City University of Hong Kong

SDSC3002

Data Mining Project Report

Semester B (2023-2024)

Stock Price Prediction

Table of contents

Part 1. Introduction

1.1 Project Description

Predicting stock prices is a complex task that necessitates a thorough comprehension of the variables affecting the stock market. To generate accurate predictions, a wide range of factors such as past data and external occurrences, are taken into account.

Large volumes of historical price data are processed efficiently by data mining algorithms, which allow analysts to spot trends and forecast future prices. Past pricing information serves as the foundation for the prediction task. Data mining algorithms can analyze patterns, trends, and cycles in the data, which can help analysts uncover valuable insights by examining price fluctuations and other technical indicators. This may help in identifying recurring patterns that might indicate potential future price movements.

The prediction of stock prices is heavily influenced by external influences, and sentiment analysis, which involves extracting sentiment from textual sources like social media and news articles to help analysts assess market psychology, becomes an essential tool in this context. A more comprehensive comprehension of the prevalent market attitude towards individual stocks or the market as a whole can be achieved by analyzing sentiments.

1.2 Description of Datasets and Data Mining Tasks

Some common approaches involved in a stock prediction task include:

1. **Machine Learning Algorithms**: Regression models can help model the relationship between past and future pricing data.
2. **Long Short-Term Memory (LSTM) Networks:** LSTM Networks are effective to use when the data is sequential, which is the case in stock price data, as they can help to capture long term dependencies in the data.
3. **Sentiment Analysis**: Sentiment from news sources or social media can help capture public sentiment on specific stocks, which might be useful in making future predictions.

The methods discussed will be implemented and a comparative analysis will be made to identify the strengths, limitations, and suitability of each technique for various scenarios.

The dataset chosen for implementing the afore-mentioned prediction tasks is the Apple Inc. (AAPL) stock data from Yahoo finance. The library called “yfinance” allows up-to-date market data to be downloaded from the Yahoo Finance website.

Additionally, News Headline Data scraped from the stock screener website FINVIZ and Twitter sentiment data for Apple stocks from Kaggle were used for the sentiment analysis task. The Twitter sentiment data includes the mean sentiment polarity scores and volume of the everyday-tweets related to stock Apple stocks.

The timeline chosen (start date = "2016-01-04 and end date = "2019-08-30") for the stock data coincides with the twitter sentiment data to ensure consistency across all methods.

Figure 1.2.1 Visualization of AAPL stock

Date	Open	High	Low	Close	Adj Close	Volume
2016-01-04	25.652500	26.342501	25.500000	26.337500	23.946920	270597600
2016-01-05	26.437500	26.462500	25.602501	25.677500	23.346825	223164000
2016-01-06	25.139999	25.592501	24.967501	25.174999	22.889936	273829600
2016-01-07	24.670000	25.032499	24.107500	24.112499	21.923878	324377600
2016-01-08	24.637501	24.777500	24.190001	24.240000	22.039804	283192000

Figure 1.2.2 AAPL stock

date	ts_polarity	twitter_volume
2016-01-01	0.119693	417
2016-01-02	0.140774	495
2016-01-03	0.181132	518
2016-01-04	0.070389	1133
2016-01-05	0.133635	1430

Figure 1.2.3 Twitter Sentiments

	ticker	date	time	headline	neg	neu	pos	compound
0	AAPL	2024-04-17	05:58PM	Tim Cook says Apple wants to invest more in Vi...	0.000	1.000	0.000	0.0000
1	AAPL	2024-04-17	05:45PM	Apple (AAPL) Stock Sinks As Market Gains: Here...	0.000	0.769	0.231	0.3400
2	AAPL	2024-04-17	04:51PM	iPhone Sales Angst Continues, Sales on Apple P...	0.000	1.000	0.000	0.0000
3	AAPL	2024-04-17	04:25PM	Apple Stock Drops As Growth Outlook Called Ane...	0.259	0.559	0.182	-0.2732
4	AAPL	2024-04-17	04:07PM	Apple stock down as Needham cuts estimates on ...	0.328	0.672	0.000	-0.5267

Figure 1.2.4 Finviz News Headline

1.3 Evaluation Metrics of the Data Mining Task

The following metrics were used to quantitatively evaluate each of the methods:

1. **Mean Absolute Error (MAE):** In this, the sum of the absolute difference between each predicted value and its corresponding actual value is calculated and then divided by the total number of observations. It represents the average of all absolute errors in a set of measurements.

$$\mathrm{MAE} = \frac{\sum_{i=1}^{n} |y_i - x_i|}{n}$$

MAE = mean absolute error
y_i = prediction
x_i = true value
n = total number of data points

Figure 1.3.1 Formula for MAE

2. **Mean Squared Error (MSE):** Here, the sum of the squared difference between each predicted value and its corresponding actual value is divided by the total number of observations.

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$

MSE = mean squared error
n = number of data points
Y_i = observed values
$\hat{Y}_i$ = predicted values

Figure 1.3.2 Formula for MSE

3. **Root Mean Squared Error (RMSE):** It is just the square root of MSE.

4. **R-squared (R^2):** It is also called the coefficient of determination and is used to assess the goodness of fit of a regression model. It can tell what proportion of the variance in the output variable can be explained by the input variables. Values close to 1 indicate a good fit, whereas a value near 0 suggests that the model explains none of the variance in the output variable.

$$R^2 = 1 - \frac{RSS}{TSS}$$

R^2 = coefficient of determination
RSS = sum of squares of residuals
TSS = total sum of squares

Figure 1.3.3 Formula for R^2

Part 2. Data Mining Techniques

2.1 Machine Learning Algorithms

Machine learning algorithms are a set of rules or processes used to discover new data insights and patterns, or to predict output values from a given set of input variables. In this project, multiple models are selected to test to choose the best model with the optimal performance. Among all the models, linear regression and MLP Regressor Model are selected to predict the future stock price. It provides a high accuracy and reliable result.

2.2 Long Short-Term Memory (LSTM) Networks

The project adopts LSTM, a type of Recurrent Neural Network (RNN) known for its ability to remember information for long periods, which is crucial for time series forecasting tasks like stock price prediction. LSTM networks are capable of learning order dependence in sequence prediction problems. This characteristic will benefit the project by effectively capturing the temporal dependencies in stock price movements, leading to more accurate predictions.

2.3 Sentiment Analysis

Incorporating Twitter sentiment analysis into stock prediction involves analyzing tweets related to Apple stocks. With its vast user base and active nature, Twitter captures a wide range of opinions and perspectives, making it a valuable tool for monitoring sentiment. By analyzing sentiment on Twitter, one can potentially identify sentiment-driven price movements and make timely decisions based on the prevailing sentiment. The immediacy and broad coverage of Twitter can offer valuable insights into public sentiment, allowing for quick reactions and adjustments to market dynamics.

Part 3. Experiments and Results

Each method was explored independently with the same basic criteria:

- Training and evaluation should be done using a 70:30 ratio for train:test split
- Stock data used should have the same timeline to allow fair comparisons
- The evaluation metrics should be the same and calculated with untransformed data

3.1 Machine Learning Algorithms

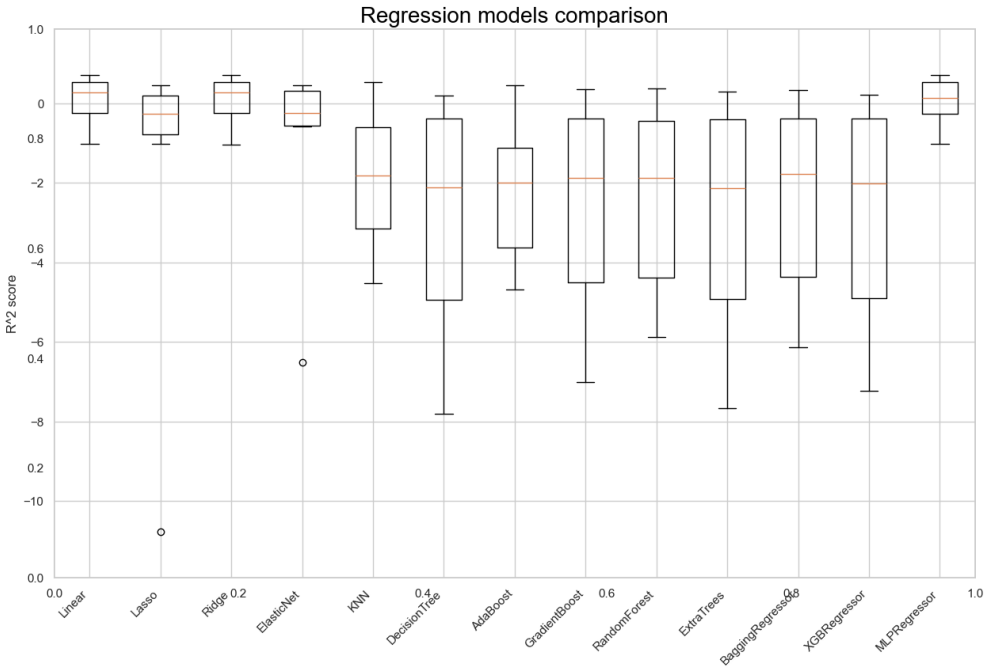

Figure 3.2.1 Regression Models Box Plot for R-squared values

To get the optimal result among different regression models, a time-series cross-validation with 7 splits was performed with the training data. The R-squared (R^2) scores were calculated and the results were visualized using a boxplot in Figure 3.2.1. Based on the cross-validation performance, the Linear Regression and Multilayer Perceptron Regressor Model were selected for forecasting the price of stock in the next 7 days.

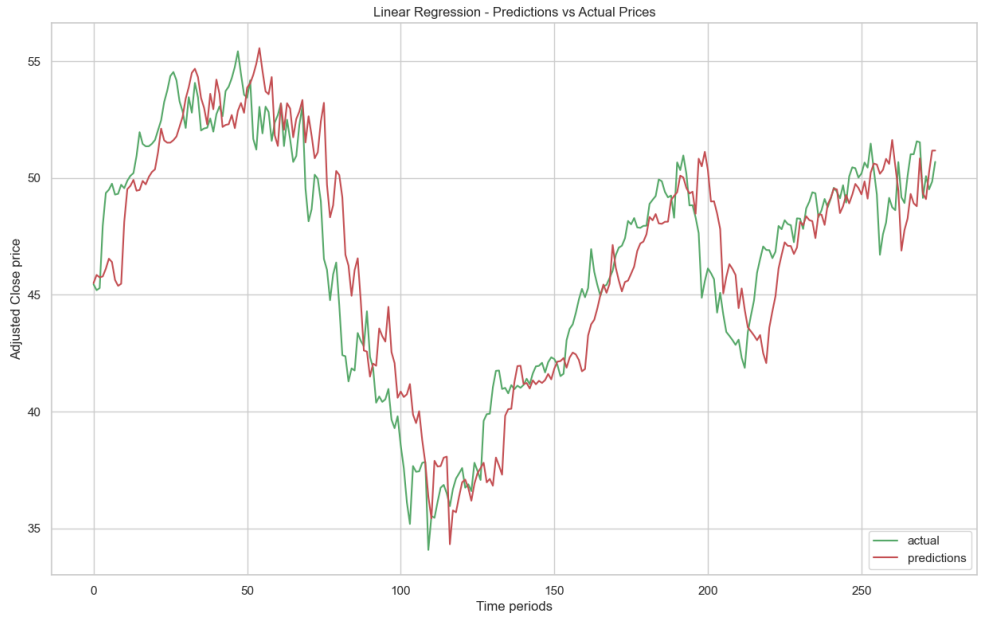

Figure 3.2.2 Result of Linear Regression Model without Hyperparameter Tuning

```
Mean Absolute Error: 1.6626
Mean Squared Error: 4.5873
Root Mean Squared Error: 2.1418
R-squared: 0.8192
```

Figure 3.2.3 The Metrics with Model Performance of No-Tuning Linear Regression Model

Using the initial parameters, the result of the linear regression model is shown in Figure 3.2.2. However, as shown in Figure 3.2.3, its performance was not good enough for predicting the actual stock price. Therefore, a grid search cross-validation approach was employed to select the best parameters of it.

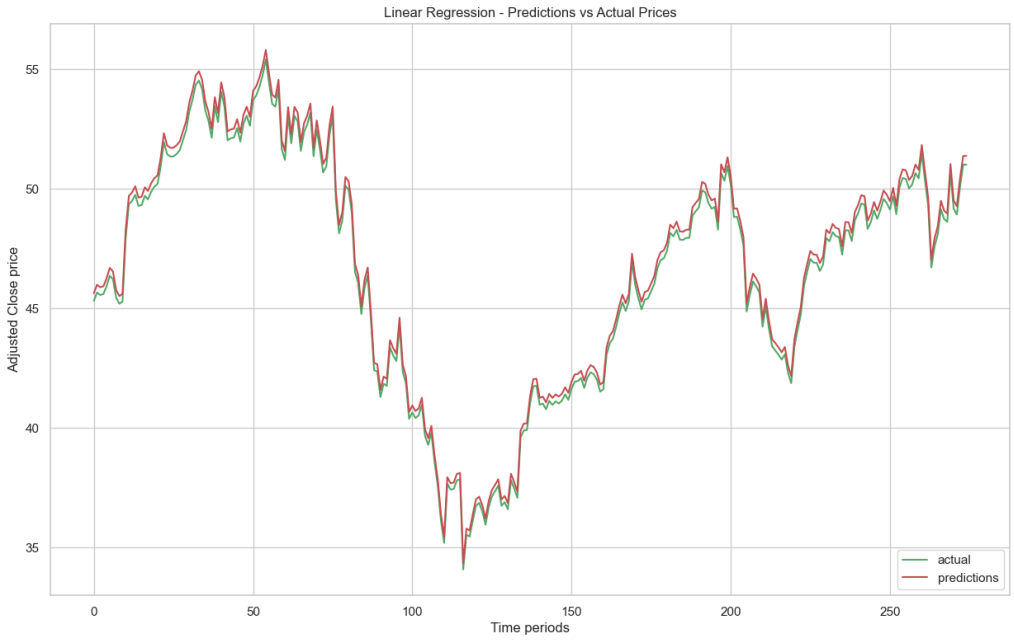

Figure 3.2.4 Result of Linear Regression Model with Best Hyperparameters

```
Mean Absolute Error: 0.3311
Mean Squared Error: 0.1109
Root Mean Squared Error: 0.333
R-squared: 0.9956
```

Figure 3.2.5 The Metrics with Model Performance of Linear Regression Model with Best Parameters

With the best parameters found by grid searching, the result of the linear regression model is magnificent. As shown in Figure 3.2.4, the prediction prices were very close to the actual stock prices. The low values of MAE, MSE, and RMSE in Figure 3.2.5 represented the prediction error was low. The 0.9956 R-squared score indicated a nearly perfect prediction accuracy.

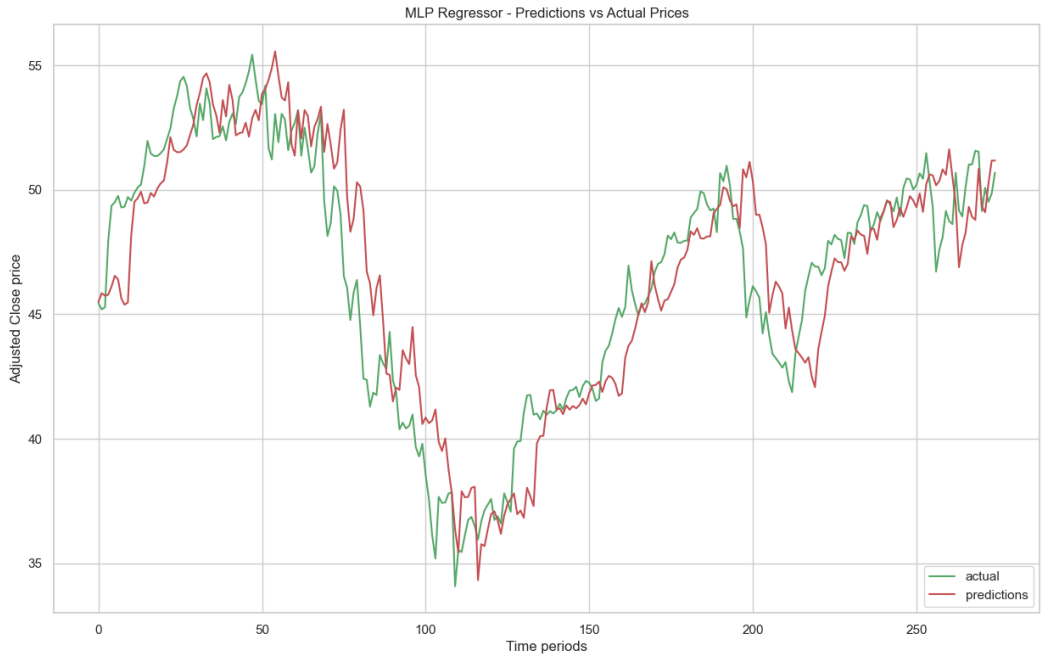

Figure 3.2.6 Result of MLP Regressor Model without Hyperparameter Tuning

```
Mean Absolute Error: 1.6626
Mean Squared Error: 4.5873
Root Mean Squared Error: 2.1418
R-squared: 0.8192
```

Figure 3.2.7 The Metrics with Model Performance of No-Tuning MLP Regressor Model

Using the initial parameters, the result of the MLP regressor model is shown in Figure 3.2.6. However, as shown in Figure 3.2.7, its performance was not good enough for predicting the actual stock price. Therefore, a grid search cross-validation approach was employed to select the best parameters of it.

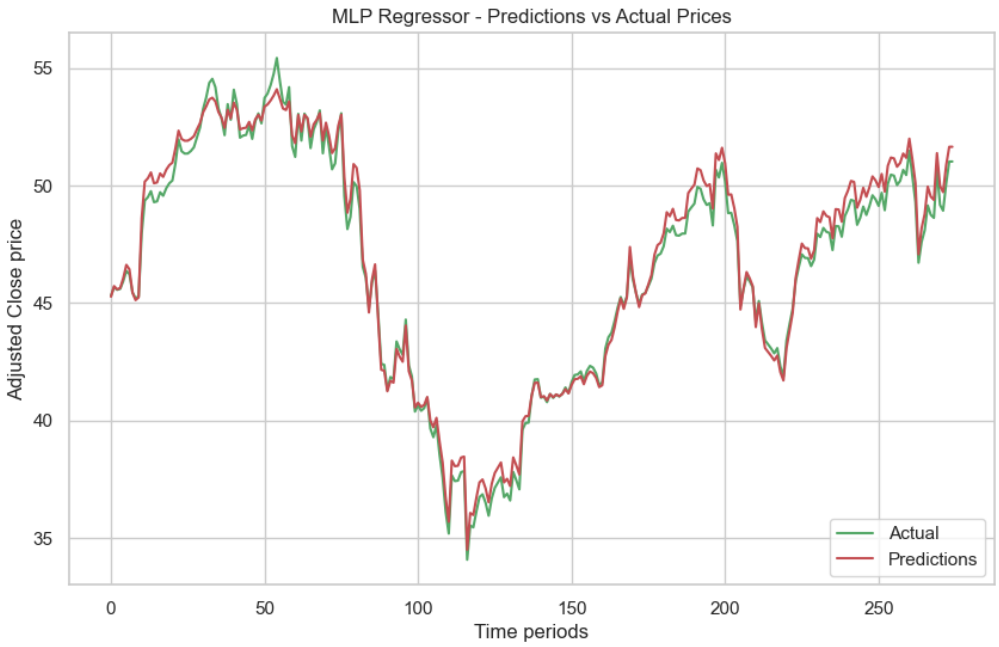

Figure 3.2.8 Result of MLP Regressor Model with Best Parameters

```
Mean Absolute Error: 0.4403
Mean Squared Error: 0.2705
Root Mean Squared Error: 0.5201
R-squared: 0.9892
```

Figure 3.2.9 The Metrics with Model Performance of MLP Regressor Model with Best Parameters

With the best parameters found by grid searching, the result of the MLP regressor model is great. As shown in Figure 3.2.8, the prediction prices were very close to the actual stock prices. The low values of MAE, MSE, and RMSE in Figure 3.2.9 represented the prediction error was low. The 0.9893 R-squared score indicated a high accuracy of stock price prediction.

Overall, both Linear Regression and MLP Regressor Model were performing well on a 7-day forecast of stock price prediction. Those results, 0.9956 and 0.9892 R-squared scores indicated that machine learning algorithms were a good choice for short-term stock price predictions.

Figure 3.2.10 Result of Linear Regression Model with Best Parameters when forecast_out = 30

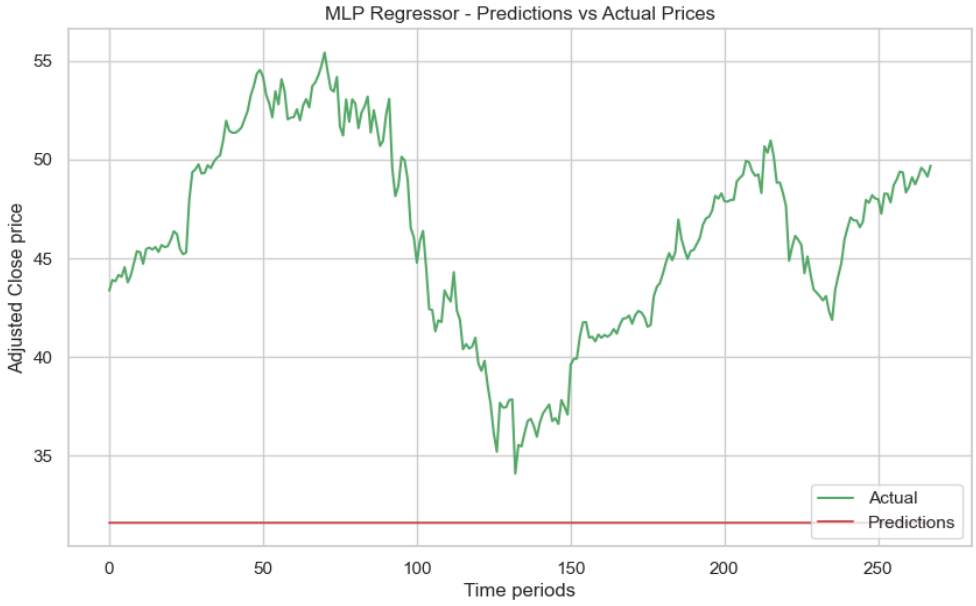

Figure 3.2.11 Result of MLP Regressor Model with Best Parameters when forecast_out = 30

To test how the models would perform when the forecast horizon is increased, forecast_out was set to 30 and it was noticed that the performance of the tuned Linear and MLP Regressor Models both reduced. The Linear Regression Model was still able to fit the data to some extent. The tuned MLP Regressor model, however, failed to fit the data entirely. Thus, there is a trade-off between model performance and prediction horizon. The farther you want to predict, the less accurate the results will be.

3.2 Long Short-Term Memory (LSTM) Networks

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 lstm (LSTM)                 (None, 90, 128)           66560

 dropout (Dropout)           (None, 90, 128)           0

 lstm_1 (LSTM)               (None, 90, 128)           131584

 dropout_1 (Dropout)         (None, 90, 128)           0

 lstm_2 (LSTM)               (None, 128)               131584

 dropout_2 (Dropout)         (None, 128)               0

 dense (Dense)               (None, 1)                 129

=================================================================
Total params: 329857 (1.26 MB)
Trainable params: 329857 (1.26 MB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

Figure 3.3.1 Model Summary

A model comprising three recurrent layers, each equipped with 128 neurons, was developed. The model's input is structured to accommodate data with 90-time steps and a single feature, while its output is designed with a time step of 1, utilizing the linear activation function. Despite the presence of dropout in the code, it will not be considered in this context, as the probability was set to 0. The model underwent compilation employing the Mean Squared Error (MSE) as the loss function and was evaluated using metrics including mean absolute error and mean squared error, with the Adam optimizer facilitating the optimization process. The training phase involved fitting the model to the training dataset over 50 epochs, with a batch size 64.

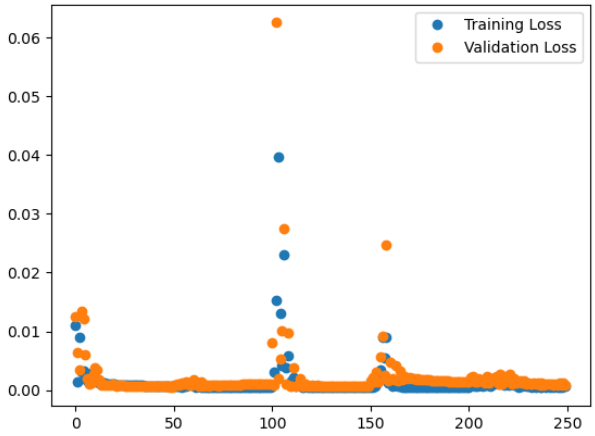

Figure 3.3.2 The losses over each epoch

A 5-fold cross-validation was also implemented through a time series split to fine-tune the LSTM network. The total time of the training model is 7 minutes and 59 seconds.

Figure 3.3.2 visualizes a machine learning model's training and validation loss over multiple epochs, which are iterations of the training process. The x-axis represents the number of epochs, and the y-axis shows the loss values. The results show fine-tuning after the 200th epochs, as the training loss and validation loss values are close. Overall, the model is learning effectively across epochs, with both training and validation losses decreasing, a positive sign of model convergence.

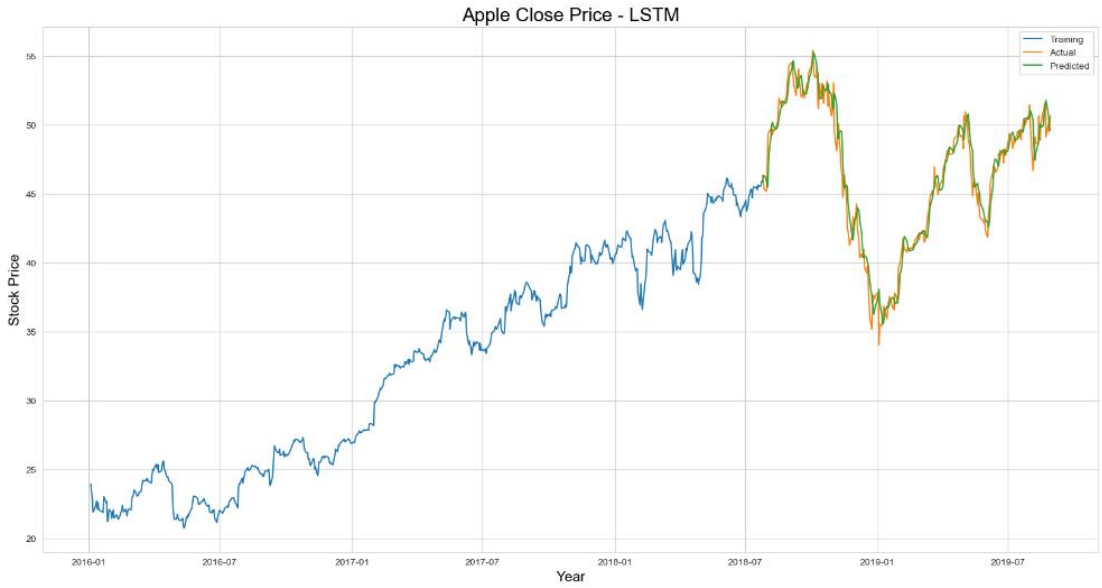

Figure 3.3.3 The testing performance on stock price

Figure 3.3.3 the line chart visualizes the model's performance by comparing the actual and predicted stock prices over time. The blue line represents the actual stock prices, while the orange line represents the predicted prices from the testing data. The green line represents the predicted prices from the training data.

```
Mean Absolute Error: 0.8267
Mean Squared Error: 1.1932
Root Mean Squared Error: 1.0923
R-squared: 0.9526
```

Figure 3.3.4 The metrics with model performance

Mean Absolute Error (MAE): The MAE is 0.8267, representing the average magnitude of the errors between the predicted and actual values. This metric tells us that, on average, the model's predictions are approximately 1.03 units away from the actual stock prices.

Mean Squared Error (MSE): The MSE is 1.1932, the average of the squares of the errors. This metric is sensitive to larger errors because it squares the differences; it shows a low MSE, suggesting the presence of low errors in the predictions.

Root Mean Squared Error (RMSE): The RMSE is 1.0923, which is the square root of the MSE. It can be interpreted as the standard deviation of the residuals and provides a measure of the magnitude of the errors. It indicates that the typical deviation from the actual to the predicted values is about 1.25 units.

R-squared (R^2): The R^2 value is 0.9526, a statistical measure of how close the data are to the fitted regression line. In this case, the model explains approximately 95.26% of the variance in the stock prices.

Overall, The low MAE, MSE and RMSE, along with a high R-squared value, suggest that the LSTM model is performing well in terms of accurately predicting or forecasting the financial data it was trained on.

3.3 Sentiment Analysis

To analyze whether sentiment from textual sources have any relationship on stock prices, a small experiment was conducted. 5-day new headlines related to AAPL stocks were scraped from the stock screener website, Finviz. Then, VADER (Valence Aware Dictionary and sEntiment Reasoner), an NLTK module that measures sentiment polarity (positive, negative, or neutral) in text data, was used to perform sentiment analysis on the headlines and calculate their compound polarity scores. Each score represented the overall sentiment expressed in the headline. Since there were many news headlines in one day, the sentiment scores were averaged by date. Finally, the sentiment trend was visualized alongside the stock price data for the coinciding dates.

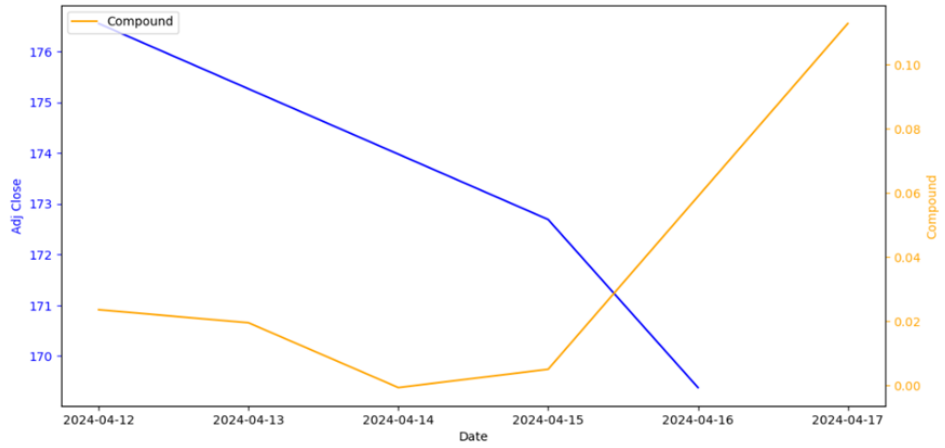

Figure 3.4.1 Average compound sentiment scores vs stock price

Figure 3.4.1 suggests signs of a discernible inverse relationship. This gave us motive to further study the method using a larger and better dataset.

The Twitter Sentiments AAPL stock data, containing Twitter sentiment scores for Apple stocks from 2016 to 2019, was then used to model the relationship.

Prior to modeling, a rolling window function was created for the input features. This function generates the input values for the model using the previous n days of input data. It was done so that the machine

learning model could capture temporal trends and patterns and dependencies in the sequential data. The data, including the input and output variables, were scaled between 0 and 1 using MinMaxScaler from the Scikit-learn library to standardize it. This was done separately on the train and test data to avoid any accidental data leaks.

The model used was XGBRegressor, which is the regression-specific implementation of XGBoost. The parameter n_estimator was set at 1000, which indicates the number of boosting rounds (trees). In short, the model was trained with 1000 trees.

Different values of n (rolling window size) were experimented with and evaluated:

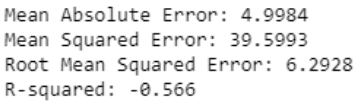

Figure 3.4.2 n = 1

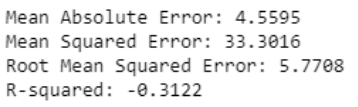

Figure 3.4.3 n = 5

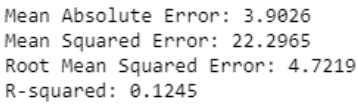

Figure 3.4.4 n = 10

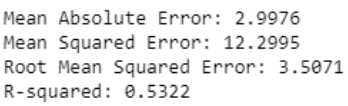

Figure 3.4.5 n = 50

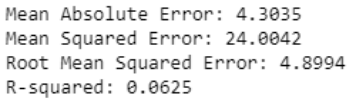

Figure 3.4.6 n = 100

The metrics were calculated after performing inverse scalar transformation on the predicted and actual values.

Generally, the errors tend to decrease as the size of the window n increases and the R-squared value increases with increasing n. For smaller values of n, like 1 and 5, the R-squared values are negative, which indicates that the model fits worse than a horizontal line (with the mean of the input variables). As the value of n increases, particularly around n = 50, the errors are at the lowest and the R-squared value is the highest. However, the model starts performing worse near the 50 mark.

The highest R-squared value is around 53% , which means the model is able to account for 53% of the variation in the stock prices based on the twitter sentiment variables included in the analysis. The remaining 47% of the variation is attributed to other factors of variability.

Figure 3.4.7 Predicted vs actual prices for n = 50

The prediction graph with a window size of 50 indicates that the model captures general trends fairly well, but is not very precise. It is still impressive considering the model was trained only using the twitter sentiment data.

Part 4. Comparison

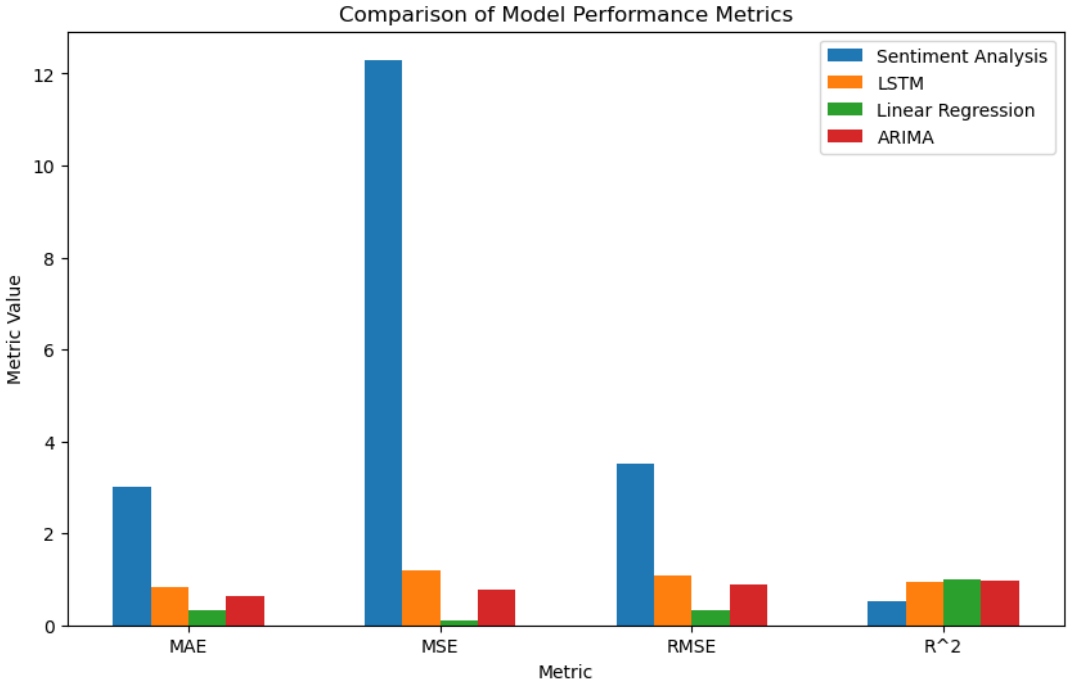

Figure 4.1 Comparison of Model Performance Metrics

The metrics of the best-performing models from each technique were plotted and compared.

Visually, the sentiment analysis model looks like the worst-performing model, which is not surprising as the model in that task was trained only using twitter sentiments. The Sentiment model has an MAE of 2.9976, which is higher than the MAE of Linear Regression (0.3311), and LSTM (0.8267). A lower MAE indicates better accuracy, so in this case, the Linear Regression and ARIMA models perform better than the Sentiment and LSTM models. The MSEs of Linear Regression, LSTM, and Sentiment Models are 0.1109, 1.1932, and 12.2995, respectively. A lower MSE also indicates better accuracy, so the Linear Regression outperform the Sentiment and LSTM models in this metric as well. The Sentiment model has an RMSE of 3.5071, which is higher than the RMSE of Linear Regression (0.3330), and LSTM (1.0923). Again, a lower RMSE indicates better accuracy, so the Linear Regression and ARIMA models perform better than the Sentiment and LSTM models. The Sentiment model has an R-squared value of 0.5322, which is lower than the R^2 values of Linear Regression (0.9956) and LSTM (0.9526). A higher R-squared value indicates a better fit of the model to the data, so the Linear Regression have a better fit compared to the Sentiment and LSTM models.

Based on these metrics, the Sentiment model generally performs worse than the other models (Linear Regression, LSTM) in terms of accuracy and fit. The Linear Regression model is the best-performing model across all metrics.

Part 5. Discussion

The Sentiment Analysis model is characterized by low accuracy and reliability. It can be useful when there is no prior stock information available, as it provides a way to analyze sentiment. However, its performance may be limited in terms of accuracy and reliability. The LSTM model, on the other hand, demonstrates fair accuracy and reliability. While it may have higher computing time compared to other models, it has the potential for further improvement. It is currently only capable of providing 1-day-ahead forecasts, which means it predicts stock prices for the next day. The Linear Regression model exhibits high accuracy and reliability. The current model allows for 7-day-ahead forecasts with high accuracy, enabling predictions of stock prices up to a week in advance. However, it's worth noting that the accuracy of the model tends to decrease as the forecast horizon increases.

Model selection depends on one’s specific requirements and constraints, considering factors such as forecast horizon, computing time, availability of prior information, and desired level of accuracy and reliability.

Models based on sentiment scores can be improved by refining methodologies for generating scores and adjusting weightings of contributing factors. It was not a possibility in this study as the sentiment data already contained the calculated sentiment scores. Due to constraints in scraping large volumes of social media or news data, existing data had to be used. Further study should focus on analyzing the correlation between sentiment scores and stock market performance, identifying patterns and trends in sentiment fluctuations, and investigating the integration of sentiment scores with other models or data sources. By fine-tuning these models and conducting comprehensive analyses, researchers can enhance the accuracy

and reliability of sentiment-based predictions and gain deeper insights into the relationship between sentiment and stock market dynamics.

To improve LSTM models, various factors can be considered. This includes optimizing the architecture and hyperparameters, experimenting with different network configurations, adjusting the sequence length or window size, and incorporating additional features or technical indicators that have shown relevance in stock price prediction. Further study could focus on investigating alternative deep learning models, such as Gated Recurrent Units (GRUs), Transformers, or hybrid models that combine LSTM with attention mechanisms. Additionally, exploring ensemble techniques or combining LSTM with other traditional time series forecasting methods could be beneficial in improving accuracy and reliability.

For Linear Regression models, feature engineering and selection techniques can be applied to identify the most relevant predictors, which can improve accuracy. Considering non-linear relationships through polynomial features or interaction terms might also enhance the model's performance. Further study could explore the application of advanced regression techniques, such as Elastic Net Regression, to handle multicollinearity and improve regularization. Additionally, investigating time-varying coefficients or incorporating external factors, such as macroeconomic indicators, news sentiment, or market data, could be valuable for enhancing forecasting accuracy.

By following these suggestions for improvement and exploring further study areas, researchers and practitioners can enhance the performance and discover new avenues in each model for accurate and reliable stock price prediction. It's crucial to tailor these approaches to the specific characteristics and requirements of the task at hand, iterating and experimenting to refine the models' capabilities.

Part 6. Source

https://drive.google.com/drive/folders/11ky8RObcOY5fKaq8qmrhNL0AMbs9nmKW?usp=sharing

Part 7. References

Brownlee, J. (2021) *Regression metrics for machine learning*, *MachineLearningMastery.com*. Available at: https://machinelearningmastery.com/regression-metrics-for-machine-learning/ (Accessed: 09 April 2024).

Community / Developer Dec 29, 2023 (2023) *Time series differencing: A complete guide*, *InfluxData*. Available at: https://www.influxdata.com/blog/time-series-differencing-complete-guide-influxdb/ (Accessed: 11 April 2024).

GeeksforGeeks (2022) *Implementing web scraping in python with beautifulsoup*, *GeeksforGeeks*. Available at: https://www.geeksforgeeks.org/implementing-web-scraping-python-beautiful-soup/ (Accessed: 11 April 2024).

Hasselgren, B. *et al.* (2022) 'Using social media & sentiment analysis to make investment decisions', *Future Internet*, 15(1), p. 5. doi:10.3390/fi15010005.

Iyinoluwa, O. (2019) 'Stock market trend prediction model using data mining techniques', *Current Trends in Computer Sciences & Applications*, 1(5). doi:10.32474/ctcsa.2019.01.000122.

Kumar, A. and Chaudhry, M. (2021) 'Review and analysis of stock market data prediction using data mining techniques', *2021 5th International Conference on Information Systems and Computer Networks (ISCON)* [Preprint]. doi:10.1109/iscon52037.2021.9702498.

Serafeim Loukas, P. (2024) *LSTM time-series forecasting: Predicting stock prices using an LSTM model*, *Medium*. Available at: https://towardsdatascience.com/lstm-time-series-forecasting-predicting-stock-prices-using-an-lstm-model-6223e9644a2f (Accessed: 05 April 2024).

Sharma, P. (2023) *Different types of regression models*, *Analytics Vidhya*. Available at: https://www.analyticsvidhya.com/blog/2022/01/different-types-of-regression-models/ (Accessed: 02 April 2024).

Xiao, Q. and Ihnaini, B. (2023) 'Stock trend prediction using sentiment analysis', *PeerJ Computer Science*, 9. doi:10.7717/peerj-cs.1293.