Artificial Intelligence

HOW MACHINES LEARN

Machine Perception

- Build a machine that can recognize patterns:
  - Email or spam
  - Speech recognition
  - Fingerprint identification
  - OCR (Optical Character Recognition)
  - DNA sequence identification

A Relevant Example

Sorting incoming fish according to species using optical sensing

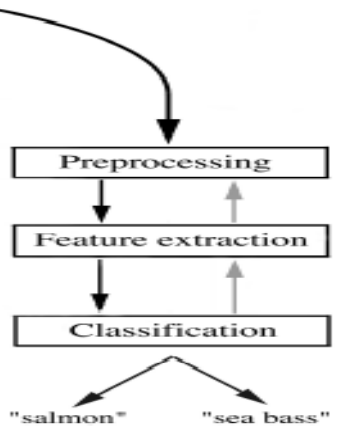
Salmon

Sea Bass

Problem Analysis

Set up a camera and take some sample images to extract **features:**

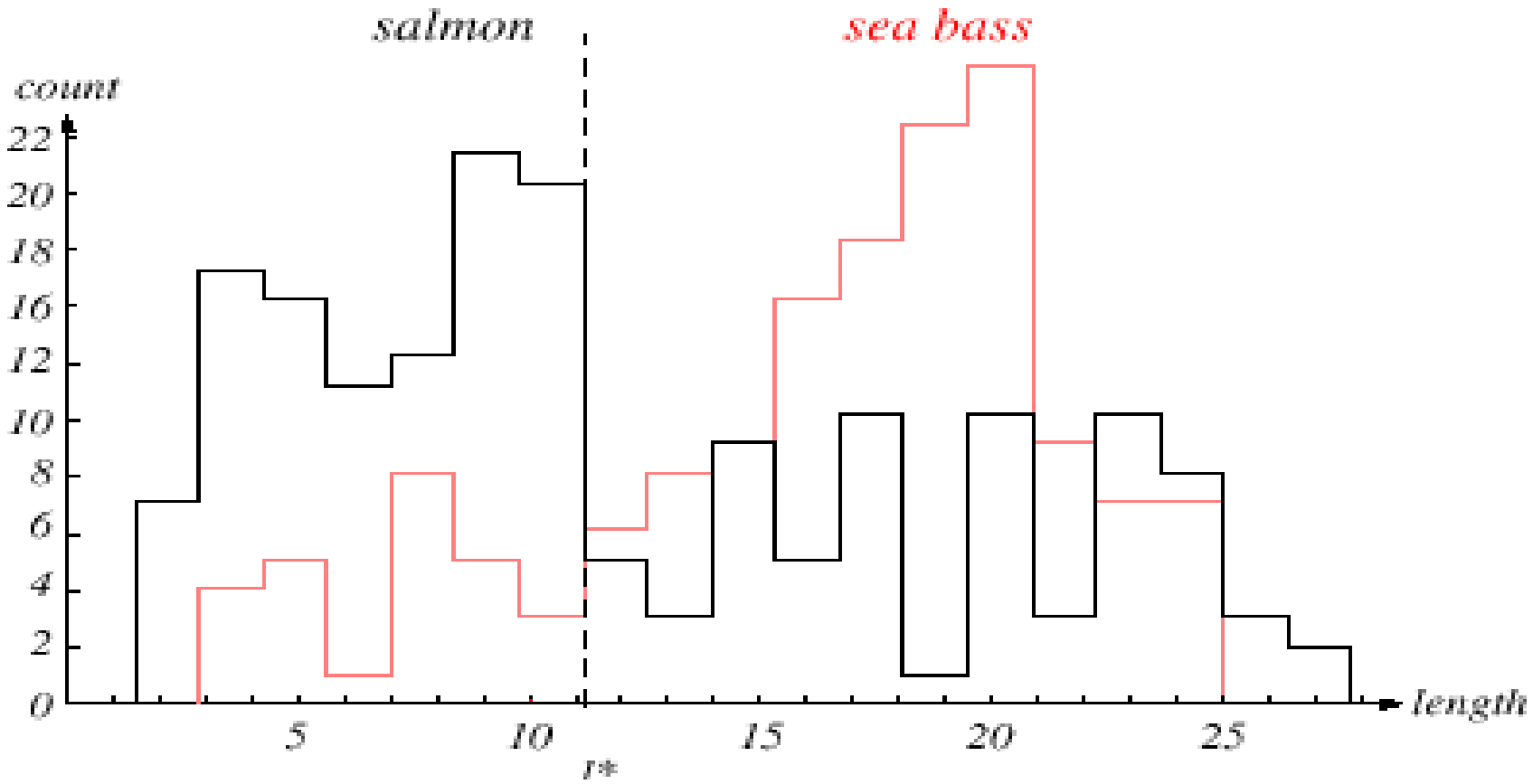
- Length
- Lightness
- Width
- Fins
- Position of the mouth
- etc…



This is the set of all suggested features to explore for use in our classifier!

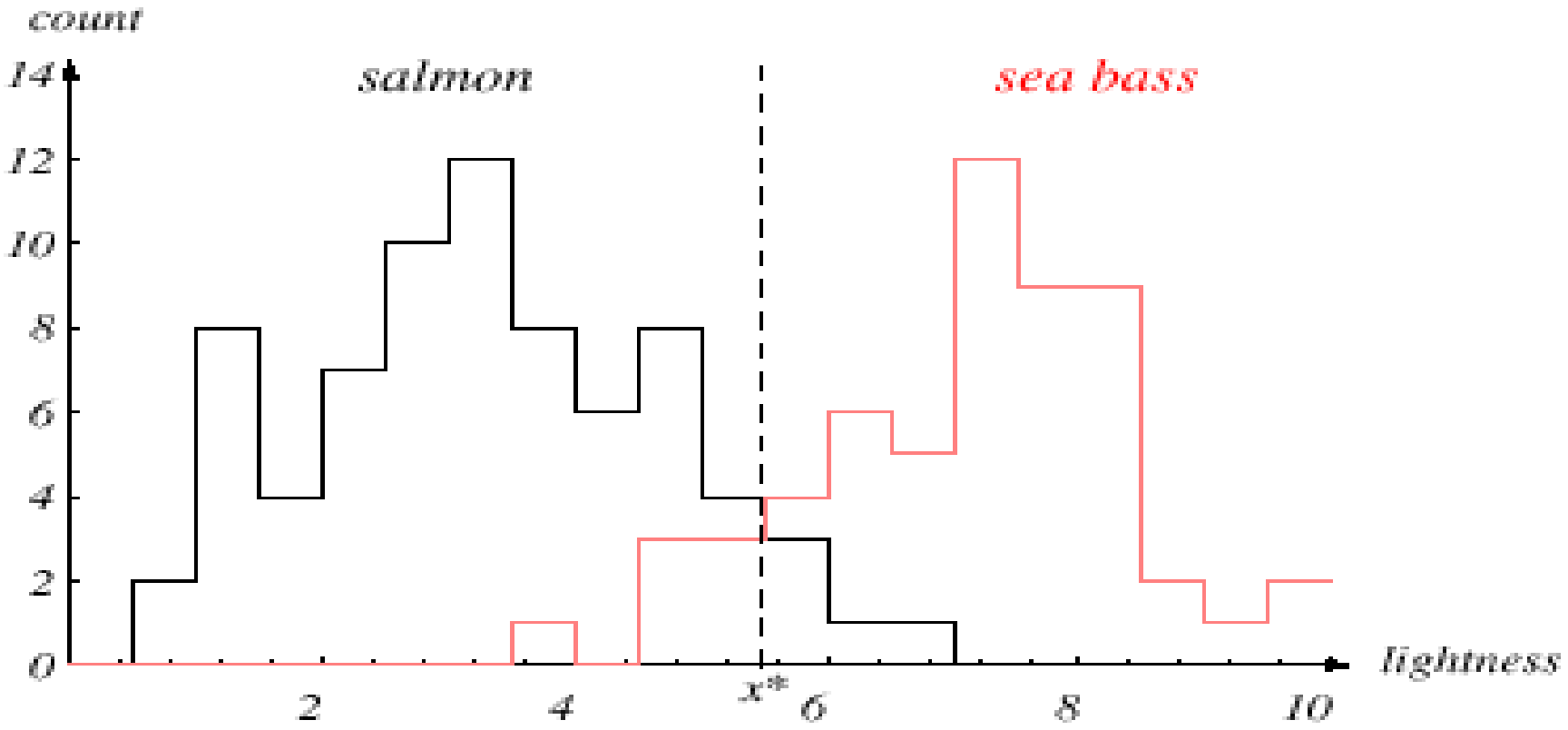
Classification

Select the length of the fish as a possible feature for discrimination



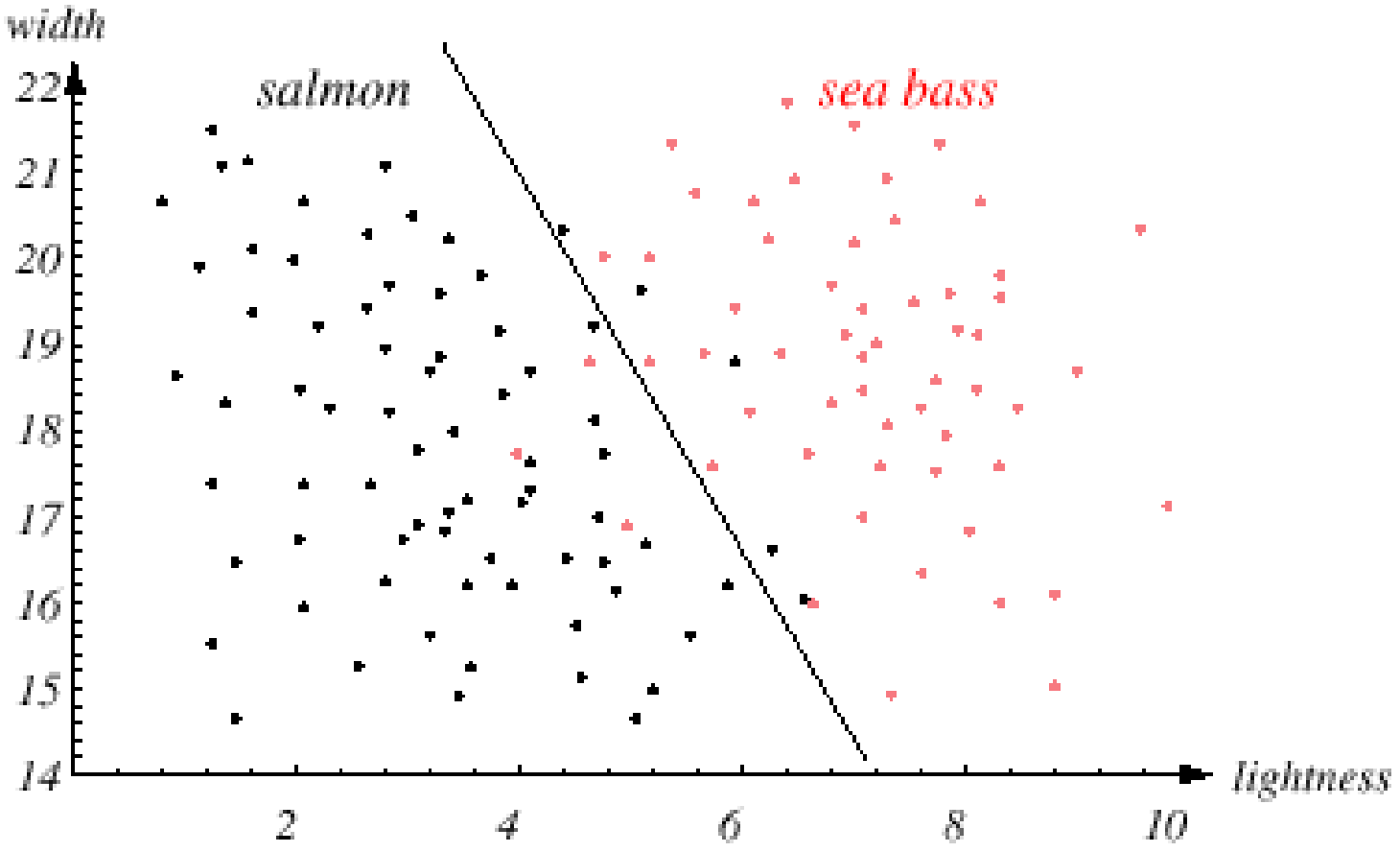
The length is a poor feature alone!

Select the lightness as a possible feature.

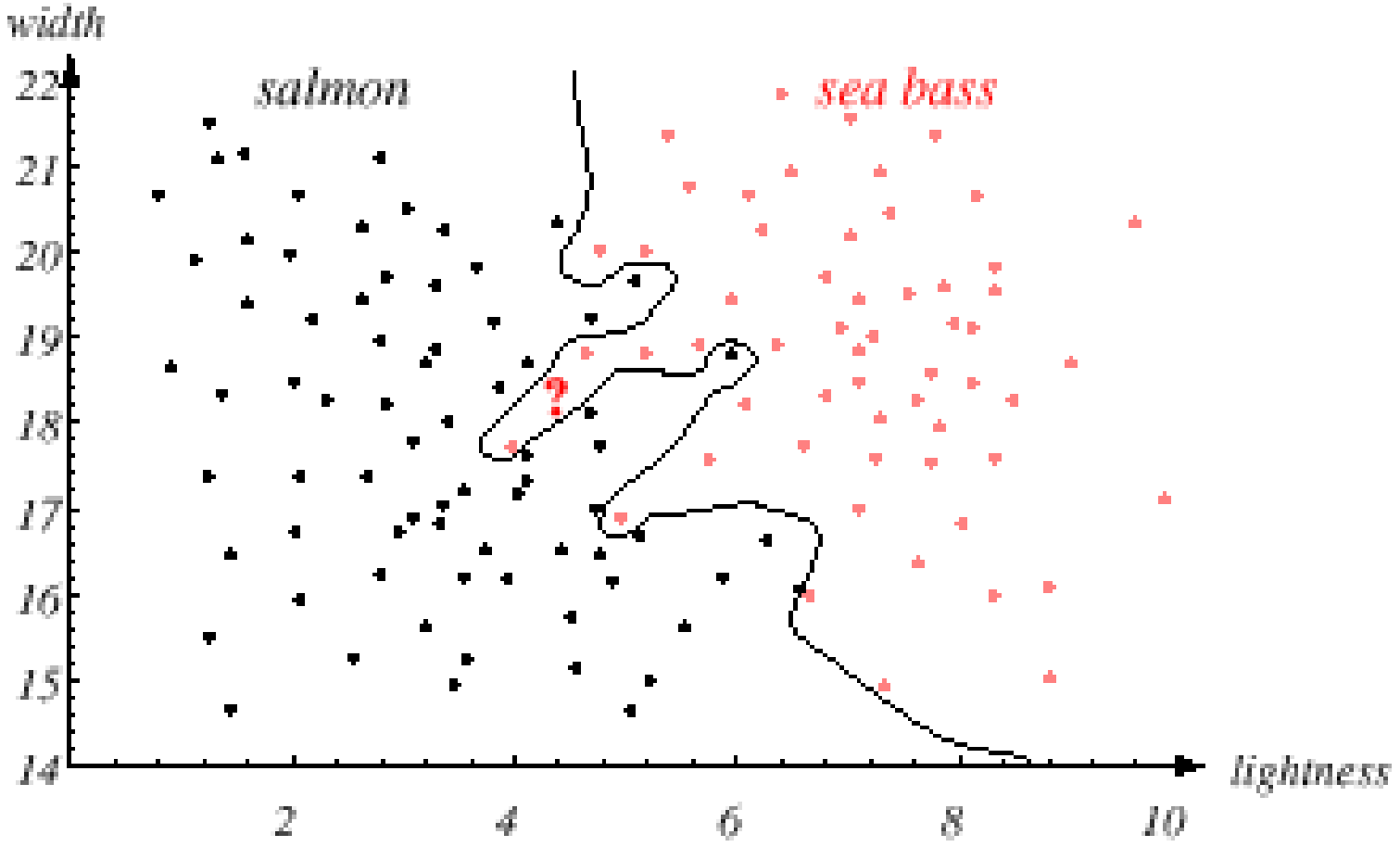


Two Features

Where to place the decision boundary? Should it be linear?

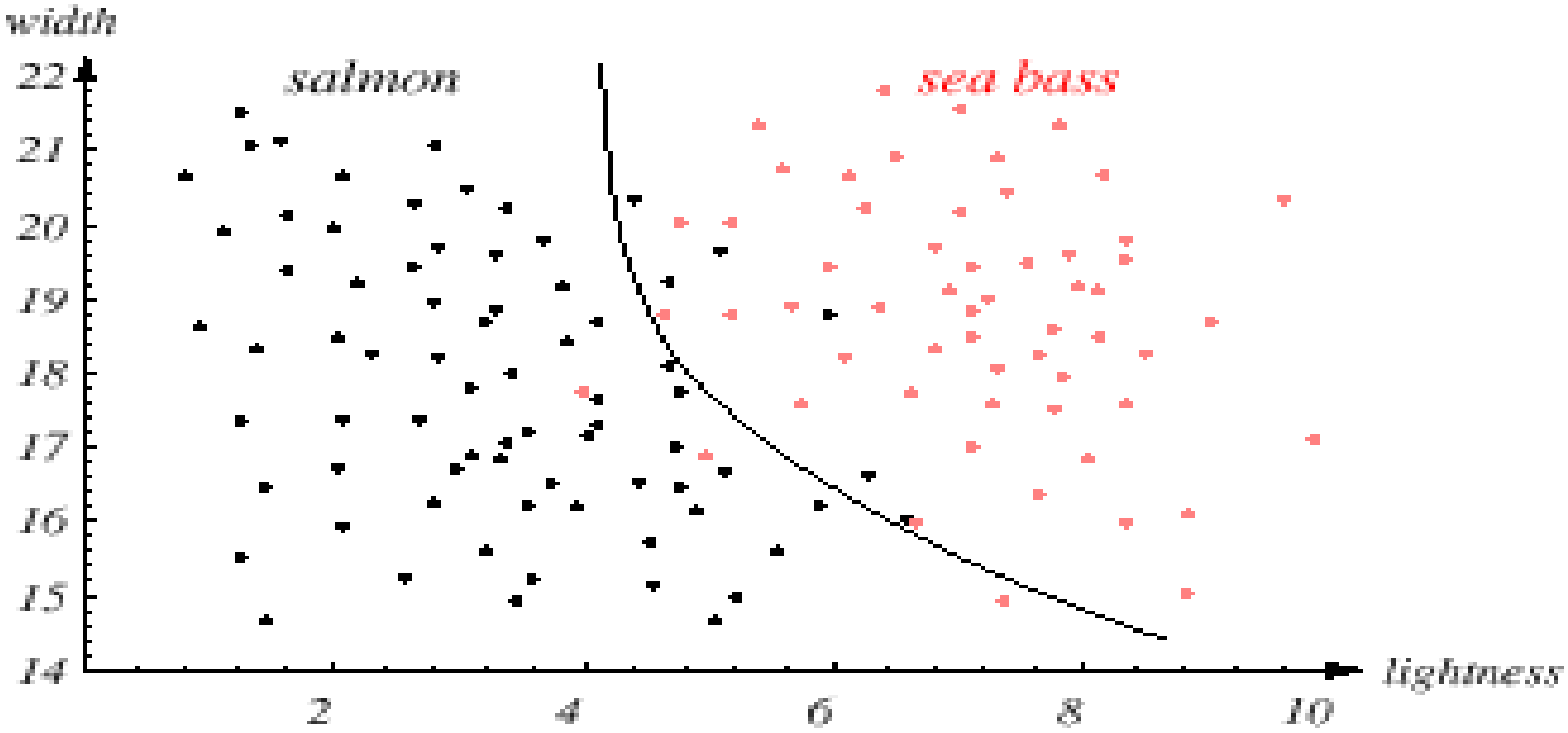


Optimal Decision Boundary?



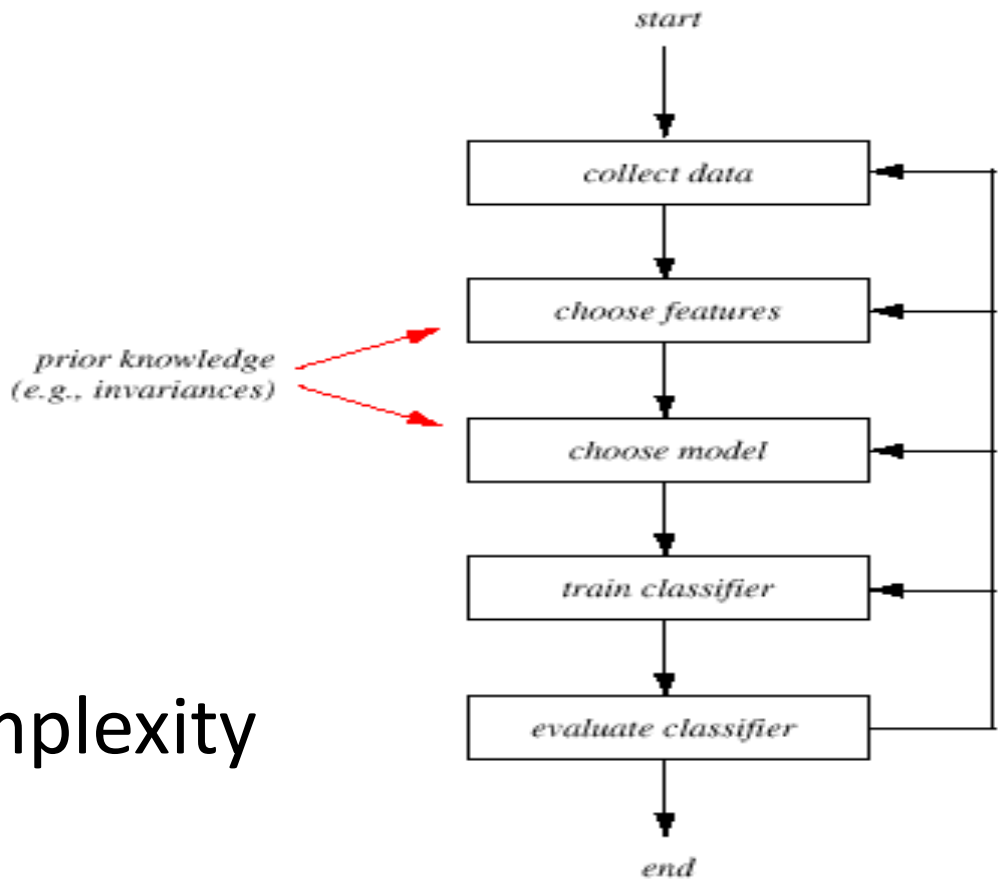
Generalization

Goldilocks' third choice



The Design Cycle

- Data collection
- Feature Choice
- Model Choice
- Training
- Evaluation
- Computational Complexity



- Data Collection

  - How do we know when we have collected an adequately large and representative set of examples for training and testing the system?

- Feature Choice

  - Depends on the characteristics of the problem domain. Simple to extract, invariant to irrelevant transformation insensitive to noise.

- Model Choice

  - Unsatisfied with the performance of our fish classifier and want to jump to another class of model

- Training

  - Use data to determine the classifier. Many different procedures for training classifiers and choosing models

- Evaluation

  - Measure the error rate (or performance and switch from one set of features to another one

- Computational Complexity

  - What is the trade-off between computational ease and performance?

  - (How an algorithm scales as a function of the number of features, patterns or categories?)

Learning and Adaptation

- Supervised learning
  - A teacher provides a category label or cost for each pattern in the training set
- Unsupervised learning
  - The system forms clusters or “natural groupings” of the input patterns

And now, Back to our textbook…

SPAM ASSASSIN

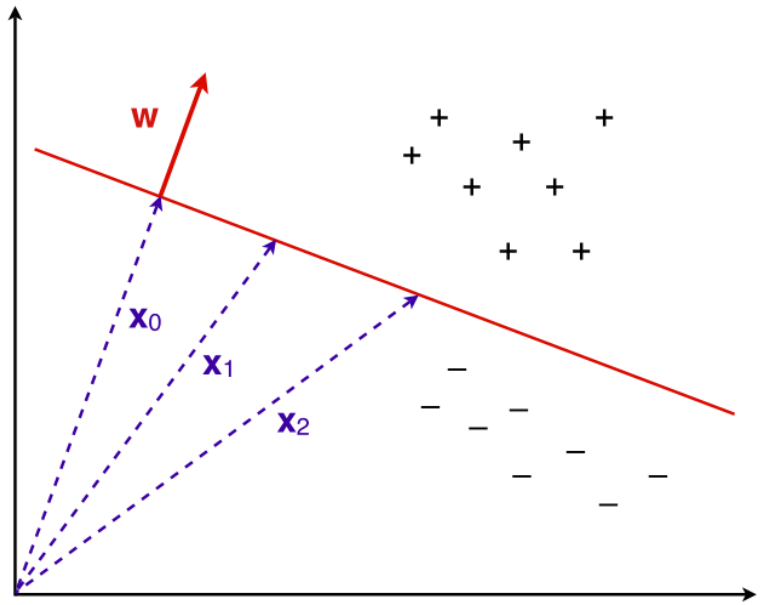
Assassinating spam e-mail

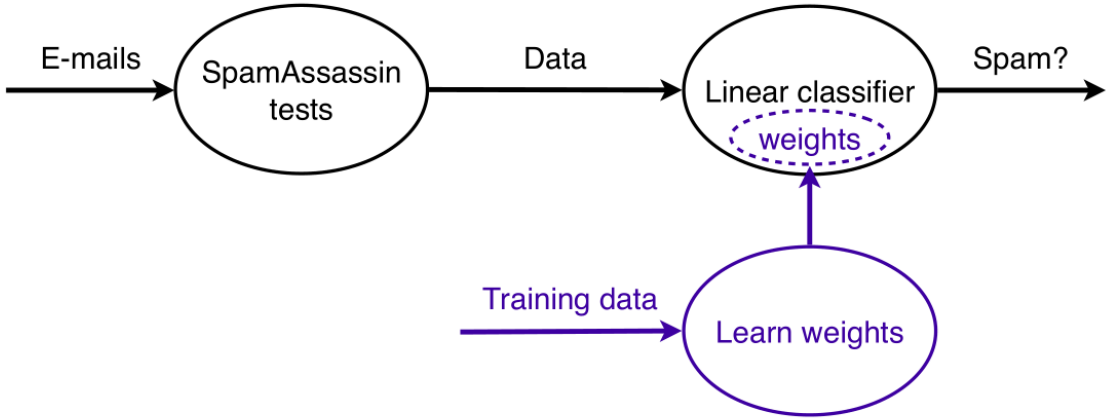
SpamAssassin is a widely used open-source spam filter. It calculates a score for an incoming e-mail, based on a number of built-in rules or ‘tests’ in SpamAssassin’s terminology, and adds a ‘junk’ flag and a summary report to the e-mail’s headers if the score is 5 or more.

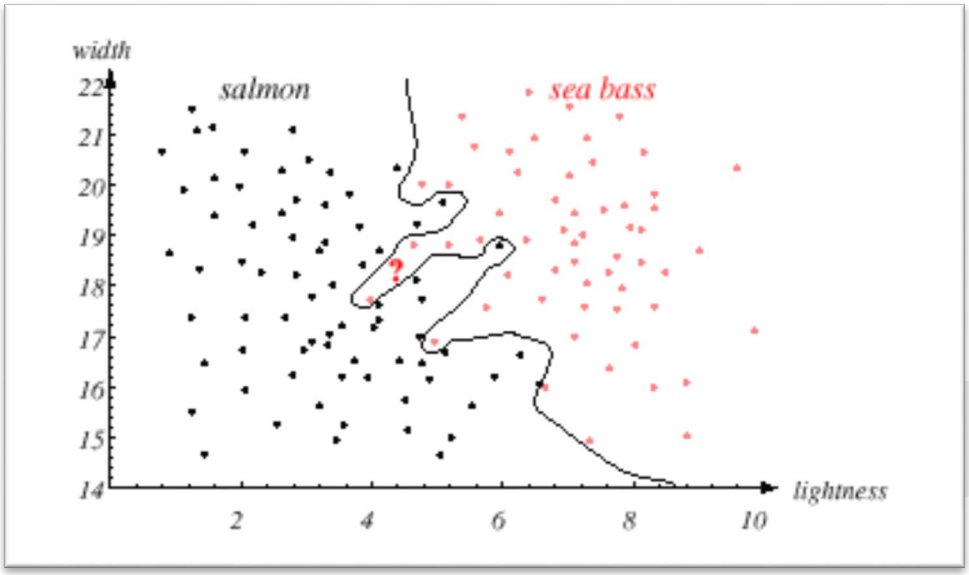
```
-0.1 RCVD_IN_MXRATE_WL       RBL: MXRate recommends allowing
                             [123.45.6.789 listed in sub.mxrate.net]
 0.6 HTML_IMAGE_RATIO_02     BODY: HTML has a low ratio of text to image area
 1.2 TVD_FW_GRAPHIC_NAME_MID BODY: TVD_FW_GRAPHIC_NAME_MID
 0.0 HTML_MESSAGE            BODY: HTML included in message
 0.6 HTML_FONx_FACE_BAD      BODY: HTML font face is not a word
 1.4 SARE_GIF_ATTACH         FULL: Email has a inline gif
 0.1 BOUNCE_MESSAGE          MTA bounce message
 0.1 ANY_BOUNCE_MESSAGE      Message is some kind of bounce message
 1.4 AWL                     AWL: From: address is in the auto white-list
```

E-mail	x_1	x_2	Spam?	$4x_1 + 4x_2$
1	1	1	1	8
2	0	0	0	0
3	1	0	0	4
4	0	1	0	4

The columns marked x_1 and x_2 indicate the results of two tests on four different e-mails. The fourth column indicates which of the e-mails are spam. The right-most column demonstrates that by thresholding the function $4x_1 + 4x_2$ at 5, we can separate spam from ham.


w
x0
x1
x2


E-mails
SpamAssassin tests
Data
Linear classifier
weights
Spam?
Training data
Learn weights



Overfitting

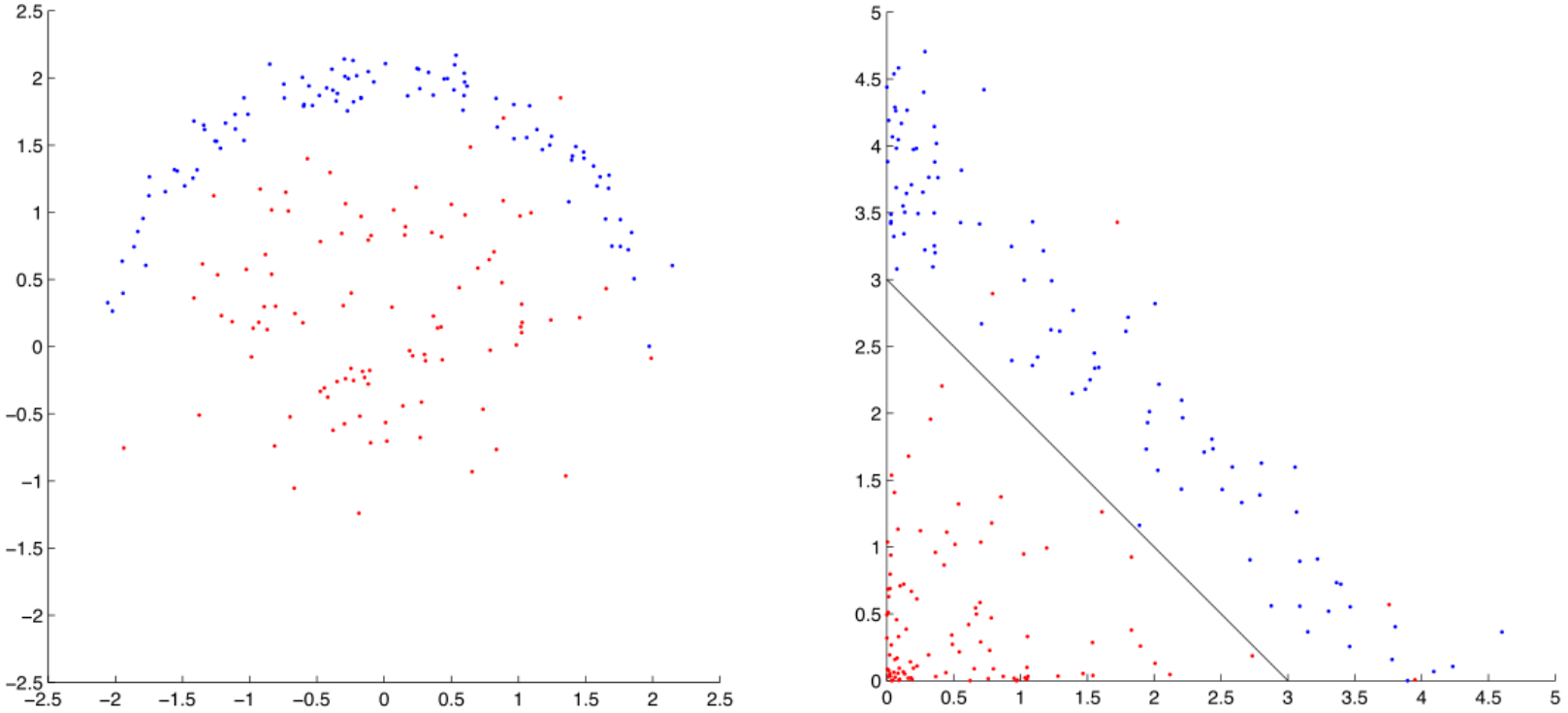
Imagine you are preparing for your *Machine Learning 101* exam. Helpfully, Professor Flach has made previous exam papers and their worked answers available online. You begin by trying to answer the questions from previous papers and comparing your answers with the model answers provided.

Unfortunately, you get carried away and spend all your time on memorising the model answers to all past questions. Now, if the upcoming exam completely consists of past questions, you are certain to do very well. But if the new exam asks different questions about the same material, you would be ill-prepared and get a much lower mark than with a more traditional preparation.

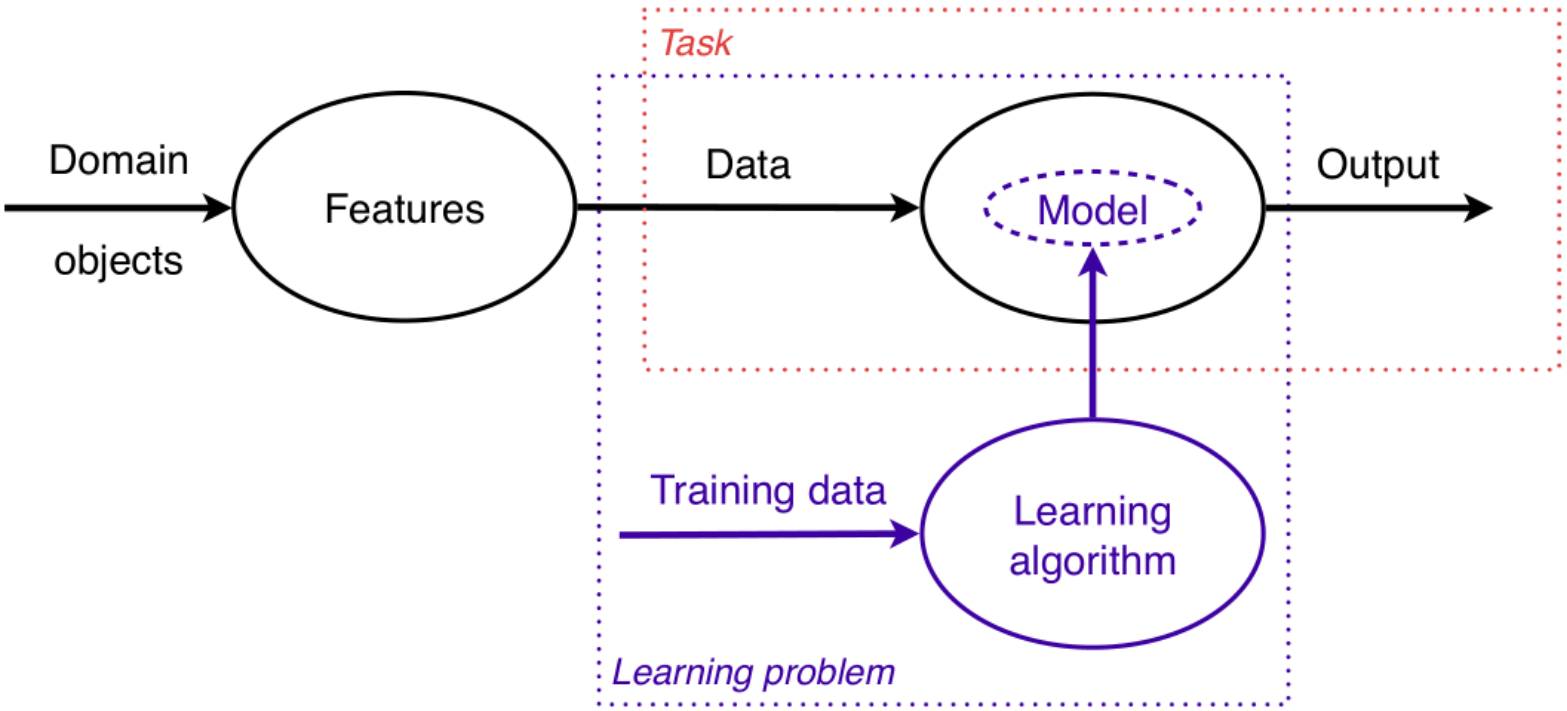
In this case, one could say that you were *overfitting* the past exam papers and that the knowledge gained didn't *generalise* to future exam questions.

Figure 1.11, p.43

Non-linearly separable data



(left) A linear classifier would perform poorly on this data. **(right)** By transforming the original (x, y) data into $(x', y') = (x^2, y^2)$, the data becomes more 'linear', and a linear decision boundary $x' + y' = 3$ separates the data fairly well. In the original space this corresponds to a circle with radius $\sqrt{3}$ around the origin.



An overview of how machine learning is used to address a given task. A task (red box) requires an appropriate mapping – a model – from data described by features to outputs. Obtaining such a mapping from training data is what constitutes a learning problem (blue box).

Machine learning models

Machine learning models can be distinguished according to their main intuition:

- *Geometric* models use intuitions from geometry such as separating (hyper-)planes, linear transformations and distance metrics.
- *Probabilistic* models view learning as a process of reducing uncertainty, modelled by means of probability distributions.
- *Logical* models are defined in terms of easily interpretable logical expressions.

Alternatively, they can be characterised by their *modus operandi*:

- *Grouping models* divide the instance space into segments; in each segment a very simple (e.g., constant) model is learned.
- *Grading models* learning a single, global model over the instance space.