**Project Objective**

To refine input prompts that effectively guide generative AI models (e.g., ChatGPT, Gemini, Claude) to produce outputs with a controlled emotional tone using sentiment analysis.

**Dataset**

The dataset used is:

* **File Name**: training.1600000.processed.noemoticon.csv
* **Source**: Twitter Sentiment Analysis Dataset
* **Description**: Contains 1.6 million tweets labeled with sentiment (0 = negative, 4 = positive)
* **Fields**:  
  sentiment, id, date, query, user, text

This dataset is used to train and validate sentiment analysis models and evaluate the sentiment of AI-generated responses.

**Libraries Used**

* pandas, numpy — data handling
* matplotlib, seaborn — visualization (optional)
* textblob, vaderSentiment, transformers — sentiment analysis
* openai, google.generativeai, anthropic — AI response generation
* sklearn — for evaluation metrics
* streamlit or gradio — UI (optional)

**Algorithm**

The following steps define the workflow of the system:

**Step 1: Prompt Design**

* **Input**: Target sentiment (positive/neutral/negative/empathetic/etc.)
* **Output**: Prompt variations aiming to elicit the target sentiment.
* **Tools**: Manual design + templates + GPT assistance.

**Step 2: Response Generation**

* **Input**: Designed prompt
* **Process**: Sent to LLM APIs (ChatGPT, Gemini, Claude)
* **Output**: AI-generated response text

**Step 3: Sentiment Analysis**

* **Input**: AI-generated response
* **Process**:
  + TextBlob: Polarity score
  + VADER: Compound score
  + BERT: Classification (positive/neutral/negative)
* **Output**: Sentiment score or label per model

**Step 4: Evaluation**

* **Input**: Desired sentiment vs. actual sentiment
* **Process**:
  + Compute match ratio
  + Identify mismatches
* **Condition**: If sentiment alignment < threshold, mark for refinement

**Step 5: Prompt Refinement**

* **Input**: Poorly performing prompts
* **Process**: Rephrase using templates, synonyms, or GPT prompt suggestions
* **Loop**: Return to Step 2 for re-evaluation

**Optional: Model Training**

* **Input**: Uploaded sentiment dataset
* **Process**:
  + Fine-tune BERT or train a classifier to predict sentiment
* **Output**: Custom sentiment model for more robust evaluation

**Control Flow:**

START

**↓**

[User defines target sentiment]

**↓**

[Generate initial prompt]

**↓**

[AI model generates response]

**↓**

[Run sentiment analysis]

**↓**

IF response sentiment ≠ target:

↳ Refine prompt

↳ Loop back to generation

ELSE:

↳ Save prompt template

**↓**

[Evaluate across multiple samples/models]

**↓**

END