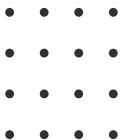
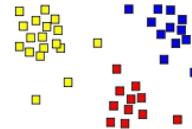
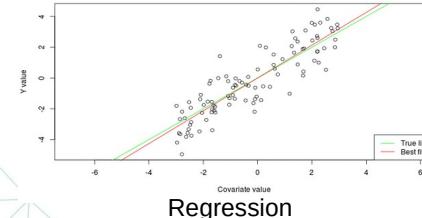
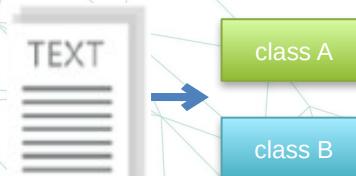


Machine Learning COURSE



Machine Learning Types

- **Supervised:** learning with **labeled data**
 - Example: email classification, image classification
 - Example: regression for predicting real-valued outputs
- **Unsupervised:** discover patterns in **unlabeled data**
 - Example: cluster similar data points



Clustering

Decision Tree and Random Forest

Let's play a Game!

Decision Trees

Quiz 1:

Which column gives us the better split?

In another words

What is the most important column?

Gender	Age	App
F	15	
F	25	
M	32	
F	40	
M	12	
M	14	

Answer

Gender	Age	App
F	15	
F	25	
M	32	
F	40	
M	12	
M	14	

Gender	Age	App
F	15	
F	25	
M	32	
F	40	
M	12	
M	14	

Gender	Age	App
F	15	
F	25	
M	32	
F	40	
M	12	
M	14	



Gender	Age	App
F	25	
M	32	
F	40	



Student Admissions

GRADES

7

2

GRADES

7

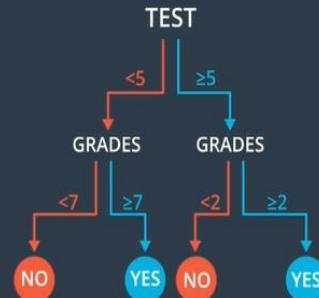
2

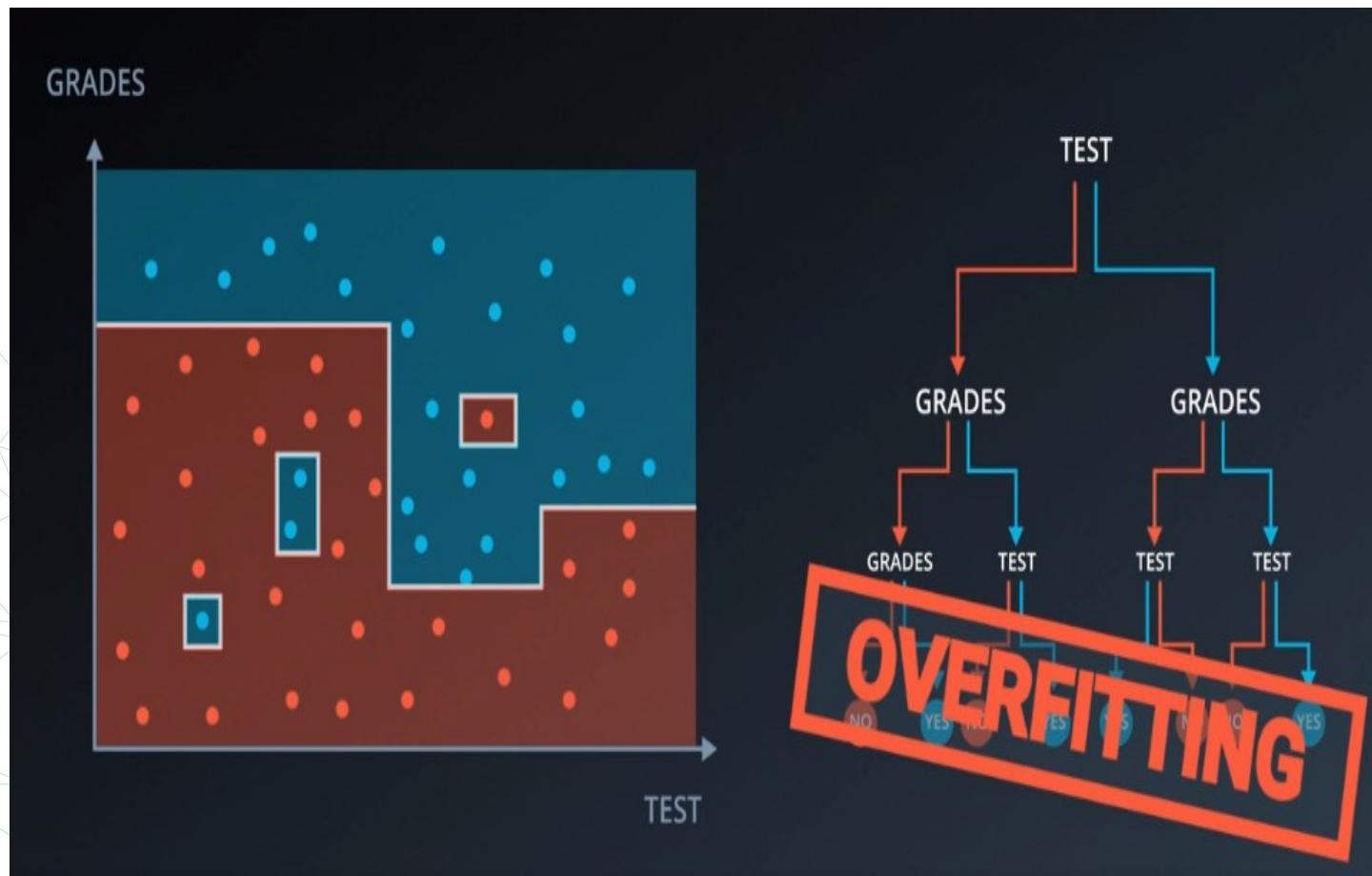
5

TEST

TEST

Student Admissions





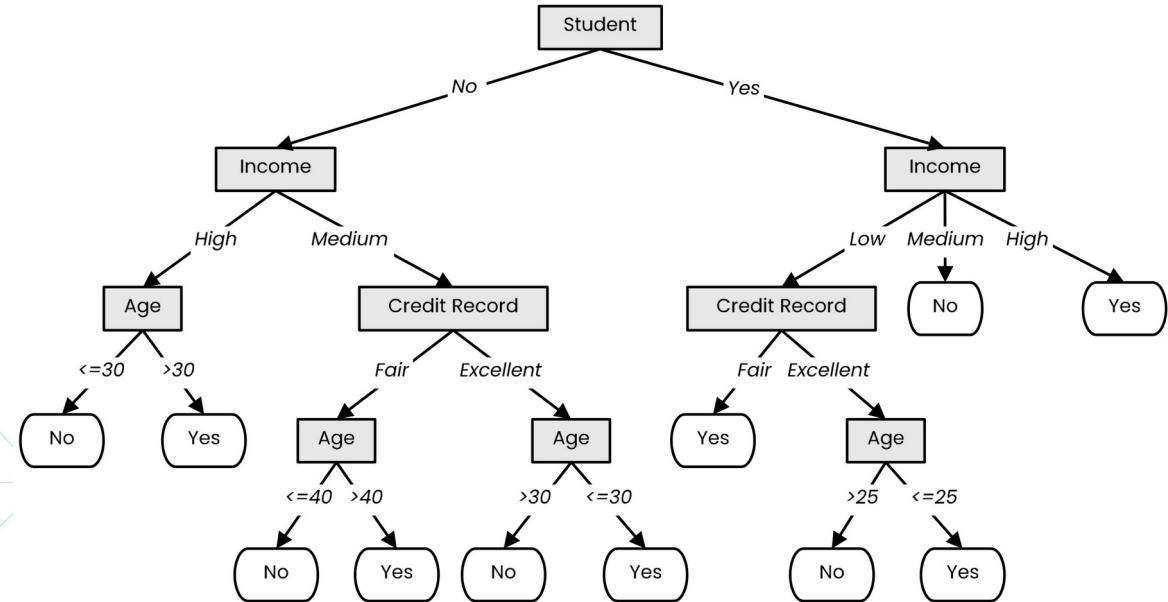
Definition

- 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



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

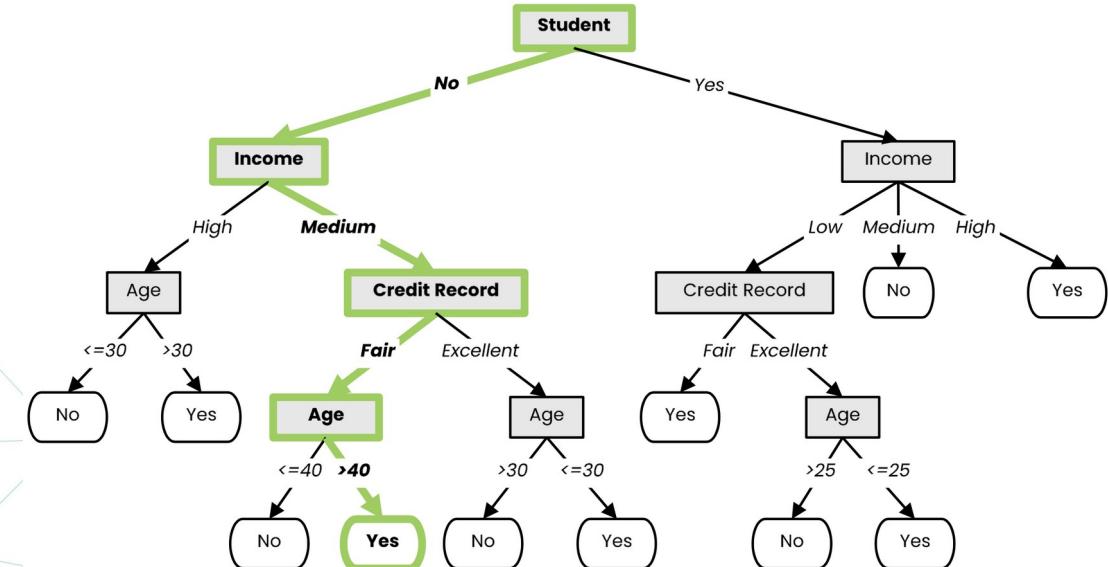
Who to loan?



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



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



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

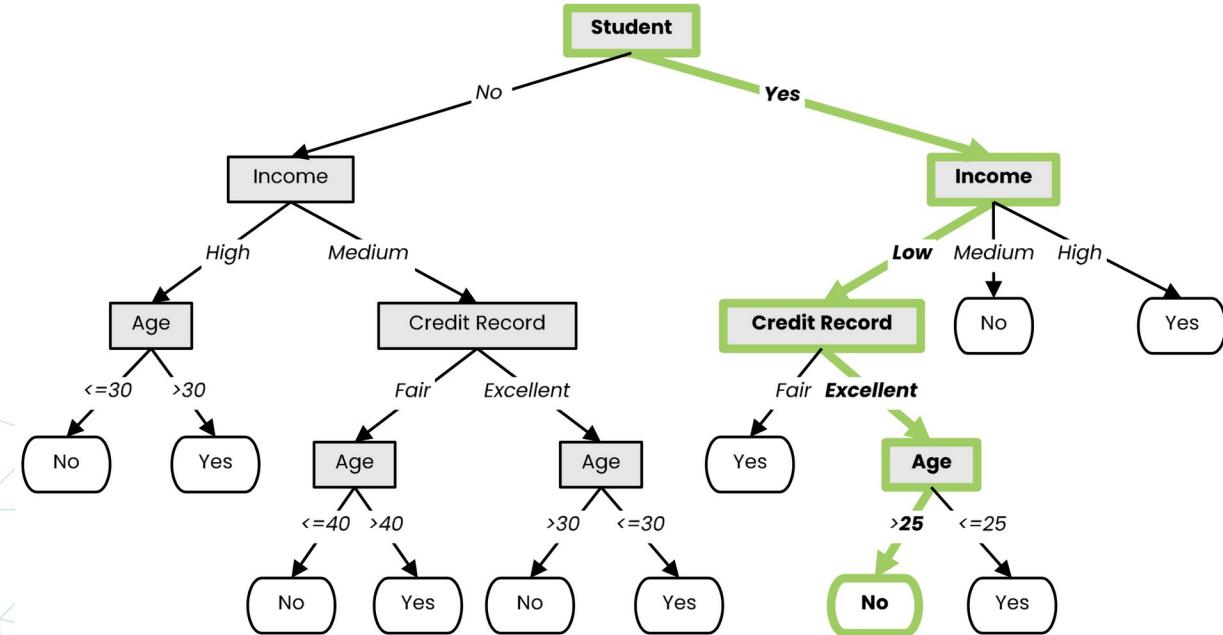
Who to loan?



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



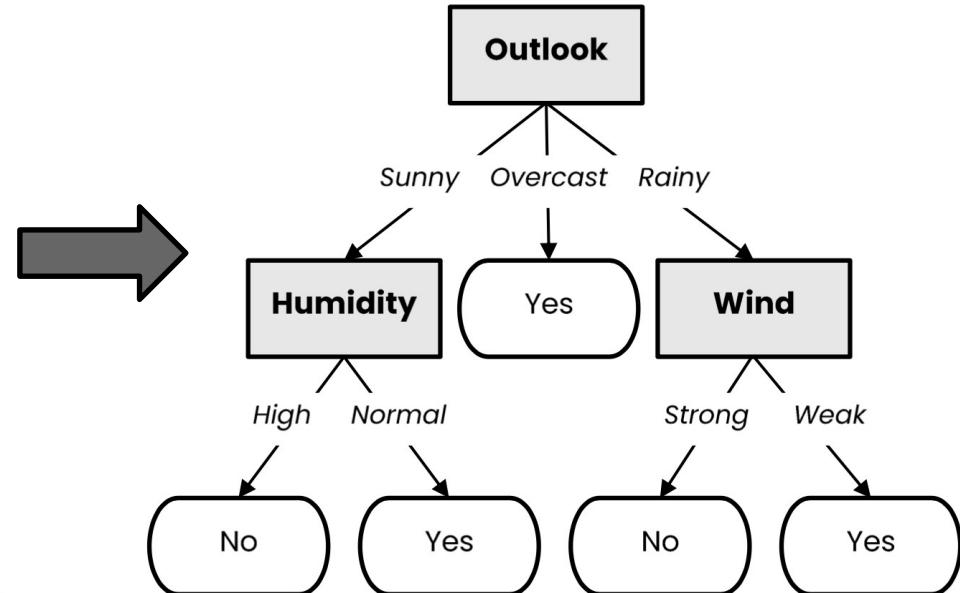
- Student
 - 27 years old
 - Low income
 - Excellent credit record
- No



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

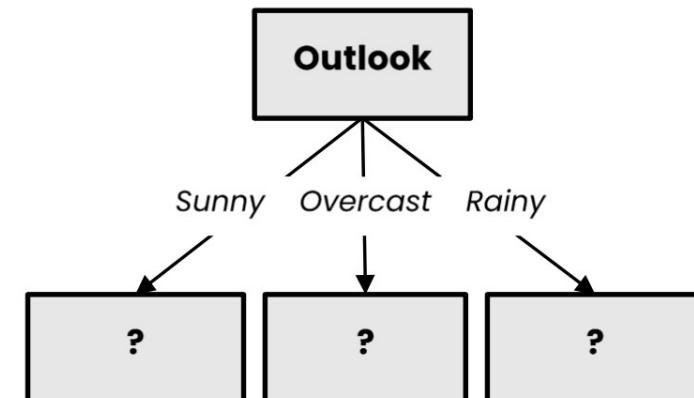
Temperature	Humidity	Wind	Play Tennis?
Hot	High	Weak	No
Hot	High	Strong	No
Mild	High	Weak	No
Cool	Normal	Weak	Yes
Mild	Normal	Strong	Yes

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

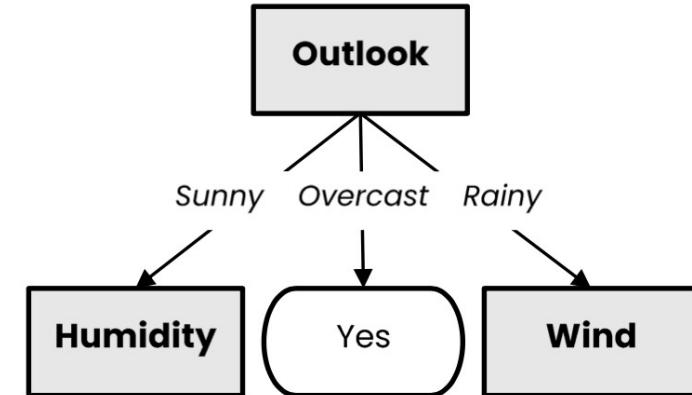
Temperature	Humidity	Wind	Play Tennis?
Hot	High	Weak	No
Hot	High	Strong	No
Mild	High	Weak	No
Cool	Normal	Weak	Yes
Mild	Normal	Strong	Yes

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		Humidity = Normal	
Temperature	Wind	Play Tennis?	Play Tennis?
Hot	Weak	No	Cool
Hot	Strong	No	Mild
Mild	Weak	No	Strong

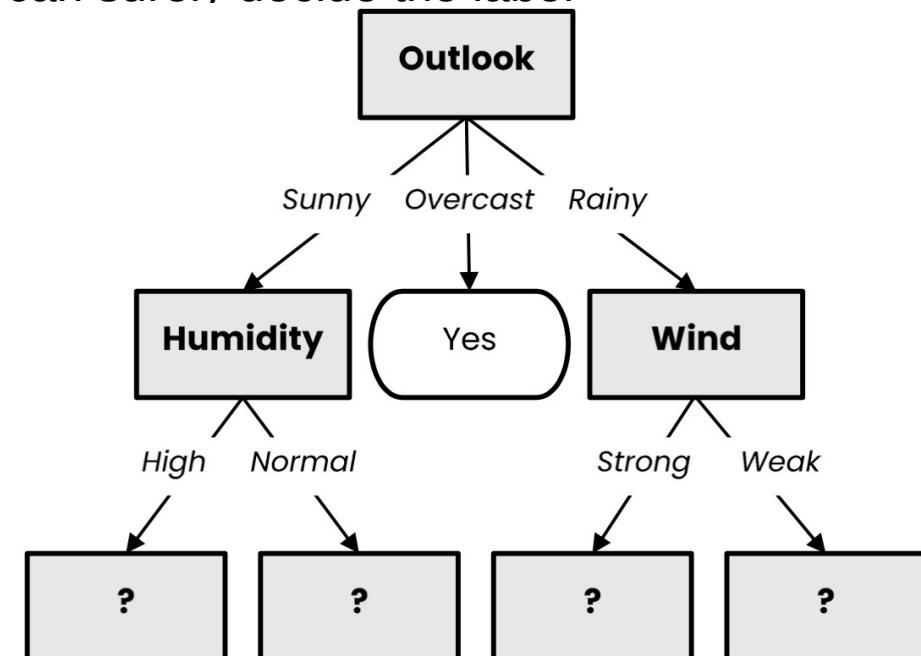
Humidity = Normal

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



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

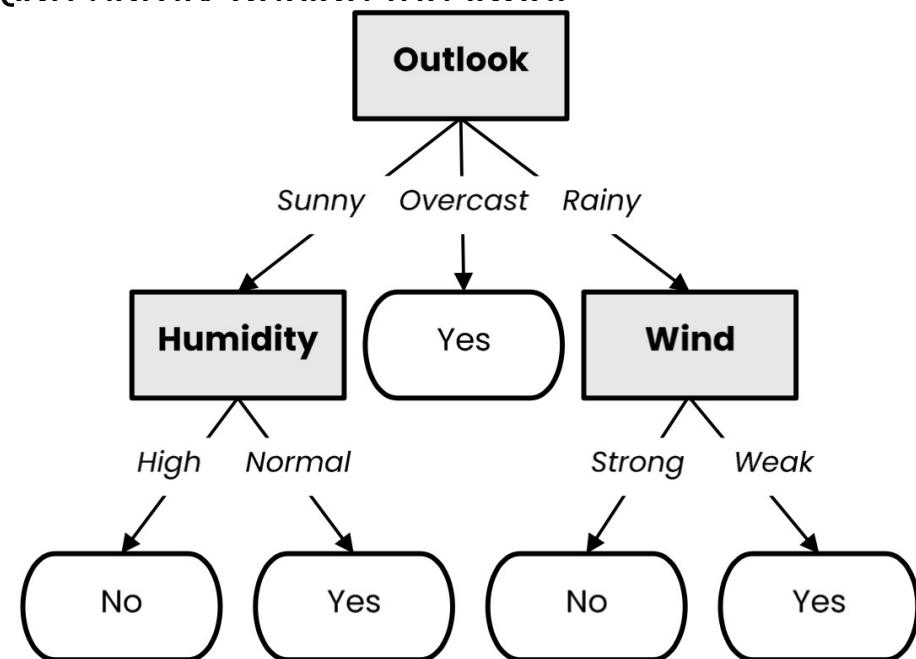
Humidity = High		
Temperature	Wind	Play Tennis?
Hot	Weak	No
Hot	Strong	No
Mild	Weak	No

Humidity = Normal		
Temperature	Wind	Play Tennis?
Cool	Weak	Yes
Mild	Strong	Yes

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

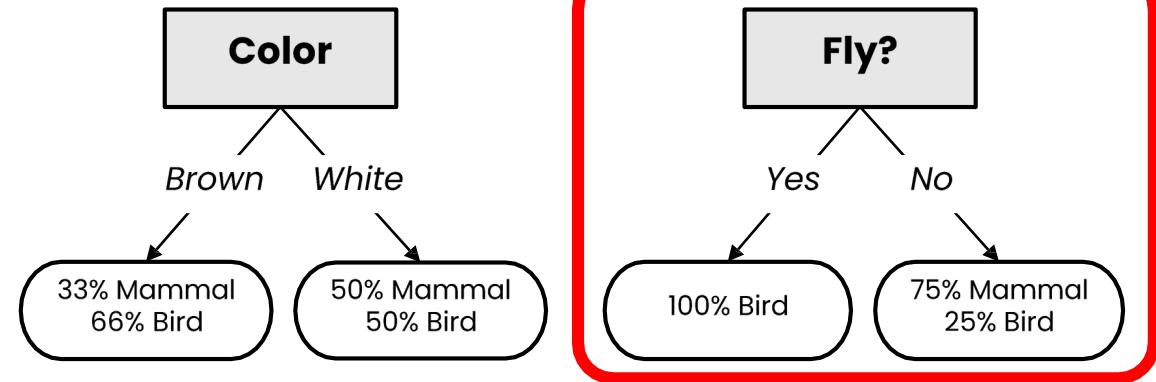
Wind = Strong		
Temperature	Humidity	Play Tennis?
Cool	Normal	No
Mild	High	No

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 splitting?
- 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)



رواد مصر الرقمية



وزارة الاتصالات
وتقنيات المعلومات

Information Gain

Entropy

- Entropy measures the degree of randomness in data

Low entropy



High entropy



- For a set of samples X with k classes:

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

where p_i is the proportion of elements of class i

- Lower entropy implies greater predictability!

Information Gain

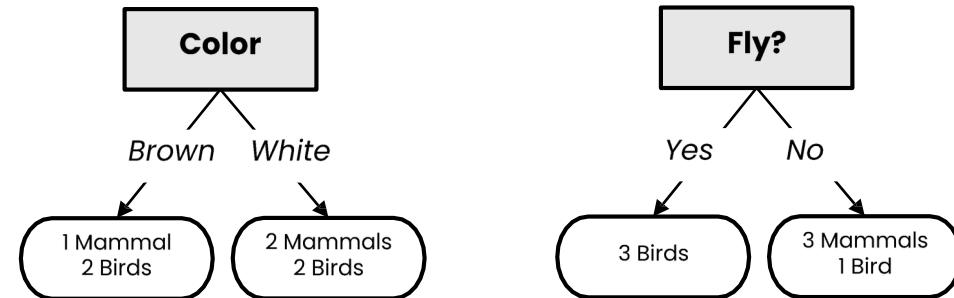
- The information gain of an attribute a is the expected reduction in entropy due to splitting on values of a :

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

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

Best attribute = highest information gain

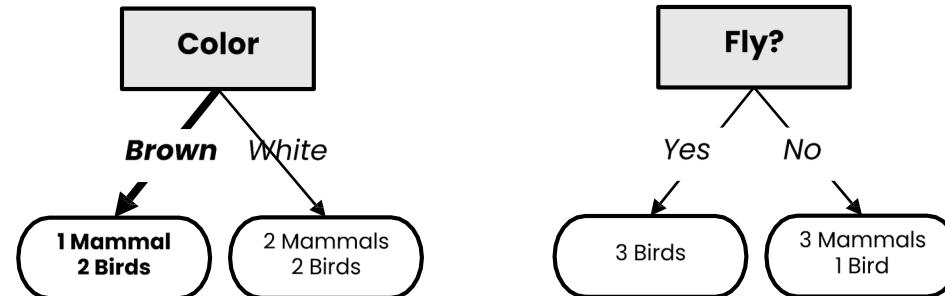
Does it fly?	Color	Class
No	Brown	Mammal
No	White	Mammal
Yes	Brown	Bird
Yes	White	Bird
No	White	Mammal
No	Brown	Bird
Yes	White	Bird



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

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

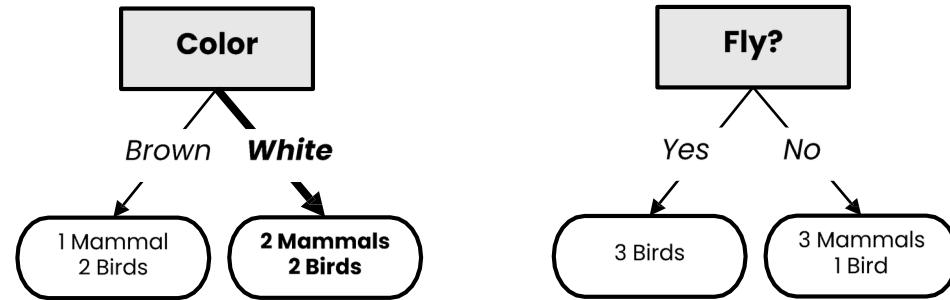


$$\text{entropy}(X) = - p_{\text{mammal}} \log_2 p_{\text{mammal}} - p_{\text{bird}} \log_2 p_{\text{bird}} = -\frac{3}{7} \log_2 \frac{3}{7} - \frac{4}{7} \log_2 \frac{4}{7} \approx 0.985$$

$$\text{entropy}(X_{\text{color=brown}}) = -\frac{1}{3} \log_2 \frac{1}{3} - \frac{2}{3} \log_2 \frac{2}{3} \approx 0.918$$

Best attribute = highest information gain

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

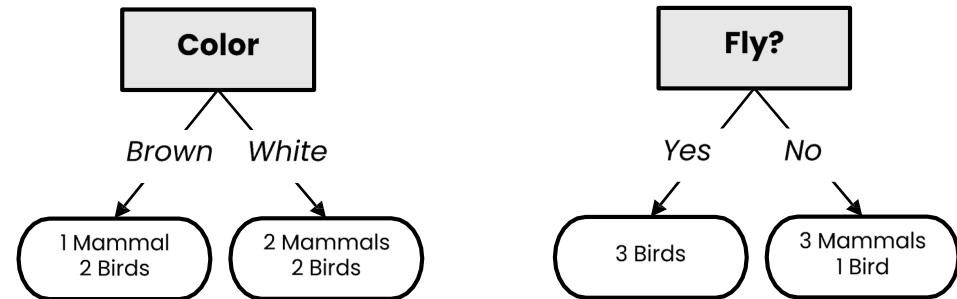


$$\text{entropy}(X) = - p_{\text{mammal}} \log_2 p_{\text{mammal}} - p_{\text{bird}} \log_2 p_{\text{bird}} = - \frac{3}{7} \log_2 \frac{3}{7} - \frac{4}{7} \log_2 \frac{4}{7} \approx 0.985$$

$$\text{entropy}(X_{\text{color=brown}}) = - \frac{1}{3} \log_2 \frac{1}{3} - \frac{2}{3} \log_2 \frac{2}{3} \approx 0.918 \quad \text{entropy}(X_{\text{color=white}}) = 1$$

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



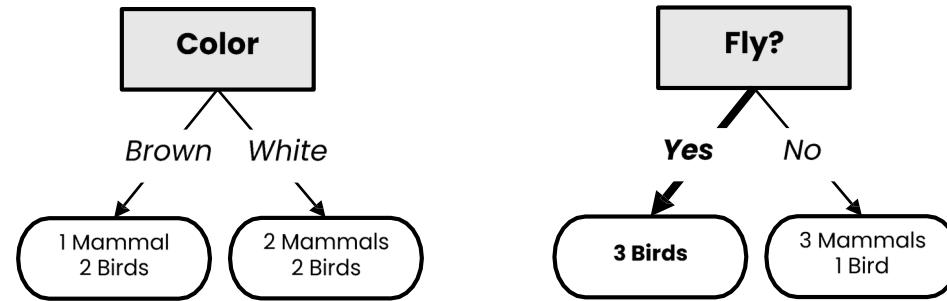
$$\text{entropy}(X) = - p_{\text{mammal}} \log_2 p_{\text{mammal}} - p_{\text{bird}} \log_2 p_{\text{bird}} = - \frac{3}{7} \log_2 \frac{3}{7} - \frac{4}{7} \log_2 \frac{4}{7} \approx 0.985$$

$$\text{entropy}(X_{\text{color=brown}}) = - \frac{1}{3} \log_2 \frac{1}{3} - \frac{2}{3} \log_2 \frac{2}{3} \approx 0.918 \quad \text{entropy}(X_{\text{color=white}}) = 1$$

$$\text{gain}(X, \text{color}) = 0.985 - \frac{3}{7} \cdot 0.918 - \frac{4}{7} \cdot 1 \approx 0.020$$

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



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

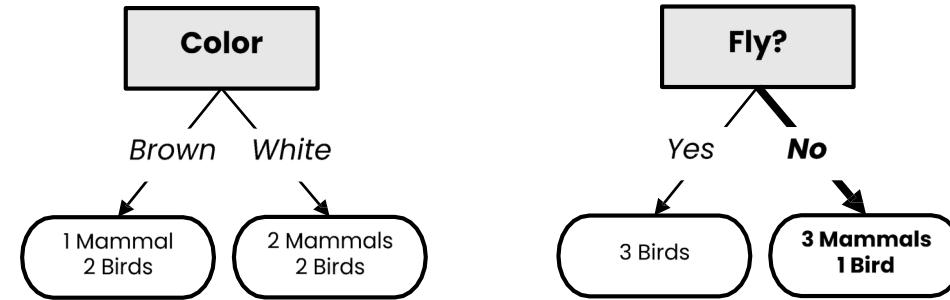
$$\text{entropy}(X_{\text{color=brown}}) = -\frac{1}{3} \log_2 \frac{1}{3} - \frac{2}{3} \log_2 \frac{2}{3} \approx 0.918 \quad \text{entropy}(X_{\text{color=white}}) = 1$$

$$\text{gain}(X, \text{color}) = 0.985 - \frac{3}{7} \cdot 0.918 - \frac{4}{7} \cdot 1 \approx 0.020$$

$$\text{entropy}(X_{\text{fly=yes}}) = 0$$

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



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

$$\text{entropy}(X_{\text{color}=brown}) = -\frac{1}{3} \log_2 \frac{1}{3} - \frac{2}{3} \log_2 \frac{2}{3} \approx 0.918 \quad \text{entropy}(X_{\text{color}=white}) = 1$$

$$\text{gain}(X, \text{color}) = 0.985 - \frac{3}{7} \cdot 0.918 - \frac{4}{7} \cdot 1 \approx 0.020$$

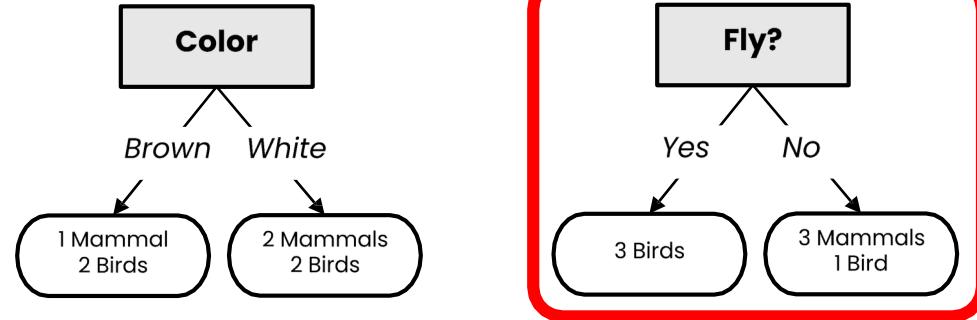
$$\text{entropy}(X_{\text{fly}=yes}) = 0$$

$$\text{entropy}(X_{\text{fly}=no}) = -\frac{3}{4} \log_2 \frac{3}{4} - \frac{1}{4} \log_2 \frac{1}{4} \approx 0.811$$

Best attribute = highest information gain

In practice, we compute $entropy(X)$ 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



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

$$entropy(X_{color=brown}) = -\frac{1}{3} \log_2 \frac{1}{3} - \frac{2}{3} \log_2 \frac{2}{3} \approx 0.918 \quad entropy(X_{color=white}) = 1$$

$$gain(X, color) = 0.985 - \frac{3}{7} \cdot 0.918 - \frac{4}{7} \cdot 1 \approx 0.020$$

$$entropy(X_{fly=yes}) = 0$$

$$entropy(X_{fly=no}) = -\frac{3}{4} \log_2 \frac{3}{4} - \frac{1}{4} \log_2 \frac{1}{4} \approx 0.811$$

$$gain(X, fly) = 0.985 - \frac{3}{7} \cdot 0 - \frac{4}{7} \cdot 0.811 \approx 0.521$$



رواد مصر الرقمية



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:

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

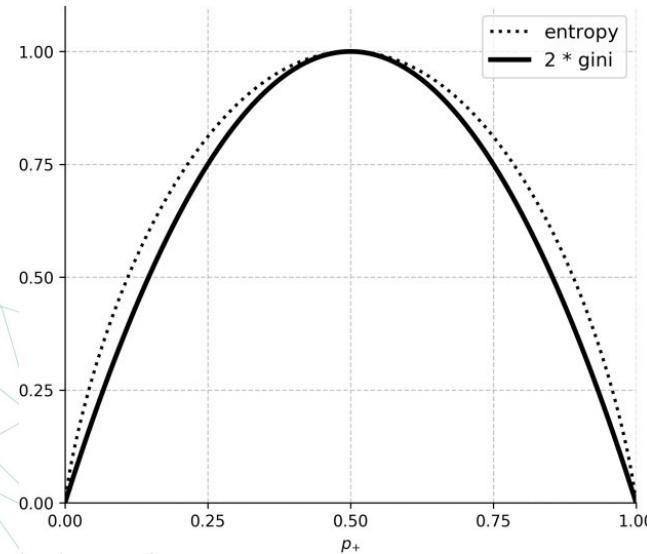
If two balanced classes 50% and 50%:
 $1 - (0.5^2 + 0.5^2) = 1 - (0.25+0.25) = 0.5$

where p_i is the proportion of elements of class i

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

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



Pruning

Pruning

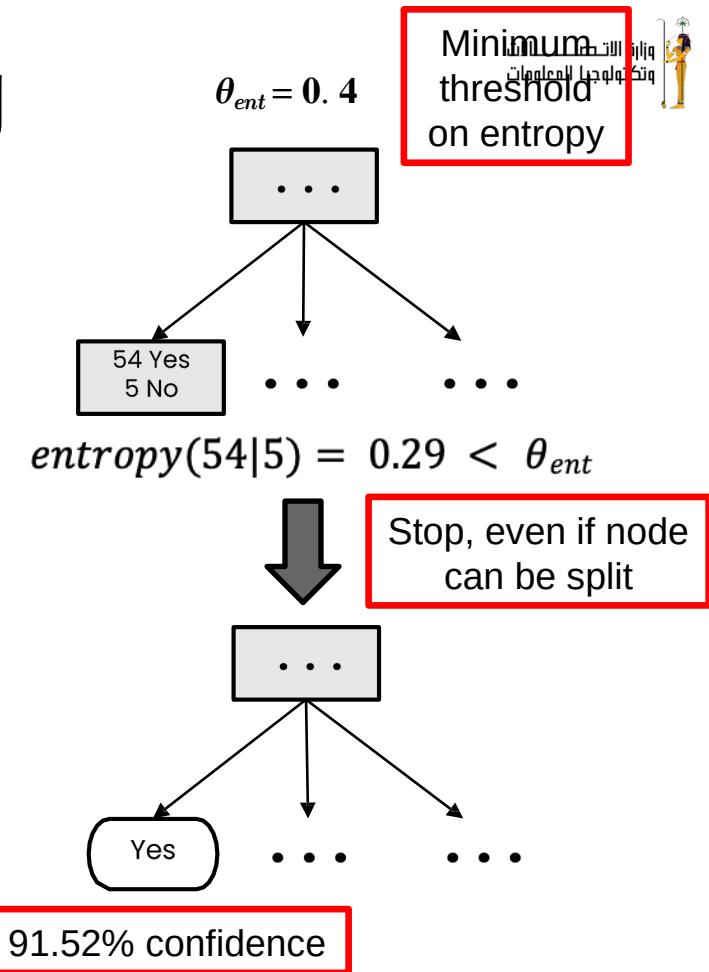
- Pruning is a technique that reduces the size of a decision tree by removing branches of the tree which provide little predictive power
- It is a **regularization** method that reduces the complexity of the final model, thus reducing overfitting
 - Decision trees are prone to overfitting!
- Pruning methods:
 - Pre-pruning: Stop the tree building algorithm before it fully classifies the data
 - Post-pruning: Build the complete tree, then replace some non-leaf nodes with leaf nodes if this improves validation error

Pre-pruning

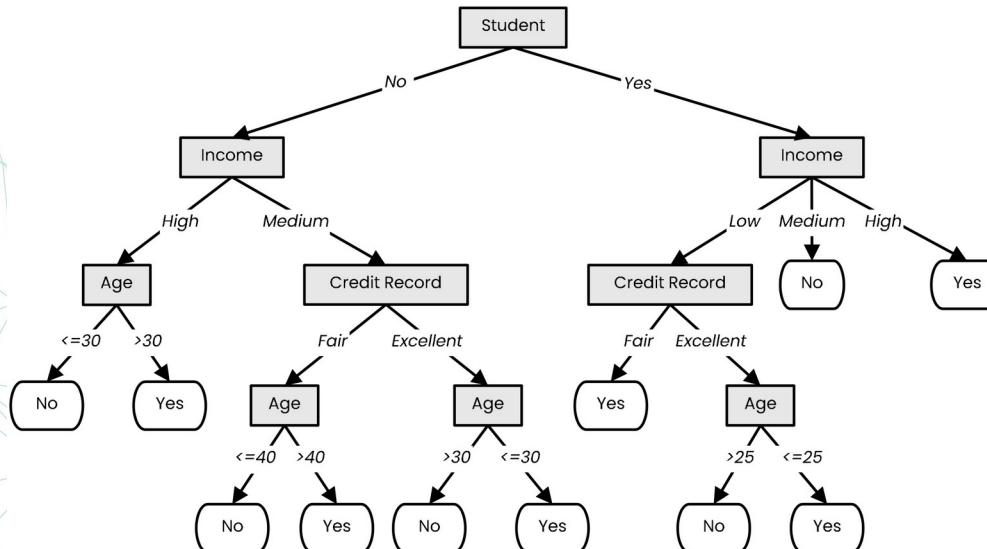
- Pre-pruning implies early stopping:
 - If some condition is met, the current node will not be split, even if it is not 100% pure
 - It will become a leaf node with the label of the majority class in the current set

(the class distribution could be used as prediction confidence)

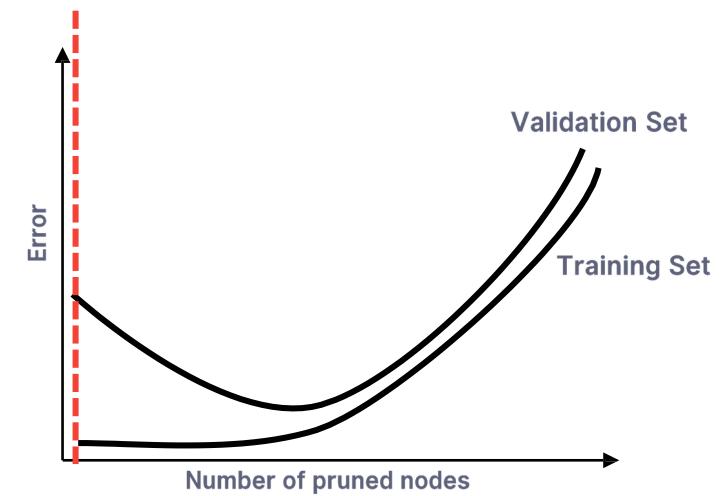
- Common stopping criteria include setting a threshold on:
 - Entropy (or Gini Impurity) of the current set
 - Number of samples in the current set
 - Gain of the best-splitting attribute
 - Depth of the tree



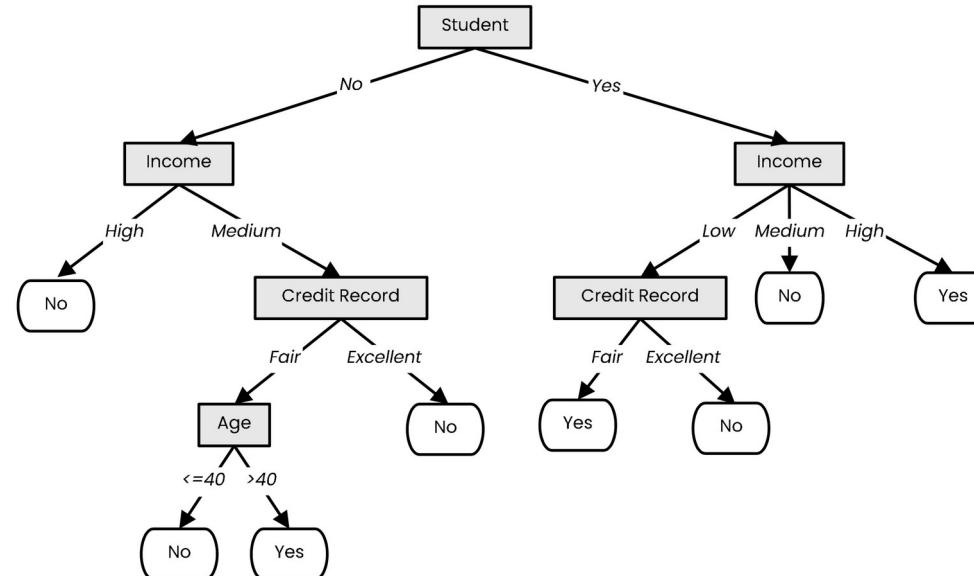
Post-pruning



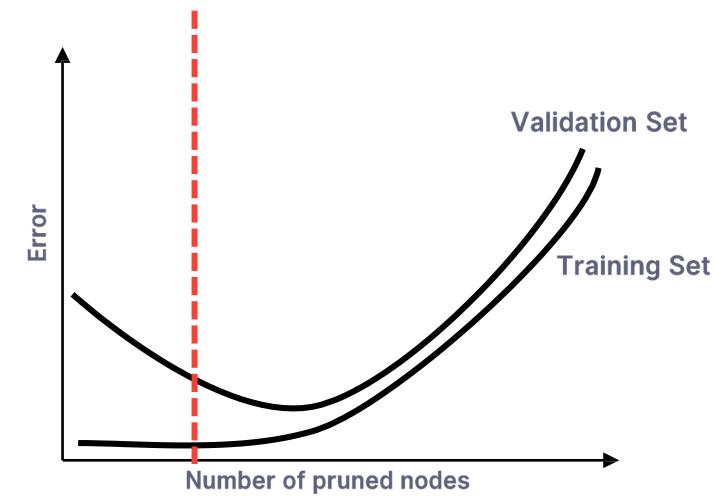
- Prune nodes in a bottom-up manner, if it decreases validation error



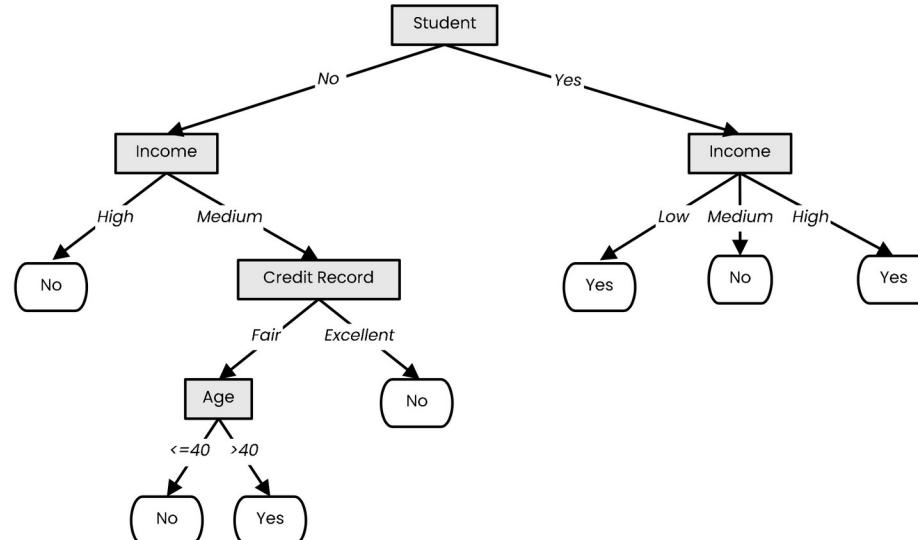
Post-pruning



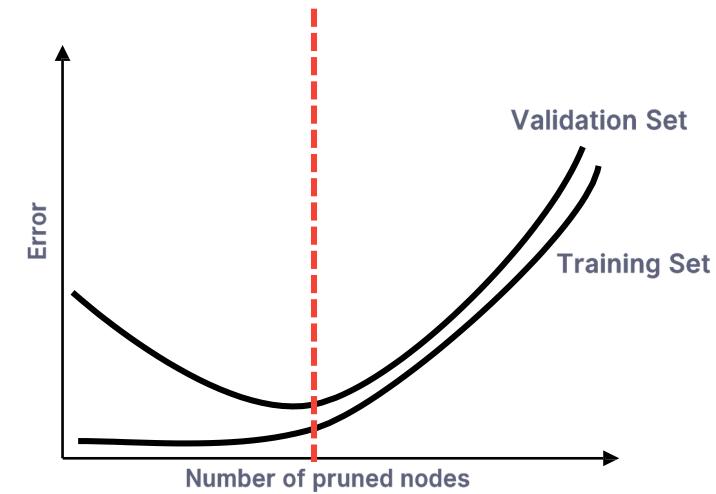
- Prune nodes in a bottom-up manner, if it decreases validation error



Post-pruning



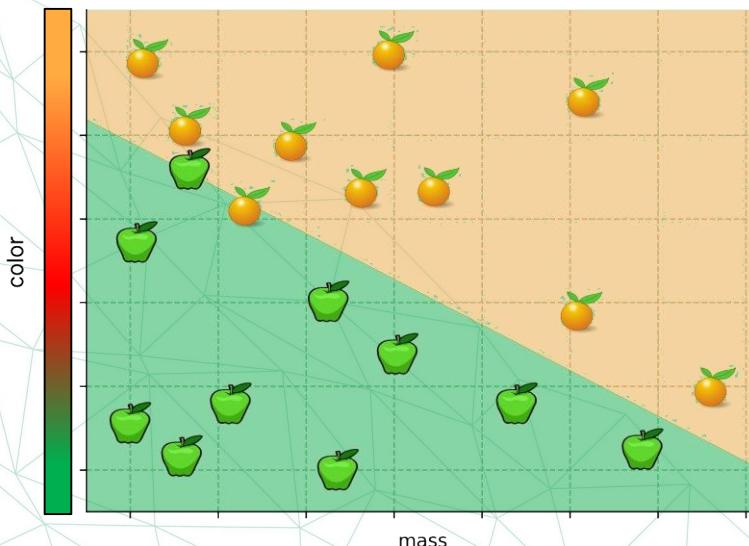
- Prune nodes in a bottom-up manner, if it decreases validation error



Decision Boundaries

- Decision trees produce non-linear decision boundaries

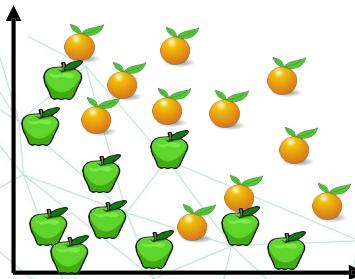
Logistic Regression



Decision Tree

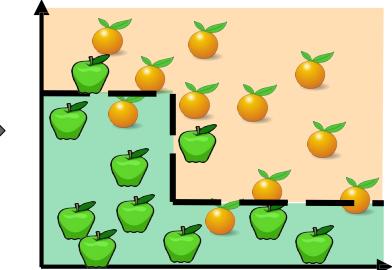


Decision Trees: Training and Inference

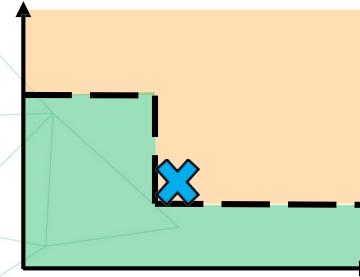
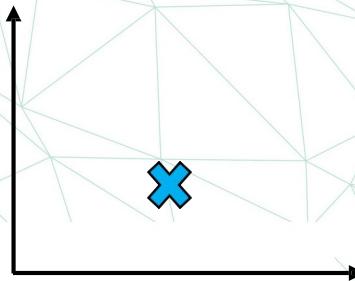


Training

Decision Tree
Learning Algorithm



Inference



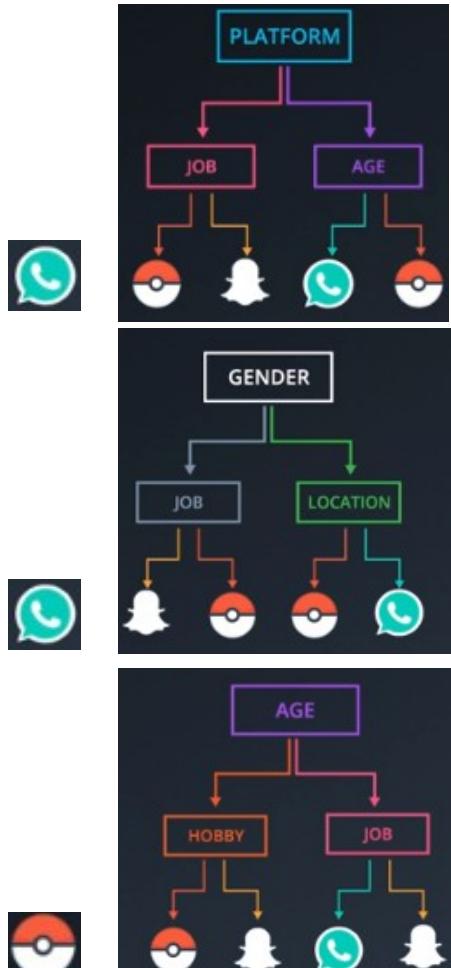
Random Forests

(Ensemble learning with decision trees)

Gender	Age	Location	Platform	Job	Hobby	App
F	30	US	IOS	School	Games	Whatsapp
F	11	France	Android	Work	Tennis	Pokemon Go
M	16	Chile	IOS	Temp	Tennis	Snapchat
F	15	China	IOS	Retired	Chess	Whatsapp
M	25	Us	Android	School	Games	Snapchat
M	32	Us	IOS	School	Tennis	Whatsapp
F	40	Egypt	Android	Work	Chess	Snapchat
M	12	France	Android	Temp	Tennis	Whatsapp
M	14	Australia	Android	School	chess	Pokemon Go



Random Forests



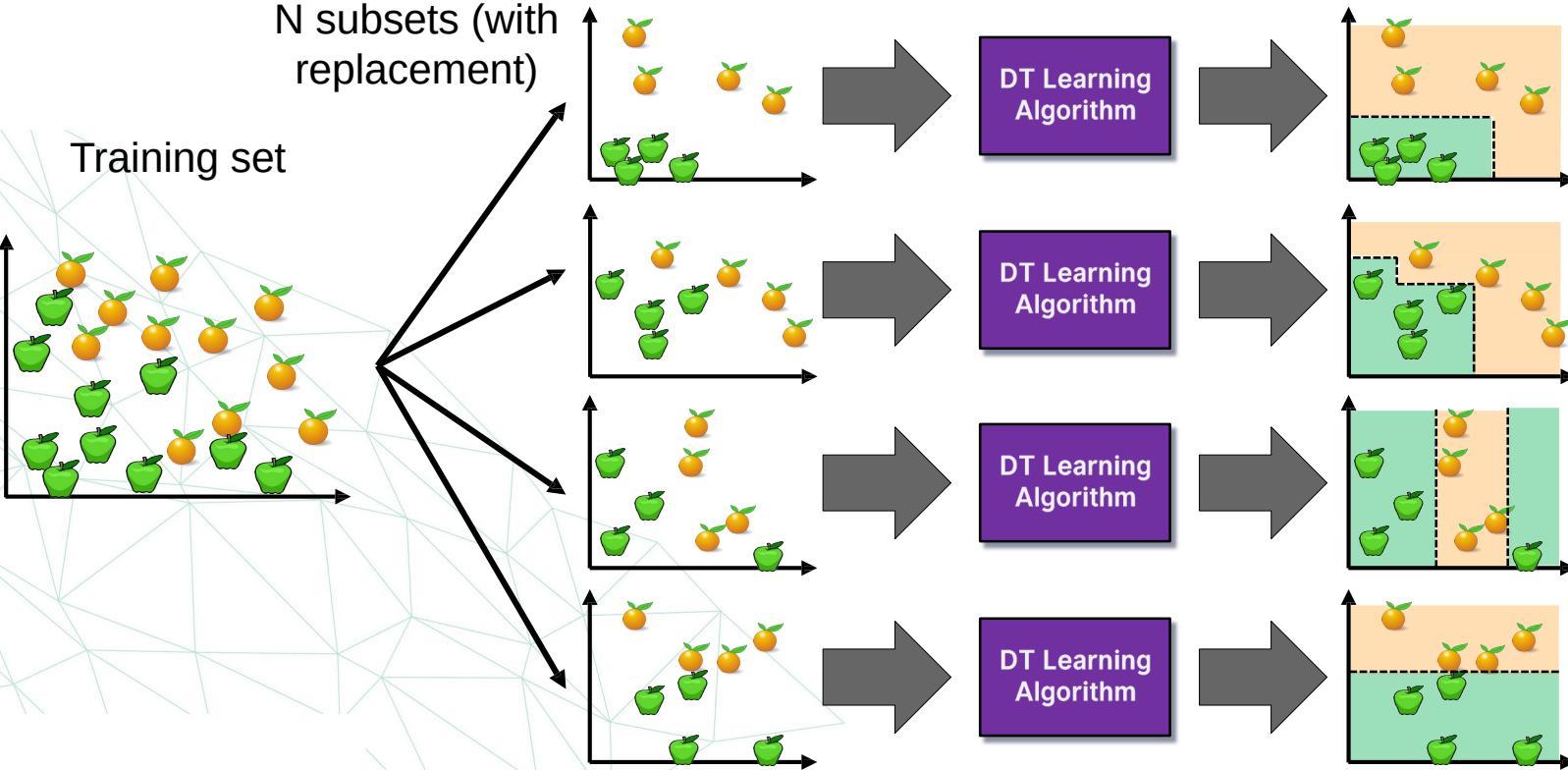
Random Forests

- Random Forests:
 - Instead of building a single decision tree and use it to make predictions, build many slightly different trees and combine their predictions
- We have a single data set, so how do we obtain slightly different trees?
 1. Bagging (**Bootstrap Aggregating**):
 - Take random subsets of data points from the training set to create N smaller data sets
 - Fit a decision tree on each subset
 2. Random Subspace Method (also known as Feature Bagging):
 - Fit N different decision trees by constraining each one to operate on a random subset of features

Bagging at training time

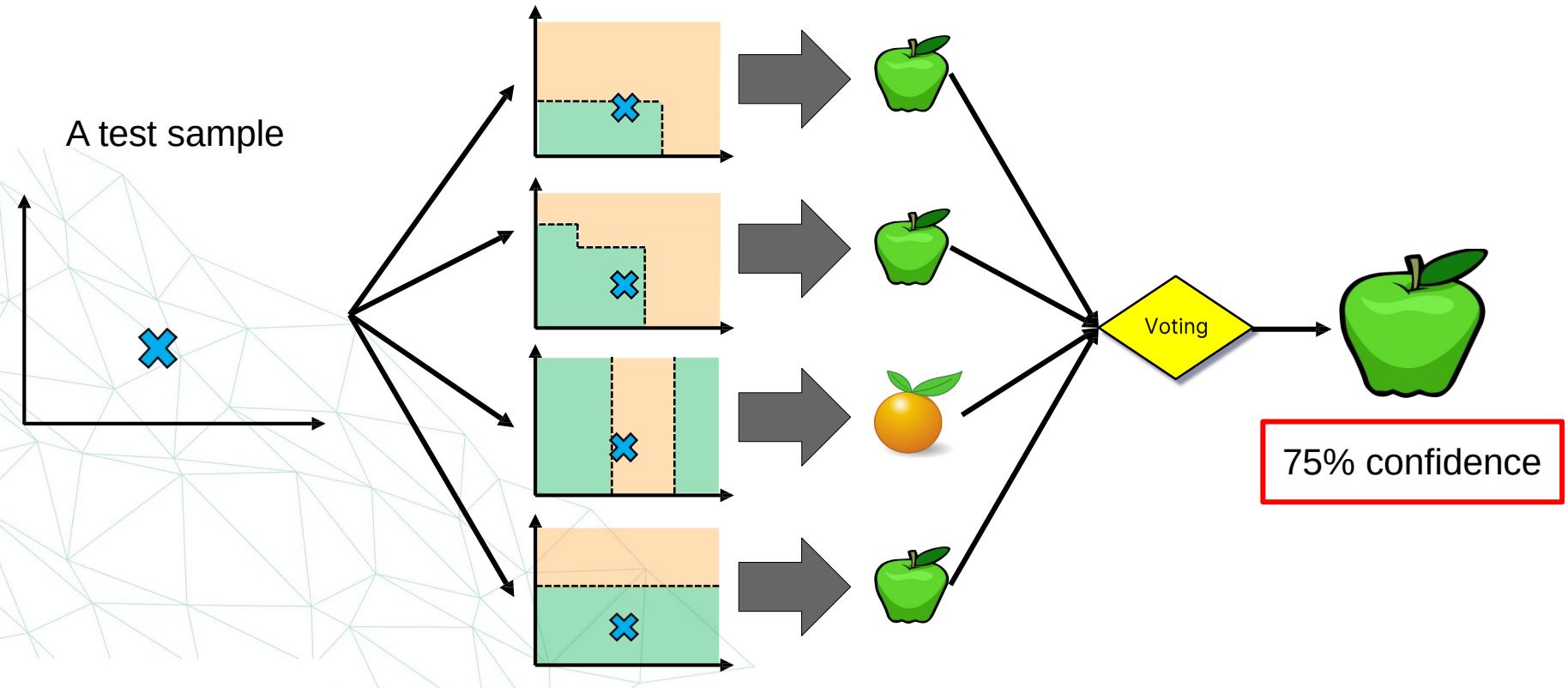
Training set

N subsets (with replacement)



Bagging at inference time

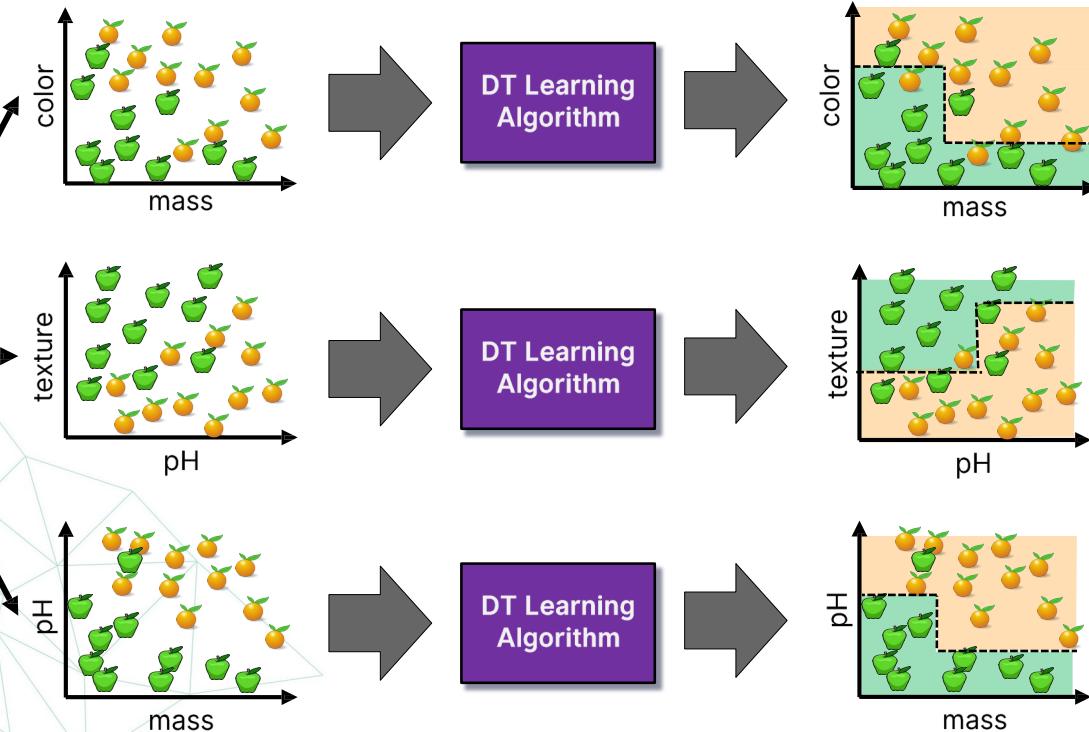
A test sample



Random Subspace Method at training time

Training data

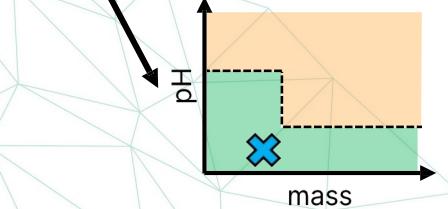
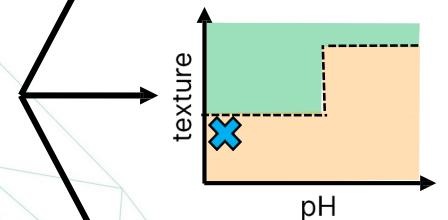
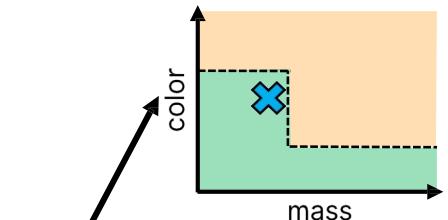
Mass (g)	Color	Texture	pH	Label
84	Green	Smooth	3.5	Apple
121	Orange	Rough	3.9	Orange
85	Red	Smooth	3.3	Apple
101	Orange	Smooth	3.7	Orange
111	Green	Rough	3.5	Apple
...				
117	Red	Rough	3.4	Orange



Random Subspace Method at inference time

A test sample

87 Red Smooth 3.1



Voting

66% confidence



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



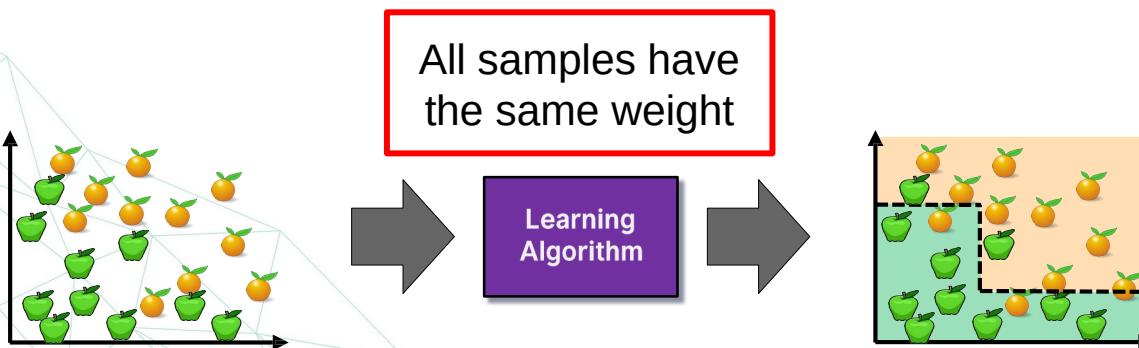
Bagging +
Random Subspace Method +
Decision Tree Learning Algorithm



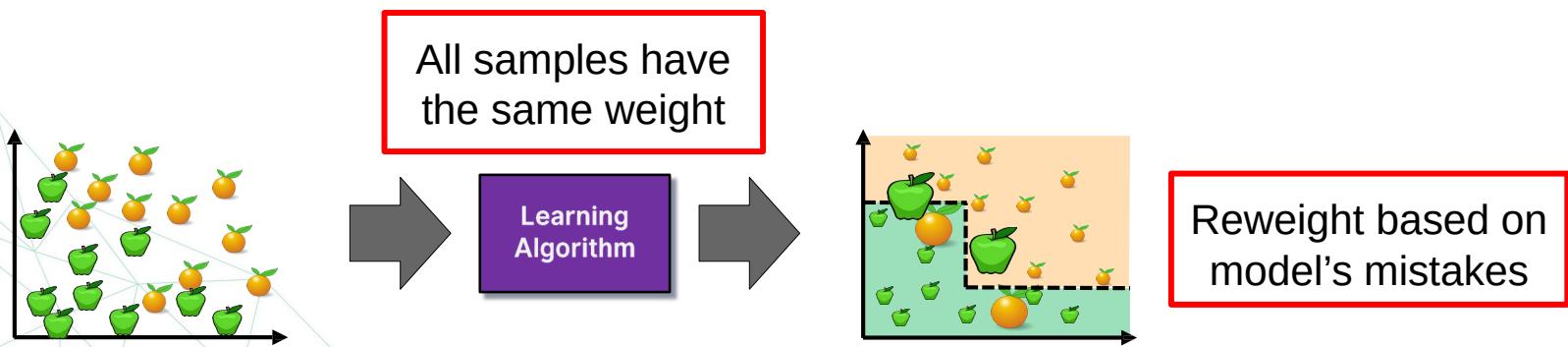
Ensemble Learning

- Ensemble Learning:
 - Method that combines multiple learning algorithms to obtain performance improvements over its components
- **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

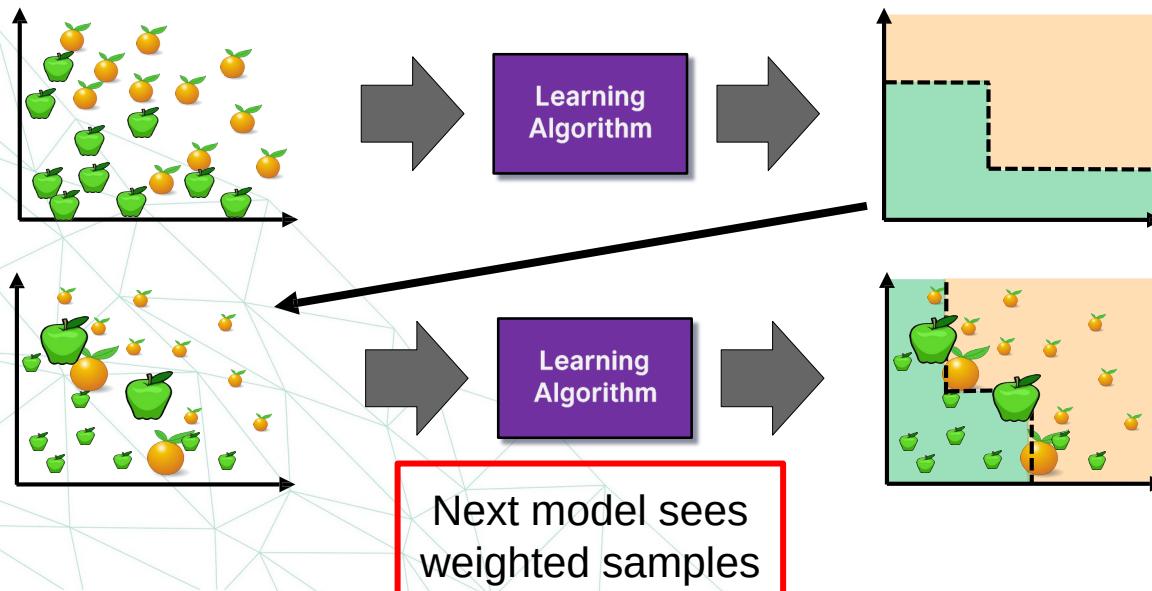
Boosting



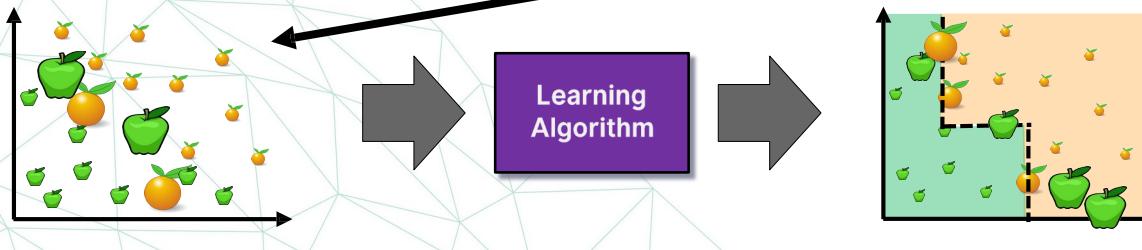
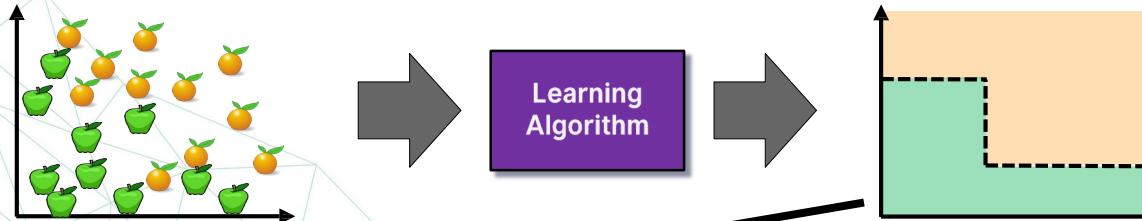
Boosting



Boosting

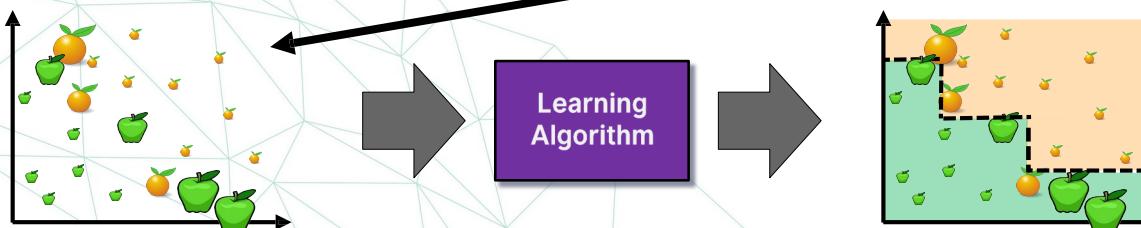
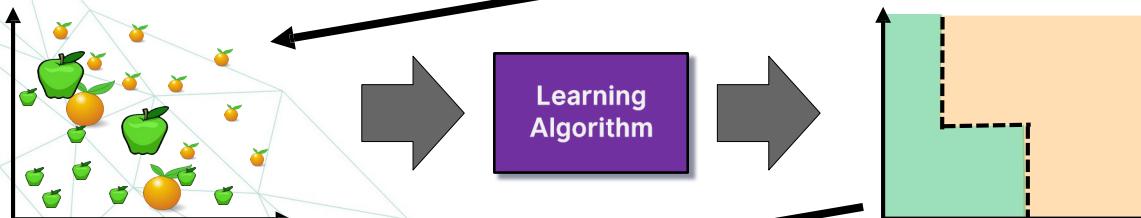
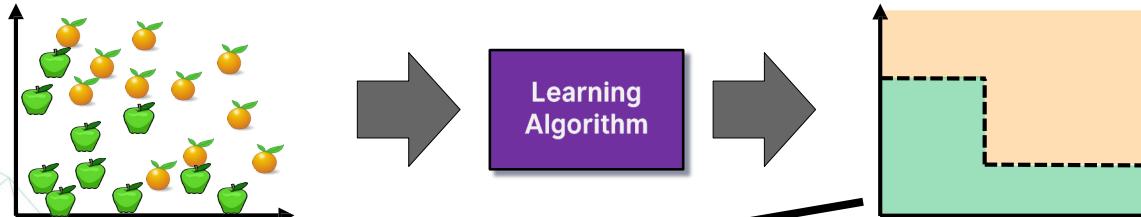


Boosting

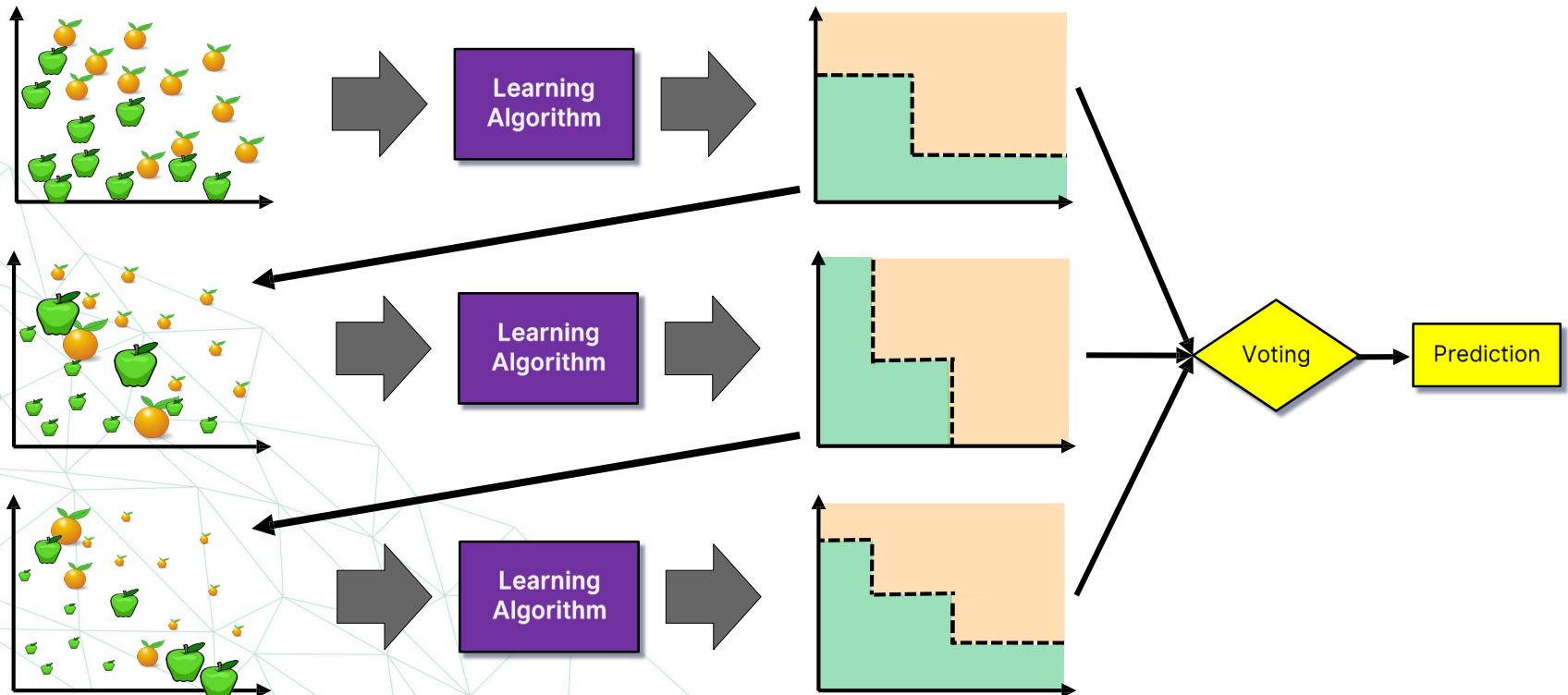


Reweight based
on current
model's mistakes

Boosting



Boosting



Summary

- Ensemble Learning methods combine multiple learning algorithms to obtain performance improvements over its components
- 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)
- **Random Forests** are an ensemble learning method that employ decision tree learning to build multiple trees through **bagging** and **random subspace method**.
 - They rectify the overfitting problem of decision trees!

Decision Trees and Random Forest (Python)

```
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier

clf = DecisionTreeClassifier(criterion = "entropy", min_samples_leaf = 3)
# Lots of parameters: criterion = "gini" / "entropy";
#                         max_depth;
#                         min_impurity_split;

clf.fit(X, y) # It can only handle numerical attributes!
# Categorical attributes need to be encoded, see LabelEncoder and OneHotEncoder

clf.predict([x]) # Predict class for x

clf.feature_importances_ # Importance of each feature
clf.tree_ # The underlying tree object

clf = RandomForestClassifier(n_estimators = 20) # Random Forest with 20 trees
```