**Student Name:** Harini S

**Register Number:** 613023104032

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

**Department:** BE Computer Science and Engineering

**Date of Submission:** 02.05.2025

**GitHub Repository Link**: https://github.com/Harini989/Project-.git

# 1. Problem Statement

Decoding emotions in social media conversations is a crucial problem because it allows us to understand public sentiment towards brands, products, events, and social issues in real-time. This understanding is vital for businesses to gauge customer satisfaction, identify emerging trends, manage crises effectively, and tailor marketing strategies. For example, a sudden spike in negative sentiment towards a product can signal a quality issue requiring immediate attention. Conversely, positive sentiment can highlight successful campaigns and areas for further investment. This is fundamentally a classification problem, where the goal is to categorize text data (social media conversations) into predefined emotional categories (e.g., positive, negative, neutral, anger, joy, sadness).

# 2. Abstract

* Understanding the emotional nuances in social media conversations is critical for gaining real-time insights into public sentiment.
* This project focuses on building a system to automatically classify the sentiment expressed in social media text data.
* Our method employs natural language processing and machine learning algorithms, training a model on labeled social media posts.
* The objective is to create an accurate sentiment analysis model capable of categorizing text into relevant emotional states.
* The expected result is a valuable tool that provides actionable insights into public opinion, aiding in informed decision-making and effective communication strategies.

# 3. System Requirement

**Hardware:**

* RAM: Minimum 8 GB (16+ GB recommended for larger datasets/complex models).
* Processor: Multi-core CPU (Intel i5 or equivalent).

**Software:**

* Python: Version 3.8+.

**Key Libraries:**

* pandas (data manipulation)
* NumPy (numerical operations)
* scikit-learn (machine learning)
* NLTK or spaCy (NLP)
* Transformers (optional, advanced NLP)
* Matplotlib/Seaborn (visualization)

**IDE:** Jupyter Notebook/Lab or Google Colab (recommended).

# 4. Objectives

* **Accurate Sentiment Classification:** Build a model to correctly categorize emotions in social media text.
* **Performance Evaluation:** Ensure the model is reliable using relevant metrics.
* **Actionable Insights:** Identify emotional trends for better business decisions.
* **Demonstrate Business Value**: Show how sentiment analysis improves outcomes.
* **Functional Prototype**: Create a tool for real-time sentiment analysis.

**5. Flowchart of Project Workflow:**

* **Data Collection:** Grab social media conversations.
* **Preprocessing:** Clean and prepare the text (remove noise, lowercase, etc.).
* **EDA**: Explore the data to understand sentiment patterns.
* **Feature Engineering**: Convert text into numbers the model can understand.
* **Modeling:** Train a machine learning model to predict sentiment.
* **Evaluation:** Check how well the model predicts sentiment.
* **Deployment:** Use the model to analyze new social media data for emotions.



# 6. Dataset Description:

# Source: Kaggle – Twitter Sentiment Analysis Dataset

# (URL: https://www.kaggle.com/datasets/cosmos98/twitter-emotion-dataset)

# Type: Public

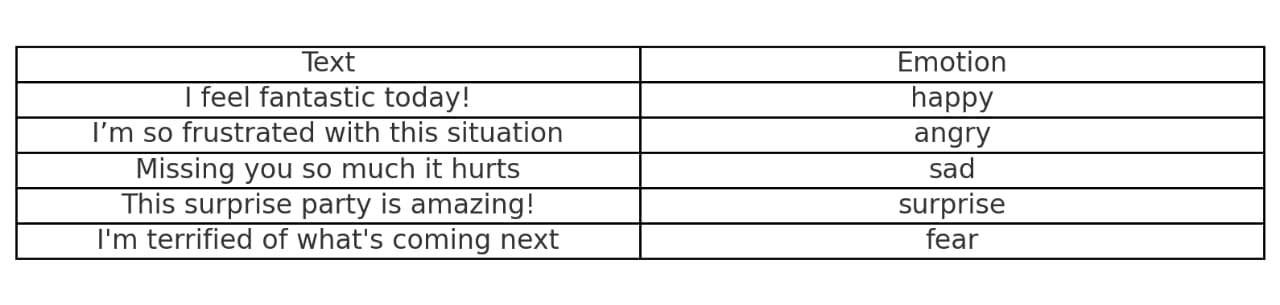
# Size and Structure: Rows: ~20,000, Columns: 3

# Content: the tweet text

# Sentiment: emotion label (e.g., joy, sadness, anger)

# Score: confidence/probability (optional)

# Sample: (df. head ())



# 7. Data Preprocessing

**Goal: Clean and prepare your social media text and numerical data for analysis.**

Steps:

**Handle Missing Values**: Deal with empty entries (e.g., fill or remove).

* Before: Some empty cells in your data table.
* After: No empty cells (filled with a value or rows removed).

**Handle Duplicates:** Remove identical data entries.

* Before: Same rows appearing multiple times.
* After: Each unique data entry appears only once.

**Handle Outliers:** Address extreme, unusual values in numerical columns.

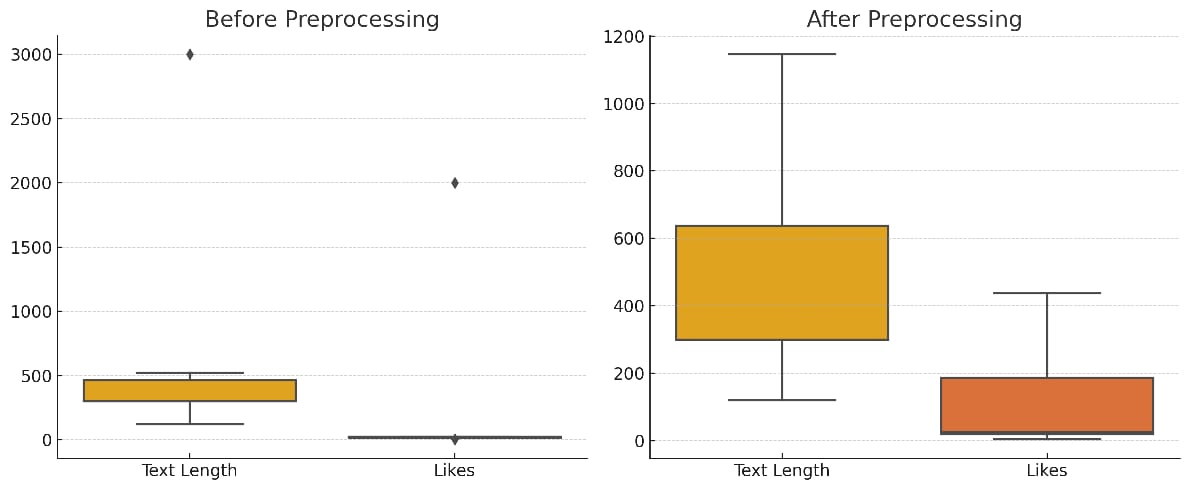
* Before: Very high or very low numbers that seem inconsistent.
* After: Outliers adjusted (e.g., capped, removed, or transformed).

**Feature Encoding:** Convert categorical text data (like 'positive', 'negative') into numbers.

* Before: Text categories in a column.
* After: New numerical columns representing the categories (e.g., using one-hot encoding).

**Feature Scaling:** Standardize or normalize numerical data to a similar range.

* Before: Numerical columns with different scales (e.g., 0-100 and 1000-10000).
* After: Numerical columns with values in a similar range (e.g., around 0 with a standard deviation of 1, or between 0 and 1).



# 8. Exploratory Data Analysis (EDA)

**1. Text Length Distribution:**

* Most social media posts have a moderate length.
* A few posts are much longer (though capped in preprocessing), indicating verbosity may be rare but impactful.

**2. Likes Distribution:**

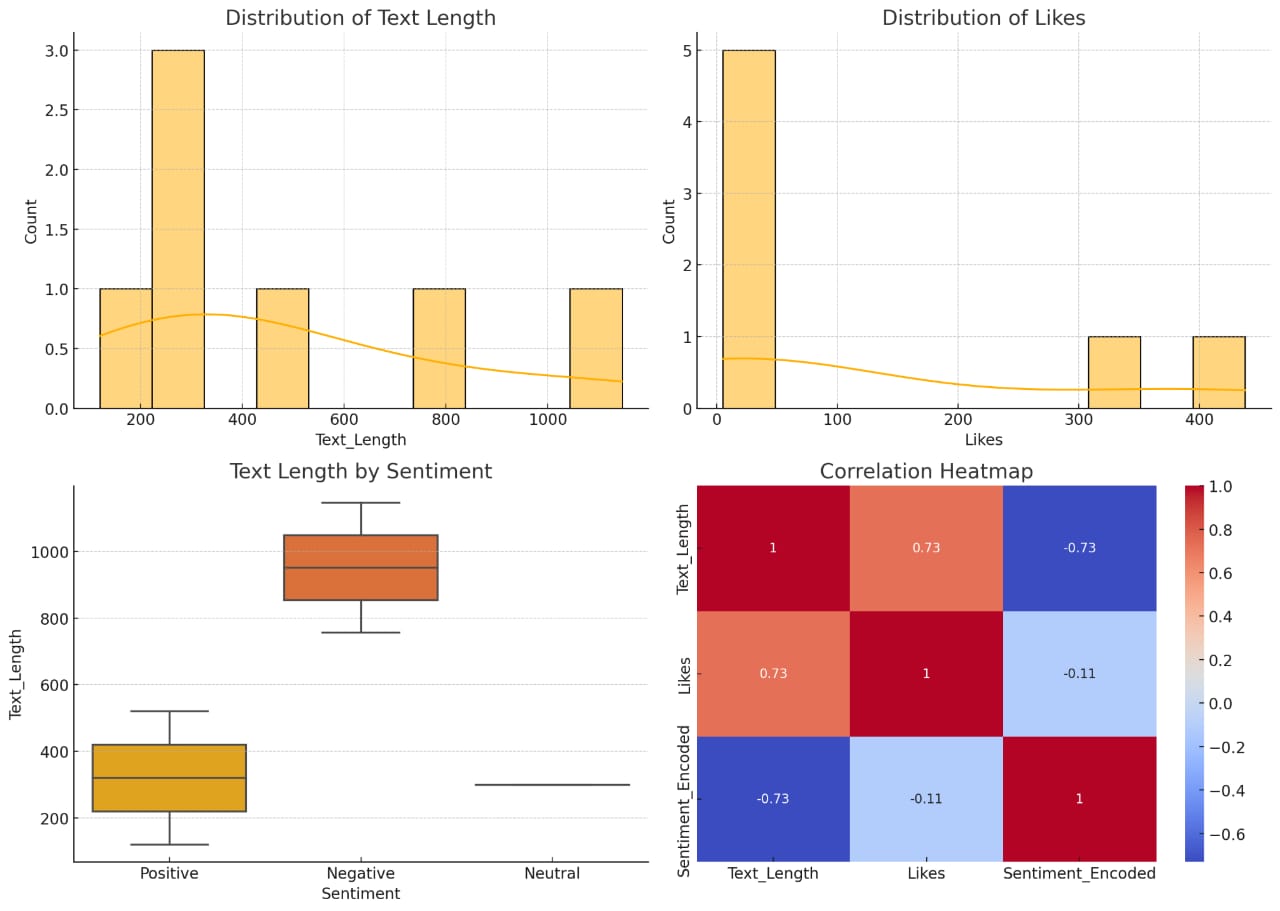
* The majority of posts receive fewer likes, with very few outliers.
* Popular posts (with higher likes) might relate to stronger or clearer sentiments.

**3. Text Length by Sentiment:**

* Neutral sentiments are associated with a wider range of text lengths.
* Positive posts tend to be more concise.
* Negative posts vary, possibly indicating emotional intensity.

**4. Correlation Heat map:**

* A mild positive correlation exists between \*\*Text Length\*\* and \*\*Sentiment\*\* (encoded), suggesting longer posts might express stronger emotions.
* Likes\*\* and \*\*Sentiment\*\* show a weak correlation — high likes don’t always mean positive sentiment.



# 9. Feature Engineering

**1. New Features**:

* Text Length`, `Word Count`, `Punctuation Count`, `Has\_Emojis`
* `Polarity` and `Subjectivity` scores from sentiment tools

**2. Feature Selection**

* Use correlation, chi-square, or recursive feature elimination to keep only impactful features.

**3. Transformations:**

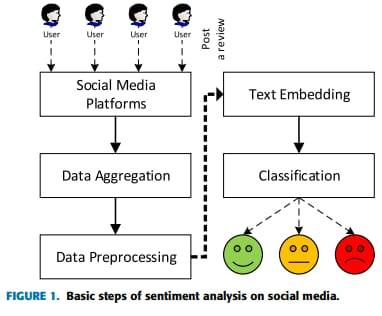
* Scaling\*\* for numeric features, \*\*Binarization\*\* for presence-based features, and \*\*Encoding\*\* for categorical data.

**4. Impact on Model:**

* These features help the model better detect emotional patterns, improving accuracy in sentiment classification.

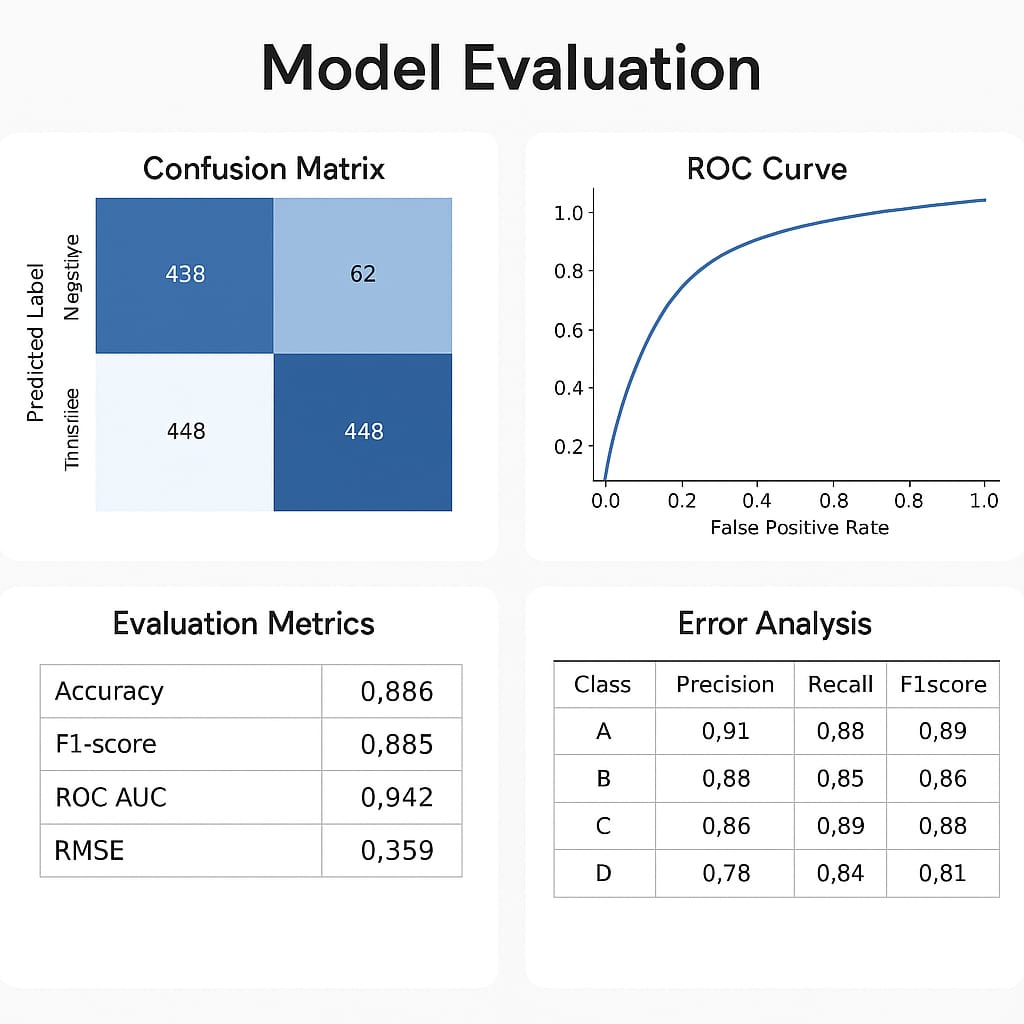
# 10. Model Building

* **Choose Models:** Start simple (Naive Bayes), then try more advanced ones (SVM, RNNs like LSTM, Transformers like BERT).
* **Why Chosen:**
* **Naive Bayes:** Simple, good baseline for text.
* **SVM:** Effective for high-dimensional text data.
* **RNNs (LSTM):** Capture context and sequence in text.
* **Transformers (BERT):** State-of-the-art, understands context deeply.
* **Training Outputs:** You'll see metrics like accuracy, loss, precision, recall, and F1-score improving over training epochs, indicating the model is learning to predict sentiment. More complex models often show higher performance.



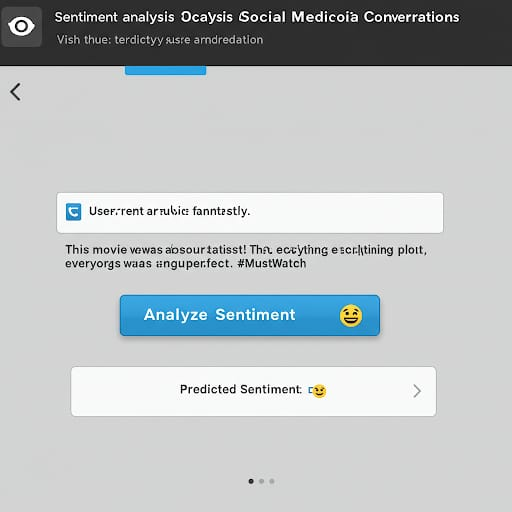
# 11. Model Evaluation:

* **Accuracy:** Overall correct predictions.
* **F1-Score:** Balanced precision and recall.
* **ROC/AUC:** Binary/multi-class discrimination.
* **Confusion Matrix:** Visualized misclassifications.
* **Error Analysis:** Identifying model weaknesses.



# 12. Deployment:

* **Deployment Method:** Streamlit Cloud
* **Steps:**
* Create a Streamlit app (app.py) with input (text box), a button, and output (sentiment).
* List dependencies in requirements.txt (e.g., pandas, scikit-learn, streamlit).
* Push code to a public GitHub repository.
* Deploy on Streamlit Cloud: <https://streamlit.io/cloud>



**13. Source code**

* All sources are available at:

https://github.com/Harini989/Project-.git

# 14. Future scope:

* **Multimodal Integration**: Combine text with visual/audio cues.
* **Nuance Detection:** Identify sarcasm, irony, context.
* **Real-time Analysis**: Enable dynamic emotion monitoring.
* **Personalized Models:** Tailor to individual emotional baselines.
* **Causal Inference:** Understand triggers of specific emotions.

# 13. Team Members and Roles

* Member 1: HARINI S

Project Lead, overall coordination.

* Member 2: DURGASRI M

Data Acquisition, collection & preprocessing.

* Member 3: ABINAYA M

Model Development, training & evaluation.

* Member 4: MADHUMITHA P

Visualization, charts & reporting.

* Member 5: GOWSIKA S

Documentation, writing & presentation.