import numpy as np

import pandas as pd

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

from sklearn.preprocessing import StandardScaler

from sklearn.linear\_model import LogisticRegression

from sklearn.metrics import accuracy\_score, classification\_report, confusion\_matrix

data = pd.DataFrame({

'Age': [25, 30, 35, 40, 45, 50, 55, 60, 65, 70],

'Income': [40000, 45000, 50000, 80000, 90000, 75000, 55000, 60000, 50000, 45000],

'Buy\_Product': [1, 0, 1, 1, 0, 1, 0, 0, 0, 1]

})

# Split the data into features (X) and the target variable (y).

X = data[['Age', 'Income']]

y = data['Buy\_Product']

# Split the data into training and testing sets (80% training, 20% testing).

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

# Standardize the feature data (Age and Income) to have zero mean and unit variance.

scaler = StandardScaler()

X\_train = scaler.fit\_transform(X\_train)

X\_test = scaler.transform(X\_test)

# Create a logistic regression model and train it on the training data.

model = LogisticRegression(random\_state=42)

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

# Make predictions on the test data.

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

# Evaluate the model's performance.

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

print("Accuracy:", accuracy)

confusion = confusion\_matrix(y\_test, y\_pred)

print("Confusion Matrix:\n", confusion)

classification\_rep = classification\_report(y\_test, y\_pred)

print("Classification Report:\n", classification\_rep)