"""

Task 1: Credit Scoring Model

- Predict creditworthiness using financial history

- Models: Logistic Regression, Decision Tree, Random Forest

"""

import pandas as pd

import numpy as np

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

from sklearn.metrics import accuracy\_score, precision\_score, recall\_score, f1\_score, roc\_auc\_score

from sklearn.linear\_model import LogisticRegression

from sklearn.tree import DecisionTreeClassifier

from sklearn.ensemble import RandomForestClassifier

# ===============================

# Step 1: Example Dataset

# ===============================

# (In real projects, load from CSV or database)

data = {

"income": [40000, 25000, 60000, 120000, 35000, 52000, 15000, 90000, 80000, 30000],

"debt": [5000, 12000, 2000, 15000, 8000, 10000, 7000, 5000, 12000, 6000],

"late\_payments": [0, 3, 1, 5, 2, 0, 4, 1, 0, 2],

"loan\_amount": [10000, 5000, 20000, 30000, 8000, 15000, 4000, 25000, 20000, 7000],

"credit\_score": [700, 580, 750, 620, 640, 710, 500, 780, 730, 600],

"label": [1, 0, 1, 0, 0, 1, 0, 1, 1, 0] # 1 = Good credit, 0 = Bad credit

}

df = pd.DataFrame(data)

X = df.drop("label", axis=1)

y = df["label"]

# ===============================

# Step 2: Split Data

# ===============================

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

# ===============================

# Step 3: Train Models

# ===============================

models = {

"Logistic Regression": LogisticRegression(),

"Decision Tree": DecisionTreeClassifier(),

"Random Forest": RandomForestClassifier(n\_estimators=100)

}

results = {}

for name, model in models.items():

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

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

y\_prob = model.predict\_proba(X\_test)[:,1]

results[name] = {

"Accuracy": accuracy\_score(y\_test, y\_pred),

"Precision": precision\_score(y\_test, y\_pred),

"Recall": recall\_score(y\_test, y\_pred),

"F1": f1\_score(y\_test, y\_pred),

"ROC-AUC": roc\_auc\_score(y\_test, y\_prob)

}

# ===============================

# Step 4: Show Results

# ===============================

print("\nModel Evaluation Results:")

for name, metrics in results.items():

print(f"\n{name}:")

for metric, value in metrics.items():

print(f" {metric}: {value:.3f}")

# Choose best model (Random Forest usually performs well)

best\_model = models["Random Forest"]

# ===============================

# Step 5: Take User Input

# ===============================

print("\n--- Credit Score Prediction ---")

income = float(input("Enter your annual income: "))

debt = float(input("Enter your total debt: "))

late\_payments = int(input("Enter number of late payments: "))

loan\_amount = float(input("Enter requested loan amount: "))

credit\_score = int(input("Enter your credit score: "))

user\_data = np.array([[income, debt, late\_payments, loan\_amount, credit\_score]])

prediction = best\_model.predict(user\_data)

prob = best\_model.predict\_proba(user\_data)[0][1]

print("\nPrediction Result:")

if prediction[0] == 1:

print(f"✅ Approved: Good Credit (Confidence: {prob:.2f})")

else:

print(f"❌ Rejected: Bad Credit (Confidence: {prob:.2f})")