Make sure you have the required libraries installed:

pip install pandas scikit-learn matplotlib

**✅ Step 2: Import Libraries**

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

from sklearn.model\_selection import train\_test\_split

from sklearn.preprocessing import StandardScaler

from sklearn.linear\_model import LogisticRegression

from sklearn.metrics import confusion\_matrix, classification\_report, roc\_auc\_score, roc\_curve

**✅ Step 3: Load Dataset**

You can either:

* Use your own dataset (CSV file).
* Or load a sample dataset from sklearn.datasets.

Example using the **Breast Cancer dataset** (binary classification):

from sklearn.datasets import load\_breast\_cancer

data = load\_breast\_cancer()

df = pd.DataFrame(data.data, columns=data.feature\_names)

df['target'] = data.target # 0 = malignant, 1 = benign

print(df.head())

**✅ Step 4: Split Features and Target**

X = df.drop('target', axis=1)

y = df['target']

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.3, random\_state=42)

**✅ Step 5: Standardize Features**

scaler = StandardScaler()

X\_train = scaler.fit\_transform(X\_train)

X\_test = scaler.transform(X\_test)

**✅ Step 6: Train Logistic Regression Model**

model = LogisticRegression()

model.fit(X\_train, y\_train)

**✅ Step 7: Make Predictions**

y\_pred = model.predict(X\_test)

y\_pred\_proba = model.predict\_proba(X\_test)[:,1] # Probabilities for ROC

**✅ Step 8: Evaluate Model**

* **Confusion Matrix**

cm = confusion\_matrix(y\_test, y\_pred)

print("Confusion Matrix:\n", cm)

* **Classification Report**

print(classification\_report(y\_test, y\_pred))

* **ROC-AUC Score**

roc\_score = roc\_auc\_score(y\_test, y\_pred\_proba)

print("ROC-AUC Score:", roc\_score)

**✅ Step 9: Plot ROC Curve**

fpr, tpr, thresholds = roc\_curve(y\_test, y\_pred\_proba)

plt.plot(fpr, tpr, color='blue', label='ROC Curve (AUC = %.2f)' % roc\_score)

plt.plot([0,1], [0,1], color='red', linestyle='--')

plt.xlabel('False Positive Rate')

plt.ylabel('True Positive Rate')

plt.title('ROC Curve')

plt.legend()

plt.show()

**✅ Step 10: Tune Threshold**

Default threshold = **0.5**. You can adjust it:

custom\_threshold = 0.4

y\_custom = (y\_pred\_proba >= custom\_threshold).astype(int)

print(confusion\_matrix(y\_test, y\_custom))

**Core Concept Reminder**

* Logistic regression uses **sigmoid function**:

σ(z)=11+e−z\sigma(z) = \frac{1}{1+e^{-z}}σ(z)=1+e−z1​

where z=w0+w1x1+⋯+wnxnz = w\_0 + w\_1x\_1 + \dots + w\_nx\_nz=w0​+w1​x1​+⋯+wn​xn​.

👉 Do you want me to **give you the full code in one block** (ready to run), **or break it into sections with explanations in a Jupyter Notebook style**, or **generate a PDF report with all steps and explanations**?

Top of Form

Bottom of Form

ChatGPT can make mistakes. Check important info. See Cookie