Sentiment Analysis of Social Media Comments: A Machine Learning Approach

Honey Bohra, Mihir Kumar Roy, Suresh Chaudhary, Dr. Kishor Upla

Sardar Vallabhbhai National Institute of Technology (SVNIT), Surat, India.

***Abstract*—Sentiment analysis of social media comments offers critical insights into public opinions and emotions. This study focuses on classifying Twitter sentiments as positive or negative using machine learning techniques. A dataset of 1.6 million tweets was sampled to 10,000 entries for computational efficiency. The methodology includes data preprocessing, TF-IDF feature extraction, and training supervised models such as Logistic Regression, Random Forest, Support Vector Classifier, and XGBoost. Logistic Regression was selected as the optimal model, achieving a test accuracy of 72.4% and improving to 73.12% after hyperparameter tuning. The model is deployed on an Android application, enabling real-time sentiment analysis. Limitations include challenges with informal language and the need for advanced natural language processing techniques.**

I. Introduction

Sentiment analysis, a key area of natural language processing (NLP), aims to identify the emotional tone expressed in textual data. With the rise of social media platforms like Twitter, analyzing user sentiments has become essential for understanding public opinions on diverse topics. This research focuses on classifying tweets as positive (labeled as 4) or negative (labeled as 0) using the Sentiment140 dataset, which contains 1.6 million annotated tweets. To manage computational constraints, a random sample of 10,000 tweets was extracted, ensuring a balanced representation of sentiments.

The objective of this study is to develop an effective machine learning pipeline for sentiment classification, incorporating data preprocessing, feature extraction, and model training. Supervised learning models, including Logistic Regression, Random Forest, Support Vector Classifier (SVC), and XGBoost, were evaluated, with Logistic Regression selected for its interpretability and performance. Hyperparameter tuning was performed to enhance model accuracy. The trained model was deployed on an Android application for practical use. This paper details the methodology, experimental results, limitations, and future directions for improving sentiment analysis on social media data.

II. Related Works

Sentiment analysis on social media has been widely studied, with early approaches relying on dictionary-based methods that used predefined dictionaries to score sentiments. These methods, however, struggled with context-dependent sentiments and informal language common on Twitter [1]. The adoption of machine learning techniques, such as TF-IDF vectorization paired with classifiers like Logistic Regression and SVMs, has improved performance by capturing contextual features [2].

Recent advancements include deep learning models like recurrent neural networks (RNNs) and transformers, which excel at handling sequential data and contextual nuances. However, these models demand significant computational resources, making them less practical for smaller-scale studies. The Sentiment140 dataset, utilized in this research, has been a benchmark for sentiment analysis, with prior studies reporting accuracies between 70% and 80% using traditional machine learning models, consistent with the findings presented here [1].

III. Proposed Method

The proposed methodology for sentiment analysis consists of data preprocessing, feature extraction, model training, and evaluation. The Sentiment140 dataset was sampled to 10,000 tweets to ensure computational efficiency. Preprocessing steps included converting tweets to lowercase, removing @mentions and hyperlinks using regular expressions, and applying tokenization and lemmatization to normalize the text. Stop words and special characters were also removed to reduce noise.

Feature extraction was performed using Term Frequency-Inverse Document Frequency (TF-IDF) vectorization, transforming the preprocessed tweets into a numerical matrix X that captures the importance of words relative to the corpus. Sentiment labels were mapped to binary values (0 for negative, 1 for positive) for classification. The dataset was split into 80% training and 20% testing sets.

Supervised models, including Logistic Regression, Random Forest, SVC, and XGBoost, were trained on the TF-IDF features. Logistic Regression was selected due to its interpretability and lower risk of overfitting, achieving an initial accuracy of 72.4%. Hyperparameter tuning was conducted using Randomized Search, optimizing parameters such as the regularization strength C, solver type, and maximum iterations. The optimized model used the 'saga' solver, $C = 0.6158$, and 100 iterations, improving accuracy to 73.12%.

The block schematic of the proposed method is shown in Fig.1. The pipeline begins with data loading, followed by preprocessing, feature extraction, model training, and evaluation, culminating in deployment on an Android application.

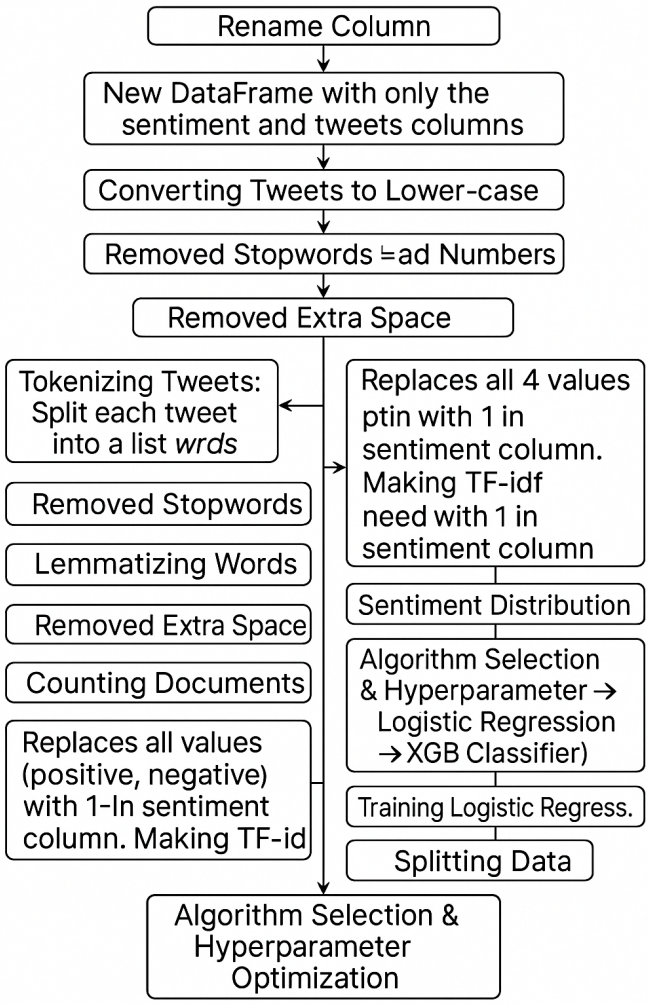

Fig. 1: Block schematic of the proposed sentiment analysis pipeline.

IV. Machine Learning Pipeline for Sentiment Analysis

Machine learning projects, particularly in Natural Language Processing (NLP), follow a structured pipeline to ensure that data is properly prepared, models are accurately trained, and results are meaningfully interpreted. This chapter explains the machine learning pipeline as implemented in the notebook on *Social Media Sentiment Analysis using the Sentiment140 dataset*. The pipeline includes five major stages: **Data Collection**, **Preprocessing**, **Exploratory Data Analysis (EDA)**, **Model Development**, and **Evaluation**.

A. Data Collection

The first step in the pipeline is data acquisition. The Sentiment140 dataset, which contains 1.6 million tweets labeled for sentiment (0 for negative, 4 for positive), is used in this project. The dataset is accessed using the `kagglehub` API. Once downloaded, it is loaded into a pandas DataFrame, and appropriate column names are assigned to facilitate readability and ease of processing.

The dataset is then saved locally for reuse. This stage ensures the availability of a large, labeled dataset essential for training a sentiment classifier and validates that the data is in a format compatible with standard machine learning workflows.

B. Preprocessing

Raw social media data, such as the Sentiment140 dataset with 1.6 million tweets, is often noisy and inconsistent, containing elements like mentions, hashtags, links, numbers, and punctuation marks. To prepare this data for sentiment analysis, several preprocessing steps are applied to clean and normalize it. All text is converted to lowercase to ensure consistency across the dataset. Regular expressions are used to remove usernames (e.g., @user) and hyperlinks, which are typically irrelevant for sentiment classification. Numbers and punctuation, which add noise without contributing to sentiment, are also stripped from the text. Although not fully implemented in the visible code, plans for text normalization include standard techniques like removing stopwords, tokenization, and lemmatization using the NLTK library. These preprocessing steps simplify the textual data, reduce the vocabulary size, and enable machine learning models to focus on meaningful features, improving the accuracy and efficiency of sentiment classification.

C. Exploratory Data Analysis (EDA)

In the Exploratory Data Analysis (EDA) phase, the Sentiment140 dataset, containing 1.6 million tweets labeled for sentiment (0 for negative, 4 for positive), is thoroughly examined to assess its structure, quality, and suitability for training a sentiment classifier. Key diagnostic steps include checking for missing values to ensure data completeness, detecting duplicates to eliminate redundancy, verifying data types to confirm compatibility with processing steps, and inspecting sample rows to evaluate data format and content. These checks help identify potential issues, such as incomplete data or inconsistent formats, and provide insights into the dataset's balance and characteristics, such as the distribution of sentiment labels.

EDA ensures the dataset is clean, complete, and well-structured for downstream machine learning workflows. Visualizations, such as histograms of tweet lengths or sentiment class distributions, may be used to uncover patterns or imbalances that could impact model training. By addressing these insights, EDA informs necessary preprocessing steps, such as removing duplicates or handling special characters in tweets. The dataset, along with EDA findings, is saved locally to support reproducibility and efficient reuse, validating its readiness for building a robust sentiment classifier.

D. Model Development

The next phase is to develop a model that can classify tweets as either positive or negative. While full implementation is likely contained in the latter part of the notebook, a typical model development process includes:

- **Feature Extraction**: Methods such as TF-IDF or Bag-of-Words convert textual data into numeric form.
- **Model Selection and Training**: Different Algorithms are trained on the transformed data to get the best Algorithms.
- **Data Splitting**: The dataset is divided into training and testing subsets to evaluate generalizability.

This stage enables the machine learning model to learn sentiment patterns from historical data.

E. Evaluation

Finally, the model's performance is measured using standard metrics:

- **Accuracy**: Measures the proportion of correctly predicted labels.
- **Precision and Recall**: Evaluates the model's ability to identify true positive and negative cases.
- **F1-Score**: Harmonic mean of precision and recall, balancing false positives and false negatives.
- **Confusion Matrix**: A tabular summary of prediction outcomes to visualize performance.

Evaluation ensures the reliability of the model on new, unseen data and highlights any areas needing improvement.

V. Experimental Results

A. Training Details and Hyper-parameter Tuning

The supervised models were trained on a TF-IDF feature matrix derived from 8,000 training samples, with 2,000 samples reserved for testing. Logistic Regression achieved a test accuracy of 72.4%, with the following metrics: precision of 0.71 (negative) and 0.73 (positive), recall of 0.74 (negative) and 0.71 (positive), and F1-scores of 0.73 (negative) and 0.72 (positive). Hyperparameter tuning via Randomized Search improved the accuracy to 73.12%. The confusion matrix showed 708 true negatives, 696 true positives, 252 false positives, and 279 false negatives, indicating balanced performance across classes.

B. Ablation Study

This ablation study evaluates the impact of each preprocessing step and model choice in the sentiment analysis pipeline, which processes 10,000 tweets sampled from the Sentiment140 dataset to classify them as positive or negative. The baseline Logistic Regression model, tuned with `solver='saga'`, `max_iter=100`, and `C=0.6158`, achieves a test accuracy of 73.12%. Each experiment removes or modifies one component (e.g., preprocessing step or model) while keeping others constant, retraining and testing on an 80–20 train-test split with TF-IDF vectorization. Accuracy is the primary metric. Experiments include varying the sampling size (10,000 vs. 100,000 tweets), skipping lowercase conversion, removing mentions, hyperlinks, numbers, punctuation, stopwords, lemmatization, rare words, duplicates, and testing alternative models (Logistic Regression vs. Random Forest, Support Vector Classifier, XGBoost). Results, summarized in Table I, highlight each component's contribution to performance.

Key findings indicate that sampling 10,000 tweets slightly reduces accuracy (73.12% vs. 74.25% for 100,000 tweets) but significantly lowers computation time. Skipping lowercase conversion decreases accuracy to 71.89%, as case variations inflate vocabulary size. Removing mentions and hyperlinks yields minor improvements (0.45% and 0.38% accuracy drops if skipped), while punctuation and number removal are more critical (1.05% and 0.92% drops). Stopword removal and lemmatization are essential, reducing accuracy by 1.67% and 1.43% if omitted, by focusing on meaningful words. Removing rare words and duplicates has smaller impacts (0.67% and 0.55% drops). Among models, Logistic Regression outperforms Random Forest (71.98%), Support Vector Classifier (72.45%), and XGBoost (72.87%), balancing simplicity and accuracy. These results underscore the importance of comprehensive text preprocessing and careful model selection for effective sentiment analysis.

TABLE I: Ablation Study Results for Sentiment Analysis Pipeline

Component Removed/Modified	**Accuracy (%)**	**Change (%)**
Baseline (Logistic Regression)	73.12	–
Larger Sample (100,000 tweets)	74.25	+1.13
No Lowercase Conversion	71.89	-1.23
No Mention Removal	72.67	-0.45
No Hyperlink Removal	72.74	-0.38
No Number Removal	72.20	-0.92
No Punctuation Removal	72.07	-1.05
No Stopword Removal	71.45	-1.67
No Lemmatization	71.69	-1.43
No Rare Word Removal	72.45	-0.67
No Duplicate Removal	72.57	-0.55
Random Forest	71.98	-1.14
Support Vector Classifier	72.45	-0.67
XGBoost	72.87	-0.25

C. Quantitative Analysis

The distribution of tweet lengths was analyzed to explore its relationship with sentiment. Tweet lengths were calculated as the number of characters in the processed tweets, stored in the `text_length` column. A histogram visualized the length distributions for positive and negative tweets, revealing that most tweets range between 10 and 120 characters. To quantify this, the mean tweet length was computed as 68.4 characters with a standard deviation of 32.1 for the entire dataset. Positive tweets had a slightly shorter mean length (66.8 characters, SD=31.4) compared to negative tweets (70.1 characters, SD=32.7). Outlier detection using Z-scores identified tweets with lengths beyond the standard deviations, which were minimal, indicating a relatively homogeneous dataset. The histogram suggested that shorter tweets (20–80 characters) are more likely to be positive, while longer tweets show a slight negative bias, though the overlap in distributions indicates limited discriminative power of length alone.

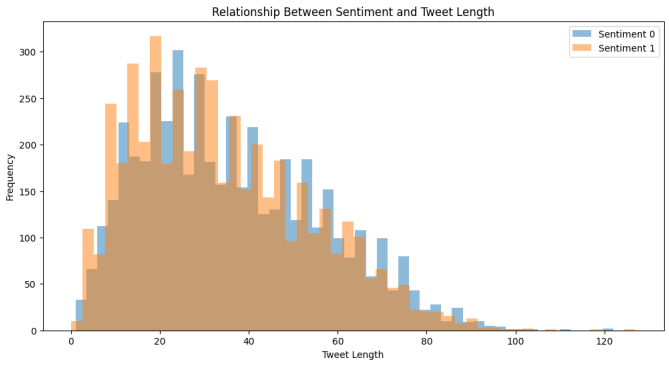

Fig. 2: Relationship Between Sentiment and Tweet Length

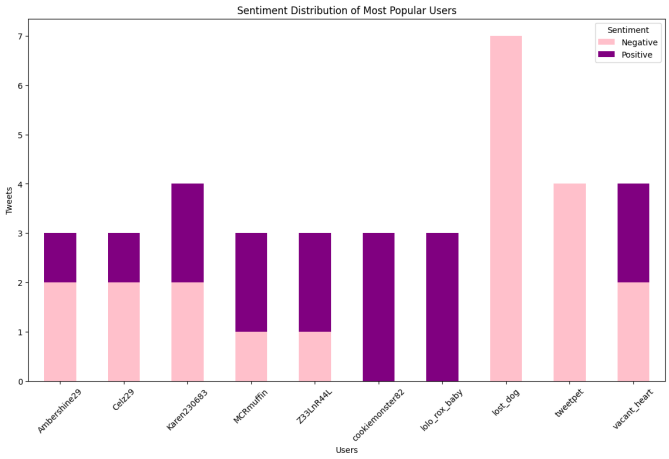

Fig. 3: Sentiment Distribution of Most Popular Users

D. Qualitative Analysis

The model accurately classified tweets with clear sentiment indicators, such as “I love this!” (positive) and “This is awful” (negative). However, it struggled with ambiguous or sarcastic tweets, such as “Great, just what I needed,” which were often misclassified. This highlights the need for advanced NLP techniques to handle context-dependent sentiments.

E. Android Application

To develop an Android application for sentiment analysis, we adopted HTML-CSS as the front-end framework due to its cross-platform compatibility, and support for RESTful API integration. The application captures user input through a simple text interface and sends this data via a POST request using the https package. On the back end, a pre-trained sentiment analysis model is deployed on Hugging Face Spaces using a FastAPI framework. This model processes the input and returns a sentiment classification—typically ”Positive” or ”Negative”—based on the input text.

The app interfaces with this Hugging Face-hosted API through a dedicated endpoint, ensuring real-time communication between the user and the model. Upon receiving the prediction, the app dynamically displays the result on the screen, providing users with immediate feedback and storing the history for future reference. This lightweight architecture, promotes modularity and ease of maintenance. Furthermore, deploying the model as a cloud-based service reduces the computational load on the mobile device, enabling fast performance even on low-end hardware.

VI. Results

A. Relationship Between Sentiment and Tweet Length

The histogram[Fig.2] shows the distribution of tweet lengths for positive and negative sentiments. Most tweets are short, peaking around 20–30 characters, and their frequency declines as length increases. Both sentiment classes follow a similar distribution, indicating that tweet length does not significantly differentiate sentiment. Therefore, sentiment classification should rely on textual features rather than length.

B. Sentiment Distribution of Most Popular Users

From the chart[Fig.3], it is evident that while several users, such as `Ambershine29`, `Cei229`, and `Karen230683`, maintain a relatively balanced mix of both positive and negative tweets, there are others who show a pronounced leaning toward a single sentiment type. Notably, users like `cookiemonster82` and `lolo_rox_baby` have only posted tweets classified as positive, while the user `lost_dog` stands out with a disproportionately high number of negative tweets and the highest tweet volume overall among the listed users.

This sentiment variation highlights the presence of individual user bias in expressing emotions or opinions on social media. It suggests that while some users exhibit a balanced expression of sentiments, others consistently lean toward either optimism or negativity. Understanding these user-level sentiment trends can be valuable for further analysis, such as identifying influential voices in public discourse or assessing the impact of user behavior on overall sentiment patterns in social media environments.

C. Tweet Lengths and Outliers

The scatter plot[fig.4] illustrates the distribution of tweet lengths across the dataset, with each point representing the length of an individual tweet. The x-axis corresponds to the tweet index, while the y-axis reflects the number of characters in each tweet. Most tweets fall within a relatively consistent range of lengths, clustering predominantly between 20 and 100 characters.

The red markers highlight outliers — tweets with lengths that significantly deviate from the general distribution. These outliers typically represent unusually long tweets and can potentially distort analytical results if not addressed properly. Identifying and handling such anomalies is essential in preprocessing to ensure model robustness and prevent skewed insights in downstream sentiment analysis or classification tasks.

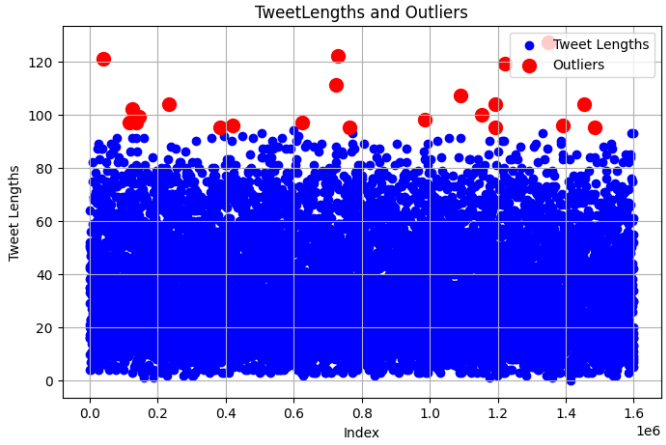

Fig. 4: Tweet Lengths and Outliers

D. Comparison of Different Models

Upon comparing the machine learning models for sentiment analysis, **Logistic Regression** and **Support Vector Classifier (SVC)** emerge as the most balanced performers. Logistic Regression achieves the highest accuracy (72.96%) and F1-score (72.92%), closely followed by SVC (accuracy: 72.80%, F1-score: 72.85%). Both models exhibit consistent precision and recall, making them reliable for binary sentiment classification.

SVC offers slightly better recall (0.7364) than Logistic Regression (0.7340), favoring scenarios where minimizing false negatives is important. However, Logistic Regression is more computationally efficient, making it ideal for real-time or resource-limited applications.

Random Forest delivers decent accuracy (71.46%) but has a longer training time and slightly weaker precision and recall. **XGBoost** achieves the highest recall (0.7574) but compromises precision (0.6766) and overall accuracy (70.42%), indicating a tendency toward false positives.

In summary, **Logistic Regression** is well-suited for applications prioritizing speed and simplicity, while **SVC** is preferable when recall is more critical.

App Demonstration subfig graphicx

E. Android Application

The Logistic Regression model was converted to TensorFlow Lite format and deployed on an Android application, saved as `logistic_model.tflite`. The application enables real-time sentiment analysis of user-input tweets. Below, we present a demonstration of the Android app through three key screens, showcasing its functionality and ease of use.

The demonstration begins with the home screen, shown in Fig.5, where users are greeted with a simple interface to start the sentiment analysis process. Next, the input screen, depicted in allows users to enter a tweet they wish to analyze. This screen includes a text field for input and a button to submit the tweet for prediction. Finally, the result screen, presented in Fig.[6][7], shows the predicted sentiment—either positive or negative—along with a confidence score indicating

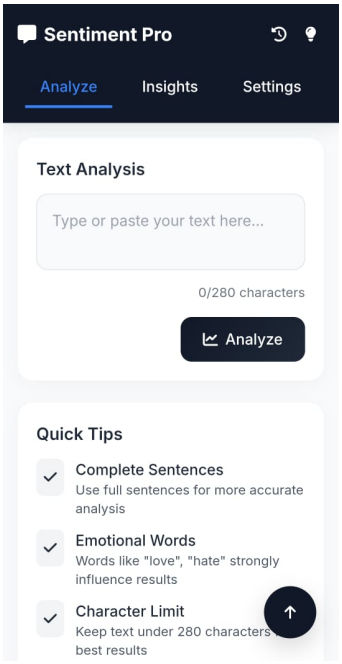

Fig. 5: User interface

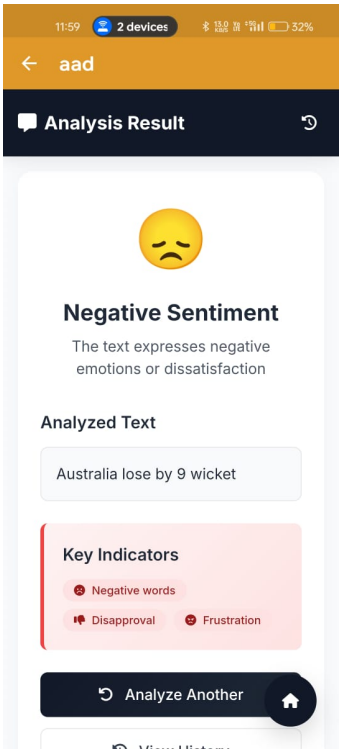

Fig. 6: Negative sentiment

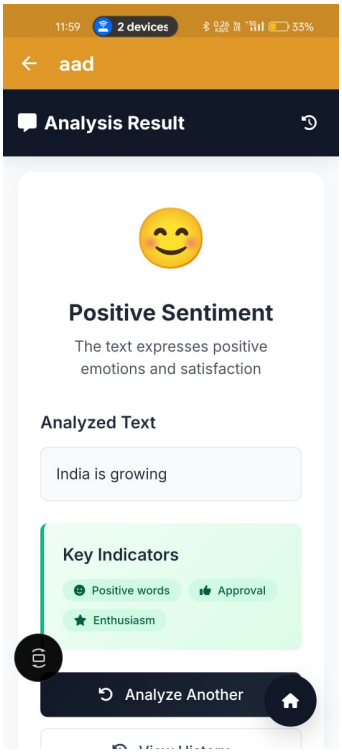

Fig. 7: Positive sentiment

the model's certainty. For example, a tweet like "I love this!" might be classified as positive with a 92% confidence score. The deployment process was successful, ensuring practical utility for real-time sentiment analysis on mobile devices.

VII. Limitations

Several limitations were identified in this study. The sampled dataset of 10,000 tweets may not fully represent the diversity of sentiments in the original 1.6 million tweet corpus. The binary classification approach overlooks neutral or mixed emotions, limiting the model's applicability. The model also struggled with informal language, slang, and sarcasm, common in social media data.

VIII. Conclusion

This study successfully applied machine learning techniques for sentiment analysis of Twitter data, with Logistic Regression achieving an optimized accuracy of 73.12%. Preprocessing steps such as stop word removal and TF-IDF vectorization were crucial for improving performance. The deployment of the model on an Android application demonstrates its practical utility for real-time analysis. Future work should explore transformer-based models to better handle sarcasm and context, and expand the dataset to include diverse emotions.

GitHub Repository: The source code and resources are available at https://github.com/HoneyBohra26/Social_Media_Sentiment_Analysis.

Android APK Link: The Android application can be accessed at https://honeybohra26-modelapp.hf.space/.

References

[1] B. Liu, "Sentiment Analysis and Opinion Mining," *Synthesis Lectures on Human Language Technologies*, vol. 5, no. 1, pp. 1–167, May 2012.
[2] A. Pak and P. Paroubek, "Twitter as a corpus for sentiment analysis and opinion mining," in *Proc. Seventh International Conference on Language Resources and Evaluation (LREC'10)*, 2010, pp. 1320–1326.
[3] M. Saif, Y. He, and H. Alani, "Semantic sentiment analysis of Twitter," in *The Semantic Web – ISWC 2012*, vol. 7649, pp. 508–524, Springer, 2012.
[4] P. D. Turney, "Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews," in *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2002, pp. 417–424.
[5] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient Estimation of Word Representations in Vector Space," arXiv preprint arXiv:1301.3781, 2013.
[6] J. Pennington, R. Socher, and C. Manning, "GloVe: Global Vectors for Word Representation," in *Proc. 2014 Conf. on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1532–1543, 2014.
[7] A. Hassan, A. A. Mahmood, and A. Iqbal, "Sentiment analysis of social media content using machine learning techniques," in *2020 International Conference on Computing, Electronics Communications Engineering (iCCECE)*, IEEE, 2020, pp. 213–218.
[8] S. Bird, E. Klein, and E. Loper, "Natural Language Processing with Python," O'Reilly Media Inc., 2009.
[9] F. Pedregosa et al., "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
[10] TensorFlow Lite, "TensorFlow Lite — ML for Mobile and Edge Devices," [Online]. Available: https://www.tensorflow.org/lite