UNDERSTANDING THE CRYPTO MARKET USING TWEETS

Mesaye BAHIRU (260634934)
Brock BECKLEY (260567741)
Danyal HAMID (261024542)
Mohamad KHALILI (260746712)
Mehdi YACHFINE (261005628)

Table of Contents

Introduction & Problem Formulation

On social media, people claiming success from cryptocurrencies are everywhere. Dubai vacation, sports cars, expensive restaurants, and flashy watches: this is the life that successful "crypto" traders showcase. What seems to be an easy and guaranteed path to wealth, is, in fact, a lot more complicated than one might expect. The cryptocurrency market is extremely unstable and incredibly complicated to gauge and understand, due to cryptocurrencies themselves being highly volatile assets. Their value fluctuates immensely in the span of a few hours. As opposed to stocks, cryptocurrencies hold no intrinsic value. The value of a stock is intrinsically correlated with a company's performance & profitability. For example, on the one hand, Amazon ($AMZN) and Netflix ($NFLX) saw their stock prices soar during the pandemic, due to an increase in online shopping and a higher demand for video streaming services. On the other hand, cryptocurrency price fluctuations are harder to capture. The reason why Bitcoin, priced at $20,118 on 12/12/2020, blew up to almost $40,000 on 07/01/2021 is quite nebulous. Rather, the price of cryptocurrencies is highly dependent on its trader/user base. If they are feeling confident or doubtful about the performance of bitcoin on a specific day, it causes its value to fluctuate either positively or negatively. Moreover, these cryptocurrencies are even more sensitive to what influential people think about them. More than once, Elon Musk's tweets either made the price of some cryptocurrencies, such as Dogecoin ($DOGE), skyrocket or completely crash. **Therefore, user sentiment can have a tremendous impact on the cryptocurrency market. Being able to gauge and quantify that sentiment could help us attain a better understanding of the cryptocurrency market and build a periodic "picture" of how the market is behaving in each timeframe.**

The purpose of the analysis presented in this project is to extract valuable information from social media to better understand the crypto market. First, tweets revolving around cryptocurrencies and bitcoin have been pulled using the Twitter API. Once these tweets have been extracted, they need to undergo some pre-processing before insights are derived. Pre-processing steps include transforming the tweet into "normal" text by removing hashtags, mentions, and characters such as Retweets. Then, conventional pre-processing steps such as removing punctuation and stop words, lowercasing, lemmatization, and stemming are applied.

Various insights have been derived from the pre-processed tweets. First, a VADER sentiment analysis was performed to obtain polarity scores for negative, neutral, and positive sentiments for each tweet. These sentiments have been calculated on an hourly basis and were linked to hourly Bitcoin prices. This information was used to train a Long Short-Term Memory (LSTM) Neural Network, to predict Bitcoin prices solely based on the sentiment of the past 3 hours. Another insight that was interesting to obtain was an aggregate of the topics people discuss when talking crypto on Twitter. To do so, a word cloud containing the 200 most mentioned topics revolving around Bitcoin and cryptocurrencies was generated. Finally, we were interested in looking at the association levels of cryptocurrencies between each other, and cryptocurrencies with high-profile influencers, such as Elon Musk (CEO of SpaceX, Tesla), Jack Dorsey (Ex-CEO of Twitter), and Changpeng Zhao (CEO of Binance).

Obtaining the tweets data and the bitcoin price

The first step is to apply for access to Twitter API through the Twitter Developer Access platform. After subscribing and completing the required setup, a "Bearer Token" is given to enable the API and make it work in the code.

The "requests" library is used to get the most recent tweets through a query toward the following address: "https://api.twitter.com/2/tweets/search/recent". The headers and the search parameters are then set to search within recent tweets those that contain the words "bitcoin" or "BTC" (bitcoin's ticker). The max result parameter is set to the maximum value available with the free API, 100 tweets. Then it's specified that the targeted tweets should be in English and that the timestamps are to be scrapped too. The Twitter API has some restrictions for which we had to find a workaround. First off, the API only allows us to query up to 100 tweets at a time. Second, it will not allow the querying of tweets that are more than a week old.

To circumnavigate this problem, we defined two functions: one to get the data from the requests and organize it into a dictionary, and another one to subtract 60 minutes from the timestamps. These two functions are then used to scrape 100 tweets for each hour of the last 7 days, to get a data frame of around 16k tweets. This dataset is then exported to a CSV format.

For the bitcoin's price data, we used the yahoo finance API "yfinance" and got the data for the last 7 days after specifying the bitcoin ticker (BTC-USD). This dataset is then converted to a CSV format.

Pre-processing

Once extracted, before being used for analysis, the tweets must be cleaned and pre-processed into simplified, usable text. The pre-processing step is highly important, as a poorly pre-processed dataset will likely contain significant noise, the sentiment analysis could become erroneous and deriving insights from it becomes more complicated. As opposed to conventional text, tweets have a very particular writing style, containing a lot of special characters. The first step in the pre-processing is to transform the tweets into "normal" text. This step consists in removing special text items such as mentions (@), hashtags (#), as well as retweets (RT). To do this, the tweet-preprocessor Python package was used. This package incorporates all these operations and is quite simple to use, and a function is used to do this step of the pre-processing.

```
def preprocess_tweet(row):
    text = row['text']
    text = p.clean(text)
    return text

tweets['Processed_Tweet'] = tweets.apply(preprocess_tweet, axis = 1)
tweets = tweets.drop(columns = 'Unnamed: 0')
tweets.head()
```

	id	created_at	text	Processed_Tweet
0	1492483100756307970	2022-02-12 13:00:00+00:00	RT @AirdropDet: 🔍New #Airdrop: payMe Trivia Qu...	: New : payMe Trivia QuizReward: Up to K payME...
1	1492483099930013698	2022-02-12 13:00:00+00:00	RT @deficonnect: Next on the list....🚀\n\nDEFI...	: Next on the list....DEFIPAY..
2	1492483099527110658	2022-02-12 13:00:00+00:00	RT @CihanYedek: BobNftDesign NFT Giveaway\nHon...	: BobNftDesign NFT GiveawayHonouring Bob Marle...
3	1492483099485085699	2022-02-12 13:00:00+00:00	RT @AirdropStario: 💧 NFTOCEAN Airdrop 💧\n\n🏆 T...	: NFTOCEAN Airdrop Task: NO Referral: NO Airdr...
4	1492483098717884416	2022-02-12 13:00:00+00:00	RT @cmttat: wen btc 100k ? :(	: wen btc k ?

Next, the regular pre-processing follows. All the subsequent steps have been done using Python's NLTK package. In texts, words such as "I", "me", "this", "that" are used very often, but do not provide any interesting insight. These words generate noise and are therefore eliminated from the tweets. The code used to remove the stop words is shown below.

```python
stop = stopwords.words('english')
tweets['Processed_Tweet'] = tweets['Processed_Tweet'].apply(lambda x: ' '.join([word for word in x.split()
                                                                   if word not in (stop)]))

tweets.head()
```

	id	created_at	text	Processed_Tweet
0	1492483100756307970	2022-02-12 13:00:00+00:00	RT @AirdropDet: 🔍New #Airdrop: payMe Trivia Qu...	: New : payMe Trivia QuizReward: Up K payME (~...
1	1492483099930013698	2022-02-12 13:00:00+00:00	RT @deficonnect: Next on the list....🚀\n\nDEFI...	: Next list....DEFIPAY..
2	1492483099527110658	2022-02-12 13:00:00+00:00	RT @CihanYedek: BobNftDesign NFT Giveaway\nHon...	: BobNftDesign NFT GiveawayHonouring Bob Marle...
3	1492483099485085699	2022-02-12 13:00:00+00:00	RT @AirdropStario: 💧 NFTOCEAN Airdrop 💧\n\n🏆 T...	: NFTOCEAN Airdrop Task: NO Referral: NO Airdr...
4	1492483098717884416	2022-02-12 13:00:00+00:00	RT @cmttat: wen btc 100k ? :(	: wen btc k ?

Once these stop words have been removed, punctuation is also removed from the text. The screenshot below presents the initial tweet (*text)*, and the pre-processed tweet (*Processed_Tweet*) is ready to be used for analysis.

```python
for i in range(len(tweets)):
    tweets['Processed_Tweet'][i] = tweets['Processed_Tweet'][i].lower()
tweets.head()
```

	id	created_at	text	Processed_Tweet
0	1492483100756307970	2022-02-12 13:00:00+00:00	RT @AirdropDet: 🔍New #Airdrop: payMe Trivia Qu...	: new : payme trivia quizreward: up k payme (~...
1	1492483099930013698	2022-02-12 13:00:00+00:00	RT @deficonnect: Next on the list....🚀\n\nDEFI...	: next list....defipay..
2	1492483099527110658	2022-02-12 13:00:00+00:00	RT @CihanYedek: BobNftDesign NFT Giveaway\nHon...	: bobnftdesign nft giveawayhonouring bob marle...
3	1492483099485085699	2022-02-12 13:00:00+00:00	RT @AirdropStario: 💧 NFTOCEAN Airdrop 💧\n\n🏆 T...	: nftocean airdrop task: no referral: no airdr...
4	1492483098717884416	2022-02-12 13:00:00+00:00	RT @cmttat: wen btc 100k ? :(	: wen btc k ?

VADER Sentiment Analysis and price prediction

Now that the tweets have been fully pre-processed, the analysis can begin. The first part of the analysis will be performing a sentiment analysis on each tweet to obtain a polarity score, quantifying the different possible sentiments, those sentiments being negative, neutral, and positive. A compound sentiment is also calculated, to get a single, standardized sentiment score for each tweet. The code below shows how the VADER sentiment analysis has been incorporated into the rest of the data, creating a unique, easy-to-understand data frame.

```python
from nltk.sentiment.vader import SentimentIntensityAnalyzer

# Instantiate the sentiment intensity analyzer
vader = SentimentIntensityAnalyzer()

# Iterate through the headlines and get the polarity scores using vader
scores = tweets['Processed_Tweet'].apply(vader.polarity_scores).tolist()

# Convert the 'scores' list of dicts into a DataFrame
scores_df = pd.DataFrame(scores)

# Join the DataFrames of the news and the list of dicts
tweets = tweets.join(scores_df, rsuffix='_right')
tweets.head()
```

	id	created_at	text	Processed_Tweet	neg	neu	pos	compound
0	1492483100756307970	2022-02-12 13:00:00+00:00	RT @AirdropDet: 🔍New #Airdrop: payMe Trivia Qu...	new payme trivia quizreward up k payme 35 m...	0.000	1.000	0.000	0.0000
1	1492483099930013698	2022-02-12 13:00:00+00:00	RT @deficonnect: Next on the list.... 🚀\n\nDEFI...	next listdefipay	0.000	1.000	0.000	0.0000
2	1492483099527110658	2022-02-12 13:00:00+00:00	RT @CihanYedek: BobNftDesign NFT Giveaway\nHon...	bobnftdesign nft giveawayhonouring bob marley...	0.000	0.663	0.337	0.6808
3	1492483099485085699	2022-02-12 13:00:00+00:00	RT @AirdropStario: 💧 NFTOCEAN Airdrop 💧 \n\n 🏆 T...	nftocean airdrop task no referral no airdrop ...	0.355	0.645	0.000	-0.5267
4	1492483098717884416	2022-02-12 13:00:00+00:00	RT @cmttat: wen btc 100k ? :(	wen btc k	0.000	1.000	0.000	0.0000

Now that we computed the sentiments, let's use them to extract some interesting insights. In the context of this study, the compound score is going to be used as a predictor to estimate the price of Bitcoin. It is important to note that 16,000 tweets have been extracted on a period of a week using the Twitter API, which amounts to roughly 100 tweets per hour. **We were, therefore, able to calculate a mean hourly sentiment: the compound sentiment of all the tweets that were written during the same hour were averaged to obtain this mean hourly sentiment, as shown below.** For example, as shown in the few rows below, the average hourly compound sentiment is shown for the 5th of February, starting at 8 PM. The reason why it starts at 8 PM only is because the tweets were pulled on the 12th of February at 8PM, and the student version of the API allows us to go back one week only.

```
#Getting the day and hour to get the mean sentiment score per hour
df = tweets[['created_at', 'compound']]
df.created_at = pd.to_datetime(df['created_at'])


#Getting the day and hour to get the mean sentiment score per hour
def get_hour(date):
    return date.hour
def get_day(date):
    return date.day

df['hour'] = df.created_at.apply(get_hour)
df['day'] = df.created_at.apply(get_day)

#Getting the mean sentiment score per hour
df2 = df.groupby(['day','hour']).mean().reset_index()
df2.head()
```

	day	hour	compound
0	5	20	0.300386
1	5	21	0.178428
2	5	22	0.171779
3	5	23	0.223601
4	6	0	0.177892

These average hourly compound values were then joined alongside the hourly closing prices of Bitcoin:

```
#importing bitcoin prices
btc_df = pd.read_csv('bitcoin_price.csv')
btc_df = btc_df[['Datetime','Close']].reset_index(drop=True)

btc_df['return']=btc_df.Close.pct_change()
btc_df.Datetime = pd.to_datetime(btc_df['Datetime'])
btc_df['hour'] = btc_df.Datetime.apply(get_hour)
btc_df['day'] = btc_df.Datetime.apply(get_day)


df = pd.merge(btc_df, df2, how='left', on=['hour','day'])
df = df.dropna()
df = df.drop(['hour','day'],axis=1)

#Dropping the rows lost due to the data being only available starting at 8PM on the 5th and until 8PM on the 12th of
#btc_df = btc_df.drop(btc_df.index[[0,1,2,3,4,5,6,7,8,9,10,11,12,13,14]])
#btc_df = btc_df.reset_index(drop = True)
#df2 = df2.drop(df2.index[[154,155,156,157,158,159,160,161]])
#df2 = df2.reset_index(drop = True)
Dataframe_Final = pd.DataFrame(columns = ['Sentiment','Price'])
Dataframe_Final['Sentiment'] = df2['compound']
Dataframe_Final['Price'] = btc_df['Close']
Dataframe_Final.head(10)
```

	Sentiment	Price
0	0.300386	41547.824219
1	0.178428	41530.023438
2	0.171779	41401.394531
3	0.223601	41450.542969
4	0.177892	41443.675781
5	0.252131	41552.460938

Having the corresponding closing price with each sentiment will allow us to predict the price of Bitcoin based on sentiment. To do so, the data frame of sentiments and prices was manipulated to obtain lag values for the previous 3 hours. The resulting data frame was used to train an LSTM neural network. We chose to perform a mini-batch gradient descent, which means that the batch size hyperparameter must be between 1 and the size of the training dataset. The batch size means the dataset is split into the selected number of batches. Predictions on each sample are made and compared with the actual values. The error is calculated and is then used to move down along the error gradient, improving the model. Hence the attribute "gradient descent". Then, the number of Epochs is set initially to 30. This means that LSTM's learning algorithm will go through the entire dataset 30 times. The Epoch and batch size hyperparameters can be tuned by cross-validation if the obtained results are unsatisfactory. Below is shown the computation of the first 3 Epochs.

```
# design network
model = Sequential()
model.add(LSTM(5, input_shape=(train_X.shape[1], train_X.shape[2])))
model.add(Dense(1))
model.compile(loss='mae', optimizer='adam')
# fit network
history = model.fit(train_X, train_y, epochs=30, batch_size=4, validation_data=(test_X, test_y), verbose=2, shuffle=
# plot history
```

```
Epoch 1/30
28/28 - 4s - loss: 0.4395 - val_loss: 0.4197 - 4s/epoch - 127ms/step
Epoch 2/30
28/28 - 0s - loss: 0.3607 - val_loss: 0.3348 - 133ms/epoch - 5ms/step
Epoch 3/30
28/28 - 0s - loss: 0.3137 - val_loss: 0.2639 - 143ms/epoch - 5ms/step
```

Plotting the loss values shows us that we reach a loss as low as 0.1:

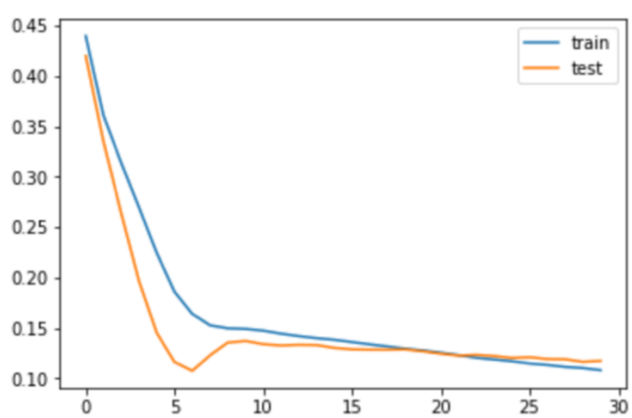

Figure 1 - Loss values

The LSTM Neural Network now being trained, we can make a prediction and see the predictive accuracy.

```
import numpy as np
# make a prediction
yhat = model.predict(test_X)
test_X = test_X.reshape((test_X.shape[0], n_hours* n_features))
# invert scaling for forecast
inv_yhat = np.concatenate((yhat, test_X[:, -1:]), axis=1)
inv_yhat = scaler.inverse_transform(inv_yhat)
inv_yhat = inv_yhat[:,0]
# invert scaling for actual
test_y = test_y.reshape((len(test_y), 1))
inv_y = np.concatenate((test_y, test_X[:, -1:]), axis=1)
inv_y = scaler.inverse_transform(inv_y)
inv_y = inv_y[:,0]
# calculate RMSE
rmse = np.sqrt(mean_squared_error(inv_y, inv_yhat))
print('Test RMSE: %.3f' % rmse)
```

```
Test RMSE: 571.941
```

As shown above, the RMSE is 571.9. Considering the price of Bitcoin is a 5-digit figure revolving around $40,000-$50,000, a RMSE of 571.9 is relatively low. The plot below gives a visual representation of the predictions made. A total of 41 predictions have been made using 110 training data points. **As we can see, even though there is quite a gap between the prediction and the actual, the overall trend is captured quite well by the neural network using only the compound sentiment as a predictor.** Note that through further hyperparameter tuning with methods such as GridSearch, and with more data points, the performance of the LSTM model could be significantly improved.

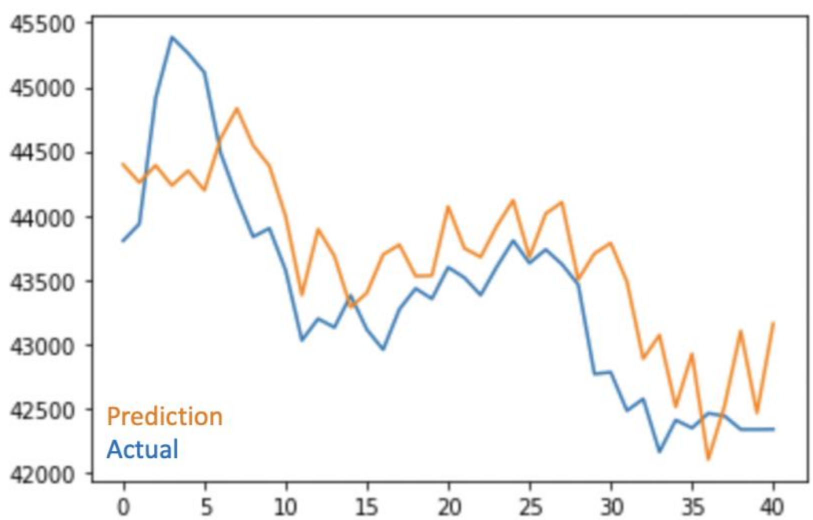

Figure 2 - LSTM prediction vs Actual price of Bitcoin

Most recurring topics around Bitcoin: A word cloud

Predicting the price of Bitcoin is not the only interesting analysis we can perform with the tweets. When tracking the cryptocurrency market, one may want to know which are the main discussion topics, that are the most recurring when people talk cryptocurrencies on Twitter. For this purpose, a word cloud presenting the top 200 most recurring words was generated.

Figure 3 - Word Cloud

As shown in the word cloud above, some topics are recurring quite often when people discuss cryptocurrencies on Twitter. For example, around Bitcoin, people frequently mentioned the topic of airdropping, which consists in sending cryptocurrencies to user's wallets. Other cryptocurrencies such as ZRK, AMP and ETH were also mentioned. The acts of referring and mining were also brought up. Finally, non-fungible tokens, also called NFTs, which are currently booming, are also mentioned in tweets revolving around Bitcoin and cryptocurrencies.

Association with other cryptos and people: lift values & MDS plot

Lift and MDS Analysis on Cryptocurrencies and Stocks

Social media platforms like Twitter have become a top choice for finance news compared to traditional finance newspapers or news networks. A Lift analysis was conducted to see which cryptocurrencies and stocks were most strongly associated together in the eyes of the Twitter financial community. There are five stages to go through for a successful lift and MDS analysis on cryptocurrencies and stocks.

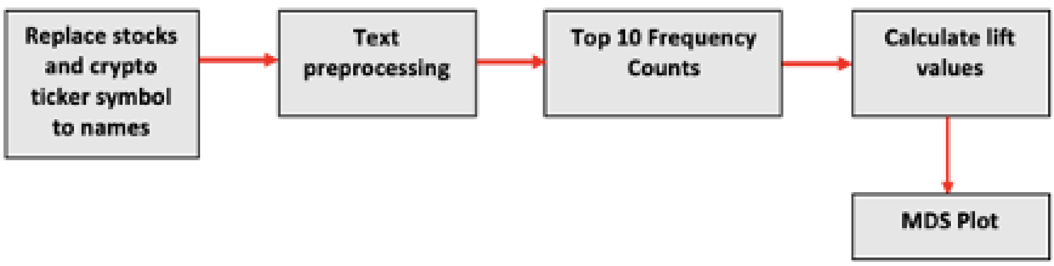

Figure 4 – flowchart for lift and MDS plot analysis

The dataset “tweetd_btc.csv” used for this analysis contains both stocks and cryptocurrency names and ticker symbols in most of the text. Find and replace method was utilized to change tickers symbols to names of coin or stocks. Furthermore, a dataset called “investment.csv” was created, with the 50 most highly traded stocks and cryptocurrency coins. Then, using this dataset the find and replace method was applied to replace ticker symbols with names of stocks or coins.

NLTK (natural language toolkit) library allowed us to transform the raw tweets containing irrelevant information into a structured text. As was explained in the pre-processing section, the data preprocessing began with removing hashtag and mentions, followed by removing punctuation, putting text in lowercase, word tokenization, removing stop words and finally stemming the text to their root forms. Once the preprocessing was completed, a frequency count was performed to identify top 10 coins or stocks with the highest mentions. The result is shown in table 1

Table 1 - Frequency of appearance of other stocks & cryptos

NAME	FREQUENCY
MACYS	8883
BITCOIN	7908
VISA	2317
ETHEREUM	2196
SIACOIN	878
300 TOKEN	358
TETHER	313
0X (ZRK)	298
DOGECOIN	235
42-COIN	223

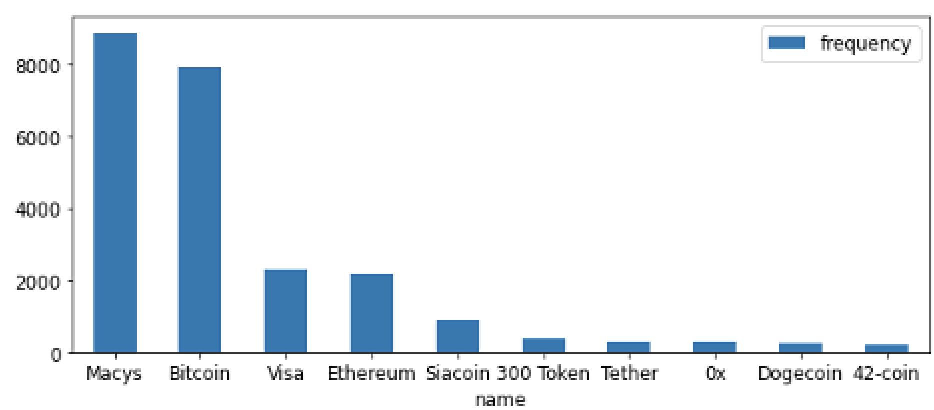

Figure 5 – Top 10 Names by Frequency

Afterwards, the lift values were calculated by computing the occurrence of two names simultaneously. Lift values provide us an insight into pairwise association of the two names. The higher the lift values, the greater the association of the names. Table 2 summarizes the lift values of the analysis and lift values highlighted in red shows the strong association between the names followed by the green highlighted values and finally the yellow highlighted values.

Table 2 - Lift values of cryptocurrencies between each other

Names	Macys	Bitcoin	Visa	Ethereum	Siacoin	300 Token	Tether	0x	Dogecoin	42-coin
Macys	0.00	1.02	1.35	1.04	1.47	1.06	1.01	1.44	1.07	1.00
Bitcoin	0.00	0.00	1.21	1.48	1.40	0.23	1.07	1.32	1.58	1.46
Visa	0.00	0.00	0.00	1.37	2.57	0.16	1.25	1.75	1.66	1.31
Ethereum	0.00	0.00	0.00	0.00	2.12	0.35	0.89	2.09	4.44	0.99
Siacoin	0.00	0.00	0.00	0.00	0.00	0.10	2.64	8.38	1.33	1.07
300 Token	0.00	0.00	0.00	0.00	0.00	0.00	0.43	0.60	0.38	0.40
Tether	0.00	0.00	0.00	0.00	0.00	0.00	0.00	2.07	4.39	2.08
0x	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	3.22	1.94
Dogecoin	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.92
42-coin	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

The final stage in the analysis is to generate multidimensional scaling (MDS) plot. For MDS plot, dissimilarity score was calculated by taking the inverse of the lift values. The plot illustrated three key findings.

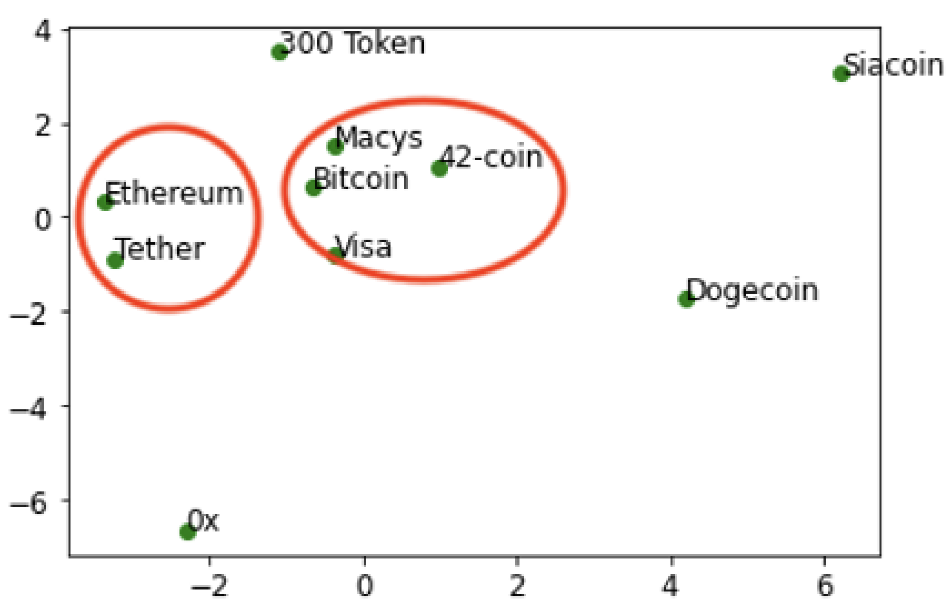

Figure 6 – MDS Plot

First, bitcoin was closely associated with Macys/42-coin/Visa, while Ethereum was closely related with Tether. This indicated that Bitcoin had a strong association not only with other cryptocurrencies but also with stocks like Visa and Macys. This finding will benefit an investor who owns stocks like Macys and Visa. If an investor holding these two stocks is looking to add cryptocurrency to his portfolio, based off of this MDS plot, Bitcoin would be the recommended option. Second, Ethereum and ether were strongly associated. Tether is the only US-backed cryptocurrency in the market and there is a strong push from entrepreneurs, investors, and crypto enthusiasts to recognize Ethereum as a legitimate payment option. Hence, both were more likely to be mentioned together and the plot supports this finding. Third, the cryptocurrency market is fast-moving, and most investors are more likely to miss out on the next big thing in the cryptocurrency world. However, our analysis can be used as a discovery tool that identifies which cryptocurrencies to place on a watchlist. There are many coins added in the market but being able to spot which ones might have a potential to succeed is key in the investing. In this analysis, it was observed that new coins added to the market like Dogecoin, 300 Token, Siacoin, and 0x had made it to the top 10 most mentioned list.

Lift analysis on influencers & high-profile people

In the world of cryptocurrencies, **some voices are louder than others**, and towards this end, we tried to take a deeper look into the most influential people in cryptocurrencies and their associations with certain cryptocurrencies as per Twitter.

In the ever-changing world of crypto, it may not be enough to know the people who are likely to be most relevant for the year ahead, one might also want to know who to watch/observe in a shorter time frame say, a week. Furthermore, one might also want to know whose opinion comes up most often in conjunction with which cryptocurrency. These are both things we investigate in this section.

We started by web scraping a list of the top 100 most influential people in the cryptocurrency space. We were able to obtain both their names and their Twitter handles. We then did frequency counts for mentions of their Twitter handles in the pre tokenized text. We tried the names approach but the lack of uniqueness of the names meant common names like

Ali would show up at a disproportionately high frequency and we would have no way of knowing with certainty if it was the actual person we were looking for. We choose to use Twitter handles (@), which are unique to each user, to ascertain mentions.

Table 3 - Frequency Table for influencers

	frequency
@elonmusk	147
@cz_binance	105
@danheld	28
@jack	27
@tyler	17
@lopp	11
@novogratz	9
@cryptomanran	9
@muneeb	7
@aantonop	6

Table 4 – Ranking for influencers from a crypto-focused website

	rank
@cz_binance	1
@VitalikButerin	2
@jihanwu	3
@BarrySilbert	4
@brian_armstrong	5
@iohk_charles	6
@michael_saylor	7
@jack	8
@RaoulGMI	9
@Kris_HK	10

On the left, we have mentions found in our dataset of tweets (in the last week), and on the right, we have the list of the top 10 most influential people in Crypto for the year 2021. The difference in the results is explained by two factors, different sets of people from that list of 100 will be relevant in different time frames, and not every influential person in crypto is in the public eye or perhaps even all that active on Twitter.

We then went one step further to investigate who has a strong association with which currency and got the following results. Computing lift values for influencers with relation to cryptocurrency:

Table 5 - Lift values between influencers and cryptocurrencies

	Macys	Bitcoin	Visa	Ethereum	Siacoin	300 Token	Tether	0x	Dogecoin	42-coin
@elonmusk	1.22	1.02	0.26	1.01	0.21	0.17	1.01	0.81	0.78	0.30
@cz_binance	1.46	1.17	1.97	1.74	1.20	0.56	1.13	1.13	1.46	1.70
@danheld	1.23	1.72	0.00	1.29	0.00	0.00	2.32	3.87	1.99	0.00
@jack	1.70	1.19	0.00	0.82	1.30	0.00	4.08	0.00	1.05	1.22
@tyler	2.37	5.07	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
@lopp	0.31	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
@novogratz	0.70	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
@cryptomanran	2.21	9.28	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
@muneeb	2.33	1.31	0.00	0.00	0.00	0.00	0.00	0.00	0.00	11.44
@aantonop	1.48	3.44	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

A lift value of 0 indicates no association, no co-mentions for the person-commodity pair in question. A lift value >1 is indicative of the pair being mentioned more often than expected and some association between the two can be established. A lift value <1 indicates that the pair is mentioned less often than expected.

It is interesting to note the CEO of Binance, an app that deals with all the above cryptos, is mentioned in relation to all the currencies as well. While others who talk specifically about a few cryptos are found to be associated with just those cryptos. We also note that while Elon musk is at least mentioned in relation to all the cryptos he isn't strongly associated with anyone with lift values hovering around 1, indicating that no rule can be surmised from this result. Elon Musk is a person who gets talked about in the Crypto space in general as people also talk about other topics and currencies.