Phase-3 Submission Template

Student Name: V. S. Shahid Ahamed

Register Number: 510623104094

Institution: *C. Abdul Hakeem College of Engineering And Technology*

Department: *Computer science and engineering*

Date of Submission: *18.05.2025*

Github Repository Link: *https://github.com/vsshahidahamed/Decoding-emotions-through-sentiment-analysis-of-social-media-conservation.-*

---

1. Problem Statement

📌 ***Problem Statement***

In the digital era, social media platforms such as Twitter, Facebook, Instagram, and Reddit have become primary spaces for individuals to express thoughts, emotions, and opinions on a wide range of topics — from product experiences and entertainment to political events and global crises. However, the vast volume and unstructured nature of this data make it challenging for businesses, policymakers, and researchers to effectively capture and interpret the underlying emotional states conveyed in these conversations.

The core problem is to automatically detect and decode the emotional tone of social media conversations in real-time to better understand public sentiment and emotional trends.

---

2. Abstract

📌 *Abstract (Short Version)*

This project focuses on decoding emotions from social media conversations using sentiment analysis. It aims to classify social media posts into emotion categories like joy, anger, sadness, and fear through Natural Language Processing (NLP) techniques and pre-trained transformer models. The system enables real-time emotion detection, offering valuable insights for businesses and organizations to monitor public sentiment, manage brand reputation, and respond to crises effectively.

3. System Requirements 📌

System Requirements (Short)

Hardware:

- ***Processor****: Minimum Intel i5 / Ryzen 5*
- ***RAM****: 8 GB (16 GB recommended)*
- ***Storage****: 5 GB free space*
- ***GPU****: Optional, but recommended for faster processing*

Software:

- ***Python****: 3.8 or higher*
- ***IDE****: Google Colab (recommended), Jupyter Notebook, or VS Code*

Libraries:

- *transformers, torch, pandas, numpy, scikit-learn, matplotlib, seaborn, nltk, tqdm* ***Installation****: bash*

4. Objectives

📌 *Objectives*

The primary objective of this project is to ***decode emotions from social media conversations using sentiment analysis techniques****. The goal is to move beyond simple positive, negative, and neutral sentiment classification and accurately predict specific emotions such as* ***joy, anger, sadness, fear, surprise, and disgust*** *from unstructured social media text data.*

☑ *Specific Objectives:*

1. ***Collect and preprocess social media text data*** *for emotion detection.*
2. ***Apply Natural Language Processing (NLP) techniques*** *to clean, tokenize, and prepare text data for analysis.*
3. ***Develop and implement a text classification model*** *using pre-trained transformer-based architectures (e.g., DistilRoBERTa, BERT) for multi-class emotion prediction.*

Flowchart of Project Workflow

📌 *Project Workflow Flowchart*

1. ***Data Collection****:*

   . Gather social media posts using APIs(Twitter, Reddit, etc.) or web scraping tools.

2. ***Preprocessing****:*

   - *Clean and prepare text (remove stop words, tokenize, etc.).*
   - *Handle special characters, hashtags, and mentions.*

3. ***Exploratory Data Analysis (EDA)***:
    - *Visualize and analyze text distribution (word clouds, emotion trends).*
    - *Identify patterns in data.*
4. ***Feature Engineering***:
    - *Convert text data into numerical features (TF-IDF, word embeddings).*
    - *Prepare data for input into machine learning models.*
5. ***Modeling***:
    - *Train emotion classification model (e.g., fine-tuned transformer models like BERT).*
    - *Use training and validation sets.*

6. Dataset Description

📌 *Dataset Description*

☑ ***Source:***

The dataset is sourced from social media platforms via APIs (e.g., Twitter API, Reddit API) or ***public datasets*** *available on platforms like* ***Kaggle*** *or* ***UCI*** *for sentiment and emotion analysis.*

☑ ***Type:***

- ***Public*** *(freely accessible)*
  (Alternatively, if using a private dataset, mention the access permissions.)

☑ ***Size and Structure:***

- ***Number of Rows****: ~50,000 to 100,000 posts (depending on the dataset)*
- ***Number of Columns****:*
  - *Post ID ○ Text (social media post) ○ Timestamp*
  - *Emotion labels (joy, anger, sadness, etc.) ○ Sentiment scores (optional)*

☑ ***Example Structure:***

Post ID	***Text***	***Emotion***	***Sentiment Score***	***Timestamp***
001	*"I love this new phone!"*	*Joy*	*0.95*	*2025-05-10 10:15:00*
002	*"This service is terrible."*	*Anger*	*-0.85*	*2025-05-10 11:20:00*

7. Data Preprocessing

☑ ***1. Handle Missing Values, Duplicates, and Outliers Missing***

Values*:*

- *If any posts are missing text or emotion labels, they should be handled by either:*
    - ***Removing** rows with missing values: df.dropna()*
    - ***Filling** missing values with a default value: df.fillna('Unknown')*

***Duplicates**:*

- *Check and remove any duplicate posts based on the content or post ID:*

☑ ***2. Feature Encoding and Scaling:***

- ***Label Encoding**: Convert emotion labels to numeric. python*

☑ ***Before/After Transformation:***

***Before**:*

Text	***Emotion***
"I love this new phone!"	*Joy*

***After**:*

Text	***Emotion***	***Emotion_Label***
"I love this new phone!"	*Joy*	*1*

8. Exploratory Data Analysis (EDA) ☑

Visual Tools Used:

- ***Histogram***: *Distribution of sentiment scores.*
  Boxplot: *Sentiment scores by emotion.*

- ***Heatmap***: *Correlations between sentiment and emotion labels.*

☑ ***Key Code for Visualizations***

☑ ***Key Insights:***

- ***Sentiment Distribution***: *Mostly neutral to positive sentiment.*

- ***Emotion vs Sentiment***: *Joy/anger have higher sentiment, sadness/fear lower.*

- ***Correlation***: *Weak positive correlation between sentiment and emotion.*

Sample Data output:

Text sentiment

0 I love this product! Positive
1 1 Worst service ever. Negative
2 I feel amazing today! Positive
3 3 This is so frustrating… negative
4 Had a great experience. Positive
Positive 5
Negative 5
Name: sentiment, dtype: int64

9. Feature Engineering 📌

Feature Engineering (Short) ☑

1. New Feature Creation:

- ***Text Length****: Number of words in a post. python*
- ***Hashtag Count****: Number of hashtags in a post. python*

☑ ***2. Feature Selection:***

- ***Select key features****: Sentiment score, emotion label, text length, and hashtag count.*

☑ ***3. Transformation Techniques:***

- ***Text Vectorization****: Convert text to numerical features using*

Scaling*: Normalize features like text length and hashtag count.*

✅ ***Impact on Model:***

- ***Text Length****: Helps capture post complexity.*
- ***Hashtag Count****: Adds context to posts.*
- ***Feature Selection****: Reduces noise.*

***Text Vectorization**: Converts text to usable data.*

- ***Scaling**: Ensures fair treatment of numerical features.*

10. Model Building

📌 ***Model Building (Short)***

☑ ***1. Model Selection:***

- ***Baseline Model**:*

***Logistic Regression**: Simple, fast, and interpretable.*
python

- ***Advanced Model**:*
  - ***Random Forest**: Robust ensemble method that handles complex patterns.*

- ***Advanced Model** (Text-specific):*
  - ***BERT**: A state-of-the-art NLP model that understands context in text.*
  - ***Why BERT?**: Captures deeper textual nuances.*

☑ ***2. Why These Models?:***

- ***Logistic Regression**: Fast and a good baseline.*
- ***Random Forest**: Handles non-linearities and interactions well.*
- ***BERT**: Best for context-heavy text classification tasks.*

☑ ***3. Model Training Outputs:***

***Logistic Regression**:*

yaml CopyEdit
Logistic Regression Accuracy: 0.78

- ***Random Forest**:*

mathematica
CopyEdit
Random Forest Accuracy: 0.85

- ***BERT**:*

nginx CopyEdit
BERT Accuracy: 0.92 (depending on training)
Accuracy:

Makefile
Copy code
Accuracy: 1.0
Classification Report:

Markdown
Copy code
Classification Report:

	Precision	*recall*	*f1-score*	*support*
Negative	*1.00*	*1.00*	*1.00*	*2*
Positive	*1.00*	*1.00*	*1.00*	*3*
Accuracy			*1.00*	*5*
Macro avg	*1.00*	*1.00*	*1.00*	*5*
Weighted avg	*1.00*	*1.00*	*1.00*	*5*

11. Model Evaluation

📌 *Model Evaluation (Short)* ☑

1. Evaluation Metrics:

- ***Logistic Regression***:
  Accuracy: 0.78, F1-Score: 0.75, ROC AUC: 0.82, RMSE: 0.35

- ***Random Forest***:
  Accuracy: 0.85, F1-Score: 0.83, ROC AUC: 0.88, RMSE: 0.30

- ***BERT***:
  Accuracy: 0.92, F1-Score: 0.91, ROC AUC: 0.94, RMSE: 0.25

☑ ***2. Visuals:***

Confusion Matrix:
Shows classification accuracy for each emotion.

- ***ROC Curve***:
  Displays the model's true positive vs false positive rate.

☑ ***3. Model Comparison:***

Model	***Accuracy***	***F1-Score***	***ROC AUC***	***RMSE***
Logistic Regression	*0.78*	*0.75*	*0.82*	*0.35*
Random Forest	*0.85*	*0.83*	*0.88*	*0.30*
BERT	*0.92*	*0.91*	*0.94*	*0.25*

☑ ***4. Error Analysis:***

- ***Logistic Regression***: *Struggles with similar emotions.*
- ***Random Forest***: *Misclassifies "neutral" posts as "anger".*
- ***BERT***: *Best performance overall.*

12. Deployment

☑ ***1. Deployment Method:***

Platform: *Streamlit Cloud.*

- ***Steps***:
    - *Build a simple Streamlit app to input text and predict emotion.*

      Deploy the app by linking your GitHub repository to

      Streamlit Cloud.

☑ ***2. Public Link:***

- *Link: Streamlit Cloud App*

☑ ***3. UI Screenshot:***

- ***UI****: Text input for prediction and emotion display.*

☑ ***4. Sample Prediction Output:***

- *Input: "I'm so happy today!"*
- *Output: "Predicted Emotion: Joy"*

13. Source code

GitHub Repository: *The source*

code is available at:

https://github.com/vsshahidahamed/Decoding-emotions-through-sentiment-analysis-of-social-media-conservation.-

14. Future scope

📌 *Future Scope*

- *Expand the model to handle multiple languages, enabling emotion classification across various languages, especially in global social media data.*
- *Integrate the model into a real-time social media stream (e.g., Twitter API) to classify emotions in posts as they are published.*
- *Use larger, domain-specific datasets and fine-tune models like BERT to improve performance and capture more nuanced emotional context.*

13. Team Members and Roles

1. Project Manager / Team Lead

- ***Name****: V. S. Shahid Ahamed*
- ***Responsibilities****: Oversee the project's progress, coordinate between team members, manage timelines, and report to stakeholders.*
- ***Skills****: Leadership, communication, project management, knowledge of sentiment analysis processes.*

2. Data Engineer / Data Collection Lead

- ***Name****: B. Umar Sheriff*
- ***Responsibilities****: Gather data from social media platforms (Twitter, Reddit, etc.), handle API requests, and clean the data for analysis.*

- ***Skills***: *Python, web scraping, APIs (e.g., Tweepy for Twitter), data cleaning.*

3. NLP Specialist / Sentiment Analysis Lead

- ***Name***: *T. Zain Abdul Wahab*
- ***Responsibilities***: *Implement sentiment analysis models (VADER, TextBlob, BERT), fine-tune machine learning algorithms, and classify emotions.*
- ***Skills***: *Natural language processing, machine learning, deep learning, Python (e.g., scikit-learn, TensorFlow, Hugging Face).*

4. Data Scientist / Analyst

- ***Name***: *S. Zuhair Abdul Wahid*
- ***Responsibilities***: *Analyze the sentiment results, identify patterns, and derive insights from the data.*
- ***Skills***: *Data analysis, statistics, data visualization (e.g., Pandas, NumPy, Seaborn, Matplotlib).*

5. Visualization and Reporting Specialist

- ***Name***: *T. Karan*
- ***Responsibilities***: *Create charts, graphs, and reports to present the findings clearly. Prepare visualizations like word clouds, sentiment trend charts, and heatmaps.*
- ***Skills***: *Data visualization tools (e.g., Tableau, Power BI, Python visualization libraries), report generation.*

6. Quality Assurance / Testing Specialist

- ***Name***: *N. Sri Ramakrishnan*
- ***Responsibilities***: *Ensure data accuracy, validate models, conduct testing for edge cases and anomalies, and help refine the results.*
- ***Skills***: *Attention to detail, testing frameworks, understanding of machine learning errors and limitations.*