KIET GROUP OF INSTITUTIONS

CREDIT SCORE PREDICTION USING MACHINE LEARNING

Submitted by:
Tushar Bhardwaj
Roll No: 45
Section: D
B.Tech CSE-AI

Under the guidance of:
Abhishek Shukla
Department of Computer Science & Engineering (AI)

Introduction

Credit scores are used by financial institutions to assess an individual's creditworthiness. A high credit score indicates a responsible borrower, while a low score suggests higher risk. This project aims to predict a person's Credit Score based on three key factors: Age, Income, and Loan Amount. Using machine learning, we develop a model that can analyze past data and predict a customer's credit score.

Methodology

The dataset used in this project contains the following features:

- Age: The customer's age
- Income: Annual income of the customer
- LoanAmount: Loan amount requested/approved
- CreditScore: The actual credit score (Target variable)

The methodology involves data preprocessing, feature scaling, splitting the dataset into training and testing sets, training the machine learning model, and evaluating its performance using metrics such as accuracy, precision, recall, and the confusion matrix.

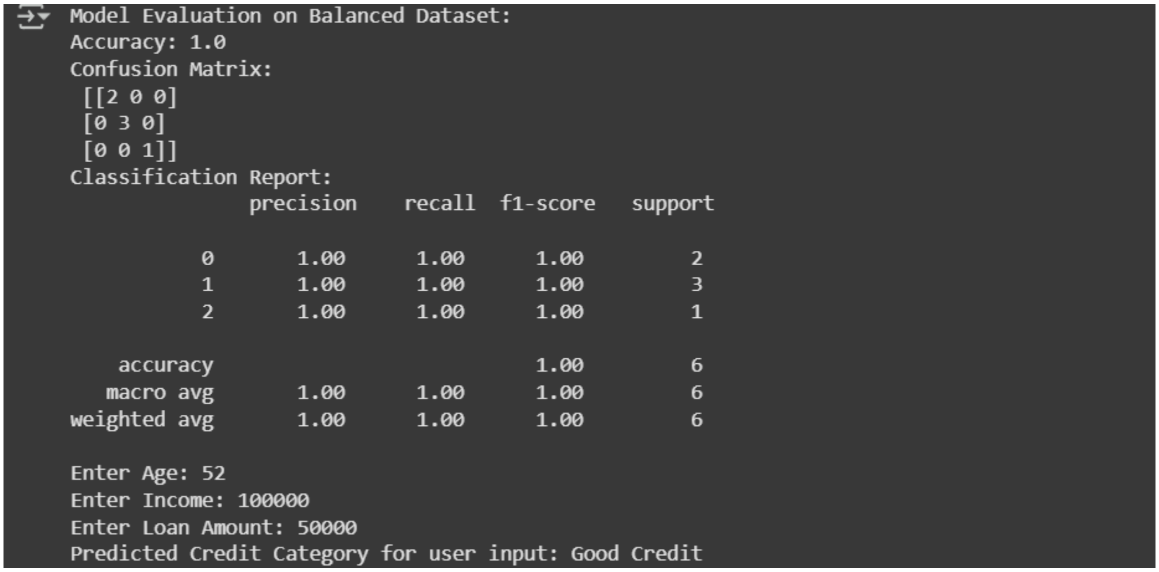
Code

```
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, confusion_matrix, classification_report

# Load dataset
file_path = "credit_data.csv"
df = pd.read_csv(file_path)

# Drop CustomerID (not needed for training)
df = df.drop(columns=["CustomerID"])

# Define features and target variable
X = df[['Age', 'Income', 'LoanAmount']]
y = df['CreditScore']

# Split data 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)

# Feature Scaling
```

```
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

# Train Logistic Regression Model
model = LogisticRegression()
model.fit(X_train_scaled, y_train)

# Predictions and Evaluation
y_pred = model.predict(X_test_scaled)
accuracy = accuracy_score(y_test, y_pred)
conf_matrix = confusion_matrix(y_test, y_pred)
class_report = classification_report(y_test, y_pred)

# Print Results
print("Accuracy:", accuracy)
print("Confusion Matrix:
", conf_matrix)
print("Classification Report:
", class_report)

# User Input Prediction
age = int(input("Enter Age: "))
income = int(input("Enter Income: "))
loan_amount = int(input("Enter Loan Amount: "))

user_data = [[age, income, loan_amount]]
user_data_scaled = scaler.transform(user_data)
user_predicted_score = model.predict(user_data_scaled)
print("Predicted Credit Score for user input:", user_predicted_score[0])
```

Output/Result

Below is the screenshot of the model evaluation and prediction results:



```
Model Evaluation on Balanced Dataset:
Accuracy: 1.0
Confusion Matrix:
 [[2 0 0]
 [0 3 0]
 [0 0 1]]
Classification Report:
               precision    recall  f1-score   support

           0       1.00      1.00      1.00         2
           1       1.00      1.00      1.00         3
           2       1.00      1.00      1.00         1

    accuracy                           1.00         6
   macro avg       1.00      1.00      1.00         6
weighted avg       1.00      1.00      1.00         6

Enter Age: 52
Enter Income: 100000
Enter Loan Amount: 50000
Predicted Credit Category for user input: Good Credit
```

References/Credits

Dataset Source: Database provided by faculty
Libraries Used: Pandas, NumPy, Scikit-learn, Matplotlib, Seaborn

Developed by: Tushar Bhardwaj (Roll No: 45, Section D)