import pandas as pd

import numpy as np

# Create the dataset

data = {

'Study Hours': [4.37, 9.56, 7.59, 6.39, 2.40, 8.80, 1.52, 7.37, 2.91, 9.73],

'Sleep Hours': [7.67, 4.83, 5.75, 6.20, 6.74, 7.08, 5.20, 4.28, 9.69, 5.02],

'Outcome': [0, 1, 0, 0, 0, 1, 0, 0, 0, 1]

}

df = pd.DataFrame(data)

# Define sigmoid function

def sigmoid(z):

return 1 / (1 + np.exp(-z))

# Prepare data

X = np.column\_stack((np.ones(len(df)), df['Study Hours'], df['Sleep Hours']))

y = df['Outcome']

# Initialize coefficients

beta = np.zeros(X.shape[1])

alpha = 0.01

iterations = 1000

# Gradient Descent

for \_ in range(iterations):

z = np.dot(X, beta)

h = sigmoid(z)

gradient = np.dot(X.T, (h - y)) / len(y)

beta -= alpha \* gradient

# Predict Outcomes

y\_pred\_prob = sigmoid(np.dot(X, beta))

y\_pred = (y\_pred\_prob >= 0.5).astype(int)

# Display Coefficients and Predictions

print(f"Intercept: {beta[0]}")

print(f"Study Hours Coefficient: {beta[1]}")

print(f"Sleep Hours Coefficient: {beta[2]}")

print("\nPredicted Outcomes:")

print(y\_pred)