Social Media Sentiment Analysis

Indian Institute of Information Technology, Allahabad

Radhika Gupta IIT2021125 | Naman Gupta IIT2021139 | Naysha Singh IIT2021140 | Kaushik Mullick IIT2021168 | Viraj Jagtap IIT2021170 | Shruti Bilolikar IIT2021172

***Abstract*—The project focuses on harnessing the vast volume of daily-generated tweets, totaling 1,600,000, as a crucial source for understanding group opinions through sentiment analysis. Employing a multifaceted approach, the dataset undergoes preprocessing, descriptive analysis, and predictive modeling using six machine learning models derived from prominent algorithms, including TF-IDF and Countvectorized Naive Bayes, Support Vector Machine, and Logistic Regression. The subsequent application of a bagging technique consolidates the results of these models, enhancing the robustness of sentiment predictions. This comprehensive methodology contributes to a nuanced comprehension of sentiments within the dynamic realm of Twitter conversations, aligning with the overarching objective of extracting valuable insights from large-scale social media datasets.**

***Index Terms*—Sentiment analysis, NLTK**

I. Introduction

Sentiment analysis, a sub-field of Natural Language Processing, is one of the most popular topics and research fields in data science. We will be working on social media sentiment analysis. We aim to be able to classify tweets, reviews and comments from social media as positive or negative.

The most important point of our project is data mining to collect a large amount of data from several sources. For this purpose, we found open source datasets such as Sentiment140 [1] and many others. After all the searching we decided to use the Sentiment140.

Most of the open-source datasets that we found on the internet are properly labeled and structured. Data collected by ourselves needs to be properly labeled. Then, we will go through the cleaning, preprocessing and separation of test and training data steps.

We searched for some tools for our project and found some popular and powerful open-source NLP frameworks in Python. We will probably use the Natural Language Toolkit (NLTK) [2]. It comes with all the pieces you need to get started on sentiment analysis.

II. Problem Statement

This project addresses the development of a sophisticated machine learning model tailored for sentiment analysis, with a primary focus on binary classification into positive and negative classes. The objective is to create an adept system capable of discerning nuanced sentiment patterns within diverse textual data sources, encompassing social media posts, customer reviews, and news articles. The model aims to achieve high precision, recall, and accuracy, thereby enhancing its applicability across various linguistic nuances and domains.

III. Literature review

A. Literature Review I

The research paper titled "Real Time Sentiment Analysis of Tweets using Naive Bayes" authored by Ankur Goel, Jyoti Gautam, and Sitesh Kumar and published on October 16, 2016, presents a comprehensive approach to real-time sentiment analysis of tweets on Twitter using Naive Bayes and SentiWordNet. It begins by discussing the process of gathering training data, which involves collecting a relevant corpus of tweets. The training data is then preprocessed to prepare it for classification using the Naive Bayes classifier. The paper also introduces a new classification technique that utilizes SentiWordNet to enhance efficiency in sentiment analysis.

The proposed workflow for sentiment analysis involves several key steps. First, the system is trained using the collected corpus of tweets. Then, the testing corpus of tweets is tokenized and preprocessed. Next, the posterior probability and SentiWordNet score of each tokenized word are determined. Finally, the class conditional probability is computed using the Naive Bayes classifier.

The paper provides examples and results of the sentiment analysis, shedding light on the effectiveness of the approach. It also delves into the use of emoticons and hashtags in tweets, highlighting their impact on sentiment analysis.

In addition to the technical details, the paper discusses the future scope and applications of sentiment analysis of tweets, emphasizing the growing relevance of real-time sentiment analysis in various domains. The conclusion summarizes the findings and contributions of the paper, underlining the significance of the proposed approach in enabling efficient and accurate sentiment analysis of tweets. Furthermore, the use of Python and NLTK for natural language processing is mentioned, showcasing the practical implementation of the proposed techniques.

Overall, the paper offers a valuable contribution to the field of sentiment analysis by presenting a robust methodology for real-time sentiment analysis of tweets, leveraging the power of Naive Bayes and SentiWordNet.

B. Literature Review II

The Research Paper titled "An Ensemble Classification System for Twitter Sentiment Analysis" authored by Ankit and Nabizath Saleena published in year 2018 delves into realm of Twitter, a dynamic online platform, allows users

to craft, post, and engage with succinct text snippets known as tweets. These tweets serve as avenues for individuals to express their opinions, thoughts, and viewpoints on various subjects. Sentiment Analysis (SA) emerges as a vital tool for discerning and categorizing the polarity of text, spanning documents, sentences, and phrases.

This paper introduces a weighted ensemble classifier specifically designed for tweet sentiment analysis. Experimental results demonstrate the superior performance of the proposed ensemble classifier compared to standalone classifiers and majority voting ensembles.

In the context of sentiment analysis, various base classifiers were employed to discern sentiments from tweets. The Naive Bayes classifier operates on probability computations, assuming conditional independence of features, whereas the Random Forest, functioning as an ensemble of decision trees, aggregates votes from multiple trees to determine the final label. Support Vector Machine (SVM) seeks to establish a decision boundary maximizing the gap between classes, without assuming class conditional independence. Logistic Regression, a regression-based classification model, aims to find a hyperplane for optimal class separation. These base classifiers were integrated into an ensemble classifier, leveraging their collective strengths to bolster performance.

An ensemble algorithm was proposed to calculate the sentiment score of tweets. It determines positive and negative scores for tweets by considering predictions from each classifier. An additional algorithm predicts tweet sentiments by assessing the positive and negative scores. In cases where scores are equal, it utilizes cosine similarity to identify the most similar tweet in the testing data and derives sentiments based on similarity calculations.

In the evaluation using the Sentiment140 dataset, multiple classification models were tested for sentiment analysis. Naive Bayes showcased a reasonable accuracy of 75.19%, while the Random Forest classifier yielded a slightly lower performance with 71.76% accuracy. On the other hand, the Support Vector Machine exhibited robustness, achieving an accuracy of 75.61%. Logistic Regression followed closely with an accuracy of 74.15%. Employing Majority Voting enhanced results marginally to 74.80%. However, the most notable performance came from the proposed ensemble method, surpassing all individual classifiers with an accuracy of 75.81%. This outcome underscores the ensemble method's superiority in sentiment analysis, showcasing its potential for more accurate and reliable sentiment classification on the Sentiment140 dataset.

IV. Proposed Methodology

The initial phase of our methodology involves a meticulous preprocessing of the dataset extracted using the Twitter API, encompassing 1,600,000 tweets classified on a sentiment scale ranging from 0 (negative) to 4 (positive). The dataset comprises six fields, namely target (integer), ids (integer), date (date), flag (string), user (string), and text (string). However, to streamline our analysis, four non-essential fields were eliminated, resulting in a refined dataset featuring only two pertinent fields: 'label' and 'tweet,'. Following this curation, any missing values within the dataset were systematically removed, resulting in a dataset distribution reflective of the preprocessing transformations. Concurrently, an embedding matrix was established, both of which are integral components for subsequent stages of the analysis. This strategic adoption of TF-IDF and Countvectorizer aligns with our pursuit of efficient and effective word representations, circumventing the computationally intensive process of training embeddings from scratch. We employed two distinct vectorization techniques, namely CountVectorizer and TF-IDF Vectorizer, as preparatory steps for subsequent model application. Specifically, for the logistic regression model, we applied CountVectorizer with the incorporation of both unigram and bigram representations. The application of TF-IDF weighting aimed to emphasize terms that were discriminative across the entire document collection, thereby enhancing the model's ability to discern relevant patterns in the data.These vectorization strategies, tailored to the specific requirements of each model, were integral to the overall feature engineering process, contributing to the optimization of model performance within the context of our experimentation.In the final phase of our modeling framework, we amalgamated the outcomes of the diverse models, namely Naive Bayes, Support Vector Machine (SVM), and Logistic Regression, each utilizing both CountVectorizer and TF-IDF Vectorizer. These six models collectively served as the foundation for the implementation of a Bagging Classifier.The Bagging Classifier, an ensemble learning technique, systematically combined predictions from the individual base models to generate a more robust and accurate aggregate prediction.

V. Experiment Setup

The initial phase of our analytical methodology involves a comprehensive preprocessing of the dataset, incorporating data reduction techniques and the removal of stop words and punctuation from all instances. Subsequently, a detailed analysis ensued, encompassing an examination of letter frequencies, the distribution of letters relative to expected English language frequencies via chi-square tests, word frequencies, and statistical parameters such as maximum, minimum, and standard deviation.

Furthermore, an in-depth exploration into the most common words within the positive and negative classes was conducted. Feature extraction methodologies, specifically bag-of-words and word embedding, were then employed. The former, utilizing TF-IDF, represents a conventional and straightforward approach to feature extraction. The correlation of words within the corpus was established and analyzed through this method.

In the subsequent phase, classification and regression experiments were undertaken, with a designated test set percentage of 20. Various models were applied, including SVM Model with TF-IDF vectorizer, SVM Model with CountVectorizer, Logistic Regression on N-gram with CountVectorizer, Logistic Regression on N-gram with TF-IDF Vectorizer, Multinomial

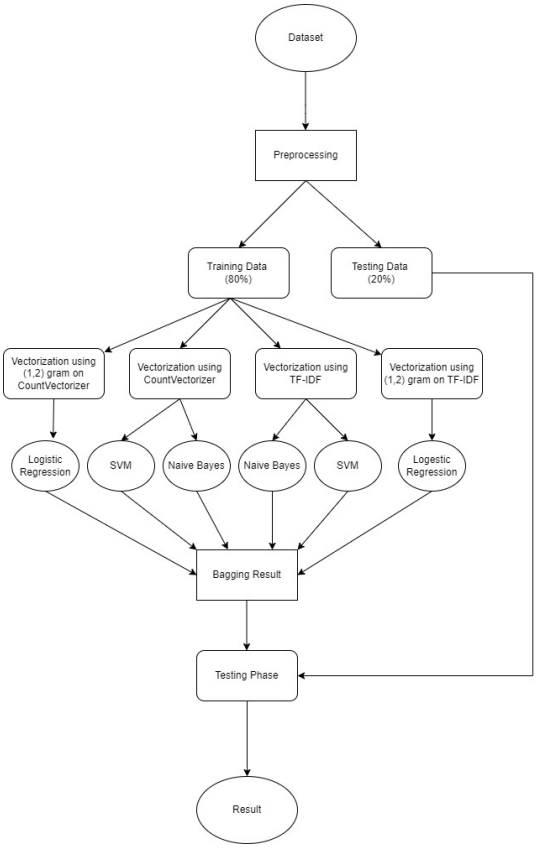

Fig. 1. Flowchart for proposed methodology

Naive Bayes Model-1 with CountVectorizer, and Multinomial Naive Bayes with TF-IDF Vectorizer.

Following individual model assessments, a bagging approach was applied to the results of the aforementioned models. The final conclusions were drawn based on this aggregated ensemble of models. Model evaluation criteria included precision, recall, and F1-score, providing a robust and multifaceted assessment of the models' predictive performance.

VI. ANALYSIS OF PROPOSED MODEL PERFORMANCE

A. Preprocessing

During the project's preprocessing phase, we conducted a comprehensive analysis of the dataset in terms of letter and word frequencies.

Firstly, we examined letter counts across the dataset, analyzing the frequency and relative frequency of each letter. We utilized the chi-square test to evaluate whether the distribution of letters aligns with typical English texts. Despite a notably high Pearson correlation (96.7%), the resulting p-value was 0, indicating a significant deviation from the expected letter frequency distribution in English.

Additionally, we investigated the character counts within each tweet. This analysis provided insights into the maximum, minimum, mean, and standard deviation of character lengths. The longest tweet comprised 189 characters, while the shortest contained only 1 character. On average, tweets were approximately 42.78 characters long, with a standard deviation of 24.16.

Moving on, we also assessed the word counts per tweet. Similar to the character analysis, we explored the maximum, minimum, mean, and standard deviation of word lengths. The lengthiest tweet consisted of 50 words, while the shortest tweet contained only 1 word. On average, tweets comprised approximately 7.24 words, with a standard deviation of 4.03.

Furthermore, we conducted an analysis of the most frequently occurring words within the positive and negative classes. This analysis aimed to identify common words associated with both sentiment categories, providing valuable insights into the language utilized in each class.

B. Predictive Analysis

nitially, our dataset comprised six features: target, id, date, query, user, and text. For our analysis, we narrowed our focus to two key features: target and text. This selection led to a notable decrease in entropy, signifying increased data organization and information concentration.

Specifically, the entropy reduction following this transformation was substantial:

Initial entropy of the dataset: 41.082

Entropy after preprocessing: 14.733

In our classification/regression experiments, we partitioned the dataset into training and testing sets, allocating 20% of the data to the test set. We evaluated four distinct models: Including SVM Model with TF-IDF vectorizer, SVM Model with CountVectorizer, Logistic Regression on N-gram with CountVectorizer, Logistic Regression on N-gram with TF-IDF Vectorizer, Multinomial Naive Bayes Model-1 with CountVectorizer, and Multinomial Naive Bayes with TF-IDF Vectorizer. Ultimately, to leverage the collective insights from these diverse models, we employed a bagging technique to derive a combined result from the outputs of the four models used.

In our extensive model training phase, we explored various models and techniques, aiming to derive optimal predictive performance. The accuracies achieved across different models are as follows:

CountVectorized Naive Bayes: Achieved an accuracy of 76.08%.

TF-IDF Naive Bayes: Yielded an accuracy of 75.42%.

TF-IDF SVM: Demonstrated an accuracy of 76.77%.

CountVectorized SVM: Attained an accuracy of 76.31%.

Logistic Regression with CountVectorized N-gram: Showcased an accuracy of 78.58%.

Logistic Regression with TF-IDF N-gram: Displayed the highest accuracy among individual models, reaching 79.13%.

Furthermore, to amalgamate the predictive power of these diverse models, we employed a technique called bagging. This amalgamated result from the four models yielded an accuracy of 77.73%.

TABLE I
RESULTS

Model	Accuracy
Multinomial Naive Bayes Model-1	76.08%
Multinomial Naive Bayes Model-1	75.42%
Support Vector Machine Model-1	76.31%
Support Vector Machine Model-1	76.77%
Logistic Regression Model-1	78.58%
Logistic Regression Model-2	79.13%
Bagging classifier	77.73%

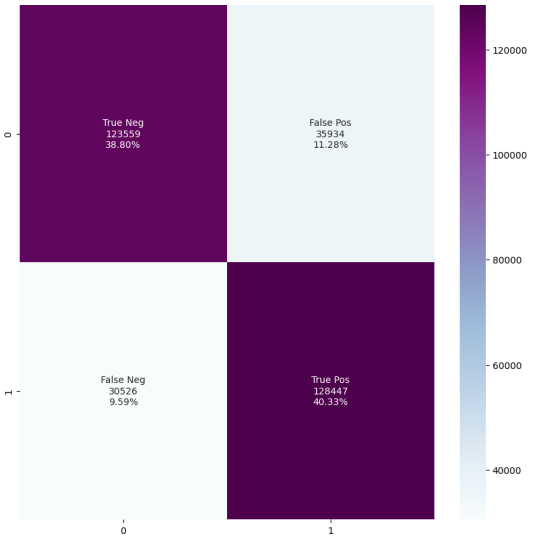

Fig. 2. Confusion matrix for Logistic Regression with TF-IDF vectorizer which gives the highest accuracy. (Accuracy:79.13%)

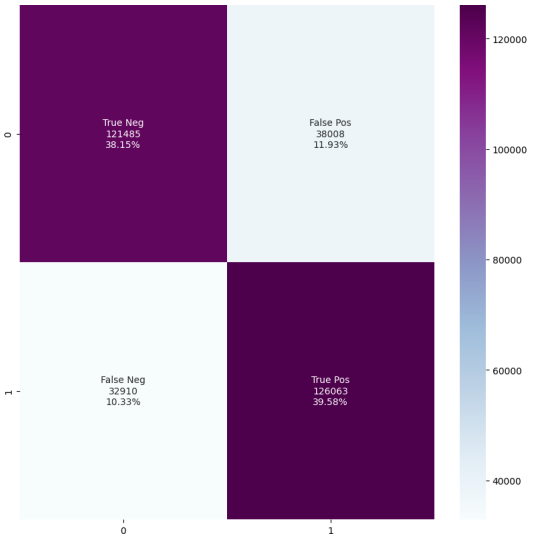

Fig. 3. Confusion matrix for Bagging Classifier (Accuracy:77.73%)

```
Bagging Accuracy: 0.7773137477784128
Bagging Classification Report:
              precision    recall  f1-score   support

    Negative       0.79      0.76      0.77    159493
    Positive       0.77      0.79      0.78    158973

    accuracy                           0.78    318466
   macro avg       0.78      0.78      0.78    318466
weighted avg       0.78      0.78      0.78    318466
```

Fig. 4. Classification report for Bagging Classifier

VII. CONCLUSION AND FUTURE WORK

The initial examination of our raw dataset revealed the presence of extraneous features not germane to our analytical objectives, with an initial entropy value of 41.08. Subsequent to the removal of these unnecessary columns and the deletion of rows with empty values, a refined dataset was obtained, resulting in a significantly reduced entropy value of 14.73. This preprocessing step underscored a substantial alteration in the dataset's information content. Notably, Naive Bayes models exhibited superior training time durations, demonstrating commendable computational efficiency compared to alternative models. The discernible speed advantage of Naive Bayes models positions them as an optimal choice for our analytical purposes within the refined dataset. Additionally, employing a bagging classifier yielded an overall accuracy of 77.73, with the highest accuracy of 79.18 achieved by logistic regression with TF-IDF vectorizer, further affirming the effectiveness of our refined dataset and modeling approach.

Future improvements in sentiment analysis could involve the integration of advanced tone detection algorithms for discerning subtle nuances in language, thereby accurately identifying optimistic or pessimistic tones. Additionally, the development of models capable of understanding the sentiment conveyed by emojis would enhance overall accuracy, given their widespread use in digital communication. Addressing challenges in classifying neutral statements versus comparative sentiments is also crucial, and implementing advanced algorithms with heightened contextual comprehension capabilities could significantly improve classification accuracy. These enhancements collectively contribute to a more nuanced and sophisticated sentiment analysis framework.

REFERENCES

[1] Ankit, Nabizath Saleena, An Ensemble Classification System for Twitter Sentiment Analysis, Procedia Computer Science, Volume 132, 2018, Pages 937-946, ISSN 1877-0509, https://doi.org/10.1016/j.procs.2018.05.109.

[2] Satya Abdul Halim Bahtiar, Chandra Kusuma Dewa, Ahmad Luthfi, Comparison of Naïve Bayes and Logistic Regression in Sentiment Analysis on Marketplace Reviews Using Rating-Based Labeling.Vol. 5, No. 3, September 2023 e-ISSN: 2656-4882 p-ISSN: 2656-5935, https://doi.org/10.51519/journalisi.v5i3.539

[3] A. Goel, J. Gautam and S. Kumar, "Real time sentiment analysis of tweets using Naive Bayes," 2016 2nd International Conference on Next Generation Computing Technologies (NGCT), Dehradun, India, 2016, pp. 257-261, doi: 10.1109/NGCT.2016.7877424.

Model	Accuracy	Positive class			Negative class		
		Precision	Recall	F1-score	Precision	Recall	F1-score
Multinomial naive bayes model-1	0.76	0.76	0.75	0.76	0.76	0.77	0.76
Multinomial naive bayes model-2	0.75	0.76	0.75	0.75	0.75	0.76	0.76
SVM Model model-1	0.76	0.75	0.79	0.77	0.78	0.74	0.76
SVM Model model-2	0.77	0.78	0.74	0.76	0.75	0.8	0.77
Logistic Regression model-1	0.79	0.77	0.81	0.79	0.8	0.76	0.78
Logistic Regression model-2	0.79	0.78	0.81	0.79	0.8	0.77	0.79
Bagging Classifier	0.78	0.77	0.79	0.78	0.79	0.76	0.77
Naive Bayes	0.75	0.75	0.74	0.75	0.74	0.75	0.75
Random Forest	0.71	0.67	0.83	0.74	0.78	0.6	0.68
Support Vector Machine	0.75	0.73	0.79	0.76	0.77	0.72	0.74
Logistic Regression	0.74	0.72	0.78	0.75	0.76	0.69	0.72
Majority Voting	0.74	0.71	0.82	0.76	0.79	0.66	0.72

Fig. 5. Comparison Table of Base Research Paper and Proposed Model

[4] Sentiment140, http://help.sentiment140.com/home
[5] Natural Language Toolkit, https://www.nltk.org/