CREDIT SCORE PREDICTION

Problem Statement

Develop a Credit Score Prediction system that cleans and transforms financial data to improve credit risk assessment models. The system should process raw data, handle missing values, encode categorical data, scale numerical values, train a machine learning model, and evaluate its performance.

Personal Details:

Name: Akeel Ahmad
Roll No.: 202401100400020

Problem Definition

In today’s financial world, predicting credit scores is a critical task for banks and financial institutions. It helps them assess the risk associated with lending money to a borrower. An accurate prediction model reduces the likelihood of defaults, ensuring the financial health of the institution and providing fair assessments for customers.

The goal of this project is to develop a machine learning-based Credit Score Prediction model. The system uses historical customer data (age, income, loan amount, credit history, etc.) to predict whether a person is a "good" or "bad" credit risk.

Approach to Solve the Problem statement

1. **Data Collection**
    - A dataset containing customer financial information was used. It includes features such as Age, Income, Loan Amount, Credit History, etc.
2. **Data Cleaning and Preprocessing**
    - Handle missing values by imputing them.
    - Encode categorical variables such as 'Purpose' and 'Credit Risk' into numerical formats using Label Encoding.
    - Scale the numerical features using Standard Scaler to normalize data.
3. **Feature Selection**
    - Select the most relevant features that contribute to the credit score prediction.
4. **Model Building**
    - Use a Random Forest Classifier for training the model due to its robustness and accuracy in handling both numerical and categorical data.
5. **Model Evaluation**
    - Evaluate the model using accuracy score, confusion matrix, and classification report to check performance.
6. **Visualization**
    - Visualize the confusion matrix for a better understanding of model predictions.

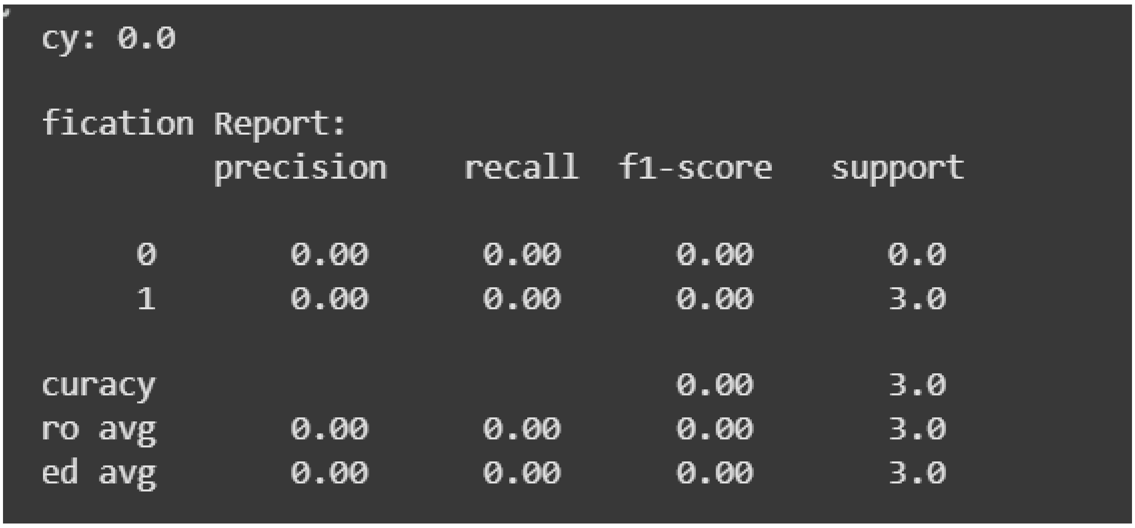
CODE:

```
# Install dependencies

!pip install pandas scikit-learn seaborn matplotlib


# Import libraries

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.model_selection import train_test_split

from sklearn.preprocessing import StandardScaler, LabelEncoder

from sklearn.ensemble import RandomForestClassifier

from sklearn.metrics import accuracy_score, confusion_matrix, classification_report


# Simulated Credit Score Dataset

data = {

    'Age': [25, 40, 50, 35, 23, 52, 46, 28, 55, 30],

    'Income': [50000, 80000, 120000, 60000, 40000, 150000, 100000, 52000, 130000, 58000],

    'LoanAmount': [20000, 30000, 40000, 15000, 10000, 50000, 25000, 12000, 45000, 17000],

    'LoanDuration': [12, 24, 36, 10, 8, 48, 20, 6, 40, 9],

    'CreditHistory': [1, 1, 0, 1, 0, 1, 0, 1, 1, 0],

    'Purpose': ['car', 'education', 'business', 'car', 'furniture', 'business', 'education', 'car', 'business', 'furniture'],

    'CreditRisk': ['good', 'good', 'bad', 'good', 'bad', 'good', 'bad', 'good', 'good', 'bad']

}
```

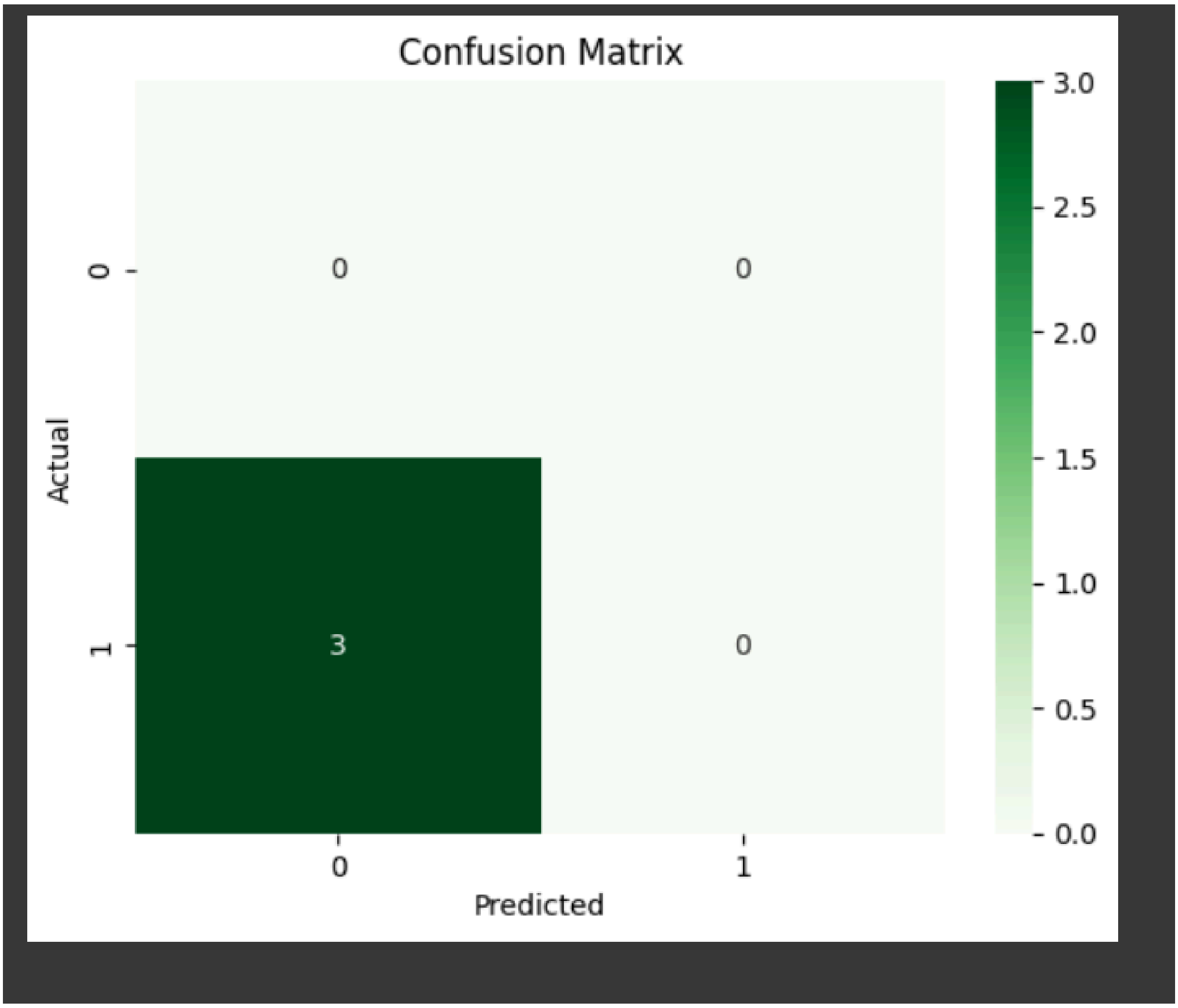
```
df = pd.DataFrame(data)

# Show first rows
print("Initial Data:")
print(df.head())

# Data Preprocessing
# Encode categorical columns
label_encoder = LabelEncoder()
df['Purpose'] = label_encoder.fit_transform(df['Purpose'])
df['CreditRisk'] = label_encoder.fit_transform(df['CreditRisk'])  # Target: good=1, bad=0

# Features and target
X = df.drop('CreditRisk', axis=1)
y = df['CreditRisk']

# Feature scaling
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# Train-test split
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.3,
random_state=42)

# Model - Random Forest Classifier
model = RandomForestClassifier(n_estimators=100, random_state=42)
```

```
model.fit(X_train, y_train)


# Predict and Evaluate
y_pred = model.predict(X_test)
print("\nAccuracy:", accuracy_score(y_test, y_pred))
print("\nClassification Report:\n", classification_report(y_test, y_pred))
# Confusion Matrix
cm = confusion_matrix(y_test, y_pred)
sns.heatmap(cm, annot=True, fmt='d', cmap='Greens')
plt.title('Confusion Matrix')
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.show()
```

RESULT :



```
cy: 0.0

fication Report:
          precision    recall  f1-score   support

     0       0.00      0.00      0.00       0.0
     1       0.00      0.00      0.00       3.0

curacy                           0.00       3.0
ro avg       0.00      0.00      0.00       3.0
ed avg       0.00      0.00      0.00       3.0
```



CREDIT

Dataset:

- The dataset used in this demonstration is simulated. However, similar datasets are available on:
  - Kaggle - Credit Scoring Dataset

Image:

- Credit Score Gauge Image by Pixabay
- Link: https://pixabay.com/illustrations/credit-score-rating-financial-2037295/

Libraries:

- Python libraries used: pandas, numpy, scikit-learn, seaborn, matplotlib.