# Ensemble Techniques

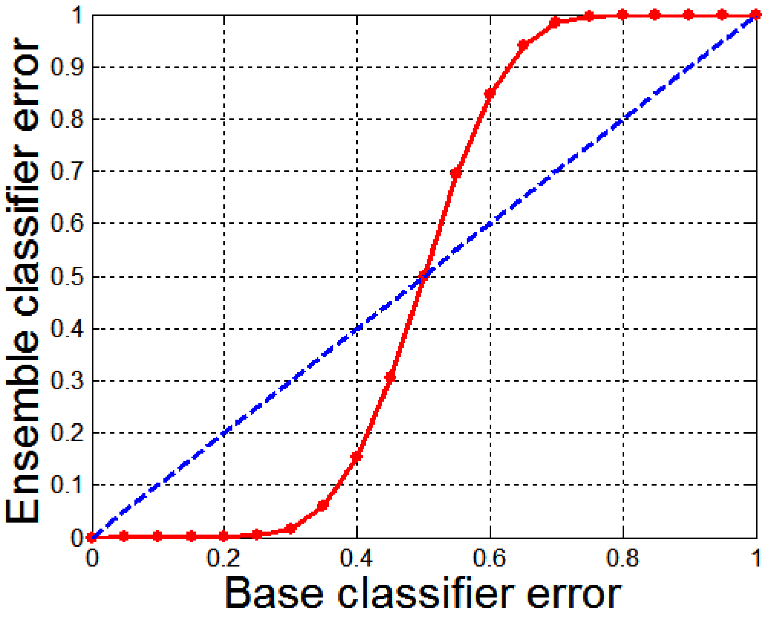
# Ensemble Methods

- Construct a set of classifiers from the training data

- Predict class label of test records by combining the predictions made by multiple classifiers

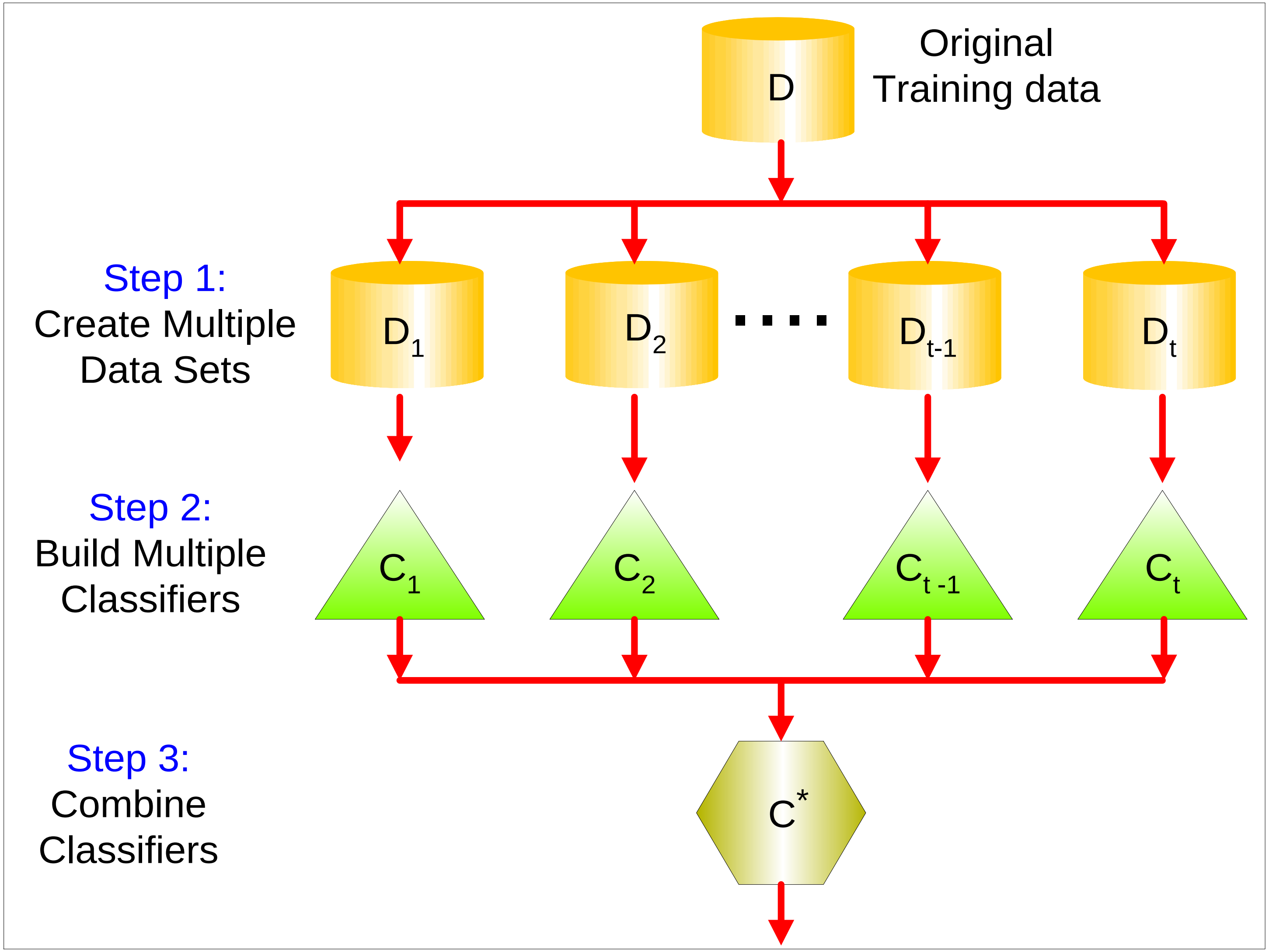
# Why Ensemble Methods work?

- Suppose there are 25 base classifiers
  - Each classifier has error rate, $\varepsilon = 0.35$
  - Assume errors made by classifiers are uncorrelated
  - Probability that the ensemble classifier makes a wrong prediction:




$$P(X \geq 13) = \sum_{i=13}^{25} \binom{25}{i} \varepsilon^i (1-\varepsilon)^{25-i} = 0.06$$

# General Approach



# Types of Ensemble Methods

- Manipulate data distribution
  - Example: bagging, boosting
- Manipulate input features
  - Example: random forests
- Manipulate class labels
  - Example: error-correcting output coding

# Bagging

- Sampling with replacement

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| **Original Data** | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| **Bagging (Round 1)** | 7 | 8 | 10 | 8 | 2 | 5 | 10 | 10 | 5 | 9 |
| **Bagging (Round 2)** | 1 | 4 | 9 | 1 | 2 | 3 | 2 | 7 | 3 | 2 |
| **Bagging (Round 3)** | 1 | 8 | 5 | 10 | 5 | 5 | 9 | 6 | 3 | 7 |

- Build classifier on each bootstrap sample

- Each data instance has probability $1- (1 – 1/n)^n$ of being selected as pat of the bootstrap sample

# Bagging Algorithm

**Algorithm 5.6** Bagging Algorithm

Let $k$ be the number of bootstrap samples.
**for** $i = 1$ **to** $k$ **do**
    Create a bootstrap sample of size $n$, $D_i$.
    Train a base classifier $C_i$ on the bootstrap sample $D_i$.
**end for**
$C^*(x) = \arg\max_y \sum_i \delta\big(C_i(x) = y\big)$, $\{\delta(\cdot) = 1$ if its argument is true, and 0 otherwise.$\}$

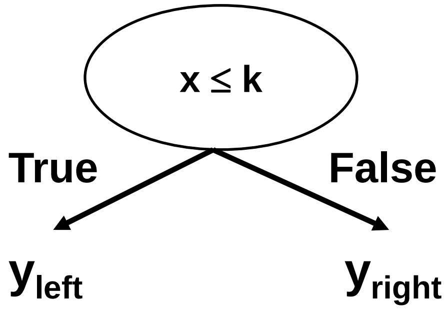
# Bagging Example

- Consider 1-dimensional data set:

**Original Data:**

| x | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 | 0.6 | 0.7 | 0.8 | 0.9 | 1 |
|---|---|---|---|---|---|---|---|---|---|---|
| y | 1 | 1 | 1 | -1 | -1 | -1 | -1 | 1 | 1 | 1 |

- Classifier is a decision stump
  - Decision rule: $x \le k$ versus $x > k$
  - Split point k is chosen based on entropy



# Bagging Example

Bagging Round 1:

| x | 0.1 | 0.2 | 0.2 | 0.3 | 0.4 | 0.4 | 0.5 | 0.6 | 0.9 | 0.9 |
|---|---|---|---|---|---|---|---|---|---|---|
| y | 1 | 1 | 1 | 1 | -1 | -1 | -1 | -1 | 1 | 1 |

**x <= 0.35 ➔ y = 1**
**x > 0.35 ➔ y = -1**

# Bagging Example

Bagging Round 1:

| x | 0.1 | 0.2 | 0.2 | 0.3 | 0.4 | 0.4 | 0.5 | 0.6 | 0.9 | 0.9 |
|---|---|---|---|---|---|---|---|---|---|---|
| y | 1 | 1 | 1 | 1 | -1 | -1 | -1 | -1 | 1 | 1 |

x <= 0.35 ➔ y = 1
x > 0.35 ➔ y = -1

Bagging Round 2:

| x | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 | 0.5 | 0.9 | 1 | 1 | 1 |
|---|---|---|---|---|---|---|---|---|---|---|
| y | 1 | 1 | 1 | -1 | -1 | -1 | 1 | 1 | 1 | 1 |

x <= 0.7 ➔ y = 1
x > 0.7 ➔ y = 1

Bagging Round 3:

| x | 0.1 | 0.2 | 0.3 | 0.4 | 0.4 | 0.5 | 0.7 | 0.7 | 0.8 | 0.9 |
|---|---|---|---|---|---|---|---|---|---|---|
| y | 1 | 1 | 1 | -1 | -1 | -1 | -1 | -1 | 1 | 1 |

x <= 0.35 ➔ y = 1
x > 0.35 ➔ y = -1

Bagging Round 4:

| x | 0.1 | 0.1 | 0.2 | 0.4 | 0.4 | 0.5 | 0.5 | 0.7 | 0.8 | 0.9 |
|---|---|---|---|---|---|---|---|---|---|---|
| y | 1 | 1 | 1 | -1 | -1 | -1 | -1 | -1 | 1 | 1 |

x <= 0.3 ➔ y = 1
x > 0.3 ➔ y = -1

Bagging Round 5:

| x | 0.1 | 0.1 | 0.2 | 0.5 | 0.6 | 0.6 | 0.6 | 1 | 1 | 1 |
|---|---|---|---|---|---|---|---|---|---|---|
| y | 1 | 1 | 1 | -1 | -1 | -1 | -1 | 1 | 1 | 1 |

x <= 0.35 ➔ y = 1
x > 0.35 ➔ y = -1

# Bagging Example

Bagging Round 6:

| x | 0.2 | 0.4 | 0.5 | 0.6 | 0.7 | 0.7 | 0.7 | 0.8 | 0.9 | 1 |
|---|---|---|---|---|---|---|---|---|---|---|
| y | 1 | -1 | -1 | -1 | -1 | -1 | -1 | 1 | 1 | 1 |

x <= 0.75 → y = -1
x > 0.75 → y = 1

Bagging Round 7:

| x | 0.1 | 0.4 | 0.4 | 0.6 | 0.7 | 0.8 | 0.9 | 0.9 | 0.9 | 1 |
|---|---|---|---|---|---|---|---|---|---|---|
| y | 1 | -1 | -1 | -1 | -1 | 1 | 1 | 1 | 1 | 1 |

x <= 0.75 → y = -1
x > 0.75 → y = 1

Bagging Round 8:

| x | 0.1 | 0.2 | 0.5 | 0.5 | 0.5 | 0.7 | 0.7 | 0.8 | 0.9 | 1 |
|---|---|---|---|---|---|---|---|---|---|---|
| y | 1 | 1 | -1 | -1 | -1 | -1 | -1 | 1 | 1 | 1 |

x <= 0.75 → y = -1
x > 0.75 → y = 1

Bagging Round 9:

| x | 0.1 | 0.3 | 0.4 | 0.4 | 0.6 | 0.7 | 0.7 | 0.8 | 1 | 1 |
|---|---|---|---|---|---|---|---|---|---|---|
| y | 1 | 1 | -1 | -1 | -1 | -1 | -1 | 1 | 1 | 1 |

x <= 0.75 → y = -1
x > 0.75 → y = 1

Bagging Round 10:

| x | 0.1 | 0.1 | 0.1 | 0.1 | 0.3 | 0.3 | 0.8 | 0.8 | 0.9 | 0.9 |
|---|---|---|---|---|---|---|---|---|---|---|
| y | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

x <= 0.05 → y = 1
x > 0.05 → y = 1

# Bagging Example

- Summary of Training sets:

| Round | Split Point | Left Class | Right Class |
|---|---|---|---|
| 1 | 0.35 | 1 | -1 |
| 2 | 0.7 | 1 | 1 |
| 3 | 0.35 | 1 | -1 |
| 4 | 0.3 | 1 | -1 |
| 5 | 0.35 | 1 | -1 |
| 6 | 0.75 | -1 | 1 |
| 7 | 0.75 | -1 | 1 |
| 8 | 0.75 | -1 | 1 |
| 9 | 0.75 | -1 | 1 |
| 10 | 0.05 | 1 | 1 |

# Bagging Example

- Assume test set is the same as the original data
- Use majority vote to determine class of ensemble classifier

| Round | x=0.1 | x=0.2 | x=0.3 | x=0.4 | x=0.5 | x=0.6 | x=0.7 | x=0.8 | x=0.9 | x=1.0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 |
| 2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 3 | 1 | 1 | 1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 |
| 4 | 1 | 1 | 1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 |
| 5 | 1 | 1 | 1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 |
| 6 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | 1 | 1 | 1 |
| 7 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | 1 | 1 | 1 |
| 8 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | 1 | 1 | 1 |
| 9 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | 1 | 1 | 1 |
| 10 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Sum | 2 | 2 | 2 | -6 | -6 | -6 | -6 | 2 | 2 | 2 |
| Sign | 1 | 1 | 1 | -1 | -1 | -1 | -1 | 1 | 1 | 1 |

**Predicted Class**

# Boosting

- An iterative procedure to adaptively change distribution of training data by focusing more on previously misclassified records
  - Initially, all N records are assigned equal weights
  - Unlike bagging, weights may change at the end of each boosting round

# Boosting

- Records that are wrongly classified will have their weights increased
- Records that are classified correctly will have their weights decreased

| Original Data | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| Boosting (Round 1) | 7 | 3 | 2 | 8 | 7 | 9 | 4 | 10 | 6 | 3 |
| Boosting (Round 2) | 5 | 4 | 9 | 4 | 2 | 5 | 1 | 7 | 4 | 2 |
| Boosting (Round 3) | 4 | 4 | 8 | 10 | 4 | 5 | 4 | 6 | 3 | 4 |

- Example 4 is hard to classify
- Its weight is increased, therefore it is more likely to be chosen again in subsequent rounds

# AdaBoost

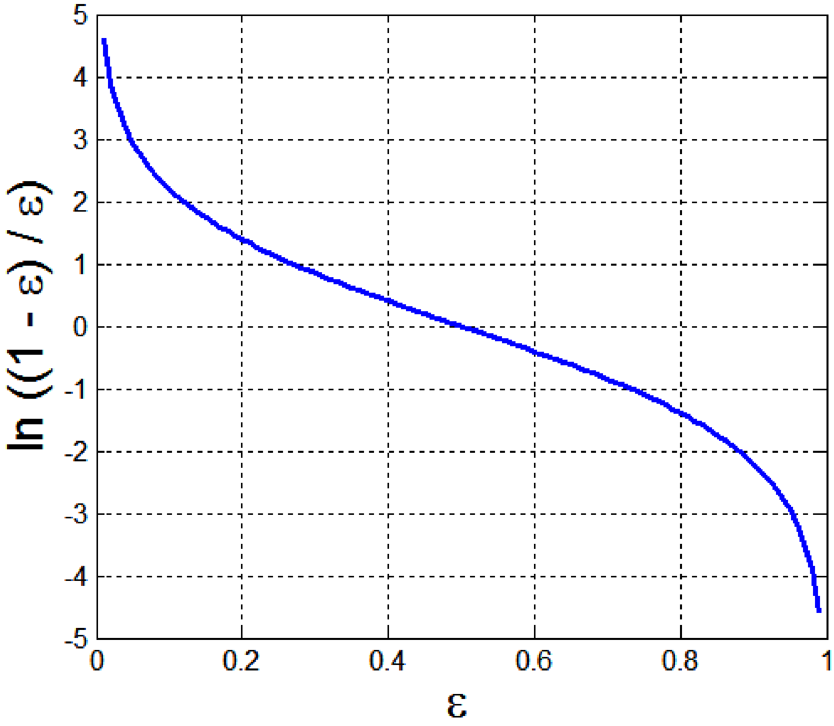
- Base classifiers: $C_1, C_2, \ldots, C_T$

- Error rate:

$$\varepsilon_i = \frac{1}{N} \sum_{j=1}^{N} w_j \delta\left(C_i(x_j) \neq y_j\right)$$

- Importance of a classifier:

$$\alpha_i = \frac{1}{2} \ln\left(\frac{1-\varepsilon_i}{\varepsilon_i}\right)$$



# AdaBoost Algorithm

- Weight update:

$$w_j^{(i+1)} = \frac{w_j^{(i)}}{Z_i} \begin{cases} \exp^{-\alpha_i} & \text{if } C_i(x_j) = y_j \\ \exp^{\alpha_i} & \text{if } C_i(x_j) \neq y_j \end{cases}$$

where $Z_i$ is the normalization factor

- If any intermediate rounds produce error rate higher than 50%, the weights are reverted back to 1/n and the resampling procedure is repeated
- Classification:

$$C^*(x) = \underset{y}{\operatorname{argmax}} \sum_{i=1}^{T} \alpha_i \delta(C_i(x) = y)$$

# AdaBoost Algorithm

**Algorithm 5.7** AdaBoost Algorithm

```
1: w = {w_j = 1/n | j = 1, 2, ..., n}.    {Initialize the weights for all n instances.}
2: Let k be the number of boosting rounds.
3: for i = 1 to k do
4:    Create training set D_i by sampling (with replacement) from D according to w.
5:    Train a base classifier C_i on D_i.
6:    Apply C_i to all instances in the original training set, D.
7:    ε_i = (1/n) [ Σ_j w_j δ(C_i(x_j) ≠ y_j) ]    {Calculate the weighted error}
8:    if ε_i > 0.5 then
9:       w = {w_j = 1/n | j = 1, 2, ..., n}.    {Reset the weights for all n instances.}
10:      Go back to Step 4.
11:   end if
12:   α_i = (1/2) ln ((1 − ε_i)/ε_i).
13:   Update the weight of each instance according to equation (5.88).
14: end for
15: C*(x) = argmax_y Σ_{j=1}^{T} α_j δ(C_j(x) = y)).
```

Where:

- Line 1: $\mathrm{w} = \{w_j = 1/n \mid j = 1, 2, \cdots, n\}$
- Line 7: $\epsilon_i = \frac{1}{n}\left[\sum_j w_j\, \delta\big(C_i(x_j) \neq y_j\big)\right]$
- Line 12: $\alpha_i = \frac{1}{2} \ln \frac{1-\epsilon_i}{\epsilon_i}$
- Line 15: $C^*(\mathrm{x}) = \arg\max_y \sum_{j=1}^{T} \alpha_j \delta(C_j(\mathrm{x}) = y))$

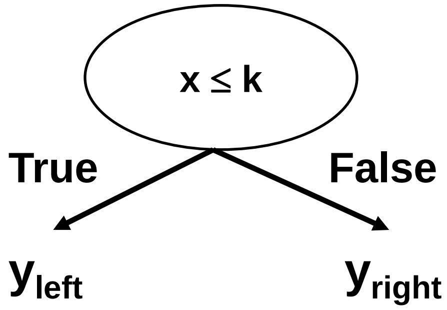
# AdaBoost Example

- Consider 1-dimensional data set:

**Original Data:**

| x | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 | 0.6 | 0.7 | 0.8 | 0.9 | 1 |
|---|---|---|---|---|---|---|---|---|---|---|
| y | 1 | 1 | 1 | -1 | -1 | -1 | -1 | 1 | 1 | 1 |

- Classifier is a decision stump
  – Decision rule: $x \leq k$ versus $x > k$
  – Split point k is chosen based on entropy



# AdaBoost Example

- Training sets for the first 3 boosting rounds:

Boosting Round 1:

| x | 0.1 | 0.4 | 0.5 | 0.6 | 0.6 | 0.7 | 0.7 | 0.7 | 0.8 | 1 |
|---|---|---|---|---|---|---|---|---|---|---|
| y | 1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | 1 | 1 |

Boosting Round 2:

| x | 0.1 | 0.1 | 0.2 | 0.2 | 0.2 | 0.2 | 0.3 | 0.3 | 0.3 | 0.3 |
|---|---|---|---|---|---|---|---|---|---|---|
| y | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

Boosting Round 3:

| x | 0.2 | 0.2 | 0.4 | 0.4 | 0.4 | 0.4 | 0.5 | 0.6 | 0.6 | 0.7 |
|---|---|---|---|---|---|---|---|---|---|---|
| y | 1 | 1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 |

- Summary:

| Round | Split Point | Left Class | Right Class | alpha |
|---|---|---|---|---|
| 1 | 0.75 | -1 | 1 | 1.738 |
| 2 | 0.05 | 1 | 1 | 2.7784 |
| 3 | 0.3 | 1 | -1 | 4.1195 |

# AdaBoost Example

- Weights

| Round | x=0.1 | x=0.2 | x=0.3 | x=0.4 | x=0.5 | x=0.6 | x=0.7 | x=0.8 | x=0.9 | x=1.0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 |
| 2 | 0.311 | 0.311 | 0.311 | 0.01 | 0.01 | 0.01 | 0.01 | 0.01 | 0.01 | 0.01 |
| 3 | 0.029 | 0.029 | 0.029 | 0.228 | 0.228 | 0.228 | 0.228 | 0.009 | 0.009 | 0.009 |

- Classification

| Round | x=0.1 | x=0.2 | x=0.3 | x=0.4 | x=0.5 | x=0.6 | x=0.7 | x=0.8 | x=0.9 | x=1.0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | 1 | 1 | 1 |
| 2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 3 | 1 | 1 | 1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 |
| Sum | 5.16 | 5.16 | 5.16 | -3.08 | -3.08 | -3.08 | -3.08 | 0.397 | 0.397 | 0.397 |
| Sign | 1 | 1 | 1 | -1 | -1 | -1 | -1 | 1 | 1 | 1 |

**Predicted Class**