Create a sentiment analysis tool using Python

(3EC504ME24 – Scripting Languages)

Submitted in partial fulfillment of the requirements for the degree of

Bachelor of Technology in Electronics and Communication Engineering

Prepared by

NEEL PATEL

JEMIT RATHOD

Submitted to
Prof. Shefali Waldekar
Assistant Professor
Department Of Electronics and Communication Engineering,
Nirma University.

Abstract :

Sentiment analysis is a crucial aspect of natural language processing (NLP) that enables machines to understand and categorize emotions expressed in textual data. This study focuses on leveraging machine learning algorithms to classify text into predefined emotional categories such as anger, fear, joy, sadness, and neutrality. By utilizing Support Vector Machines (SVM), TF-IDF vectorization, and sentiment polarity analysis, this research enhances the accuracy of emotion classification. The results demonstrate the effectiveness of integrating textual features with sentiment polarity for improved sentiment detection.

- ## Keywords :

Sentiment Analysis, Machine Learning, NLP, SVM Classifier, Text Processing, Emotion Classification

- ## Introduction:

With the rise of digital communication, vast amounts of textual data are generated daily through social media, customer reviews, and online interactions. Understanding human emotions in text is essential for businesses, social media monitoring, and public opinion analysis. Traditional sentiment analysis methods relied on lexicons and rule-based approaches, but recent advancements in machine learning have significantly improved accuracy. This study explores the implementation of an SVM model trained on a balanced dataset using TF-IDF features and sentiment polarity scores.

- ## Literature Survey/State of the Art Technology Available :

Sentiment analysis has evolved significantly over the years. Early approaches were primarily rule-based, using predefined sentiment lexicons to classify text [1]. These methods, while straightforward, often failed to capture contextual nuances. Machine learning techniques such as Naïve Bayes and logistic regression later improved accuracy by learning from labeled datasets. More recently, deep learning models like recurrent neural networks (RNNs) and transformers have pushed the boundaries of sentiment classification by considering context and sequential dependencies in text.

However, deep learning models require extensive computational resources and large datasets, which are not always available. As an alternative, Support Vector Machines (SVM) combined with TF-IDF vectorization has proven to be a reliable and interpretable approach for sentiment classification, offering a balance between efficiency and accuracy [4]. This study builds upon previous research by incorporating sentiment polarity as an additional feature, further refining the classification results.

- # Methodology :

he sentiment analysis model follows these key steps:

- **Data Collection:** Textual data is collected from labelled datasets.
- **Data Pre-processing:**
  - Converting text to lowercase
  - Removing URLs, mentions, and special characters
  - Tokenization and lemmatization
- **Feature Extraction:**
  - Applying TF-IDF vectorization to convert text into numerical features
  - Adding sentiment polarity scores from TextBlob
- **Model Training:**
  - Training an SVM classifier using a balanced dataset
  - Splitting data into train (80%) and test (20%) sets
- **Prediction and Evaluation:**
  - Measuring accuracy using classification reports and confusion matrix
  - Implementing a probability threshold for neutral emotion adjustment

- # Limitations :

Despite advancements in sentiment analysis, several limitations persist:

- **Computational Overhead:** Software-based models require significant computational resources, making real-time analysis challenging on resource-constrained systems.
- **Latency:** High latency in processing large datasets limits applicability in time-sensitive scenarios.
- **Scalability:** Scaling sentiment analysis to handle streaming data (e.g., real-time tweets) is inefficient without optimization.
- **Generalization:** Models may overfit or underperform on diverse datasets without robust pre processing and feature selection.

- **Flow Chart :**

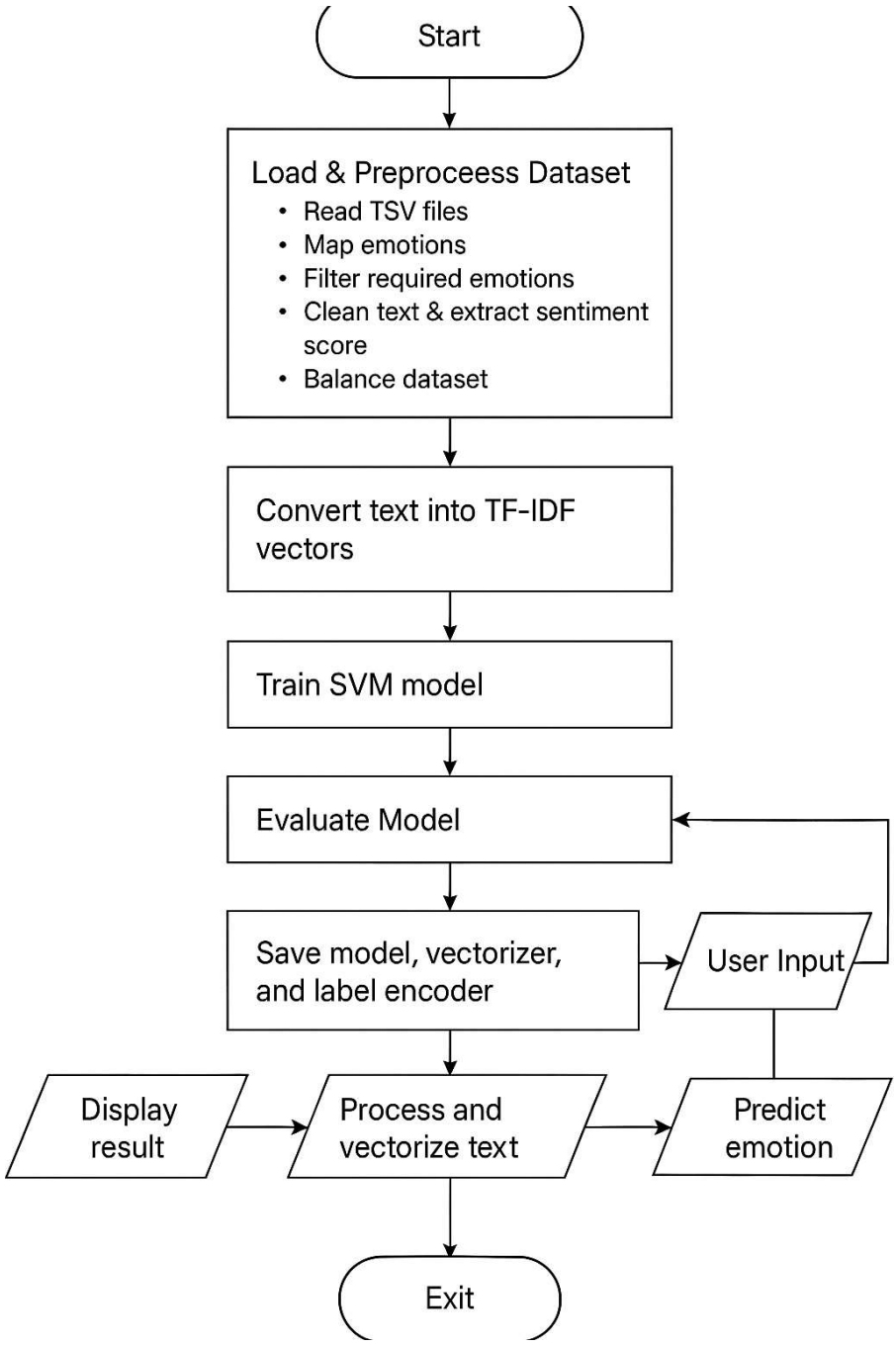

Fig 1. Flow chart for Sentiment Analysis

- **Simulation/implementation results :**

```
Enter a text to predict its emotion (type 'exit' to quit):
Enter text: I am so happy today
Predicted emotion: joy
Enter text: This is the worst day of my life
Predicted emotion: sadness
Enter text: I feel okay, nothing special
Predicted emotion: neutral
Enter text: are you mad cant you understand a single thing
Predicted emotion: anger
Enter text: I am scared to go outside alone at night.
Predicted emotion: fear
Enter text: exit
Exiting... Goodbye! 👋
```

Fig 2. Output for the Sentiment Analysis

- **Conclusion :**

In this project successfully implements an SVM-based sentiment analysis model that accurately classifies emotions in text, with the combination of TF-IDF features and sentiment polarity scores enhancing prediction accuracy. Future improvements include expanding the dataset for better generalization, using deep learning models such as BERT for improved accuracy, and deploying the model as an API for real-time sentiment analysis. This model provides a foundation for emotion classification in NLP applications, making it valuable for industries like customer feedback analysis, mental health monitoring, and social media insight