Lancaster University

SCC361: Artificial Intelligence

Week 4: Clustering and Classification Techniques 2

Hierarchical Clustering and Decision Trees

Dr Bryan M. Williams
School of Computing and Communications, Lancaster University
Office: InfoLab21 C46 Email: b.williams6@lancaster.ac.uk

Be sure to check in to all timetabled sessions using Attendance Check-in

To check in:

- Check the **Attendance Hub** in iLancaster
- Click **Check In**
- Wait for the "You are checked in" confirmation page
- Here is a the demo

Please DO NOT leave a timetabled session without your attendance being registered

Summary
Error Measures

Lancaster University

Summary

- Choice of metric depends on problem, data and what you want to know
- There are many more error metrics to consider, depending on subject and problem

Non-categorical labels:

$$Accuracy = \|l - \hat{l}\|_2$$

Can use other suitable distance function

Categorical labels:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

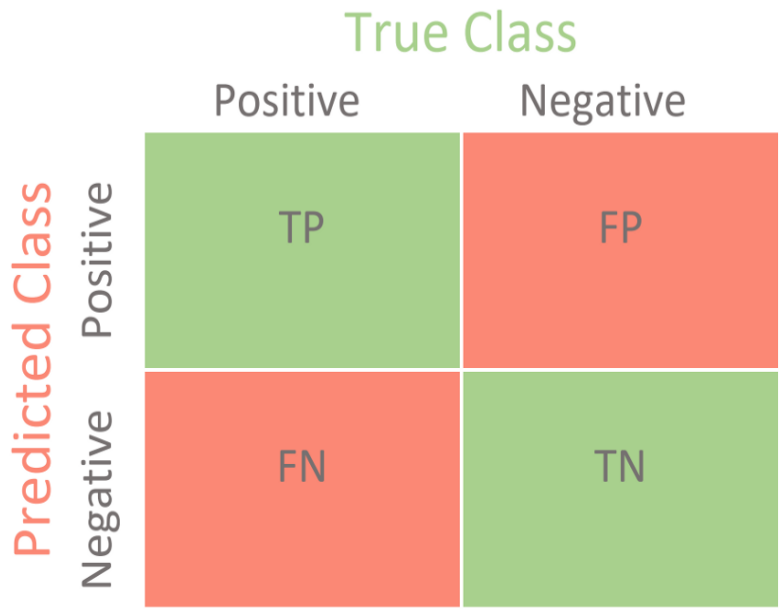
$$Sensitivity = \frac{TP}{TP + FN}$$

$$Specificity = \frac{TN}{TN + FP}$$

$$Precision = \frac{TP}{TP + FP}$$

$$NPV = \frac{TN}{TN + FN}$$

$$F_1 = \frac{2TP}{2TP + FP + FN}$$



Example: Clinical Test

Cup to disc ratio

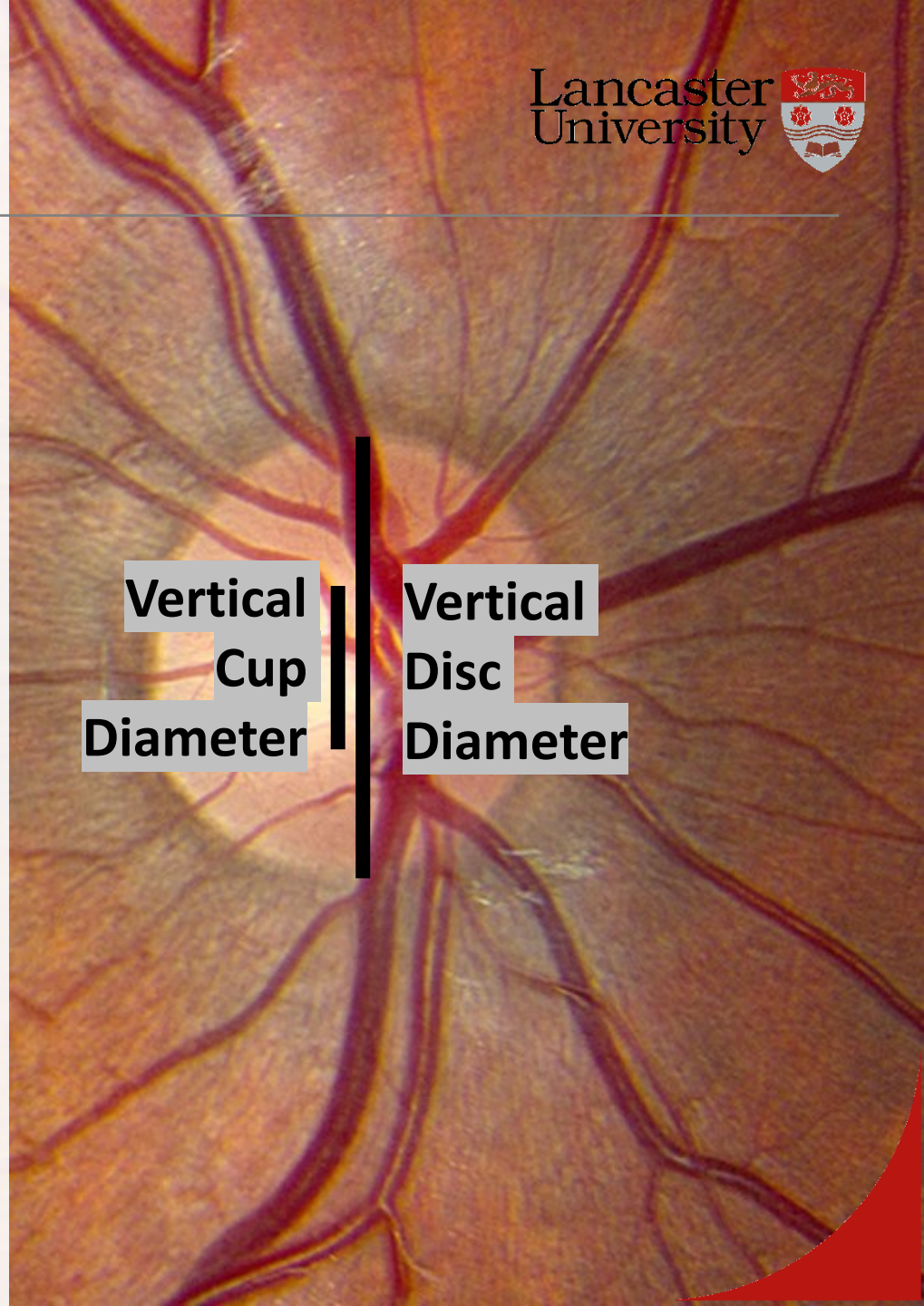
$$cdr = \frac{vertical\ cup\ diameter}{vertical\ disc\ diameter}$$

Rule:

$$cdr \geq 0.7 \Rightarrow Glaucoma$$

$$cdr < 0.7 \Rightarrow Healthy$$

Is this any good?



Example: Clinical Test

Test: 650 Cases (168 GL, 482 Healthy)

Threshold of 0.7:

Accuracy: 0.78

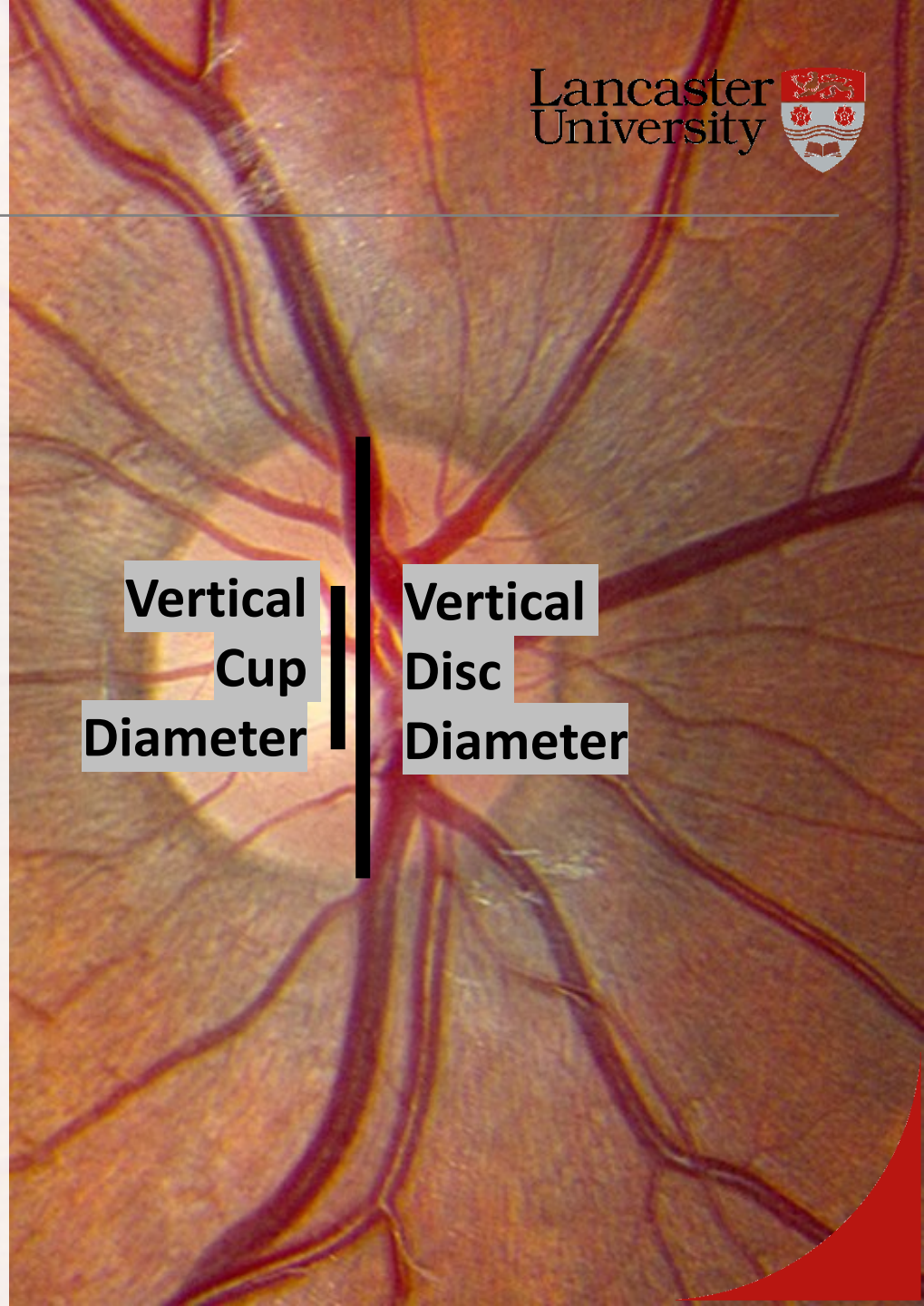
Sensitivity: 0.35

Specificity: 0.94

Precision: 0.66

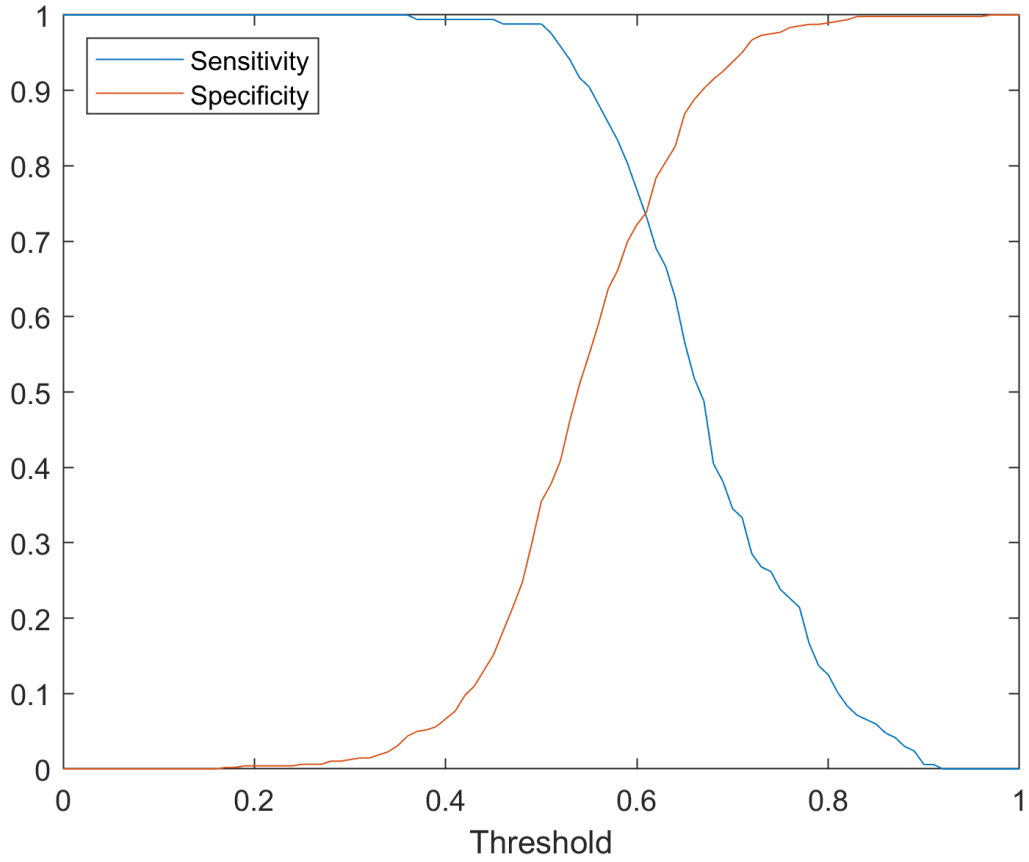
F_1 Score: 0.45

Predicted Class \ True Class	Positive	Negative
Positive	59	29
Negative	109	453



Example: Clinical Test

What if we vary the threshold?



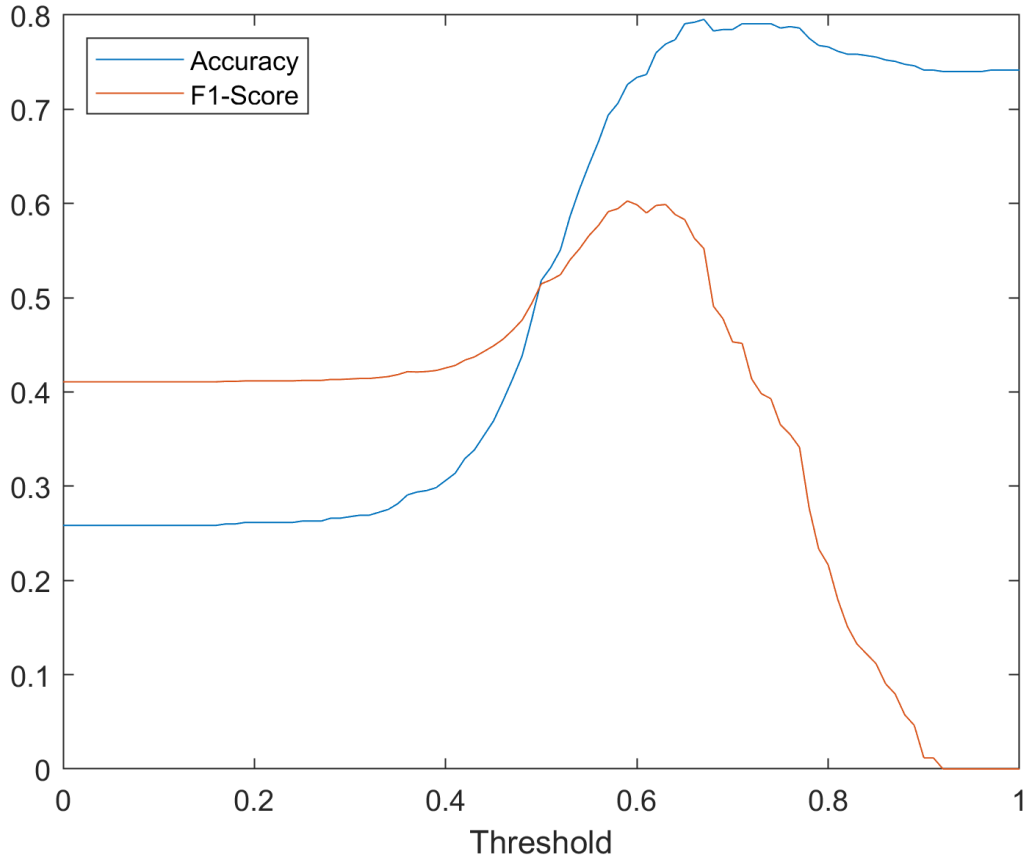


Image credit: Zeba Patel, BSc

Lancaster University

Example: Clinical Test

So how to choose threshold?

Best Accuracy	**Best F_1 Score**	**Best Youden Index**	**Best Precision**
Threshold: 0.67	Threshold: 0.59	Threshold: 0.59	Threshold: 0.83
Accuracy: 0.80	Accuracy: 0.73	Accuracy: 0.73	Accuracy: 0.76
Sensitivity: 0.49	Sensitivity: 0.80	Sensitivity: 0.80	Sensitivity: 0.07
Specificity: 0.90	Specificity: 0.70	Specificity: 0.70	Specificity: 1.00
Precision: 0.64	Precision: 0.48	Precision: 0.48	Precision: 0.92
F_1 Score: 0.55	F_1 Score: 0.60	F_1 Score: 0.60	F_1 Score: 0.13

At 0.95 sensitivity, we have: 0.41 specificity

Example: Security System

False Acceptance Rate:

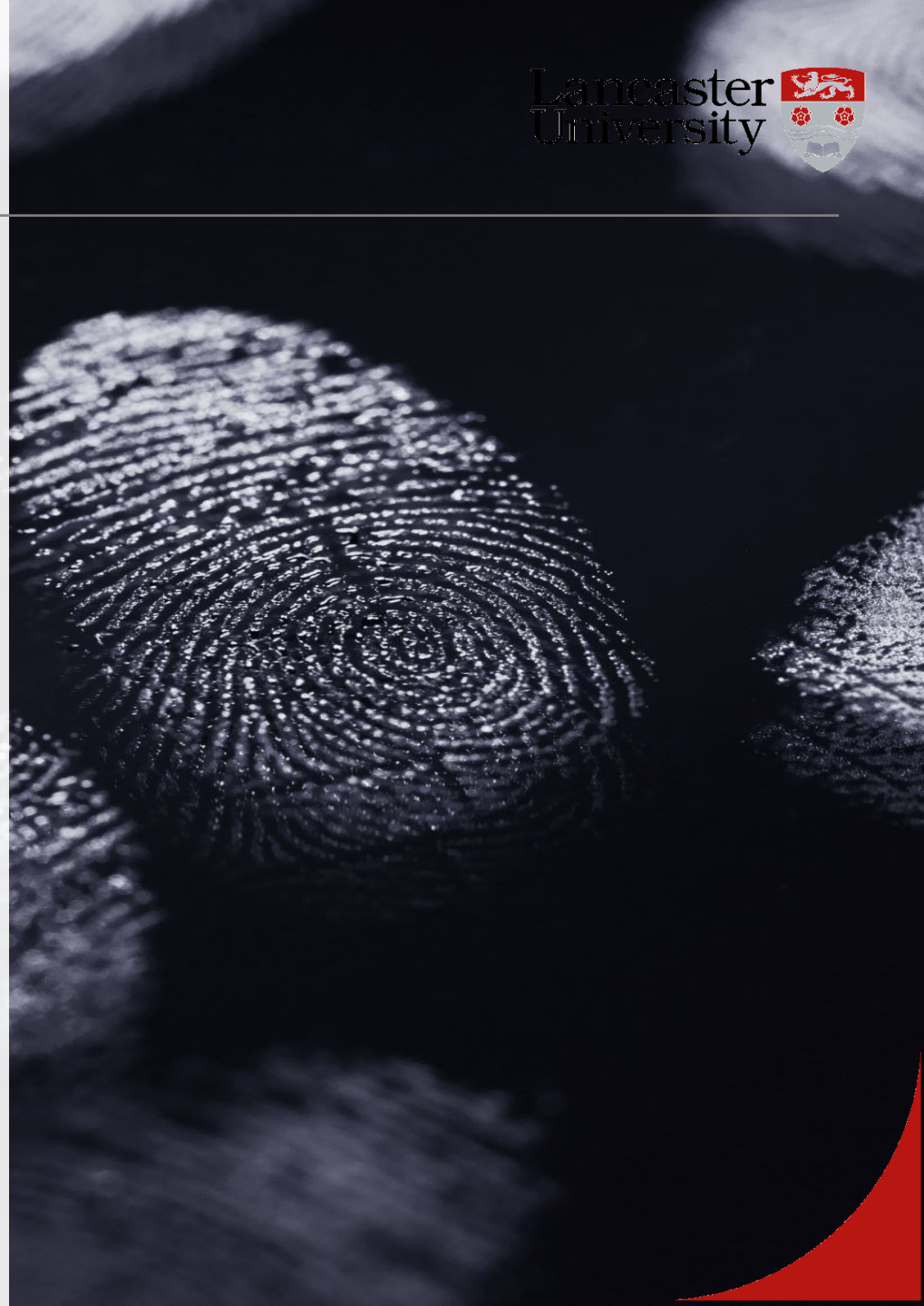
$$FAR = \frac{Incorrect\ Authentication}{Total\ Attempts}$$

False Rejection Rate:

$$FRR = \frac{Incorrect\ Rejection}{Total\ Attempts}$$

Aim: minimise both FAR and FRR.

Equal Error Rate: when FAR = FRR



Example: Security System

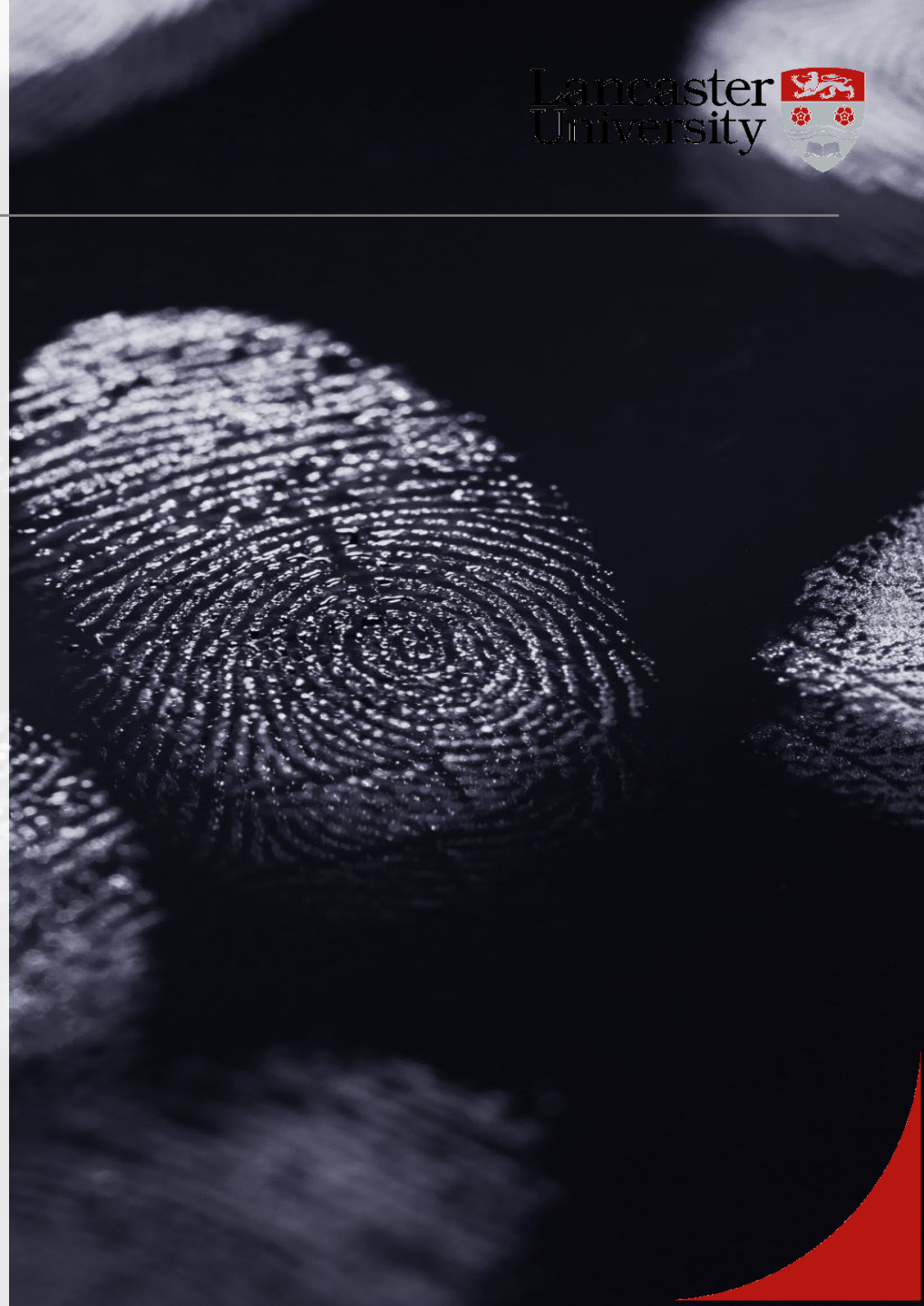
False Acceptance Rate:

If 100 people (excluding you) try to access your phone by fingerprint, what proportion will gain access?

False Rejection Rate:

Using your own phone, what proportion of times will your fingerprint be rejected?

Lower EER => Higher Accuracy

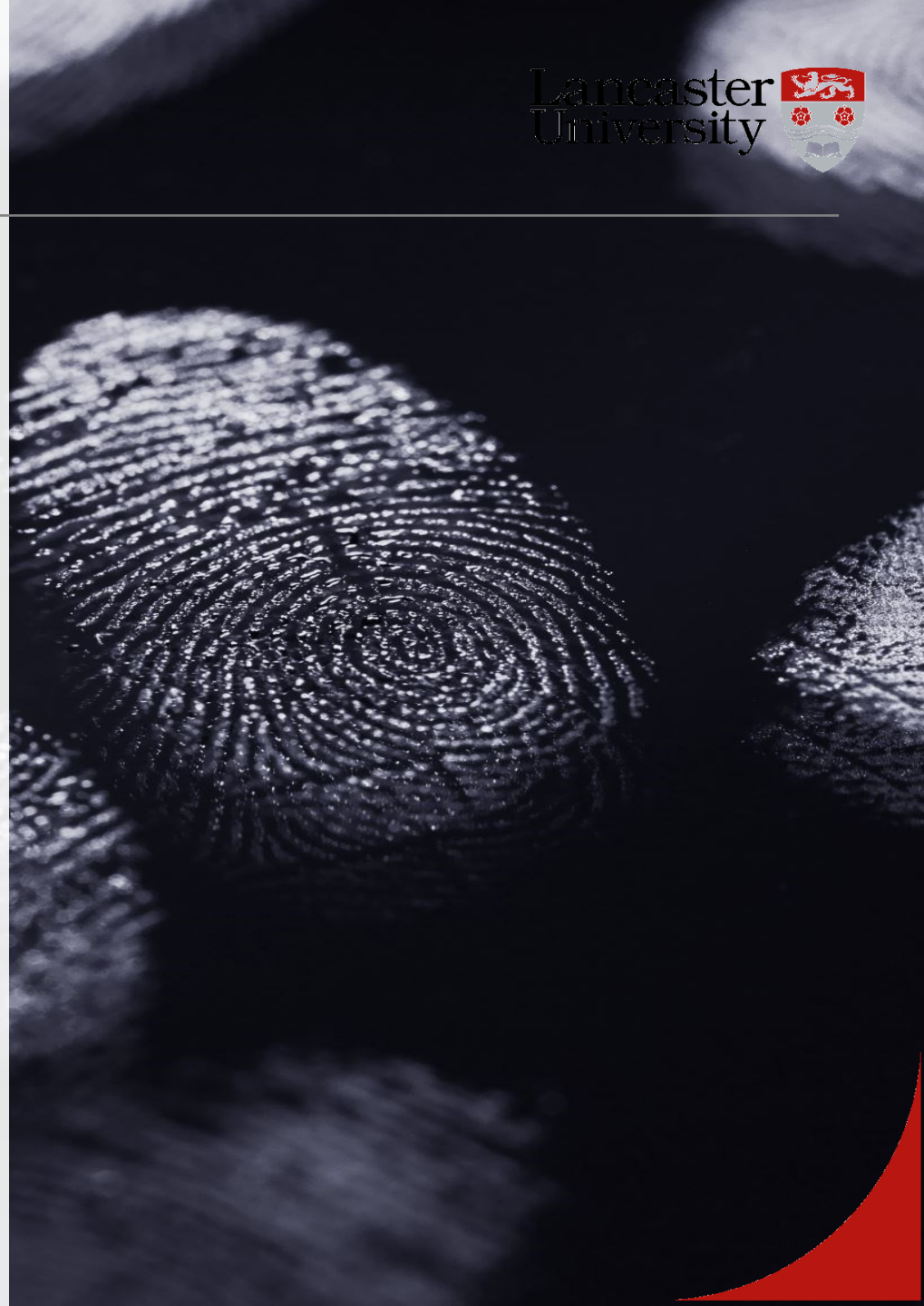


Example: Security System

Experimental Setting:

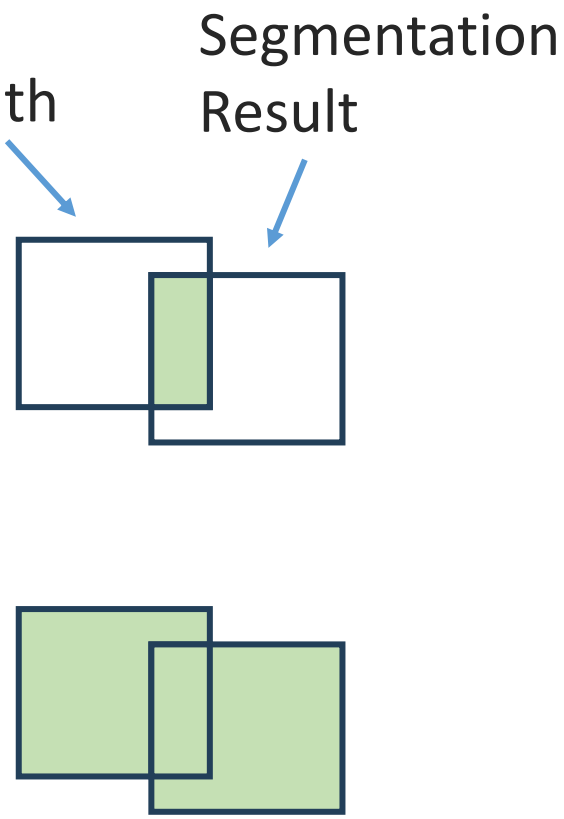
- Several “true” candidates
- Many “false” candidates
- Attempt with a large subset / whole set

Lower EER => Higher Accuracy



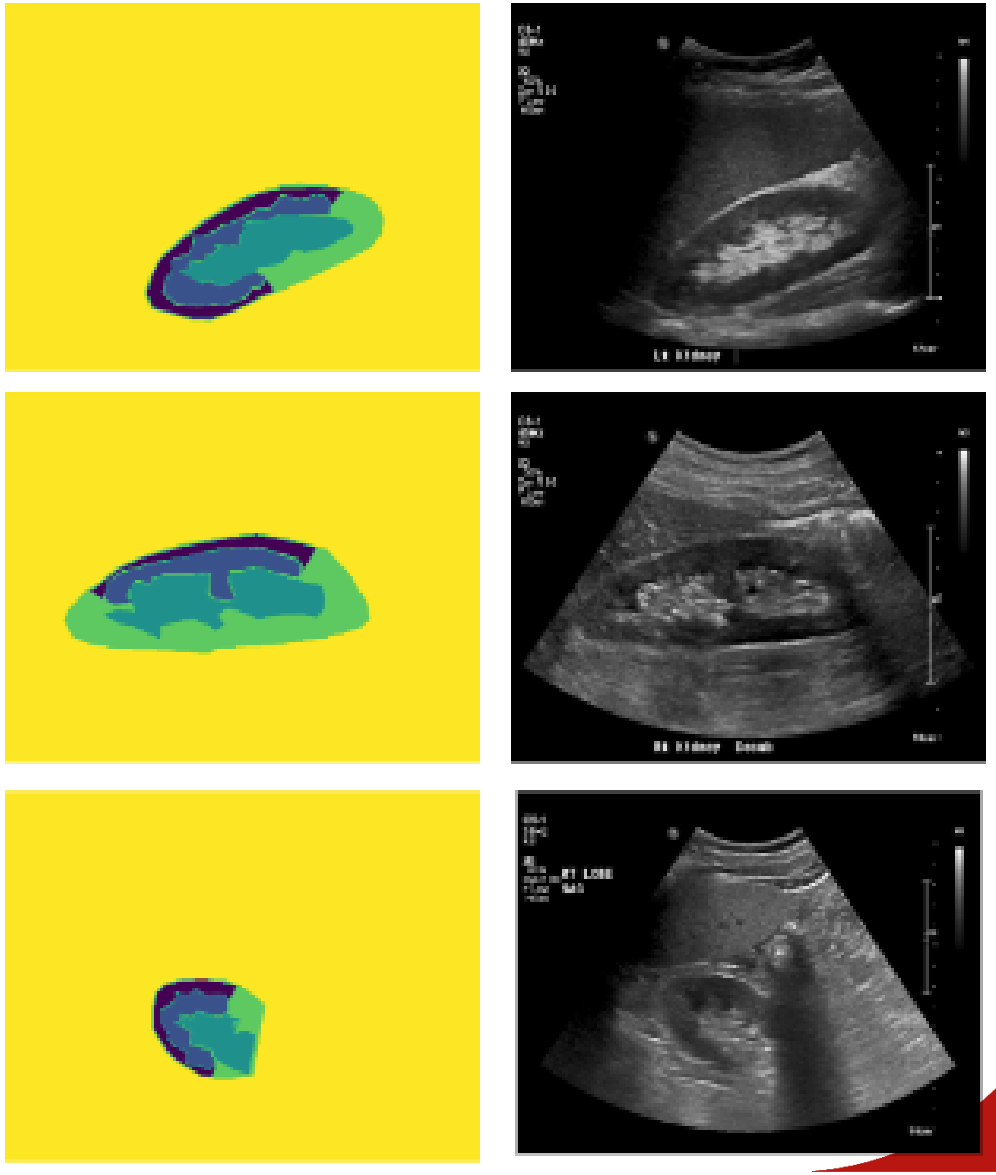
Example: Image Segmentation

IoU (Intersection over Union)



$$IoU = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$

AKA Tanimoto Coefficient / Jaccard Coefficient



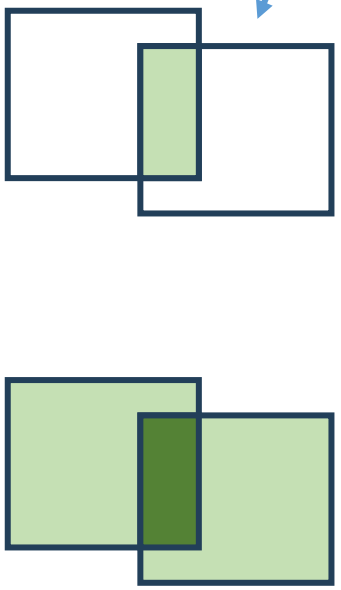
Example: Image Segmentation

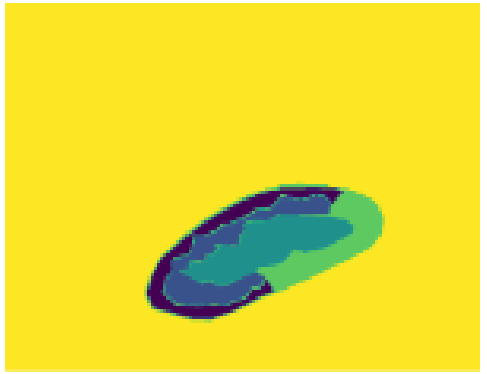
Dice Score

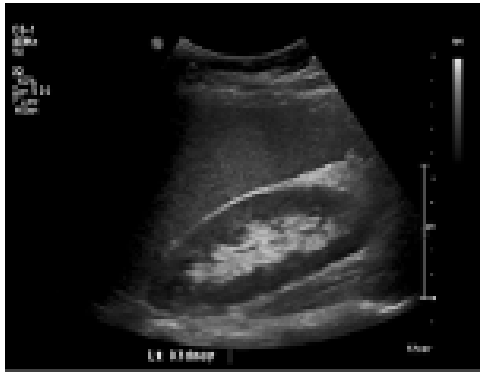
Ground Truth

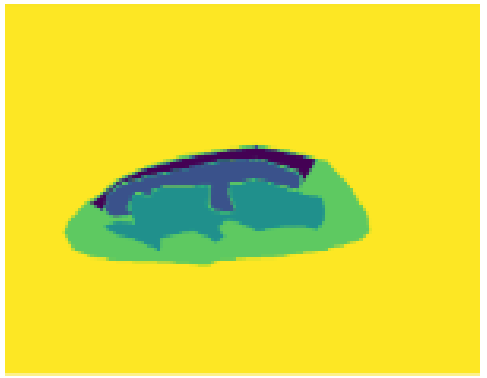
Segmentation Result

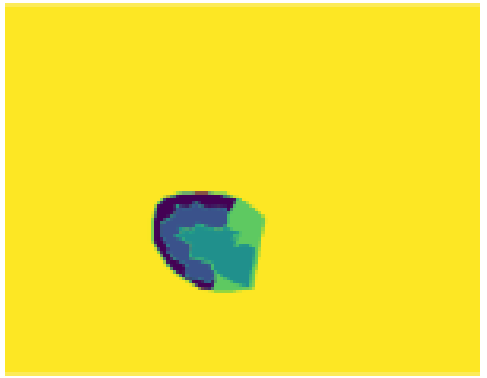
$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Sum of Areas}}$$











Lancaster University

Summary

- Choice of metric depends on problem, data and what you want to know
- There are many more error metrics to consider, depending on subject and problem

Non-categorical labels:

$$Accuracy = \|l - \hat{l}\|_2$$

Can use other suitable distance function

Categorical labels:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

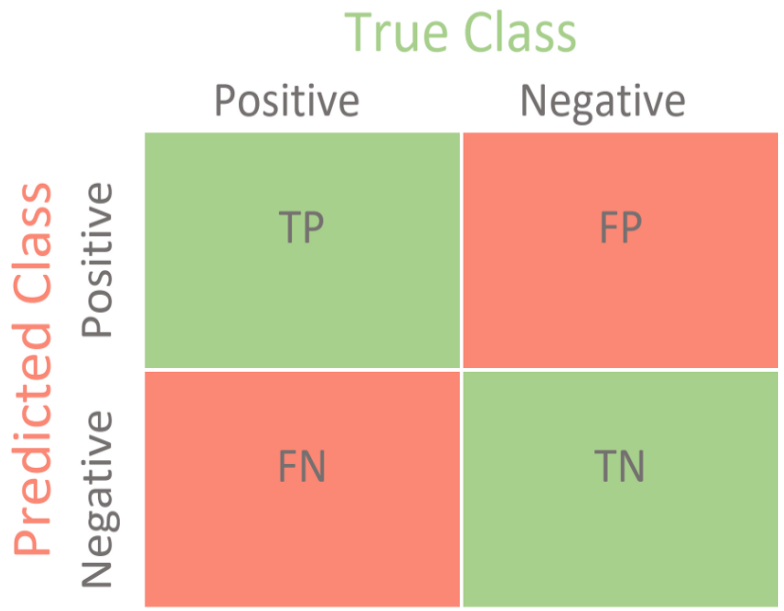
$$Sensitivity = \frac{TP}{TP + FN}$$

$$Specificity = \frac{TN}{TN + FP}$$

$$Precision = \frac{TP}{TP + FP}$$

$$NPV = \frac{TN}{TN + FN}$$

$$F_1 = \frac{2TP}{2TP + FP + FN}$$



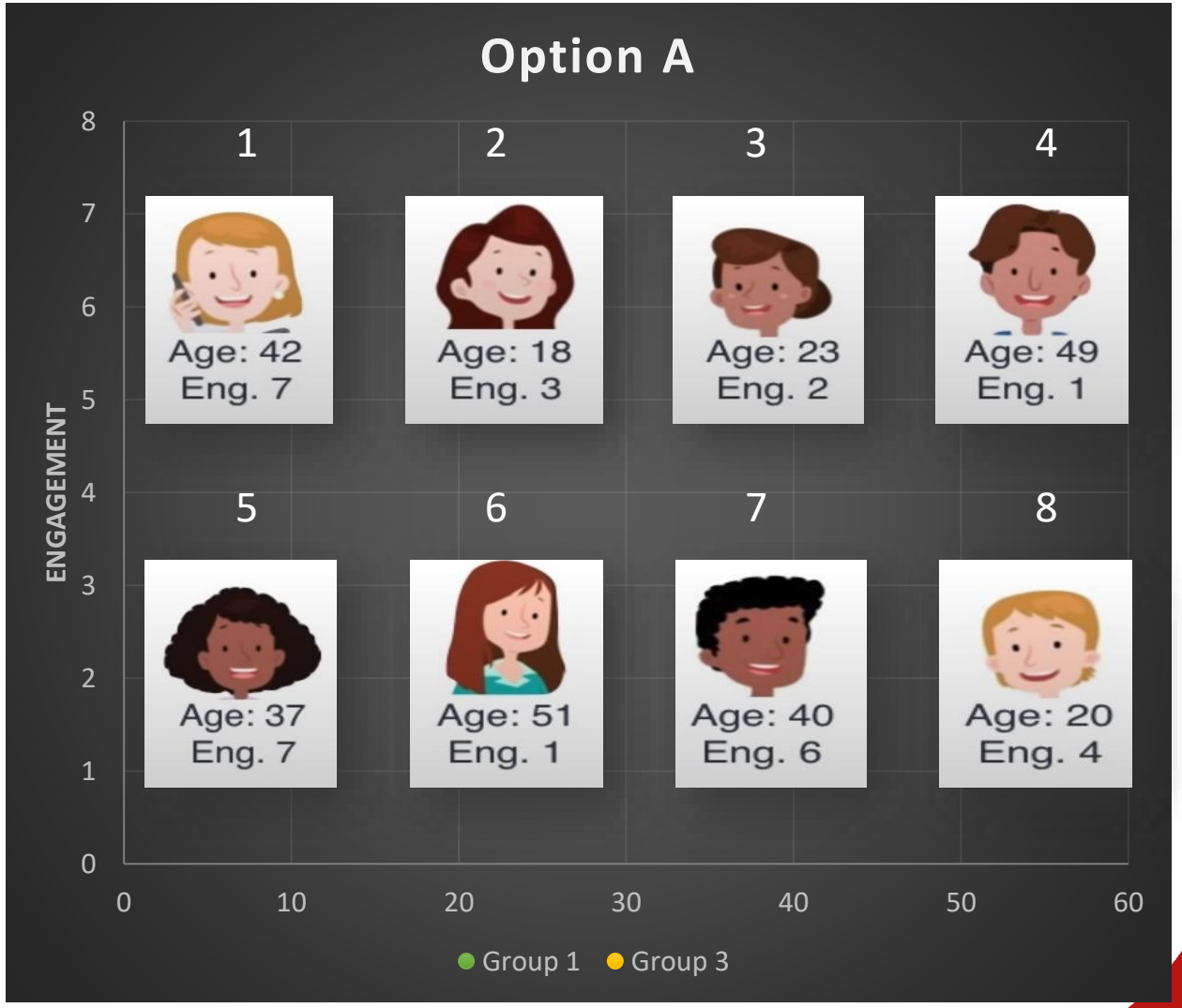
Clustering

Customer Engagement Data

Here are the **ages** (in years) and **engagements** (in days/weeks) of our customers that use our app.

If we have to put them into three groups to effectively serve them, how should we do that?

A	{1,5,6} {4,8} {2,3,7}
B	{1,8,3} {4,7,2} {5,6}
C	{2,8,3} {5,7,1} {4,6}
D	{3,7,8} {4,1} {2,5,6}



Lancaster University

What is Clustering?

Grouping data into “clusters”

Optimisation with constraints:

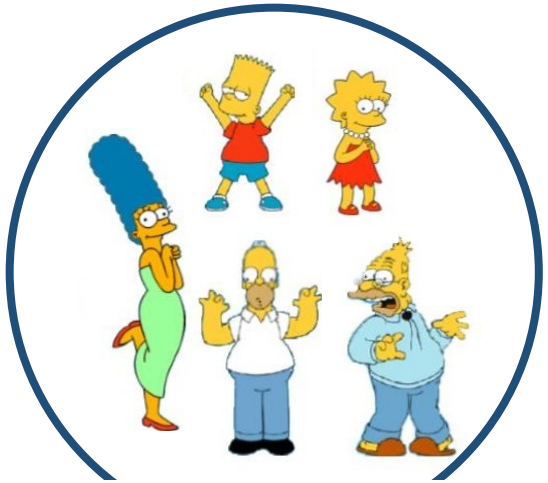
- Number of clusters
- Minimum distance between clusters

Reduce dissimilarity between members in a cluster

Two common methods:

Hierarchical

K-Means



Simpson Family

VS

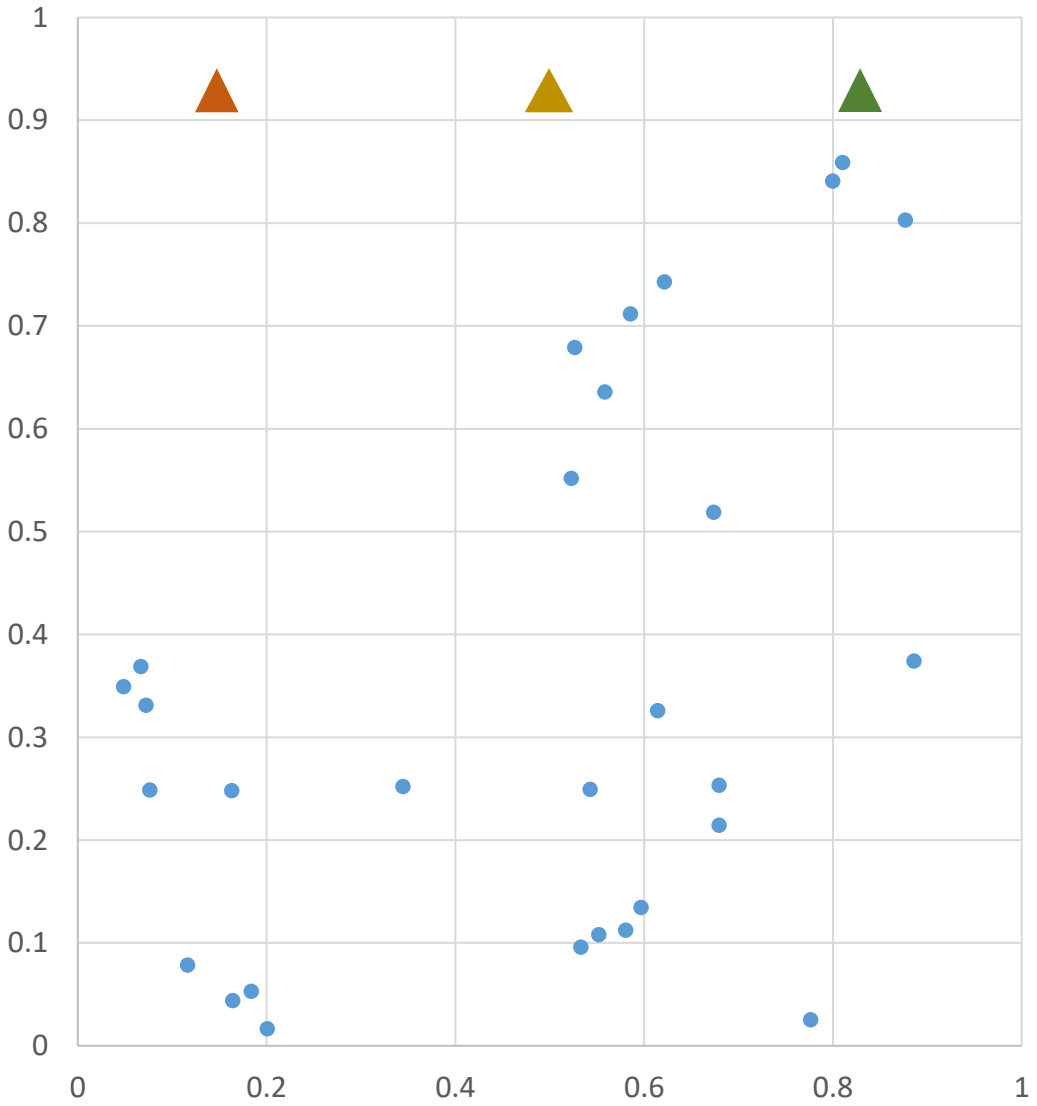
Non-Simpson Family

Lancaster University

Changing the Initialisation

Algorithm:

- Choose k, i.e. the number of clusters
- Place k centroids
- while true:
  - Create k clusters by assigning each point to closest centroid
    - Calculate distance from centroids
    - Find minimum distance
    - Copy label
  - compute k new centroids by averaging points in each cluster
  - if centroids don't change:
    - stop

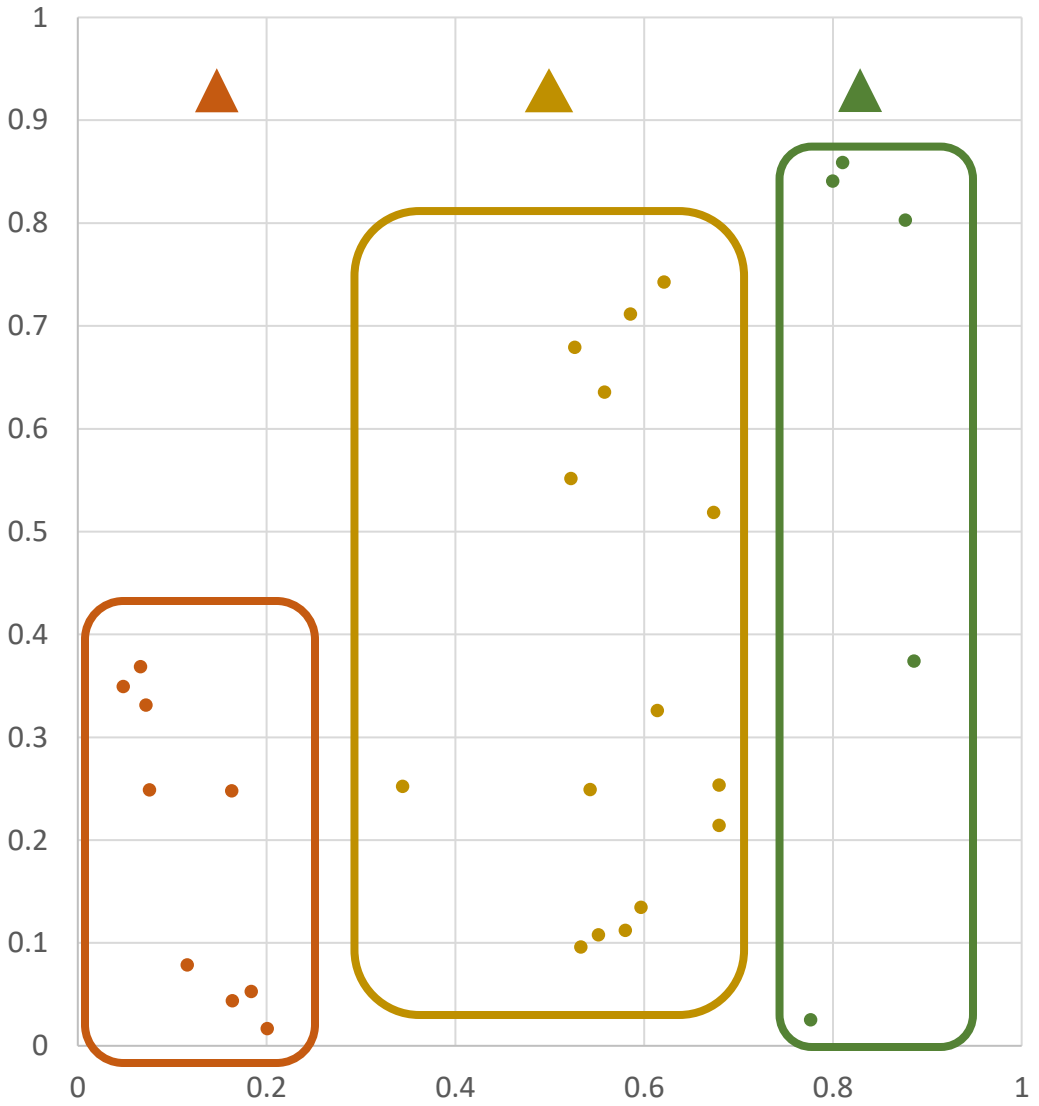


Lancaster University

Changing the Initialisation

Algorithm:

- Choose k, i.e. the number of clusters
- Place k centroids
- while true:
  - Create k clusters by assigning each point to closest centroid
    - Calculate distance from centroids
    - Find minimum distance
    - Copy label
  - compute k new centroids by averaging points in each cluster
  - if centroids don't change:
    - stop

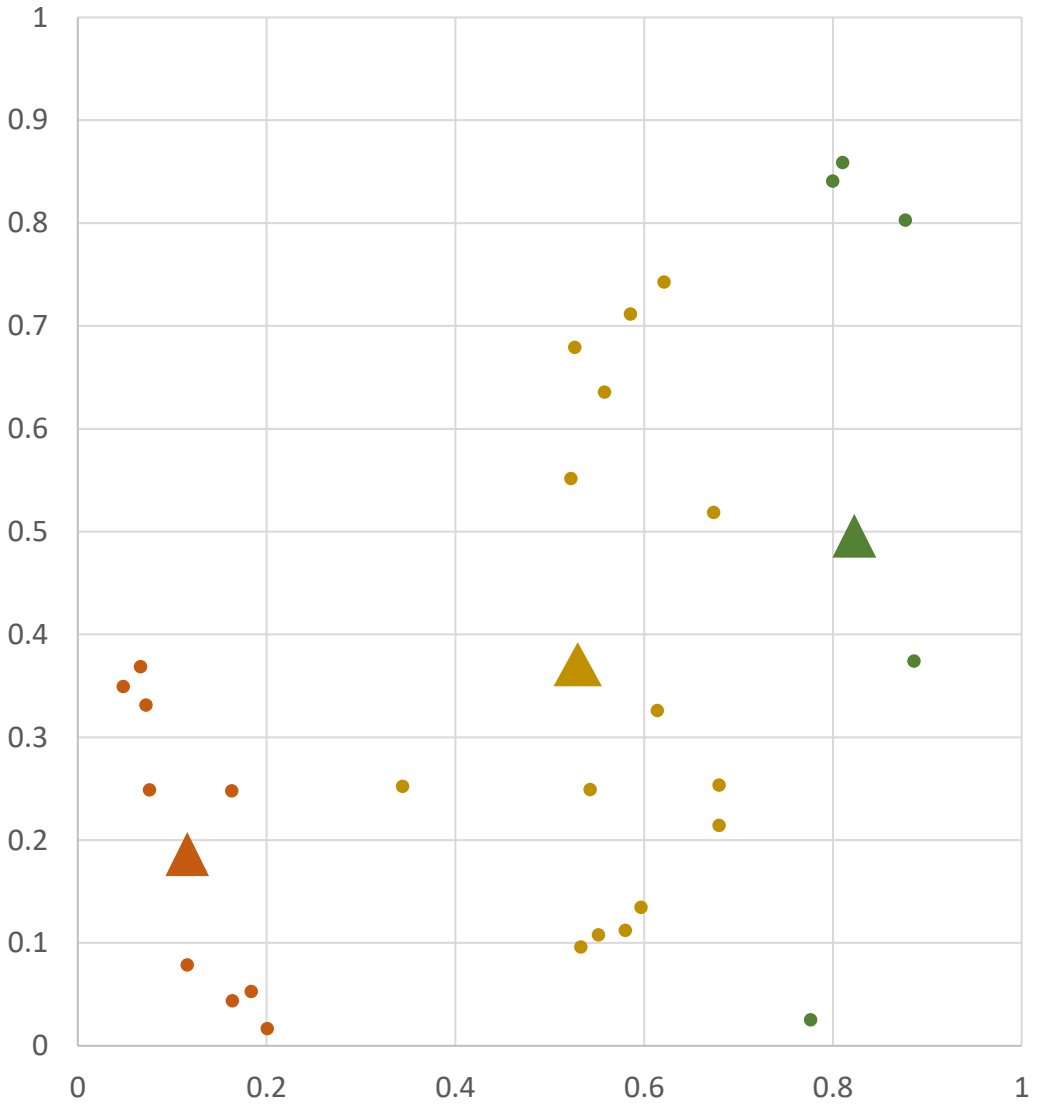


Lancaster University

Changing the Initialisation

Algorithm:

- Choose k, i.e. the number of clusters
- Place k centroids
- while true:
  - Create k clusters by assigning each point to closest centroid
    - Calculate distance from centroids
    - Find minimum distance
    - Copy label
  - compute k new centroids by averaging points in each cluster
  - if centroids don't change:
    - stop

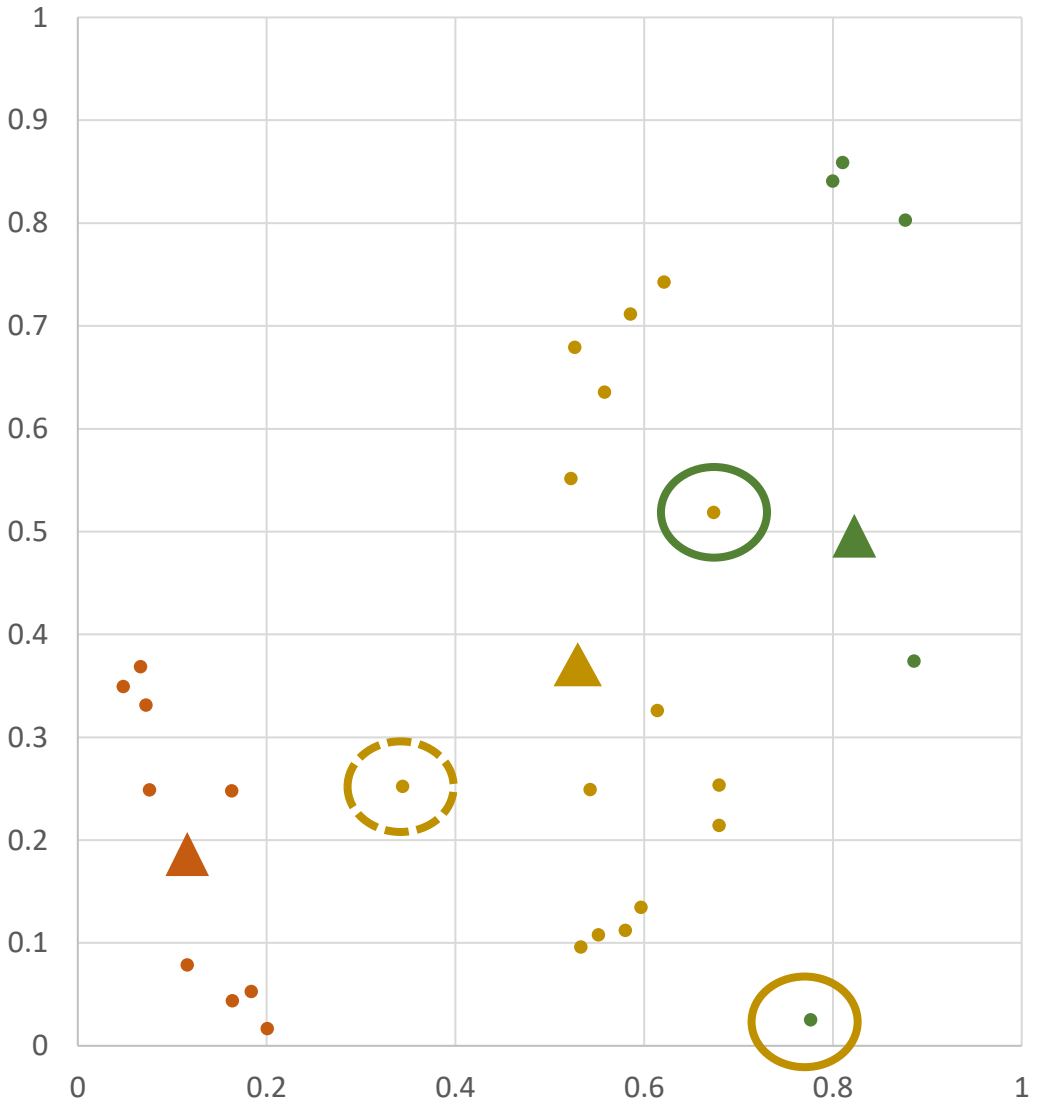


Lancaster University

Changing the Initialisation

Algorithm:

- Choose k, i.e. the number of clusters
- Place k centroids
- while true:
  - Create k clusters by assigning each point to closest centroid
    - Calculate distance from centroids
    - Find minimum distance
    - Copy label
  - compute k new centroids by averaging points in each cluster
  - if centroids don't change:
    - stop

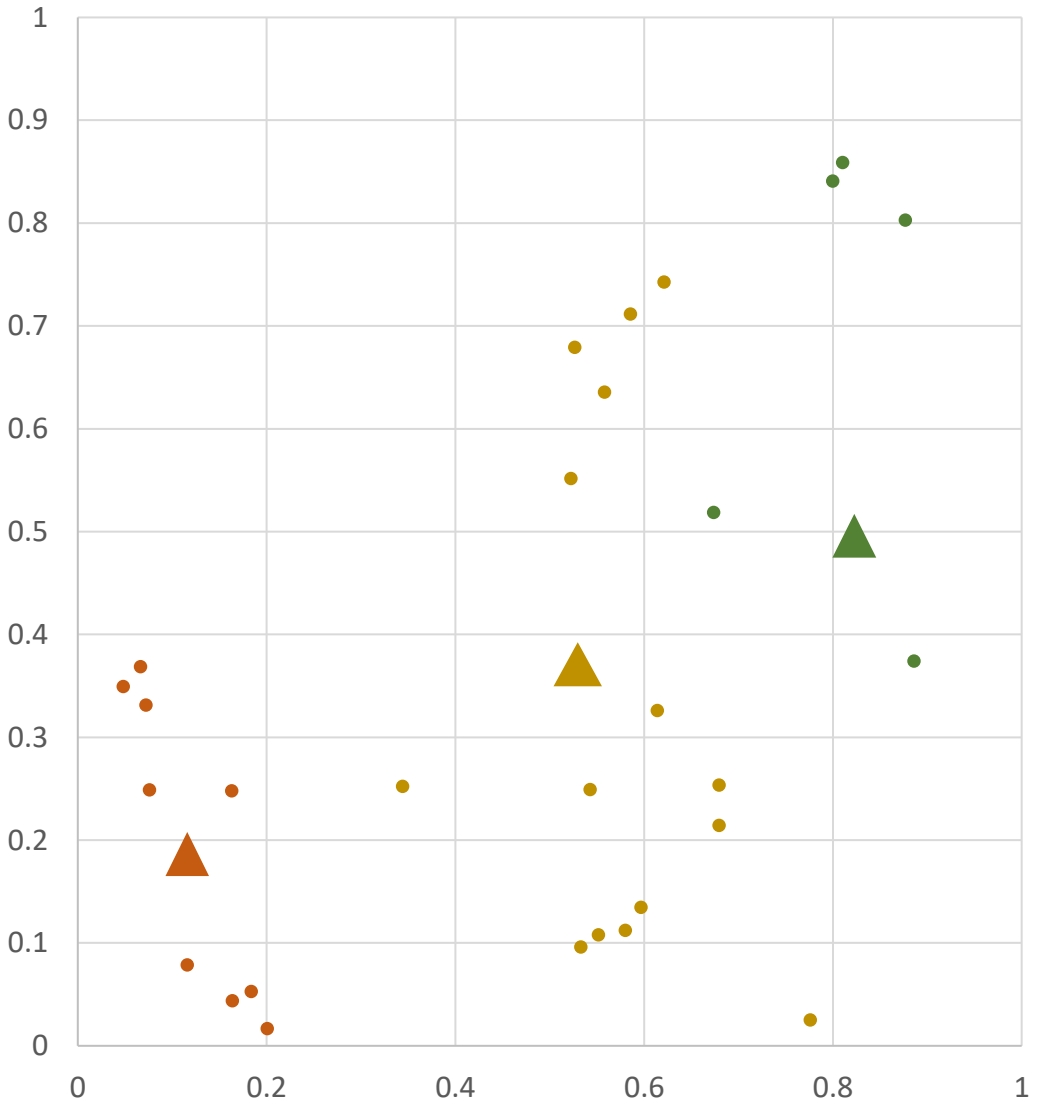


Lancaster University

Changing the Initialisation

Algorithm:

- Choose k, i.e. the number of clusters
- Place k centroids
- while true:
  - Create k clusters by assigning each point to closest centroid
    - Calculate distance from centroids
    - Find minimum distance
    - Copy label
  - compute k new centroids by averaging points in each cluster
  - if centroids don't change:
    - stop

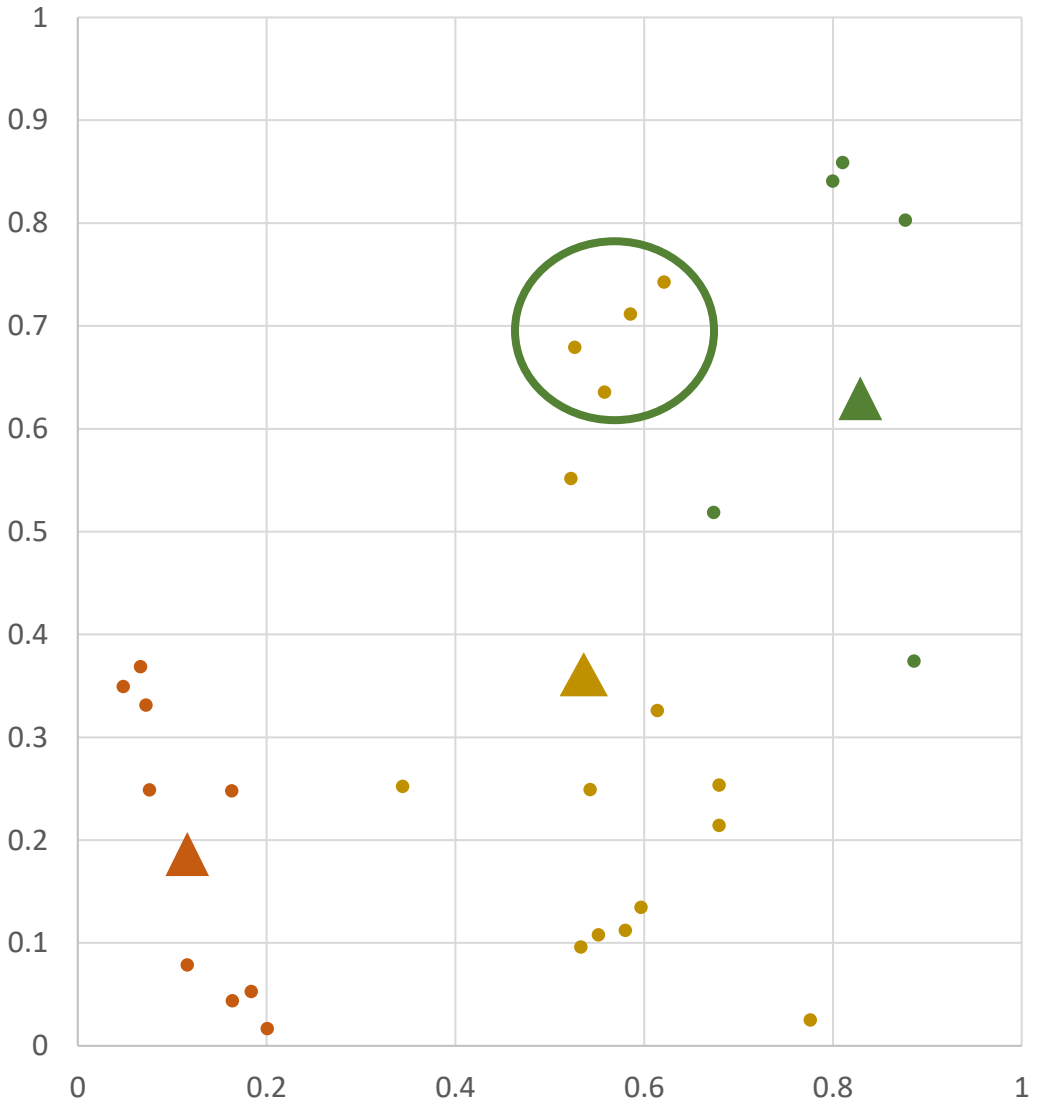


Lancaster University

Changing the Initialisation

Algorithm:

- Choose k, i.e. the number of clusters
- Place k centroids
- while true:
  - Create k clusters by assigning each point to closest centroid
    - Calculate distance from centroids
    - Find minimum distance
    - Copy label
  - compute k new centroids by averaging points in each cluster
  - if centroids don't change:
    - stop

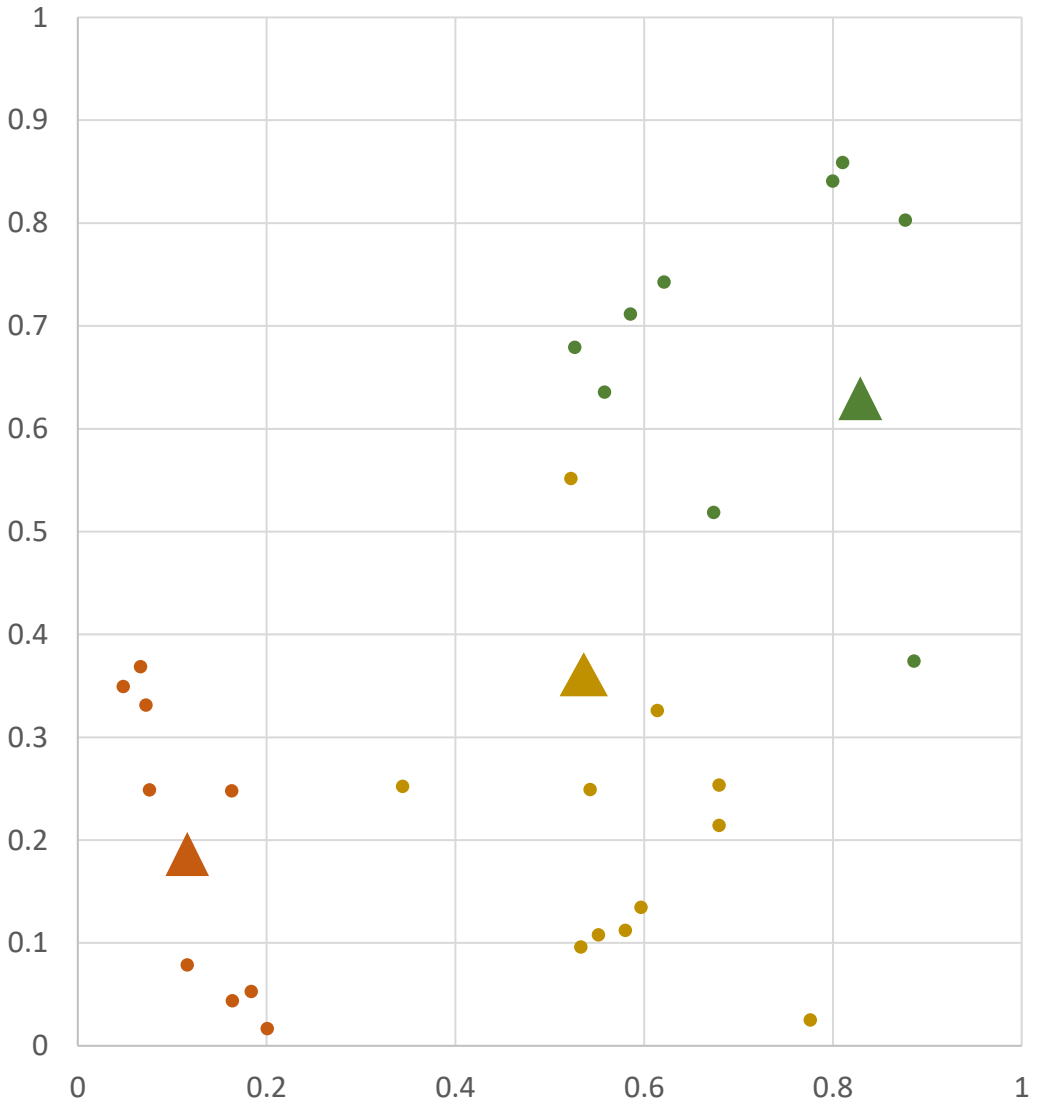


Lancaster University

Changing the Initialisation

Algorithm:

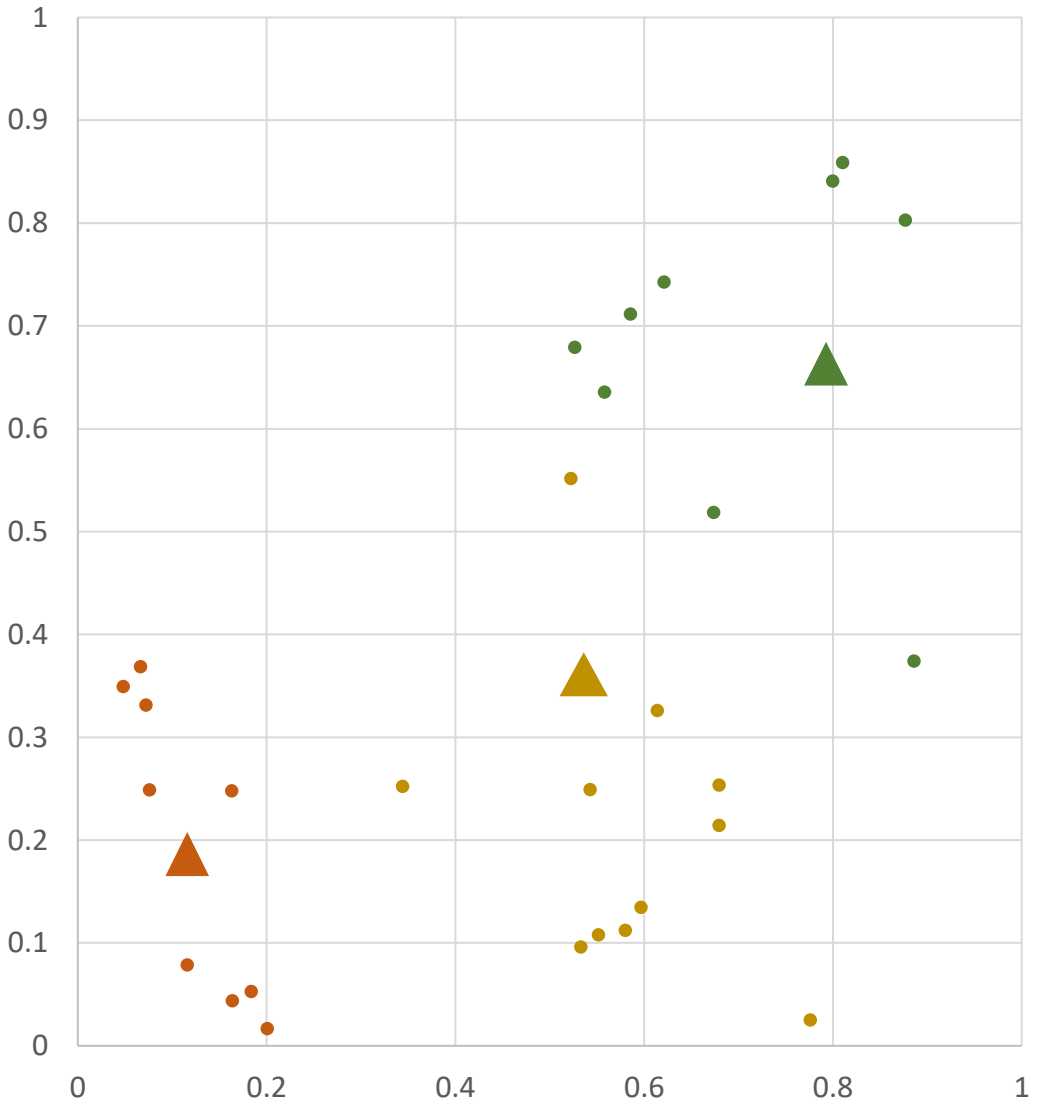
- Choose k, i.e. the number of clusters
- Place k centroids
- while true:
  - Create k clusters by assigning each point to closest centroid
    - Calculate distance from centroids
    - Find minimum distance
    - Copy label
  - compute k new centroids by averaging points in each cluster
  - if centroids don't change:
    - stop



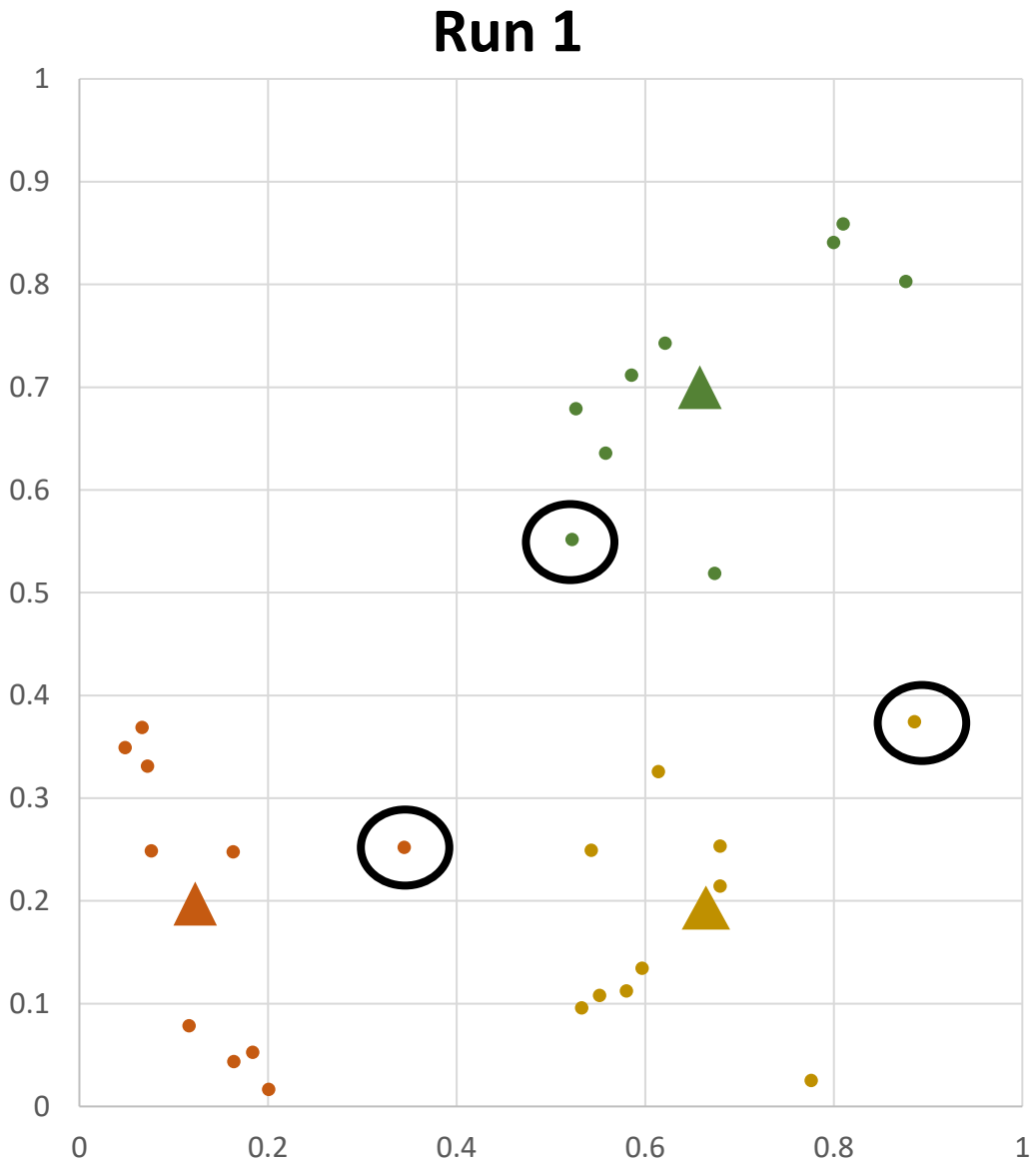
Changing the Initialisation

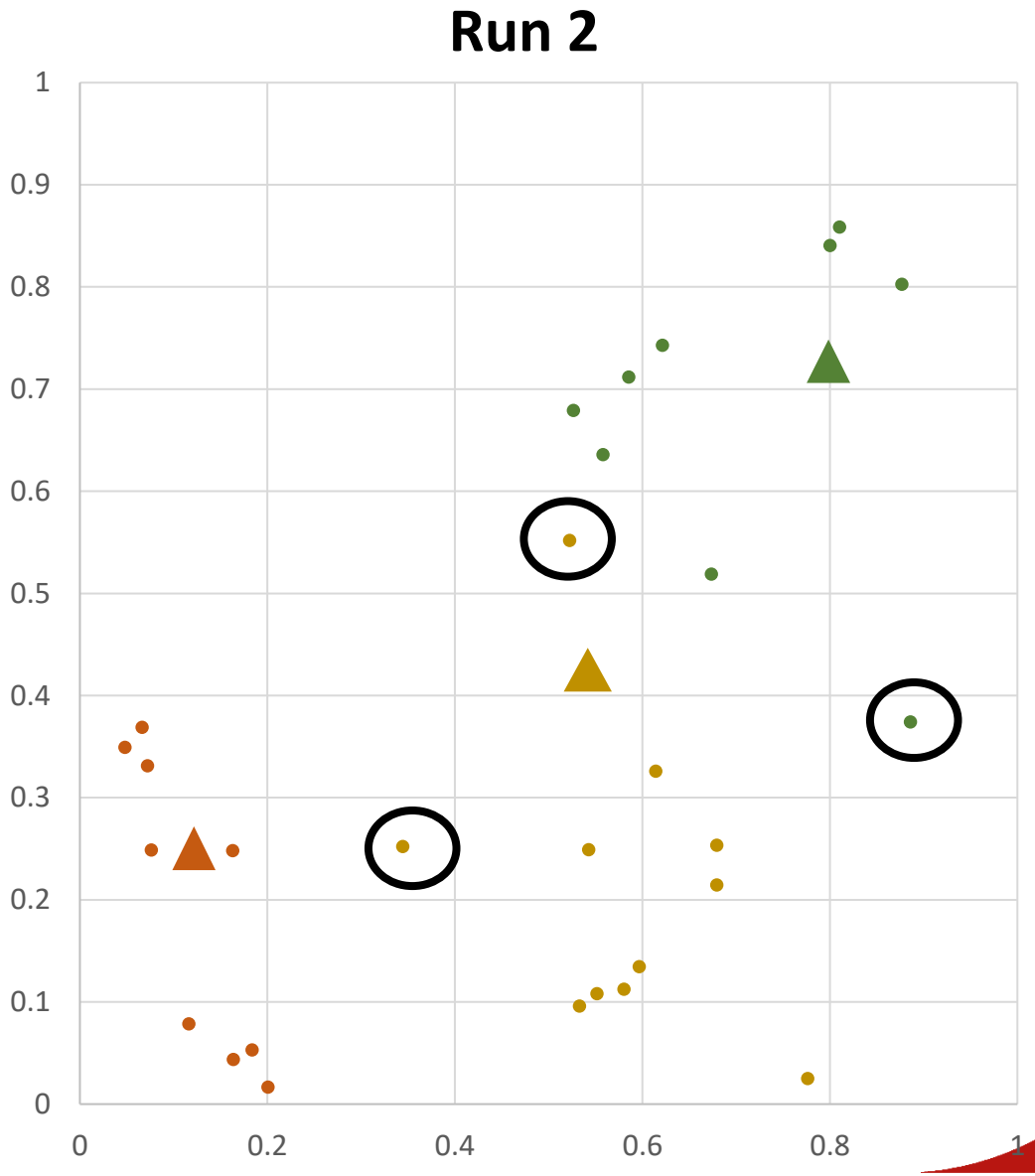
Algorithm:

- Choose k, i.e. the number of clusters
- Place k centroids
- while true:
  - Create k clusters by assigning each point to closest centroid
    - Calculate distance from centroids
    - Find minimum distance
    - Copy label
  - compute k new centroids by averaging points in each cluster
  - if centroids don't change:
    - stop



Changing the Initialisation





Lancaster University

K-Means Summary

Efficiency

- K-Mean is efficient but has some weaknesses

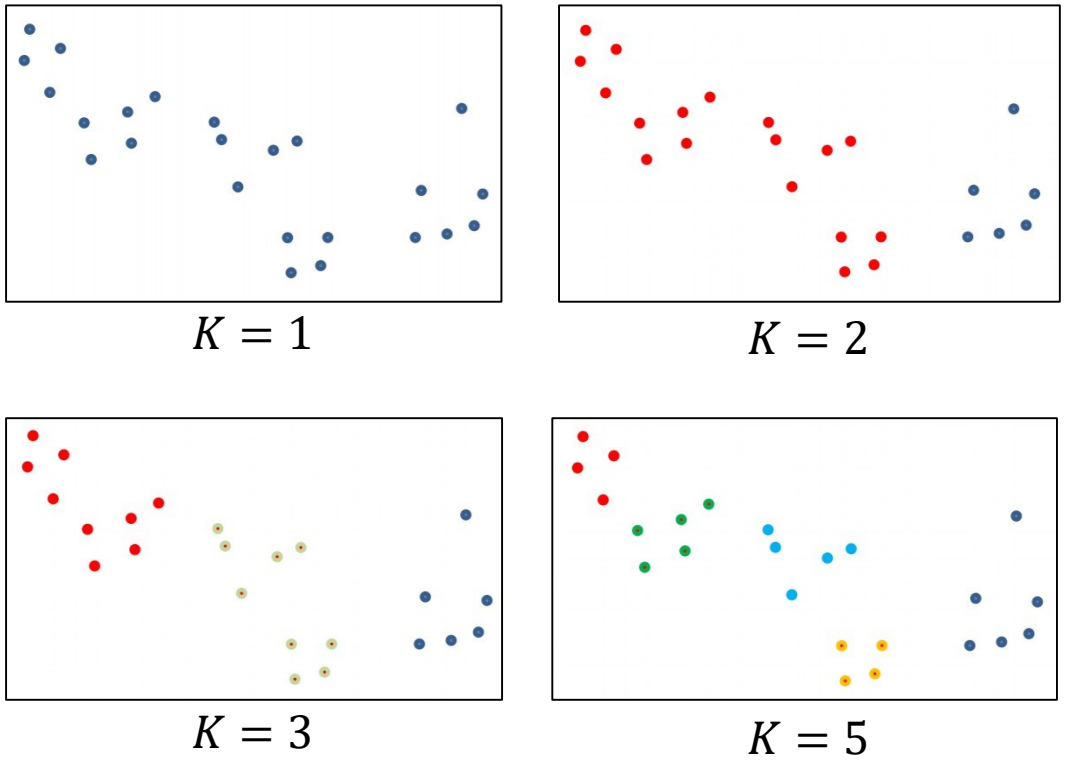
K-Means Summary

Efficiency

- K-Mean is efficient but has some weaknesses

Number of Clusters

- You don't necessarily know $\boldsymbol{k}$, i.e. number of clusters
- You can choose "wrong" k and get strange results.



How do we choose the right k?

K-Means Summary

Efficiency

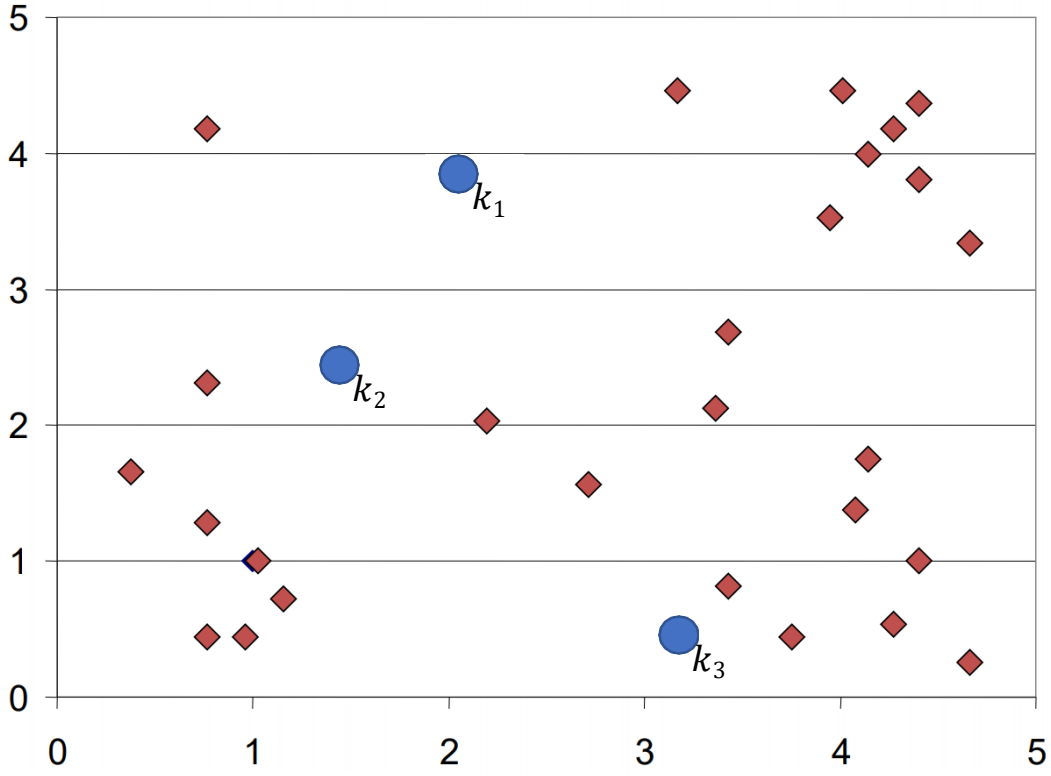
- K-Mean is efficient but has some weaknesses

Number of Clusters

- You don't necessarily know $\boldsymbol{k}$, i.e. number of clusters
- You can choose "wrong" k and get strange results.

It is non-deterministic

- Initial centroids are chosen at random



K-Means Summary

Efficiency

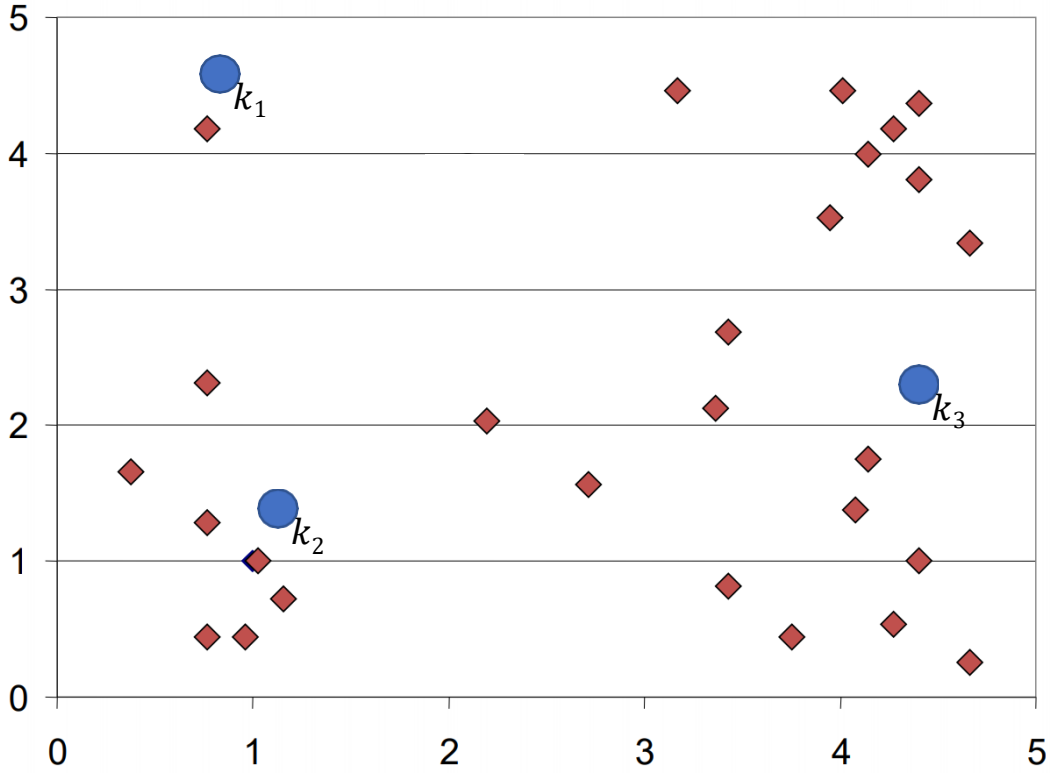
- K-Mean is efficient but has some weaknesses

Number of Clusters

- You don't necessarily know $\boldsymbol{k}$, i.e. number of clusters
- You can choose "wrong" k and get strange results.

It is non-deterministic

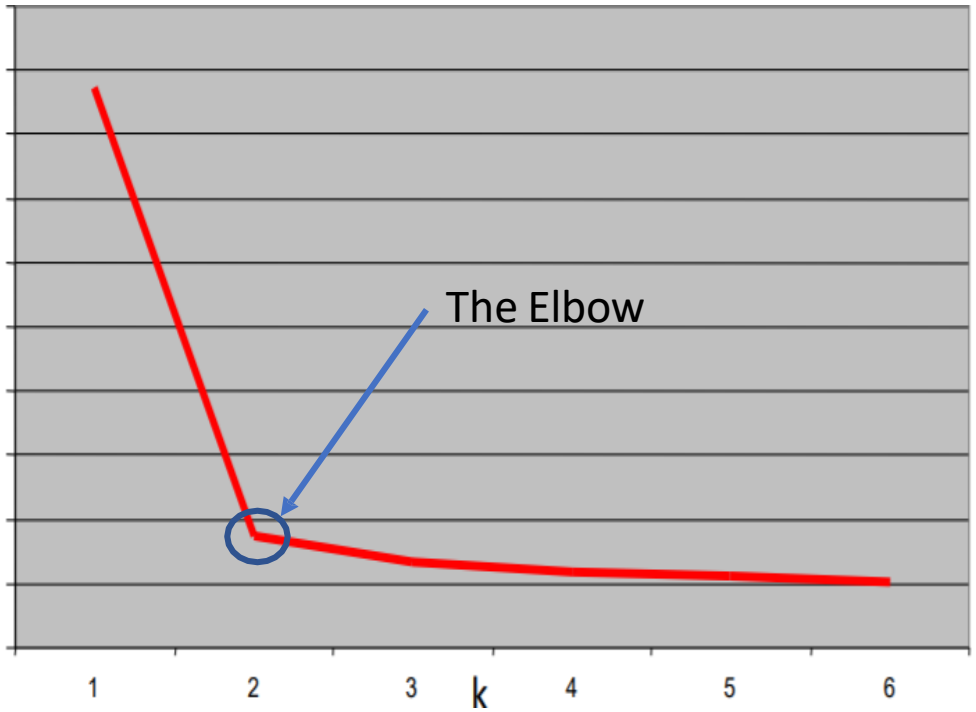
- Initial centroids are chosen at random
- Centroids can be too close



K-Means Summary

How can we choose k ?

- A prior knowledge of the data space can help
  - Three classes of flowers in the Iris dataset
  - Two types of emails: good and spam
- Use the Elbow method
  - Try different values and look for abrupt change in result
- Run hierarchical clustering on subset of data



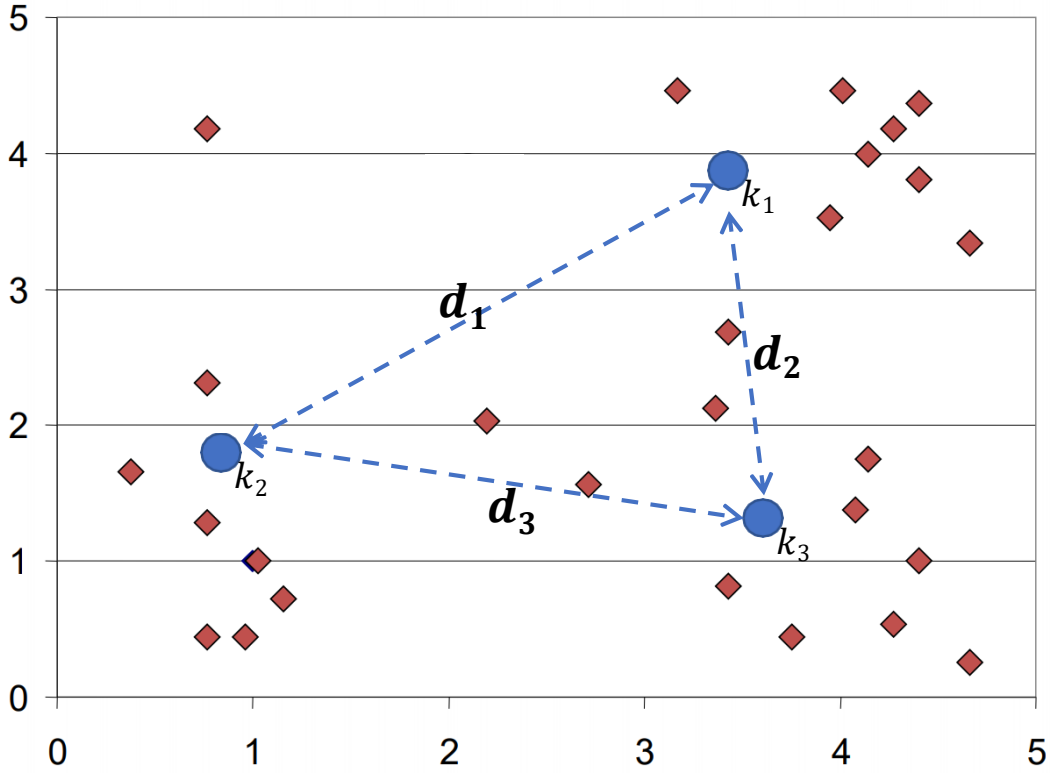
K-Means Summary

Mitigating Initial Centroids Dependency

- Use a random number seed
- Define a minimum distance $\min(d)$ between clusters:

$$d_1, d_2, d_3 \geq \min(d)$$

- Define the minimum data points in a cluster.



Hierarchical Clustering

Lancaster University

What is Clustering?

How do we group these characters?

Hierarchical Clustering

E.g. Family Tree

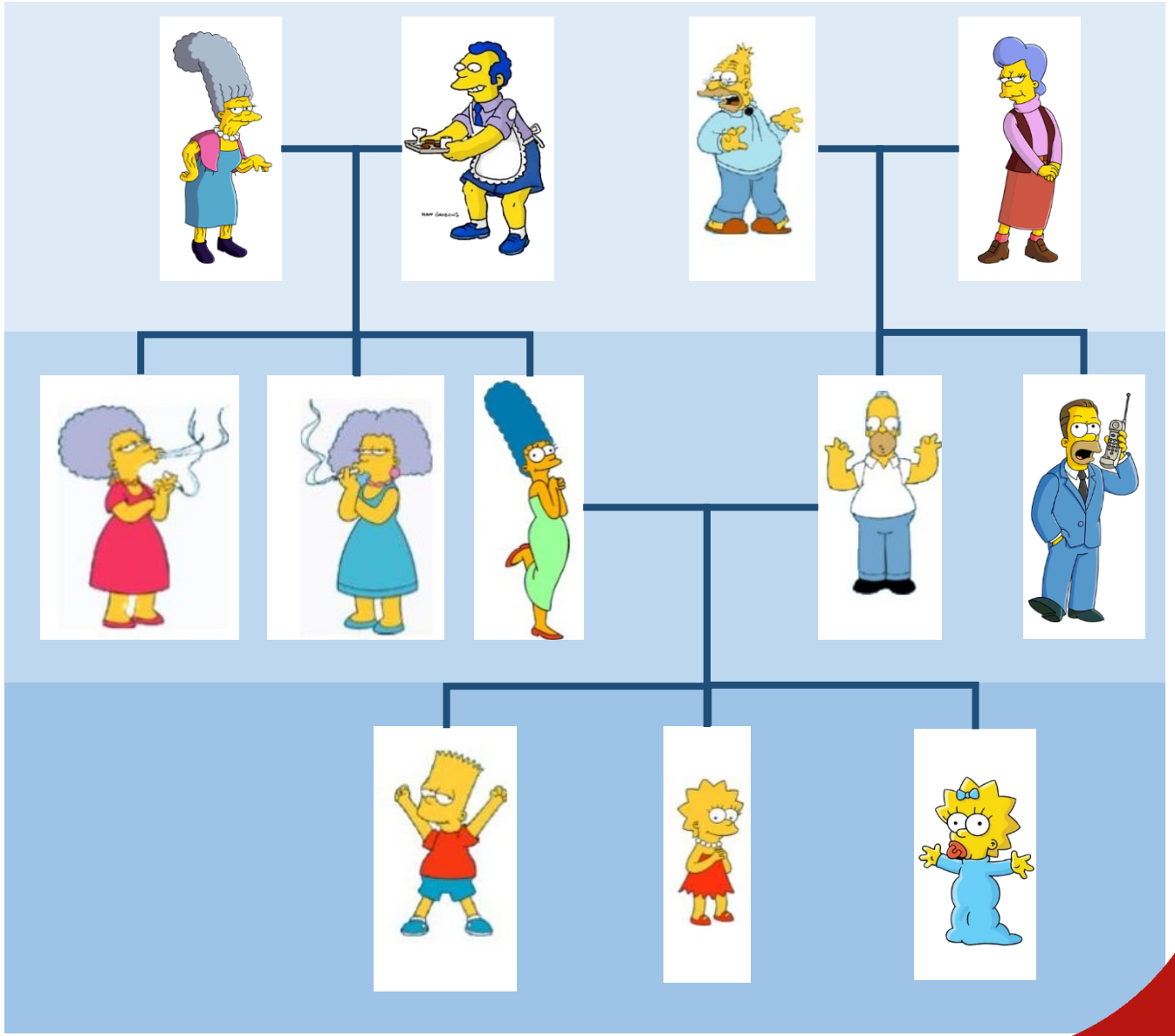
Two common approaches:

Agglomerative (bottom-up)

- Each observation starts in own cluster.
- Pairs of clusters are merged as you move up the hierarchy.

Divisive (top-down)

- All observations start in one cluster.
- Splits are performed recursively as you move down the hierarchy.



Hierarchical Clustering

Aim: Build a hierarchy of clusters

1: Initialisation

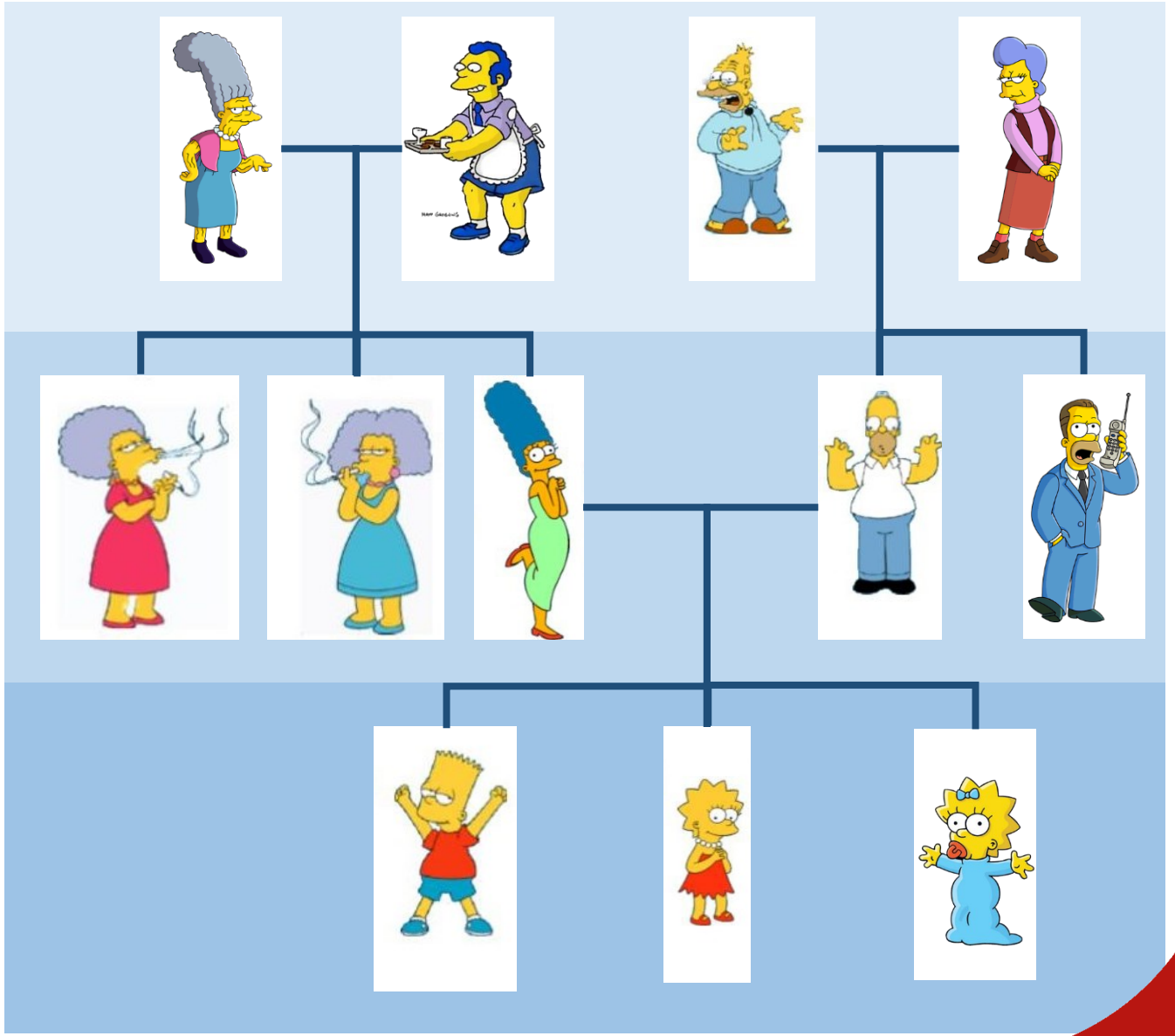
- Assign each item to a cluster
- n items results in n clusters containing 1 item each
- Here, $n = 12$

2: Merge iteratively

- Find and merge the **closest** pair of clusters into a single cluster within a range
- Continue until you have a single cluster of n items

Result

- Hierarchical decomposition of the dataset



Hierarchical Clustering – Example 1

Distance Measure

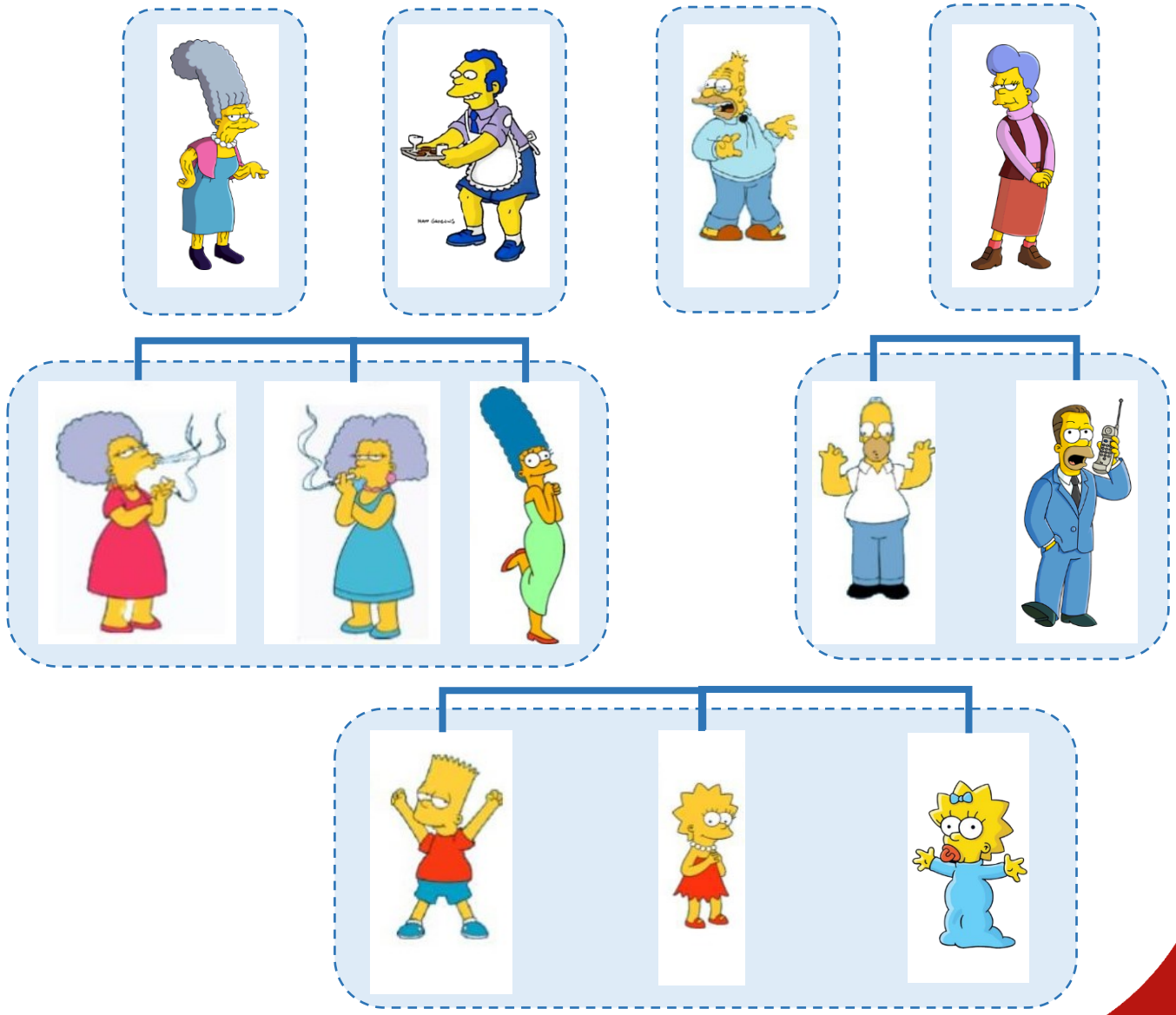
Relationship	Value
Sibling	1
Spouse	2
Parent/ Child	3

Hierarchical Clustering – Example 1

Distance Measure

Relationship	Value
Sibling	1
Spouse	2
Parent/ Child	3

Level 1: Sibling



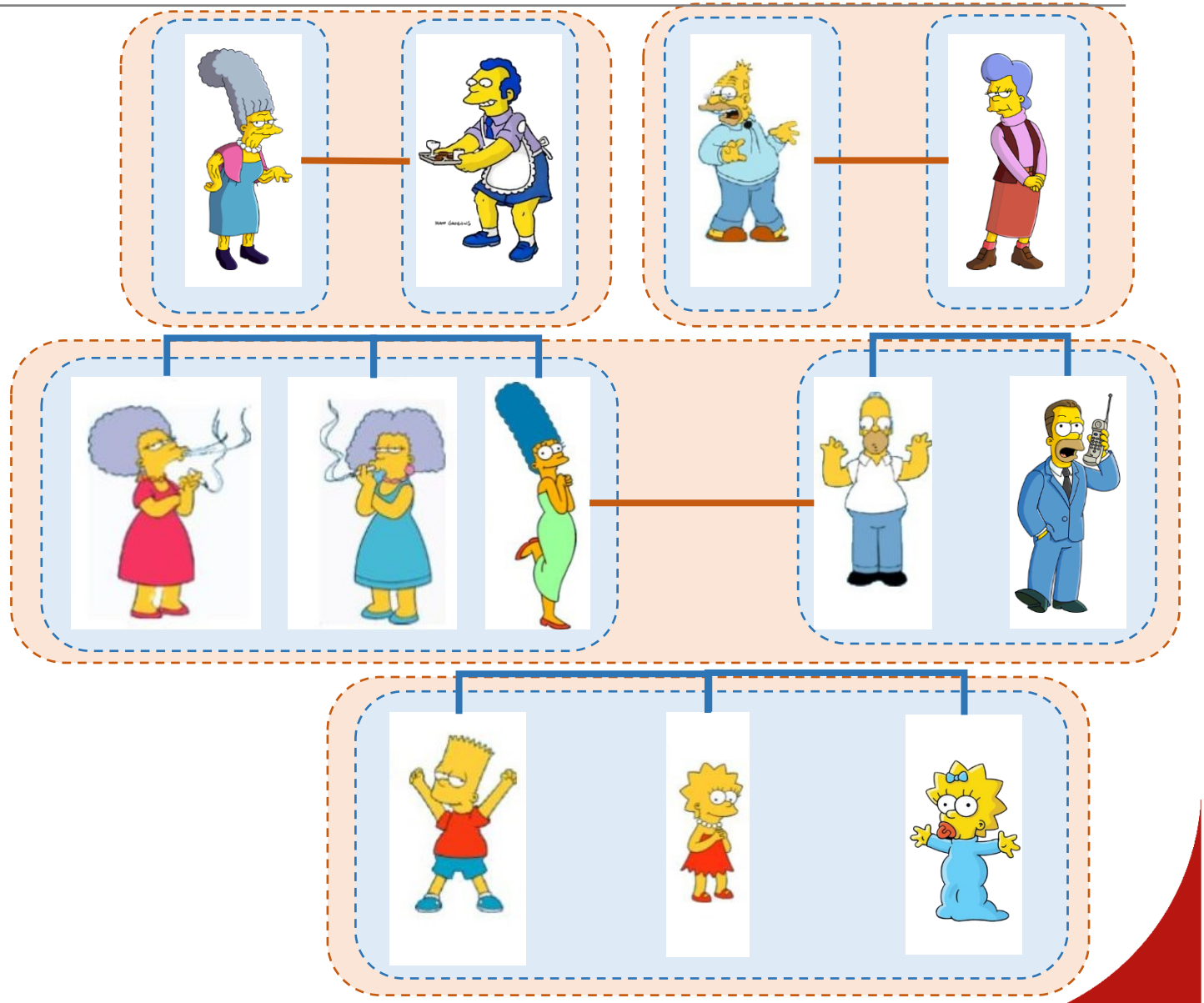
Hierarchical Clustering – Example 1

Distance Measure

Relationship	Value
Sibling	1
Spouse	2
Parent/ Child	3

Level 1: Sibling

Level 2: Spouse

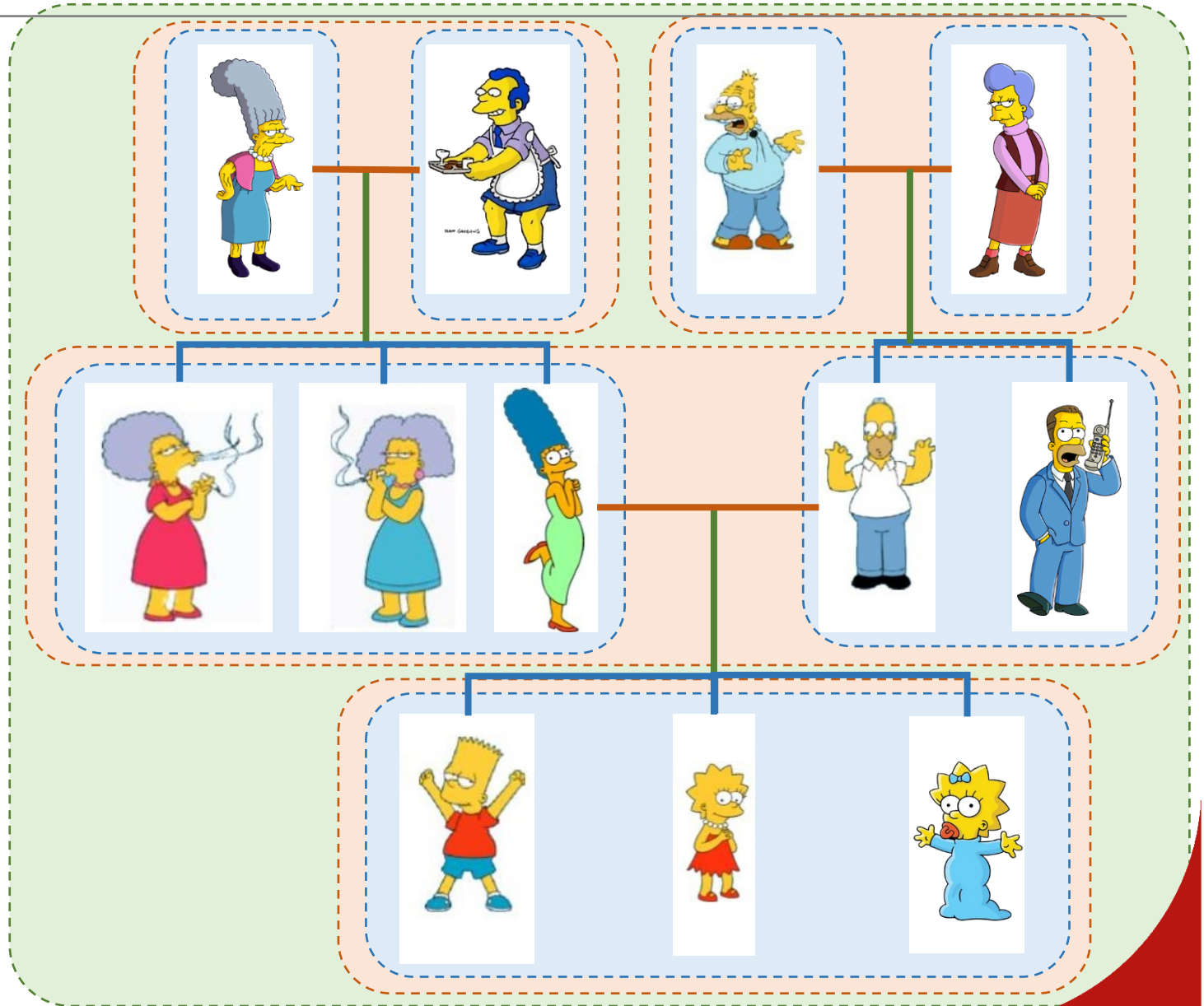


Hierarchical Clustering – Example 1

Distance Measure

Relationship	Value
Sibling	1
Spouse	2
Parent/ Child	3

Level 1: Sibling
Level 2: Spouse
Level 3: Parent/ Child

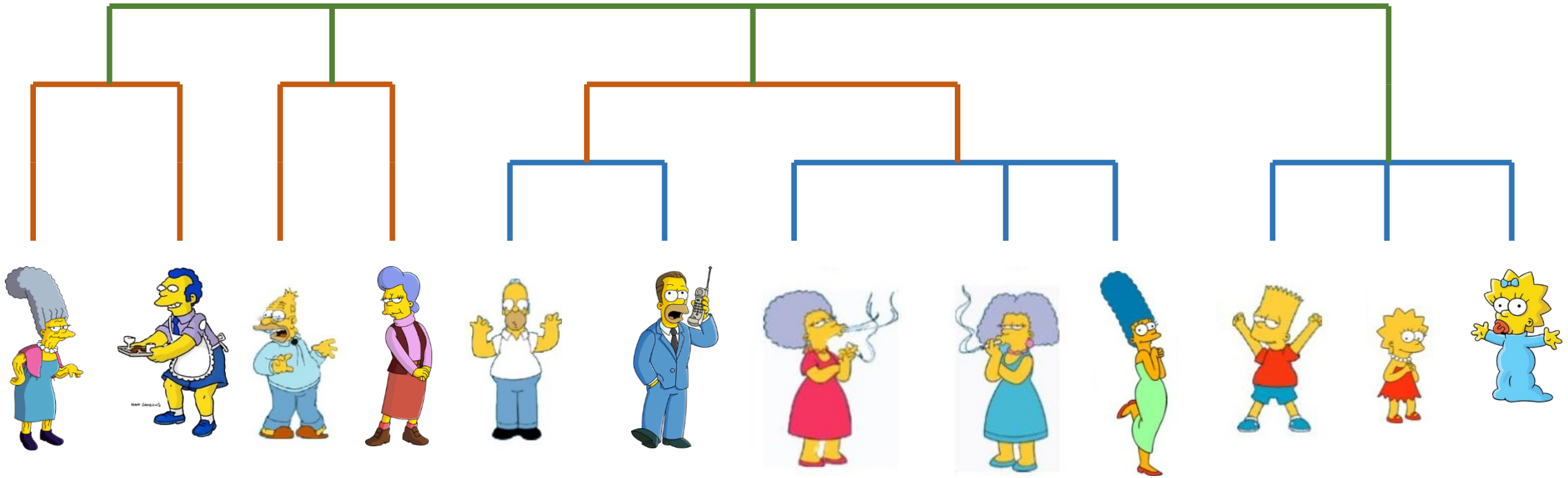


Hierarchical Clustering – Example 1

Distance Measure

Relationship	Value
Sibling	1
Spouse	2
Parent/ Child	3

Level 1: Sibling
Level 2: Spouse
Level 3: Parent/ Child

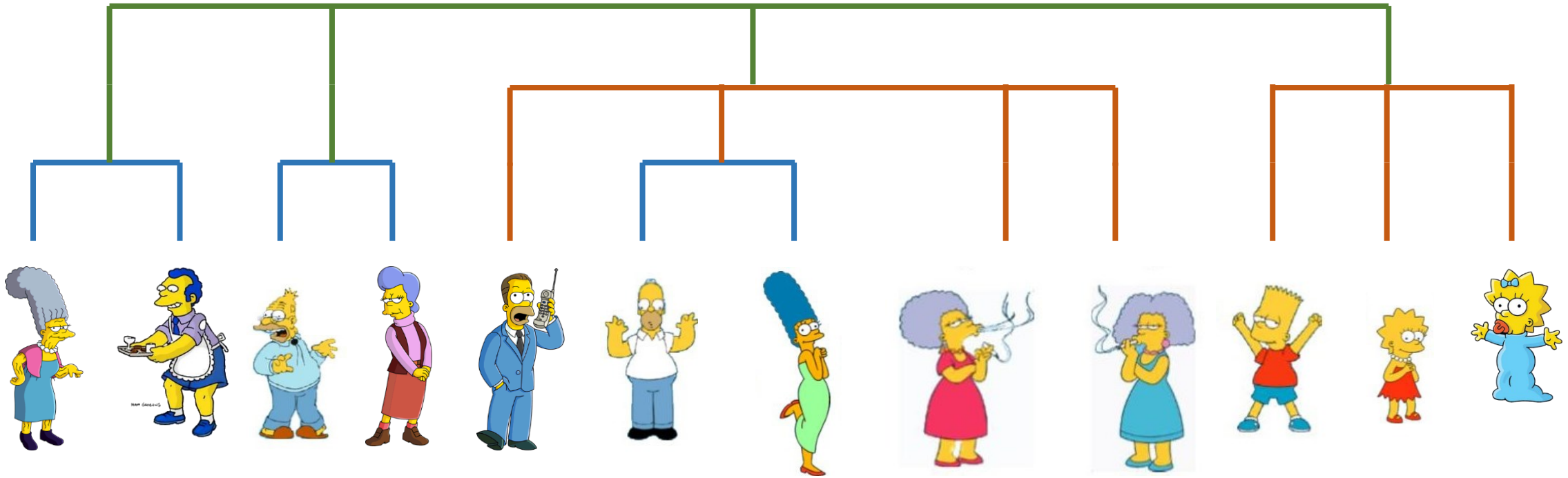


Hierarchical Clustering – Example 1

Distance Measure

Relationship	Value
Spouse	1
Sibling	2
Parent/ Child	3

Level 1: Spouse
Level 2: Sibling
Level 3: Parent/ Child



Hierarchical Clustering – Example 2

ABZ	BRS	EDI	GLA	INV	LBA	LHR	LPL	LTN	MAN	NCL	STN

IATA Codes for 12 Random UK Airports

ABZ: Aberdeen

BRS: Bristol

EDI: Edinburgh

GLA: Glasgow

INV: Inverness

LBA: Leeds Bradford

LHR: London Heathrow

LPL: Liverpool John Lennon

LTN: London Luton

MAN: Manchester

NCL: Newcastle

STN: London Stansted

Hierarchical Clustering – Example 2

	ABZ	BRS	EDI	GLA	INV	LBA	LHR	LPL	LTN	MAN	NCL	STN
ABZ	0											
BRS	403	0										
EDI	97	317	0									
GLA	125	318	41	0								
INV	73	429	112	116	0							
LBA	231	177	159	177	270	0						
LHR	402	97	332	345	443	173	0					
LPL	268	134	182	186	294	61	163	0				
LTN	375	106	307	322	418	148	28	145	0			
MAN	266	137	185	194	297	43	151	24	113	0		
NCL	242	256	91	122	195	81	252	127	225	119	0	
STN	380	131	317	335	426	158	41	164	26	146	232	0

Distance to nearest mile between airports

Source: https://airport.globefeed.com/UK_Distance_Between_Airports.asp

Hierarchical Clustering – Example 2

	ABZ	BRS	EDI	GLA	INV	LBA	LHR	LPL	LTN	MAN	NCL	STN
ABZ	0											
BRS	403	0										
EDI	97	317	0									
GLA	125	318	41	0								
INV	73	429	112	116	0							
LBA	231	177	159	177	270	0						
LHR	402	97	332	345	443	173	0					
LPL	268	134	182	186	294	61	163	0				
LTN	375	106	307	322	418	148	28	145	0			
MAN	266	137	185	194	297	43	151	24	113	0		
NCL	242	256	91	122	195	81	252	127	225	119	0	
STN	380	131	317	335	426	158	41	164	26	146	232	0

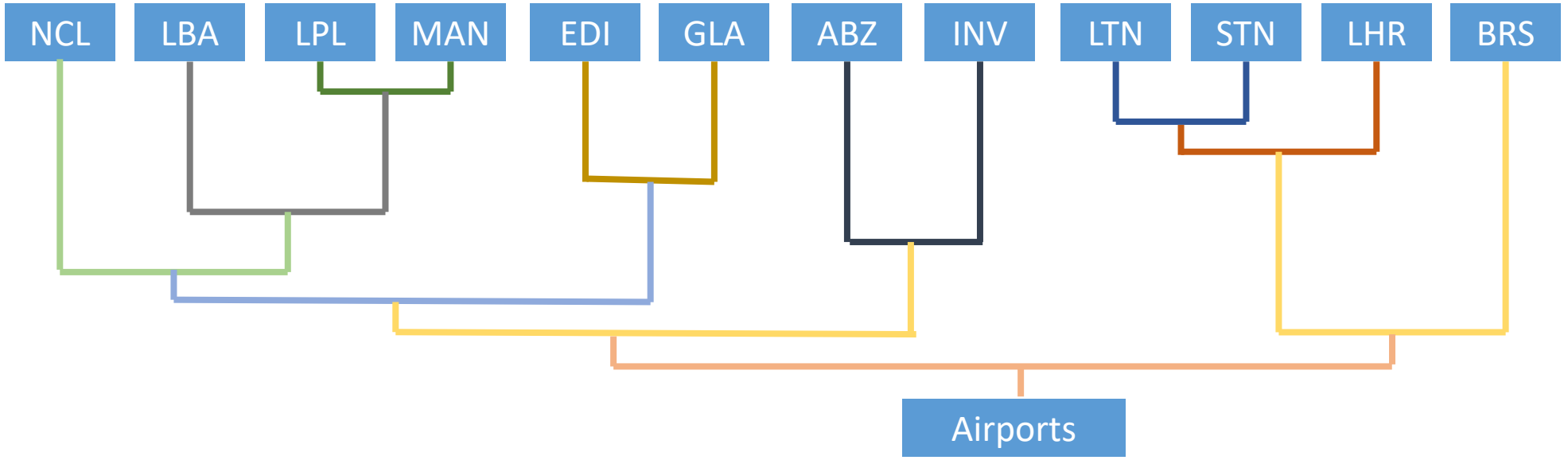
Distance to nearest mile between airports

ABZ BRS EDI GLA INV LBA LHR LPL LTN MAN NCL STN

Hierarchical Clustering – Example 2

	ABZ	BRS	EDI	GLA	INV	LBA	LHR	LPL	LTN	MAN	NCL	STN
ABZ	0											
BRS	403	0										
EDI	97	317	0									
GLA	125	318	41	0								
INV	73	429	112	116	0							
LBA	231	177	159	177	270	0						
LHR	402	97	332	345	443	173	0					
LPL	268	134	182	186	294	61	163	0				
LTN	375	106	307	322	418	148	28	145	0			
MAN	266	137	185	194	297	43	151	24	113	0		
NCL	242	256	91	122	195	81	252	127	225	119	0	
STN	380	131	317	335	426	158	41	164	26	146	232	0

Distance to nearest mile between airports

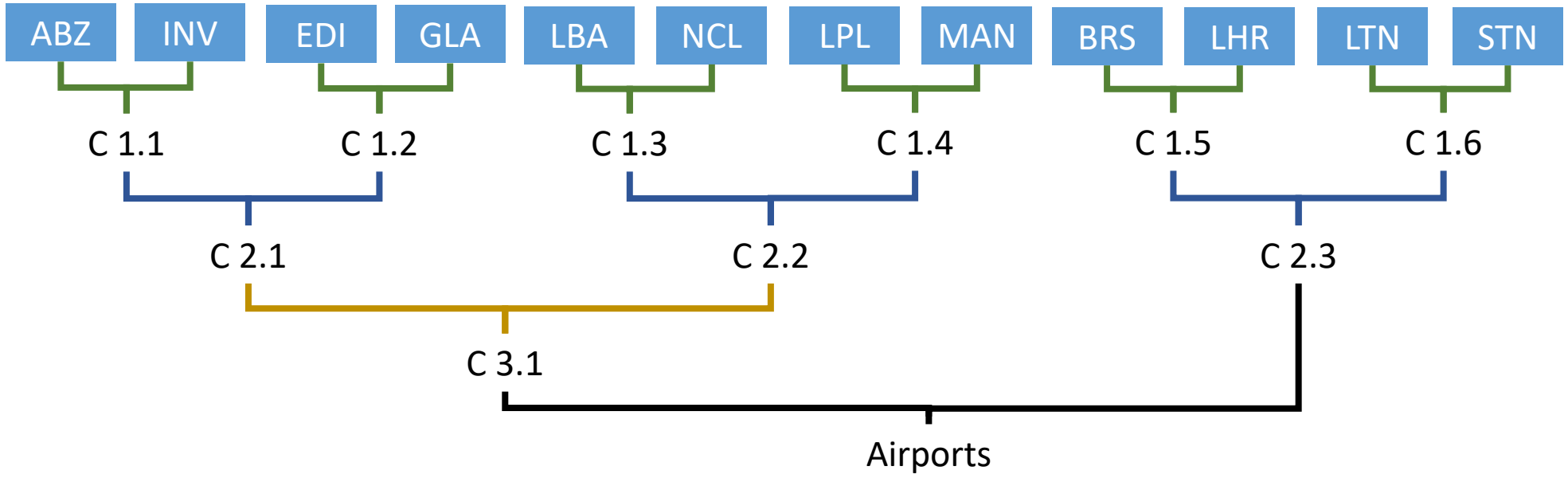


Hierarchical Clustering – Example 2

	ABZ	BRS	EDI	GLA	INV	LBA	LHR	LPL	LTN	MAN	NCL	STN
ABZ	0											
BRS	403	0										
EDI	97	317	0									
GLA	125	318	41	0								
INV	73	429	112	116	0							
LBA	231	177	159	177	270	0						
LHR	402	97	332	345	443	173	0					
LPL	268	134	182	186	294	61	163	0				
LTN	375	106	307	322	418	148	28	145	0			
MAN	266	137	185	194	297	43	151	24	113	0		
NCL	242	256	91	122	195	81	252	127	225	119	0	
STN	380	131	317	335	426	158	41	164	26	146	232	0

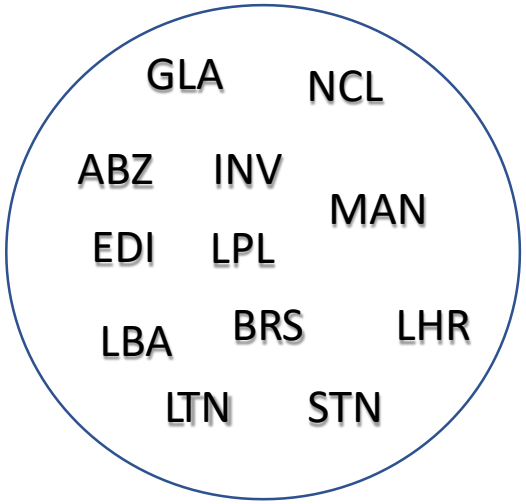
Distance to nearest mile between airports

1. Find each pair of clusters
2. Merge the closest pair



Linkage Metrics

A linkage criterion is used to determine whether cluster pairs can be merged



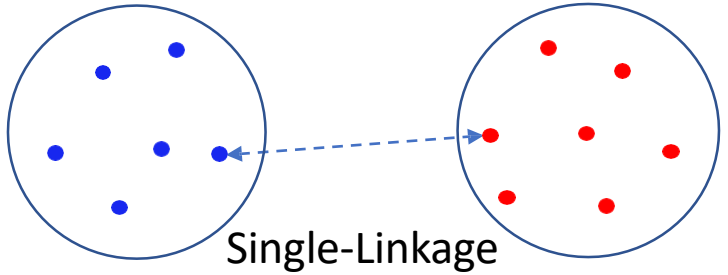
One big cluster containing all items

Different cluster containing all items

Linkage Types

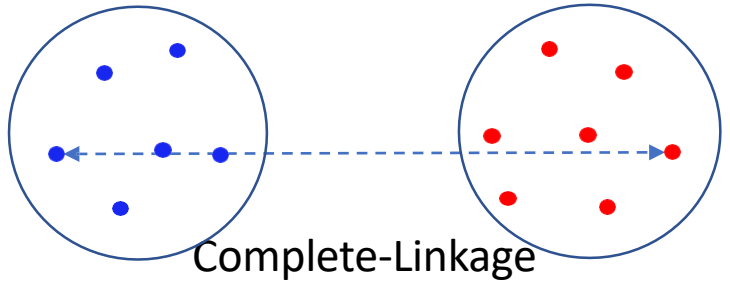
Single-linkage:

- shortest distance from any member of one cluster to any member of the other cluster



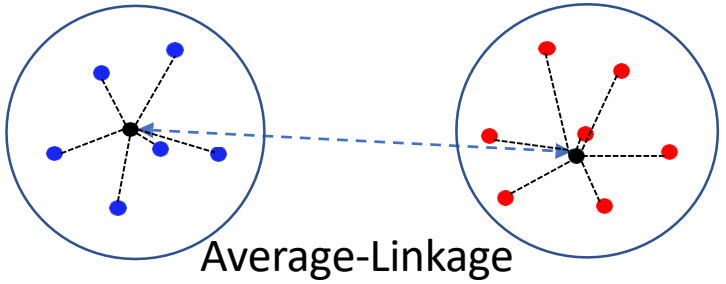
Complete-linkage:

- greatest distance from any member of one cluster to any member of the other cluster



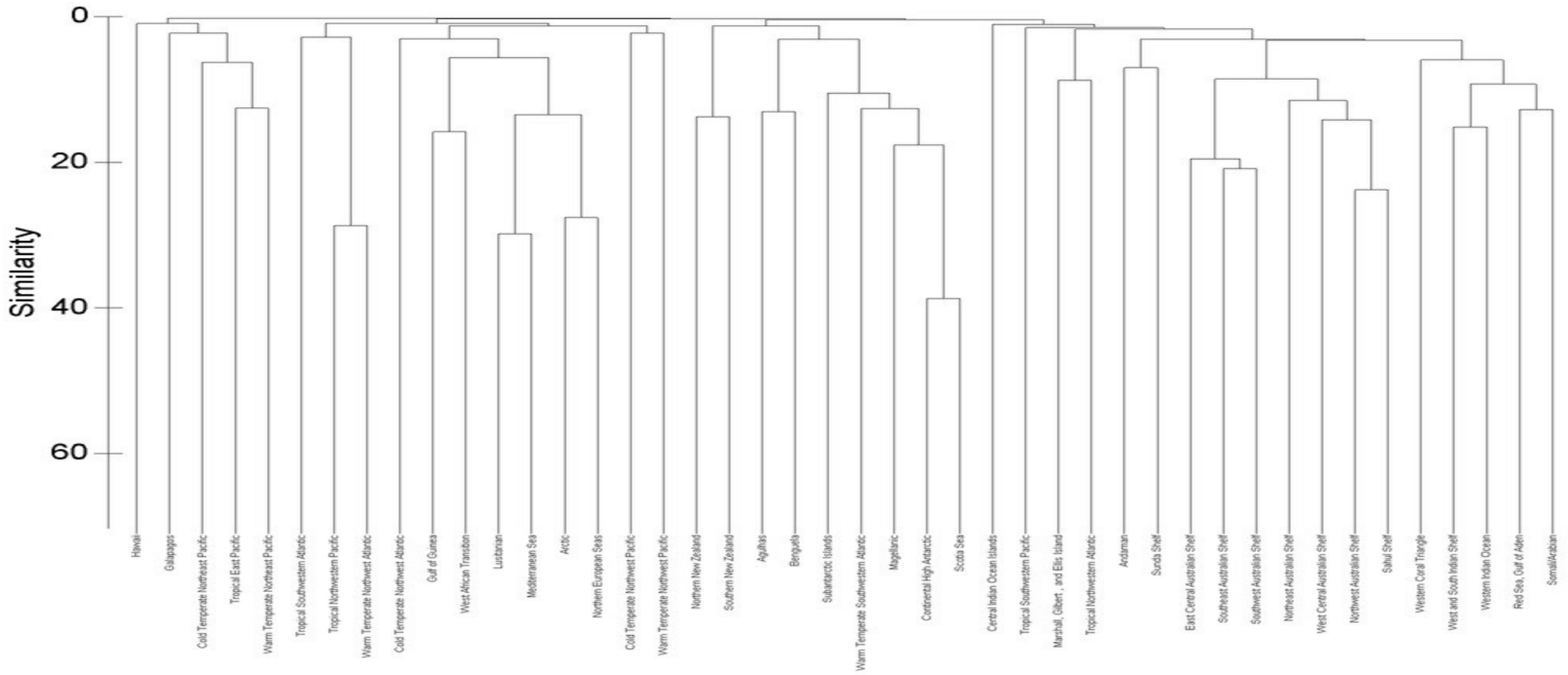
Average-linkage:

- average distance from any member of one cluster to any member of the other cluster



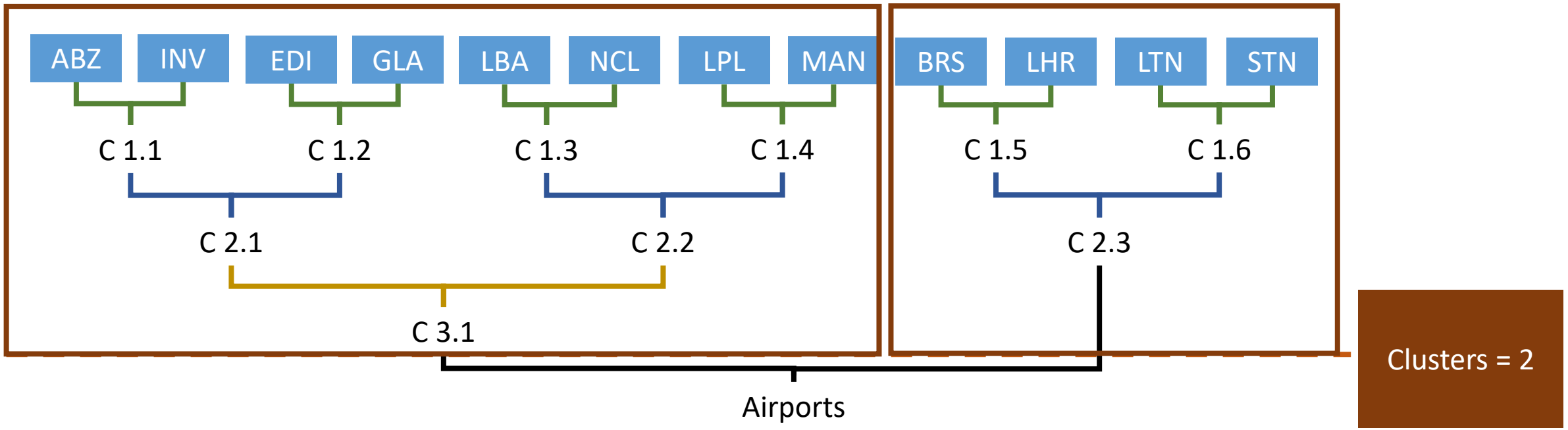
Dendrogram

A **dendrogram** is a diagrammatic representation
It is often used in hierarchical clustering



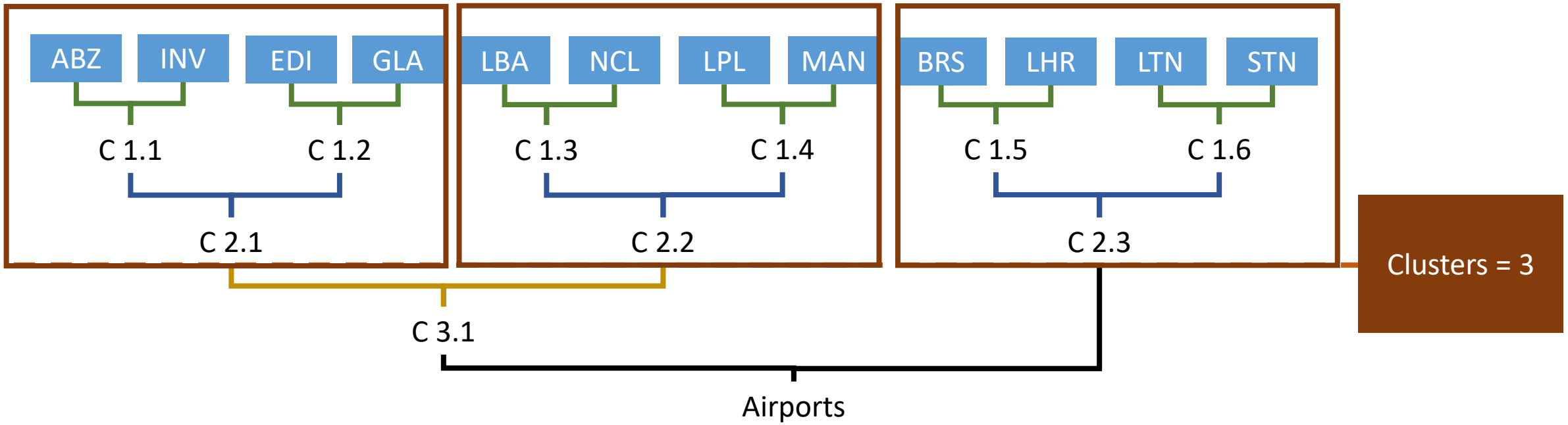
Dendrogram

We can obtain different clusterings of the data by splitting on different levels



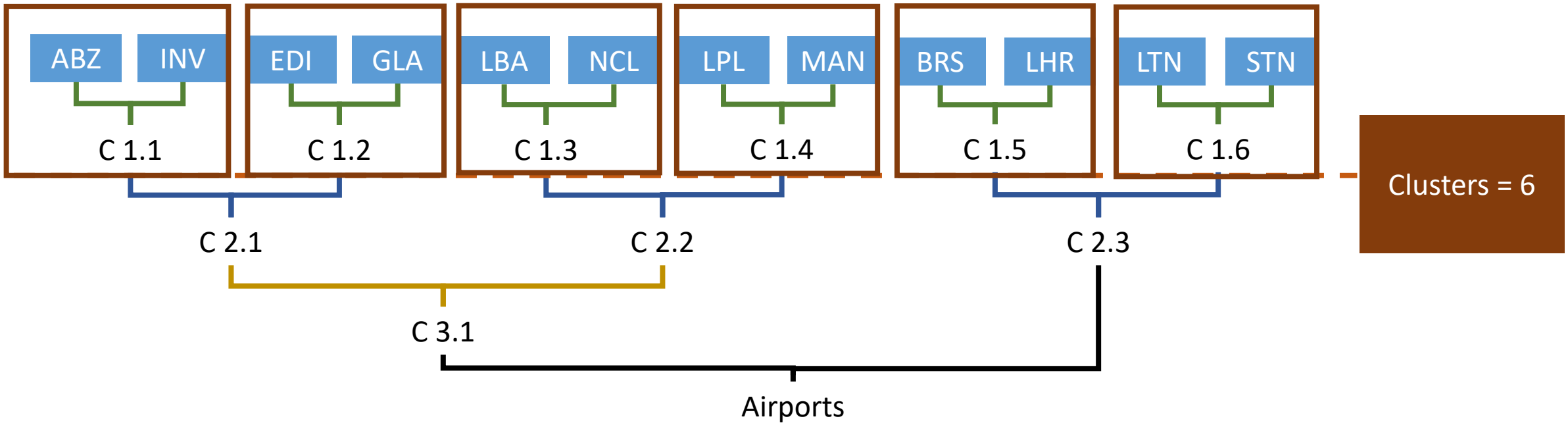
Dendrogram

We can obtain different clusterings of the data by splitting on different levels



Dendrogram

We can obtain different clusterings of the data by splitting on different levels



Lancaster University

Hierarchical Clustering Summary

Deterministic

- You can always get the same result

Inefficient

- Too many distance computations to perform
- Depends on the task

Multilevel Representation

- No need to select number of clusters
- Can be visualised with dendrogram to select k

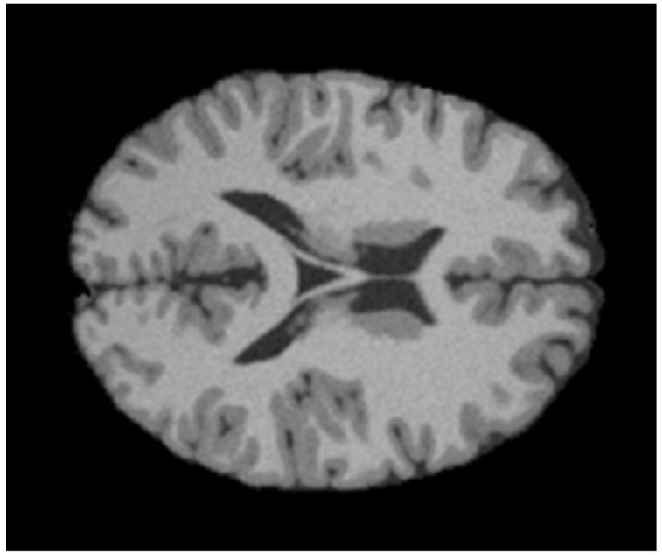
Flexible

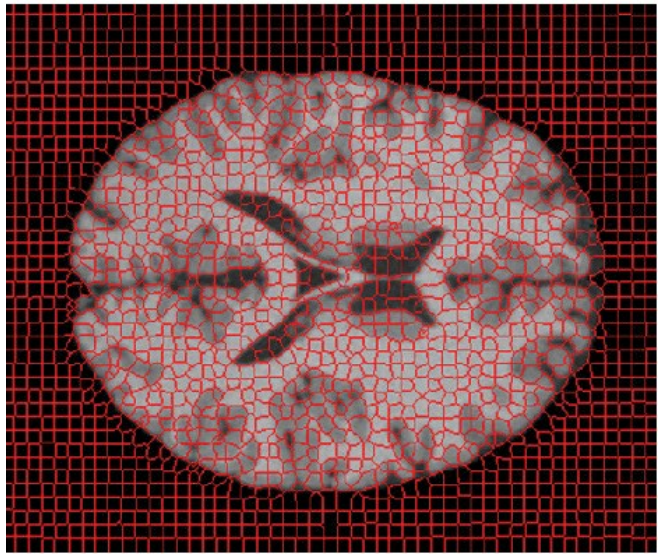
- with respect to linkage criteria

Lancaster University

Applications of Clustering

- Market segmentation
- Social network analysis
- Search results grouping
- Medical imaging
- Genetics and evolutionary biology
- Anomaly detection
- Recommender systems





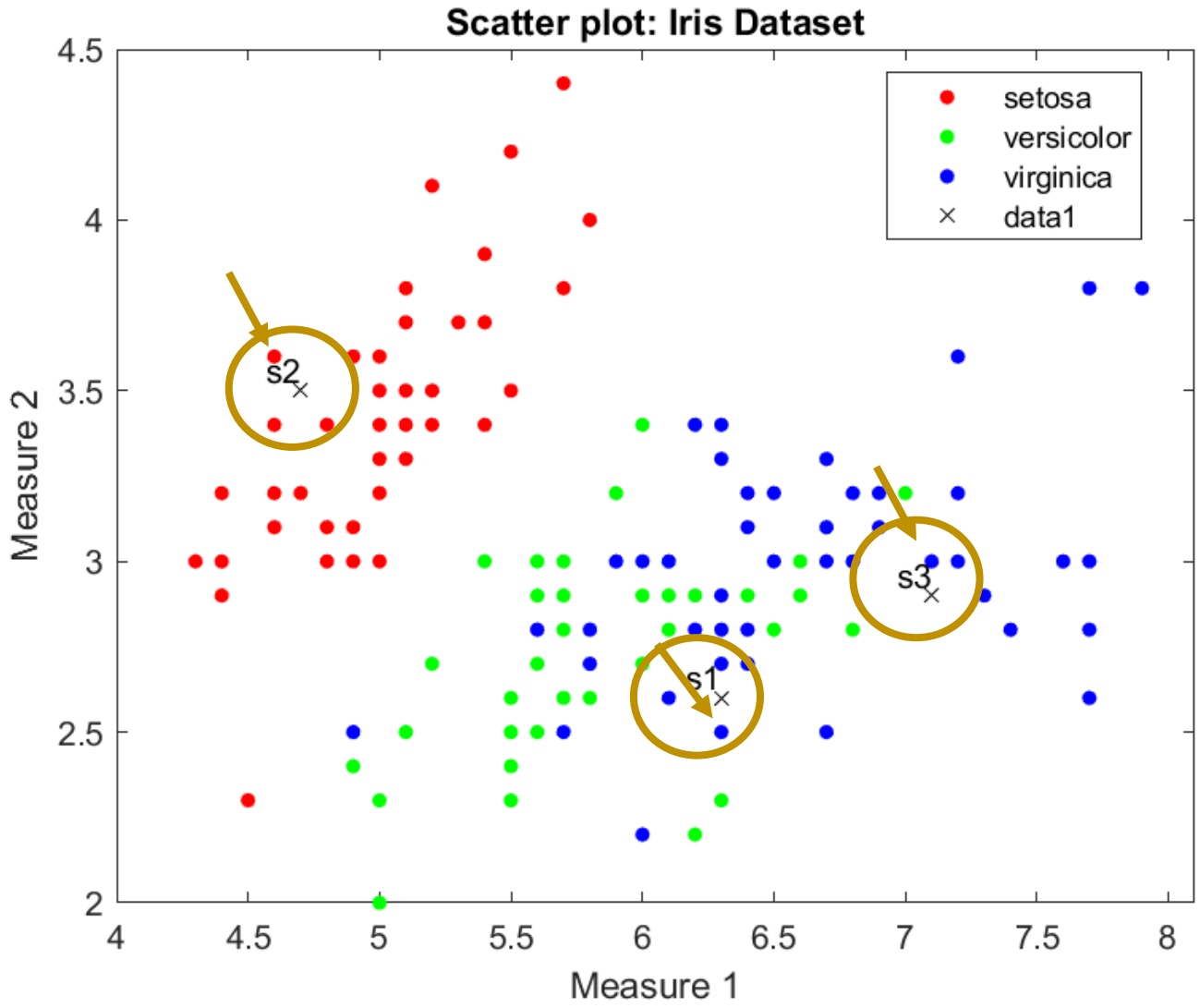
Classification / Clustering
So far…

KNN Classification

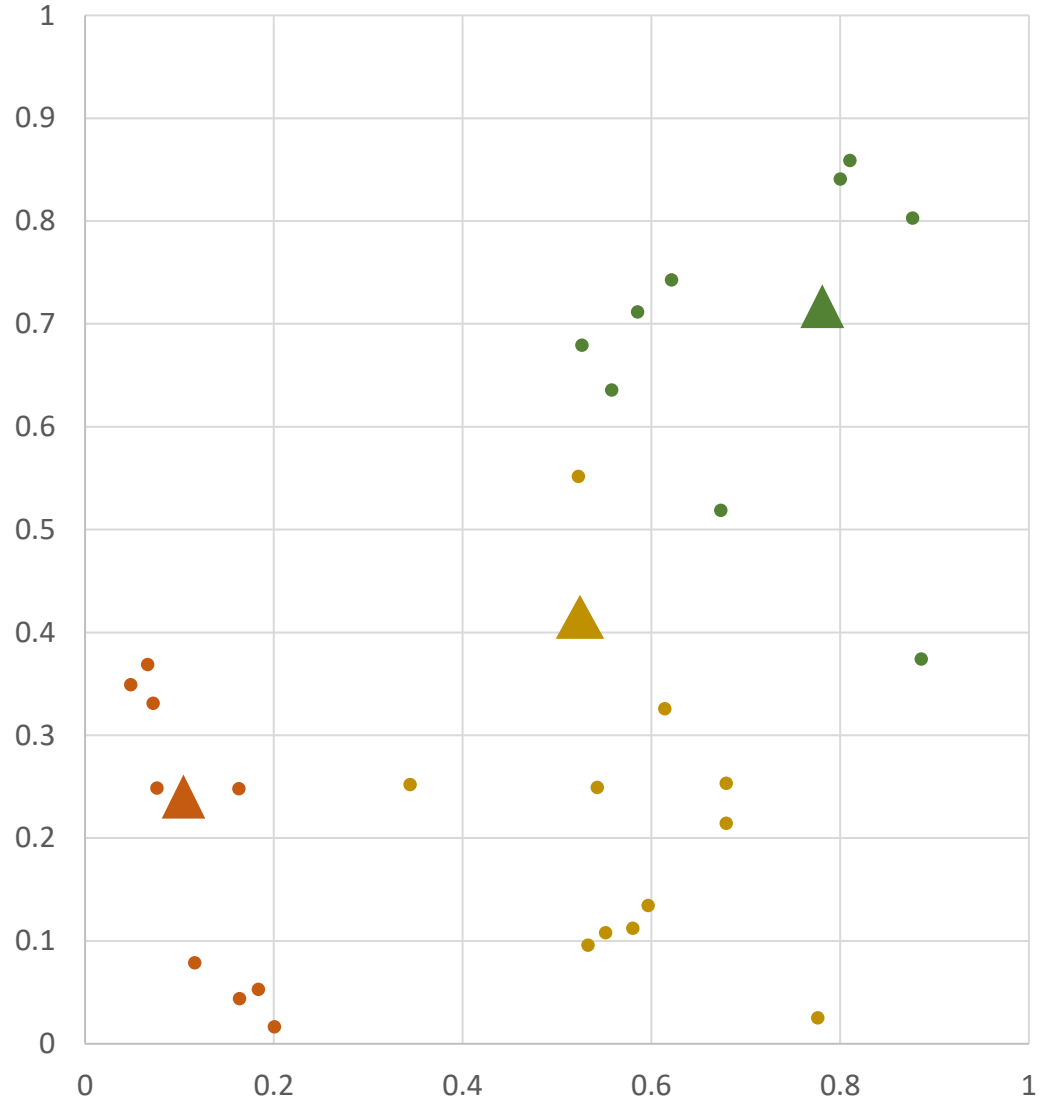
2.5: Classification

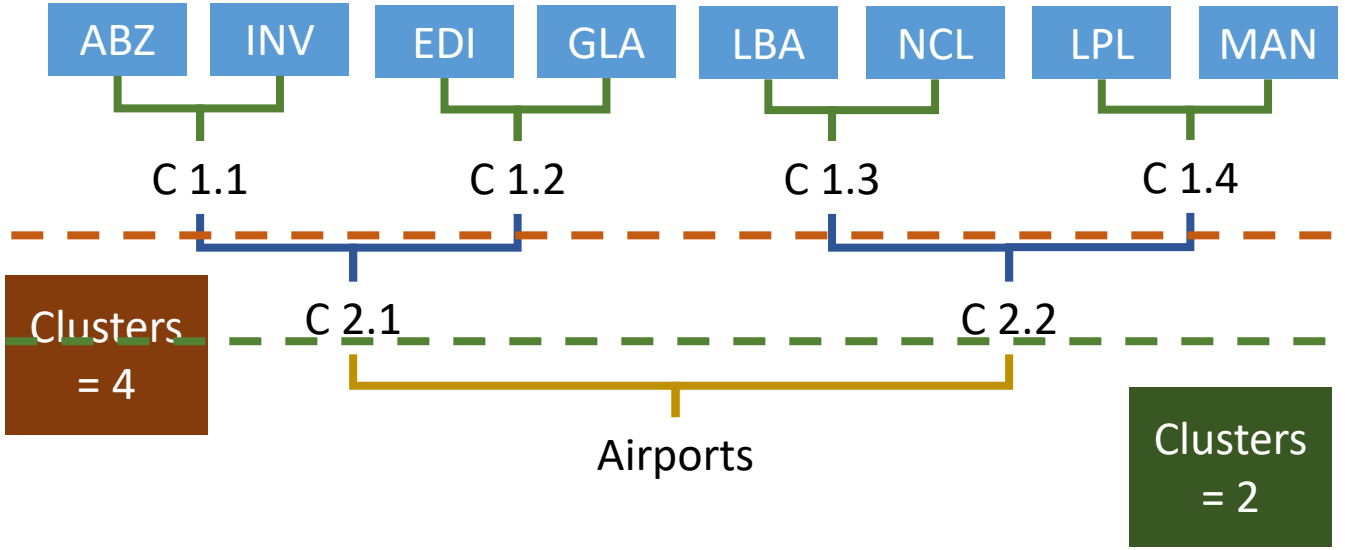
Add the following samples to the graphs created in Section 2.2 and use this to estimate the class that they belong to.

	Measure 1	Measure 2	Measure 3	Measure 4
Sample 1	6.3	2.6	4.1	1.2
Sample 2	4.7	3.5	1.5	0.3
Sample 3	7.1	2.9	5.5	2.1



K-Means and Hierarchical Clustering





Classification
Decision Trees

Decision Trees

Linearly Separable

Feature Space Division

Definitions and Structure

Multiple Solutions

Decision Trees

Linearly Separable

Feature Space Division

Definitions and Structure

Multiple Solutions

Entropy

Optimal Decision Trees

Definition

Binary and Multiclass

How to Calculate

Decision Trees

Linearly Separable

Feature Space Division

Definitions and Structure

Multiple Solutions

Entropy

Optimal Decision Trees

Definition

Binary and Multiclass

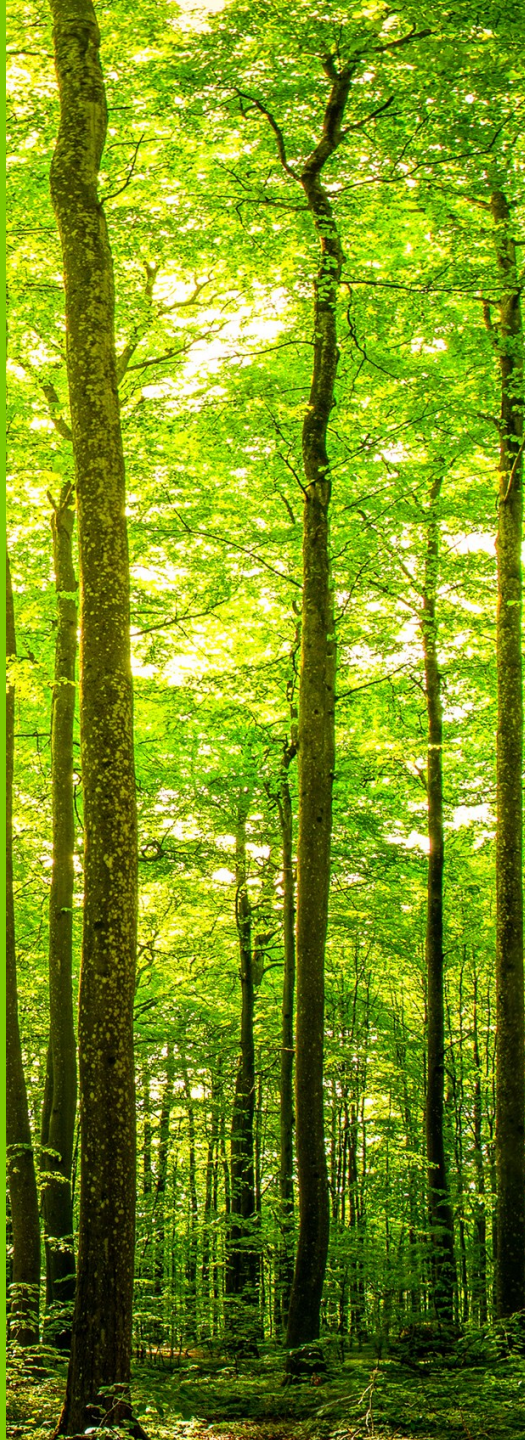
How to Calculate

Information Gain

Definition

Relevance

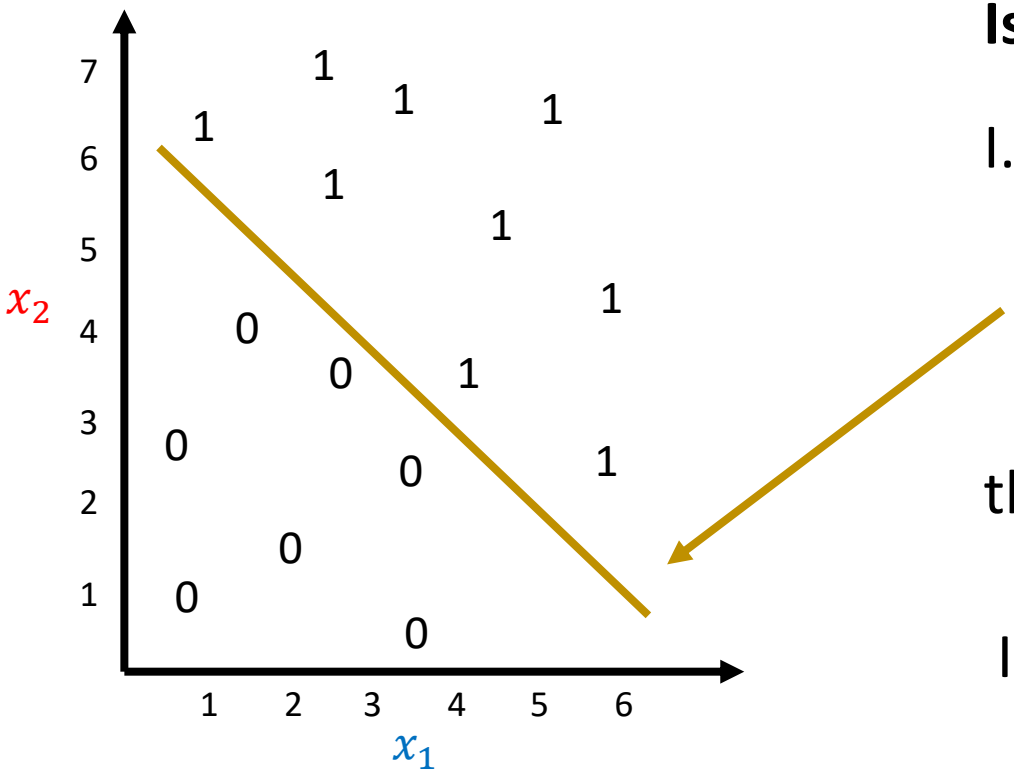
Calculation



Decision Trees

Let $x_1 \in \mathbb{R}^m, x_2 \in \mathbb{R}^n$ - i.e. continuous-valued feature-vectors of size m and n respectively
Assume binary classification output, i.e.

$$f(x_1, x_2) = \begin{cases} 0 \\ 1 \end{cases} \qquad f: (x_1, x_2) \rightarrow \{0,1\}$$



Is this data linear separable?

I.e. Can we define a straight line

$$\mathcal{L}(x_1, x_2): \alpha_1 x_1 + \alpha_2 x_2 + \alpha_3 = 0,$$

$$\alpha_1, \alpha_2, \alpha_3 \in \mathbb{R}$$

that separates the classes?

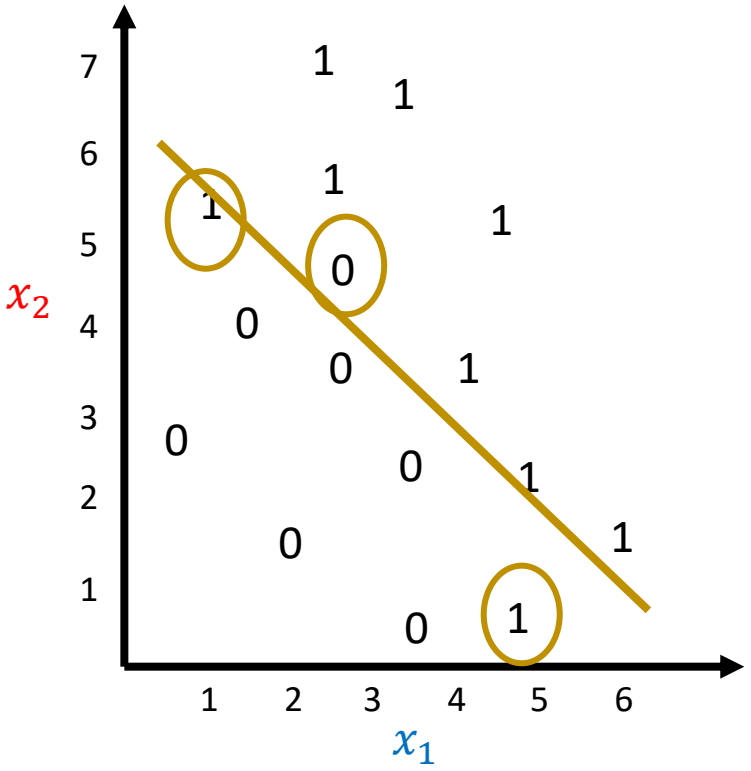
If yes: $f(x_1, x_2) = \begin{cases} 0 \ \ if \ \ \mathcal{L}(x_1, x_2) < 0 \\ 1 \ \ if \ \ \mathcal{L}(x_1, x_2) > 0 \end{cases}$

Decision Trees

Let $x_1 \in \mathbb{R}^m, x_2 \in \mathbb{R}^n$ - i.e. continuous-valued feature-vectors of size m and n respectively
Assume binary classification output, i.e.

$$f(x_1, x_2) = \begin{cases} 0 \\ 1 \end{cases} \qquad f: (x_1, x_2) \rightarrow \{0,1\}$$

Is this data linear separable?

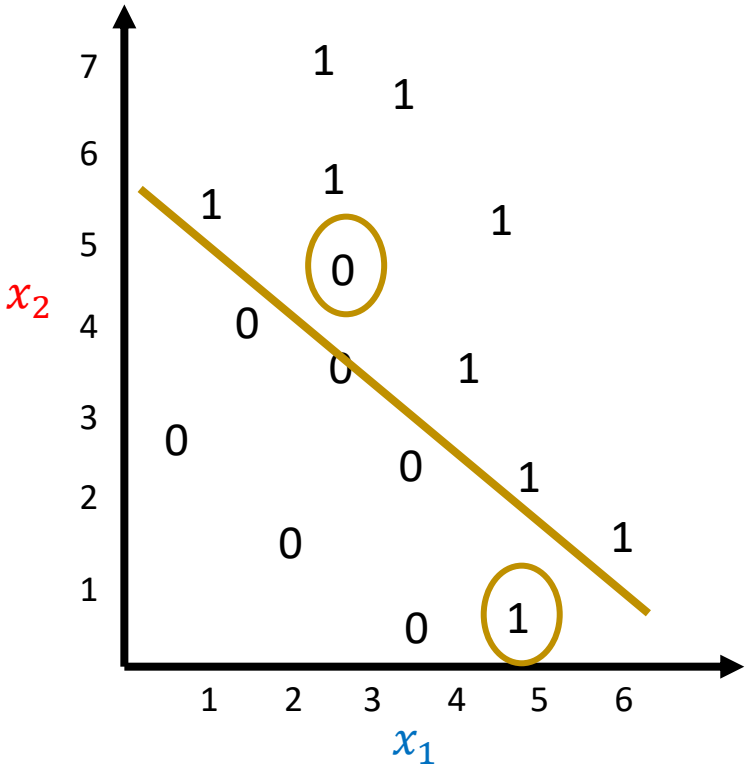


Decision Trees

Let $x_1 \in \mathbb{R}^m, x_2 \in \mathbb{R}^n$ - i.e. continuous-valued feature-vectors of size m and n respectively
Assume binary classification output, i.e.

$$f(x_1, x_2) = \begin{cases} 0 \\ 1 \end{cases} \qquad f: (x_1, x_2) \rightarrow \{0,1\}$$

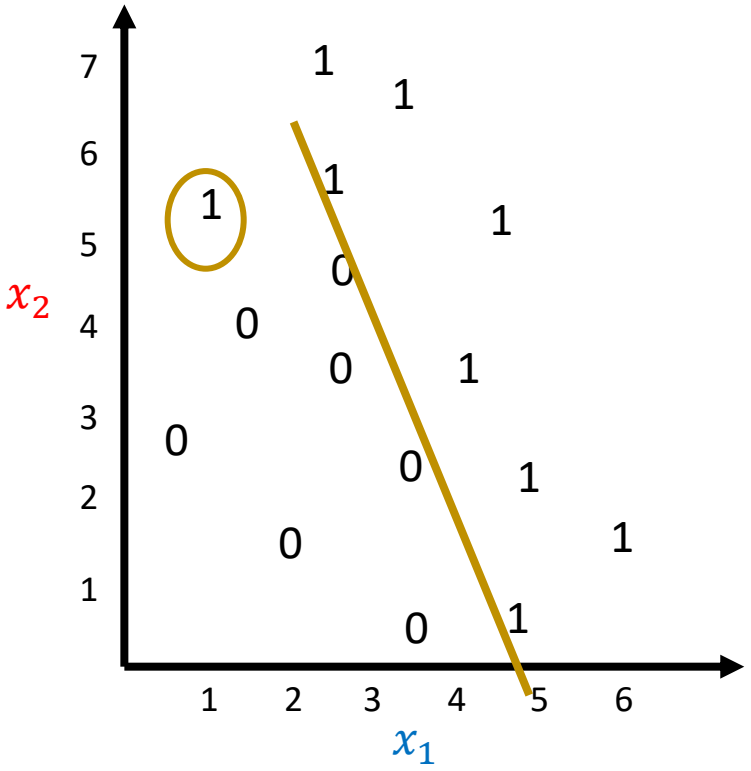
Is this data linear separable?



Decision Trees

Let $x_1 \in \mathbb{R}^m, x_2 \in \mathbb{R}^n$ - i.e. continuous-valued feature-vectors of size m and n respectively
Assume binary classification output, i.e.

$$f(x_1, x_2) = \begin{cases} 0 \\ 1 \end{cases} \qquad f: (x_1, x_2) \rightarrow \{0,1\}$$



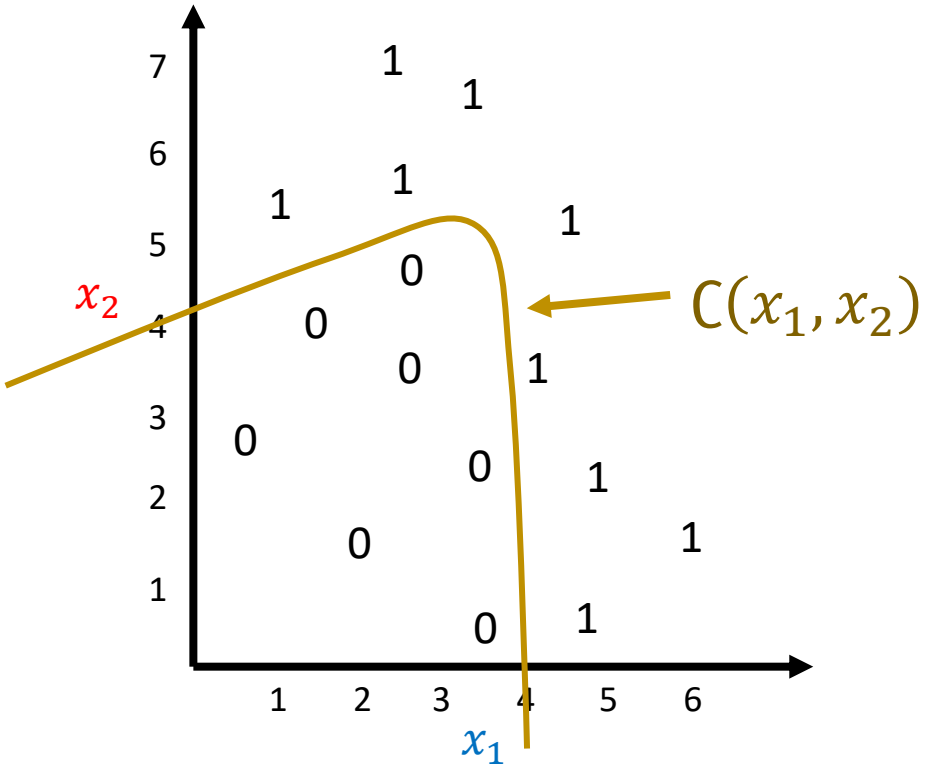
Is this data linear separable?

No ☹

Decision Trees

Let $x_1 \in \mathbb{R}^m, x_2 \in \mathbb{R}^n$ - i.e. continuous-valued feature-vectors of size m and n respectively
Assume binary classification output, i.e.

$$f(x_1, x_2) = \begin{cases} 0 \\ 1 \end{cases} \qquad f: (x_1, x_2) \rightarrow \{0,1\}$$



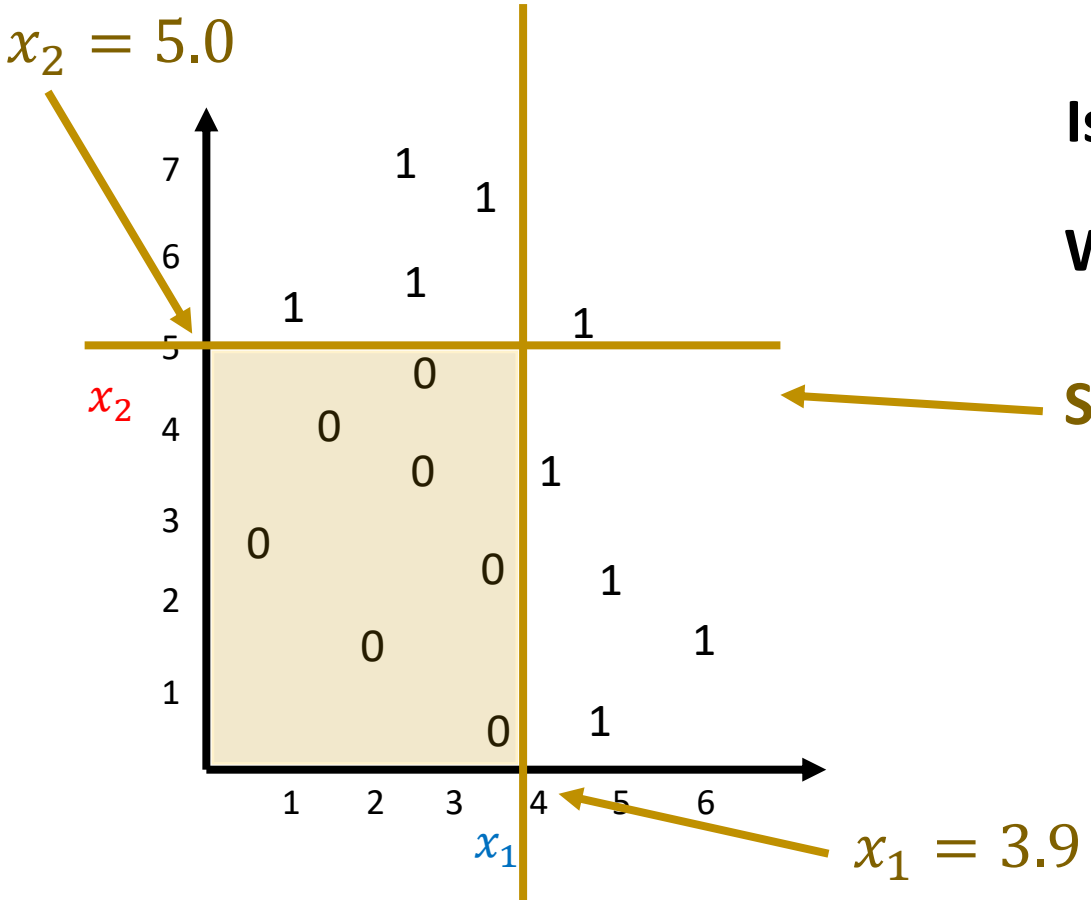
Is this data linear separable? No.

What can we do?

Decision Trees

Let $x_1 \in \mathbb{R}^m, x_2 \in \mathbb{R}^n$ - i.e. continuous-valued feature-vectors of size m and n respectively
Assume binary classification output, i.e.

$$f(x_1, x_2) = \begin{cases} 0 \\ 1 \end{cases} \qquad f: (x_1, x_2) \rightarrow \{0,1\}$$



Is this data linear separable? No.

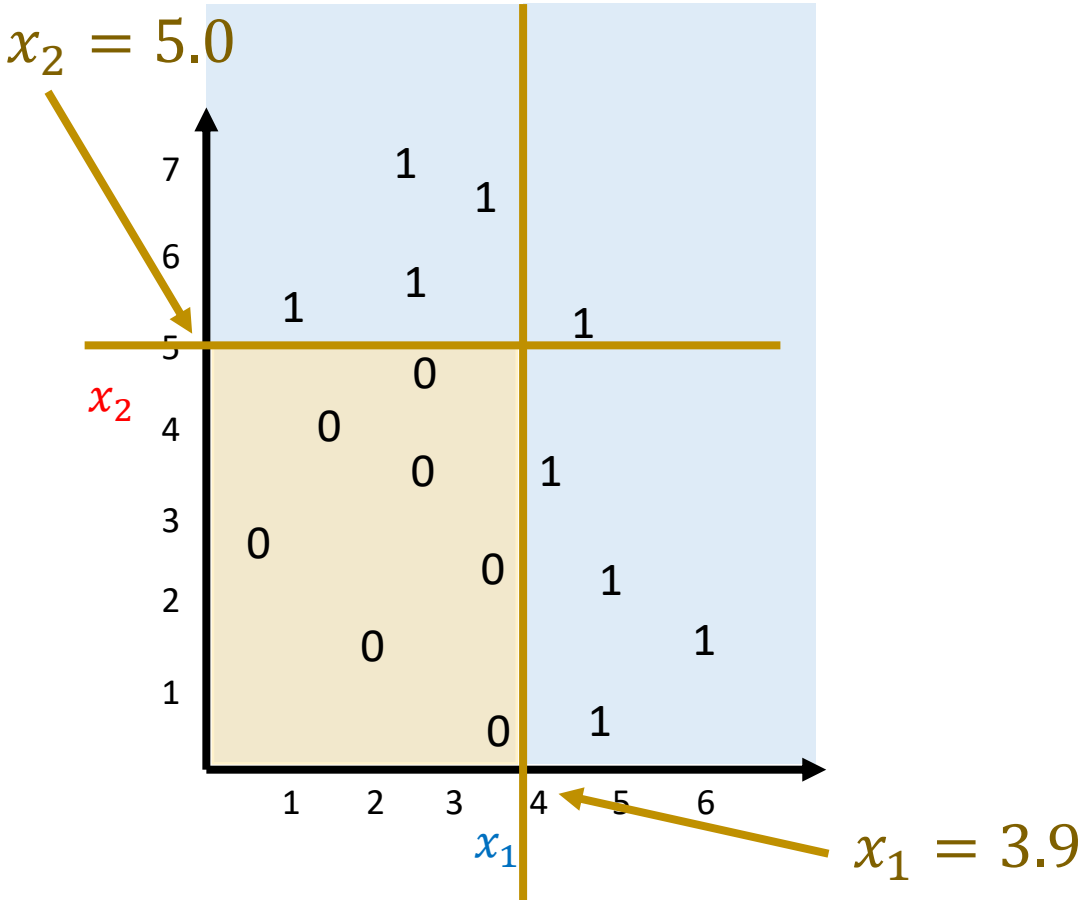
What can we do?

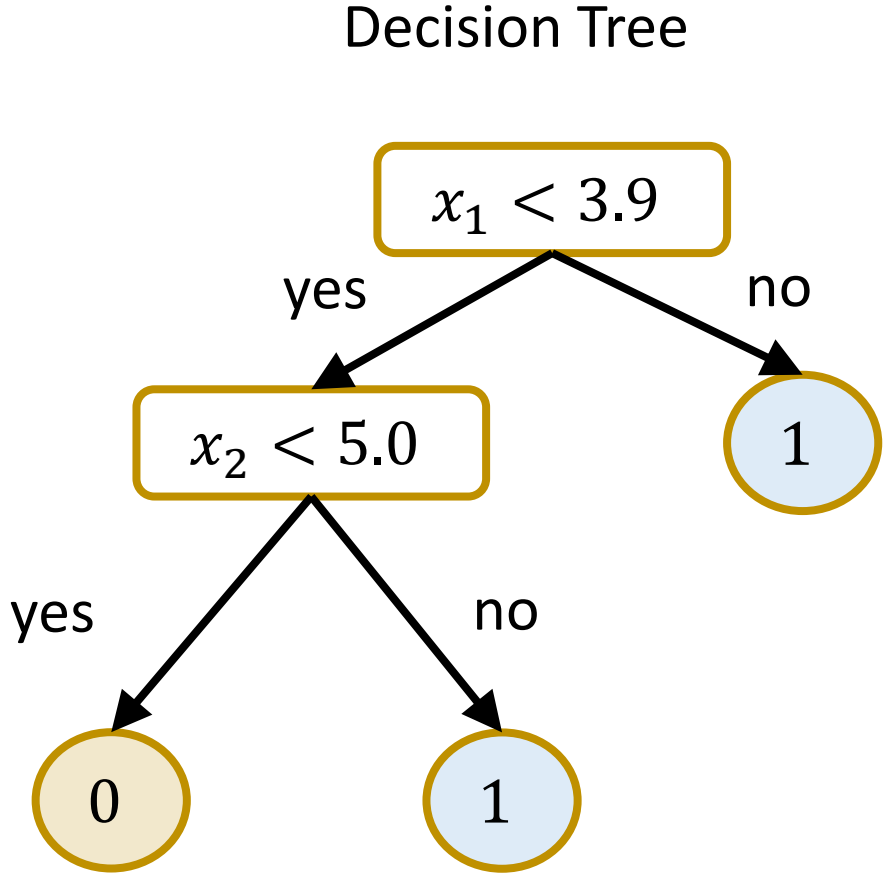
Sub-divide in a linear way

$$f(x_1, x_2) = \begin{cases} 0 \text{ if } x_1 < 3.9 \text{ and } x_2 < 5.0 \\ 1 \text{ otherwise} \end{cases}$$

Decision Trees

$$f(x_1, x_2) = \begin{cases} 0 \text{ if } x_1 < 3.9 \text{ and } x_2 < 5.0 \\ 1 \text{ otherwise} \end{cases}$$

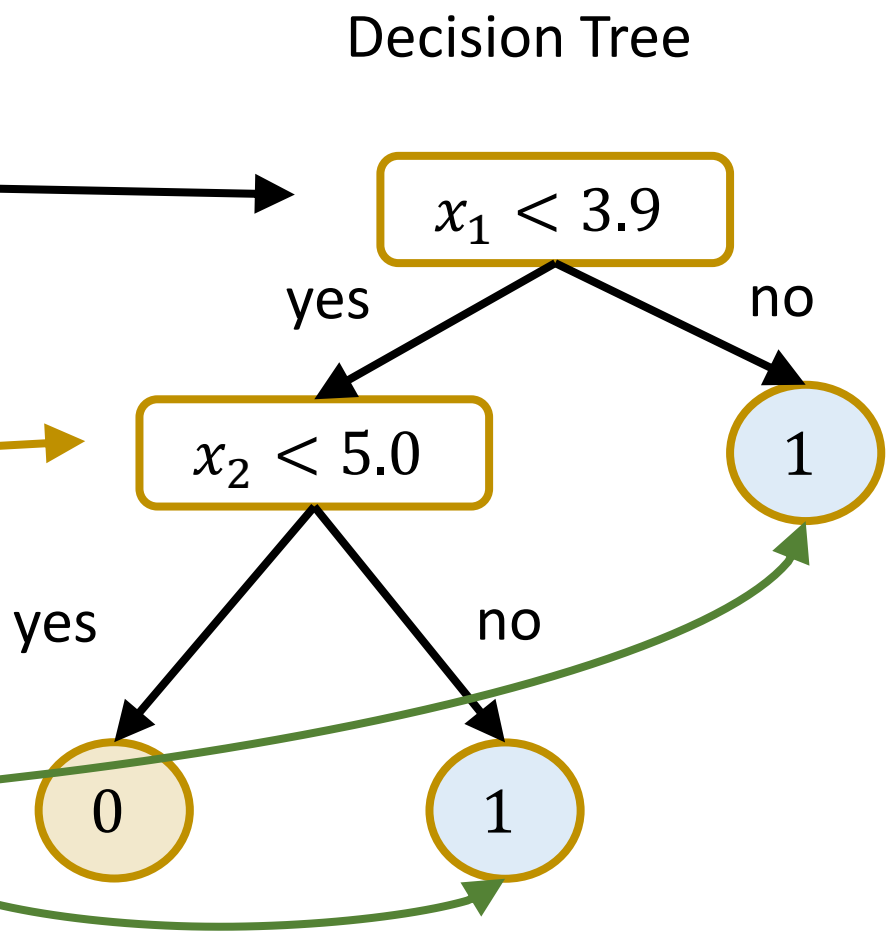




Decision Tree Structure

Decision Tree Comprises:

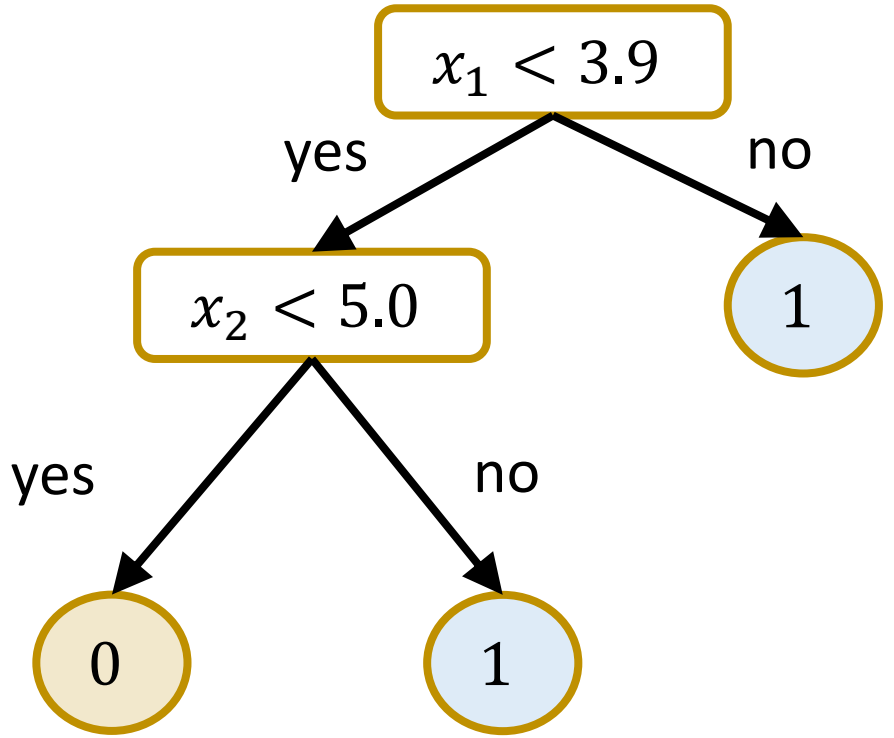
- **Root Node:**
  - No incoming edges
  - Two or more outgoing edges
  - Tests a condition
- **Internal Nodes:**
  - One incoming edges
  - Two or more outgoing edges
  - Tests a condition
- **Leaf / Terminal Nodes:**
  - One incoming edges
  - No outgoing edges
  - Gives the outcome prediction



Decision Tree Definition

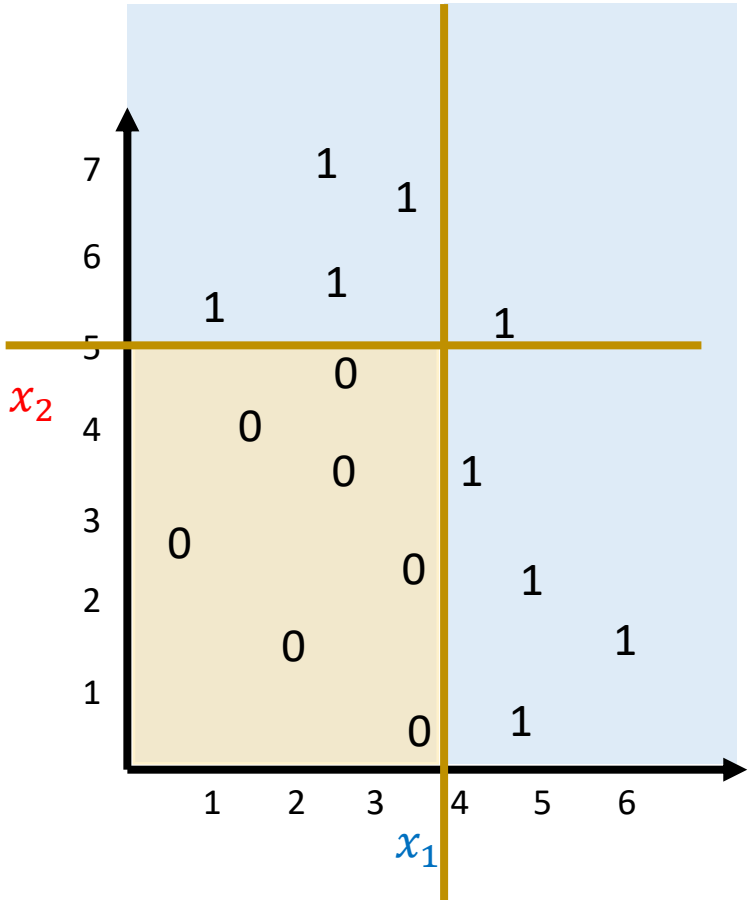
A **decision tree classifier** organizes a series of test questions and conditions in a tree-like structure containing

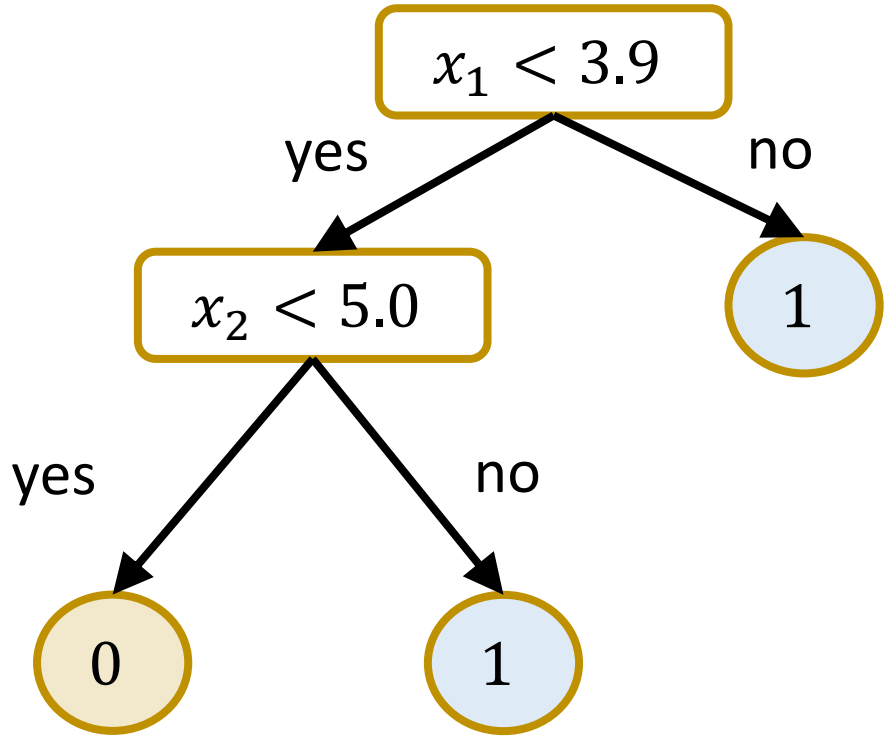
- **root** and **internal nodes** with **feature test conditions** to separate samples with different characteristics.
- **leaf nodes** that assign class labels.



Is the solution unique?

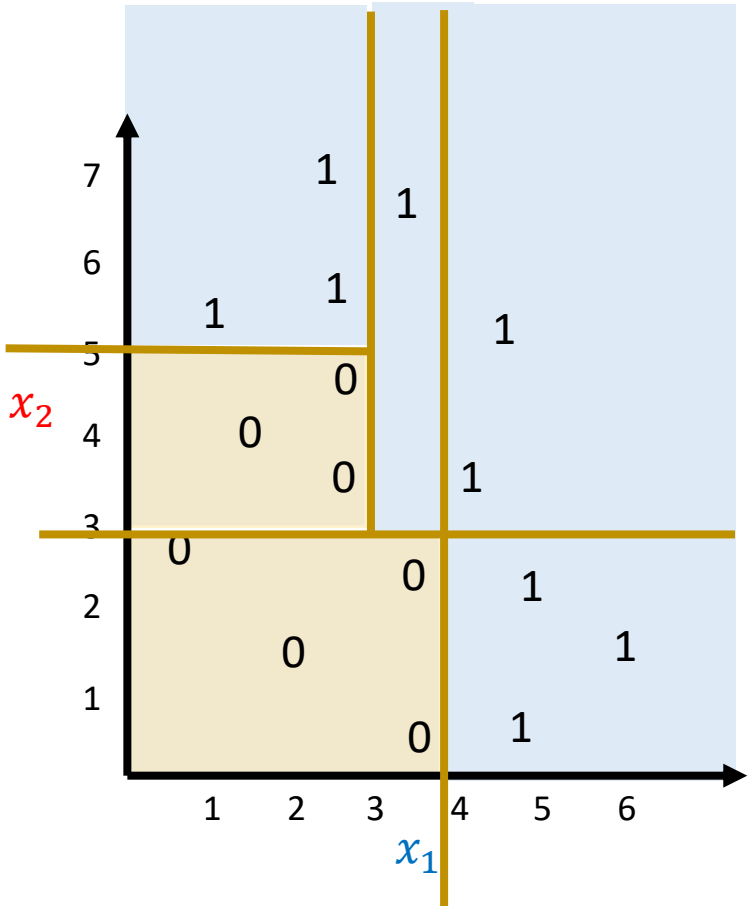
Can we partition the space in multiple ways?

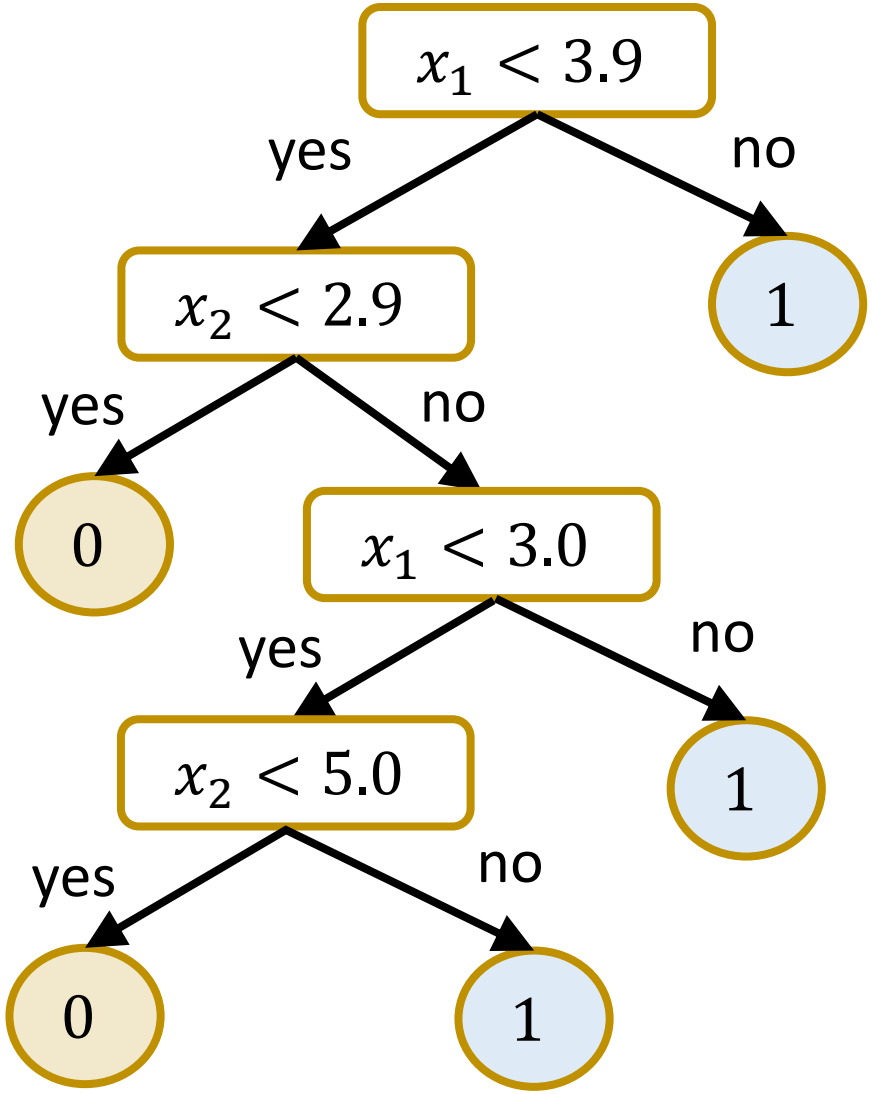




Is the solution unique?

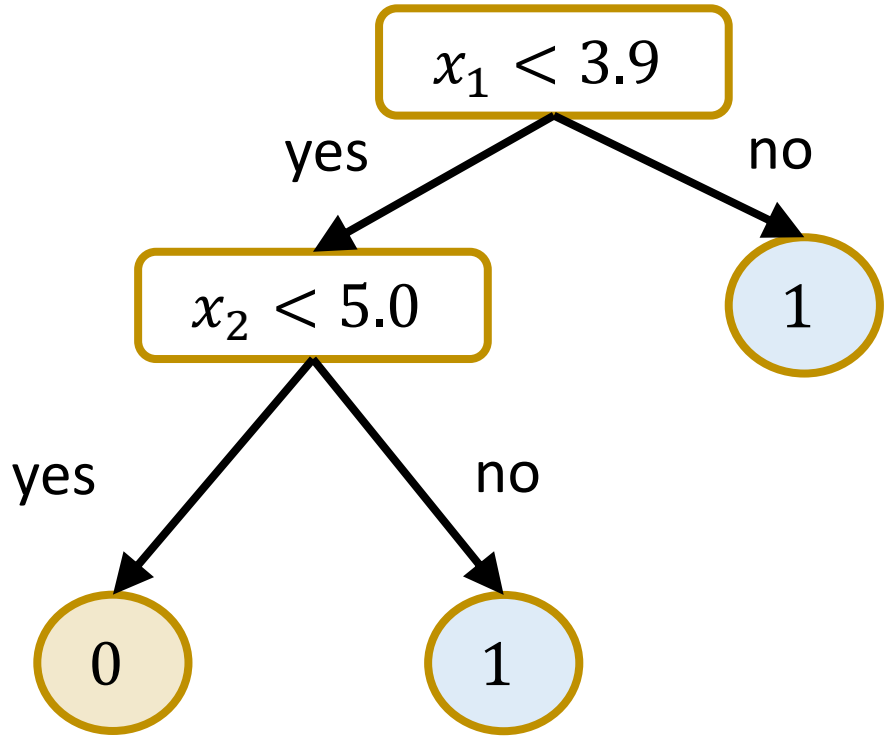
Can we partition the space in multiple ways?

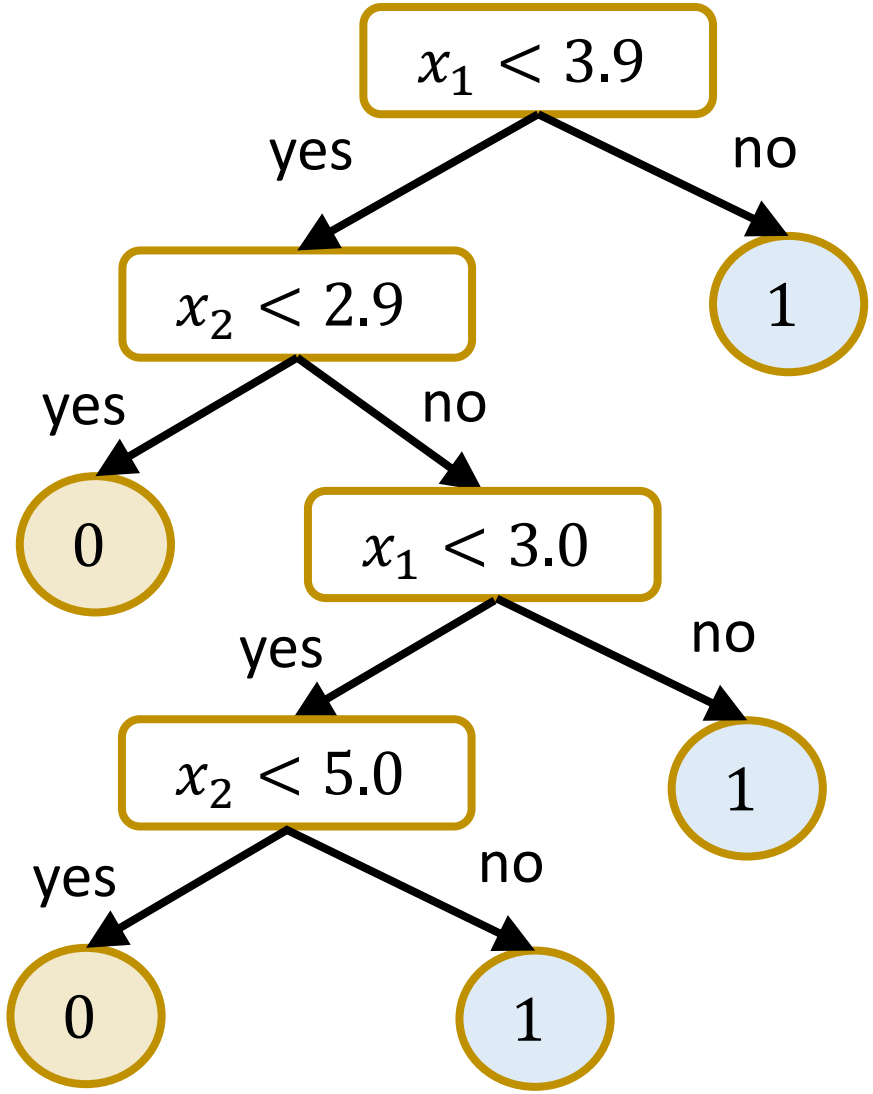




Is the solution unique?

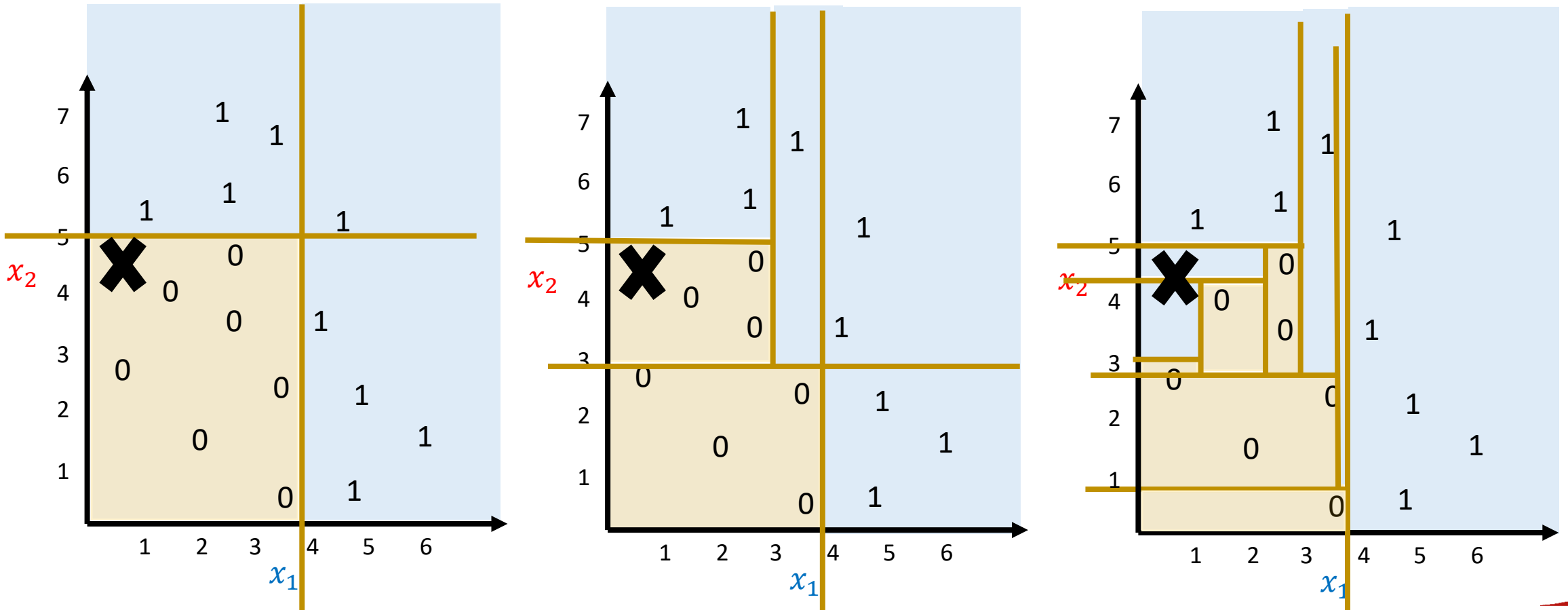
Can we partition the space in multiple ways?





Is the solution unique?

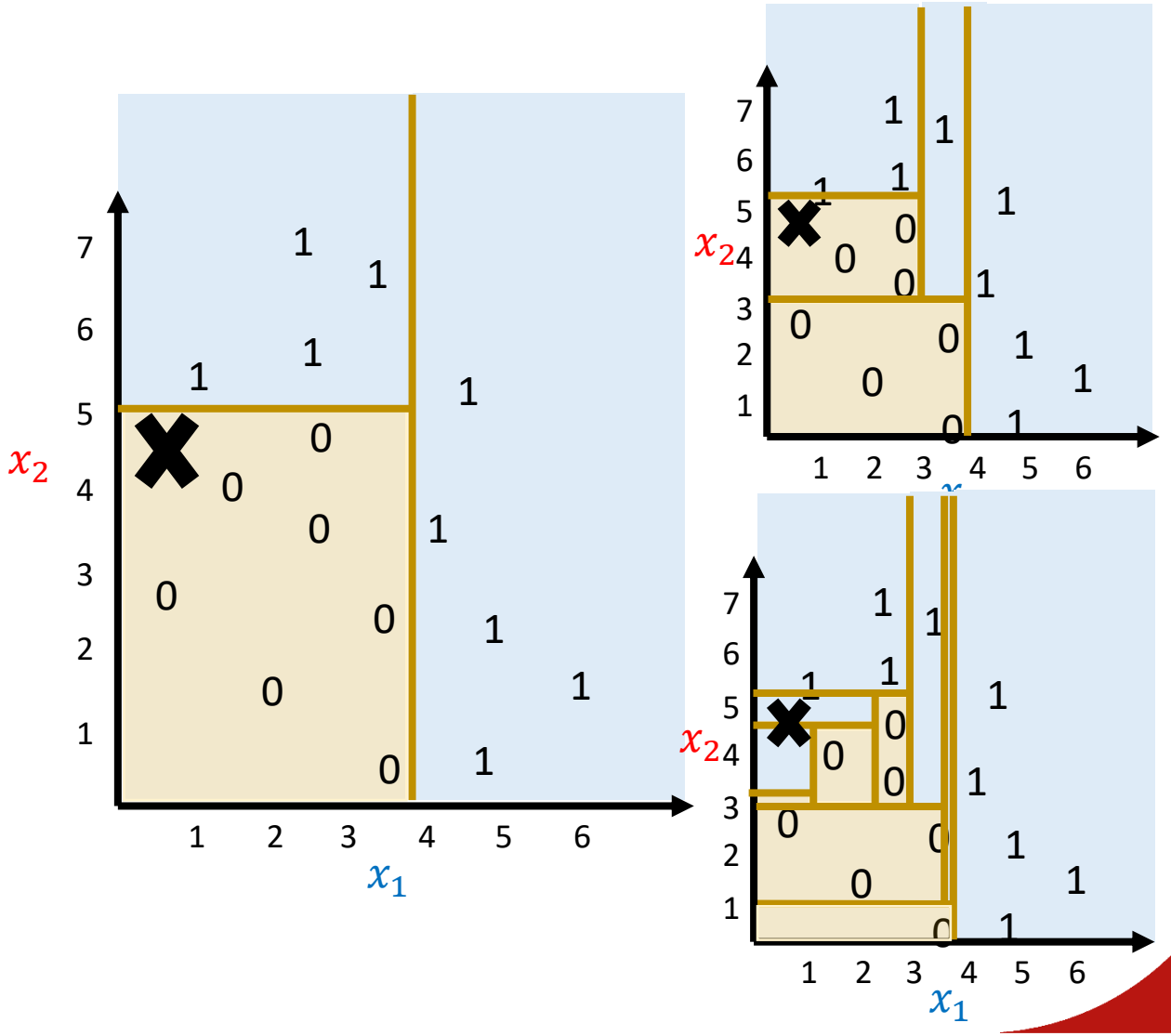
Can we partition the space in multiple ways?



Lancaster University

Is the solution unique?

- There are **exponentially many decision trees** that can be constructed from a given set of features.
- Finding the **optimal tree** is computationally infeasible because the exponential size of the search space.
- Efficient algorithms have been developed to induce a **reasonably accurate decision tree in a reasonable amount of time**.



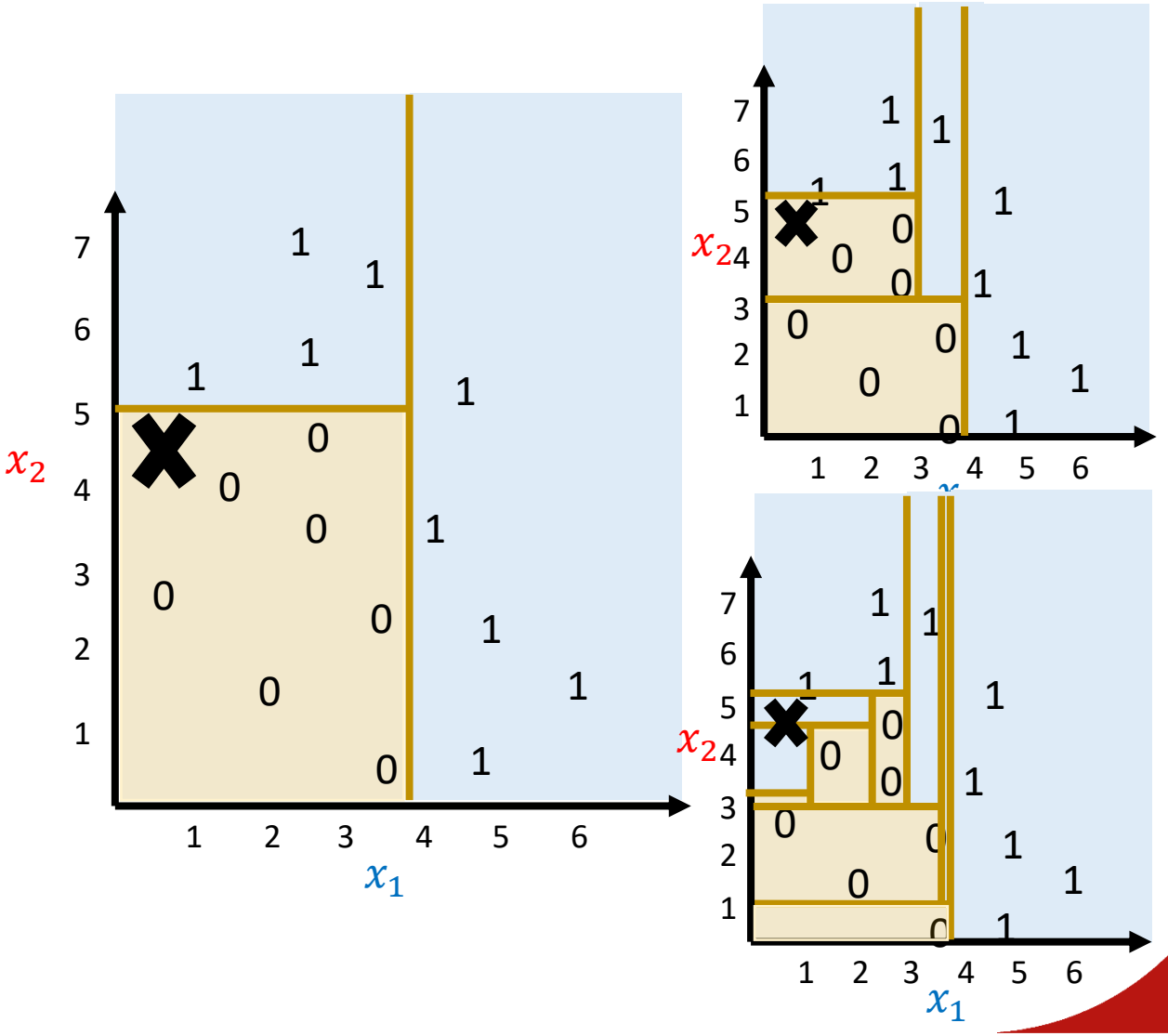
What is Good?

Accurate

- We can always build a decision such that each instance has its own leaf node, in which case the decision tree will have 100% accuracy

Small

- As small as possible
- Shallow tree, i.e., fewer tests

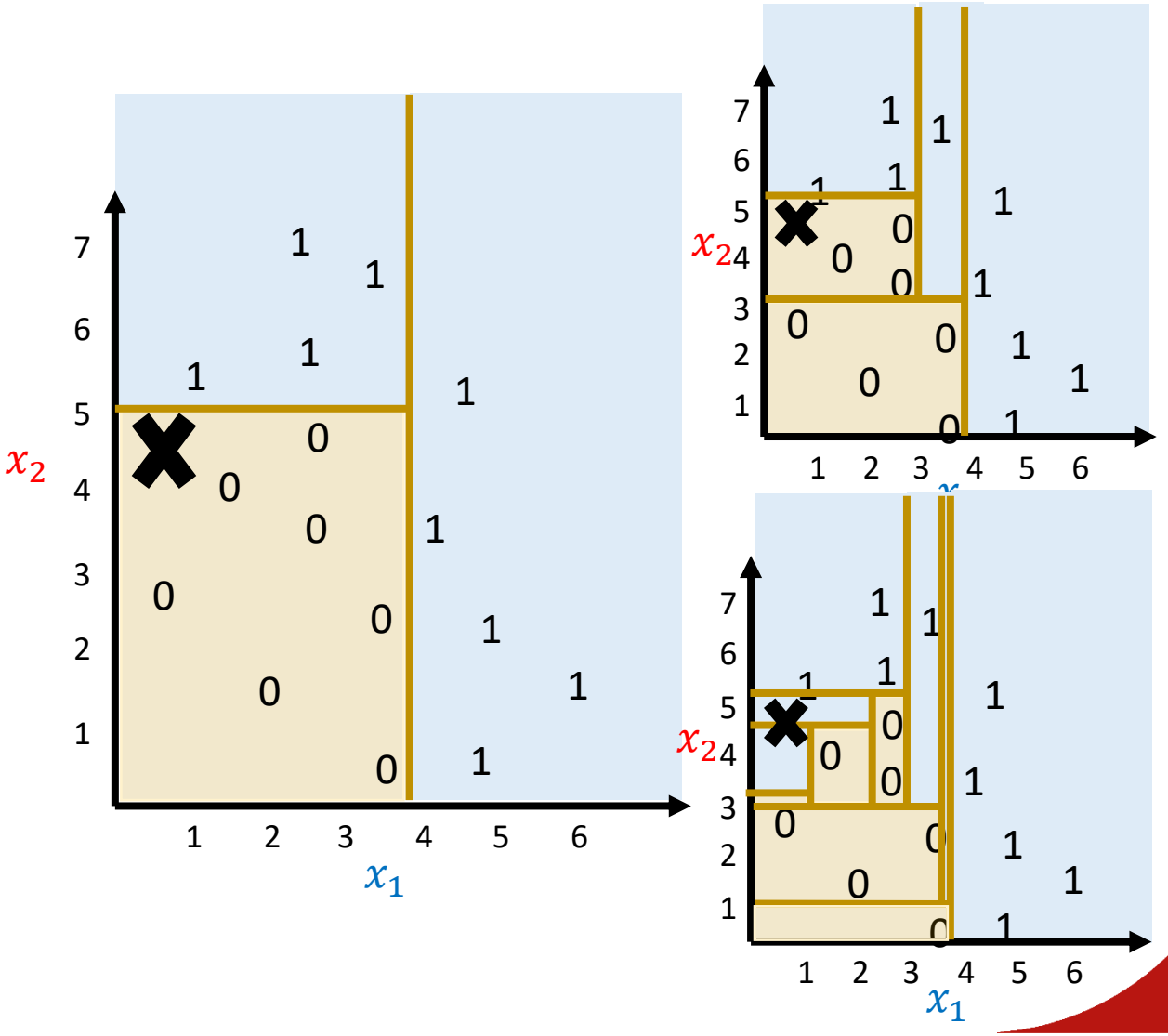


ID3 Algorithm (1983)

Test the most important feature first

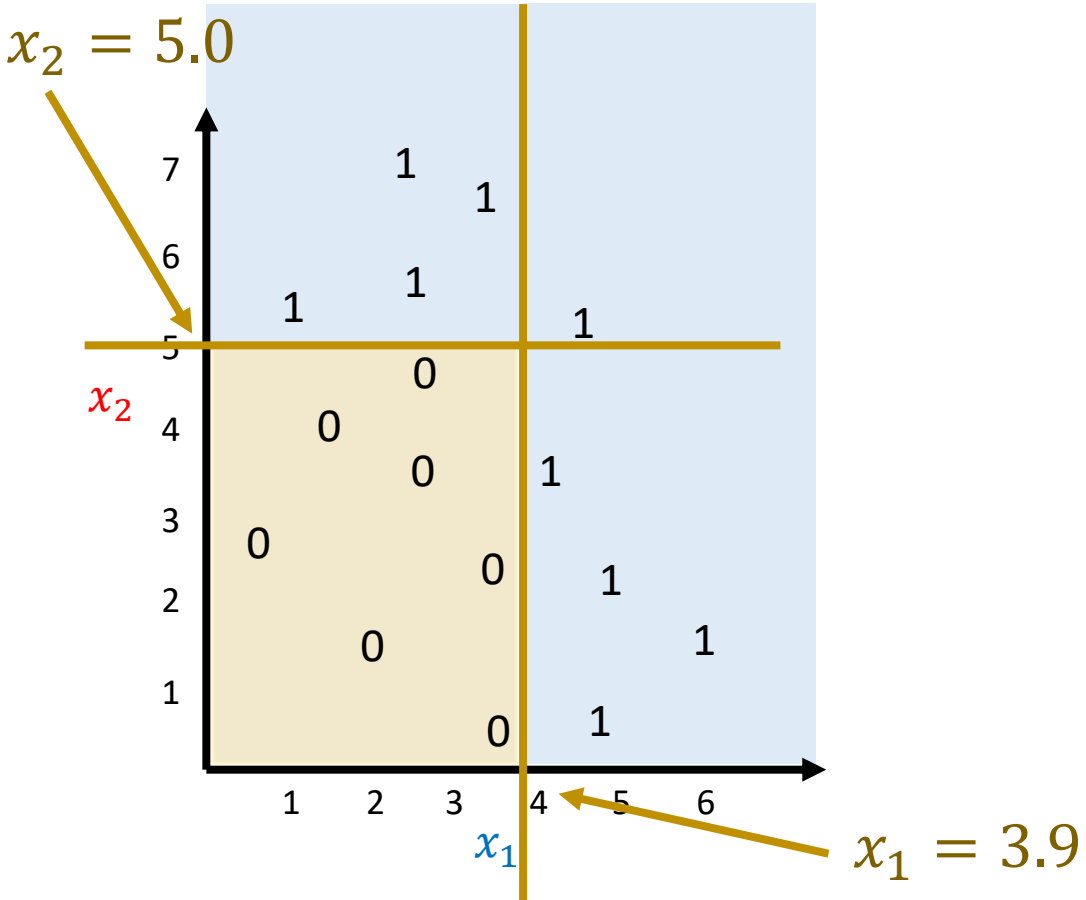
If you have only one type of example, return a leaf

Else, choose the next most important feature

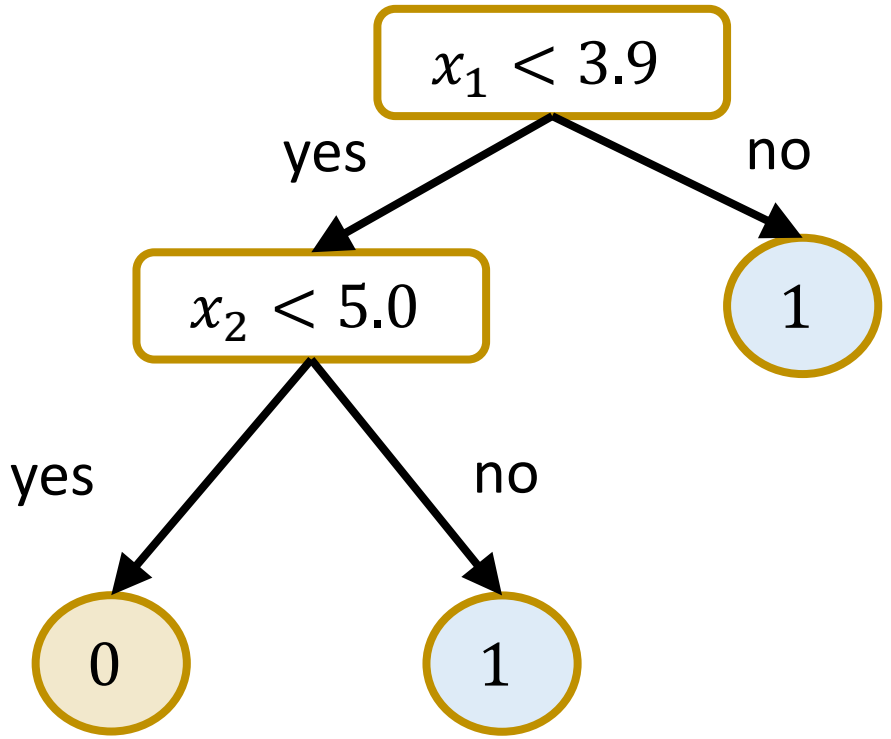


Determine the Best Feature

$$f(x_1, x_2) = \begin{cases} 0 \text{ if } x_1 < 3.9 \text{ and } x_2 < 5.0 \\ 1 \text{ otherwise} \end{cases}$$



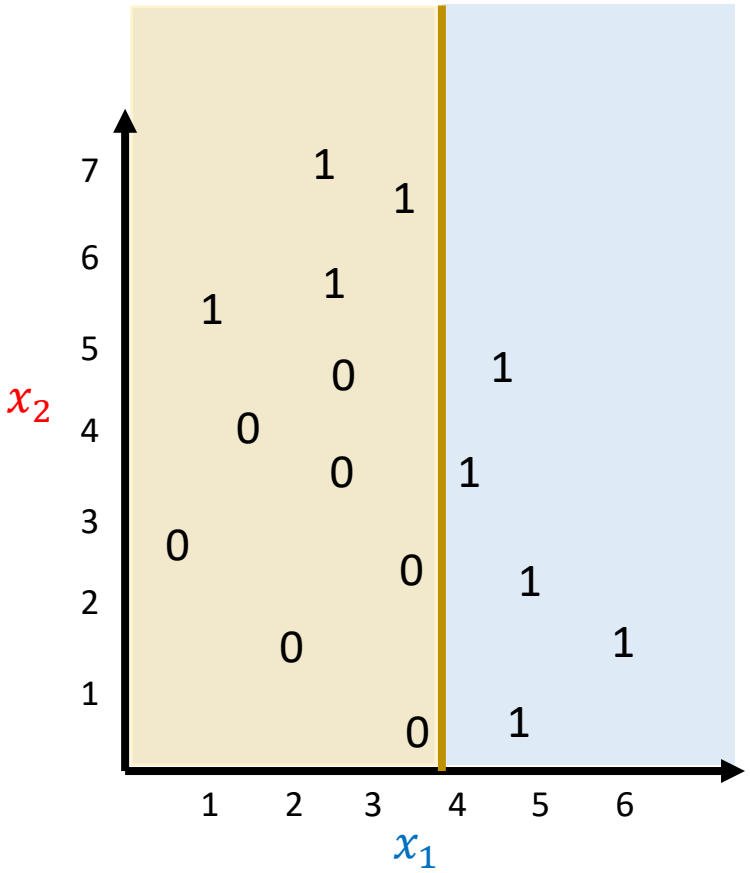
Decision Tree



Determine the Best Feature

Decision Tree #1

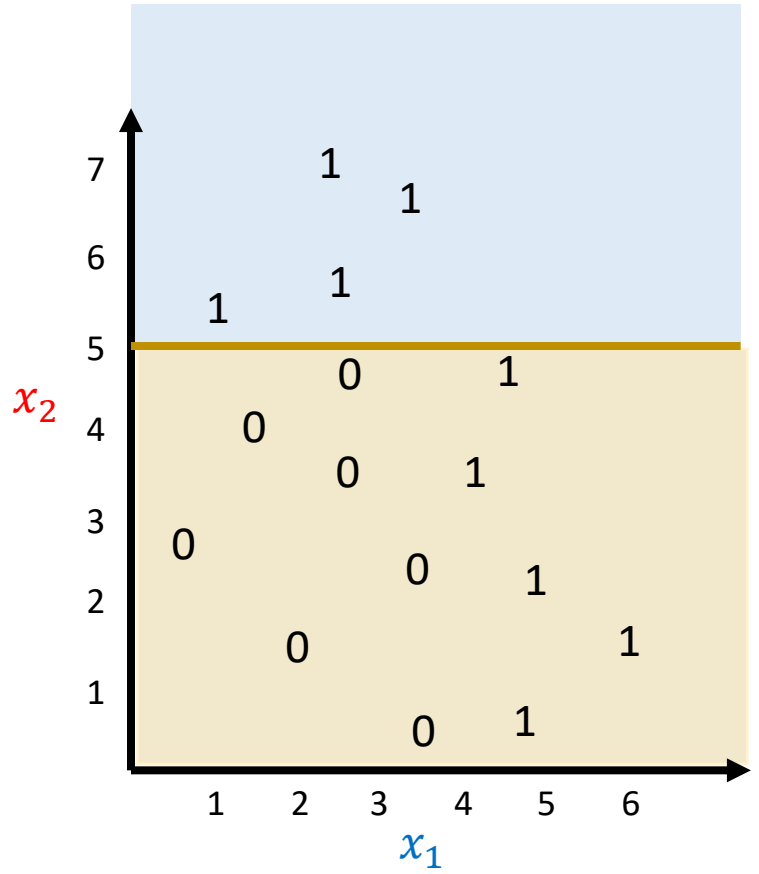
Split Feature $x_1 < 3.9$



> 3.9
0x0
5x1
< 3.9
7x0
4x1

Decision Tree #2

Split Feature $x_2 < 5.0$



> 3.9
0x0
4x1
< 3.9
7x0
5x1

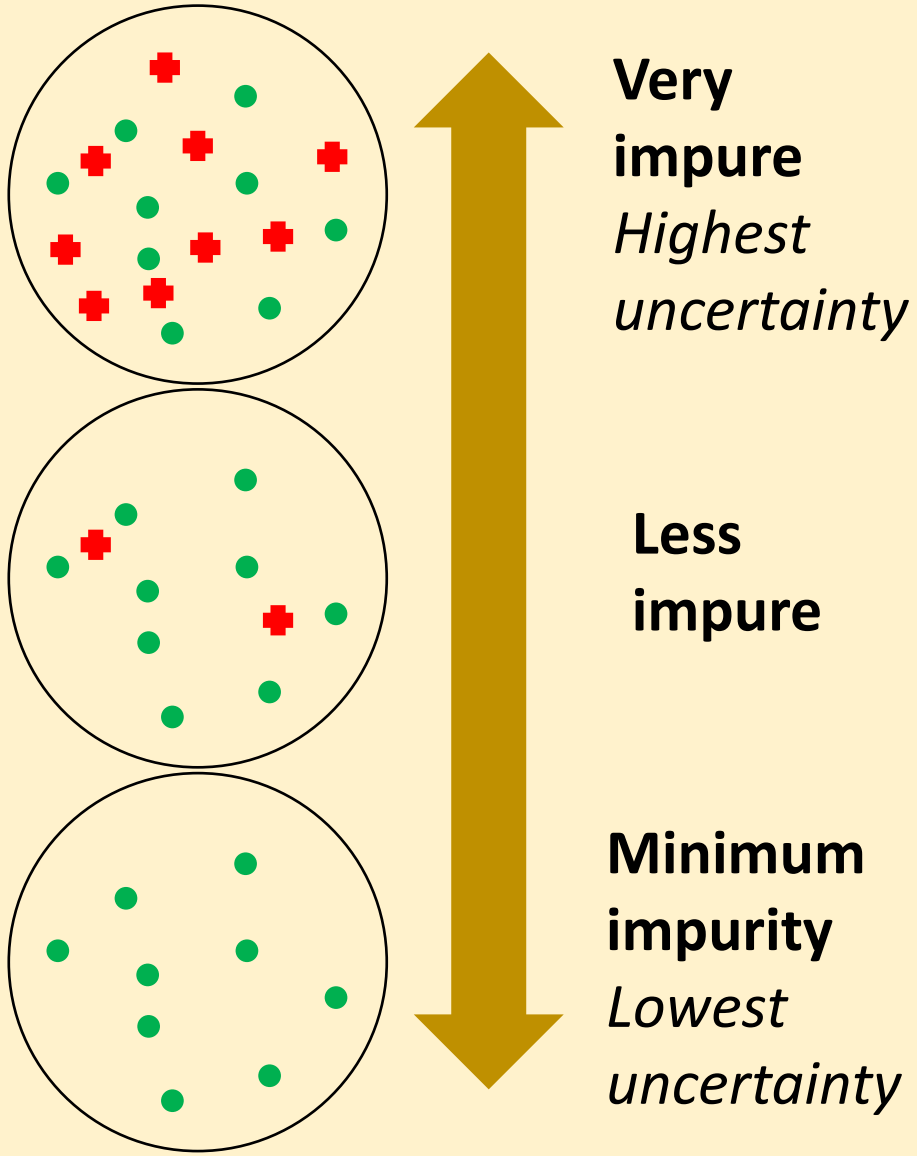
Determine the Best Feature

Good Attribute

- A good attribute splits data so that each successor node is as *pure* as possible (i.e. **reduce uncertainty and result in gain in information**)
- The distribution of samples in each node is such that it mostly contains samples of a single class

Entropy

- Measure the level of *impurity/uncertainty* in a group of samples



ASIDE: Logarithm

If $y = a^b$ then $b = \log_a y$

Special Cases:

- $a = 10$: $y = 10^b$, $b = \log_{10} y$
  - E.g. $\log_{10} 1000 = 3$ because $10^3 = 1000$
- $a = 2$: $y = 2^b$, $b = \log_2 y$
  - E.g. $\log_2 16 = 4$ because $2^4 = 16$
  - E.g. $\log_2 256 = 8$ because $2^8 = 256$
- $\log_a 1 = 0$ because $a^0 := 1$

Hence: 8-bit images

Entropy (Binary)

Let $S = S_1 \cup S_2$ denote a set of samples, $S_1 \cap S_2 = \emptyset$

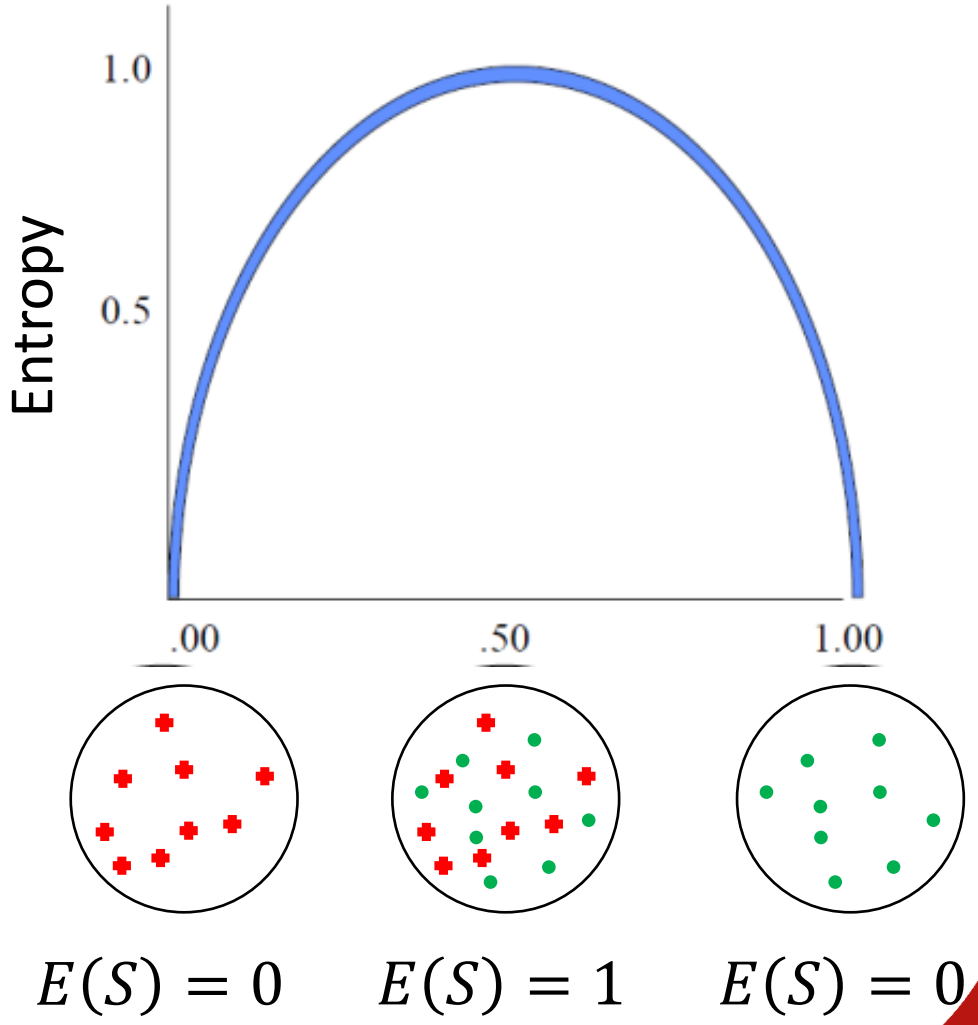
Let

$$P_1 = \frac{|S_1|}{|S|}, \qquad P_2 = \frac{|S_2|}{|S|}$$

Entropy $E(S)$ is a measure of uncertainty.

- Highest when uncertainty is greatest

$$E(S) = -P_1 \log_2 P_1 - P_2 \log_2 P_2$$

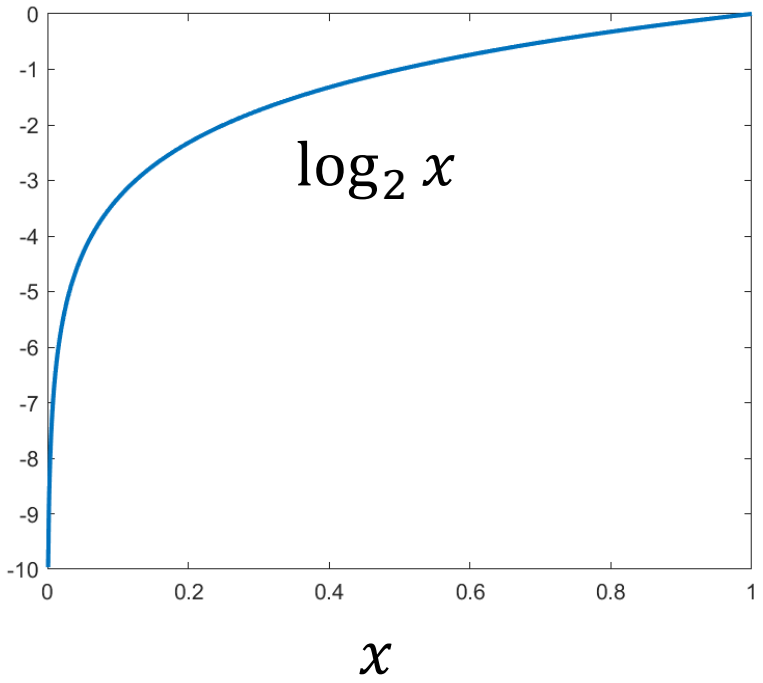


Entropy (Binary)

$$E(S) = -P_1 \log_2 P_1 - P_2 \log_2 P_2$$

We know:

$$0 \leq P_1, P_2 \leq 1$$

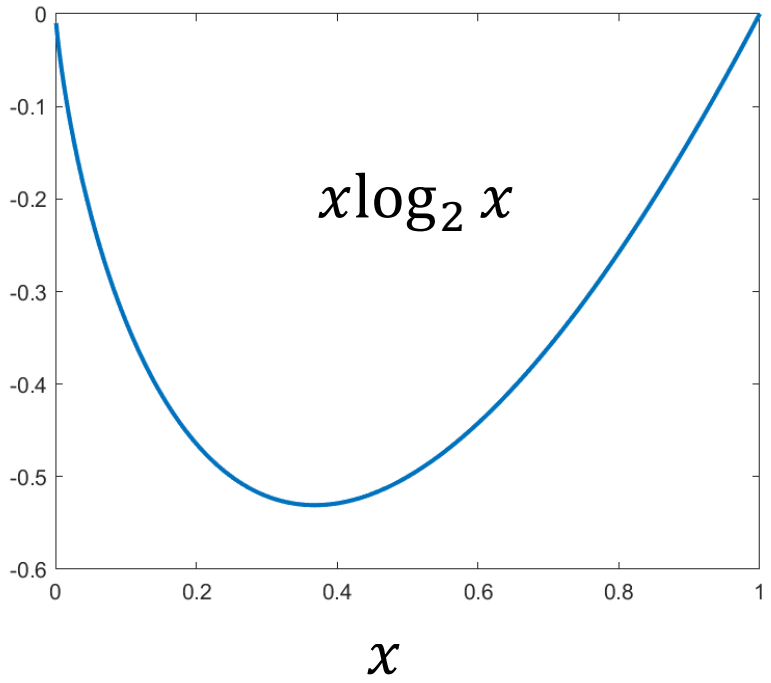


Entropy (Binary)

$$E(S) = -P_1 \log_2 P_1 - P_2 \log_2 P_2$$

We know:

$$0 \leq P_1, P_2 \leq 1$$

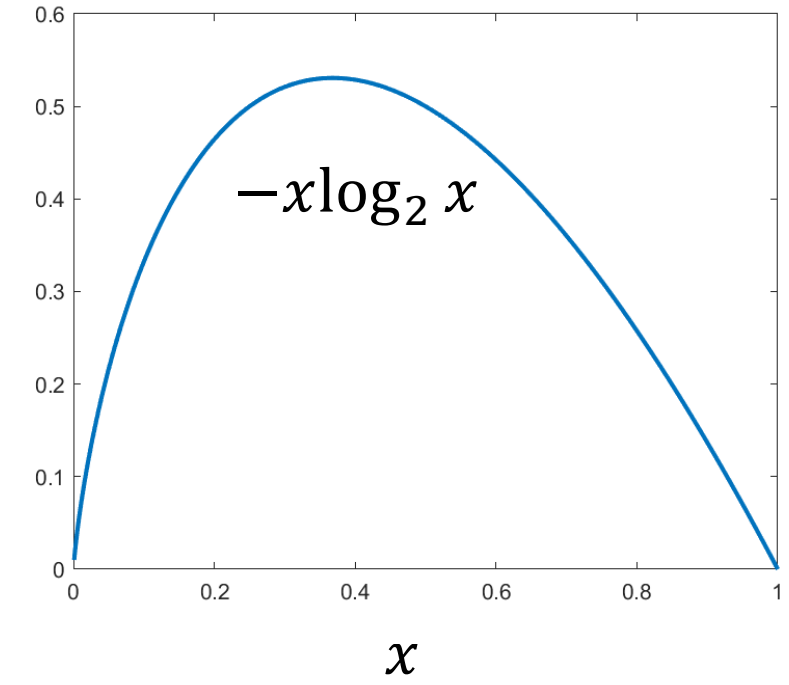


Entropy (Binary)

$$E(S) = -P_1 \log_2 P_1 - P_2 \log_2 P_2$$

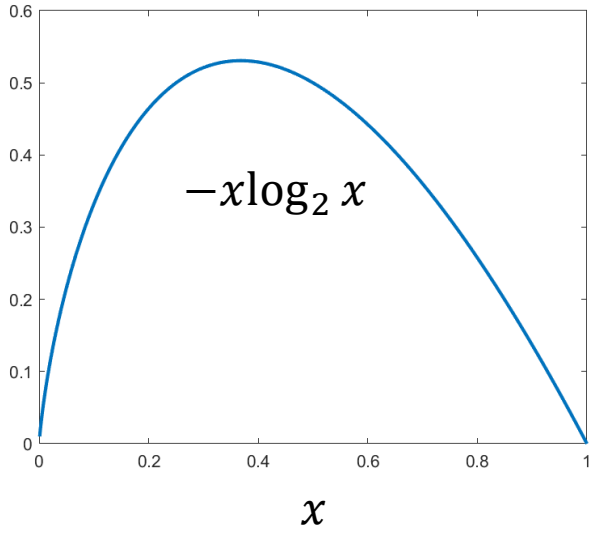
We know:

$$0 \leq P_1, P_2 \leq 1$$



Entropy (Binary)

$$E(S) = -P_1 \log_2 P_1 - P_2 \log_2 P_2$$



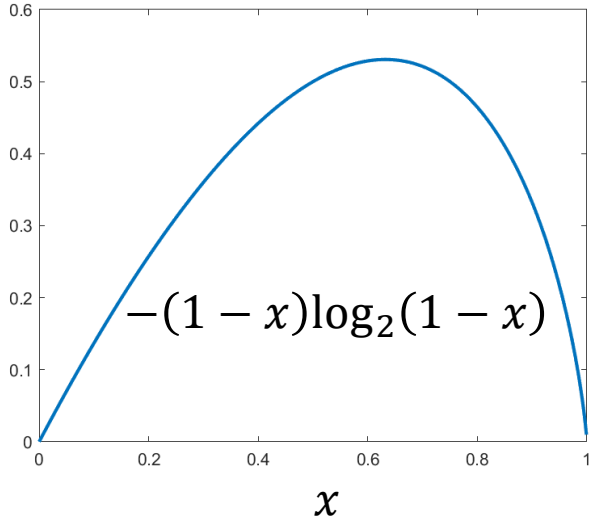
We know:

$$0 \leq P_1, P_2 \leq 1$$

$$|S_1| + |S_2| = |S|$$

$$\Rightarrow P_2 = \frac{|S_2|}{|S|} = \frac{|S| - |S_1|}{|S|} = \frac{|S|}{|S|} - \frac{|S_1|}{|S|} = 1 - P_1$$

$$E(S) = -P_1 \log_2 P_1 - (1 - P_1)\log_2(1 - P_1)$$



Lancaster University

Entropy (Binary)

$$E(S) = -P_1 \log_2 P_1 - P_2 \log_2 P_2$$

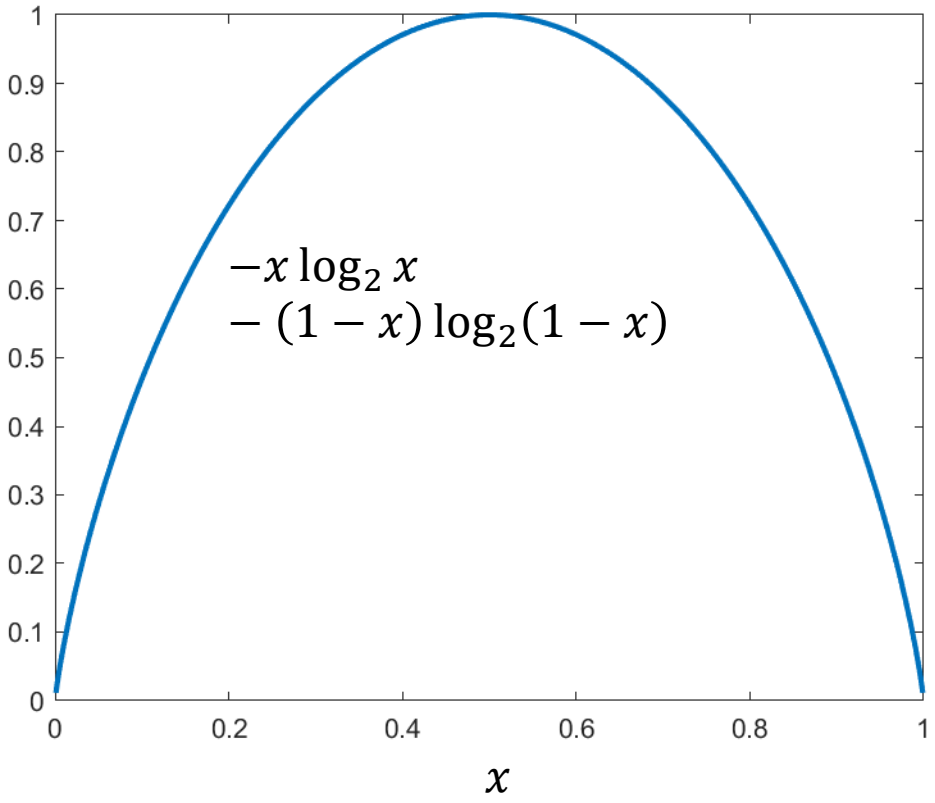
We know:

$$0 \leq P_1, P_2 \leq 1$$

$$|S_1| + |S_2| = |S|$$

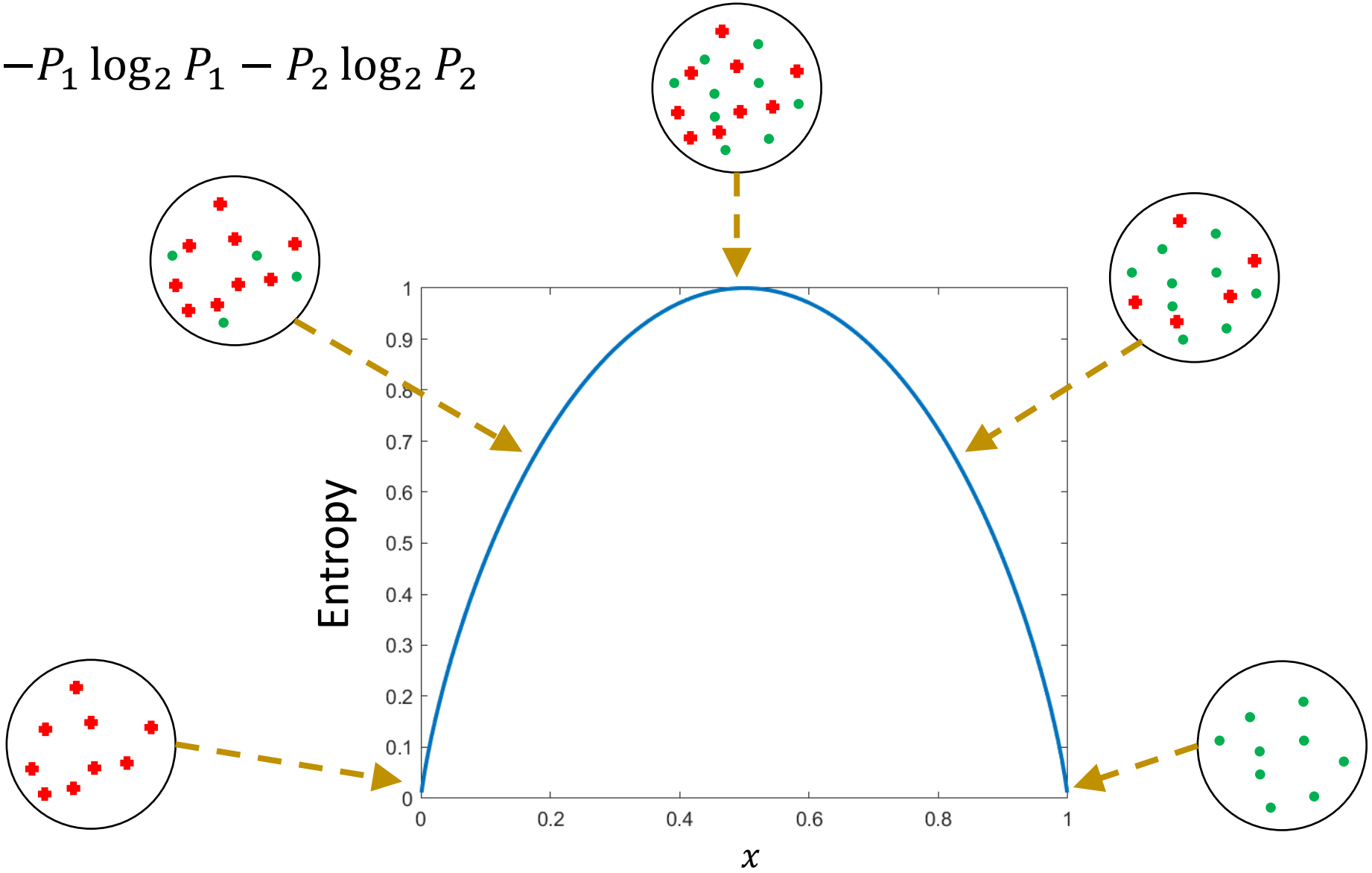
$$\Rightarrow P_2 = \frac{|S_2|}{|S|} = \frac{|S| - |S_1|}{|S|} = \frac{|S|}{|S|} - \frac{|S_1|}{|S|} = 1 - P_1$$

$$E(S) = \boxed{-P_1 \log_2 P_1 - (1 - P_1) \log_2 (1 - P_1)}$$



Entropy (Binary)

$$E(S) = -P_1 \log_2 P_1 - P_2 \log_2 P_2$$



Entropy (Non-Binary)

If we only have two classes, we calculate entropy as:

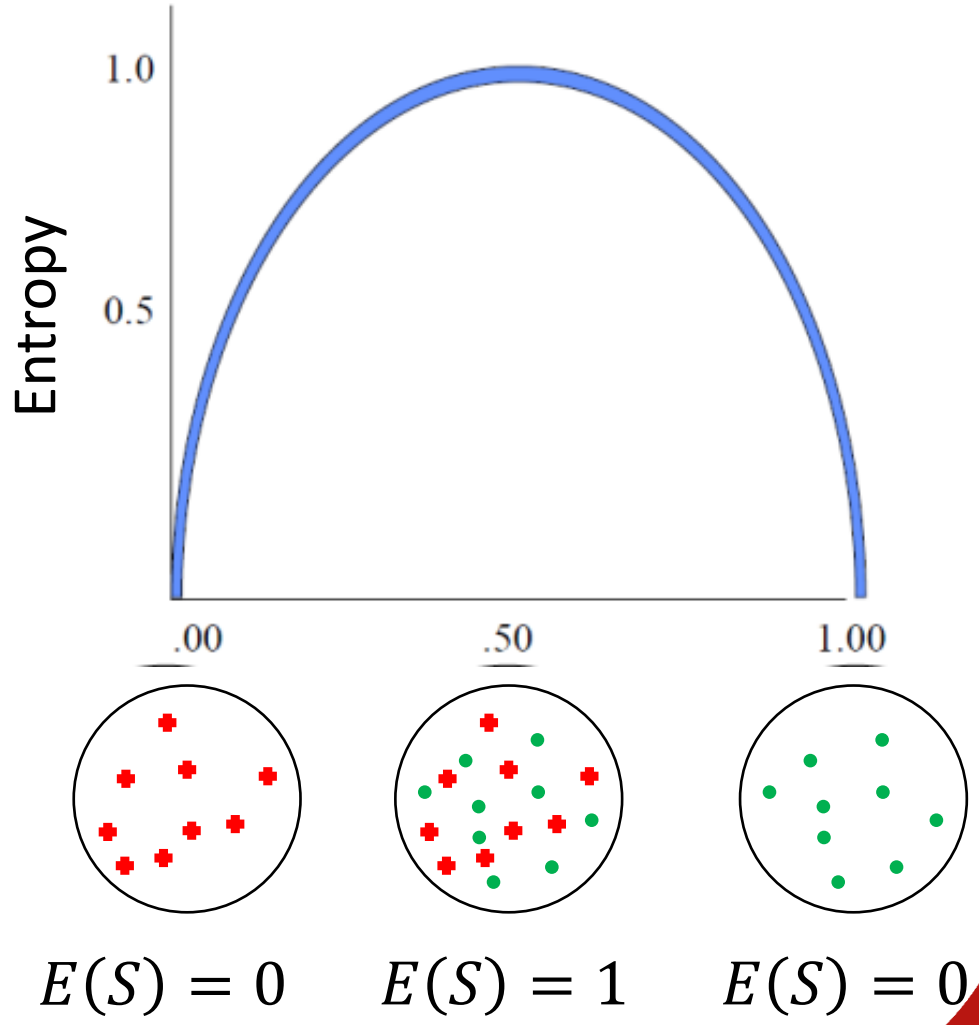
$$E(S) = -P_1 \log_2 P_1 - P_2 \log_2 P_2$$

Extension to n classes:

Let $S = \cup_i S_i = S_1 \cup \cdots \cup S_n$ where $\cap_i S_i = \emptyset$

Then **entropy** is defined as

$$E(S) = -\sum_{i=1}^{n} P_i \log_2 P_i$$



Entropy of our Examples

$S_0 =$ Set of 0s, $S_1 =$ Set of 1s

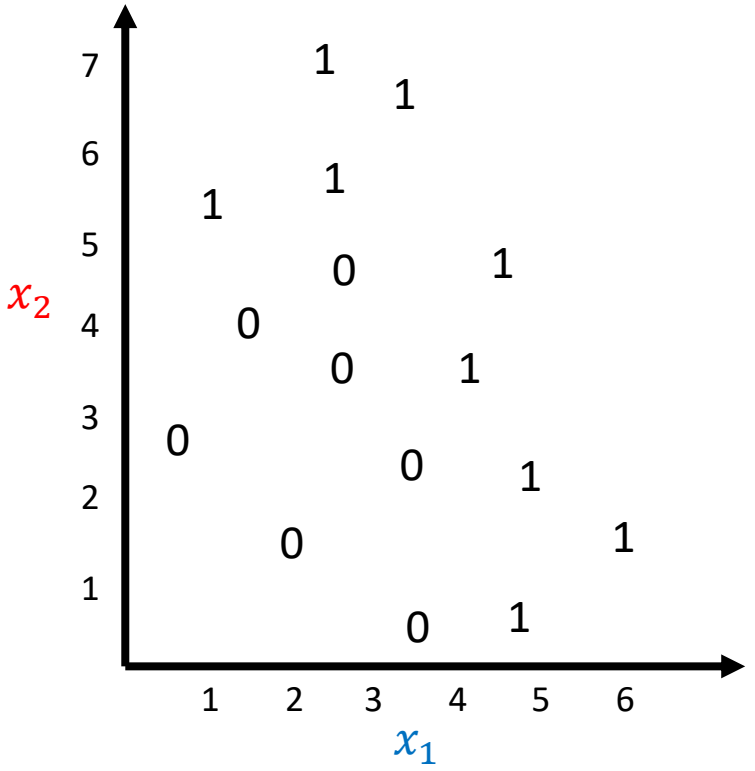
Calculate the Entropy:

$$|S| = 16,\ |S_0| = 7,\ |S_1| = 9$$

$$P_1 = \frac{7}{16}, \qquad P_2 = \frac{9}{16}$$

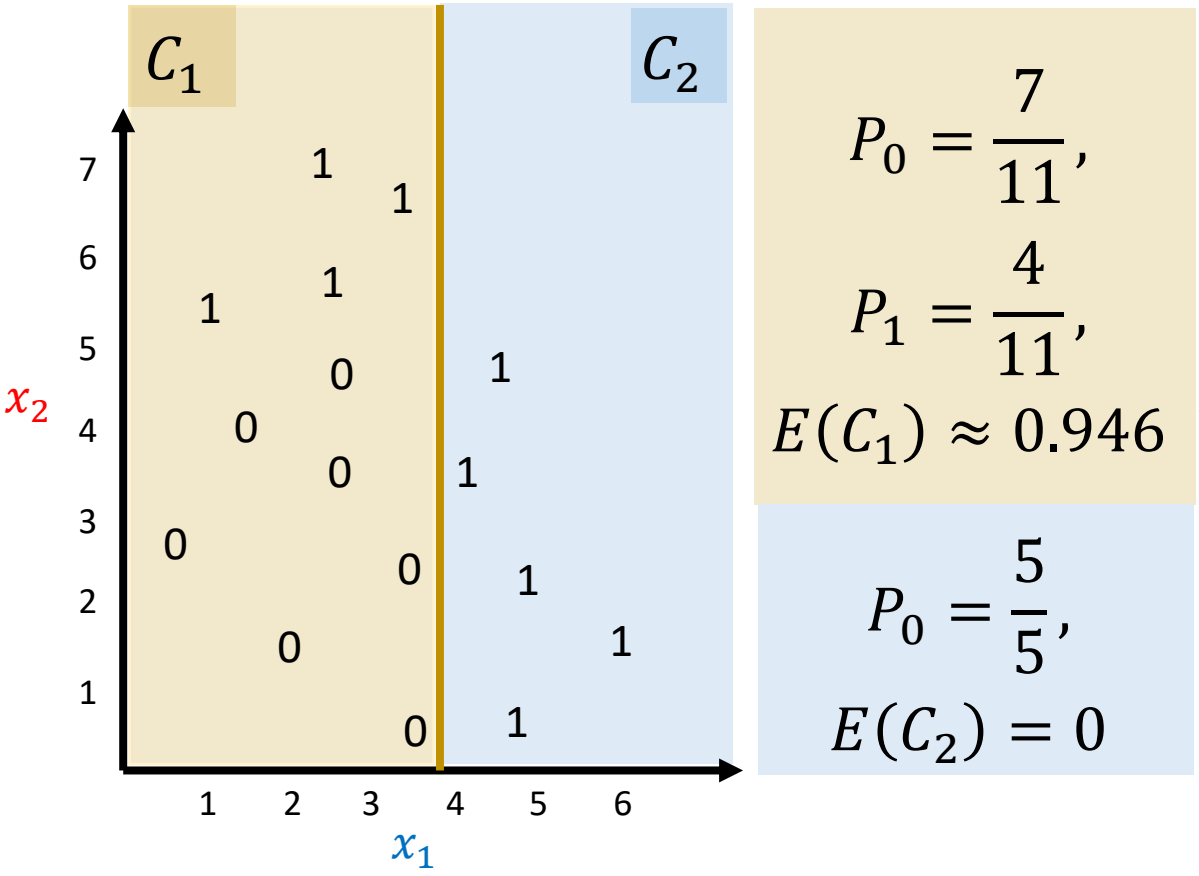
$$E(S) = -P_1 \log_2 P_1 - P_2 \log_2 P_2$$

$$= -\frac{7}{16}\log_2 \frac{7}{16} - \frac{9}{16}\log_2 \frac{9}{16} \approx 0.989$$



Entropy of our Examples – Child Nodes

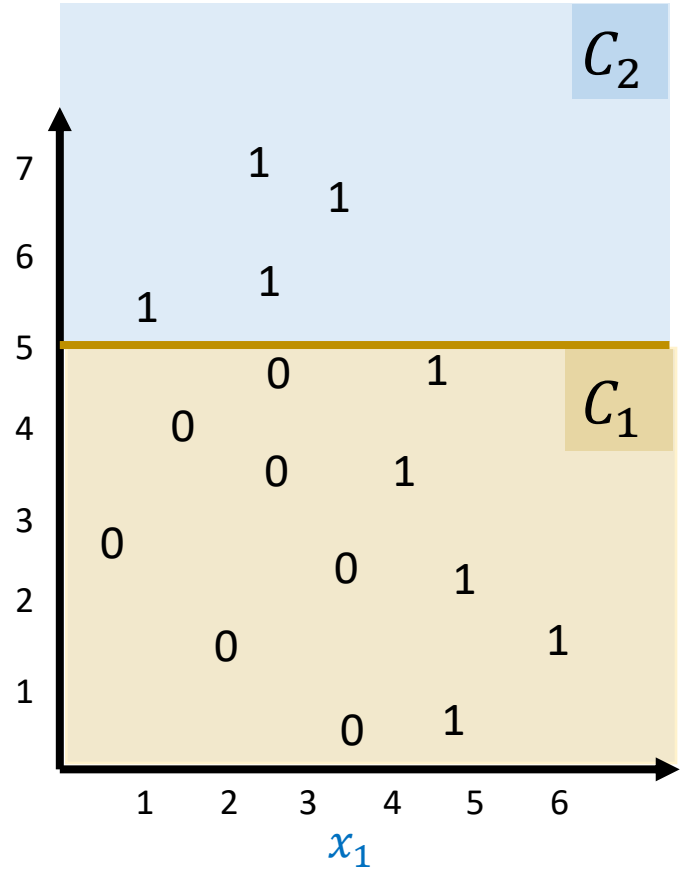
Decision Tree #1: Split Feature $x_1 < 3.9$



$$P_0 = \frac{7}{11}, \quad P_1 = \frac{4}{11}, \quad E(C_1) \approx 0.946$$

$$P_0 = \frac{5}{5}, \quad E(C_2) = 0$$

Decision Tree #2: Split Feature $x_2 < 5.0$

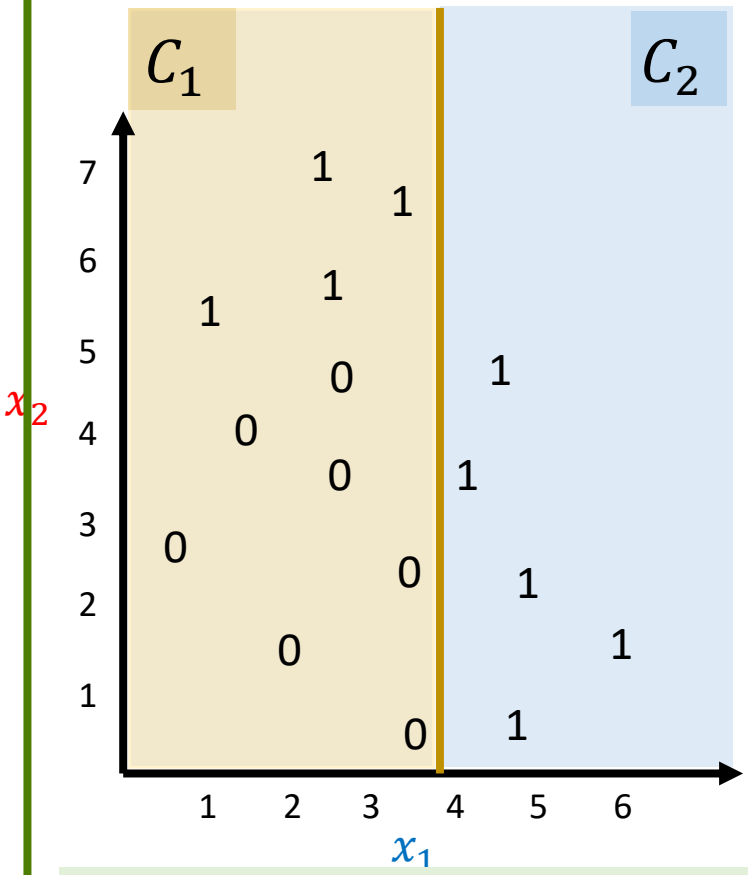


$$P_0 = \frac{7}{12}, \quad P_1 = \frac{5}{12}, \quad E(C_1) \approx 0.980$$

$$P_0 = \frac{4}{4}, \quad E(C_2) = 0$$

Entropy of our Examples – Child Nodes

Decision Tree #1: Split Feature $x_1 < 3.9$

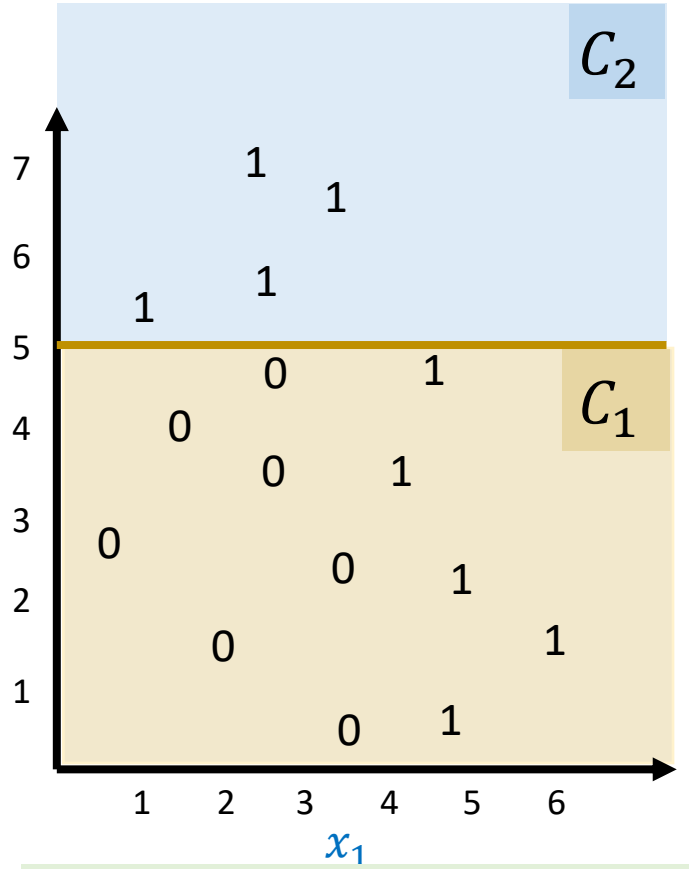


$$P_0 = \frac{7}{11}, \quad P_1 = \frac{4}{11}, \quad E(C_1) \approx 0.946$$

$$P_0 = \frac{5}{5}, \quad E(C_2) = 0$$

$$E(S) = \frac{|C_1|E(C_1) + |C_2|E(C_2)}{|S|} = 0.650$$

Decision Tree #2: Split Feature $x_2 < 5.0$



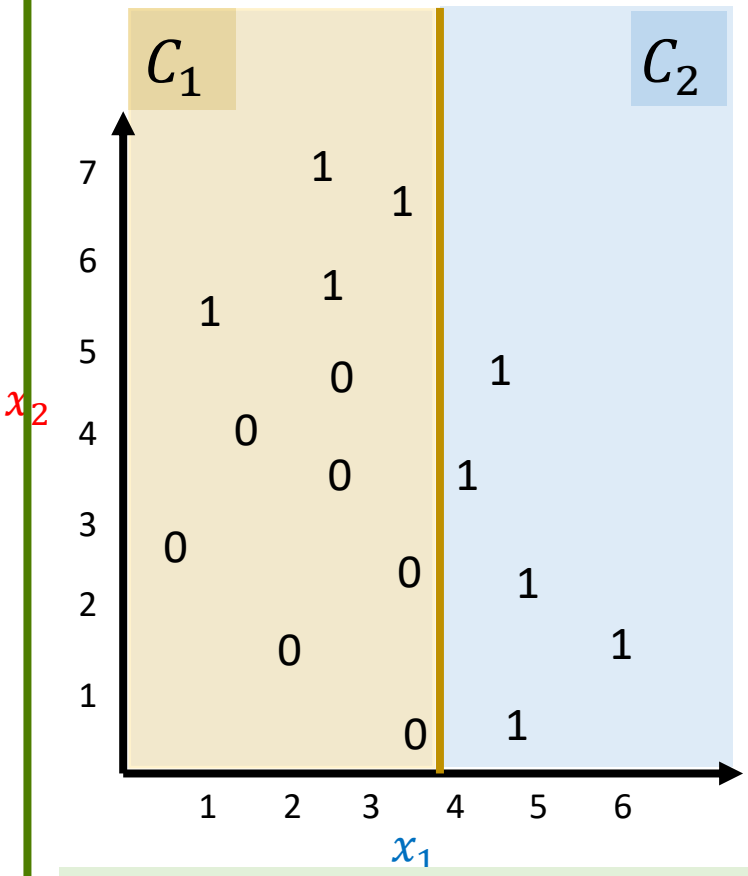
$$P_0 = \frac{7}{12}, \quad P_1 = \frac{5}{12}, \quad E(C_1) \approx 0.980$$

$$P_0 = \frac{4}{4}, \quad E(C_2) = 0$$

$$E(S) = \frac{|C_1|E(C_1) + |C_2|E(C_2)}{|S|} = 0.735$$

Entropy of our Examples – Child Nodes

Decision Tree #1: Split Feature $x_1 < 3.9$



$$P_0 = \frac{7}{11},$$

$$P_1 = \frac{4}{11},$$

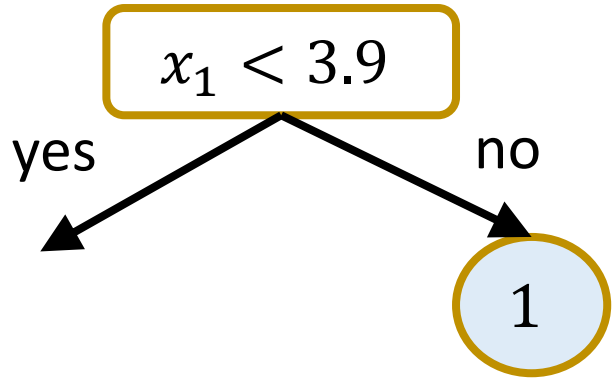
$$E(C_1) \approx 0.946$$

$$P_0 = \frac{5}{5},$$

$$E(C_2) = 0$$

$$E(S) = \frac{|C_1|E(C_1) + |C_2|E(C_2)}{|S|} = 0.650$$

Decision Tree



$E(S) \approx 0.989$

$E(S) \approx 0.650$

Lancaster University

Information Gain

Information Gain, also known as “expected reduction in entropy” is given as

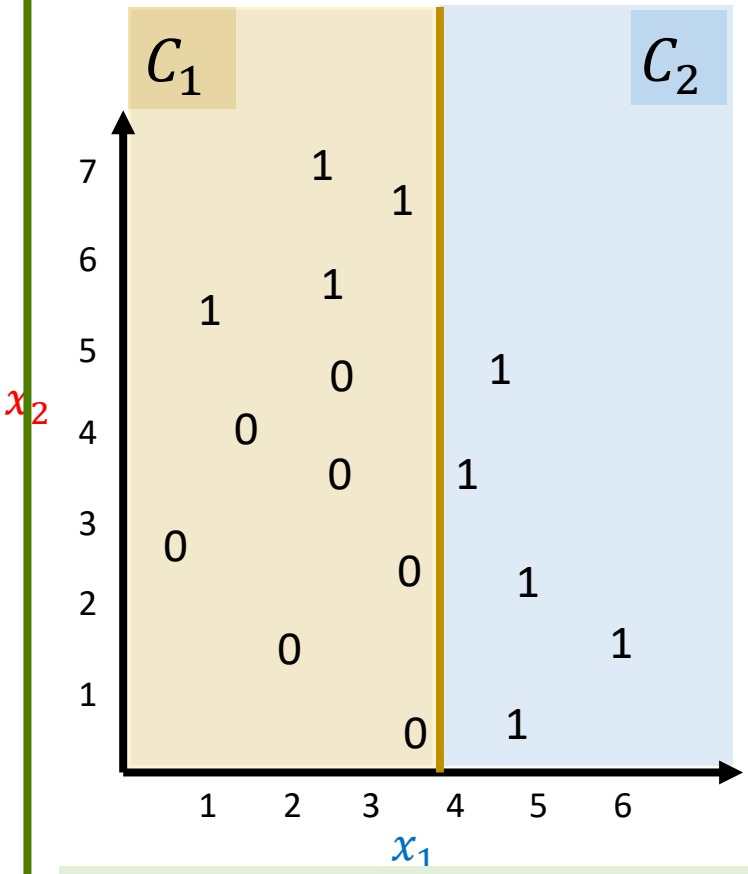
Gain(S,condition) = Entropy(parent) – [Weighted Average] Entropy(Children)

Aim

- Decision tree algorithm will **maximize information gain**
- For every node in a Decision Tree, select a feature which maximize information gain

Information Gain

Decision Tree #1: Split Feature $x_1 < 3.9$



$$P_0 = \frac{7}{11},$$

$$P_1 = \frac{4}{11},$$

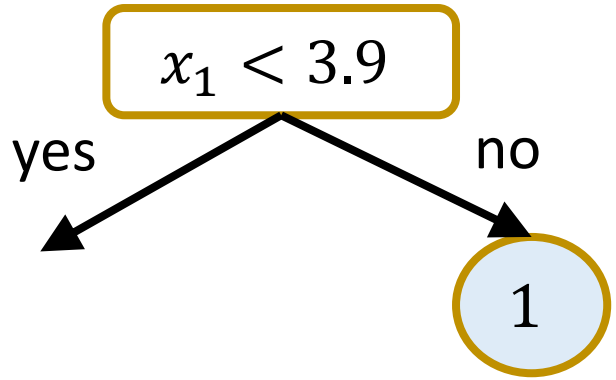
$$E(C_1) \approx 0.946$$

$$P_0 = \frac{5}{5},$$

$$E(C_2) = 0$$

$$E(S) = \frac{|C_1|E(C_1) + |C_2|E(C_2)}{|S|} = 0.650$$

Decision Tree



$$E(S) \approx 0.989$$

$$E(S) \approx 0.650$$

Information Gain:

$$\text{Gain}(S, x_1 < 3.9) = 0.989 - 0.650 = 0.339$$

Decision Trees Pros and Cons

Advantages

- Can be applied to the data from **any distribution**. E.g. data does not have to be separable with a linear boundary
- Simple to **understand and interpret**
- Able to handle both **numerical and categorical** data
- Extremely **fast**

Disadvantages

- Trees can be **ill-posed**: A small change in the training data can result in a large change in the tree and consequently the final predictions
- The problem of learning an optimal decision tree is known to be **NP-complete**.
- Decision trees are **prone to overfitting**, especially when a tree is particularly deep.

Overfitting

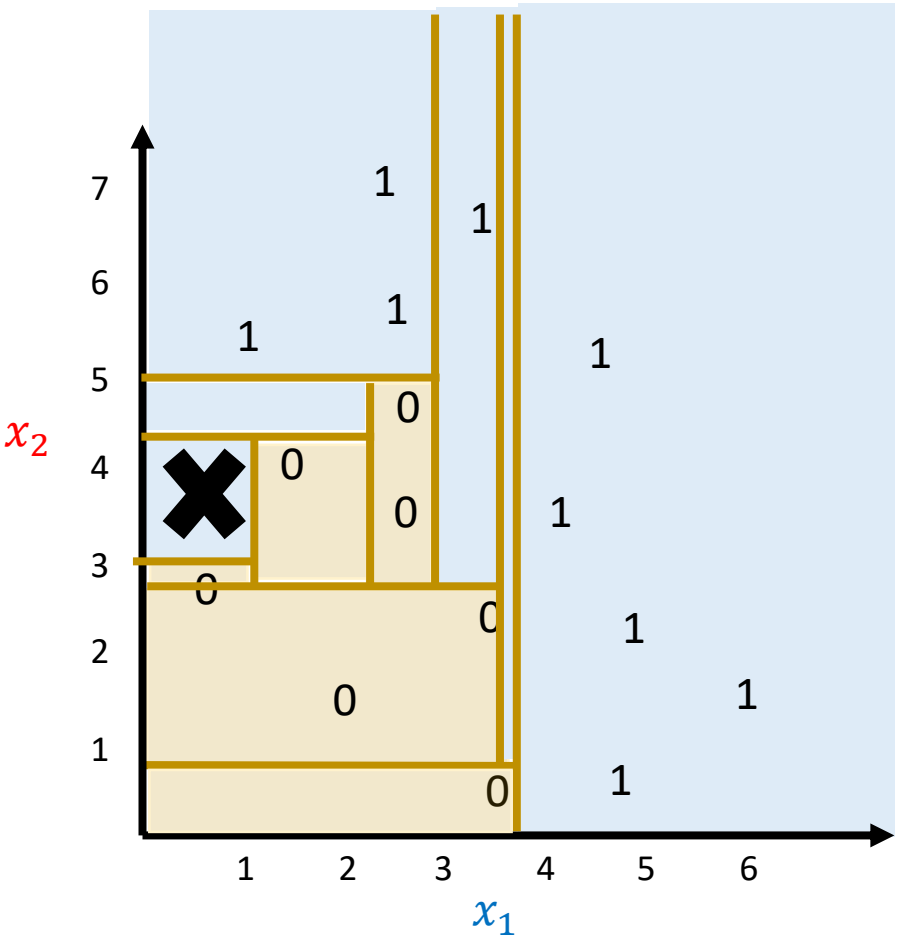
Overfitting happens when your model fits too well to the training set.

Overfitting means memorizing

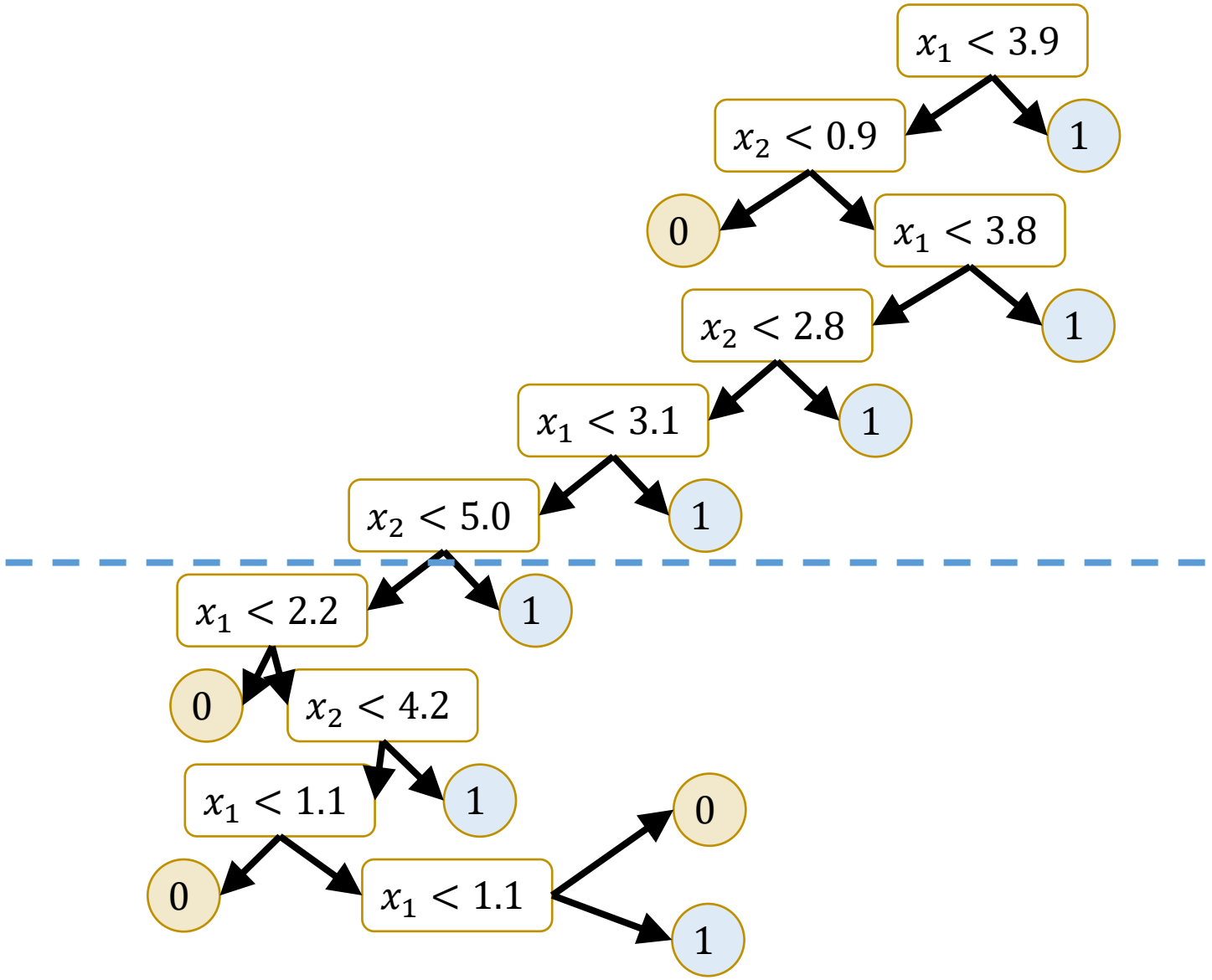
Memorizing is not learning!

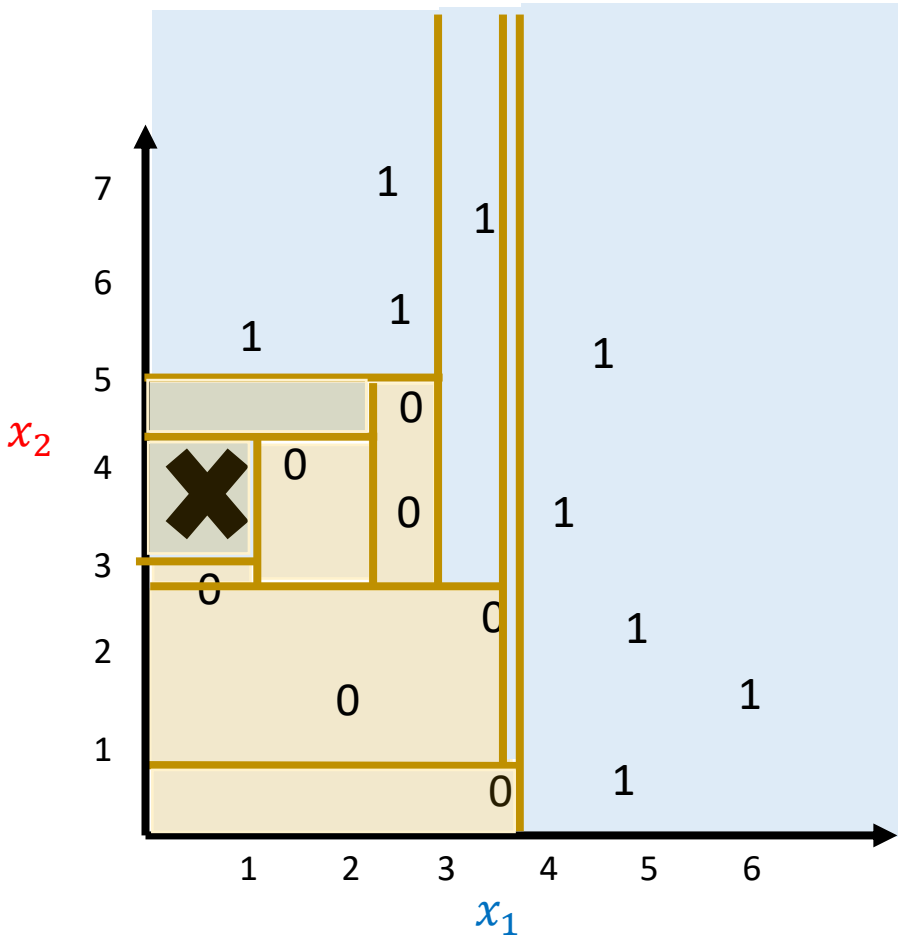
It then becomes difficult for the model to generalize to new examples that were not in the training set.

For example, your model recognizes specific images in your training set instead of general patterns.



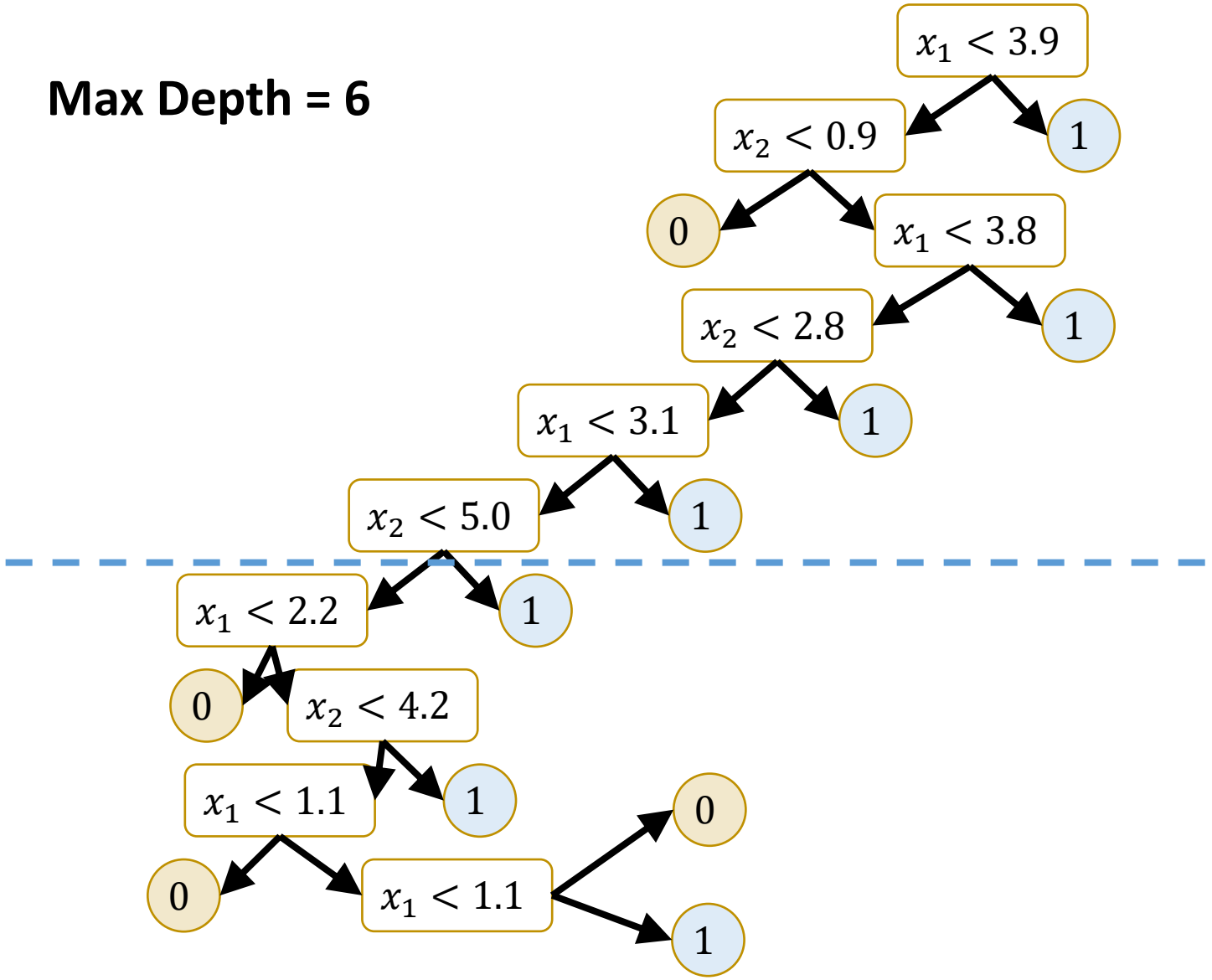
Avoiding Overfitting

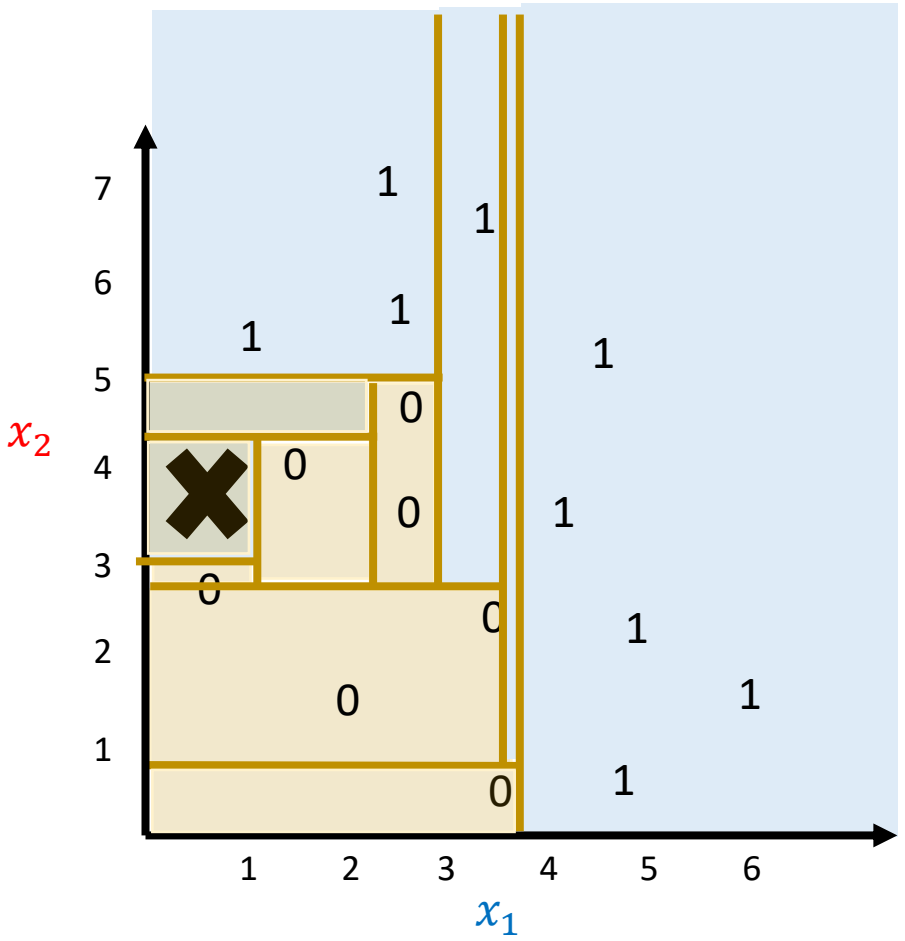




Avoiding Overfitting: Set Max Depth

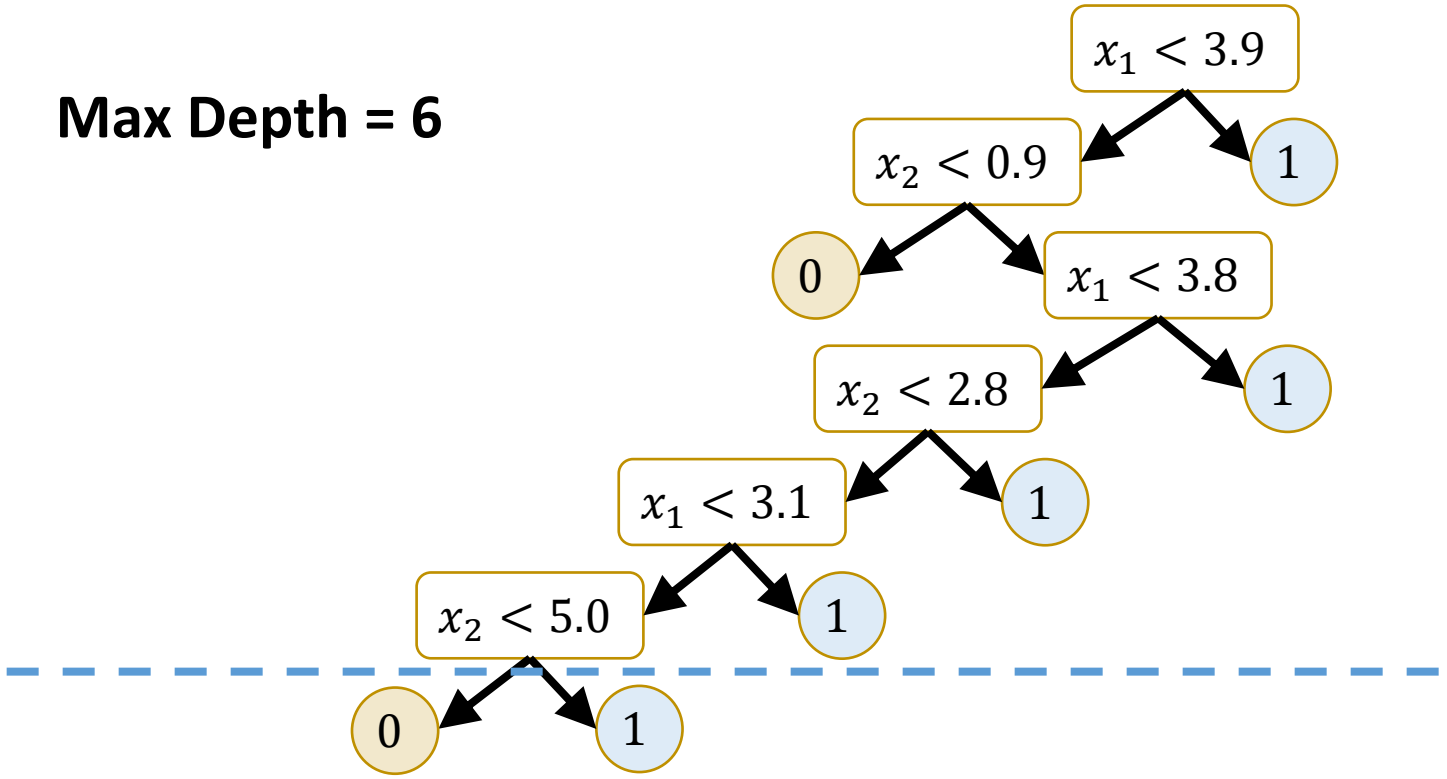
Max Depth = 6

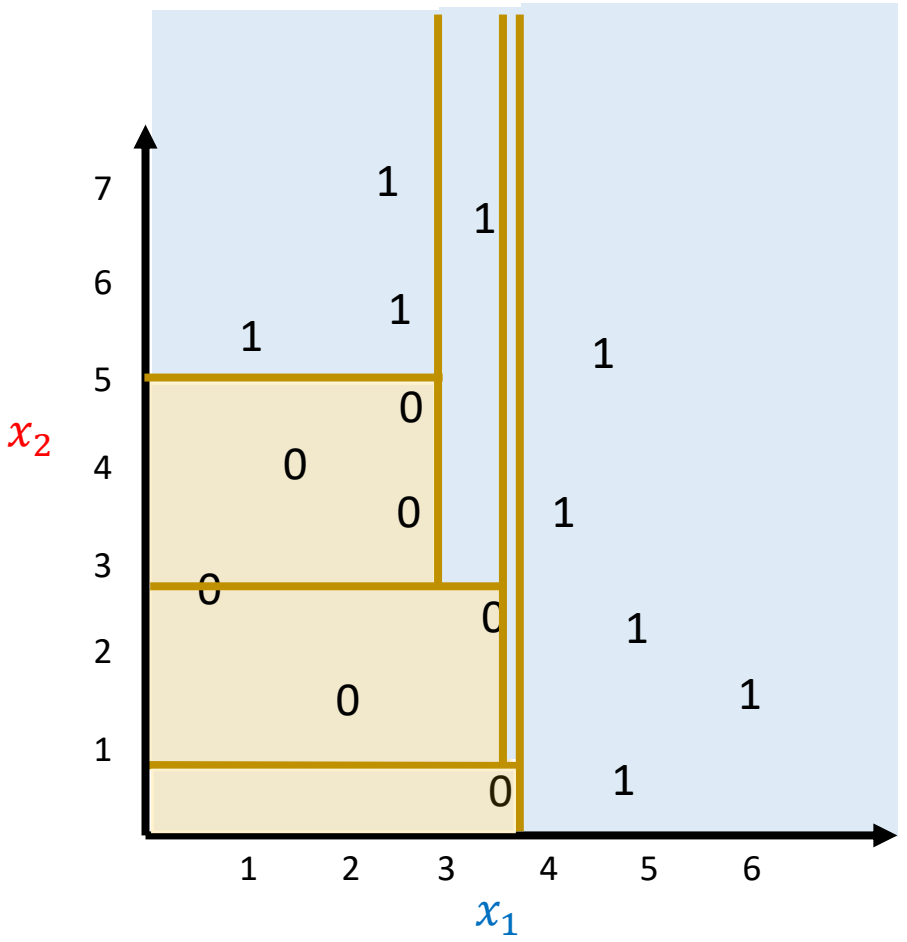




Avoiding Overfitting: Set Max Depth

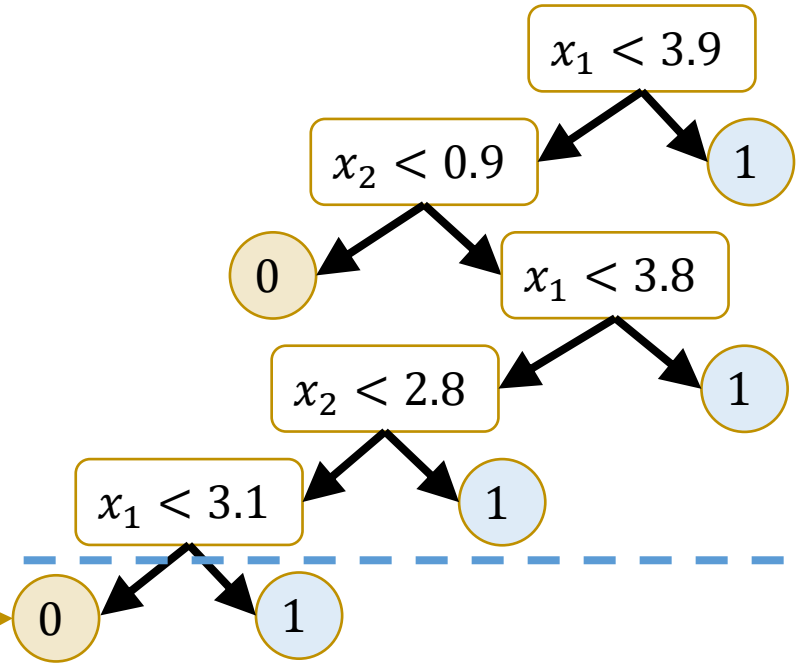
Max Depth = 6





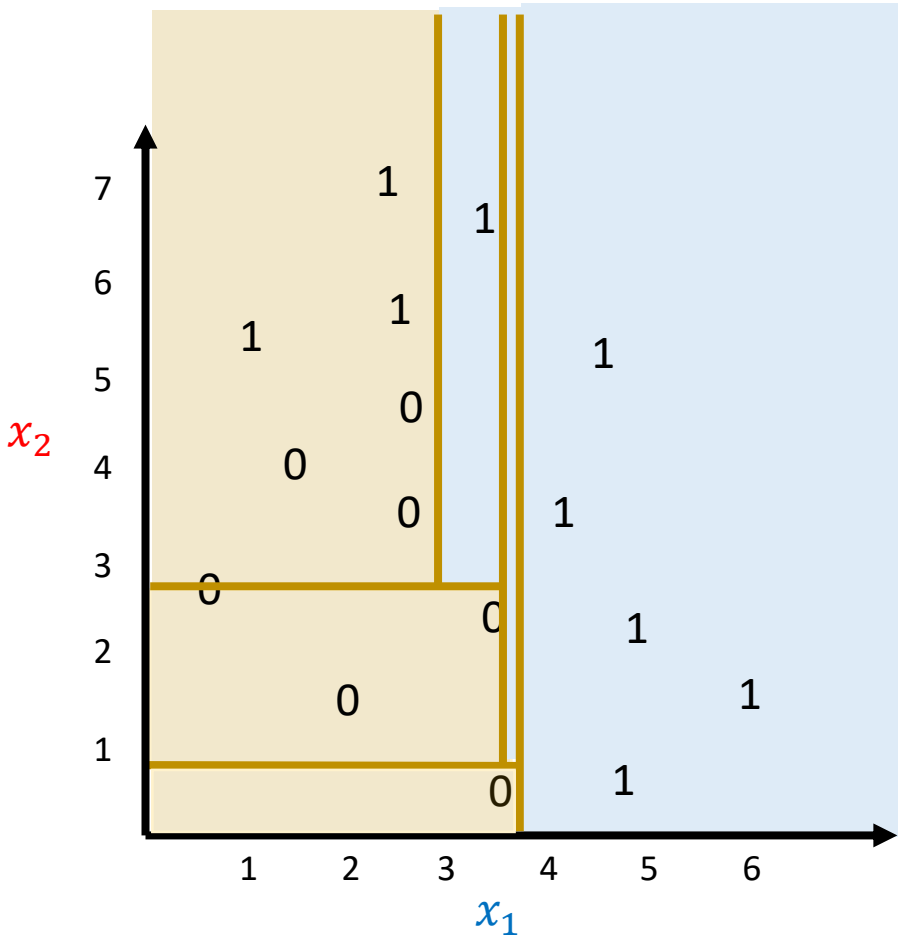
Avoiding Overfitting: Set Max Depth

Max Depth = 5



$$P(class = 0) = \frac{3}{6} = 0.5$$

$$P(class = 1) = \frac{3}{6} = 0.5$$



Lancaster University

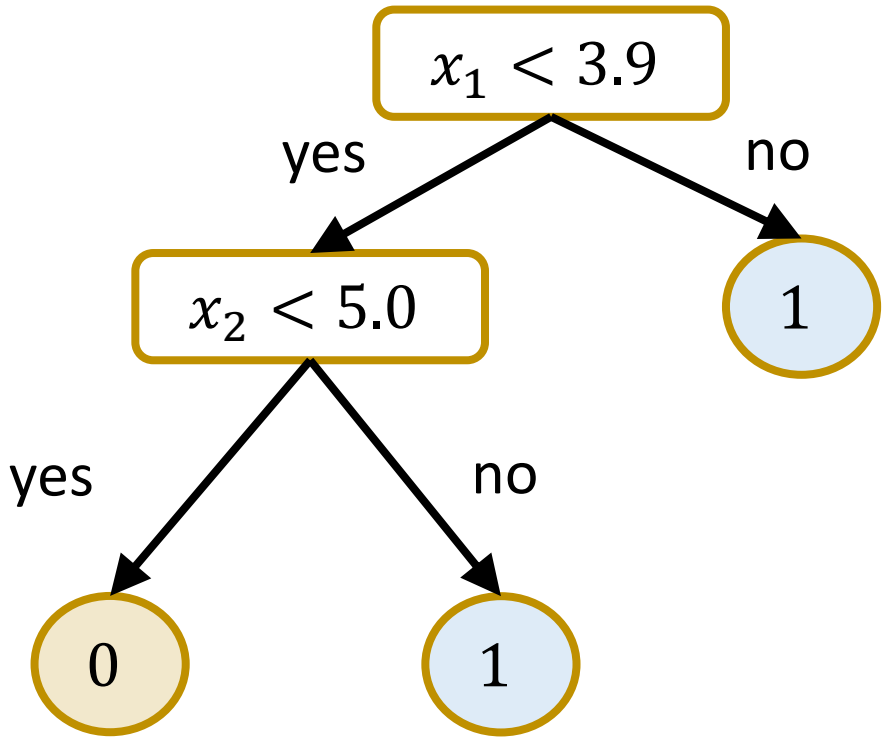
Avoiding Overfitting

Set a max depth of tree

- It will increase error

Random Decision Forests (RDFs)

- A random forest is simply a collection of decision trees whose results are aggregated into one final result.
- How to train a Random Decision Forest?
  - by training on different samples of the data
  - by using a random subset of features



Lancaster University

Random Decision Forest

Aim

- Train a classification model based on decision trees

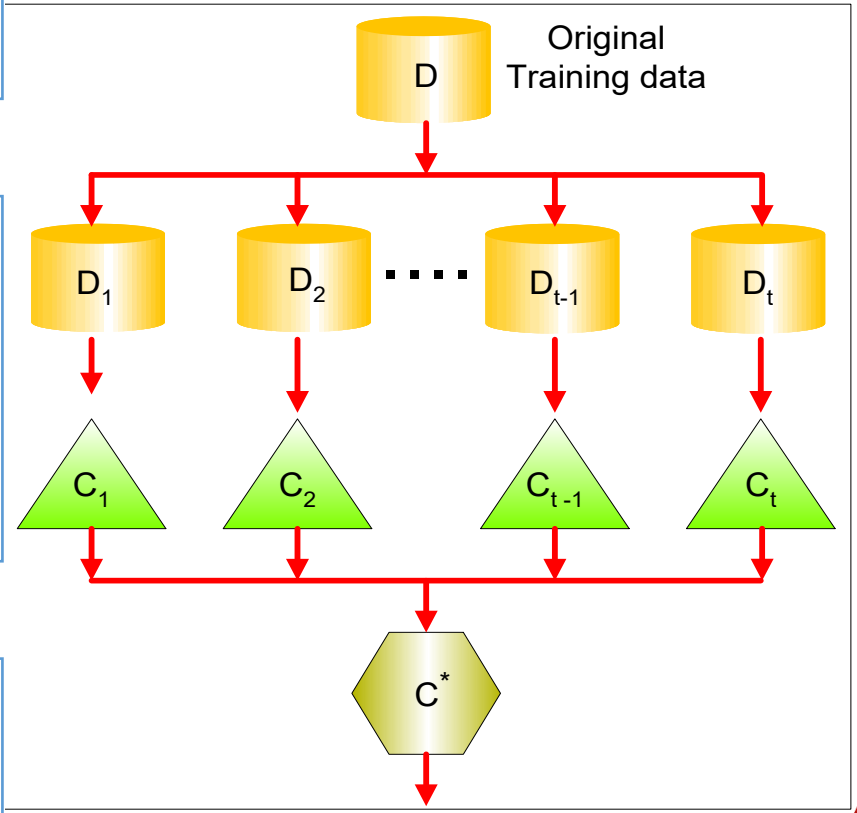
Training

- Let D denote a training set of samples
- Sample subsets D_i from D
- Train classifiers C_i from the subsets D_i

Testing

- Return a prediction from each classifier C_i
- Assign the modal classification

kNN



Lancaster University

Example

Should we wait at the restaurant or not?

Example	Input Data										Goal
	Alt	Bar	F/S	Hun	Pat	Price	Rain	Res	Type	Est	Will Wait?
x_1	Yes	No	No	Yes	Some	$$$	No	Yes	French	0-10	$y_1 = Yes$
x_2	Yes	No	No	Yes	Full	$	No	No	Thai	30-60	$y_2 = No$
x_3	No	Yes	No	No	Some	$	No	No	Burger	0-10	$y_3 = Yes$
x_4	Yes	No	Yes	Yes	Full	$	Yes	No	Thai	10-30	$y_4 = Yes$
x_5	Yes	No	Yes	No	Full	$$$	No	Yes	French	>60	$y_5 = No$
x_6	No	Yes	No	Yes	Some	$$	Yes	Yes	Italian	0-10	$y_6 = Yes$
x_7	No	Yes	No	No	None	$	Yes	No	Burger	0-10	$y_7 = No$
x_8	No	No	No	Yes	Some	$$	Yes	Yes	Thai	0-10	$y_8 = Yes$
x_9	No	Yes	Yes	No	Full	$	Yes	No	Burger	>60	$y_9 = No$
x_{10}	Yes	Yes	Yes	Yes	Full	$$$	No	Yes	Italian	10-30	$y_{10} = No$
x_{11}	No	No	No	No	None	$	No	No	Thai	0-10	$y_{11} = No$
x_{12}	Yes	Yes	Yes	Yes	Full	$	No	No	Burger	30-60	$y_{12} = Yes$

Overall Wait Decisions:

Yes: 6
No: 6

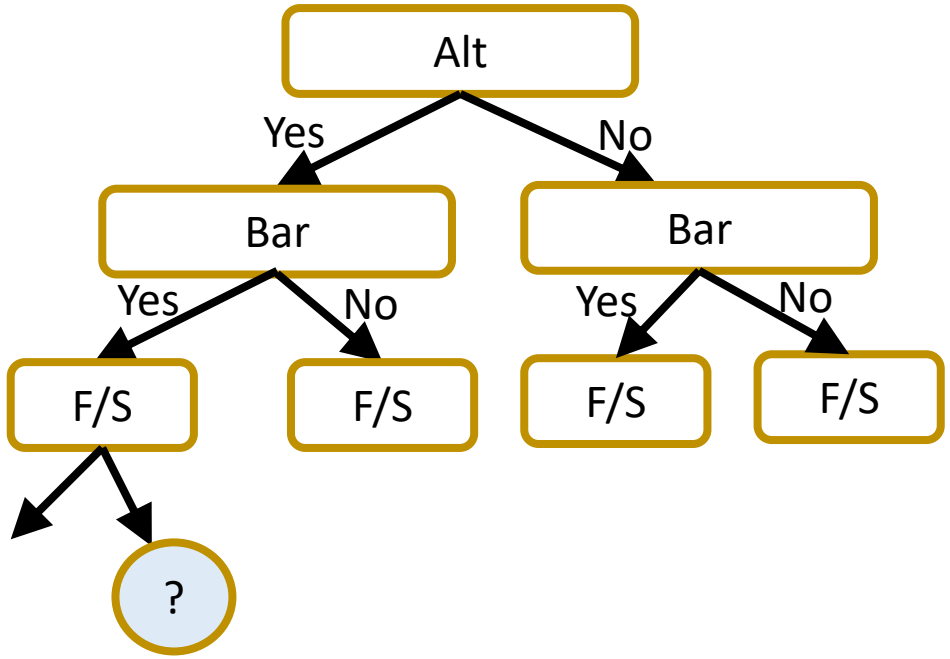
Key:

Alt: Alternative restaurant nearby
Bar: Bar area to wait
F/S: Yes on Fridays and Saturdays
Hun: whether hungry
Pat: how many people in restaurant
Price: price range
Rain: raining outside
Res: whether we made a reservation
Type: Cuisine
Est: Estimated wait

Example

Try a simple approach

Example	Input Data										Goal
	Alt	Bar	F/S	Hun	Pat	Price	Rain	Res	Type	Est	Will Wait?
x_1	Yes	No	No	Yes	Some	$$$	No	Yes	French	0-10	$y_1 = Yes$
x_2	Yes	No	No	Yes	Full	$	No	No	Thai	30-60	$y_2 = No$
x_3	No	Yes	No	No	Some	$	No	No	Burger	0-10	$y_3 = Yes$
x_4	Yes	No	Yes	Yes	Full	$	Yes	No	Thai	10-30	$y_4 = Yes$
x_5	Yes	No	Yes	No	Full	$$$	No	Yes	French	>60	$y_5 = No$
x_6	No	Yes	No	Yes	Some	$$	Yes	Yes	Italian	0-10	$y_6 = Yes$
x_7	No	Yes	No	No	None	$	Yes	No	Burger	0-10	$y_7 = No$
x_8	No	No	No	Yes	Some	$$	Yes	Yes	Thai	0-10	$y_8 = Yes$
x_9	No	Yes	Yes	No	Full	$	Yes	No	Burger	>60	$y_9 = No$
x_{10}	Yes	Yes	Yes	Yes	Full	$$$	No	Yes	Italian	10-30	$y_{10} = No$
x_{11}	No	No	No	No	None	$	No	No	Thai	0-10	$y_{11} = No$
x_{12}	Yes	Yes	Yes	Yes	Full	$	No	No	Burger	30-60	$y_{12} = Yes$



Key:

Alt: Alternative restaurant nearby
Bar: Bar area to wait
F/S: Yes on Fridays and Saturdays
Hun: whether hungry
Pat: how many people in restaurant
Price: price range
Rain: raining outside
Res: whether we made a reservation
Type: Cuisine
Est: Estimated wait

Example

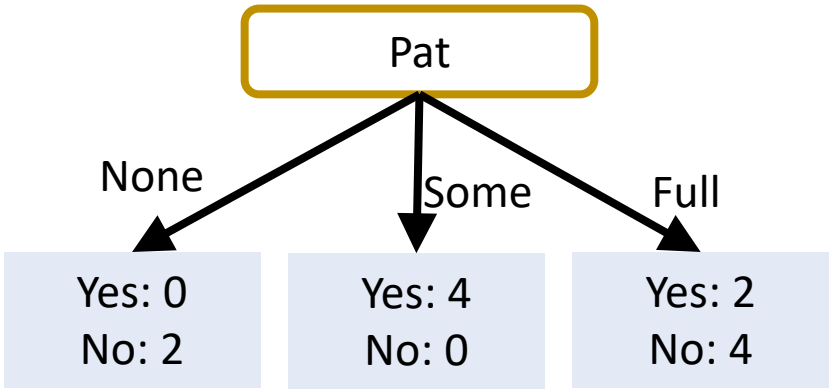
Start building Decision Tree – which feature to start with?

Example	Input Data										Goal
	Alt	Bar	F/S	Hun	Pat	Price	Rain	Res	Type	Est	Will Wait?
x_1	Yes	No	No	Yes	Some	$$$	No	Yes	French	0-10	$y_1 = Yes$
x_2	Yes	No	No	Yes	Full	$	No	No	Thai	30-60	$y_2 = No$
x_3	No	Yes	No	No	Some	$	No	No	Burger	0-10	$y_3 = Yes$
x_4	Yes	No	Yes	Yes	Full	$	Yes	No	Thai	10-30	$y_4 = Yes$
x_5	Yes	No	Yes	No	Full	$$$	No	Yes	French	>60	$y_5 = No$
x_6	No	Yes	No	Yes	Some	$$	Yes	Yes	Italian	0-10	$y_6 = Yes$
x_7	No	Yes	No	No	None	$	Yes	No	Burger	0-10	$y_7 = No$
x_8	No	No	No	Yes	Some	$$	Yes	Yes	Thai	0-10	$y_8 = Yes$
x_9	No	Yes	Yes	No	Full	$	Yes	No	Burger	>60	$y_9 = No$
x_{10}	Yes	Yes	Yes	Yes	Full	$$$	No	Yes	Italian	10-30	$y_{10} = No$
x_{11}	No	No	No	No	None	$	No	No	Thai	0-10	$y_{11} = No$
x_{12}	Yes	Yes	Yes	Yes	Full	$	No	No	Burger	30-60	$y_{12} = Yes$

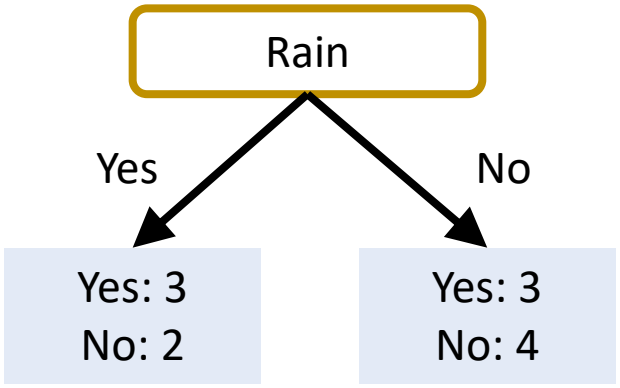
Key:

Alt: Alternative restaurant nearby
Bar: Bar area to wait
F/S: Yes on Fridays and Saturdays
Hun: whether hungry
Pat: how many people in restaurant
Price: price range
Rain: raining outside
Res: whether we made a reservation
Type: Cuisine
Est: Estimated wait

Start with Patrons?



Start with Rain?



Overall Wait Decisions:

Yes: 6
No: 6

Calculate Entropy:

Lancaster University

Example

Start building Decision Tree – which feature to start with?

Example	Input Data										Goal
	Alt	Bar	F/S	Hun	Pat	Price	Rain	Res	Type	Est	Will Wait?
x_1	Yes	No	No	Yes	Some	$$$	No	Yes	French	0-10	$y_1 = Yes$
x_2	Yes	No	No	Yes	Full	$	No	No	Thai	30-60	$y_2 = No$
x_3	No	Yes	No	No	Some	$	No	No	Burger	0-10	$y_3 = Yes$
x_4	Yes	No	Yes	Yes	Full	$	Yes	No	Thai	10-30	$y_4 = Yes$
x_5	Yes	No	Yes	No	Full	$$$	No	Yes	French	>60	$y_5 = No$
x_6	No	Yes	No	Yes	Some	$$	Yes	Yes	Italian	0-10	$y_6 = Yes$
x_7	No	Yes	No	No	None	$	Yes	No	Burger	0-10	$y_7 = No$
x_8	No	No	No	Yes	Some	$$	Yes	Yes	Thai	0-10	$y_8 = Yes$
x_9	No	Yes	Yes	No	Full	$	Yes	No	Burger	>60	$y_9 = No$
x_{10}	Yes	Yes	Yes	Yes	Full	$$$	No	Yes	Italian	10-30	$y_{10} = No$
x_{11}	No	No	No	No	None	$	No	No	Thai	0-10	$y_{11} = No$
x_{12}	Yes	Yes	Yes	Yes	Full	$	No	No	Burger	30-60	$y_{12} = Yes$

Key:

Alt: Alternative restaurant nearby
Bar: Bar area to wait
F/S: Yes on Fridays and Saturdays
Hun: whether hungry
Pat: how many people in restaurant
Price: price range
Rain: raining outside
Res: whether we made a reservation
Type: Cuisine
Est: Estimated wait

Overall Wait Decisions:

Yes: 6

No: 6

Calculate Entropy:

$$P_{Yes} = \frac{6}{12} = 0.5$$

$$P_{No} = \frac{6}{12} = 0.5$$

$$E(S) = -P_{Yes} \log_2 P_{Yes} - P_{No} \log_2 P_{No}$$

$$= -0.5 \log_2 0.5 - 0.5 \log_2 0.5$$

$$= 1$$

Example

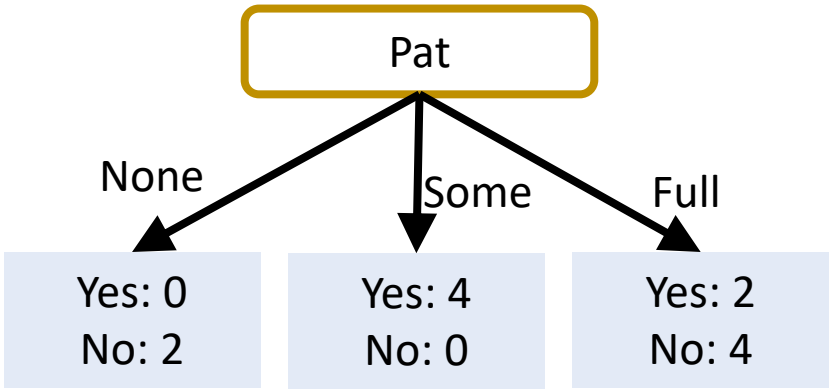
Start building Decision Tree – which feature to start with?

Example	Input Data										Goal
	Alt	Bar	F/S	Hun	Pat	Price	Rain	Res	Type	Est	Will Wait?
x_1	Yes	No	No	Yes	Some	$$$	No	Yes	French	0-10	$y_1 = Yes$
x_2	Yes	No	No	Yes	Full	$	No	No	Thai	30-60	$y_2 = No$
x_3	No	Yes	No	No	Some	$	No	No	Burger	0-10	$y_3 = Yes$
x_4	Yes	No	Yes	Yes	Full	$	Yes	No	Thai	10-30	$y_4 = Yes$
x_5	Yes	No	Yes	No	Full	$$$	No	Yes	French	>60	$y_5 = No$
x_6	No	Yes	No	Yes	Some	$$	Yes	Yes	Italian	0-10	$y_6 = Yes$
x_7	No	Yes	No	No	None	$	Yes	No	Burger	0-10	$y_7 = No$
x_8	No	No	No	Yes	Some	$$	Yes	Yes	Thai	0-10	$y_8 = Yes$
x_9	No	Yes	Yes	No	Full	$	Yes	No	Burger	>60	$y_9 = No$
x_{10}	Yes	Yes	Yes	Yes	Full	$$$	No	Yes	Italian	10-30	$y_{10} = No$
x_{11}	No	No	No	No	None	$	No	No	Thai	0-10	$y_{11} = No$
x_{12}	Yes	Yes	Yes	Yes	Full	$	No	No	Burger	30-60	$y_{12} = Yes$

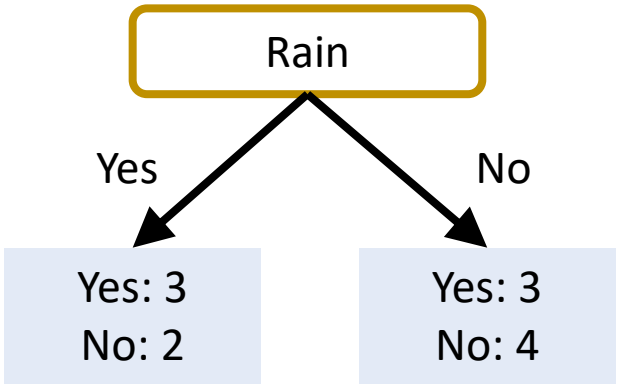
Key:

Alt: Alternative restaurant nearby
Bar: Bar area to wait
F/S: Yes on Fridays and Saturdays
Hun: whether hungry
Pat: how many people in restaurant
Price: price range
Rain: raining outside
Res: whether we made a reservation
Type: Cuisine
Est: Estimated wait

Start with Patrons?



Start with Rain?



Overall Wait Decisions:

Yes: 6
No: 6

$$E(S) = 1$$

Patrons (None):

$$P_{Yes} = \frac{0}{2} = 0$$

$$P_{No} = \frac{2}{2} = 1$$

Example

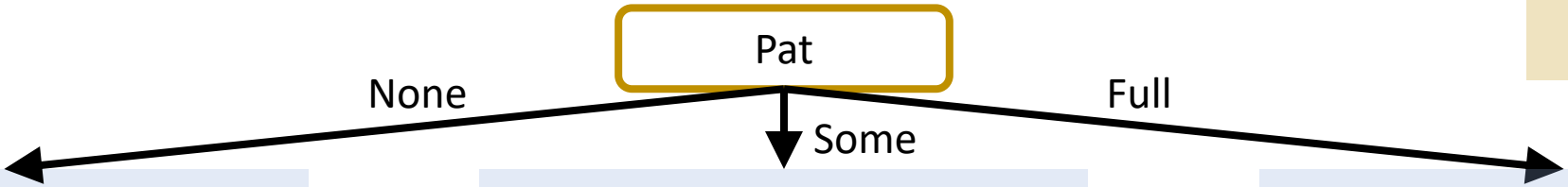
Start building Decision Tree – which feature to start with?

Overall Wait Decisions:

Yes: 6
No: 6

$$E(S) = 1$$

Start with Patrons?



None	Some	Full
Yes: 0 No: 2	Yes: 4 No: 0	Yes: 2 No: 4
$P_{Yes} = \frac{0}{2} = 0$ $P_{No} = \frac{2}{2} = 1$	$P_{Yes} = \frac{4}{4} = 1$ $P_{No} = \frac{0}{4} = 0$	$P_{Yes} = \frac{2}{6} = \frac{1}{3}$ $P_{No} = \frac{4}{6} = \frac{2}{3}$
$E(Pat = None)$ $= -P_{Yes}\log_2 P_{Yes} - P_{No}\log_2 P_{No}$ $= 0\log_2 0 - \log_2 1 = 0$	$E(Pat = Some)$ $= -P_{Yes}\log_2 P_{Yes} - P_{No}\log_2 P_{No}$ $= -\log_2 1 - 0\log_2 0 = 0$	$E(Pat = Full)$ $= -P_{Yes}\log_2 P_{Yes} - P_{No}\log_2 P_{No}$ $= -\frac{1}{3}\log_2\frac{1}{3} - \frac{2}{3}\log_2\frac{2}{3} \approx 0.918$

Lancaster University

Example

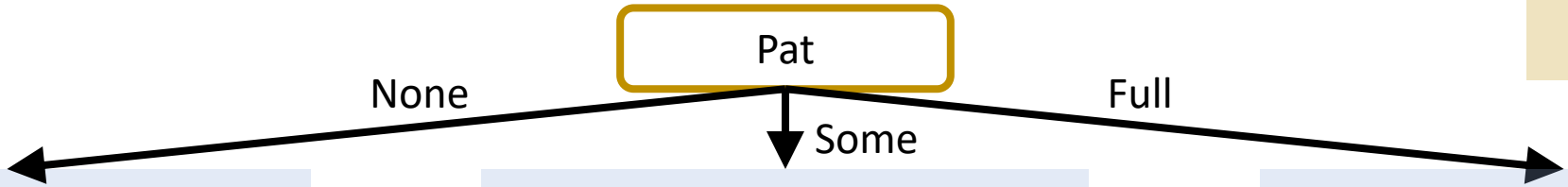
Start building Decision Tree – which feature to start with?

Overall Wait Decisions:

Yes: 6
No: 6

$$E(S) = 1$$

Start with Patrons?



None	Some	Full
Yes: 0 No: 2	Yes: 4 No: 0	Yes: 2 No: 4
$P_{Yes} = \frac{0}{2} = 0$ $P_{No} = \frac{2}{2} = 1$	$P_{Yes} = \frac{4}{4} = 1$ $P_{No} = \frac{0}{4} = 0$	$P_{Yes} = \frac{2}{6} = \frac{1}{3}$ $P_{No} = \frac{4}{6} = \frac{2}{3}$
$E(Pat = None) = 0$	$E(Pat = Some) = 0$	$E(Pat = Full) \approx 0.918$

Average Entropy: $I(S, Patrons) = \sum_i \frac{|S_i|}{|S|} E(S_i) = \frac{2}{12}0 + \frac{4}{12}0 + \frac{6}{12} \cdot 0.918 = 0.459$

Information Gain: $\text{Gain}(S, Patrons) = E(S) - I(S, Patrons) = 1 - 0.459 = 0.541$

Lancaster University

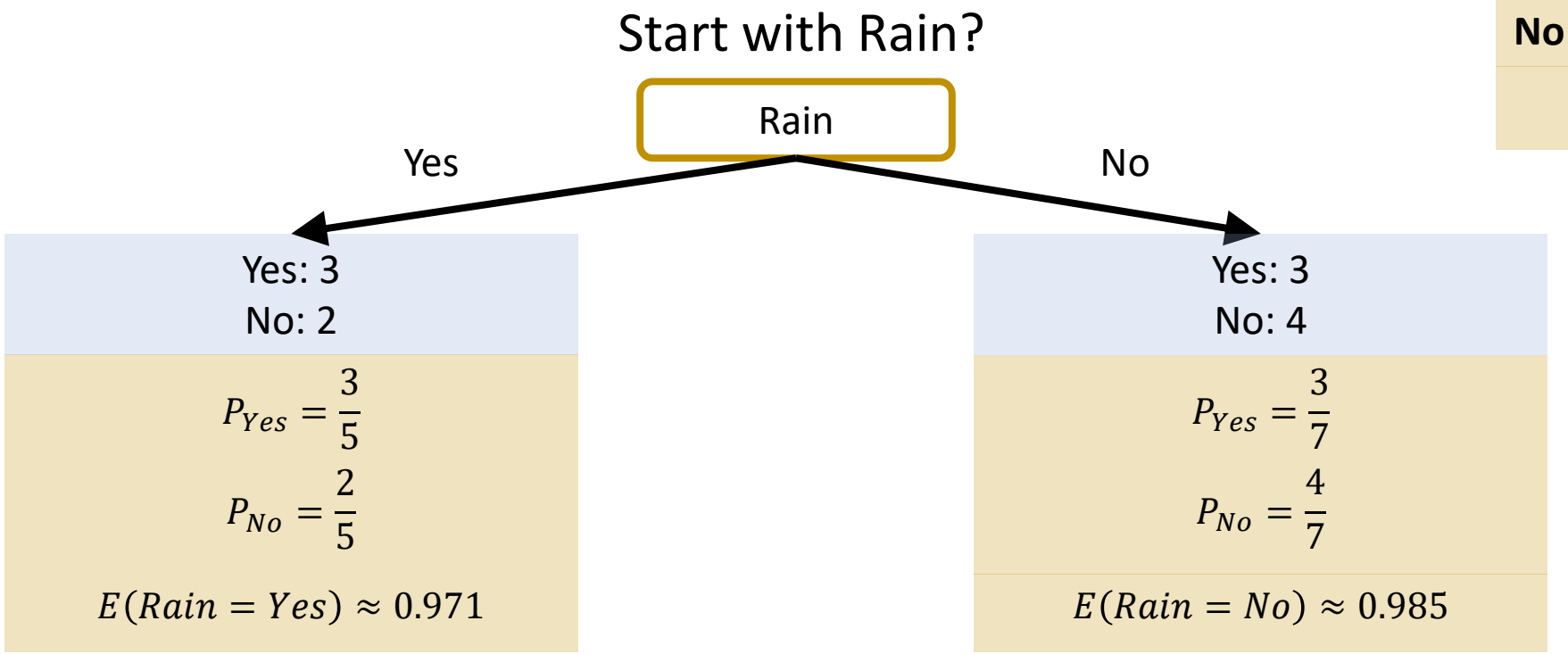
Example

Start building Decision Tree – which feature to start with?

Overall Wait Decisions:

Yes: 6
No: 6

$$E(S) = 1$$



Average Entropy: $$I(S, Rain) = \sum_i \frac{|S_i|}{|S|} E(S_i) = \frac{5}{12} \cdot 0.971 + \frac{7}{12} \cdot 0.985 = 0.979$$

Information Gain: $$\text{Gain}(S, Rain) = E(S) - I(S, Patrons) = 1 - 0.979 = 0.021$$

Example

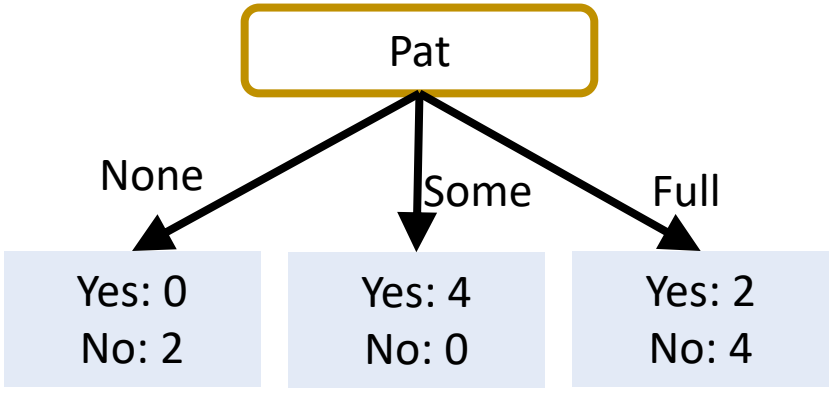
Start building Decision Tree – which feature to start with?

Example	Input Data										Goal
	Alt	Bar	F/S	Hun	Pat	Price	Rain	Res	Type	Est	Will Wait?
x_1	Yes	No	No	Yes	Some	$$$	No	Yes	French	0-10	$y_1 = Yes$
x_2	Yes	No	No	Yes	Full	$	No	No	Thai	30-60	$y_2 = No$
x_3	No	Yes	No	No	Some	$	No	No	Burger	0-10	$y_3 = Yes$
x_4	Yes	No	Yes	Yes	Full	$	Yes	No	Thai	10-30	$y_4 = Yes$
x_5	Yes	No	Yes	No	Full	$$$	No	Yes	French	>60	$y_5 = No$
x_6	No	Yes	No	Yes	Some	$$	Yes	Yes	Italian	0-10	$y_6 = Yes$
x_7	No	Yes	No	No	None	$	Yes	No	Burger	0-10	$y_7 = No$
x_8	No	No	No	Yes	Some	$$	Yes	Yes	Thai	0-10	$y_8 = Yes$
x_9	No	Yes	Yes	No	Full	$	Yes	No	Burger	>60	$y_9 = No$
x_{10}	Yes	Yes	Yes	Yes	Full	$$$	No	Yes	Italian	10-30	$y_{10} = No$
x_{11}	No	No	No	No	None	$	No	No	Thai	0-10	$y_{11} = No$
x_{12}	Yes	Yes	Yes	Yes	Full	$	No	No	Burger	30-60	$y_{12} = Yes$

Key:

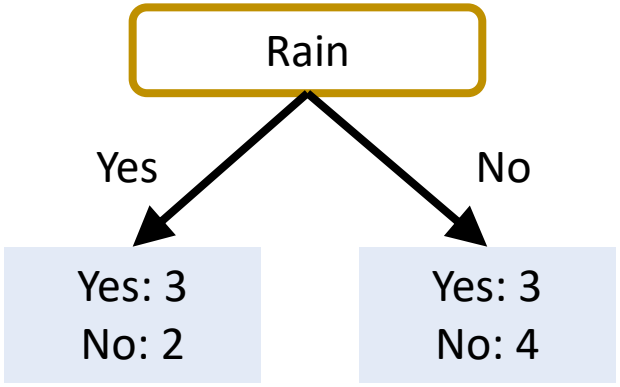
Alt: Alternative restaurant nearby
Bar: Bar area to wait
F/S: Yes on Fridays and Saturdays
Hun: whether hungry
Pat: how many people in restaurant
Price: price range
Rain: raining outside
Res: whether we made a reservation
Type: Cuisine
Est: Estimated wait

Start with Patrons?



$$Gain(S, Patrons) = 0.541$$

Start with Rain?



$$Gain(S, Rain) = 0.021$$

Example

Start building Decision Tree – which feature to start with?

Example	Input Data										Goal
	Alt	Bar	F/S	Hun	Pat	Price	Rain	Res	Type	Est	Will Wait?
x_1	Yes	No	No	Yes	Some	$$$	No	Yes	French	0-10	$y_1 = Yes$
x_2	Yes	No	No	Yes	Full	$	No	No	Thai	30-60	$y_2 = No$
x_3	No	Yes	No	No	Some	$	No	No	Burger	0-10	$y_3 = Yes$
x_4	Yes	No	Yes	Yes	Full	$	Yes	No	Thai	10-30	$y_4 = Yes$
x_5	Yes	No	Yes	No	Full	$$$	No	Yes	French	>60	$y_5 = No$
x_6	No	Yes	No	Yes	Some	$$	Yes	Yes	Italian	0-10	$y_6 = Yes$
x_7	No	Yes	No	No	None	$	Yes	No	Burger	0-10	$y_7 = No$
x_8	No	No	No	Yes	Some	$$	Yes	Yes	Thai	0-10	$y_8 = Yes$
x_9	No	Yes	Yes	No	Full	$	Yes	No	Burger	>60	$y_9 = No$
x_{10}	Yes	Yes	Yes	Yes	Full	$$$	No	Yes	Italian	10-30	$y_{10} = No$
x_{11}	No	No	No	No	None	$	No	No	Thai	0-10	$y_{11} = No$
x_{12}	Yes	Yes	Yes	Yes	Full	$	No	No	Burger	30-60	$y_{12} = Yes$

Key:

Alt: Alternative restaurant nearby
Bar: Bar area to wait
F/S: Yes on Fridays and Saturdays
Hun: whether hungry
Pat: how many people in restaurant
Price: price range
Rain: raining outside
Res: whether we made a reservation
Type: Cuisine
Est: Estimated wait

Feature	Gain
Alt	0.000
Bar	0.000
F/S	0.021
Hun	0.196
Pat	0.541
Price	0.196
Rain	0.021
Res	0.021
Type	0.000
Est	0.208

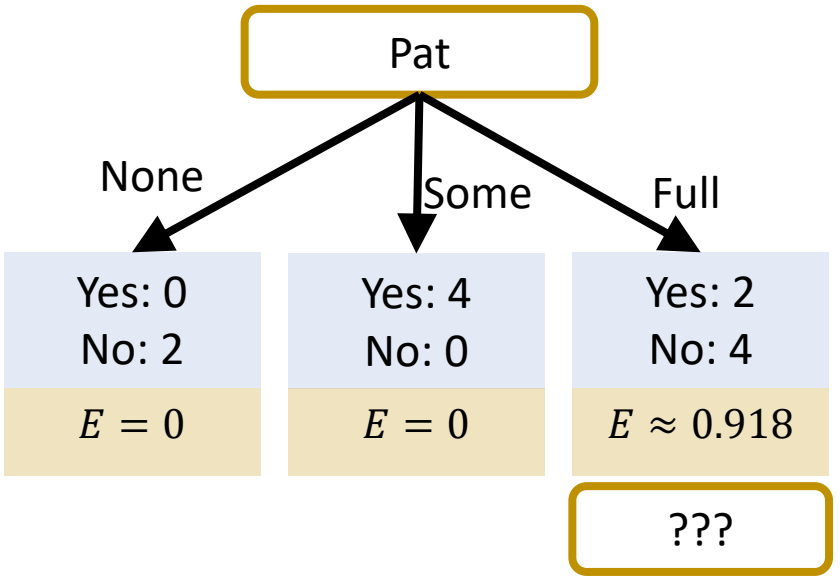
Patrons has highest gain

=> start with patrons

Example

Start building Decision Tree – which feature to start with?

Example	Input Data										Goal
	Alt	Bar	F/S	Hun	Pat	Price	Rain	Res	Type	Est	Will Wait?
x_1	Yes	No	No	Yes	Some	$$$	No	Yes	French	0-10	$y_1 = Yes$
x_2	Yes	No	No	Yes	Full	$	No	No	Thai	30-60	$y_2 = No$
x_3	No	Yes	No	No	Some	$	No	No	Burger	0-10	$y_3 = Yes$
x_4	Yes	No	Yes	Yes	Full	$	Yes	No	Thai	10-30	$y_4 = Yes$
x_5	Yes	No	Yes	No	Full	$$$	No	Yes	French	>60	$y_5 = No$
x_6	No	Yes	No	Yes	Some	$$	Yes	Yes	Italian	0-10	$y_6 = Yes$
x_7	No	Yes	No	No	None	$	Yes	No	Burger	0-10	$y_7 = No$
x_8	No	No	No	Yes	Some	$$	Yes	Yes	Thai	0-10	$y_8 = Yes$
x_9	No	Yes	Yes	No	Full	$	Yes	No	Burger	>60	$y_9 = No$
x_{10}	Yes	Yes	Yes	Yes	Full	$$$	No	Yes	Italian	10-30	$y_{10} = No$
x_{11}	No	No	No	No	None	$	No	No	Thai	0-10	$y_{11} = No$
x_{12}	Yes	Yes	Yes	Yes	Full	$	No	No	Burger	30-60	$y_{12} = Yes$



Key:

Alt: Alternative restaurant nearby
Bar: Bar area to wait
F/S: Yes on Fridays and Saturdays
Hun: whether hungry
Pat: how many people in restaurant
Price: price range
Rain: raining outside
Res: whether we made a reservation
Type: Cuisine
Est: Estimated wait

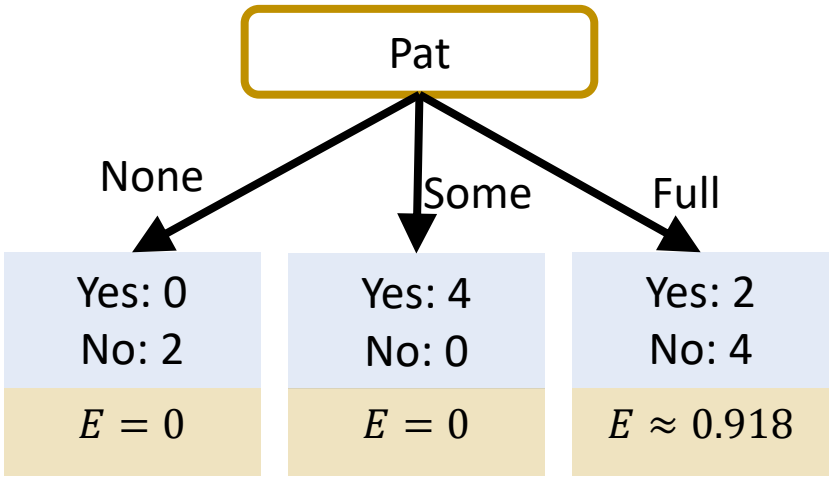
Example

Start building Decision Tree – which feature to start with?

Example	Input Data										Goal
	Alt	Bar	F/S	Hun	Pat	Price	Rain	Res	Type	Est	Will Wait?
x_2	Yes	No	No	Yes	Full	$	No	No	Thai	30-60	$y_2 = No$
x_4	Yes	No	Yes	Yes	Full	$	Yes	No	Thai	10-30	$y_4 = Yes$
x_5	Yes	No	Yes	No	Full	$$$	No	Yes	French	>60	$y_5 = No$
x_9	No	Yes	Yes	No	Full	$	Yes	No	Burger	>60	$y_9 = No$
x_{10}	Yes	Yes	Yes	Yes	Full	$$$	No	Yes	Italian	10-30	$y_{10} = No$
x_{12}	Yes	Yes	Yes	Yes	Full	$	No	No	Burger	30-60	$y_{12} = Yes$

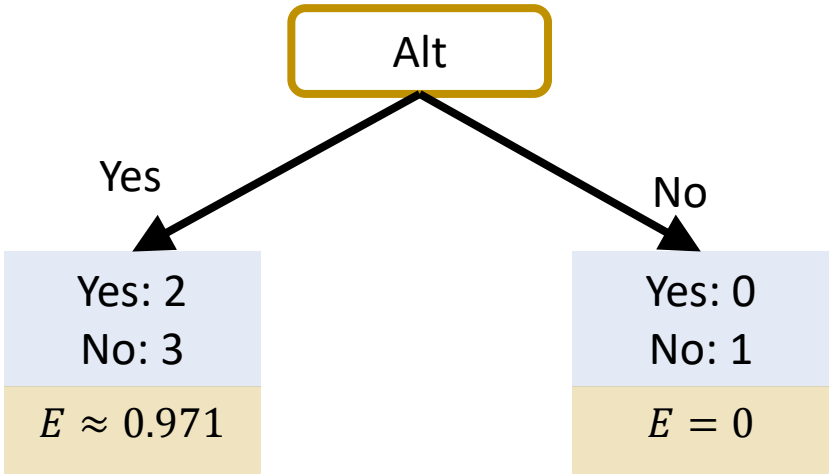
Key:

Alt: Alternative restaurant nearby
Bar: Bar area to wait
F/S: Yes on Fridays and Saturdays
Hun: whether hungry
Pat: how many people in restaurant
Price: price range
Rain: raining outside
Res: whether we made a reservation
Type: Cuisine
Est: Estimated wait



$$I(S, Alt) \approx 0.809$$

$$Gain(S, Alt) \approx 0.918 - 0.809 = 0.109$$



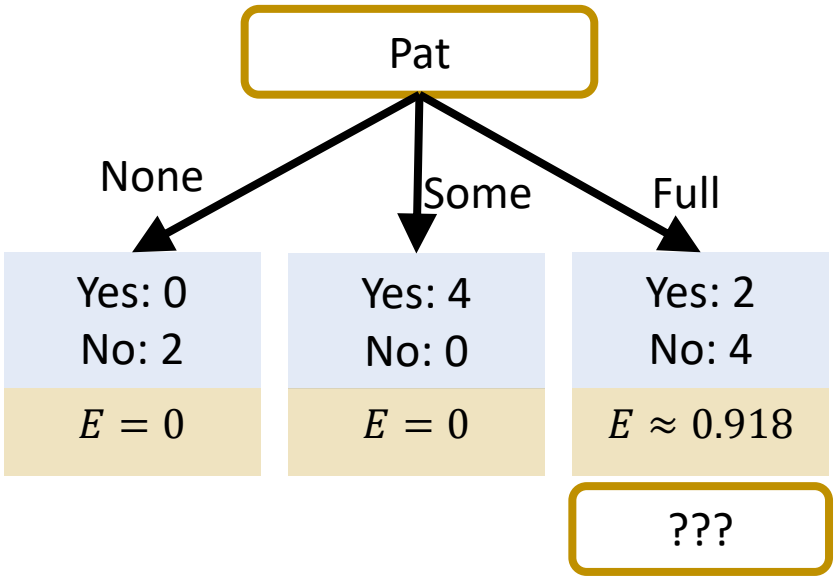
Example

Start building Decision Tree – which feature to start with?

Example	Input Data										Goal
	Alt	Bar	F/S	Hun	Pat	Price	Rain	Res	Type	Est	Will Wait?
x_2	Yes	No	No	Yes	Full	$	No	No	Thai	30-60	$y_2 = No$
x_4	Yes	No	Yes	Yes	Full	$	Yes	No	Thai	10-30	$y_4 = Yes$
x_5	Yes	No	Yes	No	Full	$$$	No	Yes	French	>60	$y_5 = No$
x_9	No	Yes	Yes	No	Full	$	Yes	No	Burger	>60	$y_9 = No$
x_{10}	Yes	Yes	Yes	Yes	Full	$$$	No	Yes	Italian	10-30	$y_{10} = No$
x_{12}	Yes	Yes	Yes	Yes	Full	$	No	No	Burger	30-60	$y_{12} = Yes$

Key:

Alt: Alternative restaurant nearby
Bar: Bar area to wait
F/S: Yes on Fridays and Saturdays
Hun: whether hungry
Pat: how many people in restaurant
Price: price range
Rain: raining outside
Res: whether we made a reservation
Type: Cuisine
Est: Estimated wait



Feature	Gain
Alt	0.109
Bar	0.000
F/S	0.109
Hun	0.251
Pat	0.000
Price	0.251
Rain	0.044
Res	0.251
Type	0.251
Est	0.251

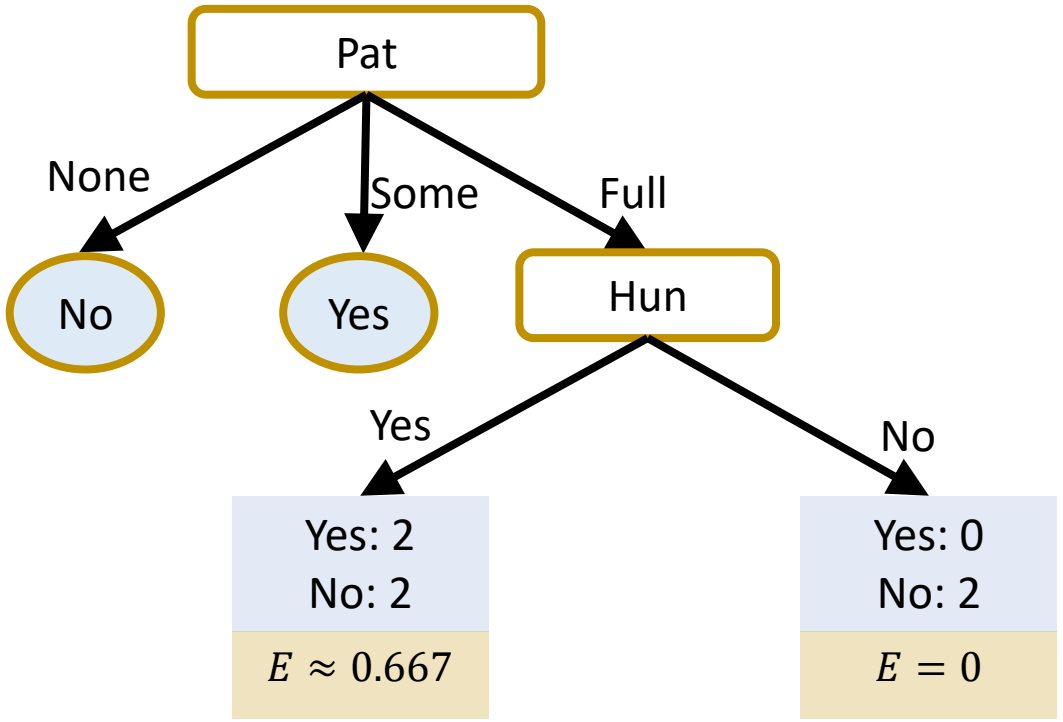
Example

Start building Decision Tree – which feature to start with?

Example	Input Data										Goal
	Alt	Bar	F/S	Hun	Pat	Price	Rain	Res	Type	Est	Will Wait?
x_2	Yes	No	No	Yes	Full	$	No	No	Thai	30-60	$y_2 = No$
x_4	Yes	No	Yes	Yes	Full	$	Yes	No	Thai	10-30	$y_4 = Yes$
x_5	Yes	No	Yes	No	Full	$$$	No	Yes	French	>60	$y_5 = No$
x_9	No	Yes	Yes	No	Full	$	Yes	No	Burger	>60	$y_9 = No$
x_{10}	Yes	Yes	Yes	Yes	Full	$$$	No	Yes	Italian	10-30	$y_{10} = No$
x_{12}	Yes	Yes	Yes	Yes	Full	$	No	No	Burger	30-60	$y_{12} = Yes$

Key:

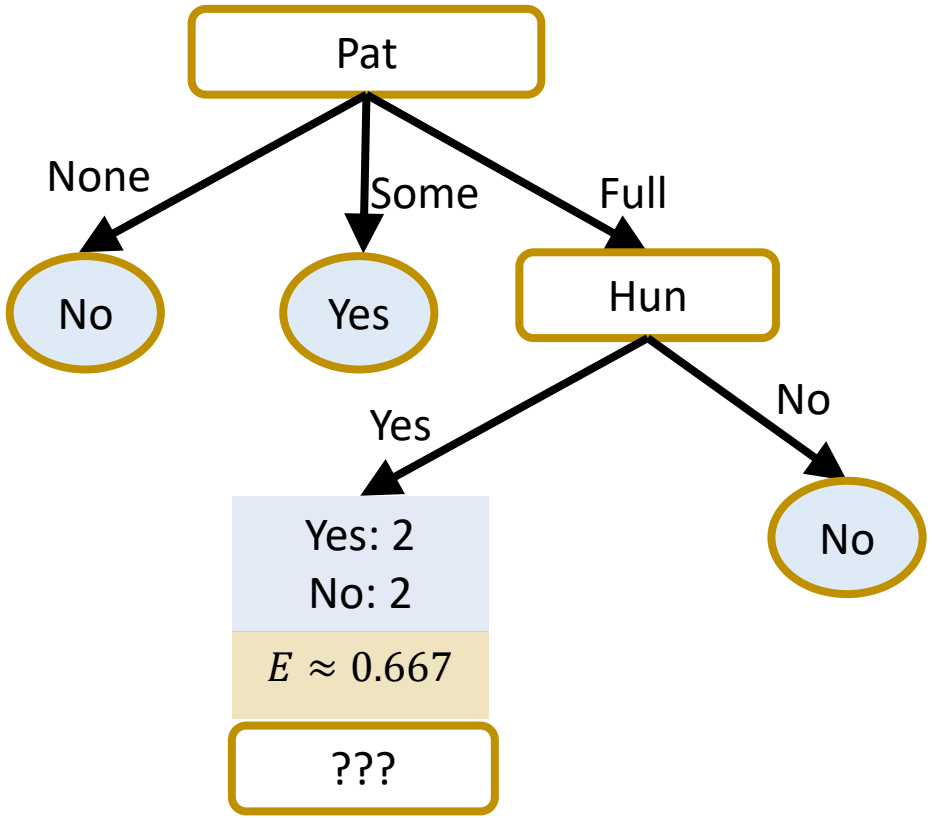
Alt: Alternative restaurant nearby
Bar: Bar area to wait
F/S: Yes on Fridays and Saturdays
Hun: whether hungry
Pat: how many people in restaurant
Price: price range
Rain: raining outside
Res: whether we made a reservation
Type: Cuisine
Est: Estimated wait



Example

Start building Decision Tree – which feature to start with?

Example	Input Data										Goal
	Alt	Bar	F/S	Hun	Pat	Price	Rain	Res	Type	Est	Will Wait?
[redacted]											
x_2	Yes	No	No	Yes	Full	$	No	No	Thai	30-60	$y_2 = No$
[redacted]											
x_4	Yes	No	Yes	Yes	Full	$	Yes	No	Thai	10-30	$y_4 = Yes$
[redacted]											
x_{10}	Yes	Yes	Yes	Yes	Full	$$$	No	Yes	Italian	10-30	$y_{10} = No$
[redacted]											
x_{12}	Yes	Yes	Yes	Yes	Full	$	No	No	Burger	30-60	$y_{12} = Yes$



Key:

Alt: Alternative restaurant nearby
Bar: Bar area to wait
F/S: Yes on Fridays and Saturdays
Hun: whether hungry
Pat: how many people in restaurant
Price: price range
Rain: raining outside
Res: whether we made a reservation
Type: Cuisine
Est: Estimated wait

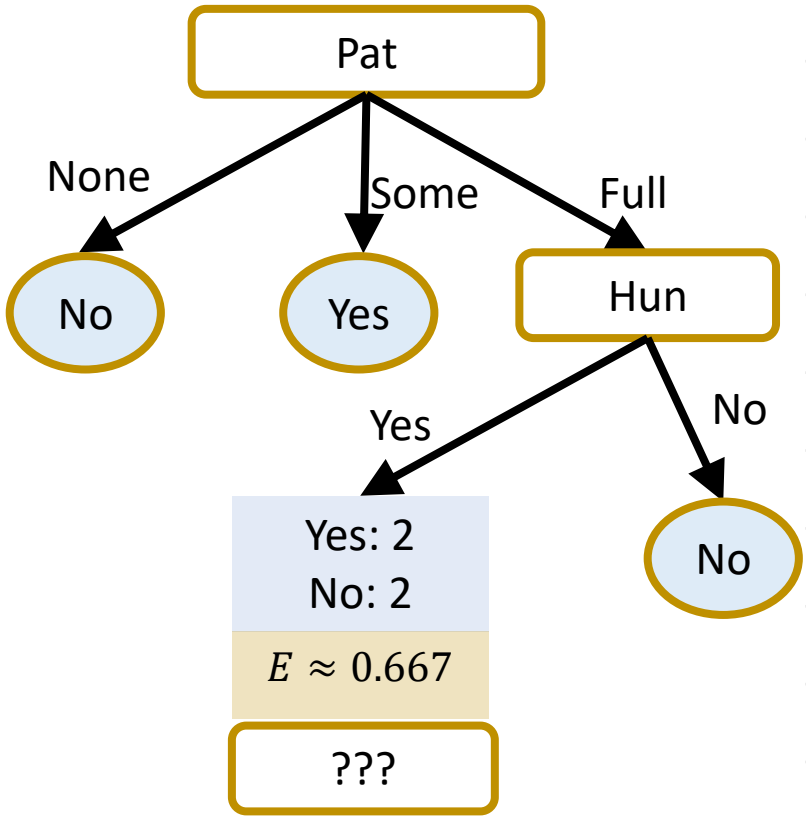
Example

Start building Decision Tree – which feature to start with?

Example	Input Data										Goal
	Alt	Bar	F/S	Hun	Pat	Price	Rain	Res	Type	Est	Will Wait?
x_2	Yes	No	No	Yes	Full	$	No	No	Thai	30-60	$y_2 = No$
x_4	Yes	No	Yes	Yes	Full	$	Yes	No	Thai	10-30	$y_4 = Yes$
x_{10}	Yes	Yes	Yes	Yes	Full	$$$	No	Yes	Italian	10-30	$y_{10} = No$
x_{12}	Yes	Yes	Yes	Yes	Full	$	No	No	Burger	30-60	$y_{12} = Yes$

Key:

Alt: Alternative restaurant nearby
Bar: Bar area to wait
F/S: Yes on Fridays and Saturdays
Hun: whether hungry
Pat: how many people in restaurant
Price: price range
Rain: raining outside
Res: whether we made a reservation
Type: Cuisine
Est: Estimated wait



Feature	Gain
Alt	-0.333
Bar	-0.333
F/S	0.022
Hun	0.251
Pat	-0.333
Price	-0.022
Rain	-0.022
Res	-0.022
Type	0.167
Est	0.333

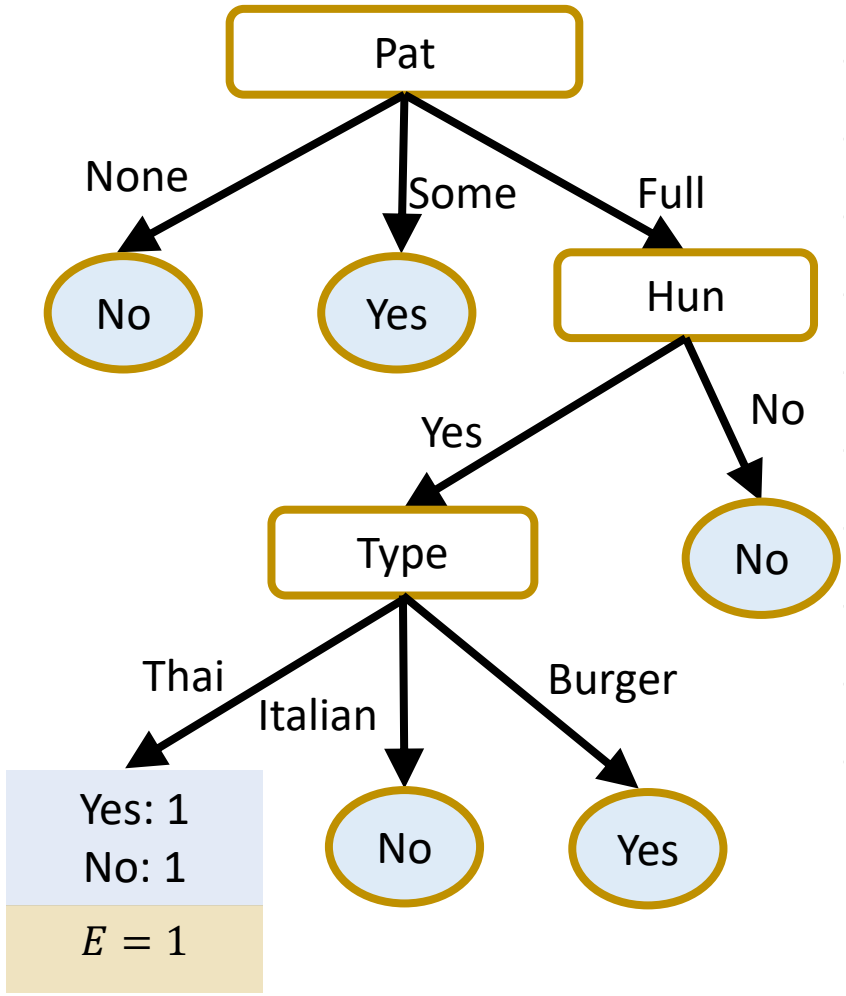
Example

Start building Decision Tree – which feature to start with?

Example	Input Data										Goal
	Alt	Bar	F/S	Hun	Pat	Price	Rain	Res	Type	Est	Will Wait?
x_2	Yes	No	No	Yes	Full	$	No	No	Thai	30-60	$y_2 = No$
x_4	Yes	No	Yes	Yes	Full	$	Yes	No	Thai	10-30	$y_4 = Yes$

Key:

Alt: Alternative restaurant nearby
Bar: Bar area to wait
F/S: Yes on Fridays and Saturdays
Hun: whether hungry
Pat: how many people in restaurant
Price: price range
Rain: raining outside
Res: whether we made a reservation
Type: Cuisine
Est: Estimated wait



Feature	Gain
Alt	0.000
Bar	0.000
F/S	1.000
Hun	0.000
Pat	0.000
Price	0.000
Rain	1.000
Res	0.000
Type	0.000
Est	1.000

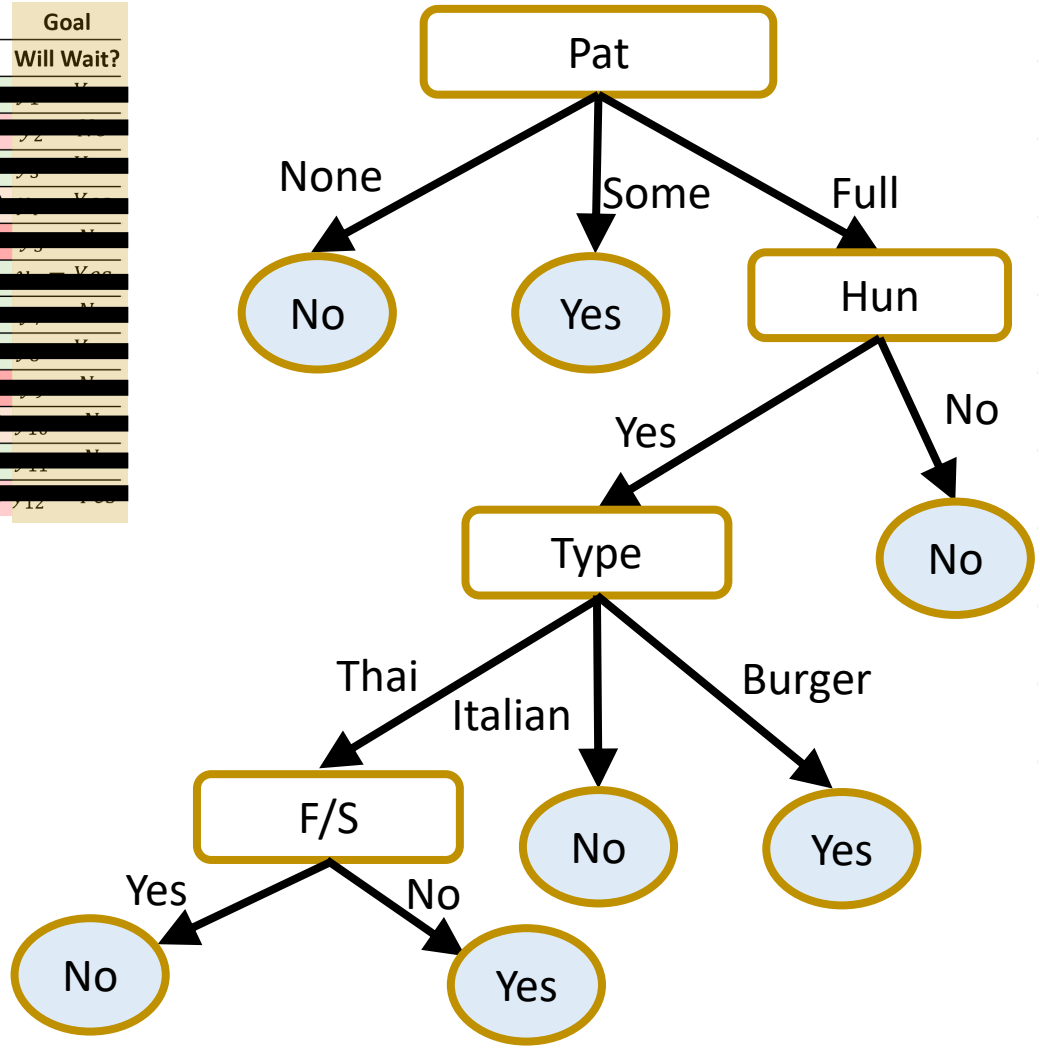
Example

Start building Decision Tree – which feature to start with?

Example	Input Data										Goal
	Alt	Bar	F/S	Hun	Pat	Price	Rain	Res	Type	Est	Will Wait?
1	[redacted]										
2											
3											
4											
5											
6											
7											
8											
9											
10											
11											
12											

Key:

Alt: Alternative restaurant nearby
Bar: Bar area to wait
F/S: Yes on Fridays and Saturdays
Hun: whether hungry
Pat: how many people in restaurant
Price: price range
Rain: raining outside
Res: whether we made a reservation
Type: Cuisine
Est: Estimated wait



Feature	Gain
Alt	0.000
Bar	0.000
F/S	1.000
Hun	0.000
Pat	0.000
Price	0.000
Rain	1.000
Res	0.000
Type	0.000
Est	1.000