Isaac Simoneau | Benjamin Gervais | Andrew Bastress
Preliminary Results

Problem Statement

Our goal is to create a machine learning model that can predict stock prices based on past stock price trends. As the stock market can often be deterministic due to periodic trends and other factors, we aim to predict certain market metrics such as trading volume and stock price (daily open, close, high, low).

Data Preprocessing

For this project, we will be working with the Alpha Vantage API to access real-time and historical stock data which provides a wide array of metrics. This data is crucial for our machine learning model, which predicts future stock prices. Our initial testing consists of analyzing historical data for a given company, and so the api was used to pull five daily metrics: the open, high, low, and close prices, as well as the volume traded, for any given company’s stock. The data is formatted in a json file which can be interpreted by our LSTM model.

Machine Learning Model

The code uses the Keras library to implement our LTSM model. Various libraries were used, including Pandas for data manipulation, NumPy for numerical operations, and scikit-learn's MinMaxScaler for data normalization.

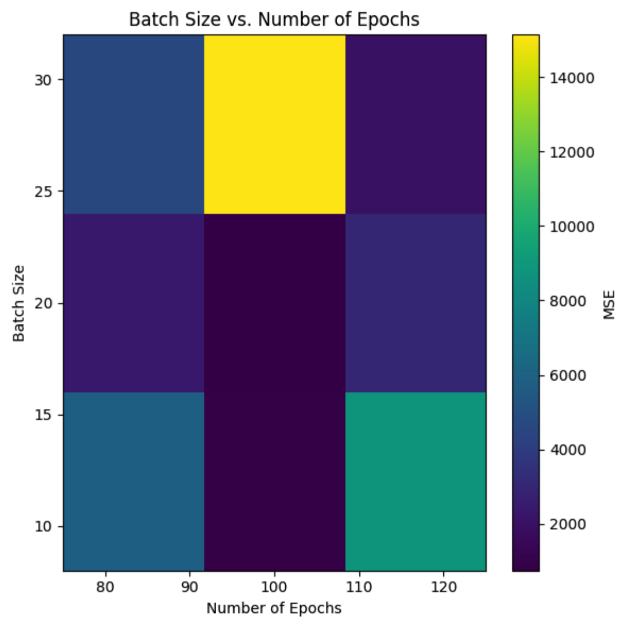
The model is designed for time series prediction, particularly for predicting stock prices. The model consists of three LSTM (Long Short-Term Memory) layers, three dropout layers, and a dense layer.

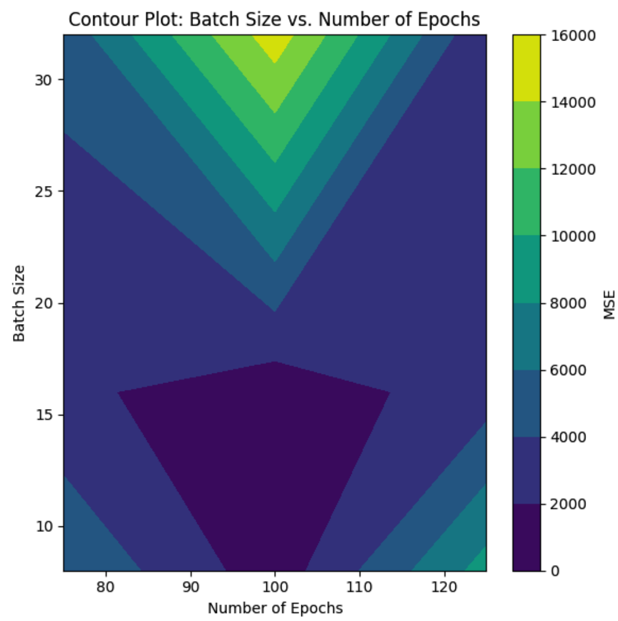
- The three LSTM layers are used for learning patterns and dependencies in the time series data. Each layer consists of 50 units.
- A dropout layer with a dropout rate of 0.2 is added after each LSTM layer to prevent overfitting.
- The dense layer is the final output layer with a single unit used for regression to predict the stock's closing price.

The Adam optimizer using stochastic gradient descent was chosen for training the model. The 50 LTSM units per layer as previously specified is a hyperparameter. Our batch size and epochs are variable parameters that are yet to be fully optimized. Finally, as Adam optimizes for learning rate, we do not include it in this section.

Preliminary results

As discussed in deliverable 1, we used Mean Square Error (MSE) to evaluate the model's accuracy and precision in predicting daily market metrics. Testing with larger batch sizes (64, 128, 256) yielded high MSE values, so we reduced the number to attempt to better predict prices. As shown in our initial graphs below, 100 epochs yielded the best results, as higher and lower values led to increased MSE values.





Next steps

Real time trends:
In our initial project proposal, we planned on using real time data to make shorter term predictions (not just next day). A further step could be incorporating hourly or per minute trends into our analysis for the most accurate model. We'd need to experiment with our hyperparameters to maximize our models effectiveness over shorter intervals.

Sentiment Analysis:
We also planned on providing sentiment analysis based on X (formerly Twitter) posts. Thus far, we have based our model on pure statistical predictions. Public sentiment, shaped by collective emotions and perceptions, significantly impacts stock prices. Positive sentiment drives demand and raises prices, reflecting confidence in a company's prospects. Conversely, negative sentiment reduces demand, leading to price declines. X is often considered the contemporary equivalent of the traditional "town square" for public discourse, where people express their opinions, share information, and engage in discussions. Incorporating sentiment analysis could vastly improve our model.