WEEK 3 WORKSHOP

Welcome to AI for Water Quality Workshop

AI-Supported Analysis and Research on Water Quality:

Evaluation, Treatment, Prediction & Management

The question this week answers:

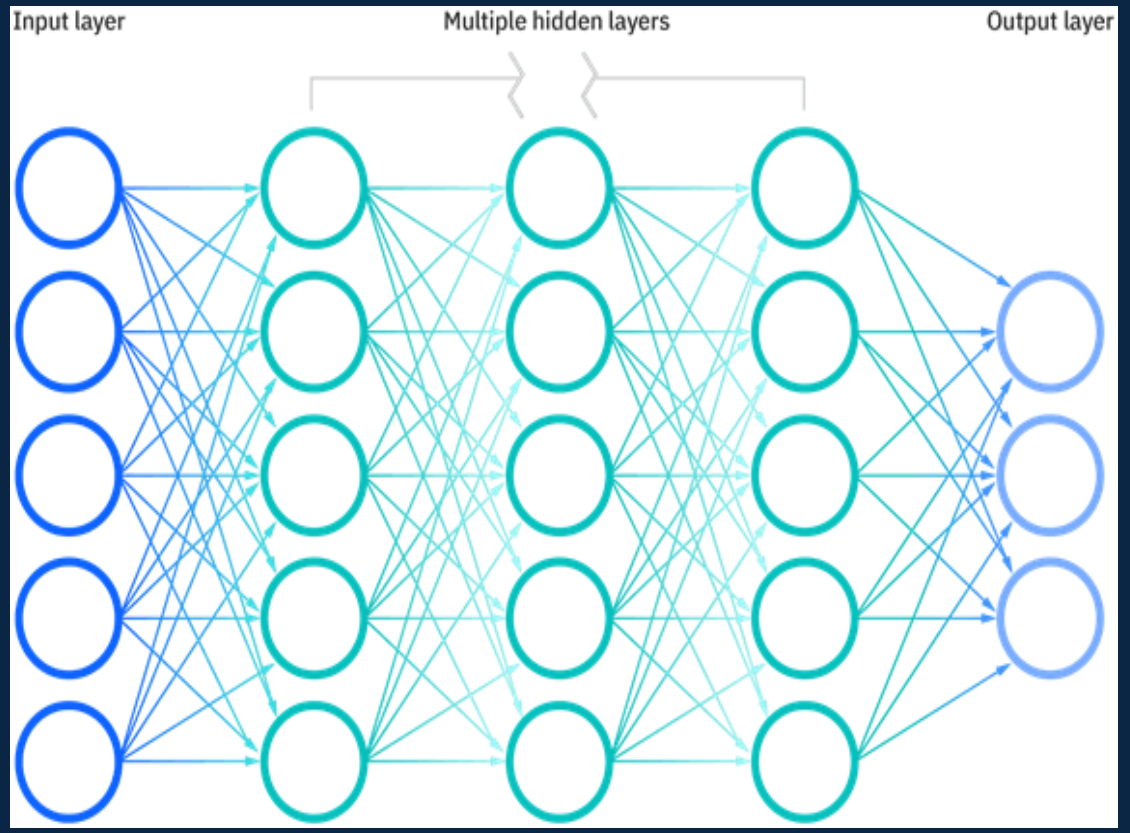
"Can a neural network catch more unsafe water samples than our Week 2 Random Forest — and is the trade-off worth it?"

Why this matters:

💧 Every missed unsafe sample is a public health risk.

What is a neural network? (deep learning)

"A neural network architecture is the arrangement of interconnected layers and neurons that determines how data flows through the network and how learning occurs.



Where we left off — the Week 2 benchmark

Before we can judge whether a neural network is better, we need to know exactly what "better" means

⚠️ The problem our Week 2 Random Forest left unsolved

?

unsafe samples slipped through as "safe"

This is your False positive count from Week 2 — find it in your Notebook from last week.

This is the number we are trying to reduce today.

The neural network wins ONLY if it brings this down.

📋 Record your Week 2 scores before the lab

Open your Week 2 notebook in Google Drive now.
Find and write down these three numbers:

🎯 **F1-Score**

for the "Not Potable" class

My Week 2 score: _____

📉 **Recall**

for the "Not Potable" class

My Week 2 score: _____

❌ **False Positives**

unsafe samples labelled "safe"

My Week 2 score: _____

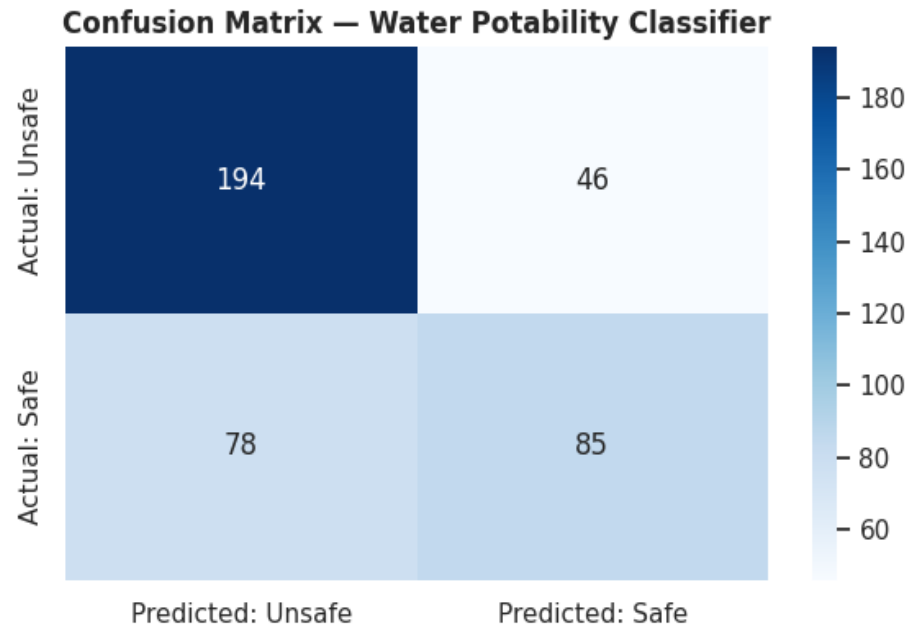
water_potability.csv

What my own Notebook says

```
Classification Report:
                  precision    recall  f1-score   support

Not Potable (0)       0.71      0.81      0.76       240
    Potable (1)       0.65      0.52      0.58       163

       accuracy                           0.69       403
      macro avg       0.68      0.66      0.67       403
   weighted avg       0.69      0.69      0.69       403
```




```
True Negatives  (correct: unsafe):  194
False Positives (said safe, not!):   46  ← public health risk
False Negatives (said unsafe, ok):   78
True Positives  (correct: safe):     85
```

F1-Score (Not Potable class): 0.76

Recall (Not Potable class): 0.81

False Positives (Unsafe samples, labelled “safe”): 46

What is a neural network



Imagine a water lab with dozens of analysts working in shifts.

Shift (Layer) 1 checks just pH and turbidity. Shift 2 takes those findings and cross-checks with chloramine and conductivity. Shift 3 looks at everything Shifts 1 and 2 found together, and makes the final call: Safe or Unsafe. Each shift learns from thousands of past cases.

Neural network contains layers that are made up of nodes.

That IS a neural network.



Random Forest (Week 2)

How it works:

One analyst. One flowchart.
Follows clear rules step by step.

Best for:

You can read every decision.
Ideal when you need to explain to a regulator WHY a sample failed.

⚠️ Misses patterns that need multiple features considered together.



Neural Network (Week 3)

How it works:

Many analysts across many shifts.
Learns from thousands of past cases.

Best for:

Spots complex multi-feature patterns the decision tree misses.
Often catches more unsafe samples.

⚠️ Cannot explain its decisions step by step.
Harder to justify to a regulator.

water_potability.csv

Step 1: What goes in — the 9 water measurements

Every water sample becomes a row of 9 numbers. Those numbers are the network's only input.

💧 One water sample — Sample #47

Collected: Riverside station, 14 Aug

pH	**6.8** (safe: 6.5–8.5)	
Hardness	**204** mg/L	
Solids	**22018** mg/L	
Chloramines	**7.2** mg/L (limit: 4)	!
Sulfate	**368** mg/L	
Conductivity	**564** µS/cm	
Organic carbon	**14.3** mg/L	
Trihalomethanes	**88.4** µg/L	
Turbidity	**4.1** NTU (limit: 4)	!

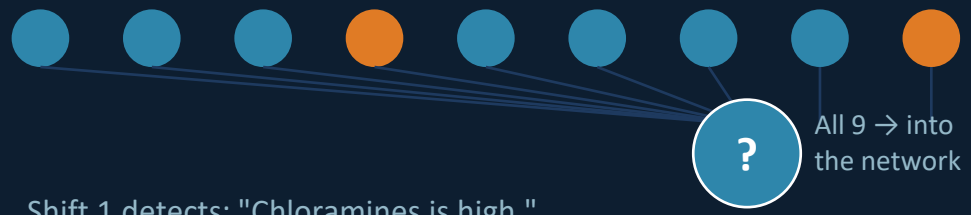
How each model reads this sample:

🌳 Random Forest (combining Decision Trees) reads ONE feature at a time

Step 1: Is Chloramines > threshold? → Yes → flag
Step 2: Is Turbidity > threshold? → Yes → flag
Step 3: Is Hardness > threshold? → No → continue
…

Each rule checked separately. Never asks:
"What if Chloramines AND Turbidity are both elevated?"

🧠 Neural network reads ALL 9 simultaneously

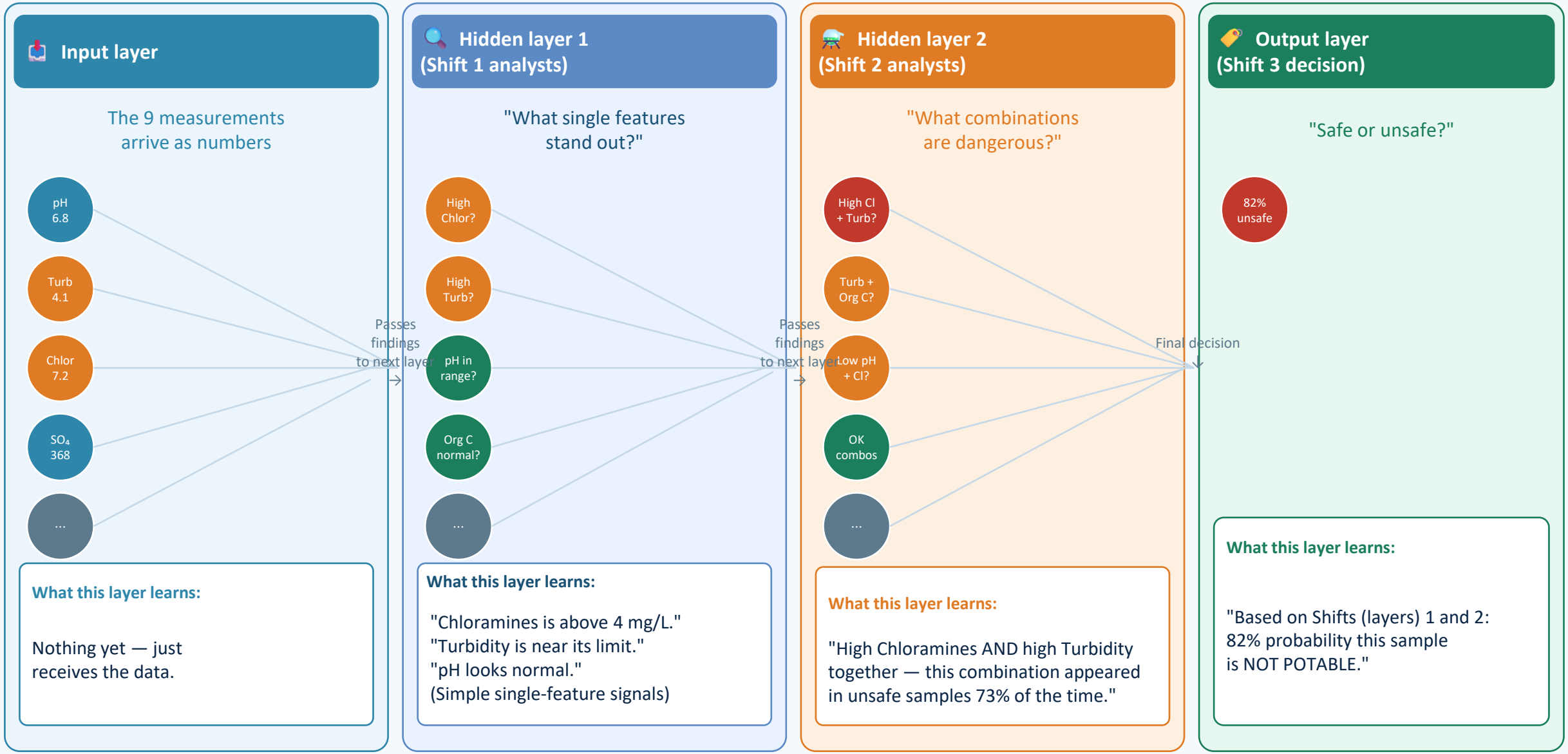


Shift 1 detects: "Chloramines is high."
Shift 2 detects: "Chloramines AND Turbidity are both elevated — that combination is dangerous."
Shift 3 makes the final call.

water_potability.csv

Step 2: What the layers do

Each layer finds patterns that are too subtle for the layer before it



water_potability.csv

Step 3: Testing: What comes out — a probability

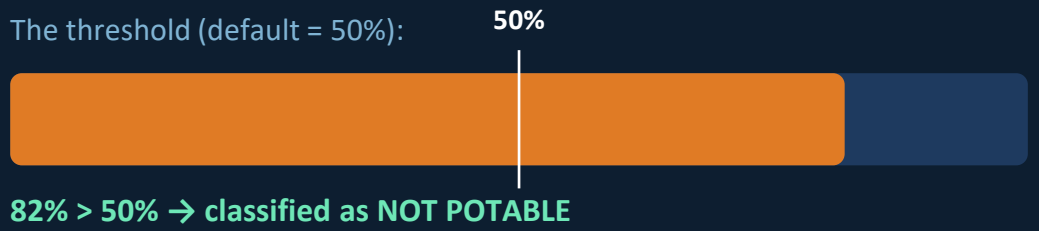
The network does not say "unsafe". It says "I think there is an 82% chance this is unsafe."

The output for Sample #47:

82%

probability of NOT POTABLE

The threshold (default = 50%):



82% > 50% → classified as NOT POTABLE

Why probabilities matter for water safety:

A sample at 51% and a sample at 98% are both labelled "unsafe" — but you might respond very differently to each. The raw probability lets you see how confident the model actually is.

⚠️ The limitation you must understand

The network tells you 82%. It cannot tell you why.

❓ Why did it flag Sample #47?

🌳 Random Forest:
"Chloramines > 4.0 AND Turbidity > 3.9 → Unsafe"
→ You can read this in the model.

🧠 Neural Network:
"82% — but I cannot show you the rule."
→ The decision lives across thousands of internal connections.

❓ What would a regulator ask?

🌳 "Show me the rule that flagged this sample."

🧠 "The model flagged it."
"But WHY?"
"...it learned from the data."

This is not a flaw to fix — it is a property to plan around.

For emergency screening: acceptable. For regulatory compliance: plan for human review.

How the network gets better

It practises. On every water sample. Thousands of times.

1



Show it a water sample

Feed in the 9 measurements.
The network makes a guess:
"I think this is safe — 72% confident."

2



Check — and measure the mistake

Compare the guess to the real label.
"Wrong. It was unsafe."
The error score is called the LOSS.

3



Adjust and try again

Tiny tweaks to every internal connection.
"Be less confident next time
when turbidity is high AND pH is low."

Two words you will see in the code and chart:

🔁 **Epoch**

One complete practice run through ALL training samples. We run many epochs — each one makes the network slightly better.

📉 **Loss**

The error score — how wrong the network was. We plot this after every epoch. A falling loss line = learning is happening.

The investigation — step by step

Every step in the lab directly serves the question we are answering

A

Load & prepare the data

Why: the same dataset and split as Week 2 — so the comparison is fair.

B

Build, train and evaluate the performance of the neural network

Why: We run it, then watch the loss curve confirm it actually learned.

C

Run the head-to-head comparison

Why: THIS is the answer. F1, Recall, and False Negatives — neural network vs Decision Tree.

D

Deliver the verdict

Why: write one sentence your team could act on. Science without a conclusion is incomplete.

📌 All code is already written with comments. Run → read the output → move on. Nothing to write from scratch.

y.csv

☕ 10-Minute Break

💬 Table discussion — 5 minutes after break

📋 **Scenario:**

Your lab's Random Forest catches 82% of unsafe samples. A neural network catches 89%.
But when a sample fails, you can only say "the model flagged it" — not why.

🏛️ Would you show this result to your water authority and recommend switching?

📞 A resident calls asking why their sample was flagged. What do you tell them?

⚖️ Name one situation where 7% more catches is worth the loss of explanation.

✋ **One person per table shares Q3. No wrong answers — this is the real tension in applied AI.**

What you will produce by the end of the lab

A comparison table and one written verdict — ready to share with your team

📊 Head-to-Head Comparison Table

Metric	Random Forest (Week 2)	Neural Network (Week 3)	Better?
F1 (Not Potable)	0.76	___	___
Recall (Not Potable)	0.81	___	___
False Positives	46	___	___
Can you explain why?	Yes ✅	No ❌	?

*Week 2 values here are mine — use your own actual notebook scores.

📝 Your conclusion (fill this in at the end of the lab):

"For water_potability.csv, the neural network [did / did not] outperform the Random Forest on catching unsafe samples [F1: ___ vs ___].
I would [recommend / not recommend] switching because _____."

Week 3 — What you now know

🧠

More

patterns found

The layers catch combinations — high chloramines AND high turbidity together — that rule-based Random Forest misses.

📉

Fewer

unsafe samples missed

On most runs, the neural network reduces false negatives — the dangerous misclassifications.

❓

Zero

explanations given

The network cannot tell you WHY a sample was flagged. That is the real trade-off this session surfaces.

The honest answer to the question:

"Yes — a neural network typically catches more unsafe water samples than a Random Forest on this dataset. But it gives up the ability to explain each decision. Whether that trade-off is acceptable depends on how the result will be used."

💾 **Save your notebook — the verdict table carries into Capstone Component 1 · See you in Week 4!**

Next: Week 4 — Generative AI applications in water research