Decoding Emotions through Sentiment Analysis of Social Media Conversations

Student name : P.Elango

Register no : 620623104011

Institution : Ganesh College of engineering

Department : B.E [ computer science and Engineering ]

Date of Submission : 07.05.2025

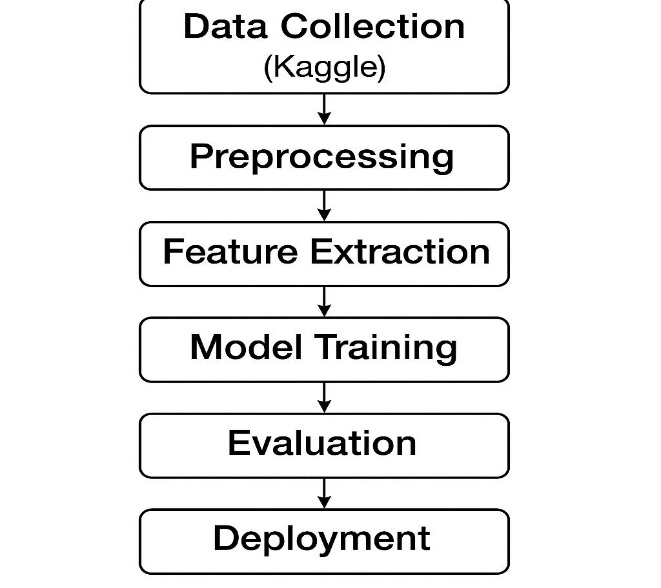
# 1. Problem Statement

Understanding public sentiment and emotions expressed in social media conversations is crucial for domains like marketing, public policy, mental health monitoring, and crisis management. This project aims to develop a sentiment analysis model that classifies user-generated text (e.g., tweets or posts) into distinct emotional categories (e.g., joy, anger, sadness, fear).  
  
The goal is to decode emotional tone from text data using natural language processing (NLP) techniques. This is a multi-class classification problem since each text input is assigned to one of several emotion labels.

# 2. Project Objectives

- Build an accurate NLP model to classify text into emotional categories.  
- Analyze patterns in emotional expression across different demographic groups or topics.  
- Visualize sentiment trends over time or by keyword.  
- Integrate the model into a web interface (e.g., Gradio) for interactive testing.

# 3. Flowchart of the Project Workflow



# 4. Data Description

- Dataset Name: Emotion Dataset (e.g., from Kaggle, CrowdFlower, or SemEval)  
- Source: Public social media or curated sentiment datasets.  
- Type: Textual, unstructured data.  
- Records & Features: e.g., 10,000+ text entries, with emotion labels  
- Target Variable: Emotion category (e.g., joy, anger, sadness, fear, etc.)  
- Attributes: Text content, metadata (optional: timestamp, user info, etc.)

# 5. Data Preprocessing

- Cleaned text (removed URLs, hashtags, emojis, punctuation).  
- Tokenized and lowercased text.  
- Removed stopwords.  
- Applied lemmatization/stemming.  
- Handled class imbalance (e.g., via SMOTE or class weights).

# 6. Exploratory Data Analysis (EDA)

- Frequency distribution of emotion classes.  
- Word clouds for each emotion.  
- Time-based emotion trends (if time info available).  
- Word frequency histograms and n-grams.

# 7. Feature Engineering

- Text vectorization using TF-IDF, Word2Vec, or BERT embeddings.  
- Engineered features like sentiment polarity, subjectivity (TextBlob).  
- Dimensionality reduction using PCA or t-SNE (optional).

# 8. Model Building

- Algorithms Used:  
 - Logistic Regression / Naive Bayes (baseline)  
 - LSTM / Transformer models (e.g., BERT) for advanced classification  
- Evaluation Metrics:  
 - Accuracy, Precision, Recall, F1-score, Confusion Matrix

# 9. Visualization of Results & Model Insights

- Confusion matrix for each model.  
- ROC/PR curves (if binary or per-class).  
- Misclassified text samples.  
- Real-time testing via Gradio interface.

# 10. Tools and Technologies Used

- Programming Language: Python 3  
- Environment: Google Colab / Jupyter  
- Libraries: pandas, numpy, matplotlib, seaborn, scikit-learn, NLTK/spaCy, transformers (Hugging Face), Gradio

# 11. Team Members and Contributions

(Replace with actual names and contributions)  
P.Elango - Data Collection & Cleaning:   
S.geenu - Exploratory Data Analysis:   
G.dhanush - Model Development & Evaluation:   
V.Devaraj - Frontend Integration (Gradio):   
E.Gokul - Reporting & Documentation: