An Efficient Machine Learning Approach to Detect Sentiments from Text Data

NAZMUL KARIM TANVIR ID:23166011*, BRAC University, Bangladesh
TASNIM FUYARA CHHOAN ID:23366035, BRAC University, Bangladesh
SAMIN YASAR ID:23273007, BRAC University, Bangladesh
A.B.M.TOUFIQUL ALAM SAMS ID:24166027, BRAC University, Bangladesh

Sentiment analysis, a leading-edge deep learning technology, delves into individuals' genuine emotions through facial speech, text, expressions, and gestures, pivotal for understanding reactions in diverse circumstances. This study aims to bolster sentiment analysis efficiency by deploying machine learning algorithms like Naive Bayes, achieving higher accuracy rates. Additionally, modern techniques such as Long Short-Term Memory (LSTM) Recurrent Neural Networks (RNN) will be explored. By leveraging a diverse array of approaches, the research endeavors to decode emotional responses within textual data. Despite considerable technological advancements, there's a pressing need for refinement, driving the quest for improved sentiment analysis. The study scrutinizes how different algorithms and models confront challenges inherent in textual data, poised to significantly contribute to the evolution of sentiment analysis.

Additional Key Words and Phrases: sentiment analysis, deep learning, naive bayes, LSTM, RNN, textual data, emotional responses, machine learning algorithms, technological advancements.

ACM Reference Format:
Nazmul Karim Tanvir ID:23166011, Tasnim Fuyara Chhoan ID:23366035, Samin Yasar ID:23273007, and A.B.M.Toufiqul Alam Sams ID:24166027. 2024. An Efficient Machine Learning Approach to Detect Sentiments from Text Data. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation emai (Conference acronym 'XX).* ACM, New York, NY, USA, 14 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Machines that have been programmed to think and behave like people use artificial intelligence (AI) to mimic human intelligence. A crucial area of artificial intelligence (AI), machine learning has essentially taken the place of AI in many applications. The major focus of our study is on automatically identifying emotions in text. Human emotions are affective states linked to physiological reactions. The [1] sentiment identification is being used in a variety of real-world scenarios where a person's emotional state acts as a clue to the machine learning system's efficiency. It may seem challenging to infer a person's emotional state from an analysis of a text document they have written, but this is frequently necessary because textual expressions are frequently the result of the interpretation of the meaning of concepts and the interaction of concepts stated in the text document. In the human-computer connection, understanding

*We are highly motivated to work more on this research.

Authors' Contact Information: Nazmul Karim Tanvir ID:23166011, nazmul.karim.tanvir@g.bracu.ac.bd, BRAC University, Dhaka, Bangladesh; Tasnim Fuyara Chhoan ID:23366035, BRAC University, Dhaka, Bangladesh, tasnim.fuyara.chhoan@g.bracu.ac.bd; Samin Yasar ID:23273007, BRAC University, Dhaka, Bangladesh; A.B.M.Toufiqul Alam Sams ID:24166027, BRAC University, Dhaka, Bangladesh.

Manuscript submitted to ACM

the text's mood is crucial. While text based emotion identification systems still require development, speech recognition has seen a great deal of progress. From an [2] application standpoint, where emotion is conveyed as joy, sadness, rage, surprise, hatred, fear, and other things, the ability to recognize human emotions in text is becoming more and more crucial. The emphasis is on the sentiment-related studies in the field of cognitive psychology because there isn't a standard sentiment word hierarchy. To identify our text-based emotional sentiments, we are going to use Naive Bayes.

First of all once more using the Bayes Theorem as its foundation, the Naive Bayes sorting algorithm presumes independence between predictors. Simply expressed, a Naive Bayes classifier thinks that the existence of one feature in a class has no bearing on the presence of any more features. In general, the Multinomial Naive Bayes classification method is a solid place to start when performing sentiment analysis. The Naive Bayes technique's basic tenet is that the probability of labels applied to texts are calculated using the cumulative values for words and classes. This characteristic helps models to comprehend how a word fits into the overall context of the phrase.

While Emotion Analysis seeks to identify certain sorts of sentiments expressed [3] in texts, such as anger, disgust, fear, happiness, sorrow, and surprise, Sentiment Analysis seeks to identify positive, neutral, or negative feelings from text.

1.1 Research Problem

Text is the basis for a machine to know what sentiment state the user is in initially and by using and storing the user's data, the machine will gradually self-taught itself on identifying the specific emotions. However, along with some advantages, there are some drawbacks in this system. According to a renowned article [4] by Apriorit, in the machine learning approach, we need to gather a huge amount of training data. It is quite a hassle to find a trusted corporation where we can receive and would be allowed to use their data for our experiment. Unfortunately, most of the public datasets are not sufficient. In those cases, we can create our own dataset or combine several datasets and keep modifying the data as the experiment progresses to solve this issue. Then, according to this research [5], inadequate or incomplete words are another challenge to solve for the model development part. In the Feature extracting process, it turns the data into features which are capable of being used for machine learning models. Gradually, the method has to come across the classification of algorithms and measure the performances, they are solved by different equations and measurements.

Finally, In natural language processing sentiment analysis is an active research field. As in blogs, social networks or product reviews it performs organizing and extracting sentiments from user generated text. In this research paper we are going to explore sentiment analysis challenges which are the most complex in natural language processing. However, many organizations face the challenges of sentiment analysis but these are not difficult to overcome with the right solutions. In this guide, we have faced some research problems related to this field in our research work. First of all, this paper tackles a fundamental problem which is word score polarity categorization. For instance, words "Happy" and "Sad" are high on positive (+) and negative (-) polarity scores but in between there is mid polarity [6]. Secondly, in our research paper these are somewhat positive and somewhat negative; these words sometimes get left out and dilute the sentiment score. Another problem we have faced is that sometimes people use memes and sarcasm in social media which make it difficult for sentiments tools to detect the actual context. Because of this reason the result sometimes shows the negative value which is basically positive.

In our research paper we have used one dataset- 1.Spotify App review dataset. Coursera Course dataset struggle to parse information because of biases. Finally, regardless of how this approach of machine learning solves the challenge of emotion recognition, we live in a period when additional issues may arise. As a result, according to this research paper [7], there will always be space for improvement in areas like language representation and categorization. The extraction of contextual information is critical during language representation because it serves as the foundation for enhancing categorization accuracy. Our goal in this paper is to train and test the selected models on dataset and find out which method is more appropriate and close to understanding a text just like a human.

1.2 Research Objective

Our primary objective is to devise robust techniques for automatic identification and analysis of emotions in textual data. Specifically, we aim to:

- Investigate state-of-the-art machine learning algorithms, particularly focusing on sentiment analysis methods such as Naive Bayes, and contemporary approaches like Long Short-Term Memory (LSTM) Recurrent Neural Networks (RNN).
- Explore the intricate correlation between textual expressions and underlying emotional states, leveraging insights from cognitive psychology and natural language processing research.
- Develop a sentiment analysis framework capable of accurately categorizing text into diverse emotional categories, spanning from joy and happiness to anger and sadness.
- Develop a sentiment analysis framework capable of accurately categorizing text into diverse emotional categories, spanning from joy and happiness to anger and sadness.
- Develop a sentiment analysis framework capable of accurately categorizing text into diverse emotional categories, spanning from joy and happiness to anger and sadness.
- Evaluate the performance of the proposed sentiment analysis framework using benchmark datasets and real-world textual data from various sources, including social media, customer reviews, and online forums.
- Evaluate the performance of the proposed sentiment analysis framework using benchmark datasets and real-world textual data from various sources, including social media, customer reviews, and online forums.
- Validate the effectiveness and practical applicability of the developed framework in scenarios such as sentiment analysis for product reviews, sentiment-based recommendation systems, and emotion-aware virtual assistants. Through these research objectives, our aim is to advance the field of emotion analysis in textual data, facilitating the development of more empathetic and context-aware AI systems that can adeptly comprehend and respond to human emotions.

2 RELATED WORK

This particular section indicates the importance of other research publications and how the works have impacted our process of understanding. After reading the research publications we have been critically analyzed that sentimental emotions play promising roles in the existence or the complete make-up of individuals. [5]

Authored by Abir Mishra, in this paper, the authors propose a robust framework for news text analysis, integrating NLP-based summarization and sentiment analysis to address information overload effectively. Despite acknowledged limitations, such as those inherent in any model, its demonstrated accuracy and utility render it valuable across industries. The authors advocate for further research aimed at augmenting its capabilities across diverse domains and languages to enhance its applicability.[8]

The paper by Aldy Rialdy Atmadja introduces a model for text summarization employing natural language processing (NLP) and sentiment analysis to mitigate information overload. Their approach, utilizing the NLTK library and Python, achieves a notable 91.67 percent accuracy in sentiment analysis. However, they acknowledge potential limitations due to reliance on a self-generated dataset and specific tools, which may restrict generalizability. Future endeavors aim to augment the model's versatility across various domains and languages through the integration of advanced tools, NLP techniques, and a recommendation system .[5]

The author Tejaswini Zope conduct a comprehensive analysis of global sentiments during the Covid-19 pandemic by examining Twitter data from diverse regions. They uncover prevalent feelings of sadness and fear, particularly pronounced in developing nations like India. However, they caution against over-reliance on findings due to challenges such as platform dependency and the complexity of detecting sarcasm, which necessitate nuanced interpretation[9].

Authored by Aldy Rialdy Atmadja,[10] through an examination of weather-related tweets originating from Phoenix and Singapore, the authors discern distinct patterns in sentiment. Notably, they observe heightened negativity during temperature rises, with Singapore consistently expressing dissatisfaction while sentiments in Phoenix vary seasonally. These findings underscore the influence of regional climates on social media expressions and highlight the long-term implications of heat-related events.[11]

According to Robert Plutchik's wheel model there are eight basic emotions such as joy, sadness, anger, fear, expectation, surprise, acceptance and disgust which are controlled by text from the paper works by other researchers. In this paper we try to show our task of sentiment analysis on text data and exchange of emotion through text message.[12] Without any interruption computers are unable to understand human emotion. [13]

A recent research work [10], by the research institute of Communication and Computer Systems (ICCS), Greece has proposed a method called NLP (Natural Language Processing) to understand human emotion by using text. It helps a computer to be capable of understanding the contents of speech which includes the contextual nuances of the languages. Among three NLP (Symbolic NLP, Statistical NLP, Neural NLP) neural NLP helps to identify the meaning of phrases which is very challenging for a computer to sort out.Additionally, It is significantly useful that the machine predicts and learns while training the data in a large dataset, but as there are lots of weights to make updates it becomes slow to train [14].

3 METHODOLOGY

This work's major goal is to employ a review dataset suitable for sentiment analysis of various human judgment and emotion spectrums that captures the ranges between the positive and negative poles. This is done to predict and detect emotions as accurately as possible; as the human mind is complex and so is human judgment which can only be satisfied through spectrum classification rather than simply the binary classification of whether the customers/ audiences were satisfied by the products they reviewed or not. In order to do so, the system needed to design a process where the machine respectively takes the input data, processes the dataset, predicts, analyzes and classifies the data and produces the correct output not only through word vectorization and token predictions but also through more layered and structured domains of text classification and context understanding.

To achieve this we took 1 model that can fall under the following categories Mid complexity model which is Naive Bayes. Textual documents are used for classification models to train the polarity and are represented by vectors to adopt the machine learning approach. Through trials on one dataset that include Spotify App Reviews the sentiment classification problem is analyzed. The suggested textual phenomena and the changes in languagespecific expressions need little computational resources, which may have an influence on automating sentiment identification and retrieval. A. Datasets To perform our experiment, we have used two different datasets which were collected from the renowned Kaggle site. We have used two datasets containing 64k Spotify App Reviews. This dataset were used in various experiments by other researchers from time to time. Aspect-based sentiment analysis in music: a case study with Spotify and Alternative Methods for Deriving Emotion Metrics in the Spotify Recommendation Algorithm, the 100K Spotify App Reviews dataset was utilized in both articles. For Spotify App Review Dataset, we kept 10000 reviews for the algorithm. However, as we used a pre-trained model in BERT algorithm that is why we kept 2000 reviews. We trained and tested on this size of dataset. For this dataset: we have chosen the columns that contains the reviews of the users and the ratings they have given to this app. From spotify dataset, we have chosen the 'Review' and 'Rating', and renamed each of these pairs of columns as 'review' and 'sentiments' respectively of this dataset for the ease of coding. This dataset were selected as each of them contain labels comprising of the rating score 1 - 5 which is helpful for detailed sentiment analysis as we can derive these scores as such - 1: negative, 2: somewhat negative, 3: neutral, 4: somewhat positive and 5: positive. Classifications of such variations allow for analysis along the spectrums of positive and negative aka the grey area between both extremes

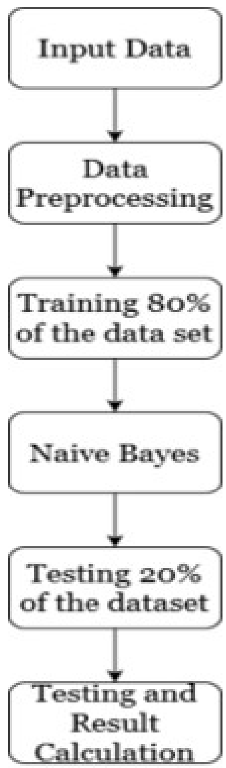

Fig. 1. Work Flow Diagram.

3.1 Exploratory Data Analysis

We've collected a dataset comprising reviews of the Spotify app collected from January 1, 2022, to July 9, 2022, sourced from the Google Play Store. The dataset consists of 61,594 entries across five columns, including Time submitted, Review, Rating, Total thumbsup, and Reply. These reviews offer valuable insights into user experiences, sentiments, and preferences regarding the Spotify app during the specified period.

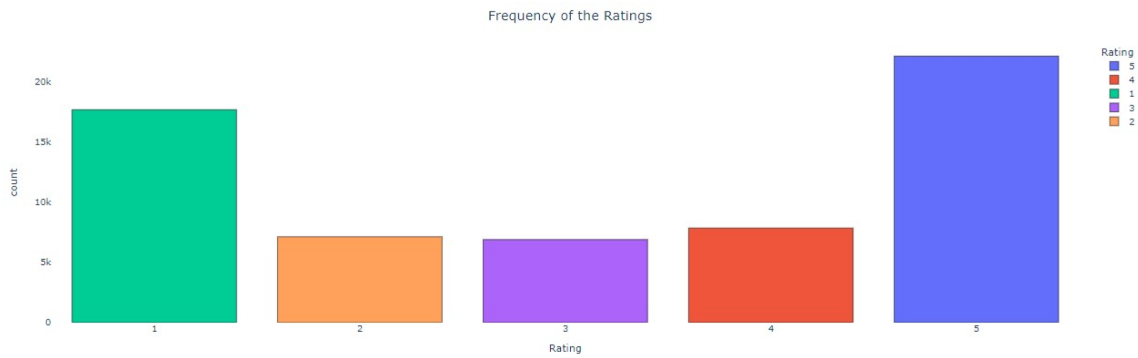

Fig. 2. Frequency of the Ratings

Analysis of rating distributions reveals a clear polarization in user sentiments, with peaks at both 1 and 5 points, indicating extremes in user experiences. Interestingly, reviews with higher ratings tend to have shorter text lengths compared to lower-rated reviews. This suggests that users may express their satisfaction or dissatisfaction more concisely.

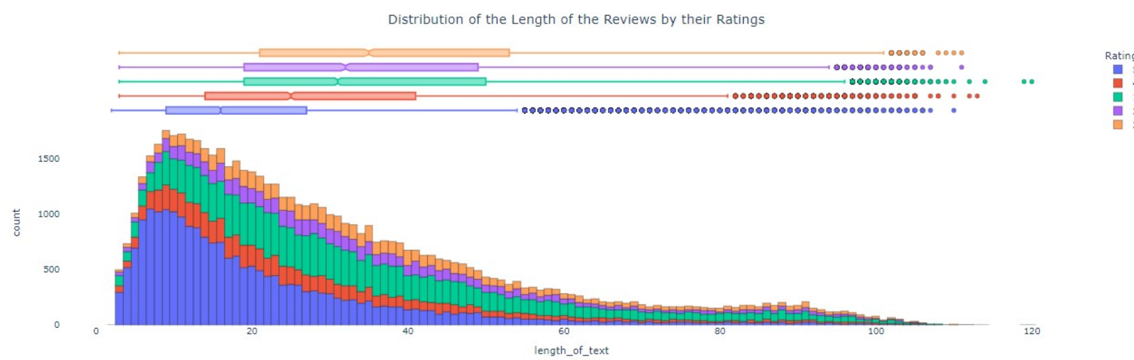

Fig. 3. Distribution of the length of the reviews by their ratings

This observation is further supported by the histogram plot (Figure 3), which shows that the median text length for reviews with a 5-star rating is significantly lower than that for reviews with a 1-star rating. In other words, positive reviews are typically shorter in length, while negative reviews tend to be longer.Figure 3 further illustrates the number of reviews over time, highlighting significant peaks on March 8, 2022, and between April 11, 2022, and April 23, 2022. These peaks suggest potential issues prompting increased user feedback, which warrants deeper investigation.Upon further breakdown of reviews during these peak periods (Figure 5), it becomes evident that there is a dominance of lower ratings (1 and 2 points), hinting at potential app functionality or user experience issues during those times. Subsequently, there is a notable increase in 5-point reviews, coupled with a decline in 1-point reviews over time (Figure 5), implying improvements in user satisfaction.

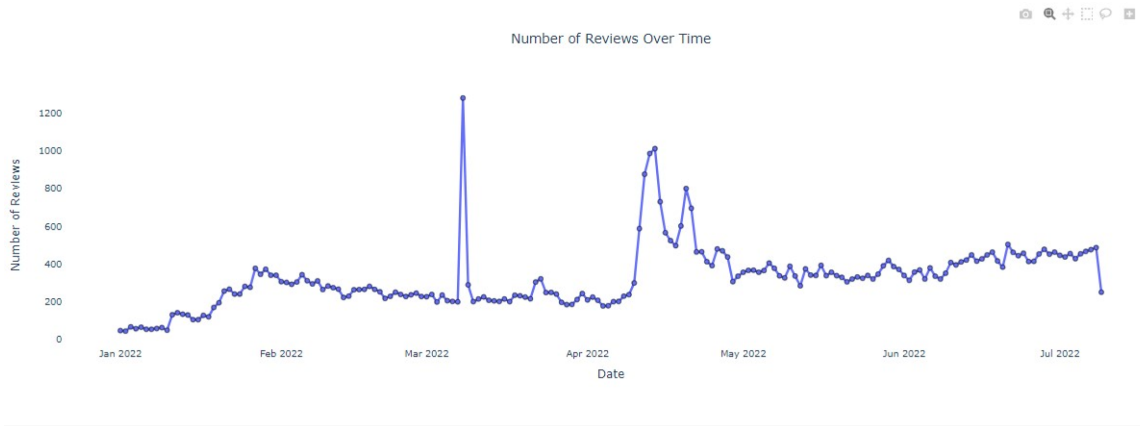

Fig. 4. Number of reviews over time

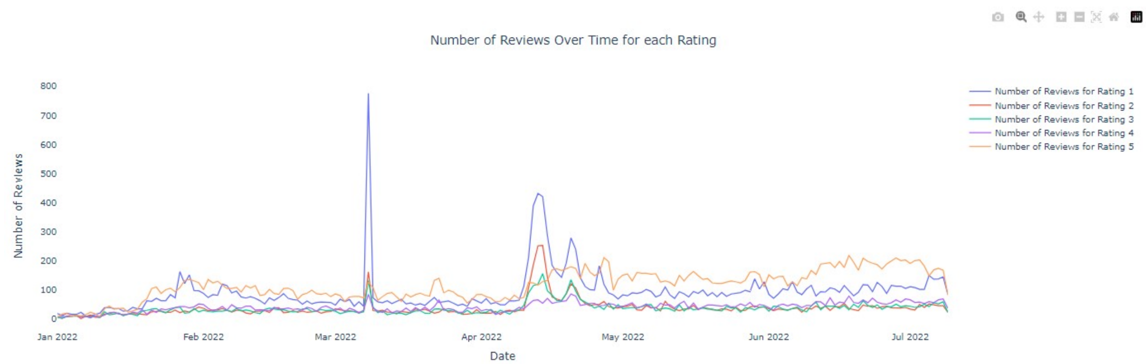

Fig. 5. Number of reviews over time for each ratings

3.2 Data Pre Processing

The datasets used in our paper did not meet all the necessary requirements to run through the codes, necessitating data cleaning first.Here, Figure 5 displays the top 100 most frequently used words before text cleaning, revealing the presence of unnecessary words such as "I," "is," and "so," which have minimal impact on sentiment classification. To enhance performance, it is essential to remove these irrelevant words.That is why we needed to clean the datasets first by removing the special characters by importing and applying regular expression operation and also kept all the alphabets in lower case.After lowercasing, expanding contractions, removing numbers, punctuation, emojis, non-Latin characters, and words less than 2 characters, we tokenize the text and remove stopwords. Then, we lemmatize the text to reduce words to their base form. Additionally, we normalize the text by removing extra whitespaces and ensuring consistent spacing between words. Moreovery, we check for and remove any duplicate rows to ensure the dataset's cleanliness before analysis. Additionaly, we used df.shape operation to know the shape of our dataset. We also collect the number of unique values of the sentiment column. To find out about the null values, we used df.isnull().sum(). Then, we needed to remove all the unnecessary rows and columns from our datasets. For doing this, we used df.drop()

function. To drop an unnecessary column we had to keep the axis as 1 and for dropping the rows, we need to keep the axis as 0. By the end of all the pre-processing, we can get the datasets that we actually going to use for our codes.

After text cleaning and pre-processing, Figure 7 shows the results, which significantly improve the efficiency of the model. Cleaning and pre-processing the data ensure that the model focuses on meaningful words for sentiment analysis. This process reduces noise in the data, resulting in more accurate sentiment classification, and ensures consistent spacing between words, further improving the model's performance. These steps contribute to better feature extraction and a more effective sentiment classification model overall.

Fig. 6. Before text cleaning

Fig. 7. After text cleaning

3.3 Naive Bayes Model Explanation:

For conducting preprocessing operations like changing all words to lowercase and deleting special characters, unnecessary rows and columns. Then, CountVectorizer implies breaking down a phrase or any text into words. Textual data has to be vectorized since NLP algorithms only take numerical data and cannot interpret language. Then, we stored the review in a variable x and the sentiment of the dataset was stored in another variable called y. Now, with the help of the train test split() function, we split the whole dataset into two and the parameters are such as x, y, random state is 42, test size is 0.20. The test size is 20 percent in this code means that 20 percent of this dataset is for testing and the rest of the 80 percent reviews of the dataset is for training.

$$\mathrm{NB} \longrightarrow \operatorname{argmax} p(y) \prod p(w \mid y)^{f_i}$$

$$\mathrm{CNB} \longrightarrow \operatorname{argmin} p(y) \prod \frac{1}{p(w \mid \hat{y})^{f_i}}$$

Fig. 8. Complement Naive Bayes

$$\hat{P}(c) = \frac{N_c}{N}$$

$$\hat{P}(w \mid c) = \frac{count(w,c)+1}{count(c)+|V|}$$

Fig. 9. Multinomial Naive Bayes

Moreover, for getting accuracy we applied MultinomialNB() classifier. When analyzing categorical text data, one of the most well-liked supervised learning classifiers is Multinomial Naive Bayes. Using the Bayes principle, it makes an educated prediction about a text's tag, like "story." It determines the likelihood of each tag for a certain sample and produces the tag with the highest likelihood. While fit function is scaling all the training data, the x test is transformed into vectorizer and stored in cv test transformed. The predict function is working with the cv test transformed in it and stored it in y pred. By applying accuracy score and classification report functions, we got an accuracy and that is 62Complement Naive Bayes is used when there is an imbalanced dataset and when Naive Bayes can not do that. So, in this case, we used it.

At first, importing ComplementNB from sklearn library and also import class weight. Then, class weight= 'balanced', classes=np.unique(y train), y = y train were the parameters of class weight.compute class weight. Then, we fit the necessary parameters by weighted model.fit. With the help of predict function, we took cv test transformed. With accuracy score, we got the accuracy. After applying oversampling, the value counts of y train are now transferred in each count class of five labels. Then, we applied the Concat function on the x train, y train, and also kept the axis as 1. Then the five sentiments of the training part were moved to the five different classes. Now, in the sampling process where we took the class 5 and replaced it with the classes from 1 to 4 for oversampling. Again, by using the Concat function all the oversampled classes and class 5 and axis as 0. Now, we assigned the oversampled train dataset into the new X train and Y train accordingly and X test = x test and Y test = y test . After fitting the training part and then transformed in cv train transformed1. In the following line, the training dataset got fitted in the classifier and also predicted the cv test transformed1.

Lastly, the accuracy was called to get the performance result and now the model is predicting each sentiment more accurately. If we label the ratings of sentiment 1 and 2 as 1, 3 as 2 and 4 and 5 as 3, that means we now we divided our dataset into positive, neutral and negative sentiment. As we apply the MultinomialNB() classifier, it gives the accuracy of 78sentiment label 3. However, class weight performed well. After oversampling all three sentiments prediction performed well.

4 RESULT ANALYSIS

In our research conducted on Google Colab, we examined the accuracy rates of various algorithms in sentiment analysis. We meticulously assessed precision, recall, and the F1-score to gauge model effectiveness.

Our datasets exhibited imbalances in sentiment class distribution, with some classes having significantly more samples than others. For example, in the Spotify App Review dataset, class 5 had 4181 samples, class 4 had 1352, class 3 had 1021, class 2 had 920, and class 1 had 2526. This imbalance posed a challenge, especially for neutral and mid-pole sentiments (classes 4 and 2).

Despite the models' ability to predict negative sentiment (class 1) effectively, they struggled with the other classes, resulting in skewed accuracy, precision, recall, and F1-score metrics.

To address this issue, we implemented Oversampling (Upsampling), increasing the sample size of minority classes to match the majority class. Class 5, representing 'positive' sentiment, was the majority class in both datasets. Thus, we adjusted the sample sizes of other classes accordingly, ensuring equal representation for each sentiment class during model training. This facilitated fair and precise evaluation of model performance.

Additionally, we recognized the inherent difficulty in predicting neutral and mid-pole sentiments due to their abstract nature. To tackle this challenge, we initially trained the models on imbalanced data and then applied bias handling techniques to compare performance before and after imbalance correction.

Our findings, presented through accuracy scores and detailed classification reports, highlighted the impact of imbalance handling techniques on model performance across both label categorizations—before and after imbalance correction. Moreover, our research underscores the importance of addressing dataset imbalances in sentiment analysis and demonstrates the efficacy of bias handling techniques in improving model accuracy and fairness across all sentiment classes.

4.1 Accuracy Score Analysis:

Initially, we observe that the accuracy scores prior to addressing imbalance or bias are notably higher compared to those after balancing the dataset through oversampling. This discrepancy arises because accuracy, being a straightforward metric, thrives when data is evenly distributed. However, its reliability diminishes significantly in imbalanced datasets.

In such cases, if models predominantly focus on or exclusively predict the majority class while neglecting minority ones, accuracy may misleadingly indicate high performance by overlooking failures in minority predictions. Conversely, the F1 score, with its weighted approach that accounts for true positives, false positives, and false negatives, emerges as a more dependable metric for imbalanced datasets. Despite seemingly high accuracy scores before bias handling, the F1 scores, particularly for lower-sampled classes, reveal significant under-performance.

For Spotify, although its accuracy may suffer due to a broader prediction range, the F1 score offers a more nuanced perspective, indicating improved performance due to its comparatively larger dataset across all sentiment classes. Upon oversampling to ensure equal data distribution, while accuracy scores decrease, there is a noticeable enhancement in precision, recall, and F1 scores for minority classes. This improvement signifies increased efforts in predicting these classes, thereby enhancing overall model performance.

Before oversampling, Naïve Bayes models exhibit higher F1 scores for neutral and mid-polar classes (3, 2, 4), a result aligned with expectations given their capacity for complex context understanding. Despite significant class sample imbalances, these models outperform simple word-vectorizing algorithms, underscoring the importance of nuanced model architectures in handling such complexities.

4.2 Imbalance Handling in Dataset:

The datasets we used for our research has asymmetric number of samples for each class especially where there is a big different in the distribution of the classes and this causes our ML models to develop a bias towards the class with the highest number of samples and this bias works against other classes with lesser number of samples respectively.

```
ComplementNB Naive Bayes prediction score for 3labels : 0.7723841220878318
Accuracy of the Model: 77.24%

classification report:
              precision    recall  f1-score   support

           1       0.72      0.87      0.79      4954
           2       0.29      0.03      0.06      1377
           3       0.83      0.86      0.85      5988

    accuracy                           0.77     12319
   macro avg       0.62      0.59      0.57     12319
weighted avg       0.73      0.77      0.74     12319
```

Fig. 10. For 3 levels after Imbalance Handling

As a result, our accuracy score becomes skewed and unreliable as the ML algorithms develop the tendency to predict the dominant classes while ignoring the minority class. Such events run the risk of algorithms entering an accuracy paradox which is such a state where the accuracy score gives a high accuracy giving the impression of high performance but in actuality the model just predicts the class with the most information available, i.e., the most data available in the training set and is the easiest to predict and predicts those classes predominantly to acquire the highest accuracy score.

```
MultinomialNB Naive Bayes prediction score for 5labels : 0.6124685445247179
Accuracy of the Model: 61.25%

classification report:
              precision    recall  f1-score   support

           1       0.57      0.85      0.68      3531
           2       0.29      0.11      0.16      1424
           3       0.29      0.10      0.15      1377
           4       0.39      0.29      0.33      1568
           5       0.78      0.86      0.82      4419

    accuracy                           0.61     12319
   macro avg       0.46      0.44      0.43     12319
weighted avg       0.56      0.61      0.57     12319
```

Fig. 11. For 5 levels after Imbalance Handling

```
Accuracy of the Model: 77.08%

classification report:
              precision    recall  f1-score   support

           1       0.72      0.88      0.79      4954
           2       0.27      0.05      0.09      1377
           3       0.85      0.85      0.85      5988

    accuracy                           0.77     12319
   macro avg       0.61      0.59      0.58     12319
weighted avg       0.73      0.77      0.74     12319
```

Fig. 12. For 3 levels after Imbalance Handling

The Spotify App Review dataset also carries such imbalances but in Spotify classes 5 and 1 have lesser difference among their sample size above the rest of the classes prove to have bigger differences in their sample sizes being compared to the majority class. Since we are observing the behavior of the one types of models we used on certain text classification problems in sentiment analysis, a skewed, unreliable and incorrect prediction rate is of little value. Therefore, we have decided to address this imbalance with the help of two techniques: 1. Class-weight computation and 2. Randomly oversampling/ Up-sampling the samples belonging to the minority classes to the sample size of the majority class. In Class weight computation, we assign certain weight to the classes present in our dataset such that different weights are assignment for the majority and minority classes. The difference in weights will influence the classification of the classes during the training phase. The whole purpose is to penalize the misclassification made by the minority class by setting a higher class weight and at the same time reducing weight for the majority class. We imported the class weight from the ‘sklearn.utils’ library and used the ‘compute class weight’ method which automatically sets the class weights for the classes present in accordance to their sample size such that the most frequent and easy to predict class gets the least precedence over the class with the lowest samples and hardest to predict.

Misclassification of the classes is penalized in accordance to the weights set. In random Oversampling we increase the sizes of the minority classes each to that of the majority class or the most frequent class in the dataset by randomly resampling them to the size of the most frequent class in the training data.

This gives us an equal distribution of all the classes present in the dataset to work with during train then we test the ML algorithm with some testing data to evaluate the changes in precision, recall and f1 scores.

4.3 Accuracy analysis:

From the above tables we can see that the accuracy of these two techniques go neck-to-neck which means that both class-weight and Random oversampling give close results for the dataset. For Complement-Na¨ıve Bayes, a variation of Multinomial Naiıve Bayes specifically designed to handle class/ sample weights, we see higher value of class weights than oversampling for 5 labels oversampling

5 CONCLUSION

People in the modern world spend a lot of time on social media. Here we have many platforms to interact with people to convey our thoughts and feelings. But sometimes by reading a text it is quite difficult to understand the feelings as we cannot see the person’s expression. Emotions in people are sentimental states which is connected to physiological reactions. Many real-world applications that use sentiment identification use a person’s emotional state as an indication to how well the machine learning system is working. Although it may seem difficult, it is frequently necessary to infer a person’s emotional state from an analysis of a text they have written because textual expressions are frequently the result of the interpretation of the meaning of concepts and the interaction of concepts stated in the text document.

In our project our work is to recognize sentiments in Text by Using Machine Learning. Many different applications, such as speech recognition, email filtering, computer vision, and medicine, use machine learning algorithms when it is difficult or impossible to develop standard algorithms to do the necessary tasks. To identify our text-based sentiments, Naive Bayes. Our work was to find out the best model which will fit in our project. Though our field of work is still new and there are lots of new things to explore. So there are still some limitations that are yet to be resolved and in future we will work on it if we find something better then Naive Bayes. We do not want to stop our work here. In this case we are wishing to work on LSTM, Tf-Idf. Lastly we want to make our research paper more efficient and resourceful for other people. So that in future people can use our paper for their own research. That’s how our project will be more enriched and can be helpful for future researchers.

REFERENCES

[1] D. Mehta, M. F. H. Siddiqui, and A. Y. Javaid, “Recognition of emotion intensities using machine learning algorithms: A comparative study,” *Sensors*, vol. 19, no. 8, p. 1897, 2019.
[2] S. N. Shivhare and S. Khethawat, “Emotion detection from text,” *arXiv preprint arXiv:1205.4944*, 2012.
[3] A. Yadollahi, A. G. Shahraki, and O. R. Zaiane, “Current state of text sentiment analysis from opinion to emotion mining,” *ACM Computing Surveys (CSUR)*, vol. 50, no. 2, pp. 1–33, 2017.
[4] A. Beliba, “Challenges of emotion recognition in images and video. apriorit,” 2021.
[5] Y. Dzyuban, G. N. Y. Ching, S. K. Yik, A. J. Tan, P. J. Crank, S. Banerjee, R. X. Y. Pek, and W. T. L. Chow, “Sentiment analysis of weather-related tweets from cities within hot climates,” *Weather, Climate, and Society*, vol. 14, no. 4, pp. 1133 – 1145, 2022. [Online]. Available:

https://journals.ametsoc.org/view/journals/wcas/14/4/WCAS-D-21-0159.1.xml

[6] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence embeddings using siamese BERT-networks," 2019.

[7] F. A. Acheampong, C. Wenyu, and H. Nunoo-Mensah, "Text-based emotion detection: Advances, challenges, and opportunities," *Engineering Reports*, vol. 2, no. 7, p. e12189, 2020.

[8] Baeldung. (2020) Introduction to emotion detection in written text. https://example.com.

[9] M. Krommyda, A. Rigos, K. Bouklas, and A. Amditis, "An experimental analysis of data annotation methodologies for emotion detection in short text posted on social media," *Informatics*, vol. 8, p. 19, 2021.

[10] SentimentGuide. Sentiment analysis guide. https://monkeylearn.com/sentiment-analysis/.

[11] A. Mishra, A. Sahay, M. a. Pandey, and S. S. Routaray, "News text analysis using text summarization and sentiment analysis based on nlp," in *2023 3rd International Conference on Smart Data Intelligence (ICSMDI)*, 2023, pp. 28–31.

[12] A. R. Atmadja, A. Rahmawati, C. N. Alam, P. Dauni, and Y. Saputra, "Sentiment analysis on tourism place using naive bayes," in *2023 17th International Conference on Telecommunication Systems, Services, and Applications (TSSA)*, 2023, pp. 1–6.

[13] T. Zope and K. Rajeswari, "Sentiment analysis of covid-19 tweets using twitter database–a global scenario," in *2022 IEEE 4th International Conference on Cybernetics, Cognition and Machine Learning Applications (ICCCMLA)*, 2022, pp. 27–30.

[14] M. M. Mutlu and A. Ozgur, "A dataset and BERT-based models for targeted sentiment analysis on Turkish texts," *arXiv preprint arXiv:2205.04185*, 2022.