

# Cours 4: Complexité, Diagnostic des courbes d'entraînement, interprétabilité et métriques de performance

Gauthier Gidel  
11 Septembre 2024

# Annonces

- Le projet va débuter d'ici la fin de la semaine.
- Remplir le formulaire (Google form)
- Super important pour le projet.

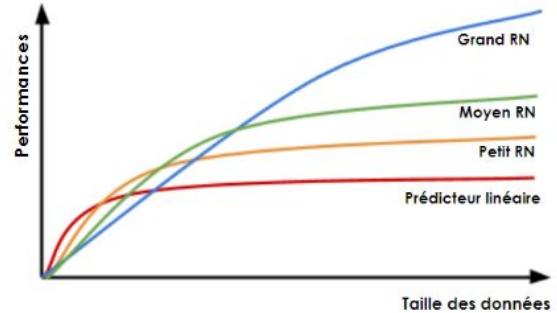
# Rappel: Complexité d'un modèle en apprentissage automatique (AA)

- Capacité à extraire de plus en plus d'information des données, jusqu'à ce qu'il (le modèle) soit complètement « saturé » d'information.
- Principaux facteurs affectant la complexité d'un modèle
  - Nombre de paramètres à entraîner
  - Structure/architecture
- Un petit réseau de neurones (RN) profond est souvent plus performant qu'un réseau peu profond, mais très large.



# Situations courantes lors de l'entraînement d'un modèle en AA

- On peut améliorer ses performances en augmentant les données d'entraînement.
- Elles finissent par saturer; ajouter des données n'améliore rien...
- Les performances maximales dépendent de sa **complexité**.
- Pour de meilleures performances, il faut augmenter sa complexité ou changer de modèle.



# Complexité et méthodes linéaires

# Exemple de régression multilinéaire

- On peut complexifier un modèle de base en y ajoutant de nouvelles combinaisons de caractéristiques:

- Modèle de base:  $\hat{y} = \beta_0 + \beta_1 x_1 + \beta_2 x_2$
- Ajout d'interactions:  $\hat{y} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1 x_2$
- Ajout de nonlinéarités:  $\hat{y} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1 x_2 + \beta_4 x_1^2 + \beta_5 x_2^2$
- Ajout de caractéristiques:  $\hat{y} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \dots$

**L interact<sup>o</sup> entre x<sub>1</sub> & x<sub>2</sub>**

- Les paramètres  $\beta$  des modèles linéaires sont faciles à estimer avec la méthode des moindres carrés.
- Le nombre d'interactions possibles augmente comme  $D^2$  où  $D$  est la dimensionnalité de  $x$

# Complexité et arbres décisionnels

# Les modèles basés sur les arbres décisionnels

- Un arbre décisionnel gère *implicitement* les interactions entre les attributs x sans avoir à les spécifier comme dans un modèle multilinéaire.
- C'est aussi le cas des modèles basés sur des arbres de décision
  - Forêt aléatoire
  - Gradient Boosted Trees
  - Boosted Forests
  - XGBoost
- Beaucoup de problèmes réels se prêtent naturellement à une analyse basée sur des arbres décisionnels.

# Exemple: la modélisation des profits d'une station de ski

- On doit tenir compte de plusieurs attributs  $x$ 
  - Météo
  - Jour de la semaine,
  - Congé?
  - Neige abondante
- **Chaque skieur utilise un arbre décisionnel avec ces attributs pour décider s'il viendra skier ou non.**
- Les profits vont donc dépendre des décisions d'un grand nombre d'arbres décisionnels.

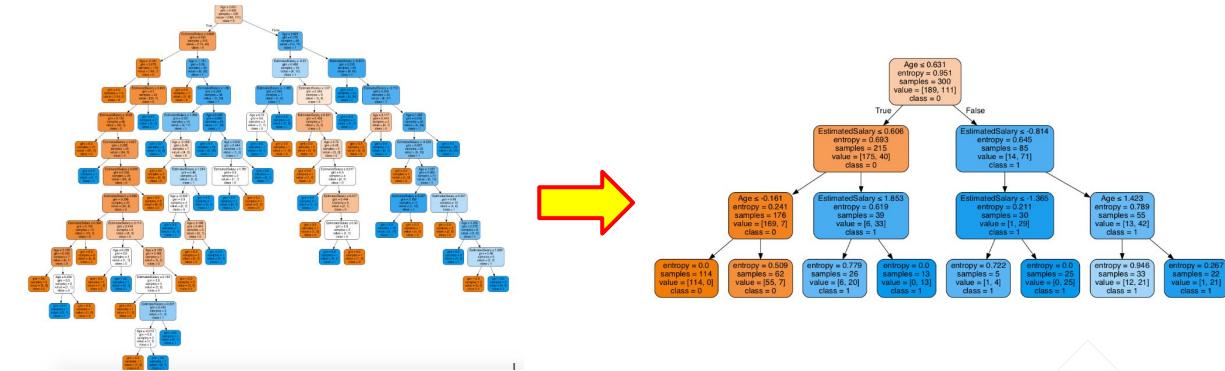


La foule décide  
comme une forêt  
aléatoire!

# Stratégies pour contrôler la complexité d'un arbre décisionnel

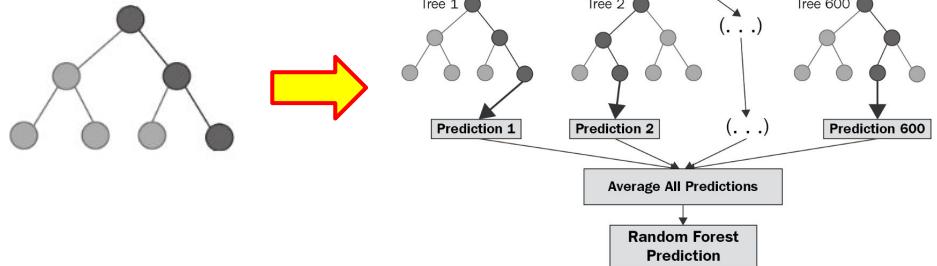
- Réduction de la complexité

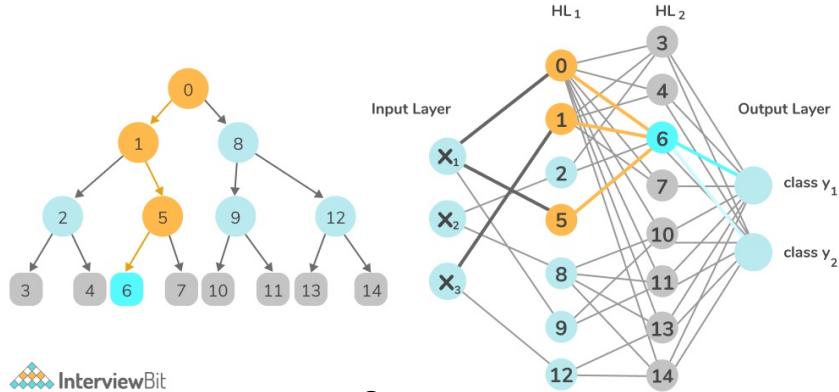
- Limiter sa profondeur



- Augmentation de la complexité

- Passer à une forêt aléatoire

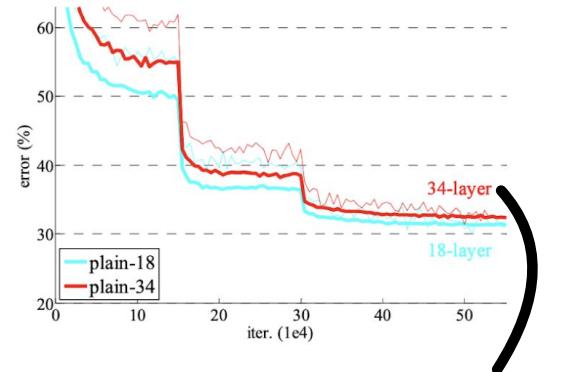




# Ajouter des couches à un réseau de neurones n'est pas toujours la solution

nb couches ↗  
important  
que la largeur

- Ça améliore les performances d'un réseau de neurones jusqu'à un certain point.
- Les performances peuvent chuter si on en ajoute trop (problème de la rétropropagation du gradient).
- C'est particulièrement vrai avec l'architecture classique des RN convolutionnels.
- L'erreur de classification finit par augmenter avec le nombre de couches...

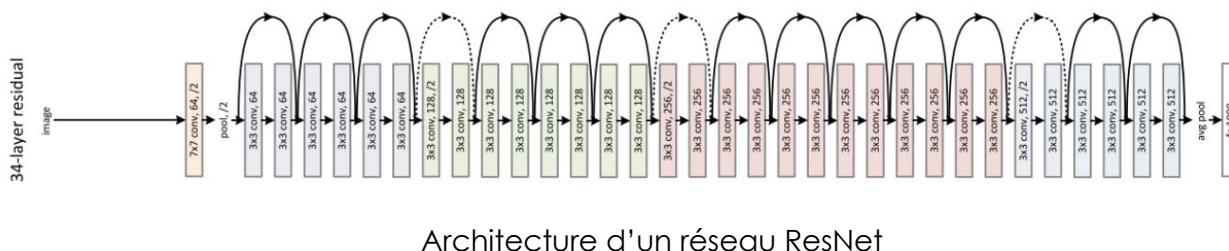
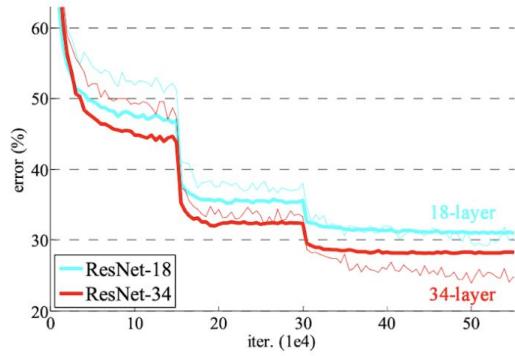


Architecture classique d'un RN convolutionnel

surappren-  
tissage

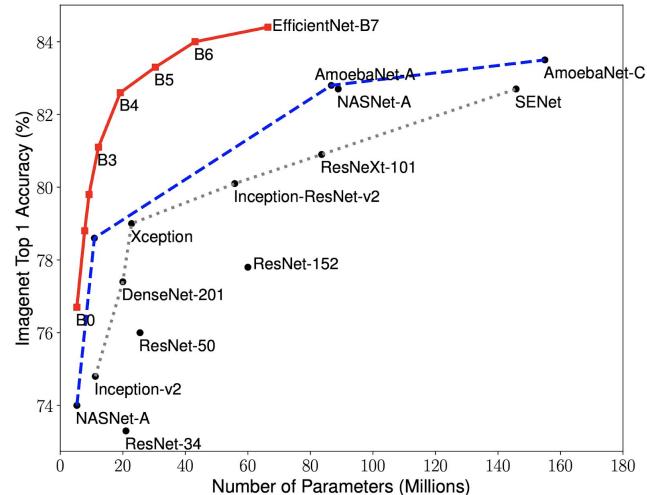
# Changer la structure du réseau de neurones est souvent plus efficace

- Le réseau ResNet surpassé les performances des RN convolutionnels classiques.
- La structure de ResNet lui permet d'améliorer ses performances en ajoutant des couches.
- Famille de modèles Resnet (18, 34 , 50, 101, 152 couches)



# Changer de famille de modèles est encore plus efficace

- Les performances finissent par plafonner pour une famille de structure donnée.
- Il faut tester des structures différentes de RN
- Trop peaufiner un modèle préféré diminue les *retours sur investissement*.
- Il est important d'être au courant de la littérature et des ressources disponibles.



# L'importance du logiciel libre (*Open Source code*)

- Il est rare qu'on ait à coder à partir de zéro un RN complexe.
- Le code source pour la majorité des modèles en AA est mis sur un site de partage après la publication des articles scientifiques associés!
- Les poids des RN déjà entraînés peuvent être téléchargés!
- **On peut ajuster un nouveau RN à son problème et ses données en utilisant le transfert d'apprentissage.**
- Plusieurs services web d'hébergement et de gestion de développement de logiciels existent. 

statech/resnet

keras-style API to ResNets (ResNet-50, ResNet-101, and ResNet-152)

1  
Contributor

58  
Used by

22  
Stars

20  
Forks

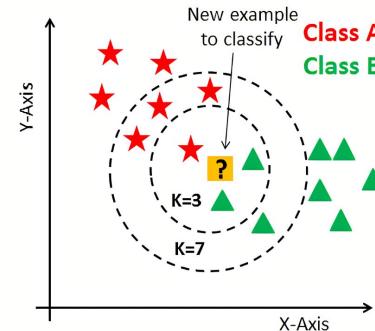


GitHub	GitLab	BitBucket
SourceForge	Cloud Source	AWS CodeCommit

# Temps d'apprentissage et d'inférence

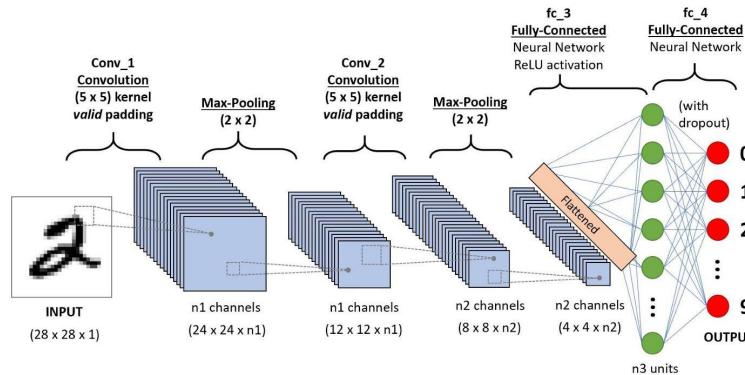
## ↓ Ce que l'on veut optimiser

- **Temps d'inférence/latence:** temps requis pour effectuer une prédiction.
- Modèle simple de classification: K plus proches voisins
  - **Temps d'apprentissage nul!**
  - Calcul des distances aux N données d'entraînement à chaque fois...
  - Temps d'inférence proportionnel à N
  - Temps en secondes si K est grand
  - **Peu pratique en temps réel...**



## Modèle complexe de classification: réseau de neurones

- Temps d'apprentissage long (heures, jours)...
- Peut être réduit grâce au transfert d'apprentissage
- Temps d'inférence constant (quelques ms à une seconde) **- dépend φ taille jeu de données**
- Couramment utilisé en temps réel!



# Conclusion

Deux grand moments 'd'utilisation' pour les modèles:

- Entraînement:
  - Important lors du développement des modèles
- Inférence:
  - Important lors de l'utilisation

Peuvent être réduits (demande plus de code et de ressources) à l'aide de:

- Parallélisation (entraînement distribué)
- Utilisation d'un cluster.
- Matériel plus performant (GPU, vs CPU)

Plus de détails dans des cours dédiés.

## Inférence

Temps qu'il faut modèle déjà entraîné pr faire des predict\* sur nulles données  
→ obj : predict\* rapide

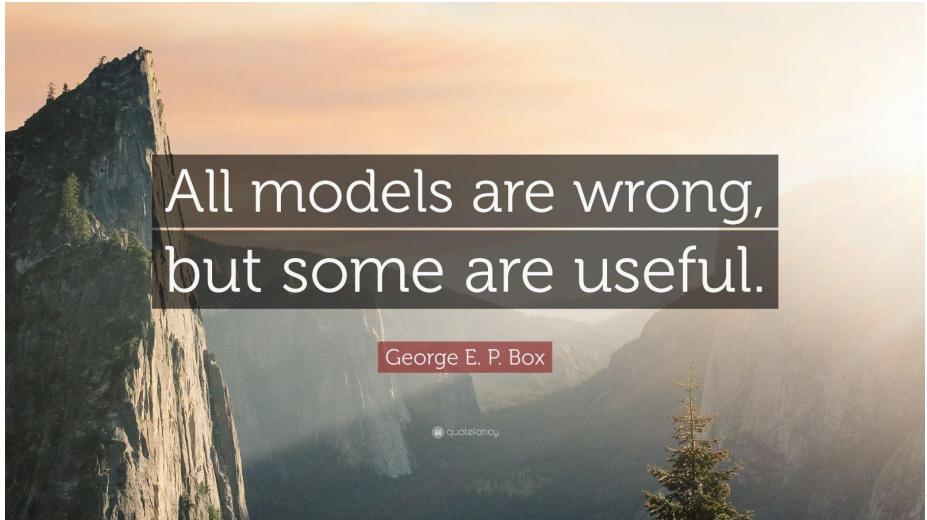
## Apprentissage

Temps nécessaire pr qu'un modèle apprenne à partir de données d'entraînement  
→ Objectif : predict\* correct

# Interprétabilité

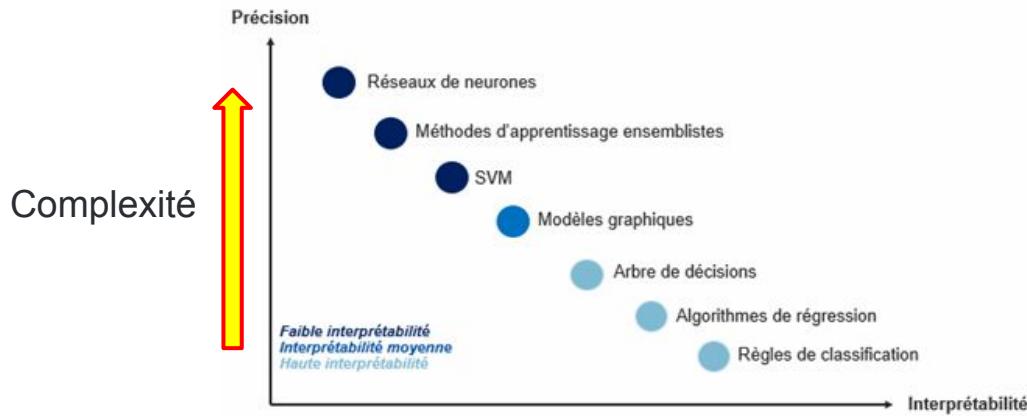
# Choisir un modèle d'apprentissage automatique

- L'AA fournit beaucoup de modèles pour analyser des données.
- Certains sont précis, mais incompréhensibles.
- D'autres sont simples, mais imprécis.
- Chacun représente une approximation de la réalité.
- Lequel choisir?



# L'importance d'expliquer les résultats des prédictions

**En général**, lorsque la complexité d'un modèle augmente, son interprétabilité diminue.



Exemples concrets:

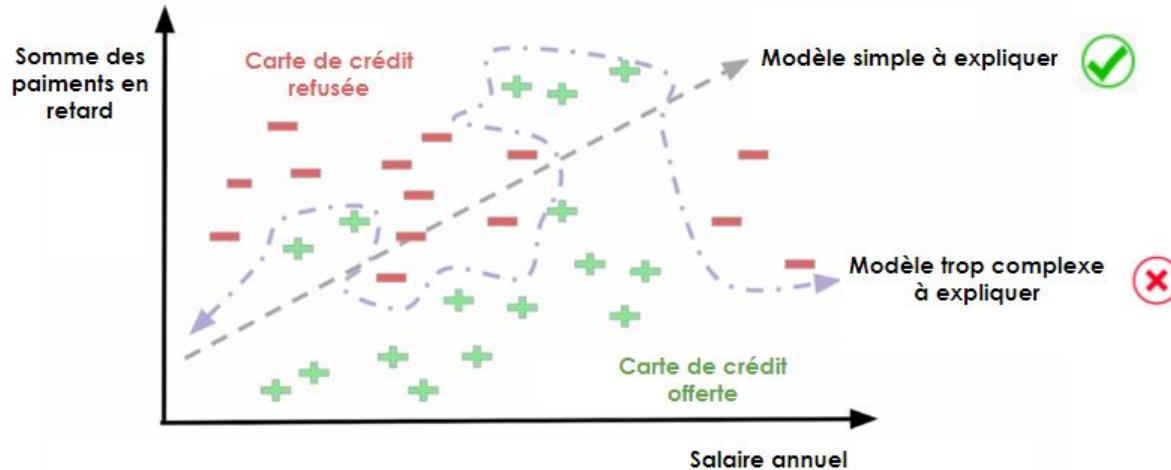
- Prêt bancaire refusé
- Détection de cancer
- Candidature non retenue
- Etc.

Pourquoi ces décisions?

✗ interprétable  
✓ explicable

# Exemple de demande de carte de crédit

Comment un banquier pourrait-il expliquer sa décision à un client?



Que vaut l'exactitude à 95% du modèle complexe, sans intervalle de confiance, et ne pouvant être expliqué?



# Biais et Interprétabilité

...

# L'interprétabilité peut aider à détecter les biais

## Exemple de **biais de sélection**

- On analyse les CV/résumés de 100 femmes et de 1000 hommes.
- Il y a plus que les caractéristiques corrélées avec l'embauche soient des caractéristiques partagées par les candidats masculins.
- Un modèle entraîné sur ces données pourrait déterminer que le genre (ou des caractéristiques corrélées) des applicants est un attribut principal de performance!
- Plus de détails dans le cours sur les biais algorithmiques.

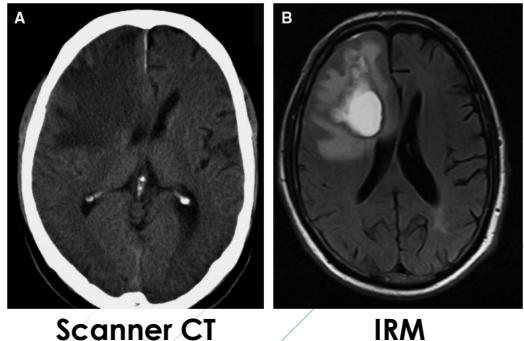


# L'interprétabilité peut aider à détecter les biais

Exemple de **biais de mesure**

- Un système d'analyse d'images médicales est entraîné à détecter des cancers dans des images de type IRM.
- Problème:
  - Les cas positifs sont confirmés avec la machine B.
  - Les cas négatifs proviennent tous de la machine A.
- Le taux de détection de cancers sera biaisé par la machine utilisée.

Données proviennent  
d'environnements  
différents





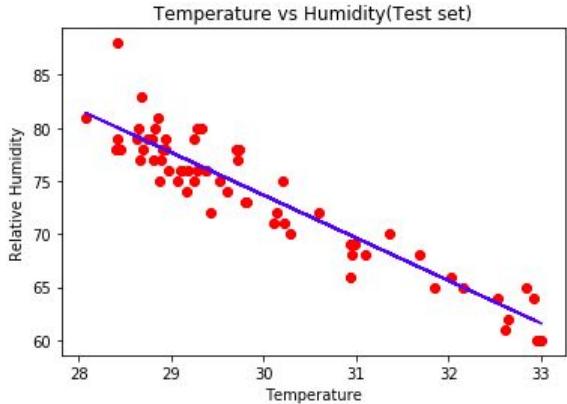
# Interprétabilité *versus* complexité

...

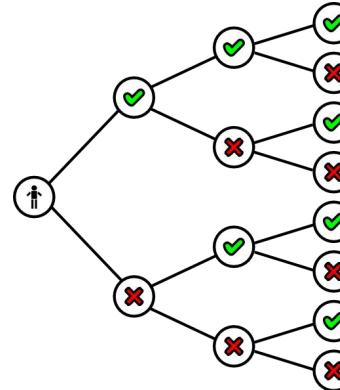
# Interprétabilité des résultats: modèles simples

Il est facile d'expliquer les décisions prises par les modèles simples tels que :

Régression linéaire

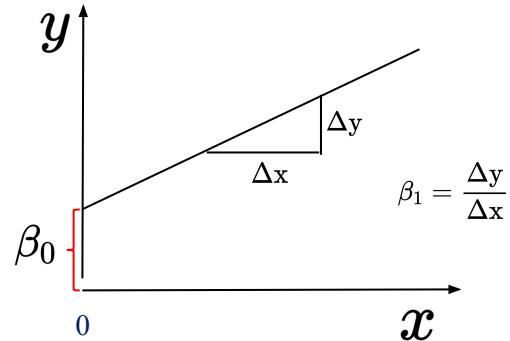


Arbre décisionnel



# Exemple de la régression linéaire

- Modèle linéaire simple :  $y = \beta_0 + \beta_1 x$

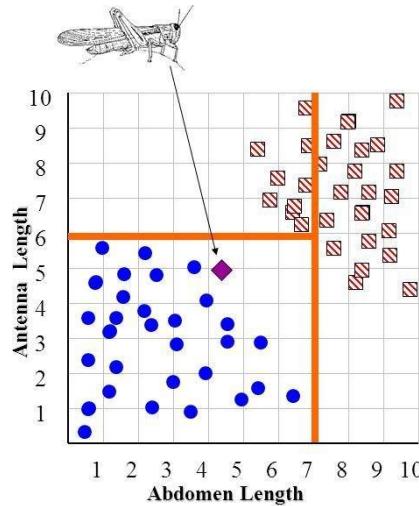
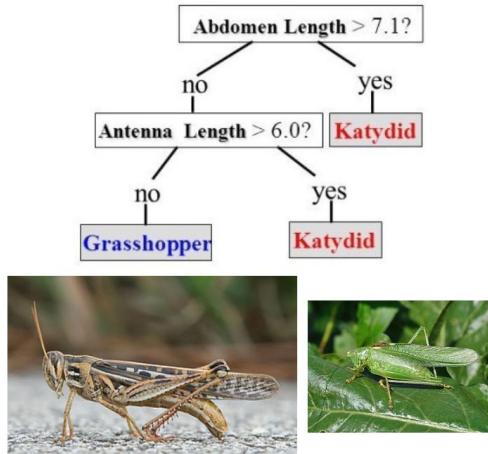


- Modèle linéaire général :  $y = \beta_0 + \beta_1 x_1 + \dots + \beta_D x_D$

- Les plus grandes valeurs  $|\beta_i|$  correspondent aux attributs (normalisés)  $x_i$  les plus importants.
- Lorsque  $\beta_i > 0$ , il existe une corrélation positive entre l'attribut  $x_i$  et la réponse  $y$ .
- Une variation  $\Delta x_i$  produit une variation de la réponse  $\Delta y = \beta_i \Delta x_i$  lorsque les autres attributs  $x_j$  restent inchangés.

# Exemple d'arbre décisionnel

- Arbre de régression ajusté sur les données morphométriques (en cm) de deux types de sauterelles (*grasshoppers* et *katydids* en anglais)
- Seuls quelques attributs X permettent de différencier les deux types d'insectes.



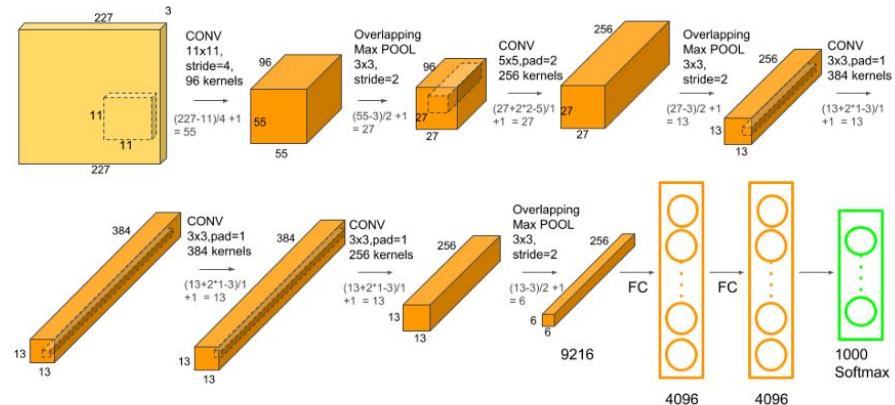
Classification d'une nouvelle sauterelle de type grasshopper.

# Interprétabilité des résultats: réseaux de neurones

- Pour faire des prédictions avec un réseau de neurones, les données d'entrée sont passées à travers de nombreuses couches effectuant des produits matriciels suivis de transformations non linéaires.
- Difficile de suivre le flux des données d'une couche à l'autre.
- Une seule prédiction peut impliquer des millions d'opérations mathématiques en fonction de l'architecture du réseau de neurones.



# Exemple de réseau de neurones profond de complexité modeste



Le réseau de neurone convolutionnel **AlexNet** contient

- Huit couches cachées
- Une fonction d'activation nonlinéaire (ReLU, Leaky ReLU, etc.)
- **62 378 344** paramètres!

# Interprétabilité des réseaux de neurones

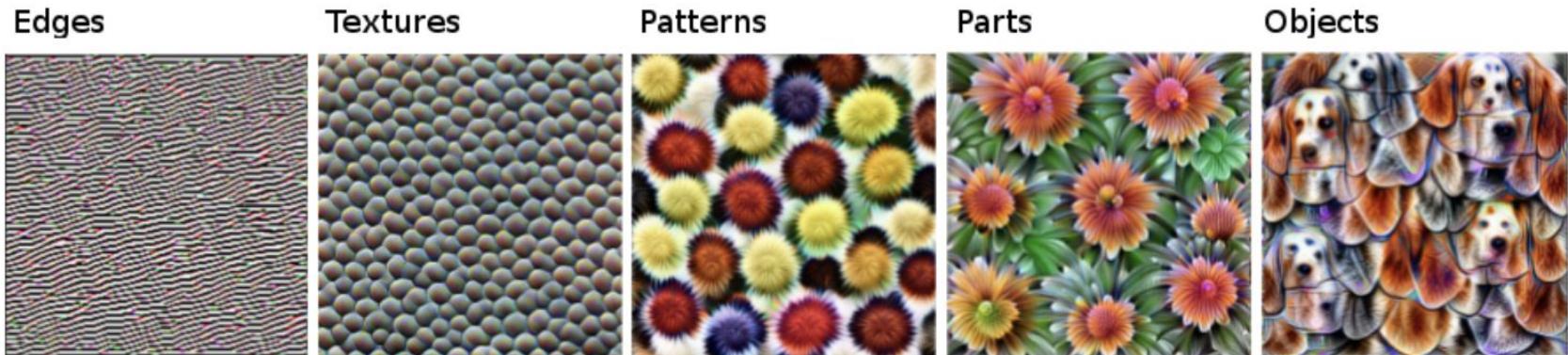
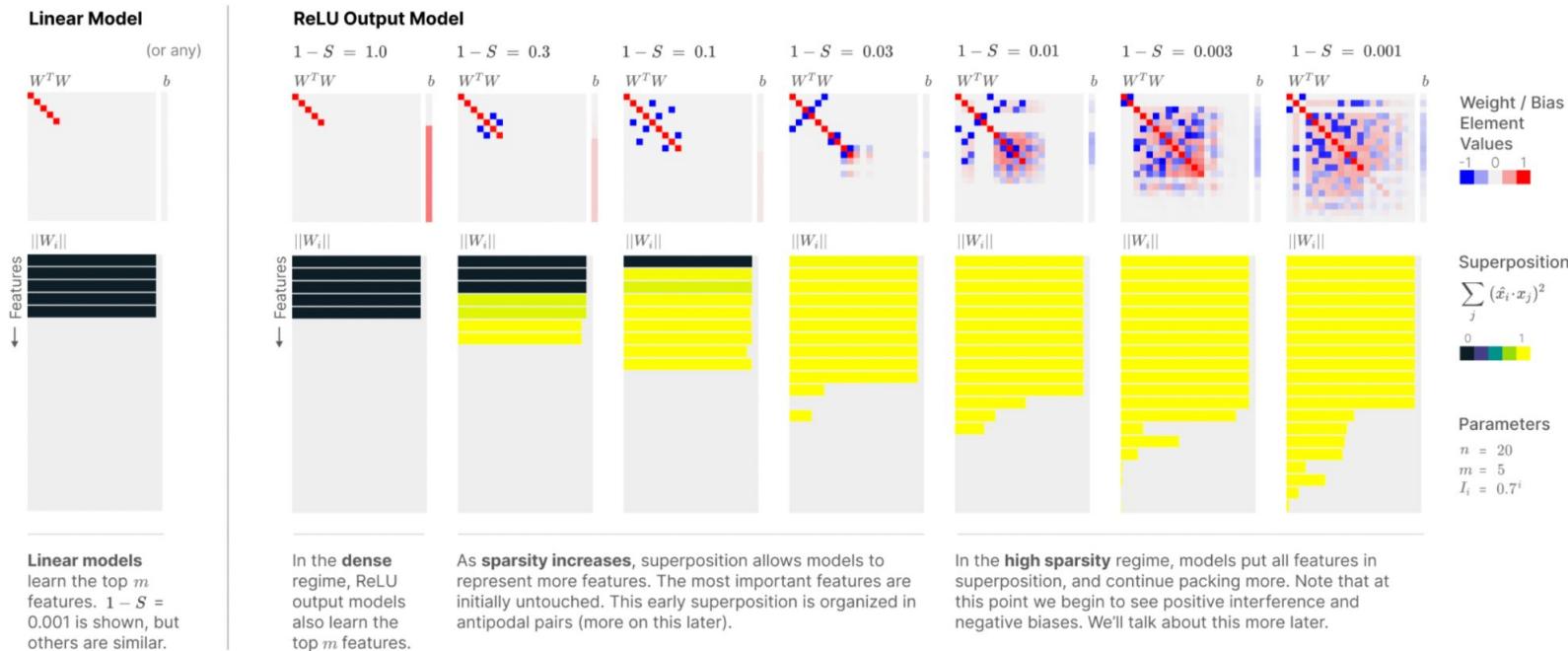


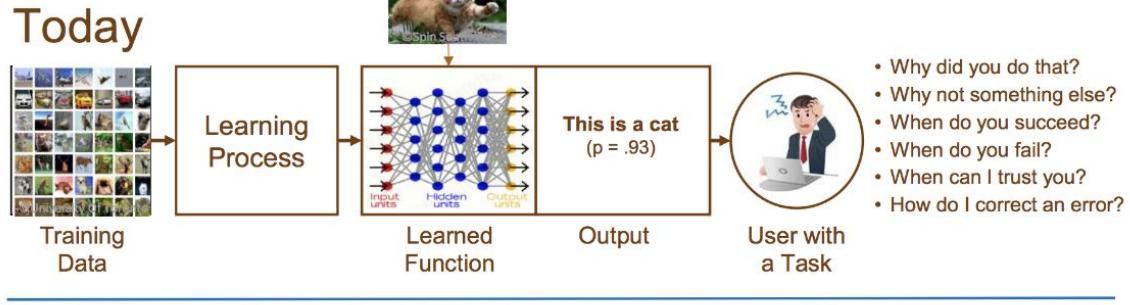
FIGURE 10.1: Features learned by a convolutional neural network (Inception V1) trained on the ImageNet data. The features range from simple features in the lower convolutional layers (left) to more abstract features in the higher convolutional layers (right). Figure from Olah, et al. (2017, CC-BY 4.0) <https://distill.pub/2017/feature-visualization/appendix/>.

# Interprétabilité des réseaux de neurones... un problème ouvert

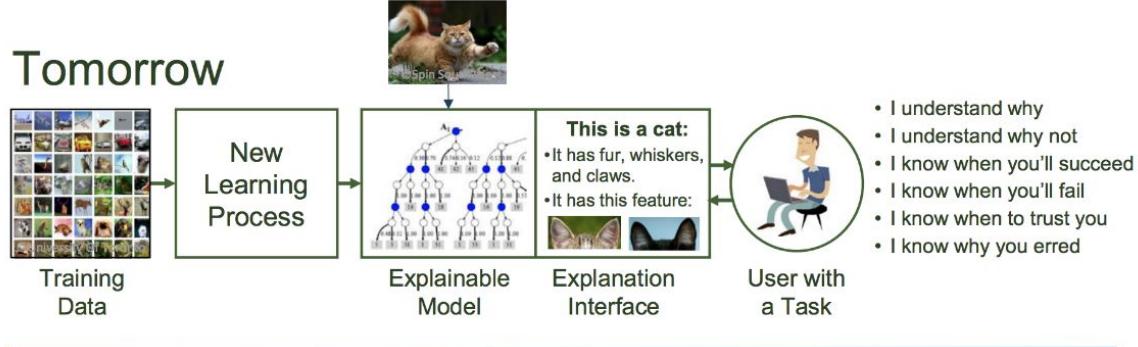


# L'interprétabilité des résultats des réseaux de neurones est un champ actif de recherche

Il faut être un expert pour comprendre un résultat et ses implications.



But: Un système expert de nouvelle génération **pourrait expliquer** ses décisions.

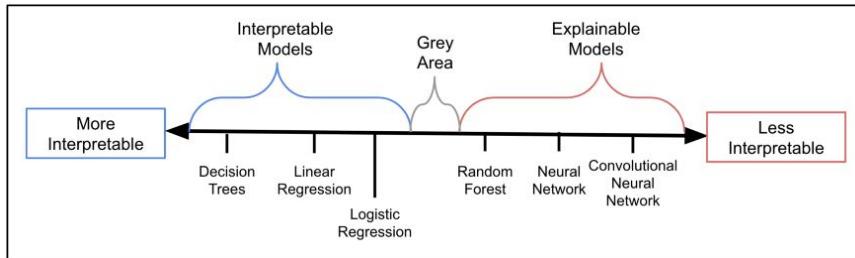




# Interprétabilité v.s. explicabilité

...

# Différences entre les modèles interprétables et explicables



- Un modèle **interprétable** est facile à comprendre pour un humain
- Sa structure est simple et le nombre de paramètres est faible.

- Un modèle **explicable** ne peut pas être compris exactement de façon simple, mais certaines stratégies permettent d'analyser les facteurs qui influencent ses prédictions.
- Sa structure est complexe et le nombre de paramètres est élevé.

# Quels sont les facteurs importants dans un modèle?

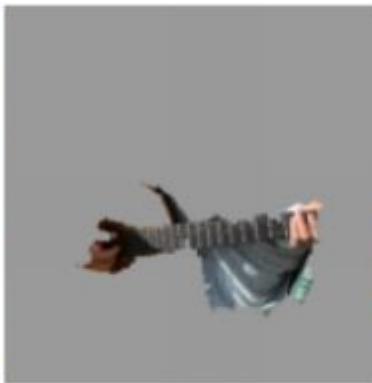
- Les modèles **interprétables** permettent souvent d'identifier les facteurs importants
  - Régression linéaire
  - Régression logistique
  - Arbres décisionnels

- Il existe des méthodes permettant d'**expliquer** les facteurs importants **localement**, i.e., pour **une donnée particulière**
  - LIME
  - SHAP
  - DeepLIFT

# Exemple d'application de LIME pour des images



(a) Original Image



(b) Explaining *Electric guitar*



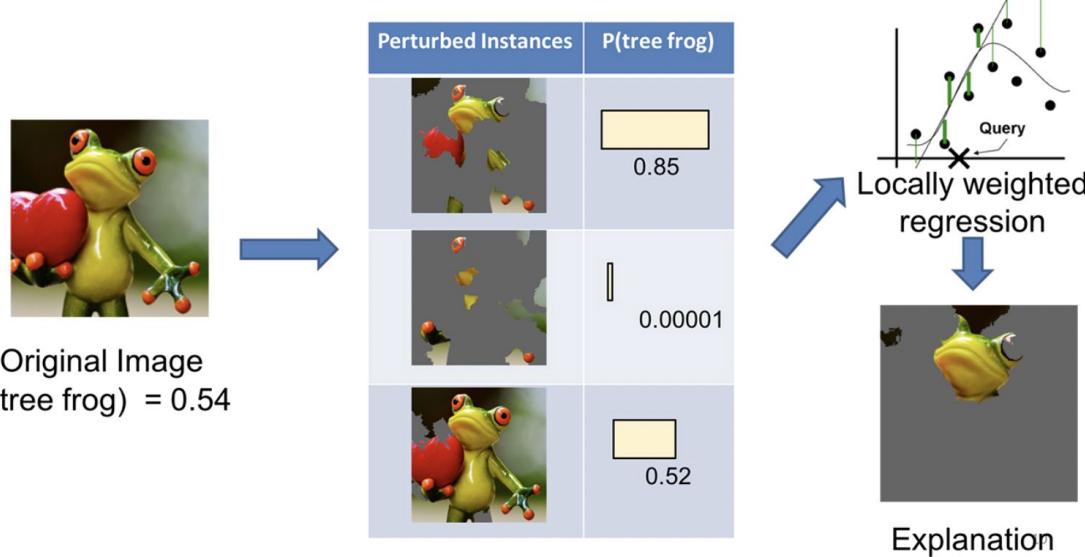
(c) Explaining *Acoustic guitar*



(d) Explaining *Labrador*

**Figure 4: Explaining an image classification prediction made by Google's Inception network, highlighting positive pixels. The top 3 classes predicted are “Electric Guitar” ( $p = 0.32$ ), “Acoustic guitar” ( $p = 0.24$ ) and “Labrador” ( $p = 0.21$ )**

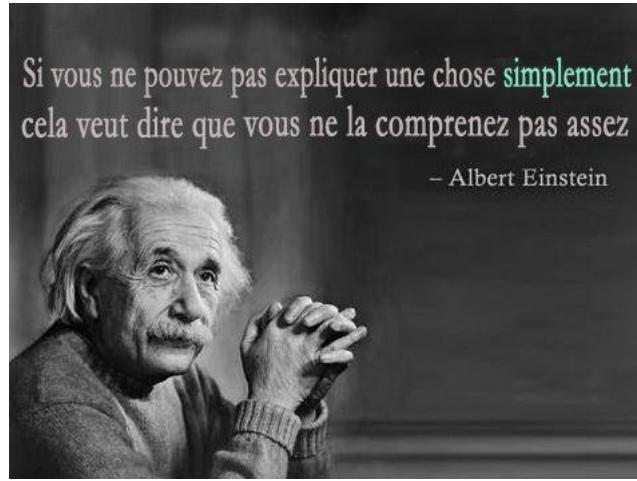
# Exemple d'application de LIME pour des images



Pour l'image ci-dessus, LIME fait ceci

- Masque dans l'image les zones où ces attributs ont un effet négligeable.
- <https://www.youtube.com/watch?v=hUnRCxnydCc>

# L'importance d'expliquer un modèle



Si vous ne pouvez pas expliquer une chose simplement  
cela veut dire que vous ne la comprenez pas assez

– Albert Einstein

C'est vrai en relativité, en mathématiques, en génomique, en biochimie, et aussi en apprentissage automatique.

# Diagnostic des courbes d'entraînement

# Le stéthoscope de l'AA: la courbe de la fonction de perte



...



Qu'est-ce qui est le mieux pour un classifieur nouvellement entraîné?

Ces résultats basés sur deux chiffres sont difficiles à interpréter!

- Cas A
  - Une précision en entraînement de 99%
  - Une précision en test de 80%
- Cas B
  - Une précision en entraînement de 85%
  - Une précision en test de 83%
- Cas C
  - Une précision en entraînement de 30%
  - Une précision en test de 25%
- Cas D:
  - Une précision en entraînement de 85%
  - Une précision en test de 90%



Qu'est-ce qui est le mieux pour un classifieur nouvellement entraîné?

Ces résultats basés sur deux chiffres sont difficiles à interpréter!

- Cas A

- Une précision en entraînement de 99%
- Une précision en test de 80%

Surapprentissage



- Cas B

- Une précision en entraînement de 85%
- Une précision en test de 83%

Optimal



- Cas C

- Une précision en entraînement de 30%
- Une précision en test de 25%

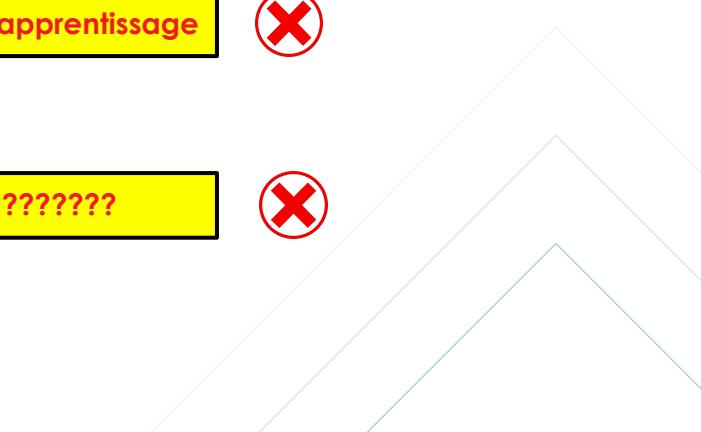
Sous-apprentissage



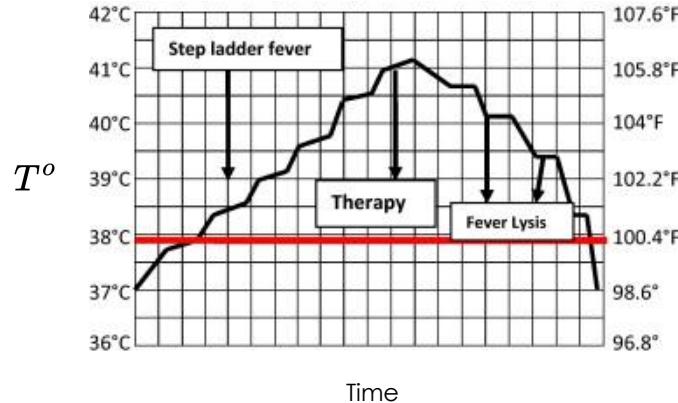
- Cas D:

- Une précision en entraînement de 85%
- Une précision en test de 90%

???????



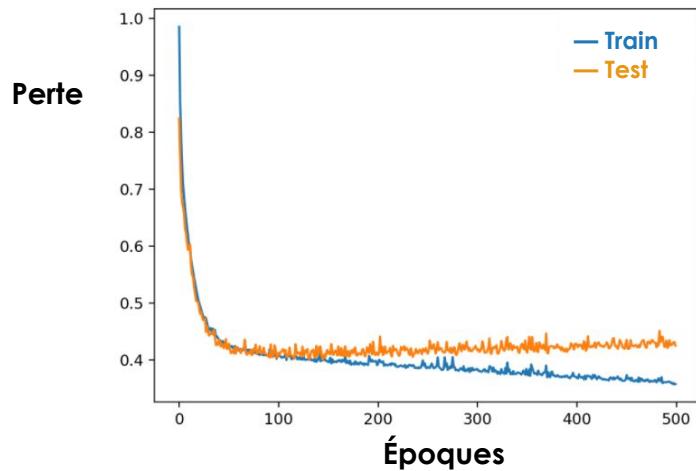
**I'évolution de la température d'un patient est souvent aussi utile que sa température à un moment donné.**



Il ne faut pas se contenter des performances finales d'un modèle pour décider s'il est bien entraîné; il faut aussi voir ses différentes 'courbes de température':

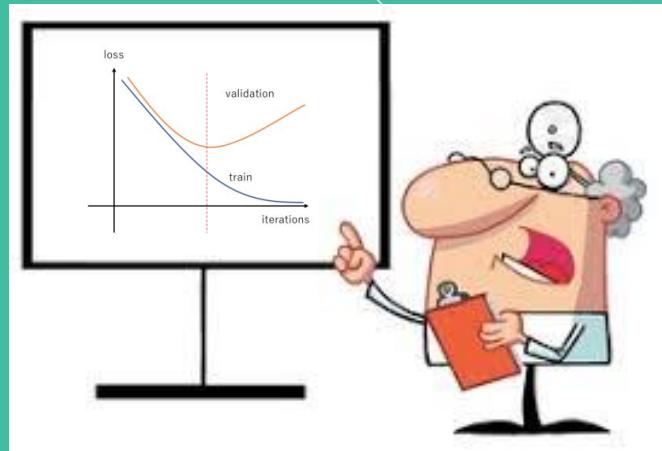
- L'évolution de la fonction de perte
- L'évolution de la métrique de performance

Courbe d'apprentissage: tracé de la performance d'apprentissage en fonction du temps.



Seuls les algorithmes qui apprennent **de manière incrémentale** ont une telle courbe.

# Courbes d'entraînement typiques de modèles ‘en santé’

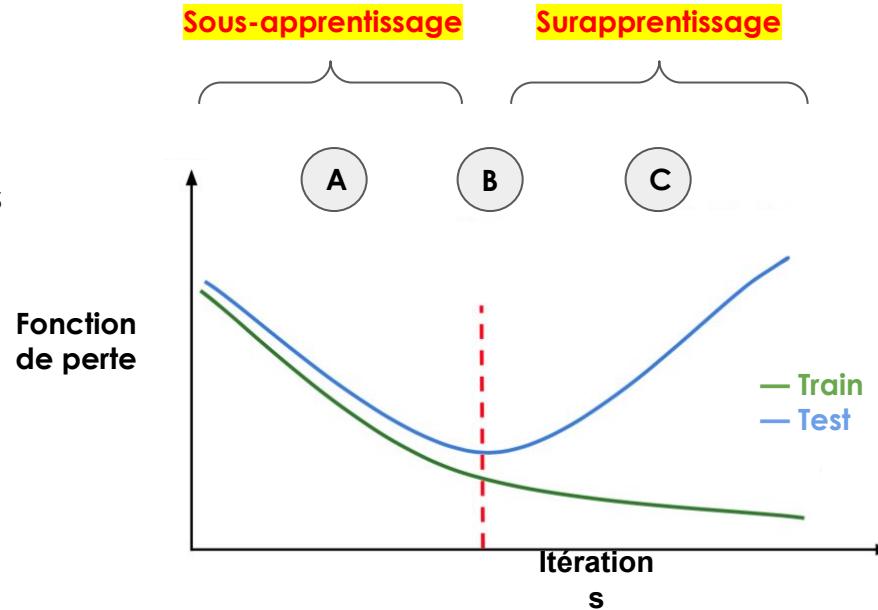


...

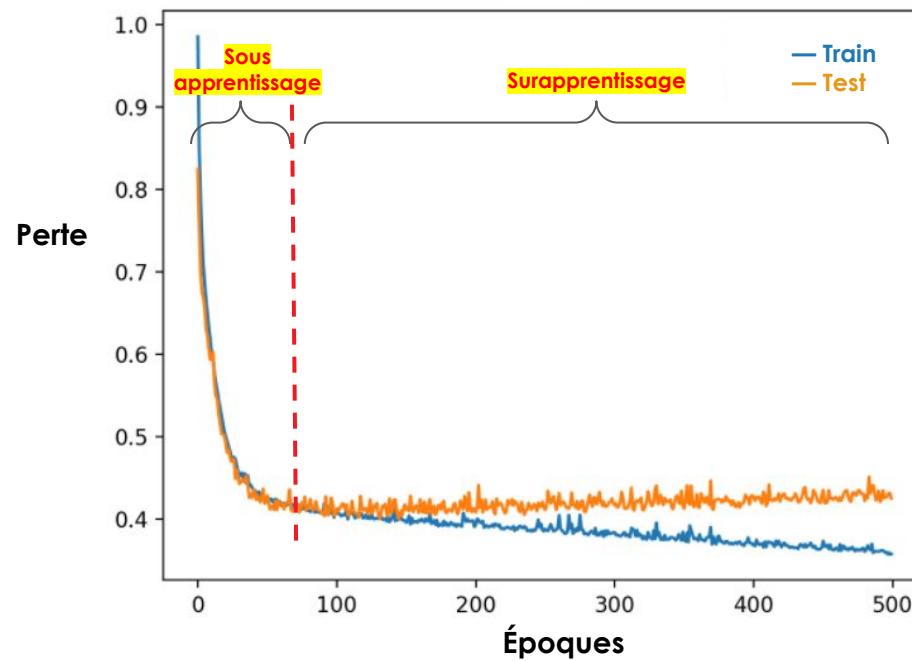
# Évolution générale d'une fonction de perte

**Un entraînement typique. Trois phases:**

- Modèle peu performant avec les deux ensembles de données, mais s'améliore lentement.
- Meilleur compromis entre mémorisation et compréhension des données.
- Le modèle mémorise les données d'entraînement, mais ne peut plus prédire correctement celles de test.



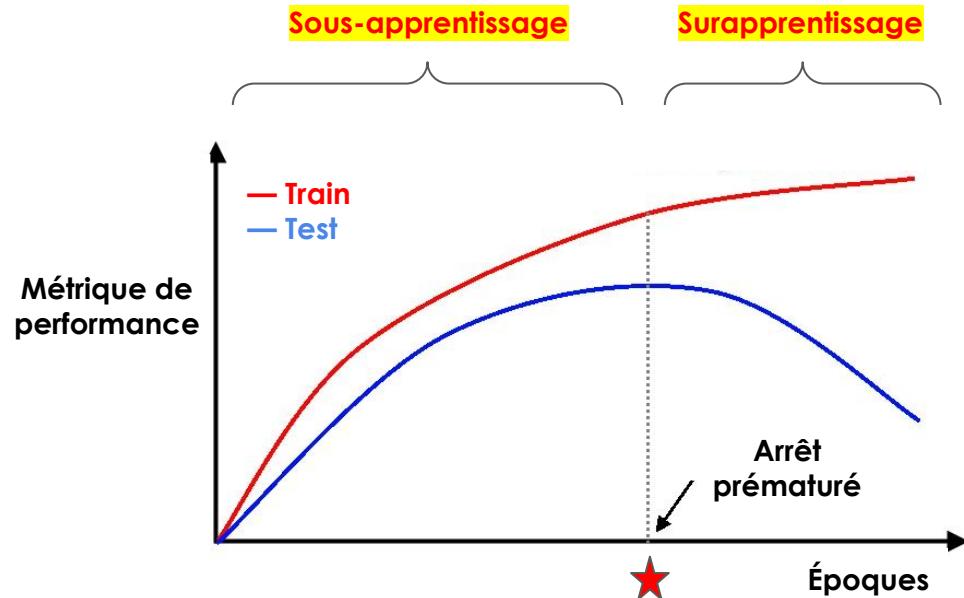
# Exemple de courbe de la fonction de perte pour un entraînement sans histoire



# Évolution générale d'une métrique de performance

Il faut cesser l'entraînement lorsque le meilleur compromis est atteint entre la mémorisation et la compréhension des données. ★

C'est la méthode de l'arrêt prématué (early stopping).

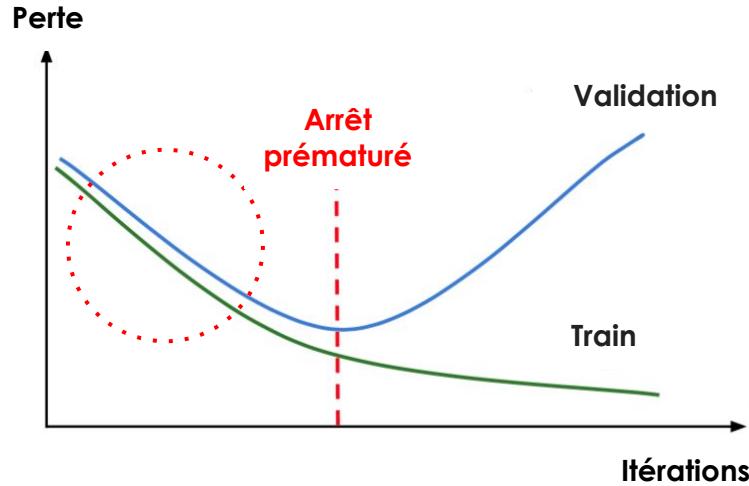
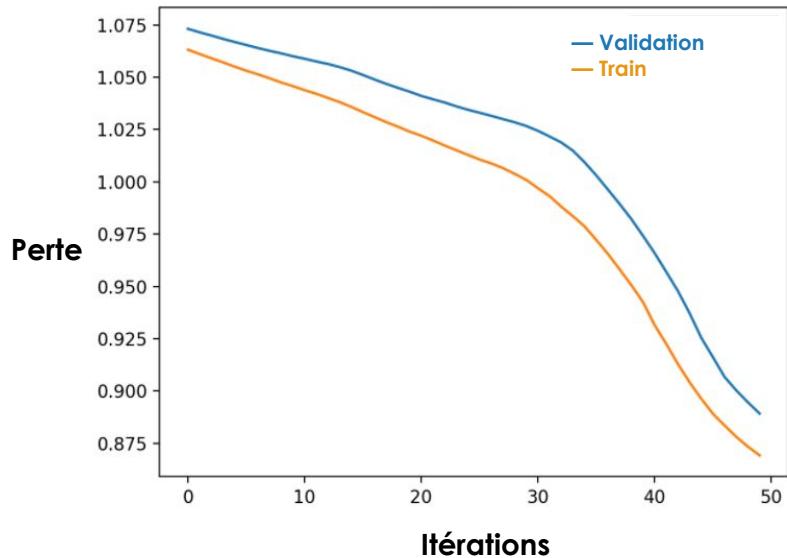


# Différents symptômes et leurs solutions



...

# Entrainement trop court



Les courbes d'entraînement et de test ont un comportement similaire et des performances médiocres.  
**Solution: entraîner plus longtemps!**

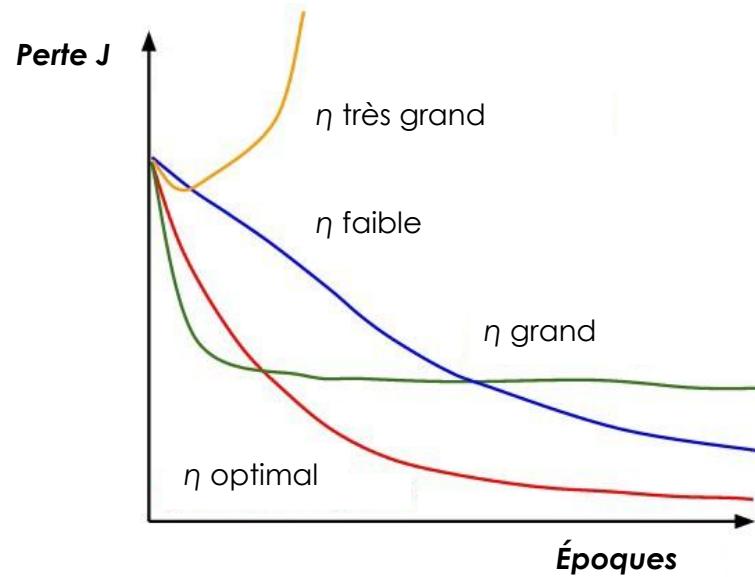
# Taux d'apprentissage $\eta$ mal ajusté

Dans l'algorithme de descente du gradient, les paramètres  $\Theta$  d'un modèle sont mis à jour selon

$$\Theta = \Theta - \eta^* \nabla_{\Theta} J(\Theta)$$

où  $\eta$  est le taux d'apprentissage et  $J(\Theta)$  est la fonction de perte.

**Le taux d'apprentissage est un des hyperparamètres les plus importants lors de l'entraînement d'un modèle.**



# Ensemble d'entraînement trop petit

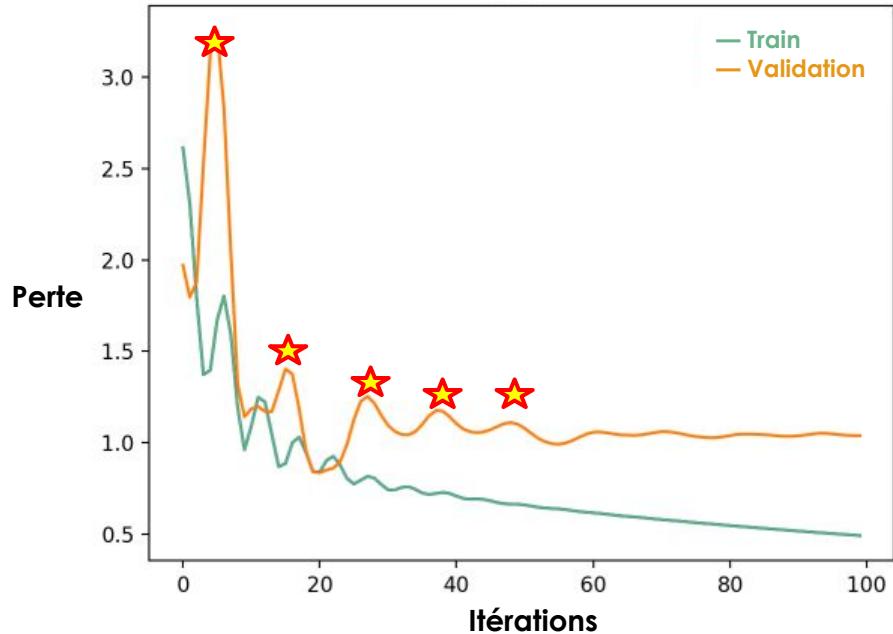
Ensemble d'entraînement trop petit:

- pas assez varié en termes de données.

Le modèle entraîné reconnaît peu de données de l'ensemble de test; Il ne peut faire de bonnes prédictions

Solutions:

- Augmenter le nombre de données
- Uniformiser les deux ensembles
- Mélanger régulièrement les données d'entraînement afin d'éviter des erreurs périodiques. 



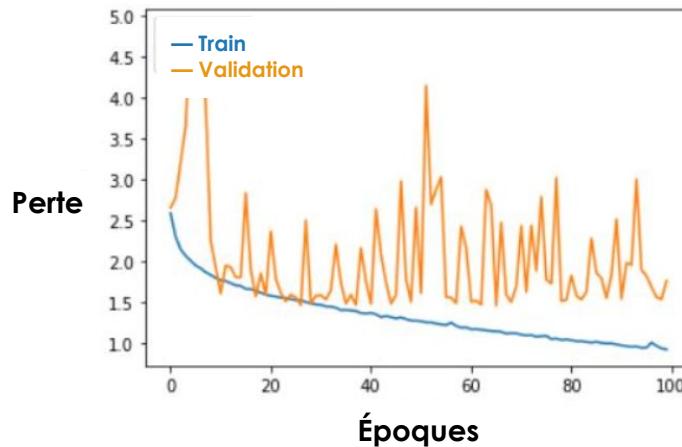
# Ensembles d'entraînement et de test trop différents

Le modèle entraîné reconnaît peu de données dans l'ensemble de test; Il ne peut faire de bonnes prédictions.

Exemple: L'ensemble d'entraînement contient de vieilles données et celui de test, de nouvelles données.

Solutions:

- Uniformiser les jeux de données avant de les séparer en deux ensembles d'entraînement et de test.



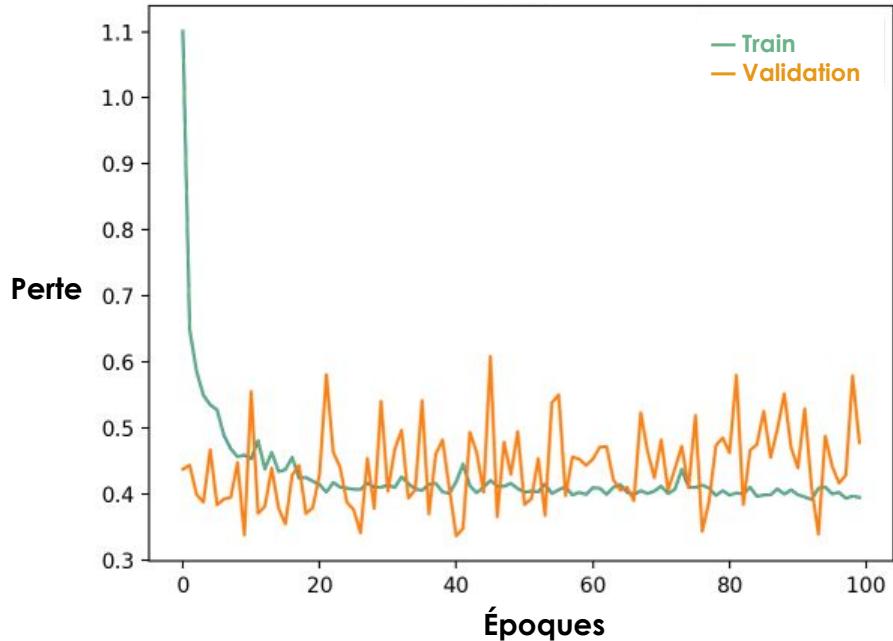
# Ensemble de test trop petit

Les statistiques basées sur un ensemble de test trop petit sont bruitées.

Les deux ensembles risquent d'être stratifiés différemment.

Solutions:

- Augmenter le nombre de données
- Uniformiser les deux ensembles



# Ensemble de test trop facile

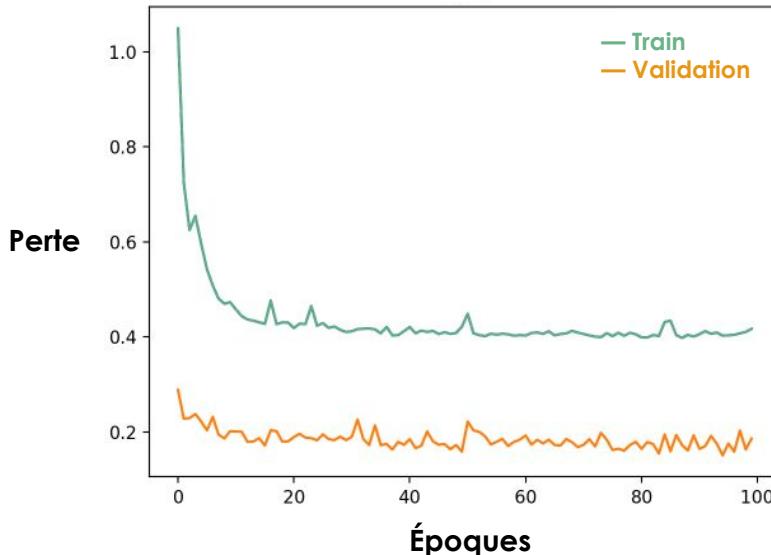
Il arrive que l'ensemble de test contient les données les plus fréquentes.

Elles sont faciles à traiter par le modèle et peu d'erreurs sont observées.

Les performances en test sont meilleures que celles en entraînement!

Solutions:

- Augmenter le nombre de données
- Uniformiser les deux ensembles



# Questions:

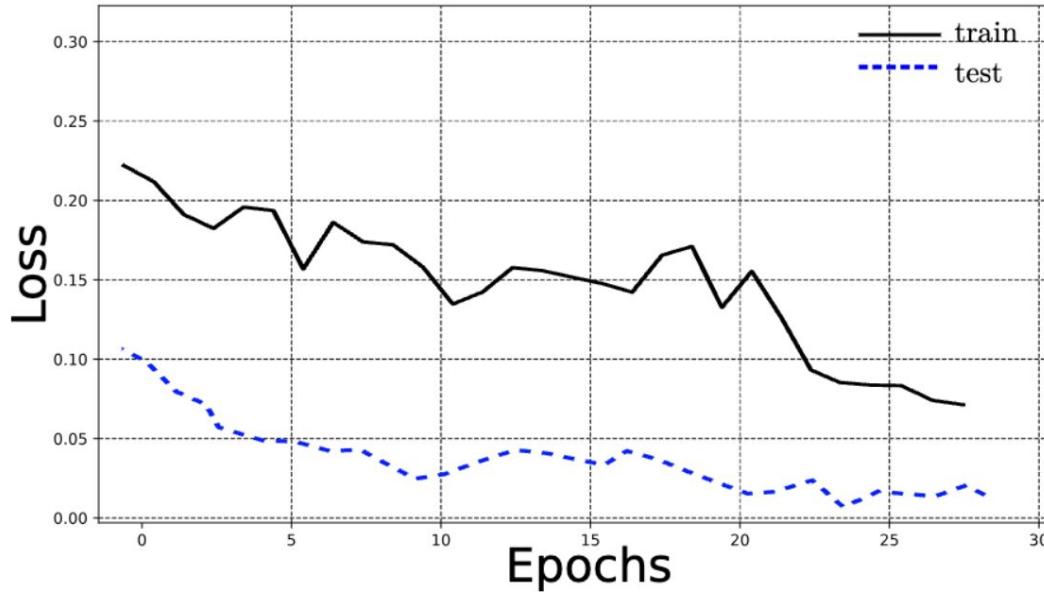
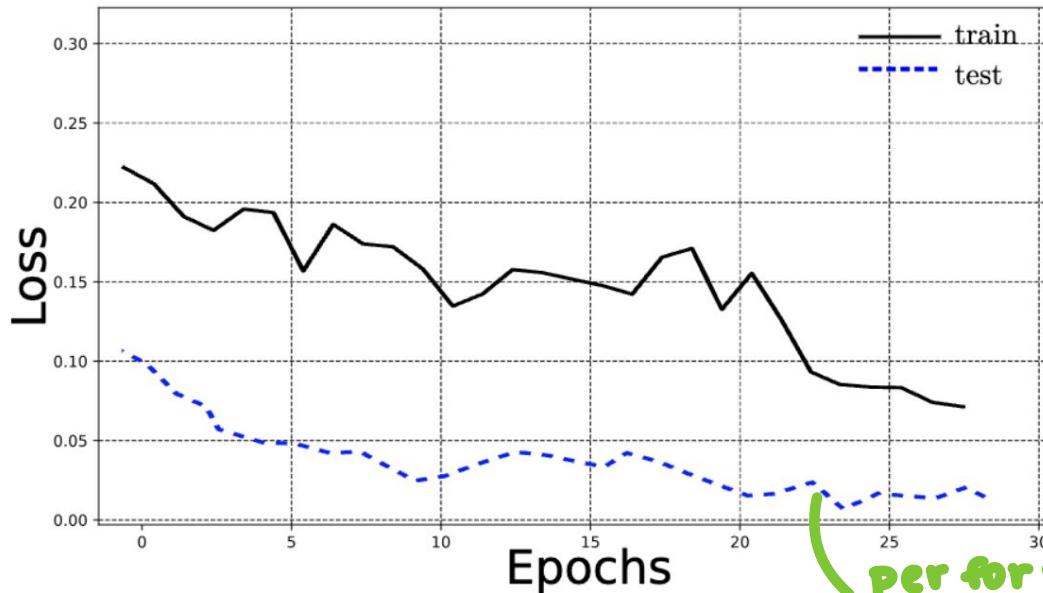


Figure 6: Sous-apprentissage :  ; Sur-apprentissage :  ; Ensemble de test trop petit :  ; Ensemble d'entraînement trop petit :  Ensembles d'entraînement et de test différents :  Ensemble de test trop facile :

# Questions:



performance en  
test meilleur

Figure 6: Sous-apprentissage :  ; Sur-apprentissage :  ; Ensemble de test trop petit :  ; Ensemble d'entraînement trop petit :  ; Ensembles d'entraînement et de test différents :  Ensemble de test trop facile :  x

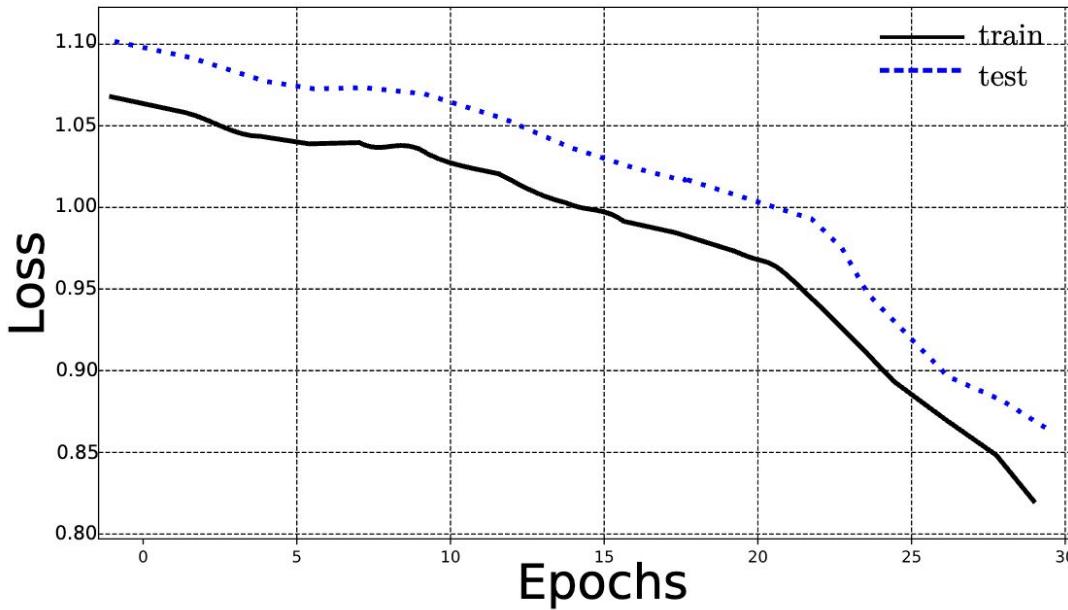


Figure 8: Sous-apprentissage :  ; Sur-apprentissage :  ; Ensemble de test trop petit :



; Ensemble d'entraînement trop petit :

Ensembles d'entraînement et de test différents :



Ensemble de test trop facile :

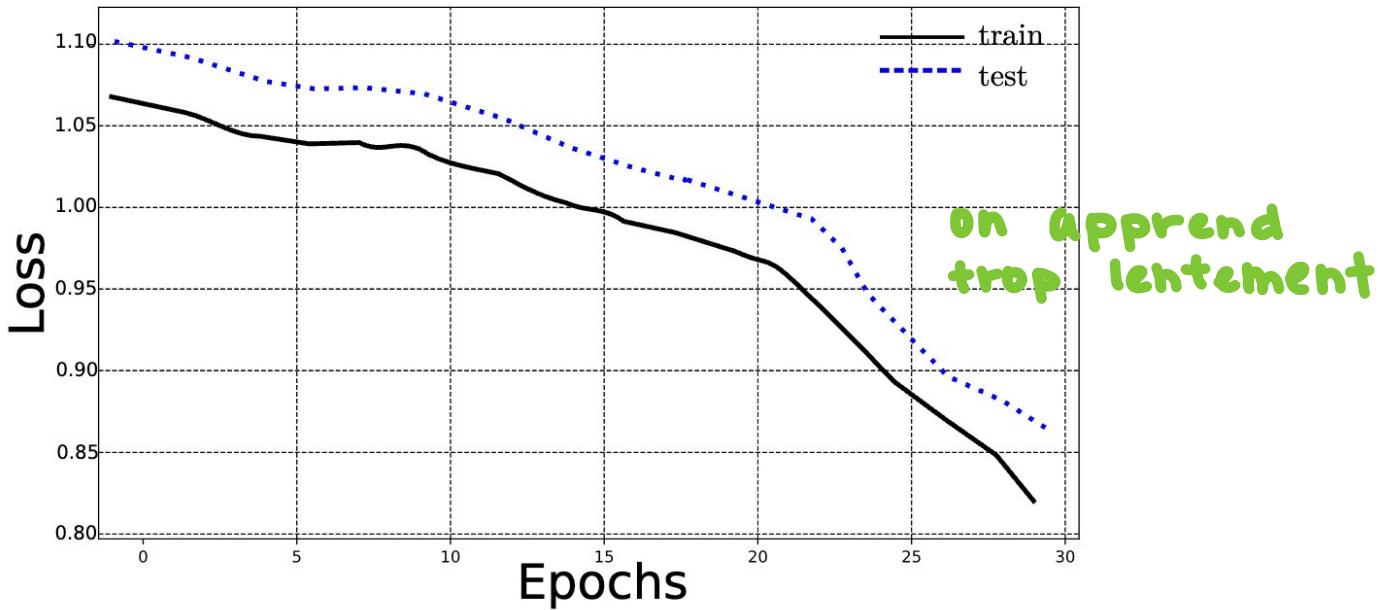


Figure 8: Sous-apprentissage :  ; Sur-apprentissage :  ; Ensemble de test trop petit :



; Ensemble d'entraînement trop petit :

Ensembles d'entraînement et de test différents :



Ensemble de test trop facile :

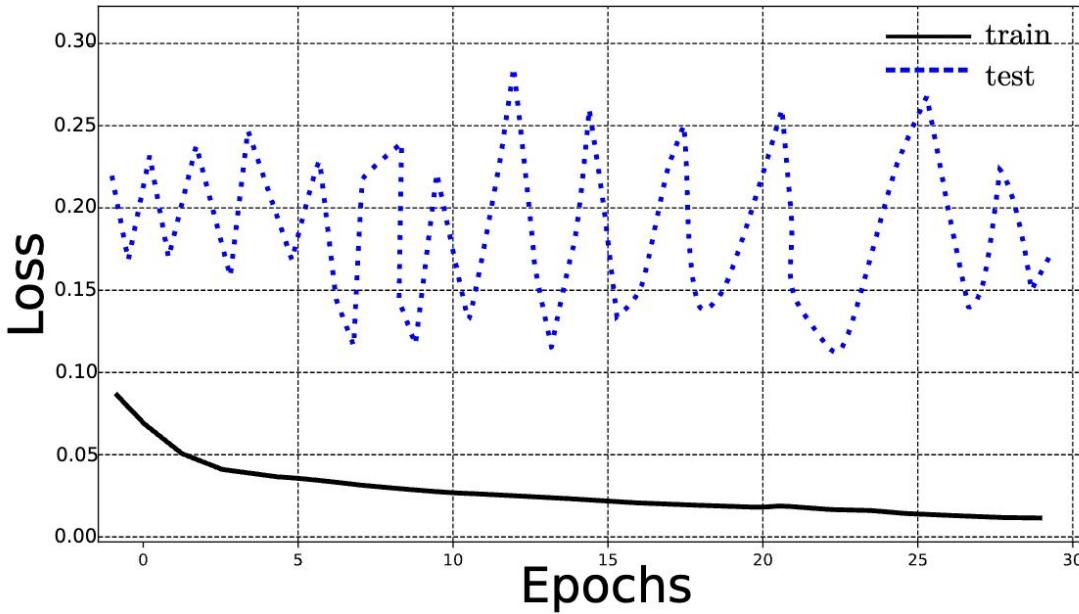
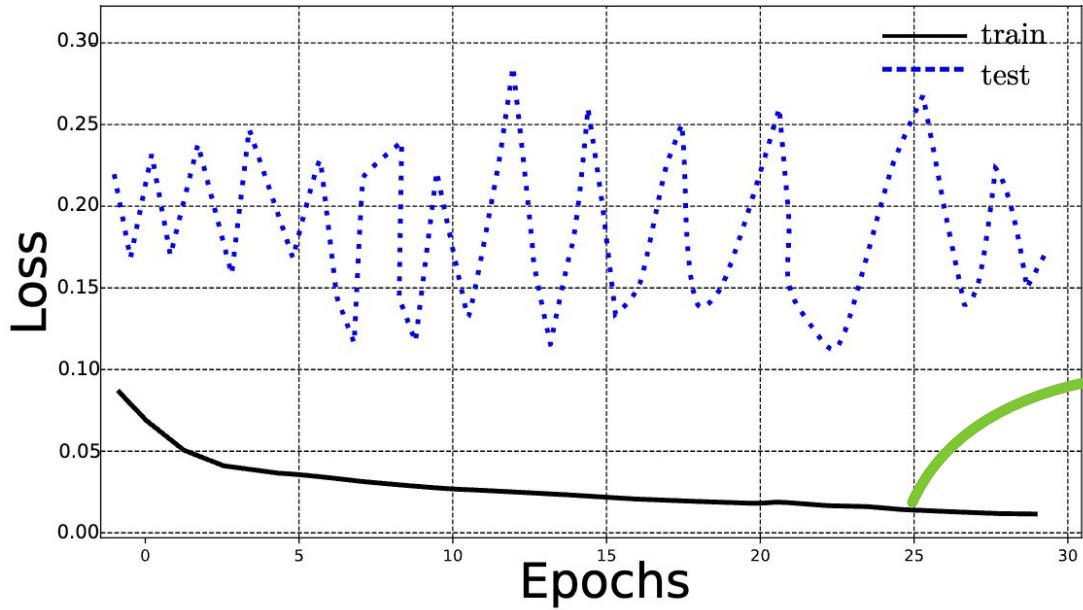


Figure 9: Sous-apprentissage :  ; Sur-apprentissage :  ; Ensemble de test trop petit :  ; Ensemble d'entraînement trop petit :  Ensembles d'entraînement et de test différents :   
Ensemble de test trop facile :



reconnait  
peu de  
données de  
l'ens. test

Figure 9: Sous-apprentissage :  ; Sur-apprentissage :  ; Ensemble de test trop petit :  ; Ensemble d'entraînement trop petit :  ; Ensembles d'entraînement et de test différents :  x Ensemble de test trop facile :

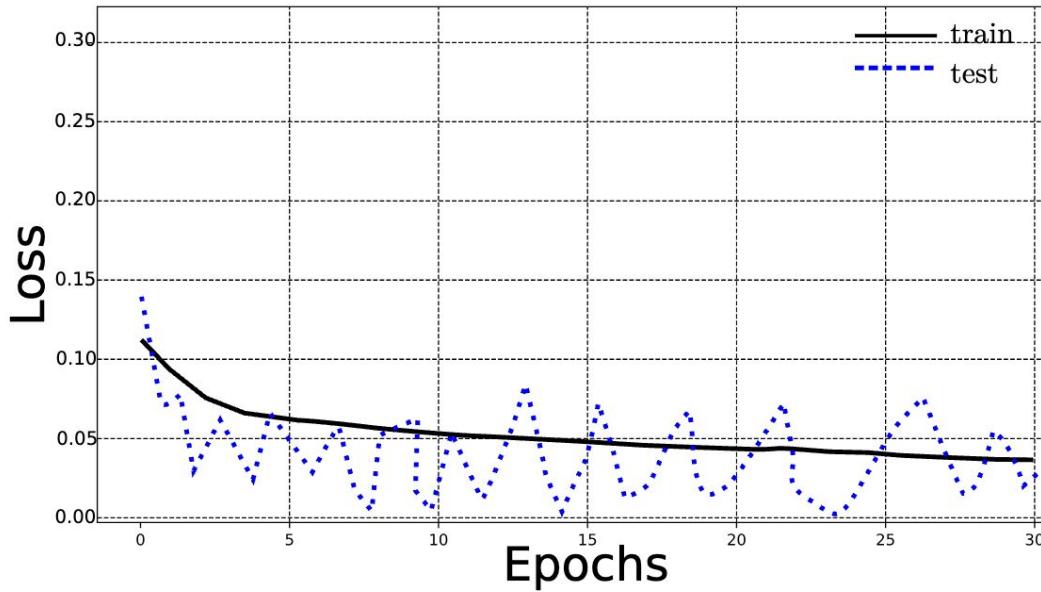


Figure 10: Sous-apprentissage :  ; Sur-apprentissage :  ; Ensemble de test trop petit :   
; Ensemble d'entraînement trop petit :  Ensemble d'entraînement et de test différents :   
Ensemble de test trop facile :

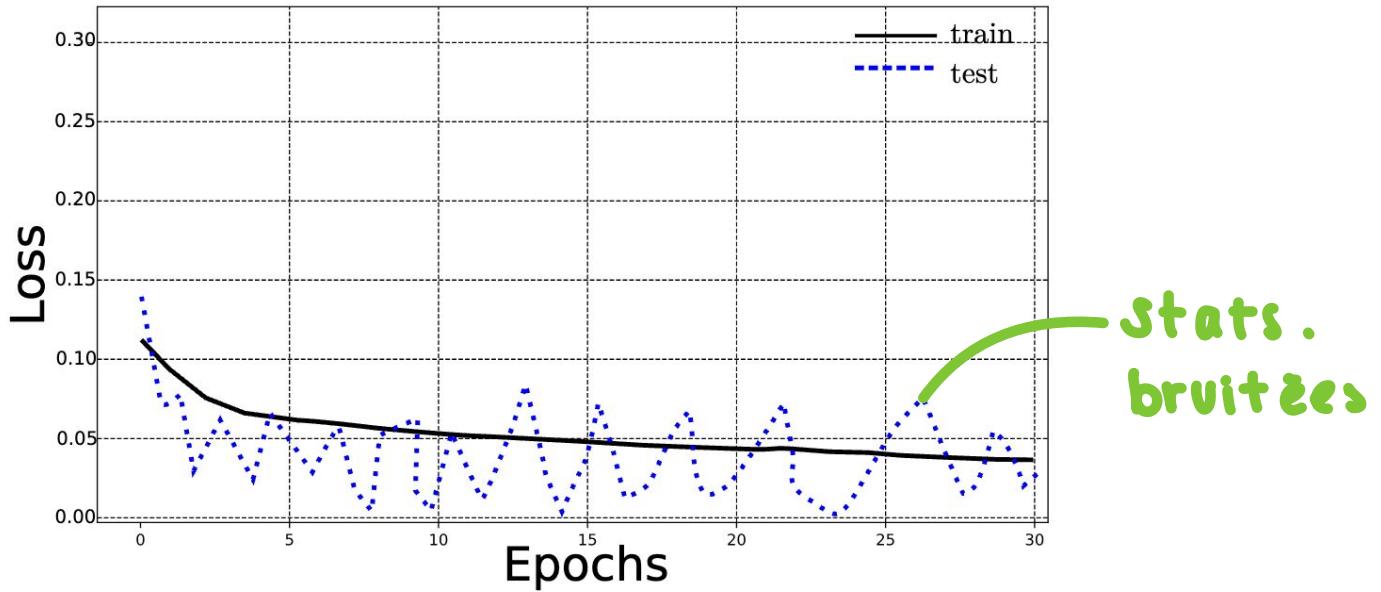


Figure 10: Sous-apprentissage :  ; Sur-apprentissage :  ; Ensemble de test trop petit :

x

; Ensemble d'entraînement trop petit :  Ensembles d'entraînement et de test différents :

Ensemble de test trop facile :

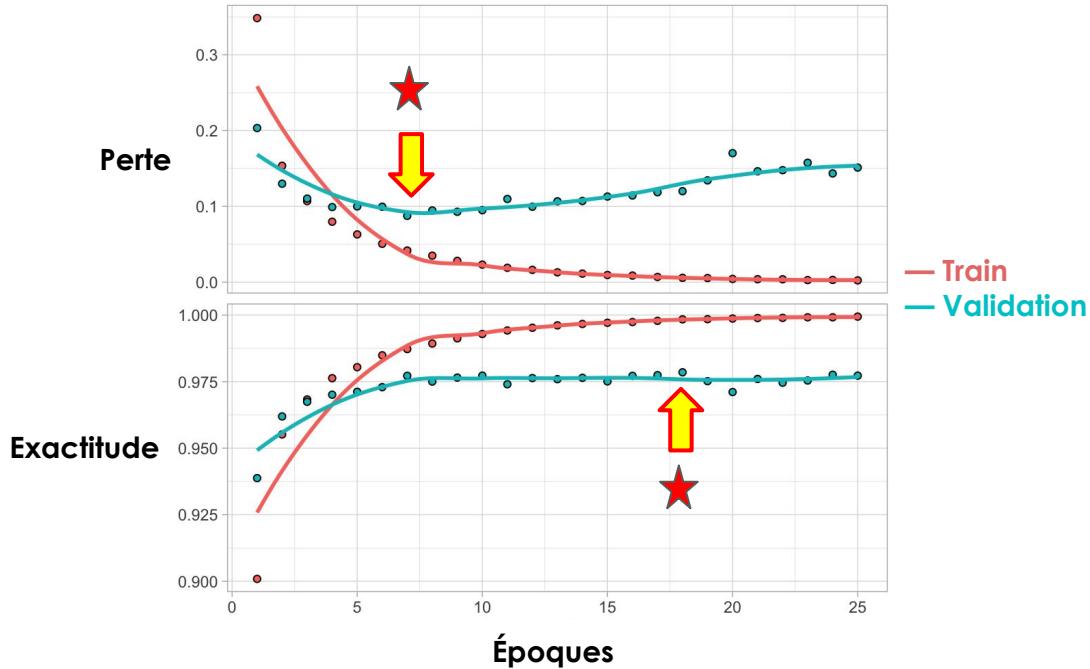
# Choisir entre la métrique ou la fonction perte



...

# Doit-on se fier à la courbe de perte ou à celle de la métrique de performance pour décider quand s'arrêter?

- Fonction de perte:
  - ne capture pas toujours ce qui est le plus important.
  - fonction dérivable qui a un comportement "similaire" à la métrique.
  - ne sert qu'à l'optimisation.
- Métrique de performance:
  - définit au mieux les performances du modèle.
  - dépend de la manière dont on a l'intention d'utiliser le modèle.



# Les métriques de performance

# Qu'est ce qu'une métrique de performance ?

- Une façon de quantifier la performance d'un modèle d'apprentissage automatique.
- Remarque : **fonction de perte ≠ métrique de performance**

- La fonction de perte est utilisée lors d'un entraînement.
- La métrique de performance est utilisée à la fin pour évaluer le modèle final.

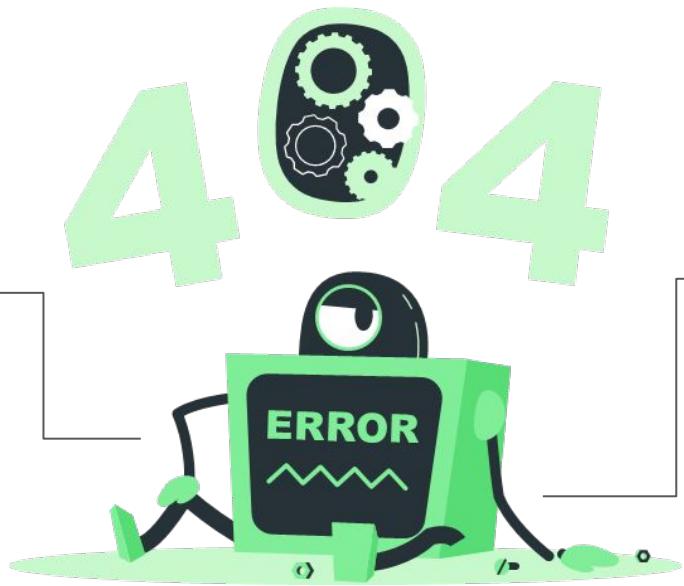
# Différentes métriques

## Régression

- R-carré
- R-carré ajusté
- MAE
- MSE
- RMSE
- ...

## Classification

- Exactitude
- Précision
- Sensibilité
- Spécificité
- ROC/AUC
- ...





# Régression

...

# Coefficient de détermination

## R-carré

- Il quantifie la proximité des données au modèle ajusté.

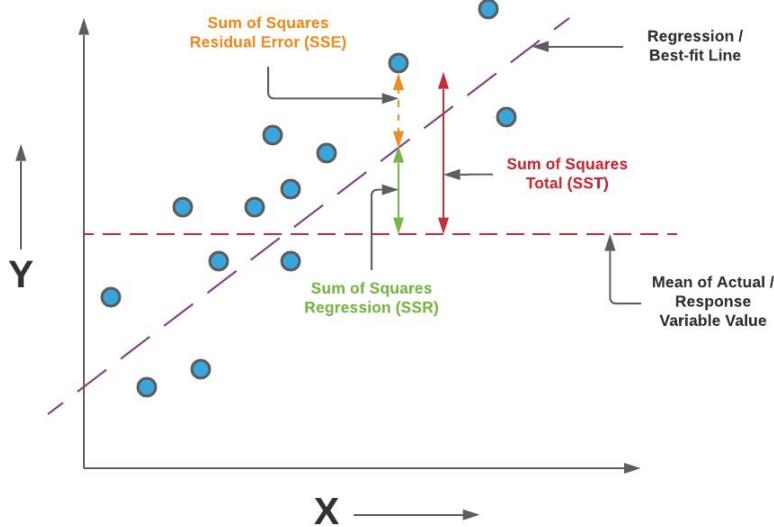
$$R^2 = \frac{\text{Variance expliquée par le modèle}}{\text{Variance des données}}$$

- Plus la valeur  $R^2$  est grande et s'approche 1, plus le modèle explique la variabilité des données.

# Coefficient de détermination R-carré

Méthode de calcul:

$$R^2 = 1 - \frac{SSE}{SST} = 1 - \frac{\sum_{t=1}^N (\hat{y}^{(t)} - y^{(t)})^2}{\sum_{t=1}^N (y^{(t)} - \bar{y})^2}$$



Exemple pour un modèle linéaire

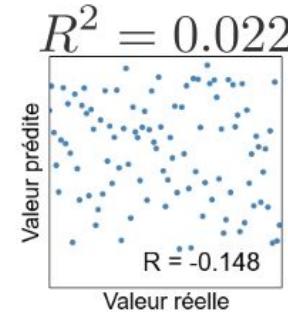
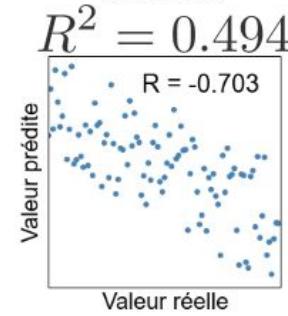
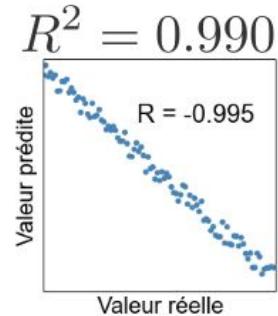
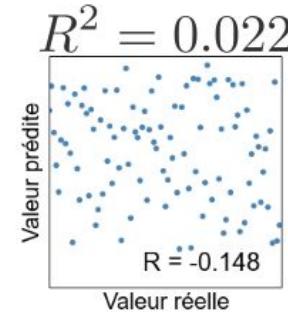
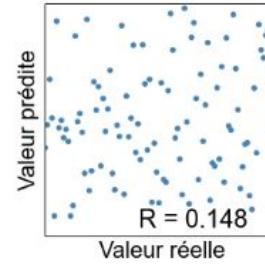
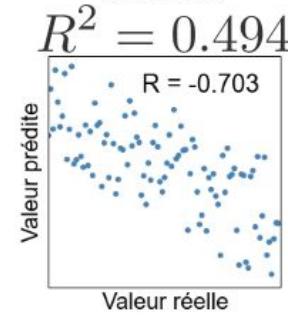
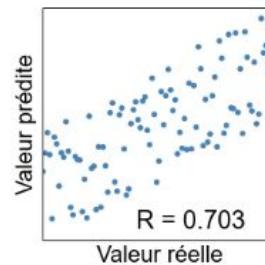
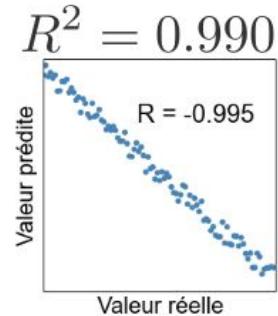
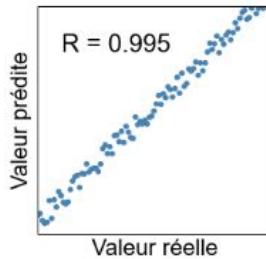
# Exemples

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

où R est le coefficient de corrélation entre deux séries de données.

Plus R-carré est grand, meilleure est l'adéquation entre

- Les prédictions d'un modèle et des observations



# R-carré ajusté

- N.B. Le R-carré tend à **surévaluer** la qualité de la régression linéaire.
- Le R-carré **ajusté** pénalise pour l'ajout de caractéristiques  $x$  supplémentaires qui n'améliorent pas un modèle.

$$R^2 = 1 - \frac{SSE}{SST}$$
$$R^2_{\text{ajusté}} = 1 - \left( \frac{N-1}{N-p} \right) \frac{SSE}{SST}$$

$R^2_{\text{ajusté}} \leq R^2$



N: Nombre de données

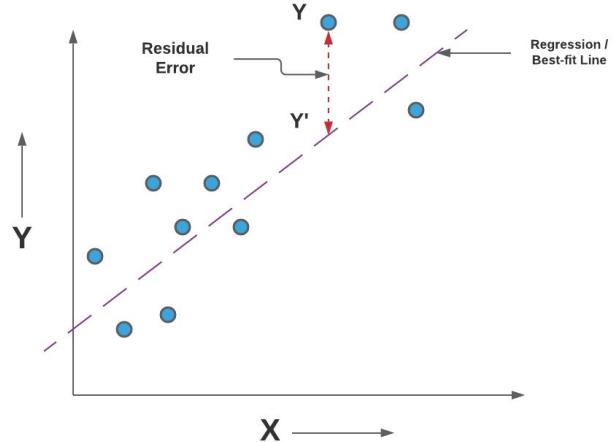
p: Nombre de paramètres d'un modèle



# Erreur quadratique moyenne (MSE)

- MSE pour *Mean Squared Error*.
- C'est la variance des erreurs du modèle.
- Elle est **sensible** aux écarts importants entre les prévisions d'un modèle et les observations.

$$MSE = \frac{1}{N} \sum_{t=1}^N \underbrace{\left( y^{(t)} - \hat{y}^{(t)} \right)^2}_{\text{Erreurs résiduelles } e}$$



# Racine carrée de l'erreur quadratique moyenne (RMSE)

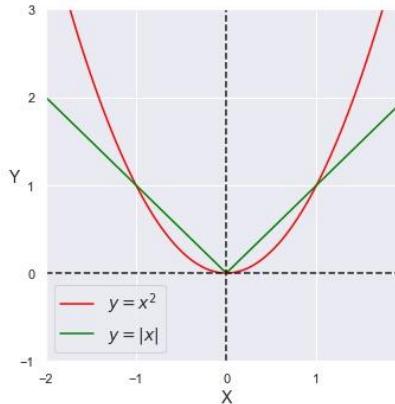
- RMSE pour *Root Mean Squared Error*.
- C'est l'écart-type des erreurs du modèle.
- Elle est **sensible** aux écarts importants entre les prévisions d'un modèle et les observations.

$$RMSE = \sqrt{\frac{1}{N} \sum_{t=1}^N \underbrace{\left( y^{(t)} - \hat{y}^{(t)} \right)^2}_{\text{Erreur résiduelle } e}}$$

# L'erreur absolue moyenne (MAE)

- MAE pour *Mean Absolute Error*.
- Elle est **robuste** aux écarts importants entre les prévisions d'un modèle et les observations.
- Raison: la valeur absolue de l'erreur résiduelle a un effet moindre sur la MAE que le carré de l'erreur sur la MSE et la RMSE.

$$MAE = \frac{1}{N} \sum_{t=1}^N \underbrace{|y^{(t)} - \hat{y}^{(t)}|}_{\text{Erreur résiduelle}}$$



# Quelle métrique choisir ?

Généralement, on rapporte le R-carré et une mesure d'erreur, par ex. la RMSE.

- Le R-carré exprime le pourcentage de la variance dans les données qui est expliquée par un modèle.
- La RMSE donne l'écart-type des erreurs de prédiction du modèle.



# Classification

...

# Matrice de confusion

- Matrice de confusion : explication des erreurs effectuées lors d'une classification binaire.
- **Ce n'est pas une métrique**; elle aide à calculer plusieurs métriques.



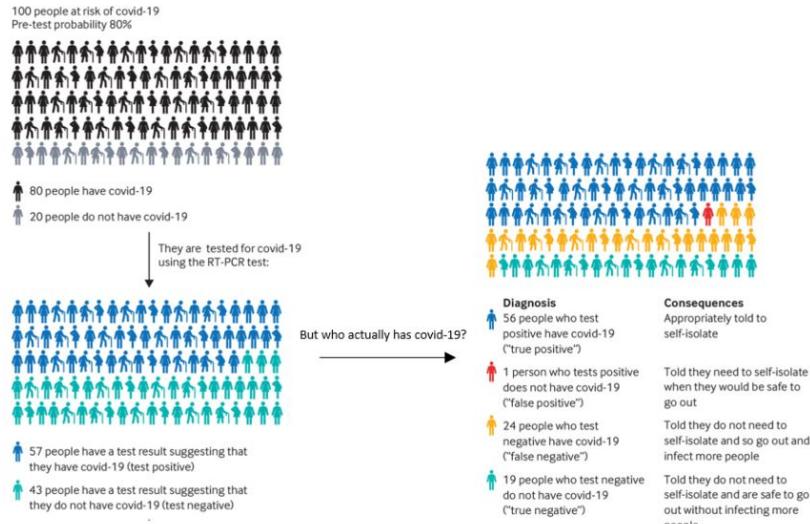
		Classe prédictive	
		Positif ( $\hat{Y} = 1$ )	Négatif ( $\hat{Y} = 0$ )
Vraie classe	Positif ( $Y = 1$ )	Vrais positifs (VP)	Faux négatifs (FN)
	Négatif ( $Y = 0$ )	Faux positifs (FP)	Vrais négatifs (VN)

Nombre de bonnes classifications: VP+VN

Nombre d'erreurs: FN+FP

# Exemple de matrice de confusion pour des sujets atteints de la Covid-19

- On suppose que 80% des sujets allant se faire tester sont atteints de la maladie.
- Les chiffres sont ramenés à une population de 100 individus.



		Classe prédictive	
		Positif ( $\hat{Y} = 1$ )	Négatif ( $\hat{Y} = 0$ )
Vraie classe	Positif ( $Y = 1$ )	Vrais positifs (VP) <b>56</b>	Faux négatifs (FN) <b>24</b>
	Négatif ( $Y = 0$ )	Faux positifs (FP) <b>1</b>	Vrais négatifs (VN) <b>19</b>

Matrice de confusion correspondante

# Exactitude (Accuracy)

- C'est la métrique la plus simple et la plus intuitive de toutes.
- Elle indique la fraction de bonnes prédictions, parmi toutes les prédictions effectuées.
- Exemple: parmi les 100 personnes testées, 75 ont reçu un diagnostic correct.

		Classe prédictée	
		Positif ( $\hat{Y} = 1$ )	Négatif ( $\hat{Y} = 0$ )
Vraie classe	Positif ( $Y = 1$ )	Vrais positifs (VP) 56	Faux négatifs (FN) 24
	Négatif ( $Y = 0$ )	Faux positifs (FP) 1	Vrais négatifs (VN) 19

$$\begin{aligned}\text{Exactitude} &= \frac{\text{Nombre de bonnes classifications}}{\text{Nombre total d'exemples}} \\ &= \frac{56 + 19}{56 + 19 + 1 + 24} \\ &= 75\%\end{aligned}$$

# Exactitude (Accuracy)

- Pour certaines maladies rares, il arrive qu'on mesure un exactitude de l'ordre de 95%.
- Est-ce réellement une bonne performance en détection ?



# Exactitude (Accuracy)

- Pour certaines maladies rares, il arrive qu'on mesure un exactitude de l'ordre de 95%.
- Est-ce réellement une bonne performance en détection ?
- **Dans le cas de données non-équilibrées, l'exactitude n'est pas une métrique utile.**



- Exemple pour une maladie rare
  - 1000 sujets testés: 950 sains et 50 malades
  - Un test trivial, qui indique toujours un sujet sain, a raison 950 fois sur 1000, soit une exactitude de 95%!

# Rappel (*Recall*)

- Rappel (ou sensibilité) : Probabilité d'une prévision positive pour un cas positif.
- En médecine, ça correspond à la probabilité d'identifier les patients malades.
- Exemple: 80 personnes ayant la Covid-19 ont été testées et 56 ont reçu un diagnostic positif.

		Classe prédictive	
		Positif ( $\hat{Y} = 1$ )	Négatif ( $\hat{Y} = 0$ )
Vraie classe	Positif ( $Y = 1$ )	Vrais positifs (VP) <b>56</b>	Faux négatifs (FN) 24
	Négatif ( $Y = 0$ )	Faux positifs (FP) 1	Vrais négatifs (VN) <b>19</b>

$$\begin{aligned}\text{Rappel} &= \frac{VP}{VP + FN} \\ &= \frac{56}{56 + 24} \\ &= 70\%\end{aligned}$$



# Précision

- Précision : Probabilité d'une prévision positive parmi les points que l'on a prédits positifs.
- En médecine, ça correspond à la probabilité que les patients diagnostiqués comme étant malades le soient véritablement.
- Exemple: 57 personnes ont reçu un diagnostic positif à la Covid-19 alors que seulement 56 l'avaient.

		Classe prédictive	
		Positif ( $\hat{Y} = 1$ )	Négatif ( $\hat{Y} = 0$ )
Vraie classe	Positif ( $Y = 1$ )	Vrais positifs (VP) <b>56</b>	Faux négatifs (FN) <b>24</b>
	Négatif ( $Y = 0$ )	Faux positifs (FP) <b>1</b>	Vrais négatifs (VN) <b>19</b>

$$\begin{aligned}\text{Précision} &= \frac{VP}{VP + FP} \\ &= \frac{56}{56 + 1} \\ &= 98\%\end{aligned}$$



# Précision ou Rappel; lequel choisir?

Cela dépend des besoins de l'enquêteur :

**On veut minimiser les faux positifs => Précision**

- Exemple: circuits intégrés faussement identifiés comme défectueux sur une chaîne de montage.



**On veut minimiser les faux négatifs=> Rappel**

- Exemple: patients atteints de cancers non détectés.



# Spécificité

- Spécificité : Probabilité d'une prévision négative pour un cas négatif.
- En médecine, ça correspond à la probabilité de ne pas identifier comme malades les patients non malades.
- Exemple: 20 personnes n'ayant pas la Covid-19 ont été testées et 19 ont reçu un diagnostic négatif.



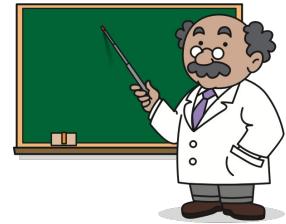
		Classe prédictive	
		Positif ( $\hat{Y} = 1$ )	Négatif ( $\hat{Y} = 0$ )
Vraie classe	Positif ( $Y = 1$ )	Vrais positifs (VP) <b>56</b>	Faux négatifs (FN) <b>24</b>
	Négatif ( $Y = 0$ )	Faux positifs (FP) <b>1</b>	Vrais négatifs (VN) <b>19</b>

$$\begin{aligned}\text{Spécificité} &= \frac{VN}{VN + FP} \\ &= \frac{19}{19 + 1} \\ &= 95\%\end{aligned}$$

# Taux de vrais et de faux positifs

- Taux de vrais positifs (*True Positive Rate*). C'est la même chose que la sensibilité ou le rappel.

$$\begin{aligned} \text{TVP} &= \frac{VP}{VP + FN} \\ &= \frac{56}{56 + 24} \\ &= 70\% \end{aligned}$$



- Taux de faux positifs (*False Positive Rate*).

$$\begin{aligned} \text{TFP} &= \frac{FP}{FP + VN} \\ &= \frac{1}{1 + 19} \\ &= 5\% \end{aligned}$$

		Classe prédictive	
		Positif ( $\hat{Y} = 1$ )	Négatif ( $\hat{Y} = 0$ )
Vraie classe	Positif ( $Y = 1$ )	Vrais positifs (VP) 56	Faux négatifs (FN) 24
	Négatif ( $Y = 0$ )	Faux positifs (FP) 1	Vrais négatifs (VN) 19

# Résumé des principales métriques d'intérêt

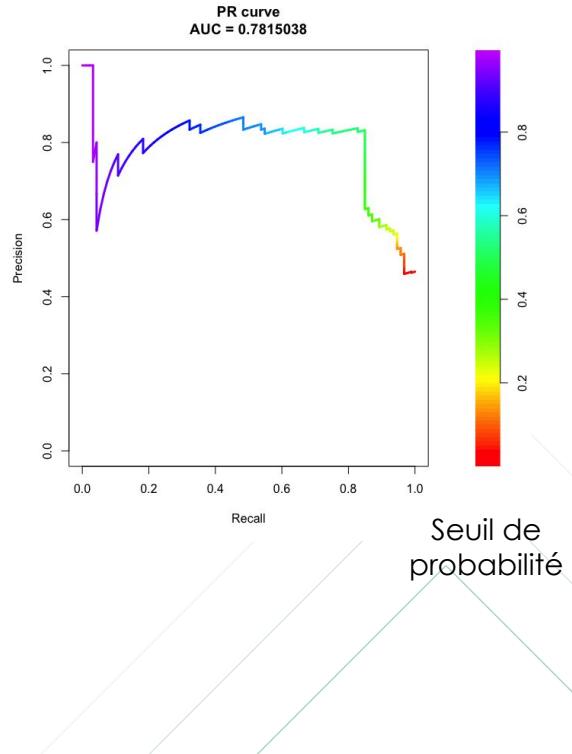
Mesure de précision	Définition	Formule
Vrais positifs (VP)	Nombre d'individus malades avec un test positif	VP
Vrais négatifs (VN)	Nombre d'individus non malades avec un test négatif	VN
Faux positifs (FP)	Nombre d'individus non malades avec un test positif	FP
Faux négatifs (FN)	Nombre d'individus malades avec un test négatif	FN
Taux de vrais positifs (TVP)	Sensibilité: proportion d'individus malades avec un test positif	$VP/(VP+FN)$
Taux de vrais négatifs (TVN)	Spécificité: proportion d'individus non malades avec un test négatif	$VN/(FP+VN)$
Taux de faux négatifs (TFN)	Proportion d'individus malades avec un test négatif	$FN/(VP+FN)$
Taux de faux positifs (TFP)	Proportion d'individus non malades avec un test positif	$FP/(FP+VN)$

# Les métriques de classification basées sur l'aire sous une courbe



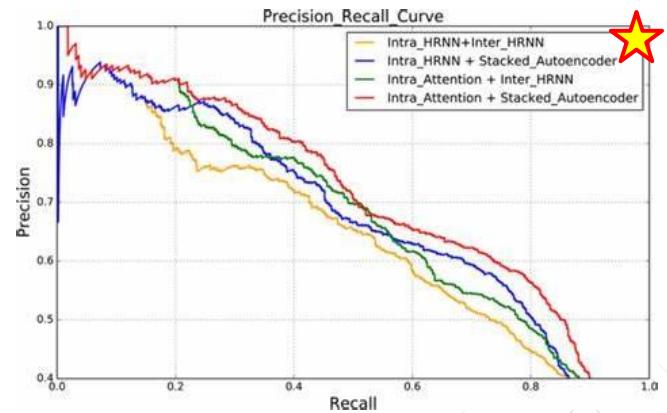
...

- Si on a un modèle donnant la probabilité d'appartenir à une classe d'intérêt, on peut ajuster le seuil et classer nos données.
- **Chaque seuil mène à une matrice de confusion.**
- Pour chacune, on calcule des métriques.
- On peut visualiser l'effet de chaque seuil à l'aide de graphiques affichant les métriques.
- **L'aire sous la courbe (*Area Under Curve* ou AUC) est une nouvelle métrique!**



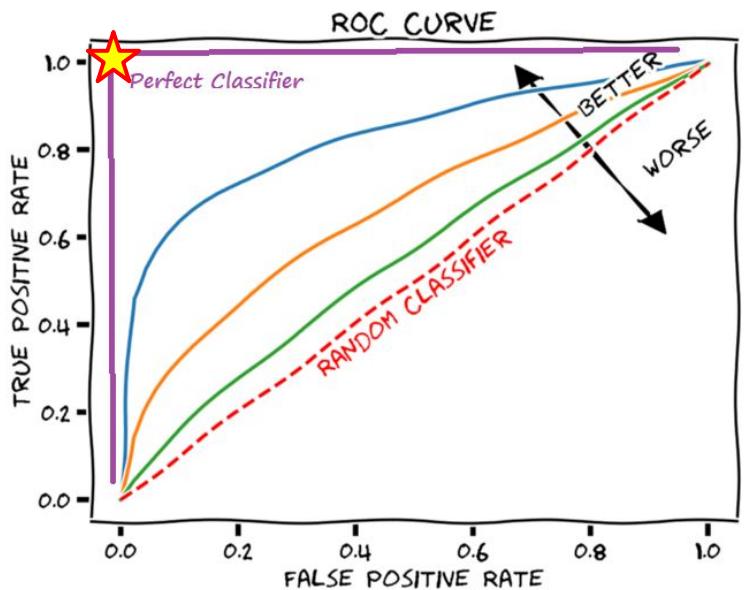
# La courbe Précision-Rappel

- Le classifieur avec la plus grande aire sous la courbe est le meilleur.
- Classifieur idéal:
  - Précision=1
  - Rappel=1



# La courbe ROC

- ROC: Receiver Operating Characteristic
- L'AUC représente la probabilité qu'un positif vs un négatif soit départagé correctement.
- Le classifieur avec la plus grande aire sous la courbe est le meilleur.
- **Cette métrique est la plus utilisée lors de débalancement des classes.**



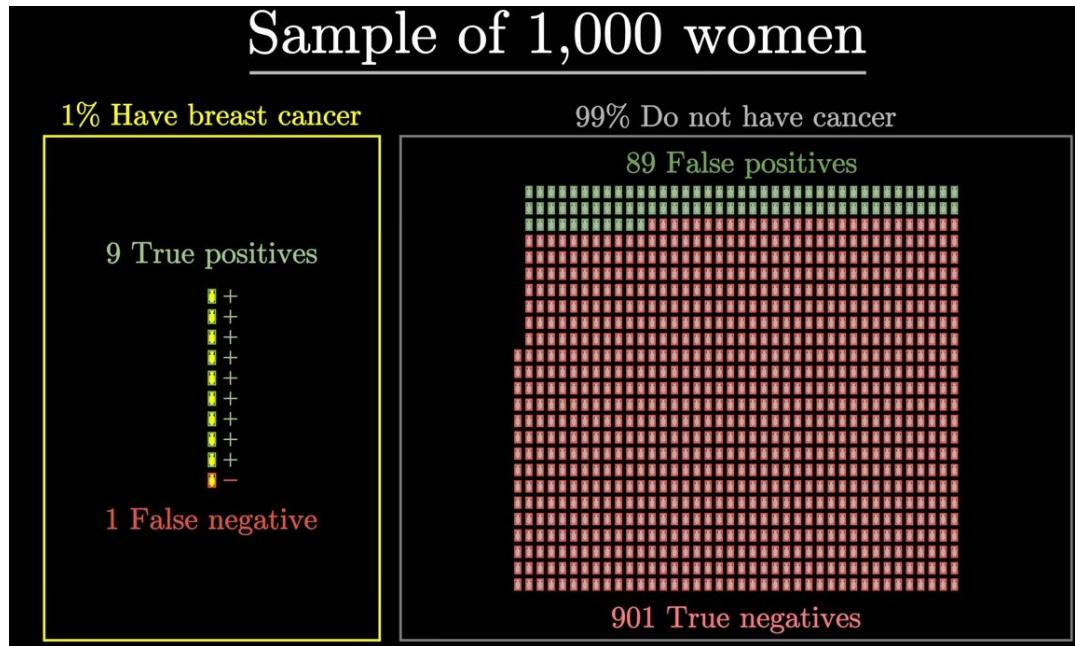
# F1 score

- Le score F1 calcule un compromis entre precision et rappel

One last time...

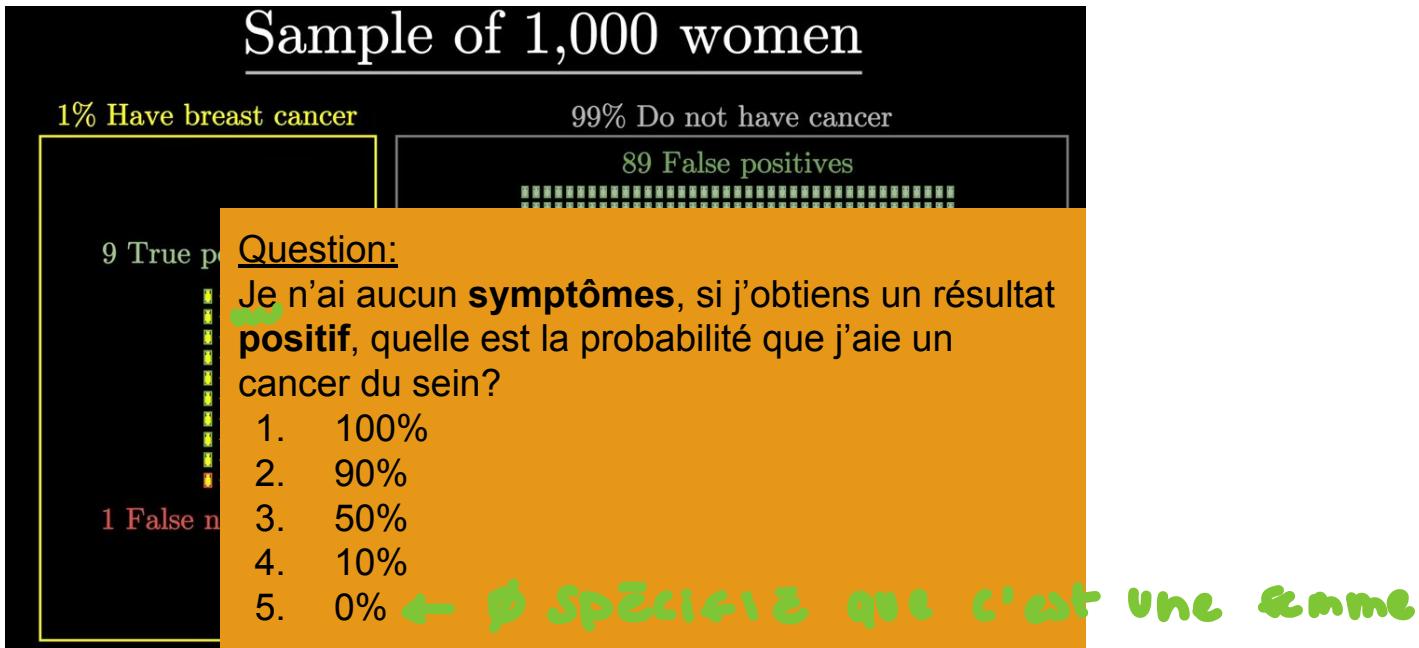
# Exemple de matrice de confusion pour des sujets atteints du cancer de la poitrine

- Le test a une sensibilité et une spécificité de ~90% (source: [3Blue1Brown](#))



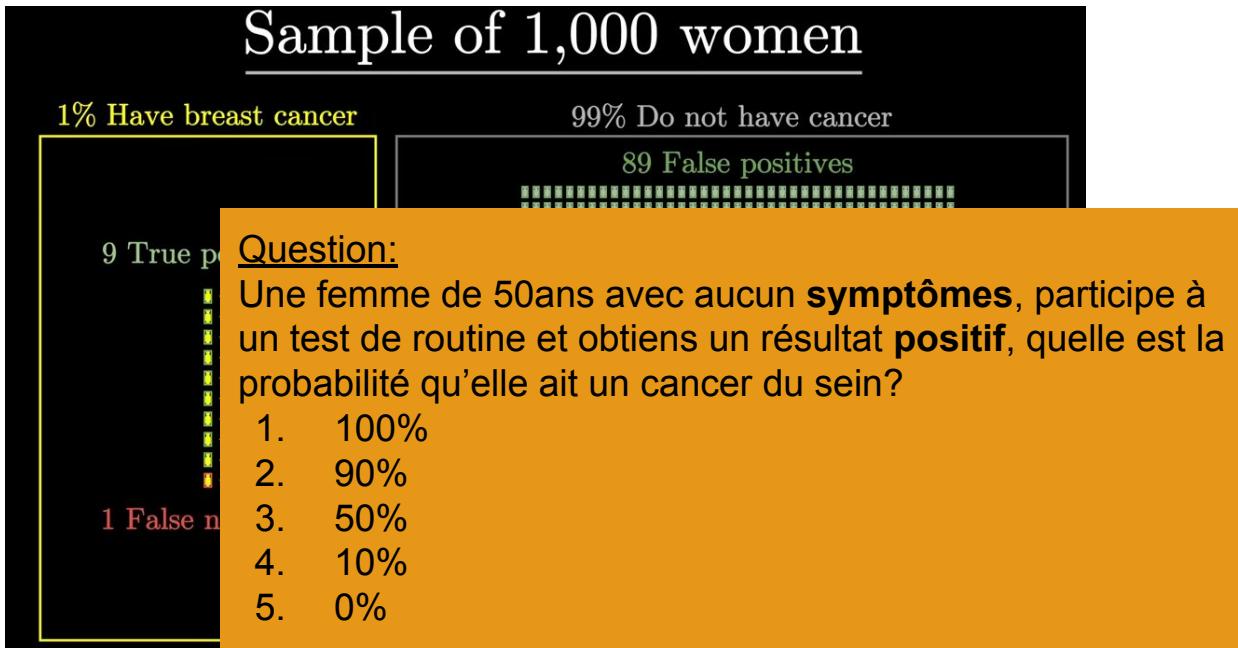
# Exemple de matrice de confusion pour des sujets atteints du cancer du sein

- Le test a une sensibilité et une spécificité de ~90%



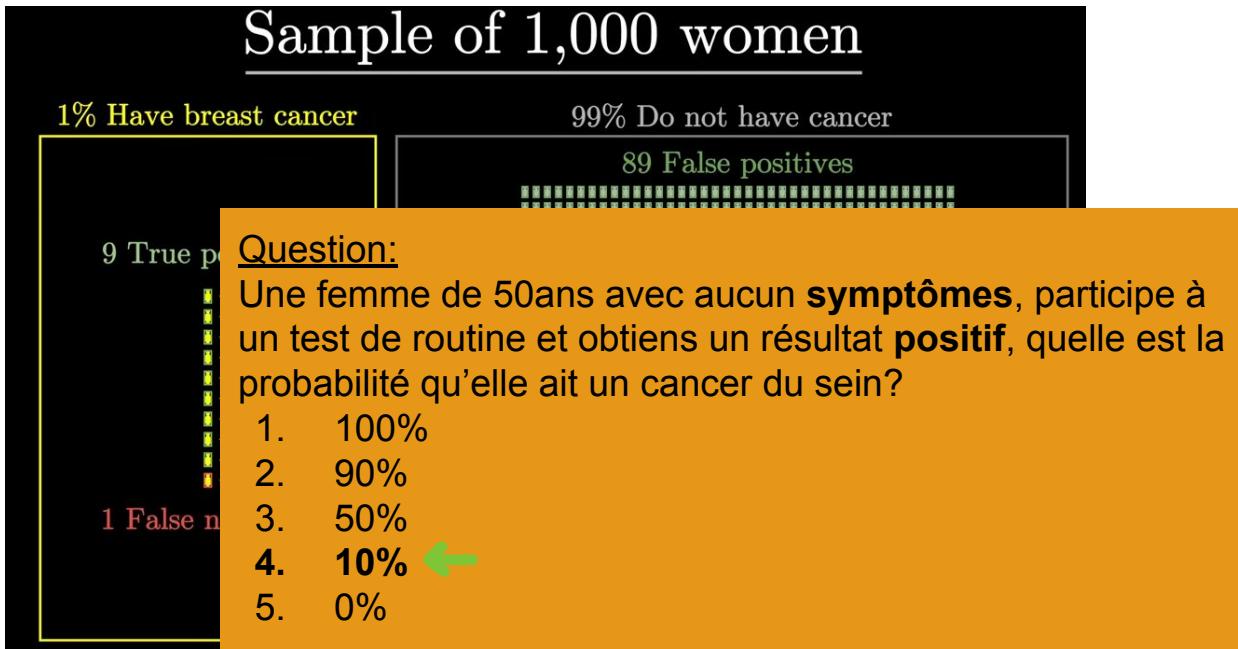
# Exemple de matrice de confusion pour des sujets atteints du cancer du sein

- Le test a une sensibilité et une spécificité de ~90%



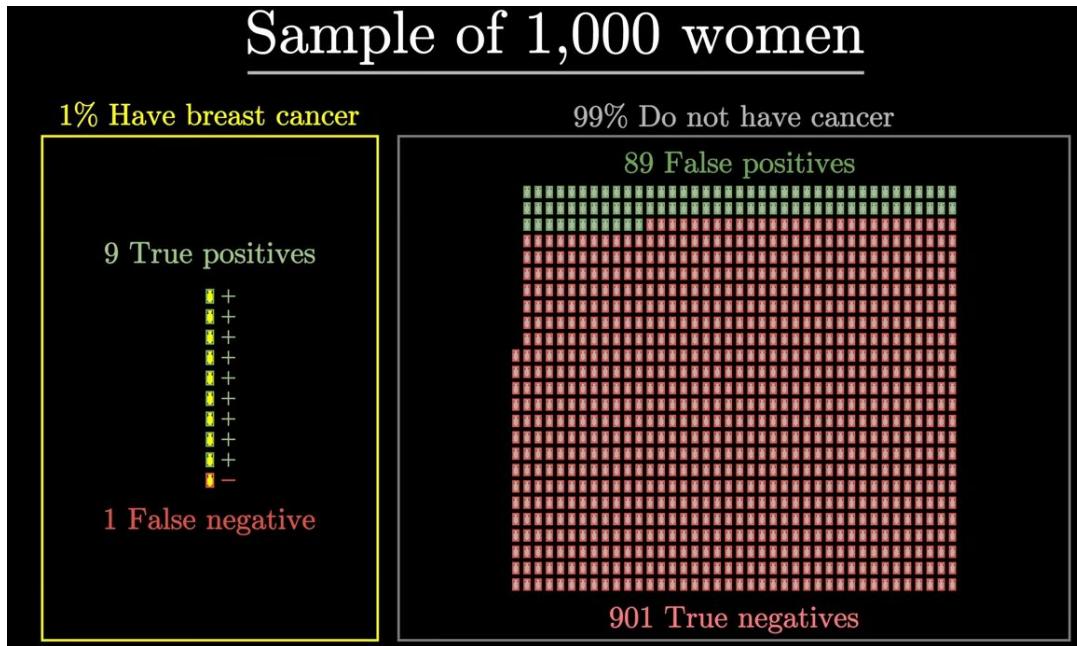
# Exemple de matrice de confusion pour des sujets atteints du cancer du sein

- Le test a une sensibilité et une spécificité de ~90%



Ex  
su

$$P(\text{Have cancer} \mid \text{positive test}) \approx \frac{9}{9 + 89} \approx \frac{1}{11}$$



~~Un test détermine si tu est positif ou non~~  
Un test détermine **la probabilité** d'être positif

~~Un test détermine si tu est positif ou non~~

~~Un test détermine la probabilité d'être positif~~

Un test **met à jour** la probabilité d'être positif

Plus de détails sur cet exemple et sur les tests statistiques dans les prochains cours.