Github Link: [**https://github.com/lokeshsivamani/sentiment-analysis-of-social-media-conversation.git**](https://github.com/Thulasimathi26/Data-Science.git)

**PROJECT TITLE: DECODING EMOTIONS THROUGH SENTIMENT ANALYSIS OF SOCIAL MEDIA CONVERSATIONS**

**PHASE-2**

# Problem Statement

# In the digital era, social media platforms have become primary outlets for people to express their emotions, opinions, and reactions. However, due to the massive volume and unstructured nature of this data, manually analyzing emotional trends is highly challenging.

# This project focuses on building a sentiment and emotion classification system using Natural Language Processing (NLP) techniques. The system will classify sentiments (positive, negative, neutral) and deeper emotions (joy, anger, sadness, fear) from textual conversations on platforms like Twitter and Reddit.

# The significance lies in applications across mental health monitoring, brand perception analysis, and public opinion tracking, helping bridge human emotions with machine understanding

# 2.Project Objectives

- Develop an NLP-based system to classify sentiments and emotions from social media text.

- Collect and preprocess data (Twitter, Reddit, Kaggle datasets).

- Perform exploratory data analysis (EDA) on emotional patterns.

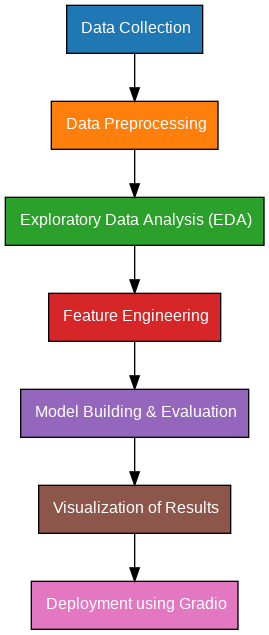
- Build and train models using Logistic Regression, LSTM, BERT.

- Evaluate model performance using precision, recall, F1-score.

- Deploy the model using Streamlit or Flask for real-time prediction.

- Visualize emotional trends and insights.

# Flowchart of the Project Workflow



# Data Description

**- Dataset Names: Sentiment140, Emotion Dataset (Twitter), GoEmotions (Reddit)**

**- Sources: Kaggle, Twitter API, Google Research**

**- Type of Data: Textual data (tweets, comments)**

**- Records and Features: Over 1 million records labeled with sentiment/emotion**

**- Target Variable: Sentiment (positive, negative, neutral) and emotions (joy, anger, sadness, fear,etc.)**

**- Static and Dynamic Datasets: Static from Kaggle, dynamic via Twitter/Reddit APIs**

**Dataset Links:**

**- Sentiment140:** [**https://www.kaggle.com/datasets/kazanova/sentiment140**](https://www.kaggle.com/datasets/kazanova/sentiment140)

**- GoEmotions:**  [**https://www.kaggle.com/datasets/debarshichanda/goemotions**](%20https:/www.kaggle.com/datasets/debarshichanda/goemotions)

# Data Preprocessing

# - Removed URLs, mentions, hashtags, emojis, and special characters.

# - Text normalization: Lowercasing, stemming, lemmatization.

# - Tokenization and stopword removal.

# - Applied TF-IDF vectorization and BERT embeddings for text representation.

# 6.Exploratory Data Analysis (EDA)

# - Univariate Analysis:

# - Word clouds for frequent terms.

# - Distribution plots for sentiment classes.

# - Bivariate & Multivariate Analysis:

# - Correlation heatmaps among emotional categories.

# - Sentiment trend graphs over time.

# Key Insights:

# - Positive sentiments dominated certain domains.

# - Fear and sadness had spikes during global events.

# 7.Feature Engineering

# - Extracted n-grams (bigrams, trigrams).

# - Word embeddings using Word2Vec, GloVe, and BERT.

# - Text vectorization with TF-IDF

# 8.Model Building

# - Algorithms Used:

# - Logistic Regression: Baseline model

# - LSTM: For sequence modeling- BERT Fine-tuning: For high accuracy

# - Model Selection Rationale:

# - Logistic Regression: Quick baseline.

# - LSTM: Captures sequence and context.

# - BERT: State-of-the-art transformer for NLP tasks.

# - Train-Test Split: - 80% training, 20% testing

# - Evaluation Metrics:

# - Accuracy, Precision, Recall, F1-Score

# - Confusion Matrix

# - ROC-AUC (for binary/multiclass classification)

# 9.Visualization of Results & Model Insights

# - Confusion matrices and classification reports.

# - Feature importance visualization (via attention maps for BERT).

# - Trend visualization using time-series plots.

# 10.Tools and Technologies Used

● **Programming Language**: Python 3

● **Notebook Environment**: Google Colab

● **Key Libraries**:

○ pandas, numpy for data handling

○ matplotlib, seaborn, plotly for visualizations

○ scikit-learn for preprocessing and modeling

○ Gradio for interface deployment

# Team Members and Contributions

|  |  |
| --- | --- |
| **Team member** | ***Contribution*** |
| Lokesh.S | *Data cleaning* |
| Vannamathi.K | *EDA* |
| Dhanasekaran T | *Feature engineering* |
| Venkades.T | *Model development* |
| Vijayalakshmi.S | *Documentation and reporting* |