Lancaster University

# SCC361: Artificial Intelligence

## Week 5: Classification

Random Forest and Naïve Bayes

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**

# Classification
# Decision Trees

# Decision Trees

Linearly Separable

Feature Space Division

Definitions and Structure

Multiple Solutions

# Entropy

Optimal Decision Trees

Definition

Binary and Multiclass

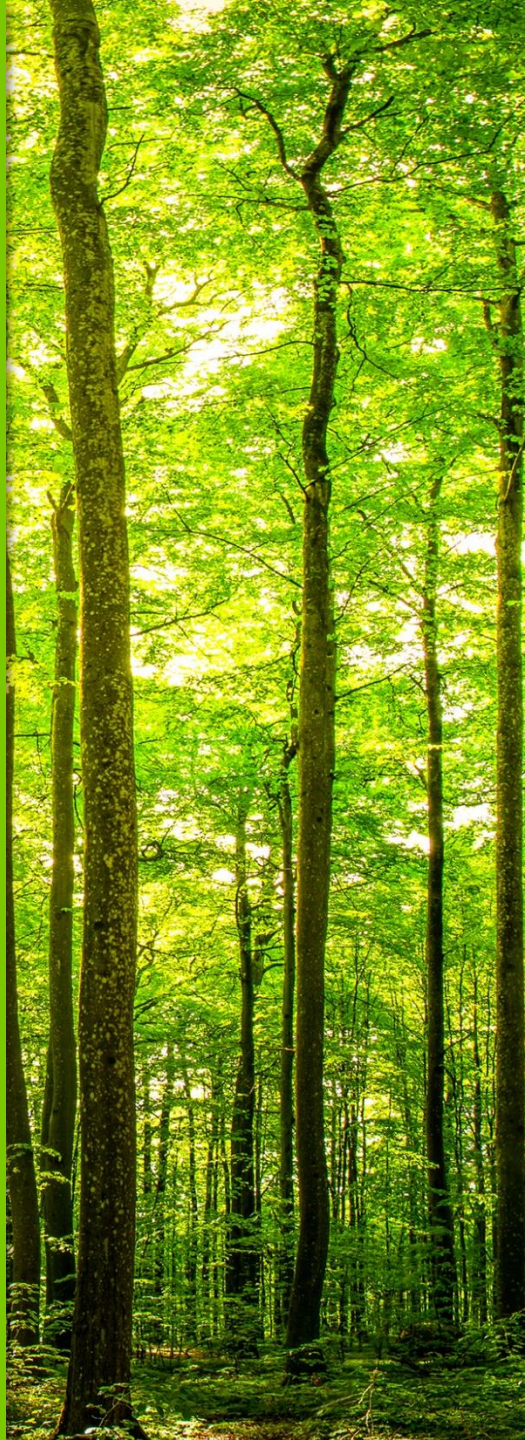
How to Calculate

# Information Gain

Definition

Relevance

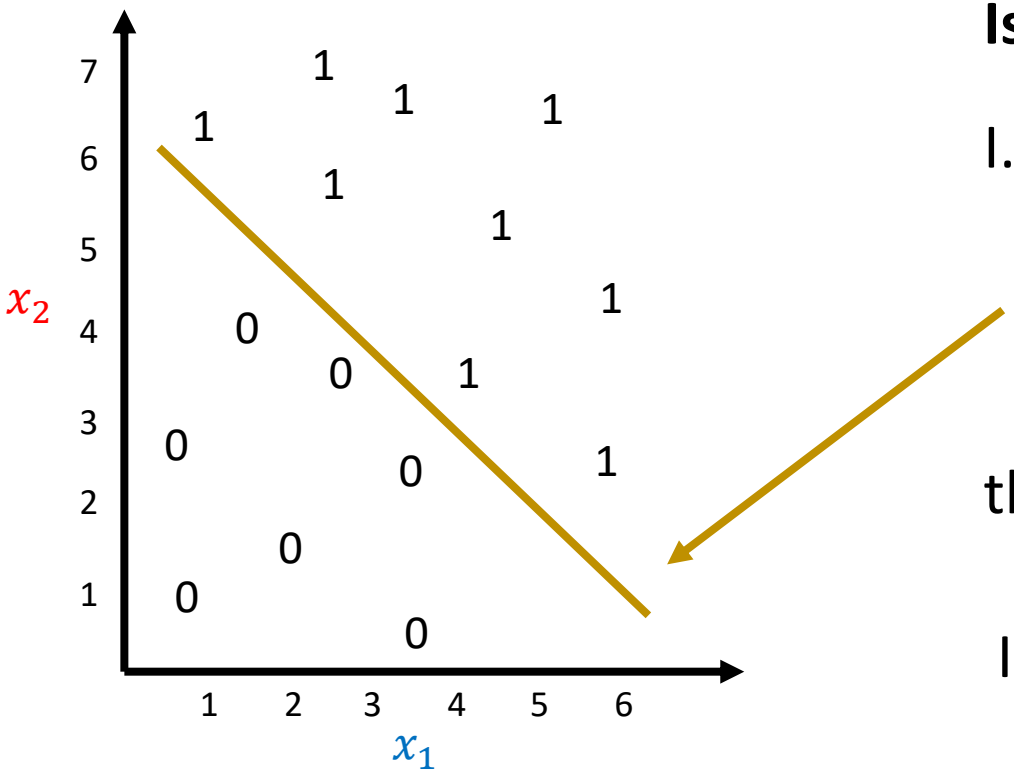
Calculation



# 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

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

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

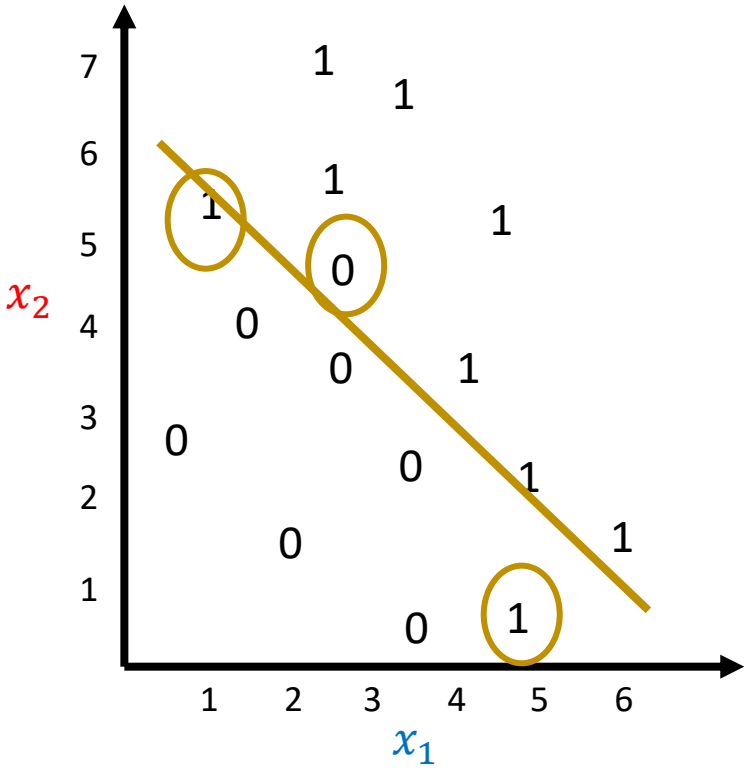
that separates the classes?

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.

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



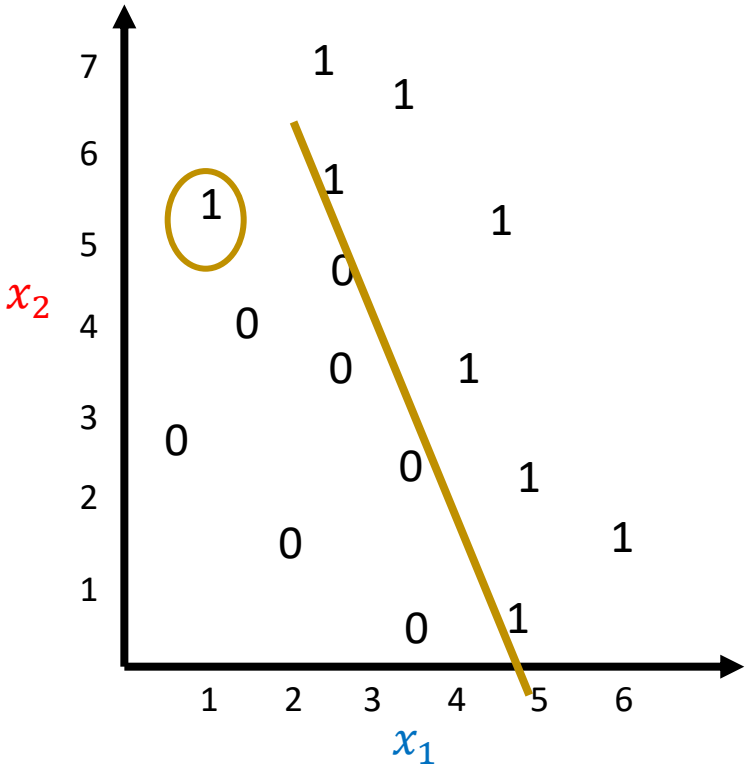

**Not linear separable**

Resub loss: $\frac{3}{16} \approx 0.1875$

# 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.

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




**Better but still not linear separable**

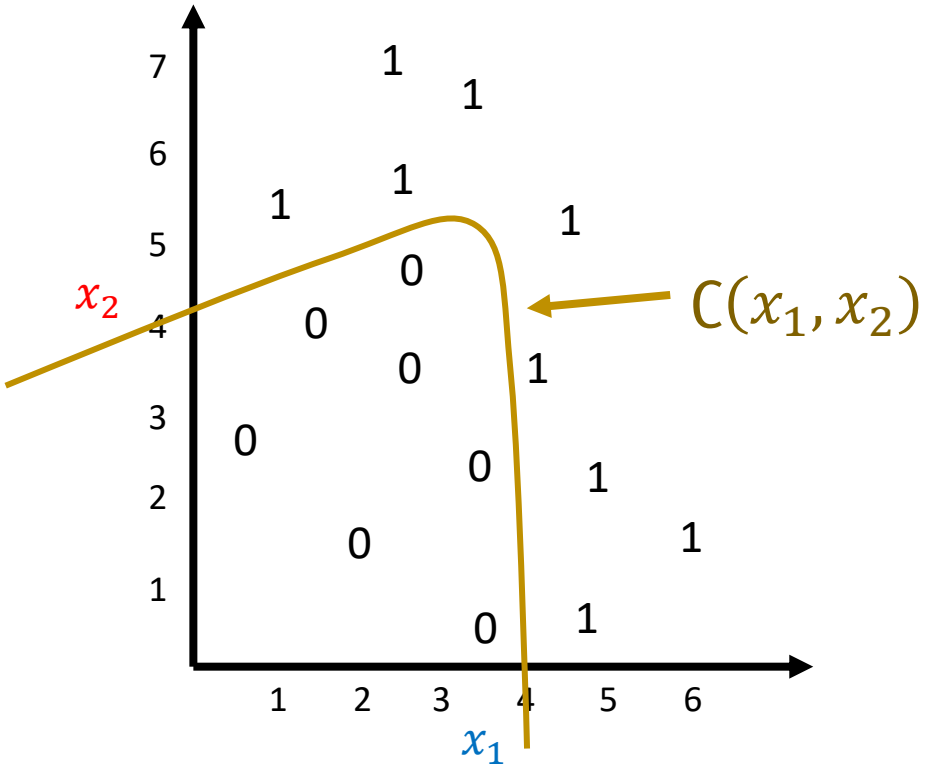
Resub loss: $\frac{1}{16} \approx 0.0625$

# 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.

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

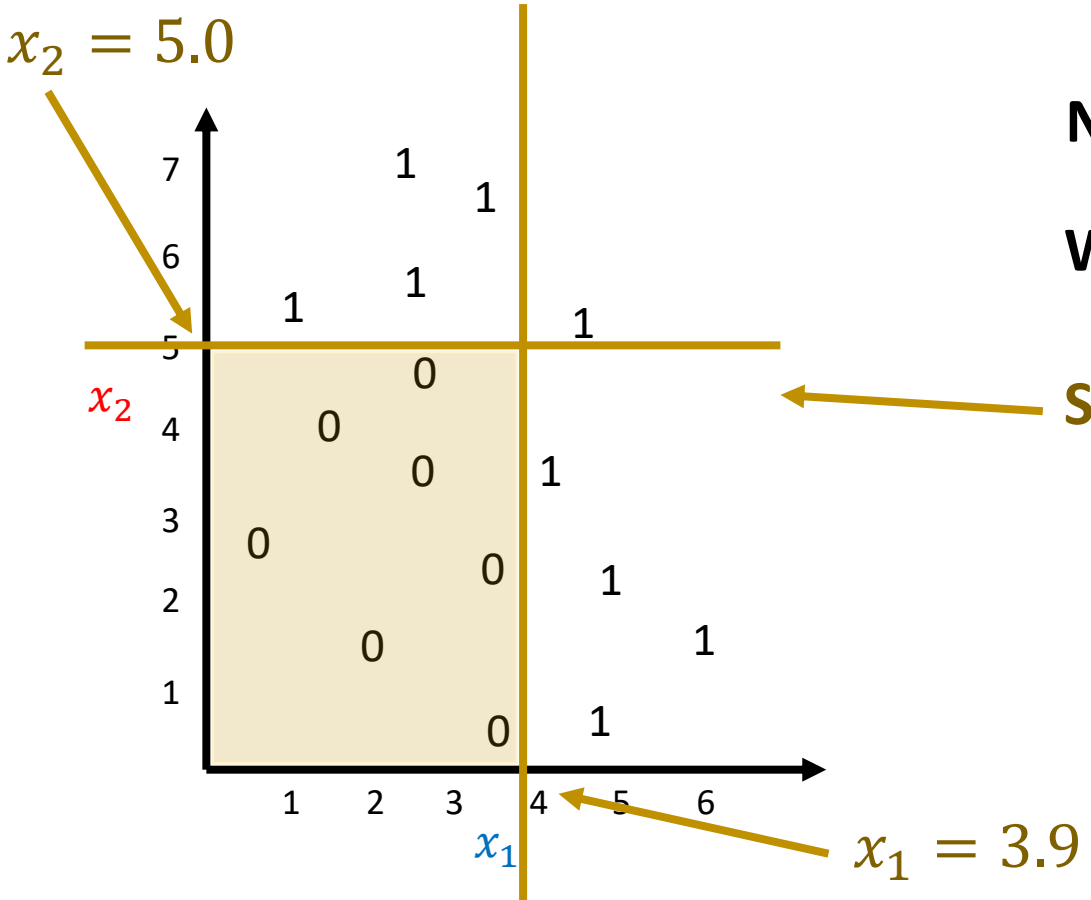
**Could try fitting a curve…?**



# 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.

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




**Not linear separable**

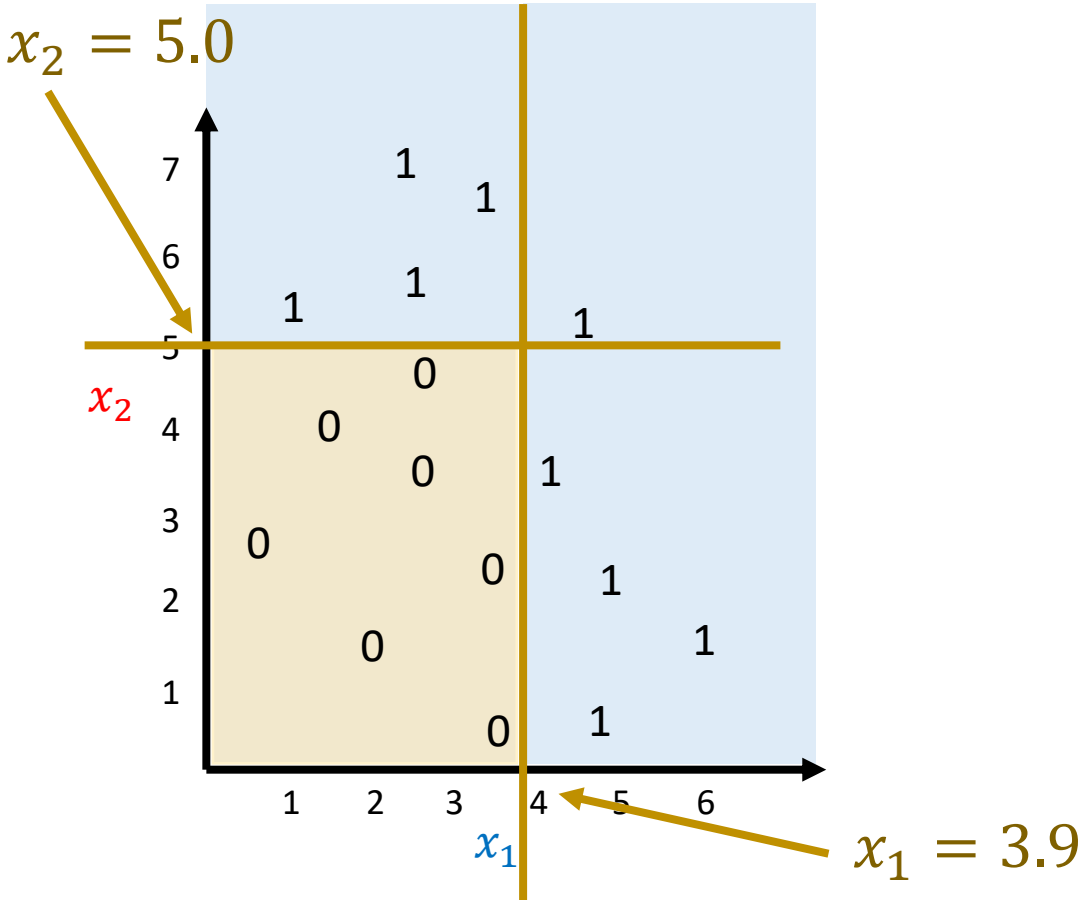
**What can we do?**

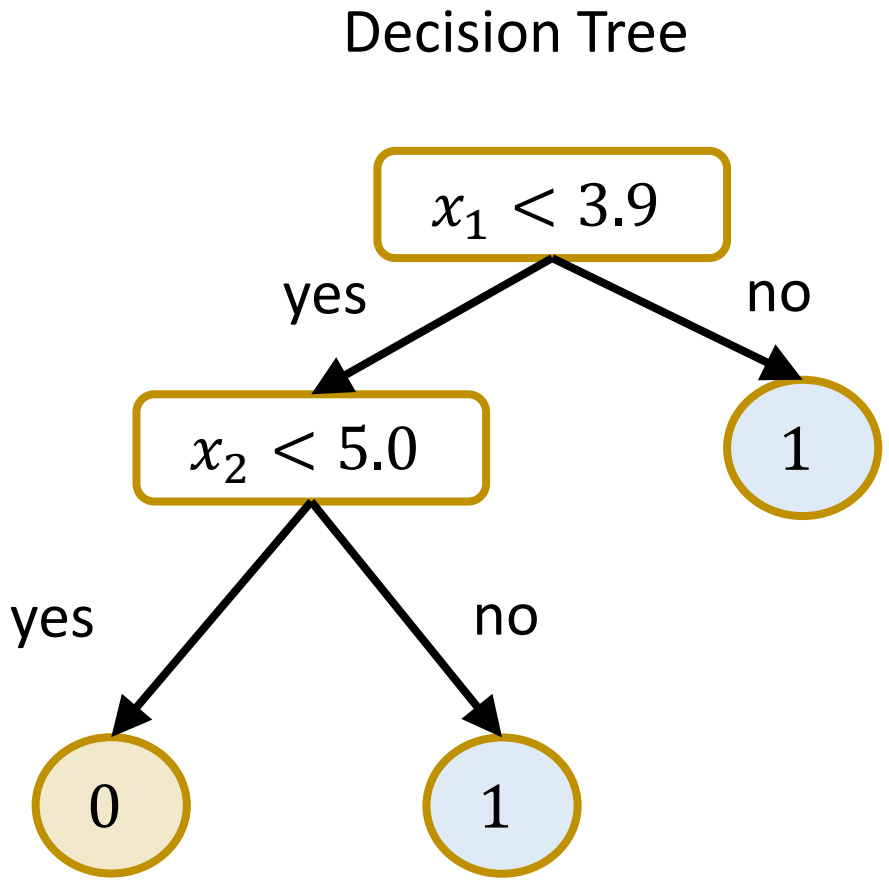
**Sub-divide in a linear way**

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

# Decision Trees

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

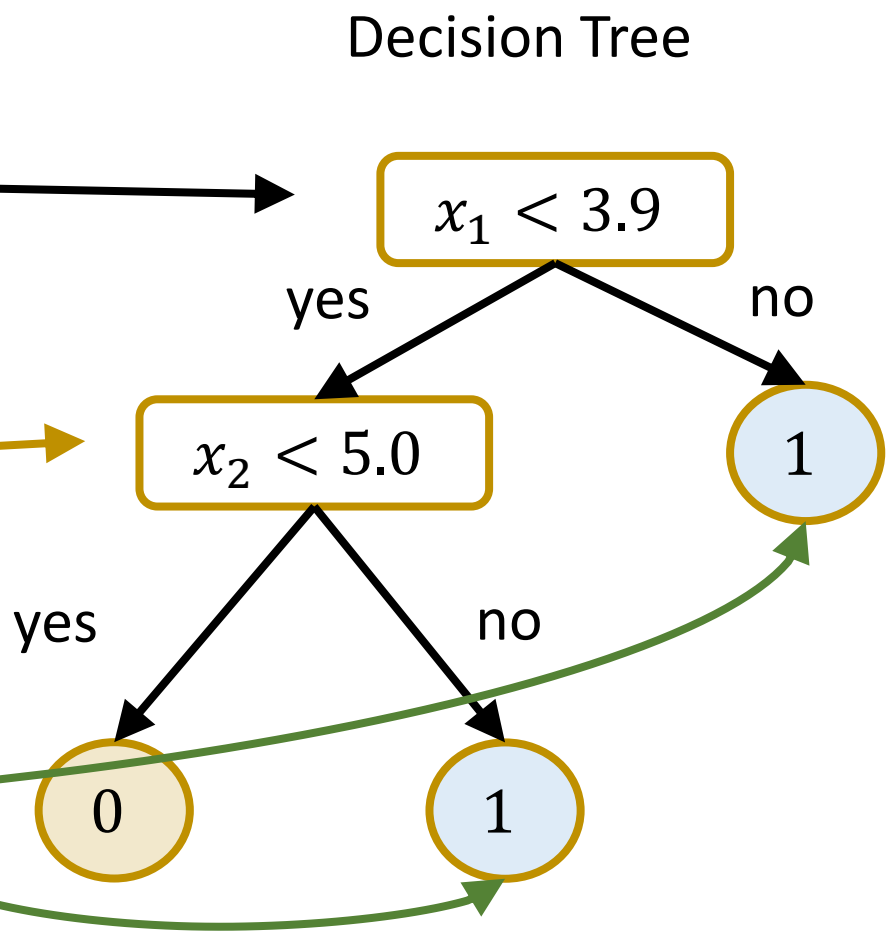


# Decision Tree Structure

Decision Tree Comprises:

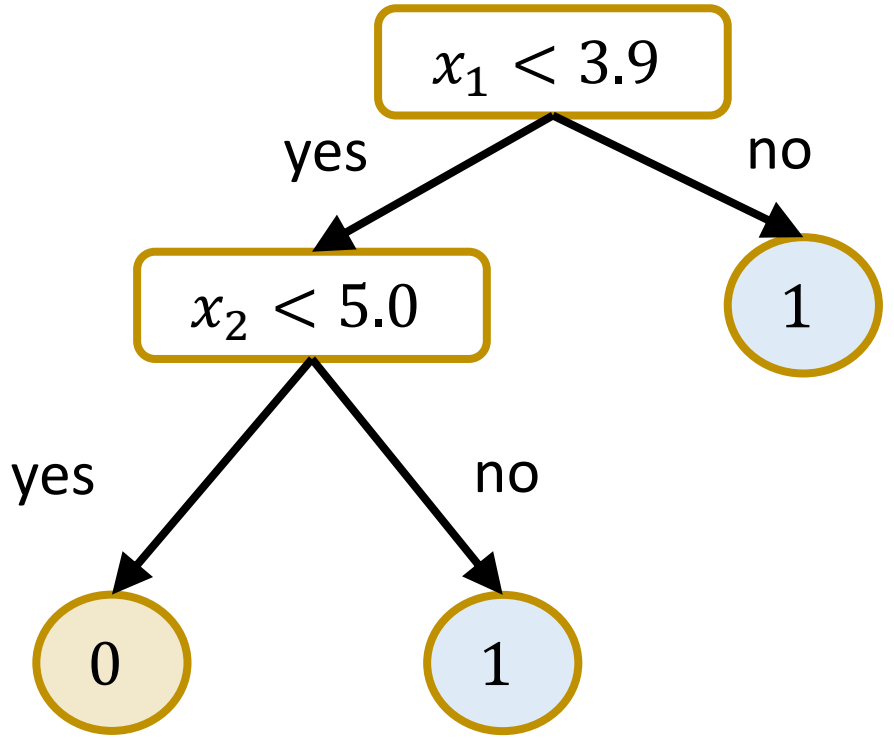
- **Root Node:**
  - No incoming edges
  - Two or more outgoing edges
  - Tests a condition
- **Internal Nodes:**
  - One incoming edges
  - Two or more outgoing edges
  - Tests a condition
- **Leaf / Terminal Nodes:**
  - One incoming edges
  - No outgoing edges
  - Gives the outcome prediction



# Decision Tree Definition

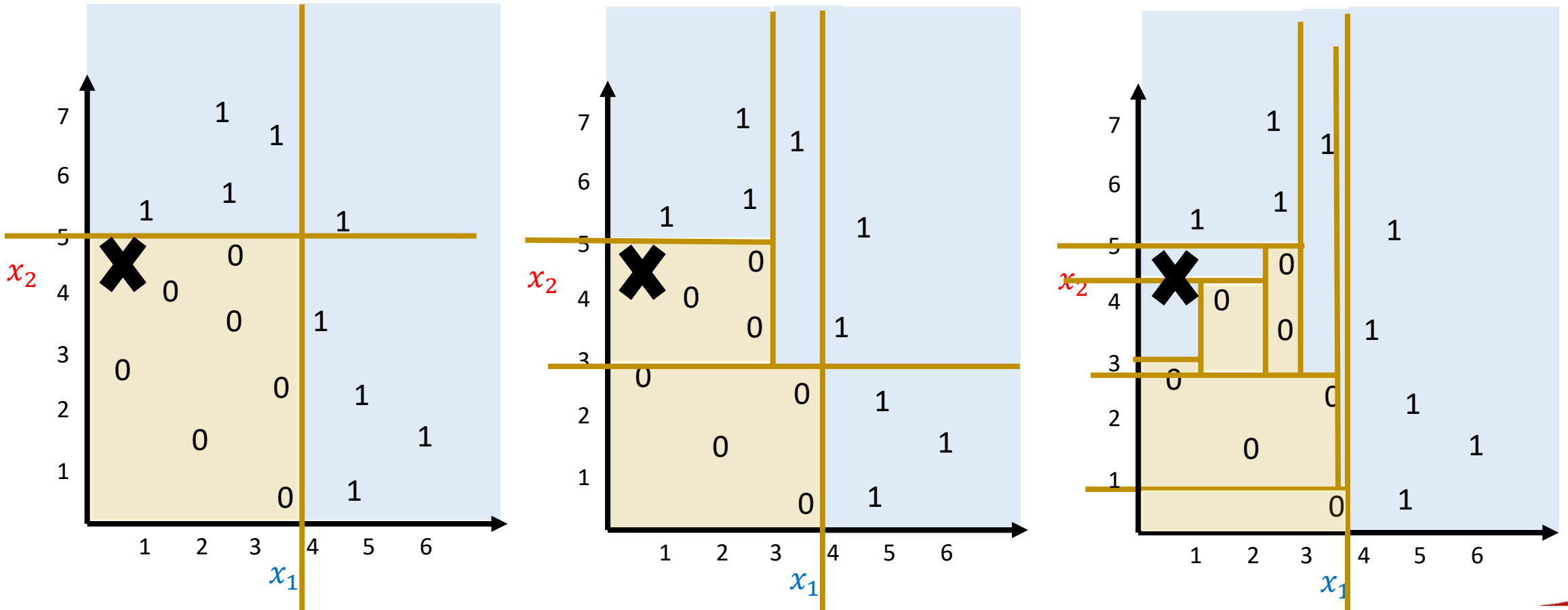
A **decision tree classifier** organizes a series of test questions and conditions in a tree-like structure containing

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



# Is the solution unique?

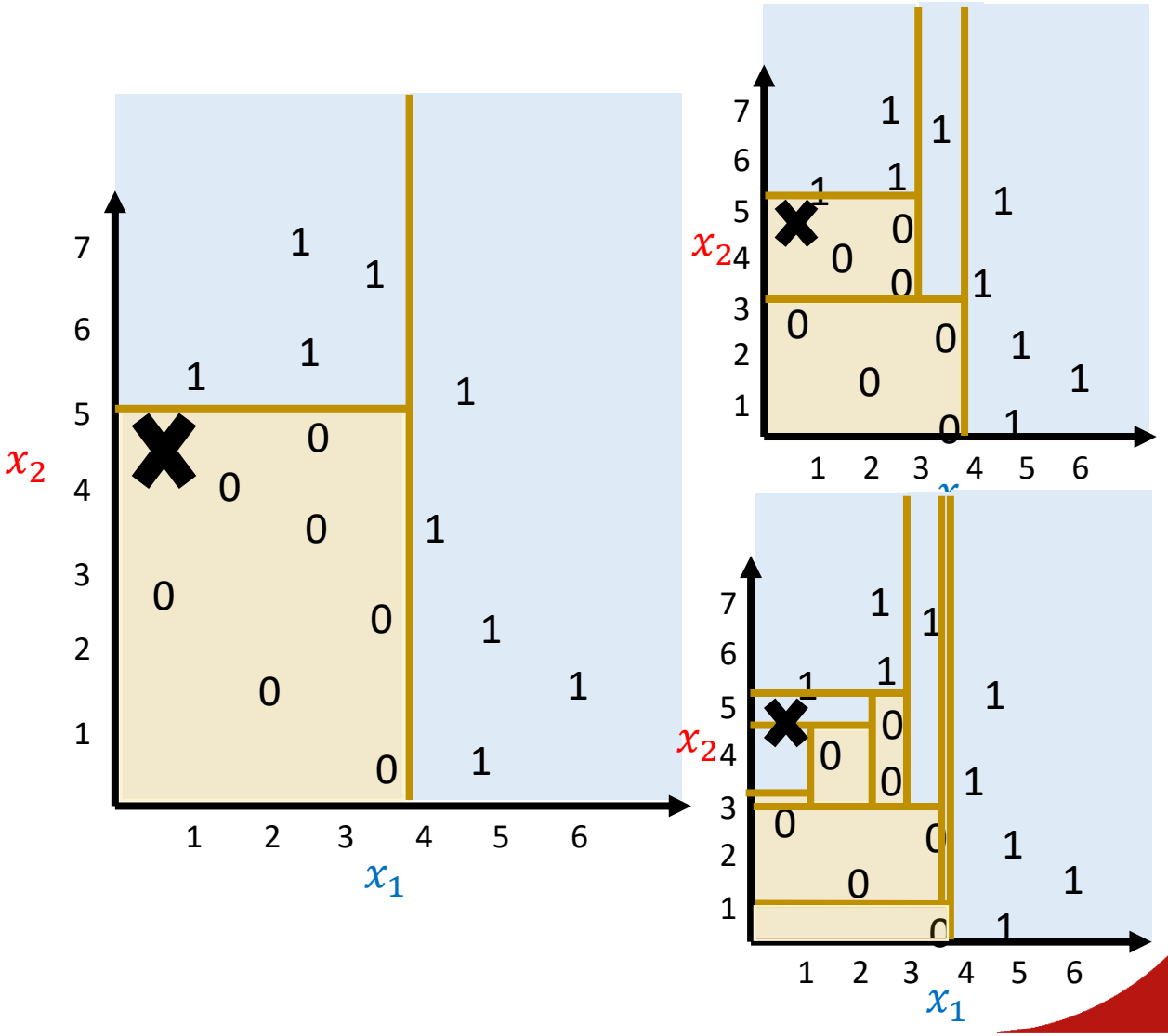
Can we partition the space in multiple ways?



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**.



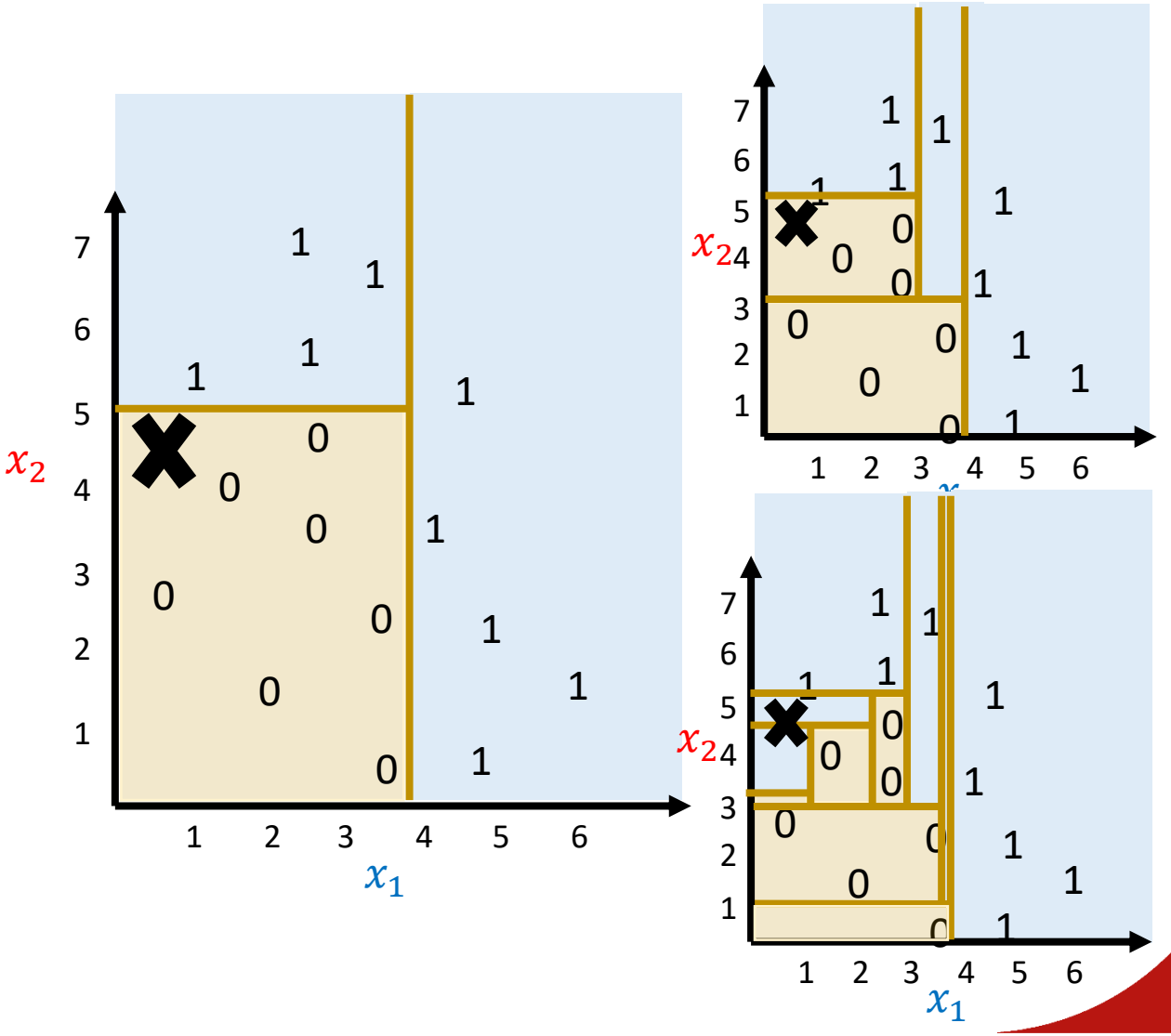
# What is Good?

## 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

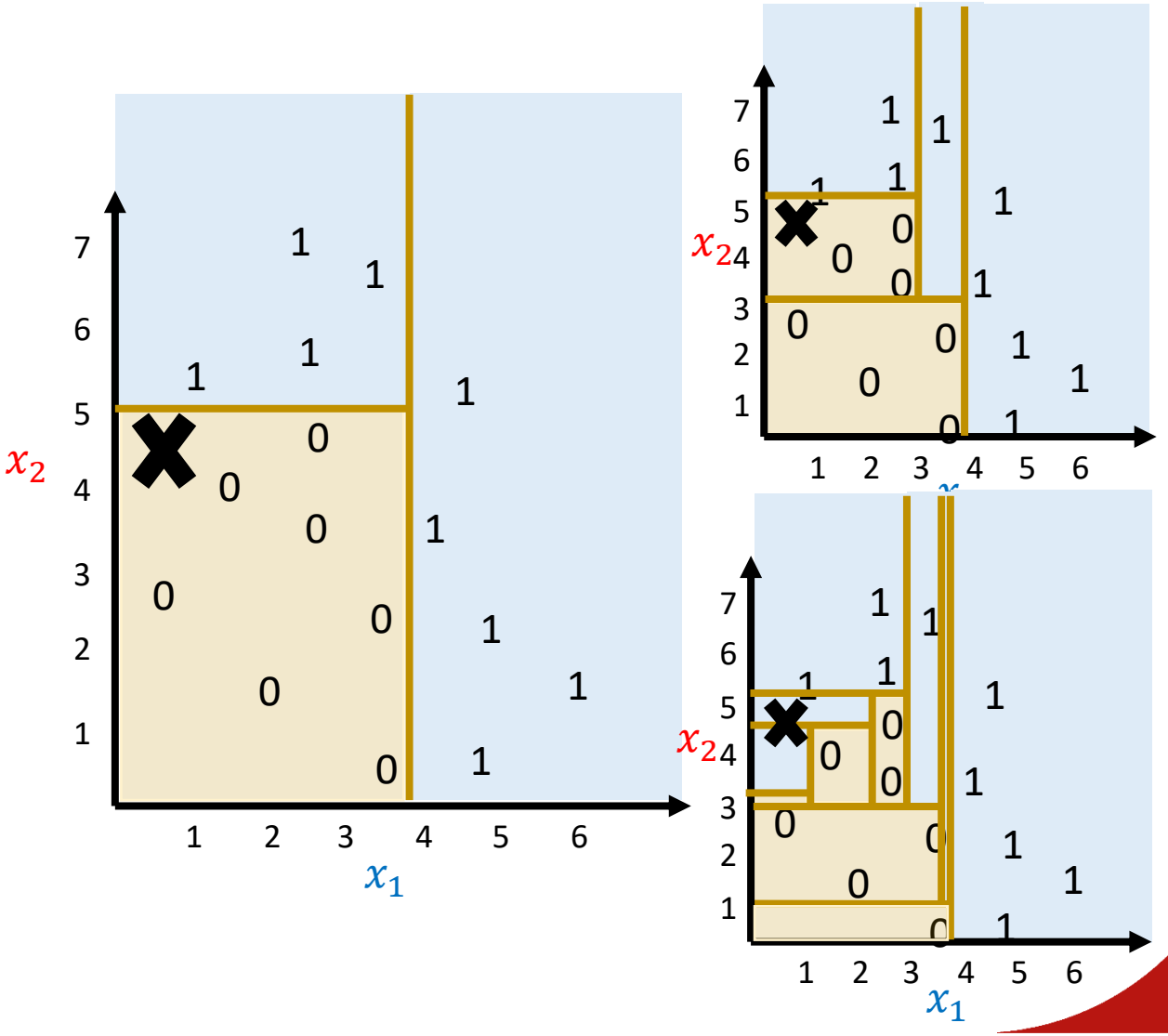


# ID3 Algorithm (1983)

Test the most important feature first

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

Else, choose the next most important feature



# Entropy (Binary)

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

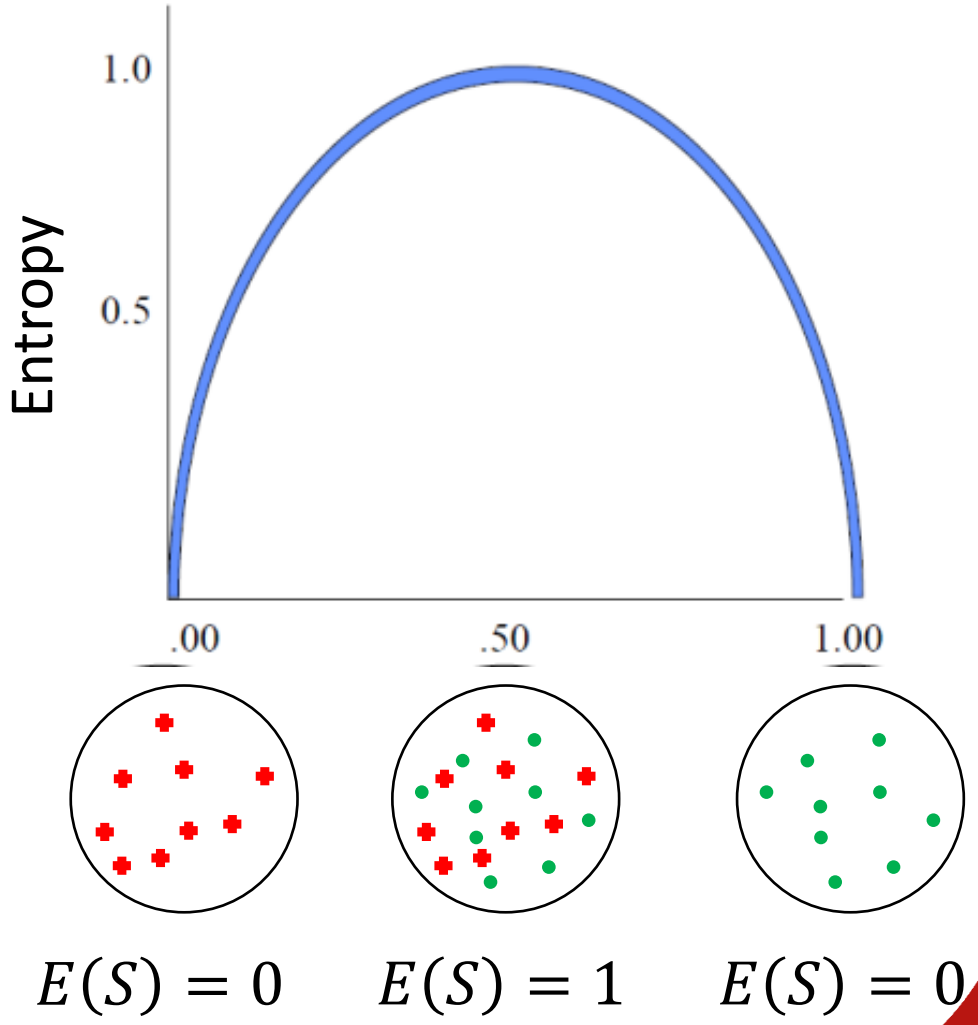
Let

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

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

- Highest when uncertainty is greatest

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



Lancaster University

# Information Gain

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

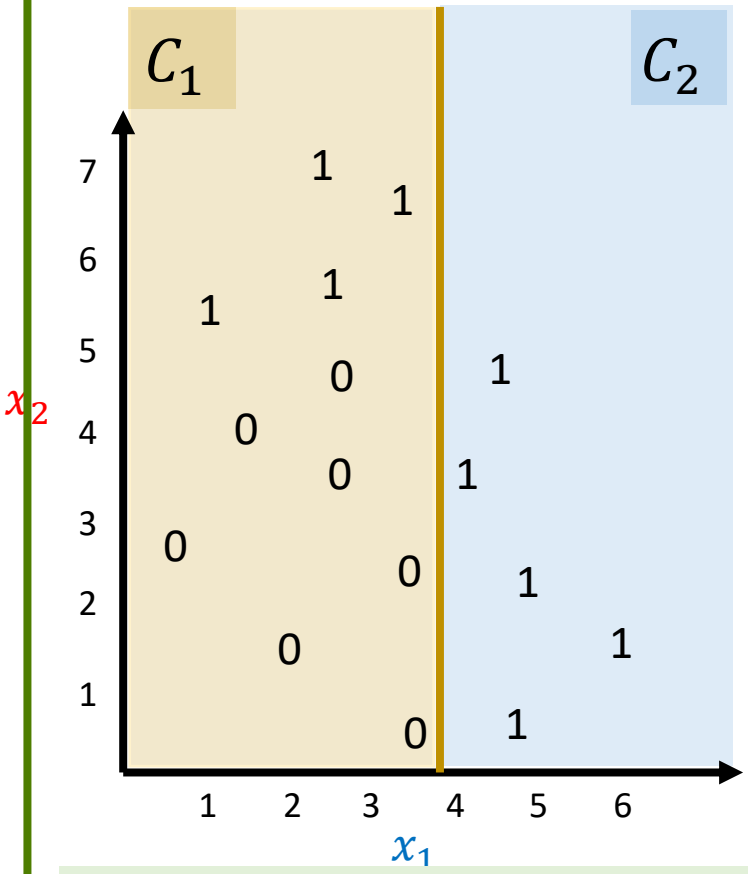
Gain(S,condition) = Entropy(parent) – [Weighted Average] Entropy(Children)

## 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 $x_1 < 3.9$




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

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

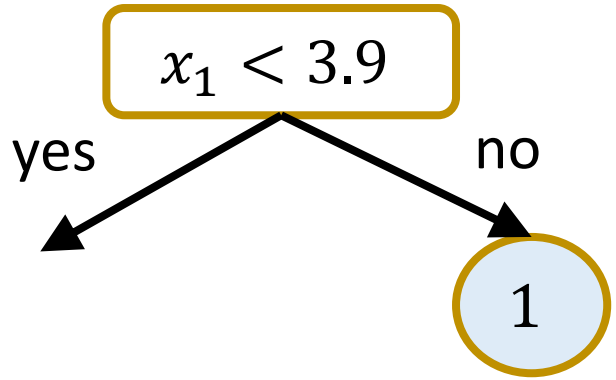
$$E(C_1) \approx 0.946$$

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

$$E(C_2) = 0$$

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

Decision Tree




$$E(S) \approx 0.989$$

$$E(S) \approx 0.650$$

Information Gain:

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

Lancaster University

# 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

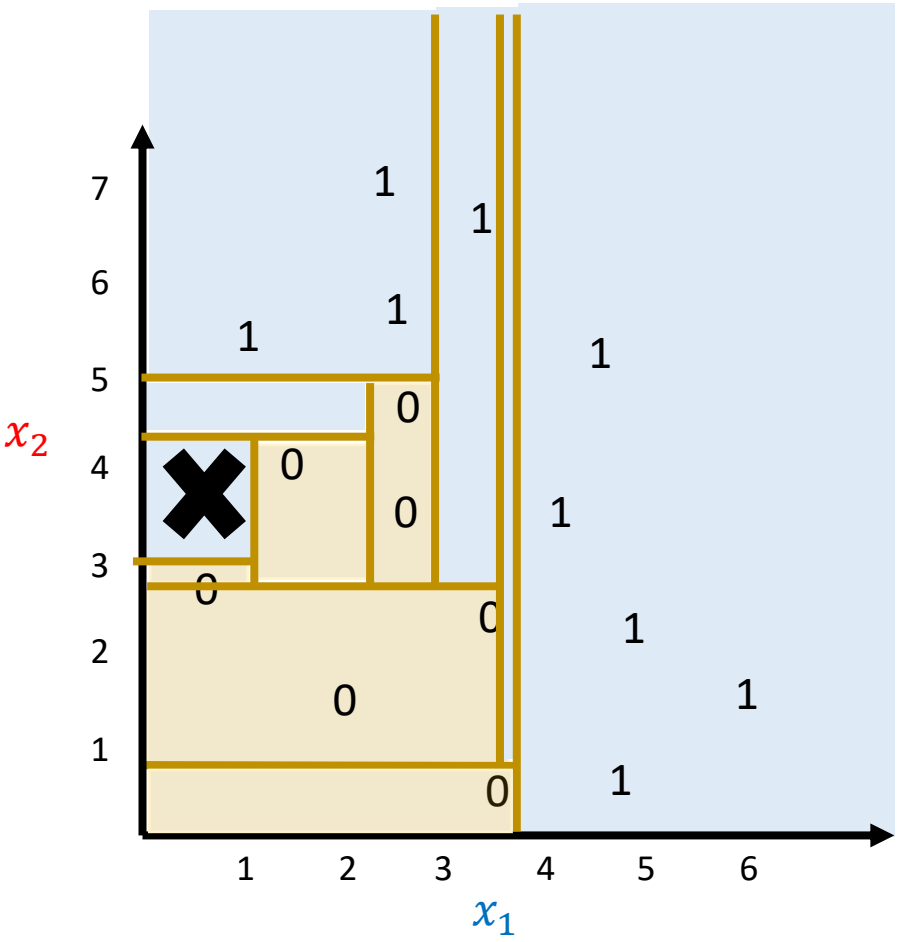
**Overfitting** happens when your model fits too well to the training set.

Overfitting means memorizing

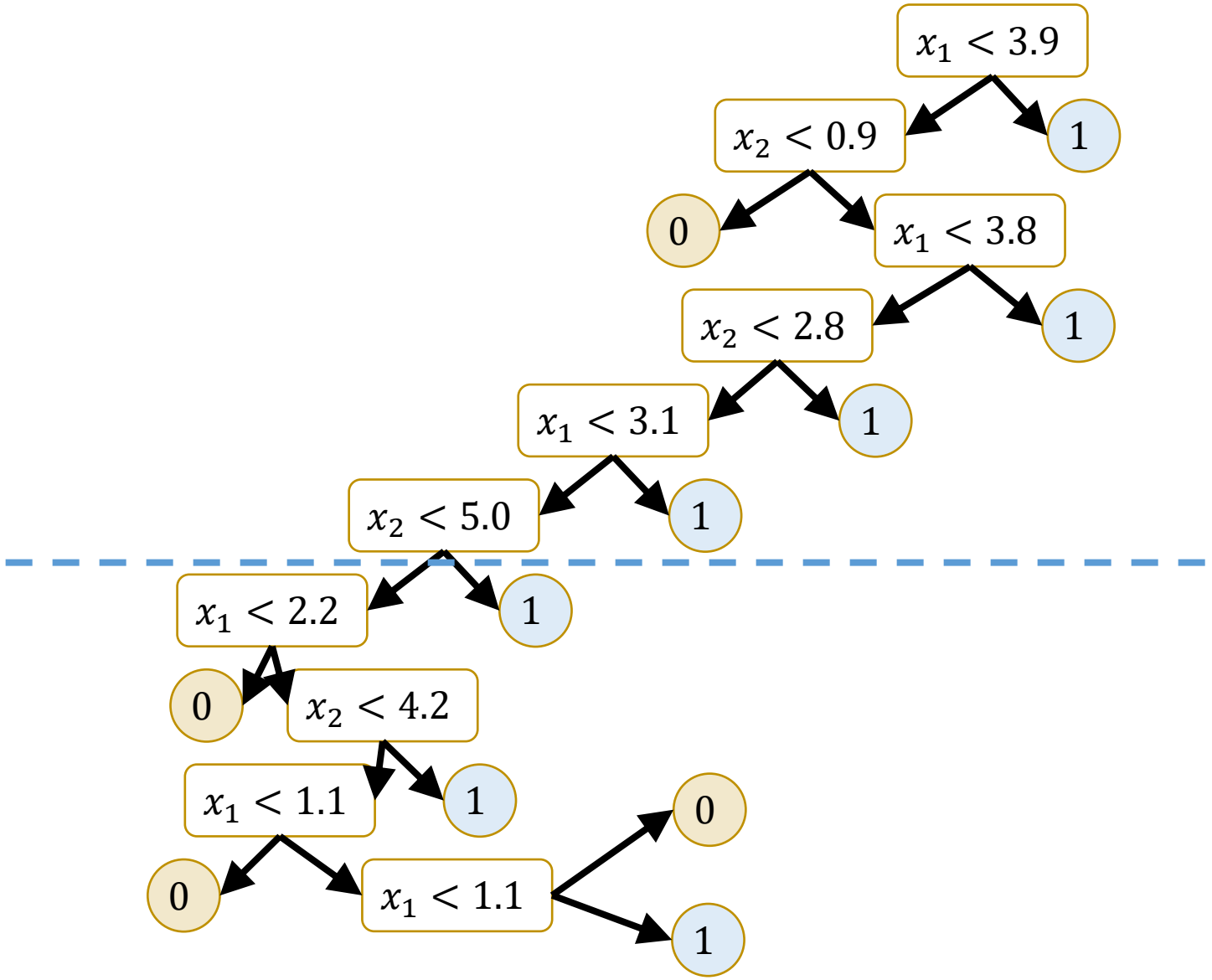
Memorizing is not learning!

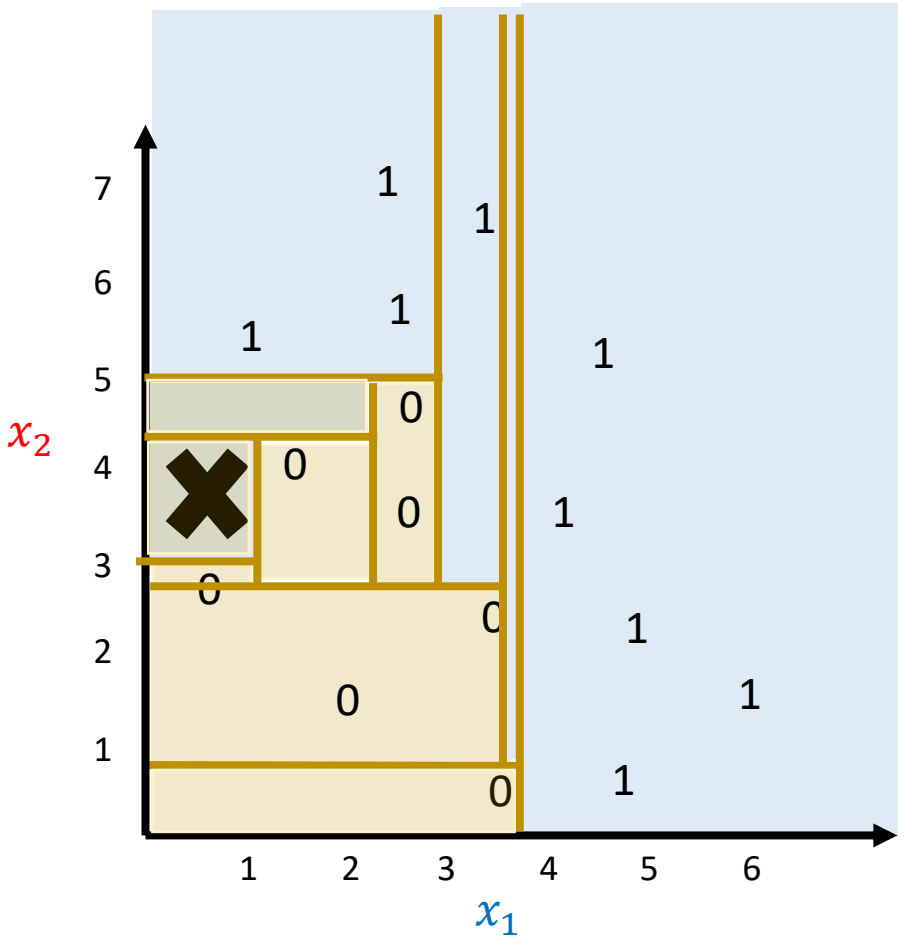
It then becomes difficult for the model to generalize to new examples that were not in the training set.

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



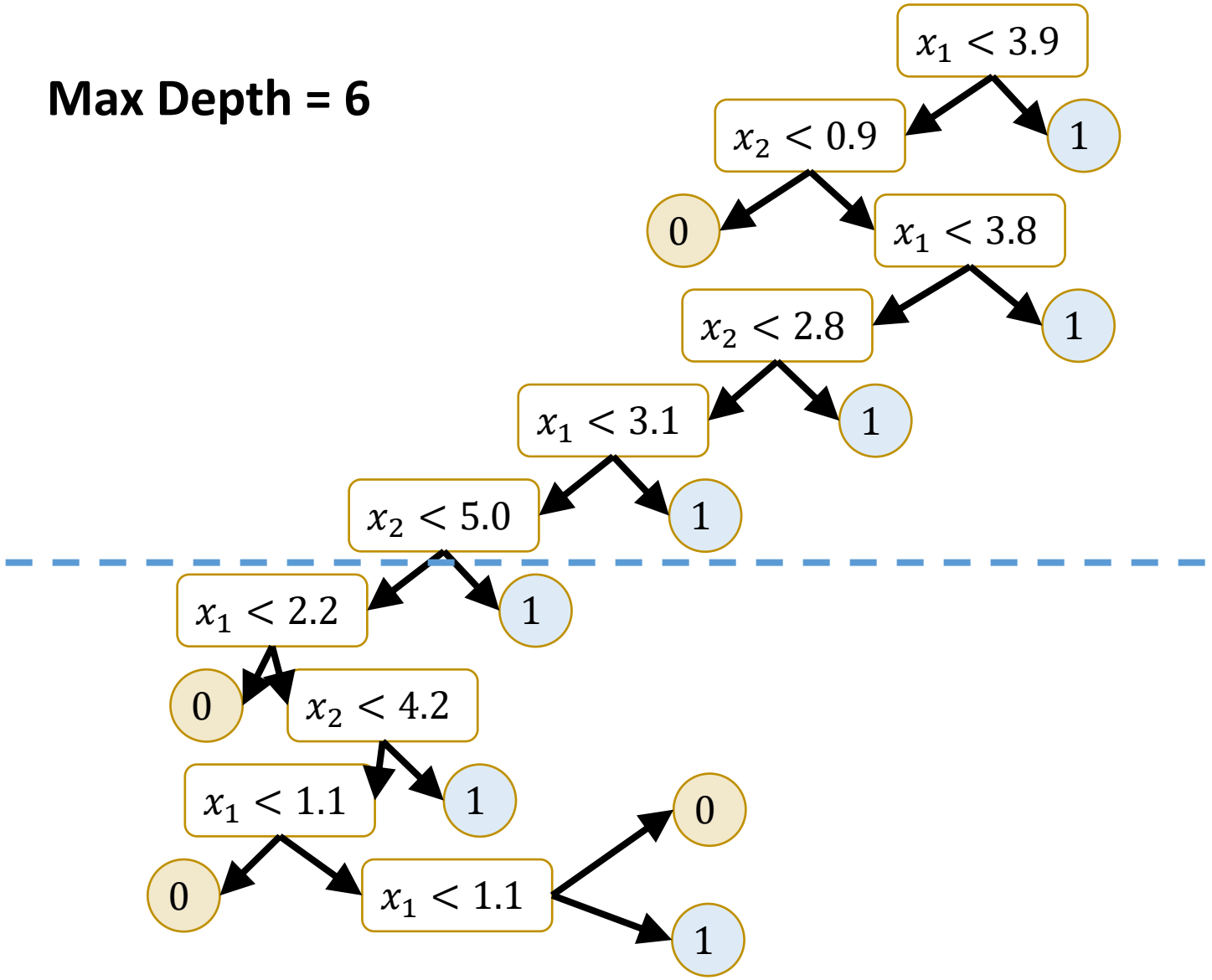
# Avoiding Overfitting

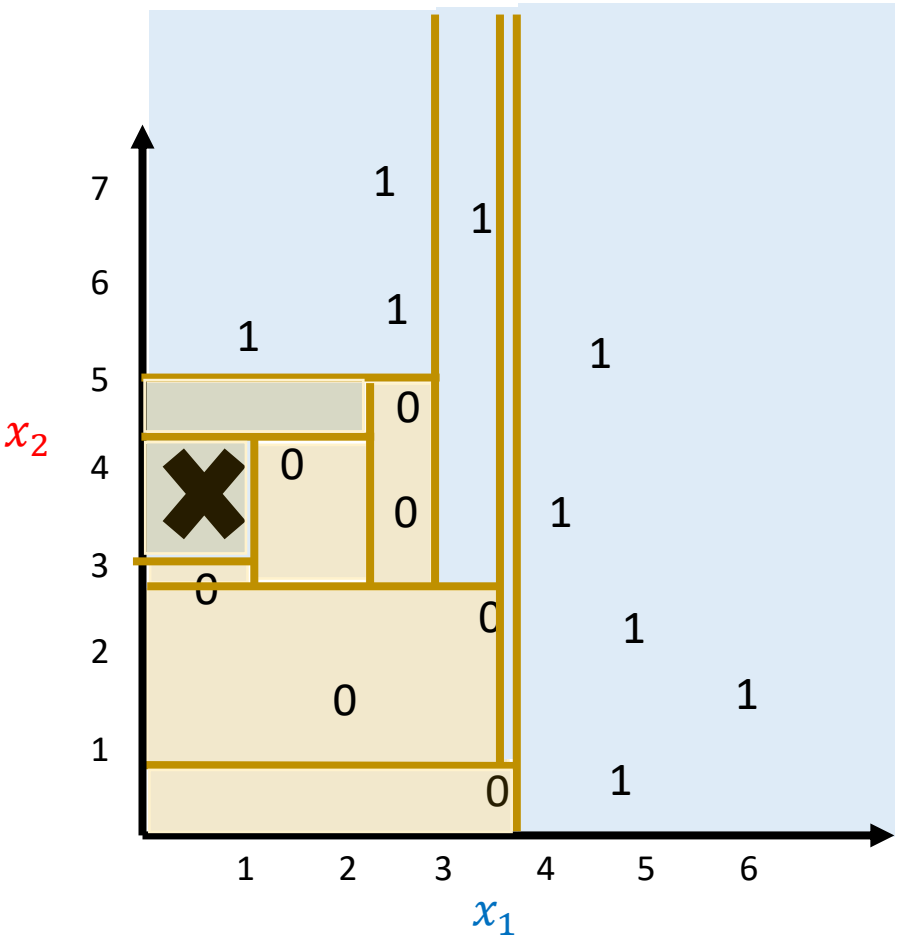


# Avoiding Overfitting: Set Max Depth

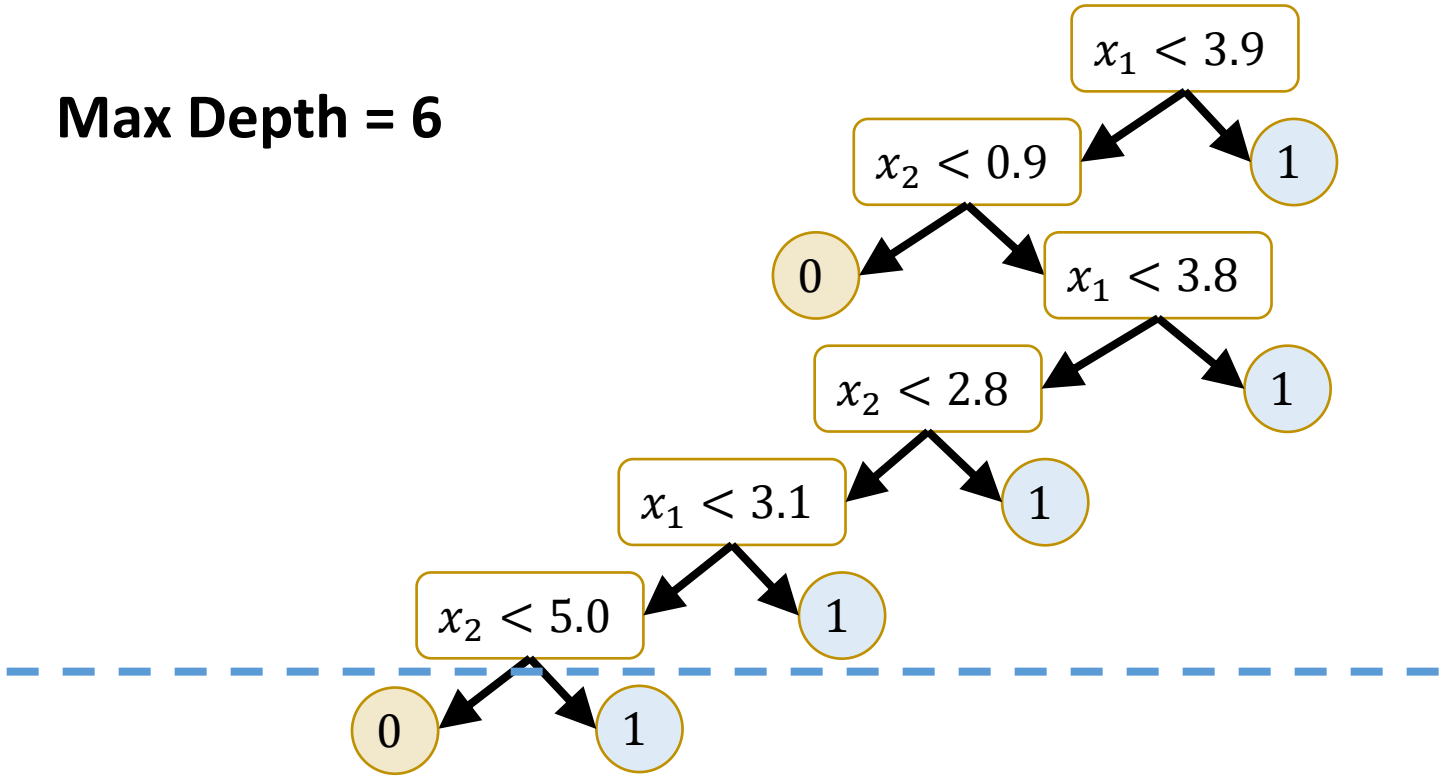
**Max Depth = 6**

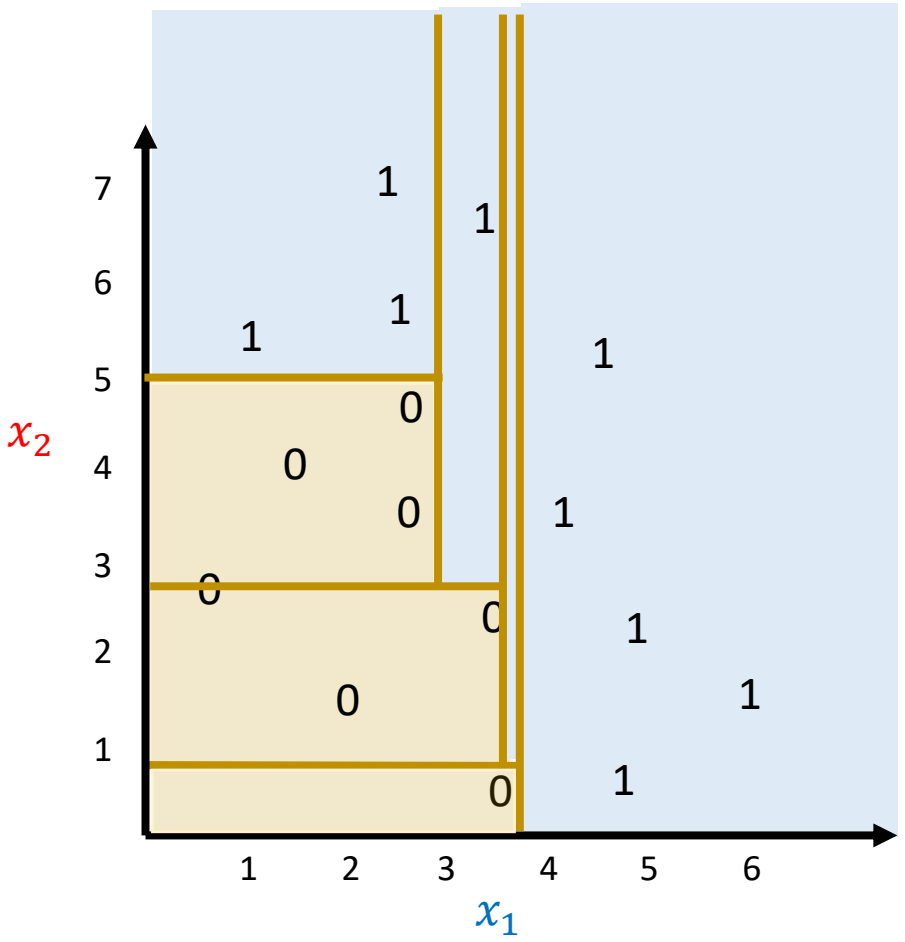


# Avoiding Overfitting: Set Max Depth

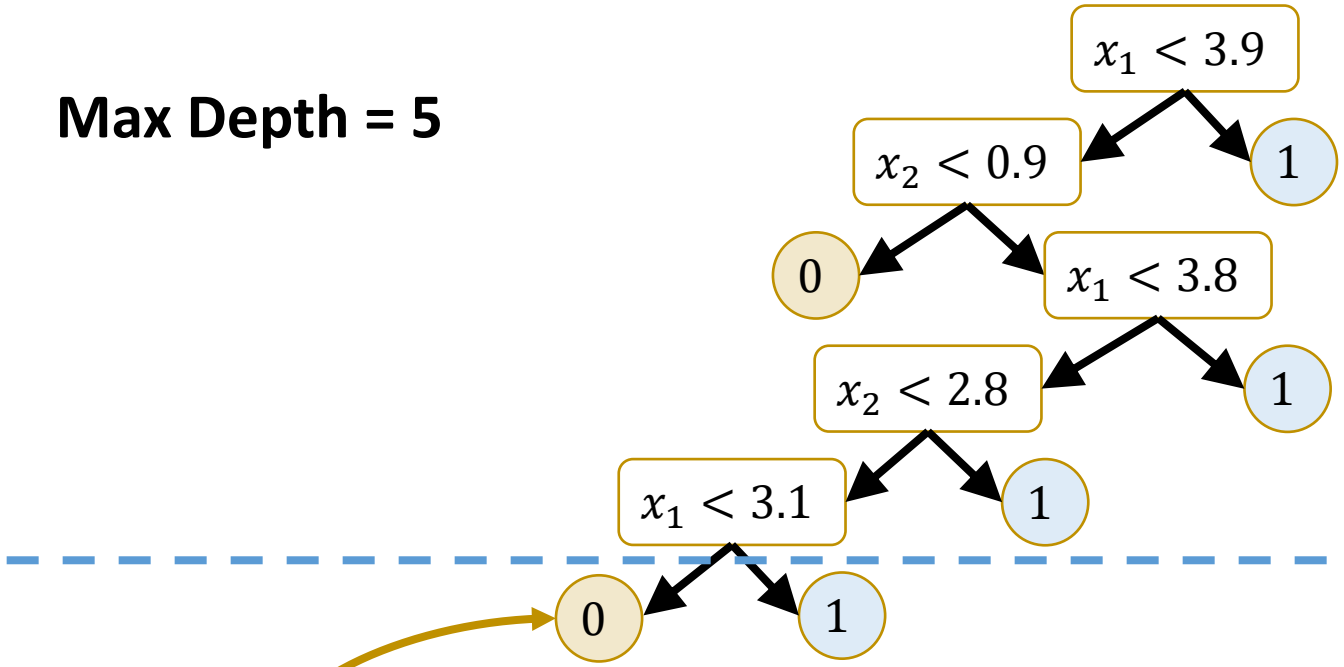
**Max Depth = 6**



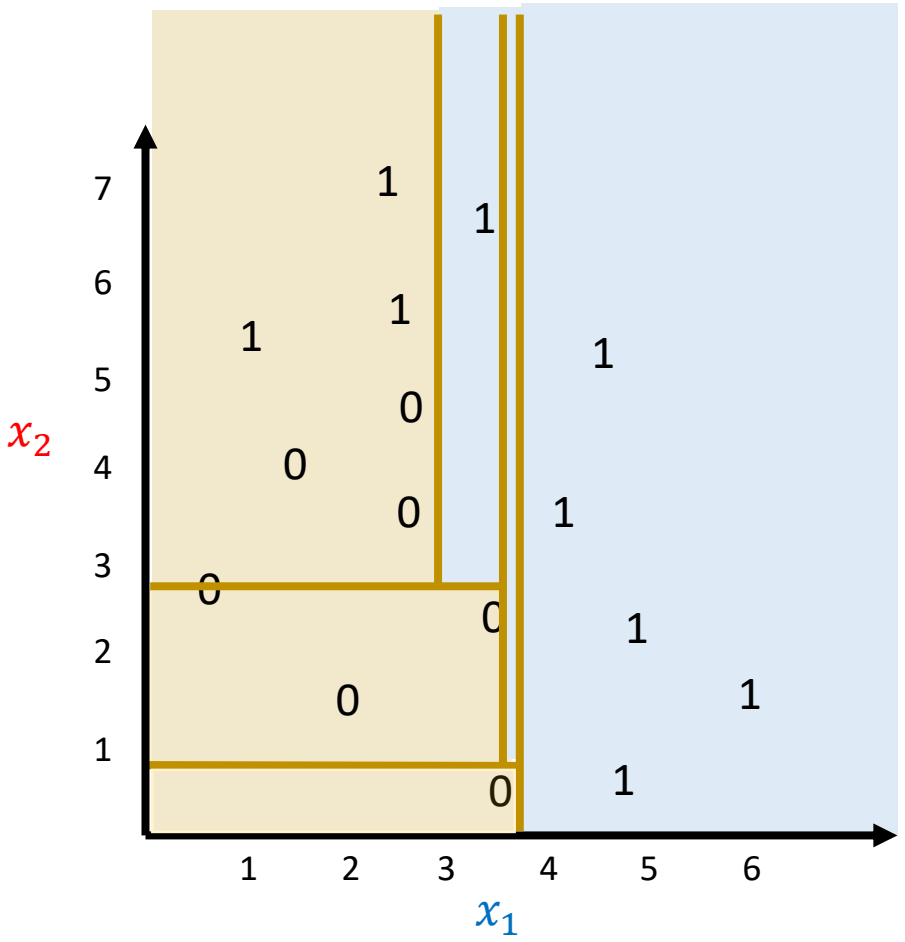
# Avoiding Overfitting: Set Max Depth

**Max Depth = 5**




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

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



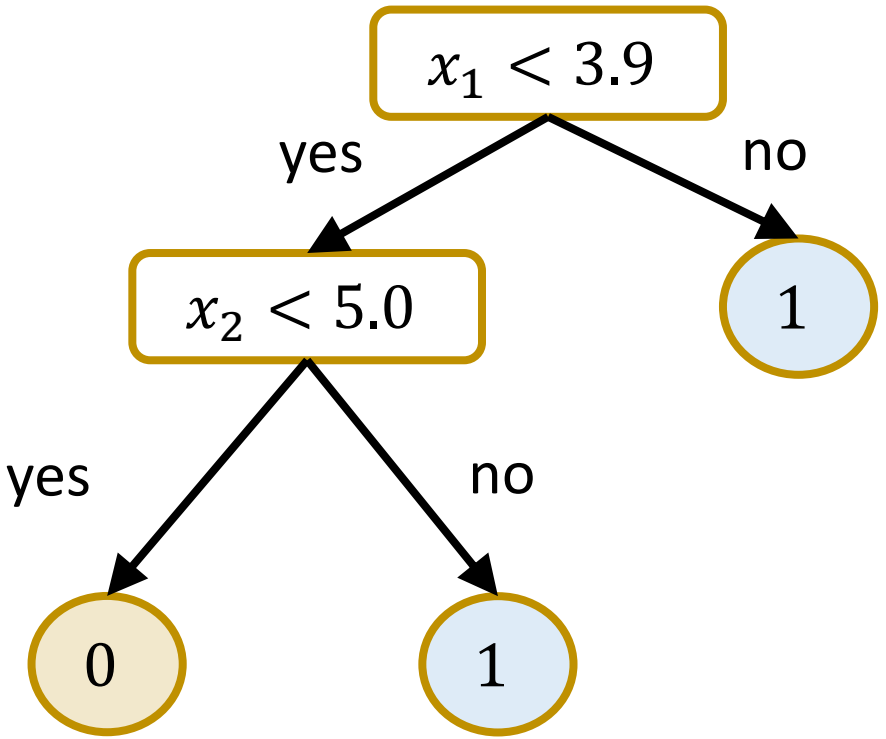
# Avoiding Overfitting

## 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

# Ensemble Approaches

## Idea

- Ensemble approaches combine multiple models to improve performance

## Types of Approach

- **Bootstrap Aggregation:** Train same algorithm on different subsets.
- **Boosting:** Sequentially correct errors from previous models

## Aims

- Reduce Error and Improve Accuracy
- Build more stable predictions

# Random Decision Forest

## Aim

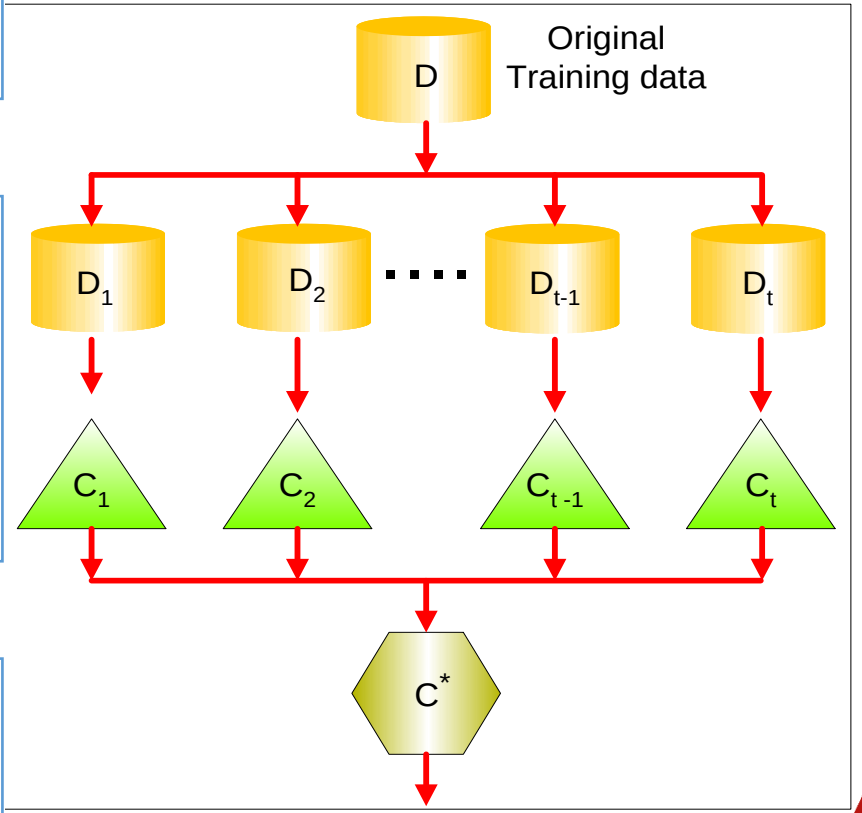
- Train a classification model based on decision trees

## 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



Lancaster University

# Random Decision Forest

## Step-by-step Process

1. **Boostrapping:** Select random sample of the dataset with replacement
2. **Random Feature Selection:** For each split, select a random subset of features
3. **Build Decision Tree:** Build decision trees in full
4. **Ensemble Aggregation:** Use majority voting to choose the modal class

## Hyper-parameters

- $n$: number of trees
- $f$: number of features
- $d$: depth of each tree

Lancaster University

# Random Forest Pros and Cons

## Advantages

- Robust to overfitting
- More accurate than individual models
- Works well with high-dimensional data
- Helps to understand feature significance

## Disadvantages

- Computationally intensive
- Less interpretable than single model
- Storage requirements can be large

# Decision Trees vs Random Forest

## 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.

## Advantages

- Robust to overfitting
- More accurate than individual models
- Works well with high-dimensional data
- Helps to understand feature significance

## Disadvantages

- Computationally intensive
- Less interpretable than single model
- Storage requirements can be large

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

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?** |
| $s_1$ | Yes | No | Yes | No | Some | Cheap | Yes | Yes | French | 30-60 | $y_1 = ?$ |
| $s_2$ | No | Yes | No | Yes | Full | Mid-range | Yes | No | Thai | 0-10 | $y_2 = ?$ |
| $s_3$ | Yes | Yes | Yes | Yes | None | Expensive | No | Yes | Italian | >60 | $y_3 = ?$ |

**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

Lancaster University

# Example (Decision Trees)

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

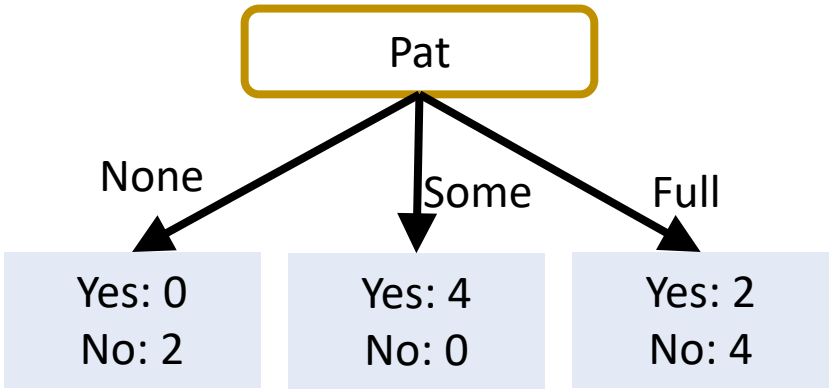
## Start building Decision Tree – which feature to start with?

| 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 | $y_{12} = Yes$ |

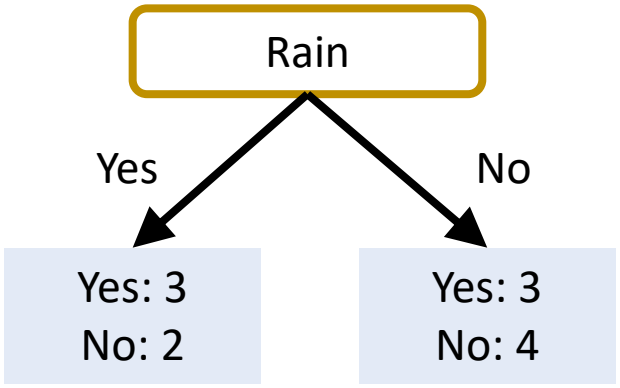
**Key:**

**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:**

# 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

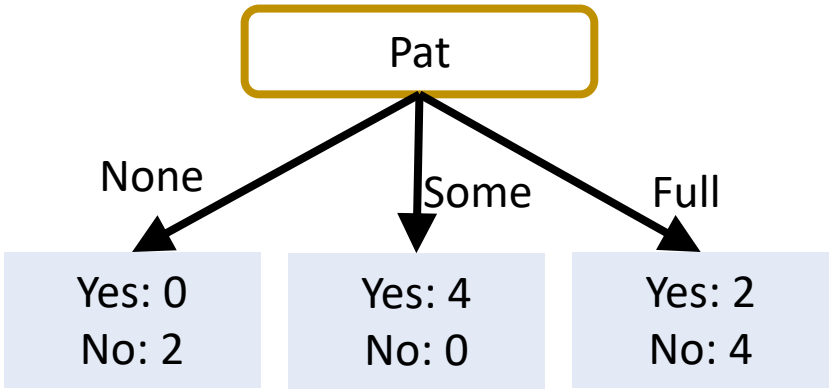
## Start building Decision Tree – which feature to start with?

| 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 | $y_{12} = Yes$ |

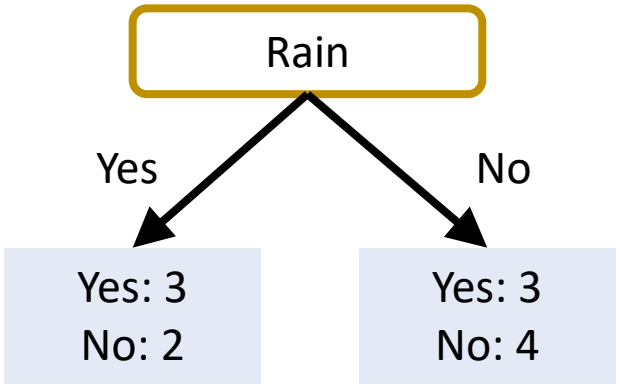
**Key:**

**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

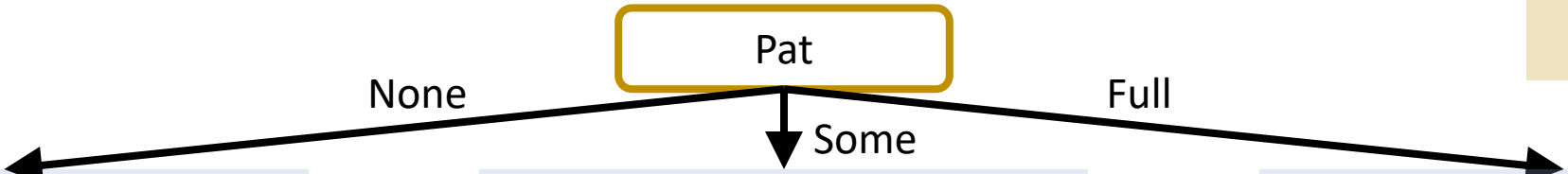
Start building Decision Tree – which feature to start with?

**Overall Wait Decisions:**

**Yes:** 6
**No:** 6

$$E(S) = 1$$

Start with Patrons?




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

Lancaster University

# Example

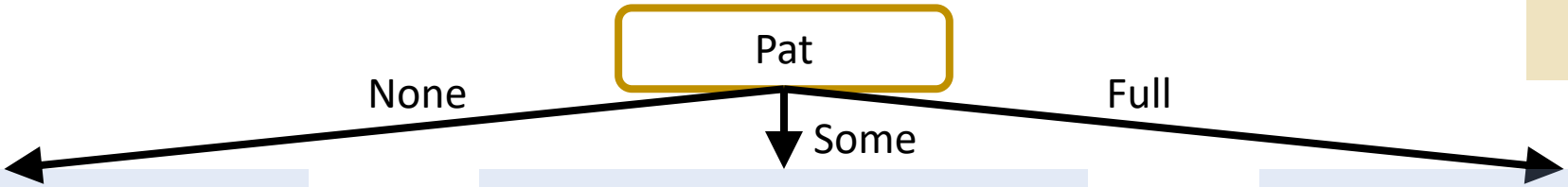
Start building Decision Tree – which feature to start with?

**Overall Wait Decisions:**

**Yes:** 6
**No:** 6

$$E(S) = 1$$

Start with Patrons?




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

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

# Example

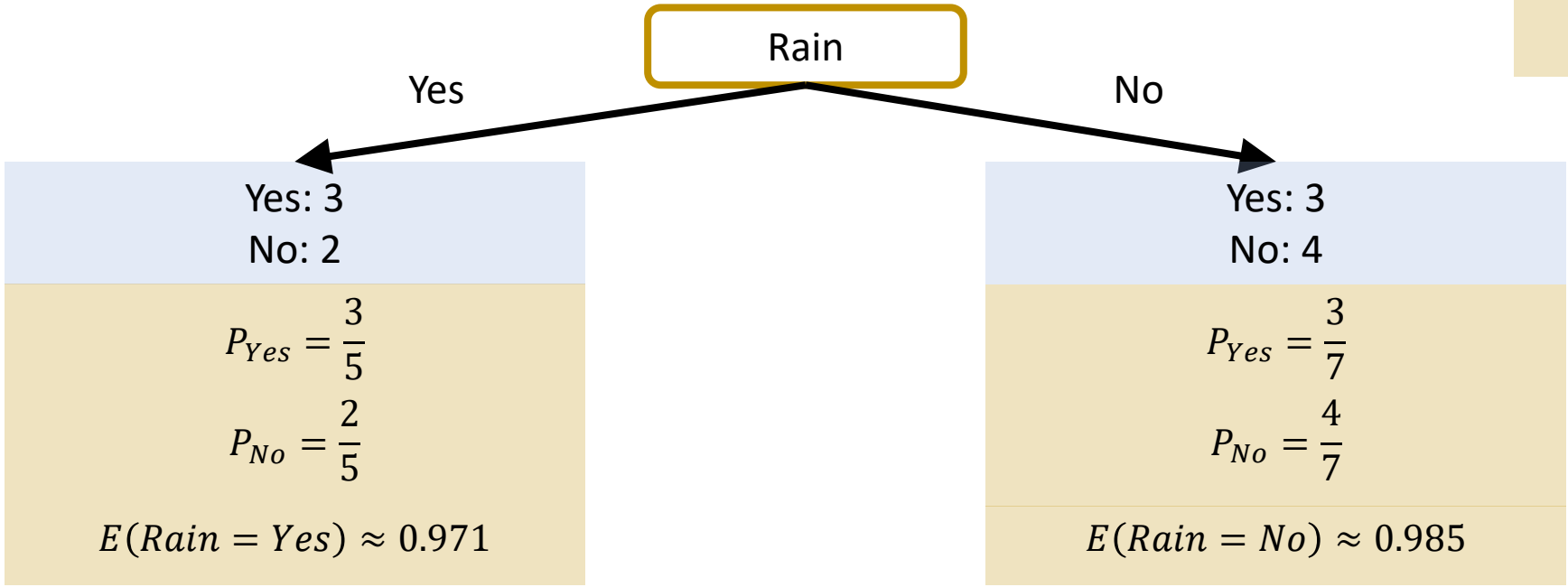
Start building Decision Tree – which feature to start with?

**Overall Wait Decisions:**

**Yes:** 6
**No:** 6

$$E(S) = 1$$

Start with Rain?




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

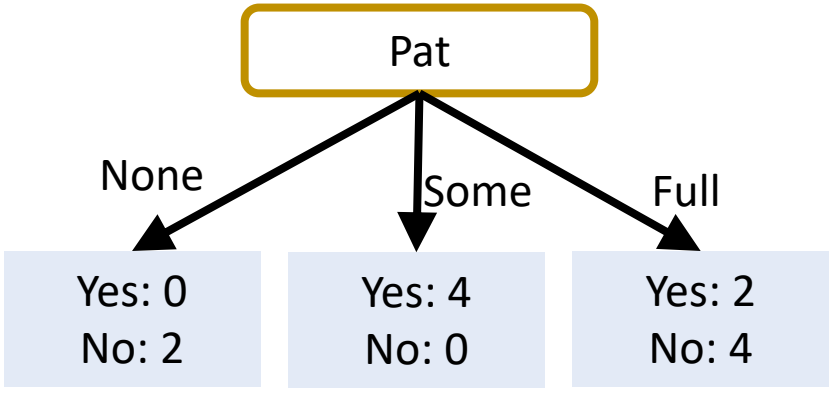
## Start building Decision Tree – which feature to start with?

| 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:**

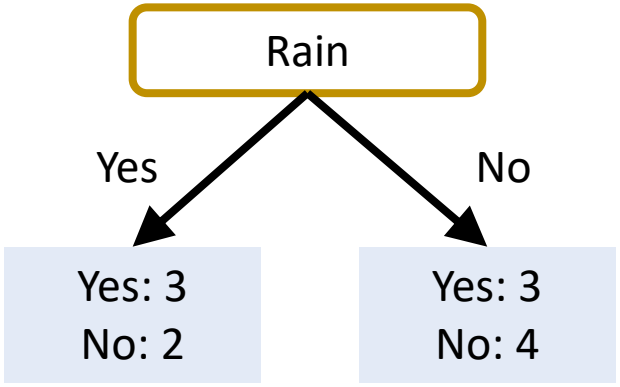
**Alt:** Alternative restaurant nearby
**Bar:** Bar area to wait
**F/S:** Yes on Fridays and Saturdays
**Hun:** whether hungry
**Pat:** how many people in restaurant
**Price:** price range
**Rain:** raining outside
**Res:** whether we made a reservation
**Type:** Cuisine
**Est:** Estimated wait

### 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 |

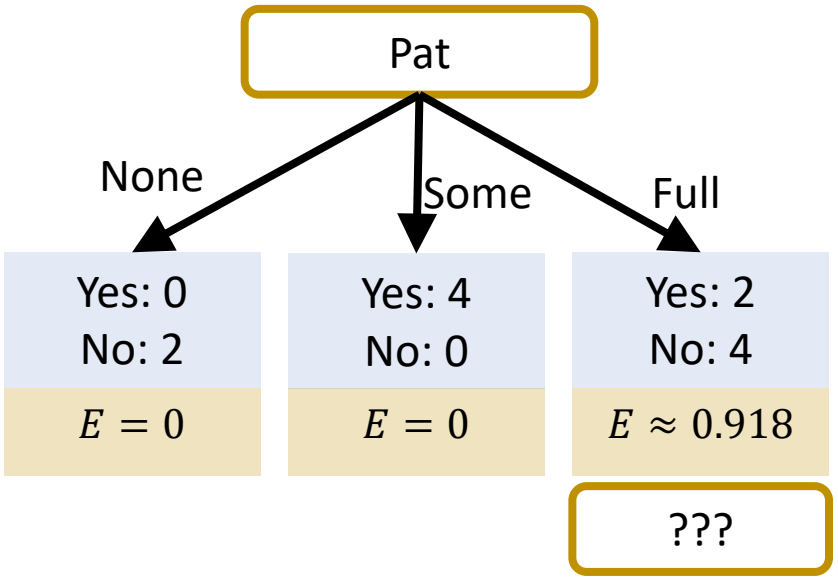
Patrons has highest gain

=> 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

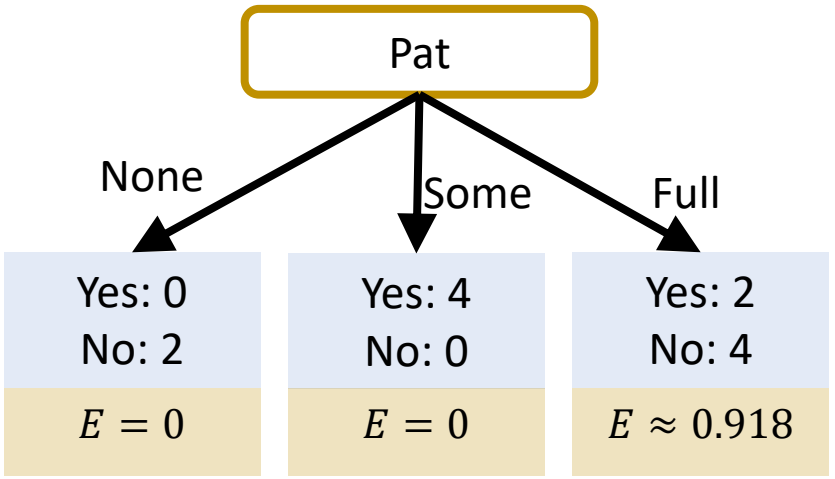
# Example

## 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 | $y_{12} = Yes$ |

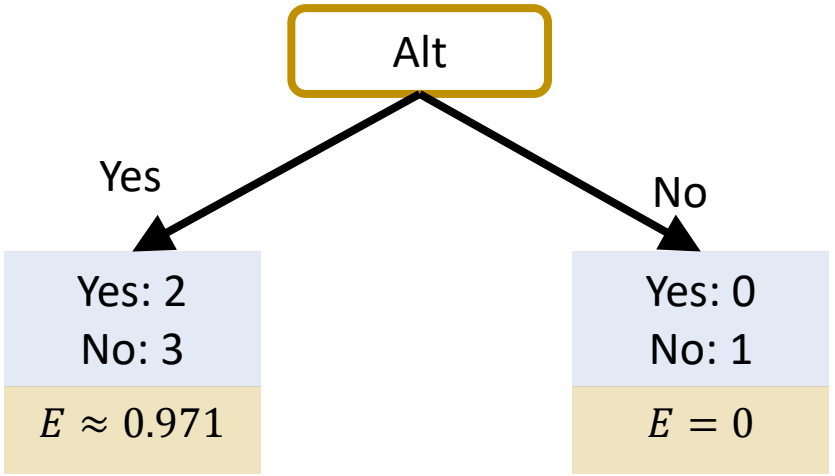
**Key:**

**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




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

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



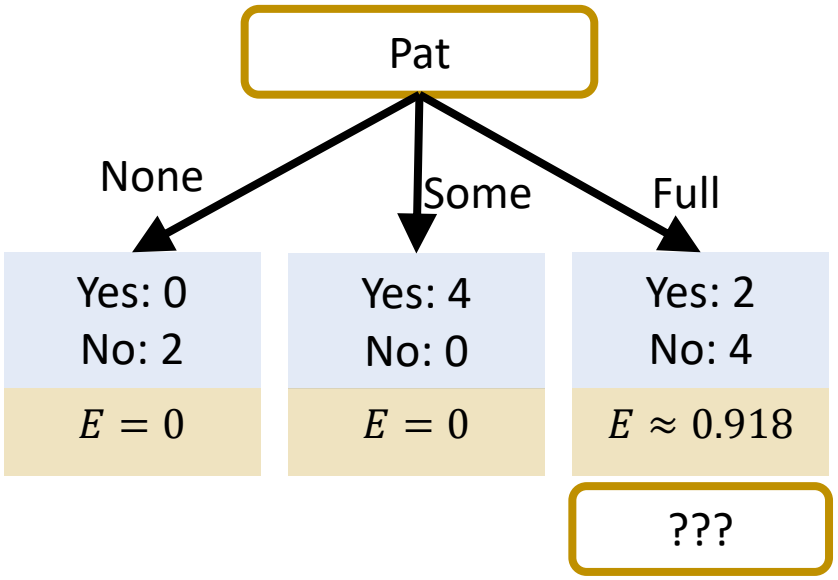
# Example

## 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

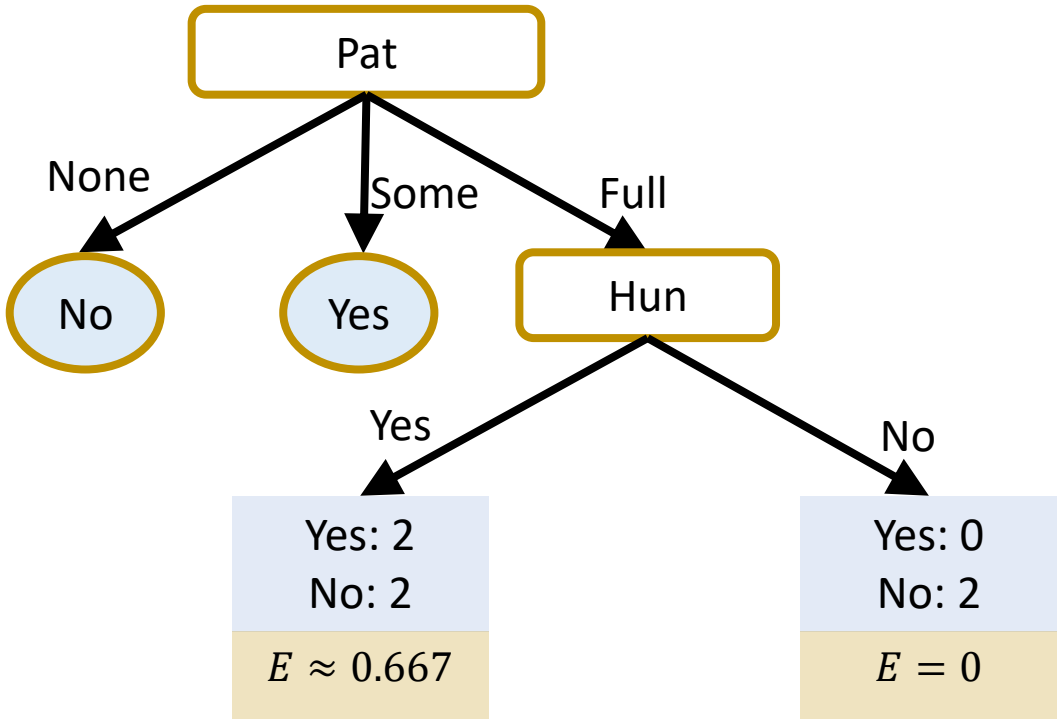



| Feature | Gain |
|---|---|
| Alt | 0.109 |
| Bar | 0.000 |
| F/S | 0.109 |
| Hun | 0.251 |
| Pat | 0.000 |
| Price | 0.251 |
| Rain | 0.044 |
| Res | 0.251 |
| Type | 0.251 |
| Est | 0.251 |

# 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 | $y_{12} = Yes$ |



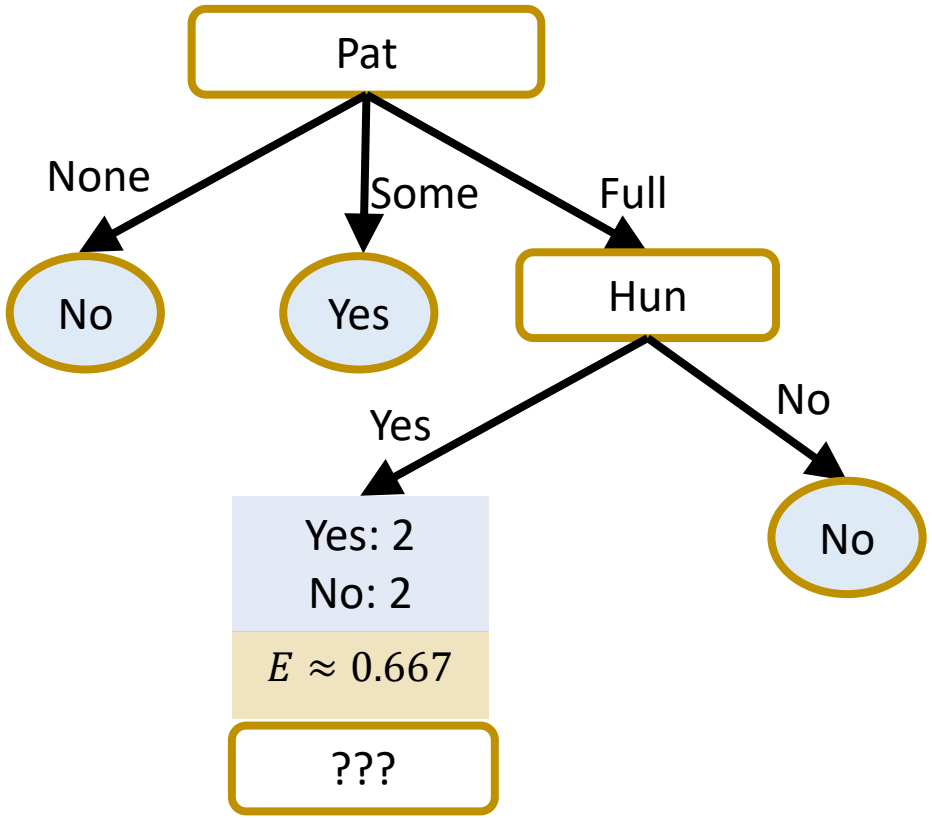

**Key:**

**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? |
| $x_2$ | 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

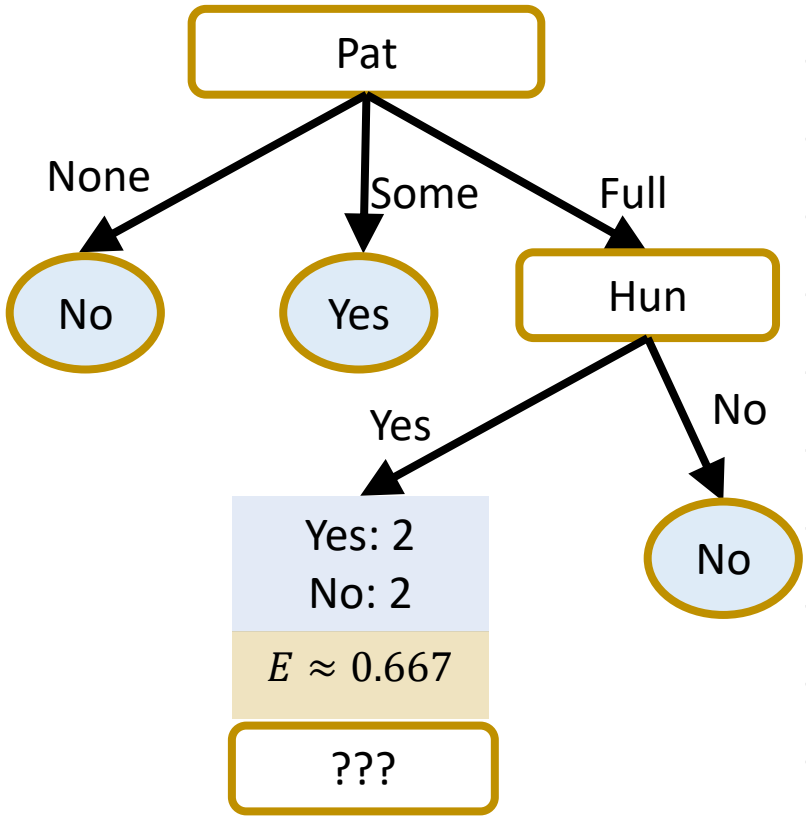
# Example

## 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 | 0.000 |
| Pat | -0.333 |
| Price | -0.022 |
| Rain | -0.022 |
| Res | -0.022 |
| Type | 0.167 |
| Est | -0.333 |

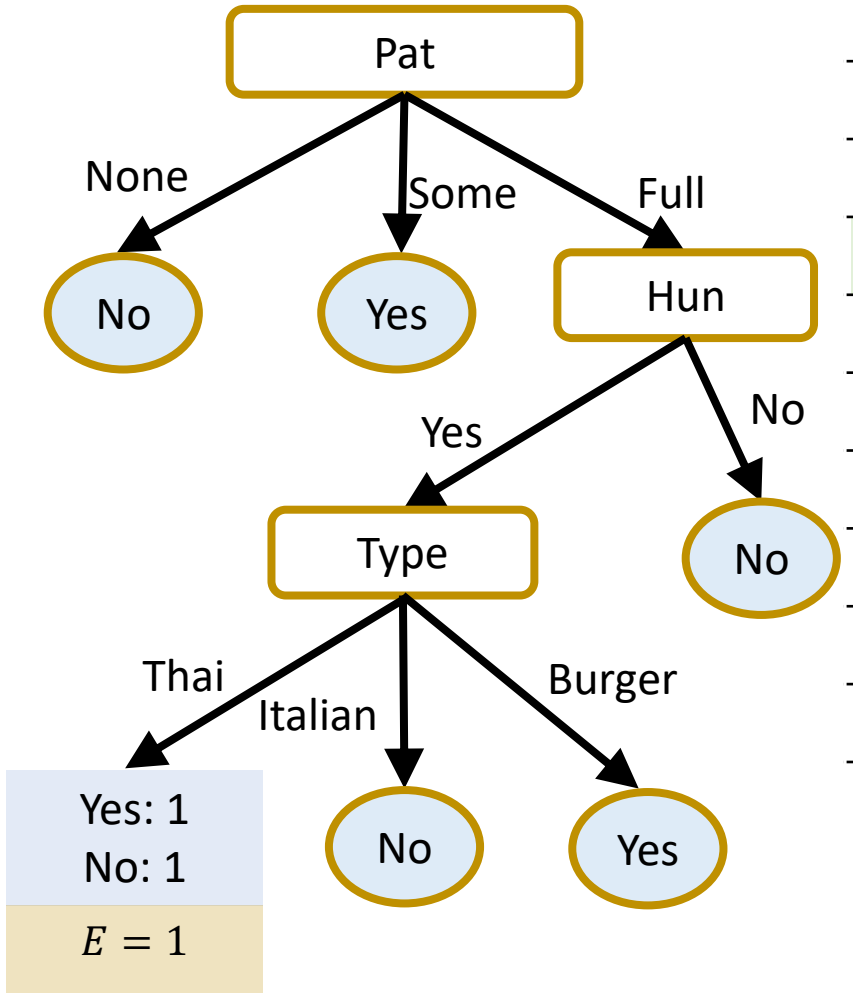
# Example

## 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 |

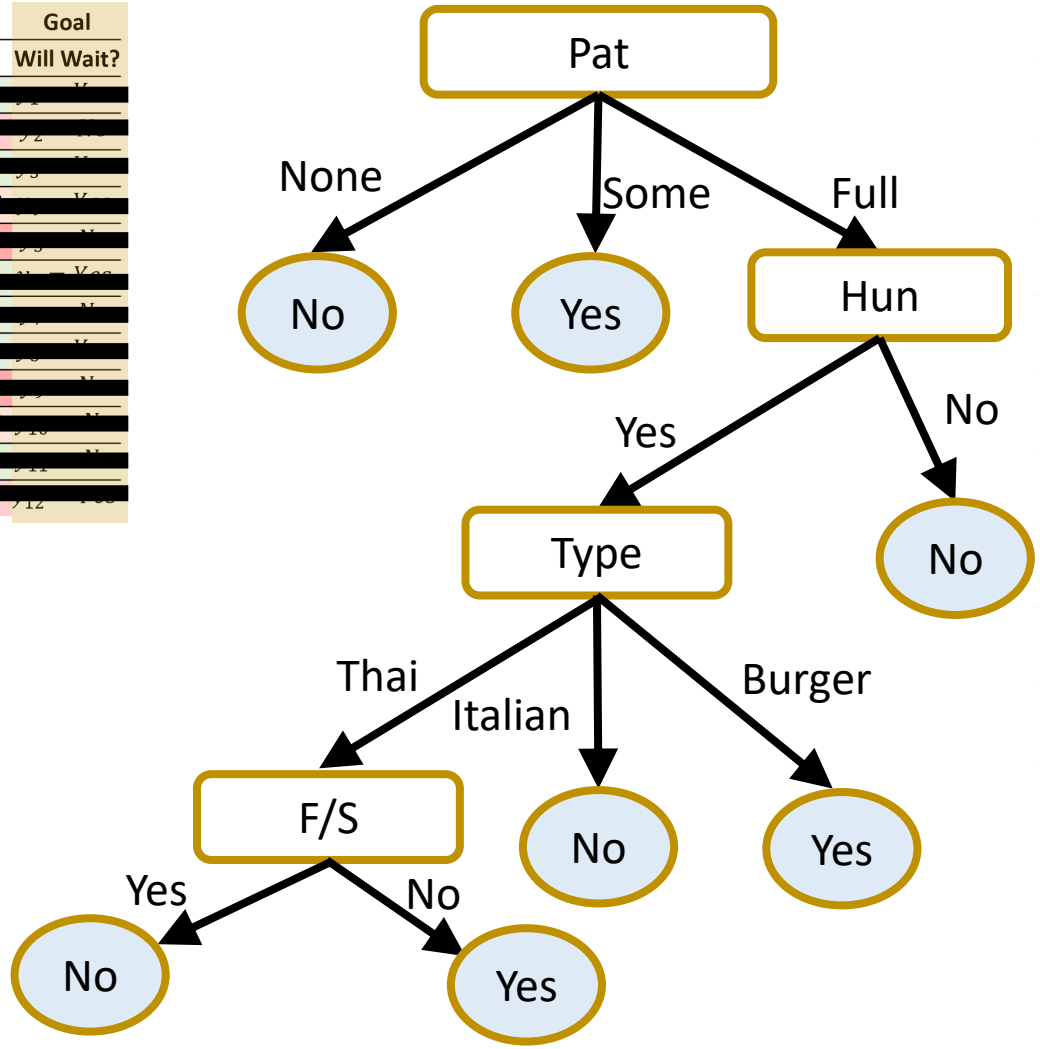
# Example

## 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? |
| x1 | Yes | No | No | Yes | Some | $$$ | No | Yes | French | 0-10 | y1 = Yes |
| x2 | Yes | No | No | Yes | Full | $ | No | No | Thai | 30-60 | y2 = No |
| x3 | No | Yes | No | No | Some | $ | No | No | Burger | 0-10 | y3 = Yes |
| x4 | Yes | No | Yes | Yes | Full | $ | Yes | No | Thai | 10-30 | y4 = Yes |
| x5 | Yes | No | No | Yes | Full | $$$ | No | Yes | French | >60 | y5 = No |
| x6 | No | Yes | No | Yes | Some | $$ | Yes | Yes | Italian | 0-10 | y6 = Yes |
| x7 | No | Yes | No | No | None | $ | Yes | No | Burger | 0-10 | y7 = No |
| x8 | No | No | No | Yes | Some | $$ | Yes | Yes | Thai | 0-10 | y8 = Yes |
| x9 | No | Yes | Yes | No | Full | $ | Yes | No | Burger | >60 | y9 = No |
| x10 | Yes | Yes | Yes | Yes | Full | $$$ | No | Yes | Italian | 10-30 | y10 = No |
| x11 | No | No | No | No | None | $ | No | No | Thai | 0-10 | y11 = No |
| x12 | Yes | Yes | Yes | Yes | Full | $ | No | No | Burger | 30-60 | y12 = 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 |

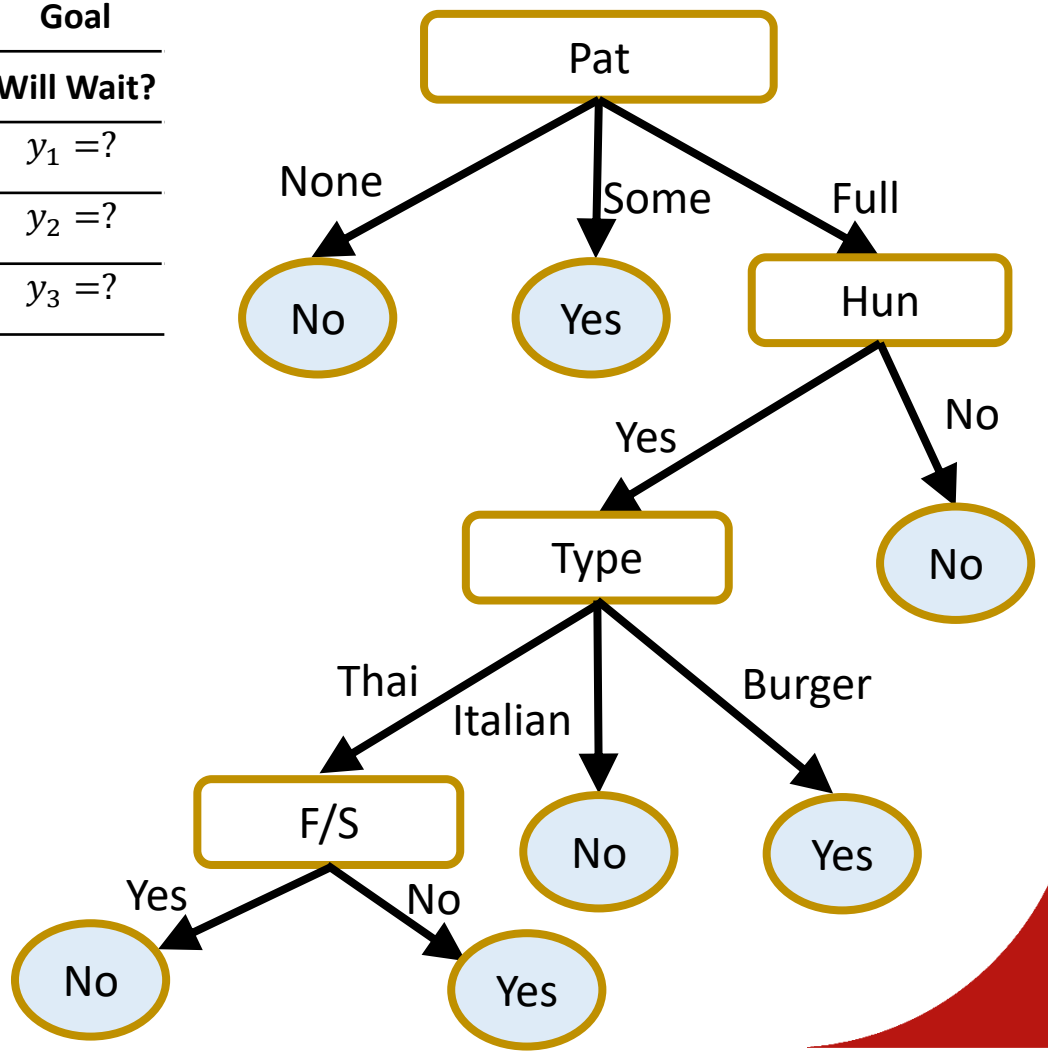
# Example

## Testing Data

| Example | Input Data | | | | | | | | | | Goal |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | **Alt** | **Bar** | **F/S** | **Hun** | **Pat** | **Price** | **Rain** | **Res** | **Type** | **Est** | **Will Wait?** |
| $s_1$ | Yes | No | Yes | No | Some | Cheap | Yes | Yes | French | 30-60 | $y_1 = ?$ |
| $s_2$ | No | Yes | No | Yes | Full | Mid-range | Yes | No | Thai | 0-10 | $y_2 = ?$ |
| $s_3$ | Yes | Yes | Yes | Yes | None | Expensive | No | Yes | Italian | >60 | $y_3 = ?$ |

**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

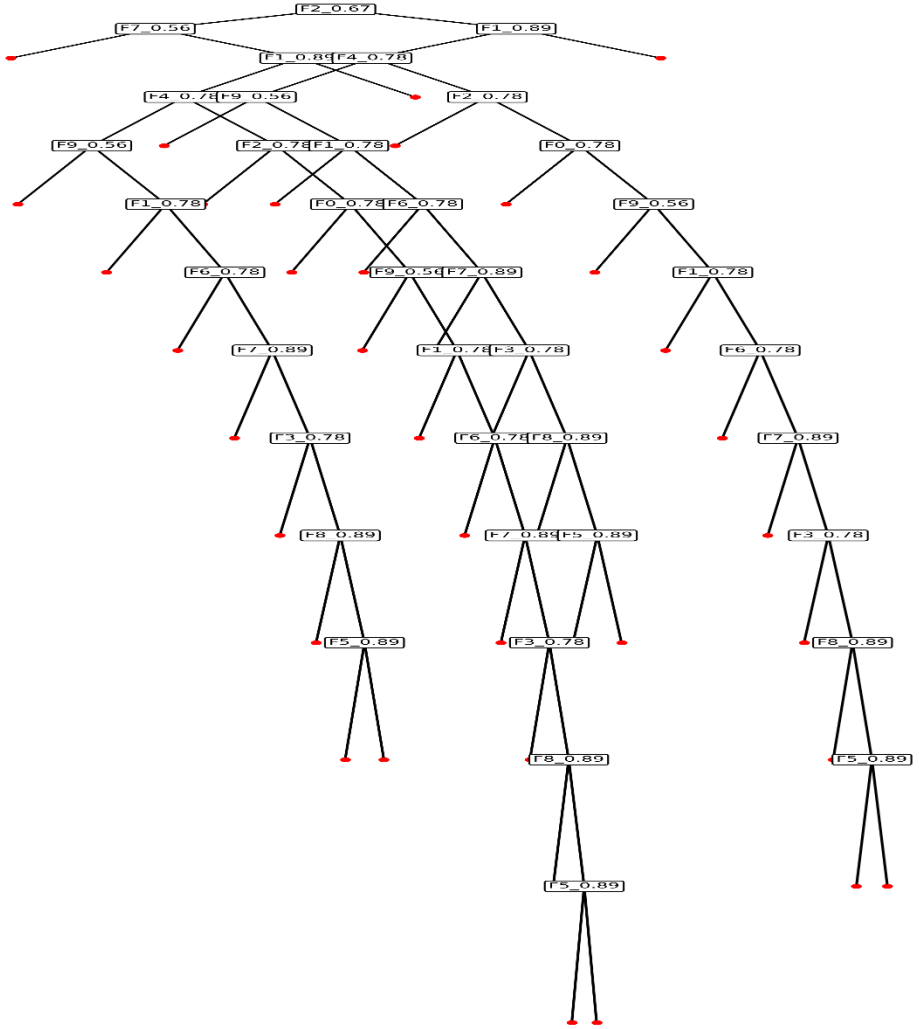


# Example 2 – Cancer Diagnosis

| laterality | view_position | height | width | breast_birads | breast_density | finding_categories | finding_birads |
|---|---|---|---|---|---|---|---|
| R | CC | 3518 | 2800 | BI-RADS 4 | DENSITY C | ['Mass'] | BI-RADS 4 |
| R | MLO | 3518 | 2800 | BI-RADS 4 | DENSITY C | ['Mass'] | BI-RADS 4 |
| R | CC | 3518 | 2800 | BI-RADS 3 | DENSITY C | ['Global Asymmetry'] | BI-RADS 3 |
| R | MLO | 3518 | 2800 | BI-RADS 3 | DENSITY C | ['Global Asymmetry'] | BI-RADS 3 |
| R | CC | 3518 | 2800 | BI-RADS 4 | DENSITY C | ['Architectural Distortion'] | BI-RADS 4 |
| R | MLO | 3518 | 2800 | BI-RADS 4 | DENSITY C | ['Architectural Distortion'] | BI-RADS 4 |
| L | CC | 3518 | 2800 | BI-RADS 3 | DENSITY C | ['Mass'] | BI-RADS 3 |
| L | MLO | 3518 | 2800 | BI-RADS 3 | DENSITY C | ['Mass'] | BI-RADS 3 |
| L | CC | 3518 | 2800 | BI-RADS 4 | DENSITY C | ['Nipple Retraction', 'Mass'] | BI-RADS 4 |
| L | MLO | 3518 | 2800 | BI-RADS 4 | DENSITY C | ['Nipple Retraction', 'Mass'] | BI-RADS 4 |
| R | CC | 3518 | 2800 | BI-RADS 3 | DENSITY C | ['Mass'] | BI-RADS 3 |
| R | MLO | 3518 | 2800 | BI-RADS 3 | DENSITY C | ['Mass'] | BI-RADS 3 |
| L | CC | 3518 | 2800 | BI-RADS 3 | DENSITY C | ['Mass'] | BI-RADS 3 |
| L | MLO | 3518 | 2800 | BI-RADS 3 | DENSITY C | ['Mass'] | BI-RADS 3 |
| L | CC | 3518 | 2800 | BI-RADS 5 | DENSITY C | ['Architectural Distortion'] | BI-RADS 5 |
| L | CC | 3518 | 2800 | BI-RADS 5 | DENSITY C | ['Mass'] | BI-RADS 4 |
| L | MLO | 3518 | 2800 | BI-RADS 5 | DENSITY C | ['Architectural Distortion'] | BI-RADS 5 |
| L | MLO | 3518 | 2800 | BI-RADS 5 | DENSITY C | ['Mass'] | BI-RADS 4 |

| BI-RADS | Feature = 0 | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 13406 | 13406 | 13406 | 13406 | 13406 | 13406 | 0 | 13406 | 13406 | 13406 | 13406 |
| 2 | 4676 | 4676 | 4676 | 4676 | 4676 | 4672 | 13 | 4671 | 4672 | 4676 | 4675 |
| 3 | 952 | 738 | 840 | 952 | 442 | 972 | 846 | 972 | 966 | 906 | 970 |
| 4 | 926 | 872 | 887 | 1001 | 524 | 993 | 971 | 1004 | 974 | 726 | 978 |
| 5 | 407 | 402 | 408 | 425 | 212 | 406 | 425 | 415 | 411 | 229 | 400 |
| | Feature = 1 | | | | | | | | | | |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 13406 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0 | 0 | 0 | 0 | 4 | 4663 | 5 | 4 | 0 | 1 |
| 3 | 20 | 234 | 132 | 20 | 530 | 0 | 126 | 0 | 6 | 66 | 2 |
| 4 | 79 | 133 | 118 | 4 | 481 | 12 | 34 | 1 | 31 | 279 | 27 |
| 5 | 20 | 25 | 19 | 2 | 215 | 21 | 2 | 12 | 16 | 198 | 27 |

# Example 3 – Forensic Identification



Lancaster University

# SCC361: Artificial Intelligence

## Week 5: Classification

Random Forest and Naïve Bayes

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**

# Naïve Bayes

# Naïve Bayes Classifier

## Probabilistic Classification

## Assumption

- Independence of features / attributes => "naïve"

- E.g.: A fruit may be classed as an *apple* if its **colour** is *red*, its **shape** is *round*, and its **size** is *approx. 3 inches in diameter*.

Lancaster University

# Naïve Bayes Classifier

**Classification Problem:**

- Given training data containing attributes $(A_1, \dots, A_n)$ and associated classes $C$
- For sample data, predict the class $C$ given attributes $(A_1, \dots, A_n)$

**Key Assumption:**

- Each attribute is independent

**Aim:**

- Learn from training data
- Estimate $P(C|A_1, \dots, A_n)$ directly from the data for each class
- Find the class $C$ that maximises

$$P(C|A_1, \dots, A_n)$$

# Naïve Bayes Classifier

## Example

| Sample ID | Attributes | | | | | | | | | | Labels |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Example | $A_1$ | $A_2$ | $A_3$ | $A_4$ | $A_5$ | $A_6$ | $A_7$ | $A_8$ | $A_9$ | $A_{10}$ | $C$ |
| | 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 | $y_{12} = Yes$ |

## Example Testing Data

| Example | $A_1$ | $A_2$ | $A_3$ | $A_4$ | $A_5$ | $A_6$ | $A_7$ | $A_8$ | $A_9$ | $A_{10}$ | $C$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | Alt | Bar | F/S | Hun | Pat | Price | Rain | Res | Type | Est | Will Wait? |
| $s_1$ | Yes | No | Yes | No | Some | $ | Yes | Yes | French | 30-60 | $y_1 =?$ |
| $s_2$ | No | Yes | No | Yes | Full | $$ | Yes | No | Thai | 0-10 | $y_2 =?$ |
| $s_3$ | Yes | Yes | Yes | Yes | None | $$$ | No | Yes | Italian | >60 | $y_3 =?$ |

Lancaster University

# Naïve Bayes Classifier

## Example Training Data

| 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 | $y_{12} = Yes$ |

## Example Testing Data

| Example | Input Data | | | | | | | | | | Goal |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | Alt | Bar | F/S | Hun | Pat | Price | Rain | Res | Type | Est | Will Wait? |
| $s_1$ | Yes | No | Yes | No | Some | $ | Yes | Yes | French | 30-60 | $y_1 = ?$ |
| $s_2$ | No | Yes | No | Yes | Full | $$ | Yes | No | Thai | 0-10 | $y_2 = ?$ |
| $s_3$ | Yes | Yes | Yes | Yes | None | $$$ | No | Yes | Italian | >60 | $y_3 = ?$ |

**Naïve Bayes:**

Can we estimate

$$P(C|A\_1, \dots, A\_n)$$

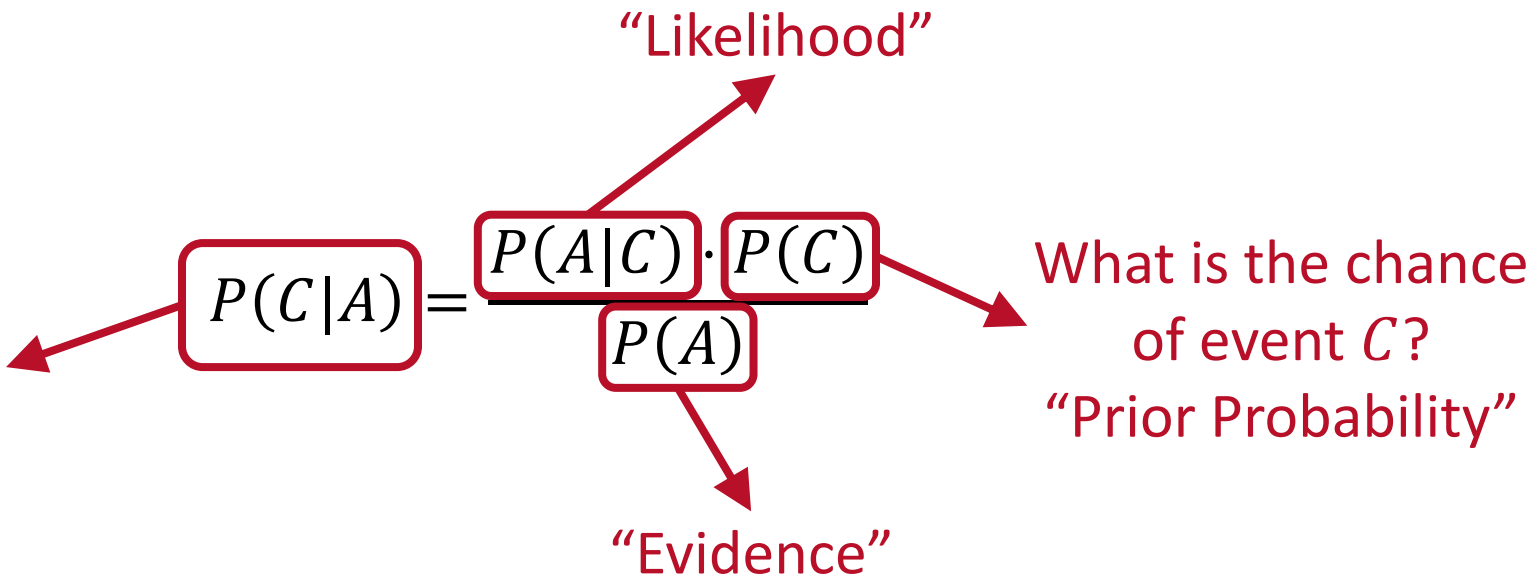
directly for the testing data using the training data?

# Bayes Theorem

Bayes Theorem:

Let $A = \{A_1, \ldots, A_n\}$ be a set of $n$ attributes.

Let $C$ denote the associated class.



Lancaster University

# Example

We want to know:

*If a patient has a fever, what is the probability that the patient has a cold?* $P(C|F)?$

Given:

- A doctor knows that a **cold** causes a **fever** 50% of the time $P(F|C) = 0.5$
- Prior probability of any patient having a cold is $\frac{1}{50000}$ $P(C) = \frac{1}{50000}$
- Prior probability of any patient having a fever is $\frac{1}{20}$ $P(F) = \frac{1}{20}$

We can calculate:

$$P(C|F) = \frac{P(F|C) \cdot P(C)}{P(F)} = \frac{0.5 \cdot \frac{1}{50000}}{\frac{1}{20}} = \frac{\frac{1}{100000}}{\frac{1}{20}} = \frac{20}{100000} = 0.0002$$

Lancaster University

# Naïve Bayes Classifier

**Classification Problem:**

- Predict the class $C$ given attributes / features $(A_1, \dots, A_n)$
- Each attribute is independent
- Find the class $C$ that maximises

$$P(C|A_1, \dots, A_n)$$

**Aim:**

- Estimate $P(C|A_1, \dots, A_n)$ directly from the data

# Naïve Bayes Classifier

Remember:

We're looking for the class $C_*$ that maximises the probability:

$$C_* = \underset{c}{\operatorname{argmax}}\, P(C|A)$$

I.e. for all $i$:

$$\left[P(C_*|A) = \frac{P(A|C_*) \cdot P(C_*)}{P(A)}\right] \geq \left[P(C_i|A) = \frac{P(A|C_i) \cdot P(C_i)}{P(A)}\right]$$

Fixed for all $i$

# Naïve Bayes Classifier

Remember:

We're looking for the class $C_*$ that maximises the probability:

$$C_* = \underset{c}{\operatorname{argmax}}\, P(C|A)$$

I.e. for all $i$:

$$\left[P(C_*|A) = \frac{P(A|C_*) \cdot P(C_*)}{\cancel{P(A)}}\right] \geq \left[P(C_i|A) = \frac{P(A|C_i) \cdot P(C_i)}{\cancel{P(A)}}\right]$$

So, we actually compare

$$P(A|C_i) \cdot P(C_i)$$

# Naïve Bayes Classifier – Multiple Attributes

So, we actually compare

$$P(A|C_i) \cdot P(C_i)$$

**What if we have multiple attributes?**

If $A = \{A_1, \dots A_n\}$, how do we compute $P(A|C_i)$?

I.e. what is

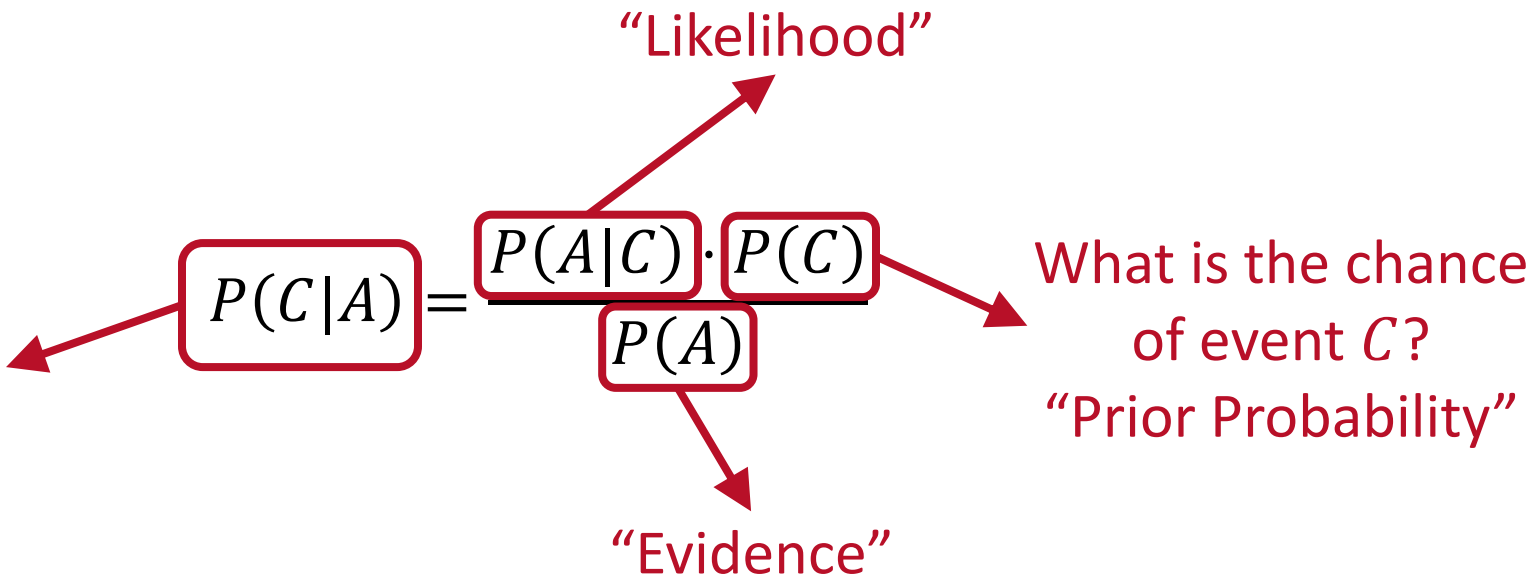
$$P(A_1, \dots, A_n \,|C)$$

# Bayes Theorem

Bayes Theorem:

Let $A = \{A_1, \ldots, A_n\}$ be a set of $n$ attributes.

Let $C$ denote the associated class.



# Bayes Theorem

Lancaster University

Some Rules:

If $A_1$ and $A_2$ are independent, i.e. each event is not affected by the other:

$$P(A_1|A_2) = P(A_1)$$

$$P(A_2|A_1) = P(A_2)$$

What is $P(C|A)$ if we can't assume independence?

Combination (Product Rule):

$$P(A_1, A_2) = P(A_1|A_2) \cdot P(A_2)$$
$$= P(A_2|A_1) \cdot P(A_1)$$

Combination of **independent** random variables:

$$P(A_1, A_2) = P(A_1|A_2) \cdot P(A_2)$$
$$= P(A_1) \cdot P(A_2)$$

$$P(A_1, \dots, A_n) = P(A_1) \cdot \dots \cdot P(A_n)$$

Bayes Theorem:

What is the chance of event $C$, given that event $A$ has happened?

$$P(C|A) = \frac{P(A|C) \cdot P(C)}{P(A)}$$

What is the chance of event $C$?

# Bayes Theorem

What is $P(C|A)$ if we can't assume independence?

Conditional probability:

$$P(C|A) = \frac{P(C,A)}{P(A)}$$

$$P(C|A) = \frac{P(A|C) \cdot P(C)}{P(A)}$$

Multiple Variables:

$$P(C|A_1, \dots, A_n) = \frac{P(A_1, \dots, A_n|C) \cdot P(C)}{P(A_1, \dots, A_n)}$$

$$= \frac{P(A_1, \dots, A_n|C) \cdot P(C)}{P(A_1) \cdot \dots \cdot P(A_n)}$$

$A_1, A_2$ independent:

$$P(A_1|A_2) = P(A_1)$$

$$P(A_2|A_1) = P(A_2)$$

Combination (Product Rule):

$$P(A_1, A_2) = P(A_2|A_1) \cdot P(A_1)$$

Combination of **independent** random variables:

$$P(A_1, \dots, A_n) = P(A_1) \cdot \dots \cdot P(A_n)$$

Bayes Theorem:

What is the chance of event $C$, given that event $A$ has happened?

$$P(C|A) = \frac{P(A|C) \cdot P(C)}{P(A)}$$

What is the chance of event $C$?

# Naïve Bayes Classifier

Remember:

$$\prod_{i=1}^{n} X_i = X_1 \cdot \ldots \cdot X_n$$

1. We assume $A_1, \ldots A_n$ are independent of each other
2. But $C$ is dependent on each of $A_1, \ldots A_n$

i.e.

$$P(A_1|A_2) = P(A_1) \quad \text{but} \quad P(A|C) \neq P(A)$$

Since $A_1, \ldots A_n$ are independent:

$$P(A_1, \ldots, A_n|C) = P(A_1|C) \cdot \ldots \cdot P(A_n|C) = \prod_{i=1}^{n} P(A_i|C)$$

# Naïve Bayes Classifier

So, for one attribute $A$, we compare:

$$P(A|C_i) \cdot P(C_i)$$

If we have $n$ attributes $A = \{A_1, \dots A_n\}$, we compare:

$$P(A_1, \dots, A_n \,|C_i) \cdot P(C_i)$$

$$= P(A_1|C_i) \cdot \dots \cdot P(A_n|C_i) \cdot P(C_i)$$

$$= \left( \prod_j^n P(A_j|C_i) \right) \cdot P(C_i)$$

# Naïve Bayes Classifier

**Approach:**

- Compute the posterior probability $P(C|A_1, \dots, A_n)$ for all values of C using Bayes Theorem

Likelihood

Prior

Posterior

$$P(C|A_1, \dots, A_n) = \frac{P(A_1, \dots, A_n|C) \cdot P(C)}{P(A_1, \dots, A_n)}$$

Evidence / normaliser

**Example:**

- Given an image with attributes / features $A_1, \dots, A_n$, classify the image as “car” or “bike”:

$$P(C = "car"|A_1, \dots, A_n) = \frac{P(A_1, \dots, A_n|C = "car")P(C = "car")}{P(A_1, \dots, A_n)}$$

$$P(C = "bike"|A_1, \dots, A_n) = \frac{P(A_1, \dots, A_n|C = "bike")P(C = "bike")}{P(A_1, \dots, A_n)}$$

# Naïve Bayes Classifier

**Approach:**

- Compute the posterior probability $P(C|A_1, \dots, A_n)$ for all values of C using Bayes Theorem

Likelihood

Prior

Posterior

$$P(C|A_1, \dots, A_n) = \frac{P(A_1, \dots, A_n|C) \cdot P(C)}{P(A_1, \dots, A_n)}$$

Evidence / normaliser

**Example:**

- Given an image with attributes / features $A_1, \dots, A_n$, classify the image as “car” or “bike”:

$$P(C = \text{"car"}|A_1, \dots, A_n) > P(C = \text{"bike"}|A_1, \dots, A_n)?$$

$$\frac{P(A_1, \dots, A_n|C = \text{"car"})P(C = \text{"car"})}{\cancel{P(A_1, \dots, A_n)}} > \frac{P(A_1, \dots, A_n|C = \text{"bike"})P(C = \text{"bike"})}{\cancel{P(A_1, \dots, A_n)}}?$$

# Naïve Bayes Classifier

**Modified Approach:**

- For each class $C$, compute the product likelihood and the prior probabilities

$$P(C|A_1, \dots, A_n) = \frac{P(A_1, \dots, A_n|C) \cdot P(C)}{\cancel{P(A_1, \dots, A_n)}}$$

- Find the class $C$ which maximises this.
- This is equivalent to finding the class $C$ that maximises the posterior probability.

Lancaster University

# Example

Features:

$$A_1: \text{ Refund},$$
$$A_2: \text{ Marital Status},$$
$$A_3: \text{ Taxable Income}$$

Class: $C = \text{Evade}$

Sample $x$:

$$A_1 = No,$$
$$A_2 = Married,$$
$$A_3 = 120$$

Aim: calculate probability of evading (or not) given the features

| ID | Refund ($A_1$) | Marital Status ($A_2$) | Taxable Income (£k) ($A_3$) | Evade ($C$) |
|---|---|---|---|---|
| 1 | Yes | Single | 125 | No |
| 2 | No | Married | 100 | No |
| 3 | No | Single | 70 | No |
| 4 | Yes | Married | 120 | No |
| 5 | No | Divorced | 95 | Yes |
| 6 | No | Married | 60 | No |
| 7 | Yes | Divorced | 220 | No |
| 8 | No | Single | 85 | Yes |
| 9 | No | Married | 75 | No |
| 10 | No | Single | 90 | Yes |

# Example

Sample: $x = (A_1 = No, A_2 = Married, A_3 = 120)$

For brevity: let $C^{Yes}$ denote "$C = yes$"

Probability of $x$ belonging class $C^{Yes}$:

$$P\left(C^{Yes} | A_1^{No}, A_2^{Married}, A_3^{120}\right) = \frac{P\left(A_1^{No}, A_2^{Married}, A_3^{120} | C^{Yes}\right) \cdot P(C^{Yes})}{P\left(A_1^{No}, A_2^{Married}, A_3^{120}\right)}$$

Probability of $x$ belonging class $C^{No}$:

$$P\left(C^{Yes} | A_1^{No}, A_2^{Married}, A_3^{120}\right) = \frac{P\left(A_1^{No}, A_2^{Married}, A_3^{120} | C^{No}\right) \cdot P(C^{No})}{P\left(A_1^{No}, A_2^{Married}, A_3^{120}\right)}$$

| ID | Refund ($A_1$) | Marital Status ($A_2$) | Taxable Income (£k) ($A_3$) | Evade ($C$) |
|---|---|---|---|---|
| 1 | Yes | Single | 125 | No |
| 2 | No | Married | 100 | No |
| 3 | No | Single | 70 | No |
| 4 | Yes | Married | 120 | No |
| 5 | No | Divorced | 95 | Yes |
| 6 | No | Married | 60 | No |
| 7 | Yes | Divorced | 220 | No |
| 8 | No | Single | 85 | Yes |
| 9 | No | Married | 75 | No |
| 10 | No | Single | 90 | Yes |

# Example

Probability of x belonging class $C^{Yes}$:

$$P\left(C^{Yes} | A_1^{No}, A_2^{Married}, A_3^{120}\right)$$

$$= \frac{P\left(A_1^{No}, A_2^{Married}, A_3^{120} | C^{Yes}\right) \cdot P(C^{Yes})}{P\left(A_1^{No}, A_2^{Married}, A_3^{120}\right)}$$

$$= \frac{P(A_1^{No} | C^{Yes}) \cdot P\left(A_2^{Married} | C^{Yes}\right) \cdot P(A_3^{120} | C^{Yes}) \cdot P(C^{Yes})}{P\left(A_1^{No}, A_2^{Married}, A_3^{120}\right)}$$

| ID | Refund ($A_1$) | Marital Status ($A_2$) | Taxable Income (£k) ($A_3$) | Evade ($C$) |
|---|---|---|---|---|
| 1 | Yes | Single | 125 | No |
| 2 | No | Married | 100 | No |
| 3 | No | Single | 70 | No |
| 4 | Yes | Married | 120 | No |
| 5 | No | Divorced | 95 | Yes |
| 6 | No | Married | 60 | No |
| 7 | Yes | Divorced | 220 | No |
| 8 | No | Single | 85 | Yes |
| 9 | No | Married | 75 | No |
| 10 | No | Single | 90 | Yes |

# Example

$$\frac{P(A_1^{No}|C^{Yes}) \cdot P\left(A_2^{Married}|C^{Yes}\right) \cdot P(A_3^{120}|C^{Yes}) \cdot P(C^{Yes})}{\cancel{P(A_1^{No}) \cdot P\left(A_2^{Married}\right) \cdot P(A_3^{120})}}$$

Prior Probability:

$$P(C^{Yes}) = \frac{3}{10}$$

| ID | Refund ($A_1$) | Marital Status ($A_2$) | Taxable Income (£k) ($A_3$) | Evade ($C$) |
|---|---|---|---|---|
| 1 | Yes | Single | 125 | No |
| 2 | No | Married | 100 | No |
| 3 | No | Single | 70 | No |
| 4 | Yes | Married | 120 | No |
| 5 | No | Divorced | 95 | Yes |
| 6 | No | Married | 60 | No |
| 7 | Yes | Divorced | 220 | No |
| 8 | No | Single | 85 | Yes |
| 9 | No | Married | 75 | No |
| 10 | No | Single | 90 | Yes |

# Example

$$\frac{P(A_1^{No}|C^{Yes}) \cdot P\left(A_2^{Married}|C^{Yes}\right) \cdot P(A_3^{120}|C^{Yes}) \cdot P(C^{Yes})}{\cancel{P(A_1^{No}) \cdot P\left(A_2^{Married}\right) \cdot P(A_3^{120})}}$$

Conditional Probabilities:

$$P(A_1^{No}|C^{Yes}) = \frac{3}{3} = 1 \qquad P\left(A_2^{Married}|C^{Yes}\right) = \frac{0}{3} = 0$$

| ID | Refund ($A_1$) | Marital Status ($A_2$) | Taxable Income (£k) ($A_3$) | Evade ($C$) |
|---|---|---|---|---|
| 1 | Yes | Single | 125 | No |
| 2 | No | Married | 100 | No |
| 3 | No | Single | 70 | No |
| 4 | Yes | Married | 120 | No |
| 5 | No | Divorced | 95 | Yes |
| 6 | No | Married | 60 | No |
| 7 | Yes | Divorced | 220 | No |
| 8 | No | Single | 85 | Yes |
| 9 | No | Married | 75 | No |
| 10 | No | Single | 90 | Yes |

# Example

$$\frac{P(A_1^{No}|C^{No}) \cdot P\left(A_2^{Married}|C^{No}\right) \cdot P(A_3^{120}|C^{No}) \cdot P(C^{No})}{\cancel{P(A_1^{No}) \cdot P\left(A_2^{Married}\right) \cdot P(A_3^{120})}}$$

Prior Probability:

$$P(C^{No}) = \frac{7}{10}$$

| ID | Refund ($A_1$) | Marital Status ($A_2$) | Taxable Income (£k) ($A_3$) | Evade ($C$) |
|---|---|---|---|---|
| 1 | Yes | Single | 125 | No |
| 2 | No | Married | 100 | No |
| 3 | No | Single | 70 | No |
| 4 | Yes | Married | 120 | No |
| 5 | No | Divorced | 95 | Yes |
| 6 | No | Married | 60 | No |
| 7 | Yes | Divorced | 220 | No |
| 8 | No | Single | 85 | Yes |
| 9 | No | Married | 75 | No |
| 10 | No | Single | 90 | Yes |

# Example

$$\frac{P(A_1^{No}|C^{Yes}) \cdot P\left(A_2^{Married}|C^{Yes}\right) \cdot P(A_3^{120}|C^{Yes}) \cdot P(C^{Yes})}{\cancel{P(A_1^{No}) \cdot P\left(A_2^{Married}\right) \cdot P(A_3^{120})}}$$

Conditional Probabilities:

$$P(A_1^{No}|C^{No}) = \frac{4}{7} \qquad P\left(A_2^{Married}|C^{No}\right) = \frac{4}{7}$$

| ID | Refund ($A_1$) | Marital Status ($A_2$) | Taxable Income (£k) ($A_3$) | Evade ($C$) |
|---|---|---|---|---|
| 1 | Yes | Single | 125 | No |
| 2 | No | Married | 100 | No |
| 3 | No | Single | 70 | No |
| 4 | Yes | Married | 120 | No |
| 5 | No | Divorced | 95 | Yes |
| 6 | No | Married | 60 | No |
| 7 | Yes | Divorced | 220 | No |
| 8 | No | Single | 85 | Yes |
| 9 | No | Married | 75 | No |
| 10 | No | Single | 90 | Yes |

Lancaster University

# Example

$$\frac{P(A_1^{No}|C) \cdot P\left(A_2^{Married}|C\right) \cdot P(A_3^{120}|C) \cdot P(C)}{\cancel{P(A_1^{No}) \cdot P\left(A_2^{Married}\right) \cdot P(A_3^{120})}}$$

| ID | Refund ($A_1$) | Marital Status ($A_2$) | Taxable Income (£k) ($A_3$) | Evade ($C$) |
|---|---|---|---|---|
| 1 | Yes | Single | 125 | No |
| 2 | No | Married | 100 | No |
| 3 | No | Single | 70 | No |
| 4 | Yes | Married | 120 | No |
| 5 | No | Divorced | 95 | Yes |
| 6 | No | Married | 60 | No |
| 7 | Yes | Divorced | 220 | No |
| 8 | No | Single | 85 | Yes |
| 9 | No | Married | 75 | No |
| 10 | No | Single | 90 | Yes |

Prior Probability:

$$P(C^{Yes}) = \frac{3}{10} \qquad P(C^{No}) = \frac{7}{10}$$

Conditional Probabilities:

$$P(A_1^{No}|C^{Yes}) = \frac{3}{3} = 1 \qquad P(A_1^{No}|C^{No}) = \frac{4}{7}$$

$$P\left(A_2^{Married}|C^{Yes}\right) = \frac{0}{3} = 0 \qquad P\left(A_2^{Married}|C^{No}\right) = \frac{3}{7}$$

Lancaster University

# Estimate Probability from Continuous Data

## Discretise the range into bins

- One ordinal attribute per bin
- Violates independence assumption

## Binary split ($A < v$) or ($A > v$)

- Choose only one of the two splits as new attribute

## Probability Density Estimation

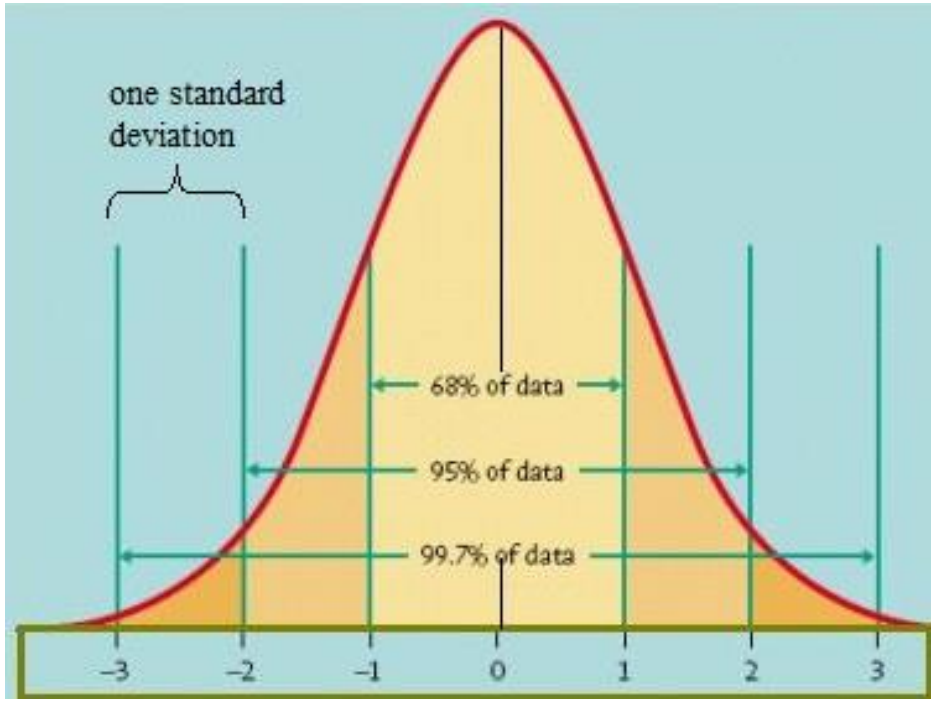
- Assume attribute follows a normal distribution
- Use data to estimate distribution parameters
- Once distribution known, estimate conditional probability $P(A_i|C)$

# Probability Density Estimation (Normal Distribution)

- The term "*bell curve*" is usually used in the social sciences;
  - in statistics, it's called a **Normal distribution** and
  - in physics, it's called a **Gaussian distribution**.
- Many phenomena have probability distributions that are bell curves, including:
  - Heights, Weights, Exam scores, etc.

**Characteristics of Bell Curves, Normal Curves:**

- The mean (average) is always in the centre
- There is only one peak
- It is symmetric about the mean. Half of data points are to the left of the mean and half are to the right of the mean.



# Probability Density Estimation (Normal Distribution)

Equation:

$$P(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

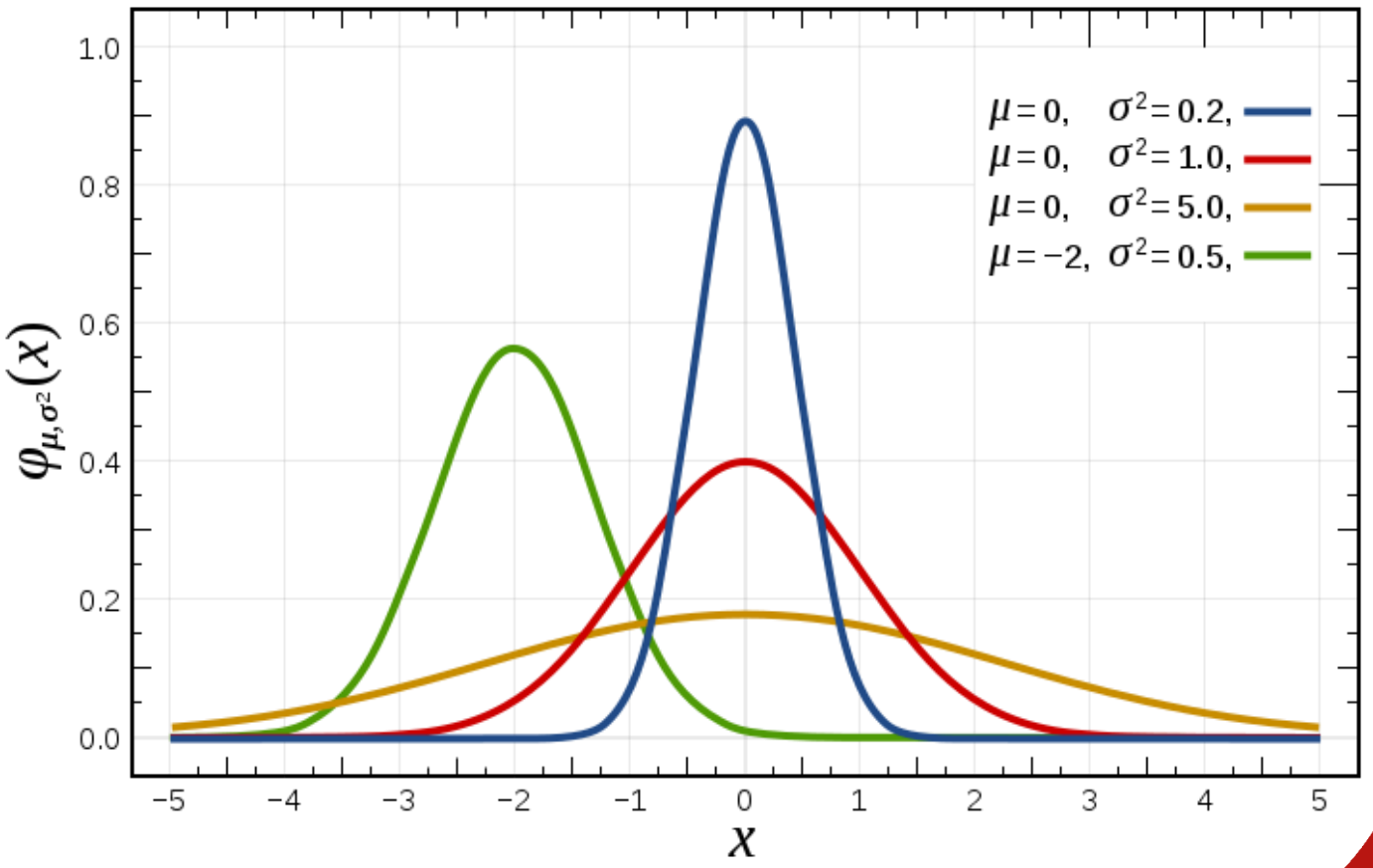
where

- $\sigma$: standard deviation
- $\sigma^2$: variance
- $\mu$: mean

Let $X = \{x_i\}$. Then

$$\mu = \frac{\sum_i x_i}{|X|}$$

$$\sigma = \sqrt{\frac{\sum_i (x_i - \mu)^2}{|X|}}$$



Lancaster University

# Back to the Example

$$\frac{P(A_1^{No}|C^{Yes}) \cdot P\left(A_2^{Married}|C^{Yes}\right) \cdot P(A_3^{120}|C^{Yes}) \cdot P(C^{Yes})}{\cancel{P(A_1^{No}) \cdot P\left(A_2^{Married}\right) \cdot P(A_3^{120})}}$$

Conditional Probability: $P(A_3^x|C^{Yes})$

- Sample mean: $\mu = \frac{95+85+90}{3} = 90$
- Sample variance: $\sigma^2 = \frac{(95-90)^2+(85-90)^2+(90-90)^2}{3} = \frac{50}{3}$

$$P(A_3^x|C^{Yes}) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} = \frac{\sqrt{3}}{\sqrt{100\pi}} e^{-\frac{3(x-90)^2}{100}}$$

| ID | Refund ($A_1$) | Marital Status ($A_2$) | Taxable Income (£k) ($A_3$) | Evade ($C$) |
|---|---|---|---|---|
| 1 | Yes | Single | 125 | No |
| 2 | No | Married | 100 | No |
| 3 | No | Single | 70 | No |
| 4 | Yes | Married | 120 | No |
| 5 | No | Divorced | 95 | Yes |
| 6 | No | Married | 60 | No |
| 7 | Yes | Divorced | 220 | No |
| 8 | No | Single | 85 | Yes |
| 9 | No | Married | 75 | No |
| 10 | No | Single | 90 | Yes |

Lancaster University

# Back to the Example

$$\frac{P(A_1^{No}|C^{Yes}) \cdot P\left(A_2^{Married}|C^{Yes}\right) \cdot P(A_3^{120}|C^{Yes}) \cdot P(C^{Yes})}{\cancel{P(A_1^{No}) \cdot P\left(A_2^{Married}\right) \cdot P(A_3^{120})}}$$

Conditional Probability: $P(A_3^x|C^{Yes}) = \frac{\sqrt{3}}{\sqrt{100\pi}} e^{-\frac{3(x-90)^2}{100}}$

Conditional Probability: $P(A_3^x|C^{No})$:

- Sample mean: $\mu = \frac{125+100+70+120+60+220+75}{7} = \frac{770}{7} = 110$
- Sample variance: $\sigma^2 = \frac{17850}{7} = 2550$

$$P(A_3^x|C^{No}) = \frac{1}{\sqrt{5100\pi}} e^{-\frac{(x-110)^2}{5100}}$$

| ID | Refund ($A_1$) | Marital Status ($A_2$) | Taxable Income (£k) ($A_3$) | Evade ($C$) |
|---|---|---|---|---|
| 1 | Yes | Single | 125 | No |
| 2 | No | Married | 100 | No |
| 3 | No | Single | 70 | No |
| 4 | Yes | Married | 120 | No |
| 5 | No | Divorced | 95 | Yes |
| 6 | No | Married | 60 | No |
| 7 | Yes | Divorced | 220 | No |
| 8 | No | Single | 85 | Yes |
| 9 | No | Married | 75 | No |
| 10 | No | Single | 90 | Yes |

# Back to the Example

$$\frac{P(A_1^{No}|C^{Yes}) \cdot P\left(A_2^{Married}|C^{Yes}\right) \cdot P(A_3^{120}|C^{Yes}) \cdot P(C^{Yes})}{\cancel{P(A_1^{No}) \cdot P\left(A_2^{Married}\right) \cdot P(A_3^{120})}}$$

Conditional Probability: $P(A_3^x|C^{Yes}) = \frac{\sqrt{3}}{\sqrt{100\pi}} e^{-\frac{3(x-90)^2}{100}}$

Conditional Probability: $P(A_3^x|C^{No}) = \frac{1}{\sqrt{5100\pi}} e^{-\frac{(x-110)^2}{5100}}$

For our example:

$$P(A_3^{120}|C^{Yes}) = \frac{\sqrt{3}}{\sqrt{100\pi}} e^{-\frac{3(120-90)^2}{100}} \approx 1.488 \times 10^{-9}$$

$$P(A_3^{120}|C^{No}) = \frac{1}{\sqrt{5100\pi}} e^{-\frac{(120-110)^2}{5100}} \approx 7.7 \times 10^{-3}$$

| ID | Refund ($A_1$) | Marital Status ($A_2$) | Taxable Income (£k) ($A_3$) | Evade ($C$) |
|---|---|---|---|---|
| 1 | Yes | Single | 125 | No |
| 2 | No | Married | 100 | No |
| 3 | No | Single | 70 | No |
| 4 | Yes | Married | 120 | No |
| 5 | No | Divorced | 95 | Yes |
| 6 | No | Married | 60 | No |
| 7 | Yes | Divorced | 220 | No |
| 8 | No | Single | 85 | Yes |
| 9 | No | Married | 75 | No |
| 10 | No | Single | 90 | Yes |

Lancaster University

# Back to the Example

$$\frac{P(A_1^{No}|C^{Yes}) \cdot P\left(A_2^{Married}|C^{Yes}\right) \cdot P(A_3^{120}|C^{Yes}) \cdot P(C^{Yes})}{\cancel{P(A_1^{No}) \cdot P\left(A_2^{Married}\right) \cdot P(A_3^{120})}}$$

$$\approx \frac{1 \cdot 0 \cdot (1.488 \times 10^{-9}) \cdot \frac{3}{10}}{\cancel{P(A_1^{No}) \cdot P\left(A_2^{Married}\right) \cdot P(A_3^{120})}} = 0$$

$$\frac{P(A_1^{No}|C^{No}) \cdot P\left(A_2^{Married}|C^{No}\right) \cdot P(A_3^{120}|C^{No}) \cdot P(C^{No})}{P(A_1^{No}) \cdot P\left(A_2^{Married}\right) \cdot P(A_3^{120})}$$

$$\approx \frac{\frac{4}{7} \cdot \frac{3}{7} \cdot (7.7 \times 10^{-3}) \cdot \frac{7}{10}}{P(A_1^{No}) \cdot P\left(A_2^{Married}\right) \cdot P(A_3^{120})} = 1.3 \times 10^{-3}$$

| ID | Refund ($A_1$) | Marital Status ($A_2$) | Taxable Income (£k) ($A_3$) | Evade ($C$) |
|---|---|---|---|---|
| 1 | Yes | Single | 125 | No |
| 2 | No | Married | 100 | No |
| 3 | No | Single | 70 | No |
| 4 | Yes | Married | 120 | No |
| 5 | No | Divorced | 95 | Yes |
| 6 | No | Married | 60 | No |
| 7 | Yes | Divorced | 220 | No |
| 8 | No | Single | 85 | Yes |
| 9 | No | Married | 75 | No |
| 10 | No | Single | 90 | Yes |

**No**

Lancaster University

# Naïve Bayes Summary

## Approach

- Compute the product likelihood and the prior probabilities for each class $C$
- Assign the class corresponding to the maximum posterior probability

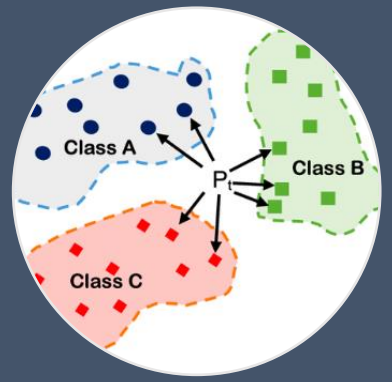
## Robust to isolated noise

## Can handle missing values

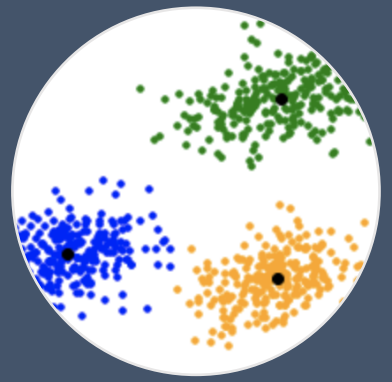
- Ignore the sample during probability estimation calculations

## Independence assumptions may not hold

# Next Four Weeks: Fundamental ML

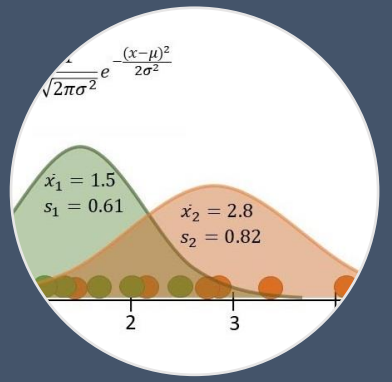



KNN



K-Means


Decision Trees




Naïve Bayes

Generic Algorithms

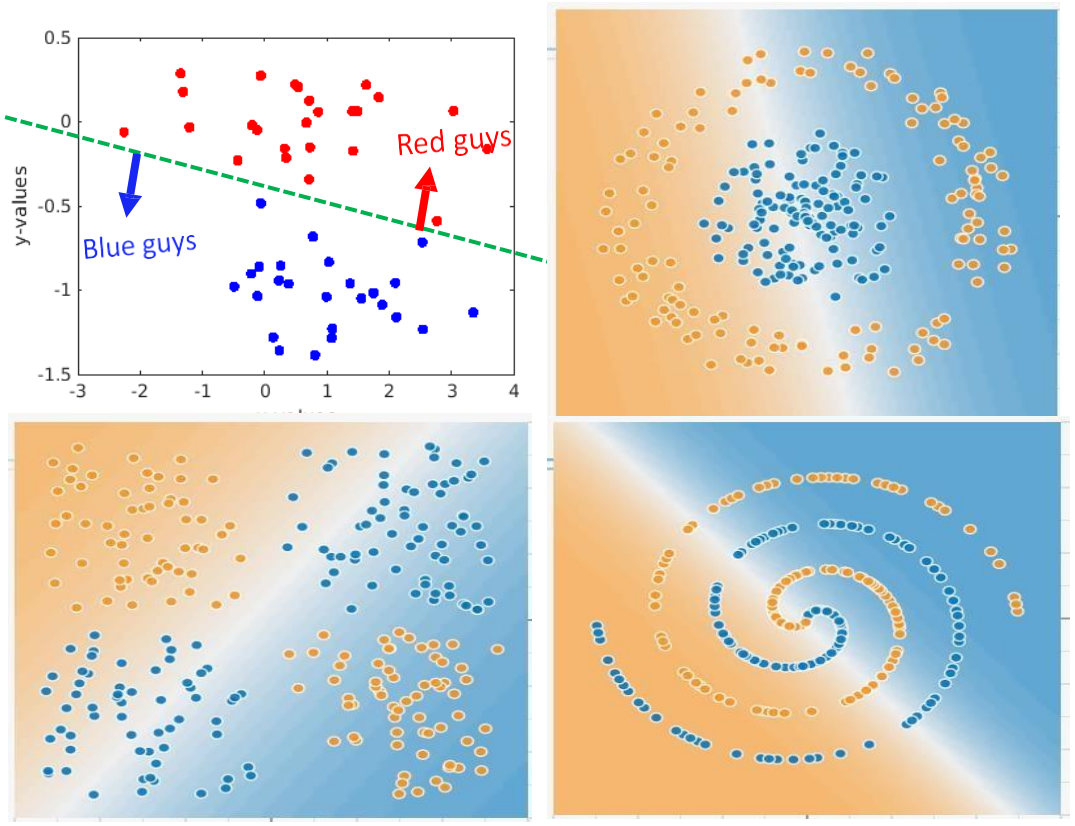
# Coming up: Remaining Problems

Typical datasets are large in size

Data points are not usually well separated by lines

Real world datasets are complex

Classic ML Algorithms struggle to learn from this data



Lancaster University

# Remaining Problem

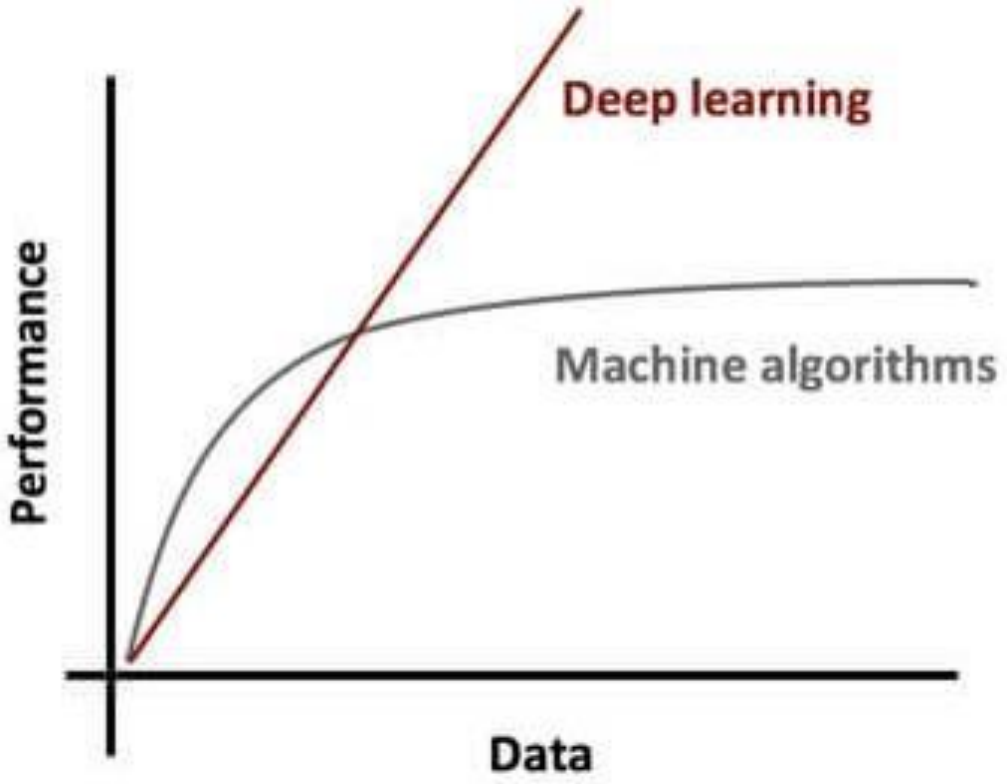
Typical datasets are large in size

Data points are not usually well separated by lines

Real world datasets are complex

ANNs can be scaled to perform well with these difficult data spaces.

Deep neural networks learn better by refining decisions at each layer



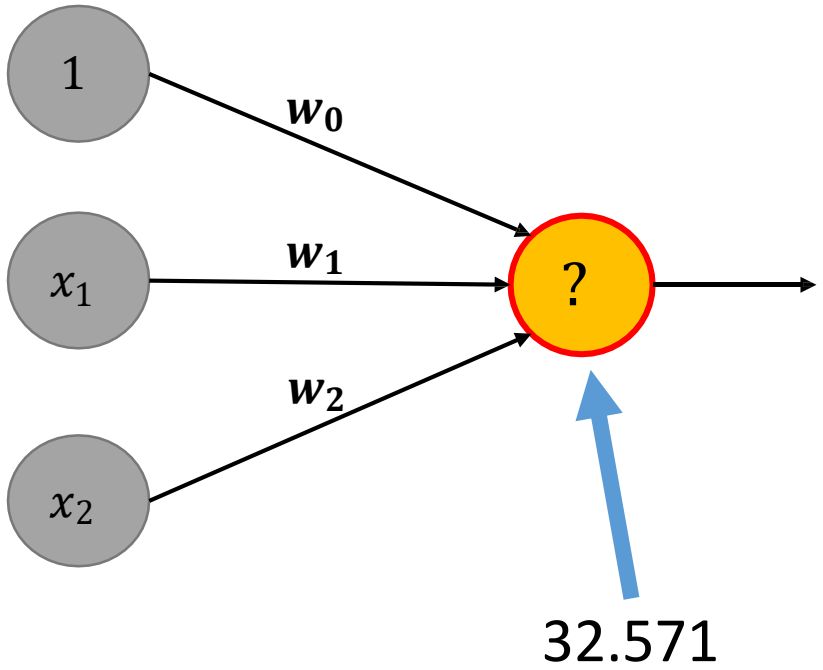

Source: https://en.wikipedia.org/wiki/Neuron

# Hypothesis Functions

$$h(x_1, x_2) = w_0 + w_1 x_1 + w_2 x_2$$

Given these values for the variables of the hypothesis function

- What will be the outcome of the hypothesis function?
- Will it **rain** or **not**?

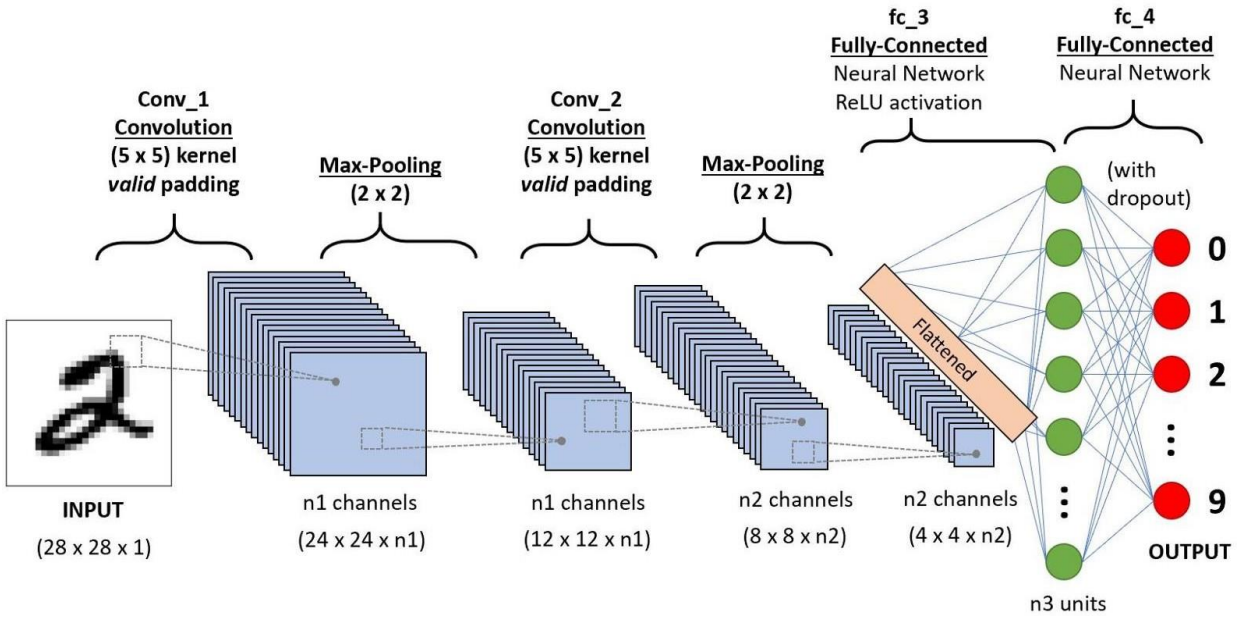



| | |
|---|---|
| $x_1$ | 9 |
| $x_2$ | 0.83 |
| $w_0$ | 1.512 |
| $w_1$ | 3.674 |
| $w_2$ | -2.418 |

Source: Wikipedia Activation Functions

# Convolutional Neural Networks

- CNNs are usually applied to computer vision tasks i.e. building models that can ‘read and understand’ images
- Application area includes
  - Image and video recognition
  - Medical image analysis
  - Recommender systems
  - etc
- Key components include:
  - Image convolution
  - Pooling (max pooling)

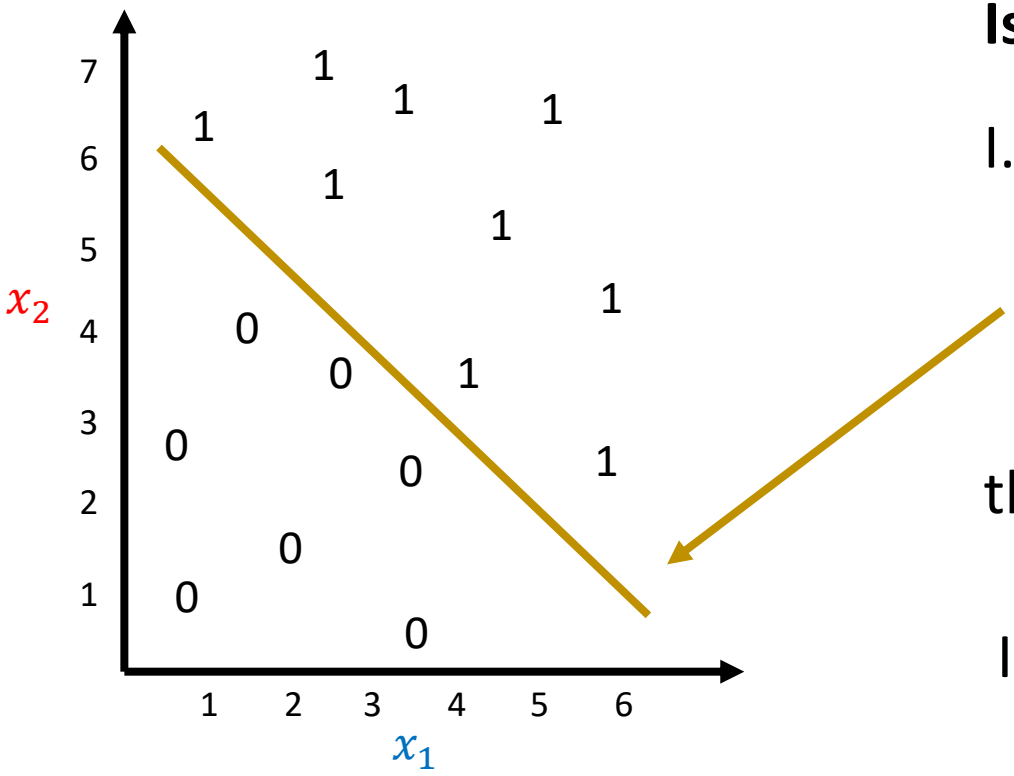


# Support Vector Machines

# Space Partitioning

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

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

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

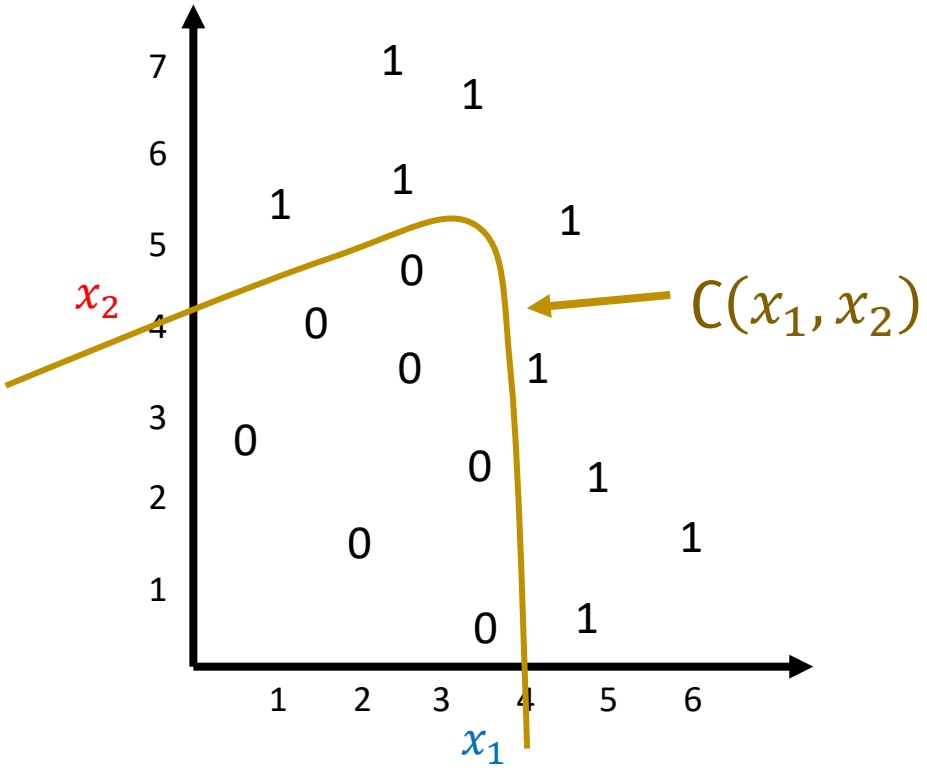
that separates the classes?

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

# Space Partitioning

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?**

Lancaster University

# Building SVM

## Problem statement

Each item $X_i$ in the set $S$ is a pair, including data and a label:

$$X_1 = (x_1, y_1), X_2 = (x_2, y_2), \ldots, X_n = (x_n, y_n)$$

where

- Each $x_i$ is a $p$-dimensional real vector
- Each $y_i$ is a corresponding label for $x_i$. For this, we will use

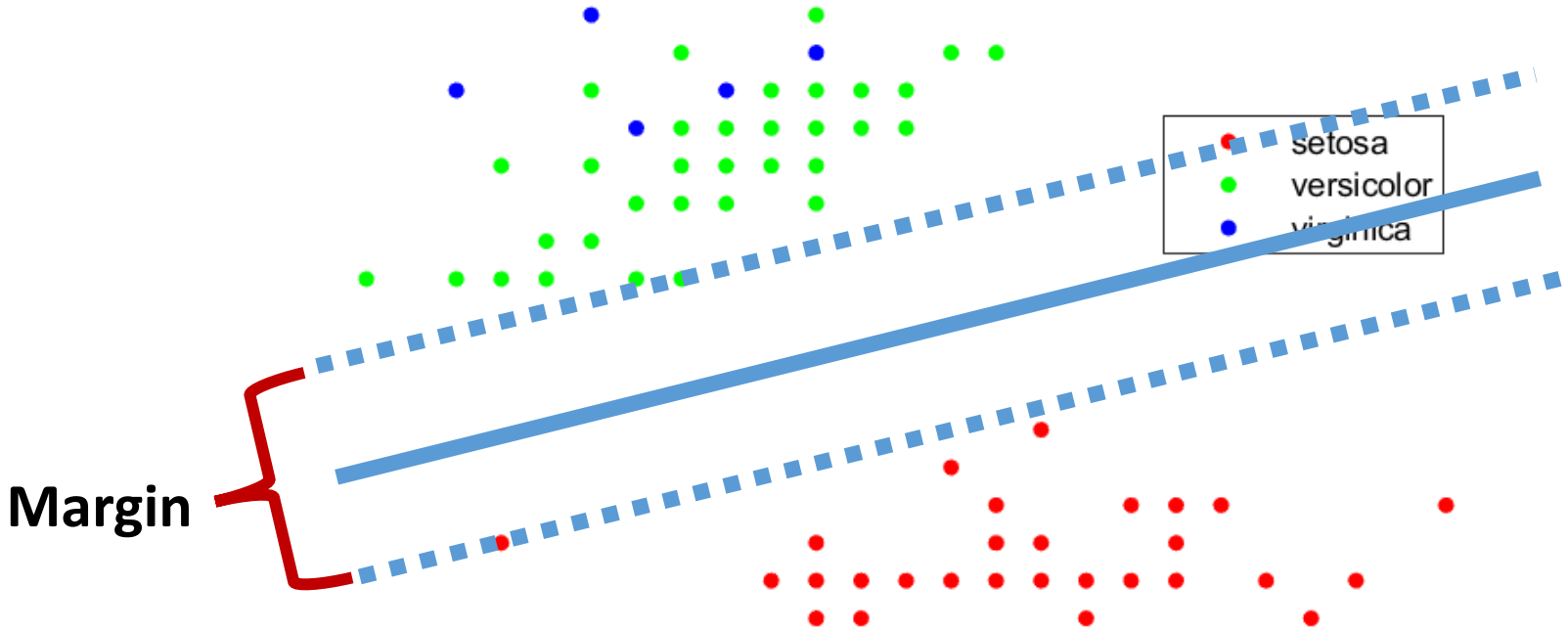
$$y_i = \begin{cases} 1 & if\ X_i \in S_1 \\ -1 & if\ X_i \in S_2 \end{cases}$$

# Building SVM

## SVM Approach

**SVM Approach:**

Find the "maximum-margin hyperplane" dividing the points $x_i \in S_1$ from the points $x_i \in S_2$ so that the distance between the hyperplane and the nearest point $x_i$ from either group is maximised.



# Building SVM

## Hyperplanes

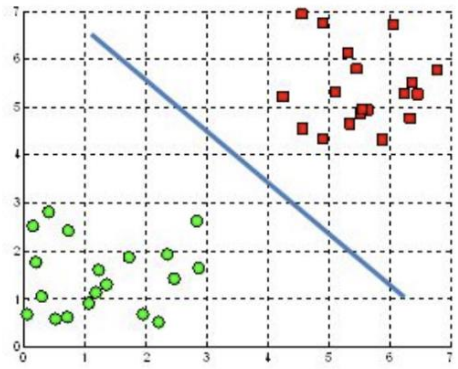
**General equation for a hyperplane:**

$$w^\top \boldsymbol{x} - b = 0, \qquad where\ \boldsymbol{x} = (x_1, x_2, \ldots, x_n)$$

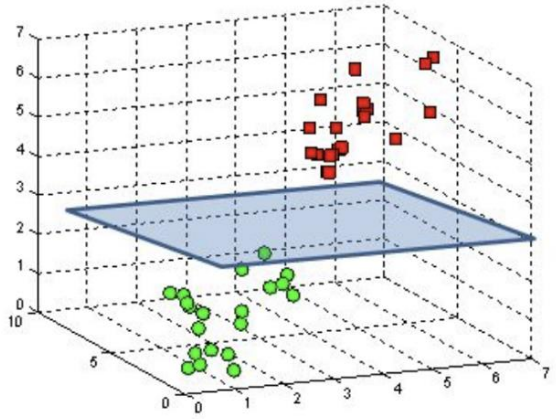
where

- $w$ is the normal vector to the hyperplane
- $\frac{b}{\|w\|}$ is the distance from the origin to the hyperplane along the normal

A hyperplane in $\mathbb{R}^2$ is a line



A hyperplane in $\mathbb{R}^3$ is a plane



Lancaster University

# Building SVM

## Margins

If the training data is linearly separable, e.g. can be separated with a straight line, we can define two parallel hyperplanes $h_1, h_2$ such that

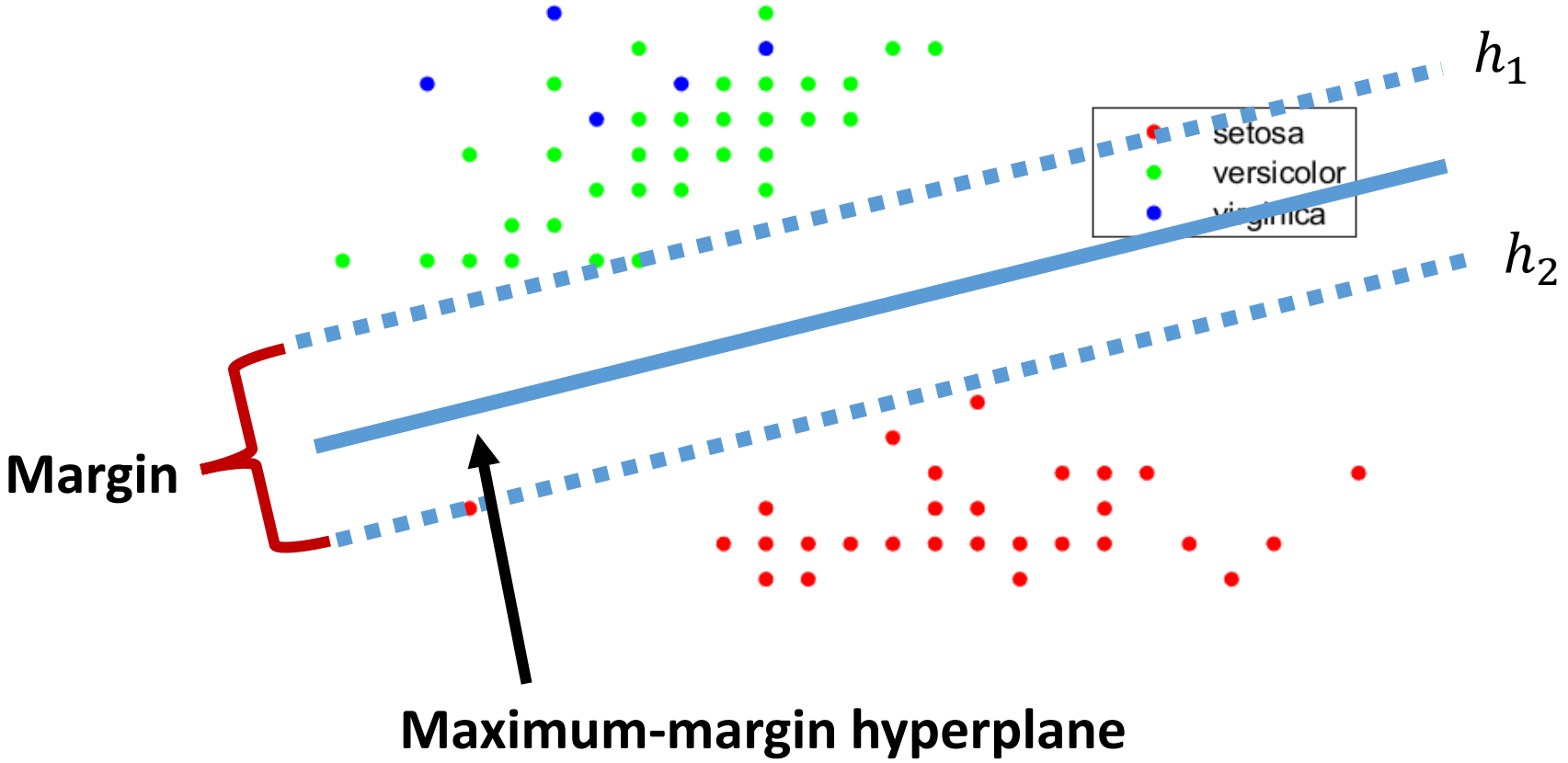
- The two classes of data are separated
  - Anything above $h_1$ is in class $S_1$
  - Anything below $h_2$ is in class $S_2$
- The distance between $h_1$ and $h_2$ is as large as possible

The region between the hyperplanes is called the ***margin***.

The hyperplane lying halfway between $h_1$ and $h_2$ is called the ***maximum-margin hyperplane.***

# Building SVM

## Margins



# Building SVM

## Margins

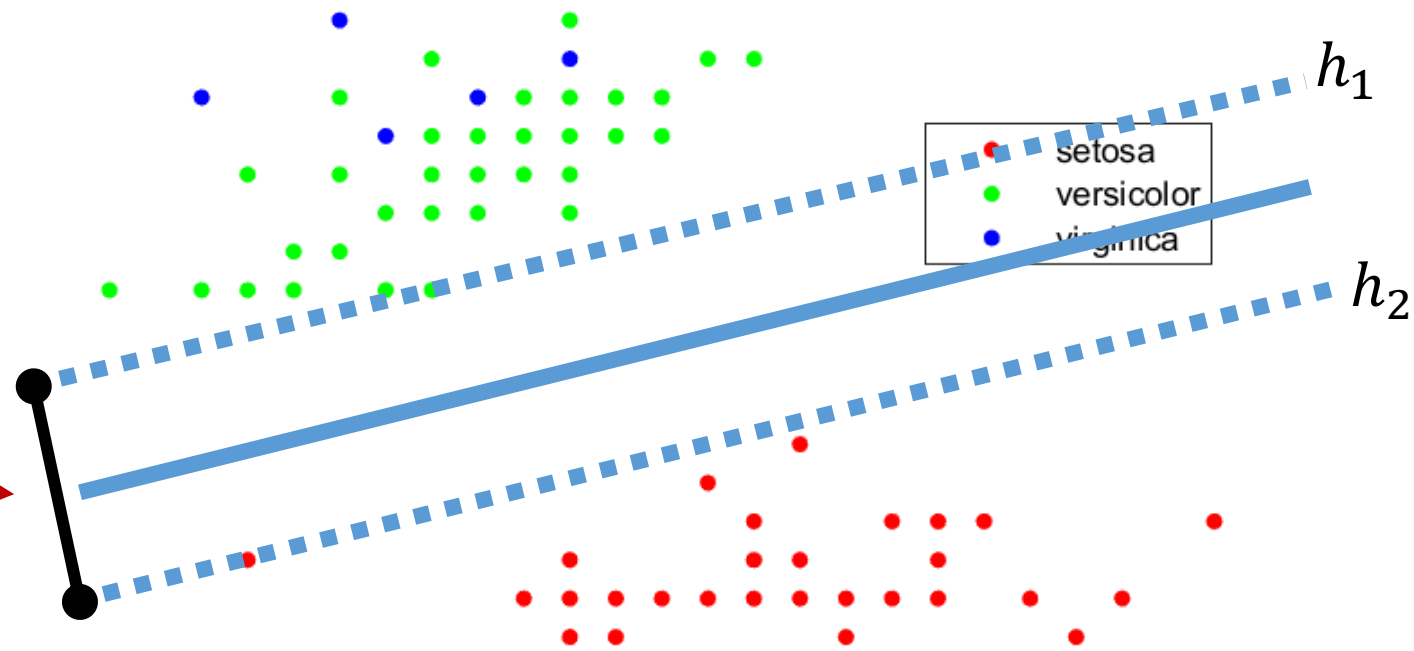
We can define these hyperplanes as

$$h_1: w^\top x - b = 1$$
$$h_2: w^\top x - b = -1$$

The distance between these is

$$\frac{2}{\|w\|}$$

To maximise this distance is the same thing as minimising $\|w\|$



Lancaster University

# Building SVM

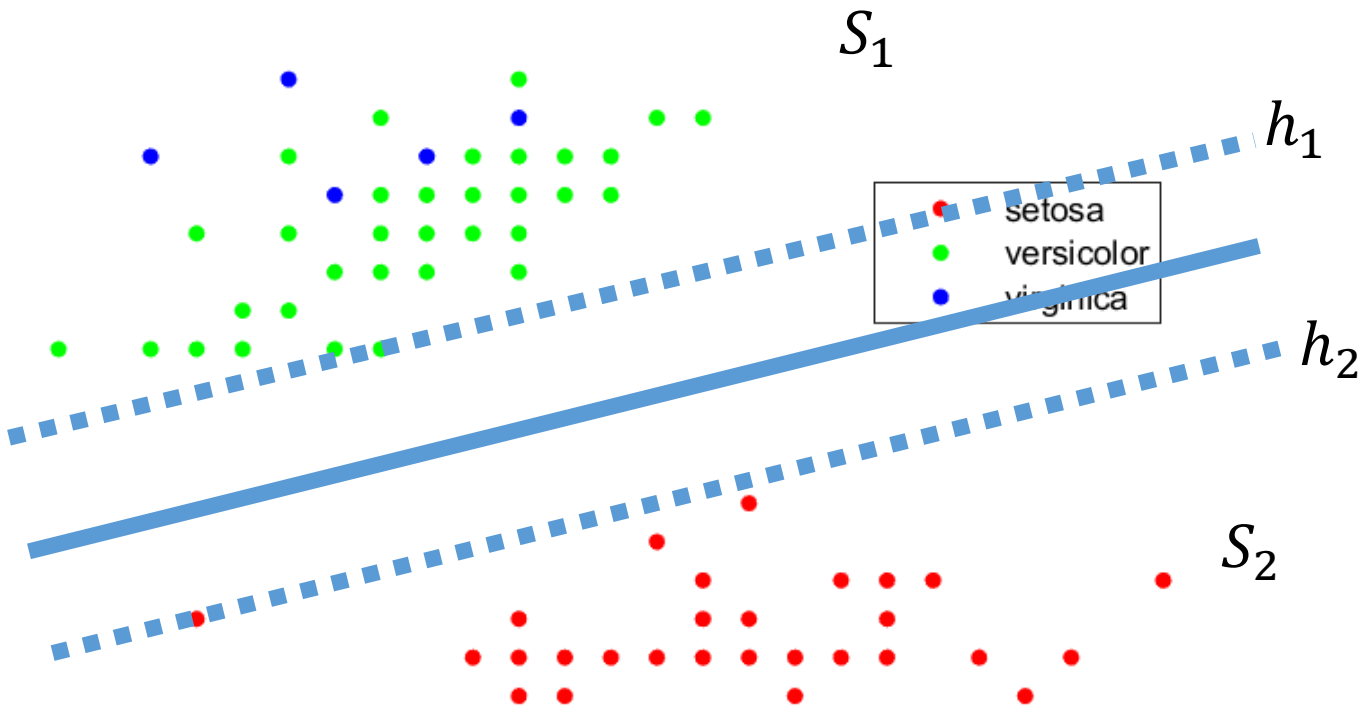
## Constraints

In order to keep the points on the correct side of the boundary, we add the following constraints

- $w^\top x_i - b \geq 1$ if $y_i = 1$ (i.e. $x_i \in S_1$)
- $w^\top x - b \leq -1$ if $y_i = -1$ (i.e. $x_i \in S_2$)

We can rewrite this as

$$y_i(w^\top x_i - b) \geq 1 \text{ for all } i = 1, \ldots, n$$



Lancaster University

# Building SVM

## Aim

So our aim is to

$$\text{minimise } \|w\|$$

subject to the condition

$$y_i(w^\top x_i - b) \geq 1 \text{ for all } i = 1, \ldots, n$$

The values of $w$ and $b$ that solve this problem determine our classifier

$$x \mapsto \text{sgn}(w^\top x - b)$$

Lancaster University

# Kernel Trick

We can create **nonlinear** classifiers by applying a kernel trick to maximum-margin hyperplanes

Linear:

$$k(x_i, x_j) = x_i \cdot x_j$$

Polynomial:

$$k(x_i, x_j) = (x_i \cdot x_j)^p$$

Radial Basis Function:

$$k(x_i, x_j) = \exp\left(-\gamma \left\| x_i - x_j \right\|^2\right)$$

Sigmoid Function:

$$k(x_i, x_j) = \tanh(\kappa_1 + \kappa_2 x_i \cdot x_j)$$