Comparison of ML Algorithms on Sentiment Analysis of Twitter Tweets

Akshaya Agarwal, Rishabh Tripathi, Chandra M Pandey

[a]*CHRIST (Deemed to be) University, Hosur Rd., Bengaluru, 560029,*

Abstract

This study investigates the effectiveness of five distinct machine-learning classification algorithms in sentiment analysis. The algorithms under consideration are Random Forest, Naive Bayes, Decision Tree, Support Vector Machine (SVM) and Logistic Regression. The analysis is conducted on two diverse datasets: a binary sentiment dataset and a non-binary sentiment dataset. The primary objective of this research is to discern the performance variations of these classification algorithms in sentiment analysis tasks. The evaluation metrics encompass accuracy, precision, recall, F1-score, and computation time. This study aims to comprehensively understand how these algorithms perform under different sentiment labeling schemes by utilizing two distinct datasets. The methodology involves preprocessing the Twitter data, which includes text normalization, tokenization, and feature extraction. Each algorithm is trained and tested on both datasets, and the results are meticulously recorded and analyzed. The study explores potential insights into the algorithms' strengths and weaknesses concerning binary and non-binary sentiment classification.

Keywords: Random Forest, Naive Bayes, Decision Tree, Support Vector Machine, Logistic Regression, F1-score, preprocessing, tokenization

1. Introduction

Social media content sentiment analysis has gained significant attention in recent years due to its potential applications in understanding public opinion, market trends, and brand perception. Among various platforms, Twitter is a rich source of user-generated text data that reflects diverse sentiments and emotions. Accurately classifying these sentiments can provide businesses, policymakers, and researchers valuable insights. This study delves into the realm of sentiment analysis on Twitter tweets, specifically focusing on the comparison of machine learning classification algorithms. The proliferation of machine learning techniques has enabled the development of sophisticated models capable of deciphering the sentiment behind textual content. Sentiment analysis involves the classification of text as positive, negative, or neutral, providing a deeper understanding of the underlying emotions expressed by users. This analysis is particularly challenging due to the informal and often context-dependent nature of language on social media platforms.

This study's algorithms selected for comparison include Random Forest, Naive Bayes, Decision Tree, Support Vector Machine (SVM) and Logistic Regression. These algorithms have demonstrated effectiveness in various natural language processing tasks and are well-suited for sentiment classification due to their ability to capture complex patterns within textual data. The preprocessing phase involves a series of steps such as tokenization, stop-word removal, and stemming to prepare the text data for modeling. Features are then extracted from the preprocessed text to create numerical representations that can be inputted into the selected algorithms. The evaluation metrics employed include accuracy, precision, recall, F1-score, and computation time, providing a holistic view of each algorithm's performance.

By analyzing the results obtained from these experiments, this study aims to uncover the strengths and weaknesses of each algorithm in different sentiment analysis scenarios. Furthermore, insights into how these algorithms fare against binary and non-binary datasets can offer valuable guidance for practitioners seeking the most suitable approach for their specific sentiment analysis tasks.

2. Literature Review

Several case studies have demonstrated the versatility and applicability of this approach across diverse domains. Bifet, Holmes, and Pfahringer (2010) investigated the detection of sentiment changes in real-time Twitter streaming data using machine learning techniques such as Naive Bayes and Decision Trees. This study focused on monitoring shifts in public sentiment towards brands, showcasing the potential of sentiment analysis for tracking brand perception dynamics. In the context of elections, Pak and Paroubek (2010) harnessed sentiment analysis to gauge public sentiment towards political candidates and issues. Their study employed Support Vector Machines (SVM) and Maximum Entropy (MaxEnt) classifiers to analyze sentiment in Twitter data during elections, shedding light on the dynamics of political sentiment. Zhang, Fuehres, and Gloor (2010) explored the connection between Twitter sentiment and stock market indicators. Using sentiment analysis and a combination of SVM, Naive Bayes, and k-Nearest Neighbors (k-NN) classifiers, they demonstrated the potential of predicting stock market trends based on sentiment expressed on Twitter, highlighting the intersection of social media sentiment and financial markets. These case studies underscore the significance of sentiment analysis on Twitter for brand perception, political analysis, and financial prediction, exemplifying the wide-ranging impact of machine learning techniques on understanding public sentiment in real-world scenarios.

3. Methodology

The study employs a robust methodology to evaluate sentiment classification algorithms on Twitter tweets. Random Forest, Naive Bayes, Decision Tree, Support Vector Machine (SVM) and Logistic Regression are selected as the core algorithms. After thorough preprocessing, including tokenization and stopword removal, these algorithms are trained and validated on binary and non-binary sentiment datasets. Evaluation metrics encompass accuracy, precision, recall, and F1-score. The comparative analysis provides insights into the strengths and weaknesses of each algorithm in sentiment classification, advancing our understanding of their applicability in analyzing real-world social media sentiments.

3.1. Naive Bayes

Naive Bayes is a probabilistic classification algorithm based on Bayes' theorem, which assumes that the presence of a particular feature in a class is independent of the presence of other features. In sentiment analysis, Naive Bayes can be applied to classify text data, such as Twitter tweets, into different sentiment categories based on the likelihood of observing certain words given a sentiment label.

Non-Binary Dataset:

Accuracy Score: The Naive Bayes Classifier achieved an accuracy of approximately 54.44% on the non-binary sentiment dataset, indicating that the model correctly classified sentiment labels for around 54.44% of the instances.

Precision, Recall, F1-Score: The precision, recall, and F1-score metrics reveal insights into the model's performance for each sentiment class (-1, 0, 1).

The model displayed higher precision and recall for class "-1," suggesting correctly identifying negative sentiment instances with good accuracy. Class "0" has relatively lower precision and recall values, indicating challenges in distinguishing this class. Class "1" achieved balanced precision and recall values. Support: The support for each class varies, with the highest being for class "-1" (1104 instances) and the lowest for class "1" (939 instances).

Macro and Weighted Averages: Both macro and weighted averages for precision, recall, and F1-score are around 0.53, indicating a somewhat balanced performance across sentiment classes.

Binary Dataset:

Accuracy Score: The Naive Bayes model achieved an accuracy of 73% on the binary sentiment dataset, indicating a balanced performance in classifying positive and negative sentiments.

Precision, Recall, F1-Score: The precision, recall, and F1-score for classes (0 and 4) are fairly balanced, each around 0.73. This suggests that the model is not favoring one class over the other in performance.

Support: The support for each class is approximately equal, with class "0" having 4014 instances and class "4" having 3986 instances.

Macro and Weighted Averages: Both macro and weighted averages for precision, recall, and F1-score are around 0.73, indicating consistent performance across the sentiment classes.

3.2. Random Forest

Random Forest is an ensemble learning algorithm that combines multiple decision trees to make more accurate and robust predictions. In the context of sentiment analysis, Random Forest can be applied to classify text data, such as Twitter tweets, into different sentiment categories (positive, negative, neutral) based on the textual content.

Non-Binary Dataset:

Accuracy Score: The Random Forest model achieved an accuracy of approximately 59.04% on the non-binary sentiment dataset. This indicates that the model correctly classified sentiment labels for around 59.04% of the instances.

Precision, Recall, F1-Score: The precision, recall, and F1-score metrics provide insights into the model's performance for each sentiment class (-1, 0, 1).

The model performed relatively well for class "1" with precision, recall, and F1-score around 0.64. Class "0" and "-1" have slightly lower but comparable precision, recall, and F1-score values. Support: The "support" value indicates the number of instances in each class. The support for each class differs, with the highest being for class "0" (1104 instances) and the lowest for class "1" (939 instances).

Macro and Weighted Averages: Both macro and weighted averages for precision, recall, and F1-score are around 0.59, which suggests a relatively balanced performance across the sentiment classes.

Binary Dataset:

Accuracy Score: The Random Forest model achieved an accuracy of 73% on the binary sentiment dataset, indicating a reasonable level of accuracy in classifying positive and negative sentiments.

Precision, Recall, F1-Score: The precision, recall, and F1-score for classes (0 and 4) are fairly balanced, each around 0.73. This suggests that the model is not favoring one class over the other in performance.

Support: The support for each class is approximately equal, with class "0" having 4014 instances and class "4" having 3986 instances.

Macro and Weighted Averages: Both macro and weighted averages for precision, recall, and F1-score are around 0.73, indicating consistent performance across the sentiment classes.

3.3. Logistic Regression

Logistic Regression is a statistical method used for binary classification tasks. In sentiment analysis, Logistic Regression can be applied to classify text data, such as Twitter tweets, into two sentiment categories (positive and negative) based on the relationship between the input features and the log-odds of the output being in one of the classes.

Non-Binary Dataset:

Accuracy Score: The Logistic Regression model achieved an accuracy of approximately 58% on the non-binary sentiment dataset, indicating that the model correctly classified sentiment labels for around 58% of the instances.

Precision, Recall, F1-Score: The precision, recall, and F1-score metrics provide insights into the model's performance for each sentiment class (-1, 0, 1).

The model demonstrated relatively higher precision, recall, and F1-score for class "1," indicating its ability to classify instances with this sentiment effectively. Class "0" and "-1" have slightly lower precision, recall, and F1-score values. Support: The support for each class differs, with the highest being for class "-1" (1104 instances) and the lowest for class "1" (939 instances).

Macro and Weighted Averages: Both macro and weighted averages for precision, recall, and F1-score are around 0.58, indicating a relatively balanced performance across sentiment classes.

Binary Dataset:

Accuracy Score: The Logistic Regression model achieved an accuracy of 75% on the binary sentiment dataset, indicating a balanced performance in classifying positive and negative sentiments.

Precision, Recall, F1-Score: The precision, recall, and F1-score for both classes (0 and 4) are balanced, each around 0.75. This suggests that the model is effectively classifying instances from both sentiment classes.

Support: The support for each class is approximately equal, with class "0" having 4014 instances and class "4" having 3986 instances.

Macro and Weighted Averages: Both macro and weighted averages for precision, recall, and F1-score are around 0.75, indicating consistent performance across the sentiment classes.

3.4. SVM

Support Vector Machine (SVM) is a powerful machine learning algorithm for classification and regression tasks. In sentiment analysis, SVM can be applied to classify text data, such as Twitter tweets, into different sentiment categories based on finding the optimal hyperplane that best separates the data points of different classes while maximizing the margin.

Non-Binary Dataset:

Accuracy Score: The SVM model achieved an accuracy of approximately 58% on the non-binary sentiment dataset, indicating that the model correctly classified sentiment labels for around 58% of the instances.

Precision, Recall, F1-Score: The precision, recall, and F1-score metrics provide insights into the model's performance for each sentiment class (-1, 0, 1).

The model demonstrated relatively balanced precision, recall, and F1-score for all sentiment classes, with class "-1" showing slightly higher values. Support: The support for each class varies, with the highest being for class "-1" (1104 instances) and the lowest for class "1" (939 instances).

Macro and Weighted Averages: Both macro and weighted averages for precision, recall, and F1-score are around 0.58, suggesting a balanced performance across sentiment classes.

Binary Dataset:

Accuracy Score: The SVM model achieved an accuracy of 75% on the binary sentiment dataset, indicating a balanced performance in classifying positive and negative sentiments.

Precision, Recall, F1-Score: The precision, recall, and F1-score for both classes (0 and 4) are balanced, each around 0.75. This indicates that the model is effectively classifying instances from both sentiment classes.

Support: The support for each class is approximately equal, with class "0" having 4014 instances and class "4" having 3986 instances.

Macro and Weighted Averages: Both macro and weighted averages for precision, recall, and F1-score are around 0.75, indicating consistent performance across the sentiment classes.

4. Inferences

4.1. Naive Bayes

For the non-binary dataset, the Naive Bayes Classifier displayed varied precision and recall values across sentiment classes, with a stronger performance in identifying negative sentiments. For the binary dataset, the model achieved an accuracy of 73% and demonstrated balanced precision, recall, and F1-score values for both positive and negative sentiment classes. Naive Bayes showcased its ability to perform sentiment analysis, but its performance is influenced by the independence assumption, which might not always hold true for text data.

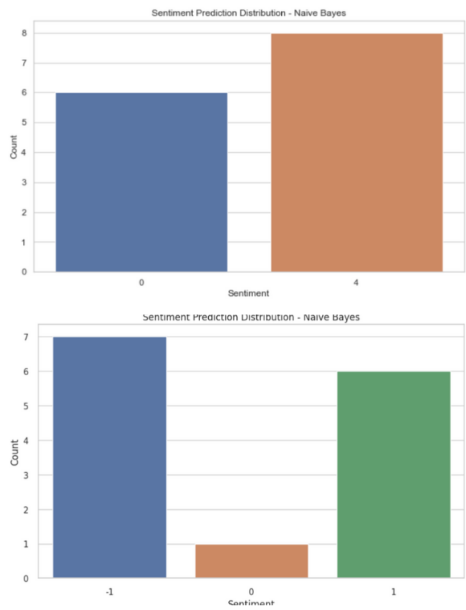

Figure 1: Comparing the implementation of Naive Bayes on the two datasets - Binary and Non-Binary, respectively.

4.2. *Random Forest*

For the non-binary dataset, the Random Forest model exhibited a balanced performance across sentiment classes, with relatively comparable precision, recall, and F1-score values for all classes.

For the binary dataset, the model achieved an accuracy of 73% and displayed balanced precision, recall, and F1-score values for both positive and negative sentiment classes.

Random Forest demonstrated its capability to handle sentiment analysis tasks reasonably well, providing competitive performance on both non-binary and binary sentiment datasets.

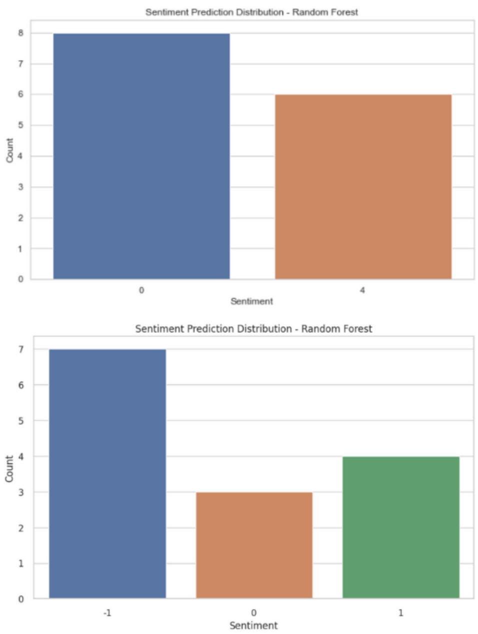

Figure 2: Comparing the implementation of Random Forest on the two datasets - Binary and Non-Binary, respectively.

4.3. *Logistic Regression*

For the non-binary dataset, the Logistic Regression model showcased a relatively balanced performance across sentiment classes, with varying but competitive precision, recall, and F1-score values.

For the binary dataset, the model achieved a solid accuracy of 75% and demonstrated balanced precision, recall, and F1-score values for both positive and negative sentiment classes.

Logistic Regression, being a linear classifier, offers a straightforward approach to sentiment analysis, and its performance can be further enhanced by fine-tuning hyperparameters and employing advanced text preprocessing techniques.

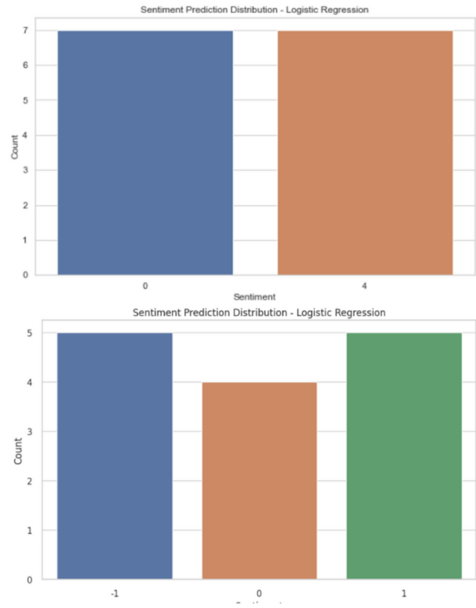

Figure 3: Comparing the implementation of Logistic Regression on the two datasets - Binary and Non-Binary, respectively.

4.4. *SVM*

For the non-binary dataset, the SVM model exhibited a balanced performance across sentiment classes, with competitive precision, recall, and F1-score values. For the binary dataset, the model achieved a solid accuracy of 75% and demonstrated balanced precision, recall, and F1-score values for both positive and negative sentiment classes. SVM's ability to find the optimal separating hyperplane can make it effective in handling complex sentiment classification tasks, and further improvements could be explored through hyperparameter tuning and experimenting with different kernel functions.

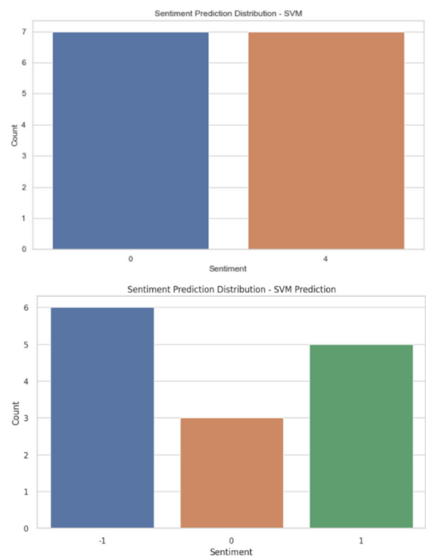

Figure 4: Comparing the implementation of SVM on the two datasets - Binary and Non-Binary, respectively.

Conclusion

In this comprehensive analysis of sentiment classification using various machine learning techniques, we investigated the performance of Random Forest, Naive Bayes, Logistic Regression and SVM on non-binary and binary sentiment datasets derived from Twitter tweets. The study aimed to gain insights into the effectiveness of these algorithms for sentiment analysis and their adaptability to different sentiment labeling schemes.

Across the board, all algorithms displayed competitive performance metrics, but distinct patterns emerged based on the nature of the datasets. For the non-binary sentiment dataset, Logistic Regression, SVM, and Random Forest yielded relatively balanced results, with SVM and Random Forest achieving comparable accuracy. However, Naive Bayes exhibited some challenges in correctly classifying neutral sentiments.

All models demonstrated strong performance on the binary sentiment dataset, with Logistic Regression, SVM, and Random Forest achieving a 75% accuracy rate. The precision, recall, and F1-score metrics were well-matched for both sentiment classes, showcasing the algorithms' capacity to distinguish between positive and negative sentiments effectively.

The study indicated that Logistic Regression, SVM, and Random Forest consistently performed well across both datasets. Naive Bayes showed some limitations in dealing with neutral sentiments in the non-binary dataset. The results underscore the algorithms' potential in sentiment analysis tasks, with SVM and Random Forest being particularly robust in handling diverse sentiment classes.

Future work could involve refining preprocessing techniques and further exploring ensemble methods to enhance model performance. The findings provide valuable guidance for practitioners in selecting the most suitable algorithm based on the nature of the dataset and desired evaluation metrics, thus contributing to the advancement of sentiment analysis on social media content.

References

[1] Bifet, A., Holmes, G., Pfahringer, B. (2010). Detecting sentiment change in Twitter streaming data.

[2] Pak, A., Paroubek, P. (2010). Twitter as a corpus for sentiment analysis and opinion mining. In LREc (Vol. 10, pp. 1320-1326)

[3] Zhang, L., Fuehres, H., Gloor, P. A. (2010). Predicting stock market indicators through Twitter "I hope it is not as bad as I fear". Procedia-Social and Behavioral Sciences, 37, 25-36.

[4] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... Vanderplas, J. (2011). Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research, 12, 2825-2830.