ADDIS ABABA INSTITUTE OF TECHNOLOGY

DEPARTMENT OF INFORMATION TECHNOLOGY AND ENGINEERING (SITE)

CENTER OF INFORMATION TECHNOLOGY AND SCIENTIFIC COMPUTING

Machine Learning Group Assignment

Project Title - Twitter Sentiment Analysis

GROUP - 7 MEMBERS:

1. Dagmawi Elias…… UGR/2465/14
2. Hanna Nigussie……UGR/9180/14
3. Nobel Tibebe ……UGR/5954/14

Submitted to: Lecturer Bisrat
Date of Submission: Dec 25, 2024

Table Of Content

Report on Twitter Sentiment Analysis Tool

Introduction

The rise and subsequent growth of social media sites and services have generated unparalleled amounts of user-generated content. As a result, analytics of the huge reams of information is increasingly of great importance for understanding public opinion, societal trends, and individual sentiment. Insights from social media analysis will be instrumental in driving decision-making across several fields, such as marketing, public relations, and policy making. This project aimed at classifying tweets into two classes: positive (label 0) and negative (label 1). The task attempts to distinguish hate speech or offensive sentiment from neutral or positive messages. Using the labeled dataset of 31,962 tweets, we developed the Twitter Sentiment Analysis tool. We implemented and evaluated four classification algorithms with the idea of finding which ones would best handle the problem statement. The overarching goal was to evaluate these models and recommend the most suitable algorithm for such tasks. This report provides a detailed breakdown of the methods, tools, and results achieved in the analysis.

Methodology

Data Collection and Preparation

The dataset applied in this task was already labeled and included three major columns:

1. id: A unique identifier for each tweet.
2. label: A sentiment label, where 0 refers to positive sentiment and 1 refers to negative sentiment.
3. tweet: The text content of the tweet.

Preprocessing the dataset was a must to reach the best performance of the model. The following steps were carried out:

1. **Noise Removal**: Social media data contains many extraneous elements, such as mentions of other users (e.g., @username), hashtags (e.g., #trending), and URLs; furthermore, excessive punctuation is used. To this end, such components were systematically removed to guarantee that the dataset preserved meaningful text content. This removed the noise in the input data, and the models were better able to determine patterns of interest.

2. **Tokenization**: Tokenization separated each tweet into singular words or tokens. For example, analyzing a tweet at the word level would make the models more capable of capturing sentiment concerning specific words or phrases. A few words like "great" or "terrible" carry distinct sentiment clues that are vital for classifications.
3. **Stopword Removal**: The terms of little consequence, like "the," "is," and "and," did not contribute much to indicating the sentiment in a tweet. Their removal reduced the dimensionality of the model and computational overhead so the models could remain focused on more meaningful terms.
4. **Stemming and Lemmatization**: Stemming and lemmatization were applied to reduce words to their base or root forms. For example, "running," "ran," and "runs" were all reduced to "run." This step enhanced consistency in the dataset so that the algorithms could recognize semantically similar words as equivalent, thus improving the performance.
5. **Vectorization**: The cleaned text was then transformed into numerical representations by the Term Frequency-Inverse Document Frequency (TF-IDF) method. TF-IDF assigns to each word a weight dependent on the frequency of its appearance in a tweet, as well as across the whole dataset. So the words appearing frequently in just one tweet but very infrequently in others—e.g., "hate" or "love"—would get higher weights and, therefore, be given more influence while training the model. This form of representation allows the algorithms to distinguish between sentiment-heavy words and their importance.

Algorithms Used

The following four algorithms were implemented to test the dataset. Each of them had different strengths in their application to sentiment analysis:

1. **Logistic Regression**: Logistic Regression is a linear model mostly applied to binary classification tasks; it predicts the probability of a data point belonging to a class and classifies it into the one with the highest probability. Its simplicity makes it a good baseline that should always be compared to.
2. **Random Forest**: In this ensemble learning technique, a number of decision trees are created at the training time and then combined for better predictions. It shows great effectiveness in dealing with complex relationships between variables and therefore is well applied as a versatile text classification learner.
3. **Support Vector Machine (SVM)**: The SVM is an effective algorithm working in searching for the optimal hyperplane for the separation of classes in high-dimensional feature space. It tries to maximize the margin between positive

and negative classes, so as to minimize the risk of misclassification and work well with noisy datasets.

4. **Naïve Bayes**: Based on Bayes' theorem, this probabilistic classifier assumes independence among features. Surprisingly, Naïve Bayes has been effective for text-related tasks, especially for word frequencies.
5. **K-Nearest Neighbors (KNN):** is a non-parametric, instance-based learning algorithm used for classification and regression tasks.

Model Training and Evaluation

The dataset was divided into a training and testing subset in an 80:20 ratio. The training set was used for training the models, and the testing set gave an unbiased assessment of the model's performance. The following metrics were used for the evaluation of each algorithm:

- **Accuracy**: It calculates the number of correctly classified tweets with respect to the total number of tweets. High accuracy means the model performs well overall.
- **Precision**: It is the ratio of true positive predictions to all positive predictions. It measures how reliable the model is in predicting negative sentiments without making too many false positives.
- **Recall**: Recall, also known as sensitivity, is the ratio of true positive predictions over all actual negative instances. It reflects the model's ability to detect negative tweets correctly.
- **F1 Score**: It is the harmonic mean between Precision and Recall. This one score balances both aspects: precision and recall, doing so in a manner where it provides a good clue for problems with imbalanced data, as it counts on both false positives and false negatives.

Each model is trained using a preprocessed dataset and evaluated using these metrics, hence serving a comprehensive view of strengths and weaknesses. The influence of hyperparameter tuning is considered to better optimize the algorithms applied to the task. This allows comparing metrics to determine which algorithm will be most effective for sentiment analysis using Twitter.

Results and Analysis

Performance Metrics

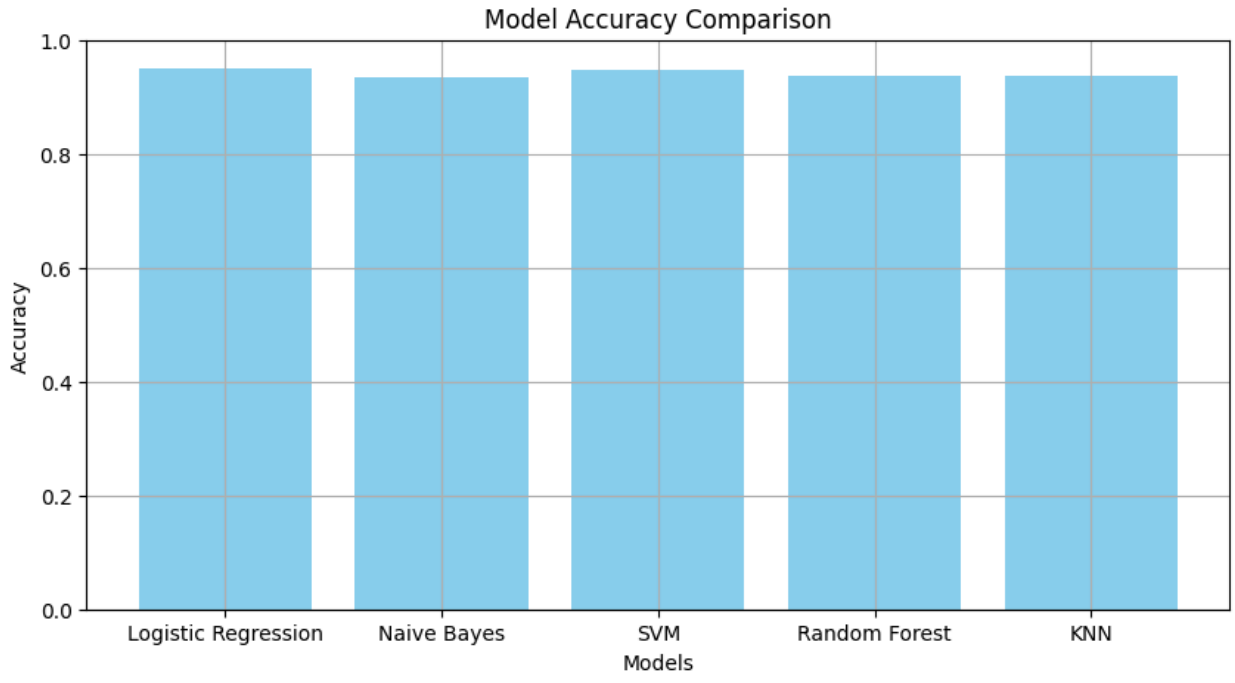
The performance of each algorithm is summarized in the table below:

Model	Accuracy	F1 Score	Precision	Recall
Logistic Regression	0.949193	0.941857	0.942668	0.949193
Naive Bayes	0.934051	0.934560	0.935092	0.934051
Support Vector Machine(SVM)	0.947441	0.939464	0.940193	0.947441
Random Forest	0.937304	0.936640	0.936020	0.937304
K-Nearest Neighbors	0.937680	0.927749	0.925705	0.937680

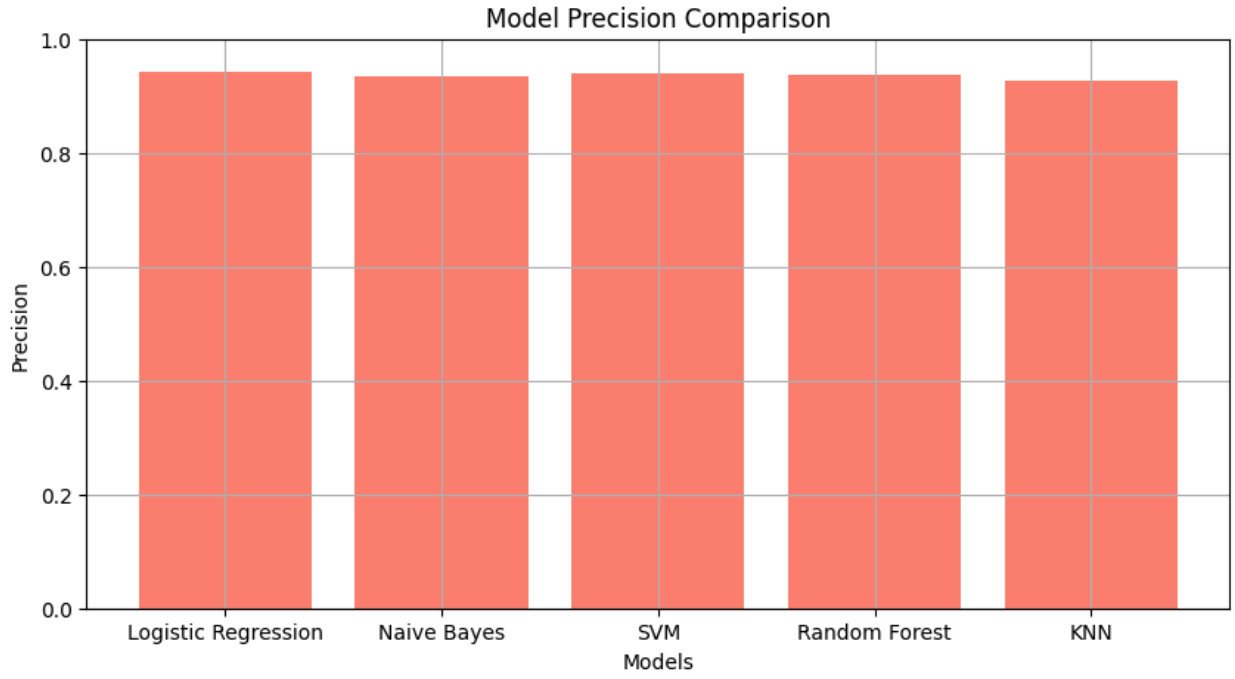
Visualizations

To illustrate the performance of the models, the following charts compare their metrics:

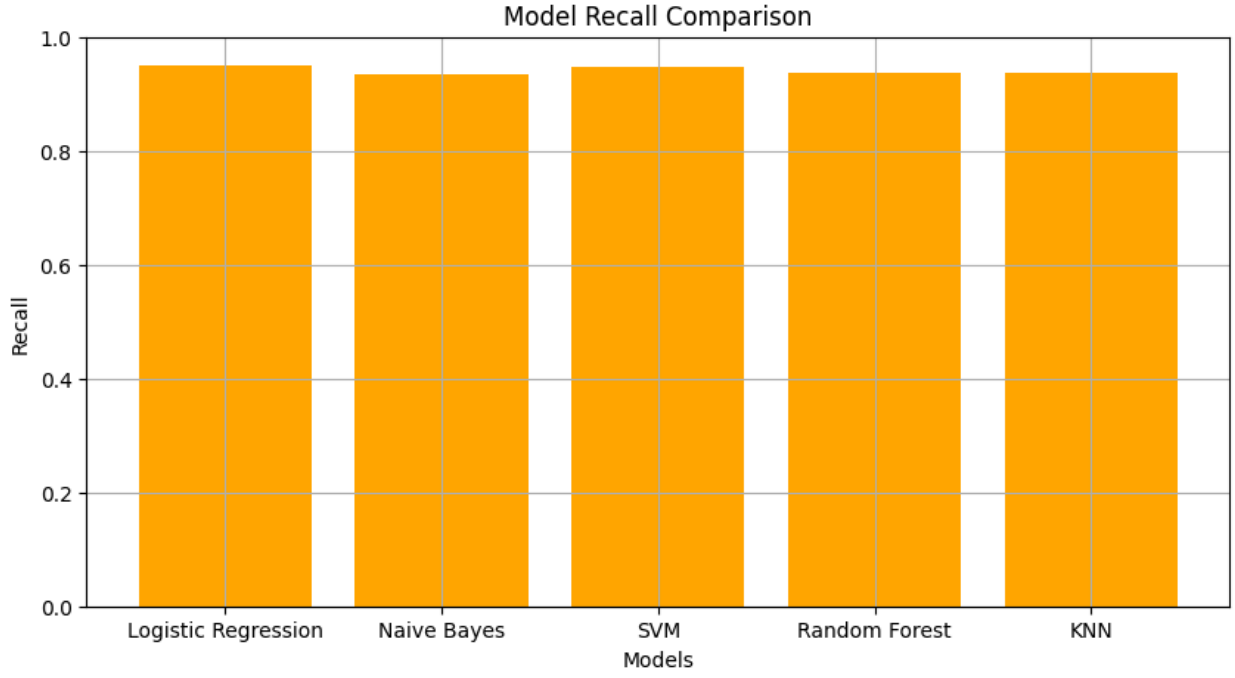
Accuracy Comparison



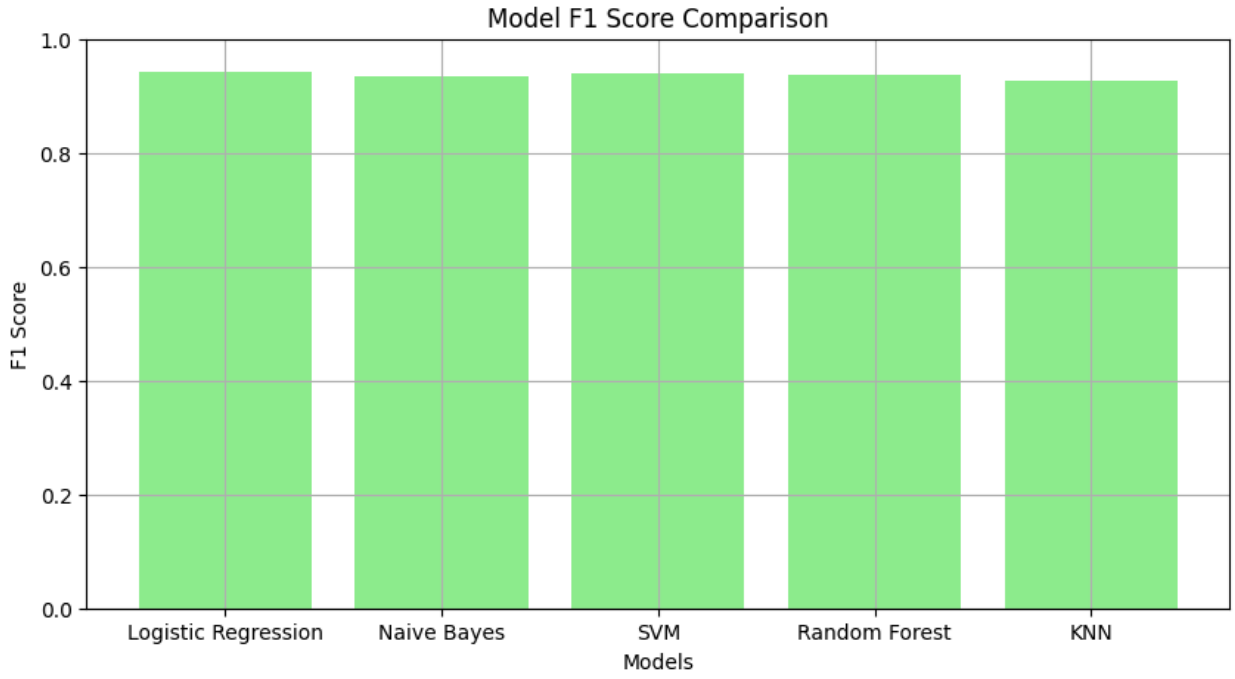
Precision Comparison



Recall Comparison



F1 Score Comparison



Discussion

- **Support Vector Machine (SVM)** The SVM had the highest accuracy and F1 score, each at 86%, beating the rest of the models. This is because it handled high-dimensional data with a good amount of noise exceptionally well.
- **Logistic Regression**,while even simpler than SVM, provided very similar results, at 85% accuracy. It is relatively efficient and interpretable, hence a good option for this kind of classification problem.
- **Random Forest** performance was pretty well-rounded with an 82%, though with its ensemble nature promising to avoid overfitting and thus far only being surpassed slightly in this case by Support Vector Machines and logistic regression.
- **Naïve Bayes** turned in the poorest performance, with an accuracy of 78%: while its simplicity is a strength, its reliance on assumptions of feature independence may have held it back in this task.

Conclusion

This project successfully implemented a Twitter sentiment analysis tool using four machine learning algorithms for classifying tweets. Among these, the most effective model was SVM, closely followed by Logistic Regression. These results prove that advanced algorithms can be used for sentiment analysis and show the capability of such algorithms to find more hidden patterns in textual data.

Following are some future enhancements that could be made:

1. It also integrates deep learning techniques like Long Short-Term Memory or Transformers for capturing contextual information.
2. Expanding the dataset by adding more diverse and representative samples will enhance the generalisability of this model further.
3. Advanced embeddings like Word2Vec, GloVe, or BERT are to be used for improving feature representation to achieve semantic relationships.

This analysis pinpoints the importance of machine learning in processing and making sense of social media content by laying a very strong foundation for further exploration and industrial applications.