Classification Supervised Learning (Part1)

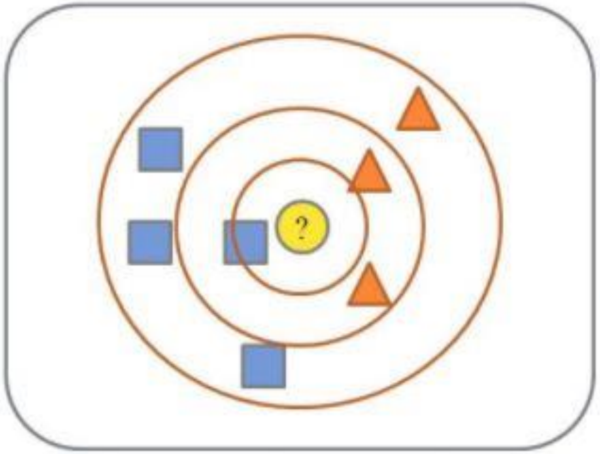
Outline

- K-Nearest Neighbor
- Decision Tree
  - CART
  - ID3

K-Nearest Neighbor

What’s K-Nearest Neighbor

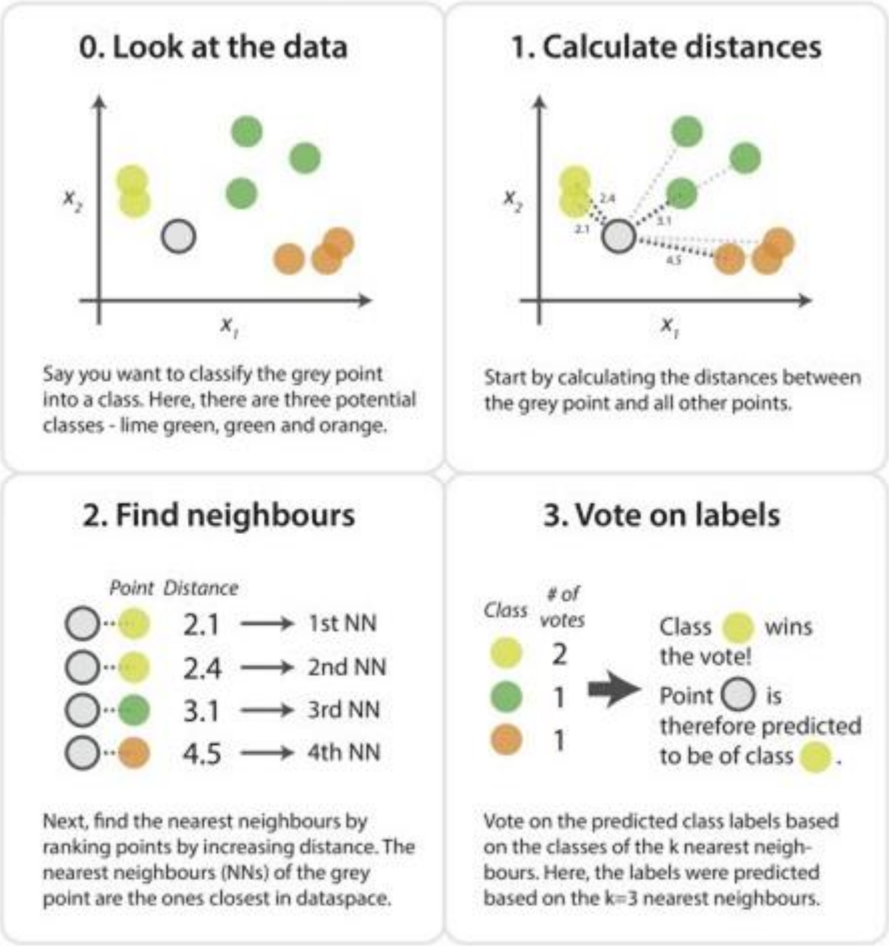
- A non-parametric method used for classification and regression
- Also called kNN
  - “k” mean how many neighbors should be considered to help classification/regression



- $k = 1$:
  - Belongs to square class
- $k = 3$:
  - Belongs to triangle class
- $k = 7$:
  - Belongs to square class

kNN intuitive concept

K-Nearest Neighbor



0. Look at the data

Say you want to classify the grey point into a class. Here, there are three potential classes - lime green, green and orange.

1. Calculate distances

Start by calculating the distances between the grey point and all other points.

2. Find neighbours

Point	*Distance*		
○··🟡	2.1	⟶	1st NN
○··🟡	2.4	⟶	2nd NN
○··🟢	3.1	⟶	3rd NN
○··🟠	4.5	⟶	4th NN

Next, find the nearest neighbours by ranking points by increasing distance. The nearest neighbours (NNs) of the grey point are the ones closest in dataspace.

3. Vote on labels

Class	*# of votes*
🟡	2
🟢	1
🟠	1

➡

Class 🟡 wins the vote!

Point ○ is therefore predicted to be of class 🟡.

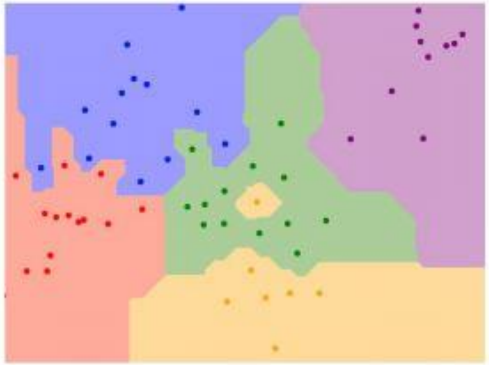
Vote on the predicted class labels based on the classes of the k nearest neighbours. Here, the labels were predicted based on the k=3 nearest neighbours.

How to Define Distance

- **L1 distance (Manhattan distance)**
- **L2 distance (Euclidean distance)**

L1 (Manhattan) distance

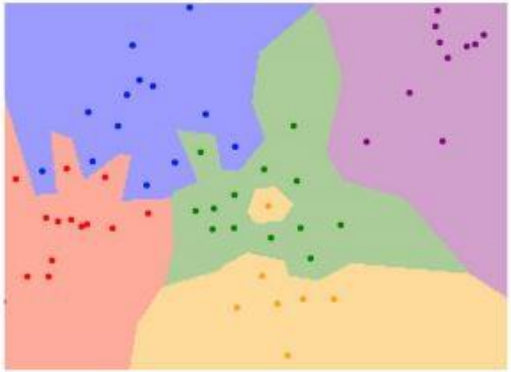
$$d_1(I_1, I_2) = \sum_p |I_1^p - I_2^p|$$



K = 1

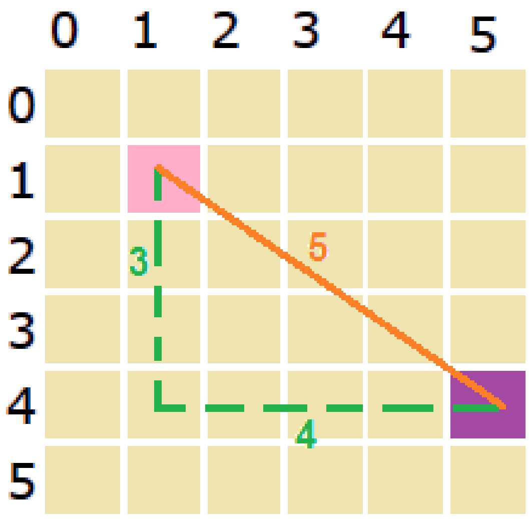
L2 (Euclidean) distance

$$d_2(I_1, I_2) = \sqrt{\sum_p (I_1^p - I_2^p)^2}$$



K = 1

Example



Point P1 = (1,1)

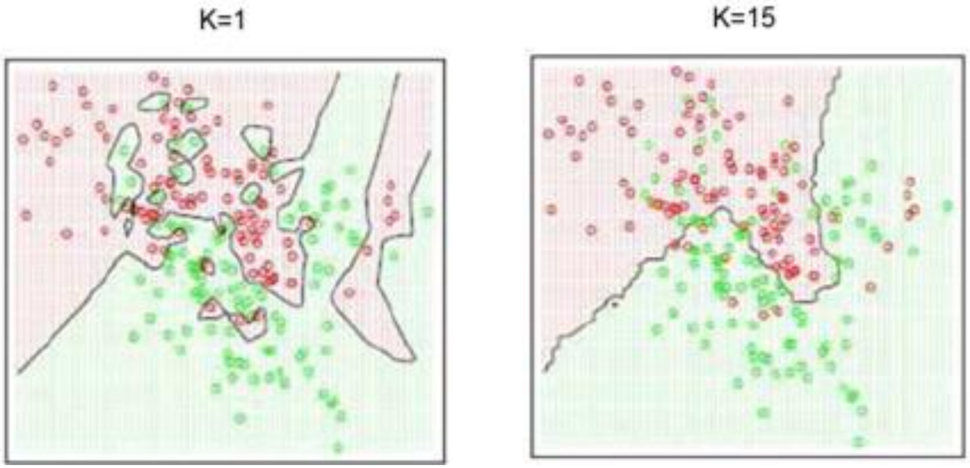
Point P2 = (5,4)

Euclidean distance = $\sqrt{(5-1)^2 + (4-1)^2}$ = 5

Manhattan distance = $|5-1| + |4-1|$ = 7

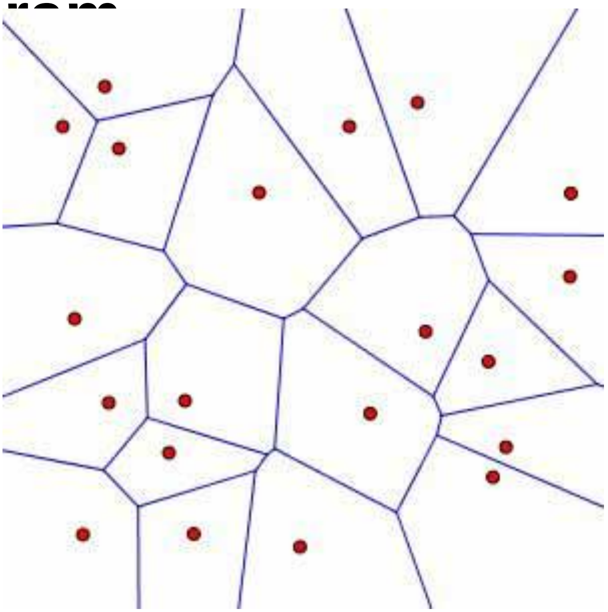
How to choose K?

- K is small
  - sensitive to noise points
- K is large
  - neighborhood may include points from other classes
  - smoother boundary
  - If too large, machine always predict majority class



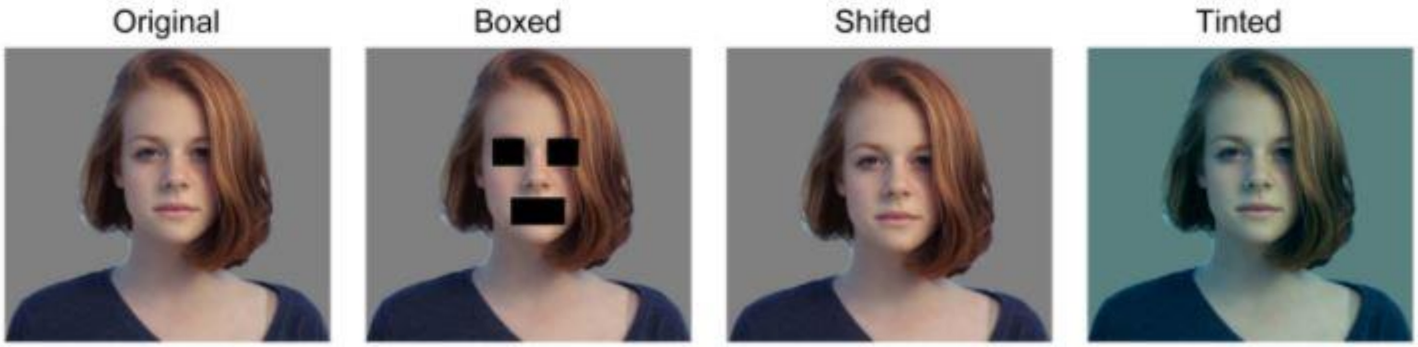
- http://vision.stanford.edu/teaching/cs231n-demos/knn/

- **1-NN**
  - **Voronoi Diagram**



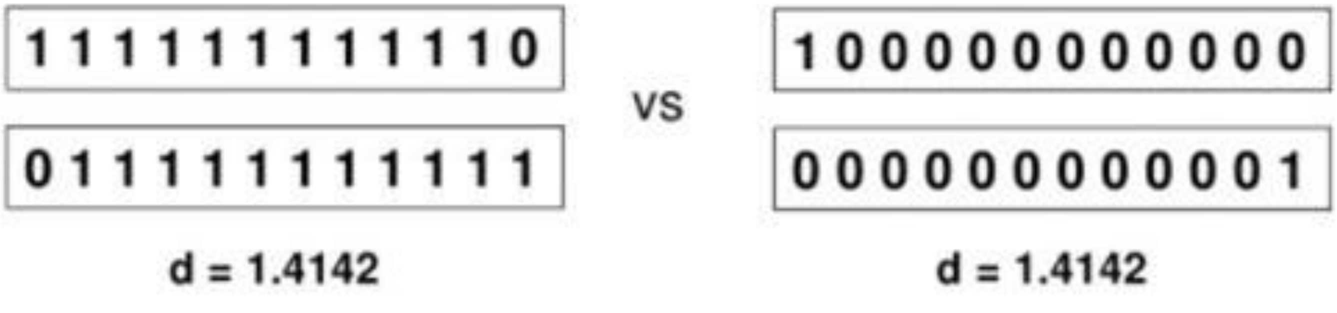
K-Nearest Neighbor

Don’t use kNN on images
(Distance between pixels are meaningless)



All 3 images have same L2 distance to the one on the left!

Problem in L2 Distance



1 1 1 1 1 1 1 1 1 1 1 0

0 1 1 1 1 1 1 1 1 1 1 1

d = 1.4142

vs

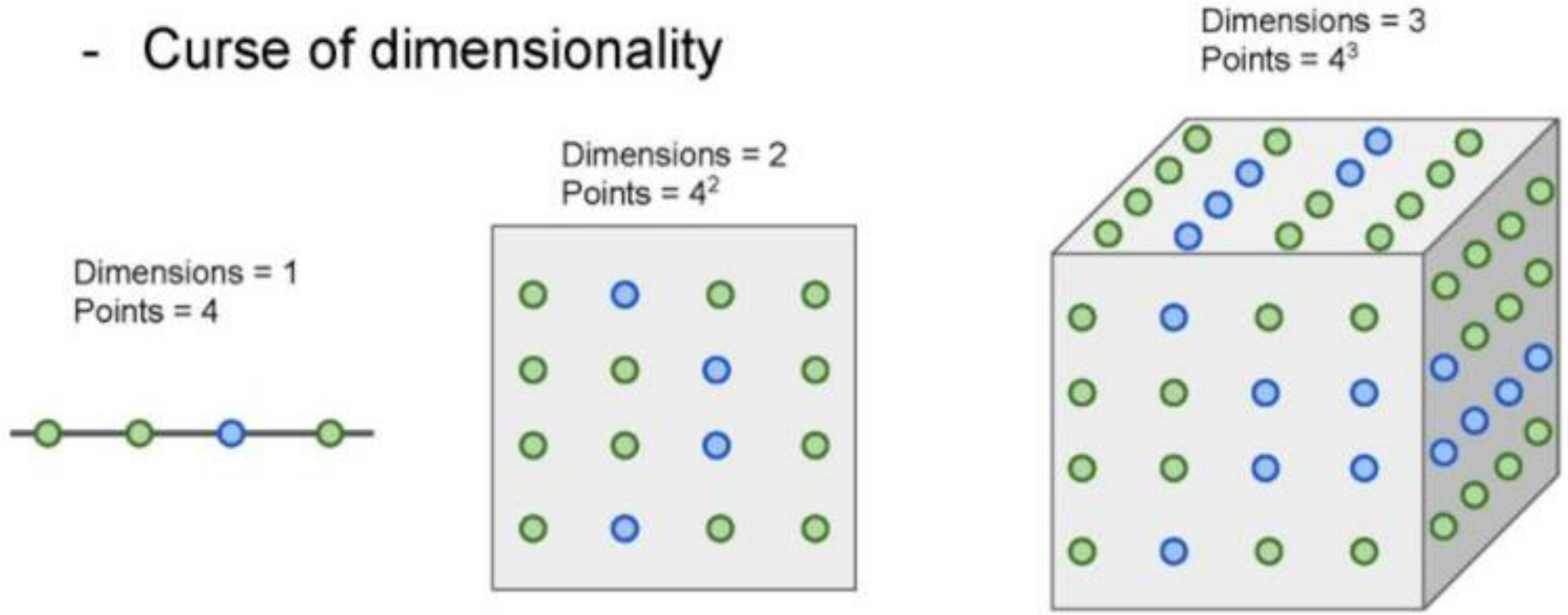
1 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 1

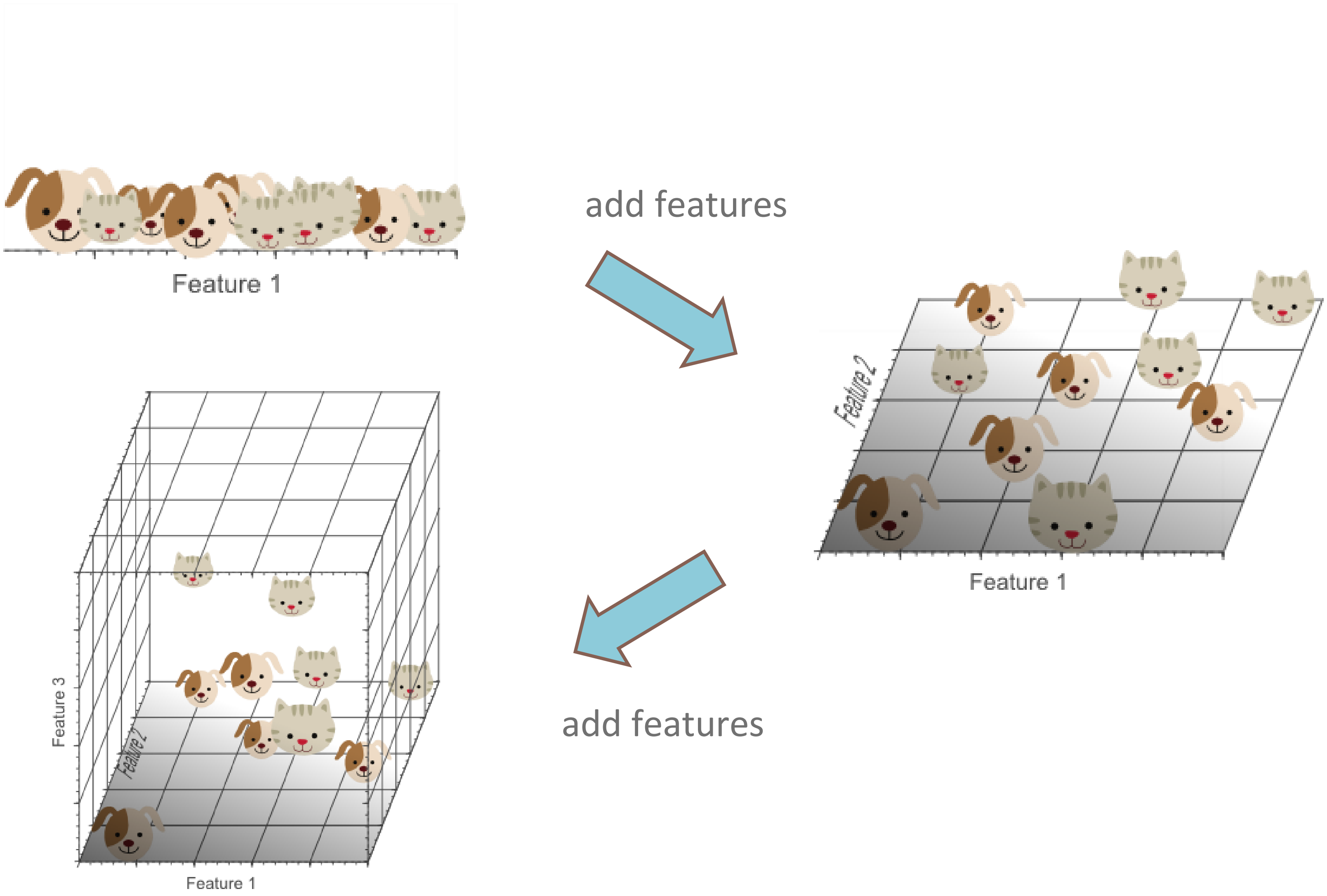
d = 1.4142

counter-intuitive results

- Curse of dimensionality



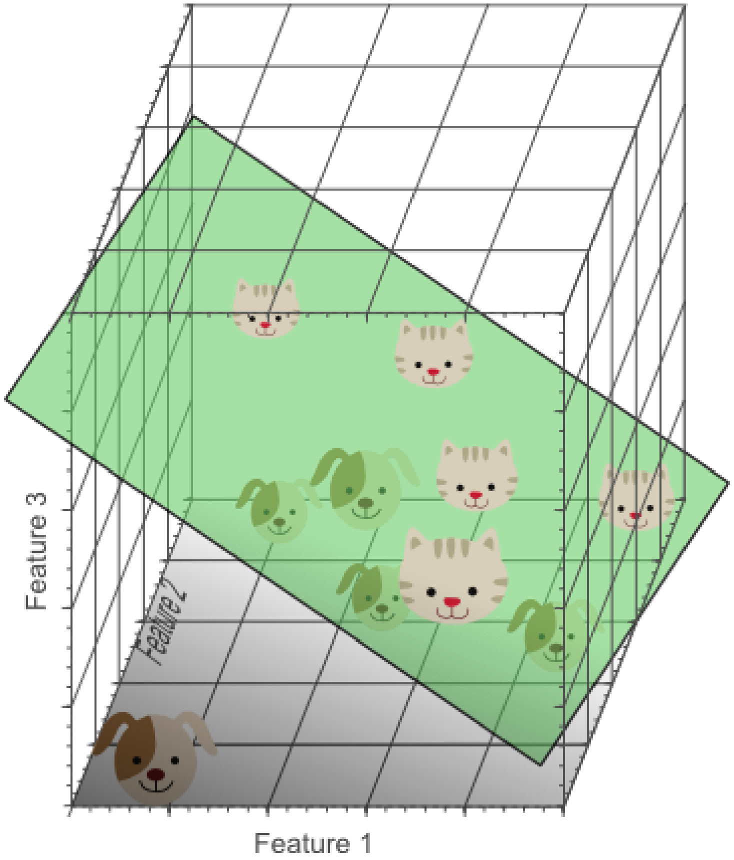
Curse of dimensionality



add features

add features

Curse of dimensionality



Linear separable in high dimensionality

Curse of dimensionality

- **Increase dimensionality may obtain perfect classfication**
- **However, extend too many dimensionality(features) lead to overfitting**

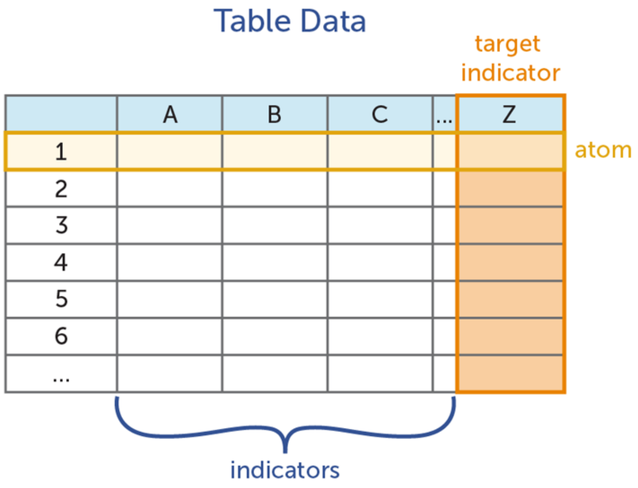
Example and Practice

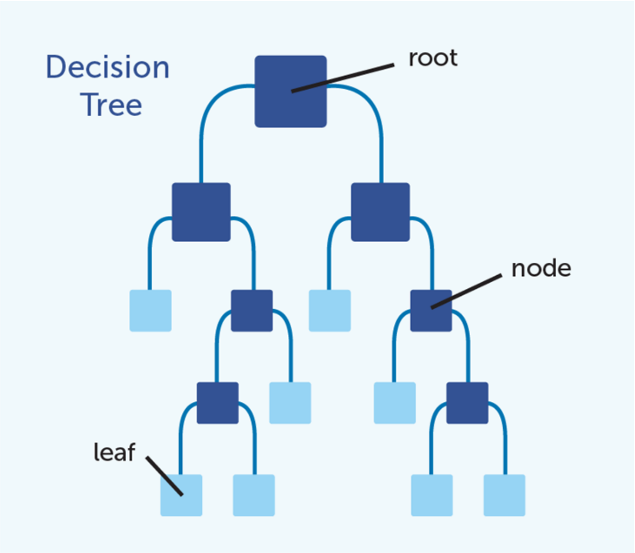
- **Example**
  - **KNN**
    - example/supervised learning
- **Practice**
  - **Try to use knn to predict different varieties of wheat**
    - dataset/seeds_dataset.csv
    - practice/supervised learning
  - **More information about the dataset**
    - https://archive.ics.uci.edu/ml/datasets/seeds#

Decision Tree

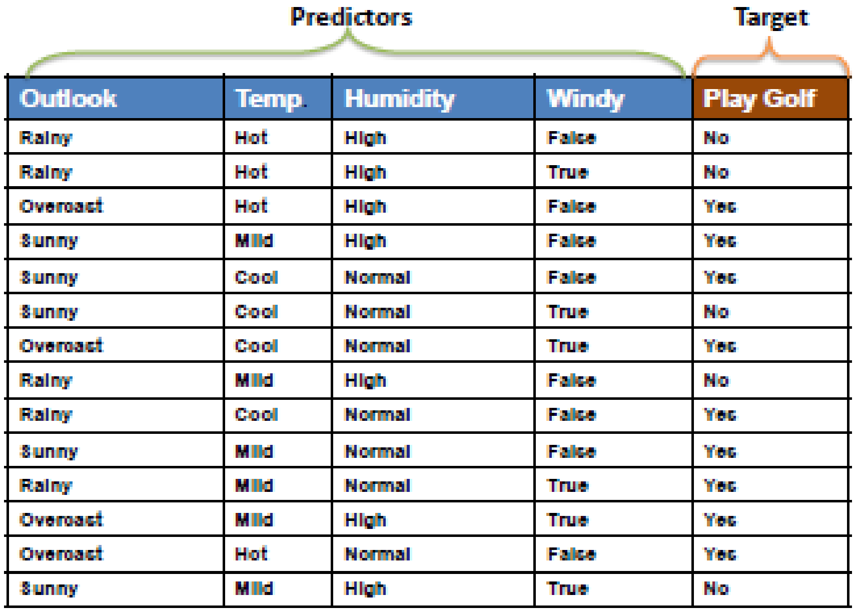
What’s Decision Tree

- A decision support tool that uses a tree-like graph of decisions and their possible consequences
- Common method in decision tree
  - ID3
  - CART





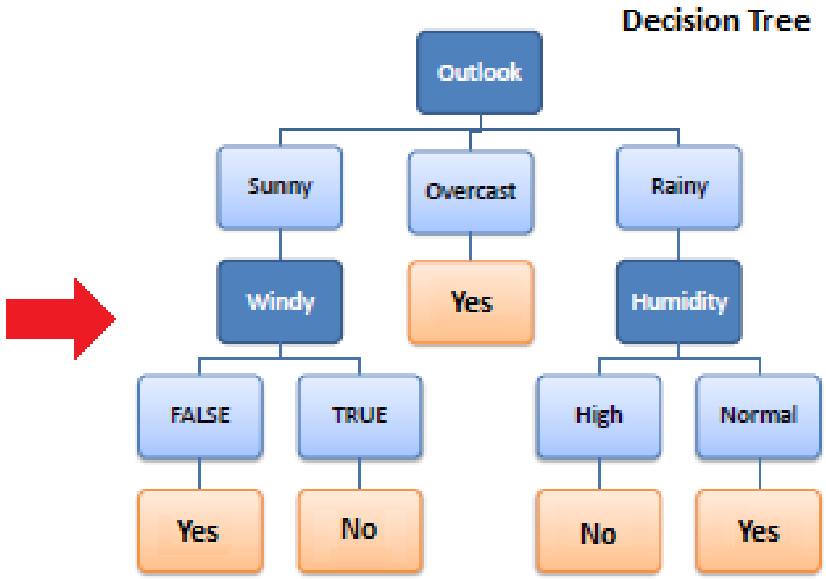
What’s Decision Tree



Predictors

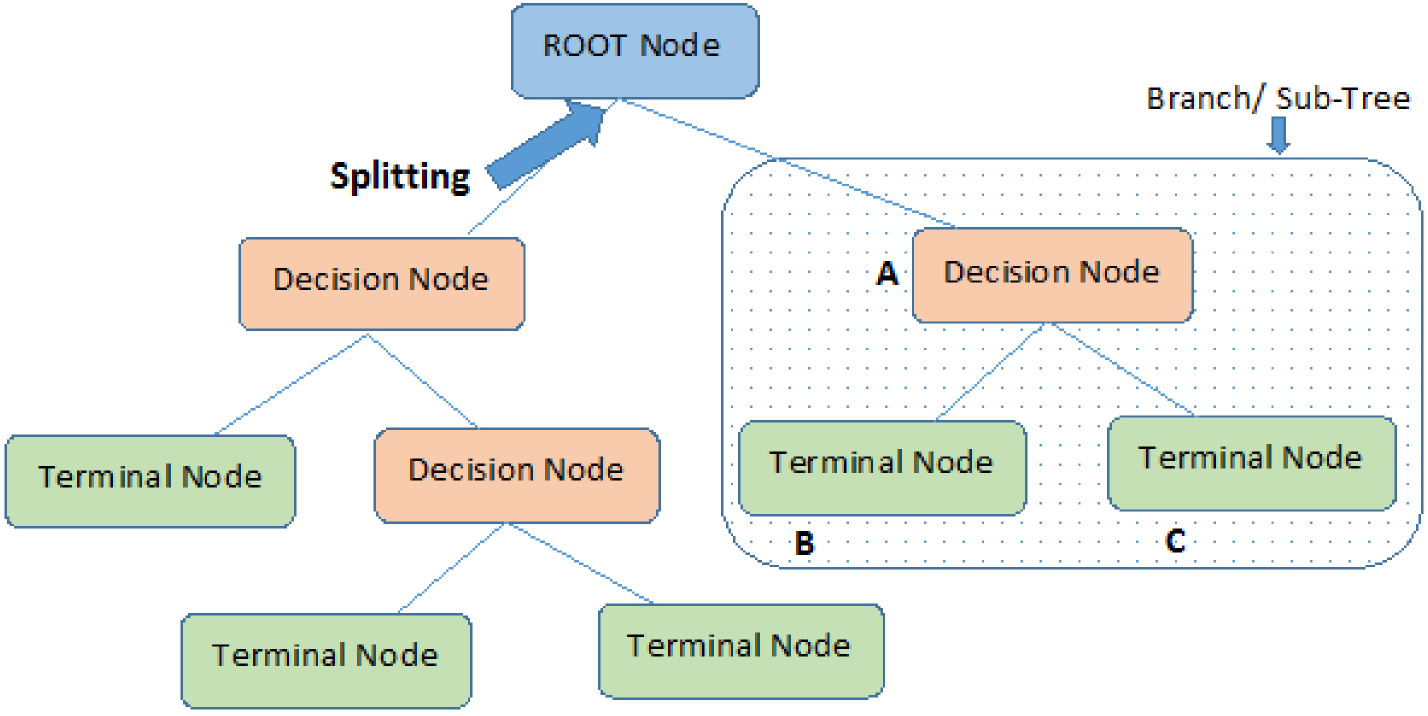
Target

Outlook	Temp.	Humidity	Windy	Play Golf
Rainy	Hot	High	False	No
Rainy	Hot	High	True	No
Overcast	Hot	High	False	Yes
Sunny	Mild	High	False	Yes
Sunny	Cool	Normal	False	Yes
Sunny	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Rainy	Mild	High	False	No
Rainy	Cool	Normal	False	Yes
Sunny	Mild	Normal	False	Yes
Rainy	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Sunny	Mild	High	True	No



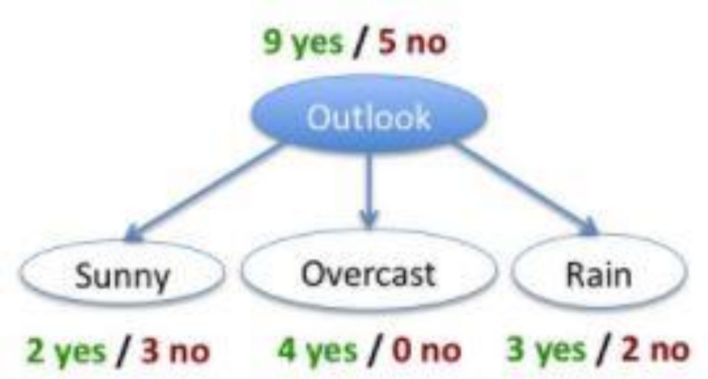
Decision Tree

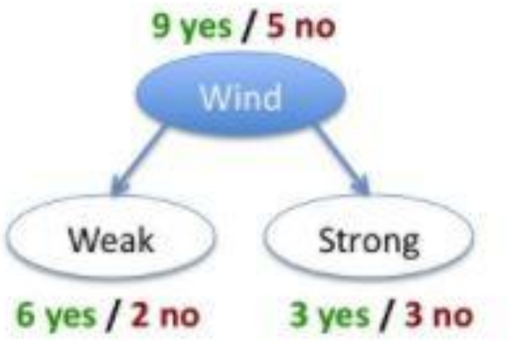
Terminology in Decision Tree



Note:- A is parent node of B and C.

How to split on each node?





How to define a good split

How to split on each node?

- **Information/Gini gain**
  - **Index to decide how to split each node**
  - **Usually, we choose max information/gini gain as candidate to split**

CART

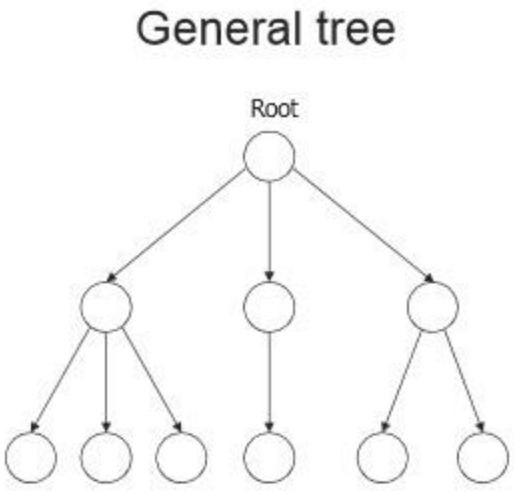
$$Gini\ Gain = Gini(before\ splitting) - E[Gini(after\ splitting)]$$

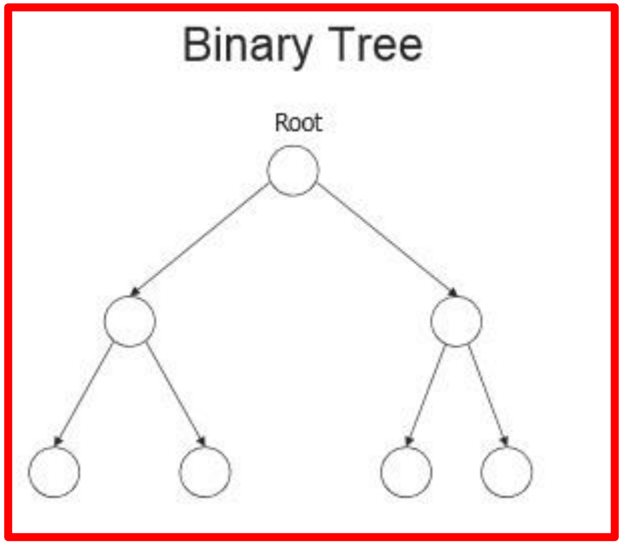
ID3

$$Information\ Gain = Entropy(before\ splitting) - E[Entropy(after\ splitting)]$$

Decision Tree - CART

- Classification and Regression Trees(CART) model is a binary tree
- Split Based on One Variable
- Use Gini impurity to define attribute complexity under each feature
- Use Gini gain to split tree





Gini Impurity

J classes and each pi is probability of class i

$$\sum_{i=1}^{J} p_i(1-p_i) = \sum_{i=1}^{J}(p_i - {p_i}^2) = \sum_{i=1}^{J} p_i - \sum_{i=1}^{J} {p_i}^2 = 1 - \sum_{i=1}^{J} {p_i}^2$$

Class 1	0
Class 2	6

$$p(class\ 1) = \frac{0}{6}, \qquad p(class\ 2) = \frac{6}{6}$$

$$Gini = 1 - (\frac{0}{6})^2 - (\frac{6}{6})^2 = 0$$

Class 1	1
Class 2	5

$$p(class\ 1) = \frac{1}{6}, \qquad p(class\ 2) = \frac{5}{6}$$

$$Gini = 1 - (\frac{1}{6})^2 - (\frac{5}{6})^2 = 0.278$$

Class 1	2
Class 2	4

$$p(class\ 1) = \frac{2}{6}, \qquad p(class\ 2) = \frac{4}{6}$$

$$Gini = 1 - (\frac{2}{6})^2 - (\frac{4}{6})^2 = 0.444$$

Example

Gini **Large** → **Less** Purity

Gini **Small** → **More** Purity

CART use Gini Gain to Split node

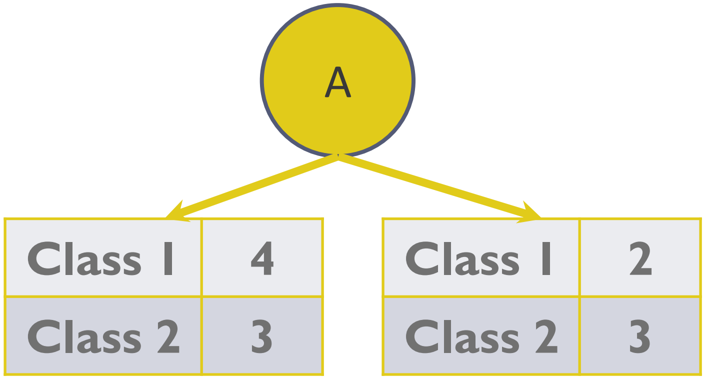
before splitting

Class 1	6
Class 2	6

Gini(before splitting) = 0.5

after splitting

suppose there are two ways (A or B) to split the data



Class 1	4
Class 2	3

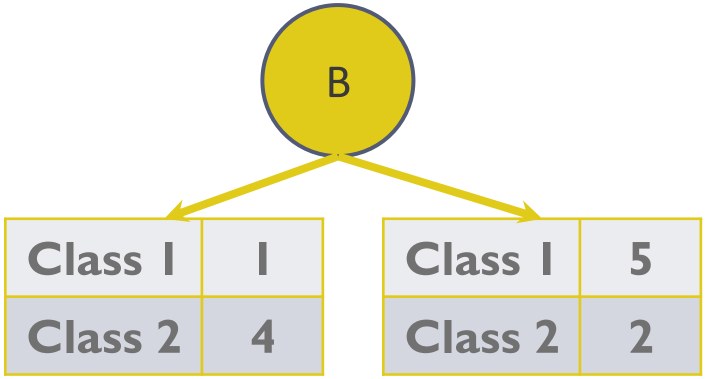
Gini = 0.489

Class 1	2
Class 2	3

Gini = 0.48

E[Gini(after splitting)]

$$=\frac{7}{12}*0.489+\frac{5}{12}*0.48=0.4852$$



Class 1	1
Class 2	4

Gini = 0.32

Class 1	5
Class 2	2

Gini = 0.408

E[Gini(after splitting)]

$$=\frac{5}{12}*0.32+\frac{7}{12}*0.408=0.37$$

CART use Gini Gain to Split node

before splitting

Class 1	6
Class 2	6

Gini(before splitting) = 0.5

after splitting

Gini Gain on A way
=Gini(before splitting) -E[Gini(after splitting)]
=0.015

Gini Gain on B way
= Gini(before splitting) -E[Gini(after splitting)]
=0.13

Split on B way is better

CART use Gini Gain to Split node

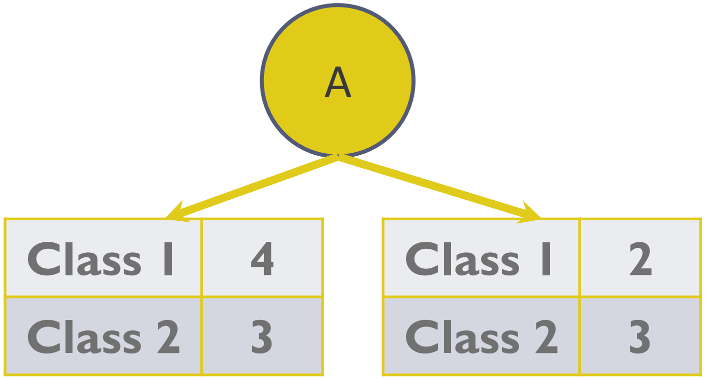
before splitting

Class 1	6
Class 2	6

Gini(before splitting) = 0.5

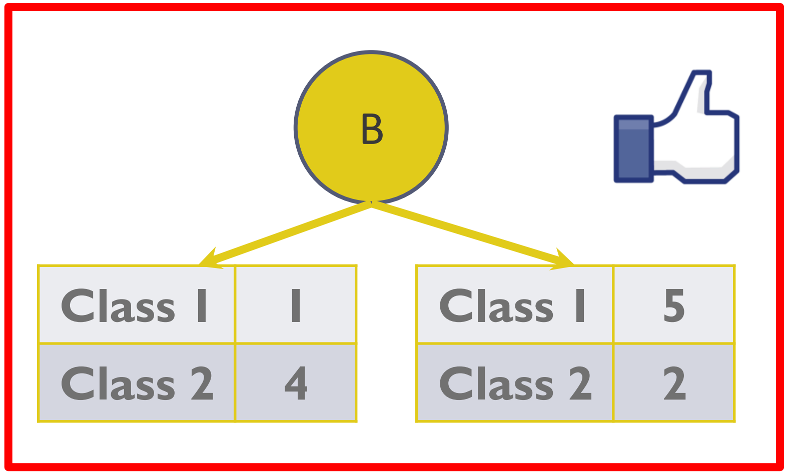
after splitting

suppose there are two ways (A or B) to split the data



Class 1	4
Class 2	3

Class 1	2
Class 2	3

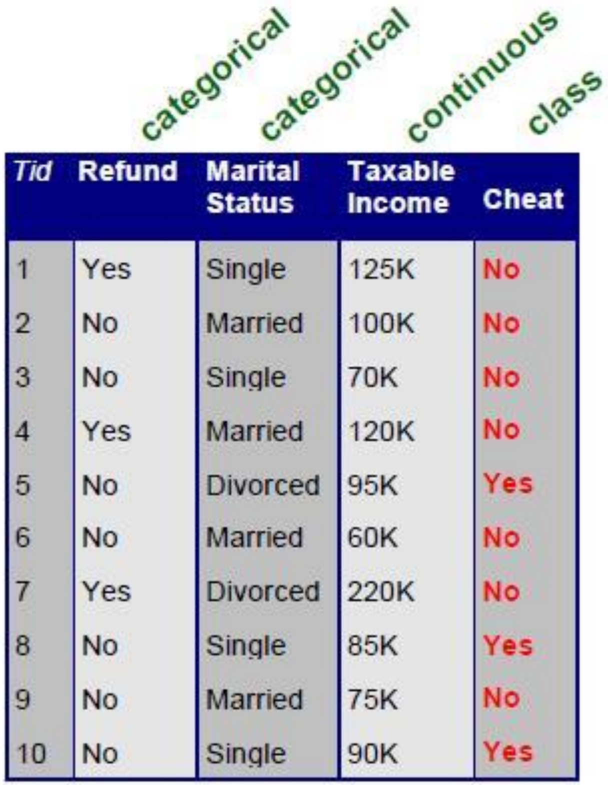


Class 1	1
Class 2	4

Class 1	5
Class 2	2

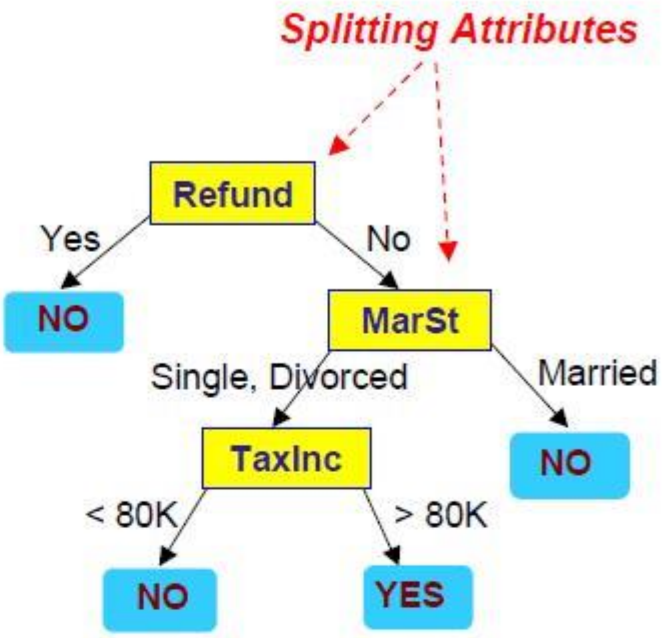
Split on B way is better

Decision Tree – CART Example



	categorical	categorical	continuous	class
Tid	**Refund**	**Marital Status**	**Taxable Income**	**Cheat**
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Training Data



Model: Decision Tree

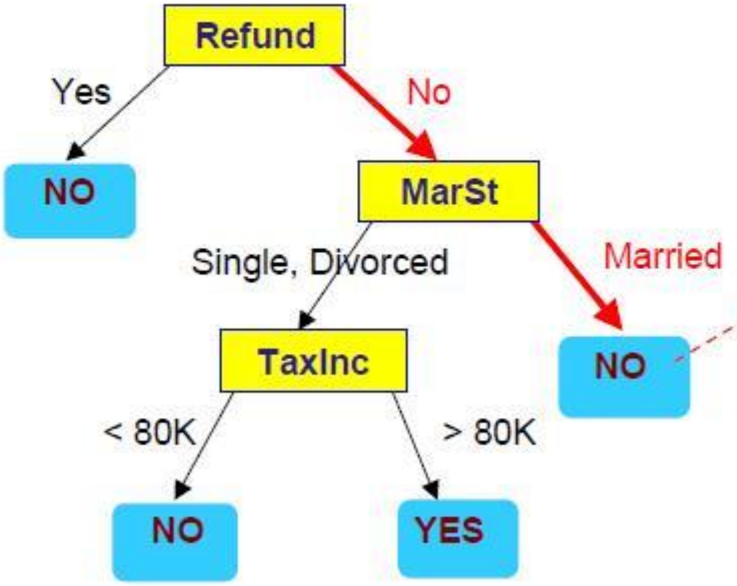
How to deal with continuous attributes

- There are many different way to deal with continuous attributes when building decision tree
  - The most simple way is to split by average of continuous attributes

Decision Tree – CART Example

Test Data

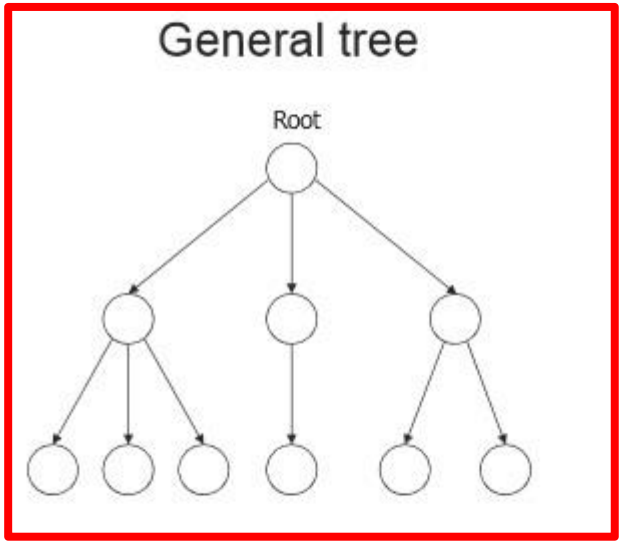
Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

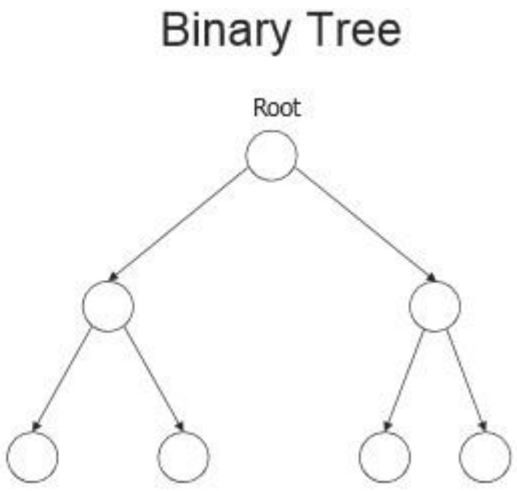


Assign Cheat to “No”

Decision Tree – ID3

- Iterative Dichotomiser 3(ID3) is a famous algorithm to generate decision tree
- Use information gain as index to split each node
- Note that ID3 can split multiple branch at each node





Decision Tree – ID3

- **Entropy**

$$E(S)=\sum_{i=1}^{c}-p_i\log_2 p_i$$

Class 1	0
Class 2	6

$$p(class\ 1)=\frac{0}{6},\qquad p(class\ 2)=\frac{6}{6}$$

$$Entropy=-0*\log(0)-1*\log(1)=0$$

Class 1	1
Class 2	5

$$p(class\ 1)=\frac{1}{6},\qquad p(class\ 2)=\frac{5}{6}$$

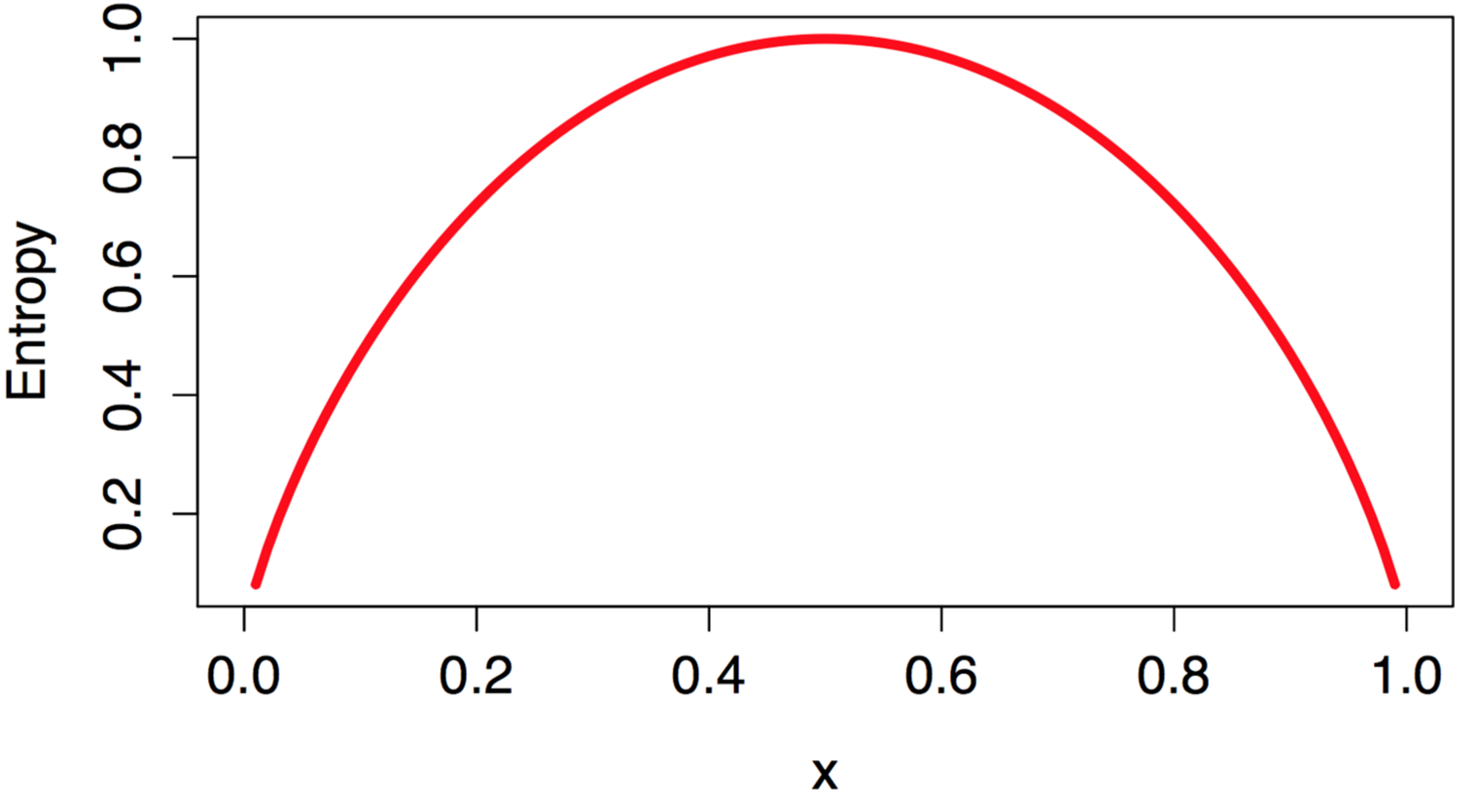
$$Entropy=-\frac{1}{6}*\log\left(\frac{1}{6}\right)-\frac{5}{6}*\log\left(\frac{5}{6}\right)=0.65$$

Class 1	2
Class 2	4

$$p(class\ 1)=\frac{2}{6},\qquad p(class\ 2)=\frac{4}{6}$$

$$Entropy=-\frac{2}{6}*\log\left(\frac{2}{6}\right)-\frac{4}{6}*\log\left(\frac{4}{6}\right)=0.91$$

Decision Tree – ID3



$$Entropy = -x * \log(x) - (1-x) * log(1-x)$$

ID3 use Entropy to Split node

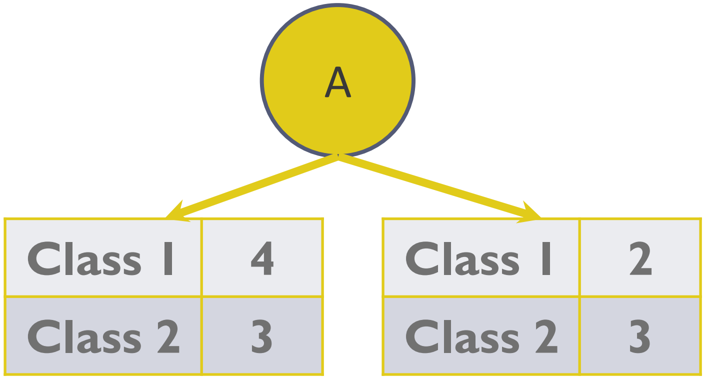
before splitting

Class 1	6
Class 2	6

Entropy(before splitting) = 0.301

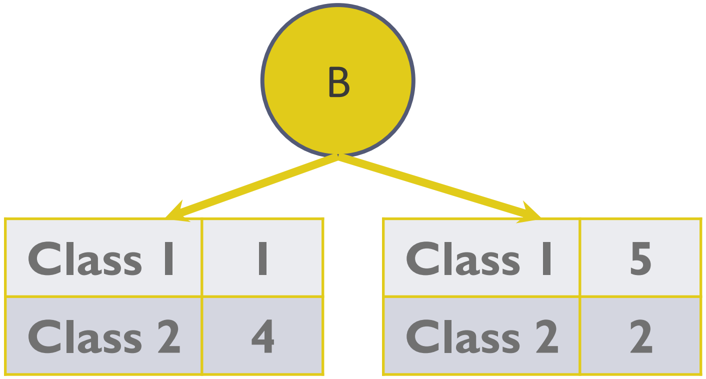
after splitting

suppose there are two ways (A or B) to split the data



Class 1	4
Class 2	3

Class 1	2
Class 2	3



Class 1	1
Class 2	4

Class 1	5
Class 2	2

Entropy = 0.297 Entropy = 0.292

Entropy = 0.217 Entropy = 0.259

E[Gini(after splitting)]

$$=\frac{7}{12} * 0.297 + \frac{5}{12} * 0.292 = 0.294$$

E[Gini(after splitting)]

$$=\frac{5}{12} * 0.217 + \frac{7}{12} * 0.259 = 0.242$$

ID3 use Entropy to Split node

before splitting

Class 1	6
Class 2	6

Entropy(before splitting) = 0.5

after splitting

Information Gain on A way
=Entropy(before splitting) -E[Entropy(after splitting)]
=0.007

Information Gain on B way
= Entropy (before splitting) -E[Entropy(after splitting)]
=0.069

Split on B way is better

ID3 use Entropy to Split node

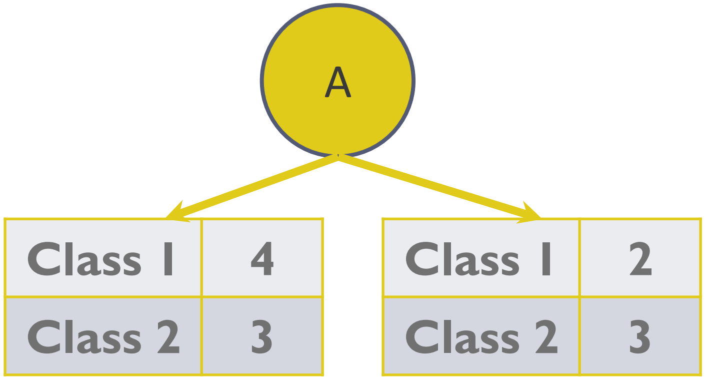
before splitting

Class 1	6
Class 2	6

Gini(before splitting) = 0.5

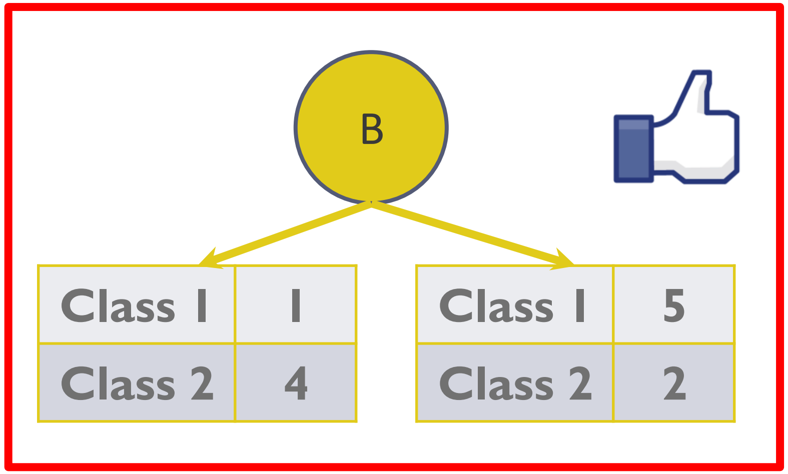
after splitting

suppose there are two ways (A or B) to split the data



Class 1	4
Class 2	3

Class 1	2
Class 2	3



Class 1	1
Class 2	4

Class 1	5
Class 2	2

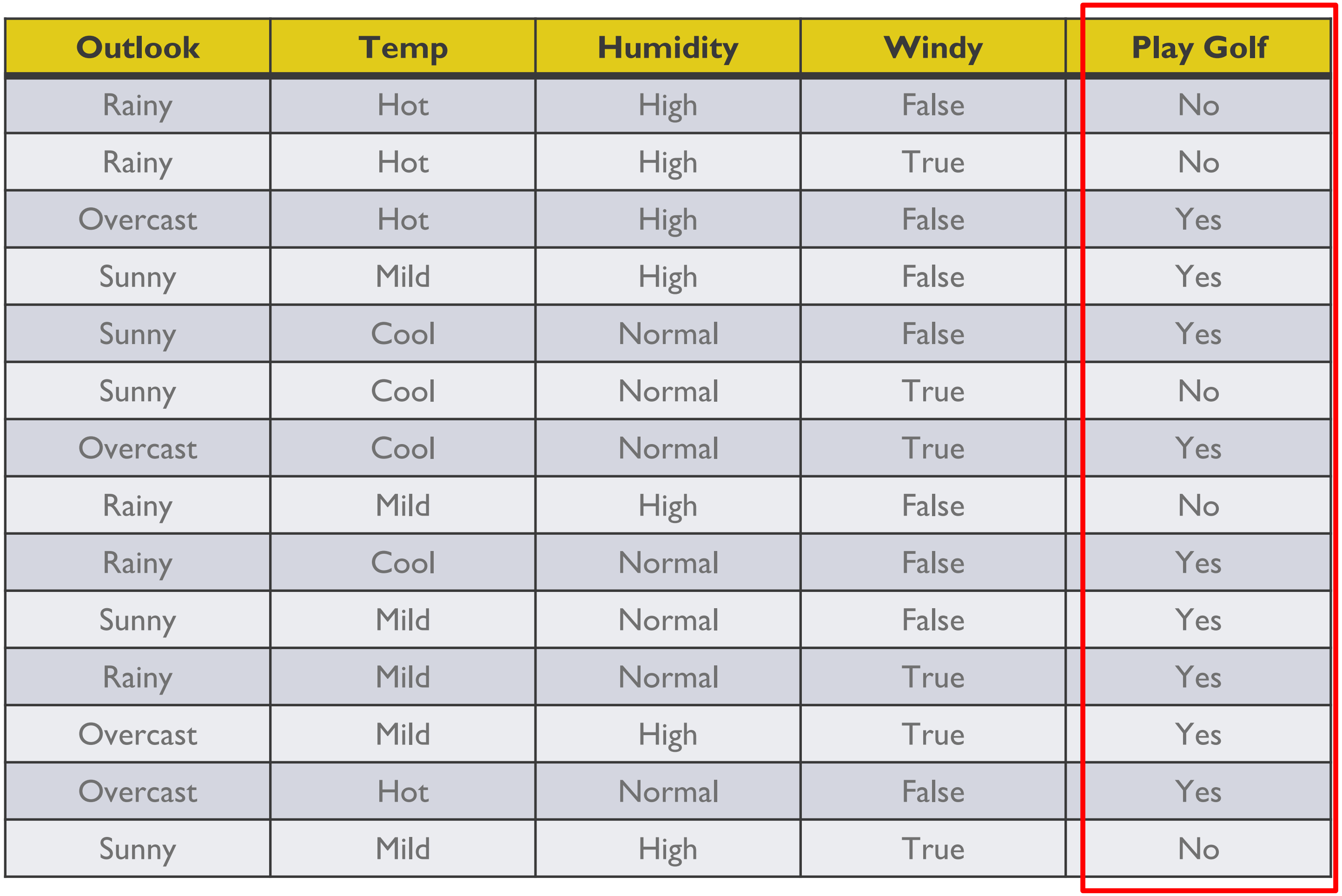
Split on B way is better

Decision Tree – ID3 Example

Predict if playing golf or not

Outlook	Temp	Humidity	Windy	Play Golf
Rainy	Hot	High	False	No
Rainy	Hot	High	True	No
Overcast	Hot	High	False	Yes
Sunny	Mild	High	False	Yes
Sunny	Cool	Normal	False	Yes
Sunny	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Rainy	Mild	High	False	No
Rainy	Cool	Normal	False	Yes
Sunny	Mild	Normal	False	Yes
Rainy	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Sunny	Mild	High	True	No

Decision Tree – ID3 Example



Outlook	Temp	Humidity	Windy	Play Golf
Rainy	Hot	High	False	No
Rainy	Hot	High	True	No
Overcast	Hot	High	False	Yes
Sunny	Mild	High	False	Yes
Sunny	Cool	Normal	False	Yes
Sunny	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Rainy	Mild	High	False	No
Rainy	Cool	Normal	False	Yes
Sunny	Mild	Normal	False	Yes
Rainy	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Sunny	Mild	High	True	No

$$Entropy(before\,split) = -\frac{5}{14} * \log\left(\frac{5}{14}\right) - \frac{9}{14} * \log\left(\frac{9}{14}\right) = 0.94$$

Decision Tree – ID3 Example

calculate entropy if splitting on **outlook** column

		Play Golf		
		Yes	No	
Outlook	Sunny	3	2	5
	Overcast	4	0	4
	Rainy	2	3	5
				14

$$E[Entropy(after\ splitting)]$$
$$= P(sunny) * E(3,2) + P(overcast) * E(4,0) + P(rainy) * E(2,3)$$
$$= \left(\frac{5}{14}\right) * 0.971 + \left(\frac{4}{14}\right) * 0 + \left(\frac{5}{14}\right) * 0.971 = 0.693$$

Decision Tree – ID3 Example

		Play Golf	
		Yes	No
Outlook	Sunny	3	2
	Overcast	4	0
	Rainy	2	3
Information Gain = 0.247			

Max Gain

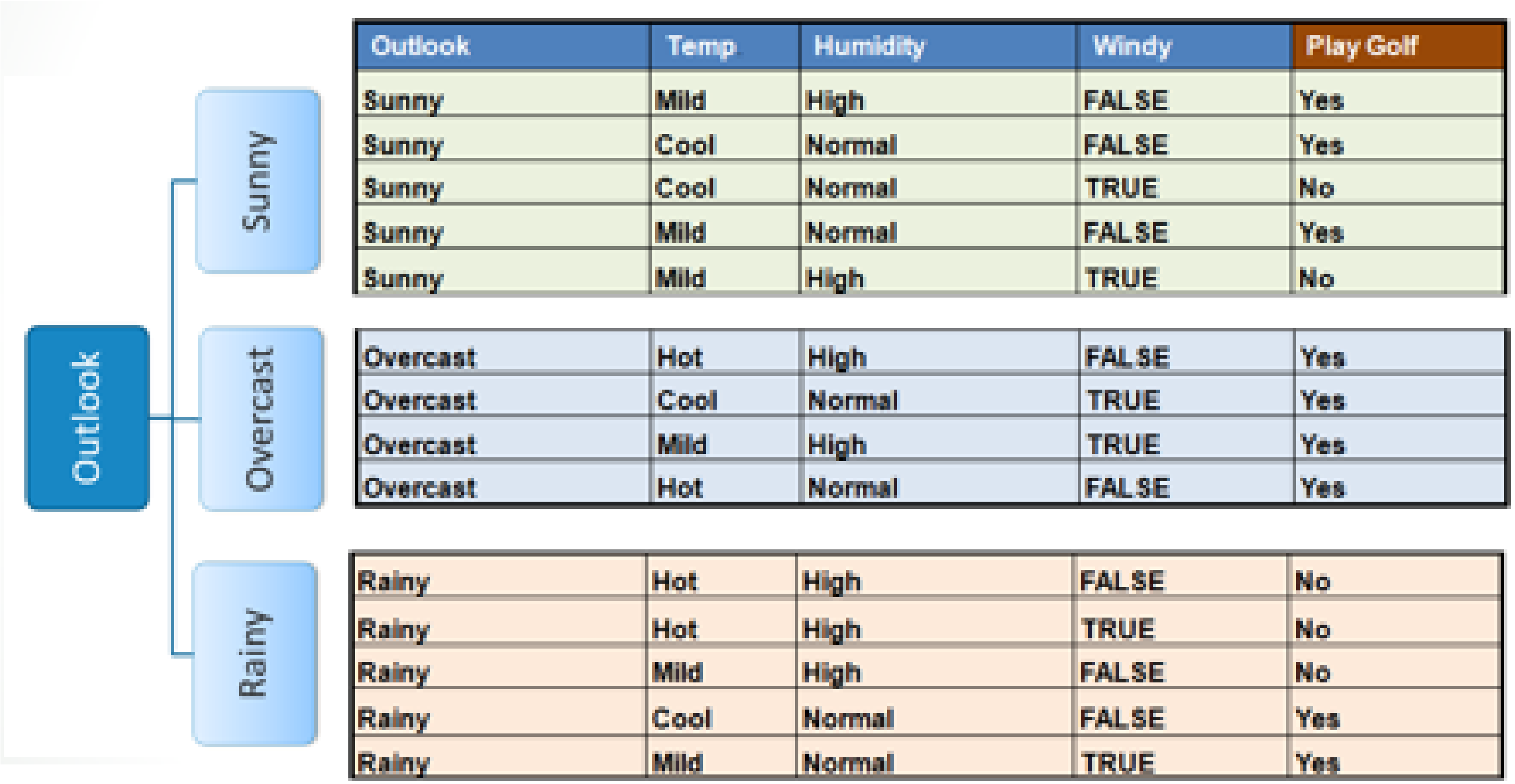
		Play Golf	
		Yes	No
Temp	Hot	2	2
	Mild	4	2
	Cool	3	1
Information Gain = 0.029			

		Play Golf	
		Yes	No
Humidity	High	3	4
	Normal	6	1
Information Gain = 0.152			

		Play Golf	
		Yes	No
Windy	False	6	2
	True	3	3
Information Gain = 0.048			

Decision Tree – ID3 Example

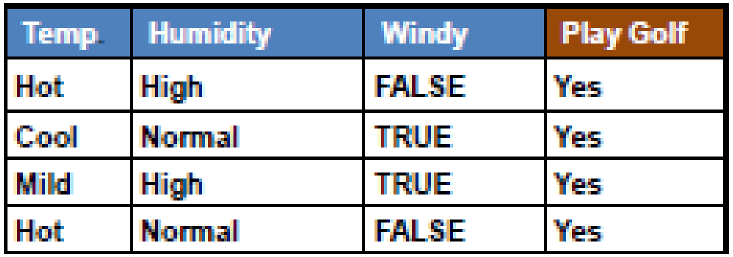
After first splitting, decision tree look like the following



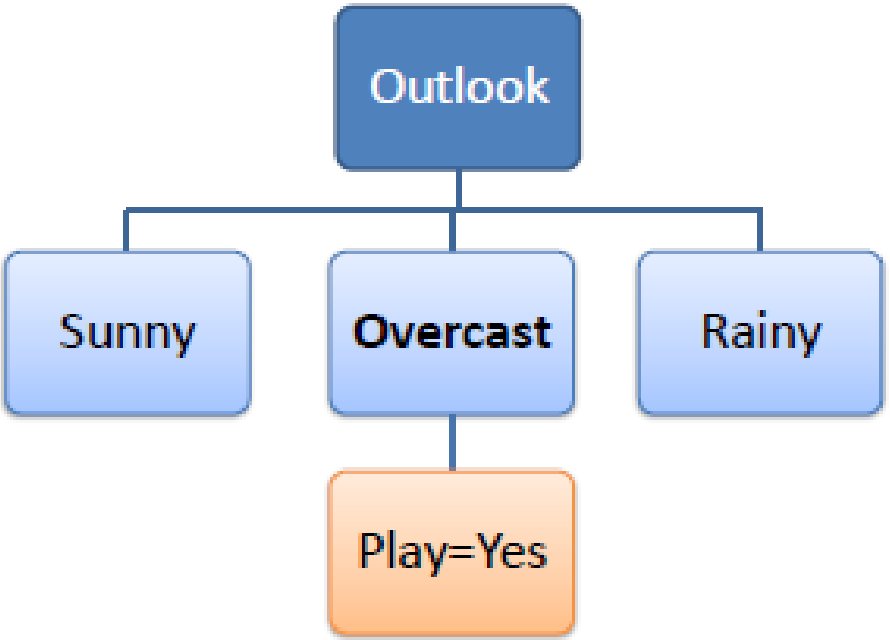
Outlook	Temp	Humidity	Windy	Play Golf
Sunny	Mild	High	FALSE	Yes
Sunny	Cool	Normal	FALSE	Yes
Sunny	Cool	Normal	TRUE	No
Sunny	Mild	Normal	FALSE	Yes
Sunny	Mild	High	TRUE	No
Overcast	Hot	High	FALSE	Yes
Overcast	Cool	Normal	TRUE	Yes
Overcast	Mild	High	TRUE	Yes
Overcast	Hot	Normal	FALSE	Yes
Rainy	Hot	High	FALSE	No
Rainy	Hot	High	TRUE	No
Rainy	Mild	High	FALSE	No
Rainy	Cool	Normal	FALSE	Yes
Rainy	Mild	Normal	TRUE	Yes

Decision Tree – ID3 Example

No need to further split overcast because all of target are “Yes”



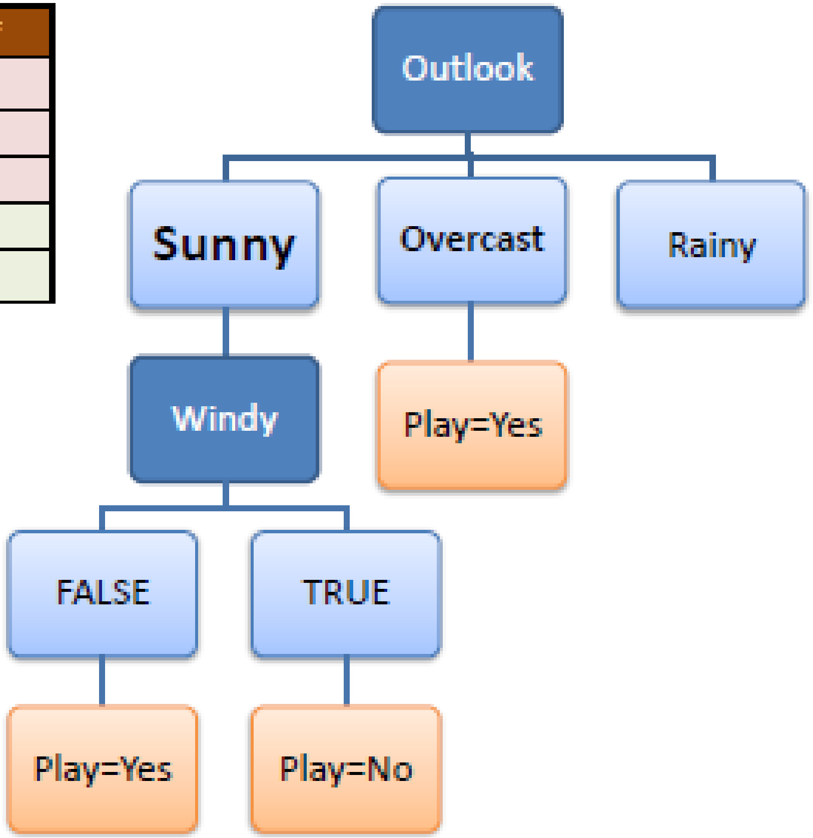
Temp.	Humidity	Windy	Play Golf
Hot	High	FALSE	Yes
Cool	Normal	TRUE	Yes
Mild	High	TRUE	Yes
Hot	Normal	FALSE	Yes



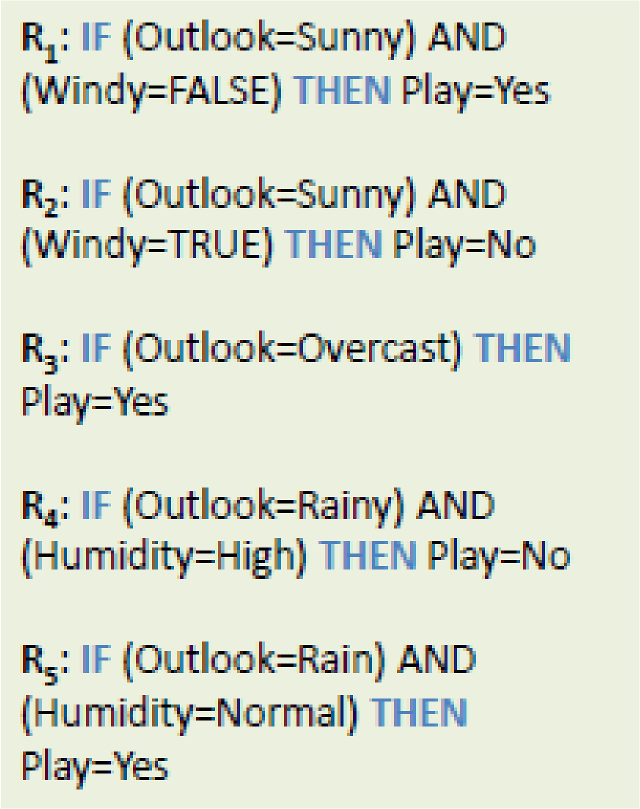
Decision Tree – ID3 Example

Continue split nodes on same method

Temp.	Humidity	Windy	Play Golf
Mild	High	FALSE	Yes
Cool	Normal	FALSE	Yes
Mild	Normal	FALSE	Yes
Cool	Normal	TRUE	No
Mild	High	TRUE	No



Decision Tree – ID3 Example



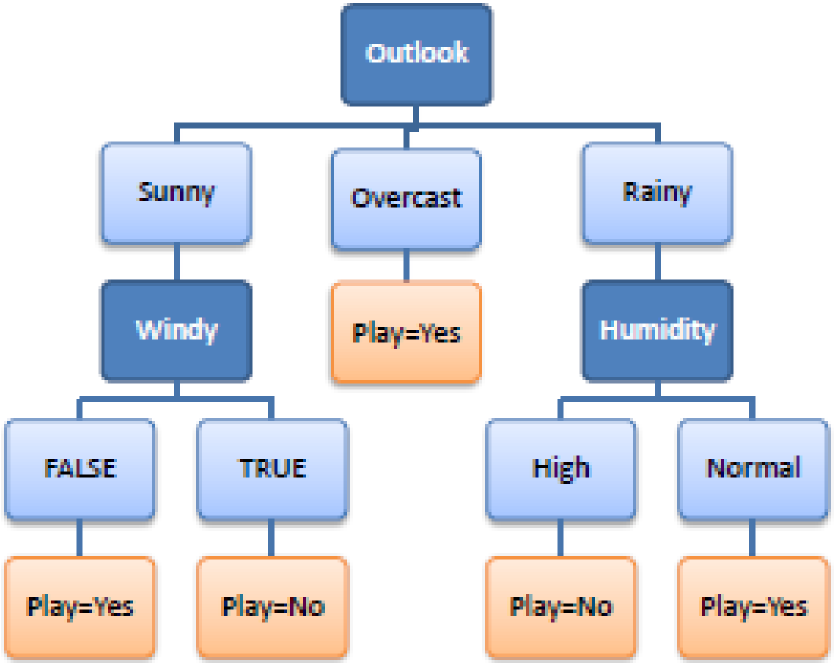
R_1: IF (Outlook=Sunny) AND (Windy=FALSE) THEN Play=Yes

R_2: IF (Outlook=Sunny) AND (Windy=TRUE) THEN Play=No

R_3: IF (Outlook=Overcast) THEN Play=Yes

R_4: IF (Outlook=Rainy) AND (Humidity=High) THEN Play=No

R_5: IF (Outlook=Rain) AND (Humidity=Normal) THEN Play=Yes



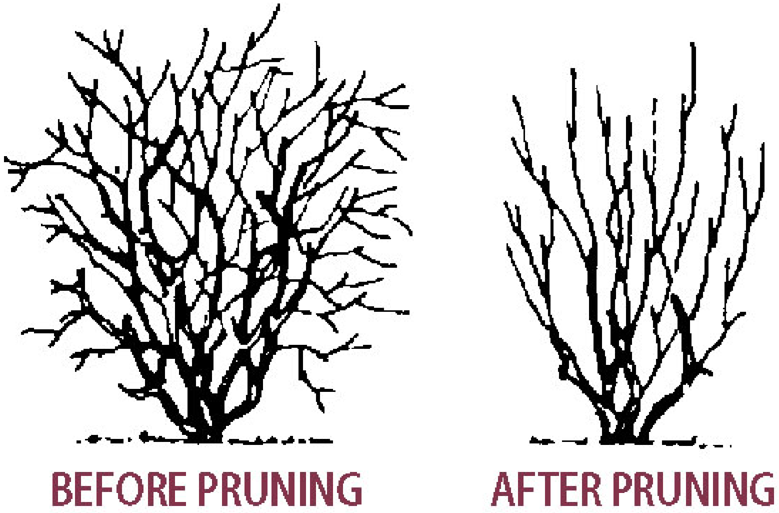
http://www.saedsayad.com/decision_tree.htm

Decision Tree – ID3

- **Calculate target Entropy**
- **Find the information gain on each attribute**
- **Split tree on an attribute which information gain is max**
- **Repeat**

Pruning

- **Pruning is a technique that reduces the size of decision trees**
  - **Reduce model complexity and overfitting**



Stopping Condition

- **Pre-pruning**
  - **Stop the algorithm before it becomes a fully-grown tree**
    - Stop if all instances belong to the same class
    - Stop if number of instances is less than some user-specified threshold
    - Stop if expanding the current node does not improve impurity measures
    - ……
- **Post-pruning**
  - **Grow decision tree to its entirety and trim the nodes of the decision tree in a bottom-up**
  - **If generalization error improves after trimming, replace sub-tree by a leaf node**

Example and Practice

- **Example**
  - **Decision Tree (CART)**
    - example/supervised learning
- **Practice**
  - **Try to use decision tree to predict if abalone is old or young**
    - dataset/abalone.csv
    - practice/supervised learning
    - we assume age > 8 is old and other is young
  - **More information about the dataset**
    - https://archive.ics.uci.edu/ml/datasets/abalone