Cryptocurrency Analysis and Forecasting

Group members:

1. Siddhesh Shaji,
2. Deepshika Reddy,
3. Yogesh Awdhut Gadade

Content

Introduction:

In the last decade, cryptocurrencies have taken up a very crucial role in the domain of international finance, attracting extensive media coverage, as well as the attention of several individual investors, academia, and the public in general. The total market cap for all the cryptocurrencies combined is at an estimated 800 billion dollars and the popularity is growing exponentially. Machine learning has been successful in predicting stock market prices through a host of different time series models, but its application in predicting cryptocurrency prices has been quite restrictive. The reason behind this is obvious as the prices of cryptocurrencies depend on a lot of factors like technological progress, internal competition, pressure on the markets to deliver, economic problems, security issues, political factors, etc. Their high volatility leads to the great potential of high profit if intelligent inventing strategies are taken. Unfortunately, due to their lack of indexes, cryptocurrencies are relatively unpredictable.

In this project, we will be going through a four-step process to predict cryptocurrency prices:
1. Scraping crypto-currency data from two different sources
2. Exploratory data analysis, Visualization
3. Data pre-processing, Feature Engineering
4. Training and testing, evaluation.
5. Forecast the price of the cryptocurrency.

Literature review:

CoinMarketCap is the world's most-referenced price-tracking website for crypto assets in the rapidly growing cryptocurrency space. Its mission is to make crypto discoverable and efficient globally by empowering retail users with unbiased, high quality, and accurate information for drawing their own informed conclusions. LexisNexis is a huge electronic database of newspaper and magazine stories, transcripts of TV broadcasts, and summaries of public records filings.

Our research will try to answer the following question: 1. Which among the two well know platforms or data sources (coinmarketcap.com, LexisNexis) is useful in providing a better forecast of respective cryptocurrencies. 2. On the side of machine learning and model implementation, we will be using NLP and Deep learning to find better performing Models. Under NLP we are going to use state-of-the-art pre-trained models on data collected from Lexis-Nexis and LSTM on data collected from coinmarketcap.com. At the end, we will compare the results from the two models. The inspiration for this approach came from the work done by Priyanka and colleagues in their paper “Stock price prediction using BERT and GAN Research” (https://arxiv.org/pdf/2107.09055.pdf) shows the comparison of BERT with other baseline models including LSTM. Based on this that we are going to perform a model comparison for cryptocurrency.

Data sources and methodology:

Data Sources:

1. Coinmarketcap.com
2. LexisNexis news database

Methodology:

1. Phase 1: Web scraping and data collection-storage
2. Phase 2: Pre-processing raw data
3. Phase 3: EDA which will include data insights, visualization
4. Phase 4: Feature Engineering
5. Phase 5: Model selection, training-testing-evaluation
6. Phase 6: Report creation

Data Scraping:

We have scraped data from coinmarketcap.com and Lexis-Nexis. The features that we are going to extract from Coinmarketcap are Date, Cryptocurrency name, value, open value, close value, Volume, Market Cap, and from Lexis-Nexis: Daily news text regarding cryptocurrency.

Since both of these sites have intense usage of javascript on them, we decided to go with selenium.

Scraping coinmarketcap:

The steps for this scraping procedure was as follows:

1. Load the historical data page for a particular cryptocurrency(you can see the desired table to be scraped on this page with features like Date, Open, High, Low, Close, Volume and MarketCap)
2. Scroll towards the very end of the page and locate the “Load More” button(since we need to load more than just 3 months of data which is the default amount of data loaded)
3. Click on the button 12*(n-1) (where n is the number of years’ worth of data you need to load up) times and simultaneously scroll down and locate the button each time we click it.
4. Then once the button has been clicked the desired number of times, we stop and collect the table contents.

Scraping LexisNexis database:

The steps for this scraping procedure was as follows:

1. Load up the LexisNexis website and login into it using selenium

2. Search for the term (in our case “bitcoin”) you want the news for
3. Select all the filters required (sort relevance: from newest to oldest news, choose the publication: Newstex Blogs, etc.)
4. Scrape all the titles of the news displayed on the page
5. Scroll down to the bottom and locate the pagination section
6. Click on the next page and repeat the same from step 4

Preliminary description of data

Description of the coinmarketcap.com Scrapped Columns:

1) **Date** - Date range of cryptocurrency for analysis

2) **Open** - Price of the cryptocurrency at the Start of the Day

3) **High** - Highest Price of the cryptocurrency

4) **Low** -Lowest Price of the cryptocurrency

5) **Close** -Price of the cryptocurrency at the End of the Day

6) **Volume** -Volume is the amount of assets traded during a specific time frame

7) **MarketCap** -Total value of cryptocurrency.

8) **CryptoName** - Name of the cryptocurrency.

Description of the Lexis-Nexis Scrapped Columns:

1) **Date** - The date of publication
2) **Titles** - The title of the news

Pre-processing methods:

Preprocessing steps implemented on scraped coinmarketcap data:

1. Renamed columns {"Open*":"Open", "Close**": "Close", "Market Cap": "MarketCap"}
2. “Date” column from String to Date time format
3. Converting all the string values in columns ['Open', 'High', 'Low', 'Close', 'Volume', 'MarketCap'] to numerical values (floats)

Preprocessing steps implemented on scraped LexisNexis data corpus:

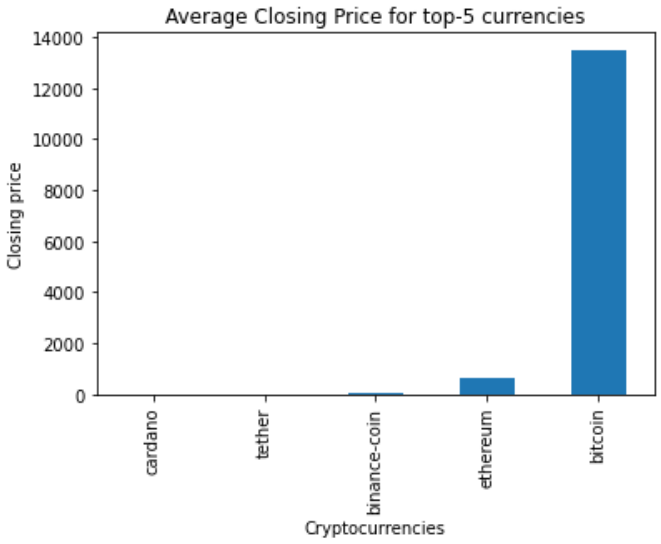
1. Lowercase each word
2. Remove digits and special characters
3. Perform lemmatization

Exploratory data analysis

Coinmarketcap.com

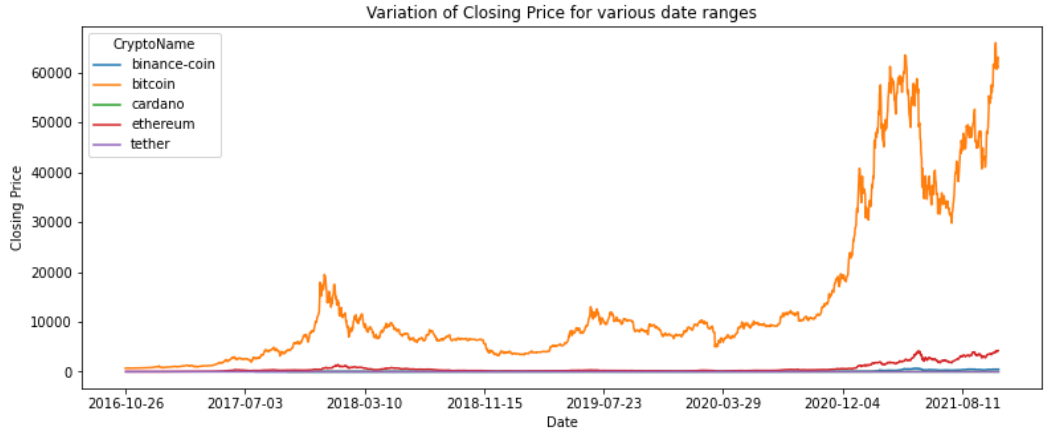
1) Average Closing Price for top 5-currencies

```
Out[25]: CryptoName
         bitcoin         13499.099984
         ethereum          640.013627
         binance-coin       77.117319
         tether              1.000770
         cardano             0.388136
         Name: Close, dtype: float64
```



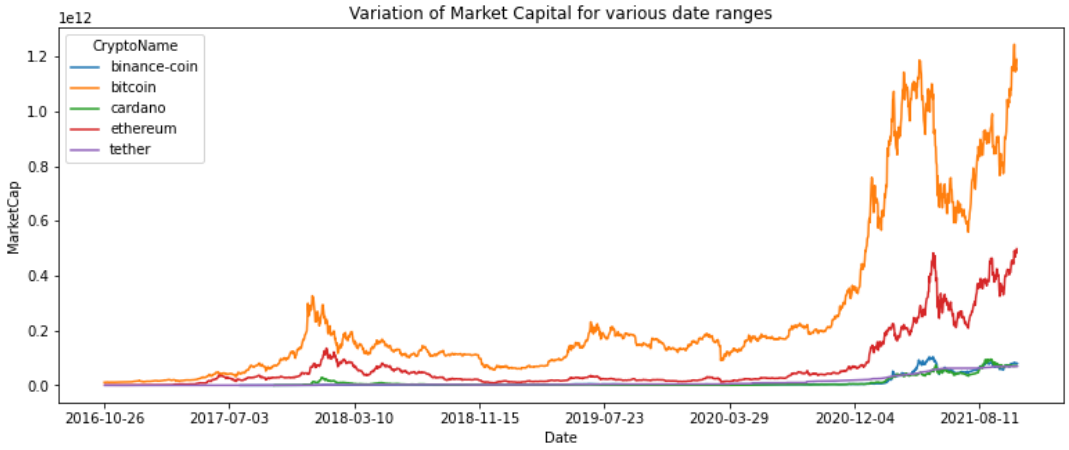
The figure above shows the average change in closing price wrt to various crypto-currencies.

2) Variation in Closing Price for various date ranges.



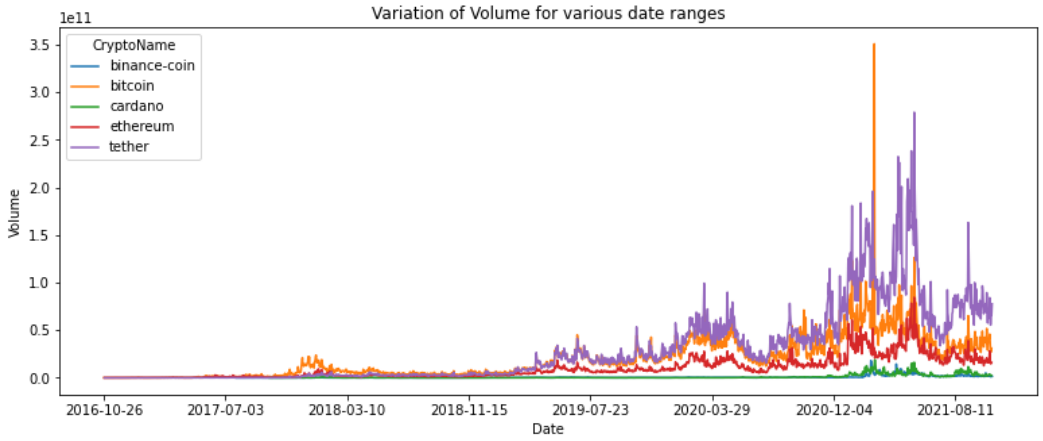
The figure above depicts the change in closing price across the date ranges.

3)Variation in Market Capital for various date ranges.



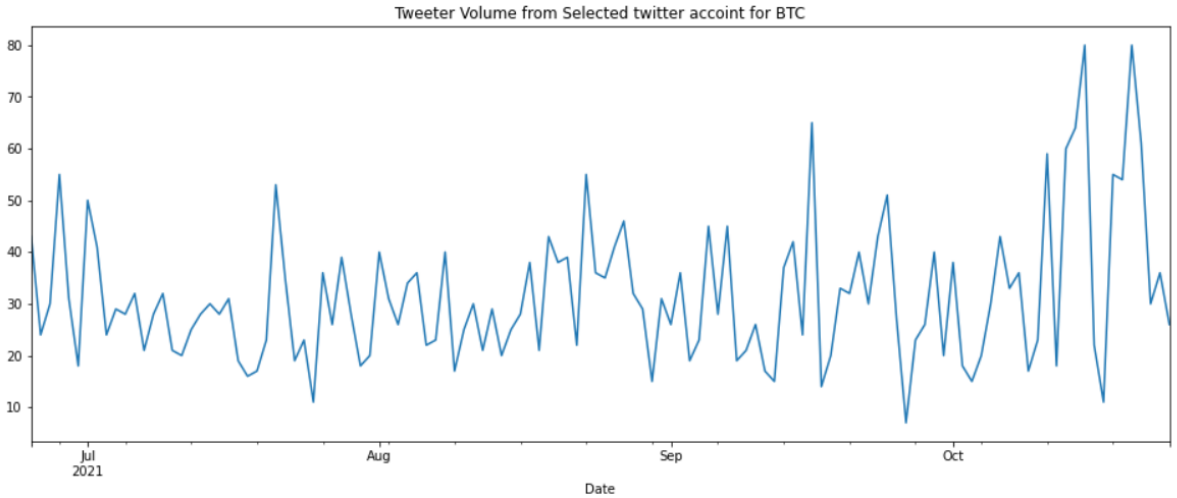
The figure above depicts the change in Market Capital across the date ranges. The highest variation was found for Bitcoin wrt MarketCapital.

4)Variation in Transaction Volume for various date ranges.



The figure above depicts the change in transaction volume across the date ranges. It is observed that all the crypto-currencies show high variations in volume beyond 4th Dec 2020.

5)Twitter tweetings volume from the selected twitter accounts



The figure above depicts the twitter tweets from various accounts over the date range. We can see that the variations are high past October 2021.

LexisNexis News

Data considered for EDA from May 2011 to Oct 2021

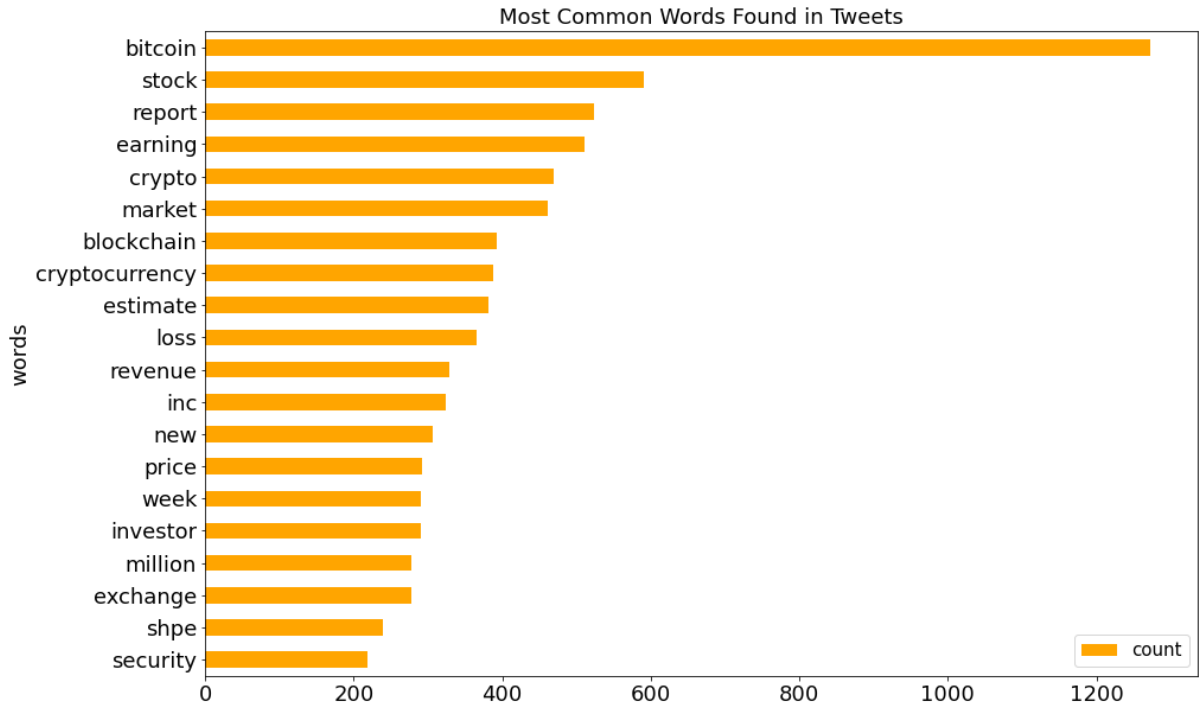
A word cloud is a pictorial representation of word frequency from a corpus of text data, the variable being size of the word in the word cloud.

Word cloud of top 80 words from the Lexis Nexis news data corpus

On python word clouds can be easily plotted using the WordCloud() method from word cloud library. The above word cloud was obtained after cleaning the corpus from stop-words, punctuations, digits, special characters, etc. Then lemmatization(the process of grouping together the different forms of a word and replacing them with a single root word) was performed as well which reduced the vocab size from 80k to around 60k.

Insights: Words like bitcoin, cryptocurrency, crypto, stock, etc. are intuitively more prevalent in the corpus.

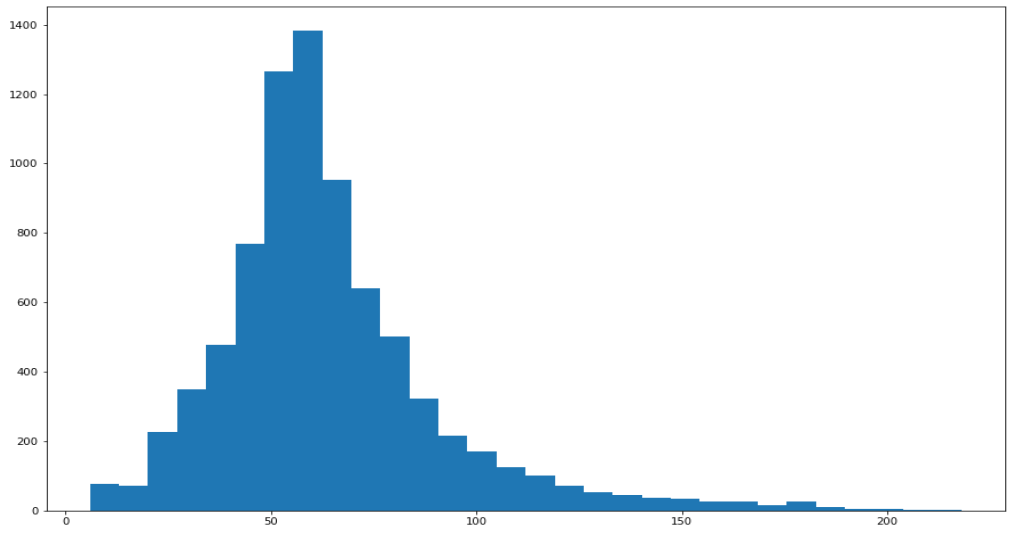
Another form of word frequency plot can be seen in the bar graph shown below.



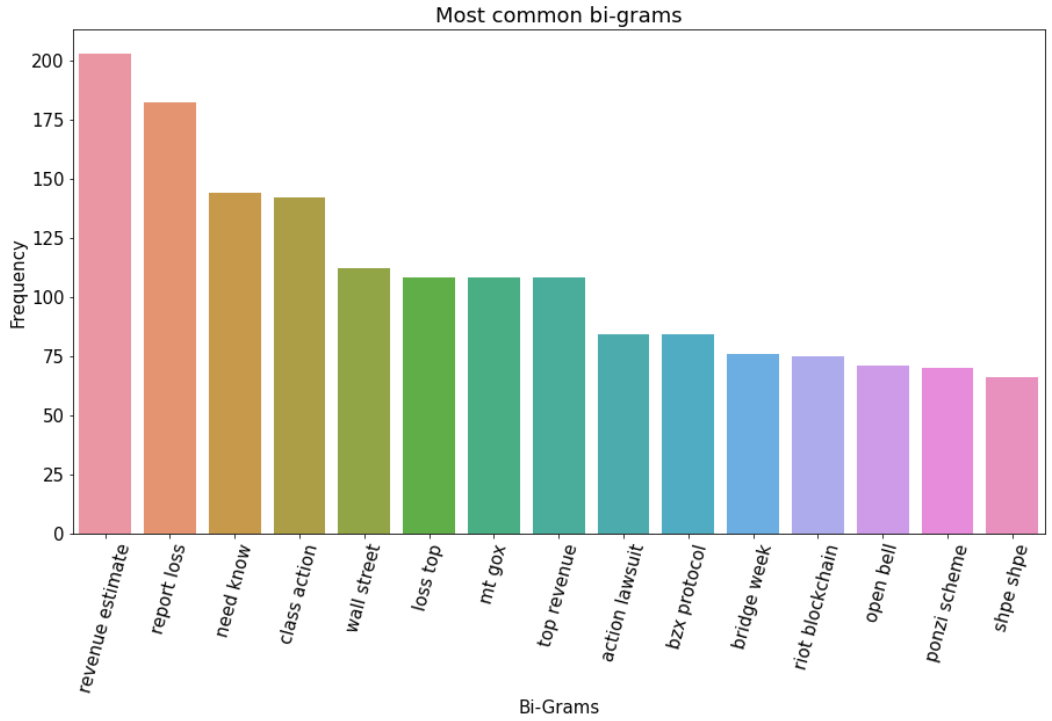
The insights are the same as mentioned before for word cloud.

The below plot shows the frequency distribution of news lengths. As expected it is a right-skewed graph(since we decided to only scrape the new article titles rather than the entire news paragraph). The dataset must be filled with shorter sentences rather than longer ones.

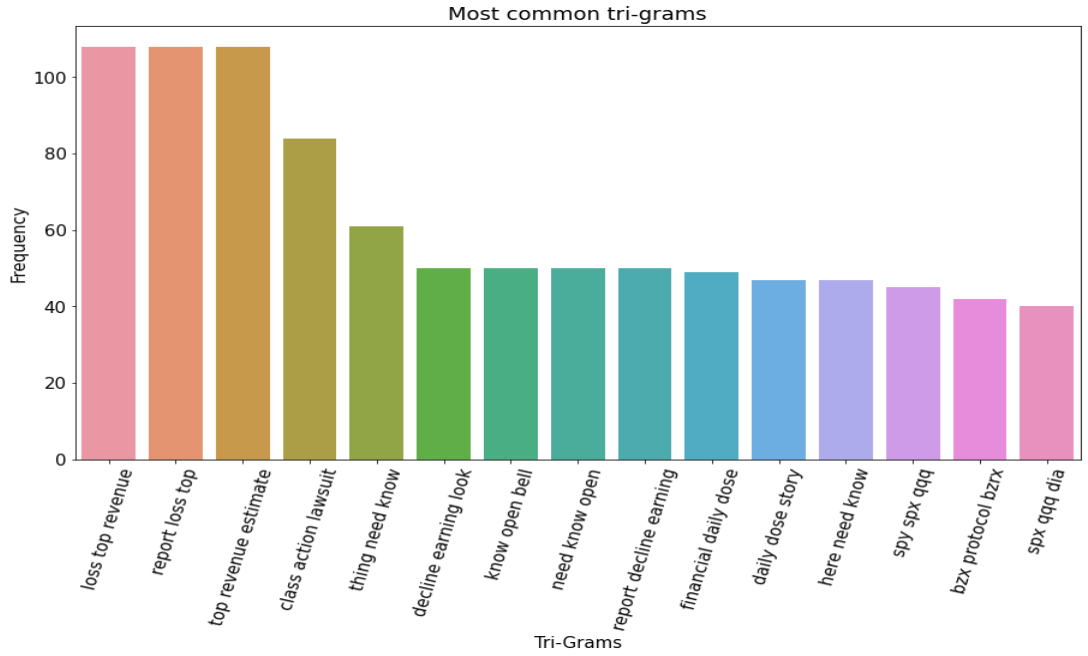
Insights: The distribution has a mean close to 60, hence most of the sentences(news titles) are more or less 60 words in length.



Then we plotted bigram and trigram frequency plots.
An n-gram is a continuous sequence of n items iteratively taken from a corpus of text.



Insights: Phrases like “revenue estimate”, “report loss”, etc. are the most occurring bigrams.



Insights: Phrases like “loss top revenue”, “report loss top”, etc. are the most common trigrams.

Preprocessing

The pre-processing for the project has been done for both structured and unstructured datasets.
1. **Structured Data**: The structured data that we obtained was after scraping from coinmarketcapital website. The data scrapped had columns like,

Date,Open,High,Low,closeValue,Volume,Market Cap.
These columns whereof 'Object' datatype and were converted to their respective datatypes. The date was converted to respected date formats. NAN and missing value checks were performed on the dataset.

```
1 pdCoinMarketData.dtypes
```

```
Date          object
Open*         object
High          object
Low           object
closeValue    object
Volume        object
Market Cap    object
dtype: object
```

The figure above shows the data types of columns before preprocessing.

	Date	Open*	High	Low	closeValue	Volume	Market Cap
0	2021-10-24	61,368.34	61,505.80	59,643.35	60,930.84	27,316,183,882	1,148,743,134,468
1	2021-10-23	60,694.63	61,743.88	59,826.52	61,393.62	26,882,546,034	1,157,410,091,263
2	2021-10-22	62,237.89	63,715.02	60,122.80	60,692.26	38,434,082,775	1,144,131,483,274
3	2021-10-21	66,002.23	66,600.55	62,117.41	62,210.17	45,908,121,370	1,172,684,282,558
4	2021-10-20	64,284.59	66,930.39	63,610.67	65,992.84	40,788,955,582	1,243,927,428,207

The figure above shows the sample of the dataset where the date is well-formatted.

2. **Unstructured Data:** The unstructured date where obtained from LexisNexus and Twitter data sources.

	titles	c
Date		
2021-10-24	NewsWatch: Big Tech stocks are the market's su...	
2021-10-24	bZx Protocol (BZRX) Price Hits $0.33 on Top Ex...	
2021-10-24	bZx Protocol Price Reaches $0.33 on Exchanges ...	
2021-10-23	Inside the Courts - An Update From Skadden Sec...	
2021-10-23	What's The Deal With Well Everything?!!	
...	...	

The figure above shows samples of text data from LexisNexis data source.

	datetime	tweet_id	text	username
0	2021-10-05 14:59:05+00:00	1445403172705607683	investor says " we are at the top of the first...	DocumentingBTC
1	2021-10-05 14:57:04+00:00	1445402664049815555	to launch custody services ' s the largest r...	DocumentingBTC
2	2021-10-04 16:40:41+00:00	1445066353556197384	is never down	DocumentingBTC
3	2021-10-04 15:09:26+00:00	1445043389666312195	on news with	DocumentingBTC
4	2021-10-04 14:17:15+00:00	1445030259527753740	growth is going vertical	DocumentingBTC

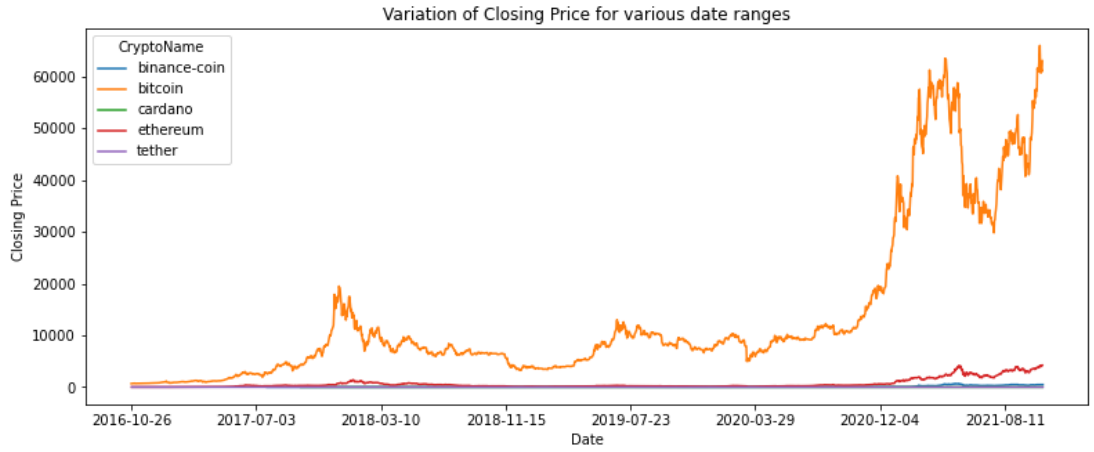
The figure above shows the tweets regarding bitcoin from twitter api.

Text Preprocessing was applied on the above two data sets. Firstly the tweets or the text data were tokenized using a text-to-word sequence. Also stop word removal, removing numbers, stemming and lemmatization was performed on the text data as part of preprocessing.

Models Selected

In the financial world, the bitcoin price prediction is very important for investors to understand the current trend and they have their own way of doing so. In this project, we are trying to provide a good tool that can provide additional insights or abilities along with their existing ways of predicting the trend. Also, they should take any price predictions with a good degree of skepticism.

We were able to scrape data not only for bitcoin but for other currencies too from all the data sources.



But we decided to stick to bitcoin for the forecasting for model development.

We selected the following models

1. Two time-series models: Facebook's Prophet, SARIMA
2. Text Processing -> TFIDF -> Models (Linear SVM, RF, XGBoost)
3. BERT -> Models (Linear SVM, RF, XGBoost)

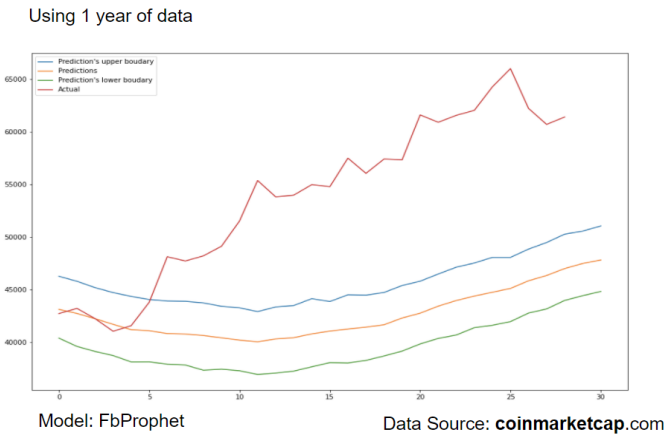
We explored these three models for the forecasting out of which 2 of them are popular time-series algorithms i.e. SARIMA and Prophet developed by my teammates and the third one is using unstructured data that I have developed.

The time series forecasting algorithms consider components such as trend, seasonality, and holidays. We gave models the closing price as input and observed the forecasting of future closing values as output

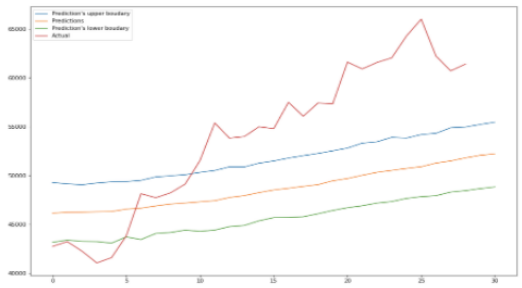
Forecasting results:

The time series forecasting algorithms consider components such as trend, seasonality, and holidays. We gave models the closing price as input and observed the forecasting of future closing values as output.

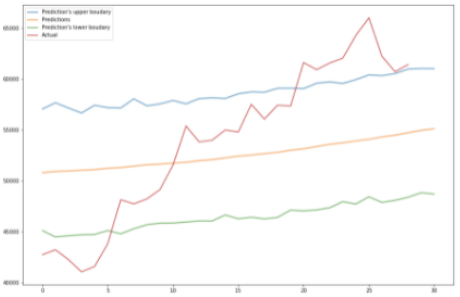
This is the forecasting using Facebook's prophet with coinmarketcap input data. Facebook's prophet as it can be seen here was not at all able to capture the trend correctly. This indicates some lack of tuning on our side



Using 2 years of data



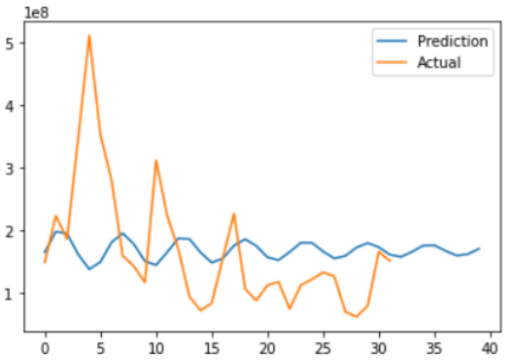
Using 5 years of data



Model: FbProphet

Data Source: coinmarketcap.com

But it can be seen that time-series models tend to improve performance with more data up to a certain extent. So with the prophet, we got these results. A little better than 1-year data. Now let's see SARIMA results:



Model: ARIMA

Data Source: **coinmarketcap**.com

It was able to perform better than Facebook's prophet but not able to make good forecasting of actual values. Now let's see forecasting using LexisNexis and Twitter dataset:

Instead of scrapping all tweets related to bitcoins, we decided to scrape tweets from selected most referred accounts only. On each sample, we performed these 5 processings: Word tokenization using text-to-word-sequence, Removed stop words like a, an, the; then joining words into sentences and then removing numbers if any; And in the end, performed Stemming & lemmatizing

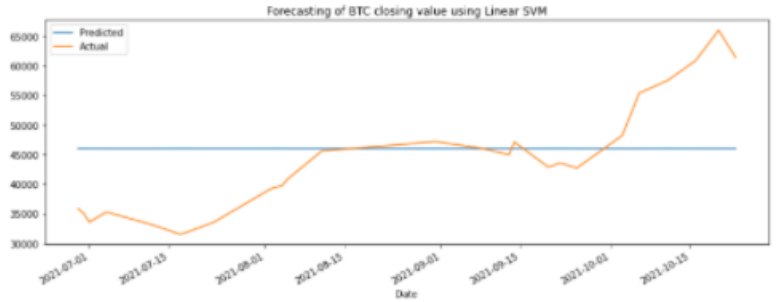
```
1 sampleText
```

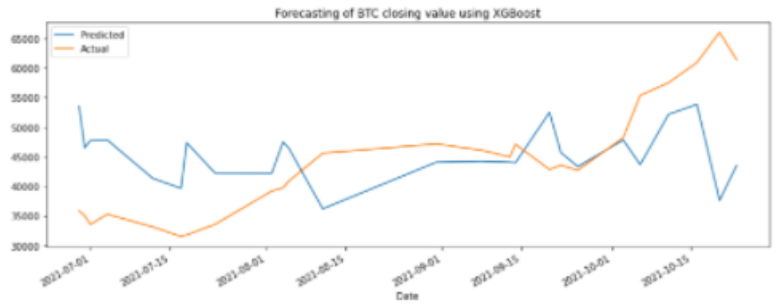
```
'The messages will be "unwrapped" by sculptor Richard Wentworth, who is responsible for decorating the tree with broken plates
and light bulbs. Artists who have decorated the Tate tree in previous years include Tracey Emin in 2002.'
```

```
1 preprocessDataset(sampleText)
```

```
--- Tokenized ----
 ['the', 'messages', 'will', 'be', 'unwrapped', 'by', 'sculptor', 'richard', 'wentworth', 'who', 'is', 'responsible', 'for', 'd
ecorating', 'the', 'tree', 'with', 'broken', 'plates', 'and', 'light', 'bulbs', 'artists', 'who', 'have', 'decorated', 'the',
'tate', 'tree', 'in', 'previous', 'years', 'include', 'tracey', 'emin', 'in', '2002']

---- Removed Stop Words ----
  ['messages', 'unwrapped', 'sculptor', 'richard', 'wentworth', 'responsible', 'decorating', 'tree', 'broken', 'plates', 'ligh
t', 'bulbs', 'artists', 'decorated', 'tate', 'tree', 'previous', 'years', 'include', 'tracey', 'emin', '2002']

---- Joined as Sentense ----
 messages unwrapped sculptor richard wentworth responsible decorating tree broken plates light bulbs artists decorated tate tre
e previous years include tracey emin 2002

---- Numbers removed ----
 messages unwrapped sculptor richard wentworth responsible decorating tree broken plates light bulbs artists decorated tate tre
e previous years include tracey emin

---- Stemmed ----
 messag unwrap sculptor richard wentworth respons decor tree broken plate light bulb artist decor tate tree previou year includ
tracey emin

---- Lemmatized ----
 messag unwrap sculptor richard wentworth respons decor tree broken plate light bulb artist decor tate tree previou year includ
tracey emin
```

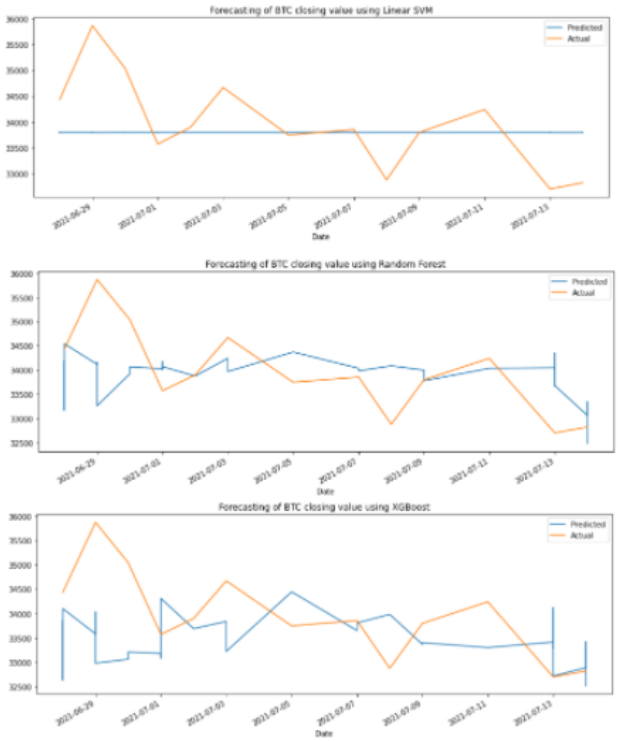
Using TF-IDF to vectorize the processed text input data I have implemented three models Linear SVM, RF, and XGBoost for the forecasting, and results can be seen here. It is able to capture the trend more accurately. But the overall accuracy is below 90%. Now let's see what happens when we use feature extraction from the text samples.





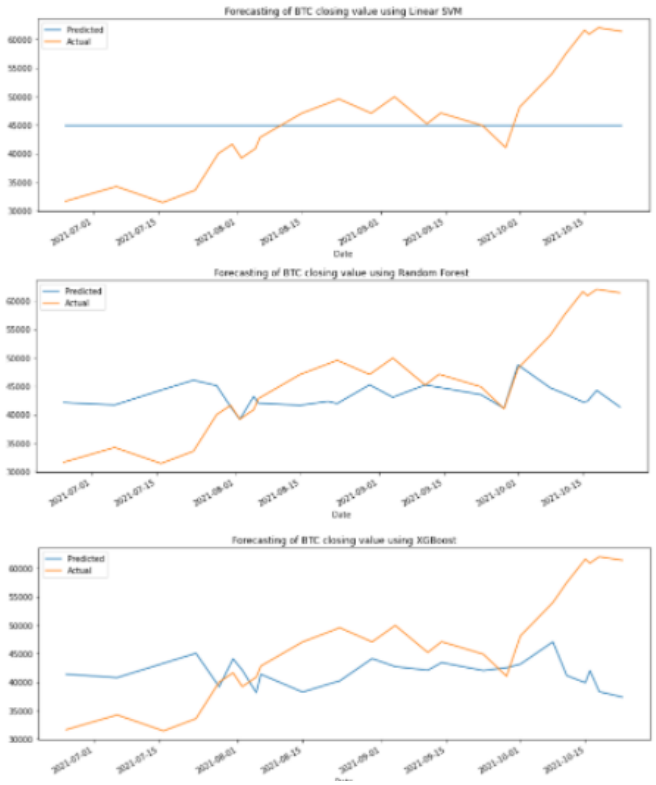
Data Source: Twitter

From the processed text data we extracted features using advanced models like BERT. This feature vector is then fed to models such as linear SVM, RF, and XGBoost for the prediction of the value. The BERT model we used is a pre-trained one from the Huggingface library without any parameter tunings. Let's see the output.



Data Source:
LexisNexis

The output for the Lexis Nexis dataset is like this and let's see the output for the Twitter dataset.



Data Source:
Twitter

The overall performance is much improved.

Conclusion

We explored which data sources can be useful in forecasting the trend. We got good results in the case of unstructured data collected from LexisNexis & Twitter compared to coinmarketcap. We tried variations in input data size used for the forecasting for example in the case of text data for training we used only the latest 3-4 months of data in the case of time-series we used 1 to 5 years of data. We explored different models and we saw the results from each model.

There is a scope to tune the models - Can tune BERT and train with more samples or use more complex models RNN or Deep Learning model. Also, there is a scope to combine useful features across different data sources in model development in addition to that we can use unused features such as Sentiment, the daily surge in tweets.

- There is a scope of improvement
    - Time series models
    - Distilled-BERT
- The business application
    - To understand the crypto trend
    - forecasting
    - grasp the sentiment
    - unusual market fluctuation
    - valuable dataset for other research work

You can find the code: GitHub