Lecture -12
Machine Learning

What is Machine Learning?

- Machine Learning
  - Study of algorithms that
  - improve their performance
  - at some task
  - with experience
- Optimize a performance criterion using example data or past experience.
- Role of Statistics: Inference from a sample
- Role of Computer science: Efficient algorithms to
  - Solve the optimization problem
  - Representing and evaluating the model for inference

What We Talk About When We Talk About“Learning”

- Learning general models from a data of particular examples
- Example in retail: Customer transactions to consumer behavior:

  People who bought “Da Vinci Code” also bought “The Five People You Meet in Heaven” (www.amazon.com)

- Build a model that is *a good and useful approximation* to the data.

Categories

- Association Analysis
- Supervised Learning
  - Classification
  - Regression/Prediction
- Unsupervised Learning

Learning Associations

- Basket analysis:

  $P(Y \mid X)$ probability that somebody who buys X also buys Y where X and Y are products/services.

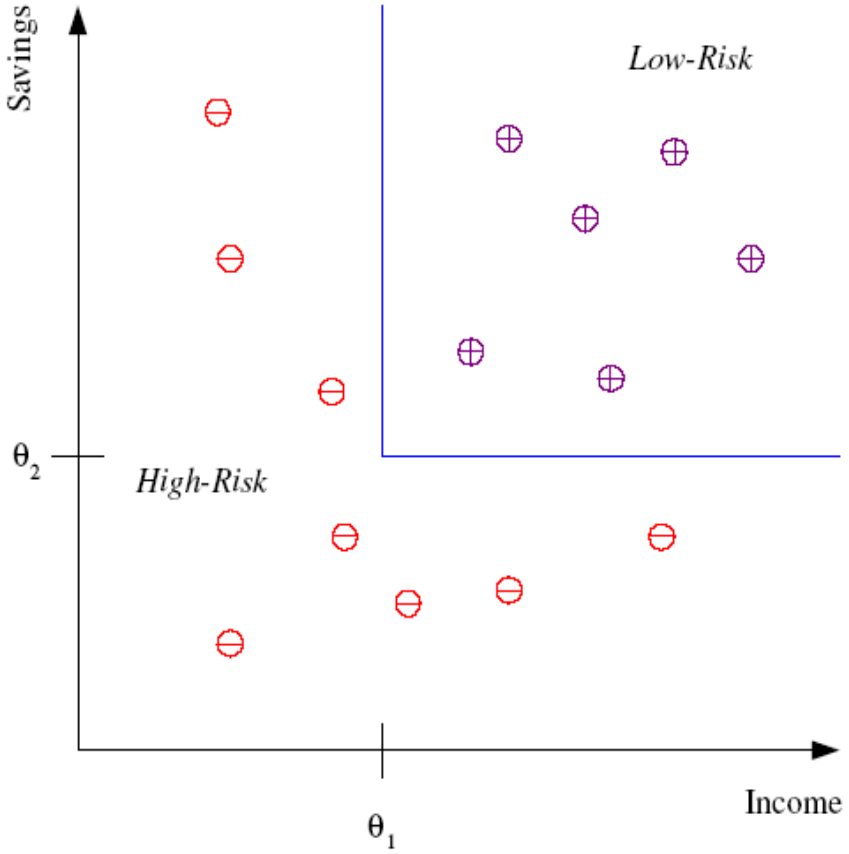
  Example: P (bread |cold drink) = 0.7

Market-Basket transactions

TID	*Items*
1	Bread, Milk
2	Bread, Diaper, Cold Drink, Eggs
3	Milk, Diaper, Cold Drink
4	Bread, Milk, Diaper, Cold Drink
5	Bread, Milk, Diaper, Water

Classification

- Example: Credit scoring
- Differentiating between low-risk and high-risk customers from their *income* and *savings*



Discriminant: IF *income* $> \theta_1$ AND *savings* $> \theta_2$
THEN low-risk ELSE high-risk

Prediction: Regression

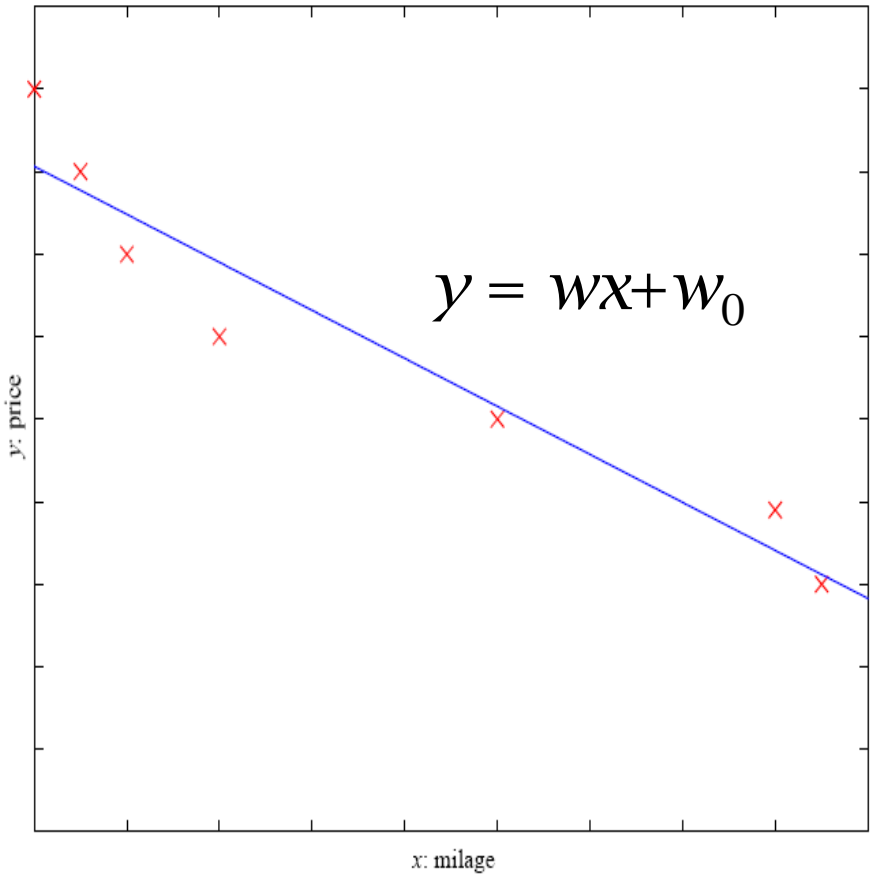
- Example: Price of a used car
- x : car attributes

  y : price

  $$y = g(x \mid \vartheta)$$

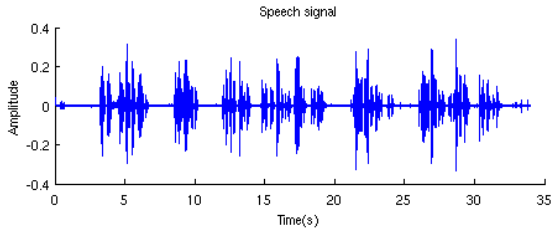
  $g(\ )$ model,

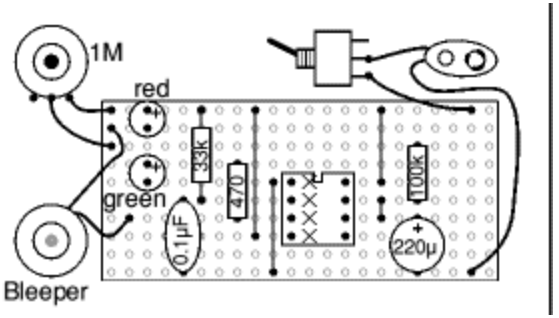
  ϑ parameters



Pattern

A pattern is the opposite of a chaos, it is an entity that can be given a name





Recognition

- Identification of a pattern as a member of a category

Classification

Apples

Oranges

Classification

What is this???

- You had some training example or '*training data*'
- The examples were '*labeled*'
- You used those examples to make the kid '*learn*' the difference between an apple and an orange

Classification

Given: training images and their categories

What are the categories of these test images?

Pattern Recognition

Given an input pattern, make a decision about the "category" or "class" of the pattern

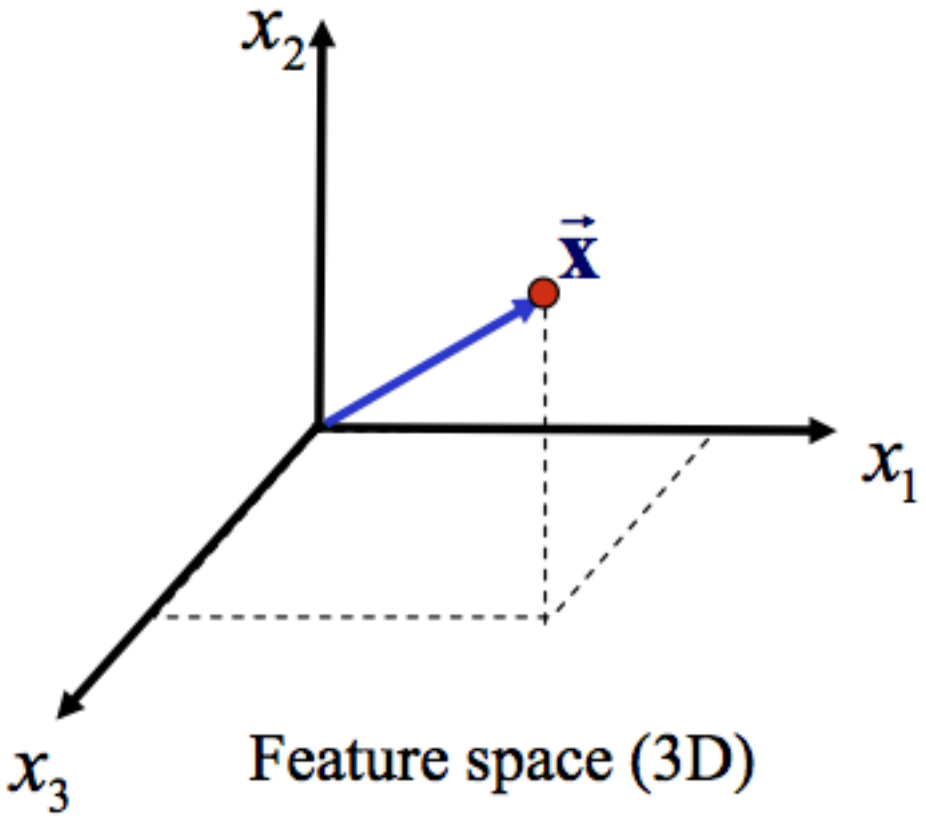
Unsupervised Learning

- Learning “what normally happens”
- No output
- Clustering: Grouping similar instances
- Other applications: Summarization, Association Analysis
- Example applications
  - Image compression: Color quantization
  - Bioinformatics: Learning motifs

Features

- **Features**: a set of variables believed to carry discriminating and characterizing information about the objects under consideration
- **Feature vector**: A collection of d features, ordered in some meaningful way into a d- dimensional column vector, that represents the signature of the object to be identified.
- **Feature space**: The d-dimensional space in which the feature vectors lie. A d-dimensional vector in a d-dimensional space constitutes a point in that space.

Features



Features

- Feature Choice
  - Good Features
    - Ideally, for a given group of patterns coming from the same class, feature values should all be similar
    - For patterns coming from different classes, the feature values should be different.
  - Bad Features
    - irrelevant, noisy, outlier?

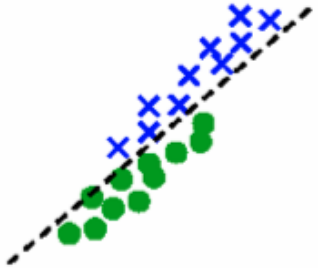
Features

"Good" features

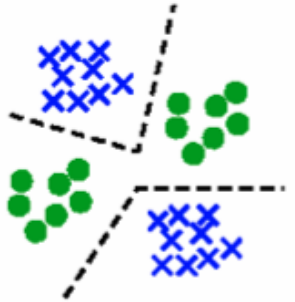
"Bad" features

Linear separability

Non-linear separability



Highly correlated features



Multi-modal

Model Choice

- What type of *classifier* shall we use? How shall we select its parameters? Is there best classifier...?
- How do we train...? How do we adjust the parameters of the model (*classifier*) we picked so that the model fits the data?

Resources: Datasets

- UCI Repository: http://www.ics.uci.edu/~mlearn/MLRepository.html
- UCI KDD Archive: http://kdd.ics.uci.edu/summary.data.application.html
- Statlib: http://lib.stat.cmu.edu/
- Delve: http://www.cs.utoronto.ca/~delve/

Resources: Journals

- Journal of Machine Learning Research www.jmlr.org
- Machine Learning
- IEEE Transactions on Neural Networks
- IEEE Transactions on Pattern Analysis and Machine Intelligence
- Annals of Statistics
- Journal of the American Statistical Association
- …

Resources: Conferences

- International Conference on Machine Learning (ICML)
- European Conference on Machine Learning (ECML)
- Neural Information Processing Systems (NIPS)
- Computational Learning
- International Joint Conference on Artificial Intelligence (IJCAI)
- ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)
- IEEE Int. Conf. on Data Mining (ICDM)

The Major Machine Learning Tasks

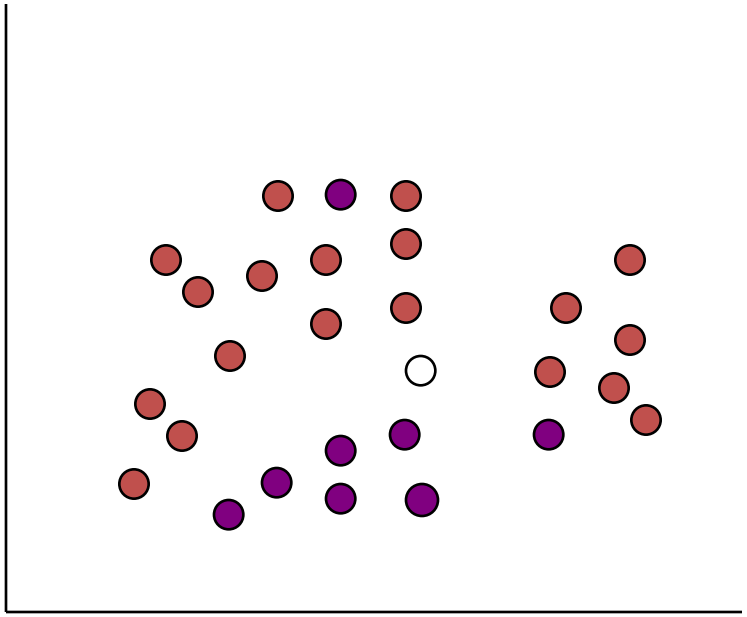
- Classification
- Clustering

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…

Classification

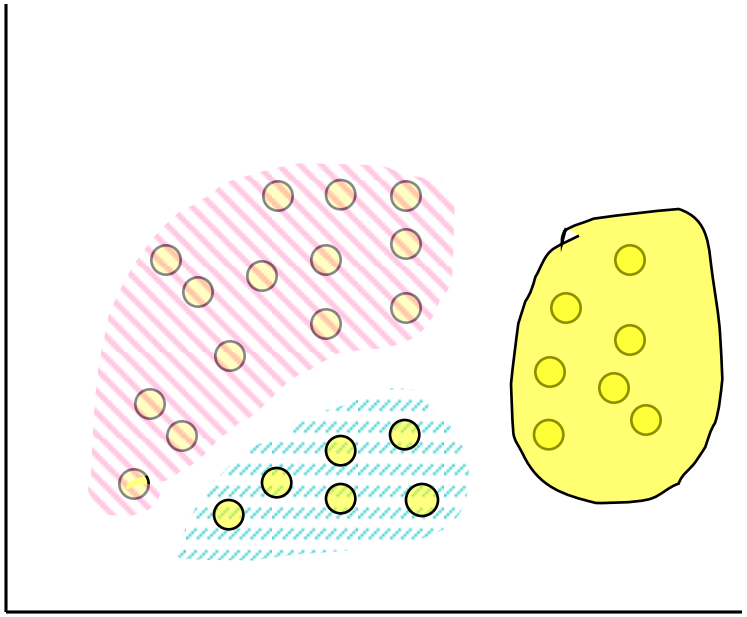
Learn a method for predicting the instance class from pre-labeled (classified) instances



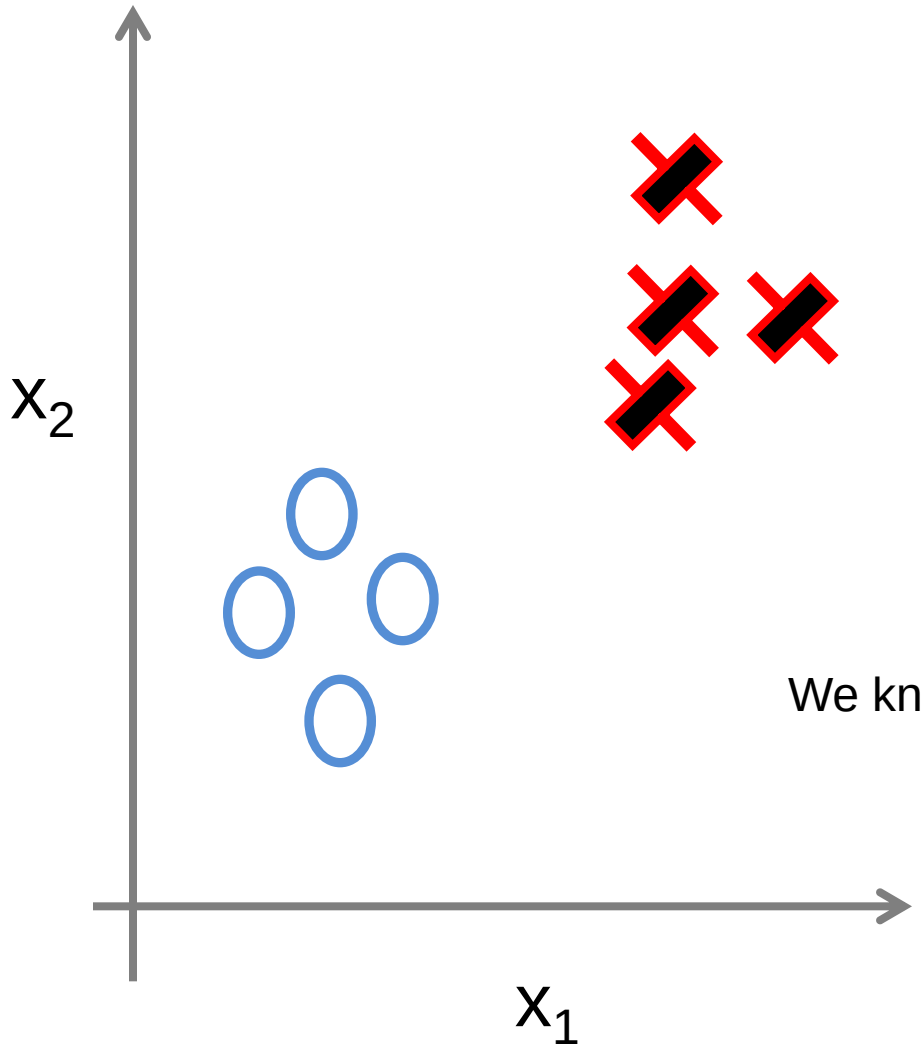
Many approaches:
Statistics,
Decision Trees, Neural Networks,
...

Clustering

Find "natural" grouping of instances given un-labeled data

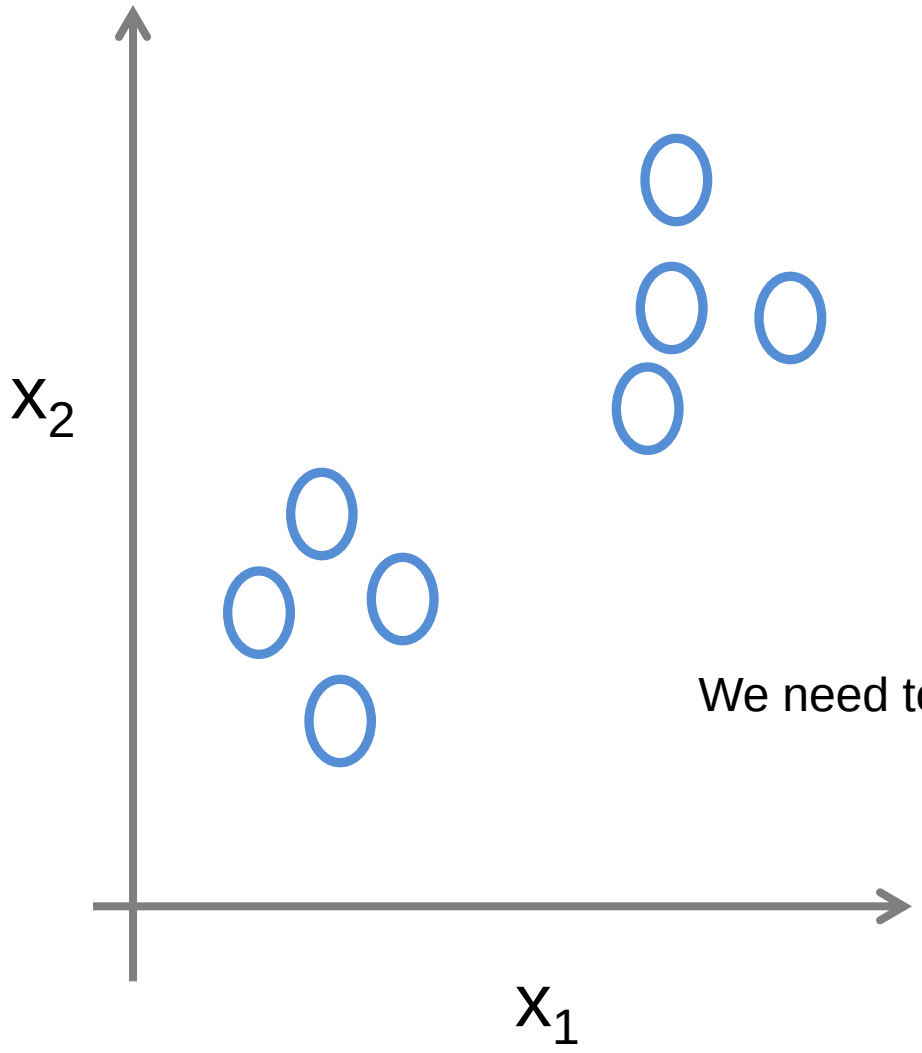


Supervised Learning



We knew the correct answers

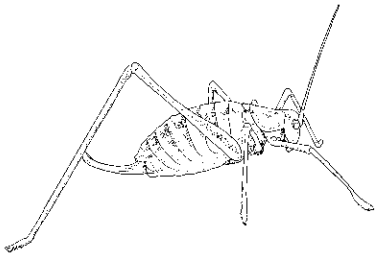
Unsupervised Learning



We need to figure out the patterns

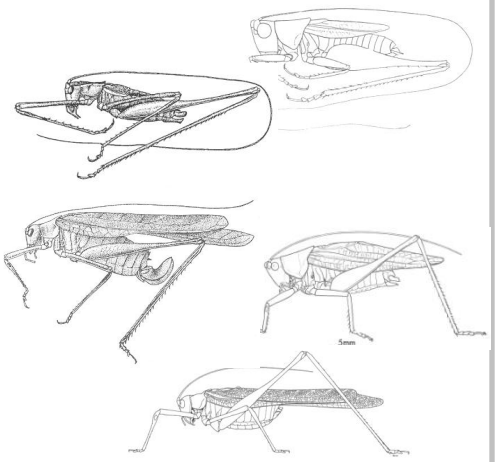
The Classification Problem
(informal definition)

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.

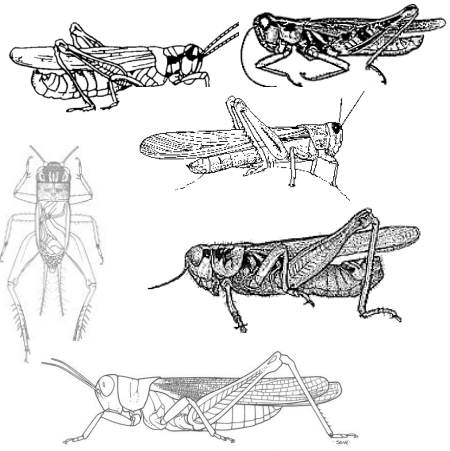


Katydid or **Grasshopper**?

Katydids



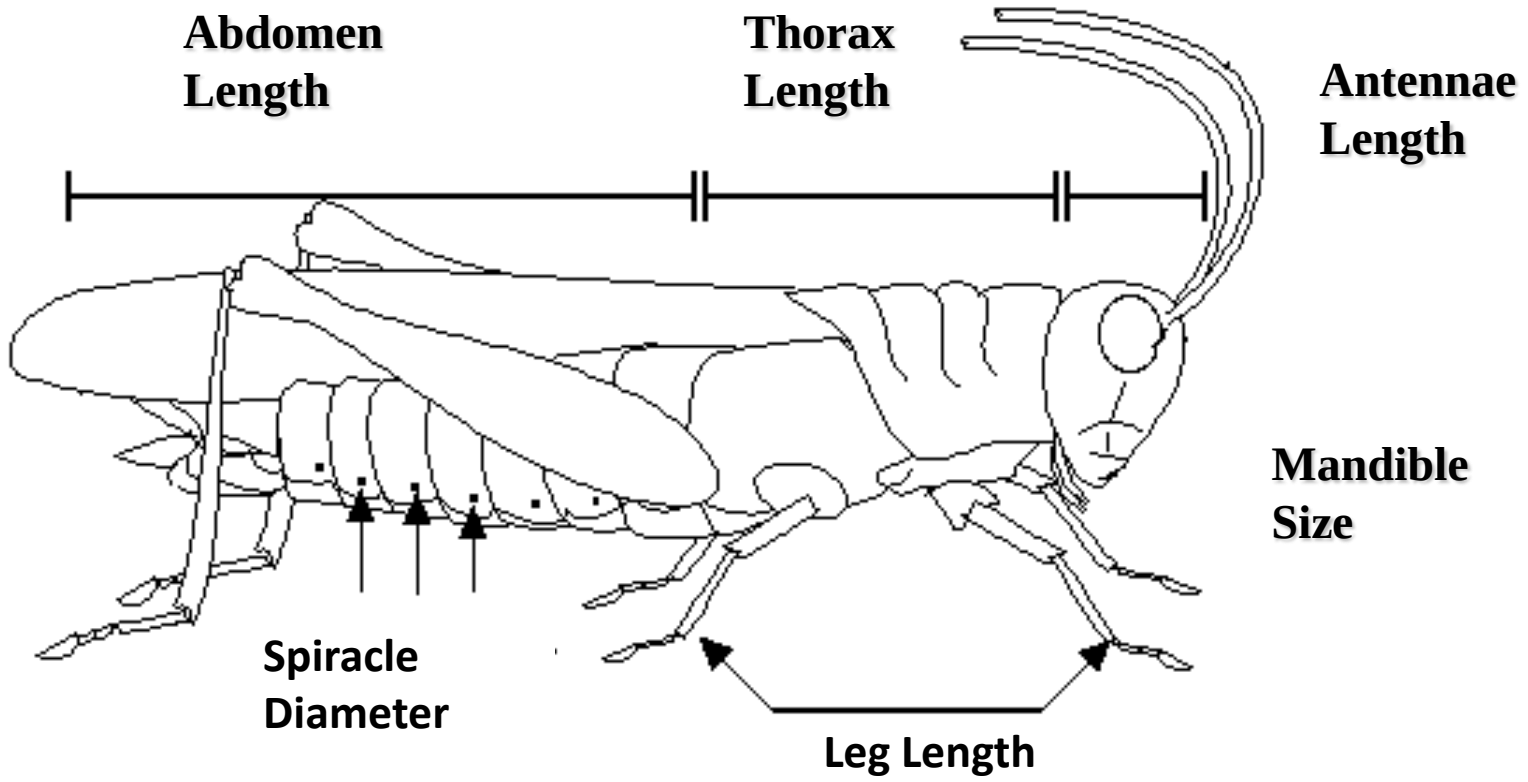
Grasshoppers



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

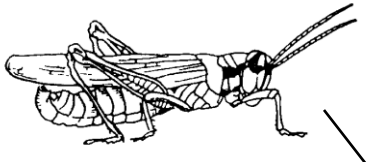
Insect ID	Abdomen Length	Antennae Length	Insect Class
1	2.7	5.5	**Grasshopper**
2	8.0	9.1	**Katydid**
3	0.9	4.7	**Grasshopper**
4	1.1	3.1	**Grasshopper**
5	5.4	8.5	**Katydid**
6	2.9	1.9	**Grasshopper**
7	6.1	6.6	**Katydid**
8	0.5	1.0	**Grasshopper**
9	8.3	6.6	**Katydid**
10	8.1	4.7	**Katydids**

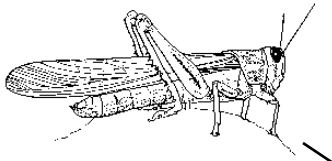
previously unseen instance =

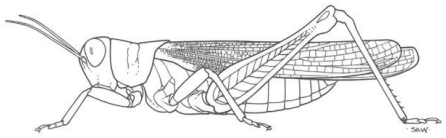
11	5.1	7.0	???????

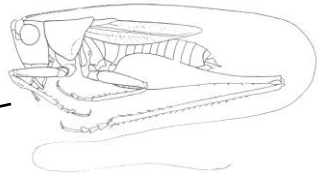
Grasshoppers

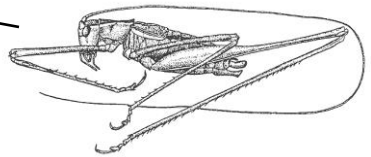
Katydids

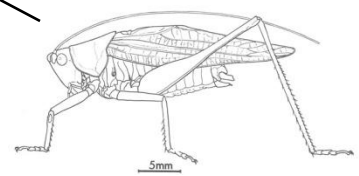


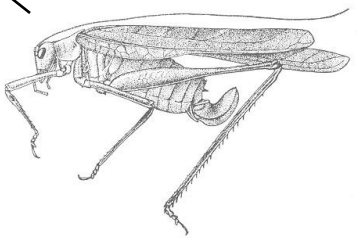


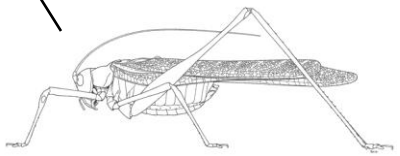


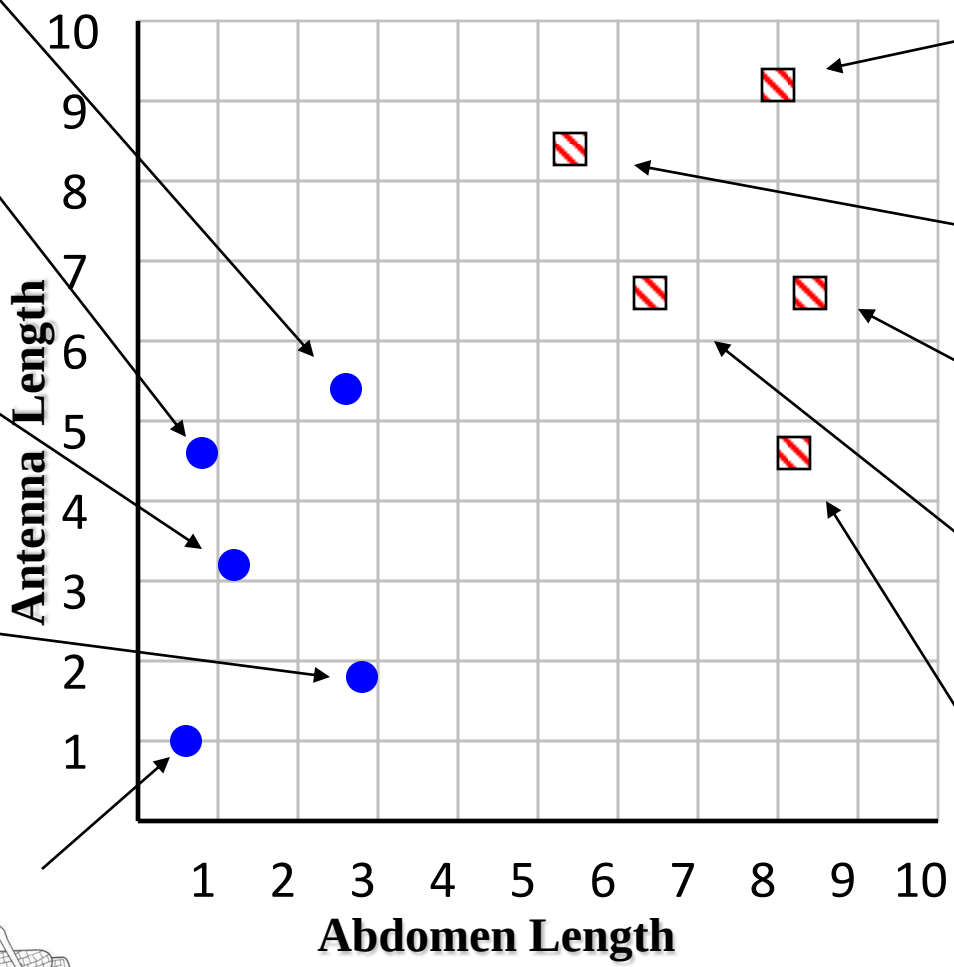


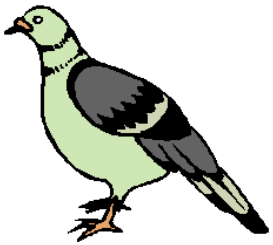












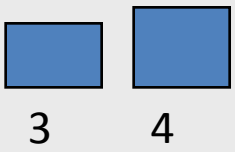
We will return to the previous slide in two minutes. In the meantime, we are going to play a quick game.

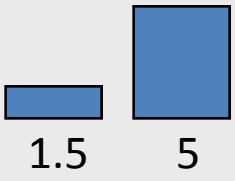
I am going to show you some classification problems which were shown to pigeons!

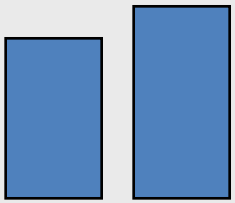
Let us see if you are as smart as a pigeon!

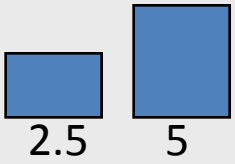
Pigeon Problem 1

Examples of class A

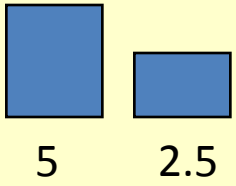


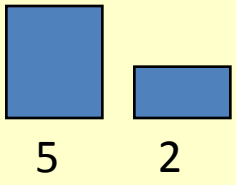


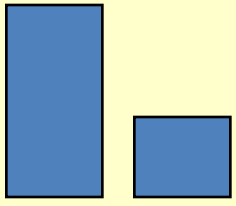


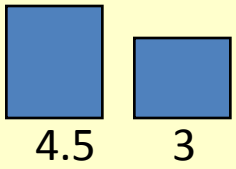


Examples of class B



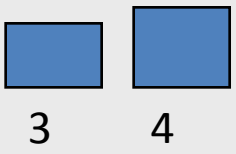


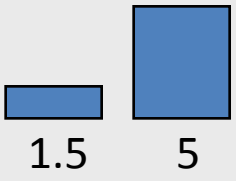


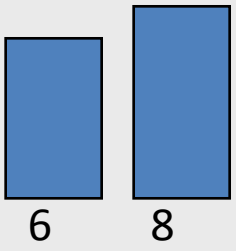


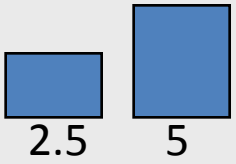
Pigeon Problem 1

Examples of class A

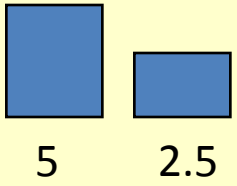


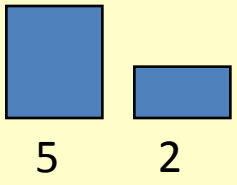


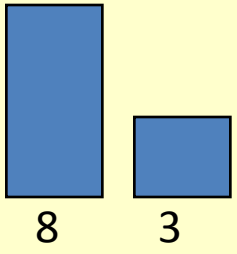


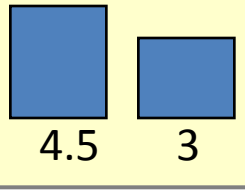


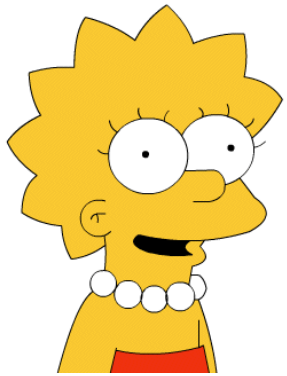
Examples of class B



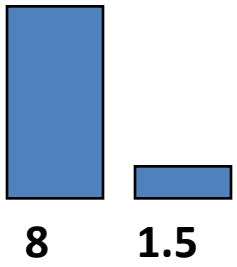




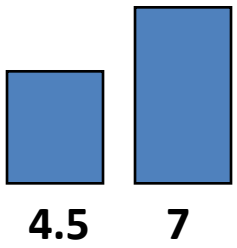




What class is this object?

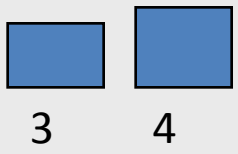


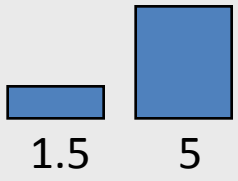
What about this one, A or B?

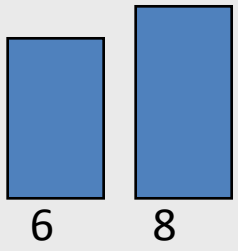


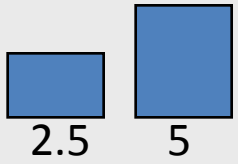
Pigeon Problem 1

Examples of class A

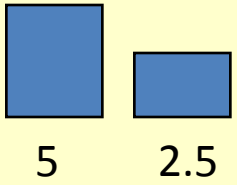


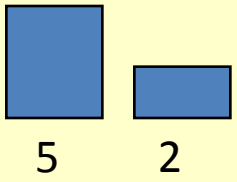


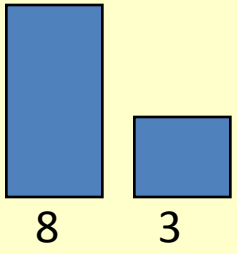


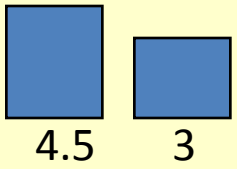


Examples of class B

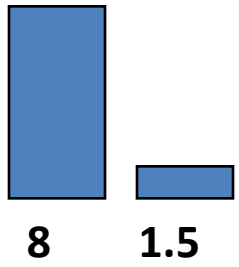








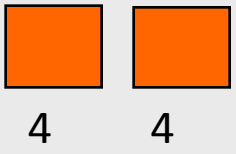
This is a B!

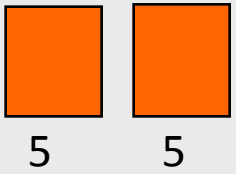


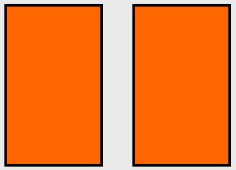
Here is the rule.
If the left bar is smaller than the right bar, it is an A, otherwise it is a B.

Pigeon Problem 2

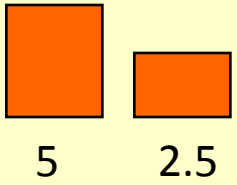
Examples of class A

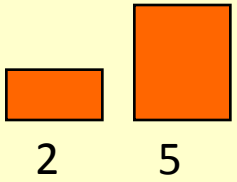


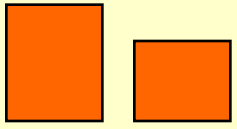


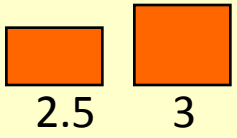


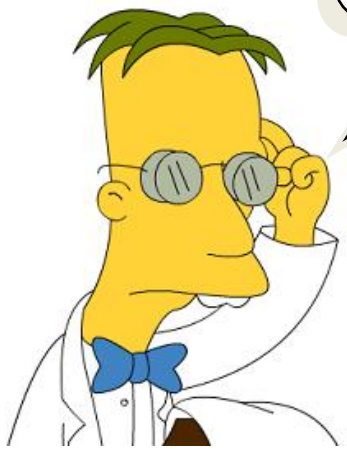
Examples of class B

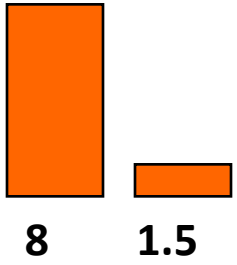


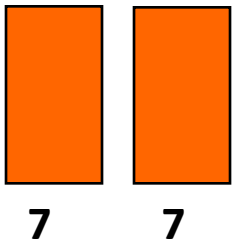






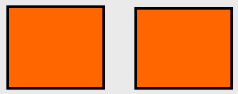


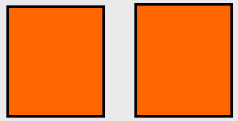


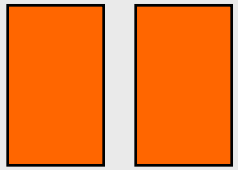


Pigeon Problem 2

Examples of class A

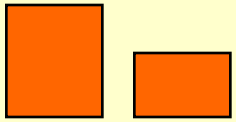

4 4

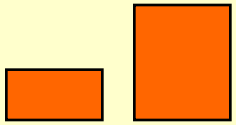

5 5

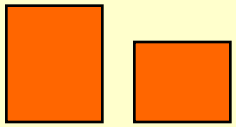

6 6

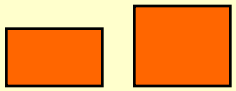

3 3

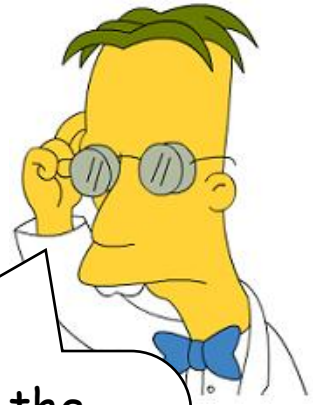
Examples of class B


5 2.5


2 5

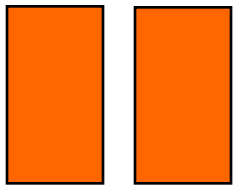

5 3


2.5 3



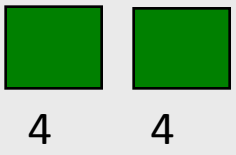
The rule is as follows, if the two bars are equal sizes, it is an A. Otherwise it is a B.

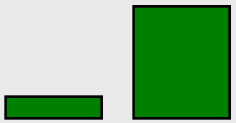
So this one is an A.

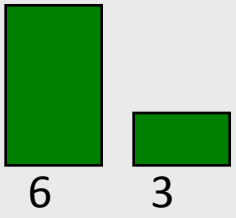

7 7

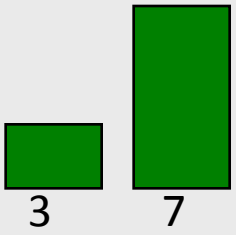
Pigeon Problem 3

Examples of class A

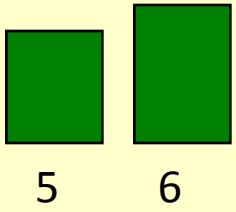


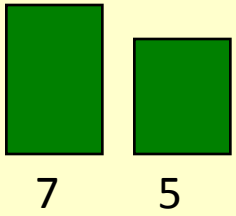


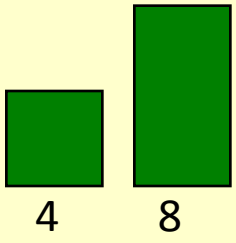


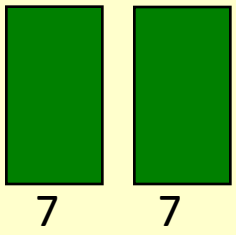


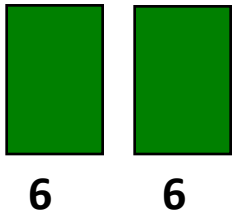
Examples of class B







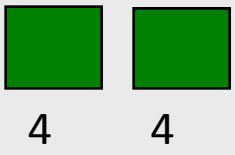


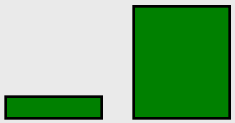


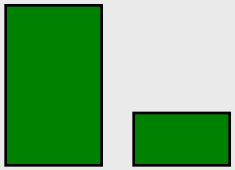
This one is really hard!
What is this, A or B?

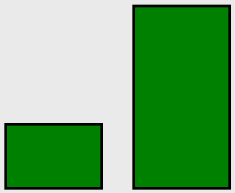
Pigeon Problem 3

Examples of class A

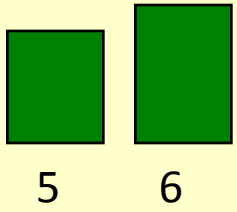


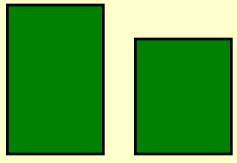


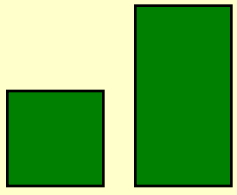


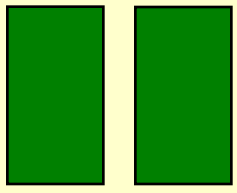


Examples of class B

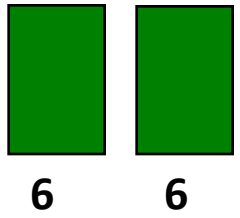








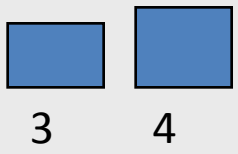
It is a B!

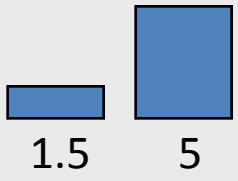


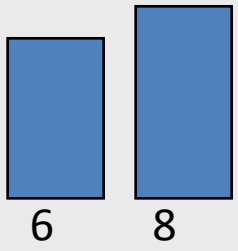
The rule is as follows, if the sum of the two bars is less than or equal to 10, it is an A. Otherwise it is a B.

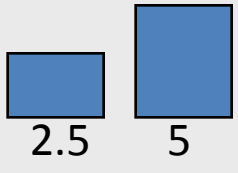
Pigeon Problem 1

Examples of class A

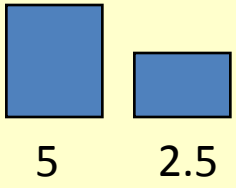


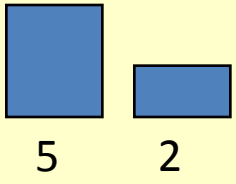


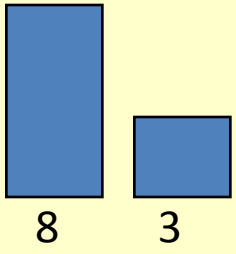


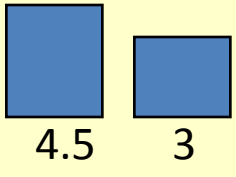


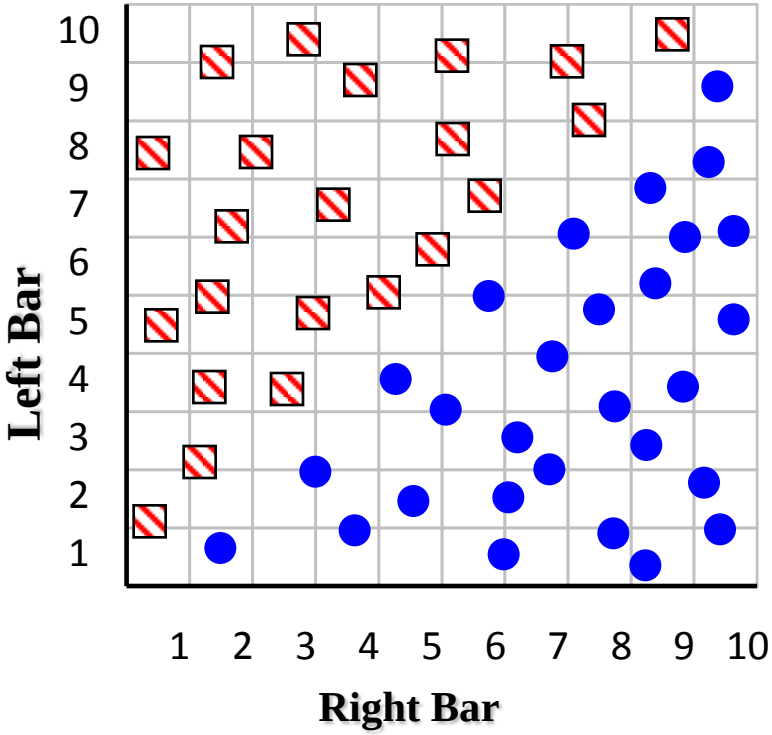
Examples of class B







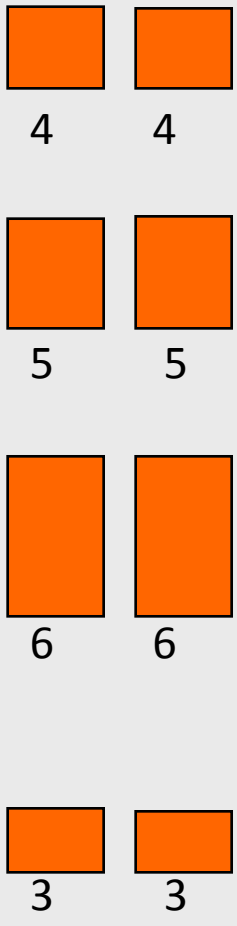




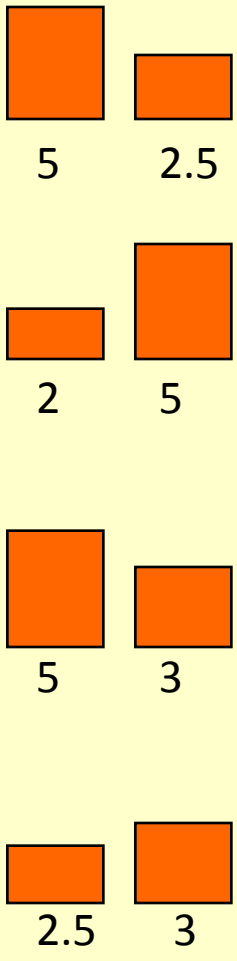
Here is the rule again. If the left bar is smaller than the right bar, it is an A, otherwise it is a B.

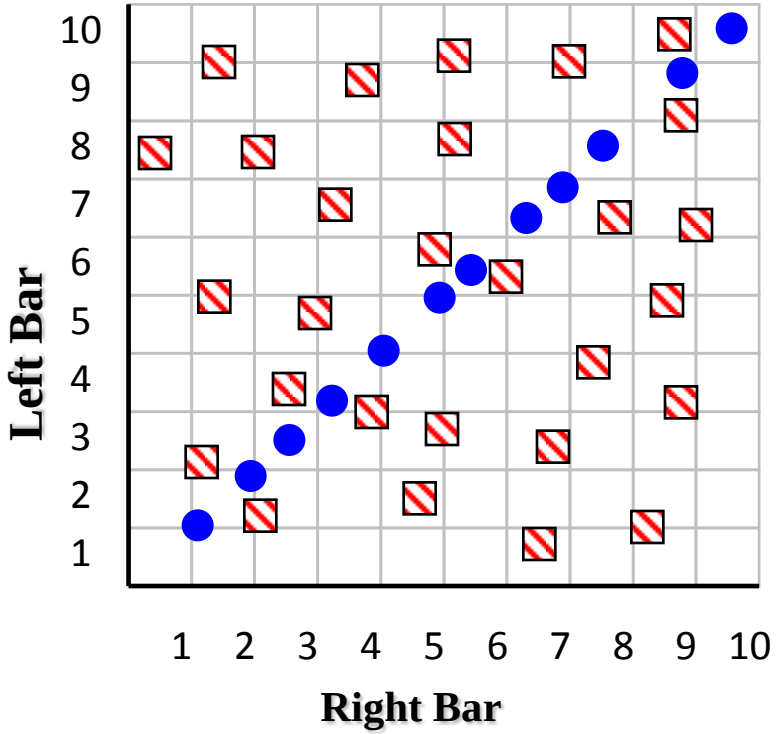
Pigeon Problem 2

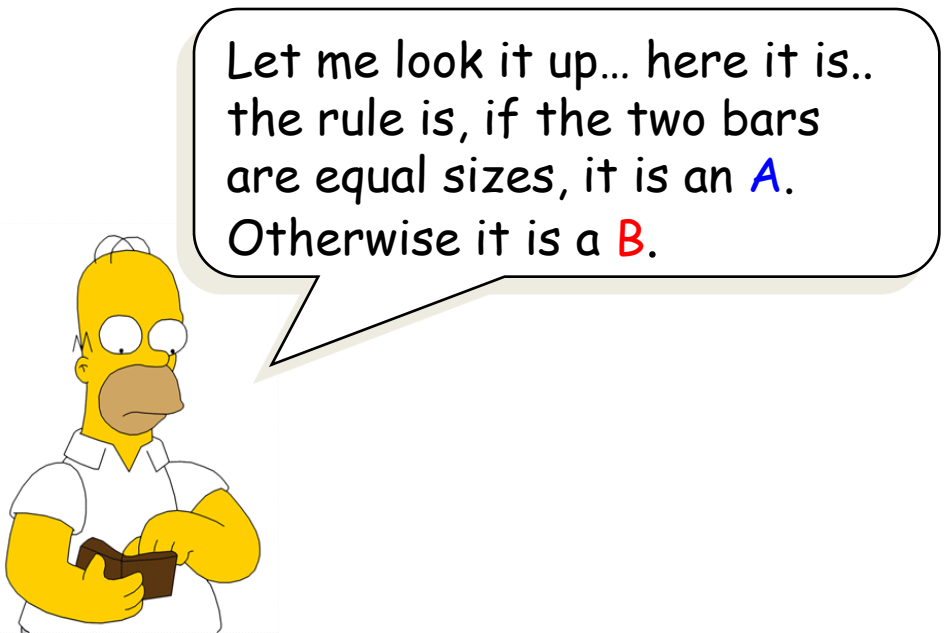
Examples of class A



Examples of class B

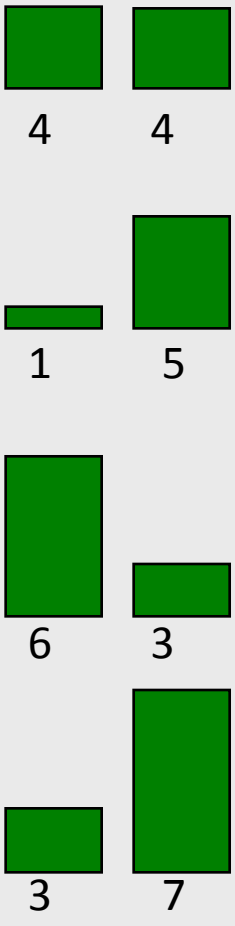




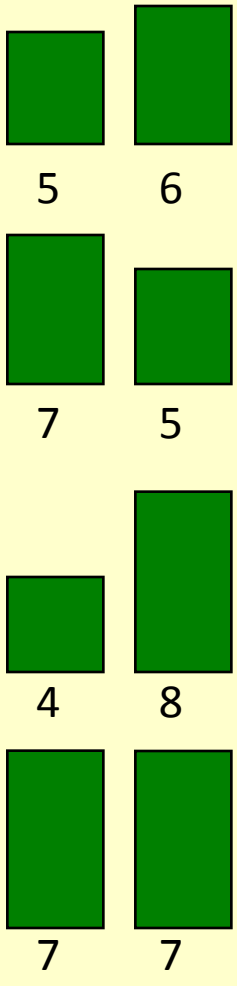


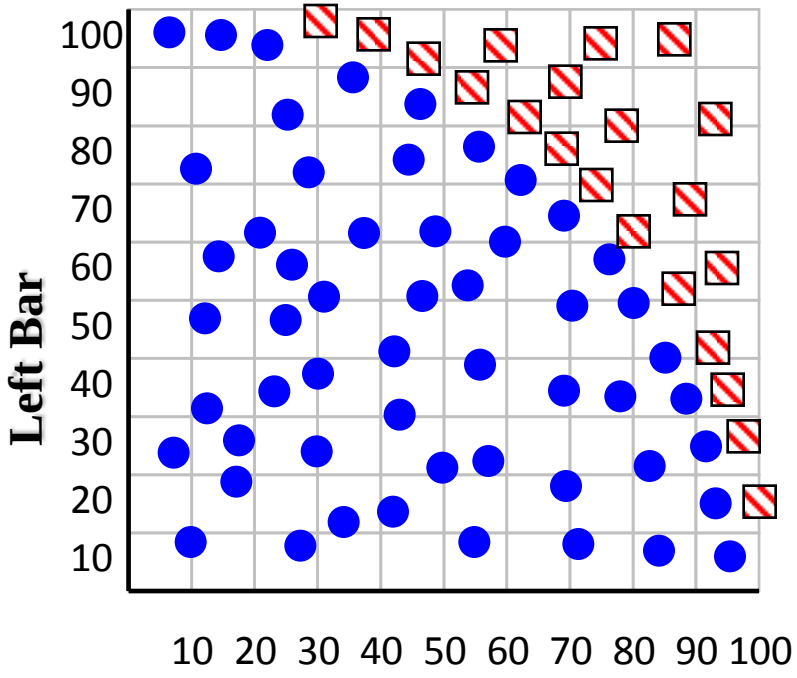
Pigeon Problem 3

Examples of class A



Examples of class B

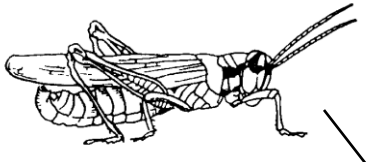


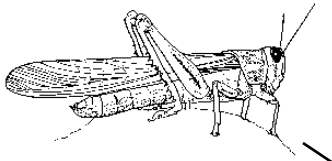


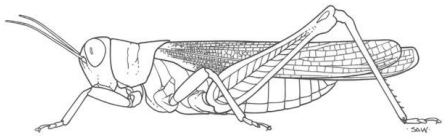
The rule again:
if the square of the sum of the two bars is less than or equal to 100, it is an A. Otherwise it is a B.

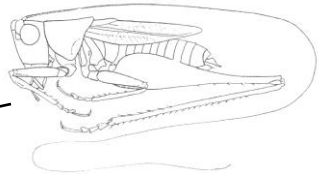
Grasshoppers

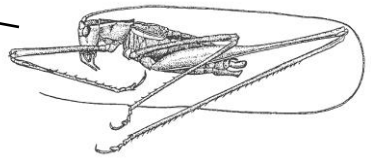
Katydids

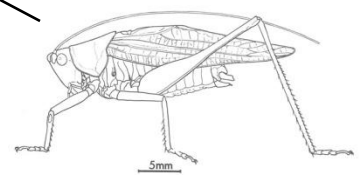


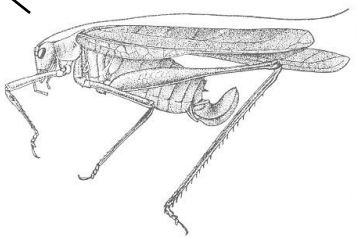


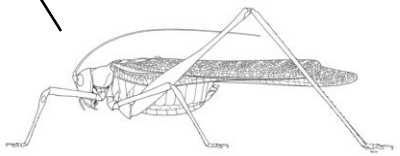


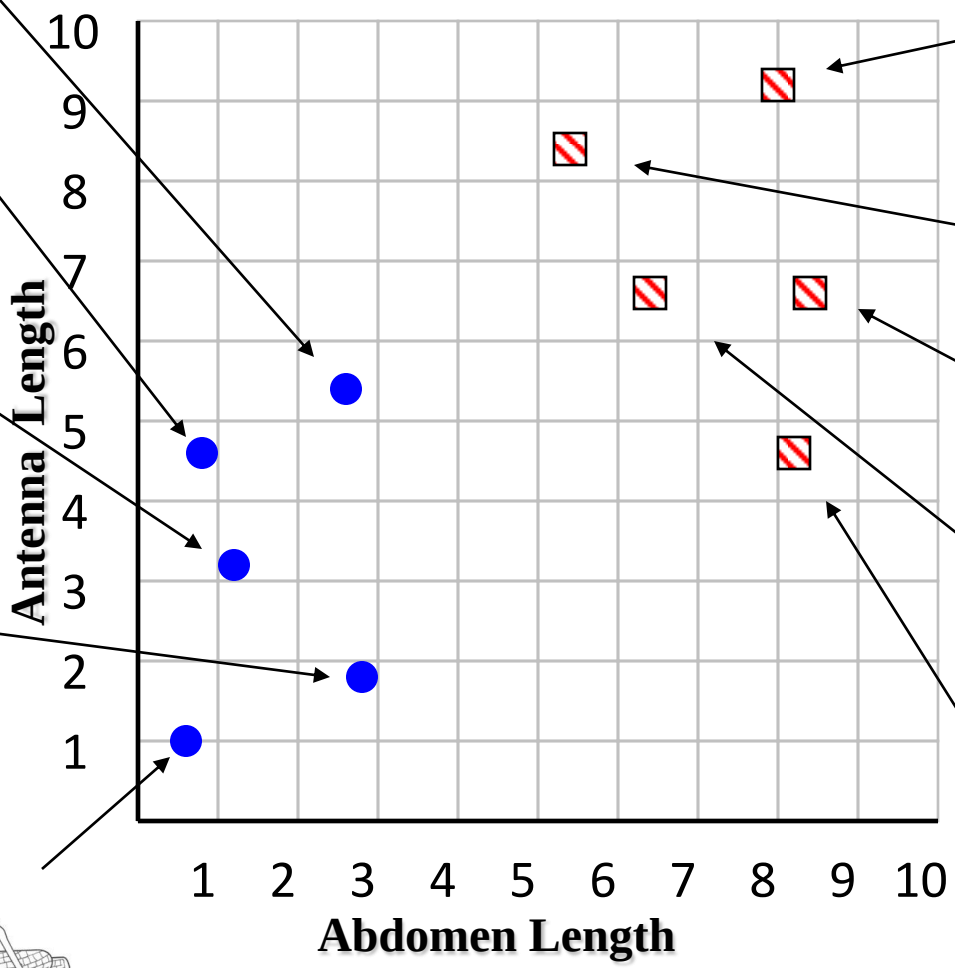






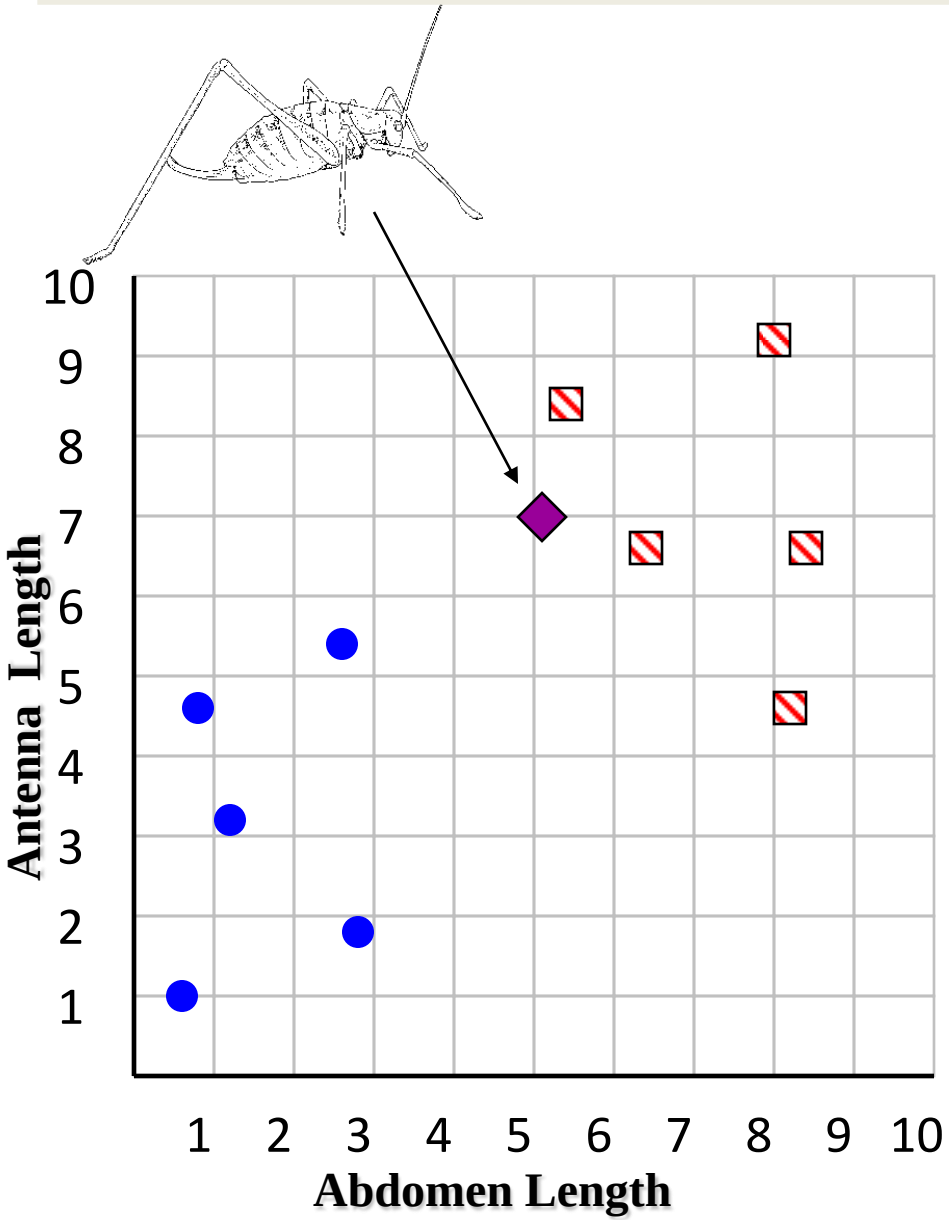






previously unseen instance =

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



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

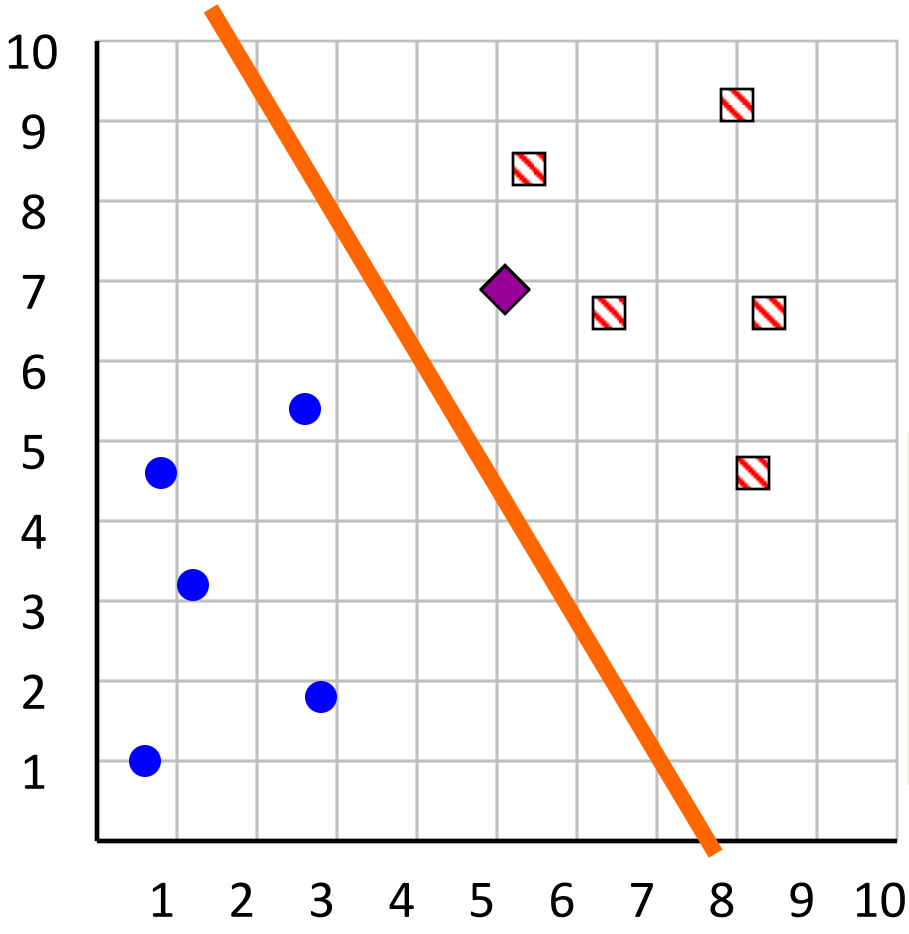
• We have now abstracted away the details of our particular problem. It will be much easier to talk about points in space.

▨ **Katydids**

● **Grasshoppers**

Simple Linear Classifier

R.A. Fisher
1890-1962

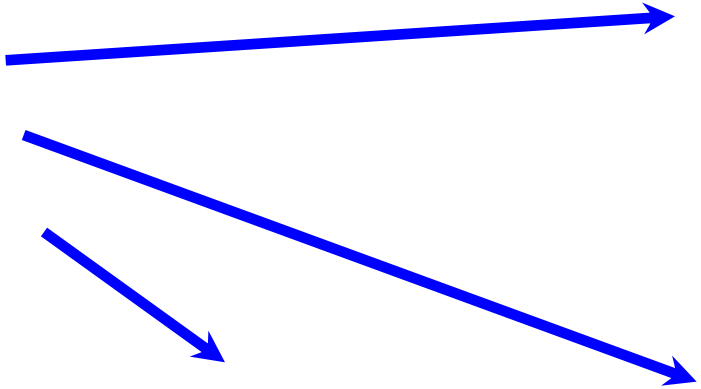


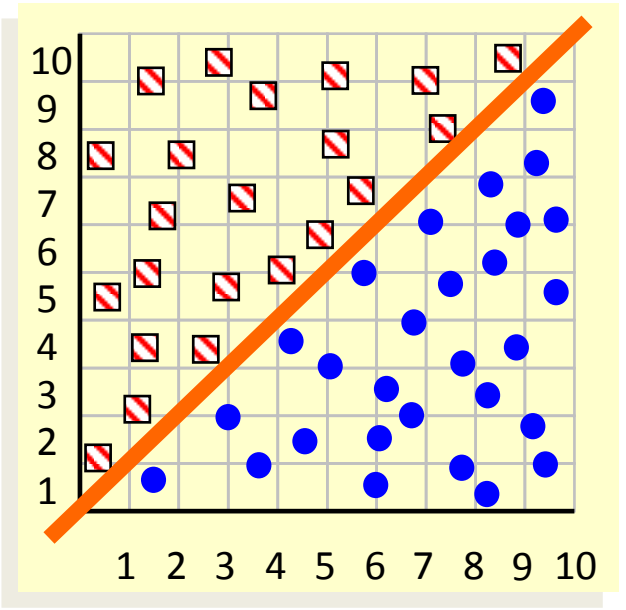
If previously unseen instance above the line
then
class is **Katydid**
else
class is **Grasshopper**

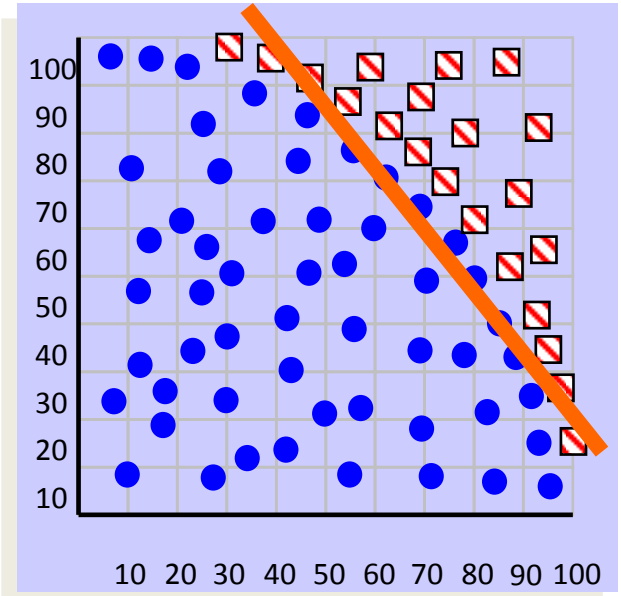
Katydids
Grasshoppers

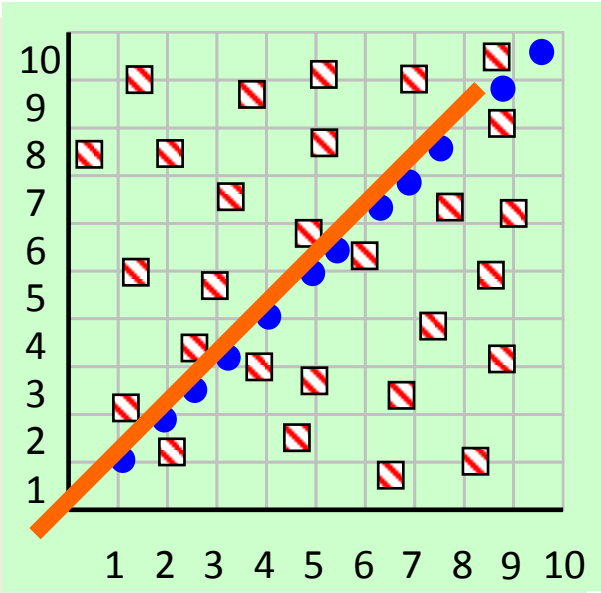
Which of the "Pigeon Problems" can be solved by the Simple Linear Classifier?

1) Perfect
2) Useless
3) Pretty Good









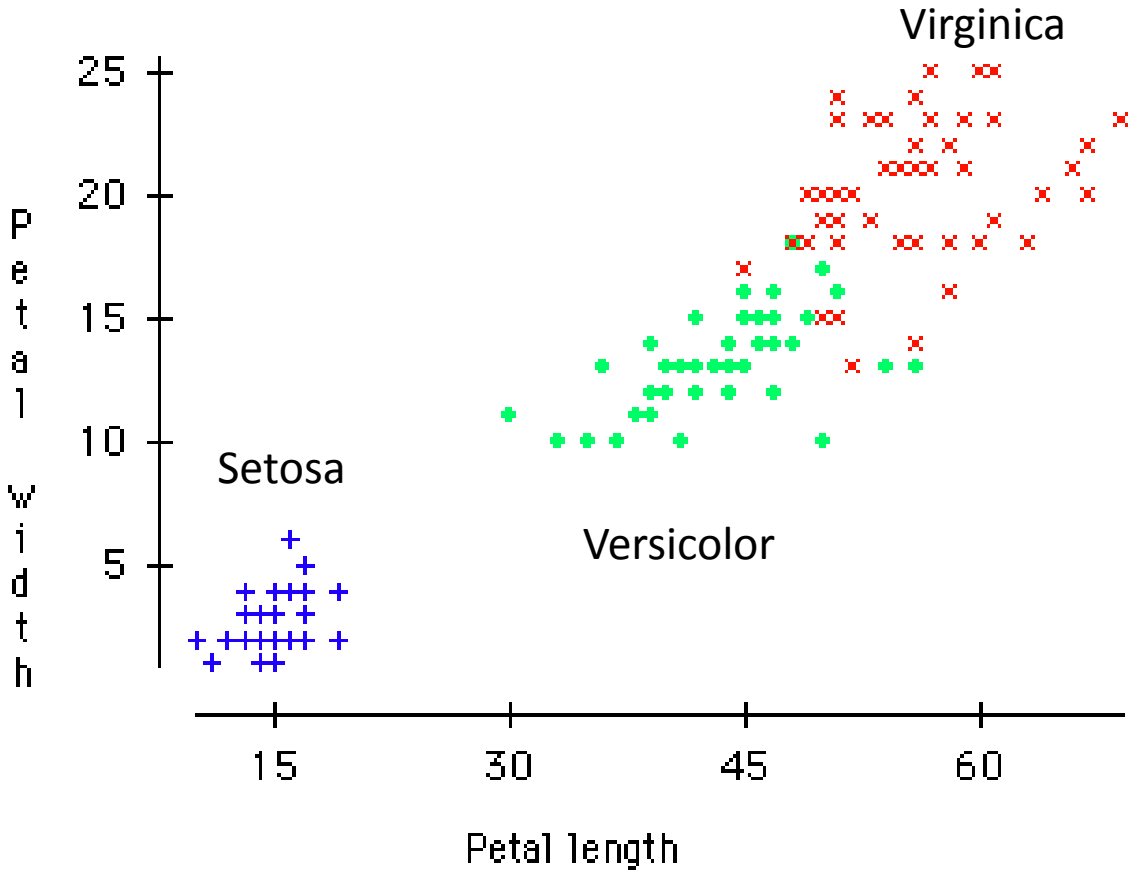
Problems that can be solved by a linear classifier are call **linearly separable**.

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.

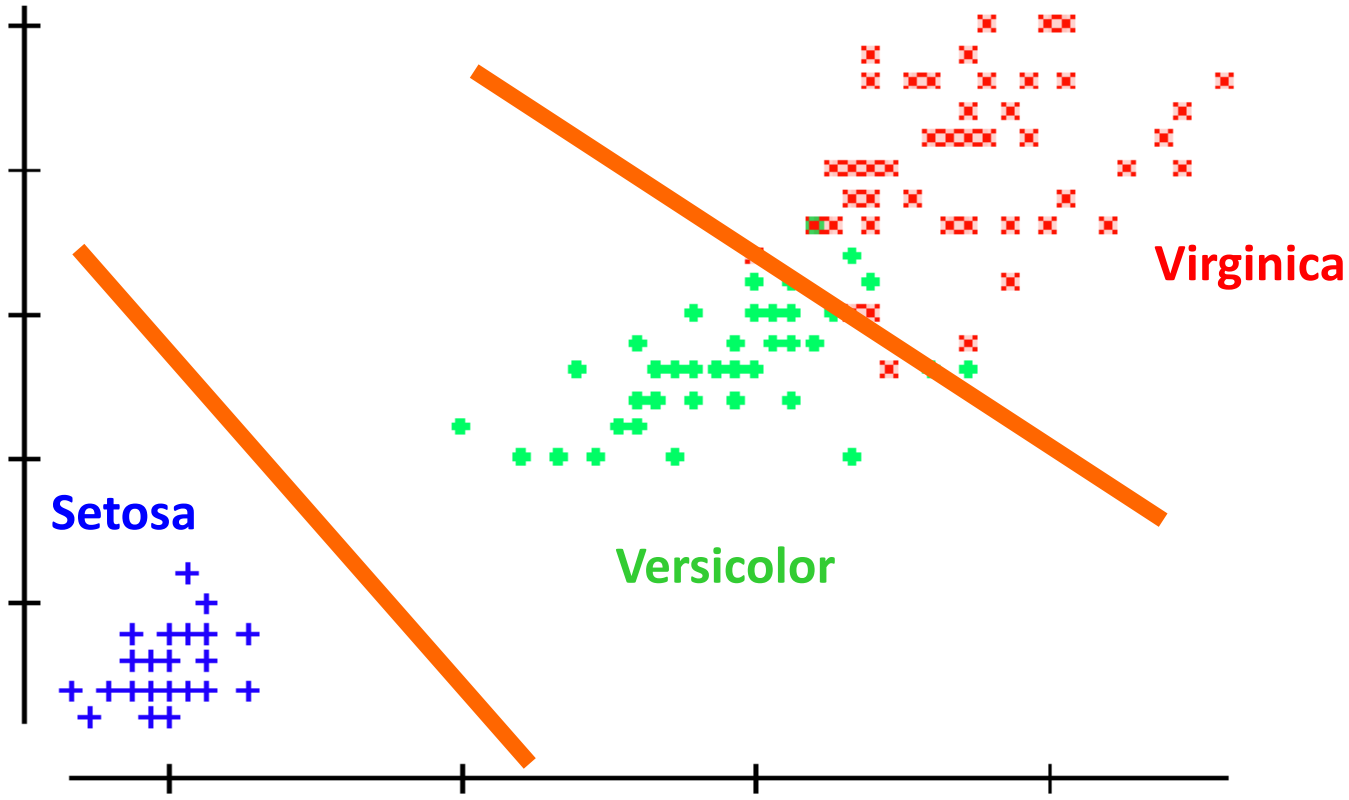


Iris Setosa

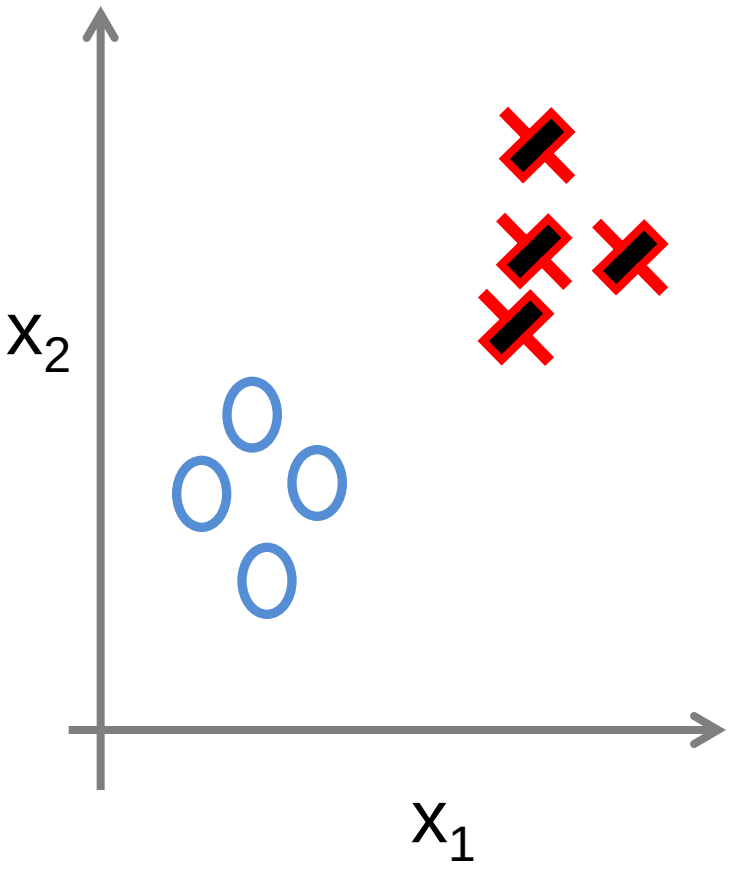
Iris Versicolor

Iris Virginica

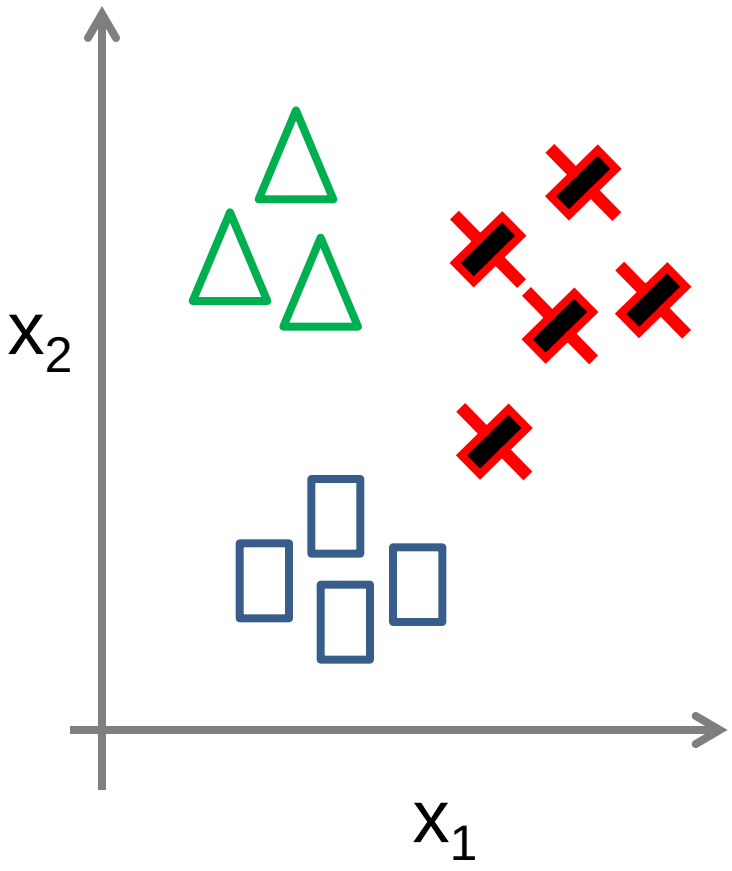
We can generalize the piecewise linear classifier to N classes, by fitting N-1 lines. In this case we first learned the line to (perfectly) discriminate between **Setosa** and **Virginica**/**Versicolor**, then we learned to approximately discriminate between **Virginica** and **Versicolor**.



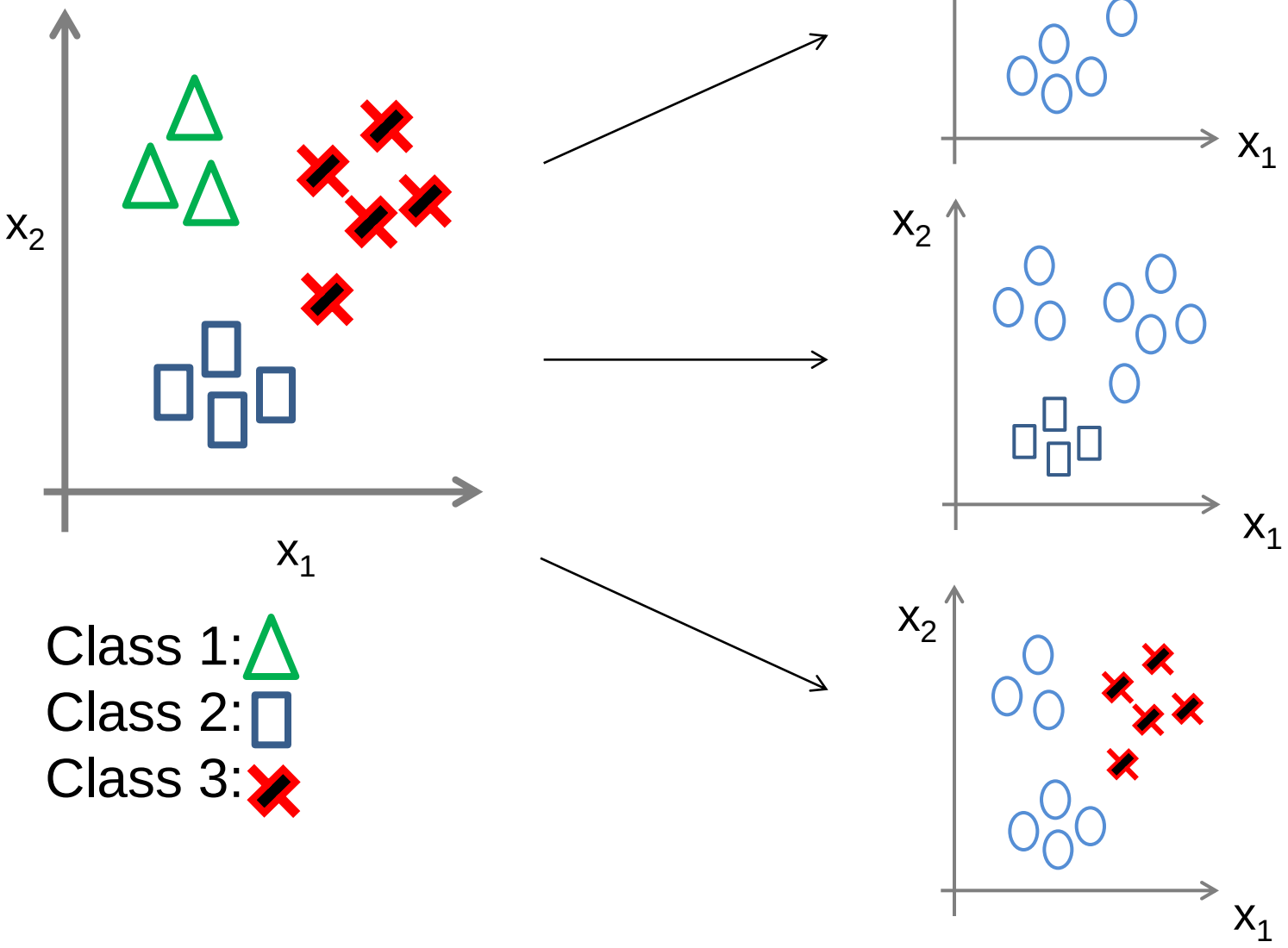
Binary classification:



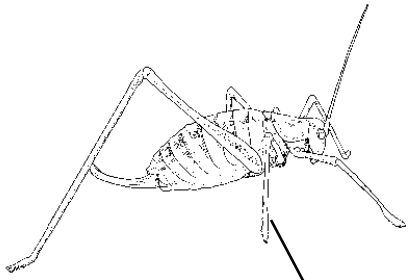
Multi-class classification:

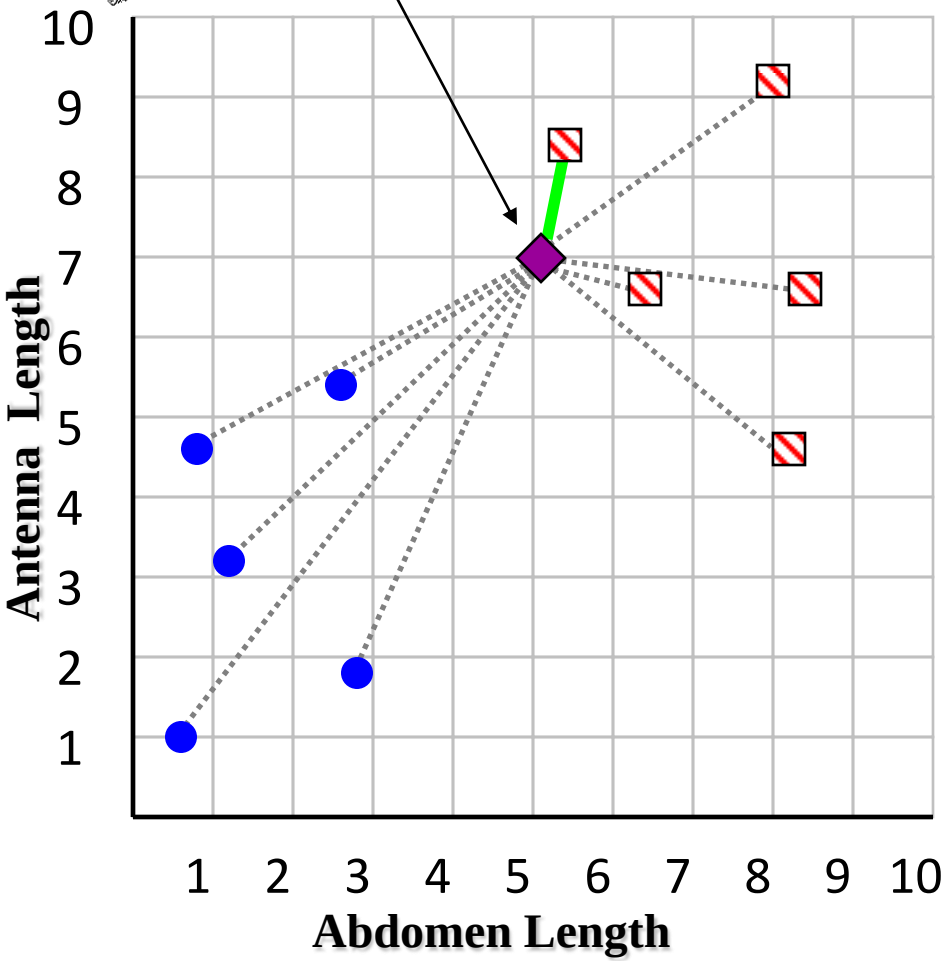


One-vs-all (one-vs-rest):



Nearest Neighbor Classifier





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

class is **Katydid**

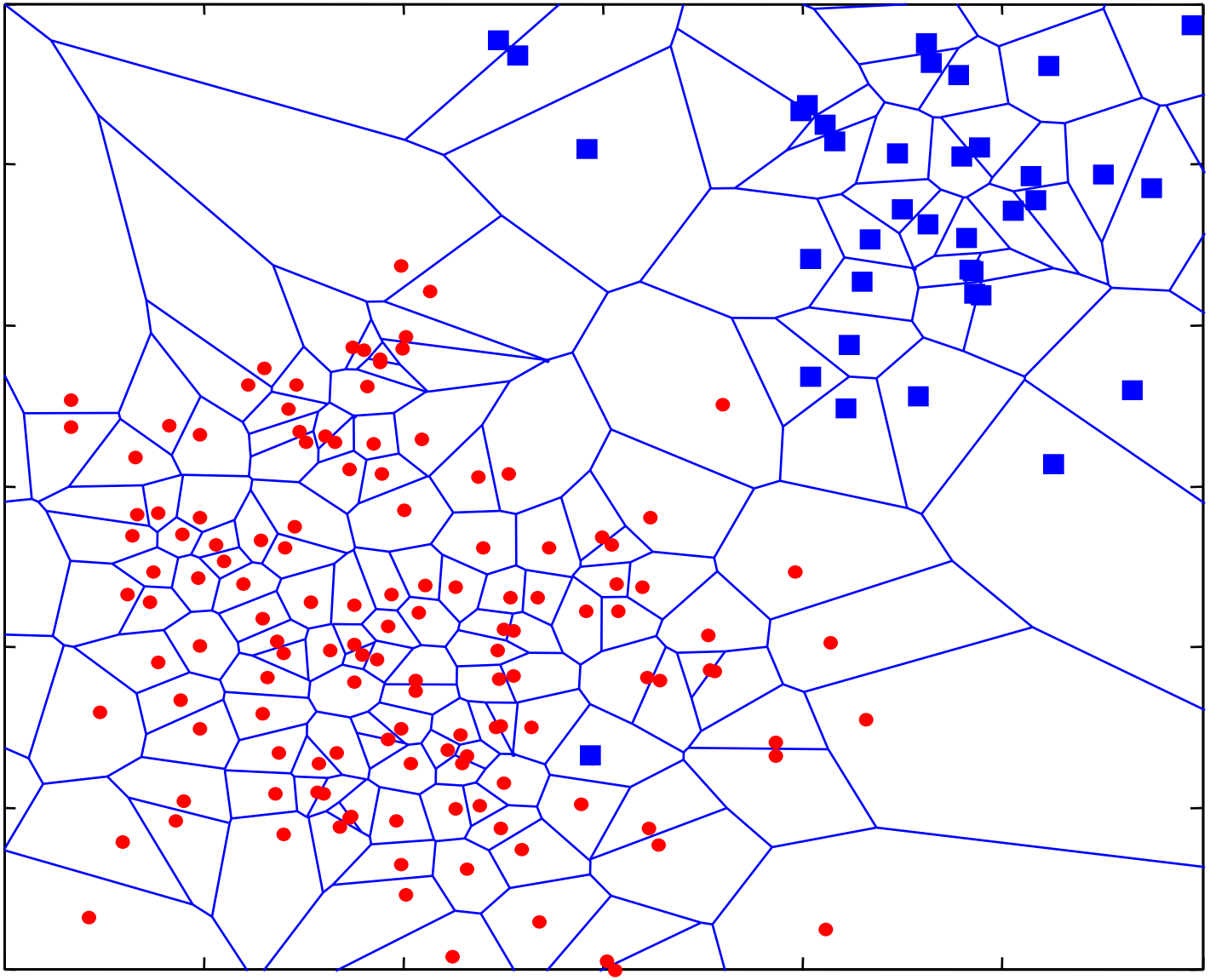
else

class is **Grasshopper**

Katydids

Grasshoppers

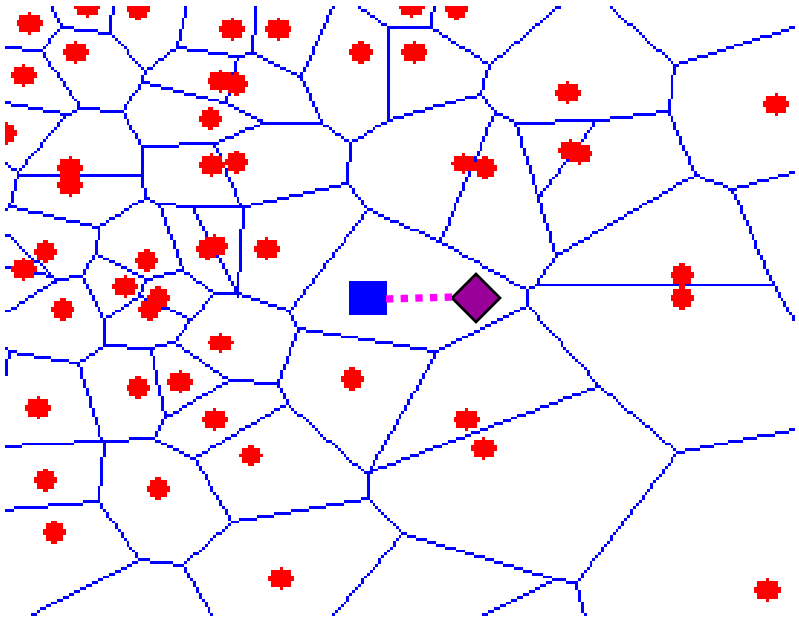
The nearest neighbor algorithm is sensitive to outliers…



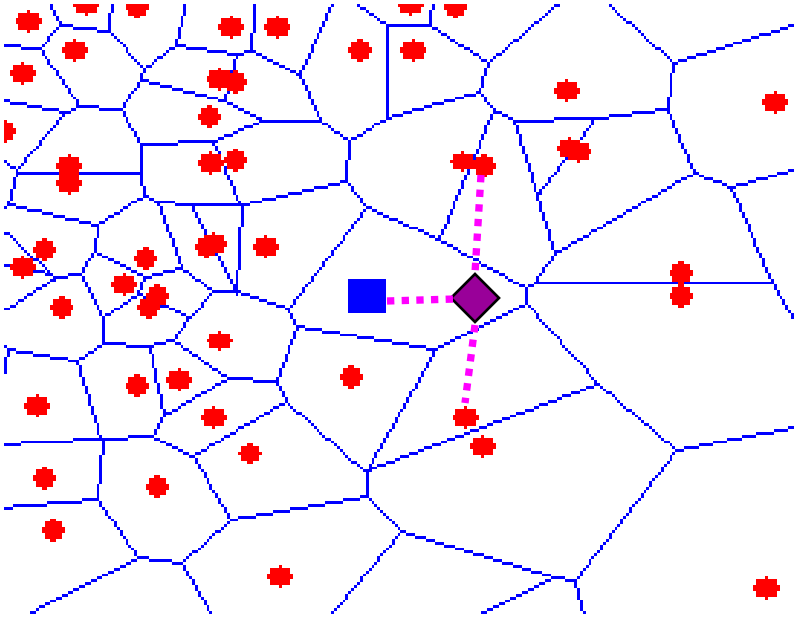
The solution is to…

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.

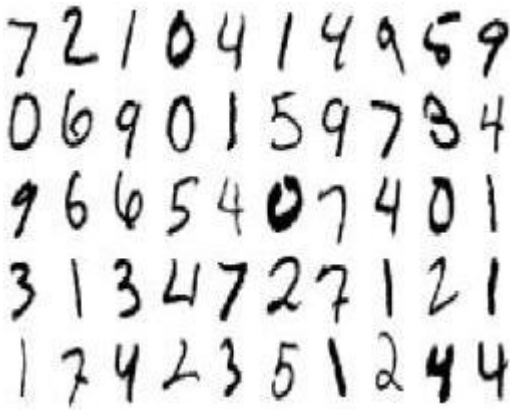


K = 1



K = 3

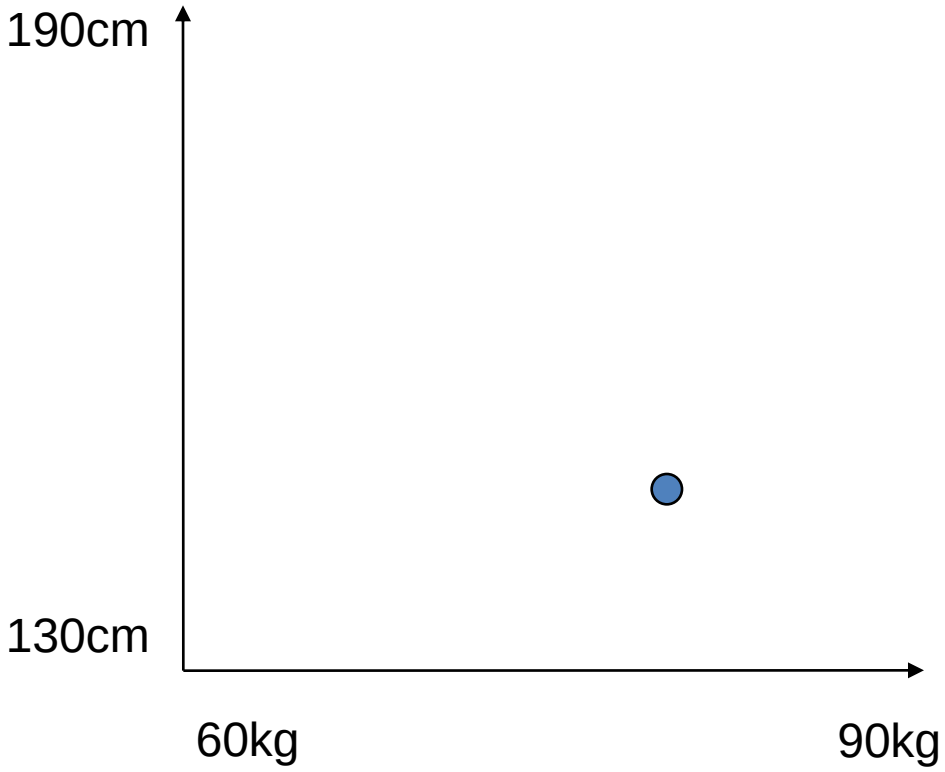
- Why recognising rugby players is (almost) the same problem as *handwriting recognition*



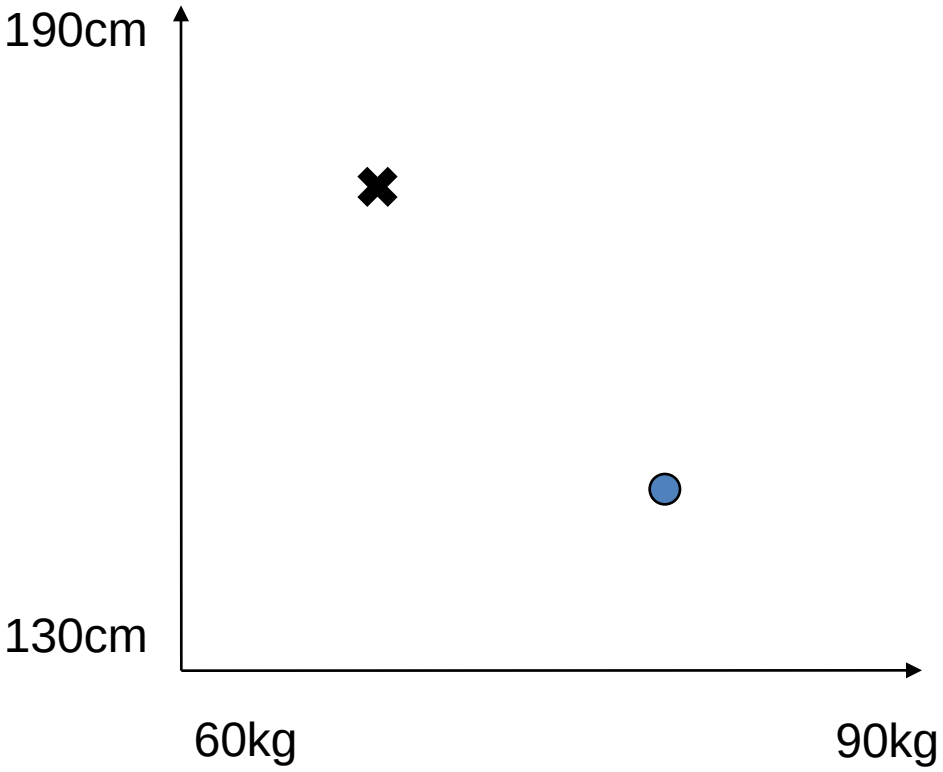
Can we LEARN to recognise a rugby player?

What are the “features” of a rugby player?

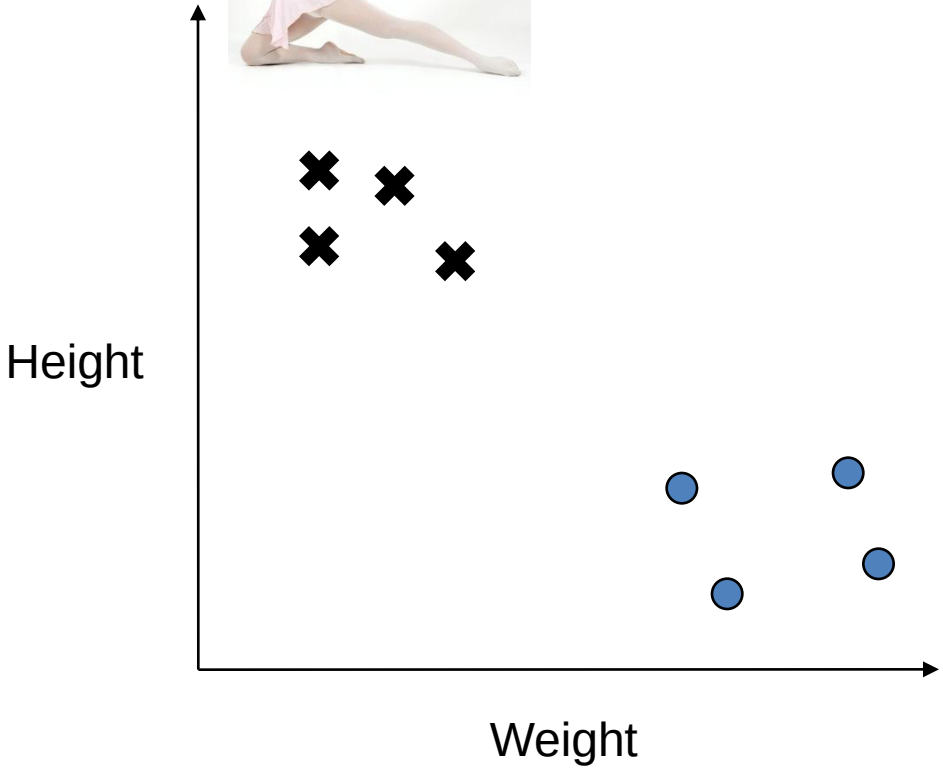
Rugby players = short + heavy?



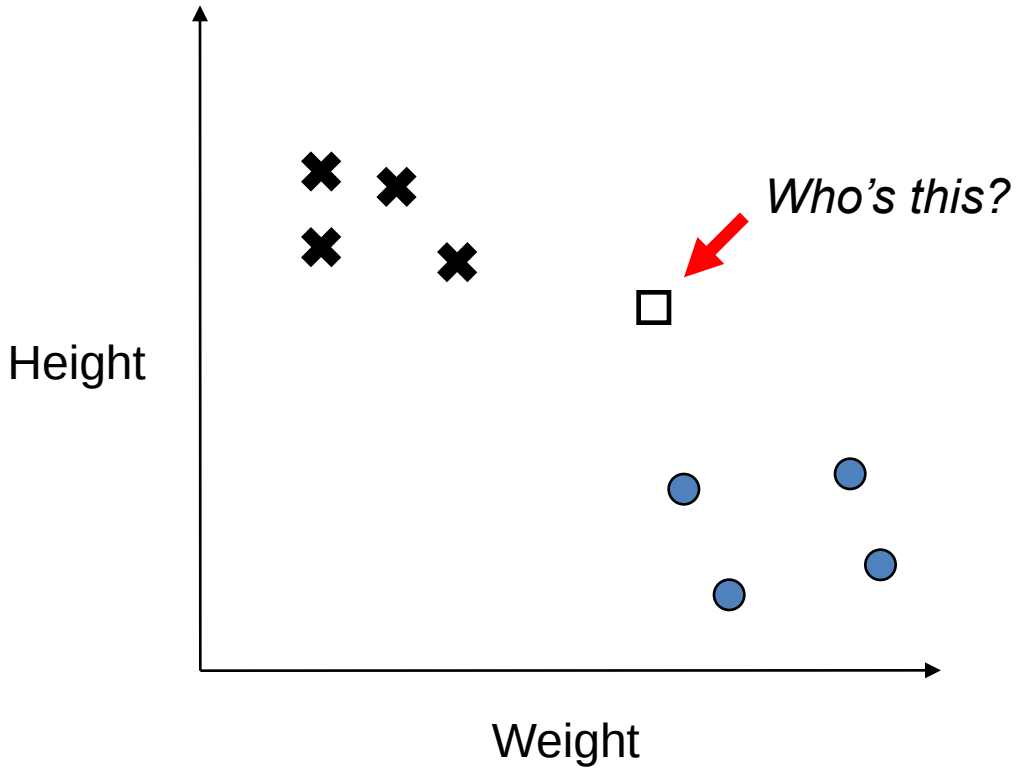
Ballet dancers = tall + skinny?



Rugby players “cluster” separately in the space.

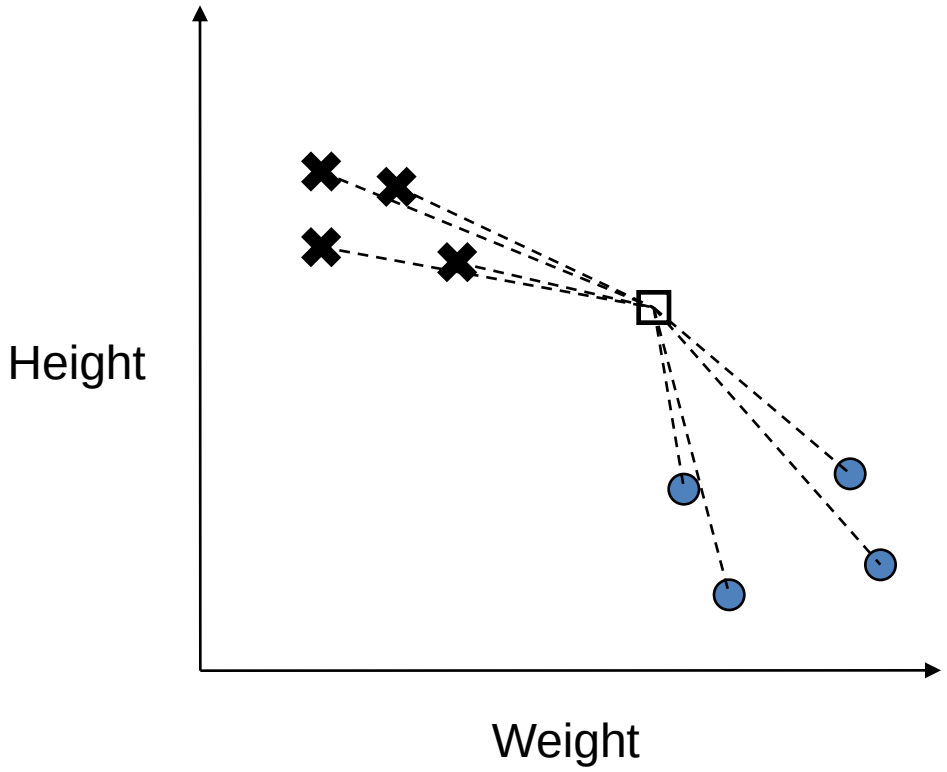


The K-Nearest Neighbour Algorithm



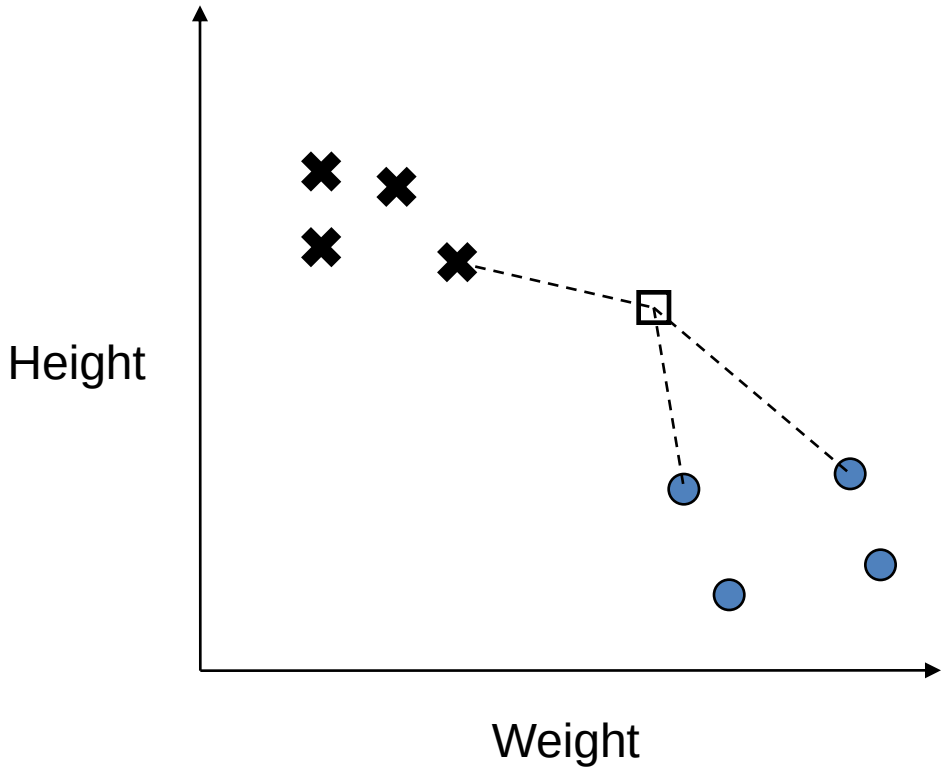
The K-Nearest Neighbour Algorithm

1. *Measure distance to all points*



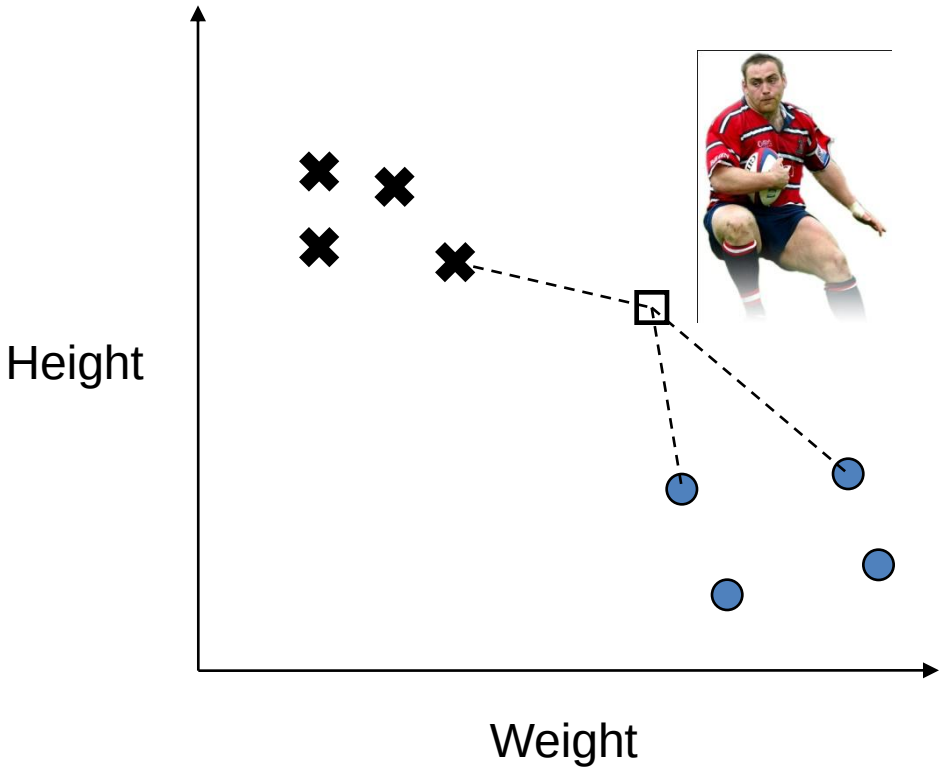
The K-Nearest Neighbour Algorithm

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



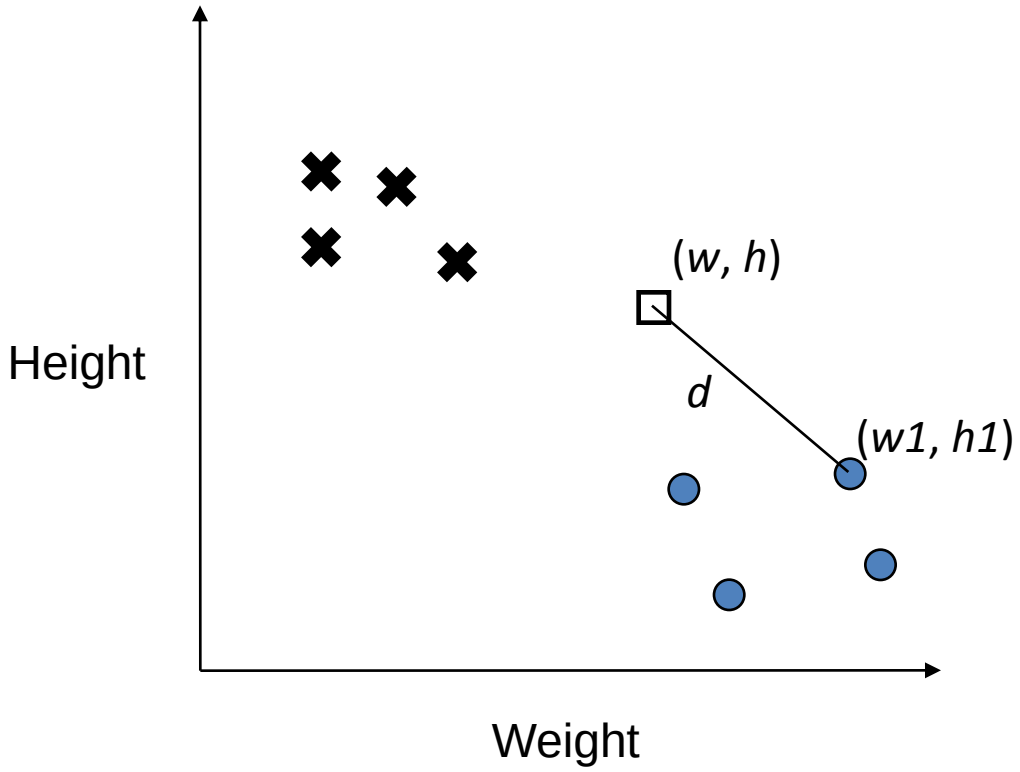
The K-Nearest Neighbour Algorithm

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….

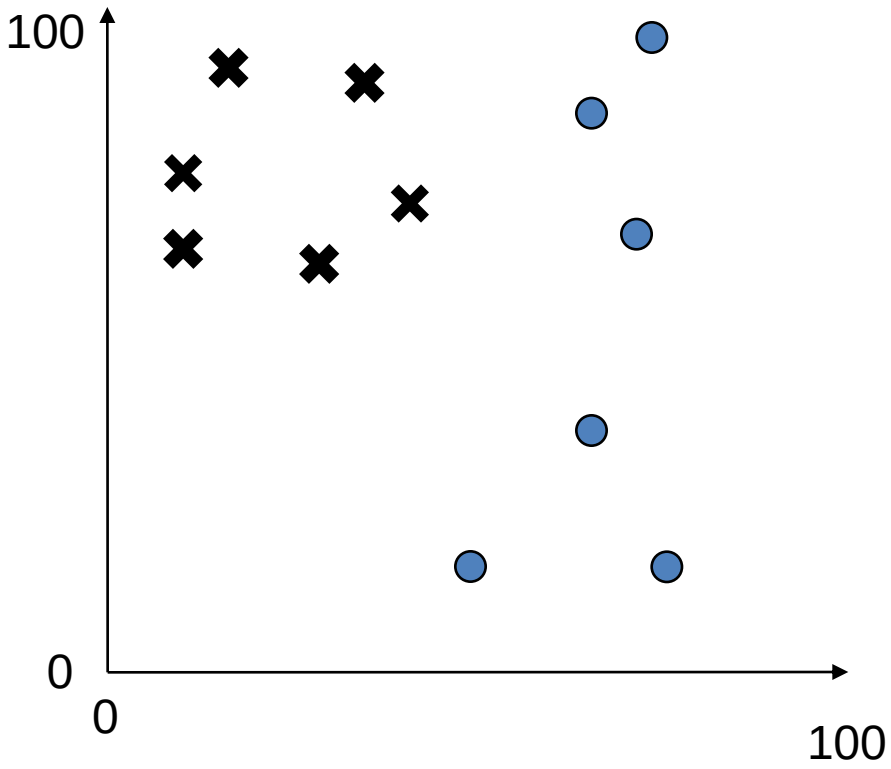
$$d = \sqrt{(x - x_1)^2 + (y - y_1)^2 + (z - z_1)^2}$$

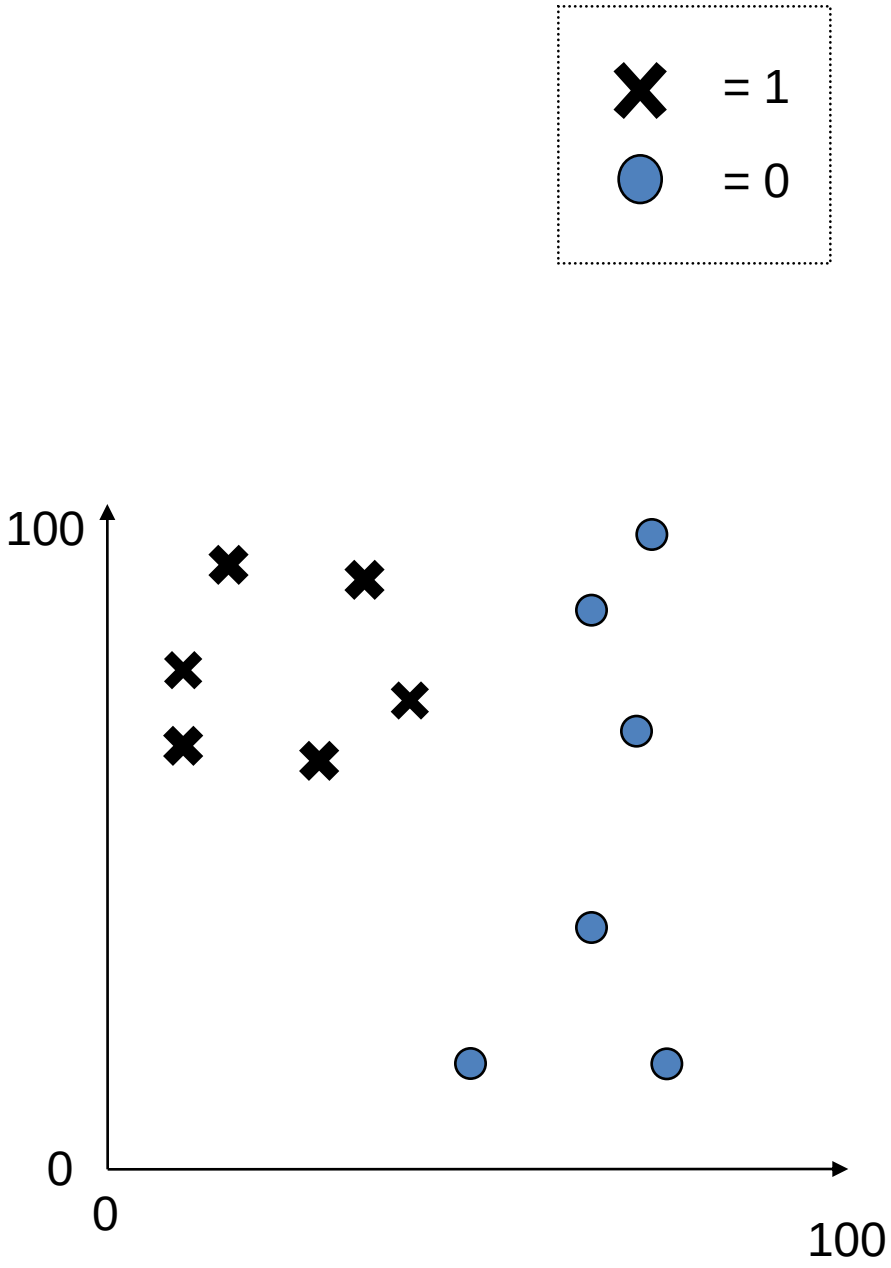
x = Height
y = Weight
z = Shoe size

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

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

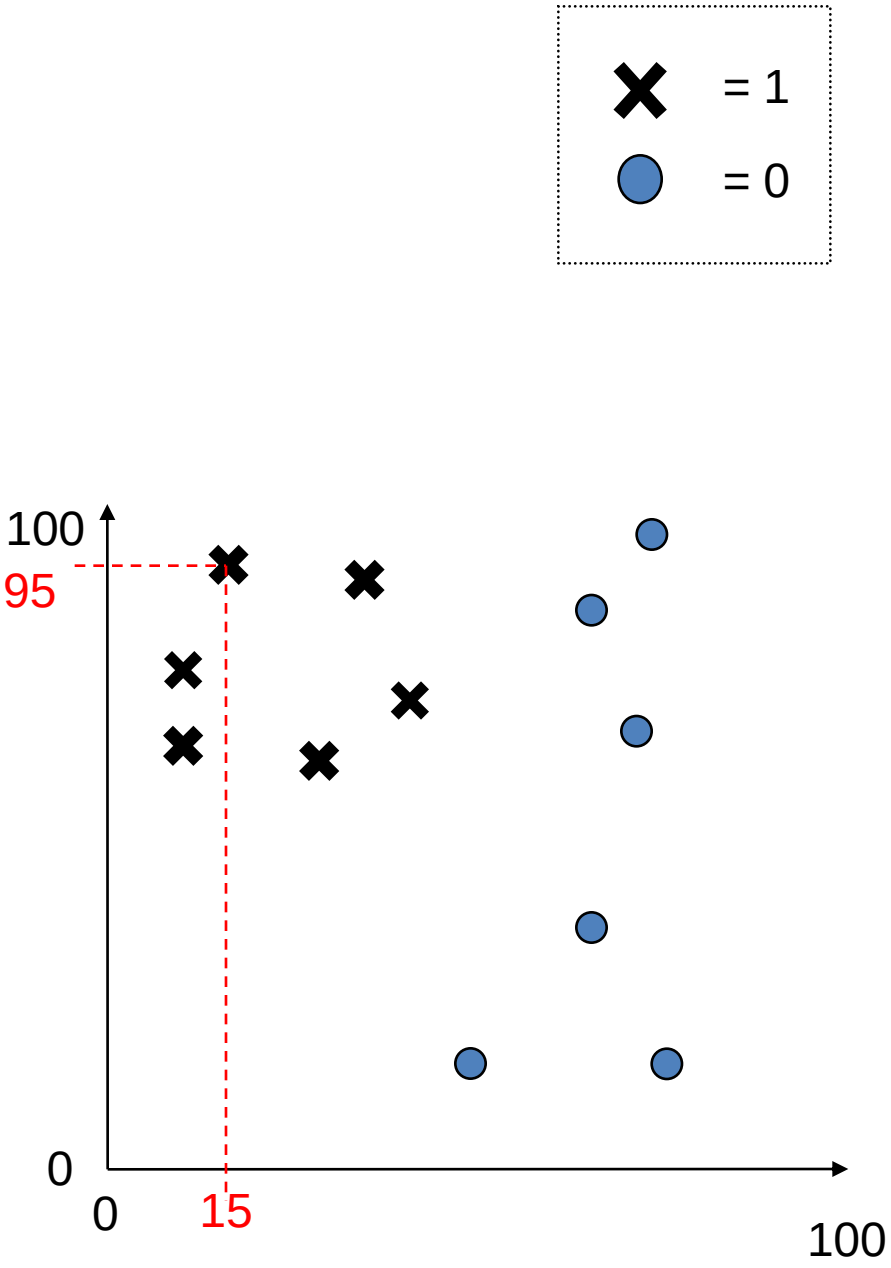
?


100
0
0
100



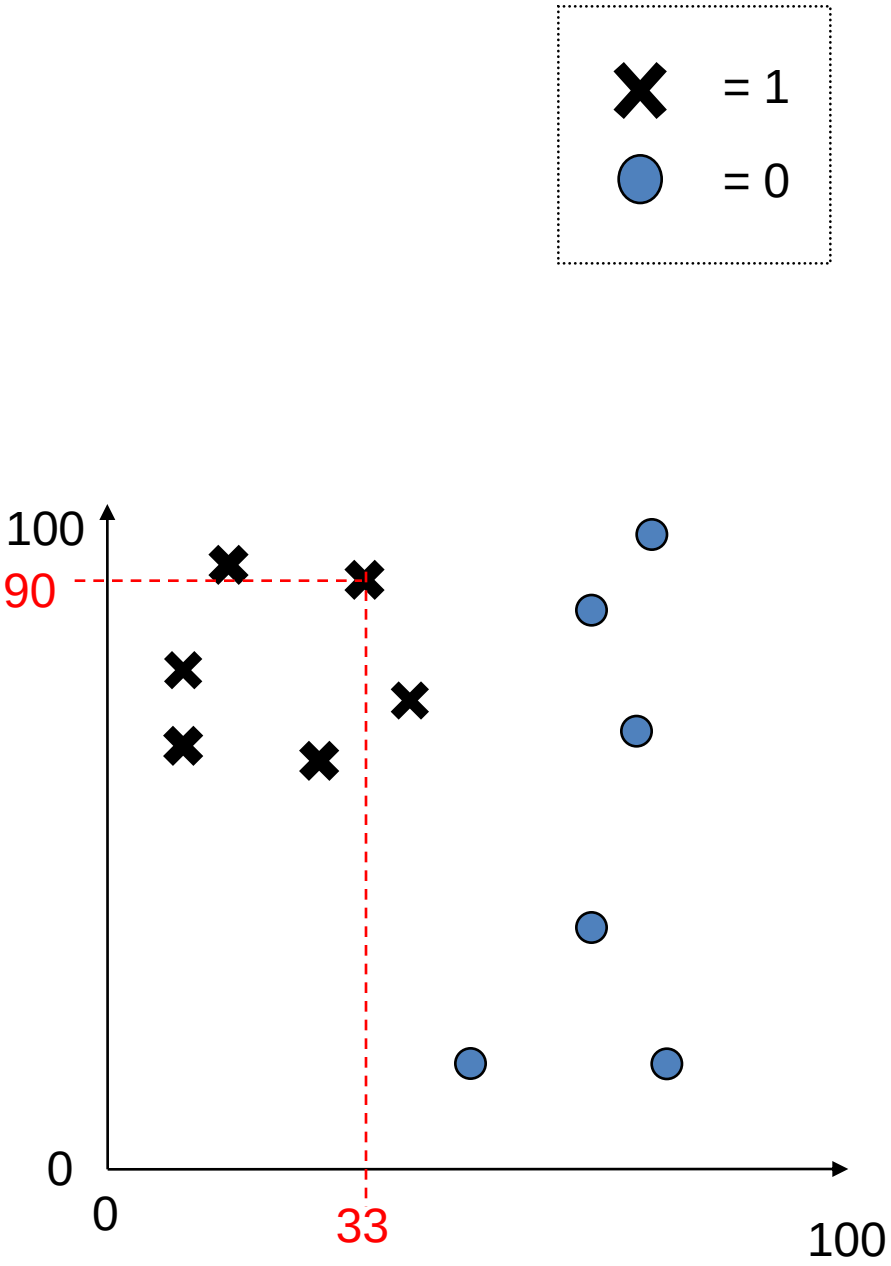
Inputs

Labels



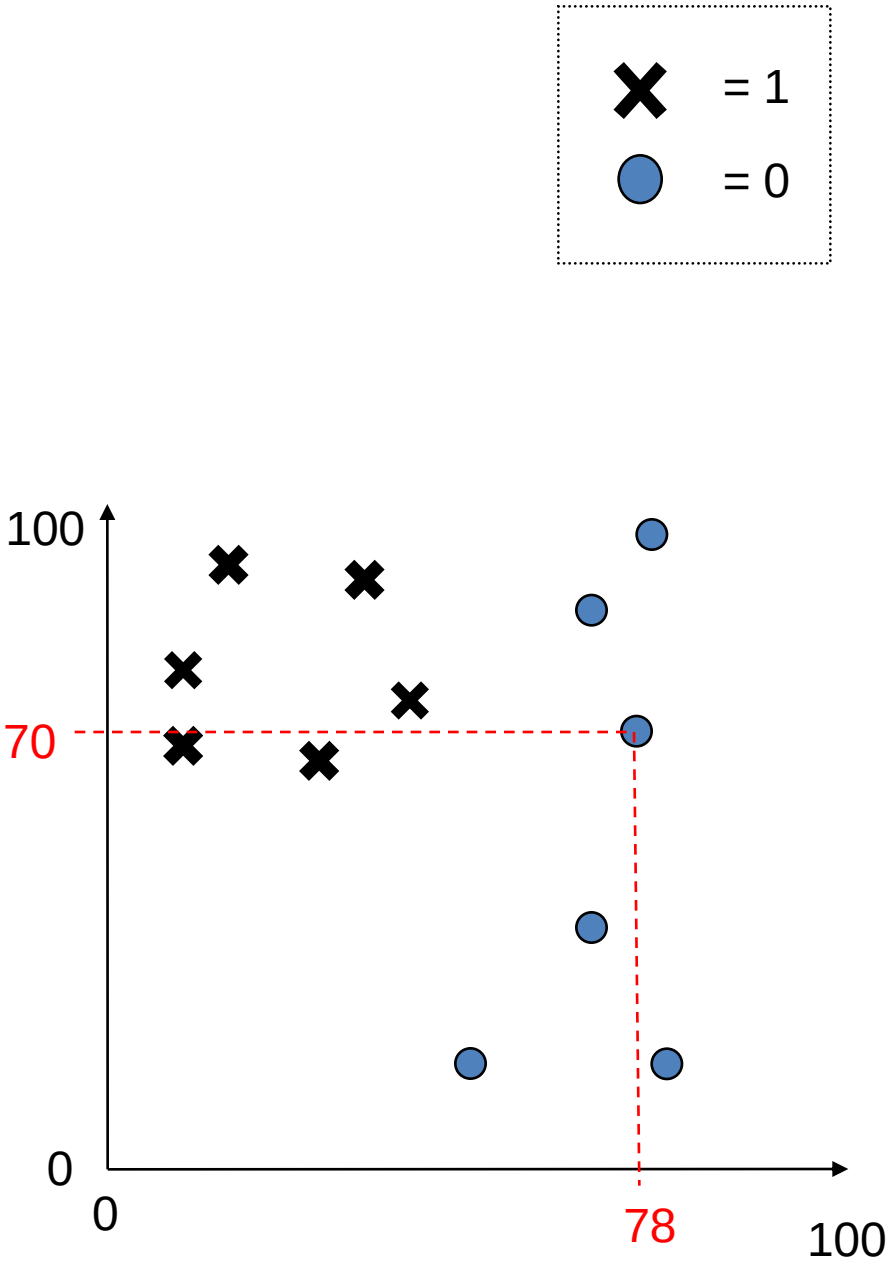
Inputs	
15	95

Labels
1



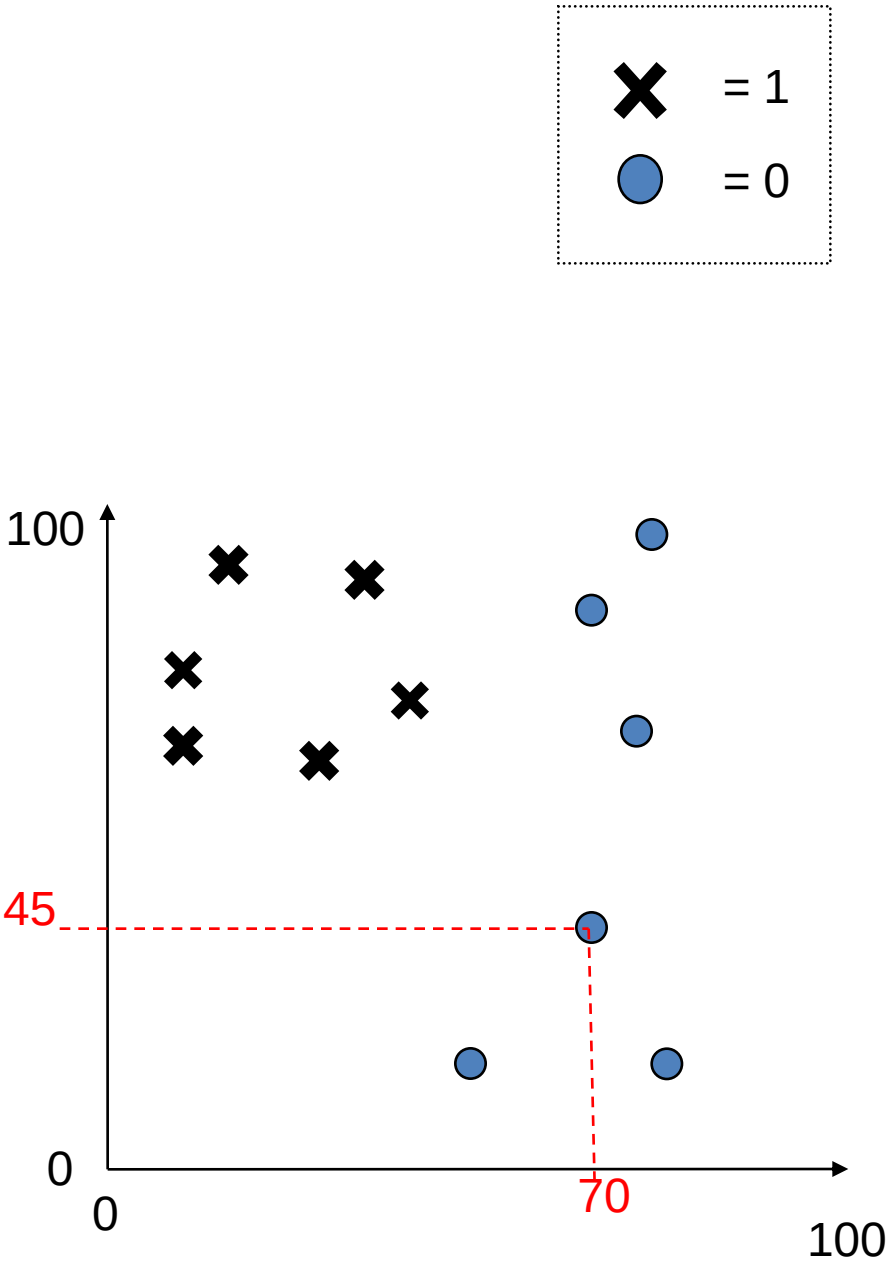
Inputs	
15	95
33	**90**

Labels
1
1



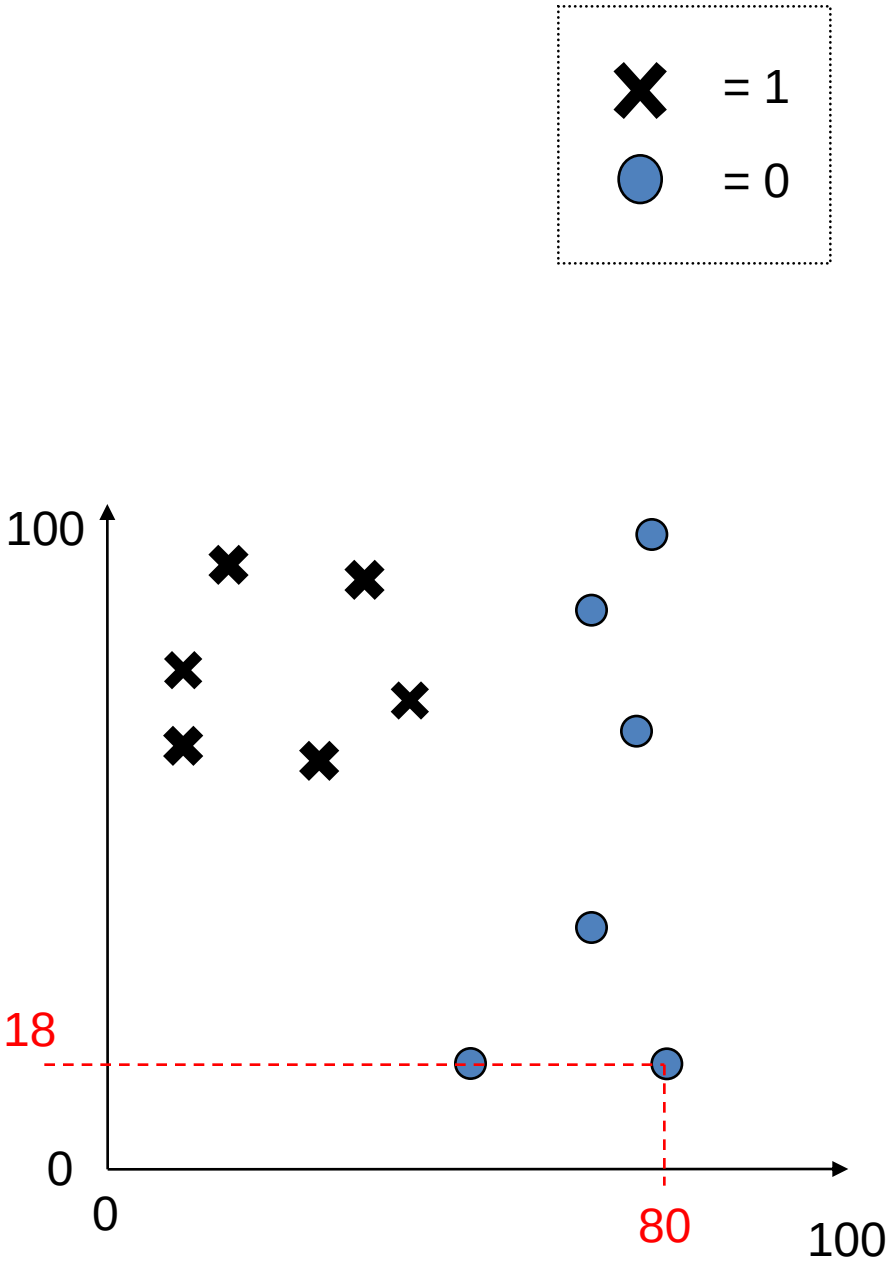
Inputs	
15	95
33	90
78	**70**

Labels
1
1
0



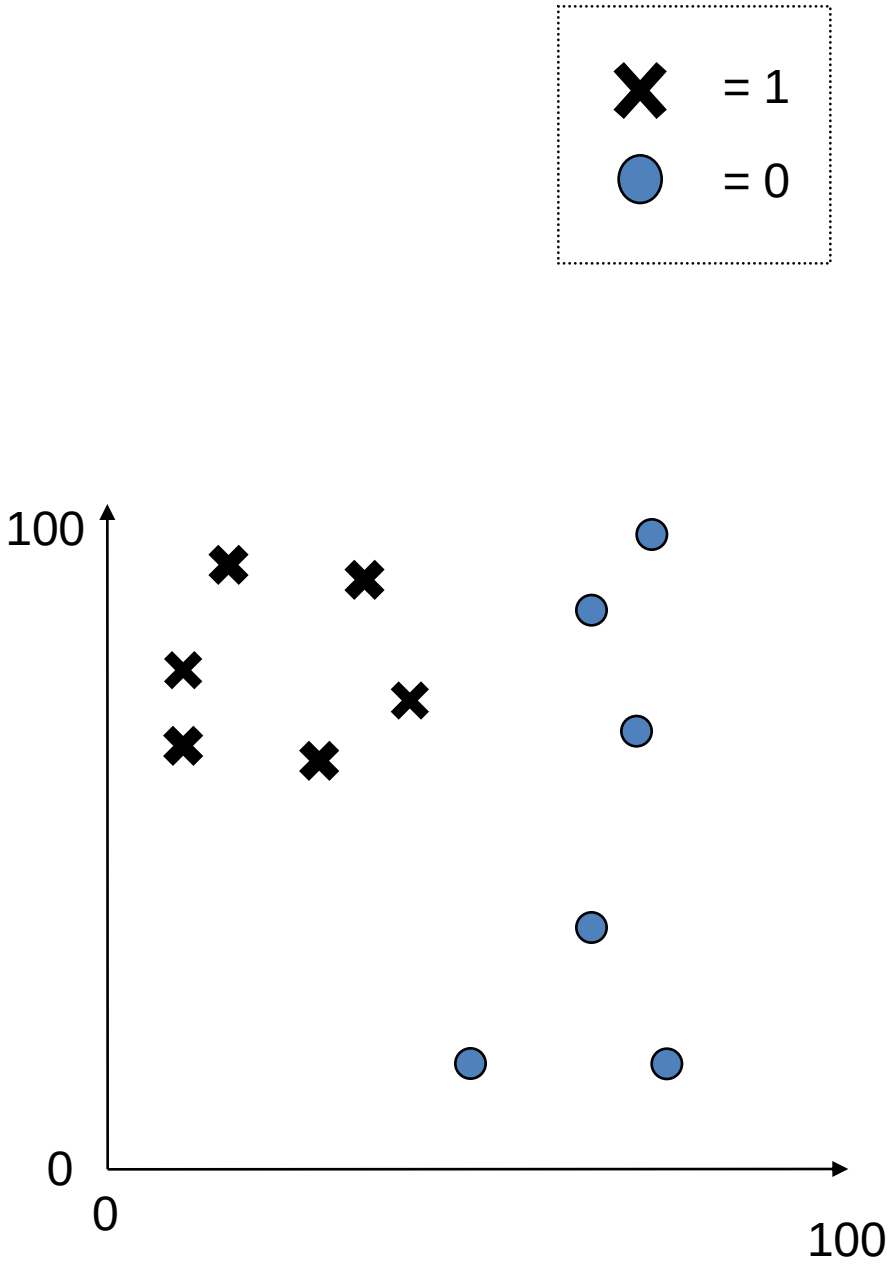
Inputs	
15	95
33	90
78	70
70	**45**

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”

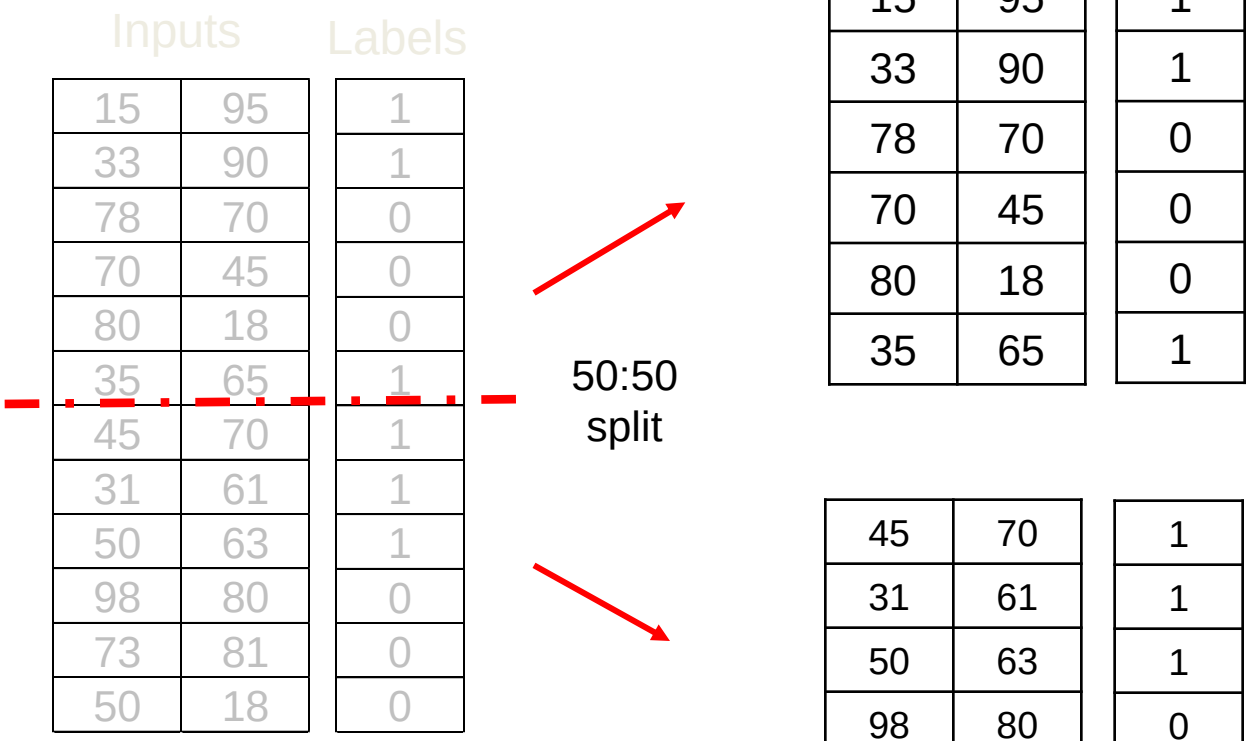
SPLITTING OF TRAINING AND TEST DATA

Dividing Up Data

- We need independent data sets to train, set parameters, and test performance
- Thus we will often divide a data set into three
  - Training set
  - Parameter selection set
  - Test set
- These **must** be independent
- Data set 2 is not always necessary

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

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

Training set

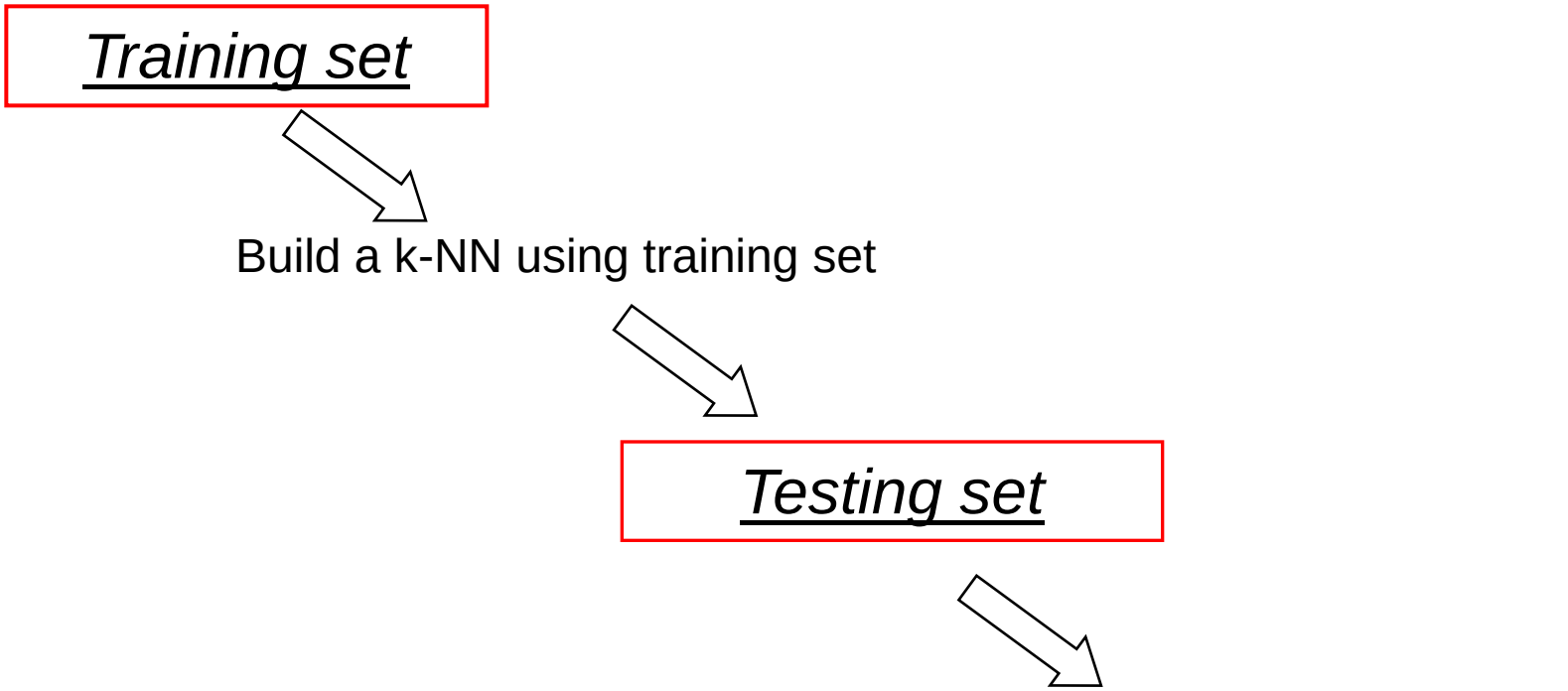
Train a K-NN on this…

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

Price of Cross Validation

- Cross-validation is computationally expensive (K-fold cross-validation requires K times as much work)
- There are attempts at estimating generalisation error more cheaply (boot-strapping) methods, but these are not very accurate
- Cross-validation is only necessary when you have little data
- Re-running code is usually cheap compared with writing the code

PERFORMANCE MEASUREMENTS

R.O.C. Analysis

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

Accuracy Measures

- Accuracy
  - = (TP+TN)/(P+N)
- Sensitivity or true positive rate (TPR)
  - = TP/(TP+FN) = TP/P
- Specificity or TNR
  - = TN/(FP+TN) = TN/N
- Positive Predictive value (Precision) (PPV)
  - = Tp/(Tp+Fp)
- Recall
  - = Tp/(Tp+Fn)

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

Minimum Distance
&
Neural Network

Minimum Distance Classifier

- For a test sample X, compute $D_j(X)$ for each class j
- Assign class with minimum D(x) value

$$D_j(\mathbf{x}) = \left\| \mathbf{x} - \mathbf{m}_j \right\|$$

$$= \left[(\mathbf{x} - \mathbf{m}_j)^T (\mathbf{x} - \mathbf{m}_j) \right]^{1/2}$$

- Here m_j is mean value of training samples from j^{th} class

Minimum Distance Classifier

Manipulating $D_j(X)$

$$
\begin{aligned}
D_j^2(\mathbf{x}) &= \left\|\mathbf{x}-\mathbf{m}_j\right\|^2 = (\mathbf{x}-\mathbf{m}_j)^T(\mathbf{x}-\mathbf{m}_j) \\
&= \mathbf{x}^T\mathbf{x}-2\mathbf{x}^T\mathbf{m}_j+\mathbf{m}_j^T\mathbf{m}_j \\
&= \mathbf{x}^T\mathbf{x}-\mathbf{2}\left(\mathbf{x}^T\mathbf{m}_j-\frac{1}{2}\mathbf{m}_j^T\mathbf{m}_j\right).
\end{aligned}
$$

Minimum Distance Classifier

- Now instead of $D_j(X)$, we compute discriminent function $d_j(X)$ for each class
- Assign class with maximum $d_j(X)$ value

$$d_j(\mathbf{x}) = \mathbf{x}^T\mathbf{m}_j - \frac{1}{2}\mathbf{m}_j^T\mathbf{m}_j \qquad j = 1, 2, \ldots, W$$

- Equation for decision boundary between two classes *i* and *j*

$$d_{ij}(\mathbf{x}) = \mathbf{x}^T(\mathbf{m}_i - \mathbf{m}_j) - \frac{1}{2}(\mathbf{m}_i^T\mathbf{m}_i - \mathbf{m}_j^T\mathbf{m}_j)$$

Minimum Distance Classifier

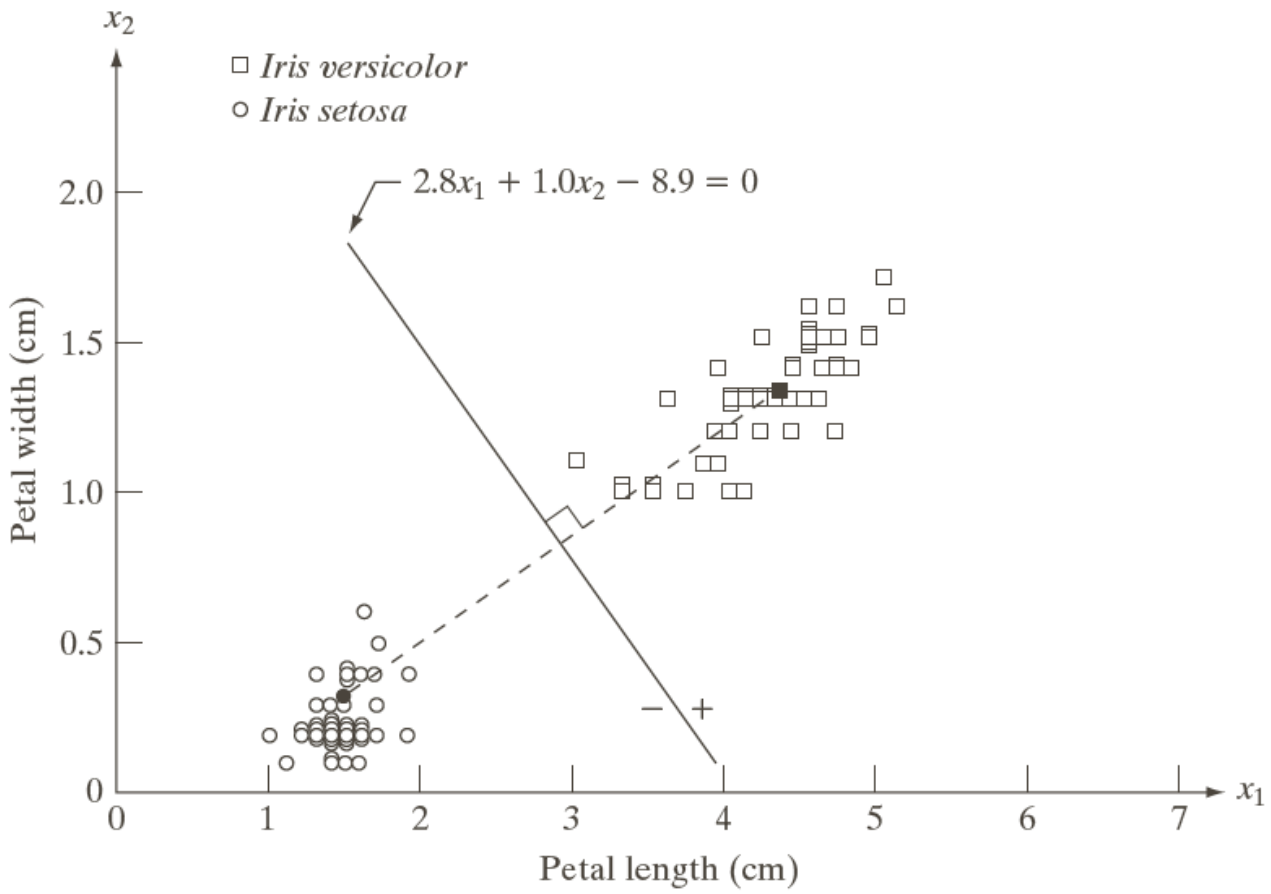


FIGURE 12.6 Decision boundary of minimum distance classifier for the classes of *Iris versicolor* and *Iris setosa*. The dark dot and square are the means.

Acknowledgements

- Digital Image Processing", Rafael C. Gonzalez & Richard E. Woods, Addison-Wesley, 2002
- Some slides are taken from Dr. Arslan's Lectures on Pattern recognition
- Machine Learning Lectures: Coursera

Material in these slides has been taken from, the following resources