# [Découvrez la science des données pour les objets connectés](https://openclassrooms.com/fr/courses/5919236-decouvrez-la-science-des-donnees-pour-les-objets-connectes)

Table des matières

[Découvrez la science des données pour les objets connectés 1](#_Toc57908464)

[Comprenez les enjeux et objectifs de ce cours 7](#_Toc57908465)

[**Objectif à terme** 8](#_Toc57908466)

[**Réalisations dans ce cours** 8](#_Toc57908467)

[**Expérimentation** 9](#_Toc57908468)

[**Signaux enregistrés** 9](#_Toc57908469)

[Initiez-vous à la science des données et à l'IA 11](#_Toc57908470)

[**De la statistique à la science des données et à l'IA** 11](#_Toc57908471)

[**Algorithme et décision automatique** 13](#_Toc57908472)

[Identifiez les étapes de la science des données 15](#_Toc57908473)

[**Préparation des données & feature engineering** 15](#_Toc57908474)

[**Apprentissage supervisé** 15](#_Toc57908475)

[Préparez-vous pour la suite de ce cours 17](#_Toc57908476)

[Objectif 17](#_Toc57908477)

[Contenu du cours 17](#_Toc57908478)

[Activités préalables 18](#_Toc57908479)

[Références 19](#_Toc57908480)

[Identifiez les enjeux et objectifs de ce cours 20](#_Toc57908481)

[Compétences évaluées 20](#_Toc57908482)

[ Question 1 20](#_Toc57908483)

[ Question 2 20](#_Toc57908484)

[ Question 3 21](#_Toc57908485)

[ Question 4 21](#_Toc57908486)

[ Question 5 22](#_Toc57908487)

[ Question 6 22](#_Toc57908488)

[ Question 7 23](#_Toc57908489)

[ Question 8 23](#_Toc57908490)

[ Question 9 23](#_Toc57908491)

[ Question 10 24](#_Toc57908492)

[Prenez en charge les données 25](#_Toc57908493)

[**Data munging** 25](#_Toc57908494)

[**Visualisez des signaux bruts** 25](#_Toc57908495)

[**Transformez les signaux** 26](#_Toc57908496)

[Comprenez l'analyse en composantes principales 28](#_Toc57908497)

[Objectifs 28](#_Toc57908498)

[Exemple jouet 28](#_Toc57908499)

[L'analyse factorielle discriminante, cas particuler d'ACP 39](#_Toc57908500)

[Explorez des données complexes 43](#_Toc57908501)

[ACP de données de pollution 43](#_Toc57908502)

[ACP des transformations des signaux 44](#_Toc57908503)

[AFD des transformations des signaux 48](#_Toc57908504)

[Mettez en pratique les concepts de la partie 2 50](#_Toc57908505)

[Que faut-il retenir ? 50](#_Toc57908506)

[Tutoriels 50](#_Toc57908507)

[Références 51](#_Toc57908508)

[Explorez des données multidimensionnelles 52](#_Toc57908509)

[Compétences évaluées 52](#_Toc57908510)

[ Question 1 52](#_Toc57908511)

[ Question 2 52](#_Toc57908512)

[ Question 3 52](#_Toc57908513)

[ Question 4 53](#_Toc57908514)

[ Question 5 53](#_Toc57908515)

[ Question 6 53](#_Toc57908516)

[ Question 7 54](#_Toc57908517)

[ Question 8 54](#_Toc57908518)

[ Question 9 54](#_Toc57908519)

[ Question 10 55](#_Toc57908520)

[ Question 11 55](#_Toc57908521)

[ Question 12 56](#_Toc57908522)

[Abordez les principes de l'apprentissage statistique 57](#_Toc57908523)

[Modélisation et prévision 57](#_Toc57908524)

[Erreur de prévision 58](#_Toc57908525)

[Sélection de modèle 59](#_Toc57908526)

[Sélection par pénalisation 61](#_Toc57908527)

[Optimisation par validation croisée V - couches 63](#_Toc57908528)

[Comprenez la classification supervisée 65](#_Toc57908529)

[Attention 65](#_Toc57908530)

[Classification binaire par régression logistique 65](#_Toc57908531)

[Courbe ROC 66](#_Toc57908532)

[Y=1Y=1 67](#_Toc57908533)

[Y=0Y=0 67](#_Toc57908534)

[yˆi=1y^i=1 67](#_Toc57908535)

[n11(s)n11(s) 67](#_Toc57908536)

[n10(s)n10(s) 67](#_Toc57908537)

[n1+(s)n1+(s) 67](#_Toc57908538)

[yˆi=0y^i=0 67](#_Toc57908539)

[n01(s)n01(s) 67](#_Toc57908540)

[n00(s)n00(s) 67](#_Toc57908541)

[n0+(s)n0+(s) 67](#_Toc57908542)

[n+1n+1 67](#_Toc57908543)

[n+0n+0 67](#_Toc57908544)

[nn 67](#_Toc57908545)

[Modélisez des données complexes 70](#_Toc57908546)

[Prévision de la concentration en ozone 70](#_Toc57908547)

[Prévision de dépassement du seuil 72](#_Toc57908548)

[Reconnaissance de l'activité humaine 73](#_Toc57908549)

[Mettez en pratique les concepts de la partie 3 75](#_Toc57908550)

[Que faut-il retenir de cette partie du cours ? 75](#_Toc57908551)

[Tutoriels 75](#_Toc57908552)

[Références 75](#_Toc57908553)

[Prévoyez une activité humaine par classification supervisée 76](#_Toc57908554)

[Compétences évaluées 76](#_Toc57908555)

[ Question 1 76](#_Toc57908556)

[ Question 2 76](#_Toc57908557)

[ Question 3 76](#_Toc57908558)

[ Question 4 77](#_Toc57908559)

[ Question 5 77](#_Toc57908560)

[ Question 6 77](#_Toc57908561)

[ Question 7 78](#_Toc57908562)

[ Question 8 78](#_Toc57908563)

[ Question 9 78](#_Toc57908564)

[ Question 10 79](#_Toc57908565)

[ Question 11 79](#_Toc57908566)

[ Question 12 79](#_Toc57908567)

[Abordez les fondements de l'intelligence artificielle 81](#_Toc57908568)

[**Historique** 81](#_Toc57908569)

[**Neurone formel** 82](#_Toc57908570)

[**Réseaux de neurones élémentaires (perceptron)** 83](#_Toc57908571)

[Appréhendez l'apprentissage des réseaux de neurones 85](#_Toc57908572)

[Fonction de transfert 85](#_Toc57908573)

[Apprentissage 86](#_Toc57908574)

[Algorithme de rétropropagation élémentaire du gradient : 87](#_Toc57908575)

[ Initialisation 87](#_Toc57908576)

[ Tant que Q< 87](#_Toc57908577)

[o pour 87](#_Toc57908578)

[Contrôle de la complexité 88](#_Toc57908579)

[Remarques 89](#_Toc57908580)

[Entraînez un réseau de neurones profond 90](#_Toc57908581)

[Préambule 90](#_Toc57908582)

[Reconnaissance d'images 90](#_Toc57908583)

[Couches pour l'apprentissage profond 92](#_Toc57908584)

[Transfert d'apprentissage 93](#_Toc57908585)

[Mettez en pratique les concepts de la partie 4 94](#_Toc57908586)

[Tutoriel 94](#_Toc57908587)

[Ce qu'il faut retenir de ce cours 94](#_Toc57908588)

[Appréhendez l'apprentissage profond (deep learning) 97](#_Toc57908589)

[Compétences évaluées 97](#_Toc57908590)

[ Question 1 97](#_Toc57908591)

[ Question 2 97](#_Toc57908592)

[ Question 3 98](#_Toc57908593)

[ Question 4 98](#_Toc57908594)

[ Question 5 98](#_Toc57908595)

[ Question 6 99](#_Toc57908596)

[ Question 7 99](#_Toc57908597)

[ Question 8 99](#_Toc57908598)

[ Question 9 100](#_Toc57908599)

[ Question 10 100](#_Toc57908600)

Le principal objectif de ce cours est d’illustrer, sur un cas d’usage provenant de la communauté IoT, les possibilités de la science des données ou encore de l'intelligence artificielle. Plus précisément, il s'agit de mettre en œuvre des algorithmes d’apprentissage automatique/statistique (*machine/statistical learning*).

En résumé :**comment embarquer de l'IA dans un objet connecté ?**

Les données proviennent en effet d’enregistrements de l’accéléromètre et du gyroscope de smartphones, auxquels sont associées des transformations ou caractéristiques (*features*) qui en découlent (moyenne, corrélations, entropie, énergie dans une bande de fréquence...).

L’objectif concret est d’identifier l’*activité du porteur*: couché, assis, debout, marche, monte ou descend un escalier à partir des signaux ou de leurs transformations. Les applications et valorisations de ces données sont assez facile à imaginer : mesure d”activités, suivi de personnes dépendantes, déclenchement d’une alarme…

Nous commencerons par analyser le problème posé et le traduire en une démarche de science des données. Cela conduit à une séquence d'étapes bien identifiées :

* explorer les propriétés des signaux issus de capteurs embarqués et de leurs transformations (*features*), en utilisant l’analyse en composantes principales (ACP) puis l’analyse factorielle discriminante (AFD) ;
* estimer un modèle statistique ou entraîner un algorithme d'apprentissage élémentaire  sur les données transformées, en apprécier la qualité ;
* entraîner un algorithme complexe *(deep learning*) sur les signaux bruts pour atteindre la même qualité mais sans transformations préalables des données, trop coûteuses en énergie pour la batterie de l'objet connecté.

**À la fin de ce cours, vous serez capable de** :

* mettre en place les outils nécessaires à la science des données ;
* exécuter une analyse exploratoire multidimensionnelle ;
* expliquer la démarche et les enjeux de la classification supervisée ;
* entraîner un algorithme de classification supervisée ;
* expliquer l'évolution de l’IA ;
* entraîner un algorithme d’apprentissage profond.

**Prérequis**

Ce cours nécessite quelques connaissances en **mathématiques** (algèbre linéaire, calcul matriciel) et des notions de **traitement du signal** (Fourier).

Les cours suivants sont des bons préalables pour comprendre ou des compléments pour approfondir les aspects méthodologiques de la démarche mise en œuvre :

* [**Explorez vos données avec des algorithmes non supervisés**](https://openclassrooms.com/fr/courses/4379436-explorez-vos-donnees-avec-des-algorithmes-non-supervises)
* [**Initiez-vous au machine learning**](https://openclassrooms.com/fr/courses/4011851-initiez-vous-au-machine-learning)
* [**Utilisez des modèles supervisés non linéaires**](https://openclassrooms.com/fr/courses/4470406-utilisez-des-modeles-supervises-non-lineaires)

**Des contenus actifs !**

Ce cours intègre dans chaque chapitre des **calepins interactifs** (*notebooks Jupyter*) ou **tutoriels**. Ces calepins soulèvent des questions et proposent des liens vers des vignettes de complément de cours : ils vous placent dans une démarche active d'apprentissage.

Le principal tutoriel, **fil rouge du cours**,  a pour résultat final un **prototype d’algorithme (IA) à embarquer** pour la **reconnaissance de l’activité du porteur d'un smartphone**.

# Comprenez les enjeux et objectifs de ce cours

Nous nous intéressons dans ce cours à des objets connectés et plus précisément aux données générées par des objets connectés. Nous apportons une réponse à la question :

Que peut apporter la*science des données,* ou plus exactement les algorithmes d'apprentissage automatique (*machine learning*), pour les valoriser ? Ou encore, comment embarquer une composante d'intelligence artificielle (IA) dans un objet connecté ?



Smartphone

*Attention*, ceci n'est pas un téléphone ou pas seulement, car cet objet connecté embarque un GPS et aussi des capteurs physiques :

* un accéléromètre, et
* un gyroscope.

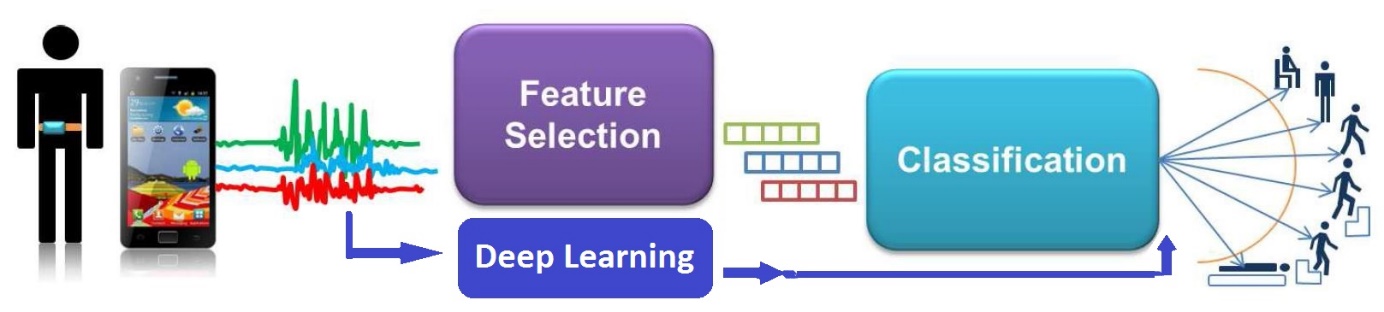
Oublions le GPS pour nous intéresser à l'**accéléromètre** et au **gyroscope**. Ces capteurs génèrent des signaux temporels. Un smartphone est capable de mesurer, enregistrer ou transmettre, en permanence, des accélérations selon les trois directions x, y et z, et des accélérations angulaires autour de ces trois mêmes axes. Imaginez le volume de données généré par des milliers voire des millions de ces objets connectés.

La question est alors de savoir que faire de cette masse de données, comment les **valoriser**. Vous pouvez par exemple installer une application **"podomètre"** sur votre smartphone ; pensez également à l'utilité d'un détecteur de chute pour personne dépendante ou pour un motard.

Ce cours vous propose un objectif plus ambitieux : apprendre à reconnaître l'activité du porteur d'un smartphone, ou plus précisément développer une capacité ou *intelligence* permettant de distinguer parmi 6 activités possibles : marcher, monter ou descendre un escalier, être couché, assis ou debout.

Si vous vous intéressez aux objets connectés et à la valorisation des données que ces objets peuvent générer, découvrez et comprenez dans ce cours les capacités de l'**IA**, et plus précisément de l'**apprentissage automatique**, à atteindre cet objectif. Réciproquement et plus généralement, si vous vous intéressez à l'IA, à la **science des données et à l'apprentissage automatique** (*machine learning*), et que vous voulez appliquer ces compétences à des signaux temporels complexes, suivez ce cours !

**Objectif à terme**



Objectif de la science des données

Le schéma ci-dessus résume le processus pour atteindre l'objectif visé à terme : une phase d'**acquisition** des signaux puis des phases aboutissant à la **reconnaissance** de l'activité. Plus précisément, deux stratégies sont possibles : une première stratégie à partir des **signaux transformés** ou une deuxième exécutée directement sur les **signaux bruts**.

**Réalisations dans ce cours**

Ce cours vous accompagne dans la réalisation des différentes étapes préparant cet objectif. Ces réalisations sont proposées sous la forme d'un tutoriel ou plus précisément d'un *notebook* ou calepin *Jupyter* écrit en Python et faisant appel aux principales librairies d'apprentissage automatique ( scikit−learnscikit−learn ) et d'apprentissage profond ou *deep learning* ( keras−tensorflowkeras−tensorflow ).

Pour atteindre l'objectif d'identification des activités, vous allez suivre et réaliser dans les calepins différentes étapes.

Une étape préliminaire d'**exploration des données** est indispensable. Elle permet de découvrir les principales caractéristiques des données : signaux bruts et transformés (*feature engineering*) ; de les représenter, afin de pouvoir évaluer leur pouvoir discriminant, c'est-à-dire leur capacité à séparer les classes d'activités les unes des autres.

L'étape suivante est celle d'**apprentissage ou entraînement des algorithmes**. Deux stratégies sont mises en œuvre et comparées :

* **la première** consiste à préalablement exécuter un ensemble de transformations de ces signaux pour en extraire des nouvelles variables ou*features* ; le choix de ces transformations est issu d'une expertise métier en traitement du signal. L'objectif de discrimination ou reconnaissance des activités est alors un problème de classification supervisée ou reconnaissance de forme. Cet objectif est atteint en entraînant un certain type d'algorithme d'apprentissage sur les transformations des signaux ;
* **la deuxième** possibilité consiste à entraîner un algorithme d'apprentissage supervisé directement sur les signaux bruts. Nous verrons alors que seul un réseau de neurones profond, c'est-à-dire du *deep learning*, permet d'atteindre l'objectif de reconnaissance avec des résultats similaires à ceux de la première stratégie.

De toute façon, dans les deux cas – données transformées ou signaux bruts – il est nécessaire d'**utiliser une base d'apprentissage ou d'entraînement** obtenue par un ensemble d'expérimentations.

Bien que plus complexe au plan théorique, la réalisation de la deuxième stratégie est motivée par des questions d'**économie d'énergie** et en conséquence d'**autonomie de la batterie**. En effet, demander à un objet embarqué de calculer, en temps réel, un ensemble complexe de transformations, n'est pas réaliste. En revanche, c'est concrètement réalisé en câblant un réseau de neurones dans des puces économes dédiées à la reconnaissance d'un ou de visages, et implantées dans un appareil photo.

**Expérimentation**

Décrivons le processus expérimental qui a permis de constituer la base des données qui serviront à l'entraînement des algorithmes.

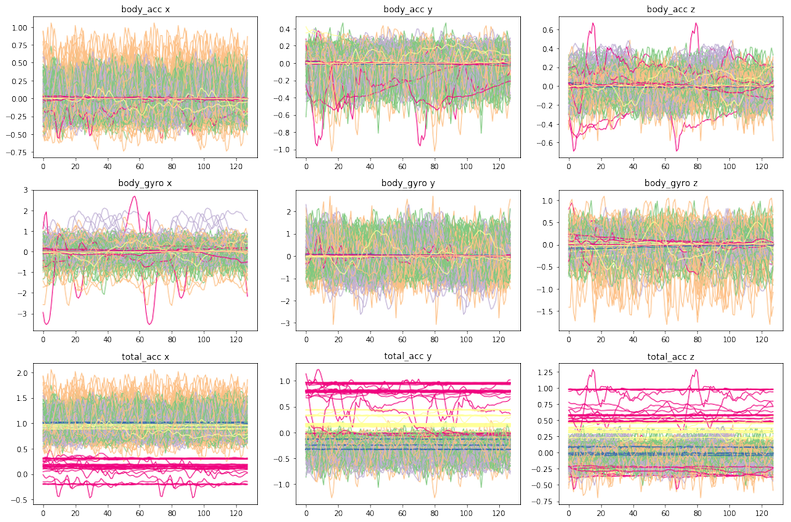
Elles sont le résultat d'une expérimentation détaillée :

* 30 volontaires ont porté un smartphone pendant des activités bien identifiées : marcher, monter ou descendre un escalier, être assis, debout ou couché ;
* qu'ils ont répétées de nombreuses fois ;
* pendant ce temps, 9 signaux sont enregistrés pendant une durée de 2,56 secondes : accélérations en x, y et z, accélérations corrigées de la pesanteur, accélérations angulaires en x, y et z. Les signaux sont échantillonnés à une fréquence de 50 Hz. Ce choix conduit à des séries de 128 valeurs par observation, séries qui peuvent donc être facilement traitées par un algorithme de transformée de Fourier rapide.

Cette expérimentation fournit un tableau ou *data frame* comportant près de 14 000 lignes et 1 152 plus une colonne. La dernière colonne est la variable qualitative YY à 6 classes, variable cible ou label de l'activité. Ce *data frame* constitue la base d'entraînement.

Ces données sont **publiques**. Elles ont été acquises par Anguita et al. (2013) qui décrit très en détail le plan expérimental. Elles sont disponibles sur un [site de l'université de Californie Irvine](https://archive.ics.uci.edu/ml/index.php) dédié à l'apprentissage automatique.

**Signaux enregistrés**

Représentation graphique des signaux produits par des accélérations en x, y et z et celles corrigées de g, accélérations angulaires en x , y et z.

La figure ci-dessus représente ainsi l'ensemble des approximativement **14 000 signaux enregistrés** pendant 2,56 secondes, échantillonnés à 50 Hz et pour chaque expérimentation. La couleur désigne l'**activité**. Il y a un ensemble de courbes par type de signal ou type d'accélération fourni par l'accéléromètre ou le gyroscope. Ces graphiques font surtout ressortir la **complexité des données**. Ce type de signal, très bruité, se retrouve dans l'observation de beaucoup de phénomènes physiques. La démarche de science des données qui suit n'est donc pas spécifique aux seules données issues de smartphones.

# Initiez-vous à la science des données et à l'IA

**De la statistique à la science des données et à l'IA**

Avant d'entrer dans le vif du sujet et de savoir ce que peuvent science des données et intelligence artificielle (IA), il est bon, en introduction, de rappeler d'où ces pratiques proviennent. Voici quelques repères historiques marquants de cette évolution.

* **1930-70 h-Octets***Statistique inférentielle*

Les années 30 voient le développement de la **statistique inférentielle** avec notamment les travaux de Sir Ronald Fisher. Une hypothèse est posée, une expérience planifiée, une statistique de test calculée (Student, ANOVA) ; la statistique est comparée à une valeur seuil. L'hypothèse est acceptée ou rejetée avec un risque αα contrôlé. Il s'agit, par exemple, de montrer qu'une molécule est significativement active, qu'une semence est significativement plus productive... De plus, la décision prise sur l'échantillon peut être inférée à la population entière. Ce cours ne s'intéresse pas à cette statistique inférentielle ni aux pratiques de test associées.

* **1940-50***Intelligence artificielle*

Parallèlement, les premiers ordinateurs sont produits pour répondre aux besoins de l'effort de guerre. Ces découvertes motivent par ailleurs les premières réflexions sur la notion d'intelligence artificielle, avec pour objectif de simuler le fonctionnement du cerveau. Mc Culloch (neurophysiologiste) et Pitts (logicien) introduisent en 1943 le ***neurone formel****,*tandis que Turing (1950) développe la première **théorisation de l'IA** et que Rosenblatt (1957) propose de modéliser le fonctionnement de la rétine par un réseau de neurones formel appelé ***perceptron***.

* **1970s kO***Analyse des données* et *exploratory data analysis*

Les années 70 connaissent une remise en cause des modèles statistiques très contraignants car basés sur des hypothèses de nature probabiliste difficiles à vérifier, voire contredites. La plus grande diffusion des ordinateurs permet une approche exploratoire multidimensionnelle de l'*analyse des données*, notamment avec l'**analyse en composantes principales** (ACP), basée sur des considérations géométriques au lieu de probabilistes. De son côté, l'IA développe des procédures de raisonnement automatique associant, dans un *système expert,* bases de connaissances constituées de règles logiques et moteur d'inférence.

* **1980s MO***IA, Réseaux de neurones, Statistique fonctionnelle*

Les années 80 connaissent des développements plus méthodologiques avec la possibilité d'estimer des modèles statistiques non plus paramétriques mais fonctionnels, c'est-à-dire de grande dimension, et d'entraîner des réseaux de neurones relativement complexes. C'est l'apparition de l'**algorithme de *rétropropagation du gradient***. L'intelligence artificielle de cette période abandonne alors les systèmes experts aux temps d'exécution rédhibitoires (NP complets) au profit des **réseaux de neurones**. L'approche symbolique est supplantée par l'approche connexionniste.

* **1990s GO***Data mining et données préacquises*

Dans les années 90, la fouille de données, ou ***data mining***, fait son apparition avec principalement des applications en marketing quantitatif dans le tertiaire : banque, assurance, vente par correspondance. Ces grandes entreprises disposent déjà de bases de données clients conséquents à des fins comptables : des milliers de clients décrits par des dizaines de variables. L'objectif est de valoriser ces données en les utilisant pour améliorer la gestion de la relation client ou *GRC* : scores d'appétence pour des campagnes publicitaires, risque de crédit.

Pour ce faire, des suites logicielles commercialisées associent des requêtes d'extraction dans des bases de données, des outils exploratoires et de classification non supervisée, des modèles statistiques comme la régression logistique ou des arbres de décision, les premiers algorithmes d'apprentissage supervisé comme les réseaux de neurones...

L'ensemble de ces capacités de gestion, traitement et analyse des données intégrées dans une même suite logicielle devient le ***data mining***.

En fait, deux choses sont nouvelles :  
- l'intégration dans une même suite logicielle ;  
- et surtout le fait que les données ne soient plus issues d'une planification expérimentale comme en statistique inférentielle. Les données sont *préalables* à l'analyse, acquises pour d'autres finalités, par exemple comptables, et il faut faire avec...

C'est, pour le statisticien, devenu un prospecteur de données, un premier *changement de paradigme*:**ne plus pouvoir planifier l'expérience**.

* **2000s TO***Apprentissage statistique, Bio-informatique :  p >> n*

Le début du siècle a connu de profondes ruptures dans les biotechnologies à la suite du premier séquençage du génome. Pour chaque échantillon biologique, ce sont maintenant des milliers voire des millions d'informations qui sont observables ; des occurrences de millions de mutations possibles sur le génome, des expressions de dizaines de milliers de gènes, des expressions de protéines, de métabolites... En conséquence, les ensembles de données à étudier présentent beaucoup plus de colonnes, variables ou features que de lignes, échantillons ou instances ; pp**, le nombre de variables, est beaucoup plus grand que n, la taille de l'échantillon**. Ceci crée une situation d'indétermination qui a poussé à des développements méthodologiques et algorithmiques originaux : modèles et algorithmes parcimonieux ou *sparse*.

C'est un deuxième changement de paradigme pour le **statisticien devenu bio-informaticien**.

À cette occasion apparaissent ou réapparaissent des algorithmes dits *d'apprentissage statistique* (statistical learning) : boosting, support vector machine, random forest, réseaux de neurones..., sous-ensemble de l'apprentissage automatique, lui-même sous-ensemble de l'intelligence artificielle.

* **2010s PO***Grosses Data, Science des données et IA p et n très grands*

Plus récemment, avec l'avènement des réseaux sociaux, le succès planétaire de Google et des autres GAFAM, le volume des données, c'est-à-dire le nombre de clients, d'échantillons, et la taille des bases de données explosent ; **la fouille devient science des données**. Leur volume associé à la puissance de calcul autorise l'entraînement d'algorithmes d'apprentissage statistique excessivement complexes, dont le *deep learning* avec des millions de paramètres ou poids à estimer. La convergence entre données massives, puissance de calcul et algorithmes d'apprentissage conduit à des succès retentissants en reconnaissance d'images, traduction automatique ou encore jeu de go et véhicules autonomes... tout ceci concourt au succès de l'IA propulsée par un battage médiatique considérable.

Le troisième changement de paradigme concerne à la fois statisticiens, informaticiens et mathématiciens, devenus depuis 2008 *data scientists*, car le traitement de ces données massives nécessite certes de la **puissance de calcul** sur des systèmes de fichiers distribués comme *Hadoop*, mais aussi de **nouvelles approches**pour la résolution des problèmes d'optimisation afférents.

Cet historique met en exergue trois changements de paradigme méthodologique permis par des disruptions technologiques. Mais, fondamentalement, les nouvelles appellations, *data mining, big data, data science, deep learning*, *IA*, tiennent plus du battage médiatique que de la nécessité d'identifier une *nouvelle science*; **la science des données comme l'IA ne sont pas des nouvelles sciences**. Un *data scientist,* c'est d'abord *une équipe* pluridisciplinaire associant compétences en statistique, informatique et mathématiques.

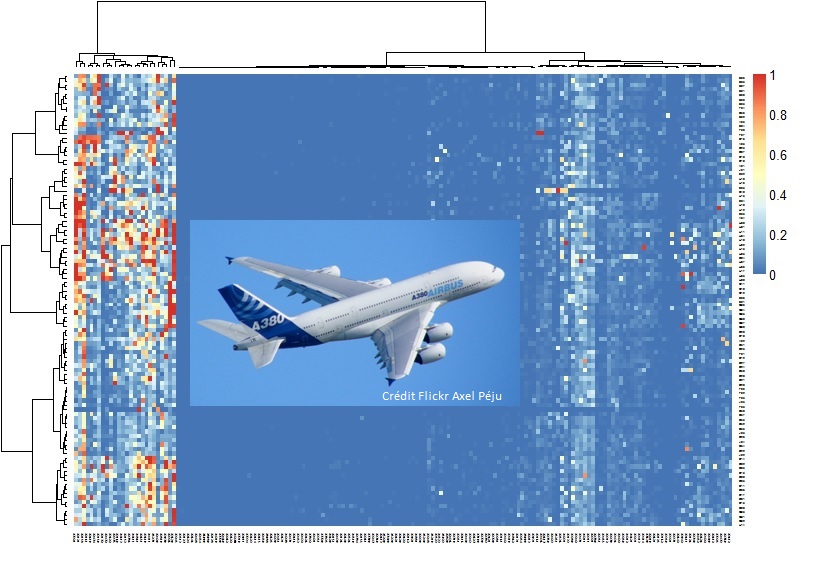
Méfiez-vous de ce battage médiatique et surtout des faux espoirs qu'il fait miroiter. Mêmes assemblées, toutes ces compétences ne peuvent rien avec des données pourries.;)

***Data Scientist (n)***: "*Person who is better at statistics than any software engineer and better at software than any statistician* " (J. Wills, *Cloudera*).

Traduction :  un Data scientist est une personne qui est meilleure en statistique que n'importe quel ingénieur en logiciel et meilleure en logiciel que n'importe quel statisticien

**Algorithme et décision automatique**

Voici quelques exemples illustratifs d'applications au quotidien de la science des données ou, plus précisément, d'algorithmes d'IA ou apprentissage automatique :

* le choix d'un traitement médical, d'une action commerciale, d'une action de maintenance préventive, d'accorder ou non un crédit, de surveiller un individu... toutes les décisions qui en découlent sont la conséquence d'une prévision ;
* la prévision du risque ou de la probabilité de diagnostic d'une maladie, le risque de rupture d'un contrat par un client, qui est le score d'attrition ou *churn*, le risque de défaillance d'un système mécanique, de défaut de paiement d'un client ou encore de radicalisation d'un individu... Les exemples sont très nombreux et envahissent notre quotidien à la suite de sa *datafication*.
* Visualisation et classification non supervisée avant analyse prédictive d'un recueil de messages d'incidents sur une flotte d'appareils

Le graphique ci-dessus est une visualisation avant analyse prédictive d'un recueil durant 6 mois de 700 000 messages d'incidents sur une flotte de 139 appareils. En ligne, les appareils ordonnés par une classification ascendante hiérarchique, en colonne les types d'incidents également ordonnés. La fausse couleur illustre un taux d'incidents.

Ces prévisions de risques ou scores, par exemple de crédit, sont produits par des **algorithmes d'apprentissage supervisé**, après entraînement sur une base de données où sont connus le comportement bancaire des clients et l'observation de bon remboursement ou non d'un emprunt.

# Identifiez les étapes de la science des données

Pour introduire plus précisément la science des données, décrivons schématiquement les deux principales phases de la démarche conduisant de l'acquisition des données à l'intégration d'une IA dans un objet connecté.

**Préparation des données & feature engineering**

La première phase constitue le prétraitement des données ou *data munging*, de l'extraction à l'exploration des données.

* Acquérir, archiver, extraire, mettre en forme des données sont des tâches de ***data management***en lien avec un gestionnaire de base de données incluant un langage de requête comme SQL ou *not only SQL*, associé au gestionnaire de données distribuées *Hadoop.*
* Il est important ensuite de s'assurer de l'**intégrité des données**, de leur **cohérence**. Cette étape exploratoire avec des outils élémentaires est un préalable important pour détecter des valeurs atypiques, éventuellement des erreurs, gérer les données manquantes par suppression des observations ou imputation de ces valeurs.  
  Cette étape permet également d'analyser la structure des données, leurs sources de variabilité et si possible de la représenter (ACP). Ces premiers résultats permettent d'apporter des réponses aux questions : faut-il transformer les données, calculer de nouvelles variables ou caractéristiques *feature engineering*?
* ***Garbage in,  garbage out***. Mêmes rudimentaires d'un point de vue méthodologique, ces étapes occupent la majeure partie du temps de l'analyse : environ 80 %. Enfin, insistons lourdement : la qualité des données recueillies, leur représentativité par rapport à la question posée, sont fondamentales afin d'obtenir des prévisions robustes, généralisables à d'autres données, c'est-à-dire, d'un point de vue statistique, non biaisées et de faible variance. La qualité des données est essentielle ; aussi performants que soient les algorithmes d'apprentissage, ils ne peuvent rien avec des données pourries !  
  Cette étape nécessite, pour être efficace et pertinente, des compétences et savoir-faire en statistique exploratoire multidimensionnelle.

**Apprentissage supervisé**

Une fois qu'une base de données d'entraînement fiable et représentative a été constituée, la phase d'apprentissage proprement dite est généralement bien définie et suit schématiquement la structure suivante.

* *Tirage aléatoire des échantillons d'apprentissage et de test.*

Commencer par **construire**, de façon aléatoire, **deux sous-échantillons** : le premier d'apprentissage, le second de test. Le premier sert à estimer les modèles ou faire apprendre les algorithmes, le deuxième n'est utilisé que pour en *prédire la variable cible Y afin*, comme les vraies valeurs ou les vraies classes de Y sont connues, de pouvoir estimer sans biais l'erreur de prévision.

Il faut distinguer l'erreur d'ajustement du modèle calculée sur l'échantillon d'apprentissage (erreur généralement biaisée car optimiste), de l'estimation de l'erreur de prévision calculée sur l'échantillon test (non biaisée car ces observations n'ont pas participé à l'estimation des paramètres du modèle ou à l'entraînement de l'algorithme).

* Pour chaque modèle ou algorithme considéré,

**itérer** ensuite les étapes suivantes pour chaque famille d'algorithme. Citons quelques possibilités : régression, SVM, réseaux de neurones, *boosting, random forest*, des plus utilisés parmi une grande farandole d'algorithmes disponibles.

1. ***Première étape*:** estimation du modèle en fonction des valeurs de certains hyperparamètres qui contrôlent la complexité du modèle, à savoir sa flexibilité, ou capacité à s'ajuster finement aux données.
2. ***Deuxième étape*:** optimisation, généralement en minimisant une estimation de l'erreur de prévision par validation croisée, de ce ou ces hyperparamètres : le nombre de variables dans un modèle, le nombre de feuilles dans un arbre, le nombre de neurones d'un réseau, la pénalisation des SVM...
3. ***Troisième étape*:** calcul la prévision de l'échantillon test pour le modèle "optimal" courant, afin,
4. ***Quatrième étape* :** d'estimer de l'erreur de prévision.

* On obtient ainsi une erreur par méthode d'apprentissage considérée ; il suffit alors de **choisir** la meilleure méthode ou le meilleur algorithme : celui qui minimise l'erreur de prévision, mais éventuellement en tenant compte de la facilité d'interpréter le modèle. Une fois l'algorithme sélectionné, il est entraîné une dernière fois sur l'ensemble des données afin de constituer l'IA embarquée dans l'objet connecté.

**Attention**, ce cours consacré à un cas d'usage ne propose pas une étude exhaustive de l'ensemble des algorithmes d'apprentissage ; seuls ceux nécessaires à la réalisation des objectifs sont brièvement décrits. Une initiation systématique aux autres algorithmes est proposée dans les tutoriels du dépôt  [**https://github.com/OpenClassrooms-Student-Center**](https://github.com/OpenClassrooms-Student-Center/ML-4-IoT)

# Préparez-vous pour la suite de ce cours

### 

### Objectif

Les objets connectés produisent des masses considérables de données.

Mais, que peut-on en faire ?

La science des données apporte les **outils de valorisation** permettant de transformer des données en prises de décision et d'introduire de l'IA dans l'objet connecté. Dans notre exemple, il s'agit de l'identification d'une activité humaine.

Dans ce cours, vous allez comprendre les possibilités offertes par la science des données lorsqu'elle est mise en œuvre sur des signaux issus de capteurs physiques.

Cet objectif est atteint au cours d'une progression en trois étapes.

### Contenu du cours

#### Partie 2

Cette partie aborde la **visualisation des données** afin d'en comprendre les structures et donc les propriétés ou caractéristiques permettant d'atteindre l'objectif d'identification des classes. Cette étape montre les bonnes propriétés des données transformées selon des méthodes et approches classiques en traitement du signal (feature engineering), alors que les signaux bruts apparaissent très confus.

#### **Partie 3**

En conséquence, dans la partie suivante, des **algorithmes rudimentaires**, à savoir linéaires, sont présentés. Ils conduisent à une très bonne identification des activités à partir des données transformées. La solution obtenue est satisfaisante en termes de décision mais pas en termes de consommation d'énergie. Calculer en permanence et en temps réel les transformations des signaux est bien trop énergivore pour la batterie d'un objet autonome.

#### Partie 4

C'est pourquoi la dernière partie se focalise sur **les signaux bruts**. Dans ce cas, seul un algorithme d'apprentissage profond ou deep learning intégrant des couches dites convolutionnelles permet d'attendre des qualités de résultats comparables. C'est la motivation d'une introduction élémentaire au deep learning.

### Activités préalables

Avant de vous lancer dans cette aventure où vous allez pratiquer ces algorithmes, il est nécessaire d'installer l'environnement informatique nécessaire. Il est simple de charger la distribution [Anaconda](https://www.anaconda.com/download) de Python 3 sous Linux, Mac OS ou Windows, puis d'installer KerasKeras pour le deep learning avec la commande  condaconda   installinstall   keraskeras .

Si vous voulez pleinement profiter des apprentissages de ce cours afin de pouvoir en réinvestir les compétences sur d'autres données, dans d'autres situations, certaines compétences préalables sont nécessaires en programmation Python et en statistique. Celles-ci sont regroupées dans 3 tutoriels :

* une introduction à Python, ou plutôt une sélection des commandes Python utiles au statisticien ou data scientist, est disponible dans le tutoriel  [ML4IoT-Intro-Python.ipynb](https://s3-eu-west-1.amazonaws.com/course.oc-static.com/courses/5919236/ML4IoT-Intro-Python.ipynb)  ;
* de même, les commandes indispensables à la préparation des données ou data munging et utilisant la librairie pandaspandas  sont l'objet du 2ee tutoriel[ML4ioT-Intro-Pandas.ipynb](https://s3-eu-west-1.amazonaws.com/course.oc-static.com/courses/5919236/ML4IoT-Intro-Pandas.ipynb)  ;
* enfin, comme la science des données débute par des éléments de statistique, quelques rappels de statistique descriptive élémentaire sont accessibles dans le début du 3ee tutoriel[ML4IoT-Tutorial-Ozone.ipynb](https://s3-eu-west-1.amazonaws.com/course.oc-static.com/courses/5919236/ML4IoT-Tutorial-Ozone.ipynb) qui servira également de fil rouge pour introduire les autres méthodes utilisées.

Tous les tutoriels de ce cours sont des notebooks Jupyter ou calepins très facilement exécutables une fois chargés et Python installé. Ils ont tous été développés sous Ubuntu 16.04 Xenial, mais sont compatibles avec Windows ou Mac OS.

**LISEZ-MOI !**

L'exécution de ces tutoriels nécessite un environnement de travail adapté. Deux solutions sont possibles :

* utilisation rudimentaire de Google Colab ;
* utilisation experte des notebooks d'un dépôt GitHub après installation de Python et clonage du dépôt.

**Solution élémentaire avec Google Colab**

* Une seule fois :
  + installer puis exécuter le navigateur Chrome ;
  + ajouter à Chrome l'extension "open notebook in google colab".
* Se connecter avec son compte Gmail.
* Lancer Chrome.
* Ouvrir l'URL de [**Google Colab**](https://colab.research.google.com/notebooks/welcome.ipynb).
* Ouvrir dans Chrome le tutoriel ou notebook du dépôt à exécuter à partir de son URL.
* Cliquer sur l’icône d\*"open notebook in google colab"\*.
* Suivre le tutoriel en exécutant chaque cellule.

**Solution experte**

* Une seule fois :
  + installer la dernière version de Python 3 avec la distribution Anaconda ;
  + charger ce dépôt, le décompresser ou le cloner.
* Pour chaque tutoriel :
  + lancer la commande jupyter notebook qui ouvre une fenêtre dans le navigateur par défaut ;
  + aller dans le bon répertoire cloné ou chargé ;
  + ouvrir le notebook ;
  + dérouler le tutoriel.

**Attention** : l'utilisation de Google Colab nécessite une connexion Internet correcte contrairement à une exécution en local, une fois le dépôt GitHub chargé ou cloné. De plus, les fichiers de données sont nécessairement chargés sur Internet pour chaque tutoriel.

Finalement, les compétences en informatique ne sont pas complètement indispensables, vous pouvez aussi suivre ce cours en vous focalisant sur les résultats, leur interprétation, sans détailler la mise en œuvre informatique et sans maîtriser les subtilités du langage Python.

Bon courage pour cette aventure !

### Références

**Anguita D., Ghio A., Oneto L., Parra X., Reyes-Ortiz J.L.** (2013). A Public Domain Dataset for Human Activity Recognition Using Smartphones, ESANN 2013 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, 437-442.

# Identifiez les enjeux et objectifs de ce cours

Bravo ! Vous avez réussi cet exercice !

# Compétences évaluées

* Mettre en place les outils nécessaires à la science des données

### Question 1

**Ce cours s'intéresse au porteur d'un smartphone. L'objectif de ce cours est de :**

*Attention, plusieurs réponses sont possibles.*

* + 

capter les signaux du smartphone

* + 

analyser les signaux

* + 

débruiter les signaux

* + 

identifier le porteur

* + 

identifier l'activité du porteur

*Reportez-vous à l'introduction du cours.*

### Question 2

**De quelles fonctionnalités d'un smartphone sont issus les signaux étudiés ?**

*Attention, plusieurs réponses sont possibles.*

* + 

Haut-parleur

* + 

Gyroscope

* + 

GPS

* + 

Accéléromètre

* + 

Micro

*Retrouvez ces éléments dans le chapitre "Comprenez enjeux et objectifs du cours".*

### Question 3

**Quelles propriétés caractérisent les signaux étudiés ?**

*Attention, plusieurs réponses sont possibles.*

* + 

Bruités

* + 

Réguliers

* + 

Périodiques

* + 

Synchronisés

* + 

Cahotiques

*Reportez-vous au chapitre "Comprenez enjeux et objectifs du cours".*

### Question 4

**Historiquement, on peut dire que :**

*Attention, plusieurs réponses sont possibles.*

* + 

la statistique précède le data mining

* + 

l'IA précède la bio-informatique

* + 

La science des données précède l'IA

* + 

les réseaux de neurones précèdent le deep learning

*L'explication se trouve dans le chapitre "Découvrez la science des données et l'IA".*

### Question 5

**De façon logique ou fonctionnelle :**

*Attention, plusieurs réponses sont possibles.*

* + 

l'apprentissage statistique est contenu dans l'apprentissage automatique

* + 

les systèmes experts sont aussi du deep learning

* + 

l'IA est contenue dans l'apprentissage machine

* + 

le deep learning est aussi de l'apprentissage automatique

*Voir chapitre "Découvrez la science des données et l'IA".*

### Question 6

**L'acquisition des données :**

* + 

doit obligatoirement être planifiée en data mining ou prospection de données

* + 

doit conduire à observer un nombre de variables ou features plus petit que la taille de l'échantillon

* + 

est suivie d'une phase essentielle de préparation, transformation, exploration des données

*Cette notion est abordée dans le chapitre "Identifiez les étapes de la science des données".*

### Question 7

**Le surajustement ou surapprentissage signifie que le modèle ou l'algorithme :**

*Attention, plusieurs réponses sont possibles.*

* + 

ajuste trop bien l'échantillon d'apprentissage

* + 

ajuste trop bien l'échantillon test

* + 

dégrade la qualité de prévision de l'échantillon test

*Reportez-vous au chapitre "Comprenez enjeux et objectifs du cours".*

### Question 8

**Quelles sont les caractéristiques du langage Python ?**

*Attention, plusieurs réponses sont possibles.*

* + 

Un langage interprété

* + 

Un langage compilé

* + 

Un langage matriciel avec la librairie  pandaspandas

* + 

Un langage objet

*Voir le tutoriel d'introduction à Python dans le chapitre "Contenu du cours, activités préalables et références".*

### Question 9

**Que signifie**DataFrameDataFrame**dans la librairie**PandasPandas**?**

*Attention, plusieurs réponses sont possibles.*

* + 

Un ensemble de données homogènes

* + 

Une classe analogue au type dataframedataframe  de R

* + 

Une liste d'objets de type seriesseries partageant le même index

* + 

Une liste d'objets de type dictionnaire

*Voir le tutoriel d'introduction à Pandas 2.2 dans le chapitre "Contenu du cours, activités préalables et références".*

### Question 10

**Dans l'analyse des données ozone, que montre la matrice des nuages de points scatter plot matrix ?**

*Attention, plusieurs réponses sont possibles.*

* + 

L'absence de liaisons non linéaires

* + 

De faibles liaisons linéaires entre les variables

* + 

De fortes liaisons linéaires entre les variables

* + 

La présence de relations causales

*Ce graphique renseigne dans ce cas sur l'absence de liaison non linéaire rendant pertinente une modélisation linéaire. Corrélation ne signifie pas causalité.*

# Prenez en charge les données

**Data munging**

La première partie insiste particulièrement sur le rôle crucial de la **préparation des données** : détection d'erreurs ou de valeurs atypiques, analyse des distributions et transformation des variables, imputations de données manquantes...

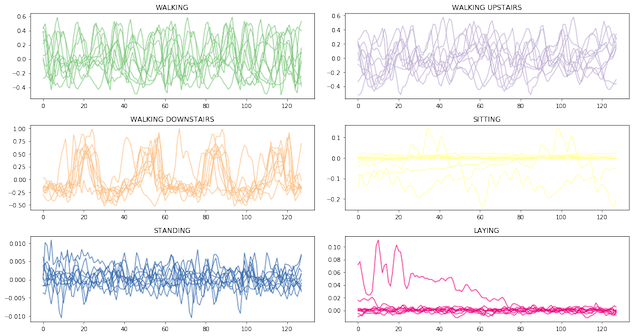
Les données fournies à la suite de l'expérimentation (Anguita et al. 2013) des différentes activités enregistrées par un smartphone sont **"propres"** ou alors elles ont été déjà nettoyées des scories, erreur de manipulations, pannes et autres sources de défaillance.

Il faut prendre conscience que, dans beaucoup de situations réelles, ce n'est pas le cas et qu'il est indispensable, en général, de passer beaucoup de temps pour l'obtention d'une base de données qui pourra être analysée avec pertinence.

**Visualisez des signaux bruts**

La première opération à réaliser consiste à **visualiser les données**. La représentation simultanée de tous les signaux vus dans la première partie est bien trop confuse pour appréhender leur structure et les questions qu'elle soulève.

Aussi, la représentation ci-dessous ne concerne qu'un seul type de signal : **accélération en x**, pour chacune des activités (de gauche à droite et de haut en bas : marcher, monter ou descendre un escalier, être couché, assis ou debout).



Accélération en x pour chacune des activités

Intuitivement, il est assez clair que certaines activités (*laying)*, produisent des signaux très spécifiques, donc très différents de ceux des autres activités. En revanche, les différents types de marches produisent des signaux similaires avec une forte composante périodique.

La principale question concerne la façon de mesurer une *distance* entre deux signaux temporels, c'est-à-dire deux courbes ou fonctions. En effet, la distance usuelle entre deux fonctions, déduite de la norme  des fonctions de carré intégrable, est l'intégrale des carrés des écarts entre 2 courbes. Comme les courbes sont **discrétisées**, il s'agit tout simplement de la somme des carrés des écarts ou plutôt de la racine carrée de cette quantité pour en faire une distance.

Le principal problème observé sur ces courbes concerne leur absence de synchronisation ou, c'est équivalent, leur déphasage ou décalage temporel. Au sens de la distance , deux signaux correspondant à la même activité peuvent être très proches ou très éloignés par le simple fait du **déphasage**.

C'est la principale raison pour laquelle vouloir analyser directement ces signaux temporels bruts, avec des distances euclidiennes classiques, est voué à l'échec.

Notons par ailleurs des enregistrements atypiques, par exemple dans l'activité **"couché"** ; certains signaux laissent penser que le porteur était en train de se coucher, même chose pour l'activité **"assis"**. Ces activités spécifiques seront évidemment difficiles à identifier correctement.

**Transformez les signaux**

Pour dépasser les questions d'absence de synchronisation des signaux, des compétences en traitement du signal sont mises à profit pour calculer toute une batterie de nouvelles **variables** ou **caractéristiques** (*features*) sur ces signaux.

Les détails de cette étape sont décrits par Anguita et al. (2013). Voici une liste des principales fonctions calculées sur chaque **signal** ou paire de **signaux** de chaque activité :

* valeur moyenne ;
* écart-type ;
* valeur absolue médiane ;
* plus grande valeur ;
* plus petite valeur ;
* zone d'amplitude du signal ;
* somme des carrés moyens ;
* interquartile ;
* entropie ;
* coefficient d'autorégression ;
* coefficient de corrélation ;
* composante de plus grande fréquence ;
* moyenne pondérée des fréquences ;
* coefficient d'asymétrie des fréquences ;
* kurtosis des fréquences ;
* énergie dans une bande de fréquences ;
* angle entre deux vecteurs...

Certaines transformations sont calculées sur une base de **décompositions de Fourier** comme l'intensité du signal dans certaines bandes de fréquences ; elles fournissent justement des quantités qui ne dépendent pas des décalages temporels des expérimentations.

Finalement, calculées sur les 9 types de signaux et leurs combinaisons deux à deux (corrélations), ce sont **p=561** variables qui sont considérées par la suite.

Aborder des données d'une telle complexité multidimensionnelle nécessite des moyens appropriés. C'est le domaine d'application privilégié de l'**analyse en composantes principales** (ACP).

# Comprenez l'analyse en composantes principales

### 

### Objectifs

Le meilleur moyen d'explorer des données complexes consiste à construire des **représentations graphiques** appropriées. Il est élémentaire et très utile de visualiser la nature de la liaison entre deux variables quantitatives avec un nuage de nn points ; c'est rappelé dans le tutoriel de statistique descriptive ( ML4IoT−Tutorial−OzoneML4IoT−Tutorial−Ozone ).

Lorsque ce sont **p** variables avec **p>3** qui sont observées, il reste possible de construire une **matrice de nuages de points** croisant toutes les variables 2 à 2 si **p** n'est pas trop grand mais l'analyse en composantes principales (ACP) apporte une solution plus satisfaisante surtout avec **p** grand.

Une [**vignette Wikistat**](http://wikistat.fr/pdf/st-m-explo-acp) décrit plus en détail les fondements mathématiques de cette méthode exploratoire multidimensionnelle très utilisée.

Considérons **p** variables quantitatives observées sur individus ou unités statistiques. L'objectif de l'ACP est de construire une double représentation graphique :

* représentation plane, ou de petite dimension, du nuage de points des **n** individus en respectant au mieux leurs positions respectives, leurs distances deux à deux ;
* représentation des **p** variables illustrant la structure des corrélations linéaires entre celles-ci.

D'un point de vue statistique, l'**ACP** est la recherche de nouvelles variables, combinaisons linéaires des variables initiales et orthogonales deux à deux, de sorte que la variance de la première combinaison soit la plus grande puis celle de la deuxième, orthogonale à la précédente, soit à nouveau de plus grande variance, etc.

Il est aussi possible de proposer **une analogie physique**. Considérons l'ensemble des **n** points ou individus dans l'espace . Ils constituent un solide dont les axes d'inertie sont déterminés par les **vecteurs propres** de la matrice d'inertie de ce solide. Ces vecteurs propres définissent les axes de plus grande dispersion des points du nuage ou solide.

Mathématiquement, l'**ACP** est un simple changement de base : passer d'une représentation dans la base canonique des variables initiales à une représentation dans la base des facteurs définis par les vecteurs propres de la matrice des variances-covariances (inerties) ou de celle des corrélations.

### Exemple jouet

#### Les données

Une présentation très élémentaire de cette démarche est proposée sur un exemple jouet de données. Considérons les notes (de 0 à 20) obtenues par 9 élèves dans 4 disciplines (mathématiques, physique, français, anglais) :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | MATH | PHYS | FRAN | ANGL |
| Jean | 6.00 | 6.00 | 5.00 | 5.50 |
| Alan | 8.00 | 8.00 | 8.00 | 8.00 |
| Anni | 6.00 | 7.00 | 11.00 | 9.50 |
| Moni | 14.50 | 14.50 | 15.50 | 15.00 |
| Didi | 14.00 | 14.00 | 12.00 | 12.50 |
| Andr | 11.00 | 10.00 | 5.50 | 7.00 |
| Pier | 5.50 | 7.00 | 14.00 | 11.50 |
| Brig | 13.00 | 12.50 | 8.50 | 9.50 |
| Evel | 9.00 | 9.50 | 12.50 | 12.00 |

Il est classique d'analyser séparément chacune de ces 4 variables, soit en faisant un **graphique**, soit en calculant des **résumés numériques**. Les **liaisons entre 2 variables** (par exemple mathématiques et français), sont illustrées en faisant un graphique du type nuage de points et évaluées en calculant leur **coefficient de corrélation linéaire**.

Mais comment faire une étude simultanée des 4 variables, ne serait-ce qu'en réalisant un graphique ?

La difficulté vient de ce que les individus (les élèves) ne sont plus représentés dans un plan, espace de dimension 2, mais dans un **espace de dimension 4**, chacun étant caractérisé par les 4 notes qu'il a obtenues.  
  
L'objectif de l'analyse en composantes principales est de projeter les points sur un espace de dimension réduite (par exemple, ici, 2) en déformant **le moins possible** la réalité, c'est-à-dire les positions respectives des élèves entre eux. Il s'agit donc d'obtenir le **résumé le plus pertinent** des données initiales.

#### Descriptions uni- et bivariée

Tout logiciel statistique fournit la moyenne, l'écart-type, le minimum et le maximum de chaque variable. Il s'agit donc, pour l'instant, d'[études univariées](http://wikistat.fr/pdf/st-l-des-uni).

**Statistiques élémentaires**

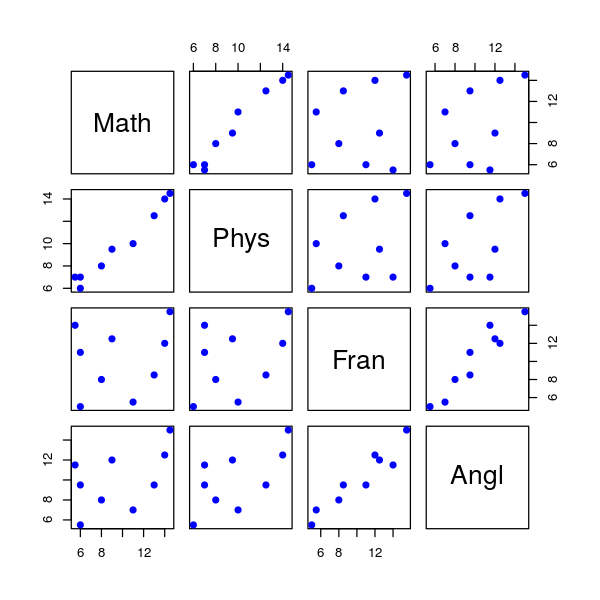
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Variable | Moyenne | Écart-type | Minimum | Maximum |
| MATH | 9.67 | 3.37 | 5.50 | 14.50 |
| PHYS | 9.83 | 2.99 | 6.00 | 14.50 |
| FRAN | 10.22 | 3.47 | 5.00 | 15.50 |
| ANGL | 10.06 | 2.81 | 5.50 | 15.00 |

Notons au passage la grande homogénéité des 4 variables considérées : même ordre de grandeur pour les moyennes, les écarts-types, les minima et les maxima.  
  
Le tableau suivant est la **matrice des corrélations**. Elle donne les coefficients de corrélation linéaire des variables prises deux à deux. C'est une succession d'[analyses bivariées](http://wikistat.fr/pdf/st-l-des-bi), constituant un premier pas vers l'**analyse multivariée**.

**Coefficients de corrélation**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | MATH | PHYS | FRAN | ANGL |
| MATH | 1.00 | 0.98 | 0.23 | 0.51 |
| PHYS | 0.98 | 1.00 | 0.40 | 0.65 |
| FRAN | 0.23 | 0.40 | 1.00 | 0.95 |
| ANGL | 0.51 | 0.65 | 0.95 | 1.00 |

Remarquons que toutes les corrélations linéaires sont positives, ce qui signifie que toutes les variables varient, en moyenne, dans le même sens, certaines étant **très fortes** (0.98 et 0.95), d'autres **moyennes** (0.65 et 0.51), d'autres enfin plutôt **faibles** (0.40 et 0.23).



Matrice des nuages de points correspondante en considérant toutes les variables deux à deux

La figure ci-dessus fournit la matrice des nuages de points en considérant toutes les variables **deux à deux**. Le principal objectif de cette représentation est de s'assurer qu'il n'existe pas de liaison **non linéaire** entre les variables. En effet, une telle liaison serait négligée par des indicateurs (corrélation) ou toute analyse linéaire comme l'ACP ou la régression.

Dans le cas contraire, une transformation de certaines variables (fonction puissance, log...) suffit généralement à linéariser les relations. C'est élémentaire mais indispensable à la prise en compte de liaisons non linéaires.

#### Décomposition spectrale de la matrice des covariances

**Résultats numériques**

Continuons l'analyse par l'étude de la **matrice des variances-covariances**. La diagonale de cette matrice fournit les variances des 4 variables considérées.

Matrice des variances-covariances :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | MATH | PHYS | FRAN | ANGL |
| MATH | 11.39 | 9.92 | 2.66 | 4.82 |
| PHYS | 9.92 | 8.94 | 4.12 | 5.48 |
| FRAN | 2.66 | 4.12 | 12.06 | 9.29 |
| ANGL | 4.82 | 5.48 | 9.29 | 7.91 |

Le tableau ci-dessous fournit les **valeurs propres** de la matrice des variances-covariances. Notez que la somme des valeurs propres est aussi la somme des variances ou trace de la matrice des covariances. D'un point de vue théorique, la trace d'un **endomorphisme** ne dépend pas de la base de représentation. Elle est identique dans la base canonique : matrice des covariances, ou celle des vecteurs propres : matrice diagonale des valeurs propres.

Valeurs propres ; variances expliquées :

|  |  |  |  |
| --- | --- | --- | --- |
| FACTEUR | VAL. PR. | PCT. VAR. | PCT. CUM. |
| 1 | 28.23 | 0.70 | 0.70 |
| 2 | 12.03 | 0.30 | 1.00 |
| 3 | 0.03 | 0.00 | 1.00 |
| 4 | 0.01 | 0.00 | 1.00 |
|  | **40.30** | **1.00** |  |

**Interprétation statistique**

Chaque ligne du tableau ci-dessus correspond à une variable virtuelle (facteur ou variable principale) dont la colonne VAL. PR. (valeur propre) fournit la variance. Un facteur ou variable principale est une combinaison linéaire des variables initiales, dans laquelle les coefficients sont donnés par les coordonnées des vecteurs propres (changement de base).

Rappelons que l'**ACP** peut être définie comme la recherche des **combinaisons linéaires de plus grande variance** des **variables initiales** (les valeurs propres).

La colonne **PCT**. **VAR**., ou **pourcentage de variance**, correspond au pourcentage de variance de chaque ligne par rapport au total. La colonne PCT. CUM. représente le cumul de ces pourcentages en dimension 1, 2... Additionnons maintenant les variances des 4 variables initiales (diagonale de la matrice des variances-covariances) : 11.39+8.94+12.06+7.91=40.30   
La dispersion totale des individus considérés, en dimension 4, est ainsi égale à 40.30.

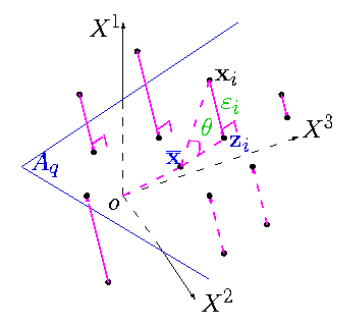
Additionnons par ailleurs les 4 valeurs propres obtenues : 28.23+12.03+0.03+0.01=40.30   
Le nuage de points en dimension 4 est toujours le même et sa dispersion globale n'a pas changé.  
Il s'agit d'un simple changement de base dans un espace vectoriel.

C'est la répartition de cette dispersion, selon les nouvelles variables de plus grande dispersion qui sont les facteurs ou encore **composantes principales**. Observer que les 2 premiers facteurs restituent à eux seuls la quasi-totalité de la dispersion du nuage, ce qui permet de négliger les 2 autres.

Par conséquent, les graphiques en dimension 2 présentés ci-dessous résument presque parfaitement la configuration réelle des données qui se trouvent en dimension 4. L'objectif de résumé pertinent des données en plus petite dimension est donc atteint.

#### **Interprétation géométrique**

Une autre interprétation est d'ordre **géométrique**. Chaque individu **xi** (resp. variable **xj**) est considéré comme un vecteur à p (resp. **n**) composantes dans un espace vectoriel. L'ACP est la recherche du meilleur plan (ou sous-espace affine) de projection **Aq** : le plus proche au sens des moindres carrés, pour obtenir la représentation la plus fidèle, ou la moins déformée, des individus (resp. des variables) dans un sous-espace de dimension réduite.



Représentation géométrique élémentaire en trois dimensions du principe d'une analyse en composantes principales

Sur ce graphique, **zi** est la projection orthogonale de **xi**, définie par le vecteur des valeurs , sur le plan **Aq** qui passe par le barycentre  du nuage des points.

#### **ACP réduite ou non**

Les sections précédentes mettent l'accent sur la matrice de variance et sa décomposition en éléments propres. Les variables de ces données fictives présentent de bonnes propriétés, elles sont de même unité, toutes des notes, et de variances homogènes.

Dans le cas contraire, **hétérogénéité des unités** ou des **variances**, il est important d'apporter une forme de **normalisation**. En effet, comme la variance dépend de l'unité choisie ou même si une ou des variables ont de très grandes variances par rapport aux autres variables, cela peut avoir un effet délétère sur l'intérêt de l'ACP. Une seule variable, celle évidemment de grande variance, peut à elle seule accaparer le premier axe au détriment de la compréhension globale des relations entre les variables.

C'est la raison pour laquelle, si les variables ne partagent pas la même unité ou si de toute façon les variances sont hétérogènes, il est vivement conseillé de **réduire** (standardiser) **les variables** en les divisant par leur écart-type. Toutes les variables sont alors sans unité et de variance 1, elles jouent le même rôle et leur structure de corrélation est mise en exergue.

Comme la matrice des variances de variables réduites est la matrice des corrélations, c'est elle qui est diagonalisée pour fournir les vecteurs propres et valeurs propres de somme **p**, le nombre de variables, car la diagonale de la matrice des corrélations est composée de 1s.

#### **Étude des variables**

**Résultats numériques**

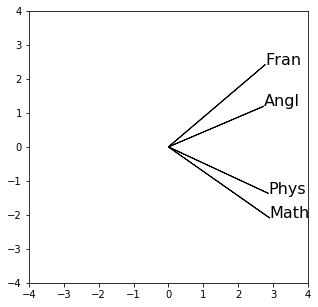
Un résultat important pour aider à l'interprétation est fourni par le tableau des **corrélations variables-facteurs**. Il s'agit des coefficients de corrélation linéaire entre les variables initiales et les nouvelles variables dites principales ou facteurs.

Ce sont ces corrélations qui vont permettre de donner une signification aux facteurs ou variables principales pour les interpréter.

Corrélations  variables-facteurs :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | F1 | F2 | F3 | F4 |
| MATH | 0.81 | -0.58 | 0.01 | -0.02 |
| PHYS | 0.90 | -0.43 | -0.03 | 0.02 |
| FRAN | 0.75 | 0.66 | -0.02 | -0.01 |
| ANGL | 0.91 | 0.40 | 0.05 | 0.01 |

Les deux premières colonnes de ce tableau permettent, tout d'abord, de réaliser le **graphique des variables** ci-dessous.



Représentation graphique des variables dans le premier plan factoriel

Noter que les deux dernières colonnes ne seront pas utilisées, puisque seulement deux dimensions sont nécessaires pour représenter les données.

**Interprétation**

Par construction, le cosinus de l'angle de deux vecteurs variables approche le **coefficient de corrélation** entre ces variables. Ainsi, on lit sur le graphique ci-dessus que le premier facteur est corrélé positivement, et assez fortement, avec chacune des 4 variables initiales : plus un élève obtient de bonnes notes dans chacune des 4 disciplines, plus il a un score élevé sur l'axe 1 ; réciproquement, plus ses notes sont  mauvaises, plus son score est négatif.

Le **premier facteur**, combinaison des notes avec approximativement les mêmes coefficients positifs, représente la note moyenne (centrée sur la moyenne de la classe) de chaque élève.

En ce qui concerne l'axe 2, il oppose d'une part le français et l'anglais (corrélations positives) et, d'autre part, les mathématiques et la physique (corrélations négatives).

Le **deuxième facteur** est approximativement la moyenne des notes littéraires déduite de la moyenne des notes scientifiques.

Cette interprétation aide à comprendre la représentation des individus.

#### Étude des individus

**Résultats numériques**

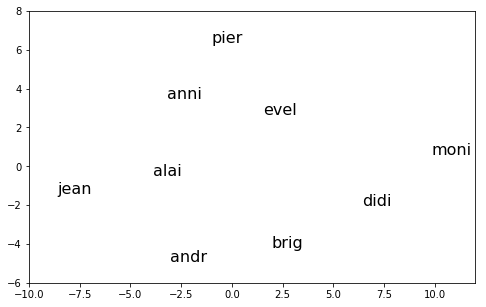
Le tableau ci-dessous contient tous les résultats importants sur les individus.

Coordonnées des individus et cosinus carrés

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | POIDS | PC1 | PC2 | COSCA1 | COSCA2 |
| jean | 0.11 | -8.61 | 1.41 | 0.97 | 0.03 |
| alan | 0.11 | -3.88 | 0.50 | 0.98 | 0.02 |
| anni | 0.11 | -3.21 | -3.47 | 0.46 | 0.54 |
| moni | 0.11 | 9.85 | -0.60 | 1.00 | 0.00 |
| didi | 0.11 | 6.41 | 2.05 | 0.91 | 0.09 |
| andr | 0.11 | -3.03 | 4.92 | 0.28 | 0.72 |
| pier | 0.11 | -1.03 | -6.38 | 0.03 | 0.97 |
| brig | 0.11 | 1.95 | 4.20 | 0.18 | 0.82 |
| evel | 0.11 | 1.55 | -2.63 | 0.25 | 0.73 |

On notera que chaque individu représente 1 élément sur 9, d'où un poids (une pondération) de  1/9=0.11, ce qui est fourni par la première colonne du tableau ci-dessus.

Les 2 colonnes suivantes fournissent les coordonnées des individus (les élèves) sur les deux premiers axes (les facteurs) et ont donc permis de réaliser le **graphique des individus** ci-dessous. Elles sont appelées composantes principales ou valeurs prises par les individus sur les variables principales, combinaisons linéaires des variables initiales.



Représentation graphique des individus dans le premier plan factoriel

**Interprétation**

L'axe 1 représente le résultat d'ensemble des élèves : leur score – ou coordonnée – sur l'axe 1 fournit le même classement que leur moyenne générale. Par ailleurs, l'élève "le plus haut" sur le graphique, celui qui a la coordonnée la plus élevée sur l'axe 2, est Pierre dont les résultats sont les plus contrastés en faveur des disciplines littéraires (14 et 11.5 contre 7 et 5.5).

C'est exactement le contraire pour André qui obtient la moyenne dans les disciplines scientifiques (11 et 10) mais des résultats très faibles dans les disciplines littéraires (7 et 5.5). Monique et Alain ont un score voisin de 0 sur l'axe 2 car ils ont des résultats très homogènes dans les 4 disciplines, mais à des niveaux très distincts, ce qu'a déjà révélé l'axe 1.

#### Compléments à l'interprétation

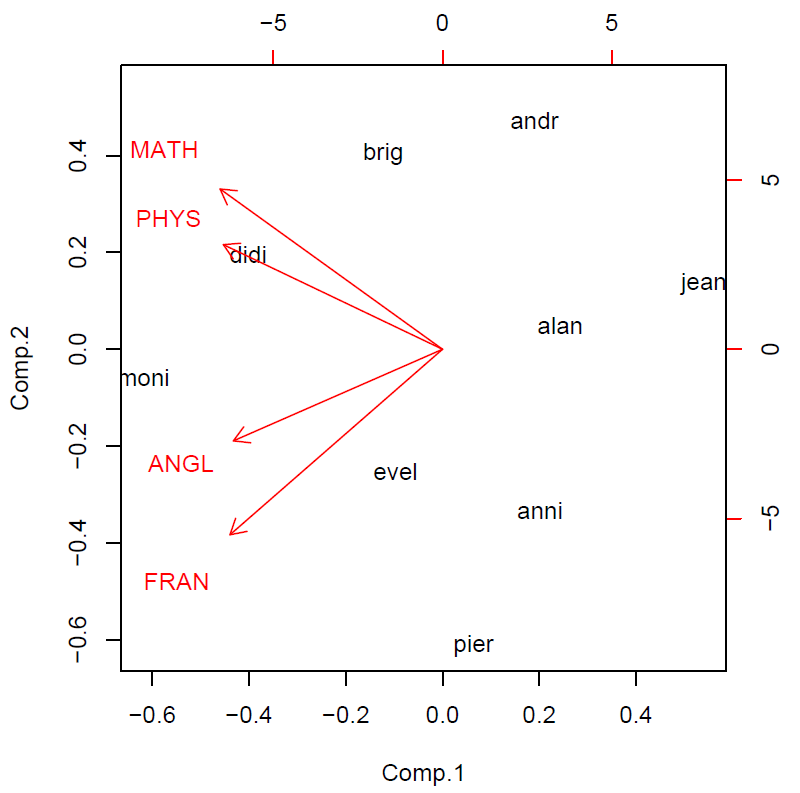
Les 2 dernières colonnes du tableau sont des cosinus carrés qui fournissent la **qualité de la représentation** de chaque individu sur chaque axe. Ces quantités s'additionnent axe par axe, de sorte que, en dimension 2, Évelyne est représentée à 98 % (0.25 + 0.73), tandis que les 8 autres individus le sont à 100 %.

Avec les données initiales, chaque individu (chaque élève) est représenté par un vecteur dans un espace de dimension 4 (les éléments – ou coordonnées – de ce vecteur sont les notes obtenues dans les 4 disciplines). Résumé en dimension 2, et donc représenté dans un plan, chaque individu est alors représenté par la projection du vecteur initial sur le plan. Le cosinus carré relativement aux deux premières dimensions (par exemple, pour Évelyne, 0.98 ou 98 %) est celui de l'angle formé par le vecteur initial et sa projection dans le plan.

Plus le vecteur initial est proche du plan, plus l'angle en question est petit et plus le cosinus, et son carré, sont proches de 1 (ou de 100 %) : la représentation est alors très bonne. Au contraire, plus le vecteur initial est loin du plan, plus l'angle en question est grand (proche de 90 degrés) et plus le cosinus, et son carré, sont proches de 0 : la représentation est alors très mauvaise.

#### Représentation simultanée

Un troisième type de représentation graphique associant individus et variables (biplot) est détaillé dans le document décrivant plus précisément l'[analyse en composantes principales](http://wikistat.fr/pdf/st-m-explo-acp). Ce graphe, représentant des vecteurs individus et variables appartenant à des espaces vectoriels différents, nécessite un développement plus détaillé pour en justifier la construction et l'interprétation.

Représentation simultanée (biplot) des variables et des individus dans le premier plan

Ce développement est basé sur la [décomposition en valeurs singulières](http://wikistat.fr/pdf/st-m-explo-alglin.pdf) de la matrice **X** des variables centrées. Géométriquement, le produit scalaire entre un vecteur variable et un vecteur individu fournit une approximation de la valeur de cette variable sur l'individu. Cela permet de comparer schématiquement la valeur prise par une variable sur un individu par rapport à la moyenne, le barycentre. Ainsi, Brigitte à une moyenne en maths supérieure à la moyenne de la classe mais inférieure à la moyenne de la classe en français. Monique a des notes plus grandes que la moyenne de la classe dans toutes les matières.

Cette représentation n'est possible que pour des jeux de données restreints : lorsque les dimensions sont trop importantes, elle devient **illisible**. Elle a été produite par la commande biplot de R plutôt qu'avec Python comme pour les autres graphiques. Notons que les signes des axes ont changé avec le changement de logiciel. Plus précisément, le signe d'un vecteur propre n'est pas une information pertinente et peut changer d'un logiciel à l'autre. Ce qui est important, c'est **la direction portée par le vecteur propre** et pas sa direction. L'interprétation est d'ailleurs identique.

### L'analyse factorielle discriminante, cas particuler d'ACP

Considérons la même situation que celle de l'ACP : **p** variables quantitatives **X**j observées sur **n** individus ou unités statistiques. Ajoutons l'observation d'une variable qualitative à **q** classes pour définir l'analyse factorielle discriminante (AFD). Comme en ACP, l'objectif est de fournir une meilleure représentation graphique en petite dimension mais en tenant compte de l'observation de la variable qualitative, donc des classes.

L'objectif devient alors la recherche de la représentation privilégiant le plus possible les différences entre les classes d'individus au détriment des différences au sein des classes.

#### Principes de l'AFD

Comme vu précédemment, l'ACP est basée sur la recherche de combinaisons linéaires de plus grande variance. L'AFD se focalise sur la variance interclasses, pour mettre en évidence les différences des classes tout en cherchant à minimiser les effets de la variance **intraclasse** qui disperse les individus au sein de chaque classe.

La justification de l'AFD est détaillée dans une [vignette de Wikistat](http://wikistat.fr/pdf/st-m-explo-afd.pdf). Il y est montré que l'AFD est un cas particulier optimal d'ACP permettant d'atteindre l'objectif de meilleure représentation des classes. L'AFD est finalement l'ACP des barycentres des classes afin de mettre en exergue la variance interclasses, mais en affectant l'espace des individus  d'une métrique particulière, dite de Mahalanobis. Pour minimiser l'effet de la variance intraclasse considérée comme parasite, du bruit, la métrique de Mahalanobis est définie par la matrice **inverse** de la variance intraclasse, matrice carrée symétrique définie positive.

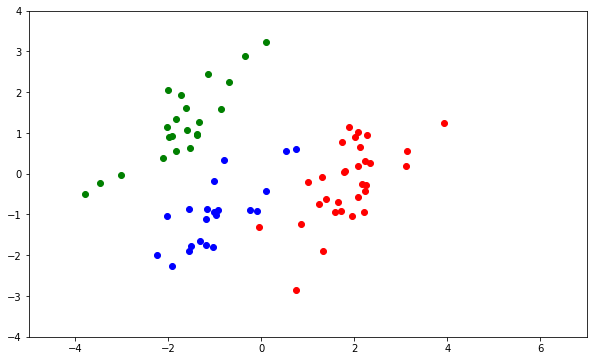
Deux techniques cohabitent sous la même appellation d'**analyse discriminante** ; celle :

* [descriptive](http://wikistat.fr/pdf/st-m-explo-afd.pdf) ou exploratoire qui vise la meilleure représentation graphique des classes ;
* [décisionnelle](http://wikistat.fr/pdf/st-m-app-add.pdf) qui, connaissant pour un individu donné les valeurs des , cherche à prédire la classe inconnue de la variable qualitative.

Il faut lire l'AFD comme un préalable à la construction d'une analyse discriminante décisionnelle ou à tout autre méthode ou algorithme de classification supervisée. L'objectif principal est de représenter les **qualités discriminatoires** des variables quantitatives pour séparer correctement les classes de la variable qualitative. Le graphique des individus permet d'apprécier la qualité de discrimination tandis que, si le nombre de variables n'est pas trop grand, une représentation des variables permet, comme en ACP, d'expliquer la discrimination des classes par les variables quantitatives.

#### Exemple jouet : les insectes de Lubitsch

Cette méthode est illustrée par une comparaison des sorties graphiques issues d'une ACP et d'une AFD. Les données décrivent trois classes d'insectes sur lesquels ont été réalisées **6 mesures anatomiques** sur les ailes, élytres, antennes, pattes des insectes. On cherche à savoir si ces mesures permettent de retrouver la typologie de ces insectes. Ce jeu de données académique conduit à une discrimination assez évidente. La comparaison entre l'ACP et l'AFD met clairement en évidence le rôle de la distance de Mahalanobis sur la forme des nuages de chaque classe en analyse discriminante.

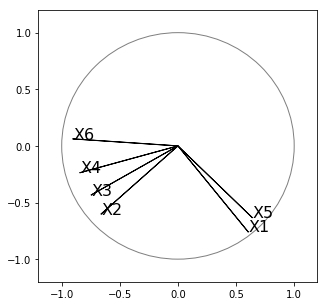


Représentation des trois classes d'insectes dans le premier plan d'une ACP

Les données académiques sont faciles à étudier, l'ACP ci-dessus montre déjà que les trois nuages d'insectes se distinguent assez bien dans le premier plan principal. Observons la forme identique des trois nuages ; les trois classes partagent approximativement la même matrice de variance qui est aussi la variance intraclasse.

Attention, bien évidemment, si les dispersions à l'intérieur de chaque classe ne sont pas homogènes (hétéroscédasticité), l'effet de la métrique de Mahalanobis ne sera pas aussi flagrant.

Le graphique ci-dessous représente les variables et plus précisément leur **structure de corrélation**.

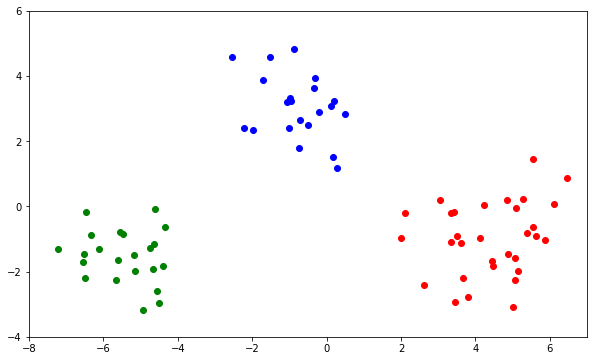


Représentation des variables avec cercle des corrélations dans le premier plan de l'ACP

Le cercle présent sur le graphique est appelé **cercle des corrélations**. Il est défini par l'intersection de la boule unité avec le plan de projection des vecteurs variables. Les variables sont centrées et surtout réduites, divisées par leur écart type. Dans ce cas, ce sont des vecteurs de longueur 1 donc placés sur la boule unité. Ils se projettent nécessairement tous à l'intérieur du cercle. Le cercle des corrélations est une aide graphique pour apprécier la qualité de représentation de chaque variable. Plus une variable est proche du cercle, meilleure est sa représentation.

Dans le cas contraire, le vecteur variable présente un angle grand, proche de 90°, avec le plan de projection, et sa représentation est mauvaise ; ne pas en tenir compte dans l'interprétation.

L'AFD produit le graphique ci-dessous. Cette ACP des barycentres des classes sépare encore mieux les classes d'insectes et surtout, le changement de métrique rend sphérique la forme des classes qui sont donc mieux regroupées autour de leur barycentre.



Représentation des trois classes d’insectes dans le premier plan de l’analyse factorielle discriminante (AFD)

## Explorez des données complexes

### 

### ACP de données de pollution

Besse et al. (2007) considèrent un jeu de données observées dans le contexte de la prévision, pour le lendemain, de la concentration en ozone dans différentes agglomérations, afin d'évaluer les risques de dépassement du seuil légal.

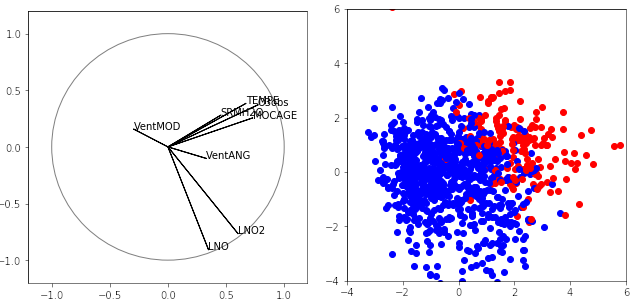
Le problème peut être considéré :

* comme un cas de régression : la variable à prévoir est une concentration en ozone ;
* mais  également comme une discrimination binaire : dépassement ou non du seuil légal.

Il n'y a que 8 variables explicatives dont une ( MOCAGEMOCAGE ) est déjà une prévision de concentration d'ozone mais obtenue par **un modèle déterministe** de mécanique des fluides (équations de Navier et Stockes). Il s'agit d'un exemple d'**adaptation statistique**. La prévision déterministe sur la base d'un maillage global (30 km) est améliorée **localement**, à l'échelle d'une ville, par un modèle statistique incluant cette prévision ainsi que des informations connues sur la base d'une grille locale, spatiale et temporelle plus fine :

* JOURJOUR : le type de jour, férié ou non ;
* O3obsO3obs : la concentration d'ozone effectivement observée le lendemain à 17 h locales, correspondant souvent au maximum de pollution observée ;
* MOCAGEMOCAGE : prévision de cette pollution obtenue par un modèle déterministe de mécanique des fluides (équation de Navier et Stockes) ;
* TEMPETEMPE : température prévue par Météo France pour le lendemain 17 h ;
* RMH2ORMH2O : rapport d'humidité ;
* NO2NO2 : concentration en dioxyde d'azote ;
* NONO : concentration en monoxyde d'azote ;
* STATIONSTATION : lieu de l'observation : Aix-en-Provence, Rambouillet, Munchhausen, Cadarache et Plan de Cuques ;
* VentMODVentMOD : force du vent ;
* VentANGVentANG : orientation du vent.

L'étude préliminaire rudimentaire a conduit à la transformation (log) de certaines variables de concentration (cf. ML4IoT−Tutorial−OzoneML4IoT−Tutorial−Ozone). Les données sont résumées par leur représentation dans le premier plan de l'analyse en composantes principales réduites.



ACP des données de concentration d'ozone : premiers plans factoriels de représentation des variables et des individus

Ce graphique résume la structure de corrélation assez intuitive des variables. Un premier groupe (température, mocage, rapport d'humidité) est corrélé et concourt à la concentration en ozone.  Deux autres variables liées à la concentration en oxyde d'azote sont également corrélées entre elles et avec le premier axe, mais décorrélées du premier groupe.

Enfin, les variables décrivant le vent sont mal représentées dans le premier plan et liées avec le 3ee axe. Le graphique des individus à droite met en évidence les **difficultés à venir** pour discriminer les deux classes respectivement colorées en rouge et bleu : présence ou non d'un pic de concentration avec dépassement du seuil légal.

En présence d'une variable qualitative à deux classes, l'AFD qui se réduit à un graphe de dimension 1, défini par l'axe reliant les deux barycentres, n'a que peu d'intérêt.

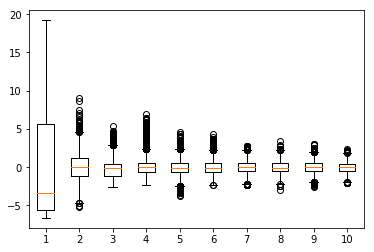
### ACP des transformations des signaux

Appliquons les deux méthodes d'exploration multidimensionnelle aux données obtenues après transformation des signaux temporels issus d'un smartphone, d'abord avec l'ACP en considérant seulement les données issues des signaux, puis avec l'AFD en ajoutant la variable qualitative correspondant au type d'activité (debout, assis, couché...).

L'ACP prend tout son sens lorsqu'elle est appliquée à des données de très grande dimension ; **p=561** variables ou caractéristiques sont observées ou plutôt calculées à partir des signaux bruts sur **n=10299** individus, ou plutôt expérimentations d'une activité. Le tutoriel **ML4IoT−UseCase−Har** réalise tous les calculs en Python pour représenter l'analyse en composantes principales réduites, puis l'AFD de ces données.

#### Décroissance des valeurs propres

Une première information est fournie par la décroissance des valeurs propres, c'est-à-dire par la décroissance des variances de chaque variable principale. Un graphique spécifique précise les choses en opérant des diagrammes boîtes parallèles des composantes principales.

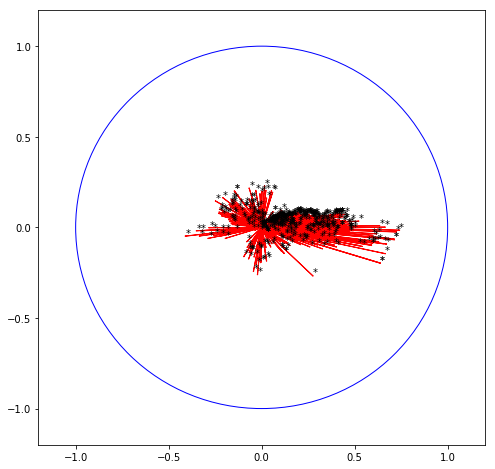


ACP des données transformées des signaux : diagrammes boîtes parallèles des composantes principales

De gauche à droite, chaque diagramme boîte représente la distribution de chaque composante principale, ici, les 10 premières des 561 composantes. La première composante de plus grande variance est associée à une grande boîte accompagnée d'une grande moustache puis, la variance décroissante, les boîtes deviennent de plus en plus petites.

Noter la présence de nombreuses valeurs **atypiques.** Nous nous limiterons aux trois premières composantes, considérant ensuite que les composantes de boîte trop petite ne sont associées qu'à du bruit sans information pour distinguer les activités.

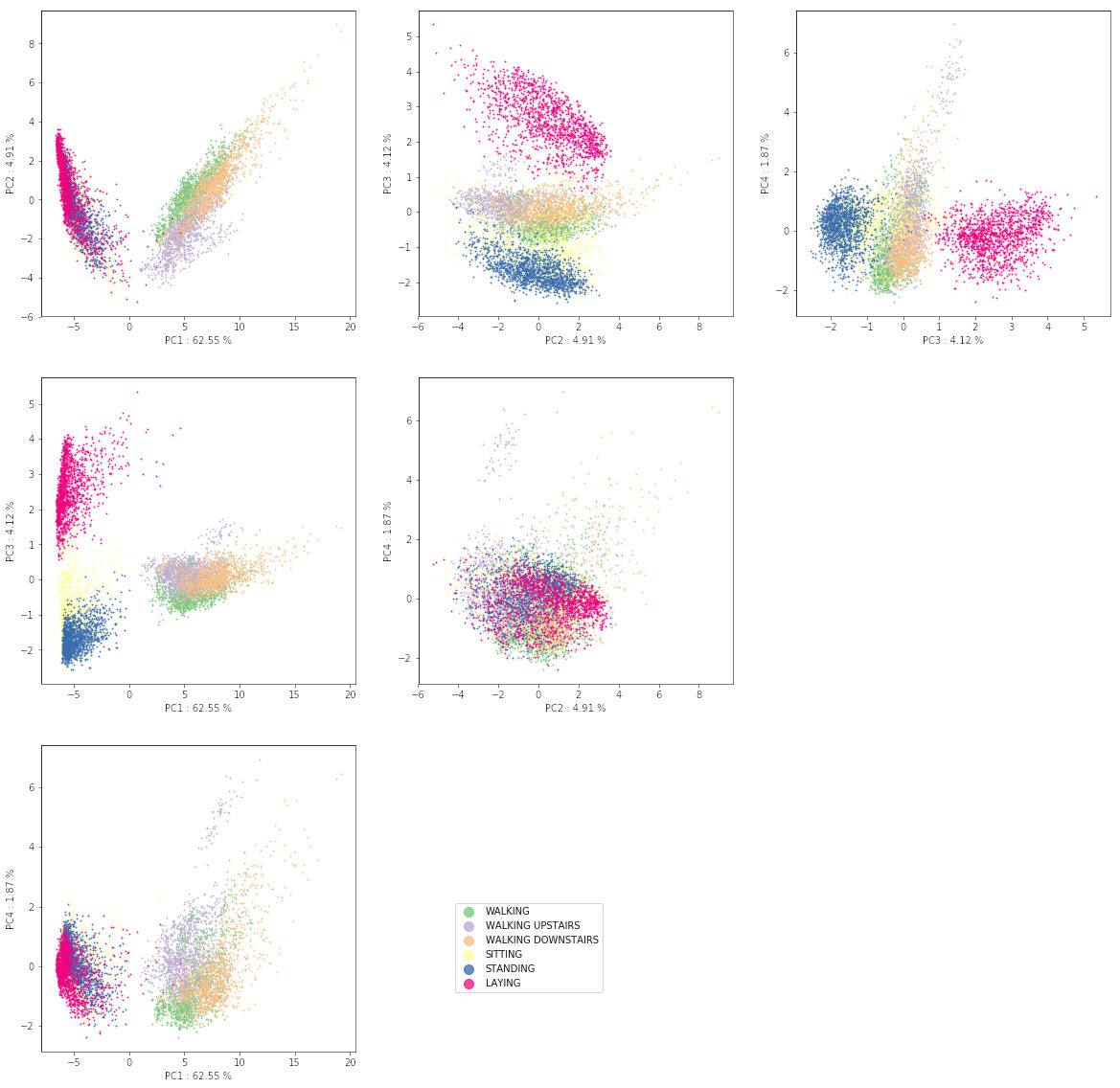
#### Représentation des variables



ACP des données transformées des signaux : représentation des variables dans le premier plan factoriel

Lorsque le nombre **p** de variables est raisonnable, la représentation de celles-ci dans le premier plan factoriel aide à comprendre la structure de corrélation. Avec **p** trop grand, cette représentation est inexploitable.

#### Représentation des individus



ACP des données transformées des signaux : représentation des individus (légende, de haut en bas : marcher, monter ou descendre un escalier, être couché, assis ou debout)

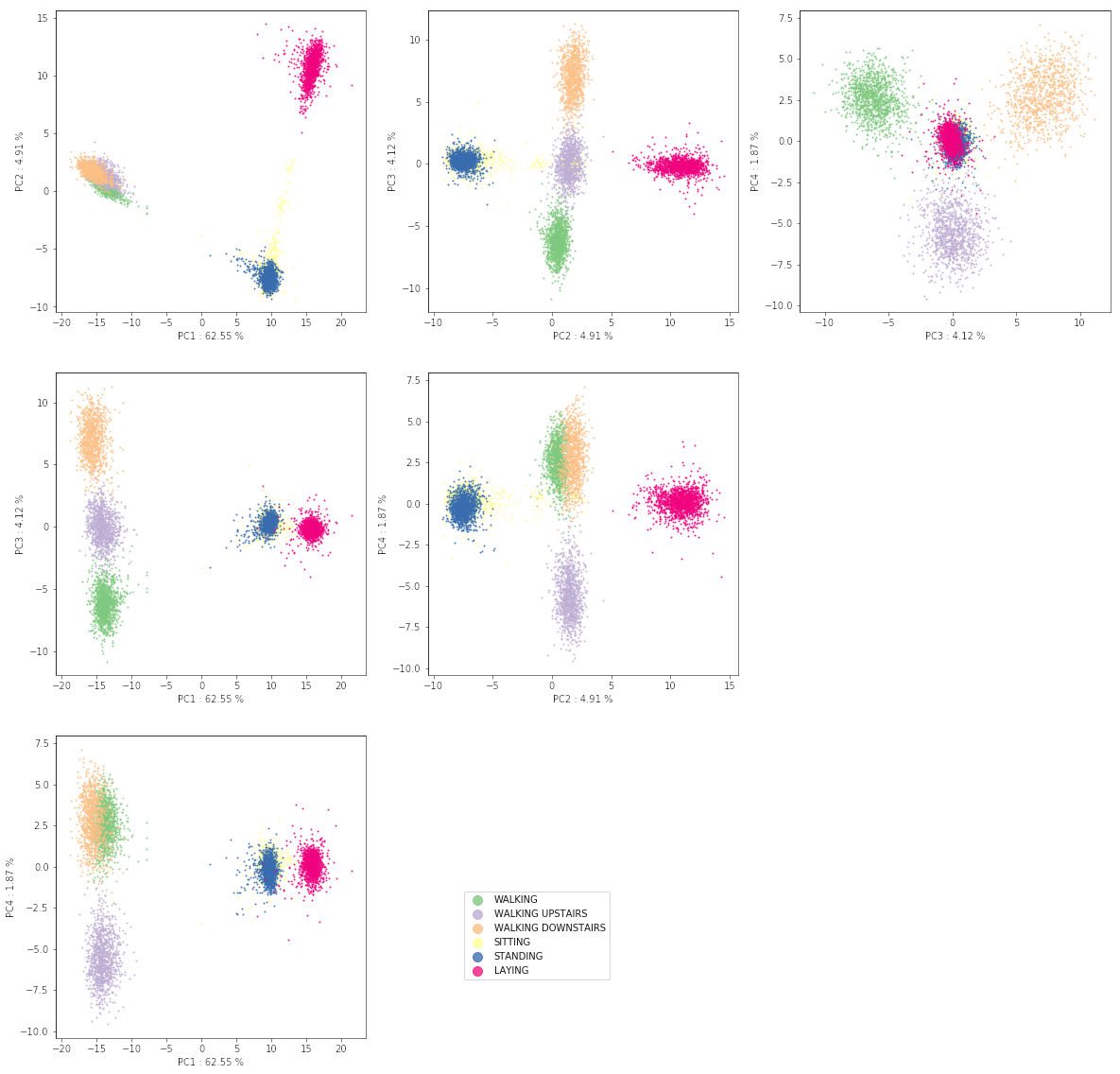
En revanche, il est très informatif de représenter chacune des 6 activités dans les trois premiers axes factoriels, en associant une couleur à chaque activité. Il apparaît que, même sans savoir qu'il y a 6 types de données, les activités se distinguent assez naturellement. Plus précisément, deux groupes d'activités : **dynamique** ou **statique,** sont clairement distingués, expliquant ainsi la grande variance du premier axe.

Néanmoins, il apparaît comme difficile de séparer, discriminer nettement certains couples d'activités.

Attention, l'ACP se comporte comme si les activités n'étaient pas connues, c'est une approche non supervisée. Comme les activités sont a priori connues, il est possible de prendre en compte cette connaissance dans une approche plus adaptée que l'ACP. Il s'agit de l'analyse factorielle discriminante (AFD), sommairement introduite dans la section précédente.

### 

### AFD des transformations des signaux



AFD des données transformées des signaux : représentation de chacun des 6 types d'activités dans les premiers plans factoriels (légende, de haut en bas : marcher, monter ou descendre un escalier, être couché, assis ou debout).

La couleur d'un point est fonction de l'activité connue. L'ACP des 6 barycentres des classes en utilisant la **métrique de Mahalanobis** (matrice inverse de la variance intra-classe) a pour effet d'encore mieux séparer les 6 classes. En considérant les trois premiers axes ou plutôt les projections sur les plans définis par ces trois premiers axes, il apparaît que les classes d'activités sont bien séparées dans cet espace à 3 dimensions. À l'exception des deux classes : assis, couché, plus difficiles à séparer, il existe un plan de projection dans lequel une classe se distingue des autres. Autrement dit, il existe des hyperplans ou séparations linéaires des classes.

La construction de ces hyperplans, frontières entre les classes, est l'objectif implicite de la prochaine partie. Leur connaissance conduit à des règles de décision pour la prévision d'une activité dont les signaux, ou plutôt les caractéristiques, sont connues mais pas la classe.

# Mettez en pratique les concepts de la partie 2

### 

### Que faut-il retenir ?

Les **signaux temporels bruts** sont très confus, notamment à cause de leurs décalages temporels ou déphasages. Ils sont difficiles à caractériser en l'état et sont pour le moment laissés de côté. En revanche, les transformations de ces données (features) par des techniques de traitement du signal, dont leur décomposition de Fourier, présentent des caractéristiques plus encourageantes.

Ce sont 561 variables qui sont calculées sur les 9 courbes enregistrées par chaque activité des porteurs d'un smartphone.  Nous sommes donc toujours dans une situation de grande dimension dont l'exploration nécessite des méthodes appropriées.

L'**analyse en composantes principales** permet de représenter le nuage des activités en seulement 3 dimensions avec des séparations encourageantes des classes d'activités, sans les connaître a priori ; 561 variables ou caractéristiques sont donc résumées de façon satisfaisante par l'ACP, avec seulement  trois nouvelles variables ou variables principales, combinaisons linéaires de celles initiales.

Néanmoins, le nombre trop important de ces variables ne permet pas d'obtenir simplement et directement une interprétation utile des variables principales.

Comme nous sommes dans un contexte supervisé car les classes d'activités sont connues, c'est l'**analyse factorielle discriminante** qui est appropriée. Cette ACP particulière, calculée sur les 6 barycentres des classes en utilisant comme métrique la matrice inverse de la variance intraclasse conduit, toujours en dimension 3, à de très bonne séparations des classes. À l'exception des deux activités : assis et couché, il semble bien que toutes les autres peuvent être séparées par des hyperplans. En d'autres termes, une **méthode** ou un **algorithme** linéaire de classification supervisée devrait permettre d'atteindre l'objectif de reconnaissance de l'activité.

Attention. Le problème initial, fil rouge de ce cours, est un problème de classification supervisée des activités. Par manque de temps ou de place, nous faisons le choix de laisser de côté les méthodes et algorithmes de [**classification non supervisée**](http://wikistat.fr/pdf/st-m-explo-classif.pdf) ou clustering. Leur objectif est la recherche de groupes ou classes homogènes des nn individus à partir de l'observation d'un ensemble de pp variables XjXj. Ces méthodes sont développées dans d'autres tutoriels d'un dépôt github.com/wikistatgithub.com/wikistat.

### Tutoriels

Trois tutoriels sont proposés pour illustrer et compléter le cours :

* [ML4IoT-Tutorial-ACPAFD](https://s3-eu-west-1.amazonaws.com/course.oc-static.com/courses/5919236/ML4IoT-Intro-ACP-AFD.ipynb)propose de s'initier à l'ACP puis à l'AFD sur des données élémentaires ;
* [ML4IoT-Tutorial-Ozone](https://s3-eu-west-1.amazonaws.com/course.oc-static.com/courses/5919236/ML4IoT-Tutorial-Ozone.ipynb), dont une section illustre l'utilisation de l'ACP ;
* [ML4IoT-Tutorial-HAR](https://s3-eu-west-1.amazonaws.com/course.oc-static.com/courses/5919236/ML4IoT-UseCase-Har.ipynb), dont la première partie explore les données issues des transformations des signaux bruts.

### 

### Références

**Wikistat** (2018). [Analyse en composantes principales](http://wikistat.fr/pdf/st-m-explo-acp.pdf), en ligne, page disponible le 17/05/2018.  
**Wikistat** (2018). [Analyse factorielle discriminante](http://wikistat.fr/pdf/st-m-explo-afd.pdf), en ligne, page disponible le 17/05/2018.

# Explorez des données multidimensionnelles

Bravo ! Vous avez réussi cet exercice !

### Compétences évaluées

* Exécuter une analyse exploratoire multidimensionnelle

### Question 1

**Qu'est-ce que la transformée de Fourier de signaux ?**

*Attention, plusieurs réponses sont possibles.*

* + 

Une représentation dans le domaine temporel

* + 

Une base de représentation adaptée aux signaux stationnaires

* + 

Une représentation invariante par déphasage

*Reportez-vous au chapitre "Prenez en charge les données".*

### Question 2

**Qu'est-ce que l'analyse en composantes principales (ACP) ?**

*Attention, plusieurs réponses sont possibles.*

* + 

Une méthode exploratoire de réduction de dimension

* + 

Une interprétation statistique d'une représentation géométrique

* + 

Un changement de base dans un espace vectoriel

*Tout à la fois ! Voir chapitre "Comprenez l'analyse en composantes principales".*

### Question 3

**Qu'est-ce qu'une ACP réduite ?**

* + 

Une ACP de la plus petite dimension optimale

* + 

Une ACP calculée sur les variables centrées et divisées par leur écart-type

* + 

Une ACP calculée sur un sous-ensemble des données

*Retrouvez cette définition dans le chapitre "Comprenez l'analyse en composantes principales".*

### Question 4

**Qu'est-ce qu'une composante principale ?**

* + 

Une combinaison linéaire des variables observées centrées

* + 

La variable observée de plus grande variance

*Reportez-vous au chapitre "Comprenez l'analyse en composantes principales".*

### Question 5

**Que représente le cosinus de l'angle entre deux vecteurs variables d'une représentation de l'ACP ?**

* + 

La covariance des deux variables de l'ACP non réduite

* + 

La corrélation des deux variables de l'ACP réduite

* + 

Une approximation de la corrélation entre les deux variables

*L'explication se trouve dans le chapitre "Comprenez l'analyse en composantes principales".*

### Question 6

**Que définit un vecteur propre associé à une valeur propre de l'ACP ?**

*Attention, plusieurs réponses sont possibles.*

* + 

Un axe d'inertie du nuage des individus

* + 

La composante principale de variance de la valeur propre

*Retrouvez cette définition dans le chapitre "Comprenez l'analyse en composantes principales".*

### Question 7

**Qu'est-ce que le cercle des corrélations ?**

*Attention, plusieurs réponses sont possibles.*

* + 

L'intersection de la sphère unité avec le plan de projection

* + 

Une aide à l'interprétation des corrélations des variables

* + 

Une aide à l'évaluation de la qualité de représentation des variables

*Cette notion est abordée dans le chapitre "Comprenez l'analyse en composantes principales".*

### Question 8

**Que représente le biplot en ACP ?**

*Attention, plusieurs réponses sont possibles.*

* + 

Une représentation graphique de la décomposition en valeurs singulières de la matrice des données centrées

* + 

La représentation simultanée des individus et des variables

* + 

Le cercle des corrélations superposé avec la représentation des variables

*Reportez-vous au chapitre "Comprenez l'analyse en composantes principales".*

### Question 9

**Qu'est-ce que l'analyse factorielle discriminante (AFD) ?**

* + 

Une ACP de barycentres avec métrique de Manhattan

* + 

Un modèle de classification supervisée

* + 

Une ACP avec métrique de Mahalanobis

*Cette définition se trouve dans le chapitre "Explorez des données complexes".*

### Question 10

**D'après le tutoriel d'introduction à l'ACP, données OCDE (**[**ML4IoT-Tutorial-ACPAFD**](https://s3-eu-west-1.amazonaws.com/course.oc-static.com/courses/5919236/ML4IoT-Intro-ACP-AFD.ipynb)**), quelle est l'interprétation en deux mots du premier axe de l'ACP (section 5.4) ?**

* + 

Axe de la taille (population) des pays

* + 

Axe de montée du chômage

* + 

Axe de développement économique

*La variable principale (premier axe) est corrélée négativement aux variables synonymes de développement et de richesse  et négativement à celles liées à la "ruralité". Le deuxième axe met en évidence la montée du chômage correspondant à la décroissance de la part d'actifs dans le secondaire.*

### Question 11

**D'après le tutoriel ACP des données concentration d'ozone (**[**ML4IoT-Tutorial-Ozone**](https://s3-eu-west-1.amazonaws.com/course.oc-static.com/courses/5919236/ML4IoT-Tutorial-Ozone.ipynb)**), quelle dimension de représentation choisir pour résumer au mieux les données ?**

* + 

Une

* + 

Deux

* + 

Trois

*Regarder la décroissance de la taille des boîtes des diagrammes boîtes parallèles des composantes principales. Les boîtes 2 et 3 sont de tailles similaires (valeurs propres proches), la 4e est plus petite. Nous pouvons considérer qu'à partir de la 4e composante, il s'agit de bruit négligeable.*

### Question 12

**D'après le tutoriel AFD des données "métier" d'un smartphone (**[**ML4IoT-Tutorial-ACPAFD**](https://s3-eu-west-1.amazonaws.com/course.oc-static.com/courses/5919236/ML4IoT-Intro-ACP-AFD.ipynb)**), quelles sont les classes d'activité qui ne se séparent pas naturellement ?**

* + 

Couché vs. assis

* + 

Assis vs. debout

* + 

Couché vs. debout

*Il est possible de trouver un plan de projection séparant tous les couples de classes, sauf celles correspondant aux activités "assis" et "debout". Néanmoins, ces graphiques montrent que des modèles linéaires atteindront correctement l'objectif de discrimination.*

# Abordez les principes de l'apprentissage statistique

### 

### Modélisation et prévision

Dans les années 30 et notamment à la suite des travaux de **Ronald Fisher**, la statistique a été développée avec une finalité principalement explicative, pour un objectif d'aide à la décision. Par exemple : **tester** l'efficacité d'une molécule et donc d'un médicament, **comparer** le rendement de semences ou **optimiser** le choix d'un engrais, **montrer** l'influence d'un facteur (consommation de tabac, de sucre) sur des objectifs de santé publique.

La prise de décision est alors soumise à un [test statistique](http://wikistat.fr/pdf/st-m-inf-test.pdf) permettant de contrôler le risque d'erreur encouru. Mais il se trouve que les mêmes modèles statistiques peuvent aussi être utilisés avec une finalité seulement ***prédictive***: prévoir la concentration en ozone du lendemain, le risque de défaut de paiement d'une entreprise...

Ils sont les premiers algorithmes d'apprentissage statistique, encore très utilisés et les rares transparents, c'est-à-dire suffisamment simples et explicites pour conduire à une interprétation sur la façon dont la prévision fonctionne.

Deux modèles statistiques sont considérés :

1. **La régression linéaire multiple** qui vise à modéliser, prévoir une variable quantitative **Y** (cible ou dépendante) à l'aide d'un ensemble de variables (features ou caractéristiques) quantitatives et/ou qualitatives .
2. **La régression logistique** qui est une adaptation de la précédente afin de prévoir l'occurrence (défaut, succès, maladie...) d'une variable qualitative binaire **Y** ou plutôt, en premier lieu, la probabilité de cette occurrence.

Commençons par introduire le modèle de régression linéaire. Ayant observé les valeurs  des variables sur un ensemble d'apprentissage **i=1…,n** d'objets individus ou instances, la modélisation par régression linéaire consiste à estimer au mieux les paramètres d'un modèle : ou matriciellement ,  
où les **εi** sont des termes d'erreur.

L'estimation des paramètres par minimisation des moindres revient à minimiser la moyenne des carrés (MSE ou mean square error). La prévision de **Y** pour un nouvel individu **x0** est alors obtenue en appliquant le modèle :

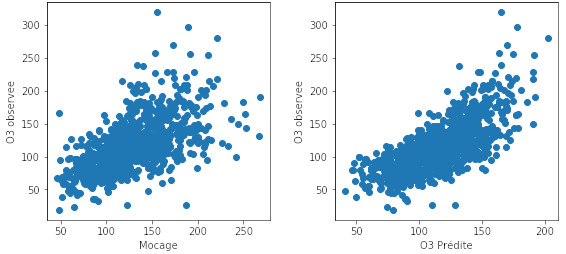


Moyennant un ensemble d'hypothèses sur la loi des résidus : normalité, homoscédasticité, l'estimation du modèle peut être définie par un principe de maximisation de la log-vraisemblance. C'est en fait la même solution que celle issue des moindres carrés, mais cette approche confère de meilleures propriétés au modèle.

Ces hypothèses rendent possibles des tests et prévisions par **intervalle de confiance** afin de contrôler le risque d'erreur. Ce point n'est pas développé : se [**reporter**](http://wikistat.fr/pdf/st-m-modlin-regmult.pdf) à l'abondante littérature sur ce sujet.

Avant d'aborder le problème plus complexe de reconnaissance d'une activité humaine à partir des signaux enregistrés par un smartphone, nous nous intéressons aux données de prévision de la concentration en ozone vues dans la partie 2 précédente, et détaillées dans le tutoriel **ML4IoT−Tutorial−Ozone**. L'objectif est d'illustrer les modèles statistiques de régression. Cet exemple, à la fois de régression et de discrimination binaire, présente des vertus pédagogiques certaines qui permettent de l'utiliser comme fil rouge de comparaison entre toutes méthodes.

Le graphique ci-dessous utilise ces données pour comparer deux ajustements de modèle :



Nuages de points du modèle déterministe Mocage (à gauche) et du modèle obtenu par adaptation statistique (à droite)

celui obtenu par le modèle déterministe MOCAGE à gauche et celui à droite obtenu par adaptation statistique en introduisant les autres variables explicatives. Le modèle (choix de variables) a été optimisé comme cela est expliqué par la suite. Les graphes représentent les valeurs observées  en fonction des valeurs prédites .

Plus celles-ci sont proches de la diagonale et meilleur est l'ajustement du modèle (plus petits résidus). Le carré  du coefficient de corrélation entre  **y** et  est appelé coefficient de détermination. Il vaut **0,10** pour MOCAGE et **0,52** pour le modèle statistique intégrant un choix optimal de variables.

### Erreur de prévision

Il est fondamental de ne pas confondre l'**erreur d'estimation** (MSE) minimisée lors de l'ajustement du modèle et l'**erreur de prévision**du modèle ou risque, erreur de généralisation qui doit nécessairement être estimée sur un échantillon indépendant.

En effet, l'erreur d'ajustement, encore appelée erreur apparente, est estimée sur des observations qui participent à l'ajustement du modèle. Par principe, elle est une version **optimiste** de l'erreur de prévision, qui concerne des observations nouvelles qui n'ont pas participé à l'ajustement du modèle. Les écarts ou résidus de la prévision sont en effet, sauf cas particuliers, de nature à être plus grands que les résidus minimisés lors de l'estimation.

Une façon élémentaire d'estimer une erreur de prévision consiste à opérer en deux phases après avoir séparé aléatoirement l'échantillon  en deux parties :   
- celle d'apprentissage pour estimer un modèle ou entraîner un algorithme ;  
- et celle de test avec **n=na+nt**.

Une estimation de l'erreur de prévision ou risque : RMSE ou root mean square error, est alors fournie par :

RMSE

L'objectif majeur d'une étape d'apprentissage statistique est de **déterminer le modèle ou l'algorithme**, parmi tous ceux disponibles ou parmi une classe réduite de ceux interprétables, qui conduit à la plus petite erreur de prévision ou plus petit risque.

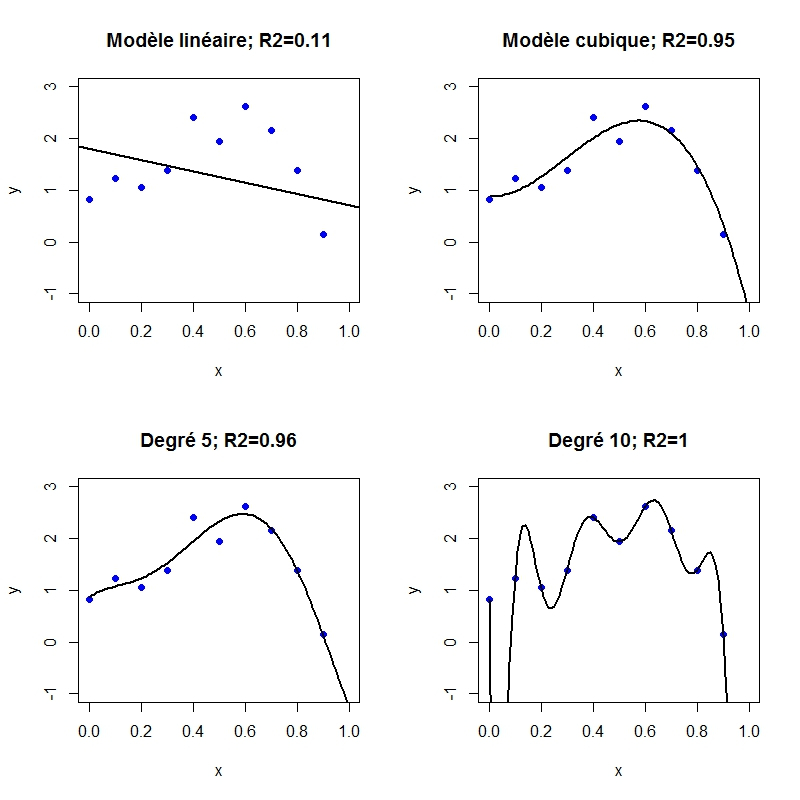
### Sélection de modèle

La recherche du modèle statistique ou de l'algorithme d'apprentissage optimal nécessite de prendre conscience d'une réalité importante :

le modèle ou l'algorithme qui ajuste le mieux les données d'apprentissage n'est pas nécessairement celui qui conduit à la meilleure prévision.

Rechercher le meilleur modèle parmi une famille donnée consiste, d'un point de vue statistique, à réaliser un **meilleur compromis entre biais et variance**. Ceci peut être illustré simplement dans le cas de la régression polynomiale.

Le graphique ci-dessous compare les ajustements d'un nuage de points par régression polynomiale. Les points sont successivement ajustés par des polynômes de degré 1, 3, 5 et 10.



Comparaison des ajustements des nuages de points par des polynômes de degrés 1, 2, 5 et 10

Les coefficients de détermination qui mesurent la qualité de l'ajustement sont successivement :  (degré 1) ,   (degré 3),  (degré 5),  (degré 10 ou interpolation exacte des points).

Clairement, la qualité de l'ajustement du modèle croît avec le  et donc avec la complexité, le nombre de paramètres ou le degré du polynôme. Intuitivement, le nuage de points a la forme d'un modèle relativement simple mais dont les observations sont entachées d'erreurs. Vouloir ajuster au mieux ces observations conduit à interpoler la composante d'erreur ou de bruit au détriment de la régularité du modèle.

La conséquence directe en est la construction de prévisions qui prendront des valeurs absolues beaucoup trop importantes par rapport à celles généralement observées. La **variance des erreurs** de prévision prend alors une très grande valeur qui conduit à l'explosion du RMSE. En revanche, accepter un modèle, éventuellement plus simple ou **biaisé** que le supposé vrai modèle, évite d'ajuster le modèle aux erreurs de mesure et réduit la **variance**. C'est le principe de la recherche d'un meilleur compromis biais / variance ; compromis qui dépend de la variance du bruit par rapport à celle des observations (rapport signal / bruit).

La stratégie de choix de modèle consiste donc à la recherche d'un modèle **parcimonieux** (sparse) au sens où sa complexité optimise le compromis entre biais et variance. C'est facilement illustré dans le cas de la régression mais ce principe se retrouve dans presque tous les algorithmes d'apprentissage dont il faut contrôler la complexité et donc la flexibilité de l'ajustement aux données.  En fonction du type d'algorithme, ce peut être le nombre de variables, de feuilles, de voisins, de neurones ou un coefficient de pénalisation : **Ridge**, Lasso (least absolute shrinkage and selection operator)... qui règle la flexibilité ou complexité du modèle.

### Sélection par pénalisation

Il existe de très [nombreuses stratégies](http://wikistat.fr/pdf/st-m-app-linSelect.pdf) (descendante, ascendante, pas-à-pas, par pénalisation Ridge ou Lasso...) de sélection de modèle en régression, stratégies basées sur une grande variété de critères : test de Fisher, critère d'Akaïke, critère bayésien (BIC),  de Mallows. La stratégie actuellement la plus utilisée est celle proposée par Tibshirani (1996) basée sur une pénalisation Lasso dite encore en norme  de la somme des valeurs absolues des paramètres. Ce qui s'écrit encore :



qui est équivalent à minimiser les moindres carrés (MSE) sous une contrainte :

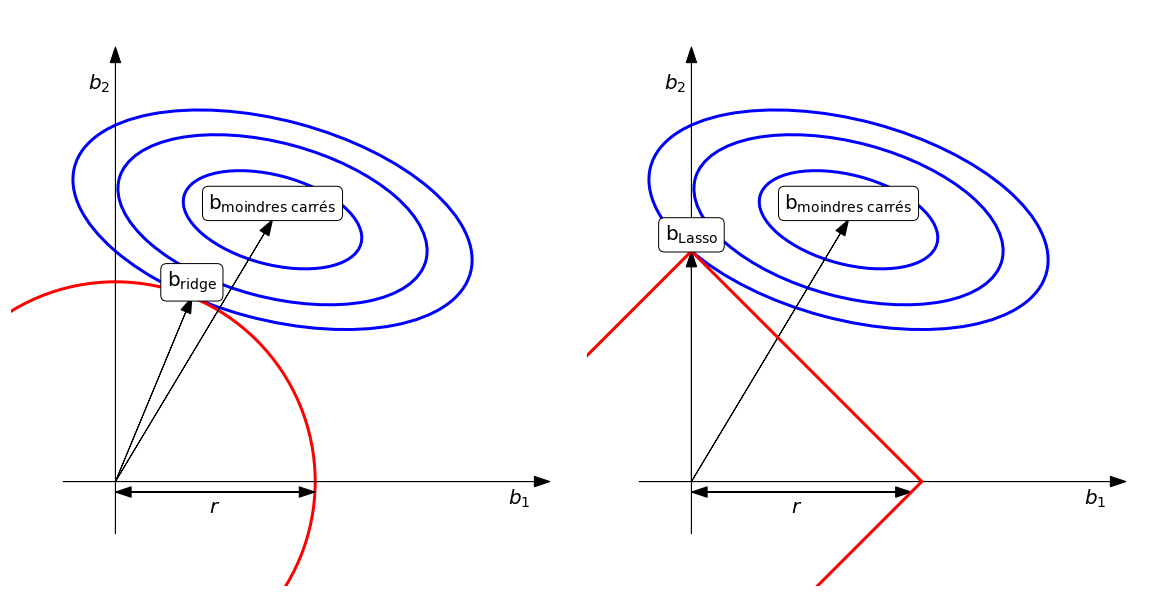


La régression Ridge satisfait à la même formule :



en remplaçant la norme  par la norme **L2**, c'est-à-dire la somme des valeurs absolues par la somme des carrés des coefficients. La régression **Ridge** suit le même principe que la régularisation de Tikhonov pour résoudre un problème inverse indéterminé.

Le graphique ci-dessous propose une interprétation géométrique des pénalisations **Ridge** et Lasso, afin d'expliquer en quoi cette dernière opère automatiquement une sélection de variables par annulation des paramètres **bj** les moins significatifs.



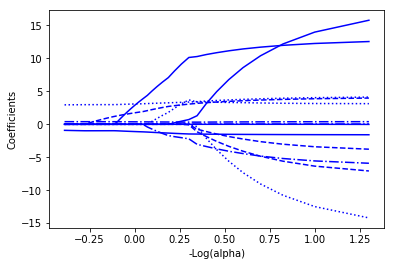
Interprétation géométrique des pénalisations Ridge et Lasso

Dans les deux graphiques, les ellipses représentent les valeurs du critère des moindres carrés. Lorsque la pénalisation est nulle (**λ=0**), les paramètres  prennent les valeurs des moindres carrés minimaux. En norme , la contrainte est visualisée par une hypersphère, un cercle en dimension 2, de rayon **r**. La régression Ridge conduit alors à un **rétrécissement** (shrinkage) à la rencontre de l'ellipse des moindres carrés et de la sphère de la contrainte.

La pénalisation Lasso remplace l'hypersphère par un hypercube, un carré en dimension 2. Dans ce cas, l'ellipse à toutes les chances de rencontrer un coin en dimension 2 ou un hyperplan de dimension **n−1** correspondant à l'annulation de certains paramètres, ici **b1=0**.

Cette stratégie de sélection de modèles peut paraître complexe mais elle est **largement utilisée**, notamment dans la librairie **Scikit-learn** de Python. Seule insuffisance de cette librairie, la difficulté, voire l'impossibilité, de pouvoir prendre en compte simplement des interactions entre les variables, alors que la syntaxe de R le permet aisément.

Le graphe ci-dessous représente le **chemin de régularisation** en pénalisation Lasso lors de la recherche d'un modèle optimal.



Modélisation de la concentration de l'ozone : chemin de régularisation en pénalisation Lasso – de droite à gauche évolution des paramètres du modèle en fonction de la pénalisation.

Il montre comment les paramètres décroissent et certains s'annulent lorsque la pénalisation croît de droite à gauche des abscisses associées aux valeurs d'un paramètre **α** qui joue un rôle similaire à **λ**.

Un troisième type de sélection de modèle dit elastic net consiste à associer pénalisations Ridge et Lasso mais nécessite le réglage de deux paramètres.

### Optimisation par validation croisée V - couches

La question alors soulevée est de savoir comment **optimiser** la valeur de ce paramètre de pénalisation. La stratégie unanimement employée consiste à minimiser un risque estimé par une procédure de [*validation croisée*](http://wikistat.fr/pdf/st-m-app-risque.pdf). L'idée est d'itérer l'estimation sans biais de l'erreur sur plusieurs échantillons dits de validation n'ayant pas été utilisés pour l'estimation du modèle ou l'entraînement de l'algorithme, puis d'en calculer la moyenne.

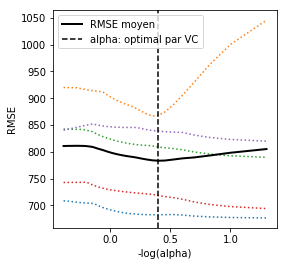
L'objectif est ainsi d'éviter un biais tout en réduisant la variance (par moyennage) de l'estimation pour améliorer la précision lorsque la taille de l'échantillon initial est réduite. Il existe plusieurs versions de cet algorithme, celle décrite succinctement ci-dessous est la validation croisée V-couches (fold), la plus habituelle.

Plus précisément, l'échantillon d'apprentissage est découpé aléatoirement en **V** groupes (par défaut 5 ou 10) de tailles similaires qui jouent à tour de rôle celui de validation. Le i groupe étant mis de côté, les **(V−1)** autres groupes constituent l'échantillon d'apprentissage servant à estimer le modèle. L'erreur de prévision ou risque  est estimée sur le groupe. Les **V** erreurs **RMSEi;i=1,…,V** ainsi obtenues sont moyennées pour calculer , l'estimation par validation croisée **V**-couches du risque ou erreur de prévision.

Minimiser le risque estimé par validation croisée est une approche largement utilisée pour optimiser le choix d'un modèle au sein d'une famille paramétrée par θ où θ est le nombre de variables, de neurones, de feuilles, une pénalisation... Dans ce cas,  est défini par



 Le graphe ci-dessous illustre l'optimisation du paramètre de pénalisation Lasso par validation croisée 5-couches.



Optimisation du paramètre de pénalisation Lasso par validation croisée 5-couches

Chaque ligne colorée représente l'évolution de l'erreur de prévision calculée sur l'un des échantillons de validation en fonction d'une pénalisation **α**, le paramètre de la librairie scikit−learn. La ligne pleine est la moyenne qui admet un minimum auquel est associée la valeur optimale du paramètre de pénalisation.

# Comprenez la classification supervisée

Le chapitre précédent développe la prévision d'une variable cible quantitative **Y** appliquée, par exemple, à la prévision de la concentration en ozone. En revanche, le problème de reconnaissance de l'activité humaine vise à la prévision d'une classe d'activité ou encore de la classe d'une variable Y , cette fois qualitative. Il s'agit d'un problème de classification supervisée ou reconnaissance de forme.

Attention, ne pas confondre avec l'objectif de classification non supervisée ou en anglais clustering, qui vise à trouver des groupes homogènes dans des individus ou instances caractérisées par pp variables. Contrairement à la classification supervisée, il n'y a pas de variable qualitative cible a priori connue et observée sur l'échantillon.

Historiquement, les deux méthodes partageant l'objectif de discrimination, ou de prévision, d'une variable quantitative Y, et apparues en premier sont :

* l'[analyse discriminante décisionnelle](http://wikistat.fr/pdf/st-m-app-add.pdf) ;
* la [régression logistique](http://wikistat.fr/pdf/st-m-app-rlogit.pdf) ou binomiale.

L'analyse discriminante développée par Fisher propose la prévision d'une classe parmi les m modalités de Y, alors qu'en principe la régression logistique est adaptée à une variable Y à deux classes ou binaire : succès ou échec, présence d'une maladie, défaut de paiement, occurrence d'un événement... Néanmoins, cette même méthode, à condition que la taille de l'échantillon le permette, est largement utilisée pour de la discrimination en m>2 classes en construisant m modèles d'une classe contre les autres.

Pour la prévision de la classe d'un nouvel individu ou d'une nouvelle instance, les m modèles fournissent m probabilités d'occurrence de chaque classe, et c'est la classe de probabilité maximale qui l'emporte. Cette stratégie est prise par défaut dans la librairie scikit−learn lors de l'utilisation de la régression logistique avec m>2.

Après ces méthodes historiques, bien d'autres modèles ou algorithmes ont été proposés avec le même objectif de prévision d'une classe ou modalité d'une variable qualitative binaire ou à m classes : [kk plus proches voisins](http://wikistat.fr/pdf/st-m-app-add.pdf), [arbre binaire de décision](http://wikistat.fr/pdf/st-m-app-cart.pdf), [réseau de neurones](http://wikistat.fr/pdf/st-m-app-rn.pdf) (perceptron), [machine à vecteur support](http://wikistat.fr/pdf/st-m-app-svm.pdf) (SVM) ainsi que les algorithmes  d'[agrégation d'arbres](http://wikistat.fr/pdf/st-m-app-agreg.pdf) : boosting, random forest... Se reporter aux tutoriels du dépôt github.com/wikistatgithub.com/wikistat pour les expérimenter.

### Classification binaire par régression logistique

Nous avons vu précédemment comment ajuster une variable Y quantitative, à valeurs dans , par une combinaison linéaire des p variables explicatives . Ce que la régression logistique vise à modéliser est la probabilité d'occurrence (succès, maladie...) d'une classe de Y qui est à valeur dans l'intervalle [0,1].

Plus précisément, le choix d'une fonction lien permet de faire correspondre les deux domaines de variation [0,1] et  afin de relier une probabilité avec un prédicteur linéaire classique Xb . Si πi désigne la probabilité de la classe 1 de Y ou P(yi=1|xi), la fonction lien dite canonique couramment utilisée est la fonction logistique et le modèle s'écrit :



L'estimation des paramètres  de ce modèle est obtenue par l'exécution d'un algorithme de maximisation (e.g. Newton Raphson) de la log-vraisemblance du modèle.

La prévision d'une probabilité  de x0 est fournie par :



La prévision de la classe de x0 est obtenue en comparant cette probabilité avec une valeur seuil ou cut-off, par défaut  sinon.

Moyennant des hypothèses sur la loi de Y (binomiale), la planification de l'expérience et la répartition de l'échantillon, des procédures de test et d'estimation par intervalle de confiance des prévisions sont accessibles. Consulter la [bibliographie](http://wikistat.fr/pdf/st-m-app-rlogit.pdf) à ce sujet. Nous nous limitons ici au seul objectif de prévision.

### Courbe ROC

#### Matrice de confusion

Une erreur quadratique moyenne (RMSE) est généralement utilisée pour évaluer une erreur de prévision ou risque en régression. Ce critère n'est pas adapté au cas de la classification supervisée. Il est souvent remplacé par un simple taux d'erreur calculé à partir de la matrice de confusion. Cette matrice est simplement une table de contingence ou tableau obtenu par le croisement des deux variables : classe observée vs classe prédite.

Dans le cas fréquent de la discrimination de deux classes, la plupart des méthodes (e.g. régression logistique) estiment, pour chaque individu i, un score ou une probabilité  que cet individu prenne la modalité Y=1. Cette probabilité comprise entre 0 et 1 est comparée avec une valeur seuil s fixée a priori, par défaut 0,5 :



Pour un échantillon de taille n dont l'observation de Y est connue ainsi que les scores  fournis par un modèle, la matrice de confusion associée à cette valeur de seuil s est :

|  |  |  |  |
| --- | --- | --- | --- |
| Prévision | Observation | Observation | Total |
|  | Y=1Y=1 | Y=0Y=0 |  |
| yˆi=1y^i=1 | n11(s)n11(s) | n10(s)n10(s) | n1+(s)n1+(s) |
| yˆi=0y^i=0 | n01(s)n01(s) | n00(s)n00(s) | n0+(s)n0+(s) |
| Total | n+1n+1 | n+0n+0 | nn |

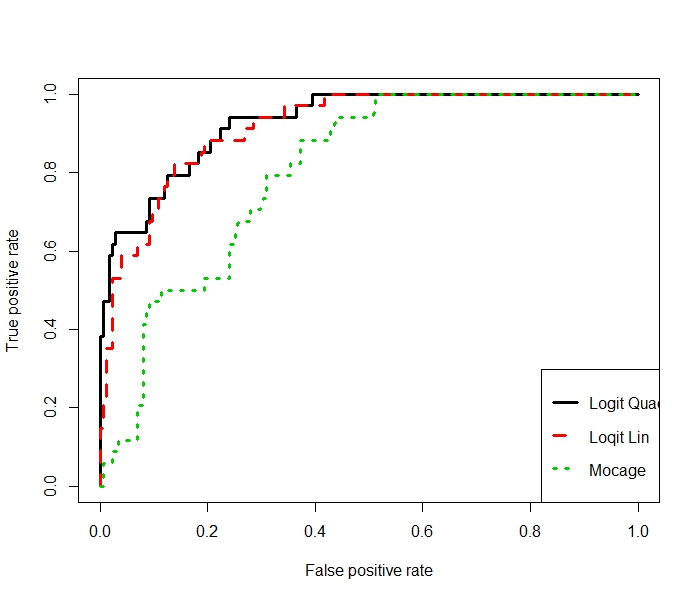
Les quantités suivantes sont considérées :

#### 

#### Courbe ROC et AUC

Les notions de spécificité et de sensibilité proviennent de la théorie du signal ; leurs valeurs dépendent directement de celle du seuil s. En augmentant s, la sensibilité diminue tandis que la spécificité augmente, car la règle de décision devient plus exigeante. Un bon modèle associe grande sensibilité et grande spécificité pour la détection d'un signal. Ce lien est représenté graphiquement par la courbe ROC (Receiver Operating Caracteristic) de la sensibilité (probabilité de détecter un vrai signal) en fonction de moins la spécificité (probabilité de détecter un signal à tort) pour chaque valeur s du seuil.

On montre qu'une courbe ROC est croissante monotone. Plus une courbe de la figure ci-dessous se rapproche du carré, meilleure est la discrimination, correspondant à la fois à une forte sensibilité et une grande spécificité. L'aire sous la courbe : AUC (area under curve) mesure la qualité de discrimination du modèle, tandis qu'une analyse de la courbe aide au choix du seuil.



Modélisation de la concentration de l'ozone : comparaison des courbes ROC de trois modèles

Ce graphique compare trois courbes ROC. Celle issue du modèle MOCAGE en vert avec celles issues de deux modèles de régression logistique, l'un linéaire, l'autre quadratique car faisant intervenir des interactions. Pour comparer des modèles ou méthodes de complexités différentes, ces courbes doivent être estimées sur un échantillon test. Elles sont bien évidemment optimistes sur l'échantillon d'apprentissage. De plus, l'AUC ne définit pas un ordre total entre modèles, car les courbes ROC peuvent se croiser.

Ces résultats montrent encore plus clairement l'intérêt de l'adaptation statistique de la prévision MOCAGE, mais aussi la difficulté de la décision qui découle de la courbe ROC. Le choix du seuil, et donc de la méthode à utiliser si les courbes se croisent, dépend d'un choix dans ce cas politique : quel est le taux de faux positifs acceptable d'un point de vue économique, ou le taux de vrais positifs à atteindre pour des raisons de santé publique ? Le problème majeur est de pouvoir quantifier les coûts afférents, par la définition d'une matrice dissymétrique de ces coûts de mauvais classement en vue d'optimiser le choix de s.

#### Autre critère pour la discrimination à deux classes

Une autre difficulté concerne les cas où les classes sont déséquilibrées ; ainsi, les jours de dépassement du seuil critique de concentration en ozone sont relativement rares.

Un modèle qui ne prédit pas ou presque pas de dépassements conduit à un taux d'erreur au plus égal au ratio du nombre de jours de dépassement. En ce sens, le taux d'erreur de classement n'est pas toujours adapté à une situation de classes très déséquilibrées.

D'autres critères ont été proposés pour intégrer cette difficulté dont le Score de Pierce basé sur le taux de bonnes prévisions :  et le taux de fausses alertes : . Le score de Pierce est alors défini par PSS =H−F et est compris entre −1 et 1 . Il évalue la qualité de la prévision. Si ce score est supérieur à 0, le taux de bonnes prévisions est supérieur à celui des fausses alertes et plus il est proche de 1, meilleur est le modèle.

Le score de Pierce a été conçu pour la prévision d'événements climatiques rares afin de pénaliser les modèles ne prévoyant jamais ces événements (H=0) ou encore générant trop de fausses alertes (F=1). Le modèle idéal prévoyant tous les événements critiques (H=1) sans fausse alerte (F=0). Une autre stratégie consiste à introduire des coûts de mauvais classement pour pondérer un score.

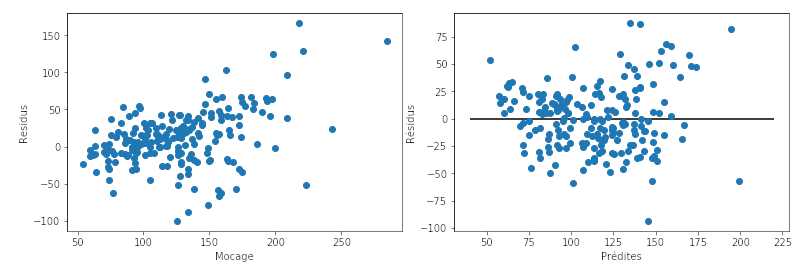
# Modélisez des données complexes

Les données permettant la prévision de la concentration (**variable quantitative**) en ozone ou de celle du dépassement du seuil (**variable qualitative**) ont servi à illustrer la procédure de choix de modèle par pénalisation Lasso. Voyons plus précisément à quels résultats cette approche conduit.

### Prévision de la concentration en ozone

Le modèle de régression, ou plutôt la sélection des variables, a été optimisé par pénalisation Lasso et optimisation du paramètre de pénalisation par validation croisée, comme cela est expliqué dans la section précédente et opéré dans le tutoriel. La comparaison des modèles est obtenue en **traçant le graphe des résidus** de la prévision de l'échantillon test en fonction des valeurs prédites.

Comme prévu, les résidus du modèle Mocage se dispersent nettement plus que ceux du modèle obtenu par adaptation statistique.



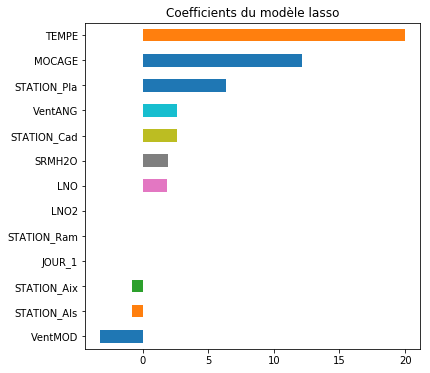
Modélisation de la concentration de l'ozone : graphe des résidus de la prévision de l'échantillon test en fonction des valeurs prédites

Il faut alors comparer les erreurs de prévisions ou risques estimés sur l'échantillon test :  1 565 (Mocage) est à comparer avec 859, de même que les  valant respectivement 0,10 et 0,52. Remarquons que l'optimisation du modèle par sélection Lasso de variable a amélioré le RMSE et le  correspondants. Ils valent respectivement 871 et 0,50 pour le modèle linéaire intégrant toutes les variables sans sélection.

Attention à la forme du nuage des résidus de la régression. La variance des résidus est plus importante pour les grandes valeurs de YY que pour les petites valeurs. Il n'y a pas homoscédasticité. En conséquence, des prévisions par intervalle de confiance ne seraient pas fiables. De plus, les résidus ne se répartissent pas de façon symétrique de part et d'autre de l'axe y=0.

Cette forme de demi-lune ou "banane" révèle une insuffisance du modèle qui ne prend pas en compte une possible composante quadratique ou interaction entre les variables. Difficile à estimer avec les possibilités offertes par la librairie scikit−learnscikit−learn, un modèle avec interactions optimisé dans R (calepin disponible) conduit à des résultats un peu meilleurs en terme de qualité de prévision, mais au détriment de la simplicité de l'interprétation et du temps de calcul pour l'optimisation du modèle.

Il est en effet intéressant de se préoccuper des **valeurs** des paramètres du modèle, afin d'évaluer l'importance des variables et de comprendre leur influence sur la concentration en ozone.

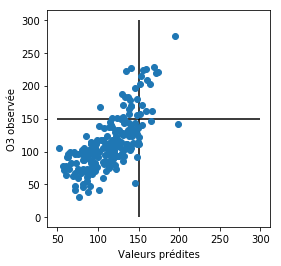


Modélisation de la concentration de l'ozone : valeurs des paramètres du modèle estimé avec pénalisation Lasso

Ces paramètres montrent des différences géographiques entre les stations. La situation est plus critique à Plan-de-Cuques (banlieue nord de Marseille) qu'à Aix-en-Provence. Ils soulignent l'importance de la température dont l'influence locale est sans doute sous-estimée dans le modèle déterministe Mocage, qui joue un rôle évidemment important dans la prévision. Un vent fort tend naturellement à **réduire** la concentration en ozone.

Attention, un paramètre nul comme pour la concentration en dioxyde d'azote ne signifie pas que cette variable n'a pas d'influence sur celle de l'ozone. Cela signifie que l'influence, si elle existe, est déjà prise en compte par les autres variables du modèle et que l'ajouter dans le modèle accroîtrait la variance des prévisions, et donc augmenterait le risque.

Une fois que la concentration en ozone est prévue, il est facile de voir si celle-ci dépasse le seuil légal dans le graphique ci-dessous associé à la matrice de confusion ;



Modélisation de la concentration de l'ozone : valeurs prédites en fonction des valeurs observées et seuils légaux pour visualiser les bonnes prévisions ainsi que les faux positifs (quadrant en bas à droite) et faux négatifs (quadrant en haut à gauche)

|  |  |  |
| --- | --- | --- |
|  | Pas de dépassement observé | Dépassement observé |
| Pas de dépassement prédit | 162 | 20 |
| Dépassement prédit | 5 | 13 |

Remarquer la **dissymétrie** de la matrice de confusion à rapprocher de la forme du nuage des résidus commentée ci-dessus. Remarquer également le nombre relativement élevé de faux négatifs : pas de dépassement prédit, alors qu'il a été observé au regard des vrais positifs. Calculer le score de Pierce à titre illustratif.

Un modèle prenant en compte les interactions apporte une légère amélioration mais la taille de l'échantillon, notamment le nombre de jours de dépassements de seuil, est trop faible pour espérer améliorer les qualités du modèle sur ces données.

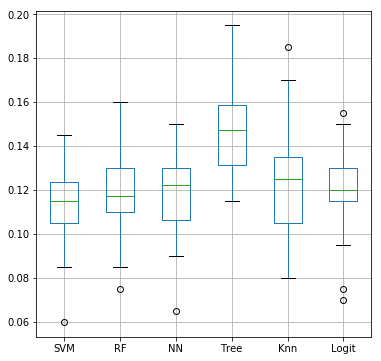
### Prévision de dépassement du seuil

Les mêmes données sont utilisées pour modéliser directement le **dépassement de seuil** sans passer par l'étape de modélisation de la concentration. Une fois optimisé par validation croisée, le modèle conduit à une prévision similaire de l'échantillon test avec la matrice de confusion ci-dessous.

|  |  |  |
| --- | --- | --- |
|  | Pas de dépassement observé | Dépassement observé |
| Pas de dépassement prédit | 162 | 18 |
| Dépassement prédit | 5 | 15 |

La qualité de prévision semble un peu meilleure mais, compte tenu de la faible taille de l'échantillon test, l'estimation du risque est peu fiable et les différences peu significatives. C'est la raison pour laquelle la procédure d'estimation du risque est itérée **B** fois en considérant différentes séparations aléatoires des échantillons d'apprentissage et de test.

Cette procédure spécifique, dite de **validation croisée Monte Carlo**, conduit à l'estimation de **B** erreurs de prévision ou risques pour comparer plusieurs algorithmes ou méthodes de prévision. Il est possible de calculer la moyenne de ces **B** erreurs, une moyenne pour chaque méthode ou encore d'afficher les diagrammes boîtes des distributions de ces erreurs.



Prévision de la concentration de l'ozone : comparaisons des performances des algorithmes par des diagrammes boîtes des distributions des erreurs

Le graphique ci-dessus compare donc plusieurs méthodes de discrimination binaire : machine à vecteurs supports, forêt aléatoire, réseau de neurones, arbre de décision, **k** plus proches voisins et régression **logistique**. Même si la taille de l'échantillon test est modeste, le résultat permet de conclure que les méthodes ne conduisent pas à des résultats significativement très différents.

Nous laisserons néanmoins de côté les arbres de décision moins performants et réseaux de neurones, **k** plus proches voisins avec des erreurs plus dispersées. Finalement, entre les SVM, un peu meilleurs mais opaques, et une régression logistique interprétable, il peut être préférable de choisir la régression logistique.

Il faudrait ajouter à ces résultats une comparaison des courbes ROC comme tracées dans la section précédente, afin de faire intervenir le choix politique du seuil de décision dans la discussion.

### Reconnaissance de l'activité humaine

Un modèle, ou plutôt 6 modèles de régression logistique, sont estimés sur les données issues des transformations des signaux enregistrés par des smartphones. En effet, par défaut, la librairie scikit−learnscikit−learn estime autant de modèles que de classes lorsque la variable YY est qualitative avec plus de 2 classes. Comme précédemment, une pénalisation Lasso est utilisée pour opérer une sélection de variables. Chaque modèle bénéficie d'une sélection de variables spécifique mais dirigée par la même valeur du coefficient **C** de pénalisation Lasso.

La sélection de variables ne conduit pas à un modèle simplifié :  l'interprétation des coefficients des 6 modèles n'est pas raisonnablement possible. Les résultats se résument finalement à une **matrice de confusion** et un taux global d'erreur de moins de 4 %, ce qui est tout à fait raisonnable et en accord avec les résultats de la phase exploratoire des données.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Marcher | Monter un escalier | Descendre un escalier | Etre assis | Etre debout | Etre couché |
| Marcher | 491 | 3 | 2 | 0 | 0 | 0 |
| Monter un escalier | 18 | 453 | 0 | 0 | 0 | 0 |
| Descendre un escalier | 4 | 5 | 411 | 0 | 0 | 0 |
| Etre assis | 0 | 4 | 0 | 430 | 56 | 1 |
| Etre debout | 2 | 0 | 0 | 12 | 518 | 0 |
| Etre couché | 0 | 0 | 0 | 0 | 0 | 537 |

Deux activités : assis vs. debout restent difficiles à discriminer.

# Mettez en pratique les concepts de la partie 3

### 

### Que faut-il retenir de cette partie du cours ?

Cette partie du cours a insisté sur la stratégie à mettre en place afin d'optimiser le choix d'une méthode ou d'un algorithme d'apprentissage. Des points fondamentaux sont à retenir :

* le critère à minimiser est une estimation du risque ou erreur de prévision ou encore erreur de généralisation ;
* ceci conduit à rechercher, pour tout algorithme ou méthode d'apprentissage dépendant d'un paramètre de complexité, une version parcimonieuse optimisant le rapport biais / variance par minimisation du risque estimé par validation croisée ;
* enfin, après optimisation, différents méthodes ou algorithmes doivent être comparés en estimant leur risque sur un échantillon test indépendant de l'apprentissage.

Cette démarche est appliquée sur deux jeux de données et déroulée dans les tutoriels à exécuter. Le deuxième exemple montre les très bons résultats de reconnaissance de l'activité à partir des données transformées issues des signaux d'un smartphone.

L'enjeu de la partie suivante est d'arriver à des résultats similaires mais à partir des signaux bruts, sans tous les calculs préliminaires coûteux pour la batterie d'un objet connecté.

### Tutoriels

Deux tutoriels sont proposés pour illustrer et compléter le cours :

* [ML4IoT-Tutorial-Ozone](https://s3-eu-west-1.amazonaws.com/course.oc-static.com/courses/5919236/ML4IoT-Tutorial-Ozone.ipynb): les sections 4.1 et 4.2 des tutoriels illustrent la modélisation par régression et régression logistique ;
* [ML4IoT-Tutorial-HAR](https://s3-eu-west-1.amazonaws.com/course.oc-static.com/courses/5919236/ML4IoT-UseCase-Har.ipynb)  : la deuxième partie du tutoriel estime une batterie de 6 régressions logistiques pour atteindre l'objectif de reconnaissance de l'activité. D'autres algorithmes d'apprentissage sont également mis en œuvre à titre de comparaison.

### Références

**Tibshirani, R.** (1996). Regression shrinkage and selection via the lasso. J. Royal. Statist. Soc B., Vol. 58, No. 1, pages 267-288).  
**Wikistat** (2018). [Sélection de modèle](http://wikistat.fr/pdf/st-m-app-linSelect.pdf), en ligne, page disponible le 17/05/2018.  
**Wikistat** (2018). [Régression logistique](http://wikistat.fr/pdf/st-m-app-rlogit.pdf), en ligne, page disponible le 17/05/2018.  
**Wikistat** (2018). [Erreur de prévision et risque](http://wikistat.fr/pdf/st-m-app-risque.pdf), en ligne, page disponible le 17/05/2018.

# Prévoyez une activité humaine par classification supervisée

Bravo ! Vous avez réussi cet exercice !

### Compétences évaluées

* Entraîner un algorithme de classification supervisée
* Expliquer la démarche et les enjeux de la classification supervisée

### Question 1

**La régression linéaire modélise une variable :**

* + 

quantitative

* + 

qualitative

*Cette notion est abordée dans le chapitre "Abordez les principes de l'apprentissage statistique".*

### Question 2

**[Suite de la question précédente] Cette variable est modélisée par des variables :**

* + 

quantitatives

* + 

qualitatives

* + 

qualitatives ou quantitatives indifféremment

*Reportez-vous au chapitre "Abordez les principes de l'apprentissage statistique".*

### Question 3

**La régression logistique modélise une variable :**

* + 

quantitative

* + 

qualitative

*Retrouvez cette notion dans le chapitre "Comprenez la classification supervisée".*

### Question 4

**[Suite de la question précédente] Cette variable est modélisée par des variables :**

* + 

quantitatives

* + 

qualitatives

* + 

qualitatives ou quantitatives indifféremment

*Reportez-vous au chapitre "Comprenez la classification supervisée".*

### Question 5

**RMSE signifie :**

* + 

Reduced Mean Square Error

* + 

Root Mean Square Error

* + 

Reduced Mean Square Estimation

* + 

Root Mean Square Estimation

*Retrouvez cette définition dans le chapitre "Abordez les principes de l'apprentissage statistique".*

### Question 6

**Quel type de pénalisation provoque une sélection de variables et donc de modèle ?**

* + 

L1

* + 

L2

* + 

 R2 ajusté

*Reportez-vous au chapitre "Abordez les principes de l'apprentissage statistique".*

### Question 7

**Qu'est-ce que la validation croisée ?**

* + 

Une estimation sans biais optimiste de l'erreur de prévision

* + 

Une estimation par pénalisation de l'erreur de prévision

*L'explication se trouve dans le chapitre "Abordez les principes de l'apprentissage statistique".*

### Question 8

**Support Vector Machine est un algorithme de :**

*Attention, plusieurs réponses sont possibles.*

* + 

classification non supervisée

* + 

classification supervisée binaire

* + 

discrimination binaire

*L'explication se trouve dans le chapitre "Comprenez la classification supervisée".*

### Question 9

**Qu'est-ce qu'une courbe ROC ?**

*Attention, plusieurs réponses sont possibles.*

* + 

Une représentation de la qualité prédictive d'un algorithme de régression linéaire

* + 

La délimitation de l'AUC

* + 

La représentation du taux de vrais positifs en fonction du taux de vrais négatifs d'une classification binaire

* + 

La représentation du taux de vrais positifs en fonction du taux de faux positifs d'une discrimination binaire

*Retrouvez cette définition dans le chapitre "Comprenez la classification supervisée".*

### Question 10

**Qu'est-ce que la validation croisée Monte Carlo ?**

* + 

Une martingale pour jouer à la roulette

* + 

Une estimation de l'erreur de prévision sur plusieurs échantillons tests aléatoires

* + 

Une estimation de l'erreur de prévision par pénalisation aléatoire

*Cette définition se trouve dans le chapitre "Comprenez la classification supervisée".*

### Question 11

**D'après le**[**tutoriel sur les données de concentration en ozone**](https://s3-eu-west-1.amazonaws.com/course.oc-static.com/courses/5919236/ML4IoT-Tutorial-Ozone.ipynb)**: si vous acceptez 10 % de faux positifs (prévision à tort de dépassement de seuil) dans vos prévisions, approximativement quelle part de vrais positifs (prévision juste de dépassement de seuil) est captée par le modèle de régression logistique ?**

* + 

60 %

* + 

70 %

* + 

80 %

*Le résultat est fourni par la lecture de la courbe ROC section 4.2. L'abscisse 0.1 correspond approximativement à l'ordonnée 0.7 de la courbe ROC associée à une valeur spécifique, à préciser, du seuil de décision.*

### Question 12

**D'après le**[**tutoriel sur les données HAR "métier"**](https://s3-eu-west-1.amazonaws.com/course.oc-static.com/courses/5919236/ML4IoT-UseCase-Har.ipynb)**: quelle est la précision obtenue par un modèle linéaire de régression logistique pour l'identification de l'activité du porteur d'un smartphone ?**

* + 

92 %

* + 

94 %

* + 

96 %

*Cette valeur est estimée par la prévision de l'échantillon test (fin de section 4.1). La valeur obtenue (96 %) est plutôt un bon résultat compte tenu de la complexité du problème.*

# Abordez les fondements de l'intelligence artificielle

**Historique**

L'**intelligence artificielle**, initialement branche de l'informatique fondamentale, s'est développée avec pour objectif la simulation des comportements du cerveau humain. Les premières tentatives de modélisation du cerveau sont anciennes et précèdent même l'ère informatique. C'est en 1943 que Mc Culloch (neurophysiologiste) et Pitts (logicien) ont proposé les premières notions de *neurone formel*.

Ce concept fut ensuite mis en réseau avec une couche d'entrée et une de sortie par Rosenblatt en 1957, pour simuler le fonctionnement rétinien et tâcher de reconnaître des formes. C'est l'origine du **perceptron**. Cette approche dite ***connexioniste*** a atteint ses limites technologiques, compte tenu de la puissance de calcul de l'époque, mais aussi théoriques au début des années 70.

L'approche connexionniste à **connaissance répartie** a alors été supplantée par une approche **symbolique** qui promouvait les **systèmes experts à connaissance localisée**, dont l'objectif était d'automatiser le principe de l'expertise humaine en associant trois concepts :

* une **base de connaissance** dans laquelle sont regroupées les connaissances d'experts humains sous forme de propositions logiques élémentaires ou plus élaborées, en utilisant des quantificateurs (logique du premier ordre) ;
* une **base de faits** contenant les observations du cas à traiter comme, par exemple, des résultats d'examens, d'analyses de sang, de salive pour des applications biomédicales de choix d'un antibiotique ;
* un **moteur d'inférence** chargé d'appliquer les règles expertes sur la base de faits afin d'en déduire de nouveaux faits, jusqu'à la réalisation d'un objectif comme le choix du traitement d'une infection bactérienne.

Face aux difficultés rencontrées lors de la modélisation des connaissances d'un expert humain, au volume considérable des bases qui en découlaient et au caractère exponentiel de la complexité des algorithmes d'inférence mis en jeu, cette approche s'est éteinte avec les années 80. Il a été montré que les systèmes basés sur le calcul des prédicats du premier ordre conduisaient à des problèmes NPNP complets.

L'essor technologique et quelques avancées théoriques :

* **estimation du gradient** par rétro-propagation du gradient proposé par Hopkins en 1982 ;
* **analogie de la phase d'apprentissage** avec les modèles markoviens de systèmes de particules de la mécanique statistique (verres de spin) résolu par Hopfield en 1982 ;

ont permis au début des années 80 de relancer l'approche connexioniste. Celle-ci a connu jusqu'au début des années 90 un développement considérable, si l'on considère le nombre de publications et de congrès qui lui ont été consacrés, mais aussi les domaines d'application très divers où elle apparaît. La motivation initiale de simulation du cortex cérébral a été rapidement abandonnée, alors que les méthodes qui en découlaient ont trouvé leur propre intérêt de développement méthodologique et leurs champs d'application.

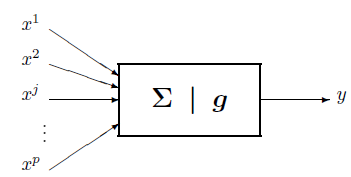
Remis en veilleuse depuis le milieu des années 90 au profit d'autres algorithmes  d'**apprentissage machine** ou plutôt statistiques : *boosting, support vector machine*..., les réseaux de neurones connaissent un regain d'intérêt et même un énorme battage médiatique sous l'appellation d'*apprentissage profond* (**deep learning**).

La taille des bases de données, notamment celles d'images issues d'Internet, associée à la puissance de calcul disponible, permettent d'estimer les millions de paramètres de perceptrons accumulant des dizaines, voire centaines de couches de neurones aux propriétés très spécifiques. Ce succès médiatique est la conséquence des résultats spectaculaires obtenus par ces réseaux en reconnaissance d'image, jeux de go, traitement du langage naturel...

**Neurone formel**

De façon très réductrice, un neurone biologique est une cellule qui se caractérise par :

* des **synapses,** les points de connexion avec les autres neurones, fibres nerveuses ou musculaires ;
* des **dentrites** ou entrées du neurones ;
* les **axones**, ou sorties du neurone vers d'autres neurones ou fibres musculaires ;
* le **noyau** qui active les sorties en fonction des stimulations en entrée.



Représentation d'un neurone formel

Par analogie, le neurone formel est un modèle qui se caractérise par un état interne s∈S, des signaux d'entrée x1,…,xp et une fonction d'activation



La fonction d'activation opère une transformation d'une combinaison affine des signaux d'entrée, **α0**, terme constant, étant appelé le *biais du neurone*. Cette combinaison affine est déterminée par un *vecteur de poids* **[α0,…,αp]** associé à chaque neurone et dont les valeurs sont estimées dans la phase d'apprentissage. Ces poids constituent la *mémoire* ou *connaissance répartie* du réseau.

Les différents types de neurones se distinguent par la nature **g** de leur fonction d'activation. Les principaux types sont :

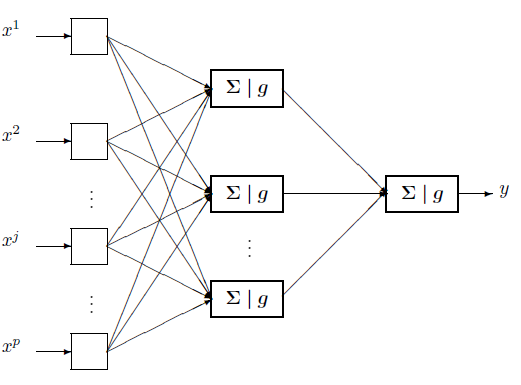
* *linéaire* **g**, la fonction identité ;
* *seuil*  ;
* *sigmoïde*  ;
* *ReLU*  (*rectified linear unit*) ;
* *radiale* ;
* ……

Les modèles linéaires, sigmoïdaux, ReLU, sont bien adaptés aux algorithmes d'apprentissage impliquant (cf. ci-dessous) une rétropropagation du gradient, car leur fonction d'activation est différentiable ; ce sont les plus utilisés. Le modèle à seuil est sans doute plus conforme à la réalité biologique, mais pose des problèmes d'apprentissage.

**Réseaux de neurones élémentaires (perceptron)**

Un **réseau neuronal**est l'association, en un graphe plus ou moins complexe, d'objets élémentaires, les **neurones formels**. Les principaux réseaux se distinguent par l'organisation du graphe (en couches, complets…… ), c'est-à-dire son architecture, son niveau de complexité (le nombre de neurones, présence ou non de boucles de rétroaction dans le réseau), par le type des neurones (leurs fonctions de transition ou d'activation) et enfin par l'objectif visé : apprentissage supervisé ou non, optimisation, systèmes dynamiques...

Nous ne nous intéresserons dans ce cours qu'à une structure élémentaire de réseau, celle dite *statique* ne présentant pas de boucle de rétroaction et dans un but d'apprentissage supervisé. Les systèmes dynamiques avec boucle de rétroaction, ainsi que les cartes de Kohonen ou cartes autoorganisatrices pour la classification non supervisée, ne sont pas abordés.



Exemple de perceptron multicouche élémentaire avec une couche cachée et une couche de sortie

Le **perceptron multicouche** (PMC) est un réseau composé de couches successives. Une **couche** est un ensemble de neurones n'ayant pas de connexion entre eux. Une couche d'entrée lit les signaux entrants, un neurone par entrée **xj**, une couche en sortie fournit la réponse du système. Selon les auteurs, la couche d'entrée qui n'introduit aucune modification n'est pas comptabilisée. Une ou plusieurs couches cachées participent au transfert.

Dans un perceptron, un neurone d'une couche cachée est connecté en entrée à chacun des neurones de la couche précédente et en sortie à chaque neurone de la couche suivante.

# Appréhendez l'apprentissage des réseaux de neurones

### 

### Fonction de transfert

Par souci de cohérence, les mêmes notations ont été conservées à travers les différentes parties. Ainsi, les entrées d'un réseau sont encore notées X1,…,Xp comme les variables explicatives d'un modèle, tandis que les poids des entrées sont des paramètres α,β à estimer lors de la procédure d'apprentissage, et que la sortie est la variable Y à expliquer ou cible du modèle.

Un perceptron multicouche réalise donc une transformation des variables d'entrée :



où α est le vecteur contenant chacun des paramètres  de la  entrée du  neurone de la  couche ; la couche d'entrée (ℓ=0) n'est pas paramétrée, elle ne fait que distribuer les entrées sur tous les neurones de la couche suivante.

De façon triviale, un réseau à un seul neurone avec une fonction d'activation linéaire est un modèle de régression linéaire. Un réseau avec un seul neurone avec une fonction d'activation logistique exécute une régression logistique.

Un théorème dit d'approximation universelle montre que cette structure élémentaire à une seule couche cachée est suffisante pour prendre en compte les problèmes classiques de modélisation ou apprentissage statistique. En effet, toute fonction régulière peut être approchée uniformément avec une précision arbitraire et dans un domaine fini de l'espace de ses variables, par un réseau de neurones comportant une couche de neurones cachés en nombre fini possédant tous la même fonction d'activation, et un neurone de sortie linéaire.

Attention, ce résultat, qui semble contradictoire avec les structures d'apprentissage profond, est théorique, il masque des difficultés d'apprentissage et de stabilité pour des problèmes complexes en très grande dimension que seules des structures complexes, multicouches, peuvent approcher de façon suffisamment robuste.

En régression (Y quantitative), la dernière couche est constituée d'un seul neurone muni de la fonction d'activation identité, tandis que les autres neurones (couche cachée) sont munis de la fonction sigmoïde. En classification binaire, le neurone de sortie est muni également de la fonction sigmoïde tandis que dans le cas d'une discrimination à m classes (Y qualitative), ce sont m neurones avec fonction sigmoïde, un par classe, qui sont considérés en sortie.

Ainsi, en régression avec un perceptron à une couche cachée de q neurones et un neurone de sortie, cette fonction s'écrit :



  avec 

### Apprentissage

Supposons que l'on dispose d'une base d'apprentissage de taille n d'observation des variables explicatives  et de la variable à prévoir Y. Considérons le cas le plus simple de la régression avec un réseau constitué d'un neurone de sortie linéaire et d'une couche à q neurones, dont les paramètres sont optimisés par moindres carrés. Ceci se généralise à toute fonction perte dérivable, et donc à la discrimination à m classes.

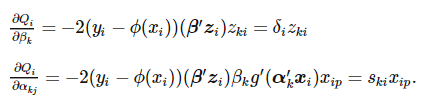
L'apprentissage est l'estimation des paramètres  et  par minimisation de la fonction perte quadratique (ou d'une fonction d'entropie en classification) :



Différents algorithmes d'optimisation sont proposés, ils sont généralement basés sur une évaluation du gradient par rétropropagation.

#### **Rétropropagation du gradient**

Il s'agit donc d'évaluer la dérivée de la fonction coût en une observation et par rapport aux différents paramètres.  
Soit .

Les dérivées partielles de la fonction perte quadratique s'écrivent : 

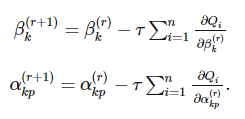
Les termes  et  sont respectivement les termes d'erreur du modèle courant à la sortie et sur chaque neurone caché. Ces termes d'erreur vérifient les équations dites de rétropropagation :



dont les termes sont évalués en deux passes. Une passe avant, avec les valeurs courantes des poids : l'application des différentes entrées xi au réseau permet de déterminer les valeurs ajustées . La passe retour permet ensuite de déterminer les  qui sont rétropropagés afin de calculer les , et ainsi obtenir les évaluations des gradients.

#### **Algorithmes d’optimisation**

Sachant évaluer les gradients, différents algorithmes, plus ou moins sophistiqués, sont implémentés. Le plus élémentaire est une utilisation itérative du gradient : en tout point de l'espace des paramètres, le vecteur gradient de Q pointe dans la direction de l'erreur croissante. Pour faire décroître Q, il suffit donc de se déplacer en sens contraire. Il s'agit d'un algorithme itératif modifiant les poids de chaque neurone selon :



Algorithme de rétropropagation élémentaire du gradient :

* Initialisation des poids   par tirage aléatoire selon une loi uniforme sur [0,1].
* Normaliser dans [0,1] les données d'apprentissage.
* Tant que Q< errmax ou niter << itermax :
  + ranger la base d'apprentissage dans un nouvel ordre aléatoire ;
  + pour chaque élément i=1,…,ni=1,…,n de la base :
    - calculer  en propageant les entrées vers l'avant,
    - l'erreur est rétropropagée dans les différentes couches afin d'affecter à chaque entrée une responsabilité dans l'erreur globale,
    - mise à jour de chaque poids 

Le coefficient de proportionnalité τ est appelé le taux d'apprentissage. Il peut être fixe, à déterminer par l'utilisateur, ou encore varier en cours d'exécution selon certaines heuristiques. Il paraît en effet intuitivement raisonnable que, grand au début pour aller plus vite, ce taux décroisse pour aboutir à un réglage plus fin au fur et à mesure que le système s'approche d'une solution.

Si l'espace mémoire de l'ordinateur utilisé est suffisant, une version accélérée de l'algorithme fait intervenir à chaque itération un ensemble (batch) d'observations pour moyenner les gradients et mises à jour des poids.

Bien d'autres méthodes d'optimisation ont été adaptées à l'apprentissage d'un réseau : méthodes du gradient avec second ordre utilisant une approximation itérative de la matrice hessienne (algorithme BFGS, de Levenberg-Marquardt), ou encore une évaluation implicite de cette matrice par la méthode dite du gradient conjugué. La littérature sur le sujet propose quantités de recettes destinées à améliorer la vitesse de convergence de l'algorithme, ou bien lui éviter de rester collé à une solution locale défavorable. D'autres heuristiques proposent d'ajouter un terme d'inertie afin d'éviter des oscillations de l'algorithme.

D'autres algorithmes encore sont des versions adaptatives, lorsque de nouvelles observations sont proposées une à une au réseau. Dans ce dernier type d'algorithme, des propriétés de dynamique markovienne (processus ergodique convergeant vers la mesure stationnaire) impliquent une convergence presque sûre : la probabilité d'atteindre une précision fixée a priori tend vers 1 lorsque la taille de l'échantillon d'apprentissage tend vers l'infini.

On pourra se reporter à l'abondante littérature sur le sujet pour obtenir des précisions sur les algorithmes d'apprentissage et leurs nombreuses variantes.

Il est important de rappeler la liste des choix qui sont laissés à l'utilisateur. En effet, même si les logiciels proposent des valeurs par défaut, il est fréquent que cet algorithme connaisse quelques soucis de convergence.

### Contrôle de la complexité

#### **Régularisation**

Dans les réseaux élémentaires, une option simple pour éviter le surapprentissage consiste à introduire un terme de pénalisation ou régularisation, comme en régression ridge, dans le critère à optimiser. Celui-ci devient alors :  .

Plus la valeur du paramètre γ (decay) est importante et moins les poids des entrées des neurones peuvent prendre des valeurs chaotiques, contribuant ainsi à limiter les risques de surapprentissage.

#### **Choix des paramètres**

L'utilisateur doit donc déterminer :

* les variables d'entrée et la variable de sortie et leur faire subir, comme pour toutes méthodes statistiques, d'éventuelles transformations, normalisations ;
* l'architecture du réseau : le nombre de couches cachées qui correspond à une aptitude à traiter des problèmes de non-linéarité, le nombre de neurones par couche cachée. Ces deux choix conditionnent directement le nombre de paramètres (de poids) à estimer et donc la complexité du modèle. Ils participent à la recherche d'un bon compromis biais/variance, c'est-à-dire à l'équilibre entre qualité d'apprentissage et qualité de prévision ;
* trois autres paramètres interviennent également sur ce compromis : le nombre maximum d'itérations, l'erreur maximum tolérée et un terme éventuel de régularisation ridge (decay) ;
* le taux d'apprentissage ainsi qu'une éventuelle stratégie d'évolution de celui-ci ;
* la taille des ensembles ou batchs d'observations considérés à chaque itération.

En pratique, tous ces paramètres ne peuvent être réglés simultanément par l'utilisateur. Celui-ci est confronté à des choix concernant principalement le contrôle du surapprentissage : limiter le nombre de neurones ou la durée d'apprentissage, ou encore augmenter le coefficient de pénalisation de la norme des paramètres. Ceci nécessite de déterminer un mode d'estimation de l'erreur : échantillon validation ou test, validation croisée ou bootstrap.

Une stratégie simple et sans doute efficace consiste à introduire un nombre plutôt grand de neurones puis à optimiser le seul paramètre de régularisation (decay) par validation croisée.

### Remarques

Les champs d'application des perceptrons multicouches (PMC) sont très nombreux : discrimination, prévision d'une série temporelle, reconnaissance de forme... Ils sont en général bien explicités dans les documentations des logiciels spécialisés.

Les critiques principales énoncées à l'encontre du PMC concernent les difficultés liées à l'apprentissage (temps de calcul, taille de l'échantillon, localité de l'optimum obtenu) ainsi que son statut de boîte noire. En effet, contrairement à un modèle de discrimination ou un arbre, il est a priori impossible de connaître l'influence effective d'une entrée (une variable) sur le système dès qu'une couche cachée intervient. Néanmoins, des techniques de recherche de sensibilité du système à chacune des entrées permettent de préciser les idées et, éventuellement de simplifier le système en supprimant certaines des entrées.

En revanche, ils possèdent d'indéniables qualités lorsque l'absence de linéarité et/ou le nombre de variables explicatives (images) rendent les modèles statistiques traditionnels inutilisables. Leur flexibilité, par l'introduction de couches spécifiques en apprentissage profond, alliée à une procédure d'apprentissage intégrant la pondération (le choix) des variables comme de leurs interactions, peuvent les rendre très efficaces.

# Entraînez un réseau de neurones profond

### 

### Préambule

Pendant les années 90 et le début des années 2000, le développement de l'apprentissage statistique, partie de l'intelligence artificielle, s'est focalisé sur les algorithmes de machines à vecteurs supports et ceux d'agrégation de modèles.

Pendant une relative mise en veilleuse du développement de la recherche sur les réseaux de neurones, leur utilisation est restée présente de même qu'une recherche fondamentale, en attendant le développement de la puissance de calcul et celle des grandes bases de données, notamment d'images, avec l'avènement du **big data**.

Le renouveau de la recherche dans ce domaine est principalement dû à Geoffrey Hinton, Yoshua Bengio, et Yann le Cun qui a tenu à jour un [célèbre site](http://yann.lecun.com/exdb/mnist/) dédié à la reconnaissance des caractères manuscrits de la base **MNIST**. La liste des publications listées sur ce site témoigne de la lente progression de la qualité de reconnaissance, de 12 % avec un simple perceptron à 1 couche jusqu'à moins de 0,3 % en 2012, par l'introduction de couches de neurones spécifiques appelées convulational neural network (ConvNet).

L'étude de ces données qui ont servi de benchmark pour la comparaison de très nombreuses méthodes, sert maintenant de données jouet pour beaucoup de tutoriels des environnements dédiés (tensorFlow, Keras, pyTorch, caffe...).

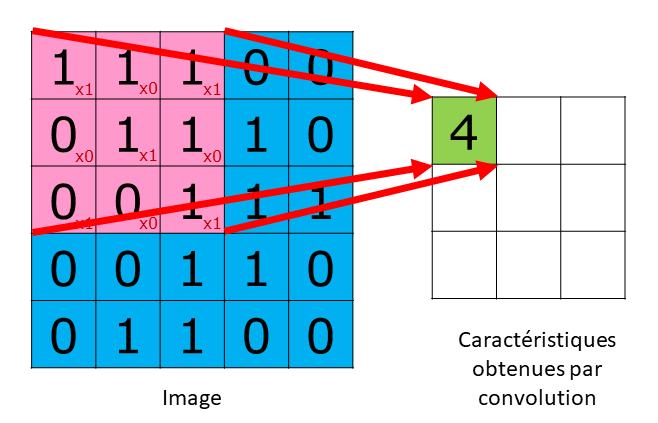
Schématiquement, trois grandes familles de réseaux d'apprentissage profond sont développées avec des ambitions industrielles, en profitant du développement des cartes graphiques (GPU) pour paralléliser massivement les calculs au moment de l'apprentissage :

* **convolutional neural networks** (ConvNet) pour l'analyse d'images ;
* **long-short term memory** (LSTM) lorsqu'une dimension temporelle ou plus généralement des propriétés d'autocorrélation sont à prendre en compte pour le traitement du signal ou encore l'analyse du langage naturel ;
* **autoencoder decoder** ou réseau diabolo en apprentissage non supervisé pour, par exemple, le débruitage d'images ou signaux, la détection d'anomalies.

Seul le premier point est développé pour illustrer les principaux enjeux.

### Reconnaissance d'images

Cette couche de neurones (ConvNet) illustrée par la figure ci-dessous ou plutôt un empilement de ces couches associé à du **max pooling**, fait émerger des propriétés spécifiques d'**invariance locale par translation**.



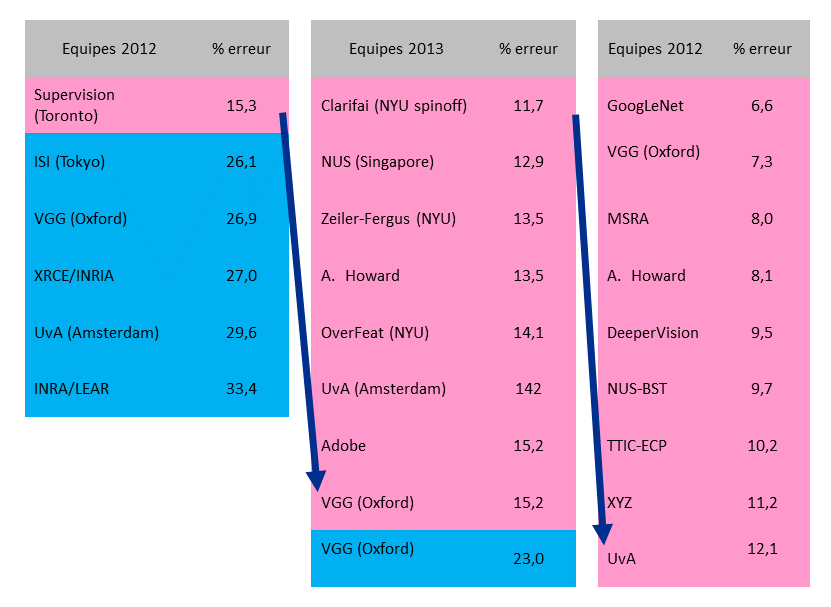
Principe élémentaire d'une couche de convolution

Cette couche applique une somme pondérée locale de chaque pixel au sein d'une fenêtre qui est translatée sur toute l'image. D'autres types de couche (pooling, subsampling) contribuent également à réduire la dimension initiale (nombre de pixels) de l'image.

Ces propriétés sont indispensables à l'objectif de reconnaissance de caractères et plus généralement d'images, qui peuvent être vues avec des positions ou sous des angles différents. C'est dans ce domaine que les résultats les plus spectaculaires ont été obtenus, tandis que l'appellation ***deep learning*** était avancée afin d'accompagner le succès grandissant et le battage médiatique associé.

La communauté de reconnaissance d'images s'est confrontée chaque année, depuis 2010, sur un jeu de données issues d'une base d'images labellisées : 15 millions d'images, 22 000 catégories hiérarchisées. De cette base sont extraites 1,2 millions d'images pour l'apprentissage avec 1 000 catégories. Les participants au concours doivent prévoir la catégorie de 15 000 images de l'échantillon test. Ce projet à l'initiative de l'Université Stanford est largement sponsorisé par Google.

Comme pour les données de reconnaissance de caractères, une progression largement empirique a conduit à l'introduction et au succès d'un réseau empilant des couches de neurones aux propriétés particulières. Cette progression est retracée dans le tableau ci-dessous :



Classements successifs (Le Cun 2016) des équipes participant au concours ImageNet. En rose, celles utilisant des réseaux de neurones profonds.

C'est en 2012 qu'une équipe utilise pour la première fois un **réseau de neurones profond**, contrairement à des traitements spécifiques et ad hoc de l'analyse d'images utilisée jusque-là. L'**amélioration était telle** que toutes les équipes ont ensuite adopté cette technologie pour une succession d'améliorations empiriques. En 2016, une équipe propose un réseau à 152 couches et atteint un taux d'erreur de 3 %, mieux que les 5 % d'un expert humain.

Ce concours est depuis lors abandonné au profit de problèmes plus complexes de reconnaissance de scènes associant plusieurs objets ou thèmes, préalables indispensables à la conduite de véhicules autonomes.

### Couches pour l'apprentissage profond

Construire un réseau d'apprentissage profond consiste à **empiler des couches de neurones** aux propriétés spécifiques dont des exemples sont résumés ci-dessous. Le choix du type, de l'ordre, de la complexité de chacune de ces couches, ainsi que de leur nombre, est complètement empirique, et l'aboutissement de très nombreuses expérimentations nécessitant des moyens de calculs et bases de données considérables :

* **fully-connected** : couche classique de perceptron et dernière couche d'un réseau profond qui opère la discrimination finale entre, par exemple, des images à reconnaître, les couches précédentes construisant, extrayant des caractéristiques (features) de celles-ci ;
* **convolution** : opère une convolution sur le signal d'entrée en associant une réduction de dimension (cf. ci-dessus : figure "Principe élémentaire d'une couche de convolution et application à une image" du paragraphe 2) ;
* **pooling** : réduction de dimension en remplaçant un sous-ensemble des entrées (sous-image) par une valeur, généralement le max ;
* **normalisation**: identique au précédent avec une opération de centrage et/ou de normalisation des valeurs ;
* **drop out** : les paramètres estimés sont les possibilités de supprimer des neurones d'une couche afin de réduire la dimension ;
* ...

### Transfert d'apprentissage

Sans bases de données très volumineuse et moyens de calcul substantiels, il est illusoire de vouloir apprendre un réseau profond impliquant l'estimation de millions de paramètres. Une mise en œuvre simple sur des données spécifiques consiste à opérer du **transfert d'apprentissage** :

* **identifier** un réseau ou modèle existant appris sur des données similaires : pour les images, considérer par exemple les versions des réseaux inception de tensorFlow ou AlexNet de Caffe ;
* **supprimer** la dernière couche du modèle dédiée à la classification ;
* **apprendre** les poids de cette seule dernière couche sur les données spécifiques.

# Mettez en pratique les concepts de la partie 4

### 

### Tutoriel

Un tutoriel est proposé pour illustrer et compléter le cours. Basé sur les données brutes directement issues d'un smartphone (**accéléromètre** et **gyroscope**), il vise à reconnaître l'activité sans transformations initiales des variables trop coûteuses pour la batterie d'un objet connecté :

* [ML4IoT-Tutorial-HAR](https://s3-eu-west-1.amazonaws.com/course.oc-static.com/courses/5919236/ML4IoT-UseCase-Har.ipynb)  : la troisième partie apprend un réseau de neurones classique puis un réseau de neurones avec couche convolutionnelle.

Ce tutoriel utilise la librairie Python  [Keras](https://keras.io/) qui permet de piloter la technologie [TensorFlow](https://www.tensorflow.org/) développée par Google mais laissée en accès libre. L'installation assez simple de **Keras** provoque implicitement celle de **TensorFlow**.

Il conduit à de bons résultats, de l'ordre de 93 % de bonnes identifications de l'activité. De meilleurs résultats nécessiteraient, comme en reconnaissance d'images, des bases de données plus volumineuses afin d'affiner l'apprentissage, et même le compléter avec la reconnaissance d'autres activités.

D'autres tutoriels sont disponibles sur le site [**github.com/wikistat**](https://github.com/wikistat/) pour approfondir cette dernière partie du cours. Le [**premier**](https://github.com/wikistat/AI-Frameworks/blob/master/MNIST/Atelier-keras-MNIST.ipynb) concerne l'utilisation d'un réseau de neurones profonds pour traiter le problème classique de reconnaissance de caractères manuscrits. Le [**deuxième**](https://github.com/wikistat/AI-Frameworks/blob/master/CatsVSDogs/Atelier-keras-CatsVSDogs.ipynb) propose une stratégie de transfert d'apprentissage pour distinguer des chats de chiens sur des images.  Attention, pour exécuter ces deux derniers tutoriels sur toutes les données, l'accès à une carte GPU est vivement recommandé.

Pour améliorer la qualité de la reconnaissance de l'activité humaine, une première approche consisterait à tester les capacités de complétion de données proposées dans Keras. Celles-ci sont introduites et mises en œuvre dans le [tutoriel](https://github.com/wikistat/AI-Frameworks/blob/master/CatsVSDogs/Atelier-keras-CatsVSDogs.ipynb) distinguant chiens et chats sur des images. Il s'agirait de **compléter la base d'entraînement** en multipliant par translation les signaux observés, afin de rendre la phase d'apprentissage moins sensible à ces translations ou déphasages.

### 

### Ce qu'il faut retenir de ce cours

Tout au long de ce cours, vous avez mis en œuvre des techniques de sciences des données et d'apprentissage automatique, dans l'objectif de valoriser des données issues de capteurs physiques, comme celles qui pourraient provenir d'un objet connecté.

La première phase est celle de **préparation et d'exploration des données**.

Attention, les données proposées dans les tutoriels étaient par principe de bonne qualité, déjà nettoyées. Ce n'est pas le cas dans la vraie vie, et cette phase essentielle pour l'obtention finale de résultats de qualité peut prendre beaucoup de temps.

La deuxième étape est celle d'**entraînement d'un algorithme** pour aboutir à une décision ou une aide à la décision automatique. Cette étape a été divisée en deux stratégies parallèles, à partir de deux jeux de données :

* la **première stratégie** analyse les signaux transformés par des fonctions expertes ou métier en traitement du signal. L'étude montre que des modèles statistiques élémentaires car linéaires conduisent à de bons résultats pour la reconnaissance de l'activité du porteur du smartphone ;
* la **deuxième stratégie** que vous venez d'accomplir dans cette quatrième et dernière partie est plus ambitieuse. L'objectif est d'**économiser** les transformations coûteuses des données pour la batterie d'un objet connecté autonome. Dans ce cas, les bonnes propriétés des réseaux de neurones profonds incorporant une ou des couches convolutionnelles permet d'aboutir à des prévisions de bonne qualité. Malheureusement, ces réseaux nécessitent des bases d'apprentissage plus **conséquentes**, plus **représentatives** de toutes les situations qui peuvent être rencontrées, en l'occurrence plus représentatives des possibilités de déphasage des signaux. Un peu plus de travail serait donc nécessaire pour atteindre la qualité de résultats de la première stratégie.

Si ce travail aboutit, la dernière phase serait celle d'embarquement de l'IA ainsi configurée dans l'objet connecté. C'est ce type d'application que vous pouvez trouver pour l'identification de la présence d'un visage sur l'écran d'un appareil photo ou d'un smartphone. C'est encore ce type d'IA qui permet de détecter la chute du porteur ou de proposer un électrocardiogramme sommaire au porteur d'une montre connectée.

**Attention :** l'objectif de ce cours se limite à l'illustration de ce que peut apporter la science des données, ou plus précisément un algorithme d'apprentissage automatique pour valoriser des données issues d'un objet connecté équipé de capteurs. Il ne propose pas un tour d'horizon complet des algorithmes d'apprentissage automatique à rechercher dans d'autres cours et [**tutoriels**](https://github.com/wikistat/). Il en existe de très nombreux :

* [**régression PLS**](http://wikistat.fr/pdf/st-m-app-sparse-pls.pdf) ;
* [**k plus proches voisins**](http://wikistat.fr/pdf/st-m-app-add.pdf) ;
* [**arbres binaires de décision**](http://wikistat.fr/pdf/st-m-app-cart.pdf) ;
* [**machines à vecteurs supports**](http://wikistat.fr/pdf/st-m-app-svm.pdf) ;
* [**forêts aléatoires**](http://wikistat.fr/pdf/st-m-app-agreg.pdf) ;
* diverses versions de [***boosting***](http://wikistat.fr/pdf/st-m-app-agreg.pdf).

Il est important de garder en mémoire qu'aucun n'est universellement meilleur, **pas même l'apprentissage profond** ; tout dépend des données, de leurs propriétés. Ainsi, un modèle linéaire suffit à résoudre la discrimination des comportements humains, alors qu'un modèle non linéaire très complexe est nécessaire pour atteindre le même objectif à partir des signaux bruts.

Vous êtes arrivé à la fin de ce cours, félicitations ! Vous avez maintenant toutes les clés pour :

* mettre en place les outils nécessaires à la science des données ;
* exécuter une analyse exploratoire multidimensionnelle ;
* expliquer la démarche et les enjeux de la classification supervisée ;
* entraîner un algorithme de classification supervisée ;
* expliquer l'évolution de l’IA ;
* entraîner un algorithme d’apprentissage profond.

N'oubliez pas de réaliser les exercices évalués à la fin de chaque partie, ils vous permettront de valider ces compétences. Je vous souhaite une bonne continuation dans tous vos projets !

# Appréhendez l'apprentissage profond (deep learning)

Bravo ! Vous avez réussi cet exercice !

### Compétences évaluées

* Entraîner un algorithme d’apprentissage profond
* Expliquer l'évolution de l’IA

### Question 1

**L'intelligence artificielle a été inventée dans les années :**

* + 

1950

* + 

1970 avec les systèmes expert

* + 

1980 avec les réseaux de neurones

* + 

2000 avec le deep learning

*Cette information est donnée dans le chapitre "Appréhendez l'Intelligence artificielle".*

### Question 2

**Un neurone formel peut opérer...**

*Attention, plusieurs réponses sont possibles.*

* + 

une combinaison linéaire des variables d'entrée

* + 

une transformation logarithmique d'une combinaison linéaire des variables d'entrée

* + 

une transformation ReLU d'une combinaison linéaire des variables d'entrée

*Retrouvez ces éléments dans le chapitre "Appréhendez l'Intelligence artificielle".*

### Question 3

**Une couche d'un perceptron est un ensemble de neurones :**

*Attention, plusieurs réponses sont possibles.*

* + 

n'ayant pas de connexions entre eux

* + 

entièrement connectés

* + 

connectés aux couches précédentes et suivantes

*L'explication se trouve dans le chapitre "Appréhendez l'Intelligence artificielle".*

### Question 4

**L'algorithme de rétropropagation du gradient :**

*Attention, plusieurs réponses sont possibles.*

* + 

est un algorithme d'apprentissage d'un réseau de neurones

* + 

converge vers un optimum global

* + 

opère une descente de gradient

*Reportez-vous au chapitre "Appréhendez l'apprentissage des réseaux de neurones".*

### Question 5

**Un réseau de neurones :**

* + 

doit être optimisé pour éviter le surapprentissage

* + 

n'est pas sensible à ce problème

*Cette information se trouve dans le chapitre "Appréhendez l'apprentissage des réseaux de neurones".*

### Question 6

**La complexité d'un réseau de neurones est régulé par :**

*Attention, plusieurs réponses sont possibles.*

* + 

le nombre d'itérations

* + 

le nombre de neurones

* + 

une pénalisation de type ridge ou L2

*Toutes ces options sont possibles et combinables. Voir chapitre "Appréhendez l'apprentissage des réseaux de neurones".*

### Question 7

**Le succès des réseaux de neurones profonds en reconnaissance d'image est dû :**

*Attention, plusieurs réponses sont possibles.*

* + 

à l'apparition des couches convolutionnelles

* + 

à l'apparition des réseaux récurrents (LSTM)

* + 

au développement du langage Python

* + 

à la mise à disposition de masses d'images labellisées (ImageNet)

*Retrouvez ces éléments dans le chapitre "Appréhendez l'apprentissage profond ou deep learning".*

### Question 8

**Quelle est la proposition correcte concernant Keras ?**

* + 

Keras est un réseau de neurones déjà appris utilisable sur d'autres données par transfert learning

* + 

Keras est une librairie Python pour utiliser TensorFLow

* + 

Keras ne fonctionne qu'avec des cartes GPU

*Cette définition se trouve dans le chapitre "Appréhendez l'apprentissage profond ou deep learning".*

### Question 9

**D'après le**[**tutoriel sur les données HAR brutes**](https://s3-eu-west-1.amazonaws.com/course.oc-static.com/courses/5919236/ML4IoT-UseCase-Har.ipynb)**: quelle est la précision obtenue avec un réseau de neurones profond pour l'identification de l'activité du porteur d'un smartphone ?**

* + 

92 %

* + 

94%

* + 

96 %

*Cette valeur est estimée par la prévision de l'échantillon test (fin de section 5.3). La valeur obtenue (92 %) sans aucune transformation des données est tout à fait raisonnable compte tenu de la complexité du problème.*

### Question 10

**Sur ces données, quelle serait la priorité pour améliorer la qualité de la prévision avec un réseau de neurones profond ?**

* + 

Augmenter le nombre de couches convolutionnelles

* + 

Augmenter le volume des données

* + 

Utiliser une carte GPU

*Une carte GPU réduit le temps d'apprentissage mais ne permet pas, à elle seule, d'améliorer la précision. La première chose à faire serait d'augmenter la taille de l'échantillon d'apprentissage en recueillant plus de données avant, si cela s'avère utile, de penser à ajouter des couches.*