Predict Stock Prices Using Time Series Analysis

Kuldeep Kumar Das
Amity Institute of Information Technology
Amity University Jharkhand, Ranchi, India
Email: kuldeep.das@example.com

Dr. Biresh Kumar
Amity Institute of Information Technology
Amity University Jharkhand, Ranchi, India
Email: biresh.kumar@example.com

***Abstract*—Financial markets are among the most volatile and unpredictable places in the modern world. To lower risks and make wise decisions, traders, investors, and financial experts are always looking for precise forecasts. Once thought to be entirely speculative, stock price prediction has evolved into a science as a result of advancements in data analytics. The goal of this research is to estimate stock prices for two important Indian financial institutions, and Bajaj Finance Ltd., using time series forecasting techniques, notably ARIMA (AutoRegressive Integrated Moving Average) and Facebook Prophet. The sentiment of the Indian economy is reflected in both of these, which are highly traded. The objective is to assess these models' predicting abilities using actual historical stock data and offer insightful commentary on their precision and applicability. The statistical model that works best with stationary time series data is the ARIMA model. It predicts future trends based on past values and errors. Facebook Prophet, on the other hand, is a forecasting model created by Meta (previously Facebook) for commercial applications. It is capable of handling outliers, missing data, and seasonality with ease. For this study, daily stock price data from 2018 to 2024 was obtained using Yahoo Finance. Before being fitted into the ARIMA and Prophet models, the data was cleaned, displayed, and examined using Python in Google Colab. MSE (Mean Squared Error), RMSE (Root Mean Squared Error), and MAE (Mean Absolute Error) are standard performance measurements of both models. The study discovered that while both ARIMA and Prophet were capable of making reasonably accurate predictions about changes in stock prices, each had advantages and disadvantages according on the type of data. Prophet performed exceptionally well in managing seasonality and long-term trend changes, whereas ARIMA performed well for short-term, smooth forecasts. The purpose of this paper is to showcase the value of data-driven decision-making in the financial industry and to give researchers, analysts, and students a practical grasp of stock forecasting approaches.**

***Index Terms*—Stock Market, Stock Price Prediction, Time Series Analysis, ARIMA Model, Facebook Prophet, Bajaj Finance, Financial Data, Python, Data Science, Google Colab, Google Drive, Exploratory Data Analysis**

I. Introduction

Stock market prediction is essential to risk mitigation, portfolio management, and investment planning in today's rapidly changing financial environment. Everyone wants to comprehend stock price trends in order to make judgments that could lead to financial gains or losses, regardless of experience level. Though markets are impacted by a wide range of factors, including economic policy, world news, political events, company performance, and even public opinion, forecasting stock prices is not a simple task. Time series analysis, a statistical method that focuses on comprehending and predicting data points gathered over time, is one of the most dependable approaches to solving this issue. Because time series models use historical data to forecast future trends, they are perfect for analyzing stock prices, which are subject to fluctuations based on prior patterns and movements. Examining and contrasting these two models' performance in terms of prediction accuracy and usability is the main goal of the study. A traditional method that has been in use for many years, the ARIMA model is well-known for its effectiveness in short-term forecasting when data is stationary. Conversely, Facebook Prophet, a more recent model created by Meta, is renowned for its robustness in capturing seasonality, ease of use, and capacity to manage outliers. It is intended for users with little to no statistical experience. This initiative promotes a data-driven decision-making approach in addition to showcasing the technical components of creating prediction models. Financial organizations and individual investors alike can reduce risk and increase profits by comprehending and predicting price patterns. Real-world stock data collecting is the first step in the process, which is followed by data cleaning and exploratory analysis before the models are put into use. The project's outcomes will show which model performs better at forecasting in various scenarios, providing valuable information for further study and practical applications.

II. Review of Literature

The science and art of stock price prediction have long captivated investors, financial experts, and scholars. The profession has changed throughout the years, moving from a preponderance of subjective judgment and intuition to data-driven, algorithm-based prediction models. The stock market has long been seen as unpredictable because of its intricate and non-linear structure, in which values change in response to a myriad of known and unknown causes. The first tool used by researchers to deal with this complexity was Time Series Analysis, a potent area of statistics created especially for examining consecutive data points gathered over time. In the early 1970s, George Box and Gwilym Jenkins created the ARIMA (AutoRegressive Integrated Moving Average) model, which is among the oldest and most widely used techniques in time series forecasting. ARIMA became well-known because of its methodical approach to detecting patterns, seasonality, and error correction in data that changes over time. According to Box and Jenkins' study, good forecasting depends on the data being stable, which means that its statistical characteristics, such as mean and variance, stay consistent throughout time. Since then, ARIMA has been widely used in areas such as sales forecasting, economics, environmental research, and stock price forecasting. The shortcomings of conventional models like ARIMA were made clear by the growth of digital data and social media. News events, international crises, and societal mood all have an impact on stock values, which frequently exhibit abrupt fluctuations and non-linear behavior that ARIMA is unable to adequately represent. More adaptable forecasting tools algorithms were made possible by this gap. Facebook Prophet, an open-source forecasting tool published by Meta in 2017, is one example of a contemporary invention. By enabling business analysts and data scientists to anticipate time series data without the need for in-depth statistical understanding, Prophet was created to make forecasting approachable to non-experts. Its automatic trend change identification, seasonality, and outlier and missing data handling are its strongest points. Prophet, in contrast to ARIMA, makes the assumption that the model structure is additive, meaning that factors such as seasonality, trends, and vacations are combined to get the final forecast. Because of this, it works especially well with stock data, which inherently exhibits uneven growth patterns and seasonal activity. It have also been investigated in the literature in addition to ARIMA and Prophet. Though they frequently call for additional processing power and meticulous hyperparameter tuning.

III. METHODOLOGY

A. Introduction

Any research endeavor is supported by its methodology, which provides a logical and systematic road map for addressing the given topic. The methodology used in this study, "Stock Price Prediction Using Time Series Analysis," is intended to forecast future stock values using historical data and validated statistical models. This project employs two very dependable models, ARIMA and Facebook Prophet, to forecast the stock values of and Bajaj Finance Limited. Data collection, preprocessing, exploratory analysis, model construction, evaluation, and prediction are among the structured processes that make up the methodology. Python programming is used to implement the project on a cloud-based Google Colab environment, guaranteeing computational power and accessibility.

B. Research Design

This project's research strategy is founded on quantitative research methodology. This kind of study deals with quantifiable data and uses computational, mathematical, or statistical methods to draw findings. The design consists of the following steps:

- Data Acquisition: collecting past stock price information from a trustworthy financial source.
- Data Preprocessing: Data preparation and cleaning for modeling.
- Exploratory Data Analysis (EDA): examining patterns and trends in data.
- Model Selection and Training: utilizing Facebook Prophet and ARIMA models.
- Model Evaluation: Measuring the accuracy of the predictions with statistical measures.
- Deployment: Predicting the future using the learned model.

This framework guarantees the study's continued applicability, transparency, and reproducibility.

C. Data Collection

Historical stock price data for Bajaj Finance Limited was obtained from Yahoo Finance, a reliable financial data platform, and stored as a CSV file ('BAJFINANCE.csv') on Google Drive for easy access in the Google Colab environment. The dataset includes daily trading data with the following attributes:

- Date: The trading date.
- Symbol: Stock ticker (e.g., BAJAUTOFIN).
- Series: Equity type (e.g., EQ).
- Prev Close: Previous day's closing price.
- Open: Opening price of the day.
- High: Highest price during the day.
- Low: Lowest price during the day.
- Last: Last traded price.
- Close: Closing price of the day.
- VWAP: Volume Weighted Average Price, the target variable for prediction.
- Volume: Number of shares traded.
- Turnover: Total value of shares traded.
- Trades: Number of trades executed.
- Deliverable Volume: Volume of shares delivered.
- %Deliverable: Percentage of shares delivered.

The data was loaded into a Pandas DataFrame for analysis using Python in Google Colab. The process involved:

1) Setting up the environment by importing necessary libraries ('numpy', 'pandas').
2) Mounting Google Drive to access the CSV file.
3) Loading the CSV file into a DataFrame and inspecting its structure.

D. Data Preprocessing

Raw stock data often contains gaps, anomalies, and noise that can affect the accuracy of any model. Therefore, preprocessing was conducted to ensure data quality. Key preprocessing steps included:

1) Handling Missing Values: Data is absent since stock markets are closed on weekends and public holidays. Forward fill (ffill) was used to fill in the missing numbers in order to preserve continuity.
2) Date-Time Indexing: To make the data time-aware for improved time series analysis, the Date column was transformed into a DateTimeIndex format.
3) Feature Selection: To simplify complexity, Open, High, Low, and Volume were eliminated, leaving only the Close Price for modeling.

4) Stationarity Check: For ARIMA modeling to work, the time series needs to be stationary. Stationarity was examined using the Augmented Dickey-Fuller (ADF) Test. Differentiation was used if the data was determined to be non-stationary.

E. Exploratory Data Analysis (EDA)

Exploratory Data Analysis was conducted to understand the dataset's structure and trends. The following steps were performed:

- Time Series Plotting: A line graph of the Volume Weighted Average Price (VWAP) was plotted to visualize price trends over time.
- Moving Averages and Volatility: Rolling means and standard deviations were computed for features such as High, Low, Volume, Turnover, and Trades over a 7-day window to identify trends and variability in the data.

F. Model Building

1) ARIMA Model: The ARIMA (AutoRegressive Integrated Moving Average) model is a traditional time series forecasting technique that combines three components:

- AR (AutoRegression): Uses the relationship between current and historical values.
- I (Integrated): Applies differencing to make the data stationary.
- MA (Moving Average): Uses past forecast errors to predict future values.

The 'autoarima()' function from the 'pmdarima' library was used to automatically select optimal parameters.

2) Facebook Prophet: Facebook Prophet is an open-source forecasting model developed by Meta for time series data with significant changepoints and seasonality. It uses an additive model:

$$y(t) = g(t) + s(t) + h(t) + \epsilon_t$$

where:

- $g(t)$ models the trend,
- $s(t)$ captures seasonality,
- $h(t)$ models holiday effects,
- ϵ_t is the error term.

Prophet is effective for financial time series like stock prices due to its ability to handle outliers, missing data, and abrupt trend changes.

G. Model Evaluation

The ARIMA model's performance was evaluated using the following metrics:

- Mean Absolute Error (MAE): Measures the average magnitude of errors.
- Root Mean Squared Error (RMSE): Penalizes larger errors more heavily.
- Mean Absolute Percentage Error (MAPE): Expresses error as a percentage for interpretability.

Visualizations were also generated to compare actual and predicted values.

H. Prediction and Deployment

The trained ARIMA model was used to forecast future stock prices, visualized through line charts comparing actual and predicted VWAP values for Bajaj Finance. These models can be integrated into web dashboards for real-time predictions by traders and analysts.

I. Miscellaneous Setup

Additional setup steps included installing required packages and suppressing warnings to streamline the analysis.

IV. Data Collection

A. Introduction

Any predictive model is built on data, but this is especially true for stock price predictions. For this endeavor, training models like ARIMA and Facebook Prophet requires historical, accurate, and trustworthy data. In the absence of high-quality data, even the most advanced models will yield incorrect findings. The collection, processing, and preparation of the data for the Bajaj Finance stocks for time series analysis are described in this chapter.

B. Data Source

Yahoo Finance, a reputable and well-known financial website, provided the stock data used in this project. Yahoo Finance provides extensive market data, including past stock values, which are available to the general public via their API. The Python module yfinance was used to retrieve the data, which was then stored as a CSV file for analysis in Google Colab.

C. Data Description

The project's database includes the daily stock prices of Bajaj Finance Limited, one of the top non-banking financial companies (NBFCs) in India, renowned for its trustworthy investors and solid market performance. The following attributes were collected:

- Date: The date of the stock's trading.
- Symbol: Stock ticker (e.g., BAJAUTOFIN).
- Series: Equity type (e.g., EQ).
- Prev Close: Previous day's closing price.
- Open: Opening price of the day.
- High: Highest price during the day.
- Low: Lowest price during the day.
- Last: Last traded price.
- Close: Closing price of the day.
- VWAP: Volume Weighted Average Price, the target variable for prediction.
- Volume: Number of shares traded.
- Turnover: Total value of shares traded.
- Trades: Number of trades executed.
- Deliverable Volume: Volume of shares delivered.
- %Deliverable: Percentage of shares delivered.

For the purpose of this project, the primary focus was on the VWAP as it reflects the average price weighted by volume, providing a robust target for prediction.

D. Data Collection Process

The data collection process was efficient and involved:

1) Library Setup: Importing ‘numpy‘ and ‘pandas‘ in Google Colab.
2) Google Drive Integration: Mounting Google Drive to access the ‘BAJFINANCE.csv‘ file.
3) Data Loading: Reading the CSV into a Pandas DataFrame and inspecting initial rows.

The code used is shown in Section 3.3.

E. Data Integrity

Stock data must be clean, consistent, and accurate. After collection:

- Missing dates (e.g., weekends, holidays) were handled during preprocessing.
- Data types were verified, with the Date column set as the index.
- Missing values in features like Trades and Deliverable Volume were addressed in preprocessing.

This ensured the dataset was suitable for time series modeling.

F. Data Storage

The data was stored in CSV format (‘BAJFINANCE.csv‘) on Google Drive, facilitating easy access and collaboration in the cloud-based Google Colab environment.

G. Tools and Technologies Used

- Python: For scripting data extraction and analysis.
- yfinance: For sourcing stock data from Yahoo Finance.
- Google Colab: For coding, model training, and visualization.
- Pandas: For data manipulation and preprocessing.
- Matplotlib: For data visualization.
- CSV: For structured data storage.

V. Analysis

A. Introduction

The core of each data science endeavor is the analysis phase. Any predictive model cannot be constructed until the data has been gathered, cleansed, and examined. This chapter examines the stock data for Bajaj Finance Limited using statistical tools and visualization approaches. In order to provide precise future projections, this analysis assists in revealing hidden patterns, trends, seasonality, and behaviors in the stock prices.

B. Exploratory Data Analysis (EDA)

One crucial phase in the analytical process is EDA. It facilitates exposing the dataset's features and examining it for irregularities, patterns, and connections. Bajaj Finance datasets underwent the following procedures.

1) Line Chart Visualization: Line charts were used to plot the close prices of both stocks.

- Compared to the index, the plot for Bajaj Finance displayed more pronounced price changes, indicating the company's expansion and more stock sensitivity.

This visual inspection helped confirm that the data is time-dependent and shows clear trends, validating its suitability for time series modeling.

2) Rolling Statistics: We computed the rolling mean and rolling standard deviation to examine the stock price's stability over time.

- A time series is deemed non-stationary if the rolling mean and standard deviation change over time.
- As anticipated in real-world stock price data, the rolling plots for both datasets indicated non-stationarity.

As a result of this finding, the data was transformed into a stationary series for ARIMA modeling using differencing.

3) Stationarity Testing: The datasets' stationarity was statistically confirmed using the Augmented Dickey-Fuller (ADF) Test.

- A high p-value suggested that there was non-stationarity in the data.
- The p-value decreased following the application of first-order differencing, indicating that the differenced data was now stationary.

In order to guarantee that the ARIMA model could be trained correctly, this was an essential step.

4) Decomposition Analysis: Decomposition was performed to separate the time series into:

- Trend: Long-term direction of the stock price.
- Seasonality: Repeating short-term cycles (if any).
- Residuals: Random variations and noise.

Bajaj Finance displayed more abrupt changes, which are more common for individual stocks than for indices.

C. Model Training and Analysis

Following EDA, the stock data was entered into Facebook Prophet and ARIMA, two forecasting models.

1) ARIMA Model Analysis: Using Auto ARIMA, the best-fit model parameters (p, d, q) were automatically selected. After training, the model's residuals were analyzed:

- A successful model fit was confirmed by residual plots, which displayed a random distribution around zero.
- During testing, the predicted and real values nearly matched, indicating that the model could reasonably reflect the underlying pricing dynamics.

2) Facebook Prophet Model Analysis: Prophet was trained on the same datasets.

- Outliers and variations in trends were automatically identified and handled by the model.
- It generated forecasts and confidence intervals, which made it simple to see how reliable the predictions were.
- Smooth predicted price lines with seasonality adjustments were shown in the forecast graphs for the Bajaj Finance.

Prophet's ability to deal with missing values and changepoints without manual tuning made it highly effective for stock price forecasting.

D. Model Performance Evaluation

Both models were evaluated using these performance metrics:

- Mean Absolute Error (MAE): calculates the average forecasted stock price error.
- Root Mean Squared Error (RMSE): Penalizes larger errors more than MAE.
- Mean Absolute Percentage Error (MAPE): provides a handy percentage representation of inaccuracies for stock price ranges.

Results:

- ARIMA performed slightly better on historical data.
- The prophet was better able to adjust to shifting market conditions and trends.

The significance of employing numerous models for reliable forecasting is demonstrated by this comparison research.

VI. Results, Findings & Outcomes

A. Introduction

This chapter shows the outcomes of applying time series forecasting models to Bajaj Finance Limited stock price data. The accuracy, insights, and predictive power of the ARIMA and Facebook Prophet models utilized in the study are described in this part following exploratory data analysis and model training.

B. Model Evaluation Metrics

To determine the effectiveness of each model, three key performance metrics were used:

- Mean Absolute Error (MAE): Measures the average size of the errors.
- Root Mean Squared Error (RMSE): Highlights larger errors more heavily.
- Mean Absolute Percentage Error (MAPE): Shows the error as a percentage of actual values.

To evaluate forecast reliability, these measures were computed for the Bajaj Finance datasets.

C. ARIMA Model Results

The ARIMA model, after tuning parameters using Auto ARIMA, performed well on both datasets.

- For Bajaj Finance, ARIMA adapted to the stock's more volatile nature, producing close-fitting forecasts during the test period.

Observation: While ARIMA produced stable forecasts, its predictive power slightly dropped when sudden market changes or unexpected volatility occurred, especially for Bajaj Finance.

D. Facebook Prophet Model Results

The Facebook Prophet model turned out to be very automated and adaptable. It excelled at handling incomplete data, seasonality, and trend detection.

- The model produced smooth predictions with confidence intervals for the Bajaj.
- Prophet provided more flexible short-term projections for Bajaj Finance and responded to trend changes more effectively than ARIMA.

Observation: Prophet's constant and adaptable performance made it appropriate for stock price prediction assignments where data patterns are subject to frequent changes over time.

E. Comparative Findings

Criteria ARIMA Facebook Prophet Trend Detection Moderate to Strong Strong Seasonality Handling Manual (needs prior detection) Automatic Outlier Tolerance Low High Model Complexity Medium Low (user-friendly) Real-World Applicability Suitable for stable financial series Suitable for fluctuating series

F. Real-World Outcomes

The models trained in this project were able to:

- Forecast the direction of the stock price movement.
- Project possible future values for Bajaj Finance.

These insights can help fund managers, financial analysts, and retail investors make data-driven judgments instead of depending only on gut feeling.

G. Visual Outcomes

The prediction results were visualized using line graphs, where:

- Actual stock prices were plotted alongside forecasted prices.
- Confidence intervals were also displayed for Prophet, offering clear visual cues for expected ranges.

These illustrations demonstrated that, particularly for short-term forecasting horizons, both models closely resemble the initial stock price movements.

H. Key Findings

1) For structured and reasonably steady stock data, such as the Bajaj, ARIMA is dependable.
2) Facebook Prophet is more appropriate for specific equities like Bajaj Finance since it performs better on data that include trend shifts, outliers, or missing numbers.
3) In general, short-term forecasts have a higher predictive accuracy than long-term projections.
4) Following stationarity corrections and data cleaning, the model's performance greatly increases.

VII. Limitations & Delimitations (Scope of Study)

A. Introduction

No matter how sophisticated or carefully thought out, every study project has limitations. This chapter addresses the study's constraints and delimitations, or the elements that either were out of the researcher's control or were purposefully chosen to specify the project's parameters. A thorough comprehension of these aids in the realistic interpretation of the findings and directs future research in the appropriate path.

B. Limitations of the Study

Although Time Series Analysis techniques were successfully employed in this work to predict stock prices, some limitations that are common in financial modeling were observed:

1) Market Volatility: Global crises, news, political developments, and economic reports all have a significant impact on stock markets. Models like Facebook Prophet and ARIMA are based on historical data and are unable to anticipate abrupt, dramatic events that can lead to unpredictable price swings, such as pandemics, wars, or changes in policy.
2) Short-Term Prediction Accuracy: The models did well for short-term forecasts, but as the prediction horizon grew, their accuracy tended to decline. Since stock markets are naturally chaotic and non-linear over extended periods of time, forecast accuracy is limited to a few weeks or months.
3) Data Quality Dependence: The models' dependability depends on the quality of the data they use. The accuracy of the model may be impacted by missing numbers, improper stock splits, or unadjusted dividend data. Real-world stock data always has some level of noise, even after thorough cleaning.
4) Stationarity Requirements: Real stock prices frequently do not naturally follow the stationary mean and variance across time required by models such as ARIMA. The underlying price patterns may be distorted if the necessary differencing and transformation stages are not carried out correctly.
5) Model Simplicity: This study concentrated on univariate time series forecasting, which makes predictions about future values based solely on the stock's previous price. However, a variety of factors, including the following, affect actual stock movements:
   - Interest rates
   - Inflation
   - Economic indicators
   - Company earnings reports

   These elements could have enhanced prediction performance but were not included in the models.

C. Delimitations (Scope of Study)

Due to the controlled and scholarly atmosphere in which this project was developed, the research was subject to some limitations. These deliberate boundaries served to keep attention on the core objective, which was to comprehend the fundamentals of time series analysis-based stock price prediction.

1) Selection of Stocks: The study only focused on:
   - Bajaj Finance Limited (Individual Stock)

   It is not possible to extrapolate the findings to all stocks, particularly in markets or industries with distinct volatility patterns.
2) Data Source: The yfinance Python package was used to obtain all of the data from Yahoo Finance. Despite being a trustworthy platform, Yahoo might not always match exchange-reported prices because of minor revisions or delays.
3) Models Used: Only two forecasting models were explored:
   - ARIMA
   - Facebook Prophet
4) Time Frame: A certain time period-typically a few years - was covered by the historical data used. The learning patterns of the models may have been different if the time period had been longer or had included more international economic events.
5) Focus on Closing Price: The project focused primarily on Closing Prices and did not include:
   - Open, High, Low prices.
   - Intraday or minute-level data.
   - External macroeconomic variables.

   For professional financial forecasting, this resulted in a model that was straightforward and easy to understand but lacking in depth.

VIII. Future Scope / Scope of Further Study

A. Introduction

Economic, social, and political factors make financial markets dynamic and constantly shifting. Although this study successfully demonstrated how to use the ARIMA and Facebook Prophet models to predict stock values, there is always room for improvement and further research. This chapter examines potential improvements, expansions, and future research directions that could boost the effectiveness and reliability of stock price prediction models.

B. Incorporating Multi-Feature Datasets

The current study was based solely on historical closing prices of Bajaj Finance. However, future research could:

- Incorporate technical indicators, such as MACD, RSI, and moving averages.
- Take into account macroeconomic variables including GDP growth rate, inflation, and currency rates.
- Add more stock indicators, such as volume, open, high, and low prices.

Prediction accuracy can be increased by giving models more context through multi-feature datasets.

C. Application of Real-Time Data

In this study, model training and prediction were done using Ditzy using historical data. Real-time stock price prediction could be the main emphasis of future systems, which would:

- Pull real-time data continuously from APIs or stock exchanges.
- Retrain and update the prediction models automatically.
- Give investors and traders up-to-date, dynamic forecasts.

Algorithmic trading platforms and sophisticated financial advice systems might be powered by such systems.

D. Sentiment Analysis Integration

Social media trends, news mood, and investor sentiment all have a significant impact on stock values. Future research might incorporate:

- Natural Language Processing (NLP) techniques.
- News Sentiment analysis scores.
- Articles, financial reports, and Twitter feeds.

It would be possible to predict market reactions to world events more accurately if sentiment data were incorporated into forecasting algorithms.

E. Enhanced Model Evaluation Techniques

Other advanced evaluation techniques, such as the following, could be employed in future research in place of merely depending on conventional error measures like MAE, RMSE, and MAPE:

- Diebold-Mariano Test for comparing predictive accuracy.
- Use walk-forward validation to model actual trading situations.
- Profit and Risk Analysis to assess forecasts' financial feasibility.

F. Expanding the Dataset Scope

The Bajaj Finance were the only two organizations on which this study focused. Future studies could look into:

- Multiple individual stocks from various industries.
- Global indices (Dow Jones, NASDAQ, FTSE).
- Markets for cryptocurrencies such as Ethereum and Bitcoin.

This would test the flexibility and scalability of predictive models across various asset classes.

G. Automation and Deployment

Developing fully automated stock prediction systems is a viable avenue for the future:

- Construct end-to-end pipelines for modeling, prediction, cleaning, and data extraction.
- Use tools such as Streamlit to deploy models for real-time visualization.
- Use cloud computing services like AWS, GCP, or Azure to host them for commercial-grade accessibility.

This would close the gap between theoretical frameworks and practical FinTech (financial technology) applications.

IX. Conclusion

Understanding and applying data science techniques to forecast Bajaj Finance Limited's stock prices in the future was the goal of the project " Predict stock prices Using Time Series Analysis." One of the most interesting and challenging applications of data science is stock market prediction since financial markets are so intricate, erratic, and volatile on a regular basis. This study focused on two popular methods, ARIMA and Facebook Prophet, and demonstrated how time series techniques may be used to analyze, model, and forecast historical stock data. Beginning with the use of Python's yfinance package to obtain accurate historical price data, a methodical approach was adopted throughout the project. Next came crucial data pretreatment steps like handling missing values, ensuring stationarity, and organizing the data for model compatibility. The work then focused on developing two different models. The ARIMA model, known for its mathematical rigor, was used to capture patterns in stationary datasets, while the Facebook Prophet model, developed by Meta, was used for its flexibility in handling trends, seasonality, and irregular events like vacations. The results confirmed that both ARIMA and Prophet could forecast future stock values based on historical trends. However, there were significant limitations to the models because it is hard to account for all of the external elements that affect the stock market with historical data alone. These factors include market speculation, investor emotions, global economic conditions, and political developments. Despite this, the models offer a strong basis for estimating trends and making short-term predictions. This endeavor not only enhanced technical proficiency in time series forecasting but also highlighted the usefulness of data-driven decision-making in financial analysis. The project also fostered an awareness of model constraints, which is crucial for any aspiring data scientist working in finance or doing stock market research. The initiative lays the groundwork for future studies into increasingly complex predictive models, such as hybrid ARIMA-ML models and sentiment-driven analysis, in order to increase forecasting accuracy and offer deeper insights. In conclusion, this project offers an academic and useful introduction to stock price prediction using time series analysis, while also opening the door for future research in the rapidly evolving field of financial data science. Additionally, it imparted valuable knowledge in critical thinking, evaluation, and model construction.

References

[1] Box, G. E. P., & Jenkins, G. M. (1970). *Time series analysis: Forecasting and control*. San Francisco, CA: Holden-Day.

[2] Taylor, S. J., & Letham, B. (2018). Forecasting at scale. *The American Statistician*, 72(1), 37–45. https://doi.org/10.1080/00031305.2017.1380080

[3] Hyndman, R. J., & Athanasopoulos, G. (2021). *Forecasting: Principles and practice* (3rd ed.). OTexts. https://otexts.com/fpp3/

[4] yfinance. (2023). yfinance: Yahoo! Finance market data downloader. *Python Package Index*. https://pypi.org/project/yfinance/

[5] McKinney, W., & others. (2010). pandas: Powerful Python data analysis toolkit. https://pandas.pydata.org/

[6] Smith, T. G., & others. (2017). pmdarima: ARIMA estimators for Python. https://alkaline-ml.com/pmdarima/