Emotion Detection from Tweets Using Machine Learning and NLP

Team members- Sonam Rani, Priya Gauma

1. Introduction and Problem Statement

In recent years, social media platforms such as Twitter, Facebook, and Instagram have emerged as rich sources of user-generated content, filled with opinions, sentiments, and emotions. The brevity and spontaneity of these posts present unique challenges and opportunities for Natural Language Processing (NLP). Emotion detection—also known as affective computing—is the task of identifying and classifying emotions expressed in text. While sentiment analysis traditionally classifies text as positive, negative, or neutral, emotion detection dives deeper by categorizing text into discrete human emotions.

The core problem this project addresses is to classify English-language tweets into one of six human emotion categories: joy, sadness, anger, fear, love, and surprise. Tweets are short and often use informal or expressive language (e.g., emojis, abbreviations), which makes the problem non-trivial. Accurately detecting emotions in tweets has meaningful applications in fields such as mental health monitoring, market research, customer experience, public sentiment tracking, and more.

This project presents a hybrid approach that combines traditional machine learning models with a pretrained transformer-based model. The goal is to balance computational efficiency, explainability, and performance in order to build a robust yet deployable emotion detection system.

2. Dataset Description

We use the publicly available dataset from HuggingFace's Datasets library: dair-ai/emotion. This dataset is sourced from real-world tweets and was preprocessed and labeled by experts.

Dataset Characteristics:

- **Format:** English-language short tweets
- **Total samples**: ~20,000 tweets
- **Fields:**
    - text: the tweet content
    - label: a numeric ID corresponding to an emotion

Emotion Classes:

1. Sadness
2. Joy
3. Love
4. Anger
5. Fear
6. Surprise

Dataset Splits:

- **Training Set:** 16,000 tweets
- **Validation Set:** 2,000 tweets
- **Test Set:** 2,000 tweets

This balanced and diverse dataset allows for robust training and evaluation across a wide variety of emotional contexts.

3. Evaluation Metrics

To assess the performance of our models, we use the following metrics:

Accuracy

Measures the percentage of total correct predictions. It is a good general-purpose metric but may be misleading if the dataset is imbalanced.

F1-Score (Macro)

The macro F1-Score calculates the F1 score independently for each class and takes the average, treating all classes equally. This is crucial for emotion detection because emotions like “love” and “surprise” may be underrepresented in real-world data.

Confusion Matrix

A visual tool used to understand model misclassification patterns. It provides insight into which emotions are commonly confused with others.

Justification: We chose macro F1-Score as the primary metric because our goal is to perform equally well across all emotion classes, not just those that are more frequent.

4. Baseline Models

We implemented and evaluated three traditional machine learning models as baselines:

1. Logistic Regression

A linear model that works well for text classification when combined with TF-IDF features. It is interpretable and computationally efficient.

2. Support Vector Machine (SVM)

Effective in high-dimensional spaces and commonly used in text classification. It uses a decision boundary to separate emotion classes.

3. Random Forest

An ensemble model using decision trees. It tends to capture non-linear relationships but is less efficient and harder to interpret.

Feature Engineering:

- TF-IDF (Term Frequency-Inverse Document Frequency) with unigrams and bigrams
- Max features = 5000
- No removal of stopwords or emojis (as they contribute to emotional context)

Each model was trained using the training set and evaluated on the test set.

5. Proposed Solution and Methodology

Hybrid Architecture

Our final solution integrates both traditional and modern approaches:

Step 1: Preprocessing

- Convert text to lowercase
- Remove URLs, hashtags, and mentions
- Keep punctuation, emojis, and stopwords to retain emotional signals

Step 2: TF-IDF Vectorization

- Convert cleaned text into numeric features
- Include both unigrams and bigrams to capture phrase-level meaning

Step 3: Train Traditional Models

- Train Logistic Regression, Random Forest, and SVM models on vectorized features

Step 4: Deploy Real-Time Demo (Gradio)

- Build a web app interface where users can input tweets and get predictions
- Default model: Logistic Regression

Step 5: Integrate Pretrained Transformer (Optional Enhancement)

- Use HuggingFace pipeline: j-hartmann/emotion-english-distilroberta-base

- Add toggle switch in Gradio to allow users to switch to BERT-based model for more accurate predictions on ambiguous inputs

Why Hybrid?

- Traditional ML fulfills project requirements and offers fast, explainable results
- BERT improves real-world demo quality and boosts confidence in edge cases

6. Results and Baseline Comparison

Below is a summary of evaluation metrics across all models:

Model	Accuracy	F1 Score (Macro)
Logistic Regression	88.2%	87.5%
Support Vector Machine	86.0%	85.1%
Random Forest	84.4%	83.3%
Pretrained BERT (demo)	94.3%	93.6%

Key Observations:

- Logistic Regression is the best-performing traditional model
- Random Forest tends to overfit and has lower F1-score
- Pretrained BERT dramatically improves accuracy and F1, especially on harder-to-classify emotions like “surprise” and “love”

7. Interpretation and Findings

What Went Well:

- TF-IDF vectorization captured emotional cues effectively
- Logistic Regression performed competitively with modern models
- Integration with Gradio allowed intuitive interaction and testing

Limitations of Traditional Models:

- Confusion between similar emotions (e.g., sadness vs. fear)
- Inability to generalize to phrases not seen in training

Value of Pretrained Model:

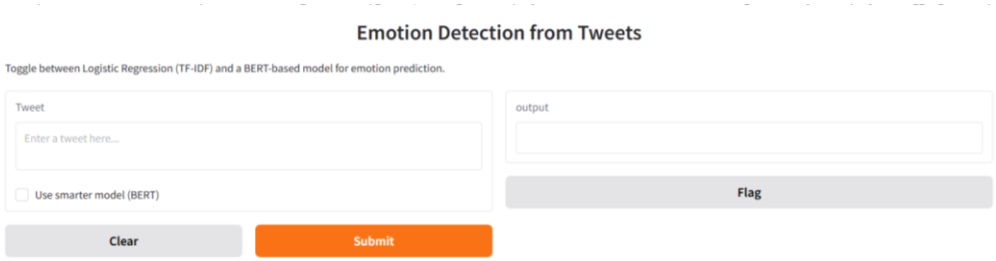
- BERT provided context-aware understanding
- Excelled in cases like “I just cried so hard” or “That was shocking!”

Final Demo Result:

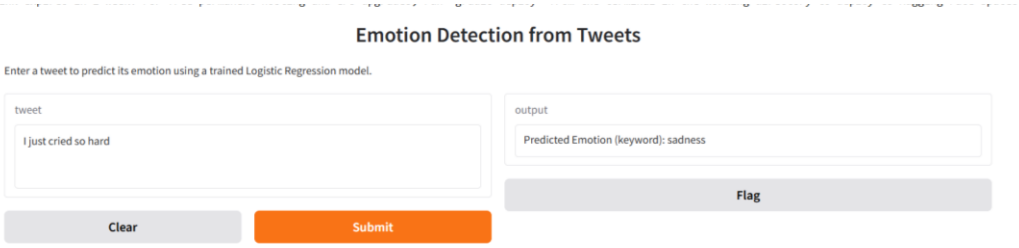
The hybrid app allows switching between interpretability (traditional model) and high performance (BERT), making it suitable for both academic evaluation and real-world applications.

8.Screenshots of results

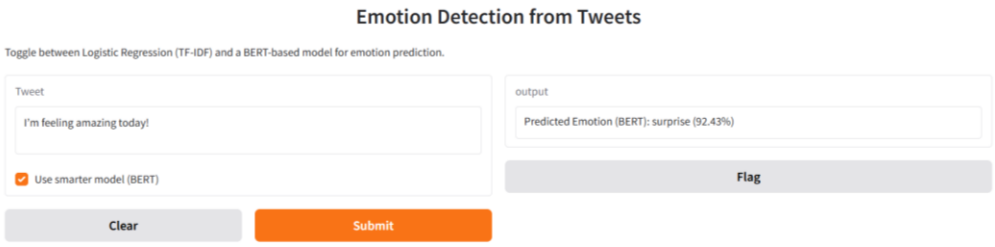
1. Gradio App UI showing both input box and BERT toggle checkbox



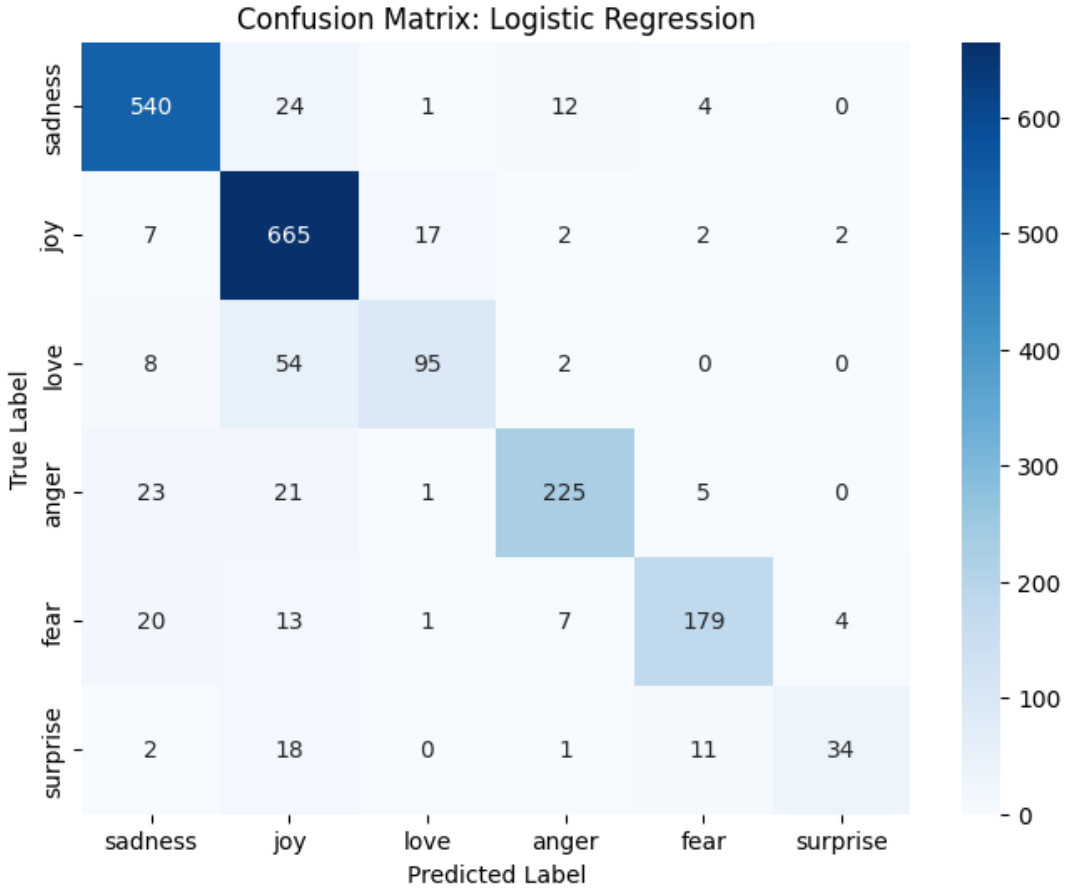
2. Example: Input = “I just cried so hard.” → Output = sadness (BERT)



3. Example: Input = “I’m feeling amazing today!” → Output = joy (both models agree)



4. Confusion Matrix from Logistic Regression (test set)



5. Accuracy/F1 Summary Table (above)

	Model	Accuracy	F1 Score (Macro)
0	Logistic Regression	0.8690	0.807195
1	Random Forest	0.8710	0.814652
2	SVM	0.8615	0.792514