APPRENTISSAGE SUPERVISÉ
DEUXIÈME PARTIE

RAPPEL

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 cherche la fonction qui les lie Y = f (X)

Optimisation des paramètres : pendant la phase d'apprentissage, grâce à la validation croisée

Phase de test : on fait prédire des valeurs à l'algorithme et on les compare aux vraies valeurs pour estimer sa performance.

Trois types de modèles : algorithmiques (semaine dernière), probabilistes (cette semaine) et boîtes noires (semaine prochaine).

Naive Bayes

PROBABILITÉS 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** 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, B) = P(A ∩ B) : **probabilité jointe** qu'il ait plu hier **et** qu'il pleuve aujourd'hui.

PROBABILITÉS 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** s'écrit : P(A|B).

Exemple 2:

A : "j'achète"

B : "j'ai vu une pub"

P(A|B) : probabilité que j'achète sachant que j'ai vu une pub .

P(B|A) : probabilité que j'aie vu une pub sachant que j'ai acheté.

P(A, B) = P(A ∩ B) : **probabilité jointe** que j'aie vu une pub **et** que j'aie acheté.

CALCUL DE CES PROBABILITÉS

Si A et B sont deux événements, alors:

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

Jour	Lundi	Mardi	Mercredi	Jeudi	Vendredi
Hier	Pluie	Pluie	Soleil	Soleil	Pluie
Aujourd'hui	Pluie	Soleil	Soleil	Pluie	Soleil

Probabilités simples : $P(\text{pluie aujourd'hui}) = 2/5,\ P(\text{pluie hier}) = 3/5$

Probabilité jointe : $P(\text{pluie aujourd'hui} \cap \text{pluie hier}) = 1/5$

Probabilités conditionnelles :

$$P(\text{pluie auj}|\text{pluie hier}) = \frac{P(\text{pluie auj} \cap \text{pluie hier})}{P(\text{pluie hier})} = \frac{1/5}{3/5} = \frac{1}{3}$$

$$P(\text{pluie hier}|\text{pluie auj}) = \frac{P(\text{pluie hier} \cap \text{pluie auj})}{P(\text{pluie auj})} = \frac{1/5}{2/5} = \frac{1}{2}$$

CALCUL DE CES PROBABILITÉS

Si A et B sont deux événements, alors:

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

Obs	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

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

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

Probabilités conditionnelles :

$$P(\text{pub}|\text{achat}) = \frac{P(\text{pub} \cap \text{achat})}{P(\text{achat})} = \frac{1/8}{2/8} = \frac{1}{2}$$

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

THEOREME DE BAYES

Si A et B sont deux événements:

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

Dans une université, il y a 34% de femmes.

Parmi les étudiants en informatique, 22% sont des femmes.
Les étudiants en informatique représentent 20% de l'université.

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

THEOREME DE BAYES

Si A et B sont deux événements:

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

Dans une université, il y a 34% de femmes.

Parmi les étudiants en informatique, 22% sont des femmes.
Les étudiants en informatique représentent 20% de l'université.

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

On cherche : $P(\text{étudier l'informatique}|\text{être une femme})$

THEOREME DE BAYES

Si A et B sont deux événements:

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

Dans une université, il y a 34% de femmes.

Parmi les étudiants en informatique, 22% sont des femmes.
Les étudiants en informatique représentent 20% de l'université.

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

On cherche : $P(\text{étudier l'informatique}|\text{être une femme})$

Solution:

$$P(\text{info}|\text{femme}) = \frac{P(\text{femme}|\text{info}) \times P(\text{info})}{P(\text{femme})} = \frac{0,22 \times 0,2}{0,34} = 12,9\%$$

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 (**spamicité** du mot W):

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

NAIVE BAYES - TEST

Pour un message de l'ensemble de test, on calcule la **probabilité que le message soit un spam** sachant qu'il contient les mots $W_1, W_2, W_3 \ldots W_n$:

$$P(S|W_1, W_2, ..., W_n) = \frac{\overbrace{p_1 p_2 \ldots p_n}^{\text{spamicité des mots du message}}}{\underbrace{p_1 p_2 \ldots p_n}_{\text{spamicité}} + \underbrace{(1-p_1)(1-p_2)\ldots(1-p_n)}_{\text{non-spamicité}}}$$

avec $p_1 = P(S|W_1)$, $p_2 = P(S|W_2)$, . . . , $p_n = P(S|W_n)$

REMARQUES

Le classifieur Naive Bayes considère que **tous les mots sont indépendants**. Autrement dit, il regarde l'effet des mots un par un, sans se soucier des autres. C'est la raison pour laquelle il est **naïf**.
Et pourtant ! Existant depuis longtemps (1996) et malgré son caractère naïf, il reste très utilisé, en particulier pour la classification du spam et donne de **bons résultats**.
Il est recommandé de **supprimer les mots apparaissant trop peu ou trop souvent** qui peuvent poser problème ou l'induire en erreur.
Il peut classifier les documents dans plusieurs catégories, le cas binaire du spam étant un cas particulier.

GÉNÉRALISATION

L’Analyse discriminante linéaire (LDA) est similaire à Naïve 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)
```

CONCLUSIONS SUR NAIVE BAYES

Avantages :

Modèle très simple.

Rapide car très peu de calculs.

Efficace dans certains cas.

Inconvénients :

Sa naïveté ne s'applique pas à tous les problèmes.

Il ne dispense pas (au contraire!) de préprocesser le texte (comme tous les algorithmes).

Régression linéaire

RÉGRESSION LINÉAIRE SIMPLE

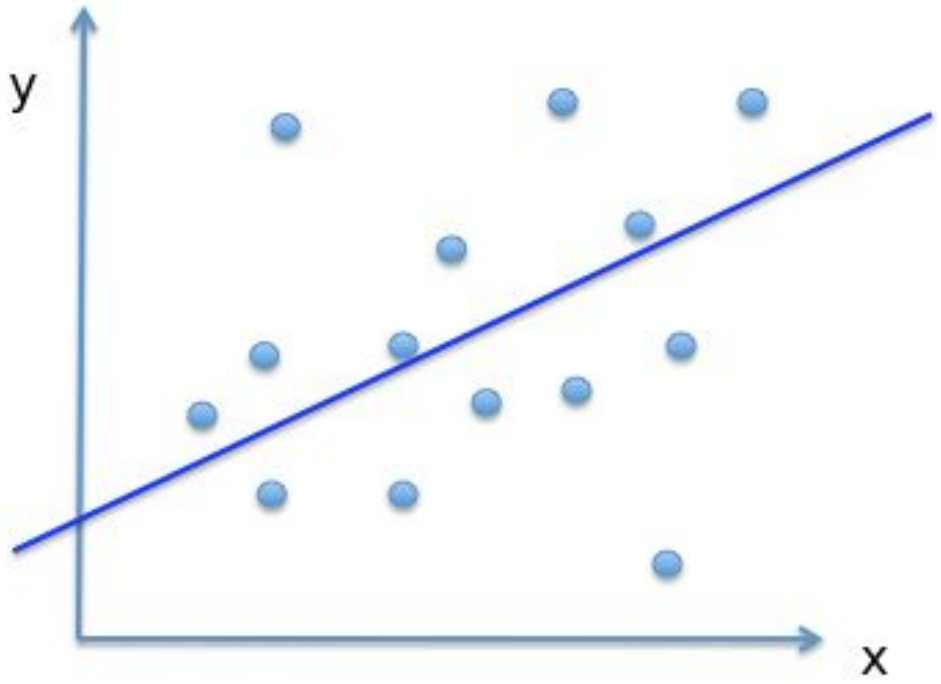
A partir de n exemples, trouver la droite qui passe en leur "centre".

β_0 : biais (valeur en x = 0)

β_1 : pente

réponse = β_0 + β_1 × (variable dépendante) + bruit

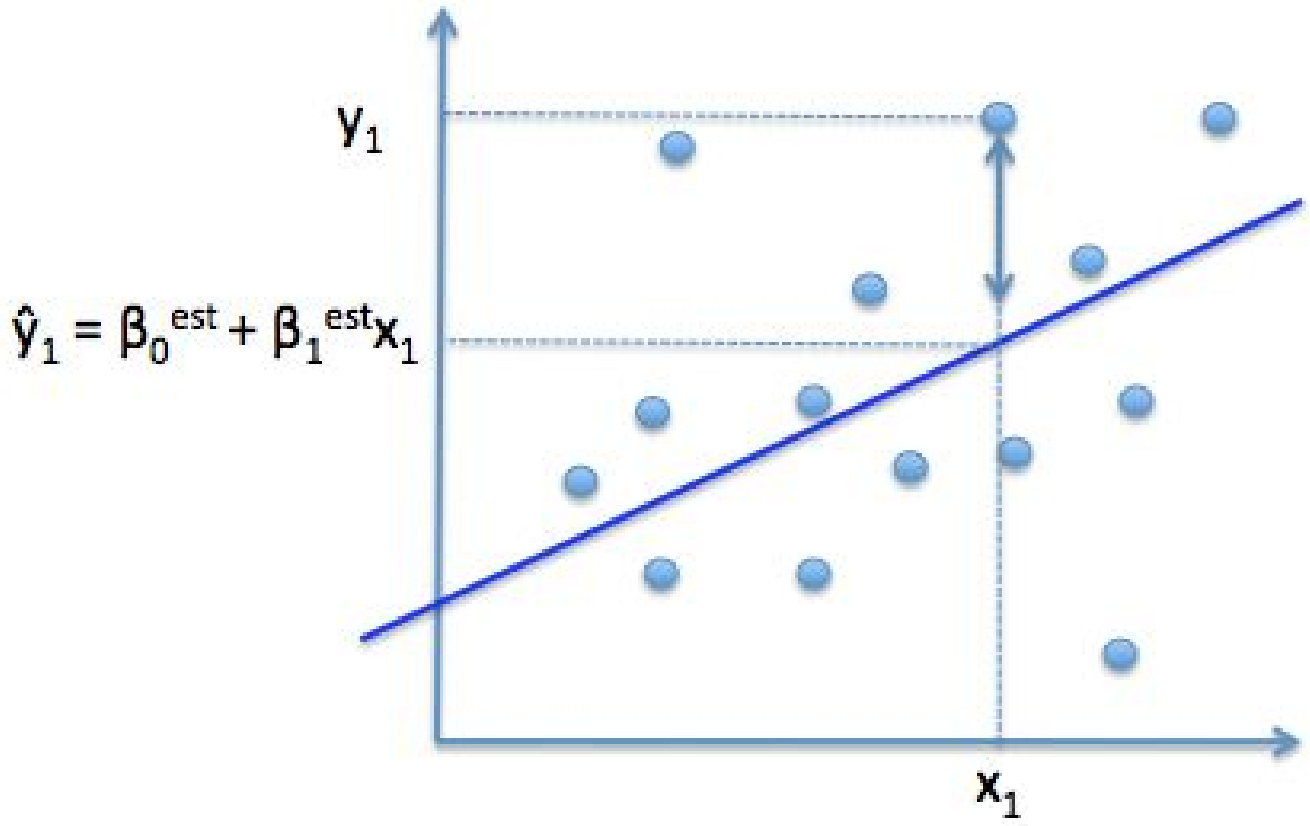
$Y = \beta_0 + \beta_1 x + \varepsilon$



ESTIMATION

Objectif : minimiser la **variance résiduelle** (i.e. ε) trouver la meilleure droite :

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



REGRESSION LINEAIRE MULTIPLE

En ajoutant de nouvelles variables :

$$\text{réponse} = \beta_0 + \beta_1 \times \text{variable}_1 + \beta_2 \times \text{variable}_2 + \cdots + \text{bruit}$$

$$Y = \beta_0 + \beta_1 \times x_1 + \beta_2 \times x_2 + \cdots + \varepsilon$$

$$Y = \beta_0 + \sum_{j=1}^{p} \beta_j x_j + \varepsilon = X\beta + \varepsilon$$

Solution :

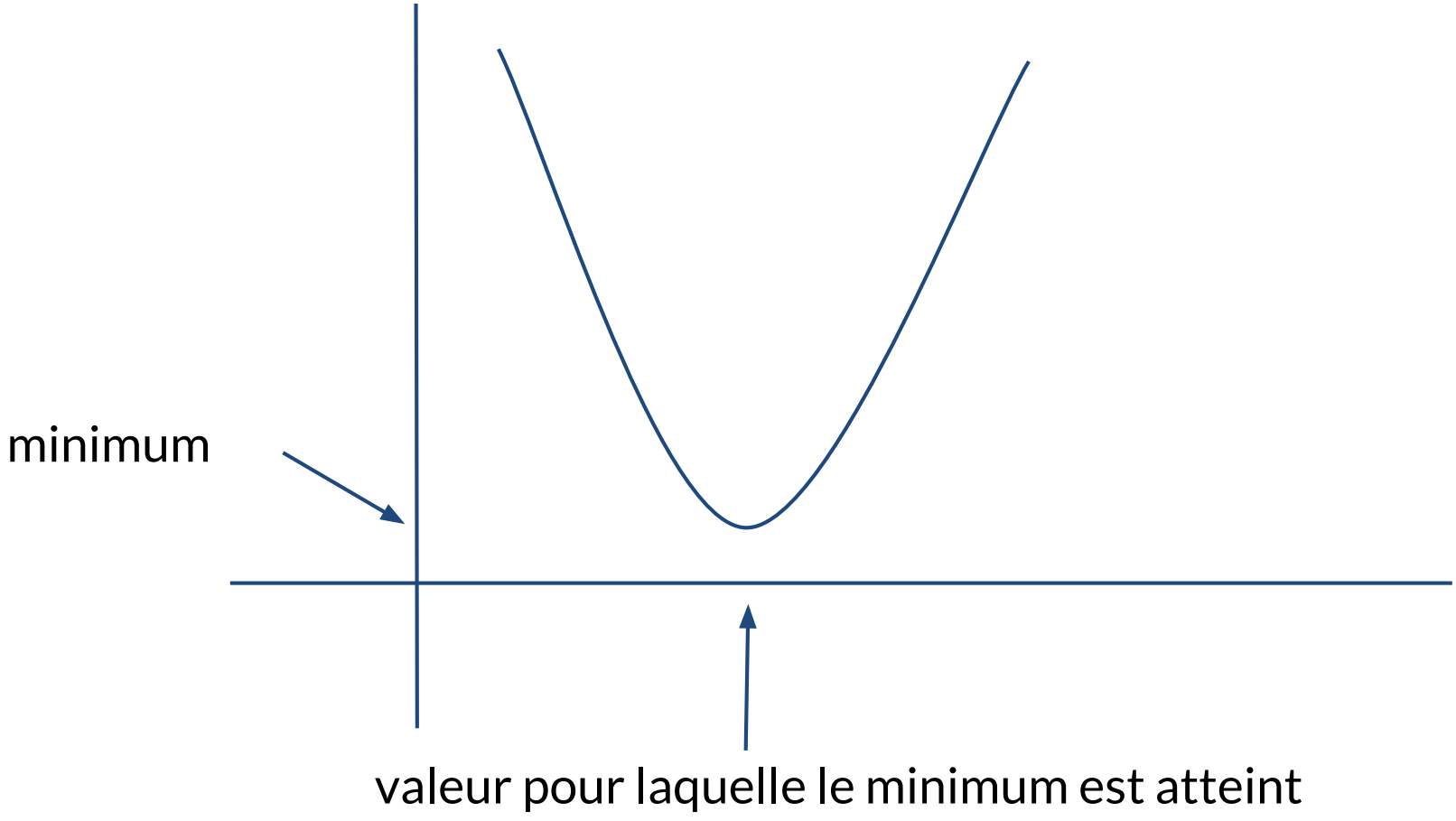
$$(\hat{\beta}_0, \hat{\beta}) = \underbrace{\arg \min_{\beta_0 \in \mathbf{R}, \beta \in \mathbf{R}^p}}_{\text{Trouver les valeurs qui minimisent}} \underbrace{\sum_{i=1}^{n} (y_i - (\beta_0 + \beta x_i))^2}_{\text{la fonction de coût}}$$

TROUVER LA SOLUTION

On cherche les poids qui minimisent cette fonction qui est convexe et dérivable.

$$(\hat{\beta}_0, \hat{\beta}) = \underbrace{\arg \min_{\beta_0 \in \mathbf{R}, \beta \in \mathbf{R}^p}}_{\text{Trouver les valeurs qui minimisent}} \underbrace{\sum_{i=1}^{n} (y_i - (\beta_0 + \beta x_i))^2}_{\text{la fonction de coût}}$$

TROUVER LA SOLUTION



Le minimum est atteint lorsque la dérivée vaut 0.

TROUVER LA SOLUTION

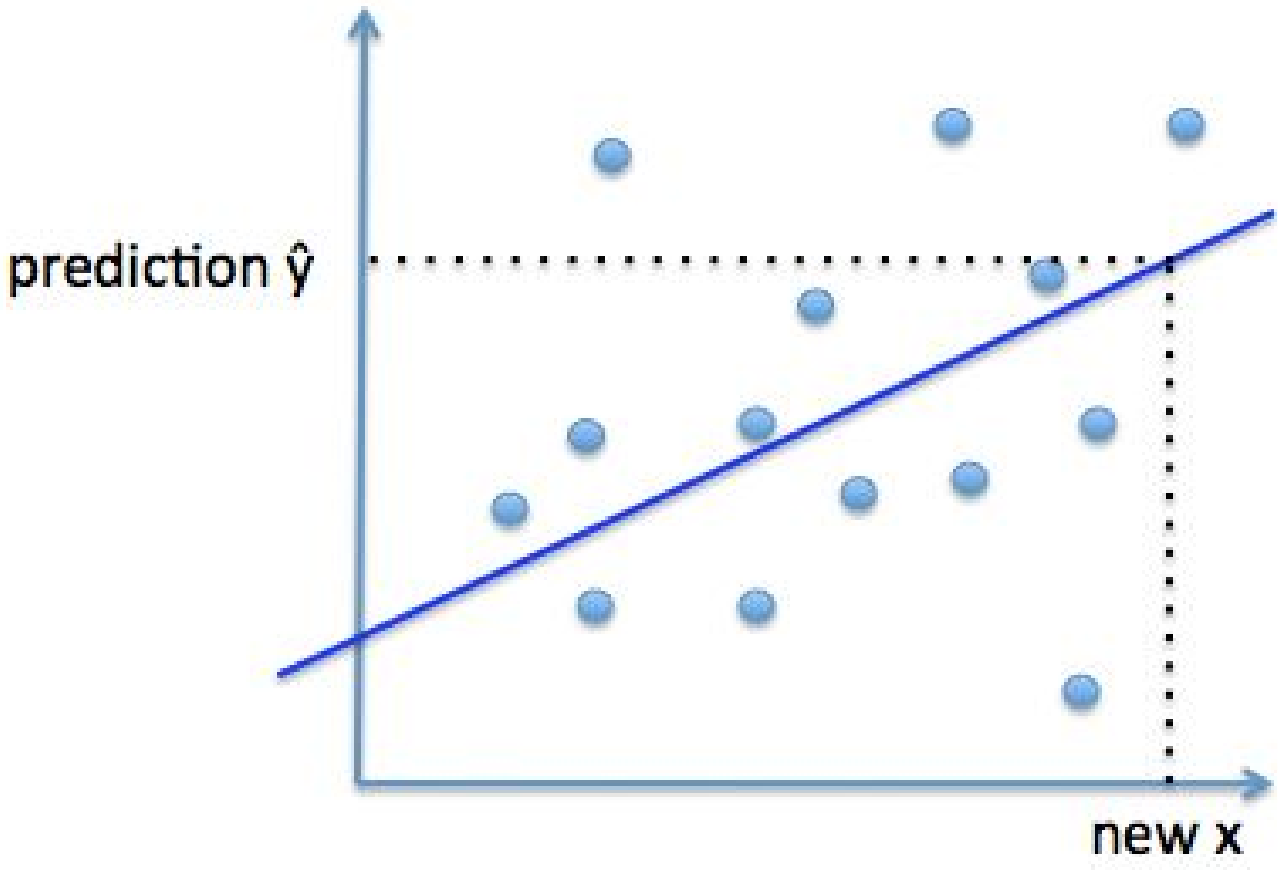
Il suffit donc de dériver la fonction par rapport aux β et de trouver la solution de l'équation dérivée = 0.

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 ŷ correspondant:



INTERPRÉTATION

Supposons un modèle :

$$\text{salaire} = \beta_0 + \beta_1 \times \text{expérience} + \beta_2 \times \text{études} + \text{bruit}$$

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.
Quelqu'un qui a un bac+5 et 5 ans d'expérience touchera 2400 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.

ÉVALUATION

On cherche à expliquer la **variance totale** (total sum of squares) :

$$TSS = \sum_{i=1}^{n} (y_i - \overline{y})^2$$

On calcule la **variance expliquée** (explained sum of squares) :

$$ESS = \sum_{i=1}^{n} (\hat{y}_i - \overline{y})^2$$

On veut minimiser la **variance résiduelle** (residual sum of squares) :

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

Et on a : $TSS = ESS + RSS$

Le **pourcentage de variance expliquée** donne donc une indication sur la qualité de la régression qu'on appelle R^2 (à calculer également sur l'ensemble de test) :

$$R^2 = \frac{ESS}{TSS} = 1 - \frac{RSS}{TSS}$$

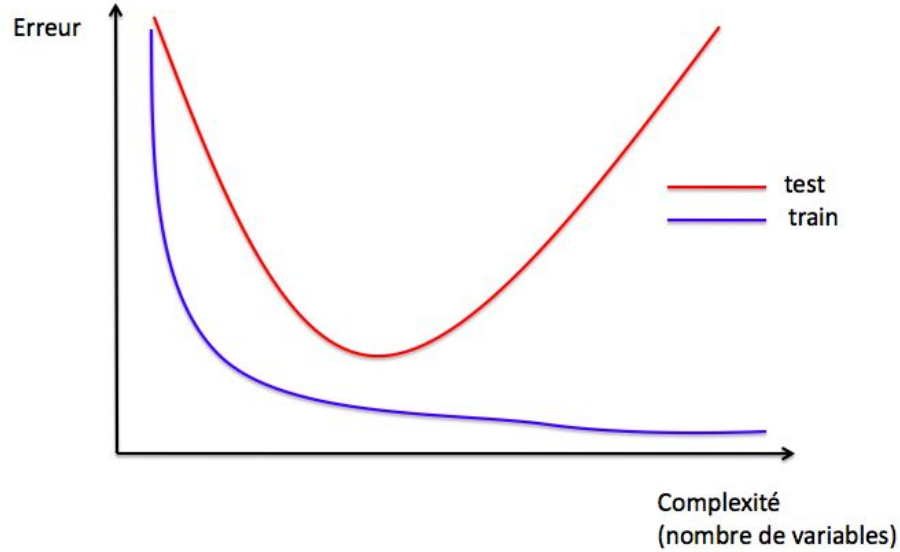
Le R^2 se trouve entre 0 et 1 :

- 0 : rien n'est expliqué par les variables
- 1 : tout est expliqué par les variables

LIMITES

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

- La solution demande l'inversion de la matrice X^TX qui n'est pas inversible si l'on a **plus de variables que d'observations**.
- Si des variables sont **corrélées**, cela va perturber le modèle : deux variables très ressemblantes n'auront pas forcément les mêmes poids.
- Plus on ajoute de variables, plus le modèle devient **instable** et on risque donc le **sur-apprentissage**.



PÉNALISATION

Un modèle linéaire avec de nombreuses variables peut être parfait sur l'ensemble d'apprentissage. Mais il risque d'être **mauvais** sur de nouvelles données.

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 similaires à avoir des poids similaires
- la régression Lasso force certains poids à être nuls (les variables associées ne comptent pas)

EN PYTHON

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

CONCLUSION SUR LA RÉGRESSION LINÉAIRE

Avantages :

Modèle simple et facile à estimer, solution “exacte”.

Interprétable.

Adapté pour les problèmes simples.

Inconvénients :

Impossible en grande dimension.

On se trouve rarement dans des situations où ce modèle est pertinent.

Régression logistique

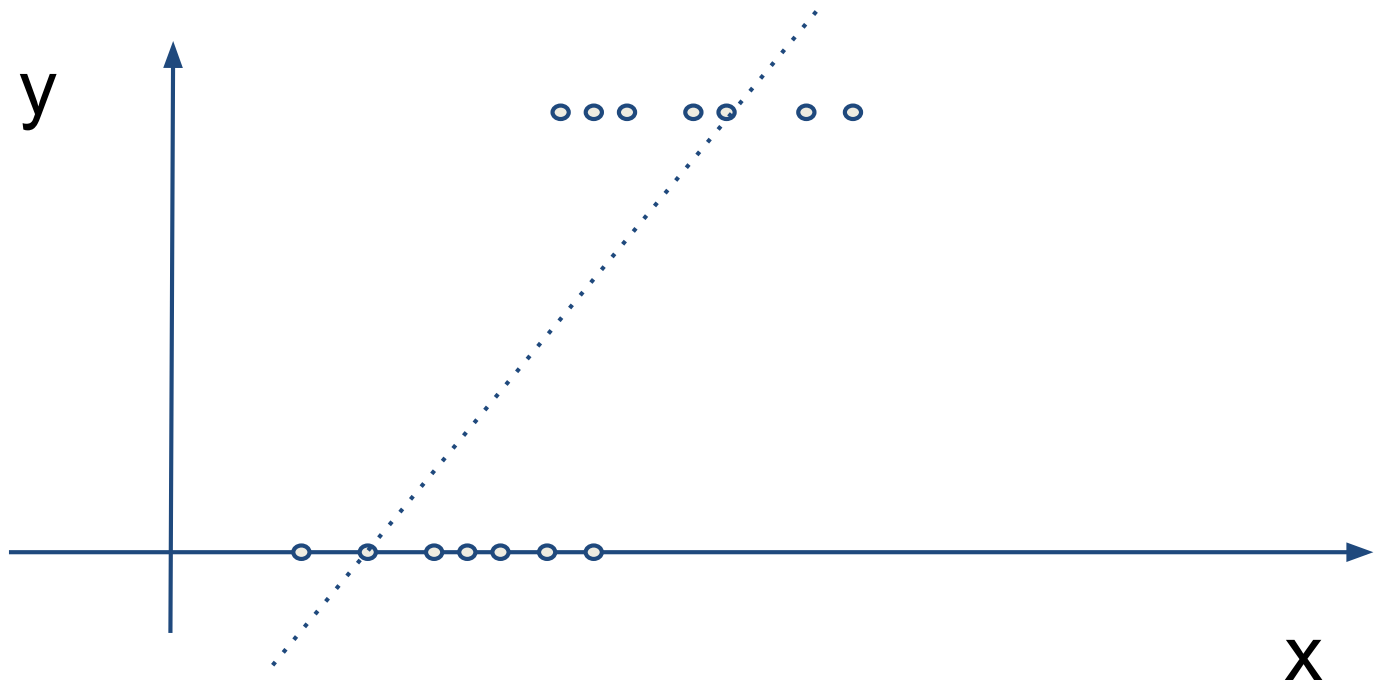
RÉGRESSION LOGISTIQUE BINAIRE

Dans la régression linéaire, on prédit une variable Y continue
La régression logistique permet en fait de faire de la **classification**.

On considère ici la classification binaire : $Y \in \{0, 1\}$.

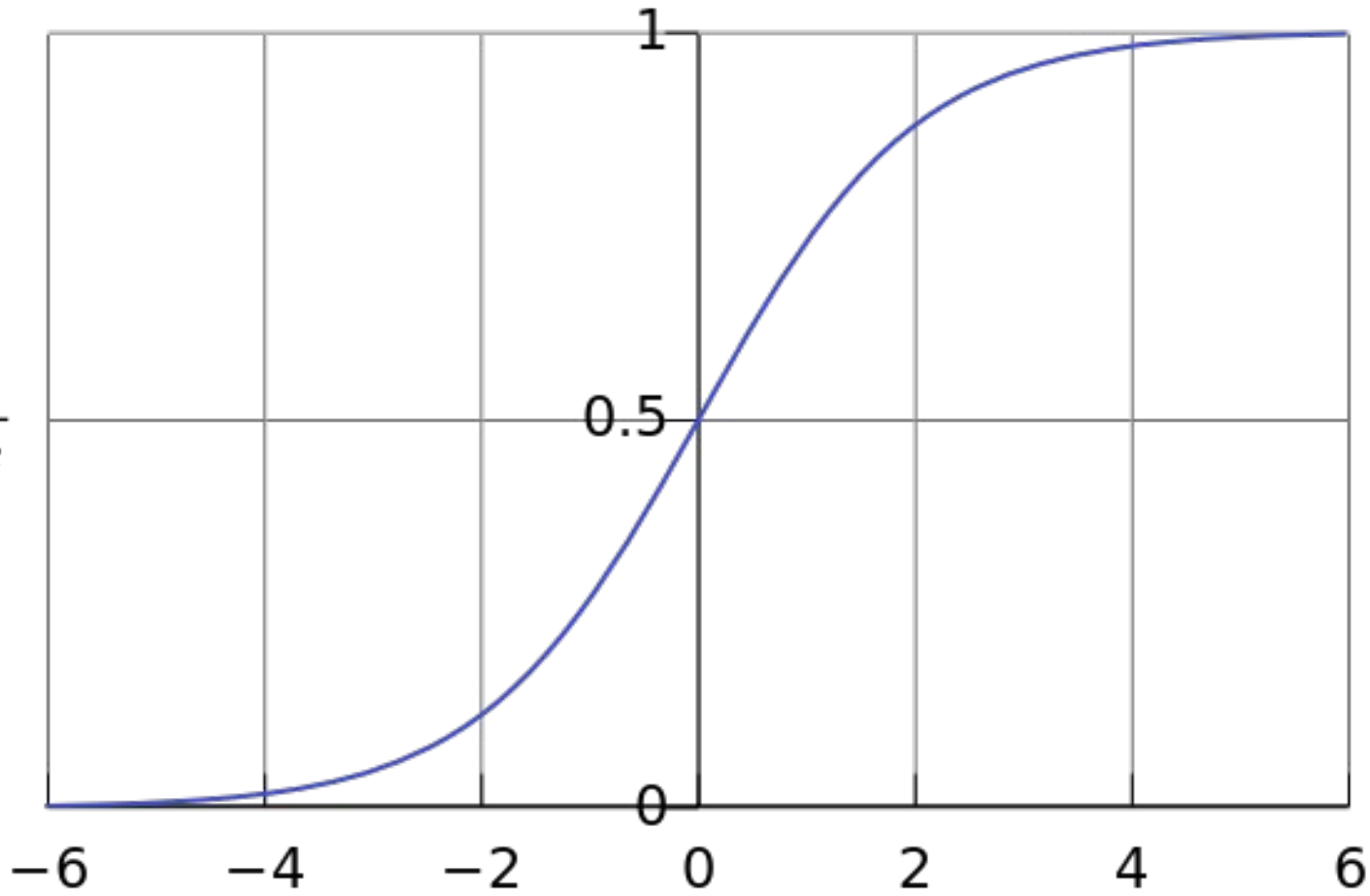
PRINCIPE

Puisque l'on a une variable à expliquer qui ne prend que les valeurs 0 et 1, un modèle linéaire n'a pas de sens.



LA FONCTION LOGIT

$$f(x) = \frac{e^x}{1 + e^x}$$



Source : wikipedia.com

La régression logistique revient à appliquer cette fonction à une régression linéaire simple pour "pousser" les valeurs vers 0 et 1.

PRINCIPE

Comme d'habitude, X est l'input, Y l'output. La régression logistique binaire repose sur le modèle suivant :

$$P(Y=1|X) = \underbrace{\frac{e^{b_0+b_1x_1+\cdots+b_px_p}}{1+e^{b_0+b_1x_1+\cdots+b_px_p}}}_{\text{toujours compris entre 0 et 1}}$$

On réfléchit en termes de **rapport de probabilités** ou encore de **odd ratios** :

$$\ln \frac{P(Y=1|X)}{P(Y=0|X)} = b_0 + b_1x_1 + \cdots + b_px_p$$

TROUVER LA SOLUTION

Il s'agit cette fois de maximiser une fonction que l'on appelle la vraisemblance du modèle. On parle de l'estimation du maximum de vraisemblance.

Intuition: "quels sont les paramètres qui ont vraisemblablement généré ces valeurs que j'observe?".

TROUVER LA SOLUTION

Il s'agit cette fois de maximiser une fonction que l'on appelle la vraisemblance du modèle. On parle de l'estimation du maximum de vraisemblance.

Intuition: "quels sont les paramètres qui ont vraisemblablement généré ces valeurs que j'observe?".

$$\max_{\beta} \prod_{i=1}^{n} P(y_i = 1|x_i, \beta)^{y_i} \times P(y_i = 0|x_i, \beta)^{1-y_i}$$

EXEMPLE

$$\max_{\beta} \prod_{i=1}^{n} P(y_i = 1|x_i, \beta)^{y_i} \times P(y_i = 0|x_i, \beta)^{1-y_i}$$

$$\begin{aligned} x_1 &= (0.5, 3, 8) & y_1 &= 1 \\ x_2 &= (0.7, 4, 9) & y_2 &= 1 \\ x_3 &= (2, 2, 1) & y_3 &= 0 \end{aligned}$$

On cherche les β qui maximisent:

$$P(y_1 = 1|x_1 = (0.5, 3, 8)) \times P(y_2 = 1|(0.7, 4, 9)) \times P(y_3 = 0|x_3 = (2, 2, 1))$$

Il suffit alors de remplacer les probabilités par leur expression logit pour avoir la fonction finale.

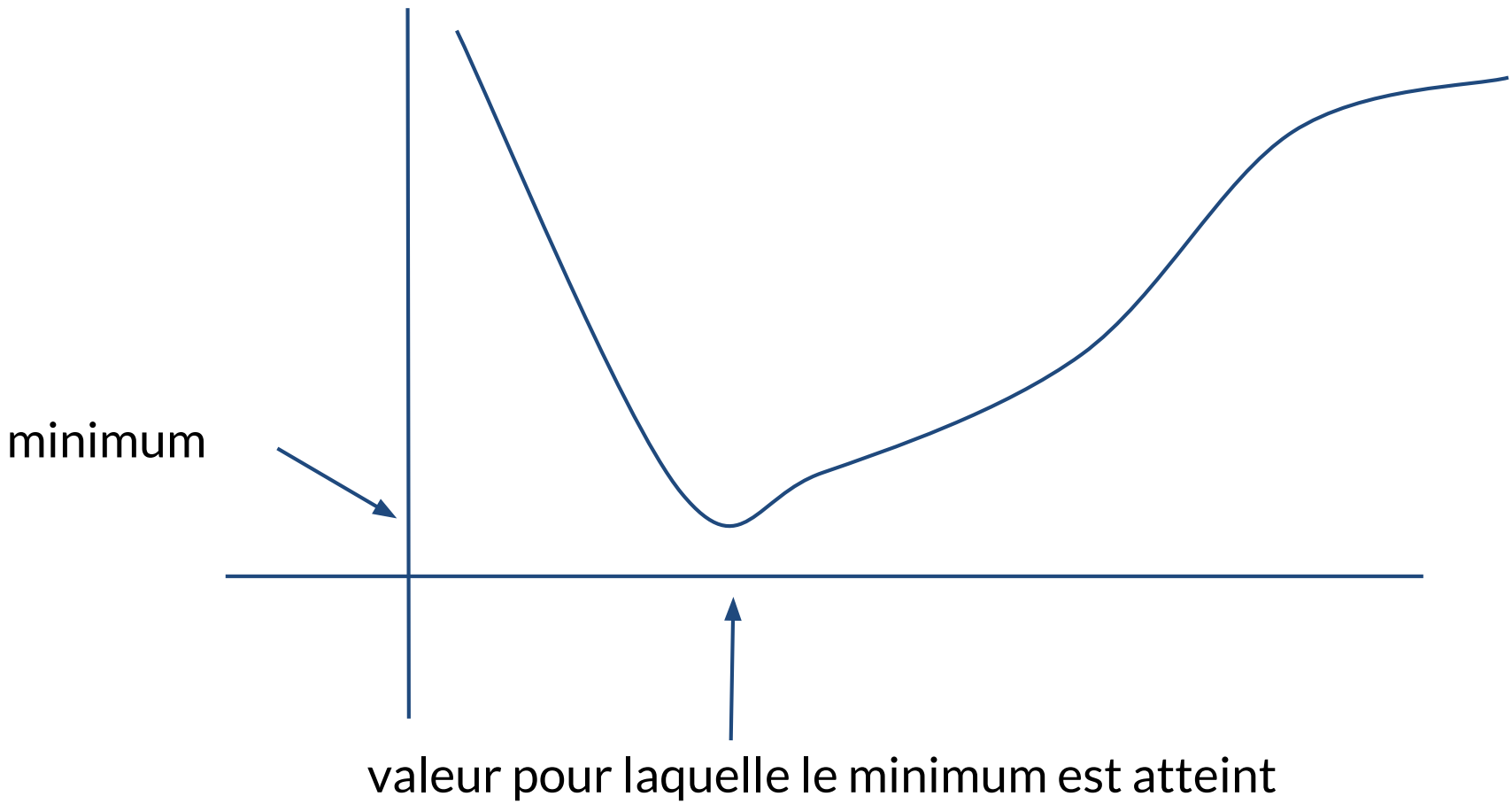
OPTIMISATION

Trouver les poids de la régression logistique revient à maximiser une fonction concave, ce qui revient à minimiser son opposée, qui est convexe.

Comme plus tôt, on calcule sa dérivée et on l'annule. Cette fois, il n'existe pas de forme fermée pour ce calcul.

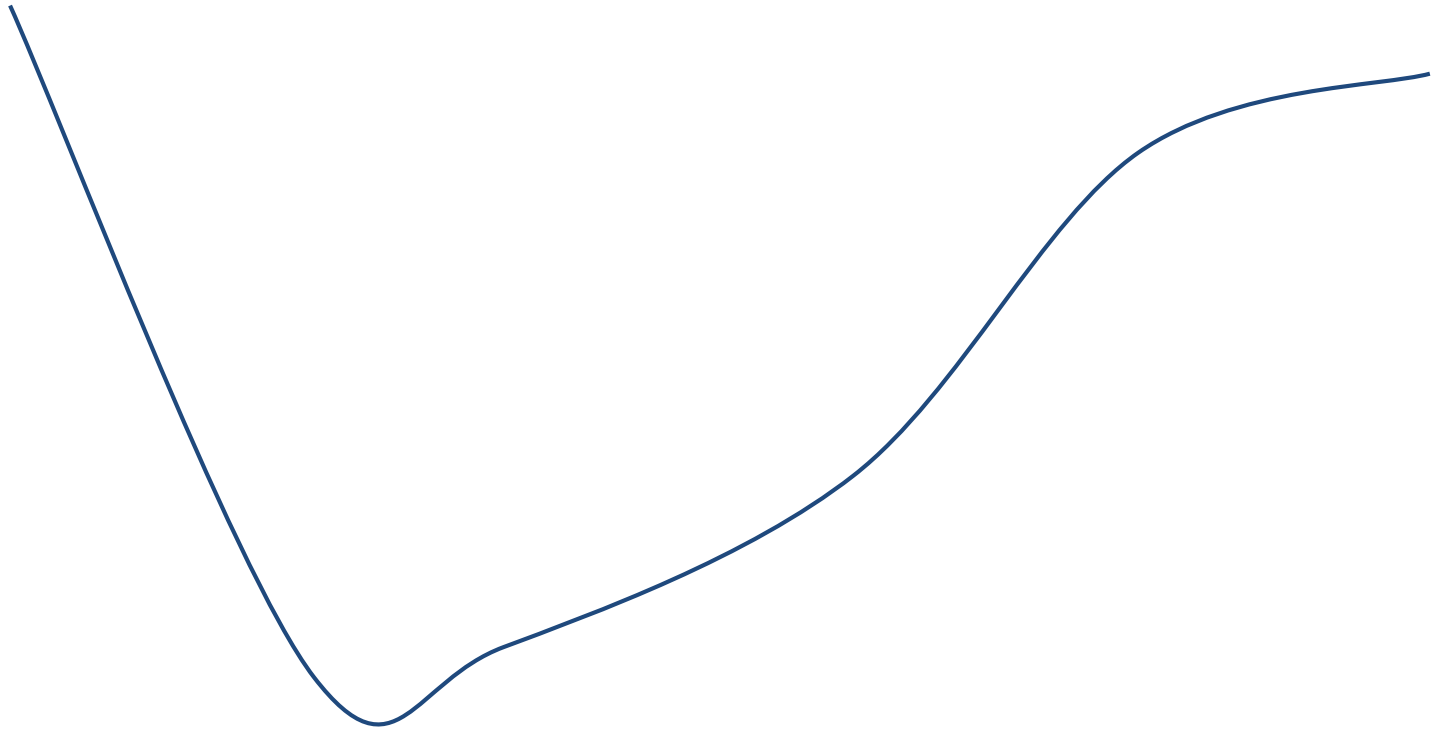
OPTIMISATION

Fonction convexe : un unique minimum.



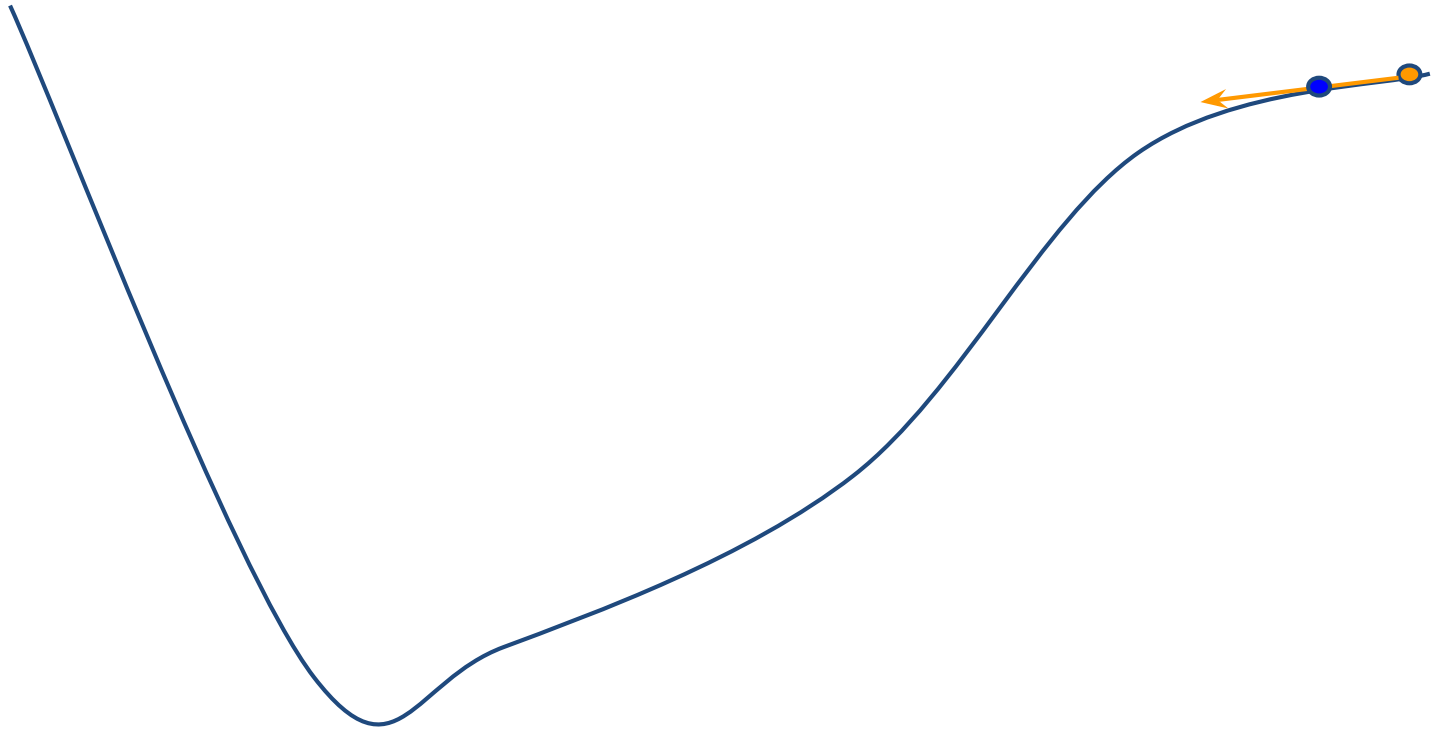
DESCENTE DE GRADIENT

Idée de la descente de gradient: faire des petits pas dans la direction de la pente jusqu'à atteindre le minimum.



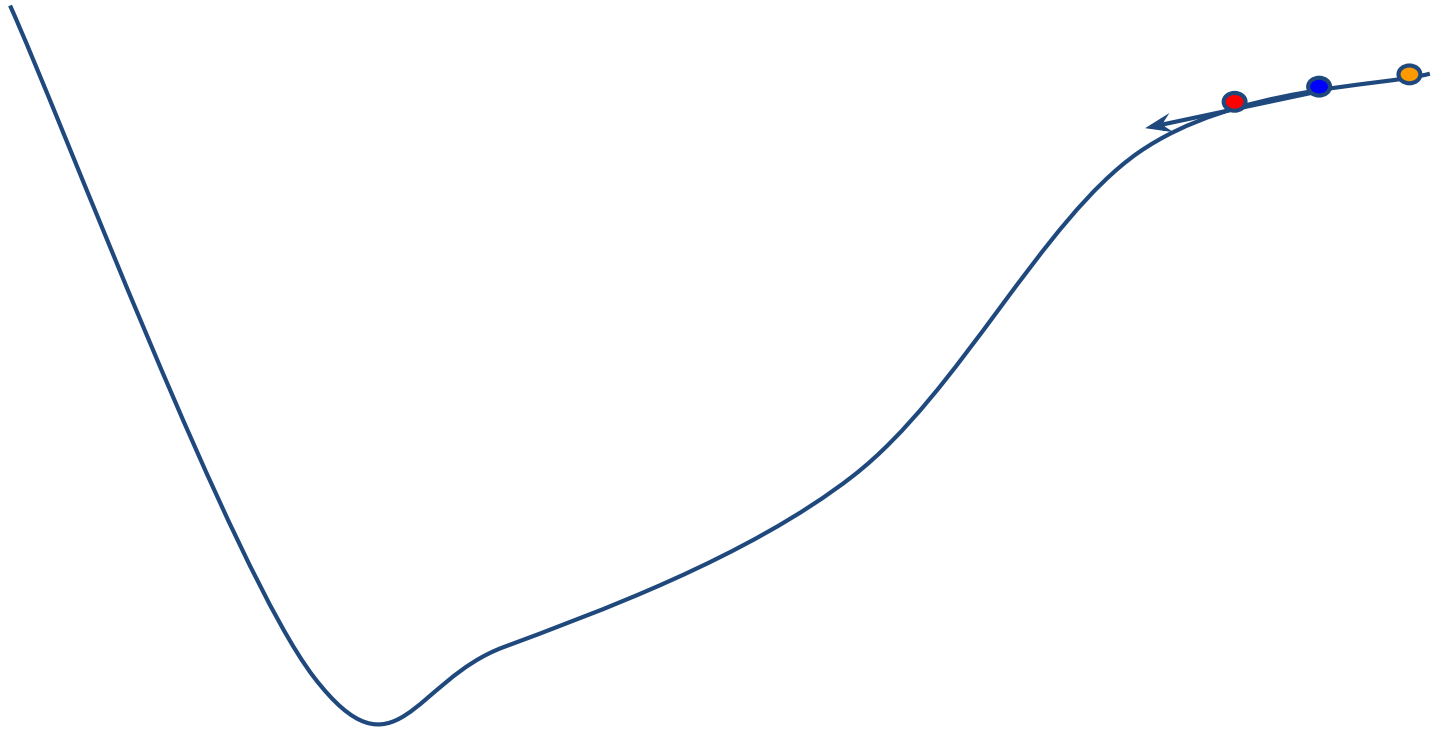
DESCENTE DE GRADIENT

Idée de la descente de gradient: faire des petits pas dans la direction de la pente jusqu'à atteindre le minimum.



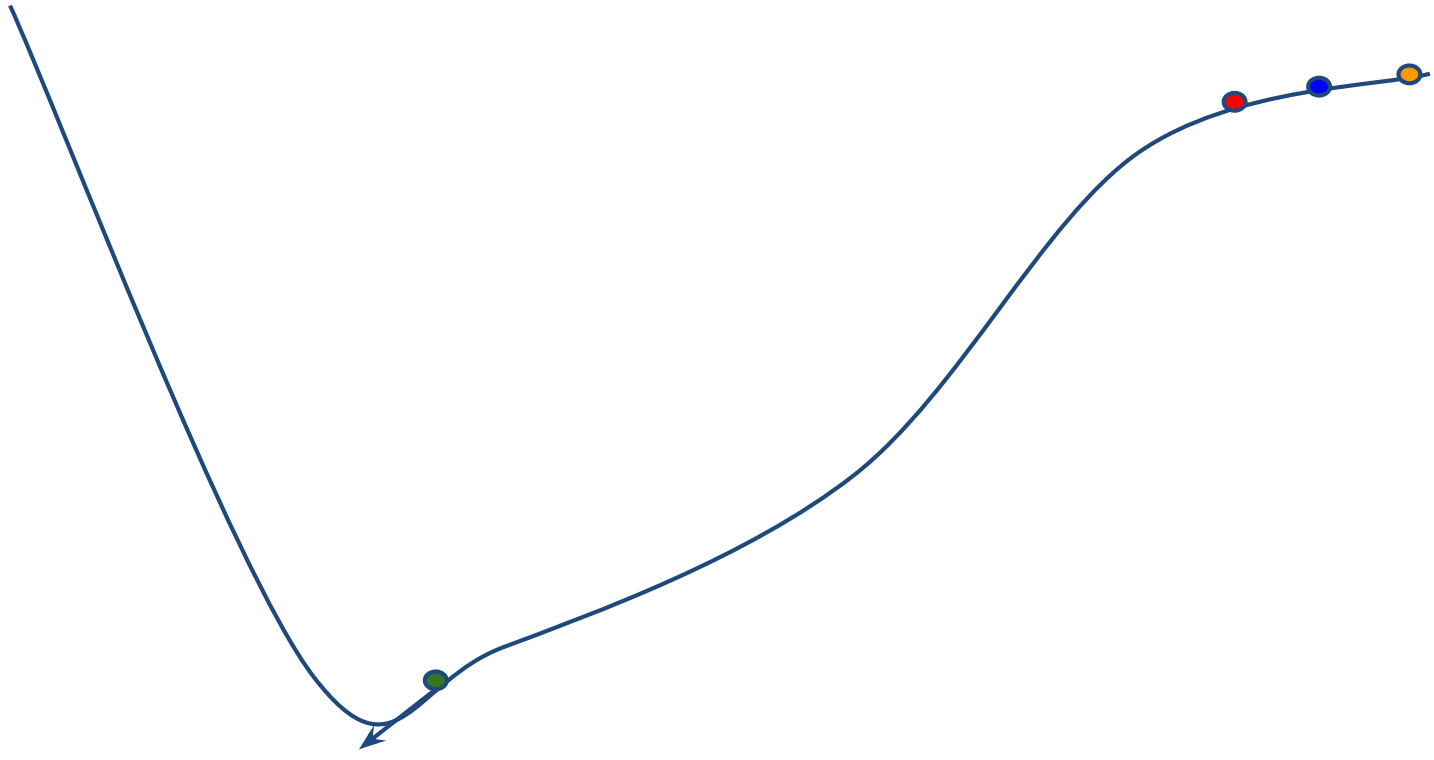
DESCENTE DE GRADIENT

Idée de la descente de gradient: faire des petits pas dans la direction de la pente jusqu'à atteindre le minimum.



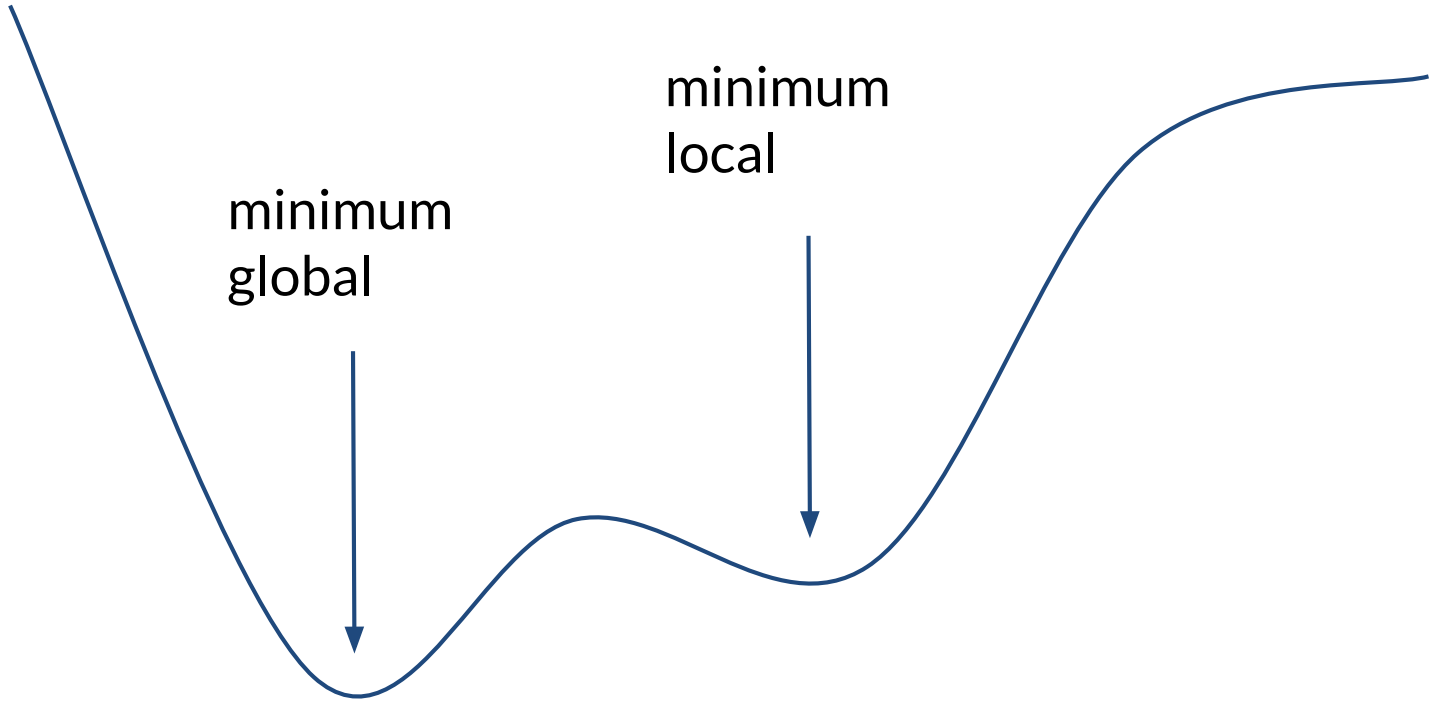
DESCENTE DE GRADIENT

Idée de la descente de gradient: faire des petits pas dans la direction de la pente jusqu'à atteindre le minimum.



DESCENTE DE GRADIENT

Remarque: si la fonction n'est pas strictement convexe, on risque de rester bloquer dans un minimum local.



INTERPRÉTATION

La régression logistique permet d'interpréter l'effet de chaque variable sur ce qu'on essaie d'expliquer.

Exemple: on cherche à expliquer si un étudiant va valider une matière.

$$\ln \frac{P(Valider = 1|\text{travail, cadeaux, video})}{P(Valider = 0|\text{travail, cadeaux, video})} = b_0 + b_1 \text{travail} + b_2 \text{cadeaux} + b_3 \text{video}$$

On trouve $\hat{b}_1 = 0.5, \hat{b}_2 = 1, \hat{b}_3 = -0.5$. On en déduit que:

- exp(0,5) = 1,65 : chaque unité de travail en plus multiplie par 1.65 les chances de valider.
- exp(1) = 2,7: chaque cadeau supplémentaire fait au professeur multiplie par 2.7 les chances de valider.
- exp(-0,5) = 0.6: chaque vidéo regardée pendant le cours divise par 1/0.6 = 1.65 les chances de valider.

RÉCAPITULONS

La régression logistique est le pendant de la régression linéaire pour la classification binaire.

On exprime les probabilités d'observer 1 ou 0 conditionnellement aux valeurs de x avec la fonction logit.

Pour trouver les poids optimaux, on cherche ceux qui rendent le plus vraisemblable d'observer ce qu'on observe. Ceci revient à maximiser une fonction concave, c'est à dire minimiser une fonction convexe, ce qu'on fait par la descente de gradient.

Les poids obtenus s'interprètent comme des multiplicateurs de probabilités et permettent maintenant de prédire la valeur de y pour un nouveau x.

EN PYTHON

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

CONCLUSIONS SUR LA RÉGRESSION LOGISTIQUE

Avantages

Une vision probabiliste de la classification

Interprétation de chaque variable de manière probabiliste

Simple et rapide à estimer

Inconvénients

Mêmes que la régression linéaire

En particulier, mauvais résultats avec beaucoup de variables.