Outline

Headings you add to the document will appear here.

I chose to complete the second prompt, which involved building a model that predicts stock prices based on social media sentiment. I took two dataset from kaggle, one that included tweets that included the top 20 most watched stocks on Yahoo Finance, and another that included information on stock prices (closing/adjusted closing price, high/low/open price, and volume). This dataset only included the data for one year (2021-2022), but with more time I would have scraped twitter data and built a dataset that spanned more years.

After combining the datasets based on date and stock price, I performed some preprocessing on the tweets. Mainly removing stopwords, usernames and the hashtag symbol, and converting everything to lowercase. With more time, I would also have done more preprocessing to remove emojis, which could be problematic when using this text to predict sentiments. I chose not to do any lemmatization/stemming, since BERT models (which I use in the next step to predict sentiments), have shown not to be affected by this type of preprocessing. Finally, I removed any duplicate tweets and rows withna's.

To find sentiment scores for the cleaned tweets, I used two models: a naive bayes classifier from python's textblob package and a pretrained BERT model from hugging face that was trained on financial data (Reuters and Financial PhraseBank). I ran the tweets through both these models, which both outputed a score from -1 (negative sentiment) to 1 (positive sentiment), and then took the average of these scores to construct a final sentiment score. Because these models took a while to make predictions from, I reduced the dataset to only 2000 rows. With more time and resources (mainly a GPU), would have used the entire dataset (around 50k rows) so that I could have a more comprehensive datasets for training my stock prediction models.

I next decided to build a random forest regression model and an xgboost model that would take sentiment score, the stock name, the previous day's adjusted closing price, open price, high, low, and volume to predict the next day's adjusted closing price. To do this I sorted the dataset by date and created a new column called 'Adj Close Next' which took a specific's