**PREDICTING STOCK MARKET TRENDS USING TIME SERIES ANALYSIS AND MACHINE LEARNING**

Chidera Nwosu

002283696

Northeastern University

IE7945

30th June 2024

**Introduction**

Investors have been interested in predicting the stock market over the years. While its dynamic nature has made perfect foresight difficult, machine learning offers new landscape navigation tools. This project explores the potential of machine learning algorithms to predict stock price trends.

It builds on existing work that demonstrates combining machine learning techniques with optimization algorithms and time series analysis for improved accuracy. The aim is to explore various models, including ARIMA, XGBoosts and CNN. Using datasets spanning 4 to 5 years, we will investigate the efficiency of different machine-learning models to predict stock prices.

Acknowledging the limitations of the dataset, the period of 4/5 years might limit the model’s ability to capture longer-term trends, and the lack of external factors would also prove a challenge. But by acknowledging these limitations. We can use techniques like complexity measures to identify suitable periods for prediction and eventually incorporate other relevant data sources for future iterations.

Eventually, the project strives to contribute to the bountiful knowledge of machine learning for stock price prediction. Analyzation of model performance and financial gain, we aim to highlight the strengths and weaknesses of this approach, providing valuable insights for investors and contributing to the ongoing conversation about the role of machine learning in the constantly changing financial landscape.

**Annotated Bibliography**

Deepika and Nirupamabhat (2020) proposed a hybrid model that combined LSSVM, ABC optimization, and feature selection to predict stock prices. They achieved better accuracy and efficiency with a reduced feature set using Principal Component Analysis. Their findings suggested that this approach is practical for time series analysis tasks like predicting stock market trends. This is relevant to this project because it demonstrates the potential of combining machine learning with optimization techniques for improved time series forecasting.

Dezhkam et al. (2023) proposed a framework to predict stock market trends using machine learning. They used various classification models (SVM. GRU. LSTM, XGBoost) and a tri-state labelling algorithm to categorize price movements. Bayesian optimization was used to tune the hyperparameters for better performance. Backtesting showed promising results, with XGBoost achieving the highest Sharpe ratio compared to other models. This is relevant to this project because it combines time series analysis with machine learning models for stock trend prediction; their framework can be adapted, specifically the tri-state labelling and hyperparameter optimization techniques.

Khan et al. (2023) compared machine learning models for stock prediction. While traditional methods used daily closing prices, they found improved accuracy using the opening 15-minute price. This strategy boosted performance across all models, highlighting the importance of prediction accuracy and financial metrics when evaluating stock prediction models. This is relevant to this project as it explores various effective machine-learning models and data considerations for improved stock price prediction.

M et al. (2022) combined machine learning (LSTM networks) and time series analysis (Holt-Winters) for stock price prediction. They also used a recommendation system based on factors like predicted price and return on investment. This aligns well with the project as the techniques can be leveraged for effective stock price prediction.

Prasad and Seetharaman (2021) explored using machine learning to predict stock prices. They analyzed various algorithms and feature variables, and their findings suggested that deep learning models (LSTMs) outperform traditional machine learning models in accuracy. Reinforcement learning models were found to be the most profitable. This is relevant to the project as it highlights practical algorithms and exploration approaches.

Raubitzek and Neubauer (2022) analyzed historical stock data to understand its predictability. Their findings showed that recent data is more random, and complex measures were used to identify this. The results suggest that machine learning might be more effective in less complex market periods. It shows the potential of complex measures to identify suitable timeframes, highlighting their relevancy to the project.

Xiao and Su (2022) explored combining ARIMA and LSTM models for predicting stock prices. Their proposed model achieved better accuracy than individual models based on loss function evaluation. This is relevant to the project as it highlights the potential of combining ARIMA and LSTM for improving stock market trend prediction.

# Dataset Overview and Repository Details

**Dataset Overview**

**Stock Price Prediction**

The Yahoo Finance API grants users access to an extensive dataset containing publicly traded companies' historical stock prices and financial data. The dataset encompasses comprehensive data on various stocks. The dataset comprises multiple financial indicators, such as initial and final prices, daily maximum and minimum prices, trading volume, and additional metrics, which makes it a valuable asset for analysts. The dataset comprises historical stock data for all the stocks available in the Yahoo Finance API. Each file is named after its corresponding stock symbol. Each CSV file contains 10972 rows and eight columns (<https://data.world/gymprathap/stock-price-prediction>)

**Dataset Repository**

This dataset and related items were stored in GitHub: [thesire/Stock-Market-Prediction-Project (github.com)](https://github.com/thesire/Stock-Market-Prediction-Project)

# Methodology

**Data Collection**

After seeing a large number of datasets available, four in the financial sector were chosen: Aflac Incorporated (AFL), The Bank of New York Mellon Corporation (BK), U.S Bancorp (USB) and Welltower Inc. (WELL). Using the first two for model training and using the latter two for validation.

**Data Cleaning**

The data was directly sourced from the company’s database; after checking for missing values and realizing there were none, the “datetime” column was converted into date-time format and extracted into separate date and time columns. The year was extracted from the newly created date column, then the datetime and date columns were dropped. The data was filtered to produce rows from the last five years. A‘source’ column was created to identify where each data came from while preparing to merge both datasets; after the merge, the new dataset was sorted according to year.

**EDA(Exploratory Data Analysis)**

**Heatmap**

**A screenshot of a computer

Description automatically generated**

It showed strong positive correlations, and there was moderate positive between the year and stock prices, which meant that the stock prices were intended to increase over time.

**Line plot**

**A graph with blue lines

Description automatically generated**

This graph shows the closing stocks over time; after choosing the most recent years, we see a cyclic pattern with many noticeable peaks and troughs.

**Line Chart**

**A blue line graph on a white background

Description automatically generated**

The stock exhibits volatility, showing constant price fluctuations; the graph's spikes also indicate rapid price changes over periods.

**Code Breakdown**

The “Close” column was selected for this feature metric due to its significant market sentiment indicator. It is also consistent and is reported daily; its consistency is a requirement for the time series analysis.

***ARIMA (Auto Regressive Integrated Moving Average)***

This model is known for excelling at identifying trends and seasonal patterns in time series data. By addressing these elements directly, ARIMA helps remove trends and seasonality, making the data more stable and suitable for machine learning algorithms. The ARIMA model was fitted, and new features were created (lagged features, etc.) and used to train the machine learning models.

***XGBoost and CNN (Convolutional Neutral Network)***

XGBoost is excellent at finding complex patterns in data, which is why I chose this model; it also helps determine the most important features in making predictions. CNN is excellent at spotting spatial patterns and dependencies in the data, making them useful for time series with local correlations. They can automatically extract important features from the input; they also handle time series data very well.

Lagged values were created, and training sets were implemented using the XGBoost model. The MinMaxScaler was important in ensuring that all the features contribute equally to the learning process, and to help the neutral network converge faster, the creation of sequence was necessary for the CNN to recognize patterns over time. The data is split into training and test sets; up to 80% of the data will be used for training, and the rest will be used for testing; this split helps evaluate the model’s unseen data.

RMSE was used to evaluate the difference between predicted and actual values; RMSE is well-suited for continuous data and helps in the straightforward measure of prediction accuracy. MAE was factored in because, unlike RMSE, it treats all errors equally, making it more robust to outliers providing a balanced view of prediction accuracy.

This model was tested against a new dataset to confirm its compatibility with other datasets and ensure it does not have an issue predicting others. All the data-cleaning processes on the initial dataset were recreated, and the same EDAs were also done after following the same steps (creating lagged values, running training sets and confirming the predictions). Finally, overfitting and percentage accuracy were checked by evaluating the models on both the training and validation datasets.

**Results**

**A graph showing different colored lines

Description automatically generated**

***Training data evaluation and results by model***

CNN: R-squared: 0.765, MAPE:4.017%, Accuracy:95.98%, RMSE: 5.639, MAE:4.017

XGBoost: R-squared: 0.849, MAPE: 3.429%, Accuracy: 96.57%, RMSE: 5.649, MAE:3.429

***Validation Data evaluation and results by model***

CNN: R-squared: 0.641, MAPE:5.639%, Accuracy:94.36%, RMSE: 6.327, MAE: 4.762

XGBoost: R-squared: 0.742, MAPE: 4.688%, Accuracy: 95.31%, RMSE: 5.649, MAE:3.429

The results from the CNN and XGBoost models offer valuable insights into their respective strengths and weaknesses in predicting stock prices.

***CNN Model Performance***

The model demonstrated strong performance on the training data with high accuracy and a low error rate. This indicates that the model effectively learns patterns from the historical data. The performance drop in the validation data suggests that the model may be overfitting. This is evident from the significant decrease in R-squared and the increase in MAPE and RMSE on the validation set.

***XGBoost Model Performance***

This model performed well on both the training and validation data, therefore showing its ability to handle complex, non-linear relationships and its robust performance metrics, which indicate that it generalizes better than the CNN model. Also, the smaller drop in r-squared and accuracy and relatively stable error rates showcase XGBoost’s strength in avoiding overfitting. Therefore, these results are instrumental in guiding our decision-making process; the XGBoost model seems more suitable for our stock price prediction due to its ability to generalize well with new data, which is essential for making informed financial decisions.

**Discussions**

**Limitations**

The CNN model demonstrated a strong performance on the initial training data but needed help with overfitting; this limits its reliability in real-world stock predictions. Another limitation of CNNS is that they require a lot of computational power and time to train, which can be a constraint if there are limited resources. The XGBoost model requires careful tuning of hyperparameters to achieve the best performance, which can be time-consuming. While it is good at handling complex data, it might not capture intricate patterns in sequential data or CNN. The year constraint of the past five years may have allowed the model to miss out on long-term trends and cycles that may be essential for making more precise predictions. There may also be significant events or anomalies that could affect the market. The model may, therefore, focus on these events rather than delivering a more stable, long-term behaviour. The models may also be hard to interpret, and using the models for large datasets may prove challenging. However, poor feature selection can lead to less effective models.

# Conclusion

The XGBoost model demonstrated better generalization capabilities, with higher accuracy and lower error rates on training and validation datasets. The CNN model, while performing well, showed signs of overfitting. Evaluating both models has provided clear evidence of their abilities and limitations; the insights have allowed us to select the most effective model for our project, which would help deliver valuable forecasts. This analysis highlights the importance of using multiple metrics to evaluate and compare models effectively.

**References**

Ali, M. M. (2024). Stock Market Analysis. [Stock Market Analysis (kaggle.com)](https://www.kaggle.com/datasets/thesnak/stock-market-analysis)

Deepika, N., & Nirupamabhat, M. (2020). An Optimized Machine Learning Model for Stock Trend Anticipation [Article]. *Ingénierie des Systèmes d'Information*, *25*(6), 783-792. <https://doi.org/10.18280/isi.250608>

Dezhkam, A., Manzuri, M. T., Aghapour, A., Karimi, A., Rabiee, A., & Shalmani, S. M. (2023). A Bayesian-based classification framework for financial time series trend prediction [Article]. *Journal of Supercomputing*, *79*(4), 4622-4659. <https://doi.org/10.1007/s11227-022-04834-4>

Khan, A. H., Shah, A., Ali, A., Shahid, R., Zahid, Z. U., Sharif, M. U., Jan, T., & Zafar, M. H. (2023). A performance comparison of machine learning models for stock market prediction with novel investment strategy. *PLoS ONE*, *18*(9), 1-19. <https://doi.org/10.1371/journal.pone.0286362>

M, I., Ahmad, S., Jha, S., Alam, A., Yaseen, M., & Abdeljaber, H. A. M. (2022). A Novel AI-Based Stock Market Prediction Using Machine Learning Algorithm. Scientific Programming, 1-11. <https://doi.org/10.1155/2022/4808088>

MIQ. (2024). NVIDIA Corporation (NVDA) Stock | 2020 to 2024. [NVIDIA Corporation (NVDA) Stock | 2020 to 2024 (kaggle.com)](https://www.kaggle.com/datasets/muhammadibrahimqasmi/nvidia-corporation-nvda-stock-2020-to-2024)

Prasad, A., & Seetharaman, A. (2021). Importance of Machine Learning in Making Investment Decision in Stock Market [Article]. *Vikalpa: The Journal for Decision Makers*, *46*(4), 209-222. <https://doi.org/10.1177/02560909211059992>

Raubitzek, S., & Neubauer, T. (2022). An Exploratory Study on the Complexity and Machine Learning Predictability of Stock Market Data. *Entropy*, *24*(3), 332-332. <https://doi.org/10.3390/e24030332>

Xiao, D., & Su, J. (2022). Research on Stock Price Time Series Prediction Based on Deep Learning and Autoregressive Integrated Moving Average. *Scientific Programming*, 1-12. <https://doi.org/10.1155/2022/4758698>