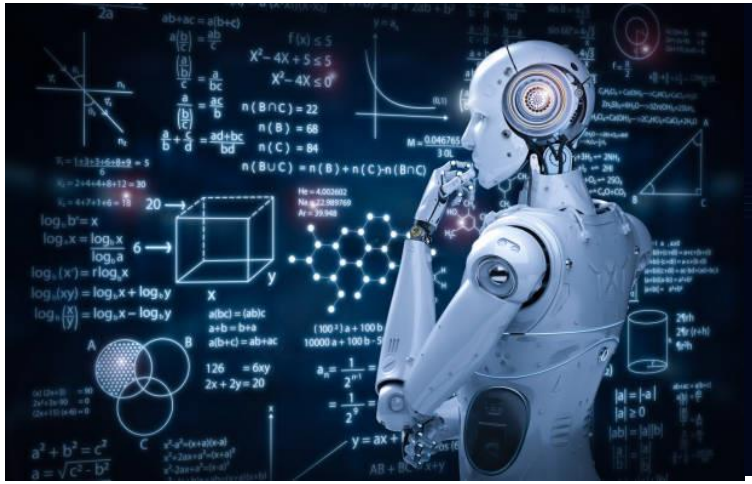
Welcome…

Data Analysis Overview

CS 797Q
Fall 2024

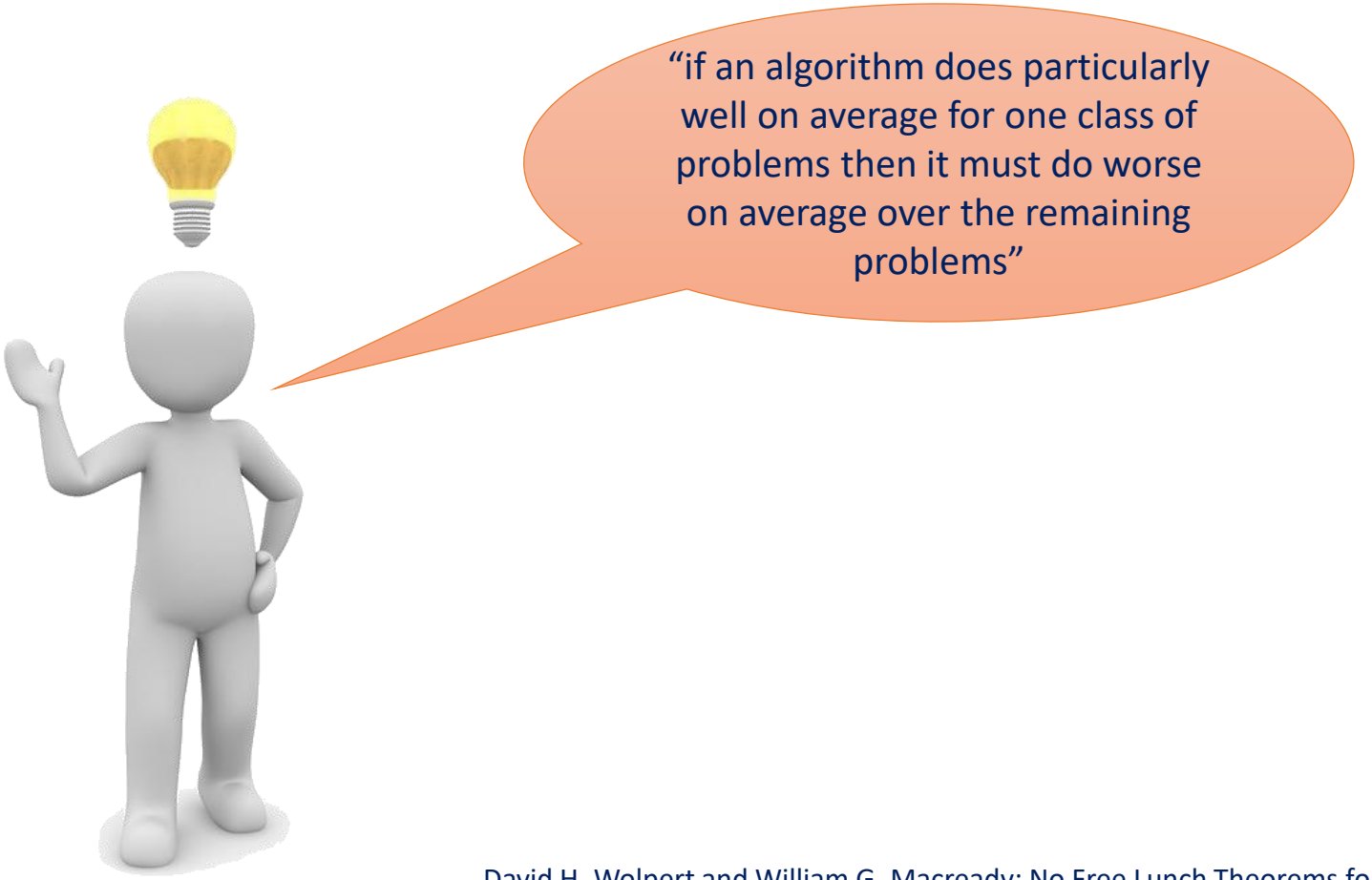
09/25/2024



Outline

- Overview
- Foundational Concepts
- Summary

Implication of the NFL Theorem



David H. Wolpert and William G. Macready: No Free Lunch Theorems for Optimization, IEEE Transactions on Evolutionary Computation, 1(1):67-82

Categories of Data Analysis Techniques

Category	Techniques Covered	Problem to be solved
Association Rules	Apriori	Relationships between items
Clustering	K-Means Clustering DB Scan	Grouping of similar items Identification of structures
Classification	K-nearest Neighbor Decision Trees Random Forests Logistic Regression Naive Bayes Support Vector Machines Neural Networks	Assignment of labels to objects
Regression	Linear Regression Ridge Lasso	Relationship between outcome and inputs
Time Series Analysis	ARMA	Identification of temporal structures Forecasting of temporal processes
Text Mining	Bag-of-Words Stemming/Lemmatization TF-IDF	Analysis of textual data

Outline

- Overview
- **Foundational Concepts**
- Summary

Machine Learning

- Definition after Tom M. Mitchell [2]:
  - 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.

- Relation to the data analysis techniques
  - Experience E: our data
  - Task T: clustering/association mining/classification/…
  - Performance Measure P: depends on tasks

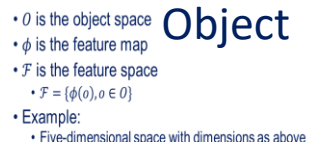
T. M. Mitchel: Machine Learning, McGraw Hill, 1997

Description of a „Whale“ Picture

- How would you describe this picture with general concepts?

Features of Objects

Feature map
$\phi: O \rightarrow \mathcal{F}$



Object

Features

hasFin	=	true
shape	=	oval
colorTop	=	black
colorBottom	=	white
background	=	blue

- O is the object space
- ϕ is the feature map
- $\mathcal{F}$ is the feature space
  - $\mathcal{F} = \{\phi(o), o \in O\}$
- Example:
  - Five-dimensional space with dimensions as above
  - $\phi(\text{"whalepicture"}) = (true, oval, black, white, blue)$

Scales of Features

- Stevens‘ levels of measurement

Categorical

Scale	Property	Allowed Operations	Example
Nominal	Classification or membership	$=, \neq$	Color as „black“, „white“ and „blue“
Ordinal	Comparison or levels	$=, \neq, >, <$	Size in „small“, „medium“, and „large“
Interval	Differences or affinities	$=, \neq, >, <, +, -$	Dates, temperatures, discrete numeric values
Ratio	Magnitudes or amounts	$=, \neq, >, <, +, -, \cdot, /$	Size in cm, duration in seconds, continuous numeric values

S. S. Stevens: On the Theory of Scales of Measurement, Science, 103(2684):677-680

Encoding Categorical Features

- Many algorithms can only work with numeric features

- Encode categorical features as binary numeric features
  - Example: $x \in \{\text{small}, \text{medium}, \text{large}\}$
  - Encode as three variables $x^{small}, x^{medium}, x^{large}$
  - $x^{small} = \begin{cases} 1 & if \;\; x \;\; = \text{small} \\ 0 & otherwise \end{cases}, \ldots$
  - Can also use one variable less, remaining case is encoded by all zeros

- This is called *One-Hot-Encoding*

Training Data

- *Instances* of objects described by their features

$\phi(o)$					
hasFin	shape	colorTop	colorBottom	background	**value of interest**
true	oval	black	black	blue	**whale**
false	rectangle	brown	brown	green	**bear**
…	…	…	…	…	**…**

- *Supervised* learning if the value of interest is known
  → Classification, regression
- Otherwise *unsupervised learning*
  → Clustering, Association Rule Mining

The Test Data

- Data for the evaluation of analysis results
  - Same distribution as training data
- Training data ≠ Test data
  - Evaluate generalization
  - Avoid overfitting
    - Analysis results only valid on training data
    - Different and not working on unseen data
- Test data often difficult to obtain

Hold-out Data

- Data not used for training at all

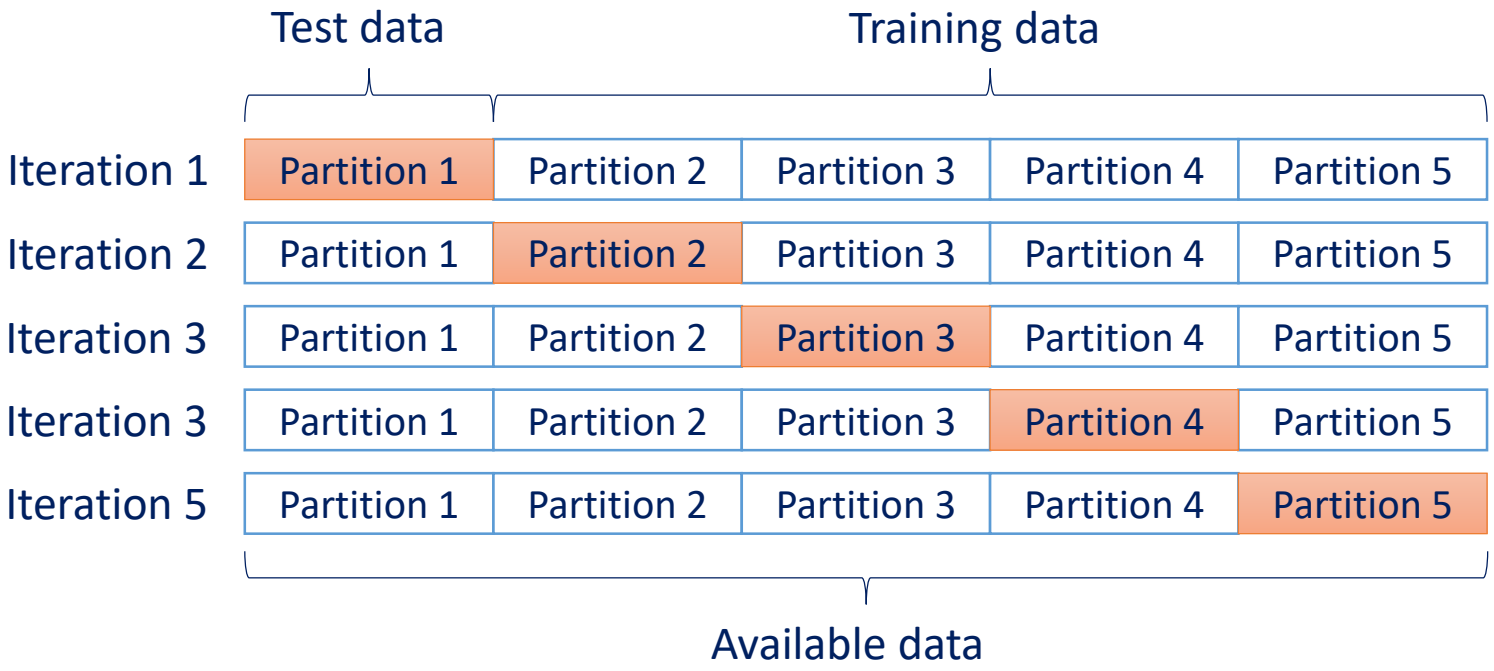
- Commonly used hold out data sizes
  - 50% of all data
  - 33% of all data
  - 25% of all data in case a validation set is used

Depends a lot on available data!

- Example:
  - Nine months of customer transactions available
  - First six months as training data
  - Last three months as test data

k-fold Cross Validation

- Create k partitions of available data
- One partition for testing, all others for training
- Estimate performance by averaging over the iterations



Outline

- Overview
- Foundational Concepts
- **Summary**

Summary

- No generic algorithm for all problems
- Objects are described by features
- Features are used for learning about objects
- Data usually split into different sets for different purposes

What is Machine Learning?

"Learning is any process by which a system improves performance from experience."

- Machine Learning is the science (and art) of programming computers so they can *learn from data*.

More engineering-oriented definition:

- Machine Learning is the study of algorithms that improve their performance P at some task T with experience E.

Example

- Your spam filter is a Machine Learning program that, given examples of spam emails (e.g., flagged by users) and examples of regular (non-spam, also called “ham”) emails, can learn to flag spam.

- The examples that the system uses to learn are called the *training set*. Each training example is called a *training instance* (or *sample*).

- This particular performance measure is called *accuracy*, and it is often used in classification tasks.

Types of Learning

- **Supervised learning**

– Given: training data + desired outputs (labels)

- **Unsupervised learning**

– Given: training data (without desired outputs)

- **Semi-supervised learning**

– Given: training data + a few desired outputs

- **Reinforcement learning**

– Rewards from sequence of actions

An agent learns to perform actions in an environment to maximize a reward.

Used heavily in robotics, gaming, and certain types of optimization problems.

Supervised Learning: Regression

- Given $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$
- Learn a function $f(x)$ to predict y given x
  - y is real-valued == regression

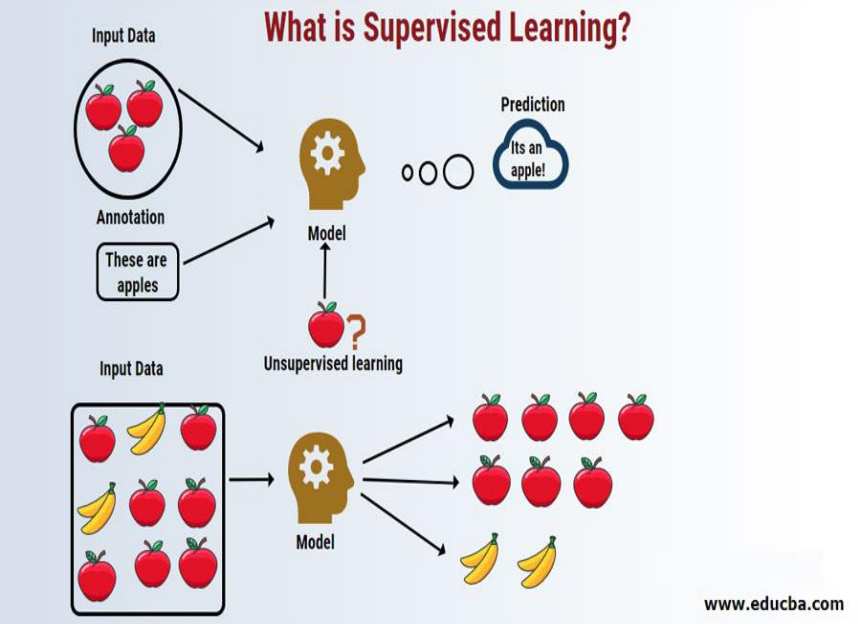
Examples of Supervised Learning:

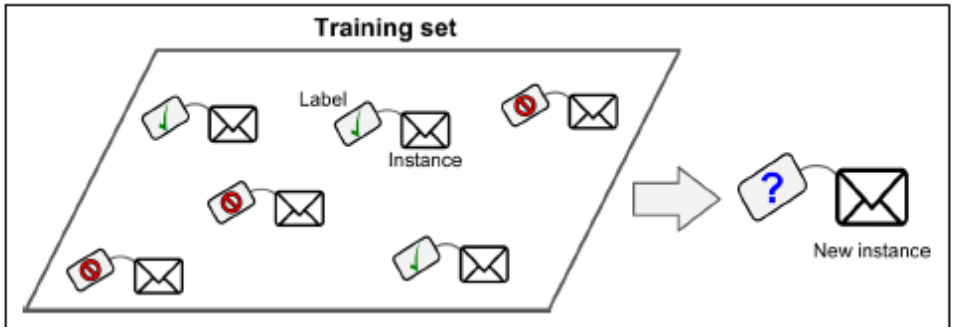
Regression: Predicting a continuous output.

- Example: Predicting the price of a house based on its features (number of rooms, location, size, etc.). Given a dataset of houses with known prices and their corresponding features.

Supervised Learning: Classification

- Given $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$
- Learn a function $f(x)$ to predict y given x
  - y is categorical == classification





Supervised Learning: Classification

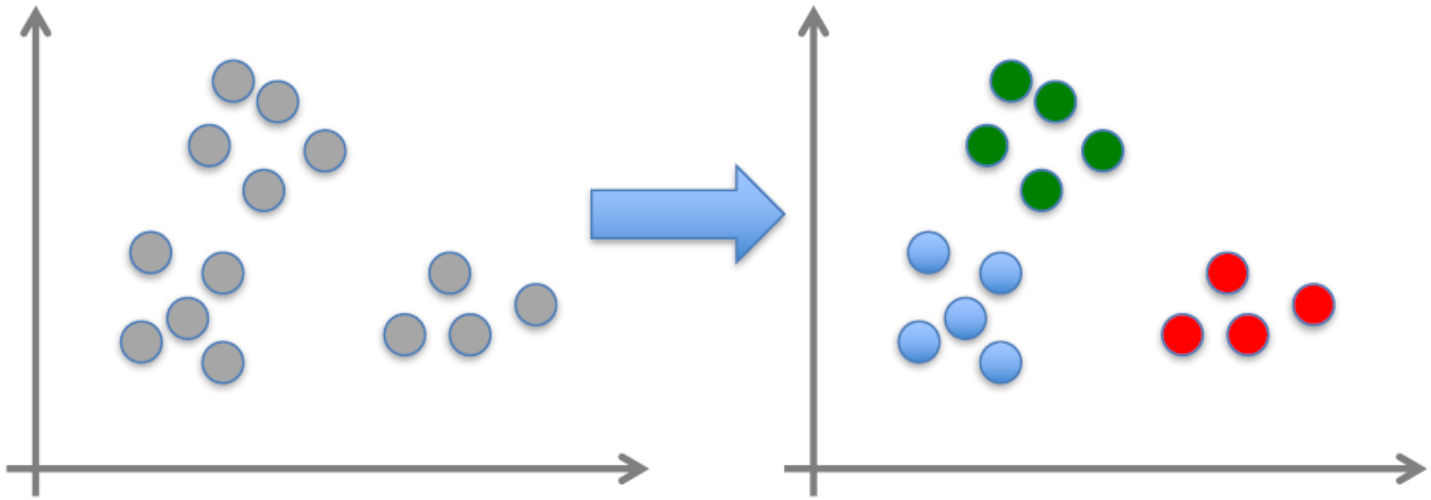
- Classification: Predicting a discrete label or category.
- Example: Email spam detection. In this case, emails are labeled either as "spam" or "not spam."

Example

- K-nearest neighbours
- Linear Regression
- Logistic Regression
- Support Vector Machines
- Decision Trees and Random Forests
- Neural Networks

Unsupervised Learning

- Given $x_1, x_2, \ldots, x_n$ (without labels)
- Output hidden structure behind the x's
  - E.g., clustering

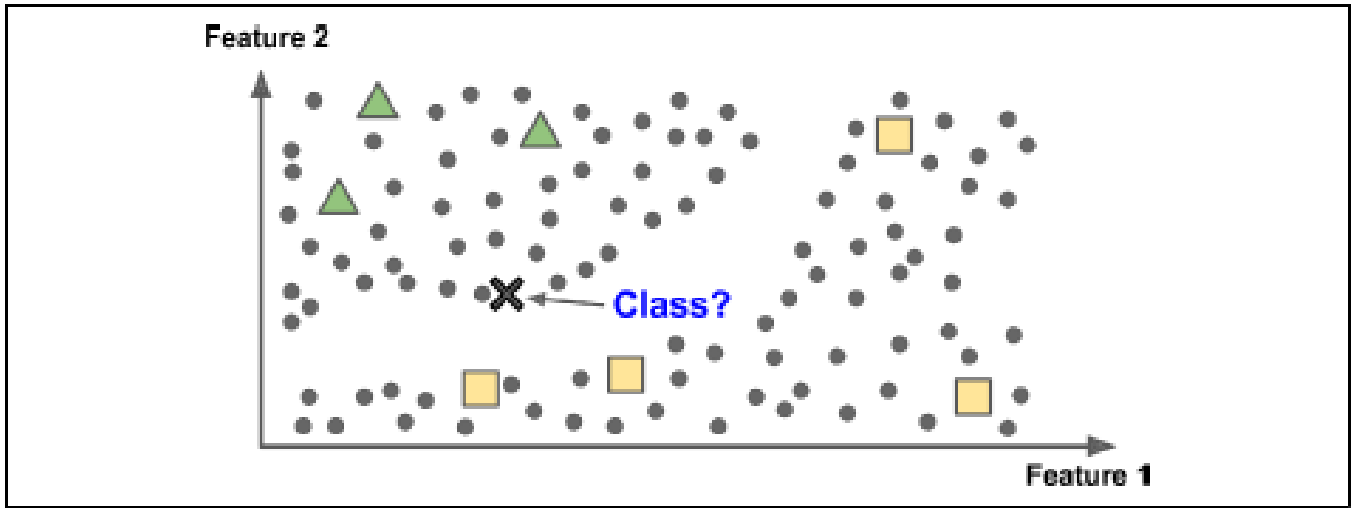


Clustering:
Clustering involves grouping data points together based on their similarities.

Clustering
—k-Means
—Hierarchical Cluster Analysis (HCA)
—Expectation Maximization

Semi-supervised learning

- Since labeling data is usually time-consuming and costly, you will often have plenty of unlabeled instances, and few labeled instances.
- Some algorithms can deal with data that's partially labeled. This is called *semi-supervised learning.*



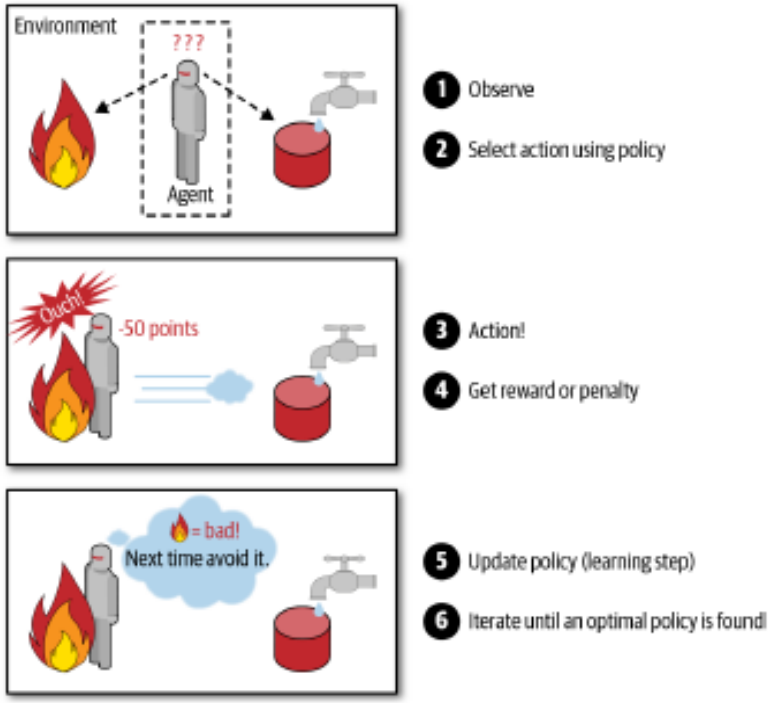
Reinforcement Learning

- Given a sequence of states and actions with (delayed) rewards, output a policy

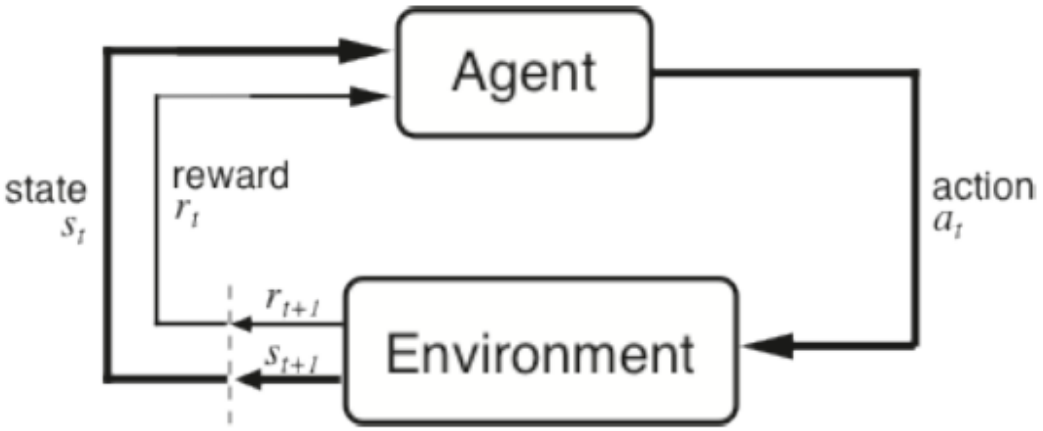
- Policy is a mapping from states actions that tells you what to do in a given state

Examples:

- Credit assignment problem
- Game playing
- Robot in a maze
- Balance a pole on your hand



The Agent-Environment Interface



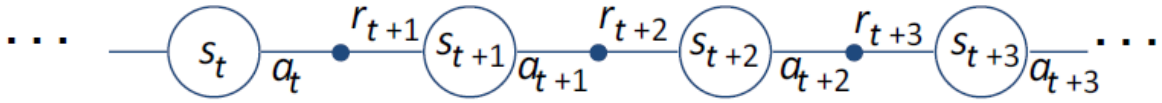
Agent and environment interact at discrete time steps : $t = 0, 1, 2, \mathrm{K}$

Agent observes state at step t: $s_t \in S$

produces action at step t: $a_t \in A(s_t)$

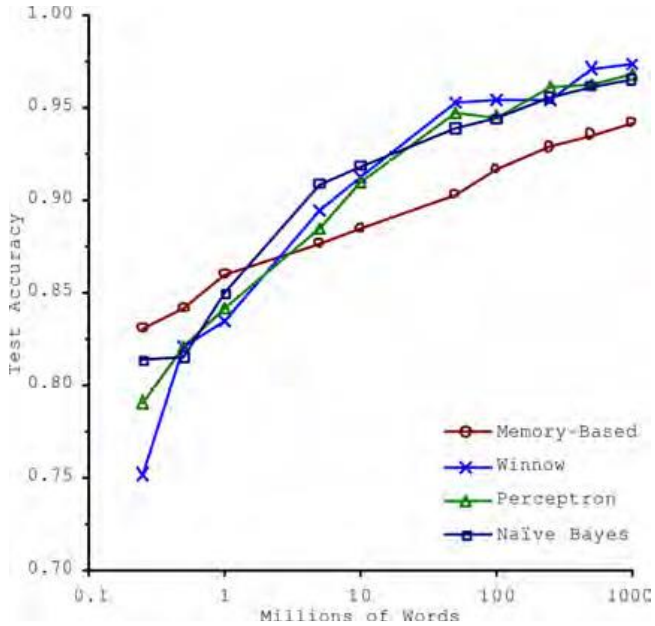
gets resulting reward : $r_{t+1} \in \Re$

and resulting next state : s_{t+1}



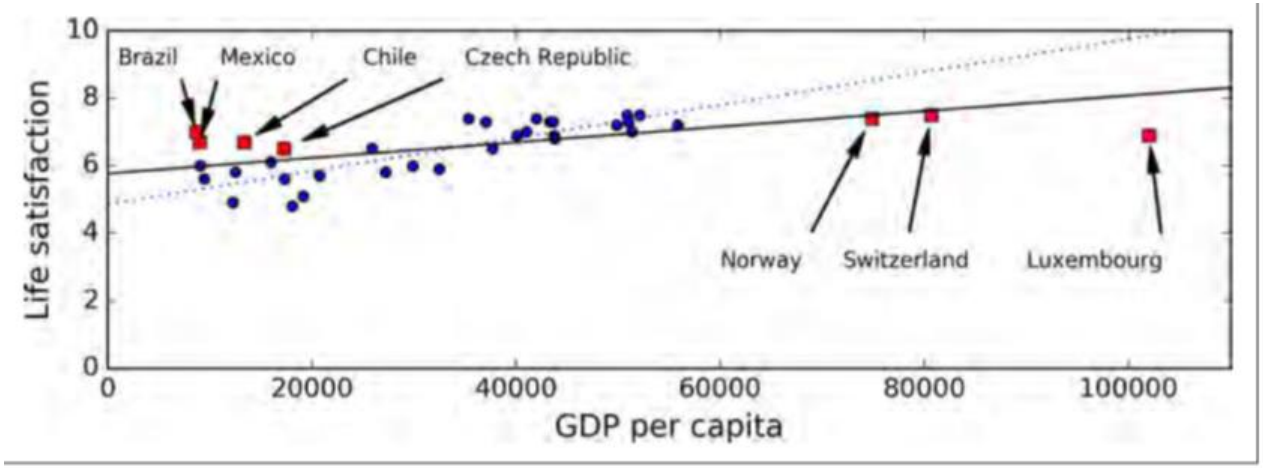
Main Challenges of Machine Learning

- Insufficient Quantity of Training Data



Non-representative Training Data

In order to generalize well, it is crucial that your training data be representative of the new cases you want to generalize to.
This is true whether you use instance-based learning or model-based learning.



Poor-Quality Data

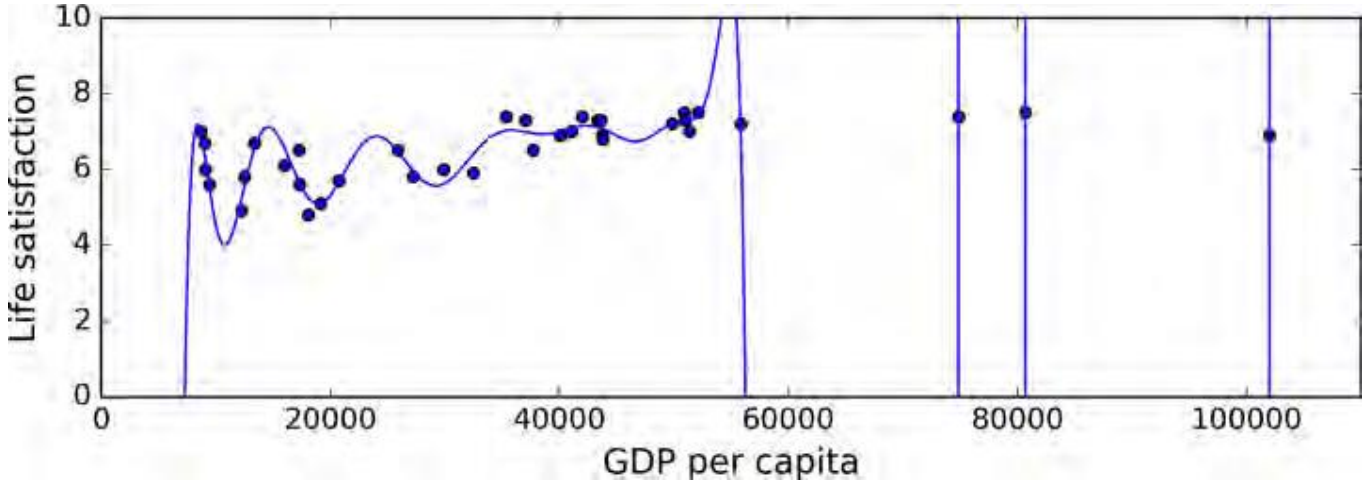
- Obviously, if your training data is full of errors, outliers, and noise (e.g., due to poor quality measurements), it will make it harder for the system to detect the underlying patterns, so your system is less likely to perform well.

- If some instances are clearly outliers, it may help to simply discard them or try to fix the errors manually.

- If some instances are missing a few features (e.g., 5% of your customers did not specify their age), you must decide whether you want to ignore this attribute altogether

Overfitting the Training Data

In Machine Learning this is called *overfitting*: it means that the model performs well on the training data, but it does not generalize well.



Avoiding Overfitting

Overfitting happens when the model is too complex relative to the amount and noisiness of the training data. The possible solutions are:

- To simplify the model by selecting one with fewer parameters (e.g., a linear model rather than a high-degree polynomial model), by reducing the number of attributes in the training data or by constraining the model

- To gather more training data

- To reduce the noise in the training data (e.g., fix data errors and remove outliers)

Contd..,

- Constraining a model to make it simpler and reduce the risk of overfitting is called *regularization*

- The amount of regularization to apply during learning can be controlled by a *hyperparameter*

- A hyperparameter is a parameter of a learning algorithm (not of the model). As such, it is not affected by the learning algorithm itself; it must be set prior to training and remains constant during training

Testing and Validating

- The only way to know how well a model will generalize to new cases is to actually try it out on new cases.

- A better option is to split your data into two sets: the *training set* and the *test set*. As these names imply, you train your model using the training set, and you test it using the test set.

- The error rate on new cases is called the *generalization error* (or *out-of-sample error*), and by evaluating your model on the test set, you get an estimate of this error.
- This value tells you how well your model will perform on instances it has never seen before.
- If the training error is low (i.e., your model makes few mistakes on the training set) but the generalization error is high, it means that *your model is overfitting the training data*.