Lesson Plan: Build a Mood2Emoji App

Age Group: 12-16 years

Duration: 60 minutes

Difficulty Level: Beginner-friendly

Tools Needed:

- Laptop or computer with internet access
- Python (installed or via Google Colab)
- TextBlob library or simple rule-based classifier
- Streamlit (optional for UI)

Lesson Goals

By the end of the session, students will:

- Understand text classification and sentiment analysis.
- Learn how computers interpret emotions in text.
- Build an app that converts sentences into mood-based emojis.
- Gain exposure to how AI models make predictions.

Topics Introduced

---

1. Natural Language Processing (NLP)
2. Sentiment Analysis / Text Classification
3. Python Libraries for NLP (TextBlob)
4. Mapping Emotions to Emojis
5. Building a Simple App Interface (Streamlit)

Topics in Detail

---

1. Introduction to NLP (10 min)

How do machines understand language? Introduce NLP with examples like Siri or ChatGPT.

Explain how computers read and respond to human language.

2. Sentiment Analysis (10 min)

Introduce positive, neutral, and negative tones.

Example:

- "I love this game!" -> Positive
- "This is okay." -> Neutral
- "I hate homework." -> Negative

3. Basic Coding Concepts (10 min)

Introduce Python basics (strings, functions, if-else) and TextBlob usage.

Example:

```
from textblob import TextBlob
text = TextBlob("I love this!")
print(text.sentiment)
```

4. Mapping Mood -> Emoji (10 min)

Rule-based approach:

```
def mood_to_emoji(sentence):
    blob = TextBlob(sentence)
    polarity = blob.sentiment.polarity
    if polarity > 0.1:
        return "Happy"
    elif polarity < -0.1:
        return "Sad"
    else:
        return "Neutral"
```

5. Building the Mood2Emoji App (15 min)

```
import streamlit as st
```

```
from textblob import TextBlob
st.title("Mood2Emoji App")
text = st.text_input("Enter a sentence:")
if text:
    blob = TextBlob(text)
    polarity = blob.sentiment.polarity
    if polarity > 0.1:
        st.write("Happy!")
    elif polarity < -0.1:
        st.write("Sad!")
    else:
        st.write("Neutral.")
```

Activity Explanation (15 min)

---

1. Students type sample sentences (e.g., "I'm excited for the trip!").
2. The app processes the text and shows an emoji with an explanation.
3. Discuss results and adjust thresholds.
4. (Optional) Add filtering for inappropriate words.

Learning Outcomes

Students will be able to:

- Explain NLP and sentiment analysis.
- Write Python code to classify emotions.
- Build and run an interactive Mood2Emoji app.
- Understand how AI interprets emotions safely and creatively.

Extension Ideas

- Add more emojis for emotions like Angry, Love, Shocked.
- Create a "Mood Tracker" that stores daily moods.
- Build a teacher dashboard showing class sentiment trends.