Sentiment Analysis using Text Mining and Natural Language Processing

[1]Yaswanth Ganapathi, [2]Shirisha Gajjela, [3]Rahul Sajith

Abstract

In today's digital world understanding people's sentiments which are expressed on social media has become important. Social media is like giant megaphone for everyone's opinions and emotions. By using the emotions in text we are able to understand the sentiments. In this study, we use machine learning techniques to predict the sentiments. To correctly predict the sentiment of text, we use Logistic Regression, Naive Bayes, Random Forest, Support Vector Machines, Boosting Algorithms like Gradient Boosting, Extreme Gradient Boosting(XG), Adaptive Boosting(ADA). Logistic Regression outperformed the other models with an accuracy of 80%. The findings imply that machine learning algorithms have the ability to accurately classify tweets by providing valuable insights into public opinion trends for better decision making in various domains.

Introduction

In the rapidly growing landscape of social media, understanding the sentiments expressed by individuals has emerged as an important task with the vast number of opinions and emotions broadcast across platforms, the underlying sentiments encoded within text has become important. In this study, we utilize machine learning techniques to predict sentiments, leveraging the pool of emotions embedded in textual data.

Sentiment analysis is a powerful tool for analyzing consumer data from a variety of demographics, including geography, race, and culture. It is used to determine the simplest details about the performance of advertising campaigns, rival brands, and the popularity of products. Due to the fact that e-commerce and hospitality have a variety of customers, linguistic sentiment studies are especially helpful in these areas, particularly when audiences are located in global locations. Natural Language Processing (NLP) is especially designed to understand and process natural language text. It plays a major role at handling linguistic nuances and sentiment in text considering factors such as negation, sarcasm, and idiomatic expressions. We utilize sentiment analysis results to improve customer and staff experience, brand awareness and perception, and efficiency in operations, depending on the industry and requirements.

In this study, we use a variety of machine learning models to precisely classify the tweets, using Logistic Regression, Random Forest, Naive Bayes, Support Vector Machines (SVM), Boosting Algorithms like Gradiant Boosting, Adaptive Boosting(ADA) and Extreme Gradient Boosting (XGB). These models have a reputation for handling complex data and providing accurate analysis. We use a dataset of tweets from twitter to train these models. This dataset contains variables including text id , text, selected text, and sentiment.

Our main goal is to assess the effectiveness of various models and identify which one offers the most accurate classification of tweets. Based on each model's accuracy, precision, recall, and F-1 score, we compare the findings. The results of this study will have a big impact on analysis of sentiments from classification of tweets accurately for better decision making in various domains.

Apart from the sentiment analysis there are still a number of dependencies and data restrictions that need to be overcome. Firstly, to obtain higher accuracy ML models need big and varied data sets. However, Twitter data pose challenges in terms of quantity and scope as the datasets may not appropriately provide wide range of sentiments which are expressed on twitter. Secondly, ML models may be overfitted or underfitted, due to the dynamic nature of twitter data and different languages used in it causes the model struggle to generalize well to unseen tweets.

To overcome this challenges it is important to select appropriate model and validation techniques. Thus we ensure accuracy and robustness of sentiment analysis from twitter data.

Related Work

The paper [1] explains the sentimental analysis based on Extraction Transform and Load Techniques,(ETL) by using NLTK library and Valence Aware Dictionary and Sentiment Reasoner (VADER) a lexicon and rule based sentiment analysis tool. The author showed the sentiment analysis in the form of polarity.

Author [2] showed the efficiencies of different machine learning algorithms with semantic analysis. Geetika and Divaker are able to draw an accuracy of 88.2% with Naive bayes and also with SVM they achieved accuracy of 85.5%

This paper [3] explains the text mining basics and preprocessing steps needed for text mining like tokenization, stemming, stop word removal. Dr.Vijayarani and Mrs.Janani

have compared the performances of available open source tools for tokenizations and visualized each and every tool.

The work by Nasifa Ira and Mohammad Rahman [4] have depicted the use of different classifiers like Random Forest, Gradient boosting, and Hist gradient boosting. They also proposed an ensemnle model which acheived an accuracy of 70%.

Data

The dataset is **Twitter Dataset** which was obtained from Kaggle. The Dataset has 27482 observations and 4 features total and Sentiment is our target variable. We classify responsiveness by sentiment. Tweets are analyzed as Positive, Negative and neutral sentiments.

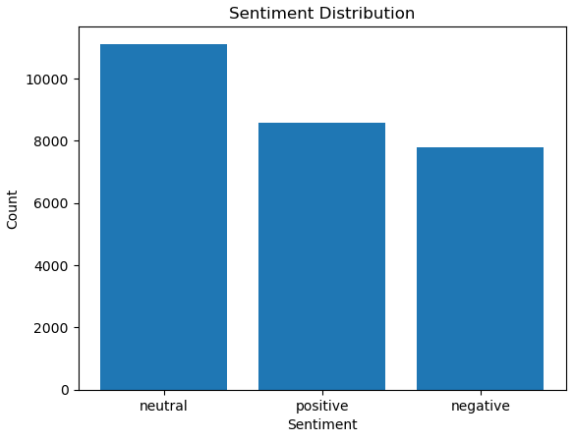

Figure 1: Bar chart for distribution of sentiment

The above figure 1 represents the bar chart that displays the distribution of sentiments.

Data Preprocessing

Data preprocessing involves preparing and cleaning raw text data for analysis, like we had removed stop words like ”and,” ”the,” ”is,” etc., that do not carry significant meaning in the context of sentiment analysis which in turn helped in reducing the dimensionality of the data and focusing on more meaningful words. We have also reduced words to their base or root form. In our analysis ”running” and ”runner” are stemmed to ”run” which helps in reducing the variation of words and treating similar words as the same.

First, all text data is converted to lowercase to ensure consistency and avoid any case-sensitive variations. Next, URLs are removed using regular expressions, as they typically do not contribute to sentiment analysis. Special characters and numbers are also eliminated to reduce noise in the text data. The text is then tokenized using NLTK's `word_tokenize` function to split it into individual words or tokens. Stopwords, common words like ”and” or ”the” that don't carry much meaning, are removed using NLTK's `stopwords` corpus. Additionally, stemming is performed using NLTK's `PorterStemmer` to reduce words to their root form and improve model performance.

Finally, the cleaned tokens are rejoined into a string, representing the preprocessed text. These preprocessing steps are applied to both the 'text' and 'selected_text' columns of the DataFrame, which respectively contain the original tweet text and the relevant portion of the tweet text for sentiment analysis.

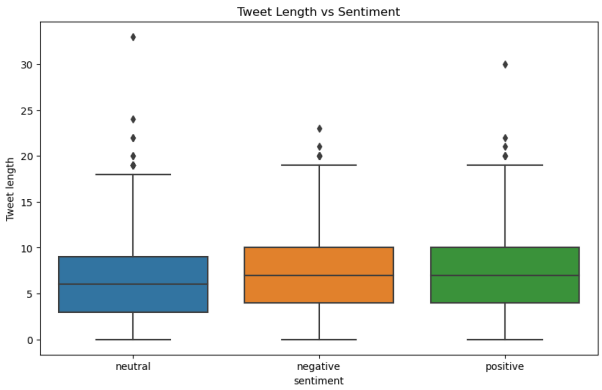

Figure 2: Tweet Length vs Sentiment after preprocessing.

The above figure represents the boxplot after preprocessing the data.

Exploratory Data Analysis (EDA)

Here, we conducted Exploratory Data Analysis (EDA) to understand the underlying patterns and relationships within the dataset before proceeding with model building. We generated word clouds to visualize the most frequently occurring word tweets which are positive, negative, and neutral separately. Figure 3 represents the word cloud for positive tweets, words like ”love”, ”thank”, ”hope”, ”life”, ”great” and ”happy” dominate, which are typically associated with positive emotions and experiences.

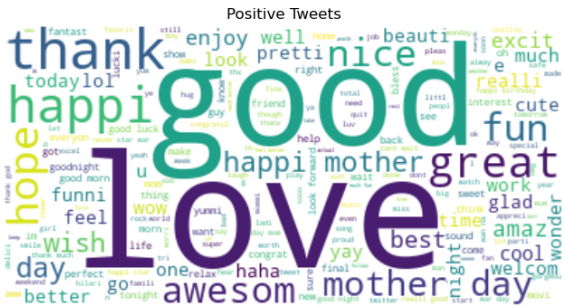

Figure 3: Word cloud for Positive tweets

Figure 4 represents the word cloud for negative tweets, here words are scattered such as ”miss”, ”sorry”, ”bore” and ”ugh”, reflecting dissatisfaction or unpleasant states.

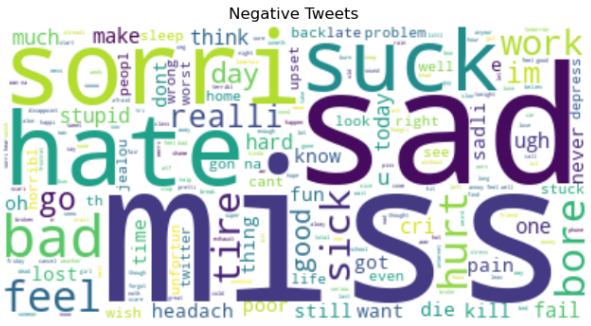

Figure 4: Word cloud for Negative tweets

We generated Word cloud for neutral tweets which are shown in below figure 5, We can see that neutral tweets contain a mix of terms that are less emotionally charged, like ”work”, ”today”, ”going” and ”watch.” This category seems to encompass a variety of subjects.

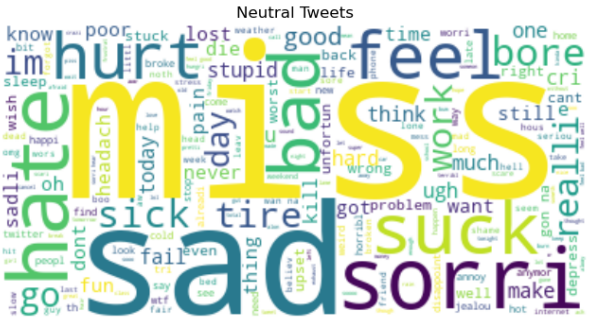

Figure 5: Word cloud for Neutral tweets

Methods: Model Building and Evaluation

We are currently applying four machine learning models which are Logistic Regression, Naive Bayes, Support Vector Machine (SVM), and Random Forest. We segregated the dataset into separate sets for training, testing, and validation to ensure robust evaluation and mitigate overfitting. We evaluated each model using precision, recall and f1 score.

Logistic Regression

Logistic regression, a statistical method used for analyzing data, demonstrates its effectiveness through its predictive capability. When we applied to the original dataset, it accurately predicted around 86 out of 100 instances. But, when tested on new data, it maintains the best accuracy of approximately 80 out of 100 predictions. Thus, it exhibits reliability in making predictions. The confusion matrix, depicted in Figure 6, represents its consistent performance across various sentiment categories. With an approximate 81% accuracy rate, the model adeptly discerns sentiments, correctly classifying 8 out of 10 tweets. We evaluated the model's precision, recall, and F1-score for three classes (test size = 0.3): 0, 1, and 2. It shows that class 1 has the highest recall (0.84) and F1-score (0.80), while class 2 has the highest precision (0.87) and an F1-score of 0.83. Overall accuracy stands at 80%.

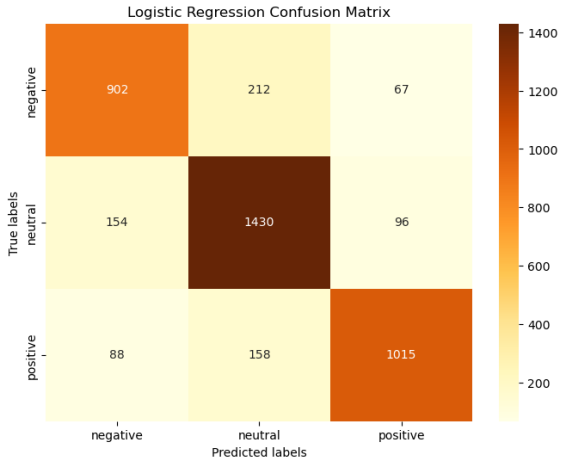

Figure 6: Confusion matrix for Logistic Regression

Naive Bayes

The model belongs to a class of linear ”probabilistic classifiers,” and similar to Logistic Regression, and demonstrates proficiency in prediction. When we train the data it correctly predicts roughly 83 out of 100 instances, while on the validation set, it achieves about 78 out of 100 correct predictions. Overall, the Naive Bayes model attains an accuracy of approximately 78%. It evaluates the precision, recall, and F1-score for three classes (test size = 0.3): 0, 1, and 2. Class 1 has the highest recall (0.88) and an F1-score of 0.78, while class 2 has the highest precision (0.85) and an F1-score of 0.82. The overall accuracy of the model is 78%.

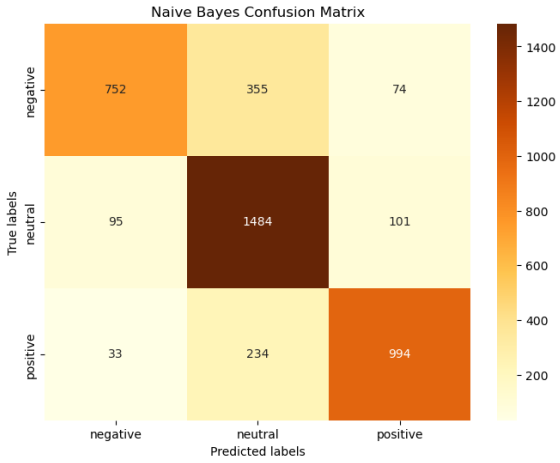

Figure 7: Confusion matrix for Naive Bayes

Support Vector Machine (SVM)

When we tested a support vector machine (SVM) on the data it was trained on, it correctly predicted around 91 out of 100 cases. When we tested it on new data (the validation set), it still achieved about 80 out of 100 correct predictions. This indicates that the SVM not only learns well from the provided data but also performs accurately when faced with new, unseen data.

It evaluates precision, recall, and F1-score for three

classes (test size = 0.3). Class 1 shows the highest recall at 0.88 and an F1-score of 0.81, while class 2 displays the highest precision at 0.88 with an F1-score of 0.83. Overall, the model achieves an accuracy of 80%.

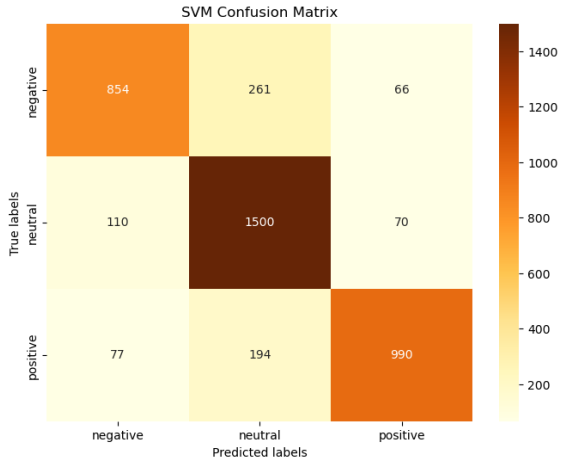

Figure 8: Confusion matrix for Support Vector Machine

Random Forests

The random forest algorithm combines multiple decision tree outcomes to generate a single prediction. When we tested the training data, it accurately predicted around 95 out of 100 cases. However, on new data (the validation set), it achieved approximately 79 out of 100 correct predictions. While performing well, it lacks the consistency of the SVM model. Figure 9 represents the confusion matrix for the random forest. It evaluates precision, recall, and F1-score for three classes (test size = 0.3): 0, 1, and 2. Notably, class 1 has the highest recall at 0.84 and an F1-score of 0.81, while class 2 shows the highest precision at 0.86 with an F1-score of 0.82. Overall, the model achieves an accuracy of 80%, with a macro average F1-score of 0.79.

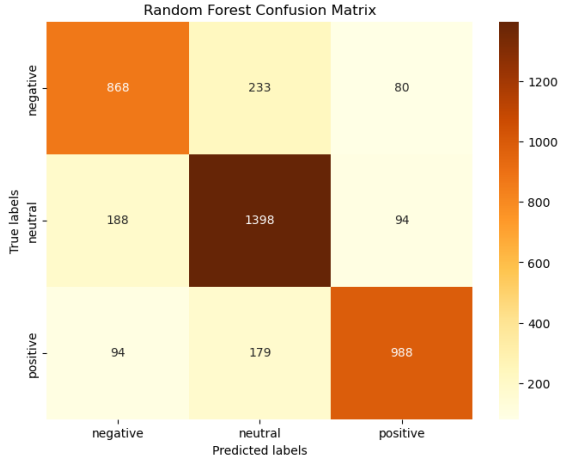

Figure 9: Confusion matrix for Random Forest

Boosting Algorithms

Furthermore, We explored advanced boosting algorithms like XGBoost and Gradient Boosting.

Extreme Gradient Boosting (XGBoost)

The XGBoost model was trained on the training data and achieved a validation accuracy of approximately 75.50%. When evaluating its performance on the validation set, We plotted a confusion which is shown in Figure 10, providing insights into its classification results.

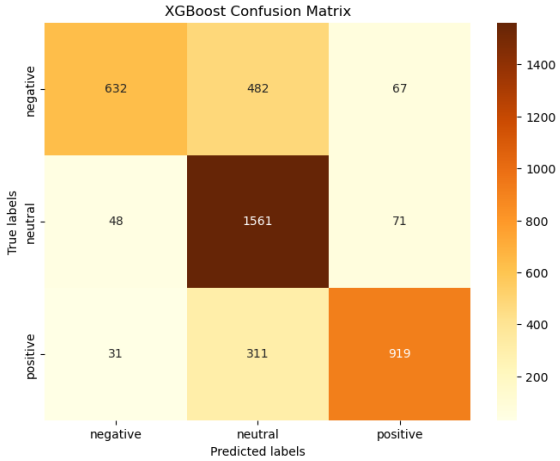

Figure 10: Confusion matrix for Extreme Gradient Boosting

The precision, recall, and F1-score are performed for three classes (0, 1, and 2). Notably, class 1 demonstrates the highest recall at 0.92 and an F1-score of 0.76, while class 2 exhibits the highest precision at 0.87 with an F1-score of 0.79. The overall accuracy of the XGBoost model on the test data stands at 75%, with a macro average F1-score of 0.74.

Gradient Boosting

Firstly, the model was trained on the training data, and when evaluated on the validation set, it achieved an accuracy of approximately 70.28%. We constructed a confusion matrix to visually represent the classification results on the validation set.

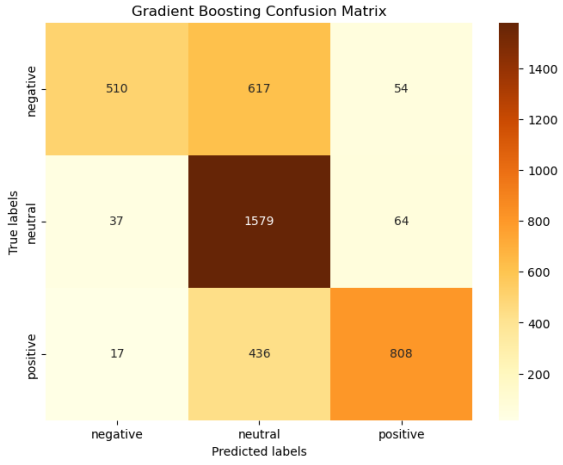

Figure 11: Confusion matrix for Gradient Boosting

The precision, recall, and F1-score are performed for three classes (0, 1, and 2). For example, class 1 has a precision of 0.59, recall of 0.94, and F1-score of 0.73. The overall accuracy of the model on the test data is 70%. The macro

average F1-score, which is the average of the F1-scores for each class, is 0.68. Additionally, the weighted average F1-score, considering class imbalance, is also 0.69.

Experiments and Results

After preprocessing we have divided the data into training and testing sets. We have used a ratio of 70:30 to split our data. We conducted experiments using precision, recall and f1-score to evaluate the performance of our models without using any hyperparameter. We achieved an accuracy of 85% in classifying customer reviews into sentiment categories. The interesting thing about this result is that we achieved this high accuracy without using hyperparameters. The part where our model might be making a mistake is that it might be overfitting the model. As you can see we have that we have a training accuracy of 85.72% and a validation accuracy of 80.16%. This difference in accuracy between the training and validation set suggests that the model may be overfitting the training data which in turn captures noise in the data rather than the underlying pattern. This indicates that the model may benefit from some form of regularization or tuning to improve its generalization to new, unseen data.

Visualization

We constructed grouped bar plots and provided a clear comparison of accuracy, precision, recall, and F1-score for each model, aiding in model selection and evaluation. Figure 12 shows grouped bar plot for Logistic Regression, Naive Bayes, SVM and Random Forest and Figure 13 represents the grouped bar plot for XGboosting and Gradient Boosting.

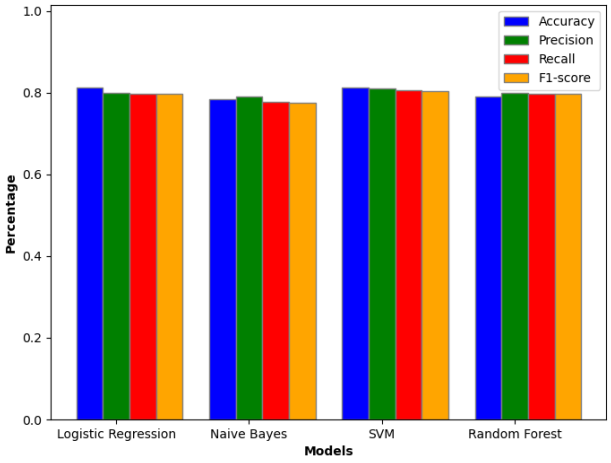

Figure 12: Grouped bar plot comparing Four Machine Learning Model performance

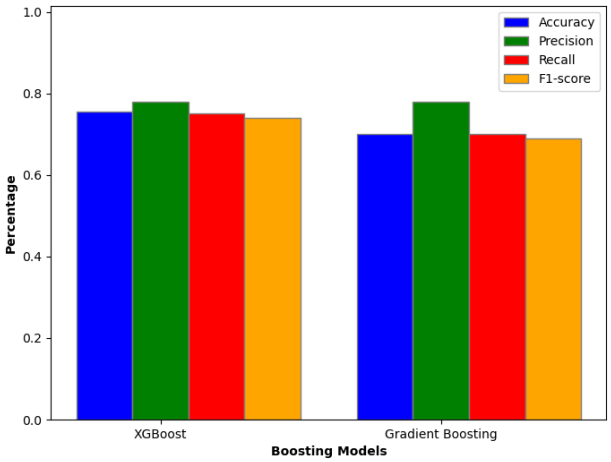

Figure 13: Grouped bar plot comparing XGboosting and Gradient Boosting

Conclusion

We can conclude that the logistic regression model beat the other models in terms of accuracy, precision and recall. The experiments illustrate the efficacy of logistic regression and SVM for sentiment analysis on Twitter data, achieving validation accuracies of approximately 81%. These results indicate that the model performs reasonably well in predicting sentiment based on text data. However, there is a slight drop in performance from training to validation, suggesting potential overfitting.

Future researchers instead of just classifying texts into positive, negative or neutral categories could explore more nuanced sentiment analysis, such as identifying specific emotions(e.g. Joy, anger, sadness) or intensity levels of sentiment.

References

[1]A. Shelar and C. -Y. Huang, "Sentiment Analysis of Twitter Data," 2018 International Conference on Computational Science and Computational Intelligence (CSCI), Las Vegas, NV, USA, 2018, pp. 1301-1302,doi:10.1109/CSCI46756.2018.00252. https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=arnumber=8947771tag=1

[2]G. Gautam and D. Yadav, "Sentiment analysis of twitter data using machine learning approaches and semantic analysis," 2014 Seventh International Conference on Contemporary Computing (IC3), Noida, India, 2014, pp. 437-442, doi: 10.1109/IC3.2014.6897213. https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=arnumber=6897213

[3]Mohan, Vijayarani. (2016). Text Mining: Open Source Tokenization Tools: An Analysis. 3. 37-47, https://www.researchgate.net/publication/329800669 $Text_Mining_Open_Source_Tokenization$

[4]N. T. Ira and M. O. Rahman, ”An Efficient Speech Emotion Recognition Using Ensemble Method of Supervised Classifiers,” 2020 Emerging Technology in Computing, Communication and Electronics (ETCCE), Bangladesh, 2020, pp. 1-5, doi: 10.1109/ETCCE51779.2020.9350913. https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=arnumber =9350913