

Welcome to the Lab

The Data Detective & Meet the Machines

Day 01 · Week 1

Setup day · meet your (locked) AI Studio

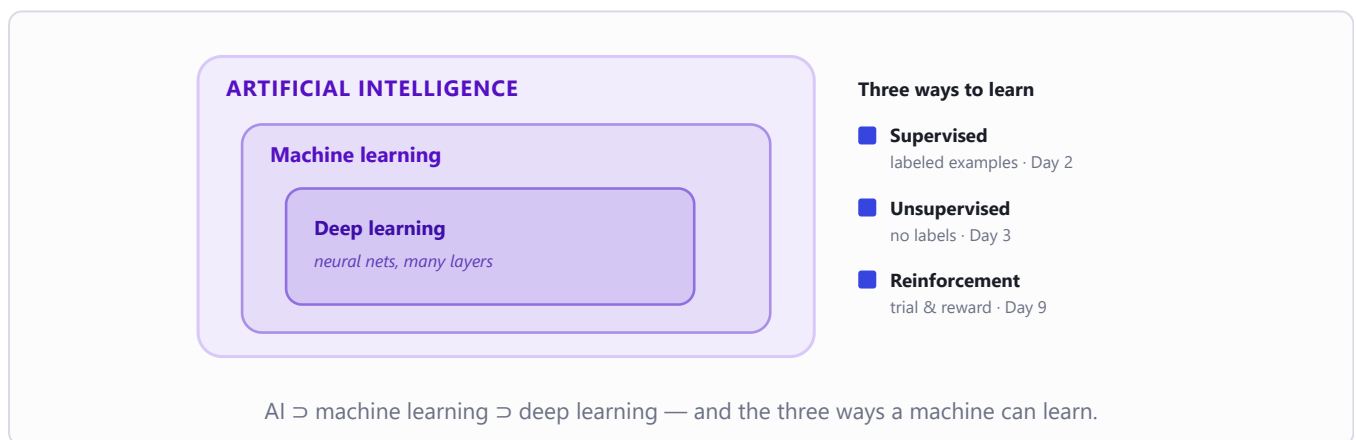
1 Mission briefing

Welcome, recruit. Before we hand you a neural network, prove you can **read data** — every AI system is built on it, and a researcher who can't read their data ships broken models. This morning a crate of penguin measurements landed from an Antarctic field station: interrogate it. Then, as a reward, you'll play with four **finished** AI machines — the very ones you'll rebuild yourself over the next two weeks.

- ☐ Load a real dataset with **pandas** and size it up
- ☐ **Filter, group, count** to answer real questions
- ☐ See the **map of AI** (ML, deep learning, superpowers)
- ☐ Meet the **AI Studio** you'll fill one tab per day
- ☐ Hunt down **missing values** and clean the data
- ☐ **Plot** the data and spot a learnable pattern
- ☐ Run four **pretrained** models — and try to fool them

2 Field guide the concepts

AI, machine learning, deep learning aren't synonyms — they nest like Russian dolls. **Artificial intelligence** is any program we'd call smart; **machine learning** is the slice that isn't hand-coded but *learns patterns from data*; **deep learning** is machine learning with neural networks stacked many layers deep. There are three ways to learn: with labels (**supervised**), without (**unsupervised**), or by reward (**reinforcement**).



How ML data is shaped. Almost all of it is a table: one **row** per example (one penguin), one **column** per **feature** you measured (bill length, mass). Sometimes one column is the **label** — the answer you'll teach a machine to predict. Today's demos also use **pretrained** models: giant networks someone else trained, that you simply borrow.

The Lab's motto: understand your data *before* you model it.

A precise reading

This course focuses on learning from data. AI also includes rule-based and search systems. Tables are a useful starting point, but images, audio and text need representations suited to their structure.

3 Lab missions check each off in the notebooks

NOTEBOOK 1A • THE DATA DETECTIVE

1 Count the gaps ☐

Tally missing values per column with `penguins.isna().sum()`.

Expect: 19 missing values in all — a few measurements, about a dozen unknown sexes.

2 The heavyweights ☐

Filter for penguins heavier than 5000 g.

Expect: about 60 birds — and nearly all the same species. Remember which.

3 The average penguin, by species ☐

`groupby("species").mean(numeric_only=True)`.

Expect: one row per species, Gentoo biggest on almost every measurement.

4 Who lives where? ☐

Count penguins per island with `value_counts()`.

Expect: Biscoe, Dream, Torgersen — Biscoe hosts the most.

5 The scatter plot that changes everything ☐

Plot bill length vs bill depth, one color per species.

Expect: three fairly distinct blobs — structure a machine could learn.

6 What moves together? ☐

Correlation matrix of the four measurements, drawn as a heatmap.

Expect: flipper length → body mass is the brightest pair, about +0.87.

7 Close the case ☐

Answer three questions with one line of pandas each.

Expect: Gentoo (heaviest), Biscoe (most Gentoo), 195.8 mm (Chinstrap flipper).

NOTEBOOK 1B • MEET THE MACHINES

1 Fool the mood reader ☐

Write a sarcastic line that *sounds* happy but means the opposite.

Goal: get the sentiment model to answer POSITIVE for a clear complaint.

2 Be the author ☐

Give GPT-2 your own opening line and generate.

Expect: a different story each run (sampling is random).

3 Invent the categories ☐

Hand the zero-shot sorter your own text and labels.

Try: a sentence that could belong to two categories at once.

4 Cheat sheet the detective's toolkit

<code>pd.read_csv(path)</code>	Load a CSV file into a DataFrame
<code>df.head()</code> · <code>df.info()</code> · <code>df.describe()</code>	Peek at rows · column types & nulls · number stats
<code>df.isna().sum()</code>	Count missing values in each column
<code>df.dropna()</code>	Drop every row that has a missing value
<code>df[df["body_mass_g"] > 5000]</code>	Keep only the rows matching a condition
<code>df.groupby("species").mean(numeric_only=True)</code>	Average each numeric column per group
<code>df["island"].value_counts()</code>	Tally how often each value appears
<code>df[cols].corr()</code>	Correlation matrix (−1 ... +1) of the columns
<code>plt.scatter(x, y, label=name)</code>	One dot per row; loop species to color them
<code>pipeline("text-classification", model=...)</code>	Load & run a pretrained model in one line

No module today — the padlock wall is the mission

Peek at the app you're about to build. From the repo root run:

```
$ python ai_studio/app.py
```

Every tab shows  — you haven't built anything yet. One notebook unlocks one tab: **Day 2** →  Prediction Machine, **Day 3** →  Pattern Finder, **Day 5** →  Digit Reader, and on to Demo Day, when every lock is gone.

6 Mini-glossary

dataset

A table you learn from — rows are examples, columns are what you measured.

label

The answer you want to predict (e.g. species). Present in supervised data.

pretrained model

A network already trained on huge data that you borrow as-is.

feature

One input measurement (a column), e.g. bill length.

DataFrame

pandas' programmable spreadsheet — the `df` object.

notebook kernel

The live Python process running your cells; restarting it wipes memory.

7 Show & tell

1. Share **one plot** you made and the single **insight** it revealed.
2. Which AI **superpower** impressed you most — and which one did you manage to **fool**?
3. What's the one **burning question** about AI you want answered by Demo Day?