

# Machine Learning - Introduction

Nicolas Bourgeois

## 1 Motivation

- Exemples
- L'apprentissage : idée générale
- Valeur d'un résultat

## 2 Visualisation

- Aux origines du problème
- Réduction dimensionnelle
- Approche non linéaire

## 3 Décision

- Introduction
- Ford-Fulkerson / Edmonds
- Construction du graphe
- MATCHING vs CLUSTERING (cliques)
- La classification hiérarchique ascendante

## 4 Qualité d'un modèle

- Différentes sources d'erreur
- Complexité

**1 Motivation**

- Exemples
- L'apprentissage : idée générale
- Valeur d'un résultat

**2 Visualisation**


- Aux origines du problème
- Réduction dimensionnelle
- Approche non linéaire


**3 Décision**

- Introduction
- Ford-Fulkerson / Edmonds
- Construction du graphe
- MATCHING vs CLUSTERING (cliques)
- La classification hiérarchique ascendante

**4 Qualité d'un modèle**

- Différentes sources d'erreur
- Complexité






[Web](#)
[Images](#)
[Vidéos](#)
[Actualités](#)

☒ France
 ☐ Filtre Parental : Strict
 ☐ À tout moment


### Europa Universalis — Wikipédia

Europa Universalis est un jeu vidéo de grande stratégie développé par Paradox Development Studio et sorti en 2000. Il est inspiré d'un jeu de plateau éponyme ...

 [https://fr.wikipedia.org/wiki/Europa\\_Universalis](https://fr.wikipedia.org/wiki/Europa_Universalis)


### Europa Universalis IV sur PC - jeuxvideo.com

Europa Universalis IV sur PC : retrouvez toutes les informations, les tests, les vidéos et actualités du jeu sur tous ses supports. Europa Universalis IV sur PC ...

 [jeuxvideo.com/jeux/pc/00046149-europa-universalis-iv.htm](https://jeuxvideo.com/jeux/pc/00046149-europa-universalis-iv.htm)


### Europa Universalis IV

【送料無料】ミズノ 様式用【グローバルエリート】MG=(金黒製/84cm/900g以上) シルバー(2th21140) 父の日 sale C1806 激安 ...

 [europauniversalis4.com](https://europauniversalis4.com)


### Europa Universalis 4 — Wikipédia

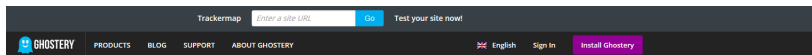
Europa Universalis 4 (stylisé Europa Universalis IV) est un jeu de grande stratégie historique développé par la société suédoise Paradox Development Studio et ...

 [https://fr.wikipedia.org/wiki/Europa\\_Universalis\\_4](https://fr.wikipedia.org/wiki/Europa_Universalis_4)

### Europa Universalis

Europa Universalis est un jeu vidéo de grande stratégie développé par Paradox Development Studio et sorti en 2000. Il est inspiré d'un jeu de plateau éponyme créé par Philippe Thibaut distribué par AWE en 1993.

 [Plus sur Wikipedia \(FR\)](#)

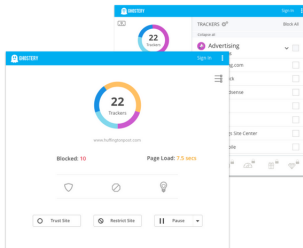


## Faster, safer, and smarter browsing

Ghostery helps you browse smarter by giving you control over ads and tracking technologies to speed up page loads, eliminate clutter, and protect your data.

Install Ghostery

Learn More



**Smart Blocking**  
automatically optimizes page performance as you browse.



**Dynamic UI**  
includes multiple displays and detailed tracker dashboard.



**Enhanced Anti-Tracking**  
anonymizes your data to further protect your privacy.

### This site uses cookies

You are not being tracked since your browser is reporting that you do not want to. This is a setting of your browser so you won't be able to opt-in until you disable the 'Do Not Track' feature.



Traduction



## Traduction de texte

*This demo platform allows you to experience Pure Neural™ machine translation based on the last Research community's findings and SYSTRAN's R&D.*

*You can translate up to 2000 characters of text in the languages proposed below. Check out the [information page](#) to learn more.*

Click h  
ENTER  
SYSTRAN

Français - Détecté



Anglais



Sélectionner un profil

12 juin 2018 - Le 20 juin prochain, le Parlement européen arrêtera sa décision sur la directive Copyright, symbole d'une nouvelle période de régulation de l'Internet. La Quadrature du Net vous invite à **appeler** les eurodéputés pour exiger qu'ils agissent contre l'automatisation de la censure au nom de la protection du droit d'auteur et, plus largement, contre la centralisation du Web.

June 12, 2018 - On June 20, the European Parliament will decide on the Copyright Directive, symbolizing a new period of regulation of the Internet. The Quadrature du Net invites you to **call** on meps to demand that they act against the automation of censorship in the name of copyright protection and, more broadly, against centralization of the Web.







## Motivation

## Exemples

Capitalisme : 0.5  
Socialisme : 0.3  
Sociales : 0.3  
Marxismes : 0.1

Islam : 0.4  
Musulman : 0.4  
Mahomet : 0.2  
Coran : 0.2

Ouvrage : 0.6  
Biographie : 0.3  
Extrait : 0.1  
Préface : 0.1

« ISLAM ET CAPITALISME », DE MAXIME RODINSON  
Le monde musulman, Marx et le socialisme

Paris en 1966. (L'islam) (capitalisme) de (capitalisme) dans le monde (musulman) Pour-  
quoi ces sociétés sont-elles « en retard » ou  
« sous-développées » ? (L'islam) lui-même n'est-il  
l'adoption de notions (socialisme) voire du (socialisme)  
(Mahomet) ? C'est ouvrage pose des problèmes  
d'une brûlante actualité : quel rapport exis-  
tence (islam) de développement (socialisme) ?

PAR ALAIN GRESH

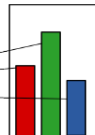
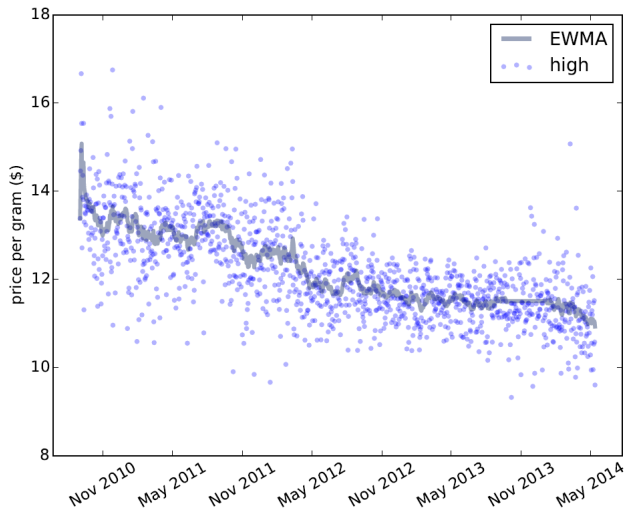
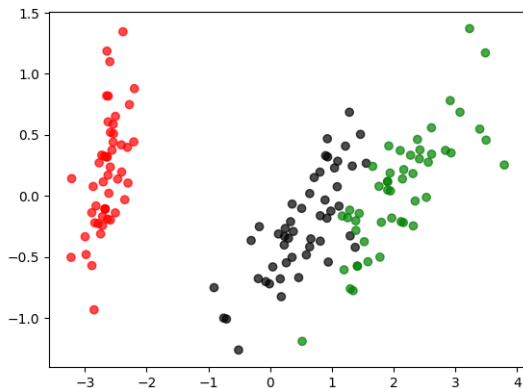


FIGURE : document

FIGURE :  
topics







Motivation

L'apprentissage : idée générale

1 **Motivation**

- Exemples
- L'apprentissage : idée générale
- Valeur d'un résultat

2 **Visualisation**

- Aux origines du problème
- Réduction dimensionnelle
- Approche non linéaire

3 **Décision**

- Introduction
- Ford-Fulkerson / Edmonds
- Construction du graphe
- MATCHING vs CLUSTERING (cliques)
- La classification hiérarchique ascendante

4 **Qualité d'un modèle**

- Différentes sources d'erreur
- Complexité

# Environnement

Python 3, avec les librairies suivantes :

- numpy, scipy
- pandas
- matplotlib, seaborn, pygraphviz
- scikit-learn, tensorflow
- jupyter

## Motivation

L'apprentissage : idée générale

## Install Party Now

&lt;tmp 1&gt; (unsaved) - Interactive Editor for Python

File Edit View Settings Shell Run Tools Help

&lt;tmp 1&gt;

1

Shells

Python

```
>>> pip install pygraphviz
Collecting pygraphviz
  Downloading https://files.pythonhosted.org/packages/98/bb/a32e33f7665b/
Installing collected packages: pygraphviz
  Running setup.py install for pygraphviz: started
    Complete output from command C:\Users\Admin\AppData\Local\Programs\Python\Python36-64\python.exe -c "import sys, os; sys.argv[1:] = ['install', '-r', 'C:\Users\Admin\AppData\Local\Temp\pip-record-...']; exec(' '.join(sys.argv))" install --record C:\Users\Admin\AppData\Local\Temp\pip-record-... running install
    include_dirs=None
    library_dirs=None
    running build
    running build_py
    creating build
    creating build\lib.win-amd64-3.6
    creating build\lib.win-amd64-3.6\pygraphviz
    copying pygraphviz\agraph.py -> build\lib.win-amd64-3.6\pygraphviz
    copying pygraphviz\graphviz.py -> build\lib.win-amd64-3.6\pygraphviz
    copying pygraphviz\release.py -> build\lib.win-amd64-3.6\pygraphviz
    copying pygraphviz\version.py -> build\lib.win-amd64-3.6\pygraphviz
    copying pygraphviz\__init__.py -> build\lib.win-amd64-3.6\pygraphviz
    creating build\lib.win-amd64-3.6\pygraphviz\tests
    copying pygraphviz\tests\test.py -> build\lib.win-amd64-3.6\pygraphviz
    copying pygraphviz\tests\test_attributes.py -> build\lib.win-amd64-3.6\pygraphviz
    copying pygraphviz\tests\test_attribute_defaults.py -> build\lib.win-amd64-3.6\pygraphviz
    copying pygraphviz\tests\test_clear.py -> build\lib.win-amd64-3.6\pygraphviz
    copying pygraphviz\tests\test_drawing.py -> build\lib.win-amd64-3.6\pygraphviz
    copying pygraphviz\tests\test_edge_attributes.py -> build\lib.win-amd64-3.6\pygraphviz
    copying pygraphviz\tests\test_graph.py -> build\lib.win-amd64-3.6\pygraphviz
    copying pygraphviz\tests\test_html.py -> build\lib.win-amd64-3.6\pygraphviz
    copying pygraphviz\tests\test_layout.py -> build\lib.win-amd64-3.6\pygraphviz
    copying pygraphviz\tests\test_node_attributes.py -> build\lib.win-amd64-3.6\pygraphviz
    copying pygraphviz\tests\test_readwrite.py -> build\lib.win-amd64-3.6\pygraphviz
    copying pygraphviz\tests\test_string.py -> build\lib.win-amd64-3.6\pygraphviz
    copying pygraphviz\tests\test_subgraph.py -> build\lib.win-amd64-3.6\pygraphviz
```

# Méthode Générale (I)

- 1 Définir (acquérir) un jeu de données



# Méthode Générale (I)

- 1 Définir (acquérir) un jeu de données
- 2 Préciser un objectif

# Méthode Générale (I)

- 1 Définir (acquérir) un jeu de données
- 2 Préciser un objectif
- 3 Choisir un modèle

# Méthode Générale (I)

- 1 Définir (acquérir) un jeu de données
- 2 Préciser un objectif
- 3 Choisir un modèle
- 4 Identifier des algorithmes

# Méthode Générale (I)

- 1 Définir (acquérir) un jeu de données
- 2 Préciser un objectif
- 3 Choisir un modèle
- 4 Identifier des algorithmes
- 5 Evaluer la performance (fiabilité)

# Apprentissage Supervisé

Observations :

- Variable empirique cible  $\tilde{Y}$  (gain d'un match)
- Variables empiriques explicatives  $\tilde{X}$  (joueurs, terrain)

# Apprentissage Supervisé

Observations :

- Variable empirique cible  $\tilde{Y}$  (gain d'un match)
- Variables empiriques explicatives  $\tilde{X}$  (joueurs, terrain)

Hypothèses :

- $\tilde{X}$  est un ensemble d'observations lié à un processus aléatoire  $X$
- $\tilde{Y}$  est un ensemble d'observations lié à un processus aléatoire  $Y$
- il existe une relation  $Y = f(X)$

# Apprentissage Supervisé

Observations :

- Variable empirique cible  $\tilde{Y}$  (gain d'un match)
- Variables empiriques explicatives  $\tilde{X}$  (joueurs, terrain)

Hypothèses :

- $\tilde{X}$  est un ensemble d'observations lié à un processus aléatoire  $X$
- $\tilde{Y}$  est un ensemble d'observations lié à un processus aléatoire  $Y$
- il existe une relation  $Y = f(X)$

Objectifs :

- Produire une fonction  $\tilde{f}$  à partir de  $\tilde{X}$  et  $\tilde{Y}$
- Telle que  $\tilde{f}$  soit une approximation fiable de  $f$
- On pourra ainsi prédire  $\tilde{Y}' = \tilde{f}(\tilde{X}')$  sur un nouvel échantillon

# Apprentissage non Supervisé

Observations :

- Variable empirique  $\tilde{X}$  (caractéristiques économiques)



# Apprentissage non Supervisé

Observations :

- Variable empirique  $\tilde{X}$  (caractéristiques économiques)

Hypothèses :

- $\tilde{X}$  est un ensemble d'observations lié à un processus aléatoire  $X$

# Apprentissage non Supervisé

Observations :

- Variable empirique  $\tilde{X}$  (caractéristiques économiques)

Hypothèses :

- $\tilde{X}$  est un ensemble d'observations lié à un processus aléatoire  $X$

Objectifs :

- Caractériser autant que possible le processus  $X$
- Par exemple pour classer l'information  $\tilde{X}$
- Ou pour la visualiser
- D'une façon qui reste fiable sur d'autres observations  $\tilde{X}'$


# Exercice


## Exercice

*Dans les exemples précédents, identifier le caractère supervisé ou non du problème et les variables en jeu.*

## Motivation

## L'apprentissage : idée générale





[Web](#)
[Images](#)
[Vidéos](#)
[Actualités](#)

☒ France
 ☐ Filtre Parental : Strict
 ☐ À tout moment

### Europa Universalis — Wikipédia

Europa Universalis est un jeu vidéo de grande stratégie développé par Paradox Development Studio et sorti en 2000. Il est inspiré d'un jeu de plateau éponyme ...

[W https://fr.wikipedia.org/wiki/Europa\\_Universalis](https://fr.wikipedia.org/wiki/Europa_Universalis)

### Europa Universalis IV sur PC - jeuxvideo.com

Europa Universalis IV sur PC : retrouvez toutes les informations, les tests, les vidéos et actualités du jeu sur tous ses supports. Europa Universalis IV sur PC ...

[jeuxvideo.com/jeux/pc/00046149-europa-universalis-iv.htm](https://jeuxvideo.com/jeux/pc/00046149-europa-universalis-iv.htm)

### Europa Universalis IV

【送料無料】ミズノ 様式用【グローバルエリート】MG=(金黒製/84cm/900g以上) シルバー(2th21140) 父の日 sale C1806 激安 ...

[europauniversalis4.com](https://europauniversalis4.com)

### Europa Universalis 4 — Wikipédia

Europa Universalis 4 (stylisé Europa Universalis IV) est un jeu de grande stratégie historique développé par la société suédoise Paradox Development Studio et ...

[W https://fr.wikipedia.org/wiki/Europa\\_Universalis\\_4](https://fr.wikipedia.org/wiki/Europa_Universalis_4)

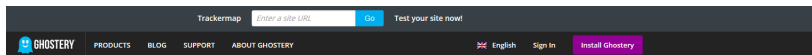
### Europa Universalis

Europa Universalis est un jeu vidéo de grande stratégie développé par Paradox Development Studio et sorti en 2000. Il est inspiré d'un jeu de plateau éponyme créé par Philippe Thibaut distribué par AWE en 1993.

[Plus sur Wikipedia \(FR\)](#)

## Motivation

### L'apprentissage : idée générale

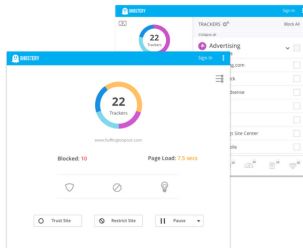


## Faster, safer, and smarter browsing

Ghostery helps you browse smarter by giving you control over ads and tracking technologies to speed up page loads, eliminate clutter, and protect your data.

Install Ghostery

Learn More



**Smart Blocking**  
automatically optimizes page performance as you browse.



**Dynamic UI**  
includes multiple displays and detailed tracker dashboard.



**Enhanced Anti-Tracking**  
anonymizes your data to further protect your privacy.

#### This site uses cookies

You are not being tracked since your browser is reporting that you do not want to. This is a setting of your browser so you won't be able to opt-in until you disable the 'Do Not Track' feature.

## Motivation

## L'apprentissage : idée générale



Traduction



## Traduction de texte

*This demo platform allows you to experience Pure Neural™ machine translation based on the last Research community's findings and SYSTRAN's R&D.*

*You can translate up to 2000 characters of text in the languages proposed below. Check out the [information page](#) to learn more.*

Click h  
ENTER  
SYSTRAN

Français - Détecté



Anglais



Sélectionner un profil

12 juin 2018 - Le 20 juin prochain, le Parlement européen arrêtera sa décision sur la directive Copyright, symbole d'une nouvelle période de régulation de l'Internet. La Quadrature du Net vous invite à **appeler** les eurodéputés pour exiger qu'ils agissent contre l'automatisation de la censure au nom de la protection du droit d'auteur et, plus largement, contre la centralisation du Web.

June 12, 2018 - On June 20, the European Parliament will decide on the Copyright Directive, symbolizing a new period of regulation of the Internet. The Quadrature du Net invites you to **call** on meps to demand that they act against the automation of censorship in the name of copyright protection and, more broadly, against centralization of the Web.



## L'apprentissage : idée générale



## Motivation

## L'apprentissage : idée générale

Capitalisme : 0.5  
Socialisme : 0.3  
Sociales : 0.3  
Marxismes : 0.1

Islam : 0.4  
Musulman : 0.4  
Mahomet : 0.2  
Coran : 0.2

Ouvrage : 0.6  
Biographie : 0.3  
Extrait : 0.1  
Préface : 0.1

« ISLAM ET CAPITALISME », DE MAXIME RODINSON  
Le monde musulman, Marx et le socialisme

Paris en 1966. [L'islam] [capitalisme] de [capitalisme] dans le monde [musulman] Pour-  
quoi ces sociétés sont-elles « en retard » ou  
« sous-développées » ? [L'islam] lui-même n'est-il  
l'adoption de notions [socialisme] voire du [socialisme]  
[Mahomet] ? Ce livre pose des problèmes  
d'une brûlante actualité : quel rapport exis-  
te-t-il entre [l'islam] et le [socialisme] ?

PAR ALAIN GRESH

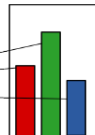
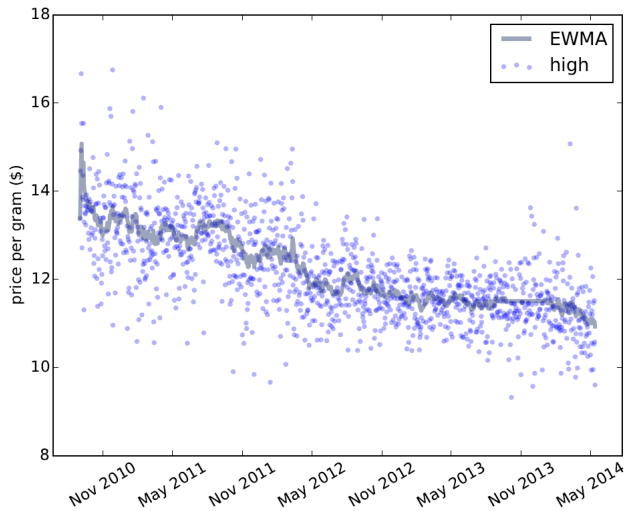


FIGURE : document

FIGURE :  
topics

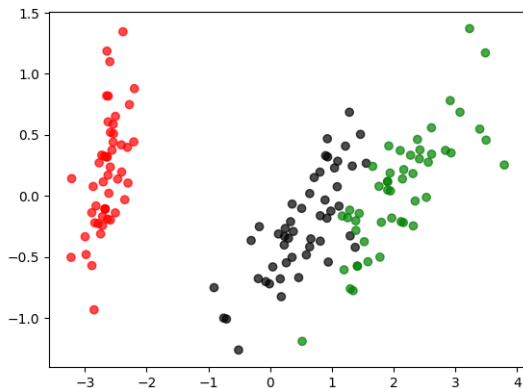
## Motivation

## L'apprentissage : idée générale



## Motivation

### L'apprentissage : idée générale



## Motivation

## L'apprentissage : idée générale



## 1 Motivation

- Exemples
- L'apprentissage : idée générale
- Valeur d'un résultat

## 2 Visualisation

- Aux origines du problème
- Réduction dimensionnelle
- Approche non linéaire

## 3 Décision

- Introduction
- Ford-Fulkerson / Edmonds
- Construction du graphe
- MATCHING vs CLUSTERING (cliques)
- La classification hiérarchique ascendante

## 4 Qualité d'un modèle

- Différentes sources d'erreur
- Complexité

# Qualité

- Adéquation de la prédiction : marge d'erreur, risque d'erreur
- S'évalue sur un échantillon de test différent de l'échantillon d'apprentissage
- Temps de calcul, vitesse de convergence

# Surapprentissage

## Exercice

*Montrez qu'il est toujours possible de trouver un modèle parfaitement fiable sur l'échantillon d'apprentissage.*

# Surapprentissage

## Exercice

*Montrez qu'il est toujours possible de trouver un modèle parfaitement fiable sur l'échantillon d'apprentissage.*

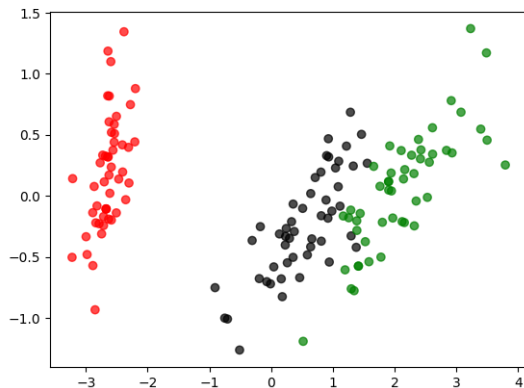
## Exercice

*Montrez que ce modèle peut être en fait très mauvais sur un échantillon de test.*



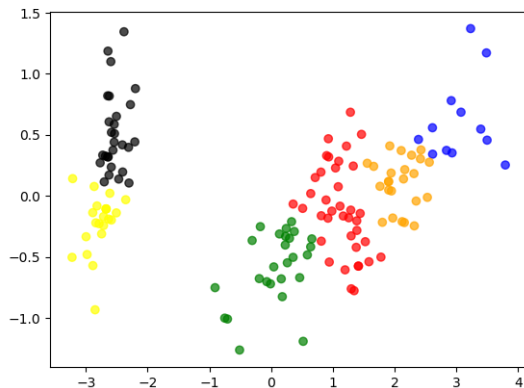
## Motivation

Valeur d'un résultat



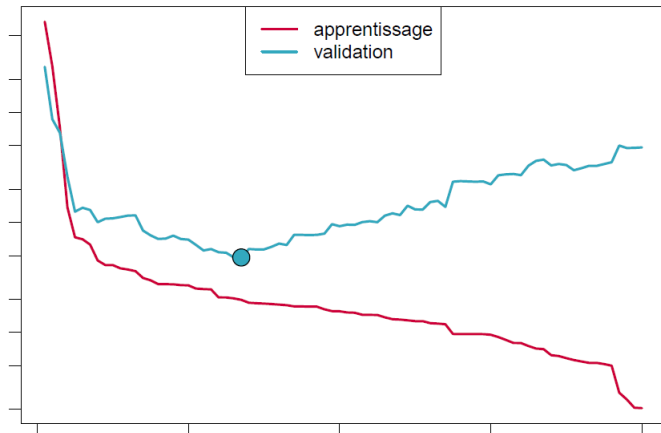
## Motivation

Valeur d'un résultat



## Motivation

Valeur d'un résultat



# Warnings

## Exercice

*Essayez d'identifier un maximum de sources d'échec dans un processus d'apprentissage.*

Solutions dans la partie 4. . .

## 1 Motivation

- Exemples
- L'apprentissage : idée générale
- Valeur d'un résultat

## 2 Visualisation

- Aux origines du problème
- Réduction dimensionnelle
- Approche non linéaire

## 3 Décision

- Introduction
- Ford-Fulkerson / Edmonds
- Construction du graphe
- MATCHING vs CLUSTERING (cliques)
- La classification hiérarchique ascendante

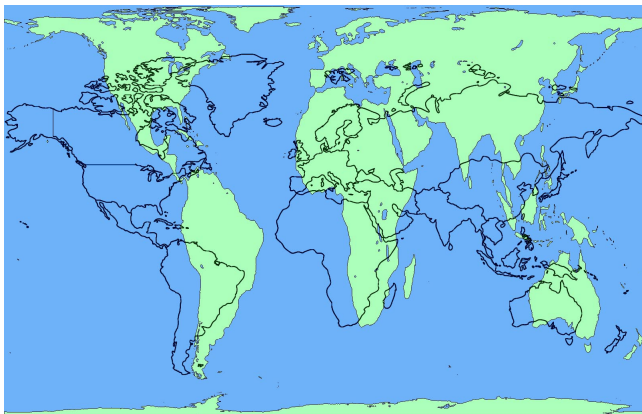
## 4 Qualité d'un modèle

- Différentes sources d'erreur
- Complexité

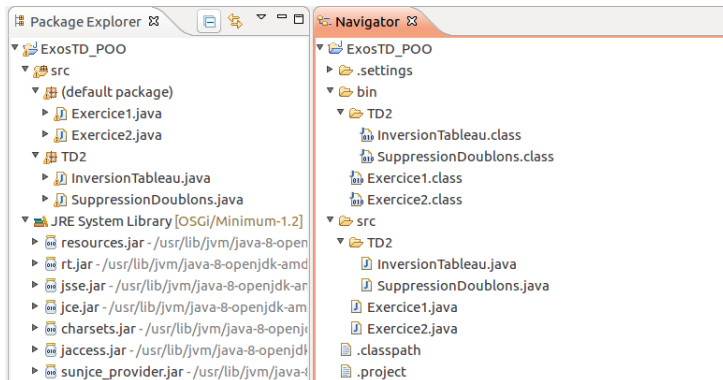
# Des représentations naturelles ?



# Des représentations naturelles ?

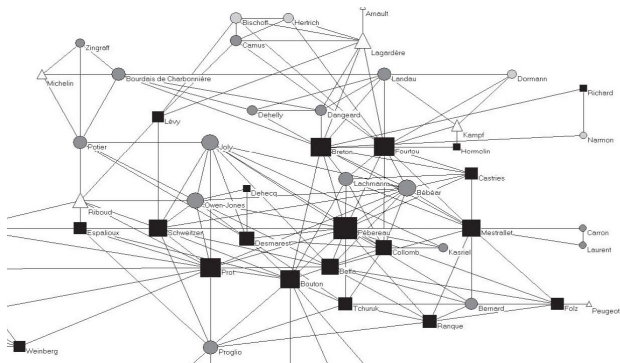


# Des représentations naturelles ?





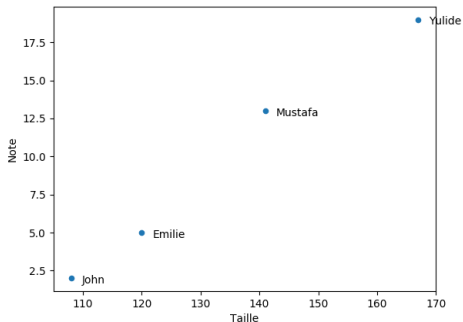
# Des représentations naturelles ?



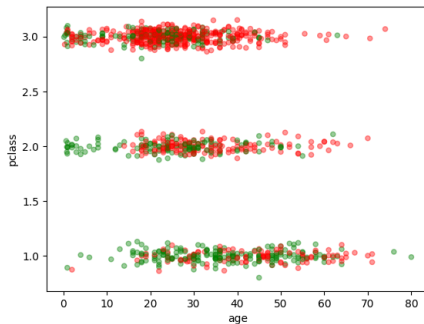
# Problème : multidimensionnalité



# Problème : multidimensionnalité



# Problème : multidimensionnalité



## Problème : multidimensionnalité

0.000	0.026	0.271	0.000	0.000	0.001	0.036	0.039	0.025	0.002	0.002	0.001	0.001	0.009	0.000	0.002	0.002	0.000	0.000	0.000
0.000	0.053	0.044	0.004	0.003	0.004	0.143	0.005	0.002	0.007	0.002	0.007	0.002	0.003	0.001	0.002	0.003	0.003	0.003	0.003
0.000	0.021	0.134	0.000	0.001	0.000	0.000	0.003	0.006	0.001	0.000	0.000	0.000	0.003	0.000	0.000	0.000	0.002	0.000	0.000
0.000	0.006	0.046	0.002	0.001	0.001	0.055	0.004	0.001	0.001	0.001	0.009	0.038	0.003	0.001	0.000	0.002	0.001	0.025	0.000
0.000	0.000	0.013	0.000	0.000	0.000	0.000	0.000	0.080	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.000	0.001	0.057	0.000	0.000	0.000	0.018	0.005	0.009	0.000	0.000	0.003	0.000	0.209	0.000	0.000	0.001	0.000	0.000	0.000
0.000	0.001	0.096	0.001	0.001	0.001	0.112	0.005	0.001	0.001	0.001	0.022	0.001	0.007	0.001	0.000	0.003	0.001	0.001	0.000
0.000	0.066	0.019	0.000	0.000	0.047	0.011	0.010	0.000	0.001	0.000	0.001	0.001	0.001	0.021	0.064	0.001	0.010	0.012	0.000
0.000	0.001	0.049	0.001	0.000	0.000	0.030	0.003	0.000	0.000	0.000	0.002	0.000	0.002	0.000	0.000	0.001	0.000	0.000	0.000
0.000	0.002	0.252	0.002	0.001	0.001	0.031	0.037	0.001	0.001	0.001	0.003	0.001	0.005	0.000	0.000	0.002	0.000	0.000	0.000
0.000	0.001	0.013	0.000	0.000	0.000	0.001	0.000	0.002	0.001	0.001	0.001	0.000	0.002	0.120	0.001	0.000	0.000	0.000	0.000
0.000	0.007	0.063	0.004	0.001	0.000	0.072	0.005	0.001	0.002	0.001	0.015	0.002	0.002	0.001	0.000	0.001	0.197	0.001	0.000
0.000	0.005	0.021	0.000	0.000	0.000	0.006	0.001	0.000	0.001	0.000	0.002	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.000	0.014	0.090	0.001	0.000	0.002	0.032	0.044	0.001	0.014	0.000	0.005	0.001	0.000	0.002	0.001	0.001	0.001	0.001	0.000

# Problème : multidimensionnalité



## 1 Motivation

- Exemples
- L'apprentissage : idée générale
- Valeur d'un résultat

## 2 Visualisation

- Aux origines du problème
- Réduction dimensionnelle
- Approche non linéaire

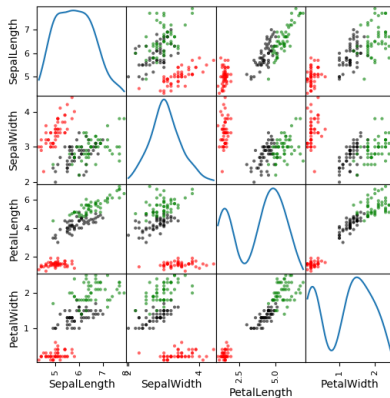
## 3 Décision

- Introduction
- Ford-Fulkerson / Edmonds
- Construction du graphe
- MATCHING vs CLUSTERING (cliques)
- La classification hiérarchique ascendante

## 4 Qualité d'un modèle

- Différentes sources d'erreur
- Complexité

# Scatter Matrix



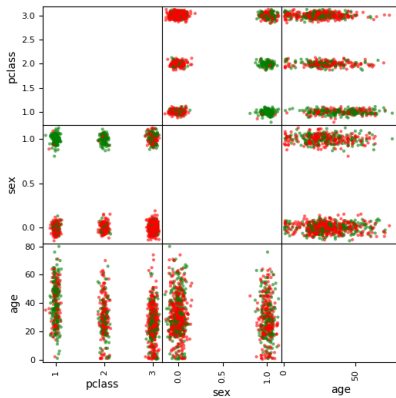


# Scatter Matrix

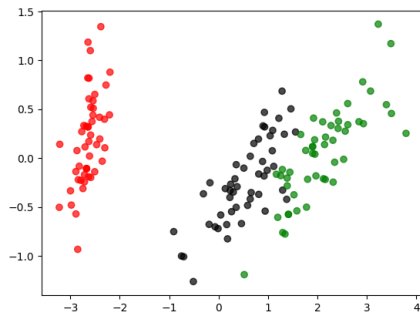
```
import pandas as pd
from matplotlib import pyplot as plt
from pandas.plotting import scatter_matrix

iris_data = pd.read_csv('./iris.csv')
colMap={"Iris-setosa":"red","Iris-virginica":"green",
        "Iris-versicolor":"black"}
colors=list(map(lambda x:colMap.get(x),iris_data.Name))
scatter_matrix(iris_data, alpha=0.6, figsize=(6, 6),
               diagonal='kde',c=colors)
plt.show()
```

# Scatter Matrix



# ACP



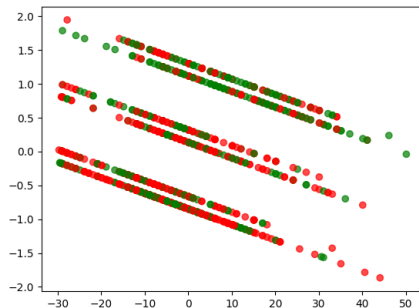
# ACP

```
import matplotlib.pyplot as plt
from sklearn import datasets
from sklearn.decomposition import PCA

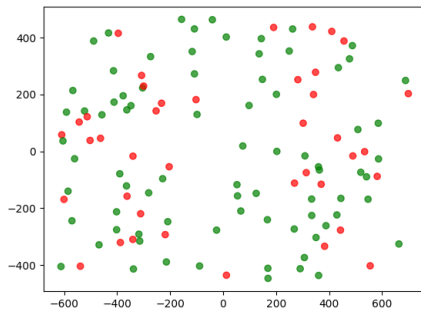
iris = datasets.load_iris()
X,Y = iris.data, iris.target
colMap={0:"red",1:"green",2:"black"}
colors=list(map(lambda x:colMap.get(x),Y))
X_2ev = PCA(n_components=2).fit_transform(X)
plt.scatter(X_2ev[:,0],X_2ev[:,1],alpha=0.7,c=colors)

plt.show()
```

# ACP



# ACP



## 1 Motivation

- Exemples
- L'apprentissage : idée générale
- Valeur d'un résultat

## 2 Visualisation

- Aux origines du problème
- Réduction dimensionnelle
- Approche non linéaire

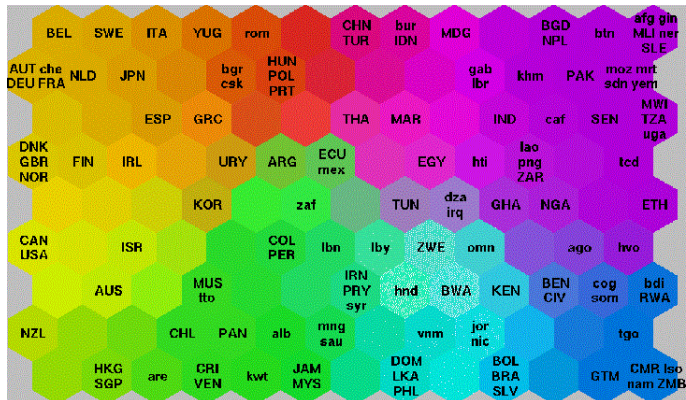
## 3 Décision

- Introduction
- Ford-Fulkerson / Edmonds
- Construction du graphe
- MATCHING vs CLUSTERING (cliques)
- La classification hiérarchique ascendante

## 4 Qualité d'un modèle

- Différentes sources d'erreur
- Complexité

# Kohonen Map

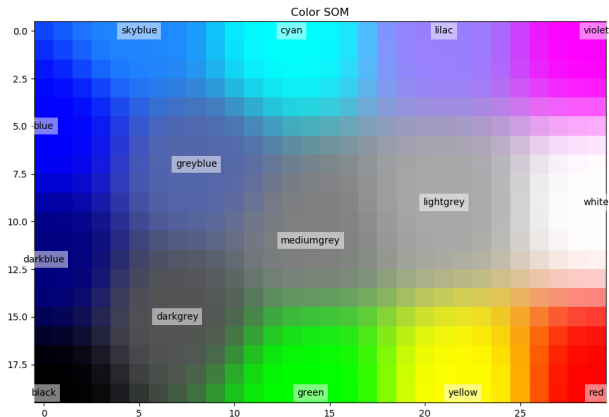




# Kohonen Map

```
from som1 import *
colors = np.array([
    [0., 0., 0.], [0., 0., 1.], [0., 0., 0.5], [0.125, 0.529, 1.0],
    [0.33, 0.4, 0.67], [0.6, 0.5, 1.0], [0., 1., 0.],
    [1., 0., 0.], [0., 1., 1.], [1., 0., 1.], [1., 1., 0.],
    [1., 1., 1.], [.33, .33, .33], [.5, .5, .5], [.66, .66, .66]])
color_names = \
    ['black', 'blue', 'darkblue', 'skyblue',
     'greyblue', 'lilac', 'green', 'red',
     'cyan', 'violet', 'yellow', 'white',
     'darkgrey', 'mediumgrey', 'lightgrey']
som = SOM(20, 30, 3, 400)
som.train(colors)
image_grid = som.get_centroids()
mapped = som.map_vects(colors)
plt.imshow(image_grid)
plt.title('Color_SOM')
for i, m in enumerate(mapped):
    plt.text(m[1], m[0], color_names[i], ha='center', va='center',
            bbox=dict(facecolor='white', alpha=0.5, lw=0))
plt.show()
```

# Kohonen Map



## 1 Motivation

- Exemples
- L'apprentissage : idée générale
- Valeur d'un résultat

## 2 Visualisation

- Aux origines du problème
- Réduction dimensionnelle
- Approche non linéaire

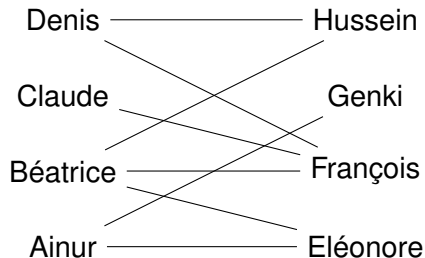
## 3 Décision

- Introduction
- Ford-Fulkerson / Edmonds
- Construction du graphe
- MATCHING vs CLUSTERING (cliques)
- La classification hiérarchique ascendante

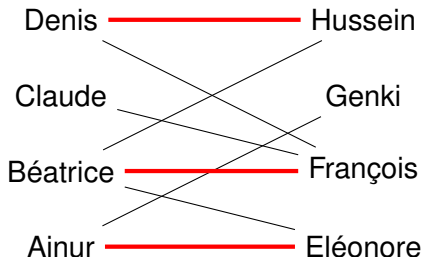
## 4 Qualité d'un modèle

- Différentes sources d'erreur
- Complexité

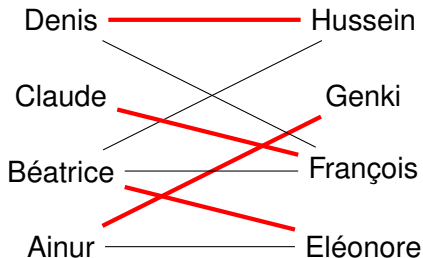
# Un graphe de compatibilité

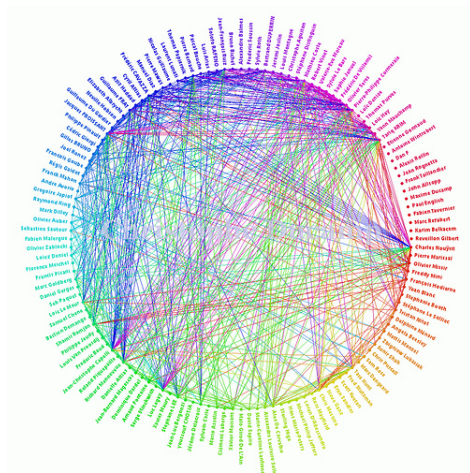


# Une allocation sous-optimale



# Une allocation optimale





# Formalisation du problème

Soit un graphe  $G$  défini par un ensemble de sommets  $V$  et un ensemble d'arêtes  $E$ . On cherche



# Formalisation du problème

Soit un graphe  $G$  défini par un ensemble de sommets  $V$  et un ensemble d'arêtes  $E$ . On cherche

un sous-ensemble d'arêtes  $F \subset E$  :

# Formalisation du problème

Soit un graphe  $G$  défini par un ensemble de sommets  $V$  et un ensemble d'arêtes  $E$ . On cherche

un sous-ensemble d'arêtes  $F \subset E$  :

tel que deux arêtes ne soient pas incidentes

# Formalisation du problème

Soit un graphe  $G$  défini par un ensemble de sommets  $V$  et un ensemble d'arêtes  $E$ . On cherche

un sous-ensemble d'arêtes  $F \subset E$  :

tel que deux arêtes ne soient pas incidentes

de taille maximale

## 1 Motivation

- Exemples
- L'apprentissage : idée générale
- Valeur d'un résultat

## 2 Visualisation

- Aux origines du problème
- Réduction dimensionnelle
- Approche non linéaire

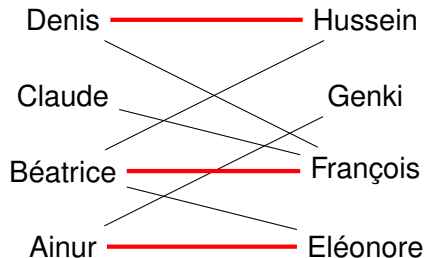
## 3 Décision

- Introduction
- Ford-Fulkerson / Edmonds
- Construction du graphe
- MATCHING vs CLUSTERING (cliques)
- La classification hiérarchique ascendante

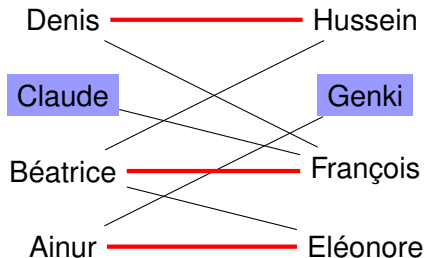
## 4 Qualité d'un modèle

- Différentes sources d'erreur
- Complexité

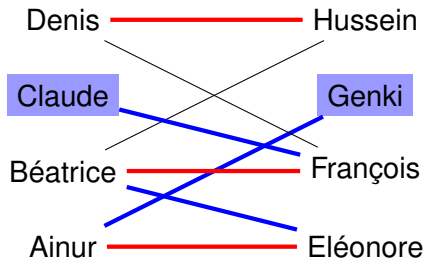
# Chaîne augmentante



## Deux sommets isolés



# Une chaîne alternée

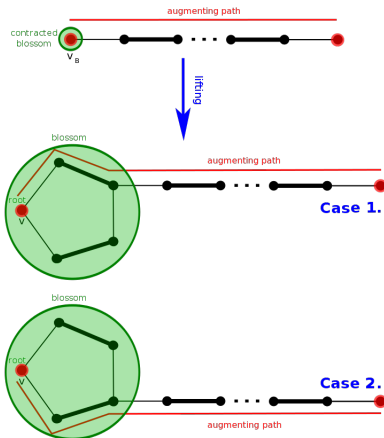


# Chaîne augmentante

```
def chaine(couplage, sommets, aretes):  
    isoles = {x for x in sommets  
              if all(x not in e for e in couplage)}  
    v = isoles.pop()  
    reste, voisin, w, sol = sommets.copy(), None, v, []  
    while reste != set() and voisin not in isoles:  
        voisin = {y for y in reste  
                  if {y, w} in aretes}.pop()  
        sol.append({w, voisin})  
        reste.remove(voisin)  
        if voisin not in isoles:  
            w = {y for y in reste  
                 if {y, voisin} in couplage}.pop()  
            reste.remove(w)  
    return(sol)
```



# Le cas non-biparti



## 1 Motivation

- Exemples
- L'apprentissage : idée générale
- Valeur d'un résultat

## 2 Visualisation

- Aux origines du problème
- Réduction dimensionnelle
- Approche non linéaire

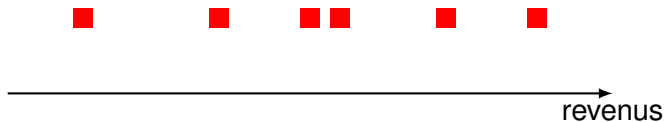
## 3 Décision

- Introduction
- Ford-Fulkerson / Edmonds
- Construction du graphe
- MATCHING vs CLUSTERING (cliques)
- La classification hiérarchique ascendante

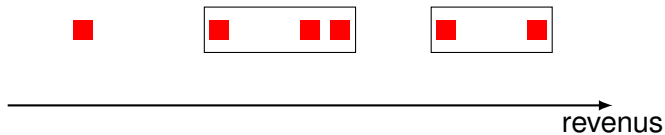
## 4 Qualité d'un modèle

- Différentes sources d'erreur
- Complexité

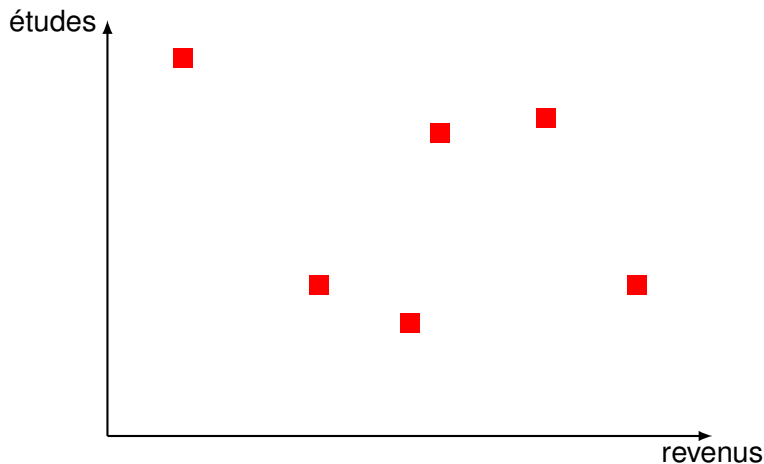
# Données monovariées



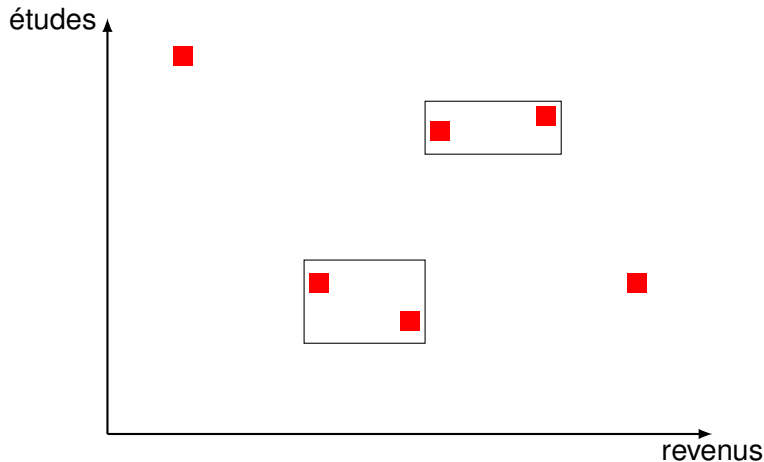
# Données monovariées



# Données bivariées



# Données bivariées

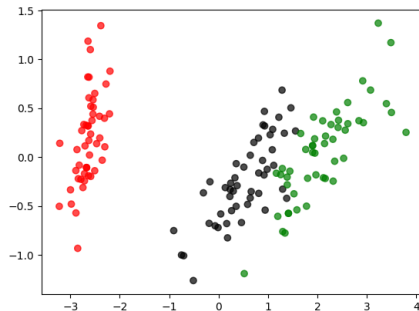


# Données multivariées

Comment représenter sur un écran un classement selon des dizaines ou des milliers de critères ?

Comment déterminer des compatibilités entre des individus représentés par autant de variables ?

# Réduction dimensionnelle





# Réduction dimensionnelle

```
import matplotlib.pyplot as plt
from sklearn import datasets
from sklearn.decomposition import PCA

iris = datasets.load_iris()
X,Y = iris.data, iris.target
colMap={0:"red",1:"green",2:"black"}
colors=list(map(lambda x:colMap.get(x),Y))
X_2ev = PCA(n_components=2).fit_transform(X)
plt.scatter(X_2ev[:,0],X_2ev[:,1],alpha=0.7,c=colors)

plt.show()
```

# À vous de jouer !

Importez le fichier `data2.csv` et essayez de construire une représentation ou de modéliser un graphe de compatibilité.

# Données numériques

Normalisation (exemple) :

$$x' = \frac{x - xmin}{xmax - xmin}$$

Agrégation (exemple) :

$$d(X, Y) = \sqrt{\sum (x_i - y_i)^2}$$

# Données par modalités

Distance binaire :

$$d(X, Y) = \#\{x_i \neq y_i\} = \sum_{x_i \neq y_i} 1$$

Distance pondérée :

$$d(X, Y) = \sum_{x_i \neq y_i} \omega_i$$

# Exemple

X : BLOND, BAC+5, MODEM, 43 ans

Y : BLOND, BAC+2, NPA, 36 ans

# Exemple

$X$  : BLOND, BAC+5, MODEM, 43 ans

$Y$  : BLOND, BAC+2, NPA, 36 ans

$X'$  : BLOND, 0.6, MODEM, 0.7

$Y'$  : BLOND, 0.3, NPA, 0.55

# Exemple

$X$  : BLOND, BAC+5, MODEM, 43 ans

$Y$  : BLOND, BAC+2, NPA, 36 ans

$X'$  : BLOND, 0.6, MODEM, 0.7

$Y'$  : BLOND, 0.3, NPA, 0.55

$$d(X, Y) = \sqrt{0 + (0.6 - 0.3)^2 + 1 + (0.7 - 0.55)^2}$$

# À vous de jouer !

Construisez une matrice de distances sur les données du fichier `data2.csv`.



# Principe

On fixe un seuil, par exemple  $S = N/4$ , où  $N$  est le nombre de variables.

# Principe

On fixe un seuil, par exemple  $S = N/4$ , où  $N$  est le nombre de variables.

On considère que deux sommets doivent être reliés si et seulement si leur distance est inférieure au seuil.

$$(X, Y) \in G \iff d(X, Y) < S$$

# Exemple

$X$  : BLOND, 0.6, MODEM, 0.7

$Y$  : BLOND, 0.3, NPA, 0.55

$Z$  : BRUN, 0.5, LR, 0.8

$T$  : BRUN, 0.2, NPA, 0.2

# Exemple

	X	Y	Z	T
X		1.45	2.30	2.90
Y			2.45	1.45
Z				1.90
T				

# Exemple

	X	Y	Z	T
X		1.45	2.3	2.9
Y			2.45	1.45
Z				1.9
T				

# À vous de jouer !

- 1) Fixez un seuil et utilisez la matrice de l'exercice précédent pour construire des proximités entre les individus.
- 2) Essayez de produire le graphe correspondant.

## 1 Motivation

- Exemples
- L'apprentissage : idée générale
- Valeur d'un résultat

## 2 Visualisation

- Aux origines du problème
- Réduction dimensionnelle
- Approche non linéaire

## 3 Décision

- Introduction
- Ford-Fulkerson / Edmonds
- Construction du graphe
- MATCHING vs CLUSTERING (cliques)
- La classification hiérarchique ascendante

## 4 Qualité d'un modèle

- Différentes sources d'erreur
- Complexité

## Rappel : le MATCHING

Soit un graphe  $G$  défini par un ensemble de sommets  $V$  et un ensemble d'arêtes  $E$ . On cherche



# Rappel : le MATCHING

Soit un graphe  $G$  défini par un ensemble de sommets  $V$  et un ensemble d'arêtes  $E$ . On cherche

un sous-ensemble d'arêtes  $F \subset E$  :

# Rappel : le MATCHING

Soit un graphe  $G$  défini par un ensemble de sommets  $V$  et un ensemble d'arêtes  $E$ . On cherche

un sous-ensemble d'arêtes  $F \subset E$  :

tel que deux arêtes ne soient pas incidentes

# Rappel : le MATCHING

Soit un graphe  $G$  défini par un ensemble de sommets  $V$  et un ensemble d'arêtes  $E$ . On cherche

un sous-ensemble d'arêtes  $F \subset E$  :

tel que deux arêtes ne soient pas incidentes

de taille maximale

## Rappel : le CLUSTERING

Soit un graphe  $G$  défini par un ensemble de sommets  $V$  et un ensemble d'arêtes  $E$ . On cherche

## Rappel : le CLUSTERING

Soit un graphe  $G$  défini par un ensemble de sommets  $V$  et un ensemble d'arêtes  $E$ . On cherche

Une division de  $V$  en sous-ensembles disjoints  $V_1, V_2, V_3 \dots$

## Rappel : le CLUSTERING

Soit un graphe  $G$  défini par un ensemble de sommets  $V$  et un ensemble d'arêtes  $E$ . On cherche

Une division de  $V$  en sous-ensembles disjoints  $V_1, V_2, V_3 \dots$

avec un maximum d'arêtes à l'intérieur de chaque  $V_i$

## Rappel : le CLUSTERING

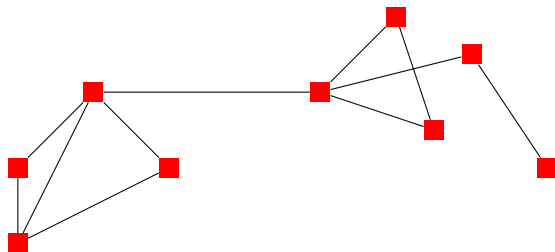
Soit un graphe  $G$  défini par un ensemble de sommets  $V$  et un ensemble d'arêtes  $E$ . On cherche

Une division de  $V$  en sous-ensembles disjoints  $V_1, V_2, V_3 \dots$

avec un maximum d'arêtes à l'intérieur de chaque  $V_i$

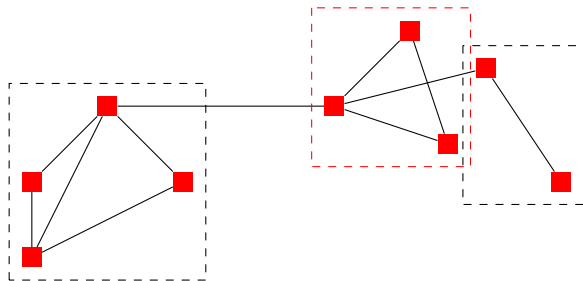
et un minimum à l'extérieur, entre les différents  $V_i$ .

# Exemple





# Exemple



## Différents types d'objectifs

- Ne regrouper que des éléments tous deux à deux compatibles :

$$x \in V_i, y \in V_i \implies (x, y) \in G$$

# Différents types d'objectifs

- Ne regrouper que des éléments tous deux à deux compatibles :

$$x \in V_i, y \in V_i \implies (x, y) \in G$$

- Ratio inter/intra minimal :

$$\min \frac{\#\{(x, y) \in G, x \in V_i, y \in V_j\}}{\#\{(x, y) \in G, x, y \in V_i\}}$$

# À vous de jouer !

Trouvez un clustering pertinent sur l'exemple des exercices précédents.

## 1 Motivation

- Exemples
- L'apprentissage : idée générale
- Valeur d'un résultat

## 2 Visualisation

- Aux origines du problème
- Réduction dimensionnelle
- Approche non linéaire

## 3 Décision

- Introduction
- Ford-Fulkerson / Edmonds
- Construction du graphe
- MATCHING vs CLUSTERING (cliques)
- La classification hiérarchique ascendante

## 4 Qualité d'un modèle

- Différentes sources d'erreur
- Complexité

# Principe

On va procéder de façon itérative.

# Principe

On va procéder de façon itérative.

A chaque étape on regroupe les deux éléments les plus proches.

# Principe

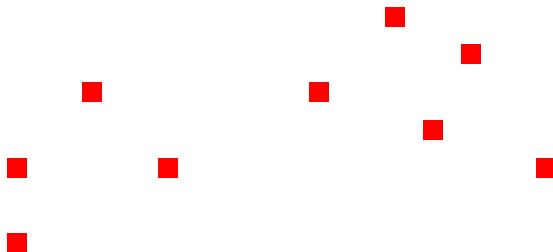
On va procéder de façon itérative.

A chaque étape on regroupe les deux éléments les plus proches.

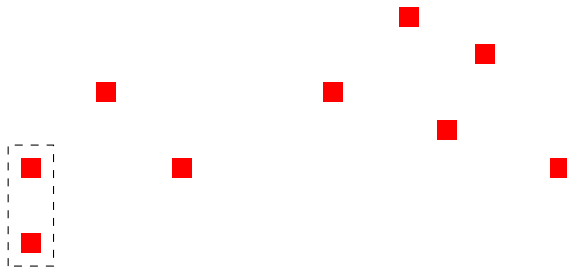
Le groupement ainsi constitué est considéré comme un pseudo-élément positionné en son barycentre.



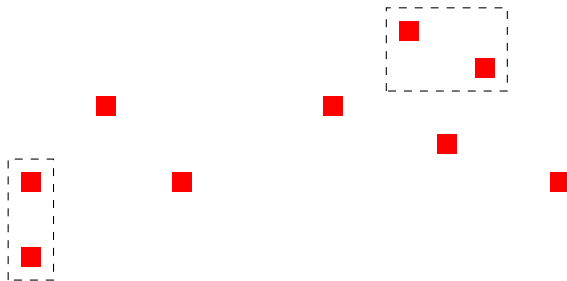
# Exemple



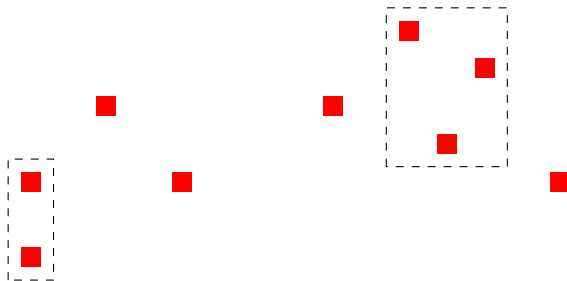
# Exemple



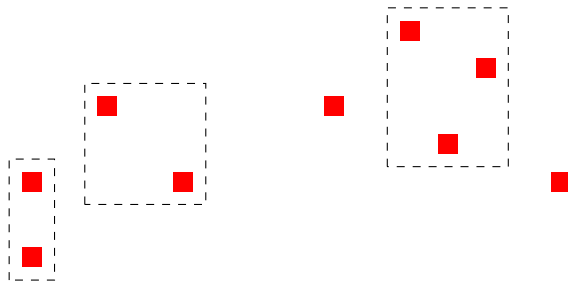
# Exemple



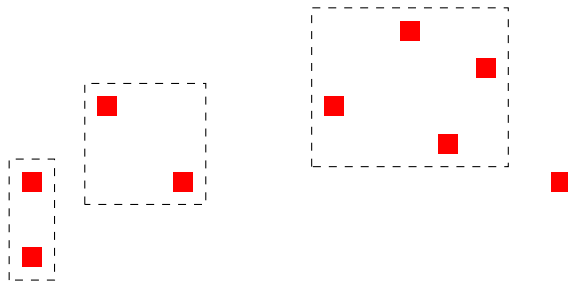
# Exemple



# Exemple



# Exemple



# À vous de jouer !

Programmez un algorithme de classification hiérarchique ascendante. Testez-le sur l'exemple précédent (à partir de la table de distances).

## 1 Motivation

- Exemples
- L'apprentissage : idée générale
- Valeur d'un résultat

## 2 Visualisation

- Aux origines du problème
- Réduction dimensionnelle
- Approche non linéaire

## 3 Décision

- Introduction
- Ford-Fulkerson / Edmonds
- Construction du graphe
- MATCHING vs CLUSTERING (cliques)
- La classification hiérarchique ascendante

## 4 Qualité d'un modèle

- Différentes sources d'erreur
- Complexité



# Dissimilarité

Choix d'une mesure de l'écart entre prédiction et observation.

# Dissimilarité

Choix d'une mesure de l'écart entre prédiction et observation.

Minimiser :

$$d(\tilde{f}(x), y)$$

# Exemples de dissimilarités

$$d(\tilde{f}(x), y) = \|\tilde{f}(x) - y\|_2^2 = \sum (\tilde{f}(x)_i - y_i)^2$$

$$d(\tilde{f}(x), y) = |\tilde{f}(x) - y| = \sum |\tilde{f}(x)_i - y_i|$$

$$d(\tilde{f}(x), y) = |\{i, \tilde{f}(x)_i \neq y_i\}|$$

$$d(\tilde{f}(x), y) = \sum w_i |\tilde{f}(x)_i - y_i|$$

$$d(\tilde{f}(x), y) = \sum \phi_i(\tilde{f}(x)_i, y_i)$$

## Exercice

### Exercice

*Trouvez des contextes pour lesquels des mesures de dissimilarités différentes sont appropriées.*

# Rappel des Hypothèses

Observations :

- Variable empirique cible  $\tilde{Y}$
- Variables empiriques explicatives  $\tilde{X}$

# Rappel des Hypothèses

Observations :

- Variable empirique cible  $\tilde{Y}$
- Variables empiriques explicatives  $\tilde{X}$

Hypothèses :

- $\tilde{X}$  est un ensemble d'observations lié à un processus aléatoire  $X$
- $\tilde{Y}$  est un ensemble d'observations lié à un processus aléatoire  $Y$
- il existe une relation  $Y = f(X)$

# Rappel des Hypothèses

Observations :

- Variable empirique cible  $\tilde{Y}$
- Variables empiriques explicatives  $\tilde{X}$

Hypothèses :

- $\tilde{X}$  est un ensemble d'observations lié à un processus aléatoire  $X$
- $\tilde{Y}$  est un ensemble d'observations lié à un processus aléatoire  $Y$
- il existe une relation  $Y = f(X)$

Objectifs :

- Produire une fonction  $\tilde{f}$  à partir de  $\tilde{X}$  et  $\tilde{Y}$
- Telle que  $\tilde{f}$  soit une approximation fiable de  $f$
- On pourra ainsi prédire  $\tilde{Y}' = \tilde{f}(\tilde{X}')$  sur un nouvel échantillon

# Erreur du modèle

La bonne mesure serait de minimiser :

$$D(\tilde{f}) = \mathbb{E}(d(\tilde{f}(x), y))$$



# Erreur du modèle

La bonne mesure serait de minimiser :

$$D(\tilde{f}) = \mathbb{E}(d(\tilde{f}(x), y))$$

Mais comme on ne connaît pas la loi de  $(X, Y)$  c'est impossible.

# Erreur moyenne empirique

On dispose d'un échantillon de test  $\tau = (X_j, Y_j)_{j \leq n}$ .

Minimiser :

$$\tilde{D}(\tilde{f}, \tau) = \frac{1}{n} \sum_{j \leq n} d(\tilde{f}(x_j), y_j)$$

# Erreur moyenne empirique

On dispose d'un échantillon de test  $\tau = (X_j, Y_j)_{j \leq n}$ .

Minimiser :

$$\tilde{D}(\tilde{f}, \tau) = \frac{1}{n} \sum_{j \leq n} d(\tilde{f}(x_j), y_j)$$

Ne pas confondre cette somme sur les données avec la somme sur les variables !

Ne pas confondre cette moyenne empirique avec la moyenne

# Convergence

D'après la loi des grands nombres, si les observations de test sont indépendantes, la moyenne empirique converge vers l'erreur du modèle.

# Pertinence du test

On cherche à évaluer la probabilité que l'écart entre les deux mesures soit faible.

$$P\left(\tilde{D}(\tilde{f}, \tau) - D(\tilde{f}) > \epsilon\right) < 1 - \rho$$

## 1 Motivation

- Exemples
- L'apprentissage : idée générale
- Valeur d'un résultat

## 2 Visualisation

- Aux origines du problème
- Réduction dimensionnelle
- Approche non linéaire

## 3 Décision

- Introduction
- Ford-Fulkerson / Edmonds
- Construction du graphe
- MATCHING vs CLUSTERING (cliques)
- La classification hiérarchique ascendante

## 4 Qualité d'un modèle

- Différentes sources d'erreur
- Complexité

# Exponentielle rapide

## Exercice

*Implémentez une fonction exponentielle. Combien de multiplications effectue-t-elle pour calculer  $3^{130}$  ?*

# Exponentielle rapide

```
def fastexp(a,b):  
    if b == 0:  
        return 1  
    if b%2 == 0:  
        return fastexp(a,b//2)**2  
    else:  
        return a*fastexp(a,b//2)**2
```



# Exemples d'algorithmes

Exponentielle rapide :  $O(\log n)$

Tri rapide, exponentielle naïve :  $O(n)$

Tri par insertion :  $O(n^2)$

Multiplication matricielle naïve :  $O(n^3)$

Énumération des sous-ensembles :  $O(2^n)$

Voyageur de commerce, Coloration :  $O(2^n)$

Énumération des permutations :  $O(n!)$

Taille	$n \log n$	$n^3$	$2^n$
$n = 20$	60	8000	1048576
$n = 50$	196	125000	1125899907000000
$n = 100$	461	1000000	12676506000000000000000000000000

## retour aux classifications

### Exercice

*Quelle est la complexité d'un clustering en force brute ? Et celle d'une classification hiérarchique ascendante ?*