Comportements d'apprentissages

Martin Vielvoye 2021

Entrainement
Validation
et Test

Pédagogie et Examen

Comment écrirait-on un examen pour au mieux tester les connaissances d'une matière acquise par les élèves?

A votre avis?

Pédagogie et Examen

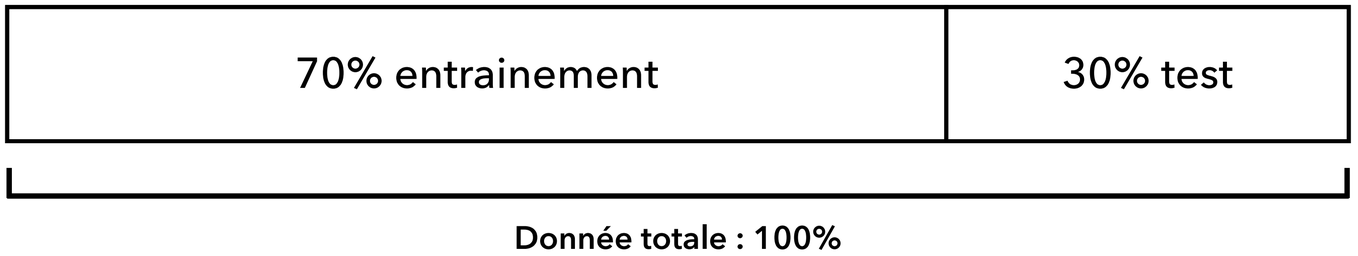
Comment écrirait-on un examen pour au mieux tester les connaissances d'une matière acquise par les élèves?

- Tout les chapitres doivent être présent;
- Les questions ne doivent pas être trop facile ou trop difficile;
- Les questions ne doivent pas avoir déjà été vu;
- Un système de notation équitable est utilisé pour évaluer les résultats.

Et maintenant la même chose pour les modèles en ML !

Entrainement et Test

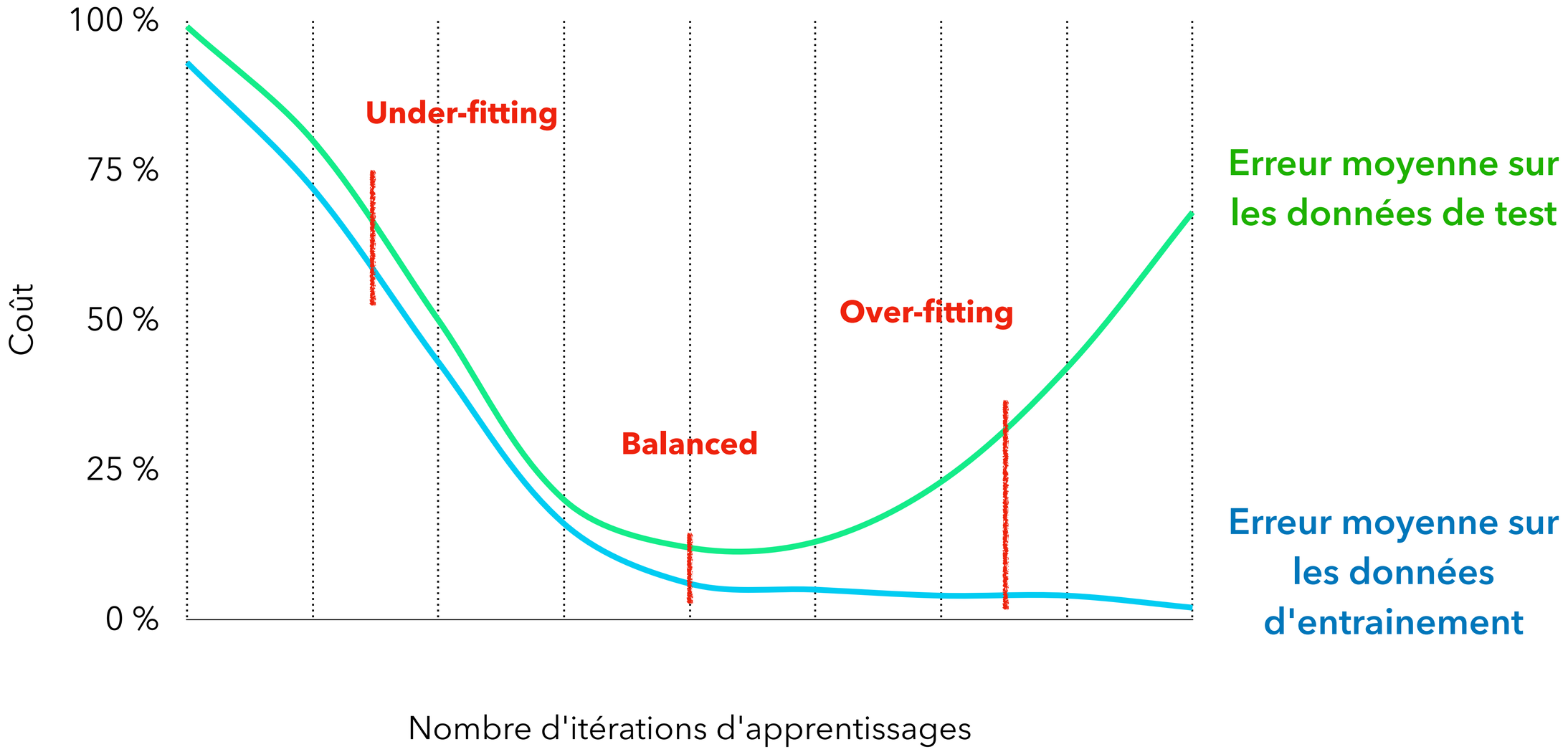
- Comme pour un examen, pour avoir une représentation la plus réelle des performances d'un modèle, les données des tests sont des valeurs que le modèle **n'a jamais vu** lors de l'entrainement.
- Combien de données réservées à l'entrainement et combien au test ?
  - Le nombre n'est pas fixe et depend beaucoup au cas par cas.
  - Si la donne est très variée et possède beaucoup de features alors il faudra de nombreuses valeurs pour que la diversité soit bien représentée.
  - Un bon point de départ serait de diviser un jeu de donnée en :



Entrainement et Test

- On peut maintenant définir l'overfitting et l'underfitting selon certaines performances sur chaque jeu de données.

Précision	Entrainement	Test	
Modèle 1	99 %	25 %	**Overfitting**
Modèle 2	60 %	57 %	**Underfitting**
Modèle 3	93 %	91 %	**Balanced**


100 %
75 %
50 %
25 %
0 %
Coût
Under-fitting
Balanced
Over-fitting
Erreur moyenne sur les données de test
Erreur moyenne sur les données d'entrainement
Nombre d'itérations d'apprentissages

Un jeu de Validation?

- La séparation **Entrainement-Test** est pratique pour évaluer un modèle, mais beaucoup moins quand on veut évaluer différentes combinaisons de paramètres.
- Pour ces cas la, on peut rajouter une nouvelle séparation, en un jeu dis de **Validation**.
- Le jeu de Validation permet de croiser les paramètres, entraînés pendant l'étapes d'entrainement, pour y trouver la meilleures combinaison.
- Un point de départ pour cette séparation serait :
    - **60** % Entrainement
    - **20** % Validation
    - **20** % Test

Overfitting et Underfitting

La Variance

Vous avoir trop chaud ou vous avoir trop froid?

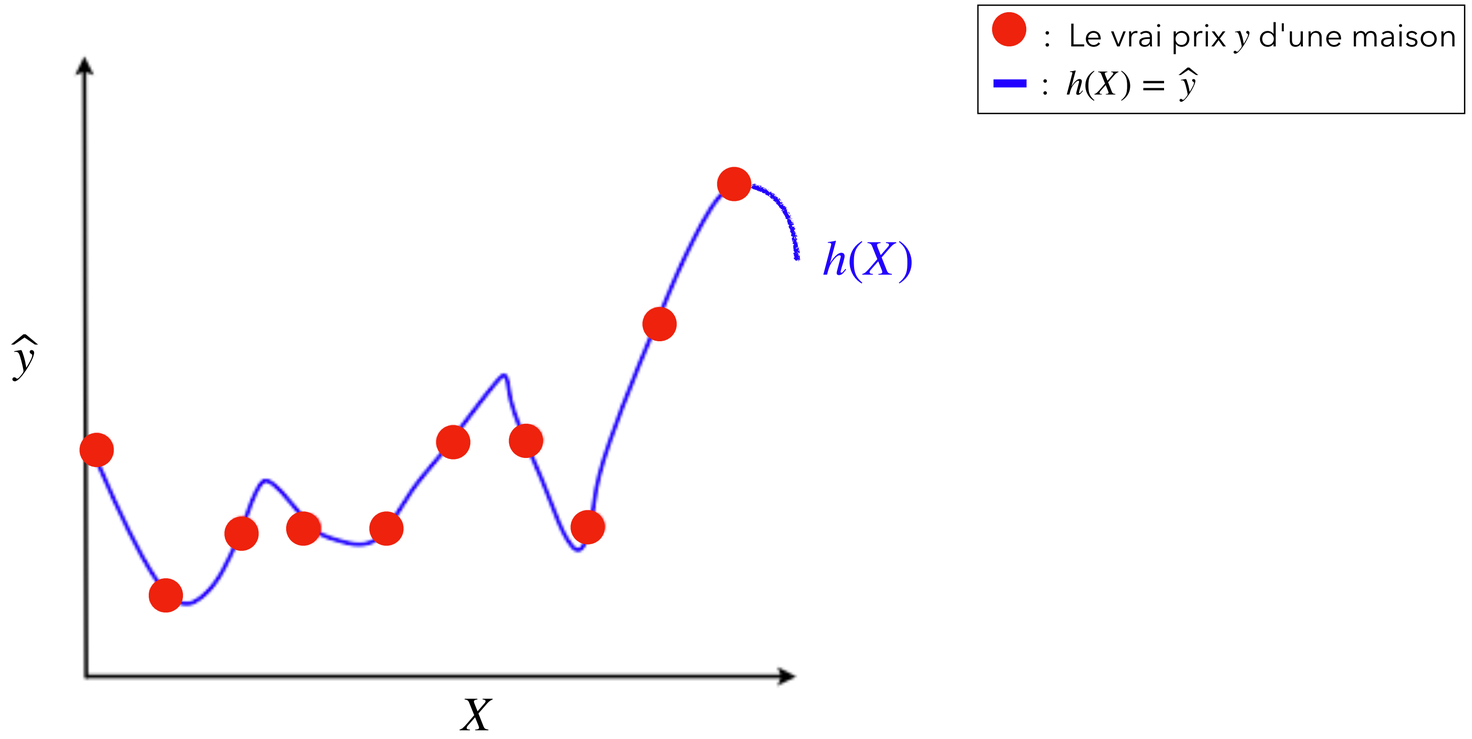
- Prenons l'exemple de la prédiction du prix d'une maison.

# de chambres	Surface m^2	Année Construction	Cheminée	...	Prix
x_1	x_2	x_3	x_4	...	y
3	8450	2003	0	...	208500
3	9600	1976	1	...	181500
3	11250	2001	1	...	223500
...	...	...	...	...	...

- Imaginons que nous avons entrainé une régression linéaire capable de prédire exactement le prix de chaque maison lors de l'entrainement !

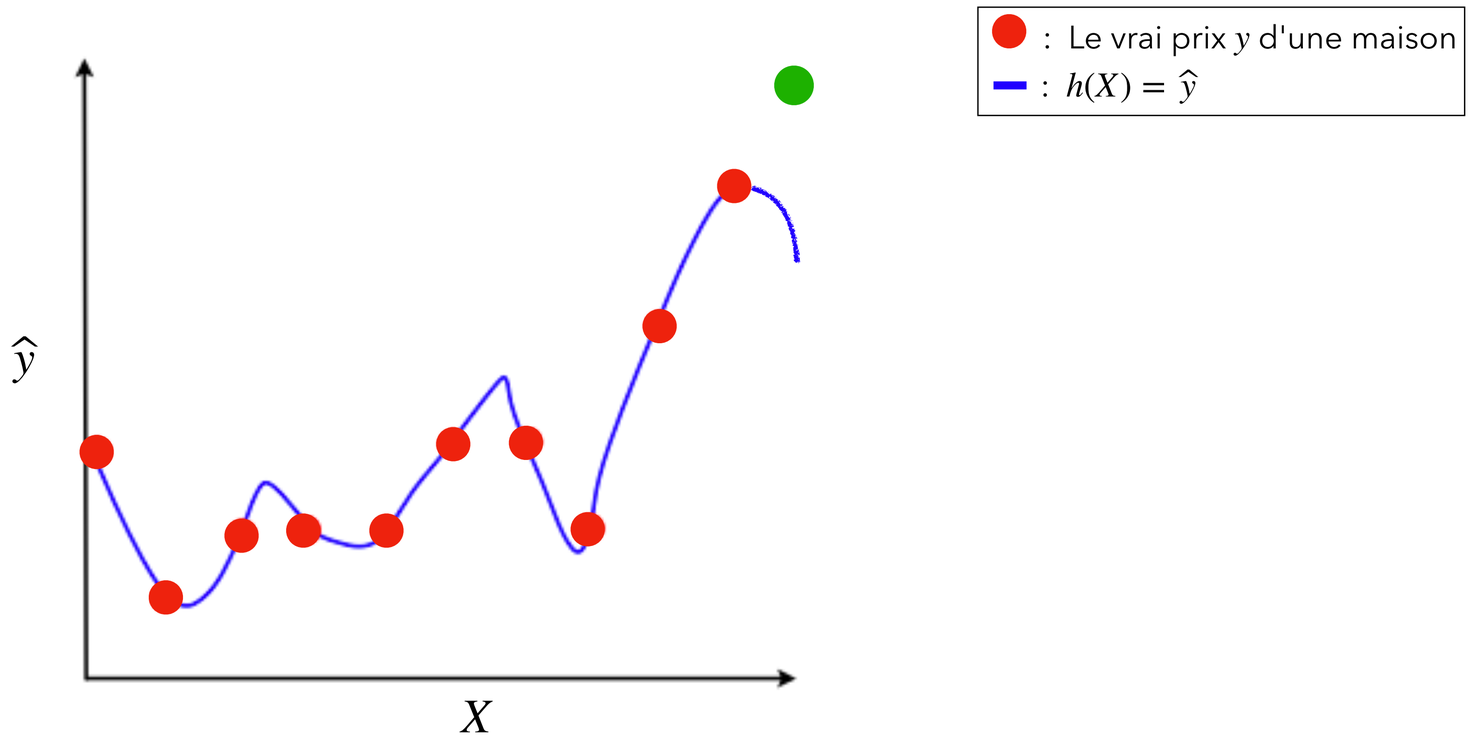
Overfitting en illustration

- Nous avons entraîné une régression linéaire capable de prédire exactement le prix de chaque maison dans le jeu d'entrainement et de validation !



Overfitting en illustration

- Nous avons maintenant une nouvelle donnée ● dont notre modèle n'a jamais vu le prix mais que nous aimerions prédire avec notre hypothèse.



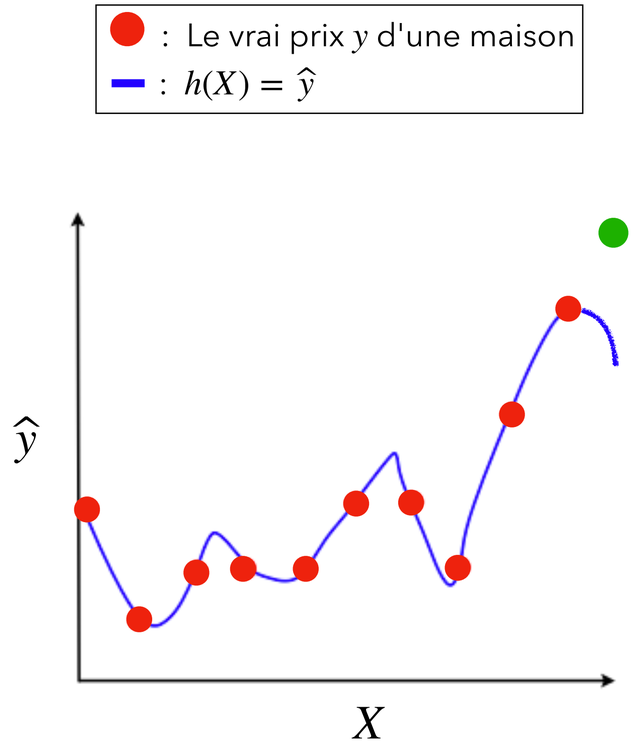
Overfitting en illustration

- Notre modèle est trop complexe et trop précis. Il a du mal à s'adapter à de nouvelles données qu'il n'a jamais vu ce qui n'est pas très efficace pour une prédiction...

- On dit que notre modèle a un problème

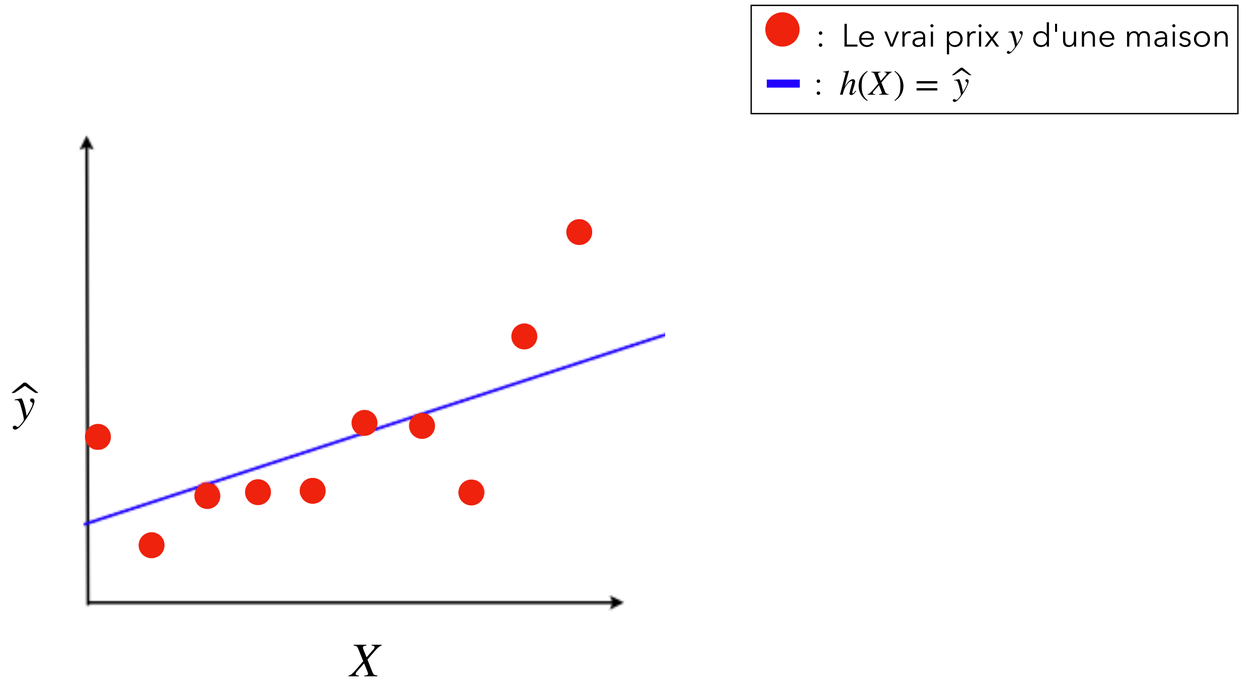
d'**Overfitting**.

(Ou de **variance** faible)



Underfitting en illustration

- On recommence avec un autre modèle cette fois ci, diminuant le nombre de répétition du Gradient Descent et nous avons ce nouvel estimateur $h(X)$.



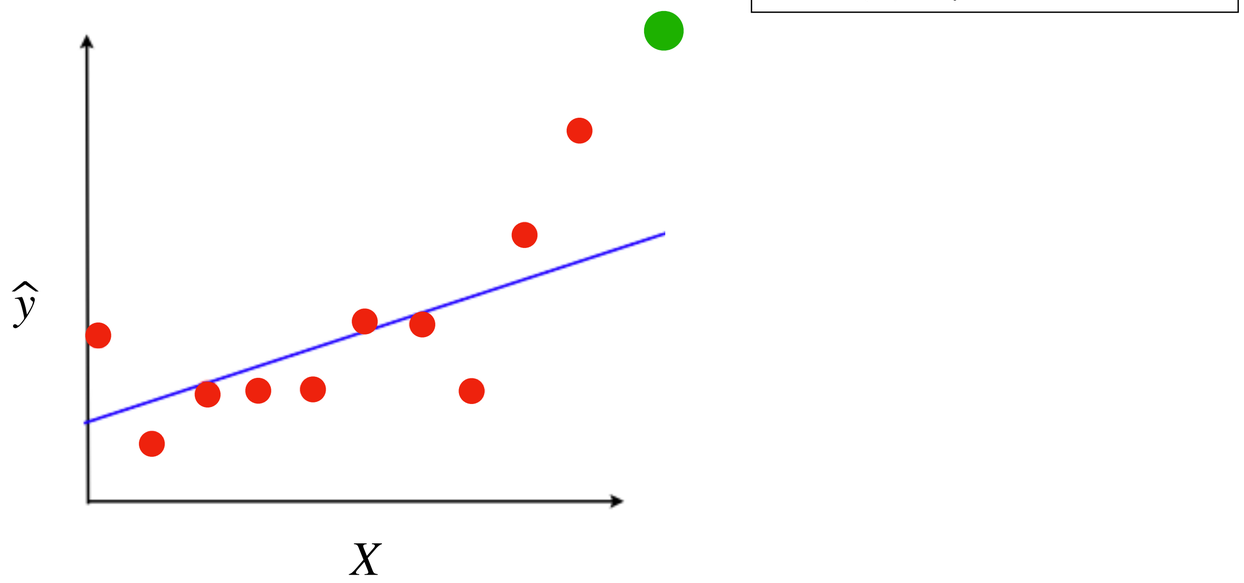
Underfitting en illustration

- Cette fois ci on a un peu du mal a prédire quoi que ce soit… Même les nouvelles valeurs !

- On dit que notre modèle a un problème

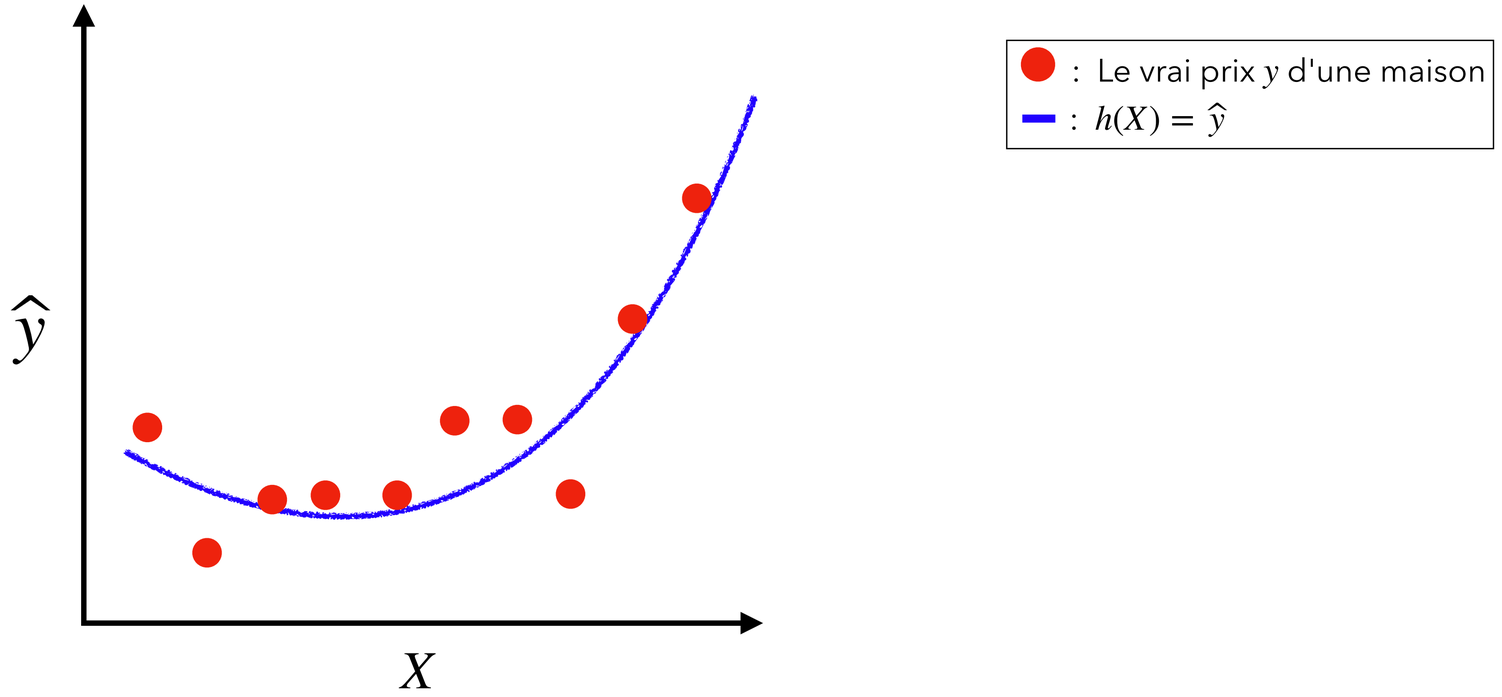
d'**Underfitting**.

(Ou de **variance** forte)



Comme ci comme ca

- On aimerait bien trouver un entre-deux!



La variance… pas forcément trivial !

Comment décririez-vous un arbre ?

Underfitting : L'arbre à du marron en bas et du vert en haut.

Overfitting : 3.52 mètres de haut, 7 racines, entre 36 et 42 branches, 13.240 feuilles, …

Quel serait une définition capable de classifier les arbres (1), des non-arbres (0) ?

https://www.omnicalculator.com/biology/tree-leaves

Diversité et représentation

Le Biais

Thinking outside the box?

- La plupart des algorithmes de ML ont du mal à généraliser en dehors de leurs donnée d’entrainement.
- Si la donnée comporte des biais, ils seront transmis à l’algorithme.

Comment la donnée peut être biaisé?

- Un **biais** est

  “un préjudice pour ou contre quelqu’un ou quelque chose”.

- En data-science/machine learning, le biais peux être résumé a

  “un groupe ou une catégorie de données *sur (ou sous)*

  représenté dans un jeu de donnée.”

- Un algorithme peut avoir du biais dans sa prédiction, souvent quand un modèle présente de l'**underfitting**.
- On appellera le **scope** les ensembles et groupes de données représenté dans le jeu de données.

Le biais de donnée n’est pas forcément mauvais!

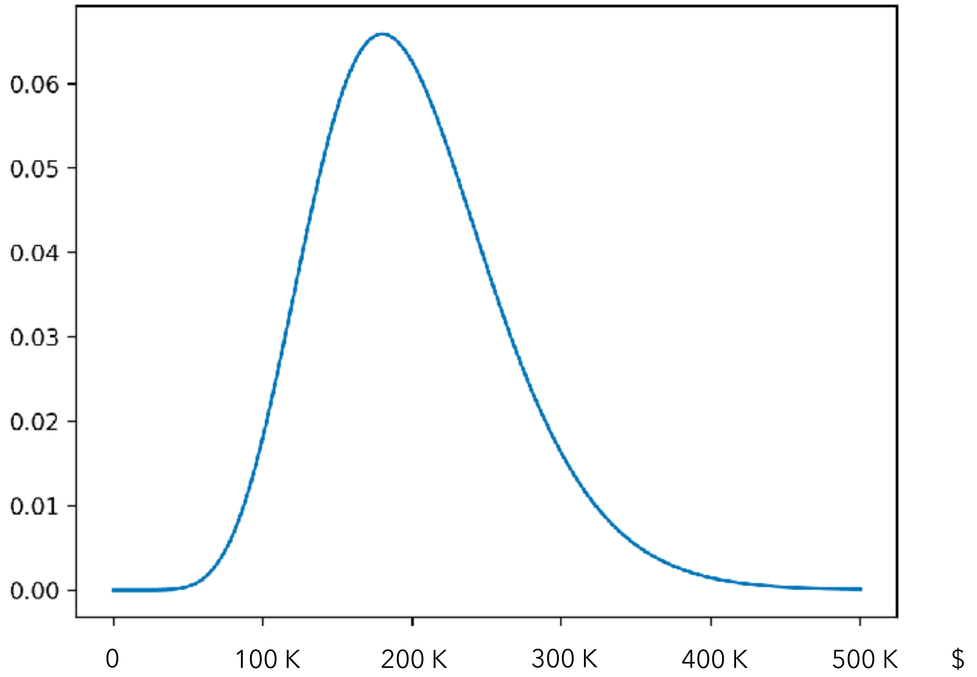
Si le biais est **connu** et **maitrisé**, et que l’algorithmique n’est pas utilisé pour prédire des valeurs sur des données en dehors de son scope, alors son utilisation est considéré '*juste*' et n'impacte pas les performances et les prédictions de l'algorithme.

Analyse d’un jeu de donnée

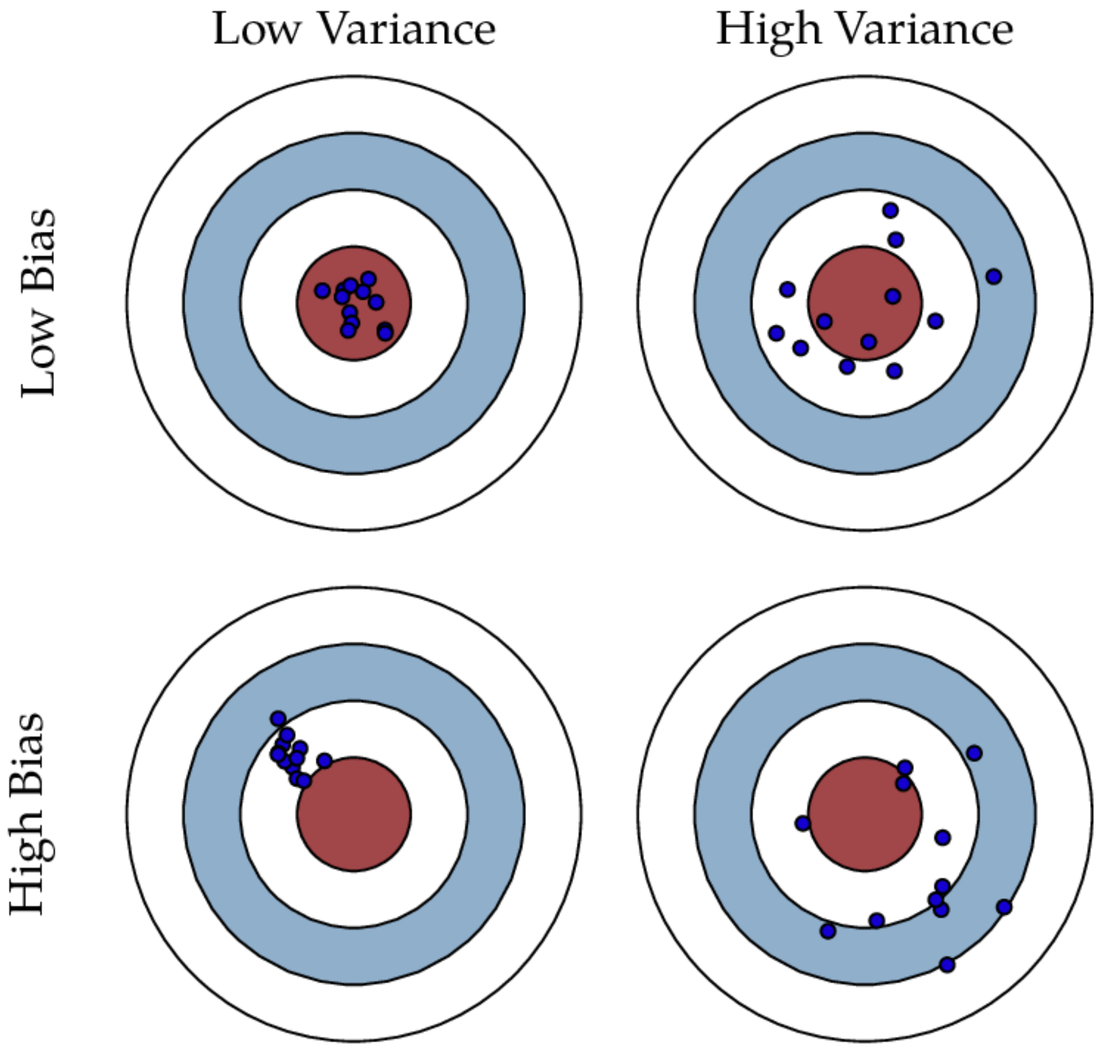
Prédire le prix d’une maison

On analyse le jeux de données sur lequel nous avons entrainé notre algorithme. On se rend compte que le prix moyen d’une maison y est de 193,025 $ et que sa distribution est la suivante :

Notre algorithme à une précision de 92% sur ce jeux de données mais qu'en est-il pour prédire le prix d'une maison qui en vaudrait 2M $? Ou 10,000 $?



Il faut que la donnée d'entrainement **représente** la donnée finale.


Low Variance
High Variance
Low Bias
High Bias

Régularisation

Stratégies contre le *Biais* et la *Variance*

Venir à bout de l'Overfitting

L'**Overfitting** arrive souvent quand un modèle c'est adapté parfaitement aux données d'entrainements.
Mais pour pouvoir faire cela, l'algorithme a besoin de beaucoup de features θ lui permettant une haute complexité. (Plus il y a de poids, plus un modèle est complexe)

1. Réduire le nombre de features
    - Analyser et choisir les features à enlever à la main pour réduire la complexité de l'hypothèse.
    - Utiliser des algorithmes de sélections de variables ou combiner certaines features ensemble.
2. Régularisation
    - Garder toutes les features mais réduire l'impact des poids θ.
3. Réduire α pour que l'apprentissage soit plus lent.

Régularisation

- On peut influencer sur chaque poids θ directement, boostant ou réduisant l'impacte des features sur notre fonction de coût J.
- La variable λ est un hyperparamètre que nous choisissons :
  - Trop large, il linéarise trop le modèle finale et créé de l'**underfitting**.
  - Trop petit et il n'aurait pas assez d'impact sur l'apprentissage du modèle et ne combat pas **l'overfitting**.

$$J = \frac{1}{2m} \sum_{i=1}^{m} (\hat{y}_i - y_i)^2 + \lambda \sum_{j=1}^{n} \theta_j^2$$

X	Y	$\hat{y}$
1101010	1101010	1101010
001010	001010	001010
1001000	1001000	1001000
1011010	1011010	1011010
0000010	0000010	0000010
…	…	
1011011	1011011	1011010