

# ■ Lesson Plan: Build a Mood2Emoji App

**Topic:** Text-to-Emoji Mood Detector (Kid-safe Sentiment Classifier)

**Age Group:** 12–16 years

**Duration:** 60 minutes

**Tools Used:** Python, Streamlit, TextBlob (or rule-based logic)

## ■ Learning Objectives

- 1 Understand how computers can detect emotions from text.
- 2 Learn basic Natural Language Processing (NLP) concepts like sentiment analysis.
- 3 Build a simple, interactive web app using Streamlit.
- 4 Apply ethical AI principles by filtering bad words and ensuring kid-safe outputs.
- 5 Visualize how data flows from user input to emotion output.

## ■ Key Concepts Introduced

**Sentiment Analysis:** Understanding the emotional tone behind text (positive, negative, or neutral).

**Text Preprocessing:** Cleaning text before analysis (e.g., expanding “don’t” → “do not”).

**Profanity Filtering:** Removing or censoring inappropriate words to keep the app safe for kids.

**Rule-based Logic:** Using simple “if-then” rules to make decisions without heavy AI models.

**User Interface (UI):** Designing an interactive app where users type messages and see emoji results.

## ■ Lesson Flow (60 Minutes)

- 1. Introduction (10 mins)** Start with examples of happy/sad texts and discuss how computers can detect emotions.
- 2. Data Understanding & Preprocessing (10 mins)** Explain how text is cleaned and why contractions or bad words must be handled.
- 3. Building the Sentiment Logic (15 mins)** Show how TextBlob or rules detect polarity and handle negations like 'not bad'.
- 4. Building the App (15 mins)** Demonstrate creating a Streamlit app step-by-step.
- 5. Data Flow Explanation (5 mins)** Explain the flow from user input to emoji output using a diagram.
- 6. Wrap-Up & Discussion (5 mins)** Discuss what students learned and responsible AI practices.

## ■ Features of the Final App

- 1 Kid-Safe Output: Filters all profanity and harmful words.

- 2 Emoji Feedback: Shows 🤔, 😊, or 😐 based on detected mood.
- 3 Cheer-Up Messages: Encouraging lines for sad or neutral moods.
- 4 Interactive UI: Simple, fun, and visual interface built with Streamlit.
- 5 Teacher Mode: Shows internal data flow for classroom learning.
- 6 Offline & Lightweight: Uses TextBlob — no external APIs or large models.

## ■ Learning Outcomes

- 1 Understand text emotion detection in AI systems.
- 2 Learn basic NLP and Python integration.
- 3 Apply ethical programming through safe word filtering.
- 4 Gain hands-on experience with Streamlit app creation.
- 5 Be inspired to explore more advanced AI topics like chatbots and emotion recognition.