



Module Code & Module Title
K20BG - Artificial Intelligence

Assessment Type
Continuous Assessment 3

Student Name: Adarsh Khatri
Registration Number : 12020971
Assignment Submission Date: 20th April, 2022

I confirm that I understand my coursework needs to be submitted online via UMS platform under the relevant module page before the deadline in order for my assignment to be accepted and marked. I am fully aware that late submissions will be treated as non-submission and a marks of zero will be awarded.

Abstract

Free market driven assets such as stocks and cryptocurrencies are highly volatile and their prices are dependent upon the public opinion. Investing in these assets without assessing the public opinion and market sentiment might lead to losses. However, analysing the public sentiment is not an easy process, especially when billions of opinions are expressed in the internet every day.

This project aims to solve that problem by training a binary naïve bayes algorithm to predict the sentiment of a text. It analyses the posts from the cryptocurrency community in reddit called r/cryptocurrency and predicts the sentiment of each post. This data is used to calculate the positive to negative sentiment ratio of two of the biggest cryptocurrencies namely, Bitcoin and Ethereum. Both the overall market sentiment and positive to negative sentiment ratio of the two main currencies are visualized and the overall market sentiment has been concluded as positive.

Acknowledgements

This project could not have been possible without the guidance of our lecturer Mr. Nitin Kumar Sir. I would not have conceptualized and understood the nuances of Machine Learning and Artificial intelligence without his lectures.

A big thanks to authors Sowmya Vajjala, Bodhisattwa Majumder, Anil Gupta, and Harshit Surana. Without their book 'Practical natural language processing', I would not have developed a fascination with the field of Natural Language Processing.

Table of Contents

1. INTRODUCTION.....	1
1.1. INTRODUCTION TO NATURAL LANGUAGE PROCESSING	1
1.2. INTRODUCTION TO SENTIMENT ANALYSIS	2
1.3. PROBLEM DOMAIN	2
2. BACKGROUND.....	4
2.1. REVIEWS OF EXISTING WORKS IN PROBLEM DOMAIN	7
2.2. ADVANTAGES AND LIMITATIONS OF USING SENTIMENT ANALYSIS	9
3. SOLUTION.....	10
3.1. SOLUTION TO THE PROBLEM	10
3.2. EXPLANATION OF THE ML ALGORITHM USED	13
3.3. IMPLEMENTATION OF THE ALGORITHM IN THE SOLUTION.	14
3.4. PSEUDOCODE OF THE PROPOSED SOLUTION.....	15
3.5. FLOWCHART OF THE PROPOSED SOLUTION.	17
3.6. DEVELOPMENT PROCESS.....	18
3.7. ACHIEVED RESULTS.....	19
4. CONCLUSION.	22
4.1. ANALYSIS OF MY WORK.....	22
4.2. HOW THE WORK SOLVES REAL WORLD PROBLEMS.....	22
4.3. LIMITATIONS OF THE SOLUTIONS.	23
4.4. FUTURE WORK.....	23
REFERENCES	24

Table of figures

Figure 1: Machine learning approach in sentiment analysis (Thakkar & Patel, 2015).....	5
Figure 2: Bayesian Probability. Source: analyticsvidhya.com.	13
Figure 3: Flowchart of the proposed solution	17
Figure 4: Bar chart depicting the positive to negative sentiment ratio of bitcoin and ethereum.....	19
Figure 5: Bar chart depicting the overall positive vs negative sentiment of r/cryptocurrency.....	19
Figure 6: Pie chart depicting the overall positive vs negative sentiment of r/cryptocurrency.....	20
Figure 7: Word cloud depicting the most discussed word in r/cryptocurrency.	20

1. Introduction

This project aims to extract information regarding the overall sentiment of the cryptocurrency community (r/cryptocurrency) from the social media site reddit. It accomplishes so by separating the posts according to different cryptocurrencies, calculating the number of positive and negative posts regarding each cryptocurrency, and by computing their positive to negative ratio. The motivation behind choosing this topic was to help investors analyse the overall sentiment of the cryptocurrency community.

1.1. Introduction to Natural Language Processing

Language is central to human thought process. In fact, some Anthropologists claim that the human ability to cognize complex thoughts originated simultaneously with language. Therefore, starting with cave-dwelling hunter gatherers, to the first civilization of ancient Sumer, to modern day countries, language has always been the predominant media of communication.

Recent advances in information sharing, such as the invention of the internet, have only fuelled this urge to formulate, construct, and share information through language. Tweets and Emails, no less than a billion move across the internet every day. Books make the use of written language to share ideas. People comment and review products on internet marketplaces using texts. Computers are given instructions in the form of written language. It could, in fact, be safely concluded that the world of today brims with language. The amount of textual data (both written and spoken) that traverses the internet every minute is larger than what is humanly possible to understand and cognize. This problem gave birth to a new set of techniques that train machines to understand and extract information from large textual data, aptly named Natural Language Processing.

Natural Language Processing, colloquially known as NLP, is a sub-domain of Machine Learning and Artificial Intelligence that deals with language. Combining Computer Science, Artificial Intelligence, and Linguistics, NLP provides machines the

ability to identify, analyse and understand human feelings, judgements, responses, and emotions through written or spoken text (Dey, et al., 2019).

1.2. Introduction to Sentiment Analysis

Sentiment Analysis is a subset of Natural Language Processing that determines whether a text (written or spoken) is positive or negative, or neutral in some cases. Modern day companies contextually mine text and extract subjective information from online texts to understand the social sentiment of their brand, product or service (Gupta, 2018).

The use cases of sentiment analysis include but aren't limited to the following:

1. Analysing the review sentiment of certain products to see whether they are reliable for long term use.
2. Analysing customer sentiments to understand their pain points.
3. Understanding the public's opinion regarding certain public matters.

1.3. Problem domain

It is common knowledge that assets regulated by free markets such as the cryptocurrency and the stock market, are sensitive to buyers' and sellers' emotions. "If more people want to buy a stock compared to the people wanting to sell it, the supply goes down and the demand goes up, which increases the price of the stock. On the flipside, if more people want to sell a stock than people willing to buy it, the supply increases and the demand decreases. This decreases the price of the stock" (Desjardins, 2021).

Understanding the market's overall sentiment is therefore crucial to make thoughtful investment and trading decisions. However, considering the colossal amount of information and opinion that moves around the internet every day, it is hard to analyse and determine the overall sentiment of the market by a single person.

This project aims to solve that problem through the use of a Naïve Bayes classifier algorithm, trained to identify the sentiment of textual data. It solely focuses on identifying the market sentiment of the cryptocurrency community in reddit, called r/cryptocurrency.

2. Background

“Sentiment analysis, refers to the study of opinions sentiments and emotions expressed in a text. (Ortigosa-Hernandez, et al., 2012)” Although the field of sentiment analysis is broad, a large body of work in this domain revolves around the classification of polarity and subjectivity/objectivity of an opinion or a body of opinions. Its main aim is to extract emotions, sentiments, and behavioural intents using the methods of natural language processing (Stoy, 2021).

Generally, sentiment analysis is performed on a body of text using one of the following three processes:

1. Lexical Analysis

This technique of analysis uses a set of dictionary, pre-tagged with words having positive or negative sentiments. The body of data, or the input text, is first tokenized (broken down into words or set of words) and matched with the words in dictionary. If the word has been pre-tagged with a positive sentiment, the score of the text's positivity is increased. If some token has been tagged with a negative sentiment a priori, the text's negativity score is increased. The text is finally tagged as positive or negative on the basis of the calculated score.

2. Machine learning methods

Sentiment analysis using machine learning can be performed by training machine learning algorithms using human-annotated datasets (Stoy, 2021). Popular machine learning algorithms for sentiment analysis include Naïve Bayes classification, Support Vector machine, Hidden Markov Model, and Conditional random fields (Vajjala, et al., 2020).

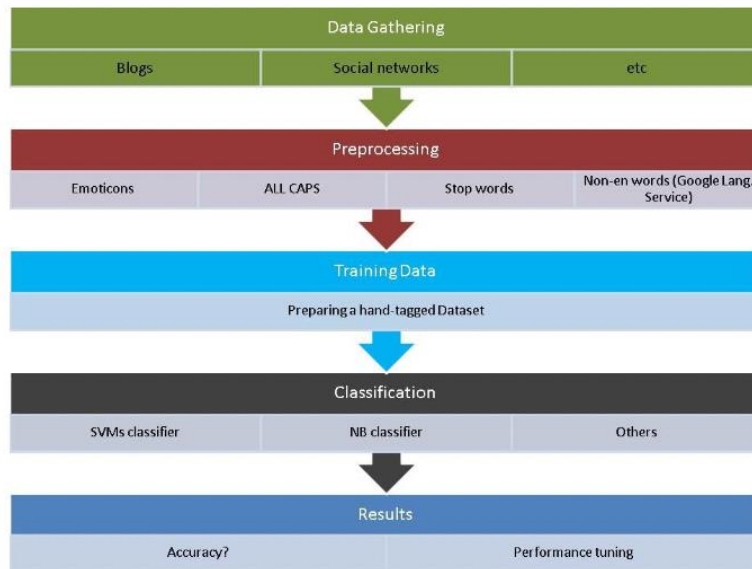


Figure 1: Machine learning approach in sentiment analysis (Thakkar & Patel, 2015).

A typical machine learning based sentiment analysis has the following steps:

1. **Data collection:** In this step, data is collected from areas such as blog posts, social media opinions, comments on e-commerce sites, depending upon the area of application.
2. **Data cleaning:** In this step, the collected data is processed by filtering out punctuations, symbols, stop words, non and other unnecessary data through the help of either regular expressions or other similar methods.
3. **Preparing the training data:** Here, the cleaned data is hand-tagged as positive or negative often by crowd sourcing (Thakkar & Patel, 2015). This tagged data is used for the machine learning model to be trained in the next step.
4. **Training the model:** In the step, the machine learning algorithms such as Naïve Bayes or Support Vector Machine are trained using the training data prepared in step 3.
5. **Results:** Finally, the trained model is used to analyse the sentiment of a large volume of text such as social media comments and user reviews.

3. Hybrid methods

This method of sentiment analysis combines both lexical and machine learning methods. Although this has not been widely explored due to inevitable complexity that often arises with combined approaches, it still is a viable option.

2.1. Reviews of existing works in problem domain

Listed below are some of the existing works done in the domain of sentiment analysis:

2.1.1. Predicting Consumers' Brand Sentiment Using Text Analysis on Reddit

By processing all posts and comments data from the r/gadgets subreddit community, (Cen, 2020) extracts frequently-discussed brands through named entity recognition, and performs sentiment analysis to generate brand sentiments. The author then uses four supervised learning algorithms to predict the public sentiments for four brands, namely, Apple, Samsung, Microsoft, and Google (Cen, 2020).

2.1.2. Sentiment analysis of financial news posted on Reddit and the Financial Times

In this report, (Lubitz, 2018) analyses posts from the subreddit r/economics and extracts news data from the links present in the post. He also uses a web-crawler to analyse news from the financial times website using python. After cleaning and separating the noun from the data sources, he then labels those data as positive (+1) or negative (-1) and puts them in a dictionary. Finally, he calculates the sentiment of the overall document (d) by counting the number of positive words (p) and negative words (n), by using the formula $(sd = (p - n) / (p + n) ; -1 \leq sd \leq 1)$ (Lubitz, 2018).

2.1.3. Forecasting of the cryptocurrency market through social media sentiment analysis

In this paper, (Salač, 2018), analyses the possibility of extracting data from reddit to predict bitcoin price movement. As opposed to the current standard of using twitter data, reddit data to analyse the public's sentiments. The author uses the VADER toolkit to compare price movement of bitcoin and user sentiments in a course of three months. (Salač, 2018)

2.1.4. NLP for Stock Market Prediction with Reddit Data (Xu, 2021).

In this report (Xu, 2021), collects data from the subreddit r/wallstreetbets and performs sentence embedding using the BERT library, uses the Doc2Vec library to combined the embedded sentences into documents, and analyses the sentiments of those documents using two libraries namely, TextBlob and VADER. Subsequently, she uses a Convolutional Neural Network to predict the stock price movements. (Xu, 2021)

2.2. Advantages and Limitations of using sentiment analysis

Advantages:

1. Sentiment analysis can be used make machines understand human language and the subtleties of emotions behind them. This helps in creation of intelligent, language gnostic machines.
2. Customers' product reviews can be analysed by companies to cater their products to create what is called 'a product-market fit.'
3. Companies can listen to customer pain-points and minimize their frustration by using sentiment analysis.
4. Individuals can analyse the overall trend in certain domains such as politics or philosophy to make informed day-to-day decisions.
5. The sentiment of all the masses towards different free market driven assets can be analysed to make informed investment decisions.

Limitations:

1. Often the data in social networks are context dependent. However, most algorithms are context-agnostic which makes the analysis somewhat unreliable.
2. These days the sentiment of most opinions in social media are laced with hashtags, emojis, and other internet-related influences. These work synergistically to determine the overall context of those opinions, which increases the difficulty in analysing the overall sentiment.
3. Almost often statistical irregularities such as the law of small numbers and data overfitting occur. Due to which the reliability of sentiment analysis algorithms goes down.
4. Most algorithms do not have a highly-reliable mathematical accuracy which, again, make the analysis undependable.

3. Solution

3.1. Solution to the problem

The steps taken to solve the problem of analysing the market sentiment of four major cryptocurrencies are listed below:

- 1. Importing necessary libraries:** Before starting the project, all the required Python libraries for text processing were imported. These include NumPy, Pandas, re (regular expression), time, wordcloud, nltk (a predominant Python library for Natural Language Processing).
- 2. Initializing the dataset as a pandas data frame:** Afterwards, a sentiment annotated dataset (imported from Kaggle) containing more than 1 million tweets was initialized as a pandas data frame to check the overall structure of the dataset.
- 3. Separating the data:** The dataset was split into two lists: one containing all the positive texts, and another containing negative texts.
- 4. Reducing the amount of data:** The total number of data in the dataset was 1.6 million. Due to which, step 3 took a lot of time for computation. To mitigate this, the amount of data in both positive and negative lists were halved. This step reduced the total number of texts to 800,000, which resulted in 400,000 positive texts and 400,000 negative texts.
- 5. Cleaning, Tokenizing, and Stemming the dataset:** In this step a function for cleaning tokenizing and stemming texts was created. Taking a single text as a parameter, the function removed the punctuations, hyperlinks, stop words, and hashtags. It then tokenized the text and converted each word into its respective lemma through the process of stemming.
- 6. Initializing the training data:** Here, the list containing all the positive texts, and the list containing all the negative texts were concatenated to create a new list. Similarly, another list containing all the sentiments of the data (0 for negative and 1 for positive) was created.
- 7. Creating a frequency dictionary:** A function that takes two parameters, a list of texts and another list of sentiments, to create a frequency dictionary was written.

This function returned a frequency dictionary by taking the two lists initialized in step 6 as parameters.

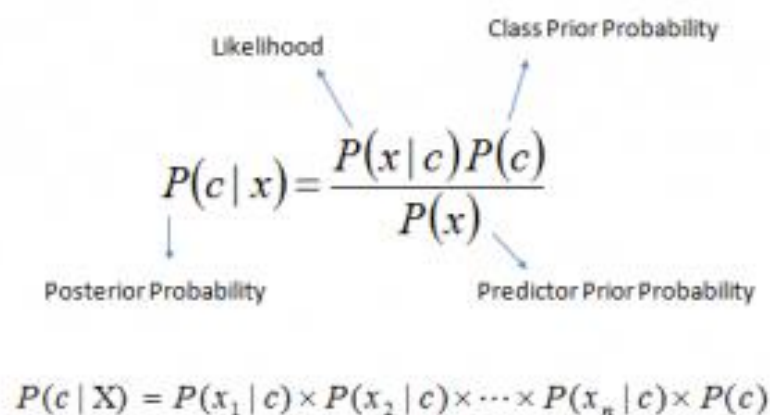
- 8. Training the Naïve Bayes algorithm:** A naïve bayes algorithm was trained using the frequency dictionary, and the two training data lists initialized in step 6. Firstly, the log (prior probability log) of a text being positive or negative was calculated. Secondly, a list containing all the unique words was created using the frequency dictionary. Afterwards, by iterating through the list of unique words, the probability of each individual word being positive $P(\text{pos})$ or negative $P(\text{neg})$ was calculated, and the ratio of the log of $P(\text{pos})$ and $P(\text{neg})$, i.e. $\log(P(\text{pos})) / \log(P(\text{neg}))$ was stored in a variable called word likelihood. Afterwards, the word, and the word's likelihood log was used as key, value pairs to construct a word likelihood dictionary.
- 9. Predicting the sentiment of a text:** A function that takes the prior probability log, the word likelihood dictionary and a text as parameters was written. The function processes the text using the function written in step 5, and adds the probability of each individual word in the text to a variable. Finally, the function adds the prior probability to the variable and returns it. A value less than 0 denotes that the text is negative. A value more than 0 denotes that the text is positive.
- 10. Importing Praw:** Praw, a python library that acts as a wrapper to reddit's api was imported.
- 11. Initializing the Praw object:** The Praw object was initialized by using reddit's developer credentials.
- 12. Fetching the prediction data:** Top 1000 newest posts from reddit was fetched into a list, and the list was separated to four lists (Bitcoin, Ethereum, Solana, and Dogecoin) on the basis of cryptocurrencies mentioned in the posts.
- 13. Analysing the sentiments of the cryptocurrencies:** All the posts were passed through the function created in step 9, and the total number of positive and negative post of each cryptocurrency was calculated and stored in a dictionary.
- 14. Calculating the Positive to Negative sentiment ratio:** The positive to negative sentiment ratio (PN ratio) of two cryptocurrencies: bitcoin and ethereum was calculated.

- 15. Visualizing the sentiment:** The PN ratio of cryptocurrencies mentioned in step 14 and their names are visualized using matplotlib.
- 16. Generating the wordcloud:** All the processed words were taken from the prediction data and was visualized using the wordcloud library.
- 17. Visualizing the overall market sentiment of r/cryptocurrency:** After counting the total number of positive and negative posts in step 13. The data was visualized using a bar chart and a pie chart.
- 18. Conclusion:** After looking at the visualizations prepared in step 15, and 17 and the word cloud generated in step 16, a conclusion that the overall outlook of the community is favourable, and bitcoin is the most positively talked about cryptocurrency amongst r/cryptocurrency, was drawn. Hence, the solution concludes that bitcoin is the safest investment according to the market sentiment of r/cryptocurrency subreddit.

3.2. Explanation of the ML algorithm used

The machine learning algorithm used here is a Naïve Bayes Classifier. It is a supervised learning algorithm that uses Bayesian Theorem to classify data into different groups. The algorithm is context agnostic, meaning it regards features present in a dataset independent from one another. For instance, a fruit might be classified as banana if its yellow in colour, has a diameter of 1-1/2 inches, and is elongated. Although these properties might be dependent upon one another, the algorithm treats them as independent features, which is why it has the term 'Naïve' in its name (Ray, 2017). Although Naïve Bayes seems primitive and simple at the outset, it has been shown to trump sophisticated classification methods (Ray, 2017).

Bayesian probability calculates the posterior probability $P(c|x)$ from $P(x)$, $P(c)$, and $P(x|c)$ using the following formula (Ray, 2017):



The diagram shows the Bayesian Probability formula with labels for its components. The formula is $P(c|x) = \frac{P(x|c)P(c)}{P(x)}$. Arrows point from the labels to the corresponding parts of the formula: 'Likelihood' points to $P(x|c)$, 'Class Prior Probability' points to $P(c)$, 'Posterior Probability' points to $P(c|x)$, and 'Predictor Prior Probability' points to $P(x)$.

$$P(c|x) = \frac{P(x|c)P(c)}{P(x)}$$

$$P(c|X) = P(x_1|c) \times P(x_2|c) \times \dots \times P(x_n|c) \times P(c)$$

Figure 2: Bayesian Probability. Source: analyticsvidhya.com.

A basic Naïve Bayes algorithm has the following steps:

1. Convert the annotated dataset to frequency tables.
2. Calculate the individual probabilities of the features present in the dataset.
3. Finally, calculate the posterior probabilities of each features using the aforementioned formula. The feature with the highest overall probability is the result of the algorithm (Ray, 2017).

3.3. Implementation of the algorithm in the solution.

The algorithm in the solution is a binary Naïve Bayes classifier. The steps taken to implement the algorithm are listed below;

1. Firstly, a frequency dictionary was created using the training data.
2. The resulting frequency dictionary and the training data were used as parameters to the training function.
3. The training function (the algorithm) is responsible for calculating two things: the prior probability of a text being positive or negative and the probability of each word in the dictionary being positive or negative.
4. The training function accomplishes its goal in two steps. First, by subtracting the logarithm of total positive texts and the logarithm of total negative texts in the training data, it calculates the prior probability of each text falling into either category. In the second step, the algorithm iterates through each unique word in the dictionary, and stores its probability of being positive and being negative in two separate variables. Finally, log of the ratio of those two variables and word for which the probability is being calculated are stored in a word likelihood dictionary as key, value pairs.
5. Afterwards, a prediction function which takes three parameters: the prior probability log, and the word likelihood dictionary (generated by the training function in step 4), and a text. The prediction function cleans the text and initializes a variable by adding the prior probability log. The function then cleans, tokenizes, and converts each word present in the text to their subsequent lemmas. The words in the processed text are iterated, and their likelihood probabilities log were taken from the likelihood dictionary and added to the variable initialized above.
6. Finally, the prediction algorithm returns the probability of the text falling into either category. If the score is less than 0, the text is considered negative. If it is more than 0, it is considered positive.

3.4. Pseudocode of the proposed solution.

START

IMPORT dataset from Kaggle

INITIALIZE the dataset

VIZUALIZE the dataset

SEPARATE the dataset into a list of positive and negative texts

INITIALIZE stemmer and tokenizer

CLEAN the training data using regular expressions, stemmer and tokenizer

INITIALIZE training data

GENERATE a frequency dictionary using training data

TRAIN Naïve bayes model using frequency dictionary and training data

TEST the algorithm for accuracy

INITIALIZE reddit's API for prediction data

FETCH data from r/cryptocurrency using the initialized API

INITIALIZE the prediction data

CLEAN the prediction data

SEPARATE the data according to different cryptocurrencies

PASS the separated data into the prediction algorithm

IF sentiment GREATER THAN 0

 ADD 1 to the number of positive texts

ELSE

 ADD 1 to the number of negative texts

CALCULATE the positive to negative sentiment ratio of each cryptocurrency

VISUALIZE the sentiment ratio of each cryptocurrency

PASS the overall prediction data through the prediction algorithm

VISUALIZE the number of positive texts vs negative texts.

DISPLAY the visualization

PASS the cleaned prediction data to a word cloud library

DISPLAY the word cloud

END

3.5. Flowchart of the proposed solution.

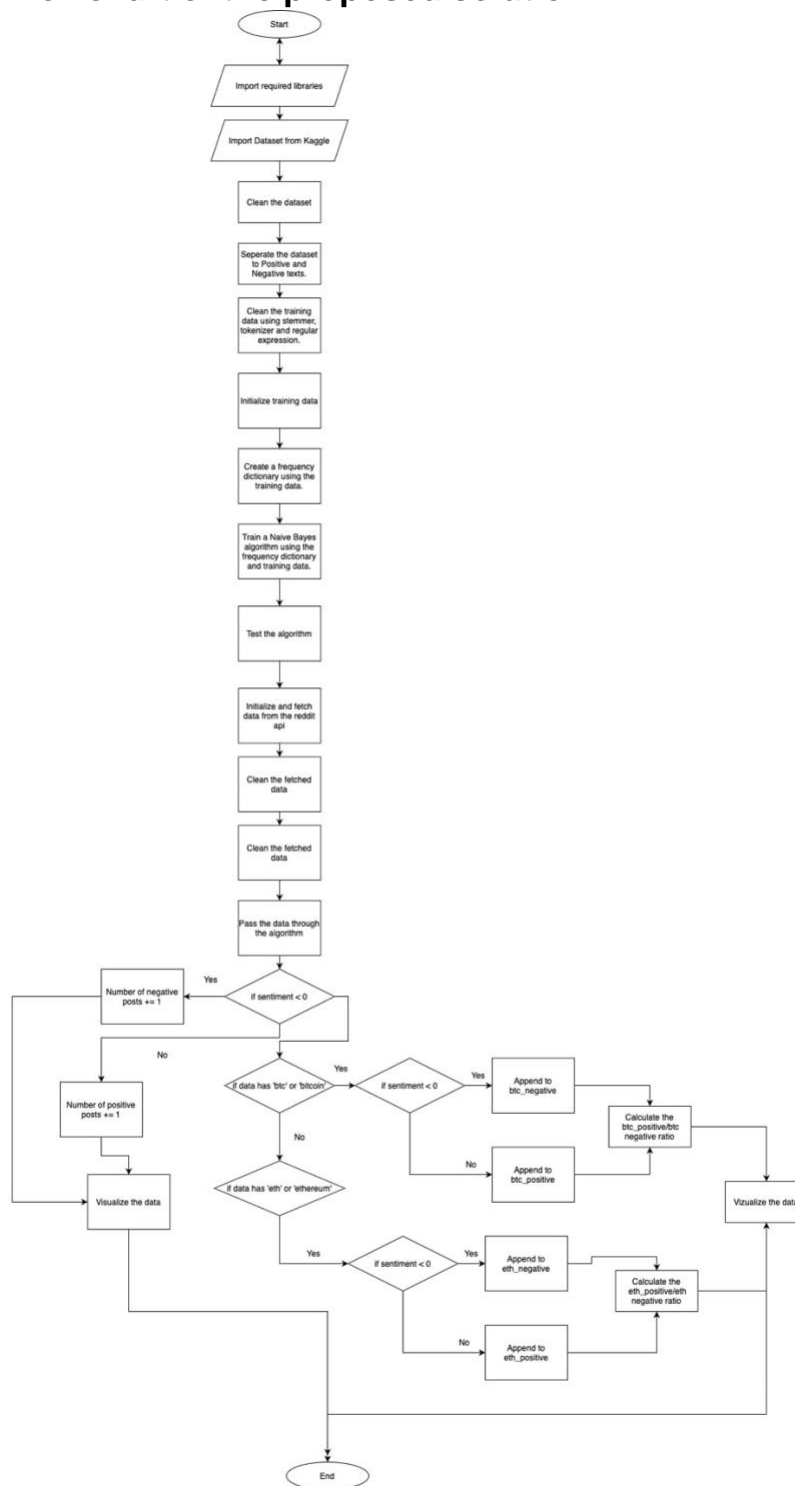


Figure 3: Flowchart of the proposed solution

3.6. Development process.

The following libraries and tools were used in the development of the solution:

1. **Pandas:** Pandas is a popular python library used to work with csv and other forms of numerical tables. It was used in this solution to visualize training and prediction data.
2. **Numpy:** NumPy is another popular python library used for scientific and mathematical computation. It was used in this solution to create sentiment arrays.
3. **Matplotlib:** Matplotlib is a python library used for data visualization. In this solution, it was used to create bar and pie charts.
4. **Time:** Time is a built-in python module used in this solution to calculate the time the computer took to separate the training data.
5. **String:** It is another built in module used to manipulate strings.
6. **Nltk:** Nltk is a popular Natural Language Processing library used mainly in this solution to tokenize, lemmatize, and remove stop words.
7. **Re:** re or regular expression is a built-in library which is used to work with regular expressions. Here, it has been used to remove hyperlinks, hashtags, and punctuations in textual data.
8. **Wordcloud:** wordcloud is another popular python library used here to create a word cloud.
9. **Praw:** Praw is a reddit api wrapper built in python which has been used here to authenticate with and fetch data from the reddit API.
10. **Jupyter notebook:** Jupyter notebook is a notebook used mainly for data science and writing scripts. All the code required for this solution was written in Jupyter notebook.
11. **Kaggle:** Kaggle is a crowd-sourced platform that has a massive number of datasets and solutions to data science and machine learning problems. The dataset used to train the naïve bayes model in this solution was imported from Kaggle.

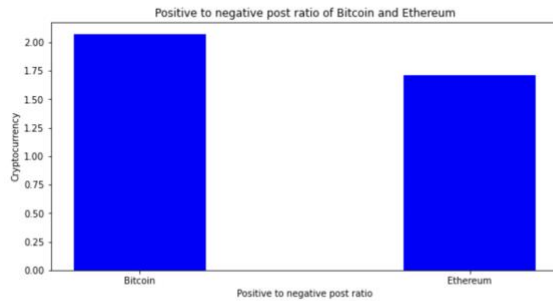
3.7. Achieved results.

Positive – Negative ratio of Bitcoin vs Ethereum.

Visualizing the positive to negative ratio of bitcoin and ethereum

```
In [57]: fig = plt.figure(figsize = (10, 5))
plt.bar(['Bitcoin', 'Ethereum'], [btc_pos_neg_ratio, eth_pos_neg_ratio], color = 'blue',
        width = 0.4)

plt.xlabel("Positive to negative post ratio")
plt.ylabel("Cryptocurrency")
plt.title("Positive to negative post ratio of Bitcoin and Ethereum")
plt.show()
plt.show()
```



Conclusion: The bar above depicts that the community r/cryptocurrency has a more positive outlook for bitcoin than ethereum. Therefore, bitcoin looks like a better investment.

Figure 4: Bar chart depicting the positive to negative sentiment ratio of bitcoin and ethereum.

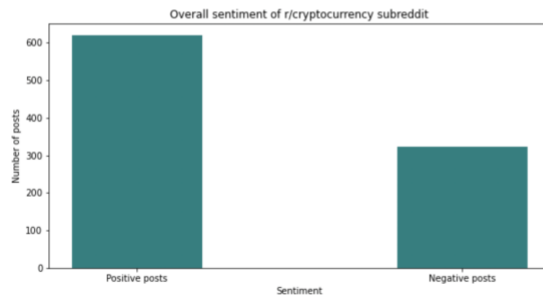
Overall sentiment of the community

Bar Chart depicting the total number of positive vs negative posts

```
In [35]: len_pos = polarity_dict['Positive']
len_neg = polarity_dict['Negative']

fig = plt.figure(figsize = (10, 5))
plt.bar(['Positive posts', 'Negative posts'], [len_pos, len_neg], color = 'teal',
        width = 0.4)

plt.xlabel("Sentiment")
plt.ylabel("Number of posts")
plt.title("Overall sentiment of r/cryptocurrency subreddit")
plt.show()
plt.show()
```



Conclusion: The bar chart shows a larger number of positive tweets than negative tweets. Which means, the outlook of the subreddit r/cryptocurrency is positive overall.

Figure 5: Bar chart depicting the overall positive vs negative sentiment of r/cryptocurrency.

Conclusion drawn from the sentiment analysis

According to the bar and pie chart depicting the current overall outlook of the community r/cryptocurrency, the bar chart showing the positive to negative ratio of Bitcoin and Ethereum, and the word cloud, the following things could be concluded:

1. The overall outlook of the community r/cryptocurrency is positive as a whole.
2. Bitcoin is the most discussed cryptocurrency.
3. The positive to negative sentiment ratio is high for bitcoin, compared to Ethereum.

Therefore, although the prices of cryptocurrencies have been declining, people have positive outlook regarding bitcoin. Which means, bitcoin looks like a safe investment.

4. Conclusion.

4.1. Analysis of my work.

This solution in this project was implemented using a binary naïve bayes classifier. It was trained using a dataset of 800,000 texts, and was used to predict the sentiment of the cryptocurrency community in reddit called r/cryptocurrency.

Albeit this project seemed simple at the beginning, it amounted to a large codebase and I had to solve multiple bugs to run the program successfully. Conceptualizing, and creating a mental model of the algorithm itself took a lot of time, and I struggled before finally understanding it.

Attempting this project taught me many concepts that I would not have otherwise looked into. I not only learned about Tokenization, Lemmatization, Frequency dictionaries, Classifier algorithms, but also developed an overall fascination with the field of Natural Language Processing. Although the solution implemented here is not production grade, and still has many shortcomings, it helped me sense the overall outlook of the reddit cryptocurrency community.

4.2. How the work solves real world problems.

The government of Nepal has banned the trading of Cryptocurrencies. However, people outside of this country can use the model trained here to analyse the current market sentiment of the cryptocurrency community to predict whether the price of certain cryptocurrencies will move up or go down. They could also see whether the cryptocurrency community itself is optimistic or pessimistic regarding its future. In combination, the data gathered from analysing the market sentiment will help traders and investors make thoughtful decisions, and help them spend their money wisely.

4.3. Limitations of the solutions.

There are multiple limitations in the solution implement in this project. The most important of them are listed below:

1. The prediction data is insufficient to make any kind of large-scale decision. It suffers from what the Behavioral Psychologist Daniel Kahnemann calls the 'Law of least numbers.' According to which, having a smaller number of data always results in a biased statistical conclusion.
2. Many people might have a positive outlook regarding cryptocurrencies not as an asset but as the upcoming revolutionary technology. Therefore, the positive sentiment might not be predictive of asset prices.
3. The total number of posts mentioning Ethereum is low. Therefore, the positive to negative sentiment ratio might not be accurate.
4. Due to the inconsistencies present in the trained model, the polarity score predicted by the model might not be fully accurate.
5. The sentiment of a single community is not generalizable to other communities.

4.4. Future work.

Further implementation of this solution that could be carried out in the future are:

1. Performing sentiment analysis on a larger and subsequently more accurate dataset.
2. Implementing a time-series analysis of all the posts and observe the sentiment trend on a larger timeframe.
3. Collect prediction data from more sources to make generalizable prediction.
4. Implementing a deep learning or LSTM model on the prediction dataset to make better sentiment analysis.

References

- Buchanan, B. G., 2006. A (Very) Brief History of Artificial Intelligence. *AI Magazine*, 26(4), pp. 53-60.
- Dey, N., Borah, S., Babo, R. & Ashour, S. A., 2019. *Social Network Analytics: Computational Research Methods and Techniques*. 1st ed. s.l.:Academic Press.
- Techtarget, 2021. *Techtarget*. [Online]
Available at: <https://searchbusinessanalytics.techtarget.com/definition/opinion-mining-sentiment-mining>
[Accessed 21 December 2021].
- Desjardins, 2021. *Desjardins*. [Online]
Available at: <https://www.disnat.com/en/learning/trading-basics/stock-basics/what-causes-stock-prices-to-change>
[Accessed 21 December 2021].
- Cowan, N., 2010. The Magical Mystery Four: How is Working Memory Capacity Limited, and Why?. *Curr Dir Psychol Sci*, 19(1), pp. 51-57.
- Ortigosa-Hernandez, J. ´. et al., 2012. Approaching Sentiment Analysis by using semi-supervised learning of multi-dimensional classifiers. *Neurocomputing*, Volume 92, pp. 98-115.
- Stoy, L., 2021. *lazarinastoy*. [Online]
Available at: <https://lazarinastoy.com/sentiment-analysis-theory-methods-applications/>
[Accessed 22 December 2021].
- Vajjala, S., Majumder, B., Surana, H. & Gupta, A., 2020. *Practical Natural Language processing*. 1 ed. California: O'Reilly.
- Thakkar, H. & Patel, D., 2015. *Approaches for Sentiment Analysis on Twitter: A State-of-Art study*, Surat, India: Department of Computer Engineering, National Institute of Technology.
- Cen, P., 2020. *Predicting Consumers' Brand Sentiment Using Text Analysis on Reddit*, Pennsylvania: University of Pennsylvania.
- Lubitz, M., 2018. *Who drives the market? Sentiment analysis of financial news posted on Reddit and the Financial Times*, Freiburg: University of Freiburg.
- Salač, A., 2018. *Forecasting of the cryptocurrency market through social media sentiment analysis*, Twente, Netherlands: University of Twente.
- Xu, M., 2021. *NLP for Stock Market Prediction with Reddit Data*, California: Stanford University.

Ray, S., 2017. *Analyticsvidhya*. [Online]

Available at: <https://www.analyticsvidhya.com/blog/2017/09/naive-bayes-explained/>

[Accessed 22 December 2021].