Short-Term Trading Strategy with Machine Learning: A SARIMA Model Approach

Introduction

The rapid advancements in machine learning and time series modeling have opened up new opportunities in the domain of stock market trading. This project explores a **short-term trading strategy** by leveraging the **Seasonal AutoRegressive Integrated Moving Average (SARIMA)** model, a powerful tool for time series forecasting. Combining statistical rigor with the computational strength of machine learning, this approach aims to predict short-term stock price movements and identify profitable trading opportunities.

The focus of this strategy is the **Nifty 50 Index**, a benchmark of the Indian stock market that represents the performance of 50 large-cap stocks across various sectors. The goal is to forecast stock prices for the next 15 trading days and evaluate the strategy's success based on predicted price movements. To ensure robustness, the project uses historical data spanning 530 trading days, applying hyperparameter tuning through the **Auto-ARIMA** library for optimal model configuration.

By analyzing successful and unsuccessful trades, this project highlights the potential and limitations of applying machine learning techniques like SARIMA in financial markets. It also introduces measures like stop-loss mechanisms and error filtering to mitigate risks, showcasing a balanced approach to trading. The results demonstrate the feasibility of using data-driven insights to craft a systematic and profitable trading strategy, offering a foundation for future refinements and applications in financial analytics.

Building the strategy

1. **Stock Selection**:
    a. All Nifty 50 stocks were included in the analysis.
2. **Short-Term Strategy**:
    a. A 15-trading-day short-term strategy was evaluated using the closing price data of these stocks across 58 distinct time frames spanning from January 2020 to October 2024
3. **Modeling Approach**:
    a. The **SARIMA model** was employed to model each stock's closing price.
    b. **Auto_arima** was utilized for hyperparameter tuning to identify the best model configuration for each stock.
4. **Data Preparation**
    a. The closing price data spanned 530 trading days for each stock.
    b. The data was split into three parts
    c. The first 500 points were used for training.

    d. The next 15 points were reserved for testing the forecasted data.
    e. The final 15 points were used as the evaluation set.
5. **Forecasting and Error Calculation**:
    a. After training the SARIMA model on the initial 500 points, the next 15-day prices were forecasted.
    b. The **Mean Absolute Percentage Error (MAPE)** was calculated to evaluate the forecast's accuracy.
6. **Final Forecasting**:
    a. The model was updated by including 15 additional data points
    b. The price was again forecasted for the subsequent 15 days.
7. **Result Compilation**:
    a. The actual prices and forecasted price changes for each 15-day period were recorded.
    b. These results were stored in a DataFrame to build the strategy.
8. **Comprehensive Analysis**
    a. The same steps were repeated for all 50 stocks across different time frames, and the data was consolidated for building the strategy.

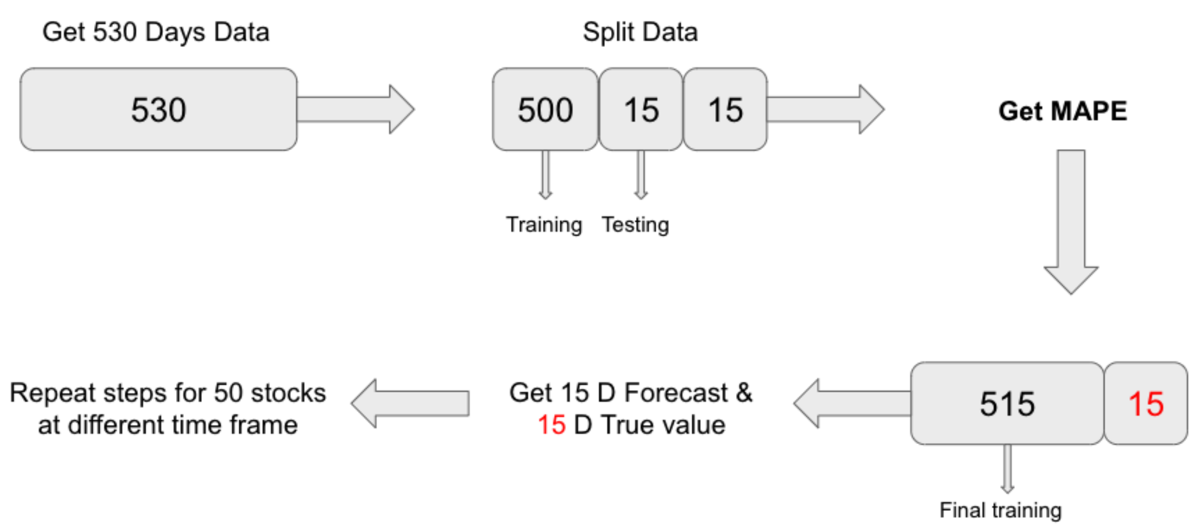

Fig 1:- Work flow of modeling

Finalizing the Strategy

The final steps to refine and evaluate the trading strategy are as follows:

1. **Filtering Based on MAPE**:
    a. From the dataset, filter out cases with low **Mean Absolute Percentage Error (MAPE)** to retain models with high forecasting accuracy.
2. **Identifying Favorable Scenarios**:

    a. Identify instances where the **forecasted price on Day 15 (P_D15)** is relatively higher compared to the Day 0 price (D0).
    b. These cases indicate potential opportunities for a successful trade.
3. **Performance Metrics**:
    a. Calculate the **total number of cases**, the **success percentage**, and the **average return** for these filtered instances.
4. **Implementing Stop Loss**:
    a. Apply a stop-loss threshold of **5%** to limit potential losses.
    b. Calculate the success percentage and average return after incorporating the stop-loss mechanism.
5. **Analyzing Results**:
    a. The results are summarized in a table showing cases for different MAPE values and forecasted price levels (P_D15).

	mape	P_D15	Total_Trade	Succ_Per	Avg_Return
0	<3	>5	148	62.84	2.54
1	<2.5	>5	78	61.54	2.54
2	<2.2	>4	82	60.98	2.34
3	<2	>4	48	60.42	1.77
4	<3	>4.5	188	60.11	2.27
5	<3	>4	233	59.66	2.26
6	<2.2	>5	44	59.09	2.48
7	<2	>4.5	34	58.82	1.34
8	<2.2	>4.5	63	58.73	2.30
9	<2.5	>4	138	57.97	2.25
10	<2.5	>4.5	106	57.55	2.09
11	<2	>5	21	57.14	0.80

Fig 2:- Summary of cases by MAPE Values and Forecasted Price Levels

6. **Key Observations**
    a. For cases where **MAPE < 3** and **P_D15 > 5**, there are **148 instances** out of a total of **2900 cases**, representing **5.1% of the total cases**.
    b. The **average return** for these instances is **2.54%**.
7. **Annualized Return Projection**:
    a. Considering data for 58 different time frames, more than **2 trades per day** can be identified.

    b. Assuming **250 trading days in a year**, the total annual return (non-cumulative) is projected to be approximately **42.33%**.

This structured approach outlines the strategy's potential and highlights the significance of filtering based on accuracy and price forecasts for optimizing returns.

Analyzing Successful and Unsuccessful Forecasts

Successful Case: Tech Mahindra Limited ("TECHM.NS")

1. **Data Overview**:
    a. **Training Data**: 500 trading days prior to September 1, 2024.
    b. **Hyperparameter Tuning**:
        i. Non-seasonal order: p=0 to 5, q=0 to 5, d=1.
        ii. Seasonal order: P=0 to 5, Q=0 to 5, D=1, m=5 (weekly seasonality).
    c. **Stationarity**: First differencing (d=1) was applied as the data became stationary, verified using the **Augmented Dickey-Fuller** test (p<5%).
2. **Model Parameters**:
    a. Best non-seasonal order: (p=0,d=1,q=0).
    b. Best seasonal order: (P=5,D=1,Q=0,m=5).
3. **ACF and PACF**
    a. The ACF and PACF plots of first differenced data for 40 lags showed no significant spikes, justifying p=0 and q=0

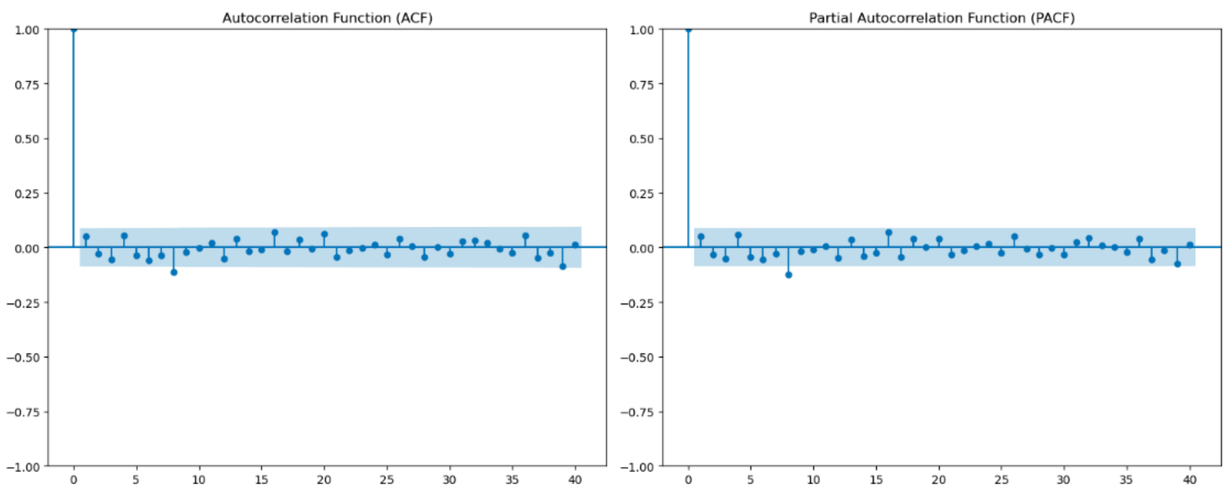

Fig 3:- ACF and PACF for 40 lags

4. **Performance**:
    a. **MAPE**: 2.55%.
    b. **Forecast**: 15-day price predicted as 4.57% higher than Day 0 price.
    c. **Outcome**: Actual 15th-day price change was 4.34%, yielding a profit of 4.34%.

d. The forecast closely matched the actual price, indicating successful prediction accuracy.

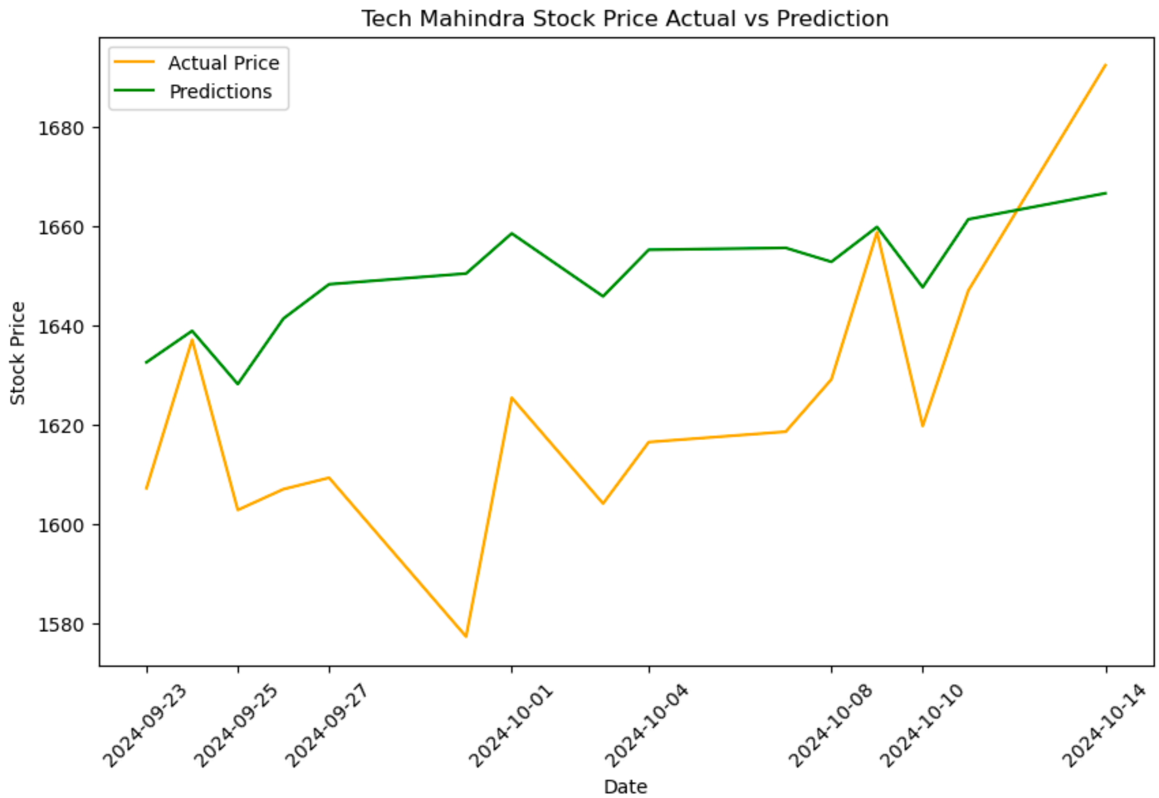

Fig 4:- Actual vs Predicted Price of Tech Mahindra

Unsuccessful Case: Coal India Limited ("COALINDIA.NS")

1. **Data Overview**:
   a. **Training Data**: 500 trading days prior to October 1, 2024.
   b. **Hyperparameter Tuning**:
      i. Non-seasonal order: p=0 to 5, q=0 to 5, d=1.
      ii. Seasonal order: P=0 to 5, Q=0 to 5, D=1, m=5 (weekly seasonality).
   c. **Stationarity**: First differencing (d=1) confirmed stationarity using the Augmented Dickey-Fuller test (p<5%).
2. **Model Parameters**:
   a. Best non-seasonal order: (p=0,d=1,q=1).
   b. Best seasonal order: (P=5,D=1,Q=0,m=5).
3. **ACF and PACF**
   a. The ACF and PACF plots of first differenced data for 40 lags showed a spike at lag 1, justifying q=1.

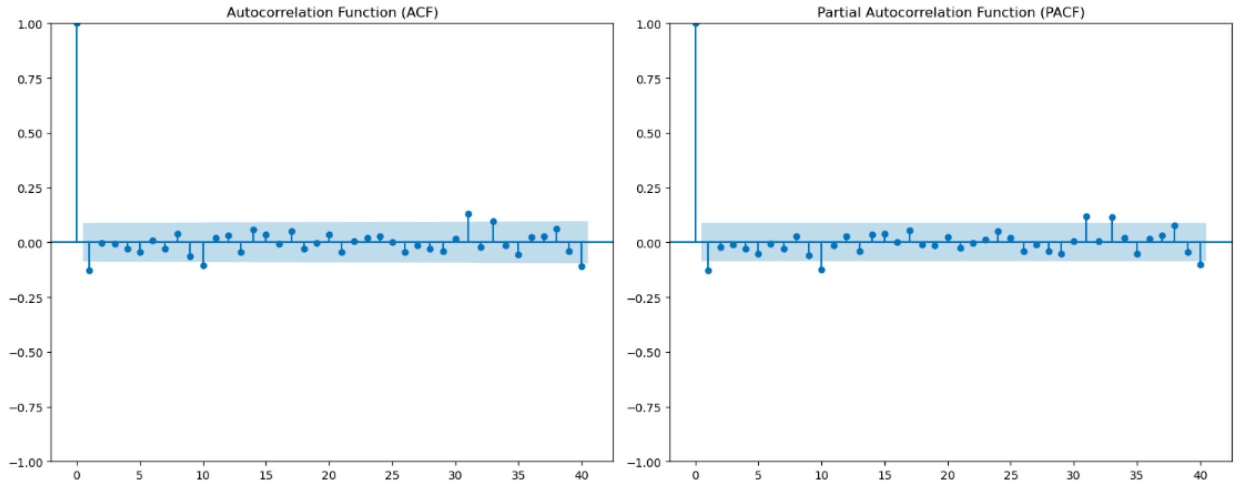

Fig 5:- ACF and PACF for 40 lags

4. **Performance**:
    a. **MAPE**: 1.95%.
    b. **Forecast**: 15-day price predicted as 4.54% higher than Day 0 price.
    c. **Outcome**: Actual 15th-day price change was -14.17%. The trade would have triggered a stop loss of -5%.

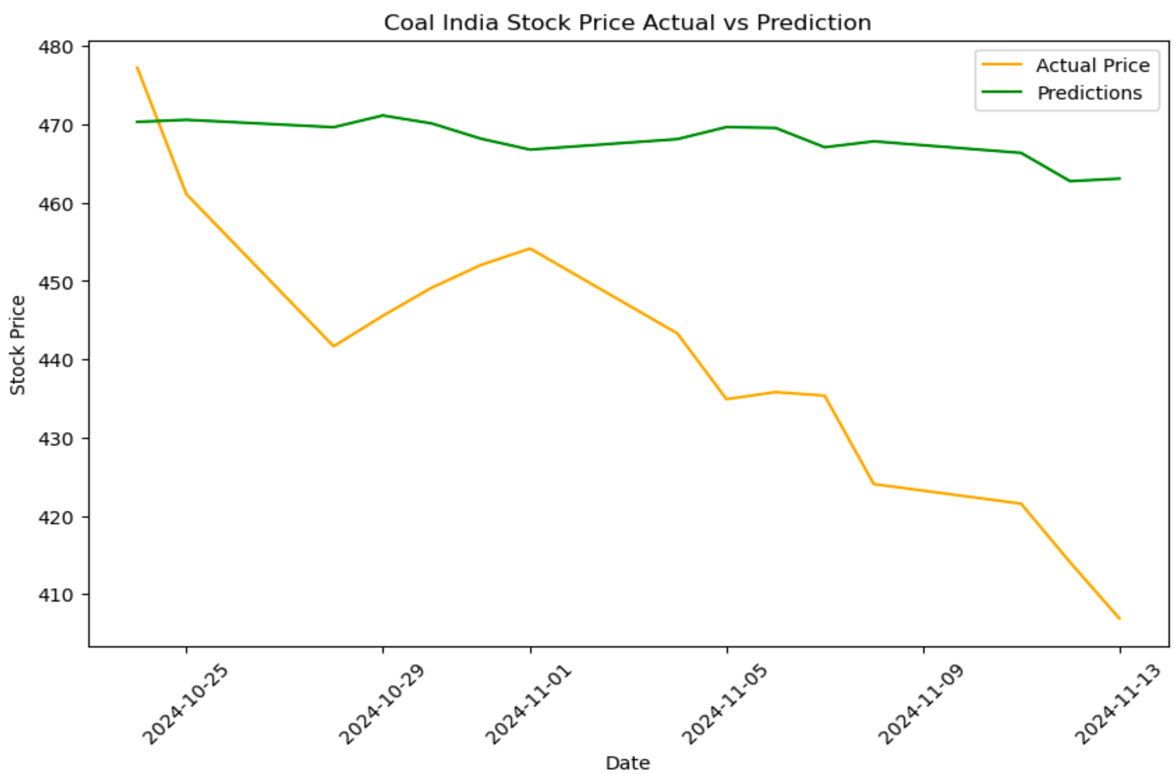

Fig 6:- Actual vs Predicted Price of Coal India

d. The significant deviation between forecast and actual price could be attributed to external factors like negative market sentiment, as the overall index was down during that period.

Observations:

1. While the strategy performed well for some stocks (e.g., Tech Mahindra), it failed in cases like Coal India due to unforeseen market conditions.
2. For scenarios where **MAPE < 3** and **P_D15 > 5**, approximately **37.16% of trades** were unsuccessful, highlighting the importance of incorporating broader market indicators into the model to improve predictive accuracy.

Conclusion

This project demonstrates the effectiveness of leveraging a SARIMA-based model for a short-term trading strategy in the Nifty 50 market. By systematically analyzing stock price trends and forecasts:

1. **Accurate Predictions**: Filtering cases with low Mean Absolute Percentage Error (MAPE) ensured that only high-confidence forecasts were considered, increasing the reliability of the strategy.
2. **Profitability**: Identifying scenarios where the 15-day forecasted price significantly exceeded the initial price helped isolate profitable trades, yielding an average return of 2.54%.
3. **Risk Management**: Incorporating a 5% stop-loss mechanism effectively managed downside risks, maintaining a balance between potential rewards and losses.
4. **Annual Returns**: With two trades available per day based on the strategy, the projected annual return of **42.33%** (non-cumulative) highlights the potential for significant profitability in the long term.

This study validates the feasibility of using time series models like SARIMA for trading strategies, especially in dynamic markets like Nifty 50. However, it is essential to consider market conditions, transaction costs, and external factors when implementing the strategy. By refining the model further and incorporating additional features like volume or macroeconomic indicators, the strategy could be enhanced for even better performance.