Predicting the Movement of Bitcoin Price using Tweet Sentiment Analysis

Lorenzo Guetta, Ivan Isaenko, Claudia Jurado

November 17, 2024

Abstract

The objective of this project is to predict the direction of Bitcoin price movement based on sentiment analysis of tweets. By framing this as a classification problem, we investigate how sentiment scores extracted with the help of different techniques can act as indicators for the sign of Bitcoin return on different timeframes.

Contents

1 Related Work

Various studies have explored the use of sentiment analysis to predict cryptocurrency prices, showing that investor confidence, often reflected on social media, can influence market fluctuations.

- **Kraaijeveld and De Smedt (2020)** [1] analyzed Twitter sentiment for nine major cryptocurrencies and found it had predictive power for Bitcoin, Bitcoin Cash, and Litecoin returns, as well as for EOS and TRON using a measure of optimism. Their study highlights the importance of social media data in understanding cryptocurrency market dynamics.
- **Miao (2023)** [2] used the VADER model to analyze cryptocurrency-related tweets, describing data collection, preprocessing, and applying VADER for sentiment extraction. Cross-correlation analysis showed that tweet sentiment often anticipated price changes, but the correlation was not strong enough to build reliable predictive models.
- **Polasik et al. (2015)** [3] emphasized the role of Twitter sentiment in determining Bitcoin prices, using a sentiment index derived from tweets.

In general, research suggests that sentiment influences cryptocurrency markets and provides valuable insights into market trends and investor behavior. However, the complex nature of these markets requires further research to develop robust predictive models.

2 Methodology

2.1 Data

Initially, the plan was to use dataset `Bitcoin Tweets` [1] which contained tweets from 2022 that included the hashtags #Bitcoin and #btc. However, it was found to be incomplete, which hindered time-series analysis. After exploring other alternatives, the decision was made to use the dataset `English Tweets Mentioning Bitcoin (2021-2022)`[2].
The dataset contains more than 9 million tweets which mention 'bitcoin', covering the first half of 2022. It also includes username of the person who posted the tweet and, what is more important, precise time of publishing the tweet. Given resource limitations, we selected a subset of the dataset focusing specifically on December 2022, instead of processing all 9 million tweets in the main dataset. This sampling yielded 1,331,668 entries
The information about Bitcoin prices was obtained from cryptodatadownload[3]. We have used 1-hour and 1-min data about the exchange rate of BTC-USDT on Binance[4].

2.2 Sentiment extraction

We have used three models for extracting sentiment from the tweets:

VADER (Valence Aware Dictionary and Sentiment Reasoner): VADER is a lexicon- and rule-based sentiment analysis tool that uses a predefined dictionary to assign polarity scores to text. Unlike neural network models, VADER is not data-trained but relies on heuristic rules and is particularly suited for analyzing social media content.

[1] https://www.kaggle.com/datasets/kaushiksuresh147/bitcoin-tweets
[2] https://www.kaggle.com/datasets/hiraddolatzadeh/bitcoin-tweets-2021-2022?select=bitcoin-tweets-2022.csv
[3] https://www.cryptodatadownload.com/data/binance/
[4] https://www.binance.com

kk08 CryptoBERT: This model is a BERT-based transformer specifically fine-tuned on cryptocurrency-related text data. CryptoBERT [5] is a version of the ProsusAI/FinBERT model and was trained on a large corpus of social media posts about cryptocurrencies. It classifies sentiment into positive or negative classes.

ElKulako CryptoBERT: Similar to kk08, ElKulako[6] is another BERT-based model pre-trained on a dataset of over 3.2 million cryptocurrency-related social media posts. This model categorizes the sentiment into three labels: Bearish (0), Neutral (1), and Bullish (2).

Using these three sentiment extraction methods enables us to compare their predictive performance on financial data. To normalize the data, we adjusted it based on each model's output format and how it handled sentiment scoring.

For CryptoBERT, which provide scores from 0 to 1 along with a label indicating sentiment—negative (closer to 0) or positive (closer to 1)—we handled normalization as follows: for samples labeled as negative (`label_0`), we subtracted the sentiment score from 1, effectively inverting it. For samples with a positive (`label_1`) label, we retained the original score. This approach allows to use only one variable describing sentiment extracted with this model instead of two, therefore avoiding multicollinearity.

For outputs from ElKulako, which categorized sentiment into three types (e.g., negative, neutral, and positive), we mapped these to values of -1, 0, and 1, respectively.

2.3 Data pre-processing

Before extracting sentiment from the tweets, a preliminary data preprocessing step was conducted:

- Tweets were converted to lowercase to ensure uniformity.
- Usernames, URLs, mentions, and non-alphabetic characters were removed, as they do not contribute to determining tweet sentiment.
- Punctuation and common stopwords were removed using the NLTK library[7] to reduce noise and focus on meaningful words.
- Lemmatization was applied to tokens with more than three characters using WordNetLemmatizer from NLTK[8], helping to standardize word forms without excessive loss of meaning.

We have decided to keep content of hashtags, since it could contain some helpful text for determining sentiment of tweets.
The preprocessed data was utilized specifically for the VADER tool, as research indicates that additional preprocessing generally does not enhance the performance of transformer-based models. Transformers are inherently equipped with mechanisms to manage noise in textual data effectively, making extensive preprocessing unnecessary. This capability is wellsupported by recent studies in the field [4, 5].

3 Data aggregation

In our work we considered three different time intervals: 5 minutes, 30 minutes, and 1 hour. Thus, firstly, we grouped financial data and tweets according to these intervals.

[5] https://huggingface.co/kk08/CryptoBERT
[6] https://huggingface.co/ElKulako/cryptobert
[7] https://www.nltk.org/
[8] https://www.nltk.org/api/nltk.stem.WordNetLemmatizer.html?highlight=wordnet

It's important to mention that tweets were grouped as follows: all tweets which appear between t and t+1 refer to time moment t+1. In this way, we avoid using current tweets to predict current price movement, which is an impractical task, and avoid endogeneity (two-ways relationship). During grouping sentiment scores were aggregated by mean. Also, returns were computed as the difference between close and open prices, divided by open price. The returns were labeled as either 1 or 0, depending on whether the Bitcoin return was positive or negative.

4 EDA

In the following section we provide a brief comparison of sentiment score distributions.
In the graph 1 , we can observe the distribution of the variable kk08_pos_score, which represents the positivity score extracted using the kk08 model. The distribution is clearly skewed to the right, with a significant concentration of tweets having scores close to 1. This indicates that a large proportion of tweets are highly positive.
Additionally, there is a noticeable scarcity of mid-range values (neither extreme nor neutral). This suggests that tweets tend to be evaluated as either very negative or very positive, rather than receiving moderate positivity scores.
In graph 2 shows the distribution of positivity scores extracted using the VADER tool. In contrast to the observations made with KK08, there is a high number of samples with neutral values, and relatively few tweets that fall into the extreme categories of completely negative or completely positive. This indicates a lower degree of polarization compared to the dataset shown in 1.
Lastly, in graph 3 we can observe that the positivity scores extracted using the Elkulako model are distributed across three categories: -1, 0, and 1, where -1 represents the most negative sentiment, and 1 represents the most positive sentiment. The distribution shows three distinct peaks, with the majority of tweets categorized as having a neutral score. Additionally, there is a noticeable imbalance in the number of tweets assigned a negative score compared to the other two categories.
It is worth noting the simplification of the scores, as they are not represented as a continuous distribution. This categorization into three distinct groups simplifies the analysis but may limit the granularity of insights that could be gained from a more nuanced scoring system.

5 Models

5.1 Training and Validation Approach

To train and fine-tune the models, a cross-validation strategy was employed. The data for each timeframe was divided into three distinct sets:

- 85% for training,
- 10% for validation (From the Traning data), and
- 15% for testing.

This split was designed to assess model performance on unseen data during training, ensuring a robust evaluation process. The separation into training, validation, and test sets allowed us to monitor performance throughout the training phase, minimize the risk of overfitting, and optimize hyperparameter selection for the specific problem.
As we said before our target variable is sign of Bitcoin return. As X we considered different inputs: each sentiment score (separately), each sentiment score with 3 lag values (separately) and all sentiment scores with lags (together). As an experiment, we have created

also some dummy variables reflecting the appearance of a certain word in a tweet, which, in our opinion, could help to predict price movement. Such words as "fall", "growth", "buy", "sell" etc. were considered. These variables together with sentiment score were also used as an input for our models.

5.2 Automated Hyperparameter Tuning

To streamline hyperparameter optimization, the Scikit-learn library was utilized to perform a grid search. This automated process systematically explored a predefined range of hyperparameter values (`grid_parameters`) to find the best configuration. By automating this step, we avoided manual tuning and ensured the selection of models delivering optimal performance.

5.3 Used Algorithms

We evaluated the performance of several algorithms to determine which is the best in predicting Bitcoin price movements. These include:

- **Random Forest (RF)**: An ensemble method that builds multiple decision trees and averages their predictions for a more accurate classification. It's robust to overfitting and works well with complex datasets.
- **XGBoost (XGB)**: A gradient-boosted algorithm that improves accuracy by sequentially building models to correct the errors of previous models. It's known for its speed and performance in large-scale datasets.
- **Support Vector Machine (SVM)**: A classifier that finds the optimal hyperplane to separate data into distinct classes. It's effective for high-dimensional data and complex problems.
- **AdaBoostClassifier**: An ensemble method that combines weak classifiers to form a stronger one by focusing on misclassified samples, improving performance on difficult predictions.
- **GaussianNB**: A probabilistic classifier based on Bayes' Theorem, assuming normal distribution of features. It's simple, fast, and works well with high-dimensional continuous data.
- **Logistic Regression**: A linear model for binary classification that estimates probabilities based on predictor variables. It's efficient and widely used for classification tasks.
- **KNeighborsClassifier**: A non-parametric classifier that predicts based on the majority class of the nearest neighbors. It's effective for non-linear decision boundaries and simple to implement [9].

For the implementation of these classifiers, we used the sklearn[10] library, and xgboost package for the XGBoost classifier.

5.4 Evaluation Metrics

To evaluate the performance of the models in predicting Bitcoin price movements, the following metrics were used: **accuracy** (to measure the proportion of correct predictions), **F1-score** (to balance precision and recall), and the **ROC AUC** (to measure the ability to distinguish between classes across thresholds).

[9] https://www.ibm.com/topics/
[10] https://scikit-learn.org/stable/

6 Results

We received some positive results for each time interval and each sentiment score. Thus, for 15-min data some models (RF, XGB) were able to achieve an accuracy of approximately 0.53-0.54 using separate sentiment scores on the test data with balanced predictions (ROC-AUC and F1-score are at reasonable level). As expected, for 30-min and 1-hour data the results are even better: 0.55 - 0.57 for the best-performed configurations. In average, tree models (RF, AdaBoost and XGB) and knn appeared to be the most suited for our task, which can be explained by their ability to catch non-linear relationship and low requirements to the size and dimension of data. Moreover, in general, using all sentiment score simultaneously doesn't help to achieve better predictions. The same can be claimed about using described dummy variables. Also, we conclude that models trained on VADER and Elkulako sentiment features in average perform better than on kk08, which could be explained by "positive bias" in the scores of kk08.

7 Future Work

In future research, several enhancements could improve our findings' robustness and applicability. Expanding data sources to include platforms like Reddit or news sites would provide a broader sentiment base, enhancing model accuracy by incorporating diverse perspectives.

Integrating advanced NLP techniques, such as transformer models or financial-specific embeddings, could yield more nuanced sentiment scores and boost predictive capabilities. Increasing computational resources would allow for processing longer periods, enabling more longitudinal analyses and better trend identification. Analyzing whether messages were generated by bots or genuine users could help filter manipulative or non-authentic sentiment, improving prediction quality. Developing a real-time sentiment analysis API could offer immediate insights, supporting predictive models for Bitcoin returns and providing timely predictions for traders and analysts.

References

[1] O. Kraaijeveld and J. D. Smedt, "The predictive power of public twitter sentiment for forecasting cryptocurrency prices," *Journal of International Financial Markets, Institutions and Money*, vol. 65, 3 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S104244312030072X?via%3Dihub#s0150

[2] J. Miao, "Research on tweet sentiment analysis based on vader in the field of cryptocurrency," pp. 256–264, 2023. [Online]. Available: https://www.atlantis-press.com/article/125995000.pdf

[3] I. Georgoula, D. Pournarakis, C. Bilanakos, D. N. Sotiropoulos, and G. M. Giaglis, "Using time-series and sentiment analysis to detect the determinants of bitcoin prices," 2015. [Online]. Available: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2607167

[4] M. Siino, I. Tinnirello, and M. La Cascia, "Is text preprocessing still worth the time? a comparative survey on the influence of popular preprocessing methods on transformers and traditional classifiers," *Information Systems*, vol. 121, p. 102342, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0306437923001783

[5] A. K. Uysal and S. Gunal, "The impact of preprocessing on text classification," *Information Processing Management*, vol. 50, no. 1, pp. 104–112, 2014. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0306457313000964

A Appendix

A.1 Sentiment Distribution

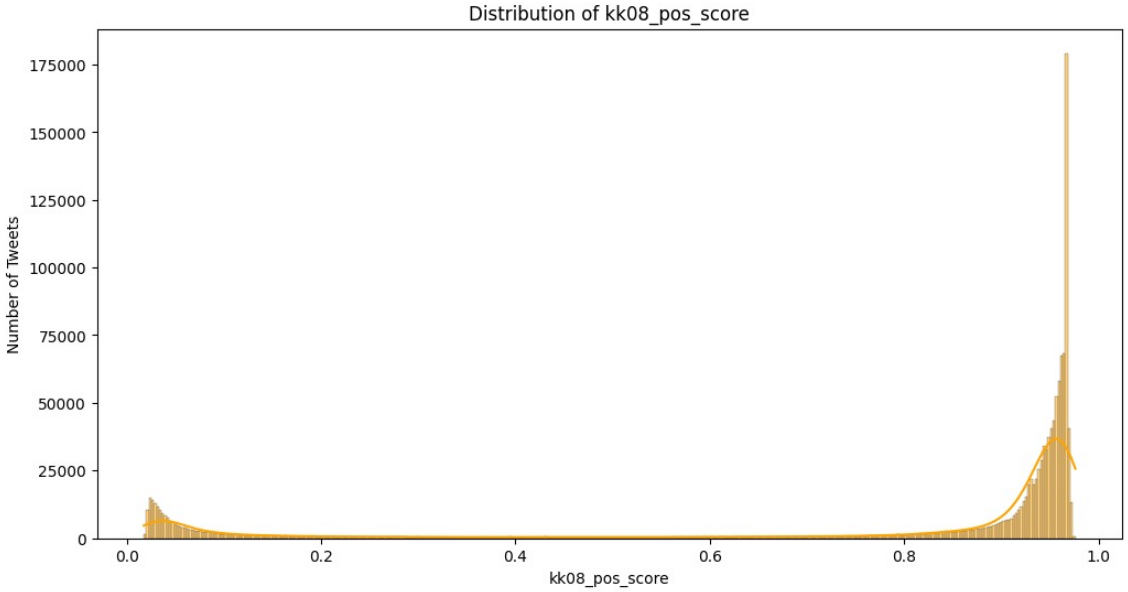

Figure 1: kk08 sentiment distribution

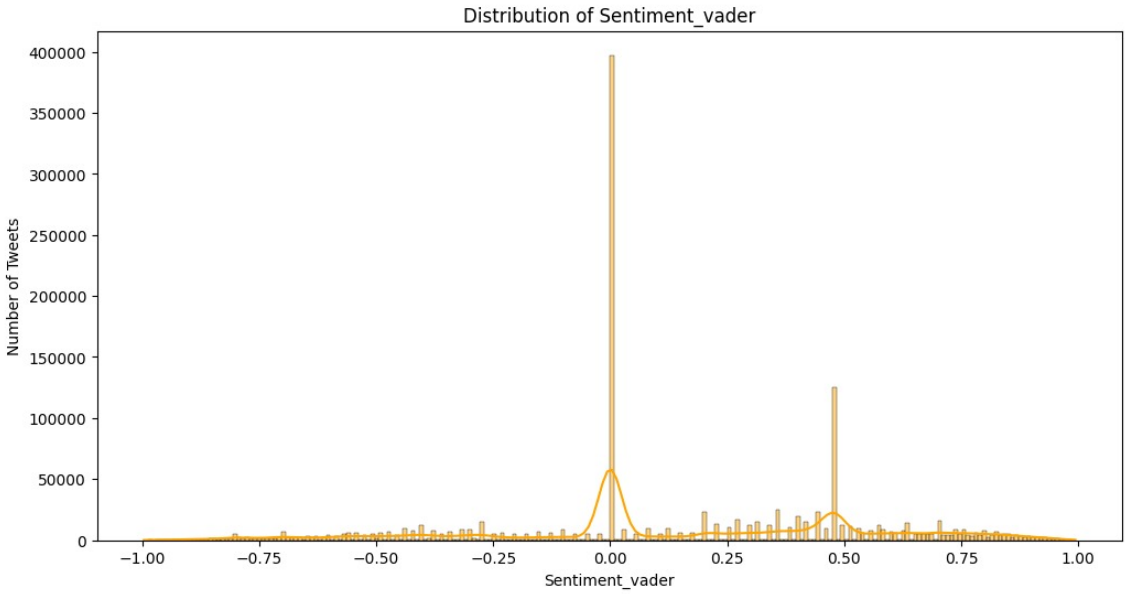

Figure 2: Vader Sentiment Distribution

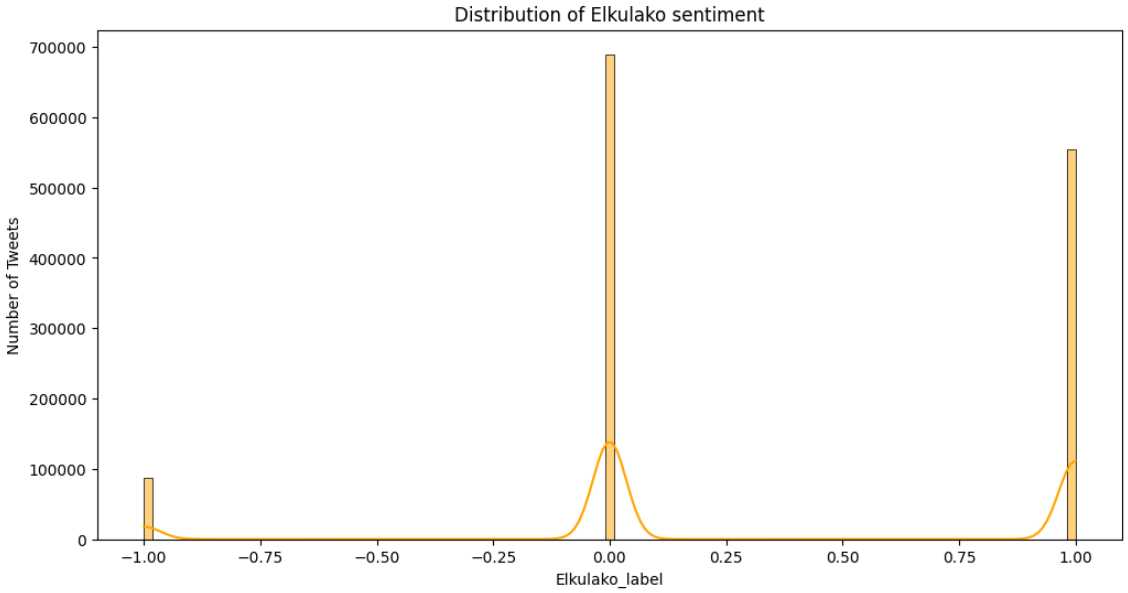

Figure 3: Elkulako sentiment distribution