Task 4: Classification with Logistic Regression.

Task 4: Classification with Logistic Regression — Full Explanation

✅ Objective

The goal of this task is to **build a binary classification model** using **Logistic Regression**.

Binary classification means the output has **two classes**, for example:

Spam vs Not Spam

Disease vs No Disease

Pass vs Fail

0 vs 1

Logistic Regression predicts the probability that a given input belongs to class **1**.

Tools Used

🧰 1. Scikit-learn

A Python machine learning library used to:

split the dataset

train Logistic Regression

evaluate the model

🧰 2. Pandas

Used for:

loading datasets

cleaning and preparing data

working with tables

🧰 3. Matplotlib

Used for:

visualizing results

plotting ROC curves, confusion matrix, etc.

Hints / Mini Guide Explanation

1. Choose a binary classification dataset

Pick any dataset where the target column has **two classes** (0/1).
Common examples:
Titanic survival
Diabetes prediction
Heart disease dataset

2. Train/test split and standardize features

Split data into:
Training set (usually 70–80%)
Testing set (20–30%)

Then **standardize (normalize) features** so they all have equal scale, e.g. using:
`StandardScaler()`

3. Fit a Logistic Regression model

Train the model using:
`LogisticRegression()`

```
The model learns patterns in
the data to predict class
labels.
```

4. Evaluate Model Performance

You need to evaluate how good the model is.

Confusion Matrix

Shows:
True Positives

True Negatives
False Positives
False Negatives

Precision

Of all predicted positives, how many were correct?

Useful when **false positives** are costly.

Recall

Of all actual positives, how many did the model detect?

Useful when **false negatives** are costly.

ROC-AUC

Measures how well the model separates the two classes.
Higher AUC → Better classifier.

5. Tune threshold & explain sigmoid function

Sigmoid Function

Logistic Regression uses the **sigmoid** to convert numbers into probabilities:

```
\sigma(z) = \frac{1}{1 + e^{-z}}
```

`Output is between` **0 and 1**.

Threshold

Default threshold = **0.5**
If probability ≥ 0.5 Class 1
If probability < 0.5 Class 0
You can change the threshold to improve precision/recall.

In Simple Words

This task teaches you how to:

1 Choose a binary dataset

2 Split and scale data
3 Build and train Logistic Regression
4 Evaluate using multiple metrics
Understand sigmoid and probability
5 threshold

If you want, I can also provide: ✅ Full code example
✅ Project report format
✅ Output explanations
Just tell me!