# Machine Learning

# What is Machine Learning?

*Aspect of AI: creates knowledge*

Definition:

“changes in [a] system that ... enable [it] to do the same task or tasks drawn from the same population more efficiently and more effectively <u>the next time</u>." (Simon 1983)

Two ways that a system can improve:

1. By acquiring new knowledge
2. By adapting its behavior

# What is Learning?

- Herbert Simon: “Learning is any process by which a system improves performance from experience.”
- the task?
  - Classification
  - Categorization/clustering
  - Problem solving / planning / control
  - Prediction
  - others

# Why Study Machine Learning? Developing Better Computing Systems

- Develop systems that are too difficult/expensive to construct manually
- Develop systems that can automatically adapt and customize themselves to individual users.
- Discover new knowledge from large databases (***data mining***).
  - Market basket analysis
  - Medical text mining

# Human vs machine learning

- Cognitive science vs computational science
- Evolution vs machine learning
  - Adaptation vs learning

# Adaptive vs machine learning

- An adaptive system is a set of interacting or interdependent entities that together are able to respond to environmental changes or changes in the interacting parts.
- Feedback loops represent a key feature of adaptive systems, allowing the response to changes
- Some artificial systems can be adaptive as well; eg, robots utilize feedback loops to sense new conditions in their environment and adapt accordingly.

# Types of Learning

- Induction vs deduction
- Rote learning (memorization)
- Advice or instructional learning
- *Learning by example or practice*
  - *Most popular; many applications*
- Learning by analogy; transfer learning
- Discovery learning
- Others?

# Training

- the acquisition of knowledge, skills, and competencies as a result of the teaching of vocational or practical skills and knowledge that relate to specific useful competencies (wikipedia).
- requires scenarios(cases) or examples (data)

# Types of training experience

- Direct or indirect
- With a teacher(supervised) or without a teacher(unsupervised)

# Types of training

- Supervised learning: uses a series of labelled examples with direct feedback
- Reinforcement learning: indirect feedback, after many examples
- Unsupervised/clustering learning: no feedback
- Semisupervised

# Types of testing

- Evaluate performance by testing on data NOT used for testing (both should be randomly sampled)
- Cross validation methods for small data sets
- The more (relevant) data the better.

# Testing

- How well the learned system work?
- Generalization
  - Performance on unseen or unknown scenarios or data
  - weak vs robust performance

# Which of these things is NOT like the others?

# Which of these things is like the others? And how?

# Usual ML stages

- Hypothesis, data
- Training or learning
- Testing or generalization

# Why is machine learning necessary?

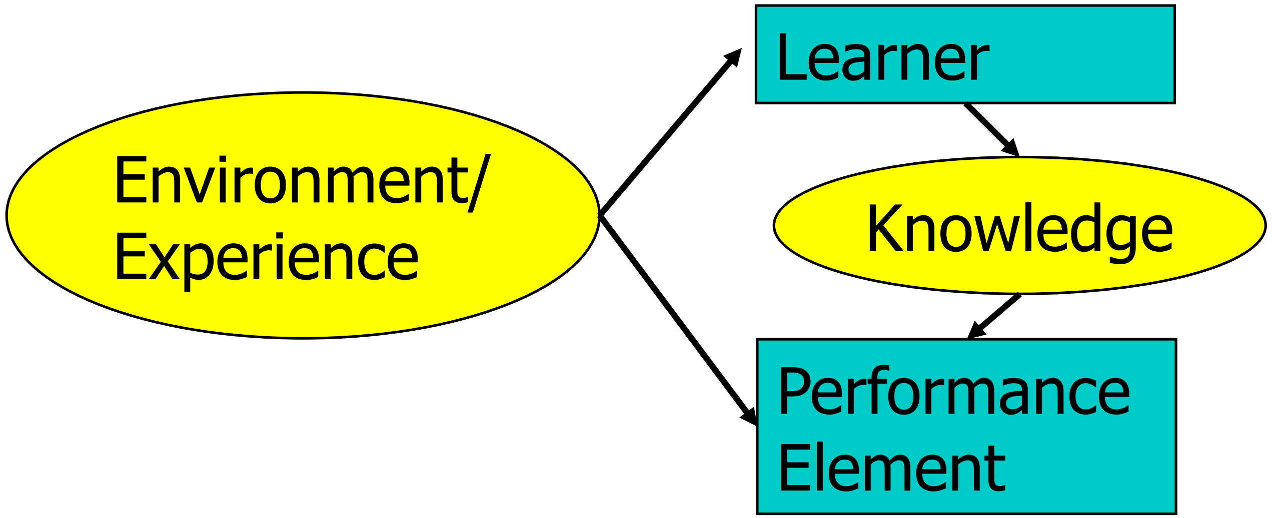
- *learning is a hallmark of intelligence*; a system that cannot learn is not intelligent.
- without learning, everything is new; a system that cannot learn is not efficient because it rederives each solution and repeatedly makes the same mistakes.

**Why is learning possible?**

Because there are regularities in the world.

# Designing a Learning System

- Choose the training experience
- Choose exactly what is too be learned, i.e. the ***target function***.
- Choose how to represent the target function.
- Choose a learning algorithm to infer the target function from the experience.



# Training Experience

- Direct experience: Given sample input and output pairs for a useful target function.
  - Checker boards labeled with the correct move, e.g. extracted from record of expert play
- Indirect experience: Given feedback which is ***not*** direct I/O pairs for a useful target function.
  - Potentially arbitrary sequences of game moves and their final game results.
- Credit/Blame Assignment Problem: How to assign credit blame to individual moves given only indirect feedback?

# Training vs. Test Distribution

- Generally assume that the training and test examples are independently drawn from the same overall distribution of data.
  - IID: Independently and identically distributed
- If examples are not independent, requires ***collective classification***.
- If test distribution is different, requires ***transfer learning***.

# Choosing a Target Function

- What function is to be learned and how will it be used by the performance system?
- For checkers, assume we are given a function for generating the legal moves for a given board position and want to decide the best move.
  - Could learn a function:

    ChooseMove(board, legal-moves) → best-move
  - Or could learn an ***evaluation function***, *V*(board) → $\mathcal{R}$, that gives each board position a score for how favorable it is. *V* can be used to pick a move by applying each legal move, scoring the resulting board position, and choosing the move that results in the highest scoring board position.

# Ideal Definition of *V*(*b*)

- If *b* is a final winning board, then *V*(*b*) = 100
- If *b* is a final losing board, then *V*(*b*) = –100
- If *b* is a final draw board, then *V*(*b*) = 0
- Otherwise, then *V*(*b*) = *V*(*b*´), where *b*´ is the highest scoring final board position that is achieved starting from *b* and playing optimally until the end of the game (assuming the opponent plays optimally as well).
  - Can be computed using complete mini-max search of the finite game tree.

# Approximating *V*(*b*)

- Computing *V*(*b*) is intractable since it involves searching the complete exponential game tree.
- Therefore, this definition is said to be ***non-operational***.
- An ***operational*** definition can be computed in reasonable (polynomial) time.
- Need to learn an operational *approximation* to the ideal evaluation function.

# Representing the Target Function

- Target function can be represented in many ways: lookup table, symbolic rules, numerical function, neural network.
- There is a trade-off between the expressiveness of a representation and the ease of learning.
- The more expressive a representation, the better it will be at approximating an arbitrary function; however, the more examples will be needed to learn an accurate function.

# Linear Function for Representing *V*(*b*)

- In checkers, use a linear approximation of the evaluation function.

$$\widehat{V}(b) = w_0 + w_1 \cdot bp(b) + w_2 \cdot rp(b) + w_3 \cdot bk(b) + w_4 \cdot rk(b) + w_5 \cdot bt(b) + w_6 \cdot rt(b)$$

  - *bp*(*b*): number of black pieces on board b
  - *rp*(*b*): number of red pieces on board b
  - *bk*(*b*): number of black kings on board b
  - *rk*(*b*): number of red kings on board b
  - *bt*(*b*): number of black pieces threatened (i.e. which can be immediately taken by red on its next turn)
  - *rt*(*b*): number of red pieces threatened

# Obtaining Training Values

- Direct supervision may be available for the target function.
  - < <*bp*=3,*rp*=0,*bk*=1,*rk*=0,*bt*=0,*rt*=0>, 100> (win for black)
- With indirect feedback, training values can be estimated using ***temporal difference learning*** (used in ***reinforcement learning*** where supervision is ***delayed reward***).

# Temporal Difference Learning

- Estimate training values for intermediate (non-terminal) board positions by the estimated value of their successor in an actual game trace.

$$V_{train}(b) = \hat{V}(\text{successor}(b))$$

  where successor(*b*) is the next board position where it is the program's move in actual play.
- Values towards the end of the game are initially more accurate and continued training slowly "backs up" accurate values to earlier board positions.

# Learning Algorithm

- Uses training values for the target function to induce a hypothesized definition that fits these examples and hopefully generalizes to unseen examples.
- In statistics, learning to approximate a continuous function is called ***regression***.
- Attempts to minimize some measure of error (***loss function***) such as ***mean squared error***:

$$E = \frac{\sum_{b \in B} [V_{train}(b) - \hat{V}(b)]^2}{|B|}$$

# Least Mean Squares (LMS) Algorithm

- A gradient descent algorithm that incrementally updates the weights of a linear function in an attempt to minimize the mean squared error

Until weights converge :

For each training example *b* do :

1) Compute the absolute error :

$$error(b) = V_{train}(b) - \widehat{V}(b)$$

2) For each board feature, $f_i$, update its weight, $w_i$ :

$$w_i = w_i + c \cdot f_i \cdot error(b)$$

for some small constant (learning rate) c

# LMS Discussion

- Intuitively, LMS executes the following rules:
  - If the output for an example is correct, make no change.
  - If the output is too high, lower the weights proportional to the values of their corresponding features, so the overall output decreases
  - If the output is too low, increase the weights proportional to the values of their corresponding features, so the overall output increases.
- Under the proper weak assumptions, LMS can be proven to eventetually converge to a set of weights that minimizes the mean squared error.

# Lessons Learned about Learning

- using direct or indirect experience to approximate a chosen target function.
- Function approximation can be viewed as a search through a space of hypotheses for one that best fits a set of training data.
- Different learning methods assume different hypothesis spaces and/or employ different search techniques.

# Evaluation of Learning Systems

- Experimental
  - Conduct controlled cross-validation experiments to compare various methods on a variety of benchmark datasets.
  - Gather data on their performance, e.g. test accuracy, training-time, testing-time.
  - Analyze differences for statistical significance.
- Theoretical
  - Analyze algorithms mathematically and prove theorems about their:
    - Computational complexity
    - Ability to fit training data
    - Sample complexity (number of training examples needed to learn an accurate function)

# History of Machine Learning

- 1950s
  - Samuel’s checker player
  - Selfridge’s Pandemonium
- 1960s:
  - Neural networks: Perceptron
  - Pattern recognition
  - Learning in the limit theory
  - Minsky and Papert prove limitations of Perceptron
- 1970s:
  - Symbolic concept induction
  - Winston’s arch learner
  - Expert systems and the knowledge acquisition bottleneck
  - Quinlan’s ID3
  - Michalski’s AQ and soybean diagnosis
  - Scientific discovery with BACON
  - Mathematical discovery with AM

# History of Machine Learning (cont.)

- 1980s:
  - Advanced decision tree and rule learning
  - Explanation-based Learning (EBL)
  - Learning and planning and problem solving
  - Utility problem
  - Analogy
  - Cognitive architectures
  - Resurgence of neural networks (connectionism, backpropagation)
  - Valiant's PAC Learning Theory
  - Focus on experimental methodology
- 1990s
  - Data mining
  - Adaptive software agents and web applications
  - Text learning
  - Reinforcement learning (RL)
  - Inductive Logic Programming (ILP)
  - Ensembles: Bagging, Boosting, and Stacking
  - Bayes Net learning

# History of Machine Learning (cont.)

- 2000s
  - Support vector machines
  - Kernel methods
  - Graphical models
  - Statistical relational learning
  - Transfer learning
  - Sequence labeling
  - Collective classification and structured outputs
  - Computer Systems Applications
    - Compilers
    - Debugging
    - Graphics
    - Security (intrusion, virus, and worm detection)
  - Email management
  - Personalized assistants that learn
  - Learning in robotics and vision

# Supervised Learning Classification

- Example: Cancer diagnosis

| Patient ID | # of Tumors | Avg Area | Avg Density | Diagnosis |
|---|---|---|---|---|
| 1 | 5 | 20 | 118 | Malignant |
| 2 | 3 | 15 | 130 | Benign |
| 3 | 7 | 10 | 52 | Benign |
| 4 | 2 | 30 | 100 | Malignant |

Training Set

- Use this training set to learn how to classify patients where diagnosis is not known:

| Patient ID | # of Tumors | Avg Area | Avg Density | Diagnosis |
|---|---|---|---|---|
| 101 | 4 | 16 | 95 | ? |
| 102 | 9 | 22 | 125 | ? |
| 103 | 1 | 14 | 80 | ? |

Test Set

Input Data Classification

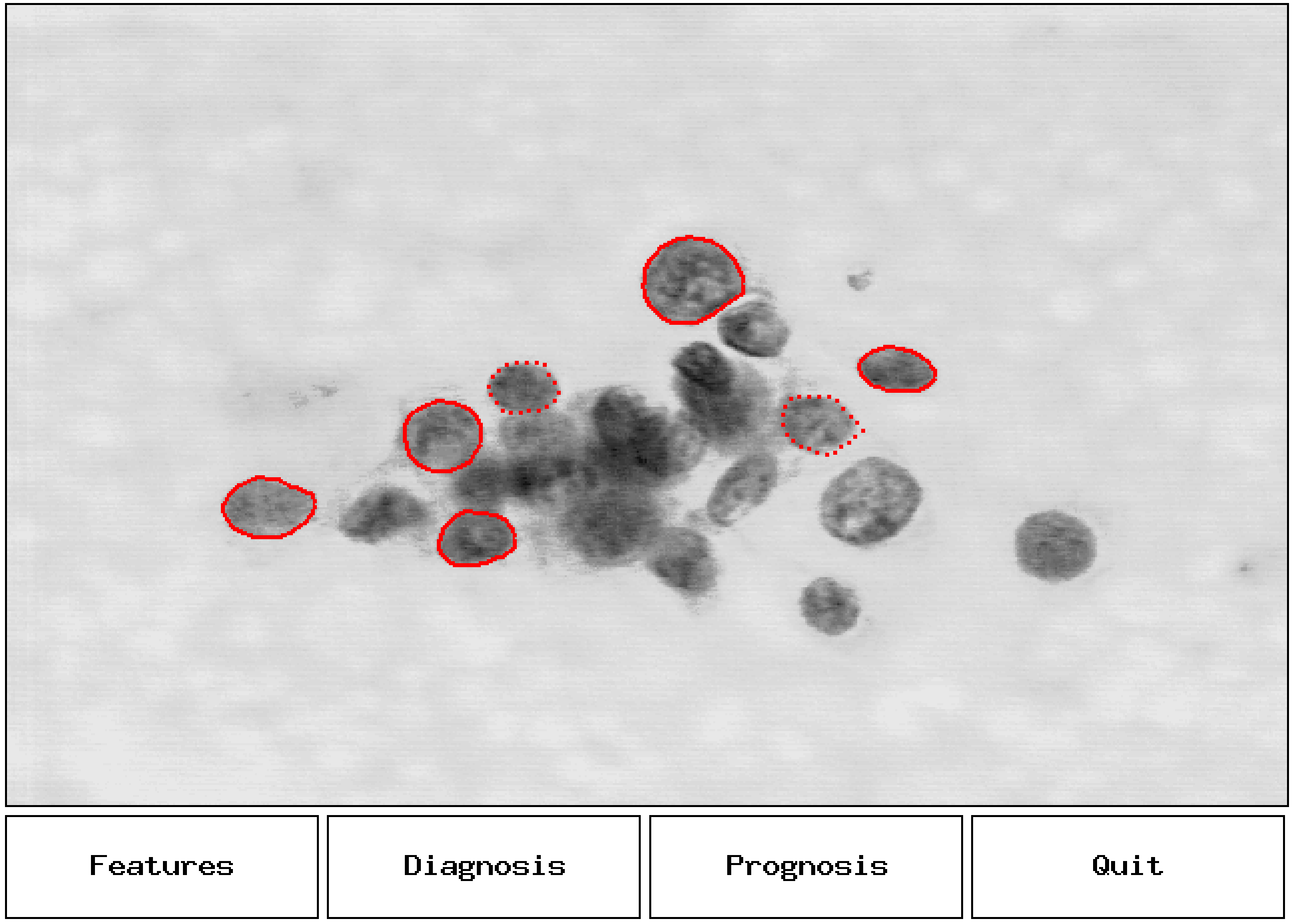
- The input data is often easily obtained, whereas the classification is not.

# Classification Problem

- Goal: Use training set + some learning method to produce a predictive model.
- Use this predictive model to classify new data.
- Sample applications:

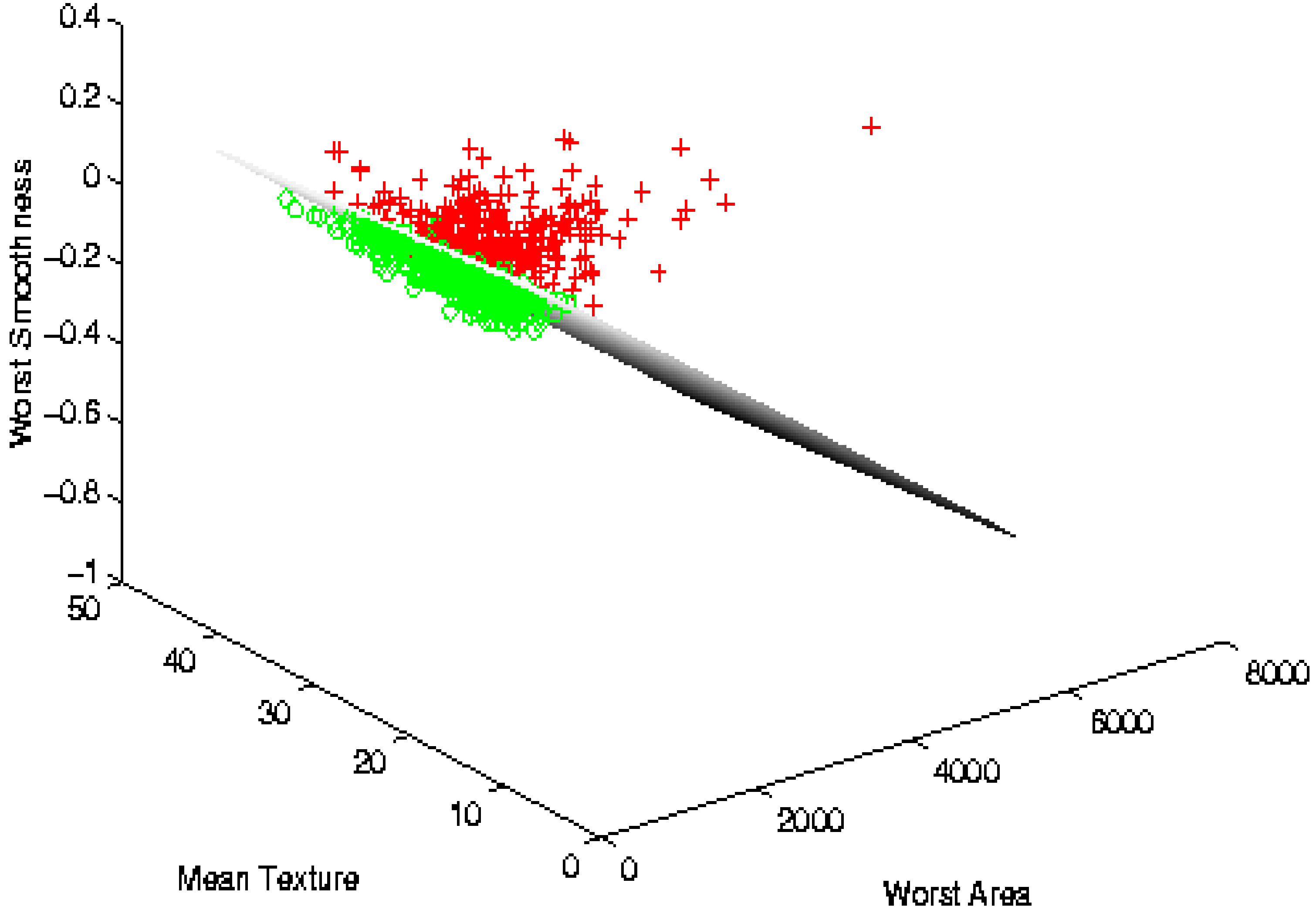
| Application | Input Data | Classification |
|---|---|---|
| Medical Diagnosis | Noninvasive tests | Results from invasive measurements |
| Optical Character Recognition | Scanned bitmaps | Letter A-Z |
| Protein Folding | Amino acid construction | Protein shape (helices, loops, sheets) |
| Research Paper Acceptance | Words in paper title | Paper accepted or rejected |

# Application: Breast Cancer Diagnosis



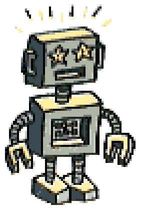

*Research by Mangasarian,Street, Wolberg*

# Breast Cancer Diagnosis Separation




*Research by Mangasarian,Street, Wolberg*

# Robotics and ML



Areas that robots are used:

- Industrial robots
- Military, government and space robots
- Service robots for home, healthcare, laboratory

Why are robots used?

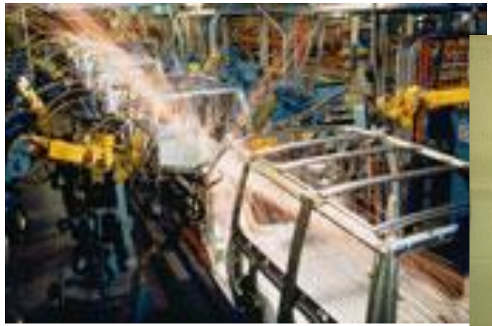
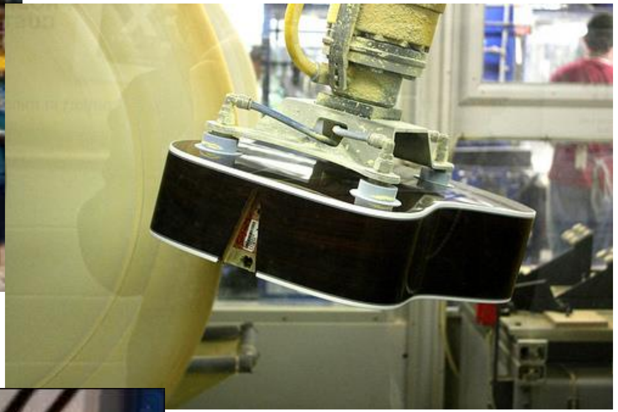
- Dangerous tasks or in hazardous environments
- Repetitive tasks
- High precision tasks or those requiring high quality
- Labor savings

Control technologies:

- Autonomous (self-controlled), tele-operated (remote control)

# Industrial Robots

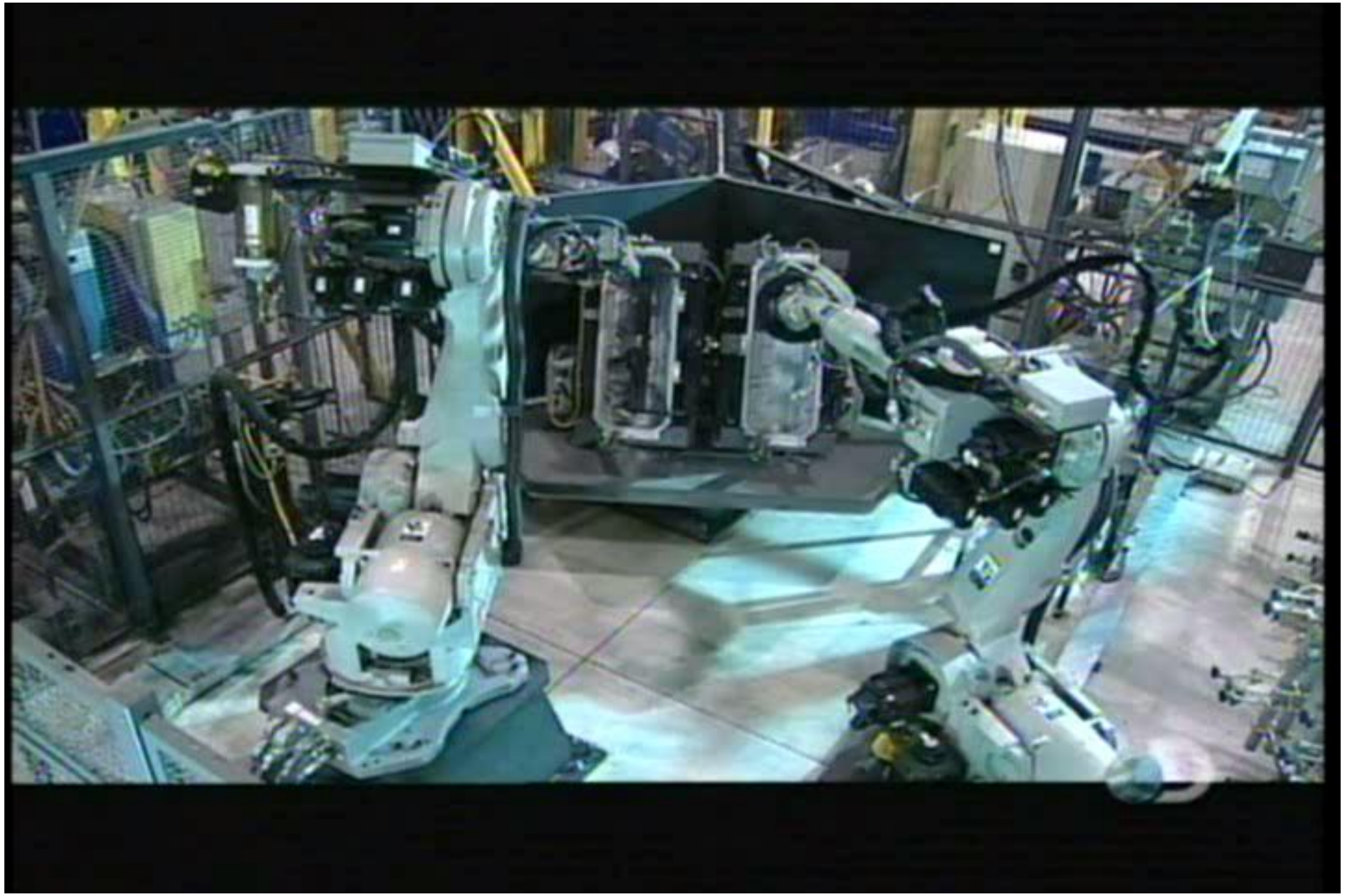
- Uses for robots in manufacturing:
  - – Welding
  - – Painting
  - – Cutting
  - – Dispensing
  - – Assembly
  - – Polishing/Finishing
  - – Material Handling
    - Packaging, Palletizing
    - Machine loading

# Industrial Robots

- Uses for robots in Industry/Manufacturing
  - Automotive:
    - Video - Welding  and handling of fuel tanks from TV show “How It’s Made” on Discovery Channel. This is a system I worked on in 2003.
  - Packaging:
    - Video - Robots in food manufacturing.

# Industrial Robots - Automotive



# Military/Government Robots

- iRobot


Remotec Andros

# Military/Government Robots

Soldiers in Afghanistan being trained how to defuse a landmine using a PackBot.

# Military Robots

- Aerial drones (UAV)

Military suit

# Space Robots

- Mars Rovers – Spirit and Opportunity
  - Autonomous navigation features with human remote control and oversight

# Service Robots

- Many uses…
  - Cleaning & Housekeeping
  - Humanitarian Demining
  - Rehabilitation
  - Inspection
  - Agriculture & Harvesting
  - Lawn Mowers
  - Surveillance
  - Mining Applications
  - Construction
  - Automatic Refilling
  - Fire Fighters
  - Search & Rescue


iRobot Roomba vacuum cleaner robot

# Medical/Healthcare Applications

DaVinci surgical robot by Intuitive Surgical.

St. Elizabeth Hospital is one of the local hospitals using this robot. You can see this robot in person during an open house (website).

Japanese health care assistant suit (HAL - Hybrid Assistive Limb)


Also… Mind-controlled wheelchair using NI LabVIEW

# Laboratory Applications

Drug discovery

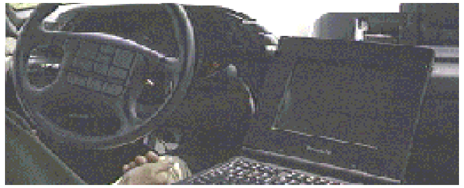
Test tube sorting

# ALVINN autonomous car

**ALVINN drives 70 mph on highways**

Drives 70 mph on a public highway
Predecessor of the Google car

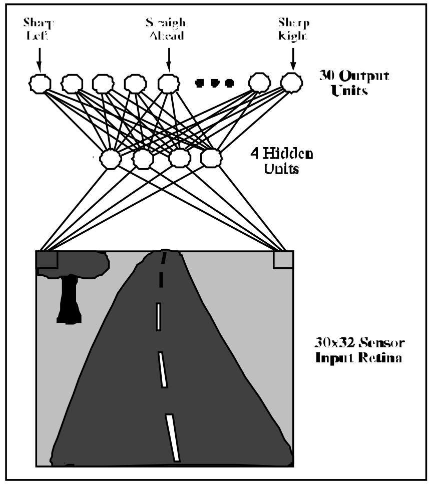
Camera image

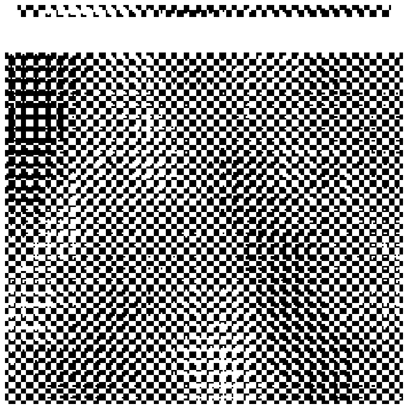


30 outputs for steering

4 hidden units

30x32 pixels as inputs



30x32 weights into one out of four hidden unit

# Learning vs Adaptation

- ”Modification of a behavioral tendency by expertise.” (Webster 1984)
- ”A learning machine, broadly defined is any device whose actions are influenced by past experiences.” (Nilsson 1965)
- ”Any change in a system that allows it to perform better the second time on repetition of the same task or on another task drawn from the same population.” (Simon 1983)
- ”An improvement in information processing ability that results from information processing activity.” (Tanimoto 1990)

# Machine Learning as

- Function approximation (mapping)
  - Regression
- Classification
- Categorization (clustering)
- Prediction
- Pattern recognition

# Working Applications of ML

- Classification of mortgages
- Predicting portfolio performance
- Electrical power control
- Chemical process control
- Character recognition
- Face recognition
- DNA classification
- Credit card fraud detection
- Cancer cell detection

# Issues in Machine Learning

- What algorithms can approximate functions well and when
  - How does the number of training examples influence accuracy
- Problem representation / feature extraction
- Intention/independent learning
- Integrating learning with systems
- What are the theoretical limits of learnability
- Transfer learning
- Continuous learning

# Measuring Performance

- Generalization accuracy
- Solution correctness
- Solution quality (length, efficiency)
- Speed of performance

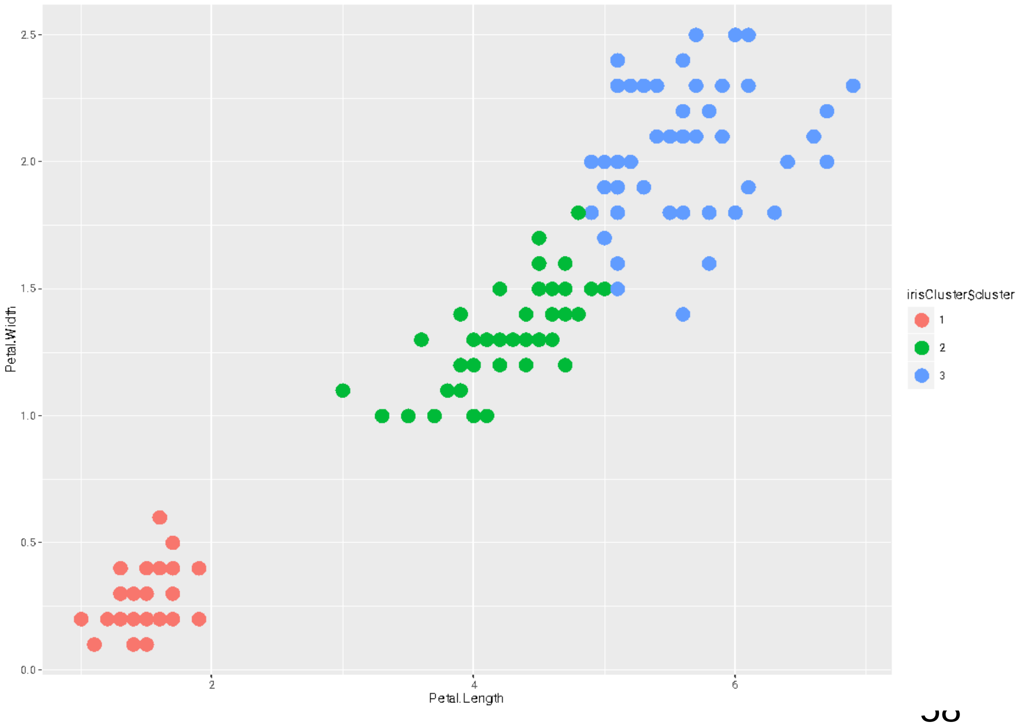
# Machine Learning versus Human Learning

- Some ML behavior can win over the performance of human experts (e.g., playing chess)
- Although ML sometimes matches human learning capabilities
- machine learning can be applied in a truly creative way similar to human
- Formal theories of ML systems exist but are often lacking (why a method succeeds or fails is not clear, explainable?)
- ML success is often attributed to manipulation of symbols or images (rather than mere numeric information)

# Supervised and Unsupervised Learning

- Unsupervised Learning
  - There are not predefined and known set of outcomes
  - Look for hidden patterns and relations in the data
  - A typical example: Clustering

| | Sepal.Length | Sepal.Width | Petal.Length | Petal.Width |
|---|---|---|---|---|
| 1 | 5.1 | 3.5 | 1.4 | 0.2 |
| 2 | 4.9 | 3.0 | 1.4 | 0.2 |
| 3 | 4.7 | 3.2 | 1.3 | 0.2 |
| 4 | 4.6 | 3.1 | 1.5 | 0.2 |
| 5 | 5.0 | 3.6 | 1.4 | 0.2 |
| 6 | 5.4 | 3.9 | 1.7 | 0.4 |
| 7 | 4.6 | 3.4 | 1.4 | 0.3 |
| 8 | 5.0 | 3.4 | 1.5 | 0.2 |
| 9 | 4.4 | 2.9 | 1.4 | 0.2 |
| 10 | 4.9 | 3.1 | 1.5 | 0.1 |



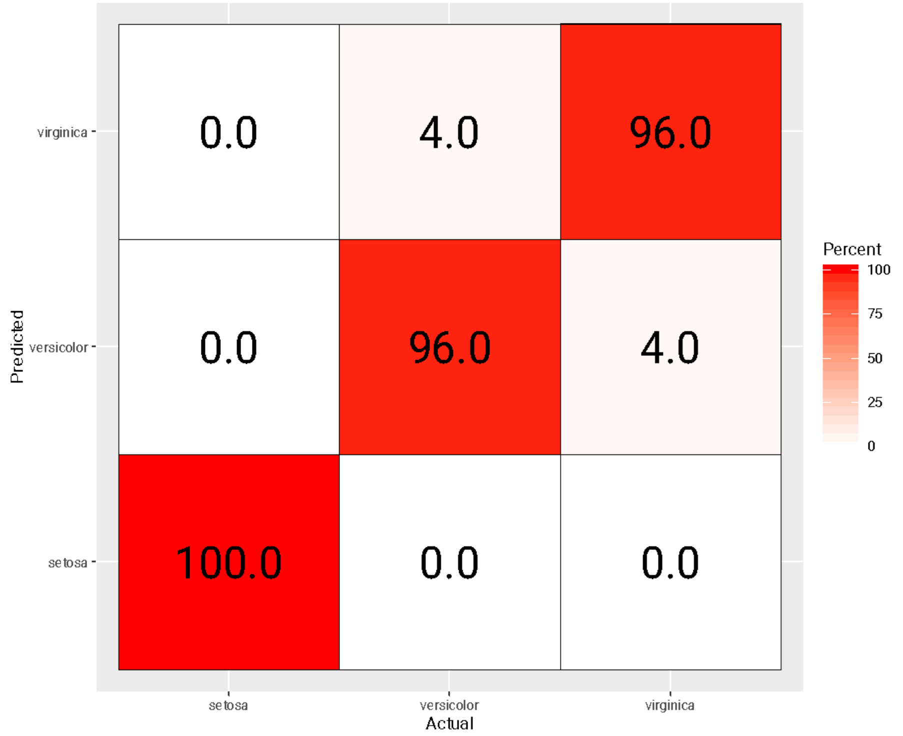
# Supervised and Unsupervised Learning

- Supervised Learning
  - For every example in the data there is always a predefined outcome
  - Models the relations between a set of descriptive features and a target (Fits data to a function)
  - 2 groups of problems:
    - Classification
    - Regression

# Supervised Learning

- Classification
  - Predicts which class a given sample of data (sample  of descriptive features) is part of (**discrete value**).

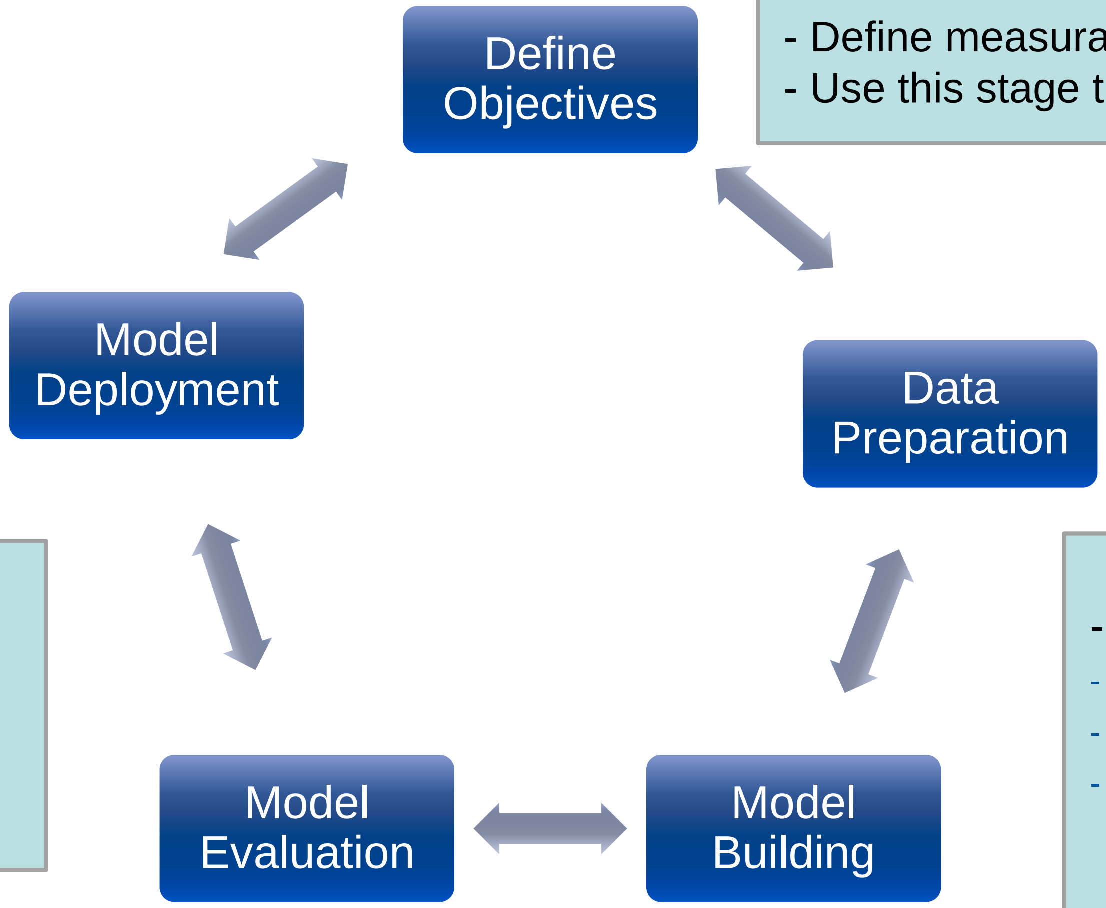
| | Sepal.Length | Sepal.Width | Petal.Length | Petal.Width | Species |
|---|---|---|---|---|---|
| 1 | 5.1 | 3.5 | 1.4 | 0.2 | setosa |
| 2 | 4.9 | 3.0 | 1.4 | 0.2 | setosa |
| 3 | 4.7 | 3.2 | 1.3 | 0.2 | setosa |
| 4 | 4.6 | 3.1 | 1.5 | 0.2 | setosa |
| 5 | 5.0 | 3.6 | 1.4 | 0.2 | setosa |
| 6 | 5.4 | 3.9 | 1.7 | 0.4 | setosa |
| 7 | 4.6 | 3.4 | 1.4 | 0.3 | setosa |
| 8 | 5.0 | 3.4 | 1.5 | 0.2 | setosa |
| 9 | 4.4 | 2.9 | 1.4 | 0.2 | setosa |
| 10 | 4.9 | 3.1 | 1.5 | 0.1 | setosa |




- Regression
  - Predicts continuous values.

# Machine Learning as a Process

- Define measurable and quantifiable goals
- Use this stage to learn about the problem



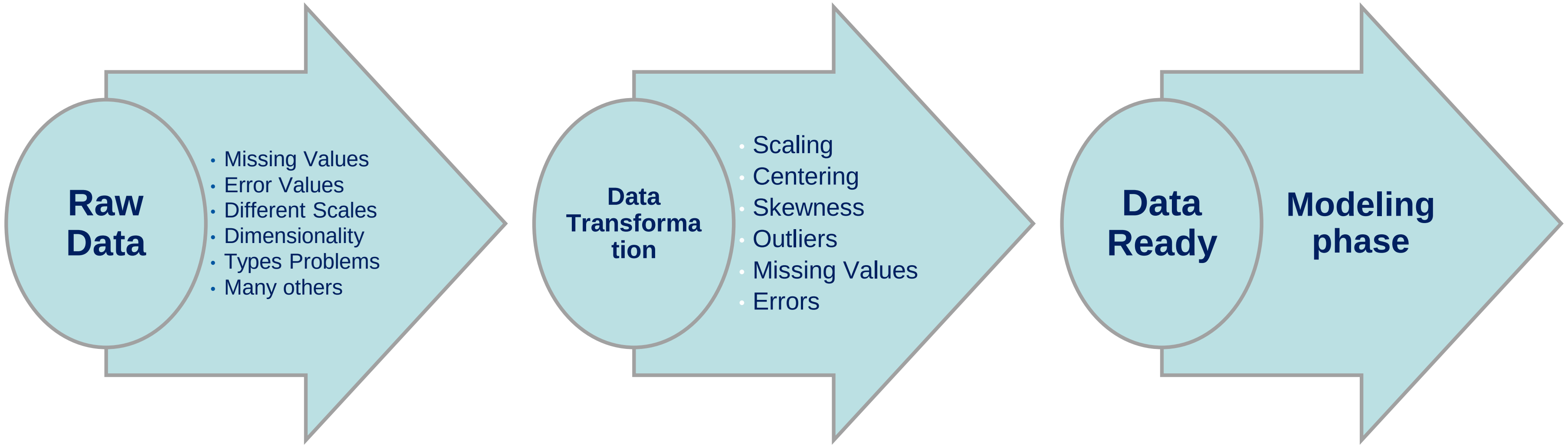

- Normalization
- Transformation
- Missing Values
- Outliers

- Data Splitting
- Features Engineering
- Estimating Performance
- Evaluation and Model Selection

- Study models accuracy
  - Work better than the naïve approach or previous system
- Do the results make sense in the context of the problem

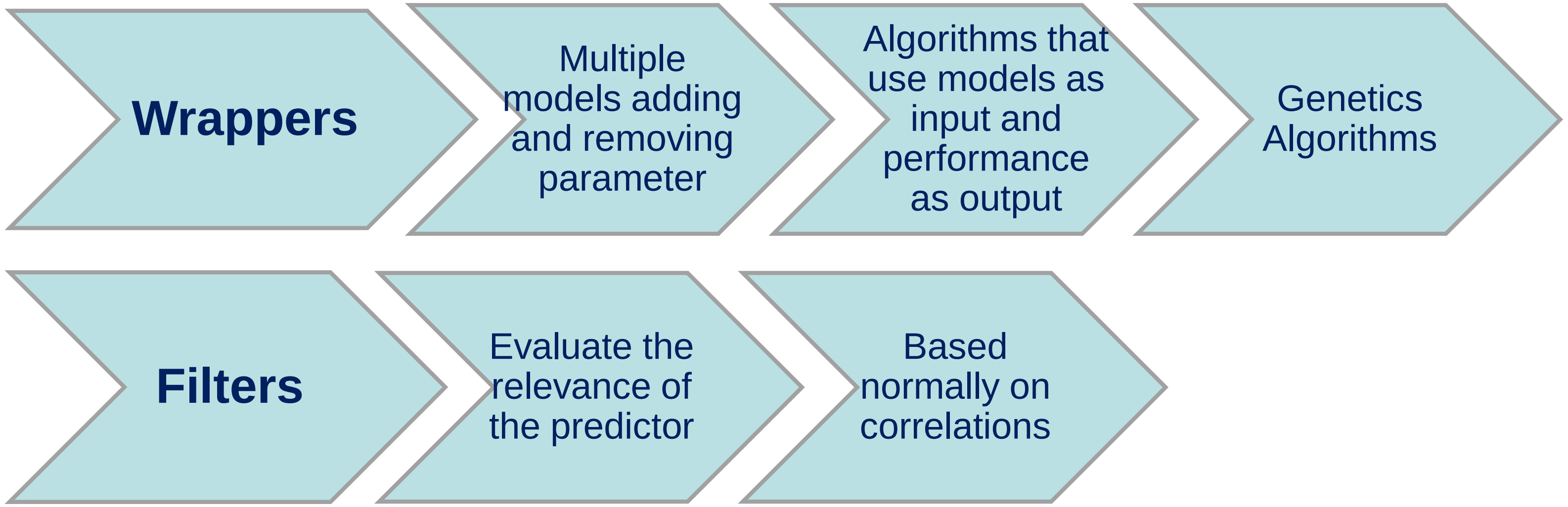
# ML as a Process: Data Preparation

- Needed for several reasons
  - Some Models have **strict data requirements**
    - Scale of the data, data point intervals, etc
  - Some characteristics of the data may impact dramatically on the model performance
- Time on data preparation **should not be underestimated**



# ML as a Process: Feature engineering

- **Determine the predictors (features) to be used** is one of the most critical questions
- Some times we **need to add predictors**
- Reduce Number:
  - Fewer predictors more interpretable model and less costly
  - Most of the models are affected by high dimensionality, specially for non-informative predictors




  - Binning predictors

# ML as a Process: Model Building

- Data Splitting
  - Allocate data to different tasks
    - model training
    - performance evaluation
  - Define Training, Validation and Test sets
- Feature Selection (Review the decision made previously)
- Estimating Performance
  - Visualization of results – discovery interesting areas of the problem space
  - Statistics and performance measures
- Evaluation and Model selection
  - The ‘no free lunch’ theorem no a priory assumptions can be made
  - Avoid use of favorite models if NEEDED