NEURAL NETWORKS AND DEEP LEARNING FOR FINANCE

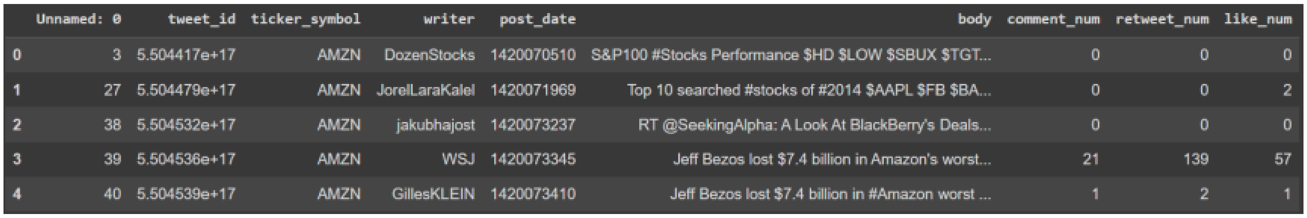
February -April 2023

Predicting Amazon stock price with text mining of tweets

Introduction

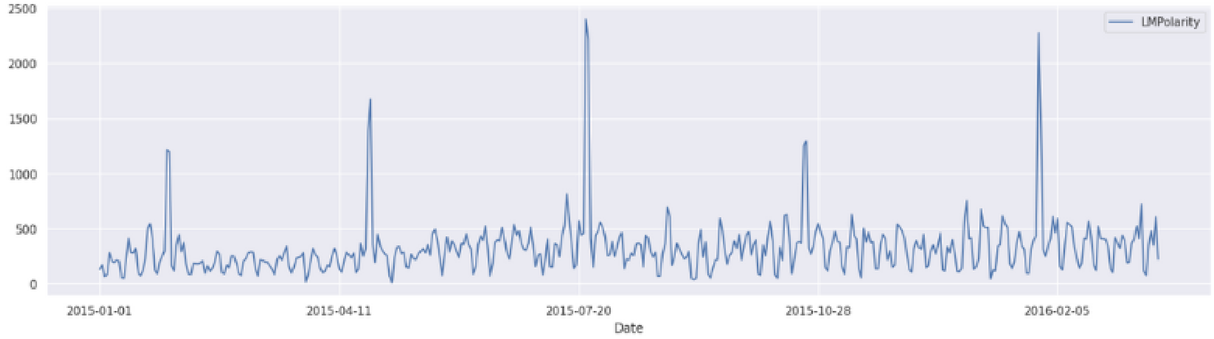
In recent years deep learning technique have been an extremely discussed topic and their usefulness to make predictions and model variables has been put in evidence. Especially, neural networks have become a very popular technique to make forecasts in finance. Throughout this course, we have been studying and experimenting with this deep learning technique and for our final project, the team was motivated to use them to try to predict the stock prices of one of the largest and most profitable companies of our time, Amazon. Additionally, in this semesters' courses we had the opportunity to dive into text mining techniques. Text mining is useful when one wishes to extract information from non-numeric data, specifically, text data. An important source of soft information is nowadays undoubtedly social media. In particular, Twitter is a place where people from all different backgrounds share their thoughts and beliefs through text posts. Without question, Twitter contains valuable information that could pinpoint where the market is heading. This includes information that could signal if a particular company's stock will go up or down as tweets that mention it can provide useful insights no other source of information can provide. This is why the team was motivated to combine text mining techniques with neural networks; as a way to mix all the knowledge acquired this semester into one final project. Our best model was able to make accurate predictions on the data provided and we took the liberty to make forecasts for 10 future periods, giving the investors who could use it a confidence interval comprehended by a pessimistic and an optimistic scenario (which the investors could replicate for even further periods as they obtain more data). Amazon is one of the most lucrative companies that offers its investors very competitive returns in comparison to the market. Therefore, our model (which is fairly accurate) is extremely valuable to investors who want to stay ahead of the market and have insights on what Amazon's stock price will be before analysts even have a forecast.

Data

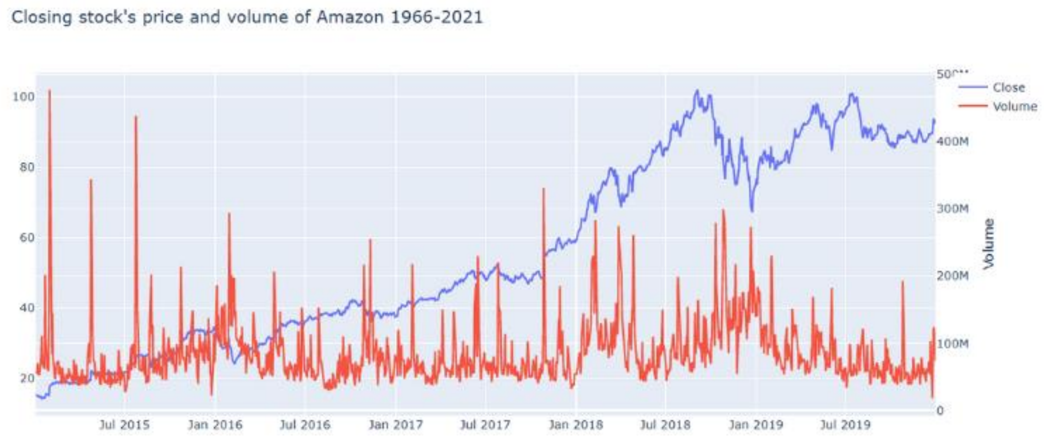


	Unnamed: 0	tweet_id	ticker_symbol	writer	post_date	body	comment_num	retweet_num	like_num
0	3	5.504417e+17	AMZN	DozenStocks	1420070510	S&P100 #Stocks Performance $HD $LOW $SBUX $TGT...	0	0	0
1	27	5.504479e+17	AMZN	JorelLaraKalel	1420071969	Top 10 searched #stocks of #2014 $AAPL $FB $BA...	0	0	2
2	38	5.504532e+17	AMZN	jakubhajost	1420073237	RT @SeekingAlpha: A Look At BlackBerry's Deals...	0	0	0
3	39	5.504536e+17	AMZN	WSJ	1420073345	Jeff Bezos lost $7.4 billion in Amazon's worst...	21	139	57
4	40	5.504539e+17	AMZN	GillesKLEIN	1420073410	Jeff Bezos lost $7.4 billion in #Amazon worst ...	1	2	1

Our database for this project comprehends 5 years of tweets (in daily frequency) that mention Amazon. We have information such as tweet id, the Twitter user, the date of publication, the content of the tweet, among others. Additionally, we possess information about the Twitter activity per day which is illustrated in the table below.



It is possible to see that the activity is mostly stationary except for a few noticeable spikes when it is very likely that big events were taking place in the world. We also have at hand information pertaining Amazon's stock such as close price and volume.



We notice an overall ascending trend. This is consistent with the financial growth Amazon has been experiencing ever since its foundation as their services revolutionized the commerce industry.

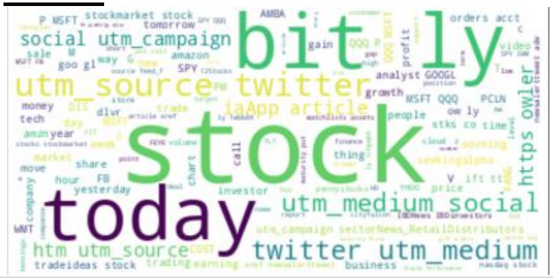
The Deep Learning Model

We started by text mining the tweets to separate the useful words from the stop words (repeated words that do not give valuable information such as "the", "with", "or", etc.) and to classify them by type. We ended up with four groups, the stop words (which would not be used), the nouns, the proper nouns, and the verbs. The word clouds below visually illustrate how the words in the tweets were classified.

<u>Stop words</u>

bit ly
Amazon
dlvr
stks co
AMZN NFLX
AMZN AMZN
GOOG AMZN
AMZN FB
ift tt
AMZN AAPL
AMZN Amazon
AAPL AMZN
X CROSSFINDER
NFLX AMZN
SIGMA X
seekingalpha
ATS LX
HFT Algo
Algos triggered
T Quant
CROSSFINDER ATS
AMZN
FB
Amazon AMZN
P T

<u>Nouns</u>



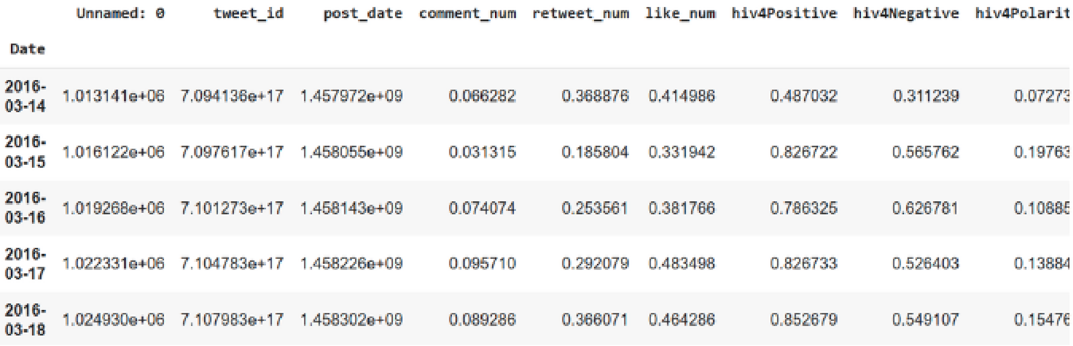
Proper Nouns

AMZN
Amazon
SIGMA
CROSSFINDER
NFLX
ATS
LX
Algos
HFT
Algo
Inc
TWTR
BABA
dlvr
EBAY

Verbs

GOOGL
Updated
dlvr
triggered
SELL
buy
invest
investing
look
think
hold
make
Read

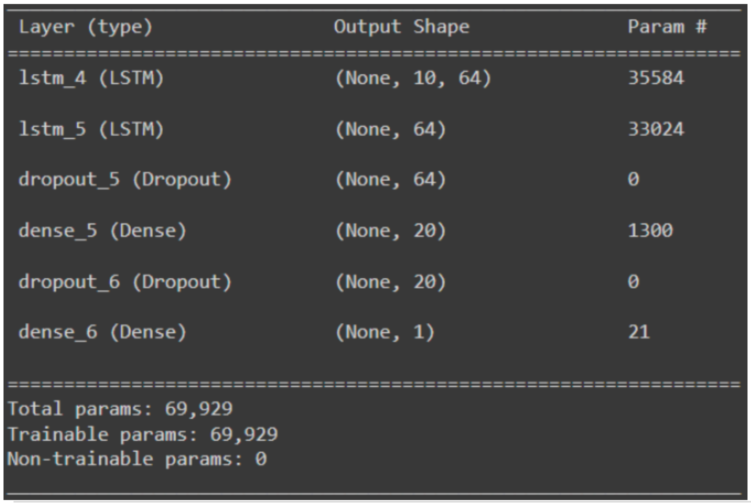
To take into account market sentiment instead of just the tokenized words, we decided to create variables that indicated what the Twitter community was feeling about Amazon, its performance and its stock. If the market was feeling optimistic about Amazon (positive tweets) it would be expected that the stock price would rise and if the market was feeling pessimistic about it (negative tweets), it would be expected that the stock price would decline. Below is a visualization of these new variables we would train the neural network model with.



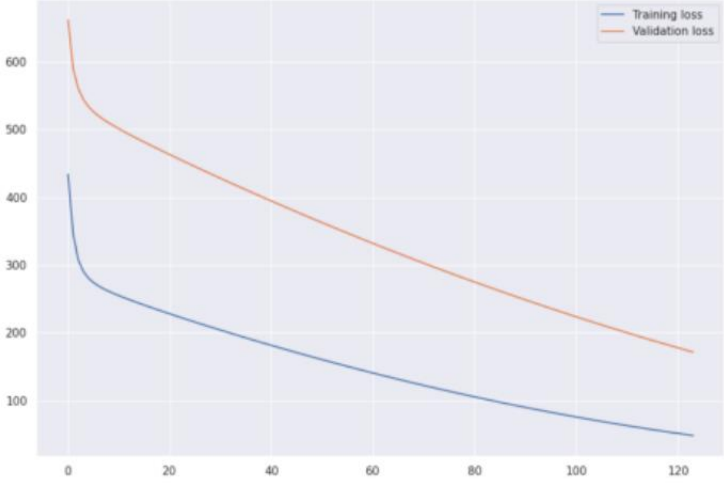
	Unnamed: 0	tweet_id	post_date	comment_num	retweet_num	like_num	hiv4Positive	hiv4Negative	hiv4Polarit
Date									
2016-03-14	1.013141e+06	7.094136e+17	1.457972e+09	0.066282	0.368876	0.414986	0.487032	0.311239	0.07273
2016-03-15	1.016122e+06	7.097617e+17	1.458055e+09	0.031315	0.185804	0.331942	0.826722	0.565762	0.19763
2016-03-16	1.019268e+06	7.101273e+17	1.458143e+09	0.074074	0.253561	0.381766	0.786325	0.626781	0.10885
2016-03-17	1.022331e+06	7.104783e+17	1.458226e+09	0.095710	0.292079	0.483498	0.826733	0.526403	0.13884
2016-03-18	1.024930e+06	7.107983e+17	1.458302e+09	0.089286	0.366071	0.464286	0.852679	0.549107	0.15476

We proceeded to split the data into train and test to estimate the following models.

Model 1:

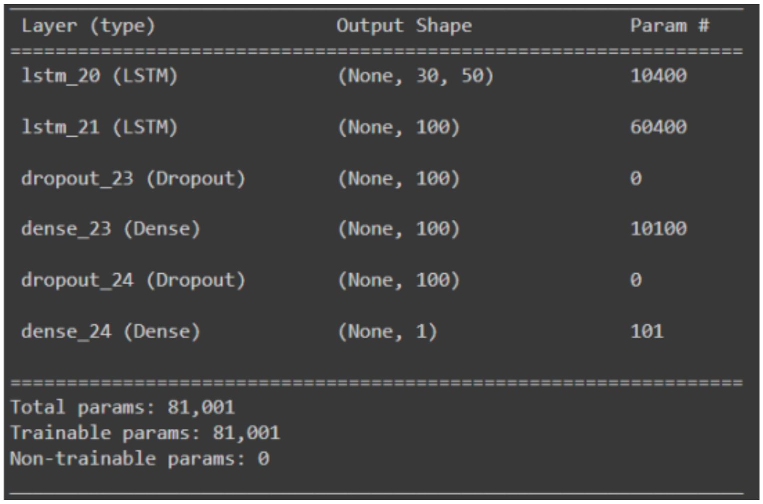


```
 Layer (type)                Output Shape              Param #
=================================================================
 lstm_4 (LSTM)               (None, 10, 64)            35584

 lstm_5 (LSTM)               (None, 64)                33024

 dropout_5 (Dropout)         (None, 64)                0

 dense_5 (Dense)             (None, 20)                1300

 dropout_6 (Dropout)         (None, 20)                0

 dense_6 (Dense)             (None, 1)                 21

=================================================================
Total params: 69,929
Trainable params: 69,929
Non-trainable params: 0
```

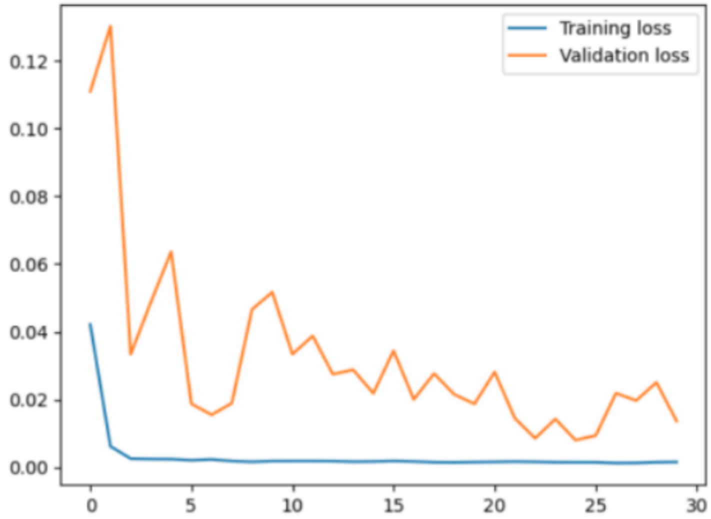


The first model is composed by two long short term memory layers, two dropouts (which randomly drop inputs), and two dense layers. The number of nodes varies from 20 to 64. In total, this model has 69,929 trainable parameters. It is observed in the graph of the loss, that this takes a long time to be minimized, so we were encouraged to try other models with different structures to see if the loss could be minimized quicker and therefore obtain a model with better predictive capacity.

Model 2:

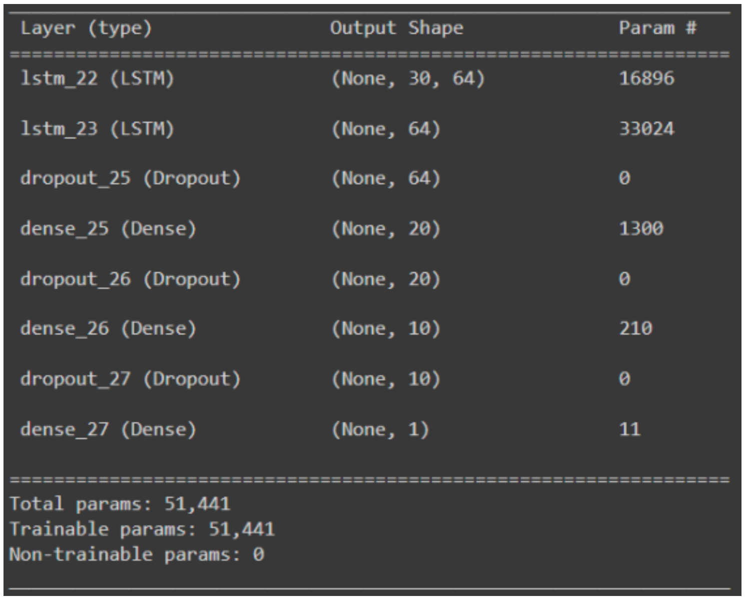


```
 Layer (type)                Output Shape              Param #
=================================================================
 lstm_20 (LSTM)              (None, 30, 50)            10400

 lstm_21 (LSTM)              (None, 100)               60400

 dropout_23 (Dropout)        (None, 100)               0

 dense_23 (Dense)            (None, 100)               10100

 dropout_24 (Dropout)        (None, 100)               0

 dense_24 (Dense)            (None, 1)                 101

=================================================================
Total params: 81,001
Trainable params: 81,001
Non-trainable params: 0
```

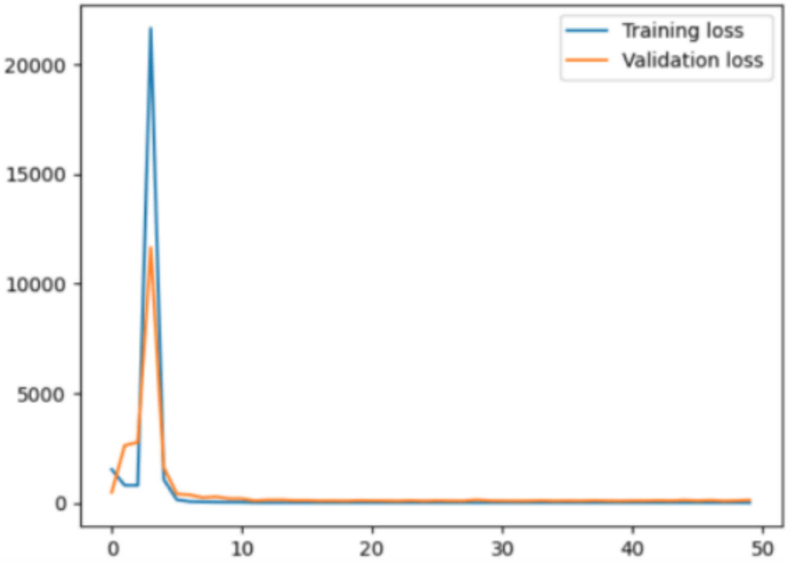


This model had more nodes inside the hidden layers and it retuned more trainable parameters (81,001). The graph of losses shows that it is minimized quicker so this might be a better model than model 1.

Model 3:



```
 Layer (type)                Output Shape              Param #
=================================================================
 lstm_22 (LSTM)              (None, 30, 64)            16896

 lstm_23 (LSTM)              (None, 64)                33024

 dropout_25 (Dropout)        (None, 64)                0

 dense_25 (Dense)            (None, 20)                1300

 dropout_26 (Dropout)        (None, 20)                0

 dense_26 (Dense)            (None, 10)                210

 dropout_27 (Dropout)        (None, 10)                0

 dense_27 (Dense)            (None, 1)                 11

=================================================================
Total params: 51,441
Trainable params: 51,441
Non-trainable params: 0
```

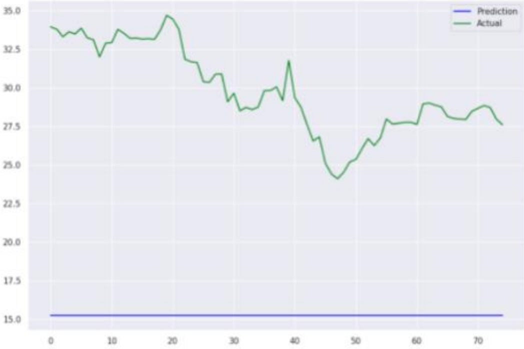


We specified this model with one additional dense layer and one additional dropout layer. There were fewer nodes and it returned the least amount of trainable parameters (51,441). The graph portrays that the losses escalate drastically at first but then they are minimized quickly.

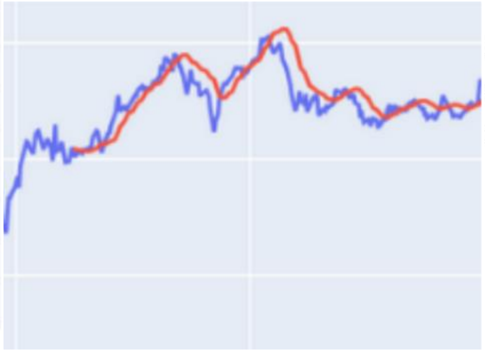
Results

Below are the models put into action. We can see that model one needs to be discarded because it has no predictive ability at all. Model 2 adapts well to the model but unfortunately is overfitting the data which can make it inappropriate when investors try it out with new data. Hence, Model 3 might be the best fitting model. It is a bit underfitting but it is able to anticipate the direction of the price with accuracy, which is valuable for the investor.

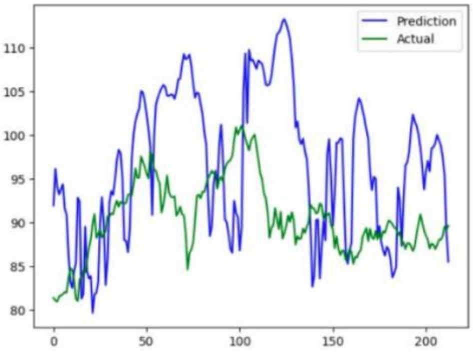
Model 1:



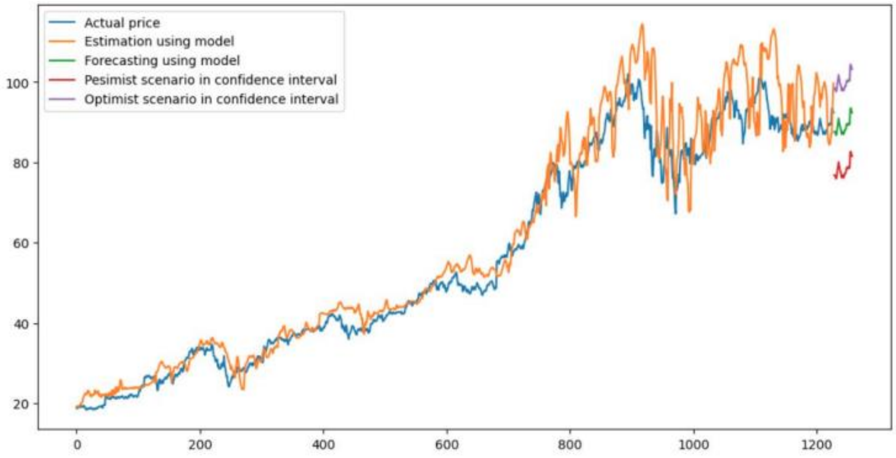
Model 2:



Model 3:



Model 3 applied on the whole dataset:



When applied to the whole dataset, we can see that our third model is very good at predicting Amazon's stock price. It is somewhat overreacting to the volatility of the stock but this can be attributed that (as this is a time series), the training part (the first chunk) followed a fairly stable upward trend, while for the test part there was more volatility. We offer the investors forecasts for the following 10 periods which indicate in this case that the stock price is likely to slightly rise.

Conclusion

We were able to create a neural network model that predicted Amazon's stock price with accuracy using sentiment analysis variables extracted from text mining tweets. Our project was however significantly limited by computational power as we ran into a lot of errors due to disconnection, system overwhelming and overheating. If we had a more powerful computer, we could have run more iterations of models to explore other options that could be better fits and give more accurate predictions. We recommend future works to invest in higher computational power, to use grid search instead of random search (which we used as it required less power and time), add other variables to the model that are not related to tweets to compliment it, and to try out other deep learning and machine learning techniques to estimate the models to explore if they yield better results.