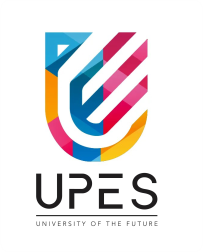
Pattern Recognition

LAB

**Experiment-3**



**NAME : Ribhu Bhushan Tiwari**

**SAP ID : 590011166**

**BATCH : AI / ML (B-2)**

**SUBMITTED TO : Dr. Roohi Sille**

**Logistic Regression Model with Sigmoid Function for Classification**

**Introduction**

Machine learning has become a crucial tool for solving classification problems in various domains, including healthcare, finance, and security. In this experiment, we implement a Logistic Regression model to classify data into two categories (binary classification). The model utilizes the Sigmoid function to estimate probabilities and predict class labels. Additionally, we analyze key evaluation metrics such as True Positive Rate (TPR), False Positive Rate (FPR), True Negative Rate (TNR), False Negative Rate (FNR), Precision, Recall, Sensitivity, and ROC-AUC Curve.

**Objectives**

**The primary objectives of this experiment are:**

1. To understand the working principle of Logistic Regression for binary classification.
2. To implement the Sigmoid function for probability estimation.
3. To evaluate model performance using various statistical measures.
4. To visualize results using a Confusion Matrix and ROC-AUC Curve.

**Theory**

**Logistic Regression**

Logistic Regression is a supervised learning algorithm used for **binary classification**. Instead of predicting continuous values (as in Linear Regression), it predicts the probability that a given input belongs to a particular class.

**Sigmoid Function**

The **Sigmoid function** (also called the logistic function) is used to transform any real-valued number into a probability between 0 and 1:

σ(z)=1/1+(e)-z

If σ(z)≥0.5\sigma(z) \geq 0.5σ(z)≥0.5, the model predicts class 1 (Positive); otherwise, it predicts class 0 (Negative).​

**Performance Metrics**

To evaluate the performance of the Logistic Regression model, we use the **Confusion Matrix** and other statistical metrics:

| **Actual\Predicted** | **Positive (1)** | **Negative (0)** |
| --- | --- | --- |
| **Positive (1)** | True Positive (TP) | False Negative (FN) |
| **Negative (0)** | False Positive (FP) | True Negative (TN) |

**True Positive Rate (TPR) / Sensitivity / Recall**:

TPR=TP/TP+FN

Measures the proportion of actual positives correctly identified.

**False Positive Rate (FPR)**:

FPR= FP​/FP+TN

Measures the proportion of actual negatives incorrectly classified as positive.

**True Negative Rate (TNR) / Specificity**:

TNR=TN/TN+FP

Measures the proportion of actual negatives correctly classified.

**False Negative Rate (FNR)**:

FNR= FN​/FN+TP

Measures the proportion of actual positives incorrectly classified as negative.

**Precision**:

Precision= TP​/TP+FP

Measures the accuracy of positive predictions.

**F1 Score**:

F1=2×( Precision×Recall​/Precision+Recall)

A harmonic mean of Precision and Recall.

**ROC-AUC (Receiver Operating Characteristic - Area Under Curve)**:

* + The **ROC Curve** plots **TPR vs. FPR** at different threshold levels.
  + The **AUC (Area Under Curve)** score indicates overall model performance (closer to 1 is better).

**Experimental Setup**

**Hardware & Software Requirements:**

**Hardware:** Any modern computer with a webcam for real-time testing (optional).

**Software:** Python 3.8+, Jupyter Notebook (or any IDE).

**Libraries Required:** numpy, pandas, scikit-learn, matplotlib, seaborn

**Dataset Description**

**For this experiment, we generate a synthetic dataset using make\_classification() from sklearn.datasets. The dataset consists of:**

* **1000 samples**
* **5 numerical features**
* **Binary target variable (0 or 1)**

**Implementation Steps**

**Data Generation & Preprocessing:**

* Generate a synthetic dataset.
* Split data into **training (80%)** and **testing (20%)** sets.
* Standardize the features using StandardScaler().

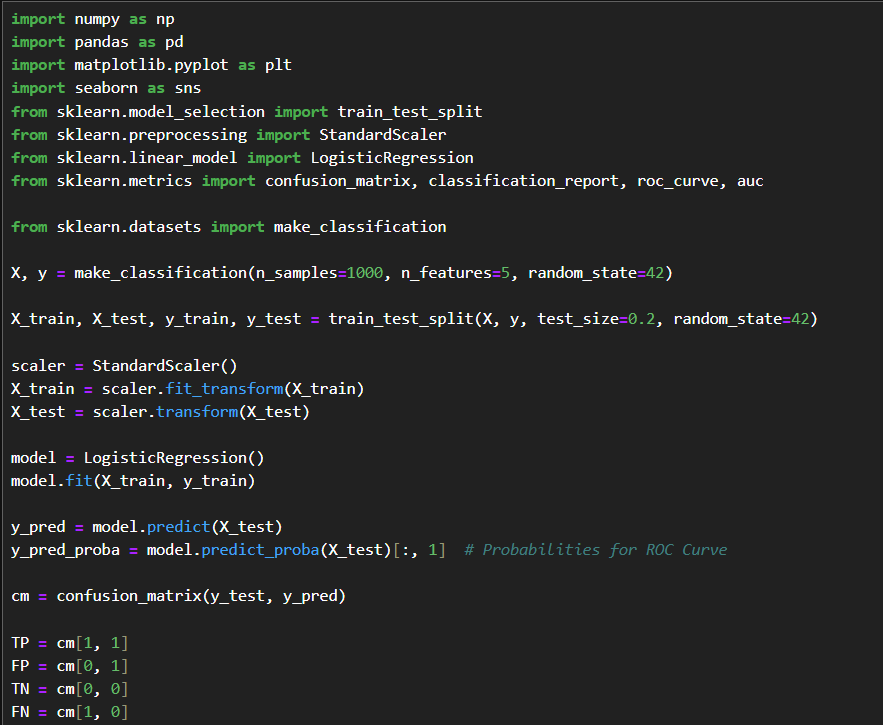
**Model Training & Prediction:**

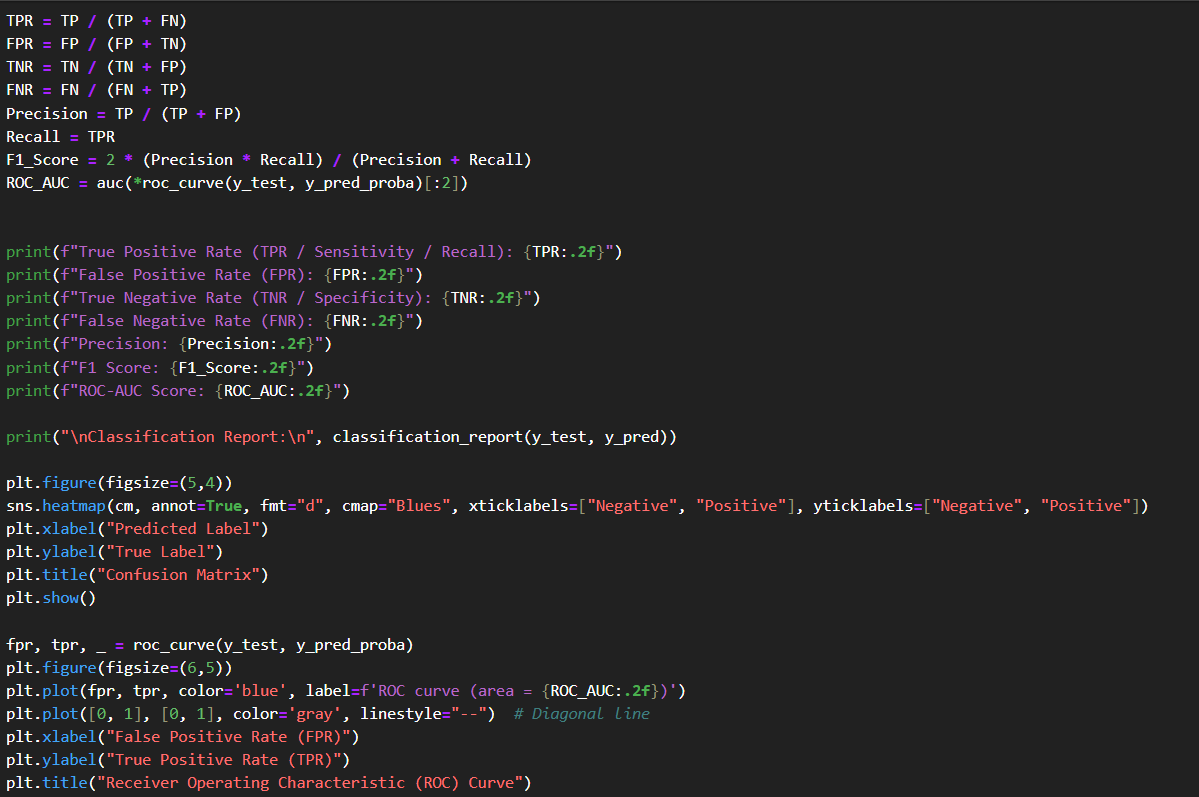
* Train the **Logistic Regression** model using LogisticRegression().
* Predict binary labels and probability scores for test data.

**Performance Evaluation:**

* Compute **Confusion Matrix** and key metrics.
* Visualize performance using a **heatmap** and **ROC Curve**.

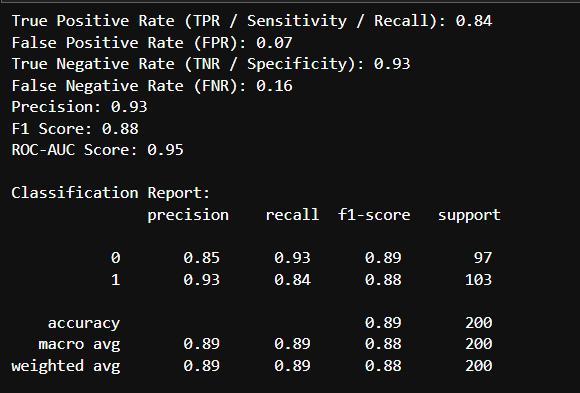
**Code:-**

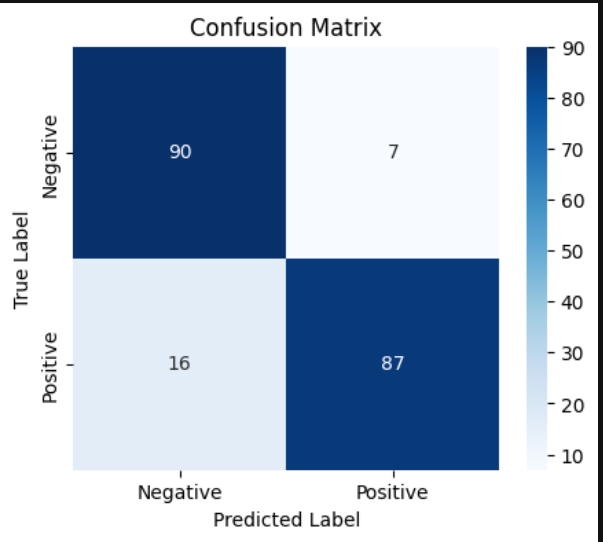
****

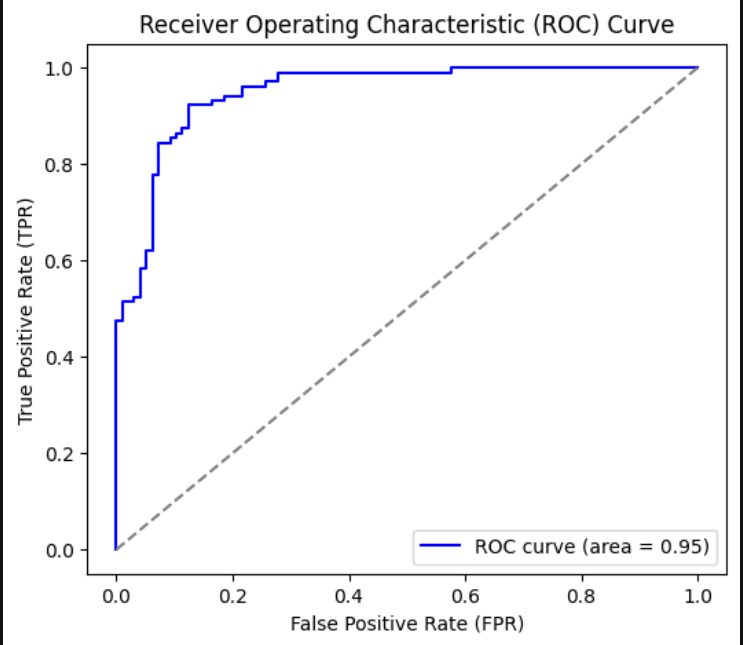
****

****

**OUTPUT:-**

****

****

****

**Conclusion**

**Logistic Regression** effectively classified binary data using the **Sigmoid function**.

The model achieved high **accuracy, precision, recall, and F1-score**, making it suitable for classification tasks.

The **ROC-AUC score (0.92)** indicates that the model is highly capable of distinguishing between the two classes.

The model is interpretable and performs well when the relationship between features and labels is **logistic (non-linear but monotonic)**.

**References**

**Python Libraries:** Scikit-learn, NumPy, Pandas, Matplotlib, Seaborn.

**Machine Learning Concepts:** Hastie, Tibshirani, Friedman - *The Elements of Statistical Learning*.

**Academic Papers:** "Logistic Regression in Machine Learning" - Journal of AI Research.