Machine Learning

PHYS 453 – Spring 2024

Dr. Daugherity

ABILENE CHRISTIAN UNIVERSITY

Introduction to Classifiers

SUPERVISED LEARNING

http://udlbook.com

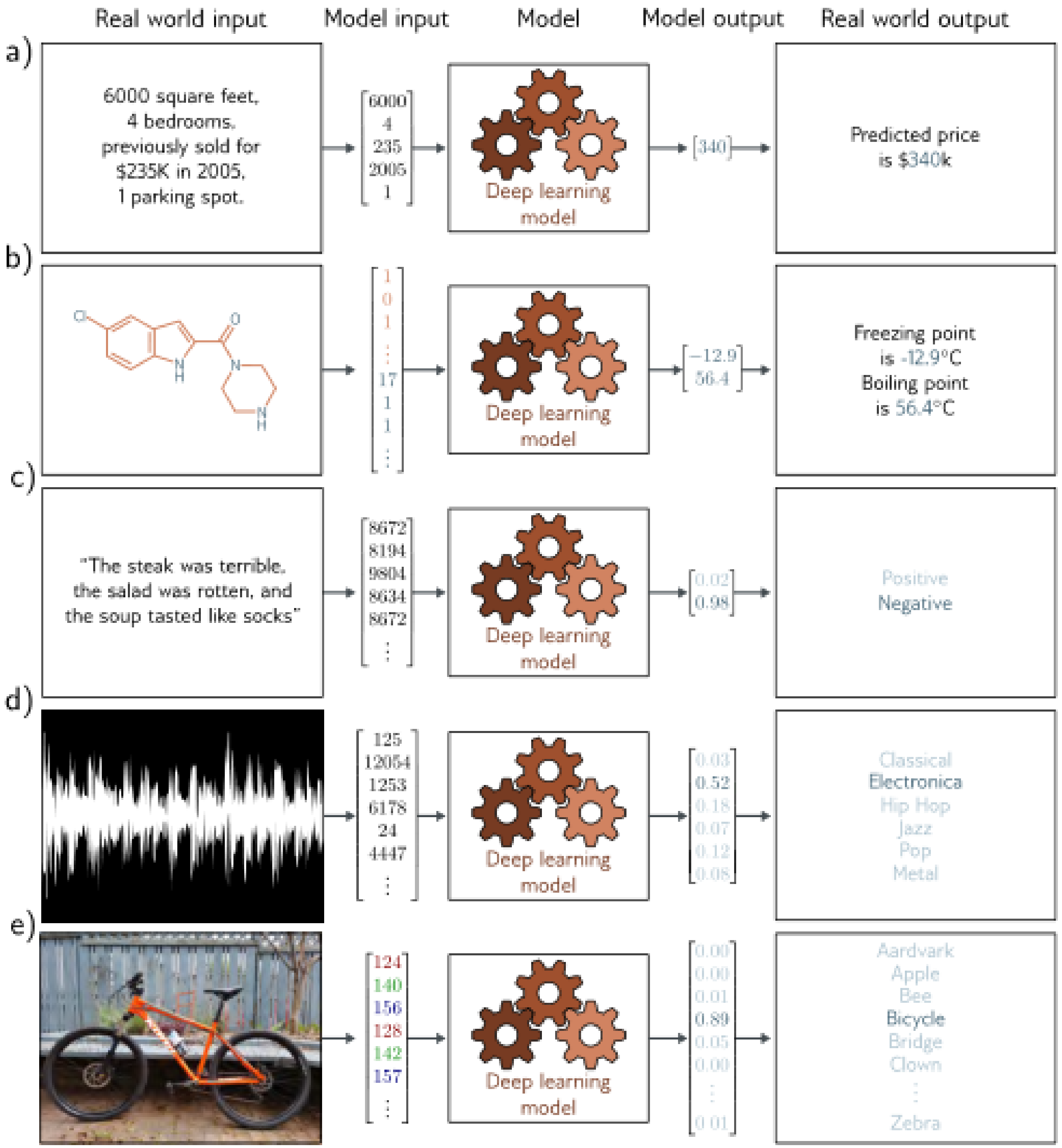


Figure 1.2 Regression and classification problems. a) This *regression* model takes a vector of numbers that characterize a property and predicts its price. b) This *multivariate regression* model takes the structure of a chemical molecule and predicts its melting and boiling points. c) This *binary classification* model takes a restaurant review and classifies it as either positive or negative. d) This *multiclass classification* problem assigns a snippet of audio to one of N genres. e) A second multiclass classification problem in which the model classifies an image according to which of N possible objects it might contain.

Supervised Learning

- **Supervised Learning** – decisions based on known training data
- **Classification** – decide which **category** a sample belongs to
- Examples:
  - Email or spam
  - Speech recognition
  - OCR (Optical Character Recognition)
- *i.e. write a program that can recognize patterns:*

Learning Types

- Supervised learning: teacher provides a category label or cost for each pattern in the training set

- Unsupervised learning: system forms clusters or “natural groupings” of the input patterns

A Relevant Example

Sorting incoming fish according to species using optical sensing

Salmon

Sea Bass

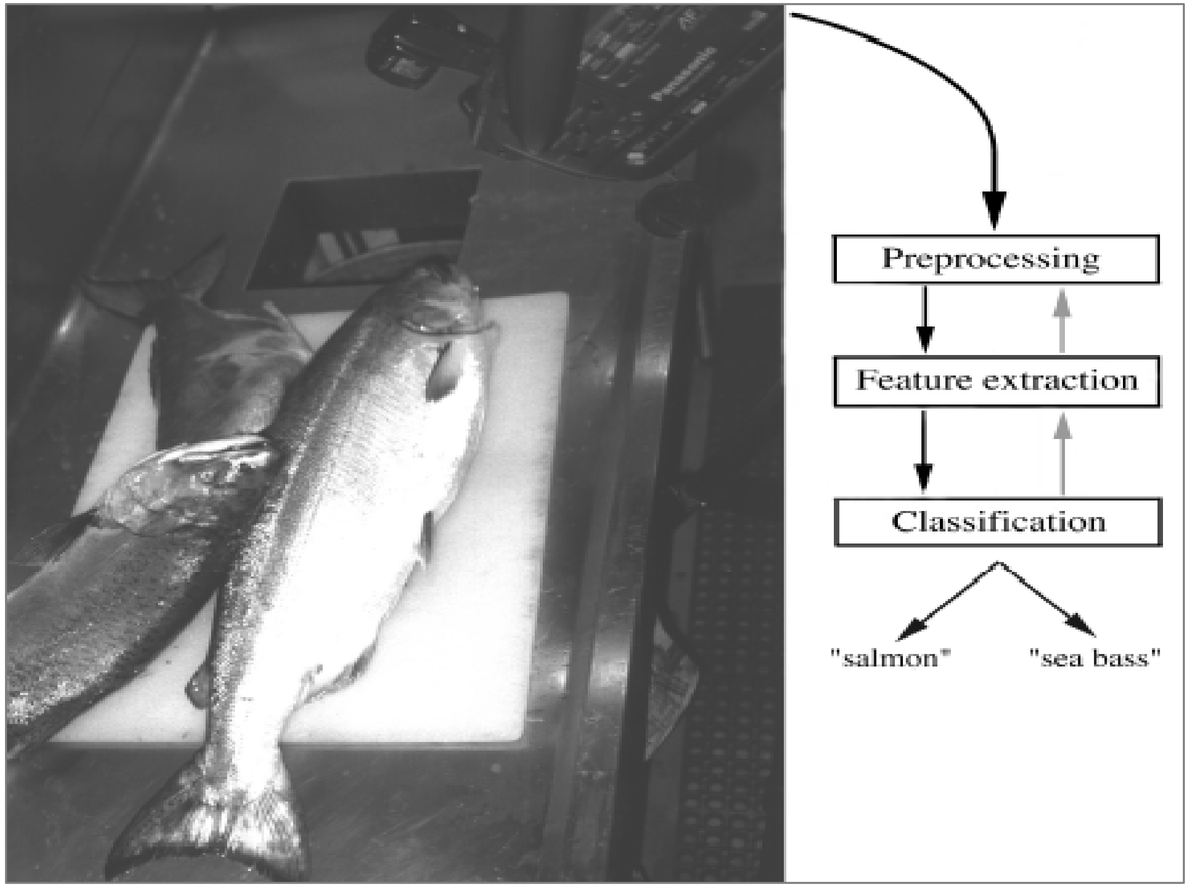
Theory

- Features:
  - Turn a sample into some numbers
  - Curse of Dimensionality
  - Symmetries!
- Models:
  - Lots and lots and lots of classifiers
  - “No Free Lunch” Theorem, and no silver bullets either
- Training: “learn” and evaluate
- Repeat?

Features

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

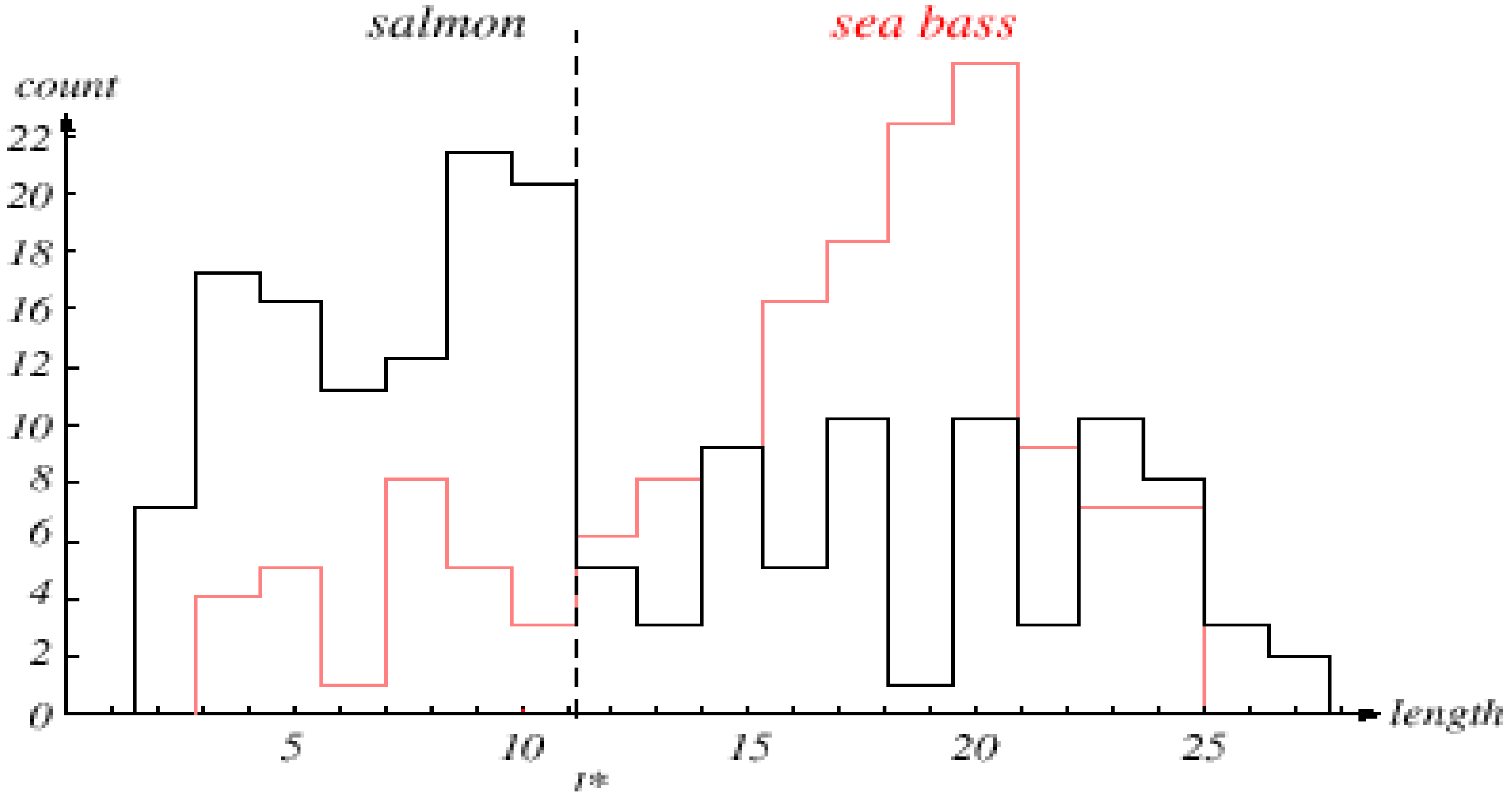
- Length
- Lightness
- Width
- Fins
- Position of the mouth
- etc…



No guarantee ahead of time what will work!

Classification

How well does length do?



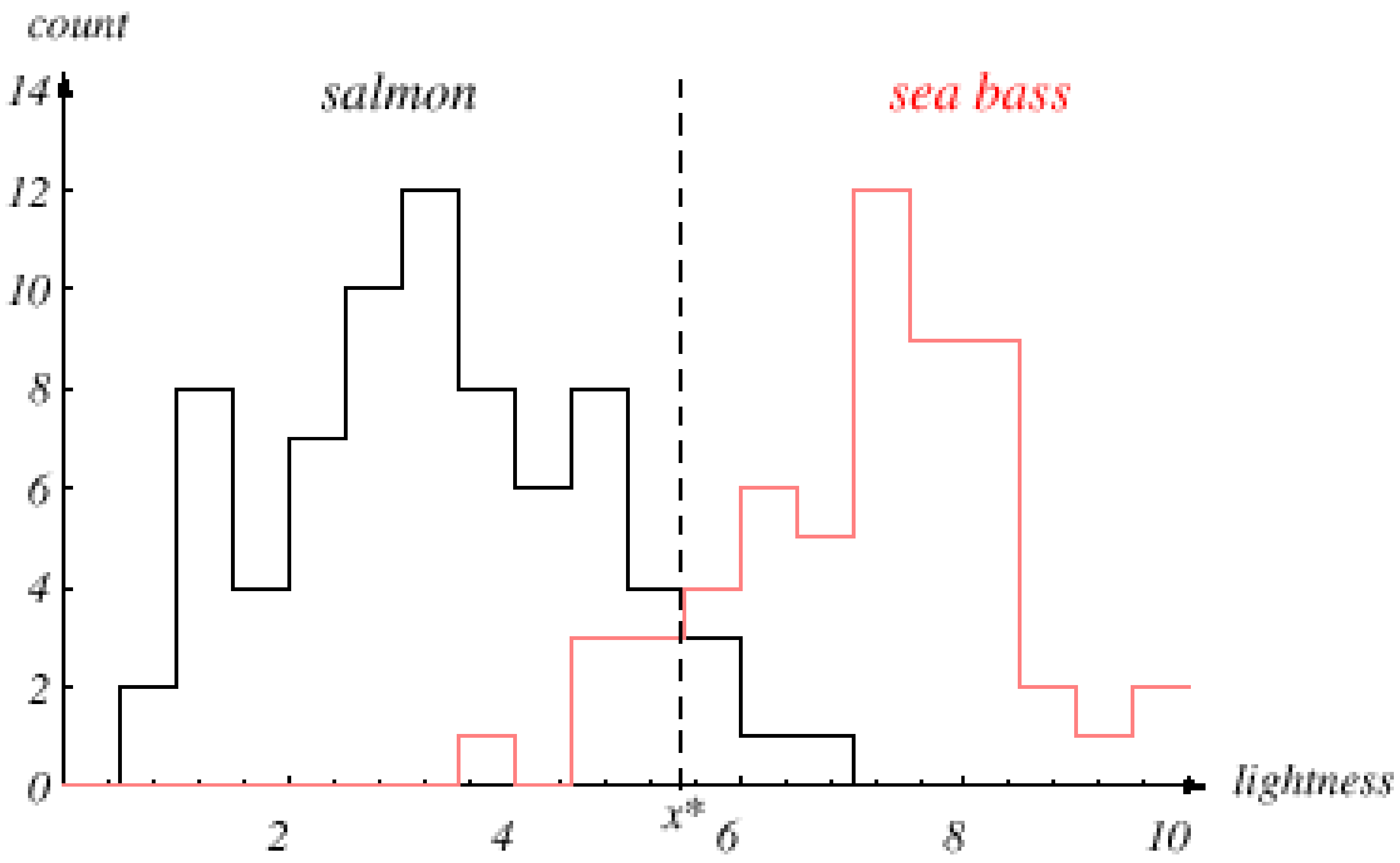
Decision Boundary – optimal classification threshold

Any fish with length < 11 = salmon, length > 11 = sea bass

Classification

The length is a poor feature alone!

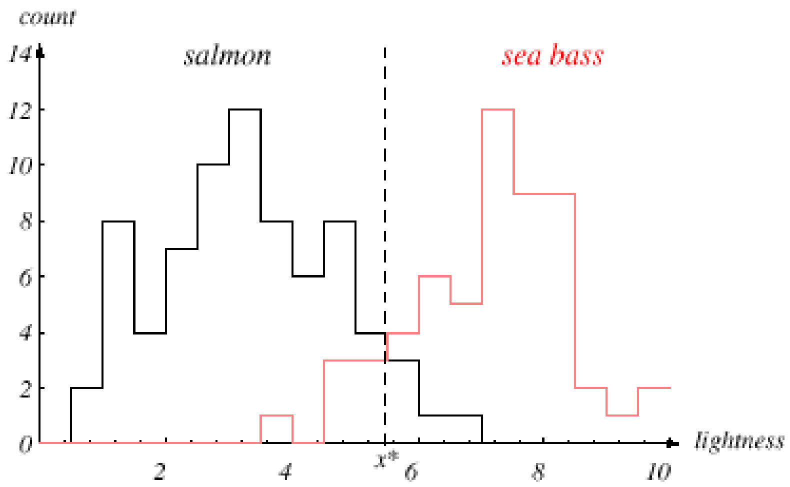
Select the lightness as a possible feature.

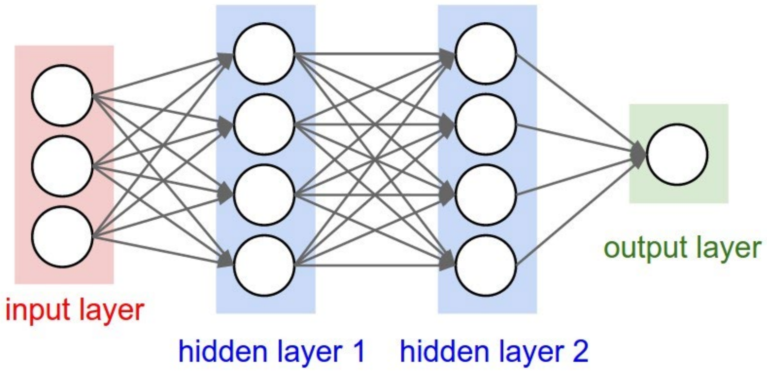


Model

This was our first example of a **model:** the way we calculate the prediction from the input.

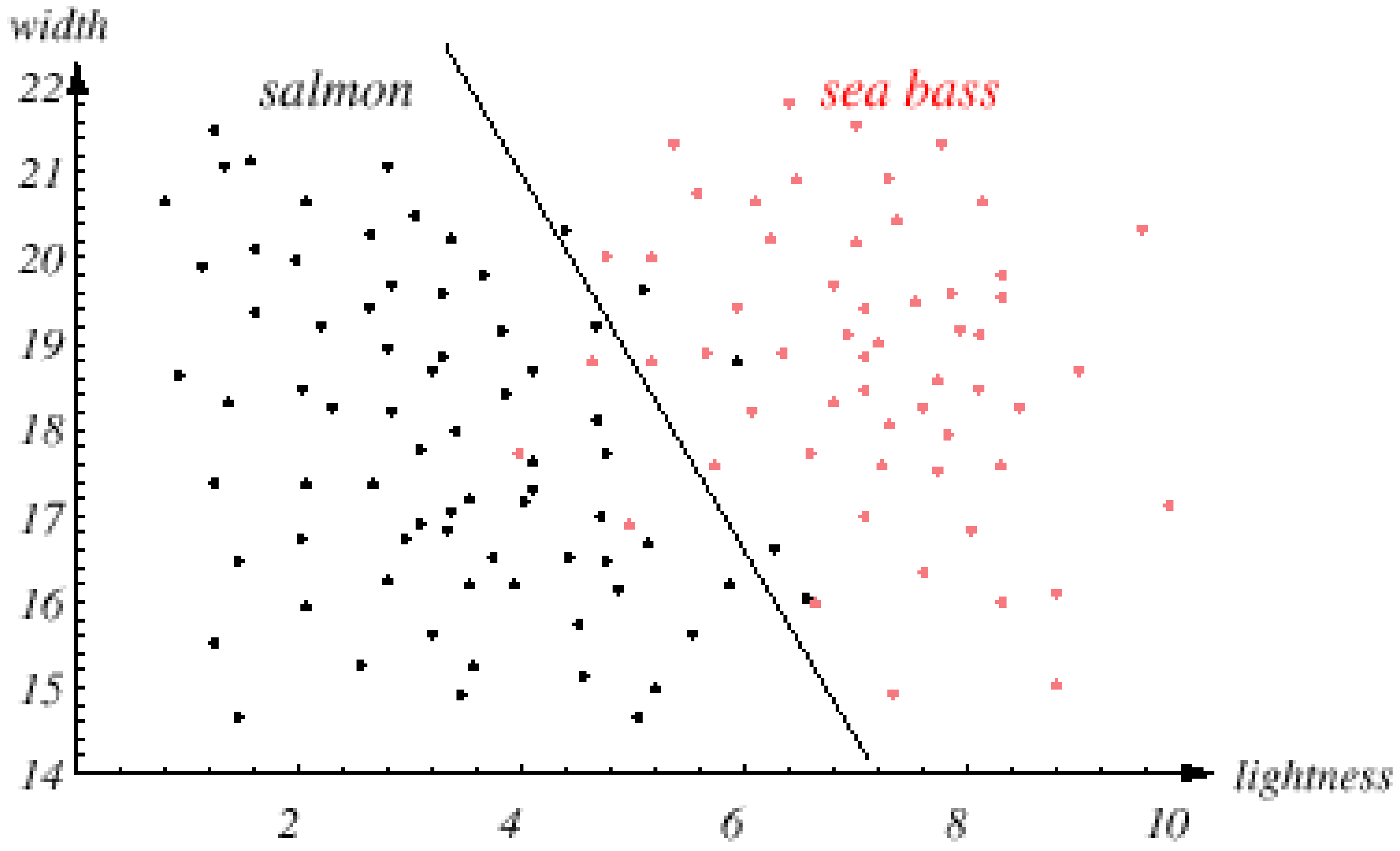
- Can be as simple as an if statement or super complicated
- Training: algorithm for how our model "learns" by using the training data to set parameters
- Predicting: using the model to classify a new sample





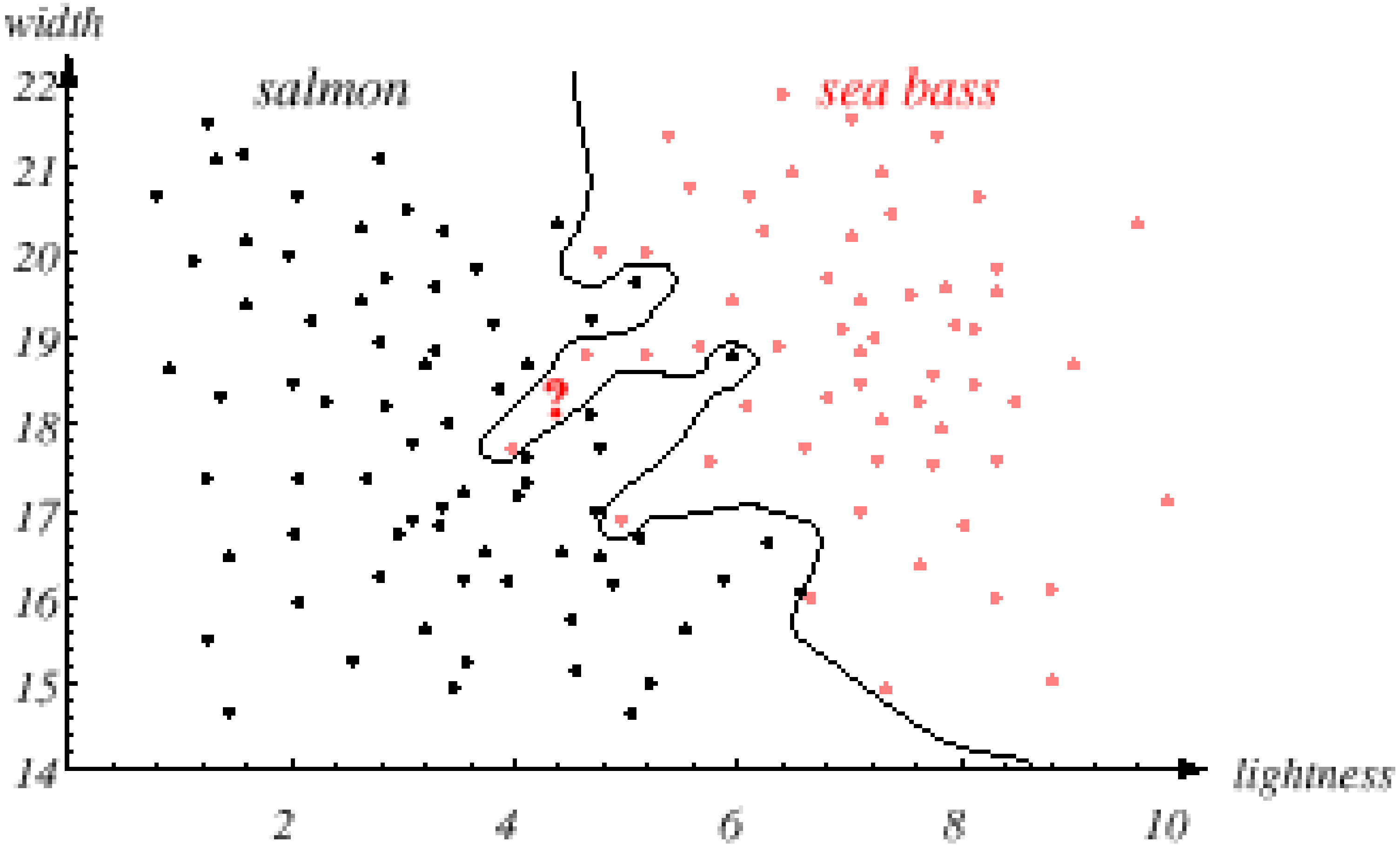
Two Features

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



2D Decision Boundary

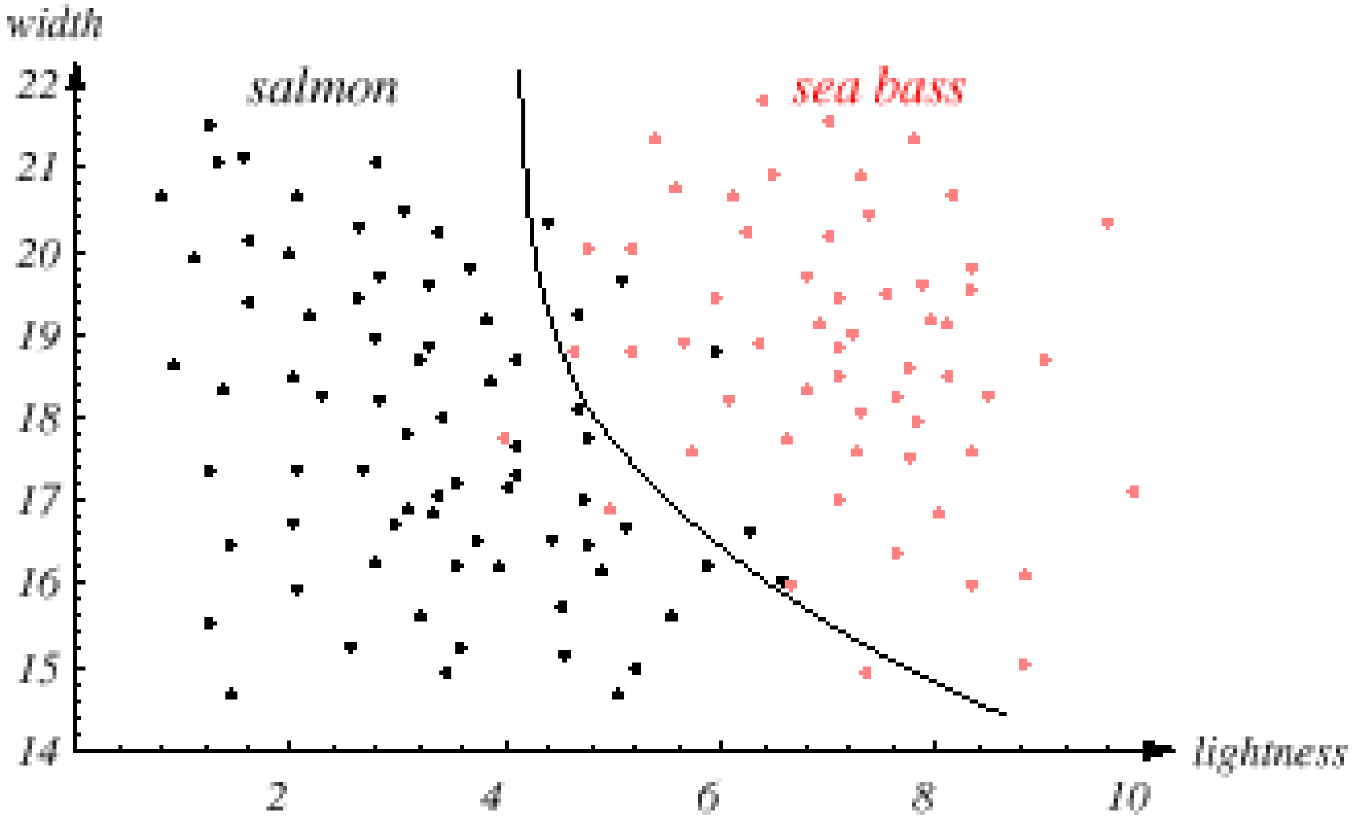
Optimal Decision Boundary?

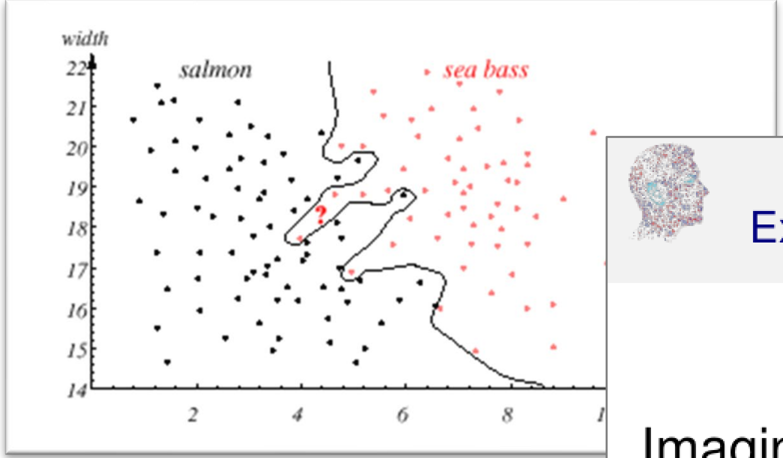


Overfitting – you nailed the training data, but do poorly on different fish. Model doesn't *generalize* well.

Generalization

Goldilocks' third choice?





Example 2, p.6

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.

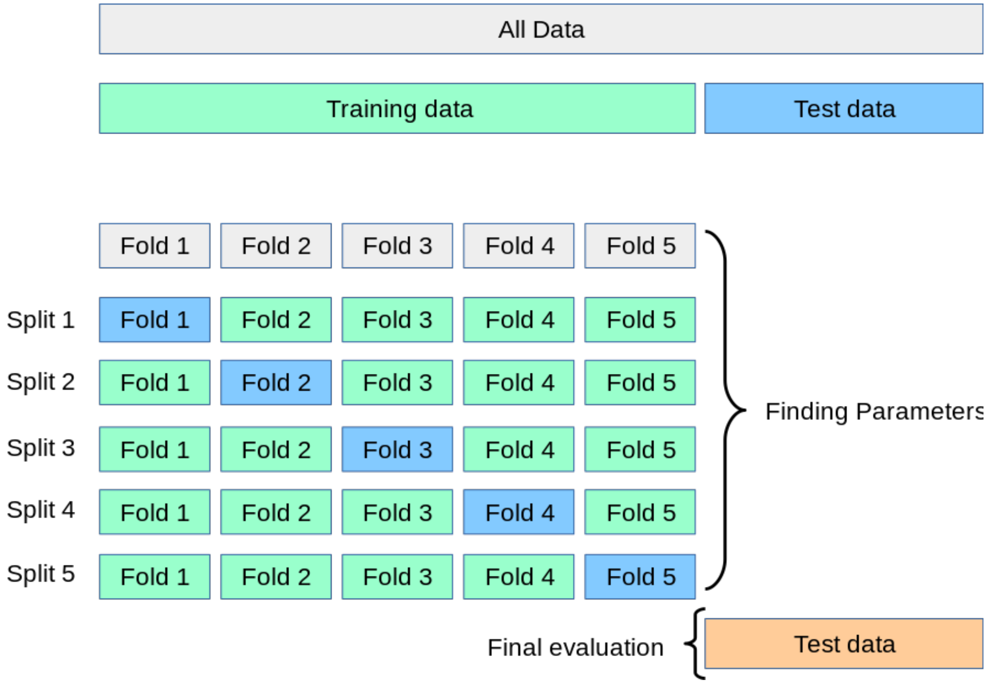
In this case, one could say that you were *overfitting* the past exam papers and that the knowledge gained didn't *generalise* to future exam questions.

Validation

- **Overfitting** – model trained to peculiarities in the training data rather than general samples
- **Underfitting** – model not able to perform well on any data

Later we will talk about some fairly simple tricks to **validate** that things are working properly.

Test/Train Split

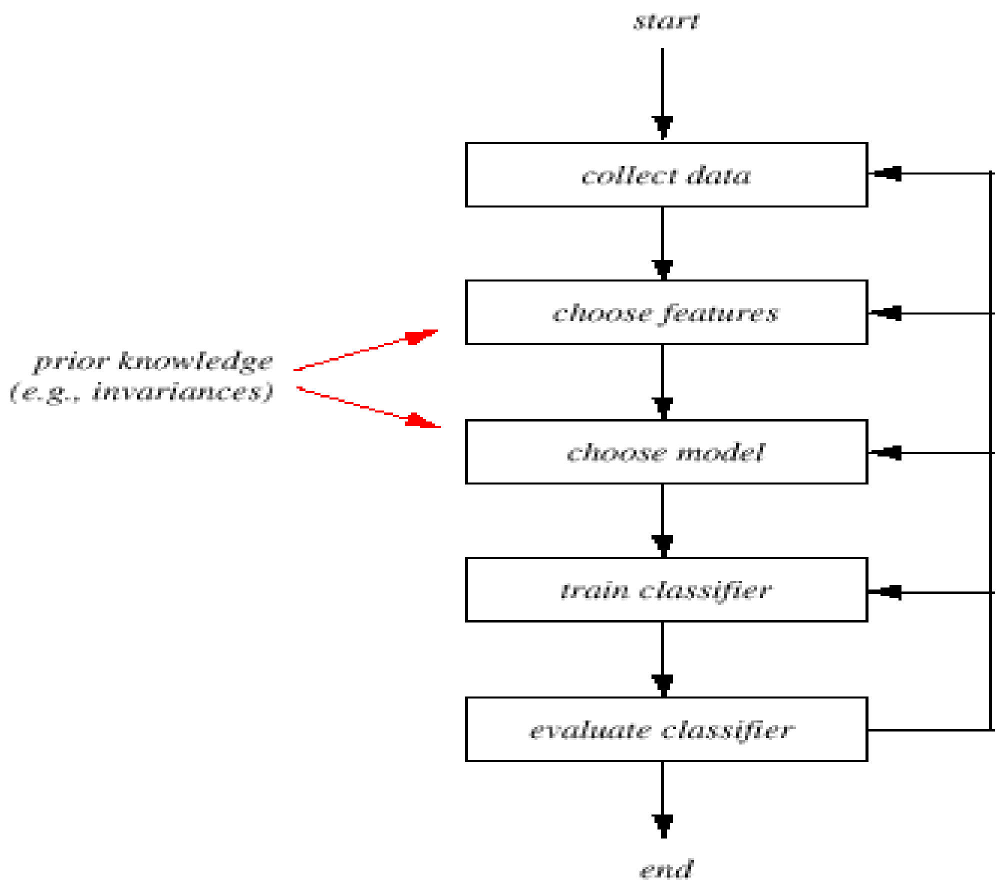


- The simplest and most common weapon against overfitting is the **test/train split**
- Train: use this part of the data for training
- Test: do **not** train on this part of the data, since we know the right answers we can use this to test how well our model works on data it hasn't seen yet

https://scikit-learn.org/stable/modules/cross_validation.html#cross-validation

The Design Cycle

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



Data Collection

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

Feature Choice

Curse of Dimensionality – having thousands of features can be bad:

- picture
- audio file
- etc

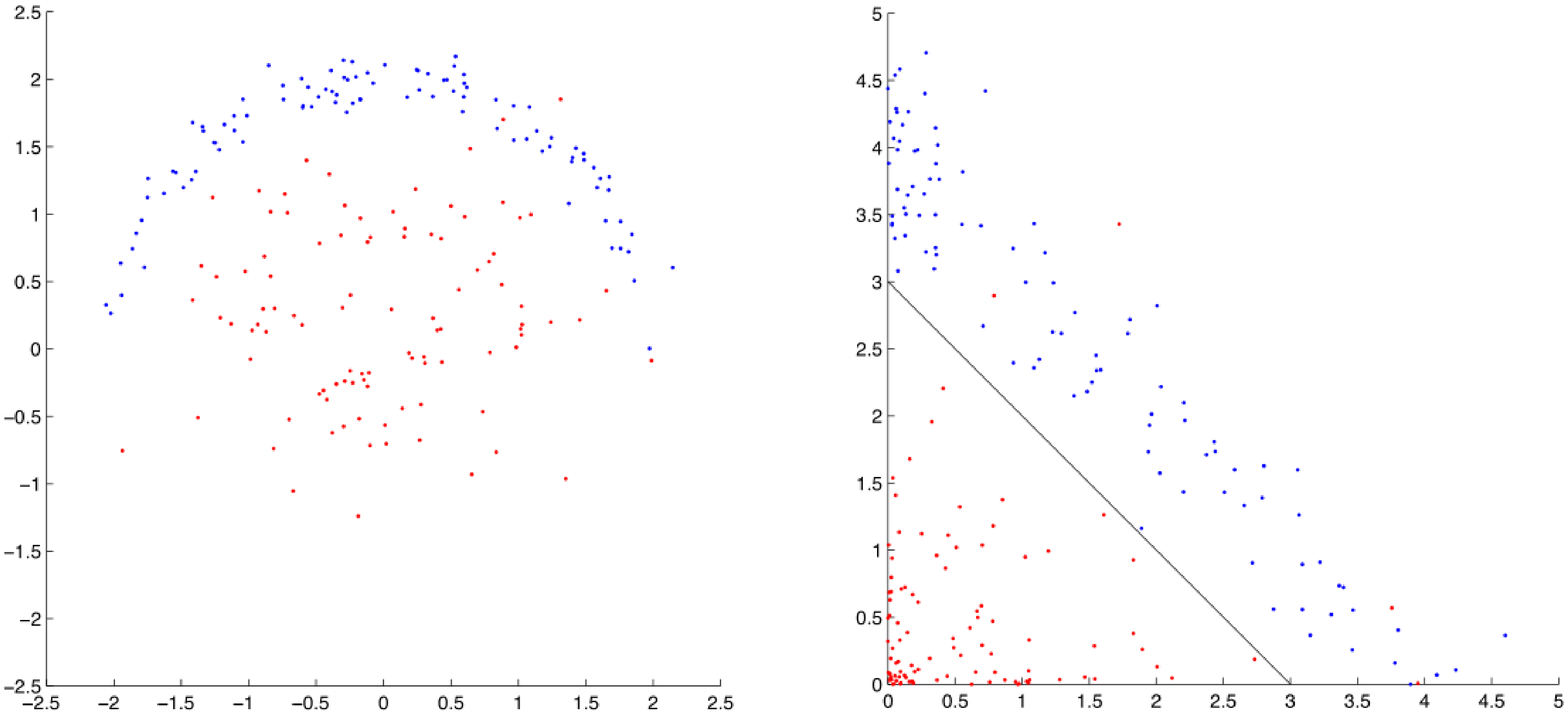
Symmetries – Features should have the properties I want. Should they depend on:

- scale?
- rotation?
- noise?

Model Choice

- Use a classifier with selected features on the training data
- Evaluate - Can I accurately sort the fish?
- Unsatisfied with the performance?
  - Tune parameters
  - Try different model
  - Change features
  - Give up and become poet

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.

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.

1. Supervised learning

- 1.1. Linear Models
- 1.2. Linear and Quadratic Discriminant Analysis
- 1.3. Kernel ridge regression
- 1.4. Support Vector Machines
- 1.5. Stochastic Gradient Descent
- 1.6. Nearest Neighbors
- 1.7. Gaussian Processes
- 1.8. Cross decomposition
- 1.9. Naive Bayes
- 1.10. Decision Trees
- 1.11. Ensemble methods
- 1.12. Multiclass and multioutput algorithms
- 1.13. Feature selection
- 1.14. Semi-supervised learning
- 1.15. Isotonic regression
- 1.16. Probability calibration
- 1.17. Neural network models (supervised)

https://scikit-learn.org/stable/user_guide.html

Example - 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.

Summary

- Supervised Learning – learning from training data
- Classification – categorize samples (as opposed to *regression*)
- Features – representation of sample
- Model – specific algorithm/method to use. Needs to learn from training data and then predict categories of samples
- Training – using pre-classified samples to fit model