**Linear Regression**

**Meaning:**

Linear regression is one of the most fundamental and widely used algorithms in statistics and machine learning. It is a supervised learning algorithm used for predicting a continuous dependent variable based on one or more independent variables (also known as features). The core idea is to find a linear relationship between the input variables 𝑋 and the output variable 𝑦.

In simple linear regression (with one feature), the model attempts to fit a straight line:

𝑦 = 𝑤0 + 𝑤1𝑥

In multiple linear regression (with multiple features), the equation becomes:

𝑦=𝑤0 + 𝑤1𝑥1 +𝑤2𝑥2 + ⋯ + 𝑤𝑛𝑥n

Where:

* y is the predicted value,
* x1​, x2​, …, xn​ are the input features,
* w1, w2, …, wnw\_1, w\_2, \dots, w\_nw1​, w2​, …, wn​ are the weights (coefficients) learned during training,
* w0w\_0w0​ is the bias (intercept).

**Use Cases:**

Linear regression is widely used in real-world applications where the goal is to estimate or predict a numerical value. Some examples include:

* Predicting housing prices based on size, location, and number of rooms.
* Forecasting sales revenue based on advertising expenditure.
* Estimating temperatures, demand, or consumption in various industries.

**Loss Function:**

Linear regression is trained using a loss function known as the **Mean Squared Error (MSE)**, which measures the average squared difference between the actual and predicted values. The goal of training is to minimize this loss.

Where:

* m is the number of training examples,
* yi is the actual value,
* y^i the predicted value.

By minimizing MSE, the model finds the best-fitting line that reduces the overall error.

**Logistic Regression**

**Meaning:**

Despite its name, logistic regression is a classification algorithm, not a regression algorithm. It is used when the dependent variable is categorical, typically for binary classification (i.e., where the output is 0 or 1).

Instead of predicting a continuous value, logistic regression predicts the **probability** that a given input belongs to a certain class. It uses the **sigmoid (logistic) function** to map predicted values to a probability between 0 and 1.

The sigmoid function is defined as:

The model predicts , we classify the input as class 1; otherwise, class 0.

**Use Cases:**

Logistic regression is commonly used in binary classification tasks, such as:

* Spam detection (spam vs. not spam),
* Disease prediction (positive vs. negative),
* Customer churn (churn vs. retain).

It can also be extended to **multiclass classification** using techniques like **one-vs-rest (OvR)** or **softmax regression**.

**Loss Function:**

Logistic regression uses the **Binary Cross-Entropy Loss** (also called Log Loss), which measures the dissimilarity between predicted probabilities and the actual class labels. The function is:

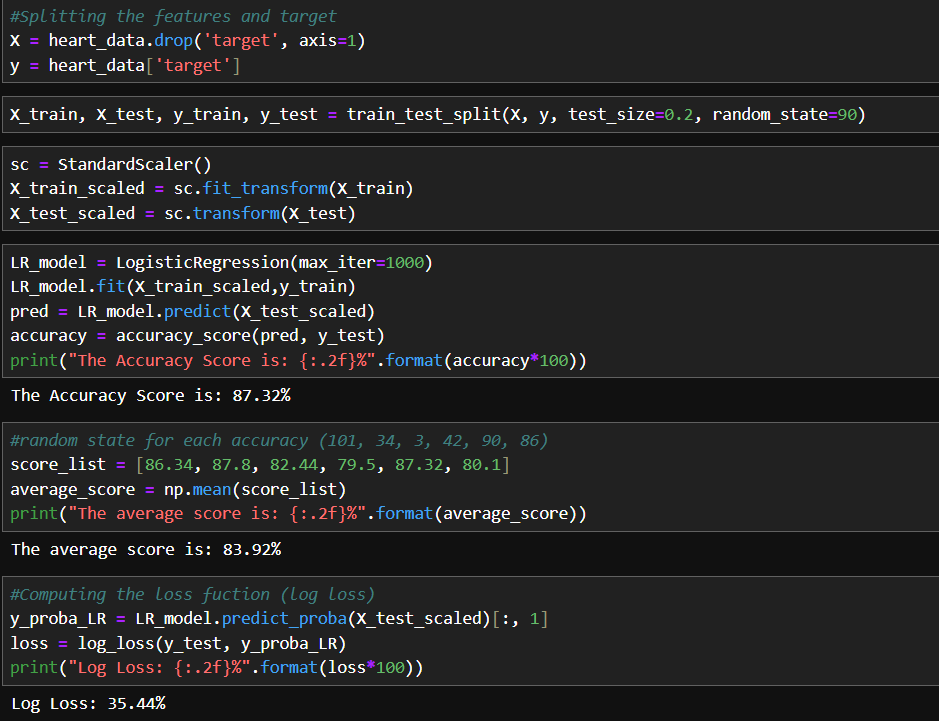
Where:

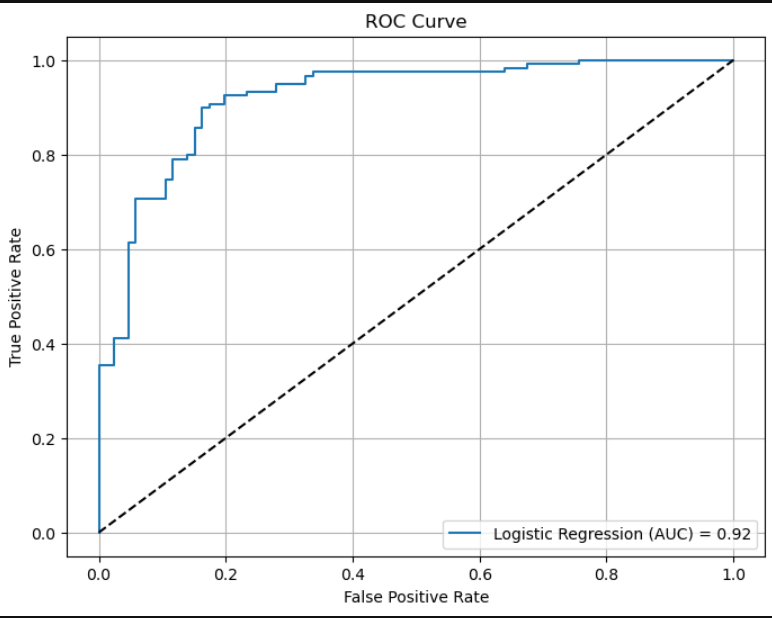
* yi is the actual class (0 or 1),
* yi^​ is the predicted probability.

The model adjusts its parameters to minimize this loss, thus improving the accuracy of its predictions.

Here's a clean, well-structured, beginner-friendly implementation of Logistic Regression from scratch in Python, using only NumPy (no scikit-learn). It's better than most because it includes:

* A clear training loop
* Sigmoid function
* Binary cross-entropy loss
* ROC Curve plot

****

****