**Student Name:** Harini S

**Register Number:** 613023104032

**Institution:** Vivekananda college of Technology for women

**Department:** BE Computer Science and Engineering

**Date of Submission:** 29.04.2025

### **1. Problem State****ment**

### **Problem:** Lots of social media text has hidden emotions about different things. Challenge: It takes too much time and effort to read and understand these emotions manually. This means we might miss important trends

### **Dataset Insight (Example**): Our data shows people express emotions in complex ways, like using sarcasm, not just simple positive or negative feelings.

### **Problem Type**: Mostly about sorting text into emotion categories (like happy, sad, angry). Could also involve guessing how strong an emotion is.

### **Why It's Important**:

### Know what people think right now.

### Help businesses understand customer feelings better.

### Spot social problems early

### Potentially help understand overall mental well-being.

### Improve how we communicate with each other.

### Show people content they might like based on their mood.

### **2. Project Objectives**

* **Main Tech Goals:**
* Create a computer model to correctly sort social media text into emotion groups (like happy, sad, angry).
* Clean up messy social media text (slang, abbreviations, emojis) so the model can understand it.
* Try different ways to turn text into numbers the model can use (like word meanings and advanced techniques).
* Test different computer programs to find the best one for sorting emotions.
* Set up a good way to check how well the model is working using different scores.
* Maybe try to guess how strong an emotion is.
* **What the Model Should Do:**
* Be very accurate at identifying emotions, better than basic tools.
* Give some idea of why it made a certain emotion guess (which words were important).
* Be useful for analyzing real social media data and giving helpful information.
* What We Learned from Looking at the Data (Example 1 - Class Imbalance):
* We saw that some emotions appear much more often than others. So, we need to make sure the model works well for all emotions, even the less common ones. We'll try things like balancing the data or telling the model to pay more attention to rare emotions.

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

### graph LR

### A[Collect Social Media Data] --> B(Preprocess Text Data);

### B --> C{Feature Extraction};

### C --> D[Train Emotion Classification Model];

### D --> E{Evaluate Model Performance};

### E -- Satisfactory --> F[Deploy Model for Analysis];

### E -- Unsatisfactory --> G[Refine Model (Go back to C or D)];

### F --> H[Analyze Social Media Conversations & Decode Emotions];

### H --> I[Generate Insights & Reports];

### Explanation of the steps:

### Collect Social Media Data: Gathering the relevant social media posts, tweets, or comments.

### Preprocess Text Data: Cleaning and preparing the text for analysis (e.g., removing noise, handling special characters, tokenization).

### Feature Extraction: Transforming the text data into a numerical format that the machine learning model can understand (e.g., using TF-IDF, word embeddings).

### Train Emotion Classification Model: Using the prepared data to train a machine learning model to classify emotions in text.

### Evaluate Model Performance: Assessing how well the trained model performs on unseen data using appropriate metrics.

### Satisfactory: If the model's performance is good enough, proceed to deployment.

### Unsatisfactory: If the performance is not good enough, go back to feature extraction or model training to make improvements.

### Deploy Model for Analysis: Making the trained model ready to analyze new social media data.

### Analyze Social Media Conversations & Decode Emotions: Using the deployed model to process social media text and identify the expressed emotions.

### Generate Insights & Reports: Summarizing the findings and creating reports based on the emotion analysis.

### **4. Data Description**

* **Dataset:** "GoEmotions", "CrowdFlower’s Emotion from "Kaggle", "Hugging Face Datasets"].
* **Data Type:** Primarily unstructured text data (social media conversations).
* **Size:** Approximately [State the number of records, e.g., "58,000"] records (e.g., tweets, comments). The main feature is the text content. There might be other features like timestamps, user information, but the focus is on the text.
* **Static/Dynamic:** Likely a static dataset for initial model development. However, the goal is to apply it to dynamic, real-time social media data in the future.
* **Target Variable:** The emotion label associated with each text sample (e.g., joy, sadness, anger, fear, surprise, neutral, and potentially more granular emotions depending on the dataset). This makes it a supervised learning task.

### **5. Data Preprocessing****:**

### **Missing Data:**

### Look for empty text or missing emotion labels. Usually, we just remove these completely.

### **Duplicates:**

### Find and delete identical social media posts.

### **Weird Text:**

### "Outliers" in text are usually gibberish or irrelevant stuff. We'll clean this up by removing too many links, symbols, or non-language text if it's a problem.

### **Consistent Data:**

### Make sure all the text is in the same format (like regular text). Make sure all the emotion labels are in a consistent category format.

### **Turning Emotions into Numbers:**

### The emotion words (joy, sadness) need to become numbers for the computer model. We’ll assign a number to each emotion.

### **Making Text Numbers Similar** (Later Step):

### After we turn the text into numbers (using techniques like word meanings), we might need to adjust these numbers to be on a similar scale, depending on the model we use.

### **Explaining What We Did:**

### We'll add comments in the code to explain each step.

### We'll also write down explanations in our project documents.

### Example Code Comment: # Removing rows with missing text

### Example Documentation: "Records with missing text or emotion labels were removed because they can't be used for training."

### Example Code Comment: # Using Label Encoder to convert emotion strings to numbers

### Example Documentation: "The 'joy', 'sadness', etc., emotion categories were converted to numerical labels (0, 1, 2, etc.) for the machine learning model."

### **6. Exploratory Data Analysis**

### Emotion Count: See how many times each emotion shows up (bar chart). Helps spot common/rare emotions.

### Text Length: See how long the posts usually are (histogram).

### Looking at How Things Relate:

### Length vs. Emotion: Do certain emotions have longer or shorter posts? (box plots).

### Words per Emotion: What are the most common words for each feeling? (word clouds, bar charts). Shows how people talk about different emotions.

### What We Learn:

### Emotion Balance: Are some emotions way more common? (Important for model training).

### Length Clues: Does post length hint at the emotion?

### Emotion Keywords: What words strongly link to each emotion? (Very important for the model).

### Key Features: The actual words are the main things the model will learn from. How often words appear and word patterns matter most. Post length might also help a bit.

### **7. Feature Engineerin****g**

### Getting More Info from the Text:

### Word Count: How many words in a post? (Longer/shorter might link to emotions).

### Character Count: How many letters/symbols? (Similar to word count).

### Emojis: Are there emojis? What kind? (Emojis are strong emotion clues!).

### Hashtags: How many hashtags? (Can show topic or feeling).

### Mentions (@): How many times someone is tagged? (Might be directed emotion).

### Links (URLs): Is there a link? (Less direct emotion info, but could give context).

### Punctuation: How many exclamation marks, question marks, etc.? (Can show feeling

### Emotion Word Scores: Use lists of emotion-related words to score each post for different feelings (like overall positive, negative, angry scores).

### Combining/Splitting (Less Common for Just Text)

### If we had post times, we could maybe see if emotions change by time of day or day of the week.

### Making Features Fewer (Usually Later):

### If we end up with too many text features (after turning words into numbers), we might need to reduce them to make the model faster and avoid confusion. But for text, we usually pick the most important features instead of just reducing them randomly.

### Why We Create These Features:

### We Know How People Talk: We know emojis and strong punctuation show emotion.

### EDA Told Us: If our earlier look at the data showed that longer posts tend to be sad, we'd use word count as a feature.

### **8. Model Building**

### **Pick Models:** Simple (Naive Bayes), Smart (SVM).

### **Split Data:** Train (80%), Test (20%), Keep emotions balanced.

### **Train:** Teach models with training data.

### **Test:** See how well they guess emotions on new data using accuracy, precision, recall, F1-score.

### **Compare:** Find the best model based on scores.

### 

### **9. Visualization of** Results **& Model In****sights**

### **Confusion Matrix:** Shows which emotions the model mixes up.

### **Model Score Bars:** Compares how good each model is (accuracy, etc.) for each emotion.

### **Word Importance** (if applicable): Shows which words the model thinks are key for each emotion.

### **Top Words:** List the most important words for each feeling and why they make sense.

### **Explain Plots:** Briefly say what each chart shows and what we learn from it (e.g., "Model often mistakes sad for scared").

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

### **Programming Language:** Python

### **IDE/Notebook:** Google Colab, Jupiter Notebook

### **Core Libraries:**

### pandas (for data manipulation)

### NumPy (for numerical operations)

### scikit-learn (for machine learning models, evaluation, and preprocessing)

### **Visualization Libraries:**

### matplotlib (for basic plotting)

### seaborn (for enhanced statistical visualizations)

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

* **Team Member 1: Harini S**

Responsibilities: Project Lead, Documentation and Reporting Lead

Contributions: Oversaw project progress, compiled and wrote the final report, managed documentation.

* **Team Member 2: Durgasri M**

Responsibilities: Data Cleaning Lead

Contributions: Performed data cleaning tasks (handling missing values, duplicates, outliers), ensured data consistency.

* **Team Member 3: Madhumitha P**

Responsibilities: Exploratory Data Analysis (EDA) Lead

Contributions: Conducted statistical and visual exploration of the data, identified patterns and insights.

* **Team Member 4: Abinaya P**

Responsibilities: Feature Engineering Lead

Contributions: Developed and implemented new features to improve model performance, justified feature choices.

* **Team Member 5: Gowsika S**

Responsibilities: Model Development Lead

Contributions: Selected, implemented, trained, and evaluated machine learning models.