Predicting Stock Prices: A Machine Learning Approach to Time Series Analysis

K.W.Helitha Nimnaka
Intern-Software Engineering

DirectFN Sri Lanka (D F N Technology)
April 15, 2024

1 Introduction

The fluctuation of stock prices on a daily basis is influenced by a myriad of factors, ranging from market sentiment to economic indicators. For investors, the ability to accurately predict future stock prices holds the promise of lucrative investment opportunities. Armed with foresight, investors can strategically allocate their funds to capitalize on anticipated market movements, maximizing profitability. Similarly, for companies, gaining insight into future price trends is invaluable, enabling informed decision-making and proactive risk management strategies. In this context, the quest to forecast stock prices accurately emerges as a pivotal pursuit, with the potential to significantly impact both individual investors and corporate entities.

2 Traditional Approach vs. Machine Learning

Traditionally, forecasting stock prices has relied heavily on statistical methods, where mathematical equations are applied to historical data to generate predictions. While these approaches provide a foundation for analysis, they often fall short in capturing the complex and nonlinear relationships inherent in financial markets. As a result, the accuracy of predictions generated through traditional methods may be limited, particularly in volatile and dynamic market conditions.

In contrast, the emergence of machine learning techniques has revolutionized the field of stock price prediction by offering a more sophisticated and adaptive approach. Machine learning algorithms, such as neural networks and ensemble methods, possess the capability to learn from vast amounts of historical data, identifying intricate patterns and trends that may elude traditional statistical models. By leveraging advanced computational techniques, machine learning models can uncover hidden insights and make more accurate predictions, even in the presence of noisy and nonlinear data.

Furthermore, machine learning algorithms have the flexibility to adapt and evolve over time, continuously refining their predictive capabilities as new data becomes available. This adaptability is particularly advantageous in financial markets, where conditions can change rapidly and traditional models may struggle to keep pace.

In essence, the transition from traditional statistical methods to machine learning represents a paradigm shift in stock price prediction, offering the potential for greater accuracy and robustness in forecasting future market trends.

3 Algorithms

Those are the algorithms that I have implemented to achieve this target.

3.1 ARIMA (Autoregressive Integrated Moving Average)

ARIMA is a popular choice for modeling stock prices. It captures both autoregressive (past values) and moving average (residual errors) components.

Parameters:
- p: Number of lag observations (lag order).
- d: Degree of differencing (to make the time series stationary).
- q: Size of the moving average window (order of the moving average).

ARIMA is a statistical model and not a machine learning model. It relies on statistical methods to analyze and forecast time series data.

3.2 SARIMA (Seasonal Autoregressive Integrated Moving Average)

SARIMA extends the ARIMA model to handle seasonal patterns in time series data. It's particularly useful for stock price forecasting, where seasonal trends often play a significant role.

Key Components:
- AR (Autoregressive): Similar to ARIMA, it models the relationship between an observation and its lagged values.
- I (Integrated): Differencing to make the time series stationary.
- MA (Moving Average): Captures the dependency between an observation and a residual error from a moving average model.
- Seasonal AR (SAR): Incorporates seasonal lags (similar to AR) to account for periodic patterns.
- Seasonal I (SI): Seasonal differencing to remove seasonality.
- Seasonal MA (SMA): Seasonal moving average terms.

Parameters:
- p: Autoregressive order (lags for AR).
- d: Degree of differencing (to achieve stationarity).
- q: Moving average order (lags for MA).
- P: Seasonal autoregressive order (seasonal lags for SAR).
- D: Seasonal differencing order (seasonal differencing for SI).
- Q: Seasonal moving average order (seasonal lags for SMA).
- s: Seasonal period (e.g., 12 for monthly data with yearly seasonality).

SARIMA models are powerful tools for capturing both short-term and long-term patterns in stock prices. It's important to note that SARIMA is a traditional statistical method and does not involve any machine learning techniques. It can be implemented using the `statsmodels` library in Python, which provides robust functionality for time series analysis and forecasting.

3.3 Facebook Prophet:

Facebook Prophet is a versatile time series forecasting tool developed by Facebook's Core Data Science team. It seamlessly combines statistical modeling with machine learning principles to provide accurate and interpretable forecasts for stock prices. By capturing both linear and non-linear trends, seasonal patterns and custom events such as holidays, Prophet offers a comprehensive approach to modeling time series data. Its automatic changepoint detection, uncertainty estimation and scalability make it a popular choice for analysts and researchers in finance and beyond. With its user-friendly interface and ability to handle missing data and outliers, Prophet empowers users to make informed decisions based on reliable forecasts.

3.4 Regression Models:

These models offer diverse approaches for stock price prediction:

Linear Regression: Assumes a linear relationship between input features and stock prices. It seeks to fit a straight line to the data, making predictions based on the relationship between independent variables and the target variable.

XGBRegressor (Extreme Gradient Boosting): An ensemble model that combines multiple decision trees to improve predictive performance. It iteratively builds new trees to correct errors made by previous models, resulting in a highly accurate prediction.

Ridge Regression: A form of regularized linear regression that adds a penalty term to the loss function, aiming to shrink the coefficients towards zero. This helps to mitigate multicollinearity and overfitting, enhancing the model's generalization performance.

KNeighborsRegressor: Predicts stock prices based on similar historical data points. It relies on the assumption that data points with similar features have similar target values, making predictions by averaging the target values of the k-nearest neighbors.

Deterministic Process: The deterministic process introduced here involves creating a model that captures deterministic trends in the data. It is utilized to incorporate systematic patterns beyond simple linear relationships into the predictive model. This process is particularly useful when there are known deterministic factors influencing the target variable, such as economic indicators or predefined trends. In the described process, a `DeterministicProcess` object is created with specified parameters, including the index representing dates from the training data, the inclusion of an intercept term, the order of the trend (e.g., quadratic), and the handling of collinearity. The resulting in-sample predictors capture the deterministic component of the model, aiding in the prediction of stock prices by accounting for systematic trends in the data.

3.5 HMMs (Hidden Markov Models):

HMMs can capture hidden states in stock price movements. Useful for identifying regimes (e.g., bull vs. bear markets).

3.6 DNN (Deep Neural Network):

DNNs with multiple hidden layers can model complex relationships. Suitable for stock price prediction when combined with appropriate features.

3.7 CNN (Convolutional Neural Network):

Although primarily used for images, 1D CNNs can process stock price sequences. Useful for identifying patterns in historical data.

3.8 RNN (Recurrent Neural Network):

General architecture for time series prediction. Can learn from past stock price data to make future predictions.

3.9 GRU (Gated Recurrent Unit)

Overview: GRU is a variant of RNNs designed to address some limitations of traditional RNNs. It has a simpler architecture compared to LSTMs.
Architecture: GRUs consist of two gates, the Update Gate and the Reset Gate, which control the flow of information within the network. This design allows GRUs to manage information flow and mitigate the vanishing gradient problem.
Advantages: GRUs are computationally efficient due to their fewer parameters and excel at capturing short-term dependencies in sequential data.
Limitations: They may struggle with capturing long-term dependencies compared to LSTMs.

3.10 LSTM (Long Short-Term Memory)

Overview: LSTMs were introduced to address the vanishing gradient problem and allow modeling of long-term dependencies. They have a more complex architecture compared to GRUs.
Architecture: LSTMs include a cell state that can store information over long sequences. They consist of three gates: the Input Gate, the Forget Gate, and the Output Gate, which regulate the flow of information within the network.
Advantages: LSTMs excel at capturing long-term dependencies and are more robust in handling vanishing gradients compared to traditional RNNs.
Limitations: They are computationally expensive due to their higher number of parameters and may overfit on small datasets if not properly regularized.

4 Dataset

The dataset comprises approximately 400 custom data entries, each stored in separate CSV files. These files contain several columns representing different features; however, for our analysis, we only utilize the time and Last Traded Price (LTP) values.

5 Prediction Methodology

The prediction methods employed primarily involve forecasting future values based on historical data using algorithms such as ARIMA, LSTM and Prophet. A key aspect of these methods is the use of a sliding window approach. This approach involves selecting a window size and using the data within that window to predict the next value. As the window moves forward in time, the model continuously predicts new values. However, when predicting over longer time horizons, such as 365 days, using a short window may result in the model relying on previously predicted values, leading to potential error accumulation and increased loss. Therefore, selecting an optimal window size and fine-tuning model parameters become critical tasks. Additionally, due to the diverse nature of the datasets, finding common parameters across all datasets poses a significant challenge.

6 Feature engineering

Feature engineering plays a pivotal role in optimizing our predictive models. Through meticulous feature engineering, we've introduced several additional features to enrich the dataset and enhance predictive accuracy. These features include the three-day moving average (AVG), which calculates a rolling average of the Last Traded Price (LTP) values over a specified window size. We've also introduced upper and lower bands, derived from the rolling average and standard deviation, to capture volatility patterns in the data. Additionally, we've incorporated the half-window average (Half AVG) to provide insights into shorter-term trends. To capture temporal dependencies, lagged features such as lagged 1 step, lagged 10 step, and lagged

50 step have been introduced, representing the LTP values shifted by one, ten and fifty time steps, respectively. Furthermore, we've introduced the exponential moving average (EMA) to capture trend behavior over time. These carefully crafted features offer valuable insights into the underlying dynamics of the dataset and empower our models to make more accurate predictions. Moreover, after adding these features, we plan to utilize dimensionality reduction techniques like Principal Component Analysis (PCA) to further optimize our model by reducing the dimensionality of the feature space while retaining the most important information. This will help streamline our modeling process and potentially improve model performance by reducing noise and computational complexity.

7 Results

After conducting numerous tests, it was found that the LSTM (Long Short-Term Memory) model consistently yielded the most promising results compared to other algorithms. Its ability to capture long-term dependencies and intricate patterns in the data proved to be advantageous, resulting in superior forecasting accuracy for stock prices.

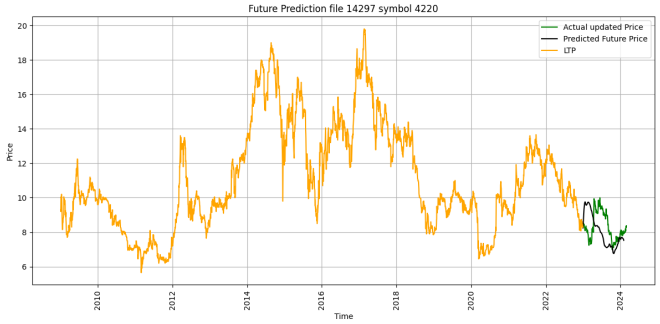

Figure 1: *Future Price Prediction - LSTM*

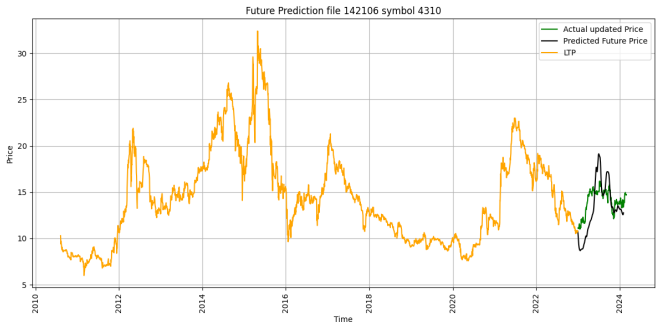

Figure 2: *Future Price Prediction - LSTM*

In some instances, historical data may fail to accurately predict future price trends. Models attempt to replicate historical patterns, but if those patterns do not repeat during the prediction period, or if the historical data is insufficient to capture the pattern, the models may yield less accurate results.

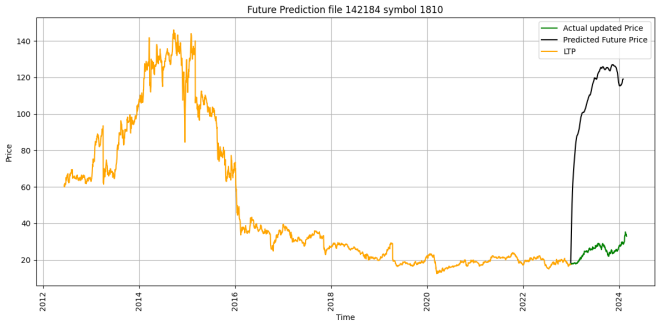

Figure 3: *Future Price Prediction - LSTM*

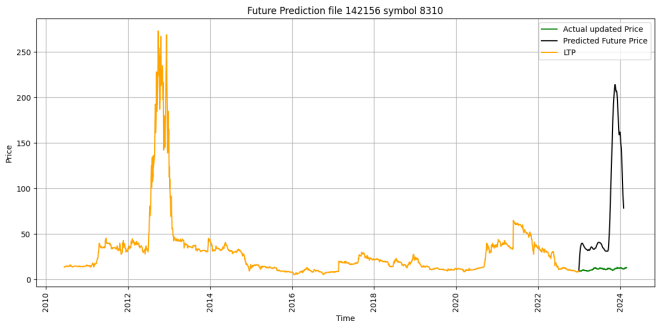

Figure 4: *Future Price Prediction - LSTM*

8 Conclusion

There are scenarios where historical data alone may not suffice to accurately predict future price trends. Factors such as sudden market shifts, geopolitical events, regulatory changes, and unexpected economic developments can all influence stock prices in ways that historical data may not fully capture. In such cases, relying solely on historical patterns for forecasting may lead to inaccuracies or incomplete assessments of market behavior. It underscores the importance of incorporating real-time information, qualitative analysis, and expert judgment alongside historical data to make more robust predictions and navigate uncertainties effectively in financial markets.

In conclusion, while LSTM models have shown promising results in stock price forecasting, it's important to recognize that stock prices are influenced by

various factors beyond historical data alone. The relationship between historical patterns and future stock prices may not always be straightforward or strong. Therefore, relying solely on historical data for accurate predictions may be insufficient. Moving forward, incorporating additional sources of information, such as natural language processing models, could enhance the forecasting capabilities by capturing non-linear relationships and nuanced market sentiments. By integrating diverse data sources and leveraging advanced analytical techniques, we can strive to improve the accuracy and reliability of stock price predictions in dynamic financial markets.

References

[1] *Stock Price Change Forecasting with Time Series: SARIMAX*. 2020. URL: https://towardsai.net/p/machine-learning/stock-price-change-forecasting-with-time-series-sarimax.

[2] *Time Series Forecasting with ARIMA , SARIMA and SARIMAX*. 2023. URL: https://towardsdatascience.com/time-series-forecasting-with-arima-sarima-and-sarimax-ee61099e78f6.

[3] *Multi-step Time Series Forecasting with ARIMA, LightGBM, and Prophet*. 2021. URL: https://towardsdatascience.com/multi-step-time-series-forecasting-with-arima-lightgbm-and-prophet-cc9e3f95dfb0.

[4] *APPLICATION OF FACEBOOK'S PROPHET ALGORITHM FOR SUCCESSFUL SALES FORECASTING BASED ON REAL-WORLD DATA*. 2020. URL: https://www.researchgate.net/publication/341071241_Application_of_Facebook's_Prophet_Algorithm_for_Successful_Sales_Forecasting_Based_on_Real-world_Data.

[5] *Stock Price Prediction by Normalizing LSTM and GRU Models*. 10(1S) 5326-5332. 2023. URL: https://sifisheriessciences.com/journal/index.php/journal/article/view/1875.

[6] *Research on Stock Price Prediction Based on PCA - LSTM Model*. ISSN 2616-5902 Vol. 4. DOI: 10.25236/AJBM.2022.040308. URL: https://francis-press.com/uploads/papers/uJ7lRxcP2AcjMC3VXJtlB5v7Yhf1JAvn8o6PcgTj.pdf.

[7] *Forecasting at Scale*. URL: https://facebook.github.io/prophet/.

[8] *Recurrent Neural Network Tutorial (RNN)*. URL: https://www.datacamp.com/tutorial/tutorial-for-recurrent-neural-network.

[9] *What is feature engineering?* URL: https://www.ibm.com/topics/feature-engineering.

[10] *What is feature engineering?* 21 Dec, 2023. URL: https://www.geeksforgeeks.org/what-is-feature-engineering/.

[11] *Principal Component Analysis(PCA)*. 6 Dec, 2023. URL: https://www.geeksforgeeks.org/principal-component-analysis-pca/.