Session 4 — Sentiment Showdown

Space Builder: Sentiment Showdown

What You’ll Build

A Space that sends the same text to three different sentiment models — one trained on movie reviews, one on tweets, one on product reviews — and shows you how they disagree.

Step 1 — Create a New Space

1. Go to huggingface.co/new-space (log in if needed)
2. In the "Space name" field, type: sentiment-showdown
3. Under "Select the Space SDK", choose Gradio
4. Under "Select the Space hardware", choose Free — CPU basic
5. Leave everything else as default
6. Click Create Space

Step 2 — Create the requirements.txt File

1. On your Space page, click the Files tab (near the top)
2. Click Add file → Create a new file
3. In the filename field at the top, type: requirements.txt
4. In the big text area below, copy and paste this exactly:

> *Your Space page will open. It’s empty right now — we’re about to add the code.*

5. Click Commit new file to main (the blue button at the bottom)

Step 3 — Create the app.py File

1. Click Add file → Create a new file (again)
2. In the filename field, type: app.py
3. In the big text area, copy and paste ALL of the code below:

```
transformers
torch
gradio
```

4. Click Commit new file to main

Step 4 — Wait for It to Build

1. Click the App tab (at the top of your Space page)

2. You’ll see "Building" — this takes 2–5 minutes

3. When it’s done, your Space will appear!

Step 5 — Try It Out!

1. Try sarcasm — which model handles it best?

2. Type something genuinely neutral — do all three agree?

3. Try a restaurant review, then a tweet, then a movie review — each model should do best on its own domain

Troubleshooting

```
from transformers import pipeline
import gradio as gr

# Load all 3 sentiment models at startup (sequential to manage memory)
print("Loading Movie Review model...")
model_movie = pipeline(
    "sentiment-analysis",
    model="distilbert-base-uncased-finetuned-sst-2-english",
)

print("Loading Twitter model...")
model_twitter = pipeline(
    "sentiment-analysis",
    model="cardiffnlp/twitter-roberta-base-sentiment-latest",
)

print("Loading Product Review model...")
model_product = pipeline(
    "sentiment-analysis",
    model="nlptown/bert-base-multilingual-uncased-sentiment",
)

print("All models loaded!")

def format_result(result):
    """Format a single model result as 'LABEL (XX% confidence)'."""
    label = result["label"]
    score = result["score"]
    return f"{label} ({score:.0%} confidence)"

def compare_sentiment(text):
    if not text or not text.strip():
        return "Enter some text first!", "Enter some text first!", "Enter some text first!"

    # Truncate to 512 chars (model token limits)
    text = text[:512]

    r1 = model_movie(text)[0]
    r2 = model_twitter(text)[0]
    r3 = model_product(text)[0]

    return format_result(r1), format_result(r2), format_result(r3)

demo = gr.Interface(
    fn=compare_sentiment,
    inputs=gr.Textbox(
        lines=5,
        placeholder="Type or paste text to analyze...",
        label="Text to Analyze",
    ),
    outputs=[
        gr.Textbox(label="Movie Review Model (distilbert-sst2)"),
        gr.Textbox(label="Twitter Model (cardiffnlp)"),
        gr.Textbox(label="Product Review Model (nlptown, 1-5 stars)"),
    ],
    title="Sentiment Showdown",
    description="Three AI models read the same text. Do they agree? Each was trained on different data — movie reviews, tweets, and product reviews — so they see the world differently.",
    examples=[
        ["The service was slow but the food was amazing."],
        ["I can't believe how terrible this is. Just kidding, it's great!"],
        ["The movie was fine. Nothing special but not bad either."],
        ["lol this is SO bad it's actually good"],
        ["The product arrived on time and works as described."],
    ],
)

demo.launch()
```

```
Make sure you copy the ENTIRE code block — from the very first line to the very last. Missing even one line can cause errors.
```

```
If you see a red error: Click the Logs tab to read the error message. The most common fix is to double-check that requirements.txt has the right contents and that app.py was copied completely.
```

Problem	Fix
"Runtime error"	Check the Logs tab. Usually means a typo in app.py. Re-copy the code carefully.
Space stuck on "Building"	Wait up to 5 minutes. Free CPU Spaces can be slow. If it’s been more than 10 minutes, try deleting the Space and starting over.
"ModuleNotFoundError"	Your requirements.txt is missing a library. Make sure it matches exactly what’s shown above.
Space loads but nothing happens	Make sure the last line of app.py is demo.launch() with no extra spaces before it.