Applications of NL(X) and LLM

Individual Assignment - Stock Price Prediction

Name: Chitra Raghavendraro Krishnarao

Andrew ID: crkrishn

1. Introduction

The primary objective of this assignment is to develop models for predicting stock prices, providing valuable insights for individuals and entities engaged in buying, selling, investing, or trading stocks. "The accurate prediction of stock trends is interesting and a complex task in the changing industrial world. Several aspects, which affect the behavior of stock trends, are non-economic and economic factors and which are taken into consideration. Thus, predicting the stock market is considered as a major challenge for increasing production" [1].

Traditionally, stock market prediction has relied on leveraging historic stock data. However, recent advancements in machine learning, particularly in natural language processing (NLP), have enabled the consideration of additional factors such as public opinion on social media platforms. In light of this, our assignment focuses on developing two distinct models: one utilizing only historic stock data and another incorporating both historic stock data and sentiment data extracted from social media.

Our analysis will center on examining the stock trends of GameStop (GME) from January 2021 to August 2021. GameStop, a prominent video game retailer, gained significant attention during this period due to its involvement in a phenomenon known as a 'short squeeze'. "A short squeeze is an unusual condition that triggers rapidly rising prices in a stock or other tradable security.[2] GameStop, once a household name in the video game industry, faced declining business as customers increasingly turned to online shopping. However, a collective effort by retail investors to exploit the company's low share price and high short interest resulted in a dramatic surge in its stock price, reaching nearly $500 in January 2021. In this assignment, we aim to use the stock trends (and sentiment data for the second model) from January 2021 to May 2021 to try and predict the stock prices for the period from June 2021 to August 2021.

2. Data Collection and Preprocessing

2.1 Data Collection

The historic stock market data was sourced using the Yahoo Finance Library, a Python module facilitating the retrieval of market data from Yahoo's financial platform. Leveraging this library enabled the seamless downloading of comprehensive historical stock market datasets.

Simultaneously, the Reddit sentiment analysis data was obtained in the form of a .csv file from Harvard Dataverse. Harvard Dataverse serves as a widely accessible repository for academic research datasets, providing open access to a diverse range of scholarly resources.

2.2 Data Preprocessing

2.2.1 Historic Stock Data

1. Sequence Formation: The data frame was transformed to include sequences from the 8th row onwards, encapsulating stock prices from the current day (t) and the preceding seven days (t-7 to t-1).

2. Normalization: Stock prices were normalized using the min-max scaler technique to standardize values within a range of 0 to 1, enhancing model training stability.

3. Feature-Label Segregation: Features, consisting of date and stock prices of the preceding seven days, were separated from labels representing the current day's stock price. These were organized into separate numpy arrays for streamlined data processing.

4. Data Splitting: The dataset was partitioned into train and test sets. Data from January to May 2021 formed the train set, while data from June to August 2021 constituted the test set. Following this, we also included an additional dimension towards the end, which is a requirement for PyTorch tensors.

2.2.2 Reddit Sentiment Analysis Data

1. Data Selection: Only the date and sentiment score columns were retained from the dataset.

2. Aggregation: Multiple records for the same date were aggregated to calculate the average sentiment scores.

3. Timeframe Filtering: The dataset was filtered to include data from January 2021 to August 2021.

4. Date Standardization: Minor adjustments were made to capitalize the date field and remove time zones.

5. Data Integration: The sentiment data was merged with the stock data based on date using a left outer join operation.

6. Missing Values Handling: 10 NaN values in sentiment scores were identified and imputed using the mean of the respective columns.

7. Data Splitting: The preprocessed data was divided into train and test sets, maintaining consistency with the time frames defined in section 2.2.1. Following this, we also included an additional dimension towards the end, which is a requirement for PyTorch tensors.

3. Model Architecture and Training

The deep learning models for stock price prediction have been developed using the LSTM (Long Short-Term Memory) architecture. This is a type of recurrent neural network that overcomes the vanishing gradient problem faced in traditional RNNs and is employed for sequence modeling tasks. While the models for both are the same, it's the data, dimensionality, parameters, and hyperparameters that differ.

1. **Input Size:** The input size here is 1 for the model; we're considering everything together as a single feature.

2. Hidden Size: The hidden size that was ideal in terms of model complexity and able to capture the underlying relationship between the features and label for both scenarios was 4.

3. **Stacked Layers:** The number of stacked layers is 1 as the requirement is fairly straightforward and simple, with the batch_first parameter being set to true to ensure compatibility with the input format and also ensure that the batch field is the first dimension.

4. **Device:** The device being used for this is a CPU, considering the volume of data and computational requirements.

Model Training:

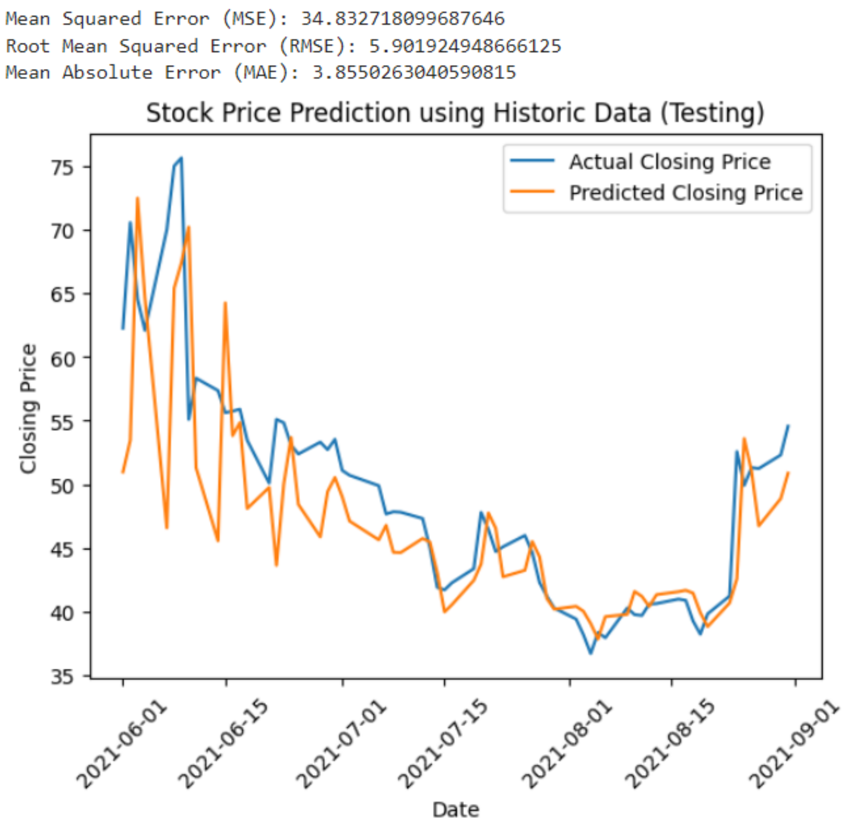
1. **Training Configuration:** The model is trained for 200 epochs with a learning rate of 0.05 and a batch size of 16.

2. **Optimization and Loss:** Adam optimizer is employed for optimization and loss is measured with the mean squared error loss function. Furthermore, backward propagation is incorporated for gradient descent.

3. **Training Process:** Each epoch involves training and testing the model, with performance metrics stored for later analysis.

4. Performance Evaluation

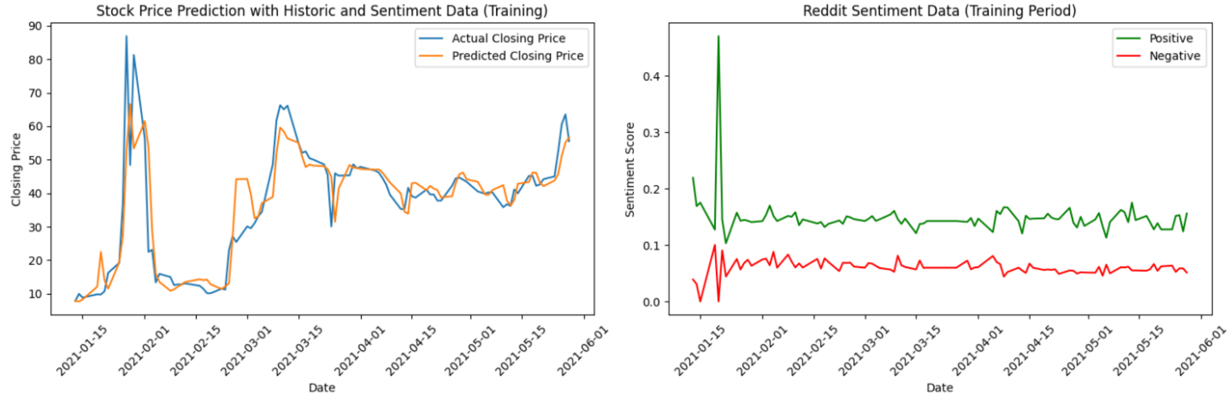
The performance of the models was evaluated using the following metrics: mean squared error, root mean squared error, and mean absolute error. The following metrics were observed for the model that used only historical data:



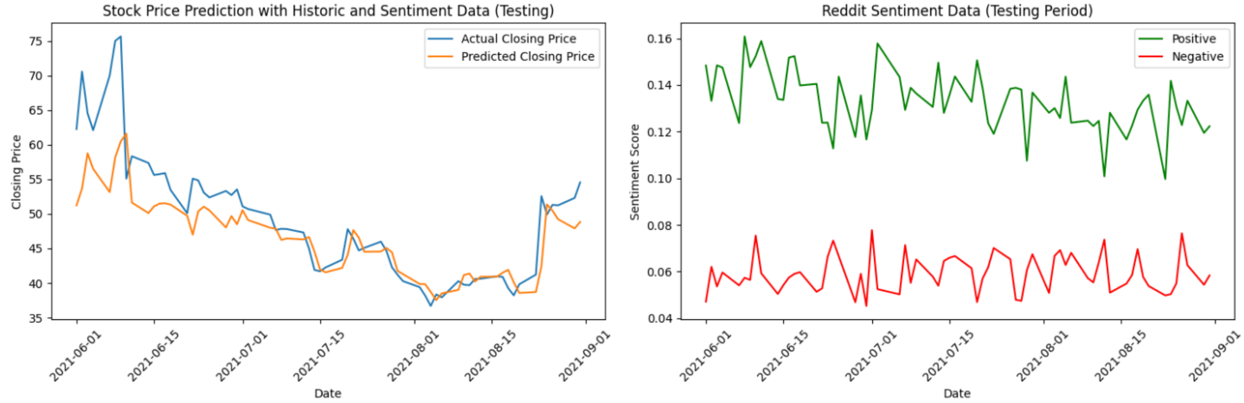
```
Mean Squared Error (MSE): 34.832718099687646
Root Mean Squared Error (RMSE): 5.901924948666125
Mean Absolute Error (MAE): 3.8550263040590815
```

The following were observed for the model that used both historical data as well as sentiment data:

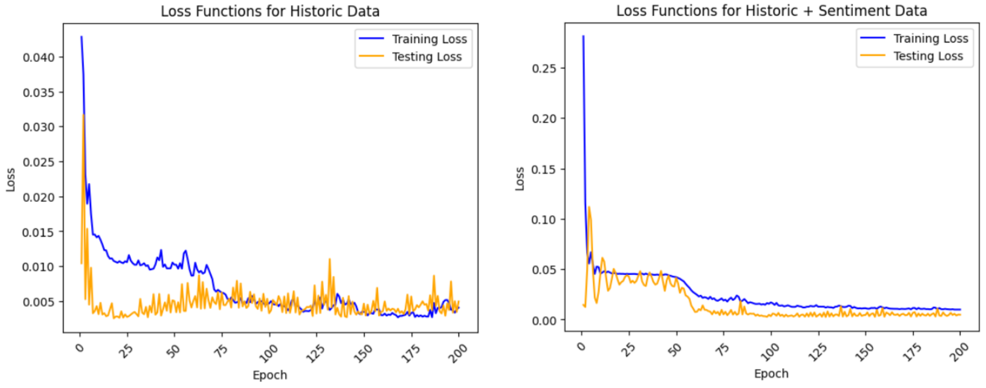
```
Mean Squared Error (MSE): 60.92041072491376
Root Mean Squared Error (RMSE): 7.805152831617954
Mean Absolute Error (MAE): 4.581051165787612
```



```
Mean Squared Error (MSE): 28.80181143838042
Root Mean Squared Error (RMSE): 5.366731914152264
Mean Absolute Error (MAE): 3.471653335264859
```



The losses for both models during training and testing are as follows:



5. Analysis

Based on the evaluation metrics, we can see that both models perform well, with the second one which combines the historical data with Reddit data performing slightly better. We can also observe that for the second model, the positive and negative sentiments have a highly significant difference only towards the beginning of the training period. We can see that there was an increase in positive sentiments around 01/15 and soon after, we can see the stock price rising during the training period. It’s also evident that the model has caught on to this and has predicted a relatively high stock price after the spiked positive sentiment. Following this, it appears that the positive sentiment score is always higher than the negative sentiment score. However, this doesn’t necessarily translate to an increase in stock prices. For instance, the period after 01/15 in the training graph shows sudden spikes and dips, but there isn’t any significant difference in the sentiment scores. The same goes for the testing period, where the sentiment scores don’t appear to influence the prices.

Furthermore, we can notice for both models that the initial errors are significantly large and they reduce gradually; this could be attributed to having more epochs. Since a single record is trained and tested during each epoch, the performance improves over epochs.

6. Conclusion

The accuracy achieved by both models is commendable, considering the size and complexity of the input data. However, it is noteworthy that the social media data had limited impact on stock price prediction; this was correctly gauged by the model and it appears to be able to distinguish between relevant and irrelevant data for analysis.

Moving forward, future research could explore innovative approaches to improve the performance of stock price prediction models that integrate social media sentiment. This may involve leveraging data from a diverse range of online platforms beyond Reddit, such as Twitter, news articles, and financial forums, to capture a broader spectrum of market sentiment. We can also try to give more weightage to opinions posted by celebrities as this could significantly impact the stock prices. For instance, the time when Elon Musk tweeted “Use Signal”, the stocks oIt’s also essential to consider a much bigger dataset, that has many more features, such as perhaps the stock prices of other companies which are on the same level as the one being examined.

7. References:

[1] TicknorJ.L., YehC.Y., BoyaciogluM.A., AraúJoR.D.A., ChenM.Y., KimK., WeiL.Y., HassanM.R., KaraY., GhiassiM., HuangW., CarlssonC., EnkeD., ChakravartyS., ZhongX., HadavandiE., PatelJ., MoghaddamA.H., GuresenE., … NandaS.R. (2019, August 28). *Systematic analysis and review of Stock Market Prediction Techniques*. Computer Science Review. https://www.sciencedirect.com/science/article/pii/S157401371930084X#b2

[2] Mitchell, C. (n.d.). *Short squeeze: Meaning, overview, and faqs*. Investopedia. https://www.investopedia.com/terms/s/shortsqueeze.asp

[3] Model Development - YouTube. (2023, April 8). *Amazon Stock Forecasting in Pytorch with LSTM Neural Network (time series forecasting) | tutorial 3*. YouTube. https://www.youtube.com/watch?v=q_HS4s1L8UI&t=681s