EC-350 AI and Decision Support Systems

Week 9
K-Nearest Neighbour Classifier

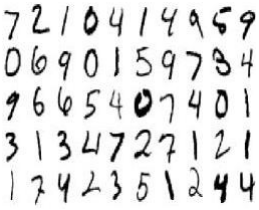
Dr. Arslan Shaukat

NUST

Acknowledgements: Lecture slides material from
Duda, Hart and Stork, Dr. Gavin Brown

Problem Statement

- Why recognising rugby players is (almost) the same problem as *recognising handwritten digits*



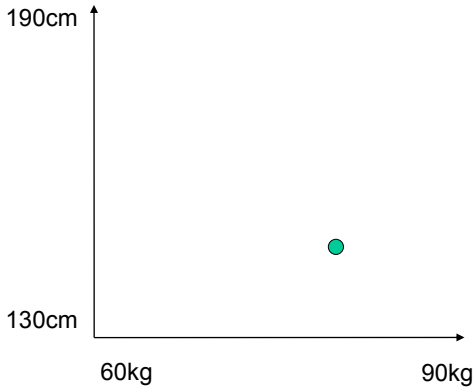
Problem Statement

Can we LEARN to recognise a rugby player and ballet dancer?

What are the “features” of a rugby player?

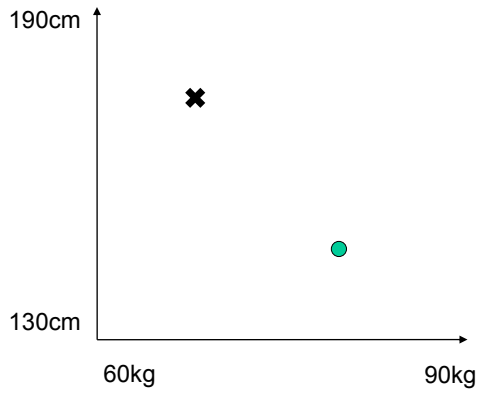
Features

Rugby players = short + heavy?



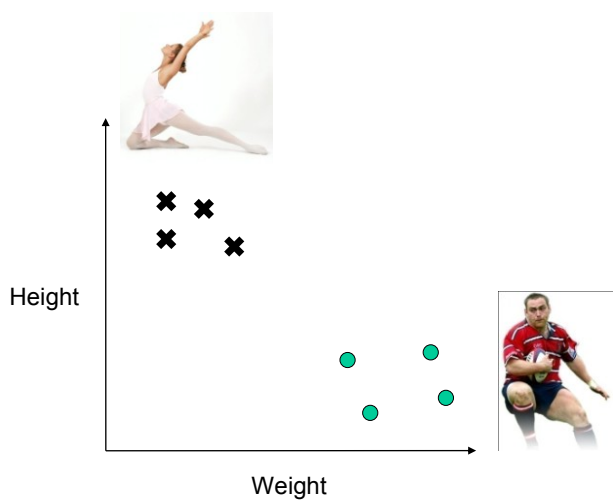
Features

Ballet dancers = tall + skinny?

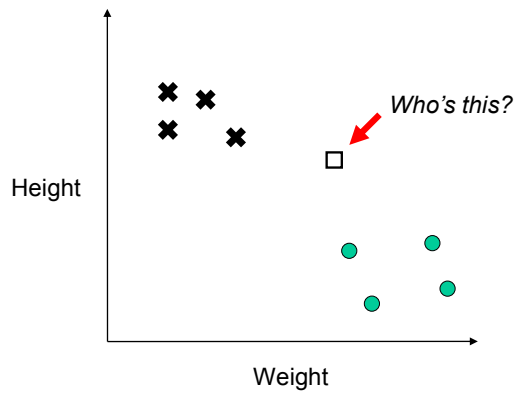


Feature Space

Rugby players “cluster” separately in the space.

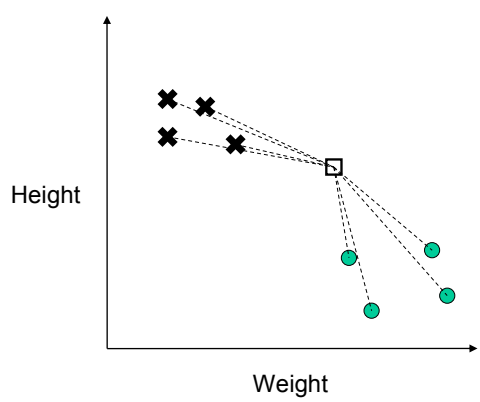


The K-Nearest Neighbour Algorithm



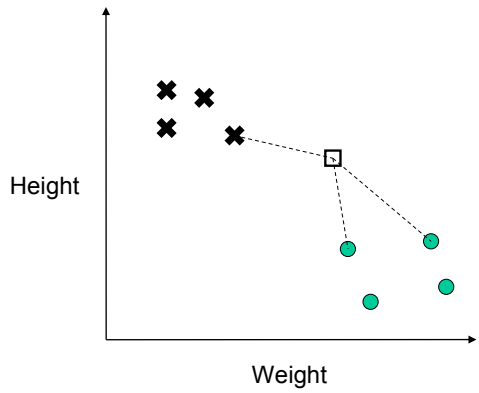
The K-Nearest Neighbour Algorithm

1. *Measure distance to all points*



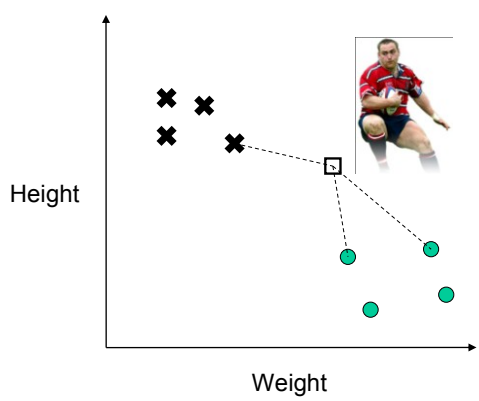
The K-Nearest Neighbour Algorithm

1. *Measure distance to all points*
2. *Find closest "k" points* ← **(here k=3, but it could be more)**



The K-Nearest Neighbour Algorithm

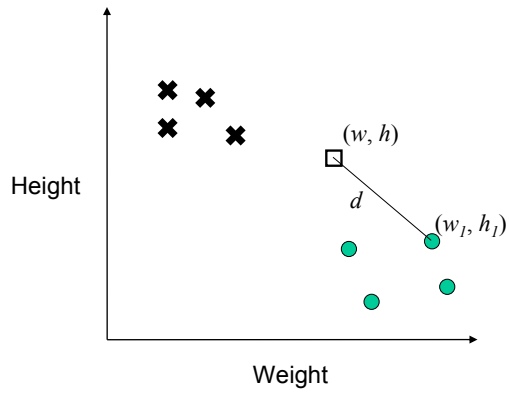
1. *Measure distance to all points*
2. *Find closest "k" points* ← **(here k=3, but it could be more)**
3. *Assign majority class*



Distance Measure

"Euclidean distance"

$$d = \sqrt{(w - w_1)^2 + (h - h_1)^2}$$



The K-Nearest Neighbour Algorithm

for each testing point

measure distance to every training point

find the k closest points

identify the most common class among those k

assign that class

end

- Advantage: Surprisingly good classifier!
- Disadvantage: Have to store the entire training set in memory

Distance Measure

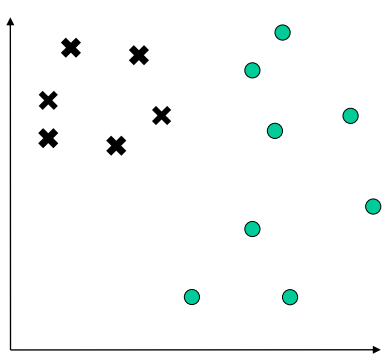
Euclidean distance still works in 3-d, 4-d, 5-d, etc….

$$d = \sqrt{(x - x_1)^2 + (y - y_1)^2 + (z - z_1)^2}$$

x = Height
y = Weight
z = Shoe size

Over-fitting

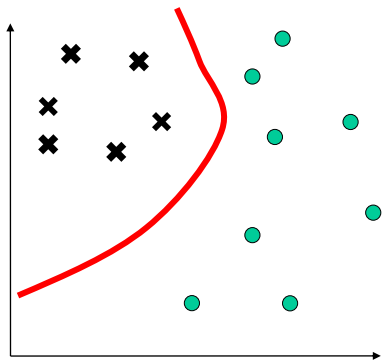
An **Important** Concept in Machine Learning



Over-fitting

An **Important** Concept in Machine Learning

Looks good so far…

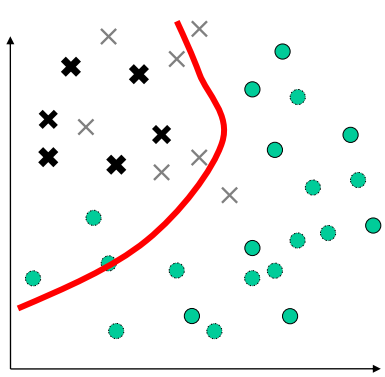


Over-fitting

An **Important** Concept in Machine Learning

Looks good so far…

Oh no! Mistakes!
What happened?

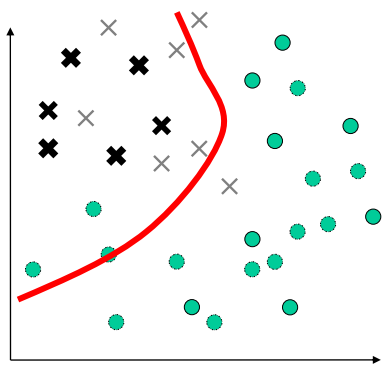


Over-fitting

An **Important** Concept in Machine Learning

Looks good so far…

Oh no! Mistakes!
What happened?



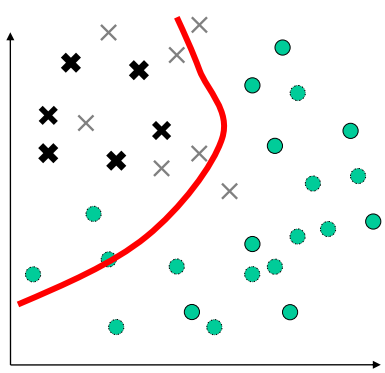
We didn't have all the data.

We can never assume that we do.

This is called "OVER-FITTING" to the small dataset.

Over-fitting

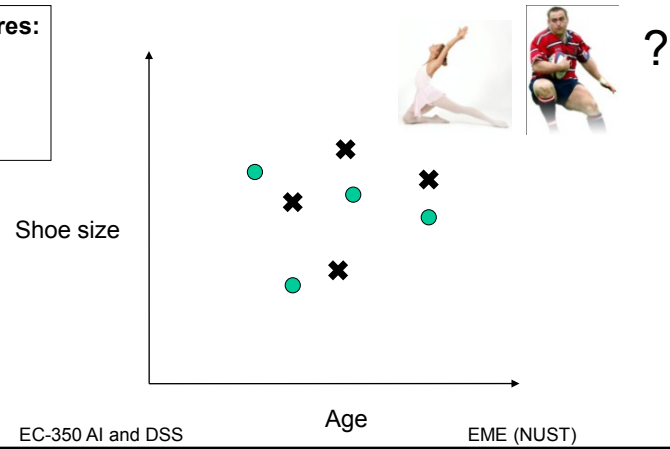
- While an overly complex boundary may allow perfect classification of the training samples, it is unlikely to give good classification of novel patterns



Features

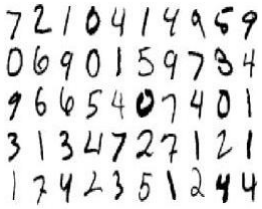
Choosing the wrong features makes it difficult
Too many features
It's computationally intensive

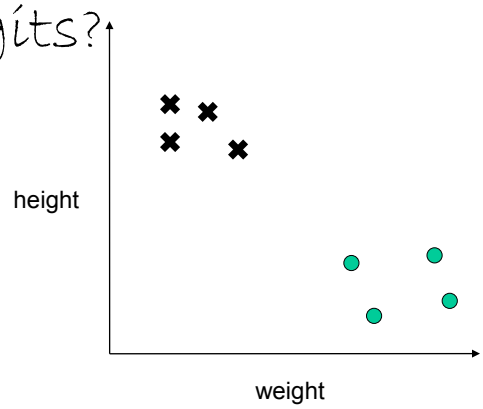
Possible features:
- Shoe size ✔
- Height
- Age ✔
- Weight



Our Problem

Now – how is this problem like *recognising handwritten digits?*

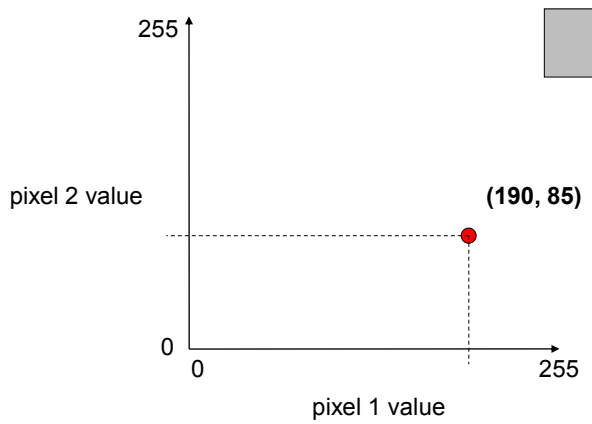




Recognising Handwritten Digits

Let's say the axes now represent **pixel values.**

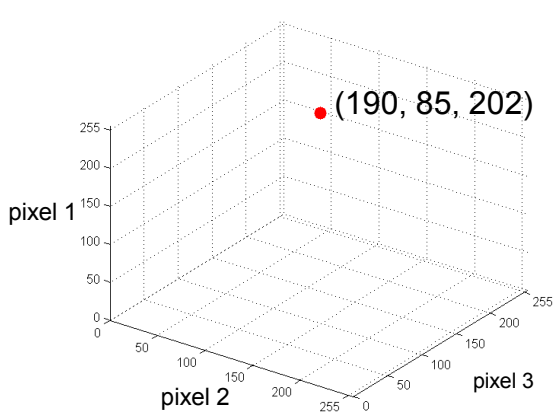
A two-pixel image



Recognising Handwritten Digits

Three dimensions…

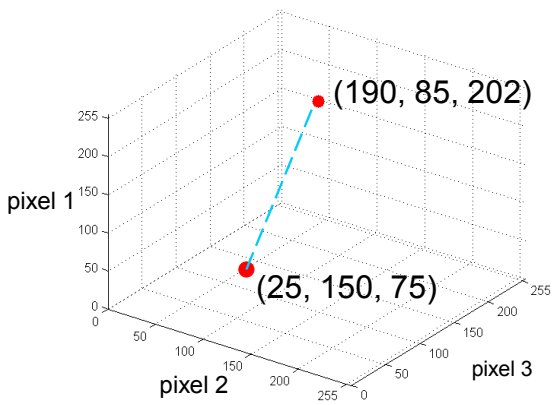
A three-pixel image



This 3-pixel image is represented by a **SINGLE** point in a 3-D space.

Recognising Handwritten Digits

Distances between images



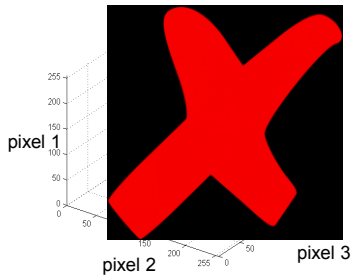
A three-pixel image

Another 3-pixel image

Straight line distance between them

Recognising Handwritten Digits

Four dimensions? Five? Six-dimensional spaces?



A three-pixel image

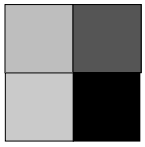
A four-pixel image.

A five-pixel image

Recognising Handwritten Digits

A four-pixel image.

A different four-pixel image.

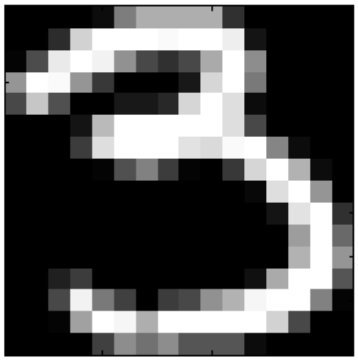


(190, 85, 202, 10)
Same 4-dimensional vector!

Assuming we read pixels in a systematic manner, we can now represent any image as a <u>single point in a high dimensional space.</u>

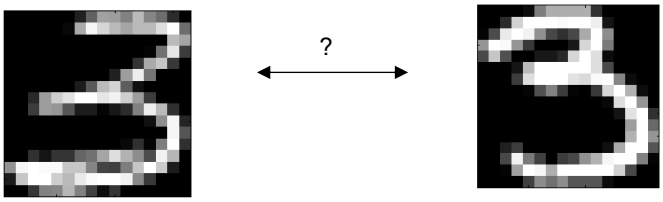
Recognising Handwritten Digits

16 x 16 image…. How many dimensions?

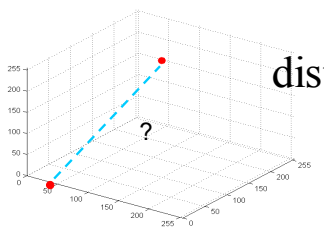


Recognising Handwritten Digits

Distances between digits….

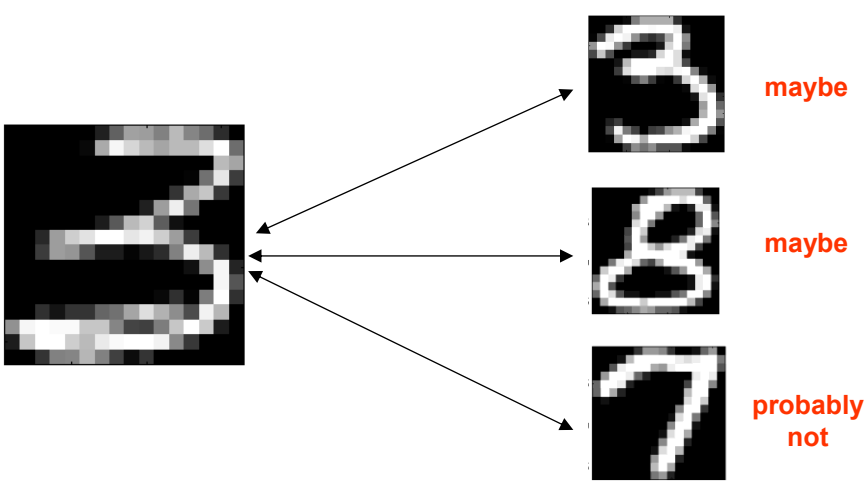


Straight-line distance in N-dimensional space?



$$\text{distance}(x, x') = \sqrt{\sum_{i=1}^{i=256} (x_i - x'_i)^2}$$

Recognising Handwritten Digits

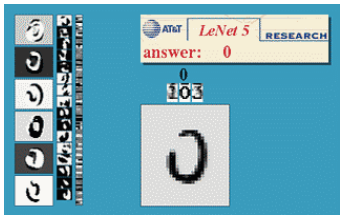


Which is **closest neighbour** in N-dimensions?

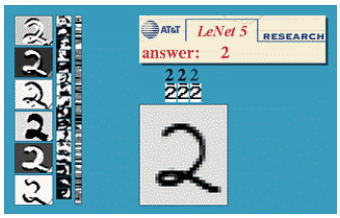
Recognising Handwritten Digits

AT&T Research Labs

The USPS ZIP code reader – **learnt** from examples.

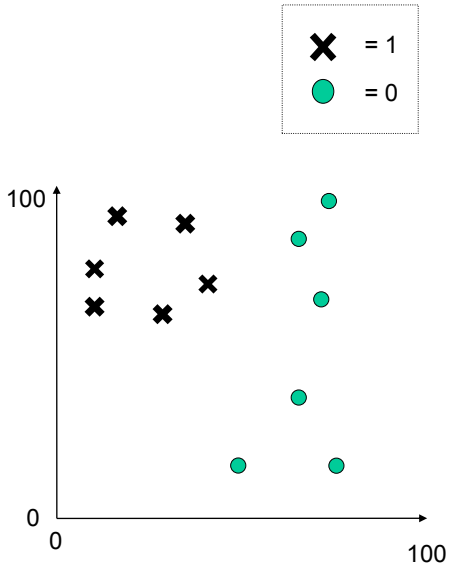


FAST recognition



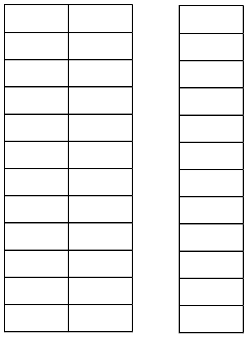
NOISE resistant, can **generalise** to future unseen patterns

Training and Testing Data

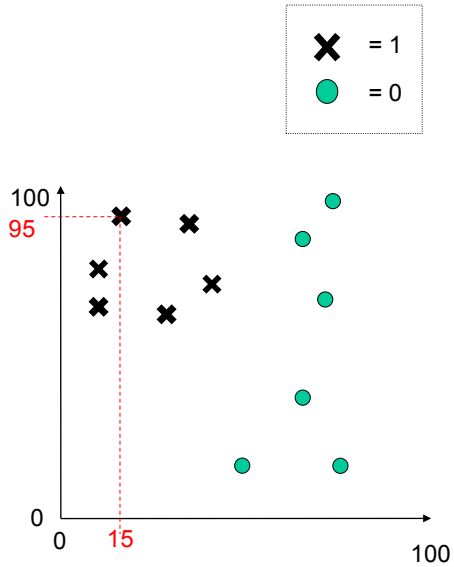


Inputs

Labels

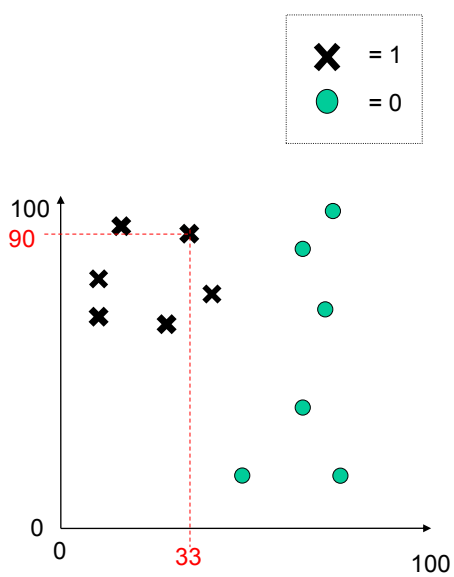


Training and Testing Data



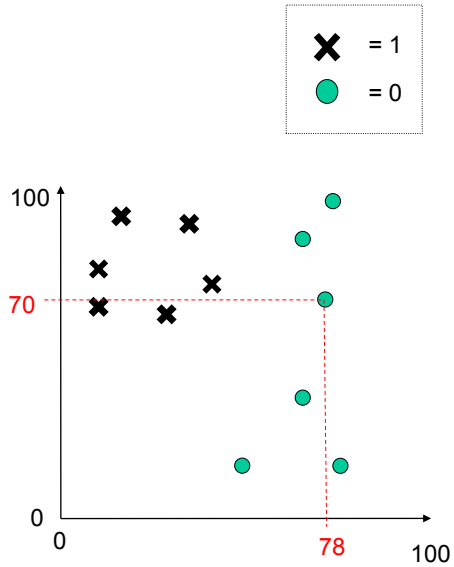
Inputs		Labels
15	95	1

Training and Testing Data



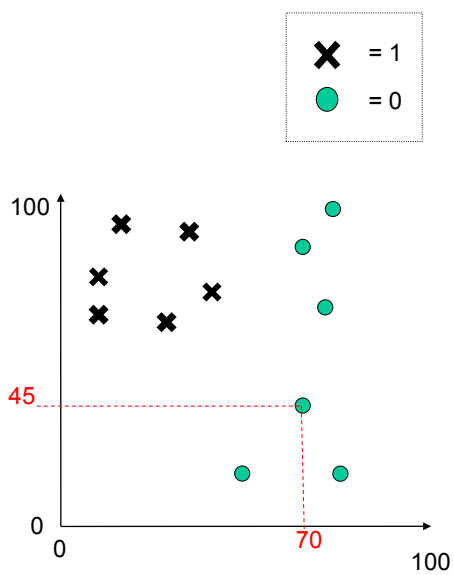
Inputs		Labels
15	95	1
33	90	1

Training and Testing Data



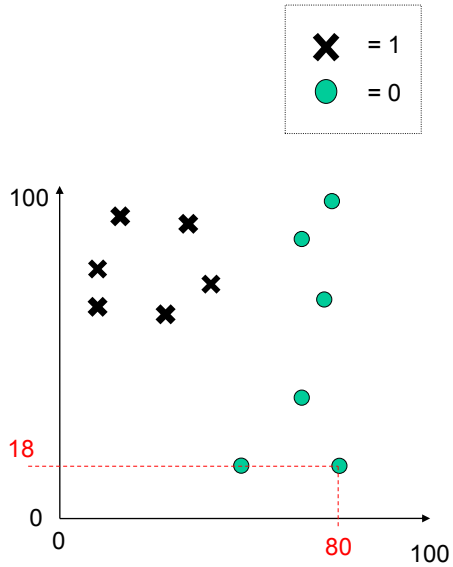
Inputs		Labels
15	95	1
33	90	1
78	**70**	**0**

Training and Testing Data



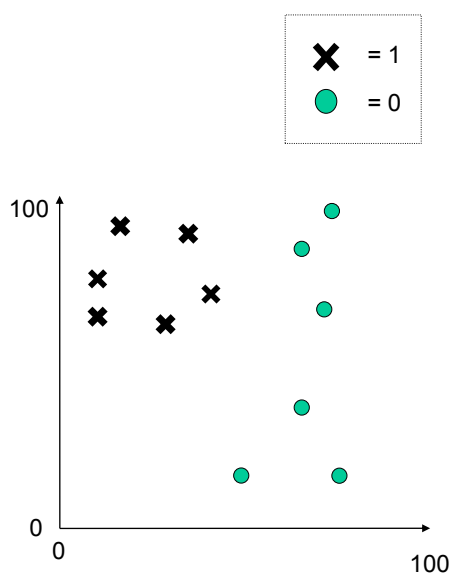
Inputs		Labels
15	95	1
33	90	1
78	70	0
70	**45**	**0**

Training and Testing Data



Inputs		Labels
15	95	1
33	90	1
78	70	0
70	45	0
80	**18**	**0**

Training and Testing Data



Inputs		Labels
15	95	1
33	90	1
78	70	0
70	45	0
80	18	0
35	65	1
45	70	1
31	61	1
50	63	1
98	80	0
73	81	0
50	18	0

2 “features”
12 “examples”
2 “classes”

Training and Testing Data

Dataset

Inputs		Labels
15	95	1
33	90	1
78	70	0
70	45	0
80	18	0
35	65	1
45	70	1
31	61	1
50	63	1
98	80	0
73	81	0
50	18	0

Training and Testing Data

Inputs		Labels
15	95	1
33	90	1
78	70	0
70	45	0
80	18	0
35	65	1
45	70	1
31	61	1
50	63	1
98	80	0
73	81	0
50	18	0

50:50 split

15	95	1
33	90	1
78	70	0
70	45	0
80	18	0
35	65	1

45	70	1
31	61	1
50	63	1
98	80	0
73	81	0
50	18	0

Training and Testing Data

If too small... you'll make many mistakes on the testing set.

Needs to be big. Bigger is better.

15	95	1
33	90	1
78	70	0
70	45	0
80	18	0
35	65	1

Training set

Train a K-NN on this…

45	70	1
31	61	1
50	63	1
98	80	0
73	81	0
50	18	0

Testing set

… then, test it on this!

"simulates" what it might be like to see new data in the future

Training and Testing Data

Training set

Build a k-NN using training set

Testing set

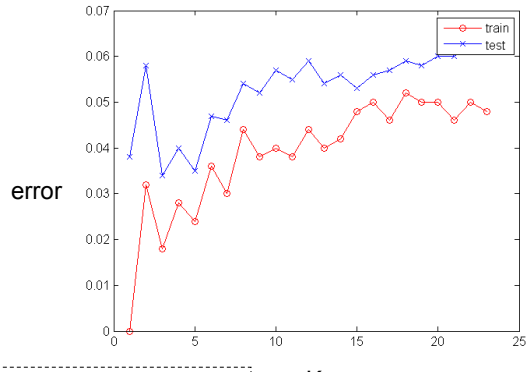
How many incorrect predictions on testing set?

Percentage of incorrect predictions is called the "error"

e.g. "Training" error
e.g. "Testing" error

Classifying '3' versus '8' Digits

Plotting error as a function of 'K' (as in the K-NN)

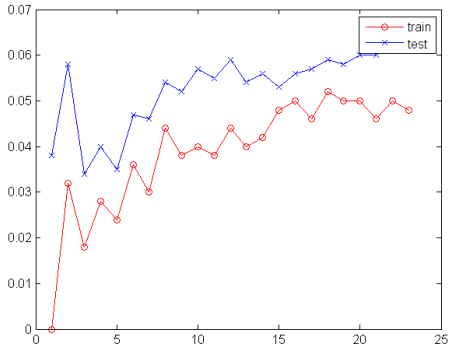


Percentage of incorrect predictions is called the "error"

e.g. "Training" error
e.g. "Testing" error

Training data can behave very differently to testing data!

Classifying '3' versus '8' Digits

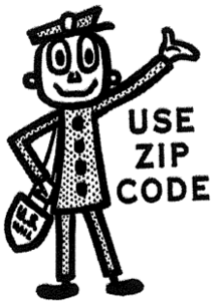


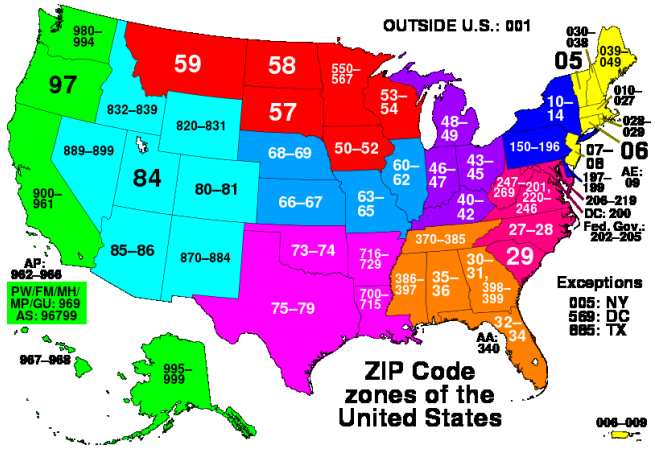
Best testing error at *K=3*, about 3.2%.

Is this "good"?

Classifying '3' versus '8' Digits

Zip-codes: "8" is very common on the West Coast, while "3" is rare.
Making a mistake will mean your Florida post ends up in Las Vegas!





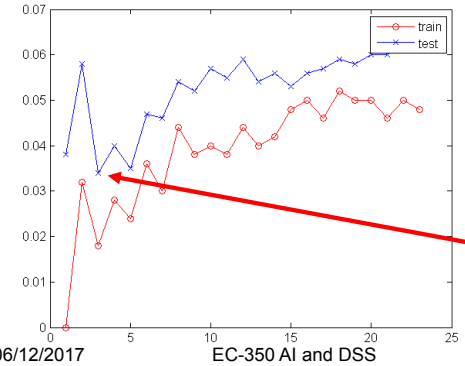
Classifying '3' versus '8' Digits

Sometimes, classes are rare, so your learner will not see many of them.

What if, in testing phase you saw 1,000 digits ….

32 instances

968 instances



3.2% error was achieved by just saying "8" to everything!

Classifying '3' versus '8' Digits

Solution?

32 instances

968 instances

0% correct

100% correct

Measure accuracy on each class separately.

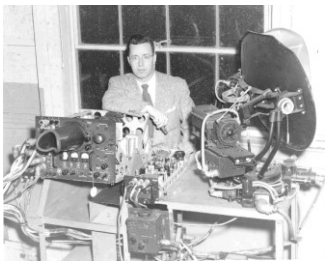
R.O.C. Analysis

32 instances 968 instances

A statistical framework.

Receiver Operator Characteristics
Developed in WW-2 to assess radar operators.

"How good is the radar operator at spotting incoming bombers?"

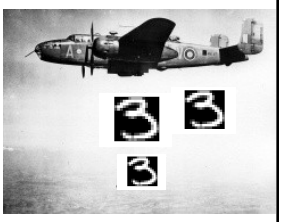


False positives
– i.e. falsely predicting a bombing raid

False negatives
– i.e. missing an incoming bomber (VERY BAD!)

R.O.C. Analysis

The "'3" digits are like the bombers.
Rare events but costly if we misclassify!



False positives – i.e. falsely predicting an event
False negatives – i.e. missing an incoming event

Similarly, we have "true positives" and "true negatives"

		Prediction	
		0	*1*
Truth	*0*	TN	**FP**
	1	**FN**	TP

Building a "Confusion Matrix"

		Prediction	
		0	*1*
Truth	*0*	TN	**FP**
	1	**FN**	TP

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

… chances of spotting a "3" when presented with one
(i.e. accuracy on class "3")

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

… chances of spotting an 8 when presented with one
(i.e. accuracy on class "8")

R.O.C. Analysis

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

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

Prediction

Truth	*0*	*1*
0	60	**30**
1	**80**	20

TN	**FP**
FN	TP

60+30 = 90 examples in the dataset were class 0

80+20 = 100 examples in the dataset were class 1

90+100 = 190 examples in the data overall

Assignment # 3

- Use K-NN to recognise handwritten USPS digits.
- You will give the printed report including
  - *Code*
  - *Results (Figures, Graphs, Tables etc.)*
  - *Your description (Max. 1 page)*
- Deadline for submission
  - *After 2 weeks*