Github Link: https://github.com/Annamalai4536/phase-2

Project Title: Decoding emotions through sentiment analysis of social media conversations

PHASE-2

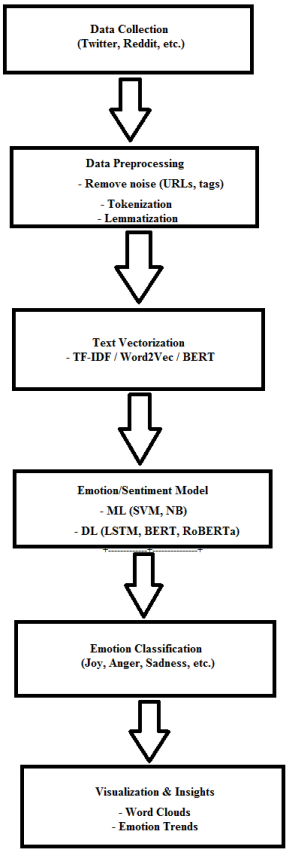
- **Problem Statement**

The problem statement is to develop a reliable method for accurately decoding emotions expressed in social media conversations through sentiment analysis, addressing challenges such as sarcasm, irony, and cultural nuances. This will enable better understanding of public perception, facilitating more effective marketing, customer service, and public opinion analysis. .

- **Project Objectives**

· **Emotion Categorization**: Classify emotions (e.g., happiness, anger) in social media posts.
· **Public Opinion Analysis**: Track sentiment trends on specific topics or events.
· **Real-Time Detection**: Monitor emotional shifts in real-time during events.
· **Demographic Mapping**: Link emotions to demographics (age, location, etc.).
· **Visualization Tools**: Create dashboards to visualize sentiment data.
· **Content Impact**: Analyze how different content types affect emotions.
· **Sentiment Bias**: Detect bias or emotional manipulation in posts.
· **Brand Monitoring**: Track customer sentiment to enhance brand strategies.
· **Predict Trends**: Use emotions to predict social movements or behaviors.
· **Ethical Analysis**: Address privacy, bias, and ethical issues in sentiment analysis.

- **Flow chart of the Project Work flow**



- **Data Description**

  - **Dataset Name**: Text.csv
  - **Source**: kaggle
  - **Type of Data**: Tweets, replies, retweets
  - **Recordsand Features**: 1 million posts ,Categorical Features (12) + Numerical Features (5)
  - **Target Variable**: Happiness,Sadness,Anger,Fear,Surprise,Disgust,Neutral.
  - **Static or Dynamic**: Static dataset
  - **Attributes Covered**: For emotion classification: Text_Content, Emotion_Label, Emotion_Score and For sentiment analysis: Sentiment_Label, Text_Content.
  - **Dataset Link**: https://www.kaggle.com/datasets/adhamelkomy/twitter-emotion-dataset

- **Data Preprocessing**

  - **Text Cleaning**

i. Remove URLs, mentions (@user), hashtags, special characters
ii. Lowercase all text

- **Emoji and Emoticon Handling**

i. Convert emojis to text (e.g., □ → "happy face")
ii. Remove or translate emoticons (e.g., :), :()

- **Tokenization**

i. Split text into individual words or tokens

- **Stop Word Removal**

i. Remove common words (e.g., "the", "is", "and") that do not carry emotion

- **Lemmatization / Stemming**

i. Reduce words to their root form (e.g., "running" → "run")

- **Noise Filtering**

i. Remove numbers, repeated characters (e.g., "soooo" → "so"), and excess whitespace

- **Handling Imbalanced Classes (optional)**

i. Apply techniques like SMOTE or undersampling if some emotions are underrepresented

- **Vectorization**

Convert text to numerical features using methods like:

i. TF-IDF
ii. Word2Vec / GloVe
iii. BERT embeddings

- **Exploratory Data Analysis (EDA)**

1. Emotion Distribution
- What to check: Distribution of emotions (e.g., joy, anger, sadness).
- Visualization: Bar chart or pie chart.

sns.countplot(x='emotion', data=df)

2. Text Length
- What to check: Length of social media posts.
- New Feature: text_length = df['text'].apply(len)
- Visualization: Histogram of text length.

df['text_length'] = df['text'].apply(len)
plt.hist(df['text_length'], bins=30)

3. Sentiment Distribution (if available)

- What to check: Distribution of sentiments (positive, neutral, negative).
- Visualization: Bar chart.

4. Most Frequent Words (Optional)

- What to check: Top words in the dataset.
- Visualization: Word cloud or bar plot.

Key Insights

- Emotion class imbalance.
- Outliers in text length.
- Sentiment skew.

- **Feature Engineering**

  - Text Length and Word Count help understand the volume of emotion in a post.
  - Sentiment Score quantifies the overall emotional tone.
  - Keyword Presence captures emotional cues from specific words.
  - POS tags and Emotion Lexicons reveal deeper insights into emotional content.
  - TF-IDF and Word Embeddings convert text into machine-readable vectors.

- **Model Building**

Algorithms Used:

- Traditional Algorithms (Logistic Regression, Naive Bayes, SVM) work well for small datasets and simpler tasks.
- Deep Learning (LSTM, CNN, Transformers) excels for large datasets and complex emotion detection in sequential and contextual text.
- Ensemble Methods (Random Forest, XGBoost) improve classification accuracy through multiple models.

Model Selection Rationale:

- Data Size: Simple models (e.g., LR, Naive Bayes) work well with small data, but deep learning models require large datasets.
- Performance vs. Interpretability: Deep learning models provide high performance but are less interpretable compared to traditional models like SVM or Logistic Regression.

- Resources: Complex models (e.g., BERT) require more computational power (GPU/TPU), while simpler models are less resource-intensive.

Train-TestSplit:

- Use stratification to maintain balanced emotion classes in both training and test sets.
- Randomly shuffle the data to avoid any inherent order in the data.
- Split your data into at least 80% training and 20% testing to ensure sufficient data for training while still evaluating the model on unseen data.

Evaluation Metrics:

- Accuracy: Overall correctness.
- Precision: Correctly predicted positive instances out of all positive predictions.
- Recall: Correctly predicted positive instances out of all actual positives.
- F1-Score: Balanced measure of Precision and Recall.
- Confusion Matrix: Provides a detailed breakdown of true vs. predicted labels.
- ROC-AUC: Measures model performance in binary classification tasks.
- Macro, Micro, and Weighted Averaging: For multiclass evaluation.
- Matthews Correlation Coefficient (MCC): Balanced performance metric for imbalanced datasets.

- **VisualizationofResults&ModelInsights**

  - **Feature Importance**:
    - For Tree-based models: Feature importance can be directly extracted from models like Random Forest, Gradient Boosting, and Decision Trees.
    - For Linear models: Coefficients from models like Logistic Regression and SVM provide insight into feature importance.
    - For any model: Permutation Importance is a model-agnostic technique to measure feature impact on performance.

        - Visualization: Use bar plots or word clouds to visualize the most important features.

- **Model Comparison**:
    - Logistic Regression (LR)
    - Support Vector Machine (SVM)
    - Random Forest (RF)
    - Naive Bayes (NB)
    - Recurrent Neural Networks (RNN)
    - Long Short-Term Memory (LSTM)
    - Transformer-based Models (BERT, RoBERTa)

- **Residual Plots**:
    - In the context of emotion detection and sentiment analysis of social media conversations, residual plots are typically used to visualize how well a model's predictions align with the actual target values.
    - Residuals are the differences between the predicted values and the true values. For classification tasks (like emotion detection), residual plots may not be as commonly used as in regression, but the concept can still be adapted to evaluate how well the model is predicting different classes and if there are patterns in its errors.

- **User Testing**:
    - User testing is an essential part of evaluating the performance and usefulness of sentiment analysis models, especially when applied to social media conversations.
    - In emotion detection, user testing helps ensure that the model is performing as expected and delivers reliable, actionable insights. User testing can focus on the accuracy, usability, and real-world application of the emotion detection system.

- **Tools and Technologies Used**

- **ProgrammingLanguage**:Python3
- **NotebookEnvironment**:GoogleColab
- **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**

1) Arunachalam P - Data cleaning,EDA

- Responsible for Data cleaning. It is a critical step in the process of sentiment analysis, especially when working with social media data
- Maintain to helps to understand the structure, distribution, and patterns in the data before building sentiment/emotion models.
- Maintain the concise, organized template that you can use or adapt for academic, business, or research purposes.

2) Deepakraj T - Feature engineering

- Responsible for Text Length and Word Count help understand the volume of emotion in a post.
- Maintain the TF-IDF and Word Embeddings convert text into machine-readable vectors.
- Responsible for Proper documentation and reporting are essential for clearly presenting your project, methodology, results, and insights.

3) Annamalai C - Model development

- Responsible for Feature importance can be directly extracted from models like Random Forest, Gradient Boosting, and Decision Trees.

- Maintain the Traditional Algorithms (Logistic Regression, Naive Bayes, SVM) work well for small datasets and simpler tasks.