**Phase-2 Submission Template**

**Student Name:** K. NITHYAKALA

**Register Number:** 721123104063

**Institution:** STUDY WORLD COLLEGE OF ENGINEERING

**Department:** COMPUTER SCIENCE AND ENGINEERING

**Date of Submission:** 09/05/2025

**Github Repository Link:** [**https://github.com/AshokKumarReddy07/Ashok\_Phase\_-2/blob/480e40d7ada0dac7d391cf258a345155e4f5b1c7/Phase-2\_code**](https://github.com/AshokKumarReddy07/Ashok_Phase_-2/blob/480e40d7ada0dac7d391cf258a345155e4f5b1c7/Phase-2_code)

# 1. Problem Statement

# *In today's digital age, people often express emotions, opinions, and sentiments on social media. However, understanding the emotional tone of these conversations is challenging due to informal language, slang, sarcasm, and context variation. This project aims to build an AI-based sentiment analysis system that accurately identifies and decodes emotional cues from social media text using Natural Language Processing (NLP) techniques.*

# 2. Project Objectives

* Collect and preprocess social media data (e.g., tweets, comments).
* Build a sentiment classification pipeline (positive, negative, neutral, and emotion-based).
* Implement vectorization techniques like TF-IDF and word embeddings.
* Train traditional ML and DL models (e.g., Logistic Regression, LSTM, BERT).
* Visualize results and insights.
* Ensure explainability and interpretability of sentiment predictions.

.

**3. Flowchart of the Project Workflow**

Collect Social Media Data

↓

Text Cleaning & Preprocessing

↓

Feature Extraction (TF-IDF / BERT)

↓

Train Sentiment Classifier

↓

Predict Emotion/Sentiment from Input Text

↓

Visualize Sentiment Insights

# 4. Data Description

**User Data**

|  |  |  |
| --- | --- | --- |
| **Field** | **Description** | **Type** |
| id | Unique ID of the post/comment | Integer |
| text | Social media message content | String |
| sentiment | Sentiment label (Positive/Negative/Neutral or emotions like Happy, Sad, Angry) | Categorical |
| timestamp | Date/time of post | DateTime |
| username | User who posted | String |
| language | Language of the text | String |

# 5. Data Preprocessing

**Data Cleaning:**

* Remove irrelevant content (e.g., URLs, mentions, hashtags, emojis, special symbols).
* Eliminate duplicates and null entries.
* Handle missing values appropriately.

**Text Normalization:**

* Convert text to lowercase.
* Remove punctuation, numbers, and stopwords.
* Perform tokenization using NLTK or spaCy.
* Apply lemmatization to retain root forms of words (e.g., “running” → “run”).

**Encoding & Aggregation:**

* Encode categorical features (if applicable).
* Create additional fields like:
  + **Message length** (word/character count)
  + **Sentiment polarity score** (from rule-based models like TextBlob)
  + **Time of posting** (for temporal analysis)

# 6. Exploratory Data Analysis (EDA) :

**EDA offers insights into the structure and emotion trends in the dataset:**

* **Distribution of Sentiment Classes:**Analyze how many posts are positive, negative, neutral, etc.
* **Top Keywords per Emotion:**Use TF-IDF and count vectorizer to extract dominant terms associated with each sentiment/emotion.
* **Emotion Frequency Over Time:**Visualize how emotional expression changes over different days/weeks.
* User Behavior Patterns (if user data is included):  
  Understand which users frequently post emotional content.
* **Word Clouds for Each Sentiment:**Generate visual summaries of most-used words in each sentiment category.

# 7. Feature Engineering

Feature Engineering for Movie Recommendation:

**User Features**

* **Bag of Words / TF-IDF Vectors:**  
  Basic word frequency-based representations.
* **Word Embeddings (Semantic Features):**Word2Vec / GloVe: Capture semantic similarity between words.
* **BERT Embeddings:** Use transformer-based representations to capture context.

** Sentiment & Emotion Scores (External Lexicons):**

* VADER: For social media-specific sentiment scoring.
* NRC Lexicon: For mapping words to emotions (e.g., anger, joy).

** Custom Features:**

* Use of exclamation marks, capital letters (indicating emphasis)
* Emoticon counts
* Text readability index

# 8 .Model Building :

Model Building Approaches

**Machine Learning Models:**

* Naive Bayes
* Logistic Regression
* Support Vector Machines (SVM)
* Random Forest / XGBoost

 **Deep Learning Models:**

* LSTM / GRU (for sequence data)
* CNN (for spatial features in text)
* BERT or RoBERTa fine-tuned on sentiment datasets

**Training Pipeline:**

* Use Train/Test split or Stratified K-Fold cross-validation.
* Optimize hyperparameters using GridSearchCV or Optuna.
* Implement early stopping and dropout (for DL models) to avoid overfitting.

# 9*.*Visualization of Results & Model Insights :

Visualizing outcomes helps interpret model behavior and diagnose errors:

1. **Confusion Matrix**
2. **ROC Curve & AUC Score**
3. **Accuracy/Loss Curves:**
4. **Feature Importance (for tree models)**
5. **Word Clouds**
6. **Embedding Visualization**

**Model Insights Uncovered Through Visualization**

* **Confusion Matrix**: Shows how many texts were correctly vs. incorrectly classified under each emotion label. Helps identify which emotions the model confuses most often.
* **ROC Curve & AUC Score**: Useful for binary sentiment classification (e.g., positive vs. negative). For multi-label classification, use a One-vs-Rest approach
* **Precision-Recall Curve**: Useful for imbalanced datasets to highlight the trade-off.
* **Accuracy/Loss Curves**: Plot training vs. validation loss/accuracy to monitor convergence and detect overfitting.
* **Emotion Distribution Graphs:**

Visual bar charts showing how different emotions are represented in predictions and ground truth.

* **Feature Importance (for tree models)**: Identify which keywords or features (e.g., word embeddings, text length) most influence classification.
* **Word Clouds**: Visual comparison of frequently used words in fake vs. real news articles.
* **Embedding Visualization (optional)**: Use t-SNE or UMAP to plot high-dimensional BERT or Word2Vec embeddings to observe clusters of similar sentiment texts.

# 10. Tools and Technologies Used :

* 1. **Data Handling & Preprocessing**

|  |
| --- |
|  |

|  |  |
| --- | --- |
| **Tool/Library** | **Purpose** |
| Pandas, NumPy | Data loading, cleaning, and numerical transformations |
| NLTK / spaCy | Tokenization, lemmatization, and text normalization |
| Langdetect | Language detection and filtering |
| Scikit-learn | Text vectorization (TF-IDF), splitting data, basic preprocessing |

**2.Data Exploration & Visualization**

|  |  |
| --- | --- |
| **Tool/Library** | **Purpose** |
| Matplotlib | Bar plots, line plots, confusion matrix |
| Seaborn | Correlation heatmaps, distribution plots |
| WordCloud | Visual representation of emotion-specific vocabulary |
| Plotly / Altair | Interactive charts for timeline-based emotion trends |

**3. Machine Learning & Recommendation Algorithms**

|  |  |
| --- | --- |
| **Tool/Library** | **Purpose** |
| Scikit-learn | TF-IDF vectorization |
| Gensim | Word2Vec / GloVe embeddings |
| Transformers (HuggingFace) | BERT/RoBERTa contextual embeddings |
| VADER / NRC | Sentiment/emotion lexicons |

**4. Deep Learning**

|  |  |
| --- | --- |
| **Tool/Library** | **Purpose** |
| Scikit-learn | ML classifiers: Logistic Regression, SVM, etc. |
| XGBoost / LightGBM | Gradient boosting models |

**5. Evaluation & Monitoring**

|  |  |
| --- | --- |
| **Tool/Library** | **Purpose** |
| MLflow | Track model parameters, runs, and metrics |
| Weights & Biases | Visualize training progress and compare models |
| TensorBoard | Monitor deep learning metrics and loss/accuracy graphs |

# 11.Team Members and Contributions :

* 1. **[NITHYAKALA.K**]: Data collection, preprocessing, feature engineering
  2. **[MUKESH.R ]:** Model development and evaluation
  3. **[MUGHILAN.D]:** Dashboard and visualization development
  4. **[PAVUNRAJ.M]:** Documentation, testing, and project management.