Cryptocurrency Price Prediction Based on Twitter Sentiment Analysis

Bohdan Vakhitov, supervised by Yosi Shibberu

February 2021

Abstract

The goal of this project is to predict cryptocurrency prices based on the sentiment analysis of relevant tweets. I use VADER [1] to compute the sentiment score for each tweet, and then I average the score of all tweets within a certain time frame. I then represent these average scores as a time series and use an LSTM to predict price change direction and the magnitude of such change. To study the linear relationship between sentiment scores and bitcoin prices, I use Logistic Regression with bootstrapping to establish the significance of coefficients that correspond to each feature in the model. No significant relationship between sentiment scores and bitcoin prices was found in this study. To confirm whether the connection does not exist between the sentiment scores produced by VADER and the cryptocurrency prices, more data needs to be collected, and more sophisticated models need to be investigated.

1. Introduction

The market capitalization of all cryptocurrencies has surpassed one trillion dollars as of January 2021. Bitcoin (ticker symbol BTC) accounts for 69% of that market cap. Cryptocurrency prices are highly volatile because the amount of time needed for cryptocurrencies can be traded within minutes from a laptop anywhere with access to the internet. Since people invest in cryptocurrencies much like they do with stocks, the decision to reallocate funds (which affects the demand and, therefore, price) can be caused by social news, economical changes, or political legislation.

Due to Twitter's microblogging format, people can share opinions on Twitter very quickly - an average of 6000 tweets were posted every second in 2020. People can update their social feed as soon as something happens and check other people's reactions before making any further trading decisions. Therefore, I speculate that by analyzing the positive and negative sentiment of tweets, it should be possible to predict future changes in cryptocurrency prices.

To test this hypothesis, I compute the average sentiment score of all tweets posted within a specific time frame (in this study, 1-minute and 15-minute time frames were used) to assess the mood, panic, and excitement levels of all Twitter users within that time frame. Then, I pass the average of the computed scores into a machine learning model to predict future changes in cryptocurrency prices.

2. Prior Work

Hutto and Gilbert [1] developed and studied the power of a sentiment model they named VADER. Their research indicated a high accuracy rate for VADER when used for predicting the sentiment of tweets. The idea of studying the connection between social media sentiment and cryptocurrency prices was studied by Mehta, Kolase, Sathe, and Dwahale [2], who extracted all tweets of users with over 50,000 followers and analyzed this data against cryptocurrency prices. They work with data on a time scale of days and did not investigate more granular prediction accuracies. Their research mentions the success of VADER for measuring sentiment. However, their research does not contain enough information about the training and validating process to replicate their work so it was not used as a baseline in this paper.

3. Data and Preprocessing

The dataset for this study contains two types of data: tweets, collected through the Twitter API within one week (January 13, 2021, to January 19, 2021) with the help of the Twython library [3], and prices for bitcoin for the same time period were web scraped from the CryptoCompare API. The data from Twitter was then analyzed to determine the sentiment (positive neutral or negative) and then matched with the corresponding cryptocurrency price data.

2.1 Twitter data collection

Users with a Twitter developer account have access to a RESTful API that allows downloading tweets in various ways. As of January 2021, the upper limit on the number of times the developer can access the API is 450 queries within 15 minutes. Each query can return multiple tweets. The Twython library allows users to download over 1,000,000 unique tweets within one week. The API provides the unique ID of each tweet, text of the tweet, user name and followers number for the author of the tweet, number of likes, number of retweets, and the time when the tweet was created. For this study, only tweet text and the creation time were used. To prepare each tweet for further analysis, I removed URL links, hashtag signs, and focused on the text aspects due to higher amount of sentiment-related information there. I kept the capitalization and punctuation since some people use them to express emotions, which is also relevant to sentiment analysis.

2.2 Preprocessing Twitter Data with VADER

VADER has shown high accuracy [1] for predicting sentiment scores on tweets in previous works. For this reason, I decided to use VADER for sentiment analysis instead of creating a new model for that task. VADER provides the following output for the input tweet text: positive sentiment, negative sentiment, neutral sentiment, and compound sentiment scores. Positive, Negative, and neutral sentiment scores are scaled between 0 and 1 with 3 decimal digits of precision, and the sum of these three scores is equal to 1. The compound score is computed based on the positive and negative sentiment scores and is scaled to be between -1 and 1. I used all four scores for training a regressor and classifier.

2.3 Bitcoin price collection

The Cryptocompare API provides data for all major cryptocurrencies with open, close, high, and low information for each minute up to 1000 minutes. Such a level of detail provides an opportunity to get high-granularity data for training, since it provides a sliding interval of variable length matched with each cryptocurrency price entry, instead of measuring the data only at 15, 30, 45, 60, or some other pre-set marks on the clock.

2.4 Preparing data for time series analysis

To analyze tweets' sentiment score as a time series, I group the data within predefined timeframes. I used two types of sliding windows in this study: 1-minute, and 15-minute windows. For a window of N minutes long and a timestamp X, I define "tweet buckets" as all tweets that were posted within timeframe [X-N+1, X] for the first bucket, [X-2N+1, X-N] for the second bucket, and so on. For example, for a time stamp 17:35 and a window 15 minutes long, the first bucket will include tweets posted between 17:21 and 17:35, the second bucket will include tweets posted between 17:06 and 17:20, etc. Within each bucket, I find the total number of tweets in that bucket, as well as the average sentiment score. Last, I add the opening price of bitcoin at timestamp X. Each bucket is used for both training and prediction. Figure 1 below shows an example of the dataset with five 15-minute long buckets used in this study:

	open	bucket_0_count	bucket_1_count	bucket_2_count	bucket_3_count	bucket_4_count	bucket_5_count	positive_0_average	negative_0_average
0	36481.08	1542	1623	1300	1178	1981	1089	0.113890	0.024214
1	36408.29	1443	1689	1263	1102	2026	1358	0.113247	0.021491
2	36209.77	1376	1700	1370	1118	1740	1418	0.120269	0.023459
3	36220.19	1344	1657	1357	1154	1584	1633	0.118855	0.022553
4	36156.85	1417	1681	1462	1140	1542	1784	0.121162	0.021271

Figure 1: First few rows and columns of a dataset from the study. Open refers to the opening price of bitcoin. The bucket_i_count columns, i=1 to 5 refer to the total number of tweets collected in bucket i, positive_0_average and negative_0_averge are the average sentiment scores of all tweets recorded in bucket i

4. Model

Several different models were used to test the hypothesis proposed in this paper: Bidirectional LSTM was used to investigate non-linear dependencies, and Logistic Regression was used to investigate linear dependencies. Logistic Regression was used in combination with bootstrapping for 10,000 runs, to establish confidence intervals for the regression coefficients for each of the sentiment scores generated by VADER.

4.1 Bidirectional LSTM-Model

Long Short-Term Memory networks have been shown to be useful for modeling nonlinear dependencies in time series [2]. LSTMs have the power to model long-range dependencies in time series data. The difference between a Bidirectional LSTM and a regular LSTM is the amount of information that each node in LSTM has access to during training: A regular LSTM only learns using information from the past, while Bidirectional LSTM learns from the entire data series. For that reason, Bidirectional LSTM can learn patterns faster.

The model has two Dropout layers that regularize the model to prevent the model from simply memorizing the training set. The LSTM model in Figure 2 was used in this paper after a hyperparameter search with Keras Tuner was conducted:

```
Model: "sequential_2"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm_4 (LSTM)                (None, 1, 100)            184400
_________________________________________________________________
dropout_4 (Dropout)          (None, 1, 100)            0
_________________________________________________________________
lstm_5 (LSTM)                (None, 1, 25)             12600
_________________________________________________________________
dropout_5 (Dropout)          (None, 1, 25)             0
_________________________________________________________________
flatten_2 (Flatten)          (None, 25)                0
_________________________________________________________________
dense_6 (Dense)              (None, 100)               2600
_________________________________________________________________
dense_7 (Dense)              (None, 100)               10100
_________________________________________________________________
dense_8 (Dense)              (None, 1)                 101
=================================================================
Total params: 209,801
Trainable params: 209,801
Non-trainable params: 0
```

Figure 2: Bidirectional LSTM Model Summary

4.2 Logistic Regression

Part of the study focused on discovering indirect connections in the data, such as price change direction and price change magnitude (e.g. whether the change is greater than 30 dollars in 1 minute or less). To test the linear connection between sentiment scores and bitcoin prices, Logistic Regression was used. To establish confidence intervals and make assumptions the data was bootstrapped 10,000 times.

5. Results

Results of the model testing are presented in Table 1.

Blind prediction accuracy:

	1-minute buckets for tweets	15-minute buckets for tweets
LSTM	Price change direction: - 50% accuracy Price change magnitude: - 50% accuracy	Price change direction: - 50% accuracy Price change magnitude: - 50% accuracy
Logistic Regression	Price change direction: - 50% accuracy Price change magnitude: - 50% accuracy	Price change direction: - 50% accuracy Price change magnitude: - 50% accuracy

Table 1: Prediction accuracy

As you can see from the table, none of the experiments performed better than the baseline model, i.e. random guessing. The observation suggests that patterns useful for prediction were not recognized in the data and random guessing gives the same accuracy as the models tested in this study.

5.1 Learning curves for Bidirectional LSTM

Figure 3 shows the learning curves of the Bidirectional LSTM model that used Early Stopping with 10 iterations tolerance. The models in the top row were trained to predict the price change direction (increasing/decreasing) with 1-minute and 15-minute buckets for data, the models in the bottom row were trained to predict price change magnitude (greater/less than 30) with 1-minute and 15-minute buckets for data:

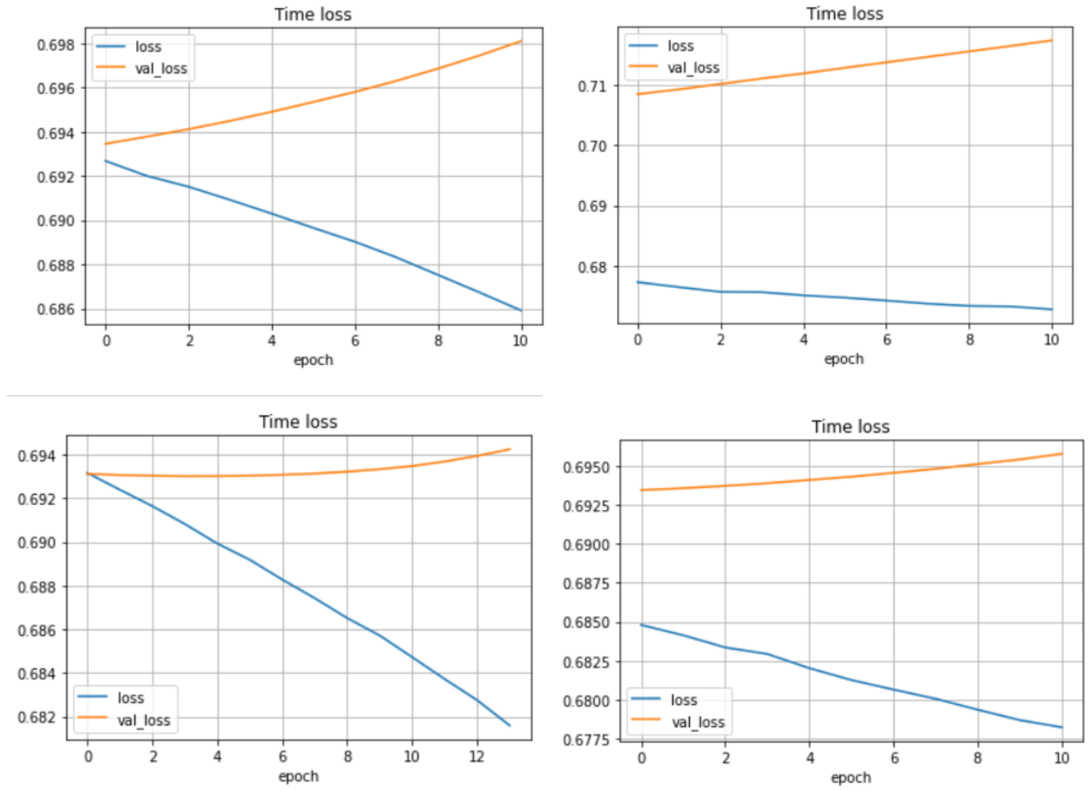

Figure 3: Learning curves of Bidirectional LSTM

Each of the learning curves indicates potential overfitting since the training curve decreases and the validation curve increases over time. To demonstrate that the model overfits, Early Stopping was removed and the number of epochs was set to 150 for each mode. Figure 4 shows the learning curves of the LSTM model that was trained to predict price change magnitude with 15-minute buckets with Early Stopping removed.

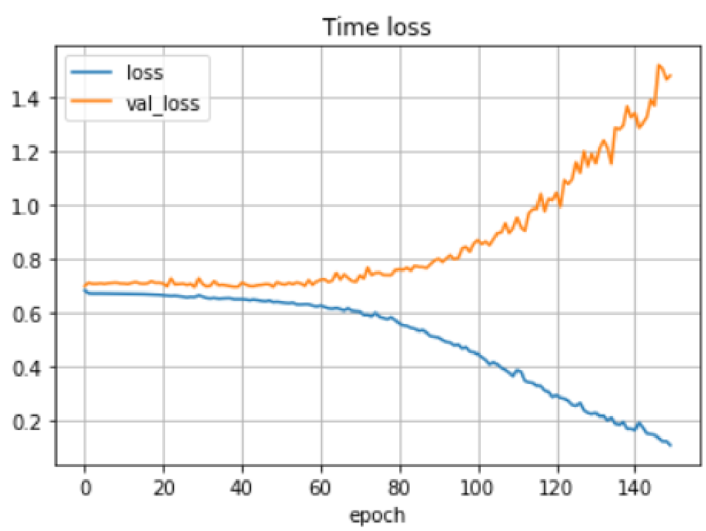

Figure 4: Learning curves of Bidirectional LSTM without Early Stopping

The training loss gets close to zero, with validation loss going up the entire time. This indicates that the model is overfitting. Curves that indicate the same behavior were obtained when Early Stopping was removed on other models.

5.2 Confidence Intervals for bootstrapped Logistic Regression

Figure 5 shows the 99% confidence intervals for the VADER score logistic regression coefficients when bootstrapped 10,000 times (sorted in the increasing order). 1-minute buckets were used in this run. In the top row, the price change direction (increasing/decreasing) was used as the prediction target, and in the bottom row, the price change magnitude (greater/less than $30 within the last time frame) was used as the target for predictions. The tables on the left correspond to 1-minute bucket size, and the tables on the right correspond to 15-minutes To interpret the table, one can say that “bucket_25_count” indicates the total number of tweets posted 25 minutes before the observed price, and the importance of the corresponding coefficient lies between -1.86 and 0.008 with 99% confidence.

	0.005	0.995
compound_80_average	-0.778309	0.093249
positive_36_average	-0.750141	-0.011116
compound_35_average	-0.722509	0.115138
compound_78_average	-0.721407	0.157566
compound_70_average	-0.720490	0.138444
...	...	...
negative_81_average	-0.060838	0.443934
negative_4_average	-0.053272	0.452203
negative_30_average	-0.041823	0.450662
negative_61_average	-0.032146	0.471648
positive_35_average	-0.023356	0.787044

	0.005	0.995
bucket_25_count	-1.841946	0.001674
bucket_80_count	-1.800812	0.099708
bucket_41_count	-1.797478	0.117944
bucket_87_count	-1.767069	0.205778
compound_77_average	-1.745952	0.267984
...	...	...
positive_16_average	-0.054614	1.730807
positive_54_average	-0.040108	1.813799
positive_35_average	-0.043267	1.780396
negative_61_average	-0.036721	1.781425
negative_30_average	-0.018373	1.691710

	0.005	0.995
positive_36_average	-0.789917	0.022868
compound_80_average	-0.768651	0.124854
compound_35_average	-0.761147	0.158719
compound_78_average	-0.752721	0.148315
compound_26_average	-0.717328	0.204005
...	...	...
negative_61_average	-0.050330	0.488579
negative_4_average	-0.041727	0.436636
negative_30_average	-0.035114	0.445964
bucket_58_count	-0.032454	0.666822
positive_35_average	-0.027450	0.796845

	0.005	0.995
compound_80_average	-0.809306	0.093465
compound_70_average	-0.773850	0.138346
compound_78_average	-0.756516	0.110249
positive_36_average	-0.733888	0.016947
compound_26_average	-0.712214	0.208666
...	...	...
negative_36_average	-0.041945	0.398369
negative_61_average	-0.038267	0.455525
negative_4_average	-0.036810	0.434202
negative_30_average	-0.017283	0.446317
positive_35_average	-0.012596	0.757318

Figure 5: Confidence intervals for Logistic Regression

5.2 Discussion

The prediction accuracies and bootstrapping indicate that there is either not enough data to learn any patterns, or that the amount of noise in data is vastly greater than any signal that is present in the data.

In an attempt to test the second assumption, I used the Savitsky-Golay filter, as it is known [4] to be used to reduce the noise in digital signal processing. However, there was no noticeable change in the learning curve, nor the prediction accuracy.

All of the confidence intervals observed contain 0 in the 99% confidence level. This means that no statistically significant linear relationship between the VADER sentiment scores and the bitcoin opening prices was found. That stays true for both 1-minute and 15-minute time frames, as well as both price change direction and magnitude.

6. Conclusion

I did not find any significant relationship between sentiment scores and bitcoin prices. However, this does not mean that there is no such connection. To more reliably test this negative conclusion, more data needs to be collected, and more sophisticated models need to be investigated.

There are several directions one can follow in further research. Further research can include other pieces of data available on Twitter API into the model: number of likes per post, which would help separate popular posts from the ones that were almost unnoticed; consideration of the user popularity, which should give an idea of how many people were reached and potentially affected by the post. Another opportunity is to work with methods used to filter out the noise: Hilbert-Huang transformation and Fourier transformation, for example.

7. References

[1] *VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text*, C.J. Hutto Eric Gilbert, Available Online at http://comp.social.gatech.edu/papers/icwsm14.vader.hutto.pdf as of 2/24/2021

[2] *Price Prediction and Analysis of Financial Markets based on News, Social Feed, and Sentiment Index using Machine Learning and Market Data*, Tapan Mehta, Ganesh Kolase, Vivek Tekade, Rahul Sathe, Anand Dhawale, June 2020, International Research Journal of Engineering and Technology, Volume 07 Issue 5

[3] *Twython Documentation*, Available online at https://twython.readthedocs.io/en/latest/ as of 2/24/2021

[4] *Methodology and Application of Savitzky-Golay Moving Average Polynomial Smoother*, E. Ostertagová and O. Ostertag, Global Journal of Pure and Applied Mathematics, Global Journal of Pure and Applied Mathematics, Volume 12, Number 4, 2016