Introduction au machine learning

Guillaume Gastineau,
Maître de conférences Sorbonne Université, LOCEAN/IPSL

guillaume.gastineau@sorbonne-universite.fr

Bibliographie

- Cours Julien Brajard, NERSC / Sorbonne Université

https://github.com/brajard/MAT330

- Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org

- Jake VanderPlas. *Python Data Science Handbook: Essential Tools for Working with Data*. O'Reilly Media, Inc., 1st edition, 2016.

Plan du cours

1) Introduction au machine learning

2) Forêts aléatoires et sélection des hyperparamètres

3) Réseau de neurones

4) Réseau de neurones convolutifs

Pour chaque séance:

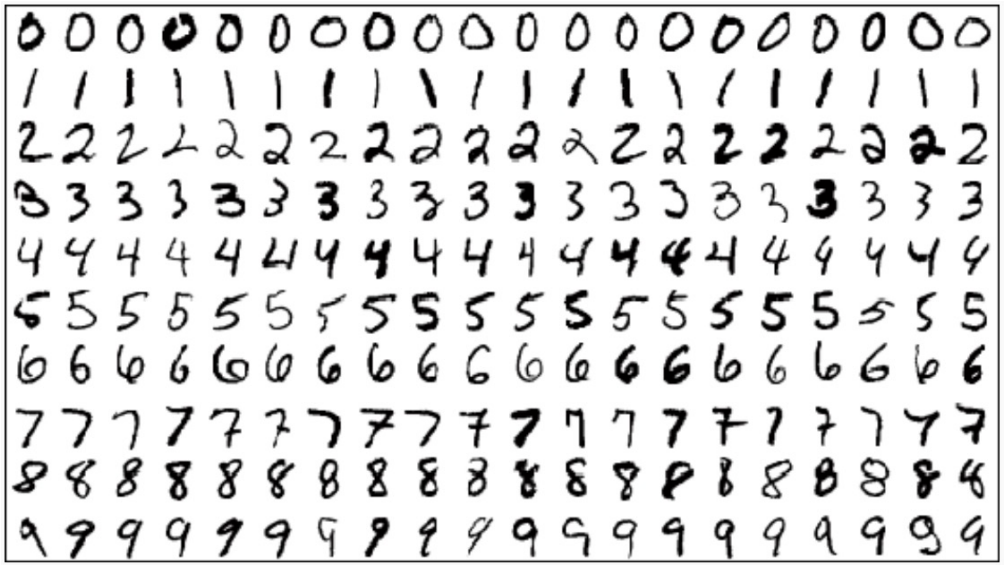
1h de cours / support transparent

2h de travaux pratiques (amener un ordinateur portable)

Introduction au machine learning

- Permet de transformer des données à haute dimension (des milliers ou des millions de dimensions) dans un espace à dimension réduite (moins de 100).
- Permet de transformer des données désorganisées et permet d'en déduire des informations.

Exemple:



Chiffre $\in \{0,1,2,\ldots,9\}$

Deux types de tache

1. Régression

Détermination d'une variable *quantitative* à partir d'un ensemble de données.

Exemple :

- Prédiction du prix d'un bâtiment à partir de différents prédicteurs (Surface, prix des matériaux et de la main d'oeuvre)
- Prédiction de la température dans le futur à partir de la connaissance des températures de la passé

2. Classification

Détermination d'une classe - Un chiffre à partir d'une image

- Identification du contenu d'une image

Types d’apprentissage

- Apprentissage supervisé

On dispose d'un ensemble de données étiquetées avec des exemples de cibles.

Exemple: on dispose d’images

- Apprentissage non supervisé

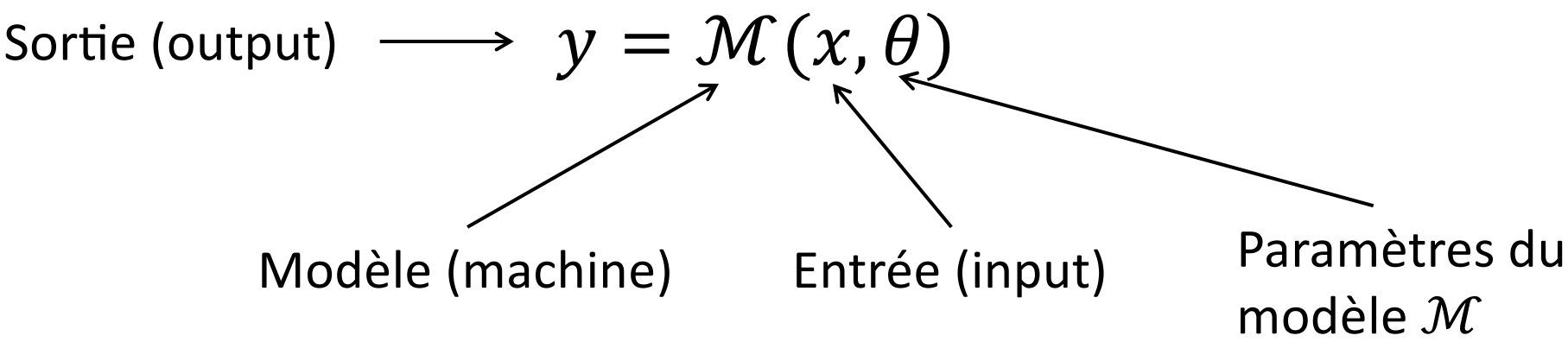
Les données sont non étiquetées, nous n'avons pas d'exemples de ce que nous voulons obtenir. Nous voulons extraire une représentation "utile" de ces données, ou quelques catégories cohérentes.

Exemple : déterminer les comportements typiques des clients dans un supermarché en sachant ce qu'ils ont acheté.

- Apprentissage semi-supervisé

Seul un petit sous-ensemble de données est étiqueté.

Definition d'un modèle



Le Machine learning vise à optimiser les valeurs de θ à partir des données disponibles. Il s'agit d'un processus d'apprentissage.

La recette

- Données :
  - x et y dans le cas d'apprentissage supervisé
  - x seul dans le cas d'apprentissage non-supervisé
- Un objectif:
  - y est une variable quantitative : regression
  - y est une variable qualitative : classification
- Un modèle:
  - linéaire, non-linéaire, forêts aléatoires, réseaux de neurone
- Un processus d'apprentissage
  - Estimation des paramètres θ.

Données multi-dimensionnelles

On suppose avoir des données d'entré de dimension m. m est le nombre de caractéristique de $\boldsymbol{x}$ (ou features). Pour la *i*-ème donnée de $\boldsymbol{x}$ noté $\boldsymbol{x_i}$:

$$\begin{pmatrix} x_{1,i} \\ \vdots \\ x_{m,i} \end{pmatrix} = \boldsymbol{x_i} \qquad \boldsymbol{x} \in \mathbb{R}^{\boldsymbol{n}}$$

La sortie est le plus souvent de dimension réduite. Dans le cas d'une régression: $y \in \mathbb{R}$

On dispose généralement d'un échantillon de taille n. pour $\boldsymbol{x}$ et y. n est le nombre d'échantillons.

Ainsi, on définit $\boldsymbol{X}$ telle que : $\boldsymbol{X} = \begin{pmatrix} x_{1,1} & \cdots & x_{1,m} \\ \vdots & \ddots & \vdots \\ x_{n,1} & \cdots & x_{n,m} \end{pmatrix}$ et $y = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}$

Illustration

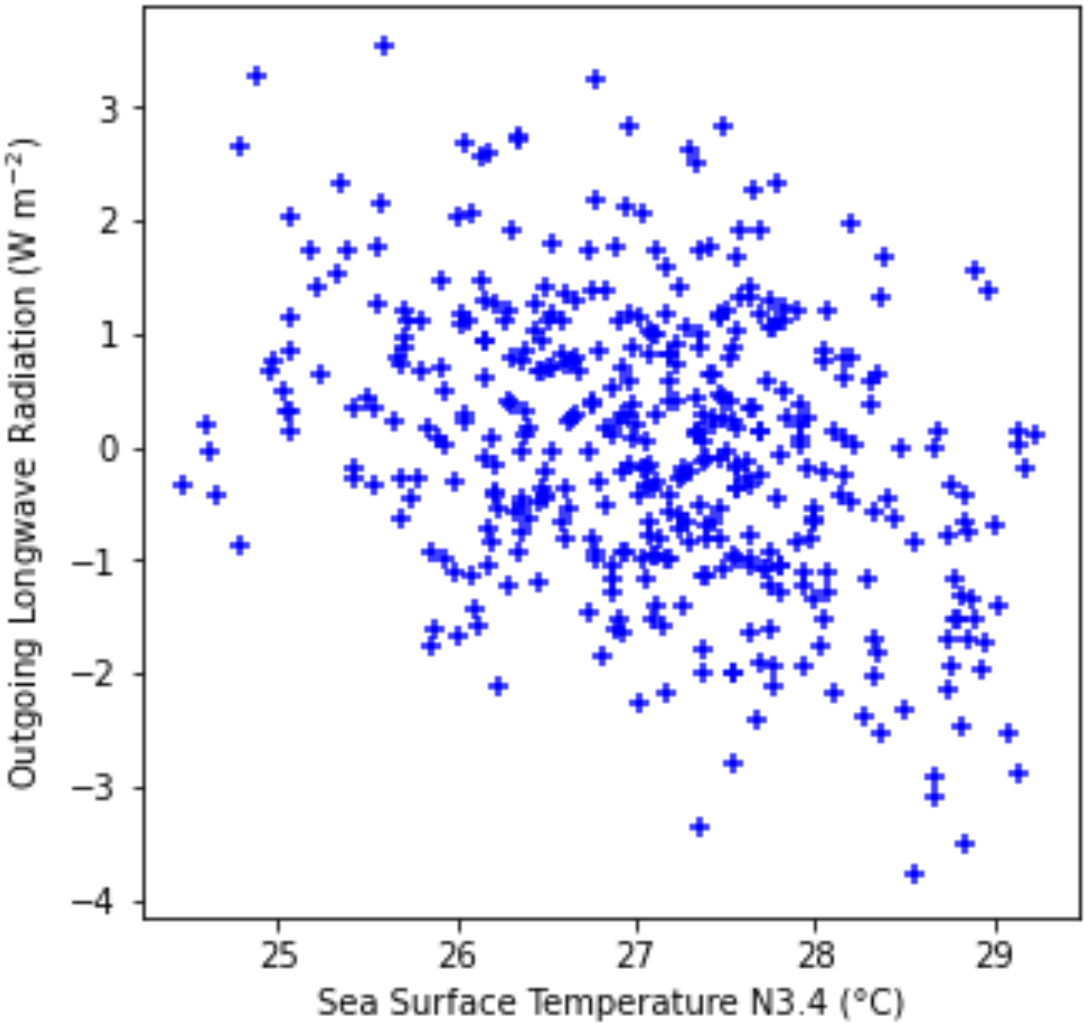
Données

$\boldsymbol{X}$ = indice Niño 3.4 de 1985 à 2019 (m = 1) => une seule caractéristique, donc $\boldsymbol{X}$ est noté $\boldsymbol{x}$

$\boldsymbol{y}$ = Radiations terrestre vers l'espace (OLR) entre 20°N et 20°S

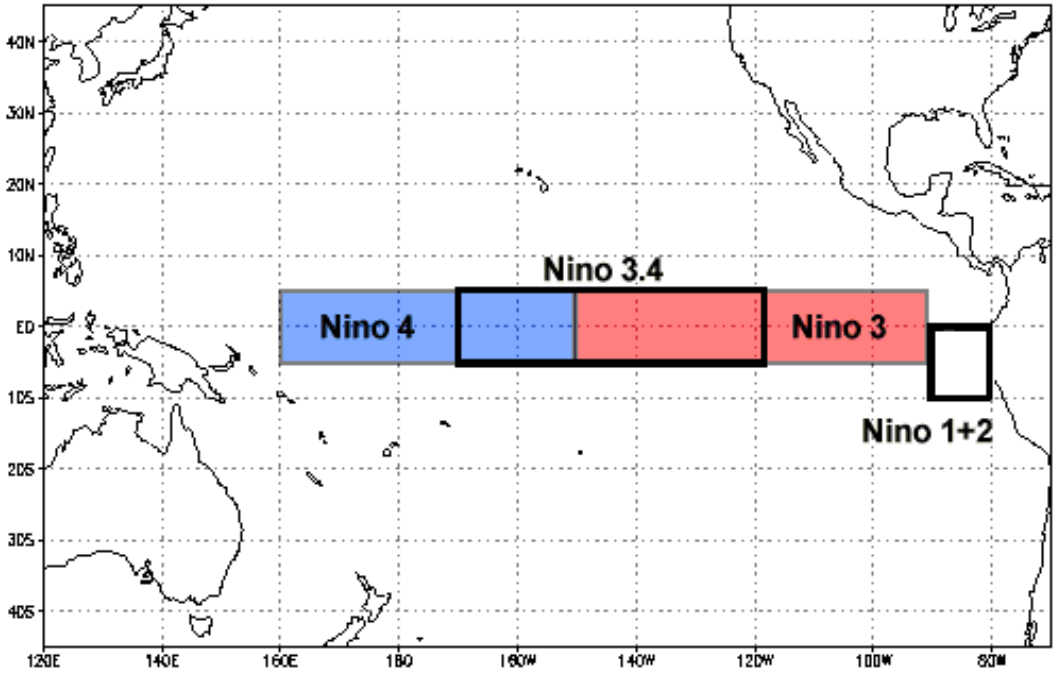
On cherche à prévoir y à partir de x :

- y connu -> *apprentissage supervisé*
- variable quantitative -> *régression*



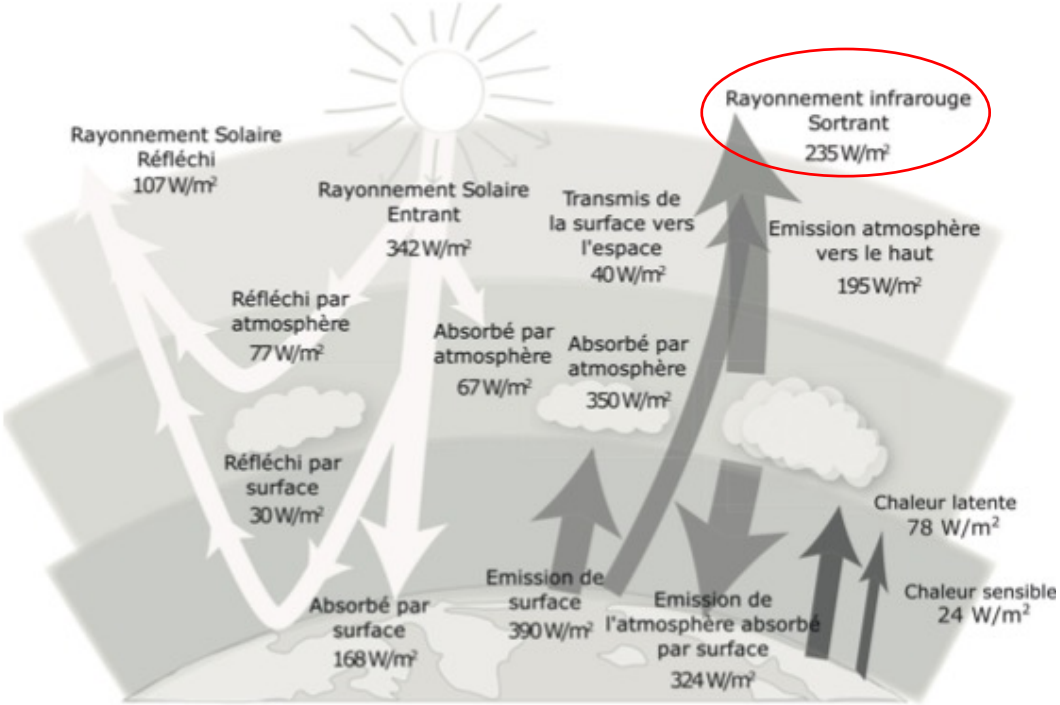
El Niño Southern Oscillation

Indice = température océanique de surface (SST) dans la région Nino 3.4



From https://www.ncei.noaa.gov

OLR = Outgoing Longwave Radiation



From Vallis (2012)

Illustration

- Objectif : estimer y à partir de x. On appelle $\hat{y}$ l'estimation de y. On souhaite minimiser:

$$\sum_{i=1}^{n}(\hat{y}_i - y_i)^2 = L_2(\hat{\boldsymbol{y}} - \boldsymbol{y})$$

où L_2 désigne la norme du même nom.

- Le modèle linéaire est : $\hat{\boldsymbol{y}} = \theta_1 \boldsymbol{x} + \theta_0$

avec θ_0 et θ_1 des paramètres.

- θ_0 est souvent appelé le biais

Illustration

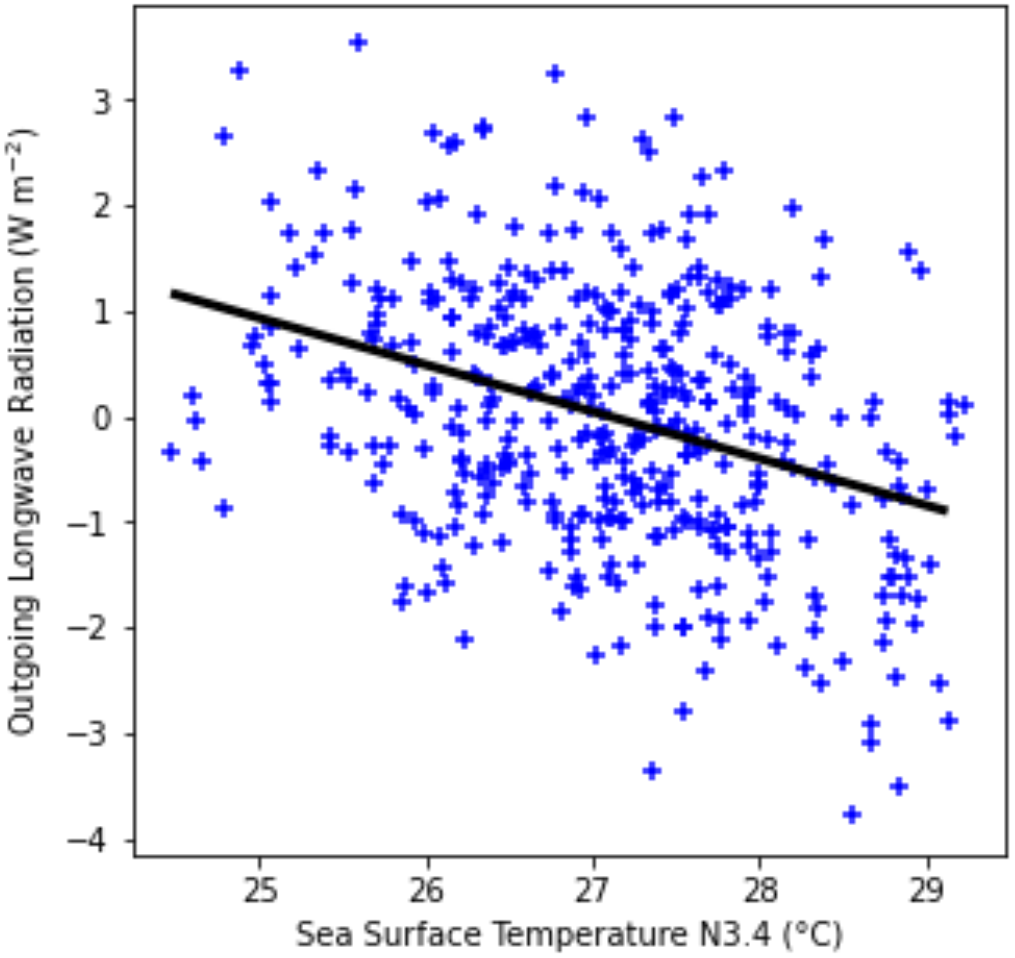
- On peut montrer que:

$$\underset{\theta_1}{\operatorname{argmin}}\, L_2(\hat{\boldsymbol{y}} - \boldsymbol{y}) = (\boldsymbol{x}^T\boldsymbol{x})^{-1}\,\boldsymbol{x}^T\boldsymbol{y}$$

- Et on peut estimer θ_0 avec :

$$\theta_0 = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - \theta_1 \boldsymbol{x_i})$$

- C'est la méthode des moindres carrés (least square.)



Si T augmente → Plus de vapeur d'eau → Plus de d'effet de serre

Illustration

La même méthode peut s'écrire avec plusieur regresseurs (nombre de caractéristiques ou features $m > 1$).

- Le modèle est : $\widehat{\boldsymbol{y}} = \boldsymbol{X\theta} + \theta_0$ avec θ_0 et $\boldsymbol{\theta}$ les paramètres.

θ_0 est le biais et $\boldsymbol{\theta}$ est un vecteur $\begin{pmatrix} \theta_1 \\ \vdots \\ \theta_m \end{pmatrix}$.

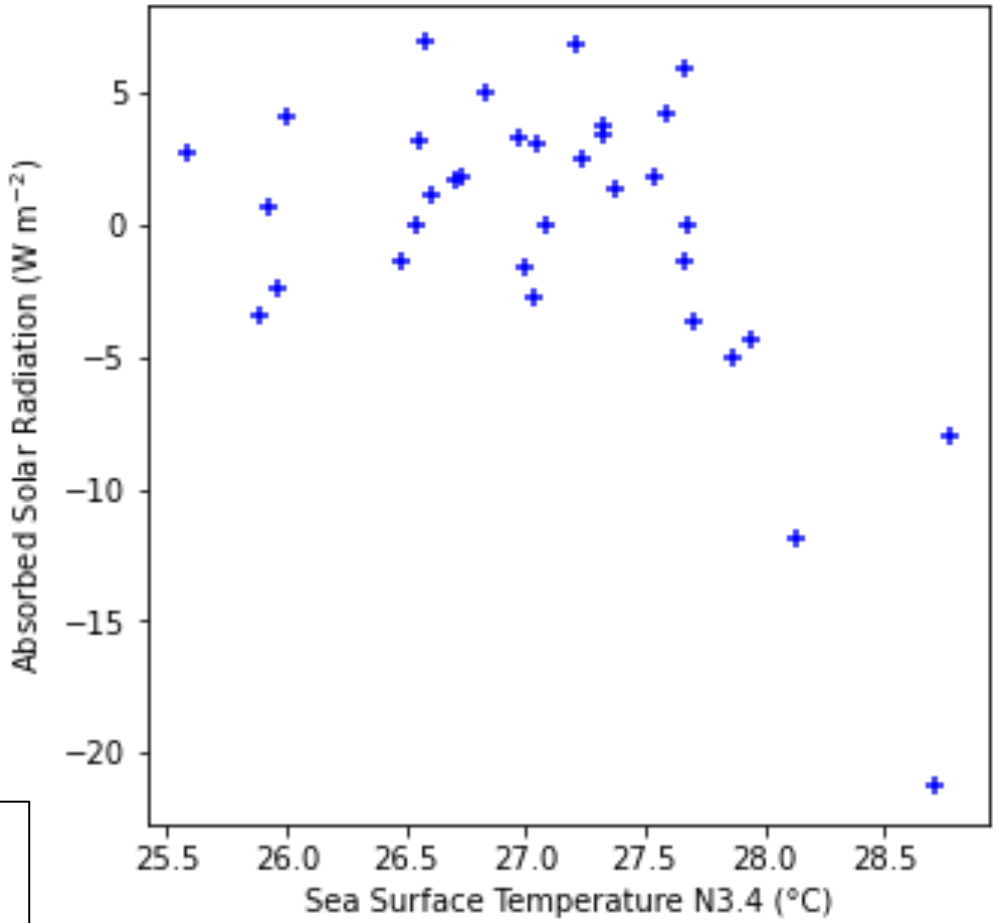
- On a alors : $\boldsymbol{\theta} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\,\boldsymbol{X}^T\boldsymbol{y}$

Sélection de modèle et validation

Données

$\boldsymbol{X}$ = indice Niño 3.4 de 1985 à 2019 (m = 1) => une seule caractéristique, donc $\boldsymbol{X}$ est noté $\boldsymbol{x}$

$\boldsymbol{y}$ = Radiations solaires absorbées au sommet de l'atmosphère dans la région Niño 3.



T augmente → Plus de nuages haut → Plus de radiation solaires réfléchies

Sélection de modèle et validation

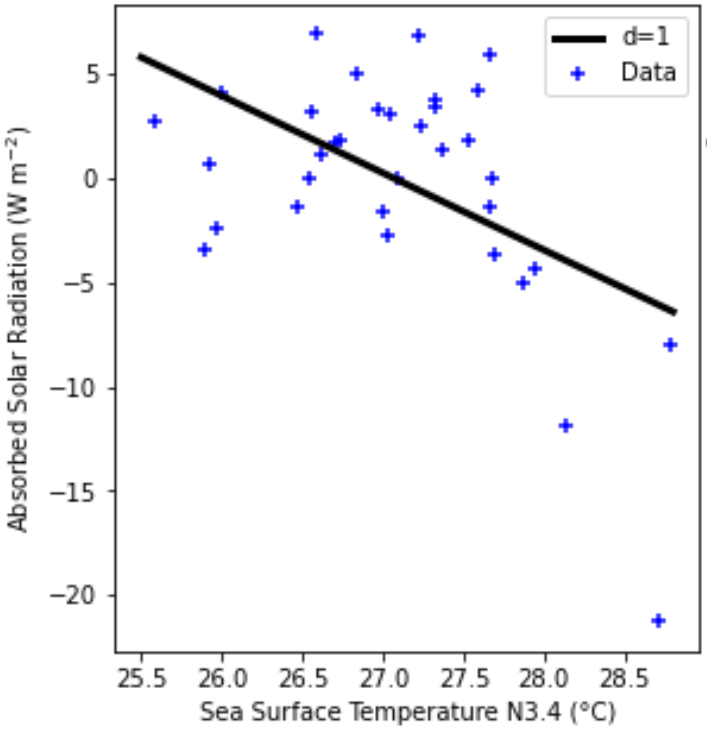
Une régression polynomiale est :

$$\widehat{y_i} = \theta_0 + \theta_1 x_i + \theta_2 {x_i}^2 + \cdots + \theta_d {x_i}^d$$

Sélection de modèle et validation

Une régression polynomiale est :

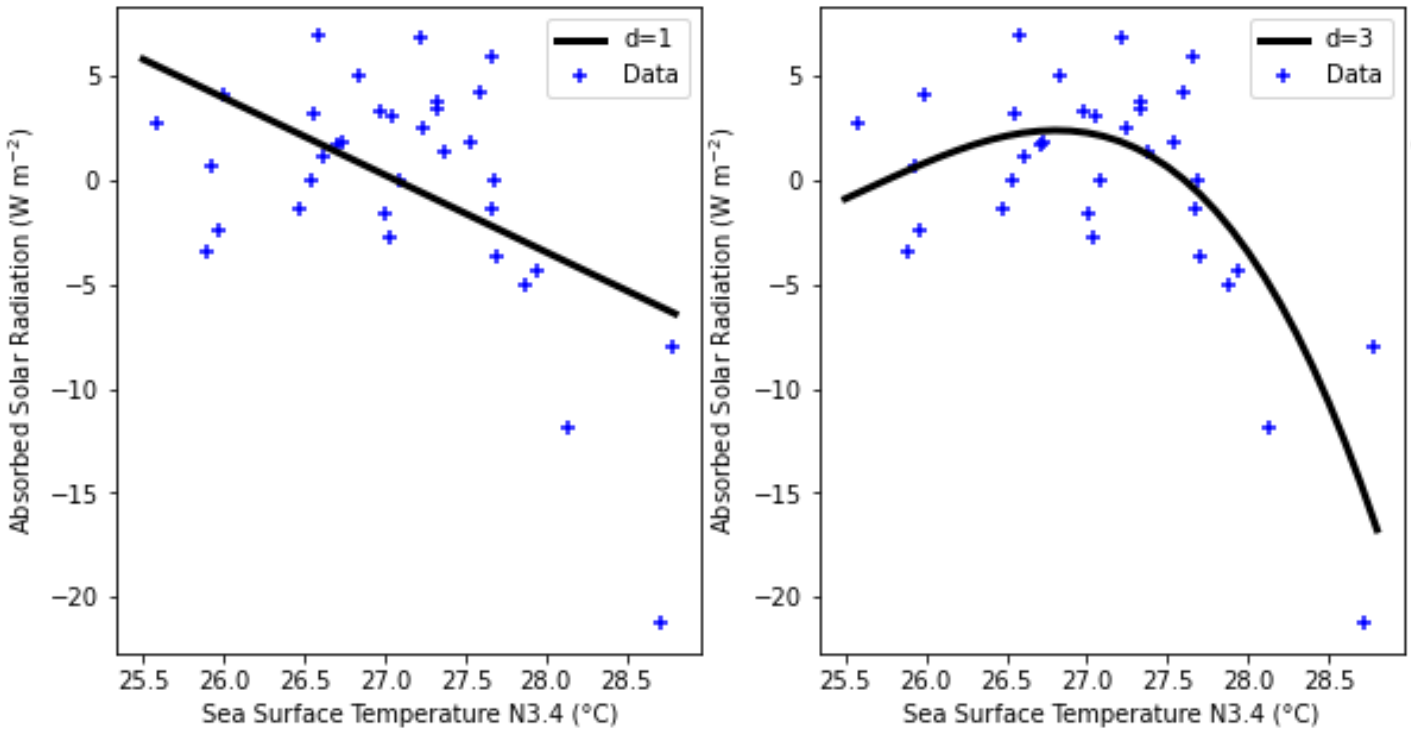
$$\widehat{y_i} = \theta_0 + \theta_1 x_i + \theta_2 {x_i}^2 + \cdots + \theta_d {x_i}^d$$



Sélection de modèle et validation

Une régression polynomiale est :

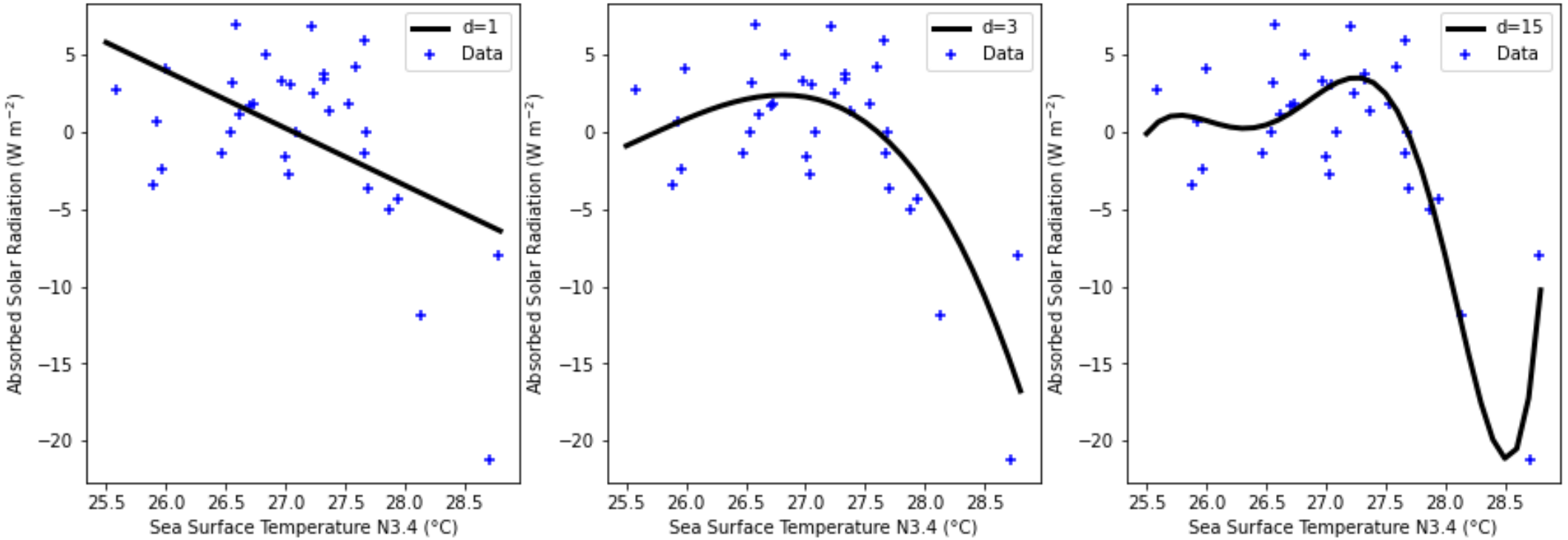
$$\widehat{y_i} = \theta_0 + \theta_1 x_i + \theta_2 {x_i}^2 + \cdots + \theta_d {x_i}^d$$



Sélection de modèle et validation

Une régression polynomiale est :

$$\widehat{y_i} = \theta_0 + \theta_1 x_i + \theta_2 {x_i}^2 + \cdots + \theta_d {x_i}^d$$

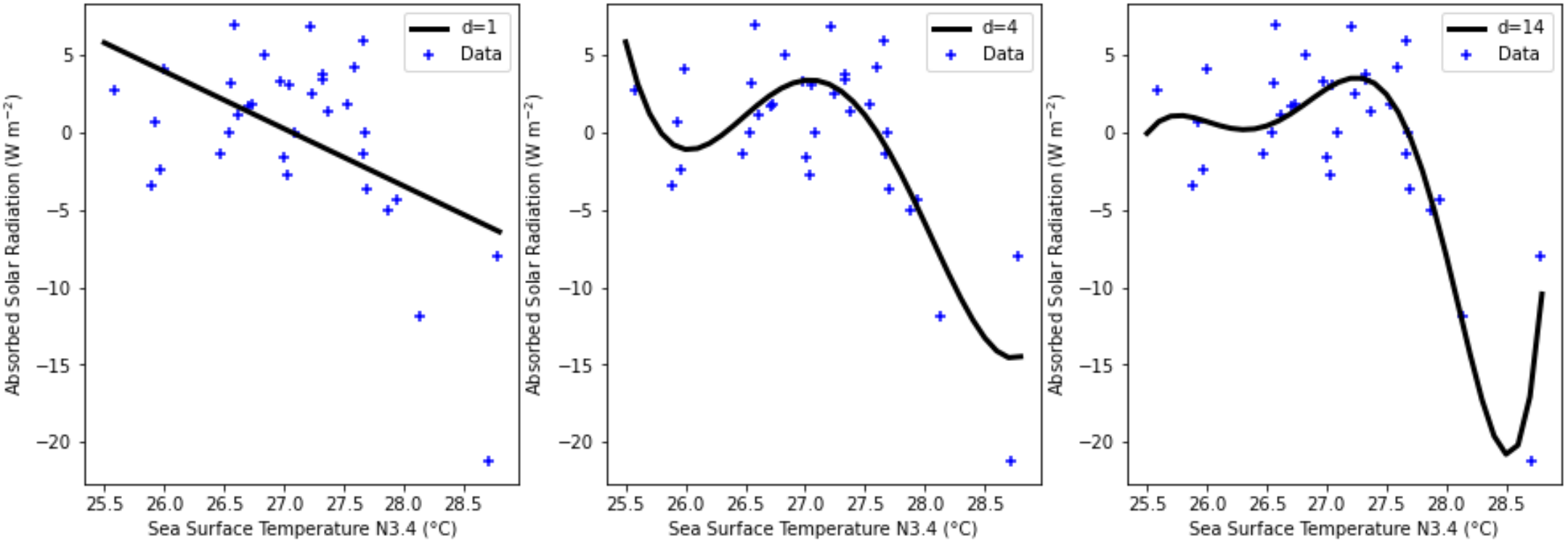


Sélection de modèle et validation

Une régression polynomiale est :

$$\widehat{y_i} = \theta_0 + \theta_1 x_i + \theta_2 {x_i}^2 + \cdots + \theta_d {x_i}^d$$

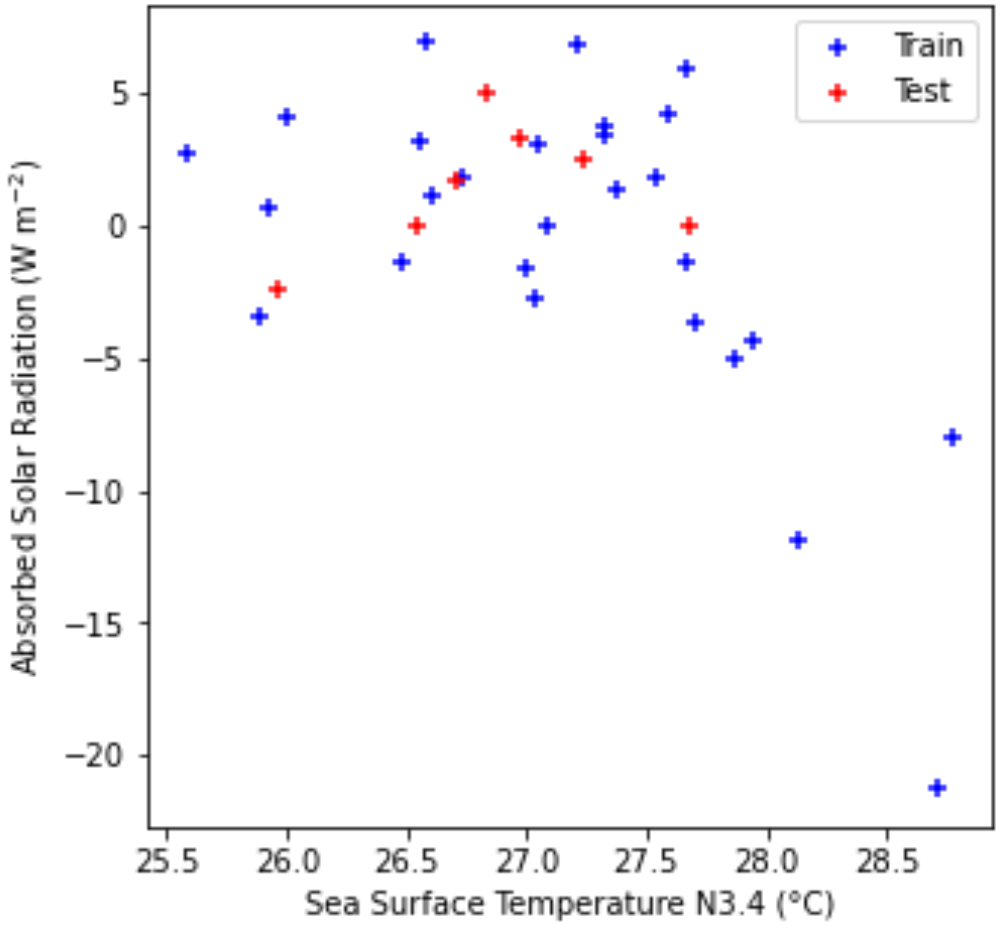
Quel est le meilleur modèle?



Diviser ses données

Pour évaluer chaque modèle, on divise sa base de données en deux :

- La base de données d'entraînement (*train*) est utilisée pour calculer les paramètres.
- La base de données de test (*test*) est utilisée pour évaluer le modèle (par exemple avec une correlation, *R*, ou une erreur quadratique moyenne, *MSE*)



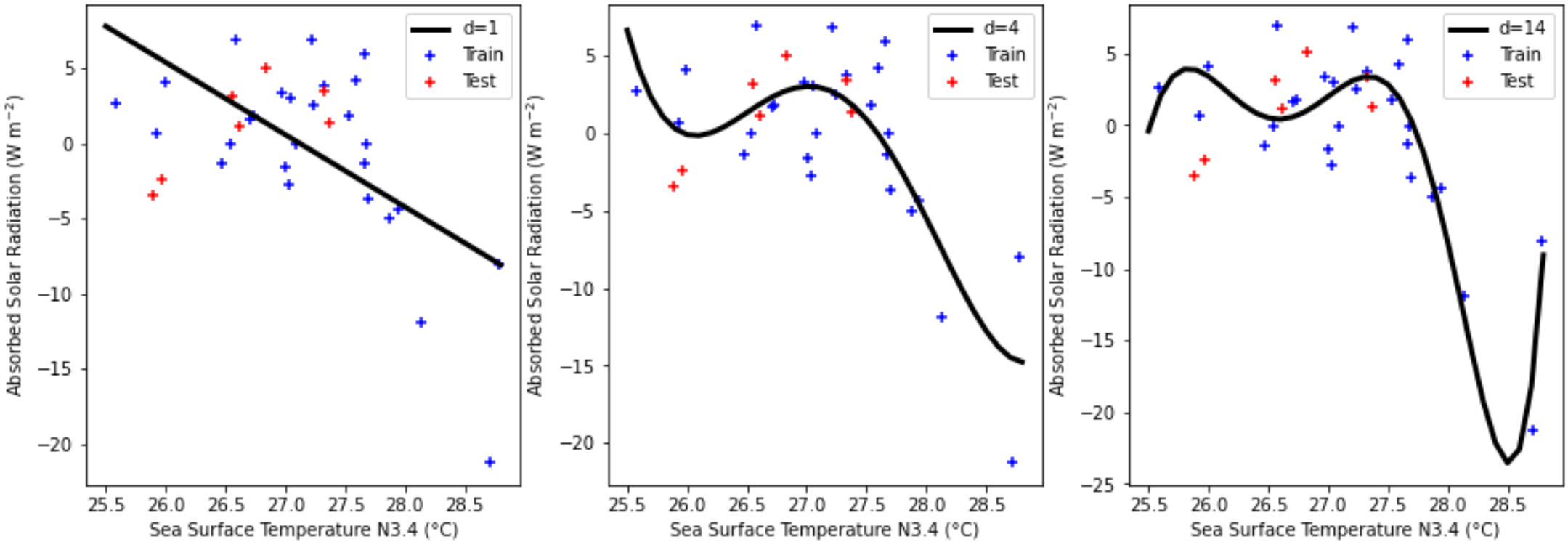
$(\boldsymbol{y}, \boldsymbol{X})$ divisé en $(\boldsymbol{y}_{train}, \boldsymbol{X}_{train})$ et $(\boldsymbol{y}_{test}, \boldsymbol{X}_{test})$

Choix du modèle

$$\mathrm{MSE}_{test} = L_2(\hat{\boldsymbol{y}}_{test} - \boldsymbol{y}_{test})$$

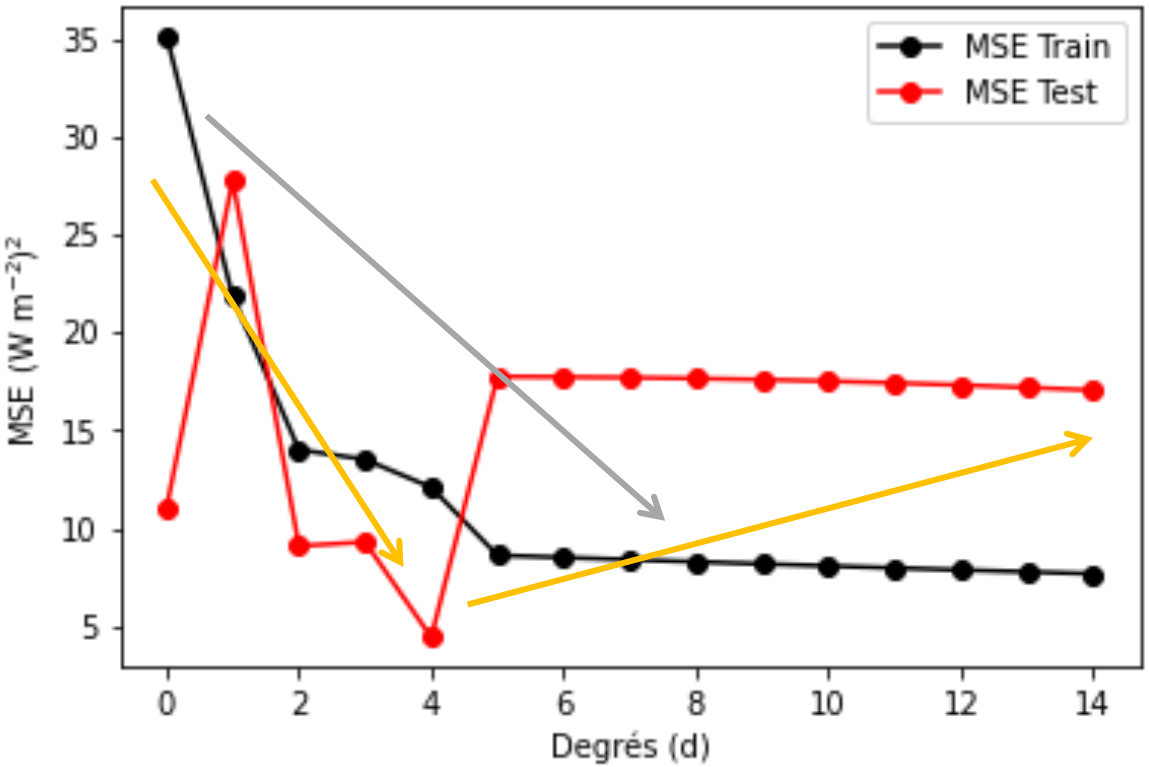
$$\mathrm{MSE}_{train} = L_2(\hat{\boldsymbol{y}}_{train} - \boldsymbol{y}_{train})$$

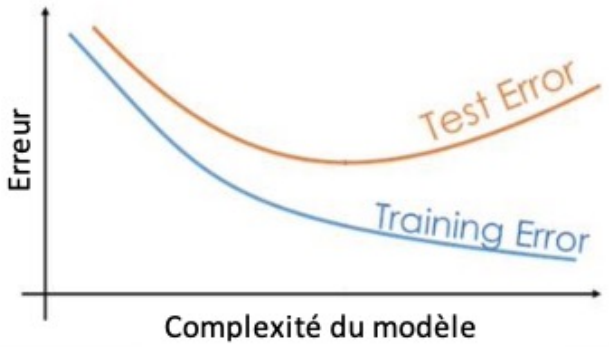
Degré	$d = 1$	$d = 4$	$d = 14$
MSE train $(\mathrm{W\ m^{-2}})^2$	27.76	4.42	17.03
MSE test $(\mathrm{W\ m^{-2}})^2$	21.80	12.07	7.66



Choix du modèle

- L'erreur de test est l'erreur de généralisation du modèle.
- L'erreur de train se réduit avec le degré du polynôme => lorsque le nombre de caractéristiques (features) augmente, celle-ci diminue toujours.

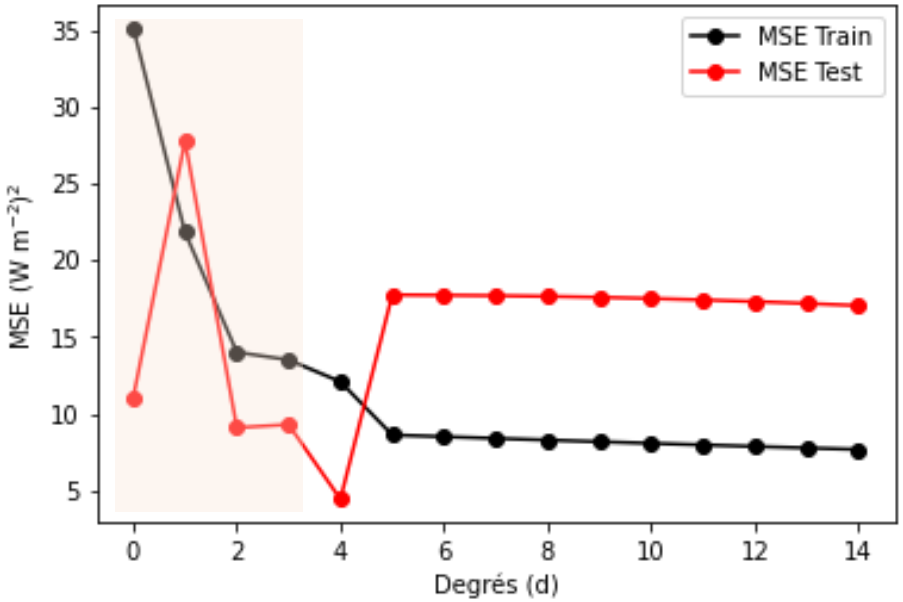


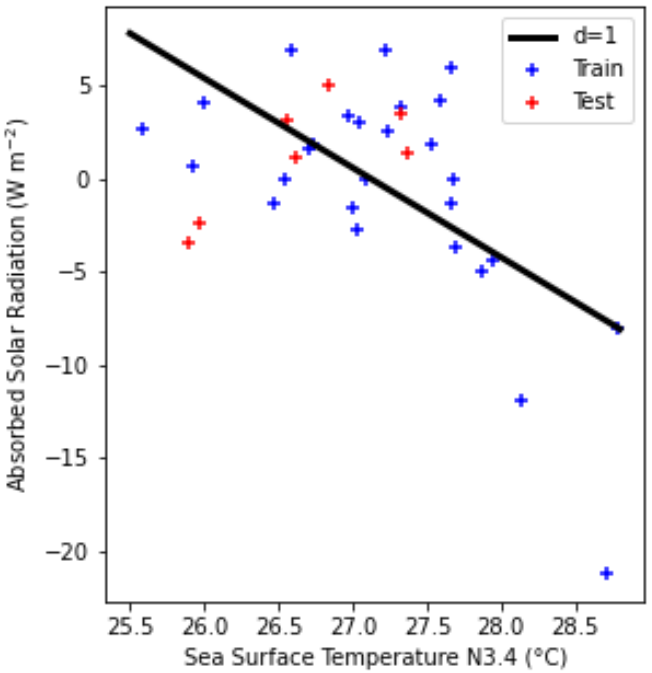


Choix du modèle

On définit classiquement trois régimes:

- Sous-apprentissage (underfitting)

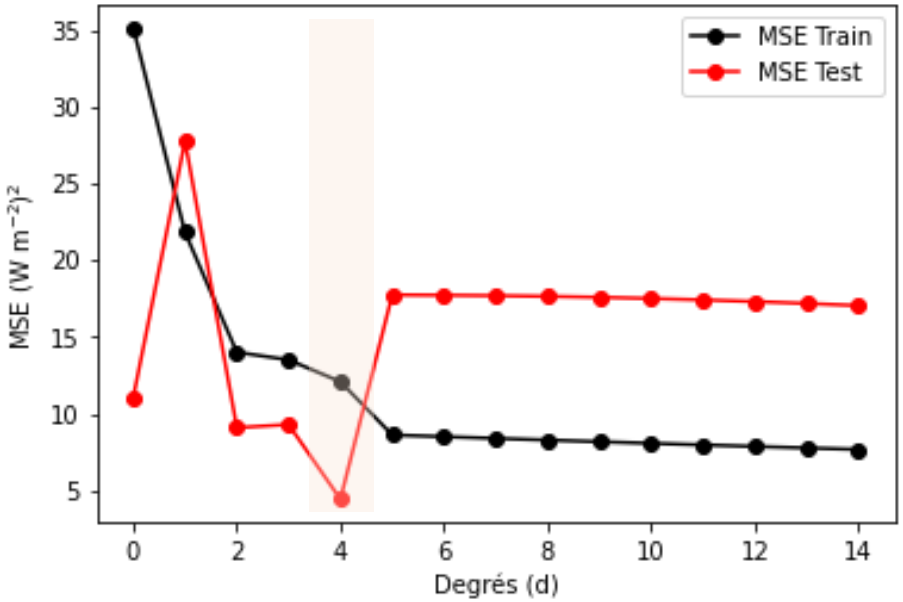


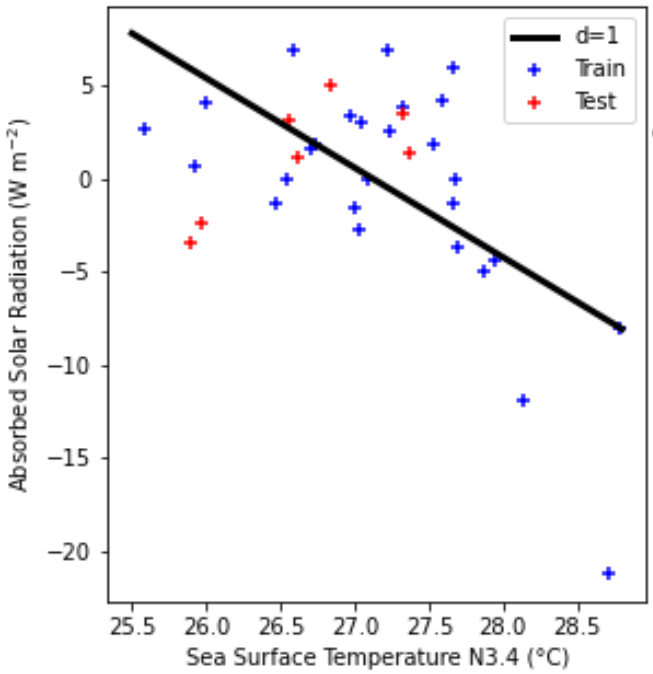


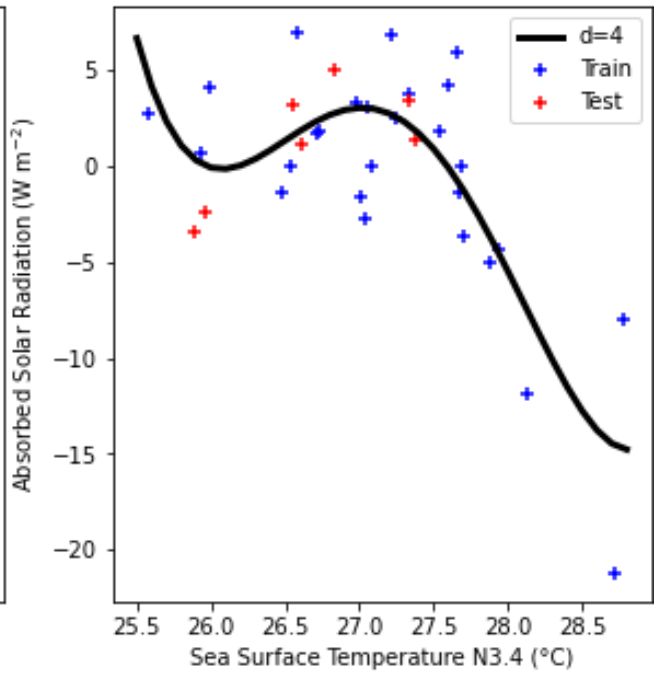
Choix du modèle

On définit classiquement trois régimes:

- Sous-apprentissage (underfitting)
- Bon apprentissage





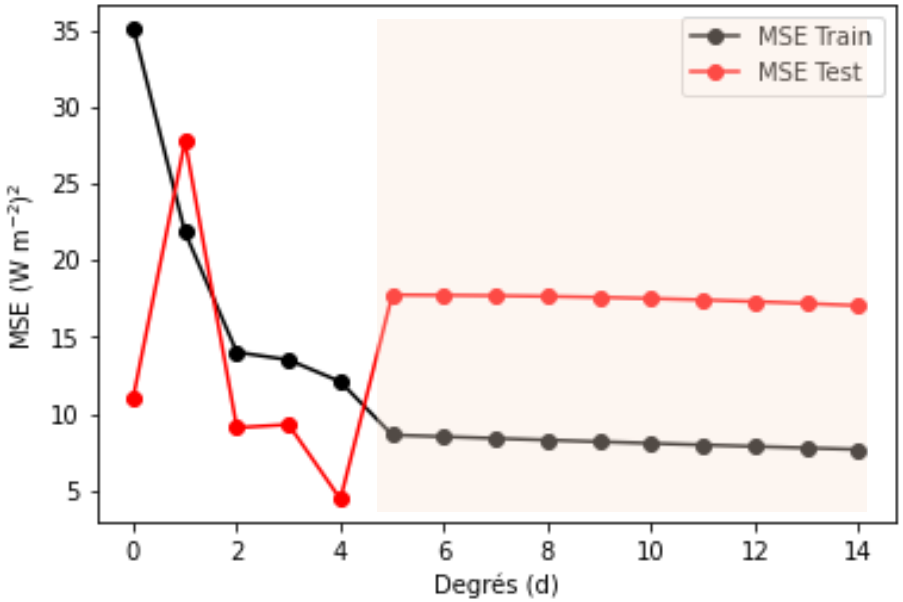


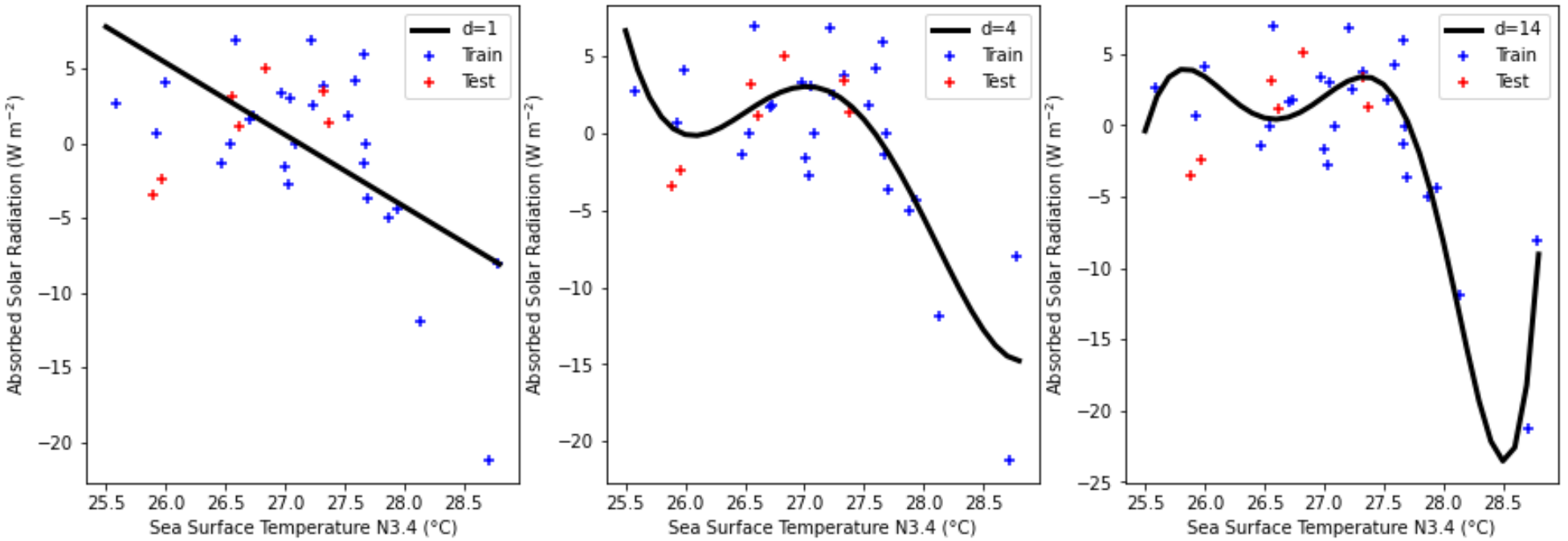
Choix du modèle

On définit classiquement trois régimes:

- Sous-apprentissage (underfitting)
- Bon apprentissage
- Sur-apprentissage (overfitting)

  -> le modèle ne sait plus se généraliser





Division de la base de données

Problèmes potentiels:

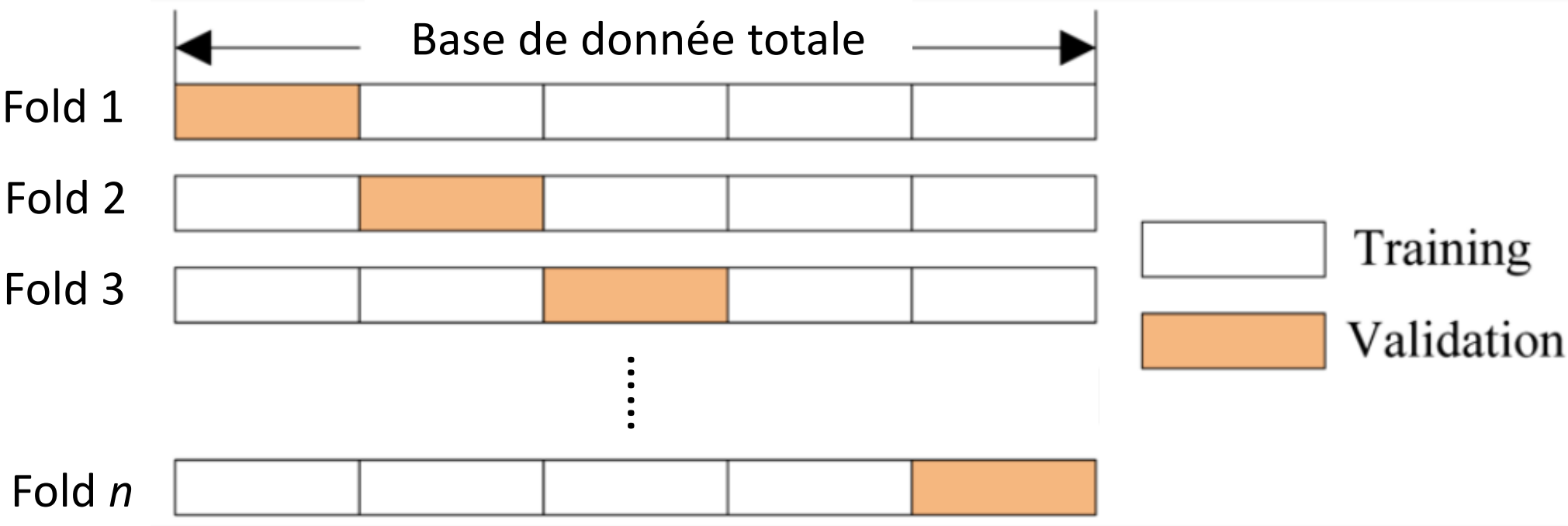
- On réduit le nombre de données -> les paramètres sont donc moins bien estimés.
- Les résultats dépendent des données sélectionnées dans les ensembles de test et d'apprentissage.

Solution = validation croisée (cross-validation)

Validation croisée

Principe:

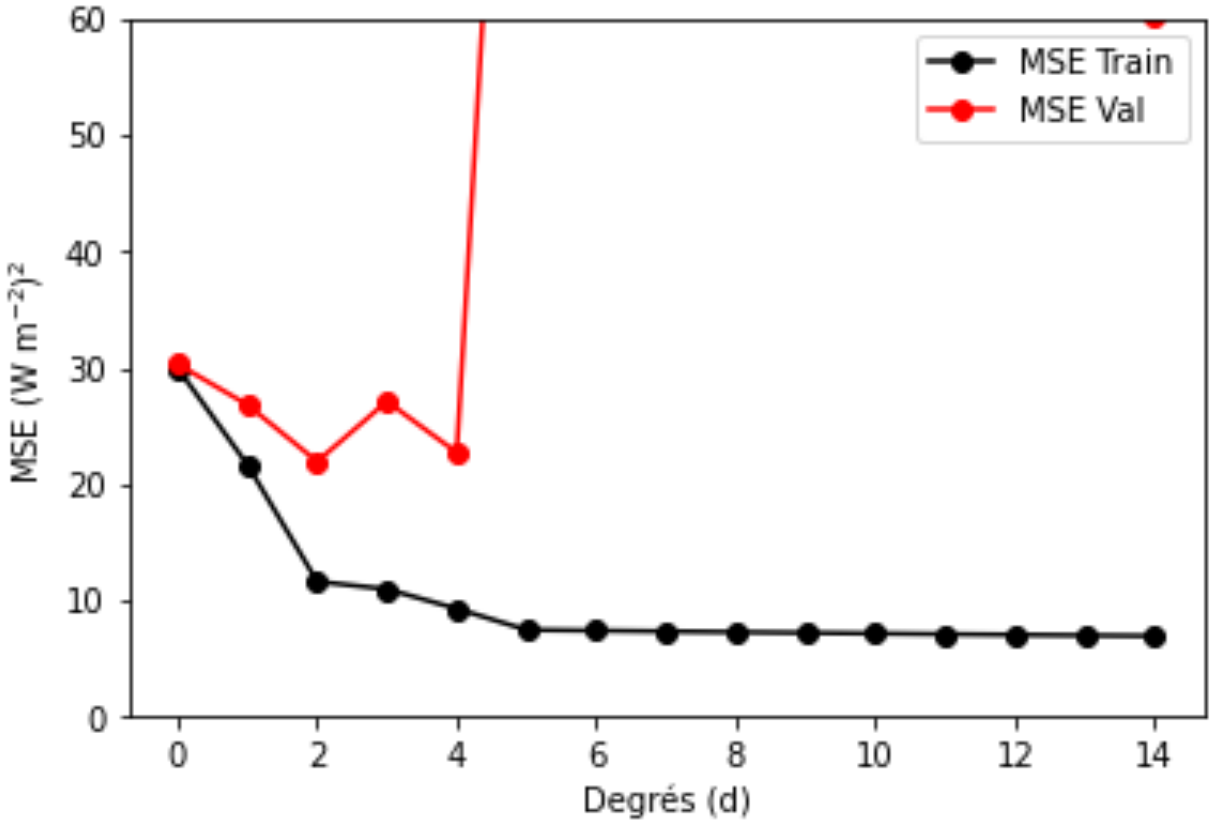
- on découpe les données en n sous-ensemble (= *fold*) d'apprentissage et de test (= validation),
- on apprend n pour chaque sous-ensemble d'apprentissage,
- on calcule n erreurs de test (ou de validation) pour chaque modèle.



Validation croisée

Exemple pour $d = 4$ et $n = 5$:

Fold	MSE validation
1	6.95
2	42.06
3	8.34
4	8.01
5	48.24
Moyenne	22.72



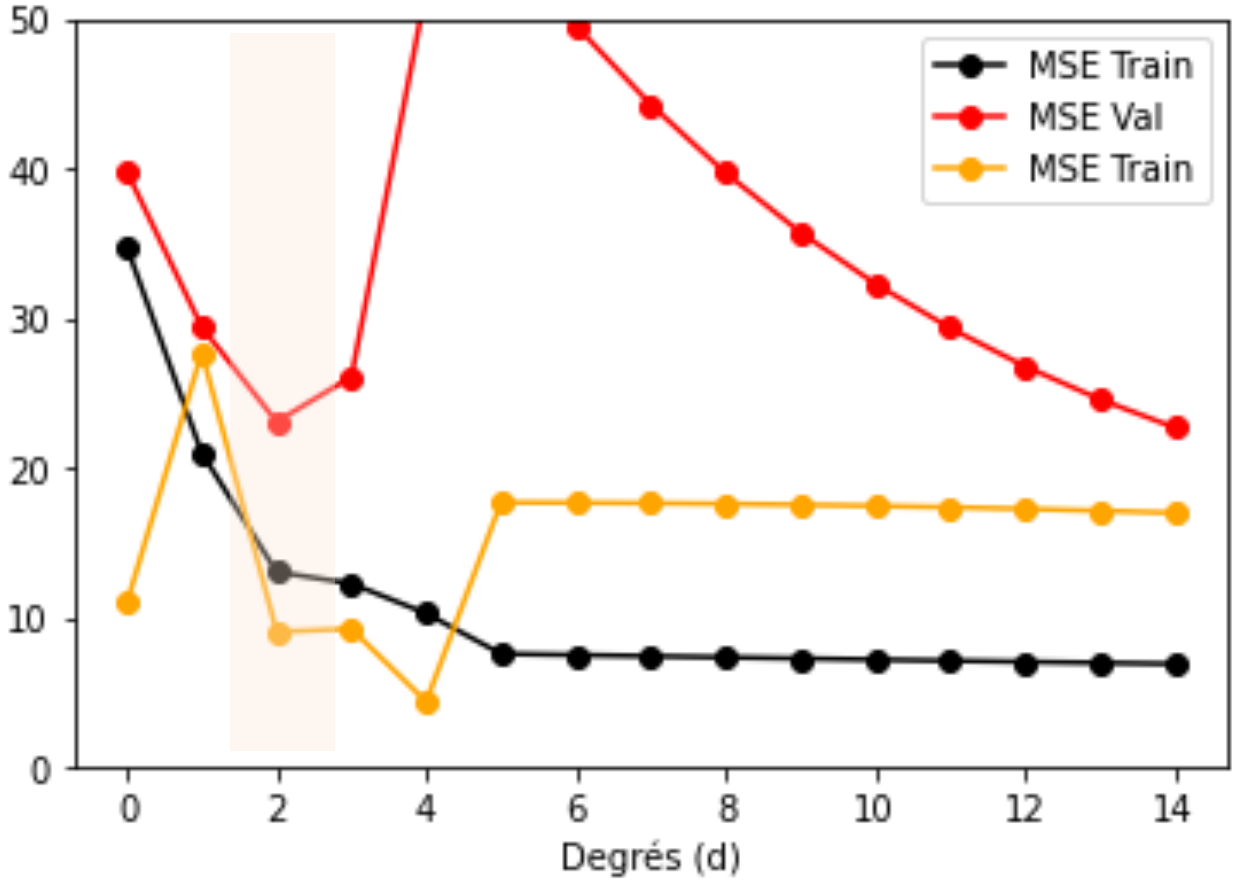
Résumé

- Un modèle est construit avec des paramètres estimés sur un ensemble d'apprentissage,
- La construction du modèle nécessite d'estimer aussi des hyperparamètres (dans l'exemple, d le degré de la regression polynomiale),
- Les hyperparamètres sont alors estimés à l'aide d'un processus de validation,
- Pour évaluer la performance du modèle, on utilise alors un troisième sous-ensemble indépendant.

Retour à l'exemple:

Méthode:

- On divise en deux la base de donnés (train et test).
- On estime les hyperparamètres avec une validation croisée
- On évalue le modèle final à l'aide des données de test.



Cas de données autocorrelés

- Une façon standard de sélectionner la validation consiste à diviser aléatoirement l'ensemble de données selon une proportion donnée.
- ATTENTION ! Cela peut entraîner des problèmes avec les données auto-corrélées (par exemple séries temporelles), plus précisément si le résidu ($\varepsilon_i = \widehat{y_i} - y_i$ ou l'erreur du modèle) est auto-corrélé.