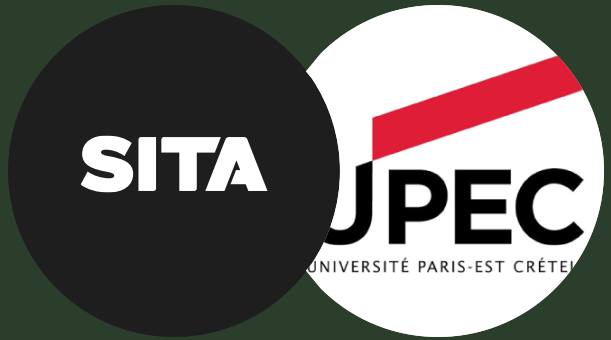


EPISEN – ING3. SI

Machine Learning

Abdallah EL HIDALI
Tech Lead Sita For Aircraft
abdallah.el-hidali@sita.aero

EPISEN
2024/2025

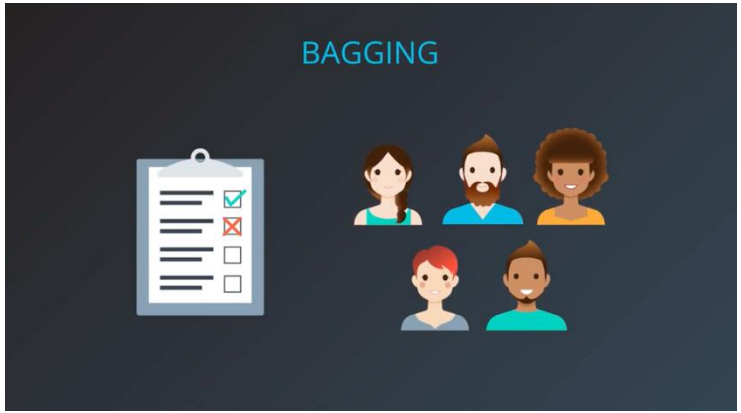
SITA

VII. Les méthodes d'ensemble

Introduction

Les méthodes d'ensemble sont des techniques en apprentissage automatique qui **combinent plusieurs modèles pour obtenir des prédictions plus robustes et précises**. Voici une explication simplifiée des deux méthodes principales : **le bagging et le boosting.**

Bagging



Mise en Situation :

1.Préparation :

- Vous avez un test à passer, composé de plusieurs questions.
- Plutôt que de répondre seul, vous décidez de demander l'aide de plusieurs amis.

2.Réalisation des échantillons :

- Chaque ami représentera un modèle dans notre système de bagging.
- Pour rendre le processus similaire à l'idée de "bootstrap," imaginez que chaque ami répondra à une version légèrement modifiée du test, où certaines questions pourraient apparaître plusieurs fois (si on voulait imiter exactement le bootstrap avec remplacement).

3.Réponse au test par les amis :

- Chaque ami répond indépendamment au test en utilisant ses propres connaissances, sans se consulter.
- Les réponses de chaque ami peuvent varier : certains peuvent exceller dans certaines questions, d'autres moins.

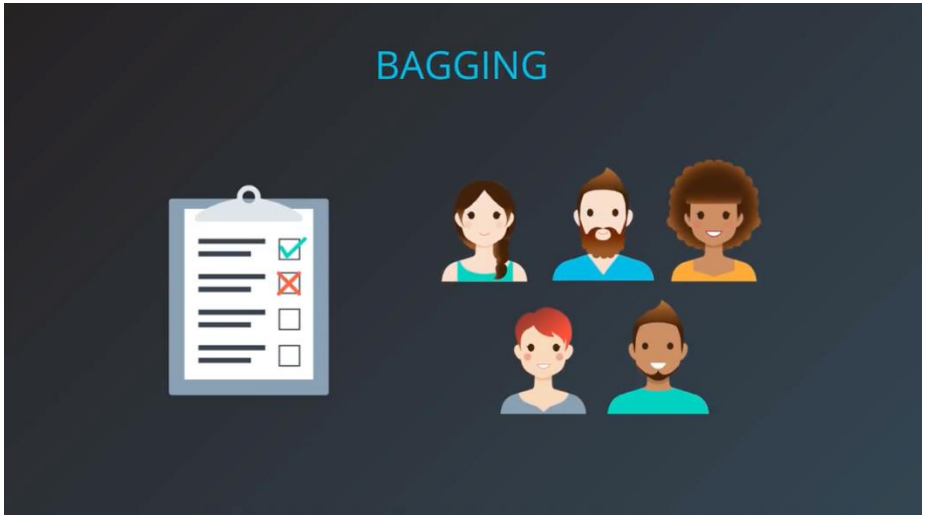
4.Combinaison des réponses par vote :

- Pour chaque question du test, vous allez regarder les réponses fournies par tous vos amis.
- Vous choisissez la réponse qui a été la plus fréquente parmi vos amis pour chaque question. C'est le principe du "vote majoritaire".

5.Résultat final :

- La combinaison finale de réponses pour le test est donc construite en utilisant les réponses majoritairement choisies par vos amis.

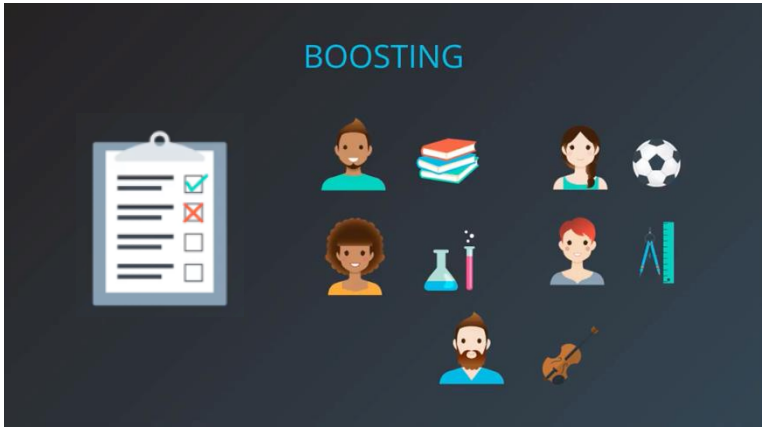
Bagging



Avantage de cette méthode :

- **Réduction de l'erreur :** Comme chaque ami (ou modèle) apporte sa propre perspective et manière de penser, les erreurs individuelles sont atténuées par le vote.
- **Diversité :** Les connaissances variées de vos amis aident à équilibrer les points faibles de chacun, produisant une meilleure réponse globale.

Boosting



Mise en Situation :

1.Commencement avec tous les amis :

- Vous demandez à vos amis de vous aider un par un, mais de manière séquentielle.

2.Premier tour :

- Le premier ami répond au test. Il fait de son mieux, mais comme tout le monde, il fait quelques erreurs.

3.Identifier les erreurs :

- Vous analysez ensemble ses réponses et identifiez les questions où il s'est trompé.

4.Deuxième tour :

- Un deuxième ami prend le relais, mais il se concentre particulièrement sur les questions où le premier ami a échoué. L'idée est qu'il corrigera ces erreurs spécifiques.

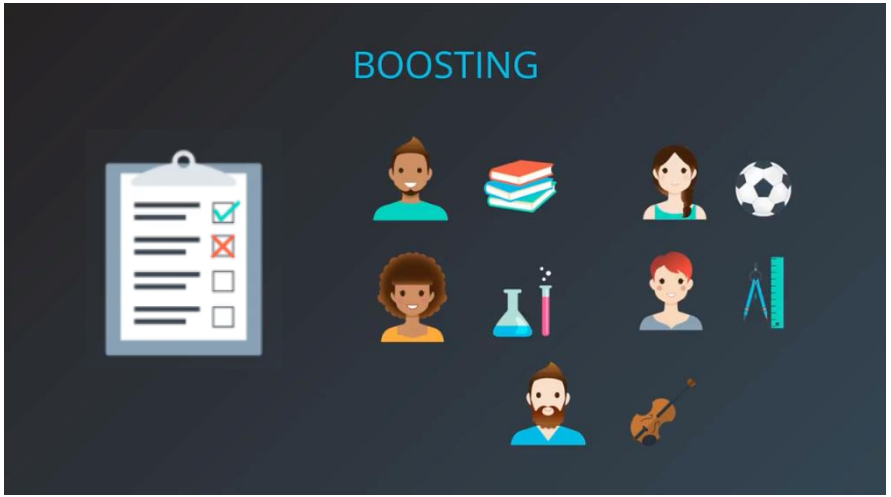
5.Répétition du processus :

- Ce processus continue avec plusieurs autres amis, où chaque nouvel ami se concentre sur les corrections des erreurs des tours précédents.

6.Combinaison des réponses :

- Enfin, vous combinez toutes les réponses, mais en prenant surtout en compte les corrections apportées à chaque étape. Les réponses finales sont fignolées pour corriger au maximum les faiblesses précédentes.

Boosting



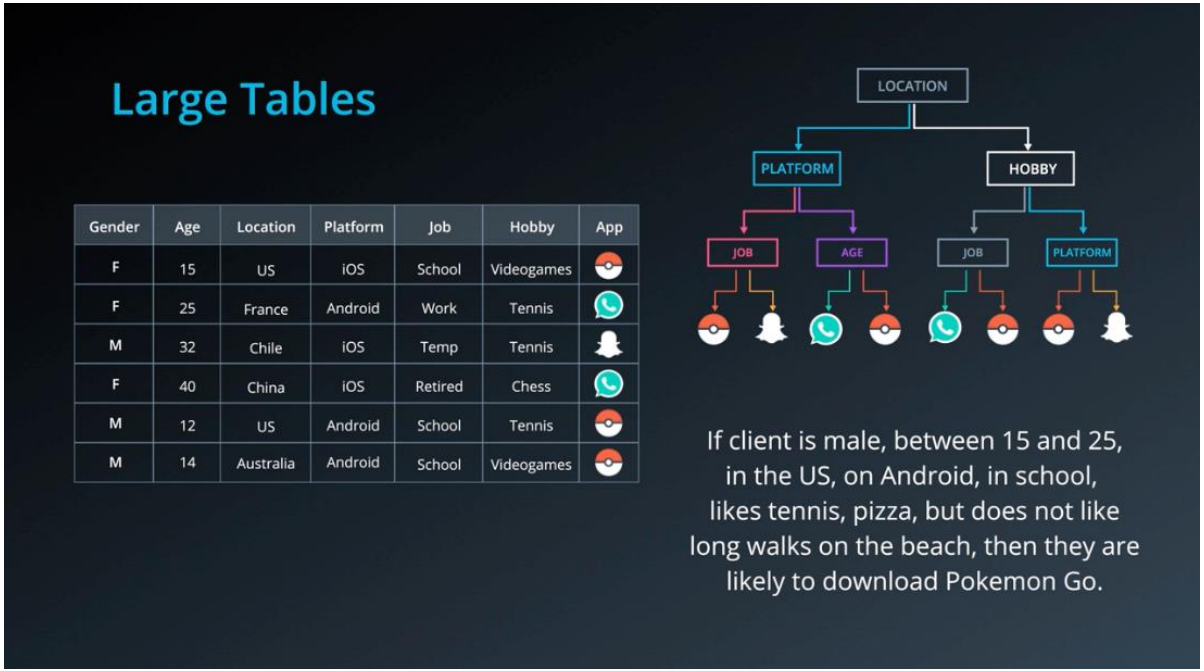
Avantage du Boosting :

- **Amélioration continue :** Chaque ami apprend des erreurs des précédents, ce qui améliore progressivement les réponses sur l'ensemble du test.
- **Réduction du biais :** En se concentrant sur les erreurs, le boosting réduit les biais et affine le modèle final pour plus de précision.

Apprenants Faibles et Apprenants Puissants

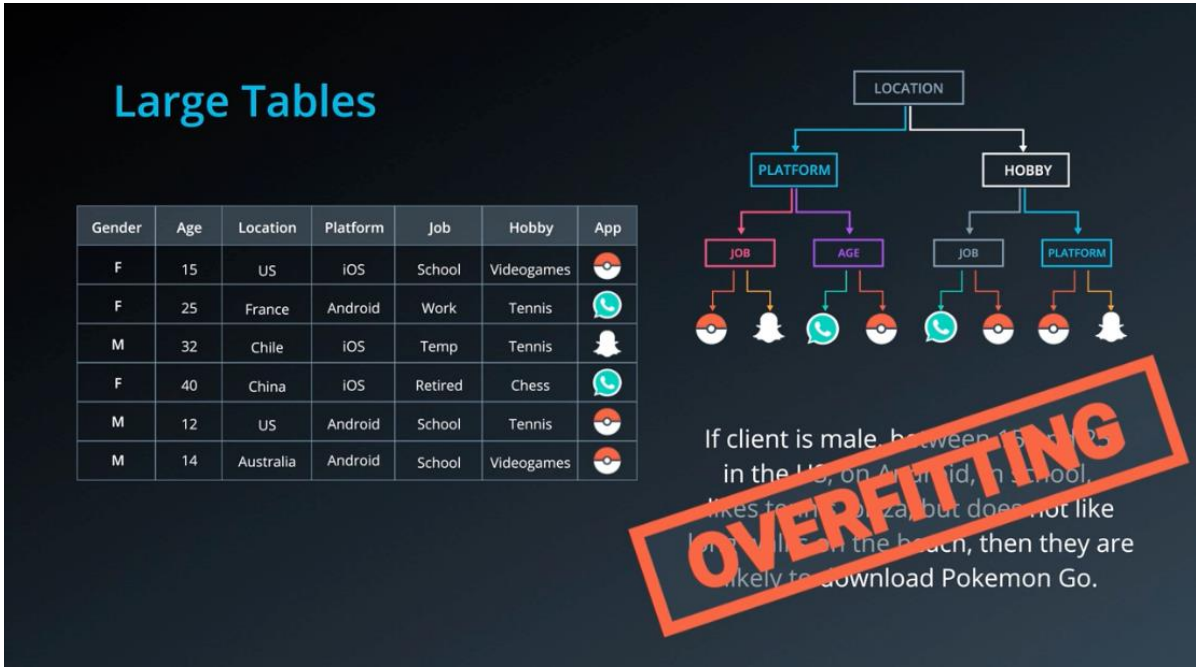
Les méthodes d'ensemble visent à combiner plusieurs apprenants faibles (weak learners) pour créer des apprenants puissants (strong learner).

Random Forests (Les forêts aléatoires)

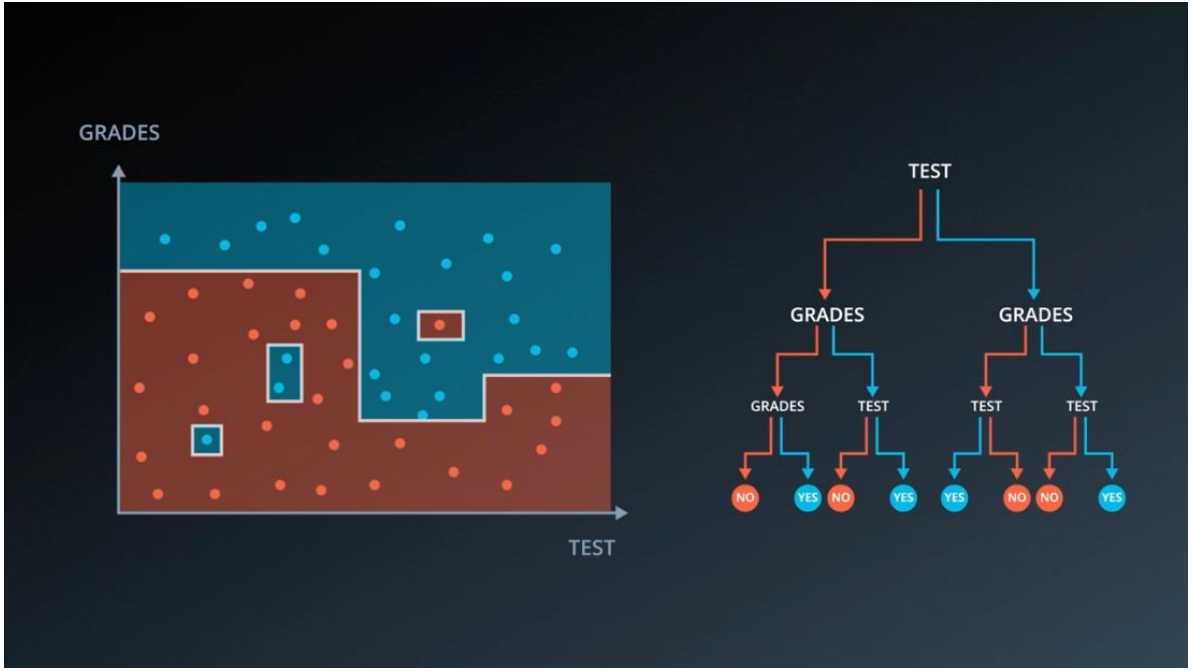


Les arbres de décision ont tendance à surapprendre les données, apprenant souvent par cœur plutôt que de généraliser.

Random Forests (Les forêts aléatoires)

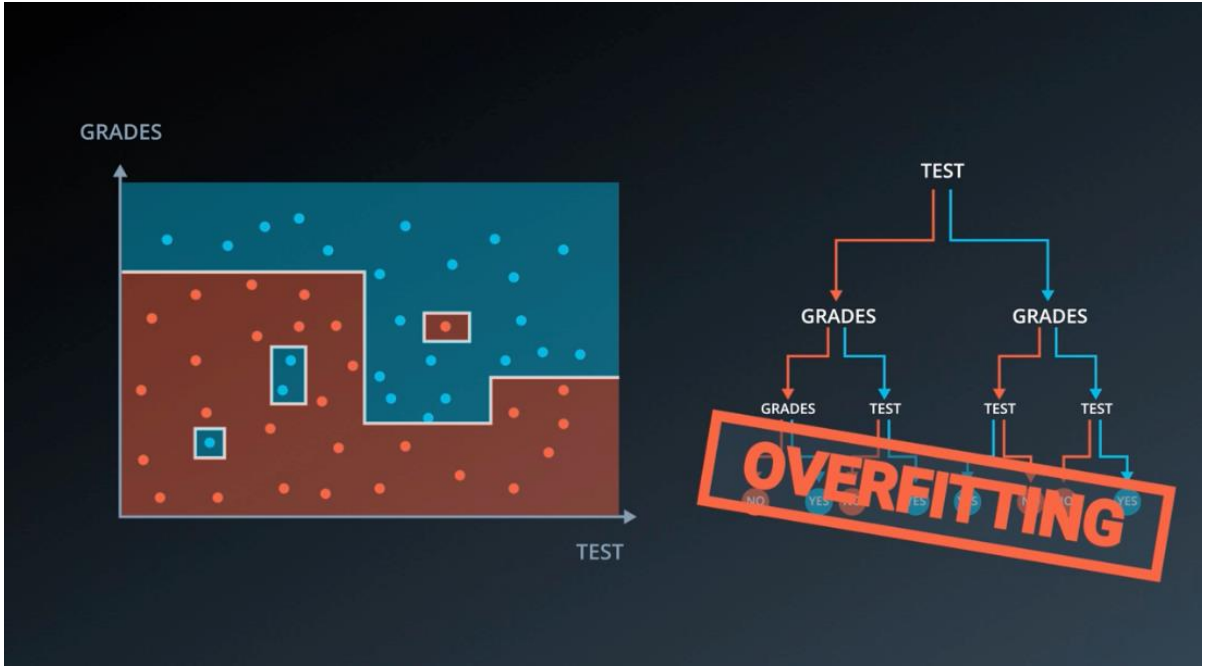


Random Forests (Les forêts aléatoires)



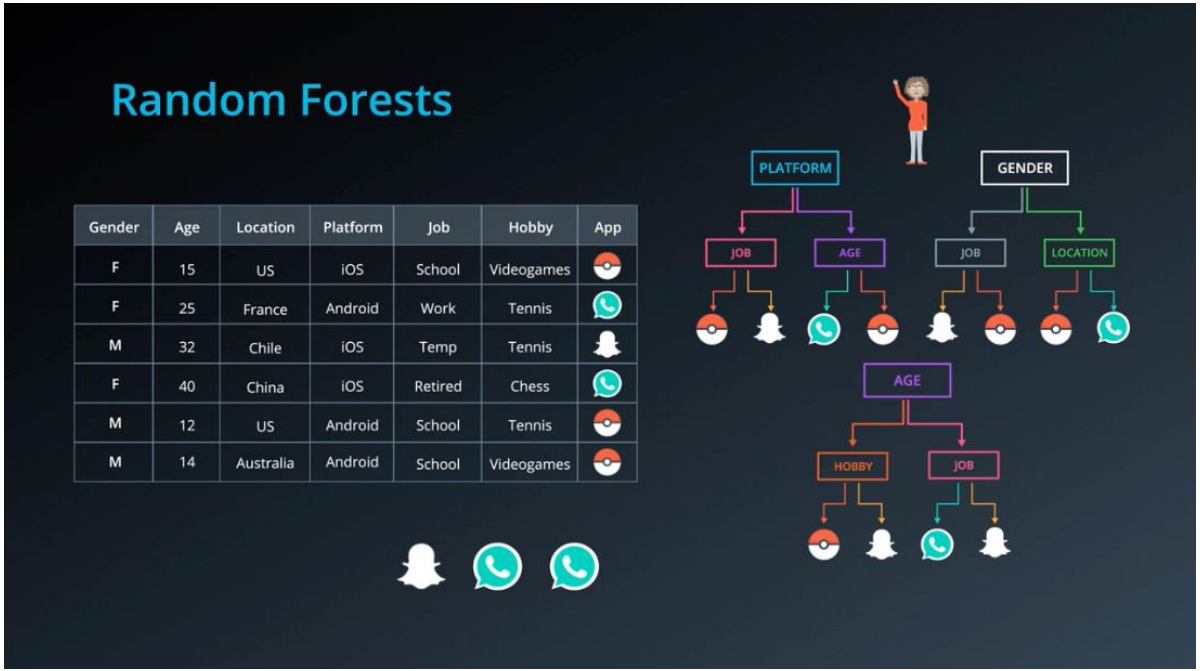
Les arbres de décision ont tendance à surapprendre les données, apprenant souvent par cœur plutôt que de généraliser.

Random Forests (Les forêts aléatoires)



Comment prévenir le surapprentissage dans les arbres de décision ?

Random Forests (Les forêts aléatoires)



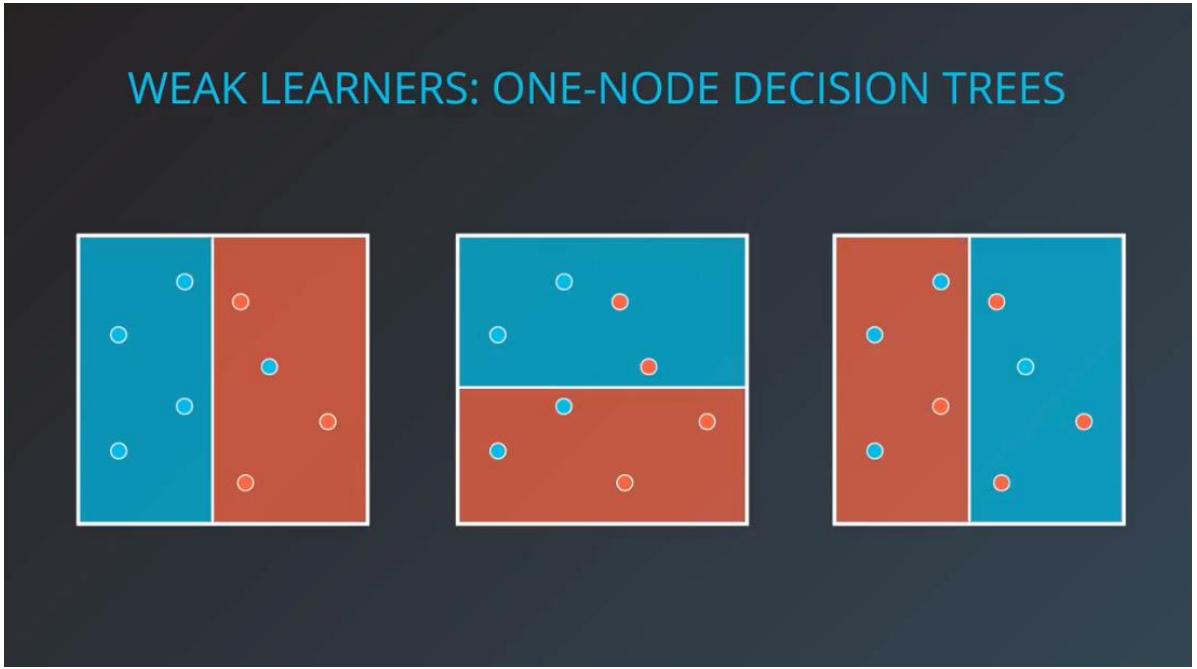
On sélectionne aléatoirement des colonnes pour construire des arbres de décision distincts pour chaque tirage.

Pour prédire, on utilise le vote majoritaire des arbres ainsi construits.

Nous découvrirons dans le cours qu'il existe une méthode plus efficace pour sélectionner les colonnes lors de la construction des arbres de décision, plutôt que de le faire de manière aléatoire.

Random Forests (Les forêts aléatoires) - Bagging

Random Forests (Les forêts aléatoires) - Bagging

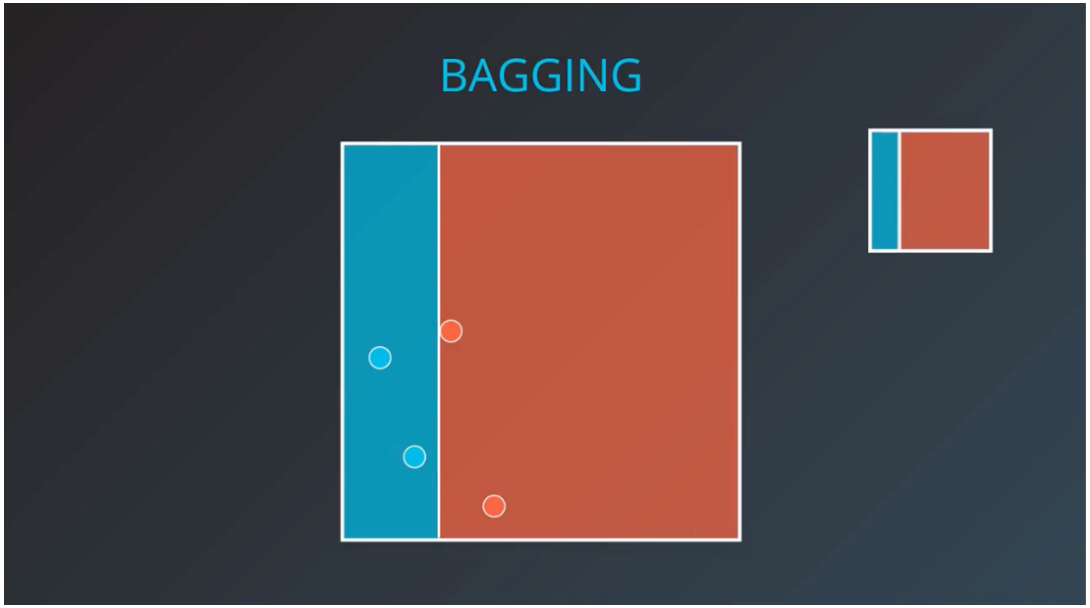


Nos apprenants faibles sont des arbres de décision qui possèdent un seul nœud de décision.

Random Forests (Les forêts aléatoires) - Bagging

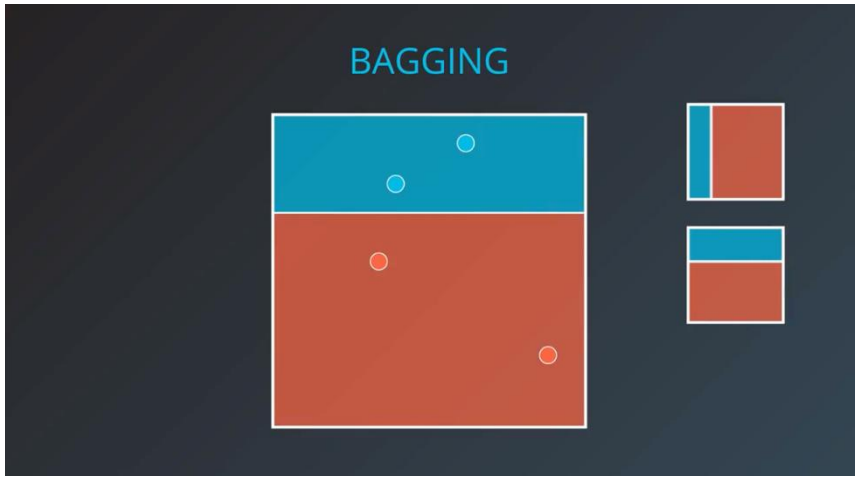
En règle générale, notre ensemble de données est de grande dimension, ce qui rend l'entraînement des apprenants faibles sur l'ensemble des données peu rentable en termes de ressources et de temps. Ainsi, nous optons pour des sous-ensembles de données (batchs) afin de former chaque apprenant faible. Ensuite, nous examinerons les méthodes pour fusionner ces apprenants.

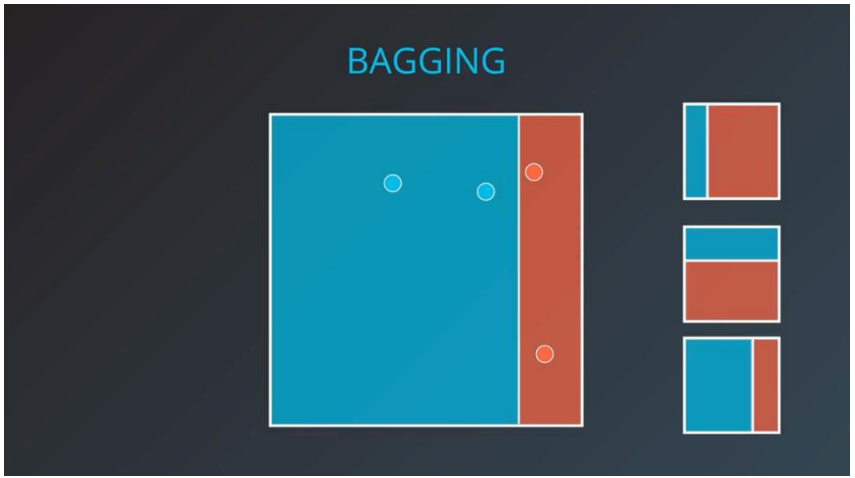
Random Forests (Les forêts aléatoires) - Bagging



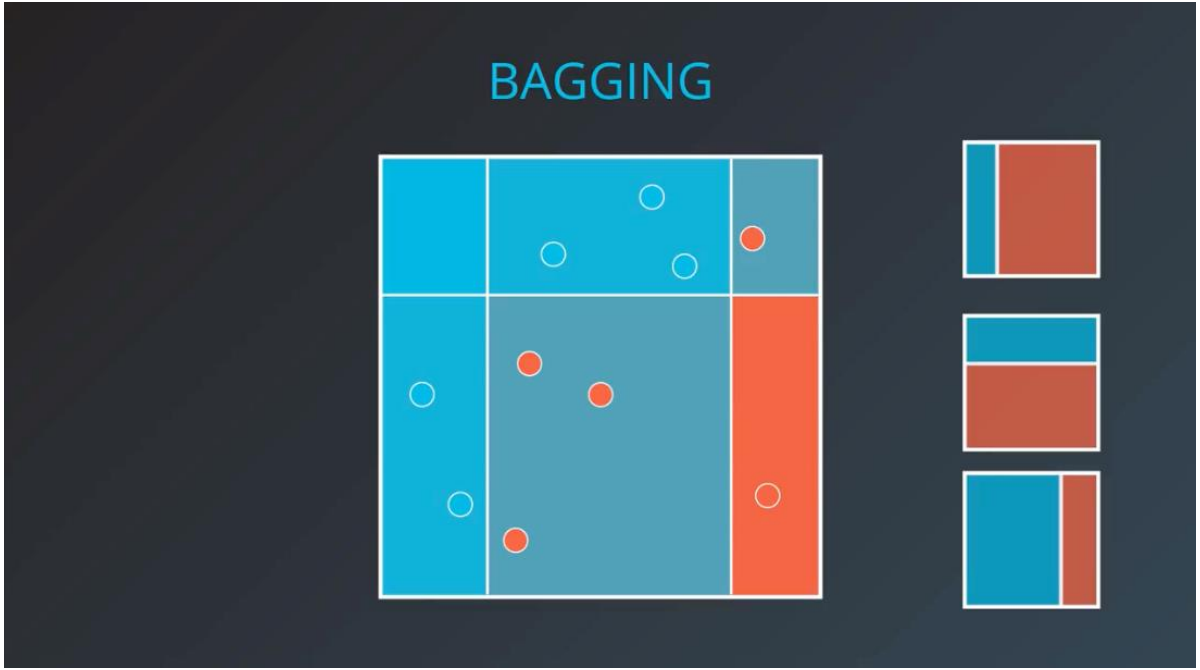
On entraine un apprenant faible (weak learner) sur un sous-ensemble de données puis on mémorise le modèle

Random Forests (Les forêts aléatoires) - Bagging



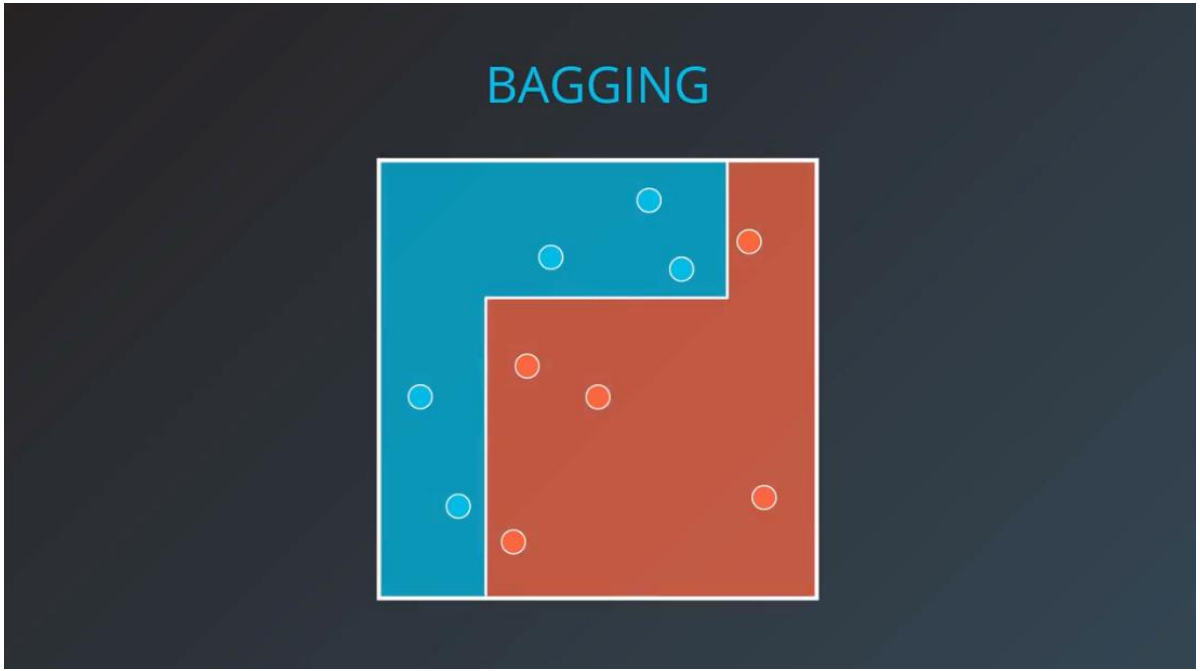


Random Forests (Les forêts aléatoires) - Bagging



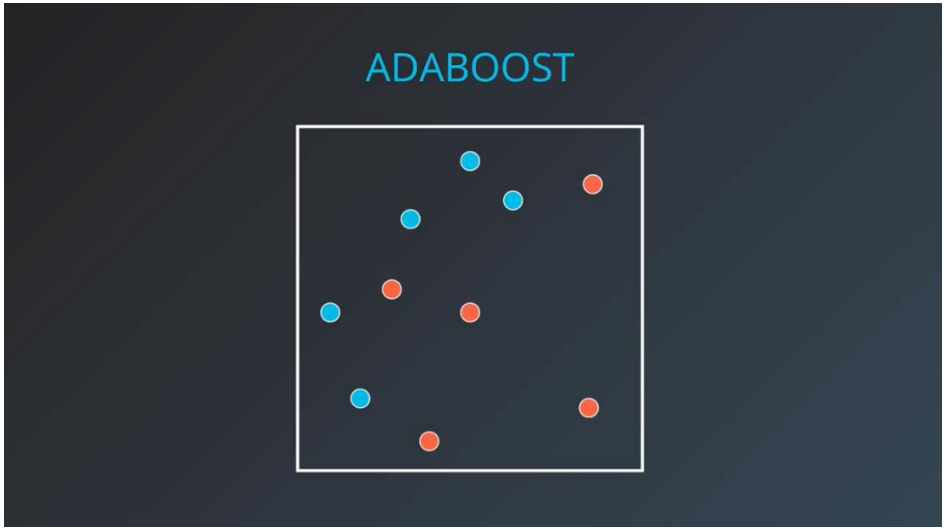
Nous combinons les modèles en utilisant un système de vote.

Random Forests (Les forêts aléatoires) - Bagging



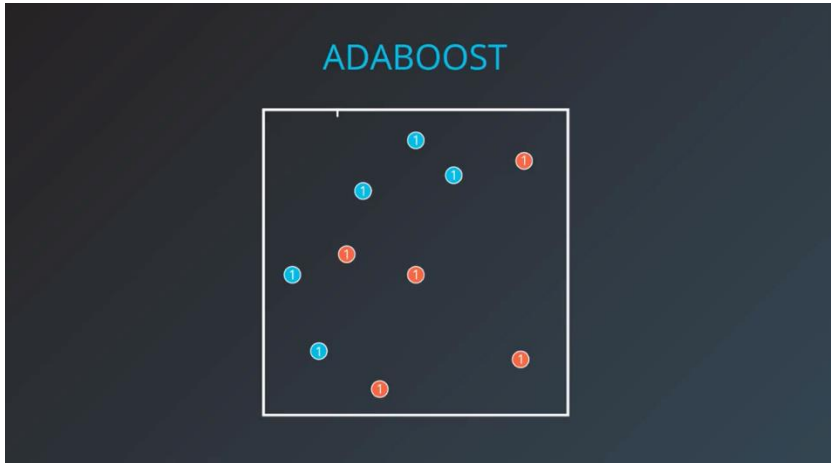
Le modèle final obtenu après la combinaison des apprenants faibles

AdaBoost

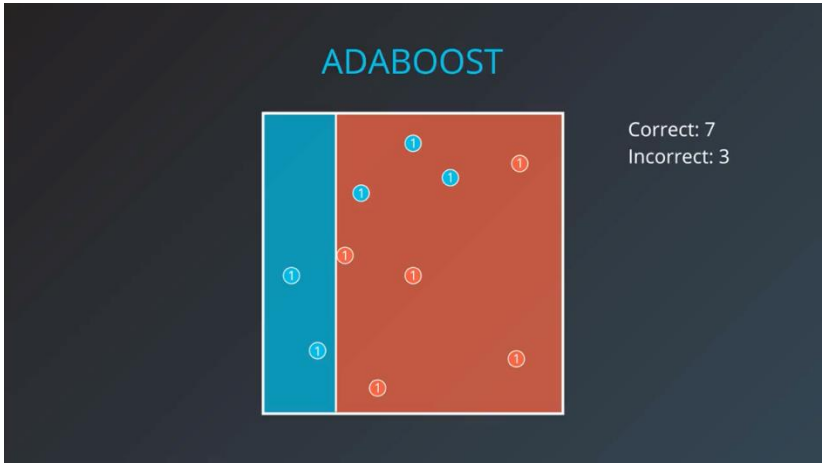


AdaBoost (Adaptive Boosting) est un algorithme d'apprentissage automatique qui combine plusieurs apprenants faibles pour créer un modèle puissant. En itérant, AdaBoost ajuste les poids des observations en fonction de leur performance, mettant l'accent sur celles mal classées par les modèles précédents. Ce processus permet à chaque nouvel apprenant de corriger les erreurs des précédents, améliorant ainsi la précision globale du modèle final.

AdaBoost

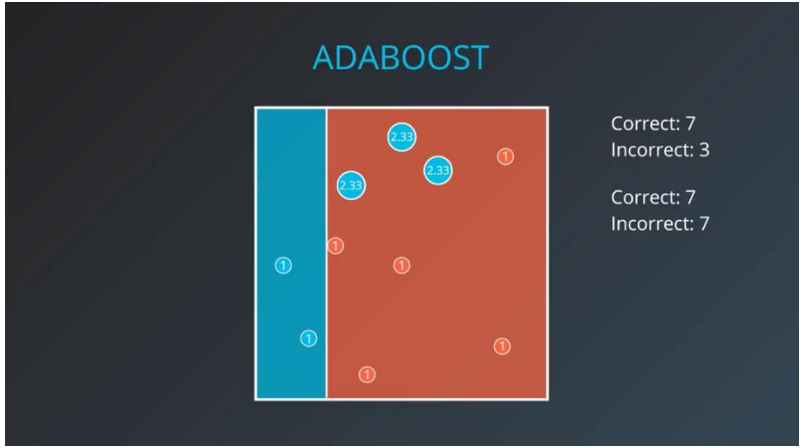


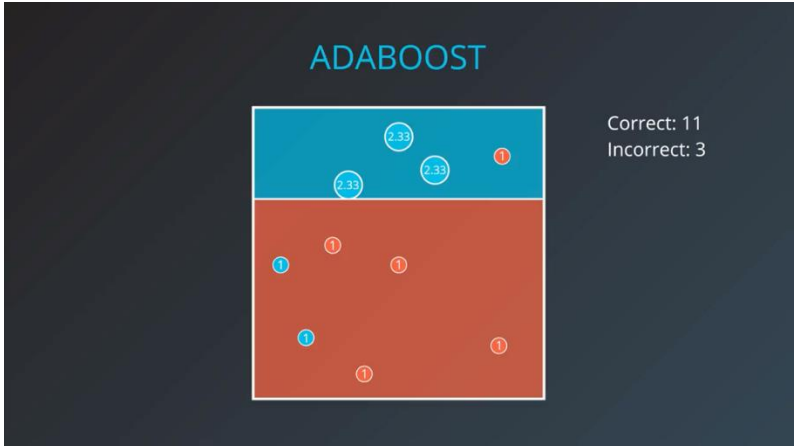
On attribue un poids de 1 à chaque point de données que l'on souhaite classer.



Nous entraînons un premier apprenant faible. Ensuite, l'objectif est de minimiser la somme des poids des points mal classés.

AdaBoost

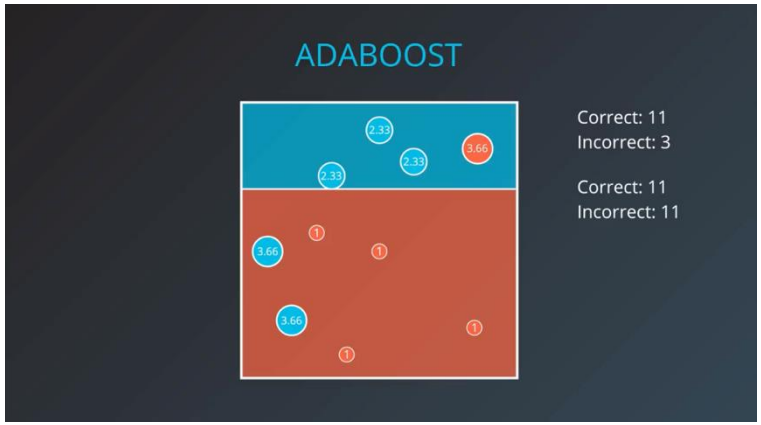




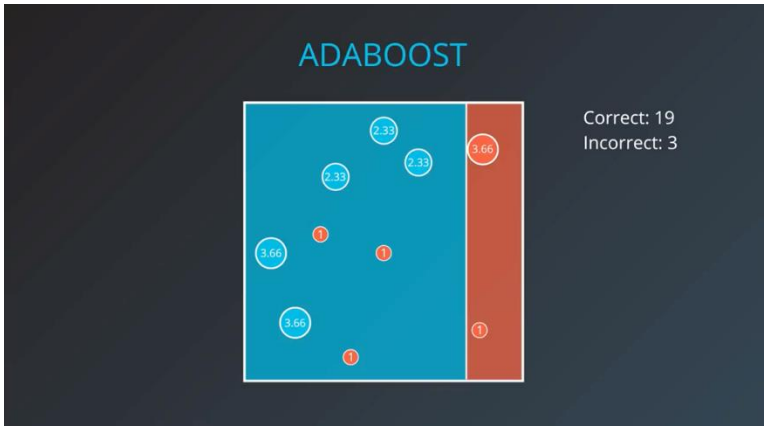
Pour orienter l'attention du modèle vers les points mal classés, on leur attribue une pondération plus élevée. Cette décision est prise afin d'assurer que le jeu de données d'apprentissage présente un équilibre entre les points correctement classés et ceux qui ne le sont pas.

Nous entraînons le deuxième apprenant faible, puis nous calculons la somme des poids des points bien classés et celle des points mal classés.

AdaBoost

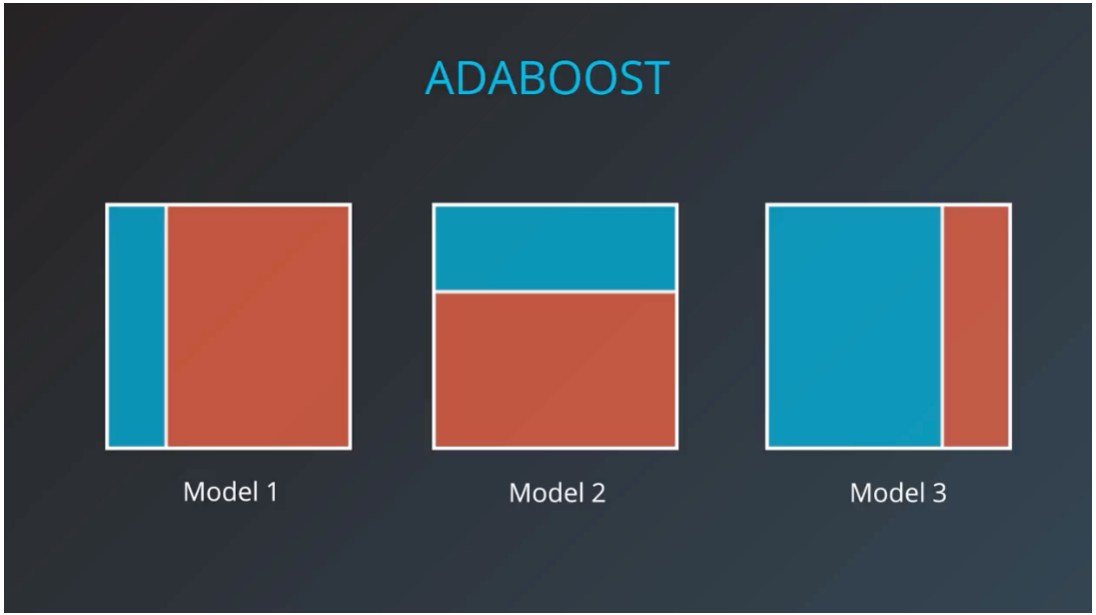


Nous ajustons les poids des points correctement classés et mal classés pour que le modèle puisse se concentrer davantage sur ceux qui sont mal classés.



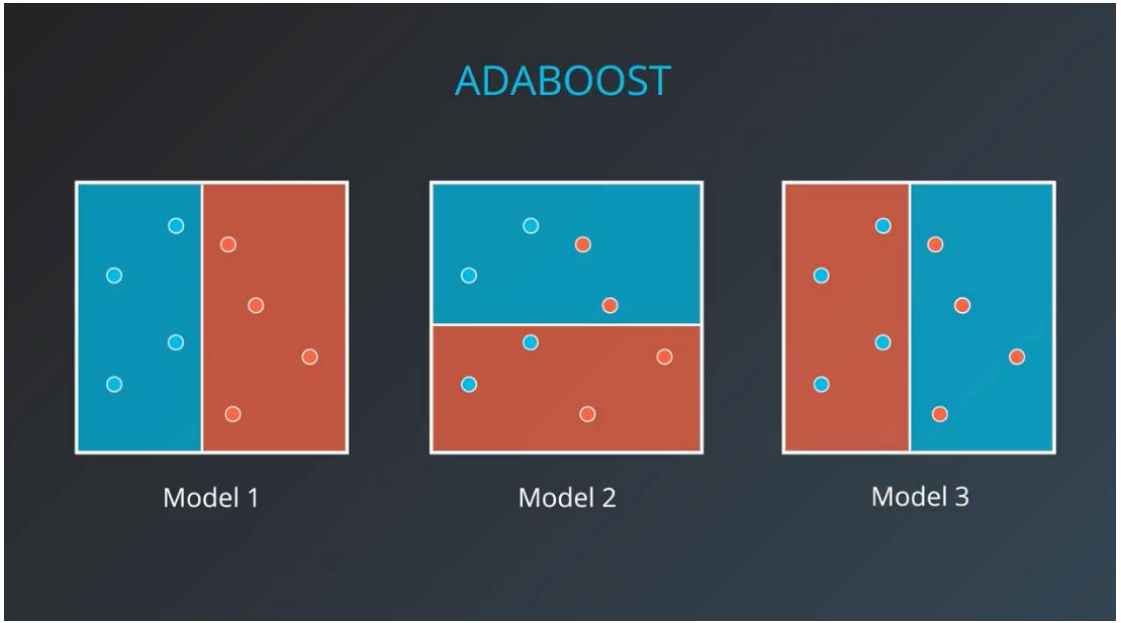
Nous pouvons répéter ce processus, mais nous allons nous arrêter ici pour le moment.

AdaBoost



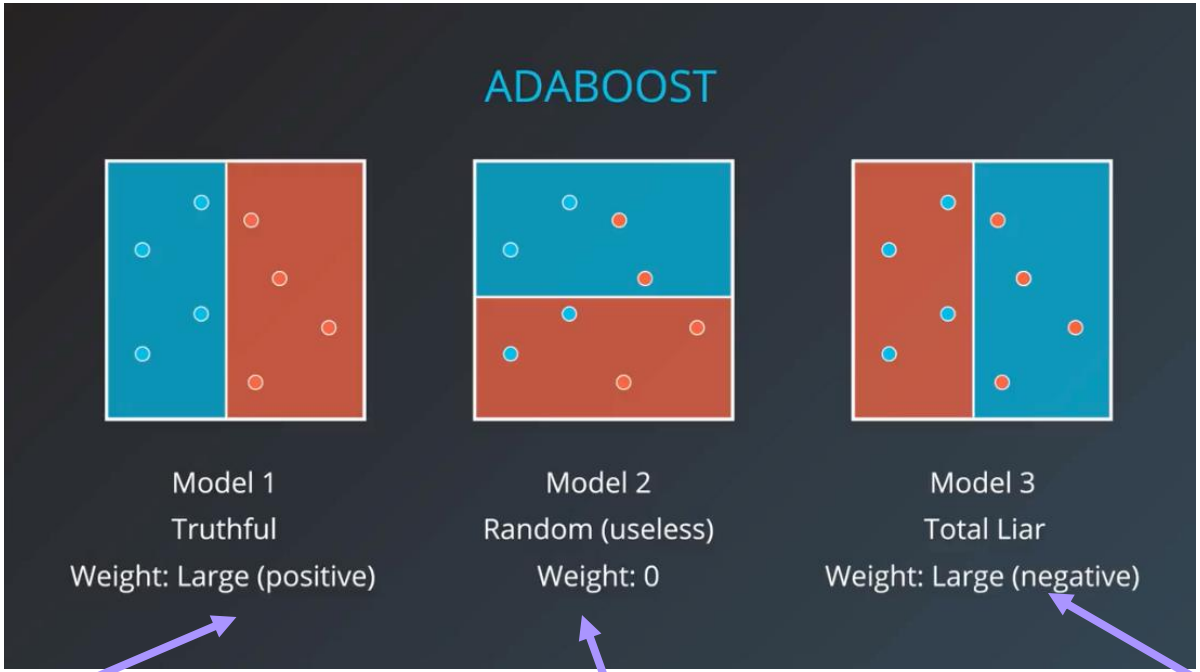
Après avoir entraîné trois apprenants faibles, la question qui se pose est : comment les combiner ?

AdaBoost: combinaison des apprenants faibles



Quel modèle est le moins performant ?

AdaBoost: combinaison des apprenants faibles

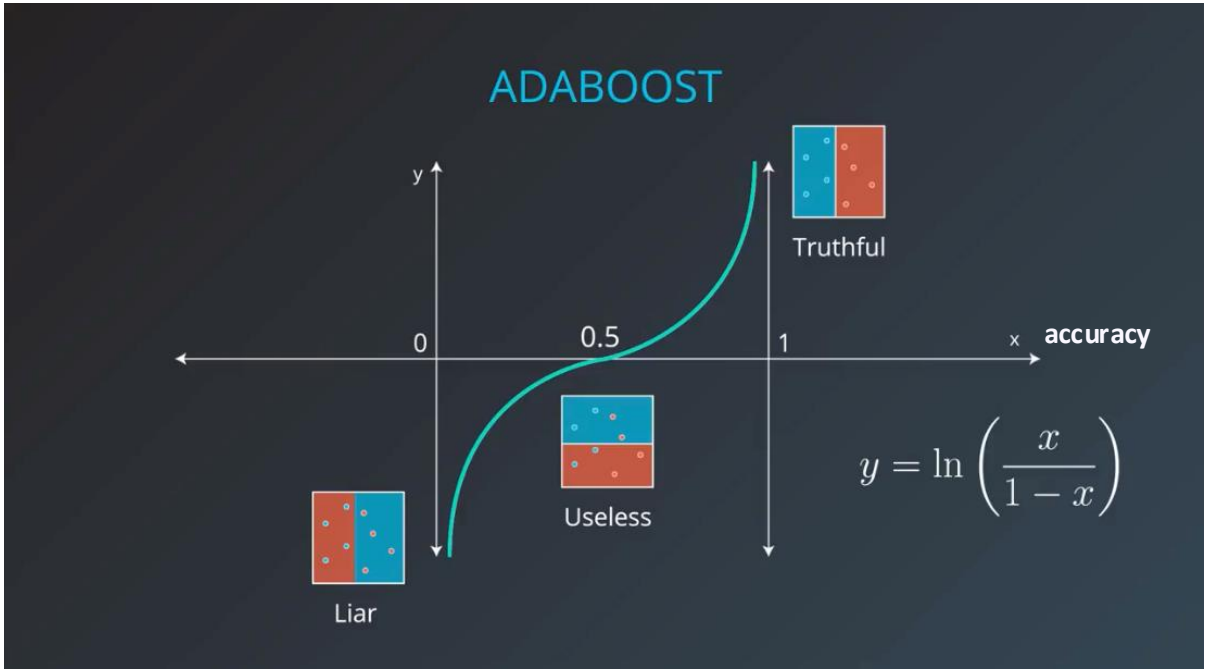


Ce modèle sépare correctement la majorité des points avec un poids important et positif, indiquant une bonne performance.

Ce modèle fait une séparation aléatoire des points et est considéré inutile, donc il reçoit un poids de zéro. Il n'influence pas la combinaison finale.

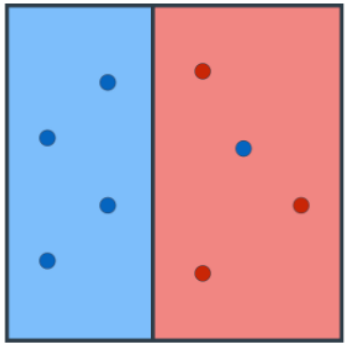
Ce modèle classe les points de manière incorrecte. Il reçoit un poids négatif important, ce qui signifie qu'il influence la décision finale de manière opposée à sa prédiction initiale.

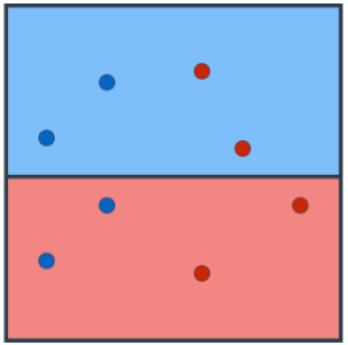
AdaBoost: combinaison des apprenants faibles

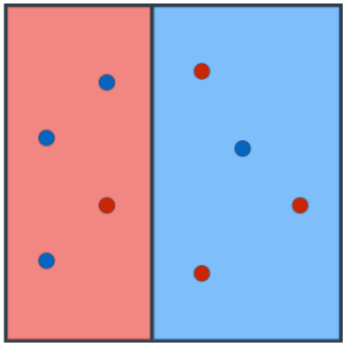


Quiz: calculez le poids de chaque modèle ci-dessous

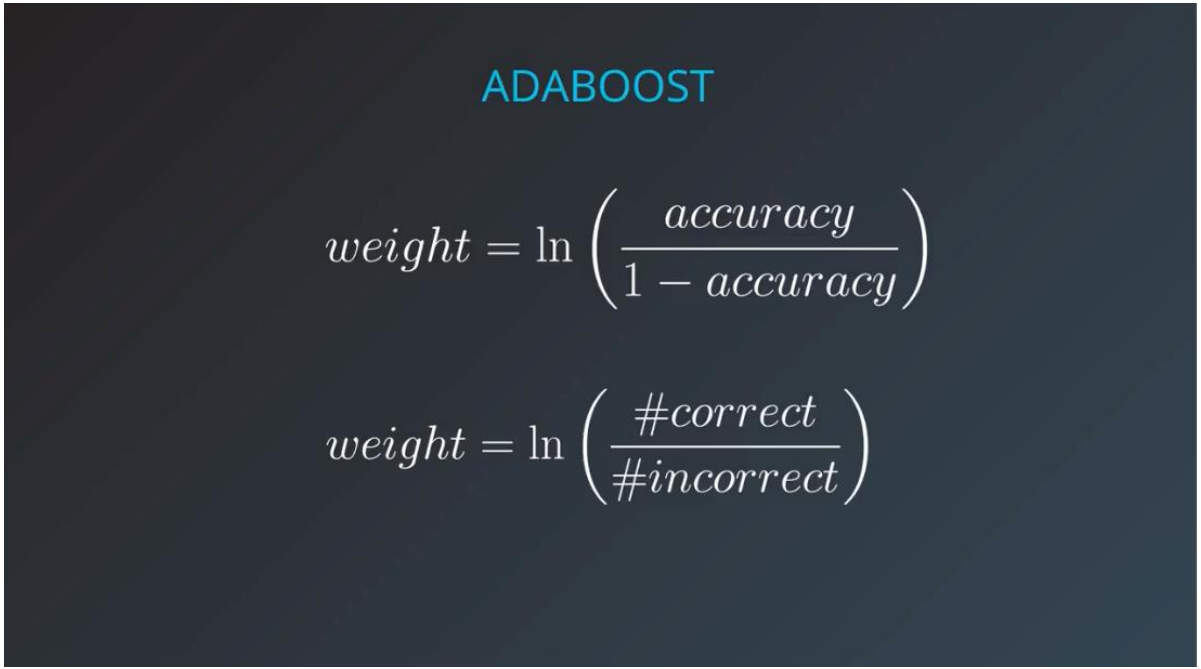
$$weight = \ln\left(\frac{accuracy}{1 - accuracy}\right)$$





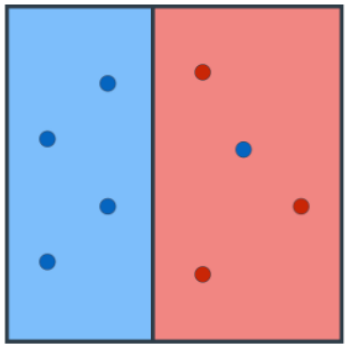


AdaBoost: combinaison des apprenants faibles

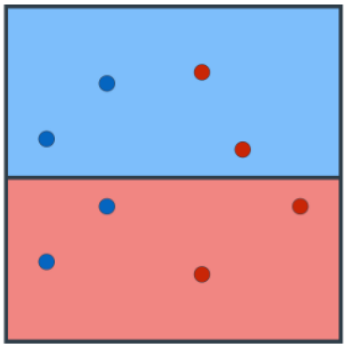


Quiz: calculez le poids de chaque modèle ci-dessous

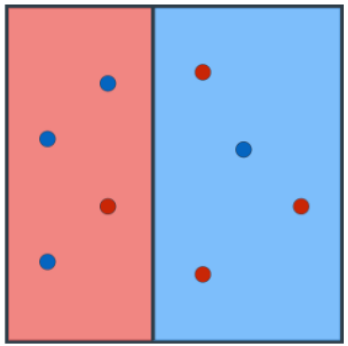
$$weight = \ln\left(\frac{accuracy}{1 - accuracy}\right)$$



$$\ln\left(\frac{7}{1}\right) = 1.95$$

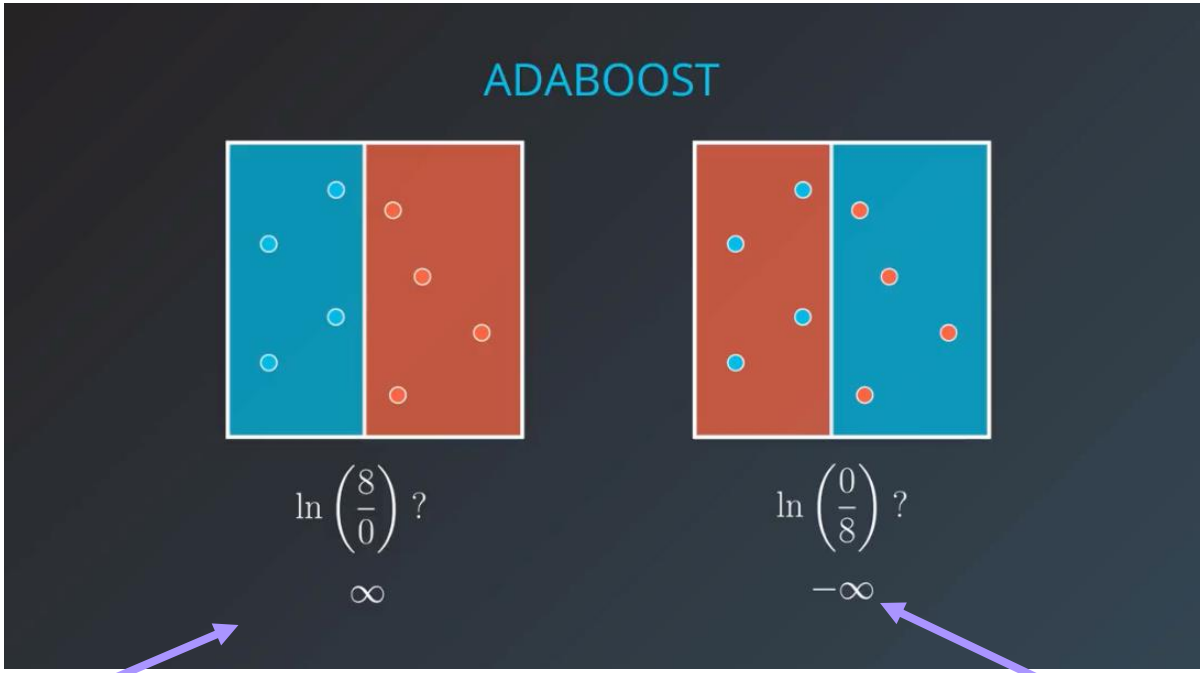


$$\ln\left(\frac{4}{4}\right) = 0$$



$$\ln\left(\frac{2}{6}\right) = -1.099$$

AdaBoost: combinaison des apprenants faibles



On fait 100% confiance à ce modèle

On accorde 100 % de confiance à l'opposé des prédictions de ce modèle.

En pratique, ces situations sont très rares.

Exercice

https://github.com/elhidali/EPISEN-2024