PREPARING ANNOTATED DATA ON COVID -19 BY EMPLOYING NAÏVE BAYES

Dipankar Das, Akash Ghosh, AdityaR Rayala, Dibyajyoti Dhar,
Vidit Sarkar,Avishek Garain, Sourav Kumar

Department of Computer Science & Engineering, Jadavpur University, India

ABSTRACT

The on-going pandemic has opened the pandora's box of the plethora of hidden problems which the society has been hiding for years. But the positive side to the present scenario is the opening up of opportunities to solve these problems on the global stage. One such area which was being flooded with all kinds of different emotions, and reaction from the people all over the world, is twitter, which is a micro blogging platform. Coronavirus related hash tags have been trending all over for many days unlikeany other event in the past. Our experiment mainly deals with the collection, tagging and classification of these tweets based on the different keywords that they may belong to, using the Naive Bayes algorithm atthe core.

KEYWORDS

Covid-19, Naïve Bayes, Clustering.

1. INTRODUCTION

COVID-19 or SARS-CoV-2 is a severe acute respiratory syndrome (SARS) that emerged from Wuhan, China in December, 2019. It has since resulted in a global pandemic situation and has resulted in over 1.5 million deaths worldwide along with mass disruption of lives for all people around the world. With a pandemic of this scale, social media has had lots to say about the virus ever since its inception. Twitter has seen an unprecedented rise in the number of tweets ever since the pandemic started [1-3] and people from various walks of life have commented on various aspects concerning the virus. These include advice from doctors regarding how to cope with the deadly disease, how people are coping with the new norm of work-from-home, tweets regarding the lockdown and other such. With such a large number of tweets, users are often left searching for what to read as all tweets might not be of interest to them. In this context, segregating the similar tweets together so that the user can read about a particular kind only is essential.

Therefore, the objective of the present work is to classify covid-19 related Twitter data into a specified number of classes and deliver an annotated dataset for the upcoming research communities. In order to accomplish our research goals, we have implemented several varieties of Multinomial Naïve Bayes algorithm from scratch using Python to classify the data into some pre-defined classes on a dataset. We have considered three different strategies to prepare the data as well as to implement the Naïve Bayes models.

David C. Wyld et al. (Eds): MLIOB, SIPO, NET, DNLP, SOEA, AISCA - 2021
Pp.131-147, 2021. CS & IT - CSCP 2021 DOI: 10.5121/csit.2021.111211

In the first case, we have collected 10,000 unique tweets from the whole dataset[1] that ranges from December 2019 till May 2020. The keywords that have been used to crawl the tweets are 'corona', 'covid', 'sarscov2', 'covid19', 'coronavirus'. For this purpose we used Tweepy and Twitter API endpoint.

In the second attempt, we also collected 10,000 tweets. On the initial unlabeled Twitter data, spectral clustering is used to automatically generate class label. This labeled data is used as training data for the classifier and accuracy of the classifier has been calculated.

In the third case also, we have proposed a Naive Bayes [4] based model wherein tweets are classified into various categories or clusters in order to help the user read only a particular type of tweets. In order to evaluate the performance of the system, a dataset which is an in house dataset containing 1000 tweets exclusively related to Covid19 was used. We have collected a dataset of roughly 1000 tweets by parsing Twitter data and collecting the relevant tweets. These tweets are then allotted into 10 classes by employing the spectral clustering method [5] on the cosine similarity of the tweets among themselves. Here, the number of classes is finalized through exhaustive experimentation as described in Section 4.

Not only to prepare a gold standard dataset, we have also proposed the Naive Bayes method which is indeed a commonly used simple but effective classifier for phrase classification. It is based on the commonly known Bayes' Theorem [6] in probability wherein in this case, the probability of each of the tweets belonging to a certain class is obtained. Thus, in a nutshell, the chief highlights of our work include:

- Proposed three novel in house datasets of 21000 tweets related to COVID-19
- Implemented three modified versions of Naive Bayes algorithm from scratch to classify the test set into appropriate classes.
- Performed exhaustive experimentation to finalize our methods which include choosing optimal number of clusters, choosing optimal clustering method, and tuning our Naive Bayes algorithm(s) to fit our purpose.
- Obtained competitive accuracies upon testing, thus proving the robustness of our dataset as well as our method.

The rest of the paper is organized as follows. Three dataset preparation strategies are discussed in Section 2, 3 and 4 respectively whereas various implementations of Naïve Bayes are described in Section 5. Section 6 illustrates the implications of different experiments along with results. Finally, Section 7 concludes the paper by mentioning future tasks.

2. DATASET 1

2.1. Background

Data Pre-processing: The steps that were used on the tweets are as 1) Removal of user, 2) Removal of URLs, 3) Removal of punctuations, 4) contracting unnecessary white spaces, 5) Replacing emoticons with corresponding meaning, 6) Partitioning hash tags etc.

Manual Reviewing: Conducting some manual reviewing we found the following observations; 1) Tweets with four or less words are likely linked to some news article and most of them didn't really convey the whole idea. So we discarded those tweets. 2) Some tweets which were replies to

[1] https://ieee-dataport.org/open-access/english-language-tweets-dataset-covid-19

other tweets sometimes didn't make sense. 3) There were some tweets which were paged like (3/3). Tweet (2/3) was after a lot of tweets. 4) Some tweets belonged to multiple classes. For example: A tweet that contains the statsof recoveries and death could be classified as both news and health.

Data Embedding: To convert the sentences into embedded vectors, TF-IDF was used, with 9774 documents.

Label Identification: A map with key as word in a tweet and value as the count of occurrences in the dataset contained nearly 30000 labels. The map was sorted in descending order, and a list was prepared, using these words. We aim to classify tweets under various classes generated during the "Covid-19 pandemic" in the months of July to August, 2020. Hence, words like "covid19","coronavirus","corona" have been filtered out as they had the highest number of occurrences. From the prepared list, 18 words which have occurred in at least 200 tweets were selected to be the final labels list. The list hence prepared consists of only unigrams.

2.2. Approach

We have chosen three different approaches to tag the given data and manually verify:

Clustering using K-RMS [7]: At first a map is generated between PCA applied embedded data and data cluster points. By that a text file is generated where one can see which tweet is in which cluster. After that labelling of 18 clusters, the output is generated label of 6000 tweets.

The algorithm, "K-RMS" is devised such that it solves issues like the handling signed data problem. It also decreases the number of iterations and increases the accuracy to a great extent. It is observed that if RMS (Root Mean Square) value is used instead of average value, it is expected that the number of iterations will decrease significantly for large datasets. This is because RMS value is much more exact and fast converging in every field of science be it chemistry (VRMS or Root Mean Square Velocity) or some other fields like electrical circuits, etc. It also takes care of negative values in datasets. The degree of changes that takes place during the workflow of the algorithm is lesser compared to that when average value is used.

Cosine Similarity: The vectors for all the tweets and the 18 labels were obtained using TF-IDF technique. Depending on the cosine-similarity value of each tweet's vector and label, if non zero then the tweet was classified under the label giving non-zero value. With this approach labels which are directly represent in the respective tweets is given more preference.

On fitting the processed tweets we obtained vocabulary whose size was 24650 and based on the 18 labels that we had taken into consideration the cosine similarity was done on the vectors of dimensions 1x24650. For each tweet, out of 18 labels the one whose cosine similarity was the highest was considered to be the label that needs to be assigned for the tweet. This strategy was used to label all of the data. The training data was 60% of all labelled tweets and rest formed the test data. We had the idea in mind that if tweet X can label "news" and has the word "news". Another tweet Y which does not have the occurrence of word news but also needs to be labelled "news" by the doing cosine similarity we will likely have the same structure for the sentence (by structure we mean words used and basically the way a tweet is presented for a particular class/label).

Multi Class Annotation: As a tweet might have more than one label, tweets have been labelled as: -> 'label(1) label(2) label(3) … label(N)' depending on whetherthe tweet contains the label(i). Using this approach, 217 odd classes with multiple labels, hereafter referred to as

S-Classes, have been obtained. Many tweets didn't have any of the labels in them. Thus, those tweets have been removed from the training dataset. The number of tweets in the labelled dataset is found to be 3646.

The tweets are transformed into vectors using TF-IDF technique. Multinomial Naive Bayes and Complement Naive Bayes models are used to fit the dataset and prepare it for testing. For testing the model, we considered two directions: 1) Exact Prediction and 2) Subset Prediction. For exact prediction, the predicted class and labelled class are compared on equality. For subset prediction, it is determined if the predicted class is a subset of the labelled class on the basis of (Predicted Class - Labelled Class) ['-' being set difference operator here]. If the operation will return a null, it is considered as a correct prediction. The data was trained using 5 -fold cross validation and the best model was picked to obtain the results. We also present the listof labels which were most incorrectly predicted.

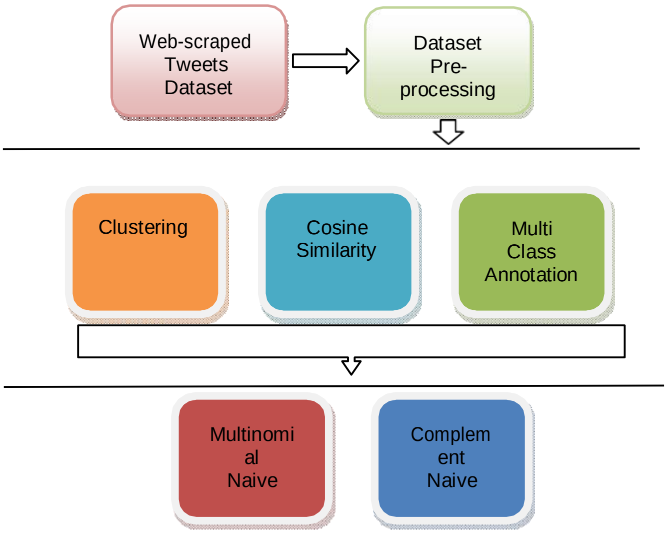


Figure 1. Workflow of Preparing Dataset 1

3. DATASET 2

We have collected 1000 relevant clean tweets related to COVID-19. Our intention was to label the tweets in order to fit them for supervised learning. For this, we have used spectral clustering as well as k-means clustering on the basis of the cosine-similarity matrix generated on the tweet dataset.

In order to obtain the cosine-similarity matrix, we need the word count of the words in each document. We first generate a sparse matrix M of dimension $m \times n$ where m is the total number of tweets and n is the total number of unique words. M_{ij} is an integer value denoting the frequency of the j_{th} word in the i_{th} tweet. For the sparse matrix M, each row indicates an n-length vector. For example, let us consider the r_{th} rowbe $[M_{r0}, M_{r1}, M_{r2}, \dots, M_{r(n-1)}]$ and let us consider the s_{th} row be $[M_{s0}, M_{s1}, M_{s2}, \dots, M_{s(n-1)}]$. Now, in n-dimensional space, the two vectors can be considered as two points. The absolute value of the cosine of the angle between the two vectors in the n-dimensional space is considered as the cosine similarity value of the two particular tweets.

Through this process, we get an $n \times n$ size cosine similarity matrix S, where S_{ij} denotes the similarity between the i_{th} tweet and the j_{th} tweet $(0<= S_{ij} <=1)$. The larger the value of S_{ij}, the more similar the i_{th} and j_{th} tweet are to each other. Clearly, S is a symmetric matrix. On S, we have done two types of clustering for experimentation andultimately choose the better one. The types of clustering experimented with aredescribed as follows:

K-means clustering: Here we have just applied the *k*-means clustering algorithm, where the distance between the two *ith* and *jth* data points are considered as the value of S_{ij} or S_{ji}, as both are equal.

Spectral clustering: Spectral clustering is an unsupervised clustering algorithm. It treats the data points as nodes of a graph. The similarity between the data points are calculated using some metric. The technique makes use of the eigenvalues of the similarity matrix and then performs dimensionality reduction. The clustering is formed in fewer dimensions [8]. The steps are as follows.

a) Here the algorithm generates an undirected graph of n nodes, considering S_{ij} as the weight of the edge between i_{th} and j_{th} node.

b) From S, we generate D as a diagonal matrix, where D_{ii} is the sum of all S_{ik} for all k between 0 and $n-1$.

c) Now we apply the formula to obtain the Laplacian Matrix L where $L = D - S$.

d) From L we calculate normalized Laplacian matrix $L_{norm} = D^{(-½)}LD^{(-½)}$

e) From L_{norm} we calculate first k eigenvectors $v_1, v_2,..., v_k$.

f) Let U be the matrix containing the vectors $v_1, v_2,..., v_k$ as columns.

g) For $i = 1,..., n$, we take the i_{th} row of U as its feature vector afternormalizing to norm 1.

h) Then, we cluster the points with k-means into k clusters $C_1,..., C_k$

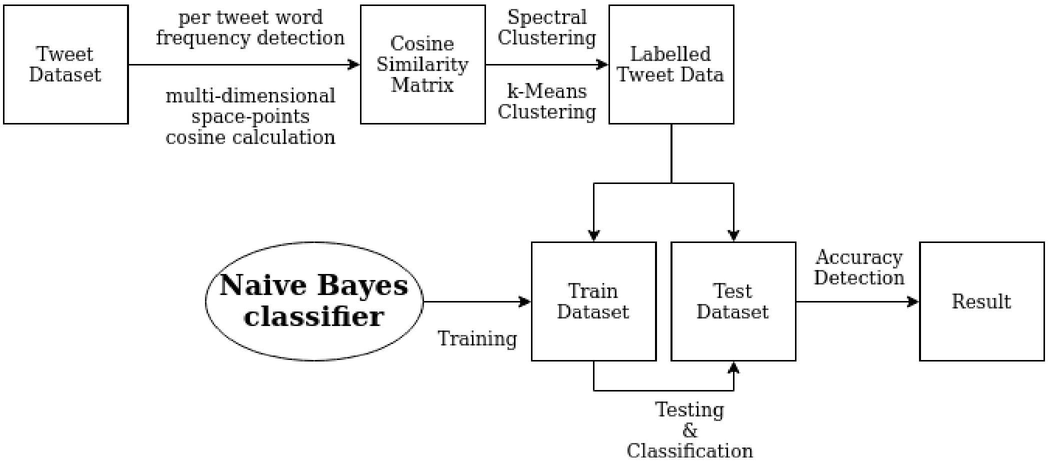


Figure 2. Flowchart representing the module-wise workflow of Dataset 2

4. Dataset 3

The dataset consists of 10,000 tweets. The data crawling is done using Tweepy, whichis an easy-to-use Python library for accessing the Twitter API. For accessing Tweepy, authentication to Twitter API is required. The data is then cleaned and preprocessed by removing urls, emojis and special characters and then stored in a .csv file. This code collects the 10,000 most recent tweets with respect to the search words, and filters out retweets. Then the preprocessing is done and the data is stored in the .csv file.

Then the first 100 tweets are manually labelled and stored in a file covid- labeled-data.csv. Then Naïve Bayes is applied to the data to generate the other labels. This is stored in a file covid-test-data.csv. Spectral clustering has been done on the datato test accuracy of the algorithm. The input for spectral clustering is covid-cleaned- data.csv. The labels are auto-generated using this method and are used to verify with the Naïve Bayes algorithm.

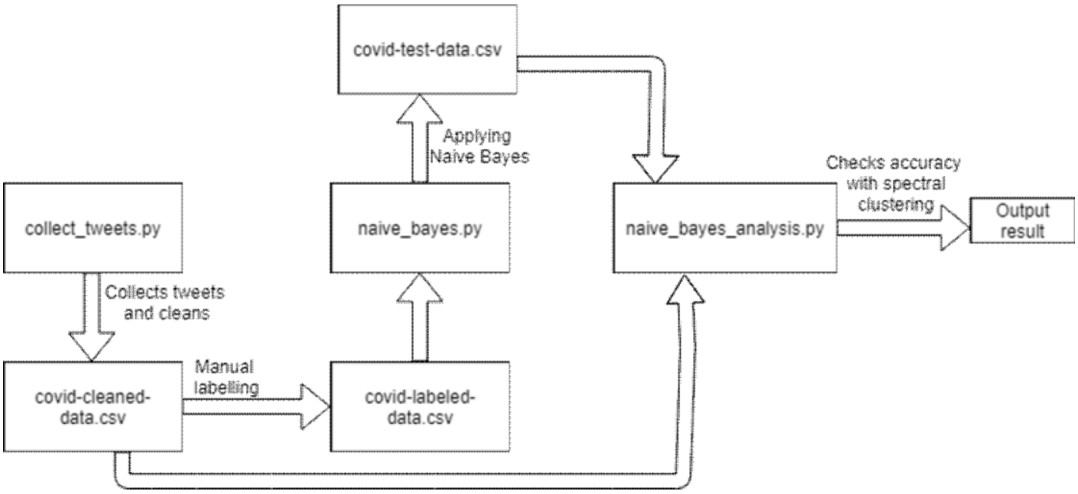


Figure 3. The structure diagram showing module-wise workflow of Dataset 3

5. Naïve Bayes Model

Multinomial Naïve Bayes is a classification technique based on Bayes' Theorem. In this classifier, "bag of words" document representation is used. The words and phrases are the features. The Naïve Bayes model assumes that the feature probabilities are independent given a class c. The conditional probability of belonging to a class c given the document d is calculated. The Naïve Bayes algorithm is given in detail in [1].

Bayes Theorem states that, for two events A and B, if probability of event A occurring is P(A), probability of event B occurring is P(B), and probability of event A occurring given that event B has already occurred is P(A/B) , then probability of eventB occurring given that event A has already occurred is P(B/A) = (P(A/B)*P(B))/P(A).

$$P(A \cap B) = P(A/B).P(B) = P(B/A).P(A)$$
$$P(B/A) = (P(A/B).P(B))/P(A)$$

, where $P(A \cap B)$ is the probability of occurring A and B together. Now, in our context, suppose c is a class and d is a document. Now given the document d, the probability of it belonging to class c is:

$$P(c/d) = (P(d/c).P(c))/P(d)$$

We will have multiple classes $\{c_1, c_2, c_3, ...c_N\}$ and we have to figure out inwhich class out of these, our given document d belongs to. So, if out of N classes, the probability corresponding to the i^{th} class $P(c_i/d)$ is highest, then we say that document d belongs to the ith class ci. One thing to notice here is that, while calculating the probability for each of the classes, the term P(d) in the denominator on the right side is exactly the same. So, we can simply ignore that term, as it won't relatively affect the results. So the our equation becomes: $P(c/d) = P(d/c).P(c)$.

Now, our document d may contain M words $x_1, x_2, x_3 \ldots x_M$. So our equation becomes:

$$P(c/d) = P(d/c).P(c) = P(\{x_1, x_2, x_3 \ldots x_M\}/c).P(c) = P(x_1/c).P(x_2/c) \ldots P(x_M/c).\ P(c)$$

We call this algorithm Naive Bayes because we simply neglect any interrelation between the words in the document. So, we calculate the probability of each class P(c) from the given training dataset. Then, we select the “Bag Of Words” which, in our case, are all the words that appear in the training dataset. Next, we calculate the probability of each unique word from the bag of words belonging to a particular class c, i.e, P(x/c). Finally, we apply the Bayes theorem on the testing data document and find the class with highest probability and then we check how accurately we are able to predict.

$$C_{map} = \text{argmax}_{c \in C} [\ P(c).\Pi_{x \in d} P(x/c)]$$

The Naïve Bayes experiments have been conducted on these three varieties of datasets and important observations were grouped into three different sections described as follows.

6. Experiments and Results

6.1. Observations 1

The general observation is that in all the different models, complement Naive Bayes works better relatively, because the model is designed to handle datasets which are class-imbalanced, and can be visualized from the above given histogram. The accuracy obtained using the cosine similarity label tagging is higher compared to the remaining two methods, since the classifier guesses the label based on its existence in the tweet. Also, in most cases this was the expected tag of the corresponding tweet. Except in some abstract cases where the label of the tweet isn’t occurring in the tweet itself.

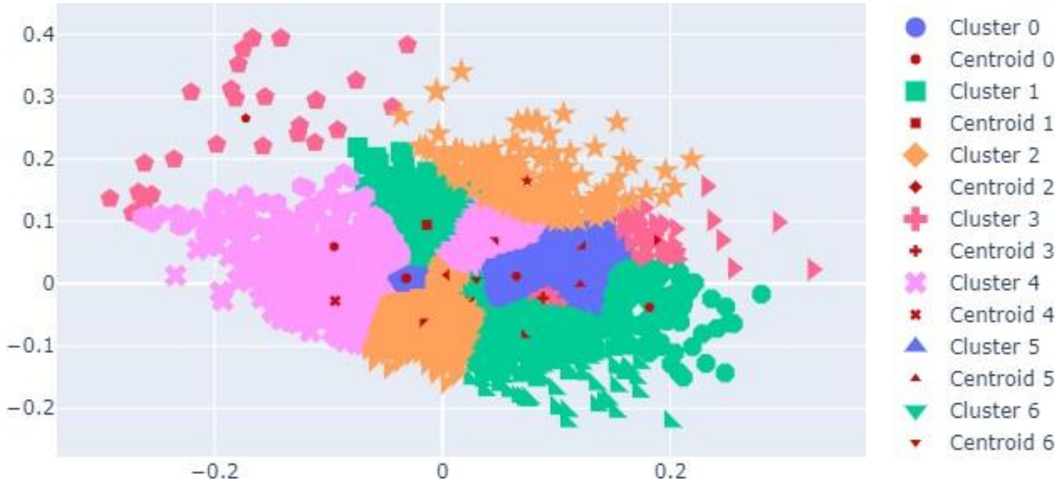


Figure 4. K-RMS Clustering results with 18 clusters on Dataset 1

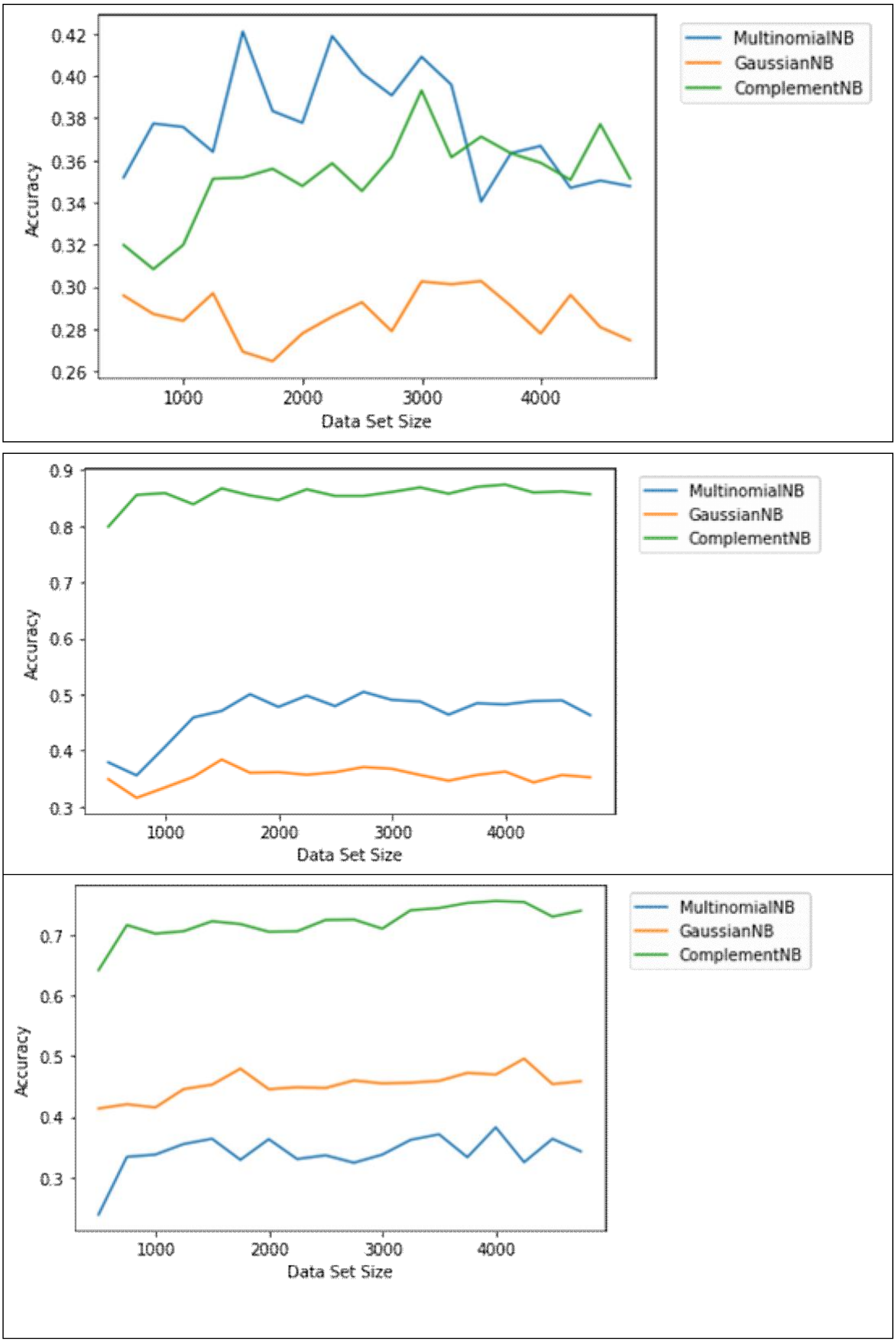


Figure 5. Performances of Multinomial, Gaussian and Complement Naïve Bayes onDataset 1

The highest accuracy obtained using 5 - fold cross validation is 88 per cent, labelled using Cosine Similarity and model used is Complement Naive Bayes (as shown in Figure 4 and Figure 5). The clustering model wasn't very accurate with the cluster formation since the classifier seems to be confused while assigning a specific label. The different methods have predicted some words most incorrectly, for example, using theclustering method, economy is the most incorrectly labelled word. It is due to the fact that the data belonging to the economy class is not properly clustered under one label, and belongs to multiple clusters. Also, mask is the most incorrectly labelled word for cosine similarity; it is due to the fact that it doesn't generally occur within a tweet. The plots comparing the different methods of tagging, are illustrated below, for comparison of performances, over increase in the size of data.

Most Incorrectly Predicted Words [Complement NB]	
KNN Cluster	**Cosine Similarity**
Economy	Mask
Outbreak	Death
Mask	China

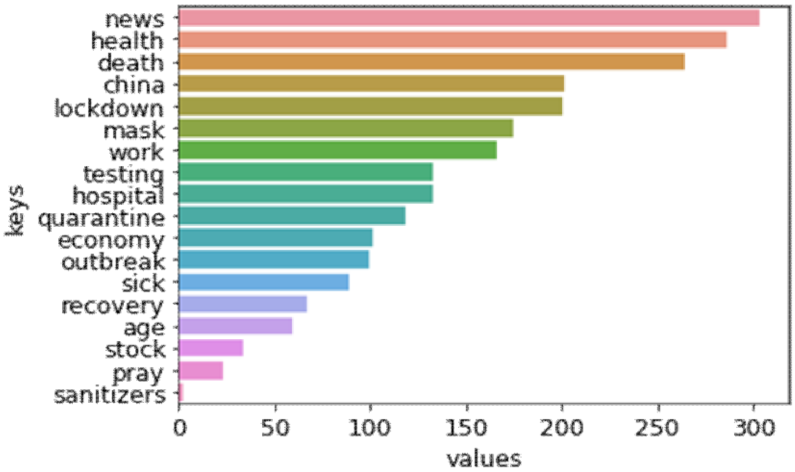
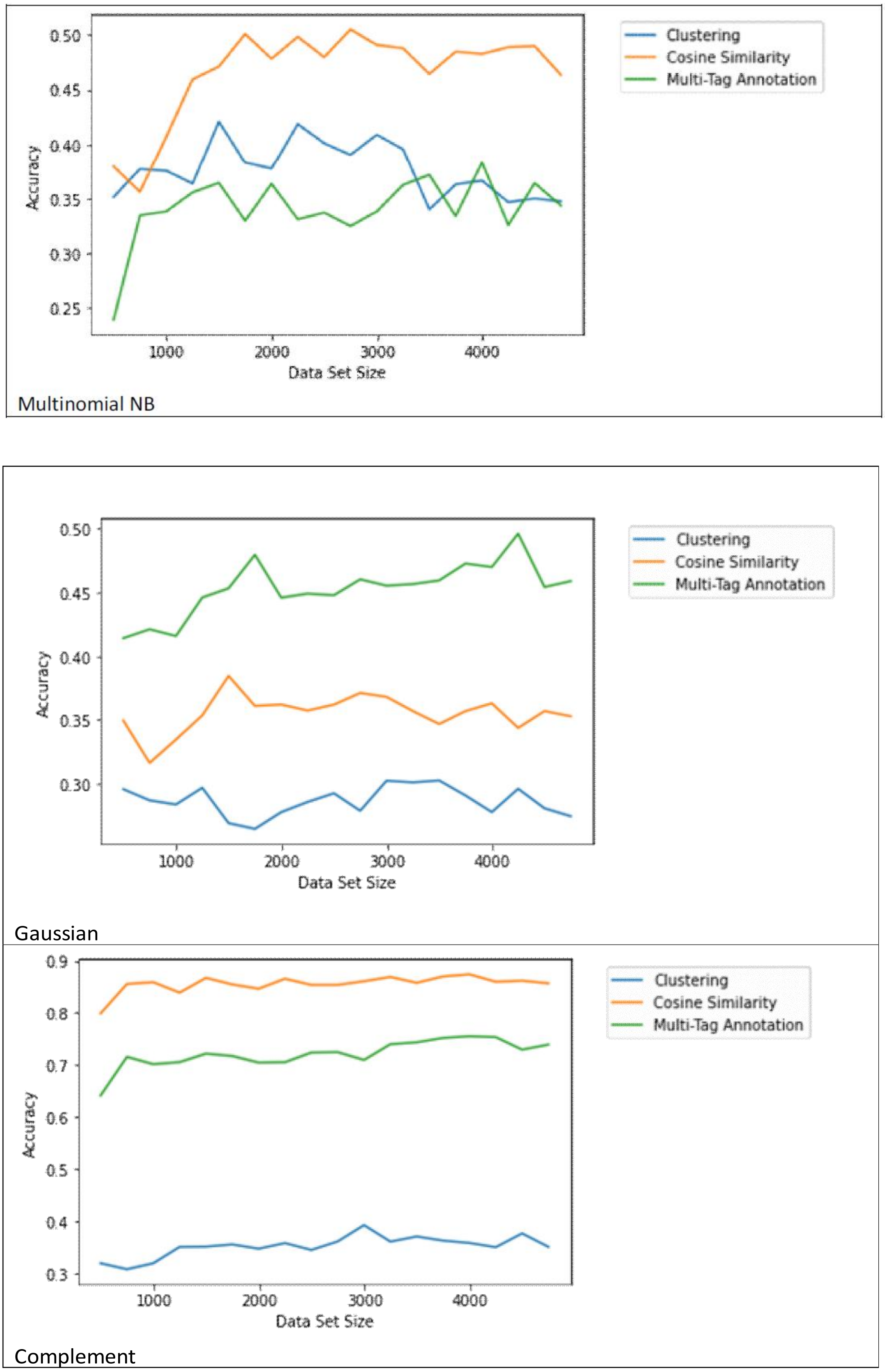


Figure 6. Predicted Results with frequencies on Dataset 1



Multinomial NB

Gaussian

Complement

Figure 7: Performances of Multinomial, Gaussian and Complement Naïve Bayes withrespect to three techniques (clustering, cosine and multi-tag) on Dataset 1

6.2. Observations 2

In this experiment,

- For each kind of clustering we have prepared datasets consisting of 5 to 15classes.
- For each such class, 70% of the total dataset has been considered for training and the rest for testing.
- The results of all these experiments are represented in a comprehensive tabular form in Table 1.

Table 1: Accuracies obtained for all 5-15 classes for both Spectral and K-meansclustering on Dataset 2

Number of classes	Accuracy (%)	
	Spectral clustering	K-means clustering
5	66.33	64
6	65	66.67
7	69	66
8	68	63.33
9	69.33	67.33
10	73.67	64
11	69.33	65.33
12	69	66.67
13	72.33	60.67
14	70.33	54
15	62	69.33

To further validate our method, we have tested the entire process on another publicly available dataset. This dataset [7] consists of a collection of 100 tweets which is divided into train and test sets in the ratio of 1:4. Since the dataset is already available, we know the most optimum number of classes that we should get as output (4 in this case). Hence, an answer near 4 will indicate the robustness of our method. As before, we test across a variety of classes (2-8) using the two clustering methods andtake the best accuracy as our answer. The results are illustrated in Table 2.

Table 2: Accuracies obtained for all 2-8 classes for both Spectral and K-meansclustering on the second dataset

Number of classes	Accuracy (%)	
	Spectral clustering	K-means clustering
2	95	95
3	85	100
4	95	95
5	100	95
6	100	90
7	95	95
8	90	90

From Table 1, it is evident that in most of the cases, Naive Bayes achieves more accuracy for Spectral clustering than K-means clustering. Except for 5-class and 15- class datasets, K-means clustering has more accurate results than that of Spectral clustering. The highest accuracy for Spectral clustering is 73.67% when there are a totalof 10 classes whereas a highest accuracy of 69.33% is achieved for the 15 class case of K-means clustering. On average, Naive Bayes achieves more accuracy for Spectral clustering than K-means clustering because of the eigenvector set generation with the help of the normalized Laplacian matrix in spectral clustering algorithm, which actually helps to more effectively cluster the data points in an n-dimensional space, in comparison to simple k-means clustering, where only cosine matrix element values are taken as the distance between two data points. In spectral clustering, k-means clustering algorithm runs at the last stage with respect to the eigenvector set. For datasets containing a wide variety of classes, spectral clustering (k-partitioning of connected graph) is more effective. Since the accuracy for 10 classes with spectral clustering is the highest, we consider that to be the output of our method. The top 5 words of each of the 10 classes are illustrated in Table 3.

Table 3: Top 5 most frequently used words in each class for 10 classes with spectralclustering along with the suggested class labels on Dataset 2

Class no.	Most Frequent Words	Suggested Class Label
1	china, pandemic, cases, india, covid19	covid impact on asian countries (withlarge population)
2	funds, news, order, until, vaccine	covid-vaccination
3	cases, deaths, people, coronavirus, covid19	covid-deadliness
4	stock, trading, warns, york, coronavirus	impact of covid on economy
5	working, nurses, salary, staff, sir	occupation and health-workers related
6	like, pictwittercom, corona, covid, coronavirus	general covid-related information (occasionally pictorial)
7	coronavirus, lot, masks, people, see	masks and covid awareness
8	affected, flood, jumps, spike, tally	covid statistics
9	positive, report, tested, thursday, washington	covid testing in washington
10	they, my, we, you, covid19	Personalised Covid-related tweets

From Table 2, we note that our method is indeed robust as for spectral clustering, the optimum value of 100 is reached with number of classes=5 which is very near the optimum value of 4. We also note that the accuracy for 4 classes is 95 which mean that only one sample is classified incorrectly which is negligible. Further, the best accuracies are obtained around 4 classes which denote that our method is logically sound.

6.3. Observations 3

In this particular experimental set up, the first 100 tweets are manually labelled. These 100 labelled tweets have been considered form the training data used for the Naïve Bayes classifier. The tweets are divided into four classes. They are as follows:

1. Covid-19 prevention mechanisms
2. Statistics related to Covid-19
3. Vaccine or medications related to Covid-19
4. Tweets related to other topics

Using manual labelling, 11 tweets were classified as class-1, 19 as class-2, 14 as class-3 and 56 as class-4. So P(1)=0.11, P(2)=0.19, P(3)=0.14, P(4)=0.56. The words and phrases related to covid-19 are stored in a list. They consist of monograms, bigrams and trigrams and were hard-coded into the list. The following is the list of words and phrases used as the features:

Table 4. Monograms, bigrams and trigrams used on Dataset 3

Monograms	**Bigrams**	**Trigrams**
Case/s, Death/s	Social distancing	Case Fatality Rate
Test/s, Hospitalization/s	Recovery rate	
Mask/s, Pandemic	Fatality rate	
China, Recovery, Lockdown	Contract tracing	
Quarantine, Vaccine/s, Moderna	Herd immunity	
Pfizer, Immunity		

The first 100 tweets were used form the training data and the remaining 9900 tweets form the test data. In the second experiment, spectral clustering is done on the dataset. The first 200 tweets are divided into four clusters using spectral clustering algorithm. The cosine similarity between the tweets is used to create the similarity matrix. The clusters are generated from the cosine similarity matrix. The data is now trained with first 100 tweets and tested for the next 100 tweets. A counter is set so that when the classes obtained by the Naïve Bayes classifier and the spectral clustering match, the counter is incremented. Using the formula for accuracy,

Classification accuracy = Correct predictions / Total predictions

The accuracy is calculated to be 86%.

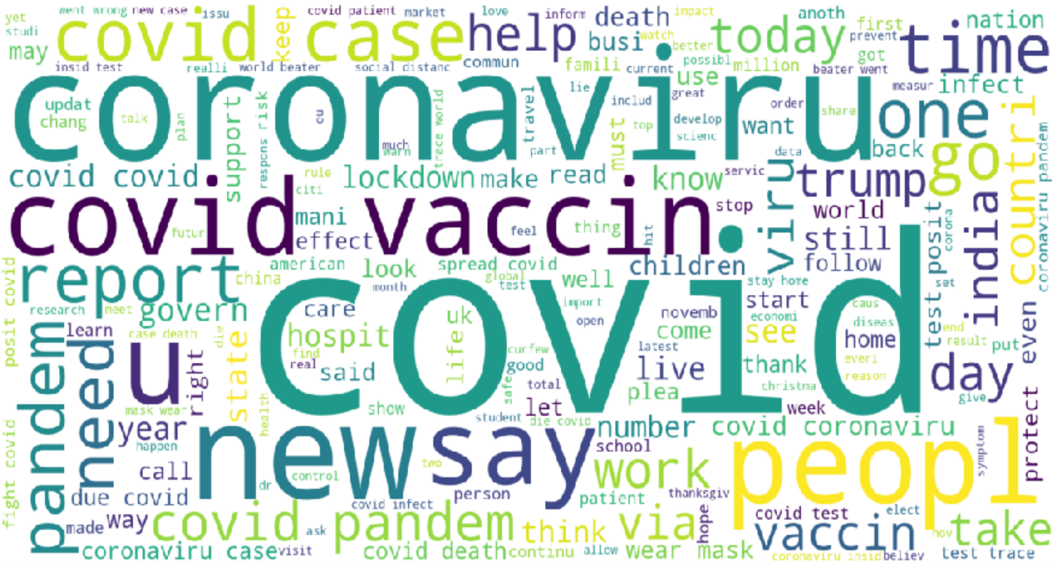


Figure 8. Word Cloud on Covid-19 Dataset 3

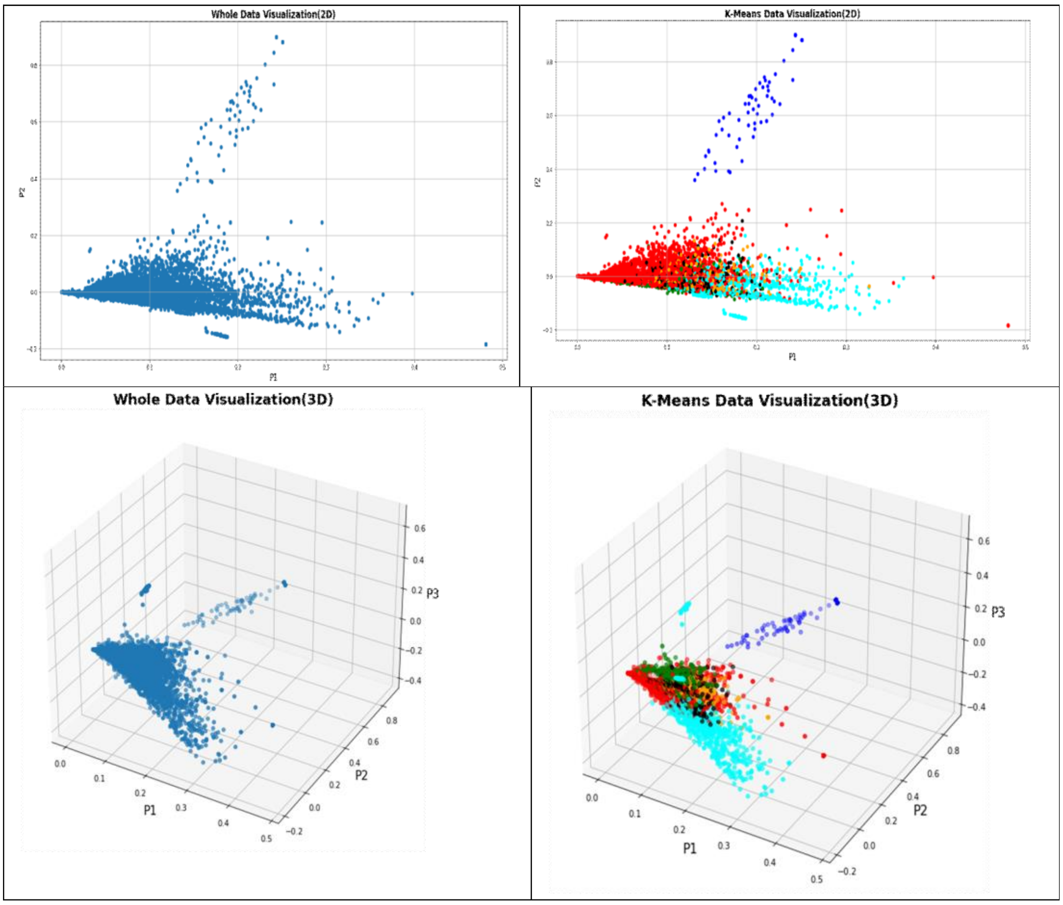


Figure 9. Clustering Results on Dataset 3 in 3D view

K-Means clustering algorithm is used here to label the tweets. The best number of clusters is found using silhouette_score. The number of clusters is varied from 2 to 9. n_clusters = 8 gave silhouette_score = 0.007378162655447713 which is the best. So, number of clusters is taken as 8. Here is different n_clusters vs silhouette_score plot,

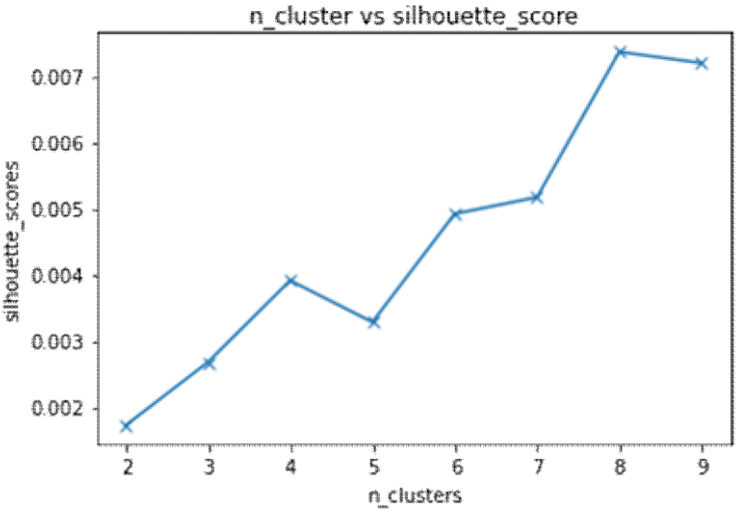
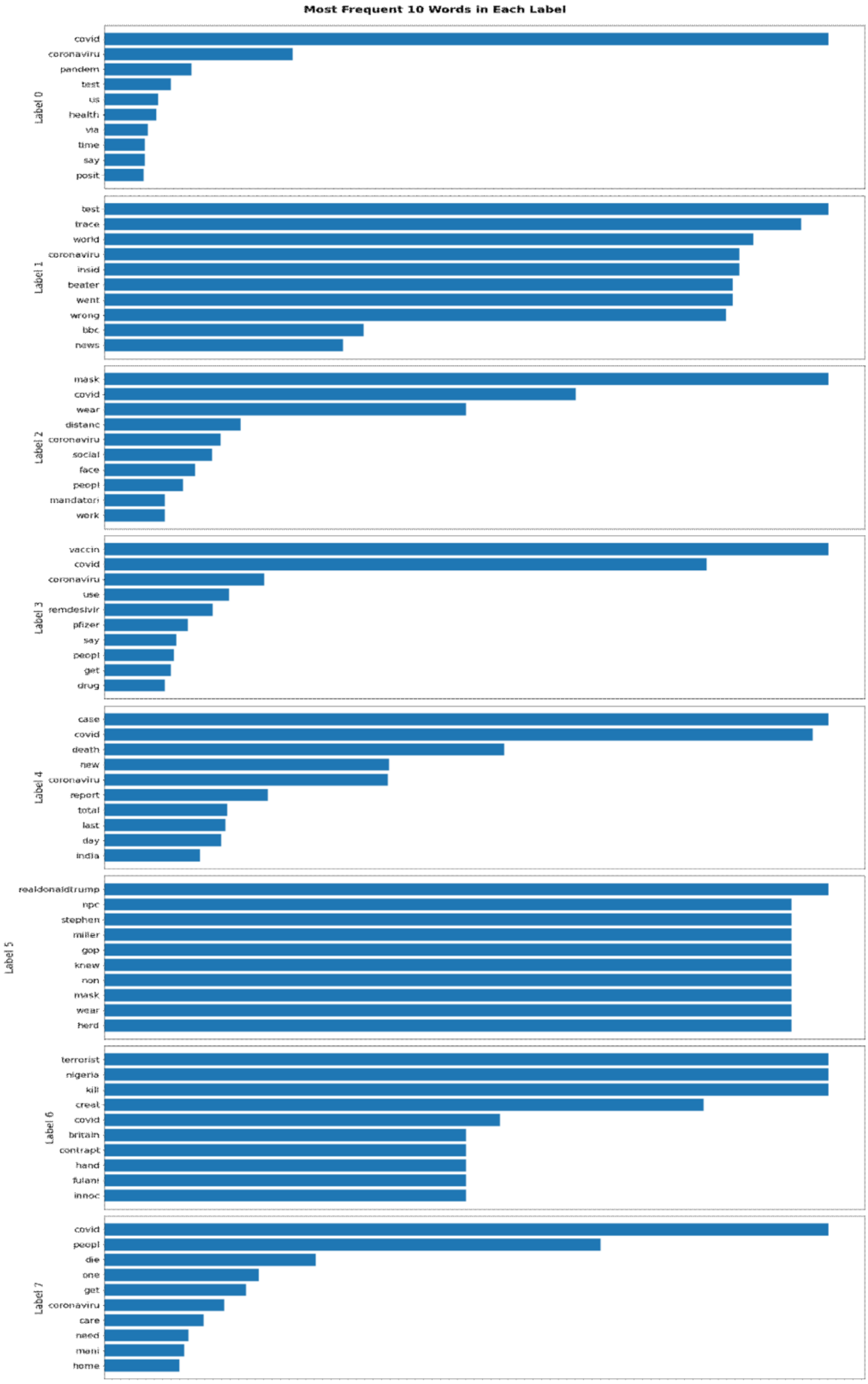


Figure 10. Plot between n_clusters and silhouette_score


Most Frequent 10 Words in Each Label
Label 0
covid
coronaviru
pandem
test
us
health
via
time
say
posit
Label 1
test
trace
world
coronaviru
insid
beater
went
wrong
bbc
news
Label 2
mask
covid
wear
distanc
coronaviru
social
face
peopl
mandatori
work
Label 3
vaccin
covid
coronaviru
use
remdesivir
pfizer
say
peopl
get
drug
Label 4
case
covid
death
new
coronaviru
report
total
last
day
india
Label 5
realdonaldtrump
npc
stephen
miller
gop
knew
non
mask
wear
herd
Label 6
terrorist
nigeria
kill
creat
covid
britain
contrapt
hand
fulani
innoc
Label 7
covid
peopl
die
one
get
coronaviru
care
need
mani
home

7. CONCLUSIONS

In our work, we have developed a Naive Bayes based Algorithm for classification of Covid-19 related tweets. We have first collected an in-house dataset consisting of 1000 tweets by crawling Twitter and collecting tweets related to Covid. Subsequently, we have assigned them different classes using spectral clustering and *k*-means clustering. Then, using Naive Bayes Classifier which we have implemented from scratch, we have classified the tweets into the various classes. In future, we would like to collect more tweets related to Covid so that the classifier can be better trained with a larger dataset to handle the tweets. Further, we would like to test with more classifiers and perform a comparative study with other classifiers in regard to the performance in classifying our dataset.

The multinomial Naïve Bayes algorithm is implemented. We plan to use more clustering techniques like k-means and experiment with different number of classes. Weplan to compare the accuracy obtained by these different methods and find out the optimal number of classes.

ACKNOWLEDGEMENTS

The work is supported by the project "*Sevak- an Intelligent Indian Language Chatbot*" funded by DST-SERB, MeiTY, Govt. of India.

REFERENCES

[1] Sharma, Karishma, Sungyong Seo, Chuizheng Meng, Sirisha Rambhatla, and Yan Liu. "Covid-19 on social media: Analyzing misinformation in twitter conversations." *arXiv e-prints* (2020): arXiv-2003.

[2] Dimitrov, Dimitar, Erdal Baran, Pavlos Fafalios, Ran Yu, Xiaofei Zhu, Matthäus Zloch, and Stefan Dietze. "TweetsCOV19-A Knowledge Base of Semantically Annotated Tweets about the COVID-19 Pandemic." In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pp. 2991-2998. 2020.

[3] Sarker, Abeed, Sahithi Lakamana, Whitney Hogg-Bremer, Angel Xie, Mohammed Ali Al-Garadi, and Yuan-Chi Yang. "Self-reported COVID-19 symptoms on Twitter: an analysis and a research resource." *Journal of the American Medical Informatics Association* 27, no. 8 (2020): 1310-1315.

[4] Xu, Shuo. "Bayesian Naïve Bayes classifiers to text classification." *Journal of Information Science* 44, no. 1 (2018): 48-59.

[5] Ng, Andrew, Michael Jordan, and Yair Weiss. "On spectral clustering: Analysis and an algorithm." *Advances in neural information processing systems* 14 (2001): 849- 856.

[6] Joyce, James. "Bayes' theorem." (2003).

[7] https://github.com/gituhin/sentence-classification-naive-bayes-