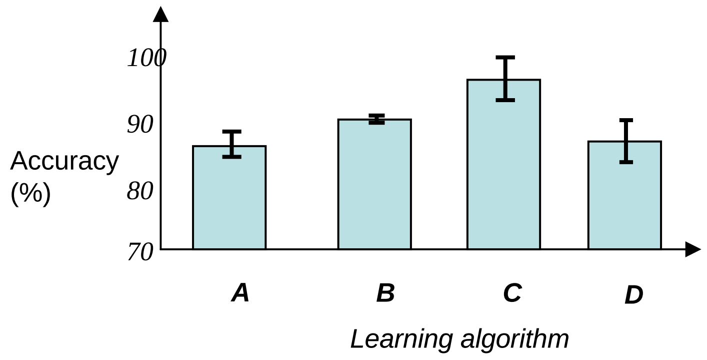
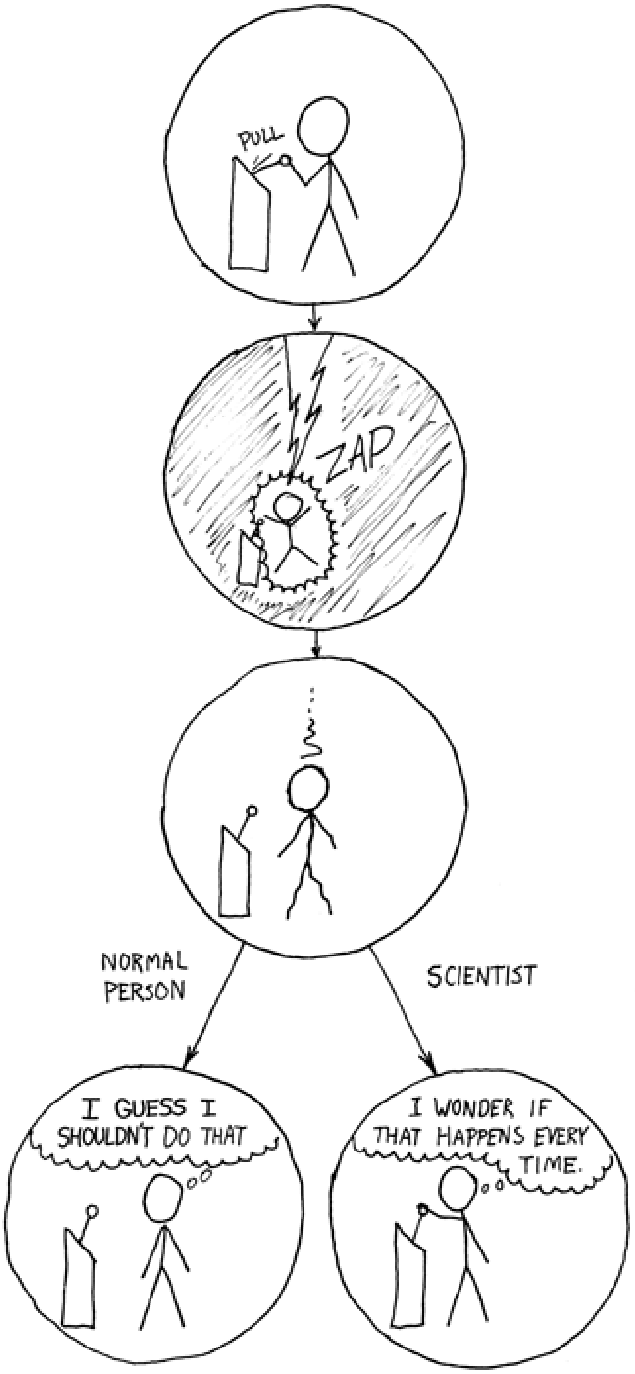
COMP24111 lecture 5

Experiments in Machine Learning



Scientists
vs
Normal People


PULL
ZAP
NORMAL PERSON
SCIENTIST
I GUESS I SHOULDN'T DO THAT
I WONDER IF THAT HAPPENS EVERY TIME.

Learning Algorithm

1. Split our data **randomly**

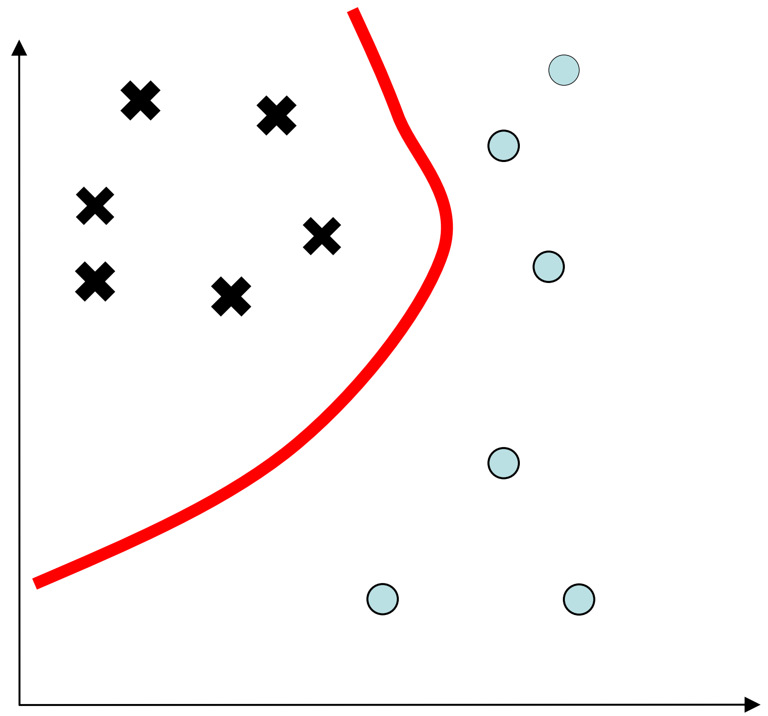
2. Train a model…

3. Test it!

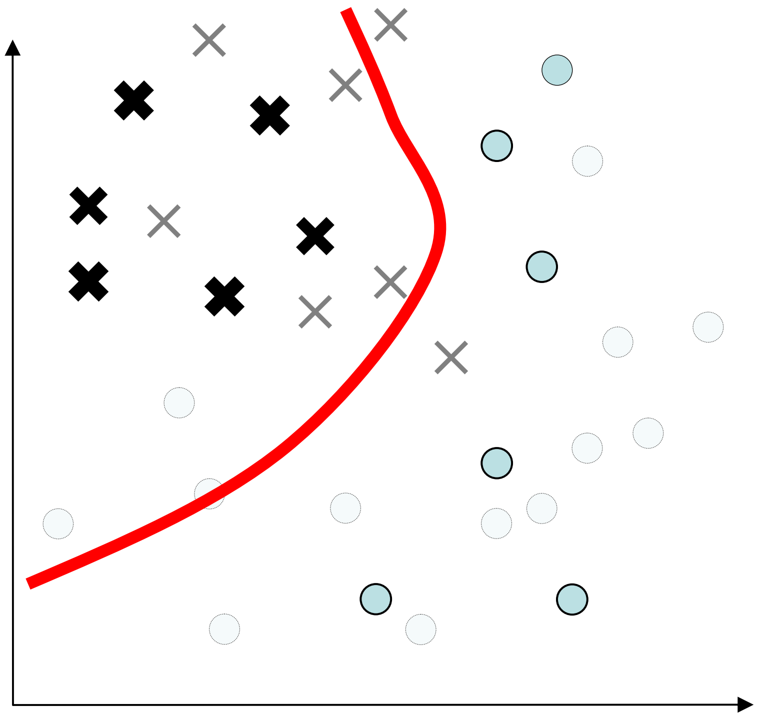
Why do we split the data?

The **Most Important** Concept in Machine Learning…

Looks good so far…



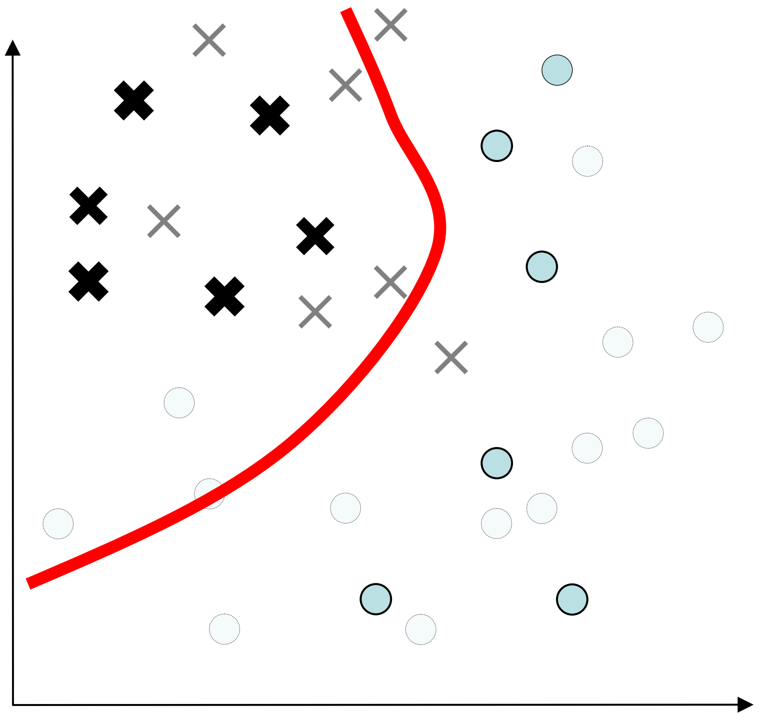
The **Most Important** Concept in Machine Learning…



Looks good so far…

Oh no! Mistakes!
What happened?

The **Most Important** Concept in Machine Learning…



Looks good so far…

Oh no! Mistakes!
What happened?

We didn't have all the data.

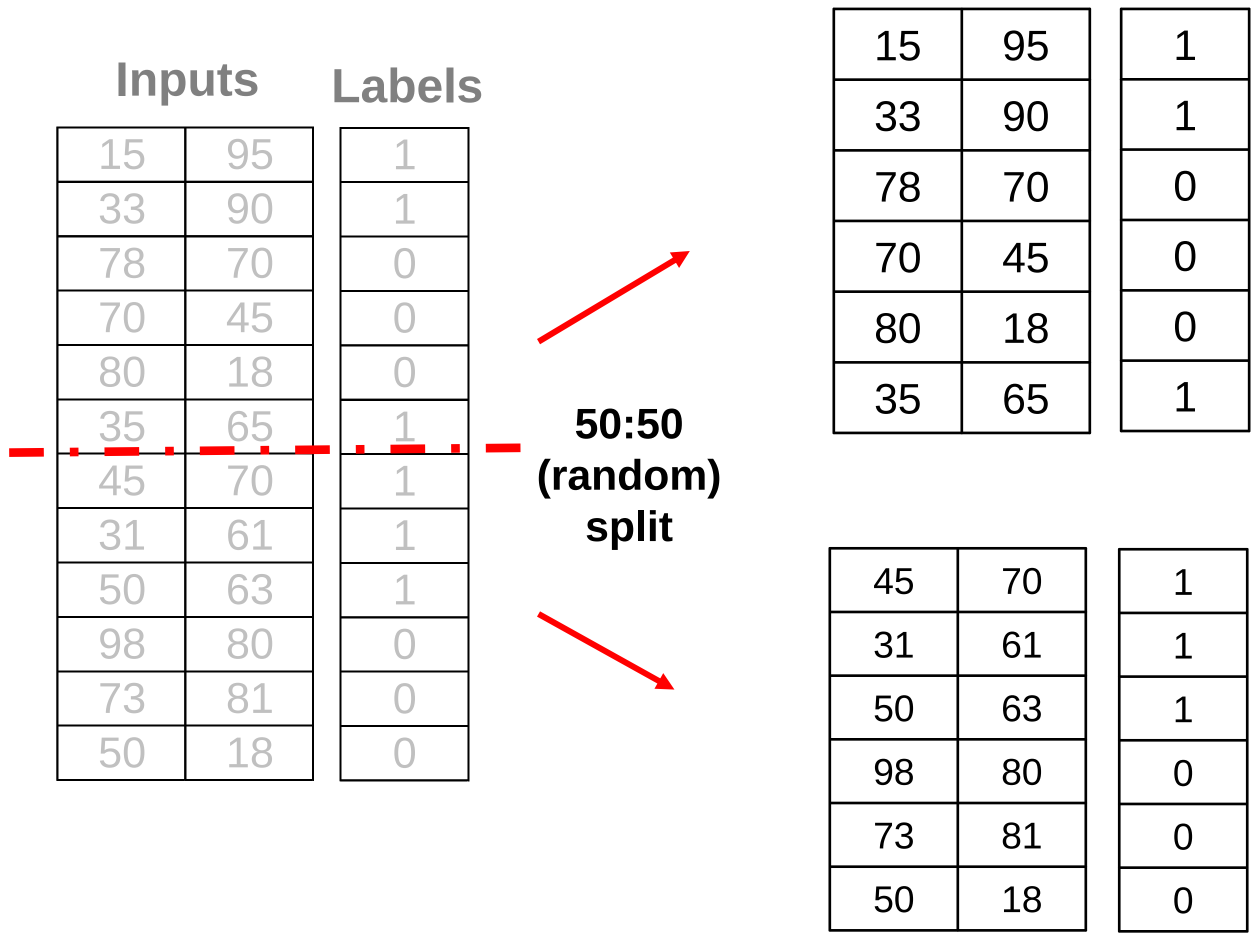
We can never assume that we do.

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

Inputs **Labels**

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

1
1
0
0
0
1
1
1
1
0
0
0



Inputs

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

Labels

1
1
0
0
0
1
1
1
1
0
0
0

50:50 (random) 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

Ideally should be small.
Smaller is better.

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

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

Training set

Train a K-NN or Perceptron on this…

Needs to be quite big.
Bigger is better.

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 set

Build a model (knn, perceptron, decision tree, etc)

Testing set

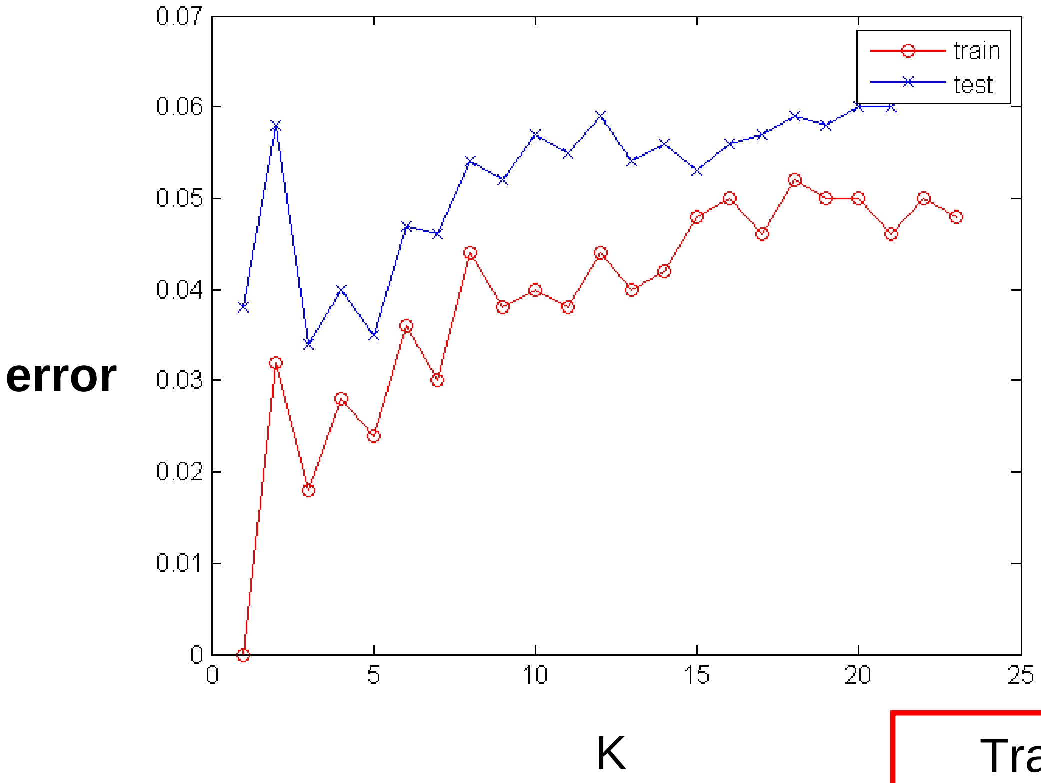
How many incorrect predictions on testing set?

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)



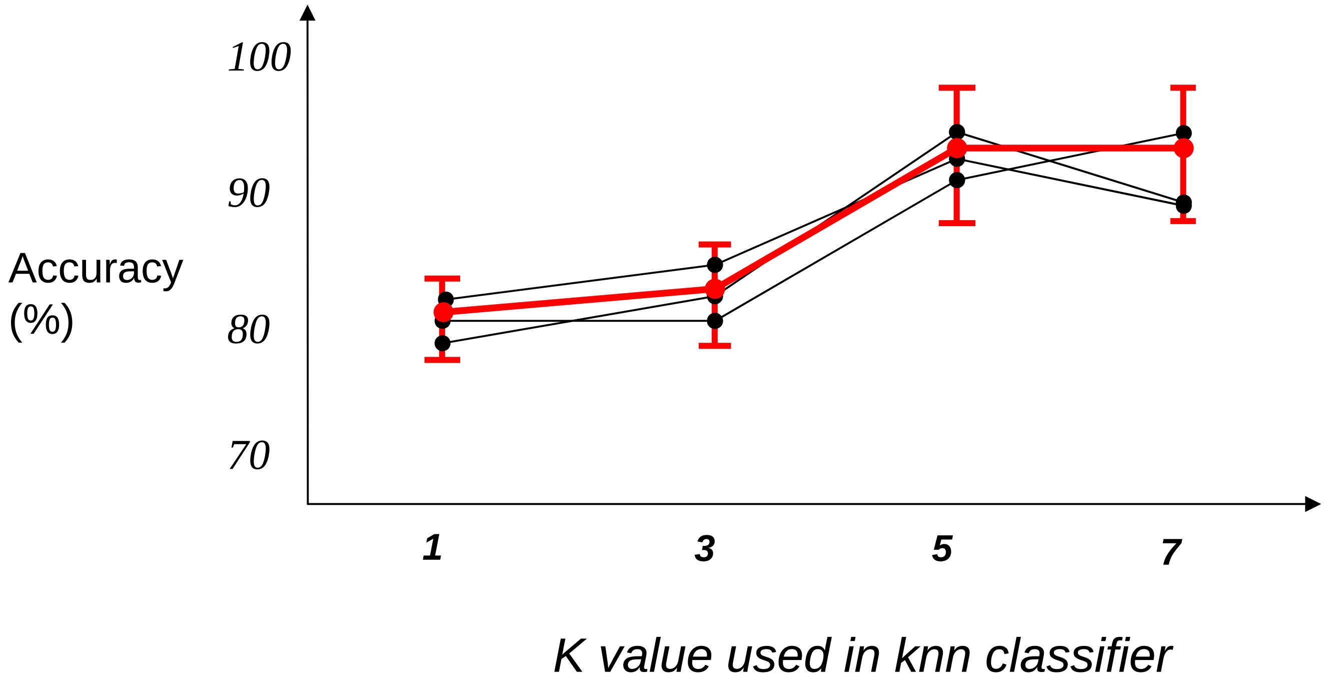
Percentage of incorrect predictions is called the "error"

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

Training data can behave very differently to testing data!

Which is the “best” value of k?

Different random split of the data
= different performance!

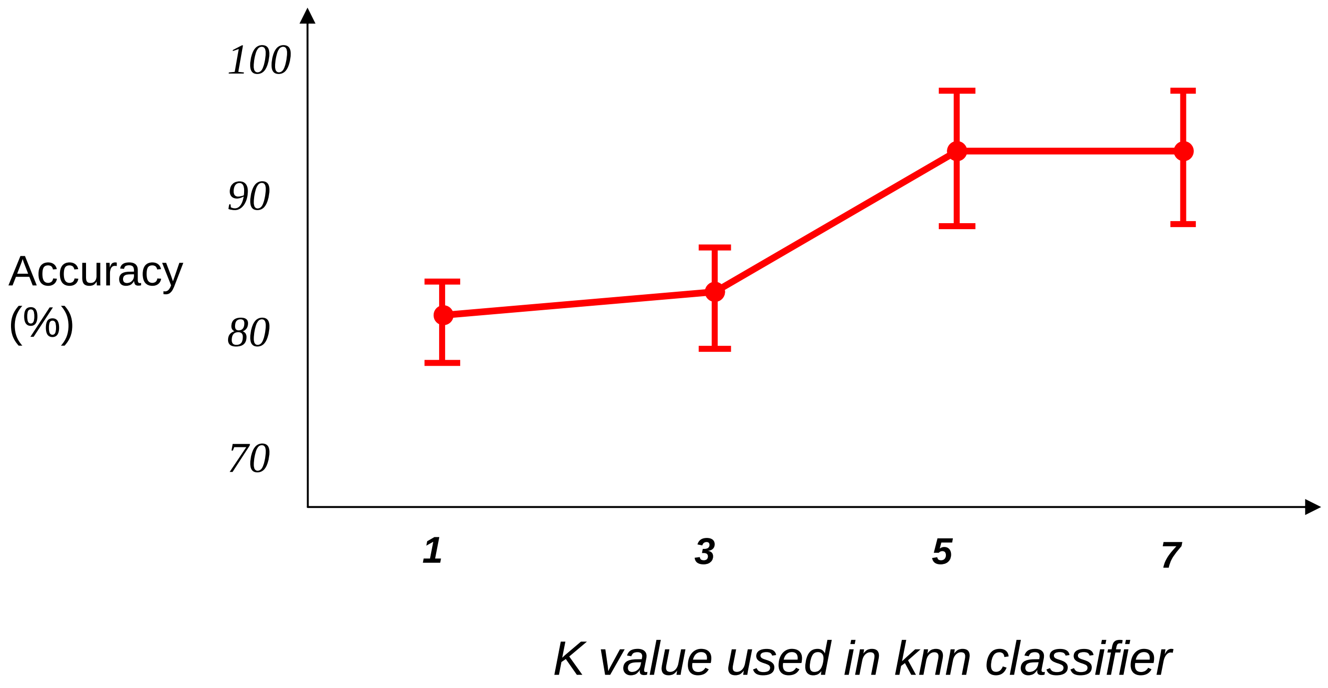


MATLAB: “help errorbar”

Error bars (a.k.a. “confidence intervals”)

Plots the average, and the standard deviation (spread) of values.

Wider spread = less stability….



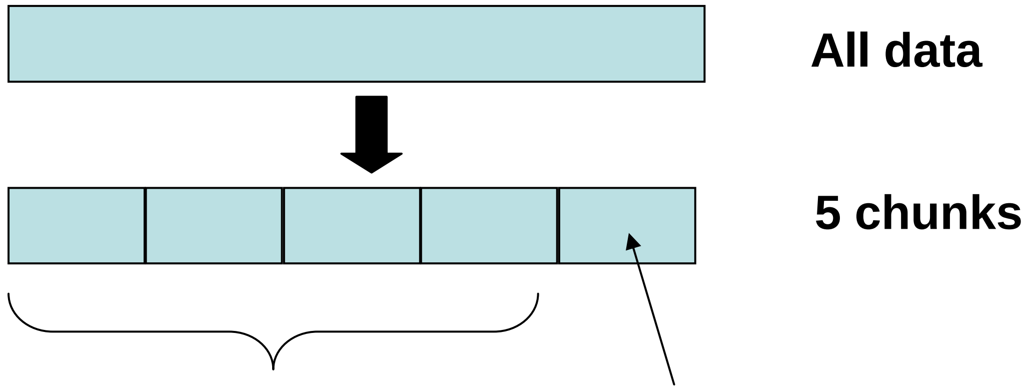
`MATLAB: “help errorbar”`

MATLAB: “help crossfold”

Cross-validation

We could do repeated random splits… but there is another way…

```
1. Break the data evenly into N chunks
2. Leave one chunk out
3. Train on the remaining N-1 chunks
4. Test on the chunk you left out
5. Repeat until all chunks have been used to test
6. Plot average and error bars for the N chunks
```



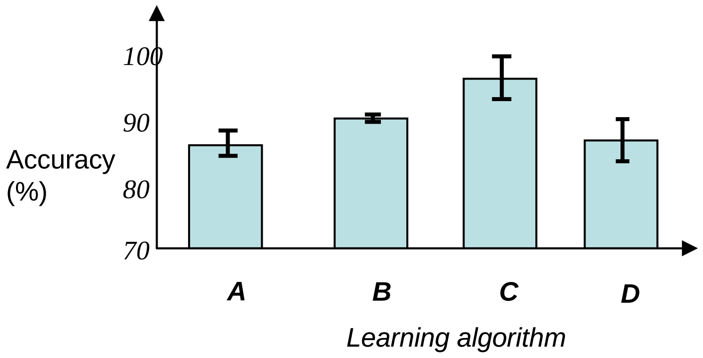
Train on these (all together)

Test on this

… then repeat, but test on a different chunk.

What factors should affect our decision?

1. *Accuracy*
   - How accurate is it?
2. *Training time and space complexity*
   - *How long does it take to train?*
   - *How much memory overhead does it take?*
3. *Testing time and space complexity*
   - *How long does it take to execute the model for a new datapoint?*
   - *How much memory overhead does it take?*
4. *Interpretability*
   - *How easily can we explain **why** it does what it does?*



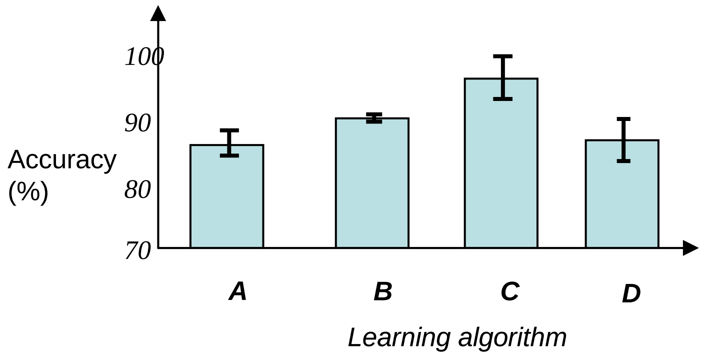
Measuring accuracy is not always useful…

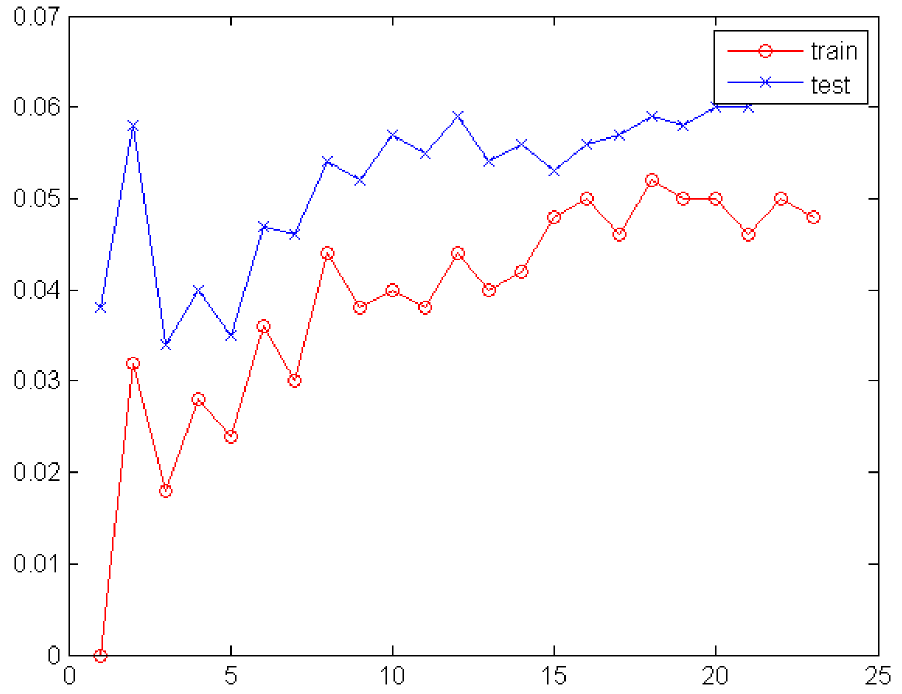
1. ***Accuracy***
   - ***How accurate is it?***

2. *Training time and space complexity*
   - *How long does it take to train?*
   - *How much memory overhead does it take?*

3. *Testing time and space complexity*
   - *How long does it take to execute the model for a new datapoint?*
   - *How much memory overhead does it take?*

4. *Interpretability*
   - *How easily can we explain **why** it does what it does?*

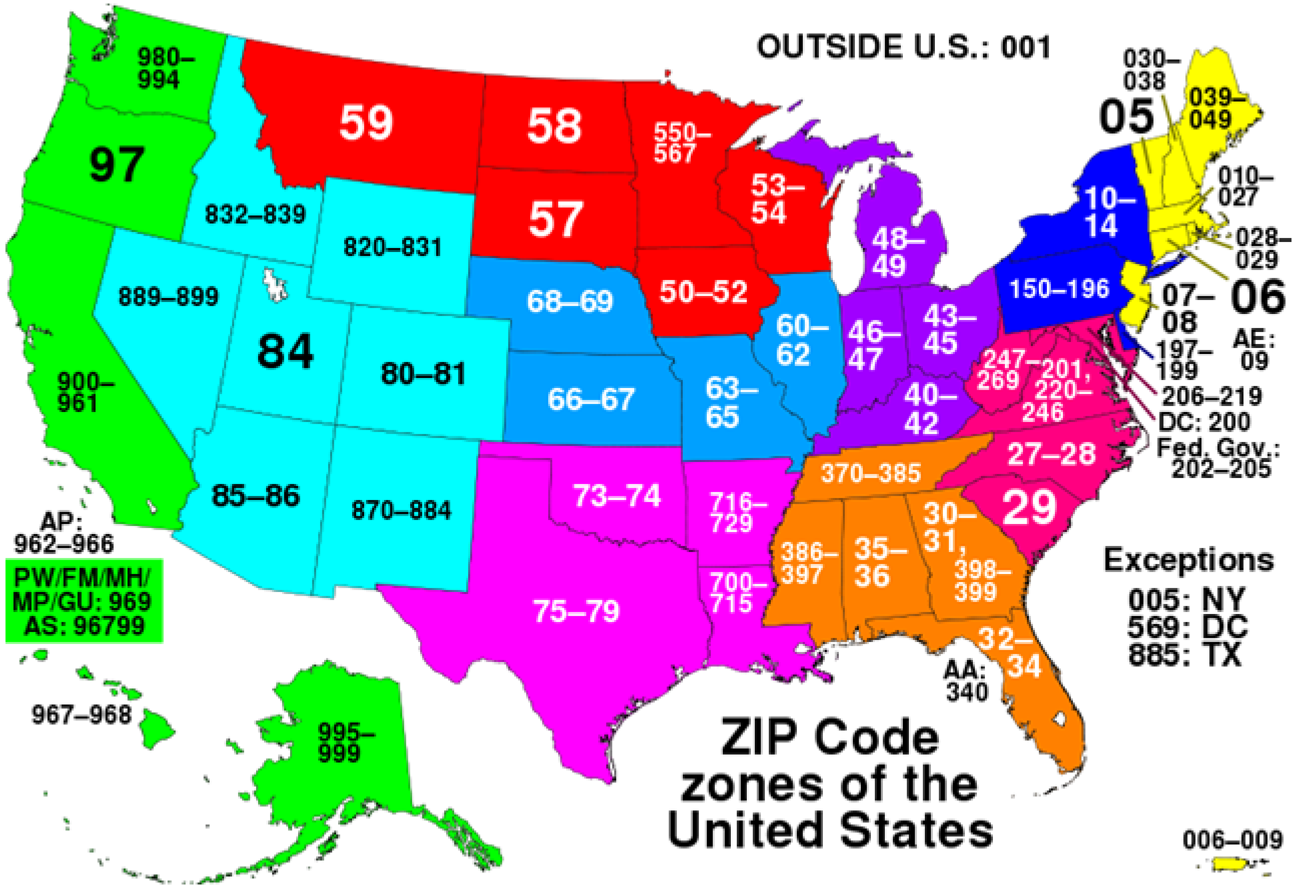




Best testing error at *K=3*, about 3.2%.

Is this “good”?

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!

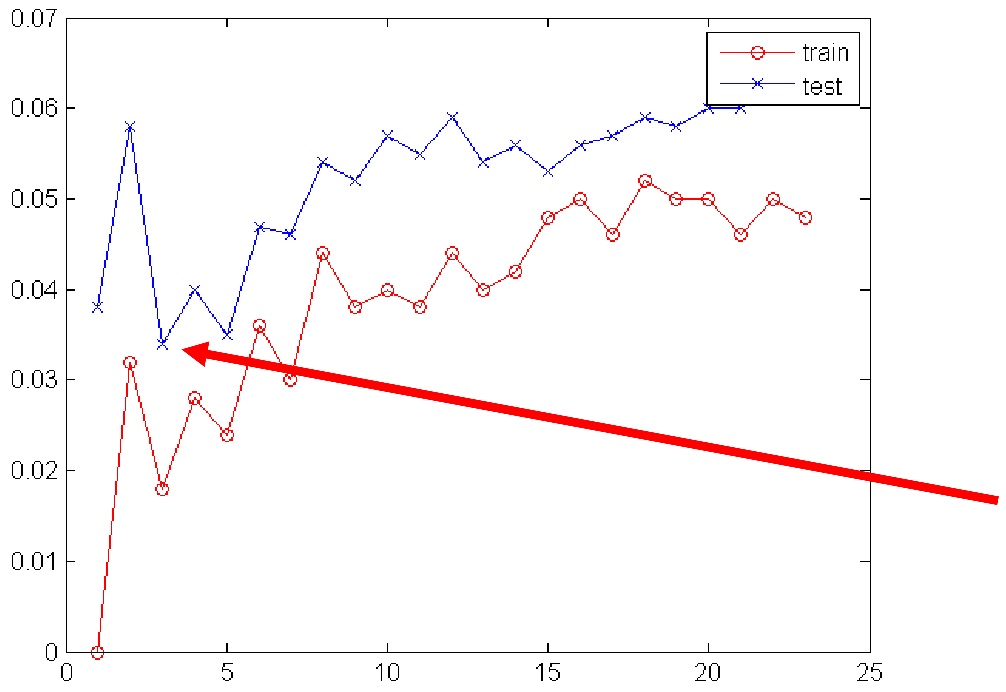


Sometimes, classes are rare, so your learner will not see many of them.

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!

Solution?

32 instances 968 instances

0% correct **100% correct**

Measure accuracy on each class separately.

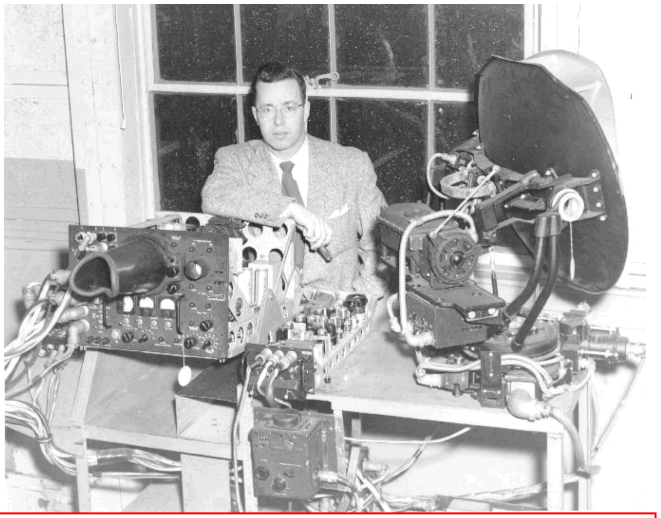
ROC analysis

32 instances 968 instances

Our obvious solution is in fact backed up a whole statistical framework.

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

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

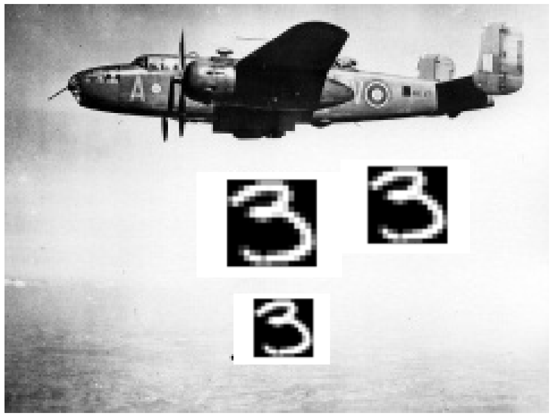


False positives
– i.e. falsely predicting a bombing raid

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

ROC 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

$$\begin{array}{cc|cc} & & \multicolumn{2}{c}{\textit{Prediction}} \\ & & 0 & 1 \\ \hline \textit{Truth} & 0 & \text{TN} & \textbf{FP} \\ & 1 & \textbf{FN} & \text{TP} \end{array}$$

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

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

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

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

ROC analysis… your turn to think…

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

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

		Prediction	
		0	*1*
Truth	*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

ROC analysis – incorporating costs…

FP = total number of **false positives** I make on the testing data

FN = total number of ***false negatives*** I make on the testing data

$$\text{Cost of my classifier} = \left[FN \times \text{cost}(FN)\right] + \left[FP \times \text{cost}(FP)\right]$$

Useful if one is more important than another. For example if a predictor…

… misses a case of a particularly horrible disease **(FALSE NEGATIVE)**

or

… sends a patient for painful surgery when it is unnecessary **(FALSE POSITIVE)**

Experiments in Machine Learning

- **Learning algorithms depend on the data you give them**

  - some algorithms are more “stable” than others
  - must take this into account in experiments
  - cross-validation is one way to monitor stability
  - plot confidence intervals!

- **ROC analysis can help us if…**

  - there are imbalanced classes
  - false positives/negatives have different costs