Machine Learning

Introduction: Machine Learning

- A branch of **artificial intelligence**,
  - concerned with the design and development of algorithms that allow computers to evolve behaviors based on empirical data

- As intelligence requires knowledge, it is necessary for the computers to acquire knowledge

- **Arthur Samuel**, a pioneer in the field of artificial intelligence and computer gaming, coined the term **“Machine Learning”**.
  - He defined machine learning as – **“Field of study that gives computers the capability to learn without being explicitly programmed”**

- Machine Learning(ML) can be explained as automating and improving the learning process of computers based on their experiences without being actually programmed i.e. without any human assistance
  - The process starts with feeding a good quality data
  - Training machines(computers) by building machine learning models using the data and different algorithms.

- **Traditional Programming :** We feed in DATA (Input) + PROGRAM (logic), run it on machine and get output.
- **Machine Learning :** We feed in DATA(Input) + Output, run it on machine during training and the machine creates its own program(logic), which can be evaluated while testing.

- **Example: Training of students during exam.**

While preparing for the exams students don’t actually cram the subject but try to learn it with complete understanding. Before examination, they feed their machine(brain) with good amount of high quality data (questions and answers from different books or teachers notes or online video lectures). Actually, they are training their brain with input as well as output i.e. what kind of approach or logic do they have to solve different kind of questions. Each time they solve practice test papers, and find the performance (accuracy /score) by comparing answers with answer key given, Gradually, the performance keeps on increasing, gaining more confidence with the adopted approach. That’s how actually models are built, train machine with data (both inputs and outputs are given to model) and when the time comes test on data (with input only) and achieve our model scores by comparing its answer with actual output which have not been fed while training. Researchers are working with assiduous efforts to improve algorithms, techniques so that these models perform even much better

- A computer is said to be learning from **Experiences** with respect to some class of **Tasks**, if its performance in a given Task improves with the Experience.
- A computer program is said to learn from experience **E** with respect to some class of tasks **T** and performance measure **P**, if its performance at tasks in **T**, as measured by **P**, improves with experience **E**

- **Example:** playing checkers.
  E = the experience of playing many games of checkers
  T = the task of playing checkers.
  P = the probability that the program will win the next game
- In general, any machine learning problem can be assigned to one of two broad classifications:
  Supervised learning and Unsupervised learning.

How things work in reality

- Online Shopping
  - There are millions of users with unlimited range of interests with respect to brands, colors, price range and many more.
  - While online shopping, buyers tend to search for a number of products.
  - Now, searching a product frequently will make buyer’s Facebook, web pages, search engine or that online store start recommending or showing offers on that particular product.
  - There is no one sitting over there to code such task for each and every user, all this task is completely automatic. Here, ML plays its role.
  - Researchers, data scientists, machine learners build models on machine using good quality and huge amount of data and now their machine is automatically performing and even improving with more and more experience and time.

Applications:

- Medical Diagnosis
  - Researchers and scientists have prepared models to train machines for **detecting cancer** just by looking at slide – cell images.
  - Diagnose a disease

  Input : Symptoms, Lab measurements, test results

  Output: One set of possible diseases
- Computer Vision:
  - Identify “what” object appear in an image
  - Convert handwritten digits to characters
  - Detect “where” objects appear in an image

- Robot Control
  - Autonomous mobile robots that learn to navigate from their own experience
- Natural Language Processing
  - Detect where entries are mentioned in NL
  - Detect what facts are expressed in NL
  - Detect if a product/movie review is positive/negative or neutral
- Speech Recognition
- Machine Translation
- Financial Application
- Business Intelligence

https://www.youtube.com/watch?v=ahRcGObyEZo

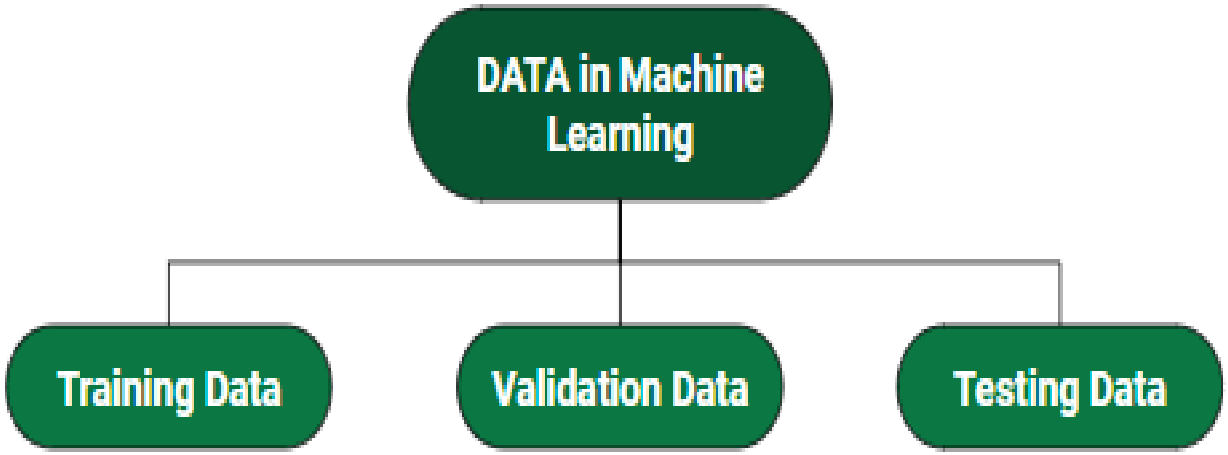
Introduction to Data in Machine Learning

DATA

- It can be any unprocessed fact, value, text, sound or picture that is not being interpreted and analyzed.
- Without data, we can’t train any model and all modern research and automation will go vain.
- Big Enterprises are spending loads of money just to gather as much certain data as possible.
- **Example:** Why did Facebook acquire WhatsApp by paying a huge price of $19 billion?
  The answer is very simple and logical – it is to have access to the users’ information that Facebook may not have but WhatsApp will have. This information of their users is of paramount importance to Facebook as it will facilitate the task of improvement in their services.

INFORMATION : Data that has been interpreted and manipulated and has now some meaningful inference for the users.

KNOWLEDGE : Combination of inferred information, experiences, learning and insights. Results in awareness or concept building for an individual or organization.



- **Training Data:** The part of data we use to train our model. This is the data which your model actually sees(both input and output) and learn from.

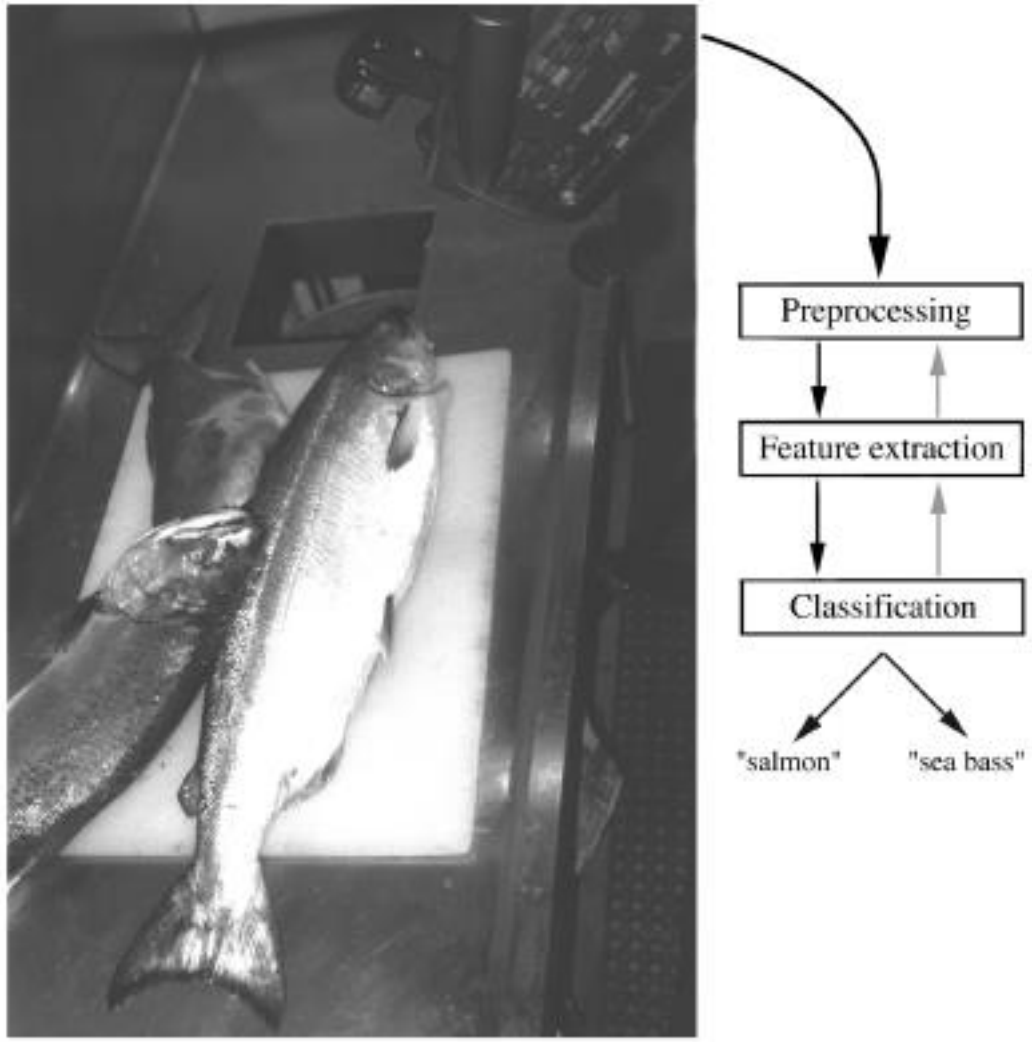
- **Validation Data:**
  - The part of data which is used to do a frequent evaluation of model, fit on training dataset.
  - This data plays it’s part when the model is actually training.
- **Testing Data:**
  - Once the model is completely trained, testing data provides the unbiased evaluation.
  - When we feed in the inputs of Testing data, the model will predict some values (without seeing actual output).
  - After prediction, we evaluate the model by comparing it with actual output present in the testing data.
  - This is how we evaluate and see how much our model has learned from the experiences feed in as training data, set at the time of training.

Machine Perception

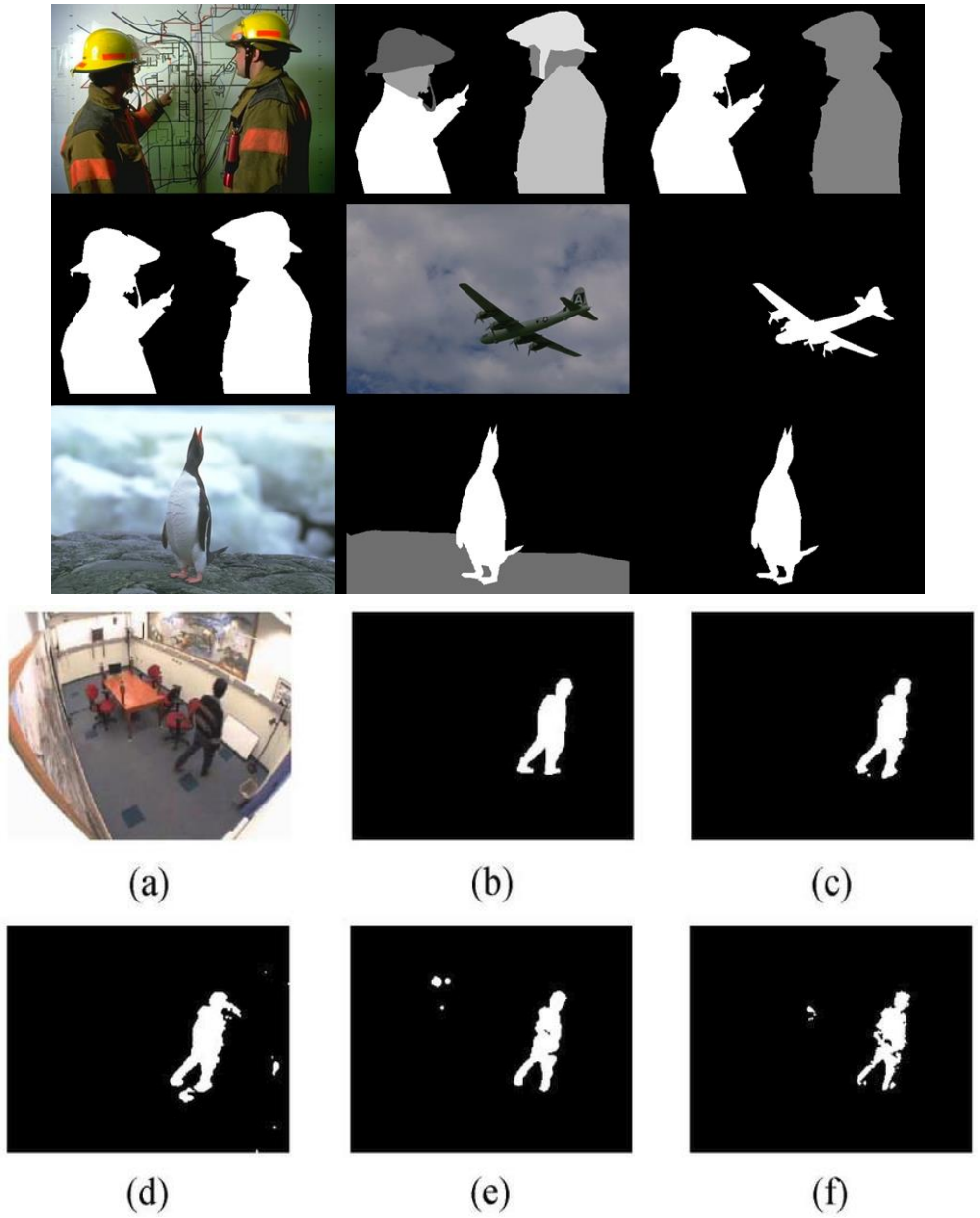
- It is natural that we should seek (attempt) to design and build machines that can recognize patterns
- From automated speech recognition, fingerprint identification, optical character recognition, DNA sequence identification and much more, it is clear that reliable, accurate pattern recognition by machine would be immensely useful
- To illustrate the complexity of some of the types of problems involved, let us consider the following imaginary and somewhat fanciful example.

- Suppose that a fish packing plant wants to automate the process of sorting incoming fish on a conveyor belt according to species.
- As a pilot project it is decided to try to separate sea bass from salmon using optical sensing
- Set up a camera, take some sample images and begin to note some physical differences between the two types of fish
  - length, lightness, width, number and shape of fins, position of the mouth, and so on
- These *features* are to be explored to use in the classifier
- Also notice noise or variations in the images — variations in lighting, position of the fish on the conveyor, even “static” due to the electronics of the camera itself

- Given that there truly are differences between the population of sea bass and that of salmon, we view them as having different *models*
  - *Models* - different descriptions, which are typically mathematical in form
- The goal and approach in pattern classification is to
  - hypothesize the class of these models,
  - process the sensed data to eliminate noise (not due to the models),
  - and for any sensed pattern choose the model that corresponds best
- Prototype system to perform this very specific task might well have the form as shown in Figure



- First the camera captures an image of the fish.
- Camera's signals are *preprocessed* to simplify subsequent operations without loosing relevant information
- Use *segmentation* operation in which the images of different fish are somehow isolated from one another and from the background



Example of segmentation

- Information from a single fish is then sent to a *feature extractor that* reduces the data by measuring certain “features” or “properties.”

- These features are then passed to a *classifier* that evaluates the evidence presented and makes a final decision as to the species.

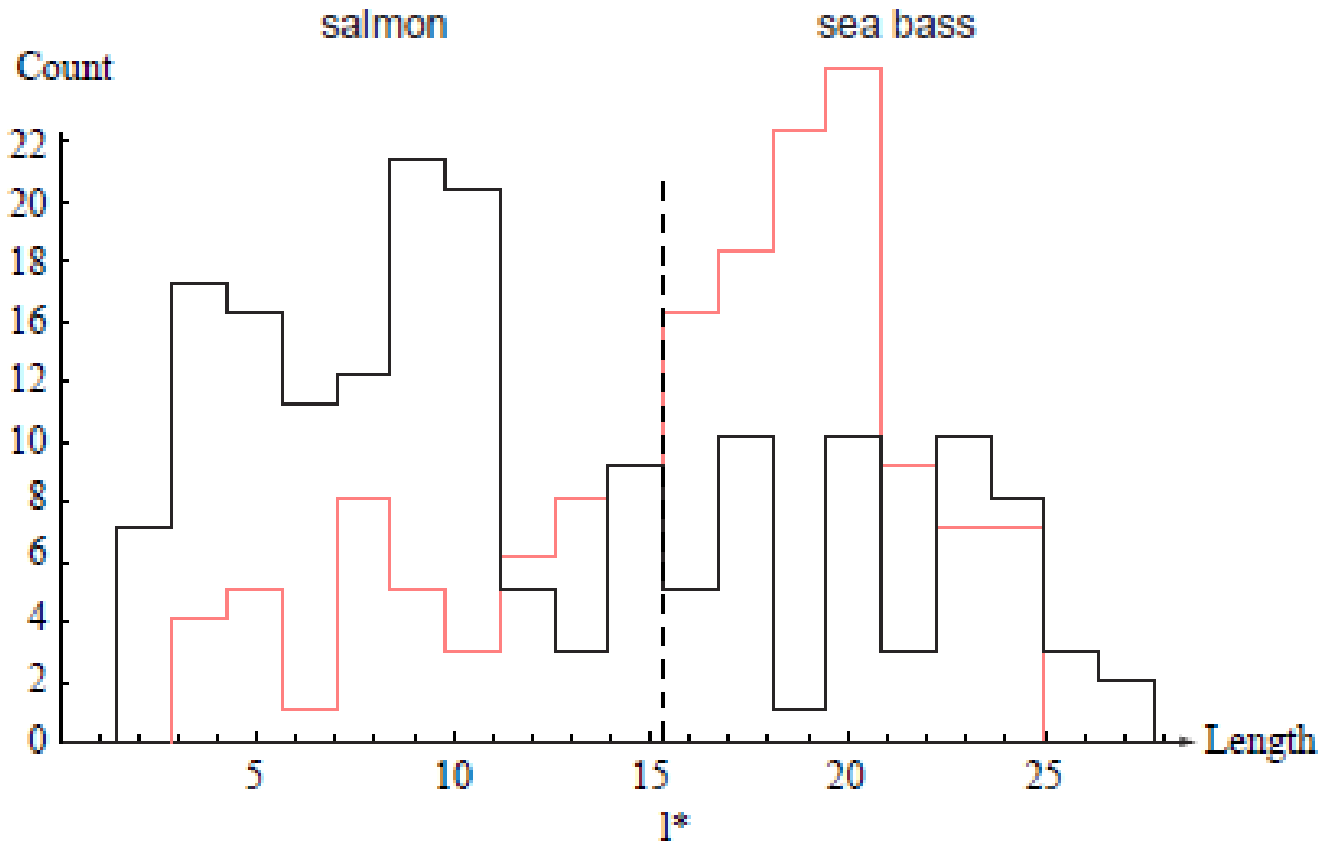
- The preprocessor might automatically adjust for average light level, or threshold the image to remove the background of the conveyor belt, and so forth

- Suppose sea bass have some typical length, and this is greater than that for salmon.
- Then length becomes an obvious *feature*,
- Classify the fish merely by seeing whether or not the length l of a fish exceeds some critical value l^*.

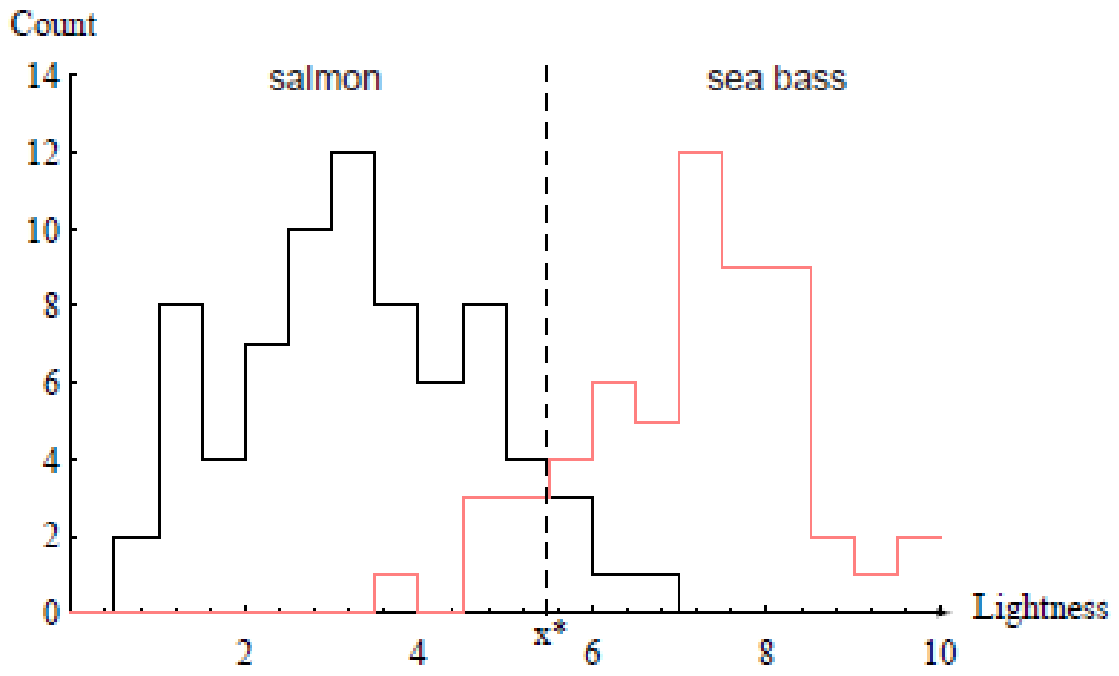
To choose l^* *(threshold)* obtain some *design* or *training samples* of the different types of fish, make length measurements and inspect the results

single criterion is quite poor

No matter how we choose l^*, we cannot reliably separate sea bass from salmon by length alone.



- Try for other features - the average lightness of the fish scales
- X^*- threshold



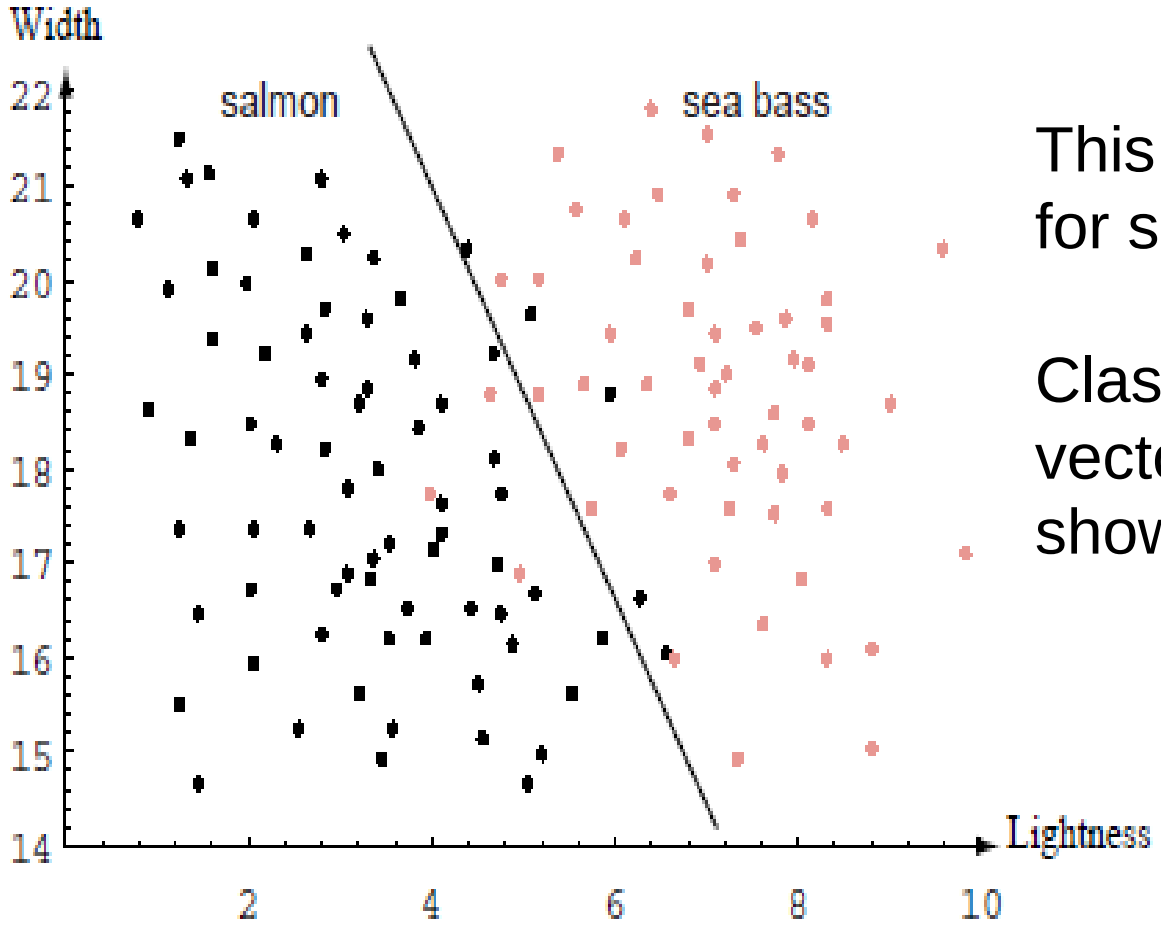
- Consequences of our actions are equally costly:
  - deciding the fish was a sea bass when in fact it was a salmon was just as undesirable as the converse

- For instance, as a fish packing company we may know that our customers easily accept occasional pieces of tasty salmon in their cans labeled “sea bass,” but they object vigorously if a piece of sea bass appears in their cans labeled “salmon.”

- In this case, then, we should move our decision boundary x^* to smaller values of lightness, thereby reducing the number of sea bass that are classified as salmon
- The more our customers object to getting sea bass with their salmon—i.e., the more costly this type of error - the lower we should set the decision threshold x^*

- Our true task is to make a decision rule (i.e., set a decision boundary) so as to minimize the cost
- This is the central task of *decision theory*
- Knowing the costs associated with decisions and choosing the optimal decision boundary x^*, resulting performance may not be satisfactory
- To improve the recognition, use more than one feature at a time
- Two features for classifying fish — the lightness x_1 and the width x_2 (by observation: sea bass are typically wider than salmon)
- feature extractor has thus reduced the image of each fish to a point or *feature vector* **x** in a two-dimensional *feature space*, where

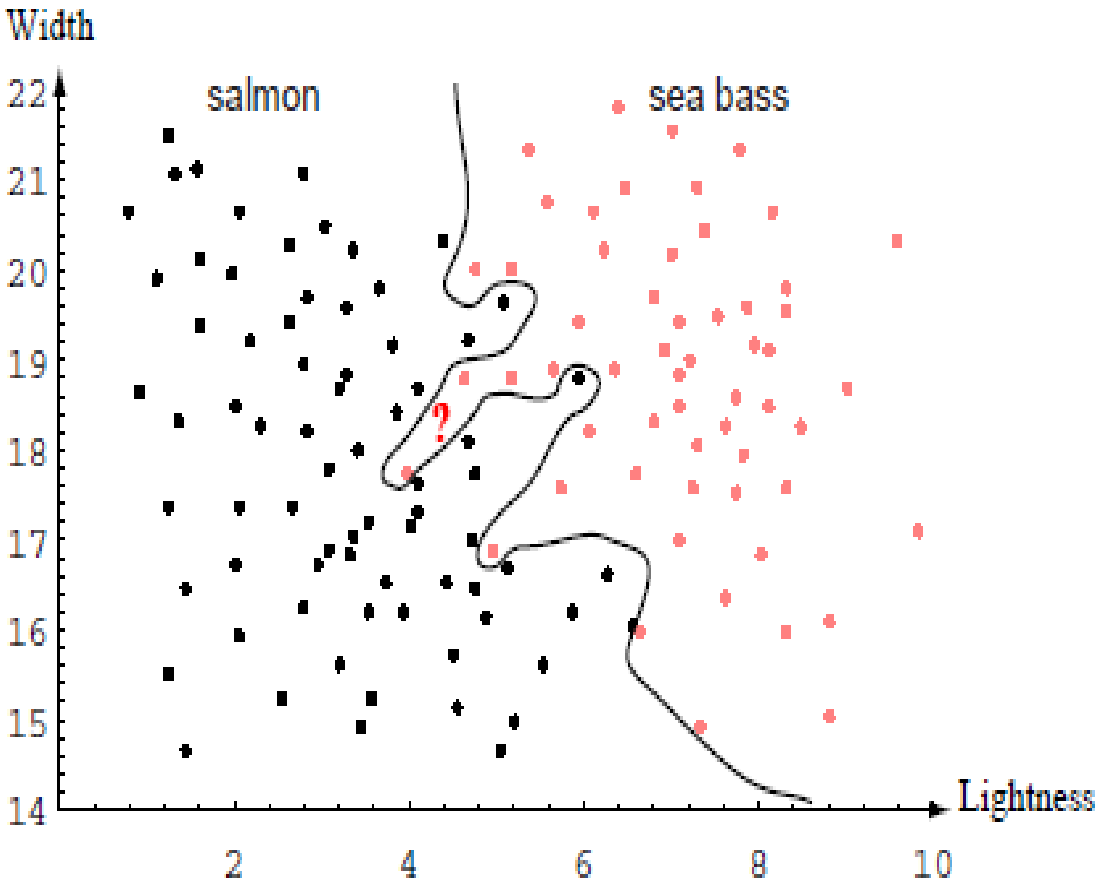
$$X = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$



This plot suggests the following rule for separating the fish:

Classify the fish as sea bass if its feature vector falls above the *decision boundary* shown, and as salmon otherwise.

- If models were extremely complicated, classifier would have a decision boundary more complex than the simple straight line

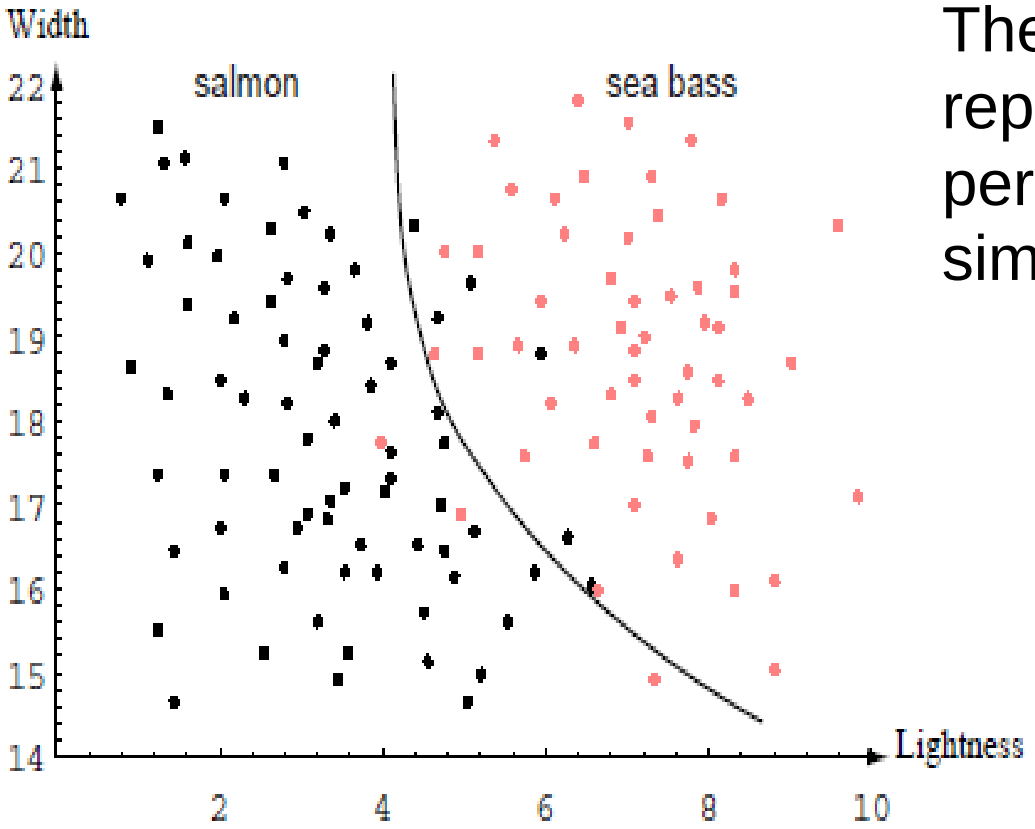


Overly complex models for the fish will lead to decision boundaries that are complicated.

Such a decision may lead to perfect classification of training samples, but it would lead to poor performance on future patterns.

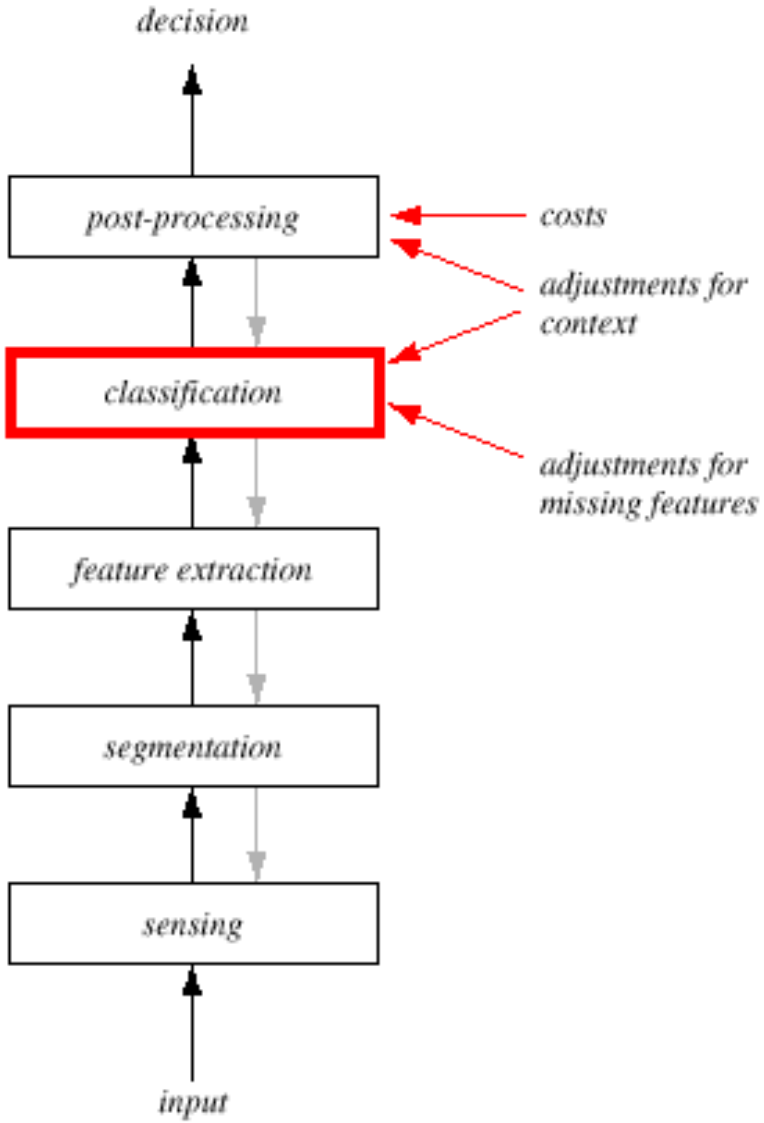
The novel test point marked **?** is evidently most likely a salmon, whereas the complex decision boundary shown leads it to be misclassified as a sea bass

- Central aim of designing a classifier is to suggest actions when presented with *novel* patterns, i.e., fish not yet seen. This is the issue of *generalization*



The decision boundary shown might represent the optimal tradeoff between performance on the training set and simplicity of classifier

Pattern Recognition System



Sensing

- Use of a transducer (camera or microphone)
- PR system depends on the bandwidth, the resolution sensitivity distortion of the transducer

Segmentation and grouping

- Patterns should be well separated and should not overlap

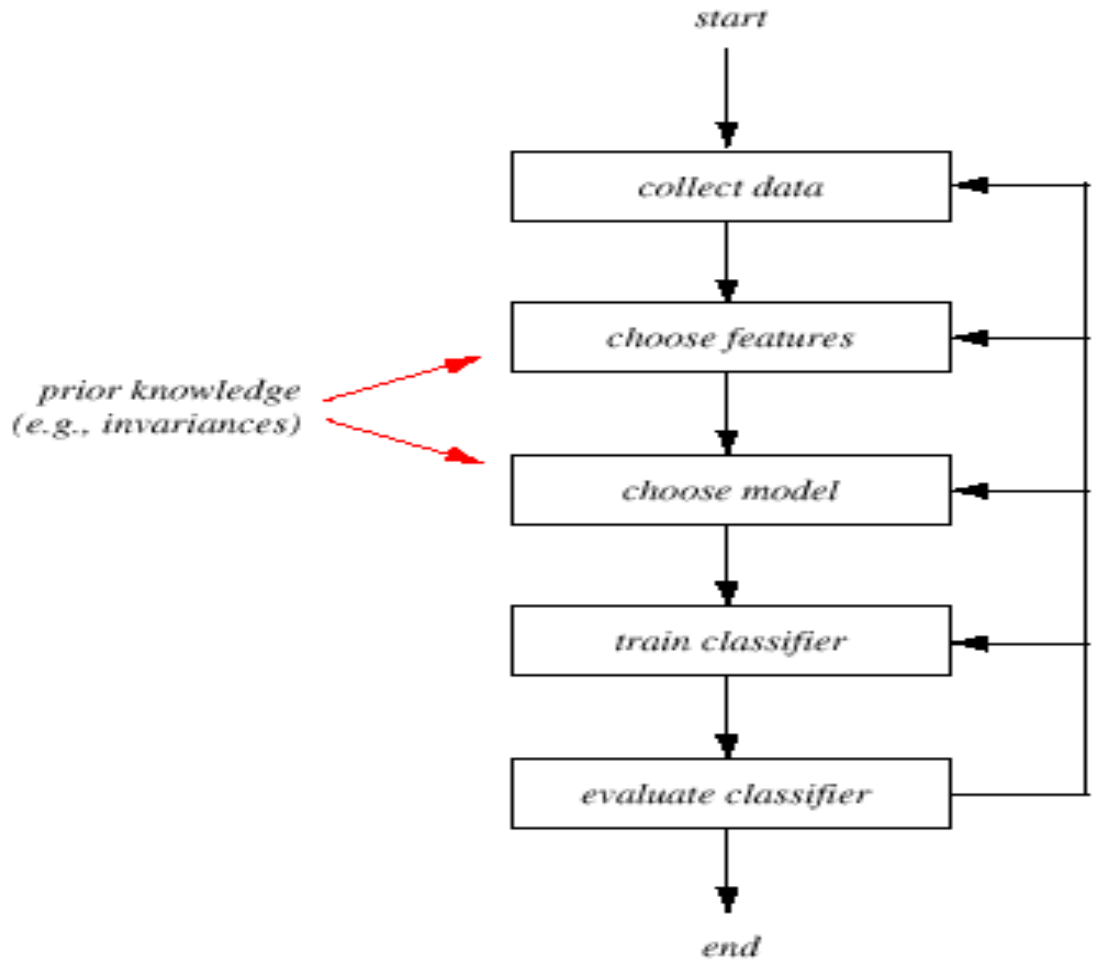
- Feature extraction
  - Discriminative features
  - Invariant features with respect to translation, rotation and scale.

- Classification
  - Use a feature vector provided by a feature extractor to assign the object to a category

- Post Processing
  - Exploit *context* input dependent information other than from the target pattern itself to improve performance

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 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?)