SESSION 10 REFERENCE GUIDE

Session 10 — Build Your Own from Scratch

AI + Research Level 2 — Supplementary Material

Concept: SUPERVISED LEARNING AND TASK DESIGN

Key idea: Students choose a task, pick a model, and build a complete Space from scratch. The act of choosing — *what task? what model? what audience?* — teaches how supervised learning works: someone decided what the labels should be, what the training data looked like, and what "correct" means.

Time Breakdown (2 hours)

0:00–0:15 — "You've Built 5 Spaces With Me. Now You Build Your Own."

Quick recap of every Space so far:

Session	Space	Task Type
1	Silly Phrase Finder	Zero-shot classification
2	Model Swap	Sentiment analysis
3	Break It on Purpose	Data cleaning
4	Sentiment Showdown	Model evaluation
5	Slider Space	Hyperparameters
6	Domain Shift Tester	Overfitting
7	Bias Detective	Fairness
8	Two-Model Pipeline	Multi-model systems
9	Restaurant Review Analyzer	UX/prompt engineering

> *"Every one of these used a model someone else trained. Today you pick the model, pick the task, and build the whole thing."*

Live demo: Browsing the Hub

1. Go to huggingface.co/models
2. Show the task filter (left sidebar)
3. Click "Text Classification" — point out the download counts
4. Click on a model card — show: task, training data, labels, size
5. Show how to check if it's free-CPU-compatible (look at model size)

0:15–0:40 — Students Browse and Pitch

Each student spends 10 minutes browsing, then pitches in 1–2 sentences:

- "I want to build a _ **that uses** ___ model."

Instructor helps scope:
- Too ambitious? → "Start with just text classification, add features later."
- Can't find a model? → Show the pre-screened list below.
- Stuck on ideas? → Hand them a rescue template from `templates/`.

0:40–1:30 — Build Time (50 minutes)

Cycling through 5–6 students (~8 min each):

Minute	Focus
0–2	Check their model choice. Does it load? Is it CPU-compatible?
2–4	Help them get the basic pipeline working (`pipeline("task", model="...")`)
4–6	Help with the Gradio interface — inputs, outputs, title, description
6–8	Test with one example. Does it produce reasonable output?

Rescue plan: If a student can't get their model working after 10 minutes, hand them a rescue template:
- `templates/text-classifier.py` — emotion detection (always works)
- `templates/text-generator.py` — text generation with temperature
- `templates/zero-shot.py` — zero-shot classification (most flexible)
- `templates/summarizer.py` — summarization

Common issues and fixes:

Problem	Fix
Model too large, Space crashes	Switch to a `distil-` variant
Pipeline task name wrong	Check model card for correct task
Output is gibberish	Check `max_length`, `temperature` settings
Space won't build	Check `requirements.txt` has all dependencies
Gradio error	Make sure function returns a string

1:30–1:50 — Test and Iterate

Students test their own Spaces:
- Try 5 different inputs
- Find one that breaks or gives a weird result
- Fix it or add a note about limitations

Peer quick-test: Each student tries the Space of the person next to them (or the person before/after in the Zoom gallery).

1:30–1:40 — Notebook Time

Share the Colab notebook link in the Zoom chat. Students open it and start experimenting with their chosen model.

What they do:

- Run the setup cell
- Uncomment the pipeline line that matches their model choice (or write their own)
- Test with 5 different inputs
- Fill in the "Plan your Space" section

Instructor role: Help anyone whose model isn't loading. If a student can't decide, point them to the pre-screened list in the notebook. The notebook has commented-out lines for each model — they just uncomment one.

GitHub skill: Show how to download the notebook from Colab and upload it to their `my-ai-portfolio` repo on GitHub. This is their first time pushing a notebook they wrote.

1:50–2:00 — Name the Concept

> "*You just did something specific: you picked a task, picked a model trained on labeled data, and designed an interface around it. That's* ***supervised learning*** *— and the choices you made (what labels? what data? what task?) are* ***task design****.*"

Talking points:

- Every model you used today was trained on data where humans provided the "right answers" — that's supervised learning
- The person who created the training data made choices: what counts as "positive"? What emotions to include? How to label translations?
- When you picked a model, you inherited those choices
- Task design = deciding what question to ask and what answers are possible

Concept card:

Concept	What It Means	What We Did
Supervised Learning	Training a model on labeled examples (input → correct output)	Used models trained on human-labeled datasets
Task Design	Choosing what question to ask and what categories/outputs to support	Picked a task type, chose a model, designed the interface

Pre-Screened Models for Free CPU

Text Classification

Model	Task	Labels	Size
`distilbert-base-uncased-finetuned-sst-2-english`	Sentiment	POSITIVE / NEGATIVE	260 MB
`j-hartmann/emotion-english-distilroberta-base`	Emotion	anger, disgust, fear, joy, neutral, sadness, surprise	330 MB
`cardiffnlp/twitter-roberta-base-sentiment-latest`	Sentiment	negative, neutral, positive	500 MB
`SamLowe/roberta-base-go_emotions`	Emotion	28 emotions	500 MB

Zero-Shot Classification

Model	Notes	Size
`valhalla/distilbart-mnli-12-3`	Fast, good quality	890 MB
`facebook/bart-large-mnli`	Better quality, slower	1.6 GB

Text Generation

Model	Notes	Size
`distilgpt2`	Fast, short outputs	330 MB

Summarization

Model	Notes	Size
`sshleifer/distilbart-cnn-12-6`	Good for news/articles	1.2 GB

Named Entity Recognition (NER)

Model	Entities	Size
`dslim/bert-base-NER`	Person, Org, Location, Misc	430 MB

Translation

Model	Direction	Size
`Helsinki-NLP/opus-mt-en-fr`	English → French	300 MB
`Helsinki-NLP/opus-mt-en-es`	English → Spanish	300 MB
`Helsinki-NLP/opus-mt-en-de`	English → German	300 MB
`Helsinki-NLP/opus-mt-fr-en`	French → English	300 MB

Image Classification (advanced — needs `Pillow` in requirements)

Model	Labels	Size
`google/vit-base-patch16-224`	1000 ImageNet classes	350 MB

Image Captioning (advanced — needs `Pillow` in requirements)

Model	Notes	Size
`Salesforce/blip-image-captioning-base`	Generates captions for photos	990 MB

Models to AVOID (too large for free CPU)

- Anything with `large`, `xl`, `xxl` in the name (except bart-large-mnli which barely fits)
- `meta-llama/*` — all Llama models
- `mistralai/*` — all Mistral models
- `tiiuae/falcon-*` — all Falcon models
- `bigscience/bloom-*` — all BLOOM models
- Any model over 2 GB

Hub Browsing Tips (share with students)

1. **Filter by task** — Use the left sidebar on huggingface.co/models
2. **Sort by downloads** — Popular models are popular for a reason (they work)
3. **Check the model card** — Look for: task description, training data, labels, model size
4. **Look at Spaces using this model** — Scroll down on the model page to see examples
5. **Test before building** — Many model pages have an "Inference API" widget. Try it first!

Pre-Session Prep

- [ ] Deploy the starter template as a Space (students will duplicate it)
- [ ] Test all four rescue templates — make sure they run on free CPU
- [ ] Have the pre-screened model list ready to screen-share
- [ ] Test Hub browsing — make sure the task filters work as expected
- [ ] Pre-load a few model cards to show if the network is slow

Instructor Tips

- **Let them struggle (a little).** The point is choosing and building, not getting it perfect.
- **The rescue templates are not failure.** Frame them as "starting points everyone can use."
- **Image models are harder.** If a student wants image classification/captioning, help them add `Pillow` to requirements.txt and use `gr.Image()` input — but only if they're comfortable.
- **"It doesn't work" is the lesson.** When a model doesn't do what they expected, that's task design in action. The model was trained for a specific task — using it for something else won't work.