Session 9 — Make It Useful

Space Builder: Restaurant Review Analyzer

What You'll Build

The same sentiment model from Session 4, but wrapped in a completely different presentation. Instead of showing "POSITIVE 92%", it shows "Happy Customer" with actionable advice for restaurant owners. Same AI, better design.

Step 1 — Create a New Space

1. Go to huggingface.co/new-space (log in if needed)
2. In the "Space name" field, type: restaurant-review-analyzer
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. Paste a real restaurant review from Google or Yelp
2. Write a mixed review — does the advice make sense?
3. Compare: type the same text into your Session 4 Sentiment Showdown — same model, totally different output!

Troubleshooting

```
import gradio as gr
from transformers import pipeline

# Load the same sentiment model students already know —
# the magic is in how we PRESENT the results, not in swapping models.
analyzer = pipeline(
    "sentiment-analysis",
    model="distilbert-base-uncased-finetuned-sst-2-english",
)

def analyze_review(review):
    """Turn raw sentiment output into advice a restaurant owner can act on."""
    if not review or not review.strip():
        return "Paste a review above to analyze!"

    result = analyzer(review[:512])[0]
    label = result["label"]
    score = result["score"]

    if label == "POSITIVE":
        mood = "Happy Customer \ud83d\ude0a"
        if score > 0.95:
            advice = (
                "This customer loved the experience! "
                "Consider thanking them publicly and inviting them back."
            )
        else:
            advice = (
                "This customer had a good experience. "
                "A simple thank-you response goes a long way."
            )
    else:
        mood = "Unhappy Customer \ud83d\ude1f"
        if score > 0.95:
            advice = (
                "This customer had a very poor experience. "
                "Reach out personally to understand what went wrong."
            )
        else:
            advice = (
                "This customer had a mixed-to-negative experience. "
                "Consider responding with an apology and an offer to make it right."
            )

    confidence_bar = "\u2588" * int(score * 20) + "\u2591" * (20 - int(score * 20))
    return (
        f"Overall Impression:  {mood}\n"
        f"Confidence: {confidence_bar} {score:.0%}\n"
        f"\nSuggested Action:\n{advice}"
    )
```

```
demo = gr.Interface(
    fn=analyze_review,
    inputs=gr.Textbox(
        lines=5,
        placeholder="Paste a restaurant review here...",
        label="Customer Review",
    ),
    outputs=gr.Textbox(label="Analysis", lines=6),
    title="Restaurant Review Analyzer",
    description=(
        "Paste a customer review and get instant sentiment analysis. "
        "Built for restaurant owners who want to quickly understand "
        "customer feedback and know how to respond."
    ),
    examples=[
        [
            "The pasta was absolutely divine and the service was impeccable. "
            "Will definitely return!"
        ],
        [
            "Waited 45 minutes for cold food. The waiter was rude and "
            "unapologetic. Never again."
        ],
        [
            "Food was okay, nothing special. Decent portion sizes but "
            "overpriced for what you get."
        ],
        [
            "Great ambiance and the appetizers were fantastic, but the "
            "main course was underwhelming. Dessert saved the meal."
        ],
    ],
)

demo.launch()
```

```
Make sure you copy the ENTIRE code block — from the very first line to the very last. Missing e
  ven 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 complet
  ely.
```

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.