© INTERNATIONAL JOURNAL FOR RESEARCH PUBLICATION & SEMINAR
ISSN: 2278-6848 | Volume: 14 Issue: 03 | April - June 2023
Paper is available at http://www.jrps.in | Email : info@jrps.in
Refereed & Peer Reviewed
Special Edition
NCASIT 2023, 29th April 2023
Department of Computer Engineering,
St. Vincent Pallotti College of Engineering & Technology, Nagpur

The implementation of the Naive Bayes Algorithm was utilized to detect tweets related to disasters

1st Jayant Arsode
Computer Science and Engineering
Yeshwantrao Chavan College of Engineering Nagpur, India
jayantarsode234@gmail.com

2nd Faizan Habib
Computer Science and Engineering
Yeshwantrao Chavan College of Engineering Nagpur, India
fhhr842@outlook.com

Abstract —Social media has become an important part of everyday life, with Twitter being a popular micro-blogging and social networking platform that allows users to share news, information, and personal thoughts. In times of emergencies or disasters, Twitter has proven to be a crucial communication medium. The widespread use of smartphones and tablets enables individuals to report emergencies in real-time, which can potentially save countless lives by alerting others to take necessary precautions. Several organizations are attempting to analyze tweets programmatically to detect disasters and emergencies, which can be beneficial to millions of internet users by providing timely alerts in the event of a crisis. However, the challenge lies in distinguishing between tweets related to a disaster and those that are unrelated. Twitter data is unstructured, making it necessary to use Natural Language Processing (NLP) to classify tweets as either "Related to Disaster" or "Not Related to Disaster". This research paper focuses on building a Naïve bayes classifier model and evaluating its accuracy by predicting on a test set created from the original dataset.

Keywords — Twitter analysis, disaster identification, text analysis, text mining, document term matrix (DTM), Natural Language Processing (NLP) and Naive Bayes

I. Introduction

Twitter was first introduced on July 15, 2006, as a messaging service for a small group of users. Over time, it evolved into a micro-blogging and social networking portal that allows users to post short messages, reviews, and updates known as "tweets." Registered users can post, like, and retweet tweets, while unregistered users can only view them. Since its inception, Twitter has experienced tremendous growth. As of October 2019, there are approximately 6,000 tweets posted every second worldwide, resulting in around 350,000 tweets per minute, 500 million tweets per day, and 200 billion tweets per year. This enormous amount of data generates millions of petabytes (210 terabytes) of information globally, including essays, poems, data, and all kinds of information imaginable.

The data obtained from Twitter is unstructured since it lacks a specific prescribed format. This unstructured nature presents a significant challenge in analyzing the data. Therefore, Natural Language Processing (NLP) is used to analyze the data. After performing data cleaning, the tweets are tokenized into words, and the tokenized text data is analyzed.

Emergencies and disasters are frequent and occur globally. With the increasing use of smartphones, laptops, and tablets, individuals can report real-time experiences during a disaster. Thus, Twitter has emerged as a crucial platform for communication during emergencies and disasters. Many data analytics agencies are attempting to programmatically monitor and analyze Twitter data, particularly news agencies and disaster relief organizations. They are trying to analyze tweets in real-time to identify disaster occurrences from tweets. This would assist millions of people in avoiding the dangers of impending disasters. If people could be informed of disasters occurring at a particular location in real-time, they could take evasive action. Government agencies could execute evacuations before the situation becomes uncontrollable.

This paper analyzes a dataset comprising of 7613 tweets, tweet-ids, locations, tweet keywords, and their respective classes - "Disaster-related" or "Not Disaster-related." A trained classifier model is created

© INTERNATIONAL JOURNAL FOR RESEARCH PUBLICATION & SEMINAR
ISSN: 2278-6848 | Volume: 14 Issue: 03 | April - June 2023
Paper is available at http://www.jrps.in | Email : info@jrps.in
Refereed & Peer Reviewed
Special **Edition**
NCASIT 2023, 29th April 2023
Department of Computer Engineering,
St. Vincent Pallotti College of Engineering & Technology, Nagpur

using these tweets to predict the class of a new tweet. The classifier model is tested using a test set of tweets to produce predictions. The Naïve bayes Classification Algorithm is used for classification in this research.

The primary objective is to classify tweets into two categories - "Disaster-related tweets" and "Not Disaster-related tweets." The accuracy of the model is determined by generating a Confusion Matrix, and the AUC score is calculated.

The Python programming language and Visual Studio IDE are used in this research paper. Python is a popular data analytics and statistical programming language that is used to generate a classifier model using the Naïve bayes.

The paper is organized into several sections. Section I provides a brief introduction to the topic. Section II outlines the research objectives and methodology. Section III presents a review of related work in this area. Section IV discusses the theoretical background of Natural Language Processing and mining of unstructured data. Section V describes the data source used for this study. Section VI explains the methodology used to analyze the data. Finally, Section VIII summarizes the key findings and conclusions of the research.

II. OBJECTIVE

The primary aim of this research paper is to analyze a dataset of tweets from Twitter and classify them as either "Disaster Related" or "Not Disaster Related." The dataset is first cleaned and tokenized before being divided into a training set and a testing set to assess accuracy. The classifier model is built using the Naïve bayes Classification Algorithm and the work falls under the domain of Natural Language Processing.

III. RELATED WORK

There exist various research works in the field of text mining using Natural Language Processing that have gained recognition. For instance, Bagheri and Islam developed a predictive model for sentiment analysis of Twitter data by classifying tweets based on movies, politics, fashion, fake news, justice, and humanity into positive, neutral, and negative categories using Textblob Python library and NLTK for NLP (Bagheri H, Islam Md J, 2017) [1]. Hasan et al. collected tweets related to specific political topics and hashtags in Pakistan and conducted sentiment analysis of tweets using Naive Bayes and SVM classifier (Hasan A, Moin S, Karim A, Shamshirband S, 2018) [2]. They classified tweets into positive, neutral, and negative categories and used Textblob, SentiWordNet, and W-WSD for text analysis. Bharti and Malhotra performed Twitter data sentiment analysis using KNN, Naive Bayes, Modified K-Means with Naive Bayes, and Modified K-Means along with Naive Bayes and KNN (Bharti O, Malhotra M, 2016) [3]. They compared the accuracy results. Ikonomakis, Kotsiantis, and Tampakas worked in the field of text mining, feature selection, and feature transformation for text data analysis (Ikonomakis Emmanouil K, Kotsiantis S, Tampakas V, 2005) [4]. They used machine learning algorithms like KNN and SVM to show how accuracy, recall, and precision can be calculated and emphasized the use of PCA and Document term matrix. Khan et al. worked on text mining of NLP using feature extraction such as tokenization, stop-words removal, and stemming, and feature selection such as document vector representation, and conducted a comparative study of text classification using SVM, Naive Bayes, and KNN (Khan A, Baharudin B, Lee L H, Khan K, 2010) [5].Rish conducted research and a survey regarding the various methods used in Bayesian classification. Bayesian classifiers assign the most probable class to an example based on its feature vector. It is possible to simplify the process of learning these classifiers by assuming that the features are independent when given a class.[7].

Juliane utilized the Naïve Bayes classifier to obtain sentiment classification results for public figures. The classification was performed on Twitter data and the analysis utilized a combination of unigram, negation, and term-frequency features.[8]. They evaluated the accuracy, recall, precision, and f-measure of the model built.

IV.LITERATURE RELATED TO TEXT ANALYSIS USING NATURAL LANGUAGE PROCESSING

This research paper aims to analyze unstructured and unformatted Twitter data that has been generated by millions of users worldwide. To process and understand such datasets, Natural Language

© INTERNATIONAL JOURNAL FOR RESEARCH PUBLICATION & SEMINAR
ISSN: 2278-6848 | Volume: 14 Issue: 03 | April - June 2023
Paper is available at http://www.jrps.in | Email : info@jrps.in
Refereed & Peer Reviewed
Special **Edition**
NCASIT 2023, 29th April 2023
Department of Computer Engineering,
St. Vincent Pallotti College of Engineering & Technology, Nagpur

Processing (NLP) is required. NLP is an interdisciplinary field that involves language processing, linguistic analysis, computer science, information technology, text mining, opinion mining, and artificial intelligence to facilitate communication between computer systems and human languages. It deals with the challenges related to text extraction, speech recognition, natural language analysis, text interpretation, and natural language generation. NLP is an automated manipulation of natural language, including speech and text, using AI-based programs. The field of NLP has been evolving for more than half a century and has become a cutting-edge technology due to advancements in linguistic-based automated tools and processing power. In essence, NLP is a field that focuses on enabling machines to understand human speech and decipher it to comprehend the meaning behind the communication. NLP finds applications in various fields, including Google Translate, Word applications, Grammarly, Interactive Voice Response (IVR), and many more.

This paper focuses on the analysis of Twitter data and its relevance to sentiment analysis. Sentiment analysis involves determining the overall opinion or mood of the public in general or with regard to a particular topic based on the text of tweets. The Twitter API is used to extract data by using "consumer key", "access key", and "access token". The extracted data is then subjected to exploratory data analysis to determine if there is a class imbalance present. Class imbalance refers to the phenomenon where one class, such as "positive", "negative", or "neutral", significantly outweighs the others. If class imbalance is present, up-scaling or down-scaling is necessary. Up-scaling involves increasing the frequency of the low-magnitude class variable, while down-scaling involves decreasing the frequency of the high-magnitude class variable. Once class imbalance has been addressed, a training model can be created for building a classifier.

To prepare the tweets retrieved from the Twitter repository for analysis, it is necessary to perform data pre-processing and cleaning. This involves several steps, including creating a corpus, converting all text to lowercase, removing stop words, removing punctuation, eliminating URLs and usernames, removing leading and trailing blank spaces, and applying text stemming.

A corpus is essentially a collection of text data. In the context of machine learning, a corpus consists of a collection of written materials. In this case, the retrieved tweets are transformed into a corpus to facilitate further processing and analysis. Although the dataset of tweets contains a range of information, including tweet IDs, usernames, locations, and timestamps, only the text itself is needed for analysis and prediction. Thus, the dataset is converted into a text corpus that organizes the tweet text pieces in a tabular format, as illustrated in Table 1 below.

TABLE 1: EXAMPLE OF A CORPUS OF TWEETS

Words	Root word
Consult , consultant , consulting , consulted , consultancy	Consult
Likes , likely , liked	Like

Corpus row id	Corpus text
1	Hi, I am at California
2	We are experiencing snowfall at Darjeeling
3	Down with a bout of flu..

Once the corpus has been created, it is important to perform some data cleaning. The tweets in the dataset may contain words written in uppercase or lowercase

© INTERNATIONAL JOURNAL FOR RESEARCH PUBLICATION & SEMINAR
ISSN: 2278-6848 | Volume: 14 Issue: 03 | April - June 2023
Paper is available at http://www.jrps.in | Email : info@jrps.in
Refereed & Peer Reviewed
Special **Edition**
NCASIT 2023, 29th April 2023
Department of Computer Engineering,
St. Vincent Pallotti College of Engineering & Technology, Nagpur

letters. To ensure consistency, the entire text corpus should be converted to either lowercase or uppercase. In this paper, we have chosen to convert all text to lowercase since most R libraries provide their bag of words in lowercase format.

Next, it is important to remove stop-words from the dataset. Stop-words refer to common conjunctions and prepositions that do not significantly affect the overall sentiment or meaning of the text. Additionally, punctuations should be removed from the corpus for the same reason.

Stemming is a crucial aspect of text analysis. It involves reducing derived or inflected words to their basic forms. However, over-stemming can occur when two words with different stems are reduced to the same root word. Table 2 provides an example of stemming.

TABLE 2: EXAMPLE OF STEMMING

There are various algorithms available for stemming, such as Potter's stemming algorithm, Lovins algorithm, Dawson algorithm, and the NGram algorithm. For the purpose of this paper, we have chosen to use Potter's stemming algorithm.

In order to analyze the frequency of word occurrences in each tweet of the corpus, the study uses a Document Term Matrix (DTM). A DTM is essentially a matrix that represents tweets as documents in rows (or tuples) and represents words as columns. The matrix stores information about the frequency of word occurrences in each tweet. A nXm DTM signifies that there are n rows representing n tweet documents, and m unique words in the corpus. Each cell in the DTM records the frequency of a word appearing in a document. Essentially, a DTM is a mathematical matrix that holds the frequency values of terms in a collection of documents. Each value within the matrix, such as DTM (i, j), represents the frequency of the jth term in the ith document.

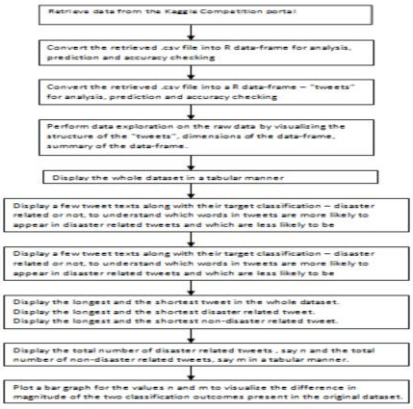
This text discusses the use of a Sparse Document Term Matrix (SDTM) in the context of analyzing word frequency in a corpus. While a Document Term Matrix (DTM) represents each document as a row and each word as a column, with the frequency of each word in a document stored in the corresponding matrix cell, a large number of values in a DTM are often zeroes, indicating that certain words do not appear in particular documents. This can cause issues when training a classifier model, as rare words can still act as factors in the model and make it less efficient.

To address this problem, a SDTM is used, which is a subset of a DTM where sparsely occurring words or terms are removed. The creation of a SDTM involves specifying a lower threshold value for word occurrence, such that only words that appear in at least a certain percentage of documents are included in the matrix. By doing so, the number of columns in the DTM is greatly reduced, which can improve the efficiency of the model without sacrificing accuracy. The Naive Bayes Classifier Algorithm was employed in the paper for data classification purposes. This algorithm is probabilistic and uses Bayes' theorem, which calculates the likelihood of an event based on prior knowledge or beliefs. Naive Bayes, as the name suggests, assumes that the presence or absence of a feature is not related to the presence or absence of any other feature. When utilized for classification purposes, the Naive Bayes algorithm relies on training data to determine the likelihood of each class based on a given set of features. The algorithm assumes that each feature contributes independently to the probability of the class and computes the joint probability of all the features for the given class. Ultimately, the algorithm identifies the class with the highest probability and assigns it to the input data point. There are multiple variations of the Naive Bayes algorithm, such as Gaussian Naive Bayes, Multinomial Naive Bayes, and Bernoulli Naive Bayes, each designed to handle different data types. Gaussian Naive Bayes works best with continuous data, Multinomial Naive Bayes is optimal for discrete count data, and Bernoulli Naive Bayes is well-suited for binary data. The Naive Bayes algorithm finds

	Class Negative Actual	**Class Positive Actual**
Class Negative Predicted	TN	FN
Class Positive Predicted	FP	TP

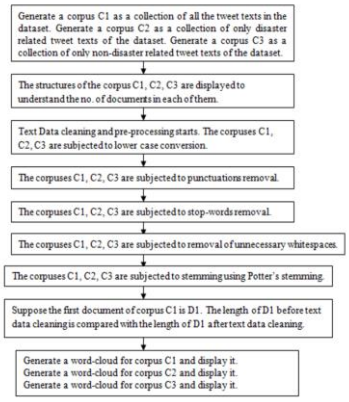
© INTERNATIONAL JOURNAL FOR RESEARCH PUBLICATION & SEMINAR
ISSN: 2278-6848 | Volume: 14 Issue: 03 | April - June 2023
Paper is available at http://www.jrps.in | Email : info@jrps.in
Refereed & Peer Reviewed
Special **Edition**
NCASIT 2023, 29th April 2023
Department of Computer Engineering,
St. Vincent Pallotti College of Engineering & Technology, Nagpur

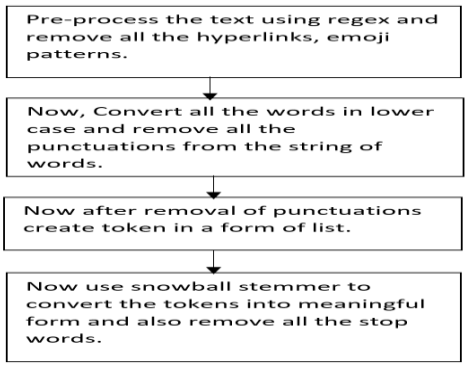
widespread use in several applications, including text classification, spam filtering, sentiment analysis, and recommendation systems, among others. The algorithm is renowned for its simplicity and speed, making it ideal for handling large datasets with high-dimensional feature spaces. Nonetheless, the



algorithm's assumption of feature independence may not always hold, which could lead to suboptimal performance in some cases.

The trained model's accuracy is evaluated by utilizing the AUC score and Confusion Matrix. The AUC score measures the "Area Under Curve" of a ROC curve, which is a graphical representation of a classification model's performance at different classification thresholds. The ROC curve plots the True Positive Rate (TPR) against the False Positive Rate (FPR), where TPR is defined as TP / (TP + FN) and FPR is defined as FP / (FP + TN). The AUC score ranges from 0 to





1, where 0 represents 100% incorrect prediction and 1 represents 100% correct prediction.

The Confusion Matrix is a tool used to evaluate the performance of a classifier model on test data where the actual reference values are already known. In this paper, the predicted class values are listed as rows and the actual class values (reference values) are listed as columns in order to generate the Confusion Matrix. The Confusion Matrix includes four values - True Positive (TP), False Positive (FP), True Negative (TN), and False Negative (FN). Table 3 provides a tabular representation of the Confusion Matrix.

TABLE 3: CONFUSION MATRIX

The Confusion Matrix is a tool that allows for the evaluation of a classifier model's performance on a test dataset, where the true reference values are known. The matrix displays predicted class values as rows and actual class values as columns, making it possible to calculate metrics such as accuracy and precision. The classifier model's performance is typically evaluated using metrics such as accuracy and precision. Two important metrics for evaluating a classifier model are accuracy and precision. Accuracy measures the proportion of correct predictions out of the total predictions, while precision measures the proportion of true positives out of the total positive predictions. Both metrics provide insights into the model's effectiveness.

Accuracy = (TP+TN) / (TP+FP+TN+FN) and
Precision = TP / (TP + FP) .

V. SOURCE OF THE DATA USED FOR THE EXPERIMENT

The dataset used in this research was obtained from the KAGGLE portal for data analytics and machine learning. It is a competition dataset related to Natural

© INTERNATIONAL JOURNAL FOR RESEARCH PUBLICATION & SEMINAR
ISSN: 2278-6848 | Volume: 14 Issue: 03 | April - June 2023
Paper is available at http://www.jrps.in | Email : info@jrps.in
Refereed & Peer Reviewed
Special Edition
NCASIT 2023, 29th April 2023
Department of Computer Engineering,
St. Vincent Pallotti College of Engineering & Technology, Nagpur

Language Processing (NLP). The dataset consists of 7613 rows representing tweets, and has 5 columns: Tweet Id, Text, Location, Keyword, and Target. In this study, the 'Target' is the dependent variable, which is classified into two classes - 1 (tweets related to disasters) or 0 (tweets not related to disasters).

VI.METHODOLOGY OF THE EXPERIMENT

The study is typically separated into four primary stages, which include data retrieval and exploration, corpus generation and exploration, document-term matrix generation and exploration, and training model generation and accuracy assessment. The flowcharts that illustrate these stages are presented in Figures 1, 2, and 3 respectively.

6.1. Data Retrieval and Exploration

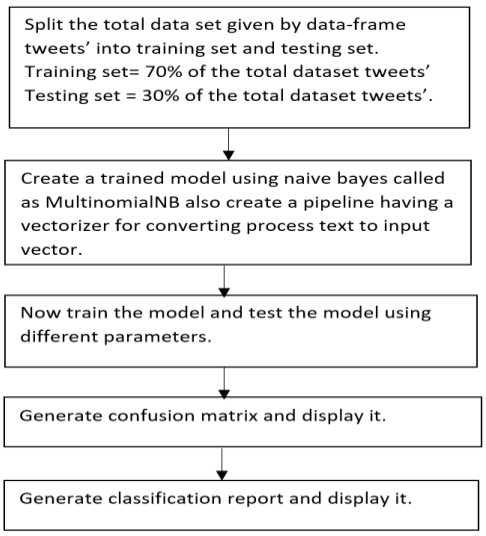
6.2. Text data preprocessing

6.3. Training Model Generation and Accuracy

Rows (tuples)	Columns (attributes)
7613	5

Tweet text	Tweet Classification
"Our Deeds are the Reason of this #earthquake May ALLAH Forgive us all"	1 (Disaster related)
"Forest fire near La Ronge Sask. Canada"	1 (Disaster related)

Checking



VII. RESULTS AND DISCUSSION

The study described in this paper was carried out using the R programming language and the RGUI IDE. The dataset used in the experiment was obtained from the KAGGLE portal for data analytics and machine learning. The dataset contains 7613 tuples (rows) representing individual tweets, with five attributes (columns) including Tweet Id, Text, Location, Keyword, and Target. A detailed structure of the data-

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7613 entries, 0 to 7612
Data columns (total 5 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   id        7613 non-null   int64
 1   keyword   7552 non-null   object
 2   location  5080 non-null   object
 3   text      7613 non-null   object
 4   target    7613 non-null   int64
dtypes: int64(2), object(3)
memory usage: 297.5+ KB
```

frame, including attribute names and data types, is illustrated in Figure 5.

Fig. 5: Structure of the data-frame

The dimension of the data-frame is given in Table 4 :

TABLE 4: DIMENSIONS OF THE DATA-FRAME

The snapshot of a portion of the data retrieved is given in Fig. 6:

© INTERNATIONAL JOURNAL FOR RESEARCH PUBLICATION & SEMINAR
ISSN: 2278-6848 | Volume: 14 Issue: 03 | April - June 2023
Paper is available at http://www.jrps.in | Email : info@jrps.in
Refereed & Peer Reviewed
Special **Edition**
NCASIT 2023, 29th April 2023
Department of Computer Engineering,
St. Vincent Pallotti College of Engineering & Technology, Nagpur

Fig. 6: A snapshot of the data

Basic exploration of the dataset was conducted next. The first two tweets along with their corresponding 'target' variable values were tabulated in Table 5 to provide an insight into the nature of tweets and their respective classifications.

TABLE 5: DATA EXPLORATION OF THE FIRST 2 TUPLES OF TWEETS

Table 6 presents information on the lengths of the longest tweets in the dataset. Specifically, it shows the length of the longest tweet overall, the length of the longest tweet related to a disaster, and the length of the longest tweet not related to a disaster.

TABLE 6: TABULATION OF THE LENGTH OF LONGEST TWEETS ACCORDING TO CATEGORIES

Table 7 displays the lengths of the shortest tweet in the entire dataset, the shortest tweet related to disasters, and the shortest tweet not related to disasters.

TABLE 7: TABULATION OF THE LENGTH OF SHORTEST TWEETS ACCORDING TO CATEGORIES

The dataset has been analyzed to determine the number of disaster-related tweets and the number of non-disaster-related tweets. Table 8 summarizes the results of this analysis.

TABLE 8: NO. OF DISASTER RELATED TWEETS AND NON DISASTER RELATED TWEETS

A bar graph is plotted for the same, which is shown in Fig. 7:

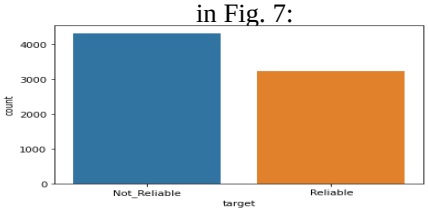


Fig. 7: Bar graph of disaster-related tweets vs. Non disaster related tweets

Length of the longest tweet in the entire dataset	Length of the longest disaster-related tweet	Length of the longest non disaster-related tweet
163 characters	161 characters	163 characters
7 characters	14 characters	7 characters

No. of disaster related tweets	No. of non-disaster related tweets
3271	4342

The preprocessing steps mentioned above result in a considerable decrease in the number of characters in each document. To investigate the magnitude of this reduction, we examine a single document from the corpus. Table 9 displays the results of the preprocessing steps applied to the first document in the corpus.

After the data pre-processing, a word-cloud is generated to represent the frequency of words in the text corpus. The font size of a particular word in the corpus is directly proportional to the frequency of the word in the text corpus. Thus the word-cloud is a very effective analysis and visualization tool for exploring

© INTERNATIONAL JOURNAL FOR RESEARCH PUBLICATION & SEMINAR
ISSN: 2278-6848 | Volume: 14 Issue: 03 | April - June 2023
Paper is available at http://www.jrps.in | Email : info@jrps.in
Refereed & Peer Reviewed
Special **Edition**
NCASIT 2023, 29th April 2023
Department of Computer Engineering,
St. Vincent Pallotti College of Engineering & Technology, Nagpur

a text corpus. In this experiment, three word-clouds are generated -

The following word-clouds were generated to represent the frequency of words in the text corpus after data pre-processing:

i) "Wordcloud_dis" for only those documents of the corpus which correspond to disaster related tweets.

ii) "Wordcloud_nondis" for only those documents of the corpus which correspond to non-disaster related tweets.

These word-clouds are displayed in Figure 8 and Figure 9, respectively.

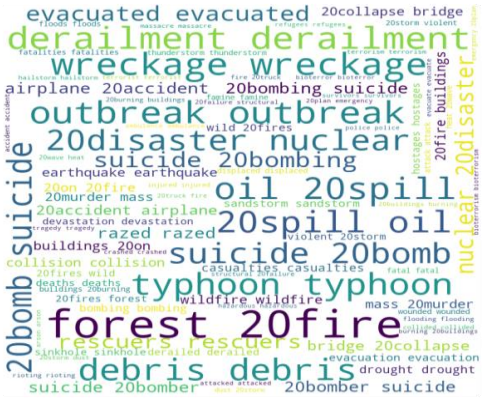


Fig. 8: Word-cloud for only those documents of the corpus which correspond to disaster related tweets.

Fig. 9: Word-cloud for only those documents of the corpus which correspond to non-disaster related tweets.

The authors of the paper have discussed the creation of a DTM (Document Term Matrix) from the text corpus, which is a mathematical matrix where the rows correspond to each document and the columns correspond to terms in the documents. The values within the matrix represent the frequency of the corresponding term in the respective document. The dimensions of the created DTM have been presented in Table 10.

Rows(documents)	Colums(Terms)
7552	14604

TABLE 10: DIMENSIONS OF DOCUMENT TERM MATRIX (DTM)

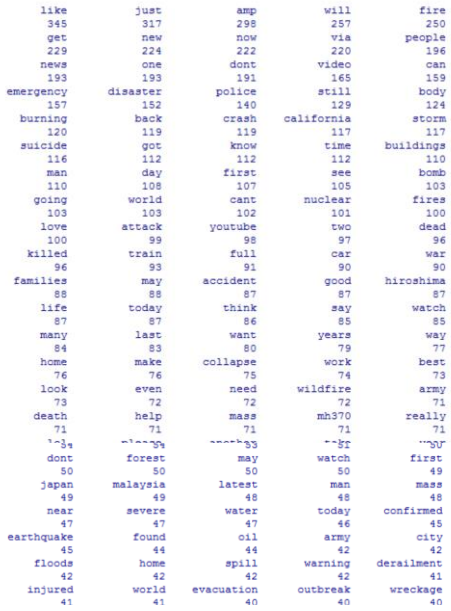
In order to handle the issue of a huge number of zero values in the Document Term Matrix (DTM), a Sparse Document Term Matrix (SPDTM) is created. This is done when a large number of values in the matrix are zero. In the DTM that has been created, only 37,906 non-zero values are present out of the total 110,289,408 values stored in the DTM. This indicates that there are many words in the DTM that have low frequency across the original corpus, and they can be ignored without affecting the predictive quality of the model. Removing these rarely occurring words reduces the number of factor variables, which leads to faster and more efficient model creation. In this study, only those terms that appear in at least 0.4% (30) of the total number of documents (tweets) are kept in the SPDTM. The dimensions of the newly generated SPDTM are provided in Table 11:

Rows(documents)	Columns(Terms)
7552	536

TABLE 11: DIMENSIONS OF SPARSE DOCUMENT TERM MATRIX (SPDTM)

© INTERNATIONAL JOURNAL FOR RESEARCH PUBLICATION & SEMINAR
ISSN: 2278-6848 | Volume: 14 Issue: 03 | April - June 2023
Paper is available at http://www.jrps.in | Email : info@jrps.in
Refereed & Peer Reviewed
Special **Edition**
NCASIT 2023, 29th April 2023
Department of Computer Engineering,
St. Vincent Pallotti College of Engineering & Technology, Nagpur

After creating the Document Term Matrix (DTM), we observed that only 37,906 out of 4,047,872 values in the matrix were non-zero, indicating the presence of a large number of words with low frequency. To enhance the efficiency of the predictive model and speed up the model creation process, we decided to remove these rarely occurring words. In order to do this, we only kept the terms that appear in at least 0.4% (30) of the total number of documents (tweets), resulting in the creation of a Sparse Document Term Matrix (SPDTM) with 457 terms, compared to the original 19,109 terms. The list of terms in the SPDTM

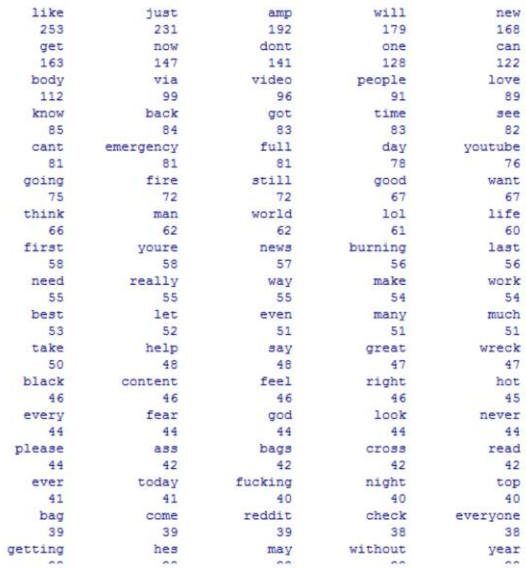


like	just	amp	will	fire
345	317	298	257	250
get	new	now	via	people
229	224	222	220	196
news	one	dont	video	can
193	193	191	165	159
emergency	disaster	police	still	body
157	152	140	129	124
burning	back	crash	california	storm
120	119	119	117	117
suicide	got	know	time	buildings
116	112	112	112	110
man	day	first	see	bomb
110	108	107	105	103
going	world	cant	nuclear	fires
103	103	102	101	100
love	attack	youtube	two	dead
100	99	98	97	96
killed	train	full	car	war
96	93	91	90	90
families	may	accident	good	hiroshima
88	88	87	87	87
life	today	think	say	watch
87	87	86	85	85
many	last	want	years	way
84	83	80	79	77
home	make	collapse	work	best
76	76	75	74	73
look	even	need	wildfire	army
73	72	72	72	71
death	help	mass	mh370	really
71	71	71	71	71
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
dont	forest	may	watch	first
50	50	50	50	49
japan	malaysia	latest	man	mass
49	49	48	48	48
near	severe	water	today	confirmed
47	47	47	46	45
earthquake	found	oil	army	city
45	44	44	42	42
floods	home	spill	warning	derailment
42	42	42	42	41
injured	world	evacuation	outbreak	wreckage
41	41	40	40	40

sorted by frequency is shown in Figure 11.

Fig. 11: Frequency of words in a decreasing order for the entire dataset

Next, we explore the frequency of terms appearing in only disaster-related tweets (documents) of the



like	just	amp	will	new
253	231	192	179	168
get	now	dont	one	can
163	147	141	128	122
body	via	video	people	love
112	99	96	91	89
know	back	got	time	see
85	84	83	83	82
cant	emergency	full	day	youtube
81	81	81	78	76
going	fire	still	good	want
75	72	72	67	67
think	man	world	lol	life
66	62	62	61	60
first	youre	news	burning	last
58	58	57	56	56
need	really	way	make	work
55	55	55	54	54
best	let	even	many	much
53	52	51	51	51
take	help	say	great	wreck
50	48	48	47	47
black	content	feel	right	hot
46	46	46	46	45
every	fear	god	look	never
44	44	44	44	44
please	ass	bags	cross	read
44	42	42	42	42
ever	today	fucking	night	top
41	41	40	40	40
bag	come	reddit	check	everyone
39	39	39	38	38
getting	hes	may	without	year

SPDTM. The frequencies of these terms are sorted in decreasing order and displayed in Figure 12:

Fig. 12: Frequency of words in a decreasing order for only those documents which are represent disaster represent disaster related tweet.

Acknowledging the source of the research paper, it is evident from the exploration conducted in the study that certain words such as "fire", "news", "disaster", and "police" have a significantly high frequency in tweets related to disasters. Additionally, the study also involved sorting the frequency of terms appearing only in non-disaster related tweets (documents) of the SPDTM in a decreasing order, and the resulting sorted list is presented in Figure 13.

Fig. 13: Frequency of words in a decreasing order for only those documents which represent non-disaster related tweets

After conducting exploratory data analysis, it was found that words such as "fire," "news," "disaster," and

© INTERNATIONAL JOURNAL FOR RESEARCH PUBLICATION & SEMINAR
ISSN: 2278-6848 | Volume: 14 Issue: 03 | April - June 2023
Paper is available at http://www.jrps.in | Email : info@jrps.in
Refereed & Peer Reviewed
Special **Edition**
NCASIT 2023, 29th April 2023
Department of Computer Engineering,
St. Vincent Pallotti College of Engineering & Technology, Nagpur

"police" were frequently used in disaster-related tweets. In contrast, common, non-specific words such as "like," "just," "will," and "new" had a high frequency in non-disaster related tweets. This information can help in identifying potential disaster-related tweets.

To proceed with building a classifier, the dataset was split into a training set and a testing set, with 70% and 30% of the data respectively. Additional fields such as tweet ID, location, keyword, and target were included in both datasets. The table below shows the structure of the training and testing sets (Table 12):

TABLE 12: DATA-FRAME STRUCTURE OF TRAINING AND TESTING DATASETS

	No. of observations	No. of variables
Training Set	5329	461
Testing Set	2284	461

To create a training model, the Naïve bayes classification algorithm was used on the Training dataset. The response variable was set as the target variable, dependent on all the terms derived from the SPDTM as well as the tweet id, location, and keyword. Using the trained model, predictions were generated for the Testing dataset. The prediction is a probability value ranging from 0 to 1, which is then rounded off to determine the predicted value of the target variable. The target variable has two possible values: 0 for non-disaster related tweets and 1 for disaster-related tweets. The ROC graph was created by plotting the True Positive Rate (TPR) against the False Positive Rate (FPR) and is displayed in Figure 14.

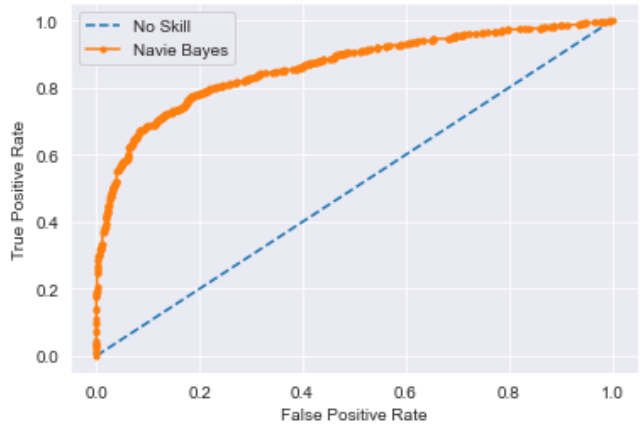


Fig 14: Roc Graph

To assess the model's accuracy, the study computes the Area Under Curve (AUC) score. This score reflects the area beneath the ROC (Receiver Operating Characteristic) curve. The ROC curve is plotted by plotting the True Positive Rate (TPR) against the False Positive Rate (FPR). The AUC score is shown in Table 13.

AUC Score	AUC Score (%)
0.861	86.1

TABLE 13: AUC SCORE

To evaluate the accuracy and precision values of the predicted model, a confusion matrix is generated. The matrix is displayed in Figure 15.

	Reference	
Prediction	0	1
0	791	74
1	206	440

Fig 15: Confusion Matrix

Table 14 presents a summary of the Accuracy, Precision, Recall, and F-score values obtained from the confusion matrix.

TABLE 14: ACCURACY, PRECISION, RECALL, F-SCORE VALUES

© INTERNATIONAL JOURNAL FOR RESEARCH PUBLICATION & SEMINAR
ISSN: 2278-6848 | Volume: 14 Issue: 03 | April - June 2023
Paper is available at http://www.jrps.in | Email : info@jrps.in
Refereed & Peer Reviewed
Special **Edition**
NCASIT 2023, 29th April 2023
Department of Computer Engineering,
St. Vincent Pallotti College of Engineering & Technology, Nagpur

	In value	In Percentage
Accuracy	0.81	81
Precision	0.79	79
Recall	0.91	91
F-measure score	0.85	85

The results of the trained model using the Naïve bayes algorithm indicate a satisfactory performance. An accuracy score of over 81% is considered good, and the F-measure score of over 85% is also deemed satisfactory. While the research paper's objective has been met, there is still scope for improvement in terms of increasing the accuracy.

VIII. CONCLUSION AND SCOPE OF FUTURE

The goal of this research paper is to classify tweets into two categories, either related to disasters or not, using text mining techniques on Twitter datasets. The experimental results suggest that Twitter data can be classified with a good level of accuracy. However, if the frequency of terms is decreased below a threshold of 0.5-1%, it may negatively impact performance. Furthermore, reducing False Positive and False Negative values can help improve the accuracy of the model.

Future research in this field can focus on using different classification algorithms such as K-nearest neighbors, SVM, or Neural Networks instead of Decision Trees. These aspects provide scope for further studies and improvements in this field.

REFERENCES

[1] Goswami, S. and Raychaudhuri, D., 2020. Identification of Disaster-Related Tweets Using Natural Language Processing: International Conference on Recent Trends in Artificial Intelligence, IOT, Smart Cities & Applications (ICAISC-2020). *IOT, Smart Cities & Applications (ICAISC-2020)(May 26, 2020)*.

[2] Bagheri H , Islam Md J (2017). Sentiment analysis of twitter data. *Computer Science Department Iowa State University,United States of America*.

[3] Hasan A , Moin S, Karim A , Shamshirband S (2018). Machine learning- based sentiment analysis for twitter accounts. *Mathematical and Computational Applications*, 23(11), 1-15.

[4] Bharti O, Malhotra M (2016).Sentiment analysis on twitter data. *International Journal of Computer Science and Mobile Computing*, 5(6): 601 – 609.

[5] Ikonomakis Emmanouil K , Kotsiantis S , Tampakas V (2005). Text classification using machine learning techniques .*Wseas Transactions on Computers*, 4(8): 966-974.

[6] Khan A , Baharudin B, Lee LH, Khan K (2010). A review of machine learning algorithms for text-documents classification. *Journal of Advances in Information Technology*, 1(1): 4-20**.**

[7] Rish, I., 2001, August. An empirical study of the naive Bayes classifier. In IJCAI 2001 workshop on empirical methods in artificial intelligence (Vol. 3, No. 22, pp. 41-46).

[8] Juliane, C., 2021, November. Implementation of Naive Bayes Algorithm on Sentiment Analysis Application. In 2nd International Seminar of Science and Applied Technology (ISSAT 2021) (pp. 193-200). Atlantis Press.

AUTHORS PROFILE

Faizan Habib - CSE @YCCE '24 | Full stack web developer | Vice President @Codeware | Technical Team Member @C.O.S.M.O.S | 3★ on Codechef | Chegg Expert | Open Source Contributor.

Jayant Arsode - Tech Enthusiast| Data Science | Artificial Intelligence | Machine Learning | Python | Learning | CSE Student at Yashwantrao Chavan College of Engineering - YCCE