Decoding Emotions through Sentiment Analysis of Social Media Conversations

# 1. Why This Project is Chosen & Its Importance

Why: Social media is a rich source of public emotion and opinion. Understanding sentiments helps businesses, governments, and researchers respond effectively.

Importance:

* - Track public mood on events/policies.
* - Improve customer service and brand management.
* - Detect trends, threats, or opportunities in real time.

# 2. Objectives of the Project

* - Extract and analyze emotional tone from social media posts.
* - Classify conversations into sentiments/emotions (e.g., happy, sad, angry).
* - Build a predictive model to automate sentiment detection.
* - Visualize trends and insights.

# 3. Scope of the Project

* - Focused on one or more platforms (e.g., Twitter, Reddit).
* - Language: Primarily English.
* - Text-based content only (no images/videos).
* - Real-time or historical analysis.

# 4. Data Sources

* - Twitter API (via Tweepy or Twitter Developer Portal)
* - Reddit API (PRAW)
* - Open Datasets (Kaggle, CrowdFlower, Sentiment140)

# 5. High-Level Methodology

a. Data Collection:

* - Use APIs to extract social media text (tweets, posts, comments).

b. Data Cleaning:

* - Remove stopwords, URLs, hashtags, mentions, emojis.
* - Normalize text (lowercase, stemming/lemmatization).

c. Exploratory Data Analysis (EDA):

* - Word clouds, frequency plots, sentiment distribution.
* - Time-based trends and correlations.

d. Feature Engineering:

* - Text vectorization (TF-IDF, Word2Vec, BERT embeddings).
* - Add meta-features (post length, number of emojis, etc.).

e. Model Building:

* - Classification models: Logistic Regression, Naive Bayes, SVM, LSTM, or Transformers (BERT).

f. Model Evaluation:

* - Accuracy, Precision, Recall, F1-score, Confusion Matrix.

g. Visualization & Interpretation:

* - Dashboards (Plotly, Seaborn, Matplotlib).
* - Sentiment over time, word importance, etc.

h. Deployment (Optional):

* - Build web app (Streamlit/Flask).
* - Host on Heroku, AWS, or Hugging Face Spaces.

# 6. Tools and Technologies

* - Programming Language: Python
* - Notebook/IDE: Jupyter Notebook, VS Code
* - Libraries:

- NLP: NLTK, spaCy, TextBlob, Transformers

- ML: scikit-learn, XGBoost, TensorFlow/Keras, PyTorch

- Data: pandas, numpy

- Visualization: matplotlib, seaborn, plotly

* - Deployment (Optional): Streamlit, Flask, Heroku, Hugging Face