Impact of Twitter Discussions on Dogecoin

Steven Li

August 26, 2021

Abstract

Elon Musk is famous for his regular Twitter posts about many different things—especially his companies Tesla and SpaceX. With about 60 million followers, he one of the hottest star on Twitter and his opinions have a big impact on technologies and companies. In the past few months his tweets also covered Dogecoin, a crypto currency featuring a dog. For this project, I am focusing on Elon Musk and all other tweets in #Dogecoin. And I will use all the mentions and retweets to build a sentiment analysis for Dogecoin daily movement. Based on the model result, I recommend the best candidate model and summarized a list of further enhancements.

1. Introduction

With the help of Elon Musk's frequent tweets, Dogecoin has become one of the most famous cryptocurrency in the world. This project is focusing on utilizing Python and Pytorch libraries To successfully predict the price movement, I prepared 3 different categories of data.

A. Since the dogecoin price is heavily rely on investors' interests and cashflow conditions. Macro Economy condition will be included in the model drivers, that includes: Bitcoin Daily Price, SP500 index and NASDAQ index.
B. Elon Musk's frequent tweets about Dogecoin also helps make this cryptocurrency more famous around the world. In that case, one of the key indicator of model is whether Elon Musk has mentioned Dogecoin in his Twitter today.
C. Besides Elon Musk, ordinary people's discussion about Dogecoin can also have impact on Dogecoin price. With more discussion about #Dogecoin, we are expecting the dogecoin price will have a big movement on that day. Here we built a sentiment analysis model to check what is people's reaction about the dogecoin, and based on the sentiment analysis result, we predict the Dogecoin's price movement.

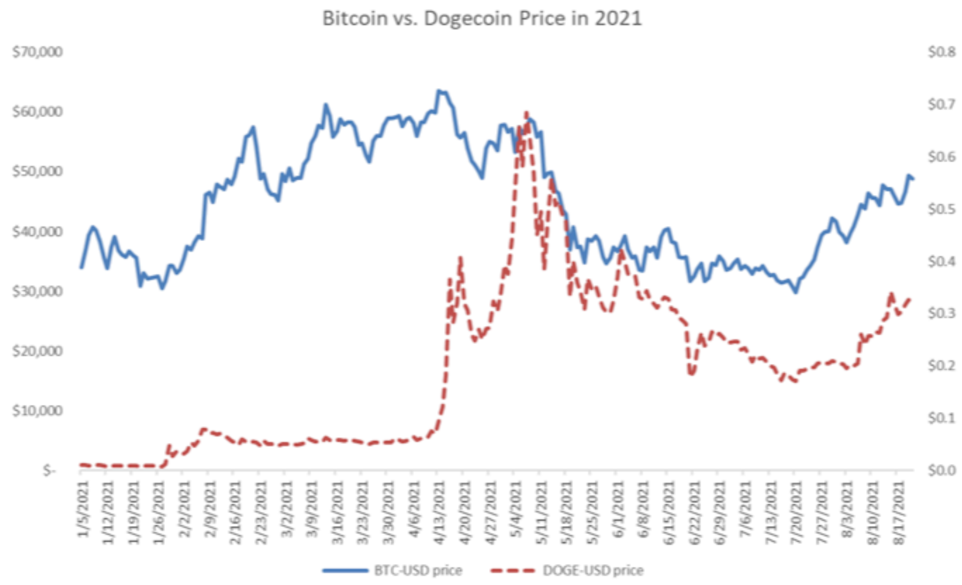
After preparing all the necessary data, a logistic regression/CNN model will be built to train the model and try to predict the Dogecoin price movement.

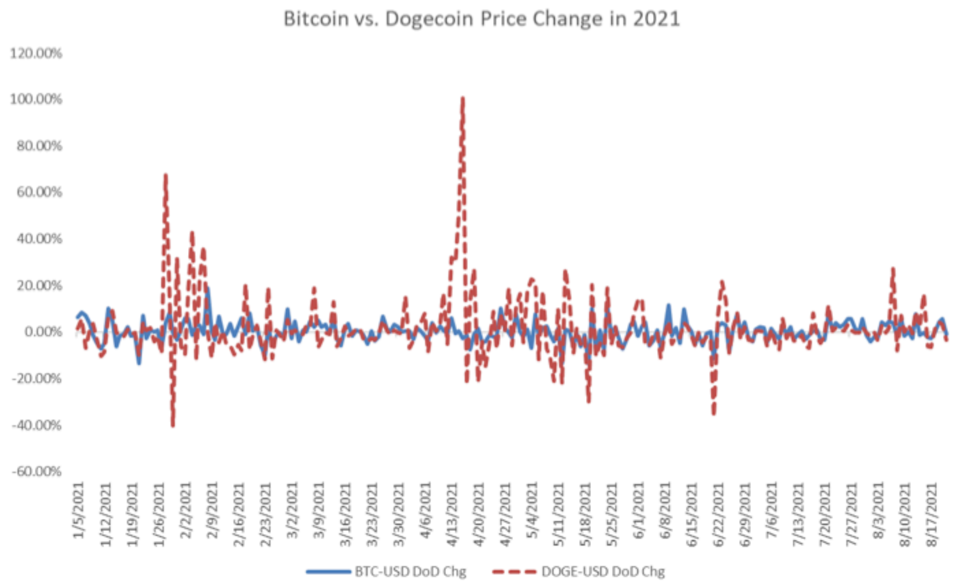
The rest of the paper is organized as following: In Section 2, I describe the Dogecoin price movement problem I have solved and define it mathematically. In Section 3, I explain the raw data processing steps I have done and introduce challenges I am currently facing. In Section 4, I introduced some data exploration I did. Section 6 explained detailed model performance while Section 7 gives a conclusion of this project and points out several potential model enhancements.

2. Problem Definition

The goal of the task in this project is to train a model to predict the price movement of Dogecoin. To simplify the task, I transformed the daily price movement into a binary option: Up or Down.

The charts below shows the Dogecoin price movement comparing with Bitcoin price since 1/5/2021. Based on the observation, before April 2021, the correlation between Bitcoin and Dogecoin is relatively low. However, after the Dogecoin price surge in April, their price movement trend became similar. And we can yield a 30% correlation between their daily movement.

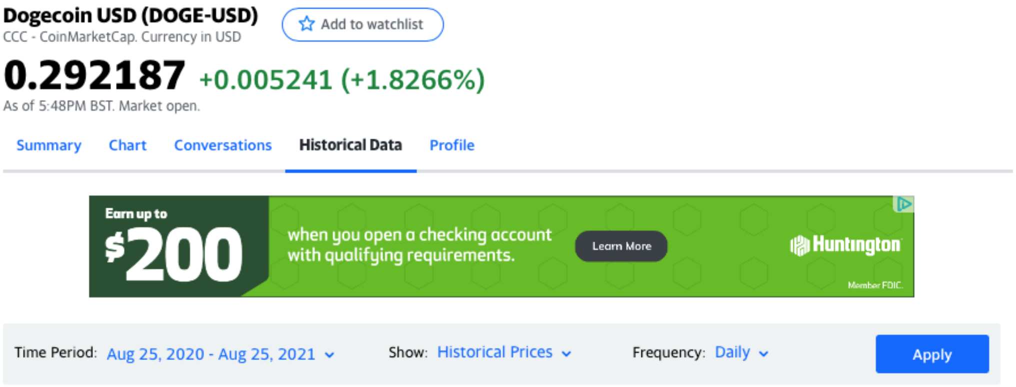




3. Gather and Clean the data

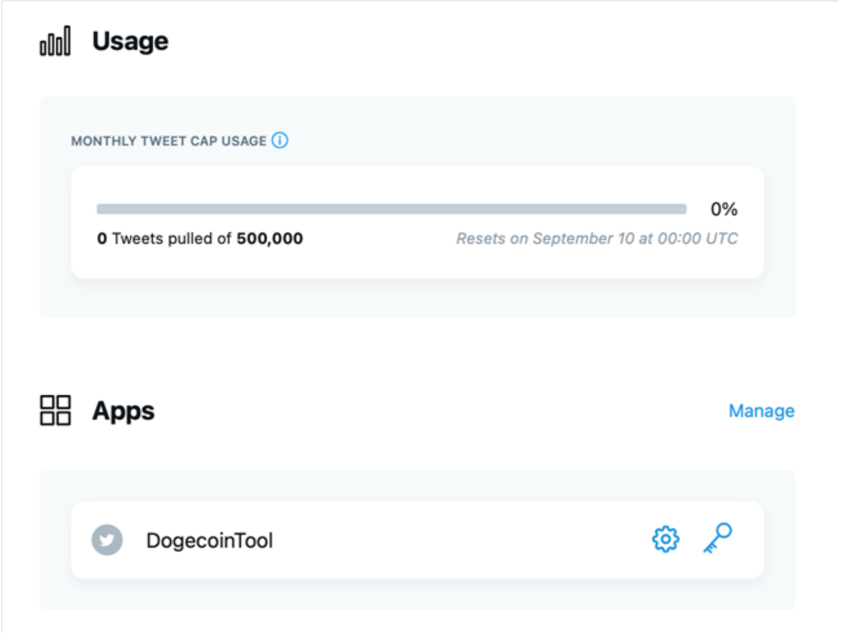
3.1 Data collection

There are mainly 2 parts of data collection, download all the independent variables and Twitter API data pulling. The first part of data collection is straight forward, since we just need daily data to predict the Dogecoin price movement. We can easily download all the related data from free data platform like Yahoo Finance.

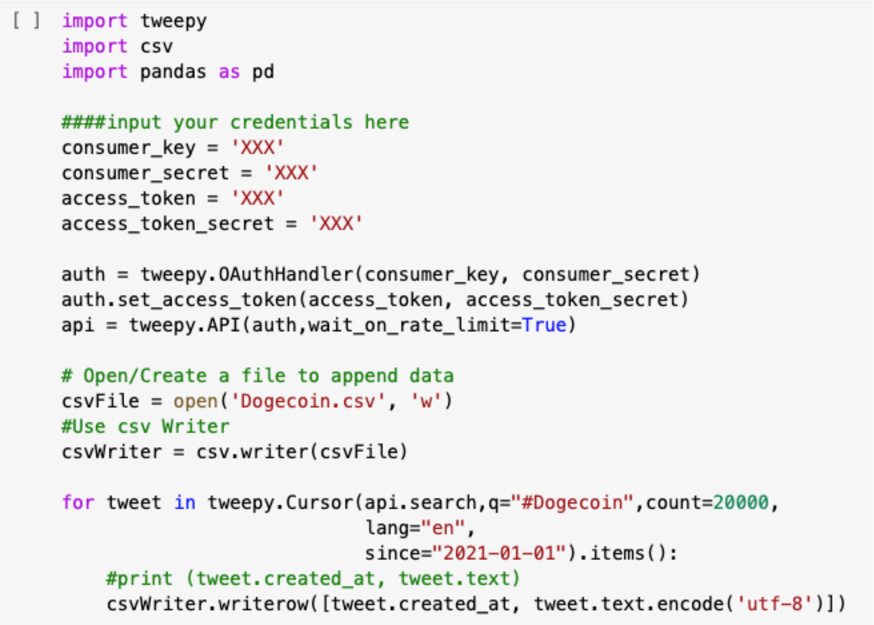

Dogecoin USD (DOGE-USD)
CCC - CoinMarketCap. Currency in USD
0.292187 +0.005241 (+1.8266%)
As of 5:48PM BST. Market open.

Time Period: Aug 25, 2020 - Aug 25, 2021
Show: Historical Prices
Frequency: Daily

The challenging part is the Twitter API data pulling. Based on Twitter's developer guidance, each individual can only pull 500,000 tweets for a single project in a single month. This means for each calendar day, we can only pull about 3000 Dogecoin related tweets. However, there are more than 300,000 #Dogecoin related tweets in a single day. To overcome this issue, I sign-up 4 different accounts to pull #Dogecoin data, and I am able to get about 20,000 tweets each day since January 1st, 2021. The limitation of this method is that the data we pulled is still only a tiny portion of the actual posts, and since the official Twitter API will always feed in the midnight data first. This means that what I got for the #Dogecoin discussions data are mainly from the midnight posts.

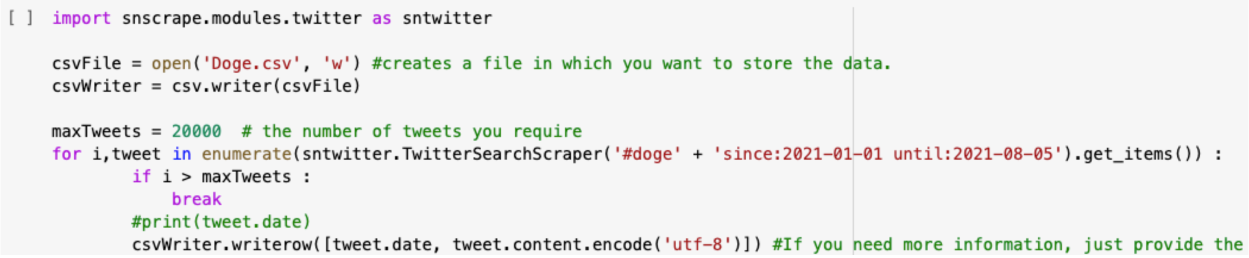

MONTHLY TWEET CAP USAGE
0%
0 Tweets pulled of 500,000
Resets on September 10 at 00:00 UTC

The second challenge I am facing from Twitter API data pulling is that the widely-used Python package Tweepy, it can only pull data for the most recent 7 days. I tried to use the query below to download all the related information for #Dogecoin, and it returns no result prior to a week ago.



```
[ ] import tweepy
    import csv
    import pandas as pd

    ####input your credentials here
    consumer_key = 'XXX'
    consumer_secret = 'XXX'
    access_token = 'XXX'
    access_token_secret = 'XXX'

    auth = tweepy.OAuthHandler(consumer_key, consumer_secret)
    auth.set_access_token(access_token, access_token_secret)
    api = tweepy.API(auth,wait_on_rate_limit=True)

    # Open/Create a file to append data
    csvFile = open('Dogecoin.csv', 'w')
    #Use csv Writer
    csvWriter = csv.writer(csvFile)

    for tweet in tweepy.Cursor(api.search,q="#Dogecoin",count=20000,
                               lang="en",
                               since="2021-01-01").items():
        #print (tweet.created_at, tweet.text)
        csvWriter.writerow([tweet.created_at, tweet.text.encode('utf-8')])
```

After some research, I found another Python package that can help me to pull historical data from Twitter. The only drawback of this package is that this is package is not actively maintained, the most recent updates are 1 year ago. But I am able to utilize this package to download the data I am looking for, there is no issue from this end.

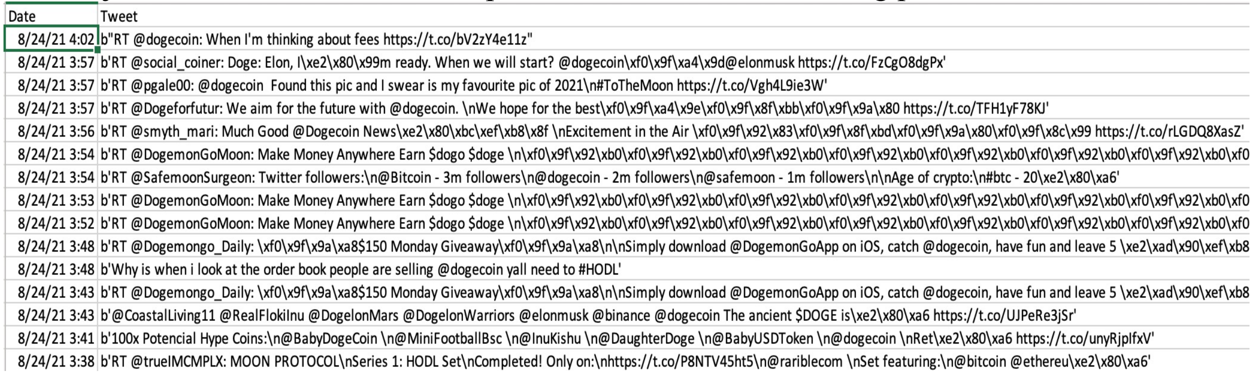


```
[ ] import snscrape.modules.twitter as sntwitter

    csvFile = open('Doge.csv', 'w') #creates a file in which you want to store the data.
    csvWriter = csv.writer(csvFile)

    maxTweets = 20000  # the number of tweets you require
    for i,tweet in enumerate(sntwitter.TwitterSearchScraper('#doge' + 'since:2021-01-01 until:2021-08-05').get_items()) :
            if i > maxTweets :
                break
            #print(tweet.date)
            csvWriter.writerow([tweet.date, tweet.content.encode('utf-8')]) #If you need more information, just provide the
```

In conclusion, data collection is a very time-consuming process, and there are certain limitation about the data I gathered. But I think the data I currently get should be good enough for me to accomplish the project.

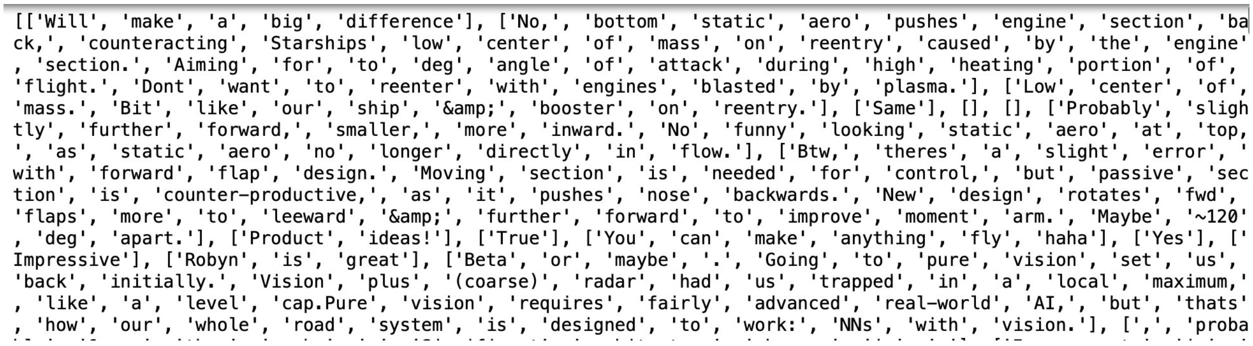
3.2 Data Cleaning

Data cleaning for the raw Twitter data is another challenging part of the project. There are existing code on different open source websites claims to clean the raw twitter data. However, after trying several different codes, they all have their own issue to tokenize all the information. In the actual data, there are a lot of irrelevant information like advertisements, I need to find their pattern to filter all the information out. Besides, there are certain emojis that I cannot clean utilizing the online sourced codes. There are emoji packages that can help me to remove part of the emojis, but I have to find all their patterns to delete the remaining portion.



Date	Tweet
8/24/21 4:02	b"RT @dogecoin: When I'm thinking about fees https://t.co/bV2zY4e11z"
8/24/21 3:57	b'RT @social_coiner: Doge: Elon, I\xe2\x80\x99m ready. When we will start? @dogecoin\xf0\x9f\xa4\x9d@elonmusk https://t.co/FzCgO8dgPx'
8/24/21 3:57	b'RT @pgale00: @dogecoin Found this pic and I swear is my favourite pic of 2021\n#ToTheMoon https://t.co/Vgh4L9ie3W'
8/24/21 3:57	b'RT @Dogeforfutur: We aim for the future with @dogecoin. \nWe hope for the best\xf0\x9f\xa4\x9e\xf0\x9f\x8f\xbb\xf0\x9f\x9a\x80 https://t.co/TFH1yF78KJ'
8/24/21 3:56	b'RT @smyth_mari: Much Good @Dogecoin News\xe2\x80\xbc\xef\xb8\x8f \nExcitement in the Air \xf0\x9f\x92\x83\xf0\x9f\x8f\xbd\xf0\x9f\x9a\x80\xf0\x9f\x8c\x99 https://t.co/rLGDQ8XasZ'
8/24/21 3:54	b'RT @DogemonGoMoon: Make Money Anywhere Earn $dogo $doge \n\xf0\x9f\x92\xb0\xf0\x9f\x92\xb0\xf0\x9f\x92\xb0\xf0\x9f\x92\xb0\xf0\x9f\x92\xb0\xf0\x9f\x92\xb0\xf0\x9f\x92\xb0\xf0\x9f\x92\xb0\xf0
8/24/21 3:54	b'RT @SafemoonSurgeon: Twitter followers:\n@Bitcoin - 3m followers\n@dogecoin - 2m followers\n@safemoon - 1m followers\n\nAge of crypto:\n#btc - 20\xe2\x80\xa6'
8/24/21 3:53	b'RT @DogemonGoMoon: Make Money Anywhere Earn $dogo $doge \n\xf0\x9f\x92\xb0\xf0\x9f\x92\xb0\xf0\x9f\x92\xb0\xf0\x9f\x92\xb0\xf0\x9f\x92\xb0\xf0\x9f\x92\xb0\xf0\x9f\x92\xb0\xf0\x9f\x92\xb0\xf0
8/24/21 3:52	b'RT @DogemonGoMoon: Make Money Anywhere Earn $dogo $doge \n\xf0\x9f\x92\xb0\xf0\x9f\x92\xb0\xf0\x9f\x92\xb0\xf0\x9f\x92\xb0\xf0\x9f\x92\xb0\xf0\x9f\x92\xb0\xf0\x9f\x92\xb0\xf0\x9f\x92\xb0\xf0
8/24/21 3:48	b'RT @Dogemongo_Daily: \xf0\x9f\x9a\xa8$150 Monday Giveaway\xf0\x9f\x9a\xa8\n\nSimply download @DogemonGoApp on iOS, catch @dogecoin, have fun and leave 5 \xe2\xad\x90\xef\xb8
8/24/21 3:48	b'Why is when i look at the order book people are selling @dogecoin yall need to #HODL'
8/24/21 3:43	b'RT @Dogemongo_Daily: \xf0\x9f\x9a\xa8$150 Monday Giveaway\xf0\x9f\x9a\xa8\n\nSimply download @DogemonGoApp on iOS, catch @dogecoin, have fun and leave 5 \xe2\xad\x90\xef\xb8
8/24/21 3:43	b'@CoastalLiving11 @RealFlokiInu @DogelonMars @DogelonWarriors @elonmusk @binance @dogecoin The ancient $DOGE is\xe2\x80\xa6 https://t.co/UJPeRe3jSr'
8/24/21 3:41	b'100x Potencial Hype Coins:\n@BabyDogeCoin \n@MiniFootballBsc \n@InuKishu \n@DaughterDoge \n@BabyUSDToken \n@dogecoin \nRet\xe2\x80\xa6 https://t.co/unyRjpIfxV'
8/24/21 3:38	b'RT @trueIMCMPLX: MOON PROTOCOL\nSeries 1: HODL Set\nCompleted! Only on:\nhttps://t.co/P8NTV45ht5\n@rariblecom \nSet featuring:\n@bitcoin @ethereu\xe2\x80\xa6'

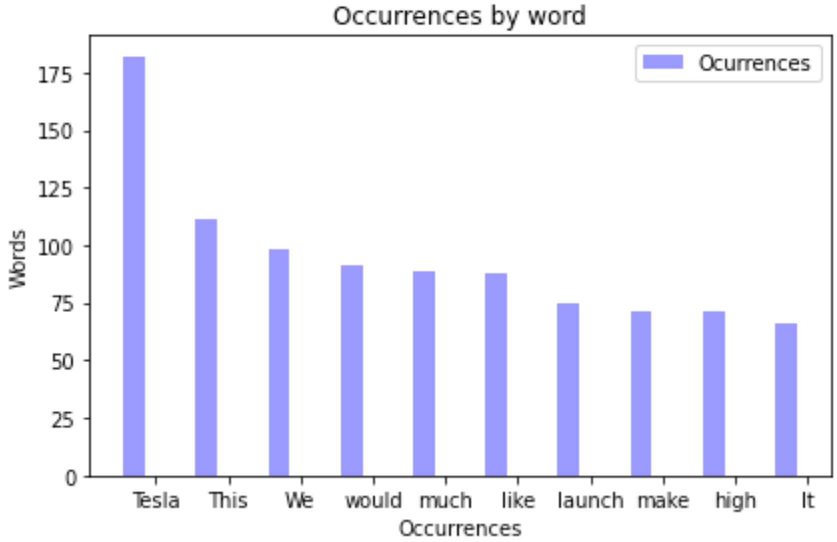
The packages I have been using to clean the data includes: nltk, ekphrasis, nltk and preprocessor. Besides that, I also utilize built-in function in Python to replace some certain phrase to make it work. After all the data clearing process, I am able to tokenize the twitter dataset into a list of words in the form below:



```
[['Will', 'make', 'a', 'big', 'difference'], ['No,', 'bottom', 'static', 'aero', 'pushes', 'engine', 'section', 'ba
ck,', 'counteracting', 'Starships', 'low', 'center', 'of', 'mass', 'on', 'reentry', 'caused', 'by', 'the', 'engine'
, 'section.', 'Aiming', 'for', 'to', 'deg', 'angle', 'of', 'attack', 'during', 'high', 'heating', 'portion', 'of',
'flight.', 'Dont', 'want', 'to', 'reenter', 'with', 'engines', 'blasted', 'by', 'plasma.'], ['Low', 'center', 'of',
'mass.', 'Bit', 'like', 'our', 'ship', '&amp;', 'booster', 'on', 'reentry.'], ['Same'], [], [], ['Probably', 'sligh
tly', 'further', 'forward,', 'smaller,', 'more', 'inward.', 'No', 'funny', 'looking', 'static', 'aero', 'at', 'top,
', 'as', 'static', 'aero', 'no', 'longer', 'directly', 'in', 'flow.'], ['Btw,', 'theres', 'a', 'slight', 'error', '
with', 'forward', 'flap', 'design.', 'Moving', 'section', 'is', 'needed', 'for', 'control,', 'but', 'passive', 'sec
tion', 'is', 'counter-productive,', 'as', 'it', 'pushes', 'nose', 'backwards.', 'New', 'design', 'rotates', 'fwd',
'flaps', 'more', 'to', 'leeward', '&amp;', 'further', 'forward', 'to', 'improve', 'moment', 'arm.', 'Maybe', '~120'
, 'deg', 'apart.'], ['Product', 'ideas!'], ['True'], ['You', 'can', 'make', 'anything', 'fly', 'haha'], ['Yes'], ['
Impressive'], ['Robyn', 'is', 'great'], ['Beta', 'or', 'maybe', '.', 'Going', 'to', 'pure', 'vision', 'set', 'us',
'back', 'initially.', 'Vision', 'plus', '(coarse)', 'radar', 'had', 'us', 'trapped', 'in', 'a', 'local', 'maximum,'
, 'like', 'a', 'level', 'cap.Pure', 'vision', 'requires', 'fairly', 'advanced', 'real-world', 'AI,', 'but', 'thats'
, 'how', 'our', 'whole', 'road', 'system', 'is', 'designed', 'to', 'work:', 'NNs', 'with', 'vision.'], [',', 'proba
```

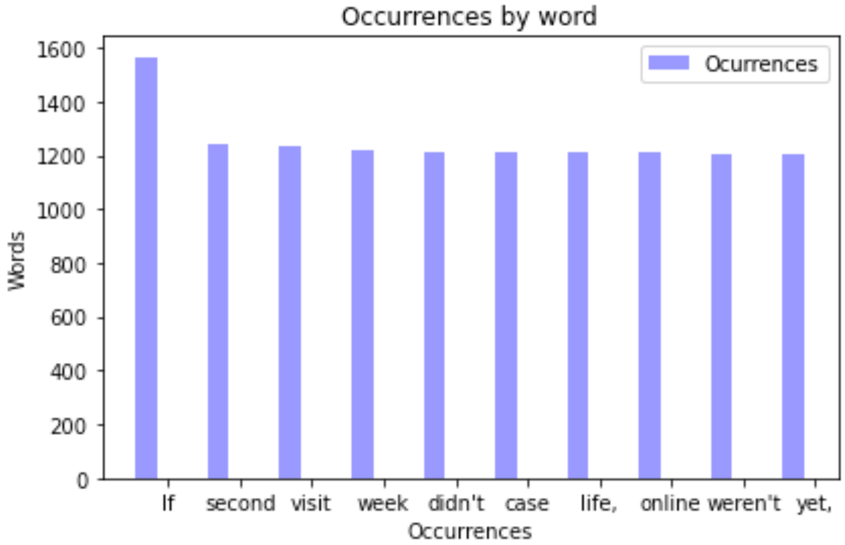
4. Explore the data

4.1 Frequent words analysis
After successfully tokenize all the twitter data, the first thing I come up with is to do a frequent words analysis and see if there is anything special or interesting of the #Dogecoin discussion and Elon Musk's tweets. Below is the frequent analysis result for Elon Musk.



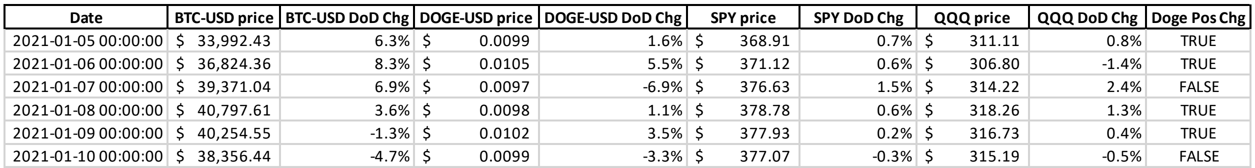
As we can see above, after filtering out the most common words, Tesla is the most common words stand out from Elon Musk's tweets, this makes perfect sense. Besides Tesla, we also see words like launch, Starship and Falcon in the Top 15 list. Out of my expectation, Dogecoin is not in the top 15 list. Elon Musk has mentioned Doge 46 times in his tweets, and it ranged around 30 in the full ranking list.

Comparing with Elon Musk's tweets, #Dogecoin tweets are having a different pattern. The #Dogecoin posters are more focusing on their imagination that if something happened, then they will do something.



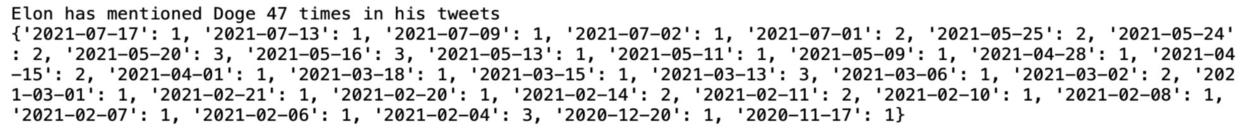
4.2 Basic Data Analysis

As I mentioned in the previous sections, I decided to divide the available data into 3 different segments, and investigate their impact on the daily price movement of dogecoin. The first and the most import part of this analysis is the macro economy independent variable including Bitcoin-USD price, SP500 Index and NASDAQ Index. Since the dogecoin price is heavily rely on investors' interests and cashflow conditions. Macro Economy condition will be included in the model drivers.



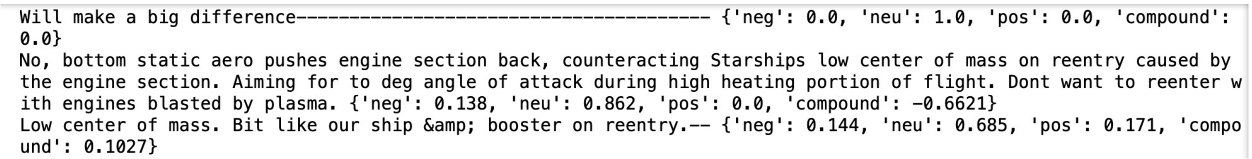
Date	BTC-USD price	BTC-USD DoD Chg	DOGE-USD price	DOGE-USD DoD Chg	SPY price	SPY DoD Chg	QQQ price	QQQ DoD Chg	Doge Pos Chg
2021-01-05 00:00:00	$ 33,992.43	6.3%	$ 0.0099	1.6%	$ 368.91	0.7%	$ 311.11	0.8%	TRUE
2021-01-06 00:00:00	$ 36,824.36	8.3%	$ 0.0105	5.5%	$ 371.12	0.6%	$ 306.80	-1.4%	TRUE
2021-01-07 00:00:00	$ 39,371.04	6.9%	$ 0.0097	-6.9%	$ 376.63	1.5%	$ 314.22	2.4%	FALSE
2021-01-08 00:00:00	$ 40,797.61	3.6%	$ 0.0098	1.1%	$ 378.78	0.6%	$ 318.26	1.3%	TRUE
2021-01-09 00:00:00	$ 40,254.55	-1.3%	$ 0.0102	3.5%	$ 377.93	0.2%	$ 316.73	0.4%	TRUE
2021-01-10 00:00:00	$ 38,356.44	-4.7%	$ 0.0099	-3.3%	$ 377.07	-0.3%	$ 315.19	-0.5%	FALSE

Elon Musk's frequent tweets about Dogecoin also helps make this cryptocurrency more famous around the world. In that case, one of the key indicator of model is whether Elon Musk has mentioned Dogecoin in his Twitter today. I use Python to read through all the tweets that Elon Musk mentioned about Dogecoin, and record them into a dictionary that when did he mentioned Dogecoin and how many time on that day did he mention it, and the output looks like below:



```
Elon has mentioned Doge 47 times in his tweets
{'2021-07-17': 1, '2021-07-13': 1, '2021-07-09': 1, '2021-07-02': 1, '2021-07-01': 2, '2021-05-25': 2, '2021-05-24'
: 2, '2021-05-20': 3, '2021-05-16': 3, '2021-05-13': 1, '2021-05-11': 1, '2021-05-09': 1, '2021-04-28': 1, '2021-04
-15': 2, '2021-04-01': 1, '2021-03-18': 1, '2021-03-15': 1, '2021-03-13': 3, '2021-03-06': 1, '2021-03-02': 2, '202
1-03-01': 1, '2021-02-21': 1, '2021-02-20': 1, '2021-02-14': 2, '2021-02-11': 2, '2021-02-10': 1, '2021-02-08': 1,
'2021-02-07': 1, '2021-02-06': 1, '2021-02-04': 3, '2020-12-20': 1, '2020-11-17': 1}
```

Besides Elon Musk, ordinary people's discussion about Dogecoin can also have impact on Dogecoin price. With more discussion about #Dogecoin, we are expecting the dogecoin price will have a big movement on that day. Here we built a sentiment analysis model with Valence Aware Dictionary and sEntiment Reasoner(VADER) to check what is people's reaction about the dogecoin, and based on the sentiment analysis result, we predict the Dogecoin's price movement. For each single day, we add all the compound reaction scores up to get a final sentiment result for that day, and we will use it as a key input for our predicting model. A sample sentiment result is shown below:



```
Will make a big difference--------------------------------------- {'neg': 0.0, 'neu': 1.0, 'pos': 0.0, 'compound':
0.0}
No, bottom static aero pushes engine section back, counteracting Starships low center of mass on reentry caused by
the engine section. Aiming for to deg angle of attack during high heating portion of flight. Dont want to reenter w
ith engines blasted by plasma. {'neg': 0.138, 'neu': 0.862, 'pos': 0.0, 'compound': -0.6621}
Low center of mass. Bit like our ship &amp; booster on reentry.-- {'neg': 0.144, 'neu': 0.685, 'pos': 0.171, 'compo
und': 0.1027}
```

After exploring the data availability and result of those 3 segments, I am comfortable to combine all the results into a Python pandas DataFrame and build model based on the numbers.

5. Model the Data

Initially, I am planning to try 3 different methodologies to model the Dogecoin price movements, that includes:

1. CNN with help of VADER(Valence Aware Dictionary and sEntiment Reasoner):
   This is the final approach I selected, the only issue with this approach is that since I am using daily data to forecast daily price movement of Dogecoin. Since Dogecoin came into people's attention only starting this beginning of the year, I am able to use about 180 days data to train the model, and another 40 days data to validate the result. Given the training and testing data size constraints, CNN might not be a good fit. After testing on all the feasible approach including Decision Tree, Logistic Regression and SVM, I finally decided to go with Logistic Regression given the time constraint and related research result.

2. LSTM or GRU:
   Both LSTM and GRU have been heavily utilized in the stock trading industry to predict the stock price movement. However, after reading through related paper and research, the structure of the model is fairly complicated, 180 training dataset might not be enough to train the model well.

3. Use 'Bidirectional Embedding Representations from Transformers'(BERT) to create sentiment analysis:
   Similar with approach 1, the only difference between 1 and 3 is utilizing different model to build the sentiment analysis. Since I didn't find a well-trained BERT model for language sentiment analysis, this method is not feasible. Another reason why I decided not to go with this approach is because for all the Twitter data I download, they are not labeled at all. This means I cannot train the model myself since it is too time consuming.

Given the reasons above, I finally decided to go with the first option: Logistic Regression with help of VADER.

6. Model Result and Analysis

6.1 Model Result
As previous section mentioned, the final modeling methodology for the Dogecoin price movement prediction is to utilize three groups of data: macro economy independent variable, Elon Musk's tweets and sentiment analysis result from Twitter users daily posts, build a logistic regression model.

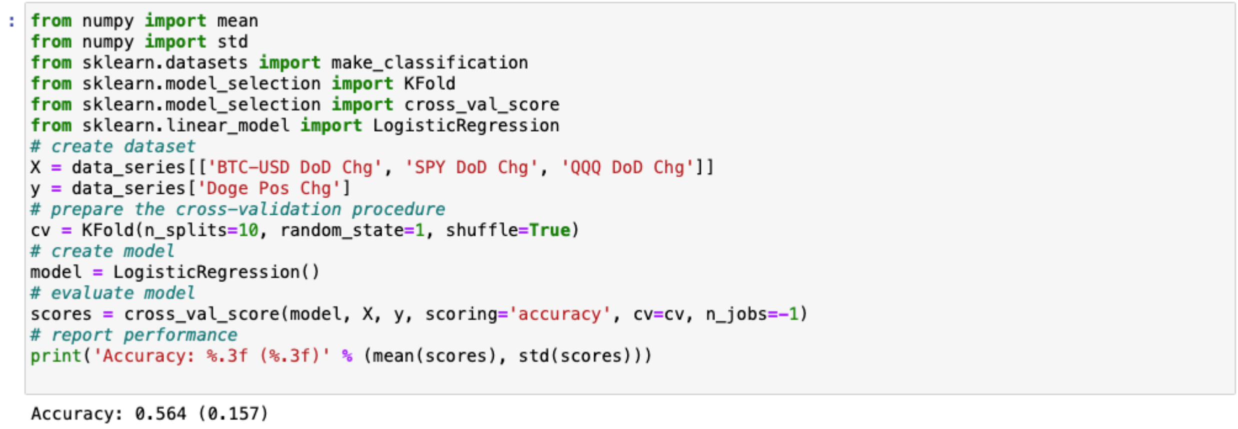
To better assess the impact of each of the potential components of the model, I arrange the testing data into four different groups, each of those groups has a little bit different model inputs, they are share the same macro economy independent variables, but they might not have Elon Musk's Twitter effect or Sentiment Analysis result. The four groups are:

1. Model with Elon Musk's Twitter Effect and Sentiment Analysis
2. Model with Elon Musk's Twitter Effect but no Sentiment Analysis
3. Model with Sentiment Analysis but no Elon Musk's Twitter Effect
4. Model without Sentiment Analysis and Elon Musk's Twitter Effect.

The model input comparison is shown in the table below:

Variable	Group 1	Group 2	Group 3	Group 4
With Elon Musk Twitter effect	Yes	Yes	No	No
With Sentiment Analysis	Yes	No	Yes	No

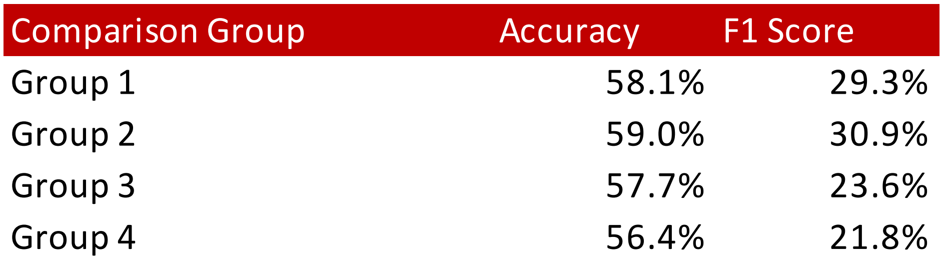
Then we will utilize the sklearn package in Python to build the model, to make sure the data works well, I used 10-fold cross validation to decrease the randomness impact of the data input to the result. The working code is show below:



```
from numpy import mean
from numpy import std
from sklearn.datasets import make_classification
from sklearn.model_selection import KFold
from sklearn.model_selection import cross_val_score
from sklearn.linear_model import LogisticRegression
# create dataset
X = data_series[['BTC-USD DoD Chg', 'SPY DoD Chg', 'QQQ DoD Chg']]
y = data_series['Doge Pos Chg']
# prepare the cross-validation procedure
cv = KFold(n_splits=10, random_state=1, shuffle=True)
# create model
model = LogisticRegression()
# evaluate model
scores = cross_val_score(model, X, y, scoring='accuracy', cv=cv, n_jobs=-1)
# report performance
print('Accuracy: %.3f (%.3f)' % (mean(scores), std(scores)))
```

```
Accuracy: 0.564 (0.157)
```

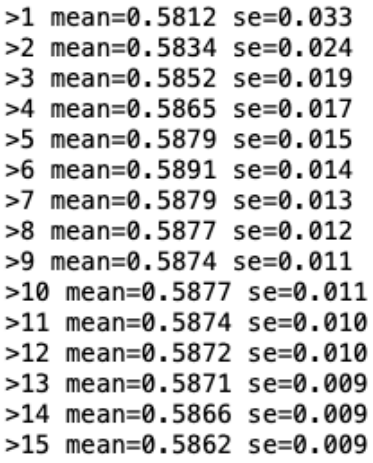
Based on this code, every time I only need to change the input X and the scoring methods to test the model performance. After built the model, I tested the result on the testing dataset and I get a model metrics comparison table as below:



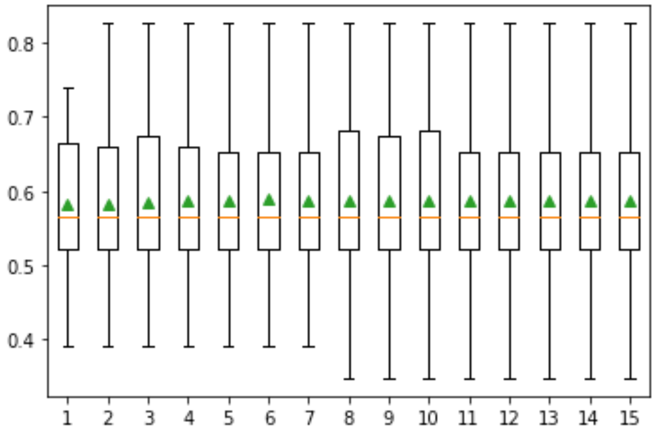
Comparison Group	Accuracy	F1 Score
Group 1	58.1%	29.3%
Group 2	59.0%	30.9%
Group 3	57.7%	23.6%
Group 4	56.4%	21.8%

Based on the model result, the best model performance is group 2, which is with Elon Musk's tweets but no sentiment analysis result. And the model won second place is the one with both Elon Musk's tweets and Sentiment Analysis. In the following second, I will compare the model performance in details and explain why the model without Sentiment Analysis can win the title and compare the actual performance of the model if we would like to use this model for trading Dogecoin in real world.

Prior to test model on the real trading days, I also make sure the model result is making sense. To do that, I have ran 15 epochs of model training process and compared their result. Here are one of the result I am able to get. Based on the boxplot observation, no matter how I choose the testing samples, the mean accuracy is stable and standard error is within the acceptable range.



```
>1 mean=0.5812 se=0.033
>2 mean=0.5834 se=0.024
>3 mean=0.5852 se=0.019
>4 mean=0.5865 se=0.017
>5 mean=0.5879 se=0.015
>6 mean=0.5891 se=0.014
>7 mean=0.5879 se=0.013
>8 mean=0.5877 se=0.012
>9 mean=0.5874 se=0.011
>10 mean=0.5877 se=0.011
>11 mean=0.5874 se=0.010
>12 mean=0.5872 se=0.010
>13 mean=0.5871 se=0.009
>14 mean=0.5866 se=0.009
>15 mean=0.5862 se=0.009
```



6.2 Model Analysis

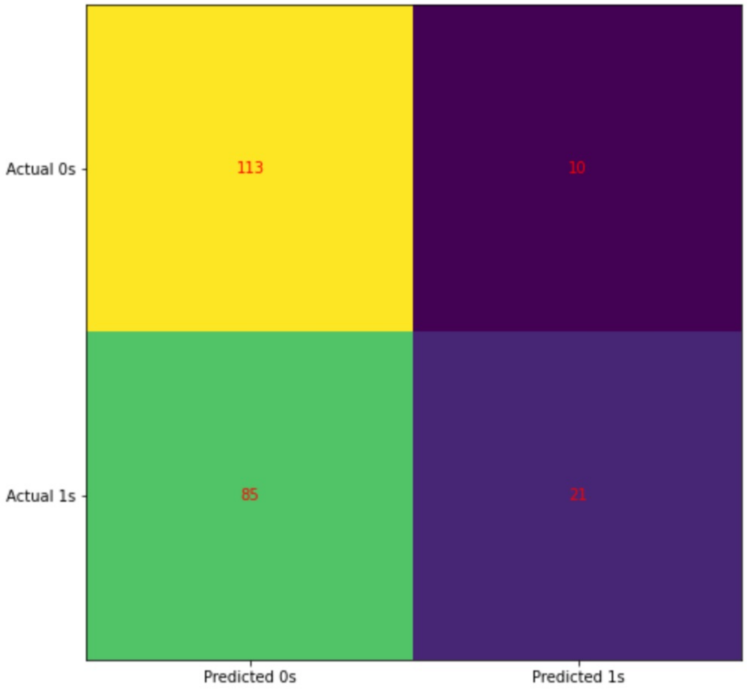
Based on section 6.1, the best candidate models are Group 1: With Elon's Tweets and Sentiment Analysis result and Group 2: With Elon's Tweets and No Sentiment Analysis result. In this section, I will mainly focus on the actual performance of those 2 models, compare their model performance and decide which is the best candidate model.

Based on the F1 score and model accuracy, group 2 model has slight advantage. And then, I build the confusion matrix for those 2 models. Below are the confusion matrix result:

Group 1: With Elon's Tweets and Sentiment Analysis

$$PredictedProbabilityofDogecoinPriceGoesHigh = -0.250+2.273*BTC-USDDoDChg + 0.079 * SPYDoDChg + 0.063 * QQQDoDChg + 0.446 * ElonMentioned + 0.704 * DogecoinSentiments$$

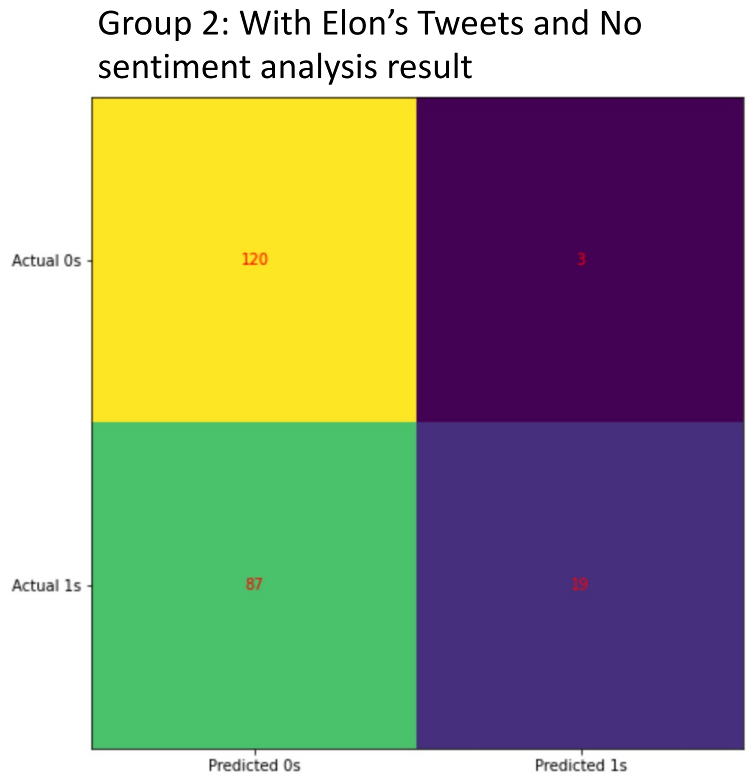
Group 1: With Elon's Tweets and Sentiment analysis result



From the confusion matrix above, we can see that the model testing dataset is relatively balanced, 0 and 1 are all having 50% of the total points. A significant observation is that, comparing with the number of Predicted 0s, Predicted 1s is much less. I believe this is driven by the event that Elon Musk has only mentioned Dogecoin 47 times in his tweets, and this might be a key driver of the model to predict 1 or 0. And it also explains why the model performance is not as good as others while we don't include Elon Mentioned Tweets in our model. Along the 31 predicted 1s, there are 10 cases of false positive, this means that we are having about 30% cases to lose money if we implement this model in our trading strategy.

Group 2: With Elon’s Tweets and without Sentiment Analysis

$$PredictedProbabilityofDogecoinPriceGoesHigh = -0.46 + 2.281 * BTC-USDDoDChg + 0.081 * SPYDoDChg + 0.065 * QQQDoDChg + 0.471 * ElonMentioned$$



Comparing with the Group 1, Group 2 has similar False Negative rate, but its False Positive rate is 3 out of 22, which is significantly lower than Group 1. This model is also heavily rely on Elon Musk’s tweets event, that explains why the model prediction is so conservative.

Reason why Sentiment Analysis does work well on the model:
As we see above, the difference of Group 1 and Group 2 model is one has sentiment analysis result but the other doesn’t. And Group 2 without sentiment analysis is even having a better performance.

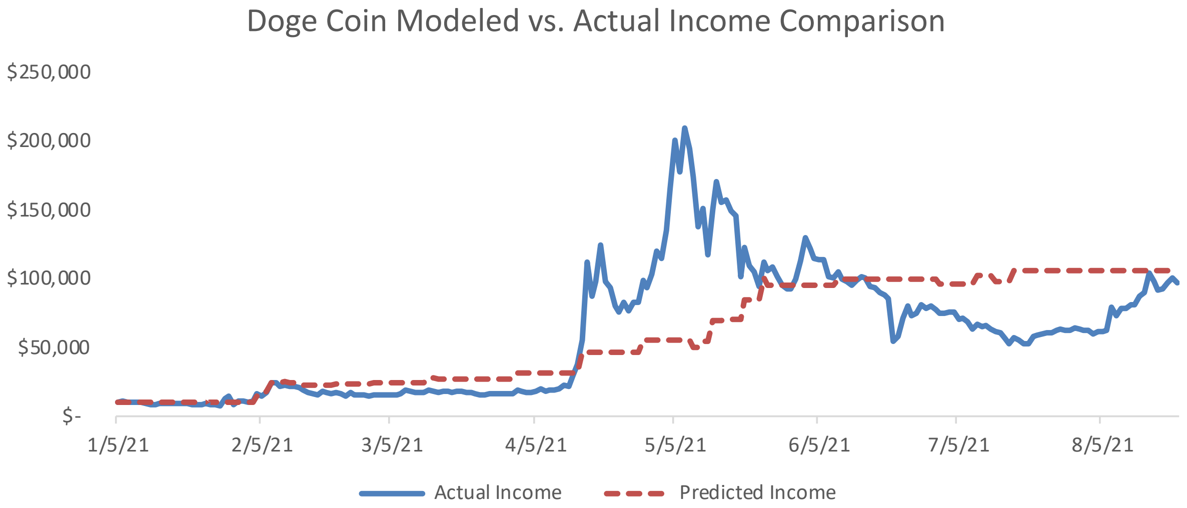
Based on my understanding, one of the key reason why this happens is that not everyone on Twitter publish their post rationally. People tend to share their good experience and result, and if they lose money they will hide from the back and do not say anything.

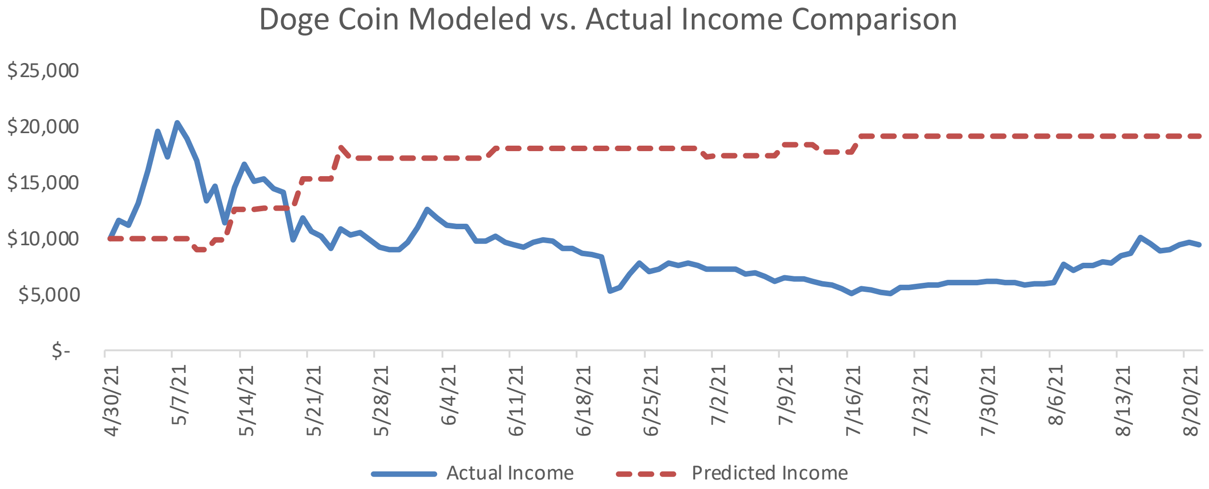
That’s why I got all positive sentiment result from more than 200 days of the twitter data, this might mislead the model to make a false prediction.

6.3 Model Performance Comparison

The most important way to decide a model works or not is to see its actual performance. Here I do a back-test run starting from 1/4/2021 until 8/19/2021, another run from 5/1/2021 to 8/19/2021. The reason to choose 5/1/2021 is because Dogecoin reach its historical top on that date, and followed with a 75% retreat. I want to assess how well the model performs when the market condition is not that good. And for some conservative investors, this will help them to make sure their money is safe. I initialize the investment about to be $10,000 and see how much the model can earn after a certain period.

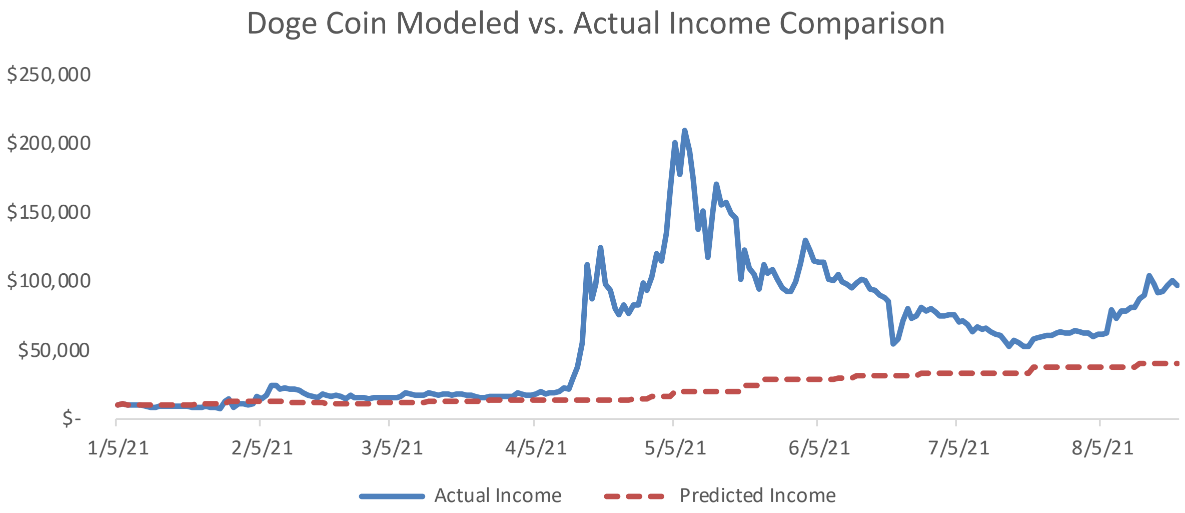
Group 1:

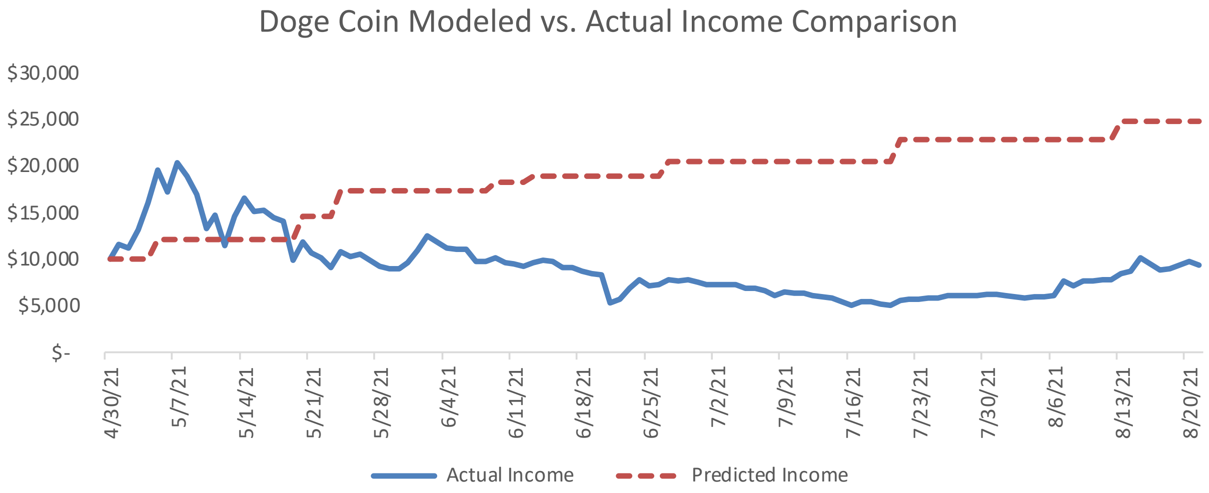

Doge Coin Modeled vs. Actual Income Comparison


Doge Coin Modeled vs. Actual Income Comparison

The Group 1 model received 31 buy signal in total, and it has about 1,000% revenue after the 8-month period. Comparing with the actual Dogecoin price movement, the model works almost the same as holding the Dogecoin. If our model works starting from historical high, the model can earn about 100% revenue until now, this is much better performance than holding the Dogecoin while holding dogecoin haven't recover to the historical high.

Group 2:





Group 2 model performance is quite different with Group 1. If we invest our money on the first day of 2021, the model can earn 500% of revenue, which is about 50% less than Group 1 model or holding Dogecoin through time. From this perspective, Group 2 doesn’t work as good as Group 1.

On the other hand, if we invest our money from the historical high of Dogecoin, we can earn about 250% revenue with Group 2 model, which is 25% better than Group 1. This also outperforms the actual holding the Dogecoin about 150%.

7. Final Model Conclusion and Further Enhancement

7.1 Model Conclusion

Based on Section 6's detailed result and analysis, both Group 1 and Group 2 has their own advantages and shortcomings. Both Group 1 and Group 2 shows capability of forecasting Dogecoin price movement trend. Based on different investor needs, different model will be selected for investment.

1. If the investor want to maximize their potential gain and do not care about the potential risk, Group 1 model will be preferred. It will maximize the potential revenue without holding the Dogecoin every day. It will be a better option than long term holding.
2. If the investor is risk-averse, Group 2 model will be selected to make investments. Group 2 model has less false positive events this means the probability of losing money is much less than Group 1. And the investment itself has much higher gain than holding Dogecoin in a long term period. The shortcoming of this strategy is that the potential maximum gain will be less than Group 1.

7.2 Model Further enhancement

I have to admit that from the model dependency and forecasting power wise, there are still improvements needed. Here are some of the areas I can think of that needs further improvements:

1. Sentiment Analysis data from Twitter is not complete dataset, I only used 2 hours of volume each day due to API volume constraints. And among the 2-hour time periods, it is more for midnight discussion, in this case I am missing useful discussions during day time or stock trading time. It will have a huge impact on the sentiment analysis result and further impact the model performance.
2. Data cleaning process is not perfect, there are potentially some advertisements remaining in the dataset. These advertisements or irrelevant tweets might influence the result of sentiment analysis just like the previous point.
3. I only includes 4 different independent variables in the model, there are potentially other independent variables that might have an effect on the Dogecoin price movement like stock/Bitcoin trading volume in prior day or some certain stock price movement indicator. In short, we can add more to enhance model performance.
4. Previously, I am planning to do a sentiment analysis based on Weibo discussion on Dogecoin in parallel. However, since I cannot find an equivalent VADER scoring model for Chinese, this initiative is currently on hold. But still I believe adding Chinese sentiment analysis to the model and potentially enhance the model performance.

Reference

- https://medium.com/swlh/analyzing-the-wall-street-bets-reddit-group-with-natural-languageprocessing-296465f90f26
- https://galhever.medium.com/sentiment-analysis-with-pytorch-part-1-data-preprocessinga51c80cc15fb
- https://web.stanford.edu/class/cs224n/reports/final_reports/report030.pdf
- https://www.kaggle.com/ragnisah/text-data-cleaning-tweets-analysis
- https://stackoverflow.com/questions/64719706/cleaning-twitter-data-pandas-python
- https://towardsdatascience.com/building-a-logistic-regression-in-python-step-by-step-becd4d56c9c8
- https://towardsdatascience.com/tweepy-for-beginners-24baf21f2c25
- https://www.earthdatascience.org/courses/use-data-open-source-python/intro-to-apis/twitter-data-in-python/