import pandas as pd

from sklearn.ensemble import GradientBoostingRegressor

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

from sklearn.metrics import mean\_squared\_error, r2\_score

import os

# File path

file\_path = r" "

# Check if the file exists

if not os.path.exists(file\_path):

print(f"File path is incorrect, file does not exist: {file\_path}")

exit()

# Load the dataset

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

# Remove extra spaces from column names

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

# Preview the first few rows of the dataset to confirm correct loading

print("Dataset loaded successfully, first few rows:")

print(df.head())

# Check if column names are correct

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

# Features and target variable

X = df[['Concentration (ppm)', 'Temperature (°C)', 'Dopant (wt%)']] # Use correct column names

y = df['Response'] # Target variable

# Split the dataset into training and test sets

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

# Initialize the Gradient Boosting Regressor model

model = GradientBoostingRegressor(n\_estimators=300, learning\_rate=0.8, max\_depth=3, random\_state=42)

# Train the model

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

# Make predictions on the test set

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

# Calculate model performance

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

r2 = r2\_score(y\_test, y\_pred)

# New data points

new\_data = [

[25, 25, 8.3], [25, 50, 8.3], [25, 100, 8.3], [25, 125, 8.3], [25, 175, 8.3],

[25, 25, 16], [25, 50, 16], [25, 100, 16], [25, 125, 16], [25, 175, 16]

]

# Use the model to make predictions on new data

new\_predictions = model.predict(new\_data)

# Output the prediction results

for i, pred in enumerate(new\_predictions):

print(f"Prediction for concentration = 10, temperature = {new\_data[i][1]}, dopant\_level = {new\_data[i][2]}: {pred}")