Lesson Plan: Build a Mood2Emoji App

Topic: Introduction to Text Classification

Duration: 60 minutes

Age Group: 12-16 years

Technology: Python, Streamlit Web App

Learning Objectives:

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

- Understand basic text classification concepts
- Explain how computers can detect emotions in text
- Modify and extend a simple rule-based system
- Identify real-world applications of sentiment analysis
- Discuss the importance of content safety in AI applications

Materials Needed

- Computers with internet access
- Mood2Emoji web app (deployed or local)
- Projector for demonstrations
- Example sentences for testing

Lesson Structure

Part 1: Introduction & Hook (10 minutes)

Hook Activity: "Emoji Match Game"

- Show 5 text sentences on screen
- Students guess which emoji ( ) matches each sentence
- Discuss: "How did you know which emotion each sentence showed?"

Key Concepts Introduced:

- What is sentiment analysis?
- Real-world examples:
  - Social media mood detection
  - Customer review analysis
  - Chatbot responses
- How computers "understand" emotions

Demo: Show the working Mood2Emoji app with fun examples

Part 2: How It Works - The Magic Revealed (15 minutes)

Visual Explanation using Teacher Mode:

text

Input: "I love pizza!"

↓

Safety Check → ✅ Passed

↓

Word Counting:

Happy words: "love" = +1 point

Sad words: none = 0 points

↓

Decision: +1 > 0 → 😊 Happy!

Detailed Topics:

- Rule-Based Systems
  - Simple if-then logic
  - Word matching approach
  - Counting and comparing

- Safety First
  - Why we filter content
  - Age-appropriate design
  - Digital citizenship

- Decision Making
  - Happy words vs Sad words
  - Equal counts = neutral
  - Safety override

Part 3: Hands-On Coding Activity (25 minutes)

Activity 1: Word List Expansion (10 minutes)

Python

```
# Students modify these lists:

happy_words = {'happy', 'good', 'great', 'awesome', 'love'}

sad_words = {'sad', 'bad', 'terrible', 'angry', 'upset'}


# Challenge: Add 3 new words to each list

# Test: "This movie was fantastic!" → should detect as happy
```

Activity 2: Custom Responses (10 minutes)

python

```
# Students customize output messages:
if happy_count > sad_count:
    return "  ", "You sound thrilled!"  # Change this message
elif sad_count > happy_count:
    return "  ", "I hear your sadness"  # Change this message
```

Activity 3: Edge Case Testing (5 minutes)

- Test sarcasm: "Great, I love failing tests"
- Test mixed emotions: "I'm sad my pet ran away but happy we found him"
- Test safety: "I hate everything"

Part 4: Discussion & Real-World Connections (10 minutes)

Group Discussion Questions:

- Where have you seen this technology in real life?
- What surprised you about how it works?
- What are the limitations we noticed?
- Why is safety filtering important for young users?

Real-World Applications:

- Mental health apps that detect user mood
- Social media platforms analyzing post sentiment
- Customer service chatbots understanding frustration
- Educational tools adapting to student emotions

Ethical Considerations:

- Privacy concerns with text analysis
- Importance of transparency
- Age-appropriate design

Assessment & Evaluation

Formative Assessment:

- Participation in hands-on activities
- Quality of word list additions
- Understanding during group discussion
- Ability to explain the process to peers

Success Criteria:

- Can explain what text classification is
- Successfully modifies the word lists
- Understands why safety filtering is important
- Can identify at least one real-world application

Differentiation Strategies

For Beginners:

- Focus on modifying word lists only
- Provide pre-written sentences to test
- Work in pairs for the coding activities

For Advanced Students:

- Add more emotion categories (excited, bored, worried)
- Implement punctuation analysis (! and ?)
- Create a "confidence score" display

Vocabulary Introduced

- Sentiment Analysis: Computer understanding of emotions in text
- Rule-Based System: Using if-then rules instead of machine learning
- Text Classification: Categorizing text into groups
- Content Filtering: Removing inappropriate material
- Natural Language Processing: Computers understanding human language

Extension Activities

For Further Learning:

- Research how machine learning improves text classification
- Design a mood tracker that saves daily emotions
- Create a "compliment generator" using positive word banks
- Explore how emojis themselves convey emotion

Safety Notes

- All content is age-appropriate and educational
- No personal data collection or storage
- Focus on positive reinforcement
- Supervise student testing with various sentences

Learning Outcomes

Technical Skills:

- Basic understanding of text processing
- Simple Python modification skills
- Web app interaction experience

Critical Thinking:

- Understanding AI limitations
- Evaluating technology ethics
- Problem-solving through iteration

Digital Literacy:

- Safe online interaction awareness
- Understanding how apps work
- Responsible technology use

Preparation Notes for Teacher

- Pre-deploy the Mood2Emoji app on Streamlit Cloud
- Test all example sentences beforehand
- Prepare backup computers in case of technical issues
- Have printed instructions for the coding activities
- Prepare discussion prompts on whiteboard or slides

Troubleshooting Tips:

- If app is slow, use local installation as backup
- If students struggle with coding, provide completed examples
- Have extension activities ready for fast finishers