Lecture 13
Machine Learning
KNN

The Major Machine Learning Tasks

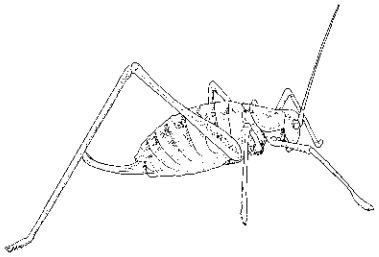
- Classification
- Clustering

Most of the other tasks (for example, outlier discovery or anomaly detection) make heavy use of one or more of the above.

So in this tutorial we will focus most of our energy on the above, starting with…

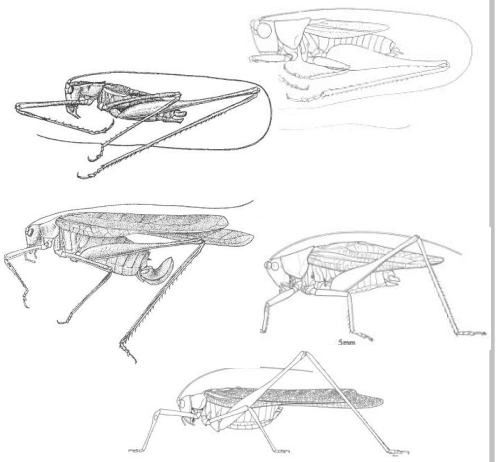
The Classification Problem
(informal definition)

Given a collection of annotated data. In this case 5 instances of **Katydids** and five of **Grasshoppers**, decide what type of insect the unlabeled example is.

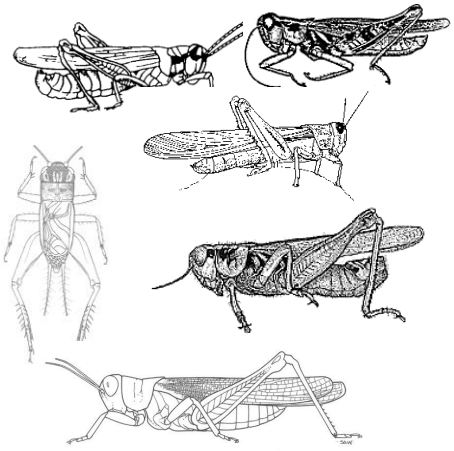


Katydid or **Grasshopper**?

Katydids



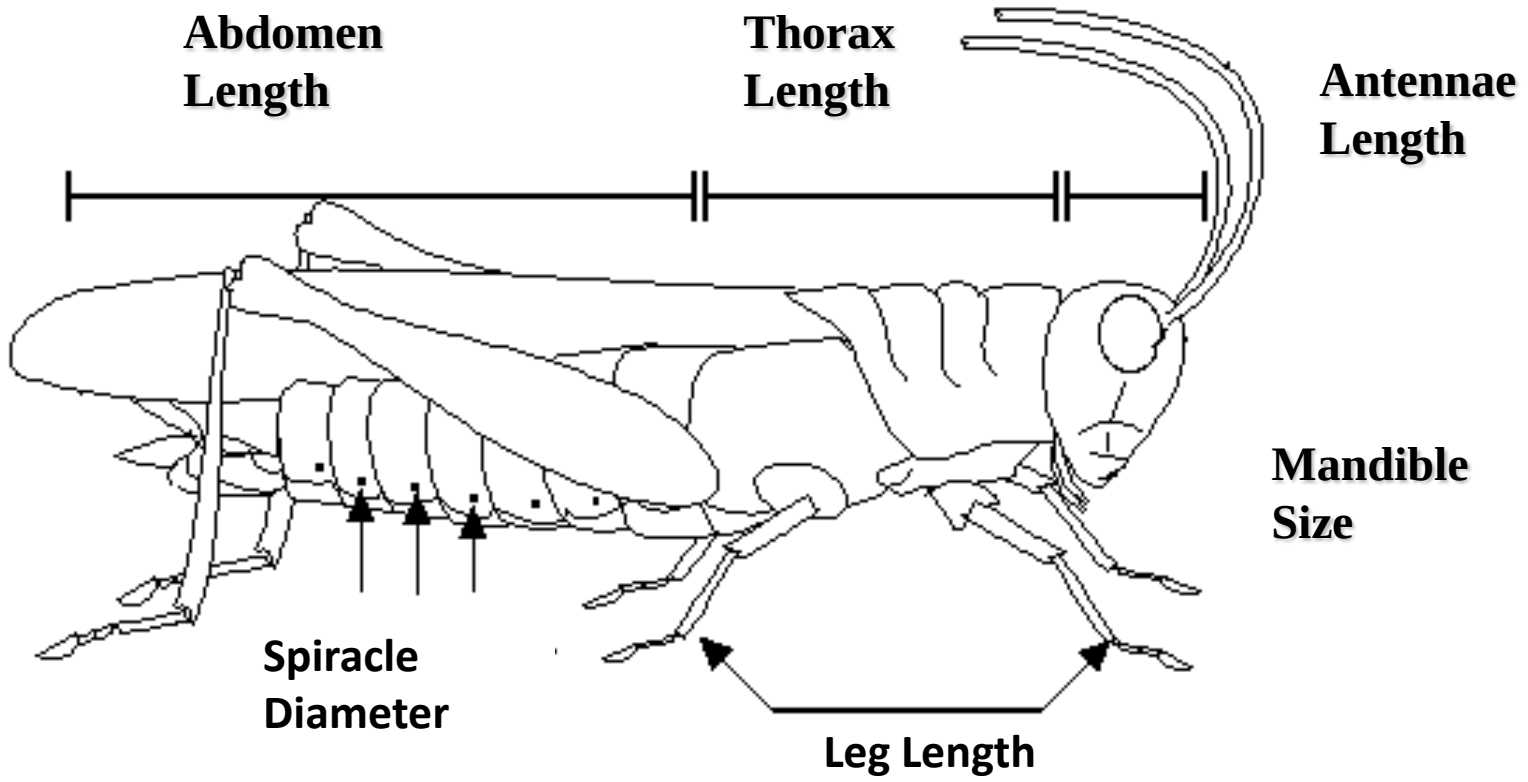
Grasshoppers



For any domain of interest, we can measure *features*

Color {Green, Brown, Gray, Other}

Has Wings?



We can store features in a database.

The classification problem can now be expressed as:

• Given a training database (**My_Collection**), predict the **class** label of a previously unseen instance

My_Collection

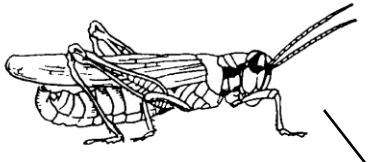
Insect ID	Abdomen Length	Antennae Length	Insect Class
1	2.7	5.5	**Grasshopper**
2	8.0	9.1	**Katydid**
3	0.9	4.7	**Grasshopper**
4	1.1	3.1	**Grasshopper**
5	5.4	8.5	**Katydid**
6	2.9	1.9	**Grasshopper**
7	6.1	6.6	**Katydid**
8	0.5	1.0	**Grasshopper**
9	8.3	6.6	**Katydid**
10	8.1	4.7	**Katydids**

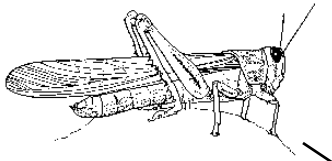
previously unseen instance =

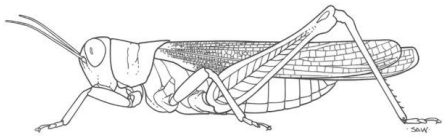
11	5.1	7.0	???????

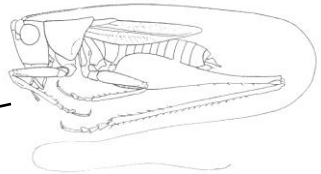
Grasshoppers

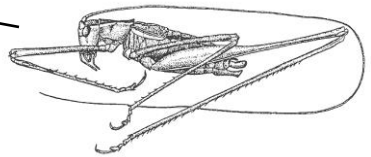
Katydids

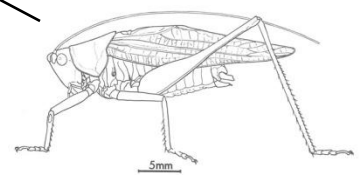


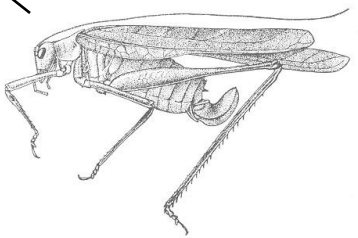


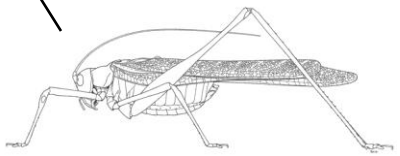


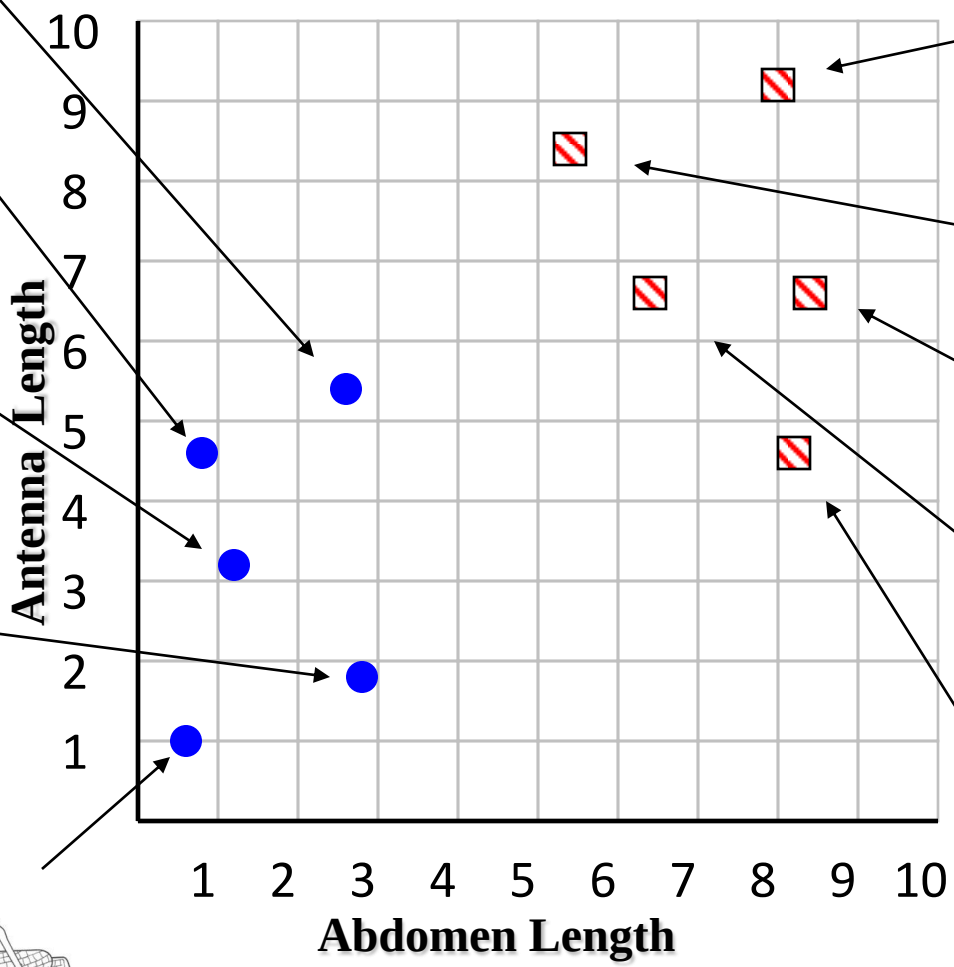


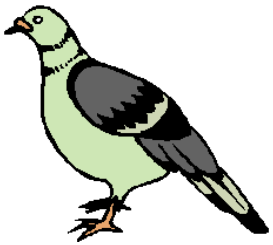












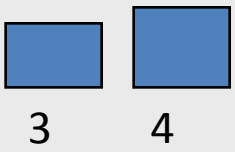
We will return to the previous slide in two minutes. In the meantime, we are going to play a quick game.

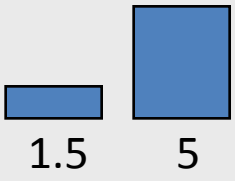
I am going to show you some classification problems which were shown to pigeons!

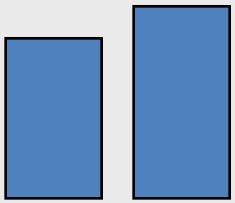
Let us see if you are as smart as a pigeon!

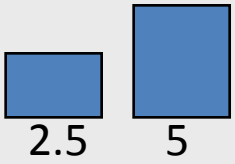
Pigeon Problem 1

Examples of class A

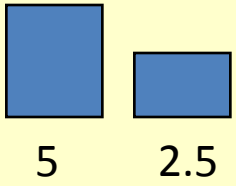


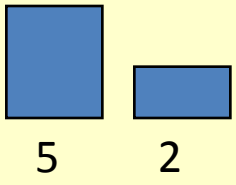


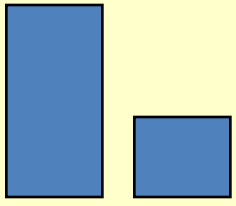


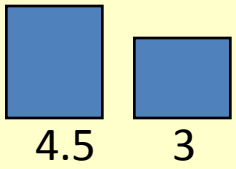


Examples of class B



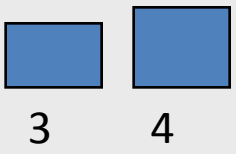


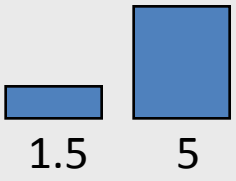


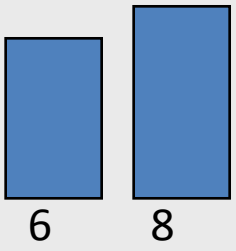


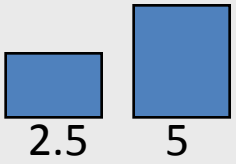
Pigeon Problem 1

Examples of class A

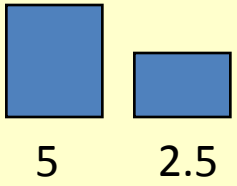


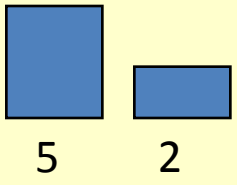


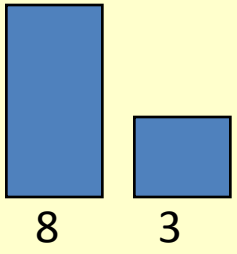


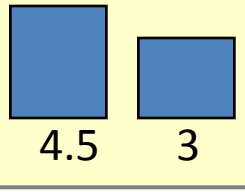


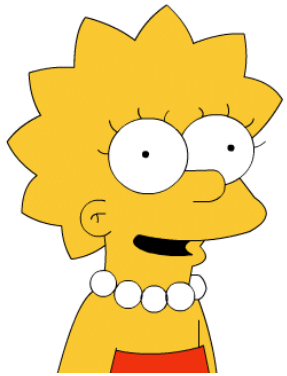
Examples of class B



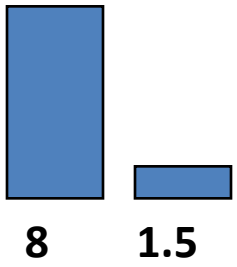




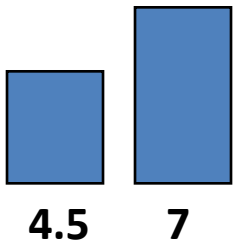




What class is this object?

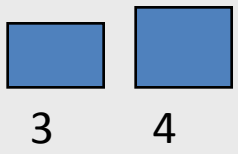


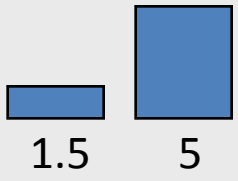
What about this one, A or B?

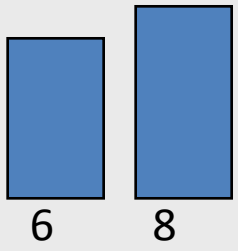


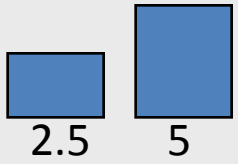
Pigeon Problem 1

Examples of class A

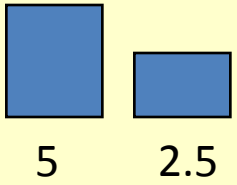


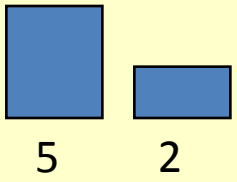


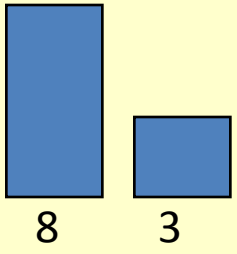


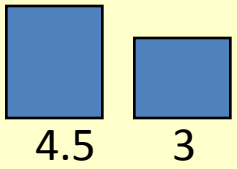


Examples of class B

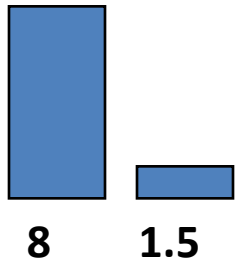








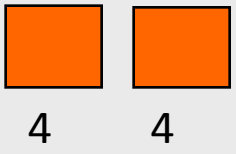
This is a B!

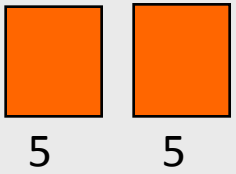


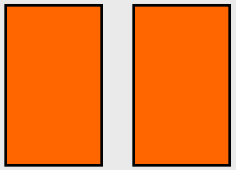
Here is the rule.
If the left bar is smaller than the right bar, it is an A, otherwise it is a B.

Pigeon Problem 2

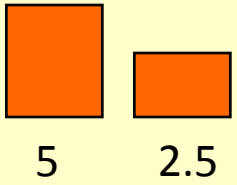
Examples of class A

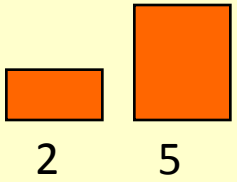


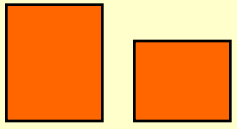


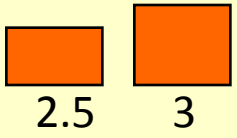


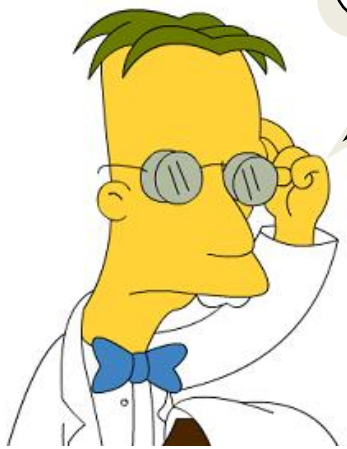
Examples of class B

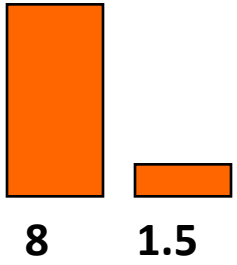


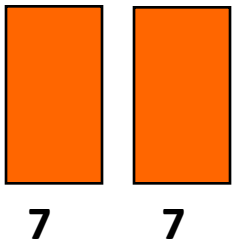






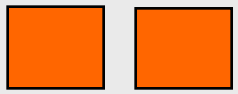


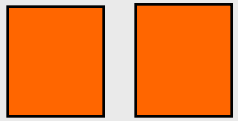


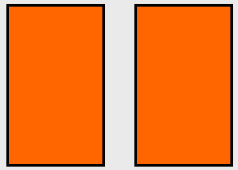


Pigeon Problem 2

Examples of class A

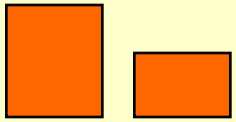

4 4

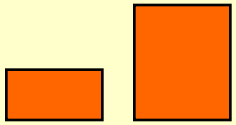

5 5

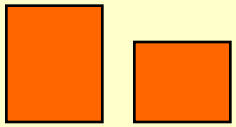

6 6

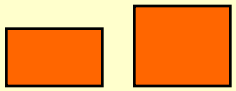

3 3

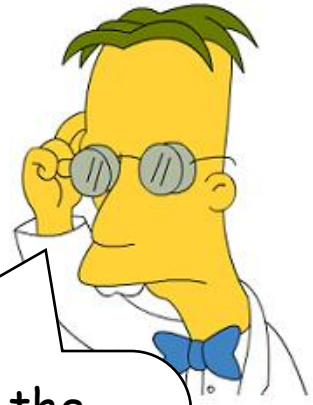
Examples of class B


5 2.5


2 5

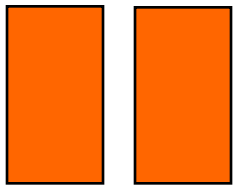

5 3


2.5 3



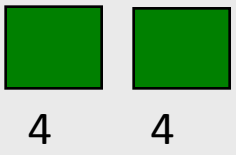
The rule is as follows, if the two bars are equal sizes, it is an A. Otherwise it is a B.

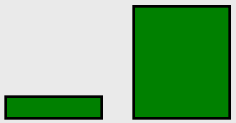
So this one is an A.

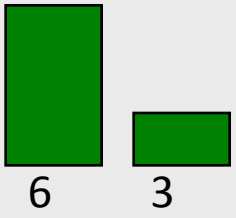

7 7

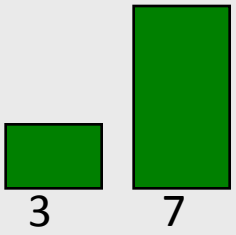
Pigeon Problem 3

Examples of class A

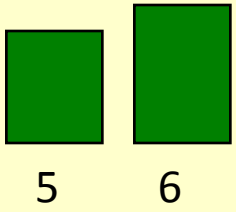


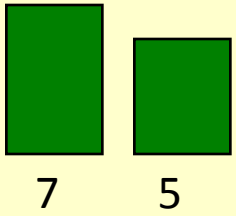


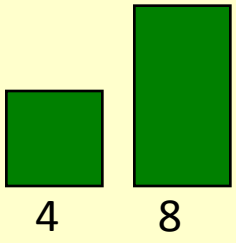


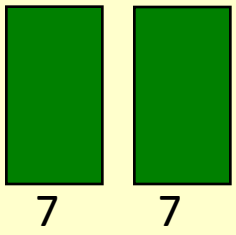


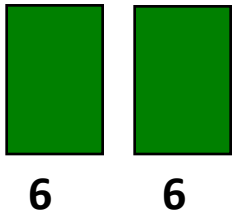
Examples of class B







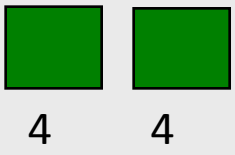


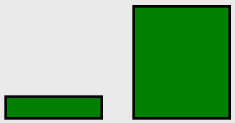


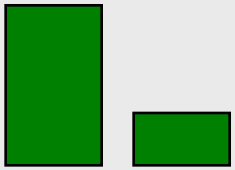
This one is really hard!
What is this, A or B?

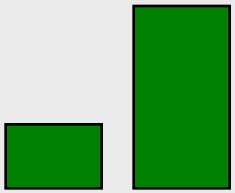
Pigeon Problem 3

Examples of class A

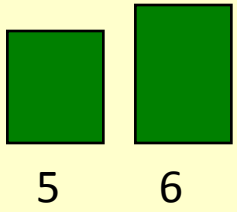


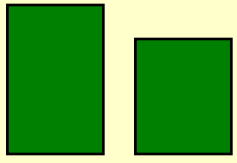


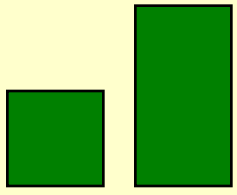


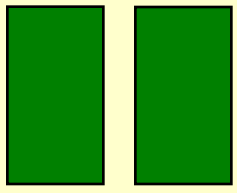


Examples of class B

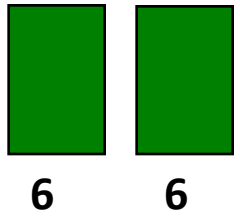








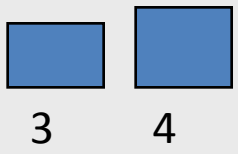
It is a B!

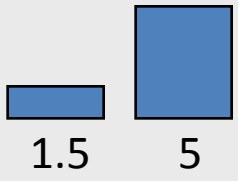


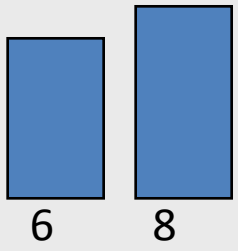
The rule is as follows, if the sum of the two bars is less than or equal to 10, it is an A. Otherwise it is a B.

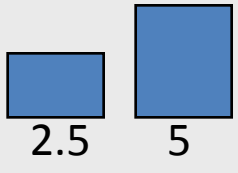
Pigeon Problem 1

Examples of class A

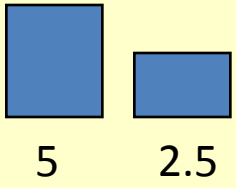


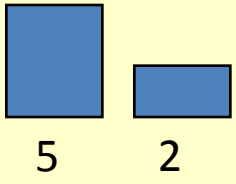


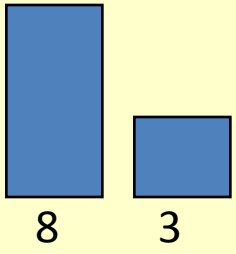


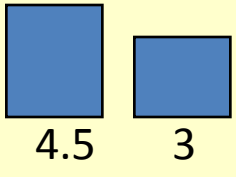


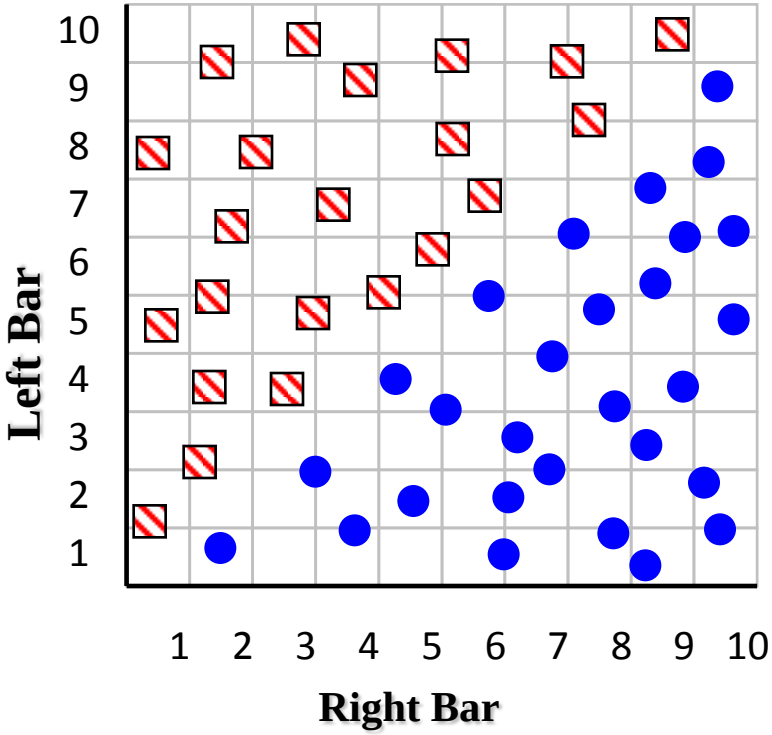
Examples of class B







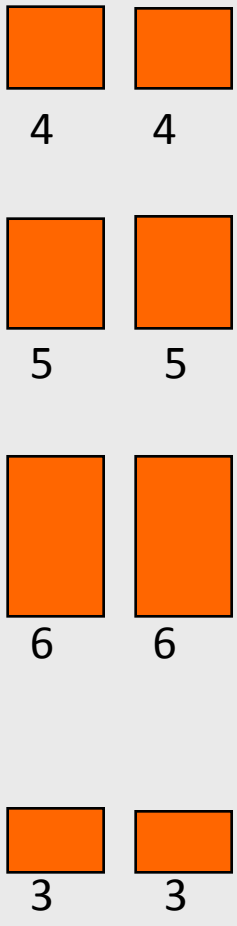




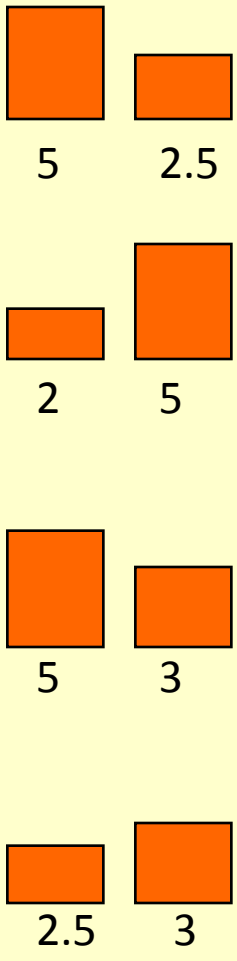
Here is the rule again. If the left bar is smaller than the right bar, it is an A, otherwise it is a B.

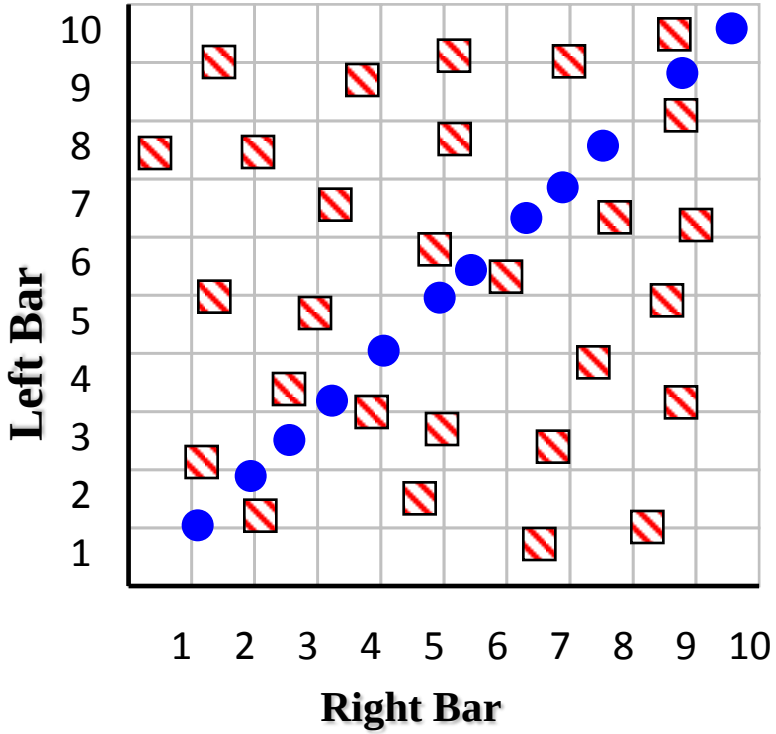
Pigeon Problem 2

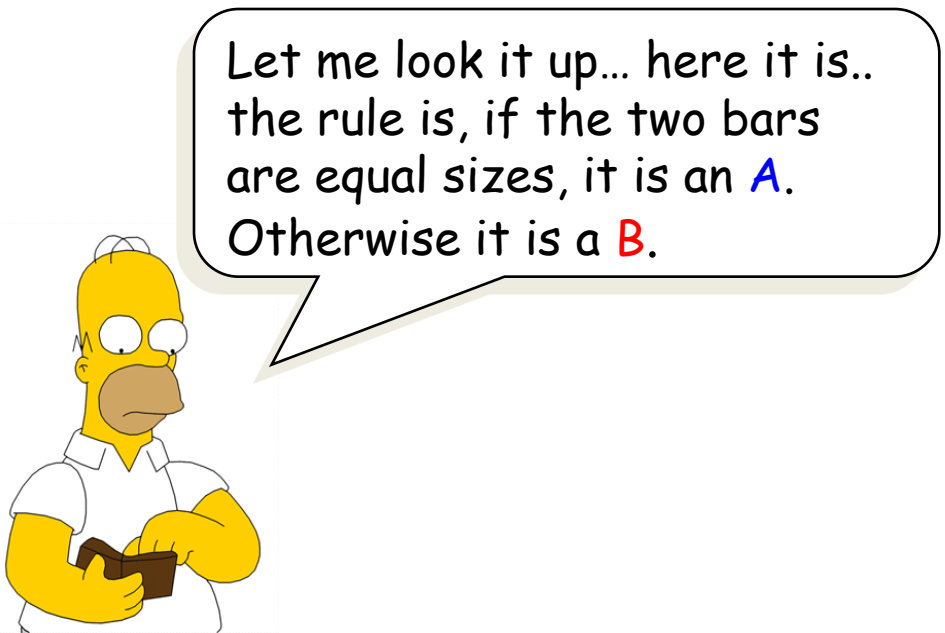
Examples of class A



Examples of class B

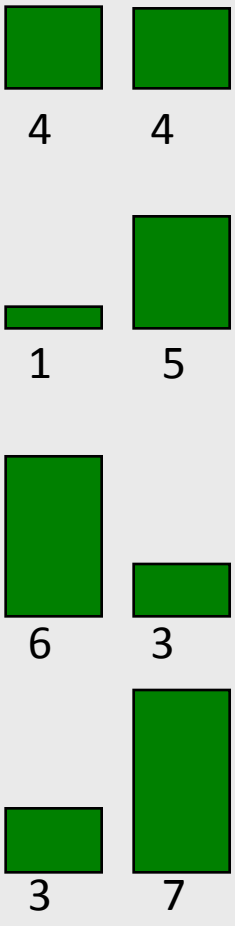




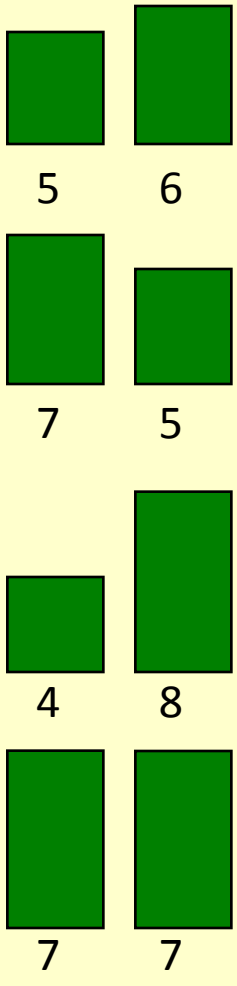


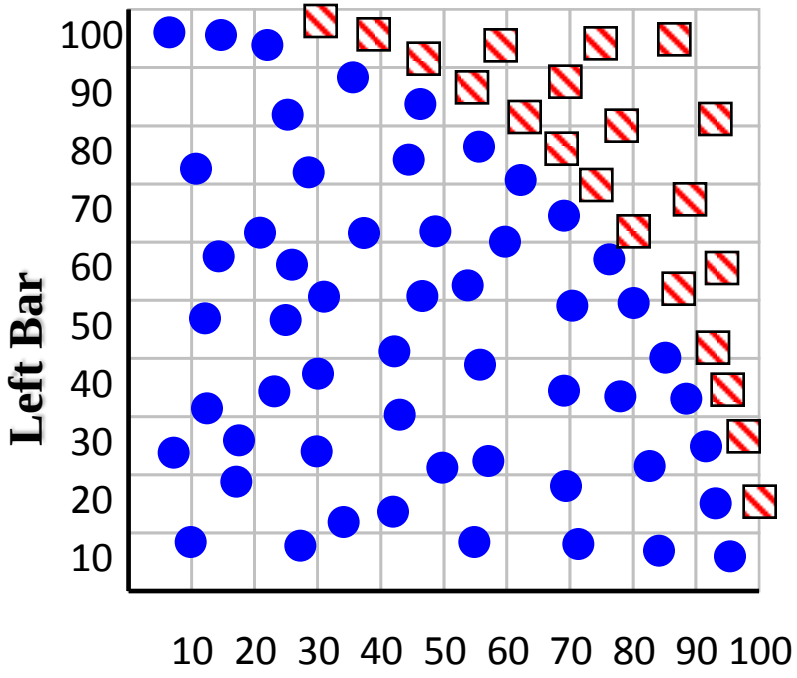
Pigeon Problem 3

Examples of class A



Examples of class B

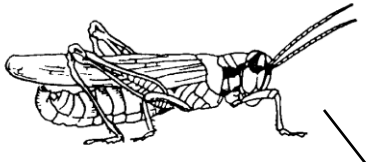


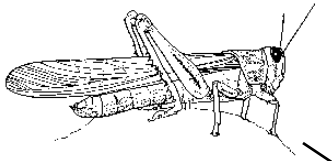


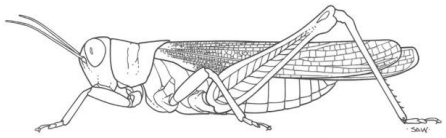
The rule again:
if the square of the sum of the two bars is less than or equal to 100, it is an A. Otherwise it is a B.

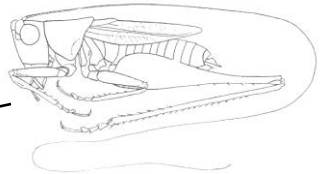
Grasshoppers

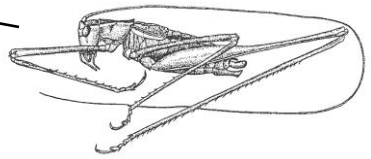
Katydids

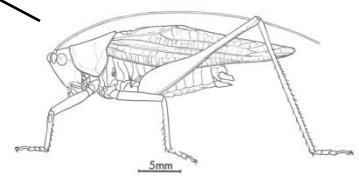


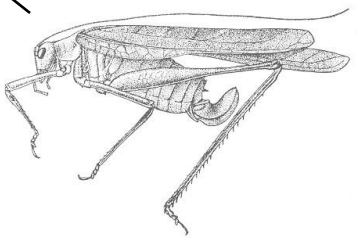


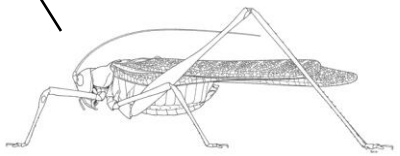


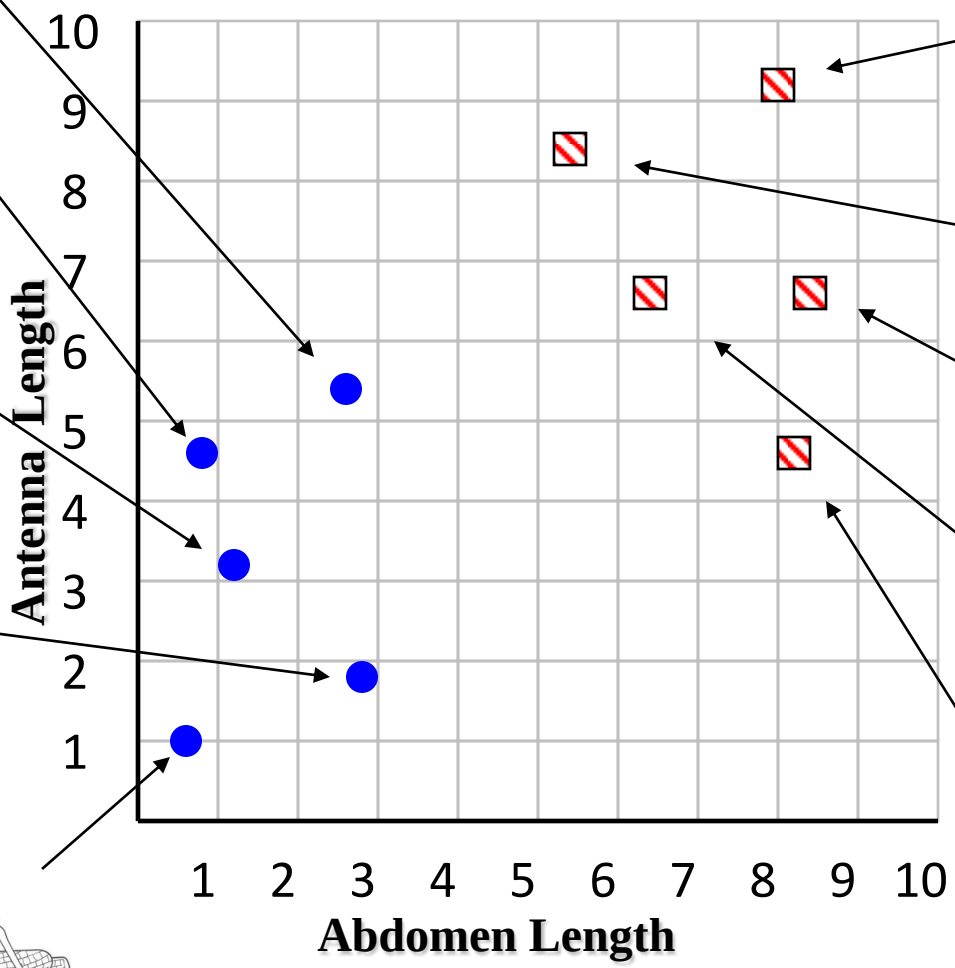






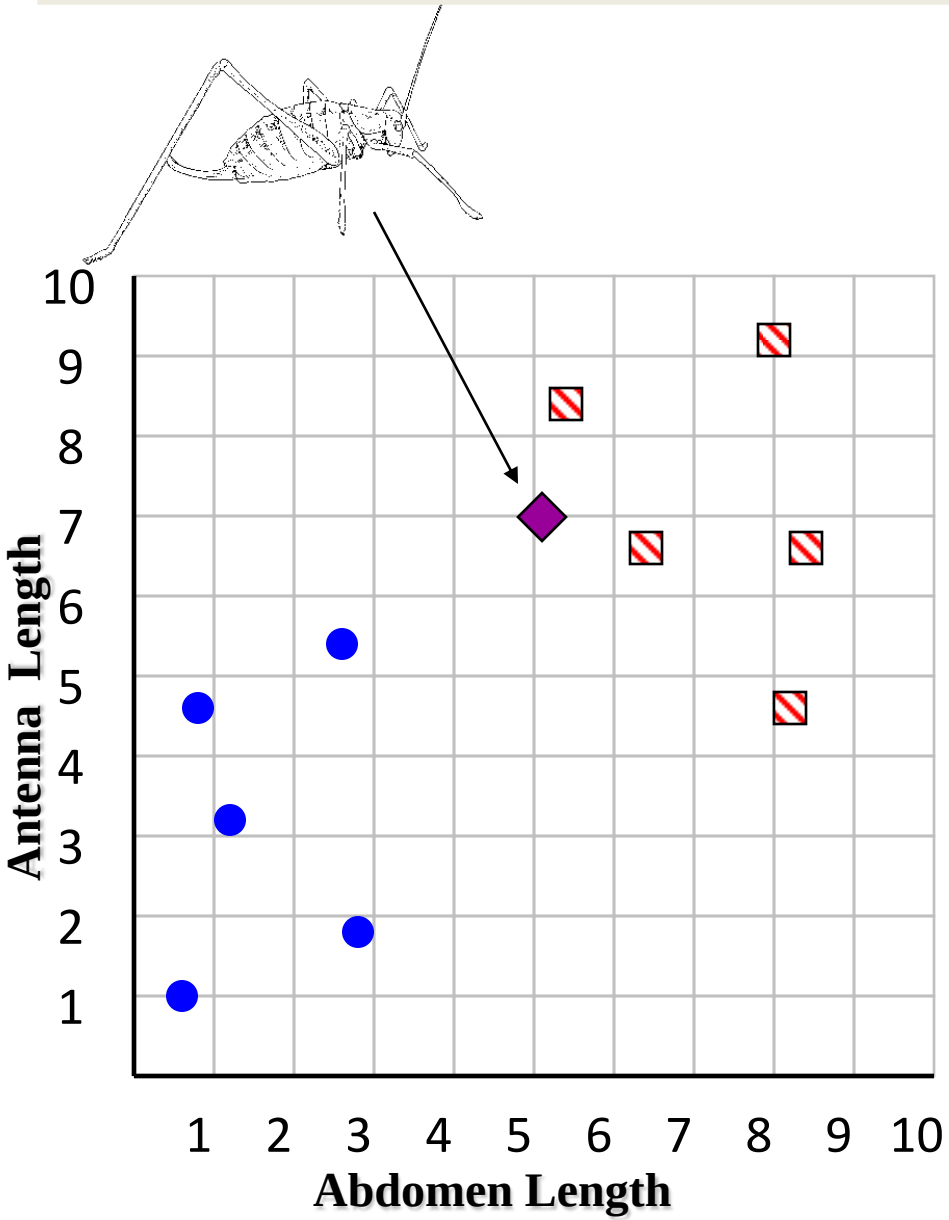






previously unseen instance =

11	**5.1**	**7.0**	**???????**



• We can "project" the **previously unseen instance** into the same space as the database.

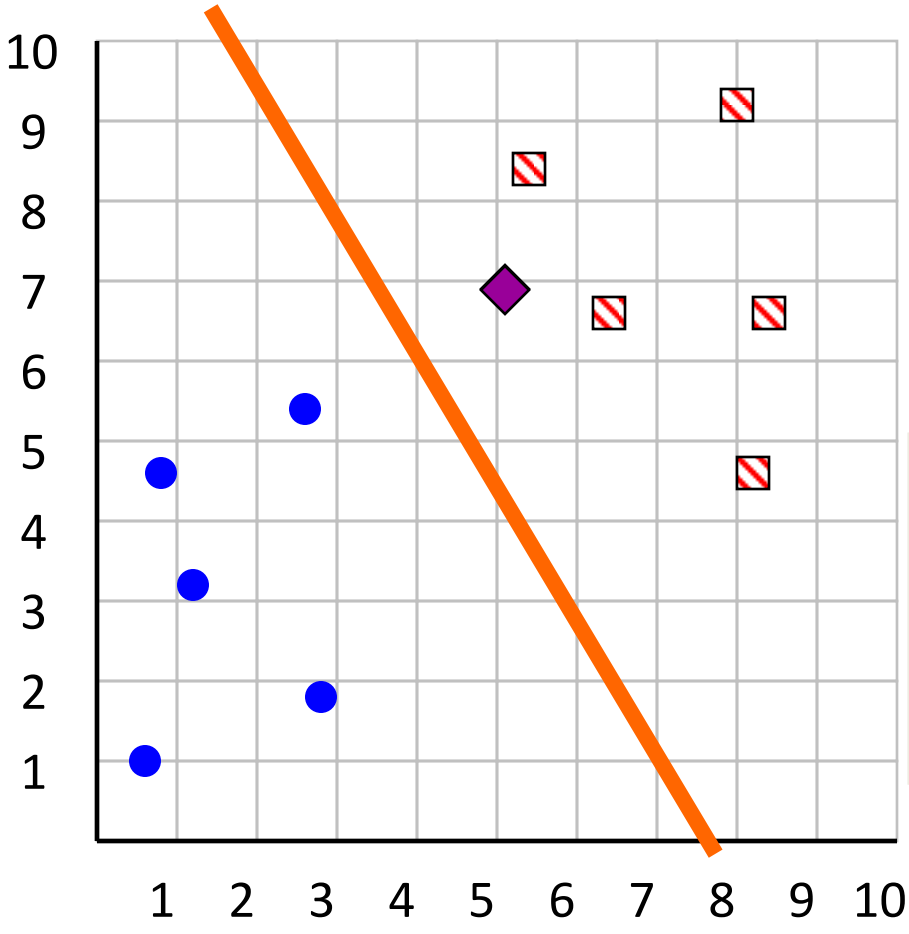
• We have now abstracted away the details of our particular problem. It will be much easier to talk about points in space.

▨ **Katydids**

● **Grasshoppers**

Simple Linear Classifier

R.A. Fisher
1890-1962

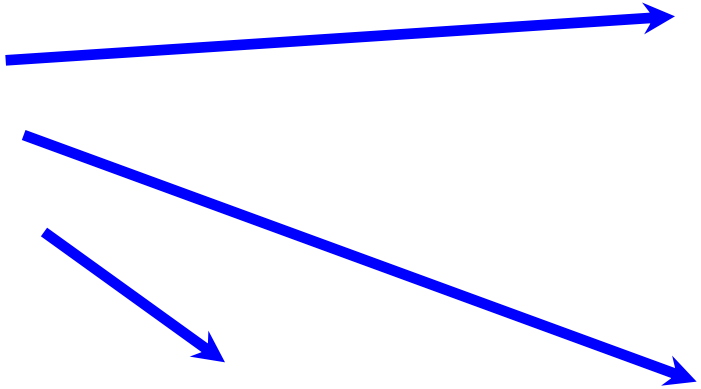


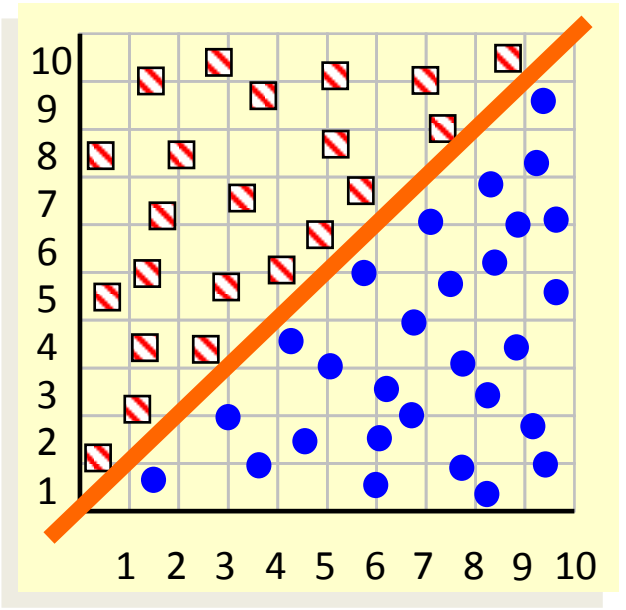
If previously unseen instance above the line
then
class is **Katydid**
else
class is **Grasshopper**

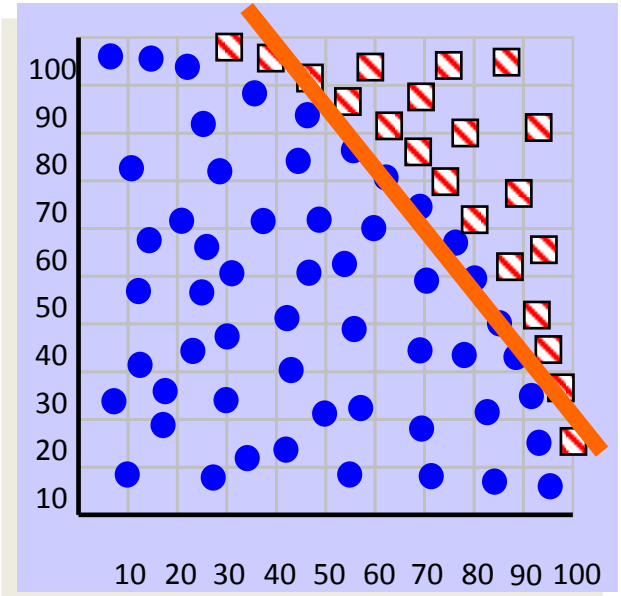
Katydids
Grasshoppers

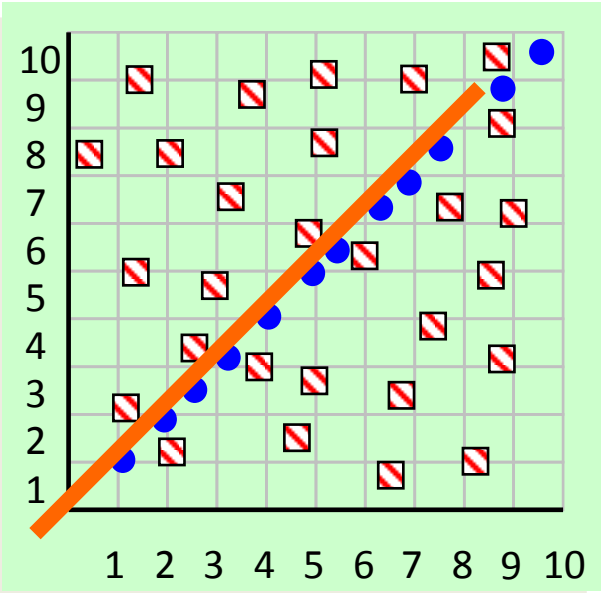
Which of the "Pigeon Problems" can be solved by the Simple Linear Classifier?

1) Perfect
2) Useless
3) Pretty Good









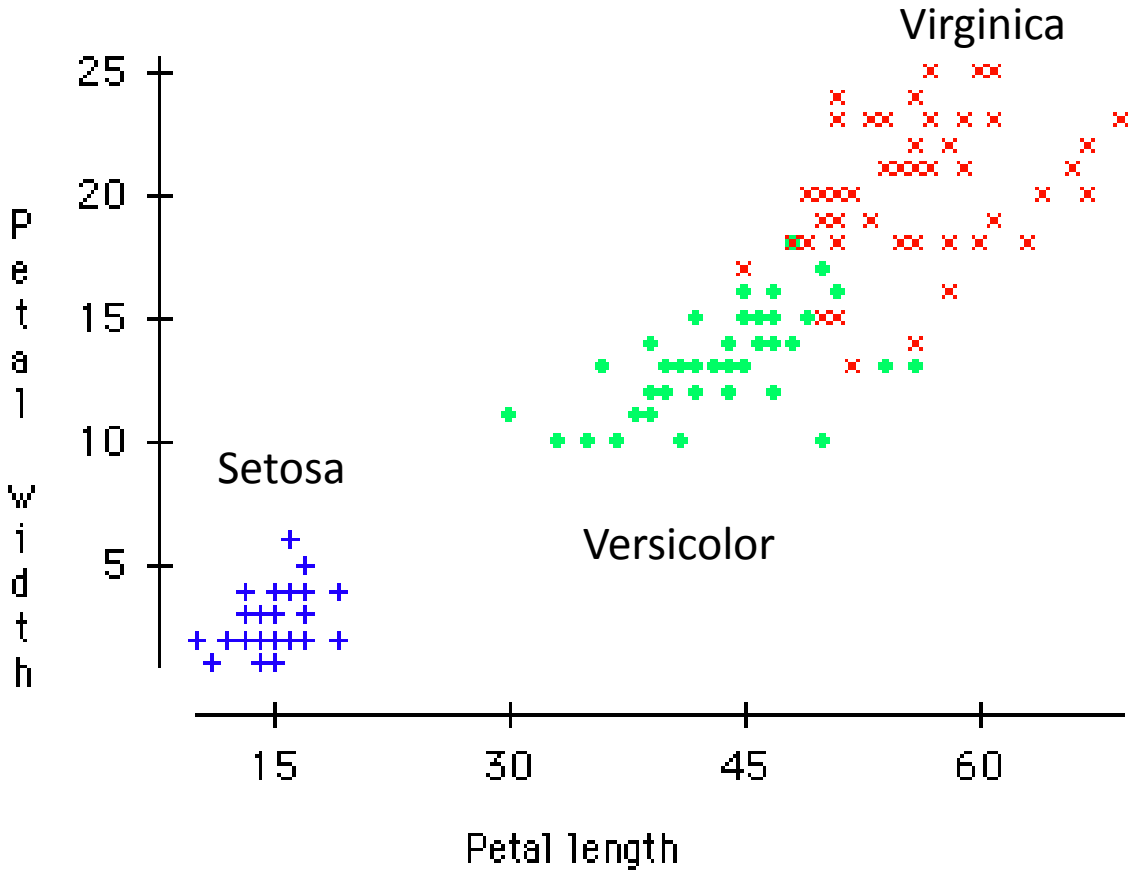
Problems that can be solved by a linear classifier are call **linearly separable**.

A Famous Problem

R. A. Fisher's Iris Dataset.

- 3 classes
- 50 of each class

The task is to classify Iris plants into one of 3 varieties using the Petal Length and Petal Width.

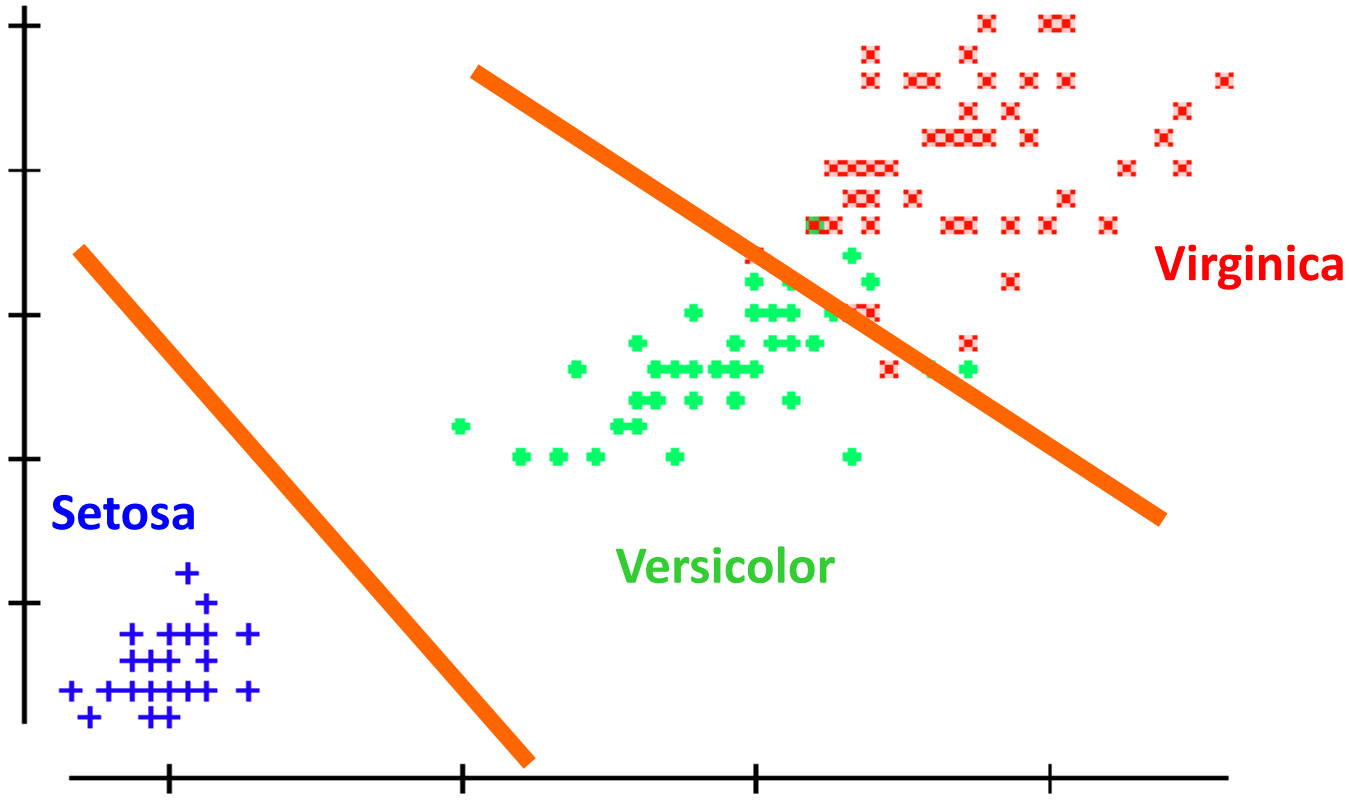


Iris Setosa

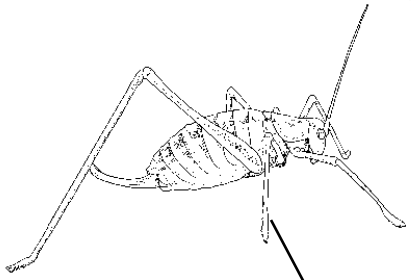
Iris Versicolor

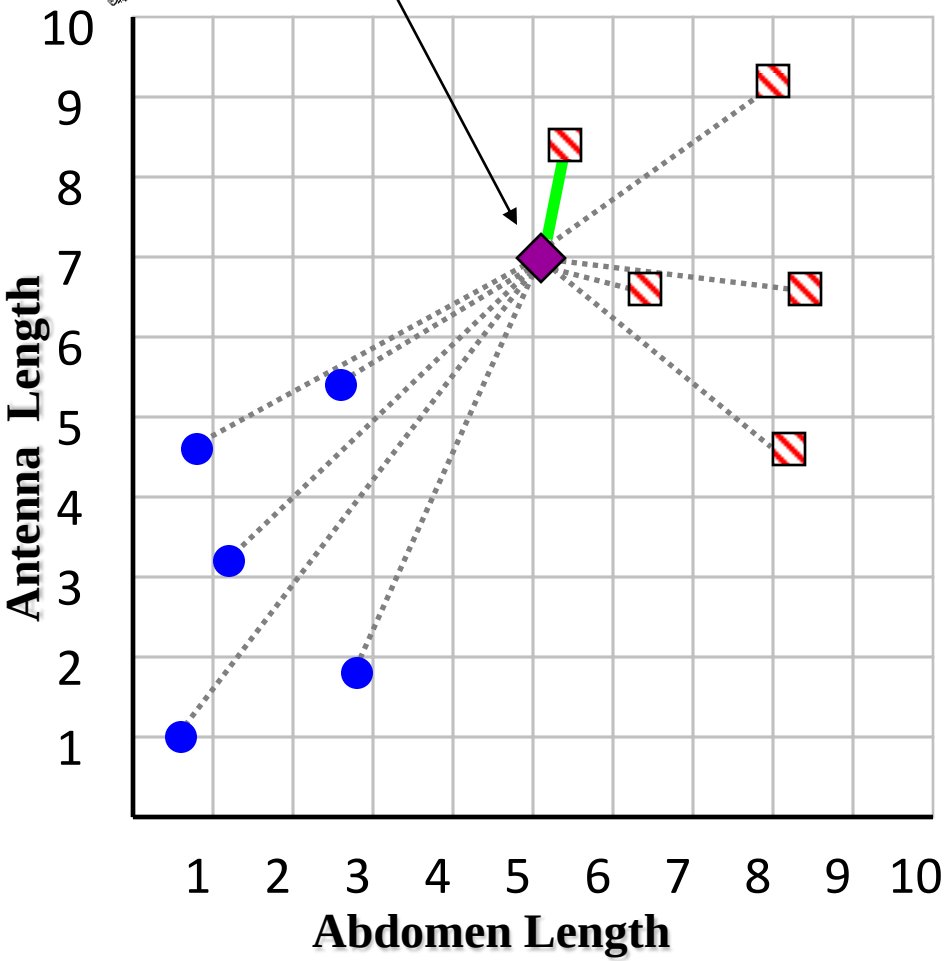
Iris Virginica

We can generalize the piecewise linear classifier to N classes, by fitting N-1 lines. In this case we first learned the line to (perfectly) discriminate between **Setosa** and **Virginica**/**Versicolor**, then we learned to approximately discriminate between **Virginica** and **Versicolor**.



Nearest Neighbor Classifier





If the **nearest** instance to the previously unseen instance **is a Katydid**

class is **Katydid**

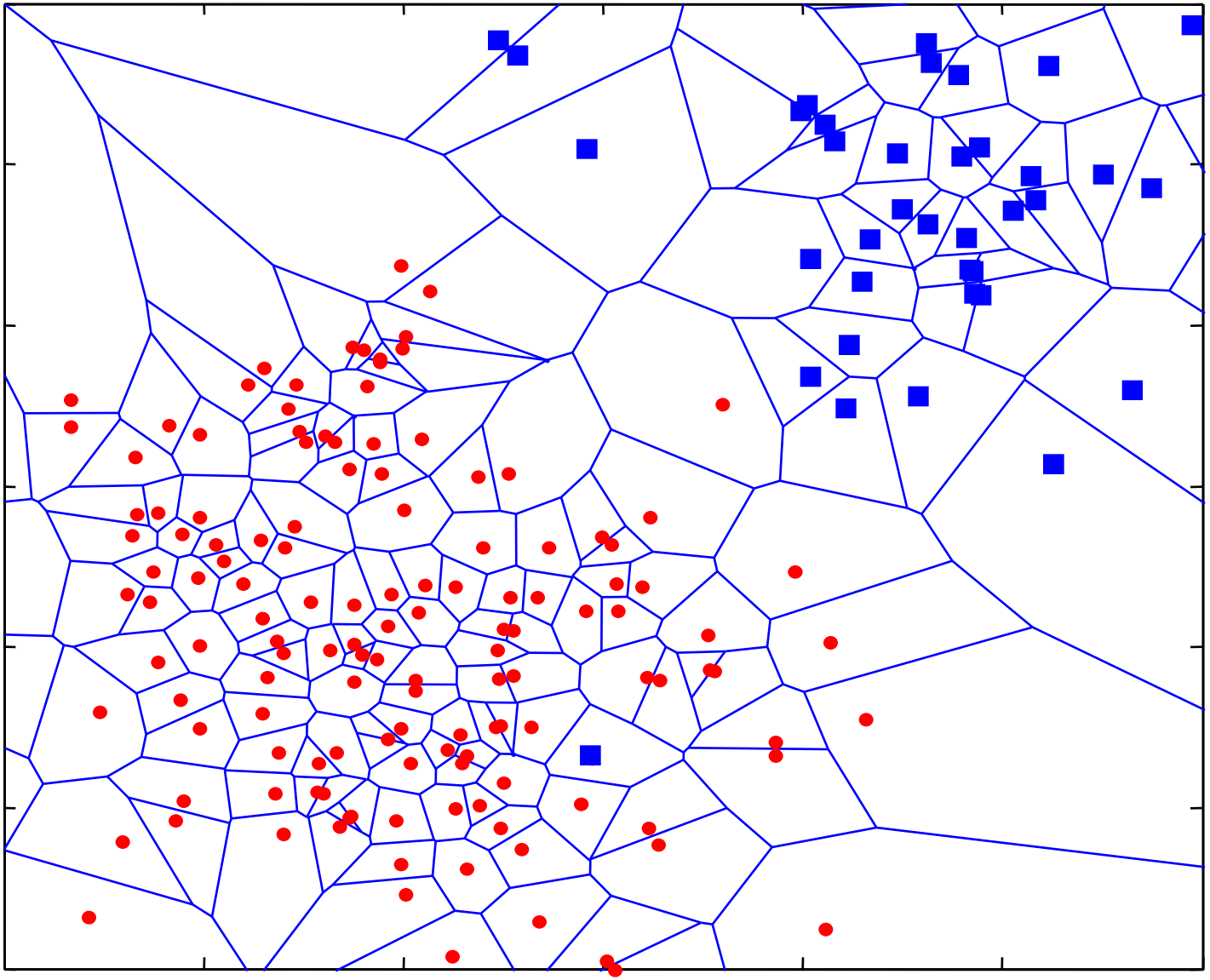
else

class is **Grasshopper**

Katydids

Grasshoppers

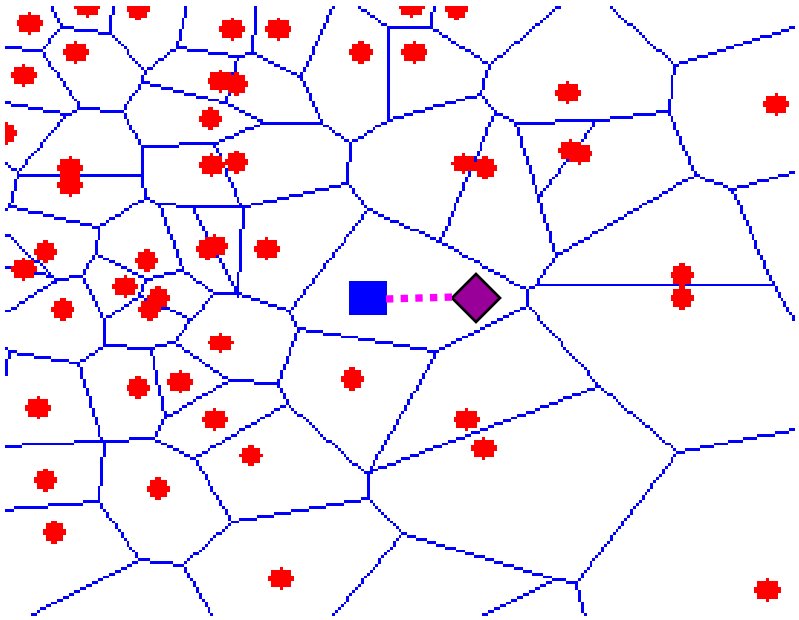
The nearest neighbor algorithm is sensitive to outliers…



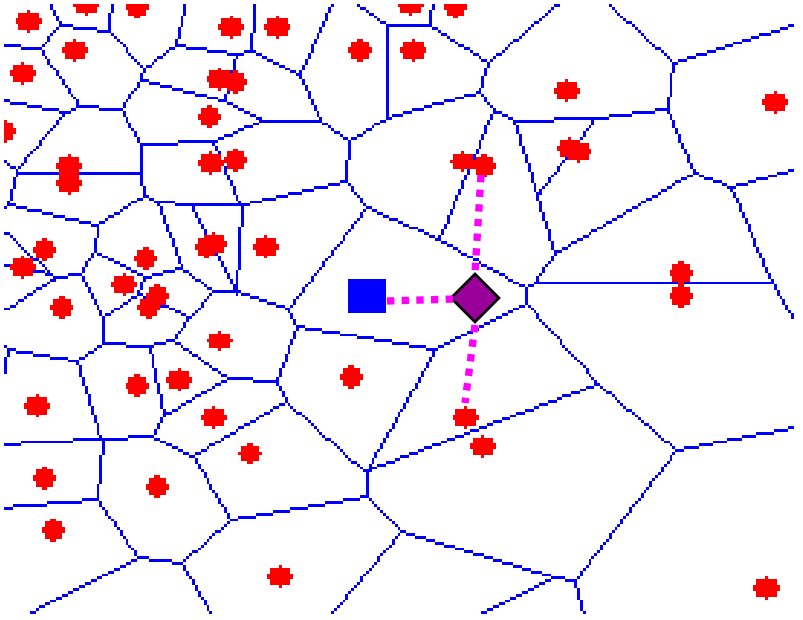
The solution is to…

We can generalize the nearest neighbor algorithm to the K- nearest neighbor (KNN) algorithm.

We measure the distance to the nearest K instances, and let them vote. K is typically chosen to be an odd number.

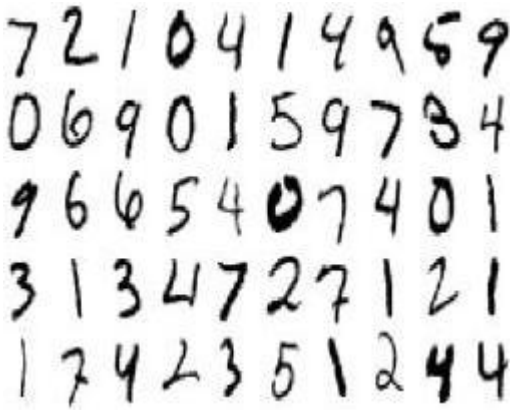


K = 1



K = 3

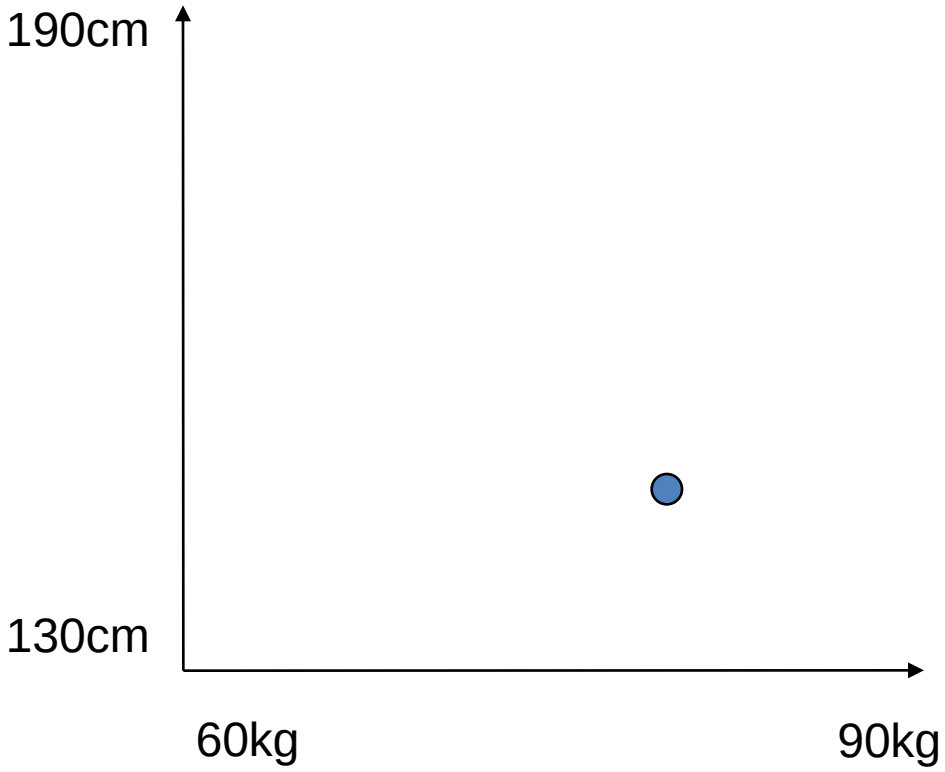
- Why recognising rugby players is (almost) the same problem as *handwriting recognition*



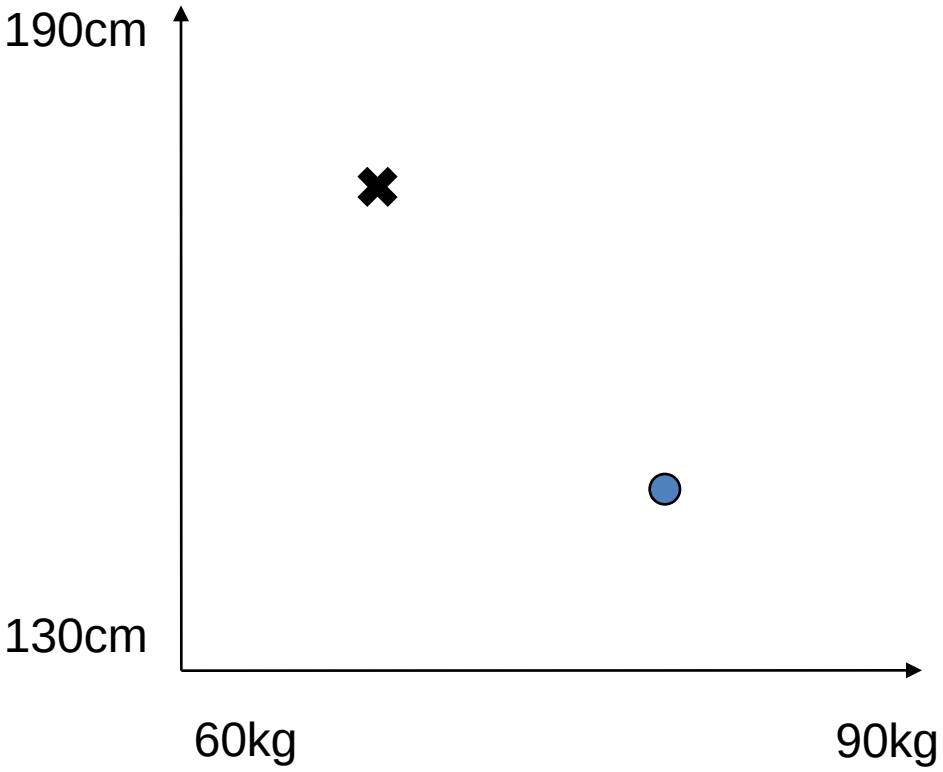
Can we LEARN to recognise a rugby player?

What are the “features” of a rugby player?

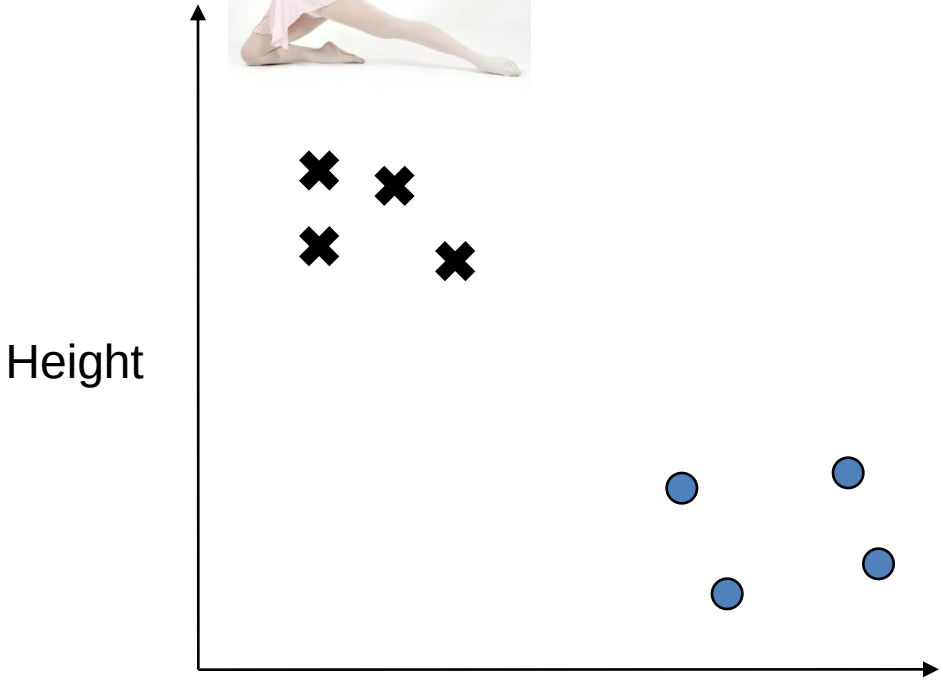
Rugby players = short + heavy?



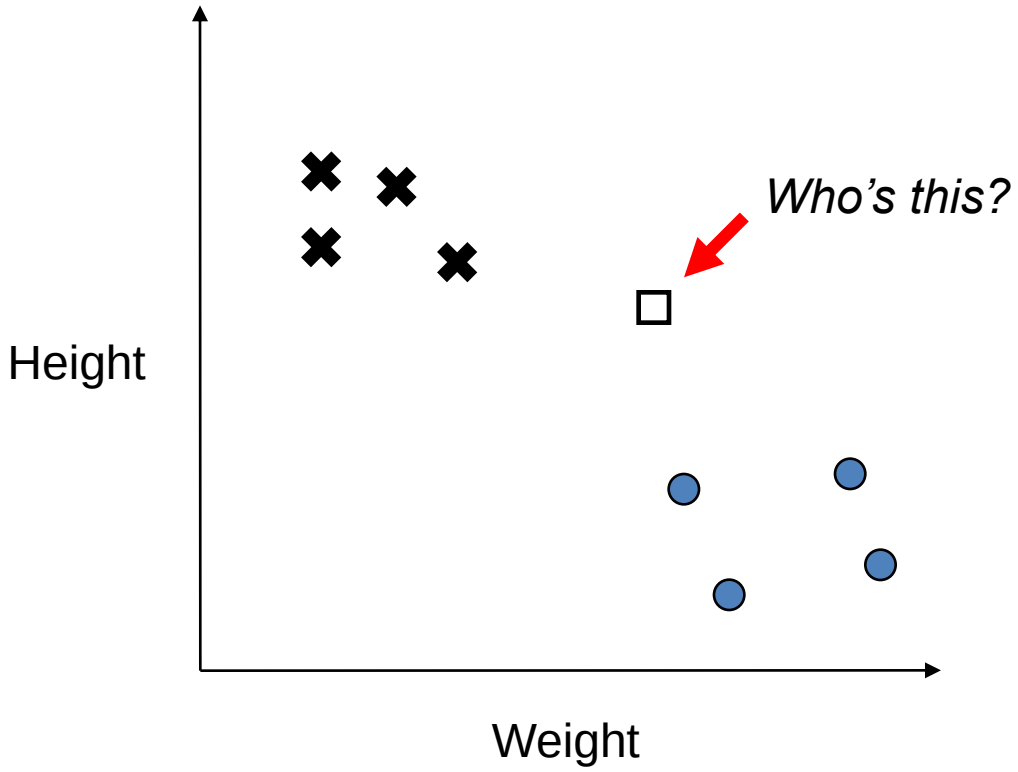
Ballet dancers = tall + skinny?



Rugby players “cluster” separately in the space.

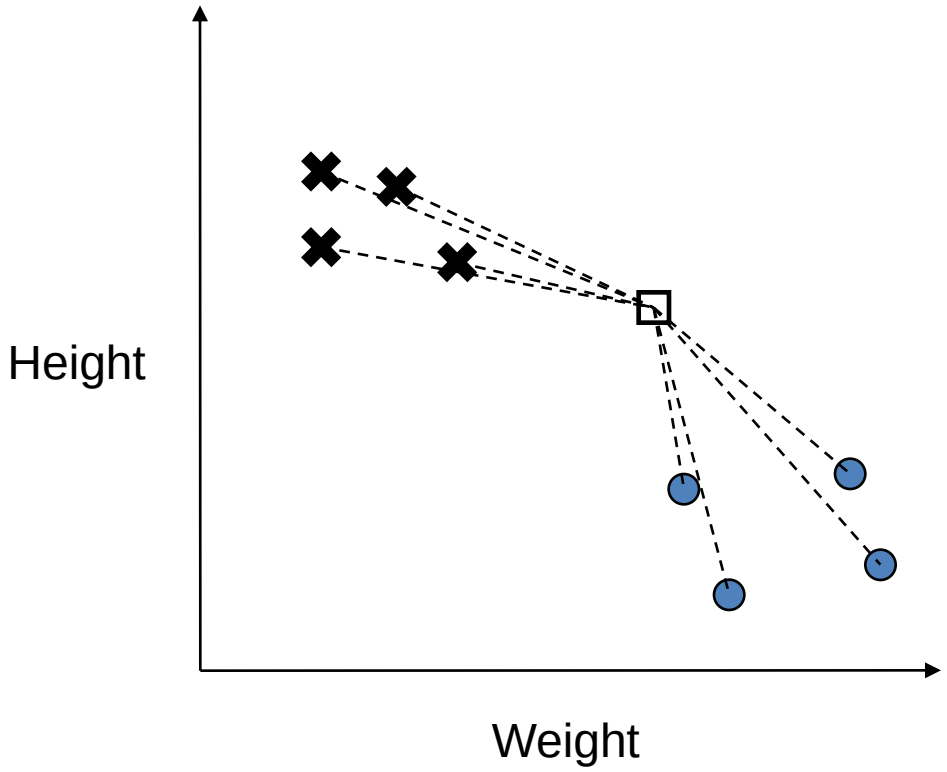


The K-Nearest Neighbour Algorithm



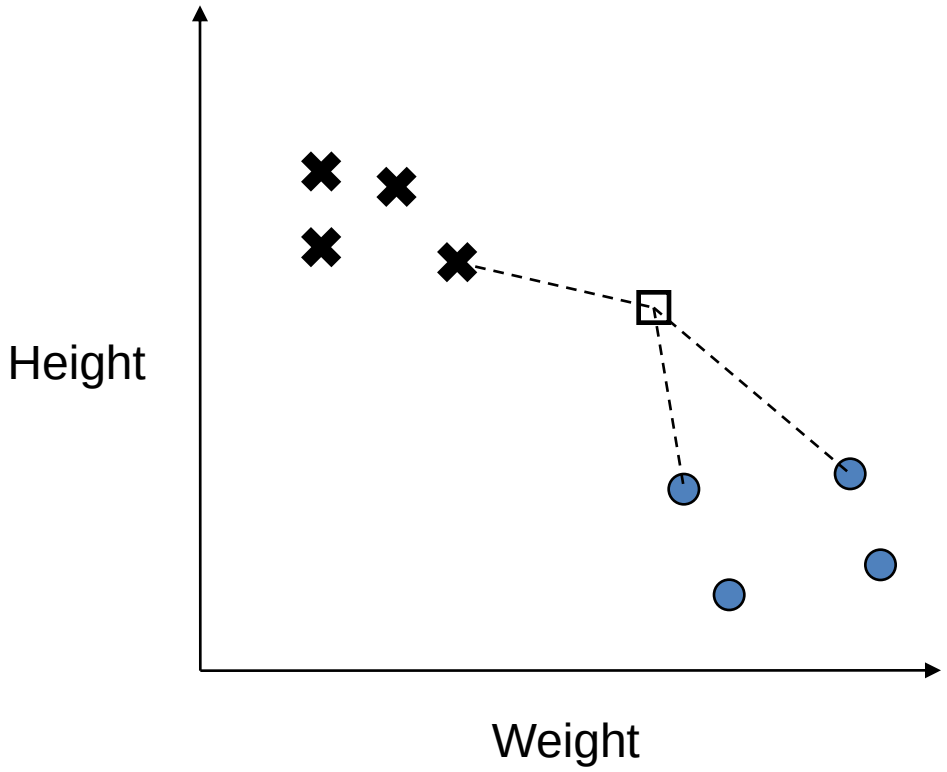
The K-Nearest Neighbour Algorithm

1. *Measure distance to all points*



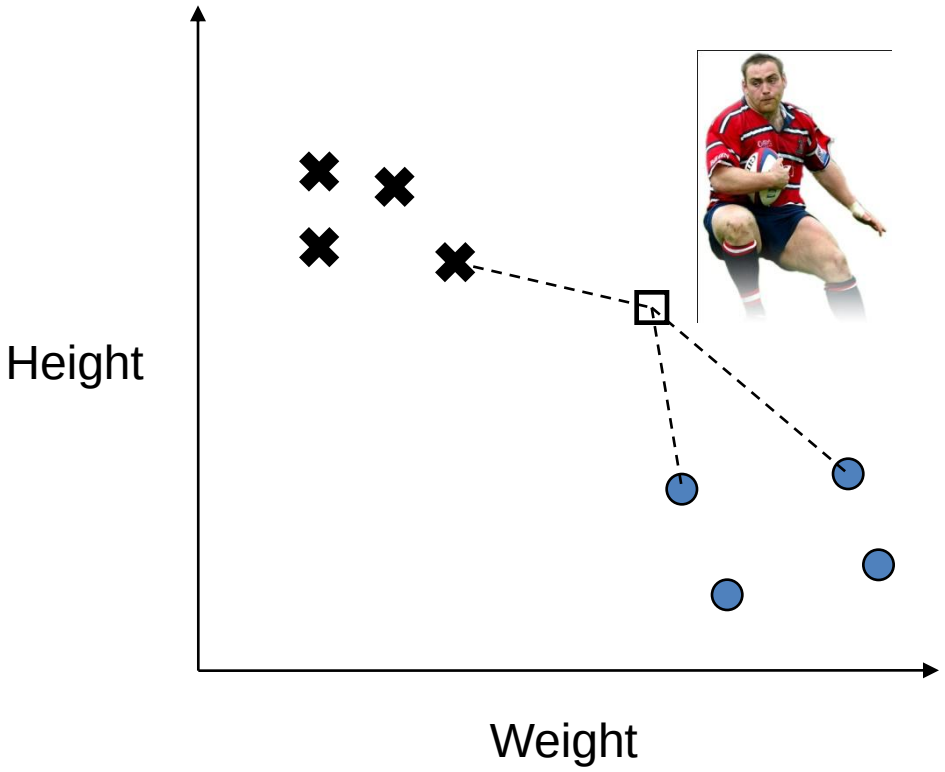
The K-Nearest Neighbour Algorithm

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



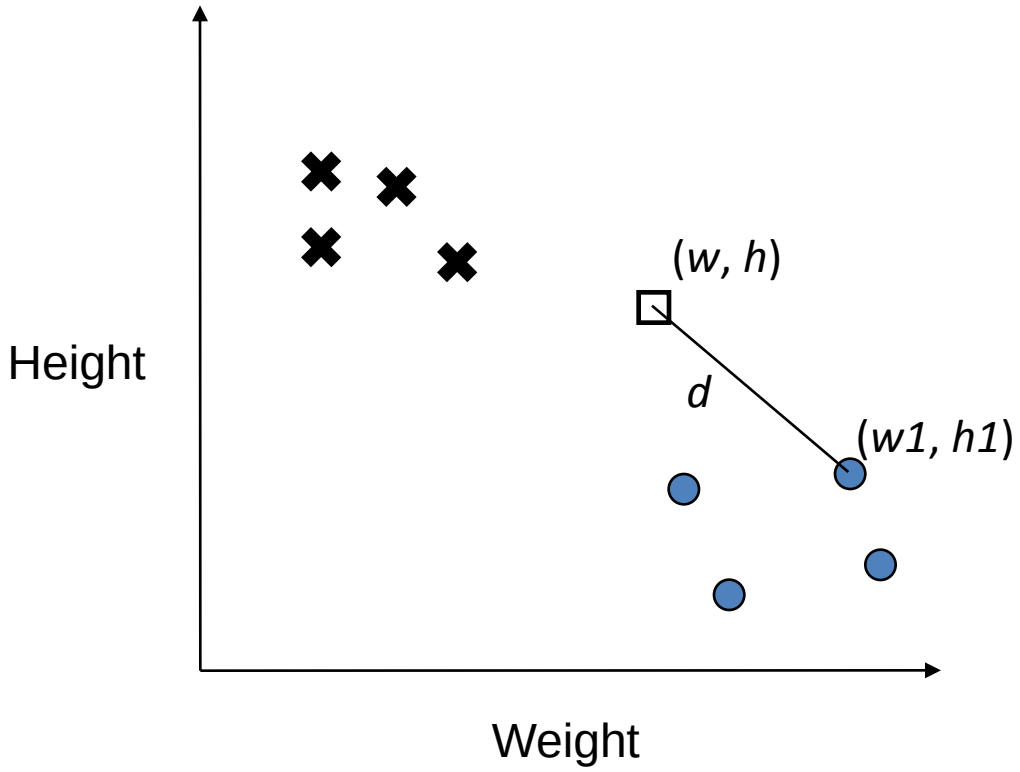
The K-Nearest Neighbour Algorithm

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



“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
   predict that class
end
```

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

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

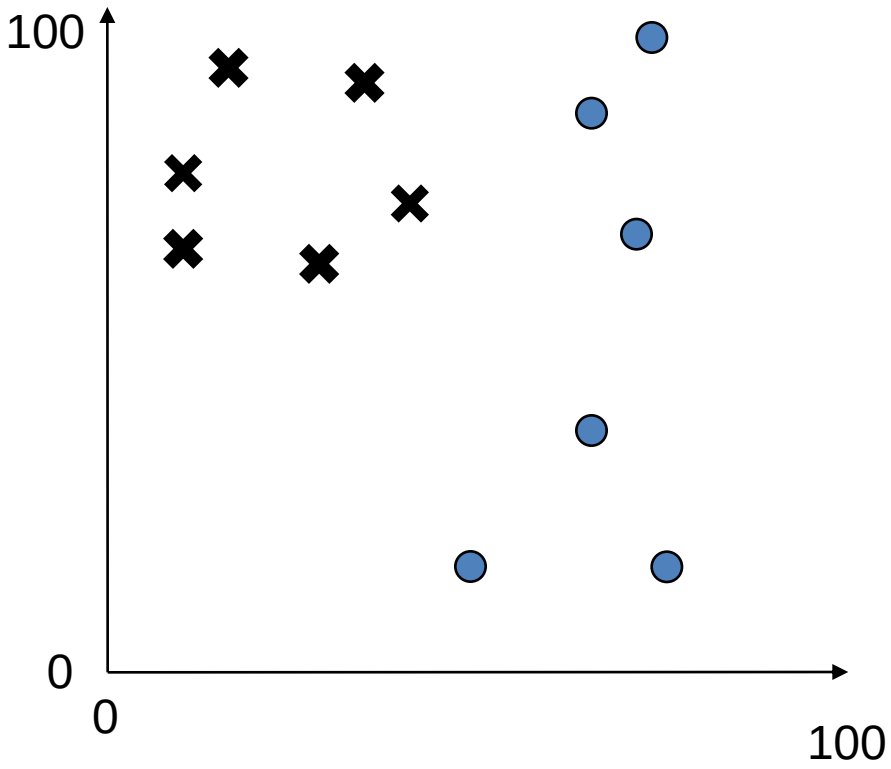
$$d = \sqrt{(x - x_1)^2 + (y - y_1)^2 + (z - z_1)^2}$$

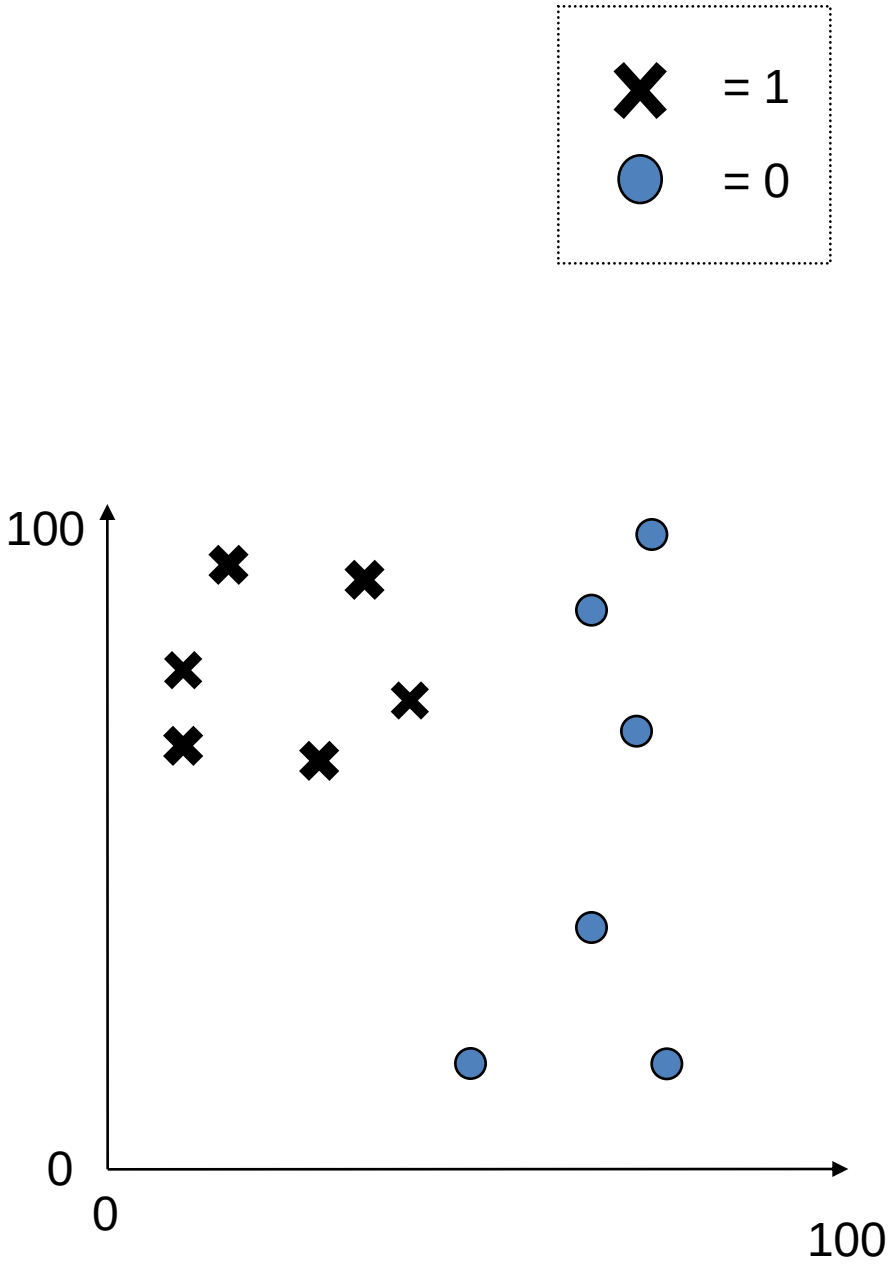
x = Height
y = Weight
z = Shoe size

Choosing the wrong features makes it difficult, too many and it's computationally intensive.

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

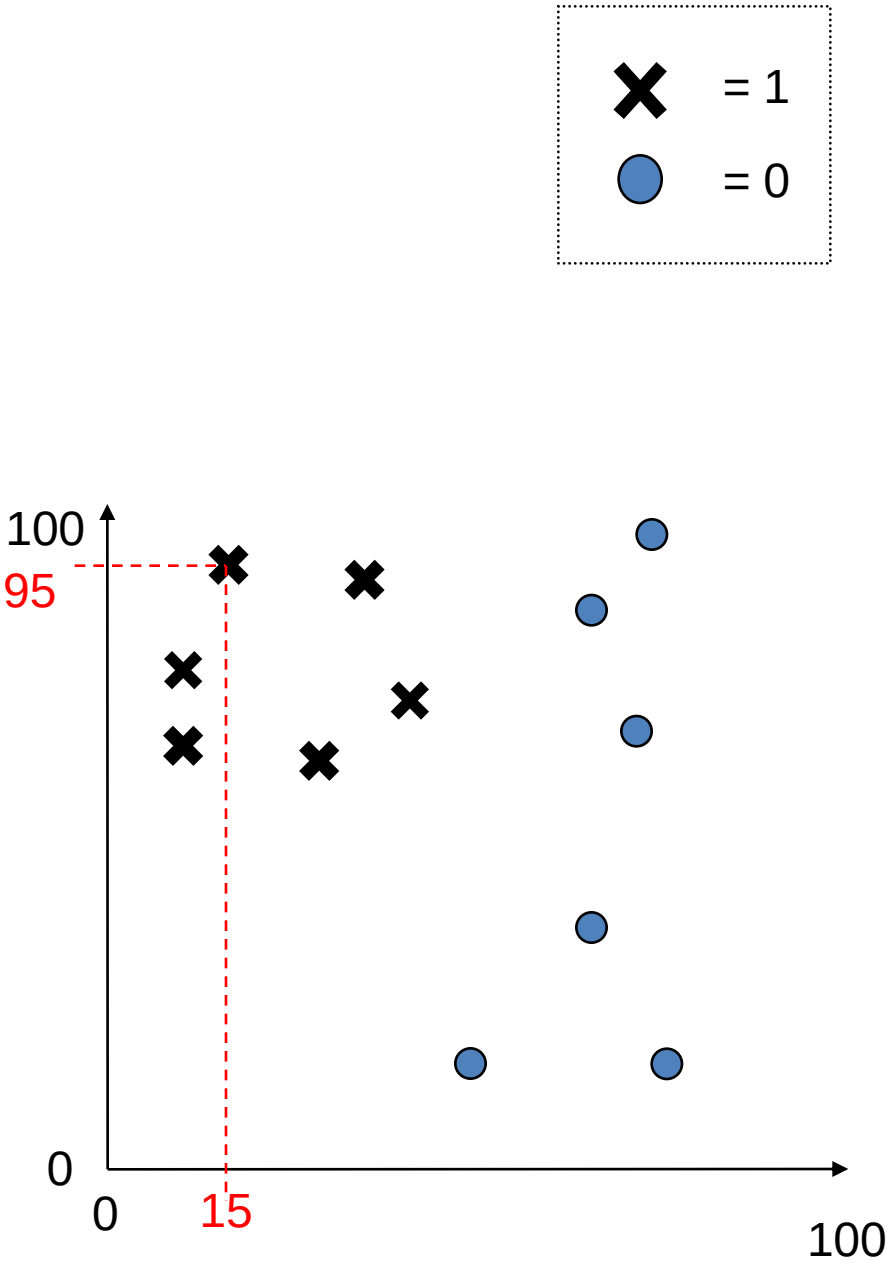
?


100
0
0
100



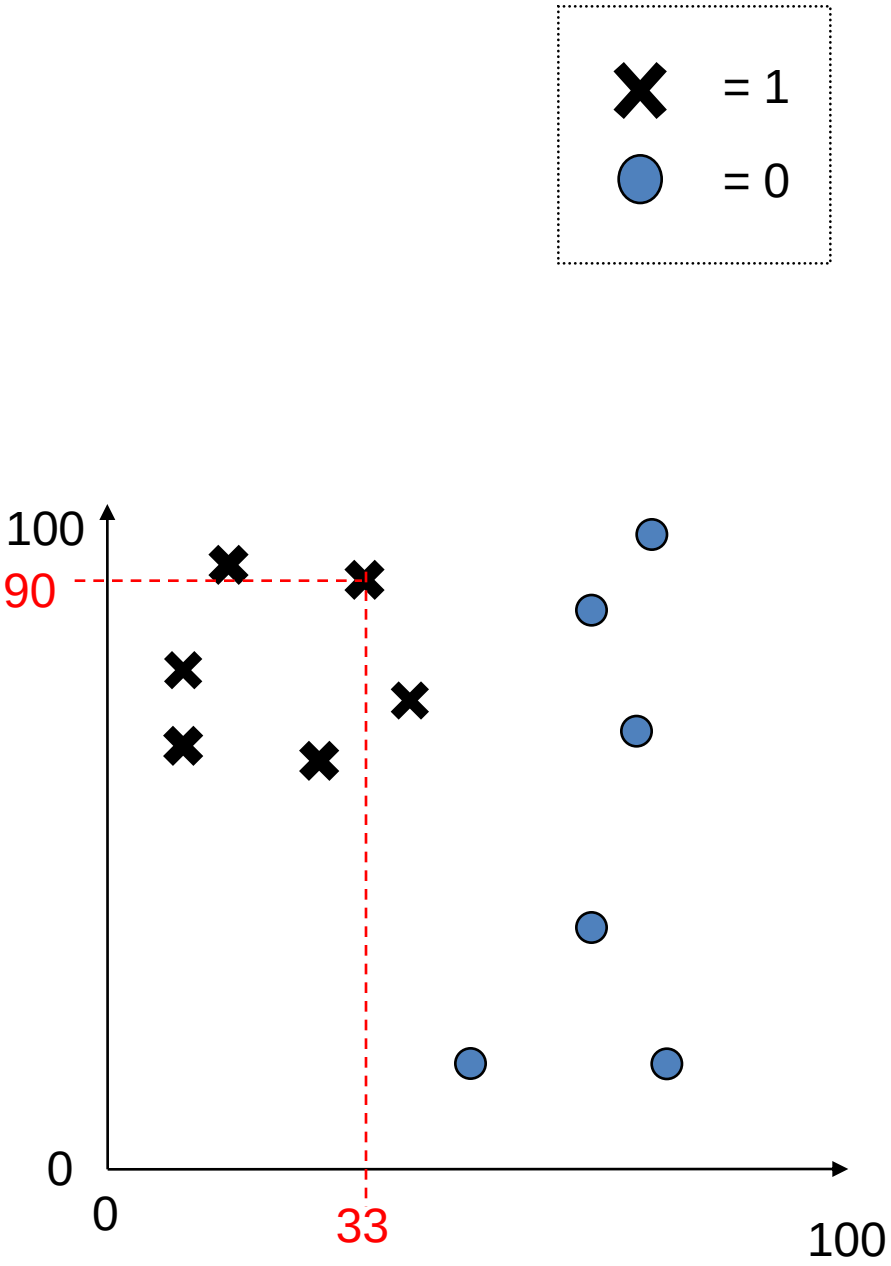
Inputs

Labels



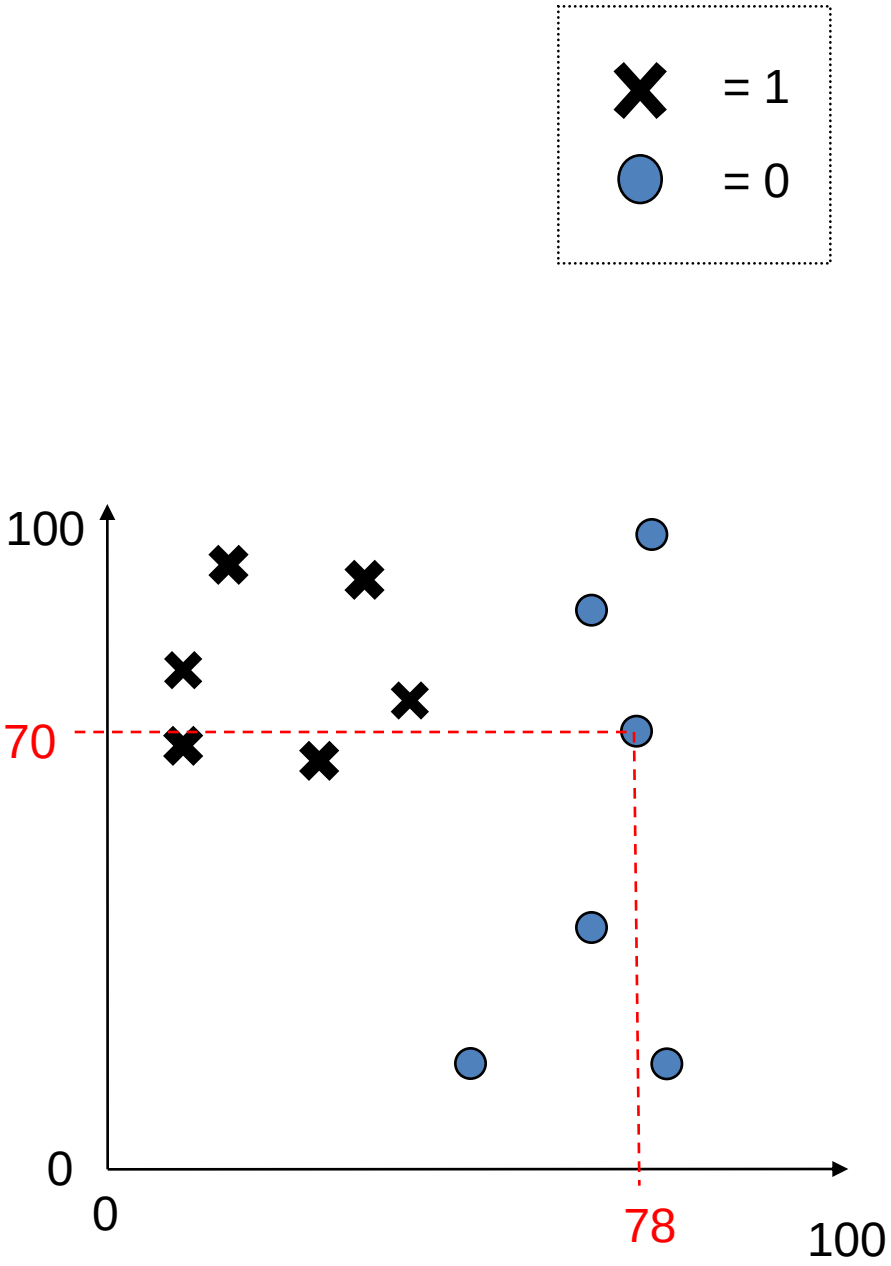
Inputs	
15	95

Labels
1



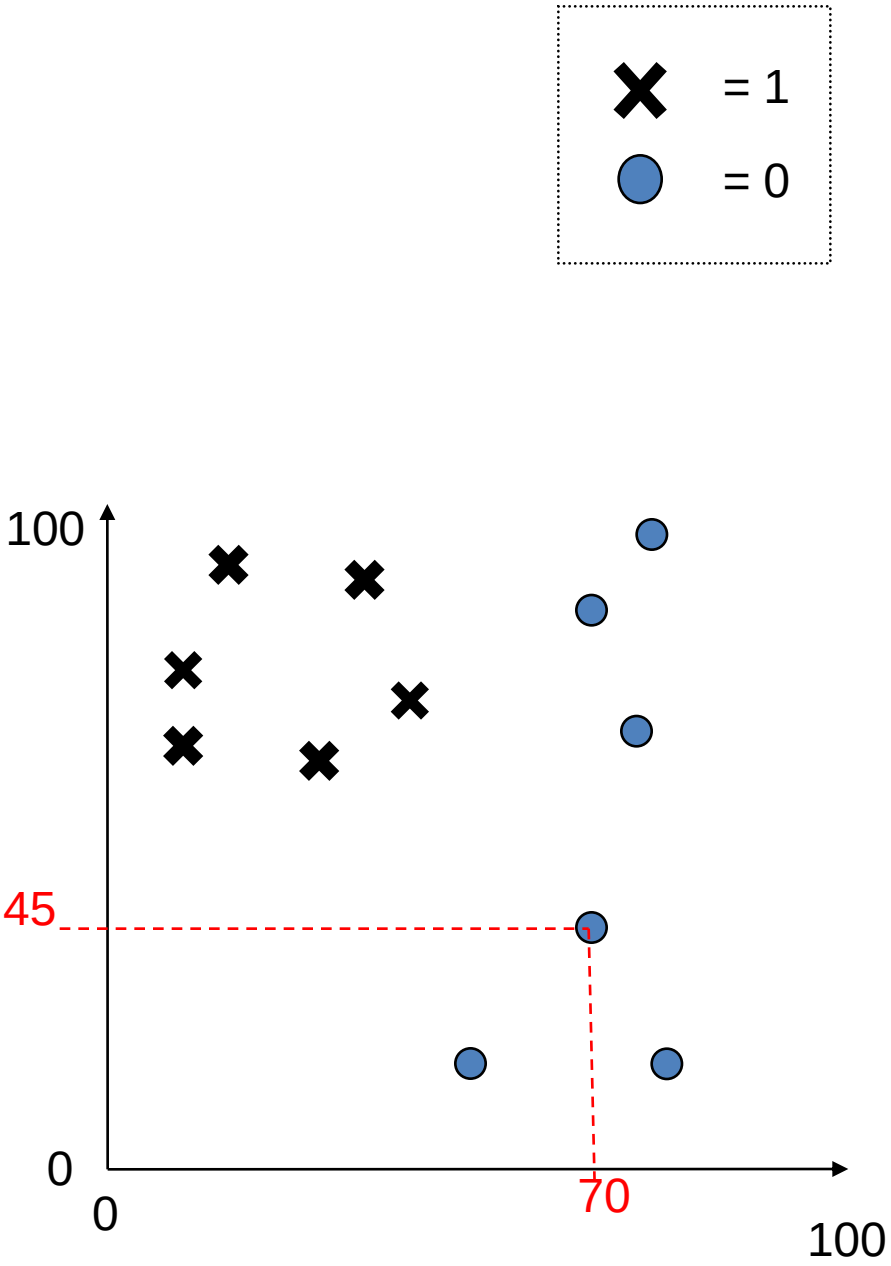
Inputs	
15	95
33	**90**

Labels
1
1



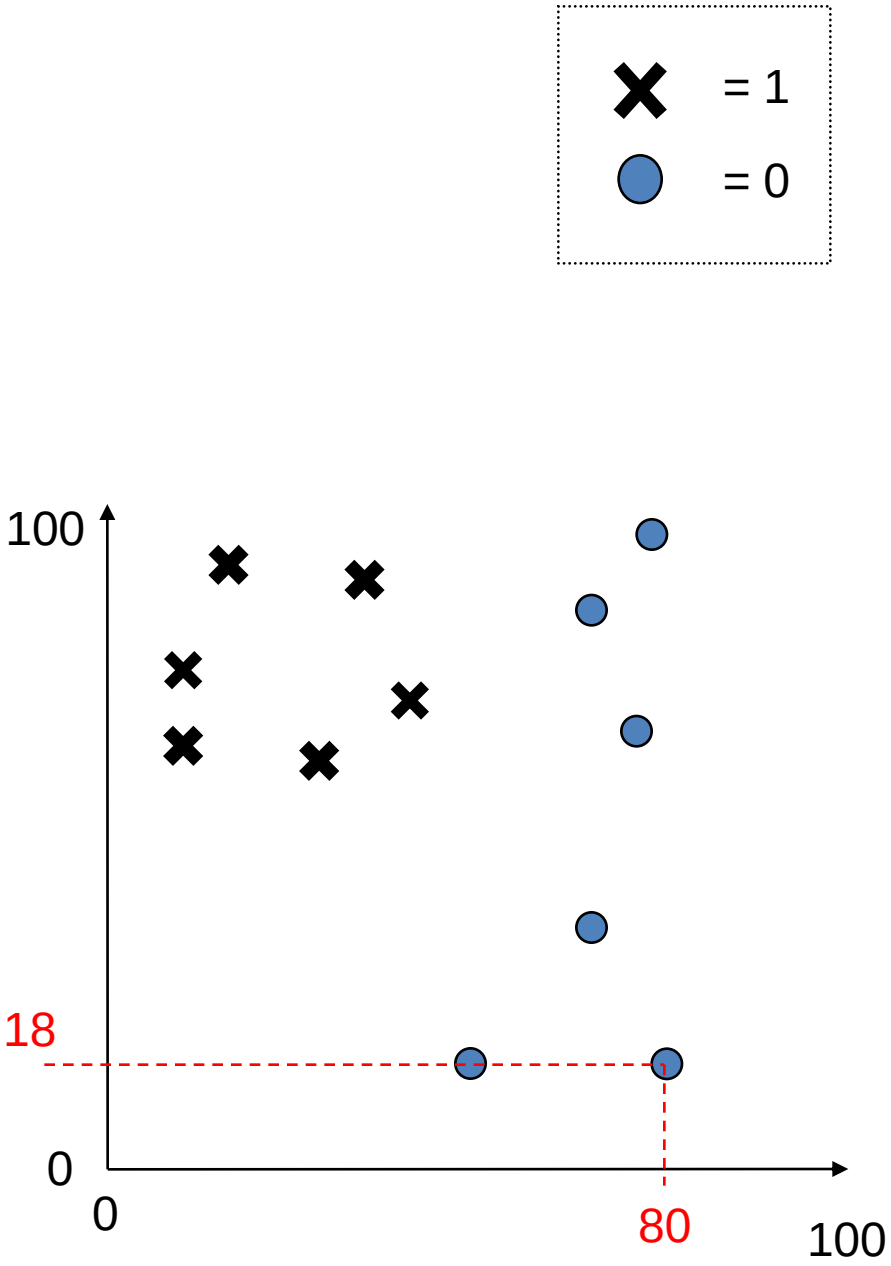
Inputs	
15	95
33	90
78	**70**

Labels
1
1
0



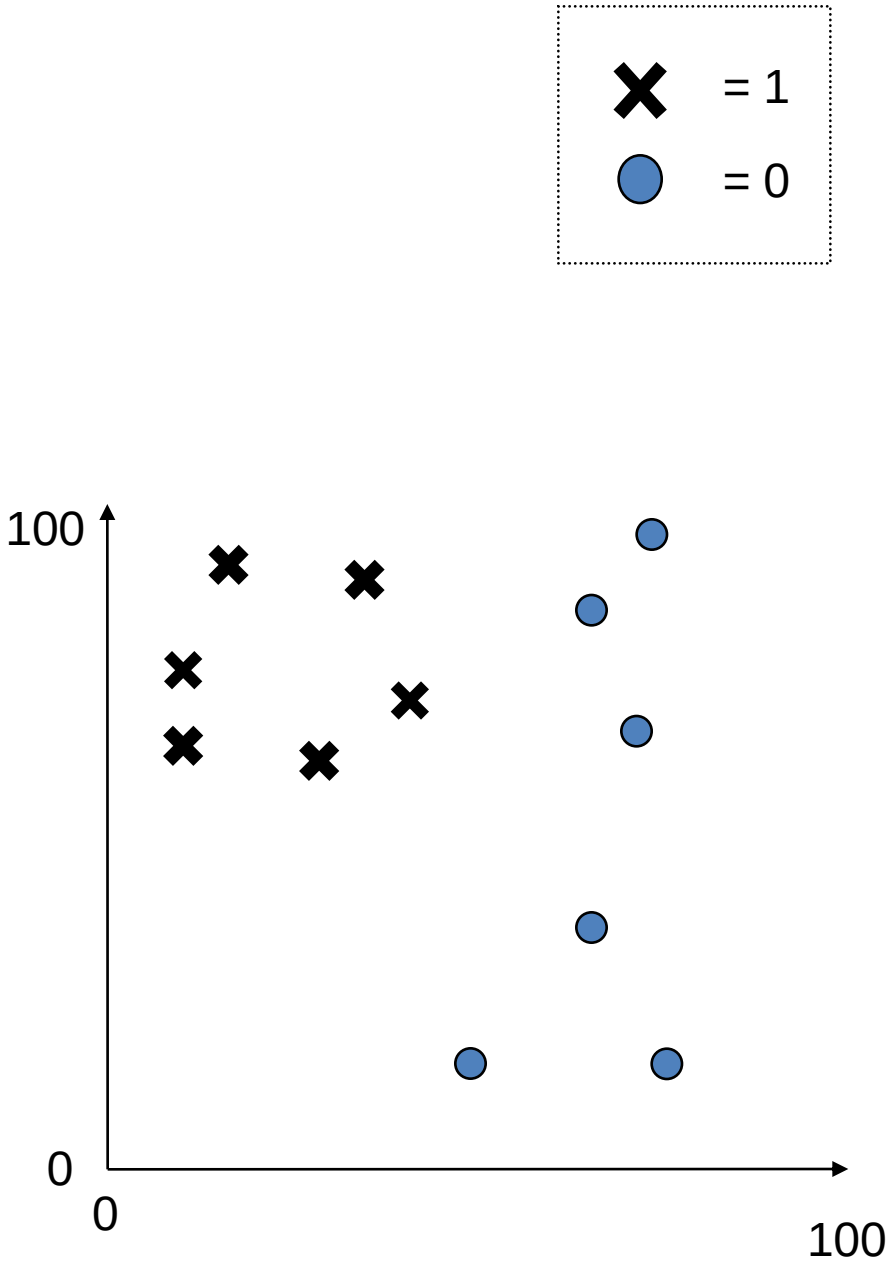
Inputs	
15	95
33	90
78	70
70	**45**

Labels
1
1
0
0



Inputs	
15	95
33	90
78	70
70	45
80	**18**

Labels
1
1
0
0
0

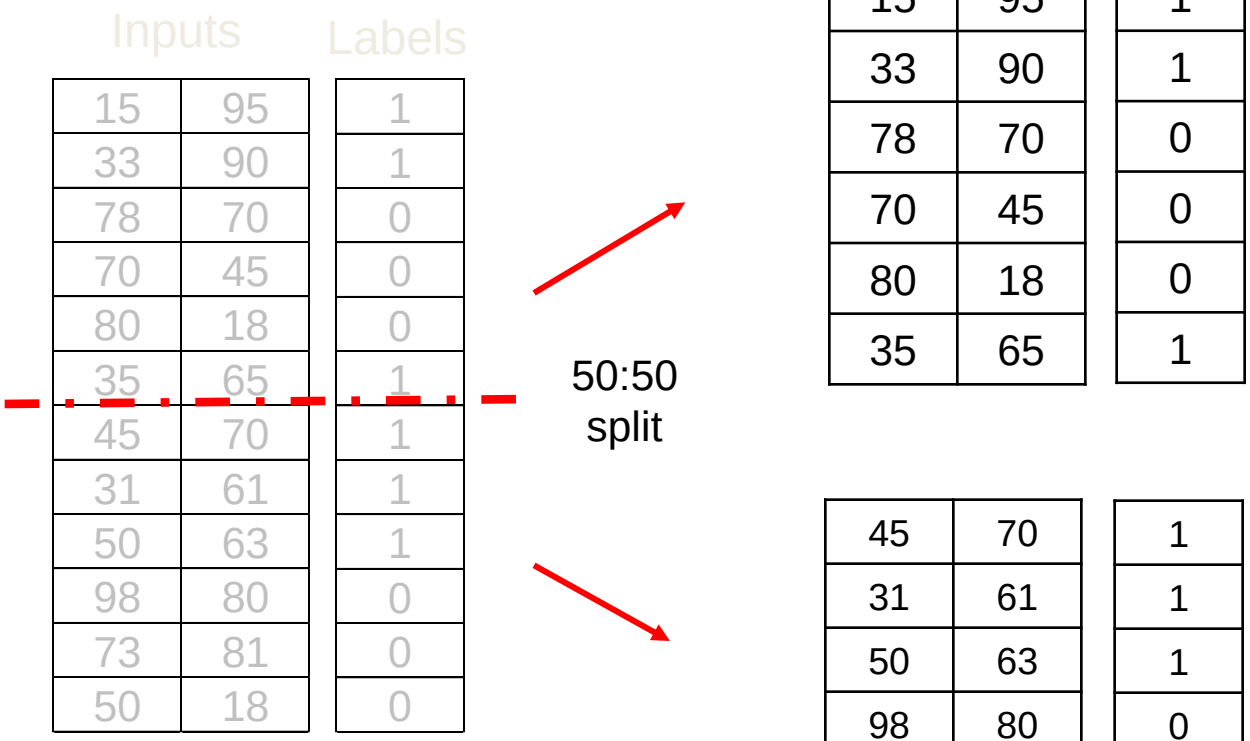


Inputs		Labels	
15	95	1	✖
33	90	1	✖
78	70	0	●
70	45	0	●
80	18	0	●
35	65	1	✖
45	70	1	✖
31	61	1	✖
50	63	1	✖
98	80	0	●
73	81	0	●
50	18	0	●

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

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


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

Ideally should be small. Smaller is better.

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

Needs to be quite big. Bigger is better.

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

Training set

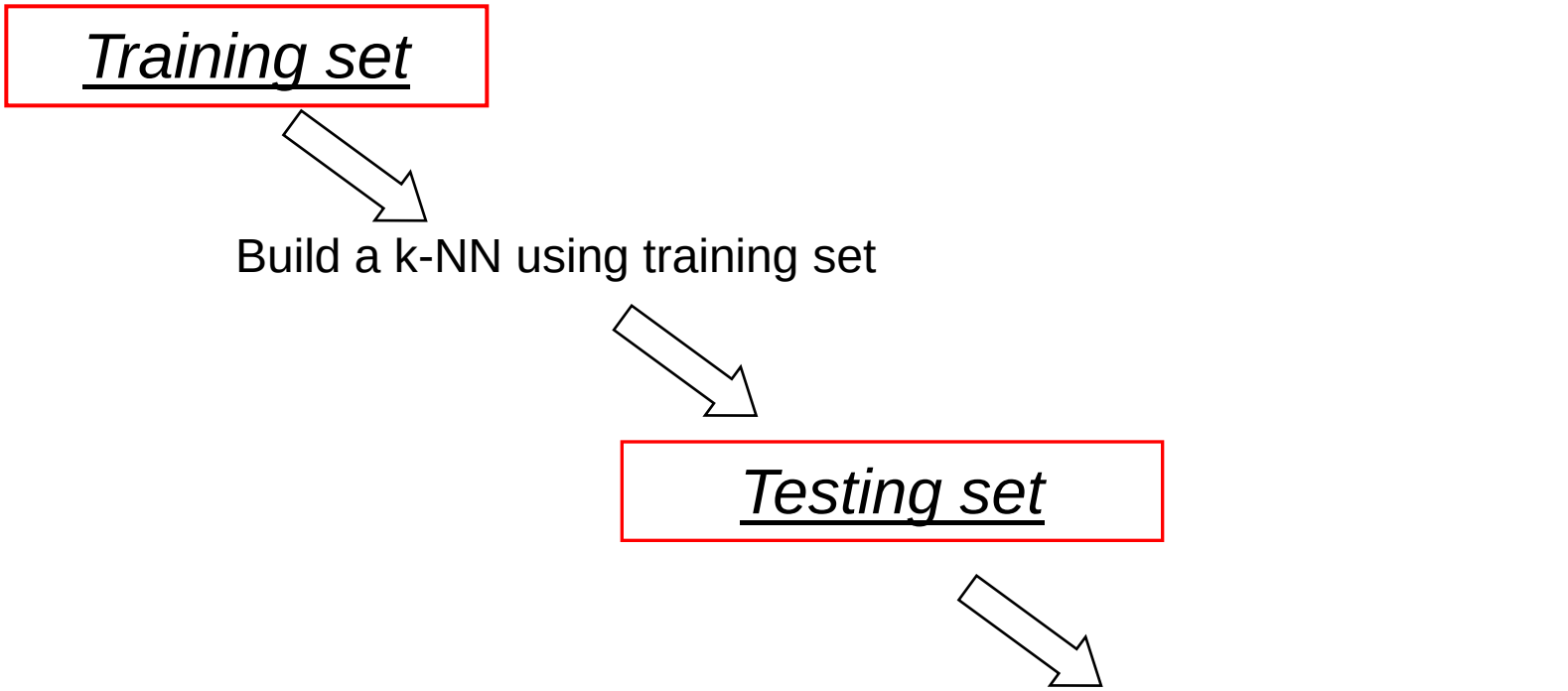
Train a K-NN on this…

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



Percentage of incorrect predictions is called the “error”

e.g. “Training” error
e.g. “Testing” error

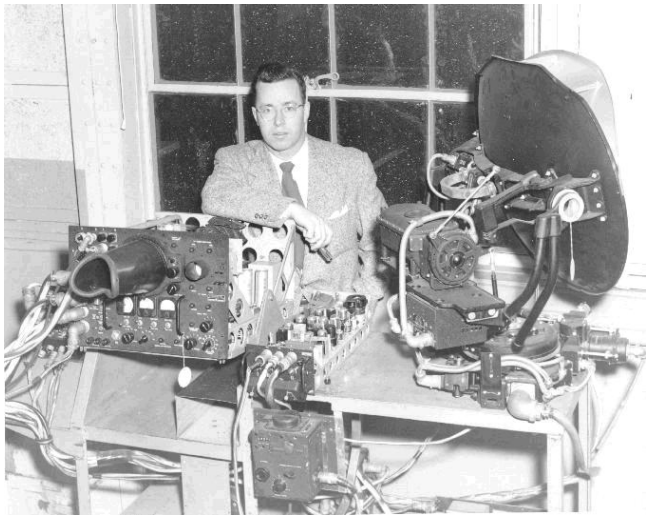
R.O.C. Analysis

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!)

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