Twitter Sentiment Analysis for Bitcoin Price Prediction

MPCS 53113 Natural Language Processing

August 17th 2022

Github Link: https://github.com/vipul-k/NLP-Project

Summary

I try to predict the direction of Bitcoin price movement using Twitter sentiment analysis. I start with sentiment analysis using VADER and plot several graphs to understand the correlation between the price movement and VADER generated sentiment scores. Then I train a model to predict the sentiment of tweets. After this I train models to predict the direction of bitcoin price movement where target is generated using different time frames. Lastly, I use FinBERT to predict the sentiment of tweets. I use my personal machine and Google Colab platform for all the work.

Introduction

All investing carries a certain degree of risk. While an investor might believe an asset's value will increase over time, when they purchase an asset it is almost impossible to know whether they will gain or lose money on that investment. Being able to predict the future price movement of an asset is consequentially an extremely powerful tool for any investor.

2021 had seen extremely volatile asset prices for both "meme stocks" and cryptocurrencies. These price movements seem to be completely removed from the asset's underlying value, and are instead shaped largely by the sentiment of investors. Being able to measure the sentiment of the average investor could, therefore, be useful in making projections about the future movements of asset prices. Since the average sentiment of investors is almost impossible to measure, I decided to test whether the average sentiment of tweets about a specific asset could be used as a substitute.

In order to narrow the scope of the project and improve the usefulness of my model, I decided to focus solely on Bitcoin and its price movements.

Related Work

Due to the significant financial upside of being able to predict the price movements of assets, many researchers have attempted to use online sentiment to predict how the value of stocks and cryptocurrencies will change.

A. Derakhshan and **H.** Beigy's *Sentiment Analysis on Stock Social Media for Stock Price Movement Prediction* focused on a relatively large number of stocks and used both English and Persian text from message boards such as Yahoo Finance [1]. They used both SVM and Neural Network models and attained an average accuracy of 56% in predicting positive or negative price movements, giving me a useful benchmark to compare my results to. In addition to being primarily predicting stock prices rather than cryptocurrencies, this paper also differed from my work by focusing on 18 US equities compared to our single currency. Our decision to focus on a single asset was made in order to increase size of the dataset on which we could train our model. Additionally, we chose to use Twitter due to both ease of data collection and also due to it being a more current platform than the Yahoo Finance dataset, which was collected during 2012 and 2013.

D. R. Pant, et al.'s *Recurrent Neural Network Based Bitcoin Price Prediction by Twitter Sentiment Analysis* was much more similar to the analysis I carried out [2]. Their model used Twitter data as an input and attempted to predict the future price of Bitcoin. However, while their paper achieved a relatively high accuracy it was also reliant on manually labeling tweets. This led to a relatively small dataset of 7454 Tweets, 1669 of which were deemed neutral or irrelevant. To increase the size of dataset and potentially increase the usefulness of model, I decided to instead use future price movements to automatically label the tweets. Additionally, their model also incorporated historical price movements in their predictions. I decided not to do the same in order to test the predictive power of only using Twitter sentiment. J. Devlin, et al.'s *BERT- Pre-Training of Deep Bidirectional Transformers for Language Understand- ing* provided details on the usage of the BERT model utilized in my experiments [3]. Additionally, Sepp Hochreiter, et al.'s Long Short-Term Memory sheds light on LSTM architecture that I have used in my model.

Datasets and Features

Tweets

I initially planned to use the Twitter API to download tweets mentioning Bitcoin over a two-week period to create a dataset. Unfortunately, the API placed a cap on the number of tweets I could download, meaning I was initially limited to a few thousand tweets per day. To expand this into a more generalizable dataset, I combined the tweets I had already downloaded with a larger dataset from Kaggle [5]. All tweets in this combined dataset mentioned Bitcoin in either a hashtag or in the main text of the tweet and were published between 5th February 2021 and 3rd July 2022. This dataset contained 3,840,626 tweets. I then used conditions 1. "user_followers" > 1000 and "user_favourites" > 1000 to filter data. I also included tweets from verified accounts irrespective of above conditions. This step left me with 457,407 tweets. Following this I cleaned the text of the tweets by removing hashtags(#), mentions(@), hyperlinks and special characters. I used embeddings to extract features from text. Various nltk tools like 'wordnet', 'stopwords', 'punkt' and 'WordNetLemmatizer' were used to further process the data before ingesting in models. Following is a snapshot of data:

	date	text
0	2021-04-12	the why behind microsoft s 19 billion nuance ...
1	2021-04-12	what are the biggest shitcoins
2	2021-04-12	market is weakly trending up current moment...
3	2021-04-12	time to get in folks massive bull run is sta...
4	2021-04-12	ewt btc eth dot ada link snx inj band...

Bitcoin Price

I collected historical Bitcoin price data from Yahoo finance API. This data was at daily level and contained following prices High, Low, Open and Close along with daily Volume. To create target I checked the ratio of current price to the price n days ahead. Here n is the number of days after which I intend to predict if price moved up or down. Based on this if this ratio is greater than 1 the target was assigned value positive or if the ration is less than 1 then negative.

Analysis

Vader

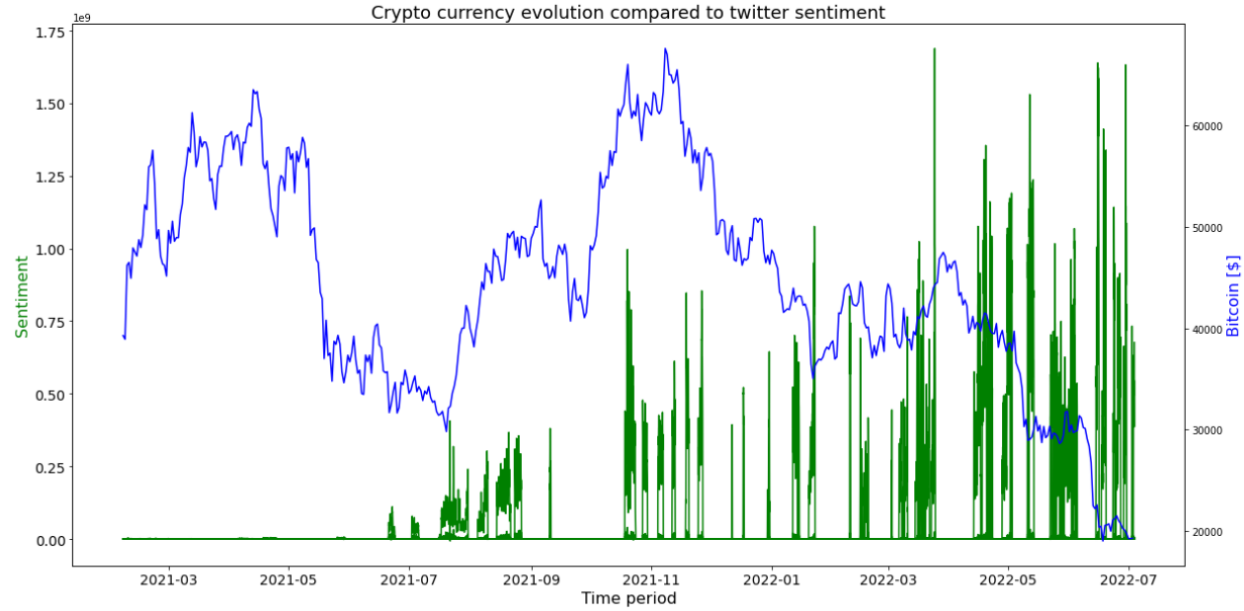
VADER (Valence Aware Dictionary for Sentiment Reasoning) is a model used for text sentiment analysis that is sensitive to both polarity (positive/negative) and intensity (strength) of emotion. It is available in the NLTK package and can be applied directly to unlabeled text data.

VADER sentimental analysis relies on a dictionary that maps lexical features to emotion intensities known as sentiment scores. The sentiment score of a text can be obtained by summing up the intensity of each word in the text.

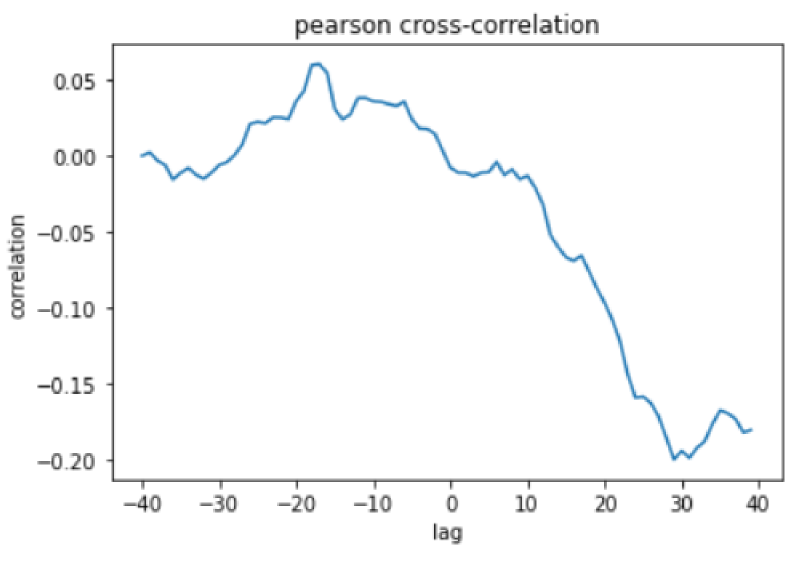
For example- Words like *'love', 'enjoy', 'happy', 'like'* all convey a positive sentiment. Also VADER is intelligent enough to understand the basic context of these words, such as *"did not love"* as a negative statement. It also understands the emphasis of capitalization and punctuation, such as *"ENJOY".*

Vader compound sentiment score ranges from -1, implying very negative, and 1, implying very positive.

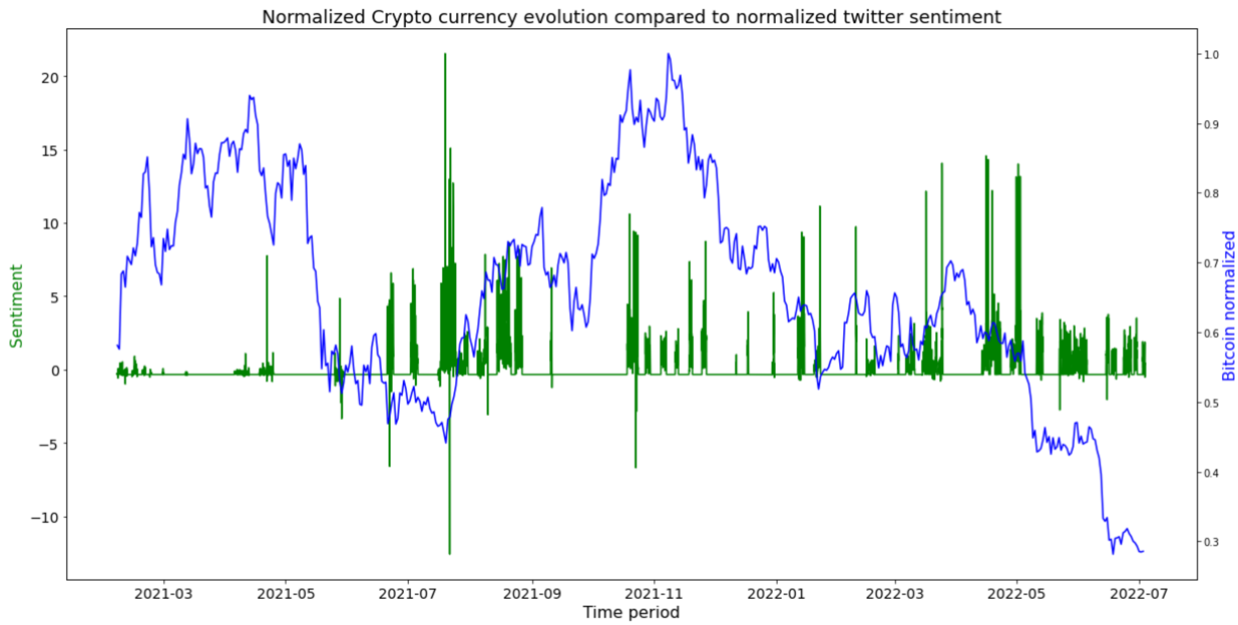
I calculated the Vader Compound Sentiment score for each tweet in my dataset and then multiplied it with number of likes to take into account the effect of engagement of the tweet. Then I plotted the price of bitcoin to aggregated sentiment score over a day.



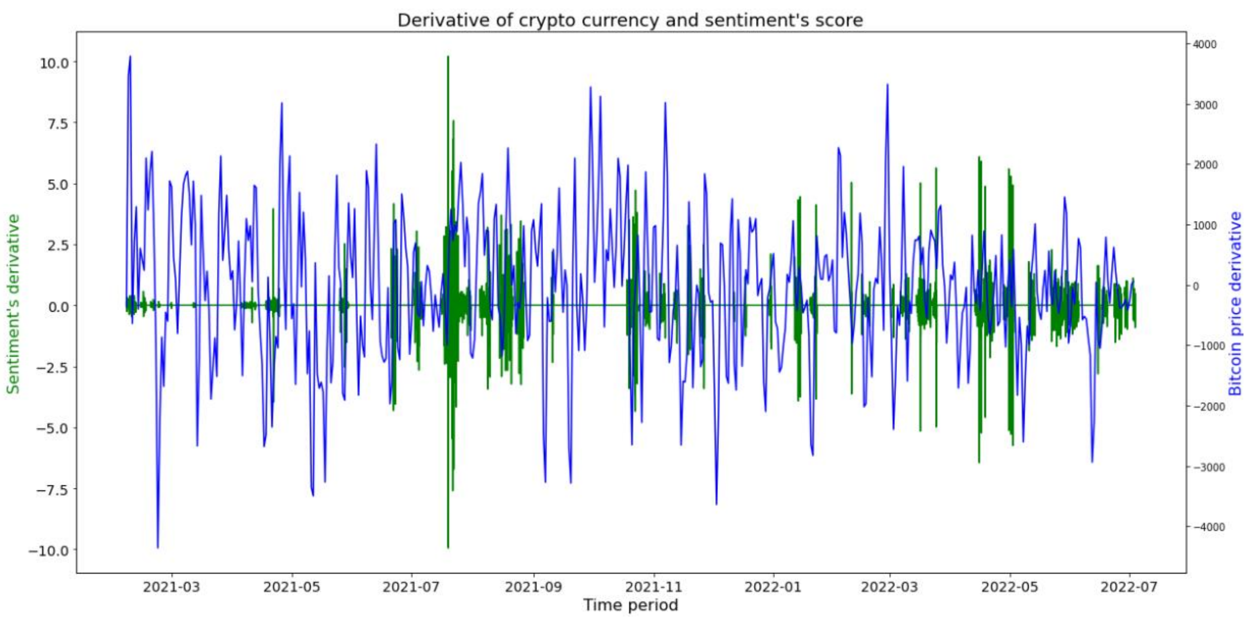
In the above plot I observe that positive sentiment shows a correlation with downwards trend in price of bitcoin. Also, sentiment scores are very high in 2022 which needs further investigation. Further following graph shows correlation between Bitcoin price and sentiment score varying lag between two from -30 to 30 days.



Strongest correlation for forward-looking price prediction happens around a lag of -20 days. I also plotted the sentiment and price graphs after normalizing the sentiment score and price to take into account the volume of tweets.



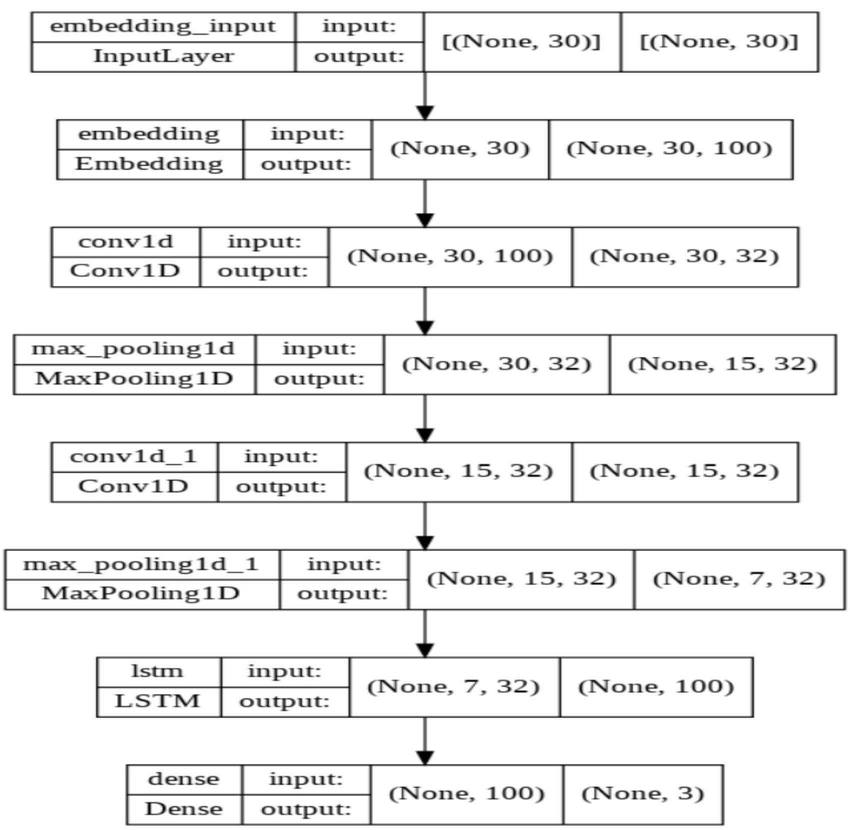
In above plot spikes in 2022 come with less amplitude showing that they were caused by increase in tweet volume. Next, I also plotted the gradient of sentiment score and price, though, no real conclusions are apparent from it.



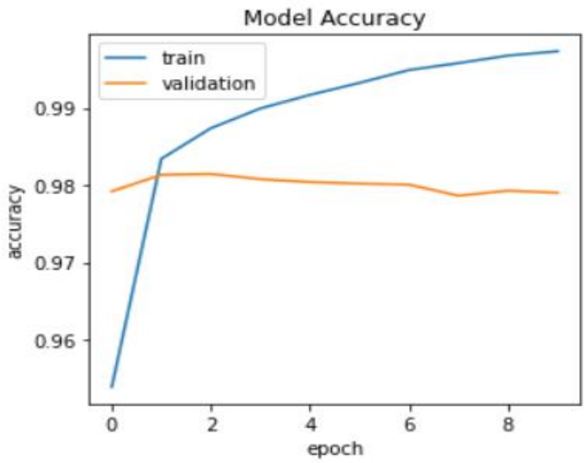
Sentiment Prediction

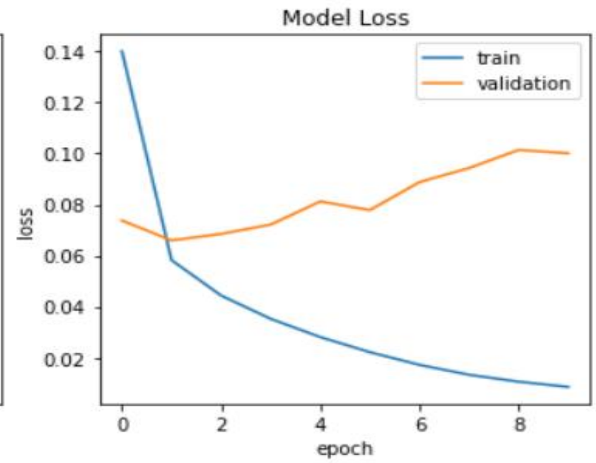
Then I used tensorflow, keras and sklearn to architect the model. I used the TEXTBLOB sentiment polarity function to label each tweet as positive, negative and neutral. I did 80:20 train test split on data.

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding (Embedding)       (None, 30, 100)           2000000

 conv1d (Conv1D)             (None, 30, 32)            9632

 max_pooling1d (MaxPooling1D  (None, 15, 32)           0
 )

 conv1d_1 (Conv1D)           (None, 15, 32)            3104

 max_pooling1d_1 (MaxPooling  (None, 7, 32)            0
 1D)

 lstm (LSTM)                 (None, 100)               53200

 dense (Dense)               (None, 3)                 303

=================================================================
Total params: 2,066,239
Trainable params: 2,066,239
Non-trainable params: 0
_________________________________________________________________
```

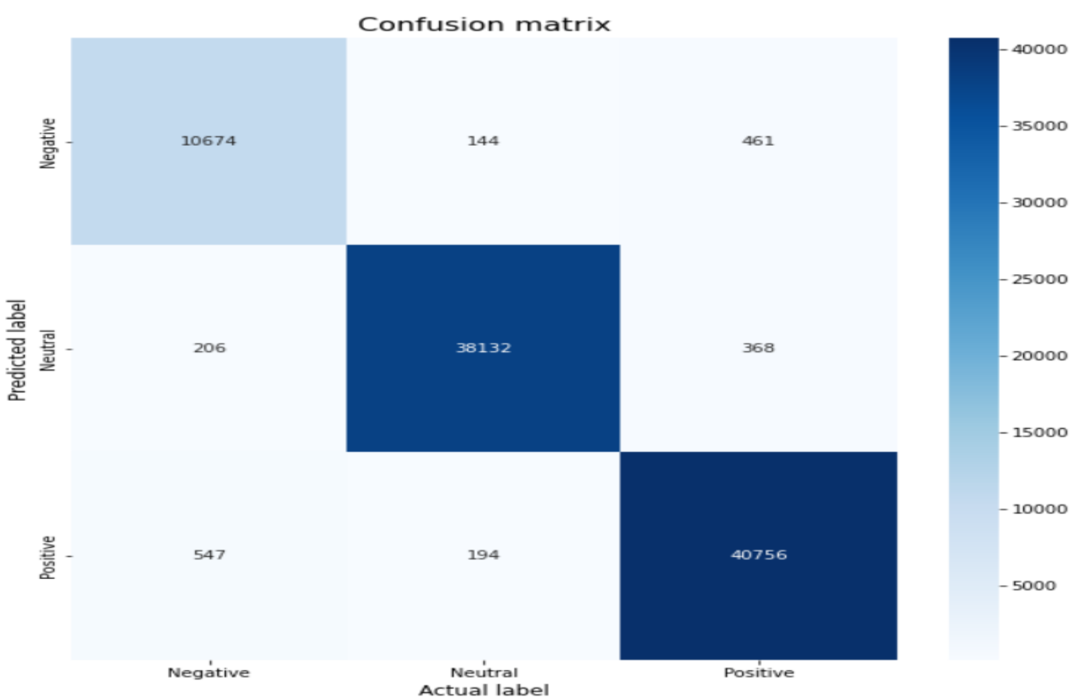


Performance






```
Accuracy:        97.9%
              precision    recall  f1-score   support

           0       0.95      0.93      0.94     11427
           1       0.99      0.99      0.99     38470
           2       0.98      0.98      0.98     41585

    accuracy                           0.98     91482
   macro avg       0.97      0.97      0.97     91482
weighted avg       0.98      0.98      0.98     91482
```



The model performed very well with an accuracy of 97.9%. Though the labels I used are using textblob and a manual labelling may be better to check if the sentiments being predicted are actually how accurate.

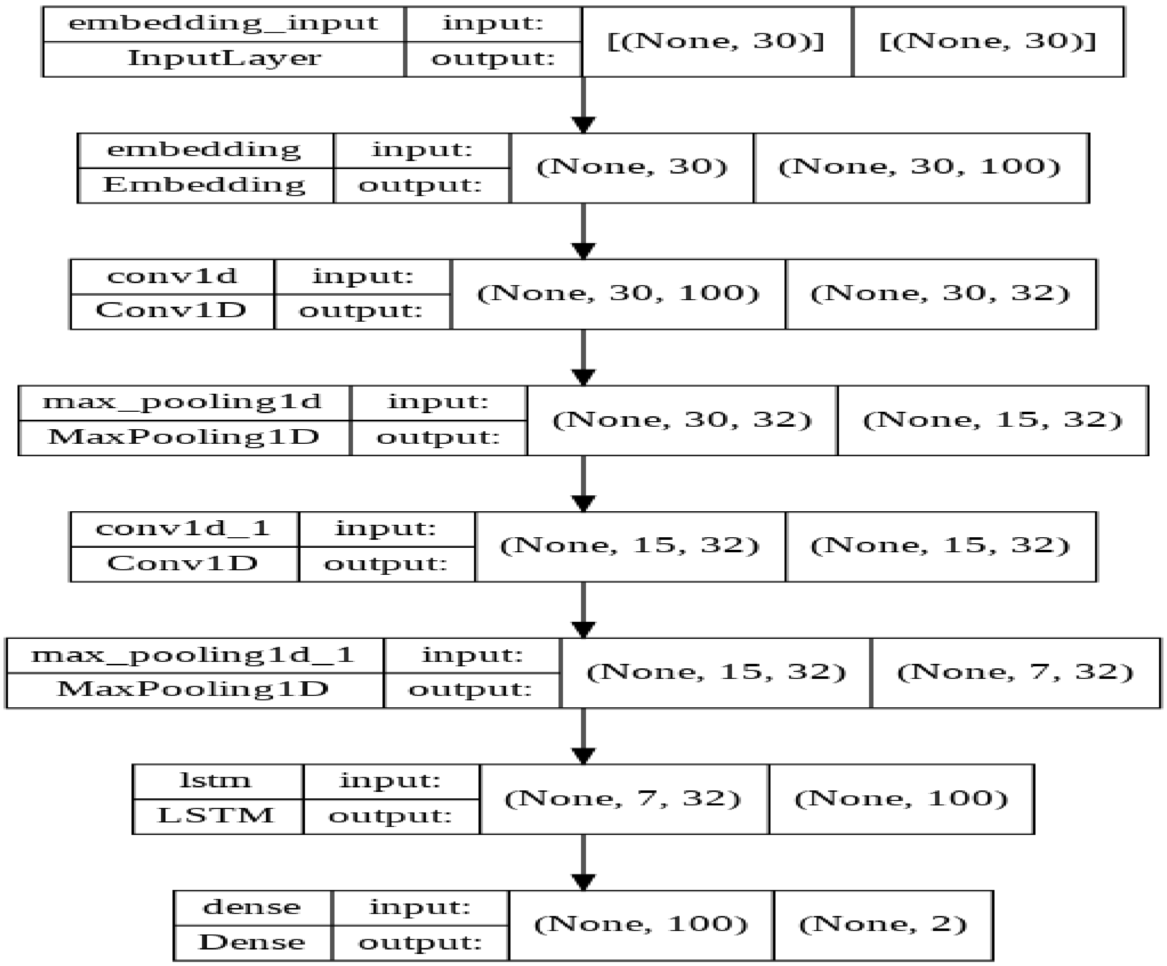
Predicting the price movement

I trained a couple of models for price prediction with primary difference being the how the target is defined. For first the target is defined using price action over the next seven days while for the second price action is defined using price action over next 20 days.

Model Architecture

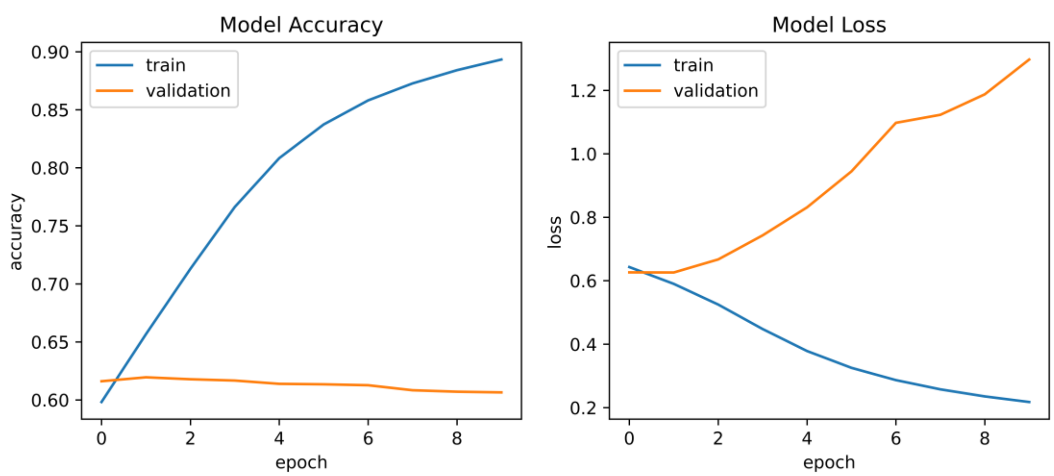
```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding (Embedding)       (None, 30, 100)           2000000

 conv1d (Conv1D)             (None, 30, 32)            9632

 max_pooling1d (MaxPooling1D  (None, 15, 32)           0
 )

 conv1d_1 (Conv1D)           (None, 15, 32)            3104

 max_pooling1d_1 (MaxPooling  (None, 7, 32)            0
 1D)

 lstm (LSTM)                 (None, 100)               53200

 dense (Dense)               (None, 2)                 202

=================================================================
Total params: 2,066,138
Trainable params: 2,066,138
Non-trainable params: 0
_________________________________________________________________
```

Model starts with an embedding layer with max_features = 20000 and embed_dim = 100. This followed by a couple of 1D convolution layers and max pooling layers, finally followed by a LSTM.



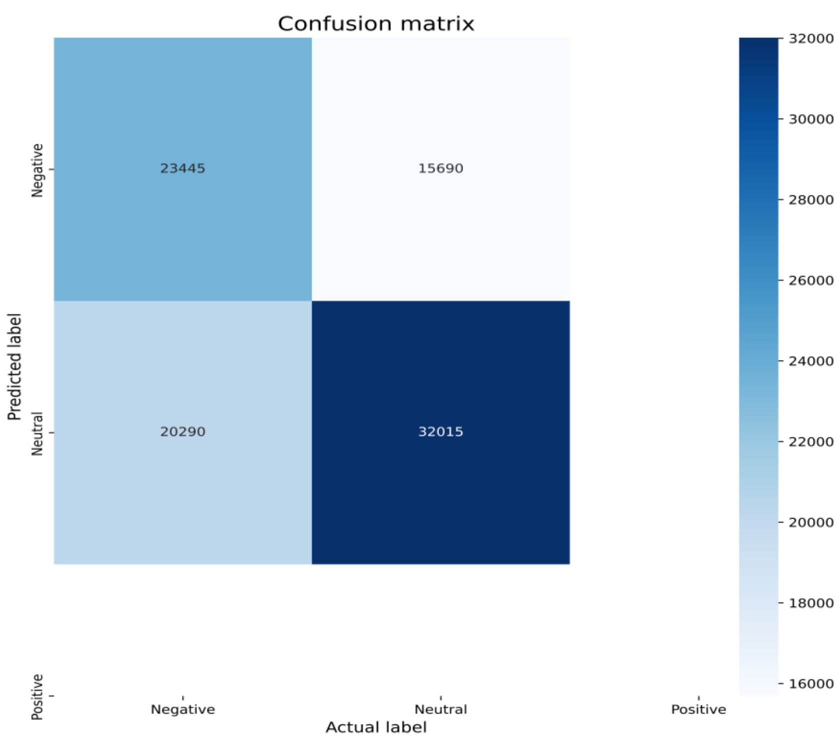
Performance

Target: Movement over 7 days



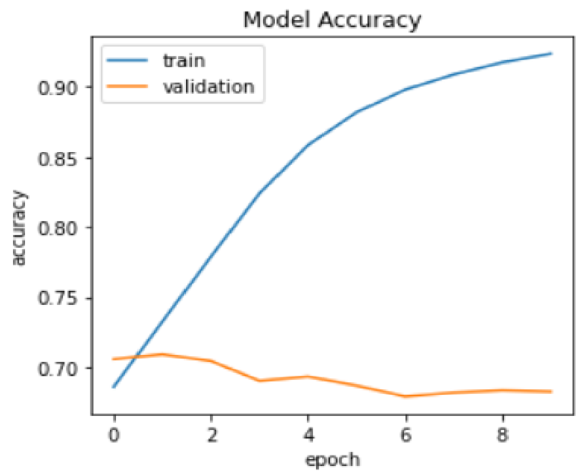
Accuracy: 60.7%

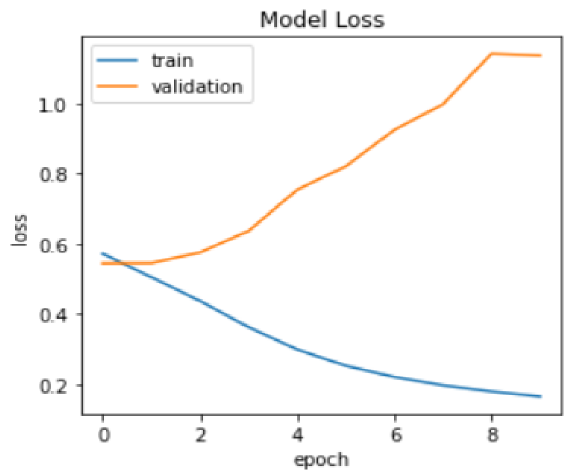
	precision	recall	f1-score
0	0.60	0.54	0.57
1	0.61	0.67	0.64
accuracy			0.61
macro avg	0.61	0.60	0.60
weighted avg	0.61	0.61	0.60



We get an accuracy of 60.7% which is higher than 56% by A. Derakhshan and **H.** Beigy's *Sentiment Analysis on Stock Social Media for Stock Price Movement Prediction* that we are using that we are using as benchmark.

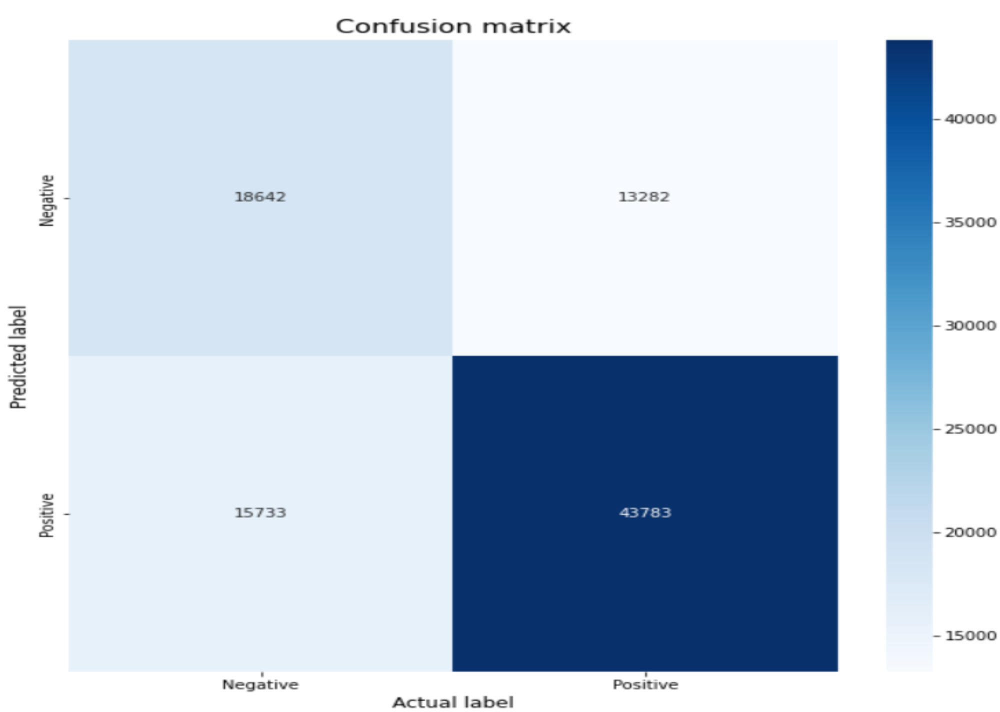
Target: Movement over 20 days





Accuracy: 68.3%

	precision	recall	f1-score
0	0.58	0.54	0.56
1	0.74	0.77	0.75
accuracy			0.68
macro avg	0.66	0.65	0.66
weighted avg	0.68	0.68	0.68



Conclusion/ Future work

Changing target from 7 day price movement to 20 day price movement improves the accuracy of the model by 7%. This makes sense as the plot of correlation between the sentiment score and the price had highest correlation at -20 days lag. In trading/investing having a win rate of over 50% usually results in gains if losses are managed using strict stop loss. Since the model predicts correct direction of price movement in 68% of the cases it provides a strong ingredient for making decision regarding trades in bitcoin.

Further I had started doing sentiment analysis using the FinBERT. FinBERT is a pre-trained NLP model to analyze sentiment of financial text. It is built by further training the BERT language model in the finance domain, using a large financial corpus and thereby fine-tuning it for financial sentiment classification. Financial PhraseBank by Malo et al. (2014) is used for fine-tuning.

Interesting example: FinBERT classifies this as positive while Vader classifies it as negative

1. "market is weakly trending up current momentum suggests the market is neutral"
2. "btc finally surges over the 60 000 mark with no hint of a market top"

Above examples show the importance of context when it comes to language models.

Once FinBERT has been used to generate sentiment scores for all the tweets in the dataset, it can be used to carry out a comparison to scores from VADER. This can shed light on effectiveness of lexicon based approach against pre trained model. Also analysis can be carried out to check if FinBERT sentiment is an indicator of forward looking price movement and how those predictions compare to my model.

References

[1] A. Derakhshan and H. Beigy. "Sentiment Analysis on Stock Social Media for Stock Price Move- ment Prediction." Engineering Applications of Artificial Intelligence, vol. 85, 2019, pp. 569-578., doi:10.1016/j.engappai.2019.07.002.

[2] D.R. Pant, P. Neupane, A. Poudel, A. **K.** Pokhrel and B. K. Lama, "Recurrent Neural Network Based Bitcoin Price Prediction by Twitter Sentiment Analysis," 2018 IEEE 3rd International Conference on Computing, Communication and Security (ICCCS), 2018, pp. 128-132, doi: 10.1109/CCCS.2018.8586824.

[3] J. Devlin, et al. "BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding." Proceedings of the 2019 Conference of the North, 2019, doi:10.18653/vl/nl9-1423.

[4] R. Wigglesworth. "Rise of the Retail Army: the Amateur Traders Transforming Markets." Subscribe to Read I Financial Times, Financial Times, 9 Mar. 2021, www.ft.com/content/7a91e3ea- b9ec-461 l-9a03-a8dd3b8bddb5.

[5] K. Suresh, "Bitcoin Tweets, Version 8," Kaggle, 13-Feb-2021. [Online]. Available: https://www.kaggle.com/kaushiksuresh147/bitcoin-tweets. [Accessed: 28-Jun-2021].

[6] https://finance.yahoo.com/quote/BTC-USD/history