AI IN WATER QUALITY WORKSHOP

Week 2

Machine Learning Fundamentals

Classification

Regression

Clustering

Dataset: water_potability.csv | Google Colab
23 September 2026

Before we start

📁 A refresher of Week 1 workshop working on Colab in Google Drive

💧 Recall: Week 1 → AI/ML/DL basics, Google Colab Walkthrough, Python basics (variables, lists, loops function etc)

🌐 Check you have internet access in Colab

📓 Have a notebook open — we build on Week 1 directly

Today's session: 2 hours

45 min lecture → 10 min break → 10 min activity → 55 min hands-on lab

More time in the lab — less time listening!

Today's Roadmap — 2 Hours

1:00–1:15	**Lecture: Classification**	What it is · Decision Tree · metrics	⏱ 15 min
1:15–1:30	**Lecture: Regression**	Predicting numbers · linear model · R^2 & MAE	⏱ 15 min
1:30–1:45	**Lecture: Clustering**	Unsupervised · K-Means · Elbow method	⏱ 15 min
1:45–1:55	☕ **Break**	Stretch, grab a drink, ask quick questions	⏱ 10 min
1:55–2:05	💬 **Group Activity / Break-out session**	"Which ML tool for the job?" — 4 scenarios	⏱ 10 min
2:05–3:00	💻 **Hands-On Lab (Colab)**	LazyPredict → individual models → clustering	⏱ 55 min

Our Dataset & Google Drive Setup

Real water_potability.csv — loaded directly from your Google Drive

💧 Dataset: water_potability.csv

3,276 water samples · 9 chemical features · 1 label

ph	Acidity / alkalinity (0–14)
Hardness	Dissolved calcium & magnesium
Solids	Total dissolved solids (ppm)
Chloramines	Disinfectant level (ppm)
Sulfate	Naturally occurring mineral
Conductivity	Electrical (µS/cm)
Organic_carbon	Organic matter (ppm)
Trihalomethanes	Disinfection by-products
Turbidity	Cloudiness (NTU)

🎯 **Potability — TARGET: 1 = Safe | 0 = Unsafe**

Step 1 — Mount Google Drive

```
# Run this FIRST every Colab session
from google.colab import drive
drive.mount('/content/drive')
# Click the link, sign in, paste code
```

Step 2 — Load the dataset

```
import pandas as pd

# Update path to match your Drive
PATH = ('/content/drive/MyDrive/
         AI_Water_Workshop/')

df = pd.read_csv(PATH +
   'water_potability.csv')

print(df.shape)
print(df.head())
```

01

Classification

"Is this water safe to drink?"

🏷️ **Sort data into labelled categories**

🌳 **Decision Tree learns from examples**

📊 **Evaluate: Accuracy, Precision, Recall, F1**

⚖️ **Handle class imbalance first**

Classification — Core Concepts

Input (Features)

9 numbers:
pH, Turbidity,
Chloramines…

→

Decision Tree Model

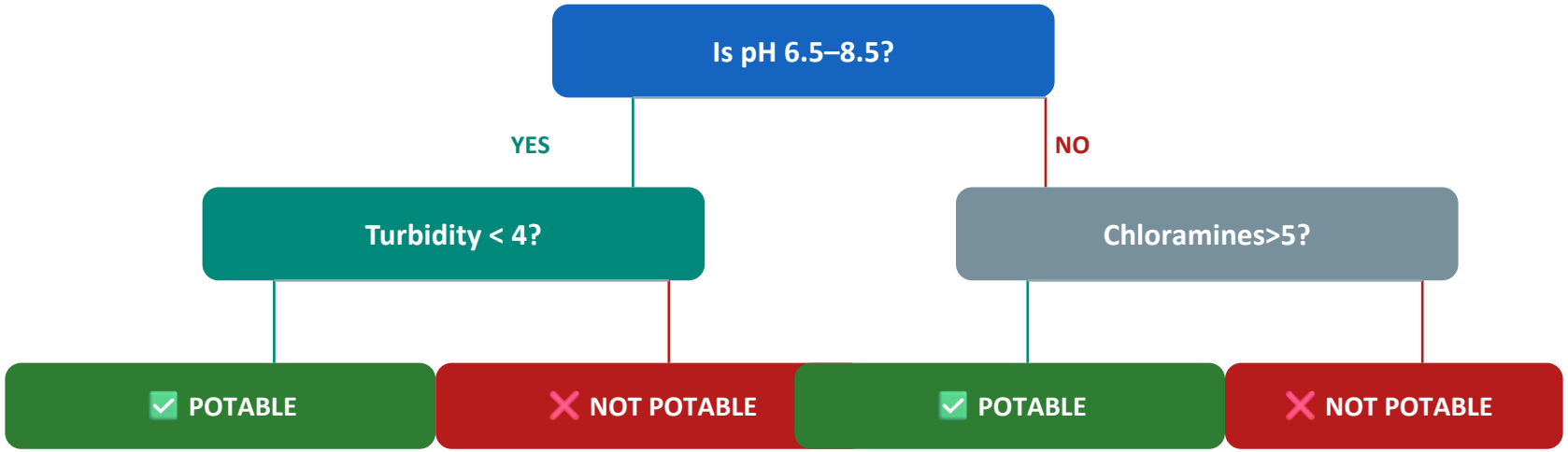
Learned flowchart
of if-then rules
from training data

→

Output (Label)

"Potable" 1
or
"Not Potable" 0

How a Decision Tree decides:



⚠️ Class Imbalance

In water_potability.csv:

~61% Not Potable (0)

~39% Potable (1)

A model guessing "Unsafe"

every time = 61% accuracy

— yet useless!

Fix: check value_counts(),

use F1-score not just

Accuracy, try SMOTE or

class_weight="balanced"

LazyPredict — Try 30+ Models in 3 Lines

Start here before spending time tuning any single model

```
!pip install lazypredict -q

from lazypredict.Supervised import LazyClassifier
from sklearn.model_selection import train_test_split

# Drop rows with missing values first
df_clean = df.dropna()
X = df_clean.drop('Potability', axis=1)
y = df_clean['Potability']

# 80/20 split, stratified to keep class ratio
X_train,X_test,y_train,y_test = train_test_split(
    X, y, test_size=0.2,
    random_state=42, stratify=y)
```

Evaluation Metrics — Quick Ref

Accuracy — Correct ÷ Total

Easy. Misleading when imbalanced.

Precision — TP ÷ (TP+FP)

When labelling safe, how often right?

Recall — TP ÷ (TP+FN)

Of all unsafe — did we catch them?

F1-Score — 2×(P×R)÷(P+R)

Best single metric when imbalanced.

ROC-AUC — Area under curve

How well model separates classes.

02

Regression

"What will the turbidity reading be?"

🔢 **Predict a number, not a category**

📈 **Linear model: best-fit line through data**

⚡ **LazyPredict runs regressors too!**

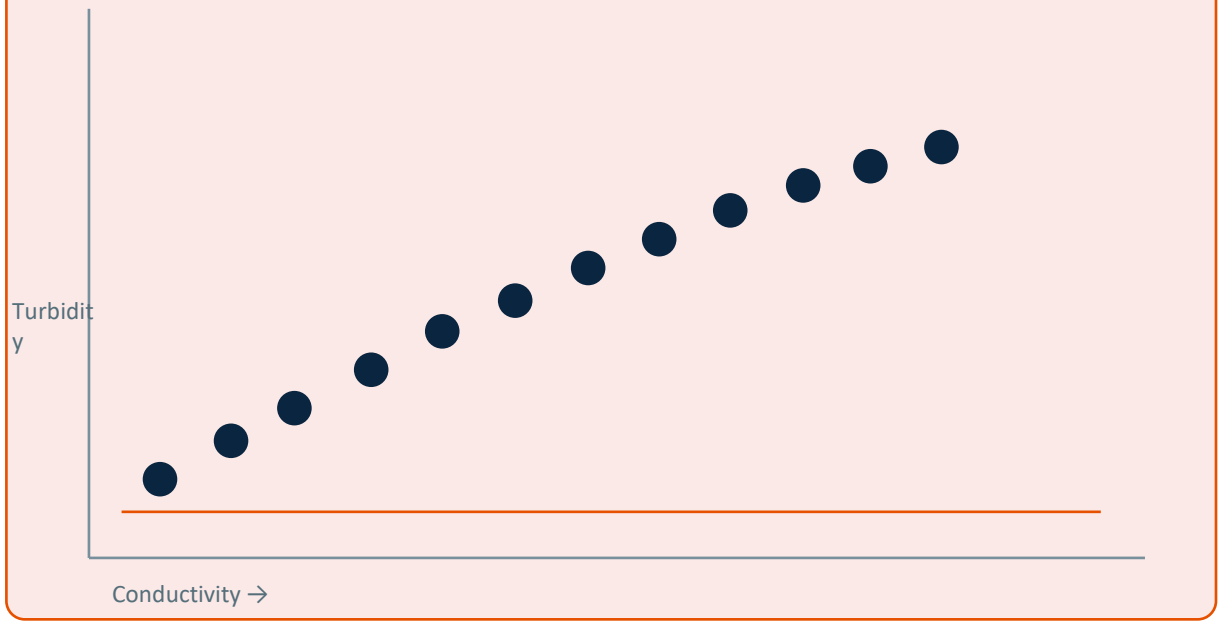
📐 **Evaluate: MAE and R^2**

Regression — Predicting Turbidity

Classification vs Regression — at a glance

	Classification	Regression
Question	Is it X or Y?	How much / how many?
Output	A label (category)	A number (continuous)
Water eg.	Safe (1) or Unsafe (0)?	Predicted turbidity = 3.7 NTU
Metric	Accuracy, F1, Recall	MAE, R^2

Linear Regression — best-fit line



How we measure accuracy:

MAE Mean Absolute Error

Average gap between prediction and reality.
MAE = 0.4 means you are 0.4 NTU off on average.
Easy to explain to stakeholders.

R^2 R-squared

Proportion of variation explained by the model.
R^2 = 1.0 → perfect | R^2 = 0 → no better than guessing the average.

03

Clustering

"Which water sources look similar?"

🔮 **No labels — unsupervised learning**

📍 **K-Means assigns samples to K groups**

📐 **Choose K with the Elbow Method**

🏷️ **Output: cluster IDs, not predictions**

K-Means Clustering

✅ Supervised (Classification / Regression)

You provide labels with training examples.
Model learns from known answers.

🔮 Unsupervised (Clustering)

No labels — just raw measurements.
Model discovers its own groups.

K-Means — 3 steps:

1 Pick K

Choose how many groups. We use K = 3 today.

e.g. K = 3 clusters

2 Assign

Each sample joins its nearest centre. Distance decides membership.

Nearest centre wins

3 Update

Centres shift to average of their group. Repeat until stable.

Repeat until done

Elbow Method — how to choose K

Try K = 1 to 10. Plot inertia (sum of squared distances) for each K.
Look for the "elbow" — where the curve stops dropping sharply. That K is your best choice.

⚠️ Always scale first!

K-Means uses distances.
StandardScaler ensures no feature dominates just because its numbers are bigger.

10-Minute Break

Group Activity (10 minutes after break)

"Which ML Tool for the Job?" — In groups of 3–4, decide for each scenario below:

A We want to predict the exact pH reading for river samples next week.

B 200 unlabelled borehole samples arrived. Find natural chemical groupings.

C Build a real-time alert: flag each sample as "safe" or "unsafe" instantly.

D Estimate chlorine dose (mg/L) needed based on current turbidity and pH.

Each group shares one answer → reveal answers together.

Hands-On Lab — 55 Minutes

Open the Week 2 Workbook from the shared Google Drive folder

Part A

Classification (Pass 1 — LazyPredict)

Mount Drive → load data → handle NaN & imbalance → LazyClassifier → rank models

⏱ ~15 min

Part B

Classification (Pass 2 — Best Model)

Pick top model → Decision Tree with class_weight → confusion matrix → F1 report

⏱ ~12 min

Part C

Regression (Pass 1 — LazyPredict)

LazyRegressor → rank by R^2 → identify top performers

⏱ ~8 min

Part D

Regression (Pass 2 — Best Model)

Linear Regression → residual plot → MAE and R^2

⏱ ~8 min

Part E

Clustering

Scale → Elbow → K-Means (K=3) → scatter plot → describe each cluster

⏱ ~12 min

Follow along in Colab with # comments — run each cell, read the output, answer the reflection questions.

Week 2 — Summary

Classification

- Label data into categories (Safe/Unsafe).
- Decision Tree + LazyClassifier to compare.
- Always check class balance first.
- Evaluate with F1-Score and Recall.

Regression

- Predict a continuous number (turbidity).
- LazyRegressor benchmarks all models.
- Best model → Linear Regression.
- Evaluate with MAE and R^2.

Clustering

- Find hidden groups — no labels needed.
- Scale features before K-Means.
- Elbow Method → choose K.
- Describe each cluster by its means.

📋 Capstone Project — Week 6

Apply today's three techniques to a water quality dataset of your choice (see the Capstone Brief):

- Component 1: Classification, Regression & Clustering analysis + code
- Component 2 : Ethics of AI — one theme (Fairness, XAI, Trust, Transparency…)