Predicting Changes in Cryptocurrency Value using Financial Indicators and Social Signals

By Irsa Ashraf, David Hackett, Ian Jeong, Yunjoo Park, Kelley Sarussi

1. Introduction

1.1 Background

From their inception, cryptocurrencies were dismissed as a fad or worse, as a scam. While many still argue that cryptocurrencies are an economic bubble waiting to burst, investment firms and individual investors have begun to include cryptocurrencies as part of their portfolios. In 2021 the “meme” coin, Dogecoin, made headlines when its value skyrocketed. Many attributed its 2000-fold increase to tweets from Elon Musk and the attention it received on social media. Its value later dropped precipitously as did the value of most cryptocurrencies over the past few months.

1.2 Research Questions

While cryptocurrencies have become ubiquitous in the news and in our culture, there remains much speculation about what drives their spikes and drops in value. Do cryptocurrencies behave like other financial assets such as stocks or gold? Or are they more responsive to social signals, rising and falling at the whim of a positive or negative tweet? Our research question is to explore the drivers behind the value of cryptocurrency. We attempt to use both classic financial indicators as well as social indicators to predict whether two cryptocurrencies, Bitcoin and Dogecoin, will rise or fall on a given day. Instead of attempting to predict the amount of change, we will try the simpler classification problem of predicting the direction of change. We chose these two cryptocurrencies both due to their prevalence in the news but also due to the fact that Bitcoin represents a more established currency and Dogecoin, a so-called meme coin, still has more of a fringe status.

1.3 Policy Implications

Understanding the mechanisms behind the changing value in cryptocurrency has many policy implications. Crypto is still largely unregulated and while some have profited greatly from investing in it, others have seen huge losses. If cryptocurrency is indeed an economic bubble, its collapse could have severe consequences not only for individual investors but for economies globally. If we understand cryptocurrencies to act like classic financial assets, the same regulations applied to stocks may be suitable for crypto. However if crypto is mostly at the whim of social sentiment we may be in a new frontier or be witnessing a pyramid scheme. Additionally we have some interest in simply taking on a challenge. The stock market has been notoriously difficult to predict, but if crypto proves to be very responsive to social sentiment we can perhaps make better predictions on its rise and fall.

2. Datasets and Features

2.1 Financial Data

2.1.1 Bitcoin and Dogecoin

We pulled our Bitcoin dataset from Kaggle. The raw dataset included the daily open and close prices as well as the high price and low price for that day and the trading volume. The data spans September 17th, 2014 to March 25th, 2022. Due to the availability of data in other datasets we opted to focus on a time range between October 20th, 2014 to February 9th, 2021 which left us with 1,558 observations. Figure 1 shows the price of Bitcoin over time while Figure 2 shows the daily increases and decreases. Over that time period Bitcoin saw a daily increase 54.8% of the time.

Figure 1: Bitcoin Price Over Time

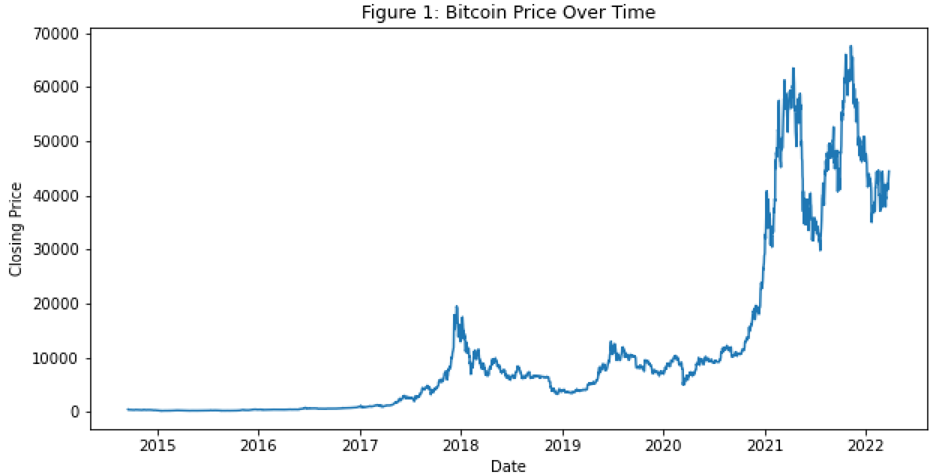
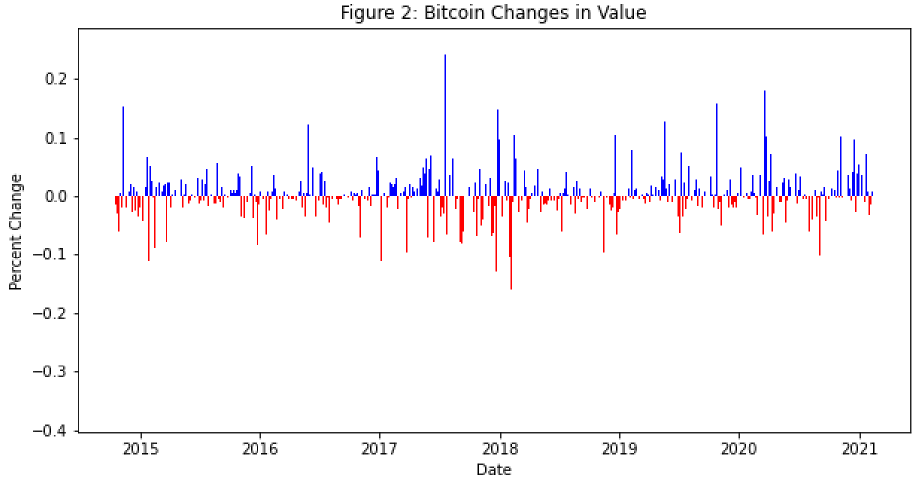


Figure 2: Bitcoin Changes in Value



We pulled our Dogecoin dataset from kaggle as well. Similarly the raw dataset included the daily open and close prices as well as the high price and low price for that day and the trading volume. The data spans from September 17th, 2014 to April 15th, 2021. Like Bitcoin we chose to focus on the time range October 20th, 2014 to February 9th, 2021 which left us with 1.637 observations. Figure 3 shows the price of Dogecoin over time while Figure 4 shows the daily increases and decreases. Over that time period Dogecoin saw a daily increase 18.1% of the time.

Figure 3: Dogecoin Price Over Time

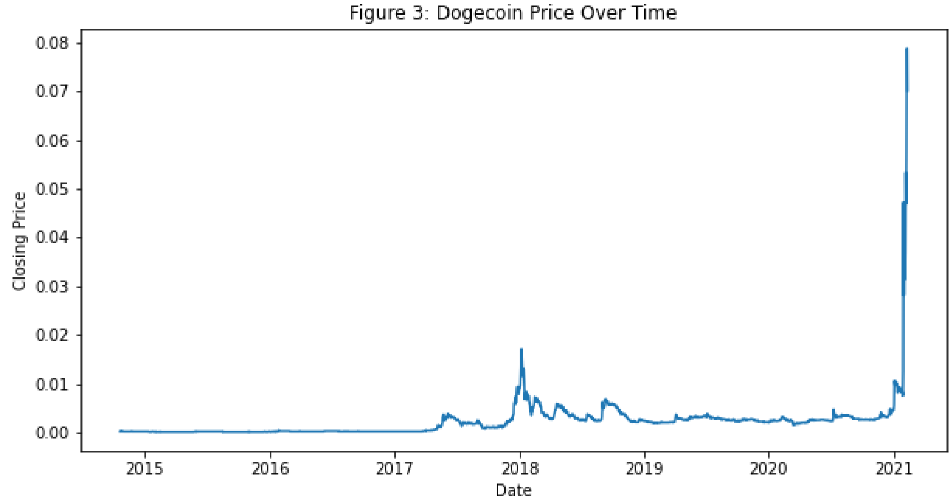
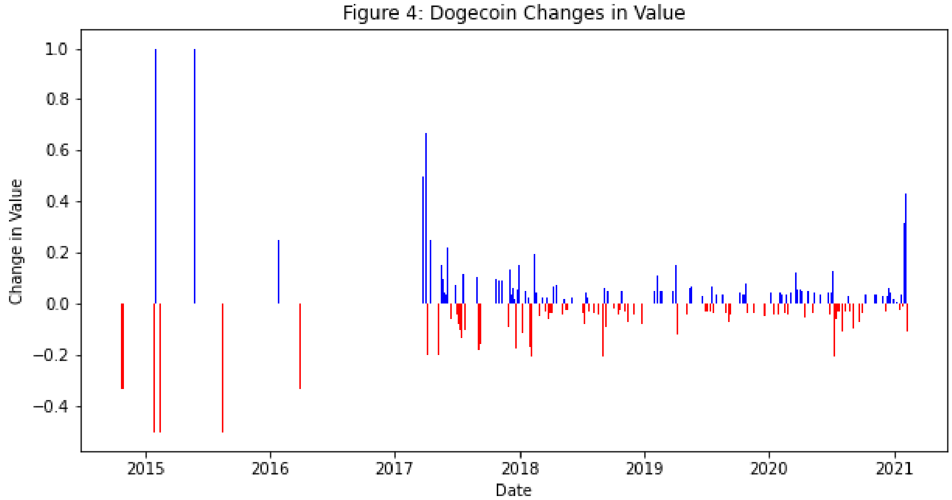


Figure 4: Dogecoin Changes in Value



From these two datasets we created the following features:

Feature	Description
Open	The opening crypto currency value (USD) for a day
High	The high crypto currency value (USD) for a day
Low	The low crypto currency value (USD) for a day
Close	The closing crypto currency value (USD) for a day
Volume	The volume of trades for a given day.
Daily_Change	The difference in crypto currency value (USD) from open to close
Daily_Change_Perc	The percent difference in crypto currency value (USD) from open to close
Daily_Change_Ind	1 if the crypto value increased on that day, 0 otherwise
MACD	The moving average convergence divergence. An indicator of the momentum of the current value of an asset.
PROC_3	Price Rate of Change for three days
PROC_5	Price Rate of Change for five days
PROC_10	Price Rate of Change for ten days
wpr	The Williams Percentage Range, or the current closing price in relation to the high and low of the past N day
sto_os	Stochastic Oscillator. Location of the close relative to the high-low range over a set number of periods
RIS	Relative Strength Index, an indicator of whether the cryptocurrency has been overbought or underbought.
streak	The number of days in a row the cryptocurrency has either risen (indicated by a positive number) or fallen (indicated by a negative number)

2.1.2 S&P 500 and Gold

We gathered data on daily S&P 500 and gold prices over time as a way to factor into our models what was going on in the financial markets over time. These data ranged from October 13, 2014 through February 20, 2021, for a total of 1,578 observations. The S&P 500 minimum and maximum values were 1,829 and 3,916, respectively. The minimum and maximum gold prices were 1,050 and 2,069, respectively. From these data, we were able to create the following features:

Feature	Description
Close/Last_Gold	Previous day's close price for gold
Open_Gold	Opening price for gold that day
Daily_Change_Gold	Difference in value (USD) from open to close for gold on a day
Daily_Change_Perc_Gold	The percent change in value from open to close for gold on a day
Increased_Gold	Indicator, 1 if gold's value increased from open to close, 0 otherwise.
Close/Last_SP500	Closing price for S&P 500
Open_SP500	Open price for S&P 500
Daily_Change_SP500	Difference in value (USD) for the SP500 from open to close
Daily_Change_Perc_SP500	Percent change of SP500 from open to close
Increased_SP500	Binary; whether or not the S&P 500 opened higher than its previous day
RIS_Gold	Relative Strength Index, an indicator of whether gold has been overbought or underbought.
RIS_SP500	Relative Strength Index, an indicator of whether the SP500 has been overbought or underbought.

2.2 Social Media Sentiment Data

2.2.1 Tweets

For our tweet dataset we again pulled from Kaggle. The dataset included tweets from January 3rd, 2013 to February 9th 2021. The dataset included roughly 825,000 tweets of which 294,562 were related to Bitcoin and 4,602 were related to Dogecoin. The most tweets in a day for Bitcoin were 1,690 and the least 1, with 121.3 being the average. Replies and retweets were also counted. The dataset also included sentiment scores for each tweet using VADER. Each tweet was given a compound, negative, neutral, and positive score. The compound score ranges from 1 to -1 with 1 being a completely positive sentiment and -1 being a completely negative sentiment. The average compound score for Bitcoin was 0.24 with the max being 0.96 and the minimum score being -0.83. The most tweets for Dogecoin on a given day was 216 with 0 being the minimum. Dogecoin averaged around 6.5 tweets for this time period. Its compound score topped out at 0.98 and had a low of -0.87. Its average was 0.45.

2.2.2 Google Trends

For our google trends we made use of the google trends api to pull the daily trend score for 90 day periods (the max) starting in 2014 to 2021. From these two datasets we created the following features:

Feature	Description
goog_trend_score	The google trend score for the crypto currency, ranges from 1 to 100
count	The number of tweets related to that cryptocurrency on a day
likes_count	The number of likes for tweets related to that cryptocurrency on a day
replies_count	The number of replies for tweets related to that cryptocurrency on a day
retweets_count	The number of retweet for tweets related to that cryptocurrency on a day
pos_weighted	The average positive sentiment score for tweets on a given day weighted by number of retweets
neg_weighted	The average negative sentiment score for tweets on a given day weighted by number of retweets
compound_weighted	The average compound sentiment score for tweets on a given day weighted by number of retweets
count_avg7	A rolling 7 day average of number of daily tweets related to that cryptocurrency
count_daily_diff	The difference in number of tweets from the day before
count_weekly_diff	The difference in number of tweets for a day from the rolling average for that day
replies_count_avg7	A rolling 7 day average of number of daily replies to tweets related to that cryptocurrency
replies_count_daily_diff	The difference in number of replies from the day before
replies_count_weekly_diff	The difference in number of replies for a day from the rolling average for that day
retweets_count_avg7	A rolling 7 day average of number of daily retweets of tweets related to that cryptocurrency
retweets_count_daily_diff	The difference in number of retweets from the day before
retweets_count_weekly_diff	The difference in number of retweets for a day from the rolling average for that day
likes_count_avg7	A rolling 7 day average of number of daily likes of tweets related to that cryptocurrency
likes_count_daily_diff	The difference in number of likes from the day before
likes_count_weekly_diff	The difference in number of likes for a day from the rolling average for that day

compound_weighted_avg7	A rolling 7 day average of the compound sentiment score of tweets weighted by retweets
compound_weighted_daily_diff	The difference in compound sentiment score weighted by retweets from the day before
compound_weighted_weekly_diff	The difference in compound sentiment score weighted by retweets from the rolling average
pos_weighted_avg7	A rolling 7 day average of the positive sentiment score of tweets weighted by retweets
pos_weighted_daily_diff	The difference in positive sentiment score weighted by retweets from the day before
pos_weighted_weekly_diff	The difference in positive sentiment score weighted by retweets from the rolling average
neg_weighted_avg7	A rolling 7 day average of the negative sentiment score of tweets weighted by retweets
neg_weighted_daily_diff	The difference in negative sentiment score weighted by retweets from the day before
neg_weighted_weekly_diff	The difference in negative sentiment score weighted by retweets from the rolling average

2.3 Feature Selection/Engineering

From our datasets we were able to generate a range of features including features meant to capture daily changes and more long term changes in financial indicators and social signals. As earlier baseline models did not perform as expected, we added more features in an attempt to increase our accuracy. With this in mind we worried that some features would be collinear while others would just add noise. In order to reduce our feature set we check how much each feature correlated with the direction of change and correlated with other features. Additionally we ran an initial random forest model on the features in order to figure out which provided the most information gain:

Table 1: Financial Features Correlation and Importance for Daily Value Change

Feature	**Correlation**	**Importance**
RSI	0.0769	0.0885
RSI_Gold	0.0709	0.0893
RSI_SP500	0.0657	0.0904
Daily_Change_Perc_SP500	0.0469	0.0906
wpr	0.0304	0.0866

Table 2: Social Features Correlation and Importance for Daily Value Change

Feature	**Correlation**	**Importance**
compound_weighted_avg7	0.0858	0.0437
pos_weighted_avg7	0.0856	0.0429
compound_weighted	0.0696	0.0433

Daily_Change_Perc_Gold	0.0277	0.0872
streak	0.0228	0.0424
PROC_3	0.0142	0.0836
MACD	0.0115	0.0921
Daily_Change_Perc	0.0092	0.0792
PROC_10	0.0075	0.0908
PROC_5	0.0000	0.0791

pos_weighted	0.0664	0.0432
likes_count_daily_diff	0.0617	0.0404
likes_count_weekly_diff	0.0607	0.0411
retweets_count_weekly_diff	0.0512	0.0412
retweets_count_daily_diff	0.0434	0.0411
replies_count_avg7	0.0396	0.0375
neg_weighted_avg7	0.0386	0.0452
neg_weighted	0.0382	0.0329
goog_trend_score	0.0296	0.0403
pos_weighted_weekly_diff	0.0292	0.0402
compound_weighted_daily_diff	0.0289	0.0430
neg_weighted_weekly_diff	0.0215	0.0401
pos_weighted_daily_diff	0.0210	0.0382
replies_count_weekly_diff	0.0201	0.0425
compound_weighted_weekly_diff	0.0195	0.0417
count_daily_diff	0.0164	0.0347
retweets_count_avg7	0.0137	0.0361
count_avg7	0.0109	0.0336
replies_count_daily_diff	0.0098	0.0395
count_weekly_diff	0.0095	0.0397
neg_weighted_daily_diff	0.0083	0.0413
likes_count_avg7	0.0024	0.0367

3. Method

We explored four types of machine learning models: logistic regression, neural network, random forest, and adaboost. We chose these models because these were commonly utilized and compared in the papers we found in our literature review. We fit our models to our train datasets and made predictions using test datasets. Our Bitcoin datasets (standardized and unstandardized) contained 1,246 in the trained sets and 312 observations in the test sets, and our Dogecoin datasets contained 1,309 observations in train sets and 328 observations in the test sets. We used an 80/20 train test split, and set the random state to 1234 for all our models. We labeled the data with a 1 if the closing price was higher than the day's opening price and with a 0 otherwise.

3.1 Logistic Regression

We used logistic regression to establish a baseline model and accuracy because logistic regression is a relatively simple classification model, and was frequently used in the literature we found related to predicting cryptocurrency prices. We ran a logistic regression on a variety of features until they converged, and used those models on separated test data. We used scikit-learn's Logistic Regression classifier, which relies on the sigmoid function, to classify our data into our binary labels. One benefit of using the Logistic Regression model is its simplicity. We reasoned that, if we could achieve a reasonable accuracy with the Logistic Regression, honing our classification accuracy would then be a matter of pruning features, and we could compare this to our other more dynamic and 'customizable' models.

3.2 Random Forest

We chose Random Forest as one of our models because we had many possible features and Random Forest allows for feature selection (based on information gain after splitting on each feature), and again was a popular method utilized in the related literature. Random Forest is a flexible model, and another advantage of using it is that it reduces bias. However, this comes at a cost of higher variance and (due to the bias-variance tradeoff) and lower interpretability. Additionally, most of the papers that we read had used random forests as their models to predict the direction of price for Bitcoin. We used scikit-learn's built-in Random Forest Classifier function to fit our model and make the predictions. We also used the built-in functions to compute accuracy, precision, recall and f1 scores to judge the performance of each of our models.

3.3 Adaboost

We chose adaboost because it is an ensemble method that uses boosting to re-assign instances that are misclassified with higher weights. This is useful because it can help reduce bias and variance in our estimates. An advantage of adaboost is that it is less prone to overfitting than a full decision tree classifier since the input parameters are not jointly optimized, so adaboost can improve the accuracy of weak classifiers. By limiting the number of levels in a tree (pre-pruning) and removing vertices that don't improve performance (post-pruning), we can avoid overfitting. Adaboost can also allow us to determine non-linear relationships between parameters, which we might expect to be useful based on our selected features. However, using adaboost increases the risk of our model being less interpretable than other models.

3.4 Neural Network

We chose to implement a neural network because unlike a simple perceptron, it can learn things that are not linearly separable. The papers we reviewed used a variety of neural networks, such as multilayer perceptron, LSTM (long-short-term-memory) and gated recurrent unit. One benefit of using a neural network is that, with additional more units and more layers, the model can "extract" more patterns from the dataset. However, blindly adding more units and layers to optimize accuracy can lead to overfitting. Therefore, when constructing a neural network, one must be careful to not overfit the model to the training data.

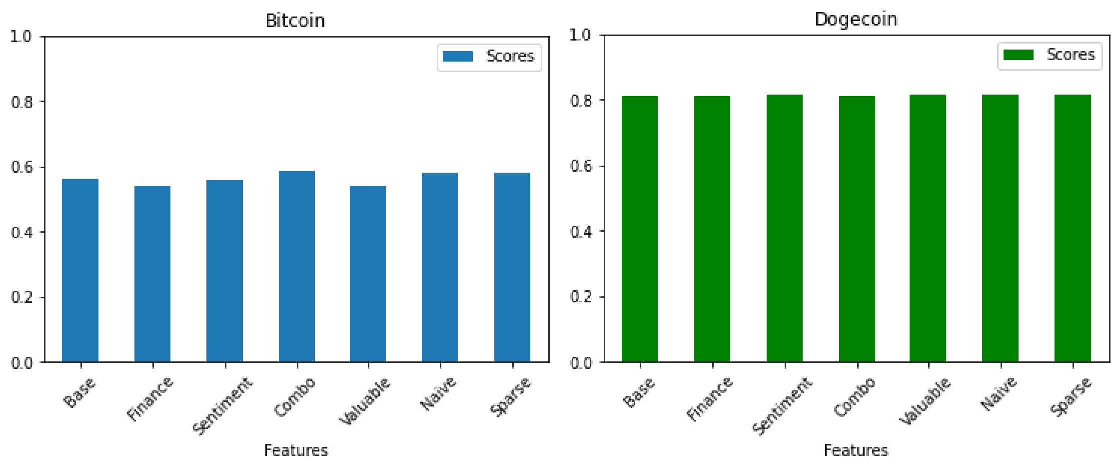
Although LSTM initially seemed like a great fit, given that it can hold "memory" and is advantageous for predictions in serial or sequential pattern, we thought that a simpler, feedforward neural network would suit our research question better. Our project aims to do a binary classification on whether a crypto currency closing price went up or not, and in our approach, we shuffled the dataset to investigate whether a certain constellation of financial and social media sentiment features could indicate whether a cryptocurrency's closing price will increase.

4. Experiment Setup

We ran each model three times, using a different set of features each time. To see the effect of social media on the price of Bitcoin and Dogecoin, we first ran each model with a set of social media features. We also ran each model with a set of financial indicator features. Finally, we ran each model with a combination of social media and financial indicator features. Our goal was that this combination of models and features would allow us to see which types of features were more important in determining the direction of price change of the cryptocurrency, and also determine which features were more and less important to each model.

4.1 Logistic regression

We used scikit-learn's Logistic Regression model, and set the max_iter attribute to 1000 so that we could achieve convergence, and then selected different features to attempt to maximize accuracy. One way we did this was to run different models using our standardized data and then evaluate the absolute value of the coefficients and find which features had the largest coefficients. Doing so yielded a relatively high accuracy score compared to other features, and for Bitcoin, we found that our highest test accuracy scores were achieved by using a combination of financial and sentiment features. For Dogecoin, we found that using different features and categories of features yielded no difference in accuracy, but a slight difference in precision–for financial features, the model recovered a precision of 33% and an F1 of 31%, but 0 for both social media and combined feature categories.



Logistic Regression performance for different feature combinations

4.2 Random Forest

For the Random Forest, we first started out with a baseline model that used the scikit-learn's default hyperparameters (with a random state of 1234) and three sets of features – financial indicators, social indicators and a combination of the two. This baseline model gave the highest score for the financial indicators and the lowest for the social indicators. After this, we tuned the hyperparameters using scikit-learn's built-in Grid Search function. We used 5-fold cross-validation in the Grid Search and tuned the following hyperparameters: n_estimators, max_features, oob-score and criterion. After using the best hyperparameters on the same set of features, our accuracy on models using all three sets of features increased slightly. Following this, we performed feature selection. Here, we used two approaches; scikit-learn's feature_importance_ attribute and a feature selection technique called "forward stepwise selection". Using these two methods, we got the best set of features and finally trained our model on the train dataset and predicted on the validate dataset to get the final accuracies. After optimizing our models for both Bitcoin and Dogecoin, we predicted on the test dataset that we had not touched so far and got our final accuracies and other metrics (given in the tables below).

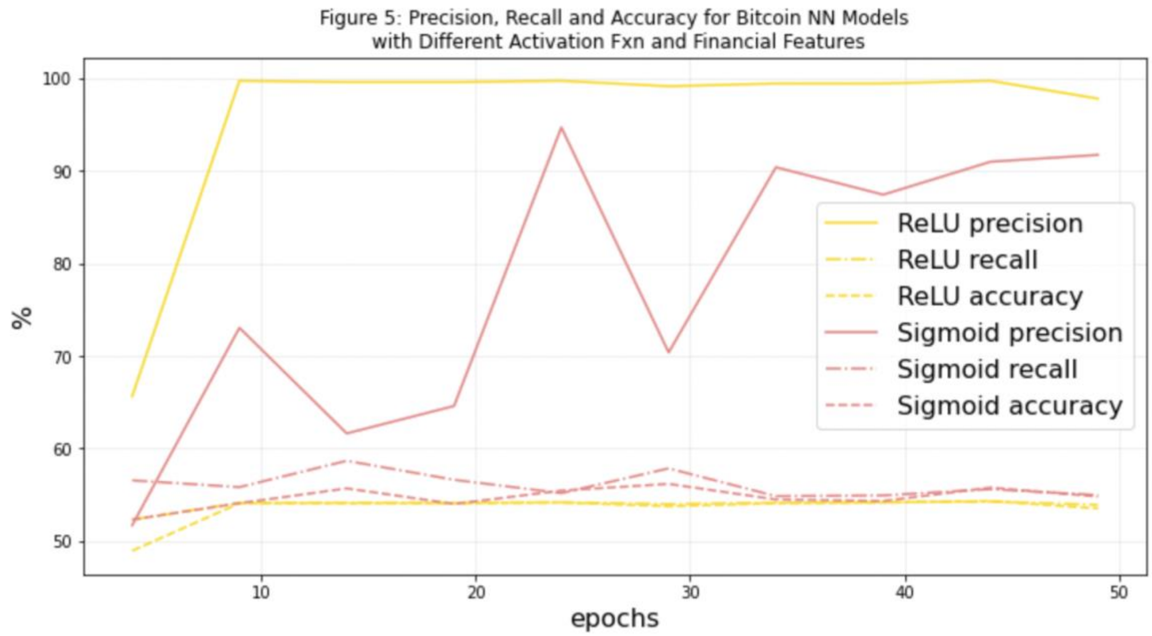
4.3 Adaboost

Adaboost randomly shuffles the data, so in order to make our model reproducible, we set the random seed to 1234. We ran the model using the default decision tree classifier, and explored making the models less weak by testing out tree depth of 1 and 2. We selected a tree depth of 1 for each model. We evaluated the model using repeated stratified k-fold cross validations and grid search to optimize the hyperparameters. We used three repeats and 10 folds. We tried 0.01 and 0.1 for the learning rate. We chose these learning rates because we did not want to move too slowly or quickly along the slope of the loss function. We explored different numbers of trees, trying 100, 500, and 1000 for the number of estimators, since a higher number of estimators is associated with a lower bias, and our goal was to limit the number of errors made. The optimal adaboost parameters for each model ran for Bitcoin and Dogecoin are shown in the table below:

TABLE 3: Adaboost Parameters and Hyperparameters						
	Bitcoin			**Dogecoin**		
	Financial Features	Social Media Features	**Combined Features**	Financial Features	Social Media Features	**Combined Features**
Tree Depth	1	1	1	1	1	1
Learning Rate	500	100	500	100	100	100
Number Estimators	0.01	0.1	0.1	0.01	0.01	0.01

4.4 Neural Network

We used the PyTorch library to implement our neural network. When initializing the neural network, we set the initial values of the weights to arbitrary, small numbers. In the process of training the model, the neural network uses the weights to first predict a value, and more importantly, then use the weights and the intermediate values from the forward propagation to update the weights using a process called back propagation. We toggled between ReLU and sigmoid as the activation function for the hidden layers in our neural networks. We compared the accuracy levels between the two activation functions and chose the activation function that led to a higher accuracy level (sigmoid, in all three different lists of features). Figure 5 captures the change in the accuracy levels through the iterations between Bitcoin financial feature models that utilize the ReLU and sigmoid functions. Though we tried different loss functions (neg log likelihood, mean squared error and cross entropy), we ultimately implemented cross entropy as our loss function because Jameslee (2019) identified cross entropy as the default loss function for binary classification. We also used a library called Optuna, which is a popular hyperparameter-tuning library, to identify the settings for the number of layers, number of units in each hidden layer, the rate for dropout layer, the learning rate and the optimizer that minimized validation loss.



5. Results

The Logistic Regression model for Bitcoin performed the highest when using a combination of both financial and sentiment features, at about 58% test accuracy. Precision was approximately 61%. For Dogecoin, we found no difference in using different features, and the accuracy held steady at 81%. While this may seem like a good accuracy rate, we notice that our models had a precision and F1 value at or near 0. This is likely given the nature of Dogecoin prices, which had mostly labels that equaled 0. Whenever the model predicted 0, it was highly likely to be correct, so the model likely learned that predicting 0 will more likely than not be correct. As a result, we cannot give much credence to this model.

The Random Forest model for Bitcoin performed reasonably well when we used only financial features (with an accuracy of 57%). Upon adding the social indicators, we suddenly saw a drop in the accuracy and so we trained our final tuned model on only the financial features and used those features in the final test dataset too when making our predictions. For Dogecoin, however, we saw the highest accuracy and best performance when we used a combination of the financial and social indicators and so our model was trained on those features and we used this set of financial and social indicators to make our final predictions on the test set too.

The adaboost model for Bitcoin with social media features had the highest accuracy of all three sets of features, at 59%. Compound_weighted_avg7 and count_weekly_diff were the most important features. Our combined set of features for Bitcoin adaboost model accuracy rate was 54%, which means that our features did not predict the label better than the daily increase of 54.8%. F1 was highest for the social media indicator model for Bitcoin at 69%. The Dogecoin adaboost model had an accuracy rate of 81%, which, similar to Bitcoin, indicated that our features were likely not doing much better than guessing since Dogecoin prices decrease nearly 82% of the time in our sample period.

For both ensemble methods, since Dogecoin was skewed towards a zero label for most days, we were unable to estimate precision or F1 because we ended up with zero observations of positively classified labels. Regardless of which features we used to predict Dogecoin in adaboost, we ended up with the same accuracy. This could mean a couple of things: adaboost and random forest were not really an appropriate model for Dogecoin classification, or that our features were not really relevant in predicting the label for Dogecoin prices. We tried reducing the different window of time for Dogecoin to see if looking at fewer years of data would improve our model, since early on, Dogecoin did not have much variation in its prices. However, even reducing the number of observations did not correct the skewness or our overall results.

TABLE 4: Accuracy						
	Bitcoin			**Dogecoin**		
	Financial Features	Social Media Features	**Combined Features**	Financial Features	Social Media Features	**Combined Features**
Logistic Regression	0.54	0.56	**0.58**	0.81	0.81	0.81
Random Forest	**0.57**	0.49	0.56	0.80	0.78	**0.80**
Adaboost	0.57	**0.59**	0.54	0.81	0.81	0.81
Neural Network	**0.55**	0.53	0.51	**0.81**	0.80	0.73

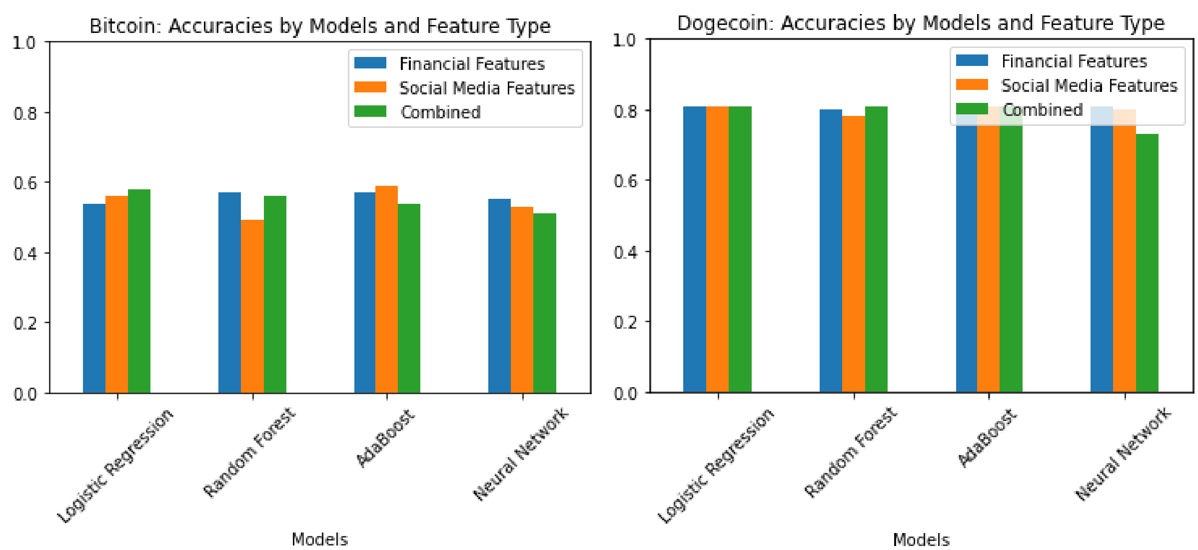


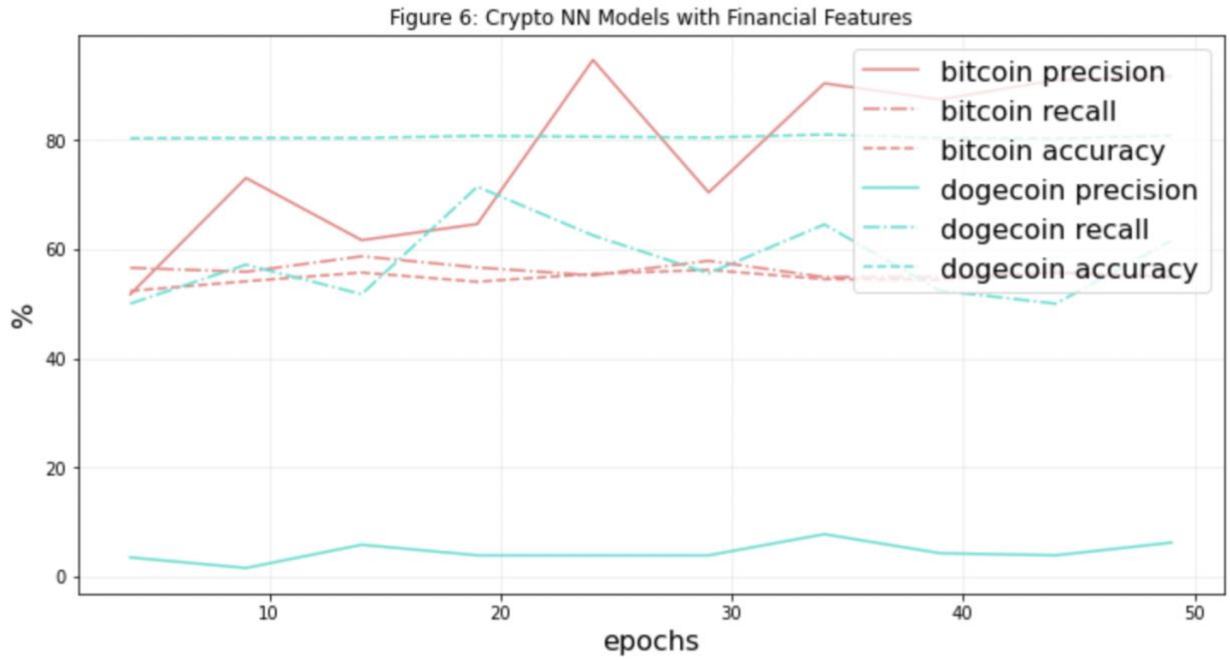
TABLE 5: Precision						
	Bitcoin			**Dogecoin**		
	Financial Features	Social Media Features	**Combined Features**	Financial Features	Social Media Features	**Combined Features**
Logistic Regression	0.58	0.59	0.61	0.33	0	0
Random Forest	0.60	0.53	0.60	0.20	0.2	0.4
Adaboost	0.60	0.60	0.59	n/a	n/a	n/a
Neural Network	0.91	0.95	0.58	0.30	0.06	0.04

TABLE 6: F1						
	Bitcoin			**Dogecoin**		
	Financial Features	Social Media Features	**Combined Features**	Financial Features	Social Media Features	**Combined Features**
Logistic Regression	0.64	0.66	0.66	0.031	0	0
Random Forest	0.63	0.55	0.57	0.63	n/a	0.26
Adaboost	0.66	0.69	0.62	n/a	n/a	n/a
Neural Network	0.69	0.69	0.56	0.07	0.11	0.31

TABLE 7: Top Two Features				
	Random Forest		**Adaboost**	
Bitcoin Features	**Best**	**2nd Best**	**Best**	**2nd Best**
Combined	MACD	Daily_Change_SP500	MACD	replies_count_avg7
Financial	MACD	Daily_Change_SP500	MACD	PROC_10

Social	compound_weighted_avg7	compound_weighted	compound_weighted_avg7	count_weekly_diff
Dogecoin Features	**Best**	**2nd Best**	**Best**	**2nd Best**
Combined	MACD	RSI	MACD	daily_change_perc
Financial	MACD	RSI	MACD	daily_change_perc
Social	goog_trend_score	likes_count_daily_diff	goog_trend_score	replies_count_weekly_diff

Figure 6 illustrates how precision, recall, and accuracy change for our neural network models for both the Bitcoin and Dogecoin datasets when we only include the financial features. Our models built with just the financial features had the highest accuracy levels. In this graph, we can see that the accuracy levels for both Bitcoin and Dogecoins hover around their initial levels. Given the generally weak classification power of our models, we can infer two lessons. First, a feedforward neural network might not be the best neural network to implement for this classification problem. Several papers we found in our literature review implemented some type of a recurrent neural network, which suggests a recurrent neural network would be more suitable for classification changes in cryptocurrency price. Furthermore, as noted with the results in our other machine learning models, the features we selected might not be the best for this classification question. Given that we see improvements in our classification power with a certain set of features, a different set of features with a feedforward neural network could lead to better classification.



5. Conclusion

5.1 Model Success

Given the majority label for Bitcoin was an increase 54% of the time our models performed little better than simply choosing the majority label. We found AdaBoost and Logistic Regression to have the highest accuracy rates at 59% and 58%, which suggests that model complexity does not necessarily yield better results. It also suggests that, for crypto-currency, the features we choose may be more important than the model. Similarly for Dogecoin we found that the models performed roughly the same as simply predicting the majority label or performed slightly worse. The accuracy topped out at 81% using a neural network. Looking at our precision scores for our Dogecoin models, we can see that oftentime the model defaulted to or came close to simply choosing the majority label (in this case no increase). This indicates that, despite having the highest accuracy scores, perhaps Random Forest is a more robust and credible model than Logistic Regression and AdaBoost for our analysis due to its higher precision and F1 scores.

5.2 Financial vs Social Indicators

Across all four models there was not a clear winner between using social or financial features or both. It is clear that given the features we choose, even maximizing our validation accuracy based on different sets of features did not yield huge gains in accuracy. Given this we cannot say anything conclusive about whether classic financial indicators or social signals are more predictive for cryptocurrencies. Other measures, such as precision and F1 were also split between the two types of indicators.

5.3 Policy Implications

We cannot make any policy recommendations based on this finding. It is clear that financial indicators and social indicators we chose were not very predictive for the changes in the two cryptocurrencies prices. From this it's not clear how cryptocurrencies will behave in the future and any attempts to regulate should proceed with caution.

5.4 Takeaways and Future Considerations

One issue we had in training our models was related to our Dogecoin data. Given our Dogecoin accuracies, precision, and F1 scores, we know that the data has many more 0 labels than 1. Perhaps having a more even representation of our labels would yield more accurate and precise results.

Additionally, while we tried many financial features, it's possible that we did not consider the right indicators in predicting the price change. The same could be said for our social indicators. It is certainly possible that Twitter, given its wide range of users, might have been the wrong place to pool social signals. A more niche social network like Reddit might have had more of an impact on the changes in cryptocurrencies. In addition, accounting for regulatory actions or media coverage related to environmental consequences of crypto mining may also have been useful to consider in our model.

It is also possible that we choose the wrong approach using a daily change as our label. Our models might have been more accurate on a longer time range (predicting weekly or 3 day changes) if perhaps the changes are not as sensitive to small changes in sentiment or financial indicators or on the other hand we might have had more success predicting a smaller time range (like hourly changes) if fluctuations in price were very responsive to these indicators. We might have also attempted to predict the magnitude of change instead of the direction.

6. References

6.1 Datasets

Kaggle. (2022). *Ethereum Cryptocurrency Historical Data to 2022.04* [Data set]. https://www.kaggle.com/datasets/psycon/ethusdt-2017-to-2022

Kaggle. (2021). *Bitcoin Historical Data* [Data set]. https://www.kaggle.com/datasets/mczielinski/bitcoin-historical-data

Kaggle. (2021). *Dogecoin Historical Data* [Data set]. https://www.kaggle.com/datasets/tunguz/dogecoin-historical-prices

Kaggle. (2021). *Cryptocurrency Tweets with Sentiment Analysis* [Data set]. https://www.kaggle.com/datasets/fabioturazzi/cryptocurrency-tweets-with-sentiment-analysis

Nasdaq. (2022). *GC:CMX Historical Data* [Data set]. https://www.nasdaq.com/market-activity/commodities/gc:cmx/historical

6.2 Papers

Abraham, J., Higdon, D., Nelson, J., & Ibarra, J. (2018). Cryptocurrency Price Prediction Using Tweet Volume and Sentiment Analysis. *SMU Data Science Review*. Vol 1: No. 3. https://scholar.smu.edu/datasciencereview/vol1/iss3/1/

Bonaparte, Y. (2021). On the Portfolio Choice of Crypto Asset Class: Why the Millennials Own Crypto? https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3829275

Brownlee, James (2019). How to Choose Loss Functions When Training Deep Learning Neural Networks. *Machine Learning Mastery*. https://machinelearningmastery.com/how-to-choose-loss-functions-when-training-deep-learning-neural-networks/

Chen, M., Narwal, N., & Schultz, M. (2017). *Predicting Price Changes in Ethereum*. https://cs229.stanford.edu/proj2017/final-reports/5244039.pdf

Chen, Z., Li, C. & Sun, W. (2020). Bitcoin price prediction using machine learning: An approach to sample dimension engineering. *Journal of Computational and Applied Mathematics*, Vol. 365. https://www.sciencedirect.com/science/article/pii/S037704271930398X

Glenski, M., Weninger, T. & Volkova, S. (2019). Improved Forecasting of Cryptocurrency Price using Social Signals. *Quantitative Finance*. https://arxiv.org/abs/1907.00558v1#

Inder, S. & Sharma, S. (2022). Predicting the Movement of Cryptocurrency "Bitcoin" Using Random Forest. *International Conference on Information Processing*. https://link.springer.com/chapter/10.1007/978-3-030-91244-4_14#Sec6

Jay, J., Kalariya, V., Parmar, P., Tanwar, S., Kumar, N., & Alazab, M. (2022). Stochastic Neural Networks
for Cryptocurrency Price Prediction. *IEEE*. Vol. 8. https://ieeexplore.ieee.org/abstract/document/9079491

Ji, S., Kim, J. & Im, H. (2019). A *Comparative Study of Bitcoin Price Prediction Using Deep Learning. Mathematics*, Vol. 7, No. 10. https://www.mdpi.com/2227-7390/7/10/898/htm

Lamon, C., Nielsen, E., & Redondo, E. (2017). Cryptocurrency Price Prediction Using News and Social Media Sentiment. https://cs229.stanford.edu/proj2017/final-reports/5237280.pdf

Phaladisailoed, T. & Numnonda, T. (2018). Machine Learning Models Comparison for Bitcoin Price Prediction. *2018 10th International Conference on Information Technology and Electrical Engineering (ICITEE)*, 2018, pp. 506-511, https://ieeexplore.ieee.org/abstract/document/8534911

Sattarov, O., Jeon, H., Oh, R., &, Lee, J. (2020). Forecasting Bitcoin Price Fluctuation by Twitter Sentiment Analysis. *Conference: 2020 International Conference on Information Science and Communications Technologies (ICISCT)*. https://www.researchgate.net/publication/349469107_Forecasting_Bitcoin_Price_Fluctuation_by_Twitter_Sentiment_Analysis

Wong, E. L. X. (2021). Prediction of Bitcoin prices using Twitter Data and Natural Language Processing. https://dukespace.lib.duke.edu/dspace/handle/10161/24081

Pano, Toni & Kashef, Rasha. (2020). A complete vader-based sentiment analysis of bitcoin (BTC) tweets during the ERA of COVID-19. Big Data and Cognitive Computing. 4. 33. 10.3390/bdcc4040033. https://www.researchgate.net/publication/346808998_A_complete_vader-based_sentiment_analysis_of_bitcoin_BTC_tweets_during_the_ERA_of_COVID-19

Xu, M. (2021). NLP for Stock Market Prediction with Reddit Data. https://api.semanticscholar.org/CorpusID:23535782

Group contributions:

David–features and feature analysis
Ian–Neural Network
Irsa–Random Forest
Kelley–AdaBoost
Yunjoo–Logistic Regression
We all equally contributed towards the literature review, midpoint reports, and progress reports.