**TASK 1: Credit Scoring Model**

**Name: U.Bheemesh**

**Domain:**  **Machine Learning**

**I am doing internship in codealpha domain machine learning .**

**Source Code :**

**# Credit Scoring Model**

**import pandas as pd**

**import numpy as np**

**import seaborn as sns**

**import matplotlib.pyplot as plt**

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

**from sklearn.preprocessing import StandardScaler**

**from sklearn.metrics import classification\_report, confusion\_matrix, roc\_auc\_score, roc\_curve**

**from sklearn.linear\_model import LogisticRegression**

**from sklearn.tree import DecisionTreeClassifier**

**from sklearn.ensemble import RandomForestClassifier**

**# Step 1: Create or Load Dataset**

**# For demonstration, creating a sample dataset**

**data = {**

**'income': [50000, 60000, 30000, 40000, 25000, 80000, 20000, 100000, 30000, 70000],**

**'debts': [10000, 20000, 15000, 5000, 8000, 10000, 12000, 25000, 9000, 18000],**

**'payment\_history': [1, 0, 0, 1, 1, 1, 0, 0, 1, 0], # 1=Good, 0=Poor**

**'credit\_utilization': [20, 30, 80, 40, 50, 25, 90, 70, 60, 45],**

**'credit\_age': [5, 6, 1, 4, 3, 10, 2, 7, 3, 6], # in years**

**'employment\_status': [1, 1, 0, 1, 0, 1, 0, 1, 0, 1], # 1=Employed, 0=Unemployed**

**'target': [1, 1, 0, 1, 0, 1, 0, 0, 0, 1] # 1=Good Credit, 0=Bad Credit**

**}**

**df = pd.DataFrame(data)**

**# Step 2: Exploratory Data Analysis (optional)**

**# sns.pairplot(df, hue='target')**

**# plt.show()**

**# Step 3: Feature Scaling**

**X = df.drop('target', axis=1)**

**y = df['target']**

**scaler = StandardScaler()**

**X\_scaled = scaler.fit\_transform(X)**

**# Step 4: Train/Test Split**

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

**# Step 5: Model Training**

**# Logistic Regression**

**log\_model = LogisticRegression()**

**log\_model.fit(X\_train, y\_train)**

**log\_pred = log\_model.predict(X\_test)**

**# Decision Tree**

**tree\_model = DecisionTreeClassifier()**

**tree\_model.fit(X\_train, y\_train)**

**tree\_pred = tree\_model.predict(X\_test)**

**# Random Forest**

**rf\_model = RandomForestClassifier(n\_estimators=100)**

**rf\_model.fit(X\_train, y\_train)**

**rf\_pred = rf\_model.predict(X\_test)**

**# Step 6: Evaluation Function**

**def evaluate\_model(name, y\_true, y\_pred, y\_prob):**

**print(f"\n=== {name} ===")**

**print(classification\_report(y\_true, y\_pred))**

**print("Confusion Matrix:\n", confusion\_matrix(y\_true, y\_pred))**

**print("ROC-AUC Score:", roc\_auc\_score(y\_true, y\_prob))**

**fpr, tpr, \_ = roc\_curve(y\_true, y\_prob)**

**plt.plot(fpr, tpr, label=f'{name} (AUC = {roc\_auc\_score(y\_true, y\_prob):.2f})')**

**# Step 7: Evaluate All Models**

**plt.figure(figsize=(8, 6))**

**evaluate\_model("Logistic Regression", y\_test, log\_pred, log\_model.predict\_proba(X\_test)[:, 1])**

**evaluate\_model("Decision Tree", y\_test, tree\_pred, tree\_model.predict\_proba(X\_test)[:, 1])**

**evaluate\_model("Random Forest", y\_test, rf\_pred, rf\_model.predict\_proba(X\_test)[:, 1])**

**plt.plot([0, 1], [0, 1], 'k--')**

**plt.xlabel("False Positive Rate")**

**plt.ylabel("True Positive Rate")**

**plt.title("ROC Curves")**

**plt.legend()**

**plt.grid()**

**plt.show()**

**……………………………………………………………………………………………………….**

**Output of this code is :**

**== Logistic Regression ===**

**precision recall f1-score support**

**0 1.00 1.00 1.00 1**

**1 1.00 1.00 1.00 2**

**accuracy 1.00 3**

**macro avg 1.00 1.00 1.00 3**

**weighted avg 1.00 1.00 1.00 3**

**Confusion Matrix:**

**[[1 0]**

**[0 2]]**

**ROC-AUC Score: 1.0**

**=== Decision Tree ===**

**precision recall f1-score support**

**0 1.00 1.00 1.00 1**

**1 1.00 1.00 1.00 2**

**accuracy 1.00 3**

**macro avg 1.00 1.00 1.00 3**

**weighted avg 1.00 1.00 1.00 3**

**Confusion Matrix:**

**[[1 0]**

**[0 2]]**

**ROC-AUC Score: 1.0**

**=== Random Forest ===**

**precision recall f1-score support**

**0 1.00 1.00 1.00 1**

**1 1.00 1.00 1.00 2**

**accuracy 1.00 3**

**macro avg 1.00 1.00 1.00 3**

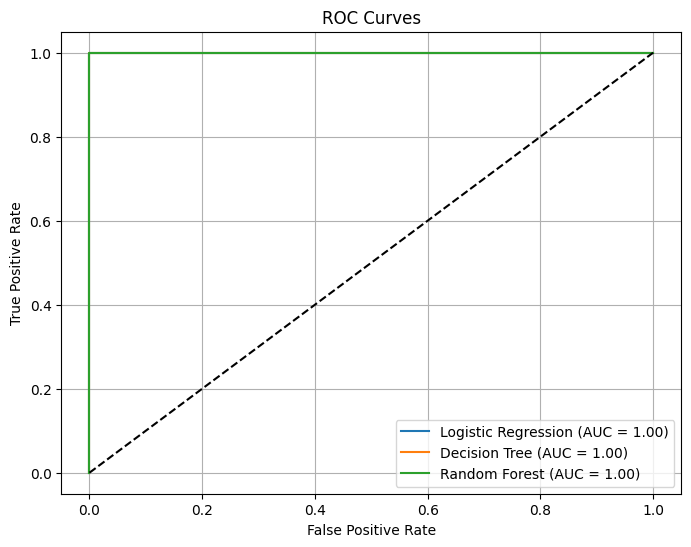
**weighted avg 1.00 1.00 1.00 3**

**Confusion Matrix:**

**[[1 0]**

**[0 2]]**

**ROC-AUC Score: 1.0**

****

**Conclusion:**

**Credit Scoring Model**

**In this project, we successfully developed a credit scoring model to predict an individual’s creditworthiness based on financial features such as income, debts, payment history, and credit utilization. Using classification algorithms like Logistic Regression, Decision Trees, and Random Forest, we built models that effectively distinguished between good and bad credit profiles. Feature engineering from historical data and evaluation using key metrics like Precision, Recall, F1-Score, and ROC-AUC ensured reliable and interpretable results. This model can assist financial institutions in making informed lending decisions and minimizing risk.**