# Tutorial Introduction

## *Machine Learning for Materials Science*

Adalberto Fazzio, Marcio Costa, Antonio C M Padilha, Gabriel R Schleder

SBF Autumn Meeting - 26 may 2019

*Tutorial Website*

*https://github.com/gschleder/MLtutorial*

Universidade Federal do ABC

CNPEM
Brazilian Center for Research in Energy and Materials

LNNano
Brazilian Nanotechnology National Laboratory

MINISTRY OF SCIENCE, TECHNOLOGY, INNOVATION AND COMMUNICATION

MINISTRY OF SCIENCE, TECHNOLOGY, INNOVATION AND COMMUNICATION

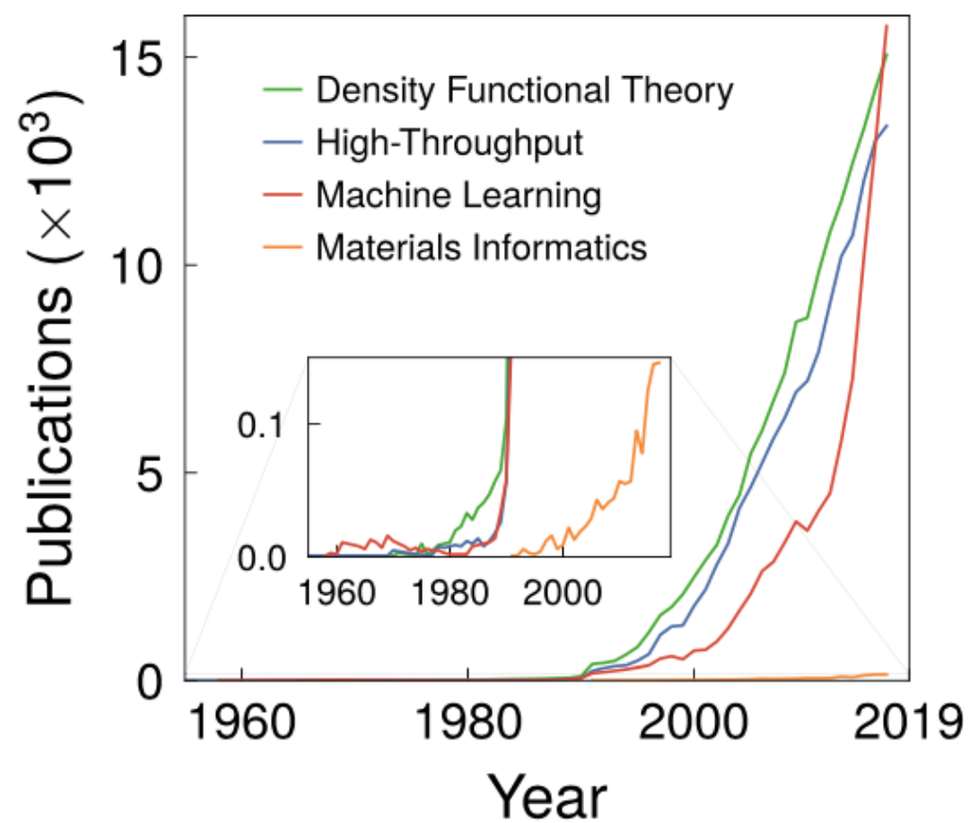
JPhys Materials

**TOPICAL REVIEW**

## From DFT to machine learning: recent approaches to materials science–a review

**Gabriel R Schleder[1,2,3], Antonio C M Padilha[2], Carlos Mera Acosta[1,2], Marcio Costa[2] and Adalberto Fazzio[1,2,3]**

[1] Center for Natural and Human Sciences, Federal University of ABC, 09210-580, Santo André, São Paulo, Brazil
[2] Brazilian Nanotechnology National Laboratory/CNPEM, 13083-970, Campinas, São Paulo, Brazil
[3] Authors to whom any correspondence should be addressed.



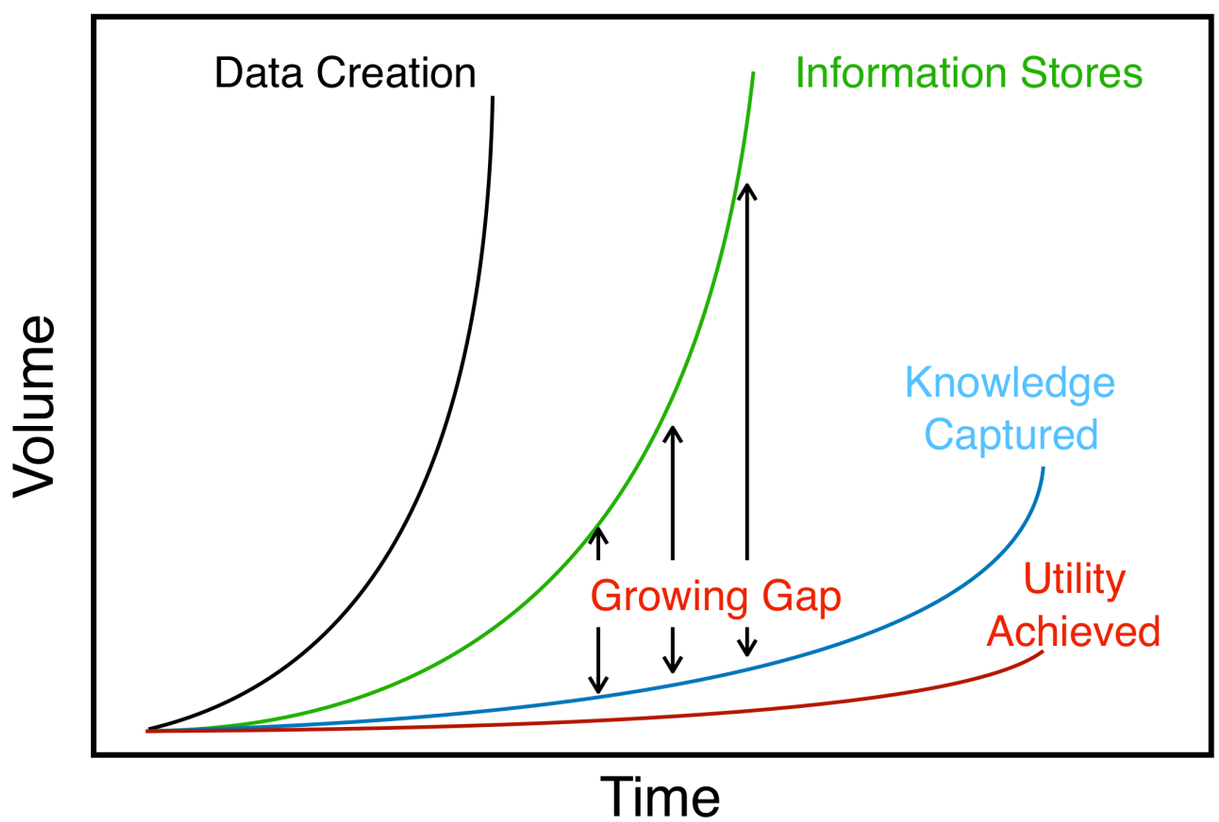
JPhys Materials

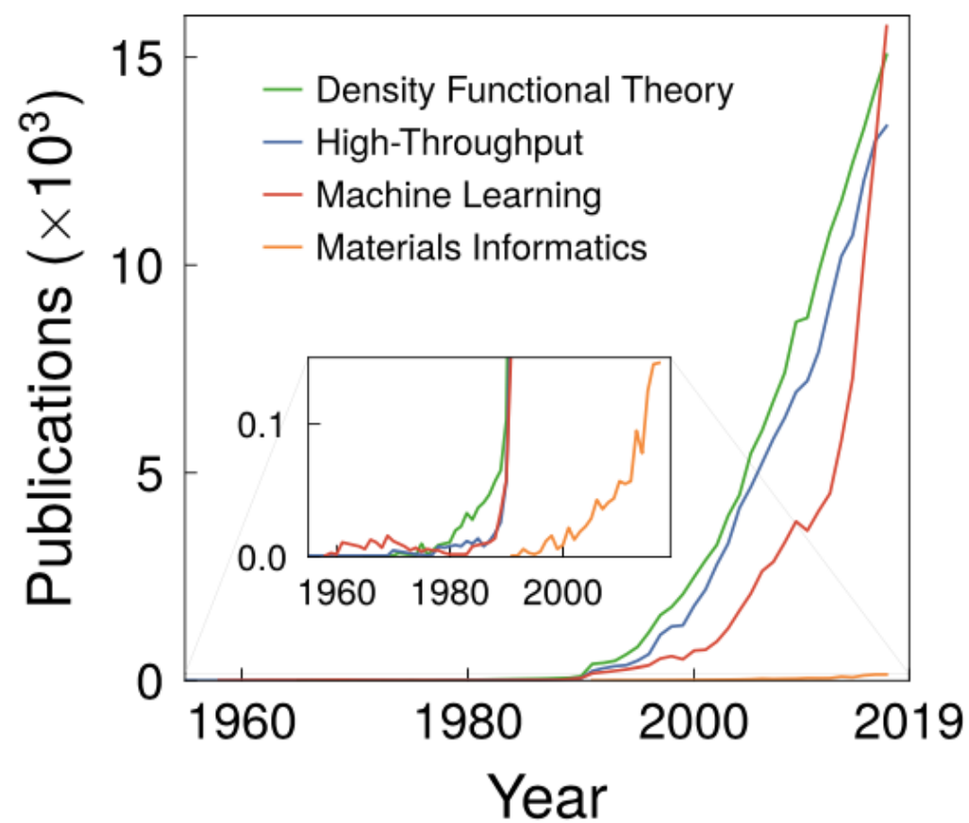
**TOPICAL REVIEW**

## From DFT to machine learning: recent approaches to materials science–a review

**Gabriel R Schleder[1,2,3], Antonio C M Padilha[2], Carlos Mera Acosta[1,2], Marcio Costa[2] and Adalberto Fazzio[1,2,3]**

[1] Center for Natural and Human Sciences, Federal University of ABC, 09210-580, Santo André, São Paulo, Brazil
[2] Brazilian Nanotechnology National Laboratory/CNPEM, 13083-970, Campinas, São Paulo, Brazil
[3] Authors to whom any correspondence should be addressed.



# Banking System



Question : How to decide whether to lend money?

# Banking System



## Question : How to decide whether to lend money?

## Banking

| Label | X1 | X2 | X3 | X4 | Xn |
|---|---|---|---|---|---|
| John | | | | | |

# Banking System



## Question : How to decide whether to lend money?

## Banking

| Label | X1 | X2 | X3 | X4 | Xn |
|---|---|---|---|---|---|
| John | Income | | | | |

# Banking System



## Question : How to decide whether to lend money?

## Banking

| Label | X1 | X2 | X3 | X4 | Xn |
|---|---|---|---|---|---|
| John | Income | Age | | | |

# Banking System



## Question : How to decide whether to lend money?

## Banking

| Label | X1 | X2 | X3 | X4 | Xn |
|---|---|---|---|---|---|
| John | Income | Age | Morgatge | | |

# Banking System



## Question : How to decide whether to lend money?

## Banking

| Label | X1 | X2 | X3 | X4 | Xn |
|---|---|---|---|---|---|
| John | Income | Age | Morgatge | Education | |

# Banking System



## Question : How to decide whether to lend money?

## Banking

| Label | X1 | X2 | X3 | X4 | Xn |
|---|---|---|---|---|---|
| John | Income | Age | Morgatge | Education | ? |

# Banking System



## Question : How to decide whether to lend money?

## Banking

| Label | X1 | X2 | X3 | X4 | Xn |
|---|---|---|---|---|---|
| John | Income | Age | Morgatge | Education | ? |

John's fingerprint!!

# Banking System



## Question : How to decide whether to lend money?

### Banking

| Label | X1 | X2 | X3 | X4 | Xn |
|---|---|---|---|---|---|
| John | Income | Age | Morgatge | Education | ? |

John's fingerprint!!

### Material Science

| Label | X1 | X2 | X3 | X4 | Xn |
|---|---|---|---|---|---|
| Bi-BCC | IE | EA | HOMO | LUMO | ? |

# Banking System



Question : How to decide whether to lend money?

Banking

| Label | X1 | X2 | X3 | X4 | Xn |
|---|---|---|---|---|---|
| John | Income | Age | Morgatge | Education | ? |

John's fingerprint!!

The bank has a large database of past loans

# Banking System



Question : How to decide whether to lend money?

Banking

| Label | X1 | X2 | X3 | X4 | Xn |
|---|---|---|---|---|---|
| John | Income | Age | Morgatge | Education | ? |

John's fingerprint!!

The bank has a large database of past loans

Can we use the historical data to predict the new loan?

# What is Machine Learning?

We would like to analyze and understand the output "P"( Property) on the input "X"(fingerprint).

$$\mathbf{P} = \mathbf{f}(\mathbf{X})$$

- How to predict property P?
- Discover the function $f(\mathbf{X})=P$
  - **X**: known (input) variables, easily obtained

# What is Machine Learning?



We would like to analyze and understand the output "P"( Property) on the input "X"(fingerprint).

$$\mathbf{P} = \mathbf{f}(\mathbf{X})$$

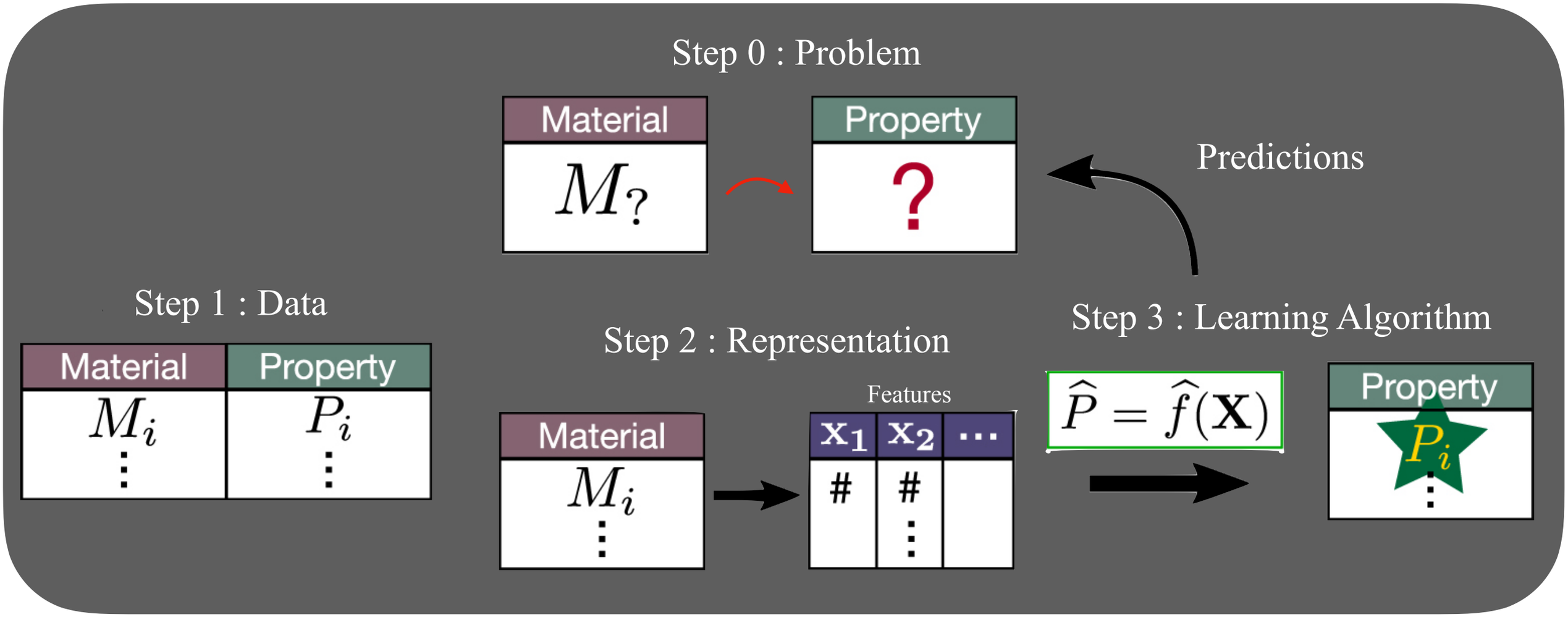
- How to predict property P?
- Discover the function $f(\mathbf{X})=P$
  - **X**: known (input) variables, easily obtained
  - Ideally, $f(\mathbf{X})$ has physical meaning

# What is Machine Learning?

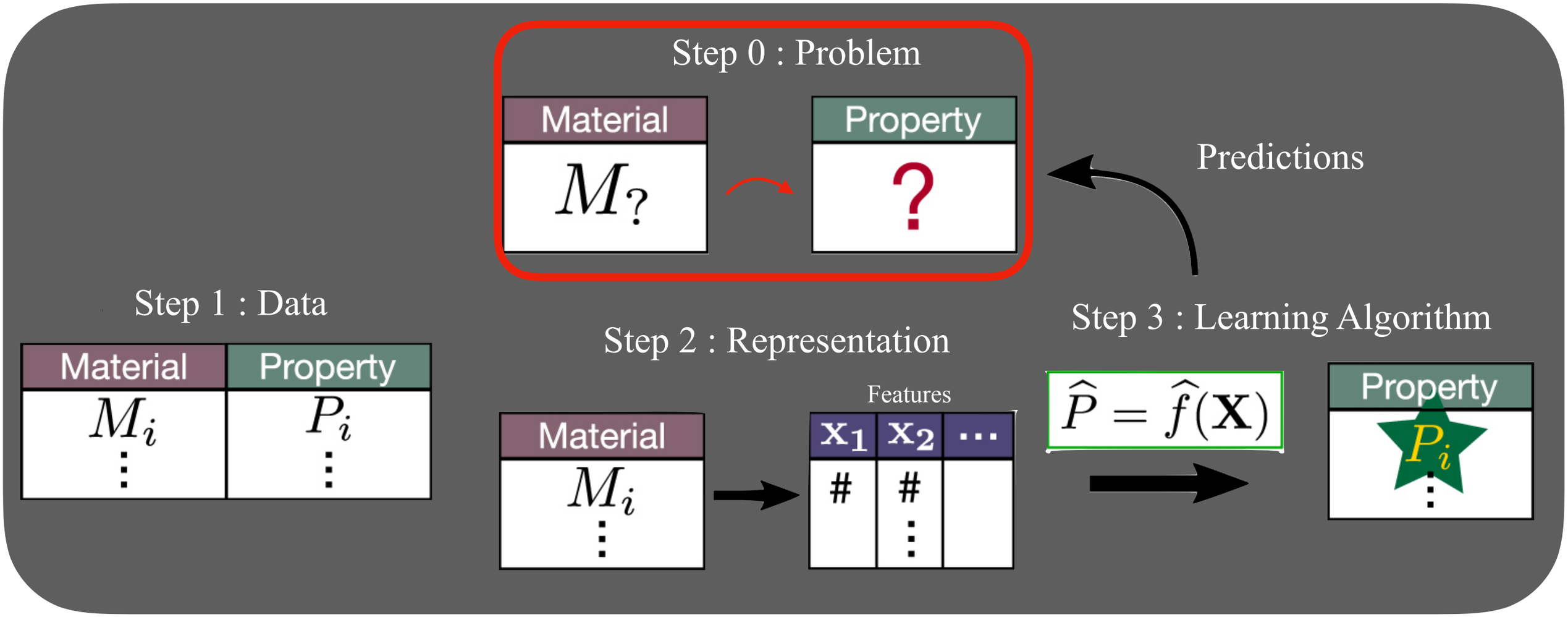


Machine learning can be defined as a class of methods for automated data analysis, which are capable of detecting patterns in data. These extracted patterns can be used to predict unknown data or to assist in decision-making processes under uncertainty.

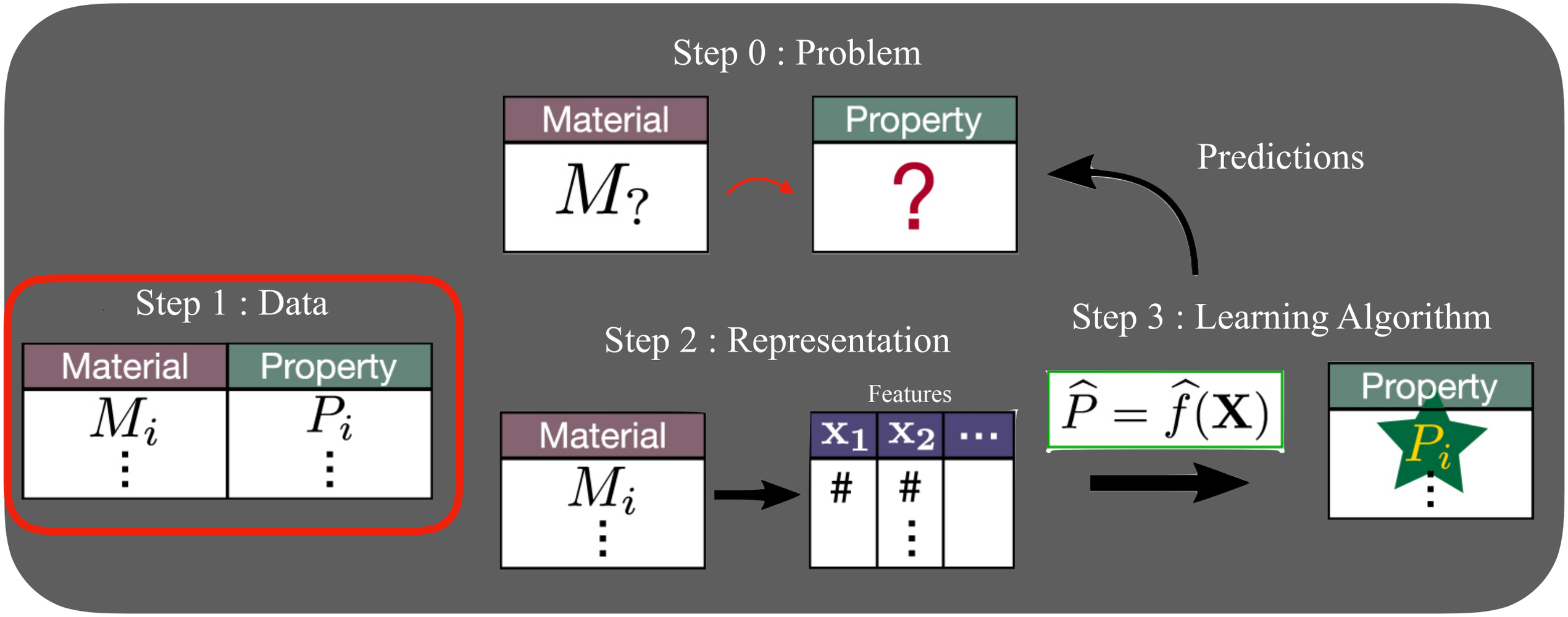
# Machine Learning Workflow



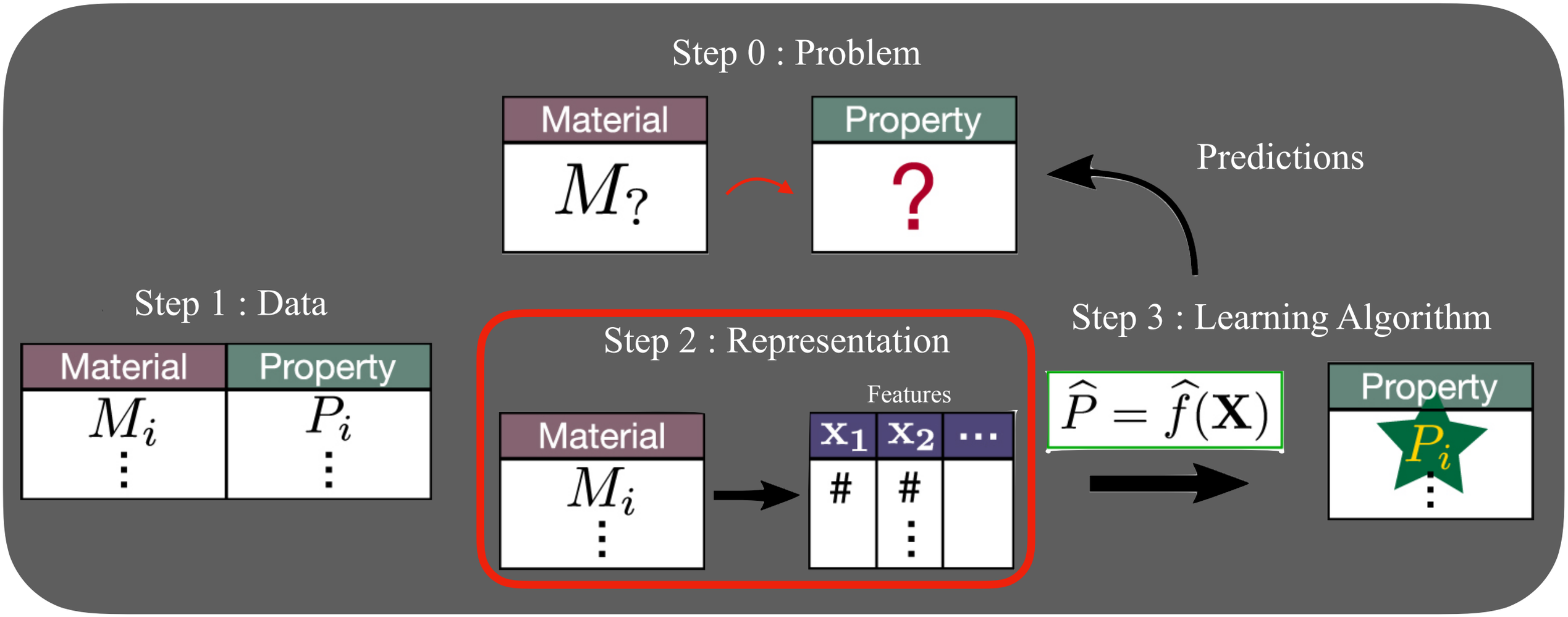
# Machine Learning Workflow



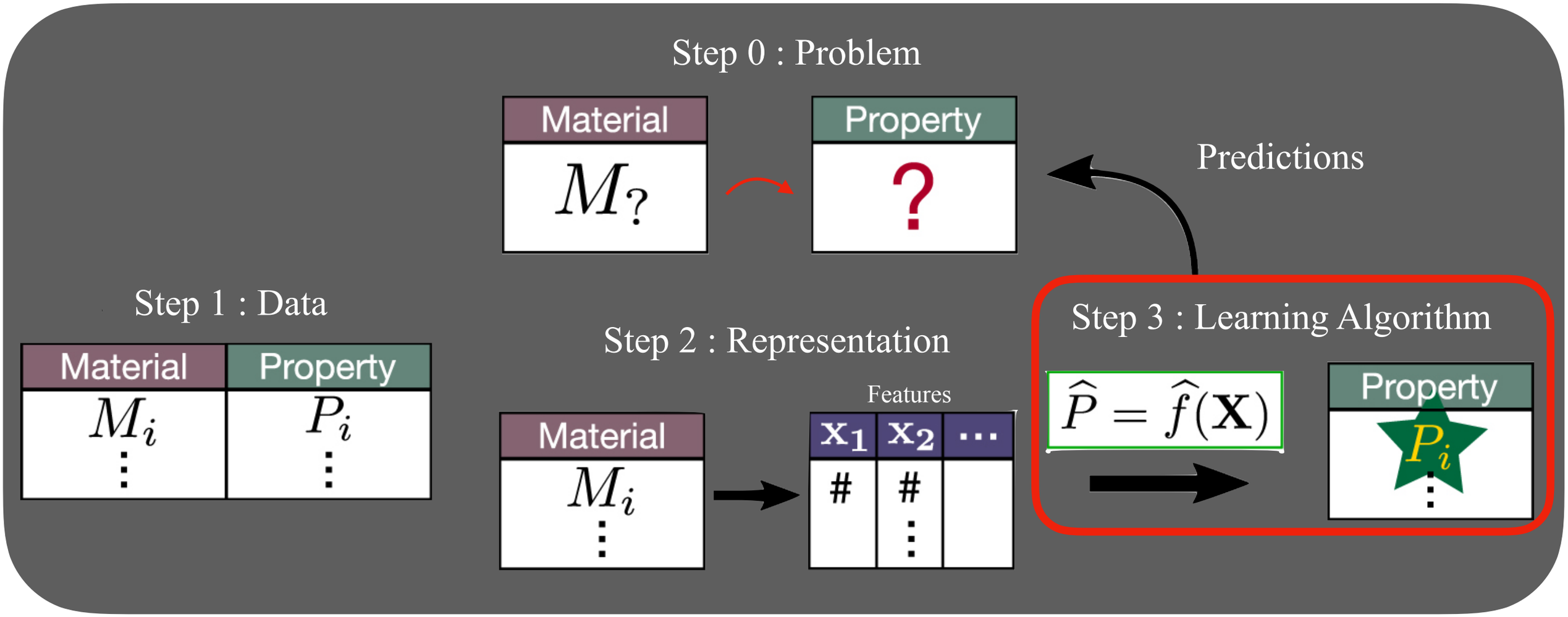
# Machine Learning Workflow



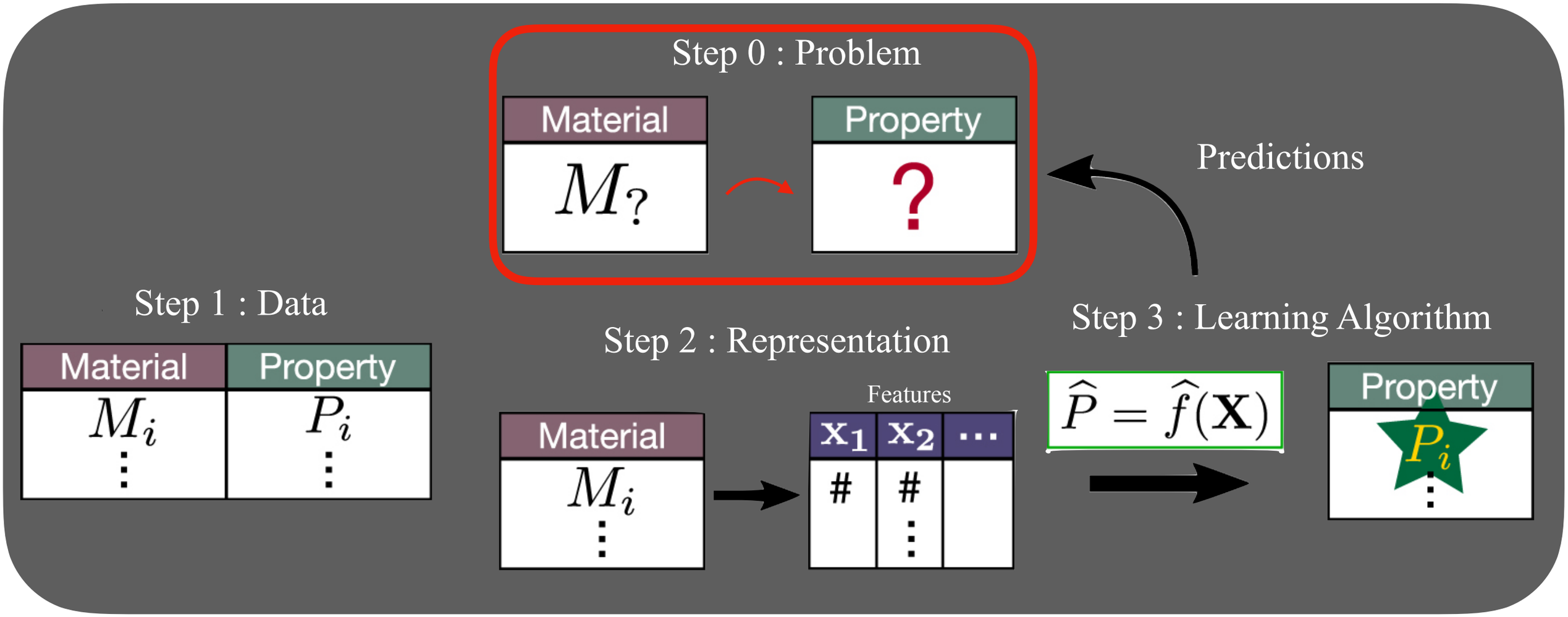
# Machine Learning Workflow



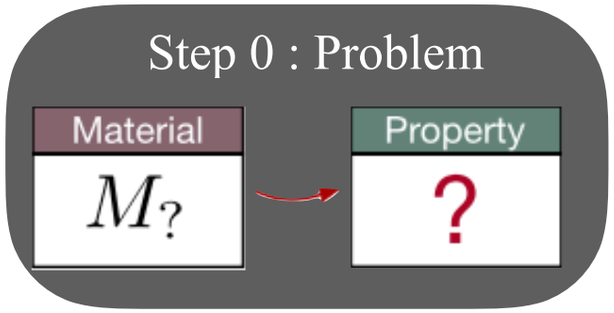
# Machine Learning Workflow



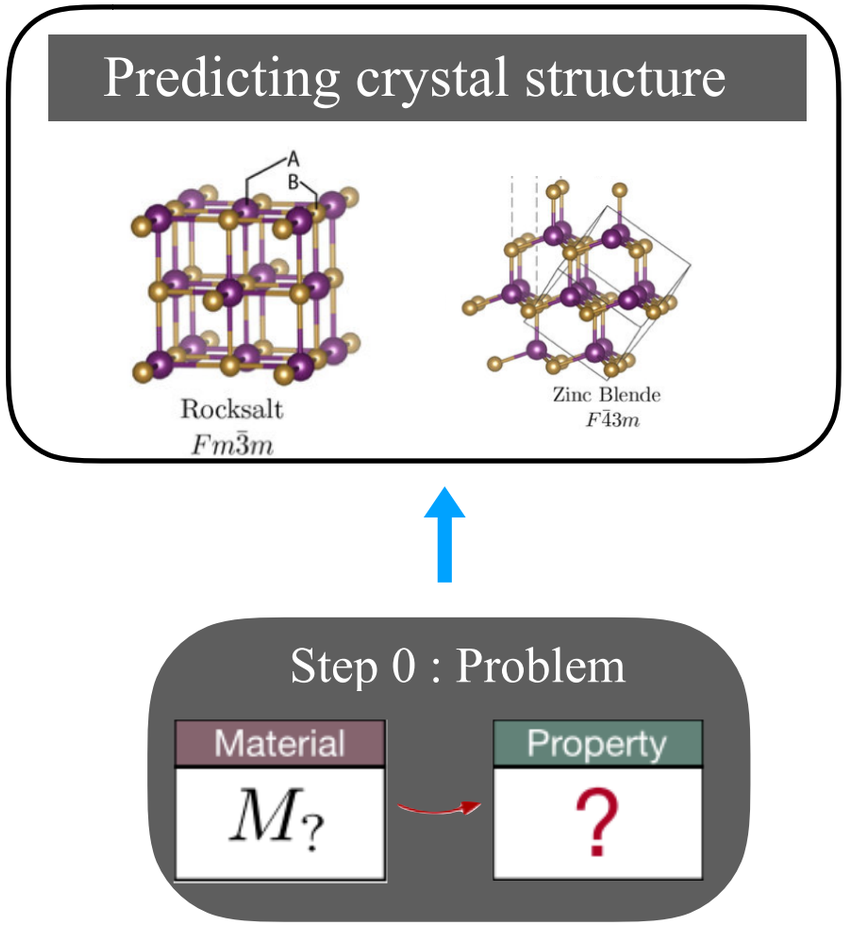
# Step 0 : Problem



# Step 0 : Problem

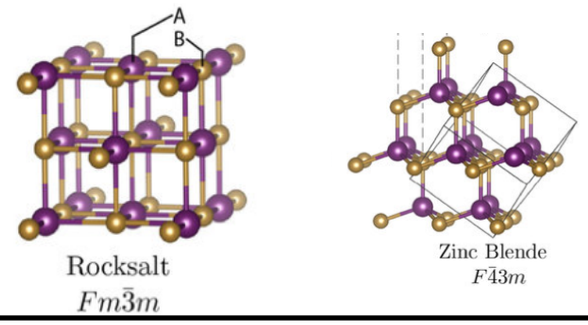


# Step 0 : Problem

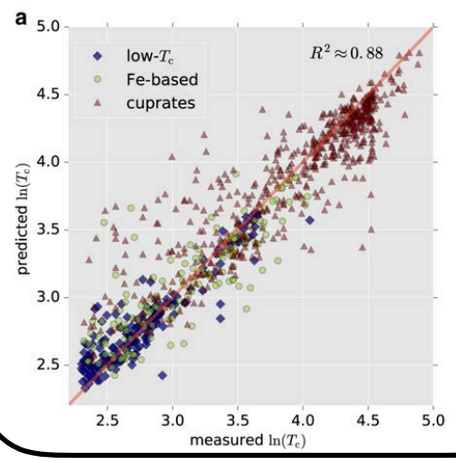


# Step 0 : Problem

## Predicting crystal structure




## Superconductivity

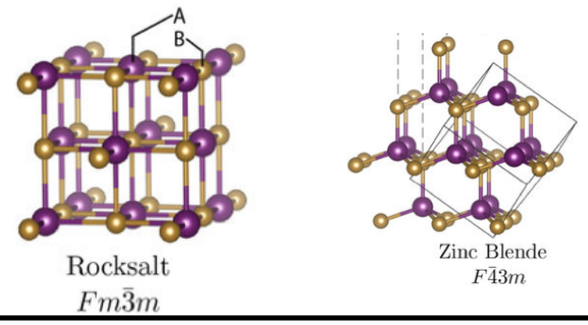



## Step 0 : Problem

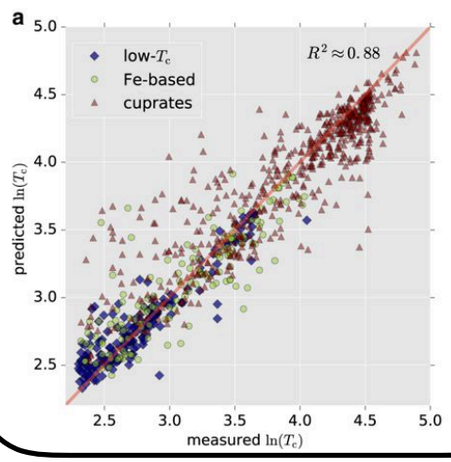
| Material | Property |
|---|---|
| $M_?$ | ? |

# Step 0 : Problem

## Predicting crystal structure




## Superconductivity

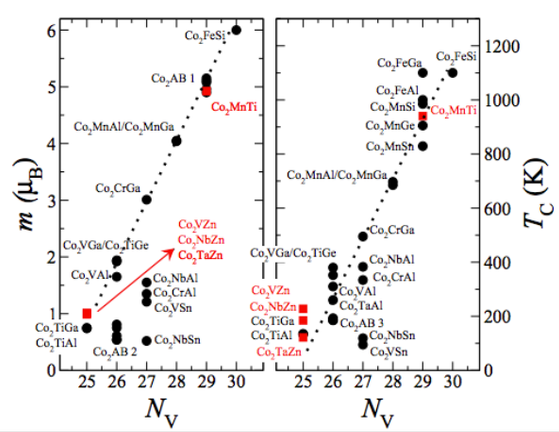



## Step 0 : Problem

Material: $M_?$

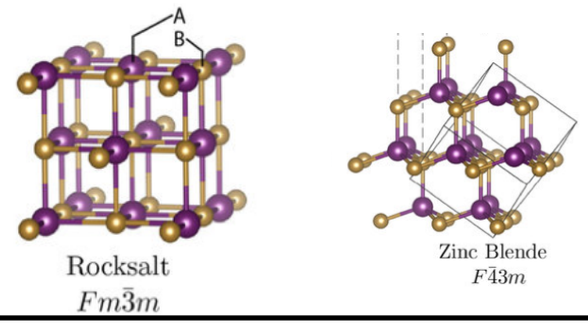
Property: ?

## Magnetic Materials



# Step 0 : Problem

## Predicting crystal structure

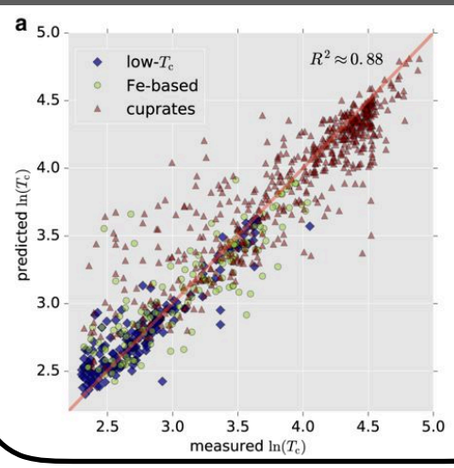



## Step 0 : Problem

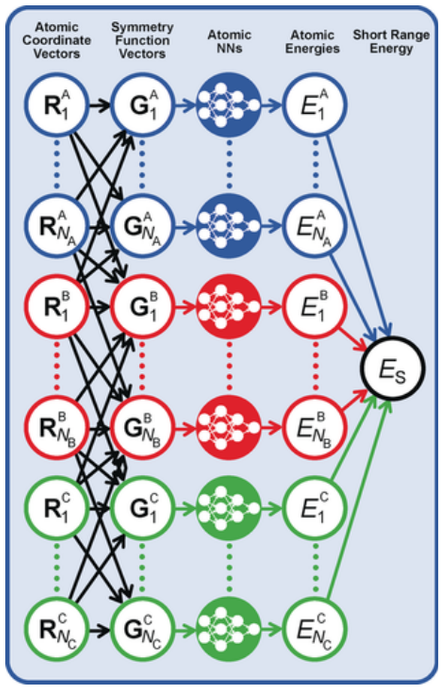
Material: $M_?$

Property: ?

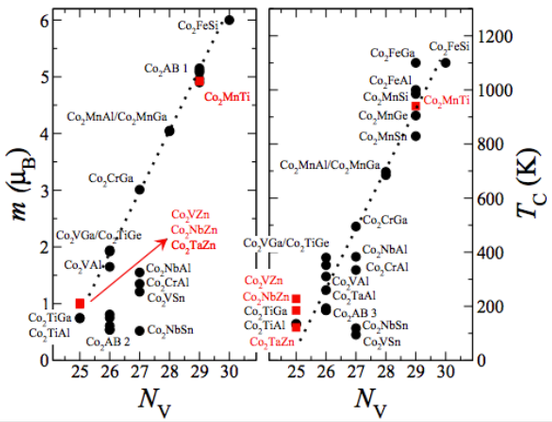
## Superconductivity




## Atomistic Potentials



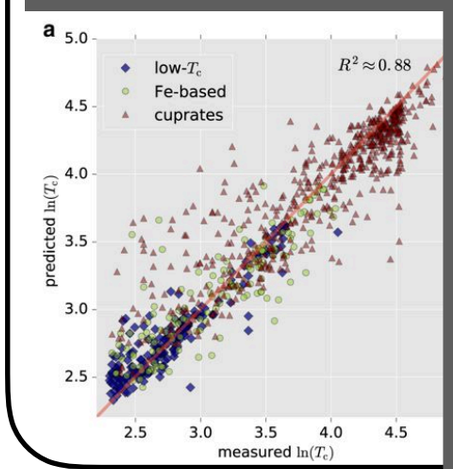

## Magnetic Materials



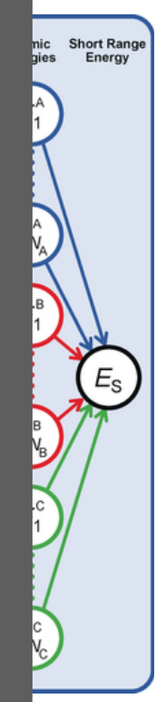
# Step 0 : Problem

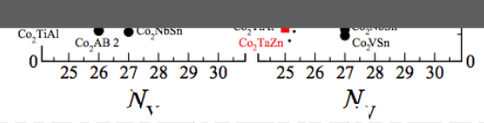
- Predicting crystal structure
- Formation energy
- Atomistic potentials
- Electronic properties : bandgap
- Electronic properties : conductivity
- Metal / Insulator?
- Permanent Magnets

Superconductiv

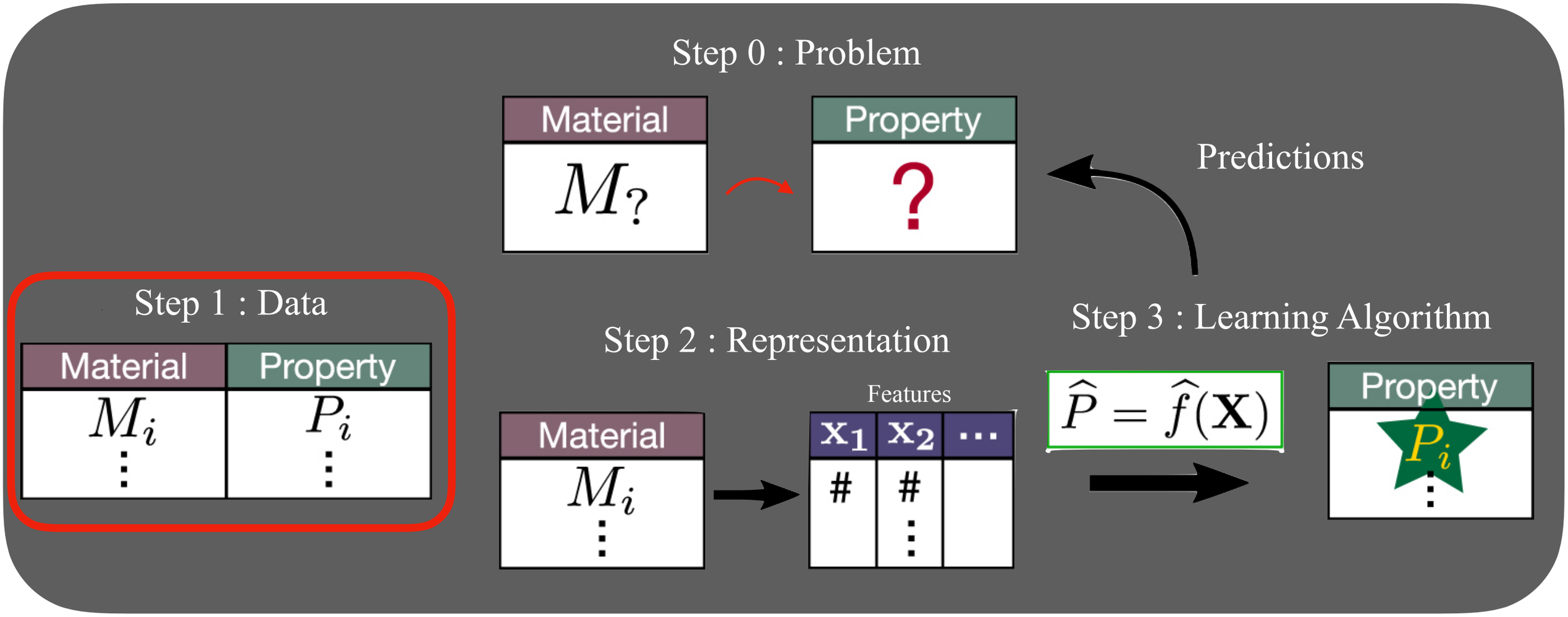



entials



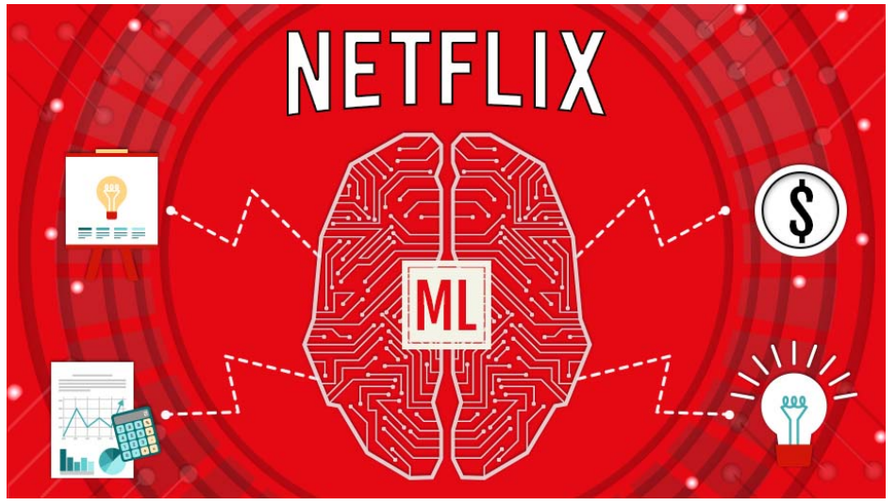
# Step 0 : Problem



# Step 0 : Problem

Pharmaceutical

Social




Step 1 : Data

| Material | Property |
|---|---|
| $M_i$ | $P_i$ |
| $\vdots$ | $\vdots$ |

Materials Science

MP

NOMAD

ICSD

AFLOW

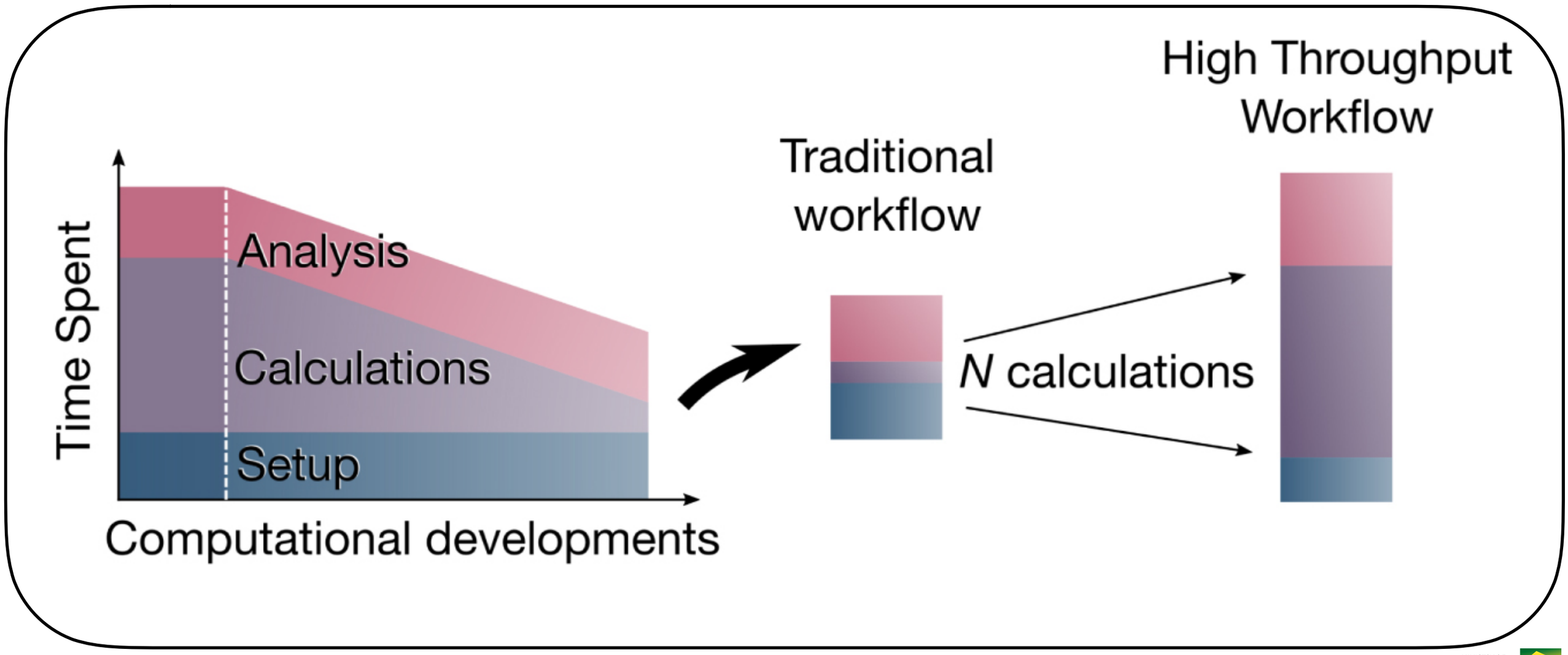
Automatic - FLOW for Materials Discovery

# Step 1 : Data

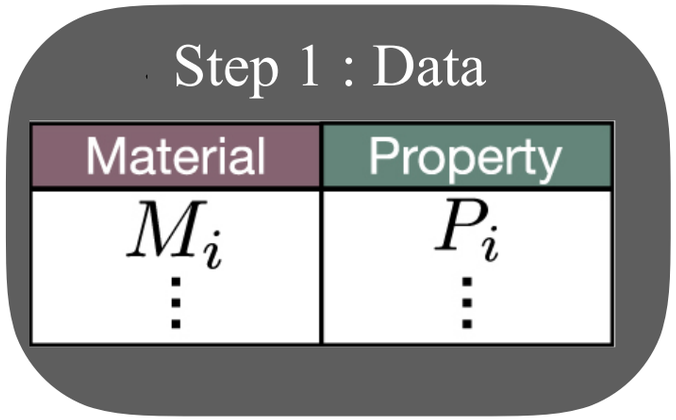
Step 1 : Data

| Material | Property |
| --- | --- |
| $M_i$ | $P_i$ |
| ⋮ | ⋮ |

## High Throughput (HT) calculations



# Step 1 : Data

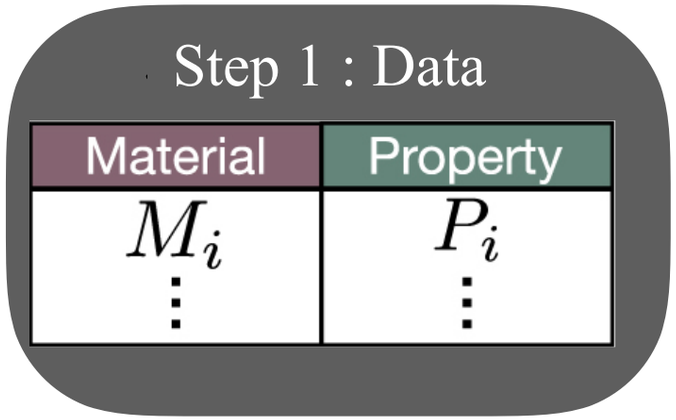



## Material Science Databases

**HT databases**

| | | | |
|---|---|---|---|
| ICSD | Inorganic experimental | http://www2.fiz-karlsruhe.de/icsd_home.html | [152] |
| COD | Organic and Inorganic experimental | http://crystallography.net | [153] |
| AFLOWlib | Multi-purpose repository | http://aflowlib.org/ | [148] |
| Materials Project | Multi-purpose repository | https://materialsproject.org/ | [149] |
| OQDM | Multi-purpose repository | http://oqmd.org/ | [150] |
| CMR | Multi-purpose (3D and 2D materials) repository | https://cmr.fysik.dtu.dk/ | [154] |
| OMDB | Organic materials database | https://omdb.mathub.io/ | [155] |
| MaterialsWeb | 2D materials (derived from Materials Project) | https://materialsweb.org/twodmaterials | [156] |
| JARVIS-DFT | 2D materials | https://ctcms.nist.gov/~knc6/JVASP.html | [157, 158] |
| NOMAD | Multiple source repository | https://repository.nomad-coe.eu/ | |
| Materials Cloud | Multiple source repository | https://materialscloud.org/discover | [3] |
| Citrination | Multiple source (experimental and theoretical) repository | https://citrination.com | [159] |
| Clean Energy Project | Multiple source repository for solar cells | | [160] |
| C2DB | 2D materials (derived from CMR) | https://cmr.fysik.dtu.dk/c2db/c2db.html | [161] |

# Step 1 : Data




## Material Science Databases

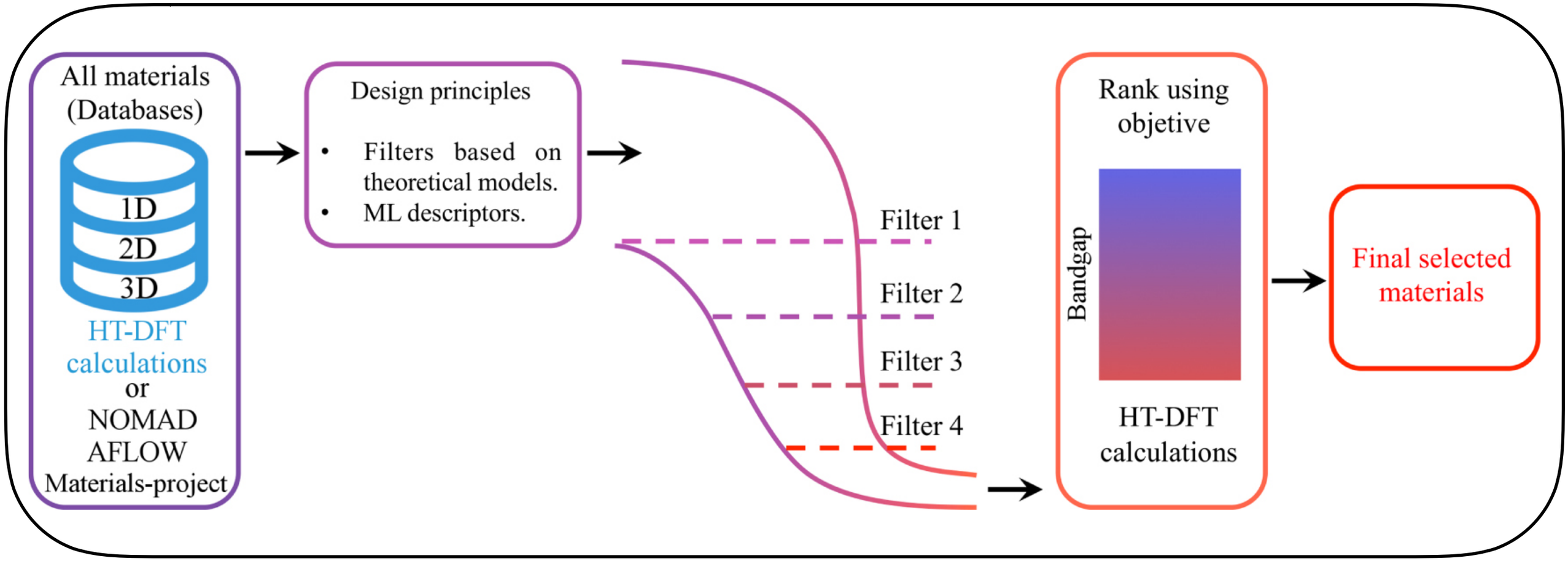
**HT databases**

| | | | |
|---|---|---|---|
| ICSD | Inorganic experimental | http://www2.fiz-karlsruhe.de/icsd_home.html | [152] |
| COD | Organic and Inorganic experimental | http://crystallography.net | [153] |
| AFLOWlib | Multi-purpose repository | http://aflowlib.org/ | [148] |
| Materials Project | Multi-purpose repository | https://materialsproject.org/ | [149] |
| OQDM | Multi-purpose repository | http://oqmd.org/ | [150] |
| CMR | Multi-purpose (3D and 2D materials) repository | https://cmr.fysik.dtu.dk/ | [154] |
| OMDB | Organic materials database | https://omdb.mathub.io/ | [155] |
| MaterialsWeb | 2D materials (derived from Materials Project) | https://materialsweb.org/twodmaterials | [156] |
| JARVIS-DFT | 2D materials | https://ctcms.nist.gov/~knc6/JVASP.html | [157, 158] |
| NOMAD | Multiple source repository | https://repository.nomad-coe.eu/ | |
| Materials Cloud | Multiple source repository | https://materialscloud.org/discover | [3] |
| Citrination | Multiple source (experimental and theoretical) repository | https://citrination.com | [159] |
| Clean Energy Project | Multiple source repository for solar cells | | [160] |
| C2DB | 2D materials (derived from CMR) | https://cmr.fysik.dtu.dk/c2db/c2db.html | [161] |

# Step 1 : Data

## Step 1 : Data

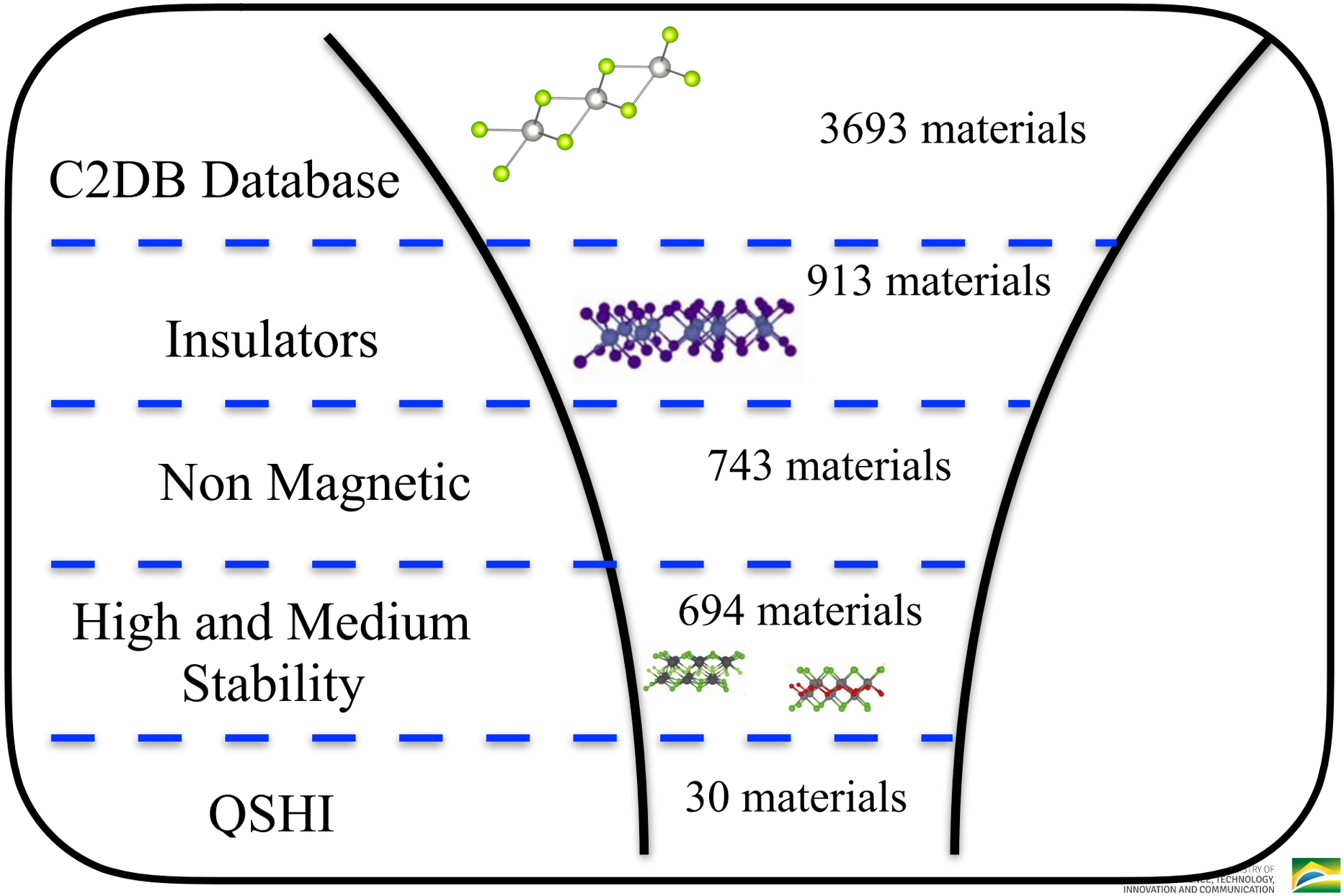
| Material | Property |
| --- | --- |
| $M_i$ | $P_i$ |
| ⋮ | ⋮ |



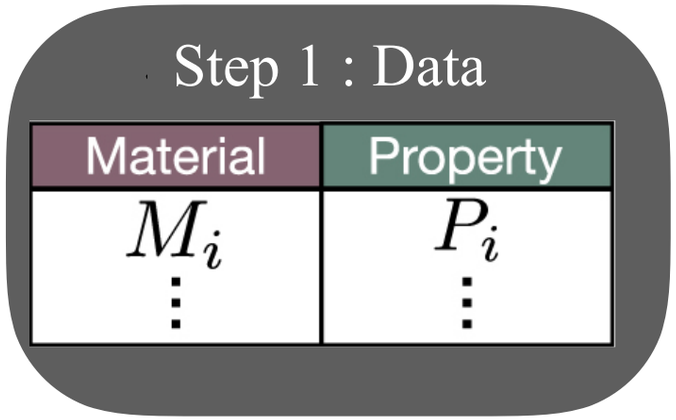
# Step 1 : Data

Step 1 : Data

| Material | Property |
| --- | --- |
| $M_i$ | $P_i$ |
| ⋮ | ⋮ |



# Step 1 : Data

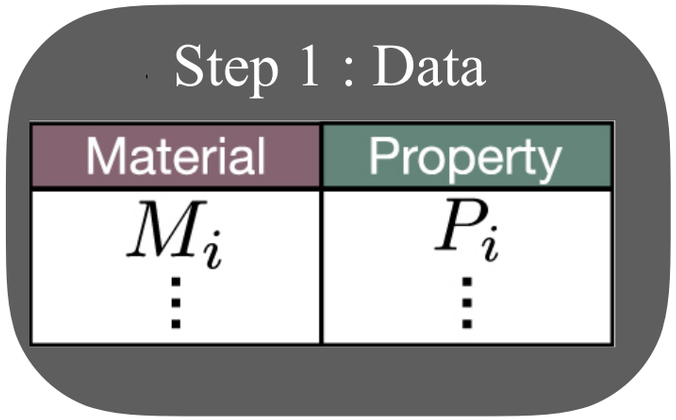



## Packages for HT and screening

**HT Codes and tools**

| | | | |
|---|---|---|---|
| ASE | Complete package for HT | https://wiki.fysik.dtu.dk/ase/ | [162] |
| Pymatgen | Complete package for HT | http://pymatgen.org/ | [163] |
| AiiDA | Framework for HT | http://aiida.net/ | [164] |
| AFLOW$\pi$ | Framework for HT | http://aflowlib.org/src/aflowpi/index.html | [165] |
| Atomate | Complete package for HT | https://atomate.org/ | [166] |
| Pylada | Framework for HT | http://pylada.github.io/pylada/ | |
| MPInterfaces | Framework for interfaces HT | http://henniggroup.github.io/MPInterfaces/ | [167] |
| Imeall | Atomistic properties of grain boundaries | https://github.com/Montmorency/imeall | [168] |
| FireWorks | Framework for HT | https://materialsproject.github.io/fireworks/ | [169] |

# Step 1 : Data

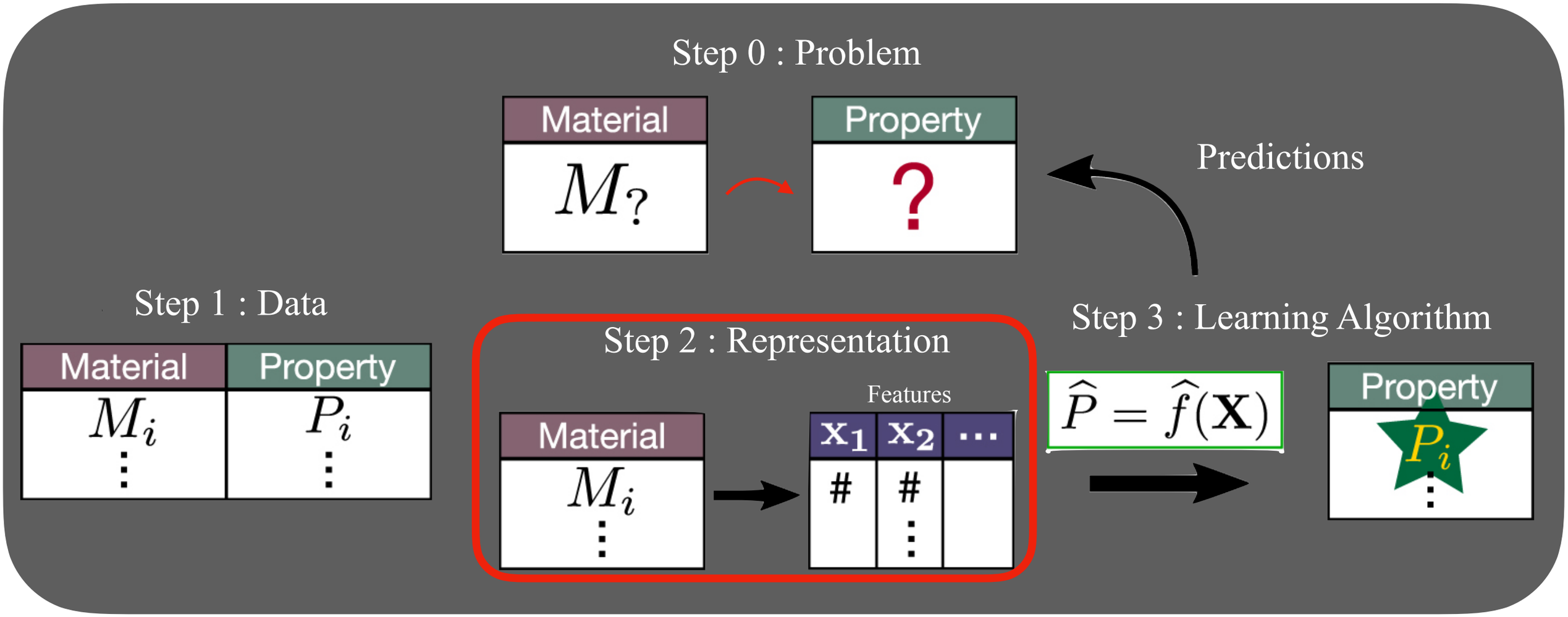



## Packages for HT and screening

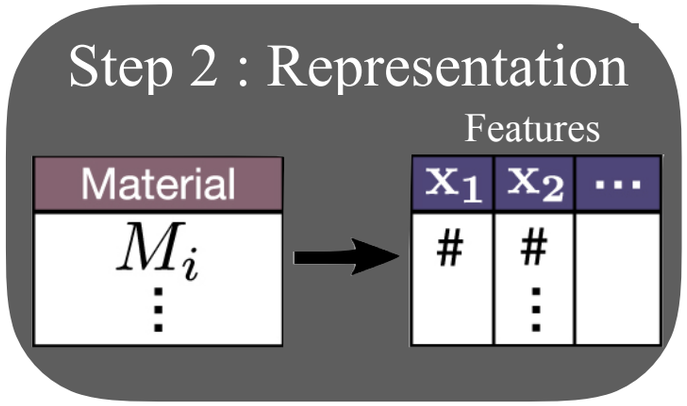
**HT Codes and tools**

| | | | |
|---|---|---|---|
| ASE | Complete package for HT | https://wiki.fysik.dtu.dk/ase/ | [162] |
| Pymatgen | Complete package for HT | http://pymatgen.org/ | [163] |
| AiiDA | Framework for HT | http://aiida.net/ | [164] |
| AFLOW$\pi$ | Framework for HT | http://aflowlib.org/src/aflowpi/index.html | [165] |
| Atomate | Complete package for HT | https://atomate.org/ | [166] |
| Pylada | Framework for HT | http://pylada.github.io/pylada/ | |
| MPInterfaces | Framework for interfaces HT | http://henniggroup.github.io/MPInterfaces/ | [167] |
| Imeall | Atomistic properties of grain boundaries | https://github.com/Montmorency/imeall | [168] |
| FireWorks | Framework for HT | https://materialsproject.github.io/fireworks/ | [169] |

# Step 2 : Representation

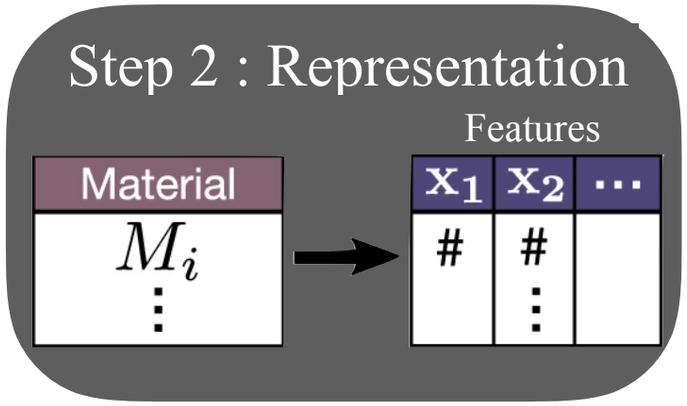


# Step 2 : Representation




i) Complete
ii) Unique
iii) Discriminative
iv) Efficient and Simple

# Step 2 : Representation



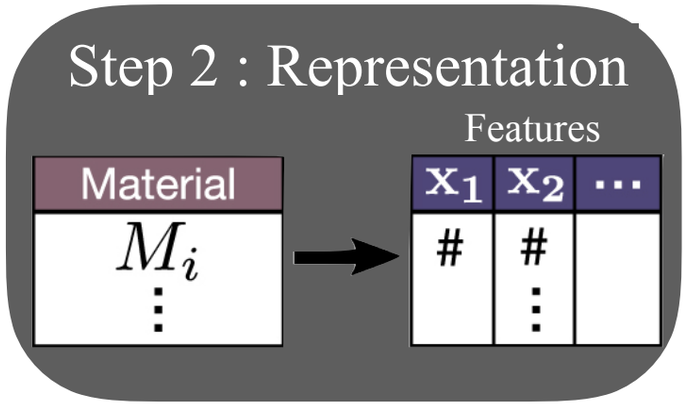

PRL **114,** 105503 (2015) PHYSICAL REVIEW LETTERS week ending 13 MARCH 2015

**Big Data of Materials Science: Critical Role of the Descriptor**

Luca M. Ghiringhelli,[1,*] Jan Vybiral,[2] Sergey V. Levchenko,[1] Claudia Draxl,[3] and Matthias Scheffler[1]
[1]*Fritz-Haber-Institut der Max-Planck-Gesellschaft, 14195 Berlin-Dahlem, Germany*
[2]*Department of Mathematical Analysis, Charles University, 18675 Prague, Czech Republic*
[3]*Humboldt-Universität zu Berlin, Institut für Physik and IRIS Adlershof, 12489 Berlin, Germany*
(Received 14 April 2014; revised manuscript received 20 October 2014; published 10 March 2015)

| $Z_A$ | A | IP [eV] | EA [eV] | EN = −1/2(EA+IP) [eV] | Highest occ. KS level [eV] | Lowest unocc. KS level [eV] | $r_s$ [Å] | $r_p$ [Å] | $r_d$ [Å] |
|---|---|---|---|---|---|---|---|---|---|
| 3 | Li | -5.329 | -0.698 | 3.014 | -2.874 | -0.978 | 1.652 | 1.995 | 6.930 |
| 4 | Be | -9.459 | 0.631 | 4.414 | -5.600 | -2.098 | 1.078 | 1.211 | 2.877 |
| 5 | B | -8.190 | -0.107 | 4.149 | -3.715 | 2.248 | 0.805 | 0.826 | 1.946 |
| 6 | C | -10.852 | -0.872 | 5.862 | -5.416 | 1.992 | 0.644 | 0.630 | 1.631 |
| 7 | N | -13.585 | -1.867 | 7.726 | -7.239 | 3.057 | 0.539 | 0.511 | 1.540 |
| ⋮ | | | | | | | | | |
| 50 | Sn | -7.043 | -1.039 | 4.041 | -3.866 | 0.008 | 1.057 | 1.344 | 2.030 |
| 51 | Sb | -8.468 | -1.847 | 5.158 | -4.991 | 0.105 | 1.001 | 1.232 | 2.065 |
| 52 | Te | -9.867 | -2.666 | 6.266 | -6.109 | 0.099 | 0.945 | 1.141 | 1.827 |
| 53 | I | -11.257 | -3.513 | 7.385 | -7.236 | 0.213 | 0.896 | 1.071 | 1.722 |
| 55 | Cs | -4.006 | -0.570 | 2.288 | -2.220 | -0.548 | 2.464 | 3.164 | 1.974 |
| 56 | Ba | -5.516 | 0.278 | 2.619 | -3.346 | -2.129 | 2.149 | 2.632 | 1.351 |

# Step 2 : Representation




PRL **114,** 105503 (2015) PHYSICAL REVIEW LETTERS week ending 13 MARCH 2015

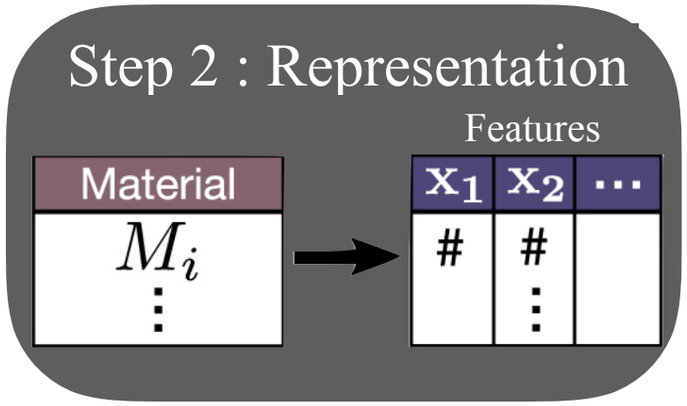
**Big Data of Materials Science: Critical Role of the Descriptor**

Luca M. Ghiringhelli,[1,*] Jan Vybiral,[2] Sergey V. Levchenko,[1] Claudia Draxl,[3] and Matthias Scheffler[1]
[1]*Fritz-Haber-Institut der Max-Planck-Gesellschaft, 14195 Berlin-Dahlem, Germany*
[2]*Department of Mathematical Analysis, Charles University, 18675 Prague, Czech Republic*
[3]*Humboldt-Universität zu Berlin, Institut für Physik and IRIS Adlershof, 12489 Berlin, Germany*
(Received 14 April 2014; revised manuscript received 20 October 2014; published 10 March 2015)

**X1**

| $Z_A$ | A | IP [eV] | EA [eV] | EN = −1/2(EA+IP) [eV] | Highest occ. KS level [eV] | Lowest unocc. KS level [eV] | $r_s$ [Å] | $r_p$ [Å] | $r_d$ [Å] |
|---|---|---|---|---|---|---|---|---|---|
| 3 | Li | -5.329 | -0.698 | 3.014 | -2.874 | -0.978 | 1.652 | 1.995 | 6.930 |
| 4 | Be | -9.459 | 0.631 | 4.414 | -5.600 | -2.098 | 1.078 | 1.211 | 2.877 |
| 5 | B | -8.190 | -0.107 | 4.149 | -3.715 | 2.248 | 0.805 | 0.826 | 1.946 |
| 6 | C | -10.852 | -0.872 | 5.862 | -5.416 | 1.992 | 0.644 | 0.630 | 1.631 |
| 7 | N | -13.585 | -1.867 | 7.726 | -7.239 | 3.057 | 0.539 | 0.511 | 1.540 |
| ⋮ | | | | | | | | | |
| 50 | Sn | -7.043 | -1.039 | 4.041 | -3.866 | 0.008 | 1.057 | 1.344 | 2.030 |
| 51 | Sb | -8.468 | -1.847 | 5.158 | -4.991 | 0.105 | 1.001 | 1.232 | 2.065 |
| 52 | Te | -9.867 | -2.666 | 6.266 | -6.109 | 0.099 | 0.945 | 1.141 | 1.827 |
| 53 | I | -11.257 | -3.513 | 7.385 | -7.236 | 0.213 | 0.896 | 1.071 | 1.722 |
| 55 | Cs | -4.006 | -0.570 | 2.288 | -2.220 | -0.548 | 2.464 | 3.164 | 1.974 |
| 56 | Ba | -5.516 | 0.278 | 2.619 | -3.346 | -2.129 | 2.149 | 2.632 | 1.351 |

# Step 2 : Representation




PRL **114,** 105503 (2015) PHYSICAL REVIEW LETTERS week ending 13 MARCH 2015

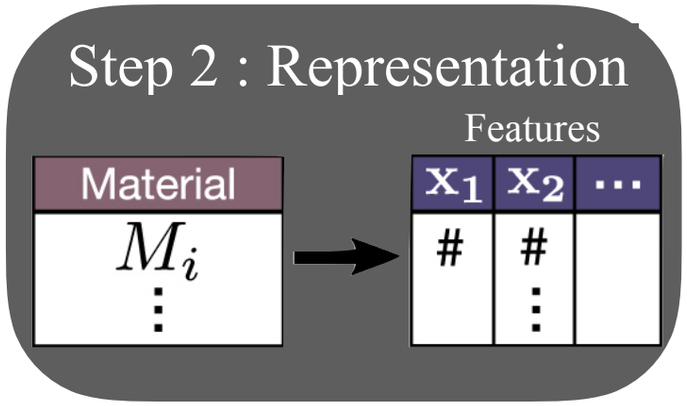
**Big Data of Materials Science: Critical Role of the Descriptor**

Luca M. Ghiringhelli,[1,*] Jan Vybiral,[2] Sergey V. Levchenko,[1] Claudia Draxl,[3] and Matthias Scheffler[1]
[1]*Fritz-Haber-Institut der Max-Planck-Gesellschaft, 14195 Berlin-Dahlem, Germany*
[2]*Department of Mathematical Analysis, Charles University, 18675 Prague, Czech Republic*
[3]*Humboldt-Universität zu Berlin, Institut für Physik and IRIS Adlershof, 12489 Berlin, Germany*
(Received 14 April 2014; revised manuscript received 20 October 2014; published 10 March 2015)

**X1 X2**

| $Z_A$ | A | IP [eV] | EA [eV] | EN = −1/2(EA+IP) [eV] | Highest occ. KS level [eV] | Lowest unocc. KS level [eV] | $r_s$ [Å] | $r_p$ [Å] | $r_d$ [Å] |
|---|---|---|---|---|---|---|---|---|---|
| 3 | Li | -5.329 | -0.698 | 3.014 | -2.874 | -0.978 | 1.652 | 1.995 | 6.930 |
| 4 | Be | -9.459 | 0.631 | 4.414 | -5.600 | -2.098 | 1.078 | 1.211 | 2.877 |
| 5 | B | -8.190 | -0.107 | 4.149 | -3.715 | 2.248 | 0.805 | 0.826 | 1.946 |
| 6 | C | -10.852 | -0.872 | 5.862 | -5.416 | 1.992 | 0.644 | 0.630 | 1.631 |
| 7 | N | -13.585 | -1.867 | 7.726 | -7.239 | 3.057 | 0.539 | 0.511 | 1.540 |
| ⋮ | | | | | | | | | |
| 50 | Sn | -7.043 | -1.039 | 4.041 | -3.866 | 0.008 | 1.057 | 1.344 | 2.030 |
| 51 | Sb | -8.468 | -1.847 | 5.158 | -4.991 | 0.105 | 1.001 | 1.232 | 2.065 |
| 52 | Te | -9.867 | -2.666 | 6.266 | -6.109 | 0.099 | 0.945 | 1.141 | 1.827 |
| 53 | I | -11.257 | -3.513 | 7.385 | -7.236 | 0.213 | 0.896 | 1.071 | 1.722 |
| 55 | Cs | -4.006 | -0.570 | 2.288 | -2.220 | -0.548 | 2.464 | 3.164 | 1.974 |
| 56 | Ba | -5.516 | 0.278 | 2.619 | -3.346 | -2.129 | 2.149 | 2.632 | 1.351 |

# Step 2 : Representation




PRL **114,** 105503 (2015) PHYSICAL REVIEW LETTERS week ending 13 MARCH 2015

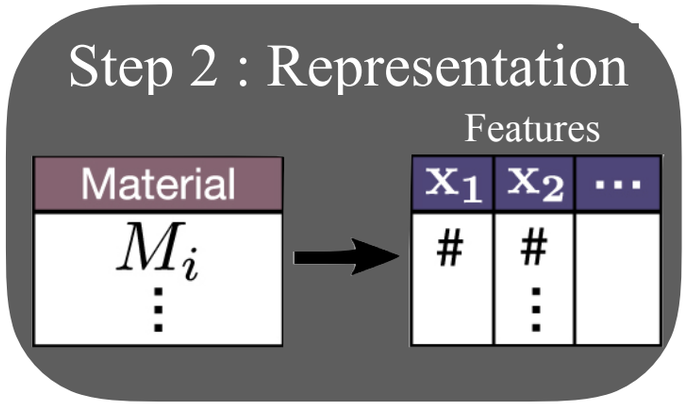
**Big Data of Materials Science: Critical Role of the Descriptor**

Luca M. Ghiringhelli,[1,*] Jan Vybiral,[2] Sergey V. Levchenko,[1] Claudia Draxl,[3] and Matthias Scheffler[1]
[1]*Fritz-Haber-Institut der Max-Planck-Gesellschaft, 14195 Berlin-Dahlem, Germany*
[2]*Department of Mathematical Analysis, Charles University, 18675 Prague, Czech Republic*
[3]*Humboldt-Universität zu Berlin, Institut für Physik and IRIS Adlershof, 12489 Berlin, Germany*
(Received 14 April 2014; revised manuscript received 20 October 2014; published 10 March 2015)

**X1 X2 X3**

| $Z_A$ | A | IP [eV] | EA [eV] | EN = −1/2(EA+IP) [eV] | Highest occ. KS level [eV] | Lowest unocc. KS level [eV] | $r_s$ [Å] | $r_p$ [Å] | $r_d$ [Å] |
|---|---|---|---|---|---|---|---|---|---|
| 3 | Li | -5.329 | -0.698 | 3.014 | -2.874 | -0.978 | 1.652 | 1.995 | 6.930 |
| 4 | Be | -9.459 | 0.631 | 4.414 | -5.600 | -2.098 | 1.078 | 1.211 | 2.877 |
| 5 | B | -8.190 | -0.107 | 4.149 | -3.715 | 2.248 | 0.805 | 0.826 | 1.946 |
| 6 | C | -10.852 | -0.872 | 5.862 | -5.416 | 1.992 | 0.644 | 0.630 | 1.631 |
| 7 | N | -13.585 | -1.867 | 7.726 | -7.239 | 3.057 | 0.539 | 0.511 | 1.540 |
| ⋮ | | | | | | | | | |
| 50 | Sn | -7.043 | -1.039 | 4.041 | -3.866 | 0.008 | 1.057 | 1.344 | 2.030 |
| 51 | Sb | -8.468 | -1.847 | 5.158 | -4.991 | 0.105 | 1.001 | 1.232 | 2.065 |
| 52 | Te | -9.867 | -2.666 | 6.266 | -6.109 | 0.099 | 0.945 | 1.141 | 1.827 |
| 53 | I | -11.257 | -3.513 | 7.385 | -7.236 | 0.213 | 0.896 | 1.071 | 1.722 |
| 55 | Cs | -4.006 | -0.570 | 2.288 | -2.220 | -0.548 | 2.464 | 3.164 | 1.974 |
| 56 | Ba | -5.516 | 0.278 | 2.619 | -3.346 | -2.129 | 2.149 | 2.632 | 1.351 |

# Step 2 : Representation




PRL 114, 105503 (2015) PHYSICAL REVIEW LETTERS week ending 13 MARCH 2015

**Big Data of Materials Science: Critical Role of the Descriptor**

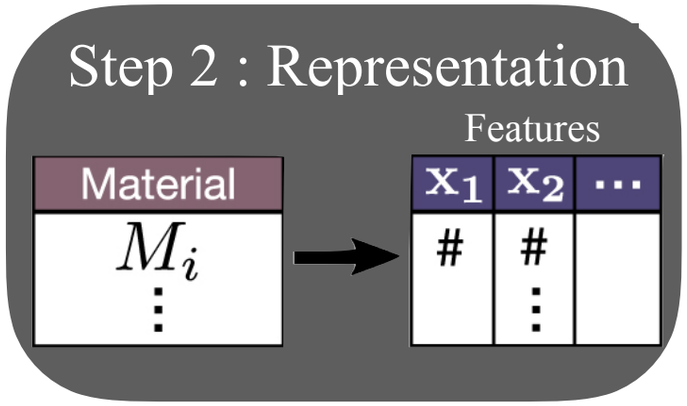
Luca M. Ghiringhelli,[1,*] Jan Vybiral,[2] Sergey V. Levchenko,[1] Claudia Draxl,[3] and Matthias Scheffler[1]
[1]*Fritz-Haber-Institut der Max-Planck-Gesellschaft, 14195 Berlin-Dahlem, Germany*
[2]*Department of Mathematical Analysis, Charles University, 18675 Prague, Czech Republic*
[3]*Humboldt-Universität zu Berlin, Institut für Physik and IRIS Adlershof, 12489 Berlin, Germany*
(Received 14 April 2014; revised manuscript received 20 October 2014; published 10 March 2015)

| X1 | X2 | X3 | X4 | X5 | X6 | X7 | X8 | X9 | X10 |
|---|---|---|---|---|---|---|---|---|---|
| $Z_A$ | A | IP [eV] | EA [eV] | EN = −1/2(EA+IP) [eV] | Highest occ. KS level [eV] | Lowest unocc. KS level [eV] | $r_s$ [Å] | $r_p$ [Å] | $r_d$ [Å] |
| 3 | Li | -5.329 | -0.698 | 3.014 | -2.874 | -0.978 | 1.652 | 1.995 | 6.930 |
| 4 | Be | -9.459 | 0.631 | 4.414 | -5.600 | -2.098 | 1.078 | 1.211 | 2.877 |
| 5 | B | -8.190 | -0.107 | 4.149 | -3.715 | 2.248 | 0.805 | 0.826 | 1.946 |
| 6 | C | -10.852 | -0.872 | 5.862 | -5.416 | 1.992 | 0.644 | 0.630 | 1.631 |
| 7 | N | -13.585 | -1.867 | 7.726 | -7.239 | 3.057 | 0.539 | 0.511 | 1.540 |
| ⋮ | | | | | | | | | |
| 50 | Sn | -7.043 | -1.039 | 4.041 | -3.866 | 0.008 | 1.057 | 1.344 | 2.030 |
| 51 | Sb | -8.468 | -1.847 | 5.158 | -4.991 | 0.105 | 1.001 | 1.232 | 2.065 |
| 52 | Te | -9.867 | -2.666 | 6.266 | -6.109 | 0.099 | 0.945 | 1.141 | 1.827 |
| 53 | I | -11.257 | -3.513 | 7.385 | -7.236 | 0.213 | 0.896 | 1.071 | 1.722 |
| 55 | Cs | -4.006 | -0.570 | 2.288 | -2.220 | -0.548 | 2.464 | 3.164 | 1.974 |
| 56 | Ba | -5.516 | 0.278 | 2.619 | -3.346 | -2.129 | 2.149 | 2.632 | 1.351 |

**Feature space**

**(*Fingerprint*)**

# Step 2 : Representation




PRL **114,** 105503 (2015) PHYSICAL REVIEW LETTERS week ending 13 MARCH 2015

**Big Data of Materials Science: Critical Role of the Descriptor**

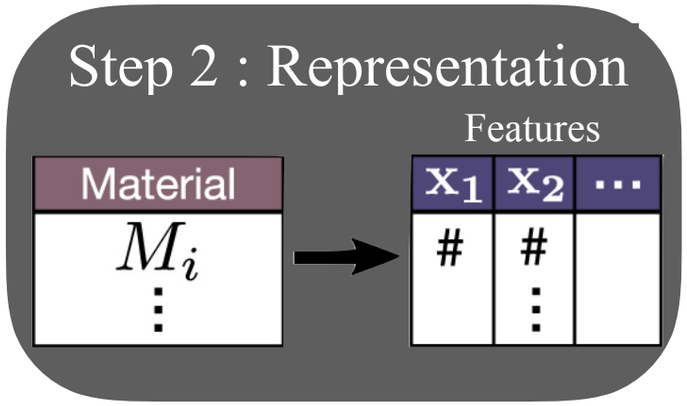
Luca M. Ghiringhelli,[1,*] Jan Vybiral,[2] Sergey V. Levchenko,[1] Claudia Draxl,[3] and Matthias Scheffler[1]
[1]*Fritz-Haber-Institut der Max-Planck-Gesellschaft, 14195 Berlin-Dahlem, Germany*
[2]*Department of Mathematical Analysis, Charles University, 18675 Prague, Czech Republic*
[3]*Humboldt-Universität zu Berlin, Institut für Physik and IRIS Adlershof, 12489 Berlin, Germany*
(Received 14 April 2014; revised manuscript received 20 October 2014; published 10 March 2015)

TABLE I. Dependence of the metal-insulator classification descriptors on the prototypes of training binary materials.

| prototypes | #materials | primary features | descriptor | classification accuracy |
|---|---|---|---|---|
| NaCl | 132 | $IE_A$, $IE_B$, $\chi_A$, $\chi_B$, $r_{\mathrm{covA}}$, $r_{\mathrm{covB}}$, $EA_A$, $EA_B$, $v_A$, $v_B$, $d_{AB}$ | $d_1 := \frac{IE_A IE_B (d_{AB} - r_{\mathrm{covA}})}{\exp(\chi_A)\sqrt{r_{\mathrm{covB}}}}$ | 100% |
| NaCl, CsCl, ZnS, $CaF_2$, $Cr_3Si$ | 217 | $IE_A$, $IE_B$, $\chi_A$, $\chi_B$, $r_{\mathrm{covA}}$, $r_{\mathrm{covB}}$, $d_{AB}$, $CN_A$, $CN_B$ | $d_1 := \frac{IE_B d_{AB}^2}{\chi_A r_{\mathrm{covA}}^2 \sqrt{CN_B}}$, $d_2 := \frac{IE_A^2 r_{\mathrm{covB}} \log(IE_A) \lvert r_{\mathrm{covA}} - r_{\mathrm{covB}} \rvert}{CN_B}$ | 100% |
| NaCl, CsCl, ZnS, $CaF_2$, $Cr_3Si$, SiC, $TiO_2$, ZnO, FeAs, NiAs | 260 | $IE_A$, $IE_B$, $\chi_A$, $\chi_B$, $r_{\mathrm{covA}}$, $r_{\mathrm{covB}}$, $d_{AB}$, $CN_A$, $CN_B$ | $d_1 := \frac{d_{AB}/r_{\mathrm{covA}} - \chi_A/\chi_B}{\exp(CN_B/IE_B)}$, $d_2 := \frac{r_{\mathrm{covA}}^3 d_{AB} IE_B}{\lvert \chi_B/\chi_A - \lvert CN_B - CN_A \rvert \rvert}$ | 99.6%[a] |
| NaCl, CsCl, ZnS, $CaF_2$, $Cr_3Si$, SiC, $TiO_2$, ZnO, FeAs, NiAs | 260 | $IE_A$, $IE_B$, $\chi_A$, $\chi_B$, $x_A$, $x_B$, $V_{\mathrm{cell}}/\sum V_{\mathrm{atom}}$ | $d_1 := \frac{V_{\mathrm{cell}}}{\sum V_{\mathrm{atom}}} \frac{\sqrt{\chi_B}}{\chi_A}$, $d_2 := \frac{IE_A IE_B}{\exp(V_{\mathrm{cell}}/\sum V_{\mathrm{atom}})}$ | 99.6%[a] |
| NaCl, CsCl, ZnS, $CaF_2$, $Cr_3Si$, SiC, $TiO_2$, ZnO, FeAs, NiAs, $Al_2O_3$, $La_2O_3$, $Th_3P_4$, $ReO_3$, $ThH_2$ | 299 | $IE_A$, $IE_B$, $\chi_A$, $\chi_B$, $x_A$, $x_B$, $V_{\mathrm{cell}}/\sum V_{\mathrm{atom}}$ | $d_1 := \frac{x_B}{\sum V_{\mathrm{atom}}/V_{\mathrm{cell}}} \frac{IE_B \sqrt{\chi_B}}{\chi_A}$, $d_2 := \chi_A^2 \lvert \lvert 1 - 2x_A \rvert - x_A^2 \frac{\chi_B}{\chi_A} \rvert$ | 99.0%[b] |

Runhai Ouyang, *et. al.* Phys. Rev. Mat. **2**, 083802 (2018).

# Step 2 : Representation




## Atomic properties

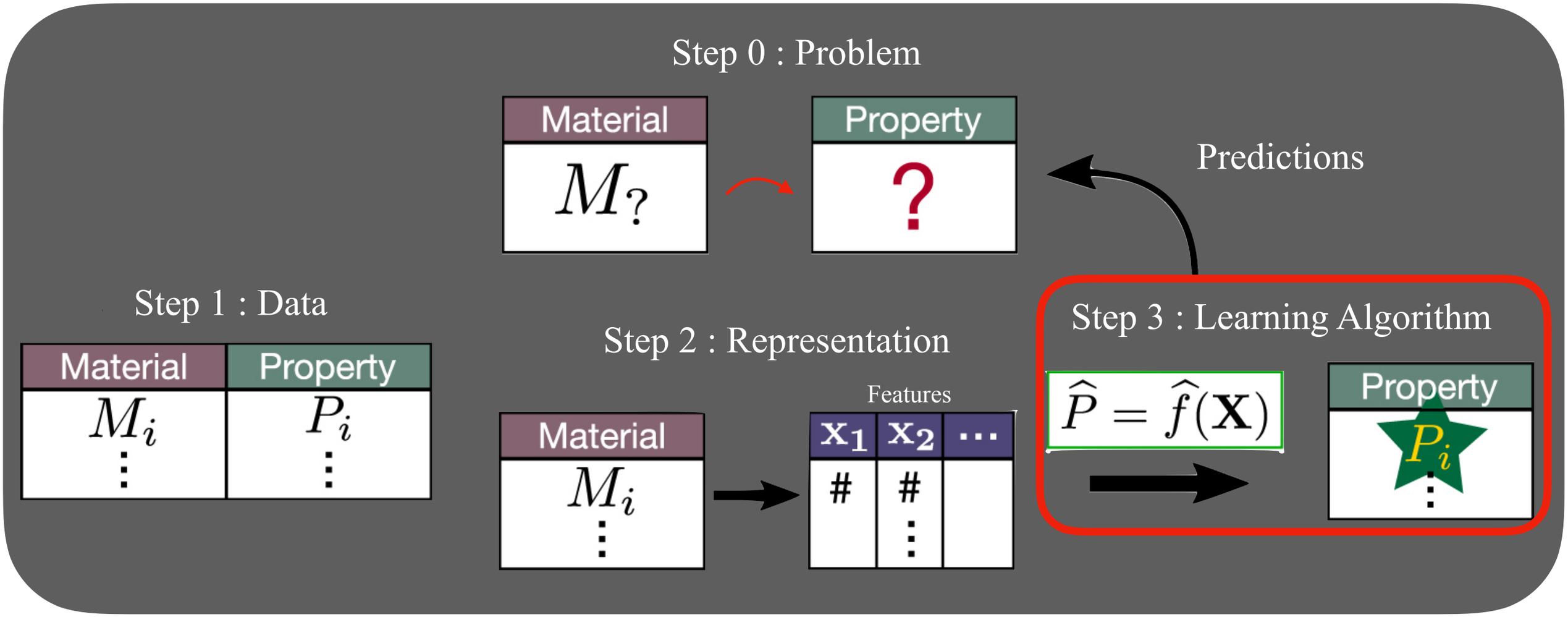
Z
EA
$r_p$
$r_s$
LUMO
HOMO
IP

## Operations

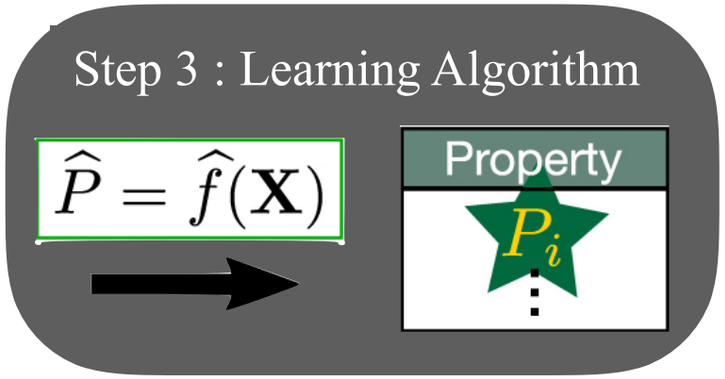
+
*
ln
-
/
$(\ )^2$
Exp.
$(\ )^3$

## Feature Space

$H_{ks}(A)+H_{ks}(B)+\ldots$
$L_{ks}(A)+L_{KS}(B)+\ldots$
$r_s(A)+r_s(B)+..$
$(r_s(A))^2,(L_{ks}(A))^2$
$(IP(A))^2/r_S(A) +r_s(B)$
……..


Step 0 : Problem
Material
$M_?$
Property
?
Predictions
Step 1 : Data
Material
Property
$M_i$
$P_i$
Step 2 : Representation
Material
$M_i$
Features
$x_1$
$x_2$
...
#
#
Step 3 : Learning Algorithm
$\widehat{P} = \widehat{f}(\mathbf{X})$
Property
$P_i$

# Step 3 : Learning Algorithms

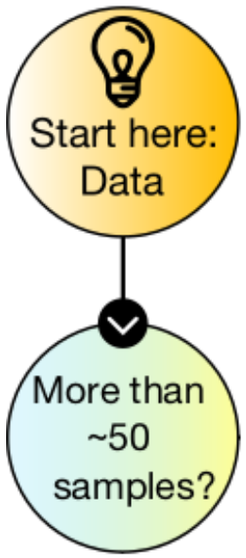



- Unsupervised
  - Principal component analysis (PCA)
  - Clustering

- Supervised
  - Linear Regression
  - Logistic Regression
  - Naïve Bayes
  - K-NN
  - Decision Trees

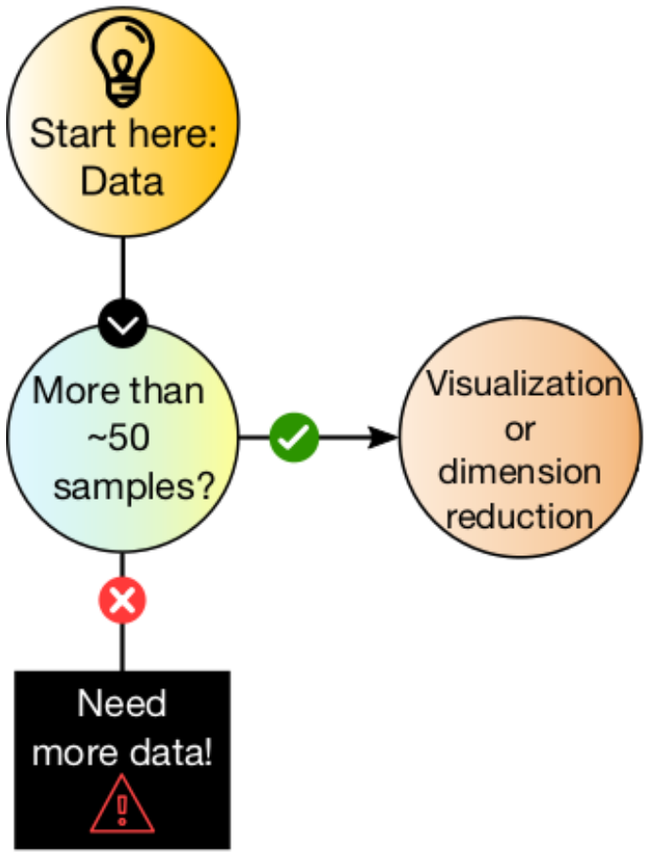
# Step 3 : Learning Algorithms

# Step 3 : Learning Algorithms

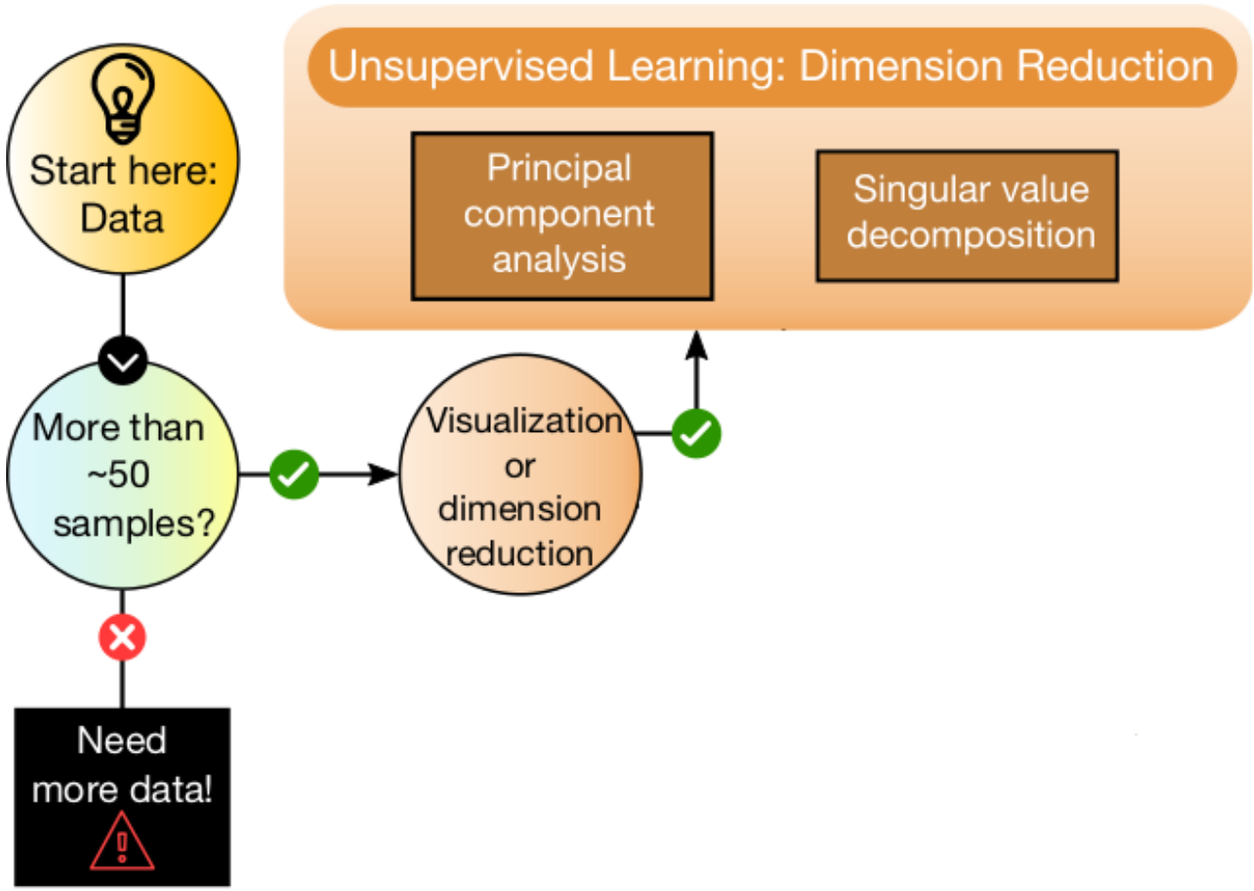


# Step 3 : Learning Algorithms

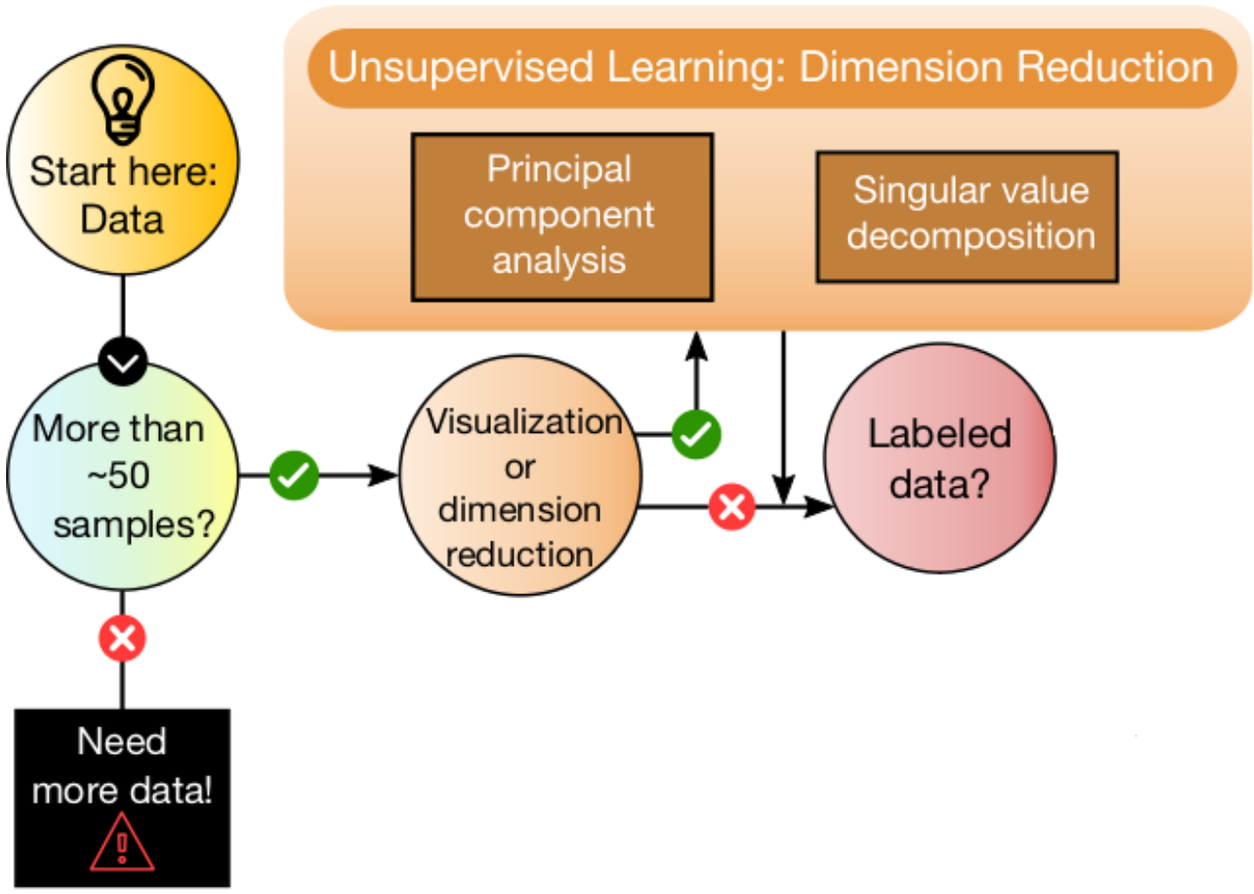
# Step 3 : Learning Algorithms



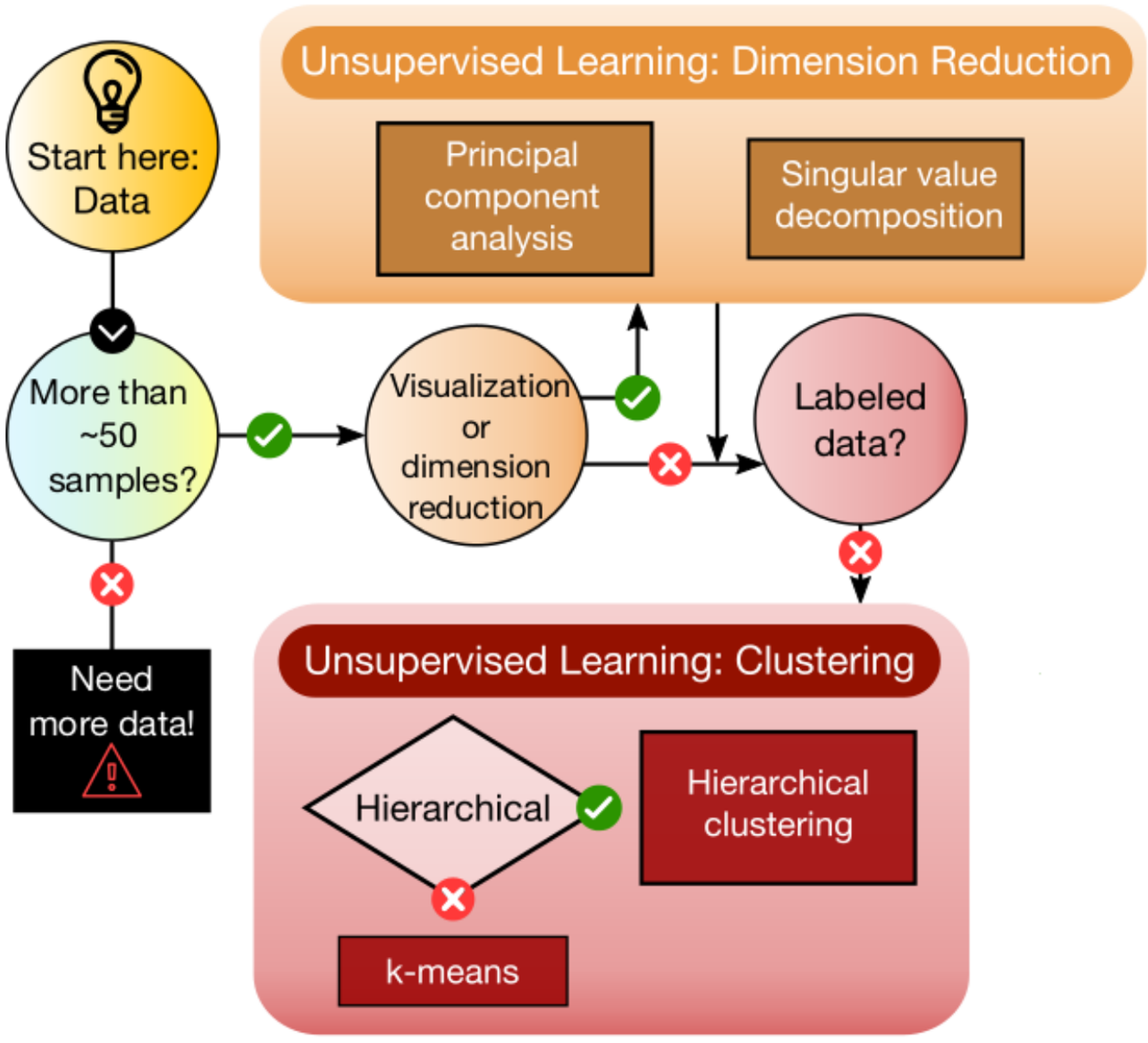
# Step 3 : Learning Algorithms



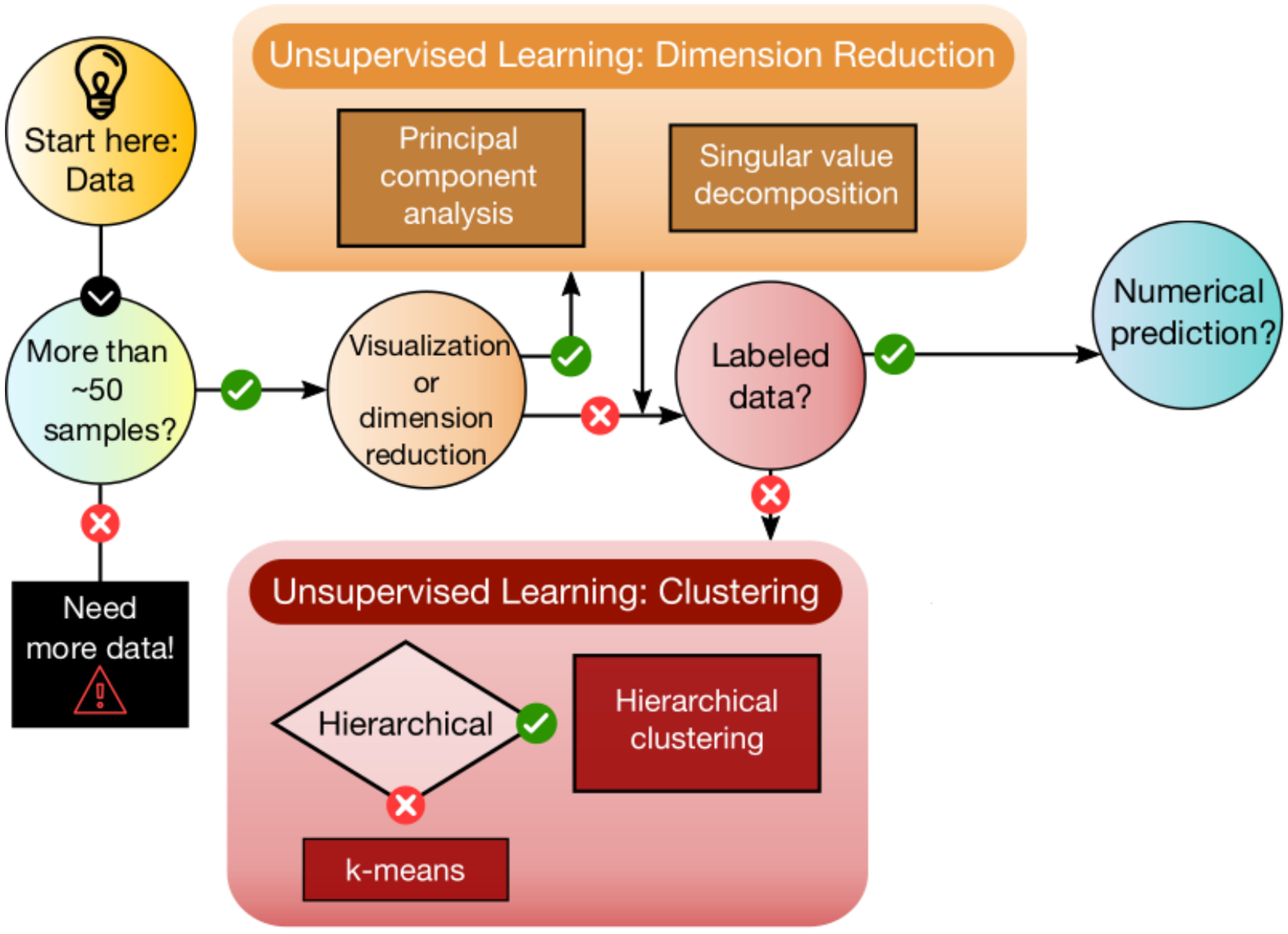
# Step 3 : Learning Algorithms



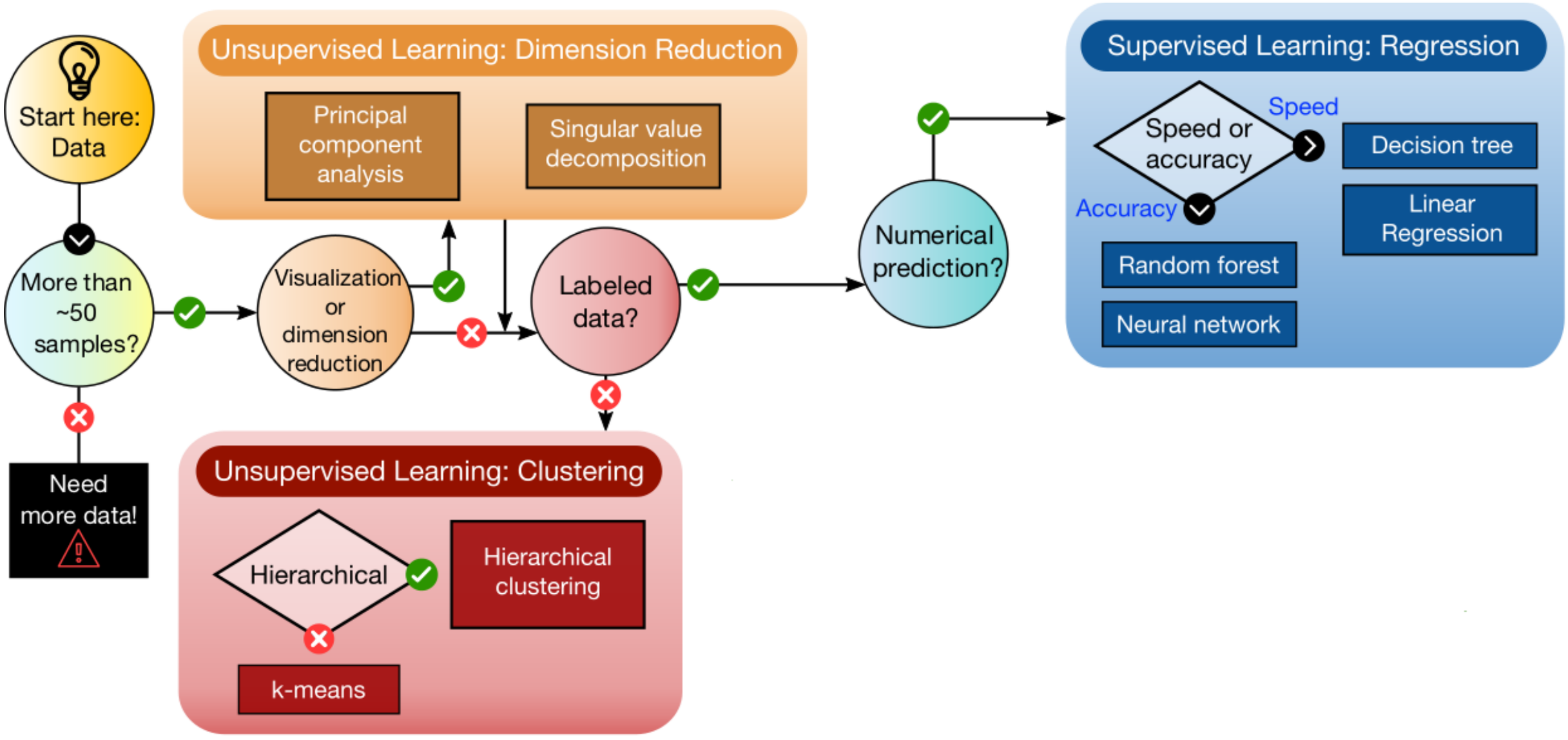
# Step 3 : Learning Algorithms



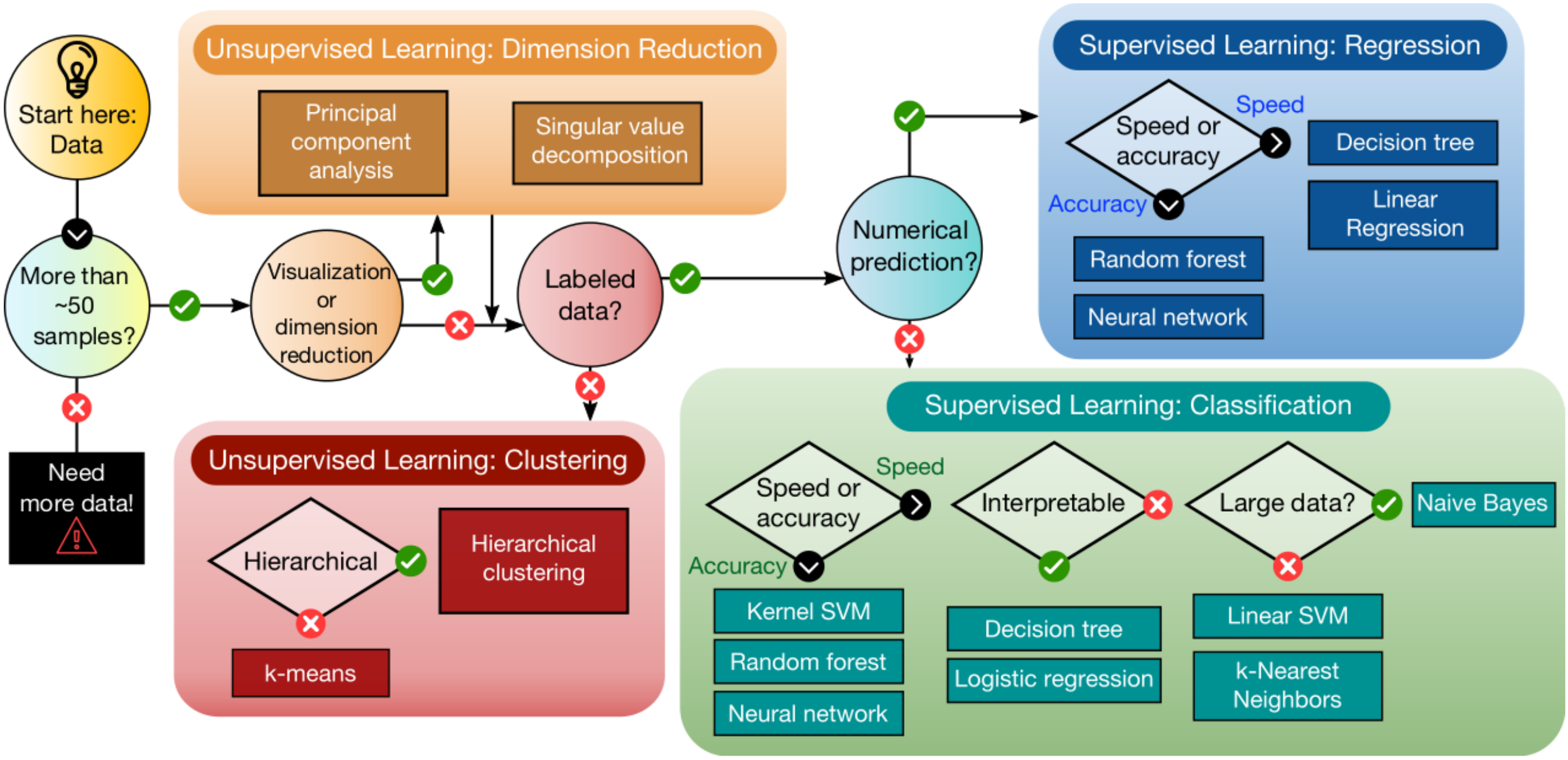
# Step 3 : Learning Algorithms



# Step 3 : Learning Algorithms



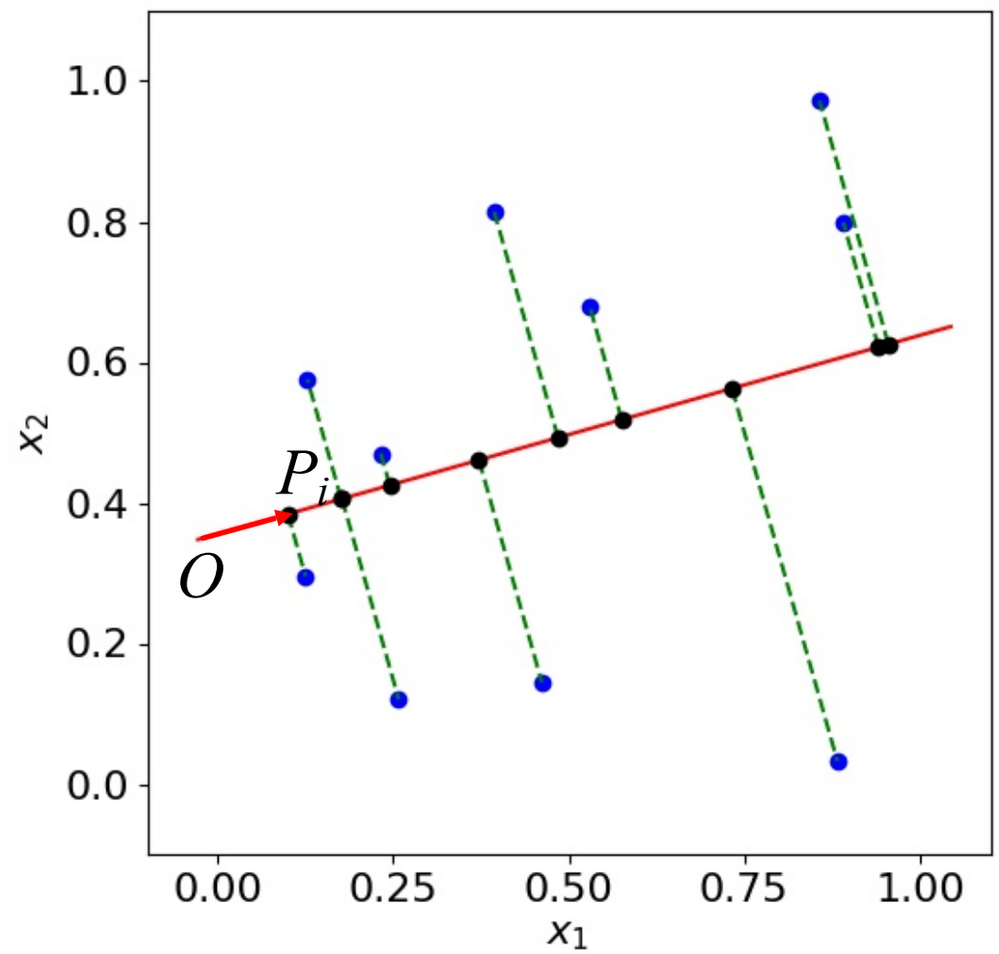
# Step 3 : Learning Algorithms



# Step 3 : Learning Algorithms (unsupervised)



## Principal component Analysis (PCA)

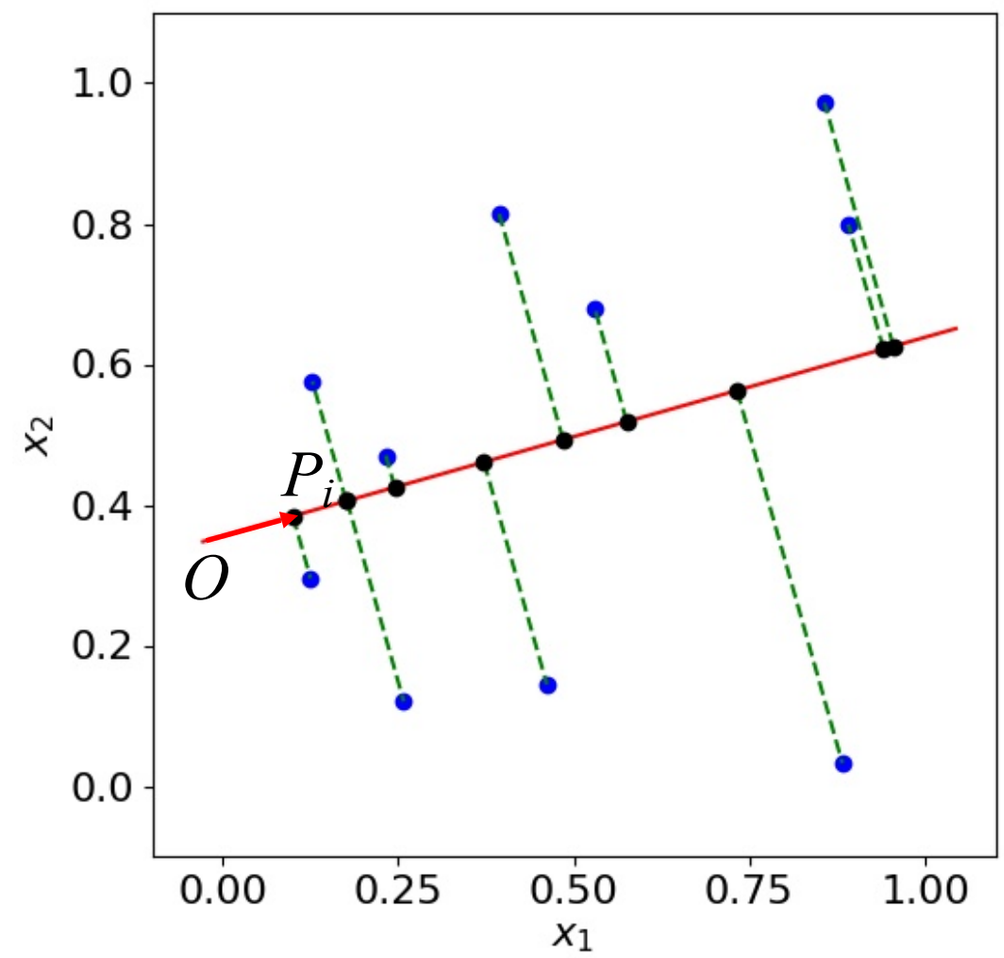


# Step 3 : Learning Algorithms (unsupervised)



## Principal component Analysis (PCA)

$$X = \begin{pmatrix} x_{11} & x_{12} & \dots & x_{1n} \\ x_{21} & x_{22} & \dots & x_{2n} \\ \vdots & \vdots & & \vdots \\ x_{p1} & x_{p2} & \dots & x_{pn} \end{pmatrix}$$

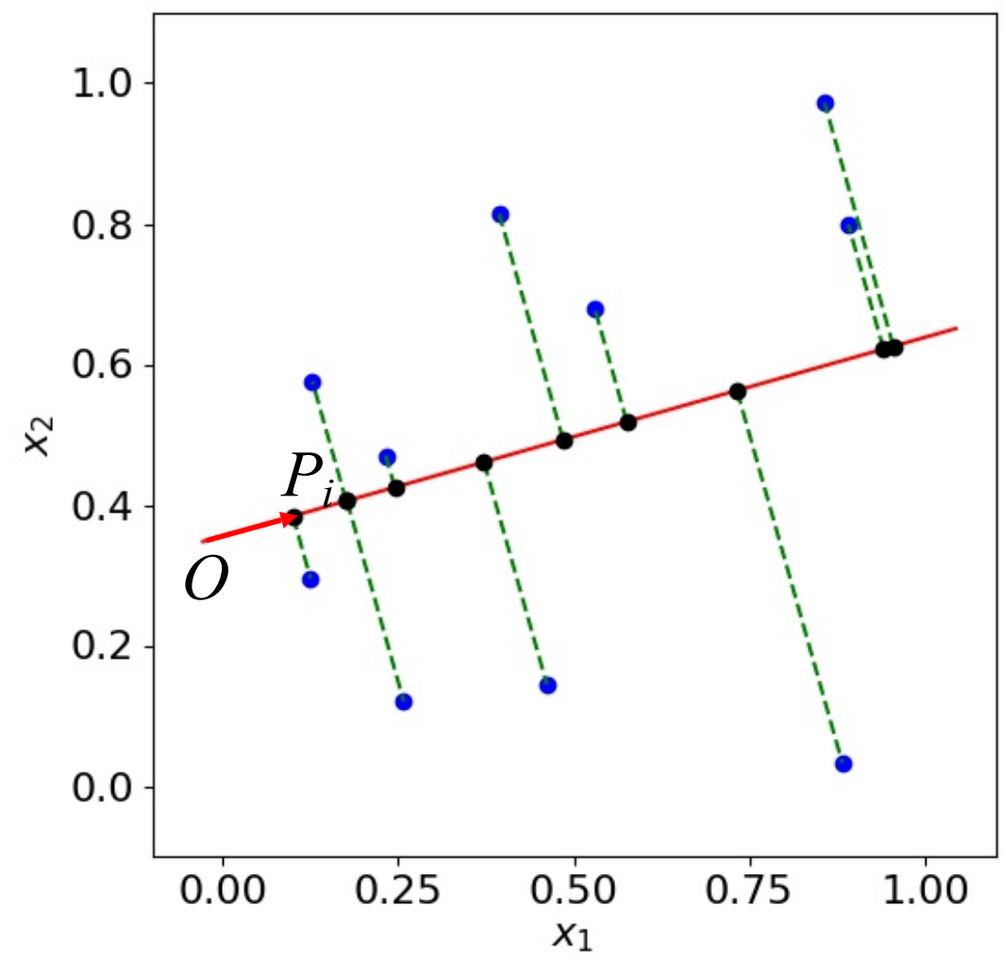


# Step 3 : Learning Algorithms (unsupervised)



## Principal component Analysis (PCA)

$$X = \begin{pmatrix} x_{11} & x_{12} & \dots & x_{1n} \\ x_{21} & x_{22} & \dots & x_{2n} \\ \vdots & \vdots & & \vdots \\ x_{p1} & x_{p2} & \dots & x_{pn} \end{pmatrix}$$



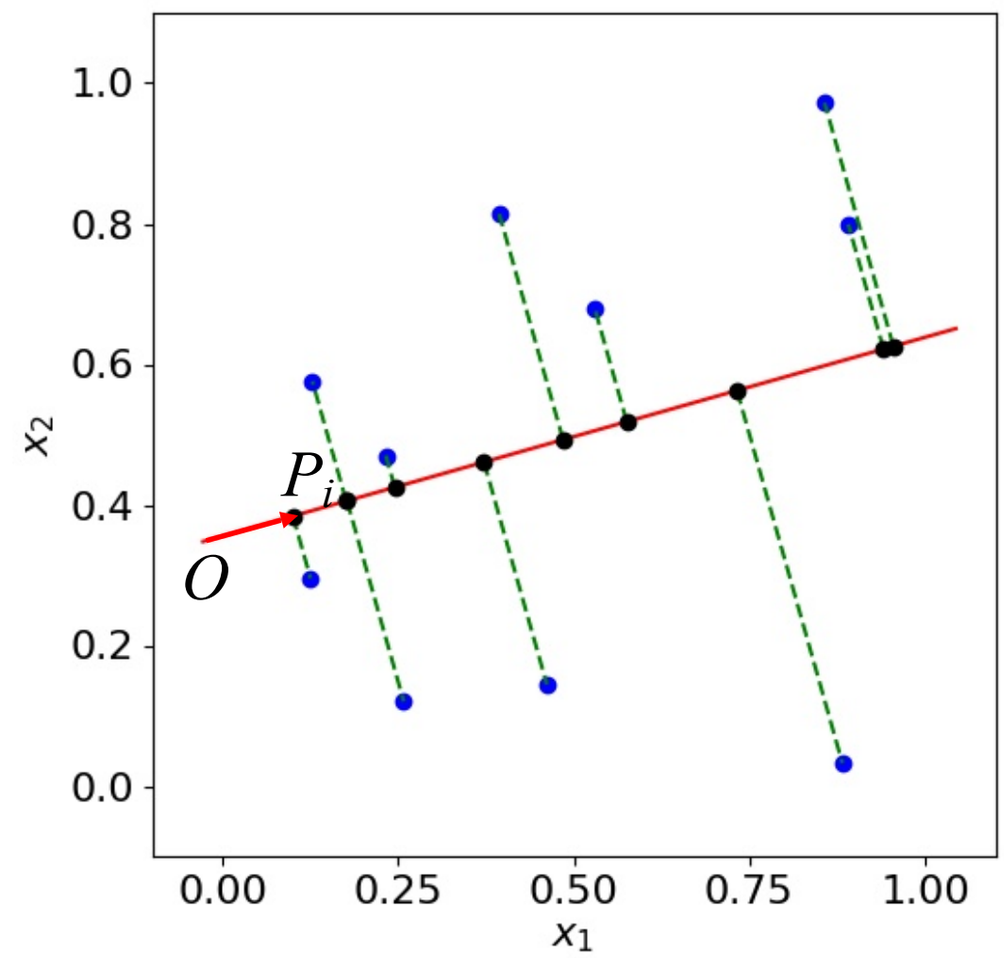
# Step 3 : Learning Algorithms (unsupervised)



## Principal component Analysis (PCA)

$$X = \begin{pmatrix} x_{11} & x_{12} & \dots & x_{1n} \\ x_{21} & x_{22} & \dots & x_{2n} \\ \vdots & \vdots & & \vdots \\ x_{p1} & x_{p2} & \dots & x_{pn} \end{pmatrix}$$

$$\sum_i \overline{OP_i}^2 = \sum_i c_i^T \cdot c_i = u^T X^T X u = u^T A u$$



# Step 3 : Learning Algorithms (unsupervised)

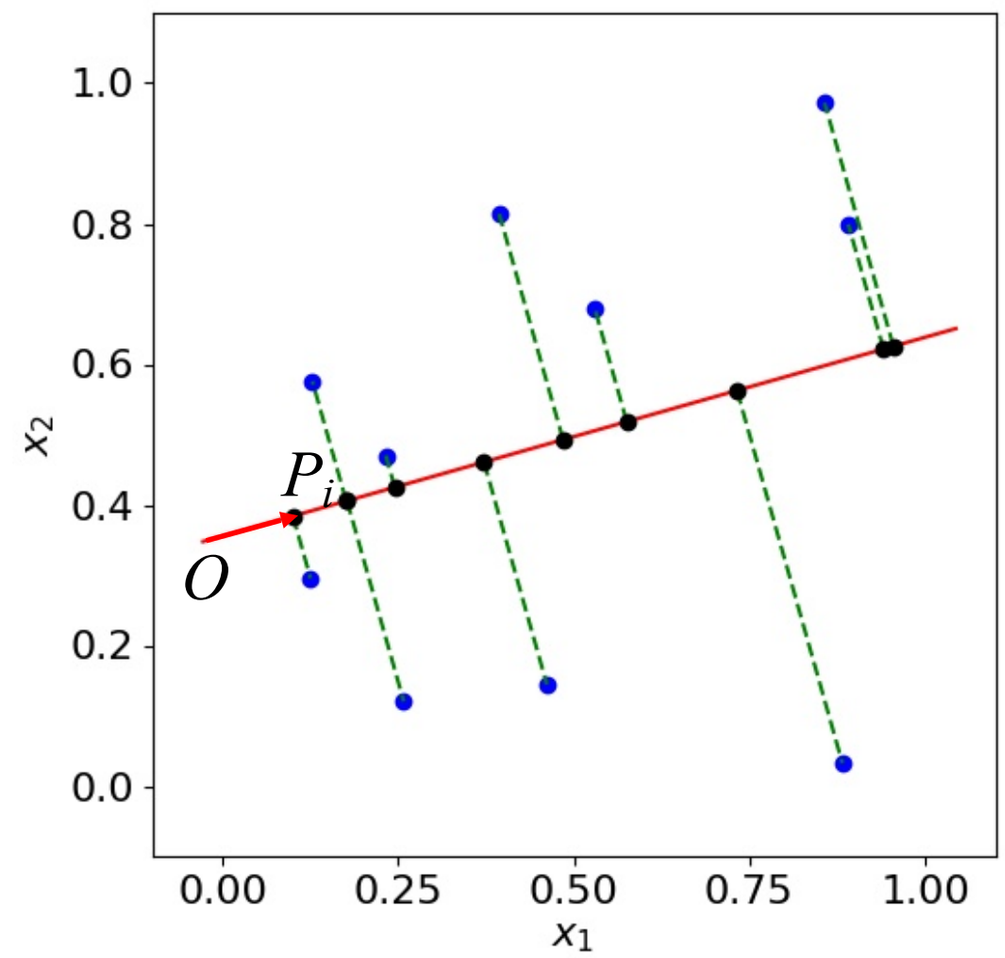


## Principal component Analysis (PCA)

$$X = \begin{pmatrix} x_{11} & x_{12} & \dots & x_{1n} \\ x_{21} & x_{22} & \dots & x_{2n} \\ \vdots & \vdots & & \vdots \\ x_{p1} & x_{p2} & \dots & x_{pn} \end{pmatrix}$$

maximze it!

$$\sum_i \overline{OP_i}^2 = \sum_i c_i^T \cdot c_i = u^T X^T X u = u^T A u$$



# Step 3 : Learning Algorithms (unsupervised)



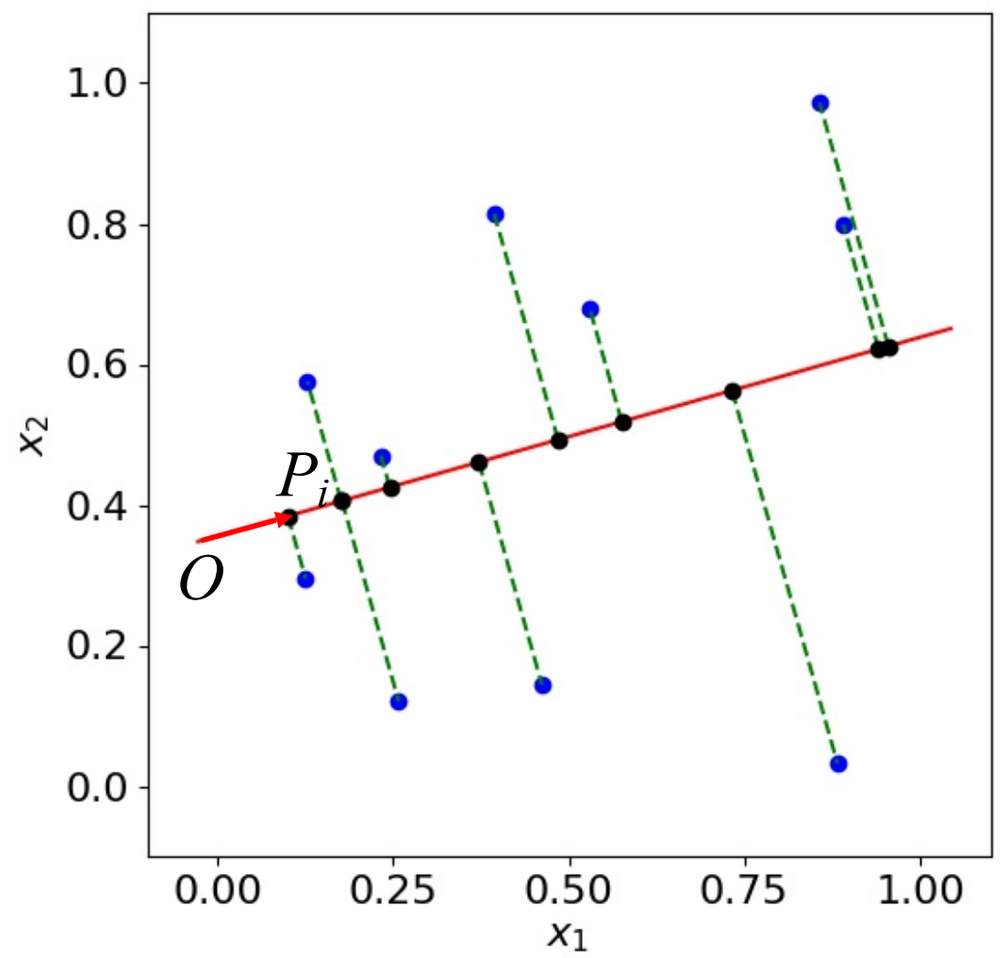
## Principal component Analysis (PCA)

$$X = \begin{pmatrix} x_{11} & x_{12} & \dots & x_{1n} \\ x_{21} & x_{22} & \dots & x_{2n} \\ \vdots & \vdots & & \vdots \\ x_{p1} & x_{p2} & \dots & x_{pn} \end{pmatrix}$$

maximze it!

$$\sum_i \overline{OP_i}^2 = \sum_i c_i^T \cdot c_i = u^T X^T X u = u^T A u$$

$$A u_i = \lambda u_i$$



# Step 3 : Learning Algorithms (unsupervised)



## Principal component Analysis (PCA)

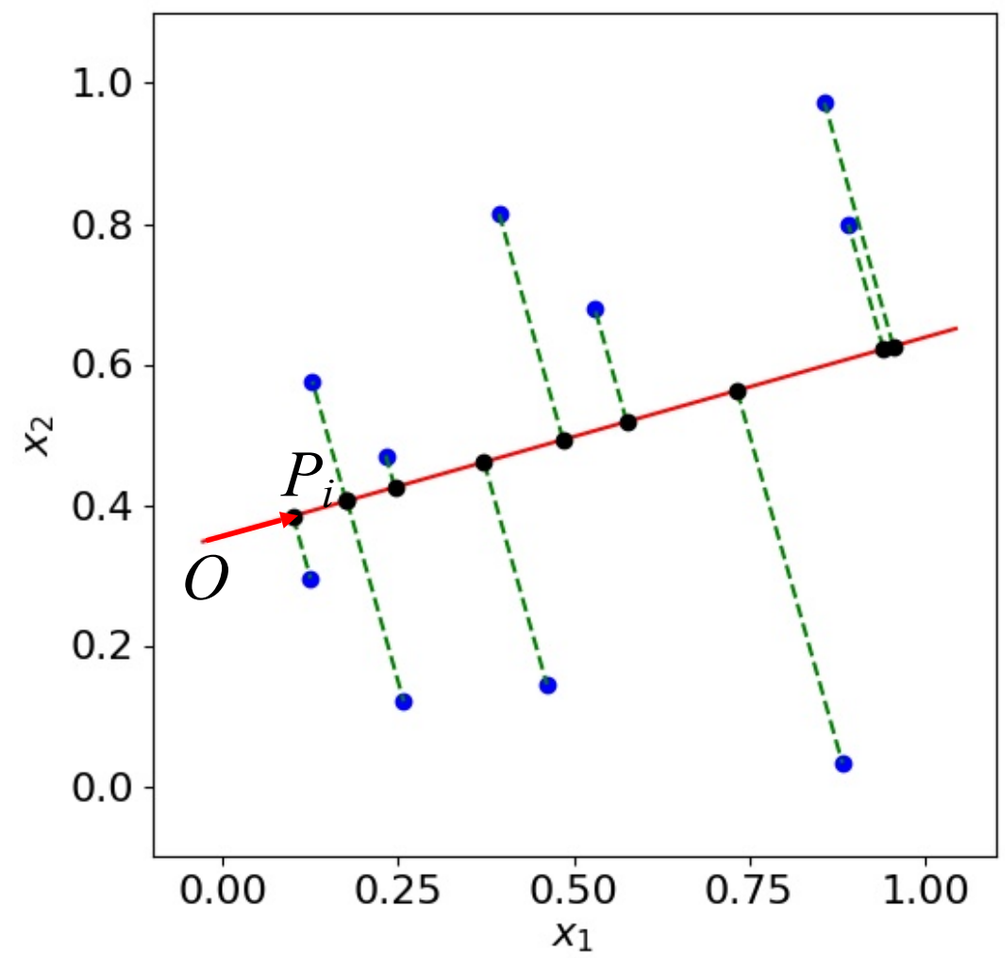
$$X = \begin{pmatrix} x_{11} & x_{12} & \dots & x_{1n} \\ x_{21} & x_{22} & \dots & x_{2n} \\ \vdots & \vdots & & \vdots \\ x_{p1} & x_{p2} & \dots & x_{pn} \end{pmatrix}$$

maximze it!

$$\sum_i \overline{OP_i}^2 = \sum_i c_i^T \cdot c_i = u^T X^T X u = u^T A u$$

$$A u_i = \lambda u_i \qquad\qquad U^T A U = \Lambda$$

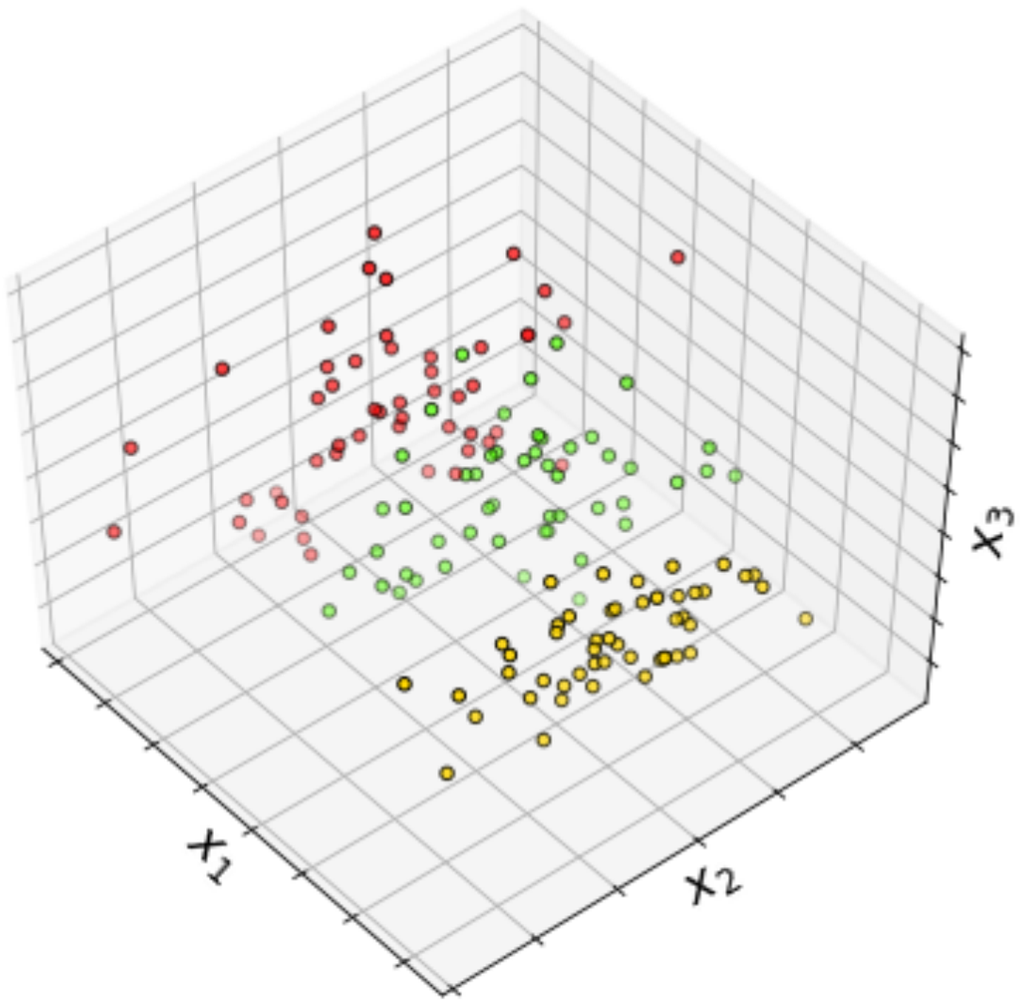
Eigenvalues: variance
Eigenvectors: corresponding direction



# Step 3 : Learning Algorithms (unsupervised)



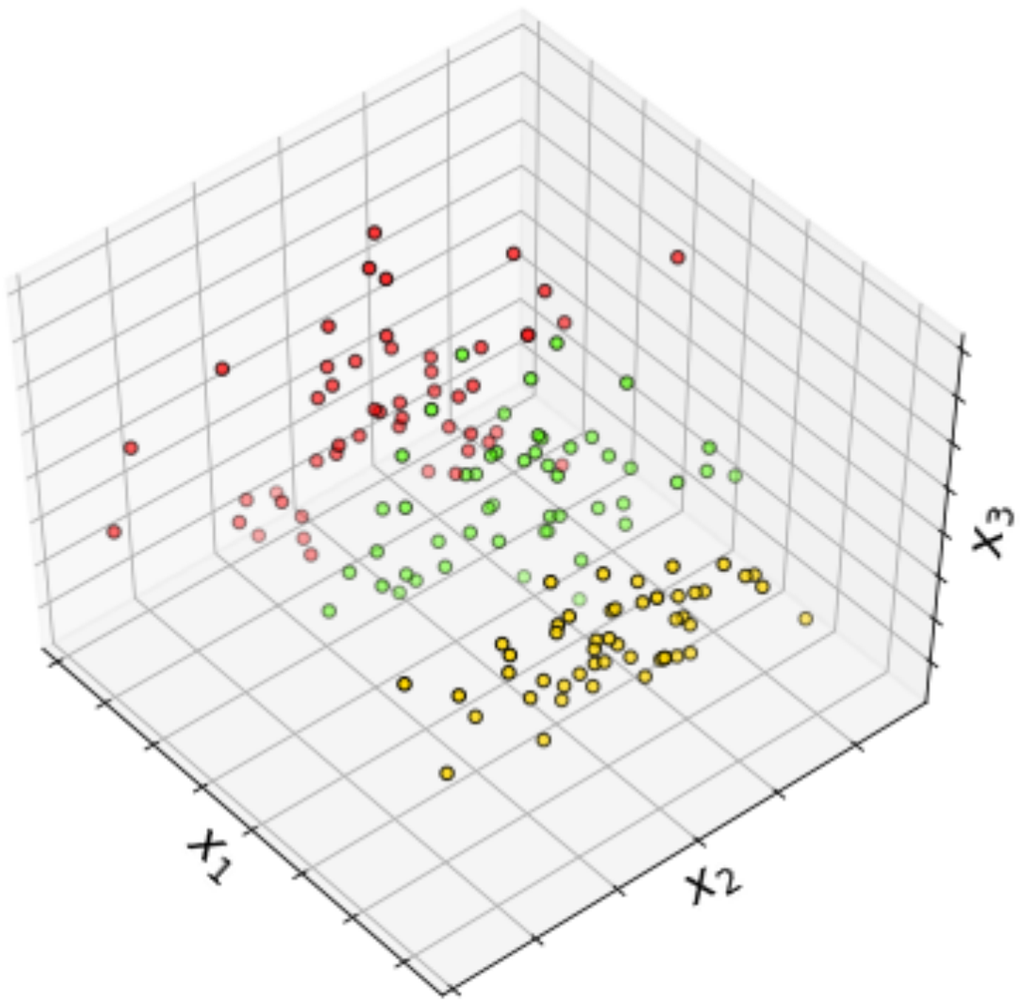
## Principal component Analysis (PCA)

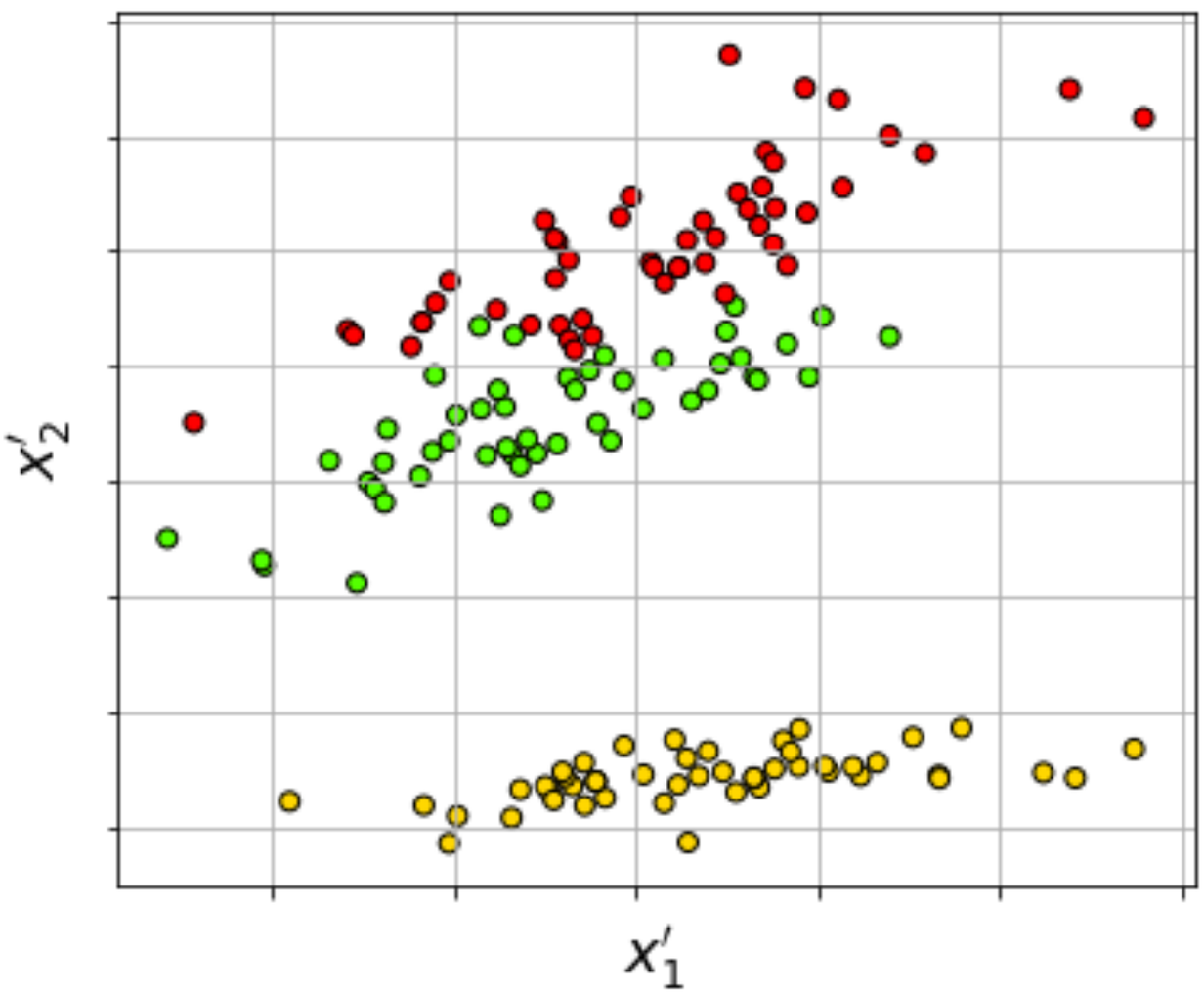


# Step 3 : Learning Algorithms (unsupervised)



## Principal component Analysis (PCA)



# Step 3 : Learning Algorithms (unsupervised)



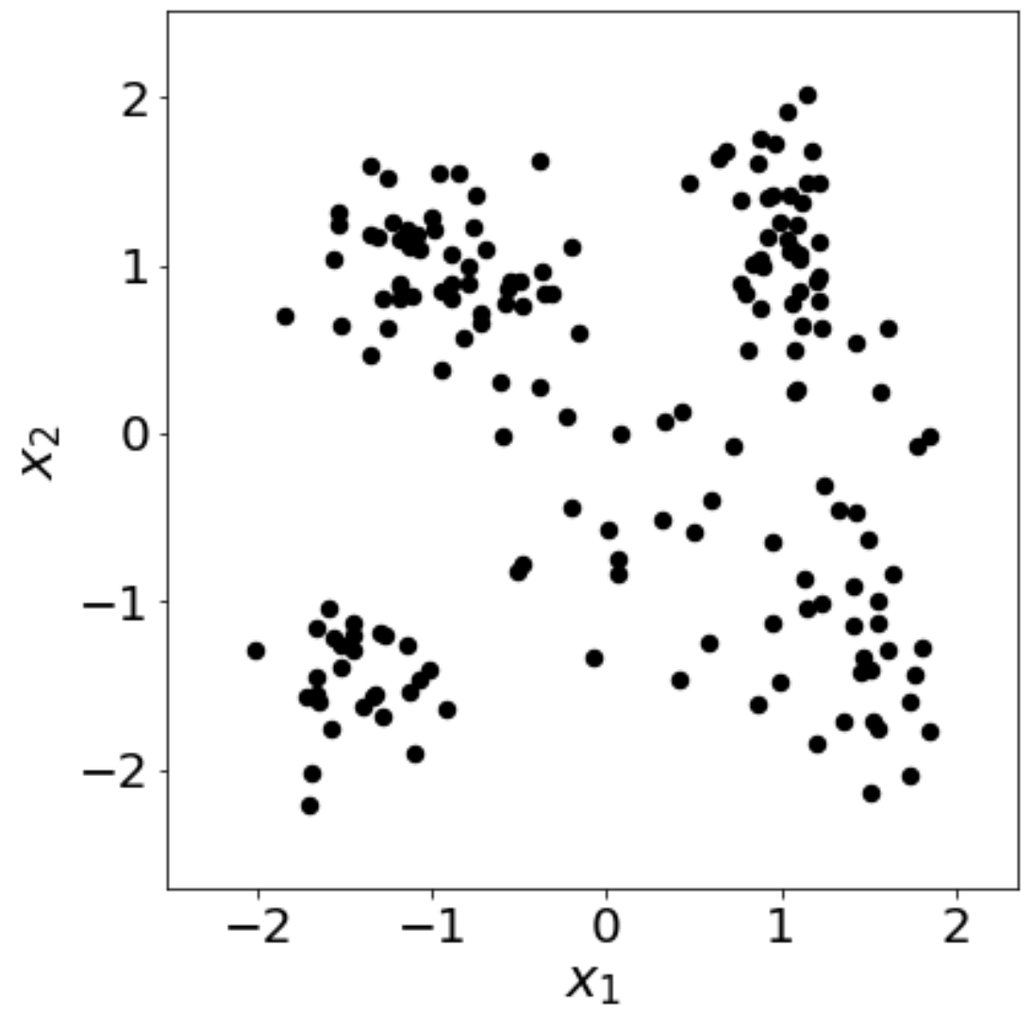
## Clustering

# Step 3 : Learning Algorithms (unsupervised)



## Clustering

- k-means



# Step 3 : Learning Algorithms (unsupervised)



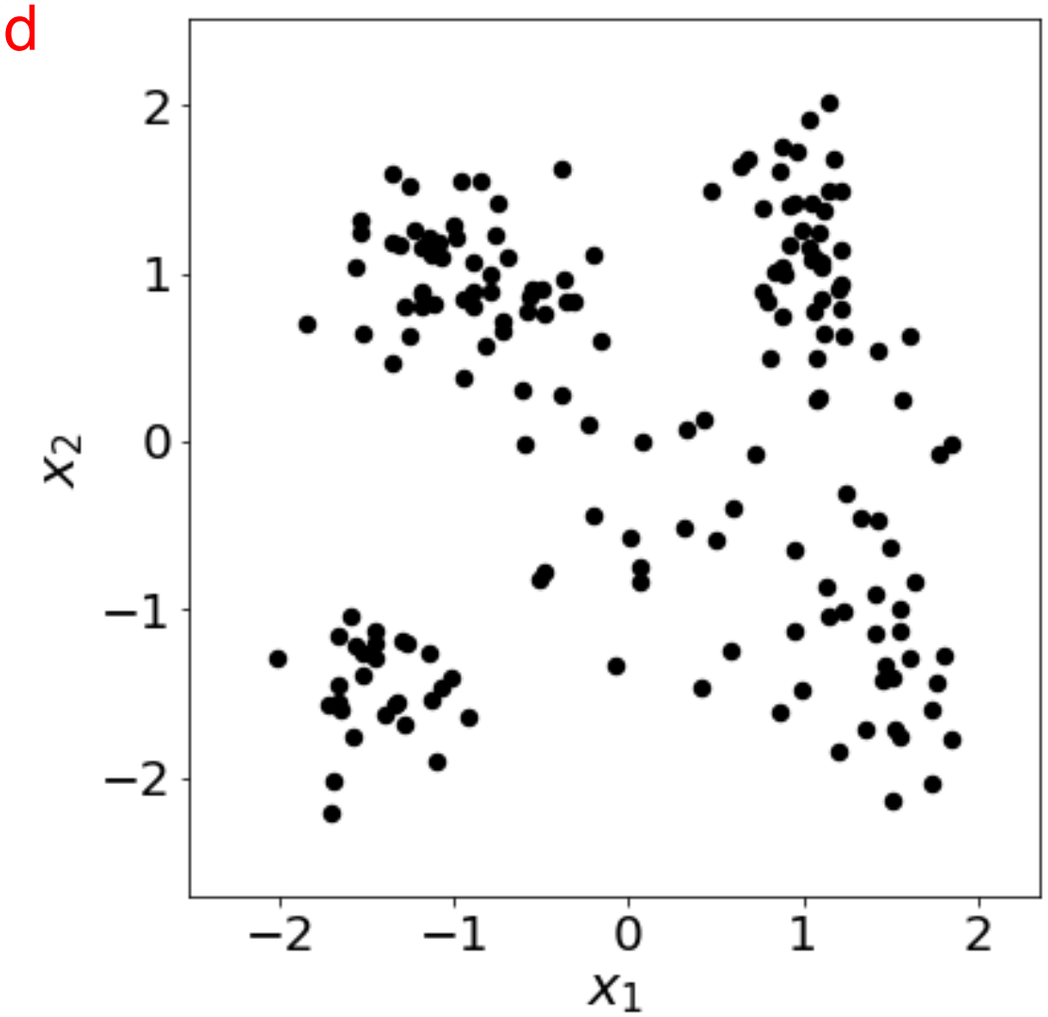
## Clustering

- k-means

i) Assign labels according to closest centroid

$$y_t^{(i)} = \text{argmin}_j \|\mathbf{x}^{(i)} - \boldsymbol{\mu}_t^{(j)}\|_p$$

$$\|\mathbf{x} - \boldsymbol{\mu}\|_p = \left[\sum_j (x_j - \mu_j)^p\right]^{\frac{1}{p}}$$



# Step 3 : Learning Algorithms (unsupervised)



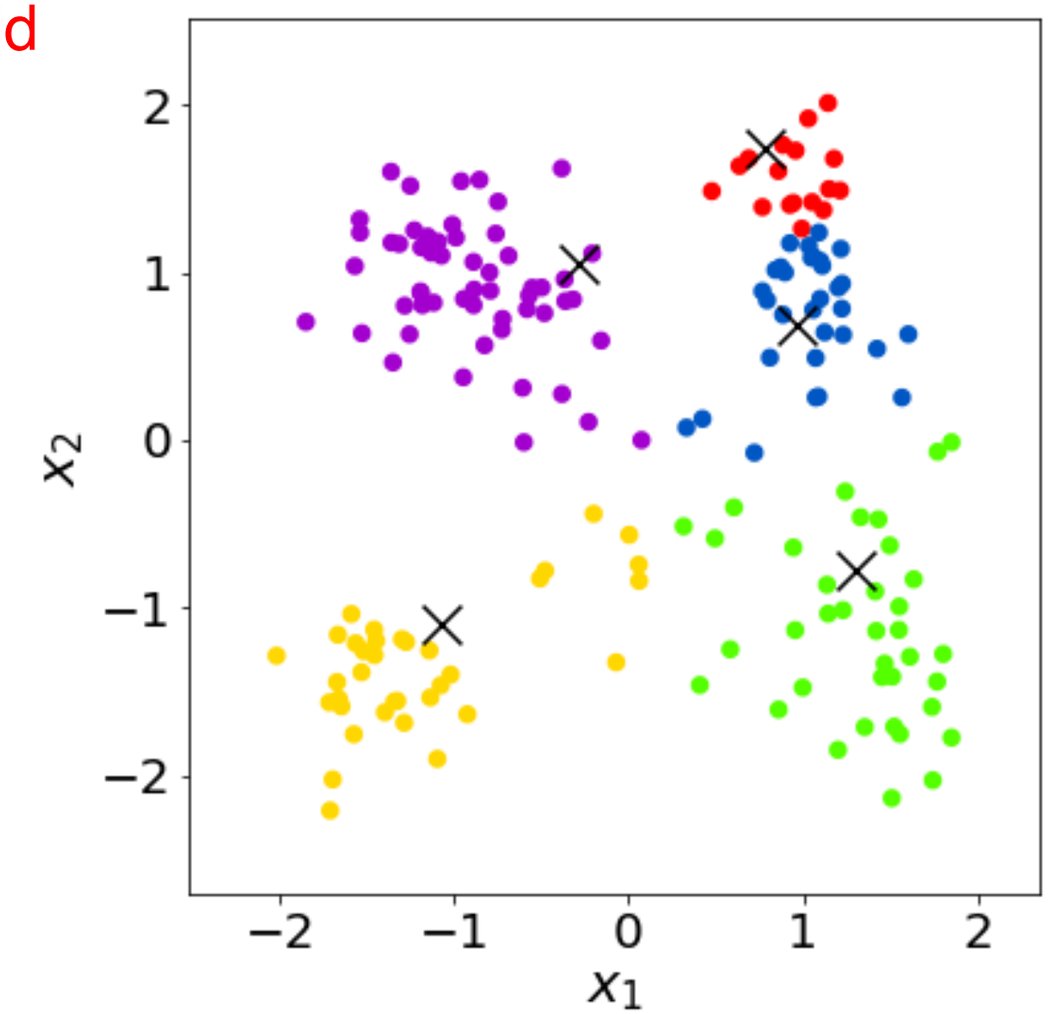
## Clustering

- k-means

i) Assign labels according to closest centroid

$$y_t^{(i)} = \text{argmin}_j \|\mathbf{x}^{(i)} - \boldsymbol{\mu}_t^{(j)}\|_p$$

$$\|\mathbf{x} - \boldsymbol{\mu}\|_p = \left[\sum_j (x_j - \mu_j)^p\right]^{\frac{1}{p}}$$



# Step 3 : Learning Algorithms (unsupervised)



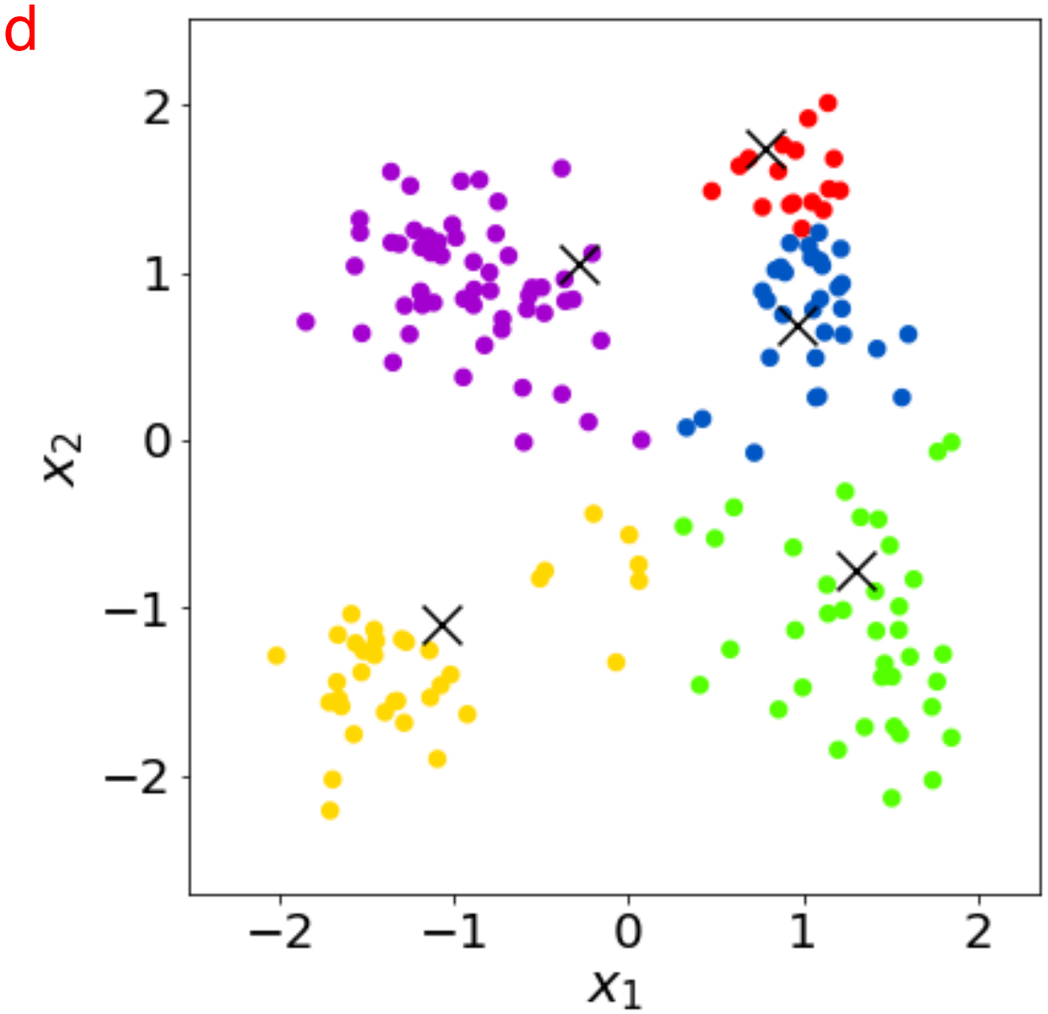
## Clustering

- k-means

i) Assign labels according to closest centroid

$$y_t^{(i)} = \mathrm{argmin}_j \|\mathbf{x}^{(i)} - \boldsymbol{\mu}_t^{(j)}\|_p$$

$$\|\mathbf{x} - \boldsymbol{\mu}\|_p = \left[\sum_j (x_j - \mu_j)^p\right]^{\frac{1}{p}}$$




ii) Update centroid's coords

$$\boldsymbol{\mu}_{t+1}^{(j)} = \frac{1}{n_j} \sum_i \mathbf{x}^{(i)} \delta_{y_t^{(i)},j}$$

# Step 3 : Learning Algorithms (unsupervised)



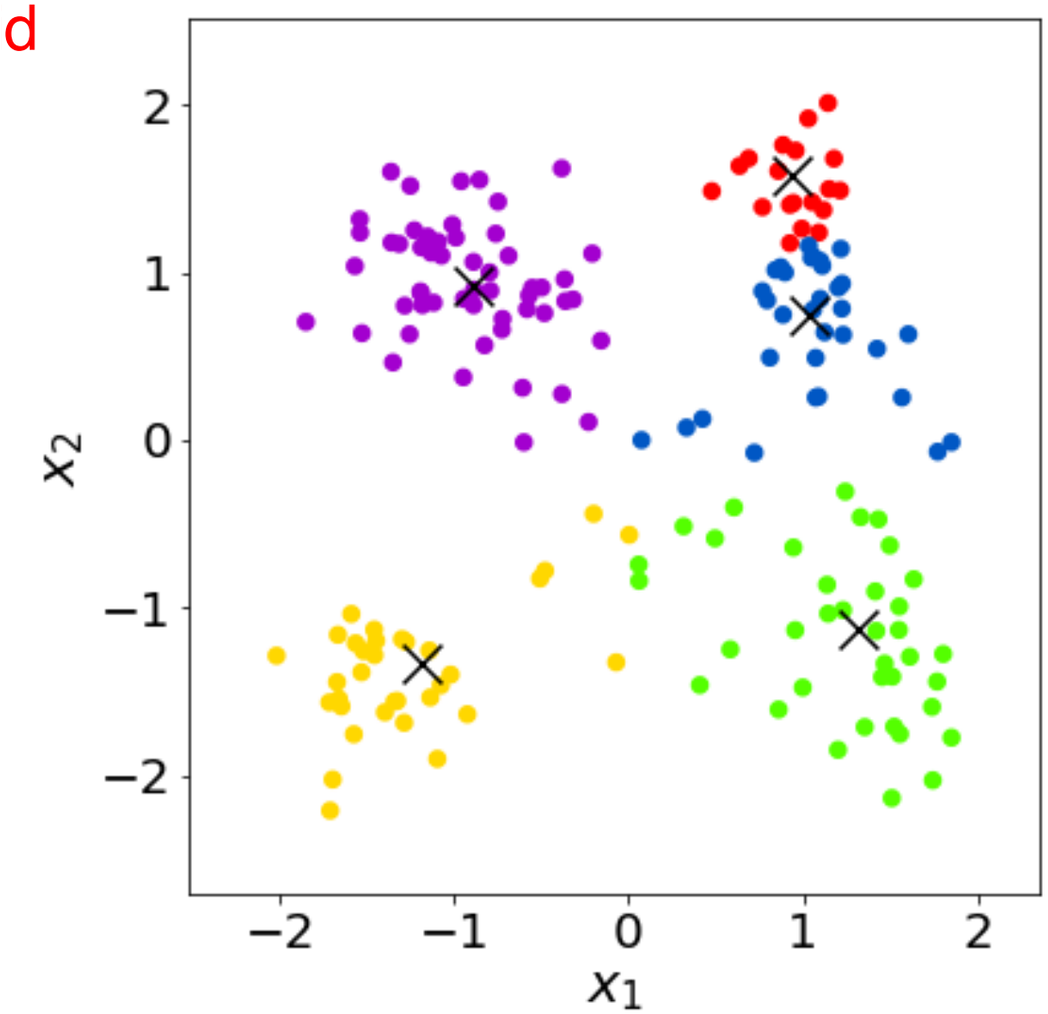
## Clustering

- k-means

i) Assign labels according to closest centroid

$$y_t^{(i)} = \mathrm{argmin}_j \|\mathbf{x}^{(i)} - \boldsymbol{\mu}_t^{(j)}\|_p$$

$$\|\mathbf{x} - \boldsymbol{\mu}\|_p = \left[\sum_j (x_j - \mu_j)^p\right]^{\frac{1}{p}}$$




ii) Update centroid's coords

$$\boldsymbol{\mu}_{t+1}^{(j)} = \frac{1}{n_j} \sum_i \mathbf{x}^{(i)} \delta_{y_t^{(i)},j}$$

# Step 3 : Learning Algorithms (unsupervised)



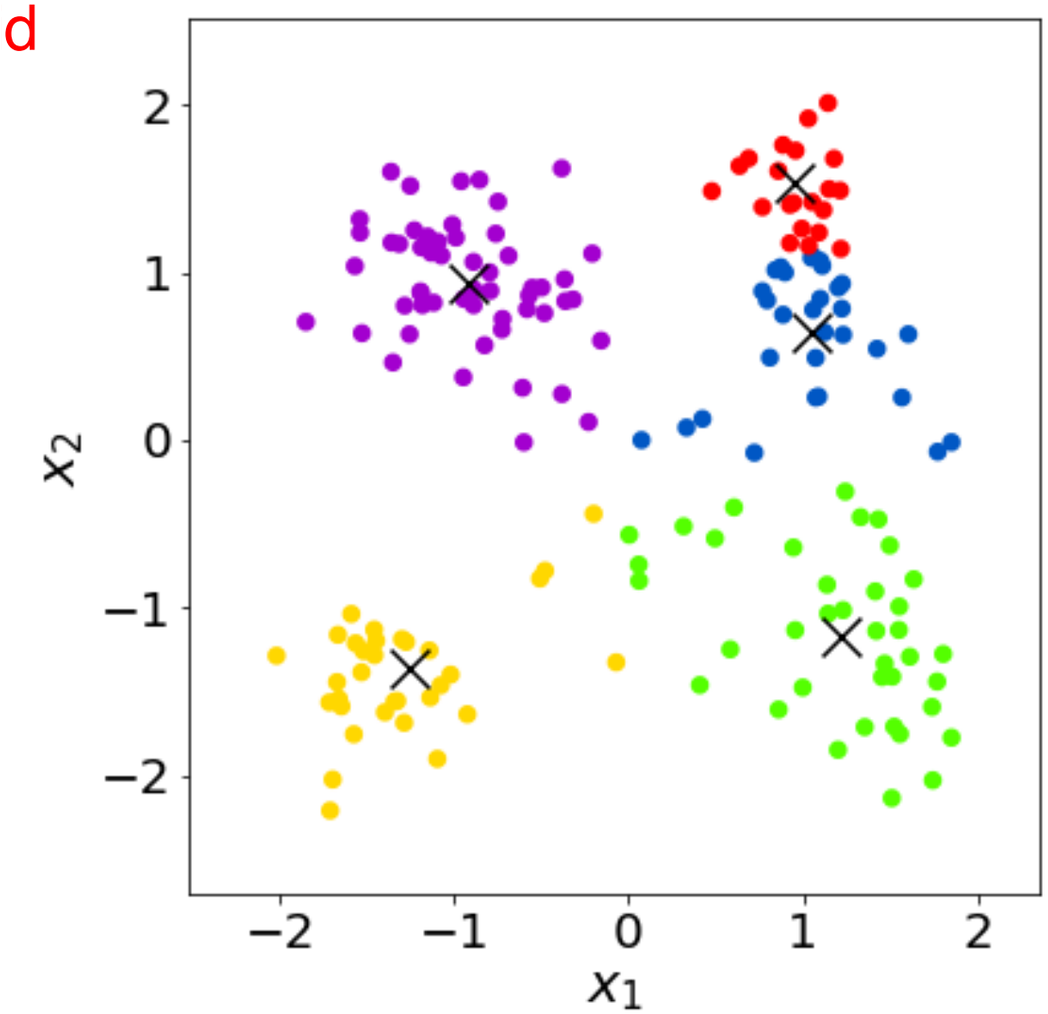
## Clustering

- k-means

i) Assign labels according to closest centroid

$$y_t^{(i)} = \mathrm{argmin}_j \|\mathbf{x}^{(i)} - \boldsymbol{\mu}_t^{(j)}\|_p$$

$$\|\mathbf{x} - \boldsymbol{\mu}\|_p = \left[\sum_j (x_j - \mu_j)^p\right]^{\frac{1}{p}}$$




ii) Update centroid's coords

$$\boldsymbol{\mu}_{t+1}^{(j)} = \frac{1}{n_j} \sum_i \mathbf{x}^{(i)} \delta_{y_t^{(i)},j}$$

# Step 3 : Learning Algorithms (unsupervised)



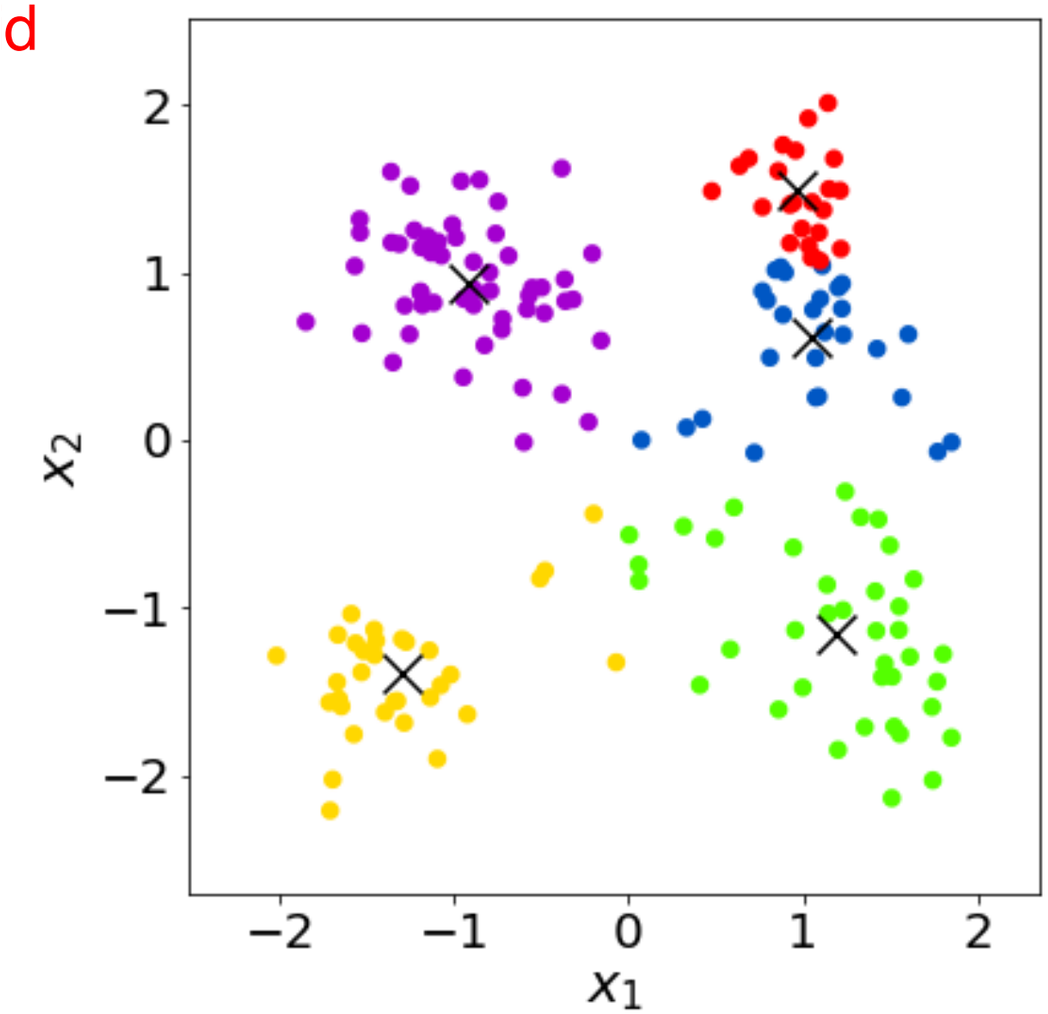
## Clustering

- k-means

i) Assign labels according to closest centroid

$$y_t^{(i)} = \mathrm{argmin}_j \|\mathbf{x}^{(i)} - \boldsymbol{\mu}_t^{(j)}\|_p$$

$$\|\mathbf{x} - \boldsymbol{\mu}\|_p = \left[\sum_j (x_j - \mu_j)^p\right]^{\frac{1}{p}}$$




ii) Update centroid's coords

$$\boldsymbol{\mu}_{t+1}^{(j)} = \frac{1}{n_j} \sum_i \mathbf{x}^{(i)} \delta_{y_t^{(i)},j}$$

# Step 3 : Learning Algorithms (unsupervised)



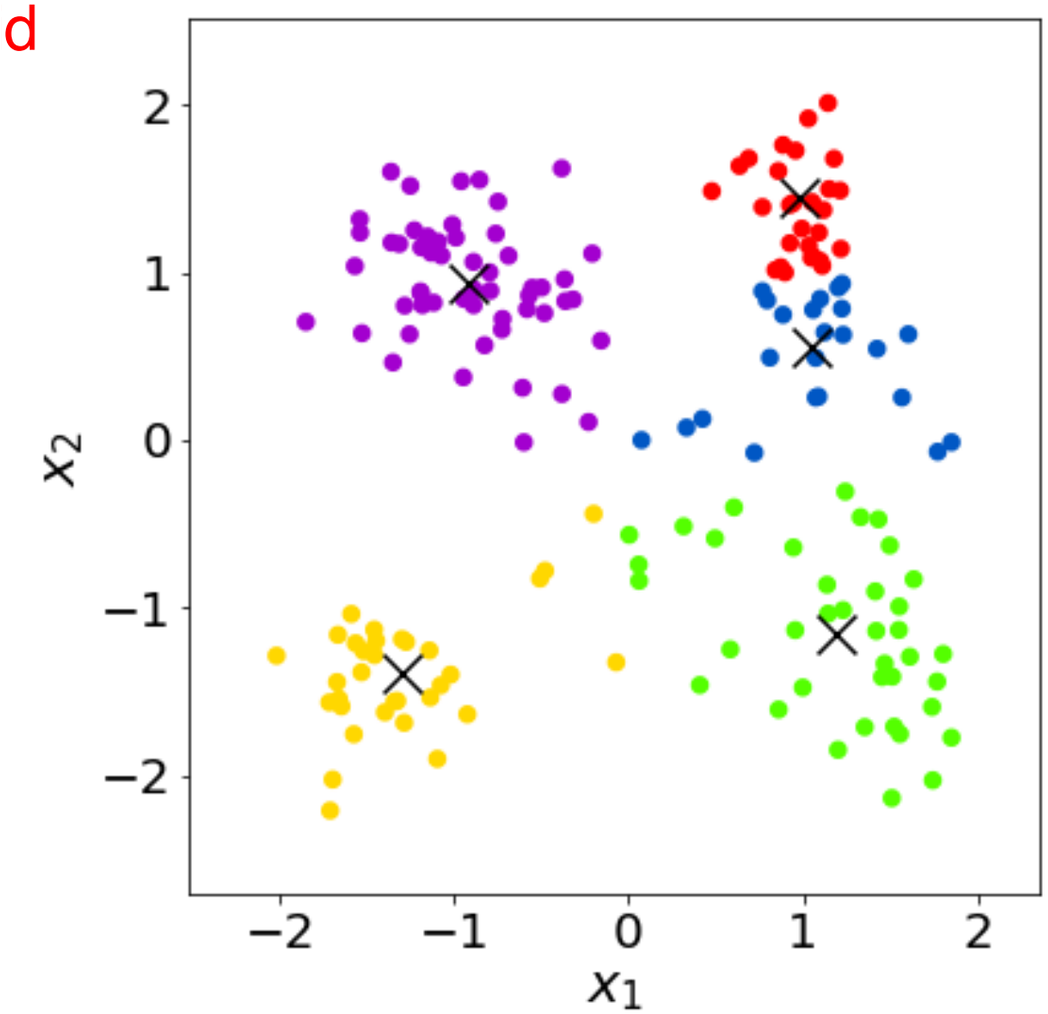
## Clustering

- k-means

i) Assign labels according to closest centroid

$$y_t^{(i)} = \mathrm{argmin}_j \|\mathbf{x}^{(i)} - \boldsymbol{\mu}_t^{(j)}\|_p$$

$$\|\mathbf{x} - \boldsymbol{\mu}\|_p = \left[\sum_j (x_j - \mu_j)^p\right]^{\frac{1}{p}}$$




ii) Update centroid's coords

$$\boldsymbol{\mu}_{t+1}^{(j)} = \frac{1}{n_j} \sum_i \mathbf{x}^{(i)} \delta_{y_t^{(i)},j}$$

# Step 3 : Learning Algorithms (unsupervised)



## Clustering

- hierarchical clustering

# Step 3 : Learning Algorithms (unsupervised)



## Clustering

- hierarchical clustering

  i) Start with one cluster for each data point

  ii) Merge clusters based on similarity

$$d(A, B) = \min_{\mathbf{x} \in A, \mathbf{y} \in B} \|\mathbf{x} - \mathbf{y}\|_p$$

# Step 3 : Learning Algorithms (unsupervised)



## Clustering

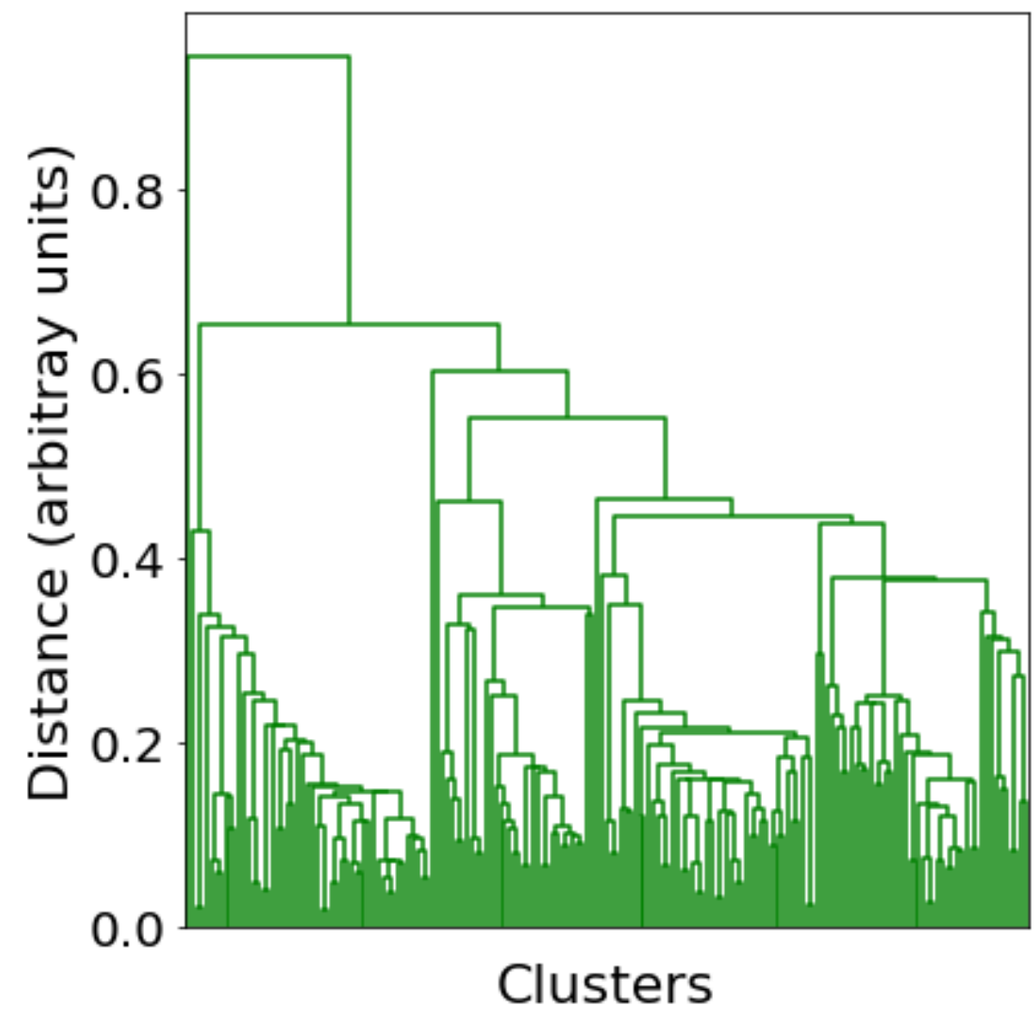
- hierarchical clustering

i) Start with one cluster for each data point

ii) Merge clusters based on similarity

$$d(A, B) = \min_{\mathbf{x} \in A, \mathbf{y} \in B} \|\mathbf{x} - \mathbf{y}\|_p$$

iii) Stop when there's only one big cluster



# Step 3 : Learning Algorithms (unsupervised)



## Clustering

- hierarchical clustering

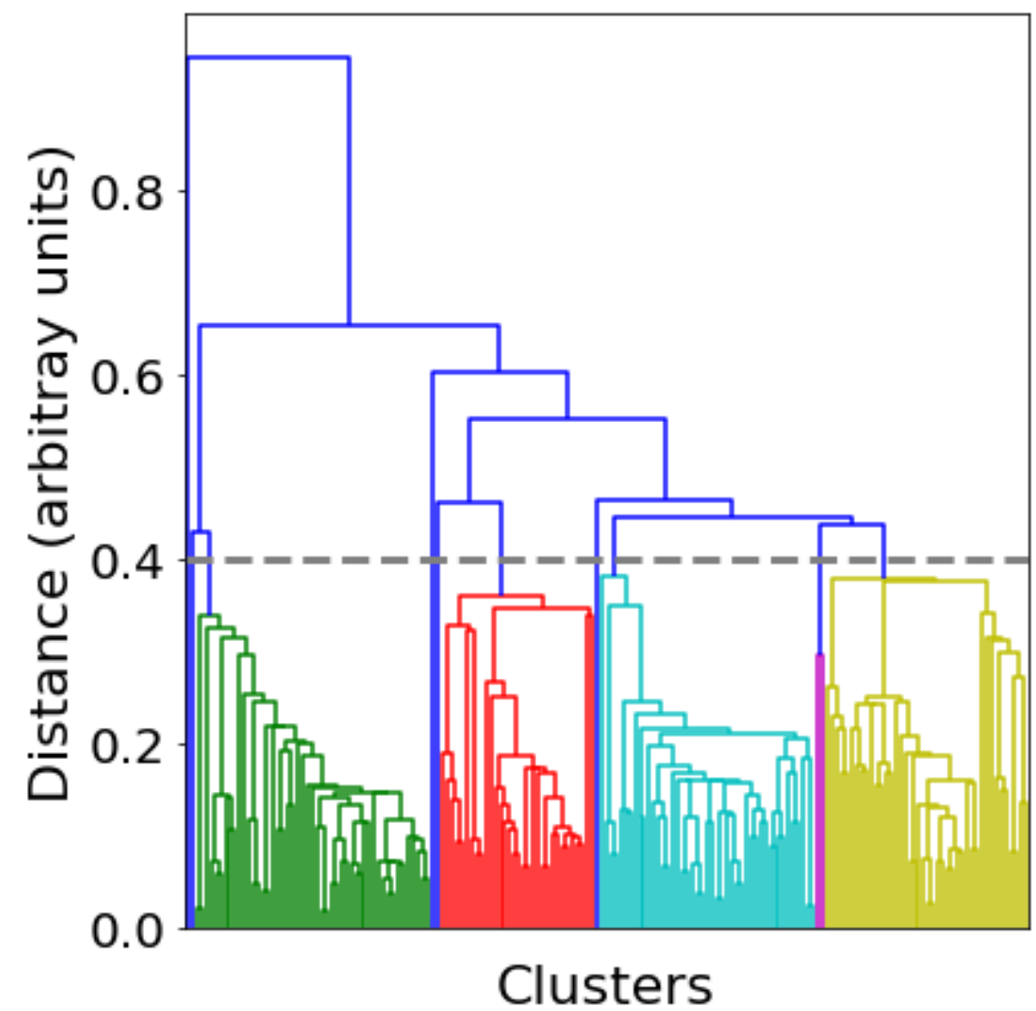
i) Start with one cluster for each data point

ii) Merge clusters based on similarity

$$d(A, B) = \min_{\mathbf{x} \in A, \mathbf{y} \in B} \|\mathbf{x} - \mathbf{y}\|_p$$

iii) Stop when there's only one big cluster

iv) Choose a cutting line (number of clusters)




**d = 0.4**
**N = 10**

# Step 3 : Learning Algorithms (unsupervised)



## Clustering

- hierarchical clustering

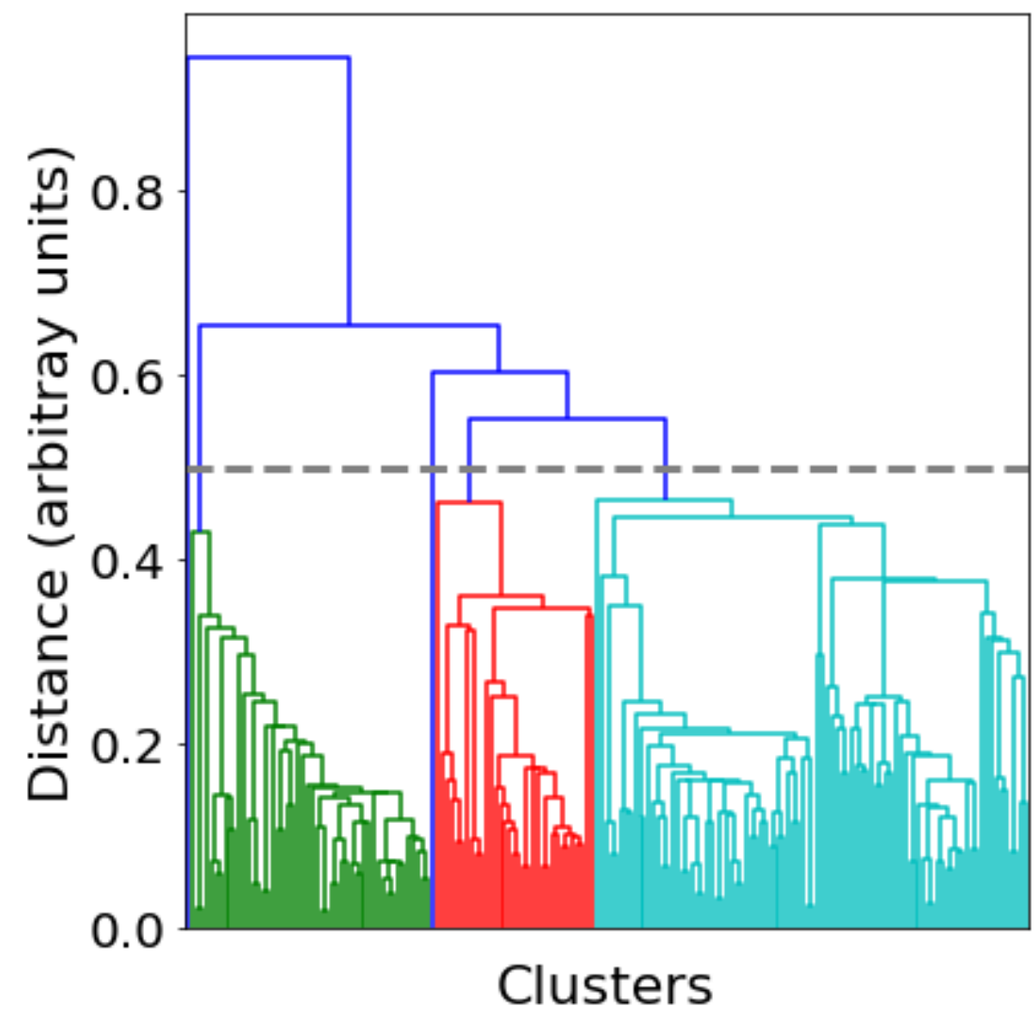
i) Start with one cluster for each data point

ii) Merge clusters based on similarity

$$d(A, B) = \min_{\mathbf{x} \in A, \mathbf{y} \in B} \|\mathbf{x} - \mathbf{y}\|_p$$

iii) Stop when there's only one big cluster

iv) Choose a cutting line (number of clusters)




**d = 0.5**
**N = 5**

# Step 3 : Learning Algorithms (unsupervised)



## Clustering

- hierarchical clustering

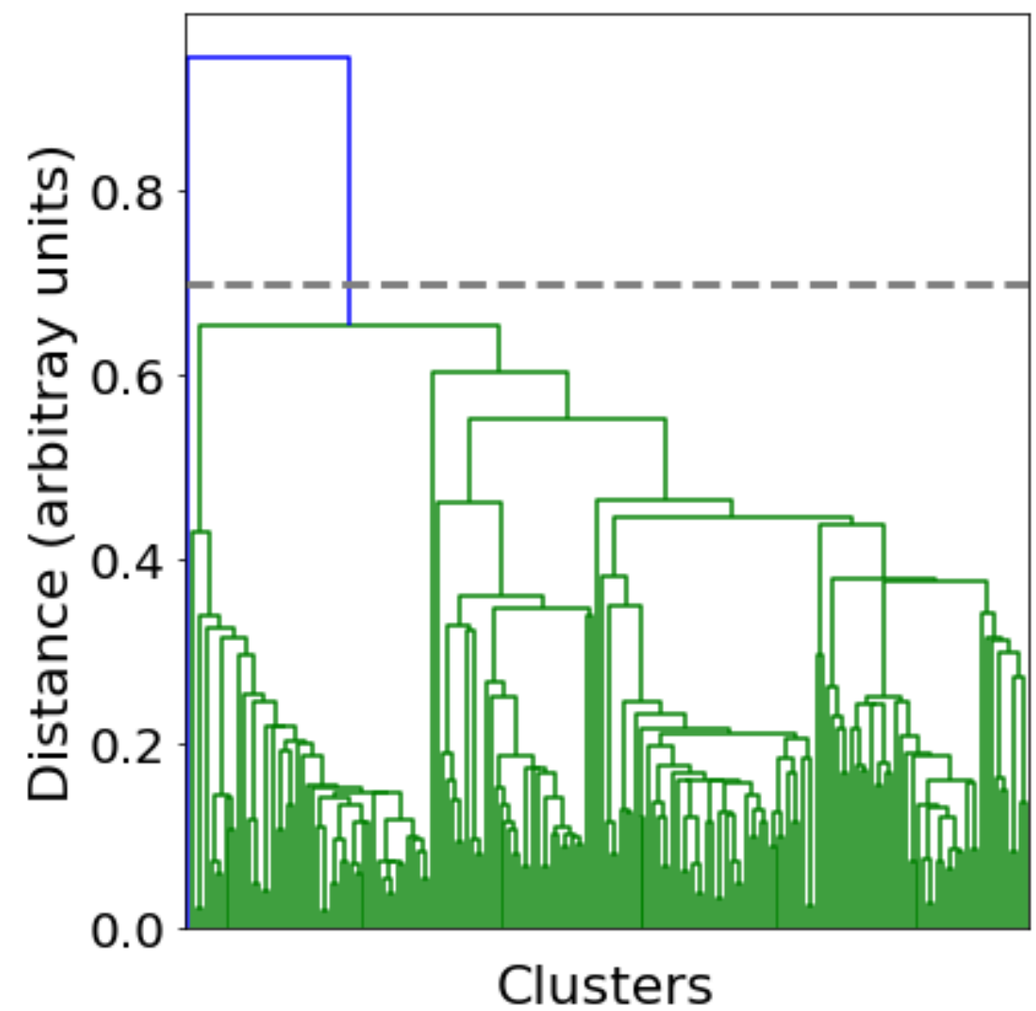
i) Start with one cluster for each data point

ii) Merge clusters based on similarity

$$d(A, B) = \min_{\mathbf{x} \in A, \mathbf{y} \in B} \|\mathbf{x} - \mathbf{y}\|_p$$

iii) Stop when there's only one big cluster

iv) Choose a cutting line (number of clusters)




**d = 0.7**
**N = 2**

# Step 3 : Learning Algorithms (supervised)



## Linear Regression

# Step 3 : Learning Algorithms (supervised)



## Linear Regression

Function fitting: $\hat{y}^{(i)} = \boldsymbol{\theta}^T \mathbf{x}^{(i)} = x_0^{(i)}\theta_0 + x_1^{(i)}\theta_1 + \ldots$

# Step 3 : Learning Algorithms (supervised)



## Linear Regression

Function fitting: $\hat{y}^{(i)} = \boldsymbol{\theta}^T \mathbf{x}^{(i)} = x_0^{(i)}\theta_0 + x_1^{(i)}\theta_1 + \dots$

# Step 3 : Learning Algorithms (supervised)



## Linear Regression

Function fitting: $\hat{y}^{(i)} = \boldsymbol{\theta}^T \mathbf{x}^{(i)} = x_0^{(i)}\theta_0 + x_1^{(i)}\theta_1 + \dots$

Training = Minimize Cost function

$$J(\boldsymbol{\theta}) = \sum_{i=1}^{n} L[\hat{y}^{(i)}(\mathbf{x}^{(i)}, \boldsymbol{\theta}), y^{(i)}] = \frac{1}{2}\sum_{i=1}^{n}(\boldsymbol{\theta}^T \mathbf{x}^{(i)} - y^{(i)})^2$$

# Step 3 : Learning Algorithms (supervised)

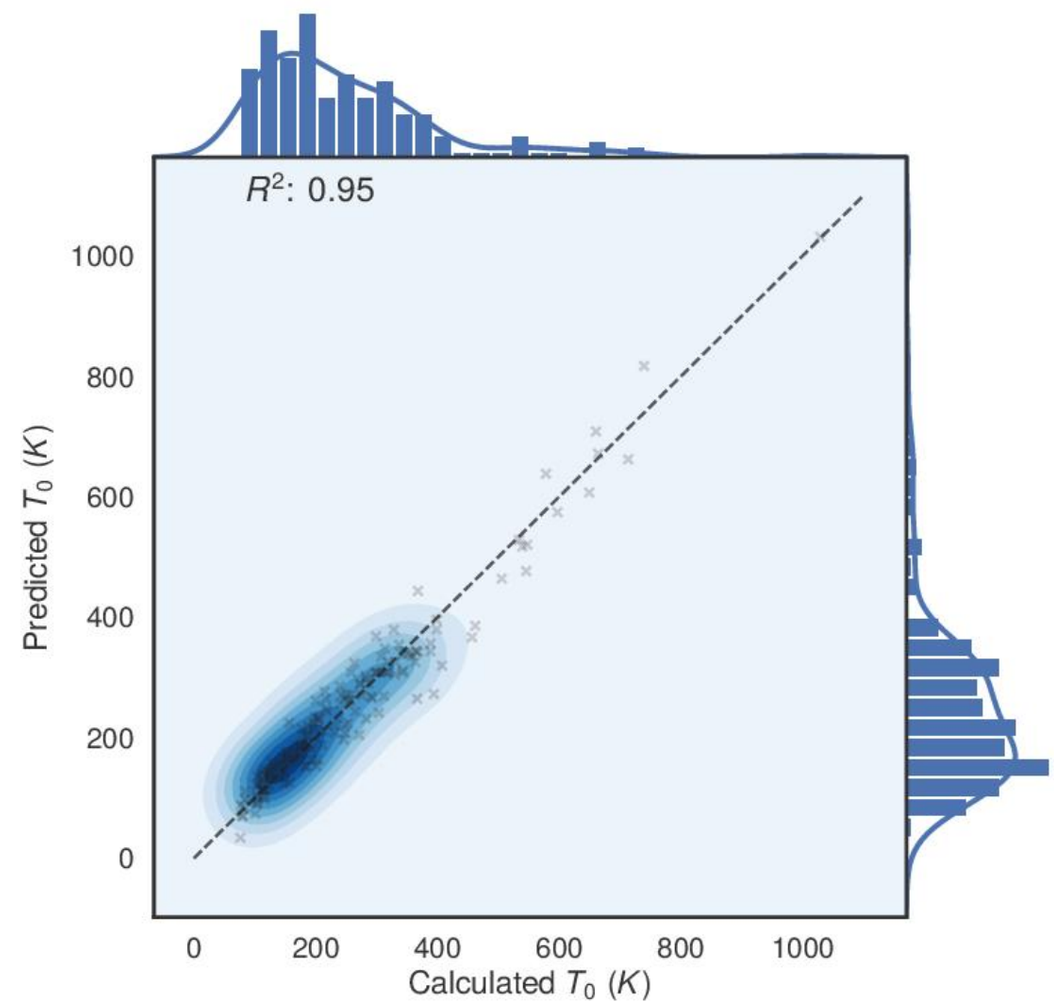
## Linear Regression

Function fitting: $\hat{y}^{(i)} = \boldsymbol{\theta}^T \mathbf{x}^{(i)} = x_0^{(i)}\theta_0 + x_1^{(i)}\theta_1 + \dots$

Training = Minimize Cost function

$$J(\boldsymbol{\theta}) = \sum_{i=1}^{n} L[\hat{y}^{(i)}(\mathbf{x}^{(i)}, \boldsymbol{\theta}), y^{(i)}] = \frac{1}{2}\sum_{i=1}^{n}(\boldsymbol{\theta}^T \mathbf{x}^{(i)} - y^{(i)})^2$$

$$\boldsymbol{\theta} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$$



# Step 3 : Learning Algorithms (supervised)

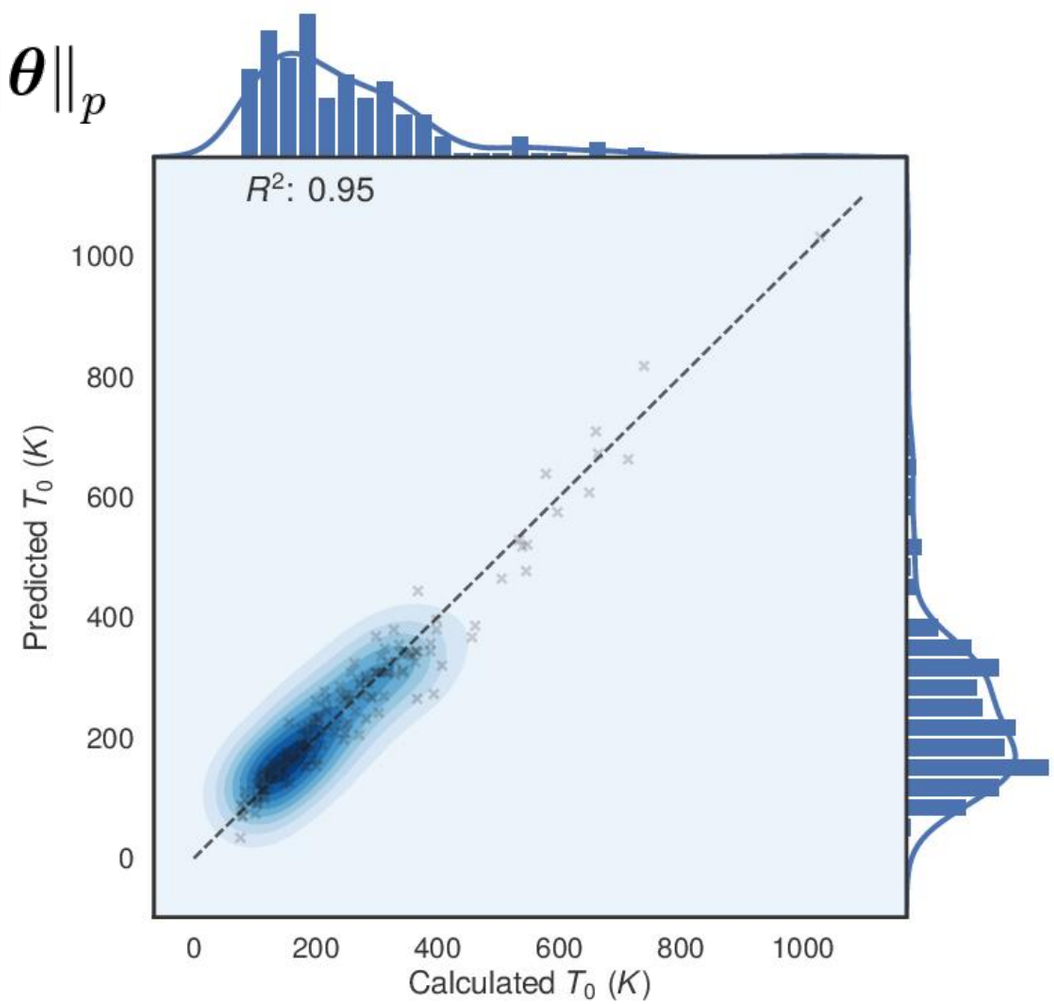
## Linear Regression

Function fitting: $\hat{y}^{(i)} = \boldsymbol{\theta}^T \mathbf{x}^{(i)} = x_0^{(i)}\theta_0 + x_1^{(i)}\theta_1 + \ldots$

Training = Minimize Cost function

$$J(\boldsymbol{\theta}) = \sum_{i=1}^{n} L[\hat{y}^{(i)}(\mathbf{x}^{(i)}, \boldsymbol{\theta}), y^{(i)}] = \frac{1}{2}\sum_{i=1}^{n}(\boldsymbol{\theta}^T \mathbf{x}^{(i)} - y^{(i)})^2 + \lambda \left\|\boldsymbol{\theta}\right\|_p$$

$$\boldsymbol{\theta} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$$



# Step 3 : Learning Algorithms (supervised)

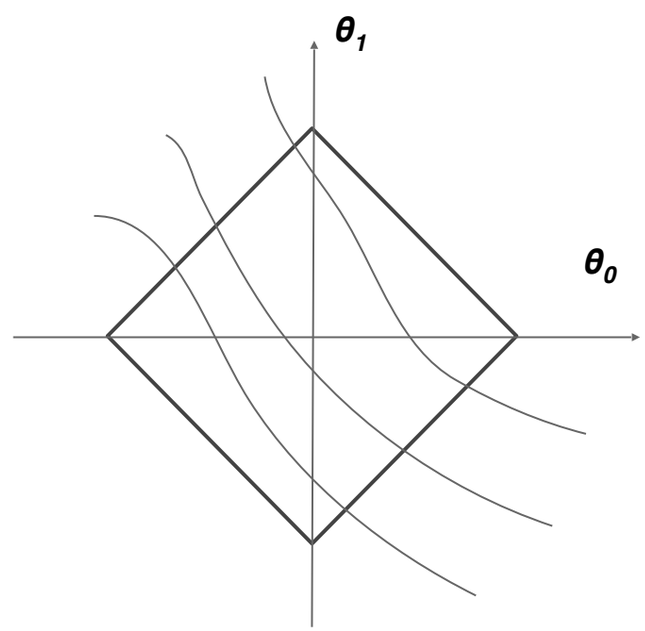
## Linear Regression

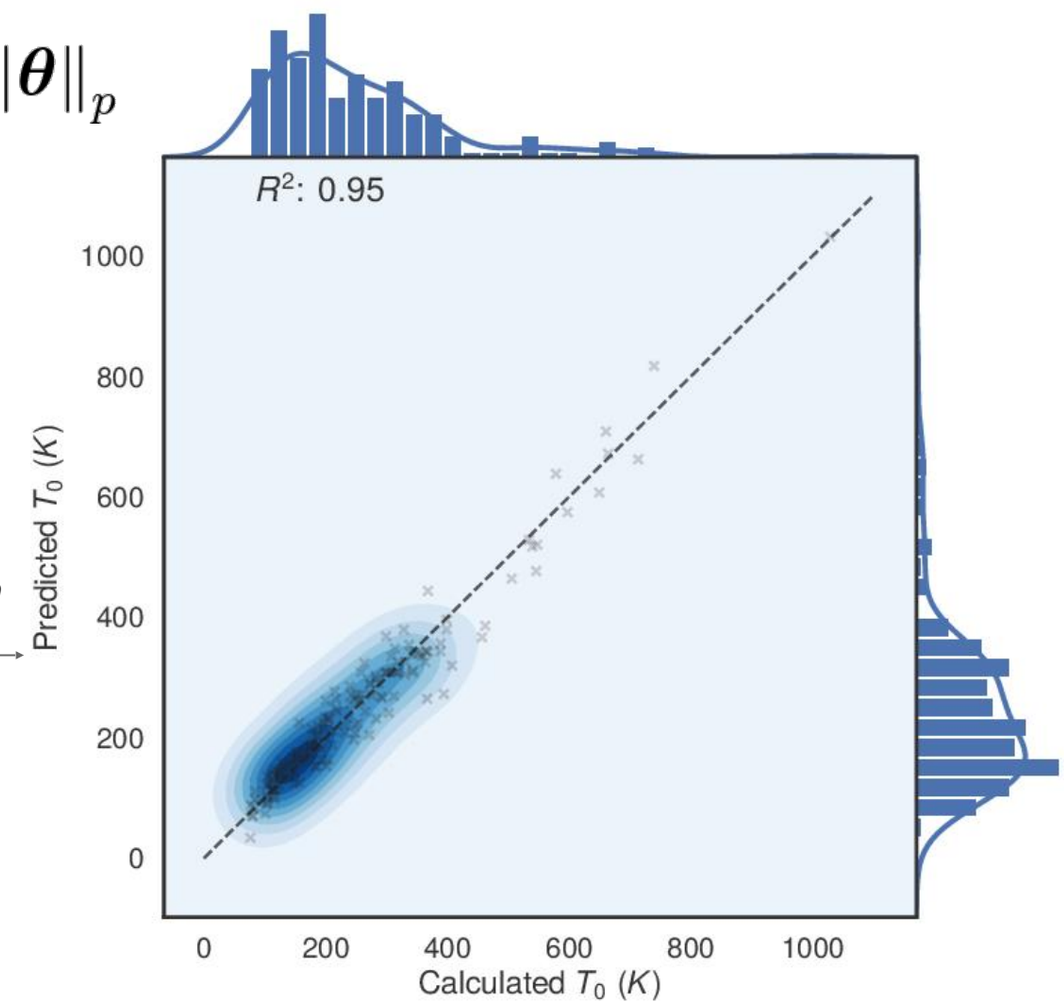
Function fitting: $\hat{y}^{(i)} = \boldsymbol{\theta}^T \mathbf{x}^{(i)} = x_0^{(i)}\theta_0 + x_1^{(i)}\theta_1 + \ldots$

Training = Minimize Cost function

$$J(\boldsymbol{\theta}) = \sum_{i=1}^{n} L[\hat{y}^{(i)}(\mathbf{x}^{(i)}, \boldsymbol{\theta}), y^{(i)}] = \frac{1}{2}\sum_{i=1}^{n}(\boldsymbol{\theta}^T\mathbf{x}^{(i)} - y^{(i)})^2 + \lambda \left\|\boldsymbol{\theta}\right\|_p$$

$$\boldsymbol{\theta} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$$



# Step 3 : Learning Algorithms (supervised)

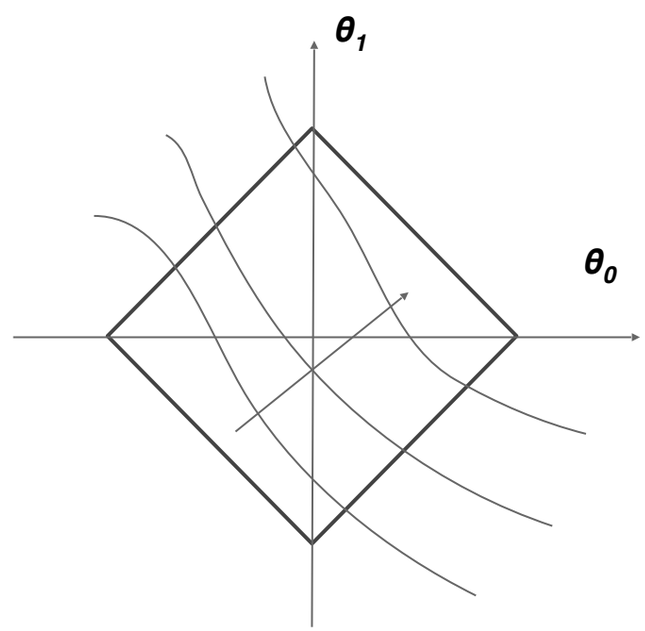
## Linear Regression

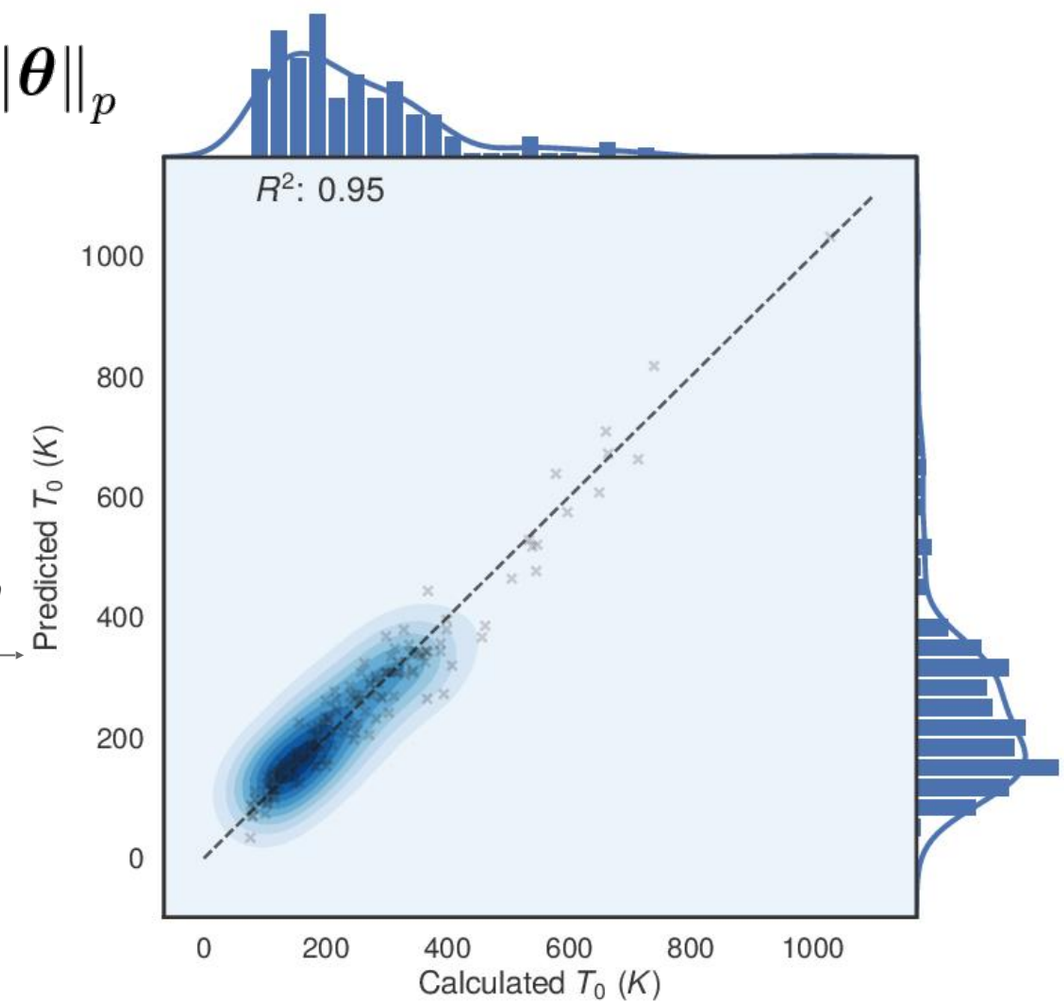
Function fitting: $\hat{y}^{(i)} = \boldsymbol{\theta}^T \mathbf{x}^{(i)} = x_0^{(i)}\theta_0 + x_1^{(i)}\theta_1 + \ldots$

Training = Minimize Cost function

$$J(\boldsymbol{\theta}) = \sum_{i=1}^{n} L[\hat{y}^{(i)}(\mathbf{x}^{(i)}, \boldsymbol{\theta}), y^{(i)}] = \frac{1}{2}\sum_{i=1}^{n}(\boldsymbol{\theta}^T\mathbf{x}^{(i)} - y^{(i)})^2 + \lambda \left\|\boldsymbol{\theta}\right\|_p$$

$$\boldsymbol{\theta} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$$



# Step 3 : Learning Algorithms (supervised)

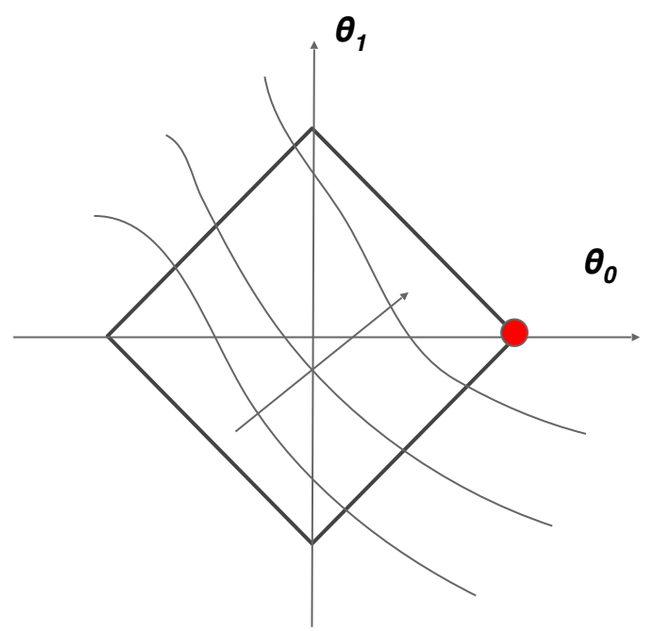
## Linear Regression

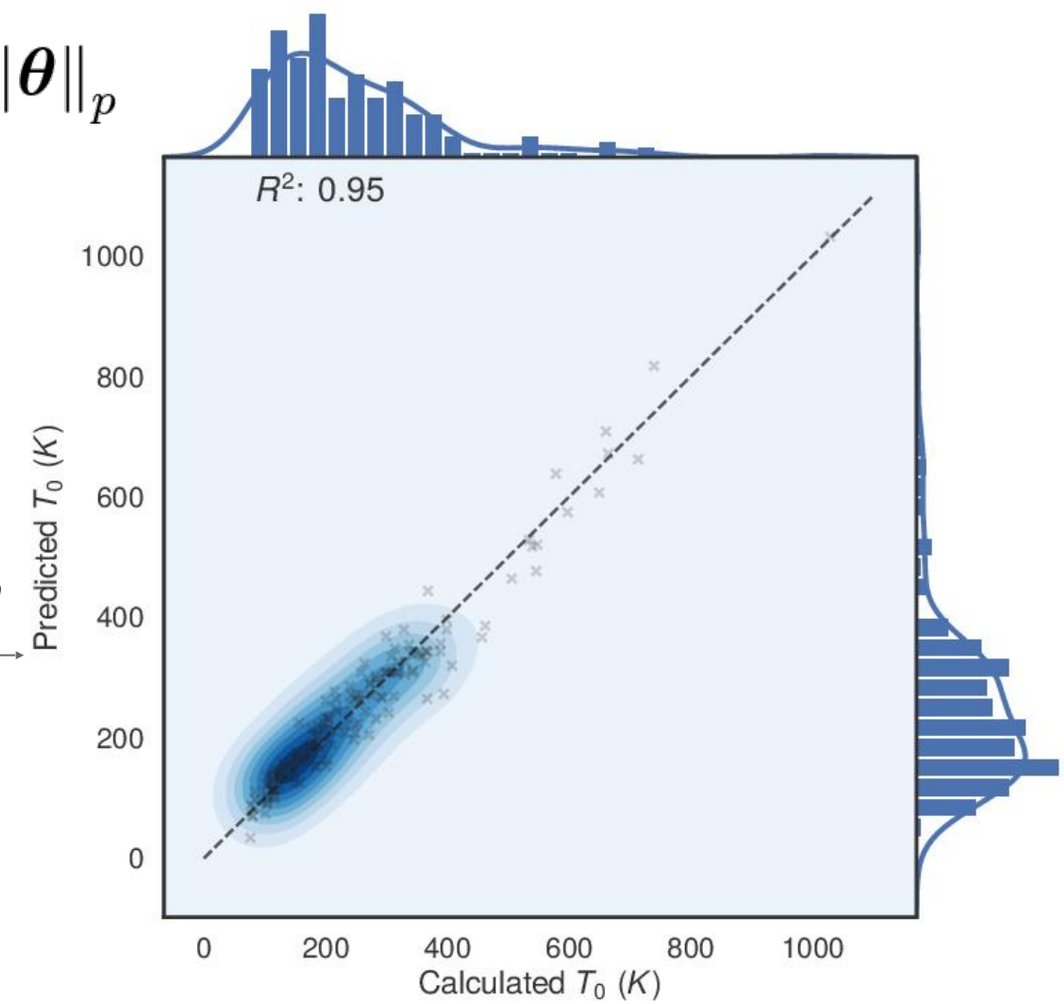
Function fitting: $\hat{y}^{(i)} = \boldsymbol{\theta}^T \mathbf{x}^{(i)} = x_0^{(i)}\theta_0 + x_1^{(i)}\theta_1 + \ldots$

Training = Minimize Cost function

$$J(\boldsymbol{\theta}) = \sum_{i=1}^{n} L[\hat{y}^{(i)}(\mathbf{x}^{(i)}, \boldsymbol{\theta}), y^{(i)}] = \frac{1}{2}\sum_{i=1}^{n}(\boldsymbol{\theta}^T\mathbf{x}^{(i)} - y^{(i)})^2 + \lambda \left\|\boldsymbol{\theta}\right\|_p$$

$$\boldsymbol{\theta} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$$



# Step 3 : Learning Algorithms (supervised)

## Linear Regression

Function fitting: $\hat{y}^{(i)} = \boldsymbol{\theta}^T \mathbf{x}^{(i)} = x_0^{(i)}\theta_0 + x_1^{(i)}\theta_1 + \dots$

Training = Minimize Cost function

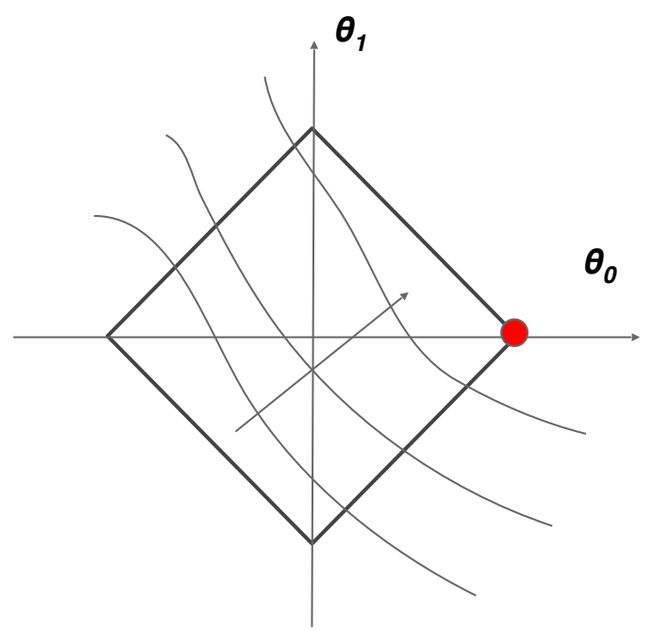
$$J(\boldsymbol{\theta}) = \sum_{i=1}^{n} L[\hat{y}^{(i)}(\mathbf{x}^{(i)}, \boldsymbol{\theta}), y^{(i)}] = \frac{1}{2}\sum_{i=1}^{n}(\boldsymbol{\theta}^T\mathbf{x}^{(i)} - y^{(i)})^2 + \lambda \|\boldsymbol{\theta}\|_p$$

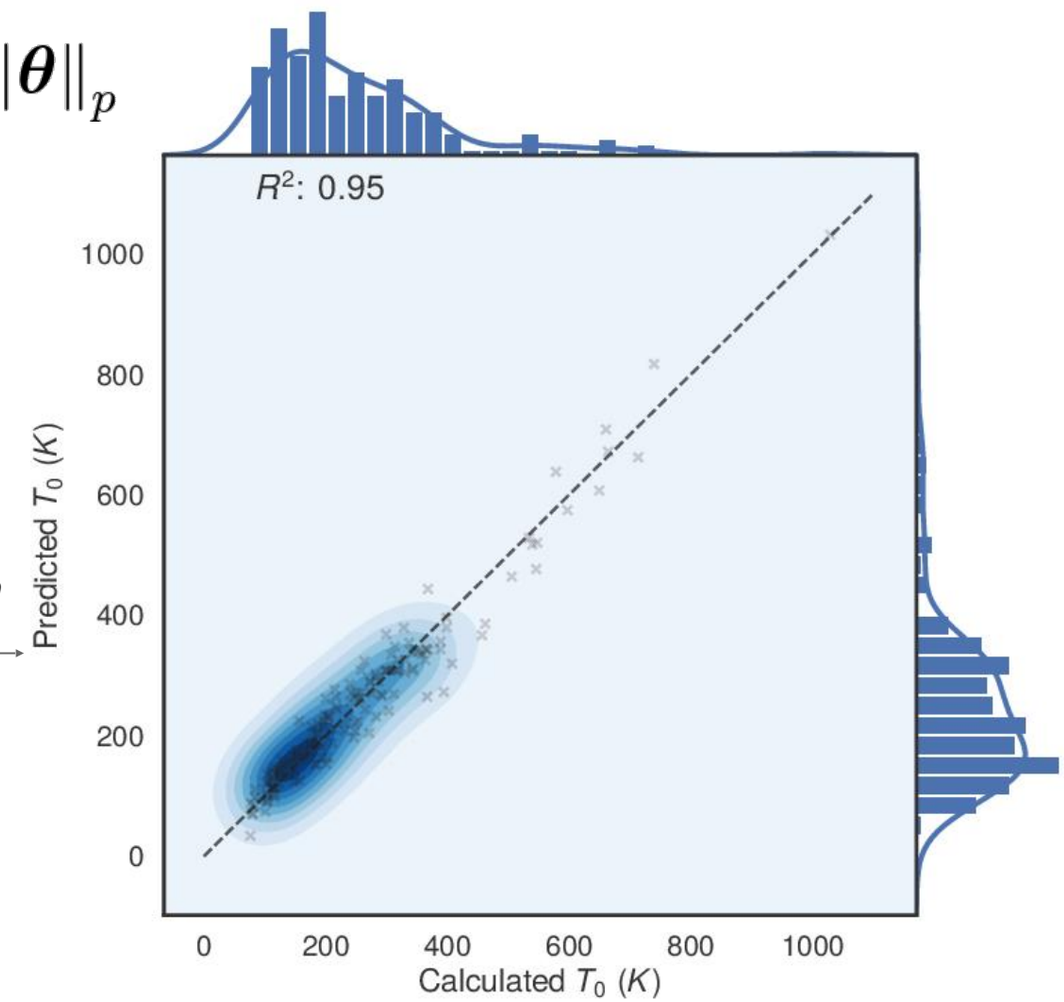
$$\boldsymbol{\theta} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$$

$$\text{MAE} = \frac{1}{n}\sum_{i}^{n}|y_i - \hat{y}_i|$$

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i}(y_i - \hat{y}_i)^2}$$

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}} = 1 - \frac{\sum_i (y_i - \hat{y})^2}{\sum_i (y_i - \bar{y})^2}$$



# Step 3 : Learning Algorithms (supervised)



## Logistic Regression

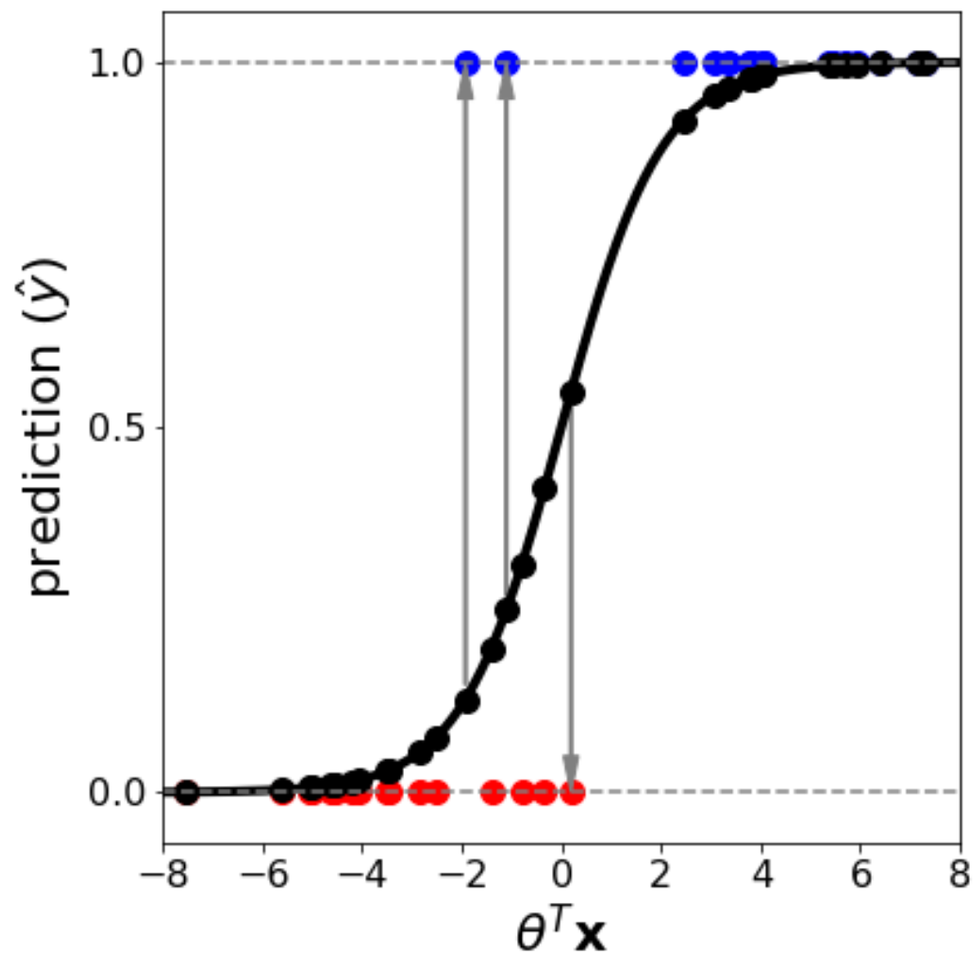
# Step 3 : Learning Algorithms (supervised)



## Logistic Regression

- Predictions: sigmoid

$$\hat{y} = \sigma(\boldsymbol{\theta}^T \mathbf{x}) = \frac{1}{1 + e^{-\boldsymbol{\theta}^T \mathbf{x}}}$$



# Step 3 : Learning Algorithms (supervised)



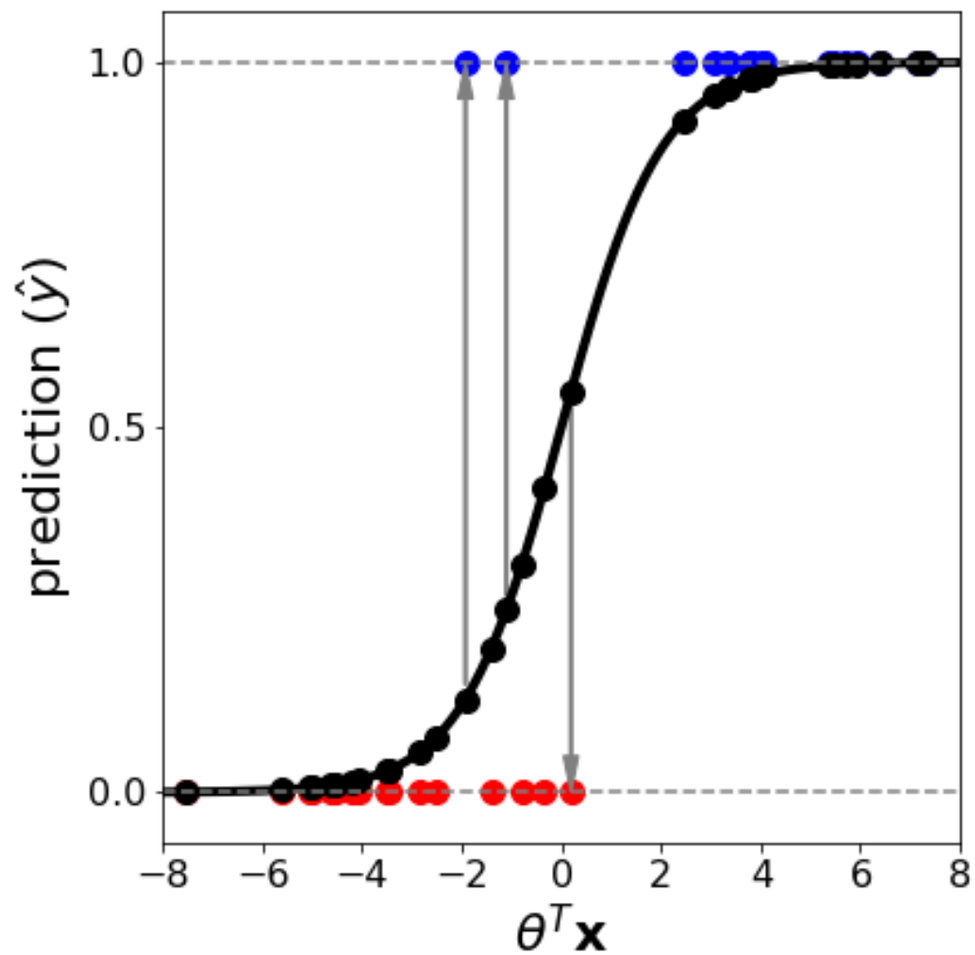
## Logistic Regression

- Predictions: sigmoid

$$\hat{y} = \sigma(\boldsymbol{\theta}^T \mathbf{x}) = \frac{1}{1 + e^{-\boldsymbol{\theta}^T \mathbf{x}}}$$

- Training : minimize cost function

$$J(\boldsymbol{\theta}) = -\frac{1}{n} \sum_{i=1}^{n} \left[ y^{(i)} \log(\hat{y}^{(i)}) + (1 - y^{(i)}) \log(1 - \hat{y}^{(i)}) \right]$$



# Step 3 : Learning Algorithms (supervised)



## Logistic Regression

- Predictions: sigmoid

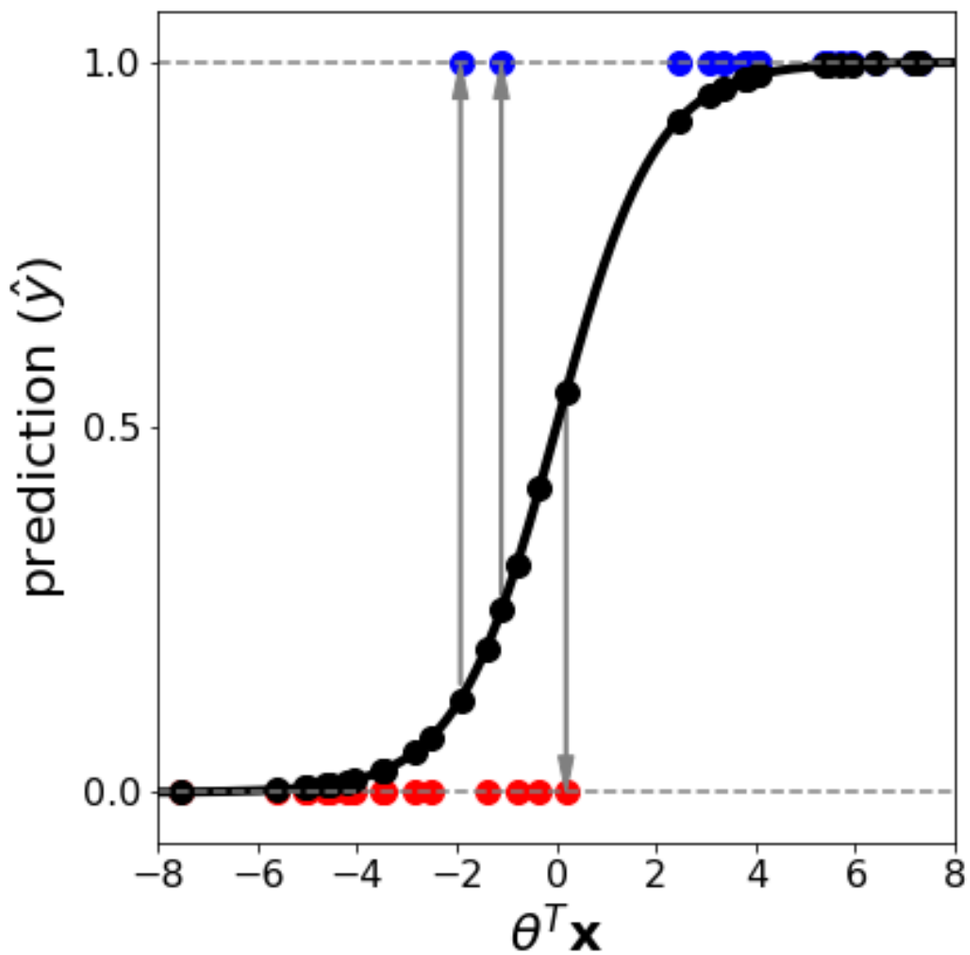
$$\hat{y} = \sigma(\boldsymbol{\theta}^T \mathbf{x}) = \frac{1}{1 + e^{-\boldsymbol{\theta}^T \mathbf{x}}}$$

- Training : minimize cost function

$$J(\boldsymbol{\theta}) = -\frac{1}{n} \sum_{i=1}^{n} \left[ y^{(i)} \log(\hat{y}^{(i)}) + (1 - y^{(i)}) \log(1 - \hat{y}^{(i)}) \right]$$

$$TPR = \frac{TP}{TP + FN}$$

$$FPR = \frac{FP}{FP + TN}$$



# Step 3 : Learning Algorithms (supervised)



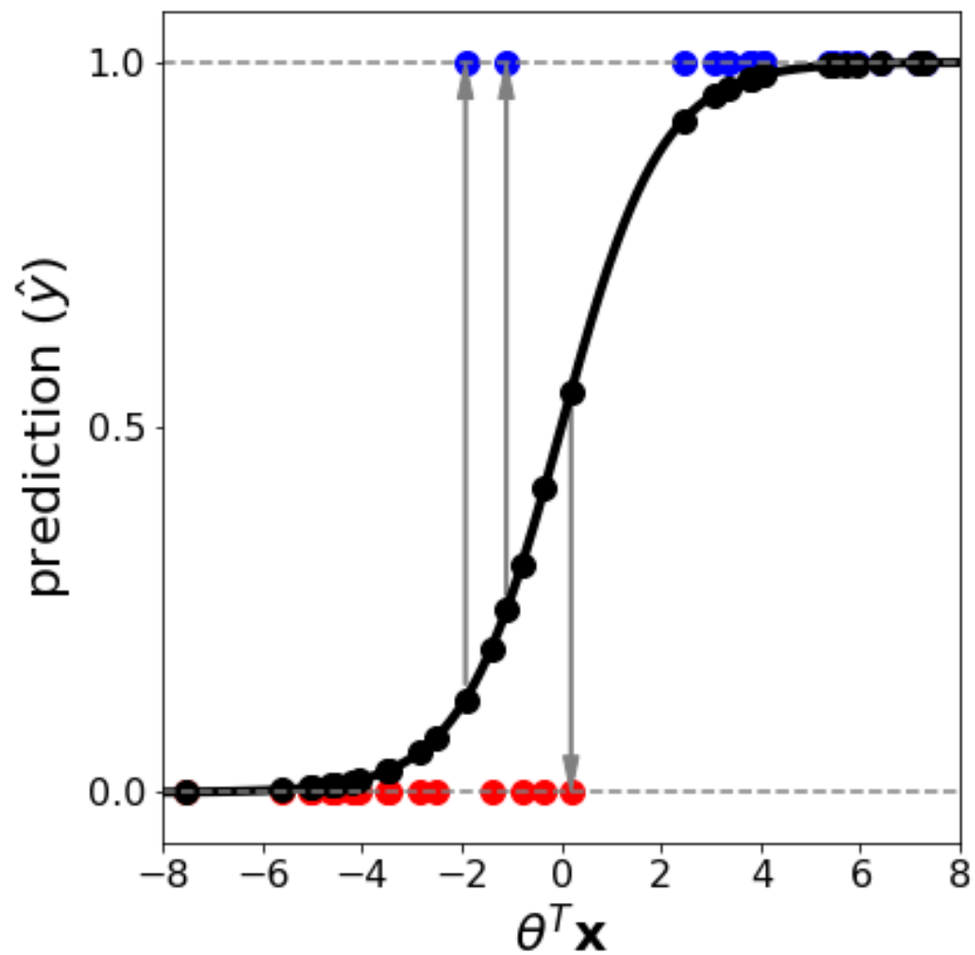
## Logistic Regression

- Predictions: sigmoid

$$\hat{y} = \sigma(\boldsymbol{\theta}^T \mathbf{x}) = \frac{1}{1 + e^{-\boldsymbol{\theta}^T \mathbf{x}}}$$

- Training : minimize cost function

$$J(\boldsymbol{\theta}) = -\frac{1}{n} \sum_{i=1}^{n} \left[ y^{(i)} \log(\hat{y}^{(i)}) + (1 - y^{(i)}) \log(1 - \hat{y}^{(i)}) \right]$$




$$TPR = \frac{TP}{TP + FN}$$

$$FPR = \frac{FP}{FP + TN}$$

| | |
|---|---|
| TP | FP |
| FN | TN |

# Step 3 : Learning Algorithms (supervised)



## Naïve Bayes

$$P(y|\boldsymbol{x}) = \frac{P(\boldsymbol{x}|y)P(y)}{P(\boldsymbol{x})}$$

# Step 3 : Learning Algorithms (supervised)



## Naïve Bayes

Naïve = conditionally independent features

$$P(y|\boldsymbol{x}) = \frac{P(\boldsymbol{x}|y)P(y)}{P(\boldsymbol{x})} = \frac{\prod_j P(x_j|y)p(y)}{\prod_j P(x_j)}$$

# Step 3 : Learning Algorithms (supervised)



## Naïve Bayes

Naïve = conditionally independent features

$$P(y|\boldsymbol{x}) = \frac{P(\boldsymbol{x}|y)P(y)}{P(\boldsymbol{x})} = \frac{\prod_j P(x_j|y)p(y)}{\prod_j P(x_j)}$$

$$P(y = \text{banana}|\text{long, sweet})$$

# Step 3 : Learning Algorithms (supervised)

## Naïve Bayes

Naïve = conditionally independent features

$$P(y|\boldsymbol{x}) = \frac{P(\boldsymbol{x}|y)P(y)}{P(\boldsymbol{x})} = \frac{\prod_j P(x_j|y)p(y)}{\prod_j P(x_j)}$$

$$P(y = \text{banana}|\text{long, sweet})$$

| Fruit | Long ($x_1$) | Sweet ($x_2$) | Yellow ($x_3$) |
|---|---|---|---|
| Orange | 0 | 1 | 0 |
| Banana | 1 | 0 | 1 |
| Banana | 1 | 1 | 1 |
| Mango | 1 | 1 | 0 |

# Step 3 : Learning Algorithms (supervised)

## Naïve Bayes

Naïve = conditionally independent features

$$P(y|\boldsymbol{x}) = \frac{P(\boldsymbol{x}|y)P(y)}{P(\boldsymbol{x})} = \frac{\prod_j P(x_j|y)p(y)}{\prod_j P(x_j)}$$

$$P(y = \text{banana}|\text{long, sweet})$$

| Label | Long | ! Long | Sweet | ! Sweet | Yellow | ! Yellow | Total |
|---|---|---|---|---|---|---|---|
| B | 400 | 100 | 350 | 150 | 450 | 50 | 500 |
| O | 0 | 300 | 150 | 150 | 300 | 0 | 300 |
| M | 100 | 100 | 150 | 50 | 50 | 150 | 200 |
| Total | 500 | 500 | 650 | 350 | 800 | 200 | 1000 |

# Step 3 : Learning Algorithms (supervised)

## Naïve Bayes

Naïve = conditionally independent features

$$P(y|\boldsymbol{x}) = \frac{P(\boldsymbol{x}|y)P(y)}{P(\boldsymbol{x})} = \frac{\prod_j P(x_j|y)p(y)}{\prod_j P(x_j)}$$

$$P(y = \text{banana}|\text{long, sweet}) = P(y = B|x_1 = \text{long}, x_2 = \text{sweet})$$

| Label | Long | ! Long | Sweet | ! Sweet | Yellow | ! Yellow | Total |
|---|---|---|---|---|---|---|---|
| B | 400 | 100 | 350 | 150 | 450 | 50 | 500 |
| O | 0 | 300 | 150 | 150 | 300 | 0 | 300 |
| M | 100 | 100 | 150 | 50 | 50 | 150 | 200 |
| Total | 500 | 500 | 650 | 350 | 800 | 200 | 1000 |

# Step 3 : Learning Algorithms (supervised)

## Naïve Bayes

Naïve = conditionally independent features

$$P(y|\boldsymbol{x}) = \frac{P(\boldsymbol{x}|y)P(y)}{P(\boldsymbol{x})} = \frac{\prod_j P(x_j|y)\boxed{p(y)}}{\prod_j P(x_j)}$$

$$P(y = \text{banana}|\text{long, sweet}) = P(y = B|x_1 = \text{long}, x_2 = \text{sweet})$$

priors

$$P(Y = B) = 500/1000 = 0.50$$
$$P(Y = O) = 300/1000 = 0.30$$
$$P(Y = M) = 200/1000 = 0.20$$

| Label | Long | ! Long | Sweet | ! Sweet | Yellow | ! Yellow | Total |
|---|---|---|---|---|---|---|---|
| B | 400 | 100 | 350 | 150 | 450 | 50 | 500 |
| O | 0 | 300 | 150 | 150 | 300 | 0 | 300 |
| M | 100 | 100 | 150 | 50 | 50 | 150 | 200 |
| Total | 500 | 500 | 650 | 350 | 800 | 200 | 1000 |

# Step 3 : Learning Algorithms (supervised)

## Naïve Bayes

Naïve = conditionally independent features

$$P(y|\boldsymbol{x}) = \frac{P(\boldsymbol{x}|y)P(y)}{P(\boldsymbol{x})} = \frac{\prod_j P(x_j|y)p(y)}{\prod_j P(x_j)}$$

$$P(y = \text{banana}|\text{long, sweet}) = P(y = B|x_1 = \text{long}, x_2 = \text{sweet})$$

priors

$$P(Y = B) = 500/1000 = 0.50$$
$$P(Y = O) = 300/1000 = 0.30$$
$$P(Y = M) = 200/1000 = 0.20$$

| Label | Long | ! Long | Sweet | ! Sweet | Yellow | ! Yellow | Total |
|---|---|---|---|---|---|---|---|
| B | 400 | 100 | 350 | 150 | 450 | 50 | 500 |
| O | 0 | 300 | 150 | 150 | 300 | 0 | 300 |
| M | 100 | 100 | 150 | 50 | 50 | 150 | 200 |
| Total | 500 | 500 | 650 | 350 | 800 | 200 | 1000 |

likelihoods (Banana)

$$P(x_1 = \text{long}|Y = B) = 400/500 = 0.80$$
$$P(x_2 = \text{sweet}|Y = B) = 350/500 = 0.70$$
$$P(x_3 = \text{yellow}|Y = B) = 450/500 = 0.90$$

# Step 3 : Learning Algorithms (supervised)

## Naïve Bayes

Naïve = conditionally independent features

$$P(y|\boldsymbol{x}) = \frac{P(\boldsymbol{x}|y)P(y)}{P(\boldsymbol{x})} = \frac{\prod_j P(x_j|y)\,p(y)}{\prod_j P(x_j)}$$

$$P(y = \text{banana}|\text{long, sweet}) = P(y = B|x_1 = \text{long}, x_2 = \text{sweet})$$

priors

$$P(Y = B) = 500/1000 = 0.50$$
$$P(Y = O) = 300/1000 = 0.30$$
$$P(Y = M) = 200/1000 = 0.20$$

| Label | Long | ! Long | Sweet | ! Sweet | Yellow | ! Yellow | Total |
|---|---|---|---|---|---|---|---|
| B | 400 | 100 | 350 | 150 | 450 | 50 | 500 |
| O | 0 | 300 | 150 | 150 | 300 | 0 | 300 |
| M | 100 | 100 | 150 | 50 | 50 | 150 | 200 |
| Total | 500 | 500 | 650 | 350 | 800 | 200 | 1000 |

likelihoods (Banana)

$$P(x_1 = \text{long}|Y = B) = 400/500 = 0.80$$
$$P(x_2 = \text{sweet}|Y = B) = 350/500 = 0.70$$
$$P(x_3 = \text{yellow}|Y = B) = 450/500 = 0.90$$

evidence

$$P(x_1 = \text{long}) = 500/1000 = 0.50$$
$$P(x_2 = \text{sweet}) = 650/1000 = 0.65$$
$$P(x_3 = \text{yellow}) = 800/1000 = 0.80$$

# Step 3 : Learning Algorithms (supervised)

## Naïve Bayes

Naïve = conditionally independent features

$$P(y|\boldsymbol{x}) = \frac{P(\boldsymbol{x}|y)P(y)}{P(\boldsymbol{x})} = \frac{\prod_j P(x_j|y)\,p(y)}{\prod_j P(x_j)}$$

$$P(y = \text{banana}|\text{long, sweet}) = P(y = B|x_1 = \text{long}, x_2 = \text{sweet})$$

$$= \frac{0.8 \times 0.7 \times 0.1}{\quad}$$

priors

$$P(Y = B) = 500/1000 = 0.50$$
$$P(Y = O) = 300/1000 = 0.30$$
$$P(Y = M) = 200/1000 = 0.20$$

| Label | Long | ! Long | Sweet | ! Sweet | Yellow | ! Yellow | Total |
|---|---|---|---|---|---|---|---|
| B | 400 | 100 | 350 | 150 | 450 | 50 | 500 |
| O | 0 | 300 | 150 | 150 | 300 | 0 | 300 |
| M | 100 | 100 | 150 | 50 | 50 | 150 | 200 |
| Total | 500 | 500 | 650 | 350 | 800 | 200 | 1000 |

likelihoods (Banana)

$$P(x_1 = \text{long}|Y = B) = 400/500 = 0.80$$
$$P(x_2 = \text{sweet}|Y = B) = 350/500 = 0.70$$
$$P(x_3 = \text{yellow}|Y = B) = 450/500 = 0.90$$

evidence

$$P(x_1 = \text{long}) = 500/1000 = 0.50$$
$$P(x_2 = \text{sweet}) = 650/1000 = 0.65$$
$$P(x_3 = \text{yellow}) = 800/1000 = 0.80$$

# Step 3 : Learning Algorithms (supervised)

## Naïve Bayes

Naïve = conditionally independent features

$$P(y|\boldsymbol{x}) = \frac{P(\boldsymbol{x}|y)P(y)}{P(\boldsymbol{x})} = \frac{\prod_j P(x_j|y)\,p(y)}{\prod_j P(x_j)}$$

$$P(y = \text{banana}|\text{long, sweet}) = P(y = B|x_1 = \text{long}, x_2 = \text{sweet})$$

$$= \frac{0.8 \times 0.7 \times 0.1 \times 0.5}{}$$

priors

$$P(Y = B) = 500/1000 = 0.50$$
$$P(Y = O) = 300/1000 = 0.30$$
$$P(Y = M) = 200/1000 = 0.20$$

| Label | Long | ! Long | Sweet | ! Sweet | Yellow | ! Yellow | Total |
|---|---|---|---|---|---|---|---|
| B | 400 | 100 | 350 | 150 | 450 | 50 | 500 |
| O | 0 | 300 | 150 | 150 | 300 | 0 | 300 |
| M | 100 | 100 | 150 | 50 | 50 | 150 | 200 |
| Total | 500 | 500 | 650 | 350 | 800 | 200 | 1000 |

likelihoods (Banana)

$$P(x_1 = \text{long}|Y = B) = 400/500 = 0.80$$
$$P(x_2 = \text{sweet}|Y = B) = 350/500 = 0.70$$
$$P(x_3 = \text{yellow}|Y = B) = 450/500 = 0.90$$

evidence

$$P(x_1 = \text{long}) = 500/1000 = 0.50$$
$$P(x_2 = \text{sweet}) = 650/1000 = 0.65$$
$$P(x_3 = \text{yellow}) = 800/1000 = 0.80$$

# Step 3 : Learning Algorithms (supervised)

## Naïve Bayes

Naïve = conditionally independent features

$$P(y|\boldsymbol{x}) = \frac{P(\boldsymbol{x}|y)P(y)}{P(\boldsymbol{x})} = \frac{\prod_j P(x_j|y)\,p(y)}{\prod_j P(x_j)}$$

$$P(y = \text{banana}|\text{long, sweet}) = P(y = B|x_1 = \text{long}, x_2 = \text{sweet})$$

$$= \frac{0.8 \times 0.7 \times 0.1 \times 0.5}{0.5 \times 0.65 \times 0.8}$$

priors

$$P(Y = B) = 500/1000 = 0.50$$
$$P(Y = O) = 300/1000 = 0.30$$
$$P(Y = M) = 200/1000 = 0.20$$

| Label | Long | ! Long | Sweet | ! Sweet | Yellow | ! Yellow | Total |
|---|---|---|---|---|---|---|---|
| B | 400 | 100 | 350 | 150 | 450 | 50 | 500 |
| O | 0 | 300 | 150 | 150 | 300 | 0 | 300 |
| M | 100 | 100 | 150 | 50 | 50 | 150 | 200 |
| Total | 500 | 500 | 650 | 350 | 800 | 200 | 1000 |

likelihoods (Banana)

$$P(x_1 = \text{long}|Y = B) = 400/500 = 0.80$$
$$P(x_2 = \text{sweet}|Y = B) = 350/500 = 0.70$$
$$P(x_3 = \text{yellow}|Y = B) = 450/500 = 0.90$$

evidence

$$P(x_1 = \text{long}) = 500/1000 = 0.50$$
$$P(x_2 = \text{sweet}) = 650/1000 = 0.65$$
$$P(x_3 = \text{yellow}) = 800/1000 = 0.80$$

# Step 3 : Learning Algorithms (supervised)

## Naïve Bayes

Naïve = conditionally independent features

$$P(y|\boldsymbol{x}) = \frac{P(\boldsymbol{x}|y)P(y)}{P(\boldsymbol{x})} = \frac{\prod_j P(x_j|y)\,p(y)}{\prod_j P(x_j)}$$

$$P(y = \text{banana}|\text{long, sweet}) = P(y = B|x_1 = \text{long}, x_2 = \text{sweet})$$

$$= \frac{0.8 \times 0.7 \times 0.1 \times 0.5}{0.5 \times 0.65 \times 0.8} = 0.11$$

priors

$$P(Y = B) = 500/1000 = 0.50$$
$$P(Y = O) = 300/1000 = 0.30$$
$$P(Y = M) = 200/1000 = 0.20$$

| Label | Long | ! Long | Sweet | ! Sweet | Yellow | ! Yellow | Total |
|---|---|---|---|---|---|---|---|
| B | 400 | 100 | 350 | 150 | 450 | 50 | 500 |
| O | 0 | 300 | 150 | 150 | 300 | 0 | 300 |
| M | 100 | 100 | 150 | 50 | 50 | 150 | 200 |
| Total | 500 | 500 | 650 | 350 | 800 | 200 | 1000 |

likelihoods (Banana)

$$P(x_1 = \text{long}|Y = B) = 400/500 = 0.80$$
$$P(x_2 = \text{sweet}|Y = B) = 350/500 = 0.70$$
$$P(x_3 = \text{yellow}|Y = B) = 450/500 = 0.90$$

evidence

$$P(x_1 = \text{long}) = 500/1000 = 0.50$$
$$P(x_2 = \text{sweet}) = 650/1000 = 0.65$$
$$P(x_3 = \text{yellow}) = 800/1000 = 0.80$$

# Step 3 : Learning Algorithms (supervised)



## K-NN

Simply calculate distances to N nearest neighbors

# Step 3 : Learning Algorithms (supervised)



## K-NN

Simply calculate distances to N nearest neighbors

$$d(\mathbf{x}, \mathbf{x}^{(i)}) = \|\mathbf{x} - \mathbf{x}^{(i)}\|_p$$

$$\|\mathbf{x} - \boldsymbol{\mu}\|_p = \left[\sum_j (x_j - \mu_j)^p\right]^{\frac{1}{p}}$$

# Step 3 : Learning Algorithms (supervised)

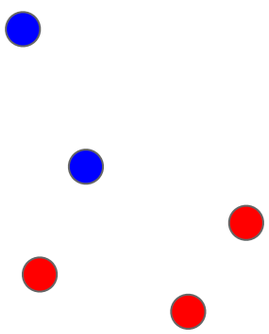


## K-NN

Simply calculate distances to N nearest neighbors

$$d(\mathbf{x}, \mathbf{x}^{(i)}) = \|\mathbf{x} - \mathbf{x}^{(i)}\|_p$$

$$\|\mathbf{x} - \boldsymbol{\mu}\|_p = \left[\sum_j (x_j - \mu_j)^p\right]^{\frac{1}{p}}$$



# Step 3 : Learning Algorithms (supervised)

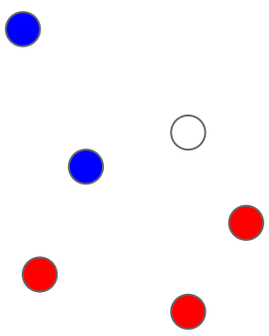


## K-NN

Simply calculate distances to N nearest neighbors

$$d(\mathbf{x}, \mathbf{x}^{(i)}) = \|\mathbf{x} - \mathbf{x}^{(i)}\|_p$$

$$\|\mathbf{x} - \boldsymbol{\mu}\|_p = \left[\sum_j (x_j - \mu_j)^p\right]^{\frac{1}{p}}$$



# Step 3 : Learning Algorithms (supervised)

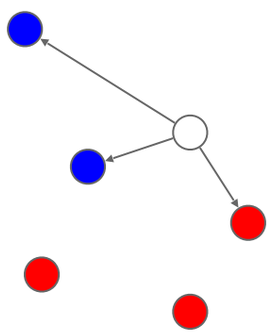


## K-NN

Simply calculate distances to N nearest neighbors

$$d(\mathbf{x}, \mathbf{x}^{(i)}) = \|\mathbf{x} - \mathbf{x}^{(i)}\|_p$$

$$\|\mathbf{x} - \boldsymbol{\mu}\|_p = \left[\sum_j (x_j - \mu_j)^p\right]^{\frac{1}{p}}$$



# Step 3 : Learning Algorithms (supervised)

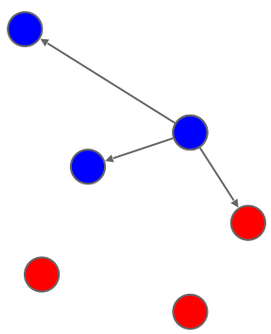


## K-NN

Simply calculate distances to N nearest neighbors

$$d(\mathbf{x}, \mathbf{x}^{(i)}) = \|\mathbf{x} - \mathbf{x}^{(i)}\|_p$$

$$\|\mathbf{x} - \boldsymbol{\mu}\|_p = \left[\sum_j (x_j - \mu_j)^p\right]^{\frac{1}{p}}$$



# Step 3 : Learning Algorithms (supervised)

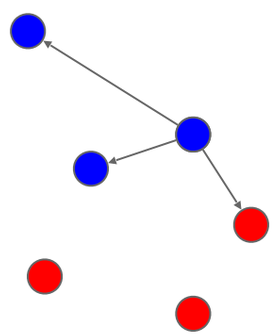


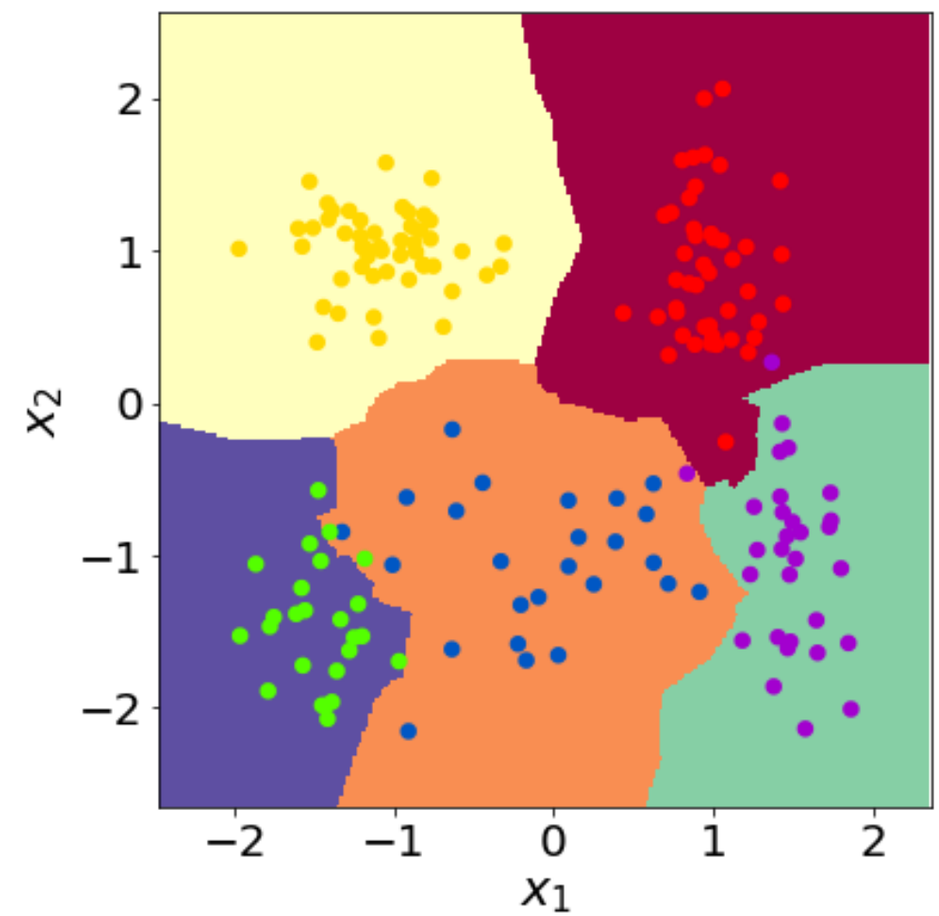
## K-NN

Simply calculate distances to N nearest neighbors

$$d(\mathbf{x}, \mathbf{x}^{(i)}) = \|\mathbf{x} - \mathbf{x}^{(i)}\|_p$$

$$\|\mathbf{x} - \boldsymbol{\mu}\|_p = \left[\sum_j (x_j - \mu_j)^p\right]^{\frac{1}{p}}$$





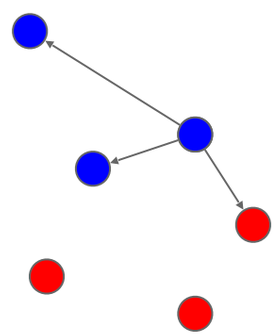
# Step 3 : Learning Algorithms (supervised)

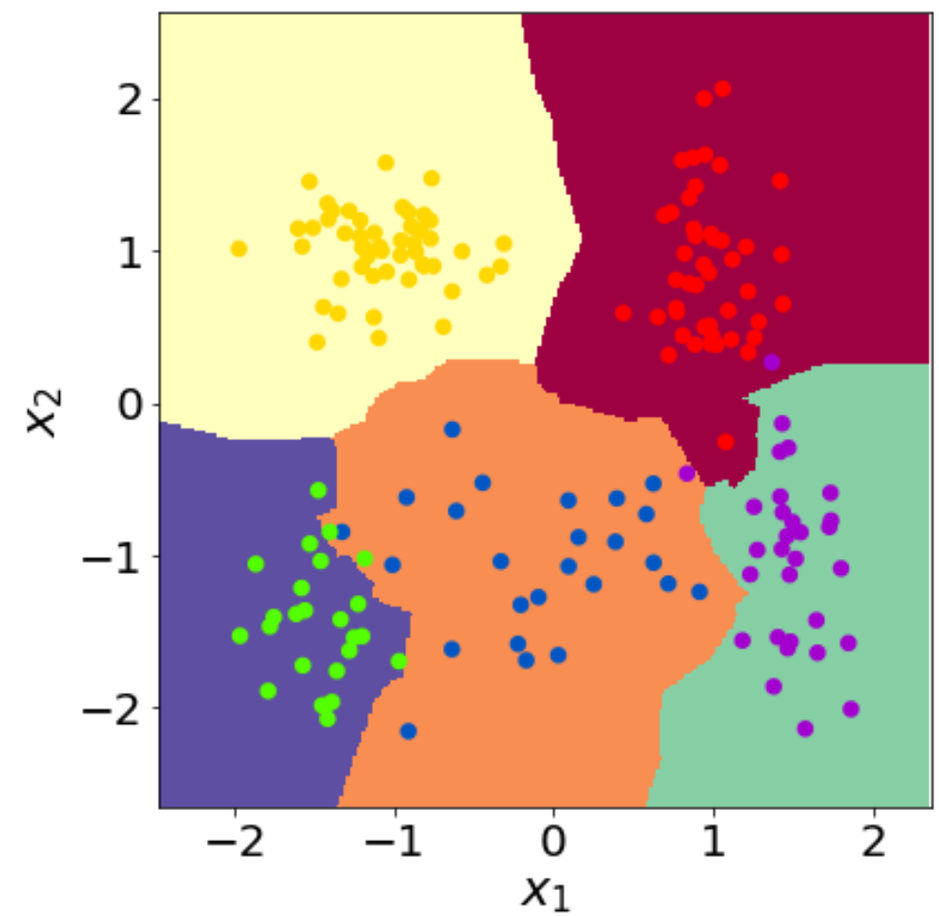
## K-NN

Simply calculate distances to N nearest neighbors

$$d(\mathbf{x}, \mathbf{x}^{(i)}) = \|\mathbf{x} - \mathbf{x}^{(i)}\|_p$$

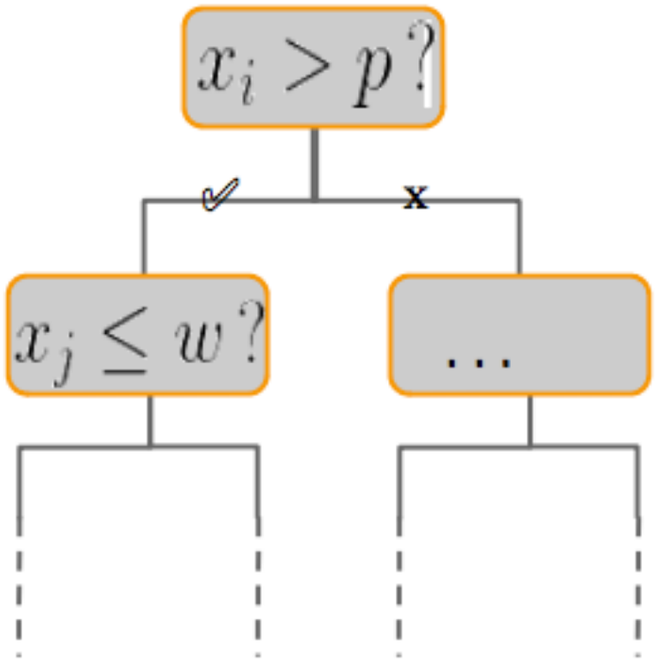
$$\|\mathbf{x} - \boldsymbol{\mu}\|_p = \left[\sum_j (x_j - \mu_j)^p\right]^{\frac{1}{p}}$$






No training required: instance-based algorithm

# Decision Trees

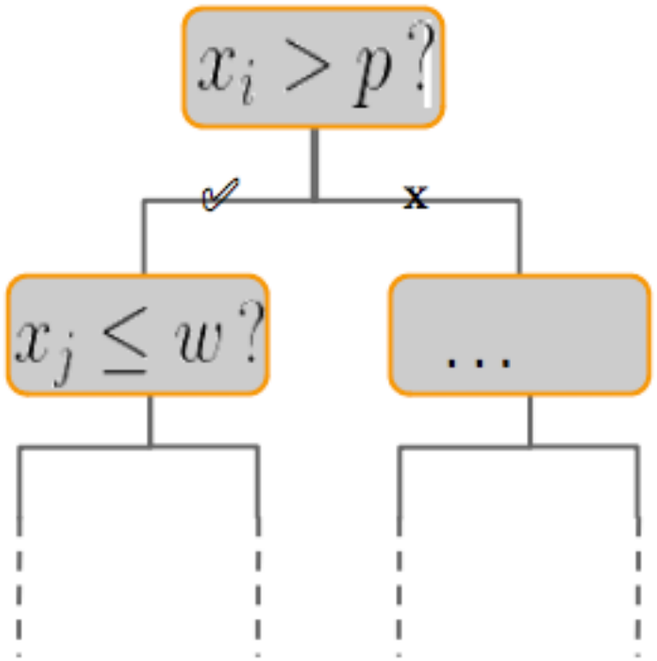


# Step 3 : Learning Algorithms (supervised)



## Decision Trees

- CART - Classification



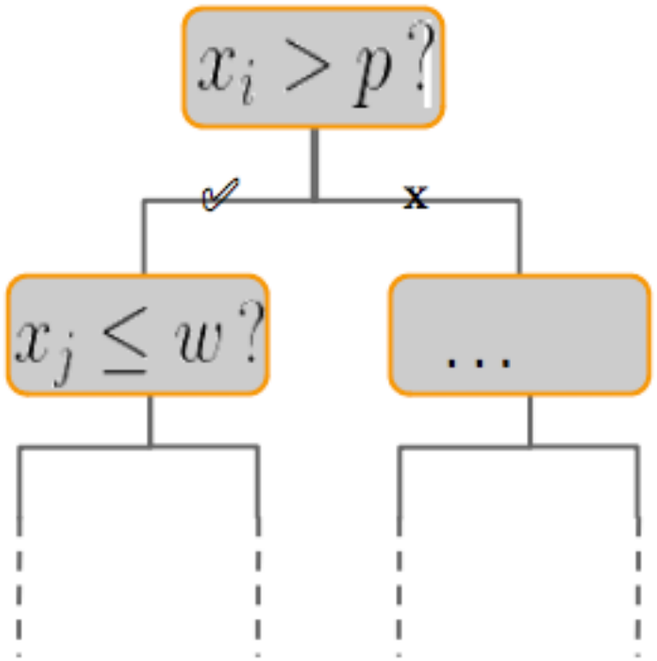
# Step 3 : Learning Algorithms (supervised)



## Decision Trees

- CART - Classification

$$I_G(S) = 1 - \sum_{j=1}^{k} f_j^2$$



# Step 3 : Learning Algorithms (supervised)

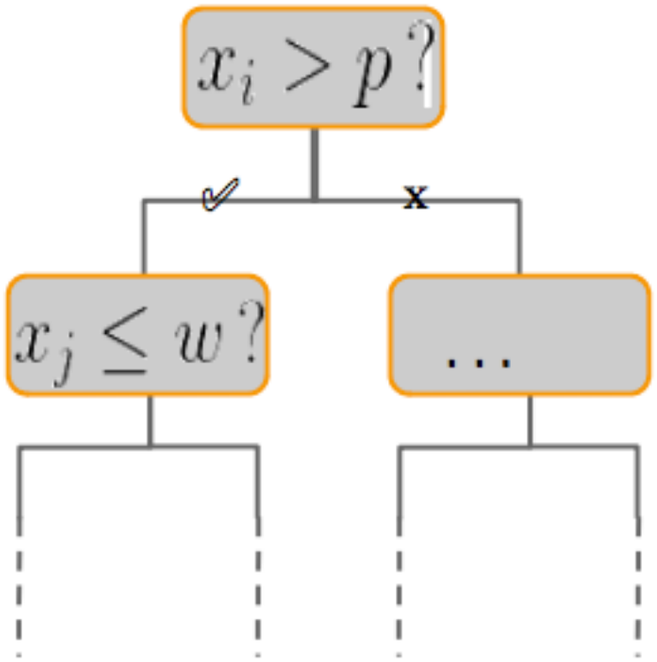


## Decision Trees

- CART - Classification

minimization of GINI coefficient

$$I_G(S) = 1 - \sum_{j=1}^{k} f_j^2$$



# Step 3 : Learning Algorithms (supervised)



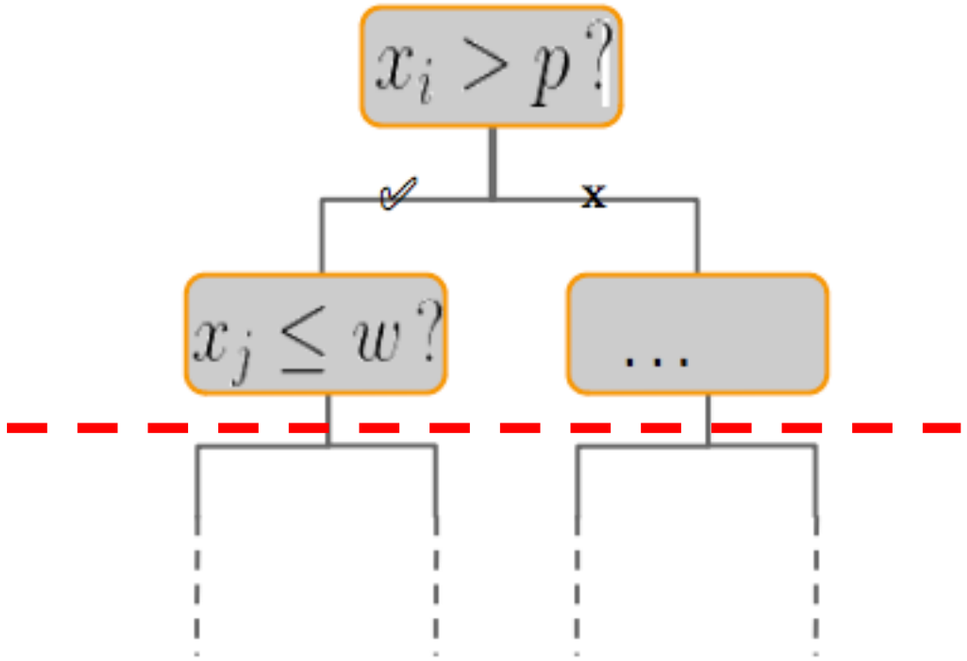
## Decision Trees

- CART - Classification

minimization of GINI coefficient

$$I_G(S) = 1 - \sum_{j=1}^{k} f_j^2$$

relative frequency



# Step 3 : Learning Algorithms (supervised)

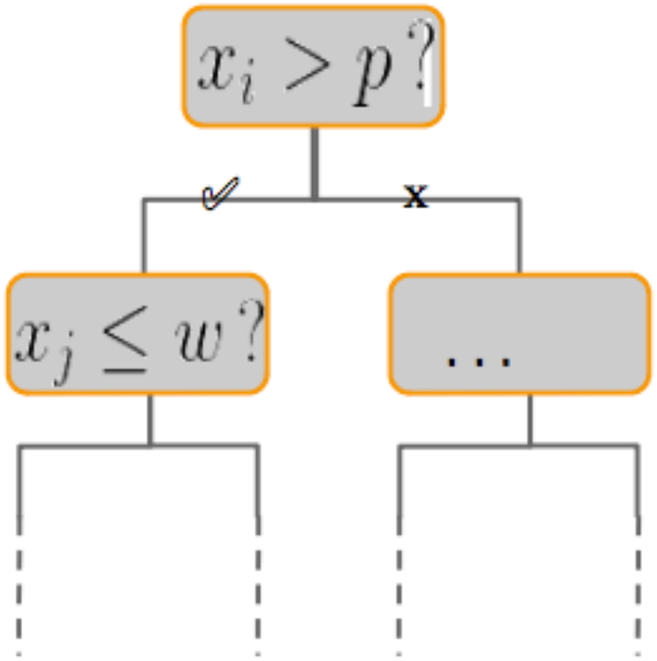
## Decision Trees

- CART - Classification

minimization of GINI coefficient

$$I_G(S) = 1 - \sum_{j=1}^{k} f_j^2$$

relative frequency




- CART - Regression

$$R(S) = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

# Step 3 : Learning Algorithms (supervised)



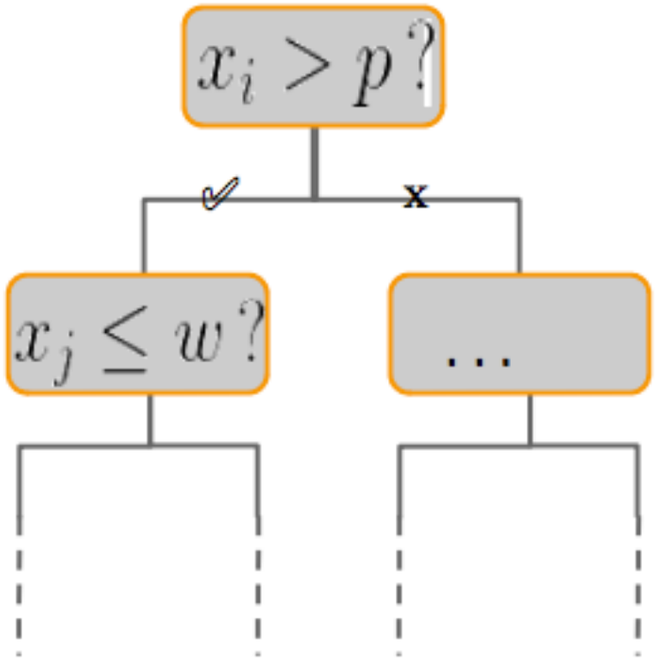
## Decision Trees

- CART - Classification

minimization of GINI coefficient

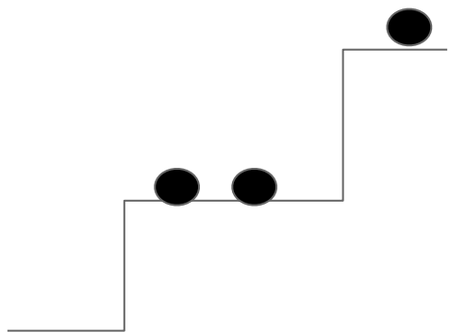
$$I_G(S) = 1 - \sum_{j=1}^{k} f_j^2$$

relative frequency




- CART - Regression

$$R(S) = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$



piecewise
constant function

# Step 3 : Learning Algorithms (supervised)



## Decision Trees

- CART - Classification

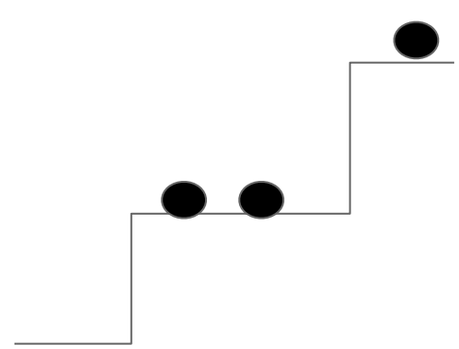
minimization of GINI coefficient

$$I_G(S) = 1 - \sum_{j=1}^{k} f_j^2$$

relative frequency

- CART - Regression

$$R(S) = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$



piecewise
constant function

# Step 3 : Learning Algorithms (supervised)



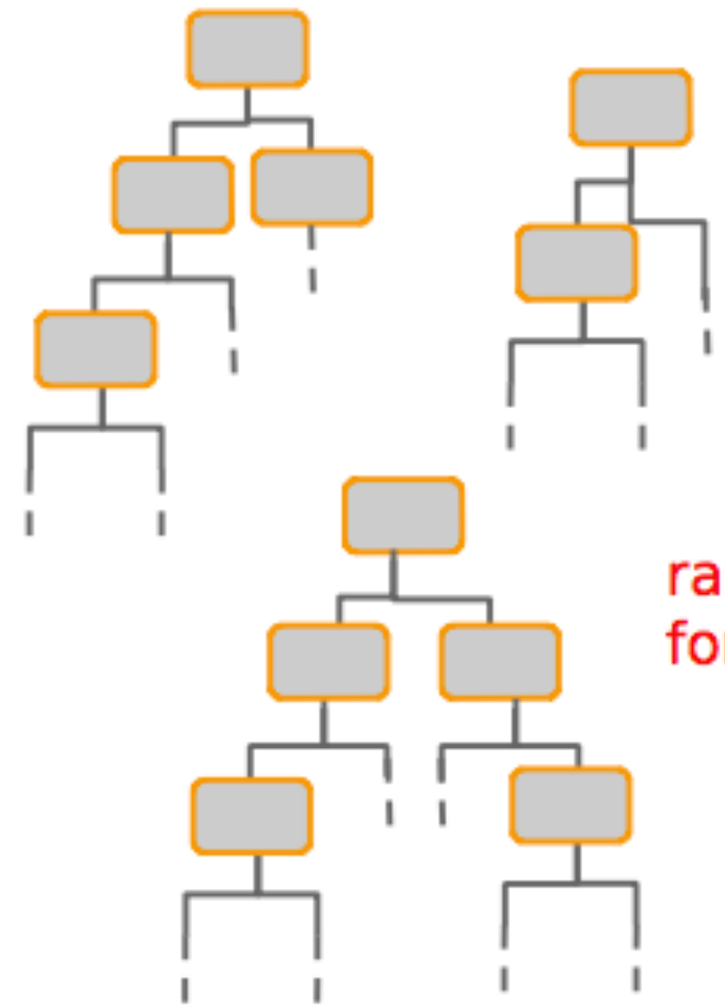
## Decision Trees

- CART - Classification

minimization of GINI coefficient

$$I_G(S) = 1 - \sum_{j=1}^{k} f_j^2$$

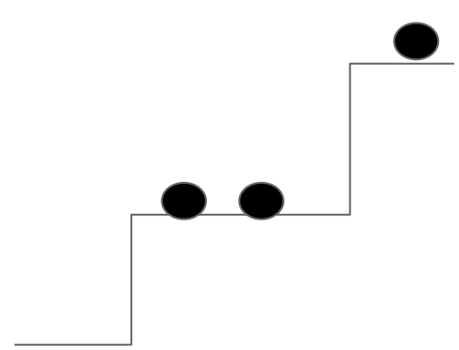
relative frequency



random forests

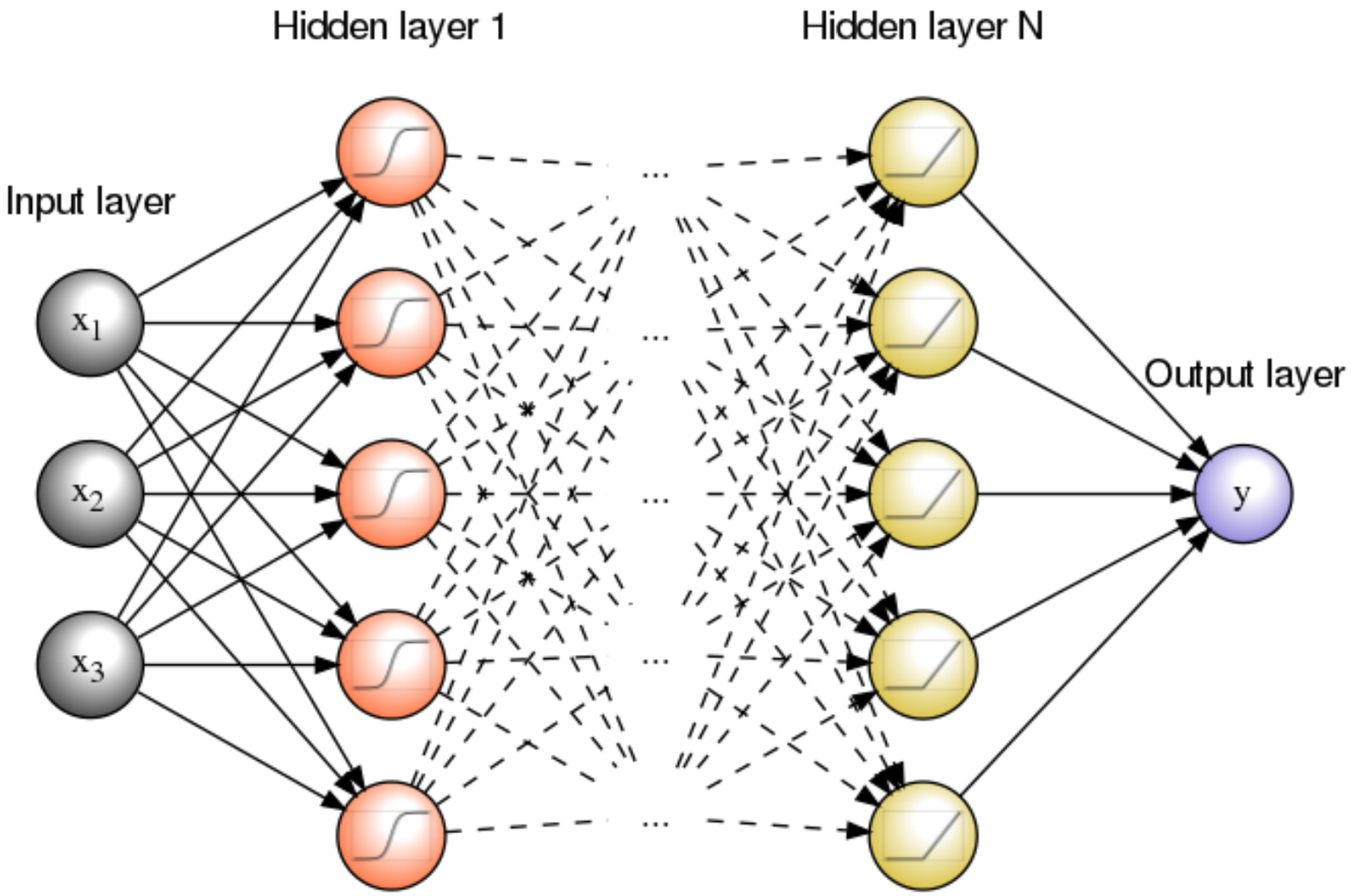
- CART - Regression

$$R(S) = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$



piecewise constant function

# Feed-Forward Artificial Neural Networks

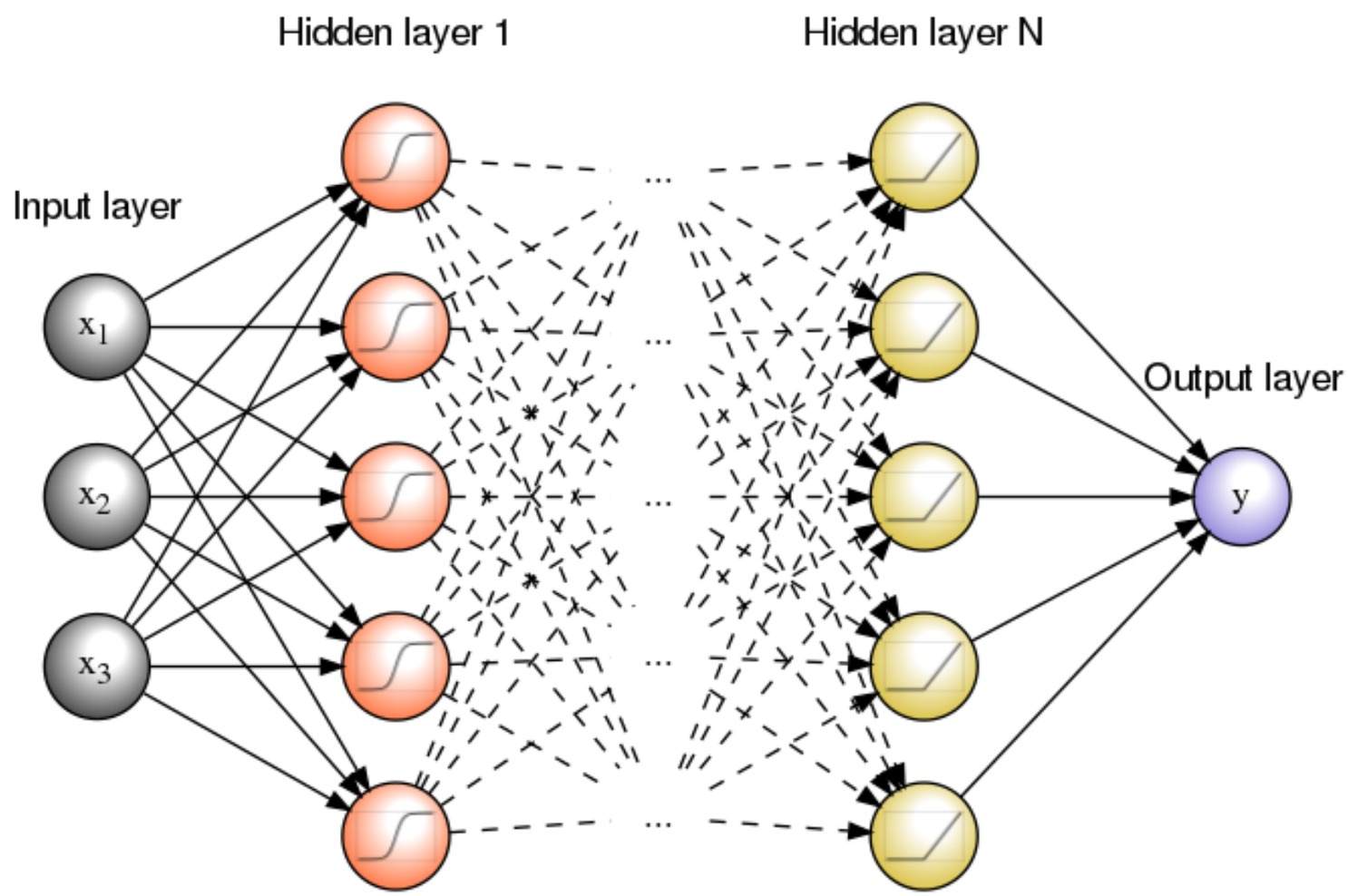


# Step 3 : Learning Algorithms (supervised)



## Feed-Forward Artificial Neural Networks

$$z_i^{(k)} = \omega_{i0}^{(k)} + \sum_j y_j^{(k-1)} \omega_{ij}^{(k)}$$



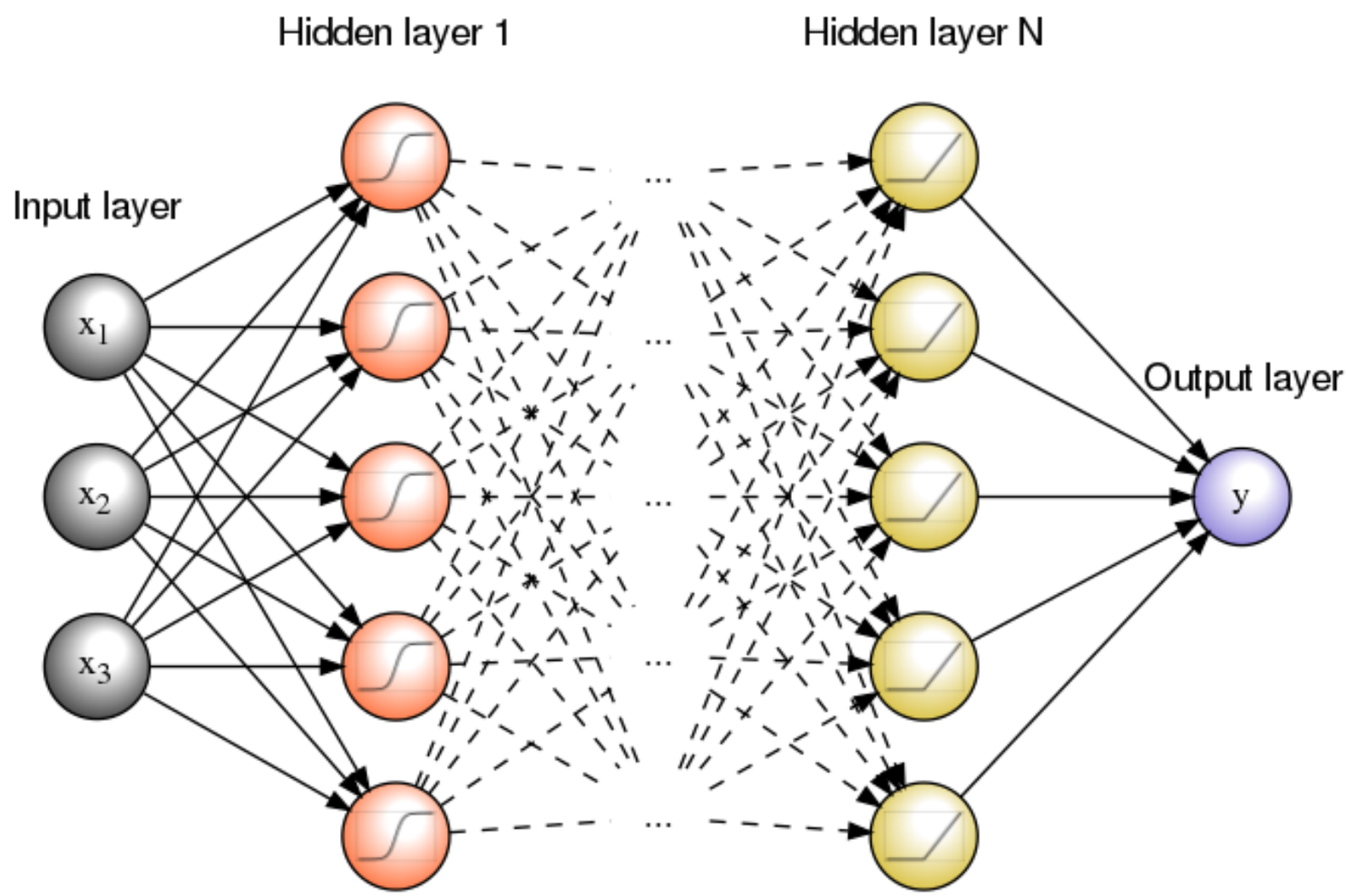
# Step 3 : Learning Algorithms (supervised)



## Feed-Forward Artificial Neural Networks

$$z_i^{(k)} = \omega_{i0}^{(k)} + \sum_j y_j^{(k-1)} \omega_{ij}^{(k)}$$

$$y_i^{(k)} = \frac{1}{1 + e^{-z_i^{(k)}}}$$



# Step 3 : Learning Algorithms (supervised)

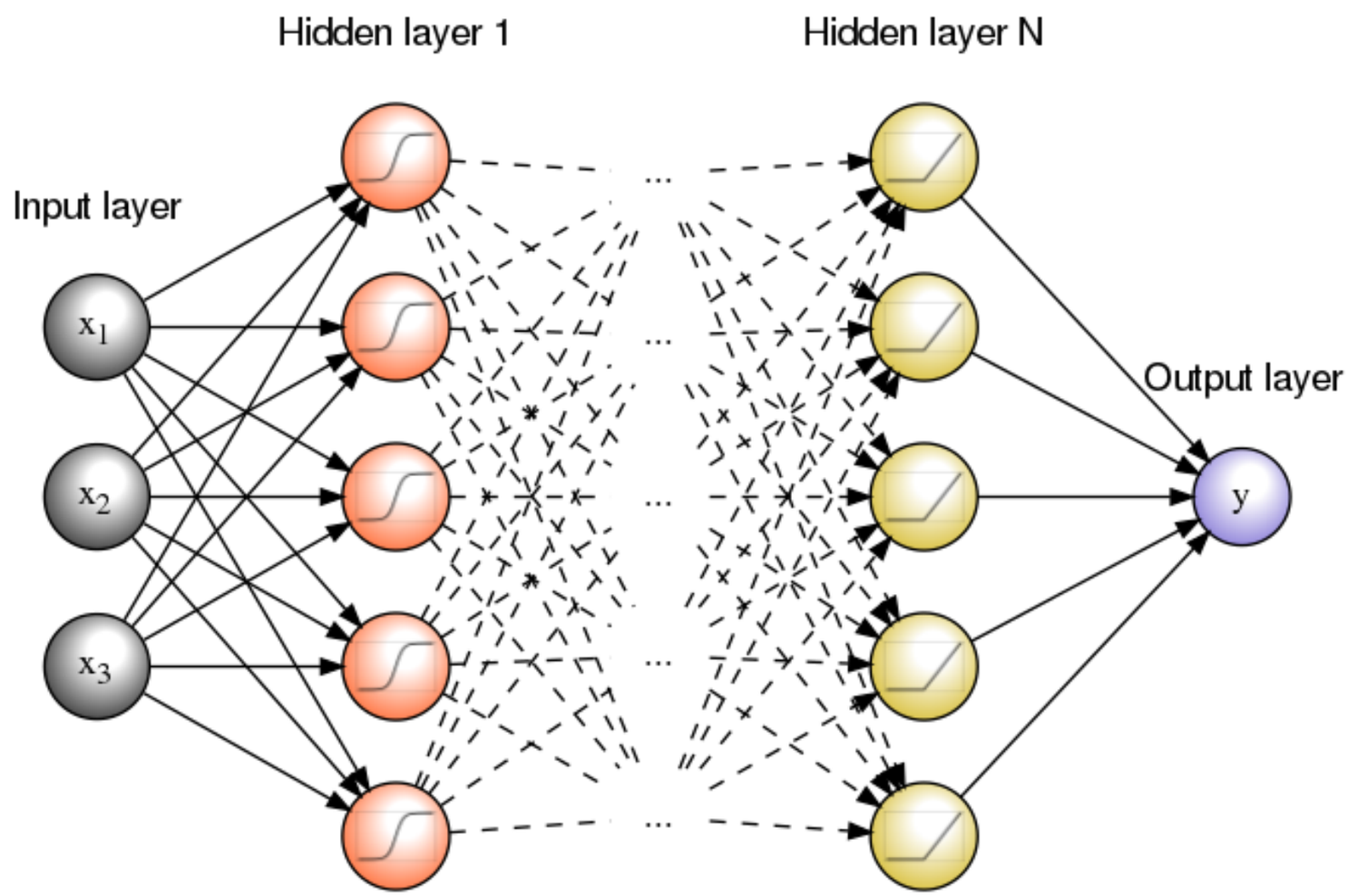


## Feed-Forward Artificial Neural Networks

$$z_i^{(k)} = \omega_{i0}^{(k)} + \sum_j y_j^{(k-1)} \omega_{ij}^{(k)}$$

$$y_i^{(k)} = \frac{1}{1 + e^{-z_i^{(k)}}}$$

$$y_i^{(k)} = \frac{e^{z_i^{(k)}} - e^{-z_i^{(k)}}}{e^{z_i^{(k)}} + e^{-z_i^{(k)}}}$$



# Step 3 : Learning Algorithms (supervised)



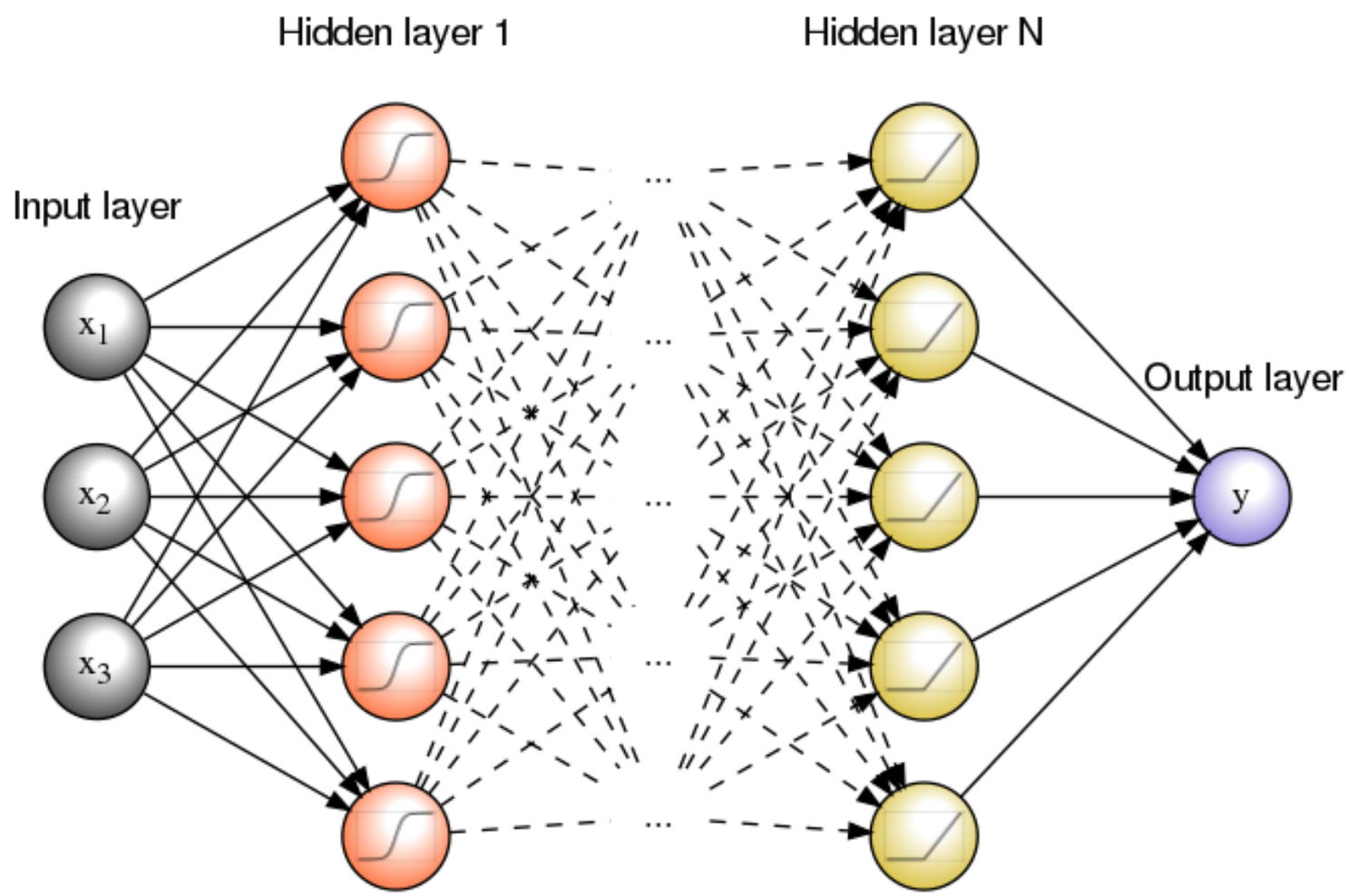
## Feed-Forward Artificial Neural Networks

$$z_i^{(k)} = \omega_{i0}^{(k)} + \sum_j y_j^{(k-1)} \omega_{ij}^{(k)}$$

$$y_i^{(k)} = \frac{1}{1 + e^{-z_i^{(k)}}}$$

$$y_i^{(k)} = \frac{e^{z_i^{(k)}} - e^{-z_i^{(k)}}}{e^{z_i^{(k)}} + e^{-z_i^{(k)}}}$$

$$y_i^{(k)} = \begin{cases} z_i^{(k)}, & z_i^{(k)} > 0 \\ 0, & z_i^{(k)} \leq 0 \end{cases}$$



# Step 3 : Learning Algorithms (supervised)



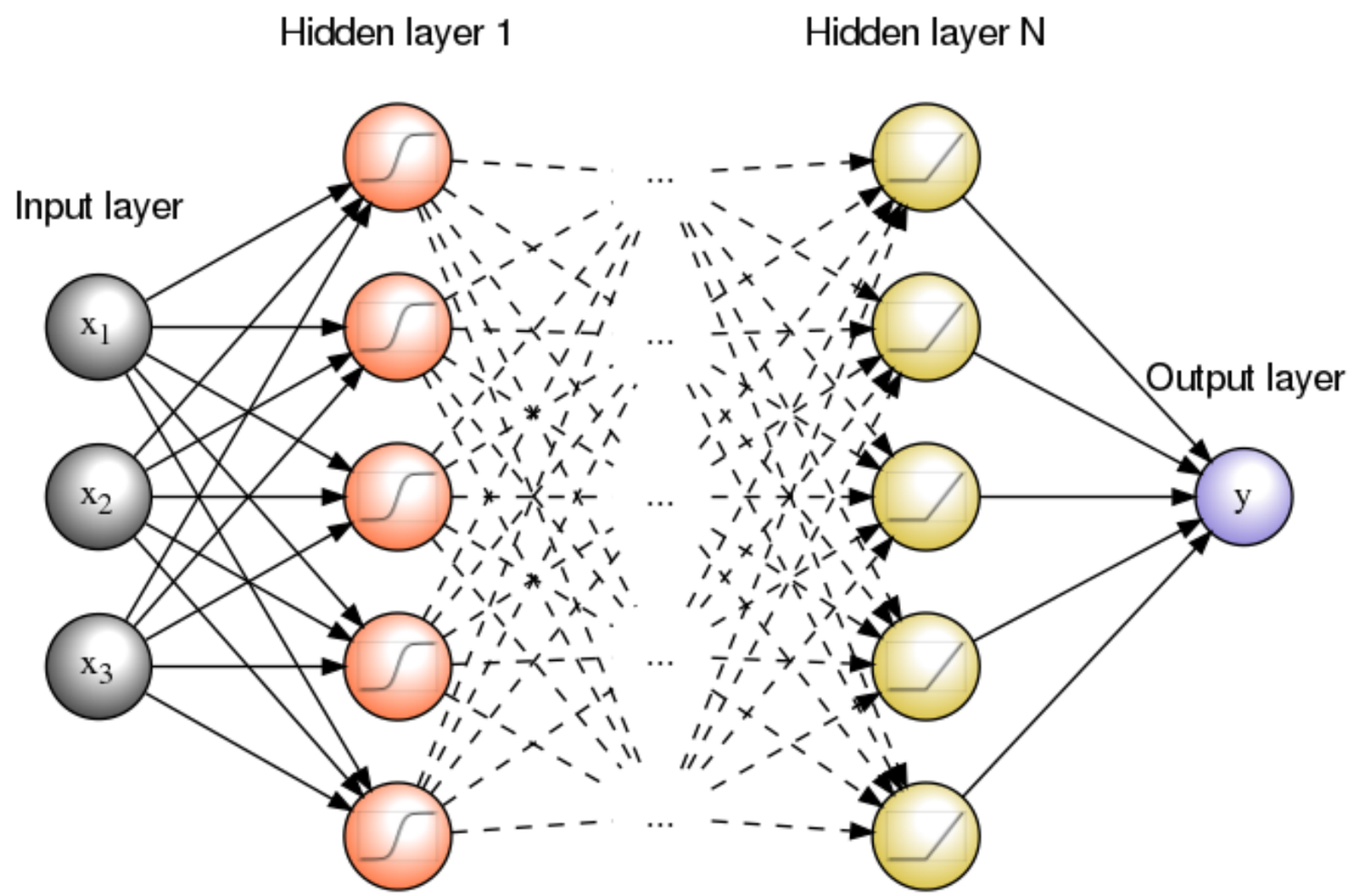
## Feed-Forward Artificial Neural Networks

$$z_i^{(k)} = \omega_{i0}^{(k)} + \sum_j y_j^{(k-1)} \omega_{ij}^{(k)}$$

$$y_i^{(k)} = \frac{1}{1 + e^{-z_i^{(k)}}}$$

$$y_i^{(k)} = \frac{e^{z_i^{(k)}} - e^{-z_i^{(k)}}}{e^{z_i^{(k)}} + e^{-z_i^{(k)}}}$$

$$y_i^{(k)} = \begin{cases} z_i^{(k)}, & z_i^{(k)} > 0 \\ 0, & z_i^{(k)} \leq 0 \end{cases}$$




training: backpropagation

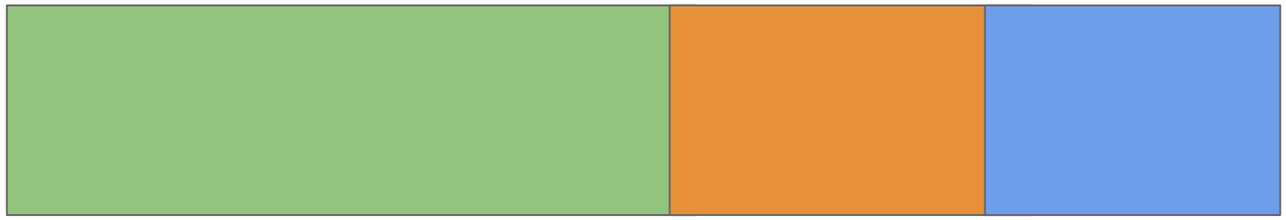
# Model Validation & Evaluation



- Train | Validation | Test split

# Model Validation & Evaluation

- Train | Validation | Test split



# Model Validation & Evaluation

- Train | Validation | Test split

# Model Validation & Evaluation

- Train | Validation | Test split

- If too small train set size: k-fold cross-validation

# Model Validation & Evaluation

- Train | Validation | Test split

- If too small train set size: k-fold cross-validation

- Hyperparameter tuning
  - # of layers in ANN and # of neurons per layer
  - regularization parameters

# Model Validation & Evaluation

- Train | Validation | Test split

- If too small train set size: k-fold cross-validation

- Hyperparameter tuning
  - # of layers in ANN and # of neurons per layer
  - regularization parameters

# Model Validation & Evaluation

- Train | Validation | Test split

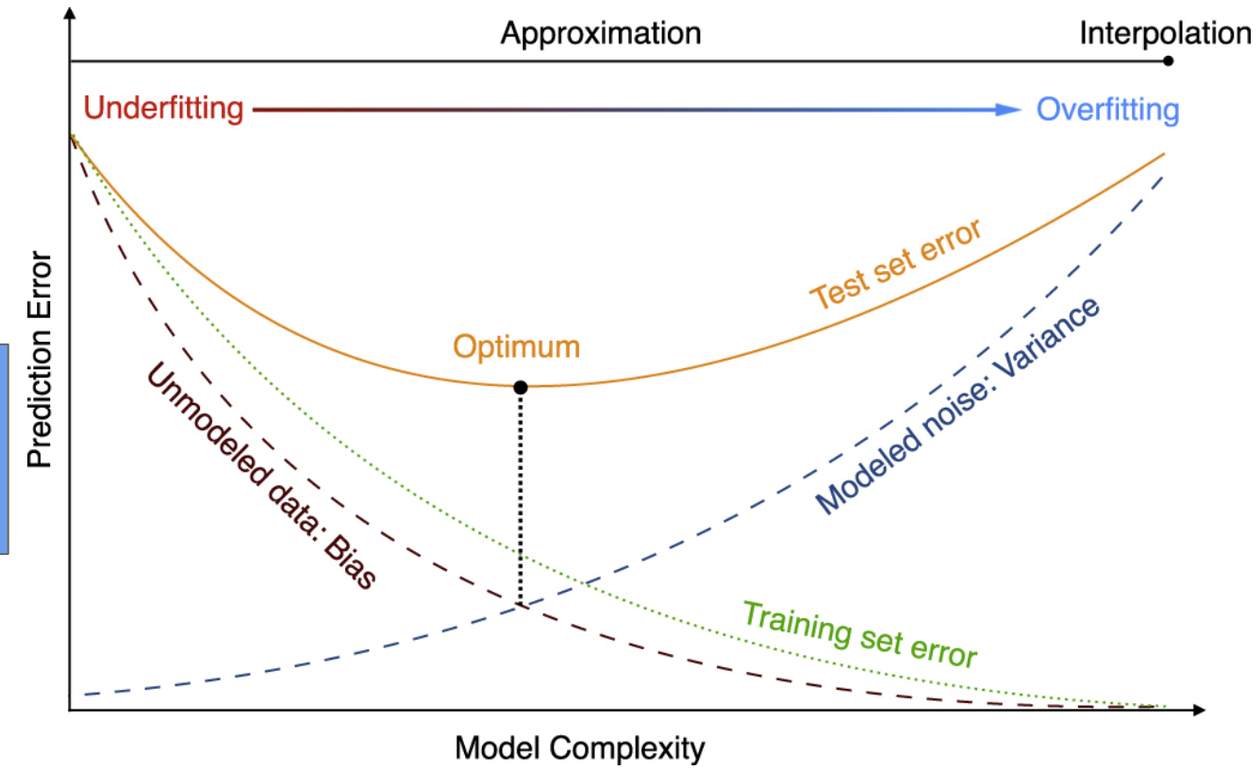
- If too small train set size: k-fold cross-validation

$$E_{cv}^{K} = \frac{1}{K} \sum_{k=1}^{K} \sum_{i=1}^{n_k} L(\hat{y}_k^{(i)}, y^{(i)})$$

- Hyperparameter tuning
  - # of layers in ANN and # of neurons per layer
  - regularization parameters

# Model Validation & Evaluation

- Train | Validation | Test split




- If too small train set size: k-fold cross-validation

$$E_{cv}^{K} = \frac{1}{K} \sum_{k=1}^{K} \sum_{i=1}^{n_k} L(\hat{y}_k^{(i)}, y^{(i)})$$

- Hyperparameter tuning
  - # of layers in ANN and # of neurons per layer
  - regularization parameters

# Acknowledgements

CNPEM
Brazilian Center for Research in Energy and Materials

LNNano
Brazilian Nanotechnology National Laboratory

UFABC

FAPESP

Tutorial Website

https://github.com/gschleder/MLtutorial