## **Challenge #7: Perceptron Learning Visualization**

### **Objective**

To visualize the learning process of a Perceptron on a 2D plane, dynamically illustrating how the decision boundary (a line) changes as weights are updated via the Perceptron learning rule.

## **1. Prompts Used**

* "Write Python code to animate a 2D perceptron learning process with matplotlib."
* "Show step-by-step how weights are updated when using the perceptron rule on a linearly separable dataset."
* "Visualize decision boundary updates after each epoch of training."
* "Use FuncAnimation to show the learning process over iterations."

## **2. Step-by-Step Documentation**

### **Step 1: Dataset Creation**

* + Generated a 2D dataset with X as input and linearly separable y = ±1 labels.
  + Decision boundary for ground truth: x + y = 0.

### **Step 2: Perceptron Initialization**

* + Input is augmented with bias → shape becomes (100, 3).
  + Initial weights set to zero.

### **Step 3: Learning Process**

* + For each misclassified point, apply:  
     
  + Each weight update is stored to track evolution over time.

### **Step 4: Visualization**

* + Used matplotlib.animation.FuncAnimation to show:
    - Data points (blue for +1, red for -1).
    - Decision boundary (black dashed line).
    - Animation of boundary update after each misclassification.

## **3. Output (Graphical Proof)**

* The animation dynamically shows the perceptron’s decision line gradually rotating and translating until it correctly separates the two classes.
* Final frame shows a decision boundary that cleanly separates red and blue points.  
  Useful for understanding convergence and dynamics of the perceptron learning algorithm.

