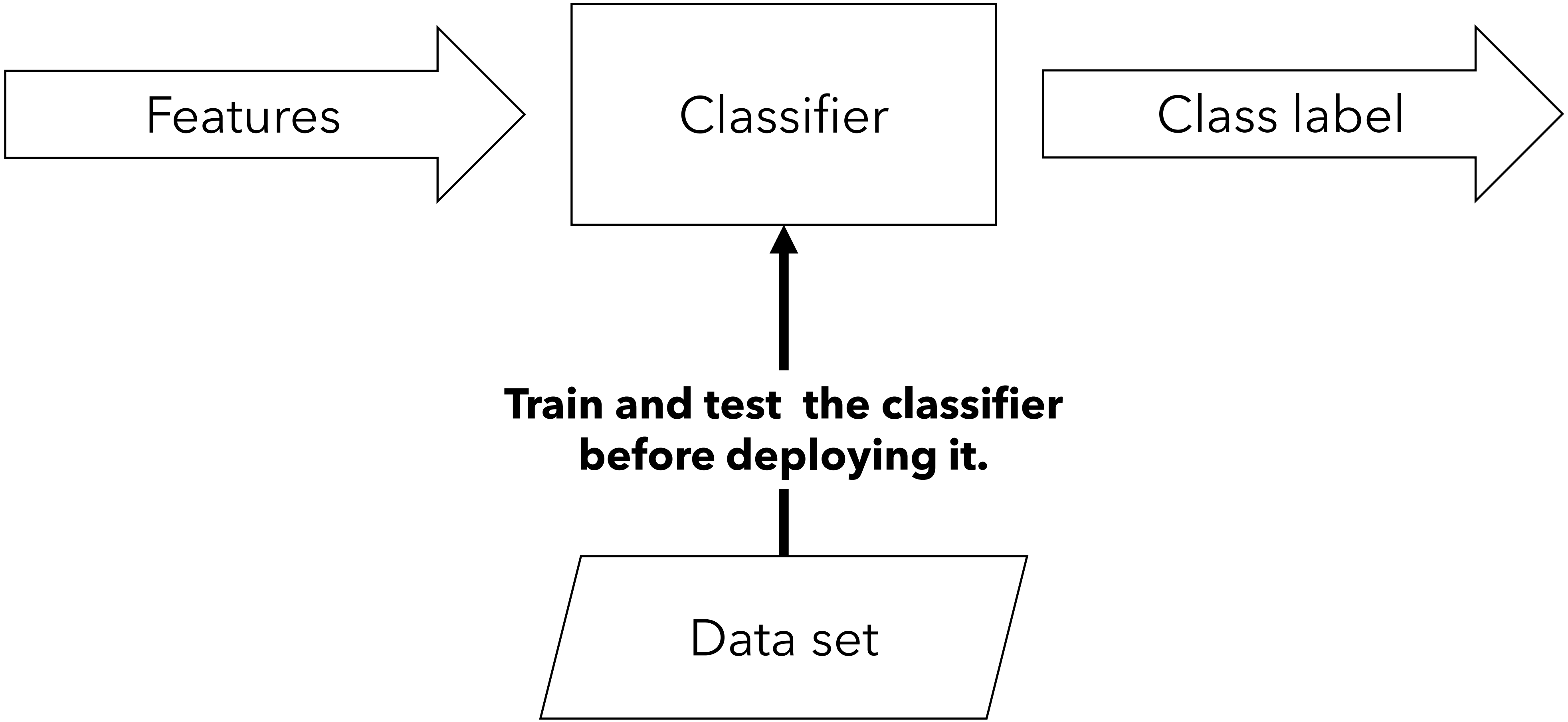
CSE 435 Pattern Recognition

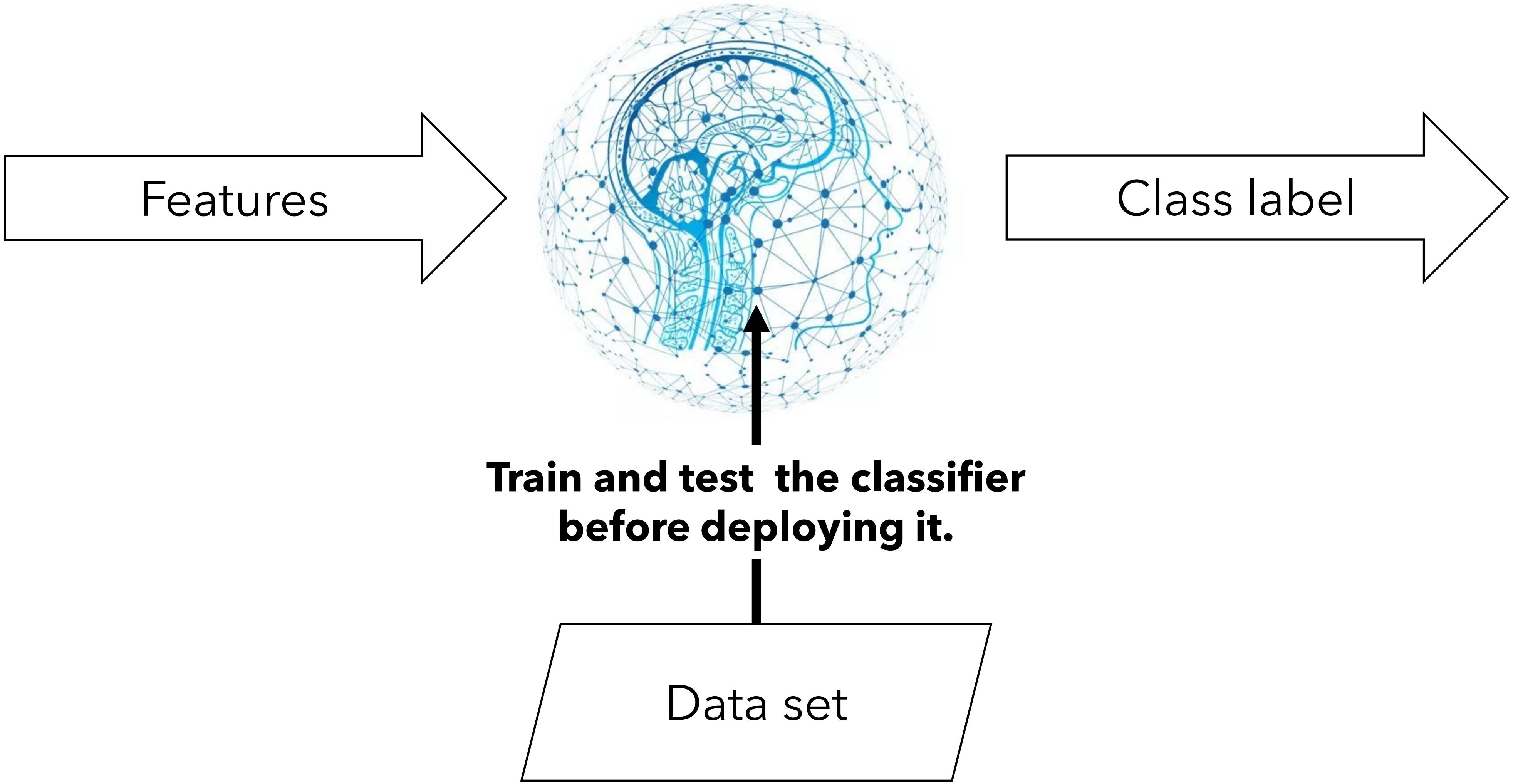
Training and testing of a classifier

Tanvir Azhar
Lecturer, EDU

Training and testing of a classifier

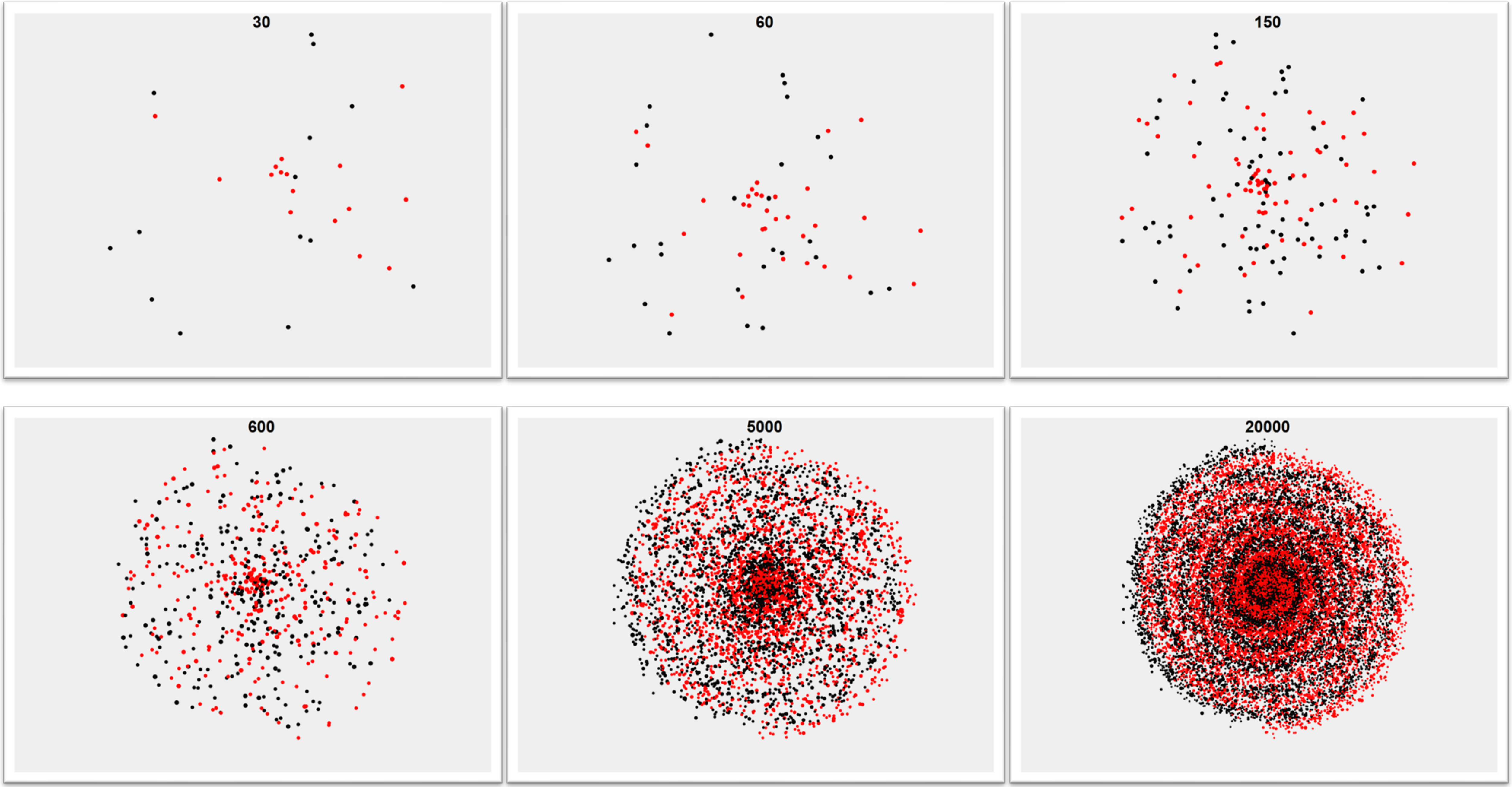
- More data = better classifier
- Overfitting and generalisation
- Training/testing protocols


Features
Classifier
Class label
Train and test the classifier before deploying it.
Data set

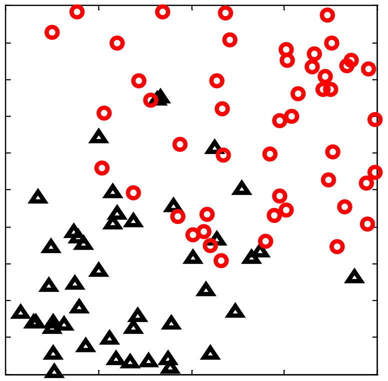

Features
Class label
Train and test the classifier before deploying it.
Data set

Bigger is better!

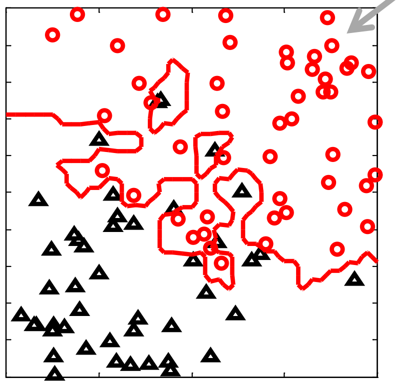
The more data we have, the better the classifier will be.


30
60
150
600
5000
20000

Data



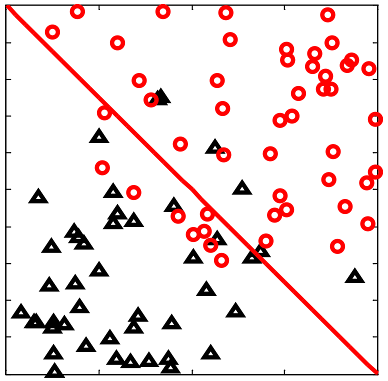
Overfitting



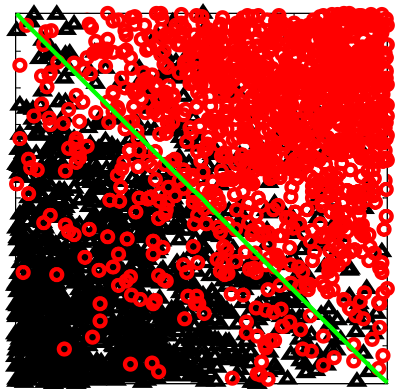
The classifier memorises the noise in the data.

Generalisation:
The ability of the classifier to handle unseen data

Optimal boundary

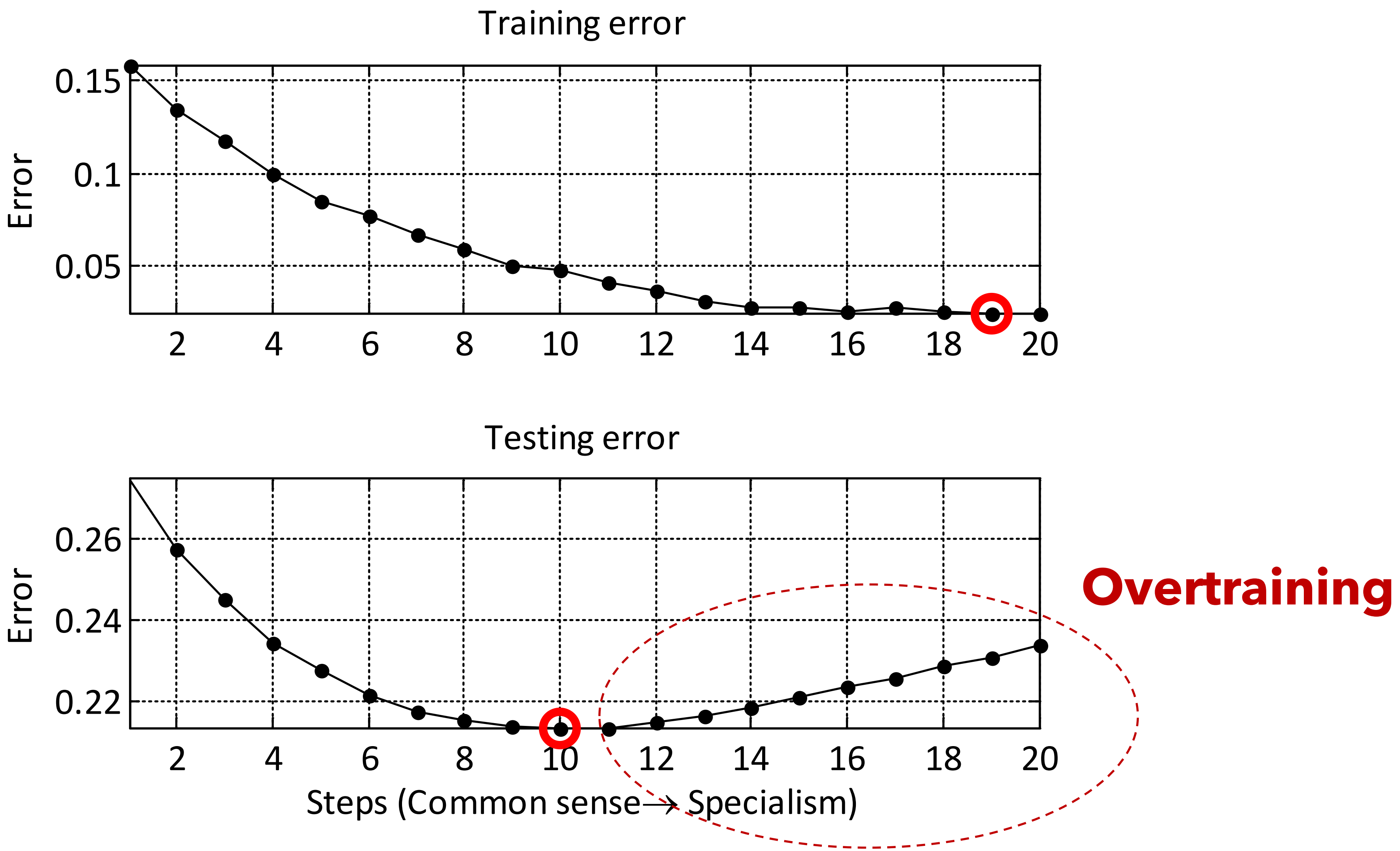


Further data from the same distribution



How can we avoid overfitting and improve generalisation?

Typical training pattern of a Neural Network

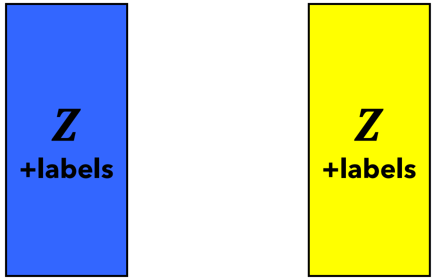


Training

Testing

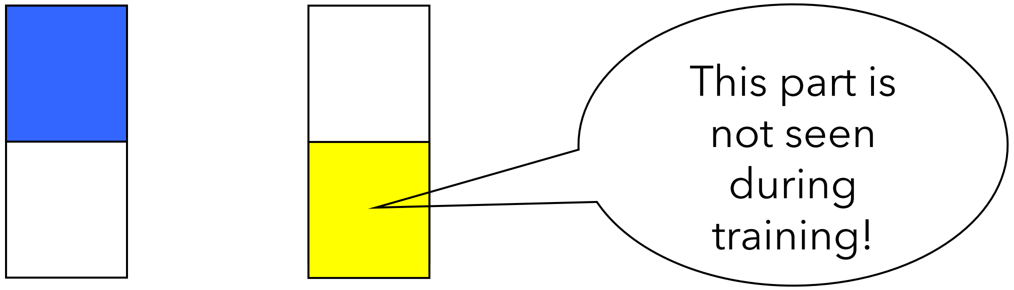
Method

Resubstitution (R-method)
= “test on the training data”

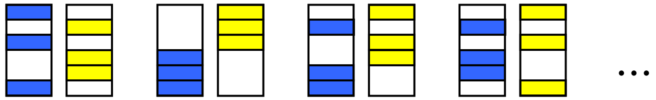


The probability of error is called the Apparent Error Rate

Hold-out (H-method)
= “split into two”
usually in random halves



Data shuffle
= “split into two + repeat”



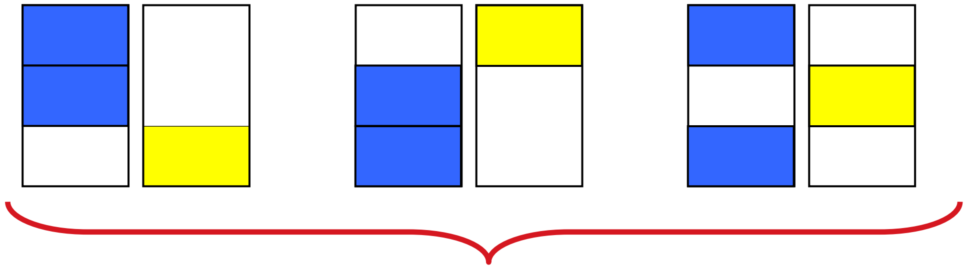
Repeat L times and average the L testing error estimates.
Typical choices: $L = 100$; the split is 90% for training and 10% for testing.

Training

Testing

Method

Cross-validation



3-fold cross-validation

Good things:

testing on non-intersecting sets,

testing on unseen data,

total testing sample size = N

Bad thing:

Testing sets may become too small and the estimate of the error may be unreliable.

Training

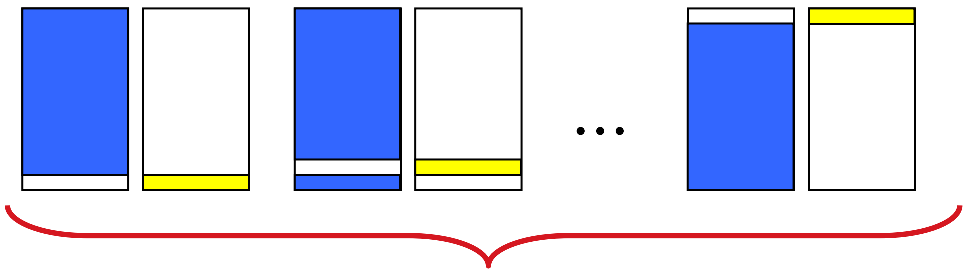
Testing

Method

Leave-one-out:

A special case of cross-validation

= N-fold cross-validation



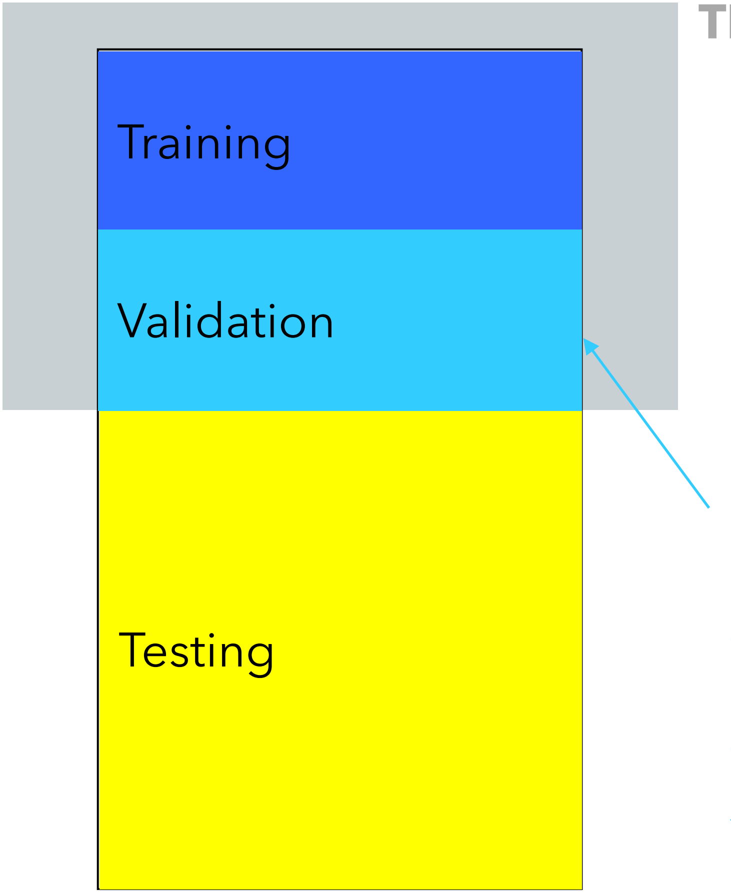
N times, leave one object aside
for testing

Extra good thing:

training on a large data set (almost the whole of $\boldsymbol{Z}$)
$\Rightarrow$ better training, better classifier

How can we avoid overfitting?

Split the data set into 3 parts



TRAINING PART

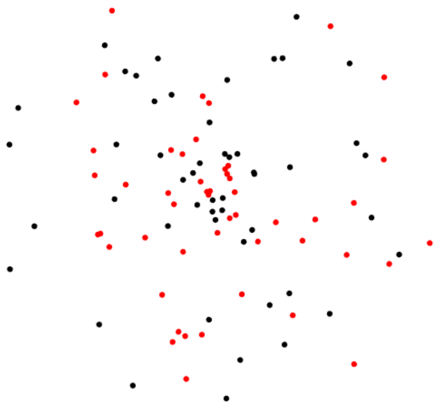
Use the Validation set to monitor how the training is going and for tuning extra parameters. Spot OVERFITTING there and stop training.

Example

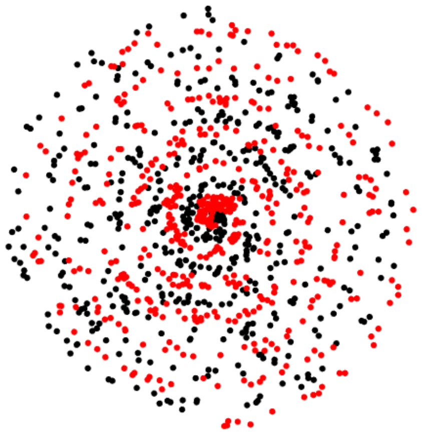
Two-spirals data

Ideal data (no noise)

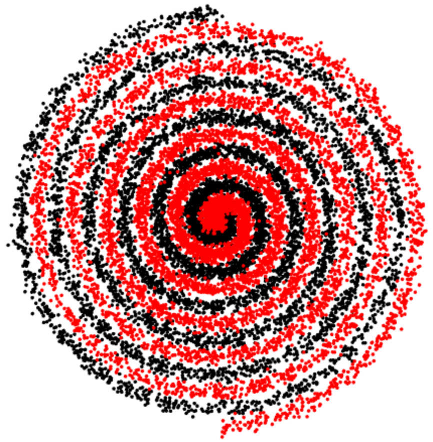
Ideal data, noisy step



Small sample (N=100)
Noise 0.02 (for radius 1)



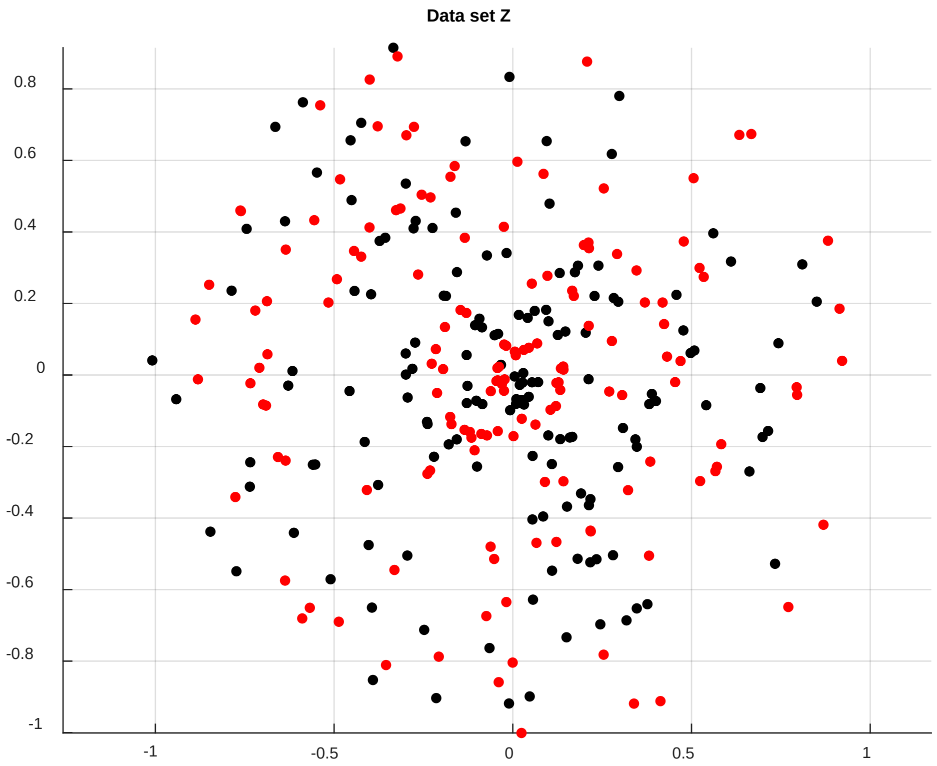
Medium sample (N=1,000)
Noise 0.02 (for radius 1)

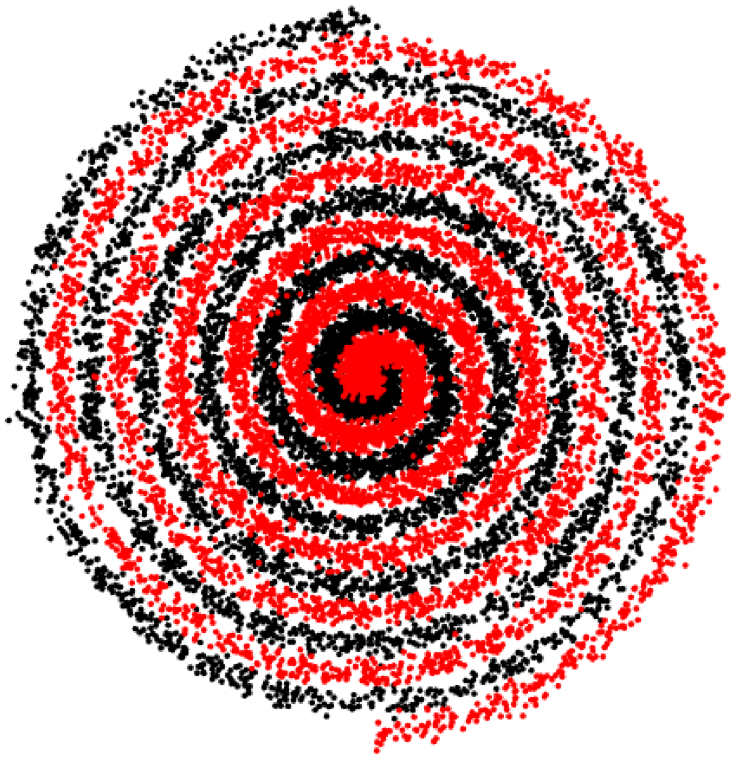


Large sample (N=15,000)
Noise 0.02 (for radius 1)

Example

Two-spirals data



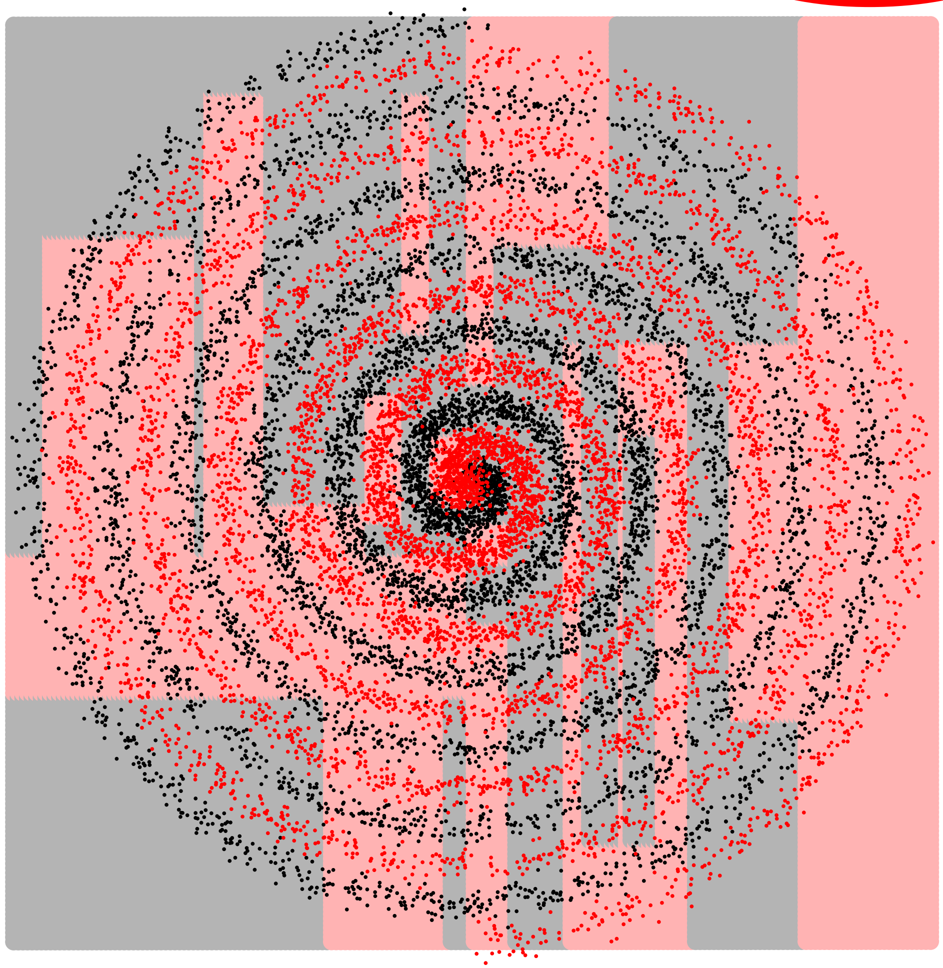


This is the data set Z from which we will cut training and testing parts.

This will serve as the REAL-LIFE data where we expect our classifier to work

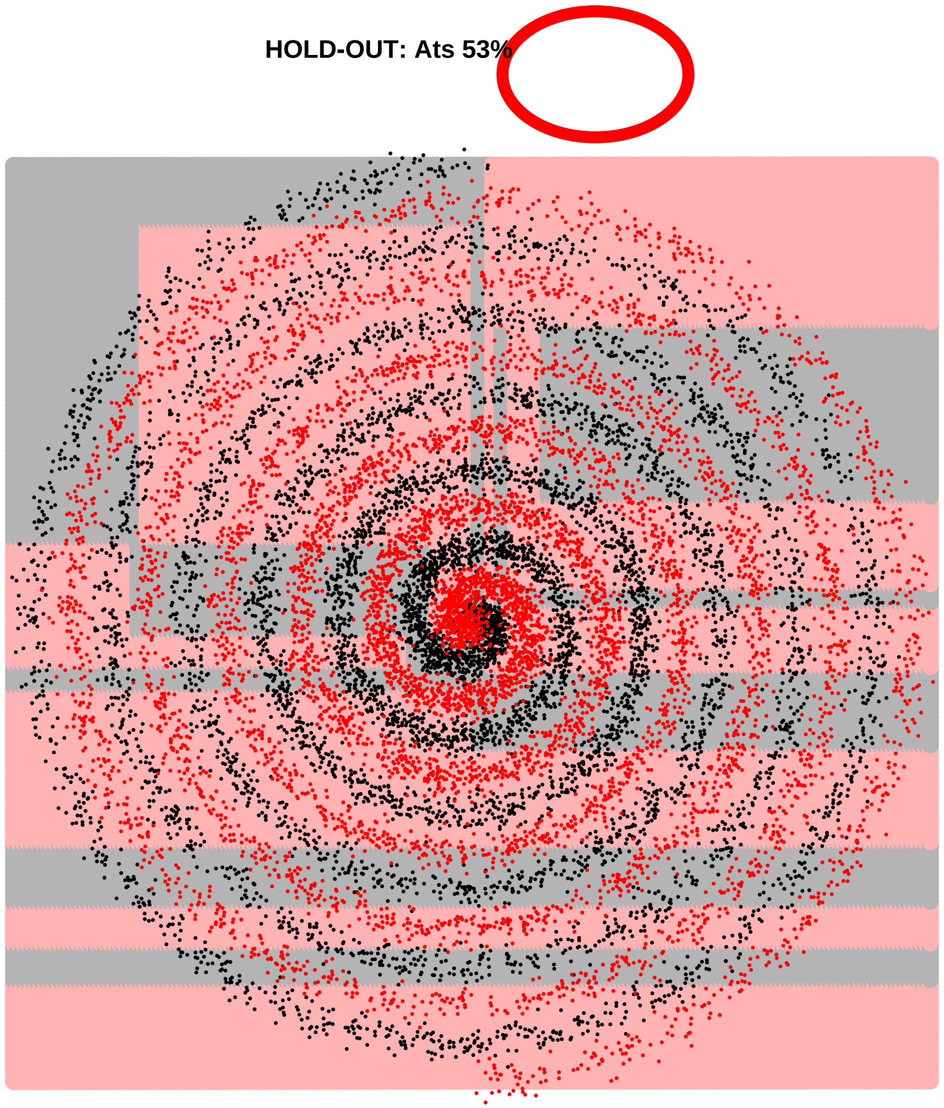
RESUBSTITUTION: Atr 84%, Ats 84% A[True] = 57%

This is what we would like to predict and return to the user as an estimate.



Train a **decision tree classifier**
(we'll see this model later)

Training and testing data are the same – all of Z.



Training on a half of Z and testing on the other half of Z.

Recall that if we train the tree classifier **on the whole of** Z, A[True] was **57%**. Hence Ats is somewhat pessimistic. But this way we are not misleading our user!

10-FOLD CROSS-VALIDATION

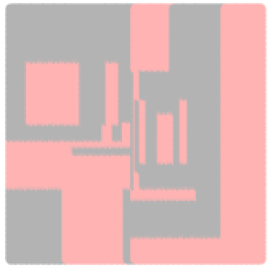
Ats = 57%

Fold 1, Ats 60%

Fold 2, Ats 73%

Fold 3, Ats 43%

Fold 4, Ats 53%



Fold 5, Ats 53%

Fold 6, Ats 57%

Fold 7, Ats 57%

Fold 8, Ats 53%

Fold 9, Ats 63%

Fold 10, Ats 53%

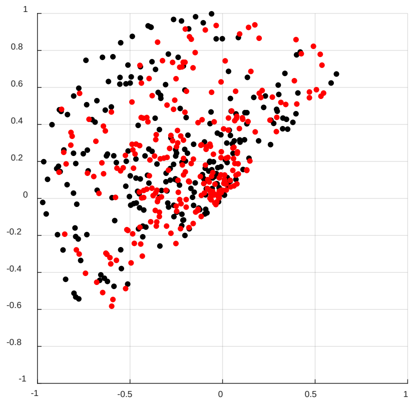
On average, closer to A[True] than Ats of hold-out. But quite variable for different splits! This is why people use 10 times 10-fold CV

Leave-one-out Ats = 55%

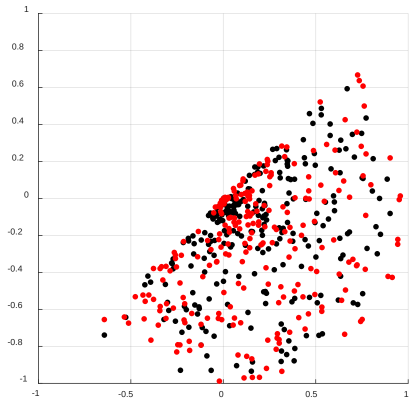
Q1. Which of the two splits of training/testing of Z would you prefer and why?

Split 1

Training

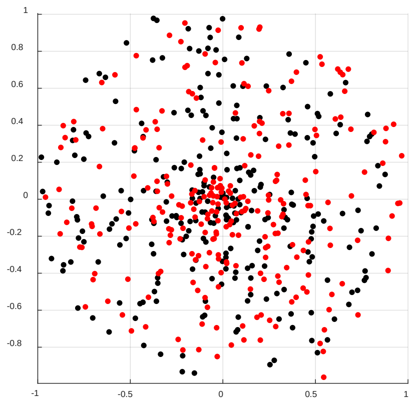


Testing

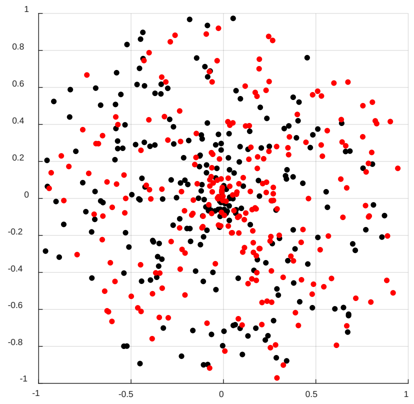


Split 2

Training



Testing



Q2. How many calculations of the testing error do you need to carry out if you run 5 times 10-fold cross-validation on a data set with $N = 1500$ objects, $c = 3$ classes and $n = 4$ features?

Q3. Given is the following 1D data set Z

x	-6	-2	0	4	11	15	24
label	1	2	2	1	2	1	1

Suppose that you are training a classifier D in the following way: find the average of all objects from class1 (call it a_1) and the average of all objects from class 2 (a_2). Then set up a threshold at t $= \frac{a_1+a_2}{2}$. Assign all objects greater than t to class 1, and less than t, to class 2.

(a) Apply the **resubstitution** protocol and evaluate the classification error of D.

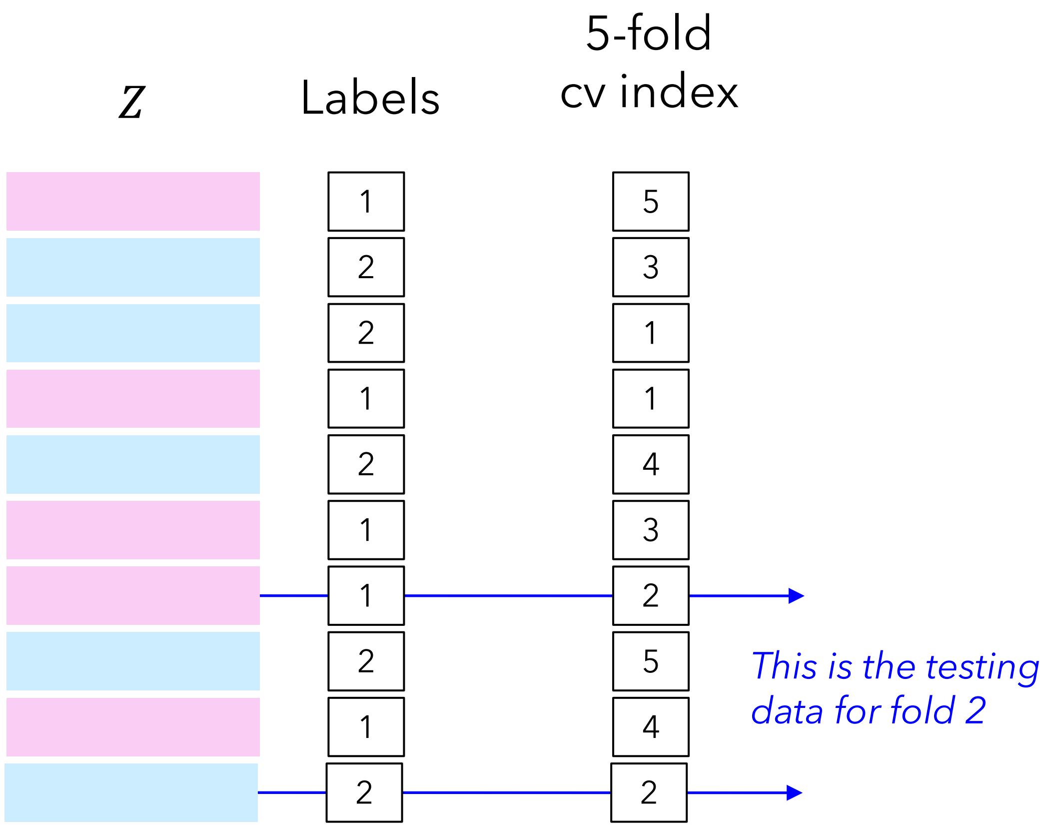
(b) Apply the **leave-one-out** protocol and evaluate the classification error of D.
(It is best to write a small piece of code for this but it is also doable by hand.)

Q4 Without using the ready-made functions for splitting a data set, write Python code to do the following:

(a) Generate a data set with two Gaussian classes in 2D, one with mean (0,0) and another with mean (1,1). Each class should have 100 objects.

(b) Split the data into two random halves, each containing 100 objects. Remember that you need to keep the labels too! The labels must correspond to the objects as in the original sample.

(c) Generate a column of indices for a 5-fold cross-validation. The column should contain values from 1 to 5, indicating the *folds* (not class labels). See the example on the right.



Answers to some questions:

Q1. You should prefer the second split. In the first split, the training and the testing data are very different. Whatever can be learned from the training data will not be applicable to the training data without further assumptions and recalculations. Neither the training data nor the testing data are representative of the full data set. This problem does not exist in Split 2.

Q2. I never promised I will be like Who-Wants-To-Be-A-Millionaire? I **will** give you trick questions now and then. In this case, "N = 1500 objects, c = 3 classes and n = 4 features" don't matter at all. The number of evaluations of the testing accuracy is 5x10 = 50.

Q3 (a). Average for class 1: $a_1 = \frac{-6+4+15+24}{4} = 9.25$. Average for class 2: $a_2 = \frac{-2+0+11}{3} = 3$. Then the threshold is $t = \frac{9.25+3}{2} = 6.125$. The assigned labels will be as below (errors are marked with x):

x	-6	-2	0	4	11	15	24
label	1x	2	2	1x	2x	1	1

Class 2 **Class 1**

Resubstitution error rate $= \frac{3}{7} = 42.86\%$