# Week 1: Water Quality Prediction - Initial Report

# 1. Import libraries

import pandas as pd

import numpy as np

from sklearn.model\_selection import train\_test\_split

from sklearn.tree import DecisionTreeClassifier

from sklearn.metrics import classification\_report, accuracy\_score

import matplotlib.pyplot as plt

import seaborn as sns

# 2. Load dataset

data\_path = 'water\_quality.csv' # Adjust path if needed

df = pd.read\_csv(data\_path)

print("Dataset Loaded Successfully.")

print(f"Shape of dataset: {df.shape}")

# 3. Initial Data Exploration

print("\nBasic Info:")

print(df.info())

print("\nMissing Values:")

print(df.isnull().sum())

print("\nDescriptive Statistics:")

print(df.describe())

# 4. Data Preprocessing

df = df.dropna() # Simple approach: drop rows with missing values

# Features and target

X = df.drop('Potability', axis=1)

y = df['Potability']

# 5. Train-test split

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

# 6. Model Training

model = DecisionTreeClassifier(random\_state=42)

model.fit(X\_train, y\_train)

# 7. Evaluation

y\_pred = model.predict(X\_test)

print("\nModel Evaluation:")

print("Accuracy Score:", accuracy\_score(y\_test, y\_pred))

print("Classification Report:\n", classification\_report(y\_test, y\_pred))

# 8. Feature Importance Visualization

importances = model.feature\_importances\_

feature\_names = X.columns

feature\_imp\_df = pd.DataFrame({'Feature': feature\_names, 'Importance': importances})

feature\_imp\_df.sort\_values(by='Importance', ascending=False, inplace=True)

plt.figure(figsize=(10, 6))

sns.barplot(x='Importance', y='Feature', data=feature\_imp\_df)

plt.title("Feature Importance - Decision Tree")

plt.tight\_layout()

plt.show()