**Sentiment Analysis Tool Using Python**

**Submitted for**

**Statistical Machine Learning CSET211**

Submitted by:

**(E23CSEU0228) YASH PADAM**

**(E23CSEU0237) ANSH KAUSHIK**

Submitted to

**DR. SANCHALI DAS**

**July-Dec 2024**

**SCHOOL OF COMPUTER SCIENCE AND ENGINEERING**

A close-up of a logo

Description automatically generated

|  |  |  |
| --- | --- | --- |
| S No. | Content | Page No. |
| 1. | Abstract | 3 |
| 2. | Introduction | 3 |
| 3. | Related Work | 3 |
| 4. | Methodology | 4 |
| 5. | Hardware/Software Used | 5 |
| 6. | Experimental Results | 5 |
| 7. | Conclusions | 8 |
| 8. | Future Scope | 8 |
| 9. | Github Link | 9 |

1. **Abstract:**

This project aims to perform sentiment analysis on a dataset containing text entries labeled with different emotions. The primary objective is to classify text data into predefined emotion categories using machine learning models. We employed Natural Language Processing (NLP) techniques for data preprocessing and used three models: Naive Bayes, Logistic Regression, and Support Vector Machine (SVM) for classification. The experimental results show that the models can effectively classify emotions with reasonable accuracy, highlighting the potential for automating emotion detection in text-based data.

1. **Introduction:**

Sentiment analysis, also known as emotion detection, is a Natural Language Processing (NLP) technique that identifies the sentiment or emotional tone behind a piece of text. It is widely used in social media monitoring, customer feedback analysis, and market research to gauge public sentiment. The objective of this project is to develop a machine learning model capable of classifying text into specific emotion categories such as joy, sadness, fear, anger, and more, using a dataset labeled with these emotions.

1. **Related Work:**

Various approaches have been proposed for sentiment analysis, ranging from lexicon-based methods to machine learning and deep learning techniques. Traditional machine learning models such as Naive Bayes and Logistic Regression have been popular due to their simplicity and efficiency. However, recent advancements in deep learning, including the use of LSTM and Transformer-based models like BERT, have shown significant improvements in handling the nuances of human emotions. This project explores the use of classical machine learning algorithms to evaluate their effectiveness in emotion detection tasks on a labeled dataset.

1. **Methodology:**

* **Data Collection**: The dataset used for this project is 'Emotion\_final.csv,' which contains text data labeled with emotion categories.
* **Data Preprocessing**:

Text was converted to lowercase, and punctuation was removed.

Stopwords were filtered out using NLTK's predefined stopword list.

Lemmatization was performed to reduce words to their base form using WordNet Lemmatizer.

* **Exploratory Data Analysis (EDA)**:

We visualized word frequency using a word cloud and plotted the distribution of emotions to understand the data.

* **Feature Extraction**:

Count Vectorizer was used to convert text data into numerical features by extracting the frequency of words.

* **Model Training**:

Three machine learning models were implemented: Multinomial Naive Bayes, Logistic Regression, and Support Vector Machine (SVM).

* **Evaluation**:

The models were evaluated using metrics like accuracy, precision, recall, and F1-score. Confusion matrices were plotted to analyze the classification performance of each model.

1. **Hardware/Software Used:**

* **Hardware**:

Processor: Intel Core i5 or higher

RAM: 8 GB or more

Storage: At least 2 GB of free space

* **Software**:

Python 3.x

Jupyter Notebook or any Python IDE

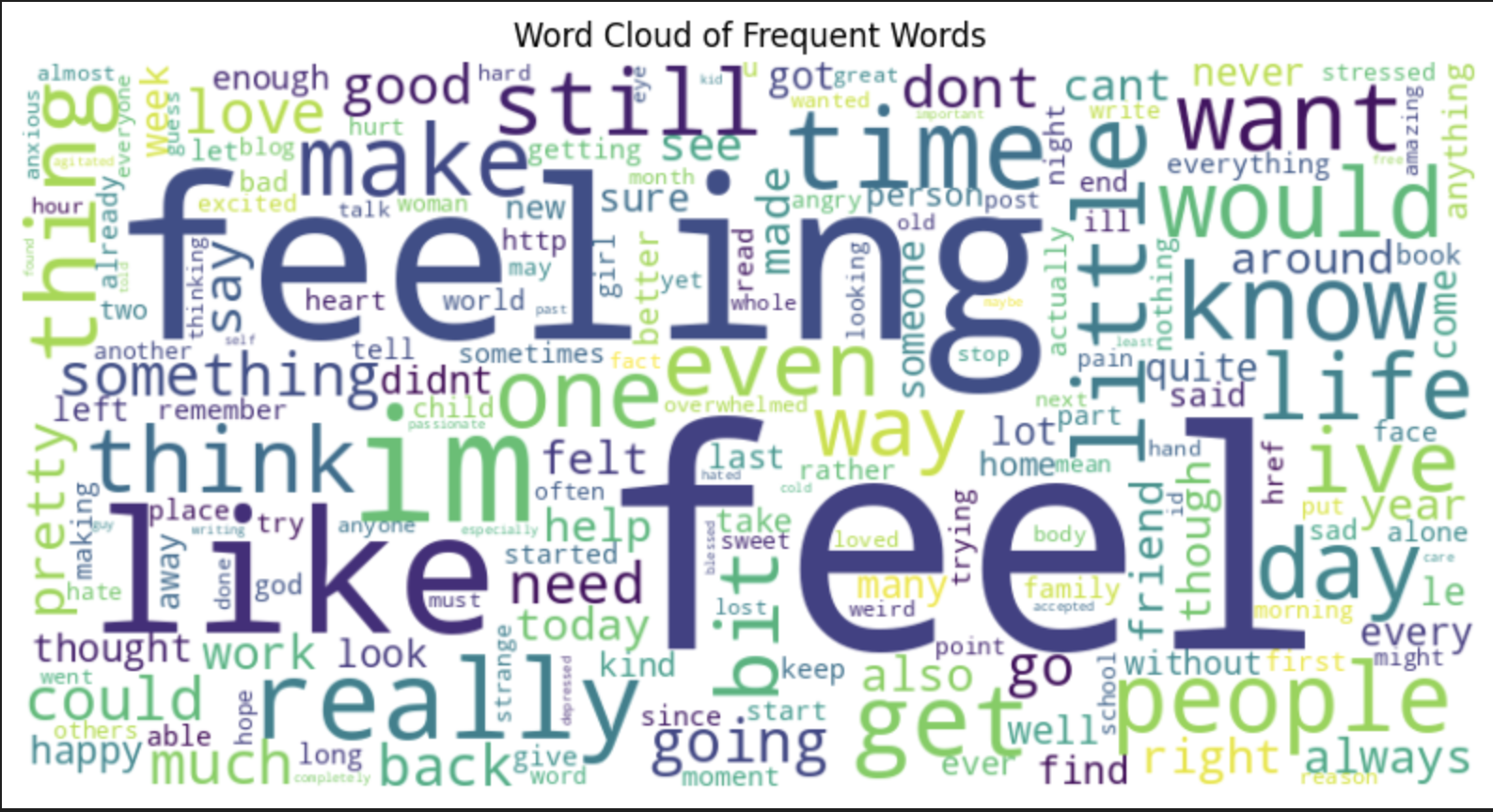
Libraries: Pandas, NLTK, Scikit-learn, Matplotlib, Seaborn, WordCloud

1. **Experimental Results:**

We evaluated the performance of three models:

* **Naive Bayes** achieved an accuracy of 0.75.
* **Logistic Regression** achieved an accuracy of 0.78.
* **Support Vector Machine** achieved an accuracy of 0.80.

The confusion matrix visualizations indicate that while the models can accurately classify emotions like 'joy' and 'sadness,' they struggle with overlapping emotions like 'fear' and 'anger.' Overall, SVM performed the best in this dataset.



A graph with purple squares

Description automatically generated

A graph of emotions

Description automatically generatedA graph with numbers and a number in blue squares

Description automatically generated

1. **Conclusions:**

This project successfully demonstrated the use of classical machine learning techniques for emotion detection in text data. The models, especially the SVM, showed promising results in classifying emotions accurately. However, there is room for improvement in handling subtle distinctions between similar emotions, such as 'fear' and 'anger.' The analysis highlights the importance of preprocessing and feature extraction in building effective sentiment analysis models.

1. **Future Scope:**

* incorporating Deep Learning Models: Future work could involve using deep learning techniques such as LSTM or Transformer models like BERT to capture more complex language patterns.
* Improving Data Quality: Expanding the dataset with more diverse text sources could enhance the model's performance.
* Real-time Sentiment Analysis: Integrating the model into a real-time application to analyze live social media data or customer feedback could be a practical extension.

1. **Github Link:**

https://github.com/yash-1105/CSET211\_Project.git