Applying NLP and Machine Learning for Sentiment Classification Across Multiple Twitter Datasets

Ervin Vladić, Benjamin Mehanović, Mirza Novalić, Dino Kečo, Dželila Mehanović

***Abstract*—**Social networks have appeared as the main opinion-sharing and discussion-enabling resources in the last ten years. At the same time, the development of machine learning (ML) and natural language processing (NLP) technologies has allowed new approaches to analyzing the huge quantities of data users create. This research uses data loading, class imbalance handling, text preprocessing and tokenization, sentiment analysis, and model assessment techniques to analyze the sentiment of the tweets across two datasets. Using metrics like accuracy, precision, recall, and F1 score the study reveals that SVM and Logistic Regression are the most suitable machine-learning models for this purpose. SVM attained an accuracy of 90% for training and 77% for testing. In comparison, Logistic Regression showed 83% for training and 78% for testing for Dataset 1, and for Dataset 2 SVM attained 98% training and 86% for testing. In comparison, Logistic Regression showed balance between training and testing achieving 93% in training and 87% in testing.

***Keywords*—** Social Media Analysis, Classification Models, Data Preprocessing, Sentiment Analysis, Logistic Regression.

I. Introduction

At present, when technology is advancing rapidly, sites such as social networking sites have become a place wherein thoughts, opinions, and expressions are becoming the chief kind of interaction. It is sometimes difficult to know whether the content of posts, tweets, and messages is neutral, positive, or negative. Earlier, it wasn't easy to analyze the tweets or demonstrate the sentiment of the crowds but since sentiment classification has emerged it has become easier to analyze and demonstrate the sentiments of the people. A preliminary version of this research was presented at the NLMLT 2024 conference, where initial findings were shared (Vladić et al., 2024) [1].

New opportunities offered by sentiment classification techniques made this process more efficient and required no manual work. Sentiment classification can be regarded as an essential subfield of natural language processing and machine learning which is focused on the task of classification of textual data according to the sentiment that has been provided [2]. The final stage is to identify the sentiment that is encoded in the text, which helps to identify the state of public consciousness in social networks. This is due to the fast growth of social media and a huge volume of created information that has led to a higher demand for sentiment categorization. Today, different businesses, researchers, and even individuals have started using sentiment analysis to get more insights about people's opinions or reactions towards their products or services, or even the trends in the market. Using sentiment analysis, a business can focus on their brands products or topics that are trending in the market to align their strategies to public opinions while getting more insight about society and what people are expressing on social media.

In sentiment classification, NLP is central because models can be endowed with an understanding of natural language. NLP algorithms not only analyze the semantic meanings of words but also analyze the nuances of contextual nuances, sarcasm, and cultural relations inherent in tweets [3]. Compared to a typical keyword search, NLP can grasp not only the word's meaning but also the reader's attitude, and different meanings of the word depending on the context and some cultural references. This sophistication is especially important when using tweets as having character and informal language are commonplace and being able to extract sentiment is more complicated.

Classification of X-tweets as bearers of sentiments is an area situated at the crossroads of engineering computer sciences, linguistics, and sociology, as it reveals the general tendencies of sentiments in our practical discursive interactions. As AI and ML develop further the possibility of getting more meaningful insights from the flood of tweets is nearly unlimited, the culture of online communication is becoming richer and AI can help to see it more clearly. Such progression enables researchers to capture progressive changes in the attitude of the public while businesses harness sentiment to execute changes concerning their audiences. By using emotive analysis conceived from AI, there is apprehension to explain the reality of our social world as confined in our social media universe, the cultural and emotive texture of our societies as defined by online conversations.

A. Objectives

The first and major research question of this research work is to determine the best-performing sentiment classification technique using standard classification models and NLP methodologies. To accomplish this, the research will focus on several core tasks: first, preprocessing and tokenizing the tweets to convert them into a form suitable to feed it into a machine learning algorithm, in terms of the feature set of positive and negative sentiment labels. For feature representation improvement, this study will incorporate Term Frequency – Inverse Document Frequency (TF-IDF) where numbers represent the word's importance in the body of text needed for accurate sentiment evaluation. A training dataset with the positive-to-negative-entry ratio 1:1 will be then developed to train and evaluate the models as the entries are mixed randomly. The same approach will be used across

Department of Information Technology, International Burch University, Sarajevo, Bosnia and Herzegovina

 ISSN (Print): 2204-0595
ISSN (Online): 2203-1731

multiple datasets.

The issues that will be addressed in the present research are also essential for comparing different machine learning models: Naïve Bayes, including Bernoulli Naïve Bayes and Multinomial Naïve Bayes, Support Vector Machine, Stochastic Gradient Descent Classifier, Logistic Regression, Passive Aggressive Classifier, and Perceptron were chosen because each of the algorithms has its strengths when dealing with textual data. Last but not least, the study will compare the performances of each model by different parameters such as accuracy, precision, recall, and F1 score to identify the right model for sentiment classification. This study helps identify which social media analytical algorithms are more effective and can be used in developing efficient sentiment analysis.

B. Overview of Subsequent Sections

The remaining part of the paper is structured as follows: The Literature Review section describes the overview of sentiment analysis studies, giving details concerning the development of the sentiment analysis methodologies and the application of the theories across different fields, the Materials and Methods section describes the sentiment analysis datasets obtained from Kaggle.com consists of the explanation of the proposed method to tackle the class imbalance and prepare the data for analysis, Exploratory Data Analysis looks at word occurrences in positive and negative connotations and discusses features of the data, as well as potential trends, Results compare the performance of various machine learning models in terms of training and testing accuracy, precision, recall, and F1 score, and provides further analysis to assist with model selection and optimization. Last, the study summarizes focus on the importance of sentiment analysis when it comes to analyzing online communities and further research prospects to develop the depth and range of sentimental analysis.

II. LITERATURE REVIEW

While writing this term paper, the literature review sections have given a rather rich background on sentiment analysis in every context. This tour through research fields has revealed premises and techniques that are important.

Sentiment analysis of the tweets about COVID-19 is considered in the present work [4]. Specifically, they intended to classify sentiments as positive, negative, or neutral depending on the 72000 tweets in the dataset. Four models such as Support Vector Machine, Perceptron, Passive Aggressive Classifier, and Logistic Regression have promising results in terms of accuracy, all of them crossing 98%. However, the AdaBoost Classifier only fared worse achieving nearly about 73% when it came to classification accuracy.

Another research [5] was related to sentiment analysis on Twitter and Mining Twitter Data. This they had preprocessed and analyzed and according to them the Decision Tree classifier was superior to both KNN and Naive Bayes with an accuracy of 84.66% which was higher than the latter one with the accuracies of 50.72% and 64. 42%, respectively.

In [6], the study evaluated the performance of a Support Vector Machine (SVM) on two pre-labeled Twitter datasets: one is based on self-driving cars while the other one is based on Apple products. The performance of SVM yields an average precision rate of 55.8 %, a recall rate of 59.9 %, and an F-score of 57.2% for self-driving cars and 70.2%, 71.2%, and 69.9% for Apple products.

In [7] utilizing the analysis and classification of US airline-related tweets, different classifiers, feature extraction techniques, and preprocessing techniques were used. The highest accuracy, 80.4% was achieved by a voting classifier (VC) using TF-IDF feature extraction, and complete preprocessing. On the other hand, the Gaussian Naive Bayes model achieved the lowest accuracy which was 49.8%. These results show that this work is valuable to advance because TF-IDF feature extraction outperformed deep learning techniques like LSTM.

In another research [8] the author reported an accuracy of up to 88 % using the Random Forest algorithm complemented the feature set by unigram features and oversampling. The overall accuracy of Dataset 2 was 68% and the overall accuracy of Dataset 3 was 61% which was the lowest. However, it was observed that in the third dataset, the accuracy achieved by the method named J48 Decision Tree was immensely good even though the total accuracy was comparatively less.

Following that, a term paper [9] has proposed a system to classify Twitter tweets as good or bad depending on the query words. Naive Bayes followed by Maximum Entropy and SVM with the overall accuracy range at 80.5% to 83. 0%. These findings were in line with the study and findings of Pang & Lee.

In the paper [10], authors Maha Heikal, Marwan Torki, and Nagwa El-Makky discuss some of the issues with sentiment analysis in Arabic. To improve the sentiment prediction accuracy the authors put forward an architecture that is a combination of CNN and LSTM networks. Their model posts an evaluation F1-score of 64.46% thus outcompeting the previous best F1-score of 53.6% reported on this same Arabic Sentiment Tweets Dataset (ASTD). Listing the shortfalls of previous techniques, the study underlines that using deep learning techniques helps to cope with the difficulties arising from Arabic sentiment analysis and encourages the development of this aspect further.

Using a classification approach, Dr. Priyanka Harjule, Astha Gurjar, Harshita Seth, and Priya Thakur from IIIT Kota [11] predict the polarity of words in Twitters. There are two categories – first, lexicon-based techniques like SentiWordNet and second, the different machine learning classification algorithms like Multinomial Naive Bayes, Logistic Regression, Support Vector Machine and Recurrent Neural Networks (RNN). When testing these classifiers, the study utilizes data sets from “Sentiment140” and “Crowdflower” showing that RNNs with the LSTM approach yield over 90% which is much better than other classic algorithms. They put much emphasis on the suitability of supervised machine learning approaches when processing informal and less structured data from the tweet data as opposed to the lexicon-based techniques.

In their paper “Sentiment Analysis of Movie Reviews using Machine Learning Techniques” [12], Baid, Gupta, and Chaplot (2017) incorporated sentiment analysis of movie reviews with different machine learning approaches such as

 ISSN (Print): 2204-0595
ISSN (Online): 2203-1731

Naïve Bayes, K-Nearest Neighbour, and Random Forest. The authors conducted a study using 2000 user-generated movie reviews obtained from IMDb and found out that the best accuracy rate was given by the Naïve Bayes classifier with an accuracy rate of 81.45 %, followed by Random Forest with an accuracy rate of 78.65 %, while the K- Nearest Neighbor resulted to an accuracy rate of 55.30%. They also stressed the role of information and sentiment analysis in gathering information and decision-making concerning several fields like marketing and analysis of various products. In their paper, the authors prove that machine learning is useful for sentiment classification and show that Naïve Bayes is the most efficient method according to their outcomes.

Lastly, the study titled "Twitter Sentiment Analysis" [13] proved the necessity of proper sentiment analysis in the Twitter environment. It also focused on some of the challenges that include the interchangeability of frequently used terms and problems with data gathering. The concepts of employing Python and how competitiveness and decision-making with the help of consumer sentiments indicated in a pie chart could be maintained were covered. The study was most often able to identify null hashtags (84.1%), with lower percentages for positive and negative, 9. 4% and 6. 5%, respectively.

A. Overview

In the next subsection, we can see an overview of all the works that we have mentioned so far, which are closely related to the topic. The table shows the authors mentioned, the algorithms used, and the results of the works.

TABLE I
LITERATURE REVIEW OVERVIEW TABLE

#	Paper	Algorithms	Results
1	Kamal Gulati, S. Saravana Kumar, Raja Sarath Kumar Boddu, and Ketan Sarvakar [4]	Support Vector Machine, Perceptron, Passive Aggressive Classifier, Logistic Regression	All models achieved over 98% accuracy; AdaBoost Classifier: 73% accuracy
2	Md Shoeb, and Jawed Ahmed [5]	Decision Tree, K-Nearest Neighbors, Naive Bayes	Decision Tree: 84.66%; KNN: 50.72%; Naive Bayes: 64.42%
3	Munir Ahmad, Shabib Aftab, and Iftikhar Ali [6]	Support Vector Machine	Self-driving cars: Precision: 55.8%, Recall: 59.9%, F-score: 57.2%; Apple products: Precision: 70.2%, Recall: 71.2%, F-score: 69.9%
4	Furqan Rustam, Imran Ashraf, Arif Mehmood, Saleem Ullah, Gyu Sang Choi, and Yar Khan [7]	Voting Classifier, Gaussian Naive Bayes	Voting Classifier: 80.4%; Gaussian Naive Bayes: 49.8%
5	Mestan Fırat Çeliktuğ [8]	Random Forest, J48 Decision Tree	Random Forest: 88%; Dataset 2: 68%; Dataset 3: 61% (J48 accuracy significantly higher)
6	Alec Go, Richa Bhayani, and Lei Huang [9]	Naive Bayes, Maximum Entropy, Support Vector Machine	Accuracy range: 80.5% to 83.0%
7	Maha Heikal, Marwan Torki, and Nagwa El-Makky [10]	CNN, LSTM	F1-score: 64.46%, improving the previous best of 53.6%
8	Priyanka Harjule, Astha Gurjar, Harshita Seth, and Priya Thakur [11]	Lexicon-based techniques, Multinomial Naive Bayes, Logistic Regression, Support Vector Machine, Recurrent Neural Networks	RNNs with LSTM: over 90% accuracy
9	Palak Baid, Apoorva Gupta, and Neelam Chaplot [12]	Naïve Bayes, K-Nearest Neighbour, Random Forest	Naïve Bayes: 81.45%; Random Forest: 78.65%; KNN: 55.30%
10	Aliza Sarlan, Chayanit Nadam, and Shuib Basri [13]	Not specified	Identified null hashtags: 84.1%; Positive: 9.4%; Negative: 6.5%

III. MATERIALS AND METHODS

This section is divided into three subsections: The first section is entitled Dataset which gives the characteristics of the tweet data in the datasets; the second section: Data Processing and Analysis which outlines the preprocessing of the data across multiple datasets, cleaning, normalization processes, tokenization, and other processes which help to regulate the class imbalance; the last section is entitled Machine Learning Models which gives details of the models applied on the tweet data for sentiment classification.

A. Datasets

In this work, we use two different datasets to increase confidence in the classification of sentiment analysis results from Twitter data. The first dataset gives a basic stock of tweets regarding various domains and moods to start up the training models and assessments. On the other hand, Dataset 2 provides the newer or topic-specific tweets which can be useful to adjust model performance and assess whether its results apply to the different types of tweets.

a. Dataset 1

The dataset 1 used in this term paper was the sentiment analysis dataset obtained from Kaggle.com by the dataset

 ISSN (Print): 2204-0595
ISSN (Online): 2203-1731

contributor Abhishek Shrivastava [14]. The type of tweets that have been collected in the given dataset is varied. The author's method adds to the dataset's variety of sentiment expressions by classifying tweets with positive emoticons (like:) in the positive category and those with:(in the negative category). There are different characteristics regarding the given dataset: polarity, tweet ID, date and time, query data, author of the tweet, and the content of the tweet. The dataset consists of:

- Tweet Polarity: This field indicates the emotional polarity of the tweet. Zero indicates a bad emotion; two indicates a neutral emotion; and four indicates a positive emotion.
- Tweet ID: A unique identifier for the tweet.
- Tweet Date: The date and time the tweet was published
- Query: This area represents the query or topic related to the tweet. If there is no specific query, the price is NO QUERY. Queries are used to categorize and filter tweets by topic.
- Tweet User: This field represents who created the tweet.
- Tweet Content: This field represents the actual content of the tweet.

b. Dataset 2

The second dataset applied in this term paper is the Twitter tweets sentiment dataset which was sourced from Kaggle by M Yasser H [15]. The characteristics of this dataset include comprehensive tweets on different areas and both positive and negative attitudes which makes sentiment analysis possible. The dataset has four key attributes:

- textID: A unique identifier for each tweet
- text: The full text of the tweet
- selected_text: Identified part of the tweet with an informed positive or negative attitude.
- sentiment: The sentiment label (e.g., positive, negative, neutral) for each tweet

B. Data Processing and Analysis

We choose to illustrate our methods graphically to show the multiple processes used in this section. A diagram will serve as an informative guide, demonstrating the various text preparation and sentiment analysis processes.

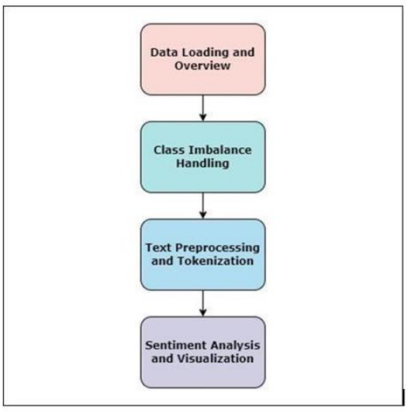

Figure 1. Workflow diagram

a. Dataset 1 Data Processing and Analysis

Firstly, we imported all libraries that were needed for us to process the data from the dataset and made sure that the dataset was imported [16]. Importing libraries include NLTK (Natural Language Toolkit) [17], a comprehensive package for symbolic and statistical natural language processing in English, implemented in Python.

After that, in this data analysis, we represented the plot before data preprocessing was done on tweets. The dataset has many tweets associated with several time stamps that indicate when people disseminated knowledge at different dates or times. To illustrate the quality of the available data, it can be stated that the raw data itself has irregular time stamps because of the formats present in the data set and the fact that some of the attributes contain missing or inaccurate values. This can prevent the exact identification of trends and the temporal analysis that is often necessary in research.

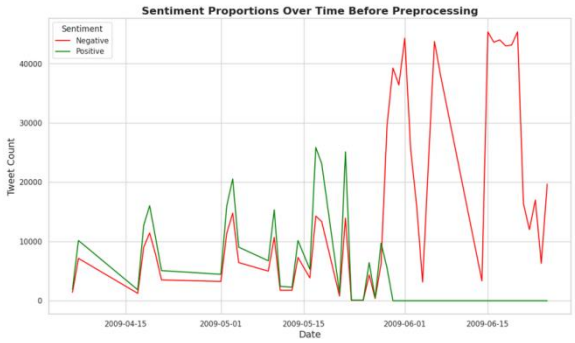

Figure 2. Daily distribution of positive and negative tweets before preprocessing

The following figure shows the daily frequency of the tweets before any form of processing and the classification is done based on the sentiment of the tweets either positive or negative. The vertical lines indicate the days when more tweets were created, and this is probably most likely due to some important event or trending topic. However, the data shown in the figure presents a preprocessed accumulation of basic sentiment tendencies in which positive and negative tweets are divided according to time. However, trends reflected here should be taken with a pinch of salt since the format of the timestamp may not be standardized affecting distribution.

Then we created a dataframe object that included text and sentiment columns and then we counted the occurrences of unique values in the sentiment column. We also made sure that negative sentiments are represented with 0 and positive sentiments are presented with 1. To ensure that both sentiments in the dataset are represented equally when we counted the occurrences of unique values, we saw that negative sentiment is much more represented.

 ISSN (Print): 2204-0595
ISSN (Online): 2203-1731

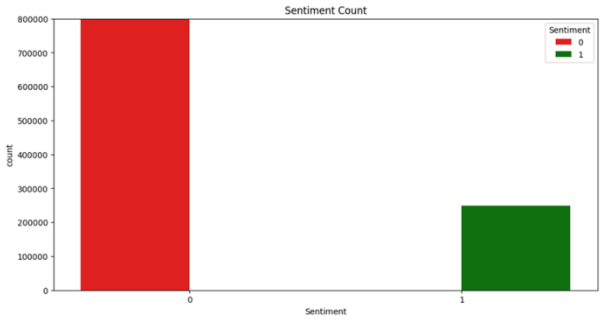

Figure 3. Original representation of sentiments in the dataset.

To deal with the problem of class imbalance [18], we used the downsampling of data which is a typical technique for handling the issues connected with unbalanced classes. Two more dataframes were generated, one for cases where the value at the 'Sentiment' column was 0 in this case for the majority class and another for those instances where the value at the 'Sentiment' column was 1 in the case of the minority class. This was followed by downsampling of the majority class to make the size of the minority class proportional. The original minority class and the resampled majority class were then joined to form the new more representative and balanced dataset.

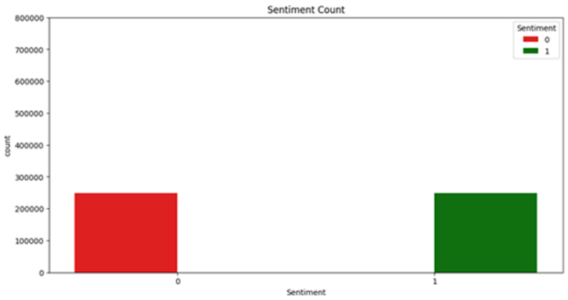

Figure 4. Dataset representation after downsampling

In the preprocessing step here, it was necessary to enhance the raw text data prepared for analysis. First of all, we constituted a list of the English stop words and all available types of punctuation marks to exclude during the process of text cleaning. As a result, in preparing the text corpus, this was followed by converting the text corpus into the list by applying Lancaster Stemmer [19]. After that, for every entry on the dataset, we performed various operations including converting all characters to lowercase, removal of HTML tags, and removal of special characters and numbers. We use the WordNetLemmatizer from the NLTK library [20] to lemmatize every word in the text. Lemmatization implements words to the basic or root word to enhance the analysis as it removes inflected words. This leads to the lemmatization of the words that are stored in the 'text' variable by using list comprehension. As a result, two separate data structures were created: one of which tokenized the text for each document and another included the preprocessed text along with the original 'Sentiment' values. Then using the text data which was modified we analyzed positive sentiments. As the positive text data is already tokenized, we formed a new method to evaluate the number of times each word repeats in it. The function delivered a dictionary that contained the exact count of words. This dictionary was then transformed into a structured data frame where column names represent individual words and column values are the number of occurrences. After that, we arranged the dataframe based on the word count with the help of the desc function for data representation. Speaking of this, it allowed us to represent the ten most frequent words related to positive sentiments.

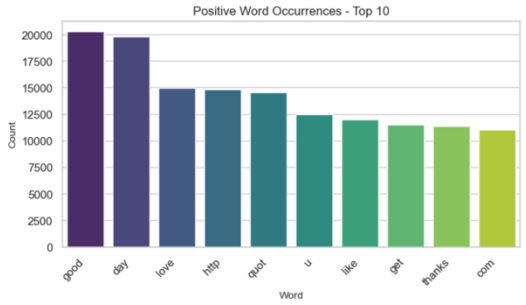

Figure 5. Positive Word Occurrences - Top 10

Figure 5 reveals the top 10 words in positive sentiment with 'good' 'day' 'love' 'http' and 'quot' as the most occurring words in the X posts. These words are considered globally positive words or words with positive connotations and are used throughout the dataset. The same process was followed to represent the dataset's 10 most frequently used words associated with negative emotion.

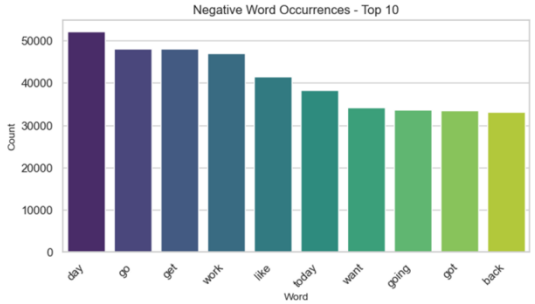

Figure 6. Negative Word Occurrences - Top 10

Figure 6 represents the top 10 most frequently used words in negative sentiments with the details of the most frequently used words being 'day', 'go', 'get', 'work', and 'like'. These words belong to the set of negative words frequently used in the dataset.

After downsampling and preprocessing the final cleaning of the dataset was performed to attain balanced sentiment classes and textual integrity. The frequency of word occurrence in the tweets was also considered and the results show that the more common are the tweets containing four, five, six, or seven words. From these distributions, it is evident that tweets are short and there are sentiments within short word length.

The next figure represents this distribution, as can be seen here that the peaks are in the range of four - seven words. Such revelation about the distribution of the length of the tweets is useful when it comes to conceptualization and

 ISSN (Print): 2204-0595
ISSN (Online): 2203-1731

structuring of the tweets particularly for feature extraction for learning models.

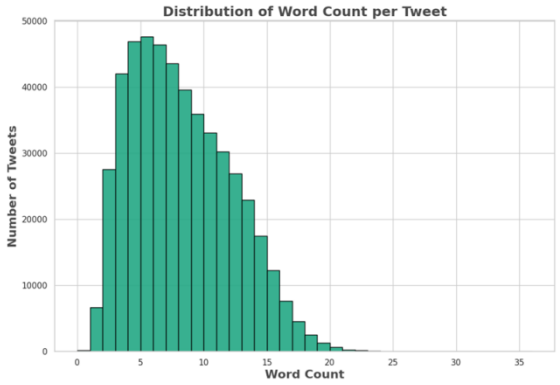

Figure 7. Distribution of word count per tweet after downsampling and preprocessing

Finally, in this analysis stage, the dataset has been preprocessed to ensure we have an equal proportion between positive and negative tweets. Starting with downsampling was done initially to balance between all the sentiments, and get a good representation of all the sentiment classes. After this, preprocessing was done to format the timestamps, manage missing values, and rule out erroneous entries to gain a better understanding of the sentiments in concern.

The plot below shows positive and negative tweets after downsampling and preprocessing, in terms of the distribution of tweets for a day. Once non-sentiment words are removed and the dataset is made more balanced, using the resulting set of words makes distribution more reflective of the sentiment changes over time, which can be useful when trying to work out if there are spikes in activity which could be due to, for example, period of higher activity around certain subtopics or events.

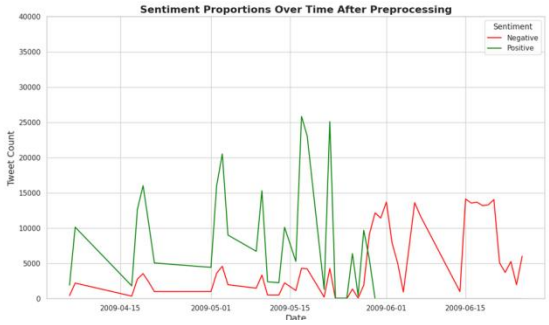

Figure 8. Daily distribution of positive and negative tweets after downsampling and preprocessing

b. Dataset 2 Data Processing and Analysis

Similar to Dataset 1, this was also preprocessed with slight modifications because of the balanced sentiment of data available. The first analysis revealed the same approximate number of positive and negative tweets; therefore, downsampling was not performed. As is seen in the figure below, Dataset 2 is also evenly split between positive and negative sentiments as shown in Figure 9. This balance saved us from having to downsample, and we advanced to preprocessing and tokenization as is.

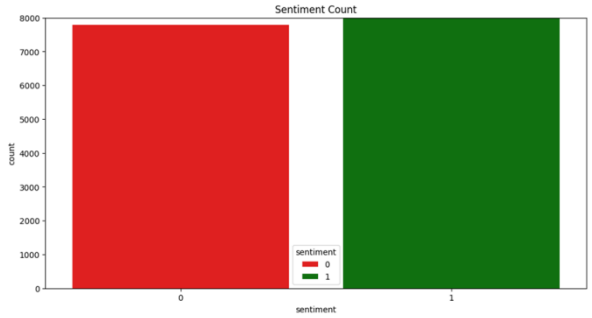

Figure 9. A balanced distribution of positive and negative sentiments in Dataset 2

The preprocessing activities involved; converting the HTML characters, removing the stop words/punctuation, and the tokenization of the text corpus. Once more, stemming and lemmatization approaches were applied to assess the sentiment with the help of only the definite terms. Following the preprocessing phase, the records in Dataset 2 were studied about word count distribution. Similar to Dataset 1, we also found that a large number of tweets have word count values of between 4 and 7. This is depicted in Figure 10 below, showing the relatively impressive trend and the length of the tweets.

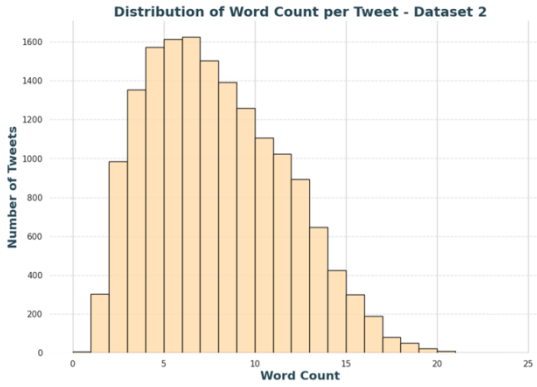

Figure 10. Distribution of word count per tweet in Dataset 2

After preprocessing procedures, we looked at the words most often connected with the positive and negative effects in both datasets. For the next step, we ensured we had the tokenized and cleaned text data and then went further to know the top 10 words that appear per sentiment. For positive sentiments, the most frequent words included: day, good, love, happy mother, thanks, great, u, like, and hope. For negative sentiments, the top words were: like, get, miss, day, go, work, sad, im, really, sorry. Notably, some words were found in two sentiment classes, e.g. of like and day, suggesting that they were used in different contexts across multiple corpora. For this task, we developed a function that serves to count the occurrence of specific words in the positive and negative sentiment partitions. In more detail, this function yielded a dictionary with the precise counts, which we redeemed into a regular structured data frame. In the data frame, the columns

 ISSN (Print): 2204-0595
ISSN (Online): 2203-1731

were individual words and the values referenced the number of times a word appeared. By applying the desc function, it was possible to sort the data frame with the result shown in the subsequent figure that indicates ten most used words in case of each sentiment. Using this approach, it was possible to arrive at a clear representation of most frequent terms with respect to the sentiment class.

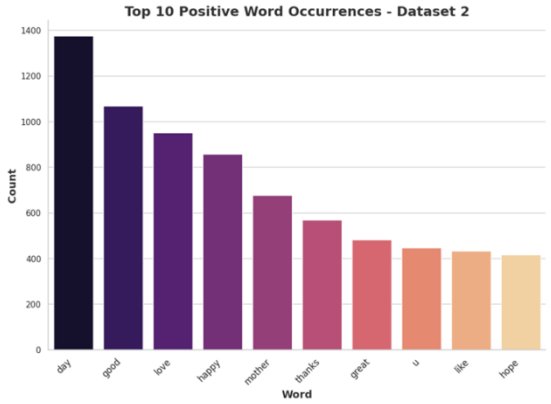

Figure 11. Positive Word Occurrences - Top 10 – Dataset 2

In Figure 11 we can see the top 10 most common word occurrences in positive tweets.

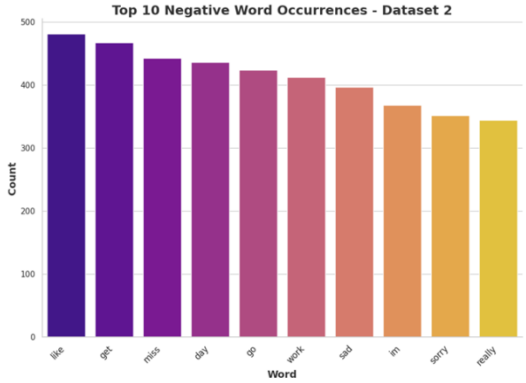

Figure 12. Negative Word Occurrences - Top 10 – Dataset 2

In Figure 12 we can see the top 10 most common word occurrences in negative tweets.

c. Machine Learning Model

First, we preprocessed the tokenized tweets to convert it into a format suitable for feeding into an ML model. The following function is in charge of forming a feature set of each tweet from a list of cleaned tokens. The tweets were then processed separately for positive and negative sentiment to produce tokenized tweets in this model-friendly format. We make entries that consist of a dictionary of features and a sentiment label of the positive as well as the negative sentiments. To ensure an overall equal distribution of positive and negative entries, the entries were randomly mixed.

For feature representation, we used scikit-learn's TF- IDF (Term Frequency - Inverse Document Frequency) Vectorizer [21], which transforms textual data inputs into numerical feature vectors. This vectorization method is used to capture the significance of words in the corpus for the input of the other machine learning model. Last of all, to help the evaluation of the models, the dataset was partitioned into training and testing datasets where 70% were assigned for training and 30% for testing using the TF-IDF vectors and sentiments. Then, for the sentiment analysis, several types of models of machine learning were employed. Some of the models are Naive Bayes, Support Vector Machine (SVM), Stochastic Gradient Descent Classifier, Bernoulli Naive Bayes, Logistic Regression, Multinomial Naive Bayes, Passive Aggressive Classifier, and Perceptron.

Naive Bayes is a probabilistic classifier that implements Bayes' theory and is widely used to analyze sentiment and text categorization [22]. Unlike the NB, SVM is a linear classification model that is effective in high dimensional space where it tries to find the hyperplane that provides the maximum margin between the classes in the feature space [23]. It is fair to conclude that the SGDClassifier is highly scalable and suitable for large datasets since it bases its model parameter updates on stochastic gradient descent that aims at minimizing a loss function [24]. Bernoulli Naive Bayes is a variation of Naive Bayes especially for the models that make use of binary feature vectors [25]. It is especially helpful when features themselves are the reflection of occurrences of a binary nature. Logistic Regression despite the name used for binary classification models a probability of a binary outcome based on predictor variables [26]. There is another type known as Multinomial Naive Bayes which is suitable to be used in a situation that involves multi-class classification and the features that are used are counts of occurrence [27]. Passive Aggressive Classifier is a family of online learning for large-scale classification in terms of streaming or online learning [28]. Last but not least, Perceptron, a linear binary classifier algorithm, learns weights to classify input data correctly, being one of the simplest forms of neural networks [29].

IV. RESULTS

As a preprocess for preprocessing, a preliminary analysis of the raw text data contained in the sentiment dataset was done. This analysis gave the overview of how the provided datasets looks like, a step towards performing adjustments on the data we are given.

	text	Sentiment
1048562	Looking forward to a mini-break in Isle of Wig...	4
1048563	GRINGO STAR tonight. Southern garage. http://...	4
1048564	@DavidBass hee hee...I'll take rain over wind ...	4
1048565	today's message in the church service was deli...	4
1048566	Back home, thought I'd done for the week, but ...	4
1048567	My GrandMa is making Dinenr with my Mum	4
1048568	Mid-morning snack time... A bowl of cheese noo...	4
1048569	@ShaDeLa same here say it like from the Termi...	4
1048570	@DestinyHope92 im great thaanks wbuu?	4
1048571	cant wait til her date this weekend	4

Figure 13. Tweets before preprocessing.

Figure 13 shows the actual raw text data from the sentiment datasets, and includes the following; Every row shows one tweet and includes several columns like sentiment, id, date, query, user, and text. These are raw tweets with several

 ISSN (Print): 2204-0595
ISSN (Online): 2203-1731

characteristics such as mentions, hashtags, and other symbols preserved. As has been outlined in the previous sections of this paper, the raw text data received under some analysis undergoes a change in its format and usefulness for further analysis. Changes carried out included lowercase the text, eliminating all the HTML tags and symbols, and applying Lancaster Stemmer.

	text	Sentiment
497142	looking forward mini break isle wight friend w...	1
497143	gringo star tonight southern garage http www f...	1
497144	davidbass hee hee take rain wind chill day lov...	1
497145	today message church service delivered via sky...	1
497146	back home thought done week call alter somethi...	1
497147	grandma making dinenr mum	1
497148	mid morning snack time bowl cheese noodle yum	1
497149	shadela say like terminiator movie come like word	1
497150	destinyhope im great thaanks wbuu	1
497151	cant wait til date weekend	1

Figure 14. Tweets after preprocessing.

The textual content and sentiment labels post-preprocessing are shown in Figure 14 to reflect the improved text data. Tokenization and other preprocessing steps were the activities of reducing the data set to the bare essentials, which were easier to analyze and made the data more manageable. Once we had preprocessed the textual data, we went to the next step of training a set of machine learning models for sentiment analysis of text. The accuracy of prediction of each model during the training and testing phase is shown in Table 2.

To evaluate the performance of these models, four key metrics were used: Accuracy, which checks the level of correctness of the proposed model [30]; Precision, which checks how many of the instances predicted to be positive are positive [31]; Recall, which checks on how many of the actual positive instances are classified correctly [32]; F1 Score, which offers another single evaluation measure that balances between Precision and Recall, thereby forming an evaluation on false positives as well as false negatives [33]. These metrics make it possible to evaluate the effectiveness of each model in sentiment classification in one way or another.

TABLE II.
ACCURACY MODEL PERFORMANCES

No	Dataset	Dataset 1		Dataset 2	
	Model	Training accuracy	Testing accuracy	Training accuracy	Testing accuracy
1.	Naïve Bayes	86%	77%	95%	86%
2.	SVM	90%	77%	98%	86%
3.	SGD Classifier	77%	76%	95%	87%
4.	Bernoulli Naïve Bayes	84%	77%	94%	86%
5.	Logistic Regression	83%	78%	93%	87%
6.	Multinomial Naïve Bayes	85%	76%	94%	85%
7.	Passive Aggressive Classifier	87%	74%	100%	82%
8.	Perceptron	87%	72%	98%	82%

Table 2 represents the accuracy of the different models that we've used on these two datasets. Accuracy is a metric that tells us how often a machine learning model correctly predicts the outcome.

Dataset 1: During the training set, the SVM model has a high accuracy of about 90% as we can see from Table 2. But this performance improvement did not affect the testing set where the accuracy was 77%. Naive Bayes achieved a training accuracy of 86% but a testing accuracy of 77%. The same is true for Passive Aggressive Classifier and Perceptron which have the corresponding training set accuracies of 87% while the testing set ones were 74% and 72% respectively. Logistic Regression also did good work in training as well as testing where Logistic Regression got 83% training accuracy and 78% testing accuracy. Further, Multinomial Naive Bayes when trained gave 85 % accuracy while the testing accuracy was 76 %. The other two algorithms, that is SGD Classifier and Bernoulli Naive Bayes also depicted a significant variation in the training and the testing accuracies 76% and 77% respectively.

Dataset 2: Models trained on Dataset 2 also showed a higher training accuracy than testing, although it was marginally lower than the values from Dataset 1. For example, SVM and Naïve Bayes attained high training estimates of about 98% and 95% respectively during the testing, their estimates were about 86%. Comparing Logistic Regression and the SGD Classifier, the performances were almost equivalent, the training accuracy of logistic regression was 93%/95% while the testing accuracy was 87%. While overfitting is significantly lower in Dataset 2 than in Dataset 1, some of the models exhibited a performance gap, as presented in Figure 15. Improving the accuracy could include feature selection, and also model regularization that can bring the chances of training and testing closer.

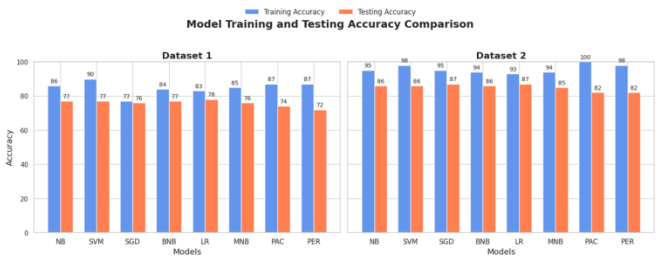

Figure 15. Comparison of training and testing accuracies for machine learning models

Figure 15 shows the training and testing accuracy values for different models used in this analysis. Each model is represented with two bars: one for training, and the other for

 ISSN (Print): 2204-0595
ISSN (Online): 2203-1731

testing. This figure allows a direct comparison of model performance and highlights the need for improvement to increase the ability to generalize unseen data.

TABLE III
PRECISION, RECALL, AND F1-SCORE COMPARISON OF MACHINE LEARNING MODELS
S

No	Dataset	Dataset 1			Dataset 2		
	Model	Precision	Recall	F1-score	Precision	Recall	F1-score
1.	Naïve Bayes	77%	75%	76 %	86%	86%	86 %
2.	SVM	76%	78%	77 %	86%	86%	86 %
3.	SGD Classifier	73%	82%	77 %	87%	87%	87 %
4.	Bernoulli Naïve Bayes	79%	73%	76 %	86%	86%	86 %
5.	Logistic Regression	77%	80%	78 %	87%	87%	87 %
6.	Multinomial Naïve Bayes	78%	73%	75 %	85%	85%	85 %
7.	Passive Aggressive Classifier	75%	71%	73 %	82%	82%	82 %
8.	Perceptron	71%	72%	72 %	82%	82%	82 %

Table 3 presents the Precision, Recall and F1-score metrics for several machine learning models that were applied to two datasets.

Dataset 1: From the evaluation measures we get from the models on Dataset 1, we notice that the Precision, Recall, and F1 scores are slightly variable, meaning that each model has a unique way of measuring sentiment. Logistic Regression provided the best F1-score of 78%, therefore providing an equal measure of precision at 77% and recall at 80% in the future. SVM and SGD Classifier were second, with F1-scores of 77%, The latter had recall of 82% and Precision of 73% which means it was inclined to recognize as many positive samples as possible with less precision (SGD Classifier) needed. In Figure 16 below, we analyze how this performance of each model is preferable in terms of precision, recall, and F1-score in the context of Dataset 1. From these results, it can be concluded that enhancing the model fine-tuning of parameters or feature engineering approaches could enhance the balance between Precision and Recall.

Dataset 2: The models' performance on Dataset 2 is higher and steadier than on Dataset 1, where most of the models got a Precision, Recall, and F1-score of 86% or above. For example, Naïve Bayes and SVM reached almost equal F1-score with 86%. The accuracy score shown here suggests that Logistic Regression and the SGD Classifier performed barely better than others, with an F1-score of 87%, meaning that it could be decided that these models are better for Dataset 2. It has been observed from Figure 16 that the consistency in the performance of all the models in the case of Dataset 2 implies that perhaps this dataset has more unique or significant features for sentiment analysis. Such improvements might involve tuning individual models for particular features to check whether this kind of consistency can be capitalized on to bring about higher Precision and Recall.

The figure below represents a comparison of precision, recall, and F1-score across various machine learning models applied on two datasets.

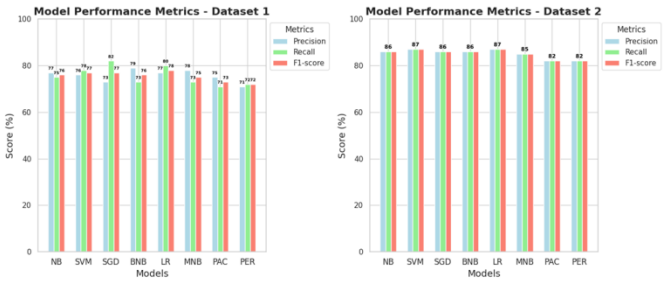

Figure 16. Comparison of precision, recall, and F1-score for machine learning models

As can be seen from Figure 16, many of the models have similar results of precision, reviewing efficiency, and the F1-score in sentiment classification. They each have three bars for easy distinction of their performance on these critical measures. Precision evaluates the quality of a positive prediction made by the model, recall checks the effectiveness of an algorithm that searches for all corresponding instances, and F1-score brings together precision and recall.

V. DISCUSSION AND CONCLUSION

While comparing results from both datasets included in the research, many similarities are present between them. The fact that the first dataset is significantly bigger than the second one should be noted. A sign for the overfitting comes from the fact that the training accuracy of models like for example SVM, Passive Aggressive Classifier, and Perceptron is significantly higher than the test accuracy. Even though the percentage of train and test accuracies are higher in the second dataset, this problem still occurs since the difference between the two accuracies is significant in both cases.

Overfitting was demonstrated with SVM where the decrease in accuracy from the training set to the testing set is also substantial and similar to other studies that raised alarm bells on overfitting. That is, the SGD classifier demonstrates signs of underfitting for the first dataset, while for the second one, the story is a little bit different. However, the second dataset exhibits high accuracy in training (95%) and low accuracy in testing (87%) which signifies that there is overfitting. These findings indicate that the model is possibly oversimplifying regarding the analysis of the sentiment patterns. Because of differences concerning other models' training and testing accuracies, Logistic Regression fared well in both cases with a

 ISSN (Print): 2204-0595
ISSN (Online): 2203-1731

training accuracy of 83% and testing accuracy of 78% in the first dataset, rising up to 93% for training and 87% for testing accuracy in the second sample of data. From these outcomes, we can see that Logistic Regression was able to find a proper balance between the way the training data are managed and the way it is possible to generalize properly to new, previously unknown, data. When we consider our results to prior art we found that the variation is present in the training and testing accuracy of different classifiers. As observed in this study and other related studies, steps should be taken to reduce any possible overfitting, while achieving high accuracy during training in addition to acquiring good generalization capabilities in practice for various datasets.

The results presented here underscore that there have always been, and continue to be, challenges in attaining model stability in applications of sentiment analysis. In conclusion, this term paper aims to discuss the sentiment analysis in which sentiment has been classified for X tweets. As new faces of human interaction, the social media platforms' interaction lies in constant evolution, and therefore, sentiment analysis comes out as an important tool to determine the thinking capability of the communities online. The paper researched several approaches on multiple datasets for sentiment analysis and also developed a literature review that discussed the methodology and findings of prior research in this field.

Regarding the research, The Kaggle datasets were subjected to sentiment analysis to reveal that most of the feelings were negative. Classification imbalance issue was tackled by downsampling and exploratory data analysis was made on occurrences of words belonging to positive and negative situations. For sentiment classification various machine learning models like Naive Bayes, SVM, SGD Classifier, Bernoulli Naive Bayes, Logistic Regression, Multinomial Naive Bayes, Passive Aggressive Classifier, and Perceptron were used. The results highlighted the variations in training and testing performances where the SVM resulted in very high training performance but low testing performance. The result will give rise to concerns regarding overfitting and hence underline the need for model development and algorithm innovation. Balanced performance was observed with the Logistic Regression model, which underscored the significance of perfect model reporting in the specific identification of pattern formation during the training process as well as utilization of model generalization during the testing process. Future research can be done to enrich the subject of sentiment analysis where more approaches and different machine learning algorithms, or models can be introduced.

REFERENCES

[1] Vladić, E., Mehanović, B., Novalić, M., Kečo, D., & Mehanović, Dž. (2024). Sentiment Classification of Tweets using Machine Learning and NLP Techniques. In Proceedings of the NLMLT 2024 Conference. DOI: 10.5121/csit.2024.142004

[2] Orza P., Sentiment Classification: A Beginner's Guide Levity, 2022.

[3] Donges N., Urwin M., Introduction to Natural Language Processing (NLP) Built In, 2023. https://builtin.com/data-science/introduction-nlp. Accessed November 6, 2023.

[4] Gulati K., Kumar S. S., Boddu R. S. K., Sarvakar K., Sharma D. K., Nomani M. Z. M., Comparative Analysis of Machine Learning-Based Classification Models Using Sentiment Classification of Tweets Related to COVID-19 Pandemic. Materials Today: Proceedings, Volume 51, Part 1, 2022, Pages 38-41. ISSN 2214-7853. https://doi.org/10.1016/j.matpr.2021.04.364.(https://www.sciencedirect.com/science/article/pii/S2214 785321032843)

[5] Shoeb M., Ahmed J., Sentiment Analysis and Classification of Tweets Using Data Mining International Research Journal of Engineering and Technology (IRJET), 04(12), 1471, 2017.

[6] Ahmad M., Aftab S., Ali I., Sentiment Analysis of Tweets using SVM International Journal of Computer Applications, 177(5), 25, 2017.

[7] Rustam F., Ashraf I., Mehmood A., Ullah S., Choi G. S., Tweets Classification on the Base of Sentiments for US Airline Companies Entropy, 21(11), 1071, 2019. doi:10.3390/e21111071.

[8] Celiktug M. F., Twitter Sentiment Analysis, 3-Way Classification: Positive, Negative or Neutral? 2018 IEEE International Conference on Big Data (Big Data), Bilkent University, Gazi University, Ankara, Turkey, 2018.

[9] Go A., Bhayani R., Huang L., Twitter sentiment classification using distant supervision. CS224N project report, Stanford, 1(12), 2009.

[10] Heikal, M., Torki, M., & El-Makky, N. (2018). Sentiment analysis of Arabic tweets using deep learning. Procedia Computer Science, 142, 114–122. https://doi.org/10.1016/j.procs.2018.07.020

[11] Harjule, P., Gurjar, A., Seth, H., & Thakur, P. (2020). Sentiment analysis of Twitter data using machine learning algorithms. International Journal of Computer Applications, 975(14), 1-6.

[12] Baid, P., Gupta, A., & Chaplot, N. (2017). Sentiment Analysis of Movie Reviews using Machine Learning Techniques. International Journal of Computer Applications, 179(7), 45-49.

[13] Sarlan A., Nadam C., Basri S., Twitter sentiment analysis. In Proceedings of the 6th International Conference on Information Technology and Multimedia, pp. 212–216. IEEE, November 2014.

[14] Shrivastava A.,Sentiment Analysis Dataset. Kaggle. https://www.kaggle.com/datasets/abhi8923shriv/sentiment-analysis-dataset/data, 2020. Accessed November 11, 2023

[15] M. Yasser H. (2020). Twitter Tweets Sentiment Dataset. Kaggle. Retrieved from https://www.kaggle.com/datasets/yasserh/twitter-tweets-sentiment-dataset/data. Accessed April 14, 2024

[16] Islam Sujan N., Top 5 Machine Learning Libraries in Python. Towards Data Science. https://towardsdatascience.com/top-5-machine-learning-libraries-in-python-e36e3e0e02af, 2018. Accessed January 12, 2024.

[17] Natural Language Toolkit. https://www.nltk.org/, 2023. Accessed January 18, 2024.

[18] [18] Google for Developers. Imbalanced Data. https://developers.google. com/machine-learning/data-prep/construct/sampling-splitting/ imbalanced-data, 2023. Accessed January 21, 2024.

[19] [19] Natural Language Toolkit. Lancaster Stemmer. NLTK Documentation. https://www.nltk.org/api/nltk.stem.lancaster.html, 2023. Accessed February 4, 2024.

[20] [20] Natural Language Toolkit (nltk).nltk.stem.wordnet Module. NLTK Project. https://www.nltk.org/_modules/nltk/stem/wordnet.html, 2023. Accessed February 6, 2024.

[21] scikit-learn. TfidfVectorizer. In scikit-learn Documentation. https://scikit- learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html, 2023. Accessed February 11, 2024.

[22] Rahat A. M., Kahir A., Masum A. K. M., Comparison of Naive Bayes and SVM Algorithm based on sentiment analysis using review dataset. In 2019 8th International Conference System Modeling and Advancement in Research Trends (SMART), pp. 266–270. IEEE, November 2019.

[23] DataTechNotes. Classification Example with Linear SVC in Python. Retrieved from https://www.datatechnotes.com/2020/07/ classification-example-with-linearsvm-in-python.html, 2020. Accessed March 8, 2024.

[24] Varshney C. J., Sharma A., Yadav D. P., Sentiment analysis using ensemble classification technique. In 2020 IEEE Students Conference on Engineering & Systems (SCES), pp. 1–6. IEEE, July 2020.

[25] Johnson M., Bernoulli Naive Bayes Classifier. Retrieved from https://mattshomepage.com/articles/2016/Jun/07/bernoulli_nb/, 2016. Accessed March 18, 2024.

 ISSN (Print): 2204-0595
ISSN (Online): 2203-1731

[26] Peng C. Y. J., Lee K. L., Ingersoll G. M., An introduction to logistic regression analysis and reporting. The Journal of Educational Research, 96(1), 3–14, 2002.
[27] Abbas M., Memon K. A., Jamali A. A., Memon S., Ahmed A., Multinomial Naive Bayes classification model for sentiment analysis. IJCSNS Int. J. Comput. Sci. Netw. Secur, 19(3), 62, 2019.
[28] Sriram, Passive Aggressive Algorithm — For big data models. Geek Culture. Retrieved from https://medium.com/geekculture/passive-aggressive-algorithm-for-big-data-models-8cd535ceb2e6, 2021. Accessed March 23, 2024.
[29] Bhardwaj A., What is a Perceptron? – Basics of Neural Networks: An overview of the history of perceptrons and how they work. Towards Data Science. Retrieved from https://towardsdatascience.com/ what-is-a-perceptron-basics-of-neural-networks-c4cfea20c590, 2020. Accessed March 24, 2024.
[30] scikit-learn. 2024. Sklearn.metrics.accuracy_score. Retrieved from https://scikit-learn.org/stable/modules/generated/sklearn.metrics.accuracy_score.html. (2024). Accessed August 25, 2024.
[31] scikit-learn. 2024. Sklearn.metrics.precision_score. Retrieved from https://scikit-learn.org/stable/modules/generated/sklearn.metrics.precision_score.html. (2024). Accessed August 26, 2024.
[32] cikit-learn. 2024. Sklearn.metrics.recall_score. Retrieved from https://scikit-learn.org/stable/modules/generated/sklearn.metrics.recall_score.html. (2024). Accessed August 26, 2024.
[33] scikit-learn. 2024. Sklearn.metrics.f1_score. Retrieved from https://scikit-learn.org/stable/modules/generated/sklearn.metrics.f1_score.html. (2024). Accessed August 27, 2024.

 ISSN (Print): 2204-0595
ISSN (Online): 2203-1731