Algo Trading

Patrick, Freddie, Terry, Hope, Arzu

Motivation & Summary

- ***Define the core message or hypothesis of your project.***
- ***REQUIRED: Describe the predictive model chosen and why this model was chosen.***

- Hypothesis:
  - Create an algo trading model for buy, sell, hold prediction, using FinViz to identify key stocks given input from current news on positive or negative sentiment in order to maximize returns.
- Models:
  - Vader: to obtain 25 stocks with the most positive scores in current news. This will be our basis for narrowing down to the optimal stocks given FinViz criteria.
  - LSTM

Model Summary

- Required: Elaborate on the predictive model used, describing why it was the best choice for the data.
- Vader
  - In order to begin our algo model, we decided to first look for stocks with a positive sentiment in the news
  - Vader provided us with a sentiment score, with which we were able to decide on the top 25 stocks with the most positive sentiment.
  - This yielded a comprehensive list of stocks to evaluate further using FinViz.
  - Vader was our first choice for preliminary stock choice to gather a larger list of stocks with the propensity for higher returns based on a positive sentiment score.
- Arima
  - We chose ARIMA as our algo-trading model as it was the simplest model with the lowest P-Value at 0.00
  - We attempted Logistic Regression, Linear Regression, GARCH, ARMA, and ARIMA to compare all models to see which would provide us the best results. Out of all models, ARIMA provided the consistent time-series predictions whereas the others varied in consistency and GARCH would have been a good “add-on”, but not a good model in it’s own in order to make time series predictions of buy, sell, hold.
  - We discarded this model in the end as it was not highly predictive and presented evaluation issues.
- LSTM
  - We moved to LSTM for our algo trading model as it was simple but highly predictive and gave us more features to adjust to get a better predictive model. This proved easier to quickly evaluate than ARIMA.

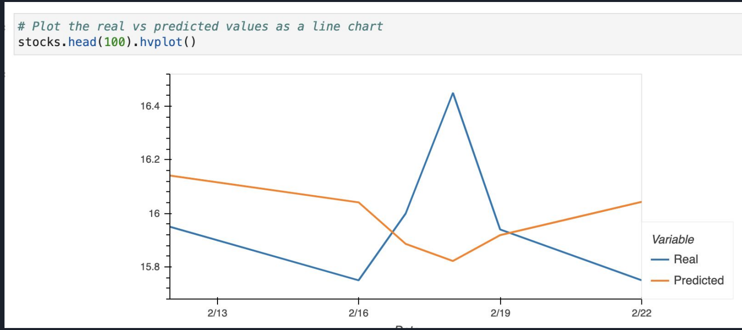
Data Cleanup & Model Training

- Describe the exploration and cleanup process.
  - We pulled stock tickers from full articles using the newsapi. After conducting sentiment analysis, we dropped tickers with negative or neutral sentiments and kept only top 25 tickers with positive sentiment scores over 0.2.
  - We utilized Alpaca to pull stock data on the list of 25 previously obtained. This data was cleaned and pushed through FinViz's screener feature to determine which stocks were optimal for the machine learning model based on the following criteria:
    1. Performance : Today +10%
    2. Current Volume : Over 10M
    3. Country : USA
- Discuss any problems that arose with preparing the data or training the model that you didn't anticipate.
  - Upon training data and obtaining prediction output, we discovered the model would react differently given different stocks as we could only insert one stock into the model at a time. We then discovered a For Loop from https://handsoffinvesting.com/an-algorithmic-trading-strategy-using-logistic-regression/ that would alow us to run multiple tickers at once.
  - We had issues with Alpaca being down which made it impossible to run and test our code, but luckily it returned. This forced us to consider using other data sources like Pandas Data Reader or Yahoo Finance.
  - We ran into issues with ARIMA as it wasn't truly a predictive model and thus did not return the predictions we were looking for. This led us to start from ground zero utilizing the Long Short Term Memory model.
- Discuss the overall training process and highlight anything of interest with the training process: Cloud resources used, training time required, issues with training.
  - ARIMA
    1. After receiving the price predictions, we found code that would have allowed us to create a dynamic & functional paper trade using the alpaca api.
    2. The most difficulties arose from comparison of our predicted prices to original real-time prices. Putting them into a DataFrame in order to assess percentage up/down proved time consuming.
  - LSTM
    1. We pulled data from Yahoo API due to difficulties with Alpaca api not supporting adjusted close.
    2. Split ratio changed the mode significantly. The more we increase the split ratio, the more accurate our model became, however, the higher we increased our split ratio, the less data we had for backtesting.
    3. It broke Patrick's computer 3 times.

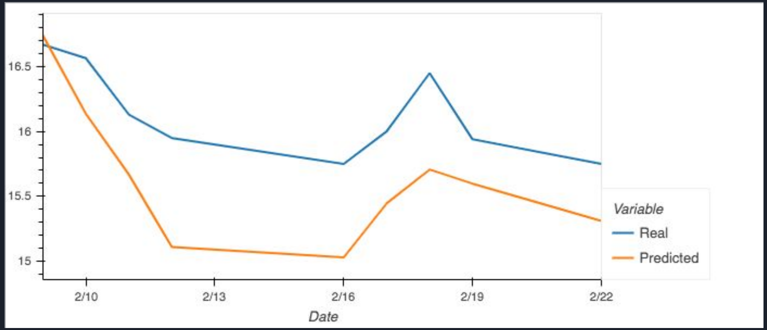
Model Evaluation

- ARIMA Model Performance:
  - AR and MA terms have p values of zero, which suggests that they are useful to predict prices.
  - Arima (1,1,1) has the lowest AIC & BIC.
- LSTM Model Performance
  - Our first LSTM model gave 19% loss so to adjust it we...
  - Adjusted:
    - split down 0.1 to 0.7,
    - Dropout layer down 0.5 to 0.2
    - Resulting in lower 18% loss
- Model Summary
  - Our Vader model returned a selection of tickers with positive sentiments that allowed us to narrow our search for tickers worth trading.
  - Our ARIMA model gave us price predictions that would have allowed us to make buy/sell/hold decisions per stock, but not as dynamically as we would have liked.
  - Our initial LSTM model resulted in 19% loss and as we adjusted we obtained a lower loss at 18% thus showing lower split, lower layer yielded better results.
  - We evaluated our model by Mean Absolute Error that computes the mean absolute error between the labels and predictions. For our 18% loss model, we obtained a 0.35 Mean Absolute Error. This is fairly low which we want to see!

Real Prices Compared to Predicted Prices 19% Loss Model



Real Prices Compared to Predicted Prices 18% Loss Model



Discussion

- Discuss the implications of your findings.
  - We did not choose a deep learning model at first due to lack of features required for this type of model. We chose ARIMA Logistic Regression due to the simplicity of the classification required for our hypothesis.
  - We then found that this model was too simplistic and not highly predictive, so we utilized LSTM which yielded better/more consistent results that were easily analyzed and gave us more opportunity to fine tune the model using dropouts, units, and splits.
  - Our findings allowed us to confidently predict if we should buy a certain stock after first undergoing a selection process comprised of sentiment analysis and FinViz qualifications. In the end, it would be interesting to remove one of these attributes, either FinViz qualifications or the sentiment analysis and see to what extent our model is affected.

Postmortem

- Discuss any difficulties that arose, and how you dealt with them.
  - Difficulty getting scanner data into Vader model due to formatting.
  - Api issues: too many calls within 24 hours
  - We had to change our search criteria on FinViz scan to compensate the API issue, which gave us less stocks in return.
  - Using data from multiple stocks for LSTM
- Discuss any additional questions or problems that came up but you didn't have time to answer: What would you research next if you had two more weeks?
  - Would choosing negative sentiment stocks yield expected negative results/returns? Does filtering by positive sentiment truly help in choosing buy,sell,hold?
  - What occurs with neutral stocks?
  - Would using different indicators or alternative differencing figures result in a better model and thus higher returns?
  - Would fine tuning the LSTM model yield a highly predictive/productive model?

Questions?

- How poorly did we do?
- Do you think we are smarter than you because of this model?
- Would you hire me?
- Would you hire you?