**Water Quality Analysis – Phase4**

**Virtualization:**

import matplotlib.pyplot as plt

import numpy as np

import pandas as pd

plt.hist(df["Solids"])

**Output:**

A graph with numbers and a black background

Description automatically generated

x=df["Sulfate"]

y=df["Trihalomethanes"]

matrix = np.corrcoef(x, y)

print(matrix)

**Output:**

A black background with white text

Description automatically generated

plt.scatter(df["ph"],df["Solids"])

**Output:**

A blue dot diagram with numbers

Description automatically generated with medium confidence

sns.barplot(x=df["Organic\_carbon"],y=df["Solids"],data=df,palette="plasma")

**Output:**

A screen shot of a computer screen

Description automatically generated

Predict :

import numpy as np

import pandas as pd

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

from sklearn.linear\_model import LinearRegression

from sklearn.metrics import mean\_squared\_error

import matplotlib.pyplot as plt

data = pd.read\_csv("/content/water\_potability.csv")

X = data['Hardness'].values.reshape(-1, 1)

y = data['Turbidity'].values

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=0)

model = LinearRegression()

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

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

mse = mean\_squared\_error(y\_test, y\_pred)

print(f"Mean Squared Error: {mse}")

plt.scatter(X, y, color='blue')

plt.plot(X\_test, y\_pred, color='red', linewidth=2)

plt.xlabel('Time')

plt.ylabel('Water Quality')

plt.show()

future\_time = np.array([2024, 2025, 2026]).reshape(-1, 1)

future\_quality = model.predict(future\_time)

print("Predicted water quality for future years:")

for year, quality in zip([2024, 2025, 2026], future\_quality):

    print(f"Year: {year}, Predicted Quality: {quality}")



A screen shot of a graph

Description automatically generated