

Homework — AI & ML Foundations

Estimated time: ~45 min · No code you have to run · A browser, a webcam, and a notebook

Three tool-tour exercises. The point is to put your hands on three very different faces of machine learning: *training* a tiny model yourself in a browser, *using* a deployed ML model that is already shaping what you see every day, and *reading* the raw material a real ML problem actually starts from — a dataset. By the end you'll have felt all three sides of the train / infer / data loop from Lesson 12's theory.

Before you start

Keep a one-page notebook open (paper or a text file). For each task, write the short answers under (a)/(b)/(c). You are not turning this in — the goal is to build the muscle of *poking at* an ML system instead of just reading about one. There is **no code you need to run** in this homework; Task 1 trains a model with mouse clicks and your webcam.

The three tasks

Task 1 — Train a model in your browser (Teachable Machine)

Open teachablemachine.withgoogle.com. Click **Get Started → Image Project → Standard image model**. You'll see two empty class boxes (*Class 1* and *Class 2*).

Rename them to any pair of webcam-friendly opposites — e.g. "*with glasses*" / "*without glasses*", or "*thumbs up*" / "*thumbs down*", or "*pen visible*" / "*no pen*".

For each class, hold down the **Webcam** button to record ~30 sample frames while you act out that class (move slightly, change angle a bit — you want variety, not 30 identical frames).

Click **Train Model**. It takes ~30 seconds — that is a real training run, just on a very small model. When it's done, the **Preview** panel comes alive and shows two prediction bars updating in real time as you move in front of the webcam.

Now play. Hold up each scenario in turn, then deliberately hold up *something the model has never seen* (an unrelated object, your other hand, your coffee mug). Watch what the prediction bars do.

Self-check — answer all three:

- **(a)** Which class scored highest when you held up the "correct" sign for it? Roughly what confidence percentage?
- **(b)** What happened when you held up something the model had never seen? (Hint: it still picked one of the two classes — the model has no concept of "*neither*" unless you teach it one. This is a foundational ML behaviour worth feeling once.)
- **(c)** What would you change to make the model less confused? (Collect more examples per class? More *varied* examples — different lighting, different angles? Add a third class called "*neither*" and train on random background frames?) Take a screenshot of the prediction bars while you're holding up the "unknown" object.

✓ You have now run the full ML loop: collected labelled data, trained a model on it, and used the trained model for inference — all without writing a line of code.

Task 2 — See a deployed ML model recommend things to you

Open **Spotify**, **Netflix**, or **YouTube** — any logged-in account where the home page is personalised to you. (If you don't have a personalised account, ask a friend or housemate to let you peek at theirs for two minutes; the exercise needs *their* recommendations, not a fresh anonymous session.)

Scroll the home page slowly. You are looking at the output of an ML recommender system that has been quietly observing your behaviour for months or years — what you played, what you skipped, what you finished, what other people with similar taste enjoyed. Every row title is a hint at what signal it is using.

Self-check — answer all three:

- **(a)** Name *three* sections of the home page whose content was clearly picked for you personally (examples — Spotify: *Made For You*, *Daily Mix*, *Discover Weekly*, *Your Top Mixes*; Netflix: *Top Picks for [Your Name]*, *Because You Watched X*, *Continue Watching*; YouTube: the main feed, *Recommended*, *Mixes*).
- **(b)** Pick one specific item shown and write one sentence on why you think the recommender chose it for you. (A song by an artist you played last week? A film in a genre you finished three of? A video from a channel you subscribed to?)
- **(c)** Find one section on the same page that is *not* personalised — usually a globally trending or editorial row (e.g. *Top 10 in your country today*, *Trending Now*, *New Releases*). Write one sentence on the difference: what is the recommender doing in the personalised row that it deliberately is *not* doing in this global row?

What you just saw

This is one of the biggest ML deployments in the world by sheer number of inference calls per second — and it has been running long before "AI" was a press headline. Recommender systems were arguably the first ML technology most people on Earth used daily without noticing.

Task 3 — Read a Kaggle dataset description (don't write any code)

Open kaggle.com/datasets. You will see a feed of public datasets uploaded by the community. Browse for a minute, then pick a **tabular** dataset (rows and columns — not images, not text). Three that work especially well for this exercise:

- *Titanic — Machine Learning from Disaster*
- *House Prices — Advanced Regression Techniques*
- *Credit Card Fraud Detection*

On the dataset's page, open the **Data** tab (or scroll to the data card). You'll see a list of columns with their names and types — this is what a real ML problem looks like before any model touches it. Read the dataset description above the column list; one column will usually be flagged as the thing everyone tries to predict.

Self-check — answer all four:

- **(a)** What is the *target* — the single column a model would predict? (Titanic: `Survived` . House Prices: `SalePrice` . Credit Card Fraud: `Class` .)
- **(b)** What are the *inputs*? Name three of the other columns and, in plain language, what they describe.
- **(c)** Is this a *supervised* or *unsupervised* problem? (Hint from Lesson 12: if you can point at a target column, it's supervised.)
- **(d)** Click the **Code** tab at the top of the dataset and skim the count of public notebooks people have uploaded for this dataset. You do **not** need to read any of them — just notice that hundreds or thousands of people have already attempted this exact problem, and that this is what a healthy ML community looks like.

Don't get pulled in

Kaggle is fascinating and competitive and there are leaderboards and prize money. For *this* exercise the only goal is to see the **shape** of an ML problem — columns, a target, examples of solutions existing. Resist the urge to open ten notebooks; close the tab when you can answer (a)–(d).

What "done" looks like

You're done when, in your notebook, you have:

- two prediction-bar observations and one screenshot from Task 1 (training)
- three section names, one why-it-picked-this sentence, and one personalised-vs-global sentence from Task 2 (inference at scale)

- one target column, three input columns, one supervised/unsupervised verdict, and one observation of the Code tab from Task 3 (data)

The aim was never to make you an ML engineer in 45 minutes. It was to make sure that next time you read "we trained a model on user data and deployed it to production", you have a concrete picture in your head of *all three* nouns in that sentence — because you've now touched each of them with your own hands.

Where this goes next

Lesson 13 takes the same train / infer / data picture and zooms into the corner of ML that exploded in 2022: **LLMs** (Large Language Models), the agents built on top of them, and the product landscape they created. Everything you just felt — a model learning a pattern from data, then using that pattern on new inputs — still applies; the model just gets very, very much bigger, and the data is the entire internet.