Revision Sheet

TABLE OF CONTENTS

1 INTRODUCTION

Accumulation of data has increased to a staggering level over the years of which nearly 18 billion text information is stored every day. This huge amount of information calls for machine learning methods to extract and analyze the data to retrieve meaningful topics, provide unique insights into texts and give you new ways to organize documents.

Topic Modeling is an information retrieval method which is the way of uncovering latent topics present in bag of data. This started in the early 1990's has wide range of applications of which used by the government to discover opinions on their services, companies to increase their revenue, to mine the sentiments from the social media, to figure out the topics and relationships from the collection of documents or artifacts etc.,

At present one such area that shoots for topic modeling is cryptocurrency. In recent times the number of users attracted to it has spiked up tremendously. As a result, a huge amount of information is generated regarding this topic and there is a need to analyze the information to extract the current market trends or to unravel any hidden relationship between corpus of words.

The aim of this project is to discover the abstract topics that occur in the collection of data from Reddit posts and comments of different users based on the discussions on various Cryptocurrency. This proposed approach will be able to draw out and classify the posts and comments to different topics and cluster the frequent similar information together. We will also be seeing how the sentiment of the daily discussions/posts impact the change in the bitcoin prices by checking the correlation between them.

2 PROBLEM STATEMENT

The objective of this project is to extract semantic topic information which is prevalent in cryptocurrency posts using various topic modelling algorithms and compare each of them to find which algorithm clusters the messages into separable and interpretable topics or groups and then see if these topics are correlated with the daily price change in Bitcoin.

2.1 Challenges

- One of the main challenges is that most of the data is present in the form of short text formats (from the comments and topic posts) which makes it tough to extract topics.
- To extract proper semantics from these short texts while making sure the sparsity and dynamicity of these short texts are captured.
- Another challenge is to use appropriate preprocessing techniques to ensure that the data is neither overfit nor underfitted as there are lot of short texts which would result in data loss.
- Next challenge is to estimate the number of important topics that are to be extracted from the posts and comments. When using the LDA algorithm there is a need to specify the number of latent topics. Therefore, an optimum number of topics must be figured out.
- Another challenge which is anticipated when we clean and remove stop words, unwanted symbols, numbers etc. A lot of the posts in our dataset are comments with a lot of such symbols or exclamations (e.g.: a thumbs up symbol in response to a post). These types of posts also must be identified and removed.
- Lastly, we also have words that are either obvious or do not hold much significance while representing the most frequent words in extracted semantic topics. We need to analyze the results and prune these words.

2.2 Related Works

The overall aim of this scientific research reading review is to aid predictive analytics project of developing social graphs and topic models. Hence, the research papers reviewed in this paper review are related to either social graphs or ways to discover abstract topics that occur in a collection of posts/comments among different users.

(Ghoorchian et al. 2020) proposed Graph-Based Dynamic Topic Models (GDTM) [1] a single-pass graph-based topic model which addressed the scalability, dynamicity, and scalability of short texts to extract topics. Another proposed model by Akhtar et al. is User Graph Topic Model [4], it is more inclined on finding relationship between users given that most of the users do not make multiple posts and one post is made only by one user. Using hashtags, user mentions and comments the user graph is developed which displays the related user information. (Golino H et al. 2021) proposed Dynamic Exploratory Graph Analysis (DynEGA) [6] which can overcome manual inputting number of important simulated topics (as in the case of LDA) by automatically identify the number of subjects and the distribution of variables (words) per topic. (Chenliang Li et al. 2016) [13] proposes a few approaches that can be used to handle data sparsity, especially with respect to co-occurrence of words. (Yuan Zuo et al. 2016) [14] talks about how using a pseudo document, created by combining short texts leads to better topic modeling results. (Kuldeep Singh et al. 2016) [15] proposes using WordNet to group words of similar meaning together. To find similarity between texts k-means and spectral k-means algorithms are used. In a research paper as proposed by (R. Churchill and L. Singh. 2022) [9] describes how the topic modeling has transformed over the years, the flow of change in the algorithms from first models DMM and LDA to the modern NLP (Natural Language Processing) methods. (Asmussen, C.B., Møller, C. 2019) [10] in a research paper states the extraction of topics from the video transcripts using the graph data structures based on edge connectivity. (Jason Thies et al. 2022) [11] describes another approach called BerTopic which uses neural topic modeling with a class-based TF-IDF procedure which discovers the latent topics from the collection of documents.

While the above discussed publications propose different techniques used for topic modelling, in our approach we have implemented and compared the results of three models namely LDA, LSA (Latent Semantic Analysis), PLSA (Probabilistic Latent Semantic Analysis) while measuring their perplexity and coherence score. We also have used a unique approach of integrating topic modelling with changes in bitcoin prices, to check the correlation between the two. This approach helps to point out the differences between topics discussed when there are changes in the prices of bitcoin.

Data we used for our implementation is scraped from Reddit using APIs (PRAW and PSAW).

2.3 Importance

- Extracting the frequent topics from the corpus of information is useful to understand which topics are being discussed when there is any change in the cryptocurrency market and forecast which topics were likely to be discussed further.
- A connection can either be established between the topics extracted and the prices of the cryptocurrency, volume traded which helps the investors to switch their plans of investment.
- Cryptocurrency is the news all over the time and a huge number of people are investing and discussing this. Many giant people are in the path of cryptocurrency as we have seen how a single tweet from Elon Musk can shake the whole market which signifies the social and business impact this has.

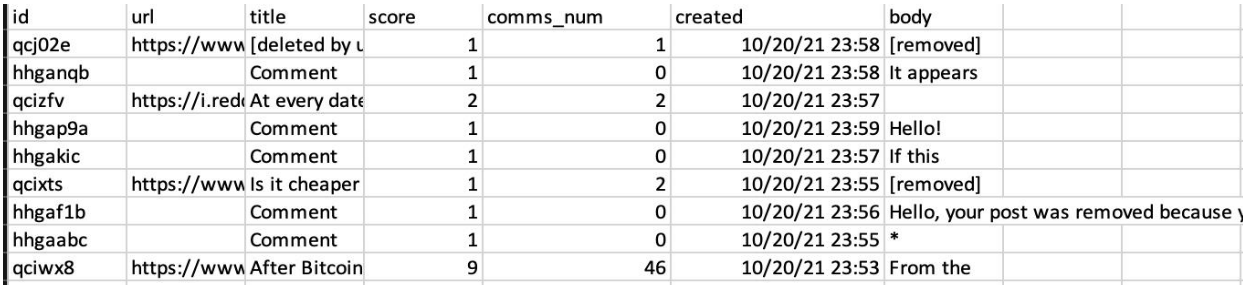
3 DATA COLLECTION

- As the goal of the project is to extract the topics from the reddit posts, the reddit APIs are used to scrape the data from the social media platform.
- To achieve the same, the PRAW and the PSAW wrappers present in the python are used where PRAW is the reddit API and the PSAW is used for searching public reddit comments/submissions via the pushshift.io API.
- The search_submissions function present in the PSAW is used to extract the posts and comments where we specify the topic name that must be extracted along with the timeline between which we need the posts.
- The above function gives the different ids of the posts that were posted during the timeline using the same ids in the PRAW we retrieve the other information like post body, created on, score, comments number etc.,

- After all the important information is retrieved the data is then converted into the data frame and then imported into a CSV file.
- Another wrapper named PMAW is also used to extract the data over a long period of time as this wrapper is optimized to extract large amounts of data faster than PSAW.
- The PMAW uses search_comments function and is connected to the same pushshift.io API.
- Along with the 4 months reddit data (October 2021 to January 2022), the bitcoin prices for the same period as of the reddit posts were extracted.
- In the reddit dataset there are 237191 no of records with 8 no of columns with lot missing values in the body and title columns.
- And, in the bitcoin dataset there are 275 no of records with 7 no of columns without any missing information.
- As the research question is to find the frequent bitcoin topics from the reddit posts and discussion, the data that was extracted either the reddit posts or prices is pertinent.

Serial Number	Columns	Description
1.	Title	Title of the posts made by the users (relevant for posts)
2.	Score	Post/comment score (relevant for posts). It is calculated on reddit impact and number of comments
3.	ID	unique id for posts/comments
4.	URL	URL of the thread of the post on Reddit
5.	Commns_num	Total number of comments of that post
6.	Created	Date at which the post/comment was created
7.	Body	The contents of the post or comment

Table 1: Description of the columns of Reddit posts dataset



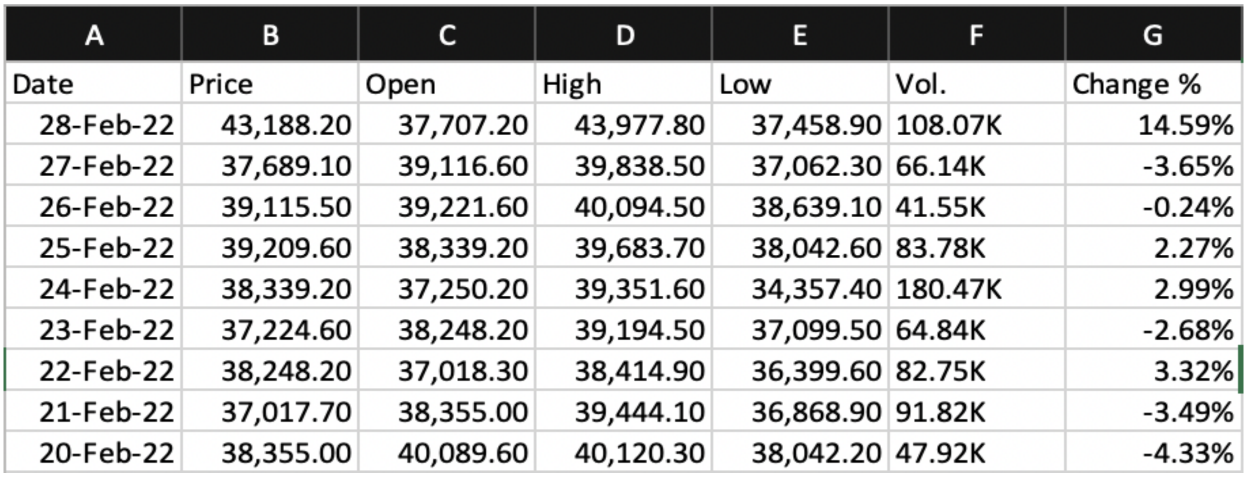
id	url	title	score	comms_num	created	body
qcj02e	https://www	[deleted by u	1	1	10/20/21 23:58	[removed]
hhganqb		Comment	1	0	10/20/21 23:58	It appears
qcizfv	https://i.red	At every date	2	2	10/20/21 23:57	
hhgap9a		Comment	1	0	10/20/21 23:59	Hello!
hhgakic		Comment	1	0	10/20/21 23:57	If this
qcixts	https://www	Is it cheaper	1	2	10/20/21 23:55	[removed]
hhgaf1b		Comment	1	0	10/20/21 23:56	Hello, your post was removed because y
hhgaabc		Comment	1	0	10/20/21 23:55	*
qciwx8	https://www	After Bitcoin	9	46	10/20/21 23:53	From the

Fig 1: Screenshot of the Reddit posts dataset after scraping

Serial Number	Columns	Description
1.	Date	Date on which the price is recorded
2.	Price	Bitcoin price on the date
3.	Open	The opening bitcoin price for the date
4.	High	The highest price for the date
5.	Low	The lowest price for the date

6.	Vol	The transacted volume of the bitcoins
7.	Change %	The percentage change of the market from the previous day

Table 2: Description of the columns of Bitcoin Prices dataset



A	B	C	D	E	F	G
Date	Price	Open	High	Low	Vol.	Change %
28-Feb-22	43,188.20	37,707.20	43,977.80	37,458.90	108.07K	14.59%
27-Feb-22	37,689.10	39,116.60	39,838.50	37,062.30	66.14K	-3.65%
26-Feb-22	39,115.50	39,221.60	40,094.50	38,639.10	41.55K	-0.24%
25-Feb-22	39,209.60	38,339.20	39,683.70	38,042.60	83.78K	2.27%
24-Feb-22	38,339.20	37,250.20	39,351.60	34,357.40	180.47K	2.99%
23-Feb-22	37,224.60	38,248.20	39,194.50	37,099.50	64.84K	-2.68%
22-Feb-22	38,248.20	37,018.30	38,414.90	36,399.60	82.75K	3.32%
21-Feb-22	37,017.70	38,355.00	39,444.10	36,868.90	91.82K	-3.49%
20-Feb-22	38,355.00	40,089.60	40,120.30	38,042.20	47.92K	-4.33%

Fig 2: Screenshot of the Bitcoin Prices dataset

4 DATA PREPROCESSING

- For topic modelling, the columns we were interested in are “Body” and “Text”. In quite a lot of cases, one or both columns had missing values
- There were a lot of missing values in other critical columns like “score” and “created”
- The column “created” had date values in many different formats, we had to standardize this and bring it into one single format
- A new column “weightedScore” was created by normalizing the column score. This will help us understand the impact, if any, a particular post had. It would also be helpful when checking for correlations
- Calculated columns using the “weightedScore” and polarities calculated from the text were also created to better understand the impact of posts
- The values of the column “score” varied greatly in a range of [20000 to –67]. Only very few posts were near the upper limit. We could not however remove these posts as the posts with the highest score are generally the most influential and make the highest impact
- As Reddit posts are social media posts and are user generated, there would generally be a lot of special characters and symbols in the text like “&”, “$” etc. These were removed using regez
- Removed several comments that were either empty or contained only emojis.
- Accentured characters have also been taken care of.
- Numbers have also been removed from the text fields
- Stop words have been removed using the standard NLTK stop word set. Other words which we found to be irrelevant were also removed
- The data has been Lemmatized and Stemmed
- Created document term matrix of the final list of cleaned texts
- Records with certain phrases e.g. (“Post Removed", "deleted”, “comment”) must be removed.

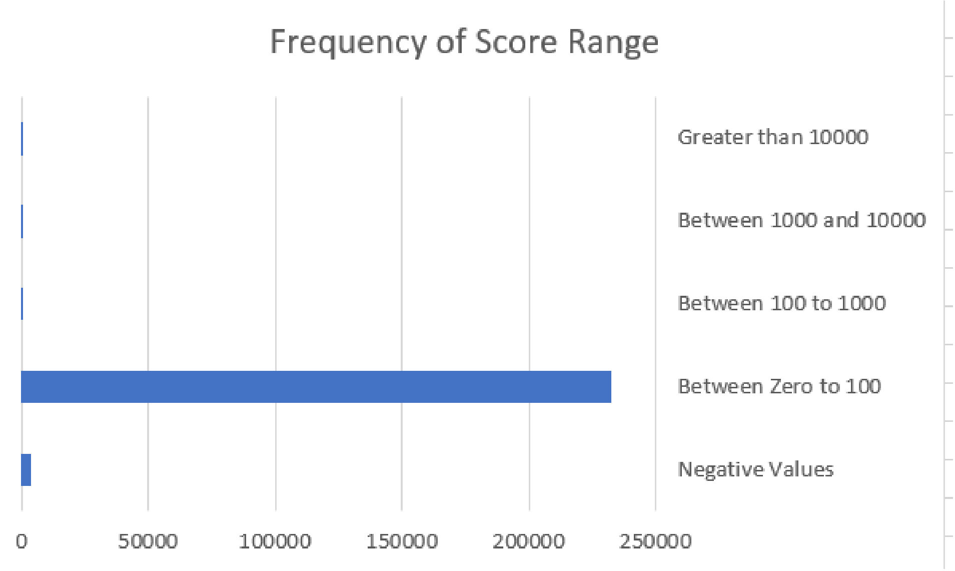


Fig 3: The Frequency chart of the score column of reddit posts dataset

Exploratory Data Analysis:

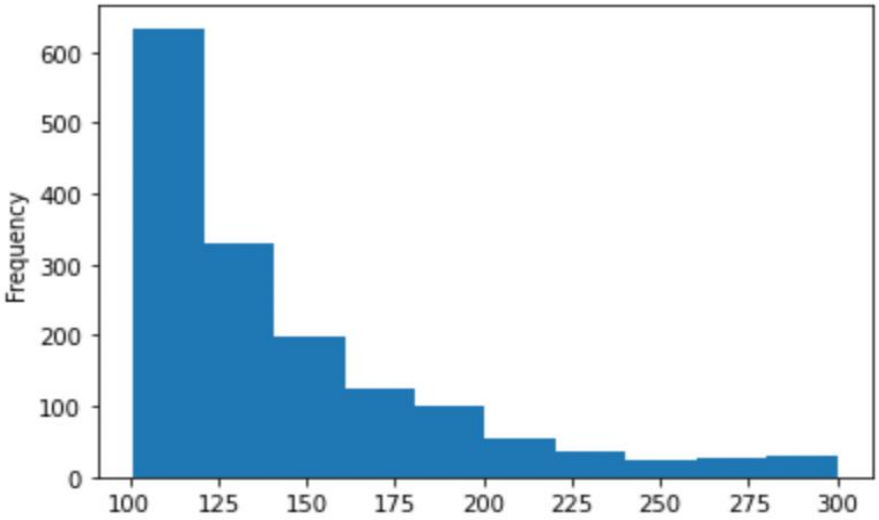


Fig 4: The histogram showing the comparison of no of different lengths of the title column.

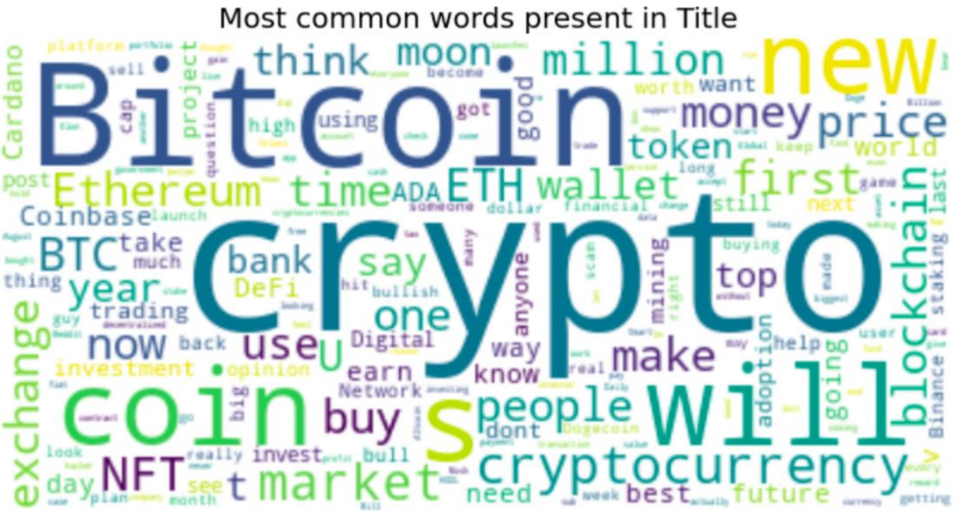


Fig 5: The word cloud exhibiting the recurrent words present in the title column

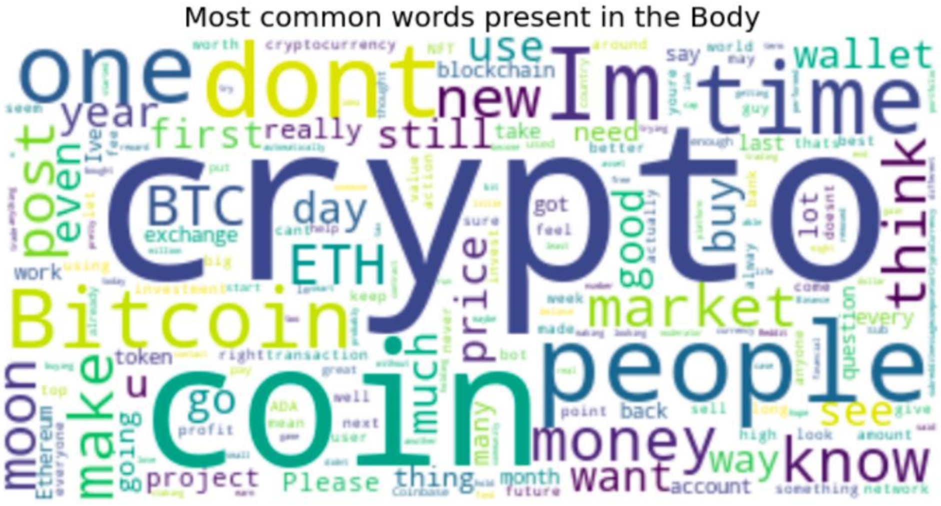


Fig 6: The word cloud exhibiting the recurrent words present in the body column

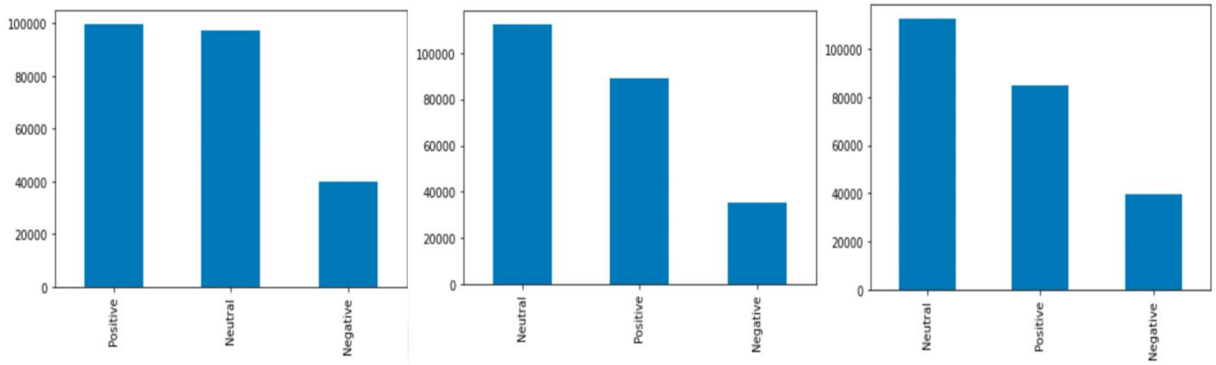


Fig 7: The count of Positive, Negative and Neutral Sentiments in the dataset (Vader, TextBlob, AFINN)

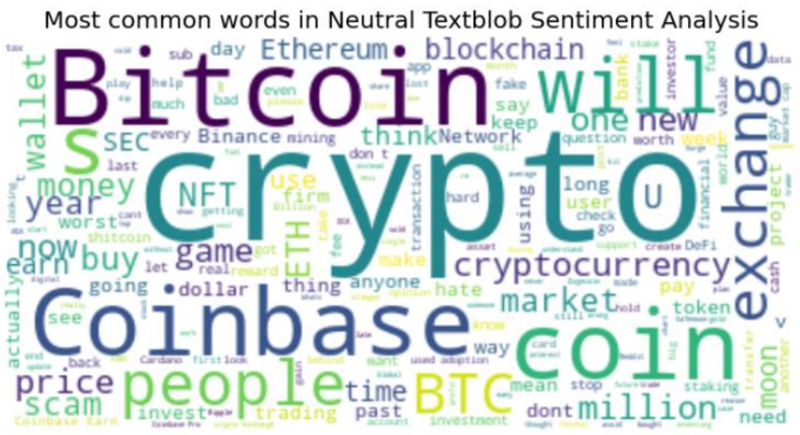


Fig 8: The word cloud exhibiting the recurrent words present in the Neutral TextBlob Sentiment Analysis

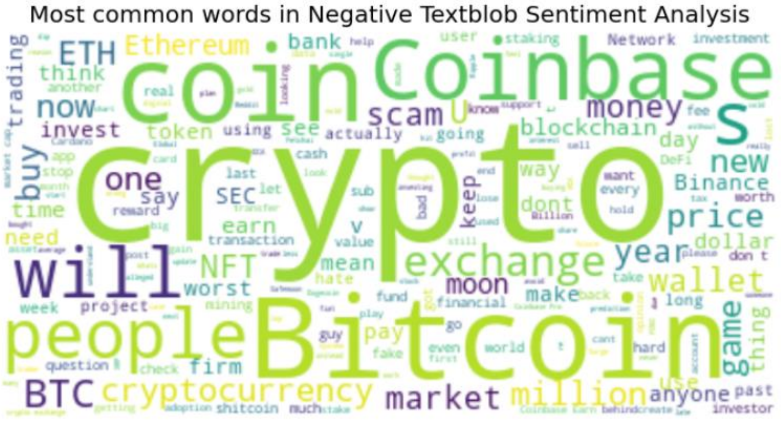


Fig 9: The word cloud exhibiting the recurrent words present in the Negative TextBlob Sentiment Analysis

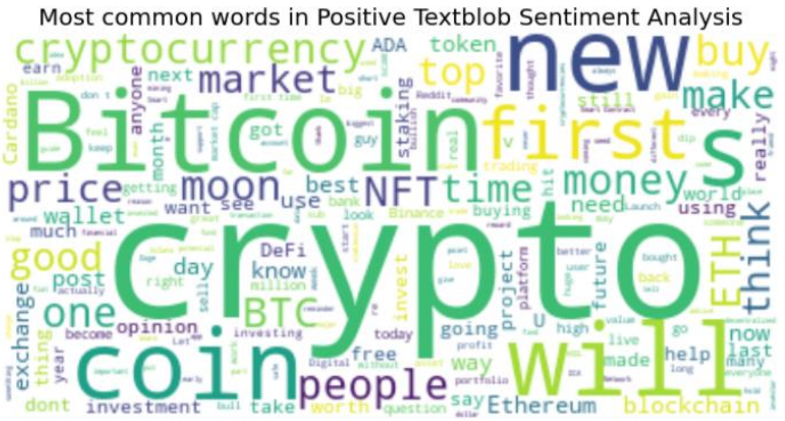


Fig 10: The word cloud exhibiting the recurrent words present in the Positive TextBlob Sentiment Analysis

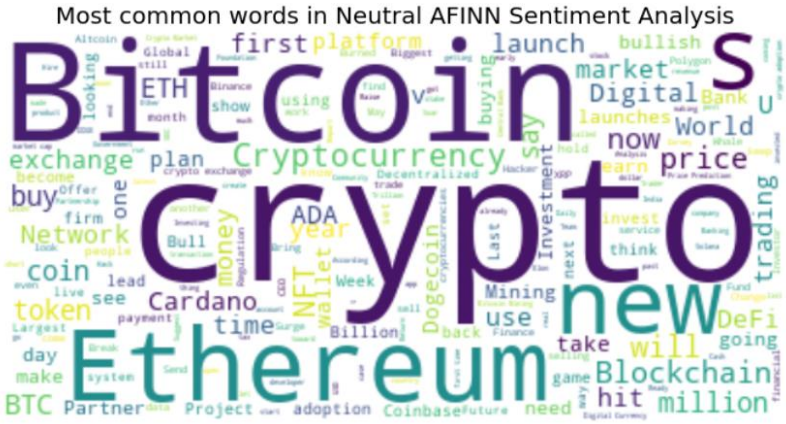


Fig 11: The word cloud exhibiting the recurrent words present in the Neutral AFINN Sentiment Analysis

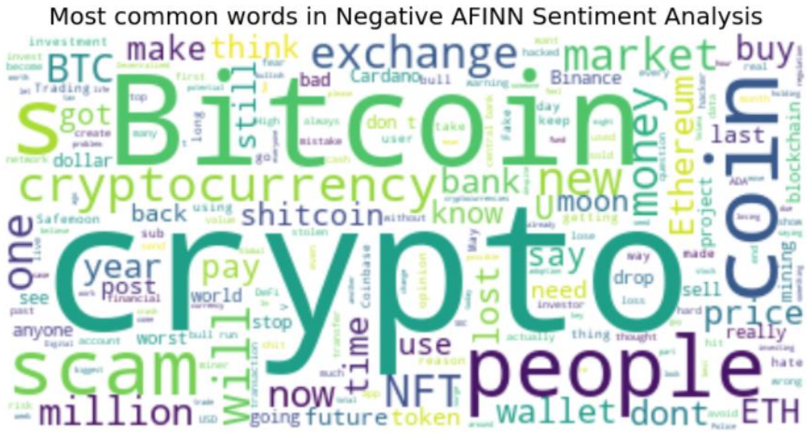


Fig 12: The word cloud exhibiting the recurrent words present in the Negative AFINN Sentiment Analysis

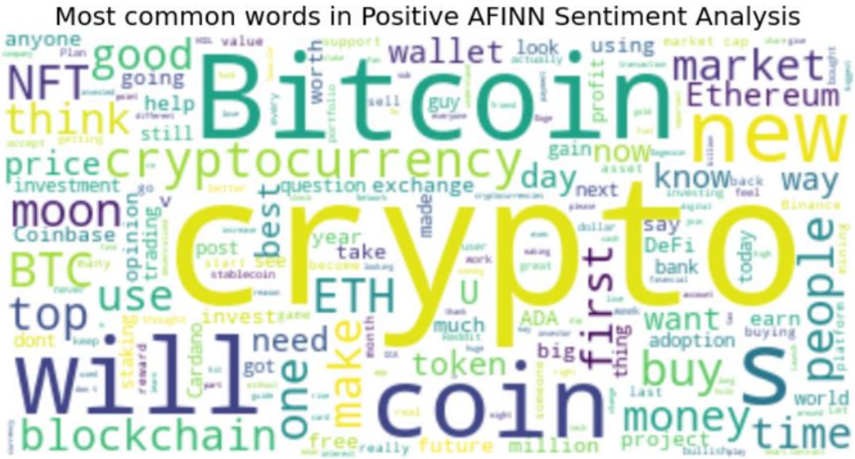


Fig 13: The word cloud exhibiting the recurrent words present in the Positive AIFNN Sentiment Analysis

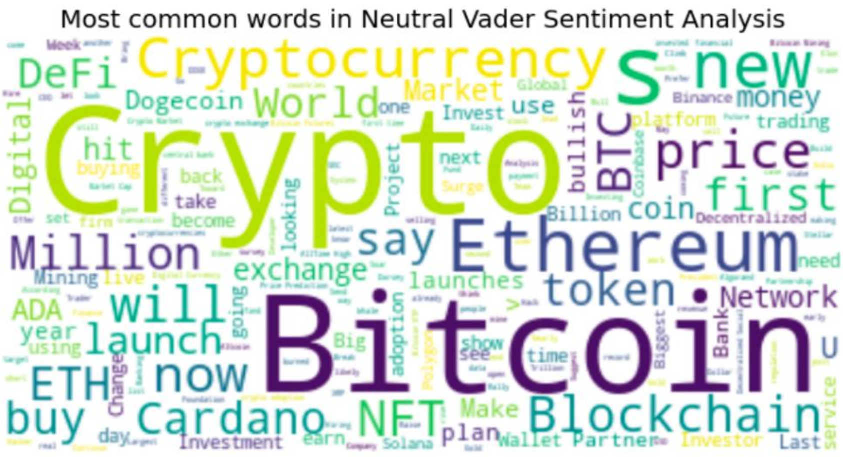


Fig 14: The word cloud exhibiting the recurrent words present in the Neutral Vader Sentiment Analysis

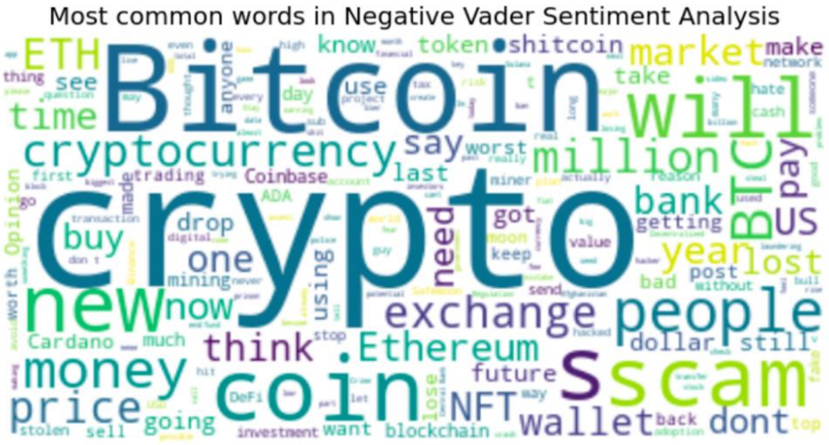


Fig 15: The word cloud exhibiting the recurrent words present in the Negative Vader Sentiment Analysis

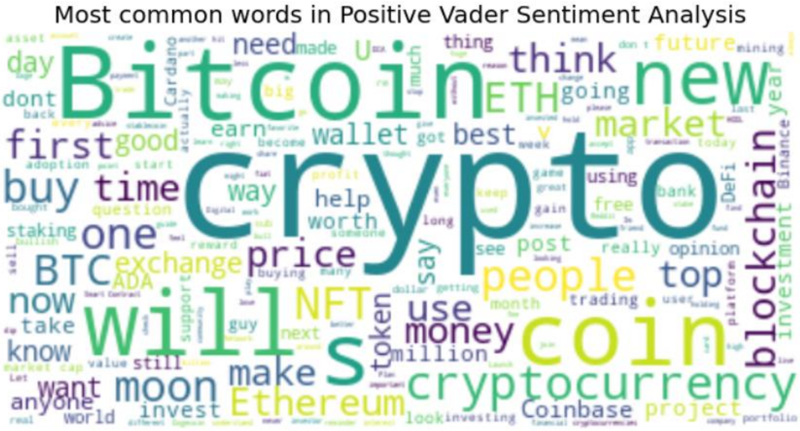
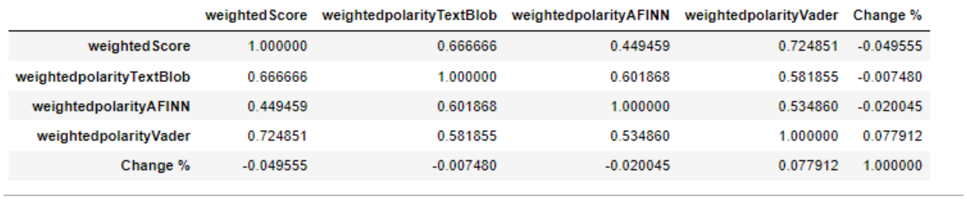


Fig 16: The word cloud exhibiting the recurrent words present in the Positive Vader Sentiment Analysis



	weightedScore	weightedpolarityTextBlob	weightedpolarityAFINN	weightedpolarityVader	Change %
weightedScore	1.000000	0.666666	0.449459	0.724851	-0.049555
weightedpolarityTextBlob	0.666666	1.000000	0.601868	0.581855	-0.007480
weightedpolarityAFINN	0.449459	0.601868	1.000000	0.534860	-0.020045
weightedpolarityVader	0.724851	0.581855	0.534860	1.000000	0.077912
Change %	-0.049555	-0.007480	-0.020045	0.077912	1.000000

Fig 17: The correlation matric between Change in Price and Weighted scores and Polarity. There is no correlation between the variables

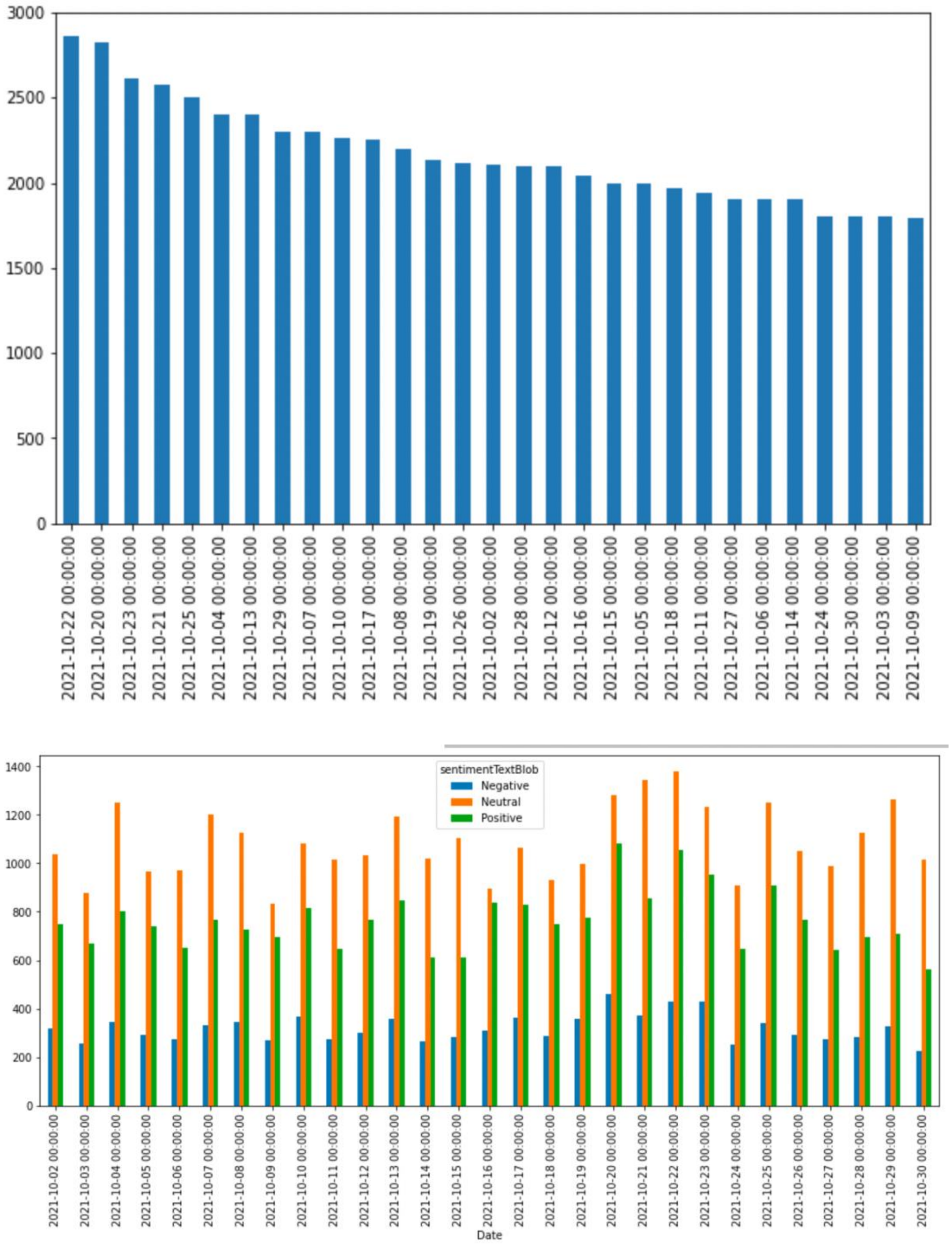


Fig 18 & Fig 19: The Number of posts per day to the sentiments observed on the same dates in the month of October 2021.

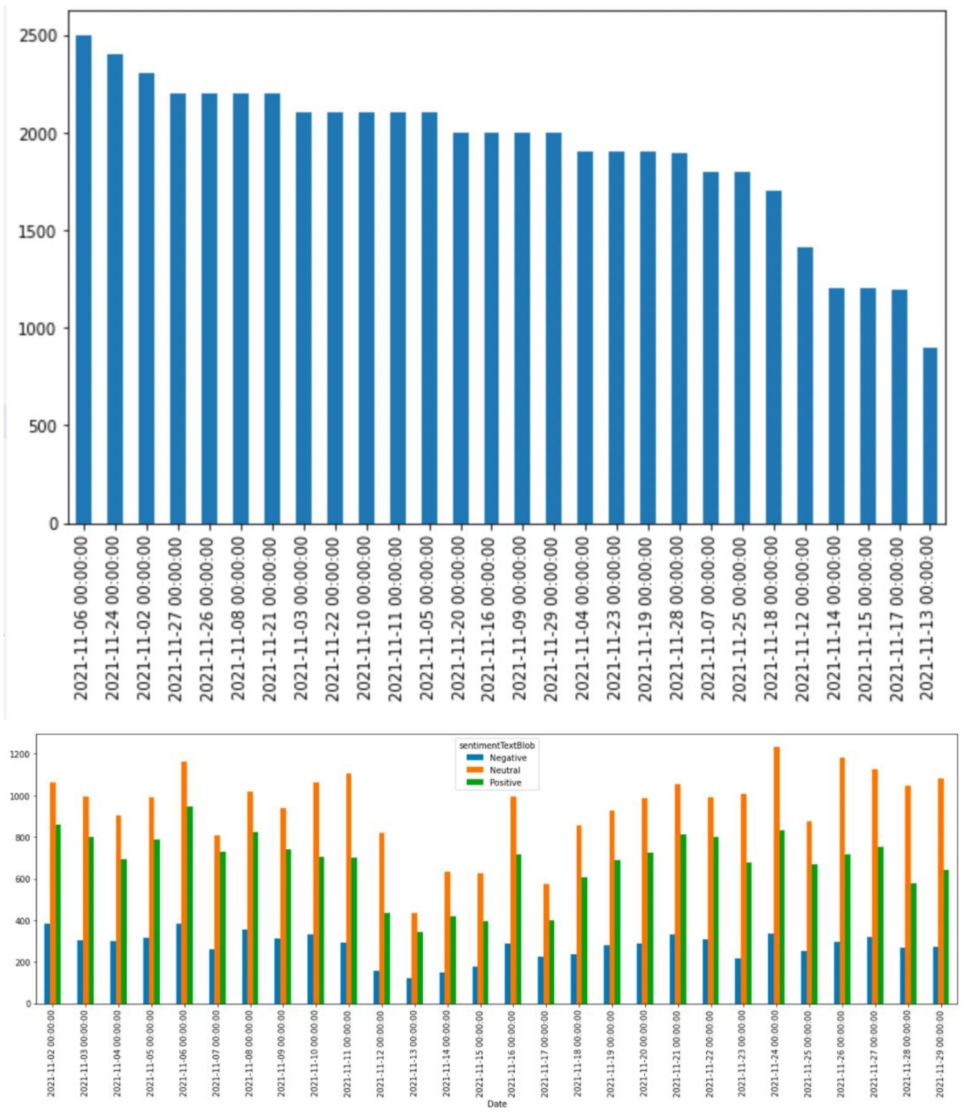


Fig 20 & Fig 21: The Number of posts per day to the sentiments observed on the same dates in the month of November 2021.

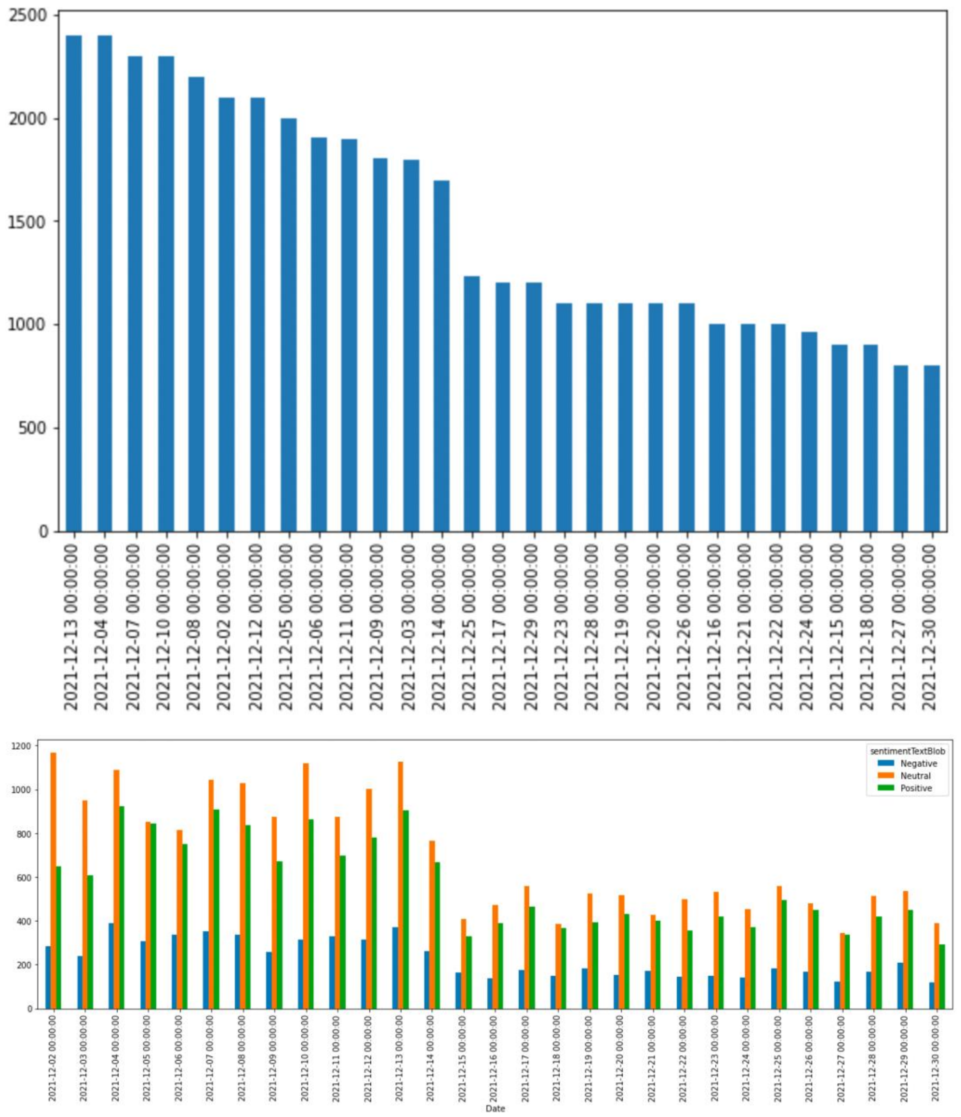


Fig 22 & Fig 23: The Number of posts per day to the sentiments observed on the same dates in the month of December 2021.

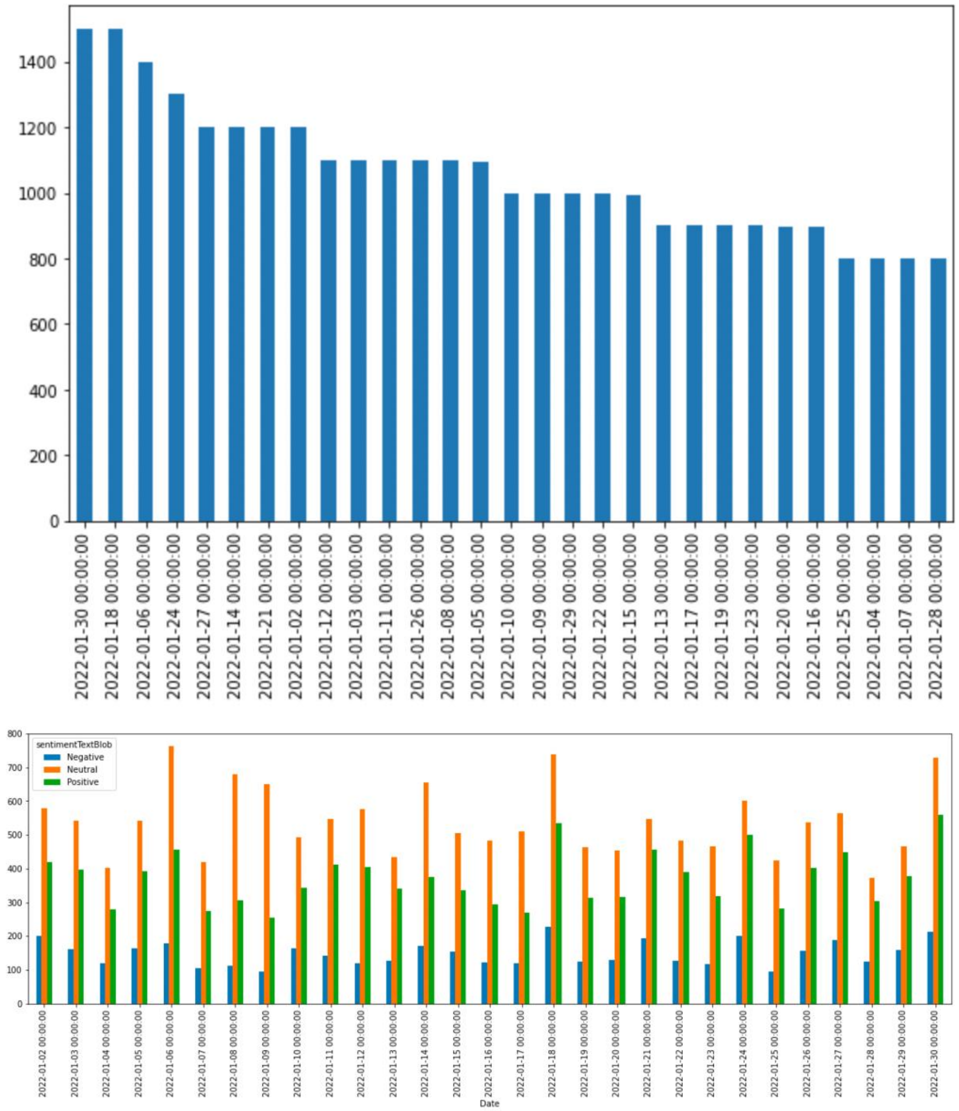


Fig 24 & Fig 25: The Number of posts per day to the sentiments observed on the same dates in the month of January 2022.

5 METHODOLOGY

These are the main methodologies that we have used in our approach:

- Topic modelling
- Sentiment analysis
- Correlation
- Feature scaling

For topic modelling we used LSA, LDA and pLSA as they are the best performing models and most popular available. Since topic modeling over short text is challenging, these models give the best results in segregating the discussions into multiple topics. These modes are good at extracting topics out of text, even taking the sparsity of texts into account. However, this approach of extracting topics of discussions is not a great parameter for prediction of changes in bitcoin prices. We have used various libraries in Python for developing the project.

The assumption/constraints we consider while applying these methods on Reddit data are:

- Sparsity of data
- Unclean data
- Our document contains various topics in it but one specific topic in a document has more weightage
- So, we're more likely to choose a mixture of topics where one topic has a higher weightage

Steps followed in our approach:

- First, we extracted data from Reddit (Posts and discussions) for four months. We extracted daily posts about Bitcoin from reddit for 4 months and the data for bitcoin price changed in the same timeframe.
- The fields "posts" and "comments" were combined to create a new text field. This made the length of the text larger and would theoretically improve the performance of the topic models. This field thus created was cleansed by:
    - Using Regex, symbols, special characters and numbers were removed
    - Regex was also used to remove words that were to us. Eg "Deleted", "Post Removed" etc.
    - Using the NLTK stop words package, stop words were removed from the text
    - Using NLTK functions, the text was tokenized, Stemmed and Lemmatized
- The "created" field contained dates in different formats, some were strings, others were datetime, some were as timestamps. Using Pandas functions like asType() , and datetime packages in python, the date field was standardized.
- In our approach we feature engineered a few attributes:
    - A new field combining two text columns "title" and "body" was created to produce longer texts that will help improve the topic modeling
    - Date component has been extracted from the field "created"
    - A new column "weightedScore" was created by normalizing the column score.
    - Calculated columns using the "weightedScore" and polarities calculated from the text were also created to better understand the impact of posts
- Then we used several topic modelling techniques like LDA, LSA (Latent Semantic Analysis) and pLSA for topic modelling.
- As proposed, we will be implementing three different topic modelling techniques to compare the results within each of them:
    - Latent Dirichlet Allocation (LDA)
    - Latent Semantic Analysis (LSA)
    - Probabilistic latent semantic analysis (PLSA)
    - Sentimental analysis (TextBlob)
- First, we built an LDA model. Basically, LDA is like PCA, but it focuses on maximizing the separability among categories/topics. PCA is for reducing the dimensions by focusing on attributes with most variations. It is mainly used when we need to plot high dimensional data onto a simple xy plane. But in our case, we are not interested in attributes(documents) with most variations, instead we are interested in maximizing the separability between K topics while minimizing the variability within topics. Basically, LDA (Latent Dirichlet Allocation) is like PCA, but it focuses on maximizing the separability among categories. This is done by projecting the data onto new axis in a way to keep the importance of each attribute and to maximizing the separation. This is done

by maximizing the distance between mean of the topics and the center point and by minimizing the variation within each category. As we know, the first axis of PCA accounts for most variation in data, but in the case of LDA it accounts for the most variation between the categories.

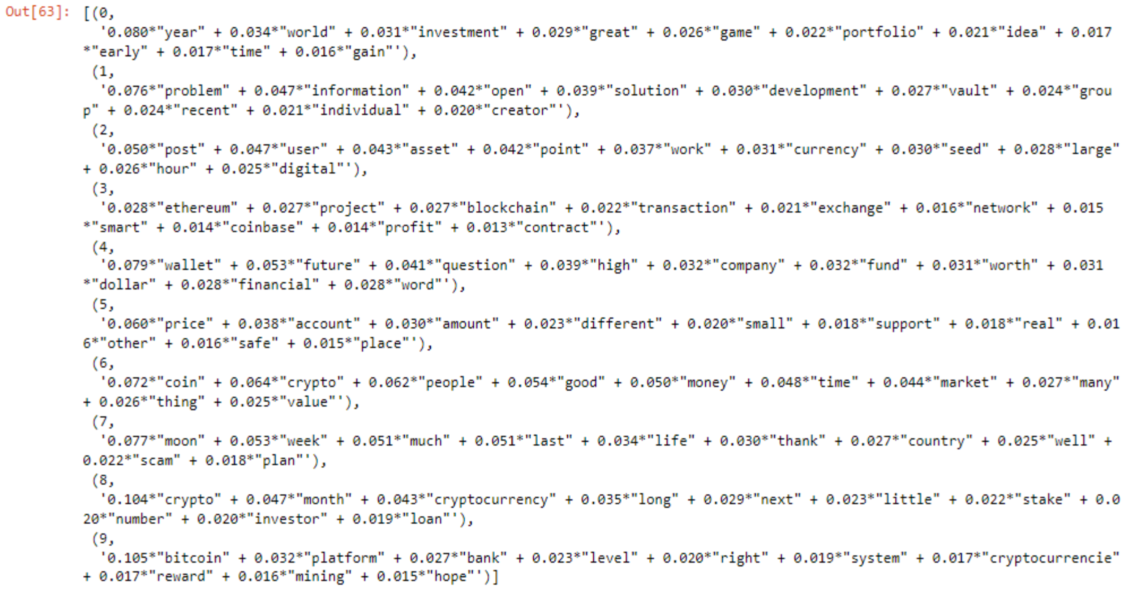
- To do so we started with baseline model of LDA with taking 10 topics into consideration.
- After which we made an interactive graph to observe the results. Followed by evaluating the performance of base model with coherence score and perplexity.
- The lower the perplexity the better. the higher the coherence score the better the model is performing
- After doing the initial topic modelling we still came across some words that were not useful, so we removed them
- We used perplexity and coherence score as topic modelling metric to compare the various models.
- Since for topic modelling using LDA we always need to specify the number of topics that need to be extracted, hence, to find the optimum topic number we compared the results of different topic numbers with coherence scores and found out that 3 topics were giving the optimum results.
- Then we rebuilt the model with the optimum number of topics and saw an improvement in the coherence score.
- To further improve the performance of our model we did grid search hyper parameter tuning on our LDA model with 3 topic heads. The hyper parameters considered here were alpha and eta. After running grid search, we found out the best values for the two were 0.91 and 0.01 respectively.
- Next, we implemented another topic modelling approach called LSA. The goal of LSA is to reduce the number of dimensions for categorization. The idea is that words with comparable meanings will appear in related pieces of literature.
- As LSA using scikit learn does not offer liberty to evaluate the performance either though coherence score or perplexity, only way we can conclude that this LSA model isn't performing well is by checking the separation between topics. As they overall fail to accomplish the principal reason of separability of topic modelling, we can say that LDA produced better results. LSA decomposed matrix is a highly dense matrix, so it is difficult to index individual dimensions and is unable to capture the multiple meanings of words.
- We implemented LSA with two different approaches (one with gensim library and the other with sklearn) as gensim library gives the liberty to evaluate the performance with coherence score calculations, whereas sklearn offers visual chart for LSA topic model.
- For LSA modelling we performed similar analyses to find the optimal number of topics by comparing the coherence scores and found out that the number of topics remained the same (three topics).
- Next, we implemented PLSA. Probabilistic Latent Semantic Analysis, or PLSA, is a technique for modeling data in a probabilistic framework. It has two components. The first latent variable model is a statistical model that underpins the probabilistic framework of pLSA. The observable variables are linked to the latent/hidden variables. Another aspect is matrix factorization which works in the same way as Latent Semantic Indexing. The goal of pLSA is to minimize the dimensionality of the sparse co-occurrence matrix by factorizing it. However, pLSA is often regarded as a more sound approach since it gives a probabilistic interpretation, whereas LSI just employs mathematical foundations to factorize (more precisely, LSI uses the singular value decomposition method).
- pLSA topic modelling technique isn't there in gensim library and with the sklearn implementation of pLSA there is no function to check the coherence score, so we can't compare the results of PSA with LDA model. LDA model with parameters alpha=1 and eta=1 works as pLSA. Setting alpha=1 and eta=1 doesn't exactly produce pLSA however it does make the Dirichlet prior dense which is more like pLSA.

- Sentiment scores are calculated using TextBlob, AFINN and Vader . We are taking the sentiment scores to of the posts and scaling it up by the weighted score . The whole purpose of this is to find if there is any correlation between the weighted sentiment scores and the change in price of Bitcoin
- Since the person(influencer) posting the comments/posts on Reddit plays a significant role in influencing the minds of others, we can use weighted sentimental scores to find the average sentiment of the day and then see if that metric has any correlation with trading volume as well as the price change.
- First, we can calculate the weighted average sentiment score of each post content and its comment content (for example giving weights based on comments upvotes and post upvotes and assigning one average sentiment score per post) The weighted average sentiment scores of the all the posts (based on upvotes).

6 RESULTS

Modeling using LDA method:

As discussed first we build LDA model with 10 topics and see the results and check the evaluation metrics.



```
Out[63]: [(0,
           '0.080*"year" + 0.034*"world" + 0.031*"investment" + 0.029*"great" + 0.026*"game" + 0.022*"portfolio" + 0.021*"idea" + 0.017
          *"early" + 0.017*"time" + 0.016*"gain"'),
          (1,
           '0.076*"problem" + 0.047*"information" + 0.042*"open" + 0.039*"solution" + 0.030*"development" + 0.027*"vault" + 0.024*"grou
          p" + 0.024*"recent" + 0.021*"individual" + 0.020*"creator"'),
          (2,
           '0.050*"post" + 0.047*"user" + 0.043*"asset" + 0.042*"point" + 0.037*"work" + 0.031*"currency" + 0.030*"seed" + 0.028*"large"
          + 0.026*"hour" + 0.025*"digital"'),
          (3,
           '0.028*"ethereum" + 0.027*"project" + 0.027*"blockchain" + 0.022*"transaction" + 0.021*"exchange" + 0.016*"network" + 0.015
          *"smart" + 0.014*"coinbase" + 0.014*"profit" + 0.013*"contract"'),
          (4,
           '0.079*"wallet" + 0.053*"future" + 0.041*"question" + 0.039*"high" + 0.032*"company" + 0.032*"fund" + 0.031*"worth" + 0.031
          *"dollar" + 0.028*"financial" + 0.028*"word"'),
          (5,
           '0.060*"price" + 0.038*"account" + 0.030*"amount" + 0.023*"different" + 0.020*"small" + 0.018*"support" + 0.018*"real" + 0.01
          6*"other" + 0.016*"safe" + 0.015*"place"'),
          (6,
           '0.072*"coin" + 0.064*"crypto" + 0.062*"people" + 0.054*"good" + 0.050*"money" + 0.048*"time" + 0.044*"market" + 0.027*"many"
          + 0.026*"thing" + 0.025*"value"'),
          (7,
           '0.077*"moon" + 0.053*"week" + 0.051*"much" + 0.051*"last" + 0.034*"life" + 0.030*"thank" + 0.027*"country" + 0.025*"well" +
          0.022*"scam" + 0.018*"plan"'),
          (8,
           '0.104*"crypto" + 0.047*"month" + 0.043*"cryptocurrency" + 0.035*"long" + 0.029*"next" + 0.023*"little" + 0.022*"stake" + 0.0
          20*"number" + 0.020*"investor" + 0.019*"loan"'),
          (9,
           '0.105*"bitcoin" + 0.032*"platform" + 0.027*"bank" + 0.023*"level" + 0.020*"right" + 0.019*"system" + 0.017*"cryptocurrencie"
          + 0.017*"reward" + 0.016*"mining" + 0.015*"hope"')]
```

Fig 26: The topics extracted which show different words and their weightages after modeling the data via the LDA method.

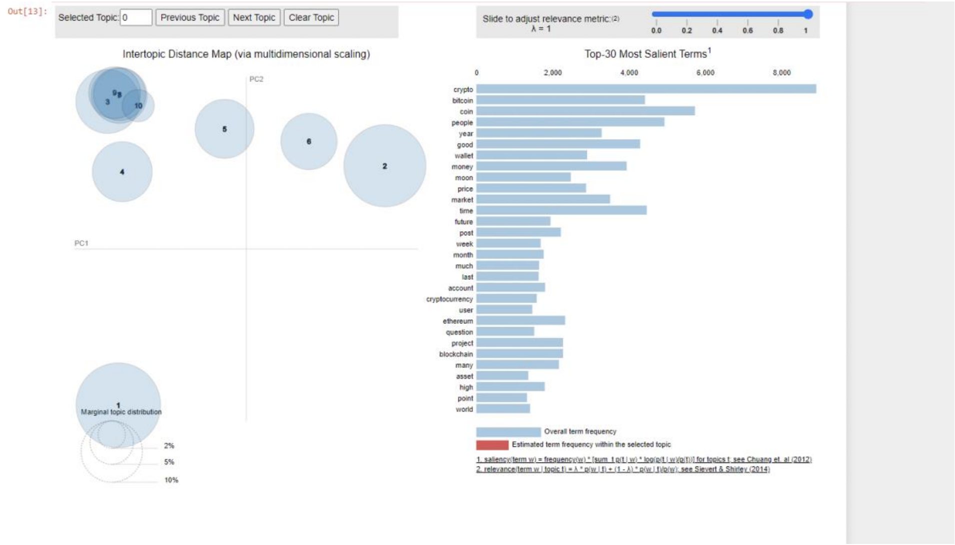
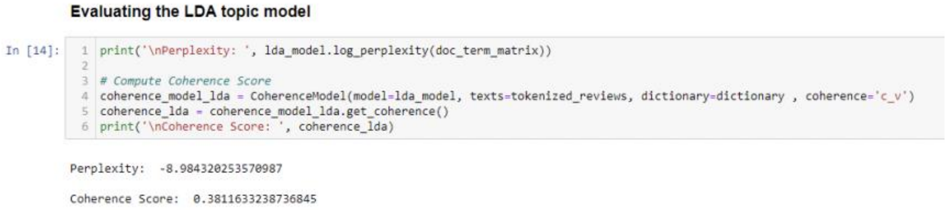

Out[13]:
Selected Topic: 0
Previous Topic
Next Topic
Clear Topic
Slide to adjust relevance metric:(2)
λ = 1
0.0
0.2
0.4
0.6
0.8
1
Intertopic Distance Map (via multidimensional scaling)
PC2
PC1
9
8
3
10
5
6
2
4
1
Marginal topic distribution
2%
5%
10%
Top-30 Most Salient Terms1
0
2,000
4,000
6,000
8,000
crypto
bitcoin
coin
people
year
good
wallet
money
moon
price
market
time
future
post
week
month
much
last
account
cryptocurrency
user
ethereum
question
project
blockchain
many
asset
high
point
world
Overall term frequency
Estimated term frequency within the selected topic
1. saliency(term w) = frequency(w) * [sum_t p(t | w) * log(p(t | w)/p(t))] for topics t; see Chuang et. al (2012)
2. relevance(term w | topic t) = λ * p(w | t) + (1 - λ) * p(w | t)/p(w); see Sievert & Shirley (2014)

Fig 27: The interactive maps created using the pyLDAvis for the topics extracted as shown above. As we can see that the 10 topics are overlapping and hence the model is not doing a great job with 10 topics as there is low separability between topics.



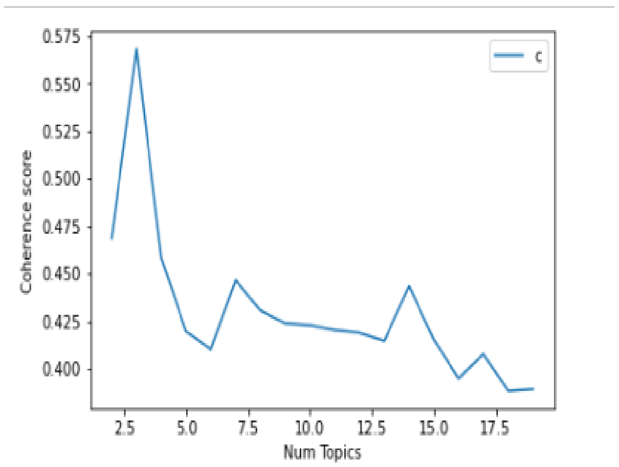
Evaluating the LDA topic model

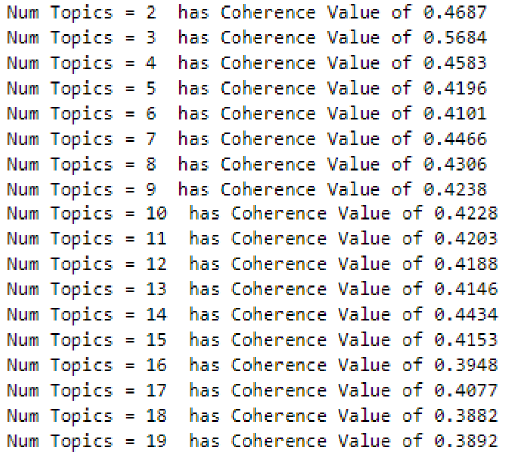
In [14]:

```
print('\nPerplexity: ', lda_model.log_perplexity(doc_term_matrix))

# Compute Coherence Score
coherence_model_lda = CoherenceModel(model=lda_model, texts=tokenized_reviews, dictionary=dictionary , coherence='c_v')
coherence_lda = coherence_model_lda.get_coherence()
print('\nCoherence Score: ', coherence_lda)
```

```
Perplexity:  -8.984320253570987

Coherence Score:  0.3811633238736845
```

Fig 28: The perplexity and the coherence values for the above extracted model. Lower perplexity and the higher coherence values show the strength of the model. The coherence score is 0.381160 and the perplexity is -8.98432.

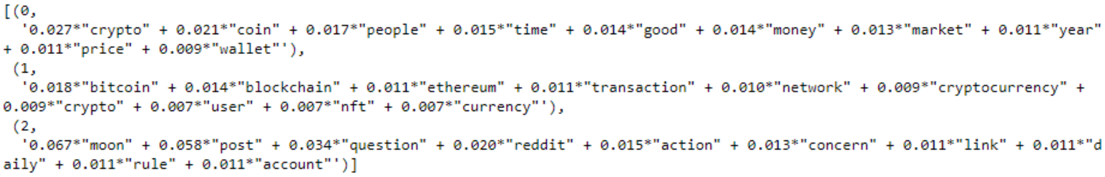




```
Num Topics = 2  has Coherence Value of 0.4687
Num Topics = 3  has Coherence Value of 0.5684
Num Topics = 4  has Coherence Value of 0.4583
Num Topics = 5  has Coherence Value of 0.4196
Num Topics = 6  has Coherence Value of 0.4101
Num Topics = 7  has Coherence Value of 0.4466
Num Topics = 8  has Coherence Value of 0.4306
Num Topics = 9  has Coherence Value of 0.4238
Num Topics = 10  has Coherence Value of 0.4228
Num Topics = 11  has Coherence Value of 0.4203
Num Topics = 12  has Coherence Value of 0.4188
Num Topics = 13  has Coherence Value of 0.4146
Num Topics = 14  has Coherence Value of 0.4434
Num Topics = 15  has Coherence Value of 0.4153
Num Topics = 16  has Coherence Value of 0.3948
Num Topics = 17  has Coherence Value of 0.4077
Num Topics = 18  has Coherence Value of 0.3882
Num Topics = 19  has Coherence Value of 0.3892
```

Fig 29: The line chart for the coherence values generated for different number of topic values. The 3 number of topics has generated the highest coherence value which is 0.5684.

Hence, we rebuilt our LDA model with 3 topic heads.



```
[(0,
  '0.027*"crypto" + 0.021*"coin" + 0.017*"people" + 0.015*"time" + 0.014*"good" + 0.014*"money" + 0.013*"market" + 0.011*"year"
+ 0.011*"price" + 0.009*"wallet"'),
 (1,
  '0.018*"bitcoin" + 0.014*"blockchain" + 0.011*"ethereum" + 0.011*"transaction" + 0.010*"network" + 0.009*"cryptocurrency" +
0.009*"crypto" + 0.007*"user" + 0.007*"nft" + 0.007*"currency"'),
 (2,
  '0.067*"moon" + 0.058*"post" + 0.034*"question" + 0.020*"reddit" + 0.015*"action" + 0.013*"concern" + 0.011*"link" + 0.011*"d
aily" + 0.011*"rule" + 0.011*"account"')]
```

Fig 30: The 3 topics that are generated when the model is rebuilt again with the optimum no of topics.

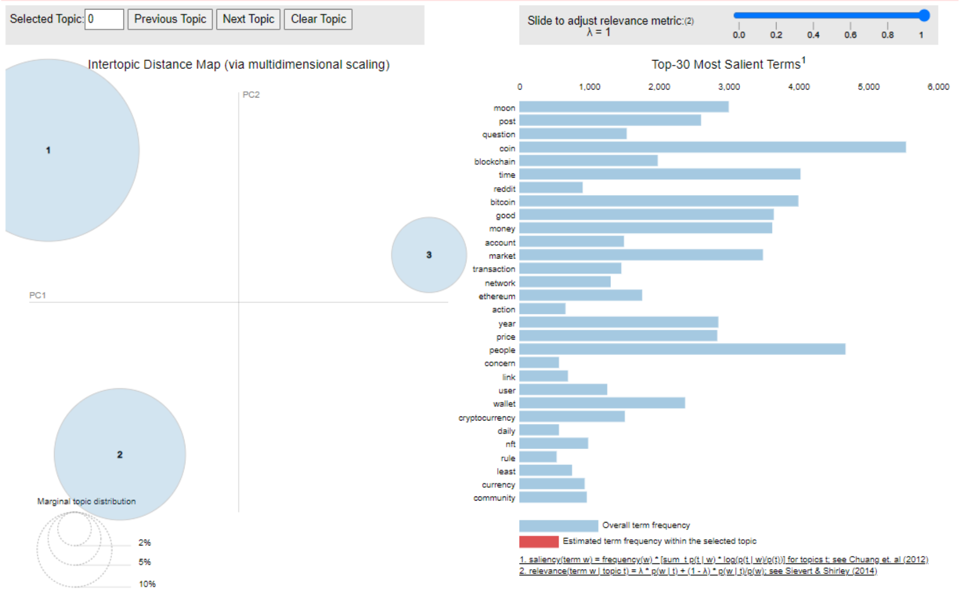
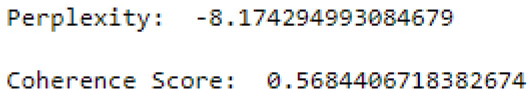

Selected Topic: 0
Previous Topic
Next Topic
Clear Topic
Slide to adjust relevance metric:(2)
λ = 1
0.0
0.2
0.4
0.6
0.8
1
Intertopic Distance Map (via multidimensional scaling)
PC2
PC1
1
2
3
Marginal topic distribution
2%
5%
10%
Top-30 Most Salient Terms1
0
1,000
2,000
3,000
4,000
5,000
6,000
moon
post
question
coin
blockchain
time
reddit
bitcoin
good
money
account
market
transaction
network
ethereum
action
year
price
people
concern
link
user
wallet
cryptocurrency
daily
nft
rule
least
currency
community
Overall term frequency
Estimated term frequency within the selected topic
1. saliency(term w) = frequency(w) * [sum_t p(t | w) * log(p(t | w)/p(t))] for topics t; see Chuang et. al (2012)
2. relevance(term w | topic t) = λ * p(w | t) + (1 - λ) * p(w | t)/p(w); see Sievert & Shirley (2014)

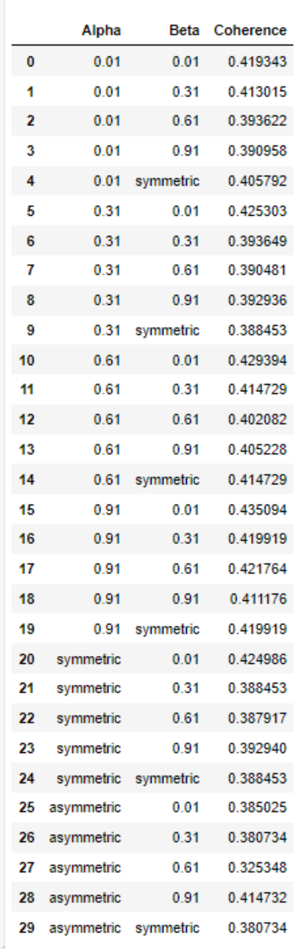
Fig 31: The interactive visualizations of the model built with the 3 topics using pyLDAvis.



```
Perplexity:  -8.174294993084679

Coherence Score:  0.5684406718382674
```

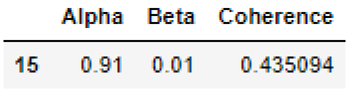
Fig 32: The model built for the 3 topics has the perplexity value of −8.17 and the coherence score of 0.56 which is better than the first model.

Now using grid search we do hyper parameter tuning to find the best values for alpha and beta for our LDA model.



	Alpha	Beta	Coherence
0	0.01	0.01	0.419343
1	0.01	0.31	0.413015
2	0.01	0.61	0.393622
3	0.01	0.91	0.390958
4	0.01	symmetric	0.405792
5	0.31	0.01	0.425303
6	0.31	0.31	0.393649
7	0.31	0.61	0.390481
8	0.31	0.91	0.392936
9	0.31	symmetric	0.388453
10	0.61	0.01	0.429394
11	0.61	0.31	0.414729
12	0.61	0.61	0.402082
13	0.61	0.91	0.405228
14	0.61	symmetric	0.414729
15	0.91	0.01	0.435094
16	0.91	0.31	0.419919
17	0.91	0.61	0.421764
18	0.91	0.91	0.411176
19	0.91	symmetric	0.419919
20	symmetric	0.01	0.424986
21	symmetric	0.31	0.388453
22	symmetric	0.61	0.387917
23	symmetric	0.91	0.392940
24	symmetric	symmetric	0.388453
25	asymmetric	0.01	0.385025
26	asymmetric	0.31	0.380734
27	asymmetric	0.61	0.325348
28	asymmetric	0.91	0.414732
29	asymmetric	symmetric	0.380734

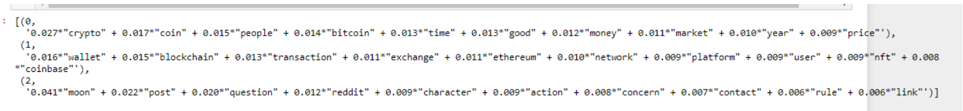
Fig 33: The table showing all the alpha, beta and their corresponding coherence values after the hyper parameter tunning is performed.



	Alpha	Beta	Coherence
15	0.91	0.01	0.435094

Fig 34: The alpha value of 0.91 with the beta value of 0.01 producing a coherence value of 0.435 are the best values.

Though the coherence score dropped slightly, we will observe below that the perplexity decreased by a great extent too, hence making the model perform better with these values.



```
: [(0,
   '0.027*"crypto" + 0.017*"coin" + 0.015*"people" + 0.014*"bitcoin" + 0.013*"time" + 0.013*"good" + 0.012*"money" + 0.011*"market" + 0.010*"year" + 0.009*"price"'),
  (1,
   '0.016*"wallet" + 0.015*"blockchain" + 0.013*"transaction" + 0.011*"exchange" + 0.011*"ethereum" + 0.010*"network" + 0.009*"platform" + 0.009*"user" + 0.009*"nft" + 0.008
 *"coinbase"'),
  (2,
   '0.041*"moon" + 0.022*"post" + 0.020*"question" + 0.012*"reddit" + 0.009*"character" + 0.009*"action" + 0.008*"concern" + 0.007*"contact" + 0.006*"rule" + 0.006*"link"')]
```

Fig 35: The topics generated after the model is rebuilt again with the ideal alpha and beta values.

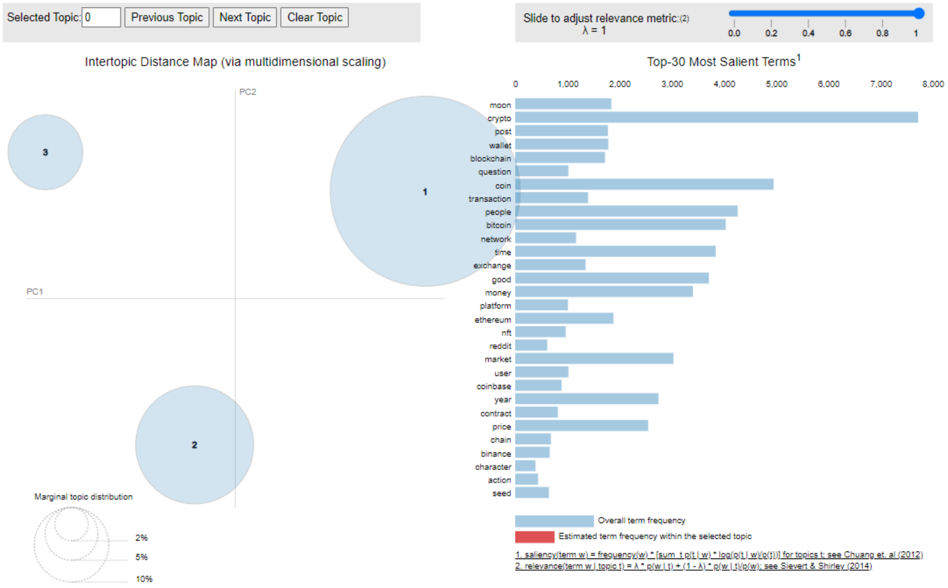
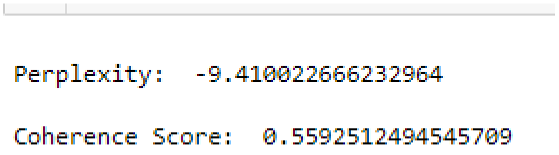

Selected Topic: 0
Previous Topic
Next Topic
Clear Topic
Slide to adjust relevance metric:(2)
λ = 1
0.0
0.2
0.4
0.6
0.8
1
Intertopic Distance Map (via multidimensional scaling)
PC2
PC1
1
2
3
Marginal topic distribution
2%
5%
10%
Top-30 Most Salient Terms(1)
0
1,000
2,000
3,000
4,000
5,000
6,000
7,000
8,000
moon
crypto
post
wallet
blockchain
question
coin
transaction
people
bitcoin
network
time
exchange
good
money
platform
ethereum
nft
reddit
market
user
coinbase
year
contract
price
chain
binance
character
action
seed
Overall term frequency
Estimated term frequency within the selected topic
1. saliency(term w) = frequency(w) * [sum_t p(t | w) * log(p(t | w)/p(t))] for topics t; see Chuang et. al (2012)
2. relevance(term w | topic t) = λ * p(w | t) + (1 - λ) * p(w | t)/p(w); see Sievert & Shirley (2014)

Fig 36: The interactive visualizations generated for the above model with the best alpha and beta values.

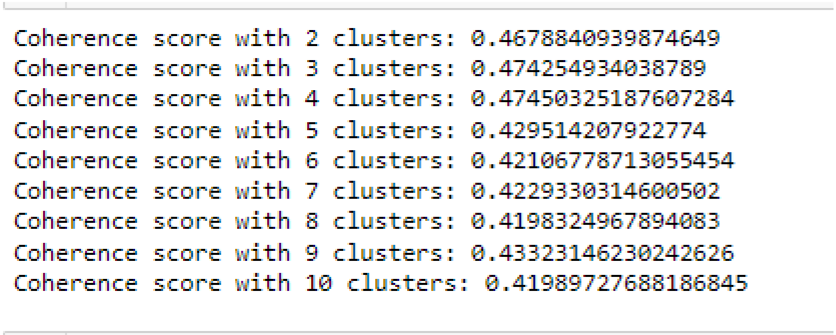


```
Perplexity:  -9.410022666232964

Coherence Score:  0.5592512494545709
```

Fig 37: The Perplexity score of –9.4 and coherence values of 0.559 and generated for the above value which is the highest among the 3 models generated using LDA.

Finally, the LDA model generated using the alpha value of 0.91, beta value of 0.01 for the 3 topics put together gave the highest coherence and the lowest perplexity values.

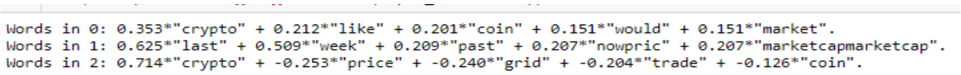
Modeling using the LSA:



```
Coherence score with 2 clusters: 0.4678840939874649
Coherence score with 3 clusters: 0.474254934038789
Coherence score with 4 clusters: 0.47450325187607284
Coherence score with 5 clusters: 0.429514207922774
Coherence score with 6 clusters: 0.42106778713055454
Coherence score with 7 clusters: 0.4229330314600502
Coherence score with 8 clusters: 0.4198324967894083
Coherence score with 9 clusters: 0.43323146230242626
Coherence score with 10 clusters: 0.41989727688186845
```

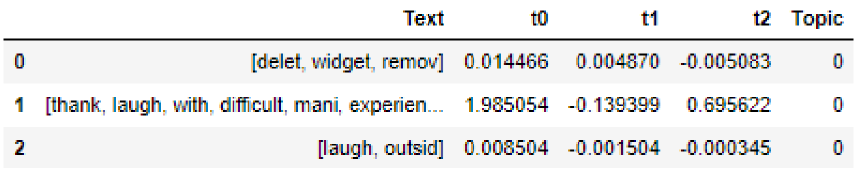
Fig 38: The coherence scores for the n number of clusters generated using the LSA model.

Here since 3 clusters gave almost similar score to 4 clusters, we chose 3 clusters as we could compare the results with other modelling techniques at later stage.



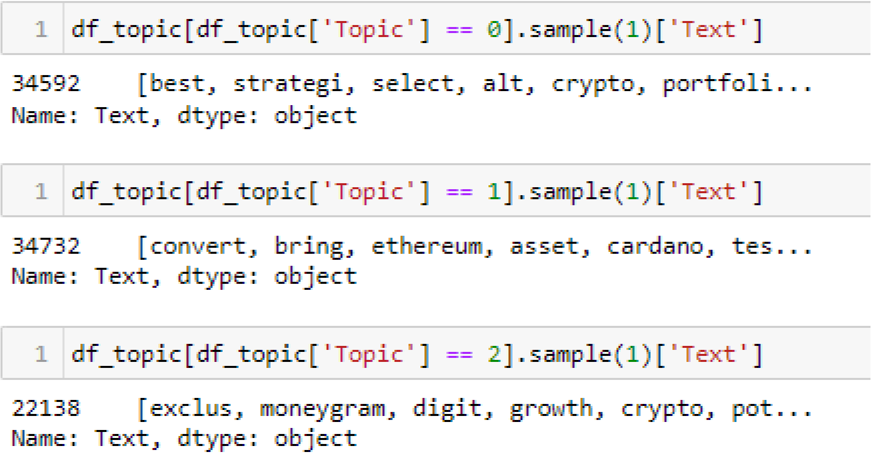
```
Words in 0: 0.353*"crypto" + 0.212*"like" + 0.201*"coin" + 0.151*"would" + 0.151*"market".
Words in 1: 0.625*"last" + 0.509*"week" + 0.209*"past" + 0.207*"nowpric" + 0.207*"marketcapmarketcap".
Words in 2: 0.714*"crypto" + -0.253*"price" + -0.240*"grid" + -0.204*"trade" + -0.126*"coin".
```

Fig 39: The topics generated when the model is built using the LSA algorithm on three topics.



	Text	t0	t1	t2	Topic
0	[delet, widget, remov]	0.014466	0.004870	-0.005083	0
1	[thank, laugh, with, difficult, mani, experien...	1.985054	-0.139399	0.695622	0
2	[laugh, outsid]	0.008504	-0.001504	-0.000345	0

Fig 40: The Data frame showing scores assigned for three topics for each review. For the above document, we can see that the topic has the highest weightage, hence that discussion belongs to topic



```
df_topic[df_topic['Topic'] == 0].sample(1)['Text']
```

```
34592    [best, strategi, select, alt, crypto, portfoli...
Name: Text, dtype: object
```

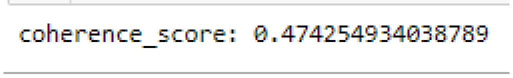
```
df_topic[df_topic['Topic'] == 1].sample(1)['Text']
```

```
34732    [convert, bring, ethereum, asset, cardano, tes...
Name: Text, dtype: object
```

```
df_topic[df_topic['Topic'] == 2].sample(1)['Text']
```

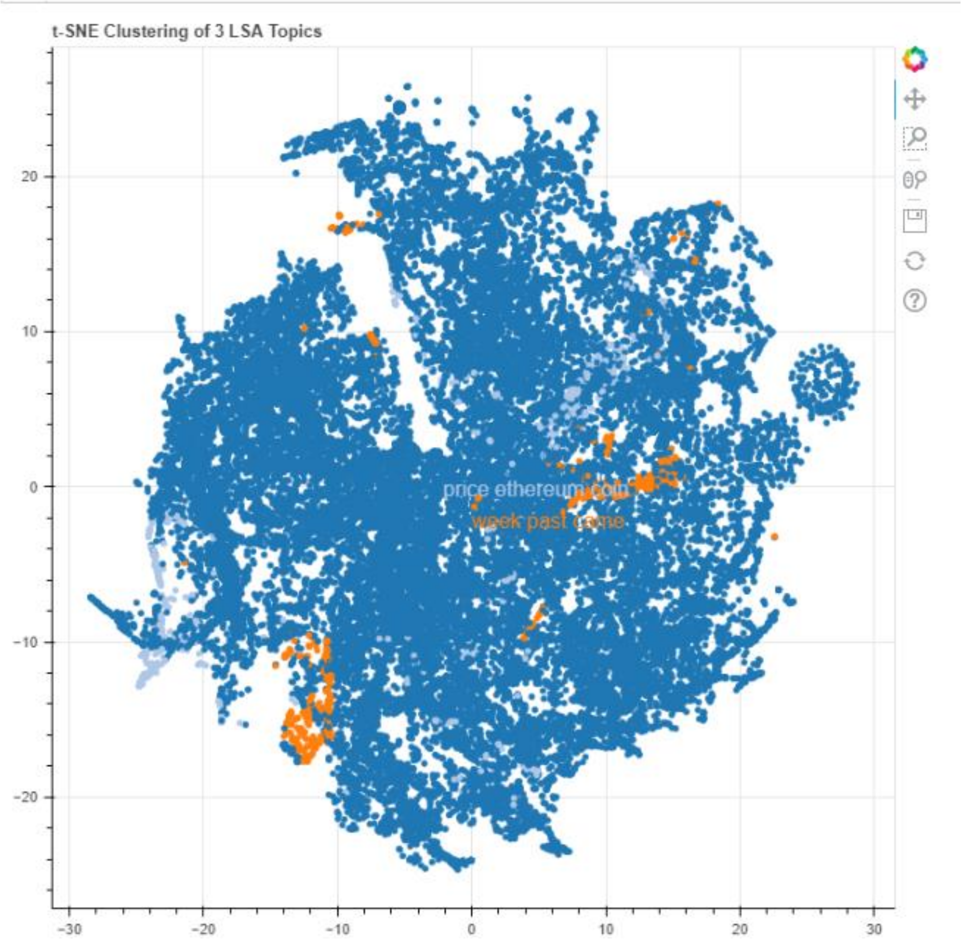
```
22138    [exclus, moneygram, digit, growth, crypto, pot...
Name: Text, dtype: object
```

Fig 41: This represents a sample post from each of the three topics after each post is categorized into one of the three topics.



```
coherence_score: 0.474254934038789
```

Fig 42: The coherence value of 0.47 is generated for the above model built using LSA.



‘

Fig 43: Clustering of the 3 topics of LSA using sklearn. This shows, one topic dominates all the other and there is no separability between topics or low variability within topics.

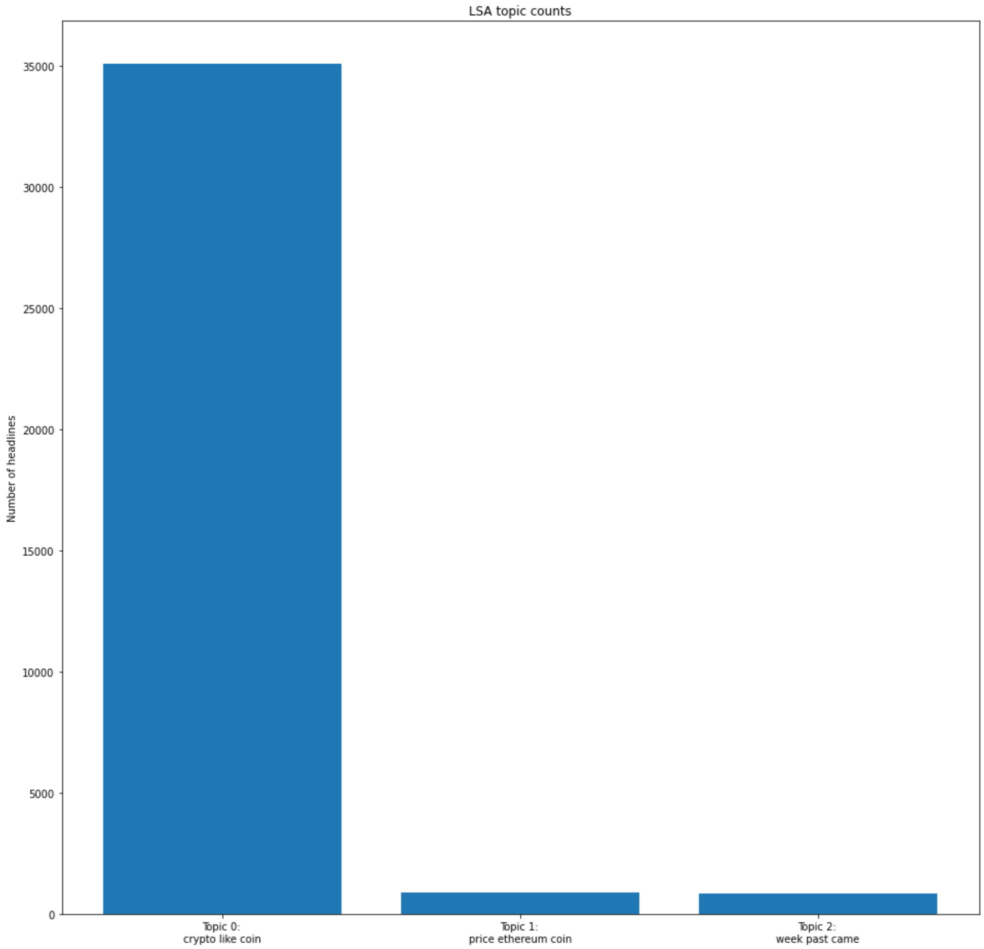
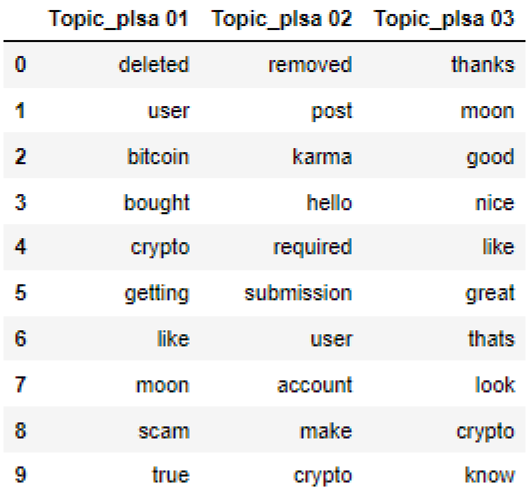


Fig 44: The bar chart showing how one topic is dominated by the other topics.

As LSA using sklearn does not offer liberty to evaluate the performance either though coherence score or perplexity, only way we can conclude that this LSA model isn't performing well is by checking the separation between topics. As they overall fail to accomplish the principal reason of separability of topic modelling, we can say that LDA produced better results.

Modeling using PLSA:



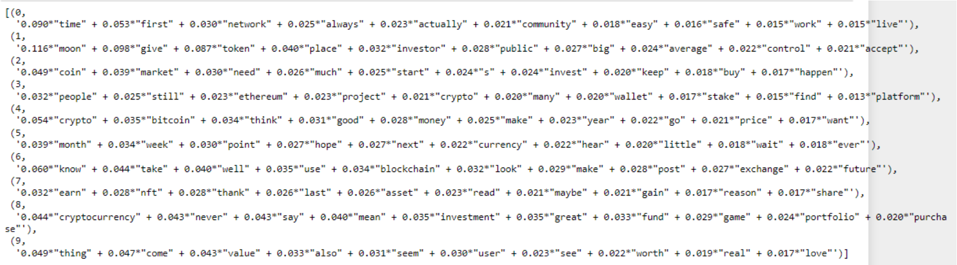
	Topic_plsa 01	Topic_plsa 02	Topic_plsa 03
0	deleted	removed	thanks
1	user	post	moon
2	bitcoin	karma	good
3	bought	hello	nice
4	crypto	required	like
5	getting	submission	great
6	like	user	thats
7	moon	account	look
8	scam	make	crypto
9	true	crypto	know

Fig 45: The most frequent words used in each of the three topics.

As PLSA implemented with sklearn does not offer liberty to evaluate the performance either though coherence score or perplexity, we cannot compare it with any other topic modelling results. As per the scientific reviews, PLSA performs quite similar to that of LDA given the LDA model is trained with parameters alpha=1 and eta=1. As we had seen above alpha=1 and eta=1 gave lesser coherence score than the alpha 0.9 and eta 0.01, we can conclude again that LDA performs better for our dataset than PLSA.

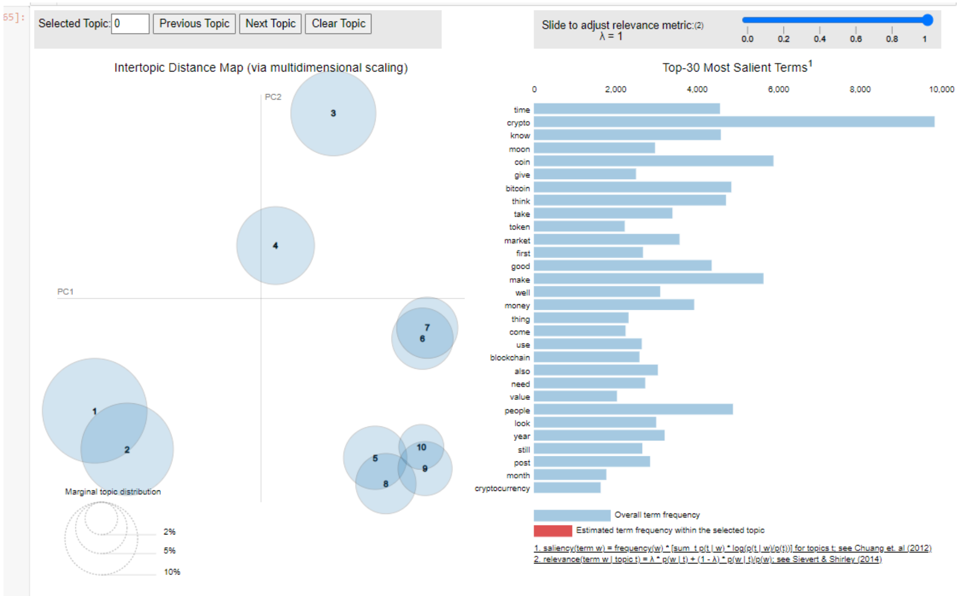
Using Bigrams and Trigrams:

We try using bigrams and trigrams to see if we have better results with our topic modelling techniques



```
[(0,
  '0.090*"time" + 0.053*"first" + 0.030*"network" + 0.025*"always" + 0.023*"actually" + 0.021*"community" + 0.018*"easy" + 0.016*"safe" + 0.015*"work" + 0.015*"live"'),
 (1,
  '0.116*"moon" + 0.098*"give" + 0.087*"token" + 0.040*"place" + 0.032*"investor" + 0.028*"public" + 0.027*"big" + 0.024*"average" + 0.022*"control" + 0.021*"accept"'),
 (2,
  '0.049*"coin" + 0.039*"market" + 0.030*"need" + 0.026*"much" + 0.025*"start" + 0.024*"s" + 0.024*"invest" + 0.020*"keep" + 0.018*"buy" + 0.017*"happen"'),
 (3,
  '0.032*"people" + 0.025*"still" + 0.023*"ethereum" + 0.023*"project" + 0.021*"crypto" + 0.020*"many" + 0.020*"wallet" + 0.017*"stake" + 0.015*"find" + 0.013*"platform"'),
 (4,
  '0.054*"crypto" + 0.035*"bitcoin" + 0.034*"think" + 0.031*"good" + 0.028*"money" + 0.025*"make" + 0.023*"year" + 0.022*"go" + 0.021*"price" + 0.017*"want"'),
 (5,
  '0.039*"month" + 0.034*"week" + 0.030*"point" + 0.027*"hope" + 0.027*"next" + 0.022*"currency" + 0.022*"hear" + 0.020*"little" + 0.018*"wait" + 0.018*"ever"'),
 (6,
  '0.060*"know" + 0.044*"take" + 0.040*"well" + 0.035*"use" + 0.034*"blockchain" + 0.032*"look" + 0.029*"make" + 0.028*"post" + 0.027*"exchange" + 0.022*"future"'),
 (7,
  '0.032*"earn" + 0.028*"nft" + 0.028*"thank" + 0.026*"last" + 0.026*"asset" + 0.023*"read" + 0.021*"maybe" + 0.021*"gain" + 0.017*"reason" + 0.017*"share"'),
 (8,
  '0.044*"cryptocurrency" + 0.043*"never" + 0.043*"say" + 0.040*"mean" + 0.035*"investment" + 0.035*"great" + 0.033*"fund" + 0.029*"game" + 0.024*"portfolio" + 0.020*"purcha
se"'),
 (9,
  '0.049*"thing" + 0.047*"come" + 0.043*"value" + 0.033*"also" + 0.031*"seem" + 0.030*"user" + 0.023*"see" + 0.022*"worth" + 0.019*"real" + 0.017*"love"')]
```

Fig 46: The topics generated after using bigrams and trigrams using the LDA algorithm.



1. saliency(term w) = frequency(w) * [sum_t p(t | w) * log(p(t | w)/p(t))] for topics t; see Chuang et. al (2012)
2. relevance(term w | topic t) = λ * p(w | t) + (1 - λ) * p(w | t)/p(w); see Sievert & Shirley (2014)

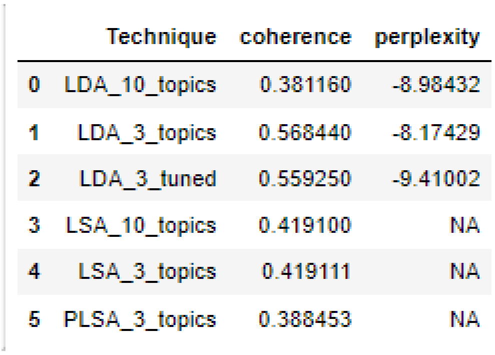
Fig 47: The interactive visualization for LDA model with bigram, trigram inputs.

```
Perplexity:  -9.366117119048669
```

Fig 48: The perplexity score of −9.36 is generated after using the bigrams and trigrams with LDA.

As we can see that there is no improvement as the Bigrams and trigrams as the perplexity is higher than LDA without bigrams and trigrams. Since bigrams and trigram were not even listed in the most frequent words in different topics, we can say that they were of low significance or low frequency in determining the frequent words in the topic model.

Comparison between models (PLSA vs LDA vs LSA):



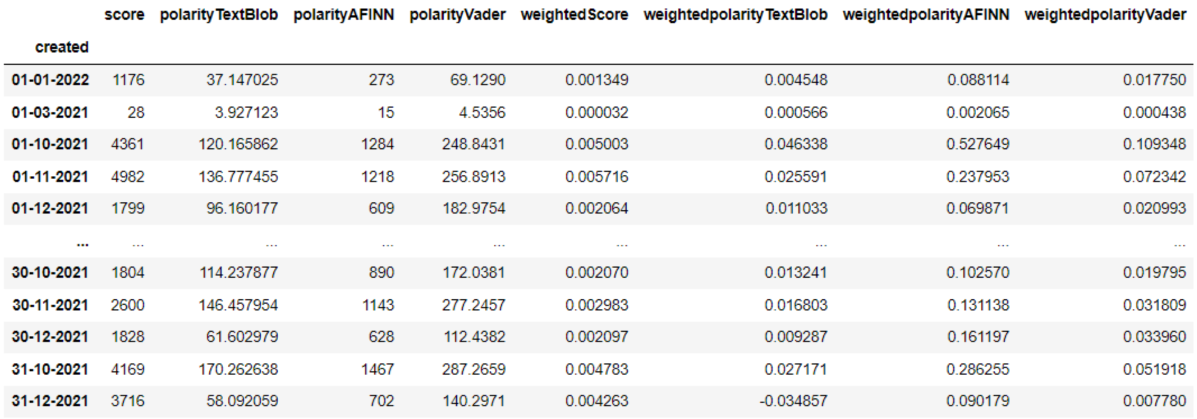
	Technique	coherence	perplexity
0	LDA_10_topics	0.381160	-8.98432
1	LDA_3_topics	0.568440	-8.17429
2	LDA_3_tuned	0.559250	-9.41002
3	LSA_10_topics	0.419100	NA
4	LSA_3_topics	0.419111	NA
5	PLSA_3_topics	0.388453	NA

Fig 49: The comparison of coherence and perplexity of different models

As we had seen above alpha=1 and eta=1 gave lesser coherence score than the alpha 0.9 and eta 0.01, we can conclude again that LDA performs better for our dataset than PLSA. Coherence score and perplexity is the best for LDA hyper tuned models with 3 topics. As proved by the separability in visualization, we can see that coherence score and perplexity is the best for LDA hyper tuned models with 3 topics.

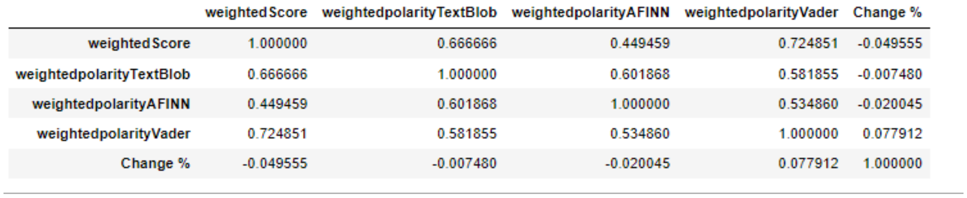
Sentiment Analysis:

The sentiment scores of each text are first calculated using the packages TextBlob, AFINN and Vader. The values obtained are then scaled using the weighted score to get a new set of values. These values are then aggregated day wise. These aggregated values are checked for correlation against the change in price of bitcoin. The correlation is very weak between the values mentioned above.



	score	polarityTextBlob	polarityAFINN	polarityVader	weightedScore	weightedpolarityTextBlob	weightedpolarityAFINN	weightedpolarityVader
created								
01-01-2022	1176	37.147025	273	69.1290	0.001349	0.004548	0.088114	0.017750
01-03-2021	28	3.927123	15	4.5356	0.000032	0.000566	0.002065	0.000438
01-10-2021	4361	120.165862	1284	248.8431	0.005003	0.046338	0.527649	0.109348
01-11-2021	4982	136.777455	1218	256.8913	0.005716	0.025591	0.237953	0.072342
01-12-2021	1799	96.160177	609	182.9754	0.002064	0.011033	0.069871	0.020993
...	...	...	...	...	...	...	...	...
30-10-2021	1804	114.237877	890	172.0381	0.002070	0.013241	0.102570	0.019795
30-11-2021	2600	146.457954	1143	277.2457	0.002983	0.016803	0.131138	0.031809
30-12-2021	1828	61.602979	628	112.4382	0.002097	0.009287	0.161197	0.033960
31-10-2021	4169	170.262638	1467	287.2659	0.004783	0.027171	0.286255	0.051918
31-12-2021	3716	58.092059	702	140.2971	0.004263	-0.034857	0.090179	0.007780

Fig 50: Aggregated Sentiment scores, day wise, using TExtBlob, AFINN and Vader



	weightedScore	weightedpolarityTextBlob	weightedpolarityAFINN	weightedpolarityVader	Change %
weightedScore	1.000000	0.666666	0.449459	0.724851	-0.049555
weightedpolarityTextBlob	0.666666	1.000000	0.601868	0.581855	-0.007480
weightedpolarityAFINN	0.449459	0.601868	1.000000	0.534860	-0.020045
weightedpolarityVader	0.724851	0.581855	0.534860	1.000000	0.077912
Change %	-0.049555	-0.007480	-0.020045	0.077912	1.000000

Fig 51: Correlation between change in price of Bitcoin and weighted sentiment scores

7 DISCUSSION OF RESULTS

People would be able to predict, with a reasonable estimate, the changes in price of Crypto currencies

- The limitations of our approach are
  - Social media data alone is not a very good indicator for Stock Price, especially Crypto Currencies
  - Most records were quite short and had to be combined with other text to create larger more meaningful text
  - Lack of meaningful correlation between sentiment and price changes
- What would be the future work?
  - Adding more data for modeling (extract more months of data)
  - Add more features and use feature engineering to integrate other data fields.

- What are the limits of your regressors or classifiers? What are your suggestions to improve them in the future?
- The major Limitations of LDA are
  - Inability to scale
  - Unsuitable for short user generated text
- The major Limitations of LSA are
  - Latent topic dimension depends upon the rank of the matrix so we can't extend that limit.
  - LSA decomposed matrix is a highly dense matrix, so it is difficult to index individual dimensions.
  - LSA is unable to capture the multiple meanings of words.
- The major Limitations of pLSA are
  - It is not fully generative and fails to assign a probability to unseen documents
  - It is prone to overfitting

8 REFERENCES

[1] Ghoorchian, K., Sahlgren, M. GDTM: Graph-based Dynamic Topic Models. Prog Artif Intell 9, 195–207 (2020). https://doi.org/10.1007/s13748-020-00206-2

[2] Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent Dirichlet allocation. J.Mach. Learn. Res. 3, 993–1022 (2003)

[3] Yan, X., Guo, J., Lan, Y., Cheng, X.: A Biterm topic model for short texts. In:Proceedings of the 22nd International Conference on World Wide Web, pp. 1445–1456, (2013)

[4] Akhtar, Nadeem and Beg, M.M. Sufyan. 'User Graph Topic Model'. 1 Jan. 2019: 2229 – 2240.

[5] G. Pedrosa, et al., Topic modeling for short texts with cooccurrence frequency-based expansion, Intelligent Systems (BRACIS) 2016 (2016)

[6] Golino H, Christensen AP, Moulder R, Kim S, Boker SM. Modeling Latent Topics in Social Media using Dynamic Exploratory Graph Analysis: The Case of the Right-wing and Left-wing Trolls in the 2016 US Elections. Psychometrika. 2021 Nov 10. doi: 10.1007/s11336-021-09820-y. Epub ahead of print. PMID: 34757581.

[7] Ghanem, B., Buscaldi, D., & Rosso, P. (2019). TexTrolls: Identifying russian trolls on twitter from a textual perspective. arXiv, (1910.01340). Retrieved from arXiv:1910.01340

[8] Zannettou, S., Caulfield, T., De Cristofaro, E., Sirivianos, M., Stringhini, G., & Blackburn, J. (2019). Disinformation warfare: Understanding state-sponsored trolls on twitter and their influence on the web. In Companion proceedings of the 2019 world wide web conference (pp. 218–226).

[9] R. Churchill and L. Singh, The Evolution of Topic Modeling, ACM Computing Surveys, Jan. 2022. https://dl.acm.org/doi/pdf/10.1145/3507900

[10] Asmussen, C.B., Møller, C. Smart literature review: a practical topic modelling approach to exploratory literature review, *J Big Data* **6,** 2019. https://doi.org/10.1186/s40537-019-0255-7

[11] Jason Thies, Lukas Stappen, Gerhard Hagerer, Bjorn W. Schuller, Georg Groh, GraphTMT: Unsupervised Graph-based Topic Modeling from Video Transcripts, arXiv:2105.01466v4 [cs.CL], 28 Oct 2021. https://arxiv.org/pdf/2105.01466v4.pdf

[12] Maarten Grootendorst, BERTopic: Neural topic modeling with a class-based TF-IDF procedure, arXiv:2203.05794v1 [cs.CL], 11 Mar 2022. https://arxiv.org/pdf/2203.05794v1.pdf

[13] Chenliang Li, Haoran Wang , Zhiqian Zhang , Aixin Sun, Zongyang Ma
SIGIR '16: Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval, July 2016 Pages 165–174 https://doi.org/10.1145/2911451.2911499

[14] Yuan Zuo , Junjie Wu ,Hui Zhang, Hao Lin, Fei Wang, Ke Xu, Hui Xiong
KDD '16: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, August 2016 Pages 2105–2114
https://doi.org/10.1145/2939672.2939880

[15] Kuldeep Singh, Harish Kumar Shakya, Bhaskar Biswas,Clustering of people in social network based on textual similarity,Perspectives in Science,Volume 8,2016,Pages 570-573,ISSN 2213-0209, https://doi.org/10.1016/j.pisc.2016.06.023

[16] Practical Text Analytics - Maximizing the Value of Text Data , Murugan Anandarajan • Chelsey Hill , Thomas Nolan

[17] Matt Podolak: How to Scrape Large Amounts of Reddit Data, 14 Feb 2021 https://medium.com/swlh/how-to-scrape-large-amounts-of-reddit-data-using-pushshift-1d33bde9286

[18] Gabriel preda: Reddit CryptoCurrency, Oct 2021, https://www.kaggle.com/datasets/gpreda/reddit-cryptocurrency

[20] Joyce Xu: Topic Modeling with LSA, PLSA, LDA & lda2Vec, 25 May 2015

[21] https://medium.com/nanonets/topic-modeling-with-lsa-psla-lda-and-lda2vec-555ff65b0b05

[22] https://www.datacamp.com/community/tutorials/discovering-hidden-topics-python

[23] LDA Topic Modelling Explained with implementation using gensim in Python -#NLPRoc tutorial

[24] https://github.com/rsreetech/LDATopicModelling/blob/main/LDADemo.ipynb

[25] https://towardsdatascience.com/evaluate-topic-model-in-python-latent-dirichlet-allocation-lda-7d57484bb5d0

[26] http://qpleple.com/perplexity-to-evaluate-topic-models/

[27] https://www.baeldung.com/cs/topic-modeling-coherence-score#:~:text=We%20can%20use%20the%20coherence,words%20are%20to%20each%20other.

[28] https://towardsdatascience.com/understanding-topic-coherence-measures-4aa41339634c

[29] https://www.machinelearningplus.com/nlp/topic-modeling-python-sklearn-examples/

[30] StatQuest: Linear Discriminant Analysis (LDA) clearly explained.

[31] Text Analytics - Latent Semantic Analysis

[32] LSA (Latent Semantic Analysis)

[33] https://lib.asu.edu/news/twitter-stock-price-prediction (https://github.com/UnitForDataScience/Stonks_3.0/blob/master/main.ipynb)

[34] https://towardsdatascience.com/sentiment-analysis-for-stock-price-prediction-in-python-bed40c65d178

[35] https://www.kaggle.com/code/ankithb21/wsb-wordclouds-sentiment-analysis-easy

[36] https://link.springer.com/chapter/10.1007/978-981-16-9669-5_7#Sec7

[37] https://www.semanticscholar.org/paper/NLP-for-Stock-Market-Prediction-with-Reddit-Data-Xu/b222e27f918d010beccc8397ecb4c5044e5277b9

[38] https://github.com/yedivanseven/PLSA

[39] https://www.geeksforgeeks.org/

[40] https://www.datacamp.com/community/tutorials/discovering-hidden-topics-python

[41] https://lazarinastoy.com/topic-modelling-lda/

[42]https://www.analyticsvidhya.com/blog/2021/06/part-17-step-by-step-guide-to-master-nlp-topic-modelling-using-plsa/

[43] https://ieeexplore.ieee.org/document/7381513

[44] https://www.investing.com/crypto/bitcoin/historical-data