Jack Flitcroft

Nathan Chapman

5/9/2022

Project Summary:

Predicting the stock market or cryptocurrencies is a big topic in tech discussions today. Our project revolves around classifying tweets based on sentiment, and using that generated model to get an overall sentiment of the current cryptocurrency market. This potentially could be used in real world trading and analysis of market data, as the current sentiment of the market is one of the biggest factors surrounding its price. We scraped tweets from the last month on twitter, cleaned the data, and vectorized each word in the set. Then we ran a K-Means model over the scraped data to cluster them into groups based on general sentiment. The groupings might not actually be around just positive and negative bitcoin tweets, but could be around other topics of discussion. Those clusters were all labeled as neutral. Then, we could count the number of positive/neutral/negative tweets to get sentiment for a tweet. Doing this to many current tweets gave us a general idea for the current sentiment in the market.

Background:

A few different projects have done similar things to us in terms of the strategy and goal of the projects.

The first project we looked at had done sentiment analysis of tweets in general. They used pre-labeled data scraped from twitter, converted and cleaned the words, and then used logistic regression, naive bayes, and an SVM model with TF-IDF matrices. From there, they attempted to correctly identify a single tweet’s sentiment. We used this project's methods of preprocessing the data as it had some good ways of doing so. However, we are working with unlabeled data so their methods for success are different from ours.

The next project we looked at did sentiment analysis on stocks from a financial news website based on the headers of the articles. Their way of capturing data we considered (as HTML data from websites rather than tweets) however, we did not end up doing that. They also did not have any preprocessing steps which likely was why their accuracy was so poor. We did not use this as a reference.

Finally, we looked at another supervised learning problem of categorizing IMDB movie reviews. The unique parts of this project were the way they tuned hyperparameters and preprocessing steps, which we will use in unison with the first group.

Data:

Data was collected from the last 30 days in tweets with the #Bitcoin on the tweet, and the word ‘Bitcoin’ in the tweet’s text as well. That way we knew with high certainty that the tweet was about bitcoin. We then dropped duplicate tweets (from the same user in that time period) and took tweets with at least 10 likes, and that were in english. This left us with 25k tweets. We then removed any non alphabetic character, lemmatized the tweets, and removed stop words from the NLTK set. We then saved the first 2000 tweets for testing (most recent tweets).

Performance:

Tools used -

Gensim’s word2vec was used to vectorize the words in the set after processing. Then K-Means was used to cluster the tweets, ideally based on sentiment. Initially with only 3 groups, we didn’t have a positive or negative grouping of words. However, after adding 2 more groups, we ended up having a positive, negative, and a couple of other groups (selling NFTs, talking about the tech, and spammers) which we identified as neutral words.

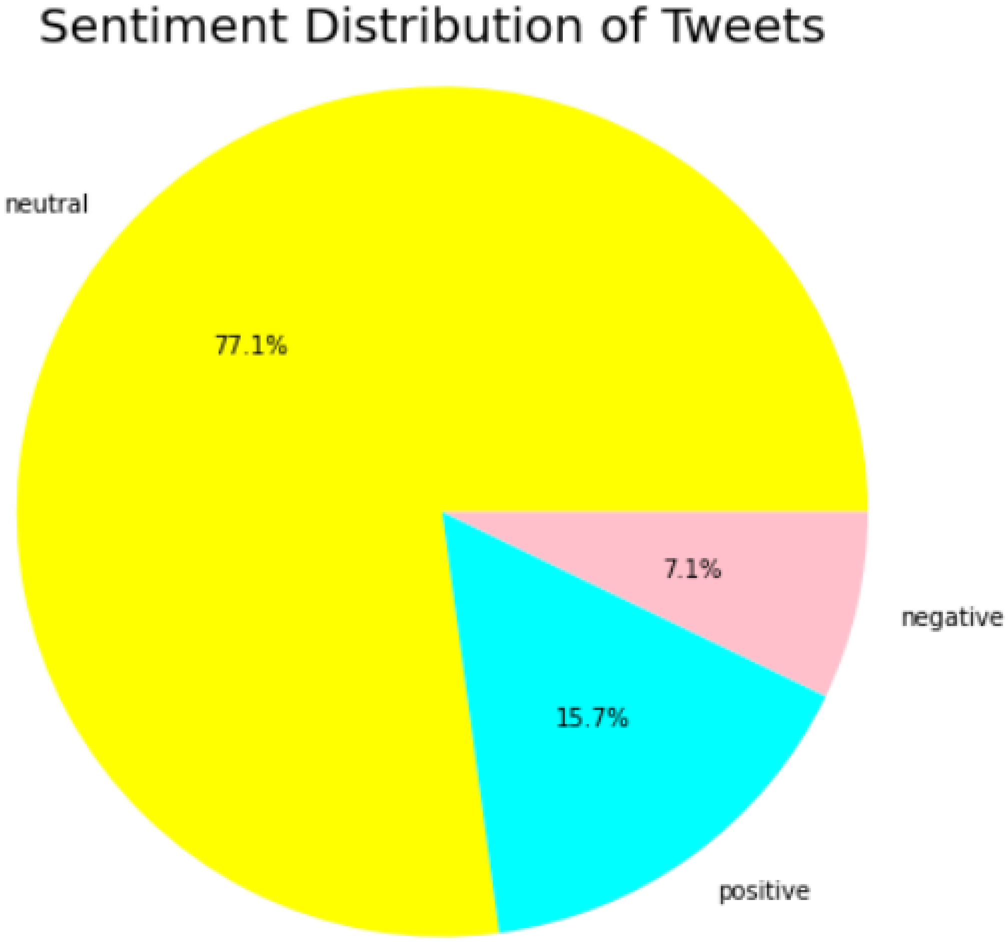
Tweaking system -

Again, initially we had 3 groups, but of different kinds of discussion. It seems that almost all the tweets are pretty neutral in nature and only about 15% are some form of sentiment. We also had to train the K-Means model over 80 epochs which was quite a lot but that led to some good overall feel of the groups. Finally, we manually set some words like ‘bull’ ‘bullish’ to positive, and others to negative to further solidify the sentiment. This step actually moved around about 3% of the tweets so it might be important to continue this.

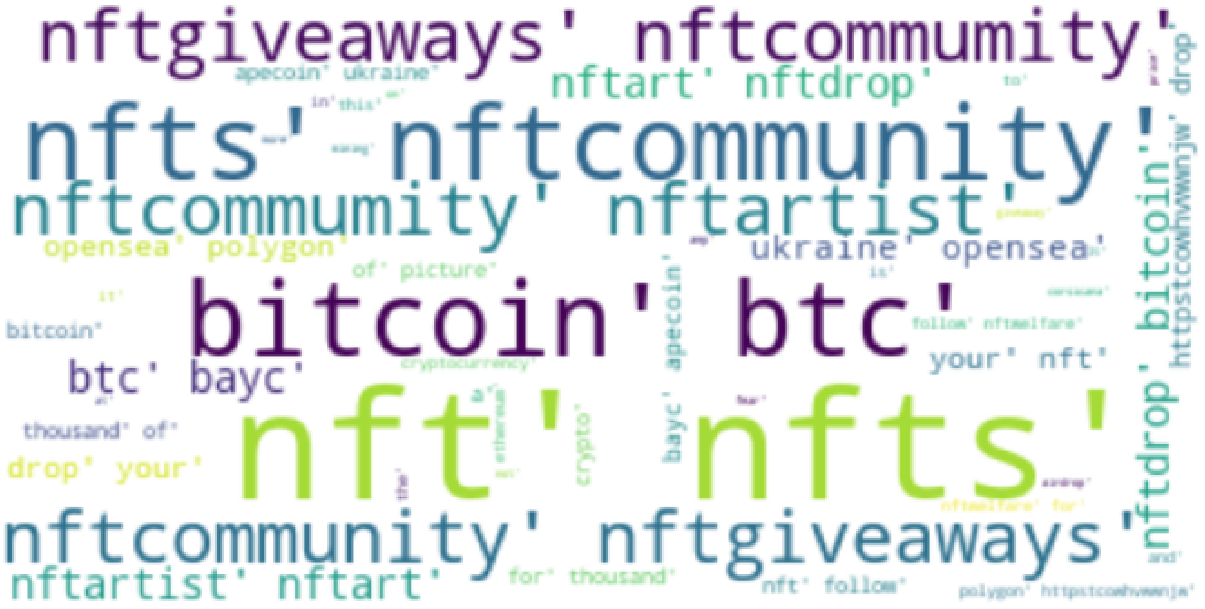
Results -

Because we are doing general sentiment classification, on unlabeled data, there’s no good statistical way to measure this. However, we found 2 solid groups that were almost all negative or positive clusters of words. From this, we took the test set and labeled each tweet based on a small scoring algorithm (it counts positive, negative, and neutral words and divides it by the total number of words getting a score between -1 and 1), and from there we could classify each tweet as positive, negative, or neutral. We adjusted that said algorithm by visually inspecting the tweets and when it had a good general idea of the tweets, we left it. Now we can get good clusters of sentiments. It should be noted, most of the positive tweets were about

NFTs, but we actually didn't see a large occurrence of NFTs in the positive words set which is interesting.



Popular words in positive grouping:



Discussion:

Next steps -

We labeled the words' sentiment to a numerical value (-1, 0, 1) which could be changed as some words might be more positive than others. Also, the words 'not' really might mess something up, if someone said 'and for that reason I am not bullish' the model would likely identify this as positive. To fix this we could give 'not' a good weight, or just use another method other than word2vec. Something that takes into account the words around words.

A way to test the success of the model might also be grabbing data during times of turmoil or bullish markets and seeing if the sentiment of that time matches correctly. If bitcoin is going up like a storm, we would expect it to have positive sentiment. This is a good way of testing the model.

Lessons learned -

We should outline some way to test the model's success rather than visual inspection if this were to be used in the real world. More consideration should also be given to words like 'not' and 'no' but that didn't seem to affect this model as much. Finally, more work should be done on data collection processes as the tool 'twint' which was used to gather tweets worked well, but if it would be used in mass, we would want a better way to collect tweets faster and over a larger span of time.