import pandas as pd

from sklearn.ensemble import GradientBoostingRegressor

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

# Load the dataset

data\_path = r" "

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

# Ensure the dataset contains the required columns

required\_columns = ["Temperature (°C)", "Concentration", "Response Time", "Response Value"]

if not all(col in data.columns for col in required\_columns):

raise ValueError(f"The dataset is missing required columns: {required\_columns}")

# Prepare the data

X = data[["Temperature (°C)", "Concentration", "Response Time"]] # Features

y = data["Response Value"] # Target variable (response value)

# Split the dataset into training and testing sets

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

# Train the Gradient Boosting Regressor model

gbr = GradientBoostingRegressor(random\_state=42)

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

# Create input data for prediction

temperatures\_to\_predict = [100, 150, 180, 230, 280, 300, 340] # Temperatures to predict

concentration\_to\_predict = [100] \* len(temperatures\_to\_predict) # Constant concentration of 100

response\_time\_to\_predict = [3] \* len(temperatures\_to\_predict) # Constant response time of 3

X\_predict = pd.DataFrame({

"Temperature (°C)": temperatures\_to\_predict,

"Concentration": concentration\_to\_predict,

"Response Time": response\_time\_to\_predict

})

# Make predictions

predicted\_values = gbr.predict(X\_predict)

# Output the prediction results

prediction\_results = pd.DataFrame({

"Temperature (°C)": temperatures\_to\_predict,

"Concentration": concentration\_to\_predict,

"Response Time": response\_time\_to\_predict,

"Predicted Response Value": predicted\_values

})

print("Prediction Results:")

print(prediction\_results)