Rules for Decision:

If x1=1 and x2=1, the prediction is y=0.  
If x1=1 and x2​=0, the prediction is y=1.  
If x1=0 and x2​=1, the prediction is y=1.  
If x1=0 and x2​=0, the prediction is y=0.

**3-NN Base Learner**

To calculate the 3-NN predictions, we need to compute the Euclidean distances between each test point and the training points, then classify based on the majority of the 3 nearest neighbors.

**1st Test Point (x1=0,x2=1):**

* Distance to (1, 1) =1
* Distance to (1, 0): ​
* Distance to (0, 1): 0
* Distance to (0, 0): 1

The 3 nearest neighbors are: (0, 1), (1, 1), and (0, 0). Their corresponding labels are y=1,y=0,y=0. So, the majority class is 0.

**2nd Test Point (x1=1,x2=0):**

* Distance to (1, 1): 1
* Distance to (1, 0): 0
* Distance to (0, 1):
* Distance to (0, 0): 1

the majority class is 0.

**3rd Test Point (x1=1,x2=1):**

* Distance to (1, 1): 0
* Distance to (1, 0): 1
* Distance to (0, 1): 1
* Distance to (0, 0):

The 3 nearest neighbors are: (1, 1), (1, 0), and (0, 1). Their corresponding labels are y=0,y=1,y=1. So, the majority class is 1.  
**Prediction by 3-NN: 1**

**Step 4: Meta-Learner's Final Predictions**

The meta-learner uses the predictions from both the decision tree and 3-NN. The logic is:

* If 3-NN predicts 1, the meta-learner predicts y=1
* If 3-NN predicts 0, it checks the decision tree's prediction.

**So, for each test point:**

* **For x1=0,x2=1:**
  + 3-NN predicts 0, Decision Tree predicts 0 → Meta-Learner predicts y=0
* **For x1=1,x2=0:**
  + 3-NN predicts 0, Decision Tree predicts 1 → Meta-Learner predicts y=1
* **For x1=1,x2=1**:
  + 3-NN predicts 1 → Meta-Learner predicts y=1.

**Step 5: Calculate Accuracy**

* Correct predictions: 2 (for the first and second test point)
* Total test points: 3

**Accuracy**: 66.67%

The model accuracy on this test data is **66.67%**.