Semaine 4 : Machine Learning supervisé

Cette semaine de cours, ainsi que les quatres suivantes sont consacrées au machine learning et aux statistiques de manière plus générale. Nous commencerons par nous intéresser au machine learning supervisé.

#### Qu’est ce que le machine learning supervisé ?

Le machine learning une science multidisciplinaire qui a pour ambition de permettre à une machine de résoudre des problèmes complexes qui ne peuvent être appréhendés par des algorithmes simples. Par exemple prédire (prendre un exemple qu’on verra en exercice) est un problème qui ne peut être résolu par une série de règles (eg est un homme qui a trois ans d’ancienneté sur le site etc).

Le **machine learning supervisé** est une branche de cette discipline qui vise à résoudre des problèmes pour lesquels on dispose d’exemples déjà résolus. Par exemple on rassemble des données sur un échantillons de logements à San Francisco qui décrivent leur localisation, divers caractéristiques et le montant du loyer, si notre problème est d’estimer le montant du loyer d’un logement qui n’est pas dans notre base, on construira à partir de nos données un modèle qui estime le montant du loyer en fonction des caractéristiques du logement et on appliquera ce modèle au logement inconnu pour en estimer le loyer. Ce problème relève de l’**apprentissage supervisé** car au moment de construire notre modèle on connaissait les valeurs prises par la **variable** que l’on souhaite estimer, qu’on appelle la **variable cible**.

#### Pourquoi faire du machine learning?

Les objectifs de la modélisation statistiques supervisée, une fois qu’on a posé un problème, choisi une variable cible et rassemblé un certain nombre de variables explicatives, peuvent se rassembler en trois grandes catégories non exclusifs :

* Description : On peut chercher à comprendre les relations qui peuvent exister entre la variable cible et les variables explicatives afin par exemple de sélectionner celle qui sont le plus pertinentes, où obtenir une visualisation des comportements dans la population observée (attention ici population est employé comme un terme statistique et peut aussi bien désigner des personnes, des pays, des transactions financières etc…)
* Explication : Lorsqu’on a une connaissance à priori du sujet traité, comme c’est souvent le cas en économie ou en biologie par exemple, l’objectif est de construire un test qui permet de vérifier ou confirmer des résultats théorique dans des situations pratiques.
* Prédiction : Ici on met l’accent sur la qualité des estimateurs et des prédicteurs, on ne cherche pas nécessaire à modéliser le mieux possible la population observée mais construire un modèle qui permette de produire des prédictions fiables pour de futures observations.

# Révision régressions linéaires

Dans cette section nous nous intéresserons aux modèles de régression, qui rassemblent les solutions qui permettent d’estimer une variable cible numérique et **continue**. On s’intéressera en particulier à l’estimation du prix de l’immobilier à Boston, grâce à un jeu de données fourni dans le package scikit learn de python.

## Régressions linéaires simples

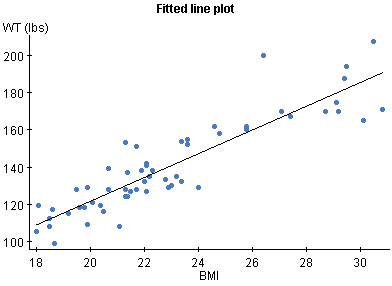
### Définition

Les modèles de régressions linéaires simples sont fondés sur l’équation linéaire suivante :

ou

Ici représente la variable cible, la variable dont on souhaite estimer la valeur, est la variable explicative que l’on a choisi pour estimer la variables cible, sont respectivement l’intercept (c’est à dire le niveau 0 de Y lorsque X vaut 0), le coefficient associé à X (c’est le paramètre du modèle qui mesure l’influence de X sur Y, si X augmente de 1, Y augmentera de , et enfin l’erreur ou résidu du modèle. En effet, l’équation ci-dessus est la représentation d’un modèle statistique, il n’a pas la prétention d’être exact, il est vrai en moyenne, voilà qui explique la présence du résidu.

En fonction des individus (ou nuage de point), votre modèle va trouver la ligne qui se rapproche le plus possible tous les individus à la fois. Voici ce à quoi cela ressemble visuellement :



#### Variables dépendantes

En Machine Learning, on distingue toujours les **variables dépendantes/variables cibles** des **variables indépendantes/variables explicatives.** Les variables dépendantes sont les éléments que vous cherchez à prédire. Dans l’équation du dessus, cela correspond à .

#### Variables Indépendantes

Les variables indépendantes, représentées par sont vos prédicteurs ou les facteurs qui vont permettre de déterminer la valeur de . Par exemple, si nous essayons de prédire le salaire de quelqu’un en fonction de son nombre d’années d’expérience. La variable indépendante correspond au nombre d’années d’expérience.

#### Coefficient

Le coefficient représente la *pente* ou le poids qu’aura votre variable indépendante dans votre équation.

#### Constante

Enfin, La constante représente l’endroit où votre ligne va commencer si . Dans le cas de prédiction de salaires par rapport aux années d’expérience, même si vous avez 0 années d’expériences (), vous aurez tout de même un salaire minimum de départ et non 0.

#### Résidu

Le résidu, souvent noté correspond à l’erreur commise lors de la modélisation. Cette erreur correspond à toutes l’information qui n’est pas expliquée par le modèle, on suppose souvent que l’erreur suit une loi de probabilité particulière.

### Les hypothèses derrière une régression linéaire

Quand vous construisez un modèle de ML, vous devrez être conscient des hypothèses que vous devez respecter pour que votre modèle fonctionne bien. Dans le cas inverse, vous aurez des performances déplorables. Voici les hypothèses d’un modèle de régression linéaire simple :

#### Linéarité

La première hypothèse est simple. Il faut que vos points suivent à peu près une droite. En d’autres termes, vous devez vous assurer que votre variable dépendante suive une croissance linéaire à mesure que vos variables indépendantes augmentent.

#### Homoscedasticité

Au delà de la complexité du modèle en lui même, cela veut dire que la variance de vos points doit être relativement la même. Si vous avez une variance énorme, cela veut dire que vous avez des points très éloignés les uns des autres et que donc il sera difficile d’avoir une ligne qui soit représentative de votre dataset.

#### Normalité des variables

Les points doivent avoir une distribution normale (ou du moins à peu de choses près). Vous n’aurez cependant rarement une distribution normale de vos points. Le tout est d’avoir une moyenne, une médiane et un mode qui ne soit pas trop éloignés.

### Estimation

#### Maximisation de la vraisemblance

##### Définition

La vraisemblance d’une famille de variables aléatoire est une fonction qui donne pour chaque réalisation possible de chaque la probabilité que cette combinaison de réalisations se produise. Dans le cas de la modélisation statistique, on dispose déjà de données, on connait donc déjà les réalisations de chaque variable aléatoire (c’est à dire la valeur des variables explicatives pour chaque observation). Il s’agit donc de trouver les paramètres qui permettront de rendre les plus probables possible les observations qu’on a à notre disposition.

La vraisemblance statistique est une fonction de probabilités conditionnelles (c’est une probabilité dont la loi dépend de paramètres). Soit un vecteur de variables aléatoires et l’ensemble des paramètres dont X dépend. La vraisemblance de X s’écrit :

###### 

###### Avec

###### 

##### Estimation par maximum de vraisemblance

Une manière d’estimer les paramètres d’un modèle est de maximiser la fonction de vraisemblance correspondante. Dans le cas de la régression linéaire simple, cette fonction est:

On peut obtenir cette équation de la vraisemblance grâce à l’hypothèse selon laquelle l’erreur suit une loi normale centrée () d’écart-type . On doit donc trouver le maximum (s’il existe) de cette fonction de vraisemblance, pour ce faire on applique un logarithme de chaque côté de l’équation afin d’obtenir une somme.

On constate que l’équation dépend uniquement des paramètres du modèle et il nous reste à trouver les valeurs pour lesquelles est maximal.

#### La méthode des moindres carrés

##### Définition

Vous vous demandez sûrement comment on sait que la ligne de notre modèle est celle qui se rapproche “Le plus” de chacun des points de notre dataset. Et bien, c’est grâce à la méthode des *moindres carrés*. Nous n’allons pas aller trop loin dans la démonstration de la formule. Ce qu’il y a à comprendre est que l’algorithme va chercher la distance minimum possible entre chaque point dans votre graphique via cette formule :

Dans le cas de la régression linéaire simple, l’estimation par le maximum de vraisemblance ou celle par les moindres carrés revient à trouver la valeur extrême de la même équation.

Dans cette équation, représente chaque individu (ou point) de votre dataset alors que représente la prédiction de votre modèle.

Après plusieurs itérations, votre algorithme est capable de trouver le nombre minimum dans cette formule et donc avoir la meilleure ligne possible qui décrit votre dataset.

## Régression linéaire multiple

### Définition

#### Pluralité des variables indépendantes

La plupart du temps, vous n’aurez pas qu’un seul facteur qui va vous permettre de prédire votre variable dépendante. Par exemple, vous pouvez prédire le salaire de quelqu’un avec son nombre d’années d’expérience mais surement aussi le type de diplôme, le secteur dans lequel la personne travaille, le sexe, le pays etc.

C’est la seule différence entre la régression linéaire simple et multiple. Vous ajoutez des variables indépendantes dans l’équation.

#### Notations mathématiques du problème

En notant la variable cible, les variables explicatives, les paramètres du modèle et le vecteur des résidus, le modèle de régression linéaire multiple s’écrit :

On peut également écrire le problème sous forme matricielle de la manière suivante :

Où est un vecteur de dimension , est une matrice de dimensions , est un vecteur de dimensions et est un vecteur de dimensions .

#### Matrice de variance/covariance

La matrice de variance covariance d’une collection de variables aléatoires indexées par de à est une matrice carré de taille dont les éléments sont :

Les éléments le long de la diagonale de la matrice de variance/covariance sont les variances respectives de chaque variable aléatoire. Les autres éléments sont :

Les covariances entre les différentes variables aléatoires.

#### Résolution matricielle par le maximum de vraisemblance

Le calcul de l’estimateur du maximum de vraisemblance est un exercice classique en statistiques, nous présentons donc le calcul ici pour celles et ceux qui sont familier ou souhaiteraient se familiariser avec le calcul matriciel.

Selon les hypothèses nécessaire pour pouvoir utiliser un modèle linéaire multiple, suit une loi Normale centrée () et de matrice de covariance diagonale . Ce qui nous amène à l’équation suivante :

est une matrice diagonale, d’où

On applique le logarithme qui est croissant et ne change donc pas le problème d’optimisation considéré :

On cherche à trouver la valeur de qui maximise l’équation ci-dessus, ce qui revient à trouver le minimum de la valeur suivante :

est un scalaire (c’est à dire un nombre réel de dimension 1), il est donc égal à son transposé , d’où :

On dérive par rapport à et on obtient :

Cette solution n’est bien définie que si est une matrice inversible, ce qui est vrai si les variables explicatives ne sont pas colinéaires et si (le nombre de variables explicatives est inférieur au nombre d’observations).

### Les hypothèses pour une régression linéaire multiple

#### Tout ce qu’il y a dans la régression linéaire simple plus NON colinéarité

Comme vous l’imaginez déjà, les régressions linéaires multiples vont suivre les mêmes hypothèses que les régressions linéaires simples car après tout, on ajoute simplement un peu de complexité. La seule chose que vous devez ajouter dans les hypothèses est la *non-colinéarité* des variables indépendantes.

Prenons un exemple, si vous essayez de prédire le salaire de quelqu’un en fonction de son âge et de son nombre d’années d’expérience, vous allez rencontrer un problème. En effet, entre l’âge et le nombre d’années d’expérience, il est tout à fait possible d’établir une relation entre les deux puisque, logiquement, plus vous êtes âgé, plus vous avez d’années d’expérience.

Si vous avez une colinéarité dans votre modèle, celui-ci va être biaisé et inutilisable car on ne pourra pas savoir quelle variable influence vraiment votre variable dépendante.

### Variables factices (*Dummy Variables*)

#### Rappel : Variables catégoriques

Pour comprendre ce que sont des variables factices, rappelons d’abord ce que sont des variables catégories. Ce sont simplement des données qualitatives. Pensez par exemple à des pays, des tailles de chaussures etc. Vous pourriez techniquement créer une catégorie pour chaque variable.

#### Encoder des variables catégoriques

Dans des régressions, vous ne pouvez pas avoir des données textes comme variables, seuls les nombres sont acceptés. C’est pourquoi on encode les variables catégoriques et les remplace par des nombres : 0 ou 1

#### Le piège des variables factices

Une fois que vous avez encodé vos variables factices, vous n’allez pas toutes les ajouter dans votre équation car vous aurez un problème de colinéarité entre votre dernière dummy variable et votre première car l’une sera l’opposé de l’autre. Ce que vous ferez donc est d’ajouter toutes les dummy variables et vous en **enlèverez 1 dans votre équation.**

En effet, une des hypothèses du modèle de régression linéaire multiple est la non colinéarité des variables explicatives. C’est à dire qu’il ne doit pas exister de combinaison linéaire qui lie les variables entre elles.

Une combinaison linéaire entre des variables est une somme de la forme :

Où les coefficients sont des nombres réels. Les variables sont dites linéairement liées, ou colinéaires, si il existe un groupe de coefficients tels que :

Dans le cas des variables factices, la somme de toutes les variables vaut toujours 1 cas chaque observation appartient nécessairement à une des modalités, de fait :

Et les variables sont colinéaires, voilà pourquoi il est essentiel de supprimer une des variables factices afin d’éviter la colinéarité des variables, la variable retirée correspondra à la modalité par défaut, et les coefficients associés aux autres variables représenteront l’influence de chaque modalité par rapport au niveau par défaut.

### Sélection de variables et choix de modèle

La caractéristique principale de la régression linéaire multiple est qu’elle fait appel à plusieurs variables explicatives conjointement. Dès lors la question qui naturellement se pose est : quelles variables dois-je utiliser pour construire le meilleur modèle possible en fonction de mes objectifs ? Cette question nous amène à introduire des critères d’évaluation de modèles et des méthodes de sélection de variables.

#### Evaluation des modèles de régression linéaire multiple

Certains des critères d’évaluation présentés ci-dessous pourront être utilisés pour d’autres modèles que la régression linéaire multiple, il est donc d’autant plus important de les introduire dès maintenant et de bien retenir leurs interprétations respectives.

##### L’analyse de la variance (ANOVA pour Analysis Of Variance)

L’analyse de la variance permet de quantifier les performances d’un modèle statistique en termes d’erreur d’estimation, les différentes valeurs dont nous allons parler maintenant vont être utilisés pour construire d’autres indicateurs.

* SST : Sum of Square Total est un indicateur de la dispersion des valeurs de la variable cible (dont les valeurs sont notées ) sur la population considérée, ce qui s’écrit mathématiquement :

C’est la somme des écart à la moyennes au carré des valeurs prises par la variable cible pour les observations considérées.

* SSE : Sum of Square Explained est un indicateur qui représente la quantité de dispersion de la variable cible qui est expliquée par le modèle, ce qui s’écrit :

C’est la somme des écarts à la moyenne au carré entre les estimations du modèle pour chaque observation et la moyenne de la variable cible pour la population considérée.

* SSR : Sum of Squared Residual est un indicateur qui quantifie l’erreur commise par le modèle, ou en d’autres termes la portion de la dispersion de la variable cible qui n’est pas expliquée par le modèle, d’où l’idée de résidu. Sa formule est la suivante :

Il est essentiel de bien comprendre ces valeurs car elles vont nous permettre de construire toutes les métriques d’évaluation des modèles de régression linéaire multiple que nous allons voir maintenant.

Pour résumer, SST est la variance totale de la variable cible, qui peut se décomposer en deux composantes : SSE la variance expliquée par le modèle, qui correspond à la quantité de variance de nos estimation par rapport à la moyenne réelle de la population observée et SSR qui est la somme des carrés des écarts entre nos estimations est les valeurs réelles de la variables cible. En d’autres termes SST est la quantité d’information totale, SSE est l’information expliquée par le modèle et SSR l’information qui reste à expliquer, ou l’erreur commise.

##### F-Statistique de Fisher

Un test statistique est un procédé par lequel on cherche à montrer si une hypothèse est confirmée ou infirmée par les données à notre disposition. Cette hypothèse de test, encore appelée hypothèse nulle et notée se traduirait par des conséquence sur les propriétés des données observées si elle est effectivement vérifiée. Ces propriétés sont résumées par une statistique de test dont la valeur donne une idée de la probabilité qu’a d’être vraie.

La F-statistique de Fisher permet de tester la véracité des hypothèses suivantes :

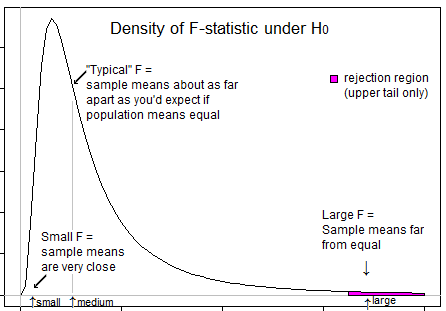
* Dans le cas où l’on applique le test de Fisher au modèle dans sa globalité, l’hypothèse nulle, notée , est “les variables choisies pour construire le modèle ne sont pas conjointement significatives pour décrire la variable cible”. Si l’hypothèse est vraie, la F-statistique devrait suivre une loi de probabilité de distribution F de paramètres où est le nombre d'observation utilisées pour construire le modèle. Or si la valeur de la F-statistique , notée se trouve en dehors des régions les plus probables de la distribution, on peut alors rejeter l’hypothèse nulle et conclure que le modèle choisi a un réel pouvoir explicatif sur la variable cible.

Mathématiquement, la F-statistique s’écrit :

Le F-test peut également comparer deux modèles emboités (le modèle 1 qui inclus variables explicatives et le modèle 2 qui inclus variables explicatives dont les variables du modèle 1 et . Dans ce cas la F-statistique suit une loi F de paramètres si l’hypothèse selon laquelle le modèle le plus simple (modèle 2) parmi les deux modèles décrit mieux la variable cible est vérifiée. La formule mathématique de F est alors :

Si la valeur de F se situe dans une région peu probable de la F-distribution qu’elle est censée suivre, alors on rejette l'hypothèse et le test suggère que le modèle 2, plus complexe, apporte une information supplémentaire significative par rapport au modèle 1, plus simple.

Graphiquement le F-test peut s’illustrer ainsi :



En noir, nous représentons la densité de la loi F, comme dans tout test on défini un niveau compris entre 0 et 1 qui va influencer la taille de la zone de rejet de l’hypothèse. Très souvent on prend lorsqu’aucune connaissance métier ne peut nous aider à moduler nos exigences de précision. Le test F est unilatéral, seules de grandes valeurs de F permettront de rejeter l’hypothèse. Plus précisément si la valeur de F se situe dans la partie supérieure de la distribution attendue équivalent à 5% de probabilité, alors on peut dire que l'hypothèse est rejetée à 95%.

Cette première métrique permet de tester l’hypothèse selon laquelle les variables explicatives n’ont pas d’influence sur la variable cible, nous allons maintenant nous intéresser à des métriques qui indiquent les performances du modèle.

* (R carré)

qu’on appelle en anglais R-squared, est une statistique qui quantifie le pouvoir explicatif du modèle par rapport à la variable cible.

est monotone croissant avec le nombre de variables explicatives qu’on ajoute au modèle. Il varie entre 0 et 1, si le modèle est peu pertinent, la somme des carrés résiduels sera proche de la somme des carrées totaux et sera plus proche de 0, au contraire, si le modèle permet d’expliquer fidèlement la variable cible, alors sera plus proche de 0 et sera plus proche de 1. Ainsi mécaniquement à chaque ajout de variable au modèle, la prédiction de , la variable cible, sera meilleur et sera plus élevé. De fait, est un indicateur de performance qui permet uniquement de comparer deux modèles qui ont le même nombre de variables explicatives.

est une version modifiée de qui pénalise le nombre de variables explicatives sélectionnées pour construire le modèle. Sa formule mathématique est :

Où est le nombre de variables explicatives utilisées et le nombres d’observations utilisées. La croissance de en fonction de est compensée par la décroissance de en fonction de . En conséquence, si l’apport d’information d’une variable explicative n’est pas assez important, alors va décroître. De fait il est possible d’utiliser cet indicateur pour comparer entre eux les performances de modèles qui n’ont pas nécessairement le même nombre de variables explicatives.

#### Sélection de modèle.

Lorsqu’on a à notre disposition variables explicatives, le nombre de modèles qu’il est possible de construire peut être décompté de la manière suivante : on considère les variables une par une et on se demande à chaque fois si on la sélectionne pour le modèle ou non, de fait si on a une variable on peut construire deux modèles, celui avec ou celui sans . Si on ajoute plus de variables explicatives possibles, on construit comme un arbre, les deux premières branches corresponde au fait de sélectionner ou non la variable , qui elles même se divisent en deux branches pour sélectionner ou non et ainsi de suite. De fait, si on a à notre disposition variables explicatives, on peut potentiellement construire modèles avec. On ne peut pas en pratique, lorsque le nombre de variables explicatives est grand, explorer les modèles qu’il est possible de construire à l’aide de ces variables afin de sélectionner le meilleur. Différentes méthodes existent qui permettent d’éviter de faire appelle à la force brute.

#### Pas à pas

La sélection pas à pas s’articule en trois variantes :

* Sélection (forward) : On ajoute une à une les variables au modèles en sélectionnant à chaque pas celle dont la p-value du test de Fisher qui compare deux modèles emboîtés et la plus faible. On arrête lorsque toutes les variables sont utilisées ou si la p-value minimale devient supérieur à une valeur seuil, qui par défaut est fixé à 0.5.
* Elimination (backward) : On démarre cette fois ci avec un modèle utilisant toutes les variables explicatives. A chaque étape, la variable associée à la plus grande p-value associée au test de Fisher est éliminée du modèle. La procédure s’arrête lorsque toutes variables restantes présentent des p-value supérieure à un seuil fixé par défaut à 0.1, mais qu’on peut adapter en fonction des besoins en précision du problème considéré.
* Mixte (stepwise) : Cet algorithme alterne entre une étape de sélection et une étape d’élimination après chaque ajout de variable afin de retirer d’éventuelles variables qui seraient devenues moins pertinente en présence de celles qui ont été ajoutées.

#### Par échange

Cette méthode a pour but de trouver le meilleur modèle pour chaque niveau (pour chaque nombre de variables explicatives indépendantes) en commençant par le niveau 1. A chaque niveau on sélectionne une variable non encore incluse dans le modèle qui maximise l’accroissement de . Puis il échange tour à tour une variable présente dans le modèle avec une variable extérieure au modèle et conserve la configuration qui maximise .

Le même algorithme peut être légèrement modifié pour sélectionner l’échange des deux variables qui réalise le plus petit accroissement de , l’idée derrière cette variante est d’explorer plus de modèles différents et ainsi d’atteindre un meilleur optimum.

#### De manière globale

L’algorithme de Furnival et Wilson permet de comparer tous les modèles possibles avec pour objectif l’optimisation d’un des critères d’évaluation parmi et .

### Dernières remarques

Les modèles linéaires sont très sensibles aux valeurs extrêmes qui peuvent être présentes dans un jeu de données, le pré-traitement de votre base d’apprentissage est donc essentiel afin de ne pas voir vos résultats complètement faussés.

Les méthode d’évaluation et de sélection de modèles introduites ci-dessus sont parfaitement valide pour l’ensemble des modèles linéaires, ainsi que la régression logistique que nous aborderons par la suite.

# Les régressions linéaires régularisées

Les modèles linéaires sont populaires pour estimer une variable cible continue qui dépend de variables explicatives . En général le volume d’observation (souvent noté ) est grandement supérieur au nombres de variables explicatives , cependant, dans certains cas on se retrouve dans la situation inverse. On dispose d’un nombre de variables explicatives bien supérieur au nombre d’observations, c’est souvent le cas, par exemple dans le domaine des statistiques génétiques, où les ressources de temps et d’argent ne permettent pas de séquencer l’ADN de plus d’un millier d’individus en général, tandis que le nombre de gènes que compte notre ADN pour produire une protéine est d’environ 20 000! Les modèles linéaires classiques ne sont pas bien adaptés pour ce type de problème à la dimensionnalité très élevée. C’est pourquoi nous aborderons dans ce cours de nouveaux modèles linéaires qui contournent cette difficulté.

## Modèle Ridge

### Pourquoi le modèle linéaire échoue?

Comme on l’a vu précédemment, le modèle linéaire s’écrit de la manière suivante :

Ou sous forme matricielle :

Si nous avons , dans la première représentation il s’agit de résoudre équations à l’aide de paramètres que sont les , or un système d’équations avec un nombre plus élevé de paramètres que d’équations indépendantes est non déterminé et possède une infinité de solutions.

En vision matricielle, est une matrice de dimensions alors que ses lignes et colonnes sont des combinaisons linéaires issues de vecteurs, le rang de cette matrice est donc inférieur ou égal à , elle est donc non inversible et la résolution matricielle que nous avons montré plus haut n’est plus valide.

### Quelle solution pour palier cette difficulté ?

#### Qu’est ce qu’une fonction de coût ?

Une **fonction de coût** est un concept qui servira énormément par la suite puisqu’il est omniprésent dans l’univers du machine learning. La fonction de coût est la quantité mathématique que l’on souhaite minimiser lors de l’optimisation d’un modèle statistique. Dans le cadre d’une régression linéaire multiple, la fonction de coût est :

qui est la norme euclidienne pour les vecteurs (c’est à dire la racine carré de la somme des ses composants au carré) au carré. De fait l’estimateur qui donne les coefficients du modèle est le vecteur qui minimise la fonction de coût, ce qui s’écrit mathématiquement :

Cette fonction de coût permet d’obtenir un estimateur peu biaisé, mais dans le cas présent et le nombre de variables explicatives, la variance est très élevée car beaucoup de paramètre ne seront pas pertinents pour l’estimation du modèle.

C’est pourquoi on introduit la notion de **pénalité,** c’est une modification qu’on apporte à la fonction de coût afin de maîtriser l’arbitrage entre **biais vs variance**.

#### Biais vs Variance

Le biais et la variance sont deux notions omniprésentes en statistiques et particulièrement en machine learning lorsqu’on cherche à faire de l’estimation ou de la prédiction.

Considérons un problème où est la variable cible la matrice des variables explicatives et le terme d’erreur de moyenne et de variance .

On souhaite modéliser à l’aide des variables explicatives et on suppose qu’il existe une fonction qui représente la vraie relation entre et telle que :

A l’issue de notre travail de data scientist, nous aurons obtenu une estimation de la fonction . Dans ce cas l’erreur au carré moyenne (souvent notée MSE pour Mean Square Error en anglais) peut se décomposer en un terme de biais (ou écart moyen à la vrai fonction) et un terme de variance (dispersion de l’estimateur par rapport à sa moyenne). Cette décomposition s’écrit ainsi :

#### Pénalité du modèle Ridge

Le modèle Ridge est une version pénalisée du modèle linéaire multiple dont la fonction de coût est :

Cette pénalité entraîne que toute variation des paramètres peut avoir un impact bénéfique sur la qualité de l’estimation (au niveau du premier terme) mais participe à l’augmentation du second terme. Cela force le modèle à favoriser les paramètres associés aux variables explicatives qui contiennent réellement une information pertinente pour décrire la variable cible et à maintenir à des valeurs plus proches de zéros des paramètres associés aux variables explicatives peu pertinentes.

En terme d’arbitrage biais vs variance, le modèle Ridge se comporte de la manière suivante :

* correspond au modèle linéaire, qui est non biaisé
* Le biais augmente lorsque augmente
* La variance diminue lorsque augmente
* correspond au modèle où tous les paramètres sont à zéros, d’où et l’estimateur vaut ou (la moyenne des valeurs de Y) si on a inclus un intercept dans le modèle.

#### Cas particulier de l’intercept

L’intercept est le paramètre du modèle, il n’est associé à aucune variable explicative, il représente l’estimation du niveau moyen de lorsque les autres variables explicatives valent . C’est pourquoi on ne le pénalise pas en pratique. La fonction de coût du modèle Ridge avec intercept est donc :

## Modèle Lasso

### Problème de sélection de variables en grande dimension

Le modèle Ridge, on l’a vu, est bien adapté lorsqu’une partie des variables explicatives n’est pas très informative dans le modèle, car il contracte les coefficients associés à ces variables. Cependant, dans le cas où la vraie valeur de nombreux coefficients est , c’est à dire que les variables explicatives auxquelles ils sont associés n’ont aucune influence sur la variable cible, le modèle Ridge donnera des résultats d’estimation corrects, mais l’interprétation du modèle est rendue plus compliquée car on ne fera pas la différence entre un coefficient qui représente une influence réelle mais faible sur la variable cible et un coefficient faible mais qui devrait être nul. C’est à dire que le modèle Ridge ne sélectionne pas les variables pertinentes.

C’est pourquoi un modèle du nom de Lasso existe, qui sélectionne les variables pertinentes et fixe à zéro les coefficients des variables parasites.

### Le modèle Lasso

#### Intuition

L’intuition fondamentale pour utiliser un modèle Lasso est qu’un certain nombre de variables explicatives à notre disposition n’a aucune influence sur la variable cible, et donc que les coefficients associés auraient pour vraie valeur dans le modèle linéaire.

Cette intuition est caractérisée mathématiquement de la manière suivante : soit le nombre d’observations, le nombre de variables explicatives et le nombres de variables explicatives pertinentes. Alors l’intuition (qu’on appelle hypothèse de **sparsité**) s’écrit de la manière suivante.

On espère en sélectionnant les variables pertinentes de se ramener dans une situation où le modèle linéaire peut s’appliquer sans difficulté.

Idéalement on souhaite trouver tel que:

Cette equation signifie que est le vecteur qui minimise la valeur sous la contrainte que le nombre d'éléments non-nuls dans soit au maximum .

Avec est la “norme zéro” qui compte le nombre d’éléments non nuls dans .

Hors cette contrainte ne permet pas de faire de l’optimisation car elle ne définit pas un espace convexe. A la place, nous sommes dans l’obligation de choisir une contrainte moins forte qui permette d’obtenir une fonction de coût convexe qui reposera sur la “norme 1” qui est défini comme la somme des valeurs absolues des composants d’un vecteur.

#### Estimateur Lasso

Nous introduisons ici le *Least Absolute Shrinkage and Selection Operator* (LASSO), défini par la fonction de coût suivante :

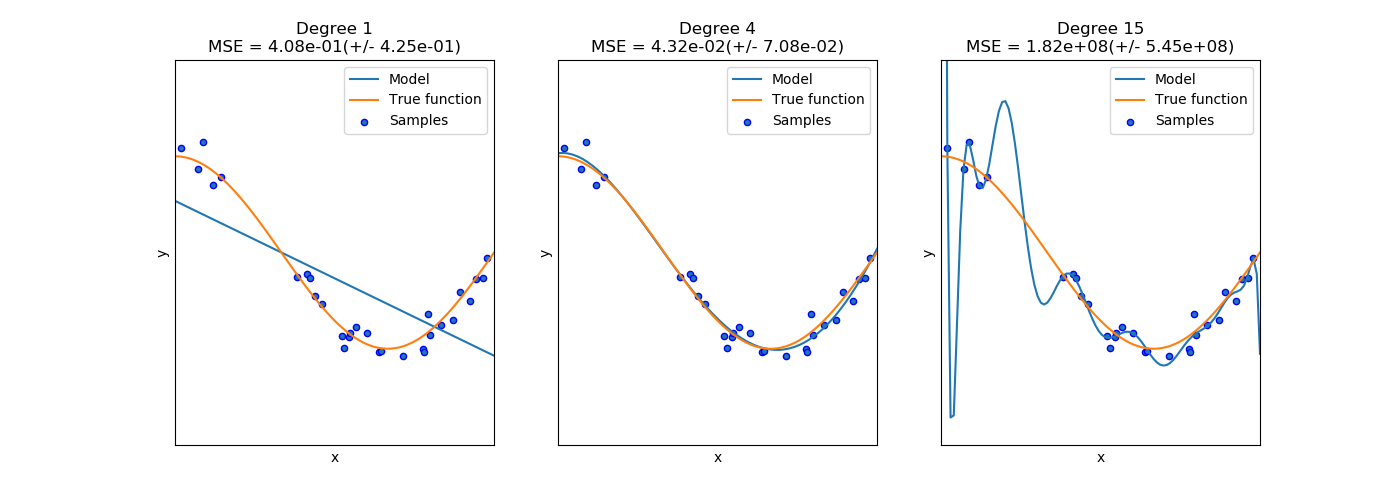
Les choses importante à connaître sur le modèle LASSO sont les suivantes :

* la constante de pénalisation doit être soigneusement choisie, en général les algorithme de résolution du LASSO procède à une cross-validation (dont la théorie est développée dans ce qui suit) et calcule l’estimateur pour de nombreuses valeur de afin d’identifier les valeurs les plus pertinentes.
* Plus est élevé, plus la solution sera sparse (c’est à dire contiendra peu d’élément non nuls), on augmente le biais et on diminue la variance.
* Plus sera petit, plus le nombre de coefficients augmente, ceci diminue le biais du modèle mais peut augmenter dramatiquement la variance (c’est ce qu’on appelle une situation de sur-apprentissage).
* En pratique vous découvrirez bientôt que le biais introduit par l’estimation LASSO peut être important. En fonction de vos contraintes de précision des résultats, il est possible d’utiliser LASSO pour sélectionner les meilleurs variables et ensuite estimer un modèle linéaire en conservant uniquement ces variables pour ôter le biais.

#### Underfitting et Overfitting (sous-apprentissage et sur-apprentissage)

On a mentionné au-dessus deux notions qui s’expliquent ainsi :

* Le sous-apprentissage est le fait qu’un modèle soit trop simple pour être une bonne estimation de la variable cible.
* Le sur-apprentissage est le phénomène inverse, lorsqu’on construit un modèle très complexe qui adhère parfaitement aux données d’apprentissage, mais inutile en pratique car il est très improbable qu’il se généralise bien à de nouvelles données inconnues.



La figure ci-dessus représente de gauche à droite, une situation de sous-apprentissage, une situation de bonne estimation, et une situation de sur-apprentissage.

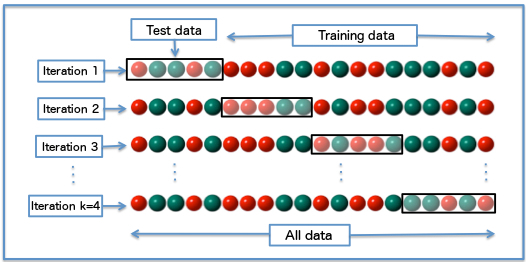
#### Comment se protéger du sur-apprentissage ?

Les situations de sous-apprentissage interviennent peu en pratique, ou elles sont souvent dues à un manque de données pertinentes ou d’autres problèmes qui ne peuvent pas être réglés directement par les data sciences. Le véritable ennemi des data scientists est le sur-apprentissage, car il donne l’illusion de la performance, mais est en réalité un piège!

Une manière simple et efficace de se garantir de ce piège est de pratiquer la cross-validation (ou k-fold cross validation). C’est un procédé qui consiste à choisir un entier k (souvent on choisit 10 par défaut), on répartit les observations au hasard dans k groupes de taille égale. Puis on répète k fois la méthode suivante :

* On isole un groupe i parmi 10 groupes, qu’on appellera **base de test**, et on rassemble les 9 autres, qu’on appellera base **d’apprentissage**.
* On estime le modèle choisi à l’aide de la base d’apprentissage.
* On calcule l’erreur commise par le modèle i sur la base de test (le groupe i) que l’on compare à l’erreur commise sur la base d’apprentissage après optimisation.

La comparaison de l’erreur d’apprentissage et l’erreur de test permet de comprendre le réel pouvoir explicatif d’un modèle, car elle quantifie la performance du modèle sur des données inconnues par rapport à sa performance sur des données connues.



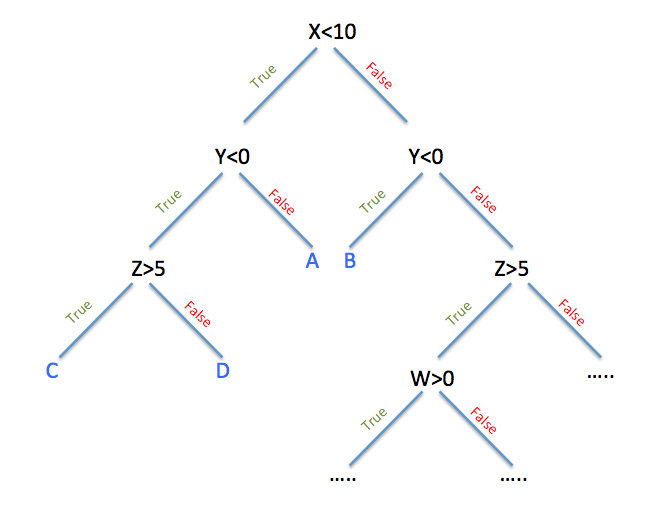
La figure ci dessus illustre le principe de la k-fold cross-validation. Chaque itération produit des résultat en termes d’erreurs de test et d’apprentissage dont on se sert pour évaluer le modèle. Pour calculer ces erreurs on se base en général sur la fonction de coût qu’on a choisit pour optimiser le modèle, ou bien tout simplement la moyenne des erreur au carré. En général on s’attend à ce que l’erreur de test et de validation soient du même ordre de grandeur, et on espère que l’erreur de manière générale sera petite par rapport aux valeurs prises par la variable cible.

# Random Forest (régression et classification)

Les random forest (en français, forêts d’arbres aléatoires) s’inscrivent dans le champ des méthodes de partitionnement récursif, qu’on connait plutôt sous l’acronyme CART (Classification And Regression Tree). On parle de classification lorsque la variable cible est qualitative (catégorielle), cas que nous traiterons dans un second temps. Ici nous nous intéressons au Regression Trees qui interviennent dans le cas d’une variable cible quantitative. Leur avantage réside dans leur représentation graphique aisément lisible. L’arbre est composé de trois types d’éléments :

* La racine, où réside l’ensemble des données d’apprentissage.
* Les noeuds/branches, qui représente les points à partir de la racine où les données sont séparées en deux groupes selon un critère lié aux variables explicatives.
* Les feuilles, qui sont les noeuds terminaux de l’arbre et auxquels sont associés un valeur dans le cas où est quantitative et une classe lorsque est qualitative.

Ainsi à partir de la racine on défini un noeud qui divise l’ensemble des données selon un critère lié à une variable explicative, pour chacune des deux branches ainsi créées on répète le même procédé, et ainsi de suite jusqu’à ce qu’aucune division ne satisfasse le critère de construction d’un noeud et on définit alors un noeud terminal ou feuille.



Voilà un exemple d’arbre aléatoire avec en haut la racine et la première division et l’enchaînement des branches, jusqu’aux feuilles. On constate d’ailleurs immédiatement l’aspect très clair et visuel des arbres aléatoires.

## Construction d’un arbre aléatoire

### Principe général

L’algorithme de construction de l’arbre aléatoire se structure de la façon suivante :

* Initialisation : On considère la racine comme l’ensemble des données de la base d’apprentissage.
* Pour chaque ensemble/sous-ensemble, on définit un noeud :
  + On sélectionne une variable explicative et on définit un critère de coupure, un seuil si est quantitative, ou un partage en groupes de modalités si est qualitative.
  + On marque le noeud comme terminal si aucun critère de division n’est pas satisfaisant, et on lui associe une valeur ou une classe selon la nature de .
* On interrompt la construction de l’arbre lorsque tous les noeuds sont terminaux.

Afin de mener à son terme cet algorithme, nous avons besoin de plusieurs choses :

1. Un critère de sélection de la meilleure division
2. Une règle pour définir si un noeud est terminal
3. Une méthode pour assigner à chaque feuille une valeur ou une classe

### Critère de division

#### Admissibilité

Un noeud est **admissible** si les branches qui en découlent portent des noeuds non vide (c’est à dire qu’au moins une observation appartient à chacun des noeuds enfant). Pour un noeud parent contenant observations il existe divisions admissibles si la variables sélectionnée pour la division est quantitative ou qualitative ordinale, et divisions admissibles si la variable selectionnée est nominale.

Afin de sélectionner la meilleure division admissible, on construit une fonction **d’hétérogénéité** qui présente deux propriétés remarquables :

* Elle vaut zéro lorsque tous les individus appartiennent à la même modalité ou présentent la même valeur de .
* Elle est maximale lorsque les valeurs de sont équiréparties dans le noeud.

On cherche donc la division qui minimise la somme des hétérogénéités des noeuds enfants.

#### Critère d’arrêt

Un noeud donné sera terminal lorsqu’il est homogène, c’est à dire que toutes les observations dans le noeud présente la même valeur ou la même modalité de , lorsqu’il n’existe plus de divisions admissibles, ou bien lorsque le nombre d’observations dans le noeud est inférieur à une valeur définie à l’avance, en général de l’ordre de quelques unités.

#### Y quantitative

Dans le cas de la régression, l’hétérogénéité du noeud s’écrit de la manière suivante :

parfois aussi noté ou encore est le nombre d’éléments dans le noeud , et est la moyenne des valeurs de parmi les observations du noeud . Ce qui est fait la variance du noeud . La division retenue est celle pour laquelle : la somme de l’hétérogénéité de la branche gauche et de la branche droite.

#### Y qualitative

Soit une variable qualitative à modalités ou catégories numérotées de 1 à m. La fonction d’hétérogénéité privilégiée la plupart du temps est la concentration de GINI:

Où est la proportion de la classe i de Y dans le noeud k.

### Elagage de l’arbre

On a vu précédemment avec les modèles LASSO et RIDGE qu’un risque important dans tout problème d’apprentissage supervisé est celui du sur-apprentissage. Le critère d’arrêt défini pour la construction de l’arbre est souvent propice au sur-apprentissage puisqu’il est très probables que la plupart des feuilles de l’arbre ne contiennent que quelques observations. Ainsi l’arbre de décision tel quel sera très instable, son biais est quasi-nul voir nul par définition, il dépend très fortement des observations de la base d’apprentissage et sera potentiellement peu généralisable aux données de la base de test.

Le problème est donc de trouver un arbre intermédiaire qui vérifie un compromis biais variance intéressant pour les besoins de l’estimation de . Le nombre de sous-arbres malheureusement est souvent très élevé c’est pourquoi on suit en général la méthode de Breiman, qui consiste à construire une suite emboîtée de sous-arbres (chaque arbre de la suite est un un sous-arbre de l’arbre précédent). Et de choisir parmi cette suite, l’arbre optimal selon un critère de généralisation.

#### Construction de la suite emboîtée d’arbres

Soit un arbre, le nombres de ses feuilles ou noeuds terminaux, qu’on appellera la complexité de l’arbre .

La **qualité d’ajustement**  de l’arbre est mesurée par :

Où est l’hétérogénéité du noeud .

La construction repose sur une pénalisation de l’arbre indexée sur sa complexité :

Pour , minimise . En faisant croître la division de qui améliore le moins devra être supprimée afin d’optimiser , car . Ce procédé permet bien de créer une suite emboîtée d’arbres puisque pour construire l’arbre suivant on lui retire un noeud.

Une fois la suite construite, on peut sélectionner celui qui minimise l’erreur de prédiction sur la **base de validation** (base de test).

### Remarques générales

* L’algorithme a tendance à favoriser la sélection de variables explicatives avec beaucoup de modalités, il convient donc de transformer les variables explicatives qualitatives en fusionnant les modalités par groupes cohérents pour éviter le sur-apprentissage.
* Les arbres de décision ne requièrent pas d’hypothèses particulières sur les distributions des variables et sont bien adaptés aux situations où les variables explicatives sont nombreuses puisque la sélection des variables fait partie de l’algorithme d’optimisation.
* La recherche d’une division dépendant uniquement de la position relative des valeurs des variables explicatives quantitatives, l’algorithme résiste donc aux valeurs atypiques et au distributions de valeurs asymétriques.
* La structure hiérarchique de l’arbre (les divisions se font une par une) favorise la propagation de l’erreur engendrée par une division à tous les noeuds enfants. Les arbres de décisions peuvent donc passer à côté d’un optimum global et donc de la vraie fonction de classification ou de régression qui lie les variables explicatives à la variable cible.
* Dans le cas de la régression, le résultat de l’arbre est une fonction étagée, toutes les observations d’une même feuille prendront la même valeur estimée de . Si la vraie fonction présente des propriété de régularité (par exemple un polynôme ou une droite affine), ces propriétés ne seront pas conservées par le modèle d’arbre aléatoire.

## Random Forest (Forêt d’arbres aléatoire) plutôt à inclure dans le chapitre sur le boosting?

Comme son nom l’indique, la random forest n’est rien de plus qu’un ensemble d’arbres aléatoire qu’on va faire coopérer afin d’obtenir de meilleurs résultats de régression ou de classification.

### Principe du Bagging

Le principe du Bagging est très simple. Soit la variable cible, liée aux variables explicatives par une fonction telle que et le nombre d’observations. En tirant échantillons indépendants à partir de l’ensemble des observations, la prévision agrégée donnée par les modèles qui découlent des échantillons s’écrit :

* quantitative : le modèle agrégé est la moyenne des fonctions estimées par les modèles, la moyenne des valeurs de pour une observation donnée..
* qualitative : le modèle agrégé est le vote majoritaire parmi les fonctions estimées par les modèles, la modalité de la plus représentée parmi les réponses des différents modèles à une observation donnée.

La principale difficulté du bagging réside dans le fait de construire échantillons indépendants, en effet, à moins de disposer d’une base de données contenant un très grand nombre d’observations, il est difficile de respecter cette contrainte dans la plupart des cas.

### Bootstrap

Le Bootstrapping est un procédé qui permet d’augmenter artificiellement le nombre d’observation d’un échantillon de données sans pour autant modifier la distribution des variables présentes dans le jeu de données. Le principe est simple, on dispose d’un jeu de données contenant observations, pour créer un échantillon de taille on tire avec remise observations parmi le jeu de données original, et chaque observation du jeu de données original a chance d’être tiré (c’est un tirage avec remise équiprobable). L’équiprobabilité du tirage est essentielle afin que la loi de distribution de l’échantillon soit la même que celle de la base initiale.

### Random Forest

La première idée derrière les random forest est d’effectuer un bagging de plusieurs arbres aléatoires. Plusieurs élagages des arbres ainsi construits sont possible :

* On peut conserver les arbres complets et éventuellement limiter le nombre minimum d’observations au niveau des noeuds terminaux.
* Conserver au plus feuilles ou limiter la profondeur de l’arbre à niveaux de noeuds.
* Adopter la méthode vue plus haut pour un arbre seul, c’est à dire construire l’arbre complet puis élaguer par validation croisée.

En général on retiendra la première stratégie, car elle représente un bon compromis entre qualité d’estimation et quantité de calculs. Chaque arbre ainsi construit aura un biais très faible et une grande variance, cependant le fait d’agréger les modèles entre eux participe justement à réduire cette variance. Cet algorithme est très simple à mettre en place, ce qui est un grand avantage, cependant le nombre de modèles à calculer avant que l’erreur de test (appelée aussi erreur de validation) se stabilise peut être très important. Le modèle final sera volumineux en terme d’espace disque car il est nécessaire de stocker la structure complète de tous les arbres pour pouvoir faire des prévisions. Enfin la multiplication du nombre d’arbres participants au modèle rend plus difficile, voir impossible l’interprétation du modèle comme cela était possible avec un seul arbre.

La seconde idée consiste à améliorer la méthode du bagging afin de créer des random forest s’appuyant sur des échantillons de données les plus “indépendants” possible. Non seulement le hasard intervient lors de la sélection des observations lors de la construction des échantillons d’apprentissage, mais on fera également intervenir le hasard dans le choix des variables explicatives retenues pour chaque échantillon sur lequel on construira un arbre aléatoire.

Ce double hasard de sélection des observations et des variables explicatives a plusieurs avantages : il permet de s’approcher de l’hypothèse d’indépendance des échantillons, il réduit le nombre de calculs à effectuer pour le construction de chaque arbre et il réduit les risques d’erreurs liés à d’éventuelles corrélations entre variables explicatives.

Dernière remarque, soit la variable cible et les variables explicatives à notre disposition, en général le nombre de variables que l’on conservera par arbre pour une classification est et pour une régression.

### Interprétation

Le fait d’agréger les modèles pour produire la prédiction finale rend difficile l’interprétation directe du modèle, cependant deux méthodes sont utilisées en pratique afin d’évaluer l’importance de chacune des variables pour la prédiction.

#### Mean decrease Accuracy

Cette méthode consiste à permuter de manière aléatoire les valeurs d’une variables explicatives, on mesure ensuite la différence entre l’erreur de validation pré et post-permutation, plus elle est élevée, plus on considère la variable en question importante pour la prévision de la variable cible.

#### Mean decrease Gini

Cette méthode permet d’évaluer l’importance d’une variable au niveau d’un noeud, elle mesure la décroissance de la fonction d’hétérogénéité si on utilise la variables explicative utilisée pour le noeud par celle que l’on souhaite évaluer. L’importance générale de la variable est alors une sommes des décroissance d’hétérogénéité mesurées, pondérée par le nombres d’observation à chaque noeud.

# Révision de régression logistique

## Définition

A la différence des régressions linéaires qui vous prédisent un nombre, les modèles de classifications vous prédisent une catégorie. Par exemple, si vous essayez de prédire si quelqu’un va vous acheter un produit en fonction de certaines variables indépendantes, vous entrez dans une problématique de classification car les catégories que vous essayez de prédire sont “oui, la personne va acheter le produit” ou “non, la personne ne va pas acheter mon produit”.

Les régressions logistiques sont une catégorie dans les modèles de classification mais vous en avez beaucoup d’autres comme les arbres de décisions (*decision tree*), SVM (*support vector machine*) ou Naive Bayes

## Equation

Lorsqu’on construit un modèle de régression logistique, on suppose qu’il existe une fonction qui lie la variables cible aux variables explicatives représentées dans la matrice de la manière suivante :

Où est l’erreur.

## Régression logistique

### 

### 

Lorsque nous avons les régressions linéaires, nous avons vu que notre prédicteur était la ligne que traçait notre modèle. Dans une régression logistique, la ligne est simplement une frontière qui sépare deux catégories. Dans le graphique du dessus, nous essayons de voir si une personne va acheter un produit (représenté par le chiffre 1) ou ne va pas acheter un produit (représenté par le chiffre 0) la ligne représente la probabilité qu’une personne achète (*purchased)* ou non en fonction de son Âge.

L’allure de la courbe est la représentation pour une variable explicative (ici l’âge) de l’équation introduite plus haut.

Dans cet exemple, nous n’avons qu’une variable indépendante et une constante. L’équation ressemble beaucoup à une régression linéaire comme vous pouvez le constater, seulement ici une fonction logistique est appliquée à la variable explicative utilisée pour la régression. Cette fonction contraint les valeurs de à rester dans l’intervalle qui est l’ensemble des valeurs que peut prendre une probabilité. En fonction de la probabilité obtenue, l’algorithme va savoir dans quelle catégorie placer notre individu.

### Comment faire notre classification

Maintenant que nous avons dessiné la ligne, nous pouvons commencer nos interprétations. Puisque notre modèle est cette fois probabiliste, les points qui auront une probabilité supérieure à 50% appartiendront à la catégorie A tandis que les points qui auront une probabilité inférieure à 50% appartiendront à une catégorie B. Selon le problème considéré, un autre seuil pourra être choisi, par exemple dans des problématiques de fraude bancaires, on aura tendance à classer dans la catégorie des fraudeurs des individus qui ont des probabilités de fraude inférieures à 50% car on souhaite sécuriser au maximum le système bancaire de menaces frauduleuses.

Prenons un exemple, en fonction de certaines variables indépendantes, nous avons découvert qu’une personne A a 60% de chance d’acheter le produit. Elle sera donc considérée comme un “acheteur” pour notre modèle. De l’autre côté, si nous avons cette fois une personne B qui a seulement 45% de chances d’acheter le produit, elle sera considérée comme “non-acheteur”.

## Faux positifs - Faux Négatifs

Puisque notre modèle est fondé sur des probabilités, il peut arriver qu’il ait tort parfois. Les faux positifs et les faux négatifs représentent les erreurs commise par notre modèle de classification.

### Faux positif

Continuons sur l’exemple du dessus. Si notre modèle catégorise notre personne A comme “acheteur” et que cette personne dans la réalité n’achète pas le produit alors nous avons affaire à un faux positif. Le modèle s’attendait à un résultat positif qui n’est finalement pas arrivé.

### Faux négatif

A l’inverse, si la personne B, que le modèle avait prédit comme non-acheteuse, achète finalement le produit, c’est un faux négatif. Nous avons prédit un résultat négatif mais il n’est pas arrivé.

### Soyez attentifs aux faux positif et SURTOUT faux négatifs

Soyez vigilant des faux positifs et négatifs car une erreur de prédiction peut avoir des conséquences plus ou moins graves en fonction de ce que vous essayez de prédire. Par exemple, si vous êtes scientifiques et que vous essayez de prédire un tremblement de terre et que vous tombez sur une faux négatif (c’est à dire que vous aviez prédit que le tremblement n’arriverait pas alors qu’il est arrivé), personne n’était de fait préparé à l’événement.

De manière général, les faux négatifs sont pires que les faux positifs car dans le premier cas, personne n’est préparé à ce que l’événement se passe. Dans le second, vous y êtes préparé et même s’il n’arrive pas, ce n’est pas le plus grave.

## Comment évaluer votre modèle

La régression logistique est une transposition dans l’intervalle de la régression linéaire multiple, les méthodes de sélection et d’évaluation utilisées pour les modèles de régressions linéaires peuvent donc être utilisés pour cette dernière. Cependant, nous aborderons ici d’autres méthodes spécifiquement adaptées aux problèmes de classification binaires.

### Matrice de confusion

## 

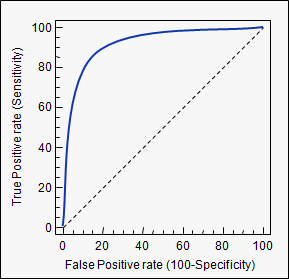
Une des façons rapides et faciles de mesurer la performance de votre modèle grâce aux matrices de confusion. L’idée est de voir les prédictions que votre modèle a vu juste ainsi que les faux-positifs et faux-négatifs. En faisant les sommes des erreurs sur le total de prédiction vous avez le taux de précision de votre modèle.

Une mesure simple et pertinente de la performance de votre modèle serait de comparer le taux de précision du modèle avec la proportion de positifs dans la base. En effet, le modèle de plus simple dans le cas d’un problème de classification et de classer tous les individus dans la même classe, dans le cas de ce modèle trivial, le taux de précision sera égal à la proportion occupée par le groupe majoritaire dans les données. Mettons que nous avons à notre disposition une base de données qui donne les résultats du baccalauréat pour un échantillon de population. Si l’échantillon contient 70% d’individus qui ont eu leur bac, alors si notre modèle prédit que tout le monde aura son bac, il a raison 70% du temps. De fait, on a intérêt à construire une modèle plus complexe uniquement si sa précision peut être plus élevée que 70%.

### Courbe ROC, AUC et indice de GINI

La courbe ROC (receiver operating characteristic curve) qui permet de visualiser les performances d’un modèle de classification binaire en fonction de son critère de discrimination (le seuil de probabilité à partir duquel le modèle estime qu’une observation est classée comme “positif”).

Cette courbe est obtenue en traçant le taux de vrais positifs (sensitivity) détectés en fonction du taux de faux positifs ( or ) pour différentes valeurs du seuil.



Une courbe ROC revêt en général un aspect similaire à l’illustration ci-dessus. Il est très rare et très mauvais signe que la courbe ROC se trouve sous la diagonale, cela signifierait que pour chaque vrai positif détecté, on récupère une quantité plus grande en proportion de faux positifs. La courbe ROC permet immédiatement de décrire les performances du modèle en termes de détection des observations positives, cependant elle permet aussi d’avoir une appréciation générale du modèle. Le biais par lequel la courbe évalue la performance générale du modèle est un indicateur numérique appelé AUC (Area Under the Curve). L’AUC est littéralement le calcul de l’aire délimitée par la courbe ROC et les côtés du carré unité.

L’AUC s'interprète comme la probabilité que le modèle donne un score plus élevé à une observation positive choisie aléatoirement qu’à une observation négative choisie aléatoirement. L’AUC est également lié à l’indice de GINI qui décrit la dispersion statistique de la population et qui est très utilisé en économie pour quantifier les inégalités.

L’AUC varie entre 0 et 1 en théorie, mais les modèles dont l’AUC est inférieure à 0.5 (50%) sont à exclure immédiatement car cela signifie que le modèle est moins bien performant que le hasard total.

# Naive Bayes

En statistiques, les classification naïves Bayésienne appartiennent à une famille de classifications probabilistes reposant sur le théorème de Bayes.

## Théorème de Bayes

Le théorème de Bayes correspond à l’affirmation suivante :

Soit et deux variables aléatoires, alors l’égalité suivante est vérifiée :

La probabilité conditionnelle de sachant est égale au produit de la probabilité conditionnelle de sachant et la probabilité de divisé par la probabilité de .

## Naive Bayes

On considère la situation où on dispose de la variable cible qualitative (c’est donc un problème de classification) que l’on cherche à prédire, et une collection de variables explicatives . Le problème revient à estimer pour chaque observation la loi , qui donne la probabilité pour de prendre chacune de ses valeurs possibles sachant les valeurs de pour cette observation.

Le théorème de Bayes intervient ici et nous donne l’écriture suivante :

Le dénominateur ne fait pas intervenir et n’a donc aucune influence sur les résultats du modèle, on s’intéressera uniquement au numérateur qu’on peut décomposer récursivement grâce aux propriétés des probabilités conditionnelles.

…

C’est ici qu’on a besoin de l’hypothèse fondamentale et naïve qui nous permet de construire nos estimations et qui dit que toutes les variables explicatives doivent être indépendantes, car ainsi on a quelque soit entre et :

De fait on obtient :

Qui est très simple à calculer puisqu’il suffit d’estimer pour chaque valeur de la loi de distribution des .

### Cas où est qualitative

Dans le cas où est une variable explicative qualitative qui prend les modalités alors on peut écrire :

On estime la probabilité que prenne la modalité sachant que comme la proportion d’observations où parmi toutes les observations où .

### Cas où est quantitative

Pour les cas où est quantitative en général on se ramène au cas qualitatif en découpant l’intervalle des valeurs des valeurs de en morceaux indexés par et délimité par les valeurs et la loi de probabilité devient :

C’est à dire la proportion d’observations pour lesquelles la valeurs de appartient à l’intervalle parmi toutes les observations pour lesquelles .

Une autre manière d’estimer la loi de sachant est de faire l’hypothèse que suit une loi normale dont on estime les paramètres et grâce aux données disponibles sur . Sous l’hypothèse de normalité, sui une loi normale de paramètres :

Qui est la valeur moyenne de parmi les individus pour qui . De même on calcule la variance de la loi normale :

L’estimateur de la variance de parmi les individus pour qui .

Une fois cette estimation effectuée, on obtient:

Une fois toutes les probabilités conditionnelles calculées on obtient pour chaque observation et pour chaque modalité de une probabilité qui détermine notre classification. Chaque observation sera classée dans la modalité de la plus probable en fonction des valeurs des variables explicatives .

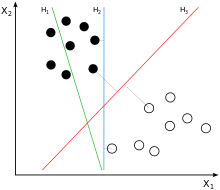
## Remarques générales

Un avantage du modèle bayésien naif est qu’il permet de ne pas faire d’hypothèse sur les lois de distribution des variables explicatives si on transforme ces dernières en variables quantitatives. Cependant il est très rare que l’hypothèse fondamentale d’indépendance des variables explicatives soit vérifiée en pratique.

Les modèles bayésiens naifs peuvent faire l’objet d’une agrégation au même titre que les arbres aléatoires, cela permet en général d’obtenir des résultats bien plus stables et également de mieux respecter l’hypothèse d’indépendance des variables explicatives si, comme on l’a vu dans le cas des random forest, on utilise qu’une partie des variables explicatives pour construire chaque modèle.

# Support Vector Machine (SVM)

Le modèle Support Vector Machine, ou en français séparateur à vastes marges, est un modèle communément utilisé pour la classification en deux classes. Il permet donc d’estimer une variable cible possédant deux modalités. Ce modèle cherche à découvrir un hyperplan qui sépare le nuage de point qui représente les observations, séparant ainsi les deux classes d’individus tout en restant le plus éloigné possible de toutes les observations.



Visuellement on a ici en vert un modèle qui ne sépare pas les observations en deux groupes homogènes selon la variable cible (blanc ou noir), en bleu un hyperplan qui sépare bien les observations mais n’est pas le plus éloigné possible de certains individus, et en rouge l’hyperplan issu du modèle de support vector machine qui sépare bien nos individus en deux classes bien distinctes.

A réfléchir mais la théorie derrière les svm est assez complexe pour des gens qui n’ont pas un niveau au moins L3 en mathématiques du coup je pense concentrer le cours sur de la pratique.

Tuning Hyperparameters  
Kernel: The main function of the kernel is to transform the given dataset input data into the required form. There are various types of functions such as linear, polynomial, and radial basis function (RBF). Polynomial and RBF are useful for non-linear hyperplane. Polynomial and RBF kernels compute the separation line in the higher dimension. In some of the applications, it is suggested to use a more complex kernel to separate the classes that are curved or nonlinear. This transformation can lead to more accurate classifiers.  
Regularization: Regularization parameter in python's Scikit-learn C parameter used to maintain regularization. Here C is the penalty parameter, which represents misclassification or error term. The misclassification or error term tells the SVM optimization how much error is bearable. This is how you can control the trade-off between decision boundary and misclassification term. A smaller value of C creates a small-margin hyperplane and a larger value of C creates a larger-margin hyperplane.  
Gamma: A lower value of Gamma will loosely fit the training dataset, whereas a higher value of gamma will exactly fit the training dataset, which causes over-fitting. In other words, you can say a low value of gamma considers only nearby points in calculating the separation line, while the a value of gamma considers all the data points in the calculation of the separation line.  
  
Advantages  
SVM Classifiers offer good accuracy and perform faster prediction compared to Naïve Bayes algorithm. They also use less memory because they use a subset of training points in the decision phase. SVM works well with a clear margin of separation and with high dimensional space.  
  
Disadvantages  
SVM is not suitable for large datasets because of its high training time and it also takes more time in training compared to Naïve Bayes. It works poorly with overlapping classes and is also sensitive to the type of kernel used.