Bitcoin

Opportunity

Bitcoin is a highly volatile and unpredictable financial instrument that poses both great promise and risk for investors. Some reputable investors claim Bitcoin is the future of financial currency and will grow to be worth $1M in 5 years (from its current ~$19k value), while others believe it will go to zero.[1][2]

Problem

Bitcoin incurs massive pricing swings where it gains or loses 10%+ percent in a single day. Bitcoin's volatility has led to investor greediness when Bitcoin's price is on the rise and fear when the price is falling. Given the day-to-day swings and strong impact they have, the focal problem is determining whether Bitcoin's price will increase or decrease the next day.

Hypothesis

Our objective is to create a model which predicts tomorrow’s price of bitcoin. Going into this project, our initial hypothesis is that the price of Bitcoin is correlated with financial data and public interest metrics and as such, can be predicted using machine learning methodologies with moderate accuracy.

Approach

Our team aims to analyze a wide range of data to predict the future price of Bitcoin by following these steps:

1. Define potentially predictive items and find relevant data sets (e.g. the price of other cryptocurrencies, S&P 500, gold)
2. Organize, input and clean the datasets into a universal dataset with lots of features that is ready for analysis

[1] https://markets.businessinsider.com/currencies/news/bitcoin-hit-million-five-years-ex-goldman-hedge-fund-boss-2020-10-1029682590#:~:text=to%20BI%20Prime-,Bitcoin%20will%20surge%20to%20%241%20million%20in%205%20years%20by,Sachs%20hedge%2Dfund%20chief%20says&text=The%20price%20of%20bitcoin%20could,said%20in%20a%20recent%20interview.

[2] https://www.cnbc.com/2019/01/23/bitcoin-price-going-to-zero-davos-future-of-blockchain-tech-.html

3. Create correlation matrices to identify collinear features to remove
    a. Visualize the data with scatterplots
    b. Remove correlated features
4. Create a linear regression to express the relationship between the price of Bitcoin and the remaining feature set
    a. Iterate as needed by eliminating features one by one that has too high of a p-value
5. Add back all features and create a decision tree to determine how multiple features impact the price of Bitcoin.
    a. Testing and training sets must be created to test the accuracy of the model.
    b. Tune the size of the decision tree to avoid overfitting
6. Create a random forest for a more robust analysis than a single tree model
7. Create a K-NN and use the training set to classify the predicted binary High / Low price of Bitcoin
8. Create linear, polynomial, an radial support vector machines for further analysis

Data Collections

Cryptocurrency

The first place we looked for Bitcoin data sets was Kaggle. but they were all 2+ years old.[3] We wanted to find the most up to date datasets given Bitcoin's recent price surge and overall volatility so we continued our search. We found a website, Crypto Data Download, with free up to the date data on Bitcoin and prominent altcoins that all had a minimum of 1492 data points.[4]

Common Financial Instruments & Google Search

For non-cryptocurrency financial data, we navigated to Yahoo Finance and downloaded historical price data for Gold, the S&P 500, and the 30 year US treasury bond.[5] Lastly, for our analysis of public interest in Bitcoin, we downloaded Google Trend data on Bitcoin's search volume over the past five years.[6]

[3]

[4] https://www.cryptodatadownload.com/data/

[5] https://ca.finance.yahoo.com/

[6] https://trends.google.com/trends/explore?date=today%205-y&geo=US&q=bitcoin

Data Limitations

Unfortunately, Google's volume data is only available to download in weeks while the rest of our data was in days. We opted to create copies of the weekly data points for each day of the week that they cover. Furthermore, while Bitcoin's is available to trade 24/7 (as Forex markets are), financial markets are only open on weekdays. So, we opted to delete all weekend dates from the dataset

Improvement Opportunities

To improve the data collections, we could've adjusted for inflation by using the following formula:
Real Dollars (base year $) = Nominal dollars (current year $) * Index (base year) / Index (current year)

As well, to improve the data collection, we could include a diverse range of asset classes (i.e. AAA US Bond Prices)

Our universal dataset only had 952 data points which is a very small amount for a predictive model. We did the best with the data we could find but ultimately, cryptocurrency is a very new field and Bitcoin has only been in existence for 12 years. We opted to include other cryptocurrencies with shorter life spans than Bitcoin which decreased data points.

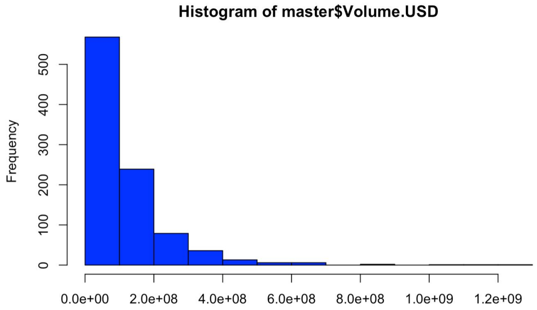
Ultimately, it'll be difficult to build a strong model given the small data set, randomness of the data, and relatively simple machine learning models.

Data Visualization & Feature Removal

Data visualization provides an overview of numerical or text data in a way that is easy to interpret many data points quickly. Data visualization is valuable for providing quick feedback on a dataset and for identifying outliers. We utilized two data visualization techniques, histograms and scatter plots.

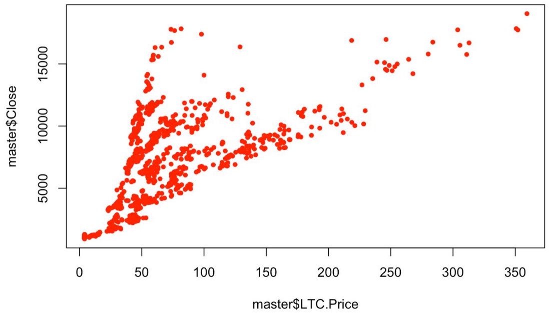
Histogram

We created a series of histograms to analyze Bitcoin's main features of Volume.BTC, Volume.USD, and Price. The steep bar on the left represents a consistent low level of Bitcoin traded with some extremely high-volume days as shown by the long-tail to the right. The price variable for Bitcoin is more evenly distributed with peak frequency hovering just below the $10k range and a rightward tail approaching $20k.



Scatter Plots

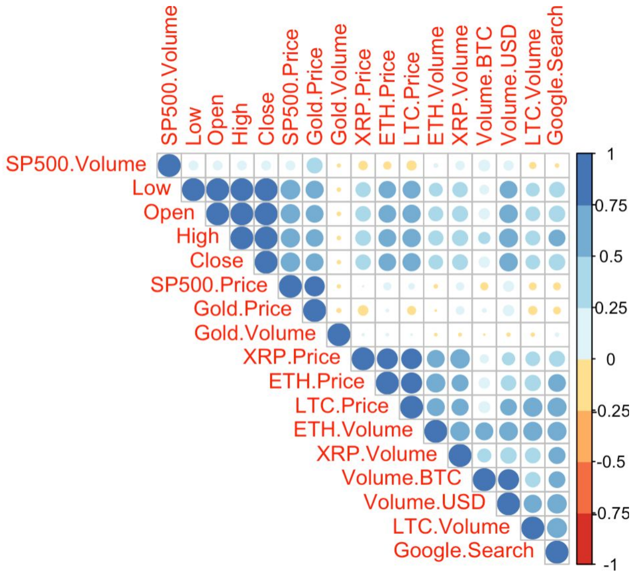
We created a scatter plot to visualize the price of Bitcoin relative to the date over the date range. Secondly, we created scatter plots between each independent variable and the price of Bitcoin to display if there was any linear correlation. Most x,y scatter plots appeared mostly randomly distributed but Ethereum Price, Litecoin Price, and the S&P 500 had a resemblance of a linear relationship. For example, the Litecoin price plot is attached below.



Correlation Matrix

A correlation matrix shows the correlation between independent variables for the purpose of removal to avoid multicollinearity. Simply, highly correlated independent variables should be removed because they can skew the model’s output. Our matrices showed that there is a high correlation between many of the independent variables, with some of them almost being entirely correlated. This is to be expected given similarity in values such as high, low, open and close, and between the performance of assets.

We used our correlation value cutoff of 0.76 to retain a significant enough sum of features for the linear regressions.



Models Used

Logistics Regression

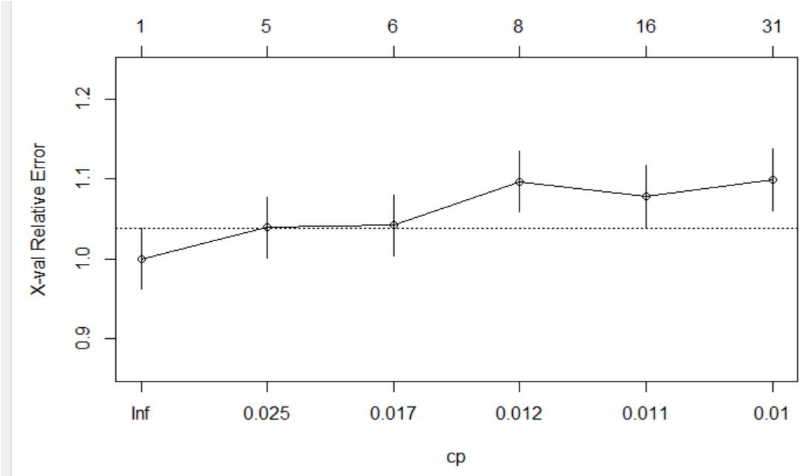
Logistic regression is a model for binary classification and a form of non-linear regression. Since we are predicting whether the next day's bitcoin price will be high or lower than today, logistic regression is favorable in comparison to linear regression. After removing highly correlated features in our data set, we ran a logistic regression. There were multiple features with high P values. According to Investopedia, "A p-value is a measure of the probability that an observed difference could have occurred just by random chance. The lower the p-value, the greater the statistical significance of the observed difference." We first removed all variables with a p-value above 0.5. This left 5 variables in V1 of this analysis: **"low", "eth.volume", "xrp.volume", "google search", and "SP500.price".**

However, the more appropriate p-value threshold would be 0.05. By running V2 of this analysis with only variables with a p-value under 0.05, only "**low" and "xrp.volume"** were included. This model demonstrates that very few variables are accurate predictors of bitcoin price. It makes sense that "low" (i.e. the lowest price of bitcoin in the day) would be a good predictor of tomorrow's price, but it was interesting to see "xrp.volume" as the other variable. (i.e. the price of Ripple). Since Bitcoin is by far the largest cryptocurrency, you would expect the price of other cryptocurrencies such as Ripple to follow the price of Bitcoin, not the other way around.

V1 had a misclassification rate of 0.46, and V2 had a misclassification rate of 0.44.

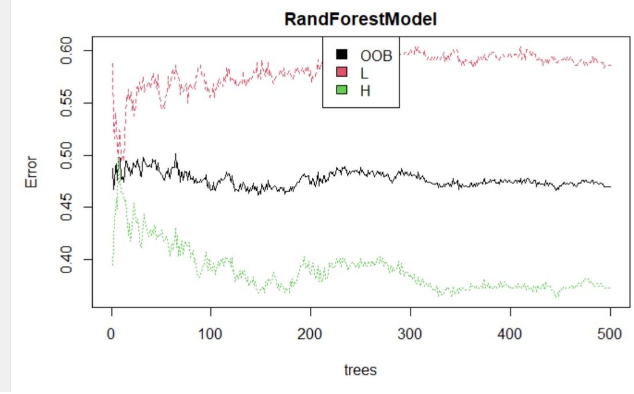
Decision Tree

Decisions tree models work by marking a recursive sequence of decisions that lead to an outcome. In pruning the tree to avoid overfitting, we plotted the relative error as a function of tree size, which resulted in the graph below. This error graph is very strange, as you would typically expect to see a downward sloping line (i.e. error decreases as the size of the tree increases). This was a clear sign that the decision tree might not yield a great model to predict bitcoin prices. By pruning the tree at cp=0.018, we ended up with a misclassification rate of 0.54.



Random Forest

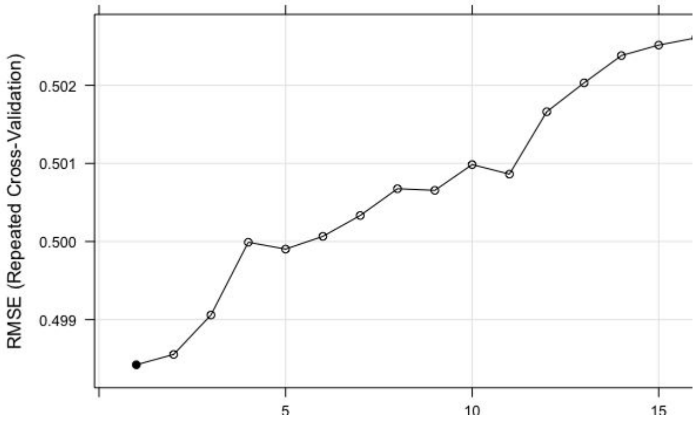
Random forest is a model in which multiple decision trees are made, each with one vote for the predicted value. The final predicted value is the one which is predicted by the most trees. The graph which displays the error rate as the number of trees used to fit the model increases looks very strange. As you can see below, the overall error (black line) barely changes as the number of trees increase, and the number of “L” misclassification increases as the number of trees increases. This is strange since usually, error tends to decrease as the number of trees increases. As with the decision tree model, this was a sign that the random forest model might not yield a great model to predict bitcoin prices. This was confirmed after running the predictions, which yielded a misclassification rate of 48%.



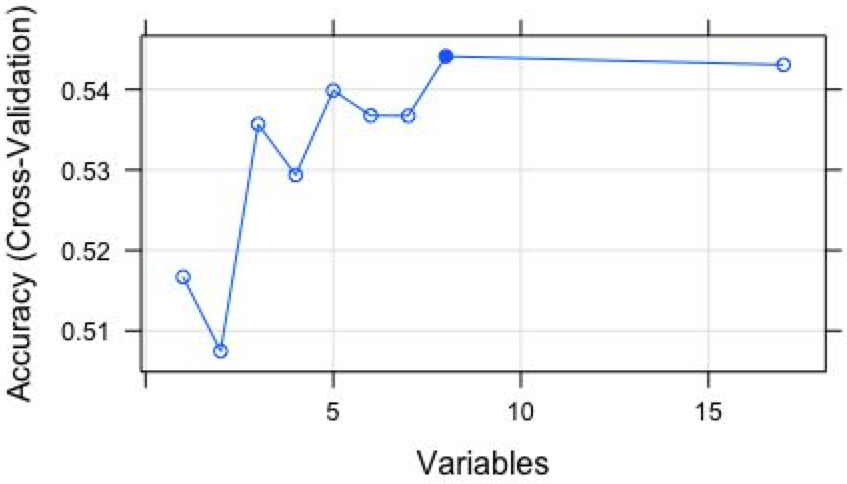
Recursive Feature Elimination

Recursive Feature Elimination was used to understand which specific features are more significant to the models and offer an efficient approach for eliminating features from a training dataset. When running the recursive, all variables were eliminated except “Low”. This is quite interesting as “Low” was one of the final variables in the final regression analysis due to eliminated variables due to P values and multicollinearity. Potentially this indicated “Low” is a strong variable to dictate Bitcoin prices.

Recursive Feature Elimination (LM)



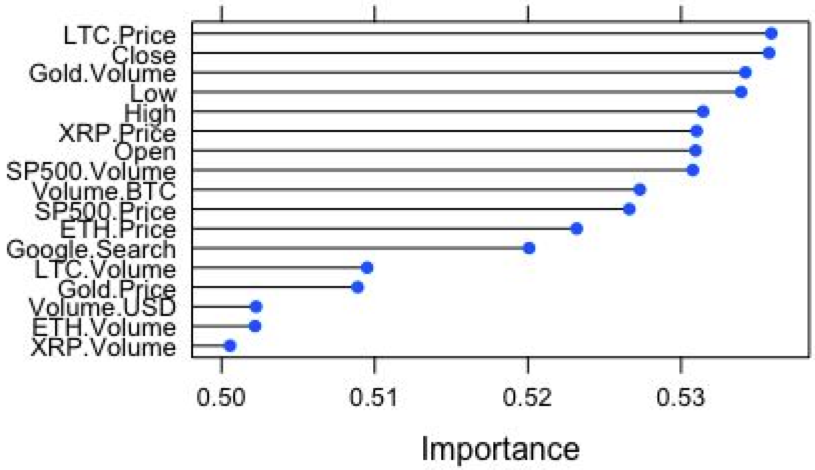
RFE- Random Forest



Features Analysis

Features analysis was used to further understand the features used within the dataframe. We first ranked the variables based on Importance.

It is interesting to see LTC.Price and Close the highest rating feature, however, the deviations between variables are quite low 50 -53. This potentially indicates collinearity between variables.



K-Nearest Neighbors

K-NN is a model used to label data when you have some data which has already been labeled (training set), and other similar data which has not (testing set). The intuition behind KNN is to categorize a piece of unlabeled data based on what the majority of surrounding (i.e. most similar data) is categorized as. KNN works with any type of distribution, and works well with datasets which have numerical features. The results of our KNN model was a 51% misclassification rate without removed variables, 56% with x variables ranked by importance, and 44% with x variables based on P values of regression and correlation matrix.

Support Vector Machine

Support vector machine is also a model used for binary classification. It is great for beginners, and has been very popular. Compared to logistic regression, SVMs are based on geometric intuition, where the best line which separates the 2 classes of data is found. A linear SVM gave a misclassification rate of 0.49, a polynomial SVM gave a misclassification rate of 0.47, and a radial SVM gave a misclassification rate of 0.52. Clearly, SVMs were not at all successful at predicting the future price of bitcoin. They were about the same as a coin toss.

Times Series

What is Time Series

We realized that one concept which could improve the predictive power of our model is time series. Though time series was not explored in class, we wanted to look into it since time series is an important concept which applies to forecasting when it involves a time component. Time series is defined as an ordered sequence of values of a variable at equally spaced time intervals. The main difference between previous models explored in our analysis and time series is that time series analysis takes into account the fact that data points have taken place over time, which may result in an internal structure (e.g. autocorrelation, seasonal variation, trend) which should be accounted for.

[7]

Time Series Components

There are 4 aspects of behaviour that make up time series[8]

1. Level: what the baseline values would be for the series if it was a straight line
2. Trend: the overall long term direction of the series, often linear
3. Seasonality: repeating cycles and patterns of behavior over time
4. Unexpected variation/ noise: variability in observations that cannot be explained by the model

All time series have a level. Trend, seasonality, and noise are optional. By combining these 4 components together in various ways, models can be made to forecast the price of bitcoin with varying levels of accuracy.

Applying Time Series to Bitcoin

Before applying forecasting models, we studied the trends and seasonality of the price of bitcoin. Understanding how these patterns apply to Bitcoin will allow us to choose a better forecasting model. Looking at the seasonality output, there is a clear uniform seasonal variation in the price, which is a trend that has been very consistent over the years. The trend output shows that there was very little movement up until 2016/17, at which point it increased sharply. Since then, it has fluctuated up and down. There are many time series models that are

[7] https://machinelearningmastery.com/time-series-forecasting/#:~:text=Forecasting%20involves%20taking%20models%20fit,them%20to%20predict%20future%20observations.&text=The%20skill%20of%20a%20time,performance%20at%20predicting%20the%20future.

[8] https://www.itl.nist.gov/div898/handbook/pmc/section4/pmc43.htm

available. We chose to use the double exponential smoothing method (Holte's forecasting model), which uses exponential smoothing based on alpha and beta values. Compared to single smoothing, double exponential smoothing is better at following data where a trend exists. Holt's does not account for seasonality. [9]

For a 9 day prediction, the results were an RMSE of 5,847. This is very poor but is consistent with previous poor performance of models, and makes sense given the volatility of the stock.
If we were to expand the scope of this analysis, we first loop the model in order to generate an average RMSE. In the current model, the test set is only run once. Furthermore, there are many other interesting time series models to explore. For example, Bayesian regression has been used by Harvard professionals to determine patterns in bitcoin data, and has far superior predictive ability compared to other time series models[10].

Sentiment Analysis

The team originally wanted to conduct sentiment analysis to understand the fluctuations of Bitcoin Price. Sentiment analysis will provide a score to reflect where investors on social media exhibit bearish or bullish emotions.

We wanted to look into the following area:

- Social Media Language (Tweets over the years will indicate an emotion)
- News Articles (The frequency of key words used in relation to Bitcoin Prices)
- Google Search Data

News Articles and Social Media

We sourced our news data from Kaggle. The data file includes article titles over the years from ABC News. (Over 20,000 rows of data)

- We then parsed every word in the title and removed stop words (ands, is, the, etc.).
- We then found the top 100 words used throughout the years. This will be used as the features of dataframe

[9] https://www.rdocumentation.org/packages/aTSA/versions/3.1.2/topics/Holt?fbclid=IwAR37YO2I2le5HZaWXggVmW7w4icM29ITOJVcOaH2lkln-_O_bnMsX6_QcFY

[10] https://www.kaggle.com/ara0303/forecasting-of-bitcoin-prices?fbclid=IwAR0uXs9IH-jzHWndtdQ6kC9nDoeZupfc1Z0puGZN1nMgy_AYTg8EosWaoj8

- A correlation matrix was created between the words to remove any multicollinearity
- A counter was used to indicate the frequency of the word on a specific date
    - Ex. Cryptocurrency was used 50 times on Feb 10

This is where we realized our computers did not have enough computing power to work through the data set. We worked around by decreasing the data set, however still ran into issues Thus, news data was eliminated as a potential model. We believed would have the same issues when looking into Social Media, and thus was eliminated

We researched more and found an API that has done a similar project. To improve this model, we can leverage a preexisting API and decrease the number of top words.[11]

Google Search Data

As discussed above, Google search data was included in the data frame for the models. However, we look into the frequency/interest over time. To improve this data, we should do a sentiment analysis on specific google search results to predict bullish or bearish trends.

Summary Chart

Model	**Misclassification Rate**
Regression Variables with P Value < 0.5	46%
Regression Variables with P Value < 0.05	44%
Decision Tree	54%
Linear SVM	49%
Random Forest	48%
KNN (Without Removed Variables)	51%
KNN (X Variables Ranked by Importance)	56%

[11] https://federicoriveroll.medium.com/predicting-bitcoin-with-news-using-r-bb1cfdf195c5

KNN (X Variables Based on P value of Regression and Correlation Matrix)	44%
Polynomial SVM	47%
Radial SVM	52%
Time Series	RMSE of 5,847 for a 9 day prediction

Conclusion

We compiled strong analysis with the models taught in class but ultimately came up short on our goal of building a predictive model. Our top misclassification rate was 44% which we achieved with both a linear regression (p value < 0.05) and KNN (*x* variables based on P value of regression and correlation matrix). The results from our time series analysis were equally poor given the high RMSE of 5,847 for a 9 day prediction. Since we were predicting a binary variable (H or L), the worst case misclassification rate would be 50%, at which point a model would perform with the same accuracy as a coin flip. Evidently, most models performed very poorly by this standard.

We did not build a neural network model which may have been stronger than the models we tried. But, interestingly, our models' predictive accuracy did not increase with the sophistication of the model. This decreases our confidence that a more advanced machine learning model is the solution to an inherently difficult problem of predicting a radically volatile asset. However, the one area where we are confident that a more sophisticated model will garner better predictions is sentiment analysis. Unfortunately, we were unable to build a sentiment analysis given their complexity and vast computing power requirements. But, we were able to learn from existing models and keep this option in the back of our minds for future learning.