

# **Lesson Plan: Build a Mood2Emoji App**

## **Introduction to Text Classification and Sentiment Analysis**

### **Lesson Overview**

**Topic:** Build a Mood2Emoji App (Introduction to Text Classification)

**Duration:** 60 minutes

**Age Group:** 12-16 years old

**Subject:** Computer Science / AI & Coding

**Difficulty Level:** Beginner to Intermediate

### **Learning Goals**

By the end of this lesson, students will be able to:

1. **Define** sentiment analysis and explain how computers detect emotions in text
2. **Identify** positive, negative, and neutral words in sentences
3. **Understand** the basic workflow of a text classification system
4. **Explore** a working sentiment analysis application hands-on
5. **Analyze** the limitations and ethical considerations of AI text analysis
6. **Apply** critical thinking to evaluate when and how AI should analyze text

### **Topics Introduced**

#### **Core Concepts:**

1. **Sentiment Analysis** - How computers understand emotions in text
2. **Natural Language Processing (NLP)** - Teaching computers to understand human language
3. **Text Classification** - Organizing text into categories
4. **Polarity Scoring** - Measuring positive vs. negative sentiment
5. **Machine Learning Applications** - Real-world uses of AI

## **Technical Skills:**

1. Reading and interpreting code
2. Understanding input-output systems
3. Basic data analysis (scores, averages)
4. Web application interaction

## **Topics in Detail**

### **1. What is Sentiment Analysis? (10 minutes)**

**Definition:** Sentiment analysis is the process of determining whether a piece of text expresses positive, negative, or neutral emotions.

#### **Key Concepts:**

1. Computers don't "feel" emotions but can detect patterns
2. Words are assigned scores based on typical emotional associations
3. Positive words: love, happy, amazing, wonderful (+)
4. Negative words: hate, sad, terrible, awful (-)
5. Neutral words: the, is, was, and (0)

#### **Real-World Examples:**

1. **Social Media:** Instagram and Twitter detect harmful comments
2. **E-Commerce:** Amazon analyzes product reviews to help shoppers
3. **Customer Service:** Companies track customer satisfaction
4. **Entertainment:** Netflix understands reactions to shows

#### **Discussion Questions:**

1. How do you express happiness in a text message?
2. Can computers understand sarcasm? Why or why not?
3. Why would a company want to analyze customer feedback automatically?

## 2. How Does the Mood2Emoji App Work? (15 minutes)

### Step-by-Step Process:

User Types Sentence → Safety Check → Word Analysis → Score Calculation → Emoji Result

### Detailed Breakdown:

1. **Input:** Student types a sentence (e.g., "I love learning new things!")
2. **Safety Filter:**
3. Checks for inappropriate words
4. Protects students from harmful content
5. Returns neutral response if problematic text detected
6. **Text Processing:**
7. Sentence is broken into individual words
8. Each word is analyzed using TextBlob library
9. Library contains pre-scored dictionary of thousands of words
10. **Polarity Calculation:**
11. Each word gets a score from -1.0 (very negative) to +1.0 (very positive)
12. Scores are averaged across all words
13. Example: "I love learning" → "I"(0) + "love"(0.5) + "learning"(0.3) = Average: 0.27
14. **Decision Logic:**

If score > 0.2 → 😊 Happy

If score < -0.2 → 😞 Sad

Else → 😐 Neutral

1. **Output:** Display emoji with encouraging message

### Demonstration:

1. Teacher shows 3-4 examples live
2. Students predict results before running
3. Discuss why the app chose each emoji

### 3. Hands-On Exploration (15 minutes)

#### Activity 1: Test Your Own Sentences (7 minutes)

1. Each student tests 5 different sentences
2. Record results in worksheet:
3. Your sentence
4. Predicted emoji
5. Actual result
6. Mood score
7. Were you surprised?

#### Example Worksheet:

Sentence: "I can't wait for summer vacation!"

My Prediction: 😊

Actual Result: 😊

Mood Score: +0.65

Surprised?: No, makes sense!

#### Activity 2: Can You Trick the Detector? (8 minutes)

1. Challenge students to find edge cases
2. Examples to try:
3. Sarcasm: "Oh great, more homework"
4. Negation: "This is not bad"
5. Mixed emotions: "I'm happy but also nervous"
6. Discuss why these are tricky for computers

#### Class Discussion:

1. What patterns did you notice?
2. Which results surprised you?
3. What did the computer get wrong?

## 4. Understanding the Technology (12 minutes)

### Code Walkthrough (Simplified for students):

Show key parts of app.py:

```
# 1. Get user input
user_input = st.text_input("Type your sentence here:")

# 2. Analyze sentiment
blob = TextBlob(user_input)
polarity = blob.sentiment.polarity # Score from -1 to +1

# 3. Decide which emoji
if polarity > 0.2:
    emoji = "😊"
elif polarity < -0.2:
    emoji = "😞"
else:
    emoji = "😐"
```

### Key Takeaways:

1. Libraries like TextBlob do the complex work for us
2. We just need to write the logic (if/else statements)
3. Real applications layer multiple technologies together

### Teacher Mode Features:

1. Open the "Teacher Mode" section in the app
2. Walk through the visual diagram together
3. Explain the scoring system with examples
4. Show the "Behind the Scenes" metrics

## **Learning Outcomes**

### **Knowledge Outcomes:**

Students will know:

1. The definition of sentiment analysis
2. How computers assign scores to words
3. Real-world applications of text classification

### **Skills Outcomes:**

Students will be able to:

1. Analyze text for emotional content
2. Read and understand simple Python code

### **Attitude Outcomes:**

Students will:

1. Think critically about ethical use of technology
2. Appreciate the complexity of human language
3. Be curious about how everyday apps work

## Detailed Lesson Timeline

### Time Activity Method Materials

0-5 min	Introduction & Hook	Demo app, ask questions	Projector, app
5-10 min	Explain Sentiment Analysis	Lecture, examples	Slides, whiteboard
10-20 min	How It Works	Teacher Mode walkthrough	Live app demo
20-35 min	Hands-On Exploration	Individual/group work	Computers, worksheets
35-45 min	Code Understanding	Code walkthrough	app.py displayed
45-53 min	Ethics Discussion	Group debate	Discussion prompts
53-58 min	Wrap-Up Quiz	Interactive Q&A	Quiz questions
58-60 min	Reflection & Homework	Individual writing	Exit ticket

**Created for:** Curriculum Developer Intern Assignment

**Topic:** Introduction to AI and Text Classification

**Target Age:** 12-16 years

**Format:** 60-minute interactive lesson with hands-on coding exploration