INT104 ARTIFICIAL INTELLIGENCE

L7- Decision Trees and Random Forests

Fang Kang

Fang.Kang@xjtlu.edu.cn

Xi'an Jiaotong-Liverpool University

西交利物浦大學

CONTENT

- **Decision Tree**
  - Decision Tree
  - Information Gain
  - Impurity
  - Regularization
- **Ensemble Learning and Random Forests**
  - Ensemble Learning
  - Bagging and Pasting/Random Subspace
  - Random Forests
  - Boosting
  - Stacking


Decision Tree

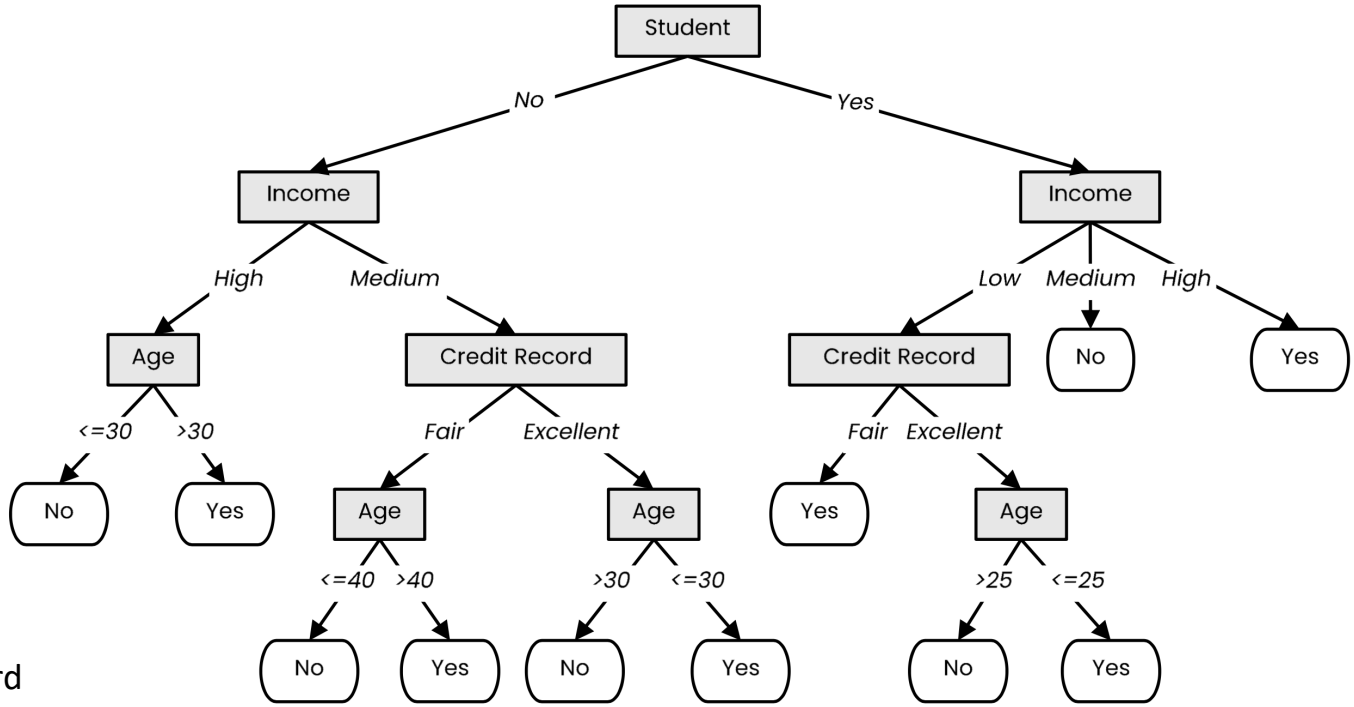
Decision Tree Definition

- A tree-like model that illustrates series of events leading to certain decisions
- Each node represents a test on an attribute and each branch is an outcome of that test

Who to loan?

- Not a student
- 45 years old
- Medium income
- Fair credit record

- Student
- 27 years old
- Low income
- Excellent credit record



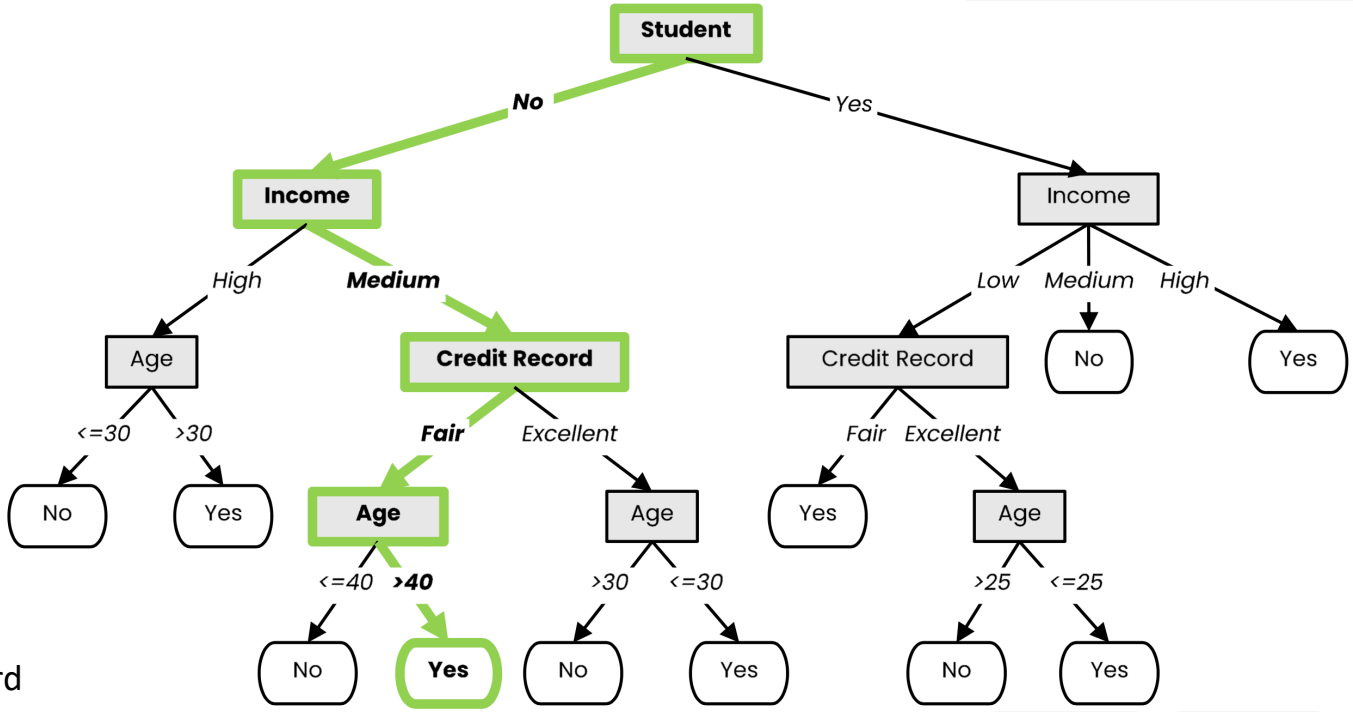
Decision Tree Definition

- A tree-like model that illustrates series of events leading to certain decisions
- Each node represents a test on an attribute and each branch is an outcome of that test

Who to loan?

- Not a student
- 45 years old
- Medium income
- Fair credit record
- ➢ Yes

- Student
- 27 years old
- Low income
- Excellent credit record



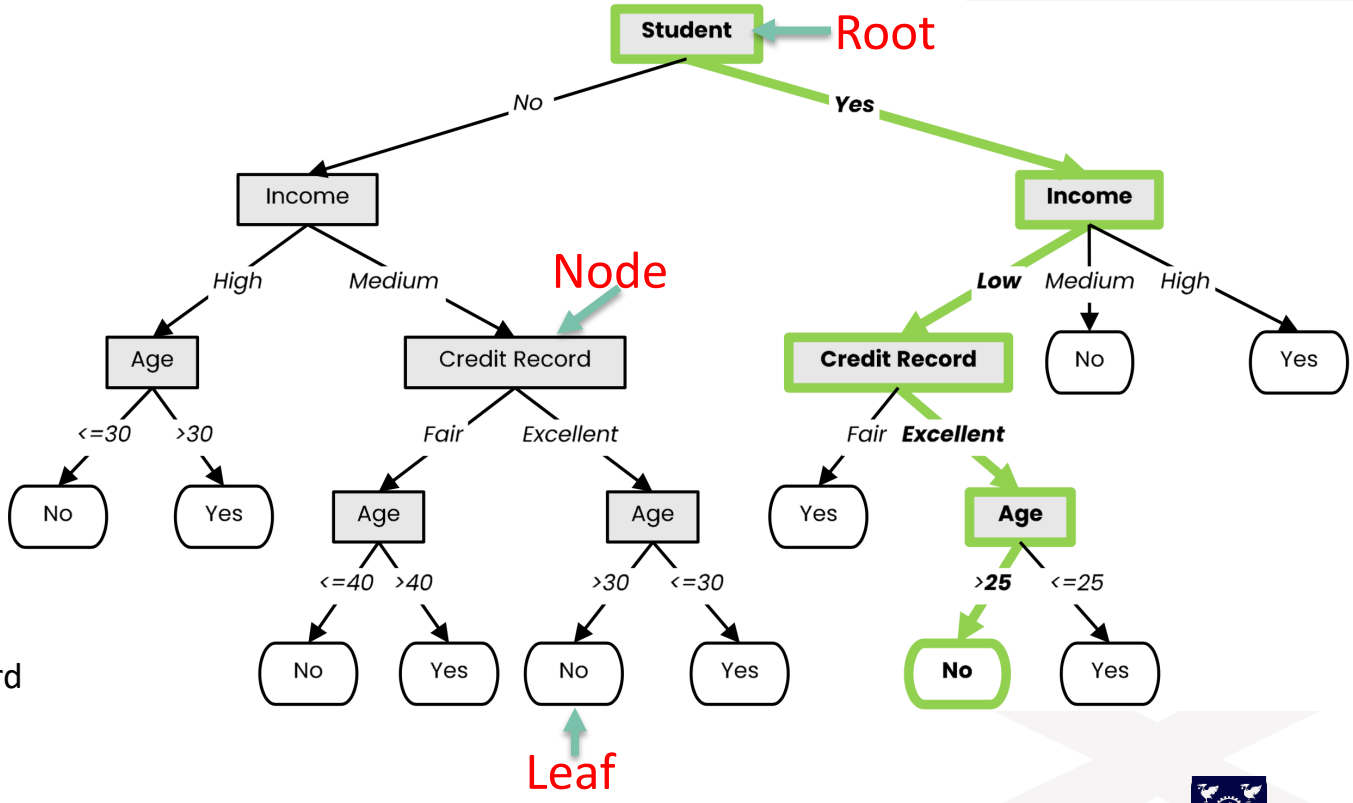
Decision Tree Definition

- A tree-like model that illustrates series of events leading to certain decisions
- Each node represents a test on an attribute and each branch is an outcome of that test

Who to loan?

- Not a student
- 45 years old
- Medium income
- Fair credit record
- ➢ Yes

- Student
- 27 years old
- Low income
- Excellent credit record
- ➢ No



Depth: the length of the longest path from the root node to a leaf node

Decision Tree Learning

- We use labeled data to obtain a suitable decision tree for future predictions
  - ➢ We want a decision tree that works well on unseen data, while asking as few questions as possible

Outlook	Temperature	Humidity	Wind	Play Tennis?
Sunny	Hot	High	Weak	**No**
Sunny	Hot	High	Strong	**No**
Overcast	Hot	High	Weak	**Yes**
Rainy	Mild	High	Weak	**Yes**
Rainy	Cool	Normal	Weak	**Yes**
Rainy	Cool	Normal	Strong	**No**
Overcast	Cool	Normal	Strong	**Yes**
Sunny	Mild	High	Weak	**No**
Sunny	Cool	Normal	Weak	**Yes**
Rainy	Mild	Normal	Weak	**Yes**
Sunny	Mild	Normal	Strong	**Yes**
Overcast	Mild	High	Strong	**Yes**
Overcast	Hot	Normal	Weak	**Yes**
Rainy	Mild	High	Strong	**No**

Decision Tree Learning

- Basic step: choose an attribute and, based on its values, split the data into smaller sets
  - ➢ Recursively repeat this step until we can surely decide the label

Outlook	Temperature	Humidity	Wind	Play Tennis?
Sunny	Hot	High	Weak	**No**
Sunny	Hot	High	Strong	**No**
Overcast	Hot	High	Weak	**Yes**
Rainy	Mild	High	Weak	**Yes**
Rainy	Cool	Normal	Weak	**Yes**
Rainy	Cool	Normal	Strong	**No**
Overcast	Cool	Normal	Strong	**Yes**
Sunny	Mild	High	Weak	**No**
Sunny	Cool	Normal	Weak	**Yes**
Rainy	Mild	Normal	Weak	**Yes**
Sunny	Mild	Normal	Strong	**Yes**
Overcast	Mild	High	Strong	**Yes**
Overcast	Hot	Normal	Weak	**Yes**
Rainy	Mild	High	Strong	**No**

Outlook

Decision Tree Learning

- Basic step: choose an attribute and, based on its values, split the data into smaller sets
  - ➢ Recursively repeat this step until we can surely decide the label

Outlook = *Sunny*

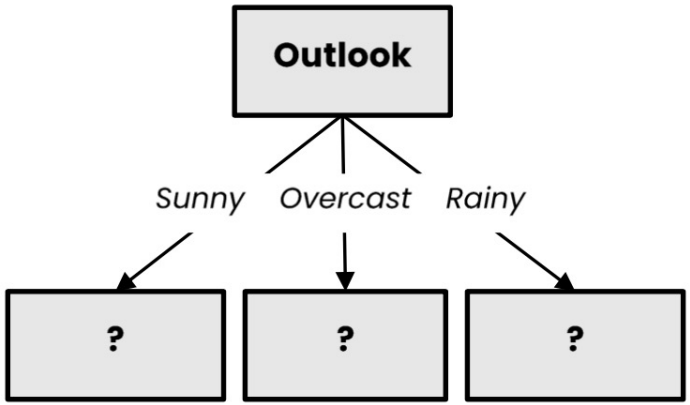
Temperature	Humidity	Wind	Play Tennis?
Hot	High	Weak	**No**
Hot	High	Strong	**No**
Mild	High	Weak	**No**
Cool	Normal	Weak	**Yes**
Mild	Normal	Strong	**Yes**

Outlook = *Overcast*

Temperature	Humidity	Wind	Play Tennis?
Hot	High	Weak	**Yes**
Cool	Normal	Strong	**Yes**
Mild	High	Strong	**Yes**
Hot	Normal	Weak	**Yes**

Outlook = *Rainy*

Temperature	Humidity	Wind	Play Tennis?
Mild	High	Weak	**Yes**
Cool	Normal	Weak	**Yes**
Cool	Normal	Strong	**No**
Mild	Normal	Weak	**Yes**
Mild	High	Strong	**No**



Decision Tree Learning

- Basic step: choose an attribute and, based on its values, split the data into smaller sets
  - ➢ Recursively repeat this step until we can surely decide the label

Outlook = *Sunny*

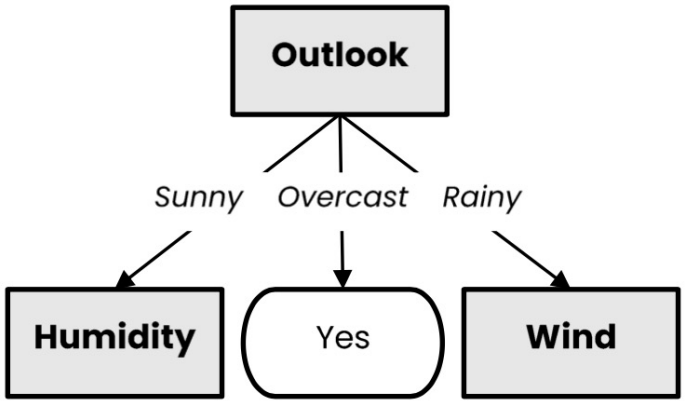
Temperature	Humidity	Wind	Play Tennis?
Hot	High	Weak	**No**
Hot	High	Strong	**No**
Mild	High	Weak	**No**
Cool	Normal	Weak	**Yes**
Mild	Normal	Strong	**Yes**

Outlook = *Overcast*

Temperature	Humidity	Wind	Play Tennis?
Hot	High	Weak	**Yes**
Cool	Normal	Strong	**Yes**
Mild	High	Strong	**Yes**
Hot	Normal	Weak	**Yes**

Outlook = *Rainy*

Temperature	Humidity	Wind	Play Tennis?
Mild	High	Weak	**Yes**
Cool	Normal	Weak	**Yes**
Cool	Normal	Strong	**No**
Mild	Normal	Weak	**Yes**
Mild	High	Strong	**No**



Decision Tree Learning

- Basic step: choose an attribute and, based on its values, split the data into smaller sets
  - ➢ Recursively repeat this step until we can surely decide the label

Outlook = *Sunny*

Humidity = High

Temperature	Wind	Play Tennis?
Hot	Weak	**No**
Hot	Strong	**No**
Mild	Weak	**No**

Humidity = Normal

Temperature	Wind	Play Tennis?
Cool	Weak	**Yes**
Mild	Strong	**Yes**

Outlook = *Overcast*

Temperature	Humidity	Wind	Play Tennis?
Hot	High	Weak	**Yes**
Cool	Normal	Strong	**Yes**
Mild	High	Strong	**Yes**
Hot	Normal	Weak	**Yes**

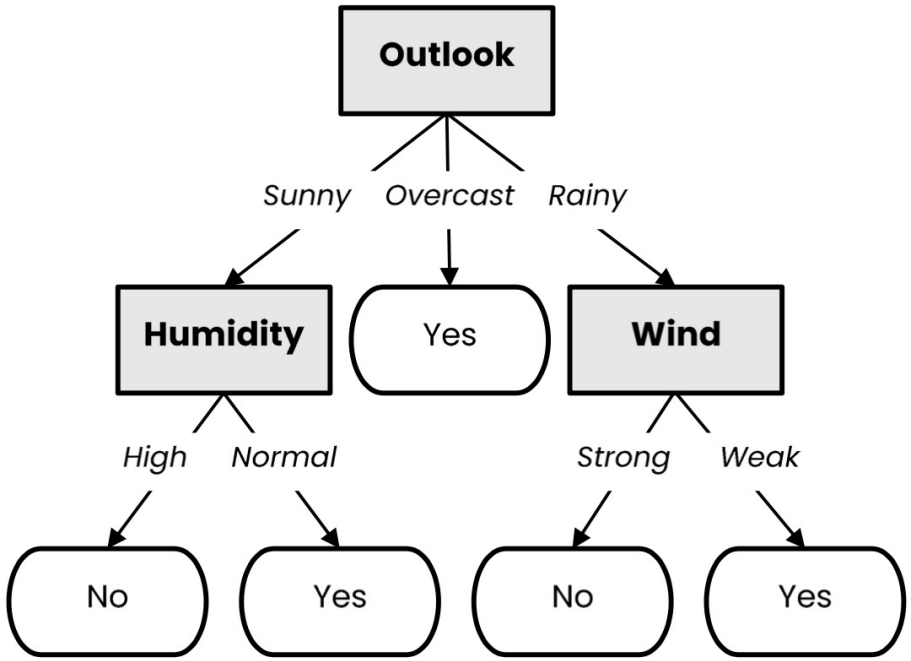
Outlook = *Rainy*

Wind = Strong

Temperature	Humidity	Play Tennis?
Cool	Normal	**No**
Mild	High	**No**

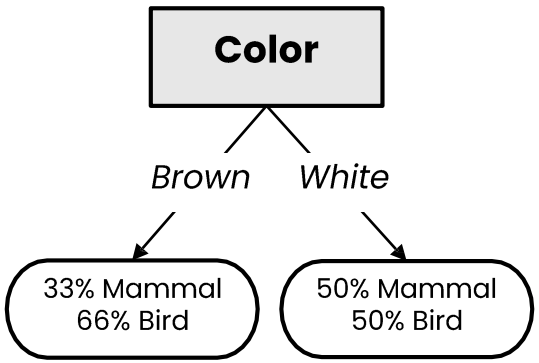
Wind = Weak

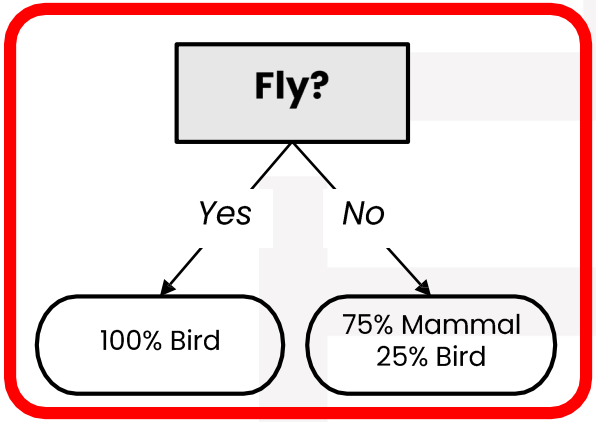
Temperature	Humidity	Play Tennis?
Mild	High	**Yes**
Cool	Normal	**Yes**
Mild	Normal	**Yes**



What is a good attribute?

Does it fly?	Color	Class
No	Brown	**Mammal**
No	White	**Mammal**
Yes	Brown	**Bird**
Yes	White	**Bird**
No	White	**Mammal**
No	Brown	**Bird**
Yes	White	**Bird**





- Which attribute provides better separating?
- Why?
  - Because the resulting subsets are more pure
  - Knowing the value of this attribute gives us more information about the label (the entropy of the subsets is lower)


Information Gain

Entropy

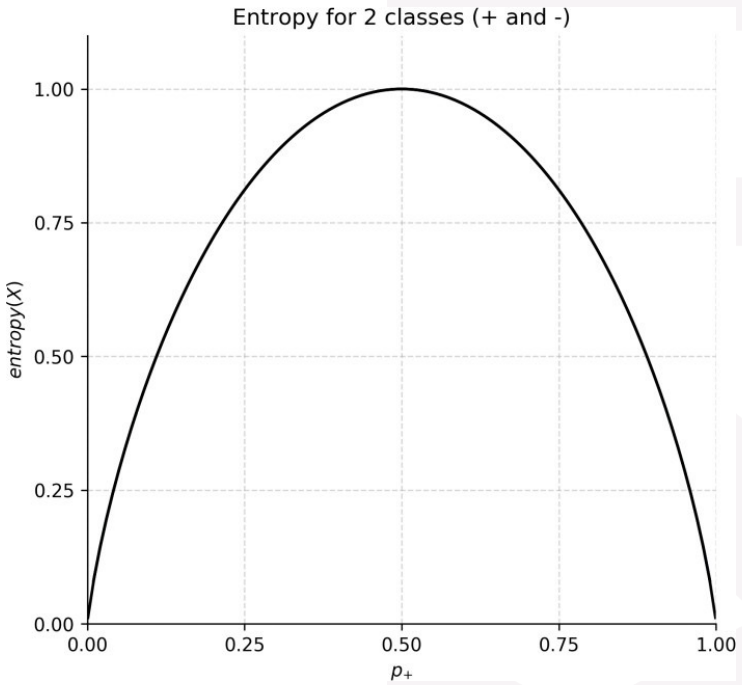
- Entropy measures the degree of randomness in data

- For a set of samples X with k classes:

$$entropy(X) = -\sum_{i=1}^{k} p_i \log_2(p_i)$$

where p_i is the proportion of elements of class i

- Lower entropy implies greater predictability!



Information Gain

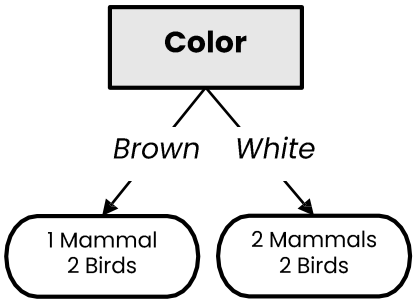
- The information gain of an attribute ***a*** is the expected reduction in entropy due to splitting on values of ***a*** :

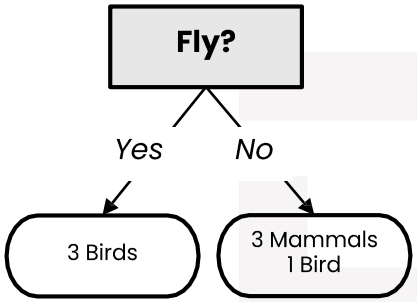
$$gain(X, a) = entropy(X) - \sum_{v \in Values(a)} \frac{|X_v|}{|X|} entropy(X_v)$$

where X_v is the subset of X for which $a = v$

Best attribute = highest information gain

Does it fly?	Color	Class
No	Brown	**Mammal**
No	White	**Mammal**
Yes	Brown	**Bird**
Yes	White	**Bird**
No	White	**Mammal**
No	Brown	**Bird**
Yes	White	**Bird**





$$entropy\ (X) = -p_{\text{mammal}} \log_2 p_{\text{mammal}} - p_{\text{bird}} \log_2 p_{\text{bird}} = -\frac{3}{7} \log_2 \frac{3}{7} - \frac{4}{7} \log_2 \frac{4}{7} \approx 0.985$$

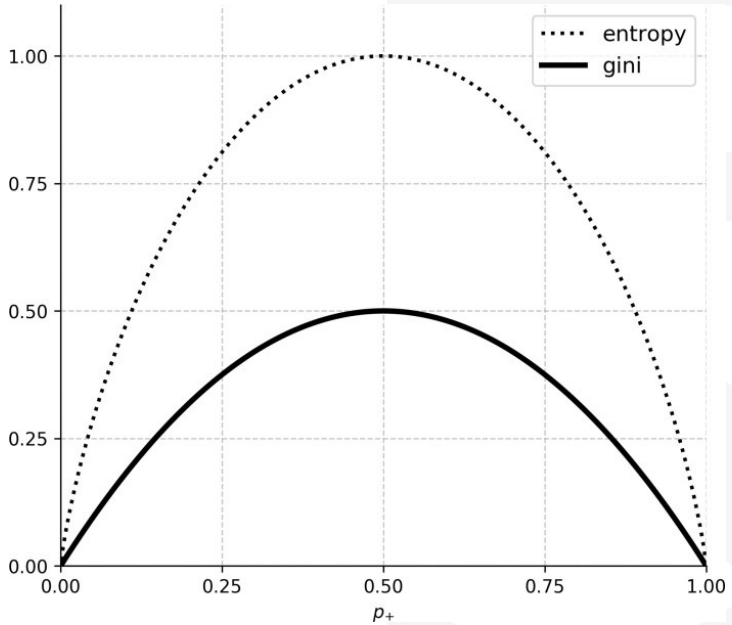
Information gain of Color and Fly ?


Gini Impurity

Gini Impurity

- Gini impurity measures how often a randomly chosen example would be incorrectly labeled if it was randomly labeled according to the label distribution

Error of classifying randomly picked fruit with randomly picked label



- For a set of samples X with k classes:

$$gini(X) = 1 - \sum_{i=1}^{k} p_i^2$$

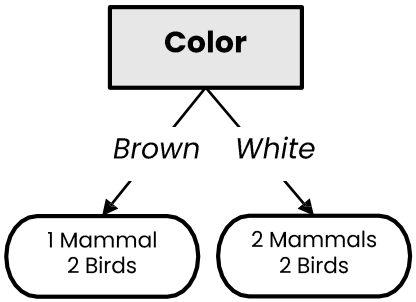
where p_i is the proportion of elements of class i

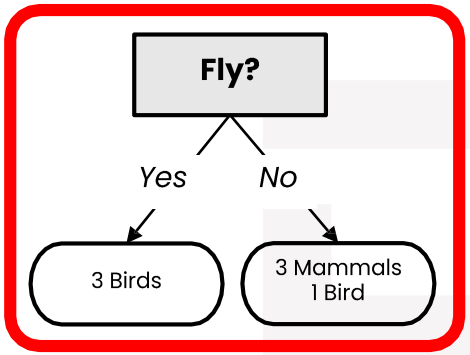
- Can be used as an alternative to entropy for selecting attributes!

Best attribute = lowest Gini impurity

In practice, we compute $gini(X)$ only once!

Does it fly?	Color	Class
No	Brown	**Mammal**
No	White	**Mammal**
Yes	Brown	**Bird**
Yes	White	**Bird**
No	White	**Mammal**
No	Brown	**Bird**
Yes	White	**Bird**





$$gini\ (X_{color=brown}) = 1 - \left(\frac{1}{3}\right)^2 - \left(\frac{2}{3}\right)^2 \approx 0.444$$

$$gini\ (X_{color=white}) = 0.5$$

$$\boldsymbol{gini\ (X, color) = \frac{3}{7} \cdot 0.444 + \frac{4}{7} \cdot 0.5 \approx 0.476}$$

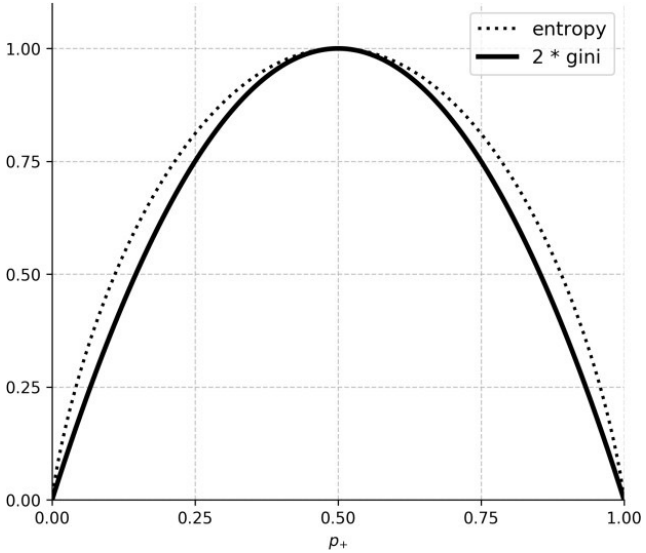
$$gini\ (X_{fly=yes}) = 0$$

$$gini\ (X_{fly=no}) = 1 - \left(\frac{3}{4}\right)^2 - \left(\frac{1}{4}\right)^2 \approx 0.375$$

$$\boldsymbol{gini\ (X, fly) = \frac{3}{7} \cdot 0 + \frac{4}{7} \cdot 0.375 \approx 0.214}$$

Entropy versus Gini Impurity

- Entropy and Gini Impurity give similar results in practice
  - They only disagree in about 2% of cases
    "Theoretical Comparison between the Gini Index and Information Gain Criteria" [Răileanu & Stoffel, AMAI 2004]
  - Entropy might be slower to compute, because of the log



- ID3 ： information gain (decrease in entropy)
- CART： Gini impurity

CART Algorithm：

Splits the training set into two subsets using a single feature (k) and a threshold (t_k)

- **Classification**

  (DecisionTreeClassifier)

  - Predict a *class* in each node
  - Minimize impurity
  - Cost function:

$$J(k, t_k) = \frac{m_{\text{left}}}{m} G_{\text{left}} + \frac{m_{\text{right}}}{m} G_{\text{right}}$$

Regression

(DecisionTreeRegressor)

- Predict a *value* in each node
- Minimize MSE
- Cost function:

$$J(k, t_k) = \frac{m_{\text{left}}}{m} MSE_{\text{left}} + \frac{m_{\text{right}}}{m} MSE_{\text{right}}$$

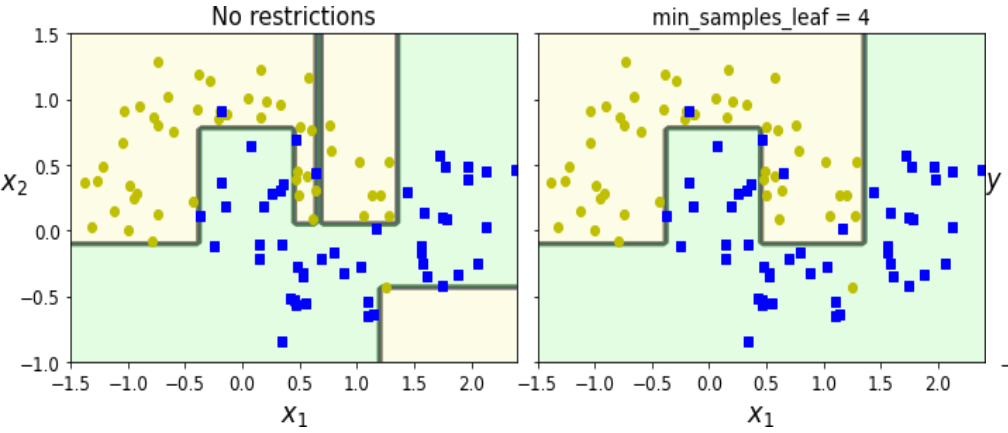
- $G_{\text{left/right}}$ measures the impurity of the left/right subset
- $m_{\text{left/right}}$ is the number of instances in the left/right subset


Regularization

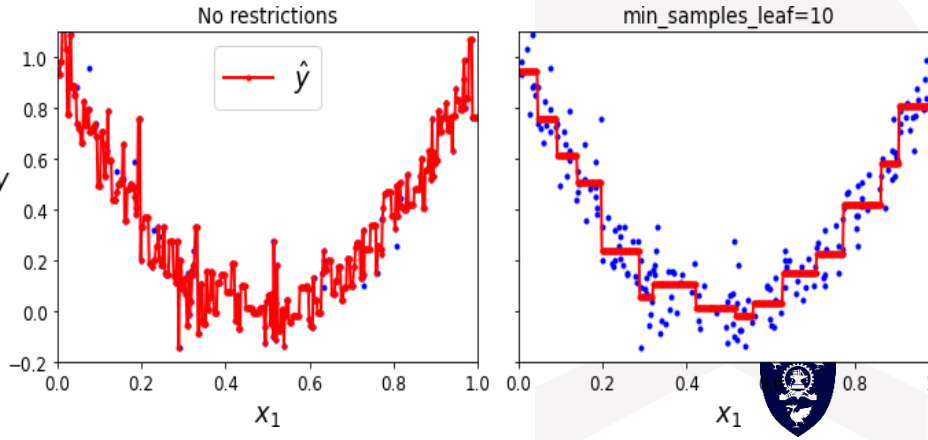
Regularization

Hyperparameter	Introduction
max_depth	The maximum depth of the decision tree
min_samples_split	The minimum number of samples required to split a node
min_samples_leaf	The minimum number of samples required to be at a leaf node
max_leaf_nodes	The maximum number of leaf nodes
max_features	The maximum number of features to consider when looking for the best split

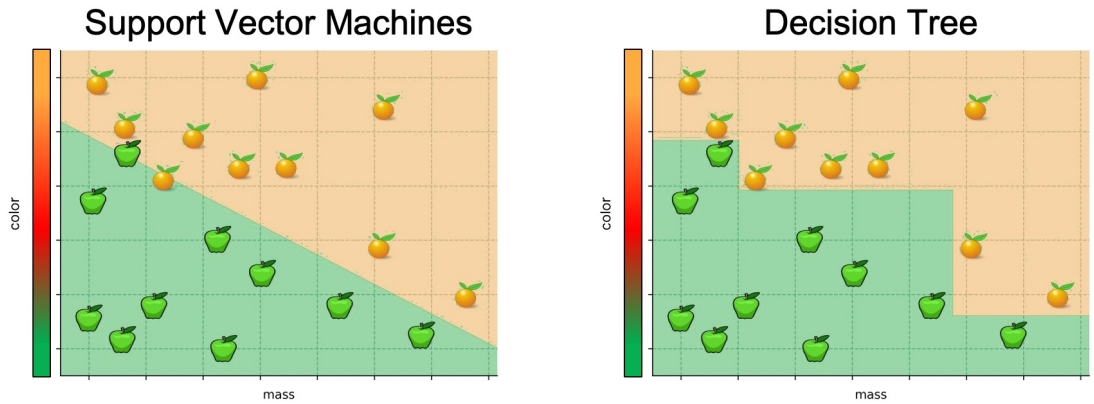
Classification



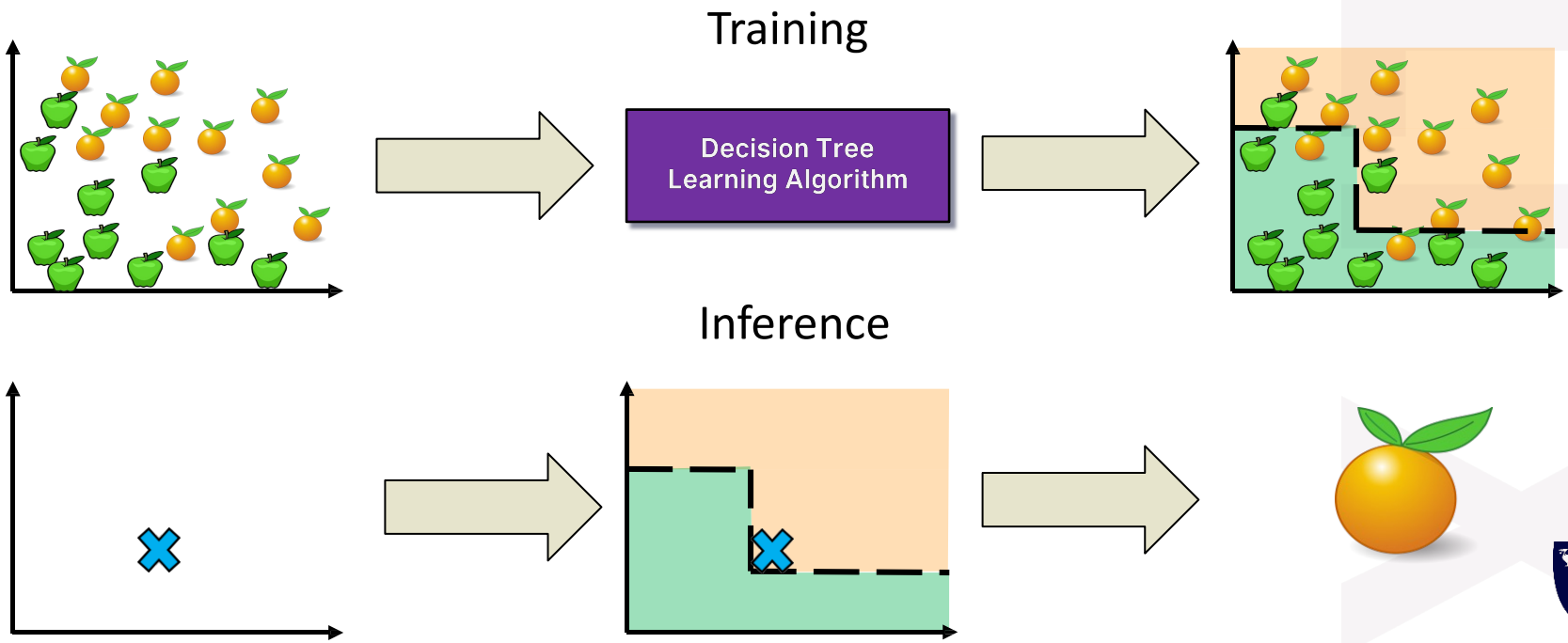
Regression



Decision Trees: Training and Inference



- Decision trees produce non-linear decision boundaries



CONTENT

- **Decision Tree**
  - Decision Tree
  - Information Gain
  - Impurity
  - Regularization and Pruning
- **Ensemble Learning and Random Forests**
  - Ensemble Learning
  - Bagging and Pasting/Random Subspace
  - Random Forests
  - Boosting
  - Stacking

Ensemble Learning

Ensemble Learning

Ensemble : A group of predictors

Voting Classifier

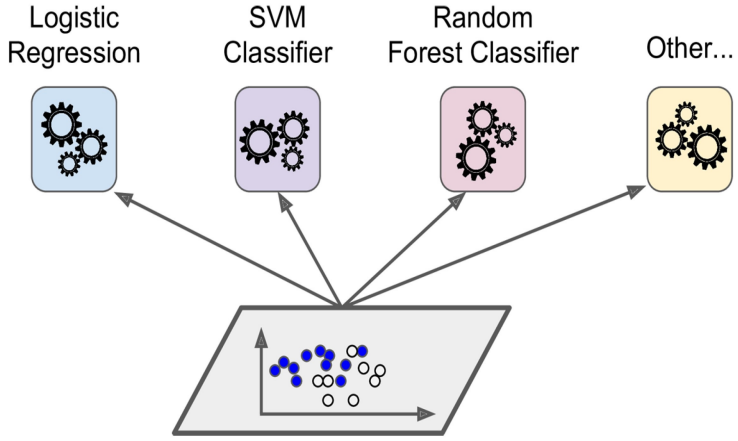


Figure 7-1. Training diverse classifiers

Hard Voting Classifier

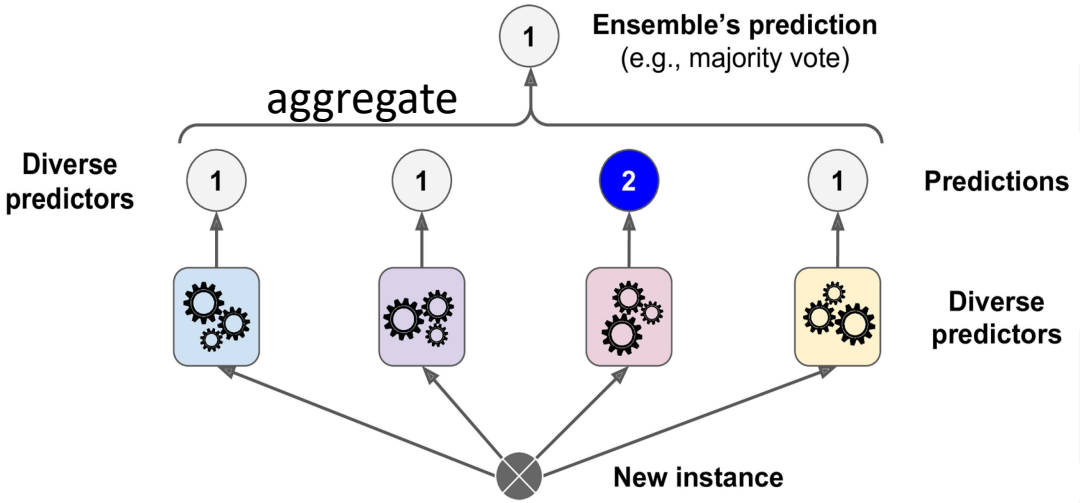


Figure 7-2. Hard voting classifier predictions


Bagging and
Pasting

Bagging (bootstrap aggregating) and Pasting

Training with different subsets of data

- Bagging: Sampling with Replacement
- Pasting: Sampling without Replacement

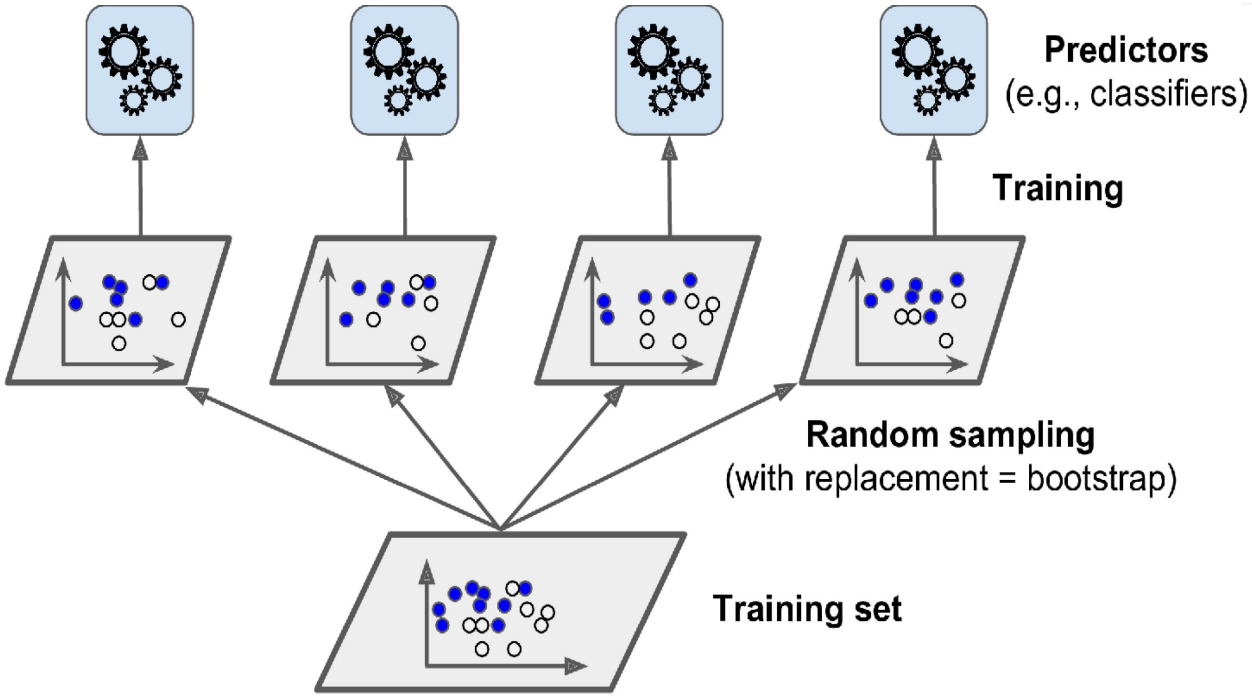
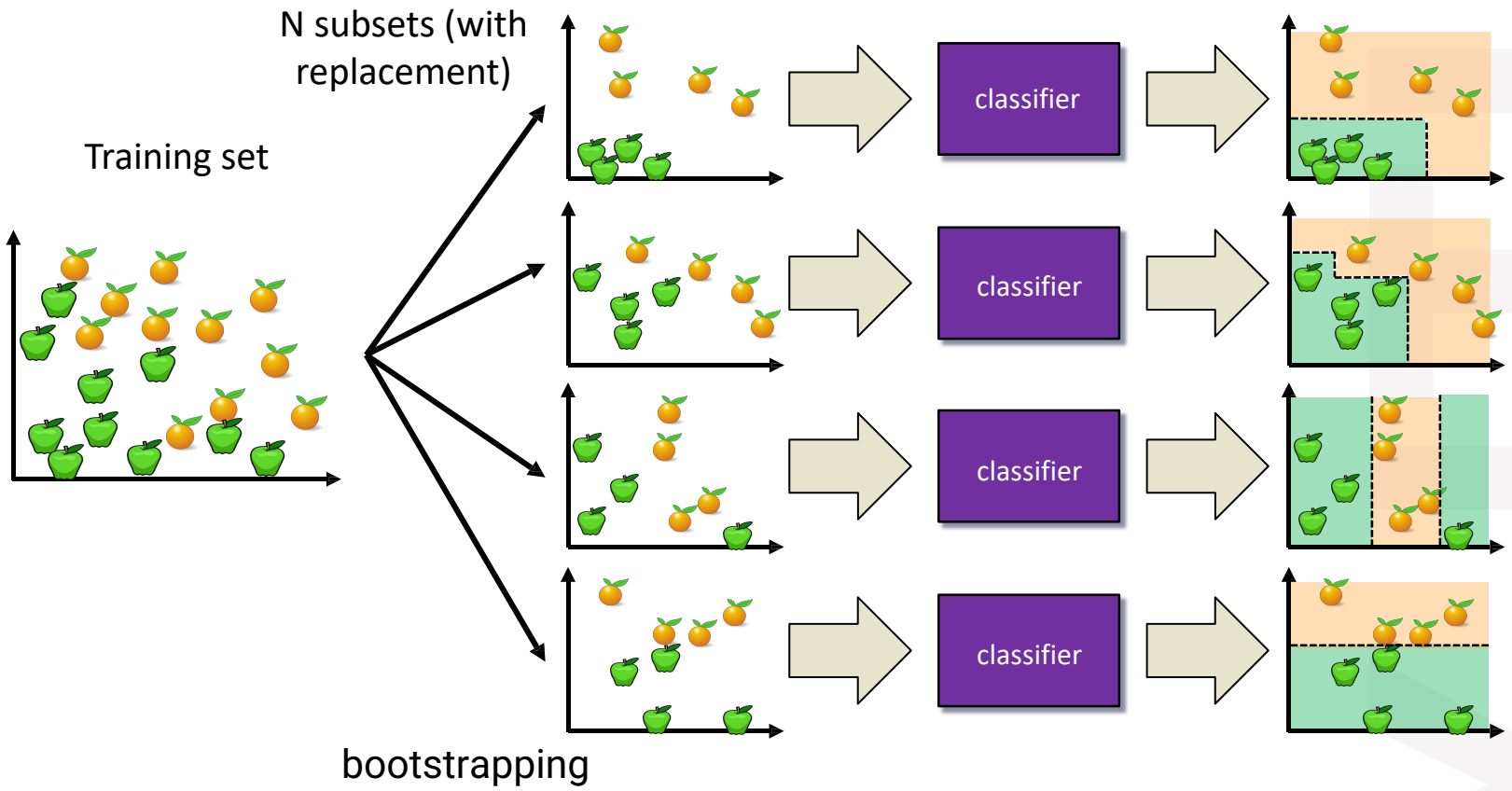
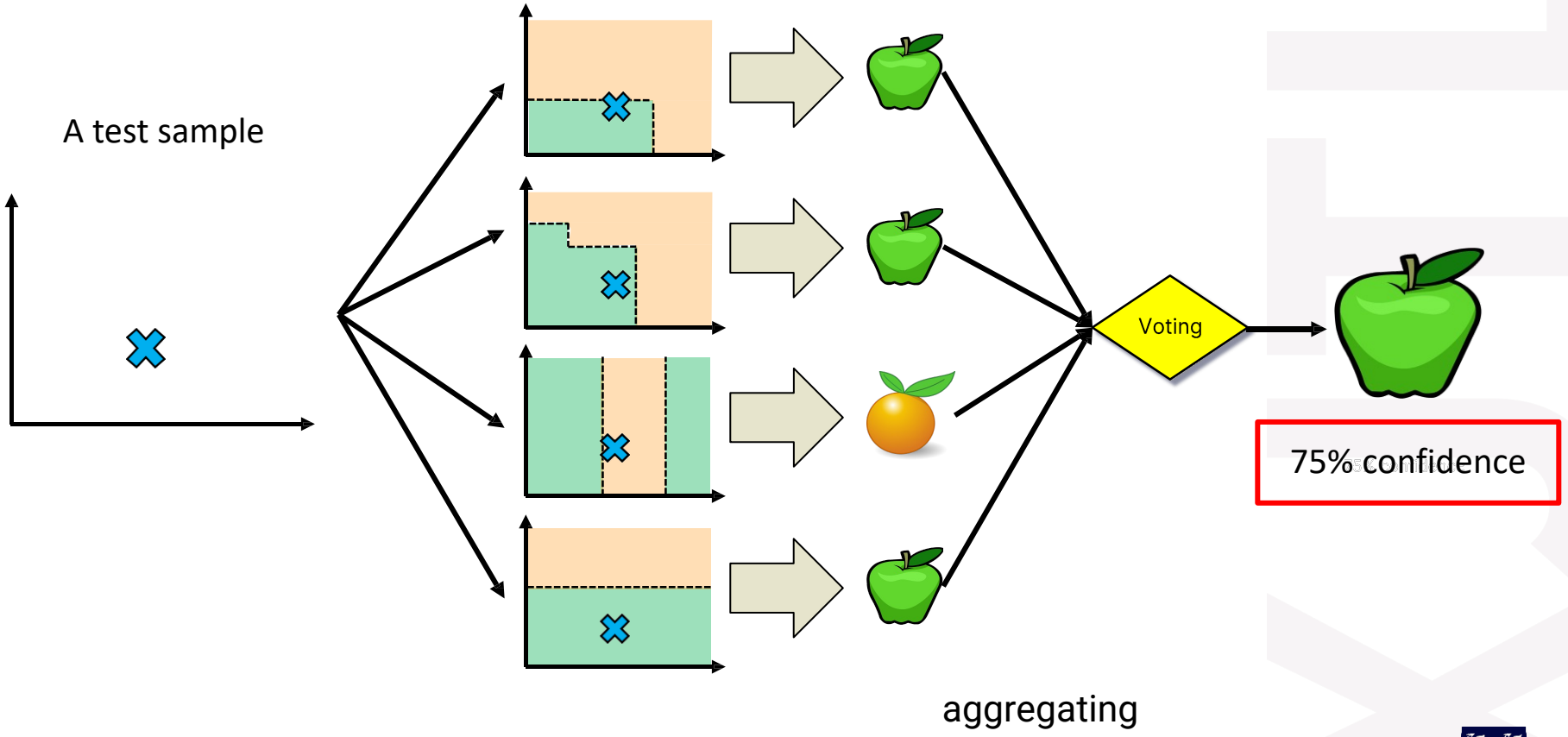


Figure 7-4. Bagging and pasting involves training several predictors on different random samples of the training set

Bagging at training time



Bagging at inference time



Decision Boundary

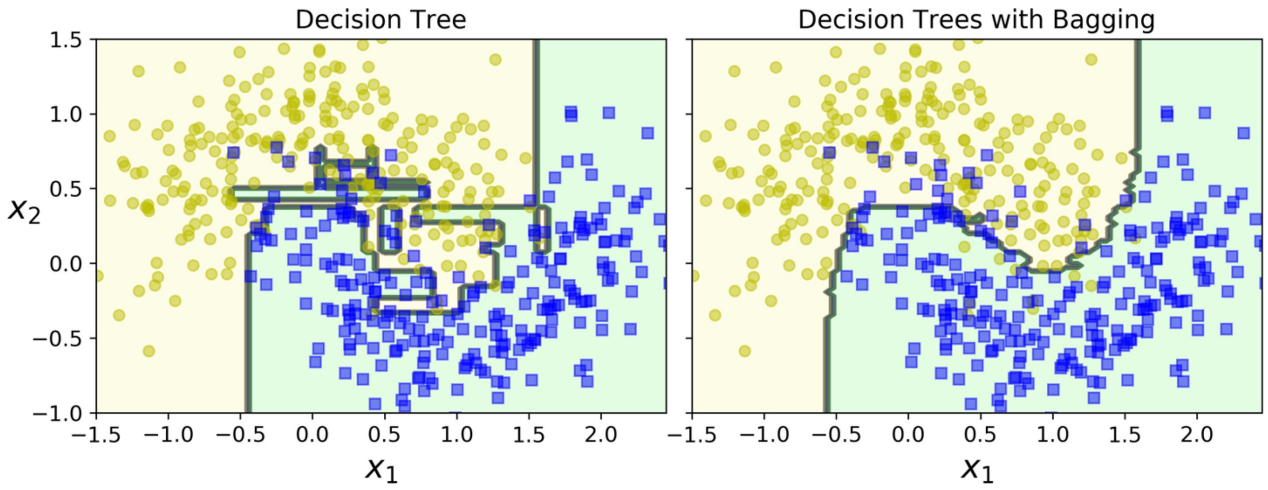


Figure 7-5. A single Decision Tree (left) versus a bagging ensemble of 500 trees (right)

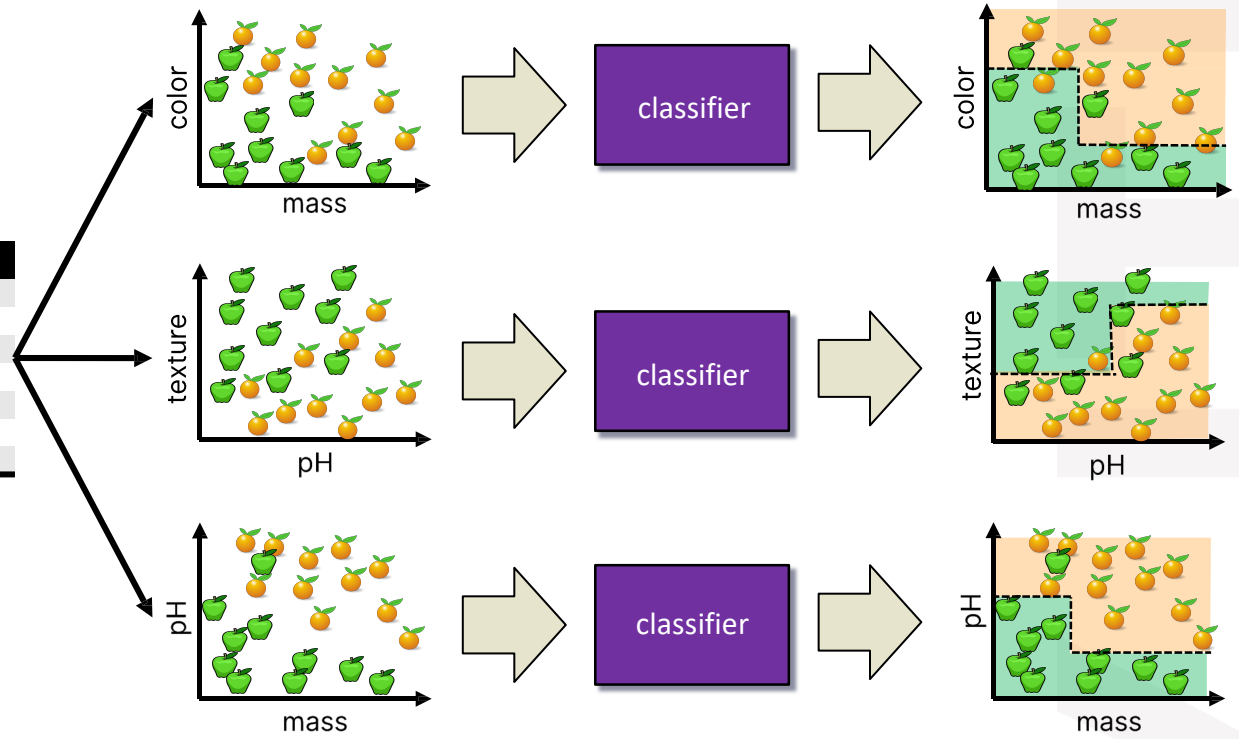
Out-of-Bag Evaluation

we sample with replacement, and therefore **not all instances are used for each bootstrap sample.** On average 1/3 of them are not used!
We call them out-of-bag samples (OOB)

Random Subspace Method at training time

Training data

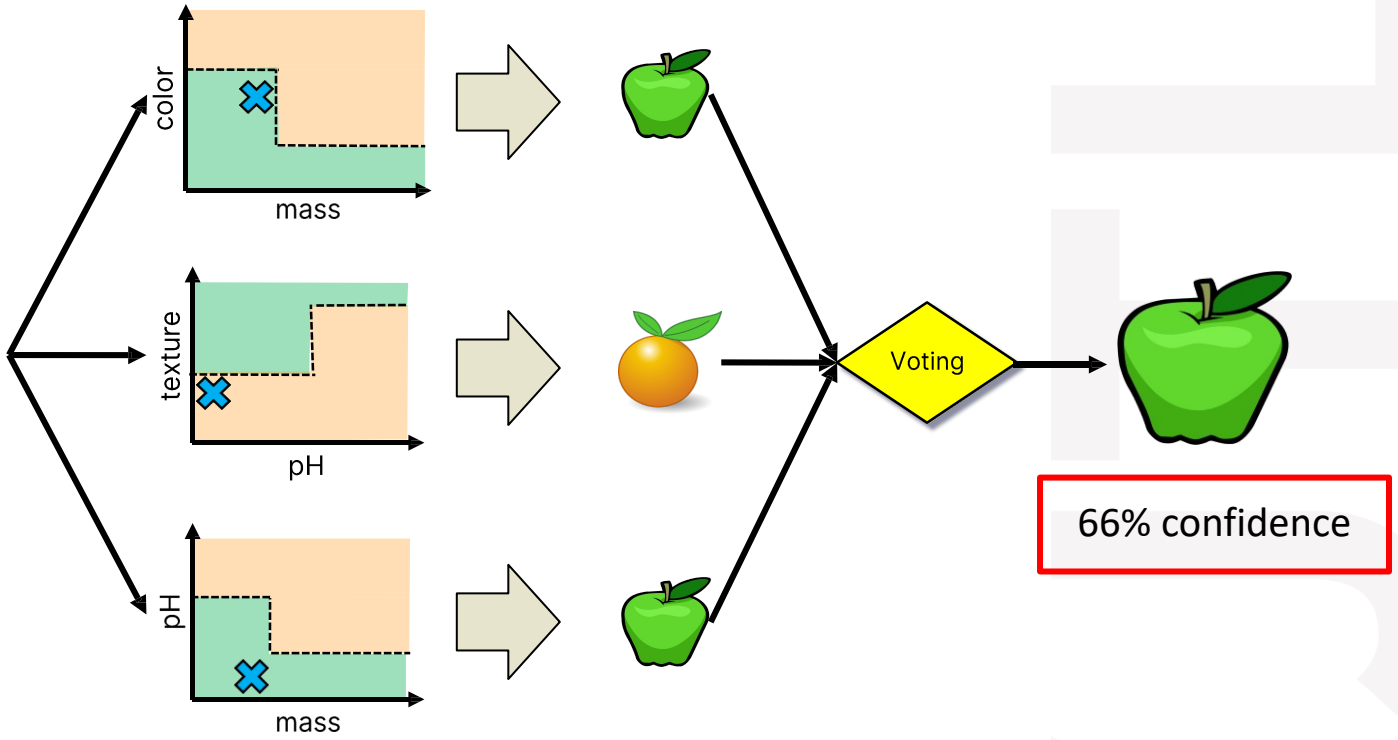
Mass (g)	Color	Texture	pH	Label
84	Green	Smooth	3.5	**Apple**
121	Orange	Rough	3.9	**Orange**
85	Red	Smooth	3.3	**Apple**
101	Orange	Smooth	3.7	**Orange**
111	Green	Rough	3.5	**Apple**
...				
117	Red	Rough	3.4	**Orange**



Random Subspace Method at inference time

A test sample

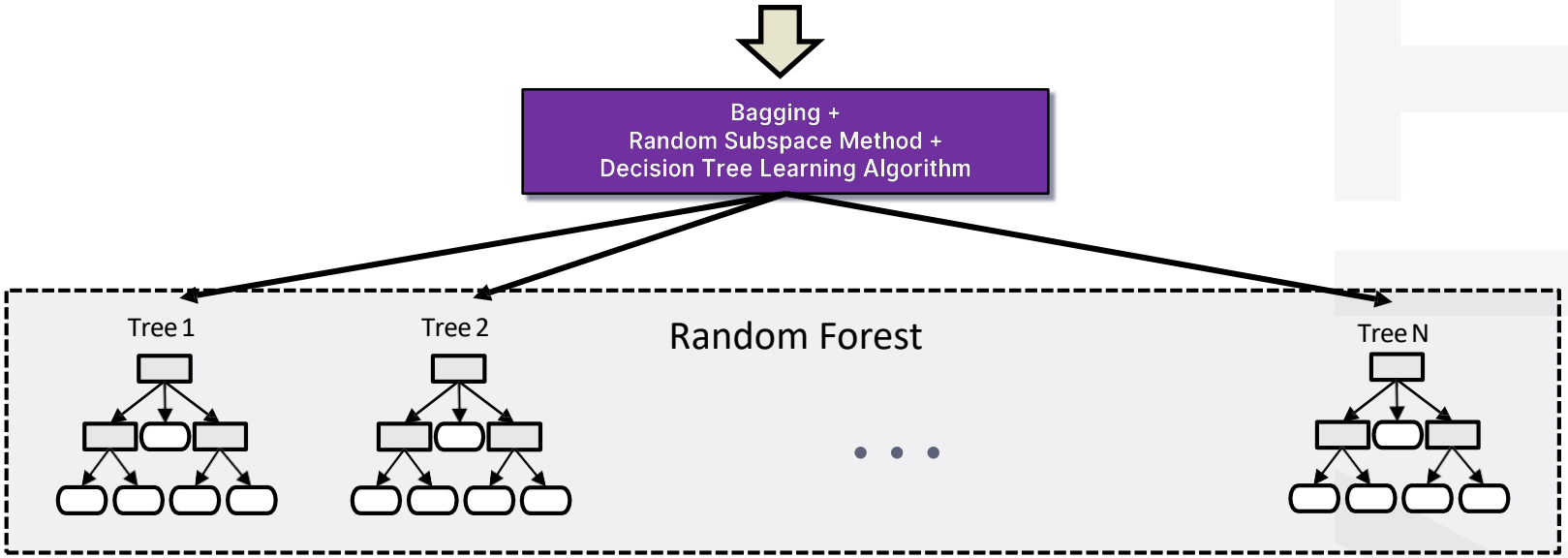
87	Red	Smooth	3.1



Sampling both training instances and features is called ***Random patches*** method.
Reduce the variance of decision trees

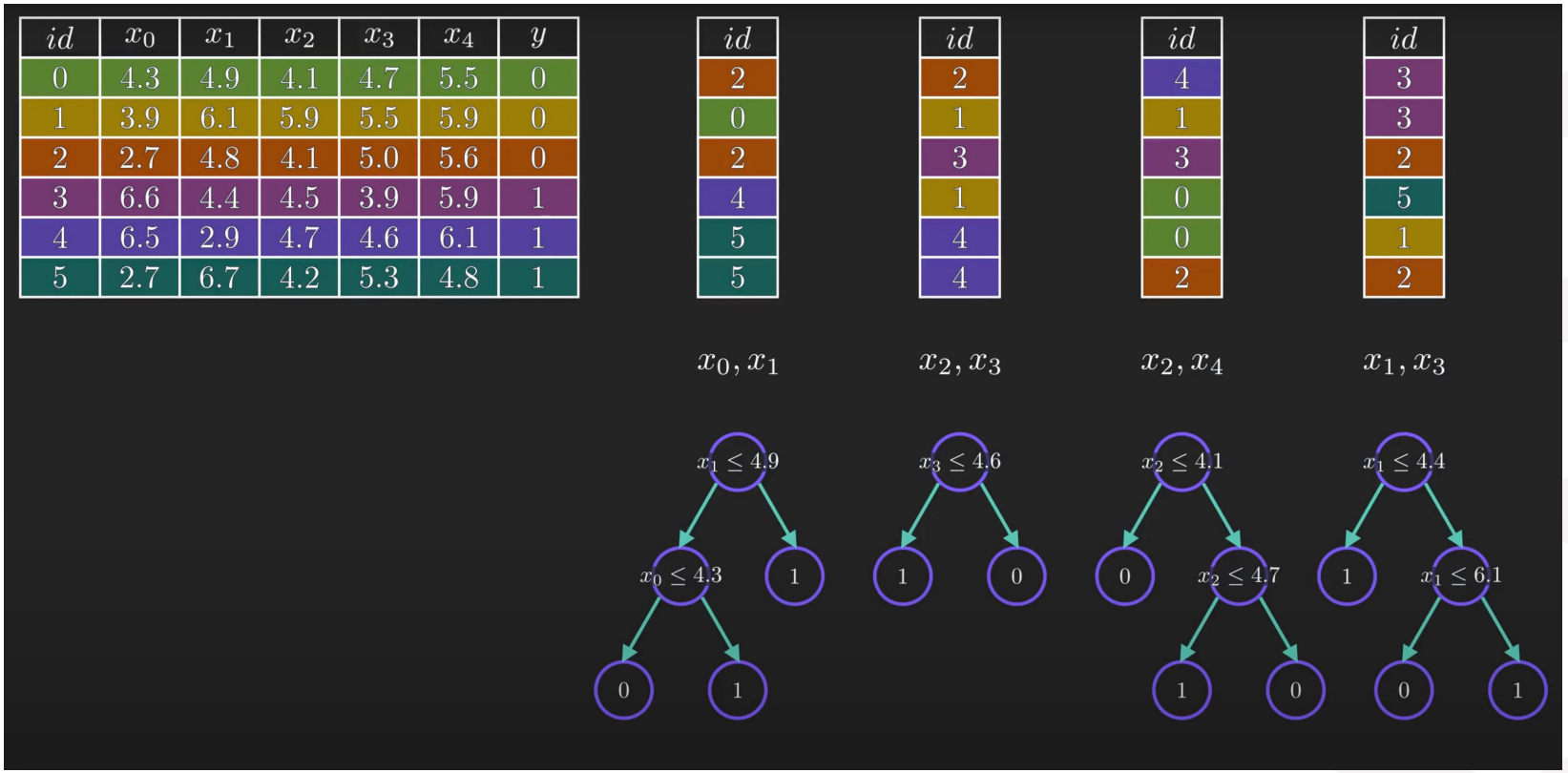
Random Forests

Mass (g)	Color	Texture	pH	Label
84	Green	Smooth	3.5	**Apple**
121	Orange	Rough	3.9	**Orange**
85	Red	Smooth	3.3	**Apple**
101	Orange	Smooth	3.7	**Orange**
111	Green	Rough	3.5	**Apple**
...				
117	Red	Rough	3.4	**Orange**



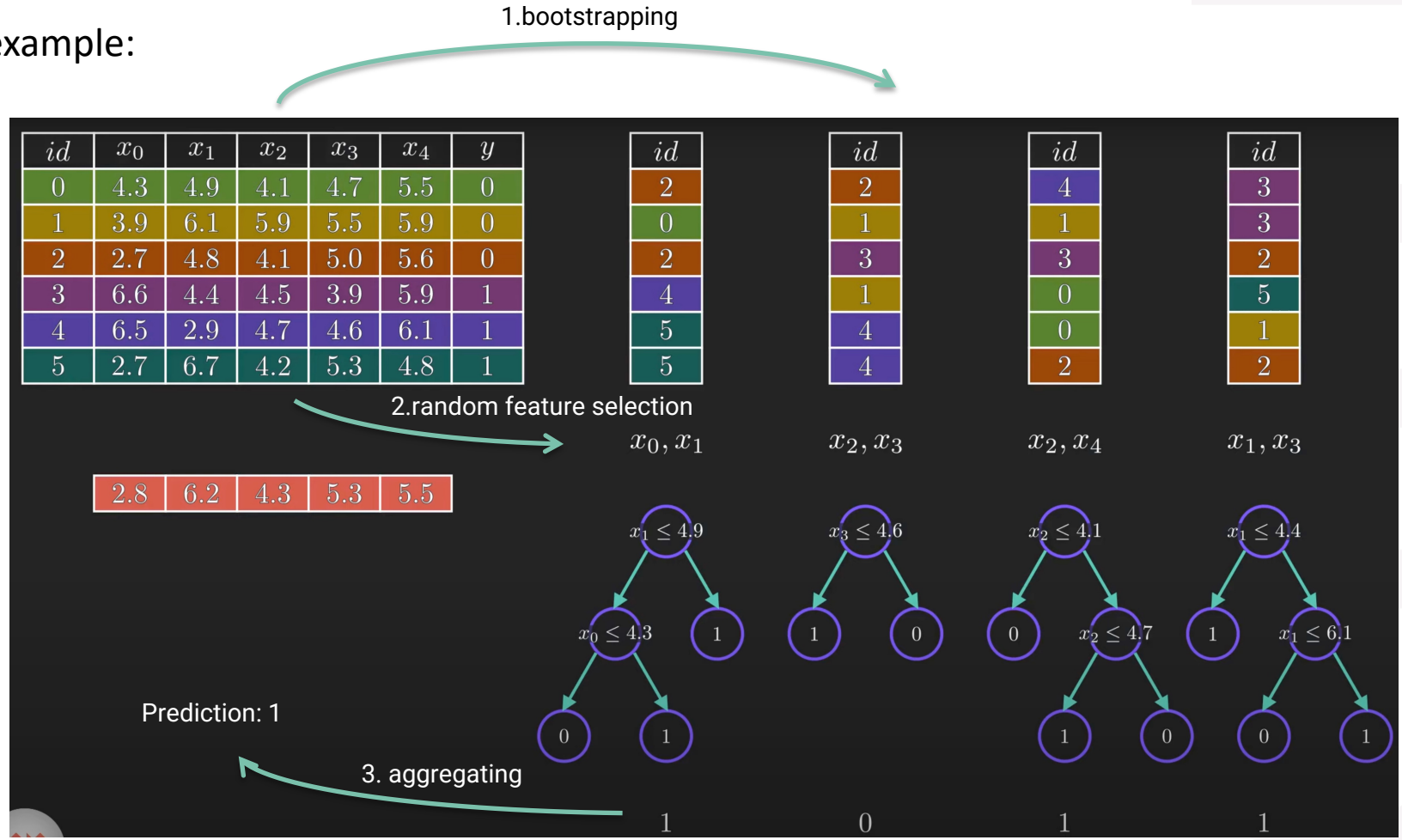
Random Forests

An example:



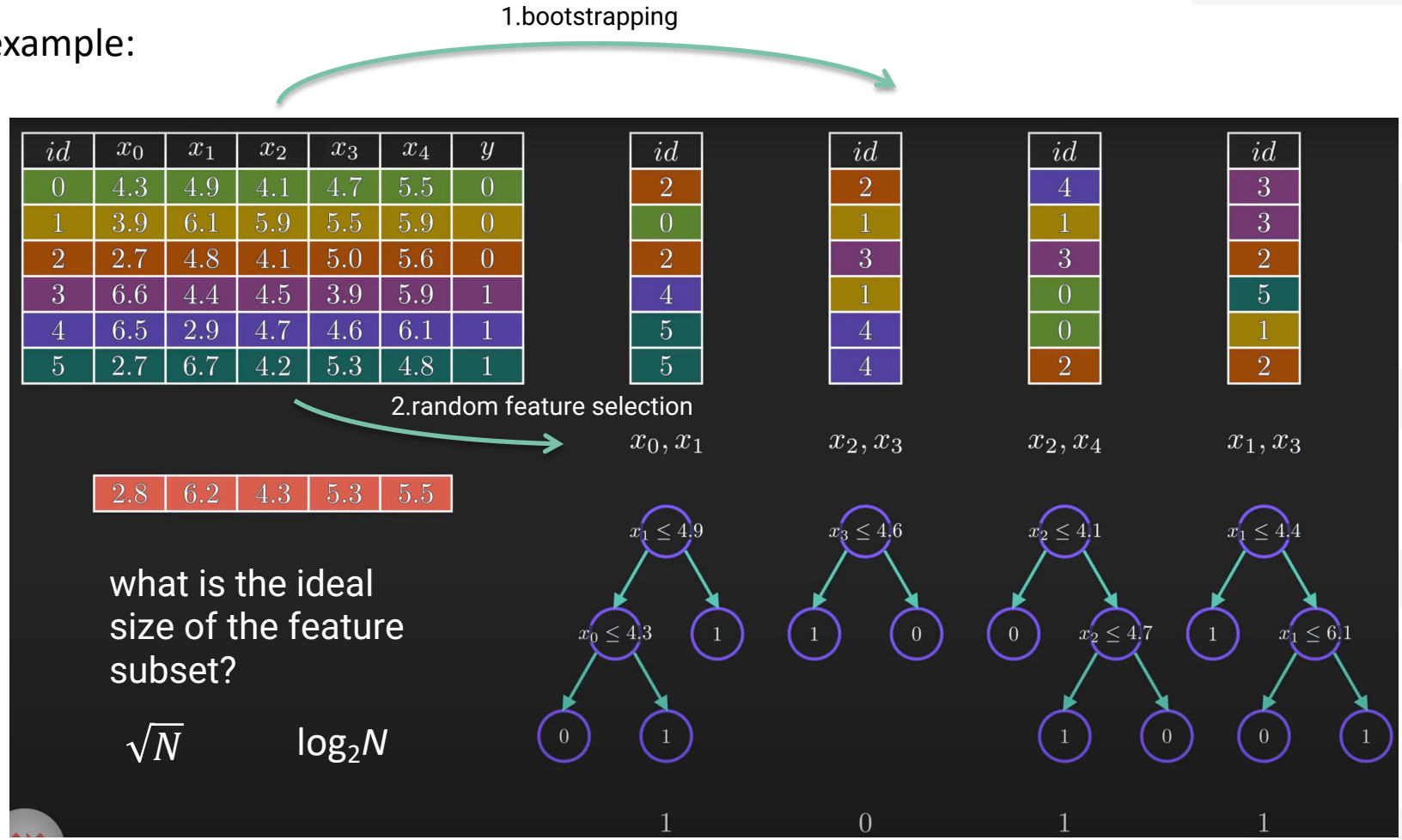
Random Forests

An example:



Random Forests

An example:

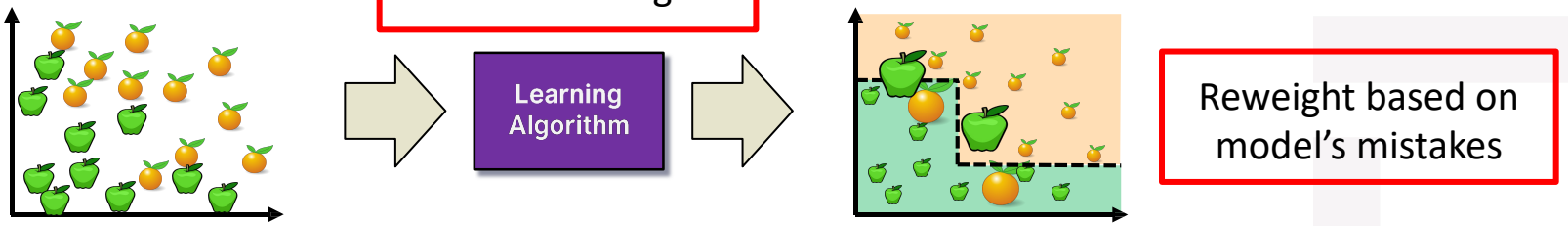


Random Forests

- **Random Forests** are one of the most common examples of ensemble learning
- Other commonly-used ensemble methods:
  - Bagging: multiple models on random subsets of data samples
  - Random Subspace Method: multiple models on random subsets of features
  - Boosting: train models iteratively, while making the current model focus on the mistakes of the previous ones by increasing the weight of misclassified samples

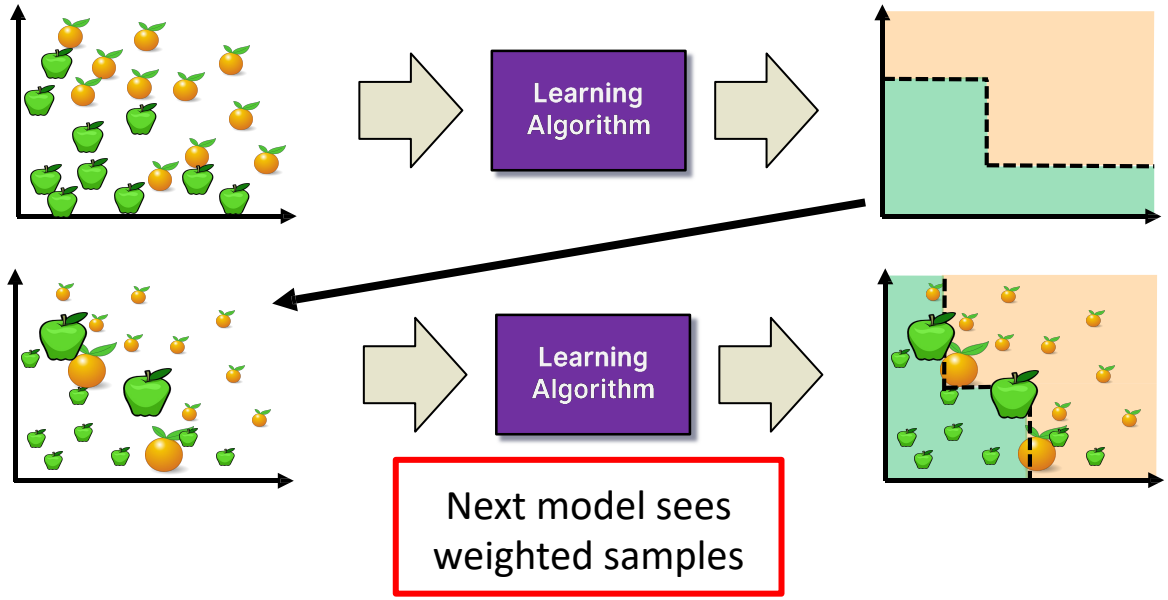
Boosting

Adaptive Boosting (AdaBoost):



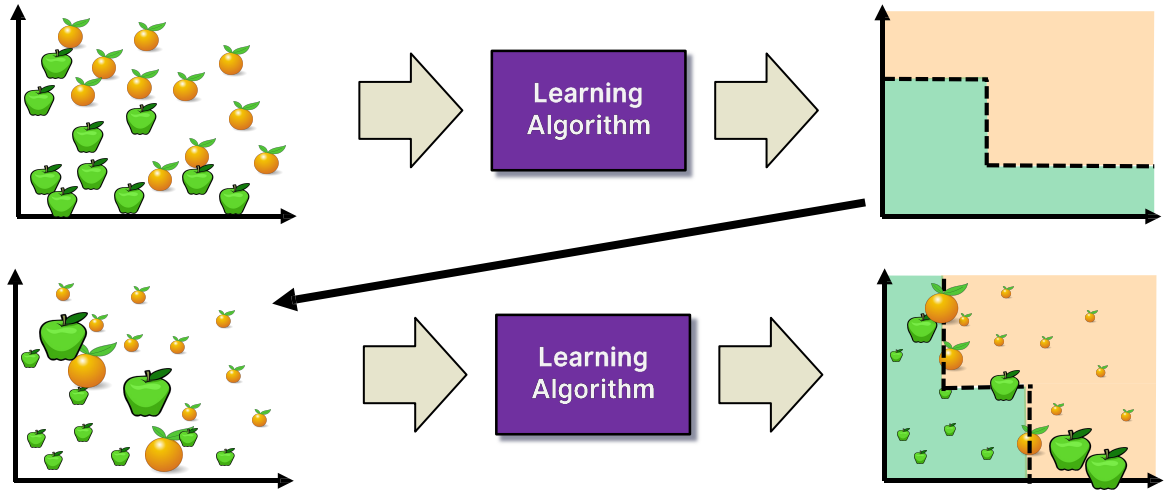
Boosting

AdaBoost：



Boosting

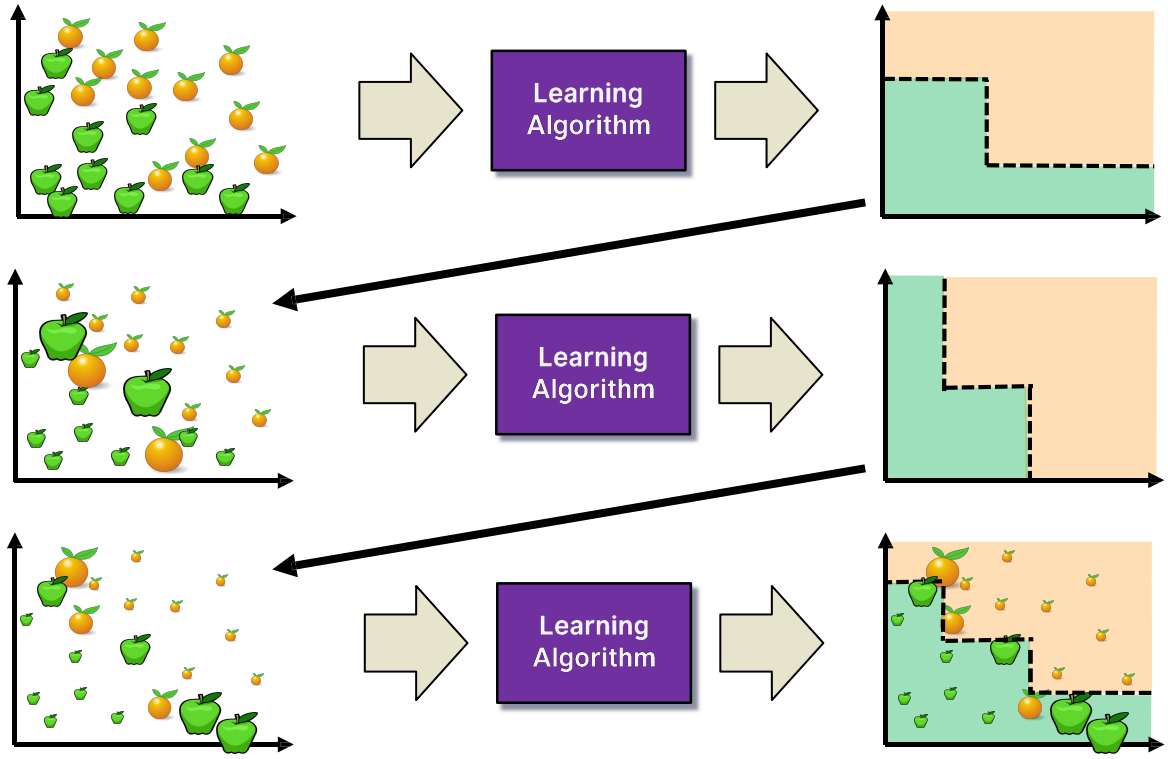
AdaBoost：



Reweight based on current model's mistakes

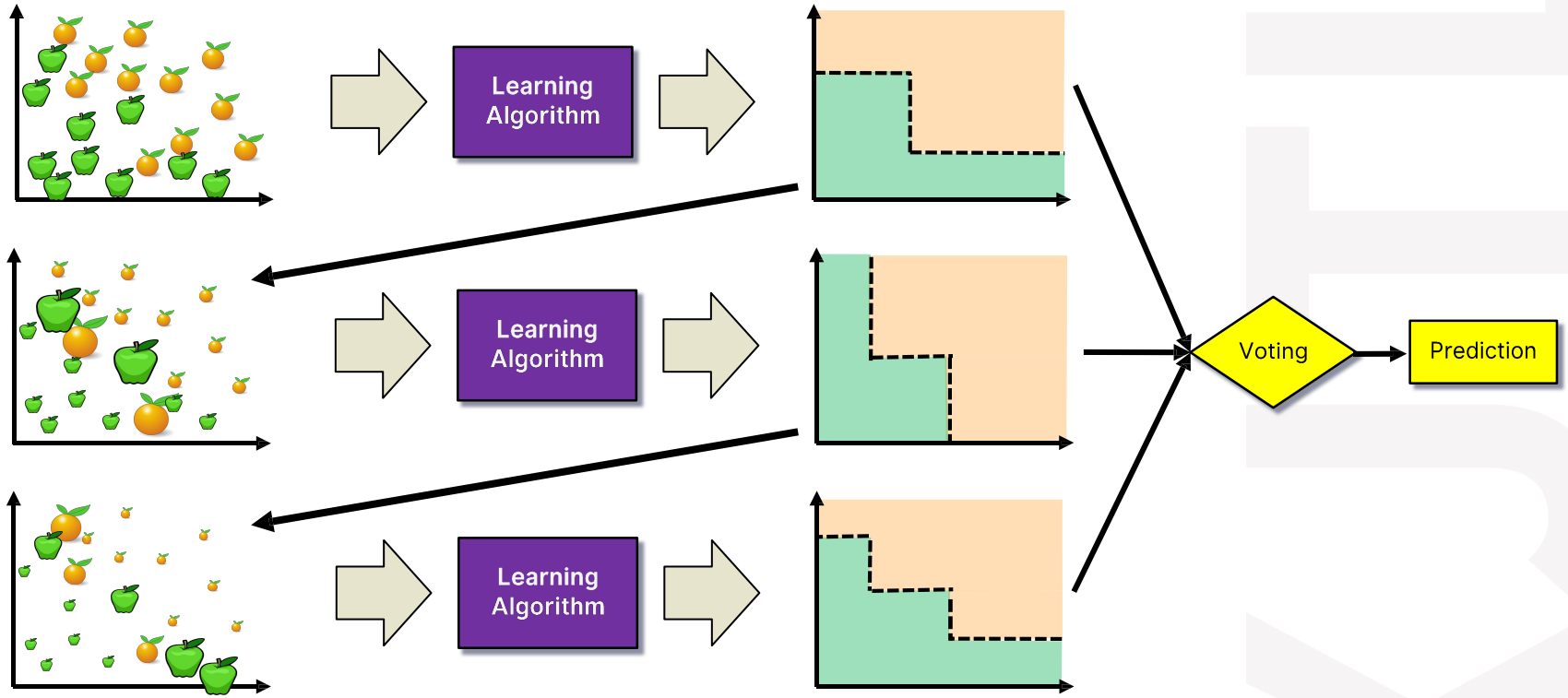
Boosting

AdaBoost：



Boosting

AdaBoost：



Gradient Boosting

- Residue is learned after each iteration

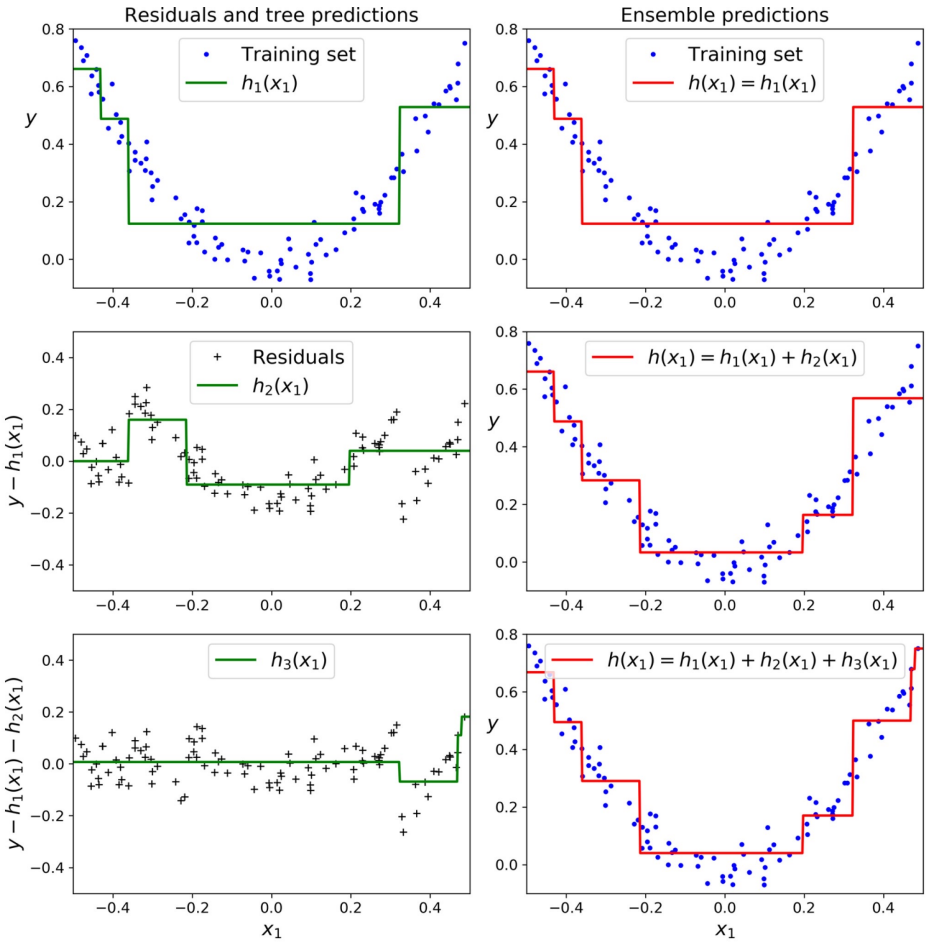


Figure 7-9. In this depiction of Gradient Boosting, the first predictor (top left) is trained normally, then each consecutive predictor (middle left and lower left) is trained on the previous predictor's residuals; the right column shows the resulting ensemble's predictions

Stacking

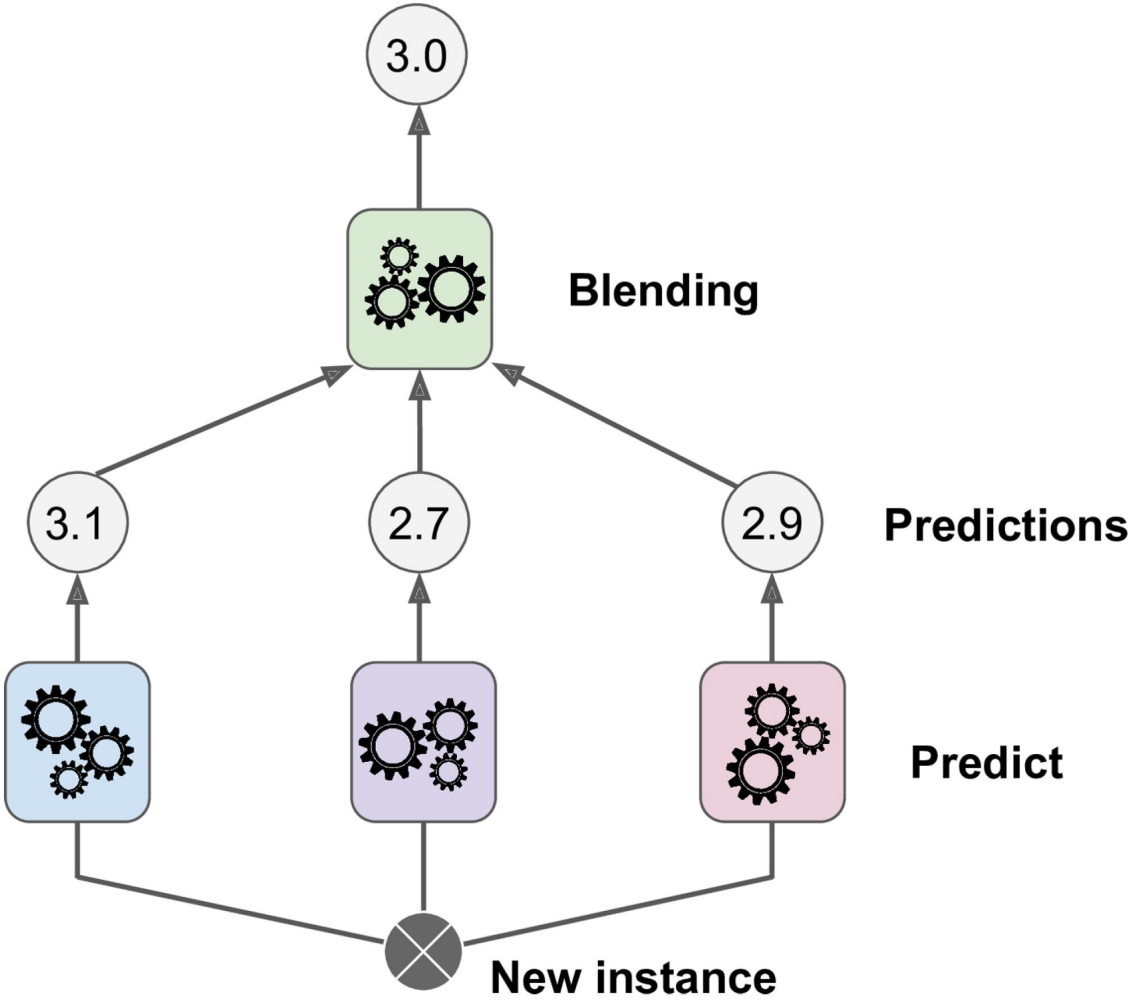


Figure 7-12. Aggregating predictions using a blending predictor

Training the First Layer

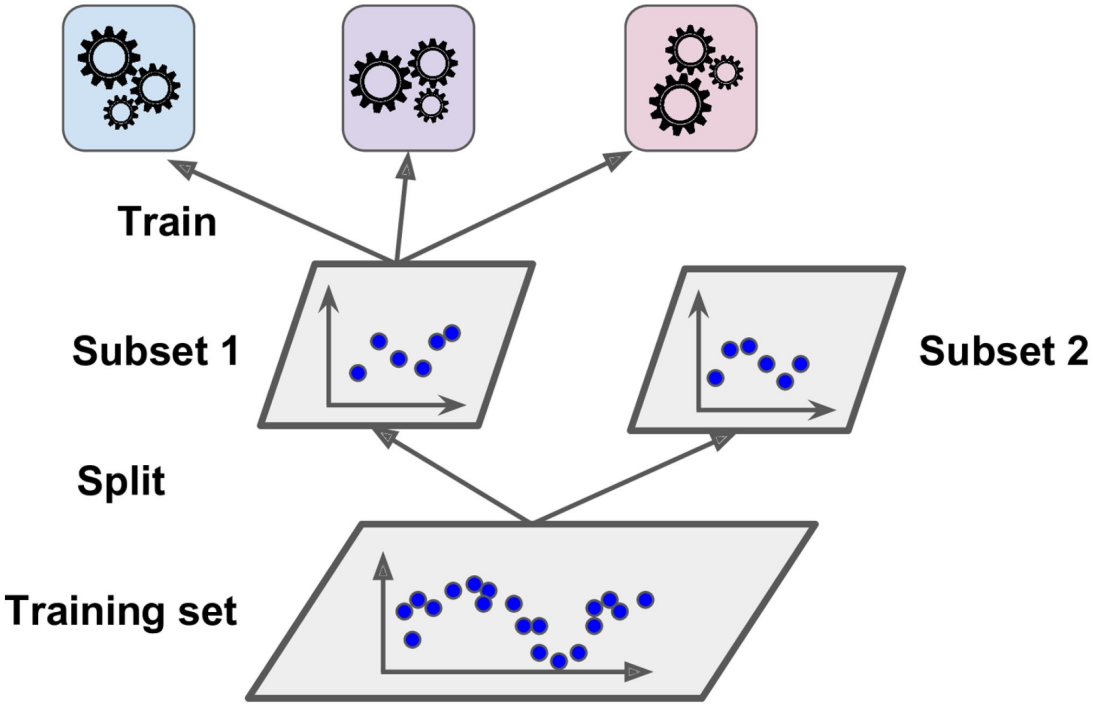


Figure 7-13. Training the first layer

Training the Blender

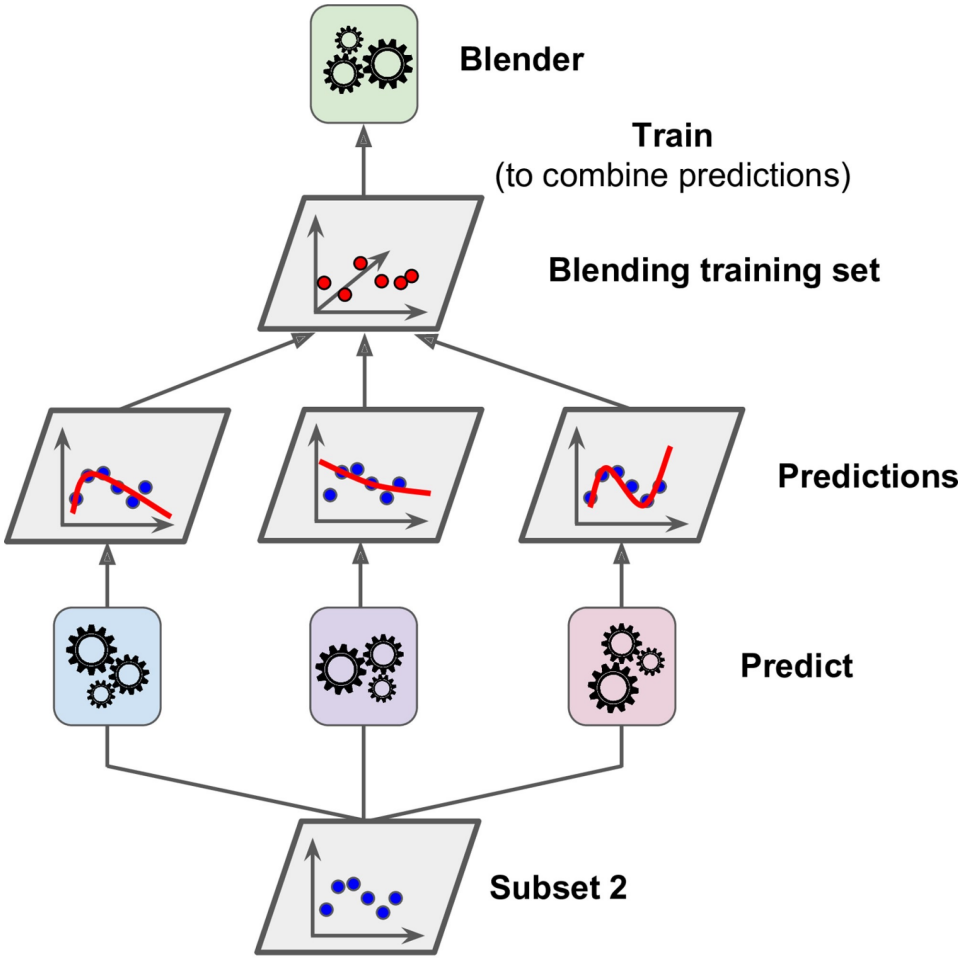


Figure 7-14. Training the blender

Predictions in a Multilayer Stacking Ensemble

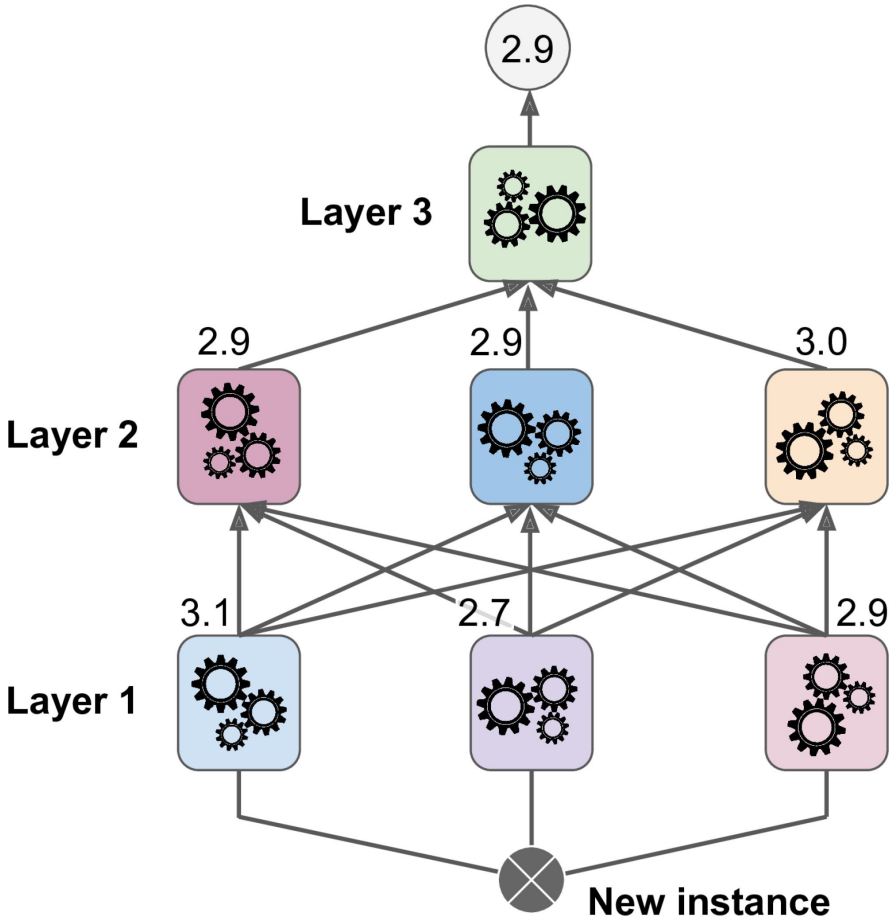


Figure 7-15. Predictions in a multilayer stacking ensemble

THANK YOU

Xi'an Jiaotong-Liverpool University
西交利物浦大學