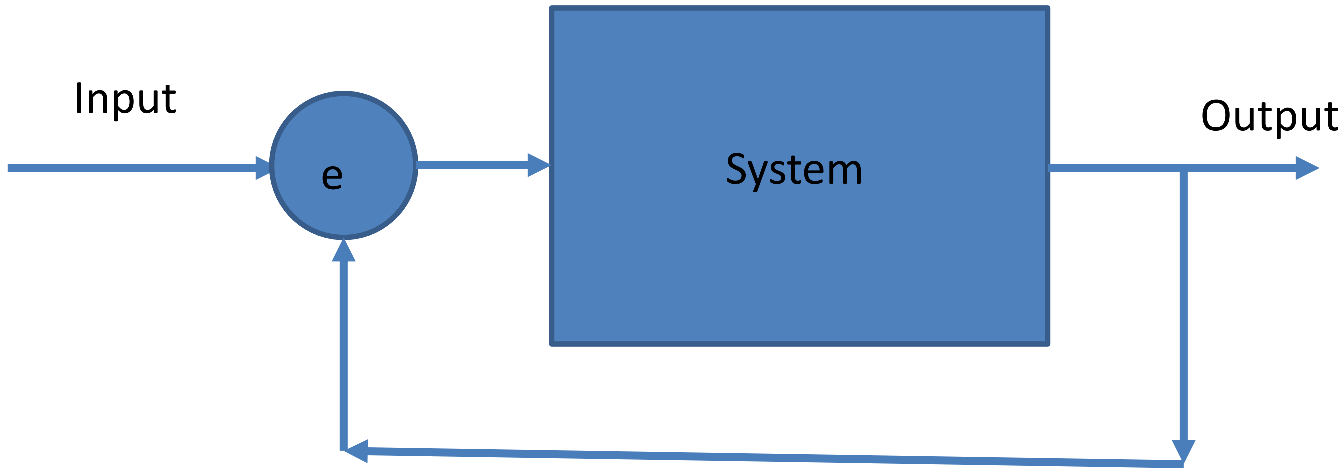
What is Machine Learning?



- Regression
- Classification

Pattern Classification

- A *pattern* is an *arrangement of descriptors*,

- The name feature is used often in the pattern recognition literature to denote a descriptor.

- A pattern class is a family of patterns that share some common properties.

Learning and Adaptation

- In supervised learning, a teacher provides a category label for each pattern in a training set. These are then used to train a classifier which can thereafter solve similar classification problems by itself.

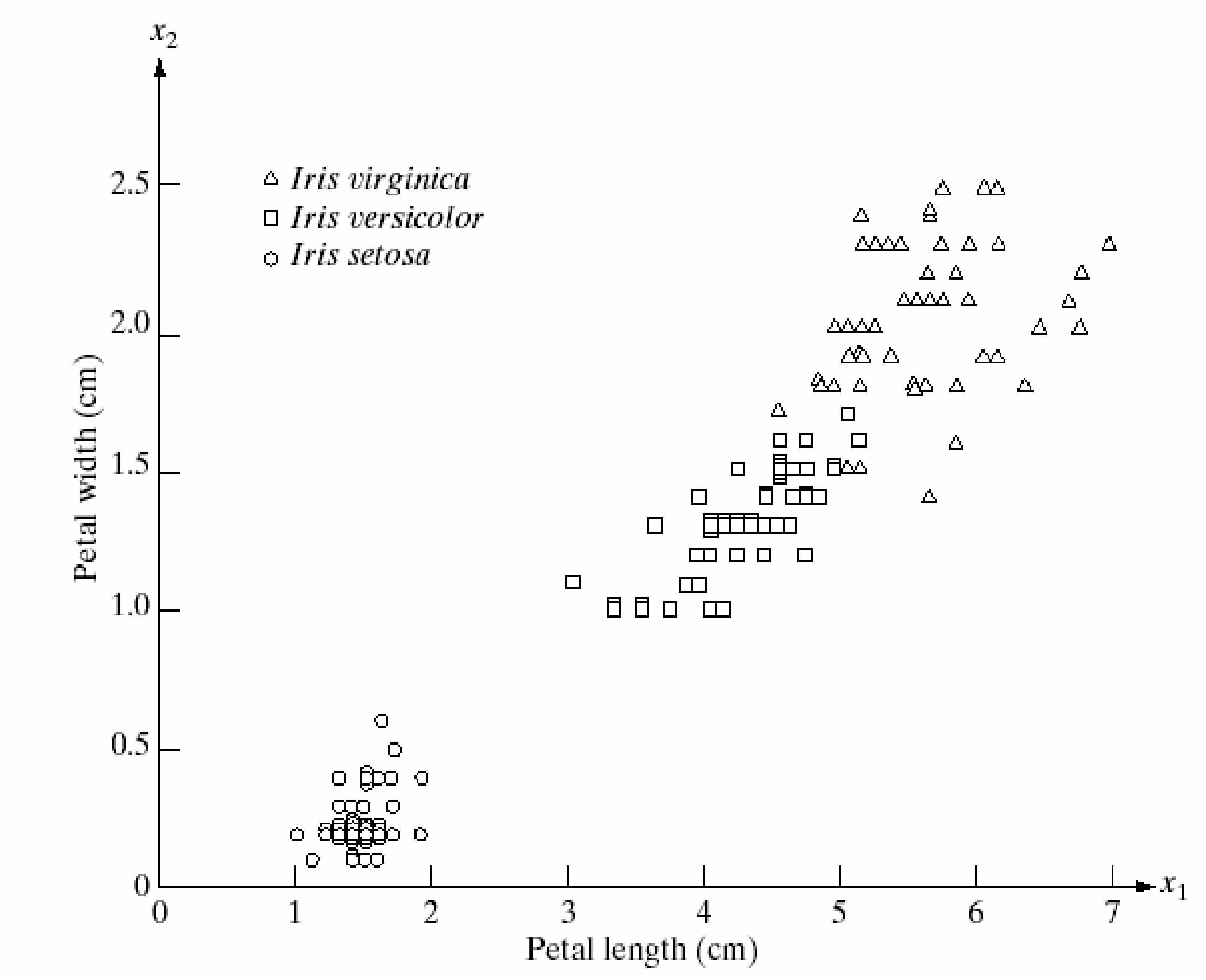
- In unsupervised learning, or clustering, there is no explicit teacher or training data. The system forms natural clusters or grouping of input patterns and classifies them based on clusters they belong to.

Learning and Adaptation

- In semi-supervised learning, a small set of labeled data is also available along with unlabeled training data.

- In reinforcement learning, a teacher only says to classifier whether it is right when suggesting a category for a pattern. The teacher does not tell what the correct category is.

An historical example of Fischer



Machine Learning for Object Detection

- What features do we use?
  - intensity, color, gradient information, boundary information, …
- Which machine learning methods?
  - generative vs. discriminative
  - k-nearest neighbors, boosting, SVMs, …
- What hacks do we need to get things working?

Recognition Based on Decision-Theoretic Methods

Let $x = \left(x_1, x_2, \quad , x_n\right)^T$ for W pattern classes $\omega_1, \omega_2, \ldots, \omega_W$

$$d_i\left(x\right) > d_j\left(x\right) \quad j = 1,2,\ldots,W; \; j \neq i$$

In other words, an unknown pattern $\mathbf{x}$ is said to belong to the ith pattern class if, upon substitution of $\mathbf{x}$ into all decision functions, $d_i(x)$ yields the largest numerical value.

Matching Minimum distance classifier

- Suppose that we define the prototype of each pattern class to be the mean vector of the patterns of that class: $m_j = \frac{1}{N_j} \sum_{x \in \omega_j} x_j \quad j = 1,2,\ldots,W$
- We then assign $\mathbf{x}$ to class ω_i if $D_i(\mathbf{x})$ is the smallest distance. $D_j(x) = \left\| x - m_j \right\|$

Minimum distance classifier

- It is not difficult to show that selecting the smallest distance is equivalent to evaluating the functions $d_j(x) = x^T m_j - \frac{1}{2} m_j^T m_j \quad j = 1,2,\ldots,W$
- assign $\mathbf{x}$ to class ω_i if $d_i(\mathbf{x})$ is the largest numerical value.
- This formulation agrees with the concept of a decision function

Minimum distance classifier

Two steps:
-“learning”
- actual recognition

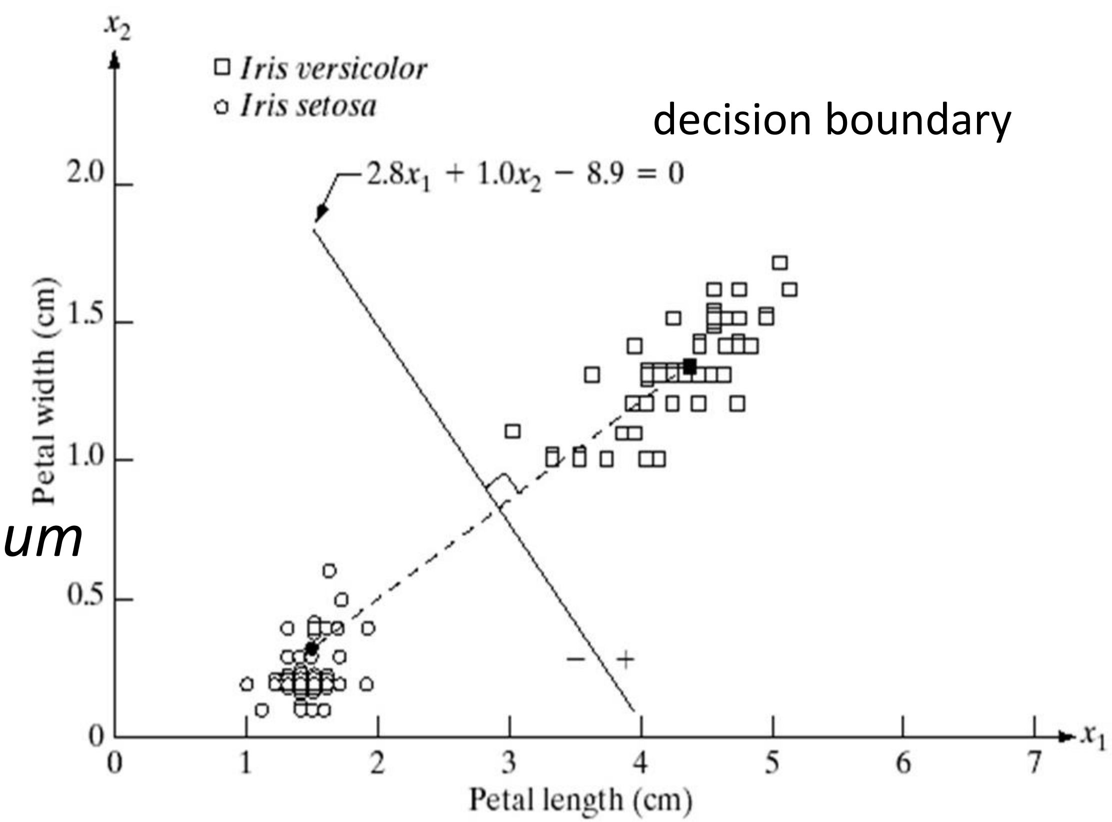


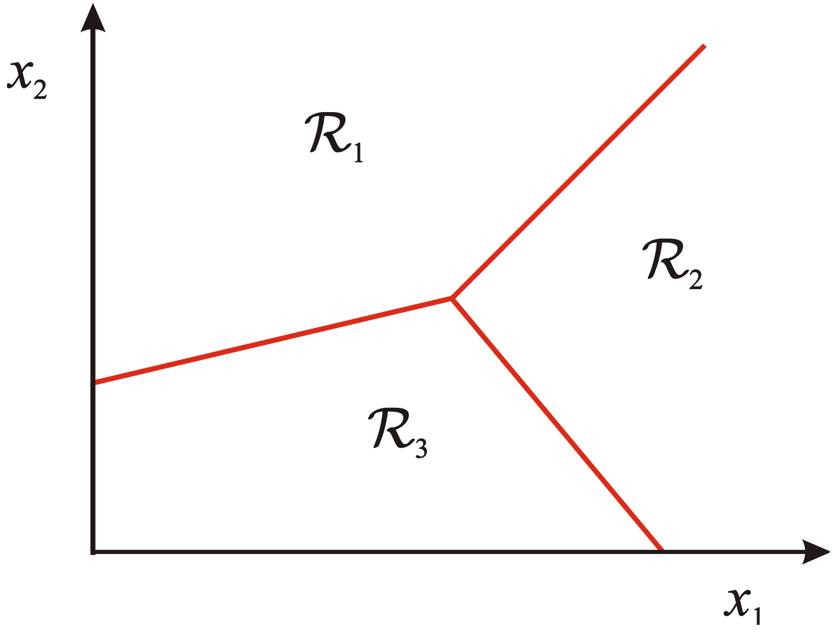
FIGURE 12.6 Decision boundary of minimum distance classifier for the classes of ***Iris versicolor*** and ***Iris setosa***. The dark dot and square are the means.

This example: *minimum distance classifier.*

Pattern vectors (PVs) can be generated in numerous ways.
Selecting the descriptors on which to base each component of the PV has a profound influence on the performance of the pattern recognition algorithm.

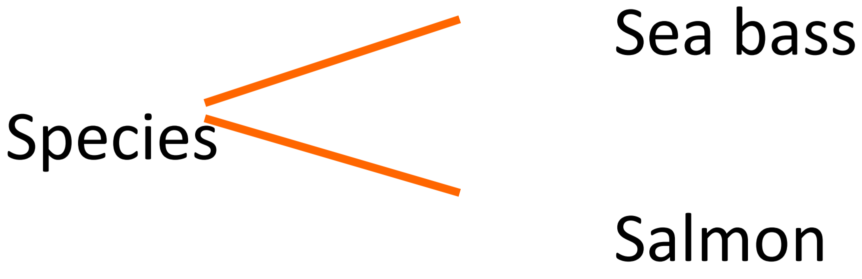
Classification

- Assign input vector to one of two or more classes
- Any decision rule divides input space into *decision regions* separated by *decision boundaries*



Another Example

“Sorting incoming Fish on a conveyor according to species using optical sensing”



Species

Sea bass

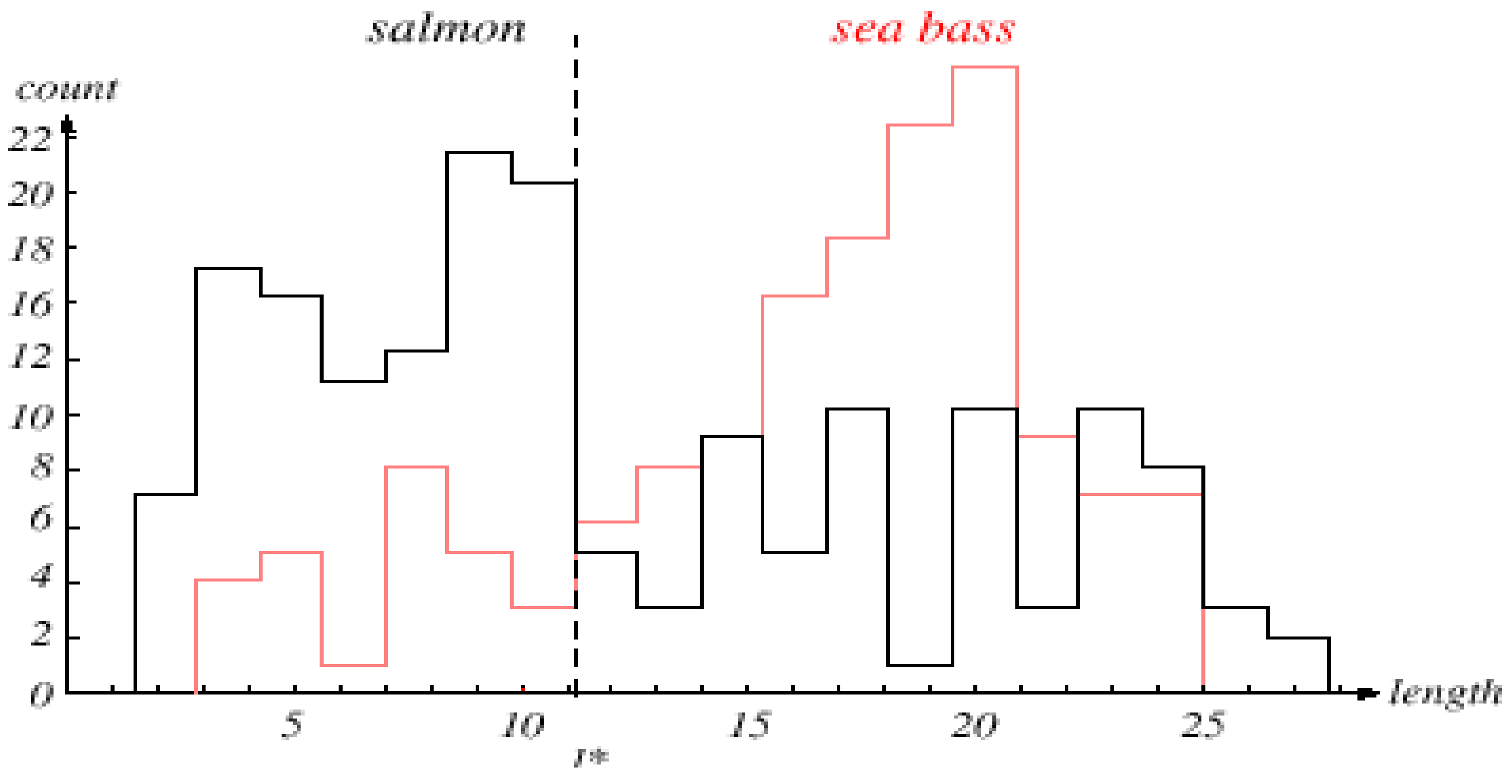
Salmon

Problem Analysis

- Set up a camera and take some sample images to extract features
  - Length
  - Lightness
  - Width
  - Number and shape of fins
  - Position of the mouth, etc…

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

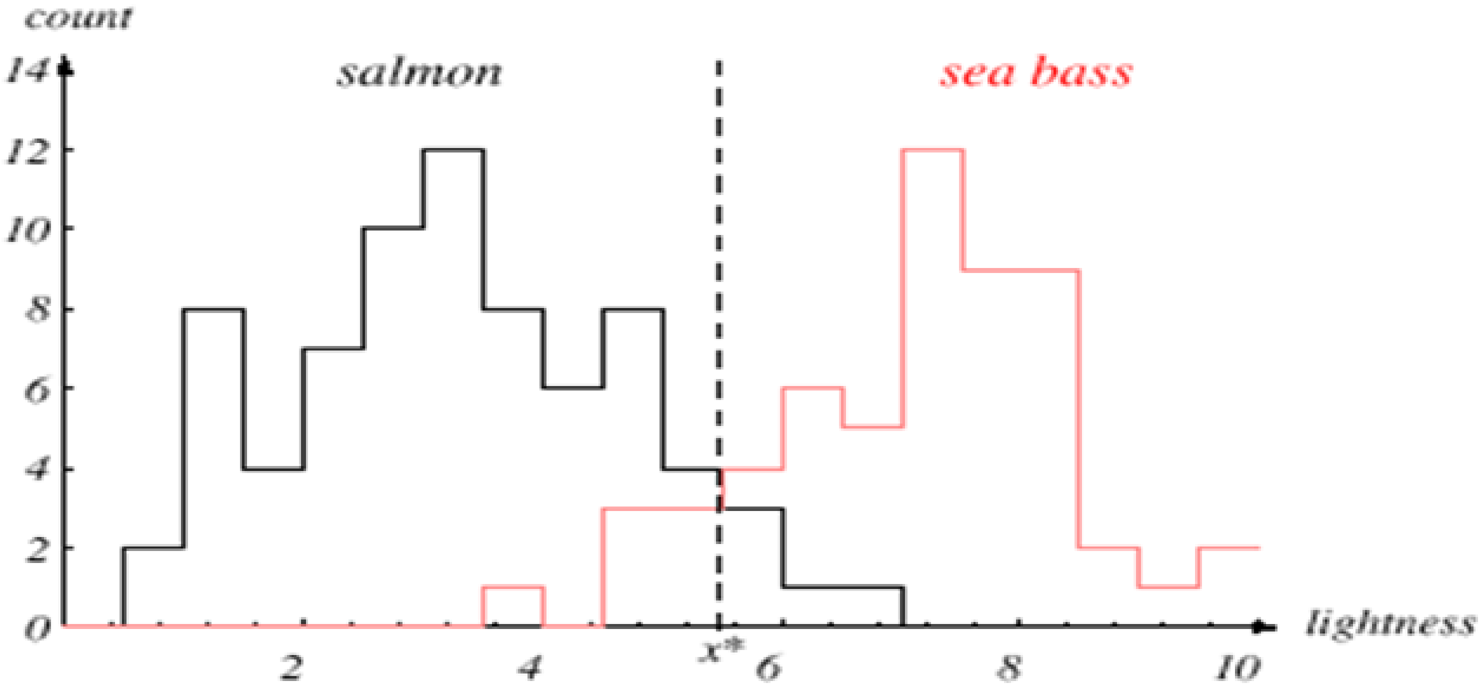
Feature Selection



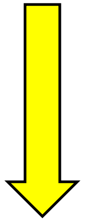
The length is a poor feature alone!

Select the lightness as a possible feature.

Feature Selection



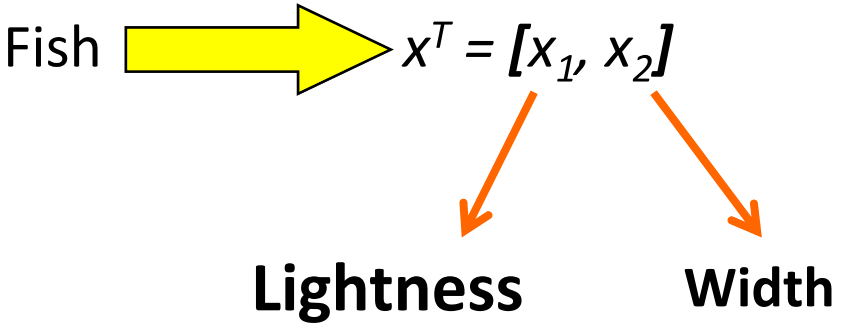
- Threshold decision boundary and cost relationship
  - Move our decision boundary toward smaller values of lightness in order to minimize the cost (reduce the number of sea bass that are classified salmon!)



Task of decision theory

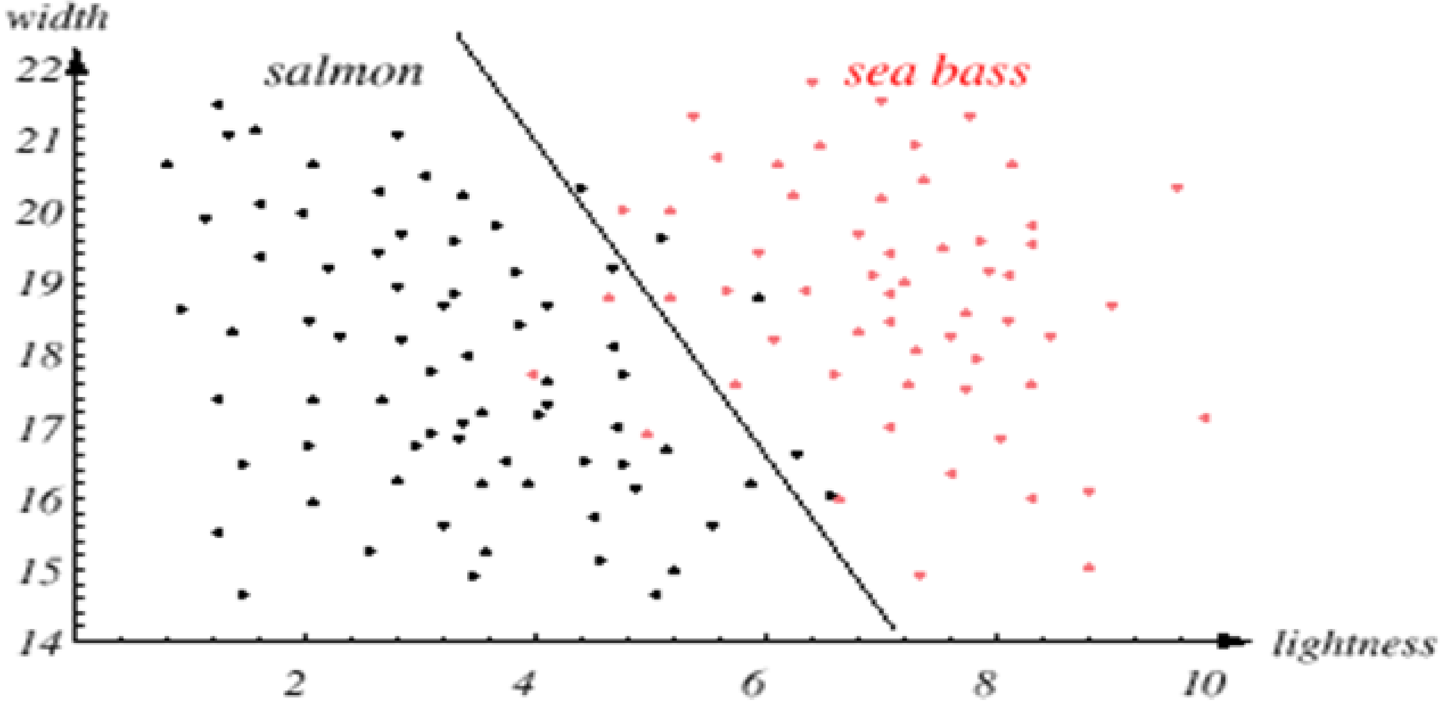
Feature Selection & Decision Boundary

- Adopt the lightness and add the width of the fish



Fish $\rightarrow$ $x^T = [x_1, x_2]$

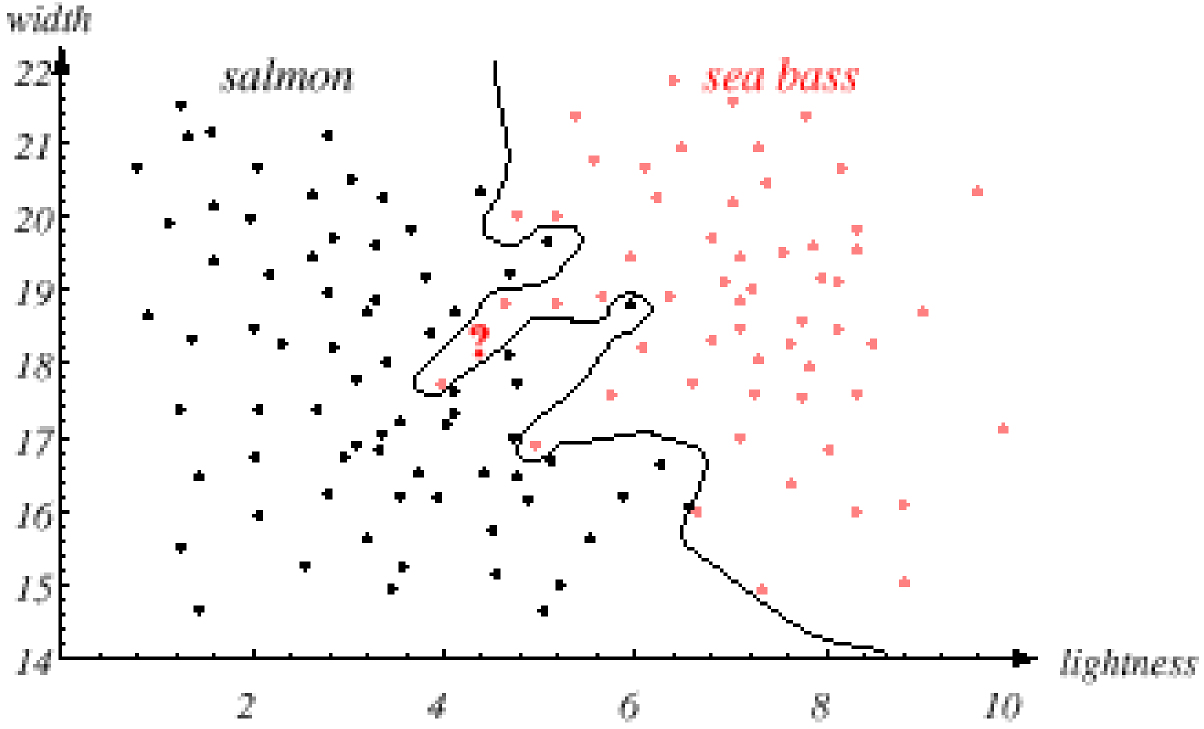
Lightness **Width**



Modeling/Training Classifier

- We might add other features that are not correlated with the ones we already have. A precaution should be taken not to reduce the performance by adding such “noisy features”

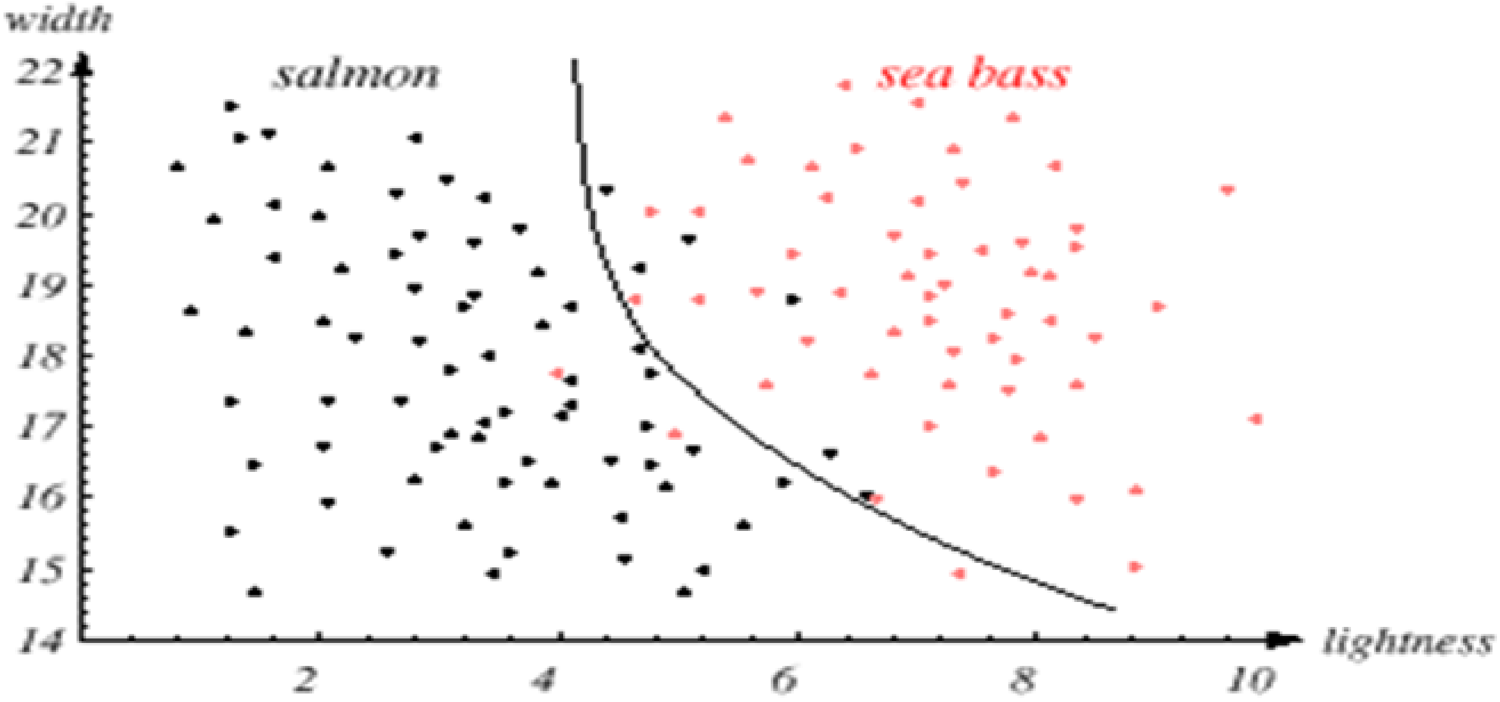
- Ideally, the best decision boundary should be the one which provides an optimal performance such as in the following figure:



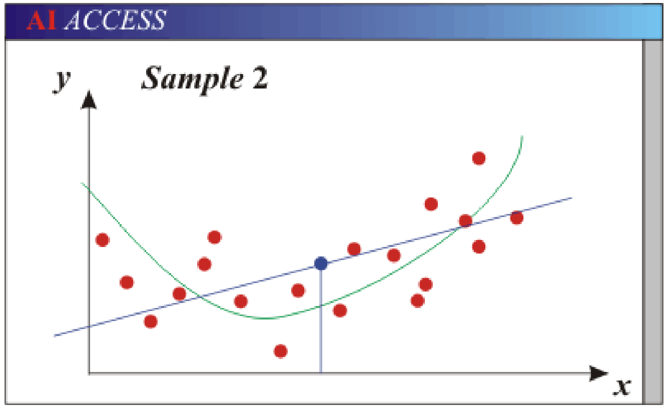
Generalization

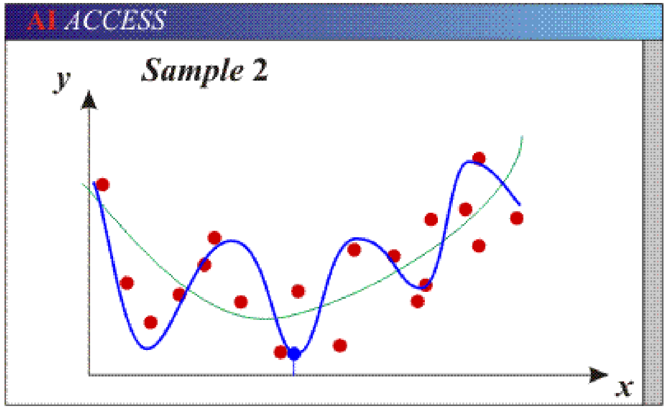
- However, our satisfaction is premature because the central aim of designing a classifier is to correctly classify novel input

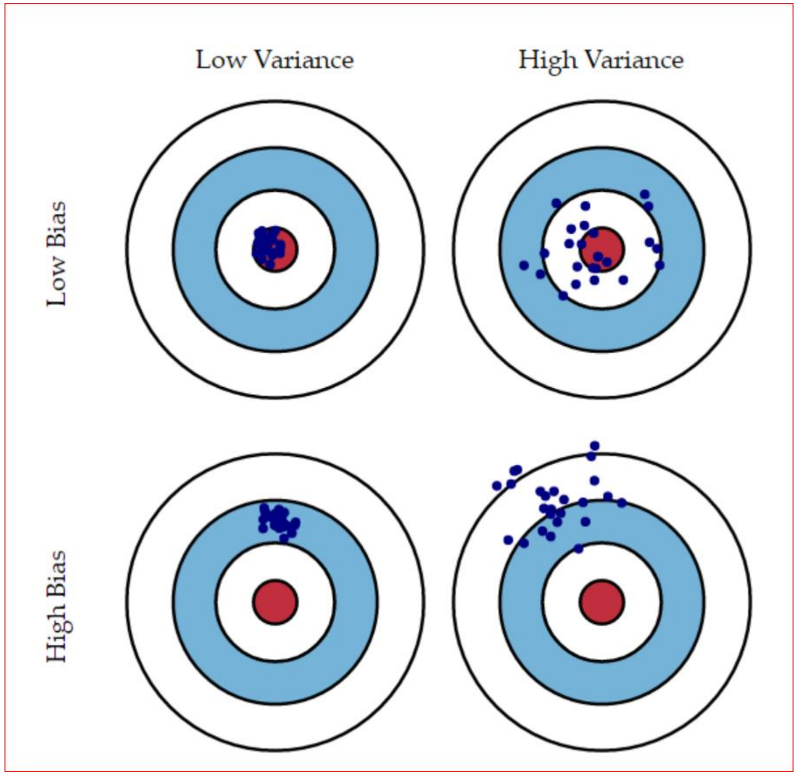
Issue of generalization!



Generalization: Bias-Variance Trade-off







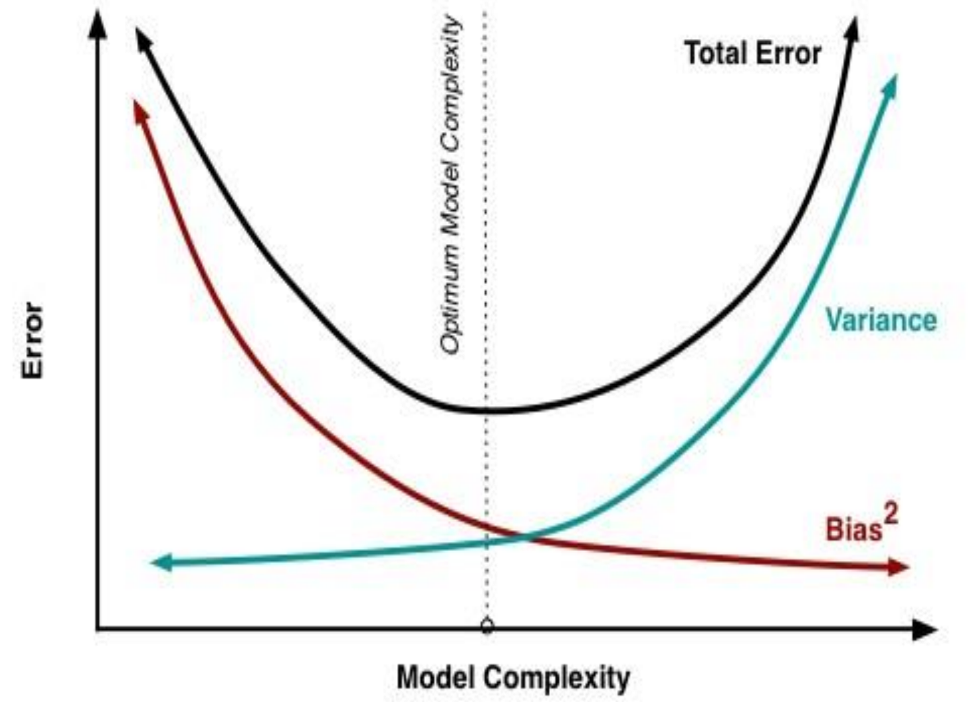
- Models with too few parameters are inaccurate because of a large bias (not enough flexibility).

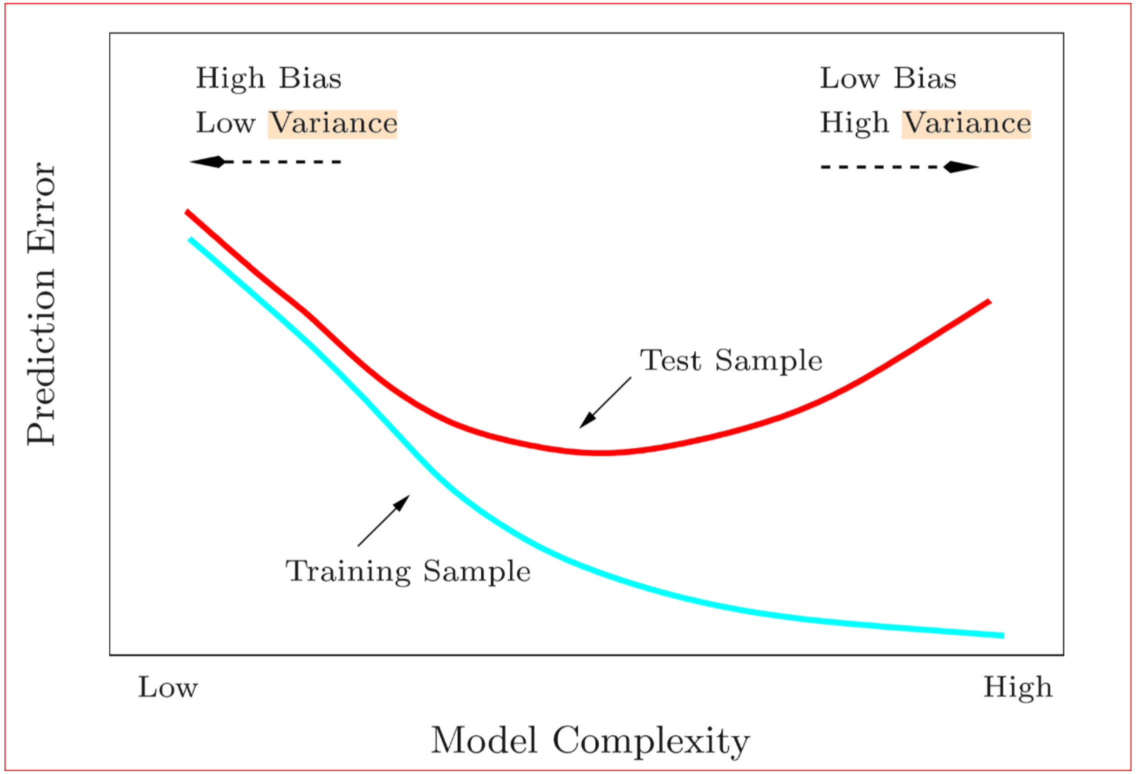
- Models with too many parameters are inaccurate because of a large variance (too much sensitivity to the sample).

Generalization: Bias-Variance Trade-off

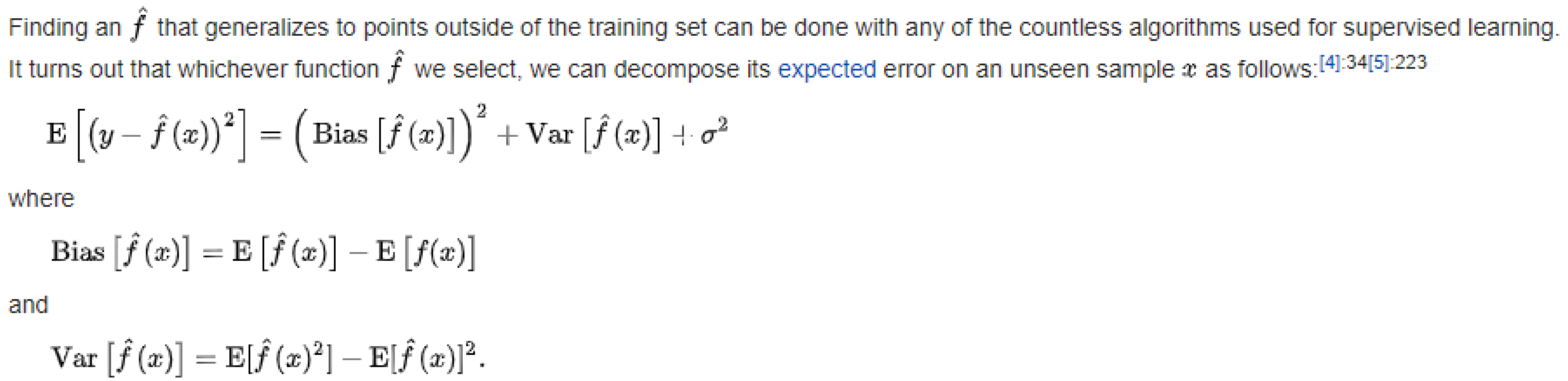
- Components of generalization error
  - **Bias:** how much the average model over all training sets differ from the true model?
    - Error due to inaccurate assumptions/simplifications made by the model
  - **Variance:** how much models estimated from different training sets differ from each other
- **Underfitting:** model is too “simple” to represent all the relevant class characteristics
  - High bias and low variance
  - High training error and high test error
- **Overfitting:** model is too “complex” and fits irrelevant characteristics (noise) in the data
  - Low bias and high variance
  - Low training error and high test error

Bias-Variance Trade-off





Why square of Bias?



Finding an $\hat{f}$ that generalizes to points outside of the training set can be done with any of the countless algorithms used for supervised learning. It turns out that whichever function $\hat{f}$ we select, we can decompose its expected error on an unseen sample x as follows:[4]:34[5]:223

$$\mathrm{E}\left[\left(y-\hat{f}(x)\right)^2\right] = \left(\mathrm{Bias}\left[\hat{f}(x)\right]\right)^2 + \mathrm{Var}\left[\hat{f}(x)\right] + \sigma^2$$

where

$$\mathrm{Bias}\left[\hat{f}(x)\right] = \mathrm{E}\left[\hat{f}(x)\right] - \mathrm{E}\left[f(x)\right]$$

and

$$\mathrm{Var}\left[\hat{f}(x)\right] = \mathrm{E}[\hat{f}(x)^2] - \mathrm{E}[\hat{f}(x)]^2.$$

No Free Lunch Theorem

In a supervised learning setting, we can't tell which classifier will have best generalization

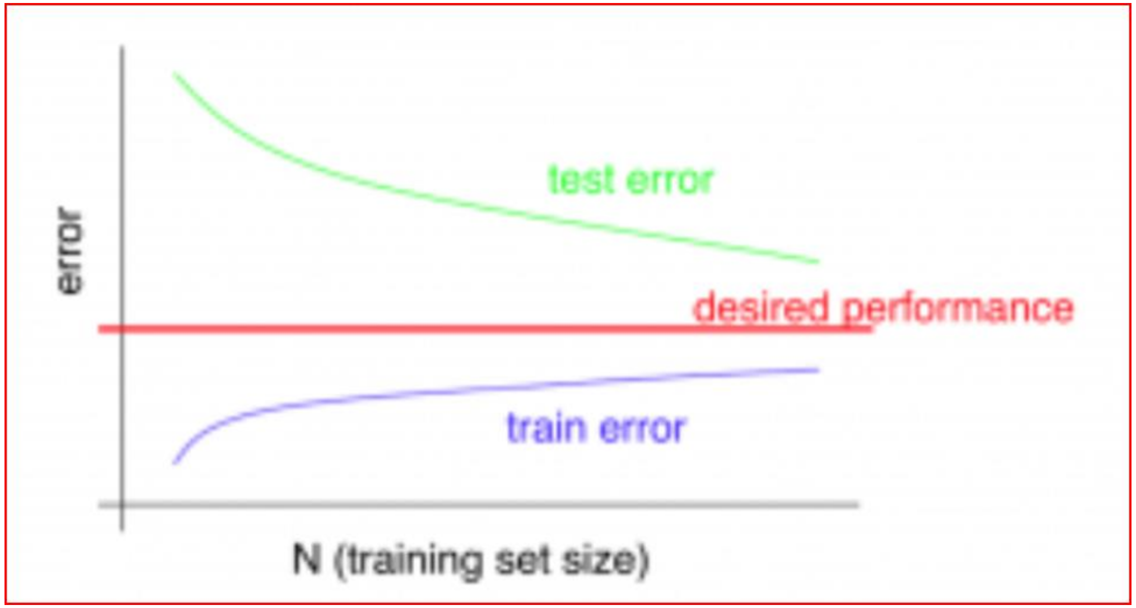
Remember…

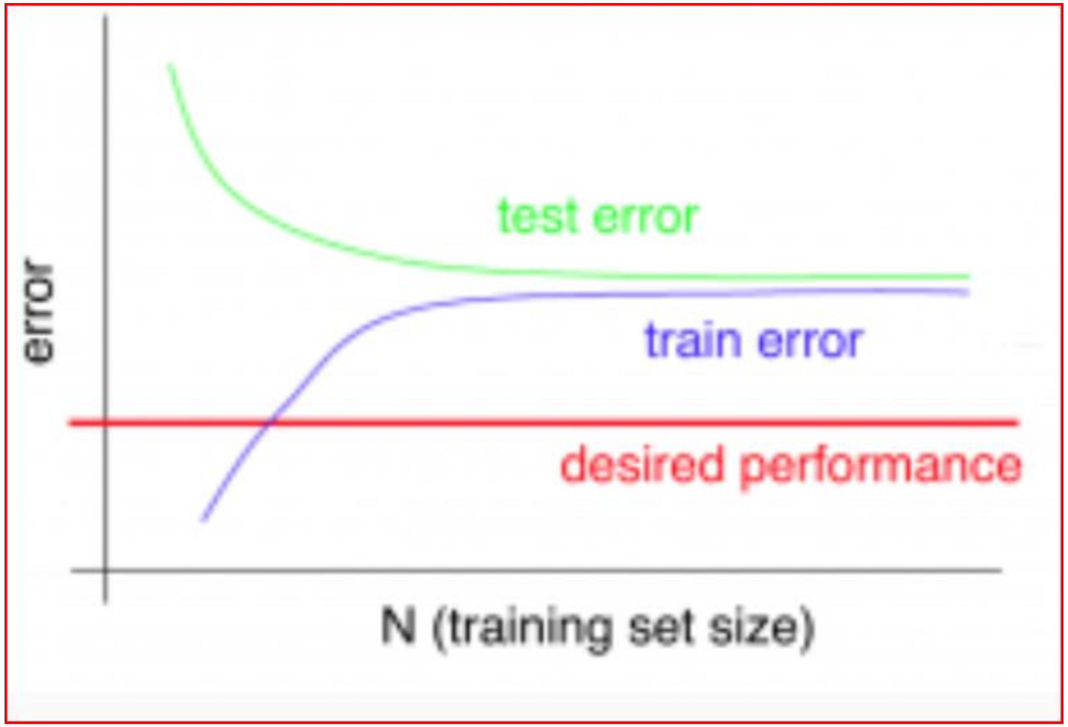
No classifier is inherently better than any other: you need to make assumptions to generalize

- Three kinds of error
  - Inherent: unavoidable
  - Bias: due to over-simplifications
  - Variance: due to inability to perfectly estimate parameters from limited data

How to reduce variance?

- Choose a simpler classifier
- Regularize the parameters
- Get more training data





How do you reduce bias?