

Cryptocurrency Recommender

Final Report

**Sean Hung, Yu-Kai Wang
Matt Uryga, Jonathan Li**

Rensselaer Polytechnic Institute
CSCI-4964
05/01/2022

1 Introduction

1.1 Overall Goals

The primary goal of the project was to construct a program to analyze cryptocurrencies and use graph-based algorithms to identify potential opportunities for profit. The initial intention was to scrape real-time data from Reddit, Twitter, and other social media platforms, and combine this data with quantitative analysis of price movements and similarities between cryptocurrencies.

1.2 Relevancy to Graph Mining

The use of graph-based algorithms was a foundational idea for the overall success of the project. On a high level, a graph would be constructed containing various information about different cryptocurrencies, and an algorithm such as triadic closure or Pagerank would be run on the graph to predict similar cryptocurrencies.

2 Background Information

2.1 Cryptocurrencies

The blockchain technology has recently exploded in popularity thanks to the sudden spike in the value of Bitcoin. As the price of Bitcoin skyrocketed, various other smaller cryptocurrency projects (often called altcoins) with lots of potential emerged. These cryptocurrencies each have a unique functionality and usage, but cryptocurrencies with similar properties tend to follow each other in terms of price. This is why, in this project, an attempt is made to predict these groups of cryptocurrencies by using different clustering metrics, graph-based neural networks, and many other algorithms.

2.2 Cryptocurrency Market

Cryptocurrencies can be traded much like stocks or any other commodity. There is an entire market that exists solely for facilitating the trade of cryptocurrencies, and just like in the stock market, there is the potential to make or lose money.

The cryptocurrency market is notoriously volatile and it is not uncommon to see price swings of 20% or more. As such, trading cryptocurrencies is typically seen as more high risk, high reward than trading stocks or precious metals.

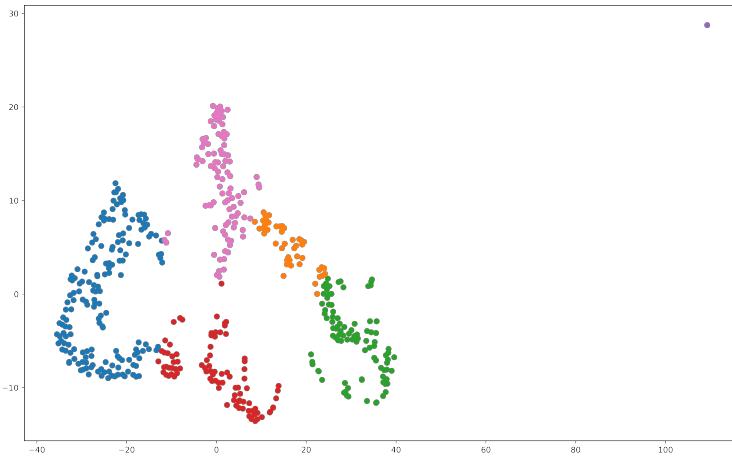
3 Algorithms

3.1 Clustering

Using the CoinMarketCap API, a variety of data was scraped for many cryptocurrencies. Among this data was market cap, fully diluted market cap, total supply, and circulating supply. These attributes were run through a t-Stochastic Neighbor Embeddings (t-SNE) algorithm to reduce the dimensionality of the data and produce a better visualization and more intuitive clusters with the following clustering algorithms.

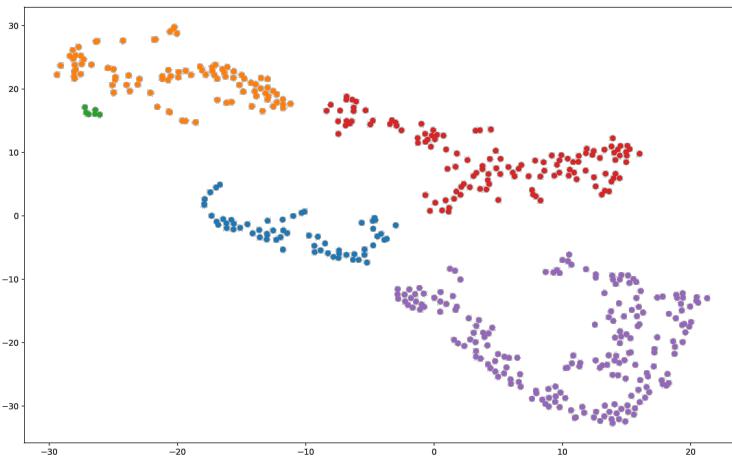
3.1.1 K-Means

K-means clustering was initially used to cluster the embeddings. When the clusters were visualized (visualization is below), it became clear that the data had a significant amount of variation and thus k-means was producing outliers.



3.1.2 DBSCAN

To eliminate any possible outliers in the clustering, the embeddings were also fed into a DBSCAN clustering algorithm. DBSCAN clustering is density based, and as such, it is extremely effective at grouping data points while leaving no outliers. The visualization of the DBSCAN clustered data is below.

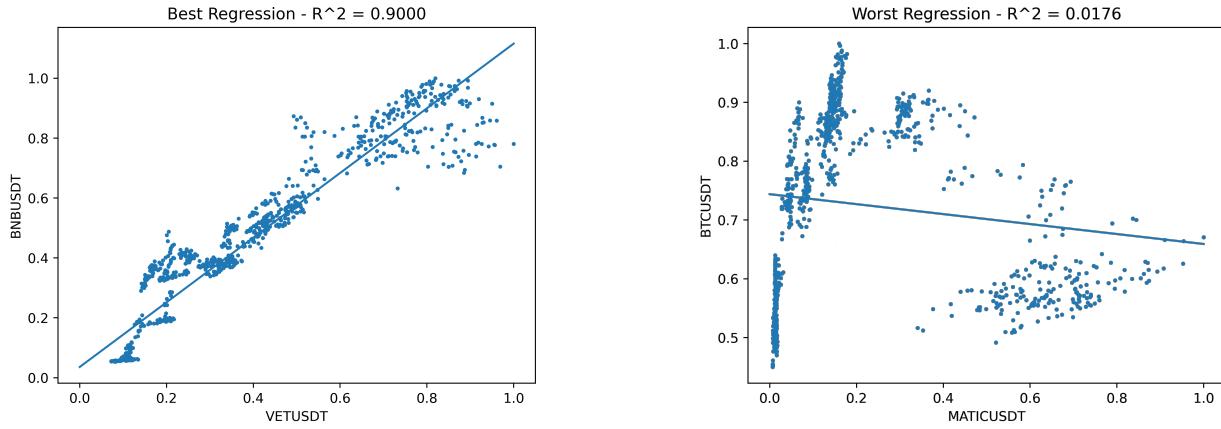


3.2 Regression Similarity

For every pair of cryptocurrencies, a simple regression was computed. First, the prices of each cryptocurrency were normalized from 0 to 1. This was done to standardize the price axis to a common scale - some cryptocurrencies have a market value in the tens of thousands, while some others are worth hundredths of a cent.

Next, a linear regression was run, and the r^2 was calculated. Overall, this produces an r^2 for each pair of cryptocurrencies that represents how similar their price histories are.

Below are some examples of the linear correlations found:



3.3 Correlation Prediction

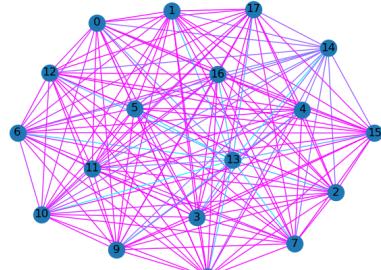
The goal of correlation prediction is to predict future correlation interaction between different pairs of cryptocurrencies, by looking at the past correlation data. Instead of using a traditional linear-regression / polynomial-fit SVM to make predictions on each possible pair, a graph-based neural network was utilized since the interaction between neighboring pairs is also a crucial piece of information

The input to the model is a series of complete graphs, with each node representing a unique cryptocurrency asset, and the edges being the correlation index calculated using the Pearson's Correlation Coefficient, for both closing price and trading volume. Each graph represents the state of the market at a specific unix timestamp. The model input is fed into multiple convolutional layers for it to learn the interaction between different pairs of cryptocurrencies as well as the temporal changes of the correlation.

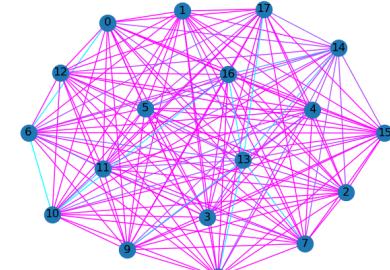
The model takes in the past 30 days of graphs (represented as an adjacency list) and predicts the next graph representing the correlation state of the next four hours. By reusing the predicted graphs as an input to the model we were able to produce a series of future correlation graphs at decreasing accuracy.

The result of the correlation prediction is visualized by varying the color of the edges. The color spectrum we use is blue to magenta, which represents a correlation index of -1 to 1. Here are several frames of such predictions:

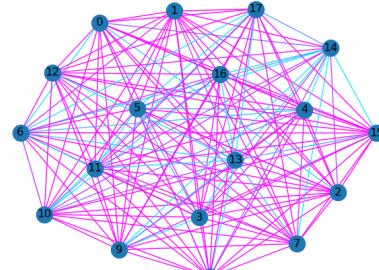
Hour 0:



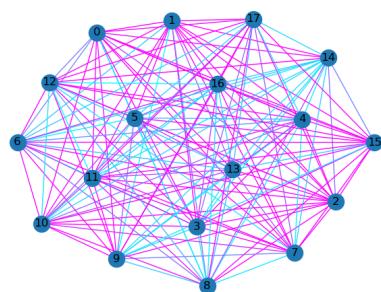
Hour 120:



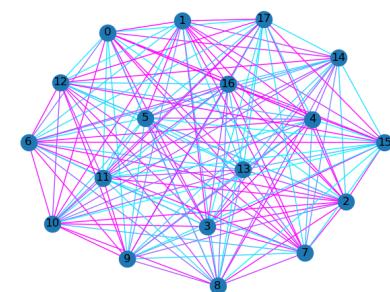
Hour 240:



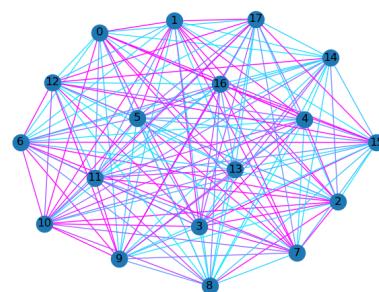
Hour 360:



Hour 480:



Hour 600:



The 3 frames above are the prediction of the state of closing price correlations between different cryptos in future timestamps. For the ease of viewing, the closing price correlation prediction was composed into a video (Video link: https://drive.google.com/file/d/16mhfgLrw0iKaZsb2u0Rs_9Fx8uMXv2al/view?usp=sharing).

An effective metric to evaluate the performance of this model has not yet been found, but for now the predictions within the first few days appear to be relatively reliable.

3.4 Sentiment Analysis

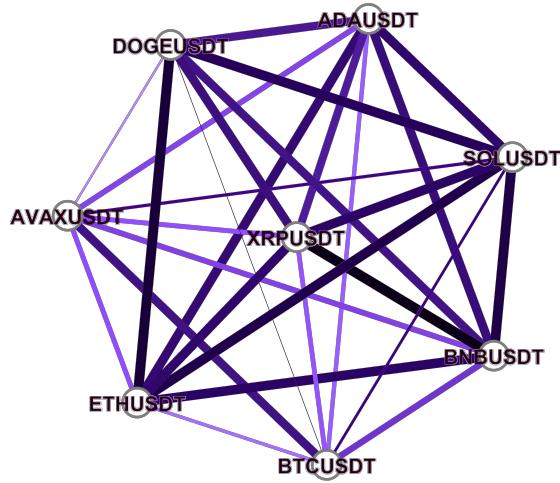
Sentiment analysis is a key part of real-time price prediction. Large cryptocurrency price movements often stem from significant news, and sentiment analysis can be used to determine the general opinion of the public with regards to each cryptocurrency.

Posts were scraped from Reddit ([r/cryptocurrency](#)), and the titles were fed through a pretrained BERT model. This model was able to determine bullish/bearish/neutral sentiment with an accuracy of over 90%.

4 Method

The backbone of the prediction model is a complete graph, where each node is a unique cryptocurrency, and each edge holds similarity data for each pair of cryptocurrencies, which includes K-means cluster ID, DBSCAN, cluster ID, linear regression r^2 , etc.

It is difficult to reasonably visualize a graph with multiple edge weights, so for the example graph below, only the weights for the linear regression are used:



Given a cryptocurrency asset, the model finds the 10 most similar cryptocurrencies based on a weighted scoring system. The aggregate score is calculated using different clustering metrics (DBSCAN, K-means, etc.) as well as the linear regression r^2 on historical prices mentioned above. Since the clustering metrics are trained with metadata of the respective cryptocurrency asset (which is constant in the short-run) and linear regression on historical price, we are able to produce a similarity score that takes into consider of both static traits (ex: max-supply, market-cap) and quantitative price data.

5 Summary of Findings

5.1 Results

An unbiased and quantitative evaluation metric for this model has not been found, but from extended testing, it was found that the resulting output of the similarity model agrees with the price correlation, and the actual price of the cryptos to some extent.

The similarity score produced by the model is trivially correlated with previous price data, and for future price data, the model is able to accurately find cryptocurrencies that tend to move with each other.

5.2 Possible Confounding of Data

Although the data and results show a strong positive correlation, it is likely that the similarities that were observed between cryptocurrencies are largely due to market-wide price movements. More often than not, the entirety (or close to the entirety) of the cryptocurrency market moves together. This leads to the price history of many cryptos appearing to be directly correlated with each other, when they are, in actuality, correlated with BTCUSDT, ETHUSDT, or another coin.

In other words, the similarity metrics that were used may not be a good representation of possible similarities between cryptocurrencies, despite the observation of good performance.

6 Conclusion

6.1 Completed Goals

Ultimately, the prediction and grouping of similar cryptocurrencies was implemented and appears to work relatively well. Also completed is correlation prediction, sentiment analysis, and real-time scraping. The completed work, however, was not analogous to the initial goals that were set at the beginning of the project.

6.2 Incomplete Goals

The largest unimplemented feature was real-time price prediction. While piecing together a working real-time module, it became evident that it was quickly going out of scope of the assignment, as it had very little to do with graph mining and graphs in general. It was extremely algorithm based, with data being stored in non-graph data structures. Additionally, issues with the Twitter API, and fundamental flaws in the scraping logic led to real-time price prediction being dropped.

A direct result of this was several unused, but functional modules that were intended to be used in real-time prediction. This includes sentiment analysis using BERT, correlation prediction, and various custom technical indicators that were implemented.

6.3 Future Work

A clear next step would be to piece together the real-time price prediction. This should theoretically be straightforward, as most of the fundamental components of the model are already completed. The primary focus would be on working around the core issues that were encountered, such as the Twitter API only storing 7 days of prior tweets, and the delay between news, price movement, and when the data could be scraped and processed.

7 Sample Output

Below is an example use of the program.

```
Enter a ticker for similar cryptocurrencies: BTCUSDT
```

```
Top 10 similar cryptocurrencies:
```

01: NEARUSDT	similarity: 0.928039
02: DOTUSDT	similarity: 0.910973
03: ATOMUSDT	similarity: 0.890866
04: EGLDUSDT	similarity: 0.886348
05: SANDUSDT	similarity: 0.880285
06: LUNAUSDT	similarity: 0.877113
07: BATUSDT	similarity: 0.826980
08: XTZUSDT	similarity: 0.822819
09: AVAXUSDT	similarity: 0.818994
10: DASHUSDT	similarity: 0.797945

```
Enter a ticker for similar cryptocurrencies: NEARUSDT
```

```
Top 10 similar cryptocurrencies:
```

01: DOTUSDT	similarity: 0.953060
02: SANDUSDT	similarity: 0.939675
03: LUNAUSDT	similarity: 0.933701
04: BTCUSDT	similarity: 0.928039
05: ATOMUSDT	similarity: 0.886720
06: ENJUSDT	similarity: 0.883375
07: BATUSDT	similarity: 0.883180
08: EGLDUSDT	similarity: 0.875387
09: UNIUSDT	similarity: 0.846779
10: FTTUSDT	similarity: 0.841460

```
Enter a ticker for similar cryptocurrencies: ETHUSDT
```

```
Top 10 similar cryptocurrencies:
```

01: MKRUSDT	similarity: 0.976515
02: WAVESUSDT	similarity: 0.962074
03: RUNEUSDT	similarity: 0.954988
04: EOSUSDT	similarity: 0.949072
05: ETCUSDT	similarity: 0.930652
06: SOLUSDT	similarity: 0.926456
07: BCHUSDT	similarity: 0.924530
08: XMRUSDT	similarity: 0.904530
09: BNBUSDT	similarity: 0.899182
10: NEOUSDT	similarity: 0.897319

```
Enter a ticker for similar cryptocurrencies: ADAUSDT
```

```
Top 10 similar cryptocurrencies:
```

01: MATICUSDT	similarity: 0.814132
02: ALGOUSDT	similarity: 0.746764

Cryptocurrency Recommender

Final Report

05/01/2022
CSCI-4964
Spring 2022

```
03: DOGEUSDT      similarity: 0.564805
04: XRPUSDT       similarity: 0.546123
05: HBARUSDT      similarity: 0.540680
06: VETUSDT        similarity: 0.533386
07: ONEUSDT        similarity: 0.519367
08: XLMUSDT        similarity: 0.519099
09: TRXUSDT        similarity: 0.513903
10: IOTXUSDT       similarity: 0.508165
```

Enter a ticker for similar cryptocurrencies: NOTACRYPTO
Ticker is not in graph.

Enter a ticker for similar cryptocurrencies: exit