CM 2
Méthodes de Simulation Informatique

Amaya Nogales Gómez
amaya.nogales-gomez@univ-cotedazur.fr

Licence 3 Informatique
Université Côte d'Azur

11 février 2022

Plan du cours

1. Introduction
   - Préliminaires
   - Python: numpy, pandas
2. Base de données
   - Generation des données synthétiques
   - Base de données reels
3. Analyse descriptive
4. Techniques d'apprentissage supervisée
   - Support Vector Machines
   - Régression logistique
5. Contrôle de connaissances
6. Techniques de validation
7. Elements de la méthodologie scientifique
8. LaTeX
   - Écriture de textes scientifiques
   - Beamer: présentations et posters scientifiques

Liste de fonctions utiles de NumPy

```
>>> import numpy as np
ModuleNotFoundError: No module named 'numpy'
>>> pip install numpy
```

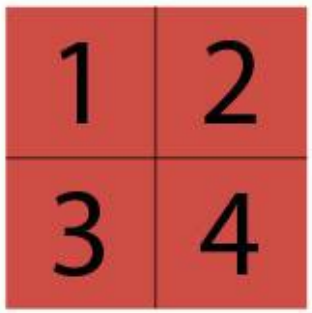
- Creation de tableaux: arange, array, copy, empty, eye, identity, linspace, logspace, mgrid, ogrid, ones, zeros
- Manipulations: concatenate, diagonal, repeat, reshape, transpose
- Questions: in, all, any, nonzero, where
- Ordernnation: argmax, argmin,max, min, sort
- Opérations: cumprod, cumsum, prod, real, sum
- Statistique: cov, mean, std, var
- Algèbre linéaire: dot, outer, vdot

Tableaux NumPy (*array*)

- Objet central du package NumPy.
- Un tableau NumPy unidimensionnel: un vecteur.
- Bidimensionnel: une matrice.
- Tridimensionnel: un tenseur (ensemble de matrices).

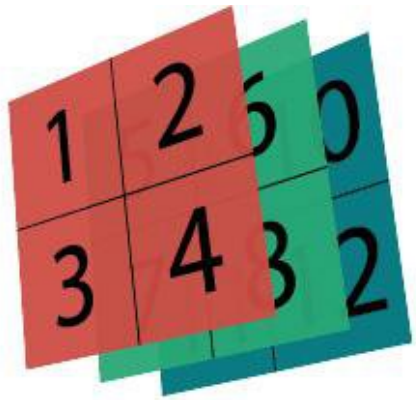
Vector

```
np.array([1, 2])
```



Matrix

```
np.array([[1, 2], [3, 4]])
```



3D Matrix

```
np.array([[[1, 2], [3, 4]],
          [[5, 6], [7, 8]],
          [[9, 10], [11, 12]]])
```

Image source: `https://www.learndatasci.com/`

Mathématiques du tableau

- Les opérations mathématiques sont appliquées élément par élément.
- Les tableaux doivent avoir la même taille:

```
>>> x = np.array([-1,7,-3], float)
>>> y = np.array([1,2,0], float)
>>> x+y
array([ 0.,  9., -3.])
>>> x-y
array([-2.,  5., -3.])
```

Opérations sur les tableaux

```
>>> a=np.array([10,20,30],float)
a.sum()
60.0
>>> a.prod()
6000.0
>>> np.sum(a)
60.0
>>> a.mean()
66.66666666666667
```

Nombres aléatoires: germe ou graine

- Une partie importante de toute **simulation** est la possibilité de tirer des nombres aléatoires.
- On utilise de routines de génération de nombres pseudo-aléatoires intégrées de NumPy dans le sous-module *random*.
- Ils sont générés à partir d'un nombre appelé germe qu'est une valeur entière.
- Tout programme qui démarre avec le même germe générera exactement la même séquence de nombres aléatoires à chaque exécution.
- Il n'est pas nécessaire de spécifier le germe.

```
>>> np.random.seed(293423)
```

Génération de nombres aléatoires

Nombres aléatoires d'une distribution uniforme.

```
>>> np.random.rand(2,3)
array([[ 0.50431753, 0.48272463, 0.45811345],
[ 0.18209476, 0.48631022, 0.49590404]])
>>> np.random.random()
0.70110427435769551
>>> np.random.randint(5, 10)
9
```

Series et DataFrames

- Principaux composants: *Series* et *DataFrame*.
- Series est essentiellement une colonne.
- DataFrame est un tableau multidimensionnel composé d'une collection de Series.

Series **Series** **DataFrame**

	apples
0	3
1	2
2	0
3	1

+

	oranges
0	0
1	3
2	7
3	2

=

	apples	oranges
0	3	0
1	2	3
2	0	7
3	1	2

Image source: `https://www.learndatasci.com/`

Créer des DataFrames à partir de zéro

L'indice (*index*) du DataFrame *covid* par defaut sont des nombres 0-3.
On peut créer nôtre propre *index*.

```
covid = pd.DataFrame(data, index=['Licence 1', 'Licence 2',
                                  'Licence 3', 'Master 1'])

covid
```

	cas positifs	cas contact
Licence 1	9	30
Licence 2	2	13
Licence 3	0	17
Master 1	1	5

Opérations avec DataFrames

```
compas_df = pd.read_csv("propublica.csv", index_col=0)
```

```
compas_df.head()
```

	Age	C_charge_degree	Race	Age_cat	Score_text	sex	priors_count	days_screening_arrest (before)	decile_sc
1	69	F	Other	Greater than 45	Low	Male	0	-1	
2	34	F	African-American	25 - 45	Low	Male	0	-1	
3	24	F	African-American	Less than 25	Low	Male	4	-1	
4	44	M	Other	25 - 45	Low	Male	0	0	
5	41	F	Caucasian	25 - 45	Medium	Male	14	-1	

Sélection de colonnes

```
age_col = compas_df['age']

type(age_col)
```

```
pandas.core.series.Series
```

Sélections conditionnelles

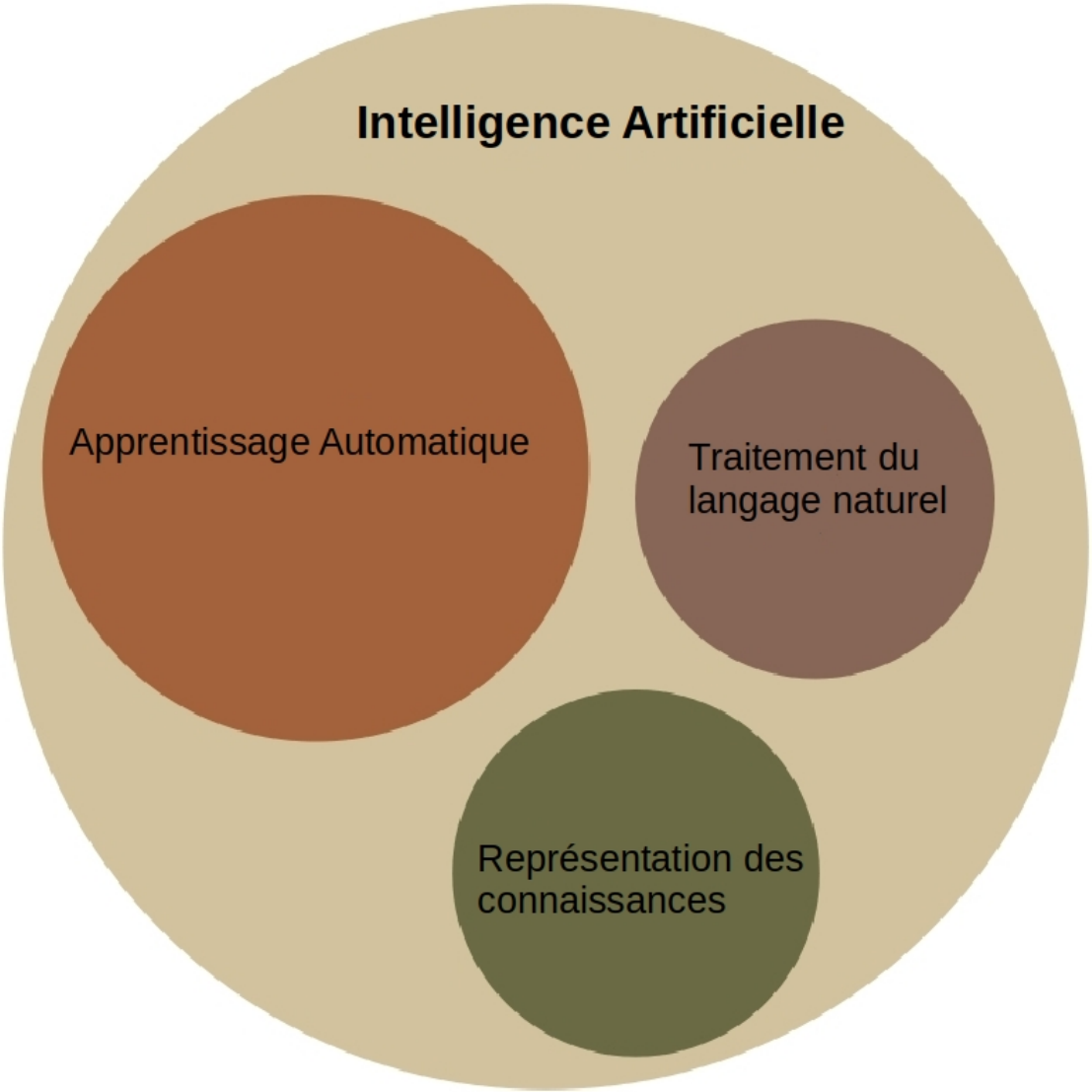
```
condition = (compas_df['race'] == "African-American")

condition.head()
```

```
1    False
2     True
3     True
4    False
5    False
Name: race, dtype: bool
```

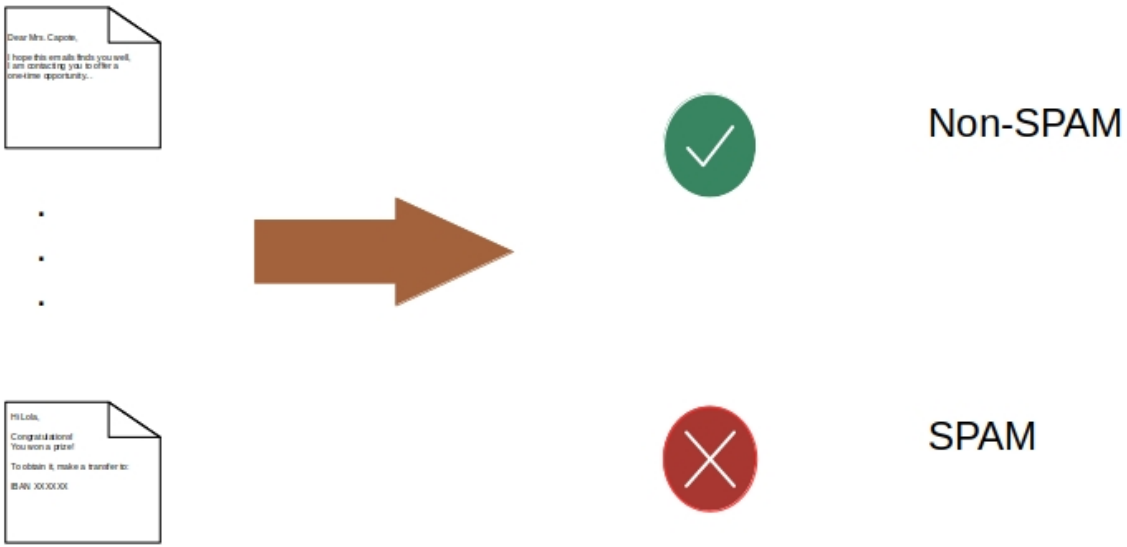
```
compas_df["decision"] = compas_df["decile_score"].apply(lambda x: 'coupable'
                                                         if x >= 5.0 else 'non-coupable')

subset=compas_df[['decile_score','decision']]
subset.head(3)
```

	decile_score	decision
1	1.000000	non-coupable
2	3.000000	non-coupable
3	4.418465	non-coupable

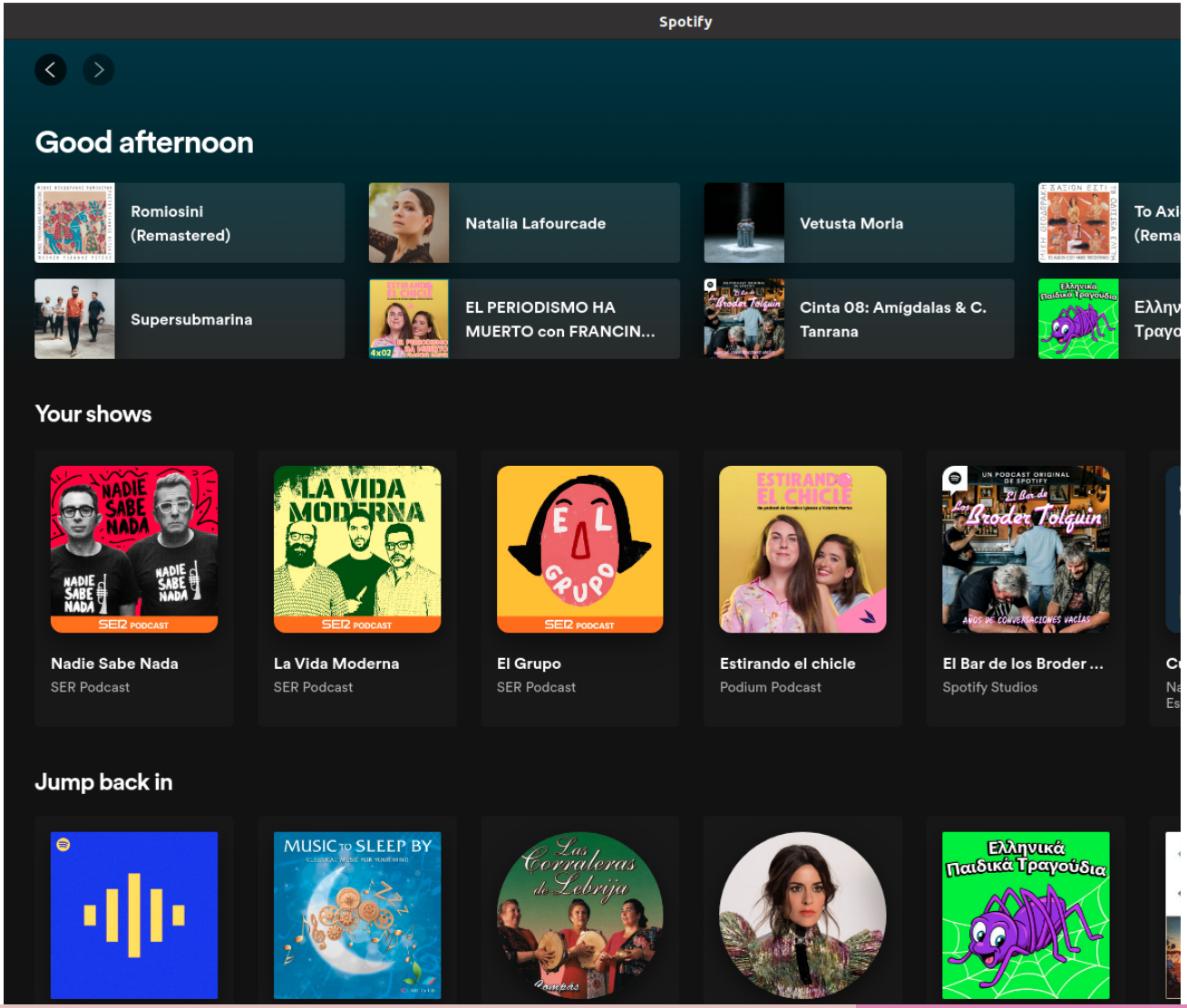
Qu'est-ce que c'est l'apprentissage automatique?



Exemple: filtrage anti-spam

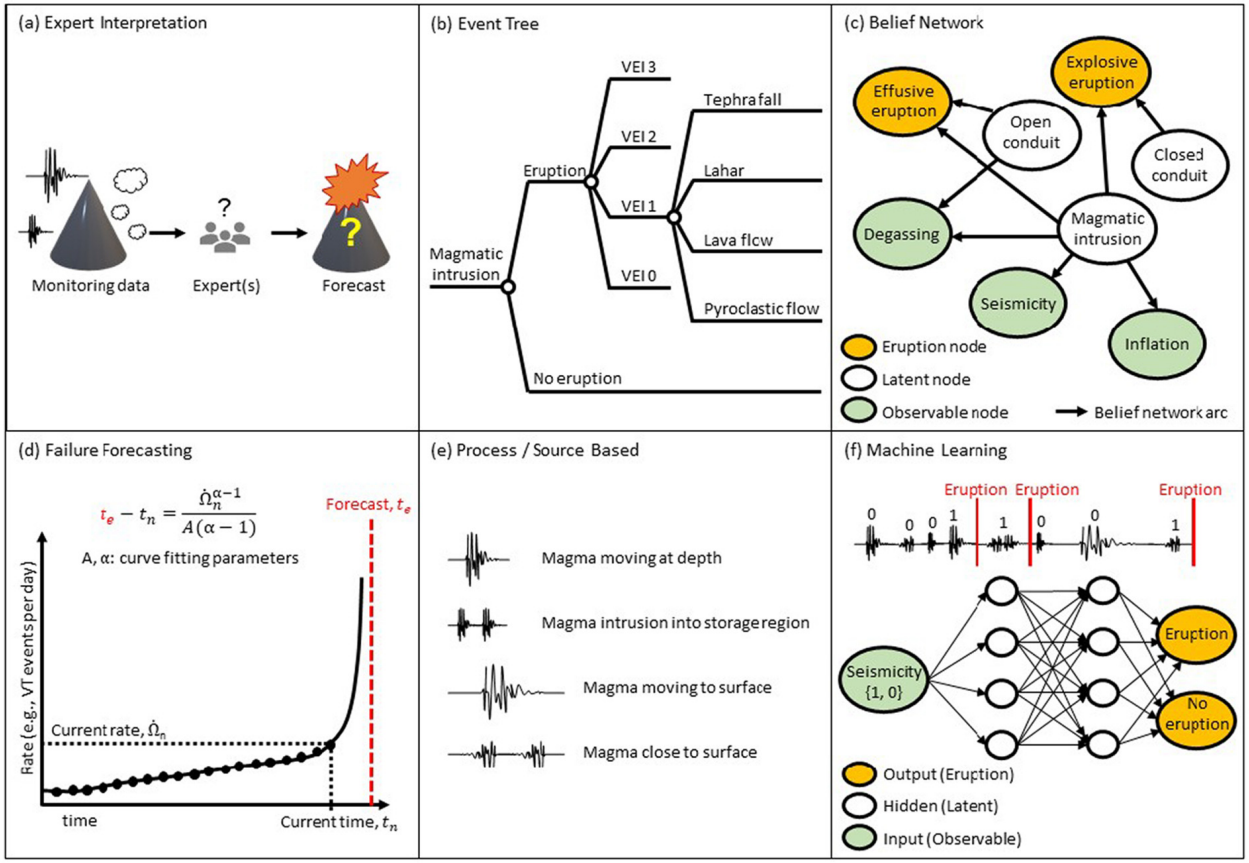


Exemple: systèmes de recommandation



Exemple: clustering

Exemple: prévision



M. G. Whitehead, M. S. Bebbington, Method selection in short-term eruption forecasting, *Journal of Volcanology and Geothermal Research*.

Exemples à l'Université Côte d'Azur

- Santé
  - Diagnostic médical & prévention.
- Industrie
  - Systèmes de recommandation: musique, vidéo.
  - Stockage d'images dans l'ADN synthétique.
- Multimédia
  - Détection du langage dans des débats politiques.
  - Analyse culturelle, de paroles et d'audio de la musique.

Intelligence artificielle

La science et l'ingénierie de la fabrication de machines intelligentes, en particulier de systèmes informatiques, qui reproduit l'intelligence humaine par l'apprentissage, le raisonnement et l'adaptation. [McCarthy89]

Apprentissage Automatique

Un algorithme qui améliore sa mesure de performance P à une certaine classe de tâches T avec de l'expérience E. [Mitchell90]
Domaine d'étude qui donne aux ordinateurs la capacité d'apprendre sans être explicitement programmés. [Samuel59]

A. Samuel, "Some Studies in Machine Learning Using the Game of Checkers". IBM Journal of Research and Development. 3 (3), 1959.

T. Mitchell, B. Buchanan, G. DeJong, T. Dietterich, P. Rosenbloom, A. Waibel, Machine learning, Annual review of computer

Les origines de l'IA

1956 Projet de recherche d'été de Dartmouth sur l'intelligence artificielle

A Proposal for the

DARTMOUTH SUMMER RESEARCH PROJECT ON ARTIFICIAL INTELLIGENCE

June 17 - Aug. 16

We propose that a 2 month, 10 man study of artificial intelligence be carried out during the summer of 1956 at Dartmouth College in Hanover, New Hampshire. The study is to proceed on the basis of the conjecture that every aspect of learning or any other feature of intelligence can in principle be so precisely described that a machine can be made to simulate it. An attempt will be made to find how to make machines use language, form abstractions and concepts, solve kinds of problems now reserved for humans, and improve themselves. We think that a significant advance can be made in one or more of these problems if a carefully selected group of scientists work on it together for a summer.

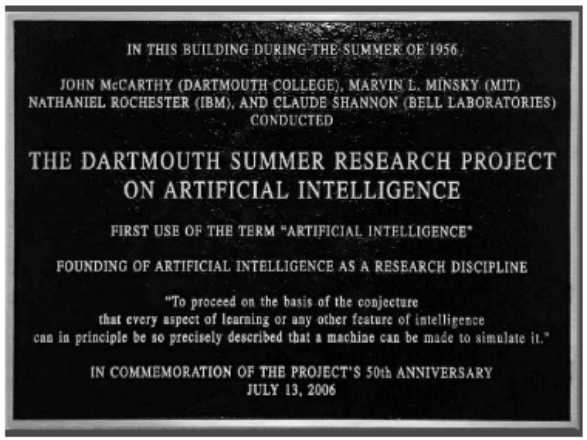
The following are some aspects of the artificial intelligence problem:

1) Automatic Computers

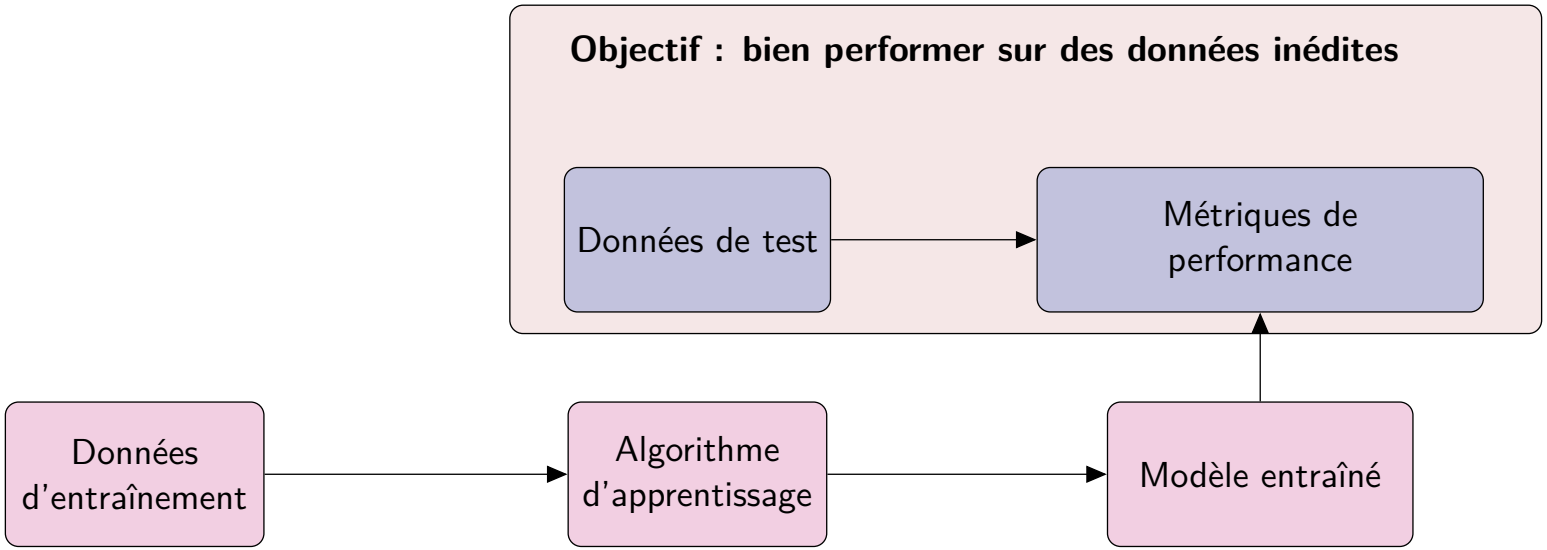
If a machine can do a job, then an automatic calculator can be programmed to simulate the machine. The speeds and memory capacities of present computers may be insufficient to simulate many of the higher functions of the human brain, but the major obstacle is not lack of machine capacity, but our inability to write programs taking full advantage of what we have.

2) How Can a Computer be Programmed to Use a Language

It may be speculated that a large part of human thought con-



Approche d'apprentissage automatique de base



Les défis de l'apprentissage automatique

- Quel type de données utiliser?
- Quelle quantité de données est suffisante?
- Comment le représenter?
- Quel algorithme utiliser?
- Comment choisir le meilleur modèle?
- Garanties de performance
- Explicabilité
- Interprétabilité

Comment modéliser un problème comme un problème de Apprentissage Automatique?

Algorithmes d'apprentissage automatique

- Apprentissage supervisé
  - Les données d'entraînement incluent les résultats souhaités
  - Ensemble de données composé d'exemples étiquetés
- Apprentissage non supervisé
  - Les données d'entraînement n'incluent pas les résultats souhaités
  - Trouver une structure dans certains exemples (pas d'étiquettes!)
- Apprentissage par renforcement
  - Récompenses de la séquence d'actions
  - Prise de décision séquentielle basée sur la rétroaction

Notation

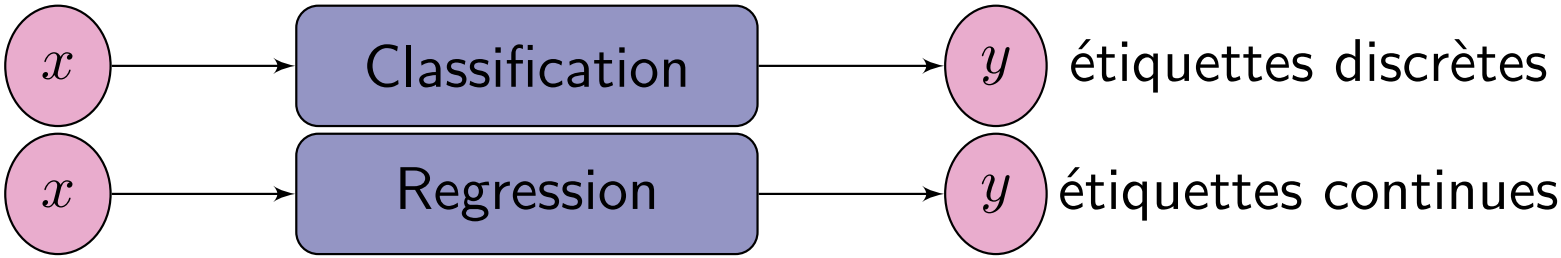
- Base de données (X, Y)
- X: variables
- $Y \in \{-1, +1\}$ étiquette, classe du groupe
- $\hat{Y} = f(X)$: prédiction

Abus de notation

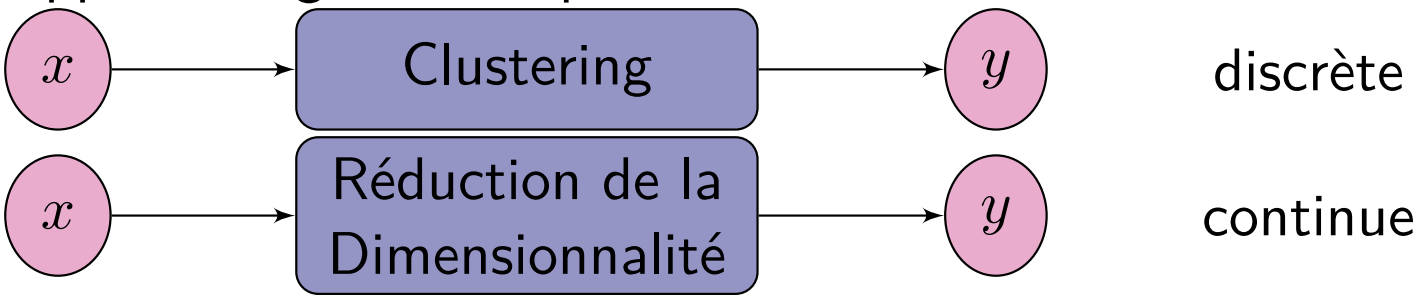
- variable $\equiv$ attribut $\equiv$ caractéristique
- objet $\equiv$ individu $\equiv$ observation

Apprentissage supervisé

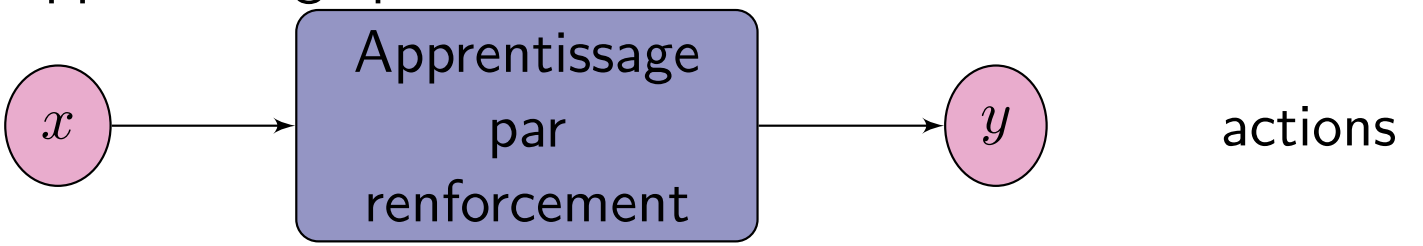
Apprentissage Supervisé



Apprentissage Non Supervisé



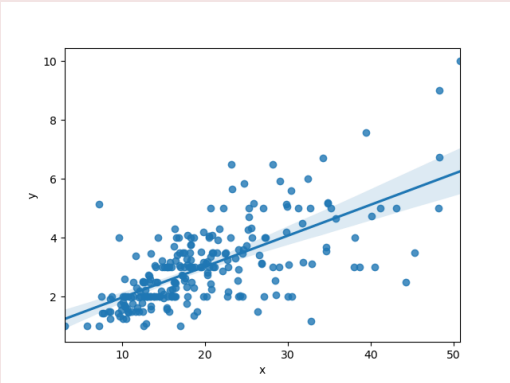
Apprentissage par renforcement



Apprentissage supervisé: régression et classification

Régression

Sortie : fonction continue.

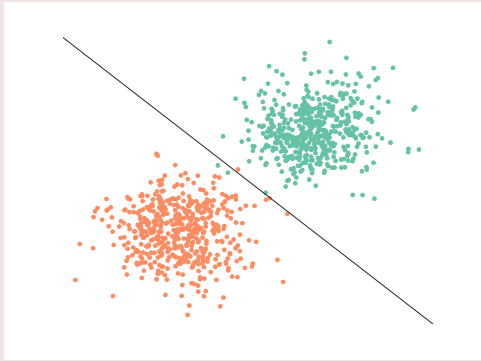


Exemples:

- Prévision (météo)
- Taille des animaux
- Actions

Classification

Sortie: règle de séparation.

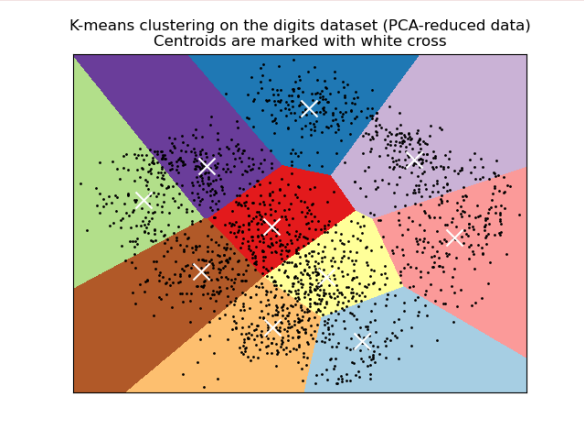


Examples:

- Rembourser un prêt
- Acceptation d'entrée à l'université
- Classement des images

Apprentissage non supervisé

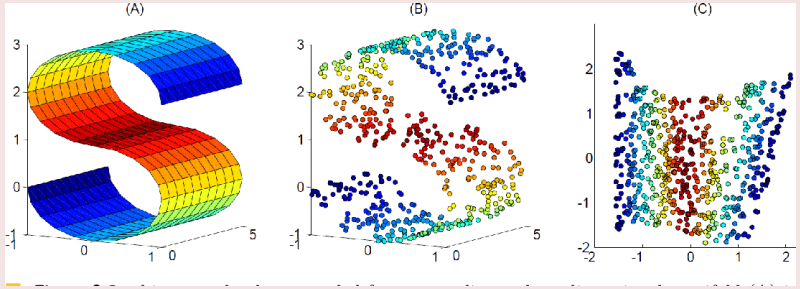
Clustering



Exemples:

- Systèmes de recommandation
- Analyse des réseaux sociaux

Réduction de la dimensionnalité



Exemples:

- Visualisation de données
- Stockage de données
- Complexité de calcul

Clustering

Trouvez des sous-types ou des groupes qui ne sont pas définis a priori en fonction des mesures.

apprentissage non supervisé

Classification

Utilisez des étiquettes de groupe a priori dans l'analyse pour attribuer de nouvelles observations à un groupe ou à une classe en particulier.

apprentissage supervisé

Objectifs généraux du clustering

1. Les observations au sein d'un cluster sont similaires

   propriété de compacité

Objectifs généraux du clustering

1. Les observations au sein d'un cluster sont similaires

   propriété de compacité

2. Les observations dans différents clusters ne sont pas similaires

   propriété de proximité

Objectifs généraux du clustering

1. Les observations au sein d'un cluster sont similaires

   propriété de compacité

2. Les observations dans différents clusters ne sont pas similaires

   propriété de proximité

Objectif : obtenir des clusters compacts et bien séparés

Cycle de l'apprentissage automatique

Préparation des données

- Selon la modalité de données et tâche, différents types de prétraitement peuvent être appliqués à la base de données avant de l'utiliser.
- Les bases de données sont généralement divisés en données d'entraînement utilisées lors du développement du modèle et en données de test utilisé lors de l'évaluation du modèle.
- Une partie des données d'entraînement peut être mise de côté en tant que données de validation.

Cycle de l'apprentissage automatique

Développement du modèle

- Un modèle est ensuite construit à l'aide de données d'entraînement.
- Un certain nombre de différents types de modèles, hyperparamètres et des méthodes d'optimisation peuvent être testées à ce stade; ces différentes configurations sont comparées en fonction de leurs performances sur les données de test, et le meilleur choisi.
- Les mesures de performance utilisées pour tels comparaisons sont choisies en fonction des caractéristiques de la tâche et des données; les choix le plus courants sont la précision, les taux de faux positifs ou de vrais positifs (TFP/TVP).

Critères de classification communs

prédiction $\hat{y}$	étiquette y	Critère
$+1$	$+1$	Taux de vrais positifs
-1	$+1$	Taux de faux négatifs
$+1$	-1	Taux de faux positifs
-1	-1	Taux de vrais négatifs

Cycle de l'apprentissage automatique

Évaluation du modèle

- Une fois le modèle final et les hyperparamètres sont choisis et l'optimisation du modèle est terminée, la performance finale du modèle sur les **données de validation** est rapportée.
- Il est important que les données de validation ne soient pas utilisées avant cette étape pour s'assurer que les performances du modèle sont une représentation fidèle des données inédites.
- Comme dans le développement du modèles, il est important de choisir des mesures de performance bien adaptées.

Cycle de l'apprentissage automatique

Post-traitement du modèle

- Une fois qu'un modèle est prêt à être utilisé, differents étapes de post-traitement peuvent être appliqués.
- Par exemple, si la sortie d'un modèle effectuant la classification binaire est une probabilité, mais la sortie souhaitée est une réponse binaire, il reste un choix de quel seuil utiliser pour arrondir la probabilité à un classification.

Cycle de l'apprentissage automatique

Déploiement du modèle

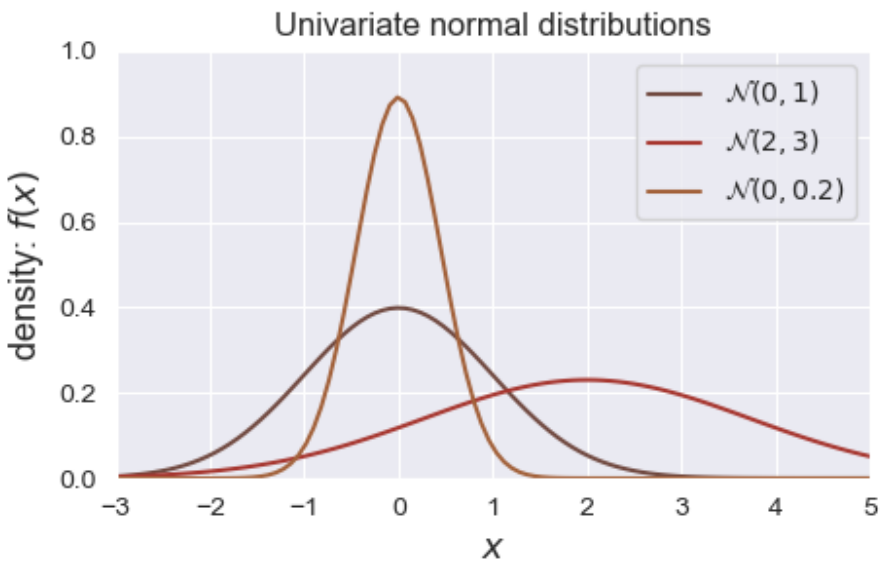
- Pour un apprentissage automatique dans le monde réel, de nombreuses étapes surviennent lorsqu'un système est effectivement déployé.
- Par exemple, un modèle peut avoir besoin d'être changé en fonction des exigences d'équité, ou il peut y avoir une rétroaction en temps réel qui devrait être réintégrée dans le modèle.

Distribution gaussienne

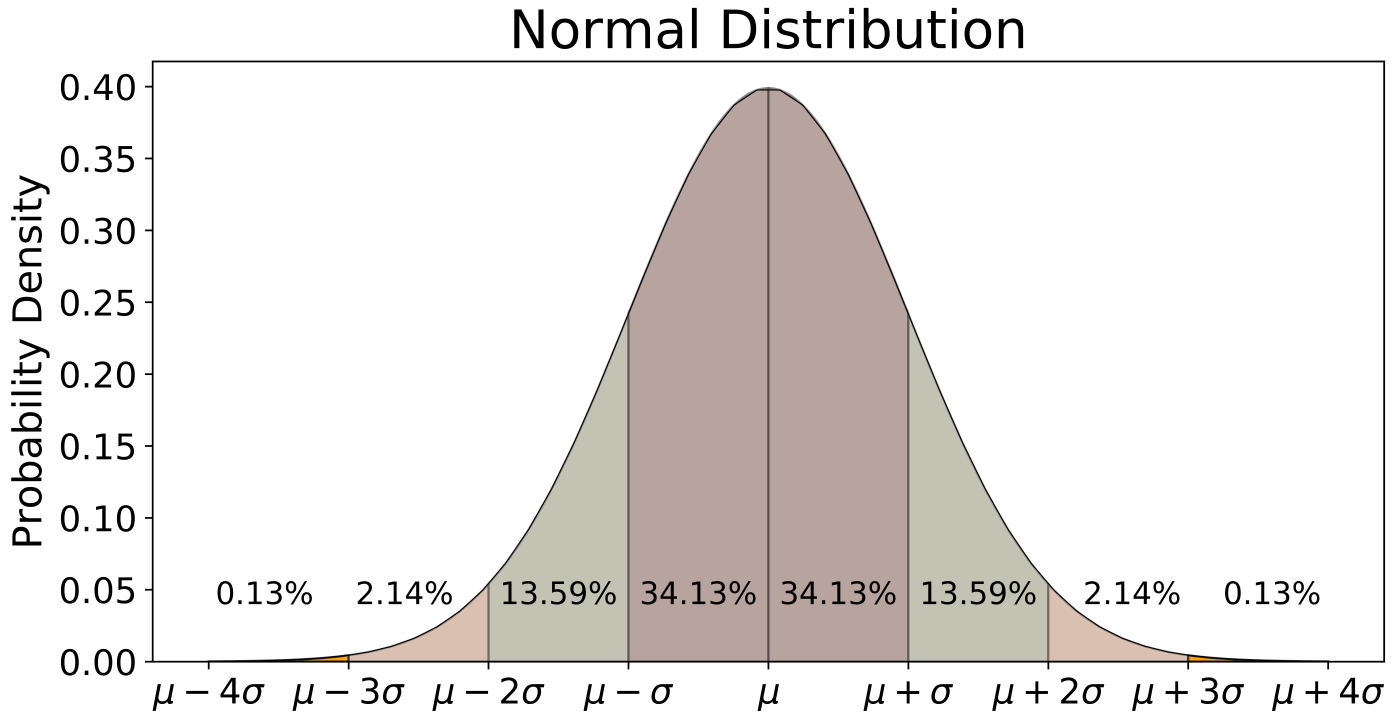
Une distribution normale (ou gaussienne) est un type de distribution de probabilité continue pour une variable aléatoire à valeur réelle. La forme générale de sa fonction de densité de probabilité est

$$X \sim N(\mu, \sigma^2)$$

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$



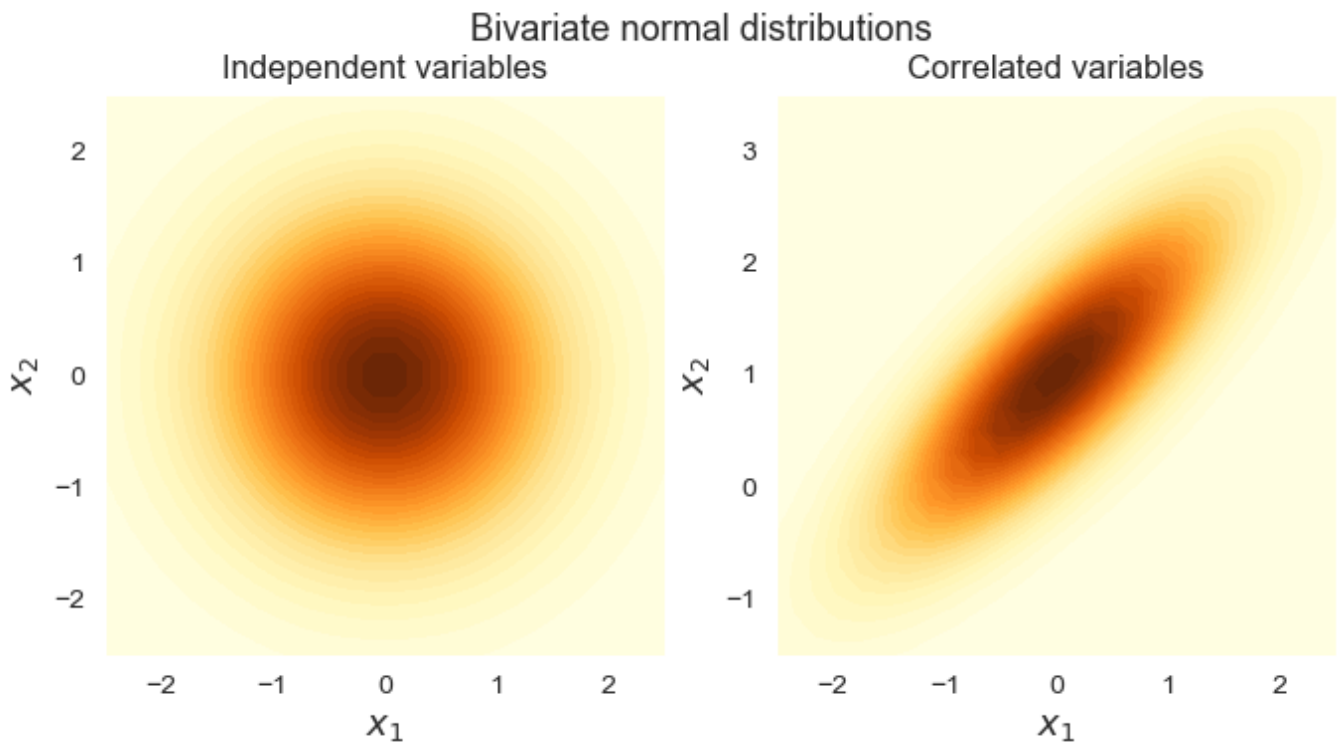
$$P(\mu - n\sigma \leq X \leq \mu + n\sigma) = F(\mu + n\sigma) - F(\mu - n\sigma)$$
$$F(x) = \int_{-\infty}^{x} f(x)dx$$



Distribution normale multidimensionnelle

La distribution normale multidimensionnelle est une généralisation multidimensionnelle de la distribution normale unidimensionnelle.
Il représente la distribution d'une variable aléatoire multivariée composée de plusieurs variables aléatoires pouvant être corrélées entre elles.

$$\begin{aligned} X &\sim N(\mu, \Sigma) \\ f(x) &= \frac{1}{2\pi\sqrt{|\Sigma|}} e^{-\frac{1}{2}\left((x-\mu)^\top \Sigma^{-1}(x-\mu)\right)^2} \end{aligned}$$



$$\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \qquad \mu = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$\Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad \Sigma = \begin{bmatrix} 1 & 0.8 \\ 0.8 & 1 \end{bmatrix}$$

Exemples réels de la distribution normale?

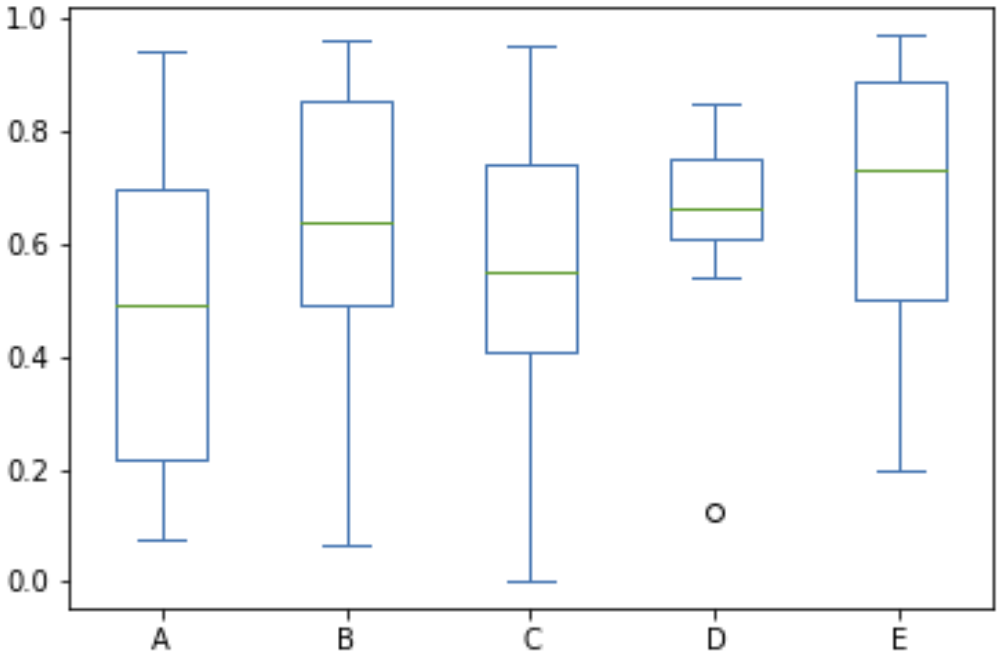
- Tailles humaines (les personnes du même sexe et du même groupe d'âge se regroupent généralement autour de la moyenne avec une distribution normale)
- Scores QI ($\mu = 100$, $\sigma = 15$)
- Notes des élèves d'une classe ($\mu = 60$, $\sigma = 20$)
- Mesure de poids ($\mu = 80$ kg, $\sigma = 10$)
- Mesure de la tension artérielle ($\mu = 120/80$, $\sigma = 20$)
- Hauteur des arbres (mesure en mètres; $\mu = 40$ m, $\sigma = 20$)

Données aberrantes: *outliers*

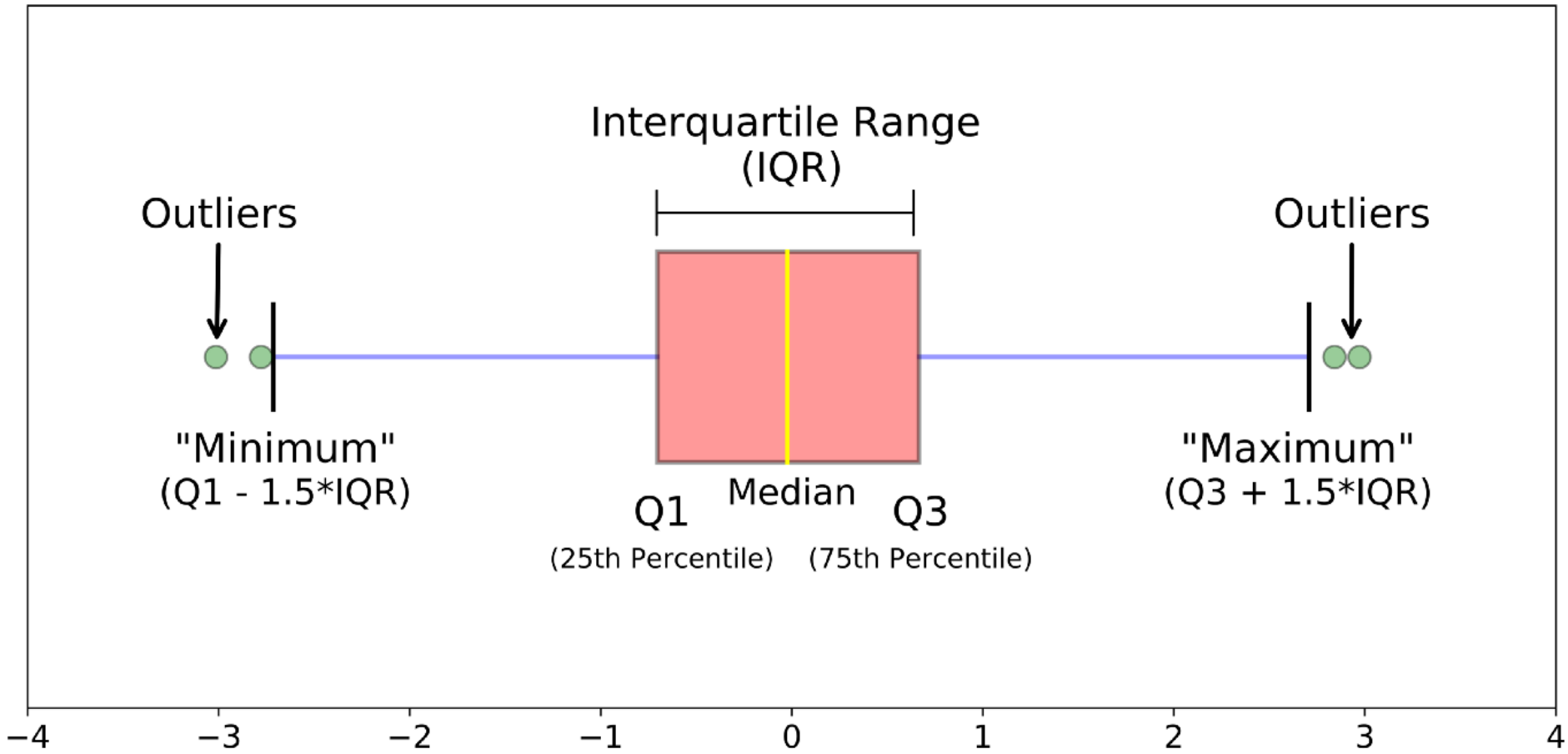
En statistique, une donnée aberrante (en anglais outlier) est une valeur ou une observation qui est "distante" des autres observations effectuées sur le même phénomène, c'est-à-dire qu'elle contraste grandement avec les valeurs "normalement" mesurées.
Une donnée aberrante peut être due à la variabilité inhérente au phénomène observé, ou indiquer une erreur expérimentale. Dans ce dernier cas, elles sont parfois écartées.

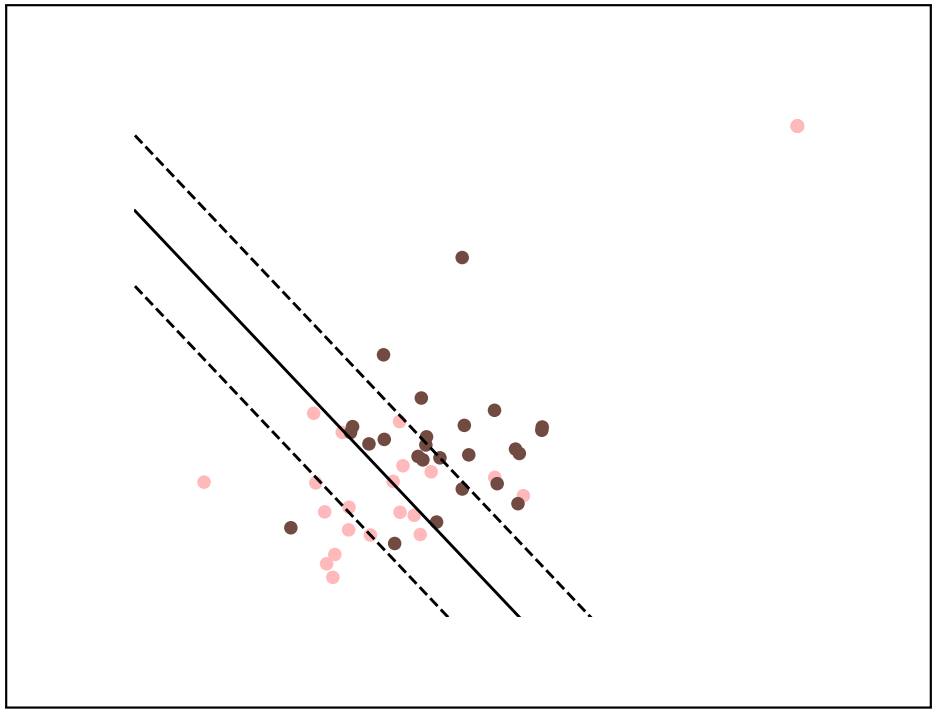
Données aberrantes: *outliers*

```
df = pd.DataFrame(np.random.rand(10, 5), columns=["A", "B", "C", "D", "E"])

df.plot.box();
```



Données aberrantes: *outliers*





- Classe positive (rose) $\sim N(\begin{bmatrix}0\\0\end{bmatrix}, \begin{bmatrix}1 & 0\\0 & 1\end{bmatrix})$
- Classe negative (marron) $\sim N(\begin{bmatrix}2/\sqrt{d}\\2/\sqrt{d}\end{bmatrix}, \begin{bmatrix}1 & 0\\0 & 1\end{bmatrix})$
- Données aberrantes (rose) $\sim N(\begin{bmatrix}10/\sqrt{d}\\10/\sqrt{d}\end{bmatrix}, \begin{bmatrix}0.001 & 0\\0 & 0.001\end{bmatrix})$

Base de données reels: *German*

- Base de données de évaluation de crédit: les bons clients définissent la classe positive $(y = +1)$ et les mauvais clients définissent la classe négative $(y = -1)$.
- `https://archive.ics.uci.edu/ml/datasets/statlog+(german+credit+data)`
- Taille: $n = 1000$
- Dimension: $d = 58$, représenté par 20 variables.
  - 7 variables continues
  - 2 variables binaires
  - 11 variables catégorielles

Base de données German

Variable	Type
Duration in month	continuous
Credit amount	continuous
Installment rate	continuous
Present residence since	continuous
Age in years	continuous
#credits at this bank	continuous
#people liable	continuous
Telephone	binary
foreign worker	binary

Base de données German: variables catégorielles

variable	#catégories
Status of existing checking account	5
Credit history	5
Purpose	10
Savings account	5
Present employment since	5
Age_cat	3
Other debtors / guarantors	3
Property	4
Other installment plans	3
Housing	3
Job	4

Comment traitons-nous les caractéristiques catégorielles?

Race

La caractéristique catégorielle race comprend les catégories suivantes:
African-American, Caucasian, Hispanic, Asian, Native American, Other.

- Nous binarisons la caractéristique catégorielle *race* en 6 caractéristiques binaires.

i	Race	African-American	Caucasian	Hispanic	Asian	Native American	Other
1	Asian	0	0	0	1	0	0
2	Other	0	0	0	0	0	1
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

Base de données German: categorie d'âge

Age_cat: categorie d'âge

- $less\_than\_25$: âge ≤ 25
- $between\_25\_45$: $25 <$ âge < 25
- $greater\_than\_45$: âge ≥ 45

Variables categorielles

```
df = (
    pd.read_csv("german.csv")
    #ON binarize la variable categorielle age_cat avec 3 categories
    #age_cat
    .assign(less_than_25=lambda x:x['age_cat'].replace({'Greater than 45':0, '25 - 45':0, 'Less than 25':1}))
    .assign(between_25_45=lambda x:x['age_cat'].replace({'Greater than 45':0, '25 - 45':1, 'Less than 25':0}))
    .assign(greater_than_25=lambda x:x['age_cat'].replace({'Greater than 45':1, '25 - 45':0, 'Less than 25':0}))

)
df.head(3)
```

core	...	two_year_recid	c_jail_in	c_jail_out	juv_fel_count	juv_misd_count	juv_other_count	is_violent_recid	less_than_25	between_25_45	greater_than_25
1	...	0	2013-08-13 06:03:42	2013-08-14 05:41:20	0	0	0	0	0	0	1
3	...	1	2013-01-26 03:45:27	2013-02-05 05:36:53	0	0	0	1	0	1	0
4	...	1	2013-04-13 04:58:34	2013-04-14 07:02:04	0	0	1	0	1	0	0

Bibliographie recommandée

- UCI Machine Learning Repository `https://archive.ics.uci.edu`
- `http://cs229.stanford.edu/section/gaussians.pdf`
- Kaggle repository `https://www.kaggle.com`