ARTIFICIAL INTELLIGENCE AND NEURAL NETWORKS

Lecture 2a – Data Workflows for AI Systems

Chizhi Chris ZHANG

zhangchizhi@ciomp.ac.cn

Advanced Computing and Digital Technology Research Center

University of Chinese Academy of Sciences

Spring 2026

Today's Question

Main question

Why do many AI systems look clever in a demo and then disappoint once they meet real data?

One direct answer

Because the model never sees "reality" directly. It sees only rows, fields, labels, and timestamps that people decided to record.

What changes in this lecture

In AI1 we asked what kinds of tasks AI can do. In AI2 we ask what kind of data story makes those tasks trustworthy in the first place.

From Form to Table

Name	Value
Name	
Sex	○ Male ◉ Female
Eye color	green
Check all that apply	☐ Over 6 feet tall ☐ Over 200 pounds
Describe your athletic ability:	
Enter my information	

	Person 1	Person 2	Person 3	Person 4	Person 5	Person 6	Person 7	Person 8	Person 9	Person 10	presence in positive vs negative cancer results	
Suspect Gene	Present	Present	Present	Present	Absent	Absent	Present	Present	Present	Absent	5/5	2/5
Partner Gene A	Present	Present	Absent	Absent	Present	Present	Absent	Present	Absent	Present	3/5	3/5
Partner Gene B	Present	Absent	Present	Absent	Absent	Present	Absent	Absent	Absent	Present	1/5	3/5
Partner Gene C	Absent	Absent	Present	Present	Present	Absent	Present	Present	Absent	Present	3/5	3/5
Develops Cancer	Yes	Yes	No	Yes	No	No	Yes	Yes	No	No		

What changed between left and right

A real situation became fields, categories, and a target column. That translation already decides what the model will be able to notice later.

The part people often skip too fast

Anything not asked, not recorded, or badly coded on the left never appears on the right.

From AI1 to AI2

What we were doing in AI1

Last time we talked about AI as a family of tasks: classify, predict, recommend, detect, generate.

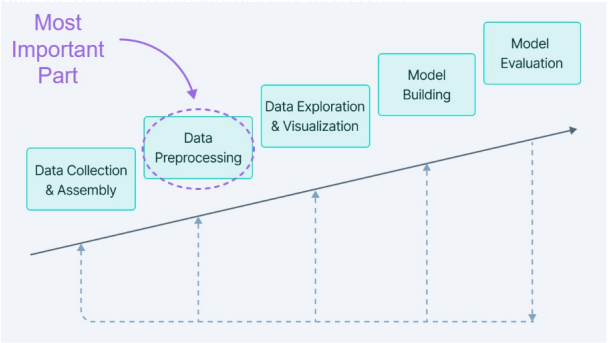
What changes in AI2

Those tasks only work well when the examples, labels, and evaluation data actually match the decision we care about.

The shift in attention

We are moving from “What can AI do?” to “What exactly is teaching it?”

The Data Path Matters



What students usually look at first. The model, the loss curve, and the final metric.

What is already shaping the result

Collection, cleaning, labeling, versioning, splitting, revisiting, and checking all shape the final behavior before tuning even starts.

Why Label Ambiguity Spreads

Why labels are harder than they look

Before there is a label, someone decides what the categories mean, which edge cases count, and what to do with uncertainty.

A plain classroom example

If one staff member records “late”, another records “absent”, and a third leaves the field blank, the model later learns from three different meanings.

Why this spreads through the whole pipeline

A label is never just a word. It is a human decision frozen into data.

A Labeling Team Story

One common situation

Three people label student support cases. One marks a case as “urgent” because the message sounds anxious. Another uses “urgent” only for safety risk. A third avoids the tag unless a supervisor confirms it.

What the model sees

Not three nuanced judgments, but one mixed column that quietly combines different personal rules into one training signal.

Why this matters

If the teaching signal is inconsistent, the model can look unstable later even when the code is perfectly clean.

Where Trouble Really Begins

Three quiet places where trouble starts

- the records were created for billing, logging, or compliance rather than for learning
- the model is trained on a frozen past but deployed into a moving present
- people may hide, exaggerate, or simplify information, and the dataset inherits that distortion

Why later cleaning cannot fully save this

If these problems enter at the source, later cleaning is often repair work rather than real control.

A station-crowding example

A crowd-prediction dataset may come from security turnstiles, be collected during an unusual semester, and reflect the fact that some students forget to scan on the way back in.

What Rows Never Tell You

What rows usually do not tell you

- why a behavior changed
- what was never recorded at all
- whether a policy, interface, or schedule changed upstream

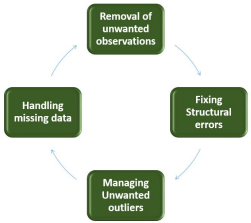
A classroom-friendly case

Restaurant ratings collected from voluntary online reviews do not describe the full customer base. They mostly describe people motivated enough to post.

This missing context never comes back by itself

Cleaning can repair many things, but it cannot recover context that was never collected.

Cleaning Is Model Preparation



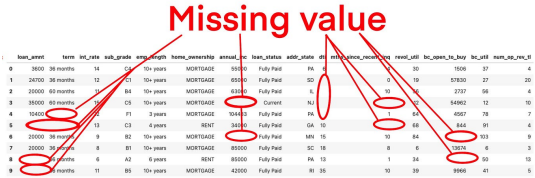
What cleaning really does

It decides what the model is allowed to treat as signal, what gets removed as noise, and what ambiguities are silently resolved before training starts.

A concrete example

If the same course appears as “AI”, “A.I.”, and “Artificial Intelligence”, the model sees three things until someone resolves the naming.

Missing Data Is a Decision Problem



Typical options

- drop incomplete rows
- fill with a statistic
- model the missing value
- add a missingness indicator

Do not treat every blank as the same kind of blank

Missingness itself can be meaningful. In medicine or finance, the absence of a measurement may already tell us something important.

Duplicates, Units, and Strange Rows

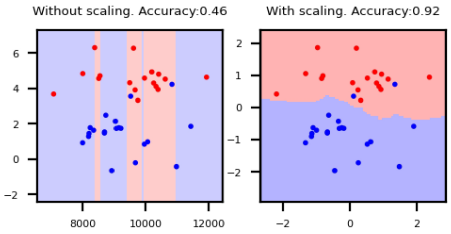
Three routine checks

- the same event may appear twice after system merges
- meters, centimeters, yuan, dollars, or percentages can silently mix
- some strange rows are errors, while others are rare events that matter most

A simple unit mistake

A spending dataset that mixes yuan and dollars can make ordinary purchases look like dramatic outliers even when nothing unusual happened.

Outliers and Scale Need Judgment



Scale problem

Large-unit features can dominate optimization even when they are not the most meaningful features.

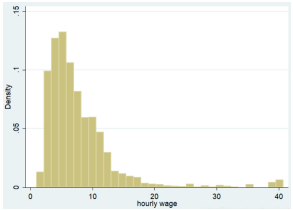
Outlier problem

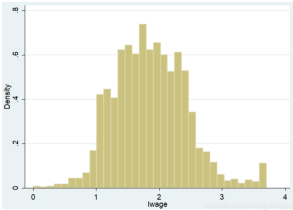
Some outliers are mistakes. Others are the exact rare events we care most about, such as fraud, faults, or emergencies.

The dangerous shortcut

Removing a “weird” point too quickly can destroy the most valuable part of the dataset.

Transformations Change Learnability





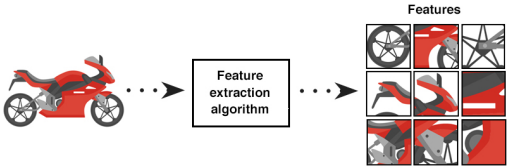
Before

Some raw variables are heavily skewed and hard for simple models to use well.

After a sensible transform

Reasonable transformations can reveal structure and make training more stable.

Features and Labels Decide What Can Be Learned



Two questions decide a lot

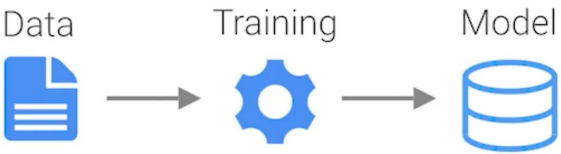
Feature question. What representation gives the model useful evidence instead of raw clutter?

Label question. Are the targets clear, consistent, and tied to the decision we really care about?

This is where many projects quietly go wrong

Weak labels do not just add noise. They teach the model the wrong boundary.

Data Shape Is Already a Choice



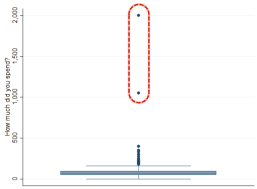
The table shape is already a modeling choice

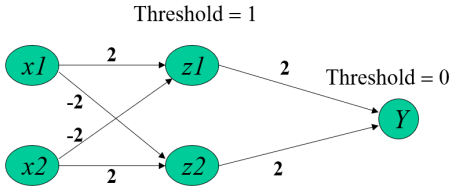
It decides which fields exist, which entities can be joined, how often data updates, and what historical trace can be recovered later.

A simple example

If attendance is stored only as a daily total, you can no longer ask whether the crowd peaked before lunch, after lunch, or only during one event.

A Small Case: Monthly Spending





Question

If a budgeting app wants to estimate next month's spending, which signals matter most: recent purchases, salary timing, rent, travel, or season?

What this case shows

The right features often improve the result more than switching from one fashionable algorithm to another.

Leakage Usually Arrives Quietly

Three common leakage paths

- future information sneaks into features that should only know the past
- preprocessing statistics are computed on the whole dataset before the split
- one feature is almost a disguised version of the target itself

A quiet leakage example

If you normalize a whole semester of student records before splitting train and test, the training pipeline has already peeked at the test distribution.

Split Discipline and Leakage

Training	Prediction
using	*answer*
data	*questions*

Why split data

We need one part for training, one for development decisions, and one for honest final evaluation.

Leakage warning

If future information slips into training features or preprocessing uses the whole dataset carelessly, the evaluation becomes flattering and untrustworthy.

A realistic failure

Many “great” AI systems collapse outside the lab because the test set was never truly isolated from earlier design choices.

Visual Geometry Still Helps

Simple structure still matters

Some datasets really do line up with a clear pattern, while others hide the useful signal inside combinations of fields, weak labels, or missing context.

Why this belongs in AI2

Before choosing a model, we should first ask whether the data representation preserves the real structure of the task or has already distorted it.

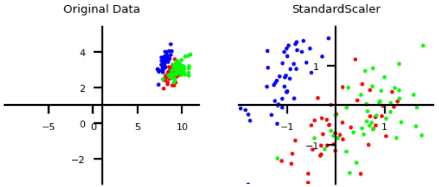
What not to do

Do not jump from “the data is messy” straight to “we need the biggest model available.” Sometimes the real fix is better collection, better labels, or better task definition.

When a Dataset Stops Matching Reality

Three ordinary kinds of drift

- **Market shift:** people behave differently after a policy change or a new competitor appears
- **Sensor shift:** instruments age, move, or get recalibrated
- **Platform shift:** an upstream software update silently changes formats or meanings



What to watch after deployment

Look for feature distribution shift, sudden error increases, subgroup gaps, and upstream source changes before you blame the optimizer.

What Teams Monitor After Launch

Useful questions after deployment

- Are inputs arriving in a different format from before?
- Are some user groups now getting worse outcomes?
- Has one important feature stopped updating correctly?
- Are humans overriding the model much more often than before?

Why this is part of AI, not maintenance trivia

Modern AI does not end when the model is trained. It stays trustworthy only if someone keeps watching whether the data story is still true.

Even Modern AI Starts With Data

What bigger models really change

They can reuse broader patterns, tolerate some noise, and reduce how much hand-built feature design is needed.

What they still cannot fix for free

They still inherit source bias, stale examples, weak labels, missing context, and dishonest evaluation.

A modern example

A chatbot may sound fluent and helpful while quietly repeating outdated facts, social bias, or feedback artifacts from the data that shaped it.

The simple reminder

More parameters do not mean more trustworthy data.

Bias Starts in the Data

Where bias usually comes from

Sampling bias, annotation bias, historical policy bias, and weak subgroup coverage can all enter long before training.

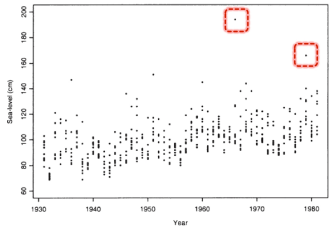
A familiar example

If a speech dataset mainly contains young urban voices, the system may look accurate overall while failing much more often on other speakers.

What careful readers check

They examine subgroup performance, not only one average score that hides unequal failure.

Mini Case: Environmental Data



Why this is a good classroom case

Environmental data is noisy, seasonal, incomplete, and affected by measurement conditions.

What this case shows

On this kind of data, cleaning strategy and split strategy can change credibility more than the algorithm family does.

When Cleaning Makes Things Worse

Three ways to overdo it

- **Over-cleaning:** rare but meaningful events get removed because they look strange
- **Over-smoothing:** transformations erase local signals that later matter for detection or diagnosis
- **Overconfidence:** neat preprocessing code makes people stop questioning the source itself

Why this page matters

The goal is not to make data look tidy. The goal is to keep the parts of reality that the task actually depends on.

What Healthy Data Judgment Looks Like

Three moments to stay awake

- before training: inspect sources, label logic, units, missingness, and subgroup coverage
- during learning: keep a simple reference, watch leakage, and compare across groups
- after deployment: watch for drift before blindly training longer or changing the model

The habit worth keeping

Good judgment does not appear only at the end when the metric is printed. It appears all the way through the pipeline.

A Simple Checklist Before Trust

Ask four things

- Do I understand where this dataset came from?
- Are the labels tied to the real decision?
- Is the evaluation split honest?
- Will this still make sense after the model is used?

Why even a general-course student should remember this

These questions are simple enough for a general-course student to remember, but strong enough to catch many serious failures early.

Who Still Matters When Models Grow

What larger models help with

They compress patterns from vast histories, reuse representations, and apply the same transformation at enormous scale.

What people still judge

People still decide whether the target makes sense, whether the error is acceptable, and whether the system should be trusted in that role at all.

What people still contribute

For a general-course student, the valuable skill is not out-computing the model. It is asking better questions about data, targets, and use.

Why AI2 and NN2 Are Paired

What AI2 has done

Data quality, feature geometry, and split discipline set the ceiling for every model family.

What NN2 adds

Now we can study perceptrons and linear boundaries on top of a cleaner understanding of what data makes possible.

Why Bigger Models Do Not Automatically Fix Bad Data

What a bigger model can do

fit subtler patterns, use richer nonlinear structure, and memorize complicated regularities

What that means on weak data

It can also fit noise, amplify imbalance, and make leakage look even more convincing.

A classroom translation

If the training set mostly reflects one unusual semester, a bigger model usually learns that one semester more perfectly. It does not magically infer the missing semesters.

Summary

- Many failures that look like model failures are actually data failures in disguise: weak sources, weak labels, leakage, or drift.
- No model can recover information that the data path never captured, and no honest metric survives a dishonest split.
- When a result looks impressive, ask where the data came from, what the labels really mean, and whether the evaluation was honest.

Next: One Boundary at a Time

Where we go next

Next we move inside the model and ask what a single-layer neural unit can actually compute once the data has been prepared honestly.

Keep this question in mind

If data sets the ceiling, how much can one straight decision boundary achieve before the model family itself becomes too weak?

Why this pairing works. AI2 handles the outside problem of trustworthy data. NN2 handles the inside problem of what one simple neural model can or cannot express.

Thank You