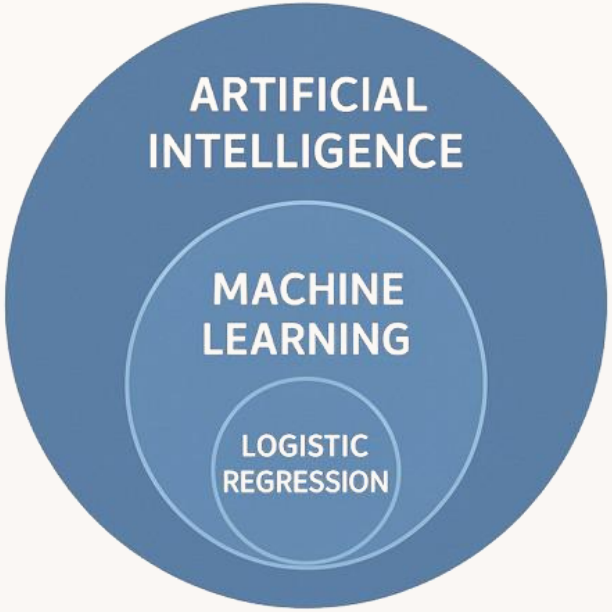


INTRODUCTION TO MACHINE LEARNING WITH LOGISTIC REGRESSION USING PYTHON & THE IRIS DATASET

LEARNING OBJECTIVES

- Understand what Logistic Regression is and why it's important & Know how it differs from Linear Regression.
- See how to implement it in Python with the Iris dataset.
- Learn how to evaluate model performance.

LOGISTIC REGRESSION

1. Introduction

Imagine you're trying to predict whether a customer will buy a product (1) or not (0) based on their age and income. This is a binary decision — either yes or no."

2. Why Not Linear Regression?

If we used linear regression, we might get predictions like 1.2 or -0.3 — which don't make sense for a yes/no decision. We need a model that gives probabilities between 0 and 1."

3. Logistic Function (Sigmoid Curve)

That's where logistic regression comes in. Instead of predicting a number, it predicts the probability of class 1, in this case Setosa (1).

$z = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_4 x_4 \cdots + \beta_n x_n$

X_1: Sepal length (cm), X_2: Sepal width (cm), X_3: Petal length (cm), X_4: Petal width (cm)

β1, β2, β3, β4: Coefficients (weights) showing how much each feature influences the prediction

β0: Intercept —The **intercept**, or bias term — it's the base prediction when all features are 0. **It shifts the sigmoid curve left or right** on the x-axis.

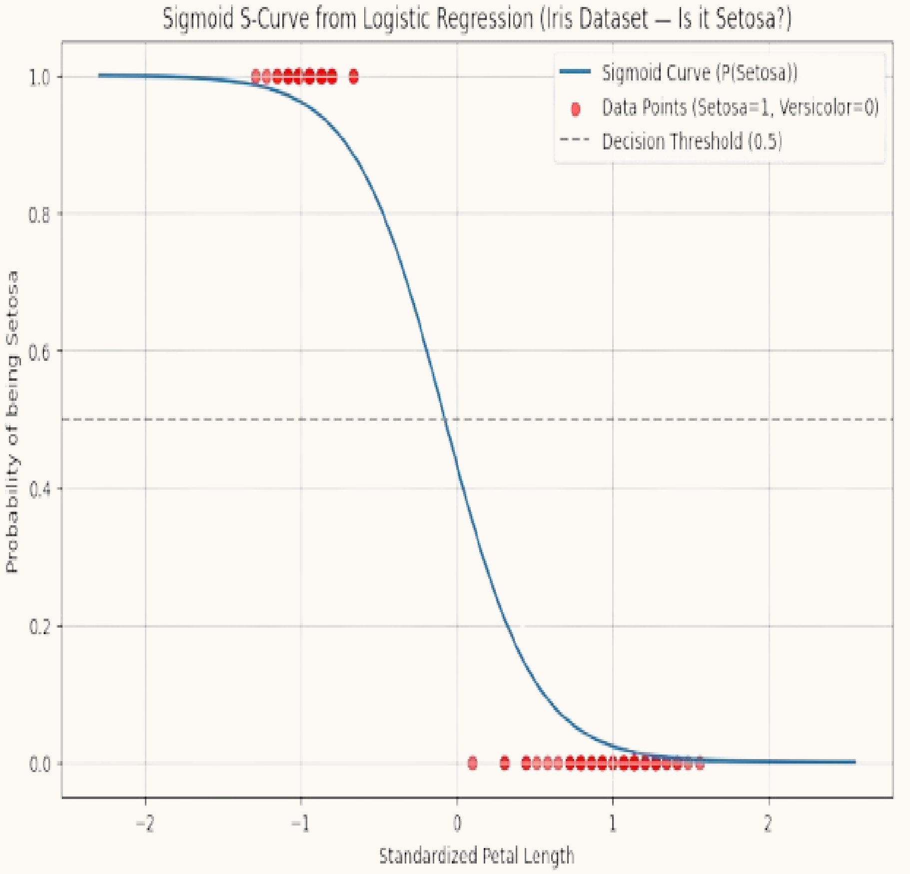
The result — z — is passed through a special S-shaped curve:

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

📈 This curve transforms the score into a value between 0 and 1 — a probability!

If the probability is above 0.5 → we predict the flower is Setosa

If it's below 0.5 → we predict it is not Setosa



PYTHON IMPLEMENTATION WITH IRIS DATA SET

Step 1

Import Libraries

Load scikit-learn, pandas, and and numpy for data handling and handling and modeling.

Step 2

Prepare Iris Data

Split the dataset into training and and testing sets.

Step 3

Train the Model

Fit logistic regression to predict predict iris species.

Step 4

Make Predictions

Apply the model to test data and and evaluate results.

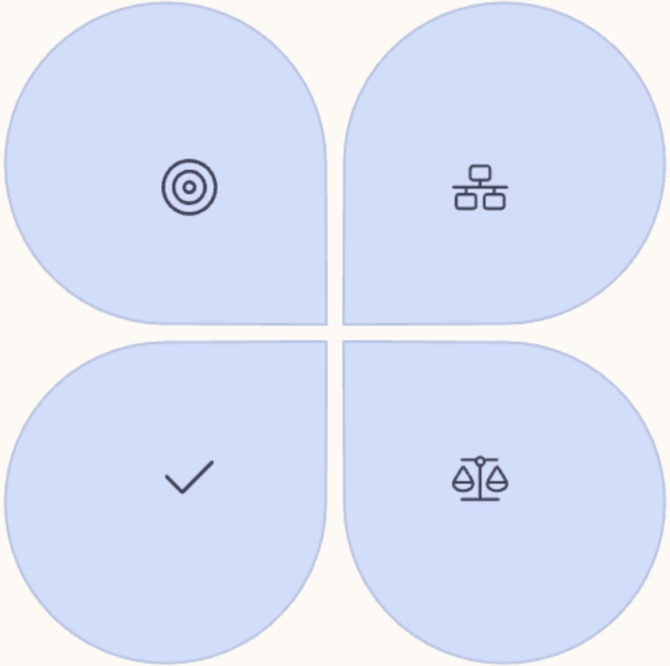
EVALUATION METRICS

Precision

Out of predicted positives, how many were correct?

Recall

Out of actual positives, how many did we catch?



Accuracy

Overall correctness, how many predictions were correct?

F1-Score

Harmonic mean balancing precision and recall.

EVALUATION METRICS WITH IRIS DATA SET

🔍 **What This Means**

• **20 True Negatives (TN)**
→ The model correctly predicted 20 flowers as "Not Setosa"

• **0 False Positives (FP)**
→ The model didn't mistakenly say "Setosa" when it wasn't — great!

• **0 False Negatives (FN)**
→ The model didn't miss any real Setosa flowers — also great!

• **10 True Positives (TP)**
→ The model correctly predicted 10 flowers as "Setosa"

✅ **Model Performance Summary**

This is a **perfect prediction**. Why?

• **Accuracy** = (TP + TN) / Total = (10 + 20) / 30 = **100%**

• **Precision** = TP / (TP + FP) = 10 / (10 + 0) = **100%**

• **Recall** = TP / (TP + FN) = 10 / (10 + 0) = **100%**

• **F1-Score** = 2 x (Precision x Recall) / (Precision + Recall) = 2 x (1x1) / (1 + 1) = 1.0(100%)

```
Predicted labels: [0 0 1 1 1 0 0 0 1 0 1 0 1 0 0 0 0 0 1 0 1 0 0 0 0 0 0 1 0 1]
Predicted probabilities: [1.09971375e-03 2.05284631e-03 9.74877314e-01 9.54396717e-01
 9.83704412e-01 3.62934376e-03 3.26804822e-03 1.08945229e-03
 9.63086537e-01 1.70078601e-02 9.76600431e-01 2.29767182e-03
 9.72331882e-01 1.39569120e-02 1.93997067e-02 9.14058434e-04
 1.23128639e-01 5.81301846e-03 9.78057039e-01 8.10347431e-03
 9.94087230e-01 2.09199834e-03 3.34017667e-03 1.46162023e-03
 2.07007947e-02 1.38116408e-03 4.29733520e-03 9.73976081e-01
 4.57494133e-04 9.76589453e-01]
🔹 Classification Report:

              precision    recall  f1-score   support

           0       1.00      1.00      1.00        20
           1       1.00      1.00      1.00        10

    accuracy                           1.00        30
   macro avg       1.00      1.00      1.00        30
weighted avg       1.00      1.00      1.00        30

🔹 Confusion Matrix:

[[20  0]
 [ 0 10]]
🔹 ROC AUC Score: 1.0000
```

	Predicted: Not Setosa (0)	Predicted: Setosa (1)
Actual: Not Setosa (0)	20 (✅ TN)	0 (❌ FP)
Actual: Setosa (1)	0 (❌ FN)	10 (✅ TP)

CONCLUSION & TAKEAWAYS

Let's wrap up with a quick recap. If you remember just 3 things, let it be these:

🎯 **Logistic Regression** is:

- A way to predict **yes/no outcomes** using probabilities
- Based on the **sigmoid function**, which maps numbers to 0–1
- Easy to implement and **easy to explain**, which makes it great for first models

To summarize, logistic regression is simple, interpretable, and very effective for binary classification problem. It is often the first model you should try when approaching a classification task.

🔭 **Up Next — What to Learn After This:**

- **ROC Curve & AUC Score** to measure how well the model separates classes
- **Multinomial Logistic Regression** — when you have 3 or more classes

THANK YOU

👏 Thanks for being an awesome class — Let’s keep learning, experimenting, exploring, building and don’t be afraid to ask questions.

See you in the next class!