# Phase-2 Submission

**Student Name:** *Subash S*

**Register Number:** *712523106018*

**Institution:** *ppg institute of technology*

**Department:** *B.E. Electronics and Communication Engineering*

**Date of Submission:** 08/05/2025

**GitHub Repository Link:**
https://github.com/subash232006/NM_SUBASH_DS

## 1. Problem Statement

*Social media is a major outlet for emotional expression, but the sheer volume of content makes manual analysis impractical. This project uses **NLP-based multi-class classification** to detect emotions—like happiness, anger, sadness, fear, and neutrality—in social media posts.*

*Type of Problem: Multi-class Text Classification (Supervised Learning)*

*Why It Matters:*

- *Enables mental health monitoring*
- *Improves customer feedback insights*
- *Supports public sentiment analysis*
- *Enhances user safety and platform experience*

## 2. Project Objectives

*The goal is to build an NLP-based system that classifies social media posts into emotional categories like happiness, sadness, anger, fear, and neutrality.*

*Key Technical Objectives:*

- *Develop a multi-class classification model for emotion detection.*

- *Apply text preprocessing techniques (tokenization, stop-word removal, vectorization).*

- *Optimize model performance using accuracy, precision, recall, and F1-score.*

- *Ensure interpretability using tools like confusion matrices or SHAP/LIME.*

- *(Optional) Deploy a prototype for real-time emotion analysis.*

*Updated Goal Post Data Exploration:*

*Focus shifted from basic sentiment polarity to detailed emotion classification to capture richer, more actionable insights.*

## 3. Flowchart of the Project Workflow
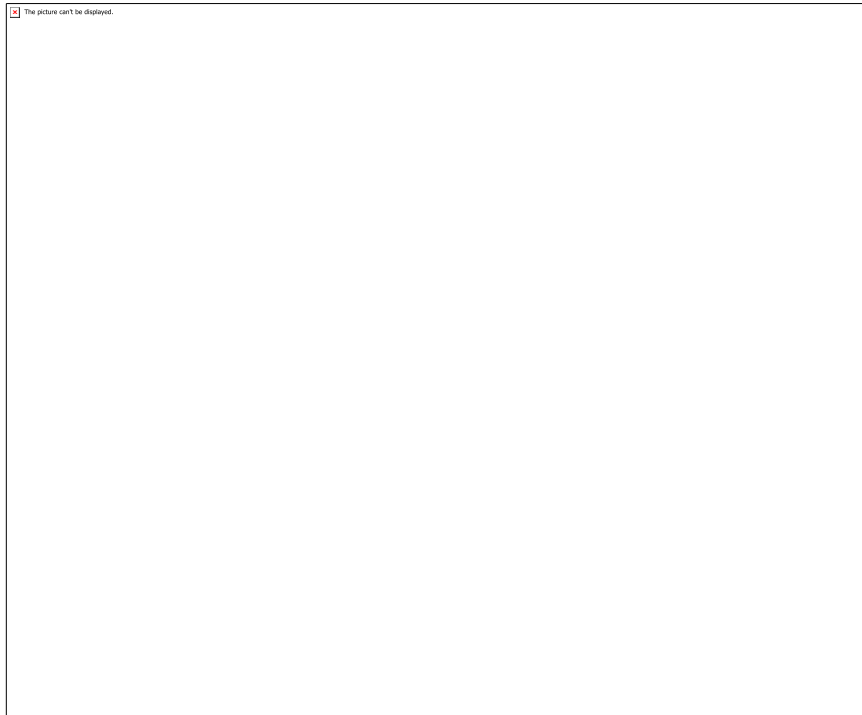
# 4. Data Description

*The dataset used for this project is a collection of* text-based social media posts, *specifically designed for* emotion classification. *It was sourced from* Kaggle, *titled* "Emotion Detection from Text", *which contains labeled examples of social media text mapped to emotional categories.*

- *Source: Kaggle (e.g., Emotion Detection from Text, CrowdFlower Emotion Dataset, or similar)*

- *Type of Data: Unstructured text data*

- *Number of Records: ~20,000+ text samples (varies depending on dataset used)*

- *Number of Features: 1 input feature (text), 1 target label (emotion)*

- *Dataset Nature: Static – does not update in real-time*

- *Target Variable: Emotion labels such as happy, sad, angry, fear, surprise, neutral, etc.*

- *Learning Type: Supervised Learning (Multi-class classification)*

# 5. Data Preprocessing

*Handling Missing Values: All rows with missing text or emotion labels were removed to ensure data completeness.*

- ***Removing Duplicates:*** *Duplicate text entries were identified and dropped to avoid data redundancy.*

- ***Outlier Treatment:*** *Extremely short texts (e.g., less than 3 words) were considered outliers and removed, as they provide little semantic value.*

- ***Data Type Conversion:*** *Text data was converted to string format, and emotion labels were cast as categorical variables for consistency.*

- ***Text Cleaning and Preparation:*** *Text was cleaned by removing URLs, punctuation, and converting to lowercase, followed by tokenization, stop word removal, and lemmatization.*

- ***Categorical Encoding:*** *Emotion categories were encoded using **Label Encoding** to prepare them for supervised learning algorithms.*

- ***Text Vectorization and Normalization:*** *TF-IDF vectorization transformed text into numerical format, providing normalized feature values suitable for model input.*

## 6. Exploratory Data Analysis (EDA)

### 🔍 *Univariate Analysis*

*Emotion Distribution: Count plot shows imbalance—some emotions (e.g., joy, sad) are more common than others like fear or surprise.*

*Text Length: Histogram reveals most social media posts are short (under 30 words), aligning with platform norms.*

### 🔍 *Bivariate/Multivariate Analysis*

*Boxplot (Text Length vs Emotion): Emotions like anger and fear often correspond to longer posts, while happy and neutral are shorter.*

*Word Frequency Patterns: Emotion-specific words like "sorry", "love", "hate", and "great" are commonly tied to respective sentiments.*

*Correlation Matrix (TF-IDF features): Reveals low multicollinearity, but some overlap in terms used across emotions.*

### 🔍 *Insights Summary*

*Class Imbalance: May lead to biased model predictions—should address using oversampling or class weighting.*

*Feature Influence: Text length and word usage are key predictors; they help differentiate emotions and improve classification accuracy.*

# 7. Feature Engineering

*Text Length & Word Count: The number of words and characters in posts helps capture emotional intensity, with longer texts indicating stronger emotions like anger or fear.*

- *Sentiment Polarity Score: Using tools like VADER or TextBlob to assign a positive/negative sentiment, complementing the multi-class emotion classification.*

- *POS Ratios: The ratio of adjectives, nouns, and verbs to identify emotion-specific language patterns (e.g., joy vs. anger).*

- *Top-N Keywords (TF-IDF): Extract important keywords using TF-IDF to focus on words strongly linked to specific emotions (e.g., love for joy).*

- *Text Length Binning: Categorizing text length into bins (short, medium, long) to help capture emotion intensity patterns.*

- *PCA for Dimensionality Reduction: Reduce feature space with PCA to improve efficiency and reduce overfitting.*

- *Temporal Features: Extract date and time features (e.g., day of the week) to capture patterns in emotion based on timing.*

# 8. Model Building

*Models Chosen:*
- *Logistic Regression: Simple, interpretable, and effective for text classification.*

- *Random Forest: Robust, handles complex data relationships well, and avoids overfitting.*

- *Data Split:*

- *Split data into **training** and **testing** sets using **stratified sampling**.*

- ***Training**:*

- *Use **TF-IDF** for text vectorization.*

- *Train both **Logistic Regression** and **Random Forest** models.*

- ***Evaluation Metrics**:*

- *Evaluate using **accuracy, precision, recall**, and **F1-score**.*

- ***Model Comparison**:*

- *Compare both models, selecting the one with the best **F1-score** for emotion classification.*

## 9. Visualization of Results & Model Insights

***Confusion Matrix**: Shows actual vs predicted emotions, highlighting misclassifications. High diagonal values indicate correct predictions.*

- ***ROC Curve**: Evaluates model performance by plotting True Positive Rate vs False Positive Rate. A higher AUC indicates better model performance.*

- ***Feature Importance Plot**: Displays which features (e.g., specific words) are most influential in predicting emotions.*

- ***Residual Plot**: Visualizes prediction errors; random scatter around zero indicates a well-fitting model.*

- ***Model Performance Comparison**: Compares models (e.g., Logistic Regression vs. Random Forest) on metrics like accuracy, precision, and F1-score to select the best performer.*

## 10. Tools and Technologies Used

- ⑩ ***Programming Language**: Python*

- ***IDE/Notebook**: Google Co lab, Jupiter Notebook*

- ***Libraries**: pandas, NumPy, seaborn, matplotlib, scikit-learn, X Boost*

- ***Visualization Tools****: Portly, Tableau (optional for advanced visualizations)*

## 11. Team Members and Contributions

| TEAM MEMBER | ROLE | RESPONSIBILITIES |
|---|---|---|
| SUBASH S | Team Lead, Model Development | Responsible for developing and training machine learning models for fake news detection. |
| RAJESHKUMAR N | Data Cleaning, Feature Engineering | Handled data preprocessing, cleaning, and feature extraction for model training. |
| KAMALESH S | EDA, Data Visualization | Conducted exploratory data analysis to understand data distribution and patterns. |
| PRASANNA S D | Documentation and Reporting | Managed project documentation, including technical reports and presentation materials. |
| SANJAY E | NLP Implementation, Model Evaluation | Implemented NLP algorithms for text processing and analysis. |