✅ Task 4: Classification with Logistic Regression — Full Explanation

📌 Objective

Build a **binary classifier** (two-class prediction) using **Logistic Regression**.
Example:
Predict whether a person has diabetes (Yes/No)
Predict if an email is spam (Spam/Not Spam)
Predict if a customer will buy a product (Buy/Not Buy)

🛠 Tools Required

Scikit-learn → For ML model training

Pandas → For data loading + processing
Matplotlib → For plotting ROC curve, visualizations

🧭 Step-by-Step Mini Guide (Explained in Detail)

1️⃣ Choose a Binary Classification Dataset

A binary dataset means the target has **two classes** such as:

Example	**Classes**
Diabetes	0 = No, 1 = Yes
Spam email	0 = Not spam, 1 = Spam
Titanic survival	0 = No, 1 = Yes

Common datasets in scikit-learn:

```
from sklearn.datasets import
load_breast_cancer
data = load_breast_cancer()
Or load your own CSV with
Pandas.
```

2 Train-Test Split + Standardize Features

Train-test split

We divide data into:

Training set (80%) → teach the model

Test set (20%) → evaluate performance

```
from sklearn.model_selection
import train_test_split


X_train, X_test, y_train,
y_test = train_test_split(X,
y, test_size=0.2,
random_state=42)
```

Standardization

Logistic Regression performs better when features are scaled.

```
from sklearn.preprocessing
import StandardScaler


scaler = StandardScaler()


X_train =
scaler.fit_transform(X_train)
X_test =
scaler.transform(X_test)
```

3 Fit Logistic Regression Model

```
from sklearn.linear_model
import LogisticRegression
```

```
model = LogisticRegression()
model.fit(X_train, y_train)
```

This learns the relationship between input features and the binary output.

4 Evaluate Model Performance

✔ Confusion Matrix

Shows:

True Positive (TP)

True Negative (TN)

False Positive (FP)

False Negative (FN)

```
from sklearn.metrics import confusion_matrix, classification_report


y_pred = model.predict(X_test)
print(confusion_matrix(y_test, y_pred))
print(classification_report(y_test, y_pred))
```

✔ Precision

Out of predicted “positive”, how many are correct?

✔ Recall

Out of actual “positive”, how many we detected correctly?

✔ ROC-AUC score

Measures how well model separates classes.

```
from sklearn.metrics import roc_auc_score, roc_curve
y_prob = model.predict_proba(X_test)[:,1]

roc_auc = roc_auc_score(y_test, y_prob)
```

✔ ROC Curve Plot

```
import matplotlib.pyplot as plt
```

```
fpr, tpr, _ =
roc_curve(y_test, y_prob)

plt.plot(fpr, tpr)
plt.xlabel("False Positive
Rate")
plt.ylabel("True Positive
Rate")
plt.title("ROC Curve")
plt.show()
```

5️⃣ Tune Threshold + Explain Sigmoid Function

✔ Sigmoid Function

Logistic Regression outputs a **probability** using:

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

This converts any number into a value between **0 and 1**.

✔ Default Threshold = 0.5

If probability > 0.5 → Predict Class 1

Else → Predict Class 0

✔ Tuning Threshold

Useful when:

You want fewer false negatives (e.g., detecting disease)

You want fewer false positives (e.g., spam filter)

Example:

```
threshold = 0.3
y_custom = (y_prob >
threshold).astype(int)
```

🎯 Final Summary

Step	What You Do
1	Pick dataset
2	Train-test split + standardize
3	Train logistic regression

4	Evaluate using confusion matrix, precision, recall, ROC-AUC
5	Tune decision threshold & understand sigmoid