Experiment No. 3

Title: Implement Logistic Regression using any dataset.

Objectives:

1. To generate a synthetic dataset with two numerical features and a binary target variable.
2. To implement and evaluate a logistic regression model for classification.
3. To analyze model performance using accuracy, confusion matrix, and ROC-AUC score.
4. To visualize classification performance using the ROC curve.

Problem Statement:

The objective of this project is to build a logistic regression model that can predict a binary outcome based on two independent numerical variables. The model's effectiveness will be assessed using accuracy, confusion matrix, and ROC-AUC score, with additional visualization techniques to interpret results.

Outcomes:

1. A synthetic dataset with meaningful feature-target relationships.
2. A trained logistic regression model capable of predicting binary outcomes.
3. Performance evaluation through accuracy, confusion matrix, and ROC-AUC score.
4. Graphical representation of model performance using the ROC curve.

Tools Required: 4GB RAM, Anaconda, Notebook

Theory:

Logistic regression is a statistical method used for binary classification. It models the probability that a given input belongs to a particular class using the sigmoid function, defined as:

$$P(Y=1|X) = \frac{1}{1+e^{-(\beta_0+\beta_1 X_1+\beta_2 X_2)}}$$

where:

- P(Y=1|X)P(Y=1|X)P(Y=1|X) is the probability of the target variable being 1.
- β0\beta_0β0 is the intercept, and β1,β2\beta_1, \beta_2β1,β2 are coefficients for features X1,X2X_1, X_2X1,X2.
- The model optimizes these parameters using maximum likelihood estimation (MLE).

The performance of logistic regression is often assessed using:

- Accuracy: Ratio of correctly predicted samples.
- Confusion Matrix: Breakdown of TP, FP, TN, and FN.

- ROC Curve & AUC Score: Measures the model's ability to discriminate between classes.

Algorithm:

Step 1: Generate Synthetic Dataset

1. Generate two independent numerical features, X1 and X2.
2. Define a linear relationship with added noise to determine binary target Y.
3. Store data in a Pandas DataFrame.

Step 2: Data Preprocessing

1. Split data into training and testing sets.

Step 3: Train Logistic Regression Model

1. Initialize the logistic regression model.
2. Train the model using the training data.

Step 4: Make Predictions

1. Predict target values on training and testing sets.
2. Generate probability scores for ROC-AUC evaluation.

Step 5: Evaluate Model Performance

1. Compute accuracy, confusion matrix, and classification report.
2. Calculate the ROC-AUC score.
3. Plot the ROC curve.

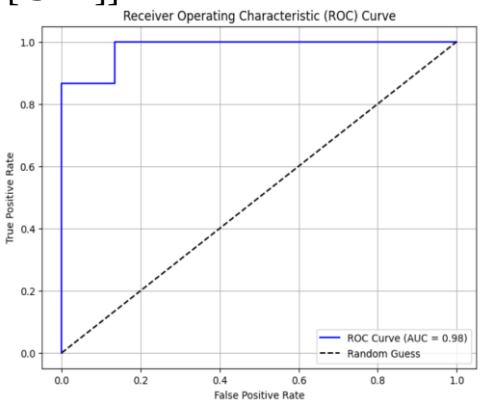
Source Code:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report,
  roc_curve, roc_auc_score
np.random.seed(42)
X1 = np.random.rand(100) * 10
X2 = np.random.rand(100) * 5
y = (2 * X1 + 3 * X2 + np.random.randn(100) * 2 > 20).astype(int)
data = pd.DataFrame({
'X1': X1,
'X2': X2,
'Y': y
})
X = data[['X1', 'X2']]
y = data['Y'] # Target
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
model = LogisticRegression()
model.fit(X_train, y_train)
y_train_pred = model.predict(X_train)
y_test_pred = model.predict(X_test)
y_test_proba = model.predict_proba(X_test)[:, 1]
```

```
train_accuracy = accuracy_score(y_train, y_train_pred)
test_accuracy = accuracy_score(y_test, y_test_pred)
conf_matrix = confusion_matrix(y_test, y_test_pred)
roc_auc = roc_auc_score(y_test, y_test_proba)
print(f"Train Accuracy: {train_accuracy:.2f}")
print(f"Test Accuracy: {test_accuracy:.2f}")
print(f"ROC-AUC Score: {roc_auc:.2f}")
print("\nConfusion Matrix:")
print(conf_matrix)
fpr, tpr, _ = roc_curve(y_test, y_test_proba)
plt.figure(figsize=(8, 6))
plt.plot(fpr, tpr, label=f"ROC Curve (AUC = {roc_auc:.2f})", color='blue')
plt.plot([0, 1], [0, 1], 'k--', label='Random Guess')
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Receiver Operating Characteristic (ROC) Curve')
plt.legend()
plt.grid()
plt.show()
```

Output:

Train Accuracy: 0.93
Test Accuracy: 0.90
ROC-AUC Score: 0.98
Confusion Matrix:
[[15 0]
 [3 12]]



Conclusion:

_____

_____

_____

_____