CS 4900 Final Project Report

Brandon Kimberly
https://github.com/Brandon-Kimberly/r-cryptocurrency_analysis

Abstract

In this project I created and trained a probabilistic classifier to determine the positivity of submission titles on the Reddit forum Reddit.com/r/Cryptocurrency. I utilized the official Reddit API as well as a third-party modification of said API to collect training data for the model and data to run through the model after being trained. After the data was collected, I used Python's "Natural Language Toolkit" module to clean the dataset before inputting it into the model. Multiple graphs were created, and will be included in this paper, that attempt to understand if there is a correlation between the positivity of post titles in this forum and the price of cryptocurrencies.

Background and Motivation

The Reddit forum /r/cryptocurrency is a website where users discuss their experiences with cryptocurrency, the state of the space, future developments in the technology, etc. There is a wide range of discussion that takes place here some of which is informative and helpful, some of which is merely humor.

By analyzing this forum's submission titles I hoped to see a relationship between the sentiment of post titles on the subreddit and the future price of Bitcoin (as well as other cryptocurrencies but most follow Bitcoin very well). The reason why I believe this sort of relationship *could* exist is because the members of this subreddit are likely the "early adopters" of the technology, and their opinions and voices could potentially influence the overall population and increase the price of cryptocurrencies. This sort of correlation, if established, could even be used as a *factor* in someone's decision to invest in the cryptocurrency markets.

Methods

This project can be delineated into four main parts: Data gathering, model training, model validation, and experimentation. I will go through each of these sections and describe the technical solution and any tools used.

Data Gathering

To get training data for the model I made use of the "Python Reddit API Wrapper" (PRAW). Using this module, I requested the titles of the most recent 1000 submissions on /r/cryptocurrency and then sequentially printed each title and asked the user (me) if the title is "positive" or "negative" in regards to investing in cryptocurrency. Gathering data for training is tedious but necessary to do as accurately as possible to ensure a high-accuracy model.

A few notes about training data classification:

1. Only 1000 submissions were used to train (split 70/30% into training/testing) due to limitations with PRAW and lack of human time to manually classify these titles.
2. Some submissions on the forum are merely questions or cannot be said to be either positive or negative. In these cases, they were marked as positive and only posts that are directly critical of cryptocurrency, relate a negative personal story, or claim/question if the price of cryptocurrency will go down are considered negative.

Model Training

Before we can send the data to the model for training, we need to clean it first. Many words and symbols in language are not impactful to the tone of what is being expressed. Additionally, many words are very similar but are perhaps conjugated differently. Leaving these words in the dataset will significantly increase training times and likely will not influence the accuracy of the model.

Therefore, we remove these words through the process of tokenization followed by lemmatization and removing noise with the help of NLTK. Tokenization simply splits the title up into its constituent words, lemmatization normalizes words (so words like “run,” “runs,” and “running” are all converted to just “run”) and removing the noise simply consists of deleting punctuation and words that are not impactful to tone such as “a” and “the.” After performing these steps the data is ready to be processed.

To perform the natural language processing, I utilized Python’s NLTK again. Specifically, within this module I used a naïve Bayes classifier object. This is a simple probabilistic classifier that attempts to predict the probability of an input title being positive or negative. This classifier specifically assumes that each word in the submission title is independent of the other words on the tone of the overall title. Of course, this is a limitation that prevents us from being able to properly analyze some portions of the vastness of language but was found to be similarly accurate to stronger classifiers while requiring less training data and training time and was therefore chosen.

This model is a simple form of natural language processing. It essentially attaches a probability of each word that it has encountered in the training dataset as being positive or negative. Then, on each input title it simply takes the conjunction of all of these probabilities (then multiplies it by an “evidence” constant that can be ignored but more can be read about it here) and assigns a probability of it being positive or negative. Whichever probability is higher is the output of the model on this given input title.

Model Validation

To validate that the model is not simply tuned to the training dataset, about 300 titles were excluded from the training set and used to calculate the accuracy of the model. After training on 700 titles the model attains an accuracy of around 75-80% most of the time. According to Lexalytics, an accuracy of around 80% should be the baseline to shoot for. So, it seems that the model is likely trained sufficiently enough to become reasonably accurate. Additionally, some submission titles were manually inspected and appeared to line up with my personal opinions most of the time as anecdotal "evidence" to the model's accuracy.

Experimentation

We are interested in more than just whether the model accurately classifies a statement as positive or negative. We set out to see if there was a relationship between the sentiment of post titles on this subreddit and the price of cryptocurrency. To test this, I gathered a representative sample (400 posts per day) over the last year using Pushshift API and plotted the sentiment scores week-by-week using Matplotlib. Pushshift was used instead of PRAW to simplify getting 150,000 titles (as PRAW attempts to limit you to 1000 requests) and because of better integration with timestamp searching. The results will be discussed in the Experimental Results section.

Contributions

For this project I adapted two tutorials that I read to learn about the various tools I used such as NLTK and PRAW. The first tutorial is found here and describes how to use NLTK to clean a dataset and how to pass that data to the provided classifier (which involves transforming the training data type from a list of strings into a tuple containing a dictionary mapping words to Booleans and the word "positive" or "negative.") The second tutorial is located here and details the process of using PRAW to scrape Reddit post data. Each of the tutorials provided a good foundational knowledge for the project but required extensive modification to adapt them to my particular use case.

The third-party tools used in this project include the NLTK, PRAW, and the Pushshift API. The first of which provides the tools used in cleaning the data and provides the naïve Bayes classifier object that comprises the model and does the natural language processing. The last two tools both provide APIs to obtain Reddit submission data.

Much of what I individually contributed to this project was the choice of data and natural language processor, as well as the integration of various simple pieces of code. I chose to use a naïve Bayes classifier due to its excelling in training with small amounts of data and fast training time. Choosing which data to use for training and experimentation was trickier than I anticipated. First off, I needed to manually classify 1000 post titles to provide training data to the model. Then, I had to decide how to obtain a representative sample of submissions without using too many API calls (which would take MUCH longer) and without skewing the dataset. I decided to gather 400 posts per day and then group them by week to reduce clutter in graphing. Additionally, I graphed a lot of interesting data to understand if a relationship exists between the posts on this subreddit and the price of cryptocurrency.

Experimental Results

The following is a graph plotting the sentiment score of the subreddit averaged by week over the last year (note bigger images can be found on the GitHub repository):

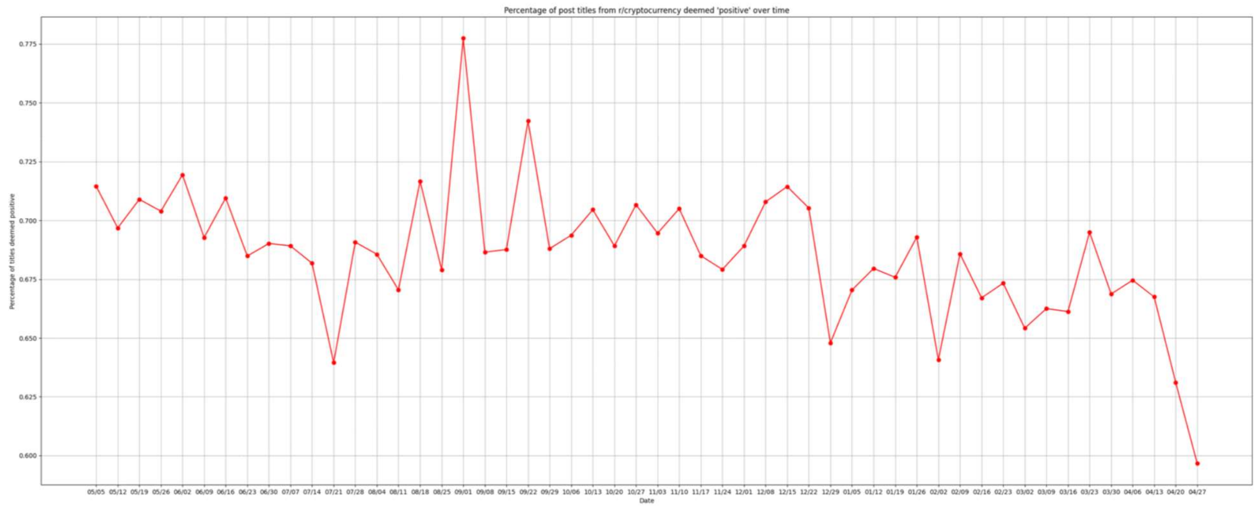

Figure 1 Sentiment Score Over the Last Year

From this graph we can see that the subreddit's sentiment is generally quite high. This is not surprising as a forum dedicated to discussing cryptocurrency likely has a favorable view of the technology. However, we are interested in seeing whether this correlates with the price of cryptocurrency. An easy way to check this is to overlay the graph of the price of Bitcoin (as most cryptocurrencies simply move in price with Bitcoin) over Figure 1. However, we must be careful to look at the delta change in price from week-to-week rather than the absolute price of Bitcoin as sentiment should stabilize around a baseline even as the price rises or lowers. This gives us Figure 2, the percent change of Bitcoin price week-over-week graphed with the sentiment score:

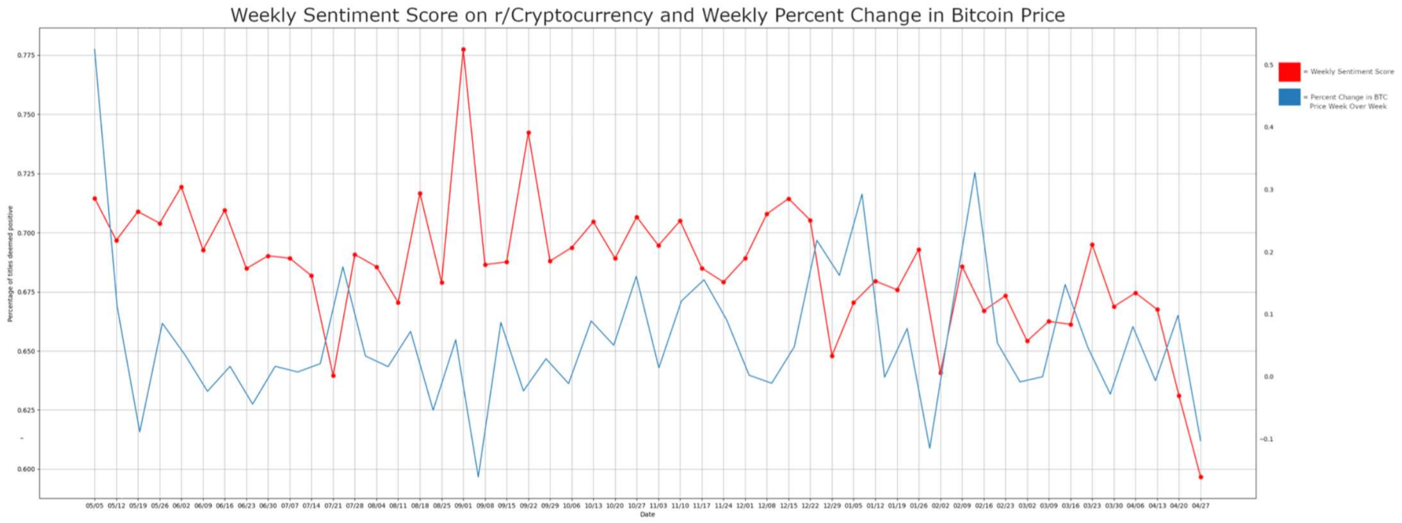

Figure 2 Sentiment Score Week by Week and Price of Bitcoin Percent Change Week-over-Week

Glancing at this seems to suggest that there is some correlation between the price of Bitcoin and the sentiment of this subreddit, but it does not seem like the subreddit predicts price changes ahead of time! It seems that the price of Bitcoin likely dominates the sentiment of the subreddit which would be expected results. Here is a graph of the difference between the above two lines (thereby avoiding the confusion of two y-axes):

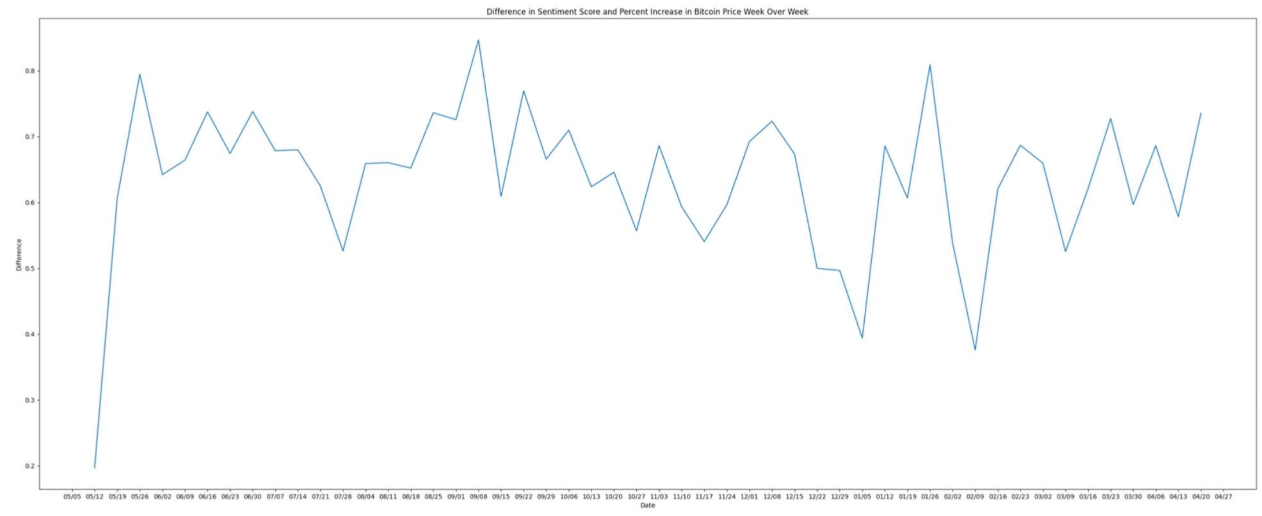

Figure 3 Difference in Sentiment and Bitcoin Percent Increase

The higher the values in this graph are the more the subreddit overestimates the current market's feelings toward Bitcoin. The more horizontal this line is the more closely correlated the sentiment of this subreddit and the price of Bitcoin are.

Conclusions

We can see from figures 2 and 3 above that it is unlikely this subreddit provides an accurate prediction for future market movement. Weeks that have high sentiment typically follow an increase in price and low sentiment is usually followed by a decrease in price. However, there does not seem to be an obvious positive relationship between the sentiment and the following weeks' price movement. This would suggest that the model is not particularly useful in providing future investing advice but simply reacts to the market's movements.

However, not everything was unsuccessful. The model appears to have an accuracy of around 80% on the validation dataset which is respectable. Using this classifier, we can semi-accurately track the mood of this subreddit, which, even if not correlated to the price of Bitcoin might still be useful in other ways. For example, it could be useful to the moderators of the subreddit to determine if changes in the rules or submission guidelines should take place.

If I had more time to work on this project, I would have provided more training data. As mentioned earlier, one of the advantages of the classifier that was used is that is can still perform well with a relatively low amount of training data, but I could still see that it was influenced too much by recent events. One of the most negative word's according to the model is 'turkey.' This likely just has to do with recent events concerning cryptocurrency and the Turkish government but is interpreted incorrectly by the model due to lack of data. Additionally, I would have grabbed more titles to plot the sentiment score over the year. Four hundred posts per day is likely sufficient but breaking it down to an even further level of temporal granularity may provide a more representative sample (though would take substantially longer to obtain).

The negative results obtained in this project could be due to a variety of factors. The foremost of which, in my opinion, is ‘Moons.’ These are a cryptocurrency that is rewarded to users for obtaining upvotes on submissions and comments on the r/cryptocurrency subreddit. Moons can be converted into any other cryptocurrency or even fiat currency so there is a monetary incentive to post things that will receive upvotes regardless of factuality or relativity to the subreddit. Additionally, some posts are simply humorous or only questions and likely have nothing to do with the forum’s actual sentiment on the cryptocurrency financial markets.

References

- NLTK tutorial
- PRAW tutorial
- PRAW API
- Pushshift API
- Lexalytics accuracy claim

Appendix

The GitHub repository can be found here. It contains a readme with useful information on how to run the program, pre-fetched example data, and the graphs found in this document.