import pandas as pd

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

from sklearn.ensemble import GradientBoostingRegressor

from sklearn.metrics import mean\_squared\_error

# Read CSV dataset

data\_path = r" "

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

# Remove spaces from column names

df.columns = df.columns.str.strip()

# Print column names to ensure they are correct

print("Column names:", df.columns)

# Ensure that the feature column names match those in the dataset

X = df[['Concentration (ppm)', 'Temperature', 'Cu Dose (mol%)']] # Features

y = df['Response'] # Target variable

# Split the dataset (70% training data, 30% test data)

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.3, random\_state=42)

# Use GradientBoostingRegressor for regression modeling

model = GradientBoostingRegressor()

# Train the model

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

# Use the trained model to make predictions

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

# Assuming you input new feature values for prediction

new\_data = [

[100, 120, 0], [100, 120, 1], [100, 120, 2], [100, 120, 3], [100, 120, 4],

[100, 160, 0], [100, 160, 1], [100, 160, 2], [100, 160, 3], [100, 160, 4],

[100, 200, 0], [100, 200, 1], [100, 200, 2], [100, 200, 3], [100, 200, 4],

[100, 240, 0], [100, 240, 1], [100, 240, 2], [100, 240, 3], [100, 240, 4],

[100, 280, 0], [100, 280, 1], [100, 280, 2], [100, 280, 3], [100, 280, 4]

]

# Convert the new data to a DataFrame and assign column names

new\_data\_df = pd.DataFrame(new\_data, columns=['Concentration (ppm)', 'Temperature', 'Cu Dose (mol%)'])

# Use the model to make predictions

predicted\_responses = model.predict(new\_data\_df)

# Print the predicted response for each combination

for i, data in enumerate(new\_data):

print(f"Concentration: {data[0]} ppm, Temperature: {data[1]} °C, Cu Dose: {data[2]} wt%, Predicted Response: {predicted\_responses[i]}")