



UNIVERSITY<sup>AT</sup>ALBANY  
STATE UNIVERSITY OF NEW YORK

# CSI 436/536 (Spring 2025)

# **Machine Learning**

## Lecture 14: Ensemble Learning

Chong Liu

Department of Computer Science

Apr 2, 2025

# Announcement

- Homework 3 has been released.
  - Due in two weeks.

# Recap: Risk Decomposition

$$\begin{aligned} & \mathbb{E}[R(\hat{h})] - R(h_{\text{Bayes}}) \\ \leq & \underbrace{\mathbb{E}[\hat{R}(\hat{h}) - \hat{R}(h_{\text{ERM}})]}_{\text{Optimization Error}} + \underbrace{R(h^*) - R(h_{\text{Bayes}})}_{\text{Approximation Error}} + \underbrace{\mathbb{E}[R(\hat{h}) - \hat{R}(\hat{h})]}_{\text{Generalization Error}} \end{aligned}$$

**Optimization Error**

How close am I from minimizing the empirical risk?

**Approximation Error**

How much worse the best “representable” classifier is from the best classifier out there.

**Generalization Error**

How different the empirical risk of my classifier is from its population risk?

Make sure you understand what each kind of error means!

# Machine learning can be viewed as a collection of techniques in minimizing the three types of errors

|                                                   | Optimization error                                                                                                                                        | Generalization Error                                                                                                                                                                        | Approximation Error                                                                                                                                                                                          |
|---------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Definition                                        | $\hat{R}(\hat{h}) - \hat{R}(h_{\text{ERM}})$                                                                                                              | $R(\hat{h}) - \hat{R}(\hat{h})$                                                                                                                                                             | $R(h^*) - R(h_{\text{Bayes}})$                                                                                                                                                                               |
| Challenges                                        | <ul style="list-style-type: none"><li>• Finding ERM for some loss functions is NP-Hard.</li><li>• Efficiency isn't enough. Need to be scalable.</li></ul> | <ul style="list-style-type: none"><li>• We do not observe Risk!</li><li>• Don't have infinite data.</li><li>• Large generalization error <math>\Leftrightarrow</math> Overfitting</li></ul> | <ul style="list-style-type: none"><li>• Don't know data distribution.</li><li>• No knowledge of Bayes optimal classifier.</li><li>• Large approx. error <math>\Leftrightarrow</math> Underfitting!</li></ul> |
| What we have learned to address these challenges? |                                                                                                                                                           |                                                                                                                                                                                             |                                                                                                                                                                                                              |

# Machine learning can be viewed as a collection of techniques in minimizing the three types of errors

|                                                   | Optimization error                                                                                                                                       | Generalization Error                                                                                                                                                                      | Approximation Error                                                                                                                                                                                        |
|---------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Definition                                        | $\hat{R}(\hat{h}) - \hat{R}(h_{\text{ERM}})$                                                                                                             | $R(\hat{h}) - \hat{R}(\hat{h})$                                                                                                                                                           | $R(h^*) - R(h_{\text{Bayes}})$                                                                                                                                                                             |
| Challenges                                        | <ul style="list-style-type: none"> <li>Finding ERM for some loss functions is NP-Hard.</li> <li>Efficiency isn't enough. Need to be scalable.</li> </ul> | <ul style="list-style-type: none"> <li>We do not observe Risk!</li> <li>Don't have infinite data.</li> <li>Large generalization error <math>\Leftrightarrow</math> Overfitting</li> </ul> | <ul style="list-style-type: none"> <li>Don't know data distribution.</li> <li>No knowledge of Bayes optimal classifier.</li> <li>Large approx. error <math>\Leftrightarrow</math> Underfitting!</li> </ul> |
| What we have learned to address these challenges? | "Just-relax" Surrogate loss, Gradient Descent, SGD                                                                                                       |                                                                                                                                                                                           |                                                                                                                                                                                                            |

# Machine learning can be viewed as a collection of techniques in minimizing the three types of errors

|                                                   | Optimization error                                                                                                                                       | Generalization Error                                                                                                                                                                      | Approximation Error                                                                                                                                                                                        |
|---------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Definition                                        | $\hat{R}(\hat{h}) - \hat{R}(h_{\text{ERM}})$                                                                                                             | $R(\hat{h}) - \hat{R}(\hat{h})$                                                                                                                                                           | $R(h^*) - R(h_{\text{Bayes}})$                                                                                                                                                                             |
| Challenges                                        | <ul style="list-style-type: none"> <li>Finding ERM for some loss functions is NP-Hard.</li> <li>Efficiency isn't enough. Need to be scalable.</li> </ul> | <ul style="list-style-type: none"> <li>We do not observe Risk!</li> <li>Don't have infinite data.</li> <li>Large generalization error <math>\Leftrightarrow</math> Overfitting</li> </ul> | <ul style="list-style-type: none"> <li>Don't know data distribution.</li> <li>No knowledge of Bayes optimal classifier.</li> <li>Large approx. error <math>\Leftrightarrow</math> Underfitting!</li> </ul> |
| What we have learned to address these challenges? | "Just-relax" Surrogate loss, Gradient Descent, SGD                                                                                                       | Holdout, Cross-Validation<br>Regularization<br>Statistical learning theory<br>(not covered)                                                                                               |                                                                                                                                                                                                            |

# Machine learning can be viewed as a collection of techniques in minimizing the three types of errors

|                                                   | Optimization error                                                                                                                                        | Generalization Error                                                                                                                                                                        | Approximation Error                                                                                                                                                                                          |
|---------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Definition                                        | $\hat{R}(\hat{h}) - \hat{R}(h_{\text{ERM}})$                                                                                                              | $R(\hat{h}) - \hat{R}(\hat{h})$                                                                                                                                                             | $R(h^*) - R(h_{\text{Bayes}})$                                                                                                                                                                               |
| Challenges                                        | <ul style="list-style-type: none"><li>• Finding ERM for some loss functions is NP-Hard.</li><li>• Efficiency isn't enough. Need to be scalable.</li></ul> | <ul style="list-style-type: none"><li>• We do not observe Risk!</li><li>• Don't have infinite data.</li><li>• Large generalization error <math>\Leftrightarrow</math> Overfitting</li></ul> | <ul style="list-style-type: none"><li>• Don't know data distribution.</li><li>• No knowledge of Bayes optimal classifier.</li><li>• Large approx. error <math>\Leftrightarrow</math> Underfitting!</li></ul> |
| What we have learned to address these challenges? | "Just-relax" Surrogate loss, Gradient Descent, SGD                                                                                                        | Holdout, Cross-Validation<br>Regularization<br>Statistical learning theory<br>(not covered)                                                                                                 | Better features<br>More flexible decision boundaries<br>Better probabilistic models                                                                                                                          |

# Machine learning can be viewed as a collection of techniques in minimizing the three types of errors

|                                                   | Optimization error                                                                                                                                       | Generalization Error                                                                                                                                                                      | Approximation Error                                                                                                                                                                                        |
|---------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Definition                                        | $\hat{R}(\hat{h}) - \hat{R}(h_{\text{ERM}})$                                                                                                             | $R(\hat{h}) - \hat{R}(\hat{h})$                                                                                                                                                           | $R(h^*) - R(h_{\text{Bayes}})$                                                                                                                                                                             |
| Challenges                                        | <ul style="list-style-type: none"> <li>Finding ERM for some loss functions is NP-Hard.</li> <li>Efficiency isn't enough. Need to be scalable.</li> </ul> | <ul style="list-style-type: none"> <li>We do not observe Risk!</li> <li>Don't have infinite data.</li> <li>Large generalization error <math>\Leftrightarrow</math> Overfitting</li> </ul> | <ul style="list-style-type: none"> <li>Don't know data distribution.</li> <li>No knowledge of Bayes optimal classifier.</li> <li>Large approx. error <math>\Leftrightarrow</math> Underfitting!</li> </ul> |
| What we have learned to address these challenges? | "Just-relax" Surrogate loss,<br>Gradient Descent, SGD                                                                                                    | Holdout, Cross-Validation<br>Regularization<br>Statistical learning theory<br><i>(not covered)</i>                                                                                        | Better features<br>More flexible decision boundaries<br>Better probabilistic models<br><br><b>But how to minimize approx. error automatically?</b>                                                         |



# Machine learning can be viewed as a collection of techniques in minimizing the three types of errors

|                                                   | Optimization error                                                                                                                                       | Generalization Error                                                                                                                                                                      | Approximation Error                                                                                                                                                                                        |
|---------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Definition                                        | $\hat{R}(\hat{h}) - \hat{R}(h_{\text{ERM}})$                                                                                                             | $R(\hat{h}) - \hat{R}(\hat{h})$                                                                                                                                                           | $R(h^*) - R(h_{\text{Bayes}})$                                                                                                                                                                             |
| Challenges                                        | <ul style="list-style-type: none"> <li>Finding ERM for some loss functions is NP-Hard.</li> <li>Efficiency isn't enough. Need to be scalable.</li> </ul> | <ul style="list-style-type: none"> <li>We do not observe Risk!</li> <li>Don't have infinite data.</li> <li>Large generalization error <math>\Leftrightarrow</math> Overfitting</li> </ul> | <ul style="list-style-type: none"> <li>Don't know data distribution.</li> <li>No knowledge of Bayes optimal classifier.</li> <li>Large approx. error <math>\Leftrightarrow</math> Underfitting!</li> </ul> |
| What we have learned to address these challenges? | "Just-relax" Surrogate loss, Gradient Descent, SGD                                                                                                       | Holdout, Cross-Validation<br>Regularization<br>Statistical learning theory<br>(not covered)                                                                                               | Better features<br>More flexible decision boundaries<br>Better probabilistic models<br><br><b>But how to minimize approx. error automatically?</b>                                                         |

Often there is a tradeoff.

More **flexible** hypothesis class  $\Rightarrow$  smaller approximation error

but larger generalization error (more overfitting) and sometimes harder optimization

# Three main approaches for expanding the hypothesis class (systematically minimizing the approx. error)

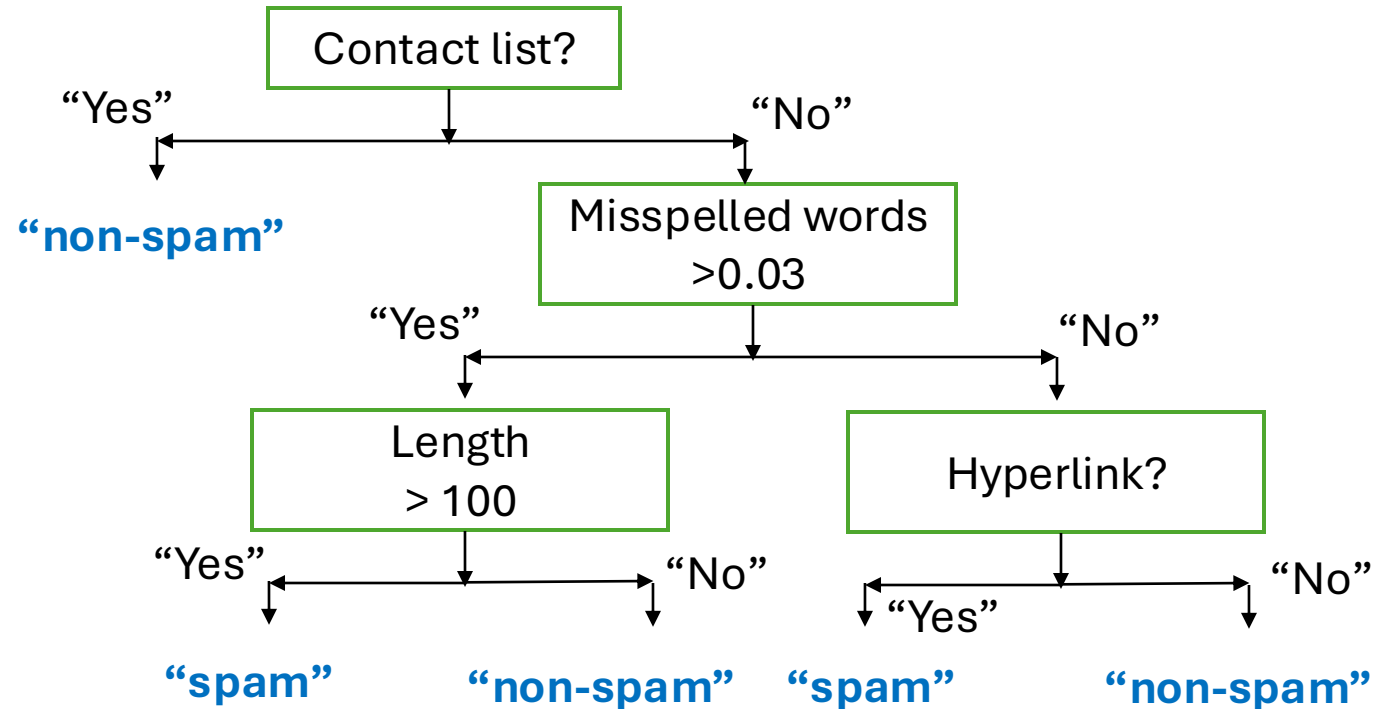
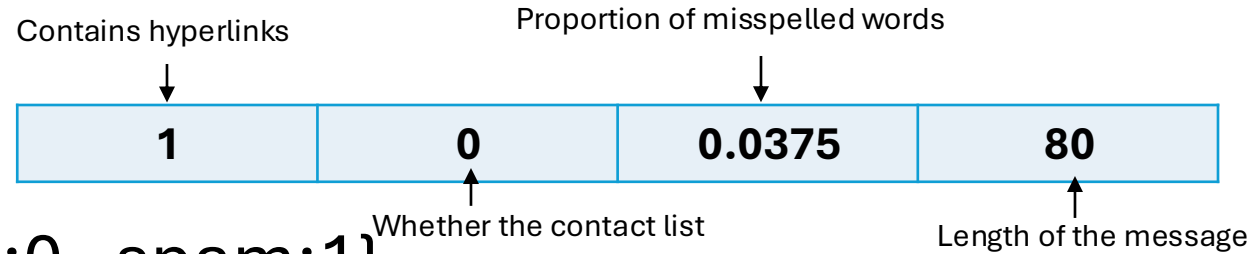
- Boosting and Bagging (Ensemble learning)
  - Combine many weak learners (e.g., decision trees with depth 3) into a strong learner
- Kernel methods (lift features to higher-dimensional space)
  - e.g., adding polynomial expansion, add interaction terms
  - Other nonlinear transformation of the original features
- Deep Learning
  - Train large neural networks using SGD
  - Learn feature representation and classification jointly.

# Today

- Review decision trees
- Approaches that minimize “approximation error”
  - Bagging and Random Forest
  - Boosting

# Recall: Decision Tree Classifiers

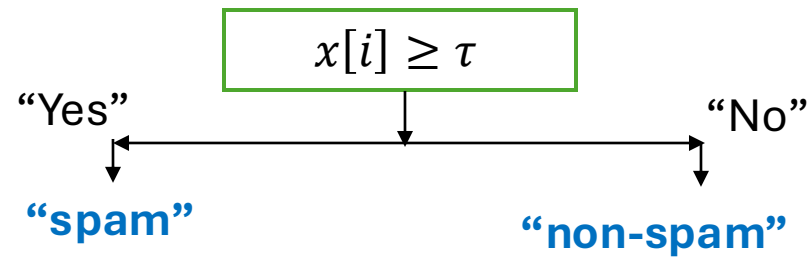
- Input features:
- Labels: {non-spam:0, spam:1}



# Let's consider a **weak learner** --- e.g., a “Decision Stump” classifier.

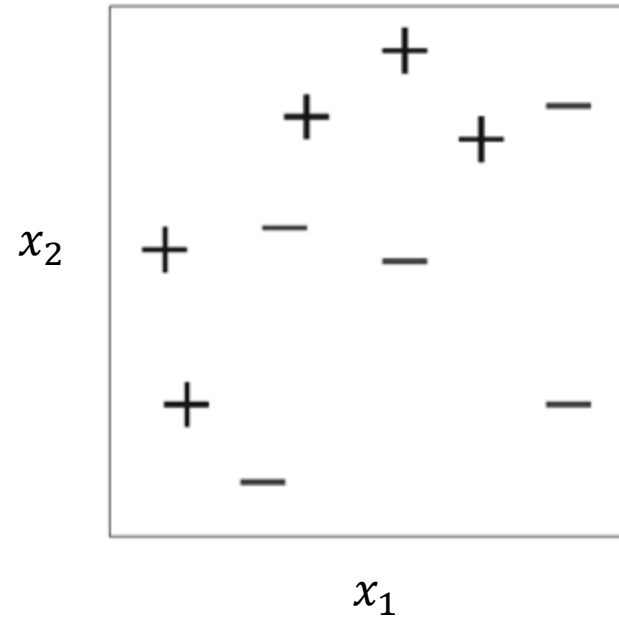
- A decision stump classifiers work as follows
  - Take exactly **one** of the features.
  - Then threshold it.

- Example:



- Parameters of the “**decision stump**” classifier
  - $i$  ----- Which feature / coordinate to use?
  - $\tau$  ----- Which threshold
  - Leaf labels ----- Assign “spam” to “Yes” or “No”

# In-class exercise: Let's work out the decision boundary of a decision stump classifier



- What is the optimal decision stump if  $x_1$  is used? What's the error rate?
- What is the optimal decision stump if  $x_2$  is used? What's the error rate?

# Ensemble classifiers

- Take multiple learner  $h_1, h_2, h_3, \dots, h_N$

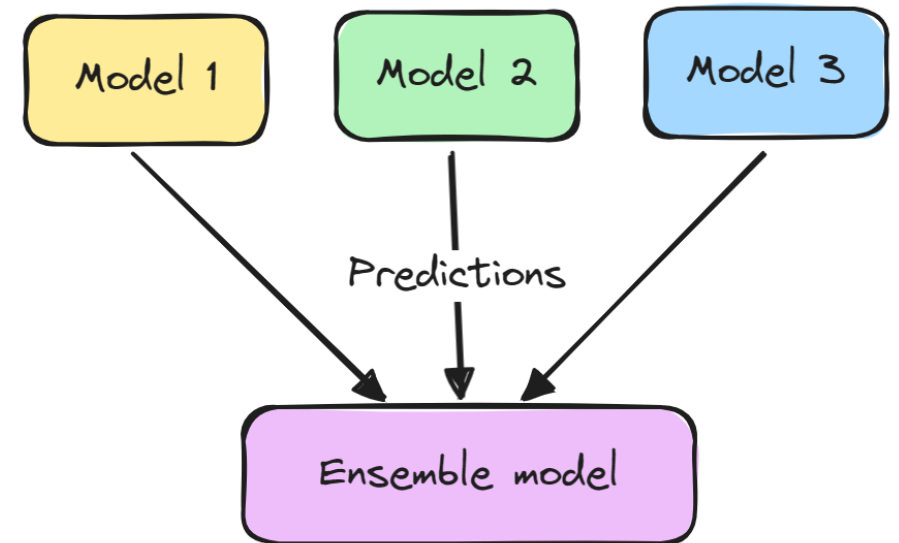
- Ensemble classifier:

- For classification problems (output space is discrete)

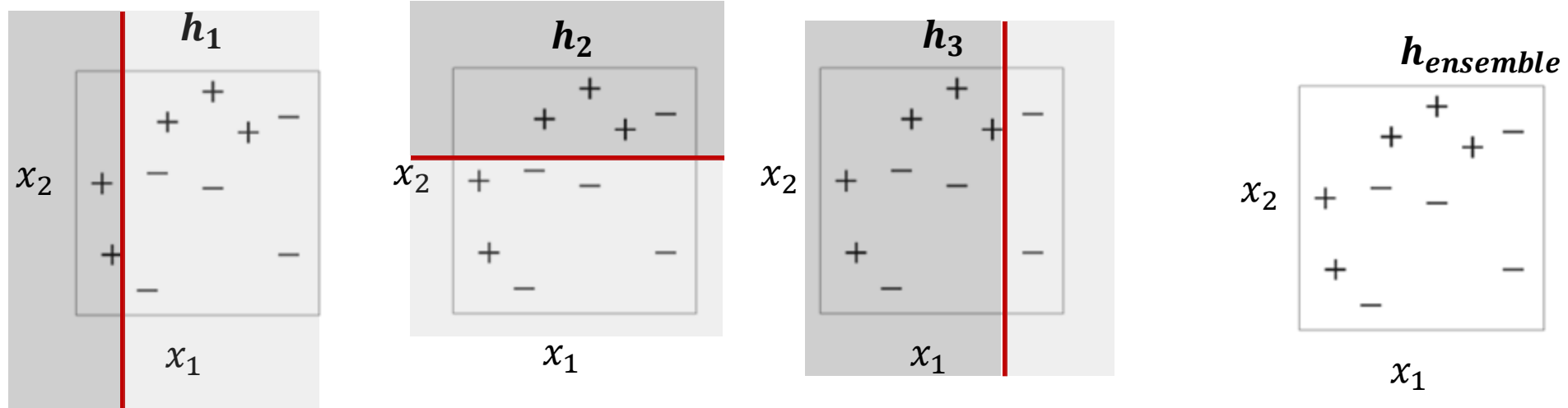
$$\arg \max_{y \in \mathcal{Y}} \frac{1}{N} \sum_{i=1}^N \alpha_i \mathbf{1}(h_i(x) = y)$$

- For regression problems (output space is continuous)

$$\frac{1}{N} \sum_{i=1}^N \alpha_i h_i(x)$$



# In-class exercise: Let's work out an example!

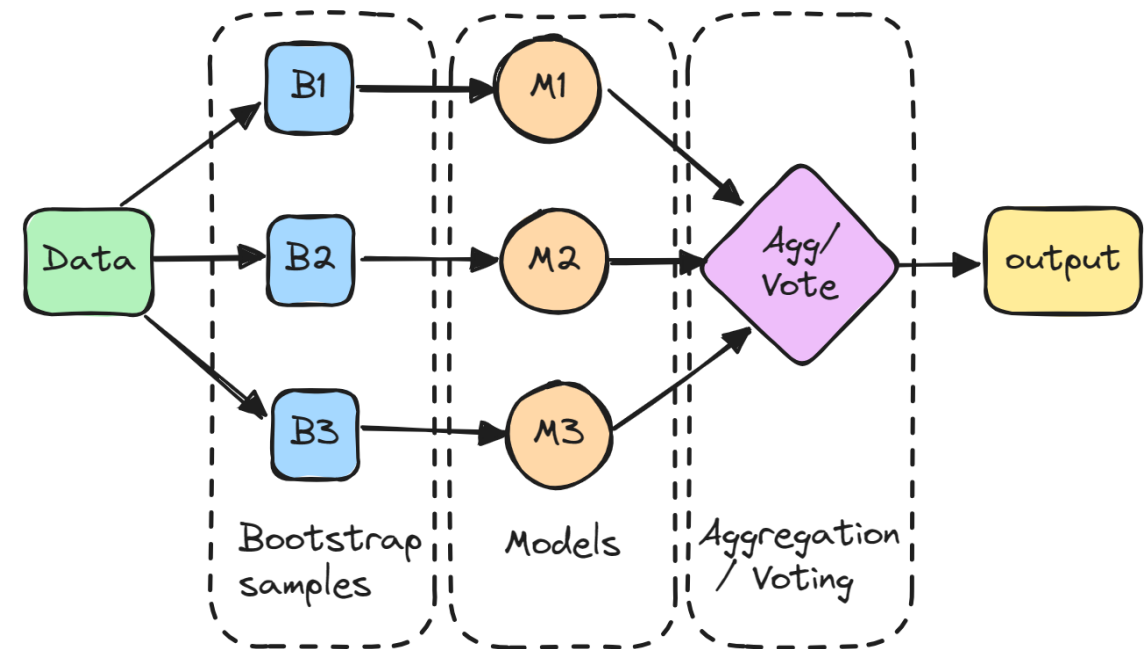


1. Circle mis-classified examples.
2. Draw the decision boundary of the ensemble classifier.

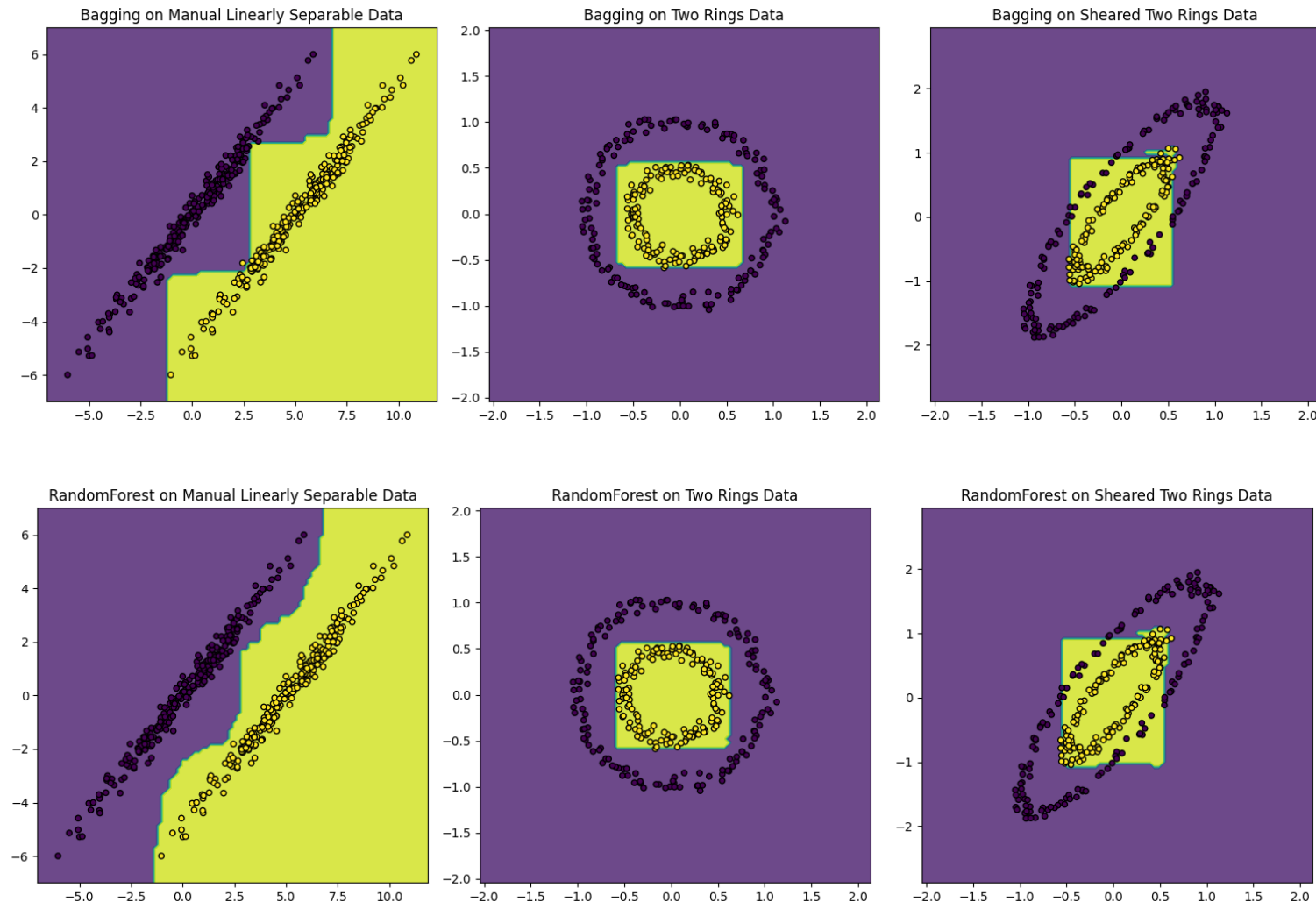


# Bagging and Random Forest Classifier are both (unweighted) ensemble classifiers (Breiman, 2001)

- Bagging:
  - Randomly sample subset of data
  - For each sample, fit a learner
  - Return an ensemble classifier
- Random Forest:
  - Randomly sample subset of data **and subset of features**
  - For each sample, fit a learner (typically decision trees)
  - Return an ensemble classifier



# Example: Decision boundary learned by Bagging and RandomForest with DecisionTrees (Depth 5, n\_trees = 1000)



# Boosting (Schapire'89)

- “*The Strength of Weak Learners*” --- can we construct a strong learner (e.g., accuracy 99%) using an ensemble of weak learners (accuracy 51%)? Answer is surprisingly positive!

For each iteration  $t$

1. Come up with a set of weights  $D_1, \dots, D_n$  for each training example (based on how incorrectly it was classified by the current classifier  $h_{ensemble}$ )
2. Fit a weak learner  $h_t = \operatorname{argmin}_h \frac{1}{n} \sum_i D_i \mathbf{1}(h(x_i) \neq y_i)$
3. Figure out a learning rate  $\alpha_t$
4. Update  $h_{ensemble}(x) = \operatorname{sign}(\sum_{j=1}^t \alpha_j h_j(x))$

**Theoretically interesting:** equivalence between weak and strong learnability.  
**Practically a very powerful algorithm** --- almost the best you can get.

# How to come up with the weights?

- AdaBoost (Freund & Schapire'95)

**Input:** Training set  $D = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_m, y_m)\}$ ;  
Base learner  $\mathcal{L}$ ;  
Number of training rounds  $T$ .

**Process:**

1:  $\mathcal{D}_1(\mathbf{x}) = 1/m$ ; Initially all data points have the same weight

2: **for**  $t = 1, 2, \dots, T$  **do**

3:  $h_t = \mathcal{L}(D, \mathcal{D}_t)$ ; Training on data points

4:  $\epsilon_t = P_{\mathbf{x} \sim \mathcal{D}_t}(h_t(\mathbf{x}) \neq f(\mathbf{x}))$ ; Calculate the error

5: **if**  $\epsilon_t > 0.5$  **then break** If binary classifier is too bad

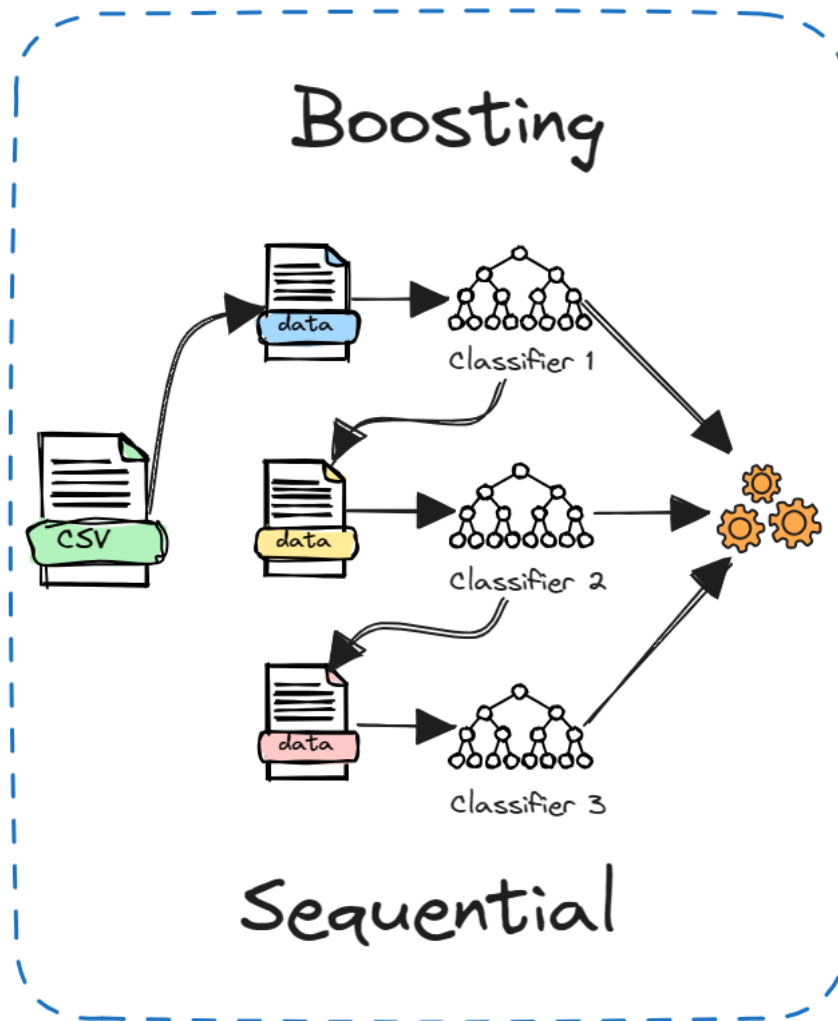
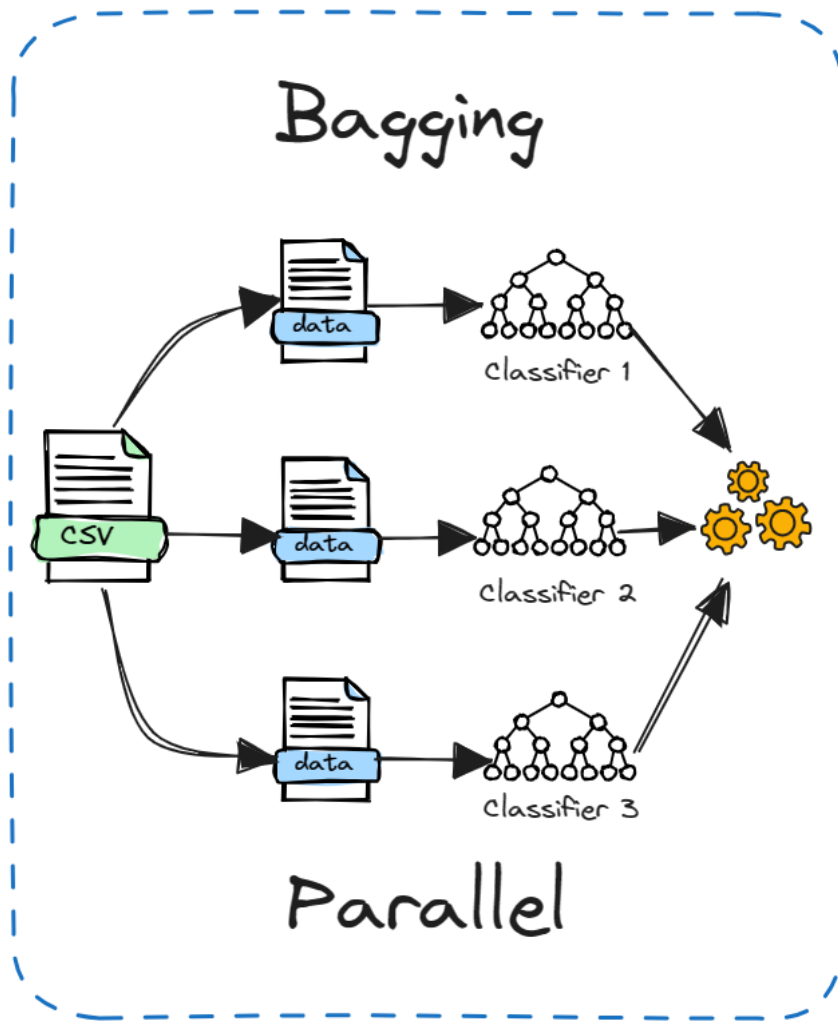
6:  $\alpha_t = \frac{1}{2} \ln \left( \frac{1-\epsilon_t}{\epsilon_t} \right)$ ; Calculate the weight parameter

7:  $\mathcal{D}_{t+1}(\mathbf{x}) = \frac{\mathcal{D}_t(\mathbf{x})}{Z_t} \times \begin{cases} \exp(-\alpha_t), & \text{if } h_t(\mathbf{x}) = f(\mathbf{x}); \\ \exp(\alpha_t), & \text{if } h_t(\mathbf{x}) \neq f(\mathbf{x}); \end{cases}$  Increase the weight of data points that are incorrectly classified; decrease - correct  
$$= \frac{\mathcal{D}_t(\mathbf{x}) \exp(-\alpha_t f(\mathbf{x}) h_t(\mathbf{x}))}{Z_t};$$

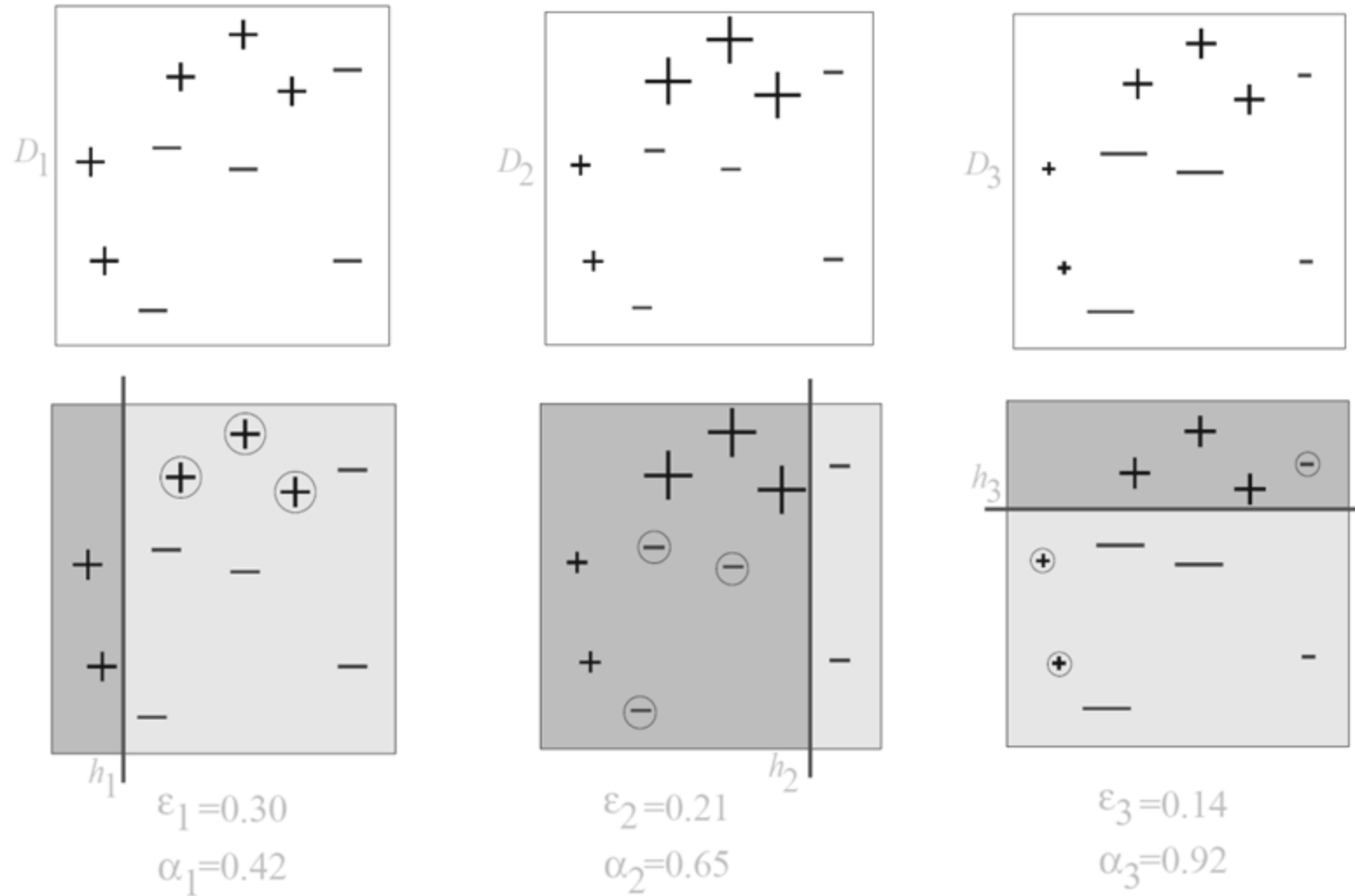
8: **end for**

**Output:**  $F(\mathbf{x}) = \text{sign} \left( \sum_{t=1}^T \alpha_t h_t(\mathbf{x}) \right)$ . Output a weighted classifier

# Bagging and boosting



# Example of AdaBoost with Decision Stumps



# Final classifier from

$$H_{\text{final}} = \text{sign} \left( 0.42 \begin{array}{|c|} \hline \text{[Diagram 1]} \\ \hline \end{array} + 0.65 \begin{array}{|c|} \hline \text{[Diagram 2]} \\ \hline \end{array} + 0.92 \begin{array}{|c|} \hline \text{[Diagram 3]} \\ \hline \end{array} \right)$$

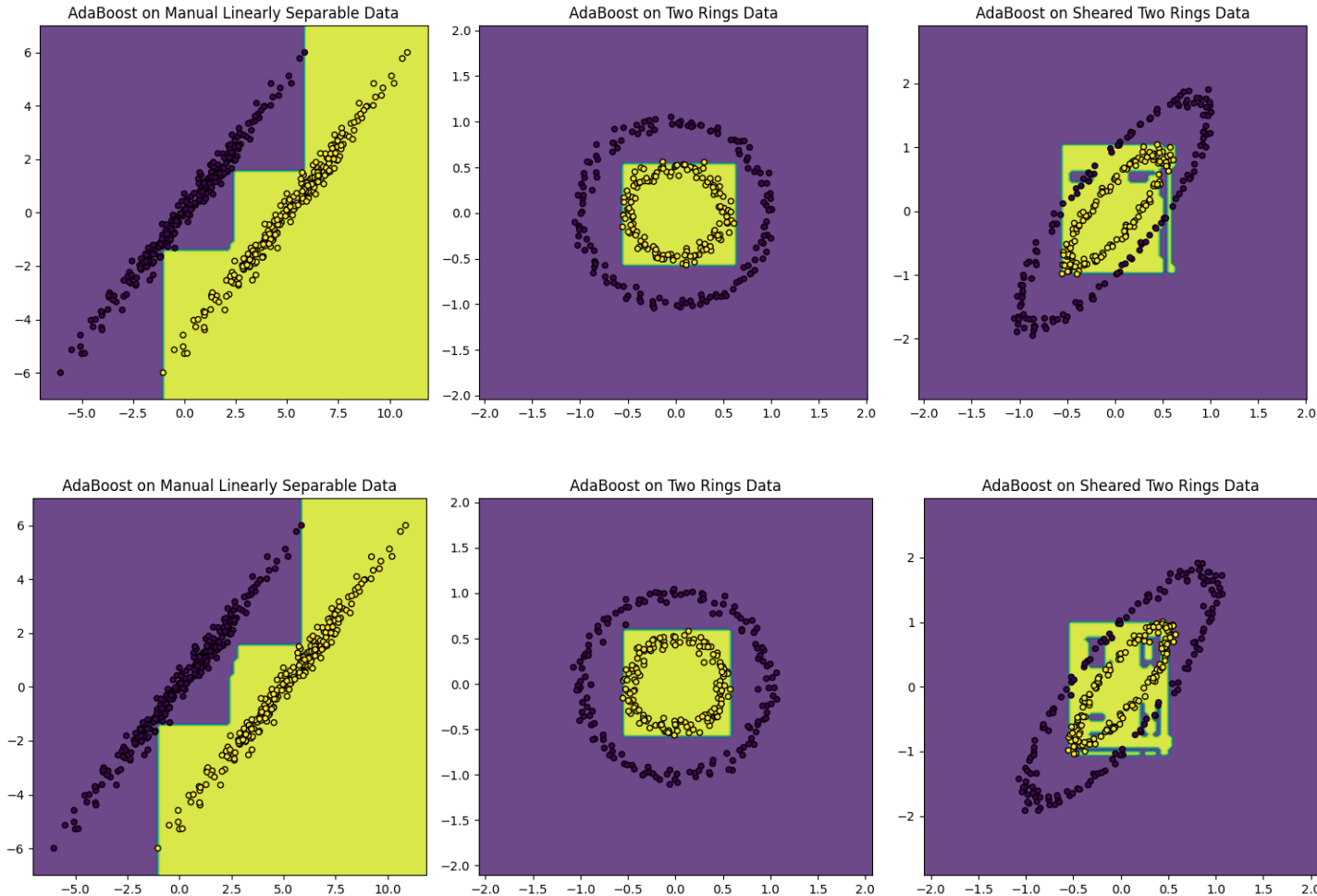
The diagrams represent 2D feature spaces with vertical decision boundaries. 
 Diagram 1: A vertical line at the left edge, with the region to the left shaded gray. 
 Diagram 2: A vertical line at the right edge, with the region to the right shaded gray. 
 Diagram 3: A horizontal line at the top edge, with the region above shaded gray.

$$= \begin{array}{|c|c|c|} \hline \text{[Diagram 4]} \\ \hline \end{array}$$

The resulting diagram is a 2D feature space with two vertical lines and one horizontal line. The regions are labeled with '+' and '-' signs. 
 The top-left region (above the horizontal line and to the left of the first vertical line) is shaded gray and contains two '+' signs. 
 The top-right region (above the horizontal line and to the right of the first vertical line) is shaded gray and contains one '+' sign. 
 The bottom-left region (below the horizontal line and to the left of the first vertical line) is shaded gray and contains two '+' signs. 
 The bottom-right region (below the horizontal line and to the right of the first vertical line) is white and contains three '-' signs.

# AdaBoost on our three examples

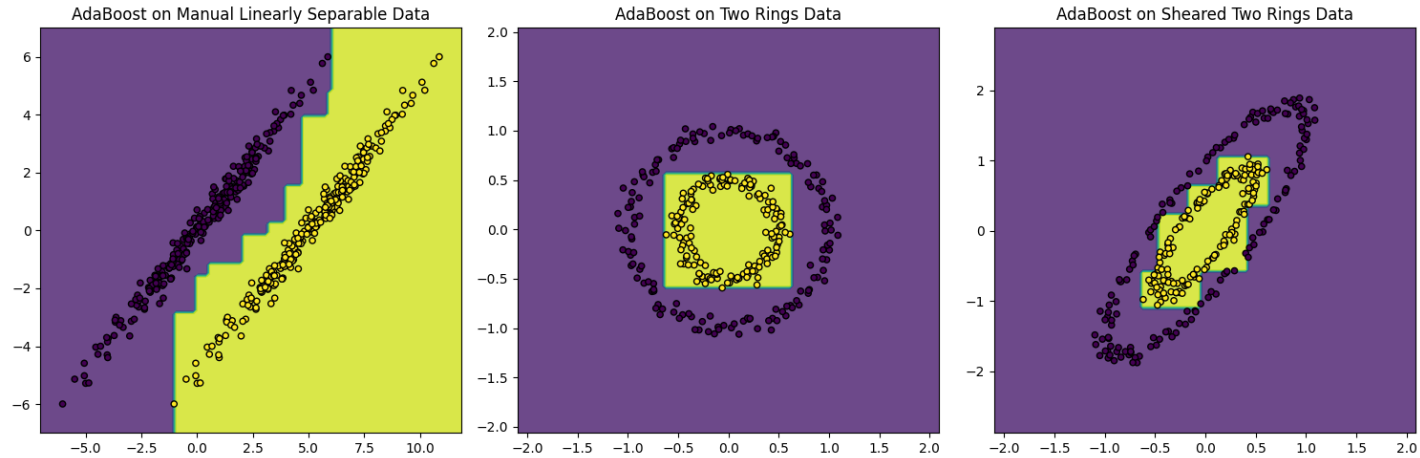
- With Decision Stumps, 100 trees



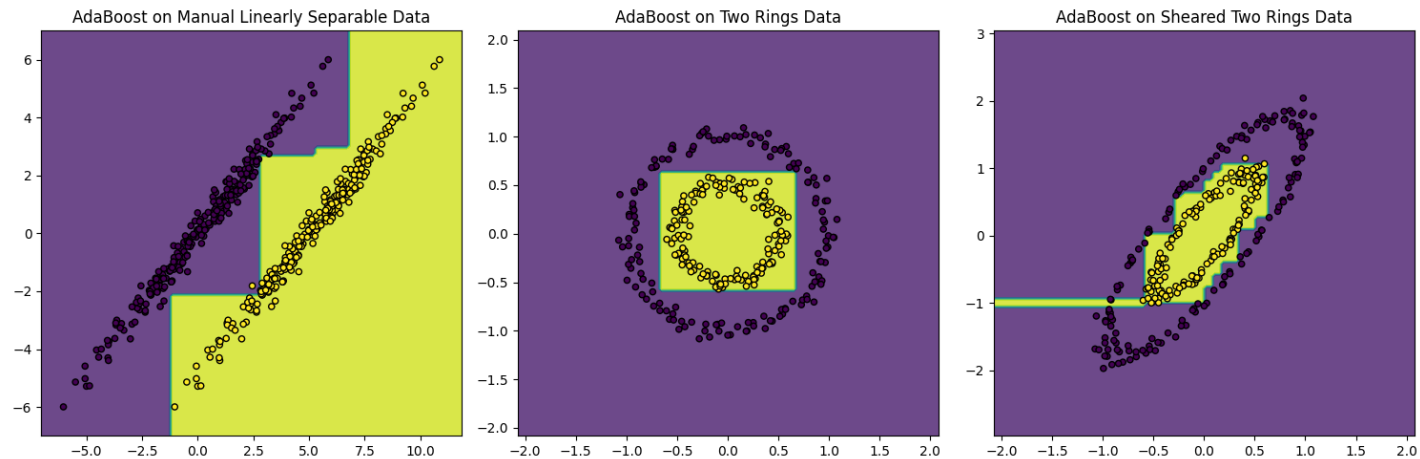


# AdaBoost on our three examples

- With Decision Trees with Depth 2, 20 trees



- With Decision Trees with Depth 5, 20 trees



# Final notes about boosting

- Boosting is a favorite among machine learners and data scientists
  - A large majority of Kaggle competitions were won using **XGBoost** --- a computationally efficient distributed implementation of a variant of AdaBoost known as Gradient Boosting.
  - XGBoost: <https://github.com/dmlc/xgboost>
- Boosting can be interpreted as gradient descent
  - Each tree is fitted to best approximate the negative gradient direction
  - Adding tree to the ensemble moves towards that direction.
- Feature learning perspective:
  - Every new tree is a new (greedily selected) feature.
  - The final classifier uses a linear combination of these learned features.