Artificial Intelligence

- An Introduction to **Machine Learning** -

Hachem Kadri

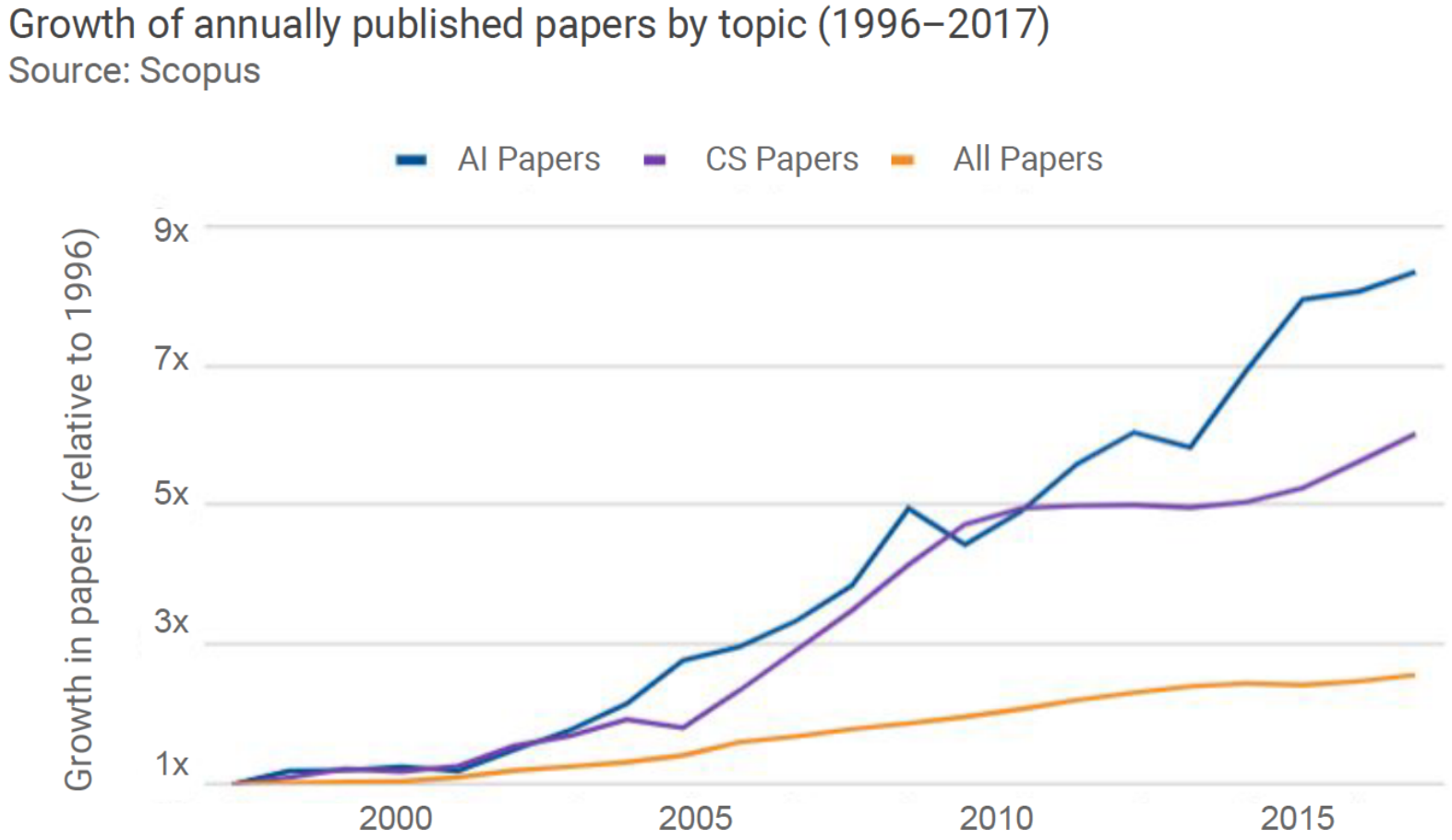
Aix-Marseille University, CNRS
Laboratoire d'Informatique et des Systèmes, LIS
QARMA team
https://qarma.lis-lab.fr/

March, 2019

Some plots from the AI Index 2018 Report

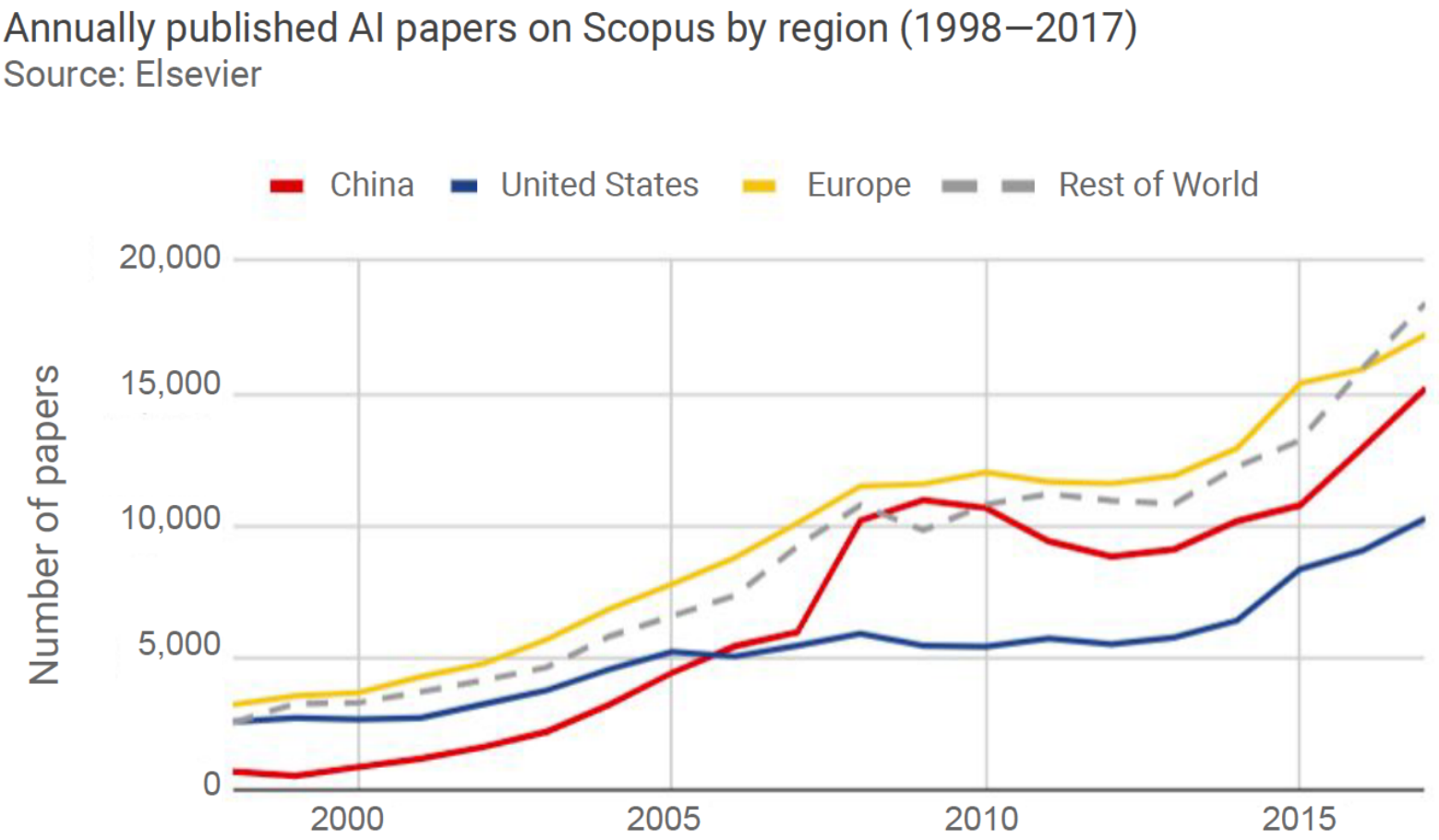
https://aiindex.org/

Published Papers: Papers by topic



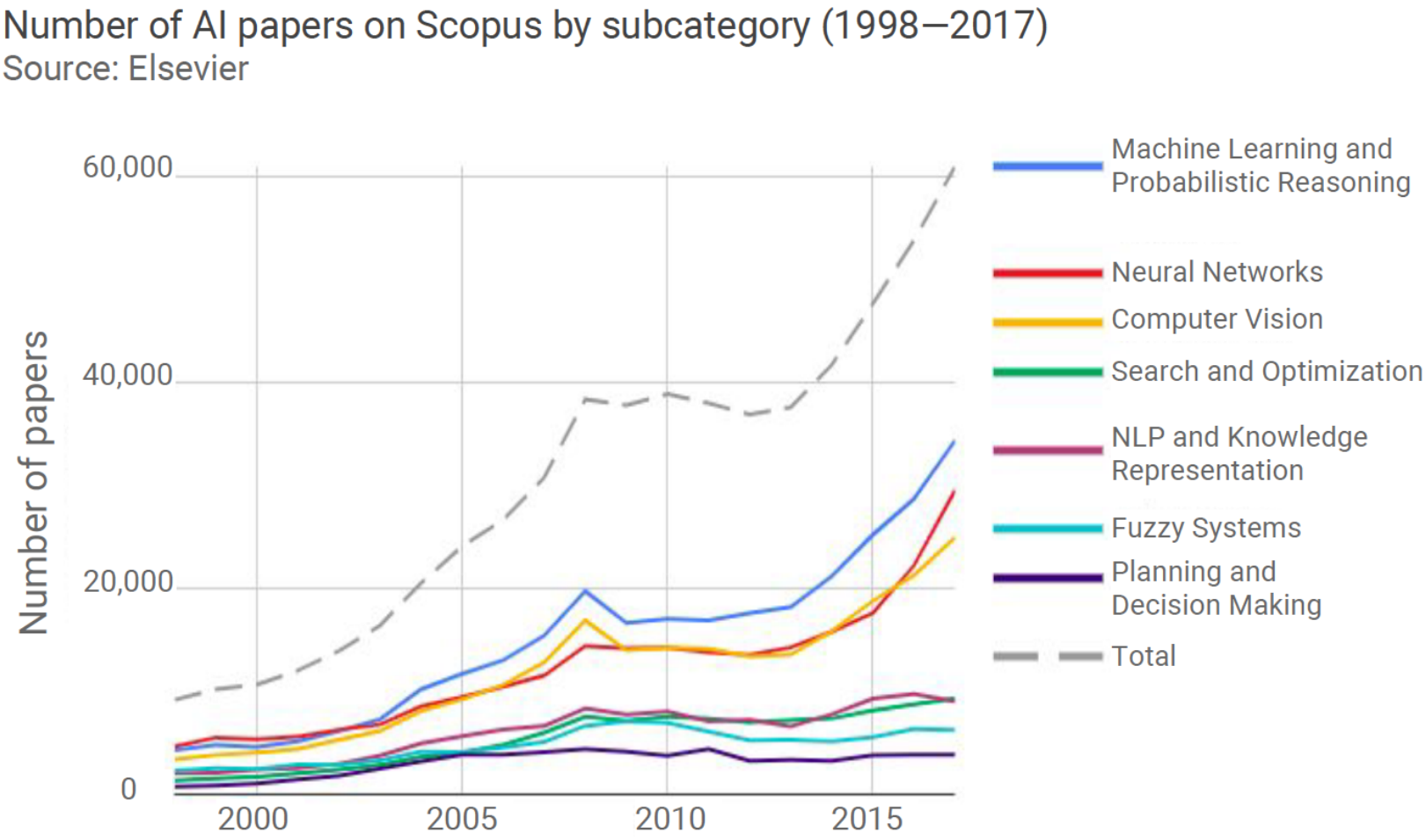
- AI papers on Scopus have increased 8x since 1996
- CS papers increased 6x during the same timeframe

Published Papers: AI papers by region



- In 2017, 28% of AI papers on Scopus were affiliated with European authors, followed by China (25%) and the U.S. (17%).

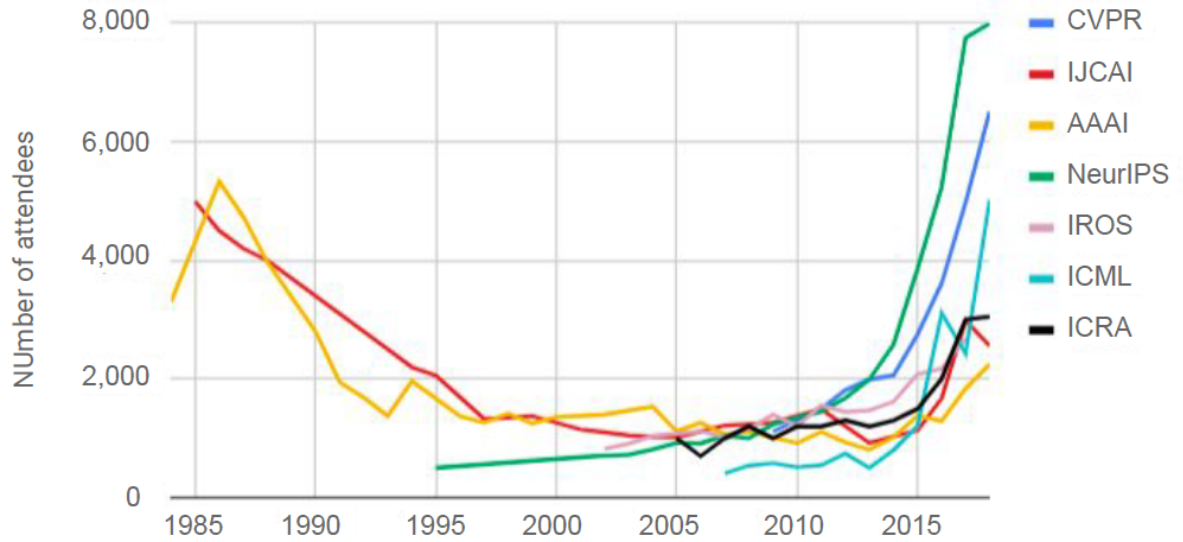
Published Papers: AI papers by subcategory



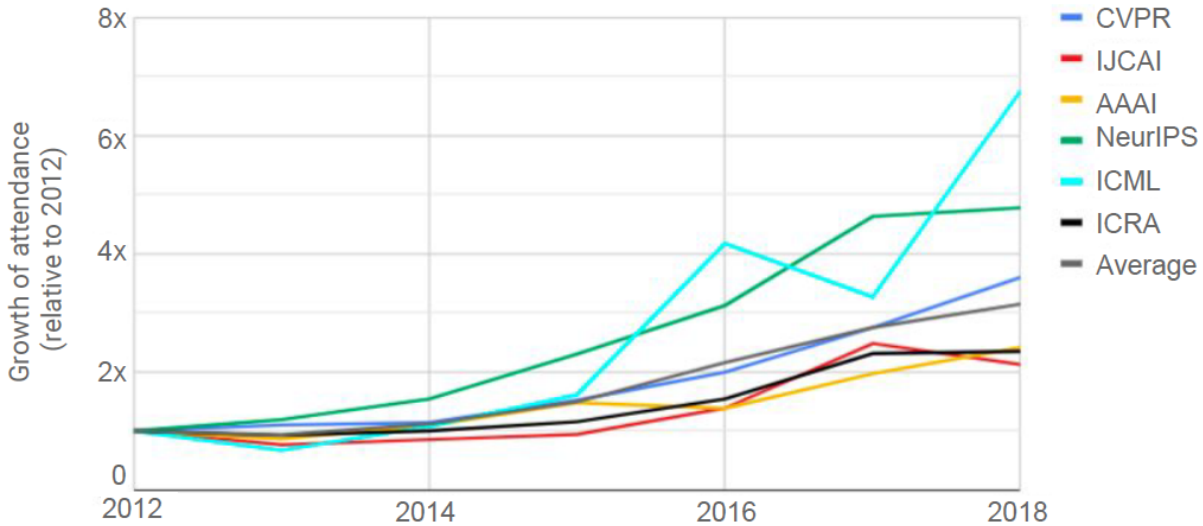
- 56% of papers fell into the Machine Learning and Probabilistic Reasoning category in 2017, compared to 28% in 2010.

Participation: Large AI conferences

Attendance at large conferences (1984–2018)
Source: Conference provided data

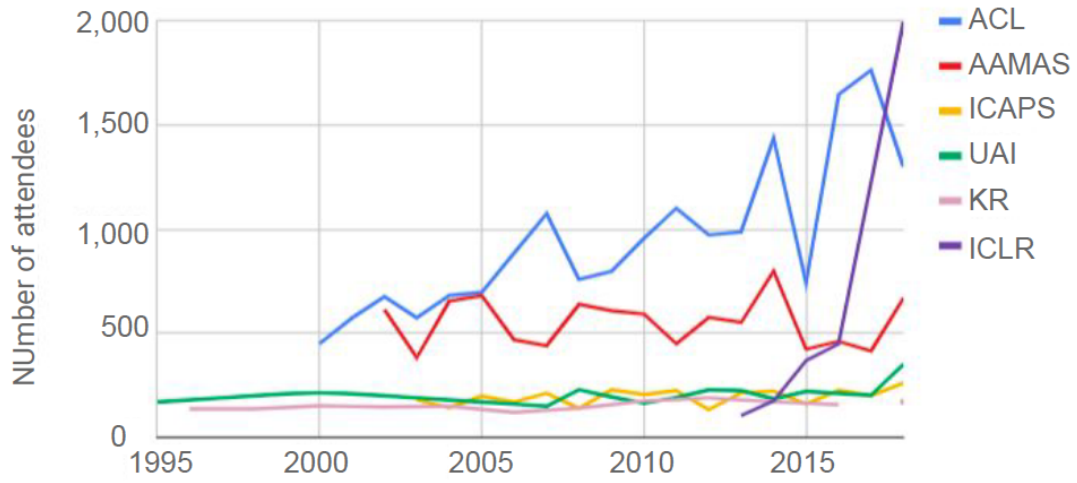


Growth of large conference attendance (2012–2018)
Source: Conference provided data



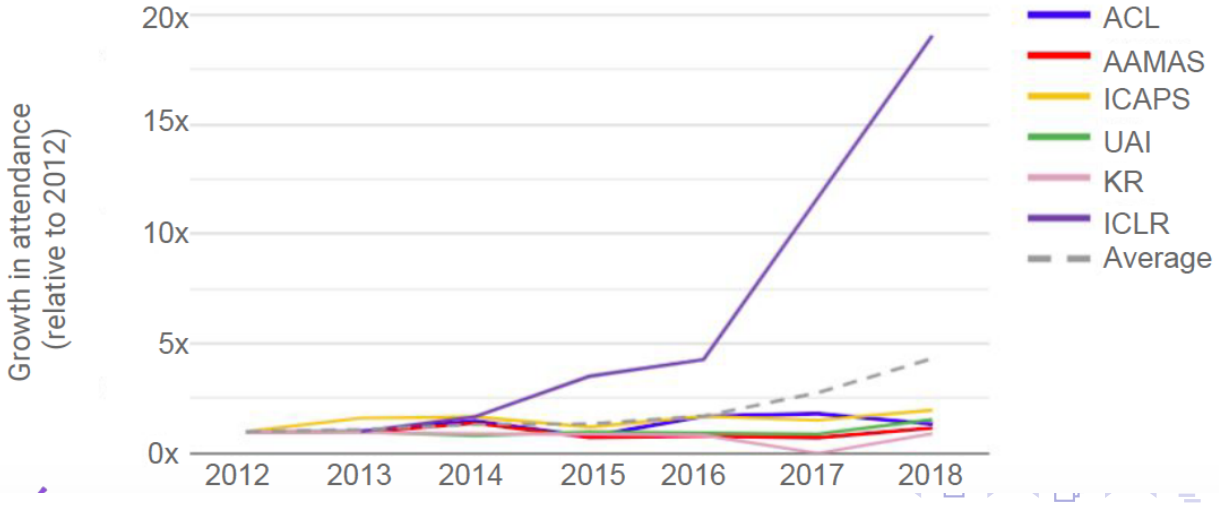
Participation: Small AI conferences

Attendance at small conferences (1995–2018)
Source: Conference provided data



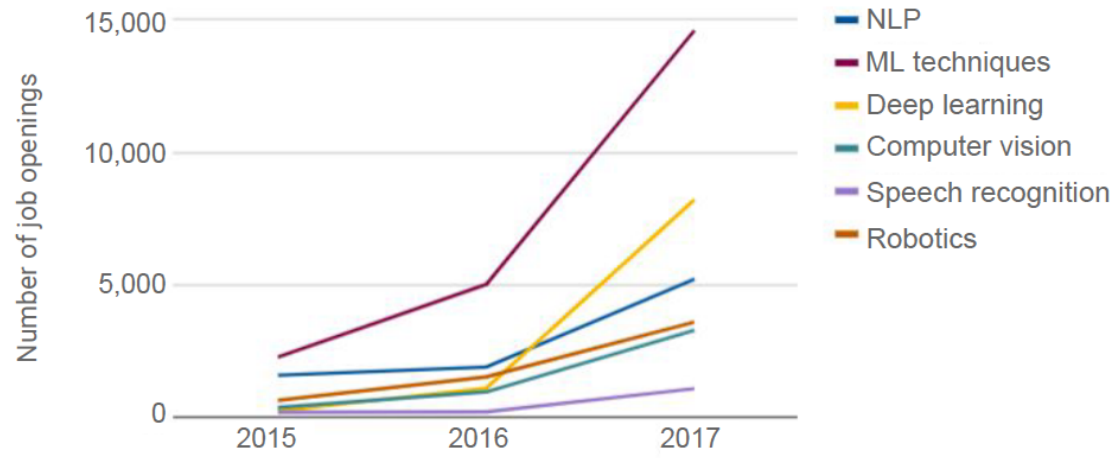
Note: 2018 was the first year that KR held a workshop. For consistency, workshop attendees are not included in KR's attendance count in the visual above.

Growth of small conference attendance (2012–2018)
Source: Conference provided data

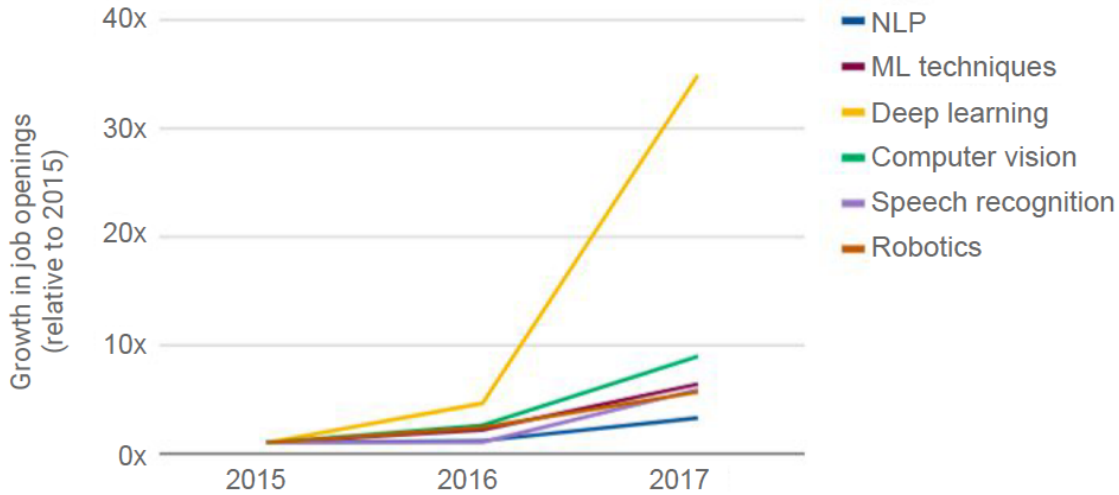


Jobs: Openings by AI skill

Job openings by AI skills required (2015 – 2017)
Source: Monster.com



Growth of job openings by AI skills required (2015 – 2017)
Source: Monster.com



Note: While AI job openings are increasing across the board, they still represent a very low number of job openings for computer engineers.

Act I: Machine Learning

Overview and basic concepts

What is Machine Learning?

"Field of study that gives computers the ability to learn without being explicitly programmed."

– Arthur Samuel, 1959

Learning versus Programming

Algorithm

A sequence of instructions that are carried out to transform the input to the output.

Sorting

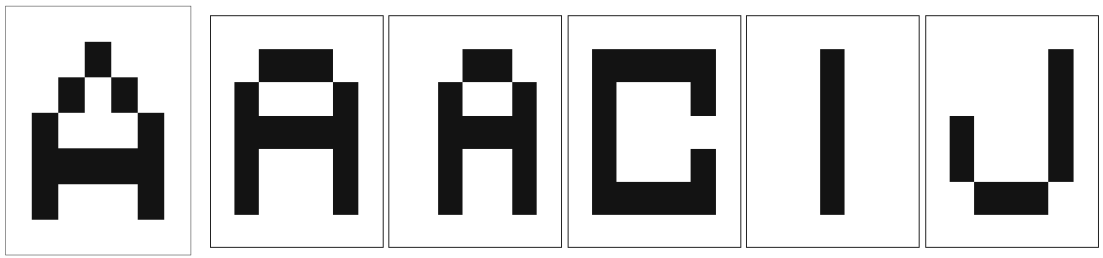
5	1	12	-5	16

unsorted

-5	1	5	12	16

sorted

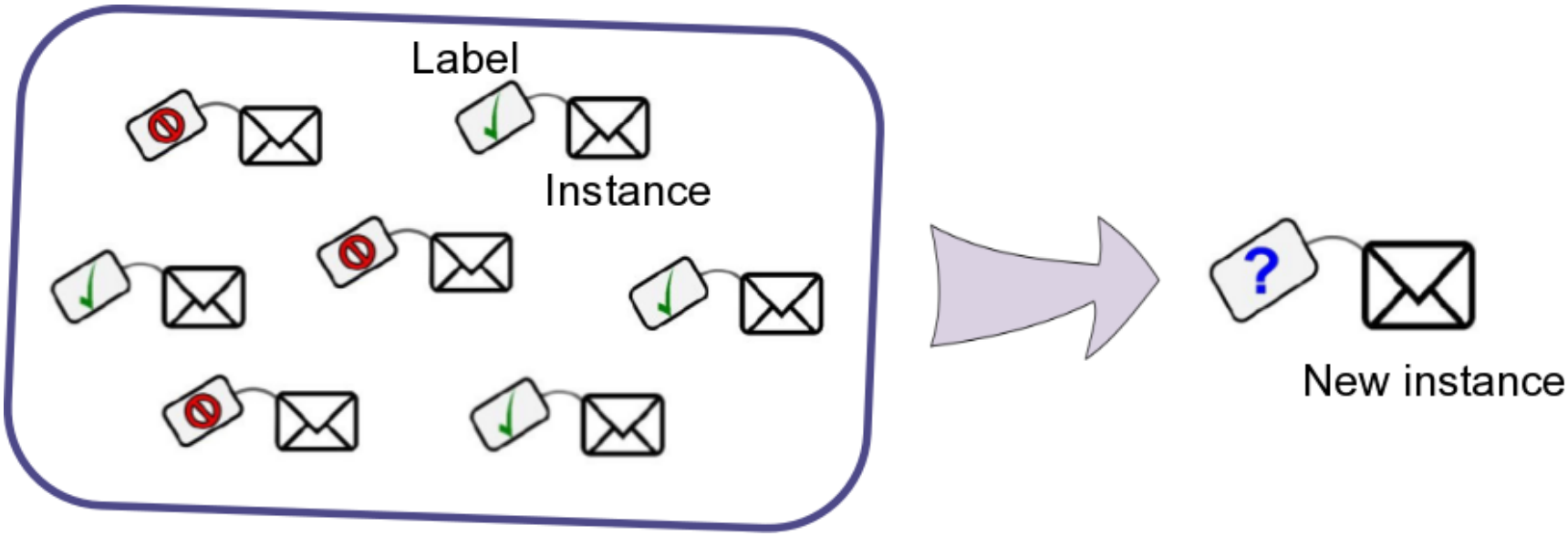
Letter Recognition



Alphabet in a bitmap representation

When do we need Machine Learning?

Tasks that are too complex to program



Spam filter (from A. Géron, 2017)

Machine Learning

"is the study of computer algorithms that improve automatically through experience."

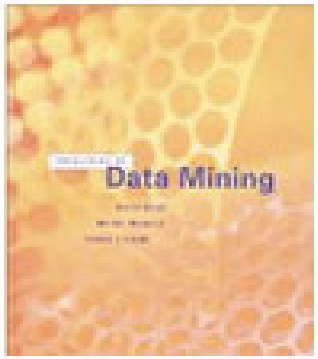
– Tom Mitchell, 1997

Recommendation System

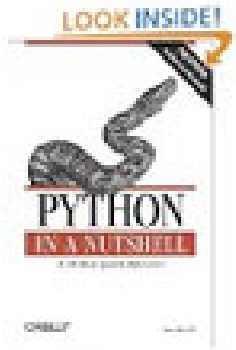
Grant, Welcome to Your Amazon.com (If you're not Grant Ingersoll, click here.)

Today's Recommendations For You

Here's a daily sample of items recommended for you. Click here to **see all recommendations**.



Principles of Data Mining (A...
by David J....
(17) $52.00



Python in a Nutshell, Secon...
by Alex Mart...
(40) $26.39

Introductory Statistics wit...
by Peter Dal...
(20) $48.56

- **Netflix Prize:**
  Beat Netflix recommender system, using Netflix data → Win $1 million
- **Data:**
  480,000 users
  18,000 movies
  100 million observed ratings = only 1.1% of ratings observed

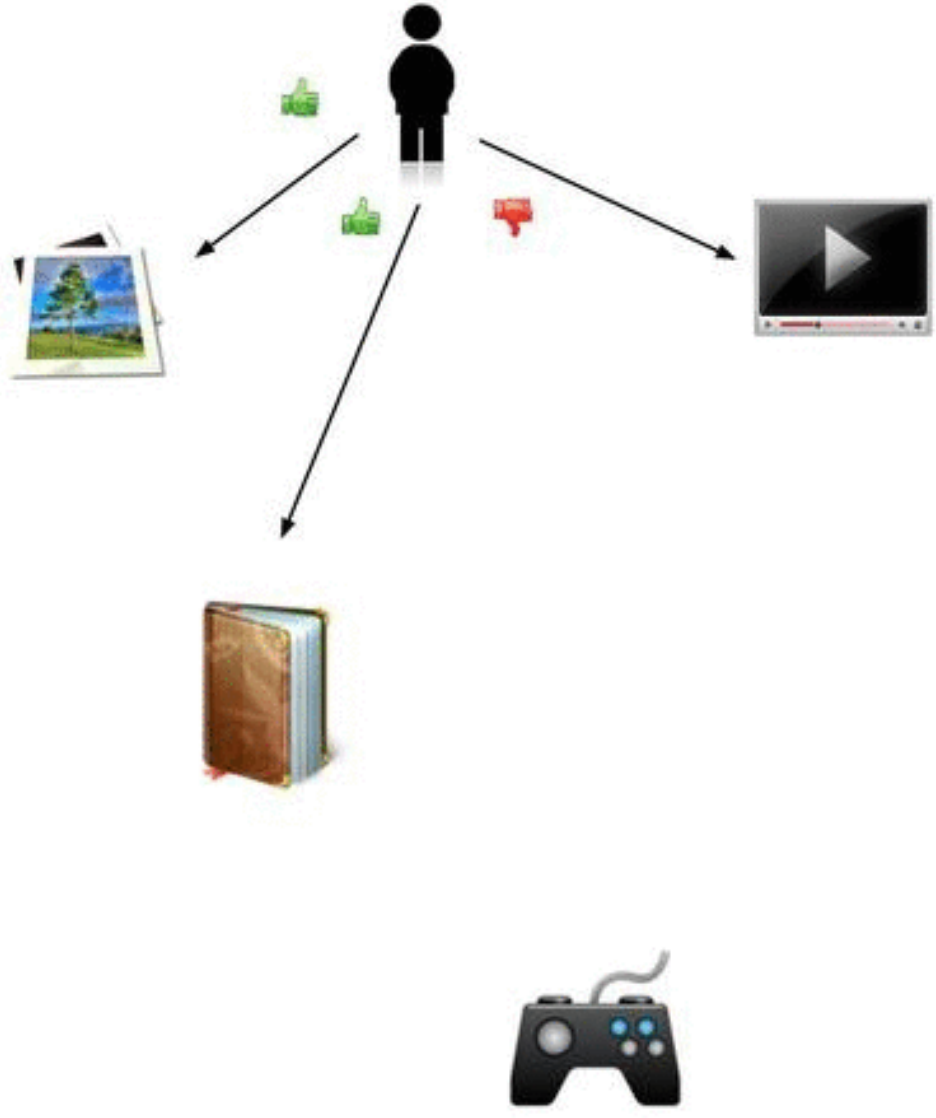
"The Netflix Prize seeks to substantially improve the accuracy of predictions about how much someone is going to love a movie based on their movie preferences."

`https://www.netflixprize.com`

Collaborative Filtering

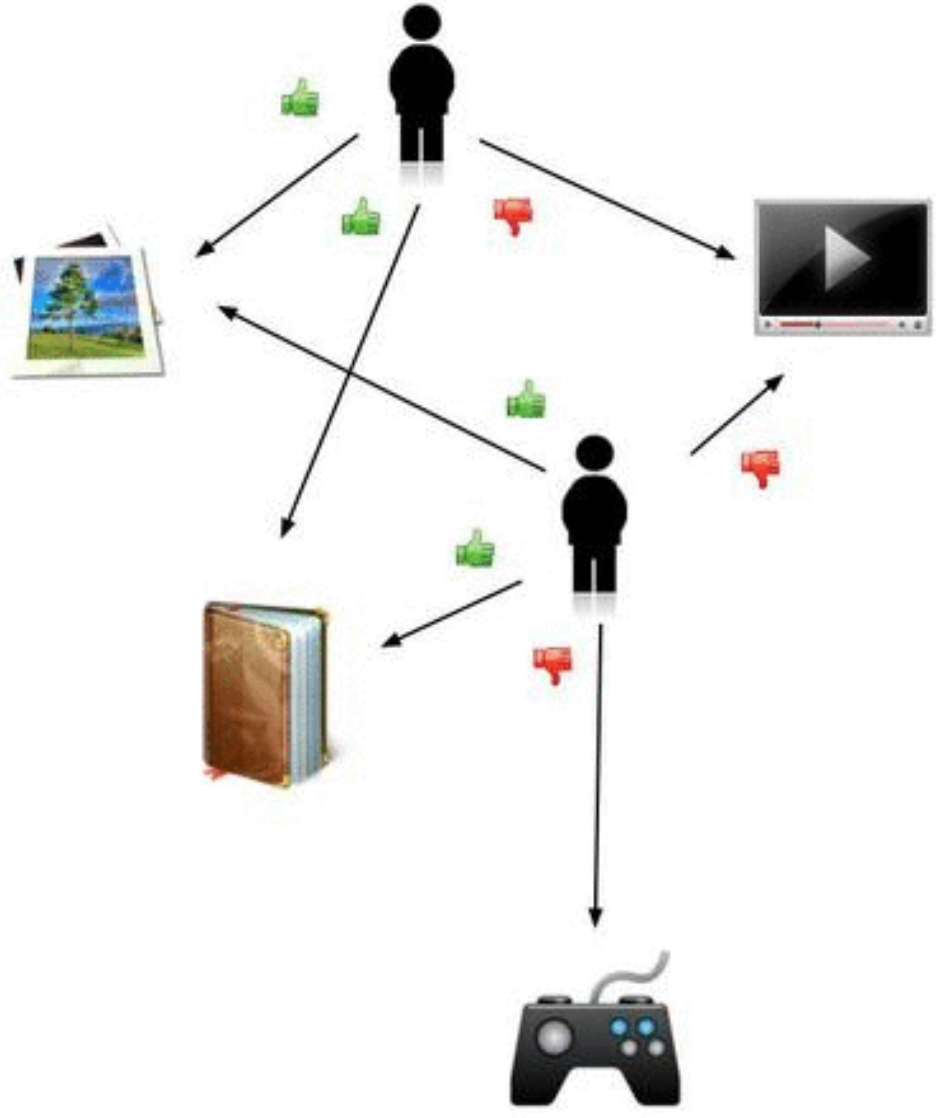
https://en.wikipedia.org/wiki/Collaborative_filtering

Collaborative Filtering



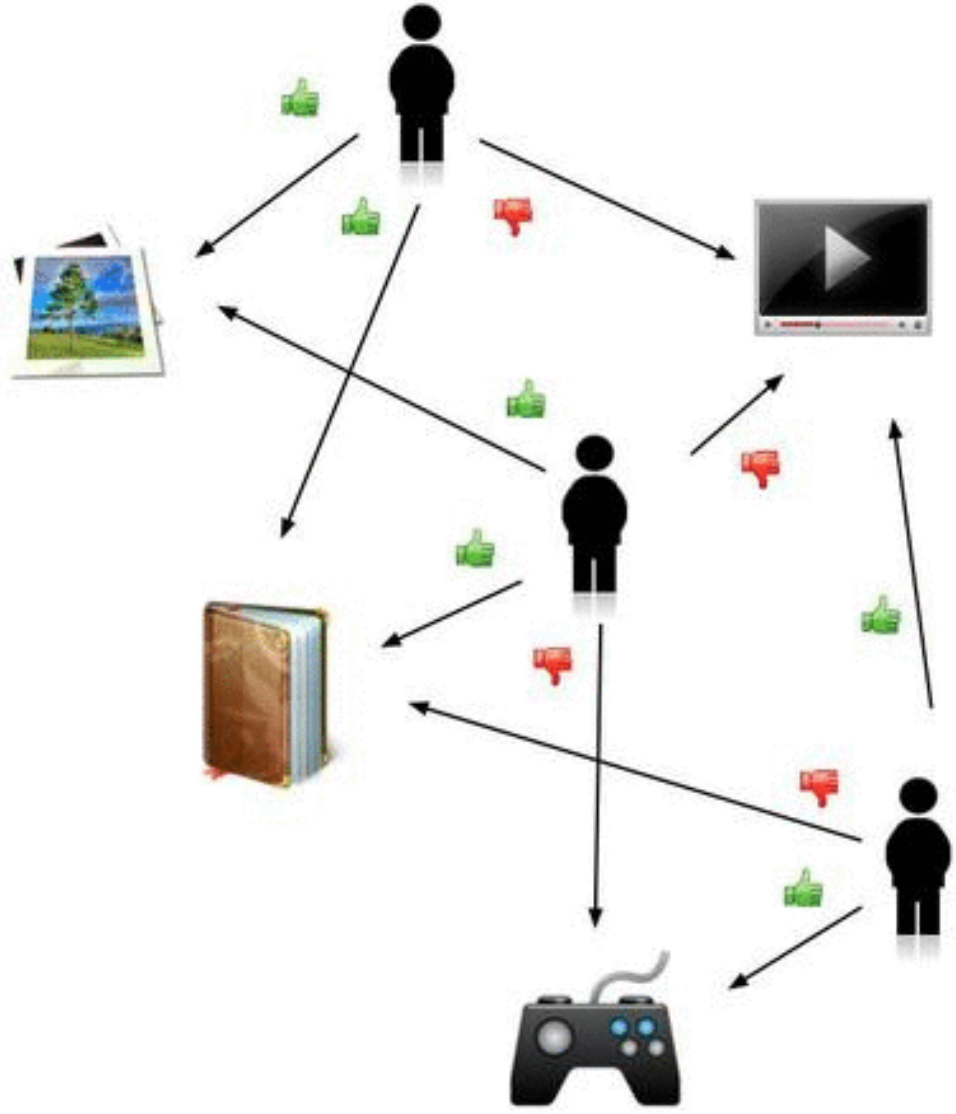
https://en.wikipedia.org/wiki/Collaborative_filtering

Collaborative Filtering



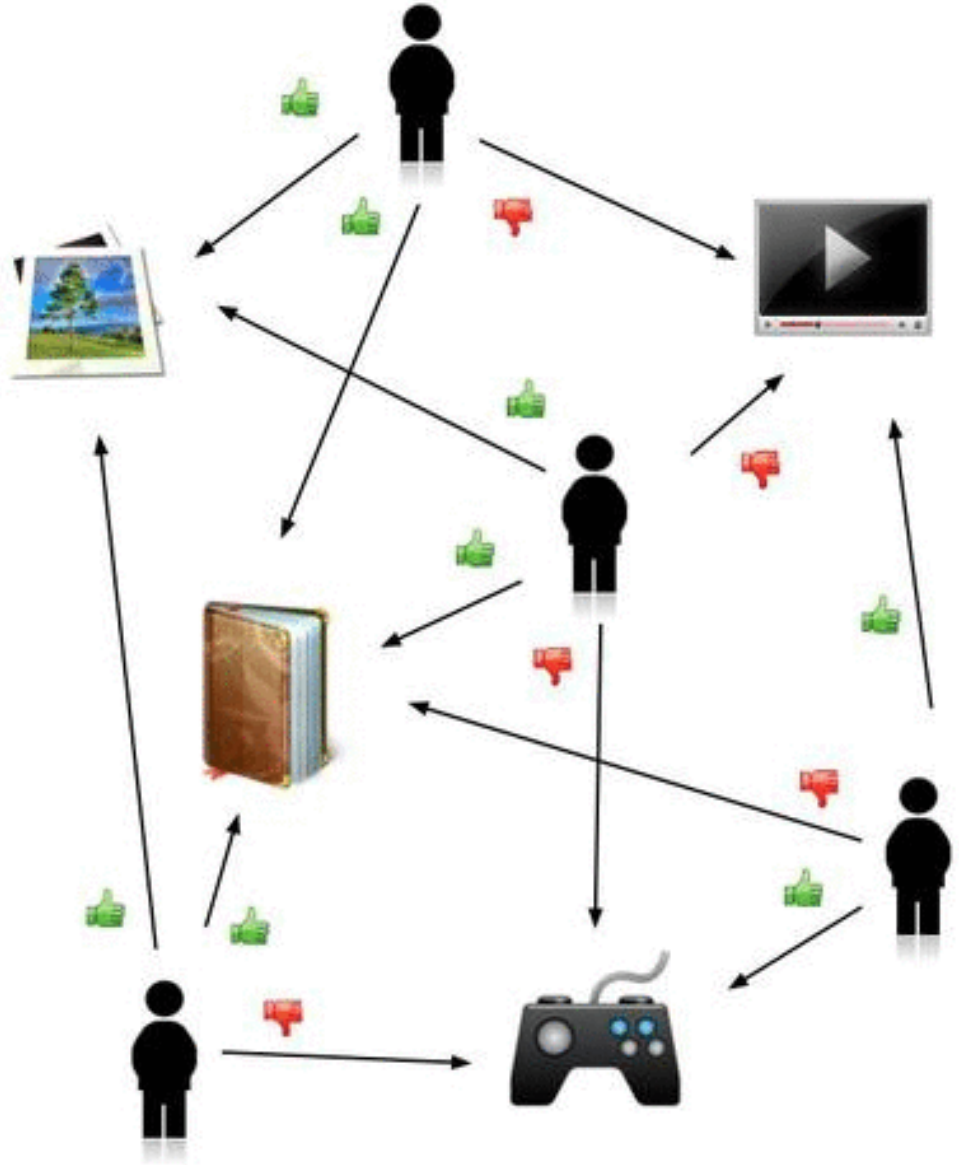
https://en.wikipedia.org/wiki/Collaborative_filtering

Collaborative Filtering



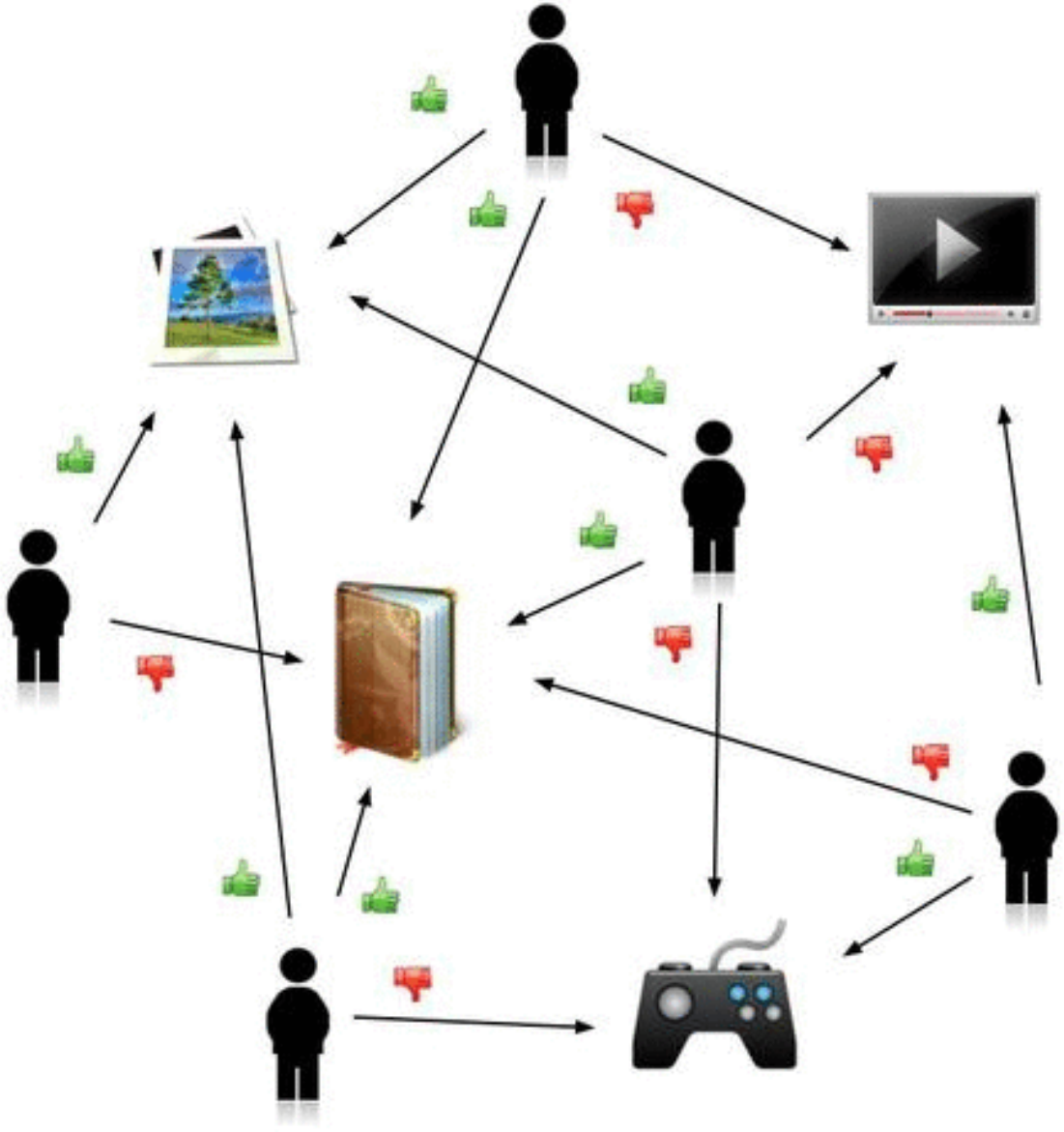
https://en.wikipedia.org/wiki/Collaborative_filtering

Collaborative Filtering



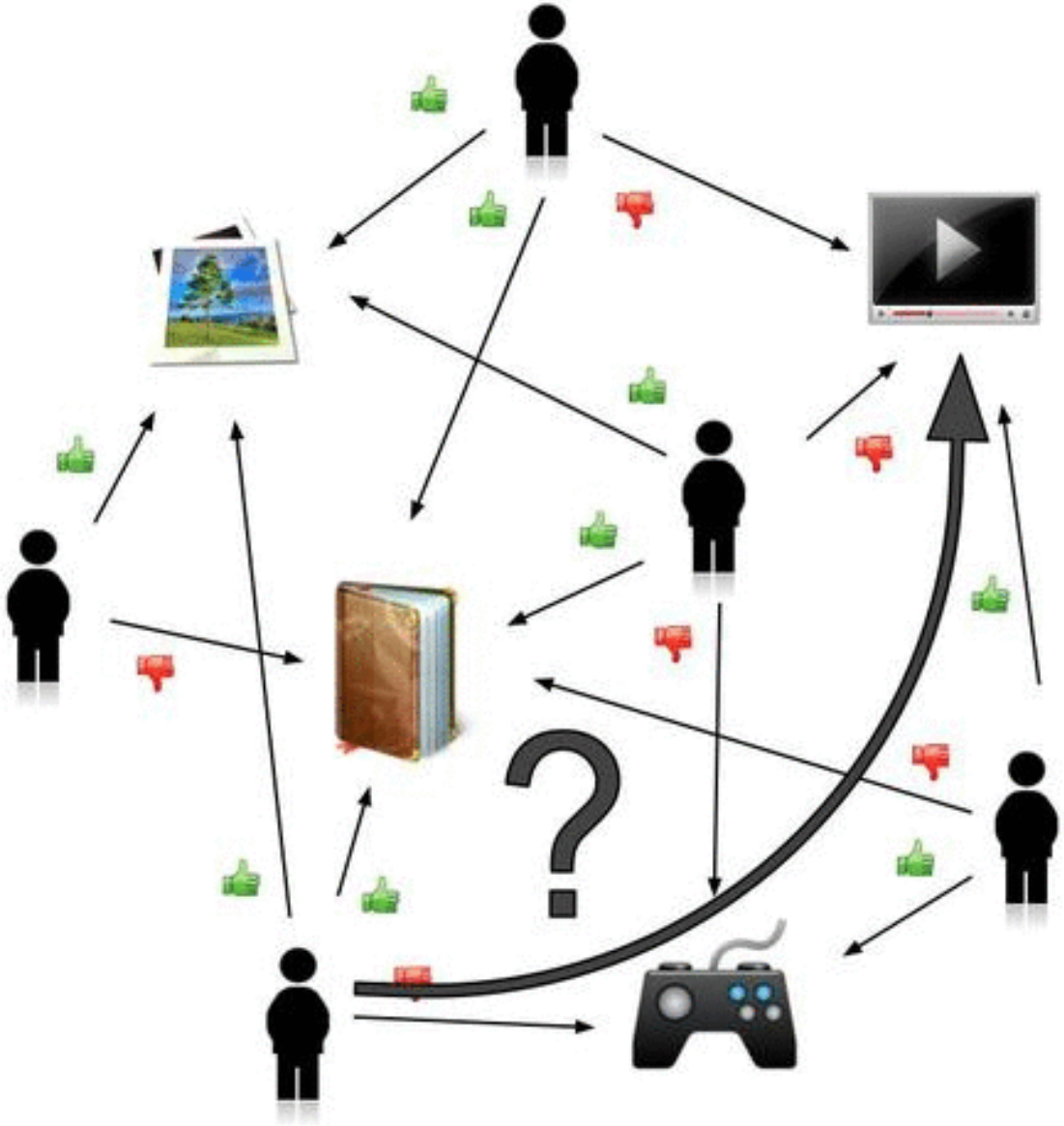
https://en.wikipedia.org/wiki/Collaborative_filtering

Collaborative Filtering



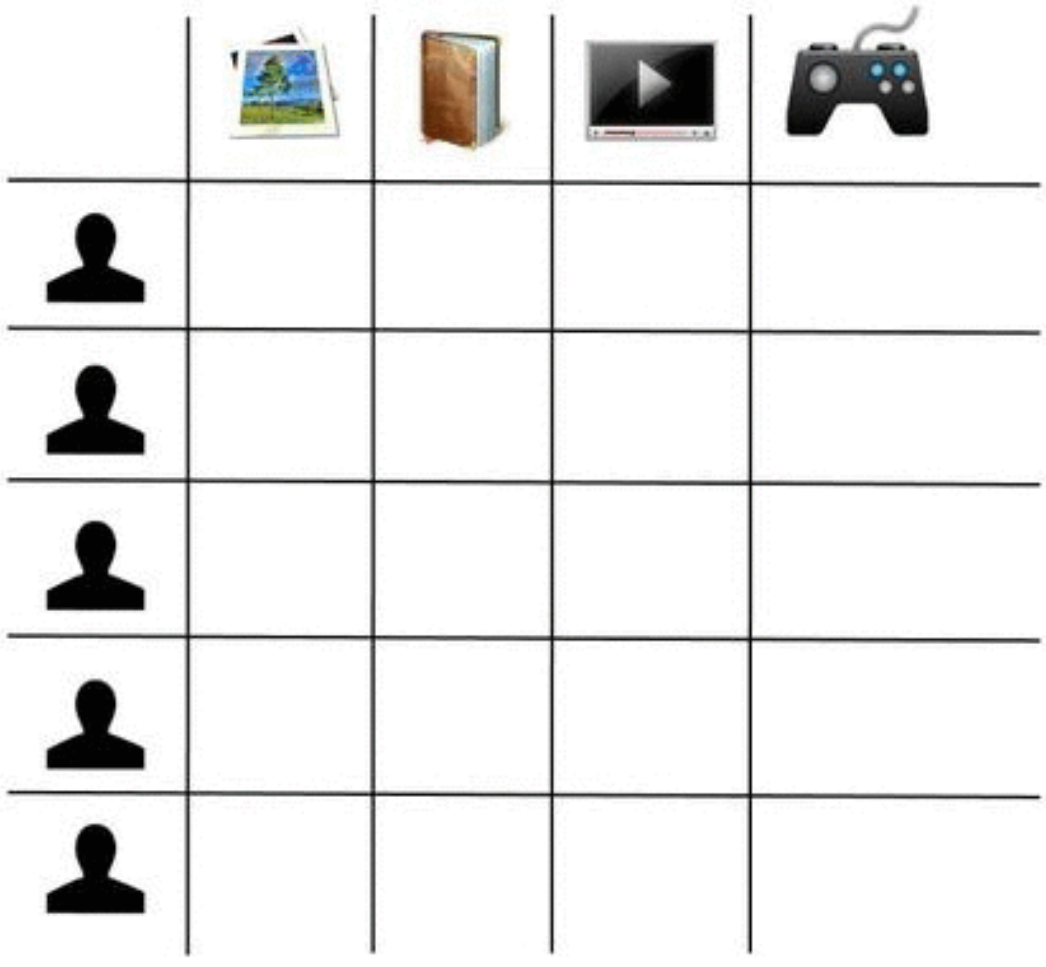
https://en.wikipedia.org/wiki/Collaborative_filtering

Collaborative Filtering



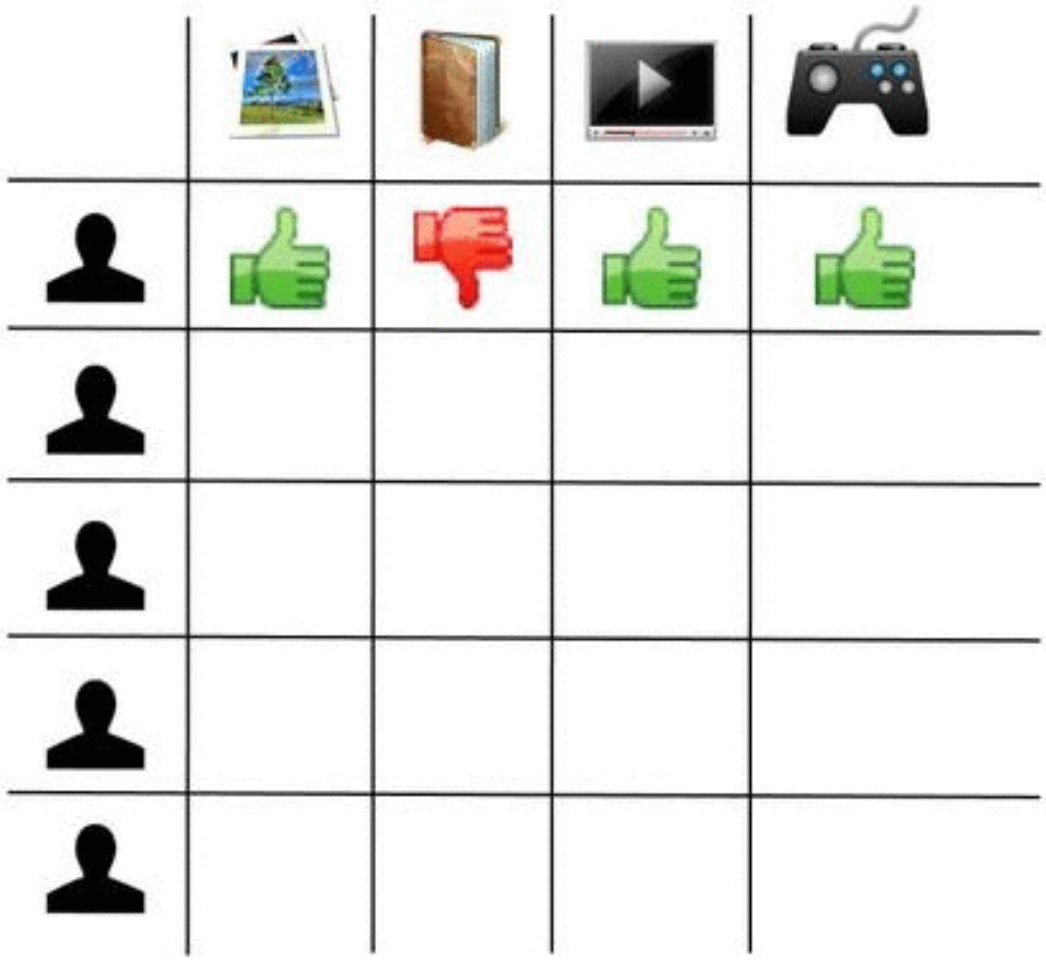
https://en.wikipedia.org/wiki/Collaborative_filtering

Collaborative Filtering



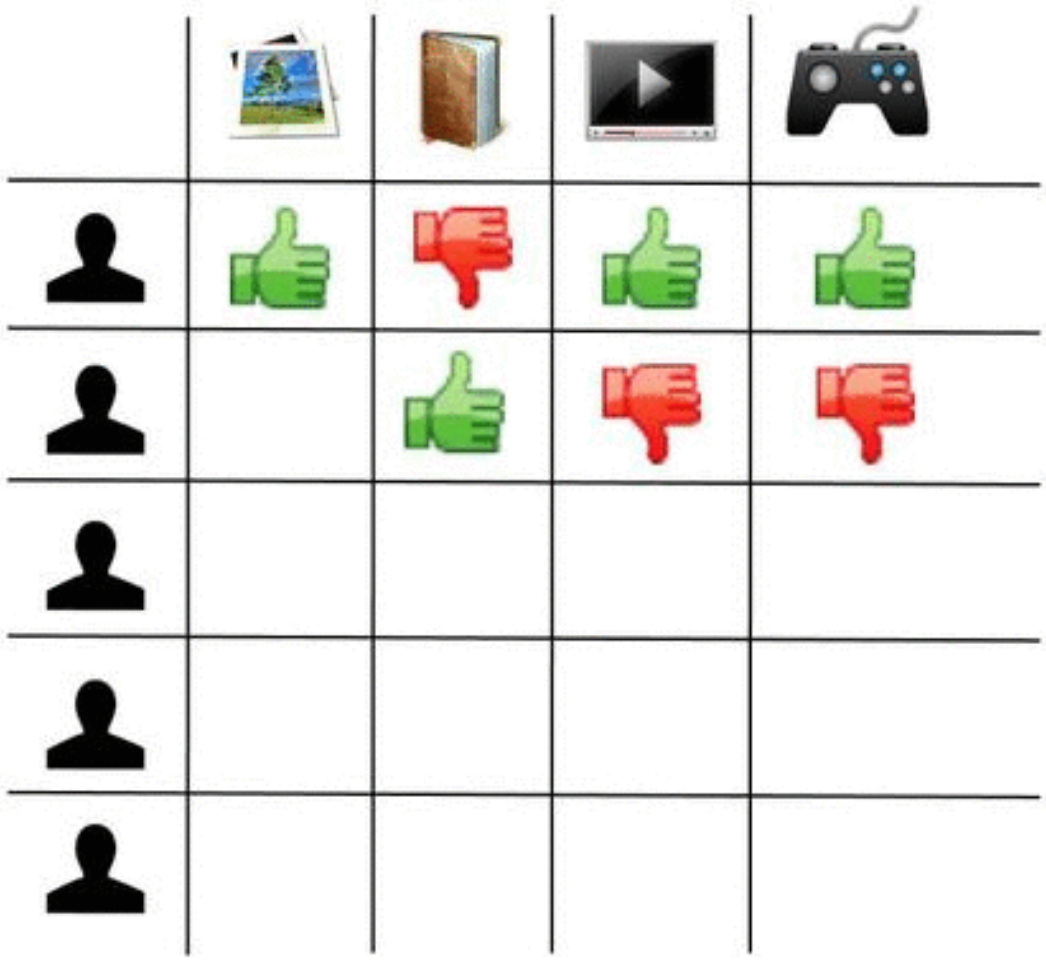
https://en.wikipedia.org/wiki/Collaborative_filtering

Collaborative Filtering



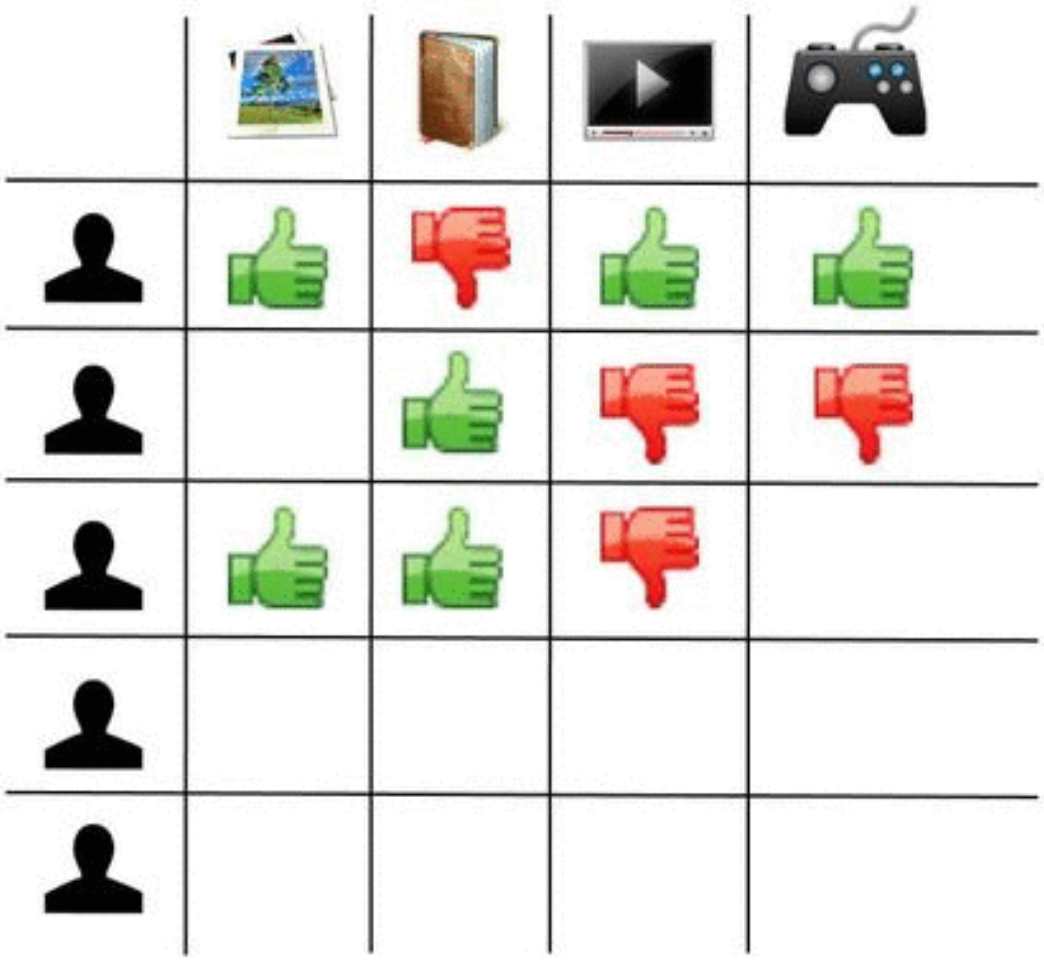
https://en.wikipedia.org/wiki/Collaborative_filtering

Collaborative Filtering



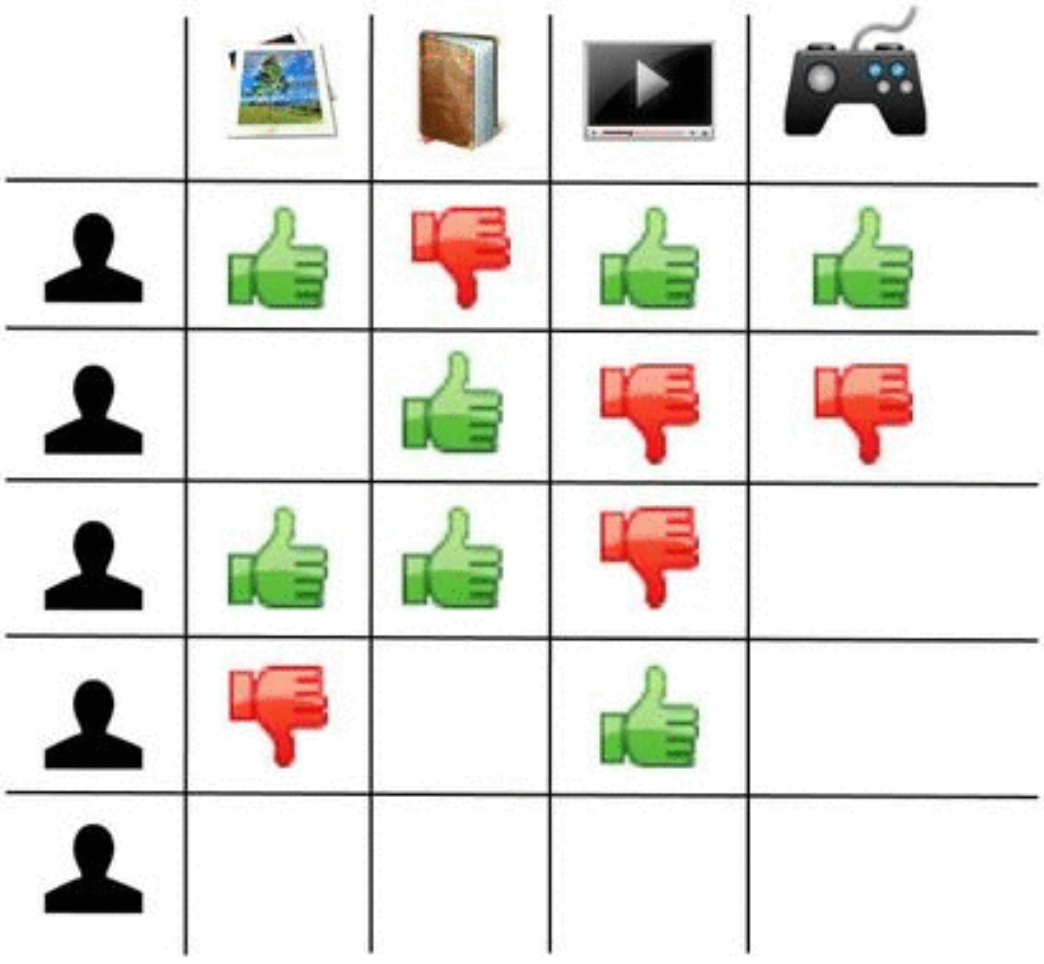
https://en.wikipedia.org/wiki/Collaborative_filtering

Collaborative Filtering



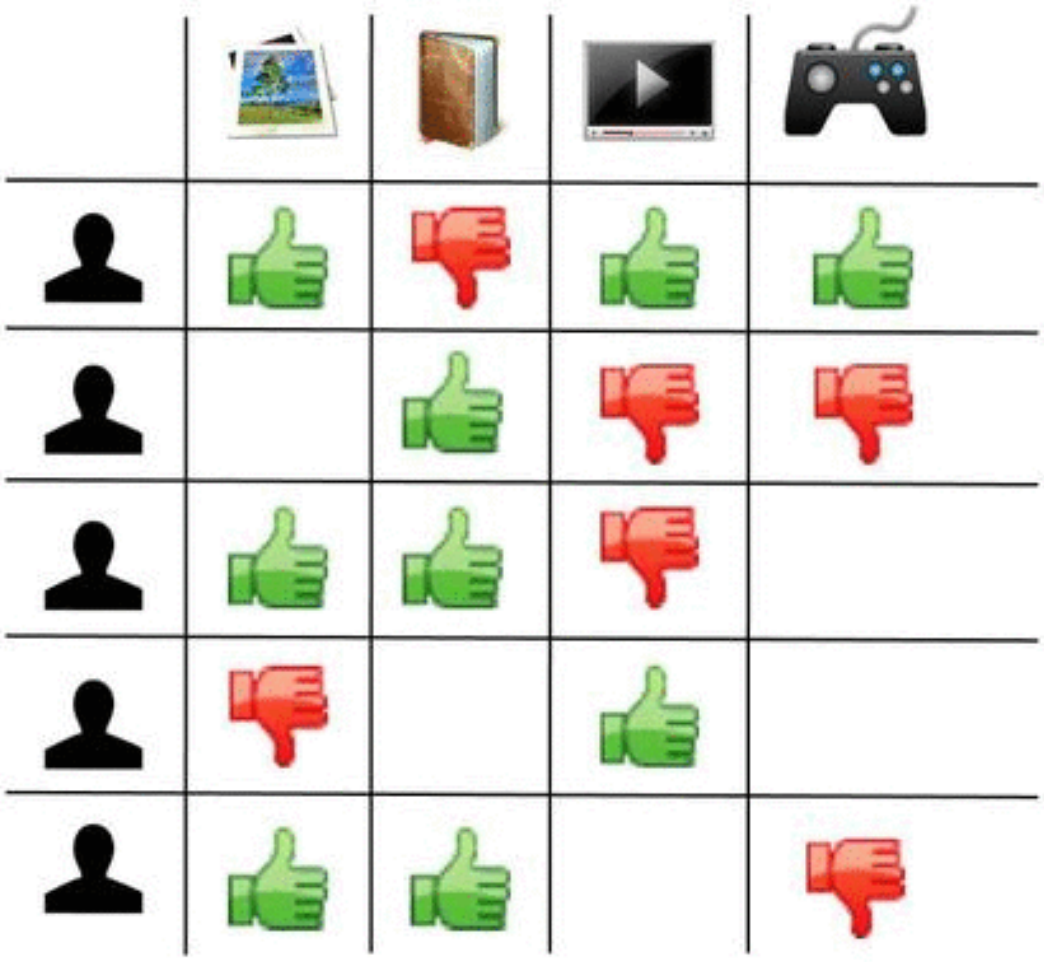
https://en.wikipedia.org/wiki/Collaborative_filtering

Collaborative Filtering



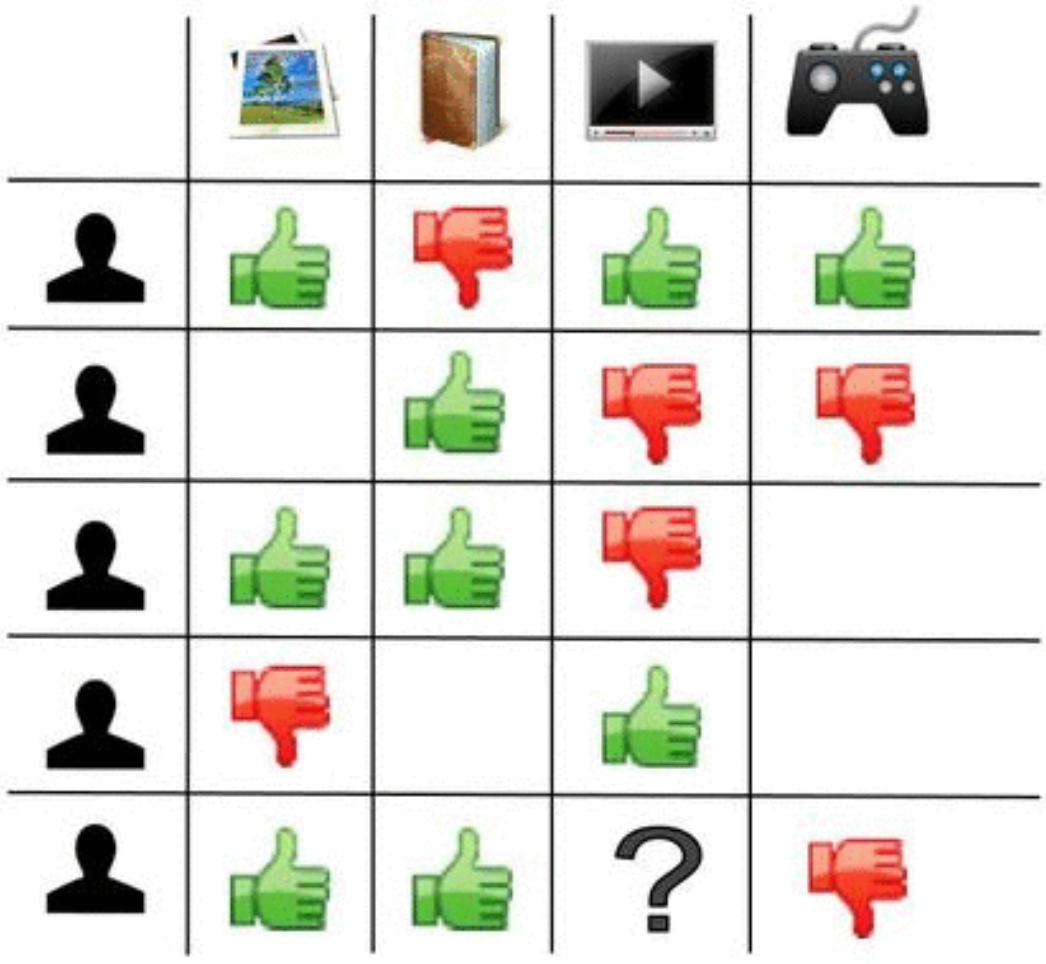
https://en.wikipedia.org/wiki/Collaborative_filtering

Collaborative Filtering



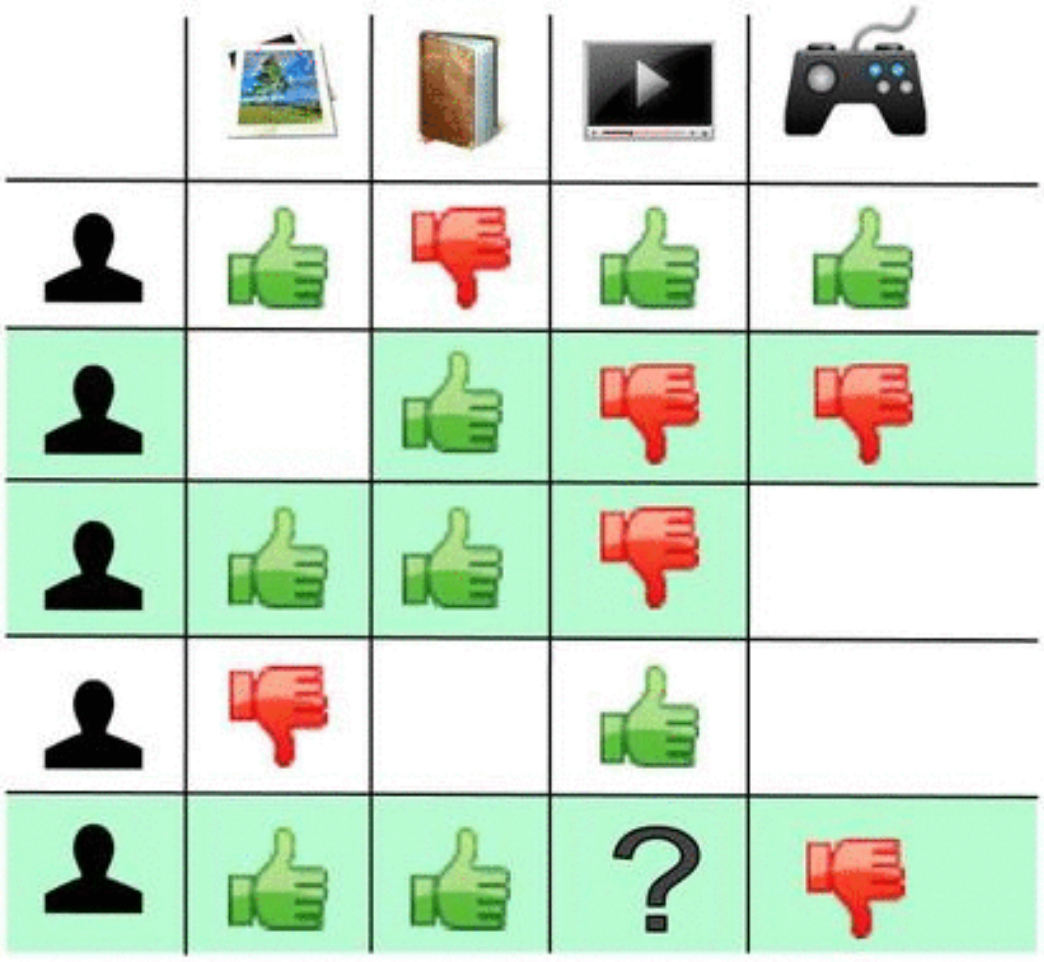
https://en.wikipedia.org/wiki/Collaborative_filtering

Collaborative Filtering



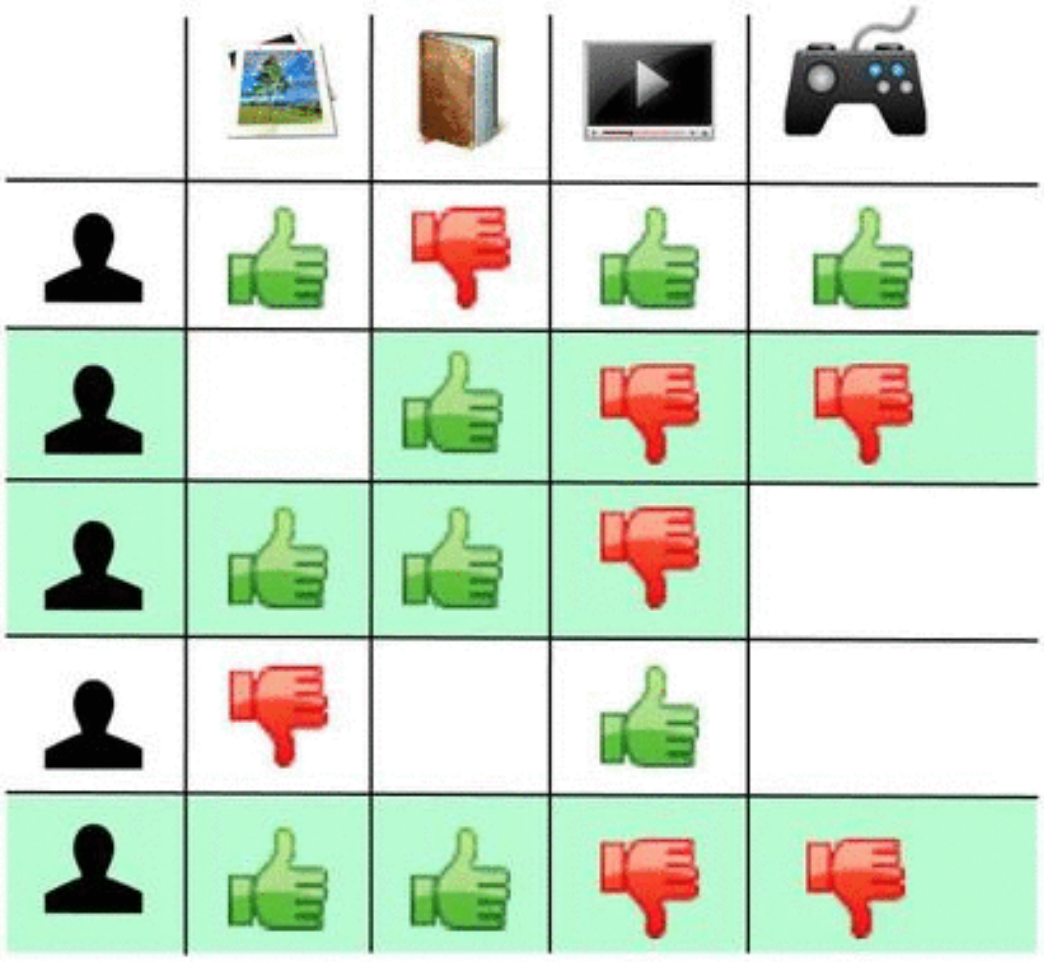
https://en.wikipedia.org/wiki/Collaborative_filtering

Collaborative Filtering



https://en.wikipedia.org/wiki/Collaborative_filtering

Collaborative Filtering

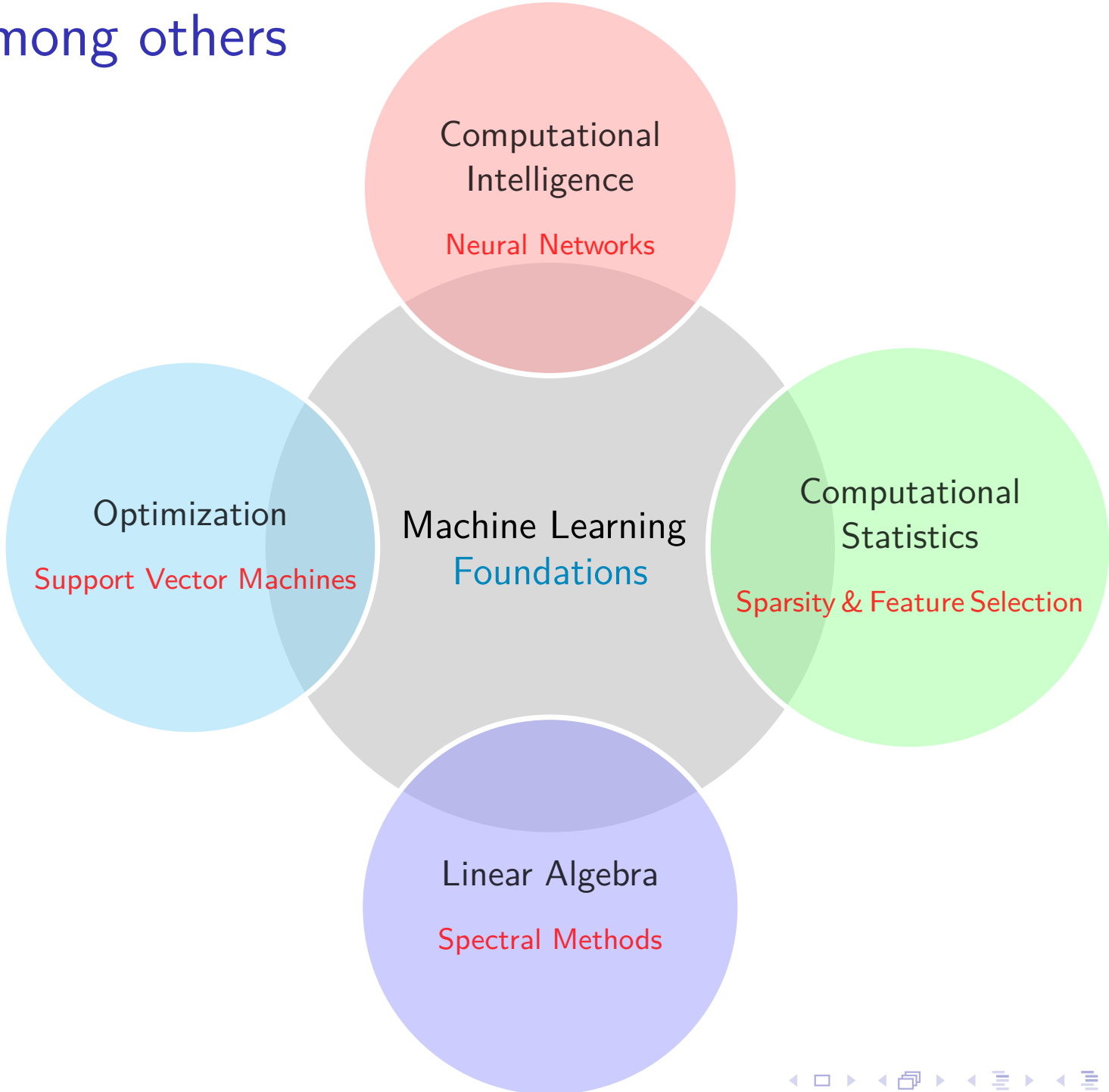


https://en.wikipedia.org/wiki/Collaborative_filtering

Collaborative Filtering

https://en.wikipedia.org/wiki/Collaborative_filtering

. . . , among others



. . . , among others

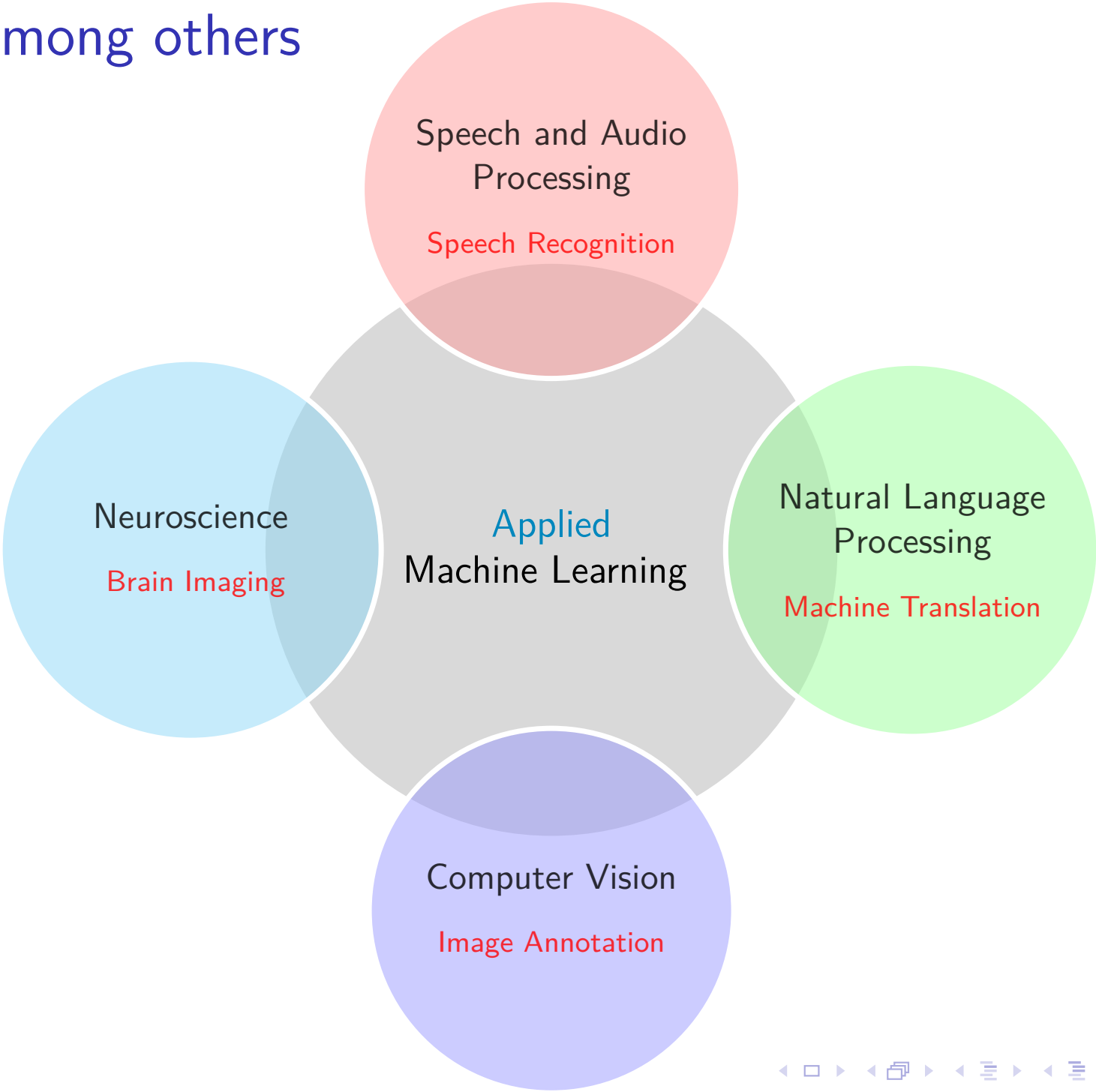
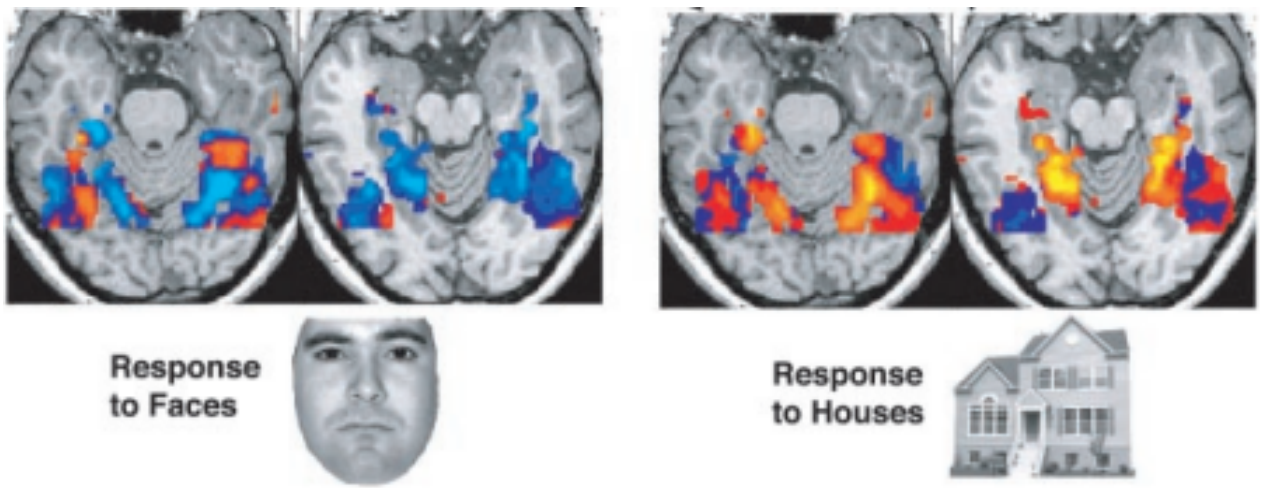


Image Annotation (from L. Ralaivola)

(from Farabet et al, 2013)



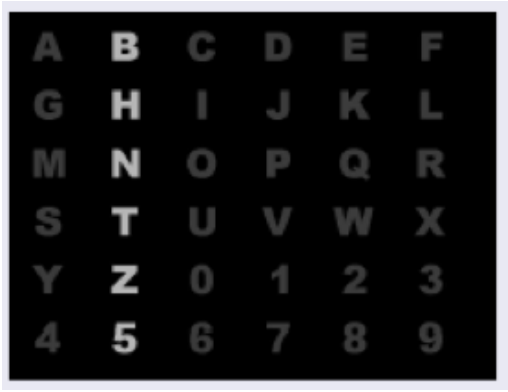
(from Haxby et al, 2001)

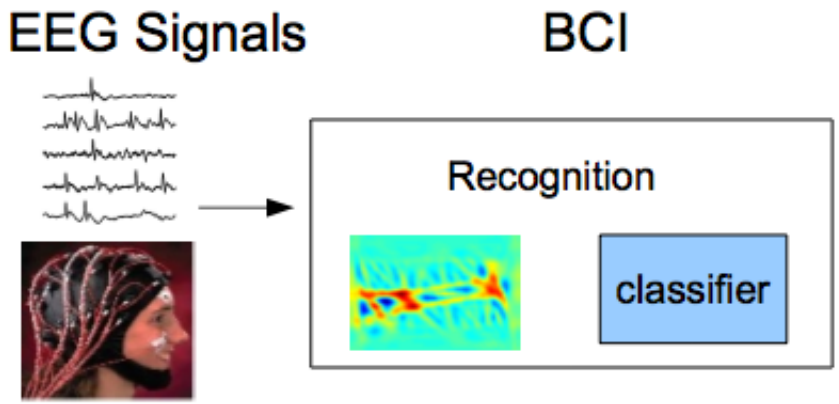
P300 Speller (from L. Ralaivola)

Vintage P300 Speller

(from *Breaking bad*)

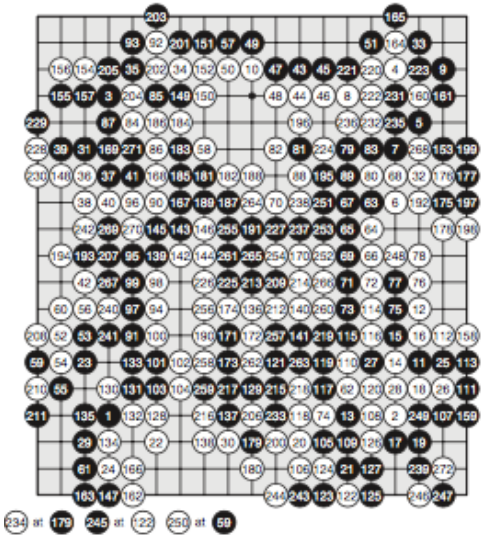
Modern P300 Speller (from A. Rakotomamonjy)



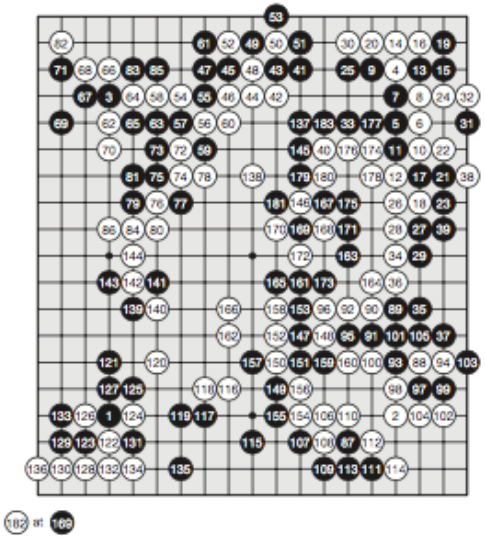


AlphaGo (Silver et al. 2016)

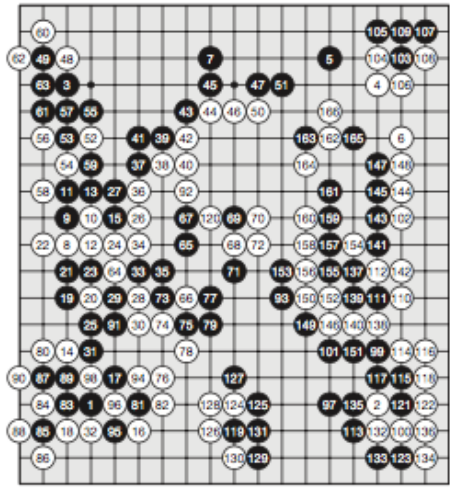
Game 1
Fan Hui (Black), AlphaGo (White)
AlphaGo wins by 2.5 points



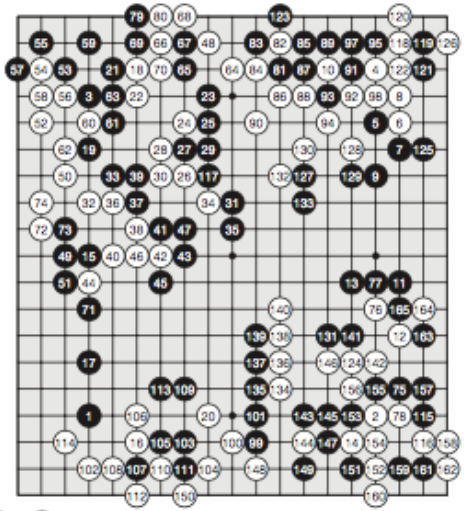
Game 2
AlphaGo (Black), Fan Hui (White)
AlphaGo wins by resignation



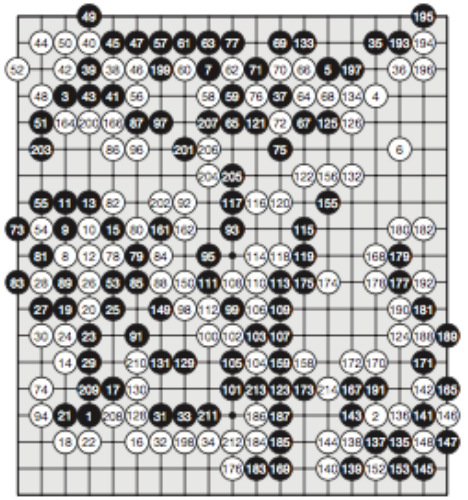
Game 3
Fan Hui (Black), AlphaGo (White)
AlphaGo wins by resignation



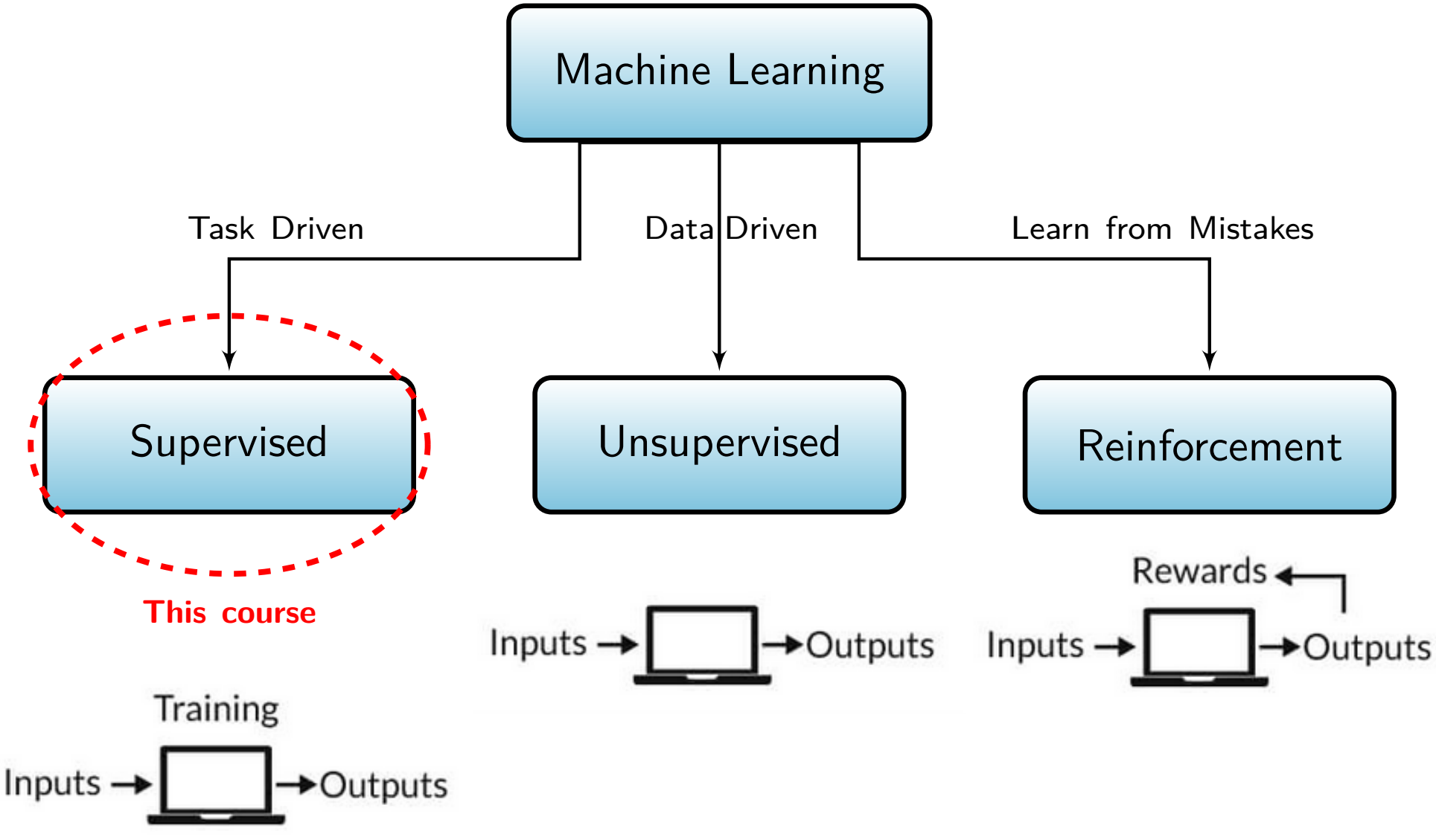
Game 4
AlphaGo (Black), Fan Hui (White)
AlphaGo wins by resignation



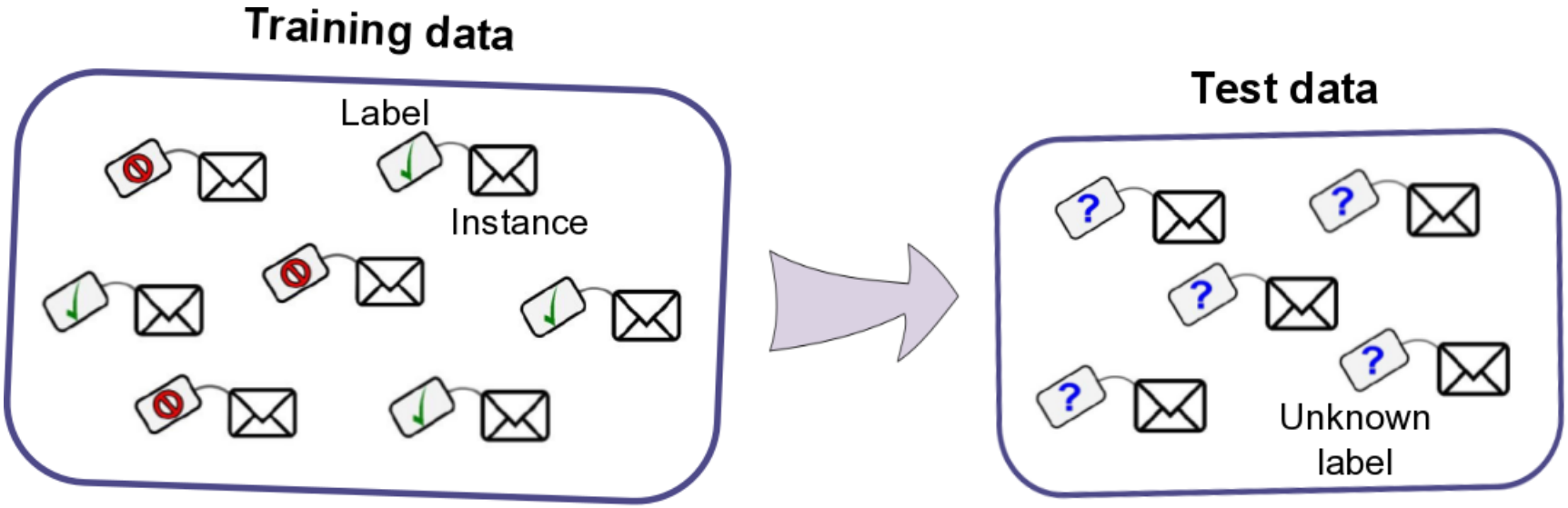
Game 5
Fan Hui (Black), AlphaGo (White)
AlphaGo wins by resignation



Main Types of Learning



Supervised Learning



Generalize from training to testing

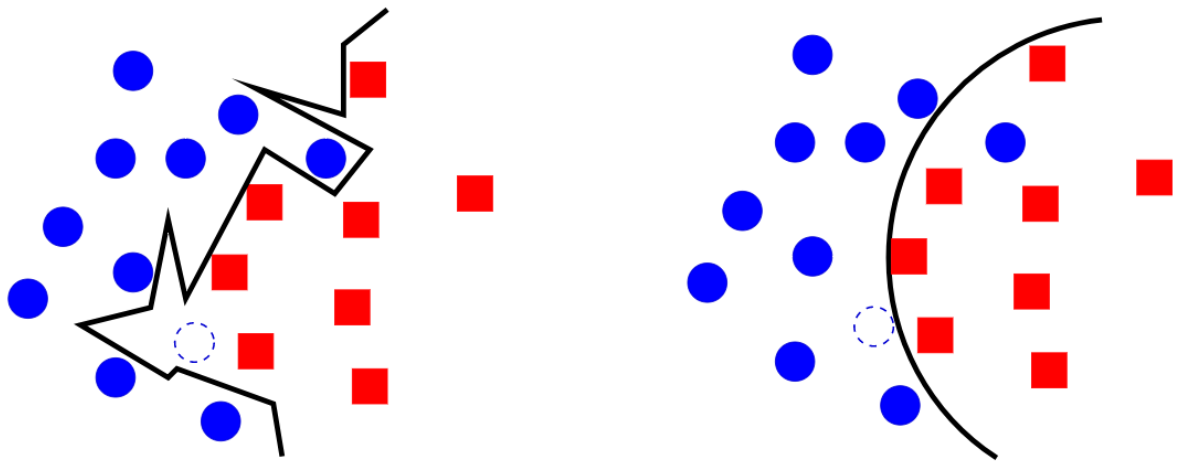
Supervised Learning – Generalization

From a training set consisting of randomly sampled pairs of (input, target), learn a function or a predictor which predicts well the target of a new data.

Supervised learning / Generalization

$\longrightarrow$ Given l training examples $(\mathbf{x_1}, \mathbf{y_1}), \ldots, (\mathbf{x_l}, \mathbf{y_l}) \in (\mathcal{X} \times \mathcal{Y})$ and u test data $\mathbf{x_{l+1}}, \ldots, \mathbf{x_{l+u}} \in \mathcal{X}$

$\longrightarrow$ Learn $f : \mathcal{X} \to \mathcal{Y}$ to generalize from training to testing



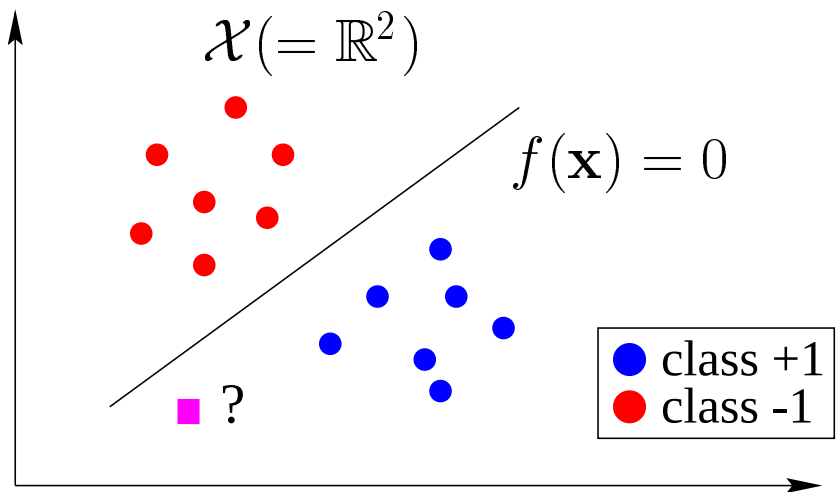
Focus: binary classification

Important notions in *learning to classify*

- a number of *training* data (pictures, emails, etc.)
- learning algorithm (how to build the classifier?)
- generalization: the classifier should correctly classify *test* data

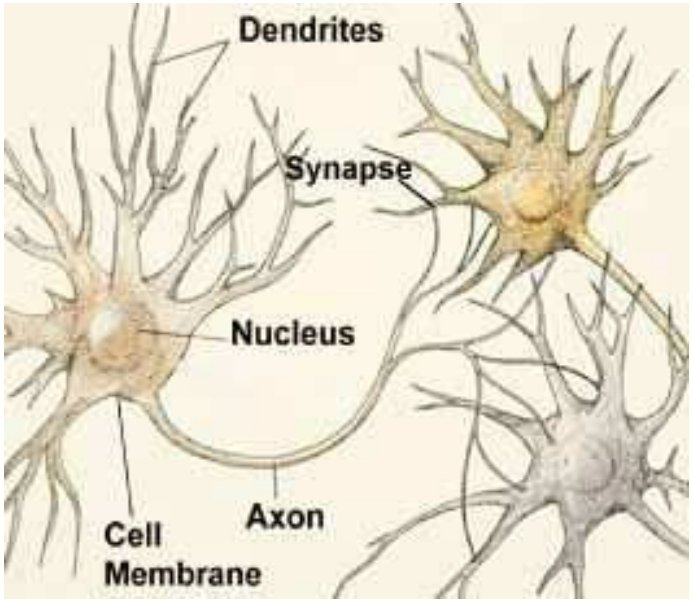
Quick formalization

- $\mathcal{X}$ (e.g. $\mathbb{R}^d, d > 0$) is the space of data, called *input space*
- $\mathcal{Y}$ (e.g. toxic/not toxic, or $\{-1, +1\}$) is the target space
- $f : \mathcal{X} \to \mathcal{Y}$ is the classifier



Perceptron (Rosenblatt, 1958)

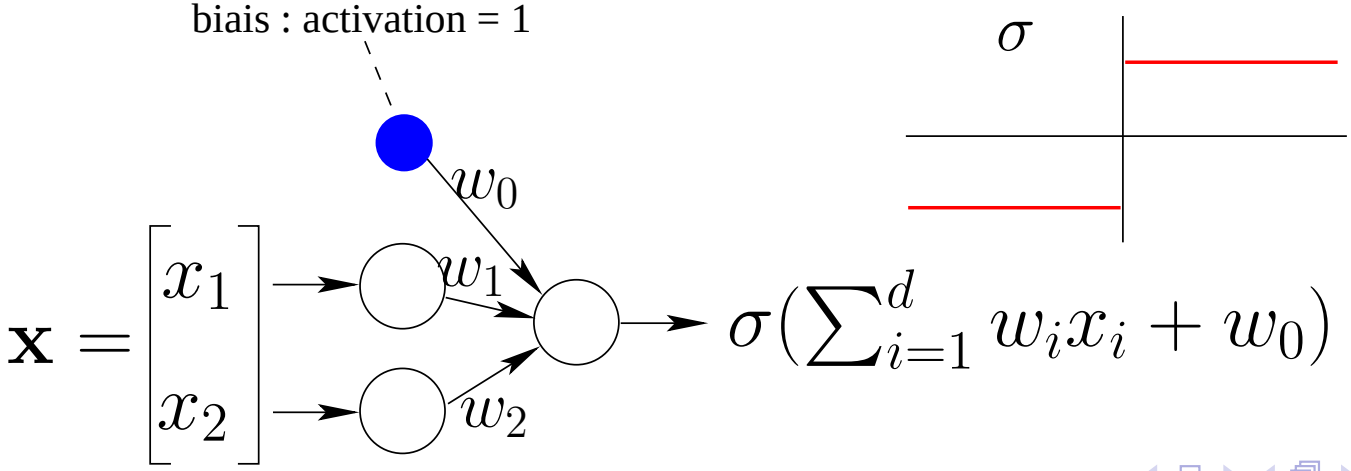
Inspiration: biological neural network



Motivations:

- Learning system composed by associating simple processing units
- Efficiency, scalability, and adaptability

Perceptron: a linear classifier, $\mathcal{X} = \mathbb{R}^d$, $\mathcal{Y} = \{-1, +1\}$



Perceptron (Rosenblatt, 1958)

A linear classifier, $\mathcal{X} = \mathbb{R}^d$, $\mathcal{Y} = \{-1, +1\}$

- Classifier weights: $\mathbf{w} \in \mathbb{R}^d$
- Classifier prediction: $f(\mathbf{x}) = \text{sign}\langle \mathbf{w}, \mathbf{x} \rangle$
- Question: how to learn $\mathbf{w}$ from training data

Algorithm: Perceptron

Input: $S = \{(X_n, Y_n)\}_{n=1}^N$

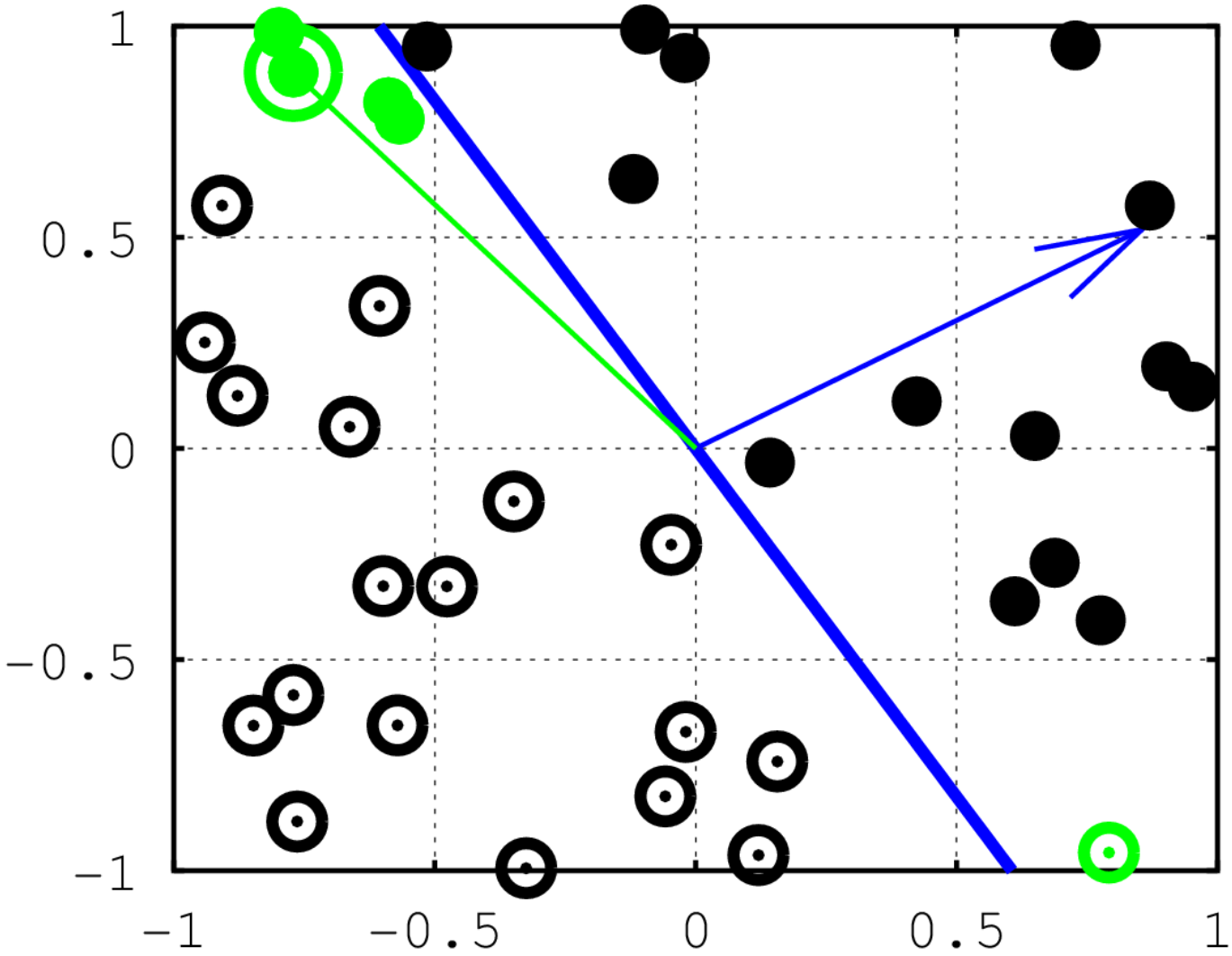
$\mathbf{w} \leftarrow \mathbf{0}$

while it exists (X_n, Y_n): $Y_n \langle \mathbf{w}, X_n \rangle \leq 0$ **do**

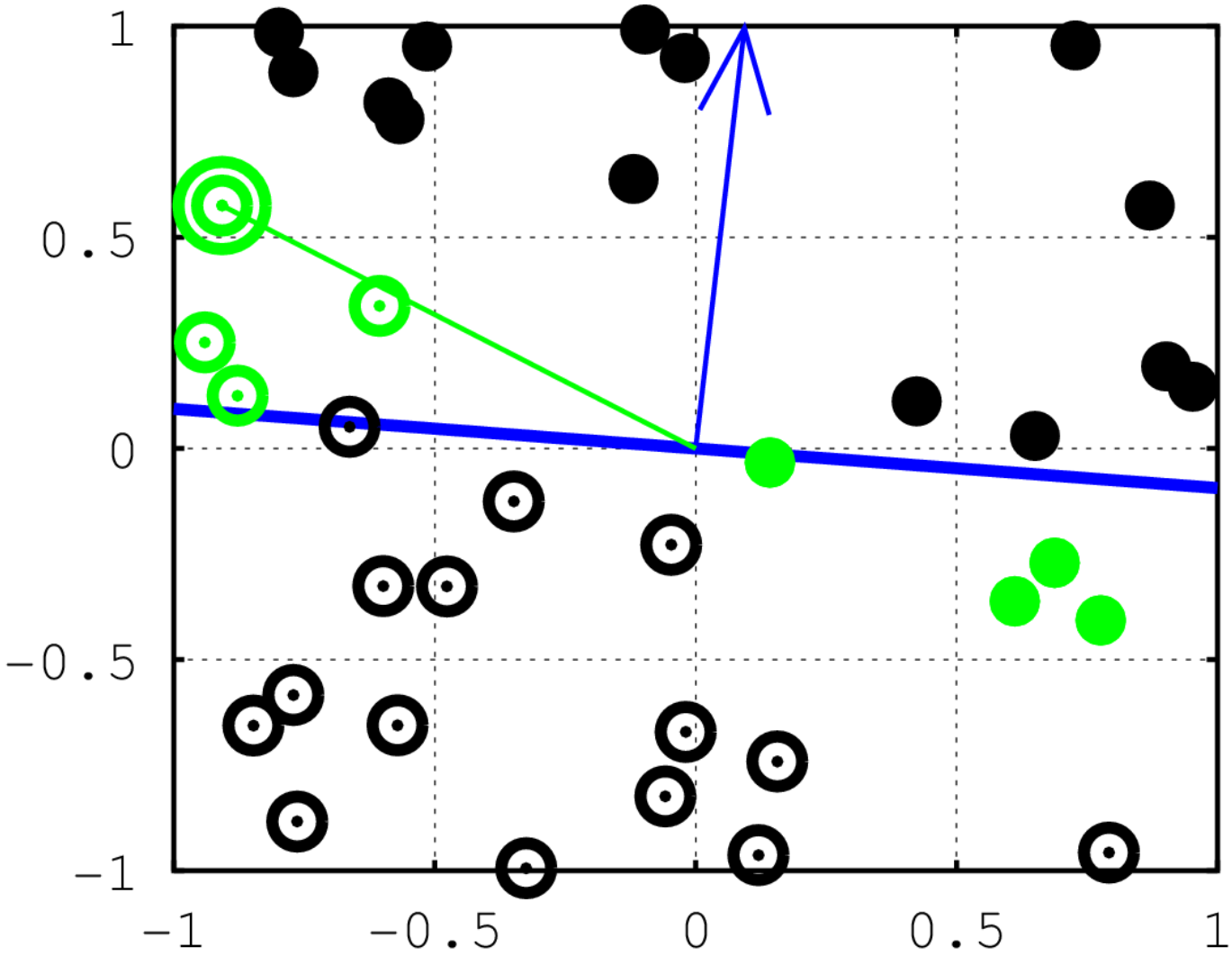
$\mathbf{w} \leftarrow \mathbf{w} + Y_n X_n$

end while

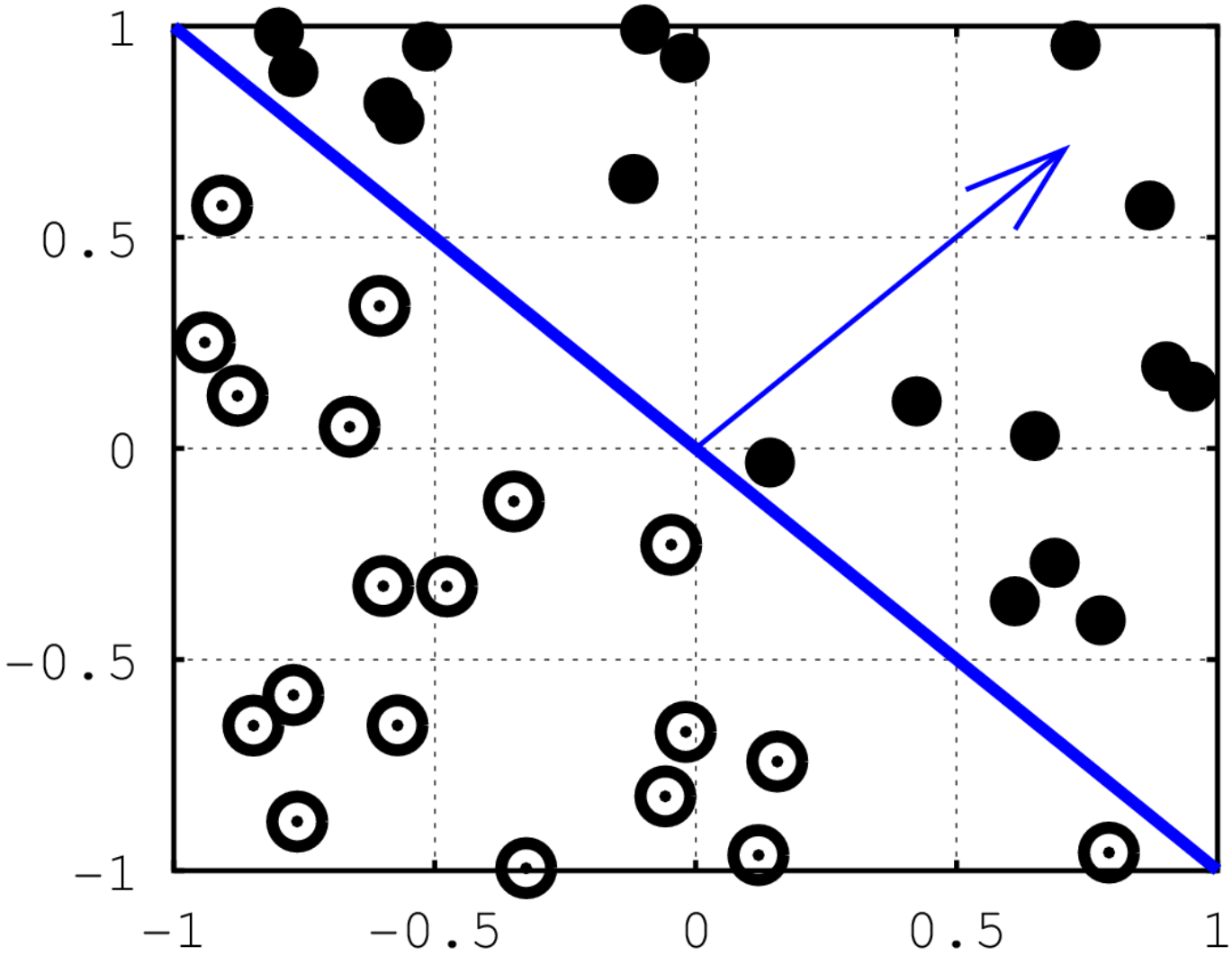
Perceptron in action



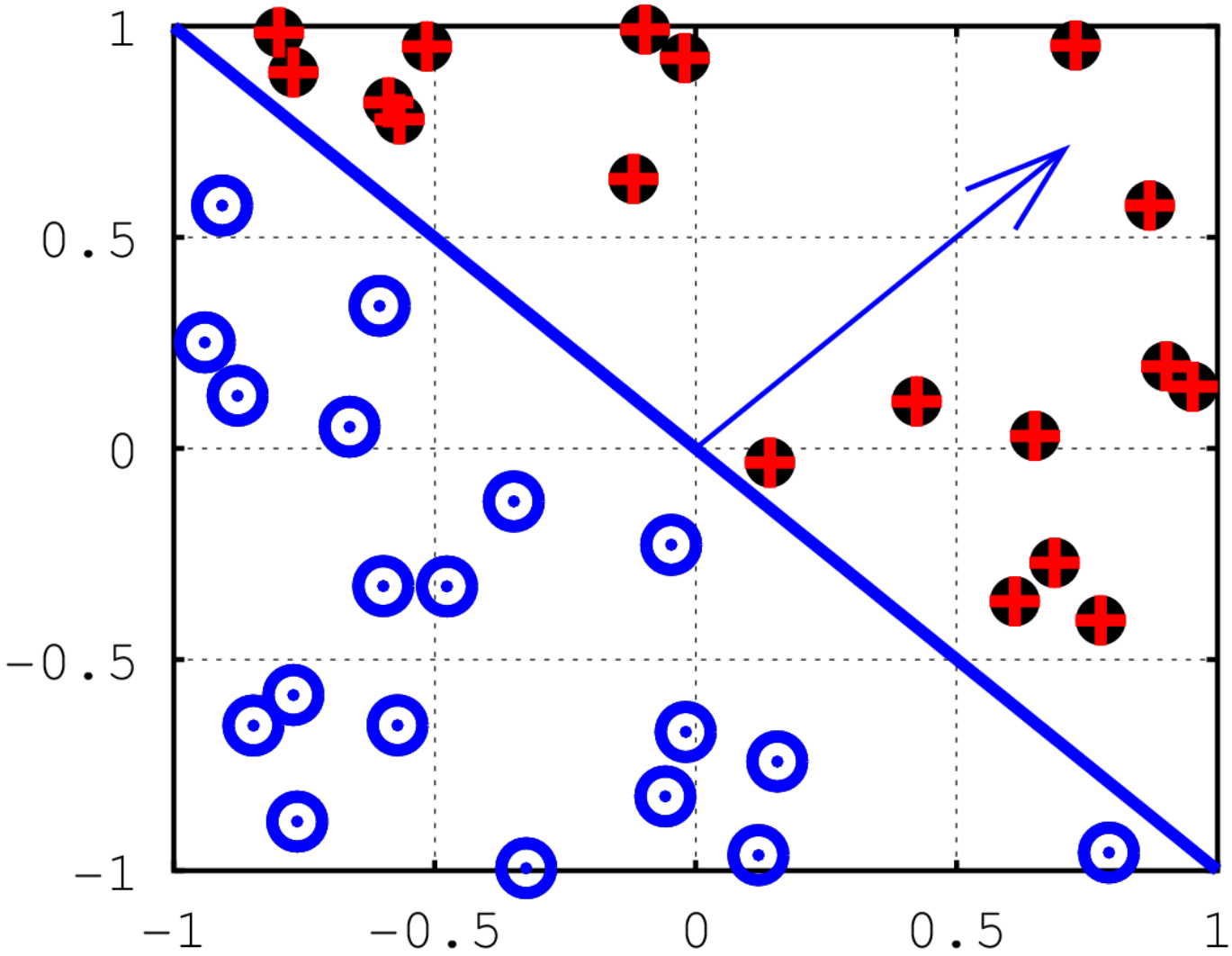
Perceptron in action



Perceptron in action



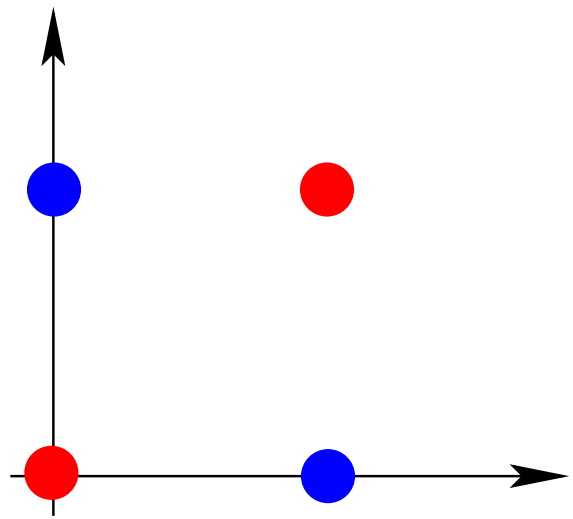
Perceptron in action



Perceptron: limitations

Theorem (XOR, Minsky, Papert, 1969)

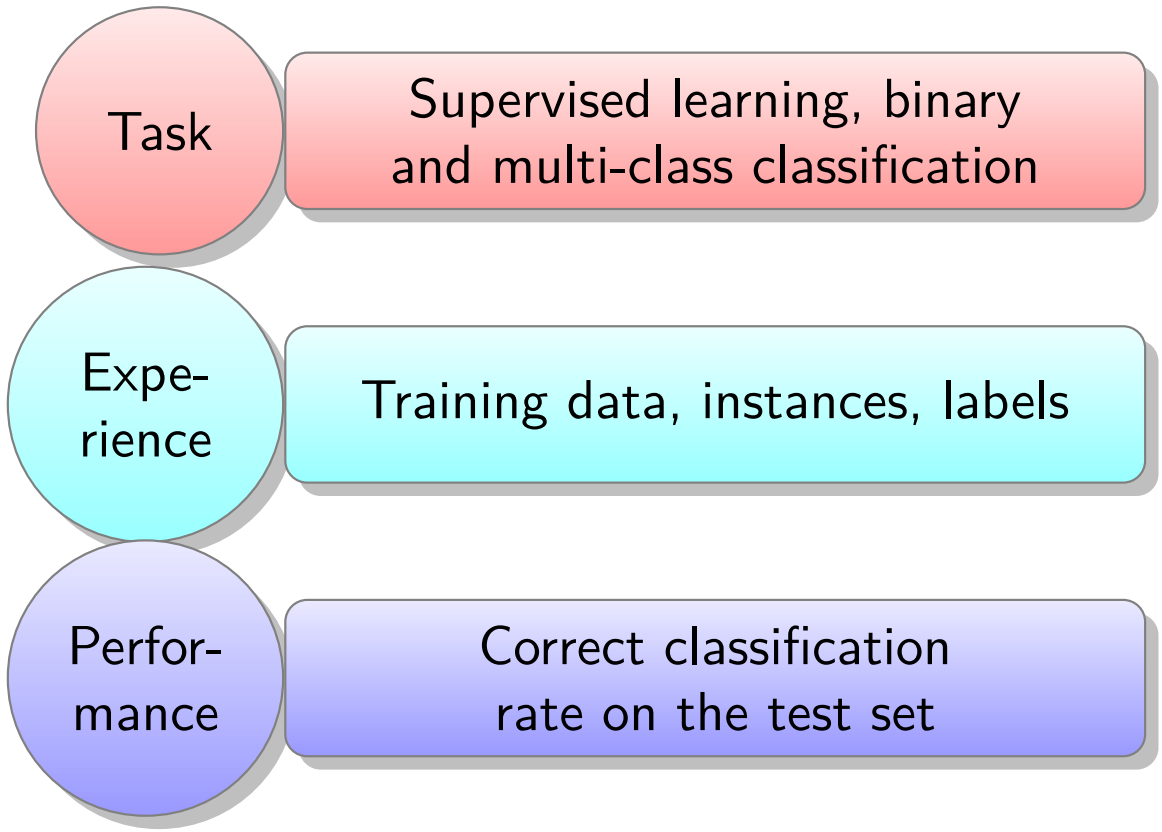
The perceptron algorithm cannot solve the XOR problem



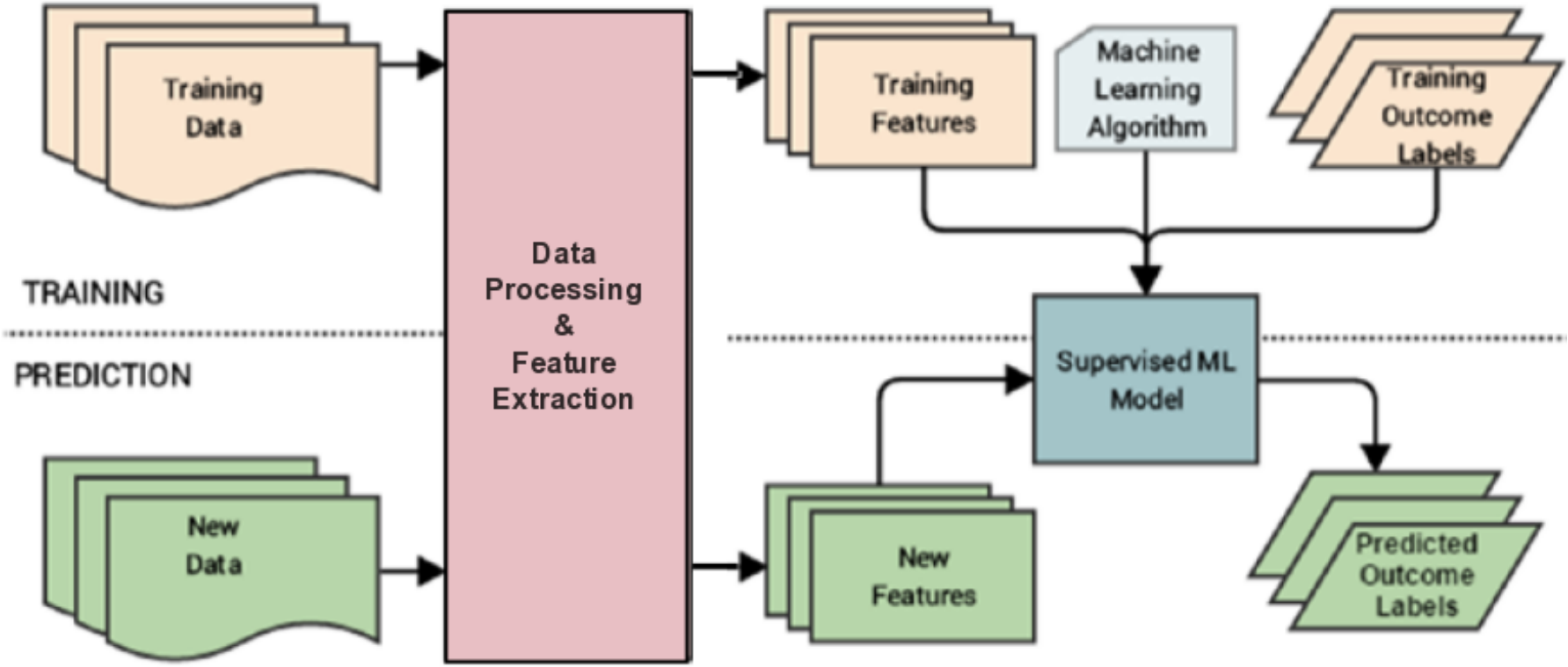
How to address this issue? See Part III

In the following . . .

"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"

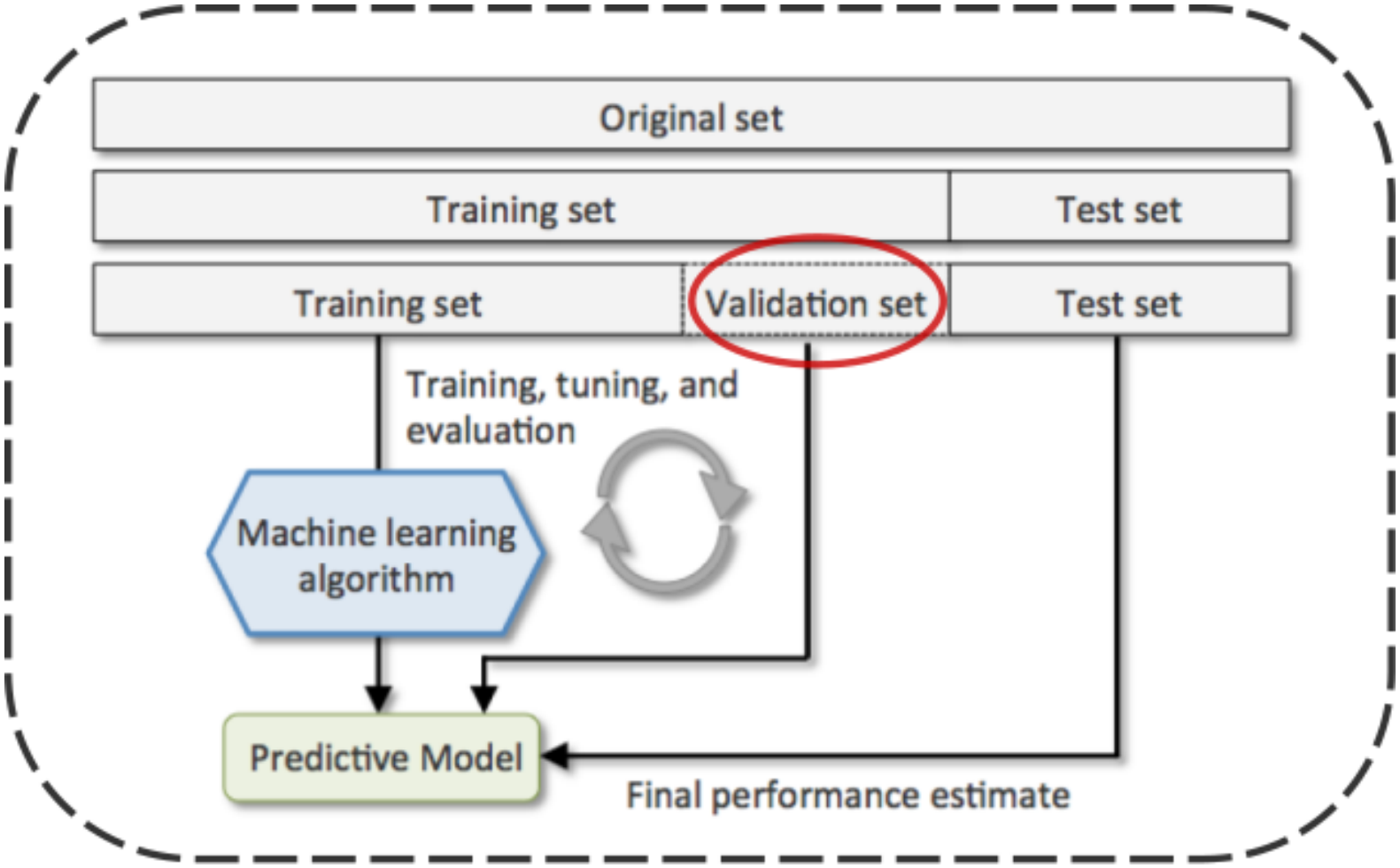


Supervised Machine Learning Pipeline



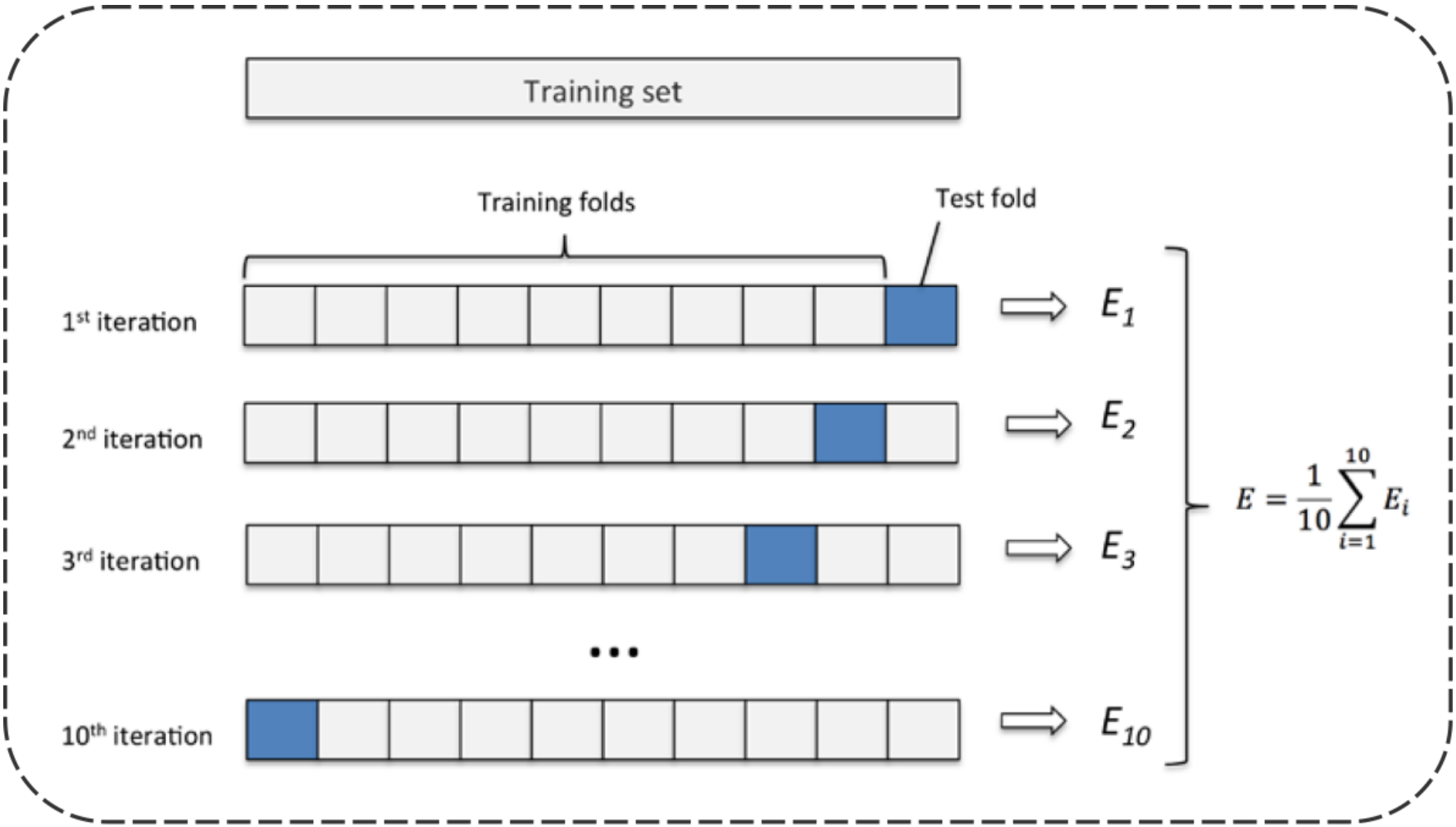
(from Sarkar et al., 2018)

Validation Set



(from Raschka, 2017)

k-fold Cross-validation (k=10)



(from Raschka, 2017)

Act II: Training Machine Learning Algorithms for Classification and Regression

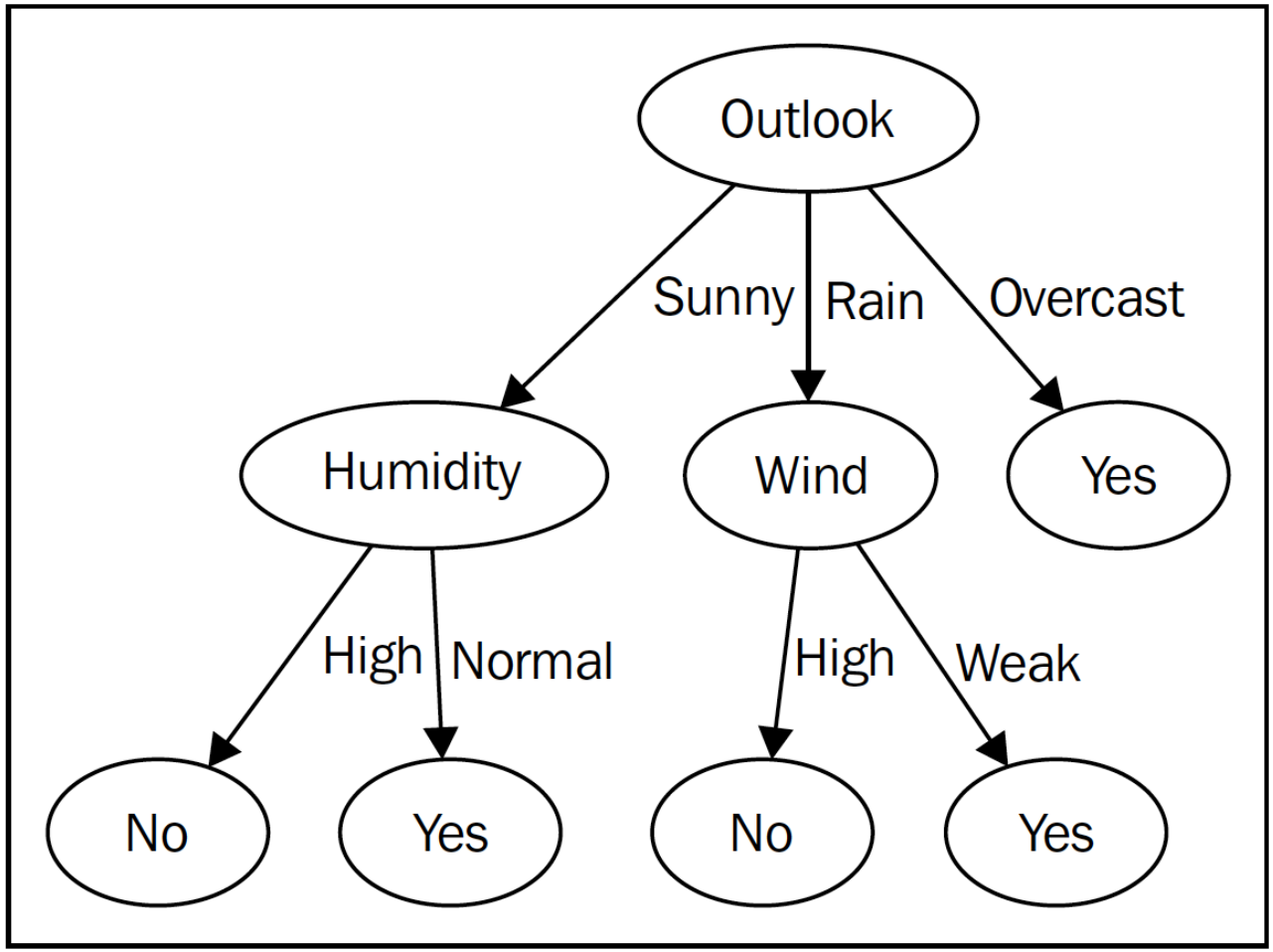
Decision Trees

Should we play tennis or not?

Play	**Wind**	**Humidity**	**Outlook**
No	Low	High	Sunny
No	High	Normal	Rain
Yes	Low	High	Overcast
Yes	Weak	Normal	Rain
Yes	Low	Normal	Sunny
Yes	Low	Normal	Overcast
Yes	High	Normal	Sunny

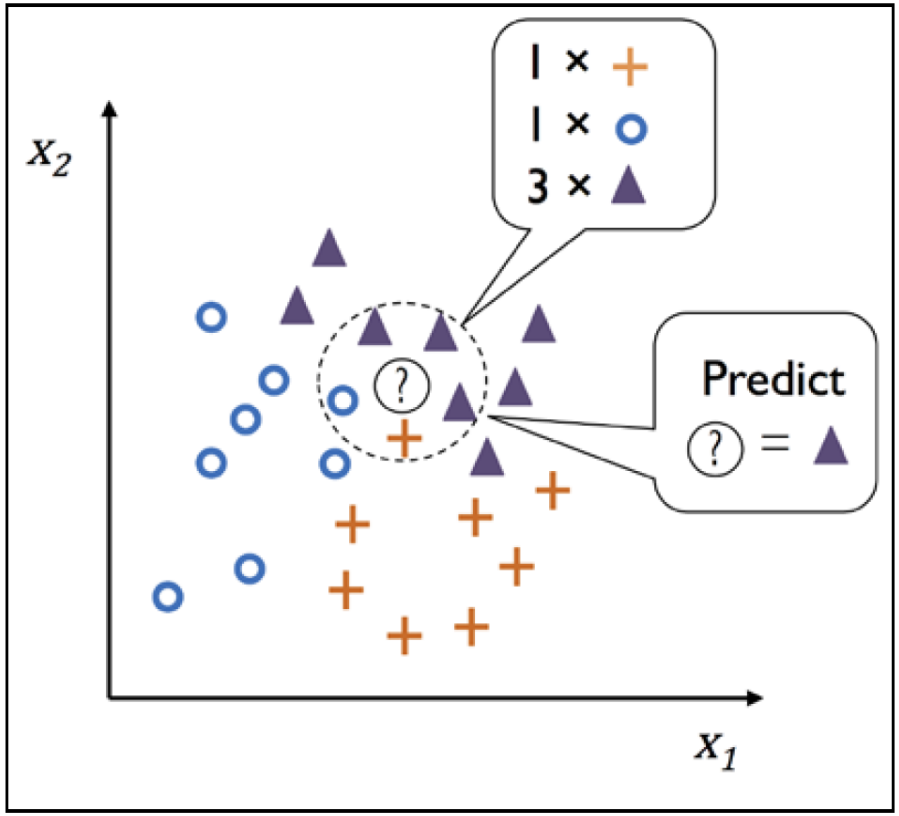
Decision Trees

- breaking down our data by making decision based on asking a series of questions



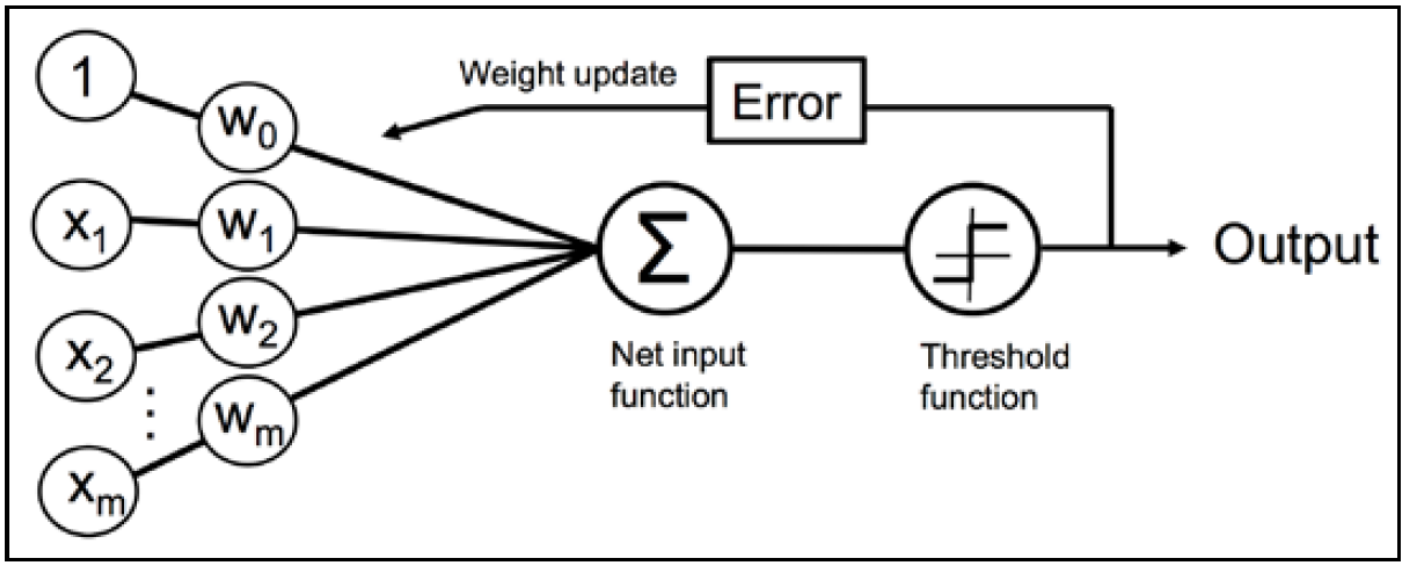
K-nearest neighbors

- Choose the number of k and a distance metric
- Find the k-nearest neighbors of the sample to be classified
- Assign the class label by majority vote



Perceptron

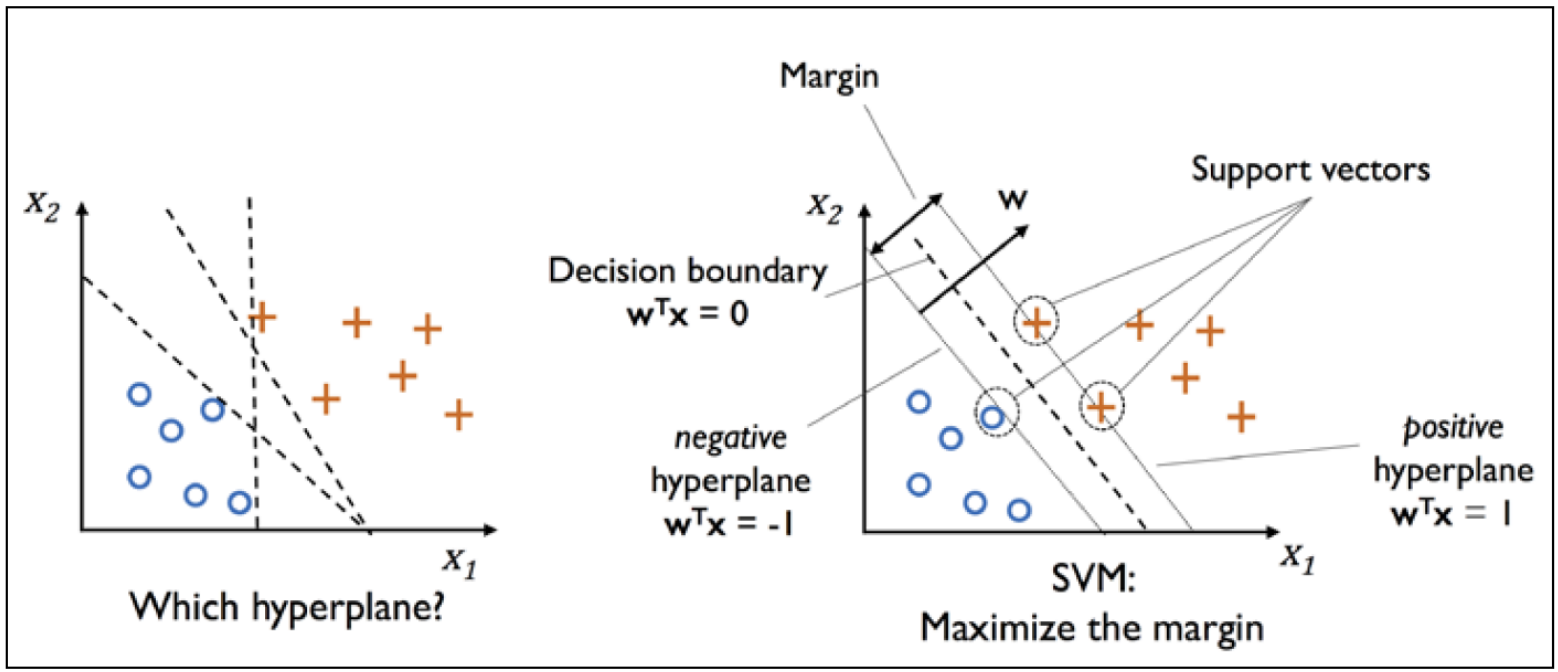
1. Initialize the weights $\mathbf{w}$ to 0 or small random numbers
2. For each training sample X_n :
   a. Compute the output value $\hat{Y}_n \triangleq sign(\mathbf{w}^\top X_n)$
   b. Update the weights if $\hat{Y}_n \neq Y_n$,



Support Vector Machines (SVM)

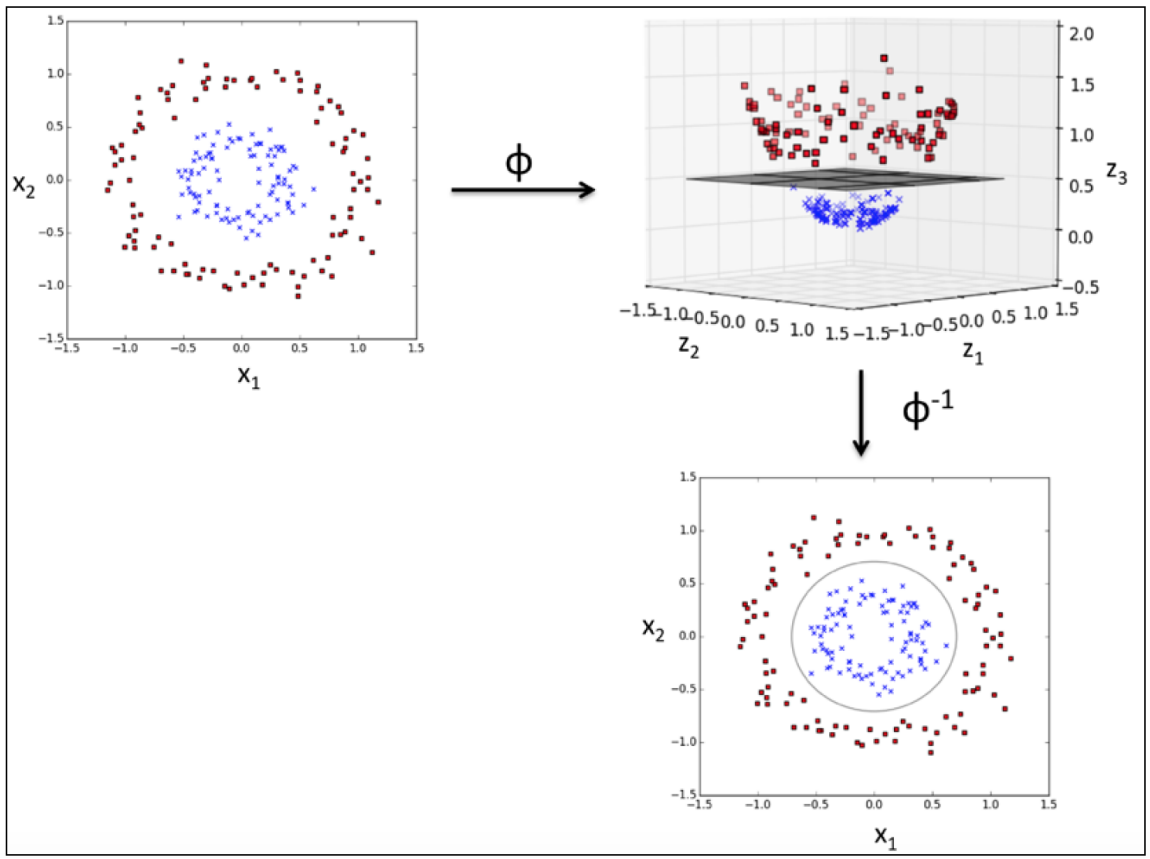
- Maximize the margin which is equal to $\frac{2}{\|\mathbf{w}\|}$
- under the constraint that the samples are classified correctly:
  a. $\mathbf{w}^\top X_n \geq 1$ if $Y_n = 1$
  b. $\mathbf{w}^\top X_n \leq -1$ if $Y_n = -1$

$\forall n = 1, \ldots, N$

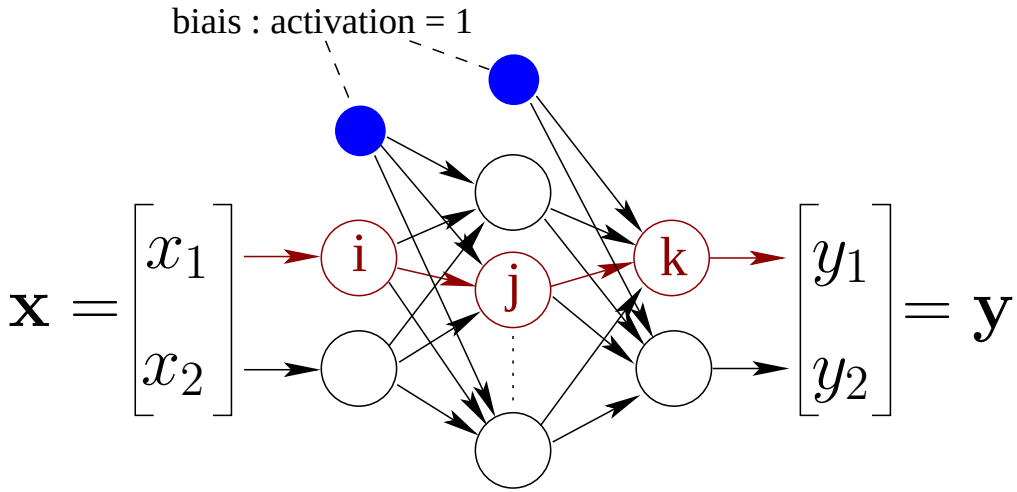


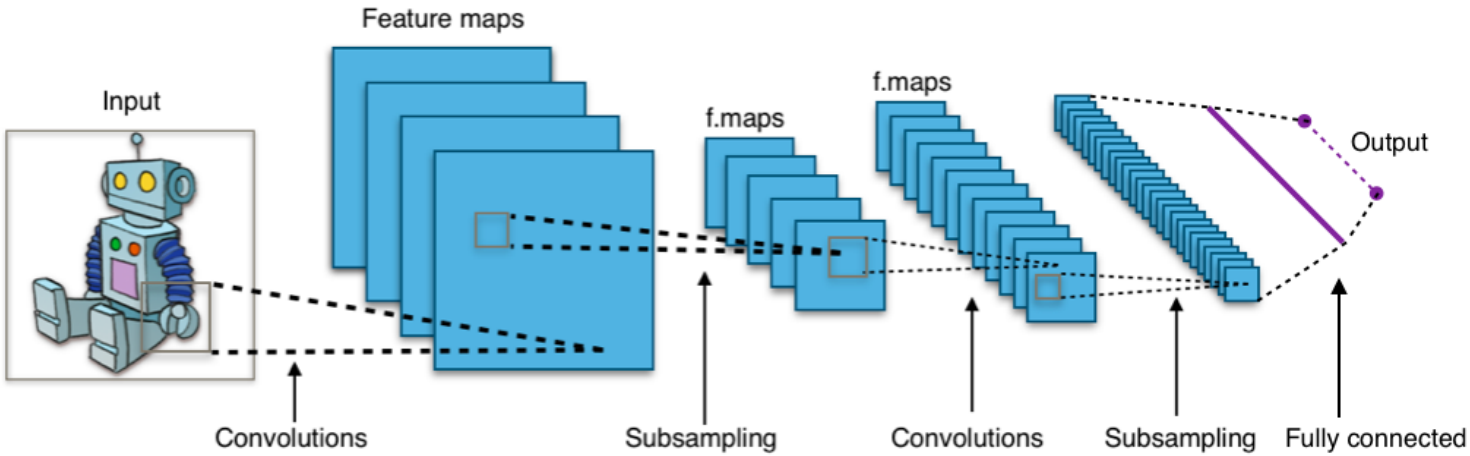
Kernel SVM for linearly inseparable data

- Project the data in a high-dimensional feature space using a kernel function
- Apply a linear classifier in the feature space



Multilayer Perceptron and Convolutional Neural Networks



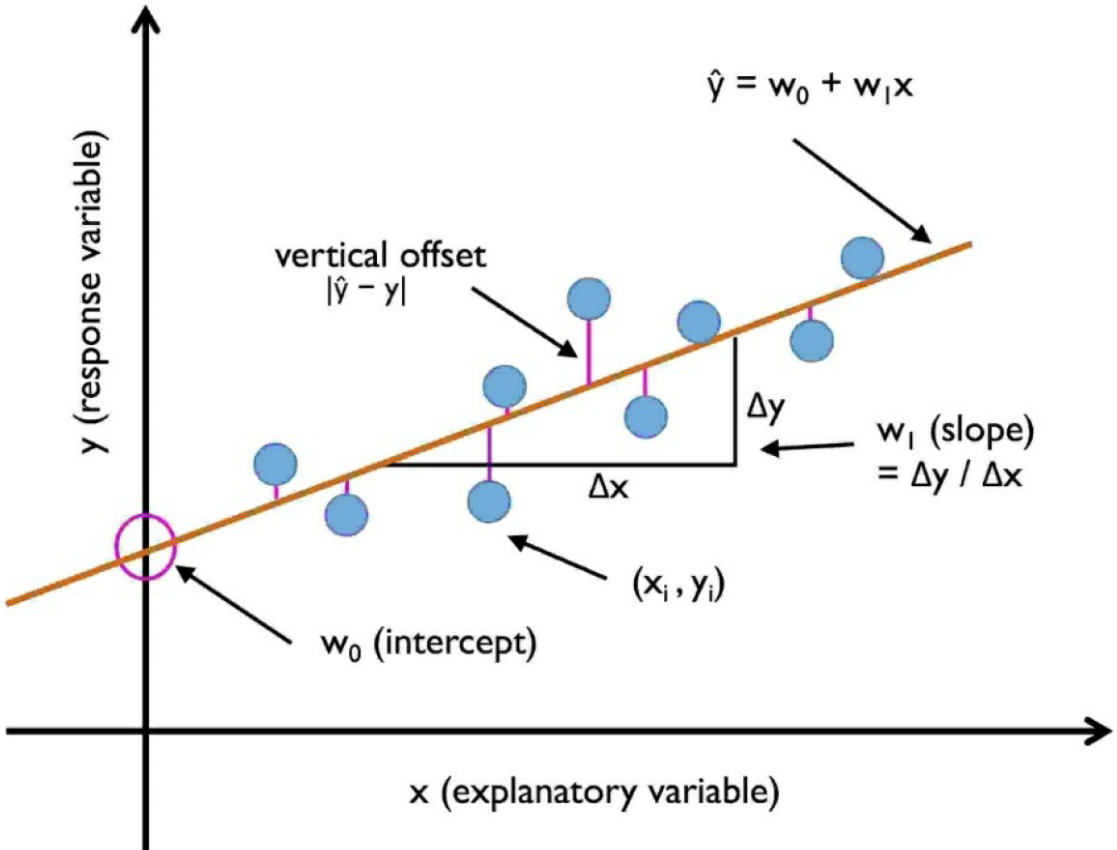


(de By Aphex34 - Own work, CC BY-SA 4.0, https://commons.wikimedia.org/w/index.php?curid=45679374)

Linear Regression

- model the relation between features (explanatory variable x) and a continuous valued response (target variable y)
- Linear model:

$$y = w_0 + w_1 x^{(1)} + \ldots + w_d x^{(d)} = \sum_{i=1}^{d} w_i x^{(i)} = \langle w, x \rangle$$



. . . How these work in practice

See the notebook

`ML_5GSC2019.ipynb`

Act III: Programming with Python for Machine Learning

Packages for Scientific Computing and Machine Learning

Throughout this course, we will mainly use

- NumPy: multidimensional array package
- SciPy: scientific computing package
- Matplotlib: plotting library for visualization
- pandas: data analysis library
- scikit-learn: machine learning library

Python-based Ecosystem for Machine Learning

Installing Python and ML packages

Anaconda Python distribution

is a free –including for commercial use– enterprise-ready Python distribution that bundles all the essential Python packages for machine learning, data science, math, and engineering in one user-friendly cross-platform distribution.

Anaconda installer can be downloaded at

https://www.anaconda.com/download/

Anaconda quick-start guide available at

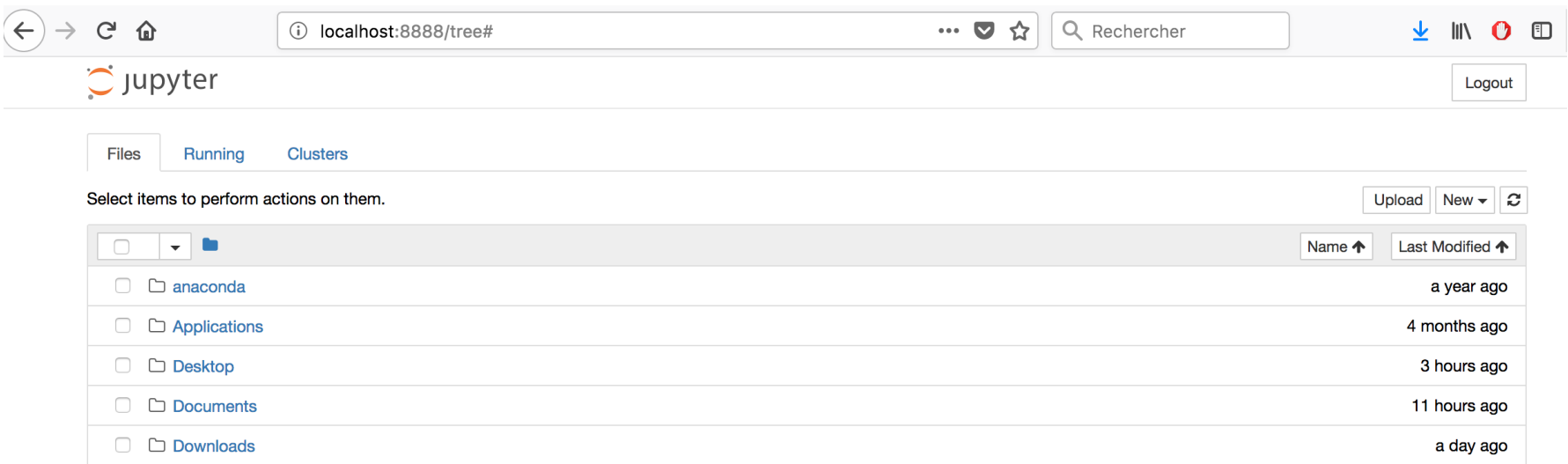
https://conda.io/docs/user-guide/getting-started.html

Jupyter Notebooks

- formerly known as IPython notebooks
- an interactive computational environment
- can contain code, text, images, output, ...
- installed by default with the Anaconda distribution

We can invoke the Jupyter notebook by executing the following command at the command prompt or terminal:

```
$ jupyter notebook
```



Python and scikit-learn: checking your installation

Open a Python console or a Jupyter notebook and execute the following:

```
>>> import sys
>>> print (sys.version)
3.6.1 |Anaconda 4.4.0 (x86_64)| (default, May 11 2017, 13:04:09)
[GCC 4.2.1 Compatible Apple LLVM 6.0 (clang-600.0.57)]
```

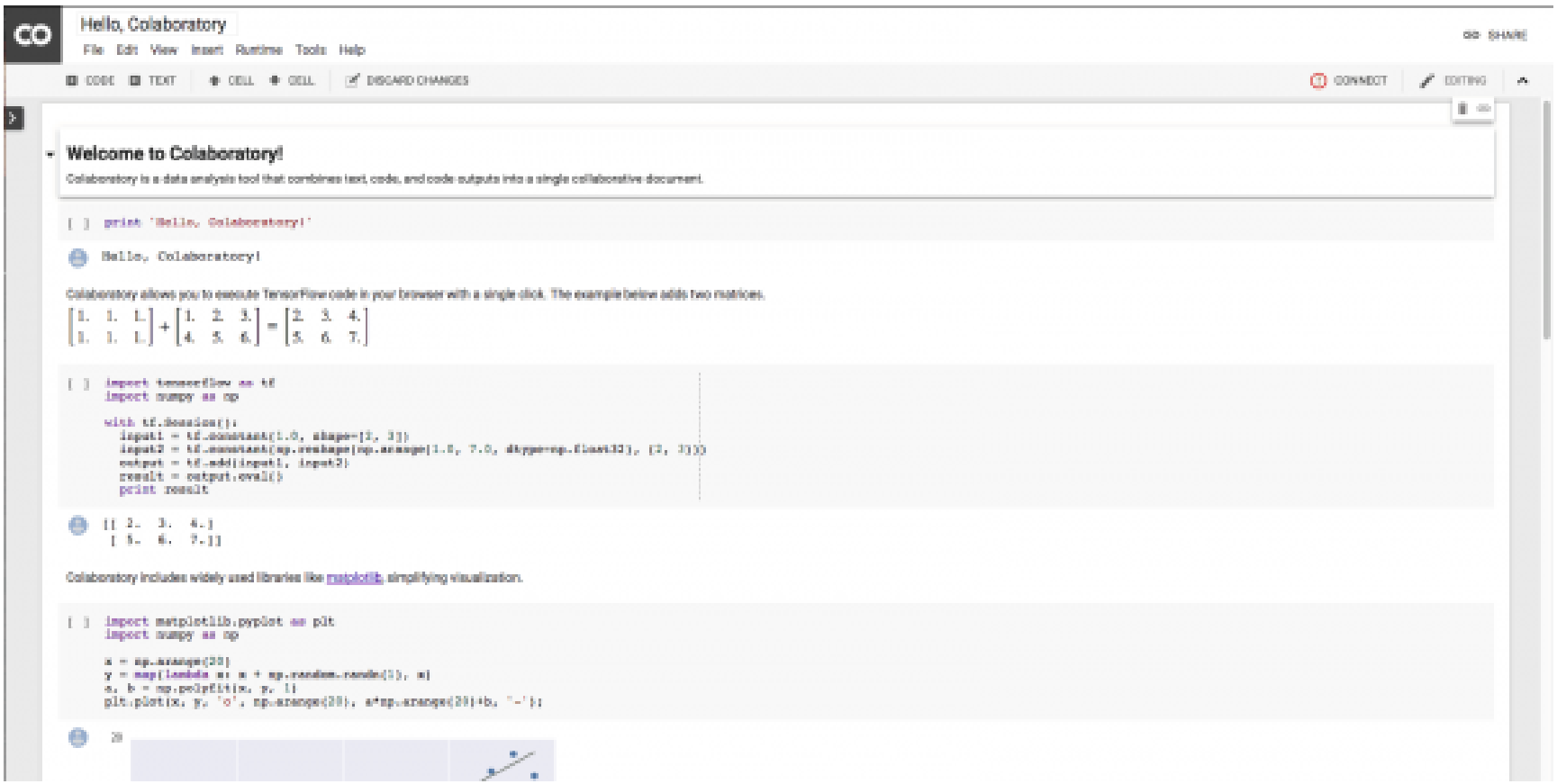
- to verify the Python version

```
>>> import sklearn
>>> sklearn.__version__
'0.18.1'
```

- to check that scikit-learn has been installed correctly

Colaboratory

Google internal tool for data science and machine learning workflow available at https://colab.research.google.com/



Python: Key Concepts

Identifiers Python entities such as class, functions, and variables

- can be a combination of upper-or lower-case letters (a to z or A to Z)
- can be any digits (0 to 9) or an underscore (_)
- cannot start with a digit. For example, 1variable is not valid, whereas variable1 is valid
- special symbols like , , #, $, % etc cannot be part of the identifiers

Keywords set of reserved words used in Python to define the syntax and structure of the language

FALSE	Class	Finally	Is	return
None	Continue	For	Lambda	try
TRUE	Def	From	nonlocal	while
And	Del	Global	Not	with
As	Elif	If	Or	yield
Assert	Else	Import	Pass	
Break	Except	In	Raise	

Python: Key Concepts

A First Python Program

```
>>> print("Welcome to the 5G&SC 2019 days!")
Welcome to the 5G&SC 2019 days!
```

Indentation

Each line of code must be indented by the same amount to denote a block of code in Python

```
# Correct indentation
x = 1
if x == 1:
print ('x has a value of 1')
else:
print ('x does NOT have a value of 1')
```

```
# incorrect indentation, program will generate a syntax error
# due to the wrong indentation in the else statement
x = 1
if x == 1:
print ('x has a value of 1')
else:
print ('x does NOT have a value of 1')
```

Python: Key Concepts

Basic Object Types

Types	Examples	Types	Examples
None	None	String	'this is a string', "also me"
Boolean	True, False	Tuple	(1, True, 'ML')
Integer	-1, 0, sys.maxint	List	[1, True, 'ML']
Long	1L, 9787L	Set	set(1, True, 'ML')
Float	3.141592654	Dictionary	{'1':'A', '2':'AA', True:1}
Complex	2+8j	File	f = open('filename', 'rb')

```
Float = 3.14
List = [1, True, 'ML'] # Values can be changed

# lets print the object type and the value
print(type(Float), Float)
print(type(List), List)

----- output ------
<type 'float'> 3.14
<type 'list'> [1, True, 'ML']
```

Python: Key Concepts

When to Use List vs. Tuples vs. Set vs. Dictionary

- List: Use when you need an ordered sequence of homogeneous collections, whose values can be changed later in the program
- Tuple: Use when you need an ordered sequence of heterogeneous collections whose values need not be changed later in the program
- Set: It is ideal for use when you don't have to store duplicates and you are not concerned about the order or the items. You just want to know whether a particular value already exists or not
- Dictionary: It is ideal for use when you need to relate values with keys, in order to look them up efficiently using a key

Python: Key Concepts

Basic Operators

- Arithmetic Operators
- Comparison or Relational Operators
- Assignment Operators
- Bitwise Operators
- Logical Operators
- Membership Operators
- Identity Operators

See the associated notebook for a collection of examples illustrating these operators

Python: Key Concepts

Iteration

A loop control statement enables us to execute a single or a set of programming statements multiple times until a given condition is satisfied

```
# First Example
print "First Example"
for item in [1,2,3,4,5]:
   print('item :', item)

# Second Example
print "Second Example"
letters = ['A', 'B', 'C']
for letter in letters:
    print(' First loop letter :', letter)

# Third Example
print "Third Example"
for index in range(len(letters)):
    print('First loop letter :', letters[index])

# Example code for while loop statement
count = 0
while (count <3):
    print('The count is:', count)
```

```
----- output ------
First Example
item : 1
item : 2
item : 3
item : 4
item : 5
Second Example
First loop letter : A
First loop letter : B
First loop letter : C
Third Example
First loop letter : A
First loop letter : B
First loop letter : C
```

Python: Key Concepts

Lists

Python list is a collection of items which is ordered and changeable. Note that the items in the list need not be of the same data type

Description	Python Expression	Example	Results
Creating a list	[item1, item2,. . .]	list = ['a','b','c','d']	['a','b','c','d']
Accessing items	list[index]	list[2]	c
Length	len(list)	len([1, 2, 3])	3
Concatenation	list_1 + list_2	[1, 2, 3] + [4, 5]	[1, 2, 3, 4, 5]
Iteration	for x in list:print x	for x in [1, 2, 3]:print x	1 2 3
Slicing	list[index:]	list = [1,2,3]; list[1:]	[2,3]
Append object	list.append(obj)	[1,2,3].append(5)	[1,2,3,5]
Remove object	list.remove(obj)	['a','c',1].remove('c')	['a', 1]
Count occurrence	list.count(obj)	[1,1,2,3].count(1)	2

Python: Key Concepts

Defining a Function

- The keyword **def** denotes the beginning of a function block, which will be followed by the name of the function and open, close parentheses. After this a colon **(:)** to be put to indicate the end of the function header
- Functions can accept arguments or parameters. Any such input arguments or parameters should be placed within the parentheses in the header of the parameter
- The main code statements are to be put below the function header and should be indented, which indicates that the code is part of the same function
- Functions can return an expression to the caller using **return**

Python: Key Concepts

Syntax for Creating Functions

without argument

```
# Simple function
def someFunction():
    print("IA&ML @ 5GSC 2019")

# Call the function
someFunction()

----- output -----
IA&ML @ 5GSC 2019
```

with argument

```
# Simple function to add two numbers
def sum_two_numbers(x, y):
    return x + y

# after this line x will hold the value 3
print(sum_two_numbers(1,2))

----- output -----
3
```

- Numpy basics

  (NumPy documentation: `http://docs.scipy.org/doc/`)

- Pandas basics

  (pandas documentation: `http://pandas.pydata.org/`

- Machine learning method using scikit-learn

  (scikit-learn documentation:
  `http://scikit-learn.org/stable/documentation.html`)

to be continued...

Go to Google Colaboratory or Jupyter notebook and run the notebook

`ML_5GSC2019.ipynb`