

QUEEN MARY SCHOOL HAINAN  
QUEEN MARY UNIVERSITY OF LONDON

# QHM5703 Principles of Machine Learning

## Supervised learning: Classification I

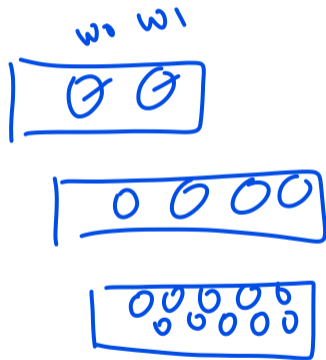
Dr Nikesh Bajaj

Week 10 - 12/13 Nov 2025



## Recap - ML Tasks

- Target population
  - Error Surface - (Empirical/True)
  - Quality (True/Testing/Training)
- 
- Test **tasks**
  - Training **tasks**
  - Validation **tasks**



Who should know about testing?

## Flexibility, complexity and overfitting

In any machine learning project we assume that our samples follow a **pattern** and any deviation from this pattern is considered to be **noise**.

Our models need to be flexible enough to **capture the complexity of the underlying pattern**, but not too flexible, as we might end up **memorising irrelevant noise** details in our data.

A **rigorous methodology** is crucial to avoid falling into common traps, such as overfitting.

So someone has collected our dataset? Great! Let's go ahead, put our methodology to work and build a fantastic model. **Great?**

## The 1936 Literary Digest Poll



Alfred Landon  
Republican Party



Franklin D. Roosevelt  
Democratic Party

- The Literary Digest conducted one of the largest polls ever.
- Predicted Landon would get 57% of the vote, Roosevelt 43%. Landon ended up getting 38% of the votes and Roosevelt 62%.
- What happened? Bad sampling. Names were taken from telephone directories, club membership lists, magazine subscribers lists, etc. Samples were **not representative** of the population.

**Know your data!**

# Agenda

Formulating classification problems

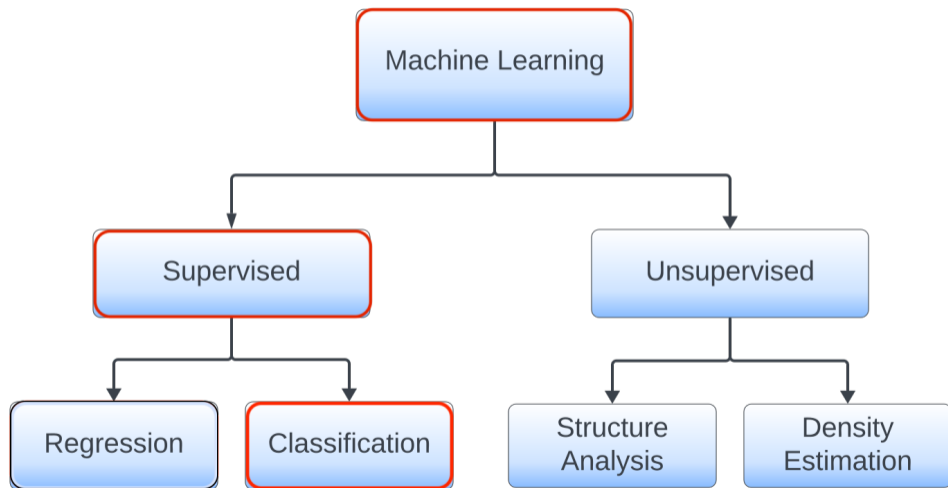
Linear classifiers

Logistic model

Nearest neighbours

Summary

# Machine Learning taxonomy



## Classification: Problem formulation

0 1 2

0 1  
A B  
(cat dog  
A B C D E

In **classification** (also known as **decision** or **detection**) problems:

- We want to build a model that produces a **label** when shown a set of **predictors**
- The label is **discrete** and each of its values is known as a **class**.

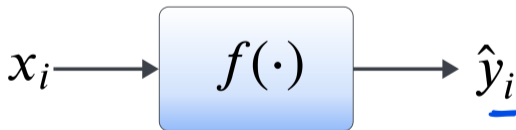
**Question:** This is not machine learning yet. Can you tell why?

# Classification: Problem formulation

In a machine learning classification problem:

- We build a model  $\hat{y} = f(x)$  using a dataset  $\{(x_i, y_i) : 1 \leq i \leq N\}$ .
- We have a notion of **model quality**.
- The pair  $(x_i, y_i)$  can be read as "sample  $i$  belongs to class  $y_i$ ", or "the label of sample  $i$  is  $y_i$ ".

$$\hat{y}_i = y_i$$



## A binary classification problem

**Sentiment analysis** seeks to identify human opinions implied by a fragment of text. Multiple opinions can be considered, but in its simplest form two are defined, namely positive and negative.

The *Large Movie Review Dataset* was created to build models that recognise polar sentiments in fragments of text:

- It contains 2500 samples for training and 2500 samples for testing
- Each instance consists of a **fragment of text** used as a **predictor** and a **binary label** (0 being negative opinion, 1 positive opinion).
- Downloadable from [ai.stanford.edu/~amaas/data/sentiment/](http://ai.stanford.edu/~amaas/data/sentiment/)

## A multiclass classification problem

Recognising digits in images containing handwritten representations is a classic multiclass classification problem. The predictor is an array of values (image) and there are 10 classes, namely 0, 1, 2, ... 9.

In machine learning we use datasets of **labelled images**, i.e. pairs of **images** (predictors) and **numerical values** (label), to build such models.



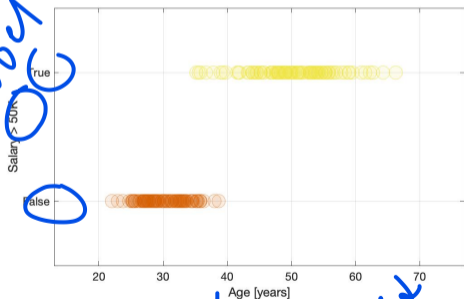
The MNIST dataset is a collection of handwritten digits:

- 60,000 images for training, 10,000 for testing
- Images are black and white, 28×28 pixels
- Downloadable from [yann.lecun.com/exdb/mnist](http://yann.lecun.com/exdb/mnist)

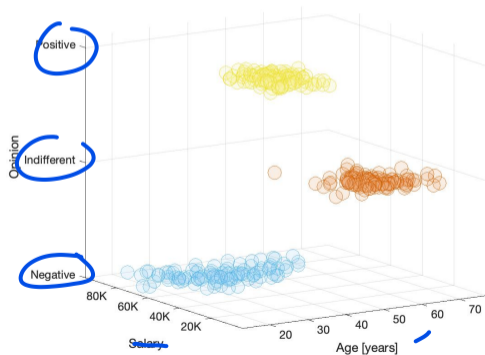
# The dataset in the attribute space

Labels can be represented by numerical values on a vertical axis. Be careful: the usual notions of **ordering** and **distance do not apply** to categorical variables.

One predictor, two classes



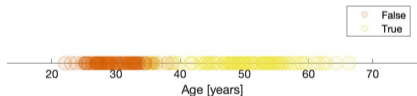
Two predictors, three classes



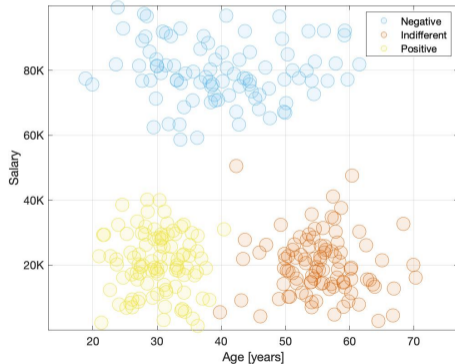
# The dataset in the predictor space

A more convenient representation consists of using **different symbols for each label** in the **predictor space**.

One predictor, two classes

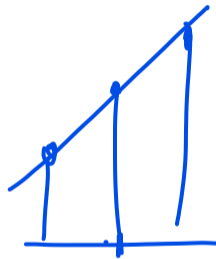
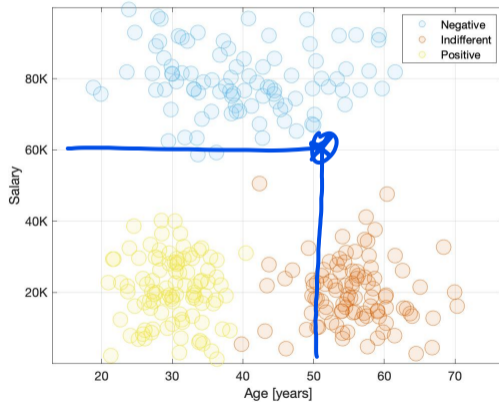


Two predictors, three classes



# What does a classifier look like?

Using the training dataset below, how would you classify an individual who is 50 years old whose salary is 60K?



Go to [menti.com](https://www.menti.com) and use code **DMT: 3397 8416** — **ICS: 4669 5998**

## What does a classifier look like?

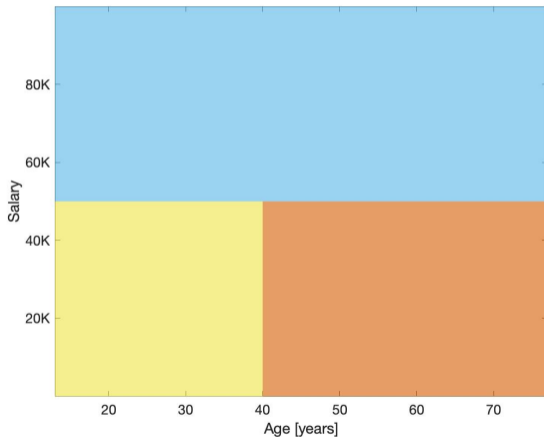
**Regression models** can be represented as curves/surfaces/hypersurfaces in the attribute space.

In **classification** problems we use the notion of **decision regions** in the predictor space.

- A decision region is made up of **points that are associated to the same label**.
- Regions can be defined by identifying their **boundaries**.
- A solution model in classification is a **partition of the predictor space** into decision regions separated by decision boundaries.

## What does a classifier look like?

In machine learning we use data to build models: When we build classifiers, we use data to define their decision regions.



# Agenda

Formulating classification problems

**Linear classifiers**

Logistic model

Nearest neighbours

Summary

# The simplest boundary

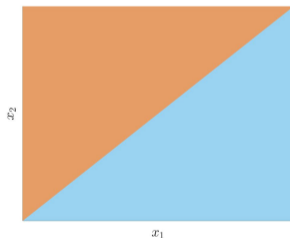
Let's consider a **binary** classification problem. The simplest boundary is:

- A single point (known as **threshold**) in 1D predictor spaces.
- A straight line in 2D predictor spaces.
- A plane surface in 3D predictor spaces.

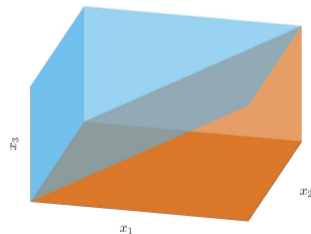
These are **linear** boundaries.



1 predictor



2 predictors



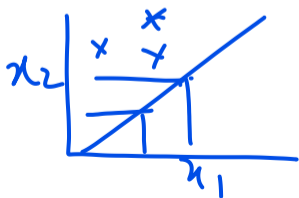
3 predictors

## Definition of linear classifiers



**Linear classifiers** use **linear boundaries** between decision regions:

- Linear boundaries are defined by the **linear equation**  $w^T x = 0$ .
- The extended vector  $x = [1, x_1, x_2, \dots]^T$  contains the predictors and  $w$  is the coefficients vector.
- To classify a sample we simply identify the **side of the boundary** where it lies



$$x_2 = mx_1 + b$$

$$b + mx_1 - x_2 = 0$$

$$[b, m, -1] \begin{bmatrix} 1 \\ x_1 \\ x_2 \end{bmatrix} = 0$$

$$w^T x = 0$$

## Definition of linear classifiers

If we know the coefficients vector  $w$  of a linear boundary, classifying a sample is very simple:

- Build the extended vector  $x_i$ .
- Compute  $w^T x_i$ .
- Classify using the following **facts**:
  - If  $w^T x_i > 0$ , we are on one side of the boundary.
  - If  $w^T x_i < 0$ , we are on the other!
  - If  $w^T x_i = 0$ ... where are we?

Now that we know how to **use** linear classifiers during deployment, let's consider the problem of **building the best** one given a dataset. To answer this question, we need to define our **quality metric** first.

# Linear classifier: Example/Question 1

Draw the boundary for the model described by  $w = [0, 1, 0]$

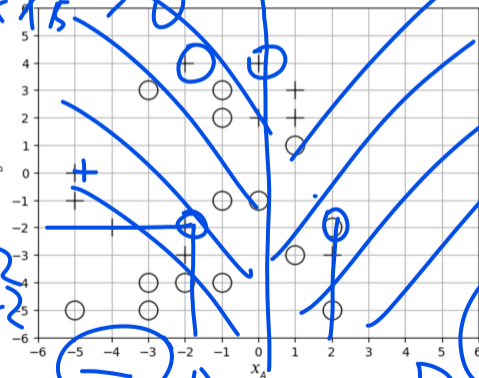
$$0x_1 + 1x_A + 0x_5 > 0$$

$$x_A > 0$$

$$(5, 0) \Rightarrow -5$$

$$(-2, 2) \Rightarrow -2$$

$$(2, -1) \Rightarrow -2$$



$$w^T x = 0$$

$$w = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \quad x = \begin{bmatrix} x_A \\ x_B \end{bmatrix}$$

$$w^T x \rightarrow 0 \quad (2 \cdot -2) = -2$$

$$0 + x_A + 0 = 0$$

$$x_A = 0$$

## A basic quality metric

The only operation that we can perform with categorical variables is **comparison**, i.e. we can assess whether  $y_i = \hat{y}_i$  is either true or false.

By comparing predictions and true labels, we can identify in a dataset:

- The correctly classified samples (**true predictions**) in each class.
- The incorrectly classified samples (**false predictions**) in each class.

Two common and equivalent notions of quality are the **empirical accuracy**  $\hat{A}$  and **error** (or misclassification) **rate**  $\hat{E} = 1 - \hat{A}$ , defined as:

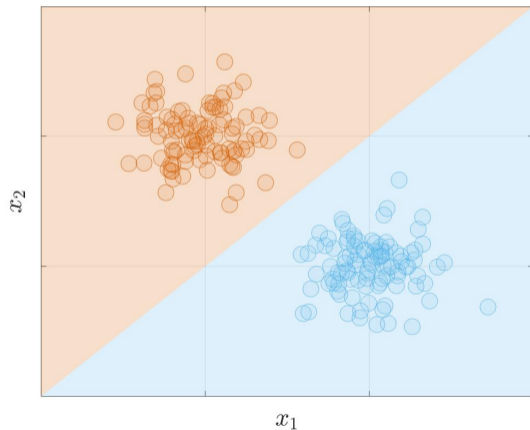
$$\hat{A} = \frac{\text{\#true predictions}}{\text{\#samples}}, \quad \hat{E} = \frac{\text{\#false predictions}}{\text{\#samples}}$$

$$E = 1 - A$$

The **true accuracy**  $A$  and **error rate**  $E$  are defined analogously on the target population.

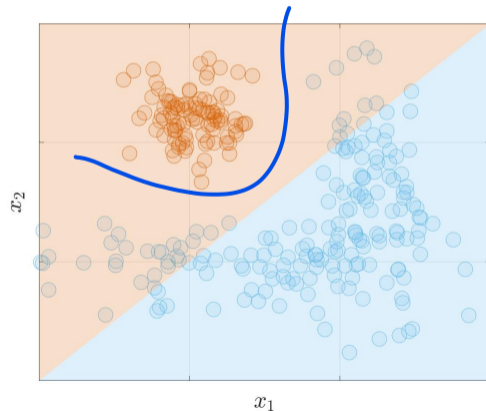
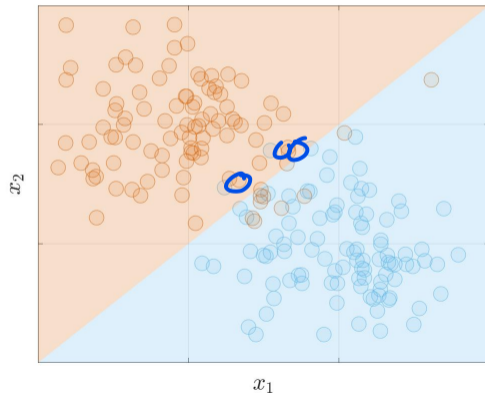
## The best linear classifier: Separable case

In linearly separable datasets, we can find a linear classifier that achieves the maximum accuracy ( $\hat{A} = 1$ ,  $\hat{E} = 0$ ).



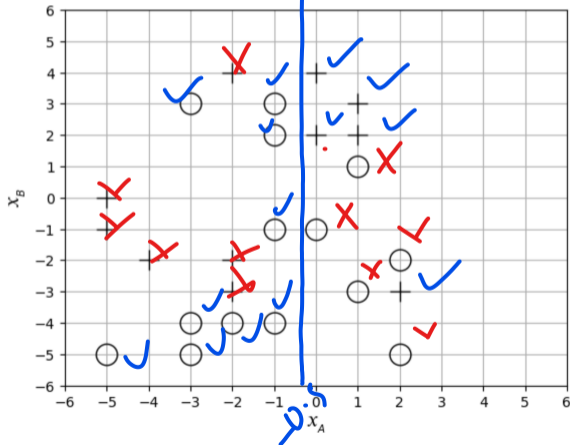
## The best linear classifier: Non separable case

In non linearly-separable datasets, the accuracy of a linear classifier is  $\hat{A} < 1$  ( $\hat{E} > 0$ ). The best one will achieve the highest accuracy.



## Linear classifier: Example/Question 2

Compute the accuracy  $\hat{A}$  and the error rate  $\hat{E}$  for the model described by  $f_1 : w = [0.5, 1, 0]$  and the decision as if  $w^T x > 0 \rightarrow \hat{y} = +$



$$\frac{14}{25} = \hat{A}$$
$$\frac{11}{25} = \hat{E}$$
$$w = [0 \ 1 \ 0]$$

## Linear classifier: boundary

What should we do, if sample is on the boundary?

# Linear classifier: Example/Question 3

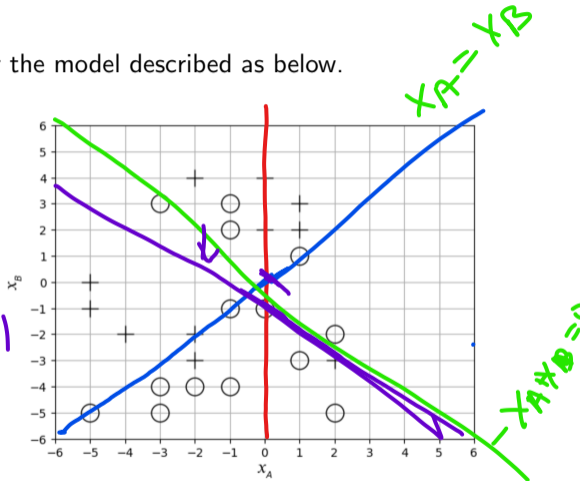
Compute the accuracy  $\hat{A}$  and the error rate  $\hat{E}$  for the model described as below.

- $f_1 : w = [0, 1, 0]$
- $f_2 : w = [0, 1, 1]$
- $f_3 : w = [1, 1, 1]$

$$x_A + x_B = 0$$

$$x_A + x_B = -1$$

Consider the decision as:  
if  $w^T x \geq 0 \rightarrow \hat{y} = +$



## Finding the best linear classifier

We have introduced the linear classifier and presented two quality metrics (accuracy and error rate). If we are given a linear classifier  $w$ , we can use a dataset to calculate its quality.

The question we want to answer now is, how can we **use data to find** the best linear classifier?

# Agenda

Formulating classification problems

Linear classifiers

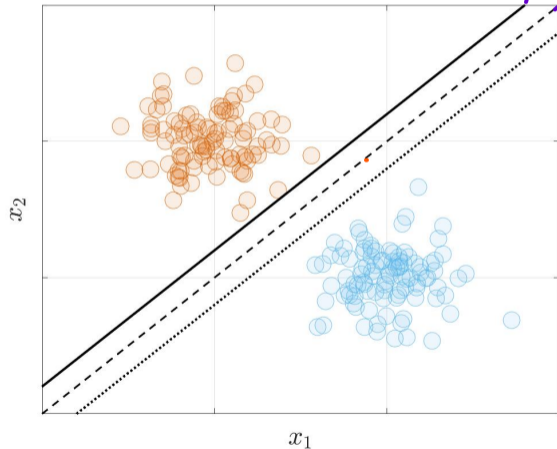
**Logistic model**

Nearest neighbours

Summary

## Best, but risky, linear solutions

Which one of the three boundaries below would you choose?



Solid  
dashed  
dotted

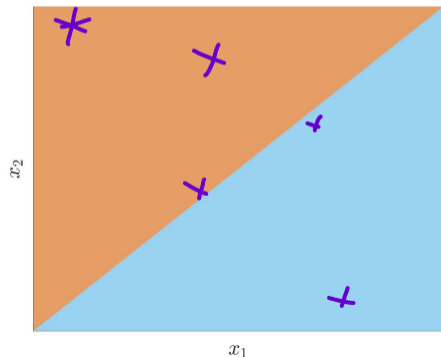
All three boundaries have  
same accuracy on given  
points.

Go to [menti.com](https://www.menti.com) and use  
code ~~1236 2386~~

## Keep that boundary away from me!

As we get closer to the decision boundary, life gets harder for a classifier: it is **noise territory** and we should beware of **jumpy samples**.

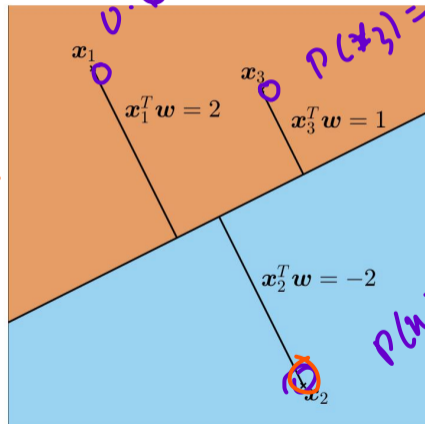
The **further** we are from the boundary, the **higher our certainty** that we are classifying samples correctly.



# How far is my sample from the boundary?

Given a linear boundary  $w$  and a sample  $x_i$ , the quantity  $d_i = w^T x_i$  can be interpreted as the **distance** from the sample to the boundary.

$$L = (+0.7) \cdot 0.8 \times 0.6$$
$$= 0.8 \times 0.8 \times 0.6$$



$$L = 0.8 \times 0.6 \times 0.2$$

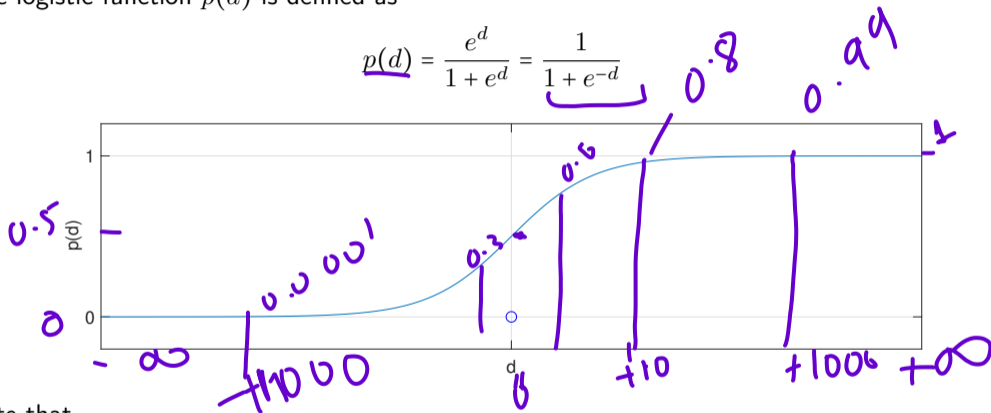
$$p(x_2) = 0.2$$

$x_1$

# Quantifying our uncertainty: The logistic model

The logistic function  $p(d)$  is defined as

$$p(d) = \frac{e^d}{1 + e^d} = \frac{1}{1 + e^{-d}}$$



Note that

- $p(0) = 0.5$ .
- As  $d \rightarrow \infty$ ,  $p(d) \rightarrow 1$ .
- As  $d \rightarrow -\infty$ ,  $p(d) \rightarrow 0$ .

## Quantifying our uncertainty: The logistic model

$$\begin{array}{l} w^T x > 0 \rightarrow A \\ w^T x < 0 \rightarrow B \end{array}$$

If we set  $d = w^T x_i$  in the logistic function, we get:

$$p(w^T x_i) \Rightarrow A$$

$$p(\underline{w^T x_i}) = \frac{e^{w^T x_i}}{1 + e^{w^T x_i}}$$

$$[0 \quad 1]$$

For a fixed  $w$ , we will simply denote it as  $p(\underline{x_i})$  to simplify the notation:

- When  $w^T x \rightarrow \infty$ , the logistic function  $p(x_i) \rightarrow 1$
- When  $w^T x \rightarrow -\infty$ , the logistic function  $p(x_i) \rightarrow 0$

We will use the logistic function to quantify the notion of **certainty** in classifiers. This certainty is a quantity between 0 and 1.

## Quantifying our uncertainty: The logistic model

Consider a linear classifier  $w$  that labels samples such that  $w^T x_i > 0$  as  $\circ$  and samples such that  $w^T x_i < 0$  as  $\circ$ .

Notice that:

- If  $w^T x_i = 0$  ( $x_i$  is on the boundary),  $p(x_i) = 0.5$ .
- If  $w^T x_i > 0$  ( $x_i$  is in the  $\circ$  region),  
 $p(x_i) \rightarrow 1$  as we move away from the boundary.
- If  $w^T x_i < 0$  ( $x_i$  is in the  $\circ$  region),  
 $p(x_i) \rightarrow 0$  as we move away from the boundary.

Here is the crucial point, so use **all your neurons**:

- $p(x_i)$  is the **classifier's certainty** that  $y_i = \circ$  true.
- $1 - p(x_i)$  is the **classifier's certainty** that  $y_i = \circ$  true.

# The logistic classifier

We can obtain the classifier's certainty that  $\mathbf{x}_i$  belongs to either  $\circ$  or  $\bullet$ . Can we calculate the certainty for a **labelled dataset**  $\{(\mathbf{x}_i, y_i)\}$ ?

The answer is yes, by **multiplying** the individual certainties:

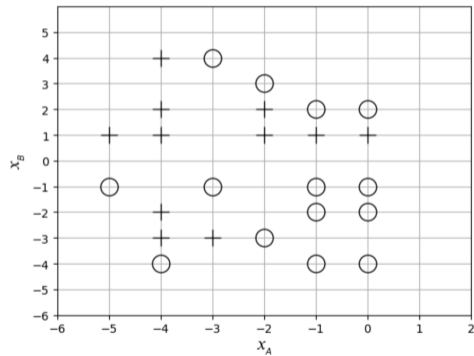
$$L = \prod_{y_i=\circ} (1 - p(\mathbf{x}_i)) \prod_{y_i=\bullet} p(\mathbf{x}_i) .$$

$L$  is known as the **likelihood function** and defines a **quality metric**. Taking logarithms, we obtain the **log-likelihood**:

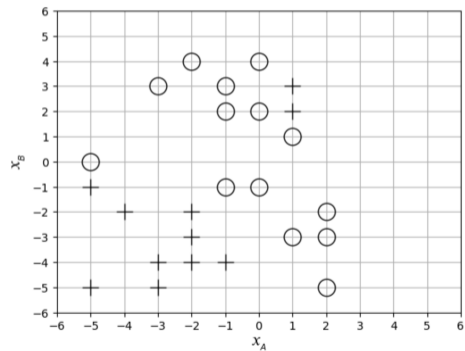
$$l = \sum_{y_i=\circ} \log [1 - p(\mathbf{x}_i)] + \sum_{y_i=\bullet} \log [p(\mathbf{x}_i)]$$

The linear classifier that maximises  $L$  or  $l$  is known as the **Logistic Regression** classifier. It can be found using **gradient descent**.

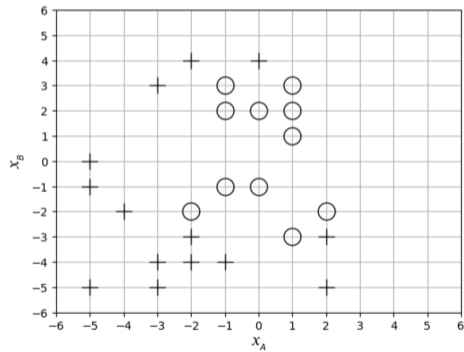
## Linear classifier: Example/Question 4



## Linear classifier: Example/Question 5



# Linear classifier: Example/Question 6



# Agenda

Formulating classification problems

Linear classifiers

Logistic model

**Nearest neighbours**

Summary

## Parametric and non-parametric approaches

Linear classifiers belong to the family of **parametric** approaches: a shape is assumed (in this case linear) and our dataset is used to find the best boundary amongst all the boundaries with the preselected shape.

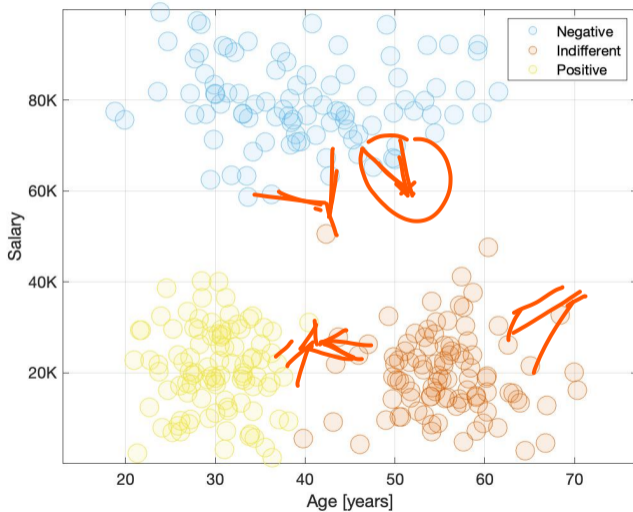
**Non-parametric** approaches offer a more flexible alternative, as they do not assume any type of boundary. In this section, we will study a popular non-parametric approach, namely **k**  
**Nearest Neighbours** (kNN).

## Nearest Neighbours

In nearest neighbours (NN), new samples are assigned the **label of the closest** (*most similar*) **training sample**. Therefore:

- Boundaries are not defined explicitly (although they exist and can be obtained).
- The whole training dataset needs to be **memorised**. That's why sometimes we say NN is an **instance-based method**.

# Boundaries in Nearest Neighbours classifiers



$k = S$

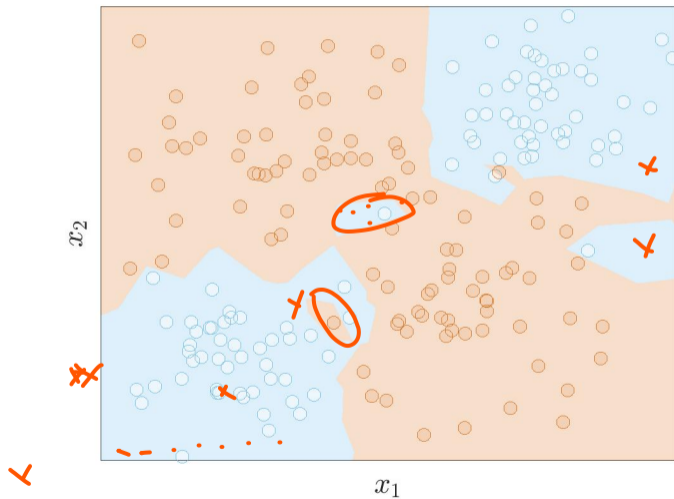
## k Nearest Neighbours

Boundaries in nearest neighbours classifiers can be too complex and hard to interpret. Can we smooth them?

K nearest neighbours (kNN) is a simple extension of nearest neighbours that proceeds as follows. Given a new sample  $x$ :

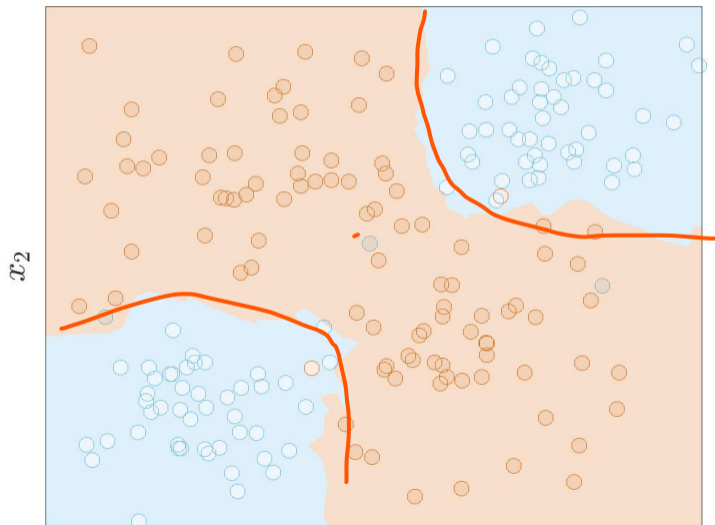
- We calculate the distance to all the training samples  $x_i$ .
- Extract the  $K$  closest samples (neighbours).
- Obtain the number of neighbours that belong to each class.
- Assign the label of the most popular class among the neighbours.

# Boundaries in kNN classifiers $k = 1$



Blue 30  
Orange 70

## Boundaries in kNN classifiers $k = 3$



## Boundaries in kNN classifiers $k = 15$



## k Nearest Neighbours

Note that:

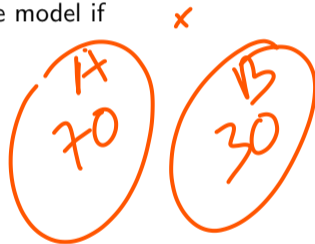
- There is always an implicit boundary, although it is not used to classify new samples.
- As  $K$  increases, the boundary becomes less complex. We move away from **overfitting** (small  $K$ ) to **underfitting** (large  $K$ ) classifiers.
- In binary problems, the value of  $K$  is usually an odd number. The idea is to prevent situations where half of the nearest neighbours of a sample belong to each class.
- kNN can be easily implemented in multi-class scenarios.

## k Nearest Neighbours: Question

Consider a dataset with 70 samples in class A and 30 samples in class B (total 100 samples). Using k Nearest Neighbours, what would be the accuracy of the model if

- we use k=100?
- we use k=71?
- we use k=31?
- we use k=200?

→ 70%



## Question [Think about it - HW]

- Can we solve a regression problem as classification?
- Can we solve a classification problem as regression?

Think about it, if we **can** and if we **should**.  
[Homework]

0 1 2 3 4 5 6 7 8 9

# Agenda

Formulating classification problems

Linear classifiers

Logistic model

Nearest neighbours

Summary

# Machine learning classifiers

- Classifiers are partitions of the predictor space into **decision regions** separated by **boundaries**.
- Each decision region is associated with one label.
- In machine learning, classifiers are built using a **dataset** (otherwise it's not machine learning!).

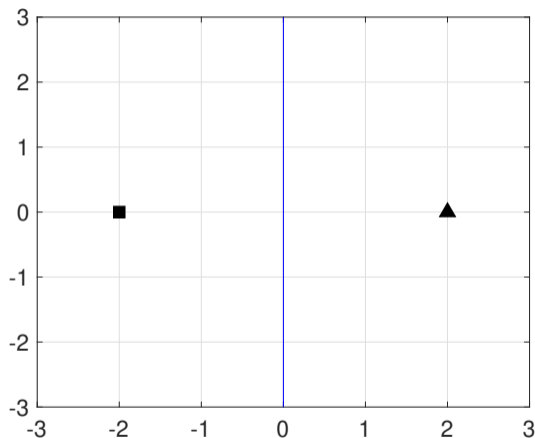
## Flexibility and complexity in classifiers

- The notions of flexibility, complexity, interpretability, overfitting and underfitting also apply to classifiers.
- Linear boundaries are simple and rigid; kNN produces boundaries whose complexity depends on the value of  $K$ .
- Logistic regression is a strategy to train linear classifiers. It's called *regression* because indirectly we solve a regression problem or the classifier's certainty.
- Weirdly, kNN does not involve training as it uses all the samples each time a new sample is to be classified.

## Hey, hold on a second, what's going on?

- In machine learning we use a quality metric to define what we mean by the *best* model.
- We have presented two quality metrics: **accuracy** and **error rate**.
- However, **neither** the logistic regression nor kNN classifiers **use the notion of accuracy**.
- What's going on?

## Example 1



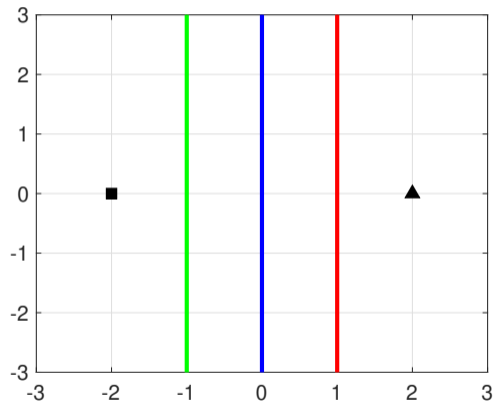
- Let's define  $d_i = \mathbf{w}^T \mathbf{x}_i$
- We can rewrite the logistic function as

$$p(d_i) = \frac{e^{d_i}}{1 + e^{d_i}}$$

- For instance  $p(0) = 0.5$ ,  
 $p(1) \approx 0.73$ ,  $p(2) \approx 0.88$ ,  
 $p(-1) \approx 0.27$  and  $p(-2) \approx 0.12$

Assume this linear classifier labels samples on the right half-plane as  $\triangle$  and samples on the left half-plane as  $\square$ .

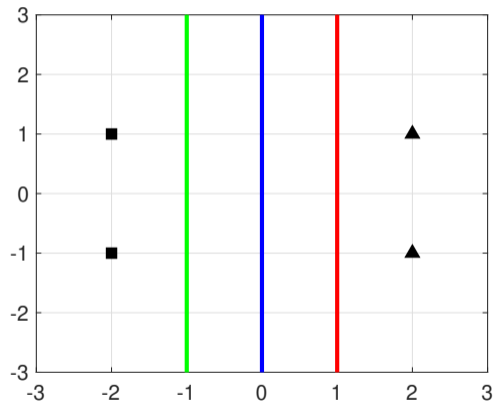
## Example II



The global certainty of each classifier (i.e. boundary) is:

- $L = p(\triangle) (1 - p(\square)) \approx 0.70$
- $L = p(\triangle) (1 - p(\square)) \approx 0.77$
- $L = p(\triangle) (1 - p(\square)) \approx 0.70$

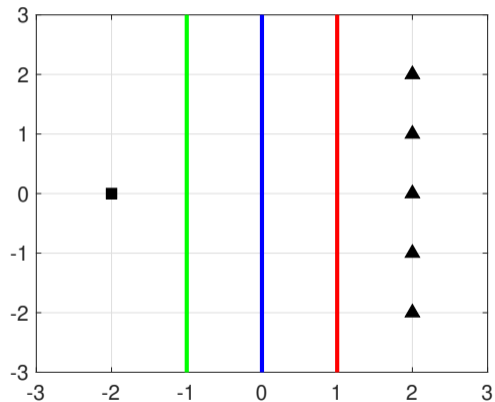
## Example III



The global certainty of each classifier (i.e. boundary) is:

- $L \approx 0.49$
- $L \approx 0.60$
- $L \approx 0.49$

## Example IV



The global certainty of each classifier (i.e. boundary) is:

- 
- $L \approx 0.20$
- $L \approx 0.47$
- $L \approx 0.57$



Queen Mary  
University of London