Machine Learning for Public Policy

Final Project: Cryptocurrency Prices

Analyzing the impact of social and financial indicators on cryptocurrency prices

Team Name: The Great Cappsby

Members: Irsa Ashraf, David Hackett, Ian Jeong, Yunjoo Park, Kelley Sarussi

Background

- The value of cryptocurrencies has skyrocketed over the past decade
- While many investors are including cryptocurrencies in their portfolios many argue they represent a economic bubble or even worse a pyramid scheme
- Do cryptocurrencies perform like classic assets/stocks or is their value determined by social signals and hype?
- Our research question, can we better predict the changes in cryptocurrency value using classic economic indicators used for forecasting stocks or social indicators or some combination of the two.
- Stick with a simpler prediction problem, classification. Can we predict whether the price of a cryptocurrency will increase or not on a given day.
- Policy implications: how should we regulate cryptocurrencies? Should they be regulated like stocks and other assets or as something more predatory?

Figure 1: Bitcoin Price Over Time

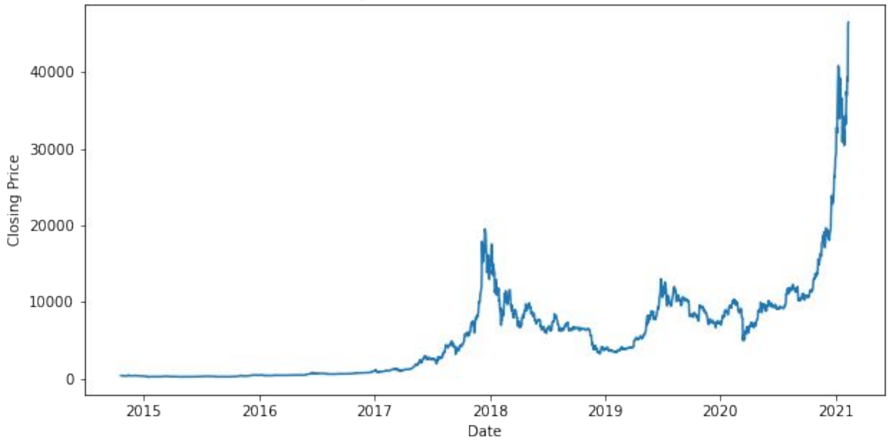


Figure 2: Bitcoin Changes in Value

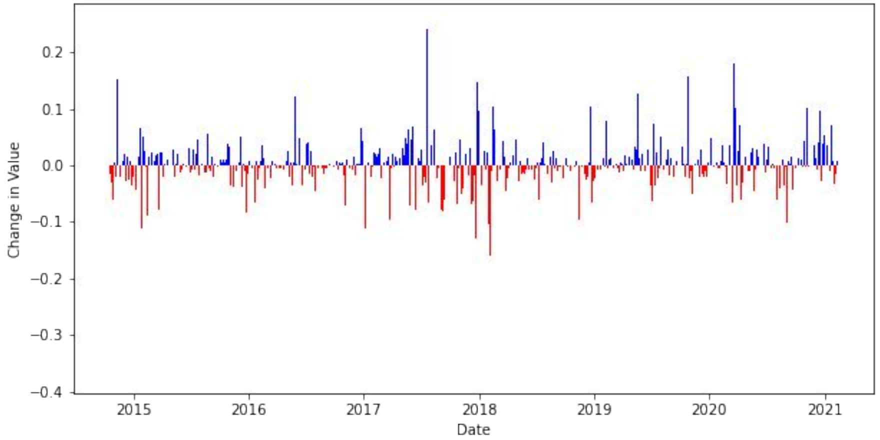


Figure 3: Dogecoin Price Over Time

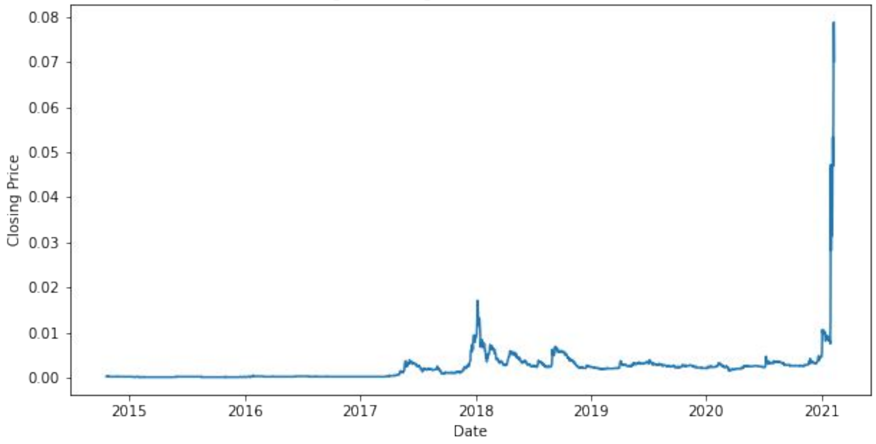
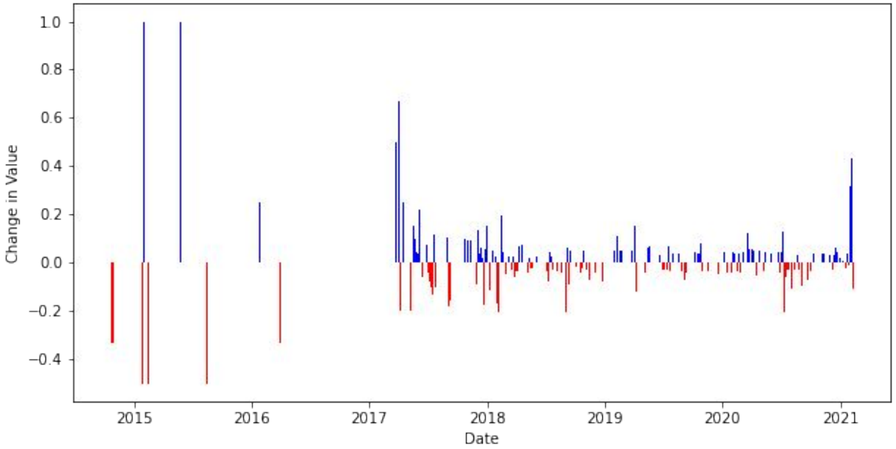


Figure 4: Dogecoin Changes in Value

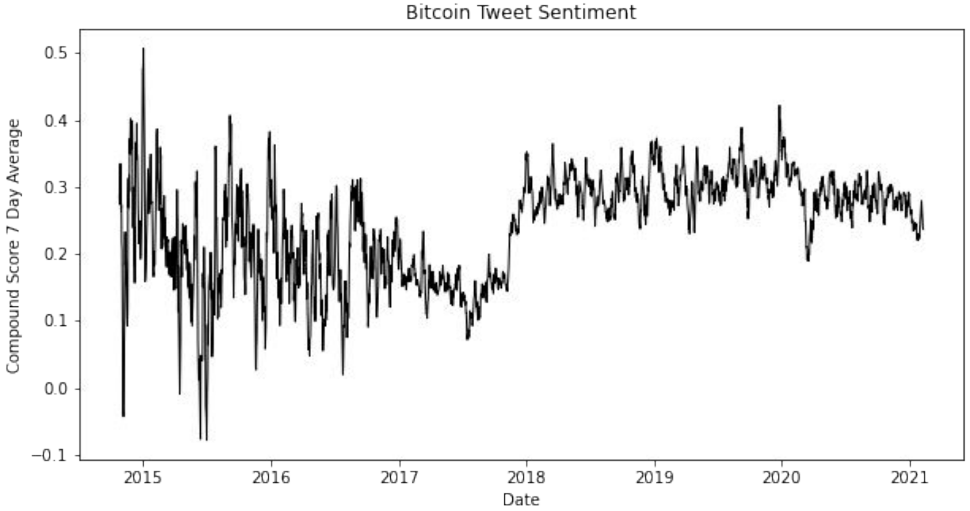


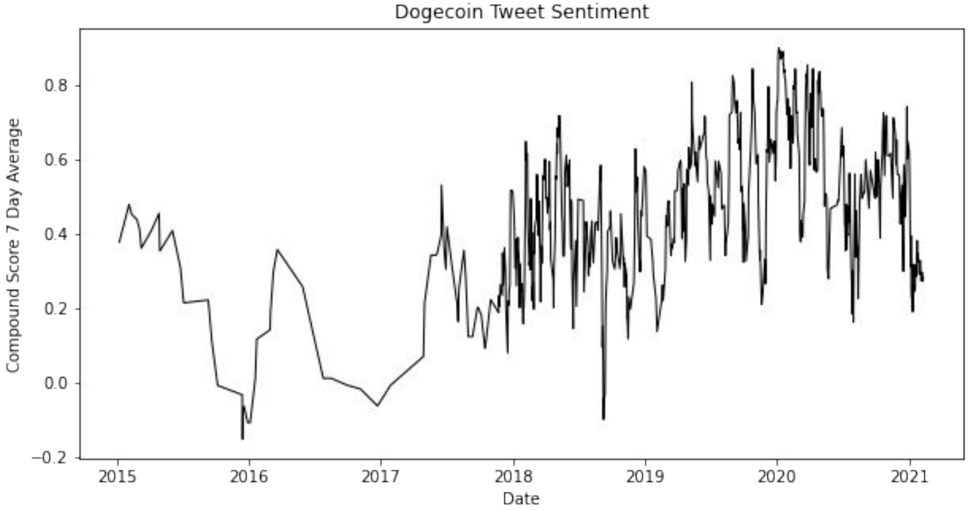
Data: Financial Indicators

- From our cryptocurrency datasets we generated some simple features like daily change, daily change percent, and weekly change
- From those datasets we also created more complicated financial indicators that are used in stock prediction: Relative Strength Index, Williams Percentage Rating, Moving Average Convergence Divergence, Price Rate of Change.
- Additionally we looked at the SP500 and a classic asset like Gold. From these we generated similar features to use as possible indicators.

Data: Social Indicators

- Pulled from a tweet dataset on Kaggle
- Dataset included sentiment scores already, using VADER, including a positive, negative, neutral, and compound score.
- We pulled out the tweets related to each cryptocurrency and aggregated them to generate features such as daily counts, daily compound scores, rolling averages of sentiment scores etc.
- Additionally we pulled daily google trend scores for the time period and used those as another social indicator.


Bitcoin Tweet Sentiment
Compound Score 7 Day Average
0.5
0.4
0.3
0.2
0.1
0.0
−0.1
2015
2016
2017
2018
2019
2020
2021
Date

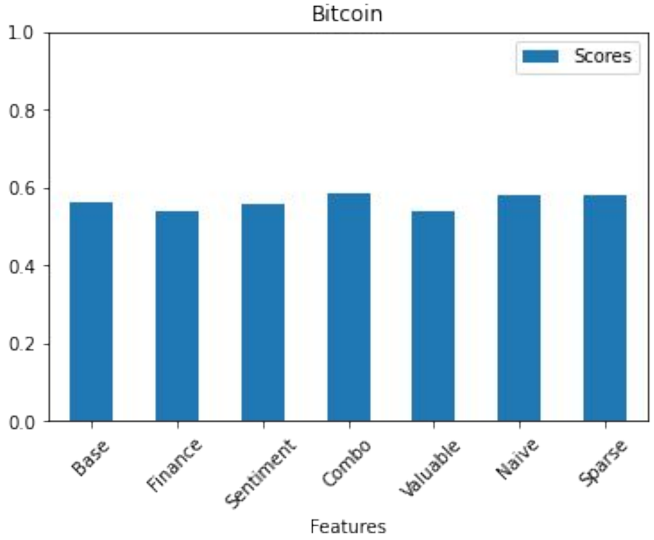

Dogecoin Tweet Sentiment
Compound Score 7 Day Average
0.8
0.6
0.4
0.2
0.0
−0.2
2015
2016
2017
2018
2019
2020
2021
Date

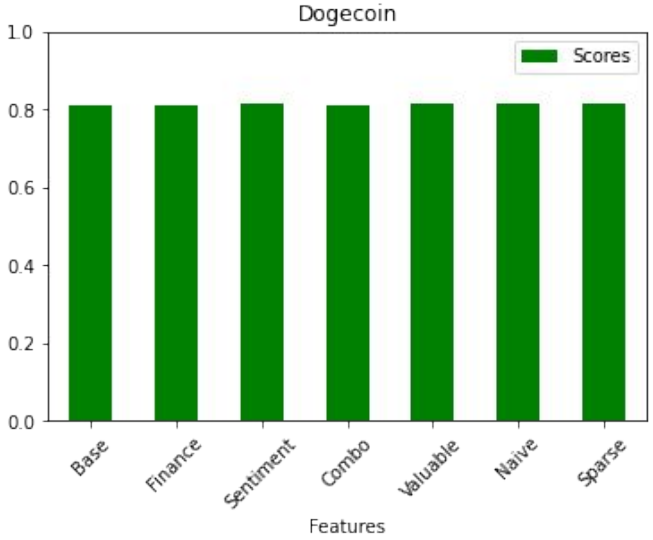
Experimental Design

- We opted to focus on two cryptocurrencies, bitcoin and dogecoin, because they represent a more popular crypto that is widely traded and a more volatile “Meme” coin.
- Models: random forest, adaboost, logistic regression, neural network.
- Same train and test data for each: 80/20 split. The training data could be further split for validation purposes.
- For each model, tested three sets of features: financial features, social features, and a combination of the two.
- Labeled data with either a 1 or 0
  - 1 if data increased from opening price
  - 0 otherwise
- Bitcoin increased 54% of the time, while dogecoin decreased 81% of the time

Models: Logistic Regression

- Ran model with features from financial, sentiment, and both categories
- Higher accuracy for dogecoin, but dogecoin slightly feature agnostic
- Highest accuracies achieved:
  - Bitcoin: combo, 0.583
  - Dogecoin: 0.814





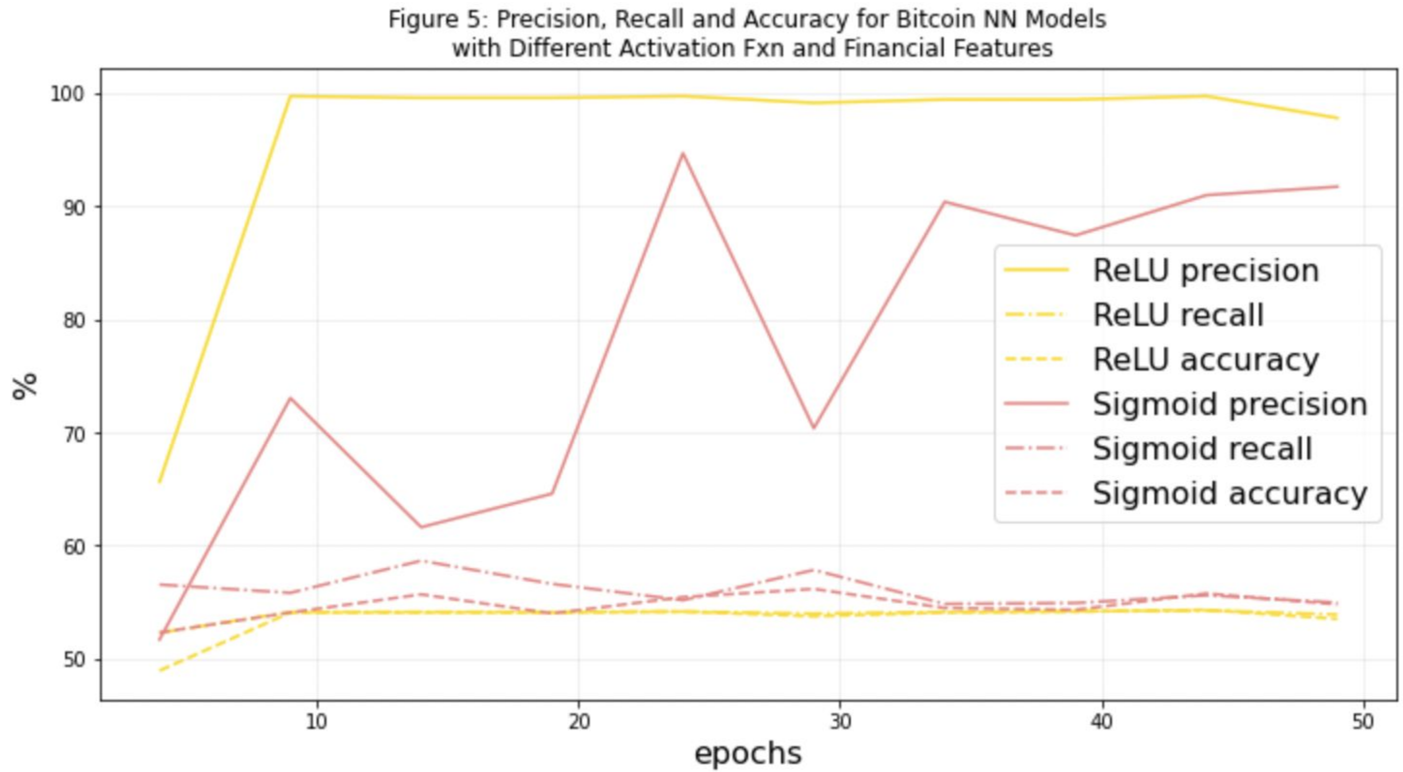
Models: Adaboost and Random Forest

- Ran the model using decision tree classifier with tree depth = 1,
    - Explored making the models less weak (tree depths = 2)
- Hyperparameters:
    - Optimized using grid search
    - Learning rate: tried 0.01 and 0.1
    - Number of estimators: tried 100, 500, 1000
- Evaluated the model using repeated stratified k-fold cross-validations, with three repeats and 10 folds
- For Random Forest found that 80 estimators did the best (using validation data)
- Used gini index, no max features, and square root of n for number of features per tree

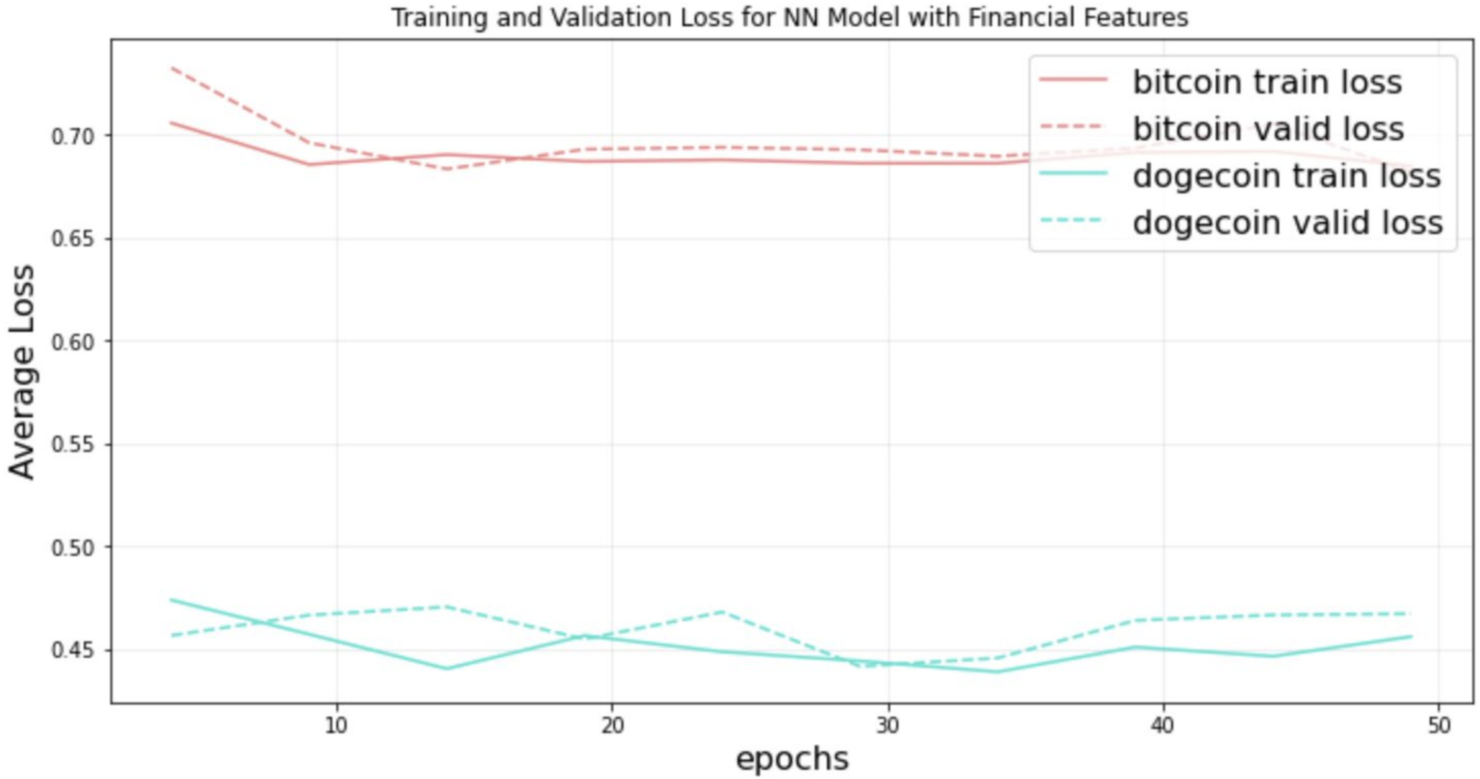
Models: Building a Neural Network and Tuning Hyperparameters

- Built a NN using PyTorch
- With the objective of minimizing the validation loss, used Optuna to identify optimal settings for:
  - number of layers
  - learning rate
  - Optimizer (SGD, Adam, RMSProp)
  - Dropout rate for each layer
- Compared training and validation losses between a model with sigmoid as its activation function and one with ReLU as its activation function in the models' hidden layers

Assessing fit of ReLU vs Sigmoid in our NN model



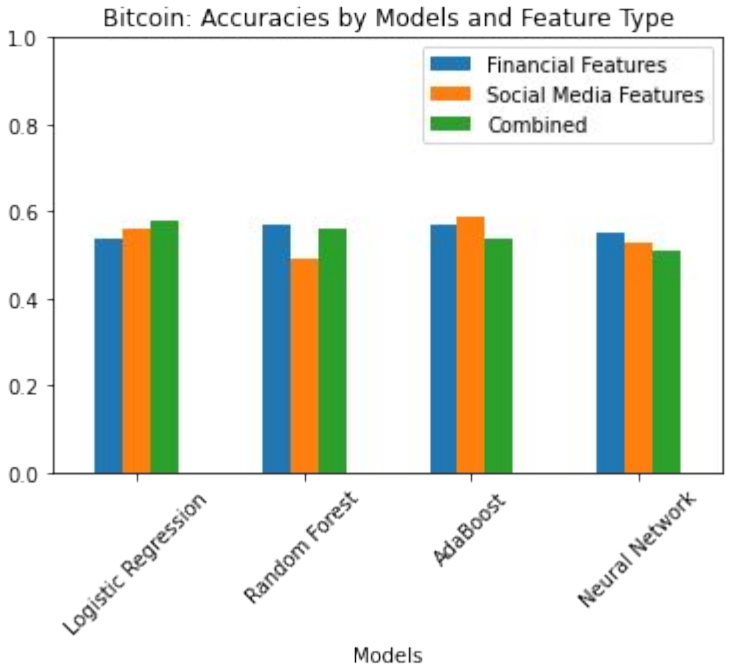
NN Model's "Learning" Through the Iterations

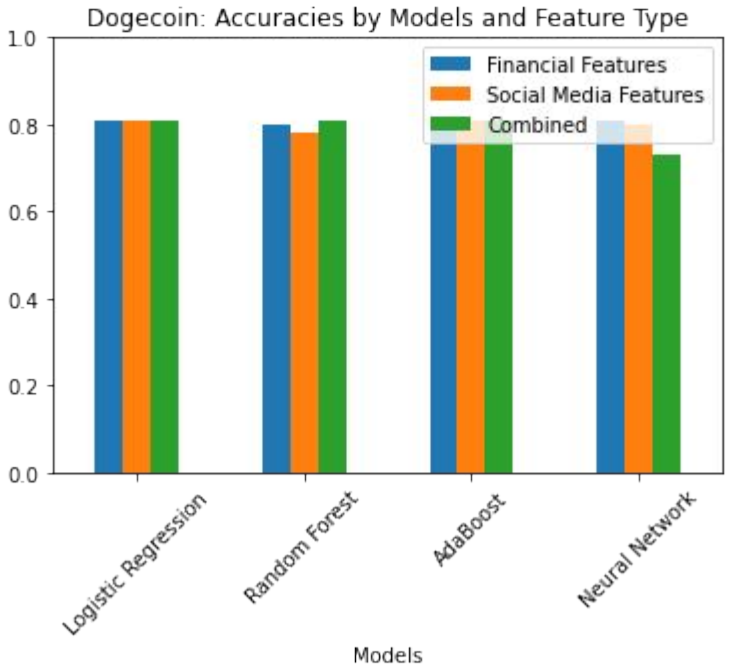


Results: Accuracy

Accuracy Rates by Features Type and Model						
	Bitcoin			**Dogecoin**		
	Financial	Social Media	**Combined**	Financial	Social Media	**Combined**
Logistic Regression	54%	56%	**58%**	81%	81%	81%
Random Forest	**57%**	49%	56%	80%	78%	**81%**
AdaBoost	57%	**59%**	54%	81%	81%	81%
Neural Network	**55%**	53%	51%	**81%**	80%	73%

Results


Bitcoin: Accuracies by Models and Feature Type
1.0
0.8
0.6
0.4
0.2
0.0
Financial Features
Social Media Features
Combined
Logistic Regression
Random Forest
AdaBoost
Neural Network
Models


Dogecoin: Accuracies by Models and Feature Type
1.0
0.8
0.6
0.4
0.2
0.0
Financial Features
Social Media Features
Combined
Logistic Regression
Random Forest
AdaBoost
Neural Network
Models

Results: Precision

Precision						
	Bitcoin			**Dogecoin**		
	Financial	Social Media	**Combined**	Financial	Social Media	**Combined**
Logistic Regression	58%	59%	61%	33%	n/a	0%
Random Forest	60%	53%	60%	20%	20%	40%
AdaBoost	60%	60%	59%	n/a	n/a	n/a
Neural Network	91%	95%	58%	30%	6%	4%

Results: F1

F1						
	Bitcoin			Dogecoin		
	Financial Features	Social Media Features	**Combined Features**	Financial Features	Social Media Features	**Combined Features**
Logistic Regression	64%	66%	66%	3.1%	n/a	0%
Random Forest	63%	55%	57%	63%	n/a	26%
Adaboost	66%	69%	62%	n/a	n/a	n/a
Neural Network	69%	69%	56%	7%	11%	31%

Conclusion & Next Steps

- Our classification accuracy was only slightly better than or the same as predicting the majority label
  - Different feature categories did better in different models
  - Given that bitcoin increased 54% of the time and dogecoin decreased 81% of the time the models did not perform particularly well
  - AdaBoost and Logistic Regression performed the best *
  - Hard to gather any policy implications from this other than crypto is still a mystery
- How might we improve?
  - Classification may not be suited for this topic
  - Garbage in: our features may not have been effective
  - Look at weekly increases or hourly increases? Change the time range
  - Cryptocurrency might just be too hard to predict