

Big Data Analytics

Jour 1 — Introduction au Machine Learning

François-Marie Giraud



<https://www.orsys.fr/>

Big Data Analytics

Présentation & coordonnées

Nom François-Marie Giraud

Courriel giraud.francois@gmail.com

Activité Consultant/Formateur indépendant

Spécialité Intelligence Artificielle

Parcours Master Intelligence Artificielle et Décision (Paris 6)

Description

Cette formation présente les **fondamentaux** de la **Modélisation Statistique** à travers des travaux pratiques.

Prérequis

- Connaissances de base en statistiques et algèbre
- Connaissances de base en python
- Avoir un **compte Google** afin de pouvoir faire les TPs dans [Google Colaboratory](#)

Objectifs pédagogiques

- Comprendre les principes de la modélisation statistique
- Comprendre les différents types de régressions
- Évaluer les performances d'un algorithme prédictif
- Sélectionner et classer des données dans de grands volumes de données
- Se familiariser avec les bibliothèques scientifiques python (NumPy, seaborn, scikit-learn, ...)

En bref, se familiariser avec les concepts et techniques de bases du
« Machine Learning »

Programme

- Introduction à la modélisation
- Évaluation de modèles prédictifs
- Les algorithmes supervisés/non-supervisés
- Projection de données par composantes
- Analyse de données textuelles

Je vous ferai parvenir les ressources informatiques utilisées à chaque début de cours. Elles sont aussi accessibles via [My Orsys](#).

Emploi du temps

- 4 jours de 9h à 12h30 et de 14h à 17h30
- Le dernier jour, on termine à 15h30. Donc à 15h00 on commence à remplir les documents administratifs.

Tour de table : présentez-vous !

- Votre nom
- Votre métier
- Votre société client si applicable
- Vos compétences dans les domaines liés à cette formation
- Vos objectifs et vos attentes vis-à-vis de cette formation

Principaux outils

Principaux outils

Python

Historique

1989 Création du langage par Guido Van Rossum

2001 Lancement de la Python Software Foundation

2001 Passage en GPL

2009 Python 3



Caractéristiques

Python est :

Interprété Et compilé à la volée, modules en C

Orienté objet (mais pas que)

Portable Compatible avec toutes les plateformes actuelles

Flexible Couteau suisse, de l'admin système au webdev

Populaire Top 5 des langages les plus utilisés depuis des années

Points forts/faibles

Atouts

- Stable
- multi-plateforme
- Facile à apprendre
- Grande communauté (le plus utilisé depuis 2019)
- un besoin, un module

Inconvénients

- Non-compilé
 - Plus lent qu'un langage bas-niveau
 - Optimiser une opération \Rightarrow pas facile à apprendre

Plateformes

Différents interpréteurs :

- Python/CPython \Rightarrow C
- Jython \Rightarrow Java
- IronPython \Rightarrow .Net

Domaines

Domaines d'applications :

- Web (Django ,Flask, ...)
- Sciences (Data mining, Machine learning, Physique, ...)
- OS (Linux, Raspberry, Script administration système, ...)
- Éducation (Initiation à la programmation)
- CAO 3D (FreeCAD, pythonCAD, ...)
- Multimédia (Kodi, ...)

Syntaxe

Utilisation de l'indentation pour délimiter les blocs :

```
a = "une chaîne de caractères"
b = a
a = 8

if a > 5:
    print(f"a = {a}; b = {b}")
else:
    print("c'est étrange")
```

```
a = 8; b = une chaîne de caractères
```

Principaux outils

Notebooks Jupyter

Présentation

Jupyter est un environnement de développement avec interface web.

Plus de 40 langages de programmation supportés, dont Python.



Démonstration

Une démonstration vaut mieux qu'un long discours !

[Jupyter Notebook Demo](#)

Principaux outils

Anaconda

Présentation

Distribution Python faite pour la « Data Science »



Points forts

- Évite les conflits de dépendances entre les principaux paquets
- Peut déléguer à pip
- S'installe facilement sous Windows, Mac et Linux

Avez-vous des questions ?

Principaux outils

Travaux Pratiques

Objectifs

Ce qui sera installé après ce tutoriel :

- Python
- Jupyter Notebook
- Des librairies de « Data Science » :
 - SciPy, Numpy
 - Pandas, seaborn
 - scikit-learn, statsmodels
 - Matplotlib

Instructions

1. [Installer Anaconda](#)
2. Lancer le navigateur Anaconda
3. Cliquer sur Jupyter
4. Charger `Anaconda.ipynb` que vous trouverez dans le dossier `ressources`
5. Éditer et exécuter les cellules pour prendre en main cet environnement de développement.

Machine Learning

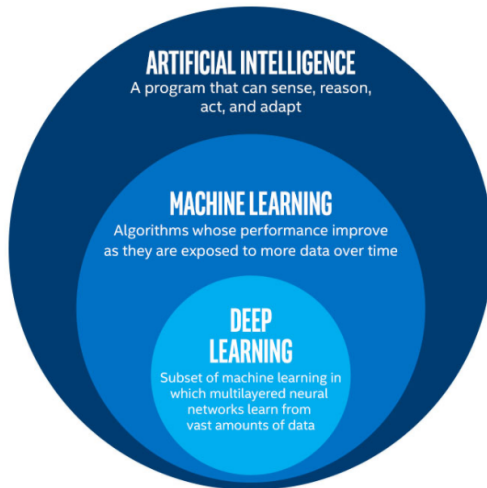
Machine Learning

Introduction

Un domaine vaste



Hierarchie des noms



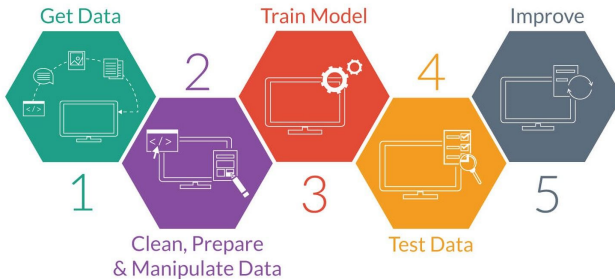
Machine Learning

Nouvelle manière d'aborder la **conception logicielle**.

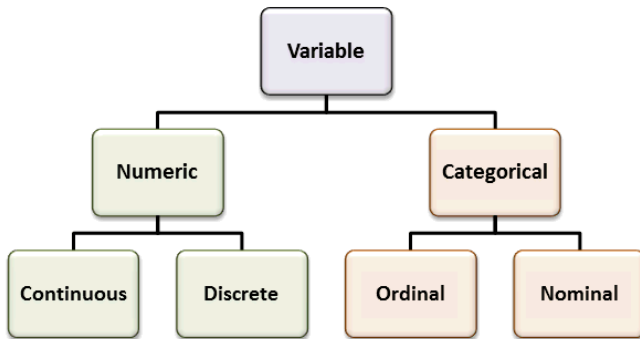
Changement de paradigme

Programmation Explicite → Programmation Implicite

Ingénierie



Matière première : les données



Grandes familles

Apprentissage **supervisé** ou **non-supervisé**, voire **par renforcement** ?

Apprentissage non-supervisé

Faire émerger des profils, des groupes

Exemple

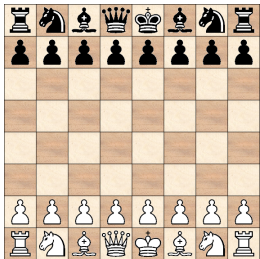
groupes de clients pour adapter sa stratégie marketing

Apprentissage supervisé

Prédire une valeur numérique (**régression**) ou l'appartenance à une classe (**Classification**).

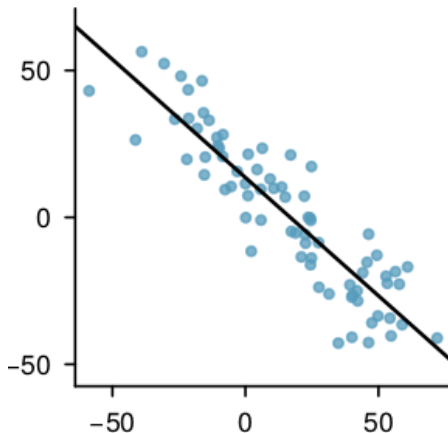
Apprentissage par Renforcement

Apprendre une **stratégie** efficace dans un **univers** où les **actions** fournissent des **récompenses** (possiblement négatives)



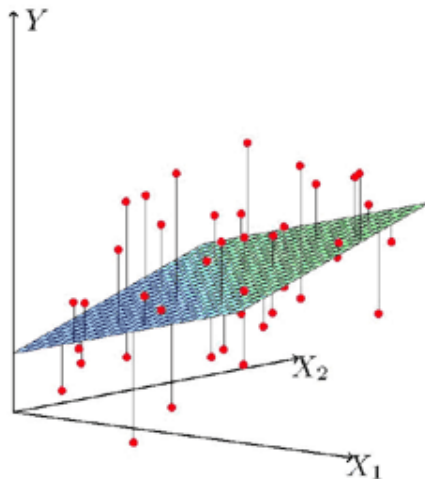
Exemple — Régression linéaire

Prédire une valeur en fonction d'une autre

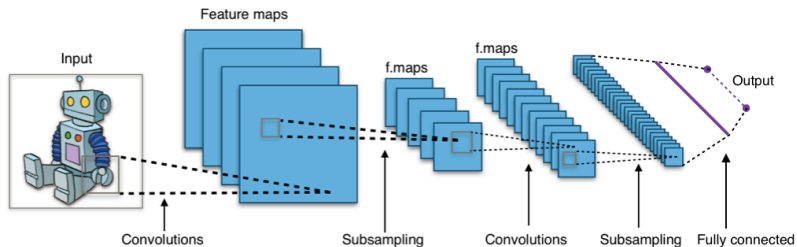


Exemple — Régression linéaire multiple

Prédire une valeur en fonction de plusieurs autres



Exemple — Classification avec des réseaux à convolutions



Exemple — Classification avec des réseaux à convolutions



[001.ak47](#)



[002.american-flag](#)



[003.backpack](#)



[004.baseball-bat](#)



[005.baseball-glove](#)



[006.basketball-hoop](#)



[007.bat](#)



[008.bathtub](#)



[009.bear](#)



[010.beer-mug](#)



[011.billiards](#)

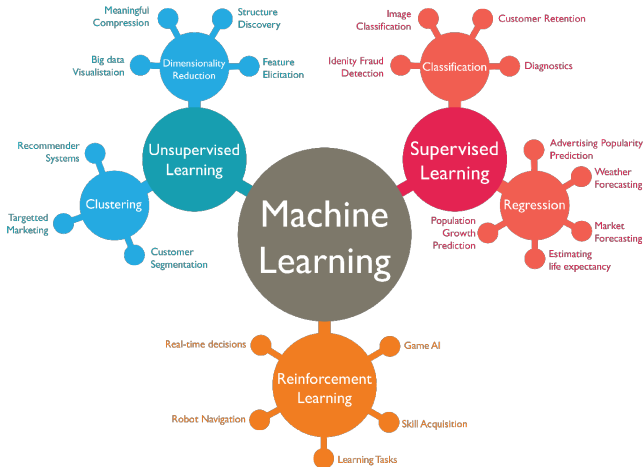


[012.binoculars](#)

Exemple — Apprentissage par renforcement



Topologie du domaine



Points de vue

Beaucoup de façons de voir le machine learning. Basées sur :

- les paradigmes (supervisé, non supervisé, renforcement, en ligne, ...)
- les modèles (arbres, grammaires, automates, réseaux de neurones ...)
- les données (tabulaire, image, texte, vidéo, graphe, ...)
- les techniques (statistiques, symboliques, probabilistes, ...)
- les contraintes (real time, embarqué, big data, multilingue, ...)

→ Domaine **extrêmement** vaste.

Choisir la bonne facette

Critères pour s'orienter dans les approches de machine learning :

- quantité de données à disposition
- qualité du signal d'apprentissage dans les données
- difficulté du problème à résoudre
- besoin d'interprétabilité
- contraintes techniques
- contraintes de délai
- ... et d'autres en fonction des domaines métiers

Conclusion

- le machine learning est un champ vaste.
- il existe sûrement un modèle/paradigme pour vos besoins
- l'important est de définir les bons critères

Discussion

- à quelles données allez-vous appliquer le machine learning ? À quels besoins ?
- aurez-vous besoin de modèles interprétables ou simplement très performant en prédiction ?
- quelles sont vos contraintes ?

Machine Learning

Quelques Prérequis Mathématiques

Objectifs

- exprimer des transformations de données grâce à l'algèbre linéaire
- minimiser des fonctions analytiquement
- décrire l'incertain
- décrire des données

Machine Learning

Algèbre Linéaire

Utilité

- décrire des transformations simples sur un dataset entier avec des mécanismes adaptés
- comprendre les possibilités et les limites de ces transformations simples.

Transformation linéaire

- algèbre linéaire = on se limite aux sommes pondérées des inputs.
- bonne nouvelle : énorme partie des opérations en machine learning

Description des données — échantillon

Python :

```
data = (1, 3)
```

Algèbre linéaire :

$$\mathbf{d} = \begin{bmatrix} 1 \\ 3 \end{bmatrix}$$

Description des données — dataset

Python :

```
data = [(1, 3),  
        (2, 2),  
        (4, 2)]
```

Algèbre linéaire :

$$\mathbf{D} = \begin{bmatrix} 1 & 2 & 4 \\ 3 & 2 & 2 \end{bmatrix}$$

Description des transformations linéaires

Python :

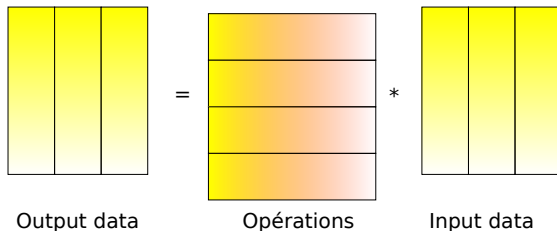
```
def weights(x, y):  
    return x * 2 + y / 2
```

Algèbre linéaire :

$$\mathbf{w} = \begin{bmatrix} 2 & \frac{1}{2} \end{bmatrix}$$

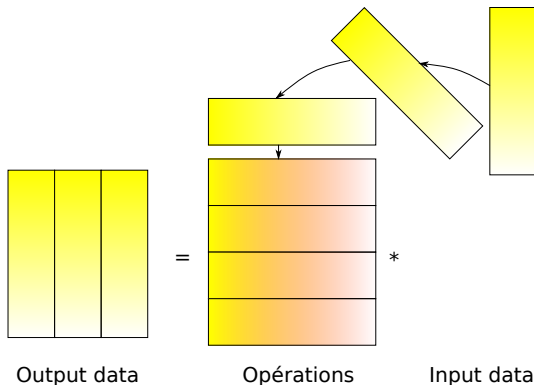
Transformation linéaire = somme pondérée.

Application d'une transformation linéaire à un exemple



Bonne intuition à garder : Verser les colonnes (les exemples du dataset) dans les lignes (les opérations).

Bonne intuition à garder



Bonne intuition à garder : Verser les colonnes (les exemples du dataset) dans les lignes (les opérations).

Application d'une transformation linéaire à un exemple

Python :

```
data = (1, 3)
```

```
def weights(x, y):  
    return 2 * x + y / 2
```

```
res = weights(*data)
```

Algèbre linéaire :

$$\begin{aligned} f &= \begin{bmatrix} 2 & \frac{1}{2} \end{bmatrix} \begin{bmatrix} 1 \\ 3 \end{bmatrix} \\ &= 2 \times 1 + \frac{1}{2} \times 3 \end{aligned}$$

Application d'une transformation linéaire à un dataset

Python :

```
data = [(1, 3),  
        (2, 2),  
        (4, 2)]
```

```
def f(x, y):  
    return x * 2 + y / 2
```

```
res = [f(x, y)  
       for x, y  
       in data]
```

Algèbre linéaire :

$$\begin{aligned} \text{res} &= \begin{bmatrix} 2 & \frac{1}{2} \end{bmatrix} \begin{bmatrix} 1 & 2 & 4 \\ 3 & 2 & 2 \end{bmatrix} \\ &= \begin{bmatrix} 3,5 & 5 & 9 \end{bmatrix} \end{aligned}$$

Application de plusieurs transformations linéaires à un dataset

Python :

```
data = [(1, 3), (2, 2),
        (4, 2)]
```

```
def f(x, y):
    return x * 2 + y / 2
```

```
def g(x, y):
    return x / 2 + y * 2
```

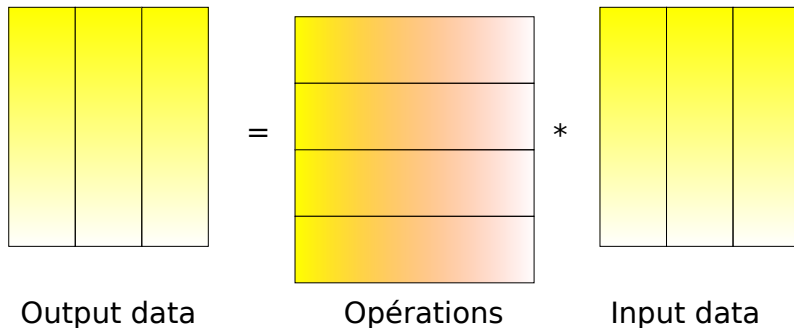
```
res = [[t(x, y) for x, y
          in data]
        for t in [f, g]]
```

Algèbre linéaire :

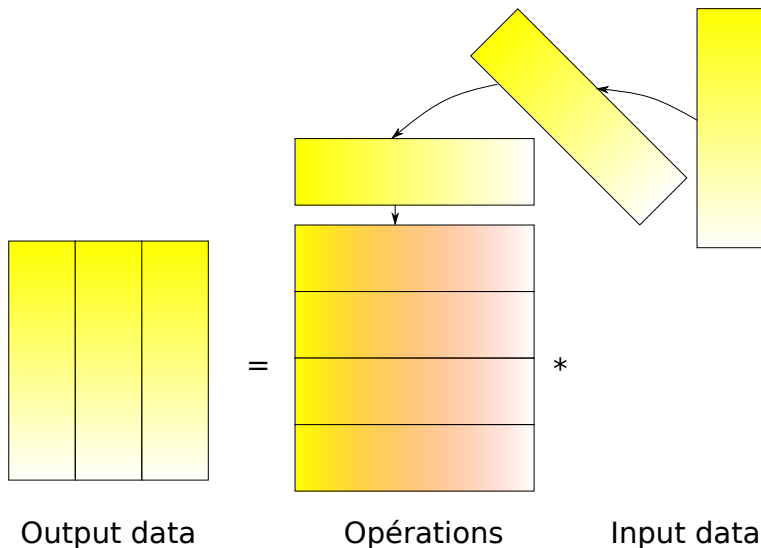
$$\text{res} = \begin{bmatrix} 2 & \frac{1}{2} \\ \frac{1}{2} & 2 \end{bmatrix} \begin{bmatrix} 1 & 2 & 4 \\ 3 & 2 & 2 \end{bmatrix}$$

$$= \begin{bmatrix} 3,5 & 5 & 9 \\ 6,5 & 5 & 6 \end{bmatrix}$$

Bonne intuition à garder



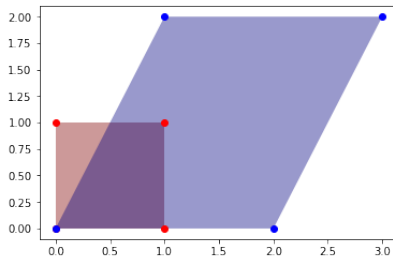
Bonne intuition à garder



Exercice

$$\begin{bmatrix} 1 & 2 \\ 2 & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix} = ?$$

Exemple de transformation

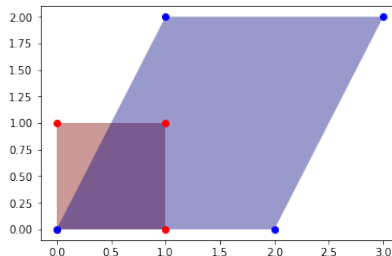


$$\text{Bleu} = \text{Transformation} \times \text{Rouge}$$

$$= \begin{bmatrix} 1 & 2 \\ 2 & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}$$

$$= \begin{bmatrix} 0 & 2 & 3 & 1 \\ 0 & 0 & 2 & 2 \end{bmatrix}$$

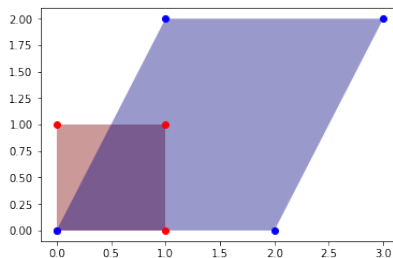
Vecteur propre



Vecteur partant de l'origine qui conserve sa direction malgré la transformation.

Pouvez-vous en trouver un ? $\begin{bmatrix} 1 \\ 0 \end{bmatrix}$ par exemple.

Valeur propre



Facteur par lequel un vecteur propre est redimensionné.

Quelle est la valeur propre de $\begin{bmatrix} 1 \\ 0 \end{bmatrix}$? 2.

Machine Learning

Analyse

Souvent besoin de minimiser une fonction en machine learning.

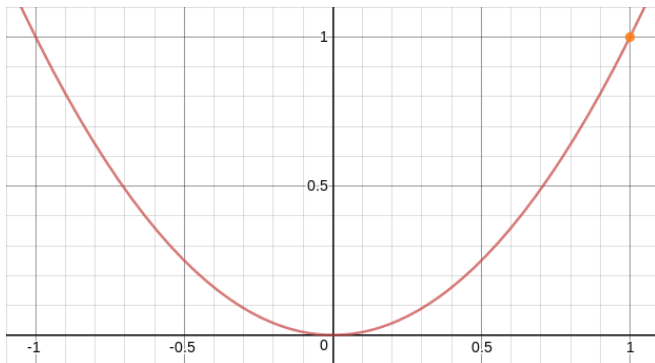
Idée clef

Décider d'un x de départ puis suivre la pente jusqu'au minimum.

Pente = dérivée

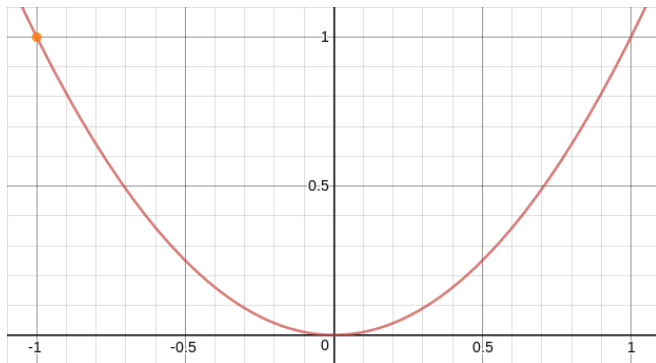
→ Modifier itérativement x par un pas vers l'opposé de la dérivée.

Pente positive



Opposé de la pente = -2 . Avec un pas de 0,1, on passe de 1 à 0,8.

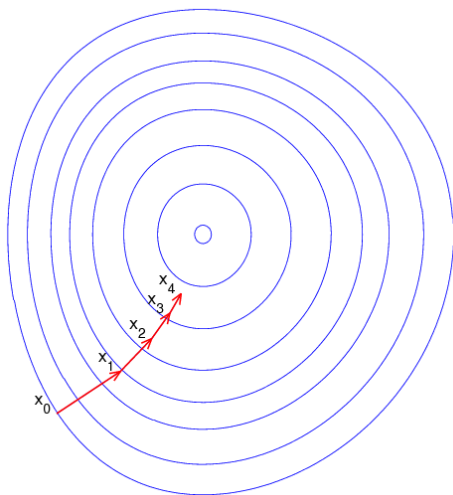
Pente négative



Opposé de la pente = 2. Avec un pas de 0,1, on passe de -1 à $-0,8$.

Exemple en 2 dimensions

dérivée \rightarrow gradient



Machine Learning

Probabilités

Utilité

- quantifier l'incertain
- support pour les statistiques

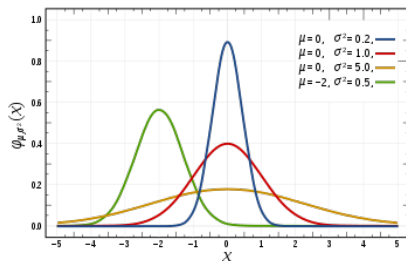
Probabilité

- la probabilité de l'événement X est notée $P(X)$
- $P(X) \in [0, 1]$
- $P(X) = 0 \iff X$ est impossible
- $P(X) = 1 \iff X$ est certain
- $P(\neg X) = 1 - P(X)$

Loi de probabilité

Décrit le comportement aléatoire d'un phénomène dépendant du hasard.

- $\sum_u P(X = u) = 1$ en discret
- $\int P(X) dX = 1$ en continu
- loi uniforme
- loi normale/gaussienne



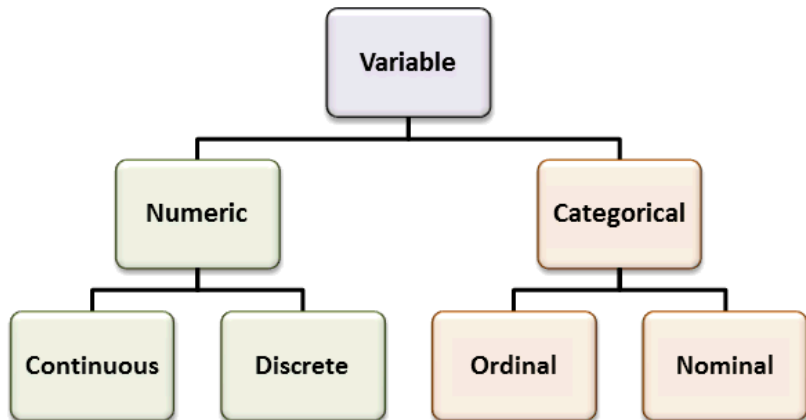
Machine Learning

Statistiques

Utilité

- description et compréhension des données
- correction pour faciliter les traitements

Types de variables



Hypothèse

Pré-requis pour les mesures statistiques qui suivent (et la plupart du machine learning) :

- les données **doivent être issues d'une même loi**
- chaque échantillon doit être **indépendant** des autres
- **pas évident en pratique ! Pourquoi ?**

Variance

Mesure la dispersion d'une série statistique (ou d'une variable) :

$$V(X) = \mathbb{E} [(X - \mathbb{E}[X])^2]$$

Pour la calculer :

$$V(X) = \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2$$

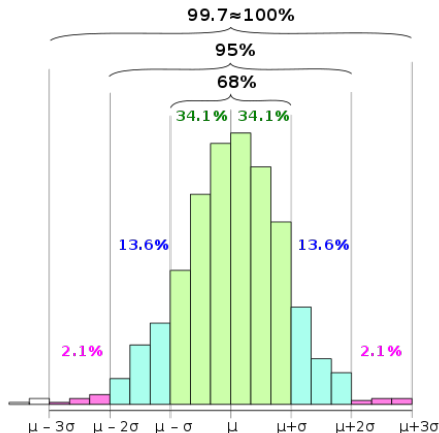
Écart-type

Racine carrée de la variance

$$\sigma(X) = \sqrt{V(X)}$$

Écart-type — règle des 68, 95 et 99,7

Pour les lois normales :



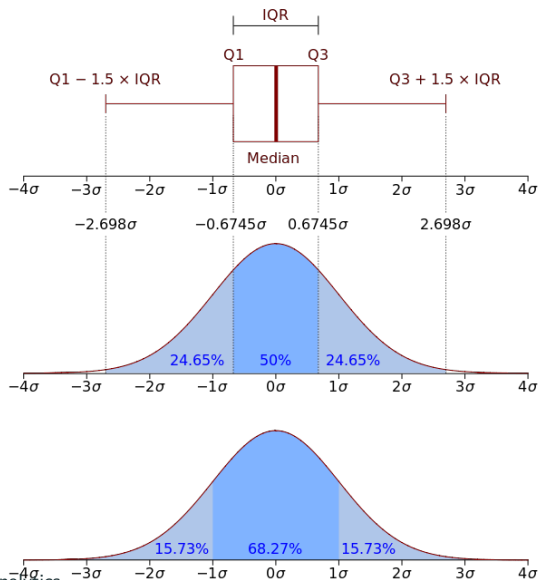
Quartile

Les quartiles (Q_1 , Q_2 et Q_3) divisent les données en 4 intervalles contenant le même nombre d'observations.

Déclinable en quantile de taille arbitraire (décile, percentile).

Que veut dire être dans le 95^e percentile ?

Boxplot



Covariance

Mesure la variabilité jointe de deux variables aléatoires :

$$V(X) = \mathbb{E} [(X - \mathbb{E}[X])(X - \mathbb{E}[X])]$$

$$\text{cov}(X, Y) = \mathbb{E} [(X - \mathbb{E}[X])(Y - \mathbb{E}[Y])]$$

Pour la calculer :

$$\text{cov}(X, Y) = \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})$$

Corrélation

Covariance divisée par le produit des écart-types :

$$\text{corr}(X, Y) = \frac{\text{cov}(X, Y)}{\sigma_X \sigma_Y}$$

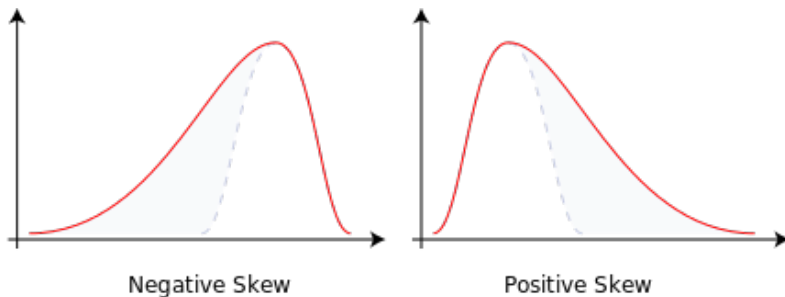
Intérêt ? Pas d'unité.

Test de normalité

Pour tester (et corriger) la normalité d'une distribution, on utilise deux mesures :

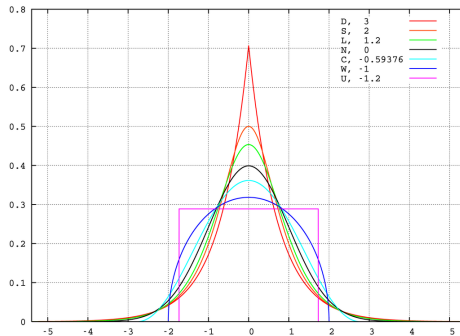
- l'asymétrie (*skew*)
- le kurtosis

Asymétrie



$$\text{asym}(X) = \mathbb{E} \left[\left(\frac{X - \bar{X}}{\sigma} \right)^3 \right]$$

Kurtosis



$$\text{kurt}(X) = \mathbb{E} \left[\left(\frac{X - \mu}{\sigma} \right)^4 \right]$$

Transformation de Box-Cox

Asymétrie et kurtosis peuvent se corriger avec la transformation de Box-Cox ou des transformations log.

Machine Learning

Conclusions

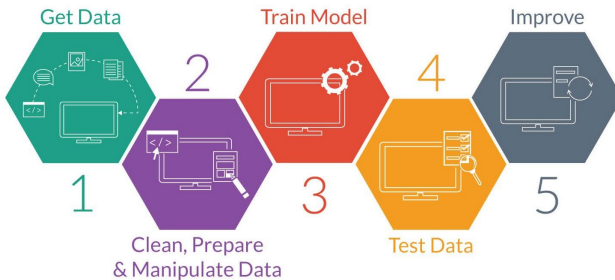
Conclusion

- algèbre linéaire → raisonner sur des opérations simples et les décrire efficacement
- minimiser une fonction continue → dérivée
- décrire l'incertain → probabilités
- caractériser une série de données → statistiques

Machine Learning

**Modélisation et préparation des
données**

Introduction



Biais statistiques

Attention aux différents biais de vos données !

- **variables confondantes** (Ex : “obésité” dans la corrélation entre “conso. viande” et “cancer colon”)
- **biais statistiques**
 - sélection, autosélection
 - mesure
 - attrition
 - ...
- trouver de fausses variables explicatives

→ Le garder en tête pendant toute l'étude.

Qualité des données

Meilleures données > Meilleurs modèles
(trash-in, trash-out)

→ À garder en tête pendant toute l'étude, en particulier durant l'entraînement de modèles

Pipeline de préparation

- valeurs manquantes
- preprocessing (texte, image)
- standardisation
- transformation

Valeurs manquantes

Gênant pour certains modèles. Plusieurs options :

- supprimer les enregistrements
- remplacer par une valeur (imputation) :
 - constante
 - moyenne de la colonne
 - prédiction d'un autre modèle

Prétraitement

- tokenizer, POS-tagger le texte (<https://spacy.io/>)
- utiliser un réseau de neurones préentraîné sur les images (<https://keras.io/applications/>)
- appliquer une transformée de fourier sur le son
- ...

Standardisation

Beaucoup de modèles travaillent mieux avec des données normales et sont plus efficaces autour de $[-5, 5]$:

- centrer sur la moyenne puis diviser par l'écart-type
- transformation de Box-Cox en cas d'asymétrie
- transformations spécifiques en fonction de la distribution

Transformation

Quand un modèle n'accepte pas de données catégorielles :

- label encoding si ordinal
- one-hot encoding sinon

Label encoding

Si les données sont ordinales :

Ordinal :

Température
Froid
Froid
Tiède
Chaud
Tiède

Label encoding :

Température
1
1
2
3
2

One-hot encoding

Remplacer une feature par n features avec n le nombre de catégories.

Catégoriel :

Couleur
Rouge
Rouge
Jaune
Vert
Jaune

One-hot :

Rouge	Jaune	Vert
1	0	0
1	0	0
0	1	0
0	0	1
0	1	0

Exploration des données

But :

- se rendre compte des prétraitements à effectuer (Box-Cox, imputations, etc)
- comprendre la variable de sortie : distribution, équilibre des classes, features les plus corrélées, ...
- détecter les corrélations
- appréhender la complexité nécessaire du modèle

Attention : garder des données de côté (test set) et ne pas les regarder.
Sinon biais statistique énorme.

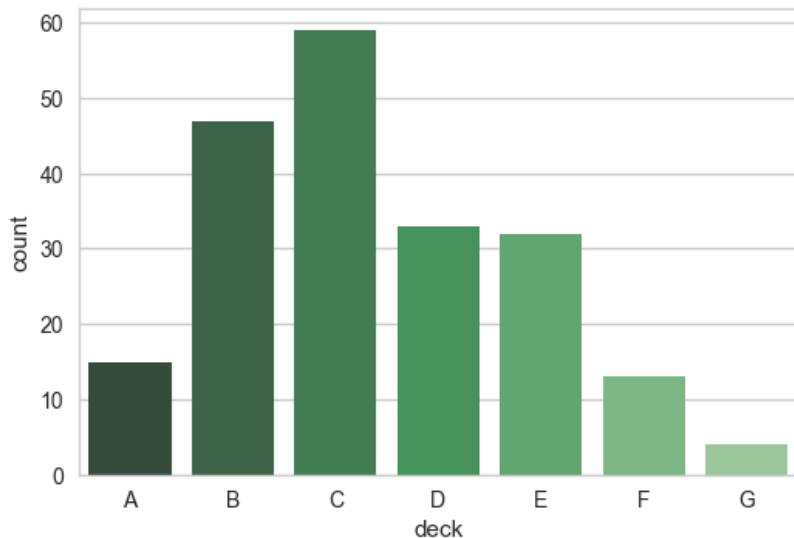
Outils

Plusieurs outils sont disponibles pour explorer des données. On utilise principalement des plots pour :

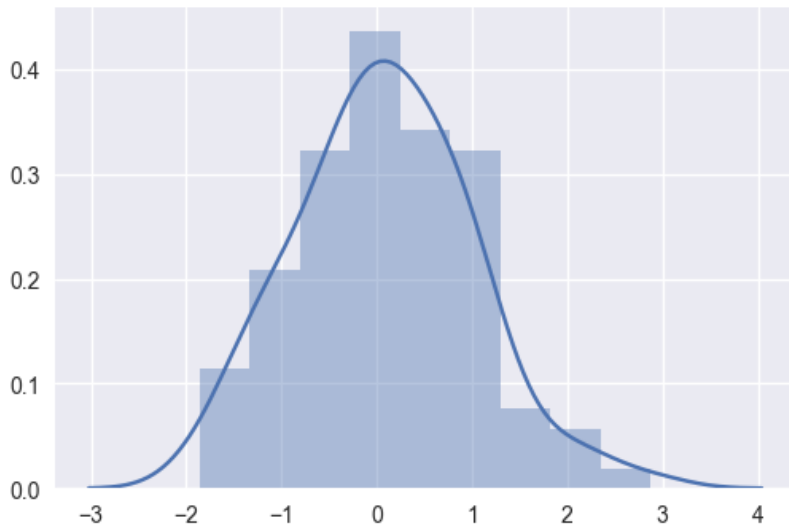
- se renseigner sur une distribution
- se renseigner sur la corrélation de deux distributions
- visualiser des corrélations linéaires

Les outils suivants sont sauf mention contraire présents dans [seaborn](#).

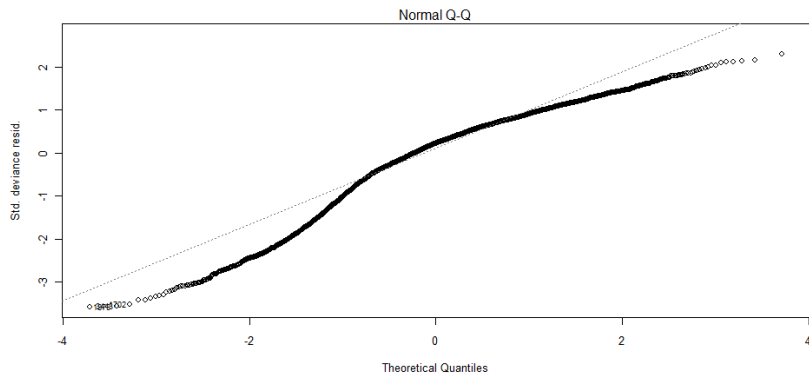
Outils — count plot



Outils — dist plot

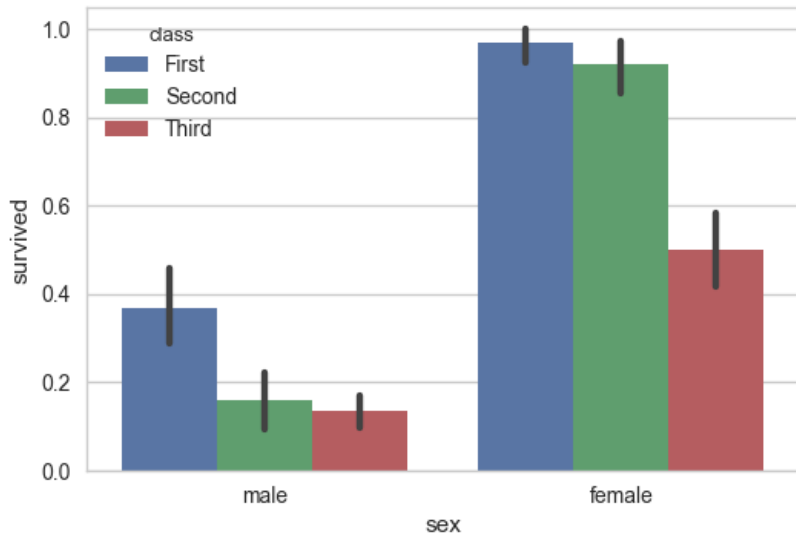


Outils — qq plot

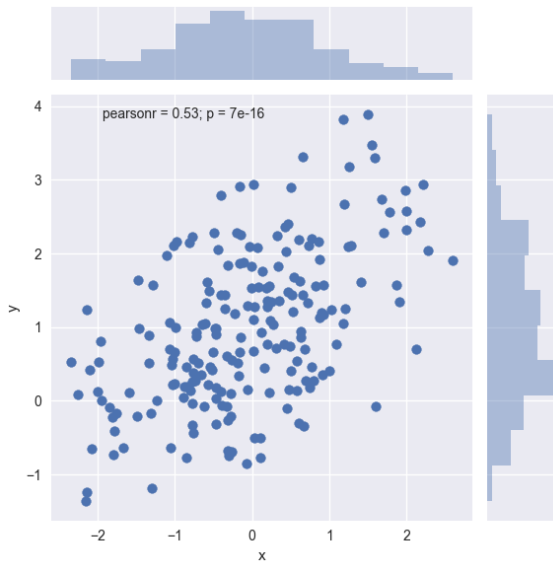


Attention, pas [seaborn](#) mais [statsmodel](#) ou [scipy.stats](#).

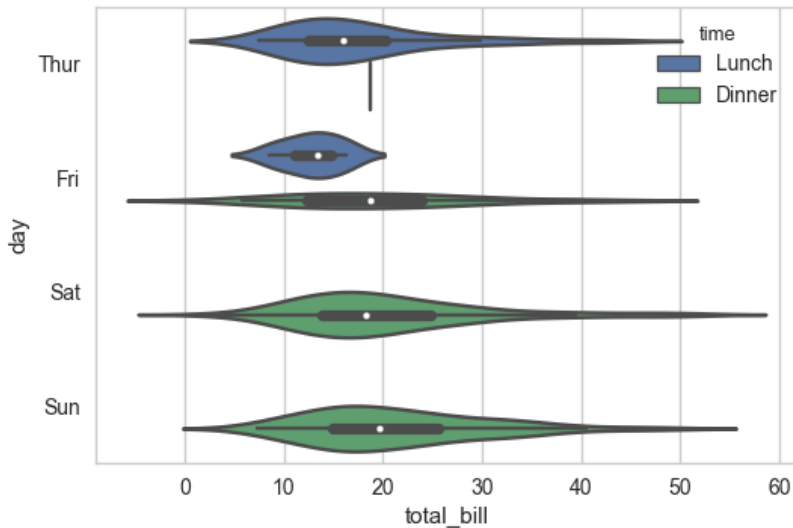
Outils — bar plot



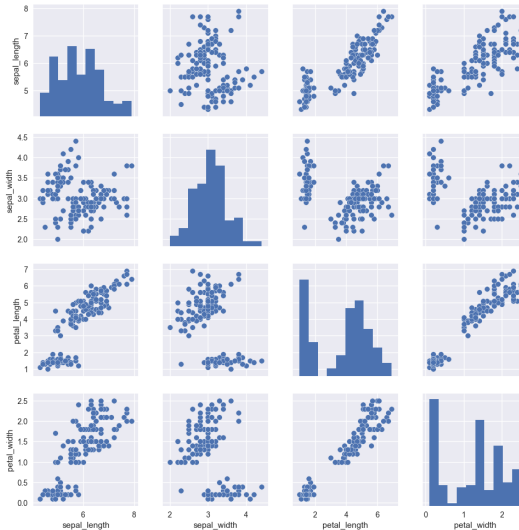
Outils — scatter plot



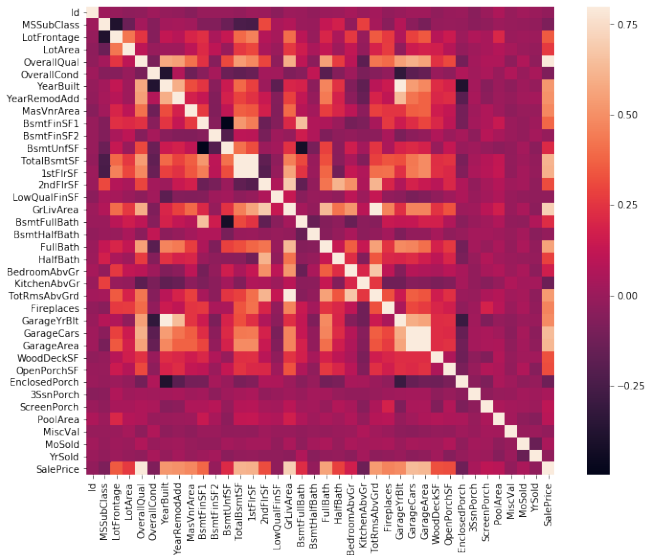
Outils — violin plot



Outils — pair plot



Outils — correlation matrix



Mode opératoire

Bonnes pratiques pour explorer un dataset :

- analyser la(es) variable(s) de sortie (countplot/distplot)
- trouver les corrélations linéaires les plus fortes
- analyser les variables correspondantes
- regarder s'il y a des outliers évidents dans ces variables

Machine Learning

Évaluation

Précision, rappel

En classification :

Précision

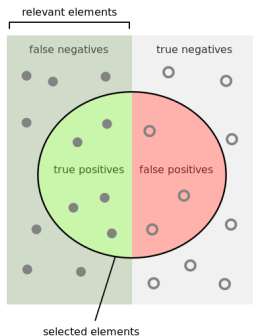
$$\frac{\text{vrais positifs}}{\text{vrais positifs} + \text{faux positifs}}$$

Rappel

$$\frac{\text{vrais positifs}}{\text{vrais positifs} + \text{faux négatifs}}$$

F-mesure moyenne harmonique entre précision et rappel (aussi appelée F1 score)

Précision, rappel



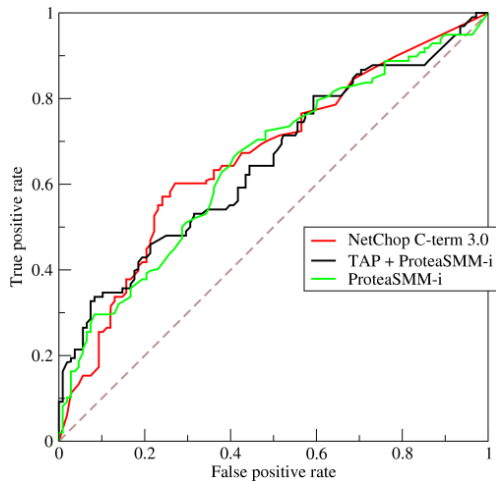
How many selected items are relevant?

$$\text{Precision} = \frac{\text{true positives}}{\text{true positives} + \text{false positives}}$$

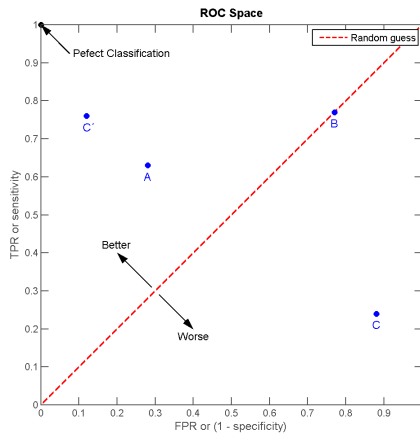
How many relevant items are selected?

$$\text{Recall} = \frac{\text{true positives}}{\text{true positives} + \text{false negatives}}$$

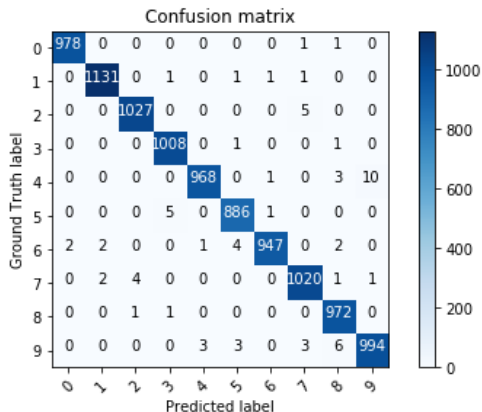
Courbe ROC



Analyse d'une courbe ROC



Matrice de confusion



Avez-vous des questions ?

Fonctions utiles

[Fonctions Utiles \(cliquez ici\)](#)

Un peu d'aide, mais ça ne vaudra jamais de regarder la documentation des librairies utilisées :

[Compilation de CheatSheet](#)

Machine Learning

Travaux Pratiques

Instructions

Exploration & préparation de données