

COVIDSenti: A Large-Scale Benchmark Twitter Data Set for COVID-19 Sentiment Analysis

Submitted in partial fulfillment of the requirements for the award of
Bachelor of Technology degree in

INFORMATION TECHNOLOGY

By

BALAJI P (38120018)
TEKKI DEEPAK (38120084)



**DEPARTMENT OF INFORMATION TECHNOLOGY
SCHOOL OF COMPUTING**

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

**Accredited with Grade “A” by NAAC | 12B Status by UGC | Approved by AICTE
JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600 119**



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Accredited with "A" grade by NAAC | 12B Status by UGC | Approved by AICTE
Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai – 600 119
www.sathyabama.ac.in

SCHOOL OF COMPUTING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the Bonafide work of **BALAJI P (Reg no:38120018)** and **TEKKI DEEPAK (Reg no:38120084)** carried out the project entitled **“COVIDSenti: A Large-Scale Benchmark Twitter Data Set for COVID-19 Sentiment Analysis”** under our supervision from OCTOBER 2021 to JANUARY 2022

Internal Guide

Head of the Department

Dr. P. AJITHA M.E., Ph.D.,

Dr. R. SUBHASHINI M.E., Ph.D.,

Submitted for Viva voce Examination held on _____

Internal Examiner

External Examiner

DECLARATION

We, **BALAJI P** and **TEKKI DEEPAK** hereby declare that the Project Report entitled done by me under the guidance of **Dr. P. AJITHA M.E., Ph.D.**, at Sathyabama Institute of Science and Technology (Deemed to be University), Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai - 600119 is submitted in partial fulfillment of the requirements for the award of Bachelor of Technology degree in Information Technology.

1.

2.

DATE:

PLACE: CHENNAI

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

We are pleased to acknowledge my sincere thanks to Board of management of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. We are grateful to them.

We convey my thanks to **Dr. T. SASIKALA M.E., Ph.D., Dean, School of Computing** and **Dr. R. SUBHASHINI M.E., Ph.D., Head of the Department, Department of Information Technology** for providing me the necessary support and details at the right time during the progressive reviews.

I would like to express our sincere and deep sense of gratitude to our Project Guide

Dr. P. AJITHA M.E., Ph.D., for her valuable guidance, suggestions and constant

encouragement paved way for the successful completion of our project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the Department of **INFORMATION TECHNOLOGY** who were helpful in many ways for the completion of the project.

ABSTRACT

The main goal is to present a domain-specific approach to understand sentiments manifested within people around the globe regarding COVID situation. In order to attain this, corona-specific tweets are acquired from twitter platform. Identification of COVID-19 sentiments from the tweets would allow informed decisions for better handling the current pandemic situation. The used dataset is extracted from twitter provided by the UCI Repository. After gathering the tweets, the dataset is cleaned using the preprocessing techniques, they are labelled and a model is developed which is effective for detecting the actual sentiment behind a tweets related to COVID-19. The contribution of this works is the performance evaluation of various machine learning classifiers using proposed feature set. Tweets are classified as positive, neutral, or negative using machine learning algorithms (Multinomial Naïve Bayes, Support Vector Machine, Logistic Regression) and the deep learning algorithm (LSTM). Performance of classifiers is evaluated on the accuracy, precision, recall and F1-score. Our experiments reveal that the proposed model performs well in perceiving the perception of people about COVID-19 with a maximum accuracy. Our findings present the high prevalence of keywords and associated terms among tweets during COVID-19. Further, this work clarifies public opinion on pandemics and lead public health authorities for a better society.

TABLE OF CONTENTS

CHAPTER	TITLE	PAGE NO
	ABSTRACT	
1.	INTRODUCTION	
	1.1. General Introduction	1
	1.2. Project Objectives	4
	1.3. Problem Statement	4
2.	SYSTEM PROPOSAL	
	2.1. Existing System	5
	2.1.1 Advantages	5
	2.2. Proposed System	6
	2.2.1 Disadvantages	6
	2.3. Literature Survey	7
3.	SYSTEM DIAGRAMS	
	3.1. Architecture Diagram	21
	3.2. Flow Diagram	22
	3.3. UML Diagrams	23
4.	IMPLEMENTATION	
	4.1. Modules	27
	4.2. Modules Description	28
5.	SYSTEM REQUIREMENTS	
	5.1. Hardware Requirements	33
	5.2. Software Requirements	33
	5.3. Software Description	34
	5.4. Testing of Products	34

6.	CONCLUSION AND FUTURE ENHANCEMENT	
	6.1. Conclusion	50
	6.2. Future Enhancement	51
7.	SAMPLE CODING AND SAMPLE SCREENSHOT	52
8.	REFERENCES	74

LIST OF FIGURES

FIGURE NO	TITLE	PAGE NO
3.1	ARCHITECTURE DIAGRAM	21
3.2	FLOW DIAGRAM	22
3.3	UML DIAGRAM	23
5.3.1	ANACONDA DISTRIBUTION	38
5.3.2	DATASCIENCE LIBRARIES	39
5.3.3	CONDA PACKAGES	40
7.9	TRAINING AND VALIDATION AND ACCURACY	73

LIST OF TABLES

TABLE NO	TITLE	PAGE NO
4.1	MODULES	
5.1	HARDWARE REQUIRMENTS	
5.2	SOFTWARE REQUIRMNETS	

ABBREVIATION**EXPANSION**

API	Application Programing Interface
NLP	Natural Language Processing
SVM	Support Vector Machine
LSTM	Long Short Term Memory
DNN	Deep Neural Networks
LMS	Language Models
ML	Machine Learning

1. INTRODUCTION

1.1 GENERAL INTRODUCTION

Sentiment analysis has attracted a lot of attention from researchers over time and has seen drastic changes in the way the analysis is done to gather and extract the emotion that has been depicted directly or indirectly by humans when expressing themselves on social media platforms. The improvement in the availability of internet across the world over the course of time has made it possible to get most of the world population online which in turn has increased the amount of information that is being generated by the social media handles, e.g. Twitter, Facebook, etc. Now, with this humongous amount of information being generated every single day, the data scientists have started to mine the valuable information from the chunk of data that will eventually allow them to study the varied opinions of netizens on social media platforms.

Sentiment analysis of tweets is a popular topic for classifying unseen tweets into various categories. Twitter has become a go-to option for literally every researcher out there for sentiment analysis. However, irrespective of how well liked the platform is, the data which twitter generates in the form of tweets has its own shortcomings.

Tweets can be a string of maximum 140 characters which have now been increased up to 280 characters as of 2020. The new limit is applicable to almost all the languages supported by twitter excluding Korean, Chinese and Japanese as they can incorporate a lot more information in a lesser number of words. Tweets are commonly informal data which may contain slang words, emoticons, abbreviations, incomplete expressions etc. To make the most out of this form of data, experts have studied different tweets and have also analysed their sentiments extensively.

COVID-19 is a highly contagious disease originated by a newly uncovered coronavirus. The symptoms of COVID-19 vary from being mild to moderate respiratory illness. People who are old or are experiencing any ongoing medical issues are more prone to develop consequential illness.

Corona viruses are a gathering of related RNA infections that cause illnesses in warm-blooded animals and feathered creatures. In people, these infections cause respiratory tract contamination that can go from mild to deadly. Mild diseases incorporate a few instances of the basic cold (which is additionally brought about by different infections, overwhelmingly rhinoviruses), while progressively deadly assortments can cause SARS, MERS, and COVID-19. Side effects in different species change: in chickens, they cause an upper respiratory tract illness, while in dairy animals and pigs they cause loose bowels. Antibodies or antiviral medications to forestall or treat human coronavirus diseases are not found yet. Coronavirus spreads from humans to humans in several manners. It causes respiratory problems and causes breathing problems. It has a low fatality rate compared to SARS or MERS. However, as there is no vaccine available and due to the nature of transmission and spread of virus, countries around the world taken lockdown and isolation as the only preventive steps. As a result people's movement being restricted they spent a good amount of time in home or place of stay. This creates an ideal scenario for people to express their views on social networking sites including Twitter. Like in other parts of the world people of India also express their views about COVID-19 on twitter. In this paper, those tweets are used to analyze the behavior and sentiment of people from December 2019 to May 2020. People were very negative about COVID in the early days. Many were not confident to win the fight against COVID. But as the Lockdown happened, people became optimistic about it. Positive tweets are tremendously high in the amount in the month of April and May. This work provides new insights on COVID-19 and people's thoughts about it. Social media is today's way to read people. There is a good number of papers appeared in the literature since the outbreak causes countries to go in Lockdown state. Sentiment analysis and data visualization of Worldwide Twitter data on Covid 19 has been presented.

In the following work, several machine learning-based algorithms are used for classifying the tweets related to COVID-19. Classification algorithms used in the following paper follows the motivation from ordinal regression which predicts the labels on a scale of emotions. Some researchers have concluded that regression problems on solving with machine learning algorithms provide better results in sentiment analysis

over twitter data. The plus point of the following method is that it did much to ameliorate the obtained results.

The current focus of this paper is to determine the impact of the coronavirus pandemic on the people's mind by analysing the sentiments of their comments expressed on social media platforms like Twitter. Tweets were gathered manually specific to #coronavirus and #COVID19 using twitters API.

Twitter is a famous social media platform and microblogging medium where people post and share messages called "tweets". About 500 million tweets per day and 200 billion tweets per year have appeared on Twitter since it has become an important data hotspot for web-based media conversation identified with public and global situations. Unfortunately, it also a major source to create a global panic situation because of spreading fake news. Chakraborty K et al. 2020 described that most COVID-19 tweets are in positive sentiment, but users are frequently engaged in the spreading of negative tweets and not many useful words were found in the word frequency calculation of tweets.

The trends of positive, negative, and neutral tweets state-wise and month-wise in India are captured and presented in this paper. In this paper we have presented the sentiment of people from the Indian state Maharastra as a case study. It shows that people were worried about the COVID-19 situation before Lockdown and after that, though COVID-19 impacted the moral as well as socio-economic life of people still there is a positive sentiment dominates in the opinion of the people. The twitter feeds were full of positive and hopeful tweets of winning the war against the Coronavirus. A dataset with minimal cleaning needed is created having tweets from November 2019 to May 2020 with their Polarity and Sentiments of each State in India. We have specifically presented sentiments around the dates of announcements about lockdown. We have considered the emotional level as positive, negative and neutral. For better understanding of emotion level of people, this model can be further extended to more complex multi-emotion levels such as joy, panic, happiness, sorrow.

The approach mainly consists of the following steps: fetching of the tweets, pre-processing of the tweets, construction of bag of words, scoring method for predicting

polarity of tweets and finally, comparison of different machine learning approaches for classification of domain-specific tweets on all the dataset.

1.2 PROJECT OBJECTIVES

- ✓ The main objective is to increase understanding of public awareness of COVID-19 pandemic trends and uncover meaningful themes of concern posted by Twitter users in the English language during the pandemic.
- ✓ To effectively classify and predict the data.
- ✓ To enhance the performance of overall prediction result.

1.3 PROBLEM STATEMENT

The given challenge is to build a classification model to predict the sentiment of Covid-19 tweets. The tweets have been pulled from Twitter and manual tagging has been done. We are given information like Location, Tweet At, Original Tweet, and Sentiment.

2. SYSTEM PROPOSAL

2.1 EXISTING SYSTEM

In existing system, sentiment analysis and textual analytics is presented, as well as that on ML methods, twitter and NLP. Significant data challenges are evolving and need to be addressed, and strategic information characteristics restructuring data, as well as the ML techniques. The analysis of past epidemics, crisis situational analysis and tracking, has also involved tracking Twitter data. A better understanding of the US's geographical spread concerning the valances of both healthy and unhealthy sentiment. The people in rural areas tweet less than those in urban areas and suburbs, using the spatial distribution of analyzed tweets. This work also notes that food tweets per capita were less in small urban areas than in larger towns and cities. It was revealed using linear regression that in low-income areas, tweets related more to unhealthy sentiment. sentiment analytics has also had avenues for the use of Twitter data.

2.1.1 DISADVANTAGES

- Data inconsistency Problems
- Theoretical Limits.
- Loss of Information.
- Incorrect Prediciton Results.

2.2 PROPOSED SYSTEM

The proposed model is introduced to overcome all the disadvantages that arise in the existing system. This system will increase the accuracy of the classification results by classifying the data based on the tweets. As raw tweets are often short, unstructured, informal, and noisy, the first step of sentiment analysis is to preprocess the data. The Sentiment timeline helps to understand trends in positive and negative sentiment over time. Multinomial Naïve Bayes algorithm predicts the tag of a text such as a piece of email or article. It calculates the probability of each tag for a given sample

and then gives the tag with the highest probability as output. The multinomial distribution normally requires integer feature counts. In the SVM algorithm, we plot each data item as a point in n-dimensional space (where n is number of features you have) with the value of each feature being the value of a particular coordinate. Support Vectors are simply the co-ordinates of individual observation. The SVM classifier is a frontier which best segregates the two classes (hyper-plane/ line). Logistic regression is estimating the parameters of a logistic model (a form of binary regression). Mathematically, a binary logistic model has a dependent variable with two possible values, such as pass/fail which is represented by an indicator variable, where the two values are labeled "0" and "1".

2.2.1 ADVANTAGES

- High performance.
- Provide accurate prediction results.
- It avoids data inconsistency.

2.3 LITERATURE SURVEY

Title: Detecting emotions in English and Arabic tweets

Year: 2019

Author: T. Ahmad, A. Ramsay, and H. Ahmed

Methodology

Assigning sentiment labels to documents is, at first sight, a standard multi-label classification task. Many approaches have been used for this task, but the current state-of-the-art solutions use deep neural networks (DNNs). As such, it seems likely that standard machine learning algorithms, such as these, will provide an effective approach. We describe an alternative approach, involving the use of probabilities to construct a weighted lexicon of sentiment terms, then modifying the lexicon and calculating optimal thresholds for each class. We show that this approach outperforms the use of DNNs and other standard algorithms. We believe that DNNs are not a universal panacea and that paying attention to the nature of the data that you are trying to learn from can be more important than trying out ever more powerful general purpose machine learning algorithms.

Given that the task we are undertaking involves assigning a **set** of labels, we have to be careful about how we use DNNs. There are two obvious ways to proceed. (i) Given a set of features, it would be possible to train a set of classifiers, one per feature. Each classifier will return YES/NO for the feature on which it was trained. Taken together the set of classifiers can be seen as a single multi-label classifier. We refer to classifiers trained in this way as multi-DNNs. (ii) Train a single classifier with a node in the output layer for each feature and then accept every label for which the output-layer node had an excitation level above some threshold. We refer to this kind of classifier as single-DNN.

Emotion classification tasks have traditionally been tackled by obtaining a large corpus, constructing feature sets, pre-processing in sophisticated ways and then making use of any number of black box training algorithms. Indeed, this approach is still

prevalent currently. We demonstrate that with a small-sized corpus and without using any black box algorithms, our results are comparable with systems that have been trained on millions (and sometimes billions) of tweets.

We show that the approach through explicitly constructing a lexicon substantially outperforms both of the DNN algorithms outlined above, and we speculate about why this should be so.

Advantage

The proposed system improves the overall system performance as well as provides easy scalability.

Disadvantage

Less Effective.

Title: Socio-analyzer: A sentiment analysis using social media data**Year: 2019****Author:** A. Bandi and A. Fellah**Methodology**

The usage of social media is rapidly increasing day by day. The impact of societal changes is bending towards the peoples' opinions shared on social media. Twitter has received much attention because of its real-time nature. We investigate recent social changes in MeToo movement by developing Socio-Analyzer. We used our four-phase approach to implement Socio-Analyzer. A total of 393,869 static and stream data is collected from the data world website and analyzed using a classifier. The classifiers identify and categorize the data into three categories (positive, neutral, and negative). Our results showed that the maximum peoples' opinion is neutral. The next higher number of peoples' opinion is contrary and compared the results with TextBlob. We validate the 765 tweets of weather data and generalize the results to MeToo data. The precision values of Socio-Analyzer and TextBlob are 70.74% and 72.92%, respectively, when considered neutral tweets as positive.

A social issue is a problem that influences a considerable number of individuals within a society or around the world. It is often caused by several factors that arise from a conflict between diverse populations. In other words, disputes' occur as a result of all forms of differences among individuals concerning the culture, gender, age, ability, religion, personality, social status, and sexual orientation. Such social issues become social movements and may lead to special law for the people at the workplace or in society. Few examples of social movements are Black Lives Matter, Me Too, Ferguson, Same-Sex Marriage, Keep Families Together, etc. Other social changes include protecting human rights, animal welfare, banning plastic, etc. In today's world, social media play a crucial role in social movements. People are open to post their feelings, opinions, reactions, and emotions towards the campaign. Sometimes we can observe these movements become social media protests or rallying calls. Sociologists study the people's responses on social movements either by conducting interviews, surveys or

through opinion polls. However, this is a tedious process and limited to only a few samples. We integrate the data science with sociology and analyze the vast data from social media by using machine learning algorithms. We mined the Twitter data using hashtags (Metoo) and API calls and analyze the results.

Advantage

High performance and accuracy.

Disadvantage

It is having some scalability problem.

Title: A survey of regularization strategies for deep models

Year: 2020

Author: R. Moradi, R. Berangi, and B. Minaei

Methodology

Deep neural networks have introduced novel and useful tools to the machine learning community. Other types of classifiers can potentially make use of these tools as well to improve their performance and generality. This paper reviews the current state of the art for deep learning classifier technologies that are being used outside of deep neural networks. Non-network classifiers can employ many components found in deep neural network architectures. In this paper, we review the feature learning, optimization, and regularization methods that form a core of deep network technologies. We then survey non-neural network learning algorithms that make innovative use of these methods to improve classification. Because many opportunities and challenges still exist, we discuss directions that can be pursued to expand the area of deep learning for a variety of classification algorithms.

DNNs allow machines to automatically discover the representations needed for detection or classification of raw input. Additionally, the neural network community developed unsupervised algorithms to help with the learning of unlabeled data. These unsupervised methods have found their way to real-world applications, such as creating generative adversarial networks (GANs) that design clothes. The term deep has been used to distinguish these networks from shallow networks which have only one hidden layer; in contrast, DNNs have multiple hidden layers. The two terms deep learning and deep neural networks have been used synonymously. However, we observe that deep learning itself conveys a broader meaning, which can also shape the field of machine learning outside the realm of neural network algorithms.

The remarkable recent DNN advances were made possible by the availability of massive amounts of computational power and labeled data. However, these advances do not overcome all of the difficulties associated with DNNs. For example, there are

many real-world scenarios, such as analyzing power distribution data, for which large annotated datasets do not exist due to the complexity and expense of collecting data. While applications like clinical interpretation of medical diagnoses require that the learned model be understandable, most DNNs resist interpretation due to their complexity. DNNs can be insensitive to noisy training data and they also require appropriate parameter initialization to converge.

Advantage

More Reliable.

Disadvantage

It is less in efficiency and not give perfect result.

Title: Using social media to mine and analyze public opinion related to COVID-19 in China

Year: 2020

Author: X. Han, J. Wang, M. Zhang, and X. Wang

Methodology

The outbreak of Corona Virus Disease 2019 (COVID-19) is a grave global public health emergency. Nowadays, social media has become the main channel through which the public can obtain information and express their opinions and feelings. This study explored public opinion in the early stages of COVID-19 in China by analyzing Sina-Weibo (a Twitter-like microblogging system in China) texts in terms of space, time, and content. Temporal changes within one-hour intervals and the spatial distribution of COVID-19-related Weibo texts were analyzed. Based on the latent Dirichlet allocation model and the random forest algorithm, a topic extraction and classification model was developed to hierarchically identify seven COVID-19-relevant topics and 13 sub-topics from Weibo texts. The results indicate that the number of Weibo texts varied over time for different topics and sub-topics corresponding with the different developmental stages of the event. The spatial distribution of COVID-19-relevant Weibo was mainly concentrated in Wuhan, Beijing-Tianjin-Hebei, the Yangtze River Delta, the Pearl River Delta, and the Chengdu-Chongqing urban agglomeration. There is synchronization between frequent daily discussions on Weibo and the trend of the COVID-19 outbreak in the real world. Public response is very sensitive to the epidemic and significant social events, especially in urban agglomerations with convenient transportation and a large population. The timely dissemination and updating of epidemic-related information and the popularization of such information by the government can contribute to stabilizing public sentiments. However, the surge of public demand and the hysteresis of social support demonstrated that the allocation of medical resources was under enormous pressure in the early stage of the epidemic. It is suggested that the government should strengthen the response in terms of public opinion and epidemic prevention and exert control in key epidemic areas, urban agglomerations, and transboundary areas at the

province level. In controlling the crisis, accurate response countermeasures should be formulated following public help demands. The findings can help government and emergency agencies to better understand the public opinion and sentiments towards COVID-19, to accelerate emergency responses, and to support post-disaster management.

Advantage

High Efficiency.

Disadvantage

Low in accuracy performance.

Title: Abusive language detection: A comprehensive review

Year: 2019

Author: U. Naseem, S. K. Khan, M. Farasat, and F. Ali

Methodology

To provide an organised literature on the detection of Abusive language on Twitter using natural language processing (NLP). In this study, the survey has been conducted on different methods and research conducted on the types of Abusive language used in social media, why it is important? How it has been detected in real time social media platforms and the performance metrics that are used by researchers in evaluating the performance of the detection of abusive language on Twitter by the users. Giving an organised review of past methodologies, including methods, important features and core algorithms, this study arranges and depicts the present condition about this area. The study also talks about the intricacy of hate speech idea which is characterised in numerous stages ad settings. This area of study has an obvious potential for societal effect, especially in digital media and online networks. A crucial step in propelling automatic hate speech detection is the advancement and systemisation of common assets, for example, clarified data sets in numerous dialects, rules, and calculations. This survey study contains all the relevant references related to detection of abusive language on social media using NLP and machine learning methods. Ultimately, it can be as source of references to the other researchers in finding the literatures that are relevant to their research area in the detection of Abusive language on Twitter.

The most recent growing crime is hated speech which is growing not just in up close and personal associations yet additionally in online correspondence. A few components are contributing to this misconduct. On the one side, on internet and informal communities specifically, individuals are most likely to adopt a forceful conduct on account of the obscurity according to these conditions. In contrary to this, people have expanded ability to tell about their expressions through internet, accordingly, adding to the engendering of abusive language too. Since then, this sort of biased

conversation might be amazingly destructive to people, so, social companies as well as government can make profit from identification and anticipation tools. This study contributes a solution to this issue by giving a methodological representation of work already done on this specific topic. In this study, the problem is framed by defining it, outlining identifying strategies and resources. A systematic approach is adopted which critically investigate and analyse both theoretical and practical aspects.

Advantage

Good in Performance.

Disadvantage

Not Reliable.

Title: A comprehensive survey on word representation models: From classical to state-of-the-art word representation language model

Year: 2020

Author: U. Naseem, I. Razzak, S. K. Khan, and M. Prasad

Methodology

Word representation has always been an important research area in the history of natural language processing (NLP). Understanding such complex text data is imperative, given that it is rich in information and can be used widely across various applications. In this survey, we explore different word representation models and its power of expression, from the classical to modern-day state-of-the-art word representation language models (LMS). We describe a variety of text representation methods, and model designs have blossomed in the context of NLP, including SOTA LMs. These models can transform large volumes of text into effective vector representations capturing the same semantic information. Further, such representations can be utilized by various machine learning (ML) algorithms for a variety of NLP related tasks. In the end, this survey briefly discusses the commonly used ML and DL based classifiers, evaluation metrics and the applications of these word embeddings in different NLP tasks.

Text-based data is increasing at a rapid rate, where the low quality of the unstructured text is growing rapidly than structured text. Textual data is very common in many different domains whether social media, online forums, published articles or clinical notes for patients and online reviews given online where people express their opinions and sentiments to some products or businesses.

Text data is a rich source of getting information and gives more opportunity to explore valuable insights which cannot be achieved from quantitative data. The main aim of different NLP methods is to get a human-like understanding of the text. It helps to examine the vast amount of unstructured and low-quality text and discover appropriate insights. Couple with ML, it can formulate different models for the classification of low-

quality text to give labels or obtain information based on prior training. For instance; researchers in the past focused on mining the opinion and sentiments of users about a product, restaurant and movie reviews etc. to predict the sentiment of users. Over the years text has been used in various applications such as email filtering, Irony and sarcasm detection document organization, sentiment and opinion mining prediction, hate speech detection, question answering , content mining, biomedical text mining and many more.

Advantage

Efficiency is good.

Disadvantage

Not give accurate prediction result.

Title: Transformer based Deep Intelligent Contextual Embedding for Twitter sentiment analysis

Year: 2020

Author: U. Naseem, I. Razzak, K. Musial, and M. Imran

Methodology

Along with the emergence of the Internet, the rapid development of handheld devices has democratized content creation due to the extensive use of social media and has resulted in an explosion of short informal texts. Although a sentiment analysis of these texts is valuable for many reasons, this task is often perceived as a challenge given that these texts are often short, informal, noisy, and rich in language ambiguities, such as polysemy. Moreover, most of the existing sentiment analysis methods are based on clean data. In this paper, we present DICE_T, a transformer-based method for sentiment analysis that encodes representation from a transformer and applies deep intelligent contextual embedding to enhance the quality of tweets by removing noise while taking word sentiments, polysemy, syntax, and semantic knowledge into account. We also use the bidirectional long- and short-term memory network to determine the sentiment of a tweet. To validate the performance of the proposed framework, we perform extensive experiments on three benchmark datasets, and results show that DICE_T considerably outperforms the state of the art in sentiment classification.

The experimental results demonstrate the importance of the suggested method, which may allow potential groups of people who want to have plastic-surgery treatments to make better decisions. Recently, Proposed Deep Intelligent Contextual Embedding (DICE), Hybrid Words Representation and Transformer-based Deep Intelligent Contextual Embedding (DICET) models to improve tweet quality by considering polysemy, syntax, semantics, and out of vocabulary (OOV) words; and handling the noise within the textual context. The evaluation of the models executed on airline-related datasets indicated that their proposed models improved the accuracy for tweets classification with an average accuracy of 93.5%, 94.2%, and 94.6%, respectively.

Advantage

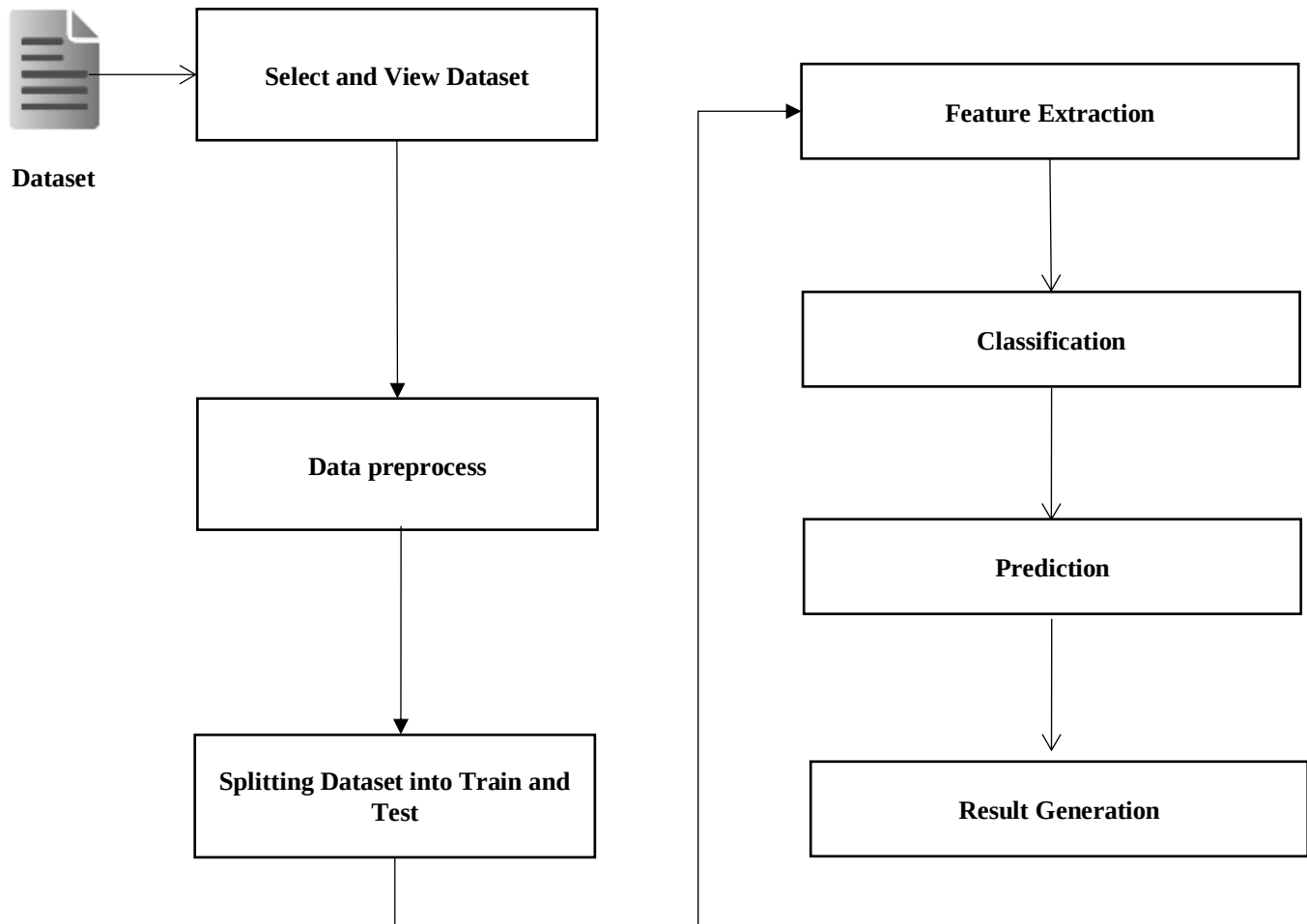
Text corpus for content mining gives good analysis result.

Disadvantage

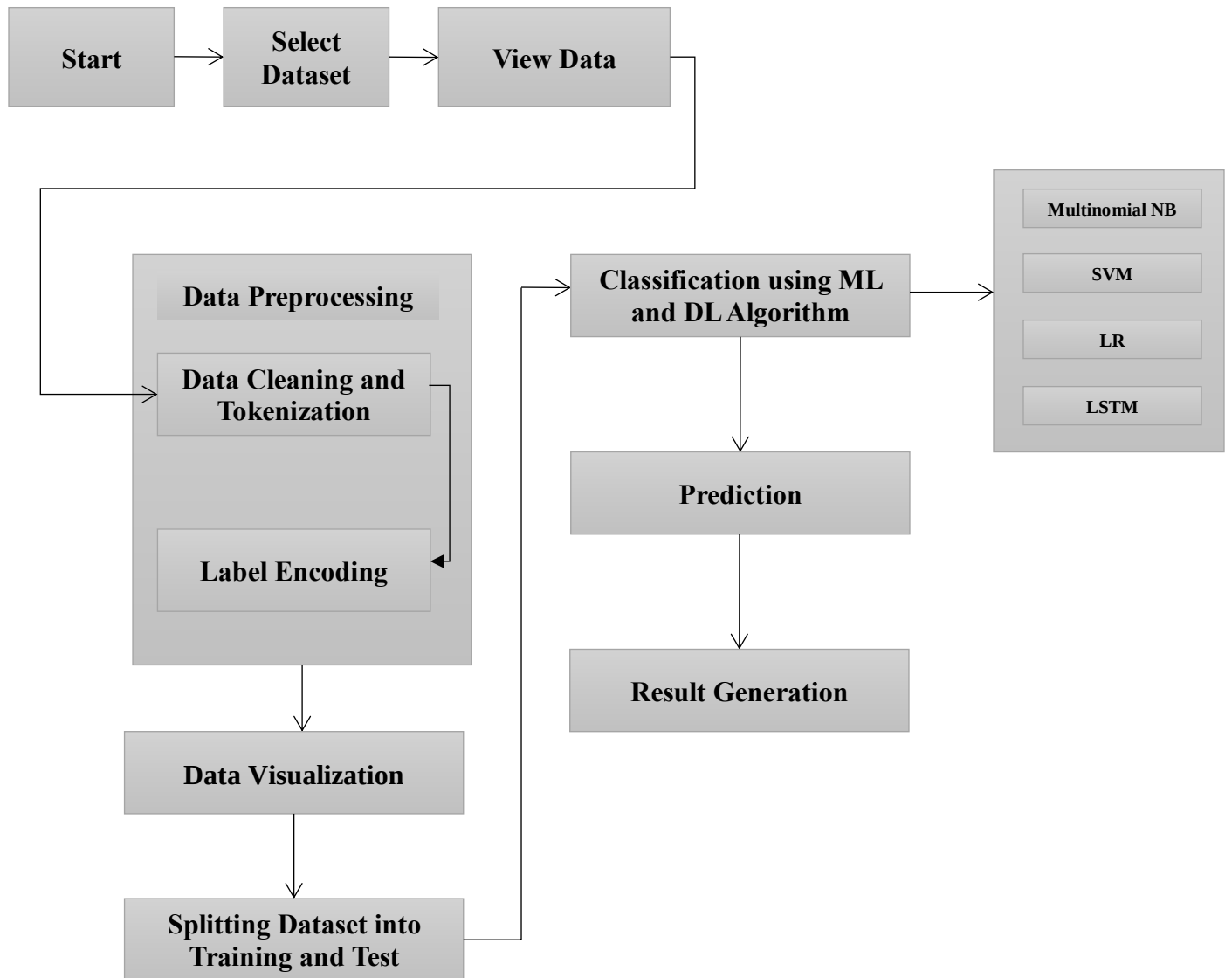
Not give accurate prediction result.

3. SYSTEM DIAGRAMS

3.1 ARCHITECTURE DIAGRAM

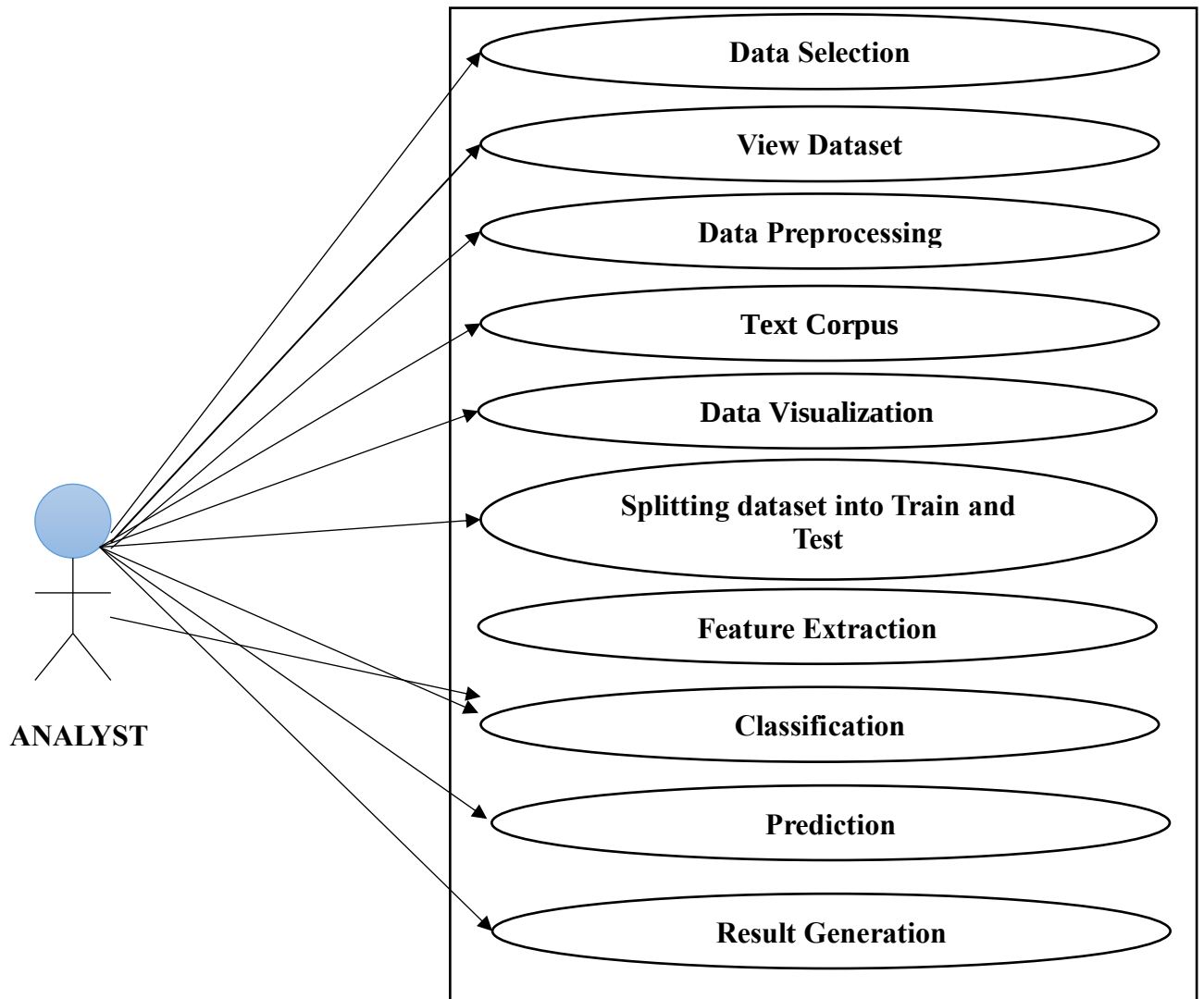


3.2 FLOW DIAGRAM

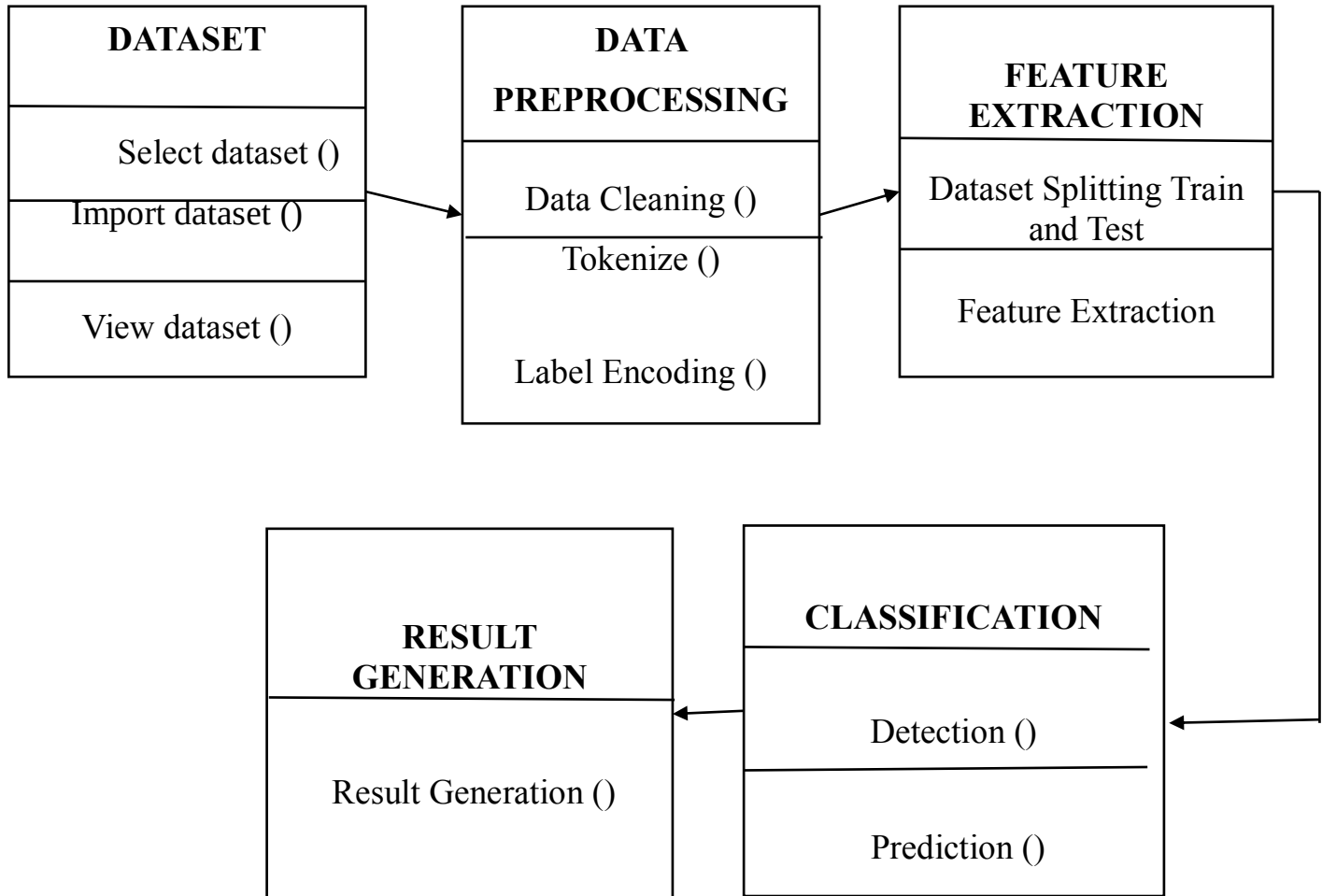


3.3 UML DIAGRAMS

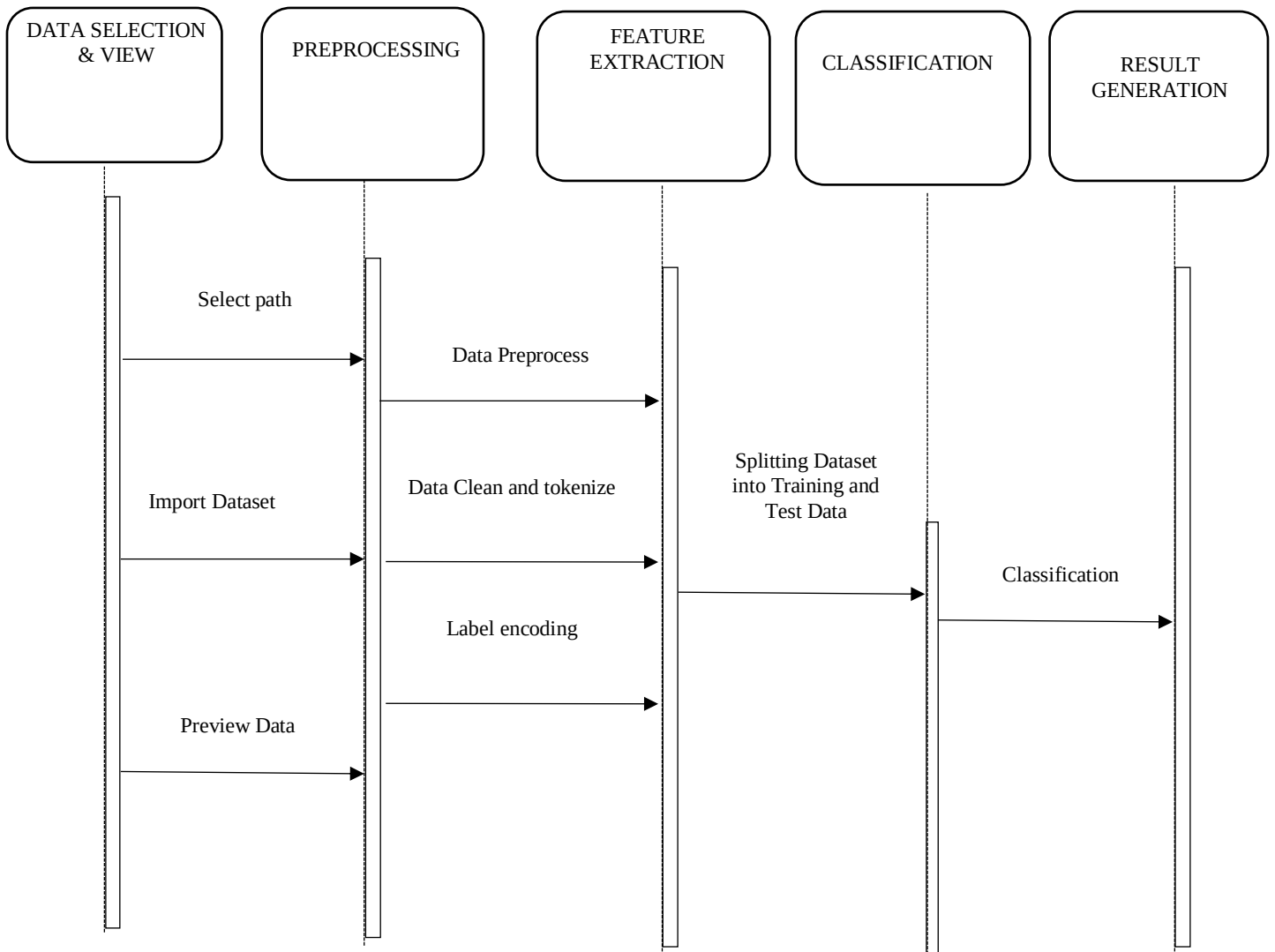
USE CASE DIAGRAM



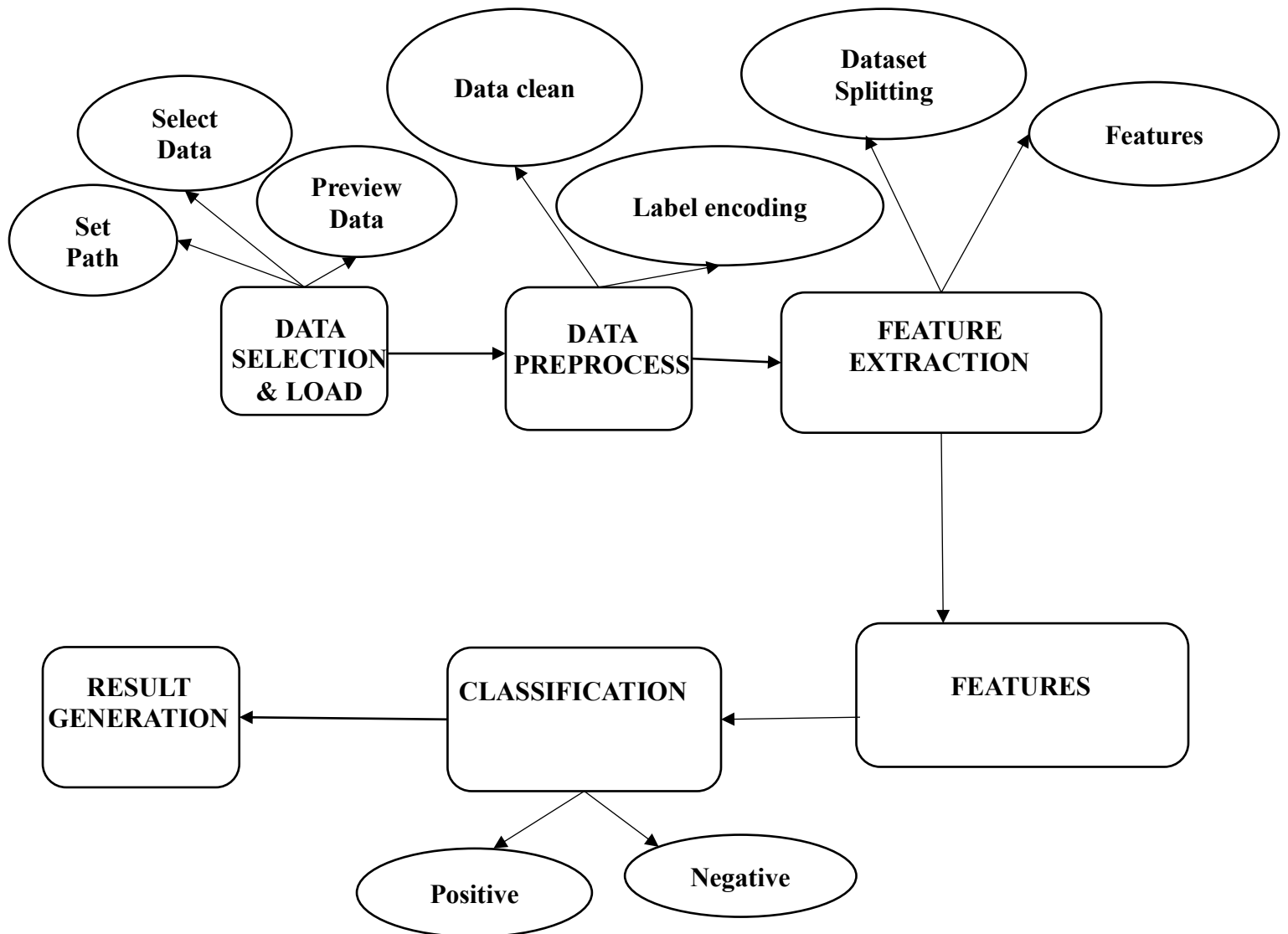
CLASS DIAGRAM



SEQUENCE DIAGRAM



ER DIAGRAM



4. IMPLEMENTATION

4.1 MODULES

- Data Selection and Loading
- Data Preprocessing
- Exploratory Data Analysis
- Splitting Dataset into Train and Test Data
- Feature Extraction
- Classification
- Result Generation

4.2 MODULE DESCRIPTION

DATA SELECTION AND LOADING

The data selection is the process of selecting the data for detecting the negative tweets from the twitter dataset. The dataset which contains the information about the UserName, ScreenName, Location, TweetAt, OriginalTweet, Sentiment. COVIDSENTI contains two months' worth of tweets. Our search was limited to tweets in English. The keywords used ensured a textual corpus focused on the COVID-19 and associated phenomena. Data collection was performed to collect all tweets from a given region, but this time separated by individual dates. This phase was separated from the month-wise collection because month-wise analysis of tweets required fast data collection to enable easy and rapid verification of ideas, while day-wise analysis required more data points and computation-intensive graphical calculations to be done on large data points.

DATA PREPROCESSING

Data pre-processing is the process of removing the unwanted data from the dataset.

Almost every social media platform uses hashtags to represent topics, i.e., #COVID-19, #StayHome, #StaySafe, and #Coronavirus. In most cases, hashtags are unnecessary to sentiment and can affect the performance. Thus, in our first step, we performed basic cleaning of the text by removing unnecessary hashtags, just the hashtag character not the hashtag text.

The second step is to case-fold the text. To avoid recognizing the same word as a different word due to capitalization, we fold all capitalized letters to lower case.

There are many words that are concatenated in other words, especially the words that are hashtagged such as “stayathomestaysafe” and “coronavirus,” should be “stay home stay safe” and “corona virus,” respectively. Hence, we perform word segmentation to achieve this.

Removing stop words is a popular method to reduce the noise in textual data. Removing stop words does not affect understanding a sentence's sentiment valence. For all but sentiment analysis, we removed stop words (words commonly found in a document of little semantic value, e.g., "for," "the," and "is").

The fifth step is lemmatization that processes the morphological analysis of words and returns words to the base or dictionary form. We used the method provided in nltk and lemmatized words (changing different forms of a word to its basic root form, e.g., "viruses" to "virus" or "went" to "go").

Textual data analysis was first held to eliminate hyperlinks, @mentions, and punctuation. We remove special characters, punctuation, and numbers from the data set as these do not help detect sentiment.

EXPLORATORY DATA ANALYSIS

Exploratory data analysis is an approach of analyzing data sets to summarize their main characteristics, often using statistical graphics and other data visualization methods. A statistical model can be used or not, but primarily EDA is for seeing what the data can tell us beyond the formal modeling or hypothesis testing task.

The objectives of EDA are to:

- Suggest hypotheses about the causes of observed phenomena
- Assess assumptions on which statistical inference will be based
- Support the selection of appropriate statistical tools and techniques
- Provide a basis for further data collection through surveys or experiments

SPLITTING DATASET INTO TRAIN AND TEST DATA

Data splitting is the act of partitioning available data into two portions, usually for cross-validator purposes. One portion of the data is used to develop a predictive model. And the other to evaluate the model's performance. Separating data into training and testing sets is an important part of evaluating data mining models. Typically, when you separate a data set into a training set and testing set, most of the data is used for training, and a smaller portion of the data is used for testing.

The train-test split procedure is used to estimate the performance of machine learning algorithms when they are used to make predictions on data not used to train the model.

It is a fast and easy procedure to perform, the results of which allow you to compare the performance of machine learning algorithms for your predictive modeling problem. Although simple to use and interpret, there are times when the procedure should not be used, such as when you have a small dataset and situations where additional configuration is required, such as when it is used for classification and the dataset is not balanced.

FEATURE EXTRACTION

Vectorization techniques and word embeddings are used for feature extraction. Term frequency-inverse document frequency (TF-IDF) has been used for vectorization. Similarly, for word embeddings, pretrained Word2Vec, GloVe, and fastText embeddings trained on Common Crawl and Wikipedia are used and have 300-D vectors. In addition, we used hybrid models, such as hybrid ranking (HyRank) and Improved Word Vector (IWV) that incorporate sentiment and context of tweets for Twitter sentiment analysis.

We have also used current state-of-the-art language models (LMs) that have altered NLP research's landscape. Generally, all training is performed in an unsupervised manner to generate an LM used as an input to the subsequent supervised

fine-tuning for NLP-tasks. Transformer-based models are trained on large volumes of data. This is why they can capture more context enriched word representations than traditional and noncontextual word representation methods.

CLASSIFICATION

Classification is a process related to categorization, the process in which ideas and objects are recognized, differentiated, and understood. In this project, the multinomial NB, SVM, LR and LSTM classification algorithm is used for classifying the data.

Multinomial Naïve Bayes algorithm predicts the tag of a text such as a piece of email or article. It calculates the probability of each tag for a given sample and then gives the tag with the highest probability as output. The multinomial distribution normally requires integer feature counts.

In the SVM algorithm, we plot each data item as a point in n-dimensional space (where n is number of features you have) with the value of each feature being the value of a particular coordinate. Support Vectors are simply the co-ordinates of individual observation. The SVM classifier is a frontier which best segregates the two classes (hyper-plane/ line).

Logistic regression is estimating the parameters of a logistic model (a form of binary regression). Mathematically, a binary logistic model has a dependent variable with two possible values, such as pass/fail which is represented by an indicator variable, where the two values are labeled "0" and "1".

Long short-term memory (LSTM) is an artificial recurrent neural network (RNN) architecture used in the field of deep learning. Unlike standard feedforward neural networks, LSTM has feedback connections. It can process not only single data points, but also entire sequences of data.

RESULT GENERATION

The Final Result will get generated based on the overall classification and prediction. The performance of this proposed approach is evaluated using some measures like,

Accuracy

Accuracy of classifier refers to the ability of classifier. It predicts the class label correctly and the accuracy of the predictor refers to how well a given predictor can guess the value of predicted attribute for a new data.

$$AC = (TP + TN) / (TP + TN + FP + FN)$$

Precision

Precision is defined as the number of true positives divided by the number of true positives plus the number of false positives.

$$\text{Precision} = TP / (TP + FP)$$

Recall

Recall is the number of correct results divided by the number of results that should have been returned. In binary classification, recall is called sensitivity. It can be viewed as the probability that a relevant document is retrieved by the query.

$$\text{Recall} = TP / (TP + FN)$$

F-Measure

F measure (F1 score or F score) is a measure of a test's accuracy and is defined as the weighted harmonic mean of the precision and recall of the test.

$$F\text{-measure} = 2TP / (2TP + FP + FN)$$

5. SYSTEM REQUIREMENTS

5.1 Hardware Requirements

- System : Pentium IV 2.4 GHz
- Hard Disk : 1000 GB
- Monitor : 15 VGA color
- Mouse : Logitech.
- Keyboard : 110 keys enhanced
- Ram : 4GB

5.2 Software Requirements

- O/S : Windows 7
- Language : python.
- IDE : Anaconda - Spyder

5.3 SOFTWARE DESCRIPTION

Python

Python is a general-purpose interpreted, interactive, object-oriented, and high-level programming language. It was created by Guido van Rossum during 1985- 1990. Like Perl, Python source code is also available under the GNU General Public License (GPL). This tutorial gives enough understanding on Python programming language.

Python is a popular programming language. It was created in 1991 by Guido van Rossum.

It is used for:

- web development (server-side),
- software development,
- mathematics,
- System scripting.

Python can be used on a server to create web applications. Python can be used alongside software to create workflows. Python can connect to database systems. It can also read and modify files. Python can be used to handle big data and perform complex mathematics. Python can be used for rapid prototyping, or for production-ready software development.

Python works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc). Python has a simple syntax similar to the English language. Python has syntax that allows developers to write programs with fewer lines than some other programming languages.

Python runs on an interpreter system, meaning that code can be executed as soon as it is written. This means that prototyping can be very quick. Python can be treated in a procedural way, an object-orientated way or a functional way.

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

- **Python is Interpreted** – Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
- **Python is Interactive** – you can actually sit at a Python prompt and interact with the interpreter directly to write your programs.
- **Python is Object-Oriented** – Python supports Object-Oriented style or technique of programming that encapsulates code within objects.
- **Python is a Beginner's Language** – Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

History of Python

Python was developed by Guido van Rossum in the late eighties and early nineties at the National Research Institute for Mathematics and Computer Science in the Netherlands.

Python is derived from many other languages, including ABC, Modula-3, C, C++, Algol-68, SmallTalk, and Unix shell and other scripting languages.

Python is copyrighted. Like Perl, Python source code is now available under the GNU General Public License (GPL).

Python is now maintained by a core development team at the institute, although Guido van Rossum still holds a vital role in directing its progress.

Python Features

Python's features include –

- **Easy-to-learn** – Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.
- **Easy-to-read** – Python code is more clearly defined and visible to the eyes.
- **Easy-to-maintain** – Python's source code is fairly easy-to-maintain.
- **A broad standard library** – Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.
- **Interactive Mode** – Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.
- **Portable** – Python can run on a wide variety of hardware platforms and has the same interface on all platforms.
- **Extendable** – you can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.
- **Databases** – Python provides interfaces to all major commercial databases.
- **GUI Programming** – Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.
- **Scalable** – Python provides a better structure and support for large programs than shell scripting.

Apart from the above-mentioned features, Python has a big list of good features, few are listed below –

- It supports functional and structured programming methods as well as OOP.
- It can be used as a scripting language or can be compiled to byte-code for building large applications.
- It provides very high-level dynamic data types and supports dynamic type checking.
- It supports automatic garbage collection.
- It can be easily integrated with C, C++, COM, ActiveX, CORBA, and Java.

Python is available on a wide variety of platforms including Linux and Mac OS X.

Python Syntax compared to other programming languages

- Python was designed to for readability, and has some similarities to the English language with influence from mathematics.
- Python uses new lines to complete a command, as opposed to other programming languages which often use semicolons or parentheses.
- Python relies on indentation, using whitespace, to define scope; such as the scope of loops, functions and classes. Other programming languages often use curly-brackets for this purpose.

Anaconda

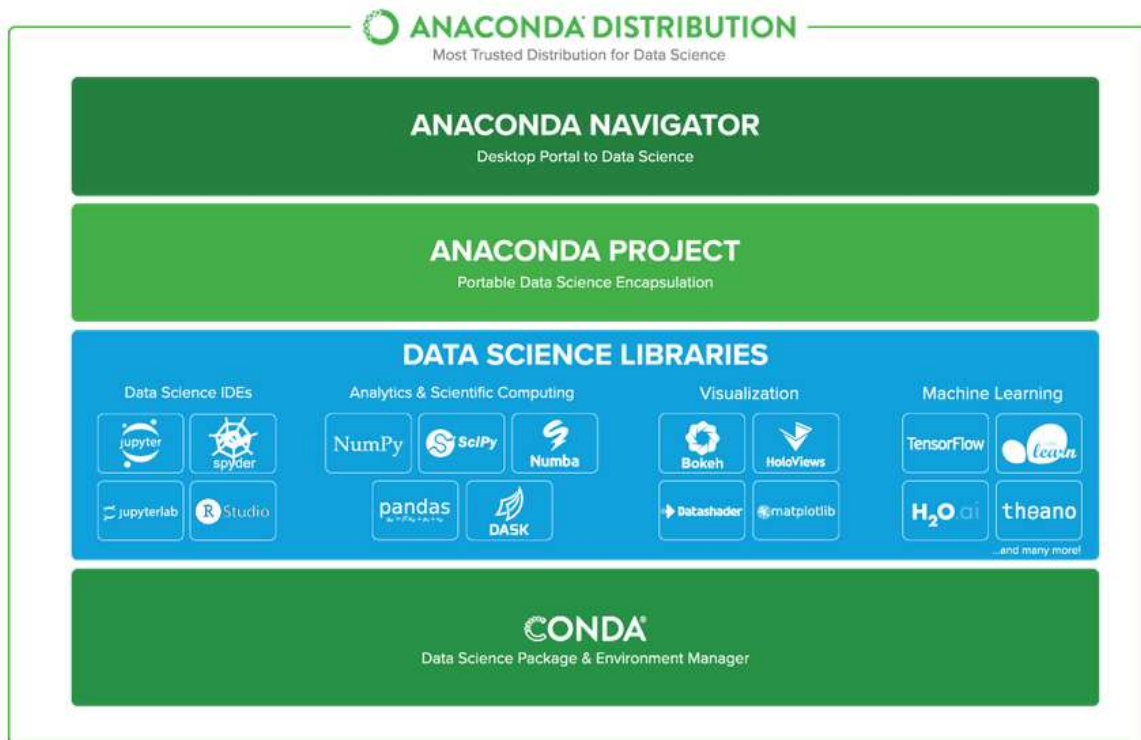
Anaconda is the most popular python data science platform.

Anaconda Distribution

With over 6 million users, the open source Anaconda Distribution is the fastest and easiest way to do Python and R data science and machine learning on Linux, Windows, and Mac OS X. It's the industry standard for developing, testing, and training on a single machine.

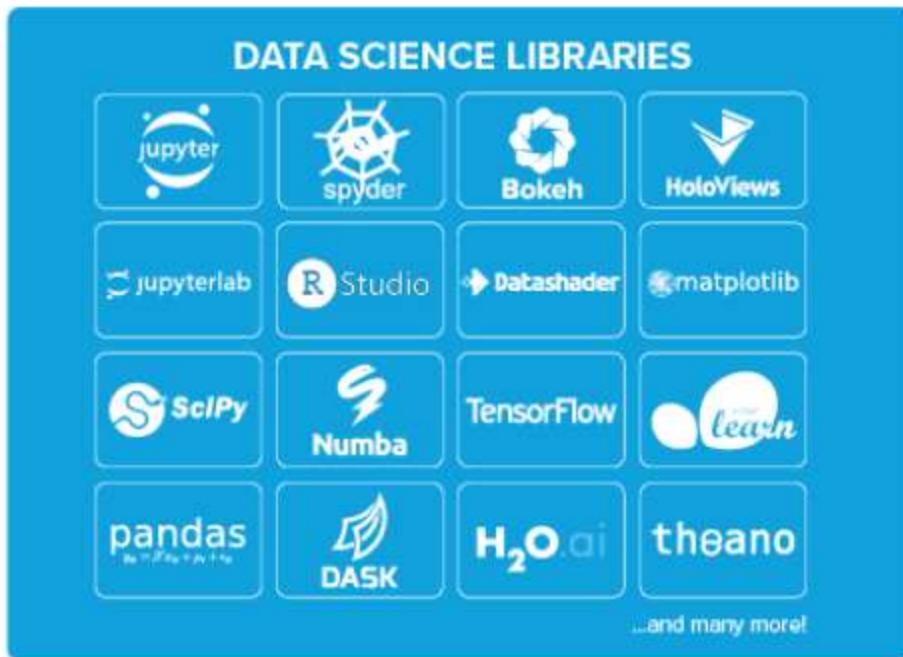
Anaconda Enterprise

Anaconda Enterprise is an AI/ML enablement platform that empowers organizations to develop, govern, and automate AI/ML and data science from laptop through training to production. It lets organizations scale from individual data scientists to collaborative teams of thousands, and to go from a single server to thousands of nodes for model training and deployment.



Anaconda Data Science Libraries

- Over 1,400 Anaconda-curated and community data science packages
- Develop data science projects using your favourite IDEs, including Jupyter, JupyterLab, Spyder, and RStudio
- Analyse data with scalability and performance with Dask, numpy, pandas, and Numba
- Visualize your data with Matplotlib, Bokeh, Datashader, and Holoviews
- Create machine learning and deep learning models with Scikit-learn, Tensorflow, h2o, and Theano



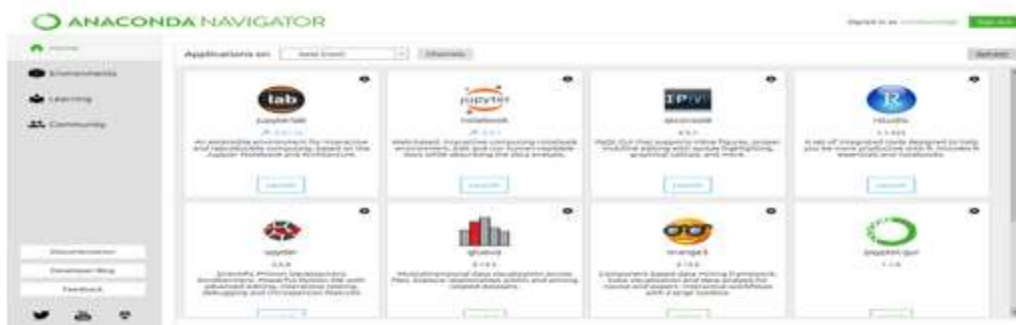
Conda, the Data Science Package & Environment Manager

- Automatically manages all packages, including cross-language dependencies
- Works across all platforms: Linux, macOS, Windows
- Create virtual environments
- Download conda packages from Anaconda, Anaconda Enterprise, Conda Forge, and Anaconda Cloud



Anaconda Navigator, the Desktop Portal to Data Science

- Install and launch applications and editors including Jupyter, RStudio, Visual Studio Code, and Spyder
- Manage your local environments and data science projects from a graphical interface
- Connect to Anaconda Cloud or Anaconda Enterprise
- Access the latest learning and community resources



Spyder

Spyder is an open source cross-platform integrated development environment (IDE) for scientific programming in the Python language. ... Initially created and developed by Pierre Raybaut in 2009, since 2012 Spyder has been maintained and continuously improved by a team of scientific Python developers and the community. Strongly recommend the free, open-source Spyder Integrated Development Environment (IDE) for scientific and engineering programming, due to its integrated editor, interpreter console, and debugging tools. Spyder is included in Anaconda and other distributions.

Spyder is a powerful scientific environment written in Python, for Python, and designed by and for scientists, engineers and data analysts. It offers a unique combination of the advanced editing, analysis, debugging, and profiling functionality of a comprehensive development tool with the data exploration, interactive execution, deep inspection, and beautiful visualization capabilities of a scientific package.

Beyond its many built-in features, its abilities can be extended even further via its plugin system and API. Furthermore, Spyder can also be used as a PyQt5 extension library, allowing developers to build upon its functionality and embed its components, such as the interactive console, in their own PyQt software.

Editor

Work efficiently in a multi-language editor with a function/class browser, code analysis tools, automatic code completion, horizontal/vertical splitting, and go-to-definition.

IPython Console

Harness the power of as many IPython consoles as you like within the flexibility of a full GUI interface; run your code by line, cell, or file; and render plots right inline.

Variable Explorer

Interact with and modify variables on the fly: plot a histogram or time series, edit a date frame or Numpy array, sort a collection, dig into nested objects, and more!

Profiler

Find and eliminate bottlenecks to unchain your code's performance.

Debugger

Trace each step of your code's execution interactively.

Help

Instantly view any object's docs, and render your own.

FEASIBILITY STUDY

The feasibility study is carried out to test whether the proposed system is worth being implemented. The proposed system will be selected if it is best enough in meeting the performance requirements.

The feasibility carried out mainly in three sections namely.

- Economic Feasibility
- Technical Feasibility
- Behavioural Feasibility

Economic Feasibility

Economic analysis is the most frequently used method for evaluating effectiveness of the proposed system. More commonly known as cost benefit analysis. This procedure determines the benefits and saving that are expected from the system of the proposed system. The hardware in system department is sufficient for system development.

Technical Feasibility

This study centre around the system's department hardware, software and to what extend it can support the proposed system department is having the required hardware and software there is no question of increasing the cost of implementing the proposed system. The criteria, the proposed system is technically feasible and the proposed system can be developed with the existing facility.

Behavioural Feasibility

People are inherently resistant to change and need sufficient amount of training, which would result in lot of expenditure for the organization. The proposed system can generate reports with day-to-day information immediately at the user's request, instead of getting a report, which doesn't contain much detail.

5.4 TESTING OF PRODUCT

Testing of Product

System testing is the stage of implementation, which aimed at ensuring that system works accurately and efficiently before the live operation commence. Testing is the process of executing a program with the intent of finding an error. A good test case is one that has a high probability of finding an error. A successful test is one that answers a yet undiscovered error.

Testing is vital to the success of the system. System testing makes a logical assumption that if all parts of the system are correct, the goal will be successfully achieved. The candidate system is subject to variety of tests-on-line response, Volume Street, recovery and security and usability test. A series of tests are performed before the system is ready for the user acceptance testing. Any engineered product can be tested in one of the following ways. Knowing the specified function that a product has been designed to from, test can be conducted to demonstrate each function is fully operational. Knowing the internal working of a product, tests can be conducted to ensure that "al gears mesh", that is the internal operation of the product performs

according to the specification and all internal components have been adequately exercised.

UNIT TESTING

Unit testing is the testing of each module and the integration of the overall system is done. Unit testing becomes verification efforts on the smallest unit of software design in the module. This is also known as 'module testing'. The modules of the system are tested separately. This testing is carried out during the programming itself. In this testing step, each model is found to be working satisfactorily as regard to the expected output from the module. There are some validation checks for the fields. For example, the validation check is done for verifying the data given by the user where both format and validity of the data entered is included. It is very easy to find error and debug the system.

INTEGRATION TESTING

Data can be lost across an interface, one module can have an adverse effect on the other sub function, when combined, may not produce the desired major function. Integrated testing is systematic testing that can be done with sample data. The need for the integrated test is to find the overall system performance. There are two types of integration testing. They are:

- i) Top-down integration testing.
- ii) Bottom-up integration testing.

WHITE BOX TESTING

White Box testing is a test case design method that uses the control structure of the procedural design to drive cases. Using the white box testing methods, we derived test cases that guarantee that all independent paths within a module have been exercised at least once.

BLACK BOX TESTING

- ✓ Black box testing is done to find incorrect or missing function
- ✓ Interface error
- ✓ Errors in external database access
- ✓ Performance errors
- ✓ Initialization and termination errors

In 'functional testing', is performed to validate an application conforms to its specifications of correctly performs all its required functions. So this testing is also called 'black box testing'. It tests the external behaviour of the system. Here the engineered product can be tested knowing the specified function that a product has been designed to perform, tests can be conducted to demonstrate that each function is fully operational.

VALIDATION TESTING

After the culmination of black box testing, software is completed assembly as a package, interfacing errors have been uncovered and corrected and final series of software validation tests begin validation testing can be defined as many, but a single definition is that validation succeeds when the software functions in a manner that can be reasonably expected by the customer.

USER ACCEPTANCE TESTING

User acceptance of the system is the key factor for the success of the system. The system under consideration is tested for user acceptance by constantly keeping in touch with prospective system at the time of developing changes whenever required.

OUTPUT TESTING

After performing the validation testing, the next step is output asking the user about the format required testing of the proposed system, since no system could be useful if it does not produce the required output in the specific format. The output displayed or generated by the system under consideration. Here the output format is considered in two ways. One is screen and the other is printed format. The output format on the screen is found to be correct as the format was designed in the system phase according to the user needs. For the hard copy also output comes out as the specified requirements by the user. Hence the output testing does not result in any connection in the system.

Agile Testing

Agile Testing is a type of software testing that accommodates agile software development approach and practices. In an Agile development environment, testing is an integral part of software development and is done along with coding. Agile testing allows incremental and iterative coding and testing.

API Testing

API testing is a type of testing that is similar to unit testing. Each of the Software APIs are tested as per API specification. API testing is mostly done by testing team unless APIs to be tested are complex and need extensive coding. API testing requires understanding both API functionality and possessing good coding skills.

Automated testing

This is a testing approach that makes use of testing tools and/or programming to run the test cases using software or custom developed test utilities. Most of the automated tools provide capture and playback facility, however there are tools that require writing extensive scripting or programming to automate test cases.

End-to-end Testing

End to end testing is performed by testing team, focus of end to end testing is to test end to end flows e.g. right from order creation till reporting or order creation till item return etc. and checking. End to end testing is usually focused mimicking real life scenarios and usage. End to end testing involves testing information flow across applications.

Exploratory Testing

Exploratory testing is an informal type of testing conducted to learn the software at the same time looking for errors or application behaviour that seems non-obvious. Exploratory testing is usually done by testers but can be done by other stake holders as well like Business Analysts, developers, end users etc. who are interested in learning functions of the software and at the same time looking for errors or behaviour is seems non-obvious.

Performance Testing

It is a type of software testing and part of performance engineering that is performed to check some of the quality attributes of software like Stability, reliability, availability. Performance testing is carried out by performance engineering team. Unlike Functional testing, Performance testing is done to check non-functional requirements. Performance testing checks how well software works in anticipated and peak workloads. There are different variations or sub types of performance like load testing, stress testing, volume testing, soak testing and configuration testing.

Penetration Testing

It is a type of security testing, also known as pen test in short. Penetration testing is done to tests how secure software and its environments (Hardware, Operating system and network) are when subject to attack by an external or internal intruder. Intruder can be a human/hacker or malicious programs. Pen test uses methods to

forcibly intrude (by brute force attack) or by using a weakness (vulnerability) to gain access to a software or data or hardware with an intent to expose ways to steal, manipulate or corrupt data, software files or configuration. Penetration Testing is a way of ethical hacking, an experienced Penetration tester will use the same methods and tools that a hacker would use but the intention of Penetration tester is to identify vulnerability and get them fixed before a real hacker or malicious program exploits it.

Security Testing

It is a type of software testing carried out by specialized team of software testers. Objective of security testing is to secure the software is to external or internal threats from humans and malicious programs. Security testing basically checks, how good is software's authorization mechanism, how strong is authentication, how software maintains confidentiality of the data, how does the software maintain integrity of the data, what is the availability of the software in an event of an attack on the software by hackers and malicious programs is for Security testing requires good knowledge of application, technology, networking, security testing tools. With increasing number of web applications necessarily of security testing has increased to a greater extent.

Sanity Testing

It is a type of testing that is carried out mostly by testers and in some projects by developers as well. Sanity testing is a quick evaluation of the software, environment, network, external systems are up & running, software environment as a whole is stable enough to proceed with extensive testing. Sanity tests are narrow and most of the time sanity tests are not documented.

Scalability Testing

It is a non-functional test intended to test one of the software quality attributes i.e. "Scalability". Scalability test is not focused on just one or few functionality of the software instead performance of software as a whole. Scalability testing is usually done by performance engineering team. Objective of scalability testing is to test the ability of

the software to scale up with increased users, increased transactions, increase in database size etc., It is not necessary that software's performance increases with increase in hardware configuration, scalability tests helps to find out how much more workload the software can support with expanding user base, transactions, data storage etc.,

Stability Testing

It is a non-functional test intended to test one of the software quality attributes i.e. "Stability". Stability testing focuses on testing how stable software is when it is subject to loads at acceptable levels, peak loads, loads generated in spikes, with more volumes of data to be processed. Scalability testing will involve performing different types of performance tests like load testing, stress testing, spike testing, soak testing, spike testing etc....

Static Testing is a form of testing where in approaches like reviews, walkthroughs are employed to evaluate the correctness of the deliverable. In static testing software code is not executed instead it is reviewed for syntax, commenting, naming convention, size of the functions and methods etc. Static testing usually has check lists against which deliverables are evaluated. Static testing can be applied for requirements, designs, and test cases by using approaches like reviews or walkthroughs.

Stress Testing is a type of performance testing, in which software is subjected to peak loads and even to a break point to observe how the software would behave at breakpoint. Stress testing also tests the behaviour of the software with insufficient resources like CPU, Memory, Network bandwidth, Disk space etc. Stress testing enables to check some of the quality attributes like robustness and reliability.

6. CONCLUSION AND FUTURE ENHANCEMENT

6.1 CONCLUSION

From the analysis, it is observed that people were mostly expressing their thoughts with positive sentiments. Since the explosion of COVID-19 conspiracy theories, social media has been widely used both for and against misinformation and misconceptions. We address the issue of Twitter sentiment on COVID-19-related Twitter posts. We benchmark sentiment analysis methods in the analysis of COVID-19-related sentiment. This analysis was unique in its way as Lockdown shows peaks and were positive. Most people were criticizing Lockdown on the face but twitter was full of positive tweets. This analysis can be further taken to new possibilities of Emotion analysis. Rather than having Positive, Negative, and Neutral tweets, we can analyze based on emotions. Text can also represent the emotions of a person writing. Tweets can have emotions like Hate, Respect, Agreement, Anger, Happiness. Each tweet can have multiple emotions and we can have the emotion having a maximum score. In these, people can have multiple emotions and several amazing insights can be generated.

6.2 FUTURE ENHANCEMENT

In the future, it holds the potential to discover the mindset of people. In this work, we have only used tweets in English language. Tweets in other Indian languages if collected will have a better representation of people's sentiments. In near future systems are moving towards Analytics and Data to find the Prediction through machine learning or deep learning.

7. SAMPLE CODING AND SAMPLE SCREENSHOT

CODING

```
import tensorflow as tf
import os
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.preprocessing.sequence import pad_sequences
import tensorflow.keras.layers as Layers
#from tensorflow.keras.losses import SparseCategoricalCrossentropy
from tensorflow.keras.optimizers import Adam
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix
from sklearn.metrics import classification_report
import matplotlib.pyplot as plt
import plotly.graph_objects as go
import plotly.express as px
import plotly.figure_factory as ff
import seaborn as sns
import numpy as np
import pandas as pd
import nltk
from nltk.stem.porter import PorterStemmer
from nltk.tokenize import TweetTokenizer
from nltk.tokenize import word_tokenize
from nltk.corpus import stopwords
import re
from sklearn.feature_extraction.text import CountVectorizer, TfidfVectorizer
from sklearn.model_selection import train_test_split, cross_val_score
from sklearn.model_selection import GridSearchCV
from sklearn.naive_bayes import MultinomialNB
from sklearn.linear_model import LogisticRegression
```

```

from tensorflow.keras import models
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Embedding, LSTM, Flatten, GlobalMaxPooling1D,
Dense, Dropout, Bidirectional

```

```

train = pd.read_csv('Corona_NLP_train.csv',encoding='latin1')
test = pd.read_csv('Corona_NLP_test.csv',encoding='latin1')

```

```

train['OriginalTweet']=train['OriginalTweet'].astype(str)
train['Sentiment']=train['Sentiment'].astype(str)

```

```

test['OriginalTweet']=test['OriginalTweet'].astype(str)
test['Sentiment']=test['Sentiment'].astype(str)
print(train.head())
print(test.head())
print(train.shape)
print(test.shape)
print(train.describe())
print(test.describe())
train.info()
test.info()
print(train.isna().sum())
print(test.isna().sum())

```

```

#Data Analysis

```

```

dist_train = train['Sentiment'].value_counts()
dist_test = test['Sentiment'].value_counts()

```

```

def ditribution_plot(x,y,name):

```

```
fig = go.Figure([
    go.Bar(x=x, y=y)
])
```

```
fig.update_layout(title_text=name)
fig.show()
```

```
distribution_plot(x= dist_train.index, y= dist_train.values, name= 'Class Distribution train')
distribution_plot(x= dist_test.index, y= dist_test.values, name= 'Class Distribution test')
```

```
df=pd.concat([train,test])
df['OriginalTweet']=df['OriginalTweet'].astype(str)
df['Sentiment']=df['Sentiment'].astype(str)
```

```
df['text'] = df.OriginalTweet
df["text"] = df["text"].astype(str)
```

```
train['text'] = train.OriginalTweet
train["text"] = train["text"].astype(str)
```

```
test['text'] = test.OriginalTweet
test["text"] = test["text"].astype(str)
```

```
# Data has 5 classes, let's convert them to 3
```

```
def classes_def(x):
    if x == "Extremely Positive":
        return "positive"
    elif x == "Extremely Negative":
        return "negative"
    elif x == "Negative":
```

```

        return "negative"
    elif x == "Positive":
        return "positive"
    else:
        return "neutral"

```

```

df['sentiment']=df['Sentiment'].apply(lambda x:classes_def(x))
train['sentiment']=train['Sentiment'].apply(lambda x:classes_def(x))
test['sentiment']=test['Sentiment'].apply(lambda x:classes_def(x))
target=df['sentiment']

```

```

df.sentiment.value_counts(normalize= True)

```

```

class_df =
df.groupby('Sentiment').count()['text'].reset_index().sort_values(by='Sentiment',ascending=False)
class_df.style.background_gradient(cmap='winter')
print(class_df)

```

```

from plotly.subplots import make_subplots
fig=make_subplots(1,2,subplot_titles=('Train set','Test set'))
x=train.Sentiment.value_counts()
fig.add_trace(go.Bar(x=x.index,y=x.values,marker_color=['#17C37B','#F92969','#FACA0C'],name='train'),row=1,col=1)
x=test.Sentiment.value_counts()
fig.add_trace(go.Bar(x=x.index,y=x.values,marker_color=['#17C37B','#F92969','#FACA0C'],name='test'),row=1,col=2)

```

```

#words in tweets
fig,(ax1,ax2,ax3)=plt.subplots(1,3,figsize=(15,5))

```

```
tweet_len=train[train['sentiment']=="positive"]['text'].str.split().map(lambda x: len(x))
ax1.hist(tweet_len,color='#17C37B')
ax1.set_title('Positive Sentiments')
```

```
tweet_len=train[train['sentiment']=="negative"]['text'].str.split().map(lambda x: len(x))
ax2.hist(tweet_len,color='#F92969')
ax2.set_title('Negative Sentiments')
```

```
tweet_len=train[train['sentiment']=="neutral"]['text'].str.split().map(lambda x: len(x))
ax3.hist(tweet_len,color='#FACA0C')
ax3.set_title('Neutral Sentiments')
```

```
fig.suptitle('Words in a tweet')
plt.savefig('images/words_in_tweet.png')
plt.show()
```

```
plt.figure()
#characters in tweets
fig,(ax1,ax2,ax3)=plt.subplots(1,3,figsize=(15,5))
```

```
tweet_len=train[train['sentiment']=="positive"]['text'].str.len()
ax1.hist(tweet_len,color='#17C37B')
ax1.set_title('Positive Sentiments')
```

```
tweet_len=train[train['sentiment']=="negative"]['text'].str.len()
ax2.hist(tweet_len,color='#F92969')
ax2.set_title('Negative Sentiments')
```

```
tweet_len=train[train['sentiment']=="neutral"]['text'].str.len()
ax3.hist(tweet_len,color='#FACA0C')
ax3.set_title('Neutral Sentiments')
```

```
fig.suptitle('Characters in tweets')
plt.savefig('images/characters_in_tweet.png')
plt.show()
```

```
plt.figure()
#Hashtags
plt.figure(figsize = (14,8))
def find_hash(text):
    line=re.findall(r'(?<=#)\w+',text)
    return " ".join(line)
df['hash']=df['text'].apply(lambda x:find_hash(x))
temp=df['hash'].value_counts()[1:11]
temp= temp.to_frame().reset_index().rename(columns={'index':'Hashtag','hash':'count'})
sns.barplot(x="Hashtag",y="count", data = temp)
plt.savefig('images/hashtags.png')
plt.show()
```

```
plt.figure()
#Mentions
def mentions(text):
    line=re.findall(r'(?<=@)\w+',text)
    return " ".join(line)
df['mentions']=df['text'].apply(lambda x:mentions(x))

temp=df['mentions'].value_counts()[1:11]
temp
=temp.to_frame().reset_index().rename(columns={'index':'Mentions','mentions':'count'})
plt.figure(figsize = (14,8))
sns.barplot(x="Mentions",y="count", data = temp)
plt.savefig('images/mentions.png')
```

```

plt.show()

train['Sentiment']
train['OriginalTweet']

#Data processing
nltk.download('words')
nltk.download('punkt')
nltk.download('maxent_ne_chunker')
nltk.download('stopwords')
stop_words = nltk.corpus.stopwords.words('english')

train['OriginalTweet'][0]
X = train['OriginalTweet'].copy()
y = train['Sentiment'].copy()

def data_cleaner(tweet):

    # remove urls
    tweet = re.sub(r'http\S+', ' ', tweet)

    # remove html tags
    tweet = re.sub(r'<.*?>', ' ', tweet)

    # remove digits
    tweet = re.sub(r'\d+', ' ', tweet)

    # remove hashtags
    tweet = re.sub(r'#\w+', ' ', tweet)

    # remove mentions

```

```
tweet = re.sub(r'@\w+', ' ', tweet)
```

```
#removing stop words
```

```
tweet = tweet.split()
```

```
tweet = " ".join([word for word in tweet if not word in stop_words])
```

```
return tweet
```

```
X_cleaned = X.apply(data_cleaner)
```

```
X_cleaned.head()
```

#In Python tokenization basically refers to splitting up a larger body of text into smaller lines, words or even creating words for a non-English language.

#The various tokenization functions in-built into the nltk module itself and can be used in programs as shown below.

```
tokenizer = Tokenizer()
```

```
tokenizer.fit_on_texts(X_cleaned)
```

```
X = tokenizer.texts_to_sequences(X_cleaned)
```

```
vocab_size = len(tokenizer.word_index)+1
```

```
print("Vocabulary size: {}".format(vocab_size))
```

```
print("\nExample:\n")
```

```
print("Sentence:\n{}".format(X_cleaned[1]))
```

```
print("\nAfter tokenizing :\n{}".format(X[1]))
```

```
# adding zeros to have all data with same dimension
```

```
X = pad_sequences(X, padding='post')
```

```
print("\nAfter padding :\n{}".format(X[1]))
```

```
encoding = {'Extremely Negative': 0,
            'Negative': 0,
            'Neutral': 1,
            'Positive': 2,
            'Extremely Positive': 2
            }
```

```
labels = ['Negative', 'Neutral', 'Positive']
```

```
word_index = tokenizer.word_index
print('Found %s unique tokens.' % len(word_index))
```

```
y.replace(encoding, inplace=True)
```

```
y.isnull().sum()
```

```
# Create feature vectors
```

```
vectorizer = TfidfVectorizer(min_df = 5, max_df = 0.8, sublinear_tf = True, use_idf =
True)
```

```
vectors = vectorizer.fit_transform(X_cleaned)
```

```
#Split train and test
```

```
train_vectors = vectors[int(vectors.shape[0]*0.8):]
```

```
test_vectors = vectors[:int(vectors.shape[0]*0.8)]
```

```
y_train = y[int(vectors.shape[0]*0.8):]
```

```
y_test = y[:int(vectors.shape[0]*0.8)]
```

SCREENSHOTS:

Index	Username	ScreenName	Location	Timestamp	OriginalTweet	Sentiment	test	sentiment	hash	mentions
0	3795	48731	London	10-03-2020	@heyryrie @Phil_Baker -	Neutral	@heyryrie @Phil_Baker -	neutral		@heyryrie @Phil_Baker C.
1	3888	48732	UK	10-03-2020	advise Tack to your neigh.	Positive	advise Tack to your neigh.	positive		
2	3881	48733	Vagabonds	10-03-2020	Coronavirus Australia: it.	Positive	Coronavirus Australia: it.	positive		
3	3882	48734	nan	10-03-2020	My food stock is not the s.	Positive	My food stock is not the s.	positive	COVID19france COVID19 COV.	
4	3883	48735	nan	10-03-2020	He, ready to go at super.	Negative	He, ready to go at super.	negative	coronavirus -	
5	3884	48736	Auf: 36.315788,-8.	10-03-2020	As news of the region.	Positive	As news of the region.	positive		Tia_Dodson
6	3885	48737	39.925541,-7.	10-03-2020	Cashier at grocery stor.	Positive	Cashier at grocery stor.	positive	Covid_19	
7	3886	48738	Austria	10-03-2020	Was at the supermarket	Neutral	Was at the supermarket	neutral	Rebel toiletpaper.	
8	3887	48739	Atlanta, GA USA	10-03-2020	Due to COVID-19 sur.	Positive	Due to COVID-19 sur.	positive		
9	3888	48740	BHARANAGAR, GJ.	10-03-2020	For corone prevention, e.	Negative	For corone prevention, e.	negative	govindia IndiaFightsC.	
10	3889	48741	Rakati, Ranika	10-03-2020	All month there hasn't.	Neutral	All month there hasn't.	neutral	media lockdown cas.	
11	3890	48742	Pitt Meadows, BC, Canada	10-03-2020	Due to the Covid-19 sit.	Extremely Positive	Due to the Covid-19 sit.	positive		
12	3891	48743	Shorningsee	10-03-2020	Shorningsee is a caring	Extremely Positive	Shorningsee is a caring	positive	shorningsee	
13	3892	48744	Chicago, IL	10-03-2020	He: I don't need to stoc.	Positive	He: I don't need to stoc.	positive	Coronavirus	
14	3893	48745	nan	10-03-2020	ADAMA Releaser COV.	Positive	ADAMA Releaser COV.	positive		
15	3894	48746	Houston, Texas	10-03-2020	Lines at the grocery stor.	Positive	Lines at the grocery stor.	positive	coronavirus covid19	
16	3895	48747	Saudi Arabia	10-03-2020	????? ????? ????? ?????	Neutral	????? ????? ????? ?????	neutral		
17	3896	48748	Ontario, Canada	10-03-2020	@eyewhetherc.	Neutral	@eyewhetherc.	neutral		@eyewhetherc.
18	3897	48749	North America	10-03-2020	Amazon glitch Styels whel.	Extremely Positive	Amazon glitch Styels whel.	positive		
19	3898	48750	Denver, CO	10-03-2020	For those who aren't strug.	Positive	For those who aren't strug.	positive		
20	3899	48751	southampton xxxx xxx	10-03-2020	with 100 nations infi.	Extremely Negative	with 100 nations infi.	negative		

Index	Type	Size	Value
0	str	1	Negative
1	str	1	Neutral
2	str	1	Positive

stop_words - List (179 elements)			
Index	Type	Size	Value
0	str	1	i
1	str	1	me
2	str	1	my
3	str	1	myself
4	str	1	we
5	str	1	our
6	str	1	ours
7	str	1	ourselves
8	str	1	you
9	str	1	you're
10	str	1	you've

Save and Close

Close

target - Series		
Index	sentiment	
0	neutral	
1	positive	
2	positive	
3	positive	
4	negative	
5	positive	
6	positive	
7	neutral	
8	positive	
9	negative	
10	neutral	
11	positive	
12	positive	
13	positive	

Format
Resize
☒ Background color
☒ Column min/max
Save and Close
Close

train - DataFrame					
Index	UserName	ScreenName	Location	TweetAt	Original
0	3799	48751	London	16-03-2020	@MeNyrb: @Phil_Ge
1	3800	48752	UK	16-03-2020	advice to your
2	3801	48753	Vagabonds	16-03-2020	Coronav: Austral:
3	3802	48754	nan	16-03-2020	My food is not t
4	3803	48755	nan	16-03-2020	Me, reac go at su
5	3804	48756	ÅT: 36.319708,-8...	16-03-2020	As news the reg:
6	3805	48757	35.926541,-7...	16-03-2020	Cashier grocery
7	3806	48758	Austria	16-03-2020	Was at t supermar
8	3807	48759	Atlanta, GA USA	16-03-2020	Due to COVID-19
9	3808	48760	BHAVNAGAR,GU...	16-03-2020	For conc prevent:
10	3809	48761	Makati, Manila	16-03-2020	All moni there ha
11	3810	48762	Pitt Meadows, BC, Canada	16-03-2020	Due to t Covid-19
12	3811	48763	Horningsea	16-03-2020	#horning is a car
13	3812	48764	Chicago, IL	16-03-2020	Me: I do

Format
Resize
☒ Background color
☒ Column min/max
Save and Close
Close

test - DataFrame					
Index	UserName	ScreenName	Location	TweetAt	Original
0	1	44953	NYC	02-03-2020	TRENDING Yorkers
1	2	44954	Seattle, WA	02-03-2020	When I couldn't
2	3	44955	nan	02-03-2020	Find out you can
3	4	44956	Chicagoland	02-03-2020	#Panic b hits #Ne
4	5	44957	Melbourne, Victoria	03-03-2020	#toilet #dunnyp
5	6	44958	Los Angeles	03-03-2020	Do you remember
6	7	44959	nan	03-03-2020	Voting age of 1
7	8	44960	Geneva, Switzerland	03-03-2020	@DrTedro can't s
8	9	44961	nan	04-03-2020	HI TWIT am a pha
9	10	44962	Dublin, Ireland	04-03-2020	Anyone b in a sup
10	11	44963	Boksburg, South Africa	04-03-2020	Best qua couches
11	12	44964	New Delhi	04-03-2020	Beware o counter
12	13	44965	USA, PA	04-03-2020	Panic fo buying :
13	14	44966	nan	04-03-2020	#Covid_

Format
Resize
☒ Background color
☒ Column min/max
Save and Close
Close

tweet_len - Series		
Index	text	
0	111	
7	119	
10	276	
16	204	
17	209	
21	176	
32	145	
35	122	
43	202	
44	90	
49	239	
51	175	
59	79	
61	159	

Format
Resize
☒ Background color
☒ Column min/max
Save and Close
Close

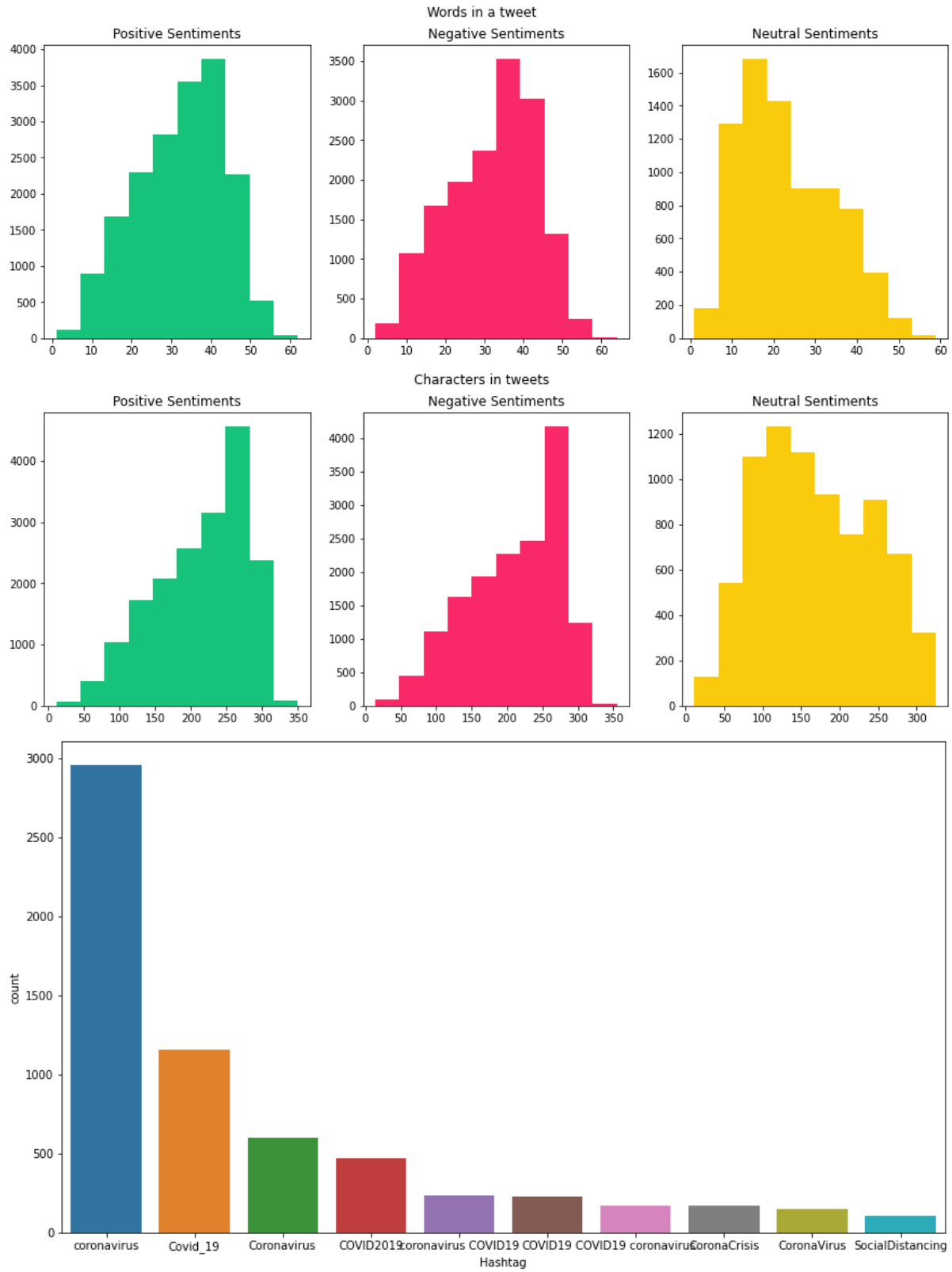
X - NumPy array						
	0	1	2	3	4	
0	0	0	0	0	0	
1	440	820	2520	201	2636	
2	26	674	4360	247	259	
3	100	5	31	33	152	
4	235	685	23	6	66	
5	103	122	13608	114	1118	
6	1174	7	4	1269	761	
7	1302	6	50	4208	52	
8	34	1	69	4	11239	
9	134	356	1967	25	70	
10	118	372	3809	179	408	
11	34	1	268	234	24	

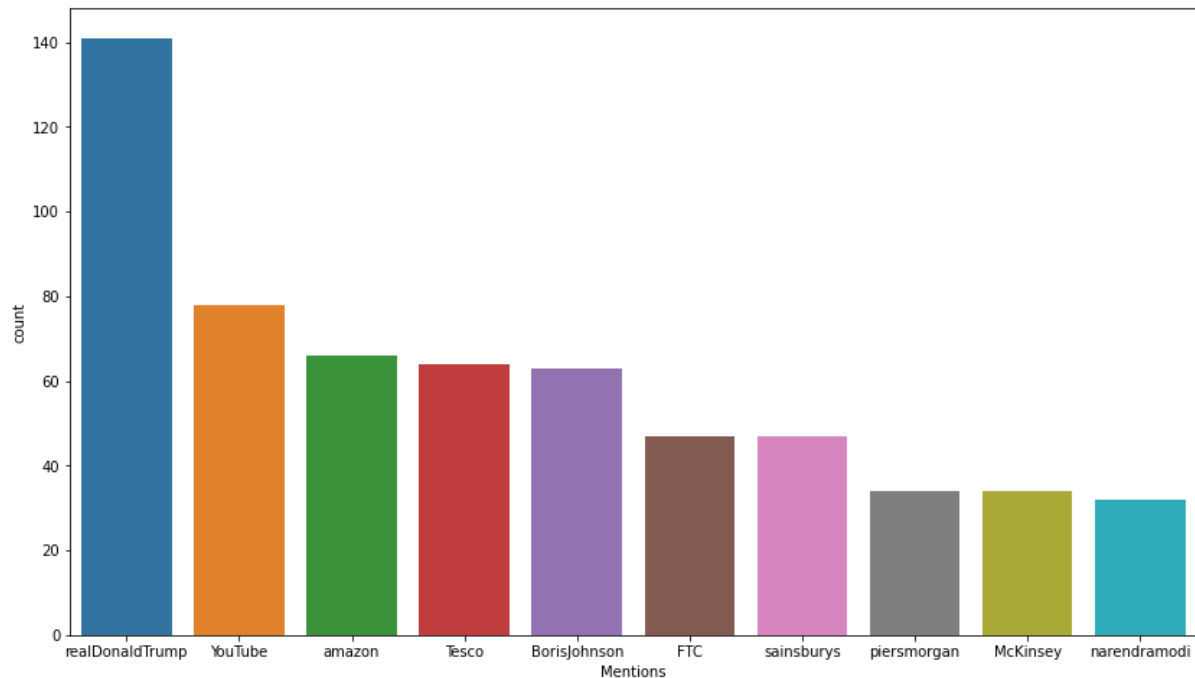
☒ Background color

y - Series		
Index	Sentiment	
0	1	
1	2	
2	2	
3	2	
4	0	
5	2	
6	2	
7	1	
8	2	
9	0	
10	1	
11	2	
12	2	
13	2	

☒ Background color
 ☒ Column min/max

	Sentiment	text
4	Positive	12369
3	Neutral	8332
2	Negative	10958
1	Extremely Positive	7223
0	Extremely Negative	6073





```
[nltk_data] Downloading package words to
[nltk_data]   C:\Users\EGC\AppData\Roaming\nltk_data...
[nltk_data]   Package words is already up-to-date!
[nltk_data] Downloading package punkt to
[nltk_data]   C:\Users\EGC\AppData\Roaming\nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package maxent_ne_chunker to
[nltk_data]   C:\Users\EGC\AppData\Roaming\nltk_data...
[nltk_data]   Package maxent_ne_chunker is already up-to-date!
[nltk_data] Downloading package stopwords to
[nltk_data]   C:\Users\EGC\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
Vocabulary size: 36117
```

Example:

Sentence:

advice Talk neighbours family exchange phone numbers create contact list phone numbers
neighbours schools employer chemist GP set online shopping accounts poss adequate supplies
regular meds order

After tokenizing :

```
[440, 820, 2520, 201, 2636, 802, 1181, 1029, 363, 320, 802, 1181, 2520, 1163, 3517, 3254,
4534, 420, 13, 12, 2257, 18314, 3046, 120, 989, 2850, 166]
```

After padding :

```
[ 440  820 2520  201 2636  802 1181 1029  363  320  802 1181
 2520 1163 3517 3254 4534  420  13  12 2257 18314 3046 120
 989 2850 166  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0]
```

Found 36116 unique tokens.

y_pred - NumPy array	
	0
0	1
1	1
2	1
3	0
4	0
5	2
6	2
7	1
8	2
9	0
10	1
11	2
12	2

Format

Resize

☒ Background color

Save and Close

Close

```

-----Multinomial NB-----
      precision    recall  f1-score   support

 Negative      0.63      0.61      0.62     12626
  Neutral      0.61      0.20      0.30      6074
 Positive      0.58      0.77      0.67     14225

 accuracy              0.60     32925
 macro avg      0.61      0.53      0.53     32925
 weighted avg   0.61      0.60      0.58     32925

```

```

-----Support Vector Machine-----
      precision    recall  f1-score   support

 Negative      0.72      0.73      0.72     12626
  Neutral      0.60      0.49      0.54      6074
 Positive      0.73      0.79      0.76     14225

 accuracy              0.71     32925
 macro avg      0.69      0.67      0.67     32925
 weighted avg   0.70      0.71      0.71     32925

```

```

-----Logistic Regression-----
      precision    recall  f1-score   support

 Negative      0.78      0.73      0.75     12626
  Neutral      0.60      0.71      0.65      6074
 Positive      0.80      0.78      0.79     14225

 accuracy              0.75     32925
 macro avg      0.73      0.74      0.73     32925
 weighted avg   0.75      0.75      0.75     32925

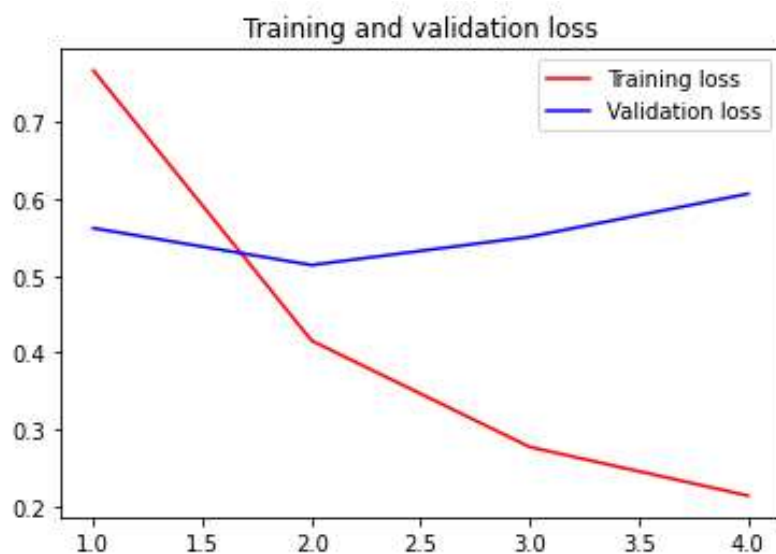
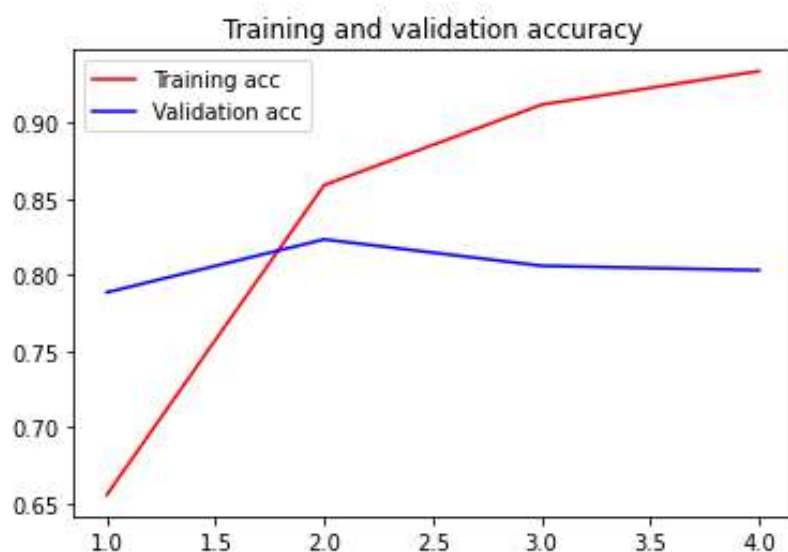
```

Layer (type)	Output Shape	Param #
embedding (Embedding)	(None, 54, 32)	1155744
bidirectional (Bidirectional)	(None, 54, 512)	591872
flatten (Flatten)	(None, 27648)	0
dense (Dense)	(None, 3)	82947

=====
 Total params: 1,830,563
 Trainable params: 1,830,563
 Non-trainable params: 0

Train on 32925 samples, validate on 8232 samples

Epoch 1/4
 32925/32925 [=====] - 356s 11ms/sample - loss: 0.7671 - acc: 0.6555 - val_loss: 0.5619 - val_acc: 0.7886
 Epoch 2/4
 32925/32925 [=====] - 352s 11ms/sample - loss: 0.4156 - acc: 0.8592 - val_loss: 0.5139 - val_acc: 0.8235
 Epoch 3/4
 32925/32925 [=====] - 352s 11ms/sample - loss: 0.2773 - acc: 0.9121 - val_loss: 0.5507 - val_acc: 0.8061
 Epoch 4/4
 32925/32925 [=====] - 349s 11ms/sample - loss: 0.2140 - acc: 0.9342 - val_loss: 0.6066 - val_acc: 0.8032



8. REFERENCES

- [1] C. C. Aggarwal and C. K. Reddy, Data Clustering: Algorithms and Applications. Boca Raton, FL, USA: CRC Press, 2013.
- [2] N. Ahmad and J. Siddique, "Personality assessment using Twitter tweets," *Procedia Comput. Sci.*, vol. 112, pp. 1964–1973, Sep. 2017.
- [3] T. Ahmad, A. Ramsay, and H. Ahmed, "Detecting emotions in English and Arabic tweets," *Information*, vol. 10, no. 3, p. 98, Mar. 2019.
- [4] A. Bandi and A. Fellah, "Socio-analyzer: A sentiment analysis using social media data," in *Proc. 28th Int. Conf. Softw. Eng. Data Eng.*, in *EPIc Series in Computing*, vol. 64, F. Harris, S. Dascalu, S. Sharma, and R. Wu, Eds. Amsterdam, The Netherlands: EasyChair, 2019, pp. 61–67.
- [5] F. Barbieri and H. Saggion, "Automatic detection of irony and humour in Twitter," in *Proc. ICCIC*, 2014, pp. 155–162.
- [6] R. Bhat, V. K. Singh, N. Naik, C. R. Kamath, P. Mulimani, and N. Kulkarni, "COVID 2019 outbreak: The disappointment in Indian teachers," *Asian J. Psychiatry*, vol. 50, Apr. 2020, Art. no. 102047.
- [7] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet allocation," *J. Mach. Learn. Res.*, vol. 3, pp. 993–1022, Jan. 2003.
- [8] P. Boldog, T. Tekeli, Z. Vizi, A. Dénes, F. A. Bartha, and G. Röst, "Risk assessment of novel coronavirus COVID-19 outbreaks outside China," *J. Clin. Med.*, vol. 9, no. 2, p. 571, Feb. 2020.
- [9] G. Carducci, G. Rizzo, D. Monti, E. Palumbo, and M. Morisio, "TwitPersonality: Computing personality traits from tweets using word embeddings and supervised learning," *Information*, vol. 9, no. 5, p. 127, May 2018.
- [10] X. Carreras and L. Màrquez, "Boosting trees for anti-spam email filtering," 2001, arXiv:cs/0109015. [Online]. Available: <https://arxiv.org/abs/cs/0109015>.

- [11] J. P. Carvalho, H. Rosa, G. Brogueira, and F. Batista, "MISNIS: An intelligent platform for Twitter topic mining," *Expert Syst. Appl.*, vol. 89, pp. 374–388, Dec. 2017.
- [12] B. K. Chae, "Insights from hashtag #supplychain and Twitter analytics: Considering Twitter and Twitter data for supply chain practice and research," *Int. J. Prod. Econ.*, vol. 165, pp. 247–259, Jul. 2015.
- [13] M. De Choudhury, S. Counts, and E. Horvitz, "Predicting postpartum changes in emotion and behavior via social media," in *Proc. SIGCHI Conf. Hum. Factors Comput. Syst.*, Apr. 2013, pp. 3267–3276.
- [14] A. Depoux, S. Martin, E. Karafillakis, R. Preet, A. Wilder-Smith, and H. Larson, "The pandemic of social media panic travels faster than the COVID-19 outbreak," *J. Travel Med.*, vol. 27, no. 3, Apr. 2020, Art. no. taaa031.
- [15] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Human Lang. Technol.*, vol. 1. Minneapolis, MN, USA: Association for Computational Linguistics, Jun. 2019, pp. 4171–4186.
- [16] M. E. El Zowalaty and J. D. Järhult, "From SARS to COVID-19: A previously unknown SARS-related coronavirus (SARS-CoV-2) of pandemic potential infecting humans—Call for a one health approach," *One Health*, vol. 9, Jun. 2020, Art. no. 100124.
- [17] I. Fung et al., "Pedagogical demonstration of Twitter data analysis: A case study of world AIDS day, 2014," *Data*, vol. 4, no. 2, p. 84, Jun. 2019.
- [18] V. Gupta and G. S. Lehal, "A survey of text mining techniques and applications," *J. Emerg. Technol. Web Intell.*, vol. 1, no. 1, pp. 60–76, Aug. 2009.
- [19] K. M. Hammouda and M. S. Kamel, "Efficient phrase-based document indexing for Web document clustering," *IEEE Trans. Knowl. Data Eng.*, vol. 16, no. 10, pp. 1279–1296, Oct. 2004.

[20] X. Han, J. Wang, M. Zhang, and X. Wang, "Using social media to mine and analyze public opinion related to COVID-19 in China," *Int. J. Environ. Res. Public Health*, vol. 17, no. 8, p. 2788, Apr. 2020.