Phase-1

Student Name: Krishnakumar k

Register Number: 712523121011

Institution: PPG INSTITUTE OF TECHNOLOGY

Department: BIOMEDICAL ENGINEERING

Date of Submission:

Decoding emotions through sentiment analysis of social media conversations

1.Problem Statement:

In the digital age, social media platforms have become powerful mirrors of human emotion, capturing raw, real-time sentiments expressed across the globe. This project explores how sentiment analysis—a natural language processing (NLP) technique—can decode these emotions by analyzing conversations on platforms like Twitter, Facebook, Instagram, and whatsapp.

2.Objectives of the Project:

- To identify emotional patterns and trends in social media conversations.
- To classify sentiments (positive, negative, neutral) and deeper emotions (joy, anger, fear, etc.) using machine learning and deep learning models.
- To assess the impact of external events (e.g., pandemics, elections, celebrity news) on collective emotional expression.
- To understand the role of language nuances, slang, emojis, and sarcasm in sentiment detection.

3.Scope of the Project:

1. Data Collection:

   - Collect social media data

2. Text Preprocessing:

   - Clean and normalize text

3.Sentiment and Emotion Analysis:

   - Apply NLP techniques and machine learning models (e.g., VADER, BERT, RoBERTa) to classify sentiments.
   - Perform multi-class emotion classification using labeled datasets (e.g., EmotionLex, GoEmotions).

4.Visualization and Interpretation:

- Visualize sentiment/emotion trends using graphs, charts, and word clouds.
- Analyze emotional responses to specific events, topics, or time periods.

5.Evaluation:

- Assess model performance using metrics like accuracy, precision, recall, and F1-score.
- Compare results across different models and approaches.

4.Data Sources:

1. Twitter API (X API)

- Why use it: Rich in real-time, short-form user-generated content with hashtags, emojis, and mentions that carry emotional context.
- Access method: Twitter/X API (academic researchers get more access).

2. Reddit API

- Why use it: Highly conversational, topic-focused discussions—good for extracting nuanced sentiments and emotions.
- Access method: Reddit API (via Pushshift or PRAW in Python).

4. Facebook or Instagram (Limited access)

- Why use it: Strong emotional expression in comment sections and posts.
- Challenges: Very restricted due to privacy and data-sharing regulations. Usually accessible only through business/partner APIs.
- Alternative: Use public pages or influencer accounts if allowed.

5.High-Level Methodology:

1. Define Objective and Scope

- Goal: Detect and classify human emotions from social media text (e.g., joy, anger, sadness, fear, etc.).
- Platforms: Choose one or more (e.g., Twitter, Reddit, YouTube).
- Emotion model: Select a framework (e.g., Plutchik's wheel, Ekman's 6 basic emotions, or use multi-label emotion detection).

2. Data Collection

- APIs/Tools Used: Twitter API, Reddit API, YouTube API, or datasets from Kaggle.
- Keywords/Hashtags: Use emotion-related keywords or trending topics.
- Metadata: Collect timestamp, user ID (anonymized), likes, etc.
- Storage Format: Store data in JSON/CSV for preprocessing.

3. Data Preprocessing

- Steps:
    - Text cleaning (remove URLs, mentions, hashtags, emojis, stopwords).
    - Normalization (lowercase, lemmatization/stemming).
    - Language filtering (if multilingual, detect and select target language).
    - Tokenization (for input to models).

4. Emotion Annotation / Labeling

- If using pre-labeled data: No annotation needed (e.g., Sentiment140, GoEmotions).
- If unlabeled:
    - Manual annotation (small datasets).
    - Distant supervision (e.g., emojis or hashtags as emotion proxies).
    - Use pretrained models (zero-shot or few-shot emotion classifiers) for weak labeling.

5. Feature Extraction

- Traditional NLP:
    - Bag-of-Words, TF-IDF
- Deep Learning / Transformers:
    - Word embeddings (Word2Vec, GloVe)
    - Contextual embeddings (BERT, RoBERTa, DistilBERT)
    - Sentence embeddings (SBERT, USE)

6. Model Training

- Traditional ML:
    - Logistic Regression, SVM, Random Forest
- Deep Learning:
    - LSTM, BiLSTM, GRU
- Transformer-based:
    - Fine-tune BERT/RoBERTa for emotion classification

7. Evaluation Metrics

- Accuracy
- Precision / Recall / F1-score
- Confusion Matrix
- AUC-ROC (if binary/multilabel)

8. Emotion Analysis & Visualization

- Time series of emotions over an event
- Emotion distribution per topic/region
- Word clouds by emotion class
- Sentiment vs Emotion heatmaps

9. Insights & Interpretation

- Discuss how different emotions are distributed.
- Link findings to real-world events, social behavior, or platform characteristics.
- Identify patterns or anomalies.

10. Conclusion and Future Work

- Summarize key insights.
- Limitations (e.g., sarcasm detection, platform bias, demographic bias).
- Future scope (e.g., multimodal emotion detection, real-time emotion dashboards)

6.Tools and Technologies:

1. Data Collection

- Twitter API / X API
- Reddit API (PRAW / Pushshift)
    - BeautifulSoup, Selenium, Scrapy
- Datasets:
    - Kaggle, Hugging Face Datasets
- Tools for Scheduling/Streaming:
    - Tweepy, snscrape, Apache Kafka (for real-time streams)

2. Data Storage and Handling

- Databases:
    - MongoDB (NoSQL for JSON data), MySQL / PostgreSQL
- File Formats:
    - CSV, JSON, Parquet
- Cloud Storage (optional):
    - AWS S3, Google Cloud Storage

3. Preprocessing & Text Cleaning

- Languages/Libraries:
    - Python (main language)
    - NLTK, spaCy (for tokenization, stopword removal, lemmatization)
    - re (regex for pattern cleaning)

- emoji, langdetect (for emoji handling and language detection)

4. Emotion Annotation / Labeling

- Manual Tools:
  - Label Studio, Prodigy, Doccano
- Automatic Tools:
  - Pretrained Models from Hugging Face Transformers (like cardiffnlp/twitter-roberta-base-emotion)
  - Emotion lexicons (e.g., NRC Emotion Lexicon)

5. Feature Engineering

- Traditional:
  - scikit-learn (TF-IDF, BoW, count vectorizer)
- Embeddings & Transformers:
  - gensim (Word2Vec)
  - spaCy (built-in embeddings)
  - transformers (BERT, RoBERTa, DistilBERT from Hugging Face)
  - sentence-transformers (for SBERT)

6. Modeling / Classification

- Traditional ML:
  - scikit-learn (SVM, Naive Bayes, Logistic Regression)
- Deep Learning:
  - TensorFlow, Keras, PyTorch
- Transformers:
  - transformers library (Hugging Face) for fine-tuning models like BERT, RoBERTa

7. Evaluation

- scikit-learn (metrics like accuracy, precision, recall, F1-score)
- matplotlib, seaborn, plotly (confusion matrices, AUC curves)

8. Visualization & Insights

- Visualization Libraries:
    - matplotlib, seaborn, plotly, wordcloud
- Dashboards (optional):
    - Streamlit, Dash, Tableau, or Power BI
- Text analysis tools:
    - LDA topic modeling (via gensim)

9. Project Management and Collaboration

- Version Control: Git, GitHub
- Notebooks: Jupyter Notebook, Google Colab
- Workflow Tools (optional): MLflow, Airflow (for pipelines

7.Team Members and Roles:

TEAM MEMBERS	ROLES
Krishnakumar K	Objectives
Justin Jerold A	High-Level Methodology
Manu M	Tools and Technologies
Dhinesh kumar M	Scope, Data Sources