Sentiment Analysis of Web Data to Predict Price Fluctuation of Cryptocurrencies

Jainam Jhaveri

Neha Sharma

Rushabh Vakharia

Sumesh Vijayan

Prof. Rong Liu

INTRODUCTION

Volatility is one of the most complex characteristics of the cryptocurrency market since it brings a great amount of risk to investors. Yet, it's also what opens opportunity for profitable investment for the ones that understand how it works. There are several factors that influence cryptocurrency market volatility. But one of the hardest to understand and anticipate is human sentiments.

On a different end, micro-blogging has become a popular communication tool, and it can aggregate opinions of a lot of people with different backgrounds located in any place and time. This information has been used in both academia and industry to obtain precious feedback from massive crowds with a lower cost and at a faster rate than surveys. Therefore, sentiment analysis is becoming a powerful decision-making tool nowadays. Before investing in a digital currency, one can leverage the opinions of the people about that currency to find out where it stands.

Since the last few years, cryptocurrency market predictors have yearned to know the future market and with the advent of machine learning technologies, their job has become quite simple. When we join hands of technology such as twitter and other websites on the internet to a domain such as cryptocurrency market analysis, we can discover the various investing and growth trends of the currencies. Our project wants to explore the relationship between people's impressions towards digital currencies and their variations.

MOTIVATION & OBJECTIVES

As the world's first completely decentralized digital payment system, Bitcoin represents a revolutionary phenomenon in financial markets. Following the path of Bitcoin, other cryptocurrencies like Litecoin and Ethereum also give customers the option to divulge in the digital currency market. Though existing only in digital form, the rise of these currencies is unstoppable. Amidst this entire hype, there is a growing concern focused on cryptocurrencies' considerable price volatility and their associated risks.

The cryptocurrency market is very unregulated and is driven by demand and supply. People usually say, write or tweet things which they are interested or disinterested in on social media. There exists a significant impact of social media on future cryptocurrency returns. The web offers substantial information about these cryptocurrencies' acceptance among the public, as well as daily fluctuations in their market sentiments.

In short, our objective is to find out sentiments of tweets and news headlines associated to a cryptocurrency and relate those sentiments to the market return of the currency.

RELATED WORK

- A lot of work has previously been done on the same topic by a wide number of audience.
- In January 2018, an article was published on “The Startup” which spoke about how the Bitcoin prices can be predicted using historical price, volume and social hype.
- Several papers also have been published which talk about the effects of social media on cryptocurrency pricing.
- Two of the papers are “Cryptocurrency price prediction using news and social media sentiment” and “The Impacts of Social Media on Bitcoin Performance”.

WHAT’S NEW

- One of the papers mentioned above used traditional supervised learning algorithms like Linear SVM and Naïve Bayes for text-based sentiment classification.
- In our case, we are clustering the news headlines using the Latent Dirichlet Allocation unsupervised learning algorithm. These clusters are then used to predict the prices of cryptocurrencies with the help of Linear/Logistic Regression and historic price data fetched from Kaggle.
- We will be performing analysis for only three cryptocurrencies viz. Bitcoin, Litecoin and Ethereum.
- We also used the Quandl API to show the price trends of cryptocurrencies. (The Cryptocurrency Price Analysis code will be attached in the zip file. The code was referenced from https://blog.patricktriest.com/analyzing-cryptocurrencies-python/)

METHODOLOGY

Data

1. For each of the three cryptocurrencies, we will be scraping news headlines from the following sites:

    a. news.bitcoin.com
    b. bloomberg.com
    c. litecoinnews.io
    d. coindesk.com

2. Along with the headlines, we will also be scraping the date of their publication.
    a. For fetching tweets regarding these currencies, we will be using the Twitter API.

3. Once the data is scraped and extracted, we will filter out the irrelevant data.

4. Next, we will use VADER to classify the headlines and TEXTBLOB to classify the tweets as positive, negative or neutral.

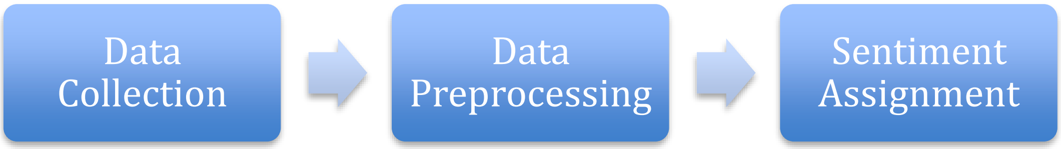
5. After classifying sentiments, we will be tokenizing, lemmatizing and removing the stop words from the data we have and perform clustering on the data using the LDA (Latent Dirichlet Allocation) unsupervised learning algorithm. Since LDA clusters documents based on token similarity, having significant repetition of those keywords in almost every headline skewed the clustering performance.

6. Initially, we decided to cluster headlines into 5 clusters and identified some primary topics into which we wanted to cluster the data. Eventually, we attempted clustering the data into 4 clusters so that we could compare the price prediction performance of each.

7. Price prediction was carried out by running regressions with article sentiment and cluster as the independent variables, and price movement (% change) as the dependent variable.

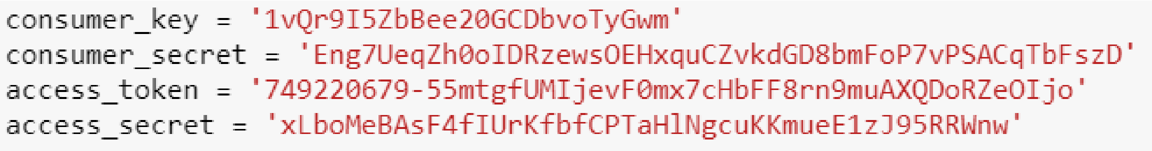
8. First a multiple linear regression was run to predict the magnitude of price movements, which did not have accurate results. Later, logistic regression was run to simply predict whether cryptocurrency prices would move up or down during a given time period. The accuracy for this model was much greater. The time intervals that were analyzed for price movements were 1 day and 3-day.

Twitter Data Collection & Sentiment Analysis



<u>Code Flow</u>

Step 1: Initialize the keys and tokens to access the Twitter API



```
consumer_key = '1vQr9I5ZbBee20GCDbvoTyGwm'
consumer_secret = 'Eng7UeqZh0oIDRzewsOEHxquCZvkdGD8bmFoP7vPSACqTbFszD'
access_token = '749220679-55mtgfUMIjevF0mx7cHbFF8rn9muAXQDoRZeOIjo'
access_secret = 'xLboMeBAsF4fIUrKfbfCPTaHlNgcuKKmueE1zJ95RRWnw'
```

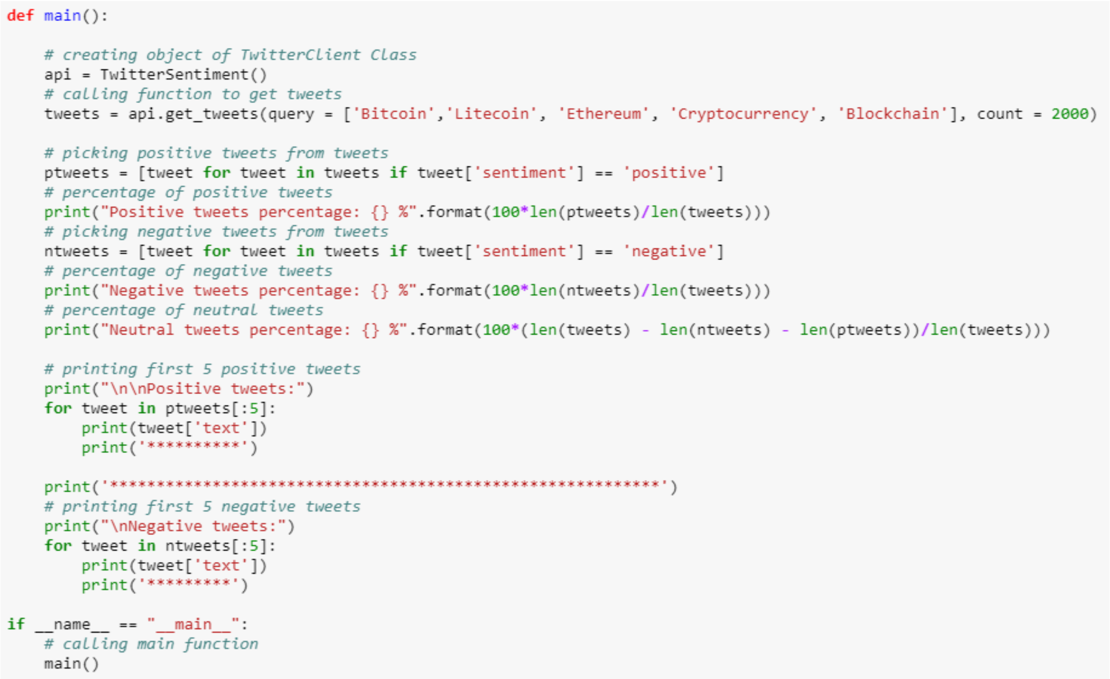
Step 2: Create a class TwitterSentiment

- A class is created with 3 functions defined for 3 individual tasks.

- clean_tweet(), get_tweet_sentiment(), get_tweets().

Step 3: Trigger Data collection and Preprocessing

- Using the twitter API, 2000 tweets are collected, cleaned and assigned a sentiment (positive, negative or neutral) using the Textblob package.
- They are then stored in a dictionary with their sentiment and text.



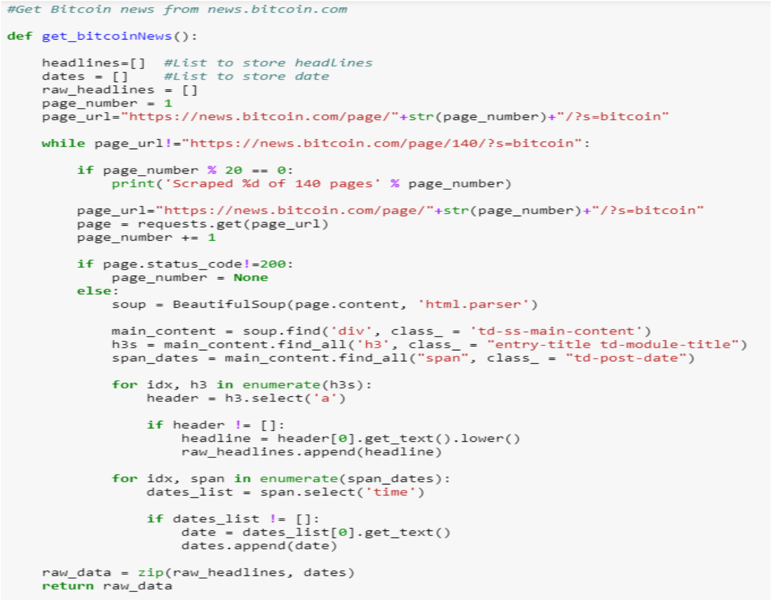
```
def main():

    # creating object of TwitterClient Class
    api = TwitterSentiment()
    # calling function to get tweets
    tweets = api.get_tweets(query = ['Bitcoin','Litecoin', 'Ethereum', 'Cryptocurrency', 'Blockchain'], count = 2000)

    # picking positive tweets from tweets
    ptweets = [tweet for tweet in tweets if tweet['sentiment'] == 'positive']
    # percentage of positive tweets
    print("Positive tweets percentage: {} %".format(100*len(ptweets)/len(tweets)))
    # picking negative tweets from tweets
    ntweets = [tweet for tweet in tweets if tweet['sentiment'] == 'negative']
    # percentage of negative tweets
    print("Negative tweets percentage: {} %".format(100*len(ntweets)/len(tweets)))
    # percentage of neutral tweets
    print("Neutral tweets percentage: {} %".format(100*(len(tweets) - len(ntweets) - len(ptweets))/len(tweets)))

    # printing first 5 positive tweets
    print("\n\nPositive tweets:")
    for tweet in ptweets[:5]:
        print(tweet['text'])
        print('**********')

    print('***********************************************************')
    # printing first 5 negative tweets
    print("\nNegative tweets:")
    for tweet in ntweets[:5]:
        print(tweet['text'])
        print('*********')

if __name__ == "__main__":
    # calling main function
    main()
```

Web Data Collection & Conversion

<u>Code Flow</u>

Step 1: Create functions to scrape web pages using BeautifulSoup and Selenium

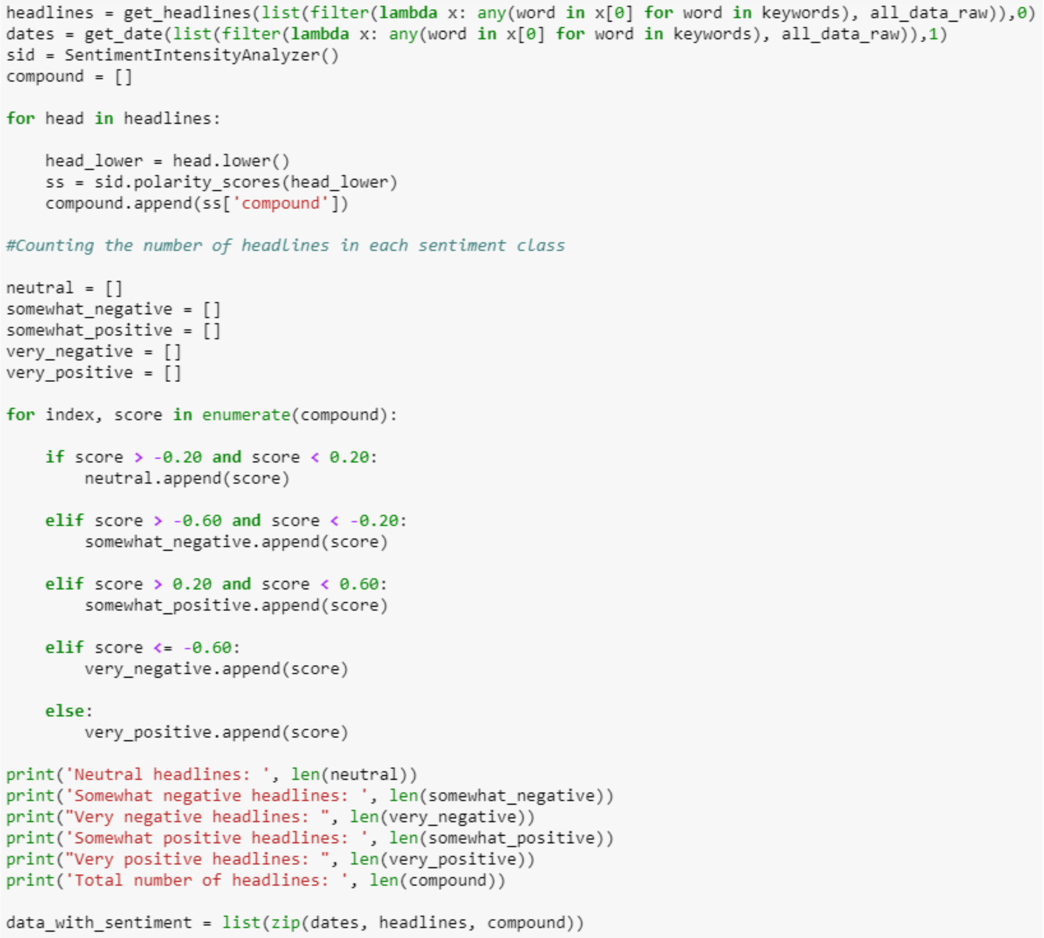
- For each of the three currencies, a separate data gathering and processing file is created.
- A function is created for fetching the news headlines and the corresponding publication dates.



```python
#Get Bitcoin news from news.bitcoin.com

def get_bitcoinNews():

    headlines=[]  #List to store headlines
    dates = []    #List to store date
    raw_headlines = []
    page_number = 1
    page_url="https://news.bitcoin.com/page/"+str(page_number)+"/?s=bitcoin"

    while page_url!="https://news.bitcoin.com/page/140/?s=bitcoin":

        if page_number % 20 == 0:
            print('Scraped %d of 140 pages' % page_number)

        page_url="https://news.bitcoin.com/page/"+str(page_number)+"/?s=bitcoin"
        page = requests.get(page_url)
        page_number += 1

        if page.status_code!=200:
            page_number = None
        else:
            soup = BeautifulSoup(page.content, 'html.parser')

            main_content = soup.find('div', class_ = 'td-ss-main-content')
            h3s = main_content.find_all('h3', class_ = "entry-title td-module-title")
            span_dates = main_content.find_all("span", class_ = "td-post-date")

            for idx, h3 in enumerate(h3s):
                header = h3.select('a')

                if header != []:
                    headline = header[0].get_text().lower()
                    raw_headlines.append(headline)

            for idx, span in enumerate(span_dates):
                dates_list = span.select('time')

                if dates_list != []:
                    date = dates_list[0].get_text()
                    dates.append(date)

    raw_data = zip(raw_headlines, dates)
    return raw_data
```

Step 2: Perform sentiment analysis using VADER

- These two values are then attached to a list as a tuple and sentiment analysis is performed on that data using VADER.

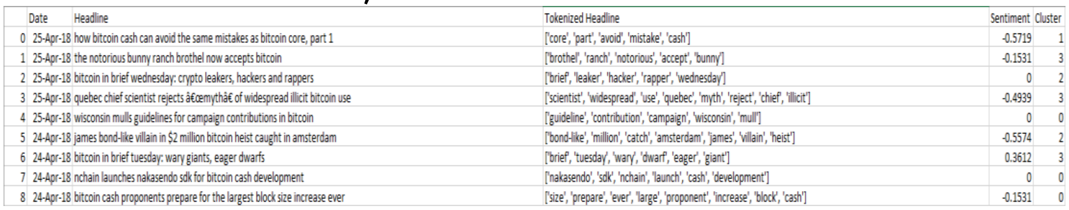


```python
headlines = get_headlines(list(filter(lambda x: any(word in x[0] for word in keywords), all_data_raw)),0)
dates = get_date(list(filter(lambda x: any(word in x[0] for word in keywords), all_data_raw)),1)
sid = SentimentIntensityAnalyzer()
compound = []

for head in headlines:

    head_lower = head.lower()
    ss = sid.polarity_scores(head_lower)
    compound.append(ss['compound'])

#Counting the number of headlines in each sentiment class

neutral = []
somewhat_negative = []
somewhat_positive = []
very_negative = []
very_positive = []

for index, score in enumerate(compound):

    if score > -0.20 and score < 0.20:
        neutral.append(score)

    elif score > -0.60 and score < -0.20:
        somewhat_negative.append(score)

    elif score > 0.20 and score < 0.60:
        somewhat_positive.append(score)

    elif score <= -0.60:
        very_negative.append(score)

    else:
        very_positive.append(score)

print('Neutral headlines: ', len(neutral))
print('Somewhat negative headlines: ', len(somewhat_negative))
print("Very negative headlines: ", len(very_negative))
print('Somewhat positive headlines: ', len(somewhat_positive))
print("Very positive headlines: ", len(very_positive))
print('Total number of headlines: ', len(compound))

data_with_sentiment = list(zip(dates, headlines, compound))
```

Step 3: Clean the data and use LDA clustering

- The data is then tokenized, lemmatized, made devoid of stop words and clustered.
- The list “data_for_clustering” is then passed for clustering process.
- In our case, we have created clusters of 5 and 4 to check which one gives better results.
- The results can be seen in the following files that are created:
  1) Cleaned_data_final_5clusters_btc.csv
  2) Cleaned_data_final_4clusters_btc.csv
  3) Cleaned_data_final_5clusters_ltc.csv
  4) Cleaned_data_final_4clusters_ltc.csv
  5) Cleaned_data_final_5clusters_eth.csv
  6) Cleaned_data_final_4clusters_eth.csv

- These files contain the date, headlines and the sentiment of each headline.



	Date	Headline	Tokenized Headline	Sentiment	Cluster
0	25-Apr-18	how bitcoin cash can avoid the same mistakes as bitcoin core, part 1	['core', 'part', 'avoid', 'mistake', 'cash']	-0.5719	1
1	25-Apr-18	the notorious bunny ranch brothel now accepts bitcoin	['brothel', 'ranch', 'notorious', 'accept', 'bunny']	-0.1531	3
2	25-Apr-18	bitcoin in brief wednesday: crypto leakers, hackers and rappers	['brief', 'leaker', 'hacker', 'rapper', 'wednesday']	0	2
3	25-Apr-18	quebec chief scientist rejects â€œmythâ€ of widespread illicit bitcoin use	['scientist', 'widespread', 'use', 'quebec', 'myth', 'reject', 'chief', 'illicit']	-0.4939	3
4	25-Apr-18	wisconsin mulls guidelines for campaign contributions in bitcoin	['guideline', 'contribution', 'campaign', 'wisconsin', 'mull']	0	0
5	24-Apr-18	james bond-like villain in $2 million bitcoin heist caught in amsterdam	['bond-like', 'million', 'catch', 'amsterdam', 'james', 'villain', 'heist']	-0.5574	2
6	24-Apr-18	bitcoin in brief tuesday: wary giants, eager dwarfs	['brief', 'tuesday', 'wary', 'dwarf', 'eager', 'giant']	0.3612	3
7	24-Apr-18	nchain launches nakasendo sdk for bitcoin cash development	['nakasendo', 'sdk', 'nchain', 'launch', 'cash', 'development']	0	0
8	24-Apr-18	bitcoin cash proponents prepare for the largest block size increase ever	['size', 'prepare', 'ever', 'large', 'proponent', 'increase', 'block', 'cash']	-0.1531	0

Step 4: Perform Linear/Logistic Regression

- On the clusters obtained, we perform linear regression to show predicted values.
- We use the daily price movement data from the csv files Bitcoin1Day, Bitcoin3Day, Litecoin1Day, Litecoin3Day, Ethereum1Day and Ethereum3Day.
- Also, logistic regression is used to help us predict if the prices will rise or fall.

Step 5: Create a file giving the actual and predicted returns

- The following files are created:
  “Actual_Predicted_3Day_4clust_btc.csv”
  “Actual_Predicted_3Day_5clust_btc.csv”
  “Actual_Predicted_1Day_4clust_btc.csv”
  “Actual_Predicted_1Day_5clust_btc.csv”
  “Actual_Predicted_3Day_4clust_ltc.csv”
  “Actual_Predicted_3Day_5clust_ltc.csv”
  “Actual_Predicted_1Day_4clust_ltc.csv”
  “Actual_Predicted_1Day_5clust_ltc.csv”
  “Actual_Predicted_3Day_4clust_eth.csv”
  “Actual_Predicted_3Day_5clust_eth.csv”
  “Actual_Predicted_1Day_4clust_eth.csv”

“Actual_Predicted_1Day_5clust_eth.csv”

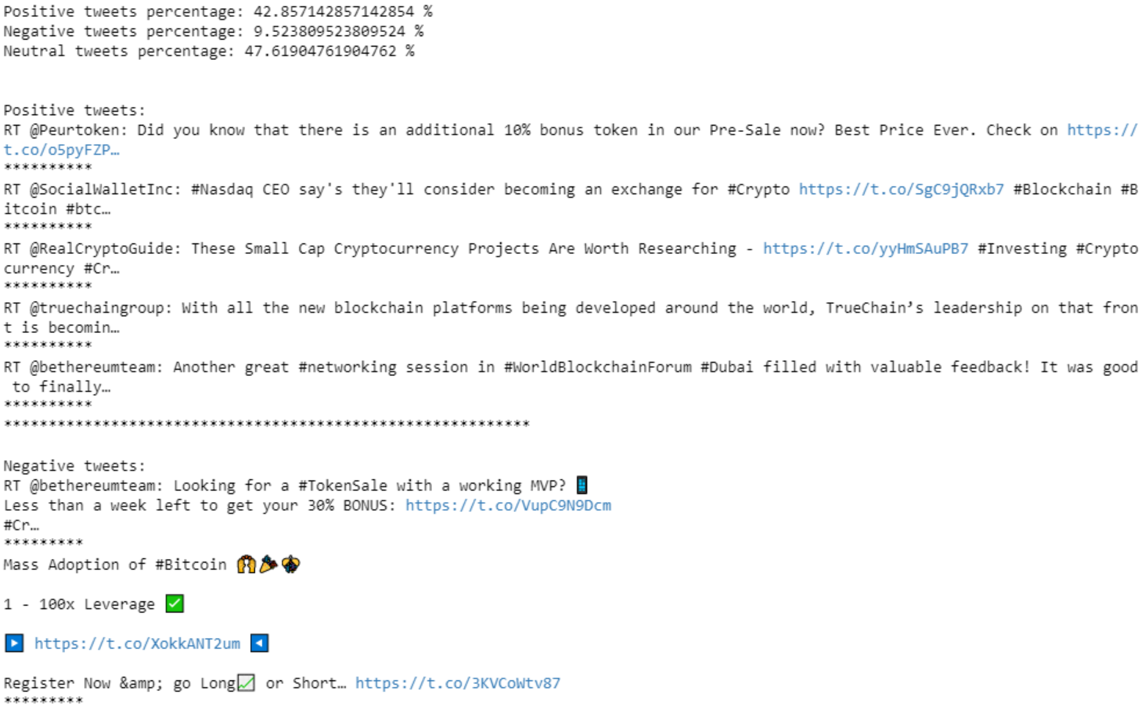
which show the actual and predicted returns.

- When we compare the sentiment of the headlines in the files created and compare the prices of the currencies on consecutive days, we can see that the results are quite satisfactory.

EXPERIMENT RESULTS ANALYSIS & CONCLUSION

Twitter Sentiment Results:

- We performed analysis on 2000 live tweets.
- We found that the data from twitter is extremely unorganized and unstructured.
- This makes analyzing twitter data very difficult.



```
Positive tweets percentage: 42.857142857142854 %
Negative tweets percentage: 9.523809523809524 %
Neutral tweets percentage: 47.61904761904762 %


Positive tweets:
RT @Peurtoken: Did you know that there is an additional 10% bonus token in our Pre-Sale now? Best Price Ever. Check on https://
t.co/o5pyFZP…
**********
RT @SocialWalletInc: #Nasdaq CEO say's they'll consider becoming an exchange for #Crypto https://t.co/SgC9jQRxb7 #Blockchain #B
itcoin #btc…
**********
RT @RealCryptoGuide: These Small Cap Cryptocurrency Projects Are Worth Researching - https://t.co/yyHmSAuPB7 #Investing #Crypto
currency #Cr…
**********
RT @truechaingroup: With all the new blockchain platforms being developed around the world, TrueChain’s leadership on that fron
t is becomin…
**********
RT @bethereumteam: Another great #networking session in #WorldBlockchainForum #Dubai filled with valuable feedback! It was good
 to finally…
**********
************************************************************

Negative tweets:
RT @bethereumteam: Looking for a #TokenSale with a working MVP? 📱
Less than a week left to get your 30% BONUS: https://t.co/VupC9N9Dcm
#Cr…
*********
Mass Adoption of #Bitcoin 🙌🎉🍾

1 - 100x Leverage ✅

▶️ https://t.co/XokkANT2um ◀️

Register Now &amp; go Long📈 or Short… https://t.co/3KVCoWtv87
*********
```

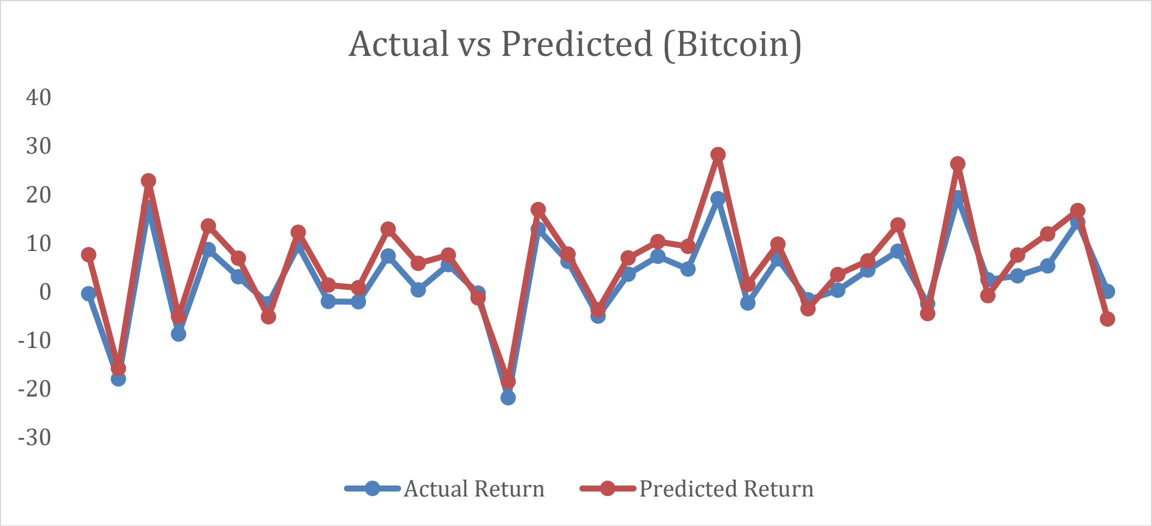
Linear Regression Results:

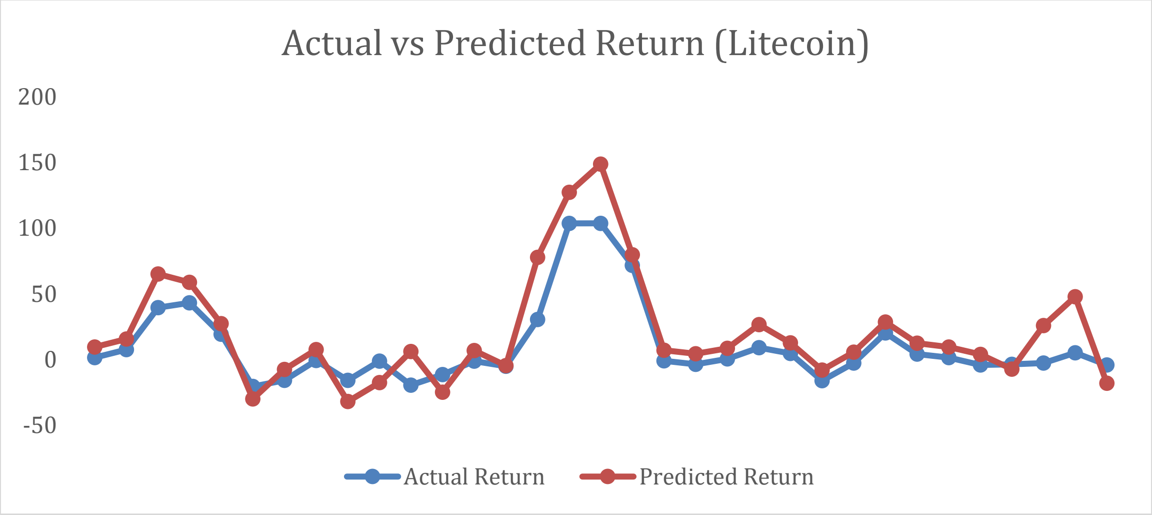
- On analyzing the R-squared values for linear regression performed on Bitcoin for all 4 model types, we found the following results:
  a. 1 day 5 clusters: 0.041
  b. 1 day 4 clusters: 0.046
  c. 3 day 5 clusters: 0.063
  d. 3 day 4 clusters: 0.029

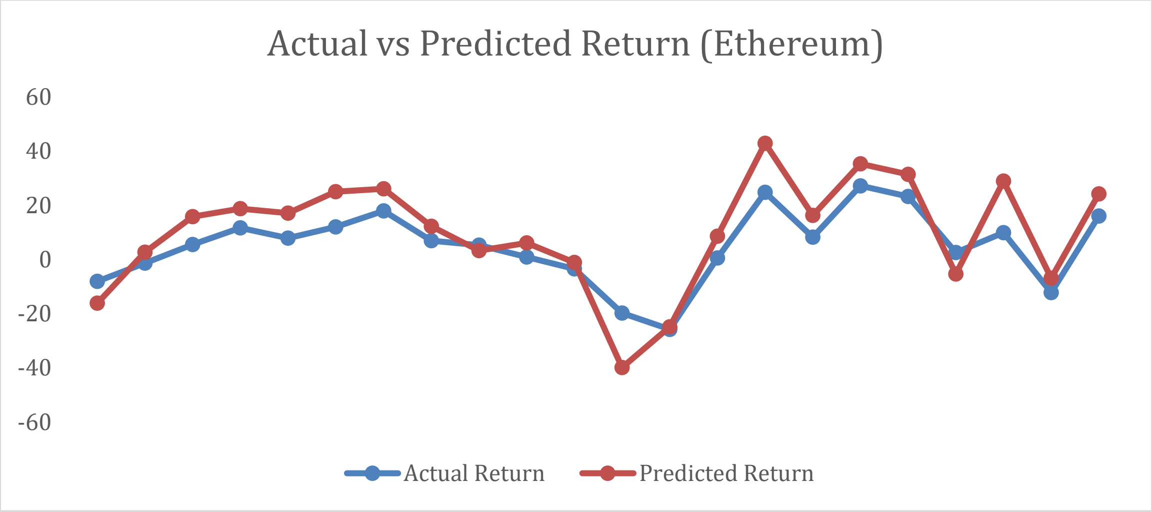
- On analyzing the R-squared values for linear regression performed on Litecoin for all 4 model types, we found the following results:
    a. 1 day 5 clusters: 0.151
    b. 1 day 4 clusters: 0.044
    c. 3 day 5 clusters: 0.114
    d. 3 day 4 clusters: 0.146

- On analyzing the R-squared values for linear regression performed on Ethereum for all 4 model types, we found the following results:
    a. 1 day 5 clusters: 0.155
    b. 1 day 4 clusters: 0.021
    c. 3 day 5 clusters: 0.413
    d. 3 day 4 clusters: 0.115

Model 3 (3 day 5 clusters) gives best performance in case of Bitcoin and Ethereum since the R-square value is the highest whereas for Litecoin, model 1 (1 day 5 clusters) gives better results.



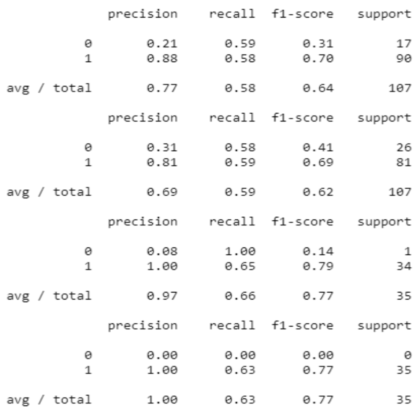




Logistic Regression Results:

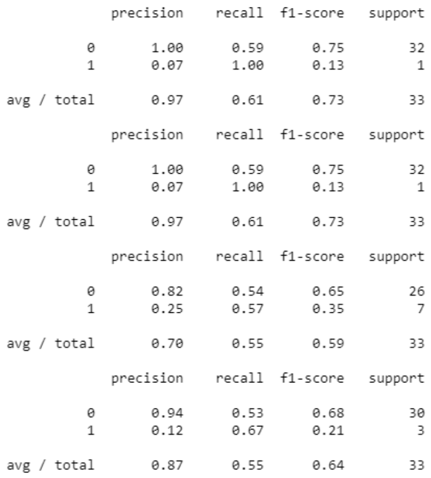
Model types in order:
1) 1 day 5 clusters
2) 1 day 4 clusters
3) 3 day 5 clusters
4) 3 day 4 clusters

Bitcoin Results:



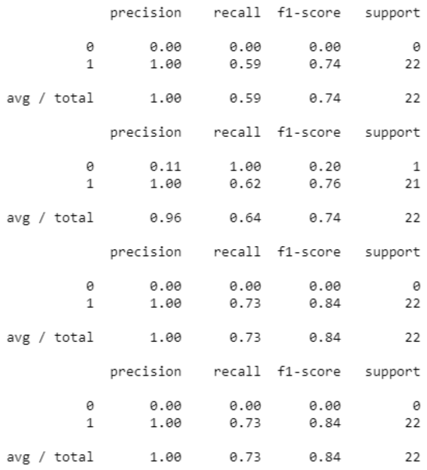
```
             precision    recall  f1-score   support

          0       0.21      0.59      0.31        17
          1       0.88      0.58      0.70        90

avg / total       0.77      0.58      0.64       107
             precision    recall  f1-score   support

          0       0.31      0.58      0.41        26
          1       0.81      0.59      0.69        81

avg / total       0.69      0.59      0.62       107
             precision    recall  f1-score   support

          0       0.08      1.00      0.14         1
          1       1.00      0.65      0.79        34

avg / total       0.97      0.66      0.77        35
             precision    recall  f1-score   support

          0       0.00      0.00      0.00         0
          1       1.00      0.63      0.77        35

avg / total       1.00      0.63      0.77        35
```

Litecoin Results:



```
             precision    recall  f1-score   support

          0       1.00      0.59      0.75        32
          1       0.07      1.00      0.13         1

avg / total       0.97      0.61      0.73        33
             precision    recall  f1-score   support

          0       1.00      0.59      0.75        32
          1       0.07      1.00      0.13         1

avg / total       0.97      0.61      0.73        33
             precision    recall  f1-score   support

          0       0.82      0.54      0.65        26
          1       0.25      0.57      0.35         7

avg / total       0.70      0.55      0.59        33
             precision    recall  f1-score   support

          0       0.94      0.53      0.68        30
          1       0.12      0.67      0.21         3

avg / total       0.87      0.55      0.64        33
```

Ethereum Results:



```
             precision    recall  f1-score   support

          0       0.00      0.00      0.00         0
          1       1.00      0.59      0.74        22

avg / total       1.00      0.59      0.74        22
             precision    recall  f1-score   support

          0       0.11      1.00      0.20         1
          1       1.00      0.62      0.76        21

avg / total       0.96      0.64      0.74        22
             precision    recall  f1-score   support

          0       0.00      0.00      0.00         0
          1       1.00      0.73      0.84        22

avg / total       1.00      0.73      0.84        22
             precision    recall  f1-score   support

          0       0.00      0.00      0.00         0
          1       1.00      0.73      0.84        22

avg / total       1.00      0.73      0.84        22
```

- Model 3 (3 day 5 clusters) gives best performance in case of Bitcoin and Ethereum since the F-score value is the highest whereas for Litecoin, model 1 (1 day 5 clusters) gives better results.

ANALYSIS

1) What part of the methodology worked (or didn't work)?

- Initially, we planned on using the Twitter data for our project.
- In due course, we found out that the data on Twitter was quite unorganized and extracting the kind of tweets we were looking for wasn't easy.
- So, we tried to concentrate more on the news headlines and that is where we think we showed better results.
- As far as Twitter is concerned, I am sure with more data and processing, the desired results could've been achieved but they wouldn't have been as good as those for news headlines since the latter were published by known publications and knowledgeable personnel as compared to tweets which are obviously from the public.

2) Why did the methodology work (or didn't work)?

- As mentioned above, the Twitter data wasn't as informative for the purpose of our project.
- The main reason was the unstructured format of data that Twitter provided.
- Some tweets had no real meaning although they spoke about the cryptocurrencies.
- This made it difficult for us to consider them in our project.

3) How to improve the methodology?

- Methodology can be improved by increasing the size of dataset that we have.
- With more data, the clustering results will improve significantly and ultimately regression results too will be better.

4) How to utilize the results? What business insights can be derived from the analysis?

- The output that regression provides could help the investors make informed investing decisions in cryptocurrencies.
- It would not be 100% true but it'll definitely help them cut their losses.
- Moreover, it will help them mitigate their investment risks in cryptocurrencies.

CONCLUSION

- The tests we ran yielded similar results with the 5-cluster model looking at the price in 3 days performing the best for Bitcoin and Ethereum.
- For Litecoin, the 5-cluster model looking at the price in 1 day performs the best.

- Proves that the price of cryptocurrencies is tied to the news coverage of the currency itself.

FUTURE WORK

- Consider using multi-classification for the clustering instead of fitting each article into only one cluster
- Include other inputs to the predictive model like technical price movement indicators.
    - Moving average
    - Bollinger Bands