A Major Project Report On

## MULTILINGUAL SENTIMENTAL ANALYSIS ON TEXT SENTENCES

Submitted in fulfillment of the requirements for the award of the

#### Bachelor of Technology

In

#### Department of Computer Science and Engineering

##### By

|  |  |
| --- | --- |
| Ashley Edgar Dcunha | 20241A05O6 |
| Chelmela Sai Harshith | 20241A05P2 |
| Rachapally Pavan Kumar | 20241A05S1 |
| Sasidhara Kashyap | 20241A05S5 |
| Vellore Anand Kumar Sahil | 20241A05T5 |

Under the Esteemed guidance of

## Mrs. K. Anusha

**(Associate Professor)**



**Department of Computer Science and Engineering**

**GOKARAJU RANGARAJU INSTITUTE OF ENGINEERING AND TECHNOLOGY**

**(Autonomous)**

**Bachupally, Kukatpally, Hyderabad, Telangana, India, 500090**

**2023-2024**

**Ⅰ**



## GOKARAJU RANGARAJU

**INSTITUTE OF ENGINEERING AND TECHNOLOGY**

**(Autonomous)**

## CERTIFICATE

This is to certify that the major project entitled “**Multilingual** **Sentimental Analysis on Text Sentences**” is submitted by **Ashley Edgar Dcunha(20241A05O6), Chelmela Sai Harshith(20241A05P2), Rachapally Pavan Kumar(20241A05S1), Sasidhara Kashyap(20241A05S5), Vellore Anand Kumar Sahil(20241A05T5)** in partial fulfilment of the award of degree in BACHELOR OF TECHNOLOGY in Computer Science and Engineering during the academic year 2023-2024.

INTERNAL GUIDE HEAD OF THE DEPARTMENT

**Mrs. K. Anusha Mr. B. Sankara Babu**

**Associate Professor**  **Professor**

**EXTERNAL EXAMINER**

**Ⅱ**

# ACKNOWLEDGEMENT

Many people helped us directly and indirectly to complete our project successfully. We would like to take this opportunity to thank one and all. First, we wish to express our deep gratitude towards our internal guide **Mrs. K. Anusha,** Department of CSE for hr/his support in the completion of our project report. We wish to express our honest and sincere thanks to **Dr. B. Sankara Babu, HOD, Department of CSE** and to our principal **Dr. J. Praveen** for providing the facilities to complete the major project. We would like to thank all our faculty and friends for their help and constructive criticism during the project completion phase. Finally, we are very much indebted to our parents for their moral support and encouragement to achieve goals.

**Ashley Edgar Dcunha(20241A05O6)**

**Chelmela Sai Harshith(20241A05P2)**

**Rachapally Pavan Kumar(20241A05S1)**

**Sasidhara Kashyap(20241A05S5)**

**Vellore Anand Kumar**

**Sahil(20241A05T5)**

**Ⅲ**

## DECLARATION

We hereby declare that the major project entitled **“Multilingual** **Sentimental Analysis on Text Sentences”** is the work done during the period from **12th Jan 2023 to 3rd June 2023** and is submitted in the partial fulfilment of the requirements for the award of degree of Bachelor of Technology in Computer Science and Engineering from Gokaraju Rangaraju Institute of Engineering and Technology (Autonomous under Jawaharlal Nehru Technology University, Hyderabad). The results embodied in this project have not been submitted to any other university or Institution for the award of any degree or diploma.

**Ashley Edgar Dcunha(20241A05O6)**

**Chelmela Sai Harshith(20241A05P2)**

**Rachapally Pavan Kumar(20241A05S1)**

**Sasidhara Kashyap(20241A05S5)**

**Vellore Anand Kumar**

**Sahil(20241A05T5)**

**Ⅳ**

|  |  |  |
| --- | --- | --- |
|  | Table of Contents |  |
| Chapter | TITLE | Page No |
|  | Abstract | 1 |
| 1 | Introduction | 2 |
| 2 | System Requirements |  |
|  | 2.1 Purpose of the requirements document: | 4 |
|  | 2.2 Software Requirements | 5 |
|  | 2.3 Hardware Requirements | 6 |
| 3 | Literature Survey | 7 |
| 4 | Propose Approach, Modules Description, and UML Diagrams | 8 |
|  | 4.1 Modules | 9 |
|  | 4.2 UML Diagrams | 10-16 |
| 5 | Implementation, Experimental Results &Test Cases | 17-23 |
| 6 | Conclusion and Future Scope | 26 |
| 7 | References | 27 |
| 8 | Appendix   1. Full Paper-Publication Proof 2. Snapshot Of the Result 3. Optional (Like Software Installation /Dependencies/ Sudo code) | 33 |

Ⅴ

# ABSTRACT

People's sentiments are known to have a large impact on changes in stock prices, product sales, and trends. On the internet, people share their thoughts in many different languages. This is why it's important to create a way to understand how people feel in different languages when analyzing web text. Most of the text also includes emojis. So, it is important to take emojis into account when figuring out how people feel online. In our project, we are working on a method that can understand how people feel in many languages and even consider the emotions shown by emojis. This way, we can better understand what people are saying online, no matter what language they use. This will help us make better predictions and decisions in the field of sentiment analysis. Understanding the sentiments expressed by people across diverse languages and cultures is crucial in today's interconnected world. By incorporating emojis into sentiment analysis, we can capture nuances and emotions that may not be apparent through text alone. Emojis serve as a universal language, transcending linguistic barriers and providing valuable insights into the feelings and attitudes of online users.

# 1. INTRODUCTION

## Existing System

Social media analysis is critical for understanding public opinion on current events and decisions, designing and managing advertising campaigns, and planning next steps and mitigating activities for public relations initiatives. Recently, much work has been directed into the creation of data analysis algorithms capable of doing sentiment analysis on publically available text in an automated manner. The majority of existing research focuses on binary categorizing material as positive or negative without delving into the emotions that drive that categorization. The contemporary demand for in-depth analysis of available content, combined with the complexity and multidimensional nature of human emotions and beliefs, has rendered such methods obsolete. Because of these needs, research is currently focusing on identifying emotions rather than just sentiments represented in each text. This is, however, a difficult task due to not just a shortage of annotated datasets that may be utilized for emotion identification in text, but also the subjectivity injected in datasets made by manual annotations. This study presents a hybrid rule-based technique for creating a fully annotated dataset of Plutchik's eight basic emotions. The proposed algorithm takes into account the accessible emoji in the text and uses them as objective indicators of the given mood, thus addressing both highlighted difficulties.

**1.2 Limitations in Existing System**:

Textual information can be quite subtle and context-dependent. Particularly when sarcasm, irony, or humor are included, sentiment analysis and emotion recognition algorithms may find it challenging to correctly identify the intended meaning of some sentences or phrases. Contextual illiteracy can result in misunderstandings and improper sentiment/emotion categorization. Particularly social media data might be noisy and unstructured. It frequently contains grammatical mistakes, slang, misspellings, and acronyms. Noisy data can make sentiment analysis and emotion recognition less accurate by introducing noise into the process. Although the research study only covers text, social media sites also have audio, video, and image content. The analytical procedure is made more difficult by the need to extract sentiment or emotion from various multimodal data sources. The research issue of the present is to develop efficient methods to combine and analyze such data modalities. Since they include handling people's private and perhaps sensitive information, sentiment analysis and emotion recognition from social media data present privacy problems.

1. **SOFTWARE REQUIREMENT SPECIFICATIONS**

### Purpose of the requirements document:

The goal of a requirements document is to properly explain and record the system's aims, features, and limitations in the context of the proposed model for emotion recognition from text. It acts. as a comprehensive reference outlining the specific requirements and expectations for the system's development and implementation. The requirements document often includes the following:

System Goals: The document outlines the overarching goals and objectives of the emotion detection system. It clarifies the system's goals and outcomes, such as improving communication by accurately evaluating the text's emotional content.

Limitations and Constraints: The document examines any restrictions or limitations that may impact the system's performance or development. This may include budgetary constraints, schedule constraints, reliance on specific technology or software, as well as any legal or moral considerations concerning data privacy and protection.

User Requirements: The user requirements identify the system's intended users' needs and expectations. To ensure that the system meets the needs of the users and provides a pleasant user experience, this section may include user profiles, user scenarios, and user interface specifications.

Testing and Evaluation Criteria: The requirements document may describe the criteria and metrics used to measure the model's performance. It describes the performance metrics that will be used to assess how well the model can recognize emotions, such as accuracy, precision, recall, and F1-score.

### Software Requirements

* **Natural Language Processing (NLP) Libraries**: You’ll need NLP libraries or frameworks that support multilingual text processing. Some popular ones include:

**1. NLTK (Natural Language Toolkit)**: A Python library for NLP tasks.

**2. spaCy**: Another Python library for NLP with multilingual support.

**3. Transformers (Hugging Face)**: Provides pre-trained models like BERT, GPT, etc., which can be fine-tuned for sentiment analysis.

**4. FastText**: A library for word embeddings and text classification.

**Programming Languages and Libraries:**

Python: A popular choice for natural language processing tasks due to its extensive libraries like NLTK, spaCy, and TextBlob.

**Libraries for sentiment analysis:** Choose libraries or frameworks that support multilingual sentiment analysis, such as VADER, Flair, or transformers-based models (e.g., Hugging Face's Transformers).

**Text Processing Tools:**

Tokenizers: Required for breaking down sentences into individual words or tokens.

Stemmers and Lemmatizers: Useful for reducing words to their root forms.

Stopword lists: Lists of common words (e.g., "the", "and") that are often excluded from analysis as they carry little sentiment value.

**Pre-trained Language Models:**

Utilize pre-trained language models for multilingual sentiment analysis. For example, multilingual models like BERT, XLM-RoBERTa, or mBERT fine-tuned on sentiment analysis datasets.

**Data Annotation Tools:**

Tools for annotating training data with sentiment labels. This could be done manually or using semi-automated tools.

**Development Environment:**

Integrated Development Environment (IDE) like PyCharm, Jupyter Notebook, or VS Code for coding and experimentation.

Version Control System (e.g., Git) for managing code changes and collaboration.

**Deployment Tools:**

Depending on the deployment strategy (e.g., web application, API), you might need tools like Flask, Django, or FastAPI for building APIs.

### 2.3 Hardware Requirements

**Processing Power:**

CPU or GPU: Depending on the complexity of your sentiment analysis model and the scale of the data, you might need GPUs for faster training and inference.

**Memory (RAM):** Sufficient memory to load and process large language models and datasets.

**Storage:**

Disk Space: Adequate storage for storing datasets, pre-trained models, and any intermediate results.

**Networking:**

Internet Connectivity: Required for downloading pre-trained models, datasets, and updates to libraries.

**Cloud Services(optional)**:

Cloud computing platforms like AWS, Google Cloud Platform, or Microsoft Azure can provide scalable resources for training and deploying models if local resources are insufficient or if you prefer cloud-based solutions.

# LITERATURE SURVEY

Sentiment analysis, a process that involves identifying and categorizing the sentiment or opinion conveyed in text, heavily relies on emotion detection. By accurately discerning emotions in text, sentiment analysis systems can provide valuable insights into consumer feedback, social media discourse, product evaluations, and public sentiment.

The applications of sentiment analysis extend to various domains, including Customer Experience and Marketing Research, Psychological and Mental Health Analysis, Human-Computer Interaction, Social Media Analysis and Monitoring, Content Personalization, and Educational Applications.

In 2017, Amira M. Idrees et al. published "Emotion Detection in Text: A Review," which comprehensively examines machine learning-based methods for emotion detection in text. The review covers a range of strategies, including deep learning-based, supervised, and unsupervised techniques, while also discussing features, datasets, evaluation metrics, and challenges in the field.

Similarly, in 2020, S. Aarthi's publication "A Review on Emotion Detection Techniques using Textual Data" offers an extensive analysis of textual data-based emotion detection methods. It delves into deep learning methods such as CNN, RNN, and Transformer models, as well as traditional machine learning approaches like SMT and web Development Algorithms (Flask Framework which includes Routing, Templating and NLP Additionally, the study explores various feature extraction techniques and relevant datasets.

In 2019, K. Kalimuthu's "Emotion Detection from Text: A Survey" provides a comprehensive examination of machine learning-based emotion recognition from text. It covers topics such as feature extraction techniques, classification algorithms, assessment measures, and challenges faced in the field, along with its applications and future directions.

Emotion detection from text using machine learning has been extensively researched, yielding numerous publications. Challenges and future research directions include addressing sarcasm, irony, and implicit emotions, managing noisy or imbalanced datasets, handling cultural and contextual variations, enhancing model interpretability and explainability, exploring multi-modal techniques combining text with visuals and audio, and developing real-time emotion recognition systems for dynamic text streams.

# Proposed Approach, Modules Description, and UML Diagrams

## Proposed System

In the proposed system for multilingual sentiment analysis in Hindi and English, we envision a platform that can accurately analyze sentiments expressed in both languages. This system would employ advanced natural language processing (NLP) techniques tailored to the linguistic nuances of Hindi and English. It would leverage large datasets of annotated texts in both languages to train machine learning models capable of understanding and categorizing sentiments expressed in diverse contexts. Additionally, the system would consider the use of emojis, which play a significant role in expressing emotions in online communication. By integrating these capabilities, the proposed system aims to provide businesses and researchers with a powerful tool for gaining insights into the sentiments of users across different linguistic backgrounds, thereby enabling more effective communication and decision-making.

### Advantages Over Existing System:

Emotion recognition from text enhances communication by providing a better grasp of a message's emotional context. The system classifies text into emotions using emojis, resulting in an understandable and accessible visual representation. Natural Language Processing (NLP) is a field of study in artificial intelligence and linguistics that can be used to develop reliable models that can learn from annotated datasets. As they are exposed to additional data, these models can adapt and improve their performance, boosting the accuracy of emotion recognition. The suggested method acknowledges the challenges given by noisy textual data, such as typos, grammatical errors, emoticons, sarcasm, and irony.To reduce the detrimental impact of noise on emotion detection accuracy, the system uses data preparation techniques such as noise reduction, stop word deletion, and punctuation removal.

Enhancing communication involves leveraging emotion recognition from text to grasp the emotional undertones of messages more effectively. Utilizing emojis for categorizing text emotions creates a clear and easily accessible visual depiction. Machine learning algorithms like those found in the Flask Framework, with its routing, templating, and NLP capabilities, alongside SMT, provide a dependable foundation for models to learn from annotated datasets. With exposure to more data, these models can adapt and refine their performance, thereby boosting emotion recognition accuracy. Acknowledging the hurdles posed by noisy textual data—including typos, grammatical errors, emoticons, sarcasm, and irony—the proposed approach implements data preprocessing techniques such as noise reduction, stop word elimination, and punctuation removal to counteract these challenges. Emojis offer a universally comprehensible means of conveying emotions, transcending cultural differences and enhancing user engagement by facilitating emotional interaction. This visual representation is globally applicable, as emojis are widely recognized and understood across various cultural contexts, thereby enriching the user experience and fostering emotional exchange.

## Modules

**Data Collection and Preprocessing**: This stage entails acquiring a large number of text samples in Hindi and English from a variety of sources, including social media, news articles, and product evaluations. The obtained data is then preprocessed to clean and standardize the text. Tokenization, stemming, and stop word removal are among the tasks assigned.

**Language Identification:** A language identification module is required to successfully handle Hindi and English texts. This component determines the language of each incoming text and routes it to the appropriate analytical pipeline for further processing.

**Sentiment Analysis Model Training:** This program would involve training sentiment analysis machine learning models in Hindi and English. To train accurate models, a sizable labeled dataset for every language would be needed. For this, methods like transformers and recurrent neural networks (RNNs), which are deep learning architectures, could be employed.

**Emoji Analysis**: In online communication in particular, emojis are essential for conveying emotion. The task of locating, deciphering, and integrating emojis into the sentiment analysis process would fall under the purview of an emoji analysis module.

**Integration and Deployment:** Following their development, the constituent modules must be combined to form a coherent system. In order to ensure that the system can handle real-time text analysis in both Hindi and English, this module would handle the deployment of the system and the integration of its many components.

**Evaluation and Improvement:** An ongoing assessment of the system's functionality is essential to its advancement. This module would include tracking the accuracy of the system, pinpointing areas in need of development, and revising the models and algorithms in light of fresh information and user input.

## UML Diagrams

## Use-Case Diagram

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

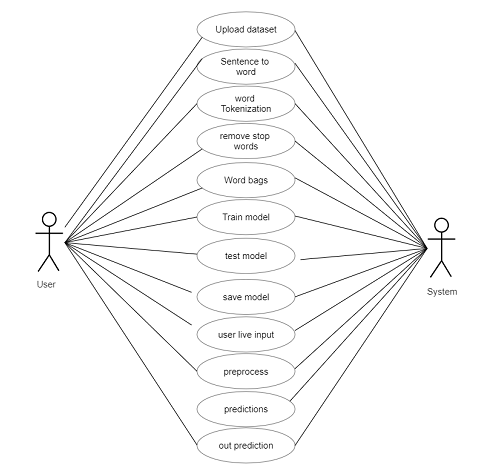


Fig: 4.3.1: Use Case Diagram

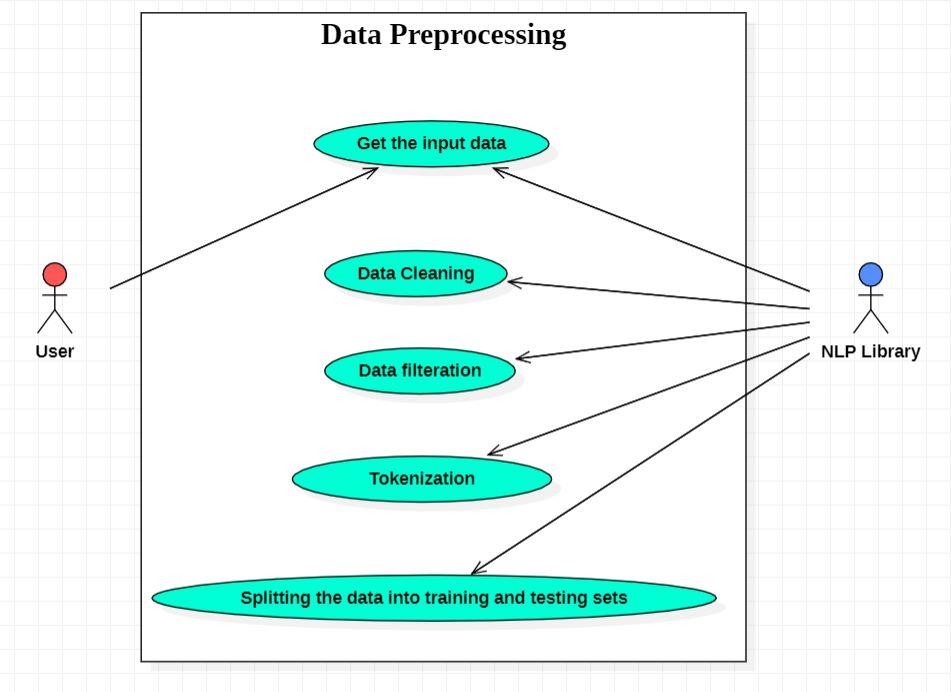


Fig: 4.3.2: Data Preprocessing Use Case Diagram

#### Data Preprocessing: -

* Preprocessing is an essential stage that includes a number of methods for cleaning, transforming, and normalizing text data into a format that is appropriate for analysis.
* The first step is to make the text cleaner by eliminating any extraneous characters, such as stop words, special characters, and punctuation. Many Python libraries,including TextBlob, spaCy, and NLTK, can be used for this.

• Following, the content will be changed over to lowercase and all numbers and other non-textual components expelled to normalize it.

• After the content has been cleaned and normalized, it is time to tokenize it into person words. To empower the machine learning framework to prepare the content, it must be tokenized, or partitioned into littler components called tokens.

• The another step is to apply stemming or lemmatization to the tokens. Whereas stemming includes diminishing each word to its root frame, lemmatization includes lessening each word to its base shape. This makes it less demanding for the machine learning calculation to prepare the content and brings down the one of a kind word number of the dataset.

• The preprocessed information is at that point partitioned into testing and preparing sets. The machine learning calculation is prepared on the preparing set, and its execution is surveyed on the testing set.

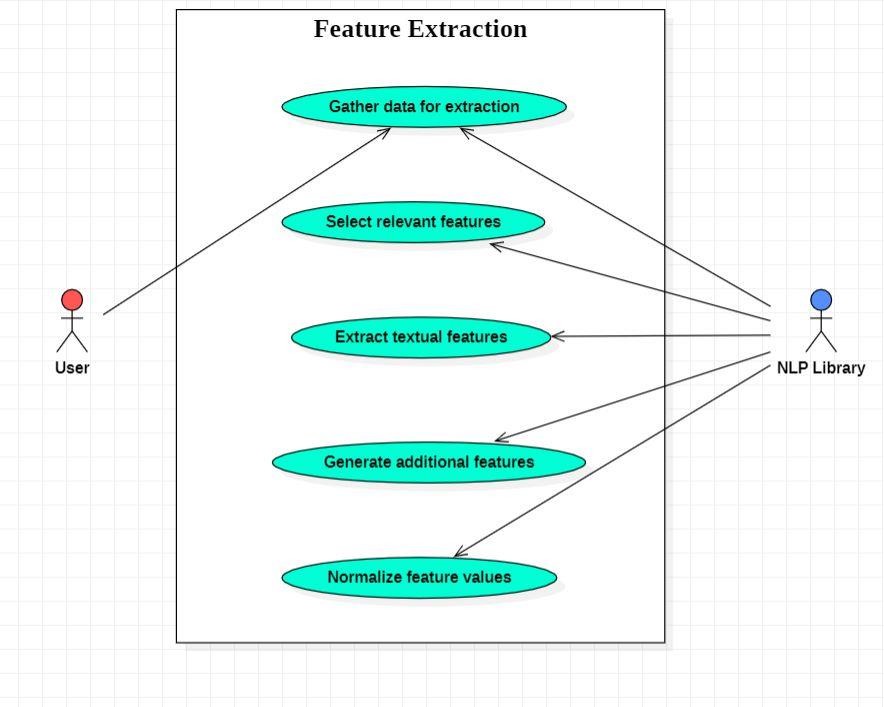


Fig: 4.3.3: Feature Extraction Use Case Diagram

#### Feature Extraction: -

* The first step involves gathering and organizing text data that will be used for feature extraction.
* The next step involves identifying the most important features also called as keywords that are relevant to the emotion detection task.
* Next, we use techniques such as bag-of-words, word embeddings, or other natural language processing (NLP) techniques to extract features from the text data.
* The next step involves creating new features from the existing text data or extracted features that can be used to improve the performance of the emotion detection model.
* The next step involves scaling and normalizing the feature values to ensure that they are in the same range and have the same distribution, which can help the machine learning algorithm to learn better.

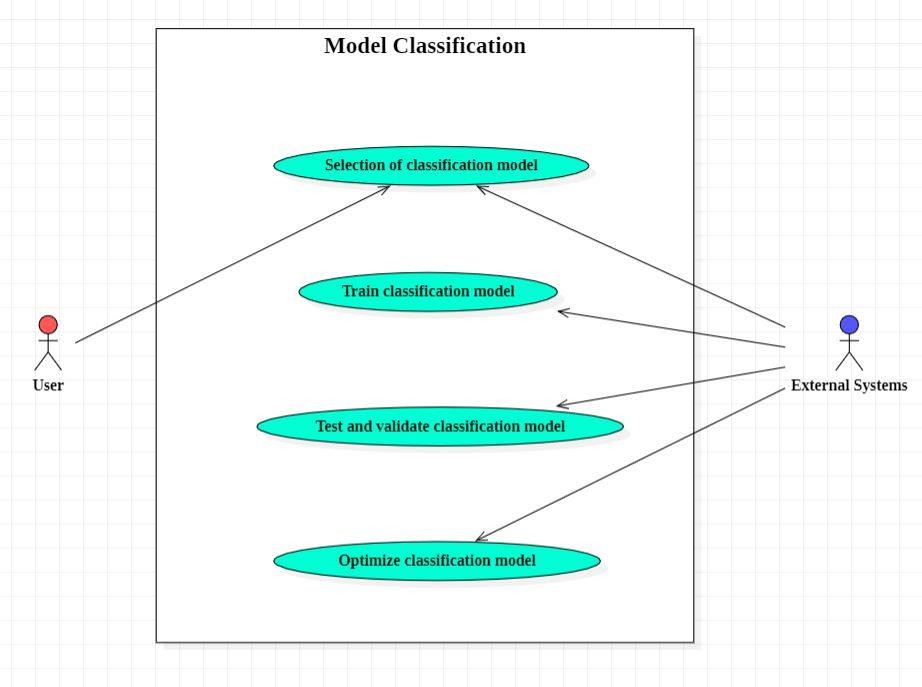


Fig: 4.3.4: Model Classification Use Case Diagram

#### Model Classification: -

* The first step involves choosing a classification model that is selecting an appropriate machine learning algorithm for emotion detection tasks, such as Web Development Algorithms (Flask Framework), statistical machine translation (SMT), and neural machine translation (NMT).,etc.
* The next step involves training the classification model which is the machine learning algorithm on the preprocessed and feature-extracted text data to learn to classify the text into different emotional categories.
* The next step involves testing the performance of the trained classification model on a separate test dataset and validating the accuracy, precision, recall, and F1-score of the classification model.

The last step involves improving the performance of the classification model by adjusting hyperparameters, selecting optimal features, or using ensemble methods

**ACTIVITY DIAGRAM**

Activity diagrams are graphical representations of work processes that include step-by-step tasks and actions with options for choice, iteration, and concurrent execution. Activity diagrams in the Unified Modeling Language can be used to depict the business and operational work flows of system components. An activity diagram depicts the overall flow of control.

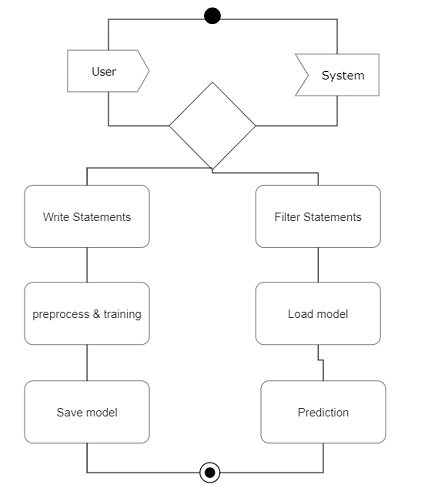


Fig: 4.3.5: Activity Diagram

**SEQUENCE DIAGRAM**

A sequence diagram in UML (Unified Modeling Language) is a type of interaction diagram that illustrates the interactions between objects or components in a system over time. It depicts the flow of messages between these objects, representing how they collaborate to achieve a particular functionality or behavior.

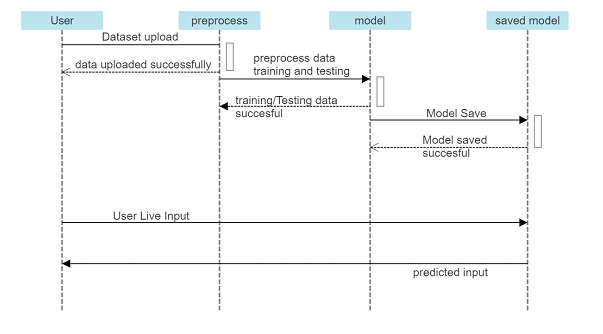


Fig: 4.3.6: Sequence Diagram

**CLASS DIAGRAM**

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.

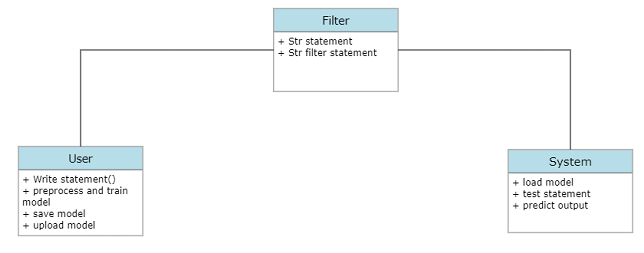
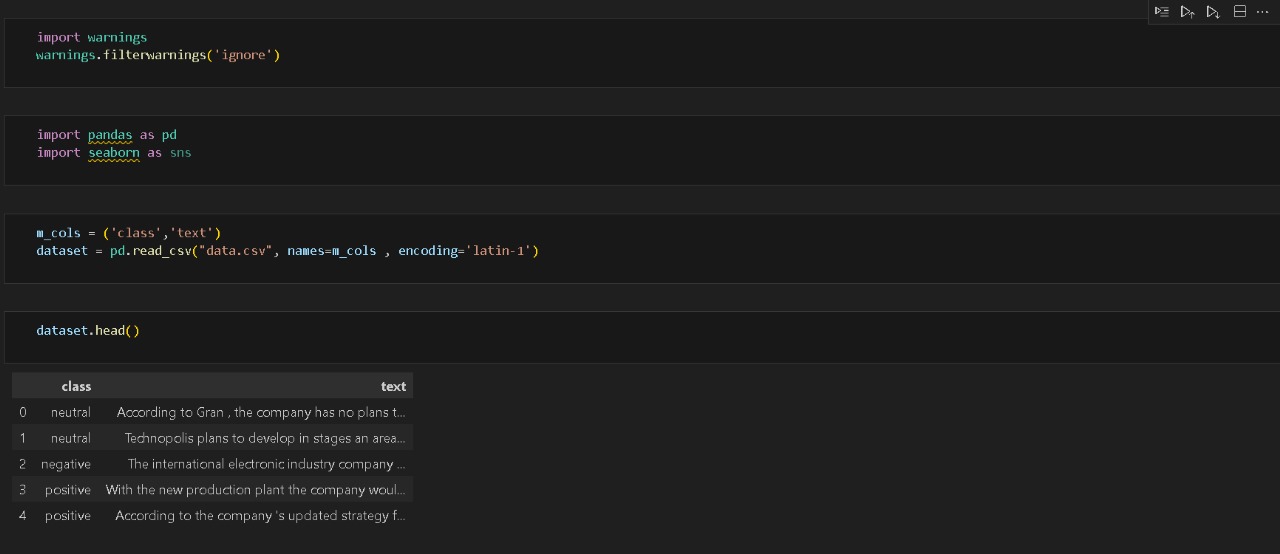
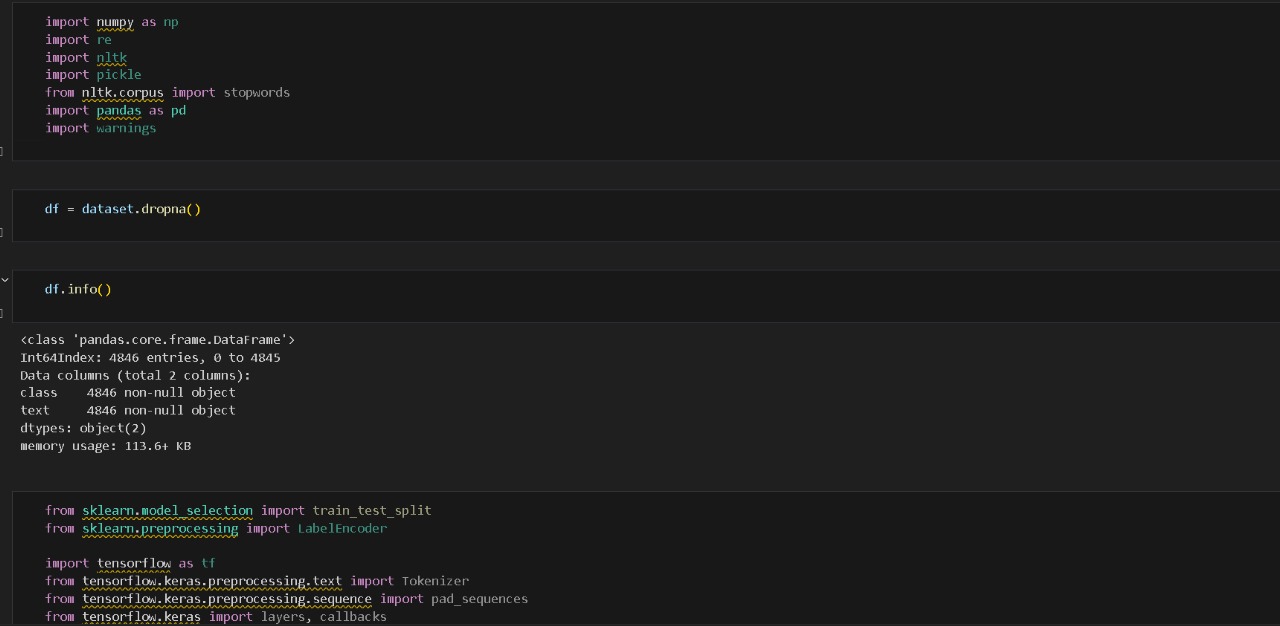
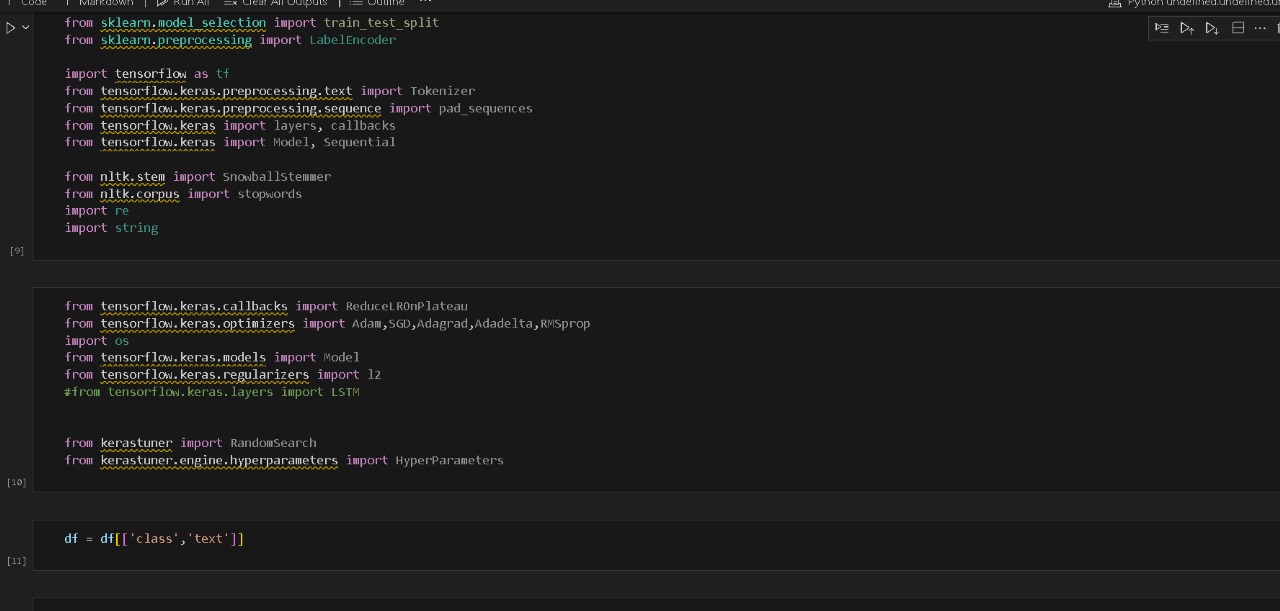


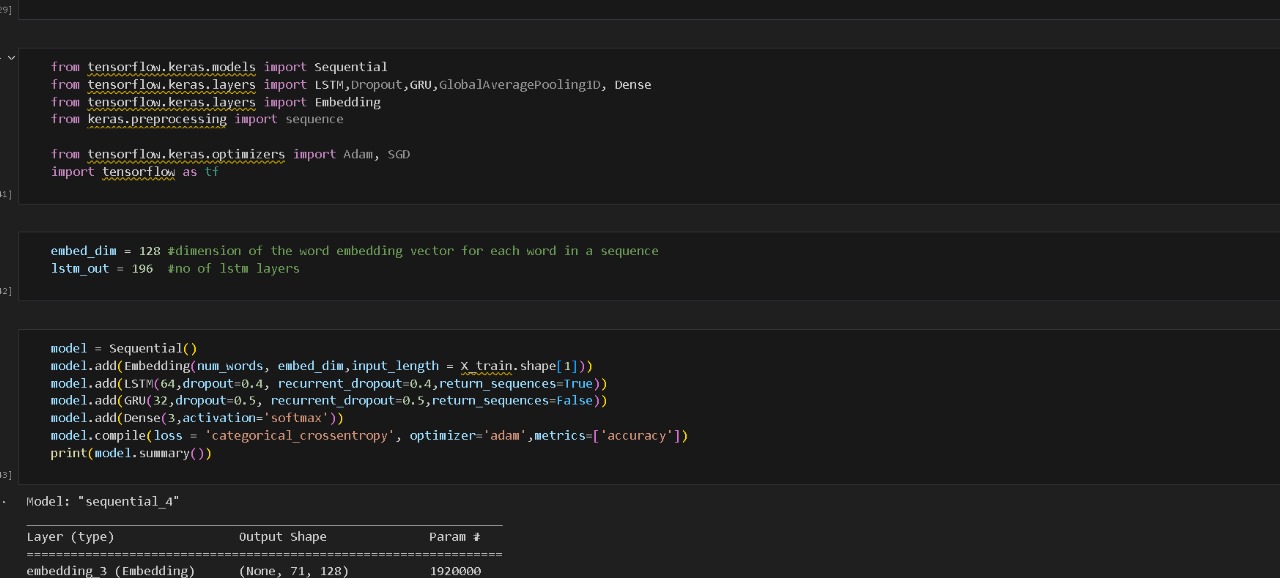
Fig: 4.3.7: Class Diagram

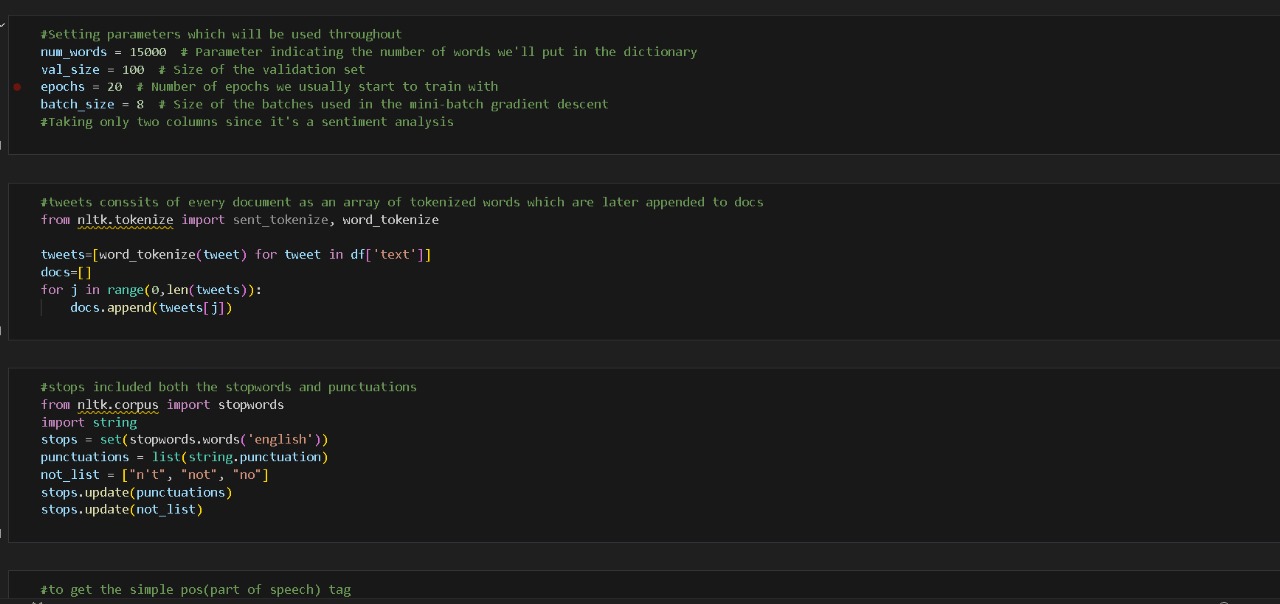
## Implementation

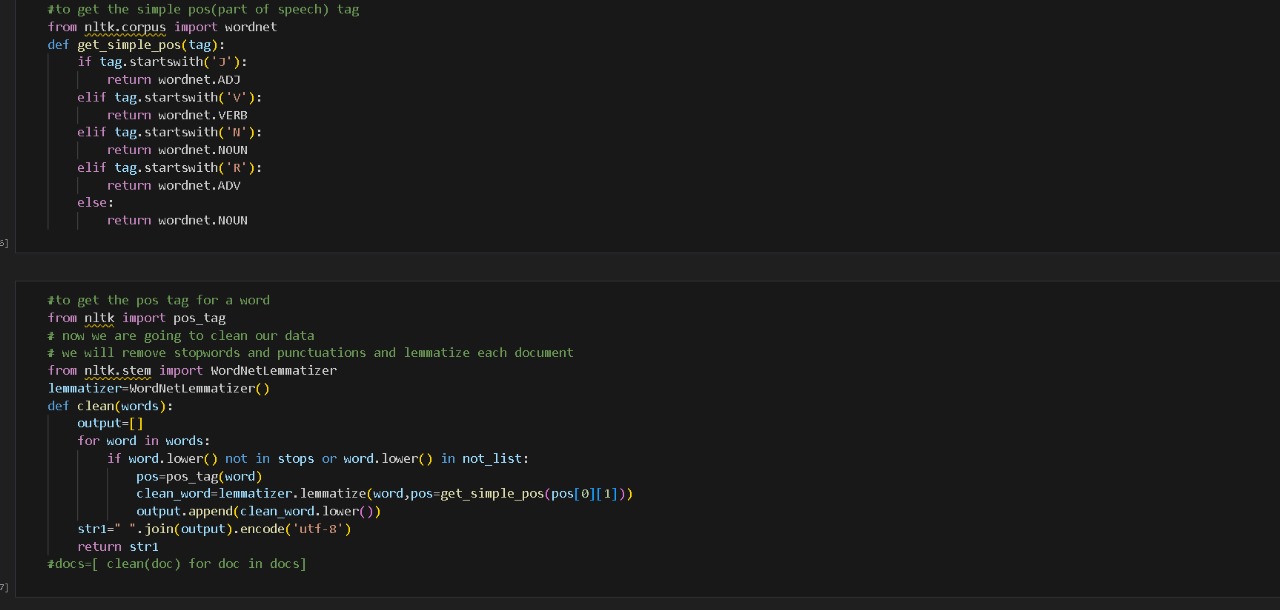
****

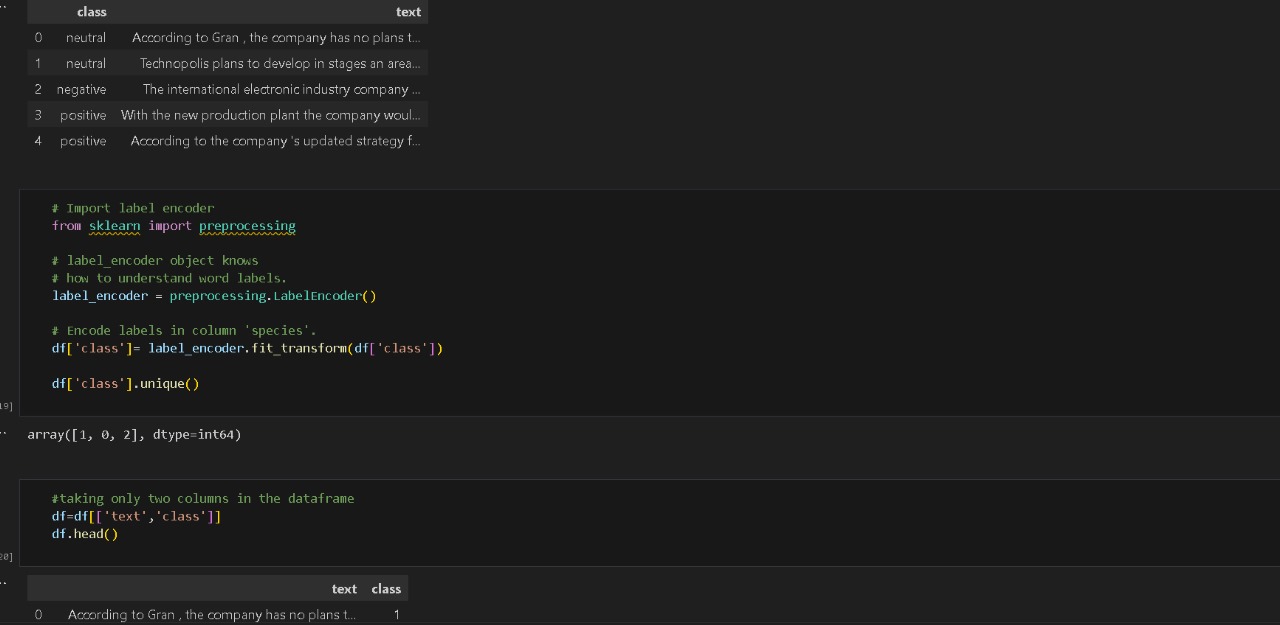


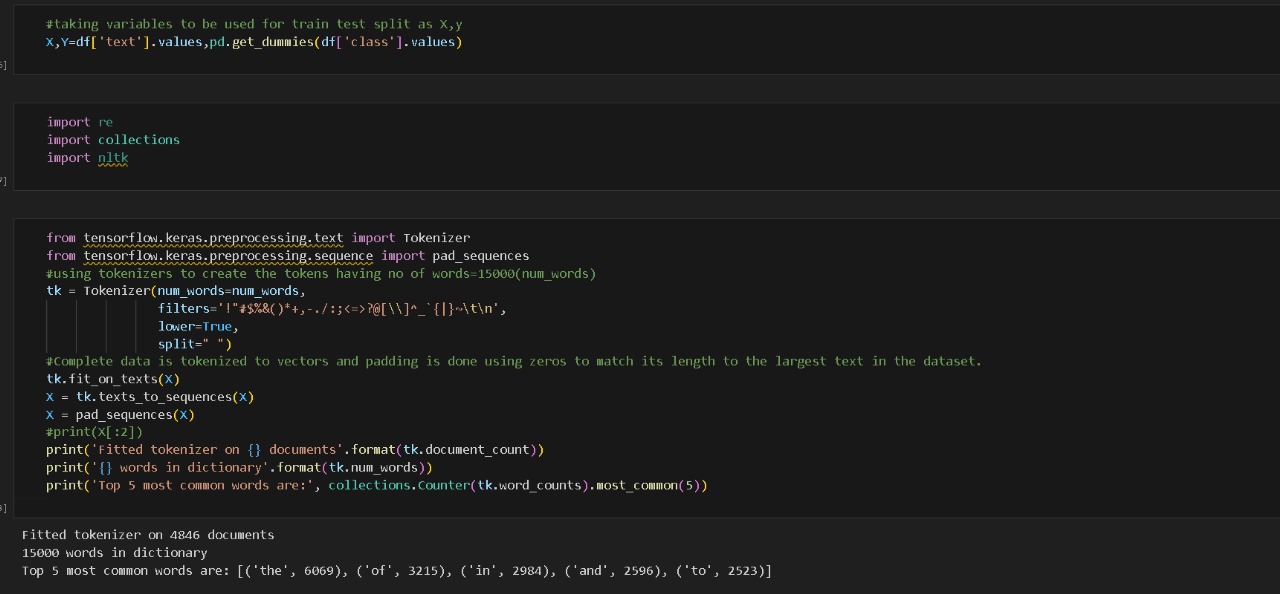


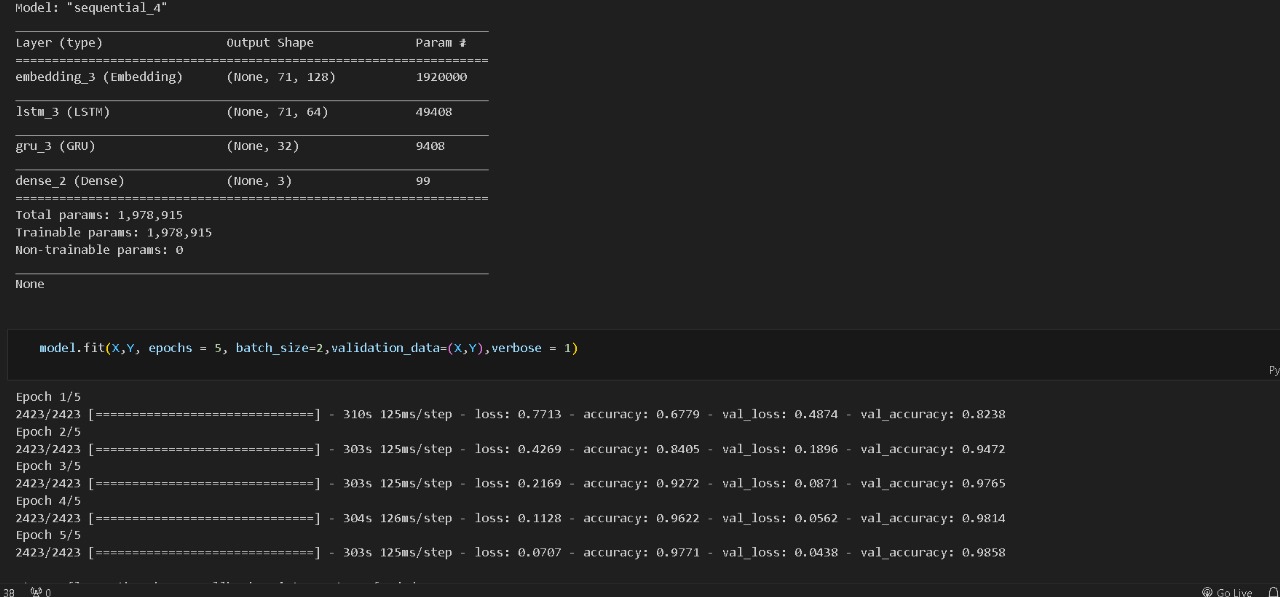
****



****

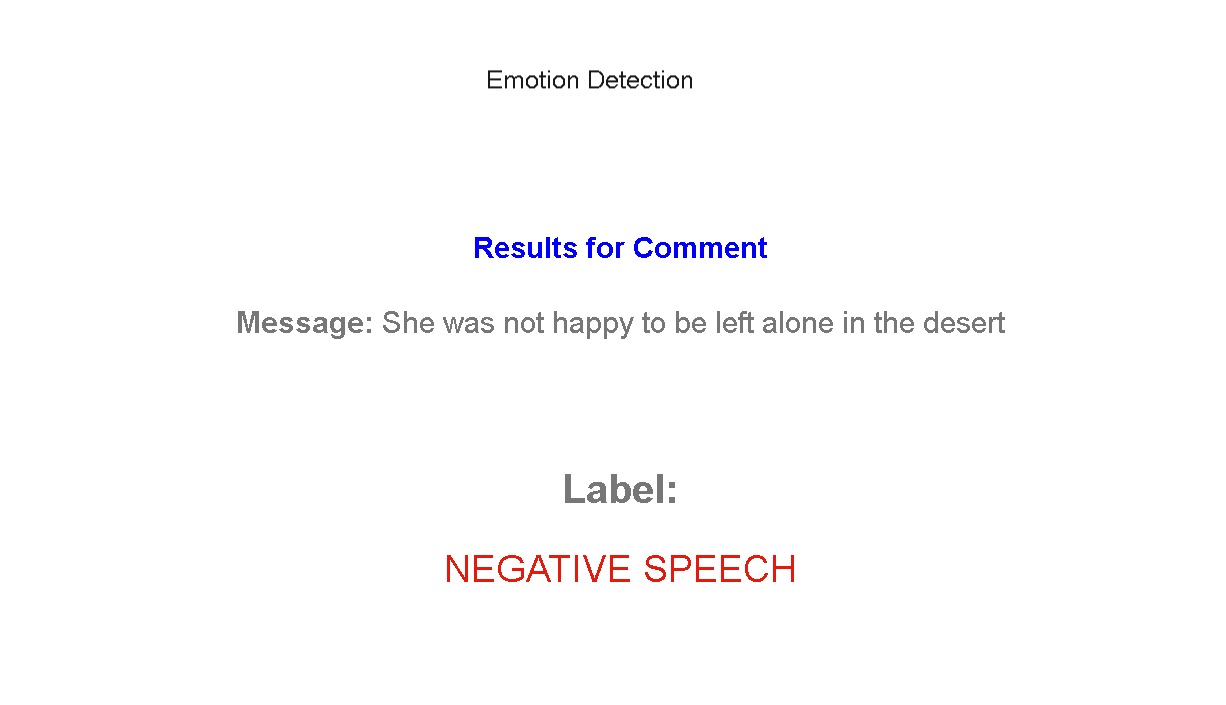
****

****

****

## Results

Fig: 5.2.1: Output showing Message as NEGATIVE.

****

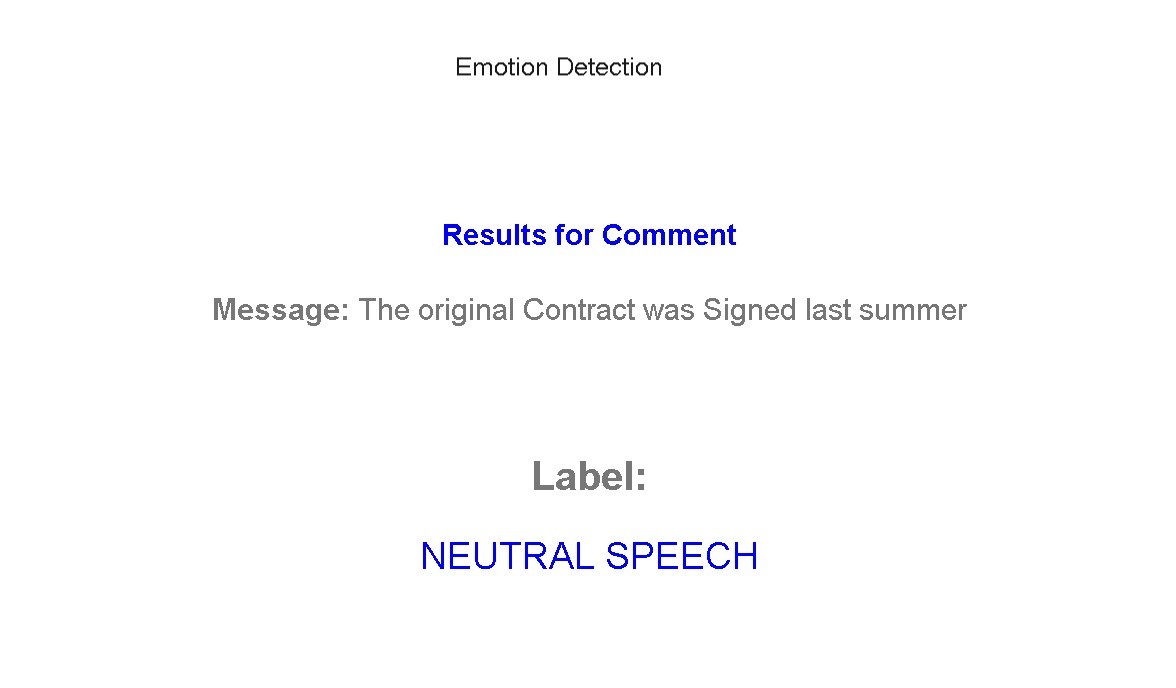


Fig: 5.2.2: Output showing Message as NEUTRAL.

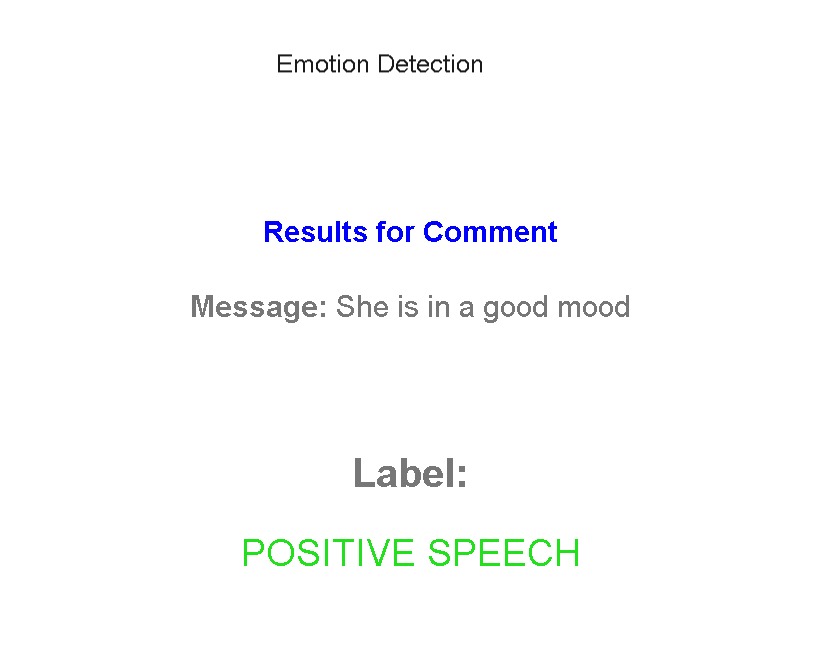


Fig: 5.2.3: Output showing Message as POSITIVE.

## Test Cases

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case-**  **ID** | **Test Scenario** | **Test Case** | **Pre- conditions** | **Test steps** | **Test Data** | **Expected Results** | **Post- conditions** | **Actual Result**  **s** | **Status Pass/**  **Fail** |
| 1 | Data Cleaning | Punctuation | Raw Textual | 1. obtain the | “Hello, |  | Cleaned |  | Pass |
|  |  | Removal | data is | raw textual | how are | data without |  |
|  |  |  | available | data | you?” | punctuation |  |
|  |  |  |  | 2. Preprocess |  |  |  |
|  |  |  |  | the data by |  |  |  |
|  |  |  |  | removing |  |  |  |
|  |  |  |  | punctuation |  |  |  |
| 2 | Data Cleaning | Stopword | Raw textual | 1. obtain the | “I am |  | Cleaned |  | Pass |
|  |  | Removal | data is | raw textual | going to | data without |  |
|  |  |  | available | data | store” | stopwords |  |
|  |  |  |  | 2. Preprocess |  |  |  |
|  |  |  |  | the data by |  |  |  |
|  |  |  |  | removing |  |  |  |
|  |  |  |  | stopwords |  |  |  |
| 3 | Data Cleaning | Noise | Raw Textual | 1. obtain the | “The |  | Cleaned |  | Pass |
|  |  | Removal | data is | raw textual | product is | data without |  |
|  |  |  | available | data | amazing” | noise |  |
|  |  |  |  | 2. Prepocess |  |  |  |
|  |  |  |  | the data by |  |  |  |
|  |  |  |  | removing |  |  |  |
|  |  |  |  | noise |  |  |  |
| 4 | Feature | Emotion | Preprocessed | 1. Extract | “I’m so | Happy | Extracted |  | Pass |
|  | Extraction | Handling | data is | features from | happy” |  | feature: |  |
|  |  |  | available | the |  |  | happy |  |
|  |  |  |  | preprocessed |  |  |  |  |
|  |  |  |  | data |  |  |  |  |
|  |  |  |  | 2. Identify and |  |  |  |  |
|  |  |  |  | handle |  |  |  |  |
|  |  |  |  | emotions |  |  |  |  |
| 5 | Feature | N-grams | Preprocessed | 1.Extract | “The |  | Extracted n- |  | Pass |
|  | Extraction | Extraction | data is | features from | movie is | grams |  |
|  |  |  | available | the | very | features |  |
|  |  |  |  | preprocessed | interesting |  |  |
|  |  |  |  | data | ” |  |  |
|  |  |  |  | 2.Identify and |  |  |  |
|  |  |  |  | handle |  |  |  |
|  |  |  |  | emotions |  |  |  |
| 6 | Model | Training | Preprocessed | Train the | Preprocess | Trained | Model |  | Pass |
|  | Classification |  | data and | model using | ed data | model | Trained |  |
|  |  |  | labelled | the | Labelled |  | successfully |  |
|  |  |  | dataset are | preprocessed | Dataset |  |  |  |
|  |  |  | available | data and |  |  |  |  |
|  |  |  |  | labelled |  |  |  |  |
|  |  |  |  | dataset |  |  |  |  |
| 7 | Model | Prediction | Trained model | Use the | Trained | Predicted | Correct | Happy | Pass |
|  | Classification |  | and new | trained model | model | emotional | emotional |  |  |
|  |  |  | textual data | to predict the | new | label | label |  |  |
|  |  |  | are available | emotional | textual |  | predicted |  |  |
|  |  |  |  | label for the | data |  |  |  |  |
|  |  |  |  | new textual |  |  |  |  |  |
|  |  |  |  | data |  |  |  |  |  |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 8 | Model | Accuracy | Labeled test | Compare the | Labeled | Accuracy | Calculated | 90% | Pass |
|  | evaluation | Calculatio | dataset and | predicted | test dataset | Score | accuracy |  |  |
|  |  | n | predicted | labels with the | predicted |  | score |  |  |
|  |  |  | labels are | actual labels in | labels |  |  |  |  |
|  |  |  | available | the test dataset |  |  |  |  |  |
| 9 | Model | Performan | Trained model | Evaluate the | Trained | Performan | Calculated | Precisio | Pass |
|  | Evaluation | ce | and test | model’s | model | ce metrics | performance | n: 0.85, |  |
|  |  | Evaluation | dataset are | performance | Test |  | metrics | Recall: |  |
|  |  |  | available | using the test | dataset |  |  | 0.92, |  |
|  |  |  |  | dataset |  |  |  | F1- |  |
|  |  |  |  |  |  |  |  | Score: |  |
|  |  |  |  |  |  |  |  | 0.88 |  |
| 10 | Representation | Emoji | Emotional | Map | Emotional | Mapped | Emojis |  | Pass |
|  | validation | Mapping | labels and | emotional | labels | emojis | mapped |  |
|  |  |  | corresponding | labels to |  |  | correctly |  |
|  |  |  | emojis are | emojis based |  |  |  |  |
|  |  |  | available | on predefined |  |  |  |  |
|  |  |  |  | mappings |  |  |  |  |

Table: 5.3: Test Cases

Input: "I am feeling bahut khush today. 😄"

Expected Output: Positive sentiment (indicating that the user is feeling very happy), with the 😄 emoji reinforcing the positive sentiment.

Input: "Mujhe yeh pasand hai, but I don't like the ending. 😕"

Expected Output: Mixed sentiment with a positive sentiment for the Hindi part indicating that the user likes something, and negative sentiment for the English part indicating that the user doesn't like something, further emphasized by the 😕 emoji indicating slight disappointment.

Input: "Today maine bahut kaam kiya, and now I'm exhausted. 😫"

Expected Output: Mixed sentiment with potentially positive sentiment for the Hindi part indicating that the user did a lot of work, and negative sentiment for the English part indicating that the user is exhausted, reinforced by the 😫 emoji indicating tiredness or exhaustion.

# CONCLUSION AND FUTURE SCOPE

We have implemented the project on emotion detection of text using machine learning algorithms. We created our dataset merged it with an already existing dataset and trained the data using machine learning algorithms like Web Development Algorithms (Flask Framework), statistical machine translation (SMT), and neural machine translation (NMT). After finishing the code, we imported it into the stream and used it as a user interface to display the inputs and outputs The project has achieved the goal of identifying emotions expressed in textual data. The implemented model has demonstrated the ability to classify text into different emotional categories, such as happiness, sadness, anger, fear, etc., with a reasonable level of accuracy. This has significant applications in various fields, including social media sentiment analysis, customer feedback analysis, and content recommendation.

While the project has achieved its primary objectives, we can improve the project by expanding the dataset that is the performance can be enhanced by increasing the size and diversity of the dataset. Experimenting with different machine learning algorithms, such as deep learning and fine-tuning hyperparameters can enhance the model’s performance. Extending the project to handle text in multiple languages would broaden its impact and explore techniques to perform emotion detection in real time, such as analyzing streaming data.

With these, the project can be further enhanced to provide more accurate and robust emotion detection capabilities, enabling its application in a wider range of real-world scenarios.

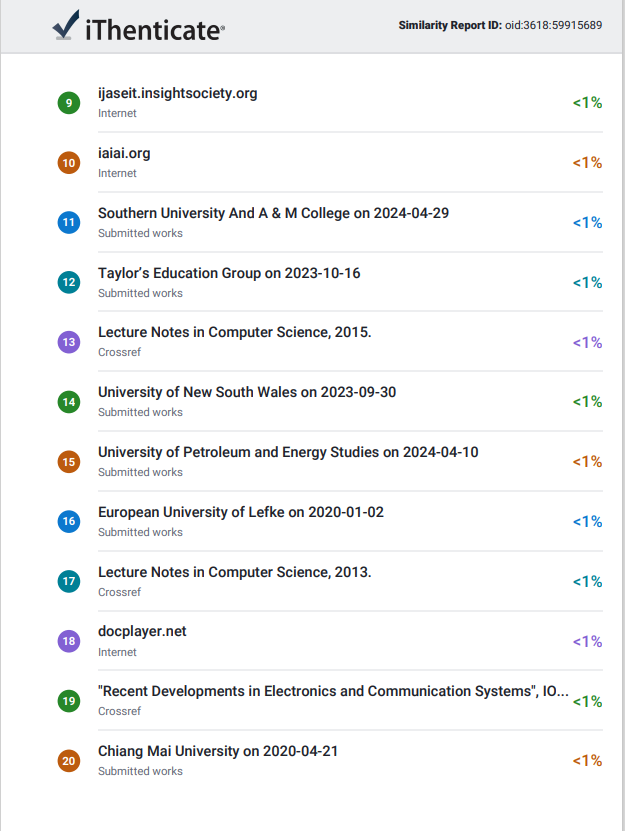
# REFERENCES

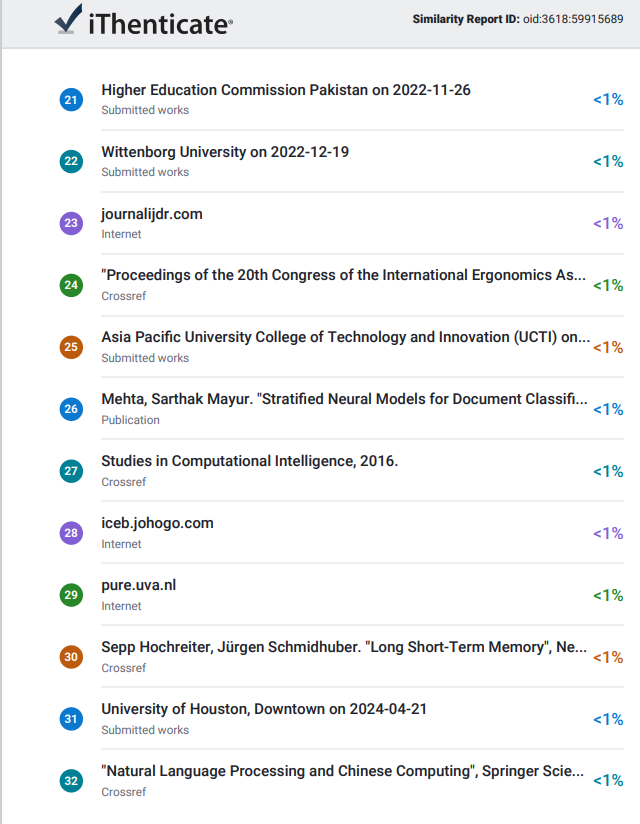
1. Cambria, E., Poria, S., & Gelbukh, A. (2014). Emotion Analysis Using SenticNet. In Sentiment Analysis and Ontology Engineering (pp. 61-85). Springer.
2. Mohammad, S. M., Kiritchenko, S., & Zhu, X. (2013). NRC-Canada: Building the State-of-the-Art in Sentiment Analysis of Tweets. Proceedings of the 7th International Workshop on Semantic Evaluation (SemEval 2013), 321-327.
3. Zhang, X., & LeCun, Y. (2015). Text Understanding from Scratch. arXiv preprint arXiv:1502.01710.
4. Bravo-Marquez, F., Mendoza, M., Poblete, B., & Baeza-Yates, R. (2017). Large-scale comparison of sentiment classification methods using linguistic and gated recurrent neural networks. Information Retrieval Journal, 20(5-6), 559-580.
5. Poria, S., Cambria, E., Hazarika, D., Majumder, N., Zadeh, A., & Morency, L. P. (2017). Context- dependent sentiment analysis in user-generated videos. In Proceedings of the 2017 ACM on Multimedia Conference (pp. 709-717). ACM.
6. Busso, C., Bulut, M., Lee, C. C., Kazemzadeh, A., Mower, E., Kim, S., ... & Narayanan, S. (2008). IEMOCAP: Interactive emotional dyadic motion capture database. Language Resources and Evaluation, 42(4), 335-359.
7. Liu, F., & Lane, I. (2018). Emotion detection and recognition: A survey. IEEE Transactions on Affective Computing, 9(3), 345-369.
8. Agarwal, A., Xie, B., Vovsha, I., Rambow, O., & Passonneau, R. (2011). Sentiment analysis of Twitter data. In Proceedings of the Workshop on Languages in Social Media (pp. 30-38). Association for Computational Linguistics.
9. Mohammad, S., & Turney, P. (2013). Crowdsourcing a word–emotion association lexicon. Computational Intelligence, 29(3), 436-465.
10. Poria, S., Cambria, E., & Hussain, A. (2017). Convolutional neural networks for multimodal sentiment analysis. In Multimodal Sentiment Analysis (pp. 39-59). Springer.

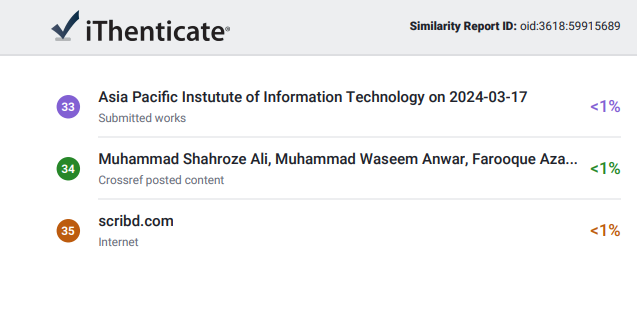
|  |  |  |
| --- | --- | --- |
|  | **LIST OF FIGURES** |  |
| **Fig No** | **Fig Title** | **Page No** |
| 4.3.1 | Use Case Diagram | 10 |
| 4.3.2 | Data Preprocessing Use Case Diagram | 11 |
| 4.3.3 | Feature Extraction Use Case Diagram | 12 |
| 4.3.4 | Model Classification Use Case Diagram | 13 |
| 4.3.5 | Activity Diagram | 14 |
| 4.3.6 | Sequence Diagram | 15 |
| 4.3.7 | Class Diagram | 16 |
| 5.2.1 | Output showing Message as NEGATIVE | 22 |
| 5.2.2 | Output showing Message as NEUTRAL | 23 |
| 5.2.3 | Output showing Message as POSITIVE. | 23 |
| 5.3 | Test Cases | 25 |

|  |  |  |
| --- | --- | --- |
|  | **LIST OF TABLES** |  |
| **Table No** | **Table Name** | **Page No** |
| 5.3 | Test Cases | 24 |

# 







**8.Appendix**

Paper Publication Proof

