**MULTIMODAL AND MULTILINGUAL BASED CLASSIFICATION OF SOCIAL MEDIA DATA**

**PROJECT SYNOPSIS**

OF MINOR PROJECT

# **BACHELOR OF TECHNOLOGY**

COMPUTER SCIENCE AND ENGINEERING

SUBMITTED BY

# AAYUSH KALIA ABHIJOT SINGH CHAITANYA ARORA

2215003 2215004 2215027

2203390

2203414

2203391

UNDER THE GUIDANCE OF

ER. MEETALI



**GURU NANAK DEV ENGINEERING COLLEGE,**

**LUDHIANA**

**ABSTRACT**

With the increasing use of social media, vast amounts of user-generated content are shared across multiple languages and formats, including text, images, and audio. However, the spread of hate speech, misinformation, and toxic content poses significant challenges, making traditional moderation methods inefficient due to their reliance on rule-based approaches and manual review. Multimodal and multilingual based classification of social media data is an AI-driven content classification and moderation system that leverages machine learning (ML) and natural language processing (NLP) techniques to analyze multimodal content. It integrates Optical Character Recognition (OCR) to extract text from images, Speech-to-Text conversion for spoken content analysis, and transformer-based models like mBERT and XLM-R for multilingual text classification. Additionally, sentiment analysis is employed to detect emotions such as joy, anger, sadness, and sarcasm while identifying hate speech, offensive language, and fake news. To enhance accuracy and efficiency, the project combines traditional ML models (SVM, Logistic Regression, Random Forest) with deep learning models (BERT/RoBERTa), ensuring robust classification across diverse content types. Designed for scalability, the system can be integrated into social media platforms and corporate moderation systems for real-time content analysis and automated moderation. By reducing reliance on manual review and improving accuracy, this project aims to foster a safer and more inclusive digital ecosystem. Future advancements may include real-time processing, adaptive learning models to address evolving online trends, and expanded multilingual support for low-resource languages, making it a powerful tool for moderating harmful content across global online platforms.

**ACKNOWLEDGEMENT**

We extend our heartfelt thanks to Dr. Kiran Jyoti, Head of the Computer Science and Engineering Department, GNDEC, Ludhiana, for her constant guidance and valuable insights. Her unwavering encouragement has been a great source of motivation throughout the project.

We are deeply grateful to Er. Meetali for her wise counsel and invaluable guidance. Her expertise and support have played a pivotal role in shaping this project, and without her assistance, its successful completion would not have been possible.

We also extend our appreciation to all the faculty members of the Computer Science and Engineering Department, GNDEC, for their intellectual support and encouragement during the course of this work. Their constructive feedback and assistance have been highly beneficial in refining our project.

Lastly, we express our gratitude to everyone who contributed, directly or indirectly, to the successful completion of this project. Their support and cooperation have been invaluable in bringing this work to fruition.

**TABLE OF CONTENT**

|  |  |  |
| --- | --- | --- |
| **SR.NO.** | **CONTENT** | **PAGES** |
| 1. | LIST OF FIGURES | 5 |
| **2.** | |  | | --- | | **INTRODUCTION** | | 6 |
| 3. | |  |  |  | | --- | --- | --- | |  | **SYSTEM REQUIREMENTS** |  | | 7 |
| 4. | SOFTWARE REQUIREMENT ANALYSIS | 8-9 |
| 5. | |  |  |  | | --- | --- | --- | |  | **SOFTWARE DESIGN** |  | | 10-13 |
| 6. | |  |  |  | | --- | --- | --- | |  | **TESTING MODULE** |  | | 14-15 |
| 7. | **PERFORMANCE OF THE PROJECT** | 16 |
| 8. | **OUTPUT SCREENS** | 17-20 |
| 9. | **REFERENCES** | 21 |

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| SR. NO. | NAME OF FIGURES | PAGES |
| 1 | Data Flow Diagram | 11 |
| 2 | Agile Model | 13 |
| 3 | Display of some data in row | 17 |
| 4 | Training and Testing Split Data | 17 |
| 5 | Sentiment Labels | 18 |
| 6 | Count of sentiments in the dataset | 18 |
| 7 | Pie Chart Distribution of Hinglish Hate Data | 19 |
| 8 | Hindi data being cleaned | 19 |
| 9 | Graphical Sentiment Distribution of Hindi Data | 20 |

## **INTRODUCTION**

**1.1 Multimodal and Multilingual based classification**

It is an innovative ML-driven content filtration system designed to tackle the increasing prevalence of harmful content on social media platforms. It addresses critical issues such as hate speech, offensive language that can harm online communities and individuals. It ensures effective, scalable, and classification of digital spaces.

**1.2 Key technologies and features include:**

**1.2.1 Sentiment Analysis:**

* Categorizes content into emotions like happiness, sadness, anger, and excitement, love.
* Enables proactive identification of harmful posts that could incite negative interactions.

**1.2.2 Machine Learning Models:**

* Utilizes state-of-the-art algorithms such as BERT, trained on diverse labeled datasets.
* Continuously improves through retraining on new data and evolving trends in social media content.

**1.2.3 Technological Framework:**

* Run the entire system locally using a Python environment, requiring minimal setup and infrastructure.
* PyTorch and Natural Language Toolkit (NLTK) for implementing the content moderation logic and machine learning models.

**1.3 Objectives of Project**

1. To identify and extract patterns from social media content based on sentiments.
2. To Classify and categorize the data using ML techniques.
3. To compare and contrast various ML methods for sentiment analysis based certain parameters.

**2. SYSTEM REQUIREMENTS**

**2.1 Hardware Requirements:**

* **Processor:**

Minimum:- Intel Core i5 or equivalent

Recommended:- Intel Core i7 or Ryzen 7 or equivalent for better performance

* **RAM:**

Minimum:- 8GB

Recommended:- 16GB for handling large datasets

* **Storage:**

At least 100GB SSD for dataset storage and processing

* **Graphics Card (GPU):**

Required for deep learning models (NVIDIA RTX 3060 or higher recommended)

* **Network:**

A stable and high-speed internet connection is required for efficient data processing

**2.2 Software Requirements:**

* **Operating System:**

Windows 10/11, Linux (Ubuntu)

* **Integrated Development Environment (IDE):**

Microsoft Visual Studio Code (VS Code) for running Python scripts, debugging, and testing.

* **Programming Language:**

Python- TensorFlow, PyTorch, OpenCV(Image processing for text extraction (OCR))

* **Python Interpreter (Required):**

Python **3.8 or later** is recommended for compatibility with ML libraries like TensorFlow, PyTorch

* **ML Libraries:**

Scikit-Learn, NLTK, SpaCy(Advanced NLP Library), etc.

* **Database (Optional for storage):**

NoSQL (MongoDB) or SQL (PostgreSQL)

**3. SOFTWARE REQUIREMENT ANALYSIS**

**3.1 Define the Problem**

With the increasing volume of user-generated content on social media platforms, the spread of hate speech, misinformation, cyberbullying, and offensive content has become a significant challenge. Traditional moderation techniques primarily rely on manual review and rule-based filtering, which are inefficient in handling the multimodal and multilingual nature of modern content.

Multimodal and multilingual based classification aims to address this issue by automating content classification and moderation using machine learning (ML) and natural language processing (NLP) techniques. However, there are several challenges:

* Detecting harmful content across different media types (text, images, audio) requires advanced AI techniques.
* Multilingual content classification is complex due to variations in language structure and meaning.
* Understanding sarcasm and implicit hate speech is difficult for traditional models.
* Ensuring scalability and efficiency while processing large volumes of data in real-time.

The proposed system seeks to overcome these limitations by implementing advanced NLP, deep learning models, and multimodal content analysis to enhance accuracy, speed, and effectiveness in content moderation.

**3.2 Technical Feasibility**

The feasibility study ensures that the system can be successfully developed using available resources, datasets, and technologies. It is built using:

| **Technology** | **Purpose** |
| --- | --- |
| **Python** | Core programming language for model implementation. |
| **TensorFlow / PyTorch** | Deep learning frameworks for training content classification models. |
| **Hugging Face Transformers** | Provides pre-trained NLP models like BERT/RoBERTa for text analysis. |
| **Scikit-learn** | Implements traditional ML algorithms such as SVM, Logistic Regression, and Random Forest. |
| **spaCy & NLTK** | Used for text preprocessing, including tokenization, stopword removal, and sentiment analysis. |
| **OpenCV** | Extracts text from images and memes for classification. |
| **Pandas & NumPy** | Data processing and manipulation for training models. |
| **Matplotlib & Seaborn** | Data visualization for model performance evaluation. |

These technologies ensure that the system is scalable, efficient, and capable of handling real-world content classification tasks. Deep learning models provide enhanced accuracy, while traditional ML models serve as backup classifiers for lower-resource environments.

**3.3 Define the Modules and Their Functionalities**

**3.3.1 Text Classification & Sentiment Analysis:**

* Detects hate speech, offensive language, and misinformation.
* Classifies content as positive, negative, or neutral etc. using BERT/RoBERTa, and SVM etc. models.

**3.3.2 Multimodal Processing (Text, Image):**

* **Web2Vec, NLTK, Scikit-learn:** Identifying patterns and extracting features.
* **OCR (Optical Character Recognition):** Extracts text from images and memes for content analysis.

**3.3.3 Multilingual Content Moderation:**

* Can useXLM-R or other models to classify text in multiple languages.

**3.3.4 Feature Extraction & Preprocessing:**

* Tokenization, stopword removal, stemming, and lemmatization for NLP tasks.
* TF-IDF and Word2Vec embeddings for feature representation.

**Current Status:**

* Feature extraction from various datasets is being processed to understand their pattern.
* Text classification model will be trained using ML algorithms
* Initial datasets collected from Kaggle (Jigsaw Toxic Comments, Sentiment140, etc.), Hugging Face, Github etc.

**4.SOFTWARE DESIGN**

**4.1 System Architecture:**

The system follows a modular design to support text, image classification, ensuring flexibility for future improvements. The Architecture Components are:

1. **Preprocessing Layer:**
   * Cleaning of data, identifying patterns, tokenization, stopword removal for text.
2. **Feature Extraction Layer:**
   * Visualizes sentiment and content distribution using graphs or pie charts, showing proportions and distributions of emotions (happy, sad, neutral etc) and classifications (harmful, non-harmful, toxic, etc).
3. **Classification Layer:**

* Machine learning models trained on labeled datasets.
* Probabilistic classification to assess content safety levels.

1. **Output & Interpretation Layer:**

* Displays sentiment scores (positive, negative, neutral, etc.) based on text analysis.
* Provides accuracy metrics (Precision, Recall, F1-score) for the classification models.
* Visual representation of results using charts, graphs, or dashboards for better understanding.
* Categorization of content into predefined labels (e.g., spam, toxic, neutral, etc).

**4.2 DATA FLOW DIAGRAM:**

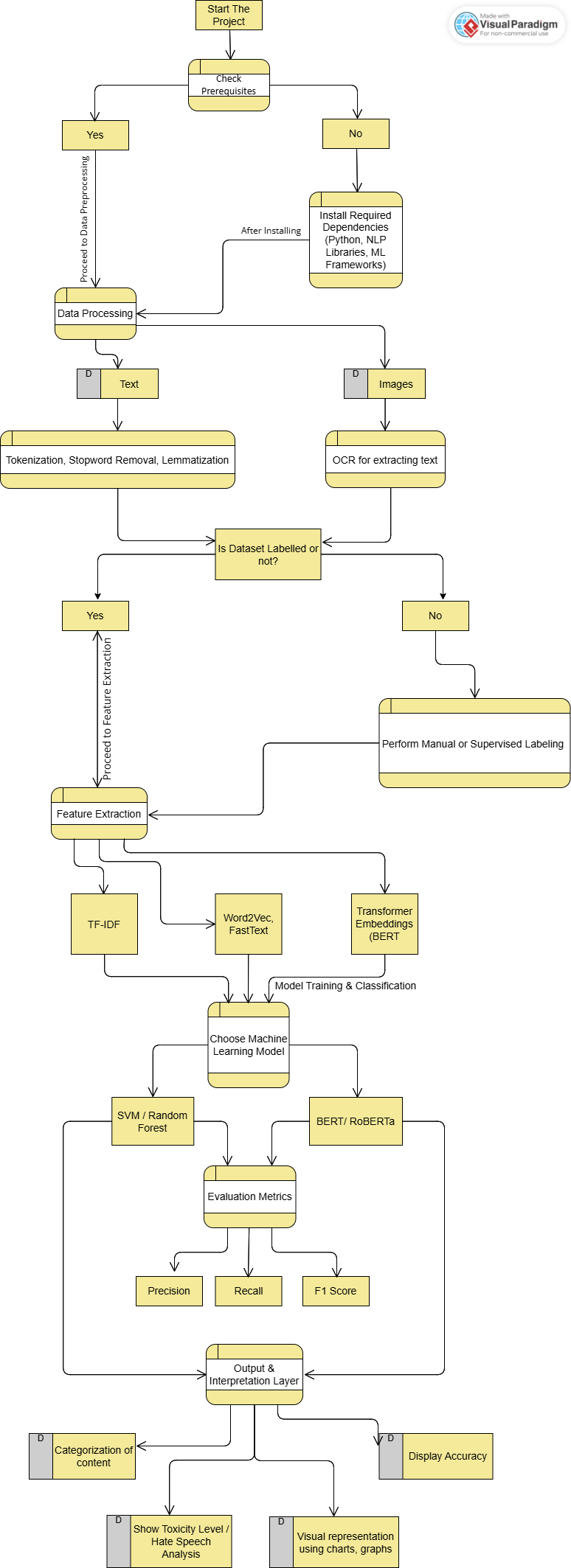
****

Fig 1. DFD

**4.3 Software Development Life Cycle (SDLC) – Agile Model**

For the multimodal and multilingual based classification, we follow the Agile methodology, which enables incremental development, iterative improvements, and flexibility in handling feature changes. Agile ensures that our classification model evolves based on feedback and continuous evaluation.

**4.3.1. Requirement Analysis (Sprint 1)**

* Identify essential components: Data Preprocessing, Feature Extraction, Model Training, and Output Interpretation
* Define the datasets needed for multilingual and multimodal sentiment analysis.
* Plan for future enhancements, such as semi-supervised training and real-time analysis.

**4.3.2. System Design (Sprint 2 & Sprint 3)**

* **Preprocessing Layer** – Cleaning, Tokenization, OCR, Stopword Removal
* **Feature Extraction Layer** – TF-IDF, Graphs or charts
* **Model Training & Classification** – ML models like SVM, Random Forest, Logistic Regression
* **Output & Interpretation Layer** – Sentiment classification, Accuracy scores etc

**4.3.3. Implementation (Sprint 4 - Sprint 6, Iterative Cycles)**

* Develop text classification using ML models.
* Implement data preprocessing (Tokenization, OCR for image text).
* Train SVM, Logistic Regression, and Random Forest models on labeled datasets.
* Validate sentiment classification results and fine-tune model performance.

**4.3.4 Testing (Sprint 7 - Sprint 8, Continuous Testing in Each Sprint)**

* Evaluate ML models based on Precision, Recall, and F1-Score.
* Perform unit testing on data processing and feature extraction pipelines.
* Validate multilingual sentiment classification accuracy across different datasets.

**4.3.5. Deployment & Maintenance (Sprint 9 - Ongoing Iterations)**

* Continuously monitor performance and retrain models with new data.
* Improve model accuracy using semi-supervised learning techniques.

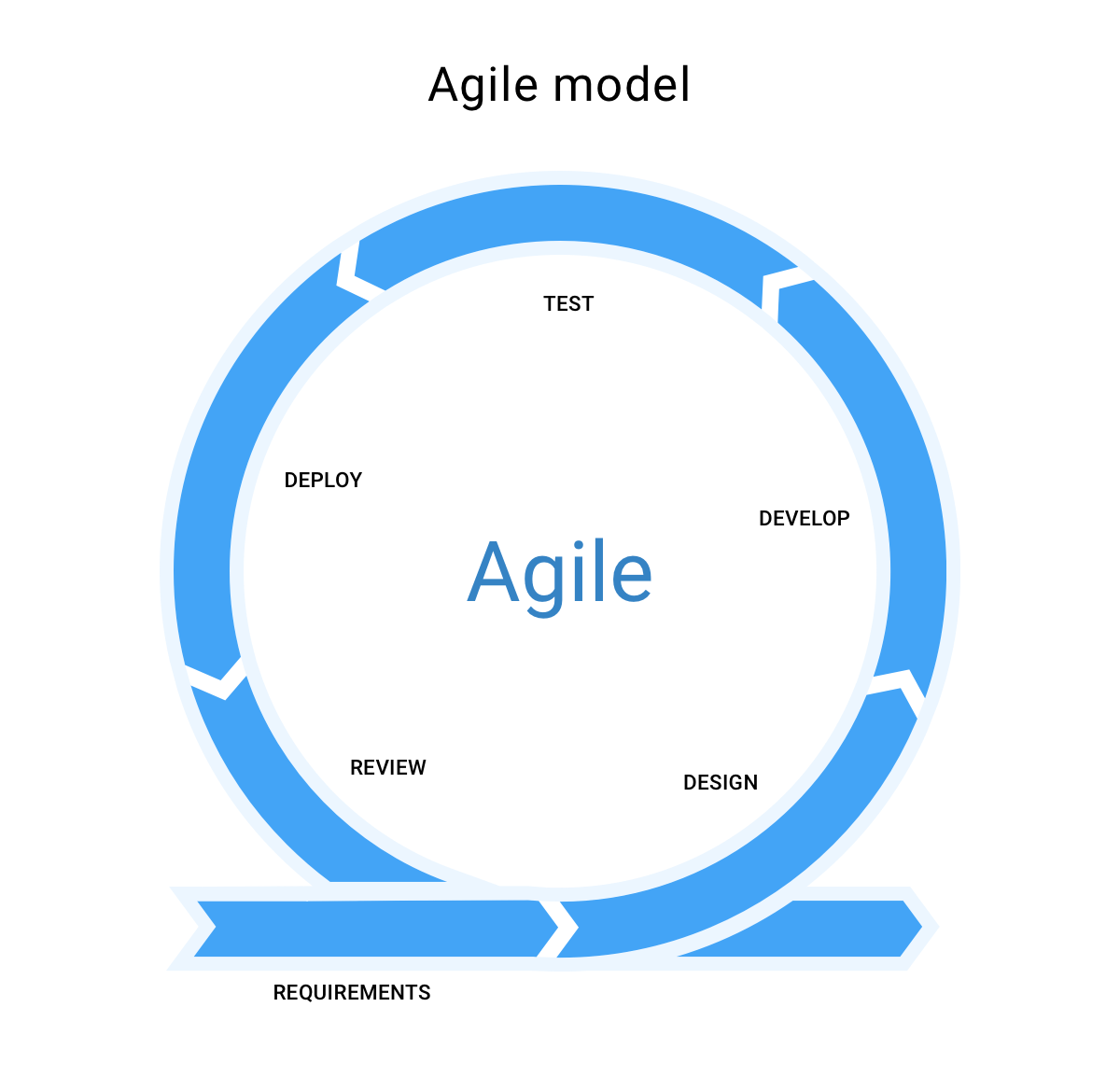


Fig 2: Agile Model

**5. TESTING MODULE**

**5.1 Data Preprocessing Test**

* Test Case: System should preprocess text (tokenization, stopword removal)
* Expected Outcome: Text should be cleaned and extracted correctly.
* Status: In Progress

**5.2 Feature Extraction Test**

* Test Case: System should extract word embeddings (Word2Vec).
* Expected Outcome: Extracted features should correctly represent text semantics.
* Status: Passed

**5.3 Classification Accuracy Test**

* Test Case: Model should classify text into appropriate categories (positive, negative, neutral).
* Expected Outcome: Classification should achieve high accuracy (above 85%).
* Status: In Progress

**5.4 Multilingual Support Test**

* Test Case: System should classify sentiment across different languages.
* Expected Outcome: Model should retain accuracy across multiple languages.
* Status: Pending

**5.5 Toxicity Detection Test**

* Test Case: System should identify and flag offensive or toxic content.
* Expected Outcome: Toxic messages should be flagged with scores.
* Status: Pending

**5.5 Sentiment Visualization Test**

* Test Case: Output layer should display sentiment scores and classification results.
* Expected Outcome: Sentiment trends should be visualized (bar charts, percentages).
* Status: Passed

**5.6 Usability Testing**

* Execution Test: Users should be able to run the Python script without errors in the interpreter.
* Input Handling Test: System should accept text and image inputs through the Python environment.
* Output Display Test: Processed results (sentiment scores, classifications) should be clearly printed in the interpreter.
* Performance Test: Execution time should remain optimal for large datasets.

**5.7 Future Testing Scope**

* Semi-Supervised Learning Test: Model should improve with additional unlabelled data.
* Scalability Testing: Assess system performance when handling large multilingual datasets.

**6. PERFORMANCE OF THE PROJECT**

The Multimodal and Multilingual Based Classification project is designed to process and analyze text efficiently using machine learning models. Since the project runs in a Python interpreter without a UI, the performance evaluation is based on model execution speed, classification accuracy, and resource utilization.

**6.1. Data Processing Performance**

* **Text Preprocessing:** Data cleaning, feature extraction and identifying patterns should complete in minimal time.
* **Observed Performance:** Data cleaning and feature extraction complete in 50-300ms per input, depending on text length and complexity.

**6.2. Resource Utilization**

* **CPU Usage:** Moderate usage depending on model complexity.
* **Memory Usage:** Efficient memory allocation to process large datasets without system crashes.
* **Observed Performance:**
  + **CPU Usage:** ~20-40% during training, 10-20% during inference.
  + **Memory Usage:** Varies between 500MB-2GB, depending on model size and dataset.

**6.3. Stability & Error Handling**

* Robust error handling should prevent crashes during execution.
* The script should handle large datasets without memory leaks.
* **Observed Performance:** No unexpected crashes detected during testing.

**6.4. Scalability & Future Optimization**

* **Expected Outcome:** The system should handle increased data loads .
* **Future Scope:** Optimize performance with semi-supervised learning to improve accuracy.

**7. SCREEN OUTCOMES**

* It is some data that is present in our datasets

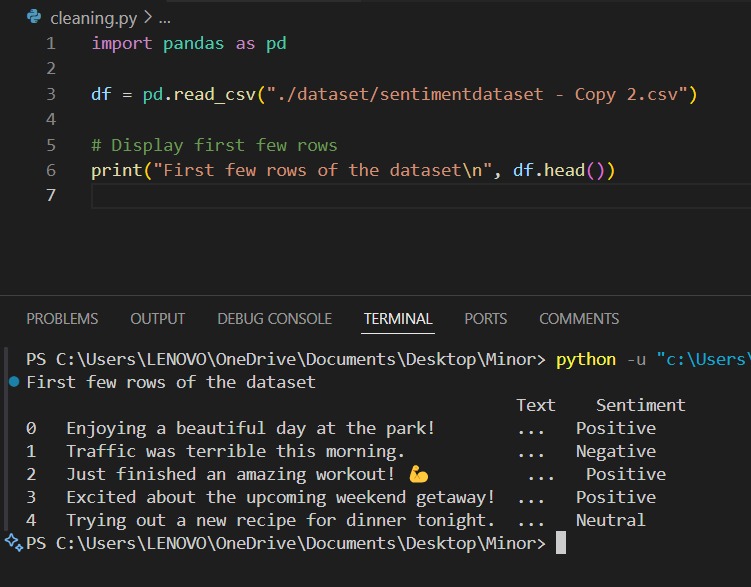


Fig 3:Display of some data in row

* It is training and test splitting data

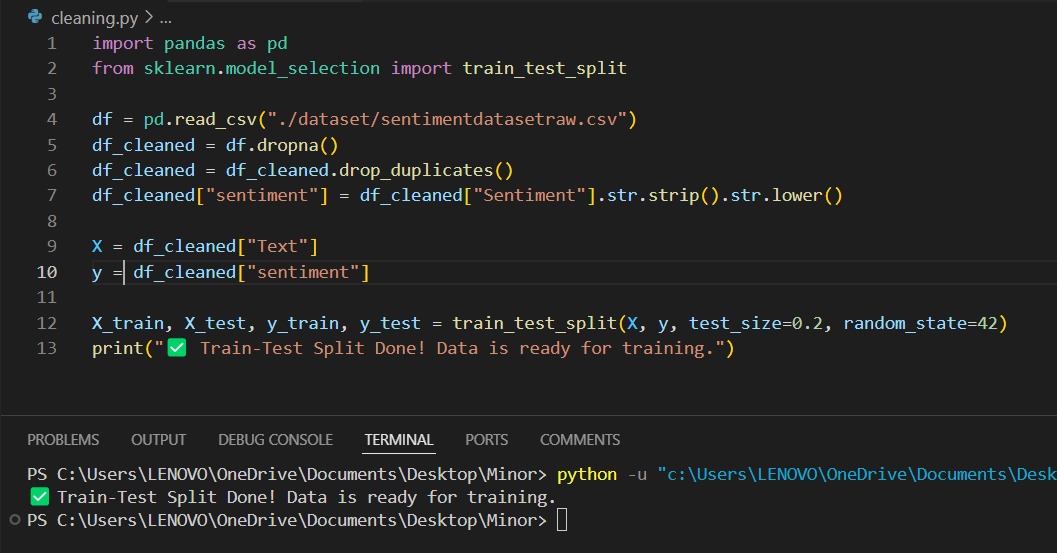


Fig 4:Training and Testing Split Data

* Unique sentiment labels that we receive after cleaning

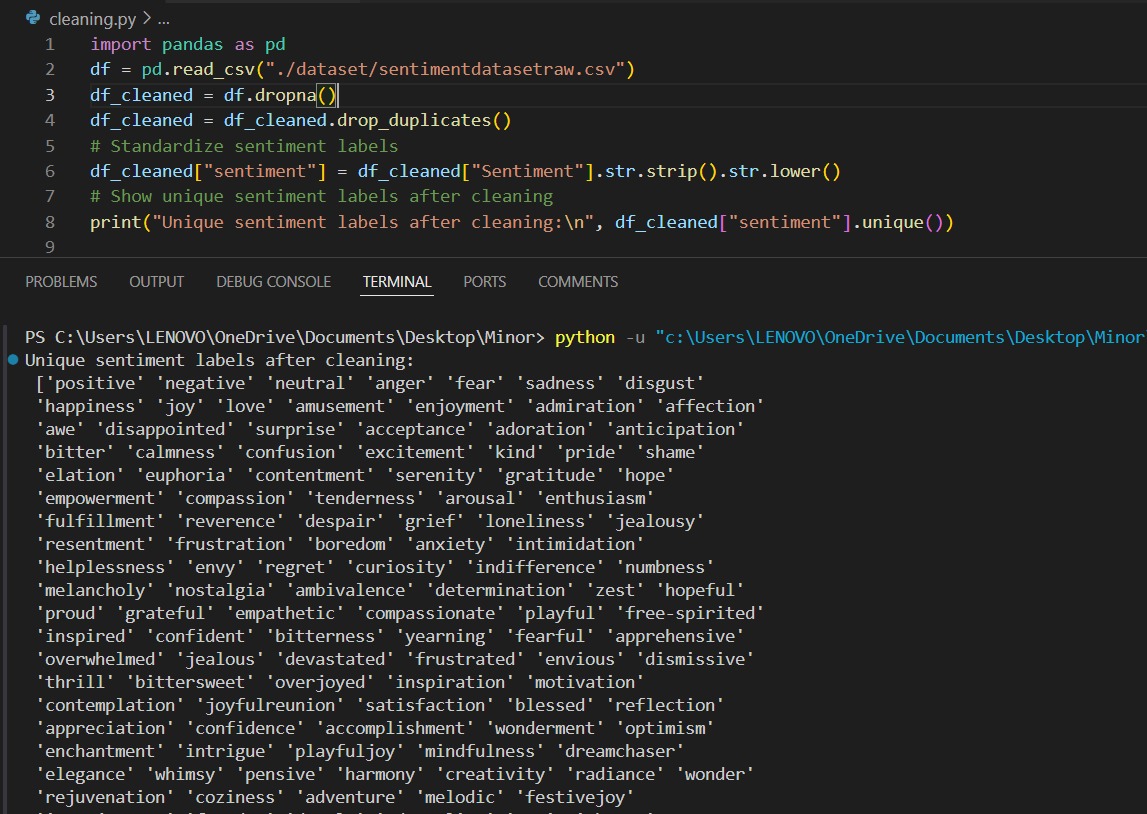


Fig 5:Sentiment Labels

* Number of sentiment labels in the particular dataset

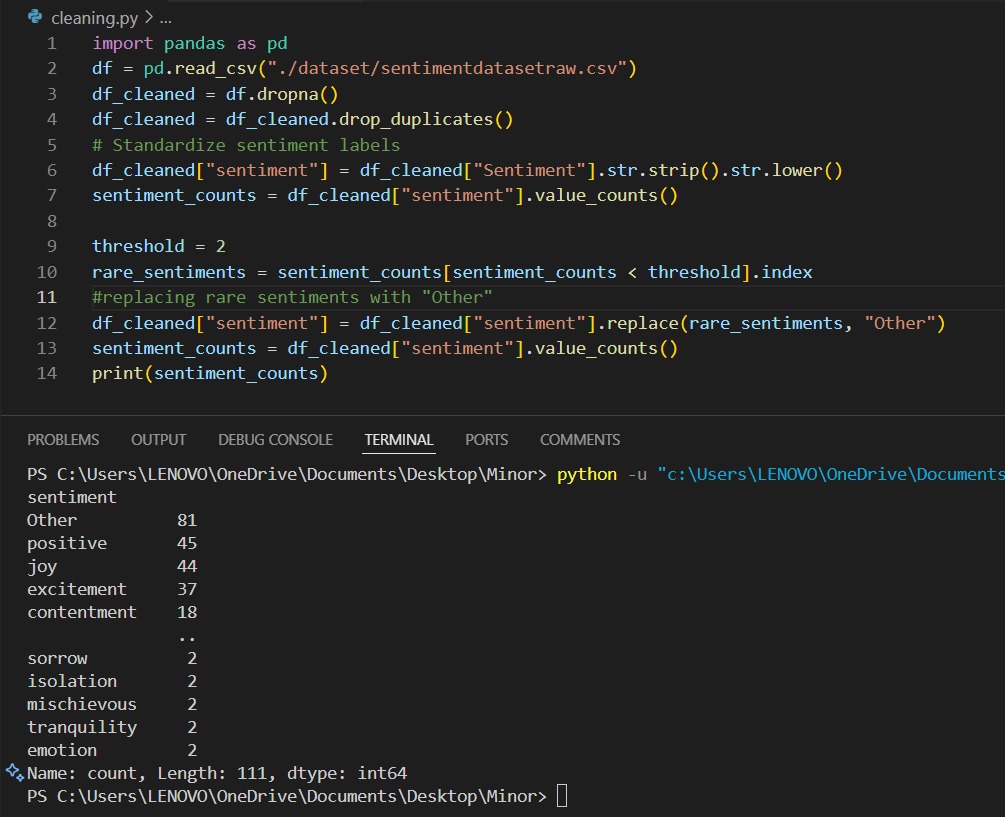


Fig 6:Count of sentiments in the dataset

* Pie Chart Distribution for Hinglish Hate Data

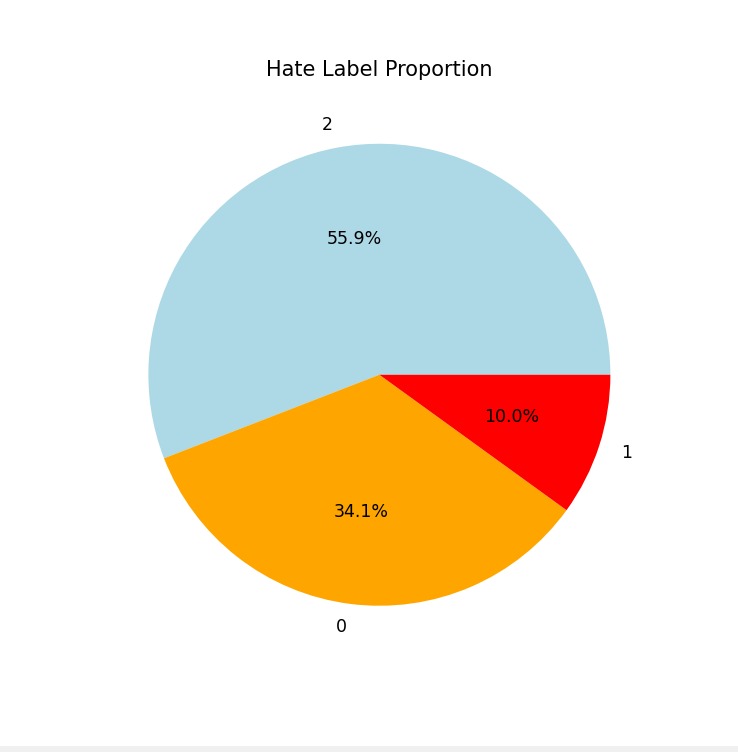


Fig 7:Pie Chart Distribution of hate data

* Cleaning of particular Hindi Dataset

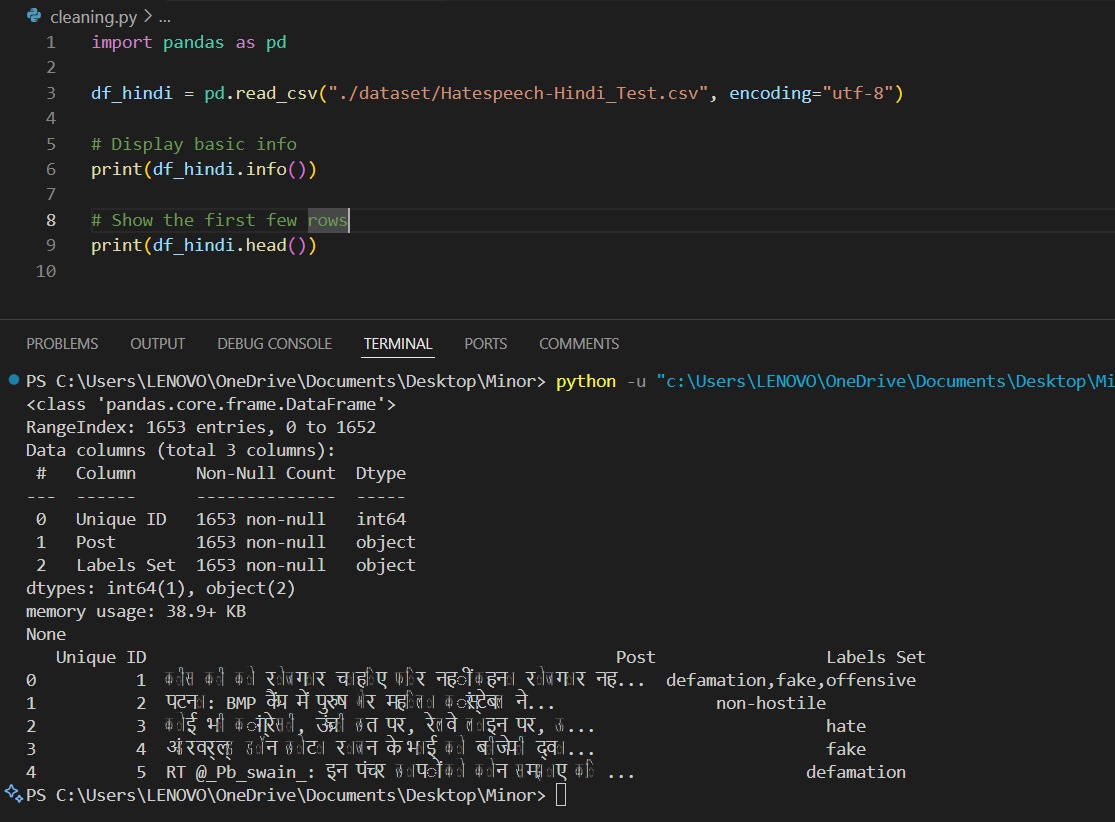


Fig 8:Hindi data being cleaned

* Some sentiment distribution of available Hindi Data

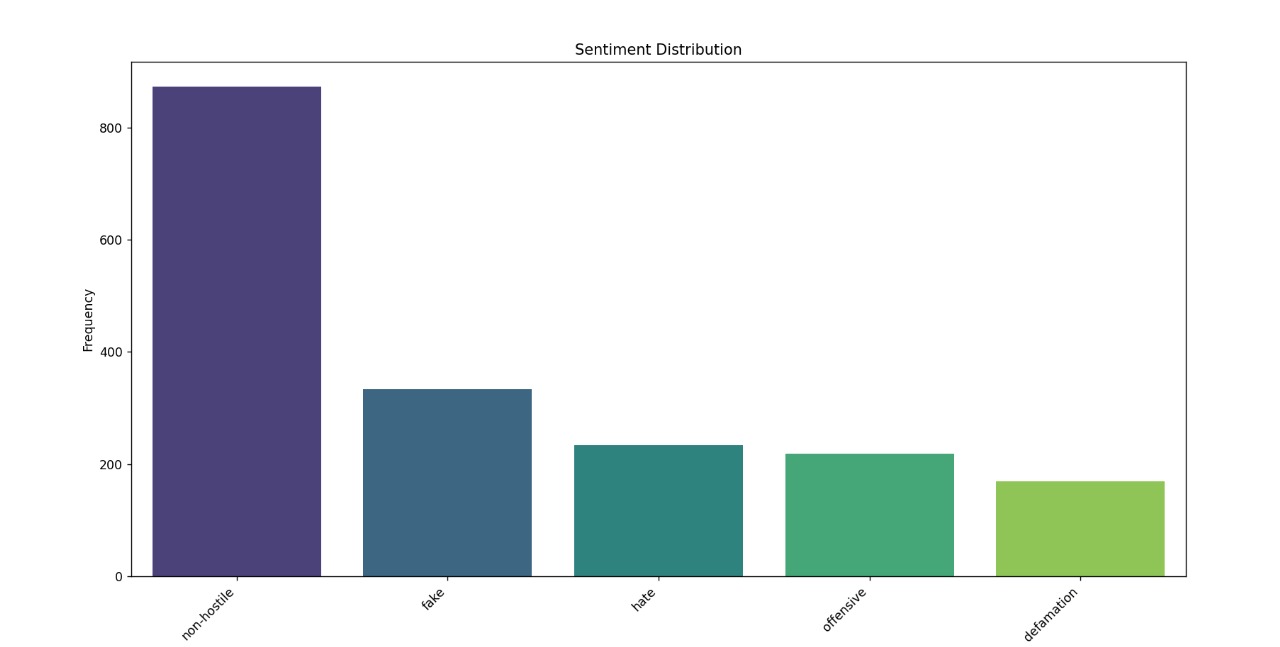


Fig 9:Graphical Sentiment Distribution of Hindi Data

**8. REFERENCES**

**[1] Python –** Python documentation [Our Documentation | Python.org](https://www.python.org/doc/)

**[2] PyTorch** - PyTorch documentation [PyTorch documentation — PyTorch 2.5 documentation](https://pytorch.org/docs/stable/index.html)

**[3] NLTK -** NLTK, is a suite of libraries and programs for symbolic and statistical natural language processing (NLP) for English. [NLTK :: Natural Language Toolkit](https://www.nltk.org/)

**[4] Scikit-Learn** - Scikit-Learn is a machine learning library for Python that provides simple and efficient tools for data mining and analysis. It supports various classification, regression, and clustering algorithms. **Scikit-Learn Documentation** — Scikit-Learn Official Documentation.

**[5] Canva** - Design Anything. Publish Anywhere <https://www.canva.com/>

**[6] Word2Vec -** Word2vec is a technique in natural language processing for obtaining vector representations of words. [models.word2vec – Word2vec embeddings — gensim](https://radimrehurek.com/gensim/models/word2vec.html)

**[7]** Multi-modal Hate Speech Detection using Machine Learning: Fariha Tahosin Boishakhi,

Ponkoj Chandra Shill, Md. Golam Rabiul Alam

**[8]** Machine Learning based Automatic Hate Speech Recognition System: P. William, Ritik Gade, Rupesh Chaudhari, A.B. Pawar

**[9]** Exploring Machine Learning Methods for Hate Speech Detection on Social Media:Deepti Negi, Mahesh Manchanda, Aditi Kala, Aditya Harbola

**[10]** Detection of hate speech in Social media memes: Ajay Nayak, Anupam Agrawal

**[11]** Detecting Hate Speech on Social Media with Respect to Adolescent Vulnerability: Anna Chiu, Kanika Sood, Ariadne Rincon, Davina Doran

**[12]** Emotion Recognition Using Multimodal Approach: Samiksha Saini, Rohan Rao, Vinit Vaichole, Anand Rane. Deepa Abin

**[13]** Multimodal Social Media Sentiment Analysis: Pratyush Muthukumar, Mubarak Ali Seyed Ibrahim

**[14]** A Review on Sentiment Analysis of Twitter Data Using Machine Learning Techniques:

Ankita Srivastava

**[15]** SVM Optimization for Sentiment Analysis: Munir Ahmad, Shabib Aftab