Decrypting the Crypto: Social Media Sentiment Analysis and its Impact on Cryptocurrency Valuation

Amari Parris - parris.a@northeastern.edu | Parth Malik - malik.p@northeastern.edu | Yaseen Ellison - ellison.y@northeastern.edu

1. Objectives & Significance

The field of cryptocurrency valuation is a continuously evolving one, with an increasing potential for the application of predictive modeling through the integration of unique data streams. Traditionally, currency valuation methods have primarily revolved around fundamental and technical analysis. Essentially, exploring the underlying technology and governance structures, like production and legislative factors, and also considering more technical aspects like historical price and volume patterns. These traditional methods often stumble in the cryptocurrency domain due to an over-reliance on qualitative factors, the scarcity of historical information, and a tendency to oversimplify the complex nature of cryptocurrencies, among other issues [7]. Given the speculative nature and high volatility often seen in cryptocurrency markets, however, there is a call for a broader analytical framework with regards to cryptocurrencies. A promising approach is the integration of sentiment analysis, which seeks to gauge market sentiment from various data sources like social media, search hits, and news articles. The impact of social movements on cryptocurrencies - as seen in the meteoric rise of Dogecoin fueled by social media hype - underscores the potential sentiment analysis may hold in capturing the broader social dynamics at play, which traditional methods may overlook.

Given the nascent stage of cryptocurrency and the variability in cryptocurrency types, no one-size-fits-all model exists for their valuation. Some argue that traditional valuation methodologies can be applied to cryptocurrencies, given the variability in user sentiment, while others argue against this due to market volatility, especially outside of more widely known cryptocurrencies. In a study by Liu and Tyvinski (2021) titled "Risks and Returns of Cryptocurrency", a link was found between “investor attention”, defined by Google search and Twitter posts, and Cryptocurrency price.[6] Additionally, a study by Nasekin and Chen (2020) demonstrated that sentiment extracted from social media platform StockTwits, using BERT with added domain-specific tokens, significantly contributed to the predictability of cryptocurrency returns, further emphasizing the value of integrating sentiment analysis into cryptocurrency valuation models. [8] By sampling a range of Cryptocurrencies, we aim to understand the impact and linkage of 'investor/user attention' or general sentiment, on the currency of interest. Through our project, we aspire to build a unique analysis of social sentiment and price, utilizing Twitter and Reddit posts as primary sources of information about cryptocurrencies. By leveraging machine learning algorithms, we aim to couple sentiment analysis and time series analysis on an exploration of crypto valuation. Unveiling the intricate relationships between the aforementioned factors.

2 Background

2.1 Sentiment Analysis with FinBERT

Sentiment analysis employs Natural Language Processing and text analysis to identify and extract subjective information from a source. In this project, this technique was applied to social media posts to determine the public sentiment surrounding cryptocurrencies.

Our team decided to use the FinBERT model for initially labeling the sentiments as our whole dataset had no labels. FinBERT is a deep learning model that leverages the architecture of BERT (Bidirectional Encoder Representations from Transformers) to perform sentiment analysis specifically within the financial domain. Designed to capture the complex semantics of financial language, FinBERT is fine-tuned from the BERT model using a large corpus of financial texts, which includes company reports, news articles, and analytical pieces. This specialization allows FinBERT to more accurately interpret the sentiment of texts related to markets, economic outlooks, and investment opinions, making it particularly adept at understanding the nuanced discourse found in cryptocurrency-related social media posts. The following are some specifications of this model:

Pre Training on Financial Text: While BERT was pretrained on a general corpus like Wikipedia and the BookCorpus, FinBERT's pre training involved an extensive financial corpus that includes the texts from the Corporate Reports, Earnings Calls, Analyst Reports, Business News, etc. This enables FinBERT to understand the intricacies and nuances of financial language.

Financial Sentiment Analysis: The FinBERT model is specifically fine-tuned on financial sentiment analysis tasks. This fine-tuning process involves adjusting the model's parameters on labeled datasets where financial sentiments (positive, negative, neutral) are annotated, allowing FinBERT to predict the sentiment of unseen financial texts with higher accuracy.

Hyperparameters: FinBERT typically shares the same hyperparameters as the base BERT model. This includes:

- 12-layer transformers
- 768 hidden units per layer
- A feed-forward network of 3072 units
- 12 self-attention heads
- A total of 110 million parameters
- Tokenization using WordPiece, handling up to a sequence length of 512 tokens

Training Mechanism: FinBERT is trained using the masked language model (MLM) and next sentence prediction (NSP) training objectives, similar to BERT. However, the training data and objectives are fine-tuned to focus on financial contexts.

Attention Mechanism: The model uses multi-headed self-attention to weigh the influence of different words within the input text. This is particularly useful in financial texts where the sentiment may be heavily dependent on a few key terms and their context.

Output Interpretation and Confidence Measurement: The FinBERT model's output is not merely categorical but probabilistic, providing a nuanced distribution over possible sentiment labels for each piece of text. These probabilities are a direct measure of the model's confidence in its predictions. For instance:

High Confidence Prediction Example: If the model outputs probabilities such as {positive: 0.90, neutral: 0.05, negative: 0.05} for a particular text, we can infer a high degree of confidence in a positive sentiment. This strong prediction could be due to the presence of definitive positive financial indicators or language within the text.

Low Confidence Prediction Example: Conversely, output probabilities like {positive: 0.33, neutral: 0.34, negative: 0.33} suggest uncertainty, indicating that the text may be more ambiguous or contain mixed signals that the model finds difficult to interpret.

2.2 Time Series Analysis

Time series analysis has been used for many years to analyze financial markets providing a systematic approach to understanding and predicting financial data. Time series analysis involves the examination of data points ordered in time. In financial contexts, this can be daily stock prices, trading volume, or cryptocurrency prices. The goal is to analyze the past behavior of a time series to make predictions about its future values.

One of the early uses of time series analysis in finance can be traced back to the 1900s with Louis Bachelier’s thesis on "The Theory of Speculation," where he modeled stock prices as a random walk [1]. Bachelier's work laid the groundwork for significant concepts in modern financial theory including the Efficient Market Hypothesis developed by Eugene Fama in the 1960s and modern financial theory.

ARIMA (Auto Regressive Integrated Moving Average):

In the mid-20th century, applications of time series analysis took a significant leap forward with the development of the Autoregressive Integrated Moving Average (ARIMA) model by George Box and Gwilym Jenkins [2]. The ARIMA model is characterized by its ability to handle different types of time series data, including those with trends and seasonality.

Over the years, the ARIMA model has been refined for financial forecasting. Numerous studies and papers have been published, demonstrating its effectiveness in predicting stock prices, exchange rates, and other financial indicators. A notable example is a paper by Robert J. Shiller published in 1981 using an ARIMA model to analyze stock prices and dividends, contributing to the debate on market efficiency and the predictability of stock returns [3].

ARIMA is widely used for non-stationary time series but requires the data to be transformed into stationary data before applying the model. A time series is said to be stationary if its properties are not dependent upon the time at which the series is observed, meaning the mean, variance, and covariance are constant. An ARIMA model with exogenous features (ARIMAX) was selected to incorporate sentiment and supplemental market data into models for individual coins. The ARIMAX model has four components, autoregression (p), differencing (d), moving averages (q), and a parameter to include exogenous features.

Autoregression represents the relationship between an observation of interest and its own prior values referred to as lag features or lagged observations. Observations are regressed on some number of lagged observations. Moving averages represent the relationship between an observation and a residual error from a moving average model applied to lagged observations. It indicates the regression error is a linear combination of error terms. The differencing indicates that the data values have been replaced by the difference between their values and the previous values. When performed once this is known as first differencing. This process can be repeated more than once to achieve stationary data.

In essence ARIMAX models are specified with four parameters: (p,d,q, exog)

p: The number of lag observations included in the model
d: The number of times that the raw observations are differenced
q: The size of the moving average window
Exog: exogenous features

In the context of cryptocurrency price prediction, ARIMAX is a viable choice, given its suitability to handle non-stationary behavior and capability to incorporate supplemental data into predictions. The volatile nature of cryptocurrencies, combined with their sensitivity to extrinsic factors necessitates a more comprehensive approach to forecasting. Traditional financial models may not fully capture these nuances, therefore it is critical to consider incorporating additional features like sentiment scores from social media to improve forecasting accuracy.

2.3 Parallel Study with Random Forest:

The Random Forest (RF) model, known for its robust ensemble learning approach, has become prominent in machine learning due to its adaptability in handling both classification and regression tasks. Used across various sectors like e-commerce, banking, medicine, and the stock market, the model operates by creating numerous decision trees during training. In classification tasks, it outputs the class most commonly selected by the trees, whereas, in regression tasks, it computes the average prediction of the individual trees.

Developed by Leo Breiman and Adele Cutler, Random Forest is an advanced form of the bagging method, combining bagging and random feature selection to create a collection of uncorrelated decision trees. This method notably mitigates the overfitting issue typically associated with individual decision trees, making it particularly effective for complex datasets such as those encountered in financial sentiment analysis. [10]

In the realm of financial sentiment analysis, Random Forest is prized for its ability to process extensive and diverse datasets, including varied sentiment scores and financial metrics. This capability aligns perfectly with the complex nature of financial data, where resistance to overfitting is essential due to the intricate interplay of market variables. A key strength of Random Forest is its feature importance analysis, enabling the identification of the most influential factors in sentiment data that affect cryptocurrency prices, thereby enhancing our understanding of market dynamics.

In our study of the cryptocurrency market, we integrated Random Forest alongside our ARIMA model to leverage its robust data handling capabilities and its proficiency in feature importance analysis. This integration followed the Time Series Analysis results, aiming to more thoroughly examine the impact of additional features on market sentiments. By running these models concurrently, we intend to achieve a comprehensive understanding of how sentiment influences cryptocurrency markets. The diverse methodological insights offered by the Random Forest and ARIMA models are expected to significantly enhance the depth and validity of our analysis.

3. Methods

3.1 Data Collection & Preprocessing

The dataset for our analysis was derived from two primary sources: a larger corpus of 250,000 tweets by the general public (Set A with no sentiment labels and raw text) and a smaller, specialized set of 16,000 tweets from cryptocurrency influencers (Set B with sentiment labels and raw + clean text). Set A provided a broad view of public sentiment, while Set B was expected to offer insight into the perspectives of sector-specific thought leaders.

However, the existing sentiment labels in Set B on the 16000 rows were too limited and lacked the necessary detail for in-depth analysis. As a result, we chose to exclude these labels and any previously cleaned text to ensure uniform processing. The final dataset, combines the unprocessed tweets from both sets, aligning them for consistent and unbiased preprocessing. This merged set aims to reflect the true sentiment in the cryptocurrency market, without prior biases from incomplete sentiment labeling or inconsistent text cleaning.

The financial dataset consisted of 18,000 rows of daily crypto market data for 17 coins and U.S. federal rates from January 2021 to November 2023. Historical market data was sourced from the Coinbase and CoinGecko API's, with U.S. federal rates, treasury yield and interest rates, sourced from AlphaAdvantage. Historical market data; market capitalization, trading volume, and daily price metrics provided a comprehensive view of market conditions to supplement close price time series. U.S. interest rates and bond yields were used as a measure of global risk appetite of investors, as U.S. treasury securities are widely recognized as a global benchmark for risk-free assets due to the perceived stability and strength of the U.S. economy.

Preprocessing Tweets:

Our preprocessing workflow involved several tailored steps to refine the tweet data for accurate sentiment analysis:

1. Custom Stop Words and Lemmatization: We identified and preserved key terms such as "not," "no," "down," "up," "above," "below," "against," and "now" within our stop words list. These terms often carry significant sentiment weight, especially in the volatile context of financial discussions. For example, "not bullish" indicates a very different sentiment than "bullish," and our approach ensured such nuances were not lost. Concurrently, we applied lemmatization to bring words to their root form, simplifying "running" to "run," thereby standardizing our dataset for the FinBERT model.

2. Text Cleaning and Normalization: Our cleaning process involved several steps to remove extraneous elements from the text. We decoded HTML entities, expanded contractions (e.g., transforming "can't" to "cannot"), and stripped away URLs and user mentions which do not contribute to sentiment, such as removing "@User321" from the tweets. We also eliminated non-ASCII characters and unnecessary whitespace, cleaning the text to a form that retains only the substantive content for analysis.

3. Domain-Specific Dictionary Application: We implemented a crypto-specific dictionary to contextualize abbreviations and jargon. Terms like "HODL" were expanded to "hold on for dear life," a phrase emblematic of the crypto community's investment stance. This ensured that the sentiment associated with such terms was accurately captured by the sentiment analysis model.

4. Noise Reduction and Tokenization: After removing punctuation, we converted text to lowercase and tokenized it, breaking it into individual word units. For instance, the tweet "HODL! Despite FUD, $BTC will rise." was tokenized into ['hodl', 'despite', 'fud', 'btc', 'will', 'rise'].

5. Negation Handling: We paid special attention to negations, a critical aspect of sentiment analysis. By prefixing "not_" to words following "not" or "no," we preserved the sentiment directionality, such as "not_increasing" versus "increasing."

6. Robustness and Error Handling: We encapsulated our preprocessing function within a safe_preprocess wrapper to handle exceptions gracefully. Texts that caused errors were excluded to maintain data integrity. For example, tweets composed solely of user mentions which often did not contain relevant sentiment information were removed from the analysis.

The outcome of this preprocessing was a 'clean_text' column in our DataFrame, presenting the text in a format primed for the sentiment analysis model, ensuring high fidelity in sentiment interpretation.

Preprocessing Time Series Data:

Thorough data preprocessing was conducted to ensure data quality and consistency. Precautions were taken with data generated via APIs to ensure financial data was consistently sampled at a daily frequency to establish appropriateness of data for time series predictions of daily close price. Dictionaries of coin tickers and names with available social media data were constructed to search exchange APIs for product ids to retrieve accurate market data for each coin. Due to limitations of API call frequency and per call data volume, data was pulled 6 months at a time and consistently reindexed to ensure proper time series alignment for each coin.

1. Imputation: Missing data points were handled by various imputation methods including linear interpolation and time-based backward fill when appropriate. Features that appeared relatively sparse, like interest rates, were imputed using backward fill where more complex features, like market capitalization, that are a function of other financial statistics were imputed using linear interpolation.

2. Normalization: Given the varied array of features and coins in the dataset, market data was normalized using Z-score normalization to handle outliers due to potential price spikes and equate feature scales to effectively gauge feature importance across small scale features such as treasury yield and large scale features such as market cap.

3. Stationarity: Most time series models assume stationarity, which implies a constant mean, variance, and autocorrelation structure over time. Augmented Dickey Fuller (ADF) was applied to assess stationarity of features and targets. Based on ADF performance, features were, initially, manually differenced. Manually differenced features were used to determine appropriate ranges for autoregressive (p) and moving average (q) terms to reduce training time and model complexity. However, poor performance led to implementation of a more comprehensive grid search to determine appropriate level of differencing.

3.2 Exploratory Data Analysis (EDA)

Exploratory Data Analysis of the Tweet Data

Our preliminary EDA was thorough, yielding insights that would shape our preprocessing and sentiment analysis strategies. We present our findings as follows:

1. Contextual Analysis of Crypto Terms: Our dictionary, tailored to the crypto domain, included 81 terms pivotal in cryptocurrency discourse. This resource was pivotal for the subsequent stages, ensuring concepts like 'All-Time High' (ATH) and 'Fear, Uncertainty, and Doubt' (FUD) were fully understood by our model. Such expansions are vital as they preserve the sentiment and context, which our analysis confirmed by matching 64 abbreviations from our dictionary in the tweets, with only 17 remaining unmatched.

2. Frequency Analysis of Words and Phrases: Our corpus analysis revealed 'the', 'to', 'and', 'on', and 'a' as the most frequently occurring words, with significant representations of cryptocurrency symbols like

'$ETH'. The bigram analysis shed light on common pairings such as ('to', 'the') and ('$BTC', '$ETH'), indicating the conversational patterns and topics prevalent in the community.

3. Analysis of Special Characters and Non-Standard Text: Our dataset was rich in special characters, with dollar signs ($) and hashtags (#) leading the tally, underscoring the financially driven nature of the communication. Emojis like rockets (🚀) and fire (🔥) were also prevalent, reflecting the emotive and speculative elements of cryptocurrency trading.

4. Hashtag and Mention Analysis: Hashtags like #safemars and #bitcoin were among the most used, indicating trending topics and market sentiments. Mentions of influencers and platforms, such as @elonmusk and @binance, highlighted the communal and influential nature of the space. Our analysis led to the decision to exclude user mentions that didn't contribute to sentiment, ensuring the relevance of our data.

5. URL Analysis: A substantial 168,501 tweets contained URLs, which we decided to remove, recognizing that while they point to additional information, they don't inherently carry sentiment value.

6. Length and Readability Analysis: The average tweet was 27.57 words long, affirming the concise nature of Twitter communication. This brevity underscored the challenge of extracting sentiment from limited text, reinforcing the need for our meticulous preprocessing approach.

The EDA not only informed our understanding of the dataset's characteristics but also validated the preprocessing steps we had envisioned. By converting abstract numbers into meaningful patterns, we were better equipped to refine our sentiment analysis model, ensuring it was attuned to the nuances of financial discourse within the crypto community.

Exploratory Data Analysis (EDA) of Financial Data:

1. Line Plots: Cryptocurrency prices and other features along with the closing price target were plotted over time to visually identify trends, seasonality, and outliers.

Figure - Line Chart of Close Price

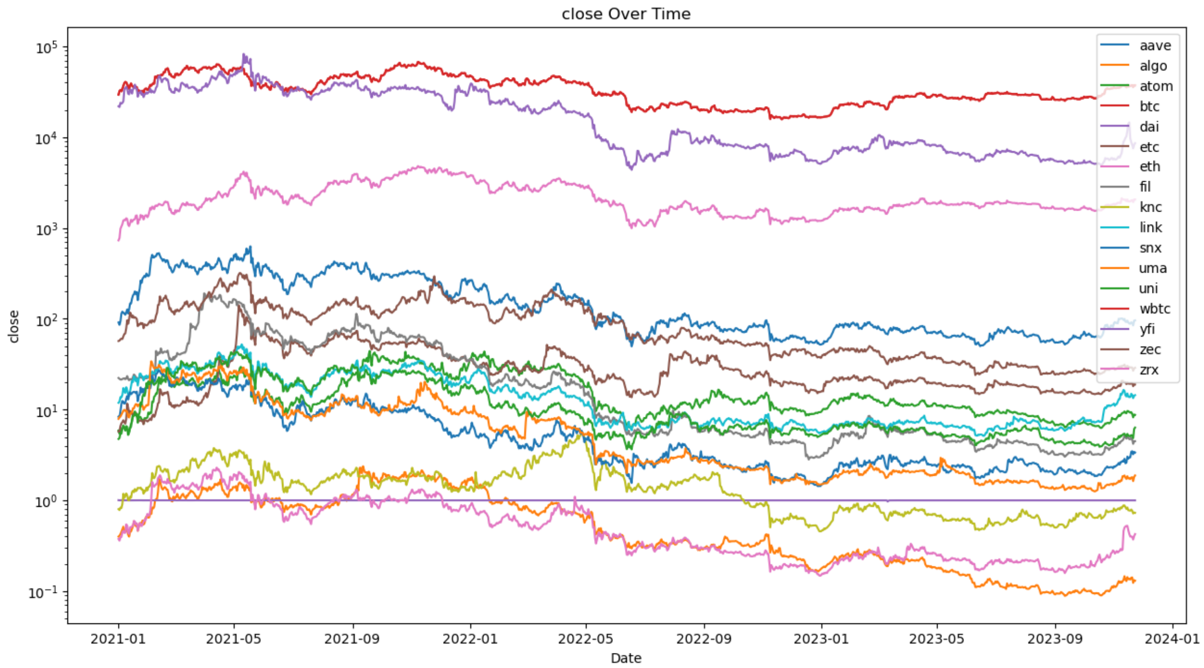
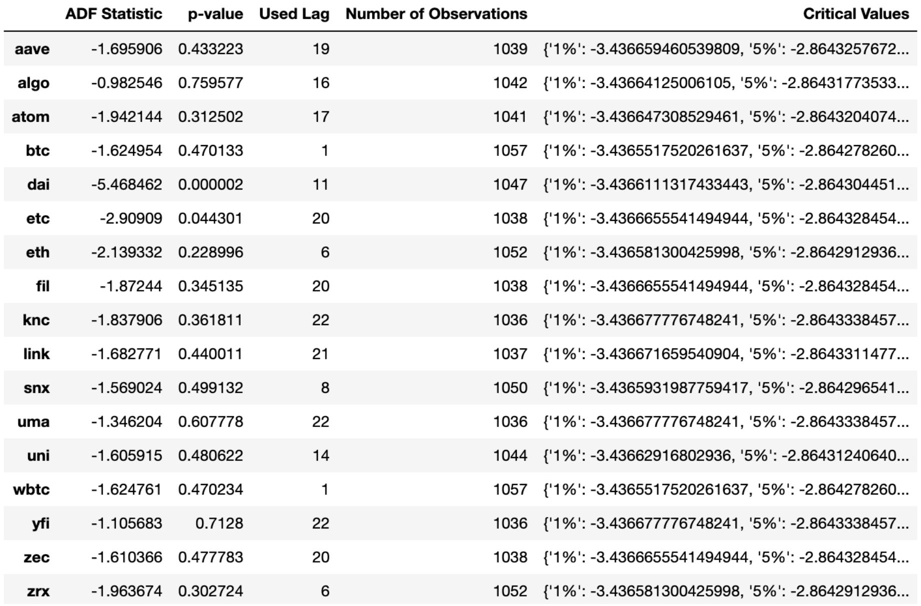


Figure - Line Chart of Close Price: There was noticeable volatility across all cryptocurrencies, which is characteristic of these types of assets. No clear seasonal patterns were observed. This could be due to the cryptocurrency market operating 24/7 and its global nature diminishing calendar effects typically found in stock markets. There are many short-term fluctuations that suggest periods of increased buying or selling activity. This could be in response to news events, technological developments, or changes in trader sentiment. Several cryptocurrencies demonstrate clear long term upward and downward growth trends suggesting non-stationary data.

2. Augmented Dickey Fuller (ADF): Augmented Dickey Fuller, a common statistical test to determine if a time series is stationary, was applied to the features and close price target. The ADF test provides critical values for different confidence levels (1%, 5%, and 10%), p-value, and a parameter for the number of lags of the time series to include in the test automatically determined based on various criteria such as the Akaike Information Criterion (AIC). If the ADF Statistic is lower than the critical value at 5% (more negative), and the p-value is below 0.05, you can reject the null hypothesis in favor of the alternative hypothesis, meaning the time series is stationary. Based on the Augmented Dickey Fuller (ADF) test results for close price, it was found that only the cryptocurrencies DAI and Ethereum Classic (ETC) exhibited stationarity at the 5% confidence interval. In the case of DAI, this was expected as it is a stable coin pegged to the U.S. dollar and thus experiences nominal price change under most market conditions. Consequently, this suggested that the order of differencing for both DAI and ETC in model training would be 0. The other 15 cryptocurrencies analyzed do not display stationarity and required differencing before modeling.

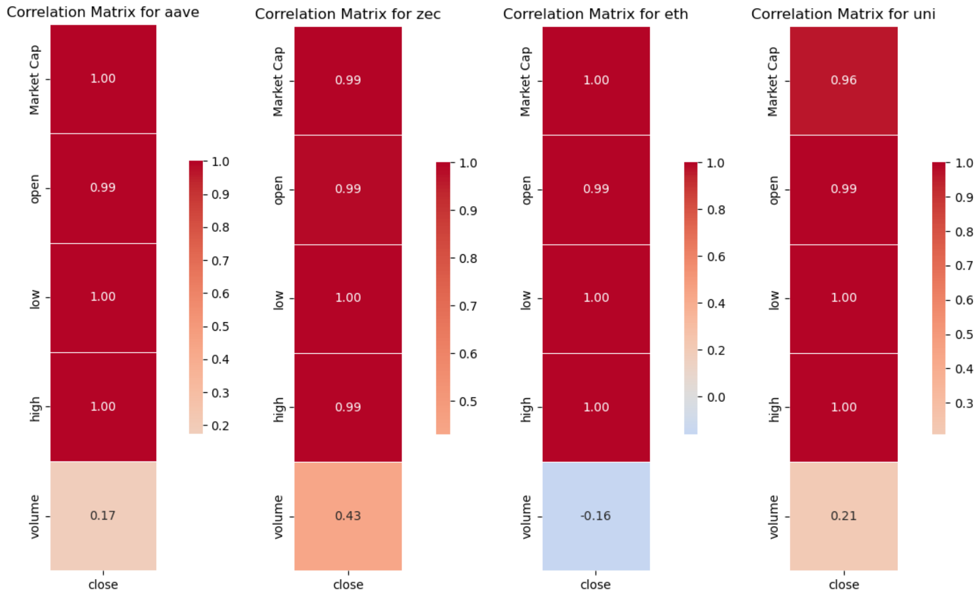
Figure - Close Price Augmented Dickey Fuller (ADF)



	ADF Statistic	p-value	Used Lag	Number of Observations	Critical Values
aave	-1.695906	0.433223	19	1039	{'1%': -3.436659460539809, '5%': -2.8643257672...
algo	-0.982546	0.759577	16	1042	{'1%': -3.43664125006105, '5%': -2.86431773533...
atom	-1.942144	0.312502	17	1041	{'1%': -3.436647308529461, '5%': -2.8643204074...
btc	-1.624954	0.470133	1	1057	{'1%': -3.4365517520261637, '5%': -2.864278260...
dai	-5.468462	0.000002	11	1047	{'1%': -3.4366111317433443, '5%': -2.864304451...
etc	-2.90909	0.044301	20	1038	{'1%': -3.4366655541494944, '5%': -2.864328454...
eth	-2.139332	0.228996	6	1052	{'1%': -3.436581300425998, '5%': -2.8642912936...
fil	-1.87244	0.345135	20	1038	{'1%': -3.4366655541494944, '5%': -2.864328454...
knc	-1.837906	0.361811	22	1036	{'1%': -3.436677776748241, '5%': -2.8643338457...
link	-1.682771	0.440011	21	1037	{'1%': -3.436671659540904, '5%': -2.8643311477...
snx	-1.569024	0.499132	8	1050	{'1%': -3.4365931987759417, '5%': -2.864296541...
uma	-1.346204	0.607778	22	1036	{'1%': -3.436677776748241, '5%': -2.8643338457...
uni	-1.605915	0.480622	14	1044	{'1%': -3.43662916802936, '5%': -2.86431240640...
wbtc	-1.624761	0.470234	1	1057	{'1%': -3.4365517520261637, '5%': -2.864278260...
yfi	-1.105683	0.7128	22	1036	{'1%': -3.436677776748241, '5%': -2.8643338457...
zec	-1.610366	0.477783	20	1038	{'1%': -3.4366655541494944, '5%': -2.864328454...
zrx	-1.963674	0.302724	6	1052	{'1%': -3.436581300425998, '5%': -2.8642912936...

3. Correlation Analysis: Correlation analysis was conducted to observe how various features correlate with cryptocurrency close prices to identify potential predictors.

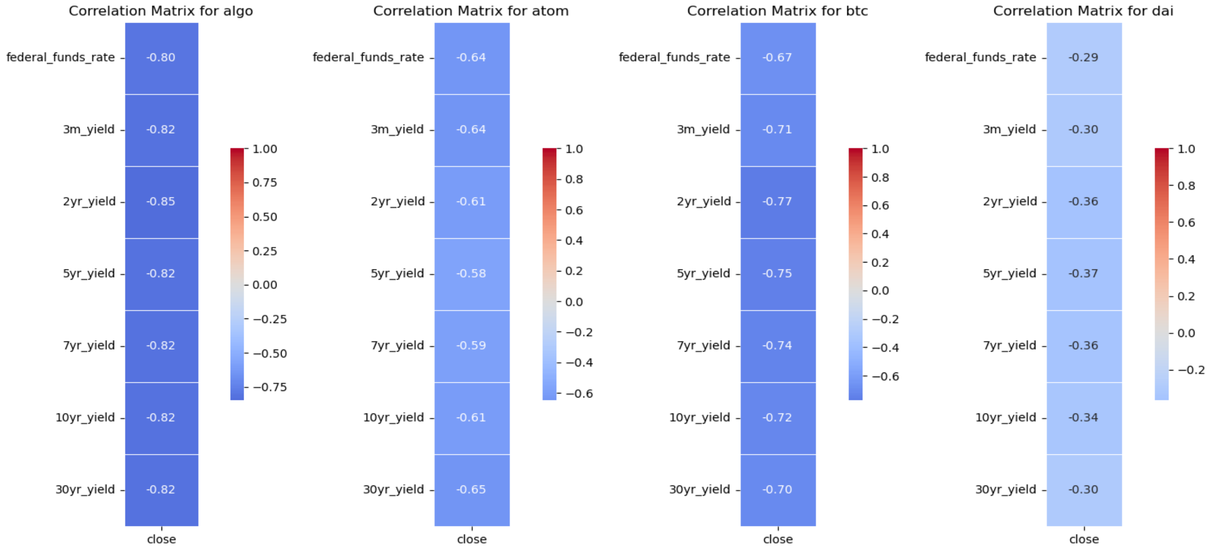
Figure - Close Price Feature Correlation (Market Data)



Feature	Average Correlation
Market Capitalization	0.8523667513752
Open Price	0.9553454940133
Low	0.9870911007228
High	0.9412074416379
Trading volume	0.2035085228112

Figure - Close Price Correlation (Market Data): All market metrics showed strong positive correlation, on average, with the target, except for trading volume. The closing price of a cryptocurrency appears to be most strongly correlated with its open, low, and high prices throughout the day, which is to be expected as these are all direct price measures. The market cap also shows a strong correlation, reinforcing the direct relationship between price and valuation. Weak correlation with volume may indicate that price movements are not always accompanied by proportional changes in volume. High volume can occur due to speculative trading, large transactions by institutional investors, or other factors independent of price changes. Given these findings all features excluding trading volume were considered in modeling.

Figure - Close Price Feature Correlation (Fed Rates)

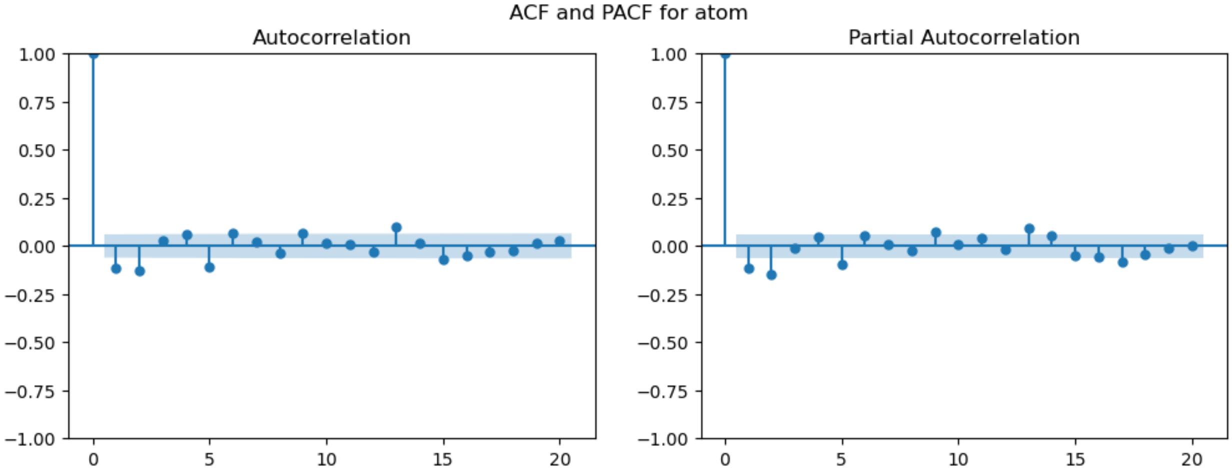


Feature	Average Correlation
Federal Funds Rate (interest)	-0.6701071603638565
3 month yield	-0.692868686165413
2 year yield	-0.738362371736154
5 year yield	-0.721589152470047
7 year yield	-0.709011433425357
10 year yield	-0.6964811773584506
30 year yield	-0.6746372764864438

Figure - Close Price Correlation (Fed Rates): Daily close prices of crypto currencies correlate strongly with two year, five year, and seven year U.S. treasury yields. Treasury yields are often considered a reflection of risk appetite in the financial markets. Cryptocurrencies are often considered high-risk assets. Strong negative correlation between yield and close price suggest that when yields are low, investors might be moving away from traditional safe-haven assets like U.S. Treasuries and into riskier assets like cryptocurrencies, driving up their prices. Given these findings the two, five, and seven year treasury yields were selected as potential features.

3, ACF and PACF: Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF) plots were used to identify a range of values for the order of the auto regressive (p) and moving average (q) terms.

Figure - ACF and PACF Plots



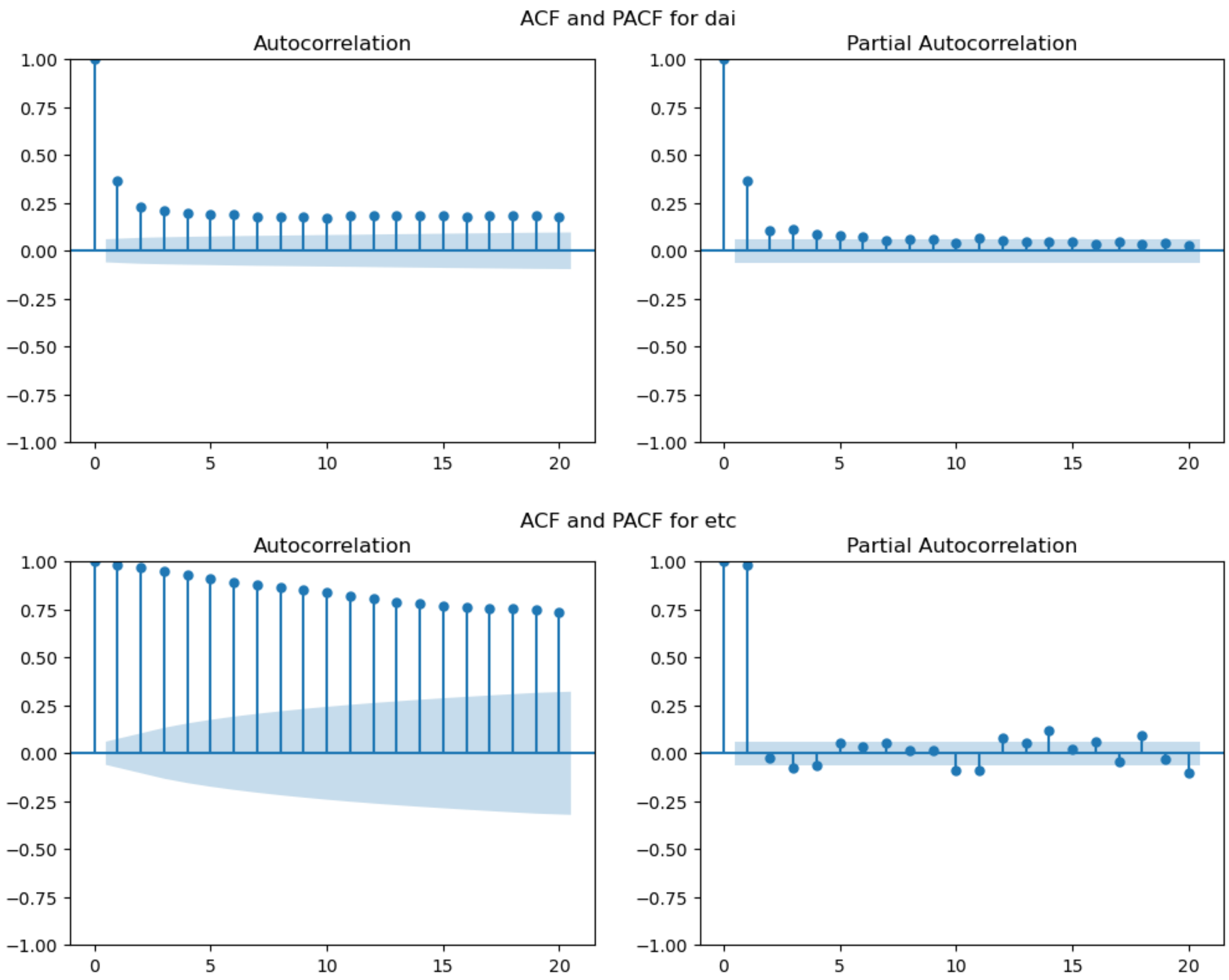


Figure - ACF and PACF Plots: 15 of the 17 coins demonstrated the same PACF and ACF trend with correlation of the series with its own lags decreasing gradually as the lags increase as demonstrated by plots for atom. Each saw a significant spike at 0, indicating correlation of the series with itself but the subsequent lags fall within the confidence interval, suggesting they are not statistically significant. Partial autocorrelations and autocorrelations insignificant for lags greater than 0, for 15 coins, suggest a p = 0 and q = 0. Dai and ethereum classic demonstrated different patterns in PACF and ACF. Dai performed similarly to the other 15 cryptocurrencies on the ACF, demonstrating no significant correlation beyond lag 0. However, the PACF plot also shows a significant spike at lag 0, followed by a significant spike at lag 1. These suggest an p = 1, q = 0. It is important to note that the target series for Dai was not differenced due to passing the Dickey Fuller test. Ethereum Classic (etc) was also not differenced but displayed vastly different trends. The ACF plot shows a gradual decline in the autocorrelation as the lags increase, with all the spikes well above the significance level (beyond the blue shaded area). This suggests a strong, positive autocorrelation at all observed lags. The slow decay of the ACF plot indicates that the series may contain a moving average component. The PACF shows a significant spike at lag 1 and 2 with subsequent lags within the confidence interval. Given the results models were tested on p and q values encompassing the observed range (0, 2) of PACF and ACF plots

3.3 Modeling

FINBERT

In our study, we deployed FinBERT utilizing the comprehensive Hugging Face transformers library, which provided an end-to-end pipeline for sentiment analysis. This pipeline included tokenization, formatting of input for the model's consumption, and the actual inference process.

The tokenizer specific to FinBERT played a pivotal role in preparing the input text for the model, executing critical steps such as tokenizing the text into subwords, appending necessary special tokens, ensuring uniform sequence length through padding, and generating attention masks. These preparatory steps are vital for the BERT architecture's operation, enabling it to understand and analyze the text's structure and semantic content effectively.

In addition to our sentiment analysis we leveraged batch processing for computational efficiency, dividing the preprocessed tweets into optimal batches of 32. This size maximized our computational resources and maintained system stability, avoiding memory overload. With a total of about 270,000 tweets, we efficiently managed this volume through approximately 8,437 batches for sentiment prediction with FinBERT. This batch processing significantly reduced our total analysis time compared to individual assessments.

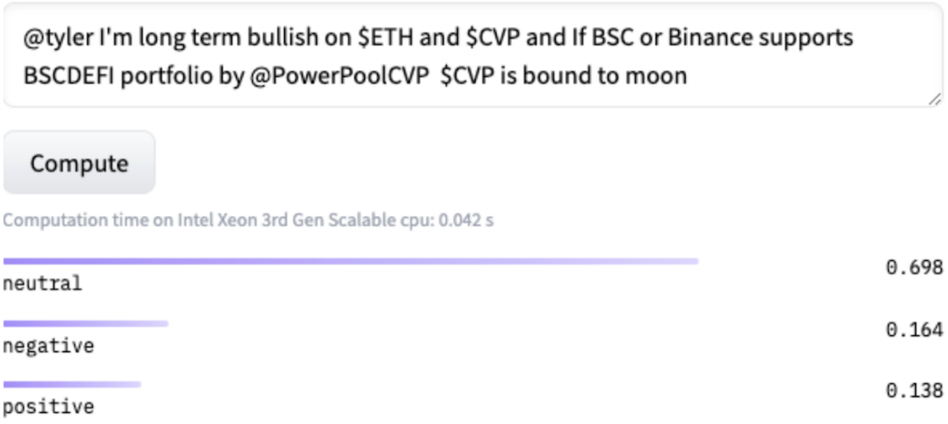
Rationale for No Gradient Calculation when using FinBERT:

Gradient calculations are integral during the training phase of machine learning models, facilitating the optimization of model weights via backpropagation. However, for a pre-trained model such as FinBERT, which is employed for inference tasks (like predicting sentiments on new data), gradient calculations are superfluous. Gradients, which are vectors that guide weight adjustments, are unnecessary when the model is already well-adjusted for the task.

Calculating gradients during inference, particularly for extensive datasets, can impose undue computational demands, resulting in inefficiencies. FinBERT has been meticulously fine-tuned on a substantial corpus of financial sentiment data, and it doesn't gain additional accuracy from gradient calculations during the inference phase. Therefore, we opted for a "no grad" approach, enabling us to process large volumes of data quickly and efficiently, eschewing the need for extra computational resources or excessive memory consumption

Our methodology incorporated thorough preprocessing to ensure data integrity, closely aligned with FinBERT's training on financial sentiment data. To ascertain the effectiveness of our approach, we conducted manual evaluations of FinBERT's performance on select data samples, contrasting it with outcomes presented on the model's Hugging Face web page. This verification ascertained that our preprocessing efforts were in sync with the model's training, guaranteeing accurate sentiment classification.

Figure 1 - Raw / Unclean Text Scores using FinBERT

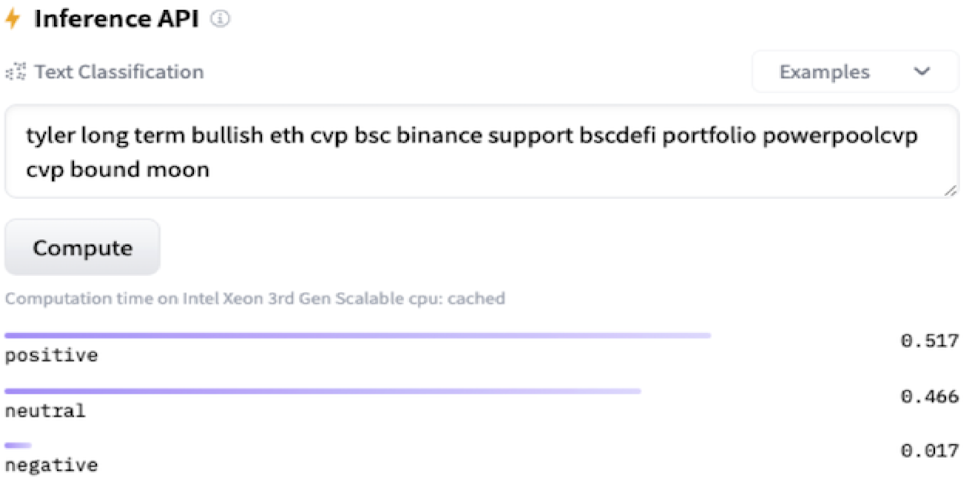


@tyler I'm long term bullish on $ETH and $CVP and If BSC or Binance supports BSCDEFI portfolio by @PowerPoolCVP $CVP is bound to moon

Computation time on Intel Xeon 3rd Gen Scalable cpu: 0.042 s

```
neutral     0.698
negative    0.164
positive    0.138
```

Figure 2 - Semi Cleaned Text Scores using FinBERT

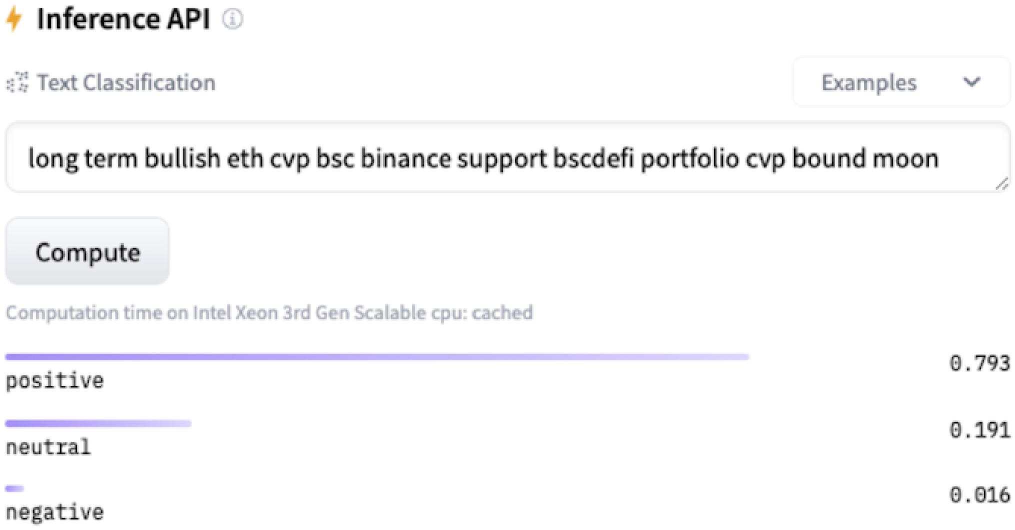


tyler long term bullish eth cvp bsc binance support bscdefi portfolio powerpoolcvp cvp bound moon

Computation time on Intel Xeon 3rd Gen Scalable cpu: cached

```
positive    0.517
neutral     0.466
negative    0.017
```

Figure 3 - Final Cleaned Text Scores using FinBERT



long term bullish eth cvp bsc binance support bscdefi portfolio cvp bound moon

Computation time on Intel Xeon 3rd Gen Scalable cpu: cached

```
positive    0.793
neutral     0.191
negative    0.016
```

The outcomes depicted in Figures 1, 2, and 3 elucidate a clear trajectory in sentiment accuracy as a function of data cleanliness. Initially, in Figure 1, the sentiment analysis of raw text yields a sentiment distribution with a prominent neutral classification, suggesting a level of ambiguity or a lack of clear sentiment cues in the unprocessed data. As we refine our data in Figure 2, the sentiment analysis on semi-cleaned text reveals a significant shift. The neutral classification decreases, which can be interpreted as a reduction in ambiguity due to the partial removal of irrelevant tokens that may dilute sentiment expression. Simultaneously, there's an uptick in positive sentiment scores, indicating that the semi-cleaning process begins to uncover the underlying optimistic sentiments present in the financial texts. The transformation culminates in Figure 3, showcasing the analysis of fully cleaned text. Here, the model's capability to discern sentiment is at its peak performance, with a dramatic escalation in positive sentiment classification, indicative of a robust bullish tendency within the financial discourse. Remarkably, there is a substantial decrease in neutral and negative sentiments, indicating that a cleaner and more focused dataset enables FinBERT to confidently classify positive sentiment expressions. The negative sentiment is minimized to its lowest across all stages, underscoring the impact of a meticulously cleaned dataset on the sentiment classification task.

These findings clearly illustrate the pivotal impact of preprocessing on sentiment analysis. The progression from raw to fully processed data not only enhanced the positive sentiment detection but also critically reduced the instances of neutral and negative classifications. This emphasizes that, for FinBERT, the quality of input data is tantamount to the clarity of sentiment expressed in its predictions and our preprocessing function was working effectively in accurately getting the true sentiments from the texts.

Daily Sentiment Score Calculation.

Among the 17 coins we were able to get adequate financial data for, the volume of tweet data varied significantly. Two coins, (BTC and ETH) had a disproportionately larger share of tweets compared to others. This disparity posed a challenge in creating a balanced and representative sentiment analysis model.

To address this, we implemented a thresholding approach, deciding to include only coins with a minimum of 50 tweets over our study period. This threshold was chosen to ensure that only coins with a reasonably sufficient volume of data were included, enhancing the reliability of our analysis. The number balanced the need to include a diverse range of coins while excluding those with sparse data that could potentially skew our analysis.

We also implemented a weighted average approach for our finalized sentiment scores. Given the varied tweet volumes across different coins, a simple average would have disproportionately favored coins with higher tweet volumes. To counteract this bias, we assigned weights inversely proportional to each coin's tweet volume. This weighting ensured that each coin's sentiment contributed more equitably to the overall sentiment score, regardless of its individual tweet volume.

Weighting Per Coin

$$W'_{coin} = \frac{1}{N_{coin}} \qquad W_{coin} = \frac{W'_{coin}}{\Sigma_{all\ coins} W'_{coin}}$$

(Where W'_{coin} *is the inverse tweet count, and* $\Sigma_{all\ coins} W'_{coin}$ *is the sum of inverse weights for all coins. Ensuring that the normalized weights across all coins sum up to 1.)*

Daily Sentiment Score

$$S_{w.coin,\ date} = W_{coin} \times S_{coin,\ date} \qquad SW_{date} = \Sigma_{all\ coins} S_{w.coin,\ date}$$

(*Where* $S_{coin,\ date}$ *is the average sentiment score for each con on a given date,* $S_{w.coin,\ date}$ *is the weighted sentiment score for a coin on a specific date, and* SW_{date} *is the sum of all weighted sentiment scores for each coin on that date)*

As a consequence of this weighting method, the aggregated daily sentiment scores across neutral, positive, and negative categories do not necessarily sum up to 1 for each day. The resulting sentiment scores should be interpreted as indicators of general sentiment trends across all coins considered each day, rather than as exact probabilities that would add up to 100%. The decision to consider all 3 scores individually as features, vs summing them in some fashion and indicating the sentiment with highest percentage, was done to keep as robust a feature set as possible for our modeling.

ARIMAX

In our study, 18 ARIMAX models were trained using the statsmodels library, incorporating both sentiment and non-sentiment data. The baseline model included lagged features of the target variable (closing price), supplemental market and federal rates features, selected based on strong correlation and best training performance.

Initial parameters for the models were derived from Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF) plots of manually differenced data. These parameters were further refined through grid search optimization. Our grid search explored predefined parameter ranges, including the manually differenced range of (0, 2), and broader ranges of (0, 4) and (0, 6). This approach enabled iterative training of models for each coin using an 80-20 test-train split.

Model evaluation focused on minimizing key criteria: Bayesian Information Criterion (BIC), Akaike Information Criterion (AIC), and Mean Absolute Percentage Error (MAPE). Additionally, model accuracy was assessed using MAPE, a widely recognized metric for measuring price differences in financial models. However, sparse data coverage for several coins in the sentiment dataset restricted our model comparisons to only 11 coins.

Random Forest Model

After completing the main time series analysis, we trained and evaluated Random Forest models using scikit-learn, both with and without sentiment data. Key evaluation metrics such as R-squared, Mean Squared Error, and Mean Absolute Percent Error were recorded at the end of each iteration to assess model performance.

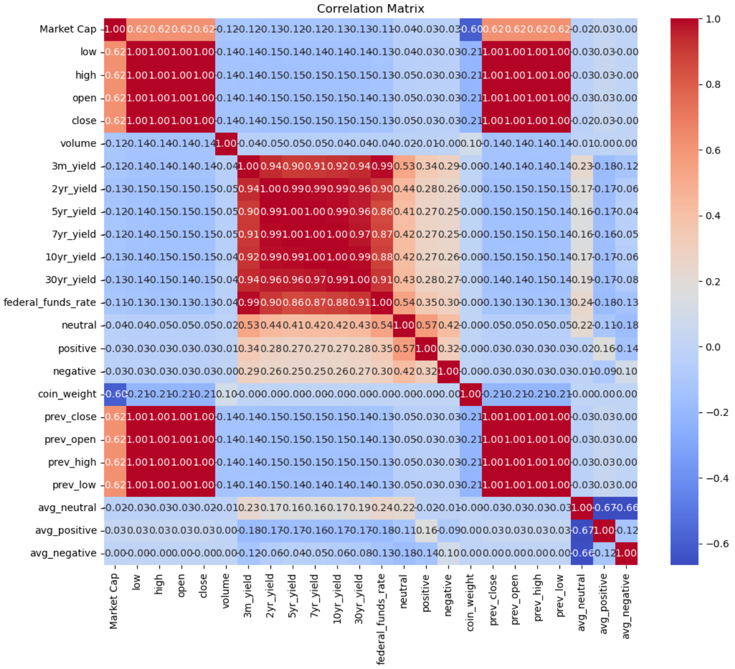


Figure: *(Sample Correlation Matrix used in feature selection. An interesting initial observation looks at our coin_weight, which seems to have a slight positive correlation of 0.10, possibly suggesting that coins that are more prevalently spoken of (lower coin_weight) have a slight tendency to have higher trading volume. Suggesting popular coins on Twitter might see slightly more trading activity.)*

Prior to feature selection, we performed a correlation analysis (see figure) to determine the relationships between features, aiding in the selection of the most relevant ones for the final model. An alternate sentiment scoring, disregarding weight and allowing the most prevalent coins to influence the score, was also considered, but did not seem to correlate anymore with financial features and was shelved. We manually created lagged features, in addition to others like daily percent change and a 7-day rolling average. These enhancements were aimed at improving model performance, especially after initial results indicated accurate model fitting but with high error rates.

Hyperparameter tuning was executed using GridSearch, testing various combinations of parameters such as max_depth and max_leaf_nodes. To address the disparity in financial data scales - with some coins' values in thousands of dollars and others in cents - we employed the RobustScaler for feature scaling. This step was crucial to normalize the financial data and facilitate effective model training. Hyperparameter optimization was performed both before and after scaling to ensure optimal tree performance.

4. Result

ARIMAX Model

No Sentiment Model Error:

Coin	MAPE (%)
aave	0.07438975898030502
algo	0.0014431929534399715
atom	0.008959707446415067
btc	0.68005706351027
etc	0.004043207611024659
eth	0.13303310450048086
link	0.011898601199658297
snx	0.003111361663218573
uni	0.0014784426614805558
wbtc	0.7512033330649626
zec	0.11055958025514617

Sentiment Model Error:

Coin	MAPE (%)
aave	0.08994690264705392
algo	0.001544080845970254
atom	0.005749028038817547
btc	1.3741354045937144
etc	0.007507029061620727
eth	0.24217534560304405
link	0.007038611168337327
snx	0.008251142411658967
uni	0.002129199286722082
wbtc	0.9589933763337966
zec	0.10006389210051302

The inclusion of sentiment data in ARIMAX modeling yielded mixed results across different cryptocurrencies. Notably, for some cryptocurrencies, incorporating sentiment data significantly altered the prediction error.

Bitcoin (BTC) and Wrapped Bitcoin (WBTC) models showed a stark difference in MAPE when sentiment data was included. The sentiment-based model for BTC exhibited a MAPE of approximately 1.37, more than double the error of the model without sentiment data, which had a MAPE of approximately 0.68. Similarly, the WBTC model with sentiment had a MAPE of 0.96, compared to a lower 0.75 when sentiment was excluded. This suggests that for these cryptocurrencies, sentiment data may have introduced noise rather than providing useful predictive signals.

However, the most dramatic difference was observed in the Yearn.finance (YFI) cryptocurrency, where the sentiment-inclusive model reported a MAPE of 58.05 compared to an exceedingly high MAPE of 409.19 for the model without sentiment data. This indicates that sentiment data dramatically improved the accuracy of the YFI model, despite the error still being substantial.
Conversely, for cryptocurrencies such as Aave (AAVE) and Filecoin (FIL), models that excluded sentiment data outperformed those that included it. AAVE's no-sentiment model had a lower MAPE of

0.074 compared to 0.089 with sentiment, and FIL showed a similar trend with a lower MAPE of 0.013 without sentiment against 0.022 with sentiment.

In the case of Algorand (ALGO), Dai (DAI), Ethereum Classic (ETC), and Kyber Network Crystal (KNC), the difference in MAPE between models was minimal, suggesting that sentiment data did not have a significant impact on predictive accuracy for these coins.

Cosmos (ATOM) and Chainlink (LINK) models performed better with sentiment data, reducing the MAPE from 0.0089 to 0.0057 and from 0.0119 to 0.0070, respectively. This implies a positive contribution of sentiment analysis to the predictive models for these particular cryptocurrencies.

Figure - Predicted vs. Actual

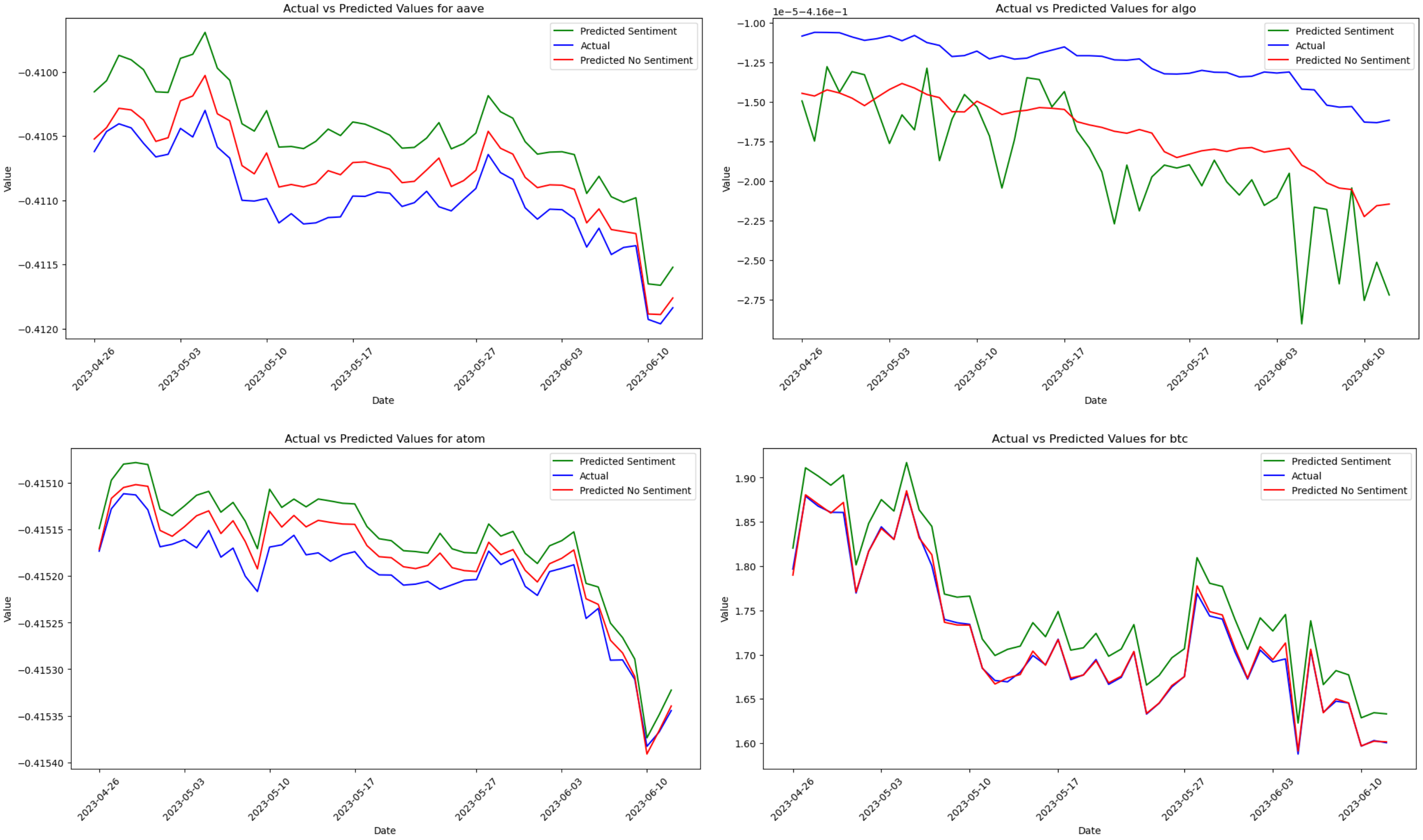


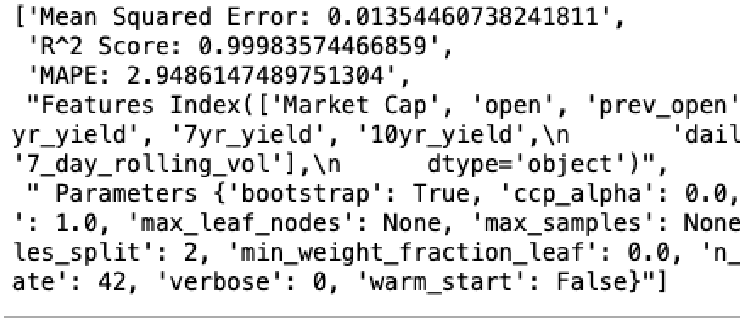
Figure - Predicted vs. Actual: The baseline model consistently outperformed the model with sentiment data, demonstrating better fit to the data suggesting that sentiment did not add any additional information that improves predictions. Both models do exhibit some optimistic bias toward overestimating predictions. In the case of the sentiment model this could be due to positive market sentiment influencing predictions.

Random Forest Model

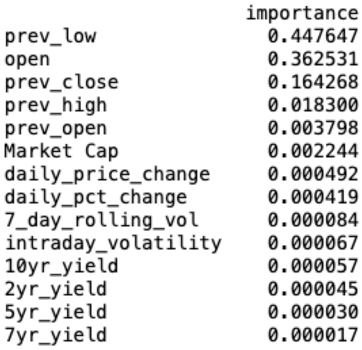
In the initial run, the model exhibited promising yet complex results. The R-squared value was high, indicating effective prediction by the closely correlated features. However, the Mean Squared Error (MSE) stood at approximately 50,000, a significantly high figure. Subsequent experimental runs using GridSearch to optimize tree parameters did not yield any improvement in error metrics. Furthermore, excluding features resulted in a marginal increase in error (by about 100), highlighting the challenge in feature selection.

To address these issues, we embarked on a sequential approach of feature engineering, selection, and hyperparameter tuning. Feature engineering efforts reduced the error to around 30,000, but this was still considerably high. We identified scaling issues, particularly due to significant outliers, as some coins had price points in the thousands while others were valued in cents. To normalize these disparities, we employed the RobustScaler, chosen for its effective handling of outliers compared to standard z-score normalization. This scaling adjustment notably resolved the scaling issue. Sentiment data was left unscaled to preserve its integrity.

Despite these efforts, the exploration was concluded at this stage. Current findings suggest that the sentiment data, in its current form, did not enhance the model's robustness. Agreeing with results from the ARIMA model. In fact, the inclusion of sentiment data resulted in a marginal performance decrement (approximately a 0.0003 increase in MSE and a similar decrease in R-squared). The high R-squared value raised concerns about potential overfitting, especially since the most influential features remained closely correlated with the closing price even after excluding the current day's high and low. Feature importance analysis further revealed that sentiment scores were among the least significant features, suggesting limited contribution to the model.

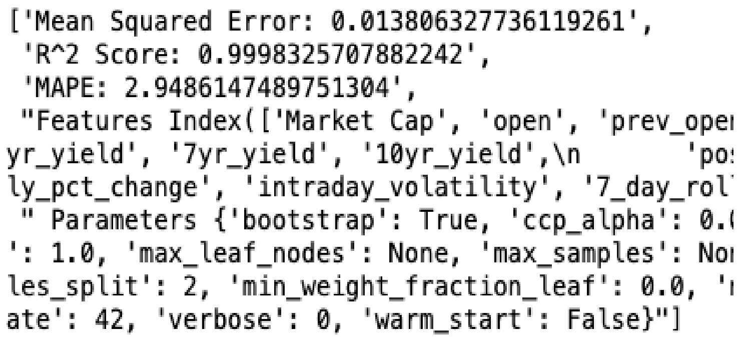


```
['Mean Squared Error: 0.01354460738241811',
 'R^2 Score: 0.99983574466859',
 'MAPE: 2.9486147489751304',
 "Features Index(['Market Cap', 'open', 'prev_open'
yr_yield', '7yr_yield', '10yr_yield',\n       'dail
'7_day_rolling_vol'],\n      dtype='object')",
 " Parameters {'bootstrap': True, 'ccp_alpha': 0.0,
': 1.0, 'max_leaf_nodes': None, 'max_samples': None
les_split': 2, 'min_weight_fraction_leaf': 0.0, 'n_
ate': 42, 'verbose': 0, 'warm_start': False}"]
```

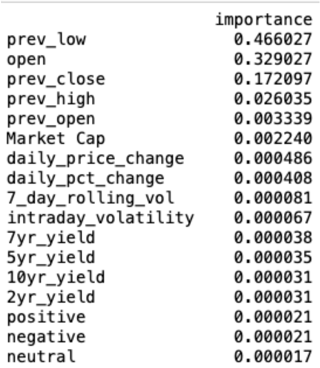


```
                      importance
prev_low                0.447647
open                    0.362531
prev_close              0.164268
prev_high               0.018300
prev_open               0.003798
Market Cap              0.002244
daily_price_change      0.000492
daily_pct_change        0.000419
7_day_rolling_vol       0.000084
intraday_volatility     0.000067
10yr_yield              0.000057
2yr_yield               0.000045
5yr_yield               0.000030
7yr_yield               0.000017
```

Figures. *(Above) Sample Results From one of the final models, indicating the evaluation metrics of the model, and the feature importance analysis. (Below). Sample results with the same hyperparameters, including the sentiment data. Note the low importance of the features, and the lack of improvement across eval metrics. MAPE has been multiplied by 100 to make percentage more intuitive.*



```
['Mean Squared Error: 0.013806327736119261',
 'R^2 Score: 0.9998325707882242',
 'MAPE: 2.9486147489751304',
 "Features Index(['Market Cap', 'open', 'prev_ope
yr_yield', '7yr_yield', '10yr_yield',\n       'po
ly_pct_change', 'intraday_volatility', '7_day_rol
 " Parameters {'bootstrap': True, 'ccp_alpha': 0.
': 1.0, 'max_leaf_nodes': None, 'max_samples': No
les_split': 2, 'min_weight_fraction_leaf': 0.0, '
ate': 42, 'verbose': 0, 'warm_start': False}"]
```



```
                      importance
prev_low                0.466027
open                    0.329027
prev_close              0.172097
prev_high               0.026035
prev_open               0.003339
Market Cap              0.002240
daily_price_change      0.000486
daily_pct_change        0.000408
7_day_rolling_vol       0.000081
intraday_volatility     0.000067
7yr_yield               0.000038
5yr_yield               0.000035
10yr_yield              0.000031
2yr_yield               0.000031
positive                0.000021
negative                0.000021
neutral                 0.000017
```

In contemplating future research directions, several avenues appear promising. First, there is potential in revisiting the method of sentiment score calculation. Considering an aggregated approach that treats all three sentiment scores as a single feature could offer a more streamlined and possibly more impactful variable. Alternatively, a revised sentiment calculation that incorporates different weighting mechanisms might provide deeper insights. Second, the scope of the data used in the study could be adjusted. A more dispersed distribution of coins over time might allow for clearer distinctions in results. In particular, a focused study on widely recognized cryptocurrencies like Bitcoin and Ethereum could yield more specific insights due to their widespread acceptance and usage. Lastly, a deeper analysis of various coin types, exploring how their specific uses and characteristics might influence or correlate with sentiment scores and financial metrics, could enhance our understanding of the complex dynamics within the cryptocurrency market. This approach would take into account the diversity in the cryptocurrency sector and how different market segments respond to sentiment.

4.3 Limitations

One of the primary limitations encountered in this study was data availability, a significant constraint given the ambitious scope of our initial pursuit. Our reliance on pre-compiled datasets was necessitated by the Twitter API's restriction on free access, limiting our ability to gather fresh, extensive sentiment data. This reliance on existing datasets inherently meant that our study was influenced by the scope and limitations of these prior compilations.

A notable challenge within the sentiment data was the uneven distribution of tweet volumes across different cryptocurrencies. Specifically, two coins, Bitcoin (BTC) and Ethereum (ETH), dominated the tweet volume, resulting in a skewed representation of sentiment data. This disparity in data availability and the consequent challenge in achieving a balanced and representative sentiment analysis model was a significant obstacle. Our attempts to address this imbalance through weighting methods aimed to capture cross-coin trends, but this approach may have contributed to the inconclusive performance metrics observed when incorporating sentiment data.

Furthermore, the financial data component of our study also faced limitations in terms of availability and comprehensiveness. We were able to reliably obtain financial data for only 17 cryptocurrencies, which might not have been a sufficiently representative sample to establish a general linkage between sentiment and the broader cryptocurrency markets. This limitation in the financial dataset potentially restricted our ability to draw more definitive conclusions about the relationship between market sentiments and cryptocurrency price movements.

5. Conclusions

In our study, we sought to unravel the complex relationship between cryptocurrency valuations and the sentiments expressed in social media, utilizing advanced machine learning techniques. Employing FinBERT, a specialized sentiment analysis model, we processed a substantial dataset of tweets, leveraging batch processing for efficiency. Concurrently, we delved into time series analysis with 18 ARIMAX models, integrating both sentiment and non-sentiment data, and focusing on minimizing key evaluation criteria.

Our findings, however, presented a nuanced picture. The sentiment data, when incorporated into the ARIMA and Random Forest models, did not significantly enhance their predictive accuracy -except in the rare outlier case-. In fact, we observed a slight degradation in model performance, evidenced by increased errors and potential overfitting issues. This was particularly evident in the sentiment-based models for BTC and WBTC, where the inclusion of sentiment data resulted in higher Mean Absolute Percentage Errors compared to models without sentiment data. This suggests that the sentiment data might have introduced more noise than predictive value.

In our Random Forest analysis, despite the application of sophisticated feature engineering and hyperparameter tuning techniques, the sentiment data's integration yielded inconclusive results. The high R-squared values indicated a strong fit to the data, yet the high Mean Squared Errors raised concerns about the models' generalizability and potential overfitting. Moreover, the sentiment data's minimal impact on feature importance further corroborated its limited utility in enhancing model robustness.

In conclusion, our comprehensive exploration into the integration of social sentiment data with financial models for cryptocurrency valuation revealed its limited effectiveness. This outcome challenges the prevailing assumption that social media sentiment is a strong predictor of cryptocurrency market movements and underscores the complexity of these markets.

6. Individual Tasks

Amari Parris:
Focus on data collection and integration. Harness coinbase, CoinGecko, AlphaAdvantge API to extract daily cryptocurrency price data. Performed ARIMA modeling for sentiment and non-sentiment model. Conducted exploratory data analysis on financial data and data cleaning. Contributed to sentiment data preprocessing for time series analysis, reindexing, interpolating, and ensuring stationary data., Developed dictionary for API data generation.

Parth Malik:
Sourced the influencers and general public's tweets data and created the final merged dataset. Led the exploratory data analysis using NLP techniques to unveil key insights and primed the

data for deep sentiment analysis. Directed the preprocessing workflow, expertly tailoring text normalization to meet FinBERT's nuanced specifications and proficiently produced sentiment labels for comprehensive modeling.

Yaseen Ellison:
Contributed to preprocessing efforts of sentiment data. Integrated sentiment data into financial models, employing weighting approach to enhance model accuracy. Conducted parallel study with Random Forest models aimed at further investigating initial findings and providing a broader perspective on the data.

References:

1. Bachelier, L. (1900). "Theory of Speculation." Annales Scientifiques de l'École Normale Supérieure, Volume 17, pages 21-86.

2. Box, G. E., & Jenkins, G. M. (1970). Time Series Analysis: Forecasting and Control. San Francisco: Holden-Day.

3. Shiller, R. J. (1981). Do Stock Prices Move Too Much to be Justified by Subsequent Changes in Dividends? The American Economic Review, 71(3), 421-436.

4. Boser, B. E., Guyon, I. M., & Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. In Proceedings of the fifth annual workshop on Computational learning theory (pp. 144-152)

5. García-Gonzalo, E., Fernández-Muñiz, Z., Garcia Nieto, P. J., Sánchez, A., & Menéndez, M. (2016). Hard-Rock Stability Analysis for Span Design in Entry-Type Excavations with Learning Classifiers. Materials, 9(7), 531. https://doi.org/10.3390/ma9070531
6. Yukun Liu, Aleh Tsyvinski, Risks and Returns of Cryptocurrency, *The Review of Financial Studies*, Volume 34, Issue 6, June 2021, Pages 2689–2727, https://doi.org/10.1093/rfs/hhaa113
7. Abiodun, Micah. “15 Pitfalls of Using Fundamental Analysis in Cryptocurrency Investing.” Cryptopolitan, 10 Oct. 2023, www.cryptopolitan.com/fundamental-analysis-pitfalls-in-investing/.
8. Nasekin, S., Chen, C.YH. Deep learning-based cryptocurrency sentiment construction. *Digit Finance* **2**, 39–67 (2020). https://deliverypdf.ssrn.com/delivery.php?ID=677119021086099006104005115121120088127008049065074002106109013027029104077122081073029003016045000030051090115085029115117084057042094035072066013003119072092127031089032086024O2

91041170750840820070880800931030881050960290270690820881220641071061060l3&EXT=pdf&INDEX=TRUE

9. *Prosusai/finbert · hugging face*. ProsusAI/finbert · Hugging Face. (n.d.). https://huggingface.co/ProsusAI/finbert
10. E R, S. (2021, June 17). *Random Forest | Introduction to Random Forest Algorithm*. Analytics Vidhya.
11. https://www.analyticsvidhya.com/blog/2021/06/understanding-random-forest/
12. Alpha Vantage. (n.d.). API Documentation. Alpha Vantage. Retrieved from https://www.alphavantage.co/documentation/
13. CryptoDataDownload. (n.d.). Data. CryptoDataDownload. Retrieved from https://www.cryptodatadownload.com/data/
14. Coinbase. (n.d.). Advanced Trade API Reference. Coinbase. Retrieved from https://docs.cloud.coinbase.com/advanced-trade-api/reference
15. CoinGecko. (n.d.). CoinGecko API v3. CoinGecko. Retrieved from https://api.coingecko.com/api/v3/coins/list
16. Binance. (n.d.). Binance API Documentation. Binance. Retrieved from https://www.binance.com/en/binance-api
17. GitHub. (n.d.). Issue 8314: ARIMA model fails with many observations. GitHub. Retrieved from https://github.com/statsmodels/statsmodels/issues/8314
18. Stack Overflow. (2017, September 5). Pandas DateTimeIndex frequency is 'None' and can't be set. Stack Overflow. Retrieved from https://stackoverflow.com/questions/46217529/pandas-datetimeindex-frequency-is-none-and-cant-be-set
19. MDPI. (2022). Algorithms. MDPI. Retrieved from https://www.mdpi.com/1999-4893/15/7/230
20. Cross Validated. (n.d.). Getting ValueError: Input contains NaN when using ARIMA model on non-null data. Cross Validated. Retrieved from https://stats.stackexchange.com/questions/587848/getting-valueerror-input-contains-nan-when-using-arima-model-on-non-null-dat