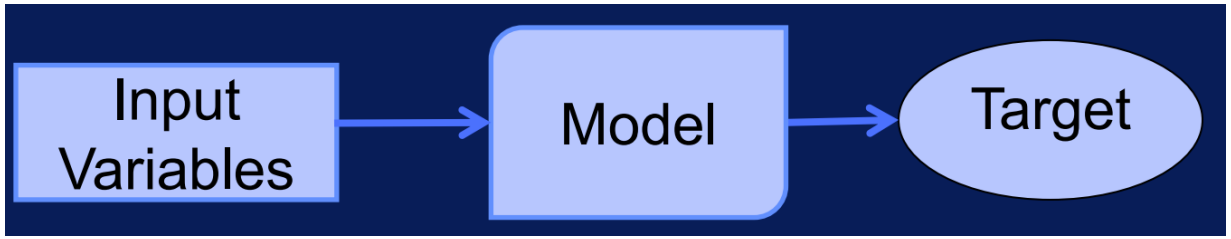
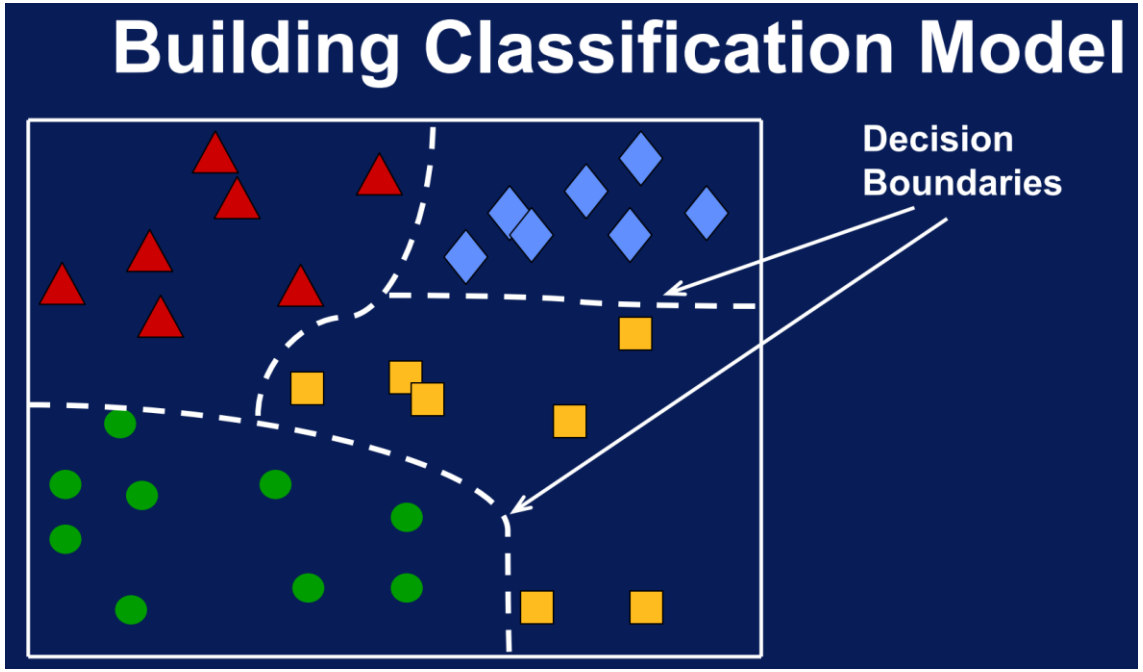
Classification Model

classification

- Predict: Category from input variables
- Goal: Match model outputs to targets (desired outputs)





Classification algorithms

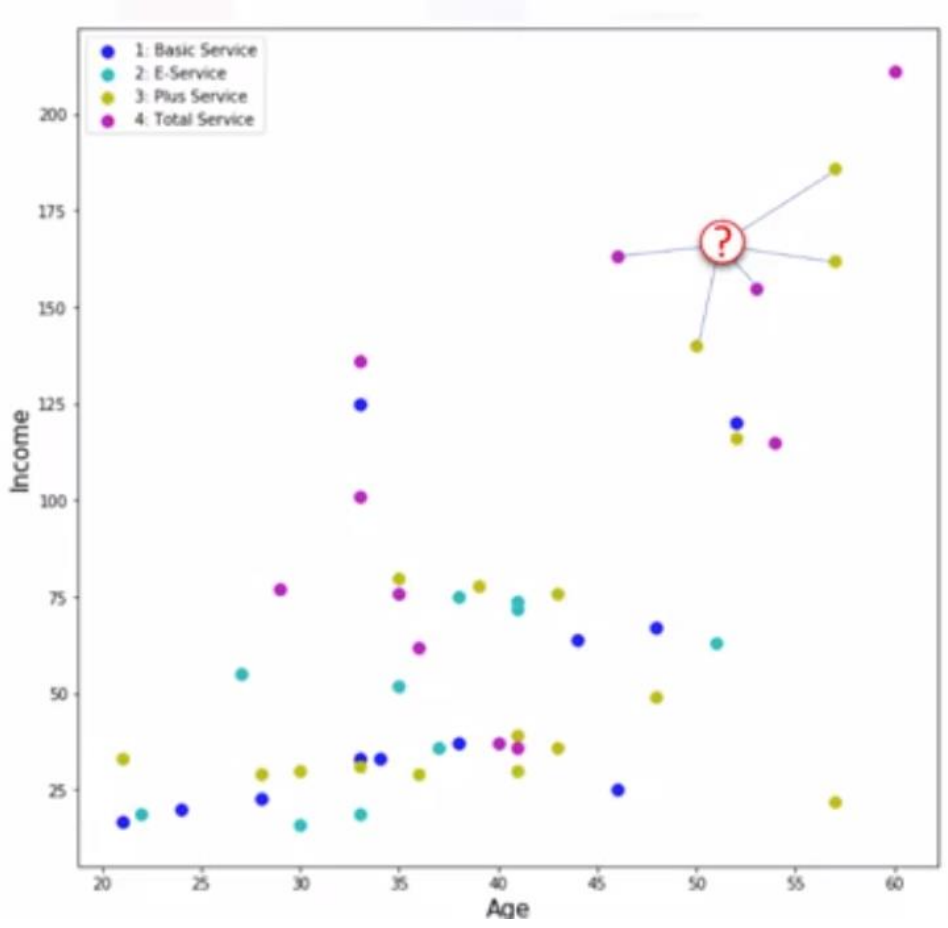
- kNN (k-Nearest Neighbor)
- Decision tree
- Logistic Regression
- SVM (Support Vector Machine)
- Neural Network

K-NN

K-Nearest Neighbor Classification

K-NN (K-nearest neighbor classification)

- Classify based on similarity to other cases
  - Cases that are near each other are said to be “neighbors”
  - Assign a class by looking at the most popular class from its k-nearest (closest) neighbors

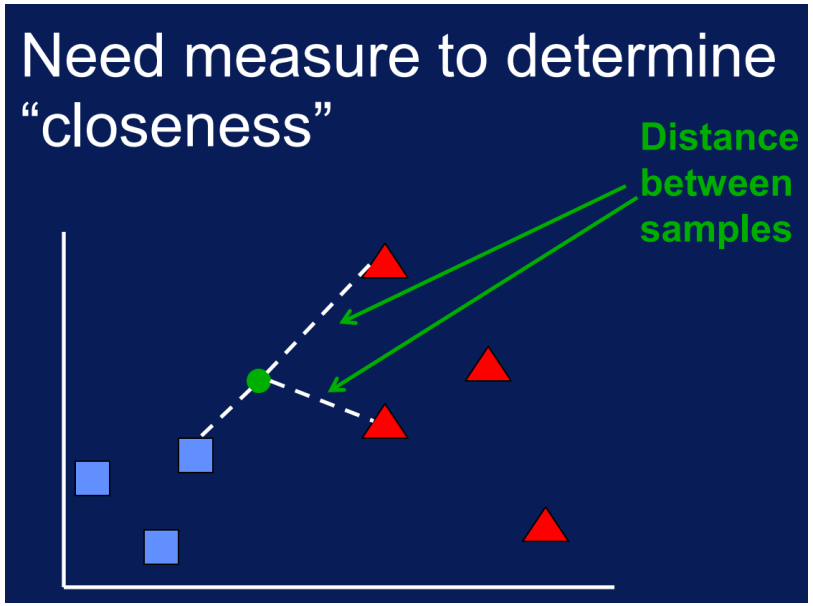


K-NN algorithm

$$\text{Dis}\,(x_1, x_2) = \sqrt{\sum_{i=0}^{n}(x_{1i} - x_{2i})^2}$$

K-NN parameter = K

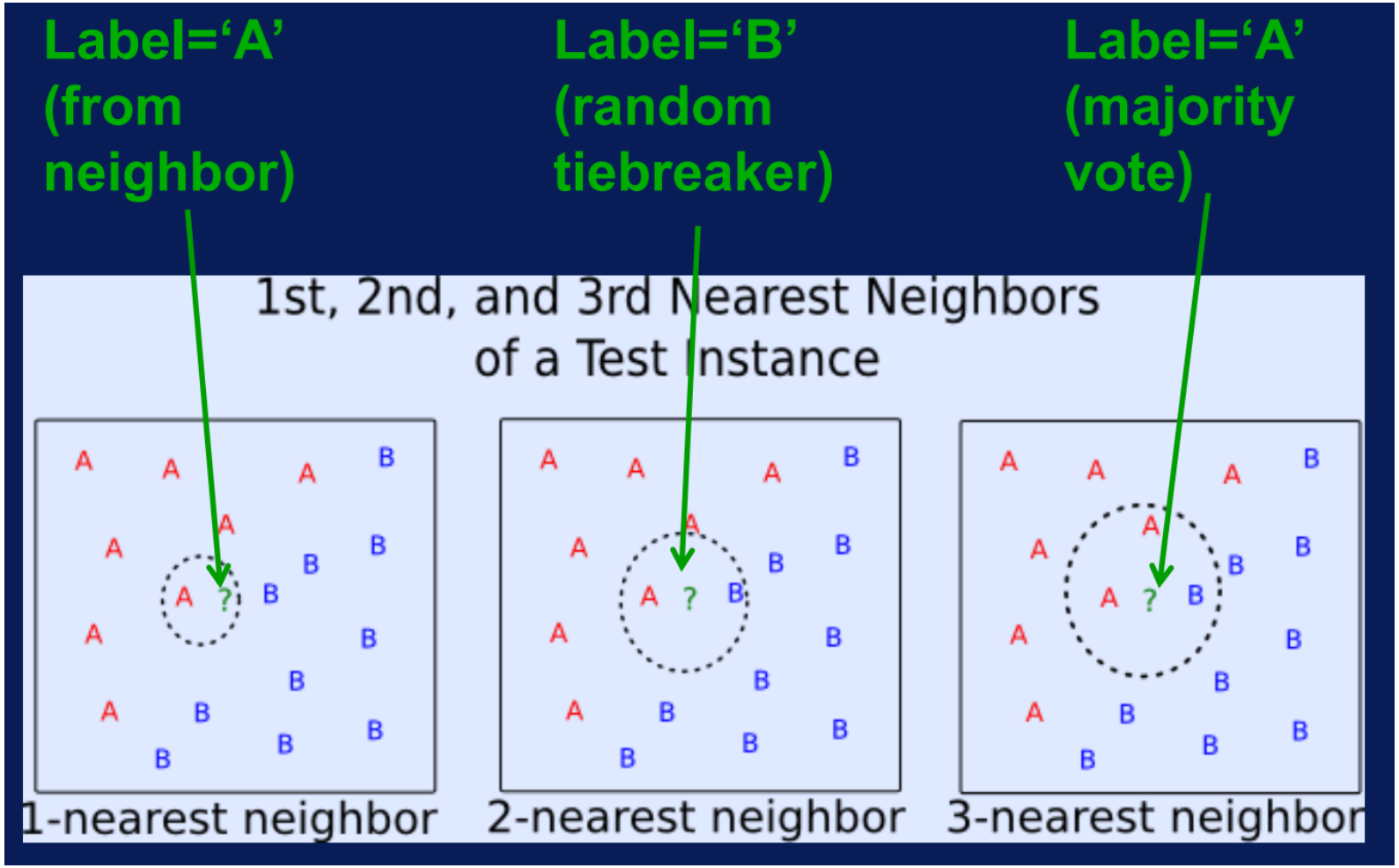
1. Pick a value for K.
2. Calculate the distance of unknown case from all cases.
3. Select the K-observations in the training data that are “nearest” to the unknown data point.
4. Predict the response of the unknown data point using the most popular response value from the K-nearest neighbors.



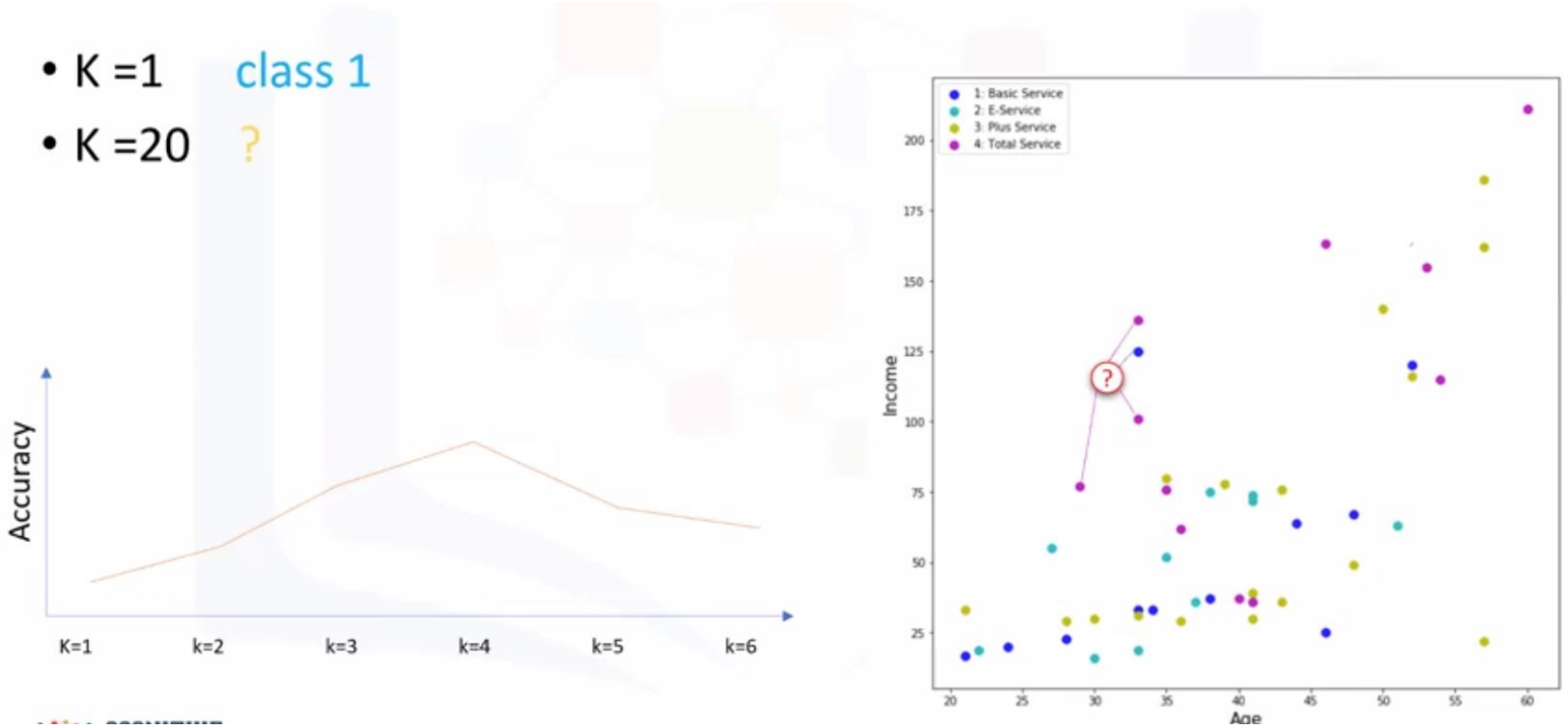
K too small -> easily cause overfitting

K-NN algorithm

$$\text{Dis}\,(x_1, x_2) = \sqrt{\sum_{i=0}^{n}(x_{1i} - x_{2i})^2}$$

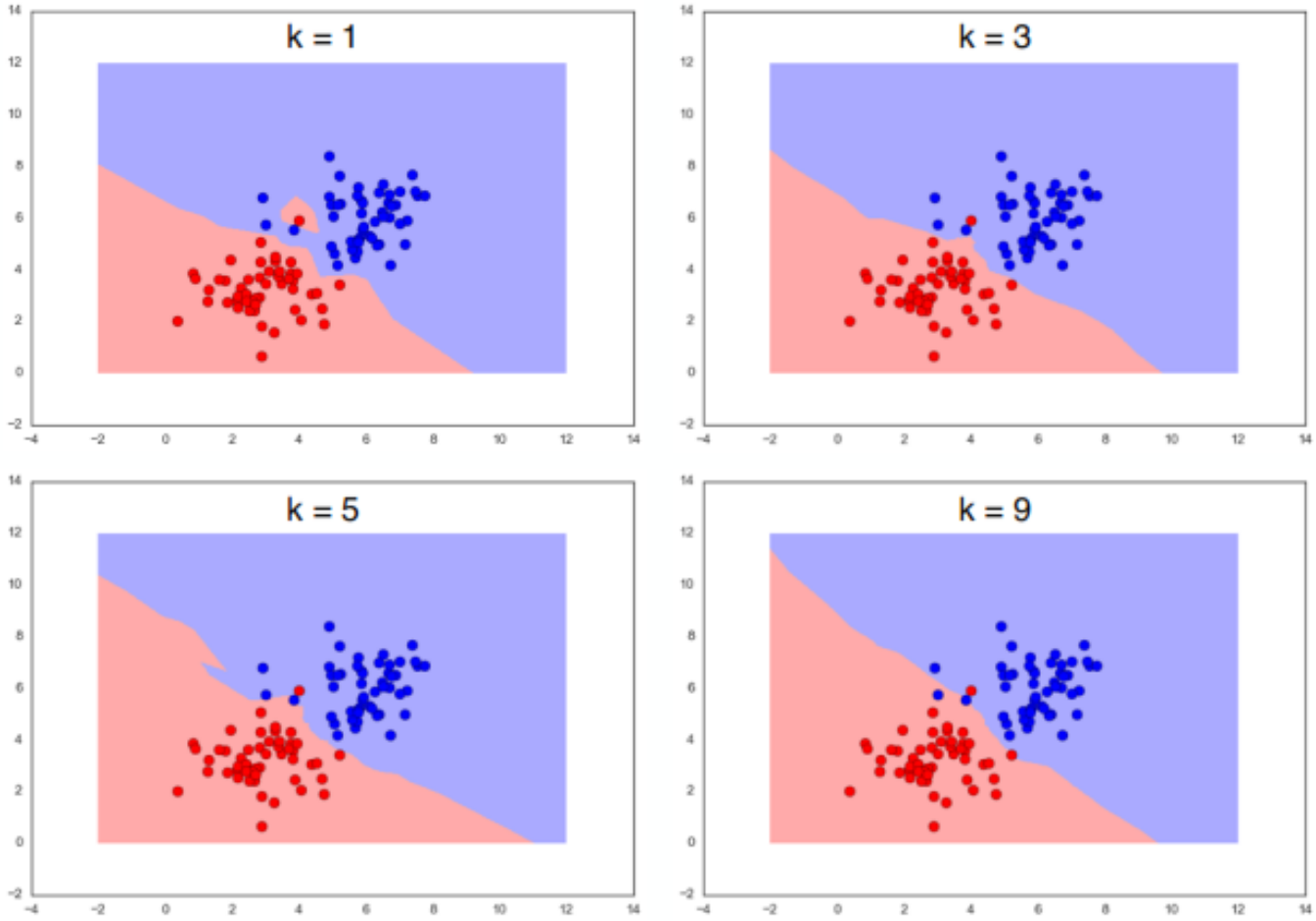


K-NN (K-nearest neighbor classification)



- Selecting best K value from training among various K

K-NN (K-nearest neighbor classification)



- Selecting best K value from training among various K

k-NN

No	Weather	Temp	Play	Weather_encoded	temp_encoded	Play_encoded
1	Sunny	Hot	No			
2	Sunny	Hot	No			
3	Overcast	Hot	Yes			
4	Rainy	Mild	Yes			
5	Rainy	Cool	Yes			
6	Rainy	Cool	No			
7	Overcast	Cool	Yes			
8	Sunny	Mild	No			
9	Sunny	Cool	Yes			
10	Rainy	Mild	Yes			
11	Sunny	Mild	Yes			
12	Overcast	Mild	Yes			
13	Overcast	Hot	Yes			
14	Rainy	Mild	No			

Test#1

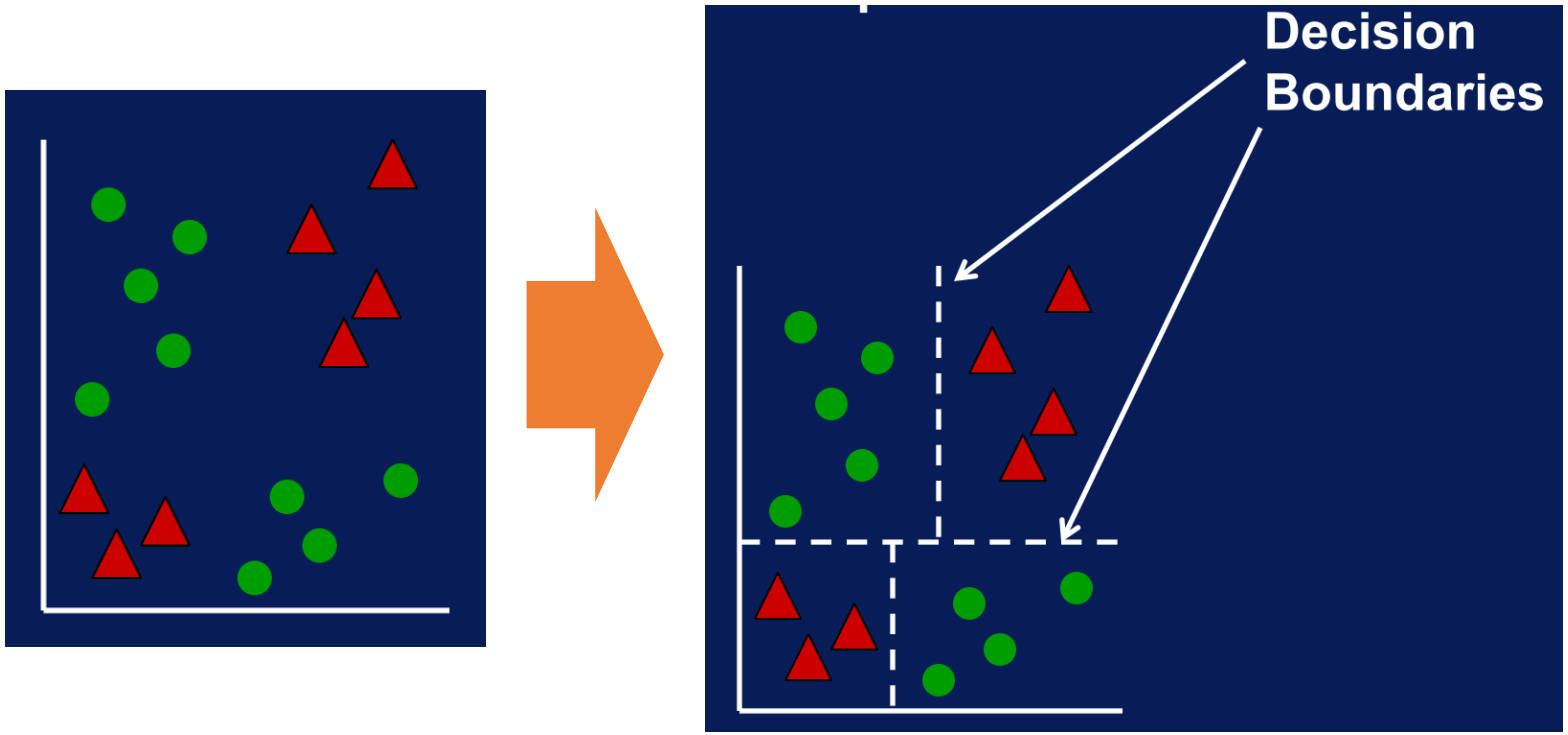
Weather = Overcast & Temp = Mild -> Play = ?
k = 1, k=3, k=6

Test#2

Weather = Rainy & Temp = Mild -> Play = ?
k=1, k=3, k=6

Decision tree

Decision tree

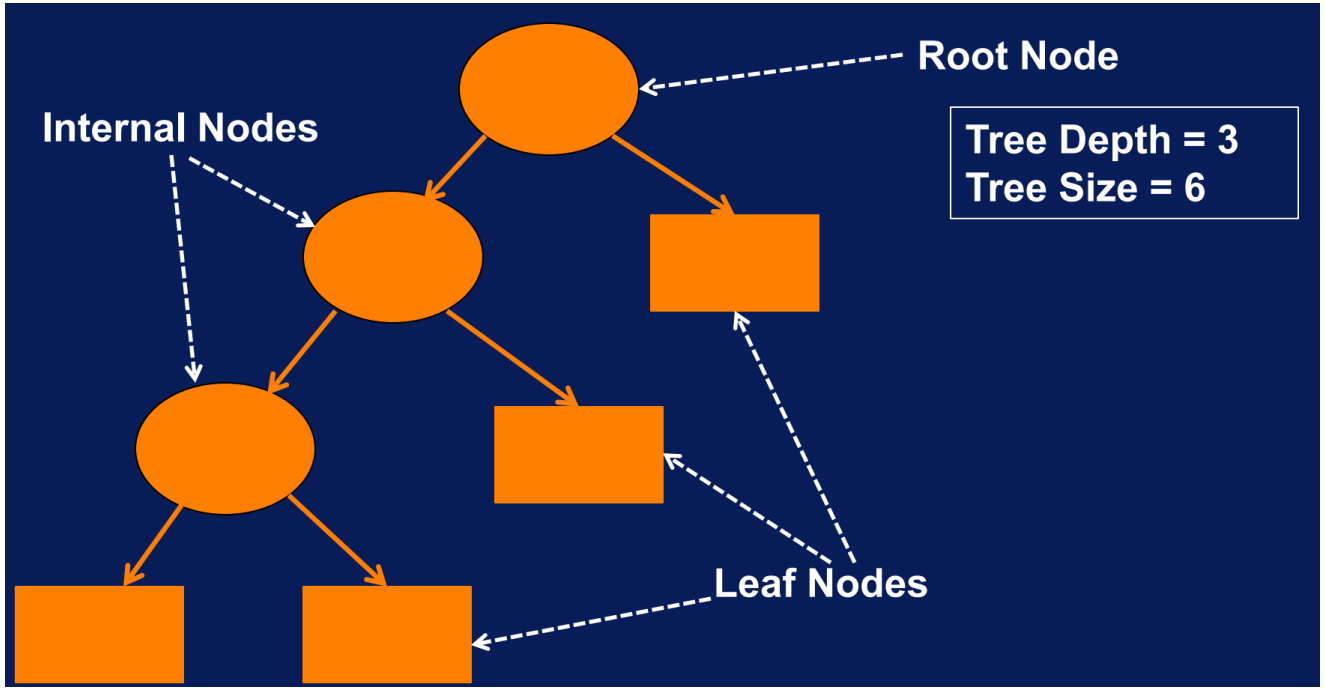


$$\text{Impurity}(S) = -\sum_{c=1}^{N} p_c \cdot \log_2 p_c$$

- Attempt to
  - Split data into “pure” regions

- Start with all samples at a node.
- Partition samples based on input to create purest subsets.
- Repeat to partition data into successively purer subsets

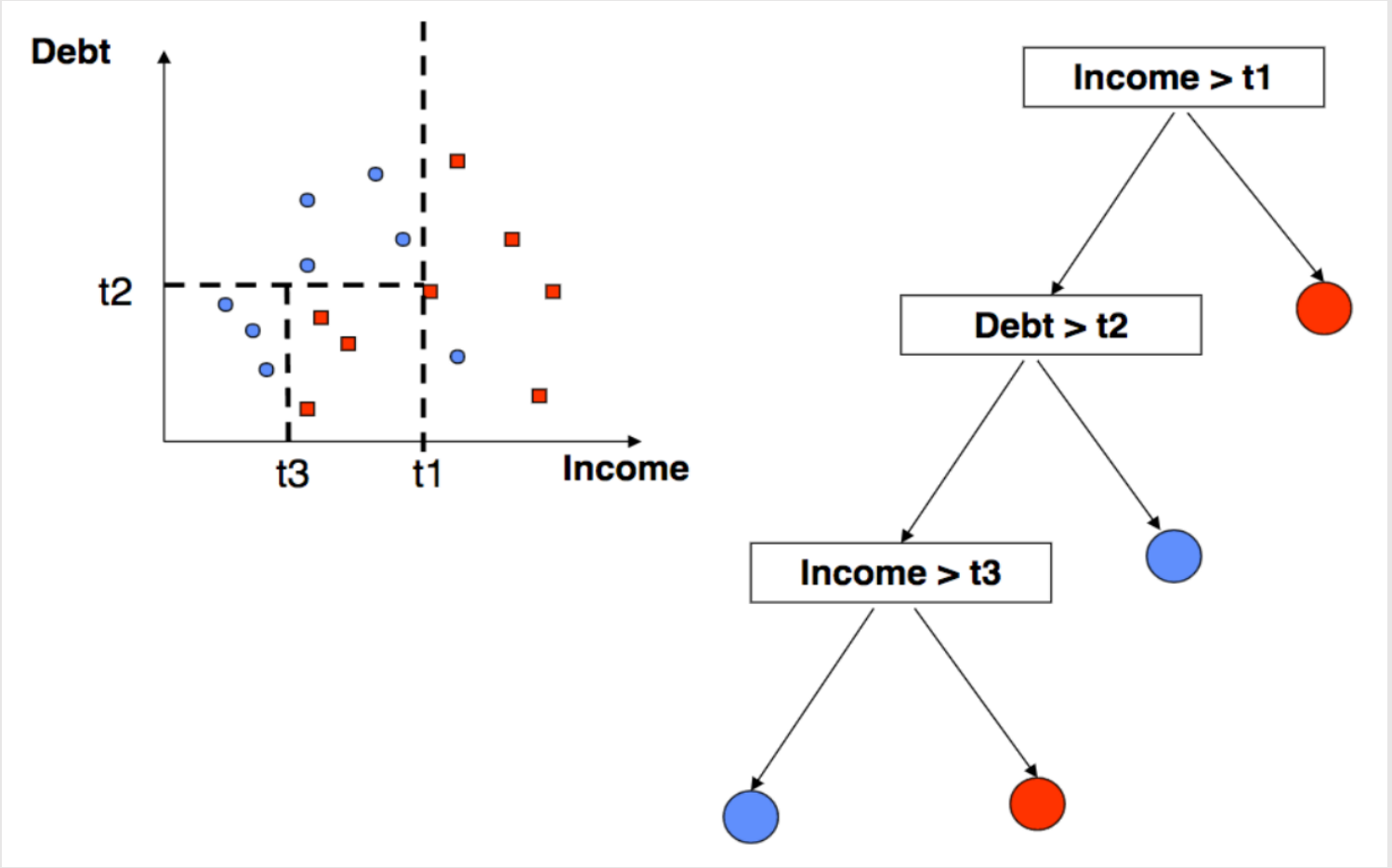
Decision tree parameters



- Decision Tree Parameters
  - Tree depth = $\theta 1$
  - Tree size = $\theta 2$

When to stop partitioning

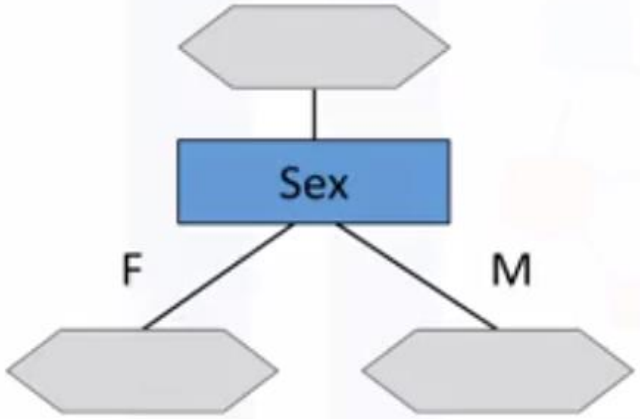
1) All (or X% of) samples have same class label in each node
2) Number of samples in node reaches minimum
3) Change in impurity measure is smaller than threshold
4) Max tree depth is reached
5) Others…



Which attribute should be selected?

S: [9 B, 5 A]
E = - p(B)log(p(B)) - p(A)log(p(A))
E = - (9/14)log(9/14) - (5/14)log(5/14)
E = 0.940

S: [9 B, 5 A]
E = 0.940

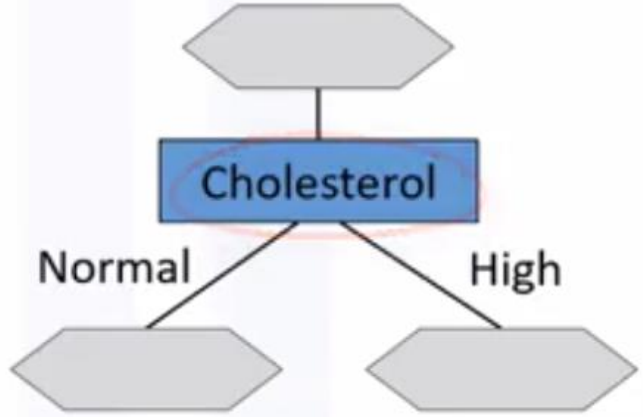


S: [3 B, 4 A]
E = 0.985

S: [6 B, 1 A]
E = 0.592

Patient ID	Age	Sex	BP	Cholesterol	Drug
p1	Young	F	High	Normal	Drug A
p2	Young	F	High	High	Drug A
p3	Middle-age	F	Hiigh	Normal	Drug B
p4	Senior	F	Normal	Normal	Drug B
p5	Senior	M	Low	Normal	Drug B
p6	Senior	M	Low	High	Drug A
p7	Middle-age	M	Low	High	Drug B
p8	Young	F	Normal	Normal	Drug A
p9	Young	M	Low	Normal	Drug B
p10	Senior	M	Normal	Normal	Drug B
p11	Young	M	Normal	High	Drug B
p12	Middle-age	F	Normal	High	Drug B
p13	Middle-age	M	High	Normal	Drug B
p14	Senior	F	Normal	High	Drug A

S: [9 B, 5 A]
E = 0.940



S: [6 B, 2 A]
E = 0.811

S: [3 B, 3 A]
E = 1.00

Gain (s, Sex)
= 0.940 – [(7/14)0.985 + (7/14)0.592]
=0.151

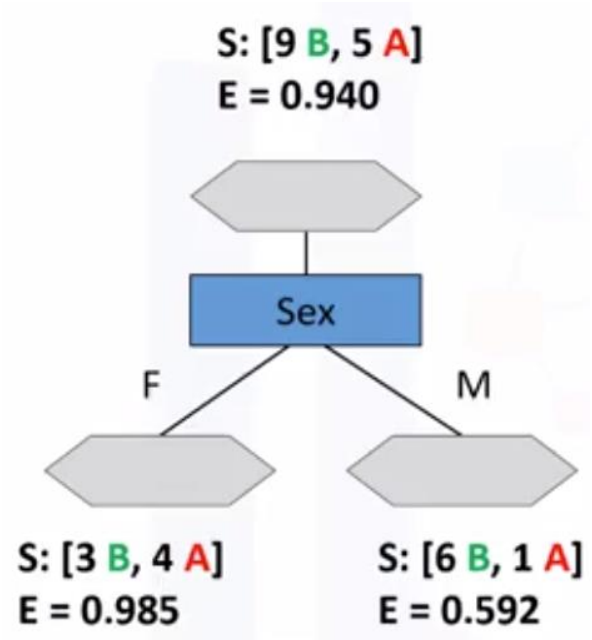
?

$$\text{Impurity}(S) = -\sum_{c=1}^{N} p_c \cdot \log_2 p_c$$

Gain (s, Cholesterol)
= 0.940 – [(8/14).811 + (6/14)1.0]
=0.048

Which attribute should be selected?

$$\text{Impurity}(S) = -\sum_{c=1}^{N} p_c \cdot \log_2 p_c$$



- **What should be next splitting rule for Female between Age or Cholesterol?**
  - **Entropy (Age)**
  - **Entropy (Cholesterol)**
  - **Information Gain (Age)**
  - **Information Gain (Cholesterol)**