

Logistic Regression

What is Logistic Regression?

Logistic Regression is a **statistical model** used for solving **binary classification problems**, though it can be extended to multiclass problems (e.g., One-vs-All).

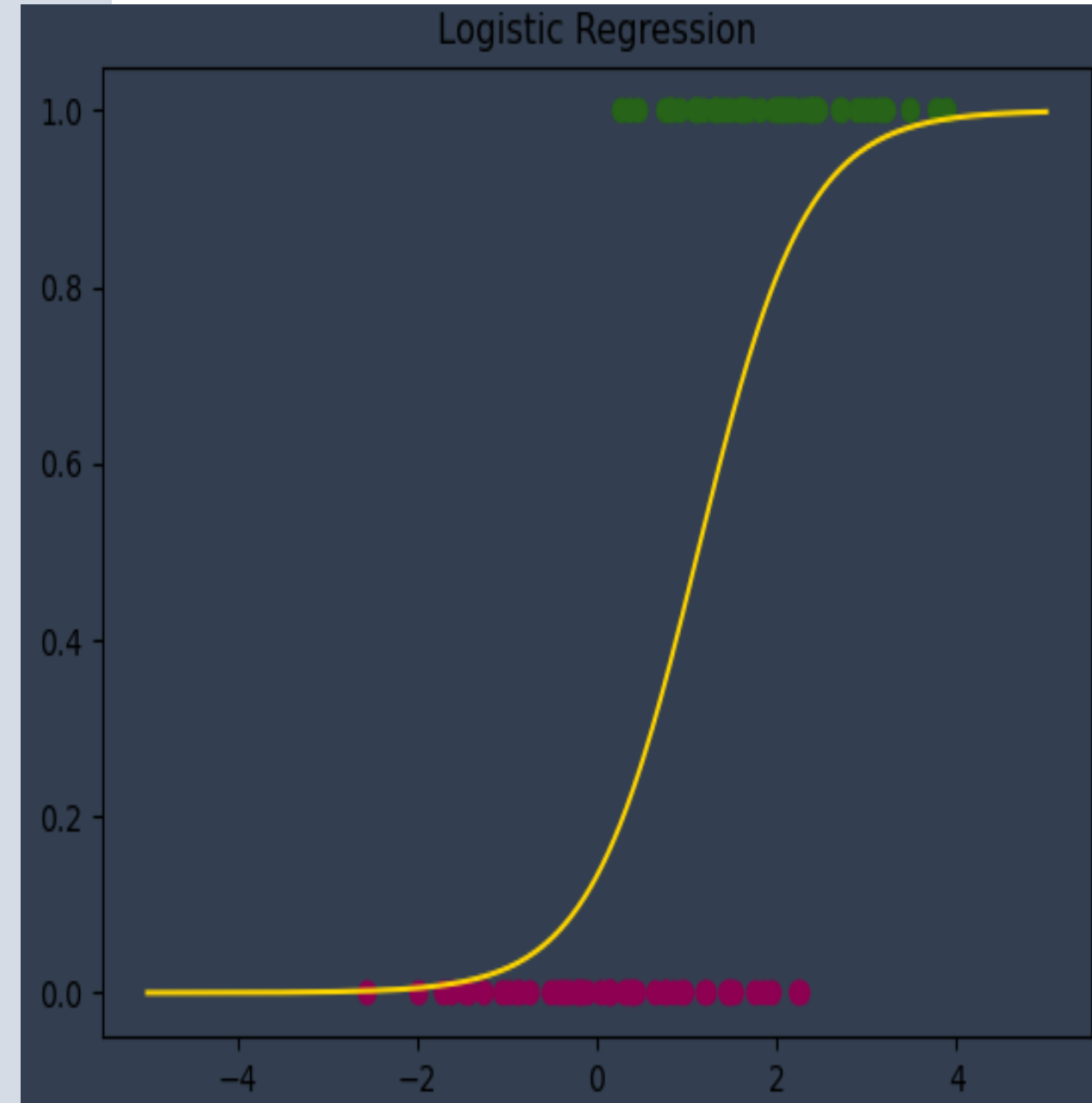
It predicts the probability that a given input belongs to a specific class.

The model uses the **sigmoid function** to map any input value to a probability between 0 and 1.

Why Use Logistic Regression?

Simple and interpretable. Effective for linearly separable datasets.

Outputs probabilities, which are useful for decision-making.



Logistic Regression

The Sigmoid Function:

The core of logistic regression is the sigmoid function which are useful for decision-making.

Input (z): The linear combination of features and weights
($z = w_1x_1 + w_2x_2 + \dots + b$).

Output: A probability between 0 and 1.

Decision Boundary:

Logistic regression uses a threshold of 0.5 to classify:

Probability > 0.5 : Class 1.

Probability ≤ 0.5 : Class 0.

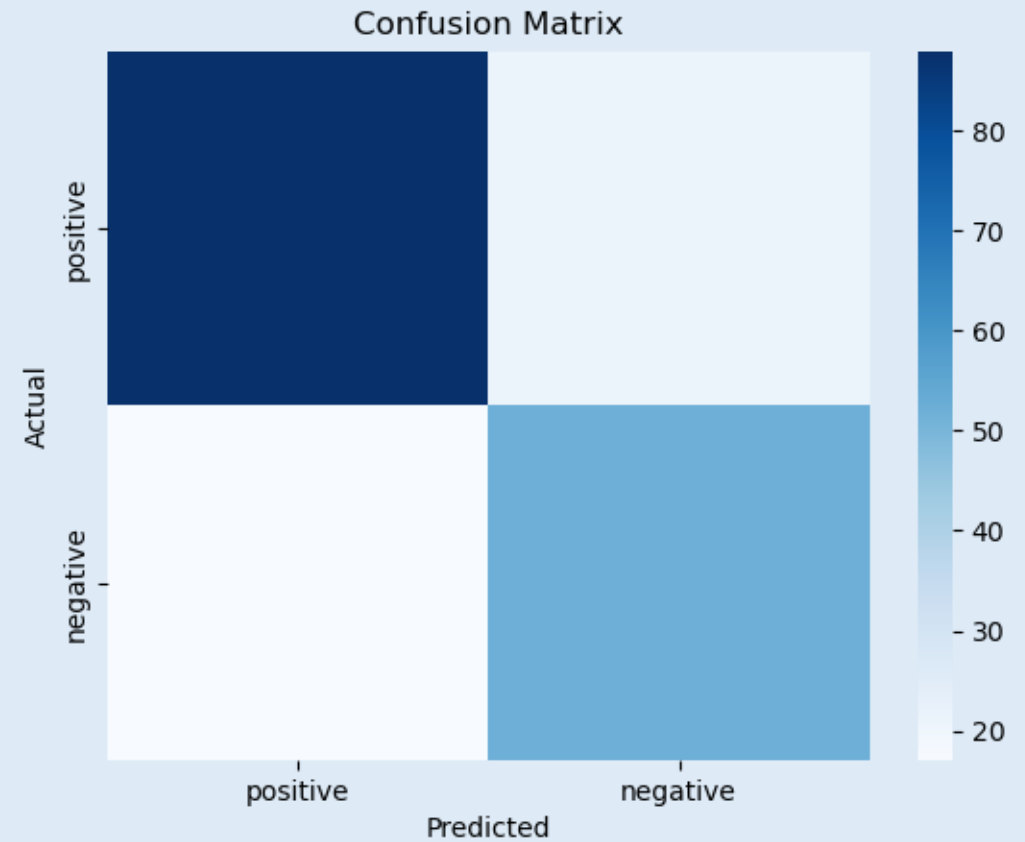
Example: Predicting if an email is spam (1) or not spam (0).

What is a Confusion Matrix?

- A confusion matrix is a **summary table** used to evaluate the performance of a classification model.
- It compares the **actual labels** with the **predicted labels**.

$$Precision = \frac{TP}{TP + FP}$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$



What Are Categorical Codes?

A method in Pandas where categorical values are **assigned unique numeric codes**.

Advantages:

- Quick and easy to use directly in Pandas.
- Works well for exploratory data analysis.

```
df['category'].astype('category').cat.codes
```

What is Label Encoding?

A method from sklearn.preprocessing where categories are encoded as integers.

Advantages:

- Built-in support for reversing the encoding using `inverse_transform`.
- Prevents unintended ordinal relationships in most cases.

```
from sklearn.preprocessing import LabelEncoder  
encoder = LabelEncoder()  
df['category_encoded'] = encoder.fit_transform(df['category'])
```