

COURS 1

1. STATISTIQUES INFÉRENTIELLES vs APPRENTISSAGE MACHINE

2. APPRENTISSAGE SUPERVISÉ vs APPRENTISSAGE NON SUPERVISÉ

1. STATISTIQUES INFÉRENTIELLES vs APPRENTISSAGE MACHINE

Statistiques inférentielles classiques		Apprentissage machine
1.1.	Modèles essentiellement interprétables	vs
1.2.	Méthode essentiellement déductive	vs
1.3.	Minimisation du biais	vs
		Compromis biais-variance

1.1. MODÈLES INTERPRÉTABLES vs MODÈLES PRÉDICTIFS

Statistiques inférentielles classiques		Apprentissage machine
1.1.	Modèles essentiellement interprétables	vs
1.2.	Méthode essentiellement déductive	vs
1.3.	Minimiser le biais	vs
		Compromis biais-variance

Historiquement:

- Les statistiques inférentielles et l'apprentissage machine ont été développés avec des objectifs différents.

Statistiques inférentielles: comprendre et expliquer un phénomène.

➤ On vise une avancée théorique.

Apprentissage machine: prédire les valeurs de nouvelles observations.

➤ On vise une application pratique.

1.1.1. STATISTIQUES INFÉRENTIELLES : MODÈLES ESSENTIELLEMENT INTERPRÉTABLES

1.1.1. STATISTIQUES INFÉRENTIELLES : MODÈLES ESSENTIELLEMENT INTERPRÉTABLES

En statistiques inférentielles classiques...

- I. On pose une question de recherche visant à comprendre un phénomène.
- II. On pose une hypothèse nulle (H_0) : une description de l'univers dans laquelle existe l'analyse.
 - On veut rejeter H_0 .
 - On établit les valeurs critiques qui permettraient de rejeter H_0 avec une faible probabilité de se tromper si H_0 est vraie.
 - Cette « faible probabilité » correspond à l'erreur de type 1.
- III. On pose une hypothèse alternative (H_A) : l'hypothèse du chercheur.
 - Le chercheur construit un modèle du phénomène qu'il tente de comprendre.
 - Le modèle inclut généralement une ou plusieurs variables indépendantes (i.e. unidimensionnel ou de faible dimensionnalité).
 - Chaque variable est généralement accompagnée d'un **paramètre**, qui reflète l'importance de la variable à l'intérieur du modèle.
 - Le chercheur s'inspire principalement de la documentation théorique pour construire son modèle.
 - Ce modèle est **rigide**.
- IV. On récolte un échantillon : un **groupe** d'observations.
 - Le chercheur vérifie les postulats permettant de valider la loi générale.
 - Le chercheur utilise ce **groupe** d'observations pour estimer les valeurs des paramètres du modèle du chercheur (i.e. l'importance des différentes variables du modèle du chercheur).
- V. On conclut.
 - Le chercheur vérifie quelle était la probabilité d'obtenir les valeurs des paramètres estimés à partir de l'échantillon si H_0 est vraie (i.e. la valeur p).
 - Si cette probabilité est plus faible que la probabilité d'erreur de type 1 maximale établie au début, on rejette H_0 .

1.1.1. STATISTIQUES INFÉRENTIELLES : MODÈLES ESSENTIELLEMENT INTERPRÉTABLES

En statistiques inférentielles classiques...

- On cherche généralement à **COMPRENDRE** un phénomène.
- Le nombre de variables indépendantes impliquées dans le modèle du chercheur est généralement peu élevé.
- On essaie d'avoir le plus grand nombre de participants possible.
 - Le nombre de variables incluses dans le modèle est généralement beaucoup plus petit que le nombre de sujets.
 - Cette structure de données est généralement nommée « données longues » (*long data*).
- La conclusion obtenue est essentiellement binaire: on rejette ou on ne rejette pas H_0 .
- La conclusion est limitée à des groupes d'individus.
 - On ne peut pas utiliser les résultats de l'analyse pour prédire de nouvelles données individuelles.

1.1.2. APPRENTISSAGE MACHINE : MODÈLES ESSENTIELLEMENT PRÉDICTIFS

1.1.2. APPRENTISSAGE MACHINE : MODÈLES ESSENTIELLEMENT PRÉDICTIFS

En apprentissage machine, il n'y a ni H_0 , ni H_A .

- I. On cible un problème à résoudre.
- II. On pose une hypothèse quant à la forme d'un modèle capable de transformer des variables d'entrée en variables de sortie appropriées.
 - Le modèle inclut généralement un grand nombre de variables.
 - Chaque variable est généralement accompagnée d'un **paramètre**, qui reflète l'importance de la variable à l'intérieur du modèle.
 - Le chercheur s'inspire principalement de la documentation pratique pour sélectionner un algorithme, **puis le modèle est induit des données**.
 - La forme du modèle est donc **flexible** et s'adapte à partir des exemples.
- III. On récolte un échantillon : un grand nombre d'observations.
 - On divise ces observations en au moins deux sous-ensembles: un ensemble d'entraînement et un ensemble de test.
- IV. On entraîne le modèle à l'aide des exemples de l'ensemble d'entraînement.
 - On utilise l'ensemble d'entraînement pour estimer les valeurs des paramètres (i.e. l'importance des différentes variables prédictives).
- V. On évalue le modèle à l'aide des exemples de l'ensemble de test.
 - On vérifie la capacité du modèle à prédire des nouveaux exemples qui n'ont jamais été utilisés pour l'entraîner.
 - On vérifie la **capacité du modèle à généraliser**.

1.1.2. APPRENTISSAGE MACHINE : MODÈLES ESSENTIELLEMENT PRÉDICTIFS

En apprentissage machine...

- On induit une loi générale à partir des données.
- On cherche généralement à prédire de nouvelles observations.
- Le nombre de variables impliquées dans le modèle du chercheur est souvent élevé.
- On essaie d'avoir le plus grand nombre d'exemples possible.
 - Néanmoins, le nombre de variables incluses dans le modèle est souvent plus grand que le nombre d'exemples.
 - Cette structure de données est généralement nommée « données larges » (*wide data*).
- La conclusion obtenue est essentiellement continue:
 - À quel point peut-on bien prédire de nouvelles observations.
- La conclusion n'est pas limitée à des groupes d'individus.
 - On peut utiliser le modèle estimé pour prédire de nouvelles données **individuelles** (c'est généralement l'objectif principal).

1.2. MÉTHODE DÉDUCTIVE vs MÉTHODE INDUCTIVE

Statistiques inférentielles classiques		Apprentissage machine	
1.1.	Modèles essentiellement interprétables	vs	Modèles essentiellement prédictifs
1.2.	Méthode essentiellement déductive	vs	Méthode essentiellement inductive
1.3.	Minimiser le biais	vs	Compromis biais-variance

1.2.1. STATISTIQUES INFÉRENTIELLES : MÉTHODE ESSENTIELLEMENT DÉDUCTIVE

1.2.1. STATISTIQUES INFÉRENTIELLES : MÉTHODE ESSENTIELLEMENT DÉDUCTIVE

En statistiques inférentielles, **toute l'analyse est conduite dans un univers gouverné par une certaine loi générale**.

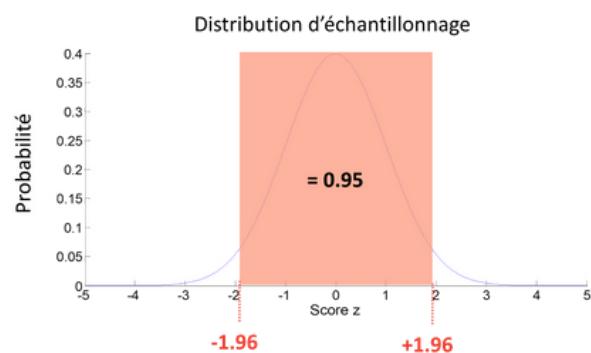
Cette loi générale correspond à une distribution d'échantillonnage.

Cette distribution d'échantillonnage correspond à la probabilité d'obtenir chaque échantillon possible, lorsque ces échantillons sont tirés d'une certaine population.

Cette population correspond à l'hypothèse nulle (H_0).

On illustre à droite un exemple de loi générale...
...qui correspond à une distribution d'échantillonnage...
...laquelle correspond à une certaine hypothèse nulle.

Une fois que cette loi générale a été établie,
la méthode des statistiques inférentielles procède
à travers un raisonnement **déductif**.



1.2.1. STATISTIQUES INFÉRENTIELLES : MÉTHODE ESSENTIELLEMENT DÉDUCTIVE

Pour pouvoir utiliser un processus déductif, on doit d'abord connaître une loi générale.

Exemple de loi générale :

- SI un être est humain,
➤ ALORS il est mortel.

Etant donnée cette loi générale, 4 cas peuvent survenir:

1. Socrate est un être humain.
➤ On peut conclure que Socrate est mortel.
2. Socrate n'est pas un être humain.
➤ On ne peut rien conclure.
3. Socrate est mortel.
➤ On ne peut rien conclure.
4. Socrate n'est pas mortel.
➤ On peut conclure que Socrate n'est pas un être humain.



1.2.1. STATISTIQUES INFÉRENTIELLES : MÉTHODE ESSENTIELLEMENT DÉDUCTIVE

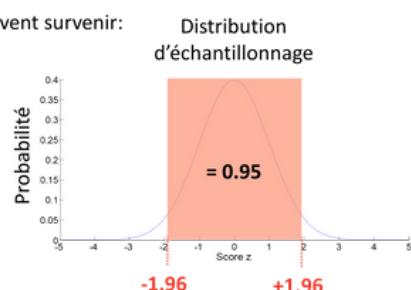
Dans le cadre d'un test d'hypothèse, la loi générale correspond à la distribution d'échantillonnage.

Prenons l'exemple d'une loi générale associée à la distribution d'échantillonnage des scores z :

- SI un échantillon est tiré de la population Y,
➤ ALORS dans 95 % des cas, son score z sur la distribution d'échantillonnage de la population Y se situera entre -1.96 et +1.96.

Etant donnée cette loi générale, si vous tirez un échantillon, 4 cas peuvent survenir:

1. L'échantillon est bel et bien tiré de la population Y.
➤ On ne peut jamais affirmer ceci avec certitude.
2. L'échantillon N'est PAS tiré de la population Y.
➤ On ne peut jamais affirmer ceci avec certitude (et on ne pourrait rien conclure).
3. Le score z de l'échantillon est situé entre -1.96 et +1.96.
➤ On ne peut rien conclure.
4. Le score z de l'échantillon N'est PAS entre -1.96 et +1.96.
➤ On peut conclure que l'échantillon n'est pas tiré de la population Y!



- ❖ Notons toutefois que la loi générale n'indique pas que TOUS les échantillons devraient se retrouver entre -1.96 et +1.96.
❖ SEULEMENT 95 % devraient se retrouver entre -1.96 et +1.96.
- ❖ Ainsi, en rejetant la loi générale (et donc H_0), on sait que si H_0 est vraie, on a une probabilité de 5 % de se tromper.
❖ C'est la probabilité d'erreur de type 1.

1.2.1. STATISTIQUES INFÉRENTIELLES : MÉTHODE ESSENTIELLEMENT DÉDUCTIVE

Ainsi, la méthode des statistiques inférentielles repose sur une loi générale.

Or, on ne peut **JAMAIS** vérifier directement que cette loi est correcte.

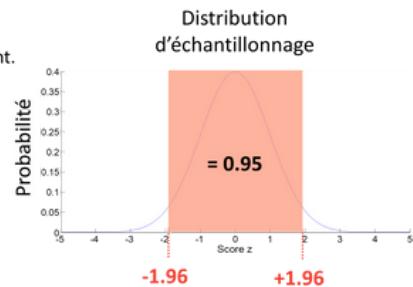
- On ne peut **JAMAIS** vérifier la distribution d'échantillonnage directement.
- On ne peut donc jamais vérifier les valeurs critiques directement.

On peut néanmoins **POSTULER** que la loi générale est respectée...

- ...**SI ET SEULEMENT SI**... certains postulats sont respectés
- (ex. normalité, homoscédasticité, additivité, linéarité, colinéarité, etc.)

Or, les méthodes de vérification de ces postulats sont approximatives.

- Ceci fragilise la validité de toute la méthode.
- Ceci contribue au problème de reproductibilité des résultats qui frappe actuellement plusieurs domaines de recherche (incluant les sciences biomédicales et humaines)!



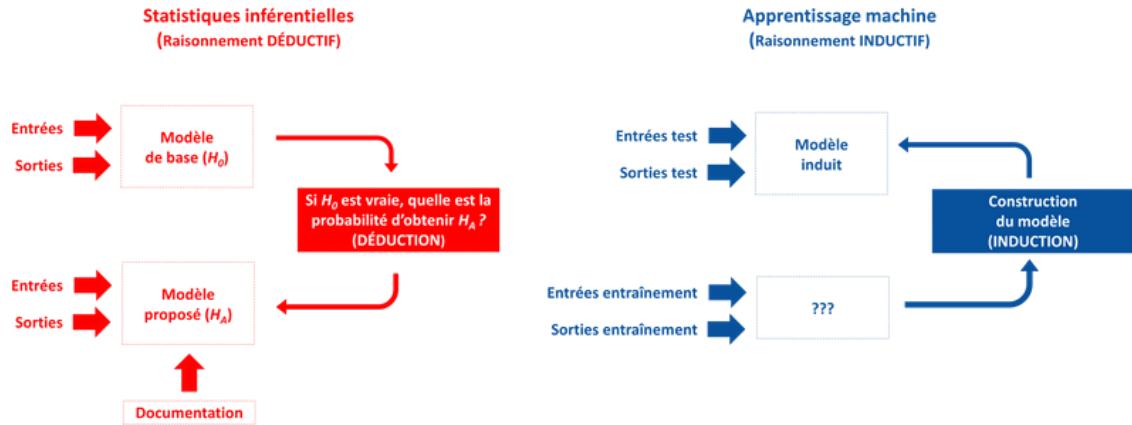
1.2.2. APPRENTISSAGE MACHINE : MÉTHODE ESSENTIELLEMENT INDUCTIVE

1.2.2. APPRENTISSAGE MACHINE : MÉTHODE ESSENTIELLEMENT INDUCTIVE

En apprentissage machine...

- On ne postule, a priori, aucune loi générale décrivant l'univers dans lequel l'analyse est réalisée.
 - Aucun postulat n'est donc absolument nécessaire afin d'assurer la validité de la méthode.
- **On se base essentiellement sur les données pour construire une loi générale.**
 - On procède donc selon un mode de raisonnement **inductif**.
- **Ex. de raisonnement inductif:**
 - Socrate, Platon et Aristote sont des êtres humains.
 - Socrate est mortel, Platon est mortel, Aristote est mortel.
 - Donc, tous les êtres humains sont mortels.





1.3. MINIMISER LE BIAIS vs COMPROMIS BIAIS-VARIANCE

	Statistiques inférentielles classiques	Apprentissage machine
1.1.	Modèles essentiellement interprétables	vs Modèles essentiellement prédictifs
1.2.	Méthode essentiellement déductive	vs Méthode essentiellement inductive
1.3.	Minimiser le biais	vs Compromis biais-variance

Où on va parler de l'estimation des paramètres à l'aide de l'échantillon d'observations.

- Il s'agit du lieu de rencontre des **statistiques inférentielles** et de **l'apprentissage machine**!

1.3.1. STATISTIQUES INFÉRENTIELLES : MINIMISER LE BIAIS

1.3.1. STATISTIQUES INFÉRENTIELLES : MINIMISER LE BIAIS

Prenons pour acquis que...:

I. Nous avons posé une question de recherche.

- Ex.: Est-ce que le nombre d'heures de sommeil la nuit précédent un examen permet de prédire la performance à l'examen.
- Hypothèse: il existe une relation linéaire entre le sommeil et la performance à un examen.
- Formalisation de l'hypothèse à travers une régression linéaire simple: $Y = b_0 + b_1 X_1 + \epsilon$
 - Où: Y : performance à l'examen,
 - X_1 : heures de sommeil,
 - b_1 : paramètre reflétant l'importance de X_1 dans la prédition de Y ,
 - b_0 : paramètre reflétant la valeur de Y quand X_1 vaut 0.

II. Nous avons posé une hypothèse nulle (H_0)

- Il n'existe aucune relation entre les deux. $H_0 : b_1 = 0$

III. Nous avons posé une hypothèse alternative (H_A)

- Il existe une relation entre les deux: $H_A : b_1 \neq 0$

IV. On a récolté un échantillon : un groupe d'observations.

- Le chercheur vérifie les postulats permettant de valider la loi générale.
- **On doit maintenant utiliser ce groupe d'observations pour estimer les valeurs des paramètres du modèle du chercheur (ici, particulièrement, estimer les valeurs de b_0 et b_1).**

V. Il ne restera ensuite plus qu'à conclure.

- Le chercheur vérifie quelle était la probabilité d'obtenir les valeurs des paramètres estimés à partir de l'échantillon, si H_0 est vraie (i.e. la valeur p).
- Si cette probabilité est plus faible que la probabilité d'erreur de type 1 maximale établie au début, on rejette H_0 .

1.3.1. STATISTIQUES INFÉRENTIELLES : MINIMISER LE BIAIS

« Estimer » la valeur des paramètres b_0 et b_1 signifie... :

- Trouver des valeurs de b_0 et b_1 qui soient les plus **représentatives** possibles des vraies valeurs dans la population.

Dans le cadre des statistiques inférentielles, **on n'a généralement recours qu'à l'échantillon** pour estimer ces paramètres.

- On considère alors que les valeurs des paramètres b_0 et b_1 les plus représentatives de la population seront...
...les valeurs des paramètres b_0 et b_1 les plus représentatives de l'échantillon!

Pour déterminer quelles sont les valeurs des paramètres b_0 et b_1 qui sont les plus représentatives de l'échantillon, on doit définir un critère... Ce critère correspond ici à ce qu'on appelle une **fonction de coût** (*cost function*).

- Une **fonction** correspond à une relation qui associe chaque valeur d'un ensemble de départ avec une seule valeur d'un ensemble d'arrivée.
- Un **coût** correspond ici au degré de fausseté.
- Une **fonction de coût** prend donc ici en entrée les valeurs prédites par le modèle, les compare avec les valeurs réelles, puis renvoie une valeur représentant le degré de fausseté du modèle.

Dans le cadre d'une régression linéaire, on utilise généralement la fonction de coût correspondant à la **somme des carrés de l'erreur de prédition** (SC).

$$SC = \sum_{i=1}^N (Y_i - \hat{Y}_i)^2$$

Note: un « chapeau » sur une variable indique qu'il s'agit d'une valeur « **prédite** » et non de la vraie valeur de l'observation.

- où : N : nombre d'exemples dans l'échantillon (ex. nombre de participants)
 i : $i^{\text{ème}}$ exemple de l'échantillon (ex. $i^{\text{ème}}$ participant)
 $\hat{Y}_i = \hat{b}_{0i} + \hat{b}_{1i} X_{1i}$: la valeur prédite (estimée) du $i^{\text{ème}}$ exemple dans l'échantillon
 Y_i : la valeur réelle du $i^{\text{ème}}$ exemple dans l'échantillon

On met les erreurs au carré, parce que si l'on additionnait simplement les valeurs d'erreurs, les erreurs positives et négatives s'annuleraient et l'on sous-estimerait la fausseté du modèle.

1.3.1. STATISTIQUES INFÉRENTIELLES : MINIMISER LE BIAIS

« Estimer » la valeur des paramètres b_0 et b_1 signifie donc ici... :

- Trouver les valeurs de \hat{b}_0 et \hat{b}_1 qui permettent de **MINIMISER SC**.

$$SC = \sum_{i=1}^N (y_i - \hat{y}_i)^2$$

Les valeurs de \hat{b}_0 et \hat{b}_1 qui permettent de minimiser SC sont donc les plus représentatives de l'échantillon.

- Toutefois, sont-elles représentatives des vraies valeurs de b_0 et de b_1 dans la population?

On peut démontrer (mais on ne le fera pas ici) que... :

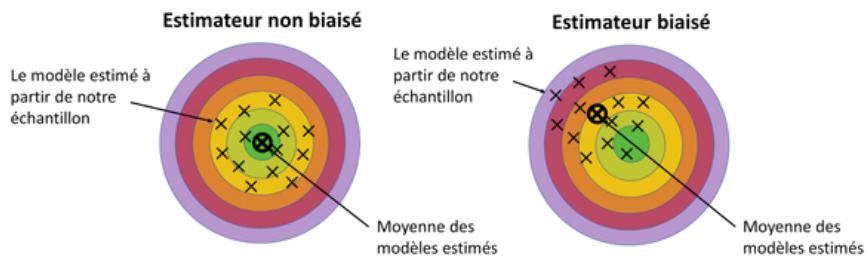
- SI les postulats (ici normalité, homoscédasticité, linéarité, indépendance des observations) sont respectés,
- ALORS la fonction de coût « SC » permet de trouver des valeurs estimées **NON BIAISÉES** pour les paramètres b_0 et b_1 .

Avoir des valeurs estimées **NON BIAISÉES** pour les paramètres veut dire que... :

- SI on tirait aléatoirement une infinité d'échantillons provenant de la même population,
- ALORS la moyenne des valeurs estimées du paramètre \hat{b}_0 sera égale à la valeur réelle du paramètre b_0 dans la population et la moyenne des valeurs estimées du paramètre \hat{b}_1 sera égale à la valeur réelle du paramètre b_1 dans la population.

Exemple d'un estimateur non biaisé (gauche) et d'un estimateur biaisé (droite).

- Le centre de la cible correspond à la vraie valeur du paramètre dans la population.
- Chaque X correspond à la valeur estimée du paramètre à partir d'un échantillon différent provenant de la même population.



1.3.1. STATISTIQUES INFÉRENTIELLES : MINIMISER LE BIAIS

En statistiques inférentielles, on accorde une très grande importance à l'obtention d'estimateurs non biaisés.

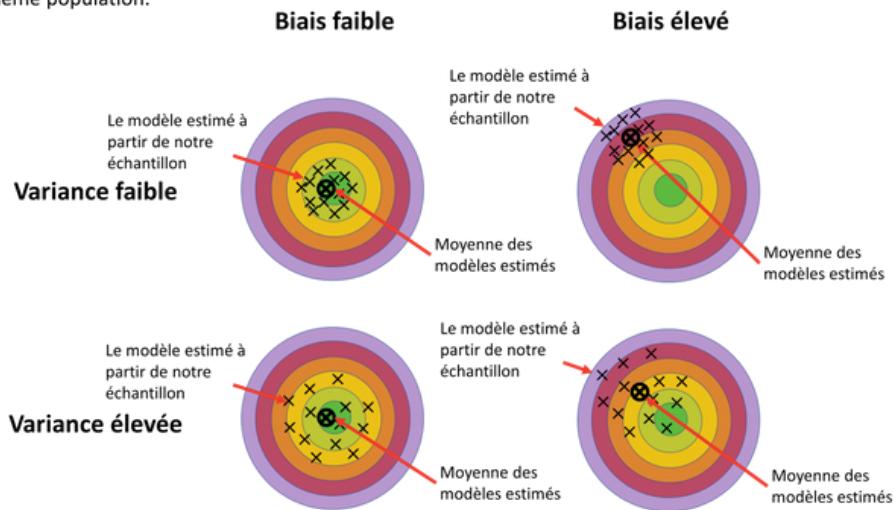
Or, on observe que les différents modèles obtenus à l'aide d'un estimateur non biaisé pour différents échantillons tirés d'une même population peuvent être très variables!

- On nomme **VARIANCE** la variabilité des valeurs des paramètres que l'on obtiendrait en tirant aléatoirement une infinité d'échantillons de la même population.

Ainsi, le degré de « **fausseté** » du modèle estimé dépend en fait de deux formes d'erreur:

- Le **BIAIS**
- La **VARIANCE**

Illustrons l'impact du biais et de la variance sur la distribution des modèles estimés à l'aide de différents échantillons d'une même population.



1.3.1. STATISTIQUES INFÉRENTIELLES : MINIMISER LE BIAIS

La **SOMME** du **BIAIS** et de la **VARIANCE** correspond à l'« **ERREUR DE GÉNÉRALISATION** » du modèle.

- Plus l'erreur de généralisation est élevée, plus on risque d'obtenir un modèle différent à partir d'un autre échantillon.
- Plus l'erreur de généralisation est élevée, moins les résultats sont reproductibles.

En statistiques inférentielles, au moment d'estimer le modèle, on ne tient compte que du biais.

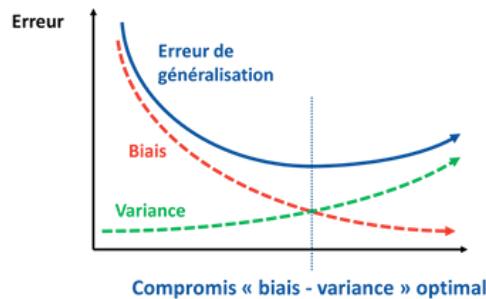
- On cherche à **MINIMISER LE BIAIS**.

Or, une diminution du biais implique généralement... une augmentation de la variance.

- Ceci implique qu'un modèle obtenu en minimisant le biais risque d'être peu généralisable.

Voici une illustration de la relation entre le biais, la variance et l'erreur de généralisation.

- Notons qu'un certain compromis entre le **biais** et la **variance** permet de minimiser l'**erreur de généralisation**.



1.3.1. STATISTIQUES INFÉRENTIELLES : MINIMISER LE BIAIS

Ainsi, en statistiques inférentielles, le modèle est estimé en tenant uniquement compte du biais.

- Sans tenir compte de la variance du modèle

Néanmoins, le test de significativité de l'hypothèse nulle, lui, prend la variance du modèle en considération.

- En effet, un test de significativité de H_0 correspond à un rapport « signal sur bruit ».

Par exemple, dans le cadre de la régression linéaire simple, on effectue, entre autres, un test t sur le paramètre b_1 .

$$t = \frac{\text{signal}}{\text{bruit}}$$

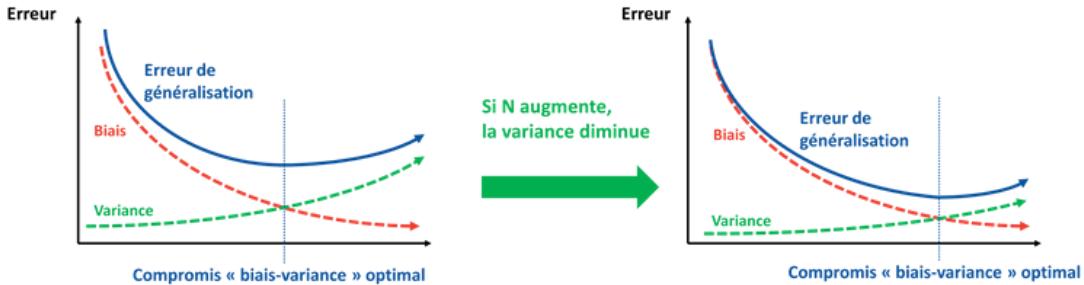
Le signal correspond à la différence entre la valeur estimée de \hat{b}_1 et la valeur correspondant à l'hypothèse nulle H_0 (i.e. généralement 0).

$$\text{signal} = \hat{b}_1 - 0$$

Le bruit correspond à la variance du modèle, estimé à l'aide de l'erreur type (i.e. une estimation de l'écart type des valeurs de \hat{b}_1 que l'on obtiendrait à partir d'une infinité d'échantillons tirés d'une même population).

$$\text{bruit} = \text{erreur type} = \frac{s}{\sqrt{N}}$$

- s correspond à l'écart type de l'erreur dans l'échantillon.
- N correspond à la taille de l'échantillon.
- ❖ Ce qui est à retenir ici: plus N est grand, plus la variance du modèle est petite!



1.3.1. STATISTIQUES INFÉRENTIELLES : MINIMISER LE BIAIS

On dit que plus N est grand, plus la variance du modèle est petite.

➤ Voyons voir pourquoi...

Dans une distribution d'échantillonnage... :

1. Plus les échantillons sont grands, plus ils sont représentatifs de la population.
2. Plus ils sont représentatifs de la population, plus ils sont similaires entre eux.
3. Plus ils sont similaires entre eux, plus la variabilité de la distribution d'échantillonnage est faible.

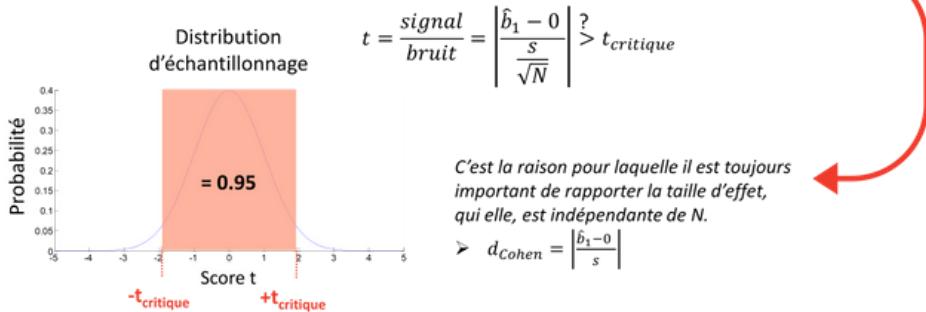
Ainsi, plus N est grand, plus la variance des modèles sera faible.

- Plus N est grand, plus la variabilité des erreurs « dans l'échantillon » (i.e. s) surestime la variance des modèles.
- C'est pourquoi on divise s par N (par \sqrt{N} pour être exact) : plus N est grand, plus le bruit dans le test d'hypothèse est petit.

$$\text{bruit} = \text{erreur type} = \frac{s}{\sqrt{N}}$$

Maintenant, si on continue le raisonnement commencé plus haut, on obtient:

4. Plus la variabilité de la distribution d'échantillonnage est faible, plus le bruit dans le test est faible.
5. Plus le bruit dans le test est faible, plus la valeur du test est grande pour un même signal.
6. Plus la valeur du test est grande pour un même signal, plus il est facile de dépasser les valeurs critique du test.
7. Plus il est facile de dépasser les valeurs critiques du test, plus il est facile de rejeter H_0 .



1.3.1. STATISTIQUES INFÉRENTIELLES : MINIMISER LE BIAIS

RÉCAPITULATIF pour la méthode du test de significativité de H_0 dans le cadre des statistiques inférentielles...

- La forme du modèle correspondant à H_A est fixée de manière rigide par l'expérimentateur.
 - La forme du modèle proposé par le chercheur est inspirée principalement de la documentation théorique.
- La forme du modèle proposé par le chercheur est interprétable.
 - Les variables et leurs relations décrivent explicitement le mécanisme réel étudié.
 - Le nombre de variables et d'interactions entre ces variables est faible.
 - Donc... le nombre de paramètres (p) à estimer est faible
(i.e. on a généralement une structure de données « longue » ; $N >> p$).
 - Ceci limite l'espace de recherche du chercheur (i.e. l'univers des modèles potentiels explorés).
- Les paramètres du modèle sont estimés de manière à minimiser le biais.
 - On vise des estimateurs non biaisés, car l'objectif est d'obtenir une conclusion sur un groupe et non de faire des prédictions sur des individus.
- Pour que la conclusion sur un groupe soit généralisable, le test d'hypothèse prend en considération la variance des paramètres estimés.
 - Plus N est grand, plus le bruit est faible.
 - Plus le bruit est faible, plus le signal minimum permettant de rejeter H_0 est petit (i.e. plus le test est dit « puissant »).
 - C'est pourquoi on doit également rapporter la taille d'effet (i.e. est-ce que l'effet est important).
- La validité de toute la méthode repose sur le respect de plusieurs postulats.
 - La vérification de ces postulats est généralement approximative et met en danger la généralisabilité des conclusions.

1.3.2. APPRENTISSAGE MACHINE : COMPROMIS BIAIS-VARIANCE

1.3.2. APPRENTISSAGE MACHINE : COMPROMIS BIAIS-VARIANCE

Rappelons qu'en apprentissage machine, il n'y a ni H_0 , ni H_A .

- I. On cible un problème à résoudre.
 - On doit transformer des variables d'entrée en variables de sortie appropriées.
 - On doit trouver « la bonne fonction ».
- II. Le modèle (i.e. la bonne fonction) induite à partir des données.
 - Le modèle inclut généralement un grand nombre de variables.
 - Chaque variable est accompagnée d'un paramètre, qui reflète l'importance de la variable à l'intérieur du modèle.
- III. On récolte un échantillon : généralement un grand nombre d'observations.
 - On divise ces observations en au moins deux sous-ensembles: un ensemble d'entraînement et un ensemble de test.
- IV. On construit le modèle à l'aide des exemples de l'ensemble d'entraînement.
 - On utilise l'ensemble d'entraînement pour estimer les valeurs des paramètres (i.e. l'importance des différentes variables prédictives).
- V. On évalue le modèle à l'aide des exemples de l'ensemble de test.
 - On vérifie la capacité du modèle à prédire des nouveaux exemples qui n'ont jamais été utilisés pour l'entraîner.

Le lieu de rencontre des statistiques inférentielles et de l'apprentissage machine est donc L'ESTIMATION (DES PARAMÈTRES) DU MODÈLE.

- Deux différences importantes:
 1. En apprentissage machine, on utilise au moins deux ensembles de données différents plutôt qu'un seul.
 2. En apprentissage machine, on cherche à trouver le meilleur compromis biais-variance plutôt qu'à minimiser le biais.

1.3.2. APPRENTISSAGE MACHINE : COMPROMIS BIAIS-VARIANCE

Tout ce que nous avons vu dans la section précédente quant à l'estimation des paramètres s'applique ici.

Toutefois, on a maintenant deux ensembles de données:

- Un ensemble d'entraînement.
- Un ensemble de test.

En statistiques inférentielles:

- On estime les valeurs des paramètres du modèle à l'aide des données de l'échantillon.
- On cherche à minimiser l'erreur de prédiction à l'intérieur de l'échantillon.

En apprentissage machine:

- On estime les valeurs des paramètres du modèle à l'aide des données de l'ensemble d'entraînement.
- On cherche à minimiser l'erreur de prédiction à l'extérieur de l'ensemble d'entraînement.
- On cherche à minimiser l'erreur de prédiction dans l'ensemble de test.

En apprentissage machine, on cherche à maximiser la généralisabilité du modèle estimé.

- On veut faire les meilleures prédictions possibles au niveau de nouvelles données individuelles.

1.3.2. APPRENTISSAGE MACHINE : COMPROMIS BIAIS-VARIANCE

Le défi en apprentissage machine est de trouver le bon compromis entre le biais et la variance.

La complexité du modèle induit n'a pas le même impact dans l'ensemble d'entraînement que dans l'ensemble de test.

- À l'intérieur de l'ensemble d'entraînement, plus le modèle est complexe, meilleure est la prédiction.
- À l'intérieur de l'ensemble de test, la relation n'est pas si simple...

La complexité du modèle est associée au nombre de paramètres estimés et à leur importance. Par exemple, pour augmenter la complexité du modèle suivant: $\hat{Y} = \hat{b}_0 + \hat{b}_1 X_1 + \hat{b}_2 X_2$, on peut:

- Augmenter le nombre de prédicteurs (ex. X_3, X_4, X_5, \dots).
- Augmenter l'ordre d'un prédicteur (ex. $X_1^2, X_1^3, X_1^4, \dots$).
- Ajouter une interaction entre deux termes (ex. $X_1 X_2, \dots$)
- Etc.

Un modèle trop simple risque de souffrir de sous-apprentissage.

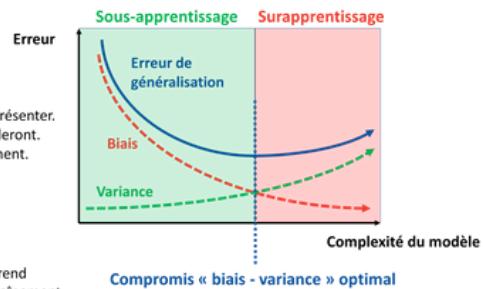
- Le modèle induit n'est pas suffisamment complexe vis-à-vis la fonction qu'il essaie de représenter.
- La variance sera faible: les modèles induits à partir de différents échantillons se ressembleront.
- Le biais sera élevé: l'erreur de prédiction sera élevée, même dans l'ensemble d'entraînement.

- ❖ Dans l'ensemble d'entraînement, l'erreur de prédiction sera élevée.
- ❖ Dans l'ensemble de test, l'erreur de prédiction sera élevée.

Un modèle trop complexe risque de souffrir de surapprentissage.

- Le modèle induit est trop complexe vis-à-vis la fonction qu'il essaie de représenter. Il apprend à prédire non seulement le signal, mais aussi du bruit (par hasard) dans l'ensemble d'entraînement.
- La variance sera élevée: les modèles induits à partir de différents échantillons seront très différents (le bruit est aléatoire et donc différent dans différents échantillons).
- Le biais sera faible: l'erreur de prédiction sera faible peu importe l'ensemble d'entraînement.

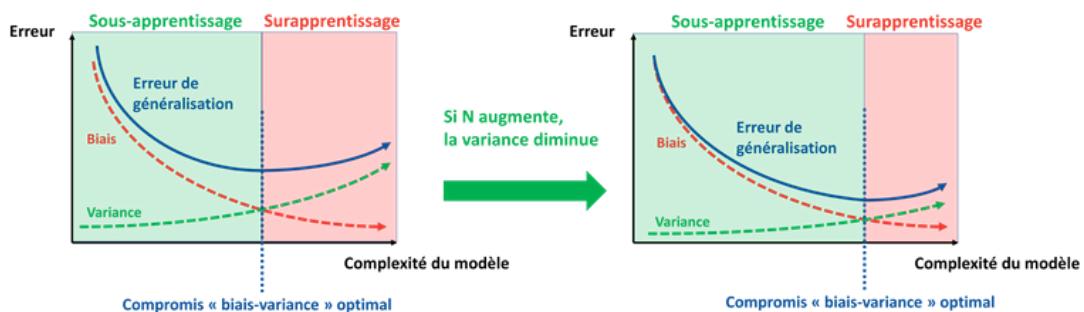
- ❖ Dans l'ensemble d'entraînement, l'erreur de prédiction sera faible.
- ❖ Dans l'ensemble de test, l'erreur de prédiction sera élevée.



1.3.2. APPRENTISSAGE MACHINE : COMPROMIS BIAIS-VARIANCE

Toutefois, comme on l'a vu plus tôt, augmenter N permet de réduire la variance du modèle.

Ainsi, dire qu'un modèle est « trop » ou « pas assez » complexe est contextuel à (entre autres) la taille de l'ensemble d'entraînement.



FAISONS UN EXEMPLE

In [1] :

```
import warnings
warnings.filterwarnings("ignore")
warnings.filterwarnings(action='ignore', category=DeprecationWarning)
warnings.filterwarnings(action='ignore', category=FutureWarning)

# -----
# -----  
# ÉTAPE 1 : importer les librairies utiles
# -----  
-----  
  
import numpy as np
import matplotlib.pyplot as plt

# -----
# -----  
# ÉTAPE 2 : importer les fonctions utiles
# -----  
-----  
  
from sklearn.pipeline import Pipeline
from sklearn.preprocessing import PolynomialFeatures
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split

# -----
# -----  
# ÉTAPE 3 : importer et préparer le jeu de données
# -----  
-----  
  
def true_fun(X):
    return np.cos(1.5 * np.pi * X)

np.random.seed(0)

n_samples = 100
degree = 10

X = np.sort(np.random.rand(n_samples))
y = true_fun(X) + np.random.randn(n_samples) * 0.1

X_train, X_test, y_train, y_test = train_test_split(X, y)

# -----
# -----  
# ÉTAPE 4 : entraîner le modèle (ensemble "Entraînement")
# -----  
-----  
  
polynomial_features = PolynomialFeatures(degree)
model = LinearRegression()
pipeline = Pipeline([("polynomial_features", polynomial_features),
```

```
("linear_regression", model]))  
pipeline.fit(X_train[:, np.newaxis], y_train)  
  
scores_train = pipeline.score(X_train[:, np.newaxis], y_train)  
  
# -----  
# -----  
# ÉTAPE 5 : vérifier la généralisabilité des résultats (ensemble "Test")  
# -----  
# -----  
scores_test = pipeline.score(X_test[:, np.newaxis], y_test)
```

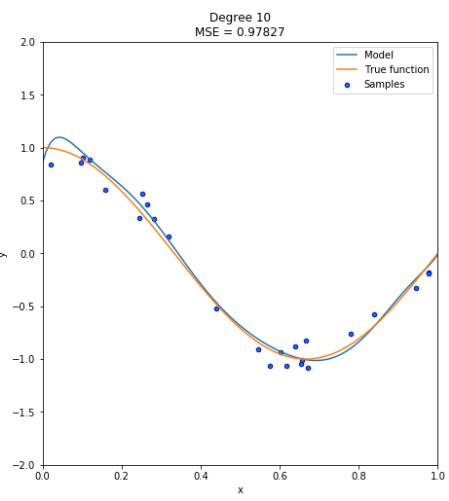
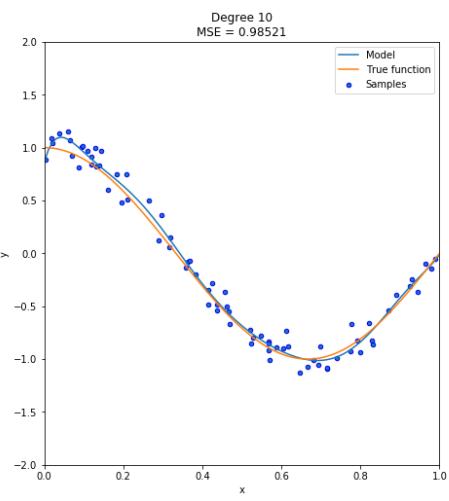
In [2]:

```
plt.figure(figsize=(16, 8))

plt.subplot(1, 2, 1)
X = np.linspace(0, 1, 100)
plt.plot(X, pipeline.predict(X[:, np.newaxis]), label="Model")
plt.plot(X, true_fun(X), label="True function")
plt.scatter(X_train, y_train, edgecolor='b', s=20, label="Samples")
plt.xlabel("x")
plt.ylabel("y")
plt.xlim((0, 1))
plt.ylim((-2, 2))
plt.legend(loc="best")
plt.title("Degree {} \nMSE = {:.5f}".format(degree, scores_train.mean()))

plt.subplot(1, 2, 2)
plt.plot(X, pipeline.predict(X[:, np.newaxis]), label="Model")
plt.plot(X, true_fun(X), label="True function")
plt.scatter(X_test, y_test, edgecolor='b', s=20, label="Samples")
plt.xlabel("x")
plt.ylabel("y")
plt.xlim((0, 1))
plt.ylim((-2, 2))
plt.legend(loc="best")
plt.title("Degree {} \nMSE = {:.5f}".format(degree, scores_test.mean()))

plt.show()
```



In [3] :

```
# -----
# -----  
# ÉTAPE 1 : importer les librairies utiles  
# -----  
-----  
  
import numpy as np  
import matplotlib.pyplot as plt  
  
# -----  
-----  
# ÉTAPE 2 : importer les fonctions utiles  
# -----  
-----  
  
from sklearn.pipeline import Pipeline  
from sklearn.preprocessing import PolynomialFeatures  
from sklearn.linear_model import LinearRegression  
from sklearn.model_selection import train_test_split  
  
# -----  
-----  
# ÉTAPE 3 : importer et préparer le jeu de données  
# -----  
-----  
  
def true_fun(X):  
    return np.cos(1.5 * np.pi * X)  
  
np.random.seed(0)  
  
n_samples = 20  
degrees = [1, 2, 4, 8, 16]  
  
X = np.sort(np.random.rand(n_samples))  
y = true_fun(X) + np.random.randn(n_samples) * 0.1  
  
X_train, X_test, y_train, y_test = train_test_split(X, y)  
  
# -----  
-----  
# ÉTAPE 4 : entraîner le modèle (ensemble "Entraînement")  
# -----  
-----  
  
plt.figure(figsize=(15, 3))  
  
for i in range(len(degrees)):  
  
    ax = plt.subplot(1, len(degrees), i + 1)  
    plt.setp(ax, xticks=(), yticks=())
```

```

polynomial_features = PolynomialFeatures(degree=degrees[i])

model = LinearRegression()

pipeline = Pipeline([("polynomial_features", polynomial_features),
                    ("linear_regression", model)])

pipeline.fit(X_train[:, np.newaxis], y_train)

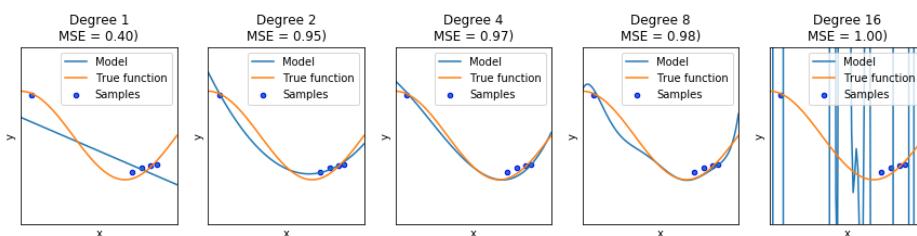
scores_train = pipeline.score(X_train[:, np.newaxis], y_train)

# -----
# ----- # ÉTAPE 5 : vérifier la généralisabilité des résultats (ensemble "Test")
# -----
# ----- scores_test = pipeline.score(X_test[:, np.newaxis], y_test)

X = np.linspace(0, 1, 100)
plt.plot(X, pipeline.predict(X[:, np.newaxis]), label="Model")
plt.plot(X, true_fun(X), label="True function")
plt.scatter(X_test, y_test, edgecolor='b', s=20, label="Samples")
plt.xlabel("x")
plt.ylabel("y")
plt.xlim((0, 1))
plt.ylim((-2, 2))
plt.legend(loc="best")
plt.title("Degree {} \nMSE = {:.2f}".format(degrees[i], scores_train.mean()))

plt.show()

```



1.3.2. APPRENTISSAGE MACHINE : COMPROMIS BIAIS-VARIANCE

Pour trouver un compromis entre le biais et la variance, on va modifier la fonction que l'on cherche à minimiser.

En **statistiques inférentielles**, on essaie de **minimiser** uniquement:

- Une **fonction de coût** qui calcule l'erreur de prédiction à l'intérieur de l'échantillon.

En **apprentissage machine**, on essaie de **minimiser** en même temps:

- Une **fonction de coût** qui calcule l'erreur de prédiction à l'intérieur de l'ensemble d'entraînement.
 - La **complexité du modèle** (i.e. la taille des paramètres du modèle).

La fonction totale à minimiser est appelée « fonction de perte » (*loss function*)

- Voici un exemple de fonction de perte:

$$\text{Fonction de coût} \quad \left[\frac{1}{N} \sum_{t=1}^N (Y_t - \hat{Y}_t)^2 \right] + C \|\hat{b}\|$$

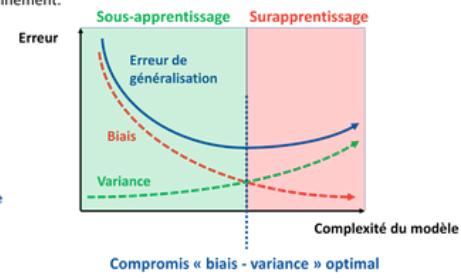
où $\|\hat{b}\| = \sqrt{\sum_{j=1}^D b_j^2}$

Complexité du modèle
« Régularisation »)

➤ où D = nombre de p

- où D = nombre de paramètres estimés
- où C est un « hyperparamètre » fixé par

> ou c'est un « hyperparamètre » fixé par le chercheur



Notons que:

- Un **paramètre** est estimé par l'algorithme d'apprentissage utilisé et reflète l'importance des différents prédicteurs.
 - Un **hyperparamètre** est spécifié par chercheur (ou estimé par un second algorithme d'apprentissage) et spécifie la forme d'un modèle ou d'un algorithme d'apprentissage.
 - Un **algorithme** est une séquence d'opérations finies permettant de résoudre une classe de problèmes.

On souhaite diminuer la complexité du modèle, car on veut diminuer la variance.

On souhaite diminuer l'erreur de la fonction de coût, car on veut diminuer le biais.

- Or, diminuer la complexité du modèle est généralement associé à une augmentation de l'erreur de la fonction de coût.
 - Et diminuer l'erreur de la fonction de coût est généralement associé à une augmentation de la complexité du modèle.

La fonction de perte implique une tension entre diminuer le biais (i.e. la fonction de coût) et diminuer la variance (i.e. la complexité du modèle).

- Les paramètres estimés reflètent ainsi un compromis entre le biais et la variance.
 - Ceci permet de minimiser l'erreur de généralisation.

1.3.2. APPRENTISSAGE MACHINE : COMPROMIS BIAIS-VARIANCE

RÉCAPITULATIF pour l'apprentissage machine.

- La forme du modèle est induite automatiquement à partir des données.
 - En réalité, le chercheur peut contraindre plus ou moins fortement la forme du modèle.
 - Concernant la forme du modèle...
 - L'algorithme d'apprentissage machine prend généralement en entrée un grand nombre de variables (appelées « caractéristiques »).
 - Le nombre d'interactions entre ces variables peut être élevé.
 - Donc... le nombre de paramètres (p) à estimer est souvent élevé (i.e. on a souvent une structure de données « large » ; $N \ll P$).
 - Le modèle est induit à partir d'un ensemble de données d'entraînement.
 - On cherche alors à estimer les paramètres de manière à minimiser l'erreur de généralisation.
 - On vise un compromis entre une diminution du biais et une diminution de la variance.
 - Parce qu'on cherche à prédire de nouvelles données individuelles.
 - La validité du modèle induit est évaluée à l'aide d'un ensemble de test.
 - Les données de l'ensemble de test n'ont alors jamais été « vues » par le modèle.

COMPARATIF DES STATISTIQUES INFÉRENTIELLES ET DE L'APPRENTISSAGE MACHINE

STATISTIQUES INFÉRENTIELLES CLASSIQUES		APPRENTISSAGE MACHINE	
OBJECTIF: COMPRENDRE UN PHÉNOMÈNE		OBJECTIF: PRÉDIRE DE NOUVELLES OBSERVATIONS	
MODÈLES SIMPLES (UNIDIMENSIONNELS / PEU DE DIMENSIONS)		MODÈLES COMPLEXES (MULTIDIMENSIONNELS)	
MODÈLES INTERPRÉTABLES		MODÈLES SOUVENT PEU/PAS INTERPRÉTABLES	
DONNÉES LONGUES		DONNÉES LARGES	
PROCESSUS DÉDUCTIF		PROCESSUS INDUCTIF	
NOMBREUX POSTULATS À RESPECTER		PEU/PAS DE POSTULATS À RESPECTER	
LA CONCLUSION DE L'ANALYSE EST BINAIRE (REJET OU NON REJET DE H_0)		LA CONCLUSION DE L'ANALYSE EST CONTINUE (CAPACITÉ À GÉNÉRALISER)	
LA CONCLUSION DE L'ANALYSE EST LIMITÉE À DES GROUPES D'OBSERVATIONS		LA CONCLUSION DE L'ANALYSE EST APPLICABLE À DE NOUVELLES DONNÉES INDIVIDUELLES	
ON ESTIME LES PARAMÈTRES EN MINIMISANT LE BIAIS		ON ESTIME LES PARAMÈTRES EN MINIMISANT L'ERREUR DE GÉNÉRALISATION (BIAIS-VARIANCE)	
ON VEUT GÉNÉRALEMENT LE PLUS GRAND N POSSIBLE		ON VEUT GÉNÉRALEMENT LE PLUS GRAND N POSSIBLE	
DANS LES FAITS, LA PLUPART DE CES POINTS SONT LES EXTRÉMITÉS D'UN CONTINUUM!			

2. APPRENTISSAGE SUPERVISÉ vs APPRENTISSAGE NON SUPERVISÉ

	Apprentissage supervisé		Apprentissage non supervisé
2.1.	Régression	2.3.	Réduction de dimension
2.2.	Classification	2.4.	Regroupement

2. APPRENTISSAGE SUPERVISÉ vs APPRENTISSAGE NON SUPERVISÉ

Structure des données

On a deux ensembles de données ayant chacun une structure sous la forme d'un tableau.

- N : nombre d'« exemples » (*examples*).
- x : exemple correspondant à un vecteur de « caractéristiques » (*features*).
- D : nombre de caractéristiques (aussi appelées « dimensions ») que comporte chaque exemple.
- ❖ $x_2^{(3)}$: 3^e caractéristique du 2^e exemple.

Exemple	$x^{(1)}$	$x^{(2)}$	$x^{(3)}$	$x^{(4)}$	$x^{(5)}$	$x^{(\dots)}$	$x^{(D)}$
x_1	$x_1^{(1)}$	$x_1^{(2)}$	$x_1^{(3)}$	$x_1^{(4)}$	$x_1^{(5)}$	$x_1^{(\dots)}$	$x_1^{(D)}$
x_2	$x_2^{(1)}$	$x_2^{(2)}$	$x_2^{(3)}$	$x_2^{(4)}$	$x_2^{(5)}$	$x_2^{(\dots)}$	$x_2^{(D)}$
x_3	$x_3^{(1)}$	$x_3^{(2)}$	$x_3^{(3)}$	$x_3^{(4)}$	$x_3^{(5)}$	$x_3^{(\dots)}$	$x_3^{(D)}$
x_4	$x_4^{(1)}$	$x_4^{(2)}$	$x_4^{(3)}$	$x_4^{(4)}$	$x_4^{(5)}$	$x_4^{(\dots)}$	$x_4^{(D)}$
x_{\dots}	$x_{\dots}^{(1)}$	$x_{\dots}^{(2)}$	$x_{\dots}^{(3)}$	$x_{\dots}^{(4)}$	$x_{\dots}^{(5)}$	$x_{\dots}^{(\dots)}$	$x_{\dots}^{(D)}$
x_N	$x_N^{(1)}$	$x_N^{(2)}$	$x_N^{(3)}$	$x_N^{(4)}$	$x_N^{(5)}$	$x_N^{(\dots)}$	$x_N^{(D)}$

2. APPRENTISSAGE SUPERVISÉ vs APPRENTISSAGE NON SUPERVISÉ

Apprentissage supervisé

On souhaite prédire l'une des caractéristiques à partir des autres caractéristiques.

- La caractéristique que l'on souhaite prédire est nommée « cible » (*target*).
- La caractéristique correspondant à la cible est notée y .
- y_3 : 3^e exemple de la cible.
- La valeur que prend la cible pour un exemple donné est nommée « étiquette » (*label*)
- ❖ $x_2^{(3)}$: 3^e caractéristique du 2^e exemple.

Exemple	$x^{(1)}$	$x^{(2)}$	$x^{(3)}$	$x^{(4)}$	$x^{(\dots)}$	$x^{(D)}$	y
x_1	$x_1^{(1)}$	$x_1^{(2)}$	$x_1^{(3)}$	$x_1^{(4)}$	$x_1^{(\dots)}$	$x_1^{(D)}$	y_1
x_2	$x_2^{(1)}$	$x_2^{(2)}$	$x_2^{(3)}$	$x_2^{(4)}$	$x_2^{(\dots)}$	$x_2^{(D)}$	y_1
x_3	$x_3^{(1)}$	$x_3^{(2)}$	$x_3^{(3)}$	$x_3^{(4)}$	$x_3^{(\dots)}$	$x_3^{(D)}$	y_3
x_4	$x_4^{(1)}$	$x_4^{(2)}$	$x_4^{(3)}$	$x_4^{(4)}$	$x_4^{(\dots)}$	$x_4^{(D)}$	y_4
x_{\dots}	$x_{\dots}^{(1)}$	$x_{\dots}^{(2)}$	$x_{\dots}^{(3)}$	$x_{\dots}^{(4)}$	$x_{\dots}^{(\dots)}$	$x_{\dots}^{(D)}$	y_{\dots}
x_N	$x_N^{(1)}$	$x_N^{(2)}$	$x_N^{(3)}$	$x_N^{(4)}$	$x_N^{(\dots)}$	$x_N^{(D)}$	y_N

2. APPRENTISSAGE SUPERVISÉ vs APPRENTISSAGE NON SUPERVISÉ

Apprentissage supervisé

Si les valeurs des étiquettes de la cible correspondent à des nombres réels (ex. taille) :

- Il s'agit d'un problème de « régression ».

Si les valeurs des étiquettes de la cible correspondent à un nombre fini de classes (ex. présence/absence de maladie) :

- Il s'agit d'un problème de « classification ».

Exemple	$x^{(1)}$	$x^{(2)}$	$x^{(3)}$	$x^{(4)}$	$x^{(\dots)}$	$x^{(D)}$	y
x_1	$x_1^{(1)}$	$x_1^{(2)}$	$x_1^{(3)}$	$x_1^{(4)}$	$x_1^{(\dots)}$	$x_1^{(D)}$	y_1
x_2	$x_2^{(1)}$	$x_2^{(2)}$	$x_2^{(3)}$	$x_2^{(4)}$	$x_2^{(\dots)}$	$x_2^{(D)}$	y_1
x_3	$x_3^{(1)}$	$x_3^{(2)}$	$x_3^{(3)}$	$x_3^{(4)}$	$x_3^{(\dots)}$	$x_3^{(D)}$	y_3
x_4	$x_4^{(1)}$	$x_4^{(2)}$	$x_4^{(3)}$	$x_4^{(4)}$	$x_4^{(\dots)}$	$x_4^{(D)}$	y_4
x_{\dots}	$x_{\dots}^{(1)}$	$x_{\dots}^{(2)}$	$x_{\dots}^{(3)}$	$x_{\dots}^{(4)}$	$x_{\dots}^{(\dots)}$	$x_{\dots}^{(D)}$	y_{\dots}
x_N	$x_N^{(1)}$	$x_N^{(2)}$	$x_N^{(3)}$	$x_N^{(4)}$	$x_N^{(\dots)}$	$x_N^{(D)}$	y_N

2. APPRENTISSAGE SUPERVISÉ vs APPRENTISSAGE NON SUPERVISÉ

Apprentissage non supervisé

On souhaite modifier la structure des données.

- Il n'y a aucune caractéristique « cible ».

On pourrait alors souhaiter modifier les colonnes (réduction des dimensions).

- Modifier l'espace des caractéristiques dans lequel sont distribués les exemples.

On pourrait plutôt souhaiter modifier les lignes (regroupement).

- Rassembler les exemples par « groupes » (*clusters*).

Exemple	$x^{(1)}$	$x^{(2)}$	$x^{(3)}$	$x^{(4)}$	$x^{(5)}$	$x^{(\dots)}$	$x^{(D)}$
x_1	$x_1^{(1)}$	$x_1^{(2)}$	$x_1^{(3)}$	$x_1^{(4)}$	$x_1^{(5)}$	$x_1^{(\dots)}$	$x_1^{(D)}$
x_2	$x_2^{(1)}$	$x_2^{(2)}$	$x_2^{(3)}$	$x_2^{(4)}$	$x_2^{(5)}$	$x_2^{(\dots)}$	$x_2^{(D)}$
x_3	$x_3^{(1)}$	$x_3^{(2)}$	$x_3^{(3)}$	$x_3^{(4)}$	$x_3^{(5)}$	$x_3^{(\dots)}$	$x_3^{(D)}$
x_4	$x_4^{(1)}$	$x_4^{(2)}$	$x_4^{(3)}$	$x_4^{(4)}$	$x_4^{(5)}$	$x_4^{(\dots)}$	$x_4^{(D)}$
x_{\dots}	$x_{\dots}^{(1)}$	$x_{\dots}^{(2)}$	$x_{\dots}^{(3)}$	$x_{\dots}^{(4)}$	$x_{\dots}^{(5)}$	$x_{\dots}^{(\dots)}$	$x_{\dots}^{(D)}$
x_N	$x_N^{(1)}$	$x_N^{(2)}$	$x_N^{(3)}$	$x_N^{(4)}$	$x_N^{(5)}$	$x_N^{(\dots)}$	$x_N^{(D)}$

2.1. APPRENTISSAGE SUPERVISÉ : RÉGRESSION

Apprentissage supervisé		Apprentissage non supervisé	
2.1.	Régression	2.3.	Réduction de dimension
2.2.	Classification	2.4.	Regroupement

2.1. APPRENTISSAGE SUPERVISÉ : RÉGRESSION

Régression

Les valeurs des étiquettes de la cible correspondent à des nombres réels (ex. taille).

Évaluation de la performance du modèle.

- On utilise généralement la moyenne des erreurs au carré (*mean squared error ; MSE*).

Fonction de coût:

$$MSE = \frac{1}{N} \sum_{i=1}^N (Y_i - \hat{Y}_i)^2$$

À cela s'ajoute un terme de « régularisation » permettant de pénaliser la complexité du modèle.

- Compromis biais-variance.

Exemple de régularisation: ajout de la fonction

$$\lambda \|\hat{b}\|$$

- où $\|\hat{b}\| = \sqrt{\sum_{j=1}^p b_j^2}$
- où p = nombre de paramètres estimés
- où λ est un « **hyperparamètre** » fixé par le chercheur

On obtient ainsi la fonction de perte suivante à minimiser:

$$\frac{1}{N} \sum_{i=1}^N (Y_i - \hat{Y}_i)^2 + \lambda \|\hat{b}\|$$

2.2. APPRENTISSAGE SUPERVISÉ : CLASSIFICATION

	Apprentissage supervisé	Apprentissage non supervisé
2.1.	Régression	2.3. Réduction de dimension
2.2.	Classification	2.4. Regroupement

2.2. APPRENTISSAGE SUPERVISÉ : CLASSIFICATION

Classification

Les valeurs des étiquettes de la cible correspondent à un nombre fini de classes (ex. présence/absence de maladie) :

Évaluation de la performance du modèle.

- Les mesures de performance sont généralement dérivées d'une « matrice de confusion ».
- Il en existe une très grande variété utilisant différentes combinaisons des cellules de base.

Matrice de confusion:

		VALEURS RÉELLES	
		Présence de maladie	Absence de maladie
VALEURS PRÉDITES	Présence de maladie	10 (VP)	10 (FP)
	Absence de maladie	1 (FN)	200 (VN)

2.3. APPRENTISSAGE NON SUPERVISÉ : RÉDUCTION DE DIMENSION

Apprentissage supervisé		Apprentissage non supervisé	
2.1.	Régression	2.3.	Réduction de dimension
2.2.	Classification	2.4.	Regroupement

2.3. APPRENTISSAGE NON SUPERVISÉ : RÉDUCTION DE DIMENSION

Réduction de dimension

On souhaite modifier les colonnes.

- Modifier l'espace des caractéristiques dans lequel sont distribués les exemples.

Utilités:

- Visualisation des données: dimensionnalité élevée vers dimensionnalité faible.
- Pré-traitement des données: par exemple, pour réduire la complexité du modèle si N est faible.
 - Éviter une trop grande variance du modèle. Éviter le surapprentissage.

Exemple	$x^{(1)}$	$x^{(2)}$	$x^{(3)}$	$x^{(4)}$	$x^{(5)}$	$x^{(\dots)}$	$x^{(D)}$
x_1	$x_1^{(1)}$	$x_1^{(2)}$	$x_1^{(3)}$	$x_1^{(4)}$	$x_1^{(5)}$	$x_1^{(\dots)}$	$x_1^{(D)}$
x_2	$x_2^{(1)}$	$x_2^{(2)}$	$x_2^{(3)}$	$x_2^{(4)}$	$x_2^{(5)}$	$x_2^{(\dots)}$	$x_2^{(D)}$
x_{\dots}	$x_{\dots}^{(1)}$	$x_{\dots}^{(2)}$	$x_{\dots}^{(3)}$	$x_{\dots}^{(4)}$	$x_{\dots}^{(5)}$	$x_{\dots}^{(\dots)}$	$x_{\dots}^{(D)}$
x_N	$x_N^{(1)}$	$x_N^{(2)}$	$x_N^{(3)}$	$x_N^{(4)}$	$x_N^{(5)}$	$x_N^{(\dots)}$	$x_N^{(D)}$



Exemple	$x^{(1')}$	$x^{(2')}$	$x^{(3')}$
x_1	$x_1^{(1)}$	$x_1^{(2)}$	$x_1^{(3)}$
x_2	$x_2^{(1)}$	$x_2^{(2)}$	$x_2^{(3)}$
x_{\dots}	$x_{\dots}^{(1)}$	$x_{\dots}^{(2)}$	$x_{\dots}^{(3)}$
x_N	$x_N^{(1)}$	$x_N^{(2)}$	$x_N^{(3)}$

2.4. APPRENTISSAGE NON SUPERVISÉ : REGROUPEMENT

Apprentissage supervisé		Apprentissage non supervisé	
2.1. Régression		2.3. Réduction de dimension	
2.2. Classification		2.4. Regroupement	

2.4. APPRENTISSAGE NON SUPERVISÉ : REGROUPEMENT

Regroupement

On souhaite modifier les lignes (regroupement).

- Rassembler les exemples par « groupes » (*clusters*).

Utilité:

- Dégager une structure **catégorielle** intrinsèque aux données.

Exemple	$x^{(1)}$	$x^{(2)}$	$x^{(3)}$	$x^{(4)}$	$x^{(5)}$	$x^{(\dots)}$	$x^{(D)}$
x_1	$x_1^{(1)}$	$x_1^{(2)}$	$x_1^{(3)}$	$x_1^{(4)}$	$x_1^{(5)}$	$x_1^{(\dots)}$	$x_1^{(D)}$
x_2	$x_2^{(1)}$	$x_2^{(2)}$	$x_2^{(3)}$	$x_2^{(4)}$	$x_2^{(5)}$	$x_2^{(\dots)}$	$x_2^{(D)}$
x_{\dots}	$x_{\dots}^{(1)}$	$x_{\dots}^{(2)}$	$x_{\dots}^{(3)}$	$x_{\dots}^{(4)}$	$x_{\dots}^{(5)}$	$x_{\dots}^{(\dots)}$	$x_{\dots}^{(D)}$
x_N	$x_N^{(1)}$	$x_N^{(2)}$	$x_N^{(3)}$	$x_N^{(4)}$	$x_N^{(5)}$	$x_N^{(\dots)}$	$x_N^{(D)}$



Classe	$x^{(1)}$	$x^{(2)}$	$x^{(3)}$	$x^{(4)}$	$x^{(5)}$	$x^{(\dots)}$	$x^{(D)}$
c_1	$x_1^{(1)}$	$x_1^{(2)}$	$x_1^{(3)}$	$x_1^{(4)}$	$x_1^{(5)}$	$x_1^{(\dots)}$	$x_1^{(D)}$
c_2	$x_2^{(1)}$	$x_2^{(2)}$	$x_2^{(3)}$	$x_2^{(4)}$	$x_2^{(5)}$	$x_2^{(\dots)}$	$x_2^{(D)}$
c_{\dots}	$x_{\dots}^{(1)}$	$x_{\dots}^{(2)}$	$x_{\dots}^{(3)}$	$x_{\dots}^{(4)}$	$x_{\dots}^{(5)}$	$x_{\dots}^{(\dots)}$	$x_{\dots}^{(D)}$
c_G	$x_N^{(1)}$	$x_N^{(2)}$	$x_N^{(3)}$	$x_N^{(4)}$	$x_N^{(5)}$	$x_N^{(\dots)}$	$x_N^{(D)}$