**Muhammad Asghar**

**Applications of NLX and Large Language Models**

**Assignment 1 Report GME Short Squeeze**

GameStop stock experimented a rise in popularity in 2021. This was due to social media activity where users attempted to “short squeeze” the stock. This project looks into the sentiment behind that phenomenon based off reddit comment history as well as GME stock price in 2021. The data comes from Yahoo Finance from 01-04-2021 to 12-31-2021 and from Harvard Dataverse Han, Jing, 2022, "Reddit Dataset on Meme Stock: GameStop".

To begin, I started with basic exploratory data analysis on general stock prices in 2021.

A graph showing a line of a price

Description automatically generated with medium confidence

A graph showing a line graph

Description automatically generated with medium confidence

A graph of blue and orange lines

Description automatically generated

Then I used a Recurrent Neural Network and Long Short-Term Memory approach to predict the stock prices after May 31, 2021. The training was done prior to May 31, and the test was after May 31.

With RNN I have the following architecture. A one-dimensional convolutional layer (Conv1D), which applies filters to the input data to extract relevant features. It uses 5 filters, each with a kernel size of 3 (i.e., a sliding window of 3 time steps). The activation function used is ReLU (Rectified Linear Unit). Then I use a Simple Recurrent Neural Network (SimpleRNN) layer to the model. This layer processes sequential data (such as time series) by maintaining an internal state. It has 32 hidden units (neurons). I then include a dropout layer to prevent overfitting during training, randomly setting 20% of the input units to 0 at each update. Then comes a fully connected (dense) layer with a single output unit. This layer performs a linear transformation on the input data. Lastly, I apply a linear activation function to the output of the previous layer.

In this case, it maintains the output as-is (no activation function).

I use the mean squared error as the loss function and the adam optimizer. Then to fit the model, I use 100 epochs, and a batch size of 18. I use 10% for the validation split.

A graph showing a line

Description automatically generated

A graph with blue and orange lines

Description automatically generated

As for performance, I calculated the Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and Mean Absolute Error (MAE).

For the LSTM model I set a window size of 20 (assuming 5 trading days). I set the number of units (neurons) to 50 and use the ReLU activation function. I then incorporate a dropout layer during training, randomly setting 20% of the input units to 0 at each update. Then I add a dense layer with the linear activation function. For the loss, I compute the mean squared error and have the optimizer set to adam.

A graph with blue lines and orange dots

Description automatically generated

As for the RSME, MAE, and MSE I got

In addition to this, I looked into the sentiment analysis and of note I was able to get the most frequent words in the comments, an overall sentiment score, and was able to extract the authors that commented the most.

A black background with red text

Description automatically generated

A graph of a bar chart

Description automatically generated with medium confidence

A graph of a number of people

Description automatically generated with medium confidence

Lastly, I combined the LSTM model with the sentiment analysis data to identify any trends with the stock prices.

I believe my model did well in terms of rising and falling fairly when there were sudden spikes in sentiments.

As I did not get to do much web scraping, I cannot comment on that process, however it has become harder and harder to scrap larger websites such as Twitter and Reddit as they incorporate anti-scraping methods. However, from an ethical issue, there are many issues of user privacy, bias and discrimination, and legal and regulatory challenges. Companies can only be proactive when it comes to the use cases of this data, but it ultimately the end user who must make wise decisions on what this data should be used for.

In conclusion, I learned a lot from this project and incorporated many of the learnings from the class. As the word embeddings, POS tagging, text stemming, lemmatization and broader cleaning of text was done for us, that made the project quicker to navigate. It was also interesting to see the emojis being converted into a vectorized form that aided in the development of the model.

References: