SENTIMENT CLASSIFICATION OF TWEETS USING MACHINE LEARNING AND NLP TECHNIQUES

Ervin Vladić[1], Benjamin Mehanović[1], Mirza Novalić[1], Dino Kečo[1] and Dželila Mehanović[1]

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

ABSTRACT

In the last ten years, social networks have appeared as main opinion-sharing and discussion-enabling resources. At the same time, the development of machine learning (ML) and natural language processing (NLP) technologies has allowed for new approaches to analyzing the huge quantities of data created by users. 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. 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 while Logistic Regression showed 83% for training and 78% for testing.

KEYWORDS

Social Media Analysis, Classification Models, Data Preprocessing

1. INTRODUCTION

At present when technology is advancing at a rapid pace, the sites such as the social networking sites have become a place wherein thoughts opinions, as well as 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 was difficult 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.

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 [1]. 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 business, researchers, or even individuals have started using sentiment analysis in order to get more insights about people's opinions or reactions towards their products or services, or even the trends in the market. 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 [2]. Classification of X-tweets as bearers of sentiments is an area situated at the crossroads of

David C. Wyld et al. (Eds): NLMLT, SEAU– 2024
pp. 33-45, 2024. - CS & IT - CSCP 2024 DOI: 10.5121/csit.2024.142004

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.

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 dataset 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.

2. 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 [3]. 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 crossed 98%. However, the AdaBoost Classifier only fared worse achieving nearly about 73% when it came to classification accuracy.

Another research [4] was related to the 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 [5], 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 [6] 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 [7] 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 [8] 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.

Lastly, the study titled "Twitter Sentiment Analysis" proved the necessity of proper sentiment analysis on 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.

3. 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 dataset; the second section: Data Processing and Analysis which outlines the preprocessing of the data, 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.

3.1. Dataset

The dataset used in this term paper was the sentiment analysis dataset obtained from Kaggle.com by the dataset contributor Abhishek Shrivastava [10]. 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.

3.2. 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.

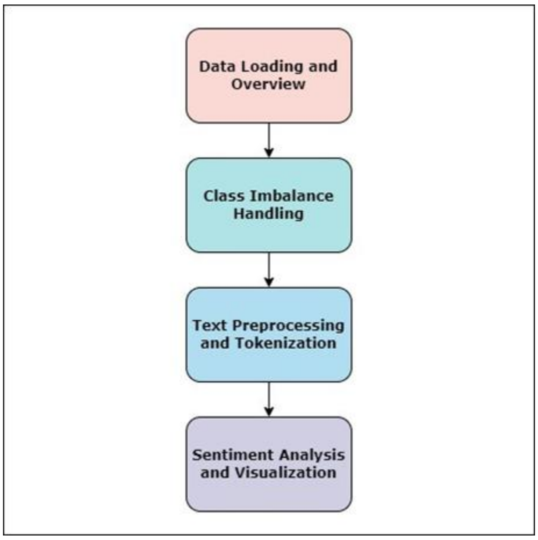

Figure 1. Workflow diagram

Firstly, we imported all libraries that were needed for us to process the data from the dataset and made sure that the dataset is imported [11]. Importing libraries include NLTK (Natural Language Toolkit) [12], a comprehensive package for symbolic and statistical natural language processing in English, implemented in Python. 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.

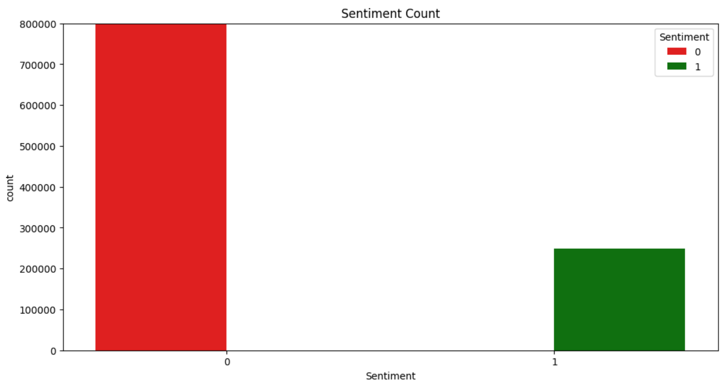

Figure 2. Original representation of sentiments in the dataset.

To deal with the problem of class imbalance [13], 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.

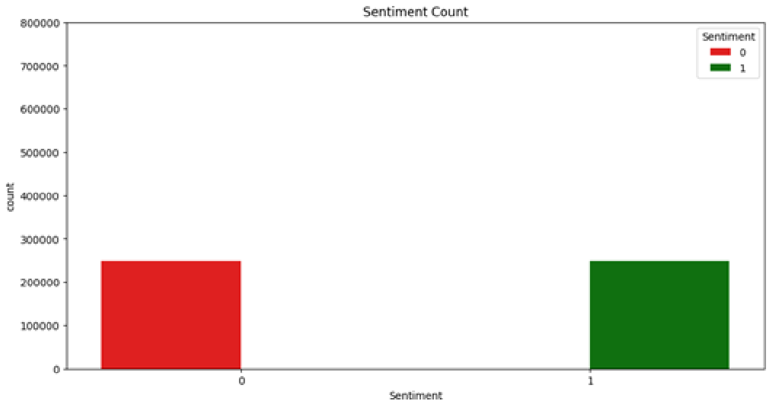

Figure 3. Dataset representation after downsampling.

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 [14]. After that, for every entry on the dataset, we performed various operations including converting all characters to lowercase, removal of HTML tags, removal of special characters and numbers. We use the WordNetLemmatizer from the NLTK library [15] to lemmatize every word in the text. Lemmatization implements words to the basic or root word so as 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 made an analysis of 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.

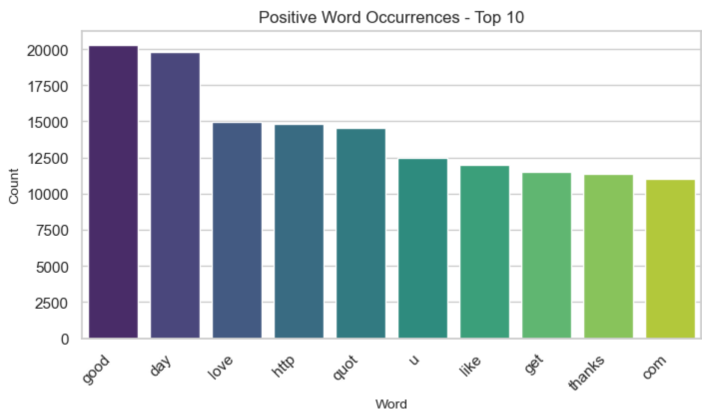

Figure 4. Positive Word Occurrences - Top 10.

Figure 4 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.

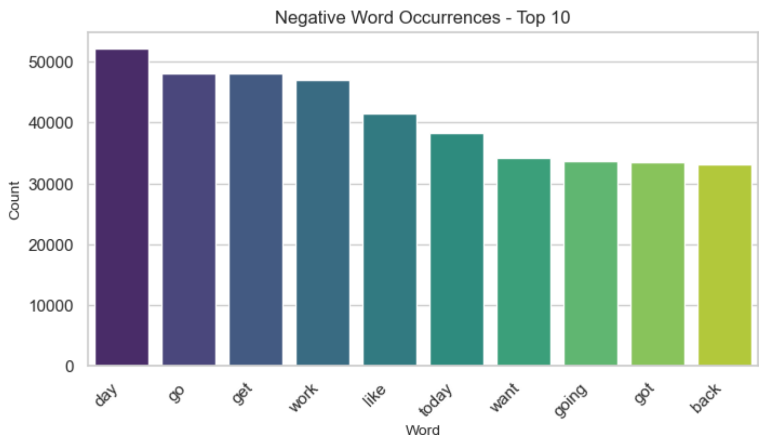

Figure 5. Negative Word Occurrences - Top 10.

Figure 5 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.

3.3. 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 consisted of a dictionary of features and a sentiment label of the positive as wells 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 [16], which transforms textual data inputs into numerical feature vectors. This vectorization method is used it 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 [17]. Unlike the NB, SVM is a linear classification model which 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 [18]. It is fair to conclude that the SGD Classifier 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 [19]. Bernoulli Naive Bayes is a variation of Naive Bayes especially for the models that make use of binary feature vectors [20]. 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 that models a probability of a binary outcome based on predictor variables [21]. 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 [22]. Passive Aggressive Classifier is a family of online learning for large-scale classification in terms of streaming or online learning [23]. 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 [24].

4. 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 dataset 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 6. Tweets before preprocessing.

Figure 6 shown the actual raw text data from the sentiment dataset, 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 a number of characteristics such as mentions, hashtags, and other symbols are 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 7. Tweets after preprocessing.

The textual content and sentiment labels post-preprocessing are shown in Figure 7 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 1.

To evaluate the performance of these models, four key metrics were used: Accuracy, which checks the level of correctness of the proposed model [25]; Precision, which checks how many of the instances predicted to be positive are positive [26]; Recall, which checks on how many of the actual positive instances are classified correctly [27]; 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 [28]. These metrics make it possible to evaluate the effectiveness of each model in sentiment classification in one way or another.

Table 1. Accuracy Model Performances.

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

During the training set, the SVM model has a high accuracy of about 90% as we can see from Table 1. 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.

Table 2. Precision, Recall, and F1-Score Comparison of Machine Learning Models

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

Performance metrics of several machine learning models employed in sentiment analysis are shown in Table 2 using the Precision, Recall, and F1-score metrics. As for the evaluation of the features, the F1-score of Logistic Regression reached its highest level of 78% with a Precision of 77% and a Recall of 80%. SVM and SGD Classifier also had high accuracy with the accuracy of both models at 77% F1-score was also 77%. These measures give us an idea of how well each model can perform sentiment analysis by giving a comparative idea of each model's performance.

5. DISCUSSION AND CONCLUSION

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. 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. 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 with a training accuracy of 83% while the testing accuracy was at 78%. 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 practical 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, the sentiment analysis comes out as an important tool to determine the thinking capability of the communities online. The paper researched several approaches for sentiment analysis and also developed a literature review that discussed upon the methodology and findings of prior research in this field. Regarding the research, The Kaggle dataset was 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 into very high training performance but low testing performance. The result will give rise to concerns regarding overfitting and hence underlined 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] Orza P., Sentiment Classification: A Beginner's Guide Levity, 2022.

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

[3] 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/S2214785321032843)

[4] 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.

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

[6] 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.

[7] 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.

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

[9] 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.

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

[11] 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.

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

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

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

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

[16] 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.

[17] 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.

[18] 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.

[19] 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.

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

[21] 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.

[22] 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.

[23] 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.

[24] 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.

[25] 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.

[26] 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.

[27] scikit-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.

[28] 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.