAE: 29.93
- RMSE: 39.07
- MAPE: 2.72

Observation: Even with default configurations, the Prophet model was able to detect and utilize patterns in the data far more effectively than the previous models. This demonstrates the robustness and generality of the Prophet framework in capturing seasonality and trend in financial data.

5.2.2 Prophet Model Tuned with Optuna

To further improve the model, we applied **Optuna,** a powerful hyperparameter optimization framework that uses Bayesian optimization (specifically, the Tree-structured Parzen Estimator - TPE) to explore the parameter space efficiently.

- **Optuna** automates the process of finding the best parameter values by maximizing/minimizing an objective function over multiple trials.
- In our case, the objective function was minimizing MAPE on a validation set.

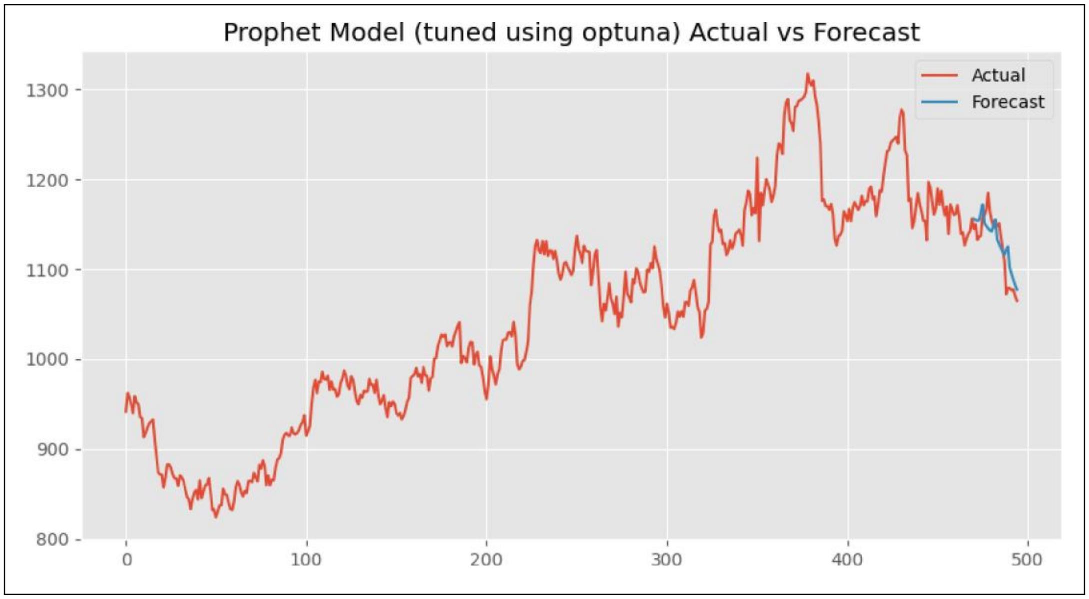


Figure 14: Forecast using Prophet tuned with Optuna

Tuned Parameters: The Prophet model was tuned using Optuna with the following hyperparameters:

- changepoint_prior_scale: 0.70
- seasonality _prior_scale: 2.611
- holidays_prior _scale: 6.6
- yearly _seasonality: 5.74
- seasonality _mode: multiplicative

Performance:

- MAE: 17.42
- RMSE: 21.12
- MAPE: 1.55

Conclusion: The Prophet model tuned via Optuna outperformed all previously used models-including ARIMA, SARIMA, and XGBoost-across all evaluation metrics. Its ability to flexibly model both trend and seasonality, combined with the power of automated hyperparameter tuning, makes it the most effective approach for our stock price forecasting task.

Chapter 6
Model Diagnostics and Reliability Assessment

To ensure that our forecasting model is not only accurate but also consistent and dependable, we carry out a series of diagnostic evaluations. This section examines the internal components of the tuned Prophet model and evaluates its forecasting stability using backtesting and cross-validation techniques.

6.1 Decomposition of Model Components

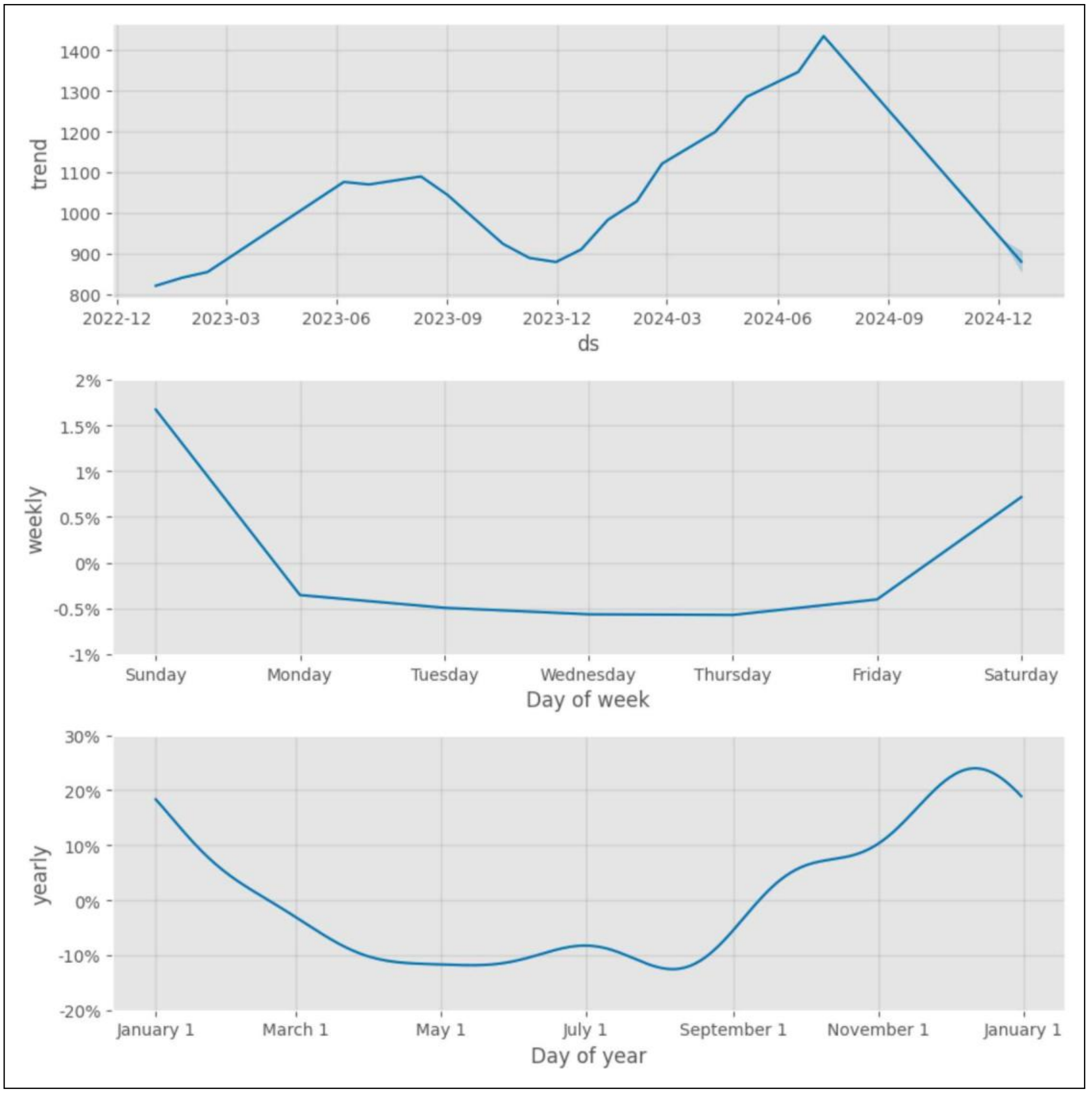


Figure 15: Trend and Seasonality Components of the Tuned Prophet Model

The Prophet model decomposes a time series into trend, seasonal (weekly and yearly), and holiday components. From the decomposition plot:

- The **trend** component reflects the overall direction of the stock price.
- The **weekly seasonality** captures intra-week patterns-likely influenced by market activity on specific weekdays.
- The **yearly seasonality** detects longer-term cyclical patterns such as quarterly financial reports or macroeconomic cycles.

These components help us interpret and understand the underlying structure in the time series and the factors influencing future predictions.

6.2 Back-Testing for Temporal Robustness

To evaluate the robustness of the tuned Prophet model over different time frames, we perform back-testing. The idea is to train the model on a historical segment of the data using fixed hyperparameters and then forecast the next 25 days. This process is repeated at different time points to assess consistency.

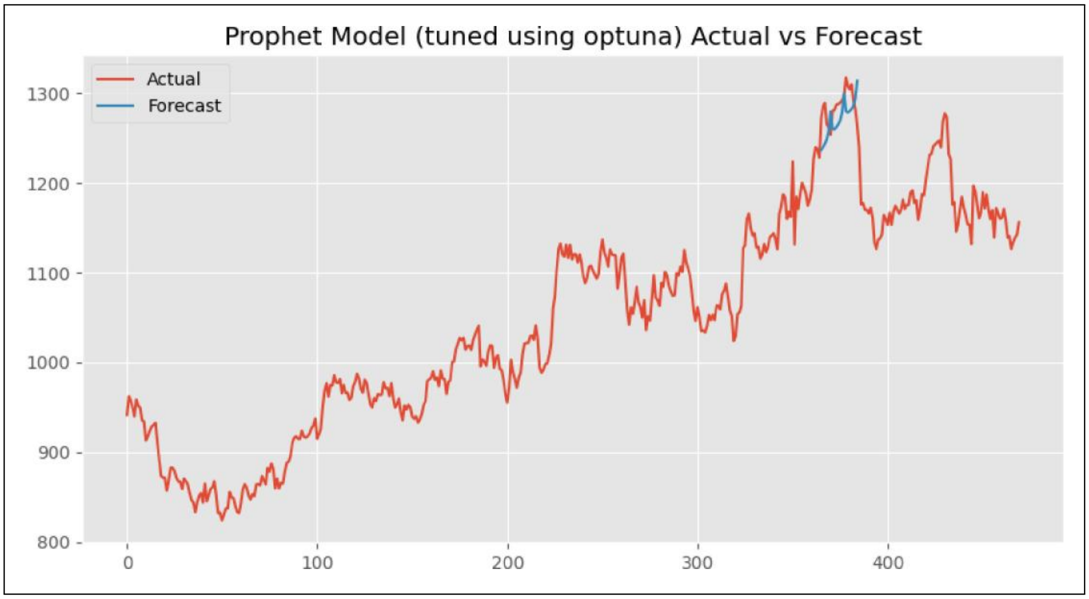


Figure 16: Back-testing Forecast - Period 1

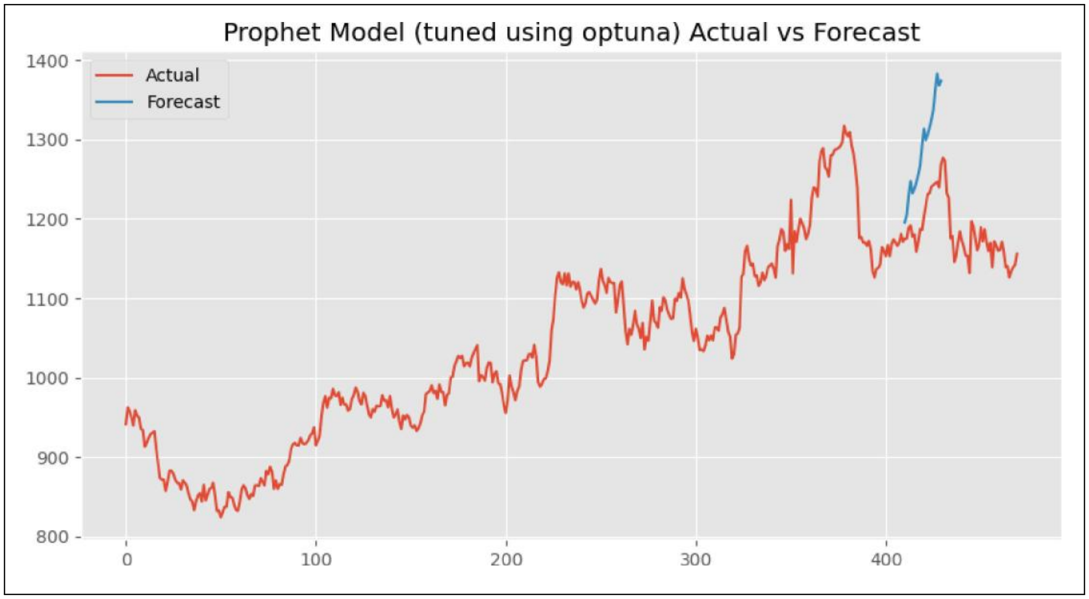


Figure 17: Back-testing Forecast - Period 2

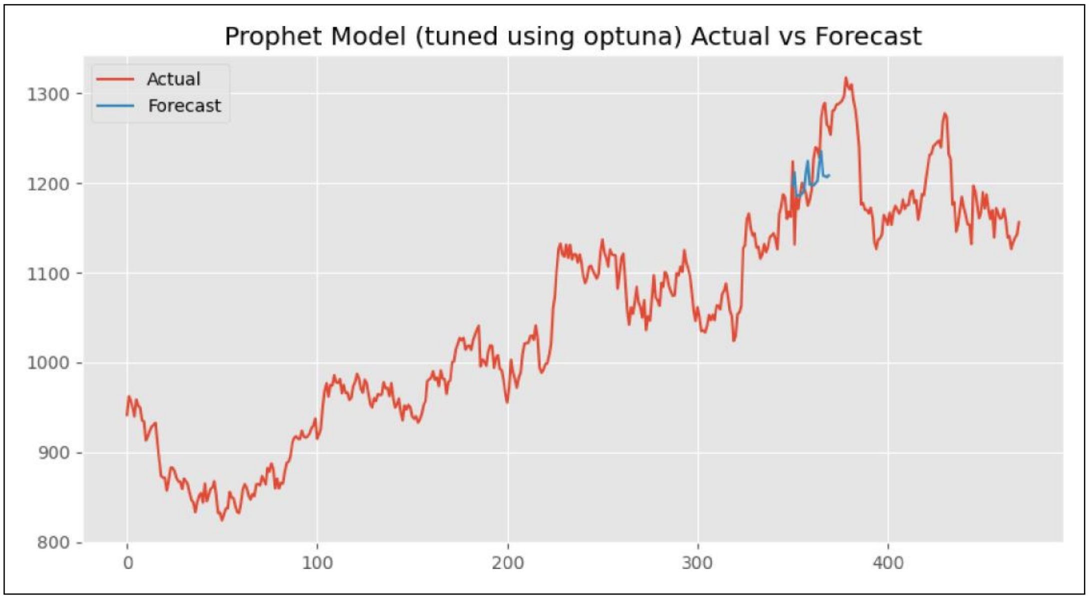


Figure 18: Back-testing Forecast - Period 3

Observations:

- Across all three test windows, the model is reasonably successful in identifying recurring patterns in the data.
- In the first scenario, the trend component is slightly overestimated.
- In the second, the model appears to capture short-term volatility effectively.

- The third back-test reflects balanced behavior, further validating the model's temporal generalization capabilities.

6.3 Cross-Validation for Forecast Horizon Analysis

To further assess forecast reliability, we perform cross-validation using Prophet's built-in utilities. The model is trained on a rolling window of past data and tested for forecast horizons ranging from 1 to 20 days.

Horizon	RMSE	MAPE
2 days	33.217293	0.025209
3 days	32.557687	0.024478
4 days	28.085512	0.020124
5 days	32.869936	0.023446
6 days	39.396445	0.031395
7 days	39.182966	0.031470
8 days	35.241813	0.026556
9 days	38.727409	0.029923
10 days	53.824822	0.044299
11 days	61.666670	0.048147
12 days	65.321264	0.050265
13 days	67.409813	0.053635
14 days	61.042026	0.046591
15 days	47.285536	0.033456
16 days	38.873875	0.025739
17 days	45.096941	0.030422
18 days	66.608069	0.050755
19 days	72.343611	0.056804
20 days	64.602382	0.048857

Table 2: Prophet Model Performance Over Forecast Horizons

Insights:

- The table summarizes the model's performance (in terms of RMSE and MAPE) over increasing forecast horizons.
- As expected, the model performs best for short-term forecasts (1-5 days ahead), with progressively higher error rates at longer horizons.
- This aligns well with the nature of stock markets, where short-term movements are often more predictable than long-term trends.

These diagnostic checks confirm that the tuned Prophet model not only offers superior accuracy but also demonstrates robustness across different time periods and forecast windows.

Chapter 7
Generalization of Prophet Model

The primary goal of this section is to assess the generalization capability of the tuned Prophet model beyond the Axis Bank stock dataset. We believe that a robust forecasting model should not only perform well on a single dataset but also extend its performance to other stocks across diverse sectors.

To facilitate this and make the model accessible to a broader audience-including analysts, traders, researchers, and even layman users-we deployed the model using the Streamlit framework. This deployment not only enables technical evaluations but also demonstrates the practical utility and ease-of-use of our approach in real-world forecasting tasks.

Why Streamlit Deployment?

- To evaluate the Prophet model's performance on different stocks from varied sectors in an interactive manner.
- To enable users without programming expertise to upload their own data and obtain accurate forecasts.
- To ensure usability and interpretability through intuitive design and rich visualizations.

In the following subsection, we present the architecture and functionality of the deployed application. This is followed by case studies on ICICI Bank, TCS, and L&T, demonstrating the model's generalization capability through this app.

7.1 Model Deployment via Streamlit App

App Link: Click here to access the Streamlit app

To bring the forecasting pipeline to users in a clean and functional interface, we developed a web-based app using Streamlit. The app abstracts all the technical complexities involved in time series modeling and provides a seamless experience to the user.
Main Components of the App:

- **Data Upload Panel:** Users can upload their own historical stock data in CSV format for forecasting.
- **Manual Hyperparameter Selection:** A sidebar interface allows users to set key Prophet parameters manually, such as changepoint prior scale, seasonality mode, and seasonality prior scale.
- **Automated Hyperparameter Tuning:** A dedicated button enables users to trigger Optuna-based hyperparameter optimization for improved model performance.

- **Forecast Generation:** After model fitting, the app can forecast stock prices for the next 25 business days using a single click.
- **Forecast Plot:** The forecasted values are plotted with historical data to visualize future trends.
- **Model Evaluation:** Key metrics such as RMSE, MAE, and MAPE are displayed to help users assess the model's prediction accuracy.
- **Interactive UI and Guidance:** Info buttons and warnings are placed near inputs and outputs to guide users and highlight potential issues in data or configuration.

This user-friendly interface bridges the gap between technical forecasting models and non-technical users, making it a valuable tool for real-world financial forecasting.

7.2 Forecasting Other Stocks Using the Deployed App

To evaluate the generalization capability of the Prophet model, we tested the deployed forecasting app on stock data from three companies across different sectors-ICICI Bank (Finance), TCS (Information Technology), and L&T (Infrastructure and Engineering).

For each company, we used the deployed application to forecast the next 20 trading days. The same tuned parameters from the Axis Bank model were used to ensure consistency. The following figures show the forecasted stock prices, along with the trend and seasonality components, as generated by the application.

7.2.1 ICICI Bank

- Sector: Finance
- Data Source: NSE
- Forecast Horizon: 25 trading days
- **Metrics: RMSE** = 24.75, **MAE=** 18.99, **MAPE** = 1.46

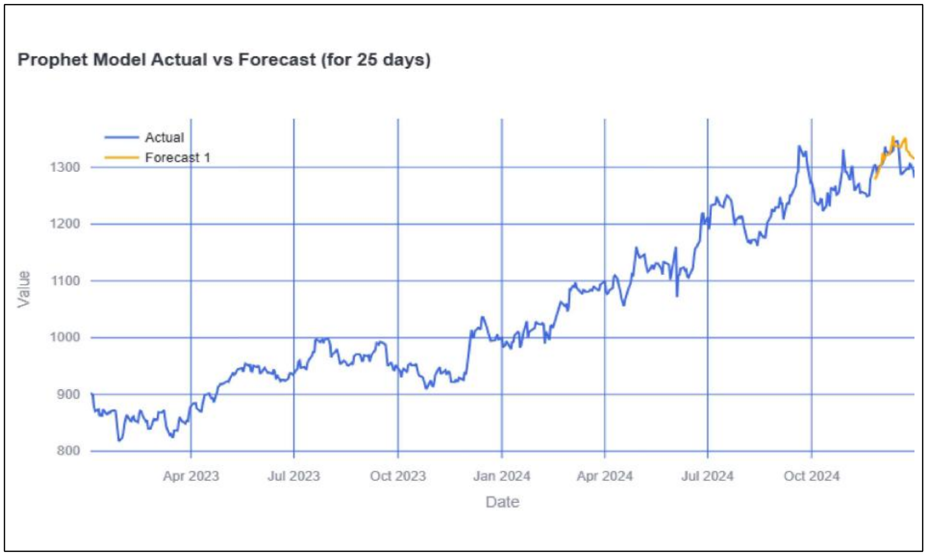


Figure 19: Forecast for ICICI Bank using the deployed Prophet model

7.2.2 TCS

- Sector: Information Technology (IT) / Consulting
- Data Source: NSE
- Forecast Horizon: 25 trading days
- **Metrics: RMSE** = 121.55, **MAE=** 102.79, **MAPE** = 2.41

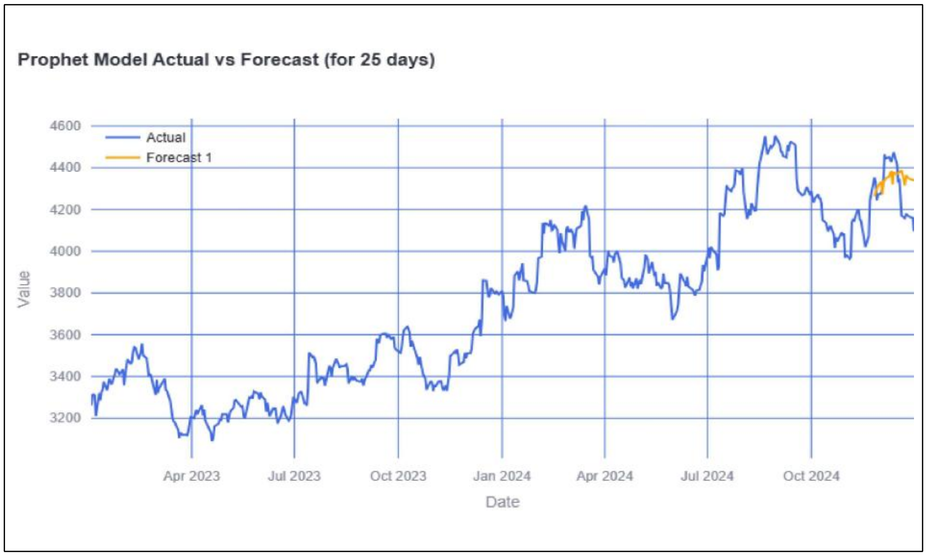


Figure 20: Forecast for TCS using the deployed Prophet model

7.2.3 L&T

- Sector: Engineering and Construction
- Data Source: NSE
- Forecast Horizon: 25 trading days
- **Metrics: RMSE** = 185.23, **MAE** = 132.31, **MAPE** = 3.6

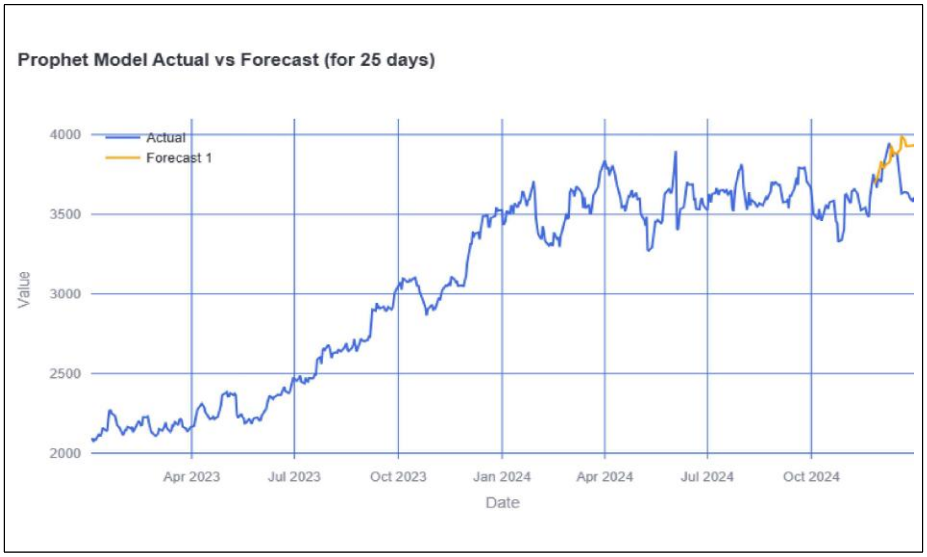


Figure 21: Forecast for L&T using the deployed Prophet model

7.3 Evaluation and Future Directions

In this section, we explored the generalization capabilities of the Prophet model by ap- plying it to stock data from various companies across different sectors. While the model performed well for most stocks, such as ICICI Bank (MAPE = 1.46), it showed slightly higher errors for others, like TCS (MAPE = 2.41) and LT (MAPE = 3.6). This indicates that while Prophet provides reliable forecasts for certain stock datasets, its performance may vary depending on the characteristics of the data.

To improve the framework's robustness, future work will focus on integrating additional time series models, such as ARIMA and machine learning-based approaches like XGBoost or LSTM, to offer more accurate forecasts across diverse stock data. This will allow users to choose the best-suited model based on the specific nature of the data.

By enhancing the framework with more forecasting techniques, we aim to provide a versatile tool for financial forecasting and extend its applicability to other domains with temporal data.

Chapter 8
Conclusion

In this project, we embarked on the challenge of forecasting stock prices using a combination of classical and advanced time series models. Starting with Axis Bank's daily stock data, we conducted detailed exploratory analysis to understand its structural properties-trend, seasonality, and volatility. This informed our decision to experiment with a suite of forecasting models including ARIMA, SARIMA, XGBoost, and Facebook Prophet.

Among these, the Prophet model especially when fine tuned using Optuna demonstrated superior forecasting accuracy and robustness. Its ability to handle non-linear trends, multiple seasonalities, and changepoints made it particularly suitable for financial time series data. The evaluation metrics (MAE= 17.42, RMSE = 21.12, MAPE = 1.55) confirmed its performance advantage over other approaches.

To test the generalizability of our approach, we extended the use of the Prophet model to stock data from different sectors including finance (ICICI Bank), information technology (TCS), and infrastructure (L&T). While performance varied slightly across stocks, the overall forecasting capability remained strong, validating the model's adaptability.

Going beyond academic exploration, we deployed the entire forecasting pipeline using a Streamlit-based web application. This tool bridges the gap between technical model development and practical utility, allowing users-irrespective of their programming background-to upload their own stock data, fine-tune hyperparameters, and generate reliable forecasts interactively.

Future Work:

- Incorporate more forecasting models such as LSTM and hybrid ARIMA-XGBoost ensembles to improve performance across volatile stocks.
- Add support for multivariate time series modeling by integrating macroeconomic indicators or sector-specific variables.
- Enhance the application interface for real-time data feeds and automated report generation.

Overall, this work presents a scalable and interpretable framework for stock price forecasting, with strong potential for real-world deployment in both academic and professional settings.