**Phase-2 Submission Template**

**Student Name: MOHANA PRIYA**

**Register Number:** 710623104047

**Institute: CSI college of engineering**

**Department: computer science and engineering**

**Date of Submission:** 10.05.2025

**Github repository link:https://github.com/indhuja20020/Sentiment-analysis-**

# 1. Problem Statement

Problem Statement:

In today’s digital age, vast amounts of user-generated content are created on social media platforms, review websites, and forums. Understanding the emotions and opinions expressed in this content is essential for businesses, policymakers, and researchers. However, manually analyzing such data is time-consuming and impractical.

This project aims to develop a sentiment analysis system that can automatically identify and classify the emotional tone (positive, negative, or neutral) of textual data collected from social conversations. By leveraging natural language processing (NLP) and machine learning techniques, the system will process large volumes of unstructured text to uncover patterns in public sentiment and emotional trends.

The ultimate goal is to decode emotions from online conversations to support decision-making, improve customer satisfaction, and gain insights into public opinion on various topics.

# 2. Project Objectives

Project Objectives:

1. Data Collection:

Gather a comprehensive dataset of social conversations (e.g., from Twitter, Reddit, or product review platforms) that reflect diverse emotional expressions.

2. Data Preprocessing:

Clean and preprocess the textual data by removing noise, handling missing values, tokenizing, and normalizing text (e.g., stemming/lemmatization).

3. Sentiment Labeling:

Annotate the dataset with sentiment labels (positive, negative, neutral) either through manual labeling, crowdsourcing, or using pre-labeled datasets.

4. Model Development:

Build and train machine learning or deep learning models (e.g., Naive Bayes, SVM, LSTM, BERT) to classify sentiment from the processed text.

**3. Flowchart of the Project Workflow**

Flow Chart:

|  |
| --- |
|  |

│ Data Collection │

|  |  |
| --- | --- |
|  |  |

│

┌────────▼────────┐

│ Data Preprocessing│

│(Cleaning, Tokenization, │

│ Lemmatization, etc.) │

|  |  |
| --- | --- |
|  |  |

│

┌────────▼────────┐

│ Sentiment Labeling│

│ (Manual/Automatic)│

|  |  |
| --- | --- |
|  |  |

│

┌────────▼────────┐

│ Model Training │

│(ML/DL algorithms)│

|  |  |
| --- | --- |
|  |  |

│

┌────────▼────────┐

│ Model Evaluation │

│(Accuracy, F1-score)│

|  |  |
| --- | --- |
|  |  |

│

┌────────▼────────┐

│ Emotion/Sentiment│

│ Classification│

|  |  |
| --- | --- |
|  |  |

│

┌────────▼────────┐

│ Visualization & │

│ Insight Extraction │

|  |  |
| --- | --- |
|  |  |

│

┌────────▼────────┐

│ Deployment/UI │

│ (Web/App/Tool) │

|  |
| --- |
|  |

# 4. Data Description

Data Description:

The dataset used in this project comprises textual data collected from [e.g., Twitter, Reddit, or product review platforms]. Each entry in the dataset represents a social conversation or user-generated comment and is annotated with sentiment polarity or emotion labels.

Key Attributes:

# 5. Data Preprocessing

[Perform and document data cleaning and preparation.

* Handle missing values (removal, imputation, etc.).

* Remove or justify duplicate records.

* Detect and treat outliers.

* Convert data types and ensure consistency.

* Encode categorical variables (label encoding, one-hot encoding).

* Normalize or standardize features where required.
* Document and explain each transformation step clearly in code and markdown.]

# 6. Exploratory Data Analysis (EDA)

Exploratory Data Analysis (EDA)

The goal of EDA is to understand the structure, patterns, and distributions within the dataset, as well as to identify any anomalies or biases that may affect model performance.

1. Sentiment Distribution

A bar chart shows the count of each sentiment class (positive, negative, neutral).

Example:

Positive: 42%

Negative: 36%

Neutral: 22%

Insight: The dataset is moderately imbalanced, requiring potential resampling or weighted loss.

1. Text Length Analysis:

Average, minimum, and maximum lengths (in words/tokens) of the text entries were calculated.

Distribution indicates most texts are between 10–50 words.

1. Word Cloud Visualization:

Separate word clouds were generated for each sentiment category.

Positive texts featured words like “love,” “great,” and “happy.”

Negative texts had “hate,” “bad,” and “angry.”

Neutral texts were more factual and context-based.

1. Frequent Words and N-grams:

Top unigrams and bigrams per sentiment were analyzed.

Common unigrams: “good,” “bad,” “product,” “service”

Common bigrams: “not good,” “very happy,” “worst experience”

1. Correlation with Time (if timestamp is available):

Trends over time were plotted to observe spikes in positive/negative sentiments.

Example: Increase in negative sentiment during a product recall or political event.

1. Location-based Sentiment (if location data is available):

Heatmaps or region-wise breakdowns showed geographical sentiment variations.

# 7. Feature Engineering

### **Feature Engineering:**

Feature engineering transforms raw text data into meaningful representations that can be used by machine learning models. In this project, the following features were engineered from the text data:

#### **1. Text-based Features:**

* **Bag of Words (BoW):**  
  Converts text into a matrix of token counts, useful for simpler models like Naive Bayes or Logistic Regression.
* **TF-IDF (Term Frequency–Inverse Document Frequency):**  
  Weights words by their importance across the corpus. Reduces the impact of common words.
* **N-grams:**  
  Inclusion of bigrams or trigrams to capture short phrases (e.g., “not good” vs. “good”).

#### **2. Lexical Features:**

* **Text Length:**  
  Number of words or characters in each text sample.
* **Word Count:**  
  Total number of words, used to measure verbosity or expressiveness.
* **Average Word Length:**  
  Indicates the complexity of language used.

#### **3. Sentiment Lexicon Features:**

* **Polarity Score:**  
  Using tools like **TextBlob** or **VADER** to assign a sentiment polarity score (-1 to 1).
* **Subjectivity Score:**  
  Measures how subjective or opinionated the text is.

#### **4. Syntactic Features:**

* **Part-of-Speech Tags (POS):**  
  Frequency of nouns, verbs, adjectives, etc., to detect emotionally charged language.
* **Punctuation Count:**  
  Frequency of exclamation marks, question marks—often correlated with emotional expression.
* **Capitalization Count:**  
  Number of words in all caps (e.g., “LOVE,” “HATE”) can signal stronger emotions.

#### **5. Word Embeddings (for Deep Learning):**

* **Pre-trained Embeddings:**  
  Word2Vec, GloVe, or FastText vectors to capture semantic relationships between words.
* **Contextual Embeddings:**  
  BERT, RoBERTa, or other transformer-based embeddings that account for context in sentences.

#### **6. Custom Features (Optional):**

* **Emoji Count:**  
  Emojis can convey sentiment (e.g., “:)” for positive, “:(” for negative).
* **Negation Handling:**  
  Special tagging or transformation when words are preceded by "not" (e.g., “not good” = negative).

# 8. Model Building

Here’s a detailed **Model Building** section for your sentiment analysis project:

### **Model Building:**

The goal of this phase is to train machine learning models that can accurately classify sentiment (positive, negative, or neutral) based on textual input. Two main modeling approaches were explored: classical machine learning and deep learning.

#### **1. Data Splitting:**

* **Training Set:** 70% of the dataset
* **Validation Set:** 15% (for tuning)
* **Test Set:** 15% (for final evaluation)
* Stratified sampling was used to maintain sentiment label balance.

#### **2. Classical Machine Learning Models:**

Text was vectorized using TF-IDF and then fed into traditional classifiers:

* **Logistic Regression**
* **Naive Bayes (Multinomial)**
* **Support Vector Machine (SVM)**
* **Random Forest**

**Hyperparameters** were tuned using Grid Search or Randomized Search.

#### **3. Deep Learning Models:**

* **LSTM (Long Short-Term Memory):**
  + Inputs were word embeddings

# 9. Visualization of Results & Model Insights

### **Visualization of Results & Model Insights**

Effective visualization helps to understand model behavior, performance, and the underlying patterns in the data. The following visual tools and insights were derived:

#### **1. Confusion Matrix**

* **Purpose:** To visualize true vs. predicted sentiment classes.
* **Insight:**
  + High accuracy in predicting positive and negative sentiments.
  + Some confusion between neutral and negative classes.
* **Tool:** seaborn.heatmap() in Python.

#### **2. Classification Report (Bar Plot)**

* **Metrics:** Precision, Recall, F1-Score per class.
* **Insight:**
  + Positive class had the highest F1-score.
  + Neutral class

# 10. Tools and Technologies

### **Tools and Technologies Used**

The project leveraged a range of tools, libraries, and platforms across data processing, modeling, and visualization stages.

#### **1. Programming Language:**

* **Python** – Primary language for data analysis, machine learning, and NLP.

#### **2. Data Collection & Processing:**

* **Tweepy / snscrape** – For collecting data from Twitter.
* **Pandas** – For data manipulation and analysis.
* **NumPy** – For numerical computations.
* **NLTK / spaCy / re** – For natural language preprocessing (tokenization, stopword removal, POS tagging, etc.).

#### **3. Feature Engineering:**

* **Scikit-learn (sklearn)** – For TF-IDF, feature extraction, and classical ML models.
* **TextBlob / VADER** – For lexicon-based sentiment scores.
* **Gensim** – For Word2Vec embeddings.
* **Hugging Face Transformers** – For contextual embeddings and BERT-based models.

#### **4. Model Building:**

* **Scikit-learn** – For classical models (Logistic Regression, SVM, Random Forest).
* **TensorFlow / Keras** – For building LSTM and CNN models.
* **PyTorch** – For fine-tuning transformer models like BERT.

#### **5. Visualization:**

* **Matplotlib / Seaborn** – For plotting sentiment distribution, confusion matrix, metrics.
* **WordCloud** – For visualizing frequent words.
* **Plotly** – For interactive visualizations.
* **SHAP / ELI5** – For model interpretability.

#### **6. Deployment (Optional/Advanced):**

* \*\*Flask

# 11. Team Members and Contributions

**R EZHILARASI**

**V INDHUJA**

**MOHANA PRIYA**

**DEEPAK**

**Clearly mention who worked on:**

* **Data cleaning- EZHILARASI**
* EDA-V INDHUJA

○ Feature engineering-MOHANA PROYA

○ Model development-DEEPAK

○ Documentation and reporting-R EZHILARASI