

APPRENTISSAGE SUPERVISÉ

UN PROGRAMME QUI APPREND

Objectif : induire la démarche de l'apprentissage par l'exemple.

- **Montrer** des exemples à votre programme en **lui disant de quoi il s'agit**.

UN PROGRAMME QUI APPREND

Objectif : induire la démarche de l'apprentissage par l'exemple.

- **Montrer** des exemples à votre programme en **lui disant de quoi il s'agit**.
- Lui faire **apprendre** une règle.

UN PROGRAMME QUI APPREND

Objectif : induire la démarche de l'apprentissage par l'exemple.

- **Montrer** des exemples à votre programme en **lui disant de quoi il s'agit**.
- Lui faire **apprendre** une règle.
- **Appliquer** la règle à de nouveaux exemples.

UN PROGRAMME QUI APPREND

Objectif : induire la démarche de l'apprentissage par l'exemple.

- **Montrer** des exemples à votre programme en **lui disant de quoi il s'agit**.
- Lui faire **apprendre** une règle.
- **Appliquer** la règle à de nouveaux exemples.
- **Évaluer** si les prédictions sont bonnes en les comparant à la réalité.

EXEMPLE 1: CLASSIFICATION BINAIRE

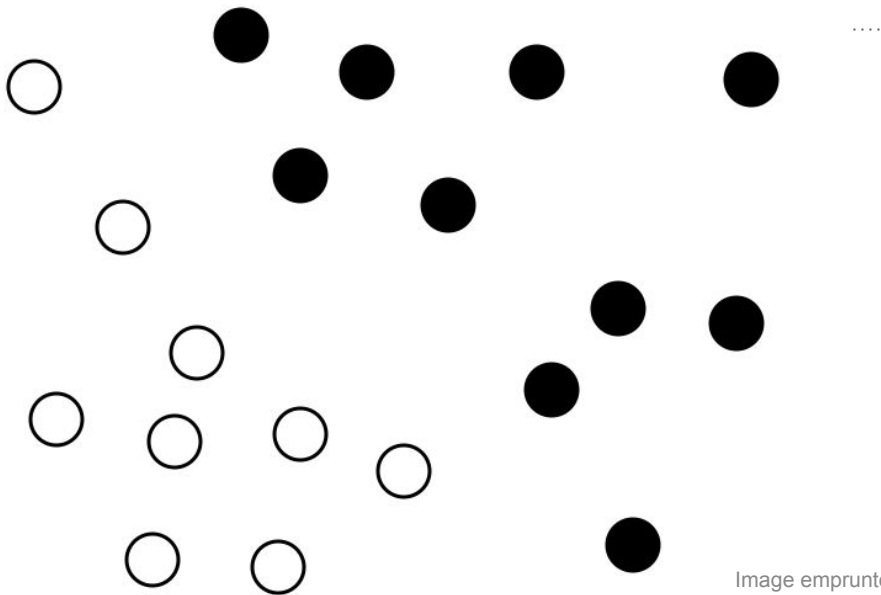


Image empruntée à
Jean-Philippe Vert

EXEMPLE 1: CLASSIFICATION BINAIRE

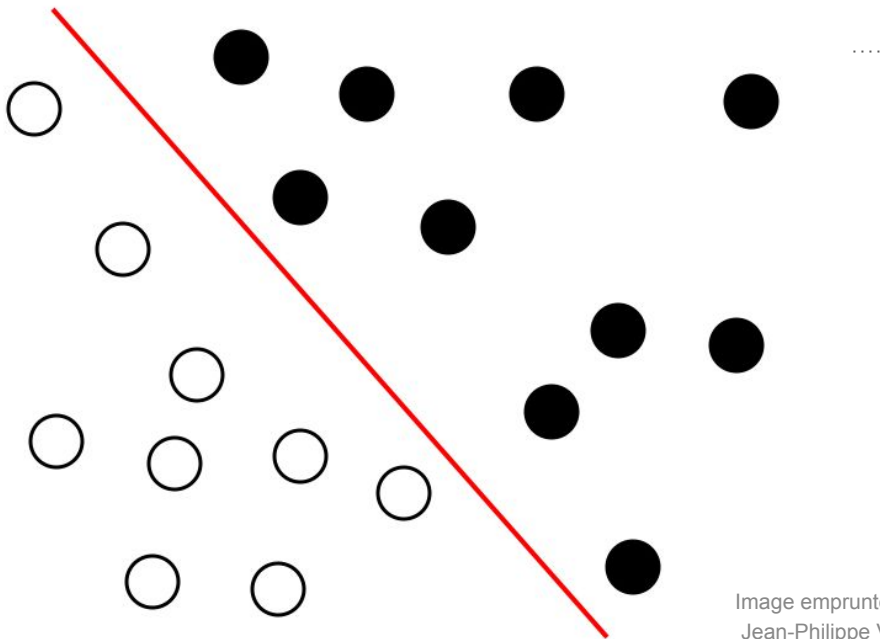


Image empruntée à
Jean-Philippe Vert

EXEMPLE 1: CLASSIFICATION BINAIRE

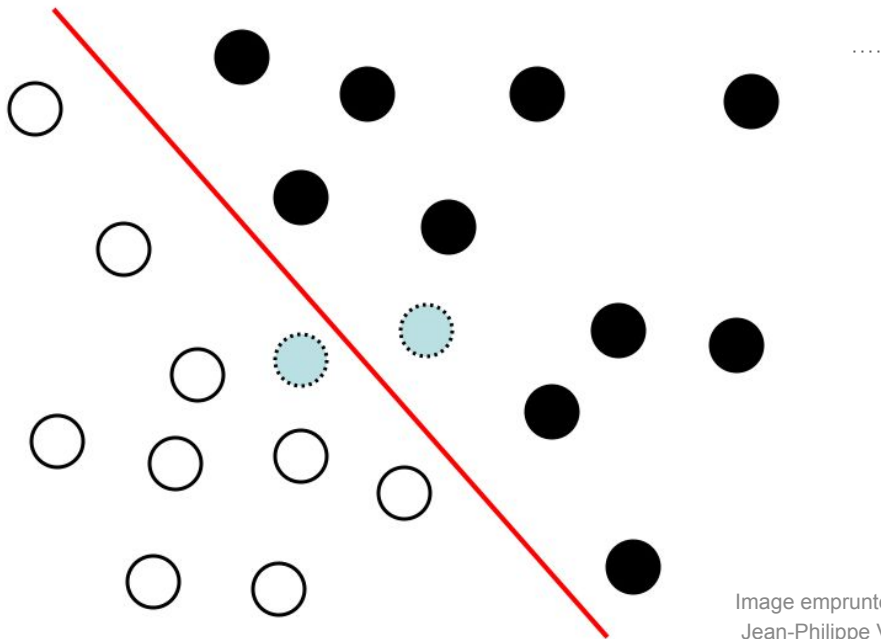


Image empruntée à
Jean-Philippe Vert

EXEMPLE 1: CLASSIFICATION BINAIRE

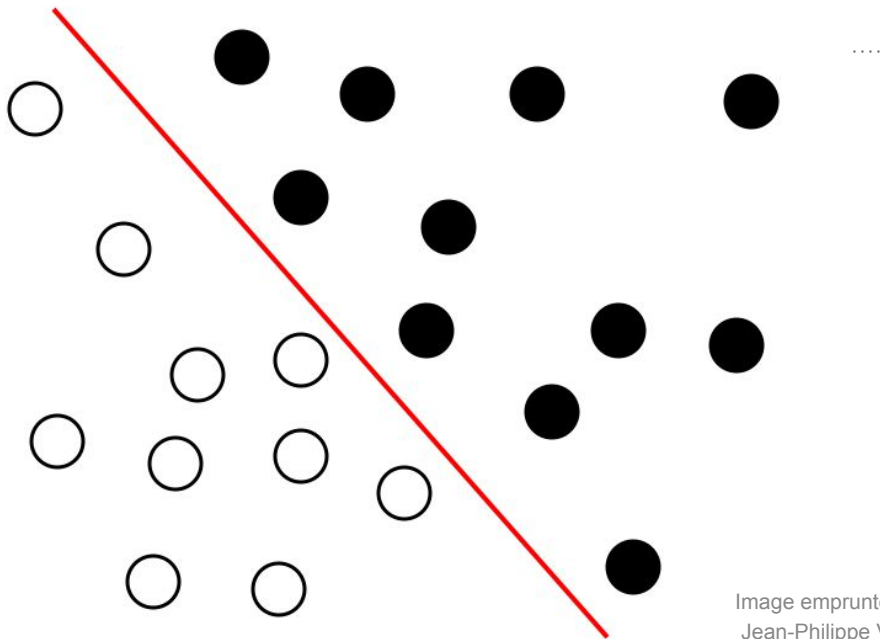
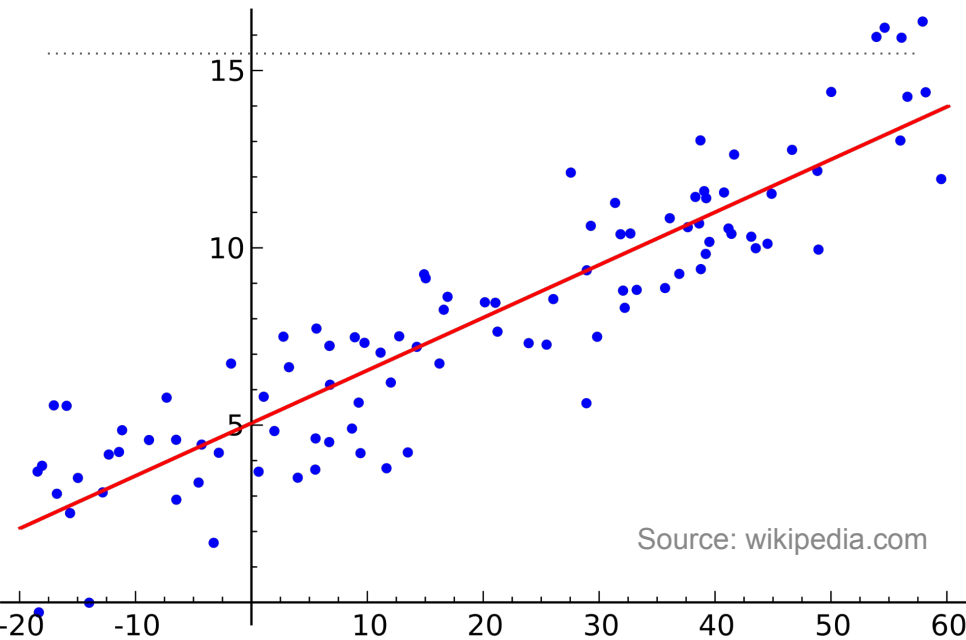


Image empruntée à
Jean-Philippe Vert

EXAMPLE 2: REGRESSION LINEAIRE



Principes

ENTREES (X) ET SORTIES (Y)

Le principe est toujours le même : **X** en entrée, **Y** en sortie. On cherche **f** tq **$Y = f(X)$** .



Exemples :

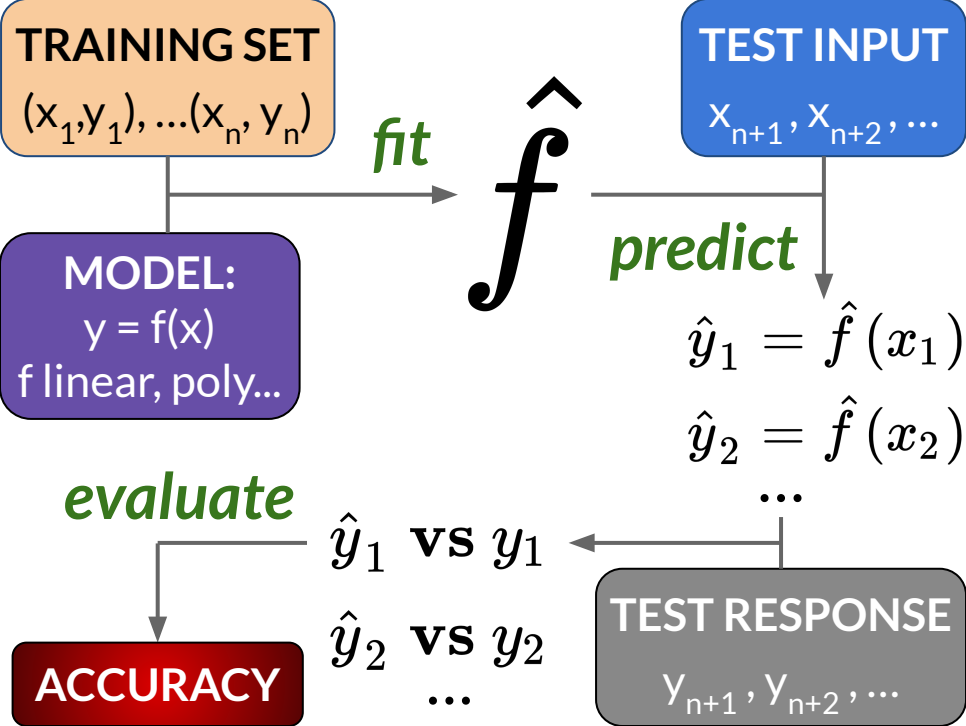
X	f	Y
emails	→	spam / non spam
historique client	→	nombre de clics
code génétique	→	état du patient
profil: age, sexe,...	→	salaire

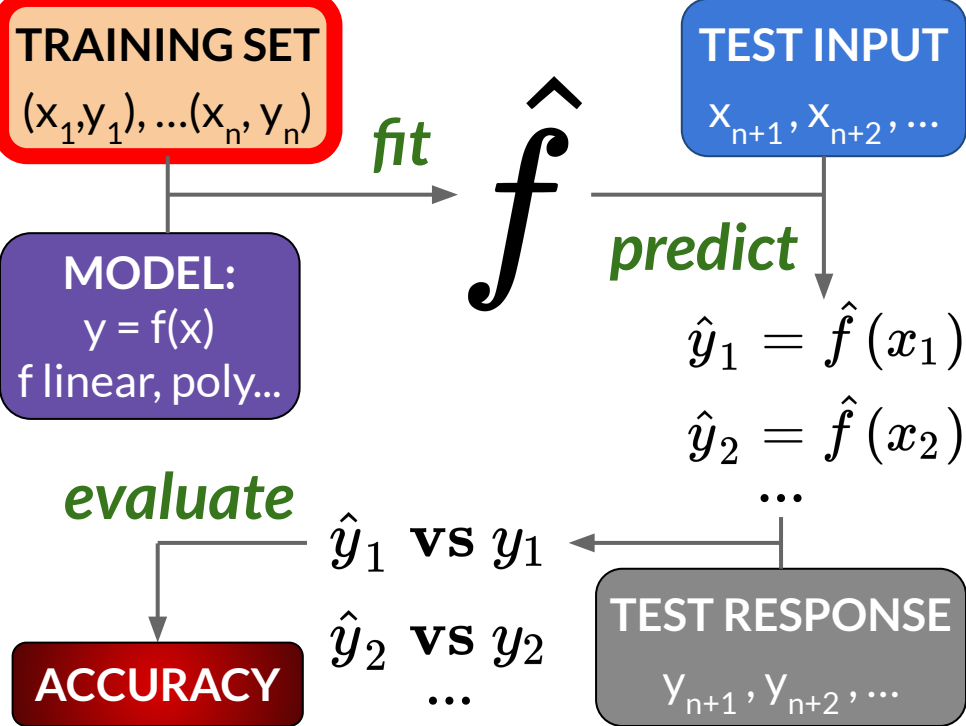
TYPES DE PROBLEMES

Régression : la réponse est un nombre réel.
Exemples : prédiction du salaire, nombre de clics, prix d'un appartement.

Classification : la réponse est une classe.
Exemples : catégorie d'un article (classification multiple), spam (classification binaire).

Attention: représenter des classes (chat, chien, rat, mouton) par des nombres (0, 1, 2, 3) ne fait **pas** de votre problème une régression!





ENSEMBLE D'APPRENTISSAGE

On entraîne le modèle sur **l'ensemble d'apprentissage** (training set).

Il est composé d'exemples de la forme (x_i, y_i) :
pour chaque exemple i , on a donc:

- la valeur d'entrée x_i : on se ramène le plus souvent à un **vecteur**, c'est-à-dire une liste de nombres... grâce à la transformation de l'input data en features (Prochain cours!)
- et la réponse y_i : ce sera un **scalaire** (un nombre)

Training set = $\{ n \text{ exemples } (x_1, y_1), \dots, (x_n, y_n) \}$

EXEMPLE

	sexe	âge	diplôme
x_1	0	30	5
x_2	1	25	2
x_3	1	53	3
\vdots	\vdots	\vdots	\vdots
x_n	0	20	0

	salaire
y_1	3000
y_2	1800
y_3	2900
\vdots	\vdots
y_n	1200

EXEMPLE

Voici un 1^{er} texte → pas un spam

Voici un 2nd texte → pas un spam

Ce doc a un texte → spam!

EXAMPLE

Voici un 1^{er} texte → pas un spam

Voici un 2nd texte → pas un spam

Ce doc a un texte → spam!

	X							
	voici	un	1 ^{er}	texte	2 nd	ce	doc	a
X ₁	1	1	1	1	0	0	0	0
X ₂	1	1	0	1	1	0	0	0
X ₃	0	1	0	1	0	1	1	1

	Y
	spam
y ₁	0
y ₂	0
y ₃	1

EXAMPLE

Voici un 1^{er} texte → pas un spam

Voici un 2nd texte → pas un spam

Ce doc a un texte → spam!

X									Y	
	voici	un	1 ^{er}	texte	2 nd	ce	doc	a	spam	
x₁	1	1	1	1	0	0	0	0	y₁	0
x₂	1	1	0	1	1	0	0	0	y₂	0
x₃	0	1	0	1	0	1	1	1	y₃	1

$$(x_1, y_1) = ([1, 1, 1, 1, 0, 0, 0, 0], 0)$$

$$(x_2, y_2) = ([1, 1, 0, 1, 1, 0, 0, 0], 0)$$

$$(x_3, y_3) = ([0, 1, 0, 1, 0, 1, 1, 1], 1)$$

EXEMPLE

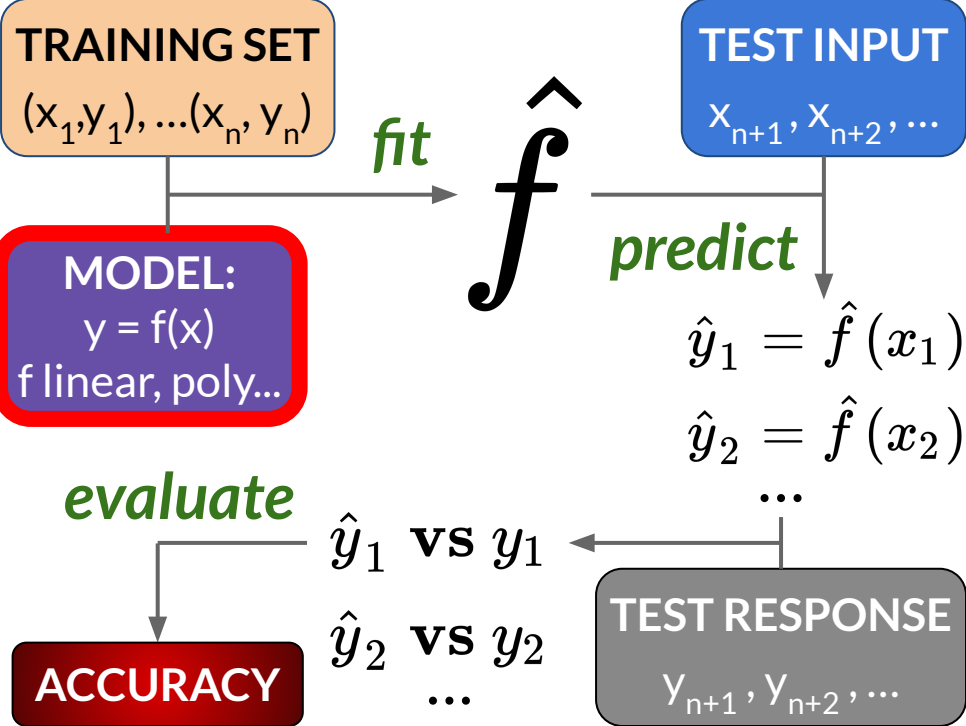
Voici un 1^{er} texte → pas un spam

Voici un 2nd texte → pas un spam

Ce doc a un texte → spam!

X									Y	
	voici	un	1 ^{er}	texte	2 nd	ce	doc	a	spam	
X₁	1	1	1	1	0	0	0	0	y₁	0
X₂	1	1	0	1	1	0	0	0	y₂	0
X₃	0	1	0	1	0	1	1	1	y₃	1

On veut apprendre à l'algorithme **ce qui fait**
que les 2 premiers messages ne sont pas des
spams, le 3ème oui, etc.



LE MODELE

C'est ici que l'on fait des **hypothèses** sur la forme de f . Par exemple:

- f est **linéaire**: $y_i = w_1 x_{i,1} + \dots + w_d x_{i,d}$
→ Il faut trouver les valeurs de $w_1, w_2, w_3 \dots$

LE MODELE

C'est ici que l'on fait des **hypothèses** sur la forme de f . Par exemple:

- f est **linéaire**: $y_i = w_1 x_{i,1} + \dots + w_d x_{i,d}$
→ Il faut trouver les valeurs de $w_1, w_2, w_3 \dots$
- f **quadratique**: $y_i = w_{1,1} x_{i,1}^2 + w_{2,2} x_{i,2}^2 + \dots$
→ À nouveau, on cherche les $w_{i,j}$ $+ w_{1,2} x_{i,1} x_{i,2} + \dots$

LE MODELE

C'est ici que l'on fait des **hypothèses** sur la forme de f . Par exemple:

- f est **linéaire**: $y_i = w_1 x_{i,1} + \dots + w_d x_{i,d}$
→ Il faut trouver les valeurs de $w_1, w_2, w_3 \dots$
- f **quadratique**: $y_i = w_{1,1} x_{i,1}^2 + w_{2,2} x_{i,2}^2 + \dots$
→ À nouveau, on cherche les $w_{i,j}$

Mais aussi et **surtout**: formulations indirectes!
(eg. Naive Bayes)

LE MODELE: CONTRAINTES ?

Soit on a une connaissance **a priori** ou une **hypothèse** pertinente sur la forme de f : alors on peut **contraindre** f , souvent sous la forme d'un **data model** (ou modèle statistique):
→ Régression Linéaire, Régression Logistique, Naive Bayes, ...

Soit on ne sait **rien** de f : on ne pose aucune contrainte. On se tournera plutôt vers des **modèles algorithmiques (cours 7)**:
→ Plus Proches Voisins, Arbres de Décision, Random Forests, SVM, Neural Nets, ...

Statistical Modeling: The Two Cultures

Leo Breiman

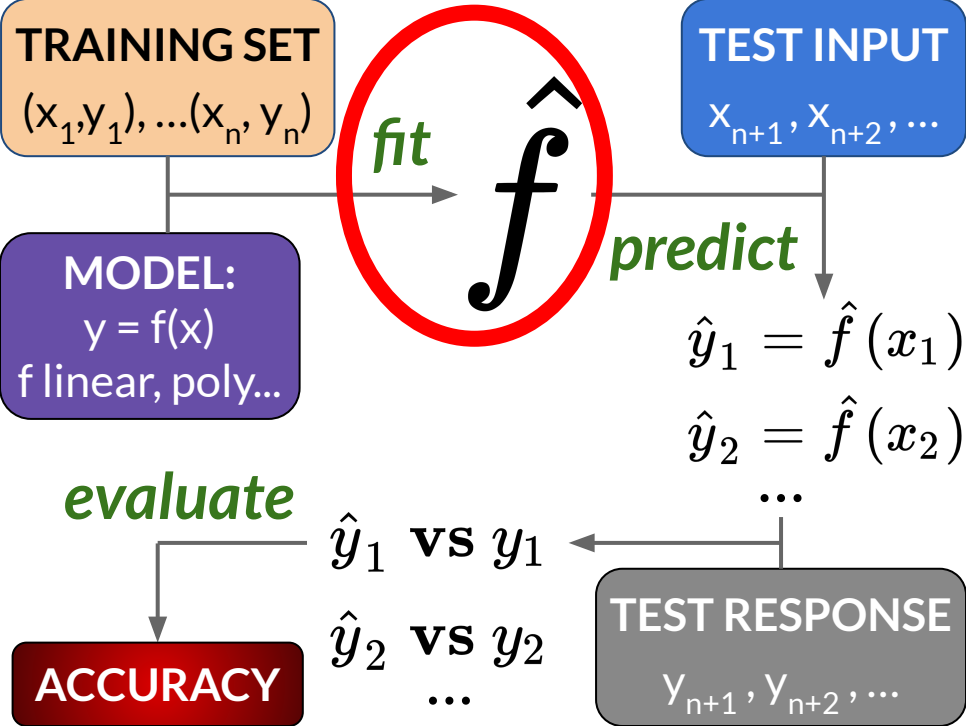
Abstract. There are two cultures in the use of statistical modeling to reach conclusions from data. One assumes that the data are generated by a given stochastic data model. The other uses algorithmic models and treats the data mechanism as unknown. The statistical community has been committed to the almost exclusive use of data models. This commitment has led to irrelevant theory, questionable conclusions, and has kept statisticians from working on a large range of interesting current problems. Algorithmic modeling, both in theory and practice, has developed rapidly in fields outside statistics. It can be used both on large complex data sets and as a more accurate and informative alternative to data modeling on smaller data sets. If our goal as a field is to use data to solve problems, then we need to move away from exclusive dependence on data models and adopt a more diverse set of tools.

DILEMME PERFORMANCE/COMPLEXITÉ

Plus le modèle est **simple**, plus il est facile de l'estimer mais moins il est proche de la réalité.

Plus le modèle est **complexe**, plus il s'approche de la réalité mais plus on risque de faire des erreurs (humaines) en l'estimant.

Dilemme complexité/performance : trouver la complexité optimale.

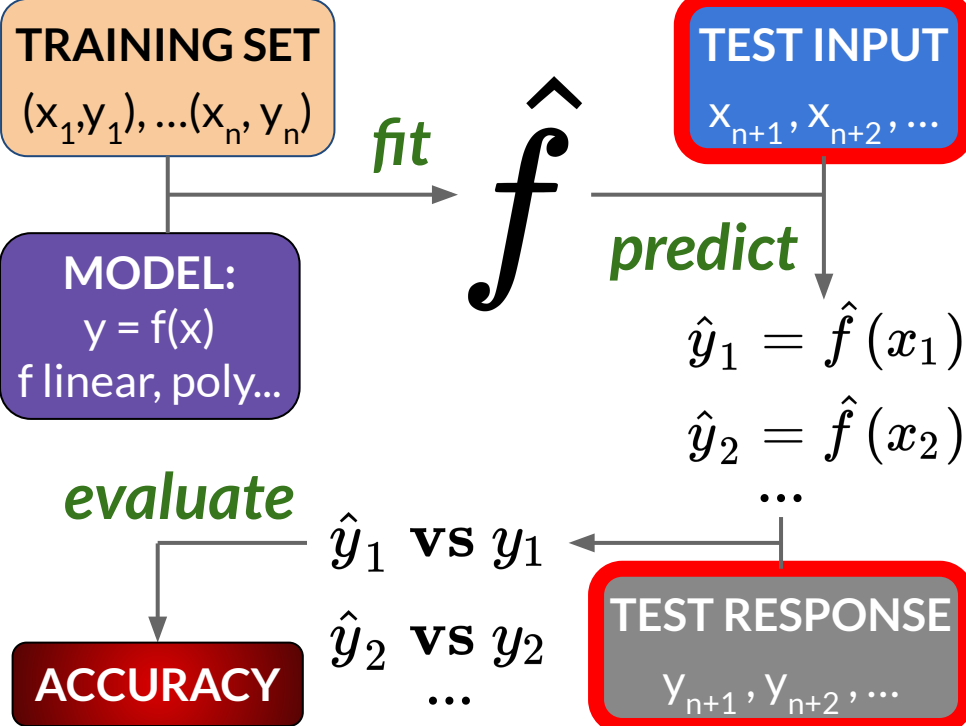


ESTIMATION

La phase **d'estimation** (fit) consiste, en fonction des hypothèses faites sur f , à **estimer la meilleure fonction f** dans le cadre des contraintes imposées.

La meilleure fonction est celle qui se **généralise** le mieux et donne les meilleures prédictions. On appelle cette fonction f .

La manière de l'estimer dépend du modèle. Nous reviendrons là-dessus plus tard.

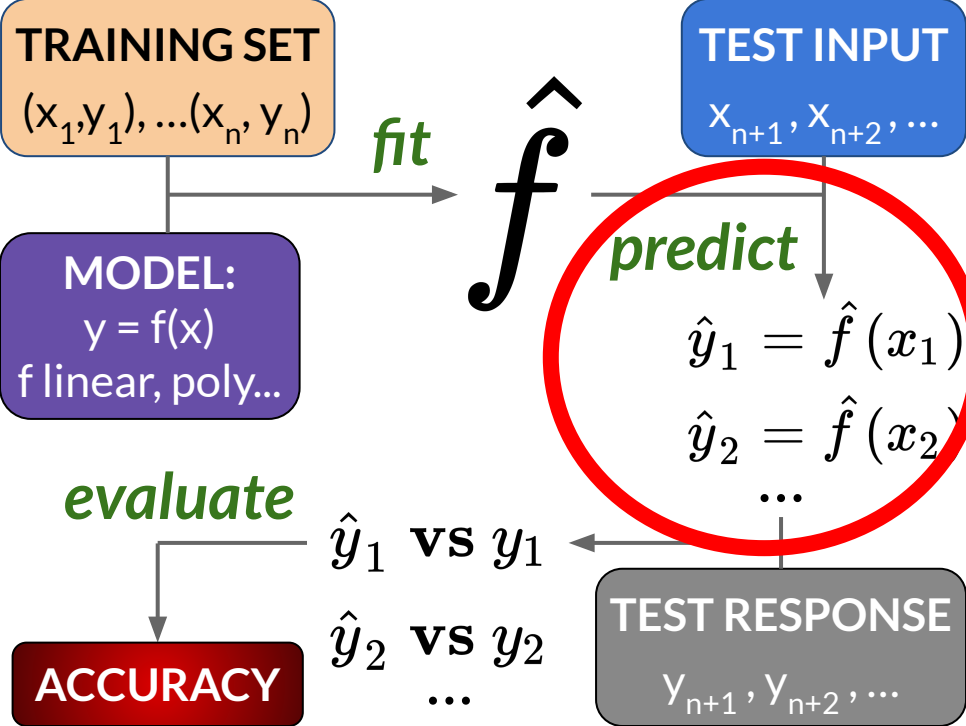


ENSEMBLE DE TEST

Comme l'ensemble d'apprentissage, l'ensemble de **test** est composé d'exemples de la forme (x_i, y_i) : pour chaque exemple i , on connaît la valeur d'entrée x_i **et la réponse** y_i .

Ce sont des exemples que l'on a **mis de côté** au départ.

Ils ne doivent **surtout pas** faire partie de l'ensemble d'entraînement!



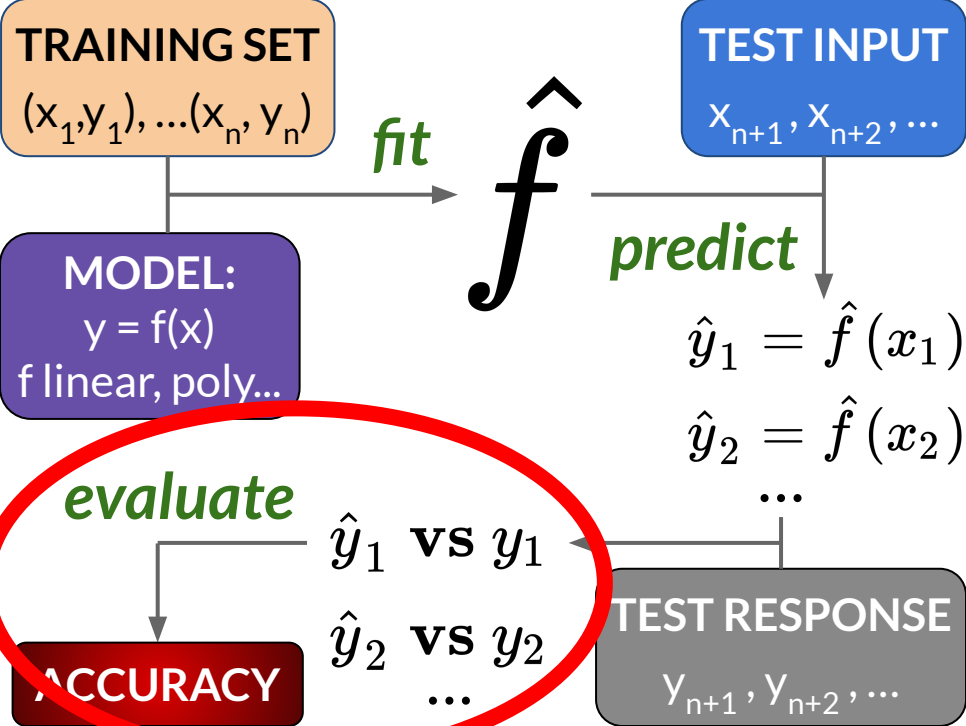
PREDICTION

Une fois le modèle estimé sur l'ensemble d'apprentissage, on l'utilise pour **prédire** les valeurs de l'ensemble de test.

Pour chaque x_i du test, on prédit la réponse \hat{y}_i avec f :

$$\hat{y}_i = f(x_i)$$

On va ensuite **comparer** le \hat{y}_i prédit avec le "vrai" y_i .



EVALUATION

On **compare** les \hat{y}_i prédits avec les "vrai" y_i .

Cette comparaison se fait rigoureusement, avec des métriques prédéfinies.

Ce n'est **surtout pas** "on vérifie sur 3 ou 4 exemples" !!

Il faut que l'ensemble de test soit assez grand pour que notre évaluation soit **très** solide.

EVALUATION

Régression : distance moyenne entre les prédictions et les vraies valeurs :

$$erreur = \frac{1}{n_{test}} \sum_{i=1}^{n_{test}} (y_i - \hat{y}_i)^2$$

EVALUATION

Régression : distance moyenne entre les prédictions et les vraies valeurs :

$$erreur = \frac{1}{n_{test}} \sum_{i=1}^{n_{test}} (y_i - \hat{y}_i)^2$$

Classification : taux d'erreur:

$$erreur = \frac{1}{n_{test}} \sum_{i=1}^{n_{test}} \delta(y_i \neq \hat{y}_i)$$

(où $\delta(A)=1$
si A vrai,
0 si faux)

EVALUATION

Régression : distance moyenne entre les prédictions et les vraies valeurs :

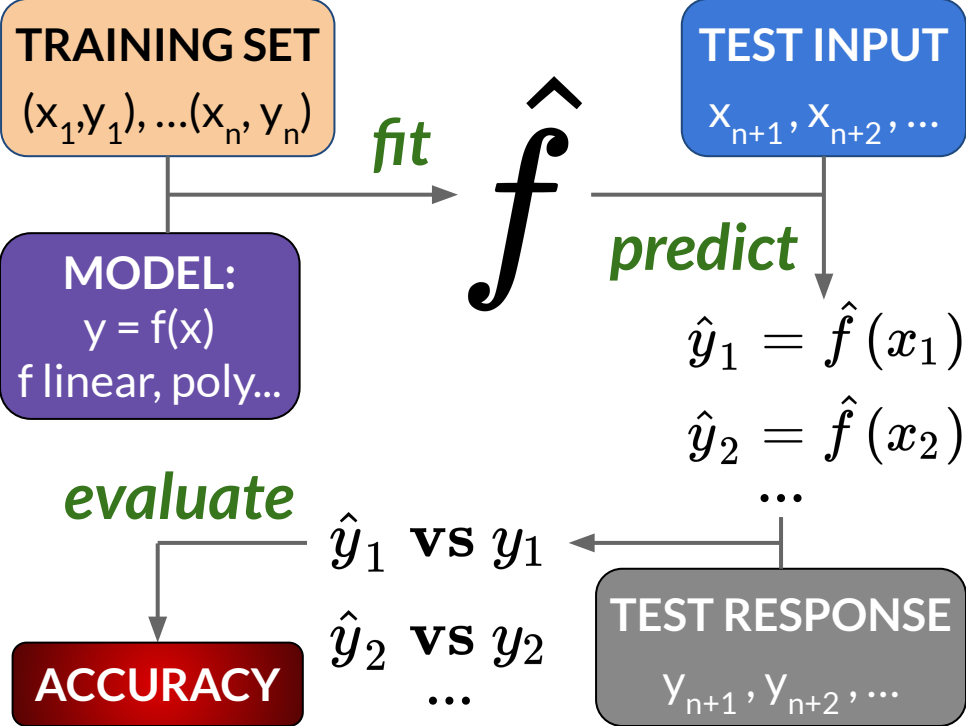
$$erreur = \frac{1}{n_{test}} \sum_{i=1}^{n_{test}} (y_i - \hat{y}_i)^2$$

Classification : taux d'erreur:

$$erreur = \frac{1}{n_{test}} \sum_{i=1}^{n_{test}} \delta(y_i \neq \hat{y}_i)$$

(où $\delta(A)=1$
si A vrai,
0 si faux)

Nous verrons plus tard des manières plus avancées pour évaluer l'algorithme.



APPRENTISSAGE: SOMMAIRE

Concept de l'**apprentissage supervisé** : faire apprendre une règle à un programme à partir d'exemples.

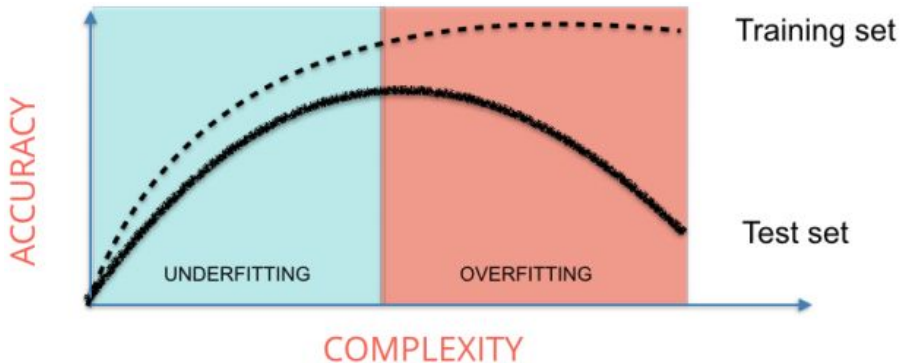
Phase d'apprentissage : on connaît l'entrée X et la sortie/réponse Y , on estime une fonction qui les lie, i.e. $Y = f(X)$

Optimisation des paramètres d'apprentissage: pendant la phase d'apprentissage, grâce à la validation croisée [plus tard!]

Phase de test : on fait prédire des valeurs à l'algorithme f et on les compare aux vraies.

SUR-APPRENTISSAGE

On parle de **sur-apprentissage** (over-fitting) lorsque l'algorithme apprend "**par cœur**" l'ensemble d'apprentissage mais n'arrive pas à **généraliser** sur l'ensemble de test.



→ **Très** important de bien **tester** le modèle!

COMMENT APPRENDRE ?

Les techniques d'apprentissage dépendent du problème.

Plusieurs techniques peuvent fonctionner.

La performance d'un algorithme dépend beaucoup de vos **données** et de leur **encodage**.

Dans ce module, nous allons voir différentes classes d'algorithmes qui ne **réfléchissent pas de la même manière**.

APPRENTISSAGE SUPERVISÉ: MODÈLES STATISTIQUES

Naive Bayes

PROBABILITES CONDITIONNELLES

Si A et B sont deux événements, la probabilité que A se produise conditionnellement au fait que B se produise se dit: **probabilité de A sachant B**, et s'écrit : $P(A|B)$. **Exemple 1:**

- A : "il pleut aujourd'hui"
- B : "il pleuvait hier"
- **$P(A|B)$: probabilité qu'il pleuve aujourd'hui sachant qu'il pleuvait hier.**
- $P(B|A)$: probabilité qu'il ait plu hier sachant qu'il pleut aujourd'hui.
- $P(A \text{ et } B) = P(A \cap B)$: **probabilité jointe** qu'il ait plu hier **et** qu'il pleuve aujourd'hui.

PROBABILITES CONDITIONNELLES

Si A et B sont deux événements, la probabilité que A se produise conditionnellement au fait que B se produise se dit: **probabilité de A sachant B**, et s'écrit : $P(A|B)$. Exemple 2:

- A : "j'achète le produit"
- B : "j'ai vu une pub pour ce produit"
- $P(A|B)$: probabilité que j'achète le produit sachant que j'ai vu une pub pour lui.
- $P(B|A)$: probabilité que j'aie vu une pub pour ce produit sachant que je l'achète.
- $P(A \text{ et } B) = P(A \cap B)$: **probabilité jointe** que j'aie vu une pub et que je l'achète.

CALCUL DE CES PROBABILITES

.....

	1	2	3	4	5	6	7	8
Pub	Oui	Non	Oui	Oui	Oui	Non	Oui	Oui
Achat	Non	Oui	Non	Oui	Non	Non	Non	Non

CALCUL DE CES PROBABILITES

	1	2	3	4	5	6	7	8
Pub	Oui	Non	Oui	Oui	Oui	Non	Oui	Oui
Achat	Non	Oui	Non	Oui	Non	Non	Non	Non

Probas simple: $P(\text{pub}) = 6/8$, $P(\text{achat}) = 2/8$

Probabilité jointe: $P(\text{pub} \cap \text{achat}) = 1/8$ ($\neq 3/16$)

Probabilités conditionnelles:

$P(\text{achat} \mid \text{pub}) =$

CALCUL DE CES PROBABILITES

∀ A et B événements : $P(A|B) = \frac{P(A \cap B)}{P(B)}$

	1	2	3	4	5	6	7	8
Pub	Oui	Non	Oui	Oui	Oui	Non	Oui	Oui
Achat	Non	Oui	Non	Oui	Non	Non	Non	Non

Probas simple: $P(\text{pub}) = 6/8$, $P(\text{achat}) = 2/8$

Probabilité jointe: $P(\text{pub} \cap \text{achat}) = 1/8$ ($\neq 3/16$)

Probabilités conditionnelles:

$P(\text{achat} | \text{pub}) =$

CALCUL DE CES PROBABILITES

∀ A et B événements : $P(A|B) = \frac{P(A \cap B)}{P(B)}$

	1	2	3	4	5	6	7	8
Pub	Oui	Non	Oui	Oui	Oui	Non	Oui	Oui
Achat	Non	Oui	Non	Oui	Non	Non	Non	Non

Probas simple: $P(\text{pub}) = 6/8$, $P(\text{achat}) = 2/8$

Probabilité jointe: $P(\text{pub} \cap \text{achat}) = 1/8$

Probabilités conditionnelles:

$$P(\text{achat} | \text{pub}) = P(\text{pub} \cap \text{achat}) / P(\text{pub}) = \frac{1/8}{6/8}$$

CALCUL DE CES PROBABILITES

∀ A et B événements : $P(A|B) = \frac{P(A \cap B)}{P(B)}$

	1	2	3	4	5	6	7	8
Pub	Oui	Non	Oui	Oui	Oui	Non	Oui	Oui
Achat	Non	Oui	Non	Oui	Non	Non	Non	Non

Probas simple: $P(\text{pub}) = 6/8$, $P(\text{achat}) = 2/8$

Probabilité jointe: $P(\text{pub} \cap \text{achat}) = 1/8$











Probabilités conditionnelles:

$P(\text{achat} | \text{pub}) = P(\text{pub} \cap \text{achat}) / P(\text{pub}) = 1/6$

$P(\text{pub} | \text{achat}) = P(\text{pub} \cap \text{achat}) / P(\text{achat}) = 1/2$

CALCUL DE CES PROBABILITES

$$P(A|B) = \frac{P(A \cap B)}{P(B)}$$

	Lundi	Mardi	Merc.	Jeudi	Vend.
Hier					
Aujourd'hui					

Probas simple: $P(\text{cloud with rain} \text{ aujourd'hui}) = 2/3$, $P(\text{cloud with rain} \text{ hier}) = 3/5$











Probabilité jointe: $P(\text{cloud with rain} \text{ hier et aujourd'hui}) = 1/5$

Probabilités conditionnelles:

$P(\text{cloud with rain} \text{ aujourd'hui} \mid \text{cloud with rain} \text{ hier}) = P(\text{cloud with rain} \text{ hier et aujourd'hui}) / P(\text{cloud with rain} \text{ hier})$

CALCUL DE CES PROBABILITES

$$P(A|B) = \frac{P(A \cap B)}{P(B)}$$

	Lundi	Mardi	Merc.	Jeudi	Vend.
Hier					
Aujourd'hui					

Probas simple: $P(\text{cloud with rain} \text{ aujourd'hui}) = 2/5$, $P(\text{cloud with rain} \text{ hier}) = 3/5$

Probabilité jointe: $P(\text{cloud with rain} \text{ hier et aujourd'hui}) = 1/5$

Probabilités conditionnelles:

$$P(\text{cloud with rain} \text{ aujourd'hui} \mid \text{cloud with rain} \text{ hier}) = (1/5) / (3/5) = 1/3$$

$$P(\text{cloud with rain} \text{ hier} \mid \text{cloud with rain} \text{ aujourd'hui}) = (1/5) / (2/5) = 1/2$$

THEOREME DE BAYES



∀ A et B événements,

$$P(A|B) = \frac{P(A \cap B)}{P(B)} = \frac{P(B|A) \times P(A)}{P(B)}$$

THEOREME DE BAYES



∀ A et B événements,

$$P(A|B) = \frac{P(A \cap B)}{P(B)} = \frac{P(B|A) \times P(A)}{P(B)}$$

Exemple:

- Dans une université, il y a 34% de femmes.
- Parmi les étudiants en informatique, 22% sont des femmes.
- 20% des étudiants de la fac sont en info.

→ **Question:** Quelle est la proportion d'étudiantes en informatique parmi les femmes de l'université ?

THEOREME DE BAYES



∀ A et B événements,

$$P(A|B) = \frac{P(A \cap B)}{P(B)} = \frac{P(B|A) \times P(A)}{P(B)}$$

Exemple:

- Dans une université, il y a 34% de femmes.
- Parmi les étudiants en informatique, 22% sont des femmes.
- 20% des étudiants de la fac sont en info.

Autrement dit: **sachant que l'étudiante est une femme, quelle est la proba qu'elle fasse de l'info?** On la note $P(\text{info} | \text{♀})$.

THEOREME DE BAYES



∀ A et B événements,

$$P(A|B) = \frac{P(A \cap B)}{P(B)} = \frac{P(B|A) \times P(A)}{P(B)}$$

Exemple:

- Dans une université, il y a 34% de femmes.
- Parmi les étudiants en informatique, 22% sont des femmes.
- 20% des étudiants de la fac sont en info.

$$P(\text{info} | \text{♀}) = \frac{P(\text{♀} | \text{info}) \times P(\text{info})}{P(\text{♀})} = \frac{0.22 \times 0.2}{0.34}$$

NAIVE BAYES

Pour chaque mot W rencontré dans l'ensemble d'apprentissage, on calcule la probabilité qu'un message soit un spam **sachant** qu'il contient le mot W :

$$P(S|W) = \frac{\overbrace{P(W|S)}^{\text{\% de spams contenant } W} \times \overbrace{P(S)}^{\text{\% de spams}}}{\underbrace{P(W)}_{\text{\% de messages contenant } W}}$$

$P(W|S) / P(W) = \text{"spamicit  "} \text{ du mot } W.$

NAIVE BAYES - TEST

On généralise à plusieurs mots:

Pour chaque message, on va calculer la
probabilité que ce message soit un spam
sachant qu'il contient les mots

$W_1, W_2, W_3 \dots W_n$:

Au tableau!

REMARQUES

Le classifieur Naive Bayes considère que **tous les mots sont indépendants** entre eux. C'est la raison pour laquelle il est **naïf**.

Pourtant, malgré cela et son âge (1960!), il reste utilisé et donne de **bons résultats**.

Il peut classer les documents dans plusieurs catégories, le cas binaire du spam étant un cas particulier.

On recommande de **supprimer des mots apparaissant trop peu ou trop souvent**, qui peuvent poser problème ou l'induire en erreur.

UNE GENERALISATION

L'Analyse discriminante linéaire (LDA) est similaire à Naive Bayes, avec la distinction fondamentale que les variables ne sont plus considérées comme indépendantes, mais comme suivant une loi normale avec des covariances non nulle au sein de chaque classe.

EN PYTHON

```
.....  
from sklearn.naive_bayes import\  
    BernoulliNB  
model = BernoulliNB()  
model.fit(Xtrain,Ytrain)  
predictions = model.predict(Xtest)
```

EVALUATION

Classification binaire ou multiple **équilibrée**

$K \geq 2$ classes \approx équi-réparties, même importance.

- Orientation politique d'un tweet: gauche / droite
 - Thème d'un article $\in \{\text{tech, santé, monde, ...}\}$
- le taux de réussite (ou d'erreur) marche bien

Classification binaire **déséquilibrée**

2 classes, mais pas du tout équivalentes. Ex:

- Mon email est-il un spam ou pas?
- **Recall** : taux des spams détectés.
- **Precision** : parmi les doc classifiés comme étant des spams, quel ratio l'était réellement?

RECALL / PRECISION

Exemple:

- Ensemble de test = 10000 SMS
 - 1000 Spam, 9000 "Ham" (des non-spams)
- Mon algo de classification prédit:
 - Sur les 1000 Spams: 930 "Spam", 70 "Ham".
 - Sur les 9000 Hams: 17 "Spam", 8983 "Ham".
- **Recall** : taux des spams détectés
$$= 930 / 1000 = 0.93 = \mathbf{93\%}$$
- **Precision** : parmi les SMS classifiés comme span, quel ratio était réellement des spams?
$$= 930 / (930 + 17) \approx 0.982 = \mathbf{98.2\%}$$

CONCLUSIONS SUR NAIVE BAYES

Avantages :

- Modèle très simple.
- Très rapide: très peu de calculs.
- Marche bien, dans certains cas.

Inconvénients

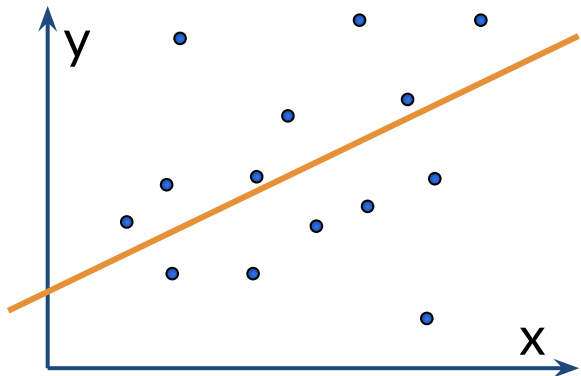
- Sa naïveté ne s'applique pas à tous les problèmes.
- Pour bien marcher, il nécessite de bien préprocesser le texte (comme tous les algorithmes, en fait)

Régression linéaire

REGRESSION LINEAIRE SIMPLE

A partir de n exemples $(x_1, y_1), \dots, (x_n, y_n)$, trouver la droite qui passe *au milieu* : $\hat{y} = \beta_0 + \beta_1 x + \varepsilon$

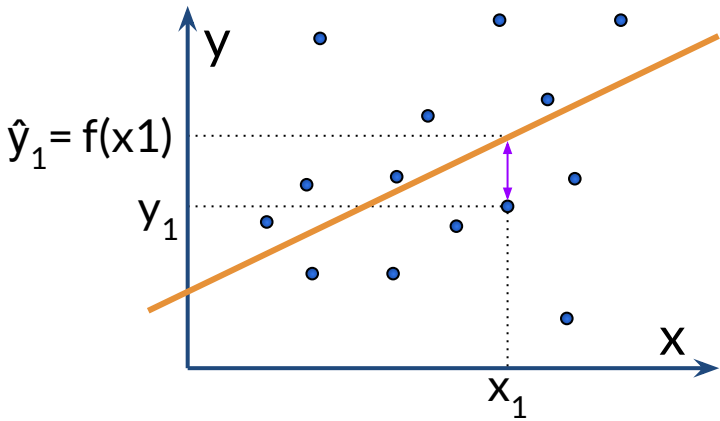
- β_0 : biais (valeur en $x = 0$)
- β_1 : pente
- ε : erreur



REGRESSION LINEAIRE SIMPLE

A partir de n exemples $(x_1, y_1), \dots, (x_n, y_n)$, trouver la droite qui passe *au milieu* : $\hat{y} = \beta_0 + \beta_1 x + \varepsilon$

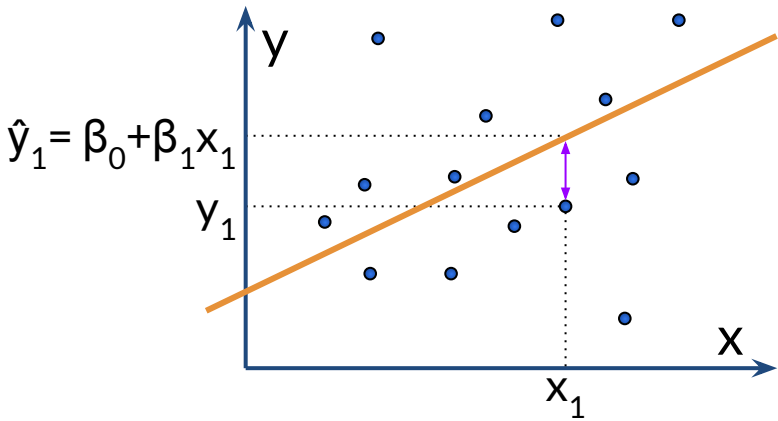
- β_0 : biais (valeur en $x = 0$)
- β_1 : pente
- ε : erreur



REGRESSION LINEAIRE SIMPLE

Objectif : minimiser la **variance résiduelle** :

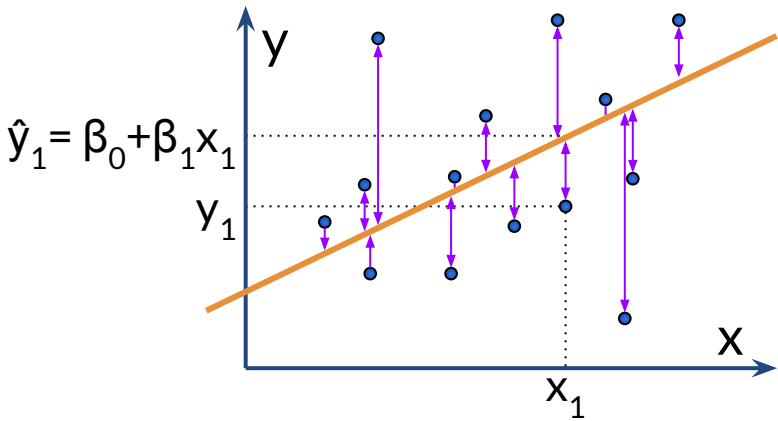
$$\min \sum_{i=1}^n (y_i - \hat{y}_i)^2 \Leftrightarrow \min_{\beta_0, \beta_1} \sum_{i=1}^n (y_i - (\beta_0 + \beta_1 x_i))^2$$



REGRESSION LINEAIRE SIMPLE

Objectif : minimiser la **variance résiduelle** :

$$\min \sum_{i=1}^n (y_i - \hat{y}_i)^2 \Leftrightarrow \min_{\beta_0, \beta_1} \sum_{i=1}^n (y_i - (\beta_0 + \beta_1 x_i))^2$$



REGRESSION LINEAIRE MULTIPLE

On généralise: au lieu d'une seule variable réelle x on peut en avoir d (la dimension):

$$\hat{y} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_d x_d$$

→ en notation vectorielle: $\hat{y} = \beta_0 + \vec{\beta} \cdot \vec{X}$

Et la regression devient:

$$(\hat{\beta}_0, \hat{\vec{\beta}}) = \underbrace{\arg \min_{\beta_0 \in \mathbb{R}, \vec{\beta} \in \mathbb{R}^d}}_{\text{...la fonction de coût}}$$

Trouver (β_0, β) qui minimise...

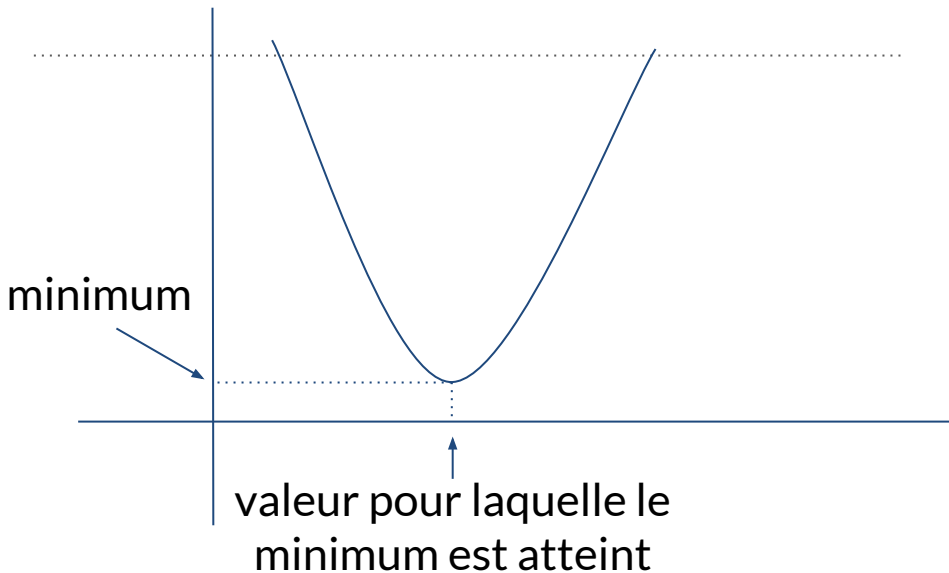
TROUVER LA SOLUTION

$$(\hat{\beta}_0, \hat{\vec{\beta}}) = \underbrace{\arg \min_{\beta_0 \in \mathbb{R}, \vec{\beta} \in \mathbb{R}^d}}_{\text{...la fonction de coût}}$$

Trouver (β_0, β) qui minimise...

Pour un ensemble donné (même grand) de (X_i, y_i) , c'est une fonction **convexe** de $d+1$ variables ($\beta_0 = 1$ var, $\beta = d$ var), et **dérivable**.

TROUVER LA SOLUTION



Le minimum est atteint lorsque la dérivée vaut 0

TROUVER LA SOLUTION

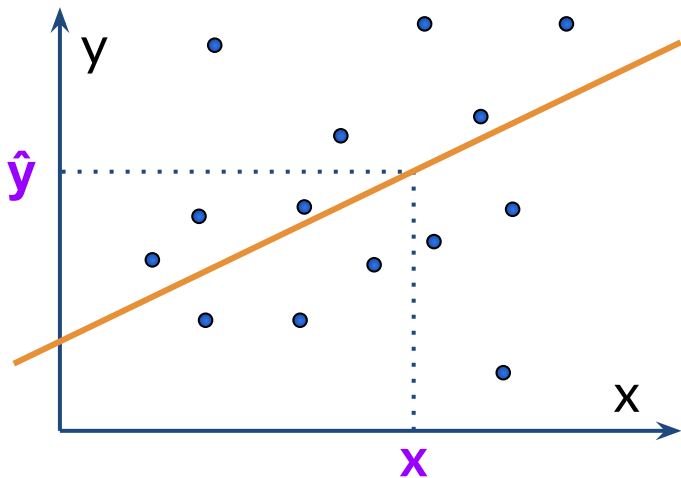
Il suffit donc de dériver la fonction par rapport aux β et de trouver la solution de l'équation dérivée = 0.

Pour la régression linéaire, on est chanceux: cette solution a une forme fermée, c'est-à-dire qu'on peut trouver la solution de cette équation à la main. Elle s'exprime de manière matricielle (pas besoin de retenir cette formule!):

$$\hat{\beta} = (X^T X)^{-1} X^T y$$

PHASE DE TEST

Quand un **nouveau** point **x** est donné, on peut **prédire** le **\hat{y}** correspondant:



INTERPRETATION

Supposons un modèle :

$$\text{Salaire} = \beta_0 + \beta_1 \times \text{expérience} + \beta_2 \times \text{études} + \varepsilon$$

Et supposons que les résultats soient :

$$\beta_0 = 900, \beta_1 = 100, \beta_2 = 200$$

On interprète que :

- Quelqu'un qui n'a ni études ni expérience touchera 900 euros en moyenne.
- Une année d'expérience supplémentaire rapporte en moyenne 100 euros de plus.
- Une année d'études supplémentaire rapporte en moyenne 200 euros de plus.

EVALUATION

$$erreur = \frac{1}{n_{test}} \sum_{i=1}^{n_{test}} (y_i - \hat{y}_i)^2$$

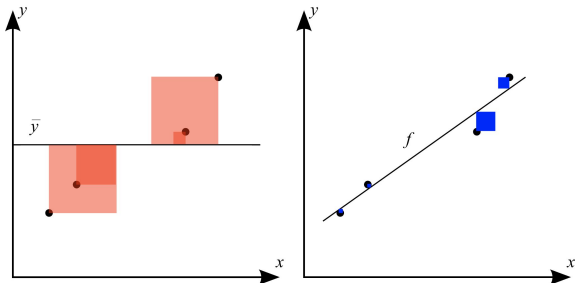
EVALUATION : le R^2

“C’est quoi le R^2 de ta régression linéaire?”

$$R^2 = 1 - \frac{\sum (\hat{y} - y)^2}{\sum (y - \bar{y})^2}$$

← Erreur quadratique

← Variance



EVALUATION : le R^2

“C’est quoi le R^2 de ta régression linéaire?”

$$R^2 = 1 - \frac{\sum (\hat{y} - y)^2}{\sum (y - \bar{y})^2}$$

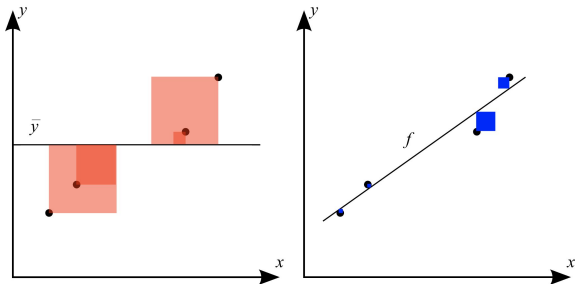
← Erreur quadratique

← Variance

$R^2 \in [0,1]$:

$R^2 = 0 \Rightarrow$ pas
mieux que
“Moyenne”.

$R^2 = 1 \Rightarrow$
Parfait



EVALUATION : le R^2

“C’est quoi le R^2 de ta régression linéaire?”

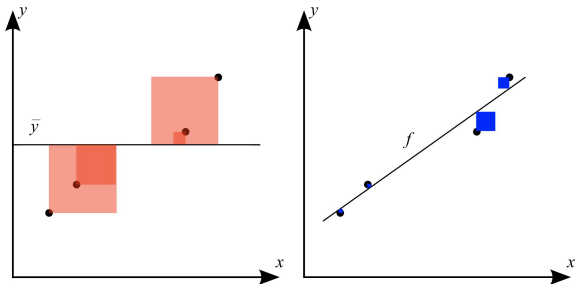
$$R^2 = 1 - \frac{\sum (\hat{y} - y)^2}{\sum (y - \bar{y})^2}$$

← Erreur (quadratique)
← Variance

$R^2 \in [0,1]$:

$R^2 = 0 \Rightarrow$ pas
mieux que
“Moyenne”.

$R^2 = 1 \Rightarrow$
Parfait



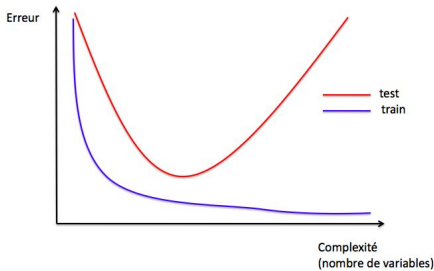
R^2 a plein de problèmes



LIMITES

Le modèle linéaire est séduisant par sa simplicité mais vite limité :

- Ne marche **pas** si **# variables > # observations**.
- Si des variables sont **corrélées**, cela peut nuire à l'interprétation, ex: jours vacance/experience
- Plus on ajoute de variables, plus le modèle est **instable** et risque le **sur-apprentissage**.



PENALISER LA COMPLEXITE

Un modèle linéaire avec de nombreuses variables peut être quasi parfait sur l'ensemble d'apprentissage. Mais il sera alors **mauvais** sur de nouvelles données (e.g. l'ensemble test).

Une solution : **pénaliser** la complexité du modèle en forçant les poids à se comporter d'une certaine manière. Par exemple:

- la régression **Ridge** force les variables corrélées à avoir des poids similaires;
- la régression **Lasso** limite le nombre de poids non-nuls (→ le nb de variables "qui comptent").

EN PYTHON

```
from sklearn.linear_model import \
    LinearRegression
model = LinearRegression()
model.fit(Xtrain,Ytrain)
predictions = model.predict(Xtest)
```

CONCLUSION SUR LA REGRESSION LINÉAIRE

Avantages :

- Modèle simple, facile à estimer, solution “exacte”.
- Interprétable.
- Adapté pour les problèmes simples.

Inconvénients :

- Impossible en très grande dimension
- En grande dimension, sur-apprentissage fréquent (mitigé par Ridge / Lasso)
- On se trouve rarement dans des situations où ce modèle est le meilleur.