EXPERIMENT – 11

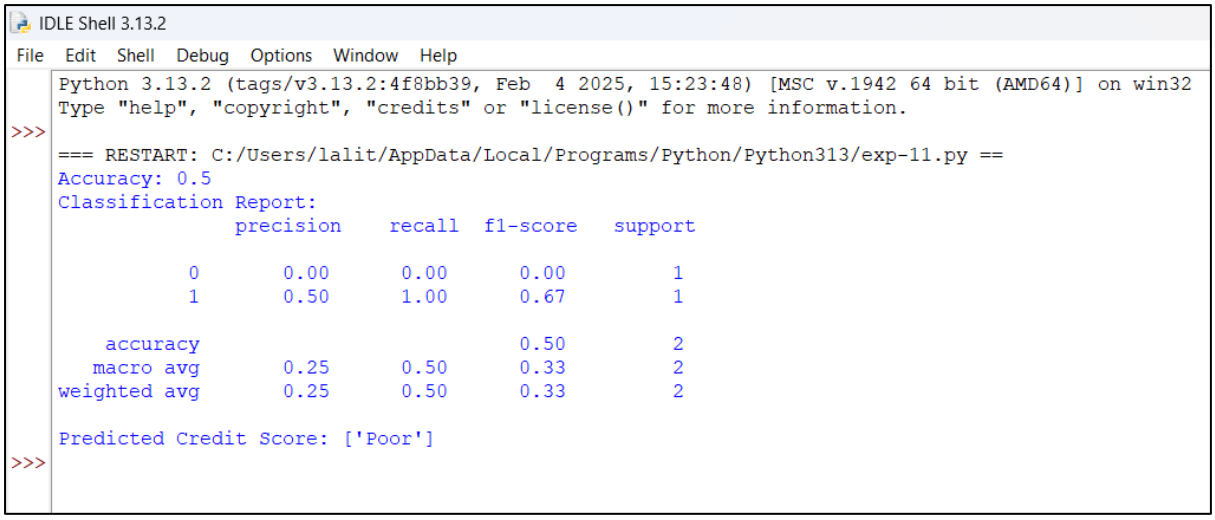
Write a program for the task of Credit Score Classification

CODE :

```
# Credit Score Classification using Logistic Regression
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, classification_report

# Sample dataset
data = {
    "Age": [25, 45, 35, 50, 23, 40, 60, 48],
    "Income": [30000, 80000, 50000, 90000, 28000, 65000, 120000, 70000],
    "Loan_Amount": [5000, 20000, 15000, 25000, 4000, 18000, 30000, 22000],
    "Credit_Score": ["Poor", "Good", "Average", "Good", "Poor", "Average", "Good",
"Average"]
}

df = pd.DataFrame(data)

# Encode target
le = LabelEncoder()
df["Credit_Score"] = le.fit_transform(df["Credit_Score"])

# Features & target
X = df[["Age", "Income", "Loan_Amount"]]
y = df["Credit_Score"]
```

```
# Train-test split
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.25, random_state=42
)

# Model training
model = LogisticRegression(max_iter=1000)
model.fit(X_train, y_train)

# Prediction
y_pred = model.predict(X_test)

# Output (warning fixed)
print("Accuracy:", accuracy_score(y_test, y_pred))
print("Classification Report:\n",
      classification_report(y_test, y_pred, zero_division=0))

# New customer prediction
new_customer = pd.DataFrame(
    [[30, 60000, 12000]],
    columns=["Age", "Income", "Loan_Amount"]
)

prediction = model.predict(new_customer)
print("Predicted Credit Score:", le.inverse_transform(prediction))
```

OUTPUT :


IDLE Shell 3.13.2
File Edit Shell Debug Options Window Help
Python 3.13.2 (tags/v3.13.2:4f8bb39, Feb 4 2025, 15:23:48) [MSC v.1942 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
=== RESTART: C:/Users/lalit/AppData/Local/Programs/Python/Python313/exp-11.py ==
Accuracy: 0.5
Classification Report:
precision recall f1-score support
0 0.00 0.00 0.00 1
1 0.50 1.00 0.67 1
accuracy 0.50 2
macro avg 0.25 0.50 0.33 2
weighted avg 0.25 0.50 0.33 2
Predicted Credit Score: ['Poor']
>>>