Student Information

Student Name: AYYANAR S

Register Number: 620123106011

Institution: AVS Engineering College

Department: Electronics and Communication Engineering (ECE)

Date of Submission: 10/05/2025

GitHub Repository Link: https://github.com/ayyanar12345/Ayyanar.git

1. Problem Statement

The project "Decoding Emotions Through Sentiment Analysis of Social Media Conversations" addresses the need to automatically detect and classify human emotions from social media platforms like Twitter.

- Refinement from Phase-1: Focus shifted from general sentiment (positive/negative) to detecting specificemotions such as happiness, anger, sadness, etc.
- Problem Type: Multi-class text classification.
- Why It Matters: Emotion detection helps businesses, organizations, and governments gauge public opinion,improve customer service, respond to mental health signals, and manage crises effectively.

2. Project Objectives

- Develop a machine learning model that accurately classifies tweets into emotion categories.
- Improve accuracy and generalizability on informal, noisy social media text.
- Identify key features and patterns that distinguish one emotion from another.
- Updated Goal: After data exploration, we focused on better handling of class imbalance and enhancingcontext understanding using NLP techniques.

3. Flowchart of the Project Workflow

1. Data Collection
2. Data Preprocessing
3. Exploratory Data Analysis
4. Feature Engineering

5. Model Selection & Training

6. Model Evaluation

7. Visualization of Results

8. Conclusion & Future Work

4. Data Description

- Dataset Name: Emotion Classification Dataset (Twitter-based)

- Source: Kaggle

- Data Type: Unstructured textual data (tweets)

- Records and Features: ~40,000 tweets with one text column and one target label (emotion) - Nature: Static dataset

- Target Variable: Emotion label (joy, anger, sadness, fear, love, surprise)

5. Data Preprocessing

- Removed null or incomplete entries.

- Cleaned text by removing URLs, mentions, emojis (converted to text), and special characters.

- Applied tokenization, stopword removal, and stemming.

- Encoded target labels using LabelEncoder.

- Transformed text using TF-IDF Vectorizer for model input.

- Ensured uniform data format and handled imbalances using oversampling.

6. Exploratory Data Analysis (EDA)

- Univariate Analysis:

- Count plots showed 'joy' and 'sadness' were most common.

- Word clouds helped visualize dominant words for each emotion.

- Bivariate/Multivariate Analysis:

- Correlation plots showed frequent co-occurrence of certain words with specific emotions.

- Insights:

- Words like "happy", "love", "hate", and "cry" were strong indicators.

- Imbalance noted in emotion categories, addressed in model stage.

7. Feature Engineering

- Extracted new features: tweet length, polarity scores, emoji-to-text mappings.
- Created n-grams (bigrams/trigrams) to capture word sequences.
- Merged similar emotion classes for experimental trials.
- Attempted dimensionality reduction for visualization (PCA, t-SNE).

8. Model Building

- Models Used:
- Logistic Regression (baseline)
- Random Forest (for interpretability)
- Multinomial Naive Bayes (for text data)
- Justification: Text data suits probabilistic and ensemble models.
- Data Split: 80% train, 20% test with stratified sampling.
- Evaluation Metrics:
- Accuracy
- Precision, Recall, F1-Score (macro avg)
- Confusion matrix for visual error detection

9. Visualization of Results & Model Insights

- Confusion Matrix: Showed misclassification between similar emotions like love and joy.
- Feature Importance: Displayed top TF-IDF words contributing to each emotion.
- ROC Curve: One-vs-rest ROC curves for multi-class evaluation.
- Model Comparison: Logistic Regression performed best on macro F1-score.
- Key Takeaway: Common words and slang can heavily influence emotion prediction.

10. Tools and Technologies Used

- Programming Language: Python
- IDE/Notebook: Google Colab

- Libraries: pandas, numpy, seaborn, matplotlib, scikit-learn, nltk, emoji, xgboost

- Visualization: seaborn, matplotlib, wordcloud, Plotly

11. Team Members and Contributions

Team Member	Responsibility
Ajithkumar E	Data preprocessing, feature engineering
Meganadhan M	EDA, model training and performance evaluation
Ayyanar S	Visualization, model insights, interpretation
Gowtham V	Documentation, GitHub setup, report finalization