Emotion Intensity Detection Using Machine Learning

1. Project Overview

The Emotion Intensity Detection project aims to analyze social media tweets and predict the emotional state of users such as Anger, Fear, Joy, Neutral/Calmness, and Sadness. The system uses Machine Learning and Natural Language Processing (NLP) techniques to classify emotions based on tweet text and related attributes.

2. Dataset Description

- Total Records: 101
- Total Features: 8

Dataset Columns

- ID – Unique identifier
- Tweet – Original tweet text
- Tweejt – Cleaned tweet text after preprocessing
- Emotion – Target emotion class
- Age – User age
- Gender – User gender
- Platform – Social media platform
- User Location – Location of the user

Data Properties

- All columns are of object data type

- No missing values found (df.isnull().sum() = 0)
- Emotion distribution includes 6 unique emotion categories
- Dataset size is relatively small, which affects model performance

3. Data Preprocessing

The following preprocessing steps were performed:

- Removal of unnecessary columns
- Text cleaning:
    - Converted text to lowercase
    - Removed stopwords using NLTK
- Conversion of Age to numeric format
- Removal of invalid or null age values
- Feature–target separation
- Train–test split:
    - 80% training data
    - 20% testing data
    - random_state = 42 for reproducibility

4. Feature Engineering

- CountVectorizer was used to convert tweet text into numerical form
- Text data was transformed into word frequency vectors
- Same vectorizer was applied to training and testing data to maintain consistency

5. Model Implementation

Machine Learning Models Used

- Logistic Regression for emotion classification
- Model trained on vectorized tweet text

Model Saving

- Trained model saved as:
    - log_reg_model.pkl
- Vectorizer saved as:
    - vectorizer.pkl

These files were later used for deployment.

6. Model Evaluation

Accuracy

- Model Accuracy: 0.25 (25%)

This indicates that the model correctly predicts emotions 25% of the time.

Confusion Matrix (Sample Output)

```
[[1 1 0 0 0]
 [5 1 0 0 0]
 [2 0 2 0 2]
 [1 0 0 1 0]
[3 0 1 0 0]]
```

- Shows how predictions are distributed across emotion classes
- High misclassification observed due to:
    - Small dataset
    - Class imbalance

Classification Report

Emotion	Precision	Recall	F1-Score
Anger	0.08	0.50	0.14
Fear	0.50	0.17	0.25
Joy	0.67	0.33	0.44
Neutral/Calmness	1.00	0.50	0.67
Sadness	0.00	0.00	0.00

- Joy and Neutral/Calmness performed comparatively better
- Sadness had poor performance due to fewer samples
- Overall macro and weighted averages are low, indicating imbalance

7. Deployment

The trained model was successfully deployed using:

- Streamlit – Interactive web interface
- Flask / FastAPI – API-based emotion prediction

Functionality

- User enters a tweet
- Text is vectorized using saved vectorizer
- Model predicts emotion
- Result displayed in real-time

8. Observations & Limitations

- Small dataset size (101 records)
- Class imbalance across emotions
- Limited textual diversity
- No deep learning or word embeddings used
- Accuracy is low but acceptable for academic demonstration

9. Conclusion

This project successfully demonstrates an end-to-end machine learning pipeline for emotion detection, including:

- Data preprocessing
- Feature extraction
- Model training
- Evaluation
- Model saving
- Deployment

Although the accuracy is low, the project effectively shows the practical implementation of NLP and ML concepts and can be improved further with a larger dataset and advanced models.

10. Future Improvements

- Increase dataset size
- Use TF-IDF or word embeddings
- Apply Deep Learning models (LSTM, BERT)
- Handle class imbalance using resampling
- Improve text preprocessing techniques

Final One-Line Summary

This project implements an emotion intensity detection system using NLP and Logistic Regression, demonstrating the complete machine learning workflow from data preprocessing to deployment.