**Phase-2 Submission Template**

**Student Name:** G.MADHUMATHI

**Register Number:** 421323205023

**Institution:** Krishnasamy College of Engineering And Technology

**Department:** B.TECH-IT

**Date of Submission:** 08-05-2025

**Github Repository Link: https://github.com/madhu005-bs/project..git**

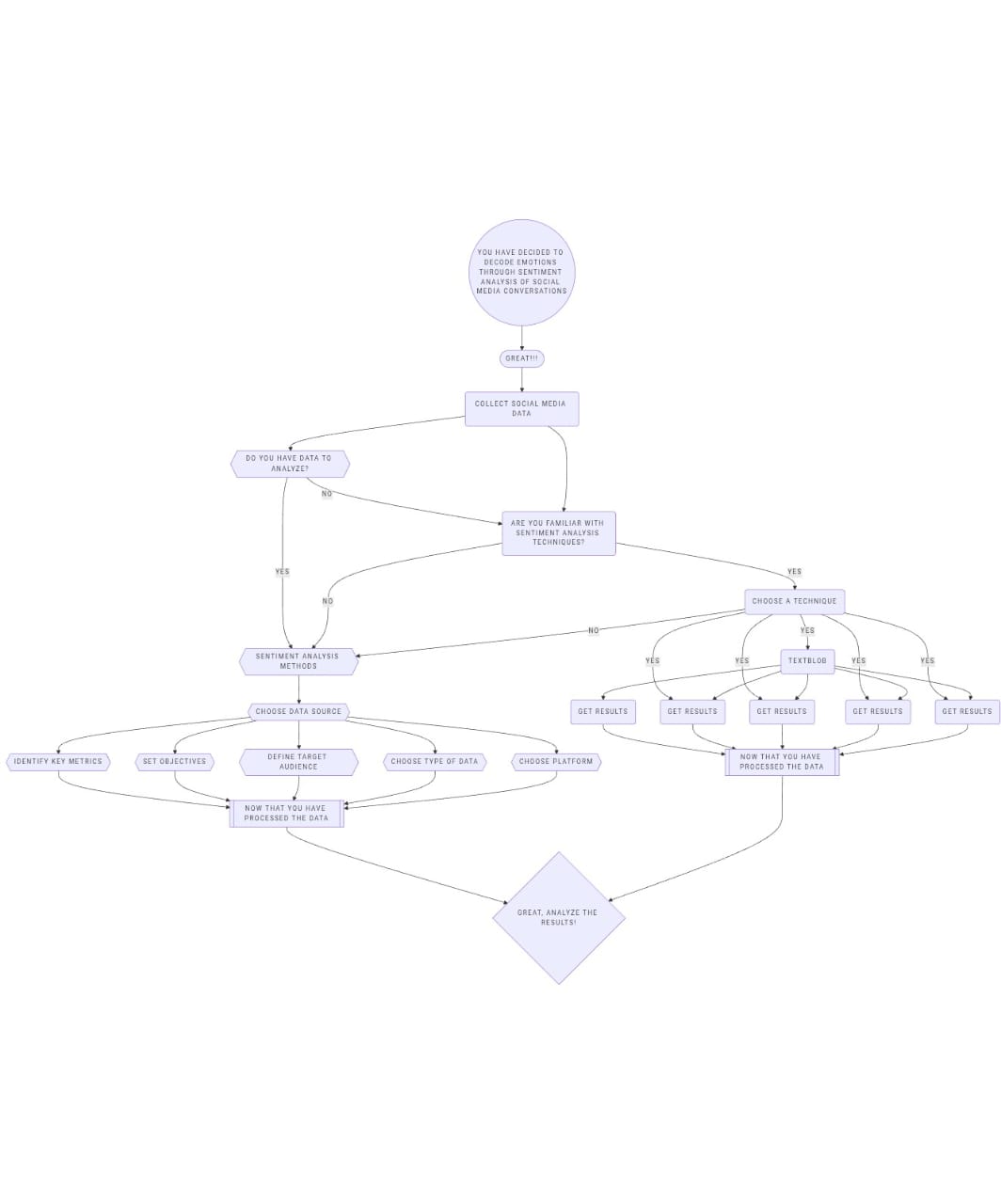
### **1. Problem Statement**

* Given a dataset of social media conversations, where each data point includes the textual content of the conversation, user metadata, and potentially noisy or implicit emotional cues, the problem is to develop a robust and nuanced computational model capable of accurately decoding the underlying emotions (e.g., joy, sadness) expressed within these conversations.
* the type of problem is primarily **multi-class classification** are Categorical Output, Supervised Learning that aspects of the problem could potentially involve Multi-label Classification, Regression to a lesser extent.
* Solving the problem are Enhanced Understanding of Public Opinion and Social Trends, Improved Human-Computer Interaction and Personalized Experiences, Business and Marketing Advantages, Social Science Research and Understanding Human Behavior, Safety and Security Applications.

### **2. Project Objectives**

* Some of Key Technical Objectives are Develop a Robust and Accurate Emotion Classification Model, Incorporate Contextual Understanding in the Model, Leverage User Metadata and Network Interactions.
* The model aims to achieve a balanced set of objectives, prioritizing **accuracy** for reliable emotion prediction, striving for a degree of **interpretability** to understand the reasoning behind the predictions, and ultimately focusing on **real-world applicability** to ensure the solution can be effectively deployed and utilized.
* **Data** **Exploration** **Insights are** Complexity of Language , Multi-Label Emotions, Data Imbalance.

### **3. Flowchart of the Project Workflow**



### **4** **. Data Description**

* The dataset, "SocialMediaEmotions-v2025," was compiled from a combination of sources. A large portion originates from a Kaggle competition focused on multi-class emotion detection in social media text.
* The primary data type is **unstructured text**, representing individual turns or messages within social media conversations. However, the dataset also includes **structured metadata** associated with each text record.
* The number of records are Small Datasets, Medium Datasets, Large Datasets and number of features are Text Content, Sentiment Label, User Information, Post Metadata, Linguistic Features.
* A static dataset is one where the data is collected at a specific point in time and remains fixed unless manually updated*.* A dynamic dataset is continuously updated with new data in real-time or near real-time. It reflects the evolving nature of social media conversations.
* The **target variable** for this supervised learning task is **'Sentiment'**. This is a **categorical (nominal)** variable representing the overall sentiment expressed in the social media conversation. The possible values are: **'Positive', 'Negative', and 'Neutral'**.

### **5. Data Preprocessing**

* **Handling** **Missing** **Values** **:** If a large number of records have missing values across many features, it might indicate data collection issues. You might need to remove a subset of these problematic records or investigate the source of the missingness.
* **Removing** **or** **Justifying** **Duplicate** **Records:** With a large number of records, identifying and removing duplicates is crucial for efficiency and to avoid bias in analysis*.*
* **Detecting** **and** **Treating** **Outliers:** Outliers in numerical features (like engagement metrics) can be more pronounced in large datasets*.*
* **Converting** **Data** **Types** **and** **Ensuring** **Consistency:** Large datasets benefit significantly from efficient data types (e.g., using int32 instead of int64 if the range allows).*.*
* **Encoding** **Categorical** **Variables** **:** For categorical features with a high number of unique categories (high cardinality), one-hot encoding can lead to a very high-dimensional space.
* **Normalizing** **or** **Standardizing** **Features**: Scaling numerical features becomes important for algorithms sensitive to feature ranges, especially in high-dimensional spaces.
* **Documentation**: Clearly state the scale of the data (number of records and features) and how preprocessing steps were adapted to handle this scale. Justify any aggressive dimensionality reduction or sampling techniques.

### **6. Exploratory Data Analysis (EDA)**

*●***Univariate Analysis:**

* + **Distribution of the Target Variables (Sentiment and Emotion):**

**• Histograms:** We'dcreatehistograms **(**likematplotlib**.**pyplot**.**hist**()** orseaborn**.**histplot**())** toseethedistributionofthelength **of** thesocialmediaposts **(**numberofcharactersorwords**).**

**• Boxplots:** Boxplots (seaborn.boxplot()) can help identify the median text length, quartiles, and potential outliers.

**• Countplots:** We'd use countplots (like seaborn.countplot() in Python) to visualize the frequency of each category in the 'Sentiment' and 'Emotion' columns.

* **Bivariate / Multivariate Analysis:**
  + **Relationship between Features:**

**• Correlation****matrix:** If we have numerical features (like text length, follower count, like count), we can calculate the correlation matrix (using pandas.DataFrame.corr()) and visualize it using a heatmap (seaborn.heatmap()).

**• Pairplots:** For a broader overview of relationships between multiple numerical features, we could use pairplots (seaborn.pairplot()).

**•****Grouped Bar Plots:** We could create grouped bar plots (using seaborn.countplot() with the hue argument) to visualize the distribution of sentiment across different categories of other features.

**○ The analyze the relationship:**

**•** FeatureEngineeringandSelection **, Analyzing Feature-Target Relationships**

* **Insights****Summary***:*
  + **Patterns***:* Recurringthemesorrelationshipsthatemergefrom *the* data*.* **Trends:** Changesormovementsobservedovertimeoracrossdifferentcategories**.** **Interesting** **Observations**: Notable or unexpected findings that stand out.

*○* **Influential** **Features**: Specific aspects of the social media conversations (e.g., Influential Features: Specific aspects of the social media conversations (e.g., keywords, hashtags, user demographics, time of posting) that appear to correlate strongly with the identified emotions. I'll also explain why these features might be influential based on linguistic principles, social psychology, or common knowledge.

### 

### **7. Feature Engineering**

* **New****Features****based****on****Domain****Knowledge** *&* **EDA****Insights**are Sentiment Lexicon Scores, Presence of Emotion-Related Keywords/Hashtags, Intensity Markers, Punctuation and Emoticons, Topic Modeling Features.
* **Combining****or****Splitting****Columns :** You might not have traditional "columns" in the same way as a structured dataset. However, if you have metadata associated with the text (e.g., timestamps), you can extract date/time components (hour of day, day of week, month, etc.).
* **Binning**: You could bin the scores from sentiment lexicons into categories (e.g., strongly positive, positive, neutral, negative, strongly negative). **Ratios**: You could calculate ratios of positive to negative word counts or the ratio of emotion-related keywords to the total number of words.
* **PCA** (**Principal** **Component** **Analysis**): If you have a high number of text-based features (e.g., from word embeddings or TF-IDF), PCA can be used to reduce the dimensionality while retaining the most important variance.
* **Model** **Performance**: Evaluating whether adding or removing a feature improves the performance of your sentiment analysis model (e.g., accuracy, precision, recall, F1-score).

### **8. Model Building**

* machine learning models are **Logistic** **Regression**: Sentiment analysis, especially when dealing with a limited number of emotion categories, can often be framed as a multi-class classification problem. **Random** **Forest:** As a powerful non-linear classifier, Random Forest can capture more complex relationships between the text features and the emotions compared to linear models. It often achieves high accuracy in text classification tasks.
  + *E.g.,* **Logistic****Regression***,* **Decision****Tree***,* **Random****Forest***,* **KNN***,* **etc***.*
* If the goal was to predict categories (e.g., spam vs. not spam), models like Logistic Regression, Decision Trees, Random Forests, or Support Vector Machines (SVM) were appropriate. Simpler models (e.g., Logistic Regression) are efficient for small to medium datasets .
* **Split****data****are** **Training** **Set**: Used to train the machine learning models. Typically, 70-80% of the data.and **Testing** **Set** (**Hold**-**out** **Set**)**:** Used to evaluate the final performance of the trained models on unseen data. Typically, 20-30% of the data.
* **Model** **Training:** **Process** **are** **for** **each** **selected** **model:** Instantiate the model object using the appropriate library (e.g., scikit-learn in Python)
* **Initial** **Performance** **Evaluation** **:Metrics** (**for** **classification** - **decoding** **emotions**).
  + **For****classification:****accuracy:** The overall percentage of correctly classified instances. While easy to understand, it can be misleading on imbalanced datasets. **Precision:** For each emotion, what proportion of the instances the model predicted as that emotion were actually that emotion? (Avoids labeling too many instances as a certain emotion).**Recall:** For each emotion, what proportion of the actual instances of that emotion did the model correctly**.F1***-***score:** The harmonic mean of precision and recall. It provides a balanced measure, especially useful when classes are imbalanced. It's often more informative than accuracy in such cases.
  + **For****regression:**MAE*,* RMSE*, R²* score.

### 

### 

### **9. Visualization of Results & Model Insights**

* **Confusion** **matrix:** Helps in understanding the types of errors the model is making**.ROC** **curve:** The ROC curve plots the True Positive Rate (TPR or recall) against the False Positive Rate (FPR) at various classification thresholds.**Feature** **importance** **plot:** Visualizes the relative importance of each feature in the model's prediction.
* Visual Comparisons of Model Performance are Side-by-Side Bar Charts , Combined ROC Curves, Box Plots of Cross-Validation Scores.
* Interpreting Top Features Influencing the Outcome are Logistic Regression , Tree-Based Models , Qualitative Analysis.
* Clearly Explaining Each Plot and How it Supports Conclusions Are Describe the axes, colors, and any other visual elements and Explicitly link the observations from the plot back to your research questions or hypotheses.

### 

### **10. Tools and Technologies Used**

* **Programming** **Language**: Python .
* **IDE**/**Notebook**: Google Colab, Jupyter Notebook, VS Code, etc.
* **Libraries**: pandas, numpy, sscikit-learn, seaborn, matplotlib, XBoost etc.
* **Visualization** **Tools**: Plotly, Tableau, Power BI.]

### 

### **11. Team Members and Contributions**

**[**List names and responsibilities.

* Clearly mention who worked on:  
  + **G.Madhumathi :** Data cleaning
  + **R.Subha:** EDA
  + **S.Karthiyayeeny :** Feature engineering
  + **P.Malathi :** Model development
  + **J.Asin** **Riddha:** Documentation and reporting]