Arbres de Décision et Forêts Aléatoires

I : Arbres de Décision - Indice de Gini

Arbres de Décision : Une Approche de Classification Supervisée

Définition : Les **arbres de décision** sont des modèles d'apprentissage supervisé permettant de prédire une valeur cible en appliquant une séquence de tests sur les attributs des données.

- Ils sont particulièrement utilisés en **classification**, où ils attribuent une classe à une observation en suivant un chemin dans l'arbre.
- Ils sont également adaptés à la **régression**, où ils prédisent une valeur numérique en moyenne sur les feuilles.
- L'apprentissage se fait en construisant un arbre qui divise les données de manière optimale en minimisant l'impureté (ex. Indice de Gini ou Entropie).

Types de Problèmes : Classification et Régression

Arbres de Décision :

- **Classification supervisée** :
  - Les classes sont qualitatives (exemple : Oui/Non, chat/chien, ...).
  - Les feuilles de l'arbre indiquent la classe la plus probable.
- **Régression supervisée** :
  - La sortie est une variable numérique (exemple : un prix, un salaire, ...).
  - Les feuilles indiquent en général la *moyenne* des valeurs de la cible.

Méthode Générale

Quel que soit le type (classification ou régression), l'arbre effectue des **tests sur les attributs** successifs pour partitionner au mieux les données, jusqu'à aboutir à des **feuilles cohérentes**.

Qu'est-ce qu'un Arbre de Décision ?

Définition :

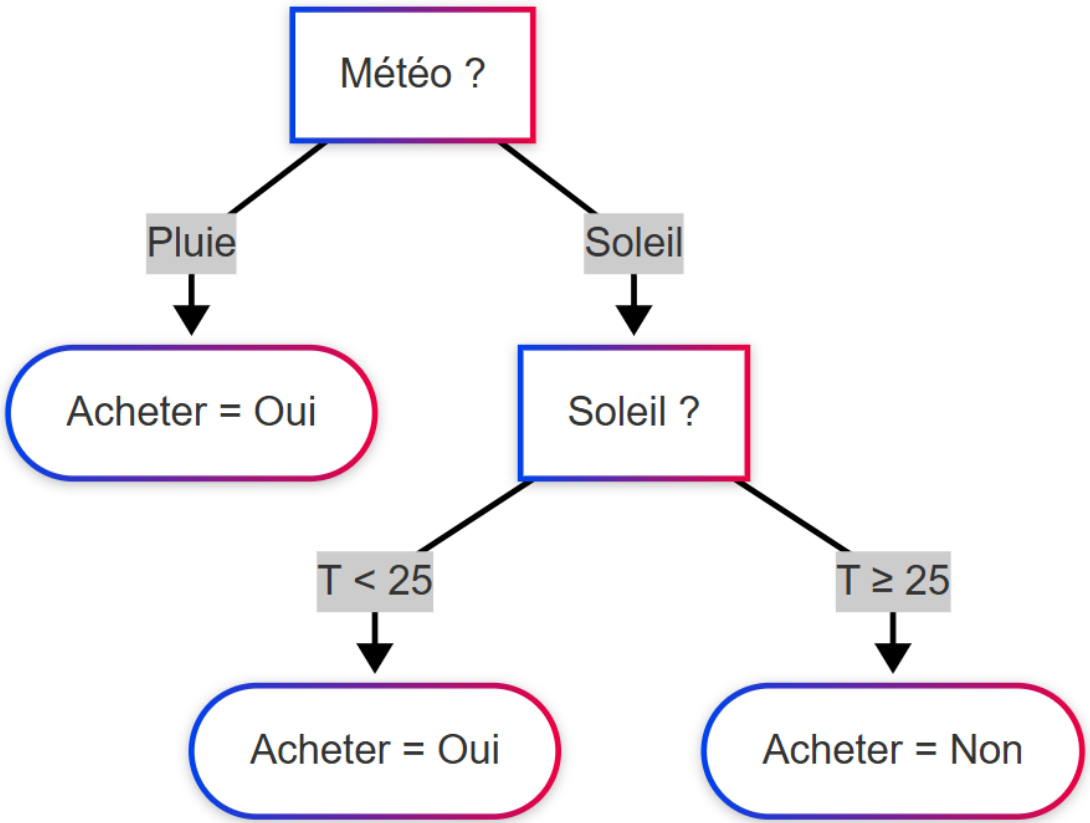
- Un **arbre de décision** est une structure en forme d'arbre où chaque **nœud** représente un **test** sur un attribut (par exemple, Météo=Pluie ?).
- Chaque **branche** correspond à un résultat possible de ce test (oui, non, ou d'autres valeurs).
- Les **feuilles** indiquent la décision finale ou la classe prédite (par exemple, "Acheter=Oui").

Caractéristiques principales

- **Facile à interpréter** : la décision se lit depuis la racine jusqu'à la feuille.
- **Gère** aussi bien des attributs numériques que qualitatifs.

Arbre de Décision : Acheter ou non un parapluie

Exemple :



L'Indice de Gini : Détails et Sélection d'Attribut

Définition : L'Indice de Gini mesure l'**impureté** d'un ensemble de données S.

$$Gini(S) = 1 - \sum_{i=1}^{C} (p_i)^2$$

- S : ensemble d'exemples.
- C : nombre de classes (ex. Oui / Non).
- p_i : proportion d'exemples de la classe i dans S.

Interprétation

Plus le Gini est **faible**, plus l'ensemble est **pur** (c.-à-d. dominé par une seule classe).

Calcul de l'indice $Gini_{après}$ d'un attribut A :

$$Gini_{\text{après}}(A) = \sum_{k=1}^{K} \left(\frac{|S_k|}{|S|} \times Gini(S_k) \right)$$

- S_k : sous-ensemble des données où la valeur de l'attribut A prend une certaine modalité (ou se trouve dans un certain intervalle).
- $|S_k|$: nombre d'exemples dans le sous-ensemble S_k.

Gain de Gini :

$$\text{Gain}(A) = Gini(S) \; - \; Gini_{\text{après}}(A)$$

Quand choisir l'attribut ?

On choisit l'attribut A qui **maximise le Gain de Gini**, c'est-à-dire celui qui **réduit le plus l'impureté**.

Exemple : Prédire "Acheter un Parapluie" ?

Données (8 exemples, 2 attributs) :

Météo	**Température (°C)**	**Acheter ? (Oui/Non)**
Soleil	35	Non
Soleil	28	Non
Soleil	20	Oui
Pluie	18	Oui
Pluie	22	Oui
Nuage	19	Oui
Pluie	16	Oui
Nuage	25	Non

- **Attributs** :
  - **Météo** : {Soleil, Pluie, Nuage}
  - **Température** : valeur numérique (de 16° à 35°, ici).
- **Classe** : Acheter *Parapluie* ? (Oui ou Non).

Étape 1 : Gini de l'Ensemble Global

Total : 8 exemples

$$\text{Classe Oui} = 5 \quad (\text{Soleil} : 1, \text{Pluie} : 3, \text{Nuage} : 1)$$

$$\text{Classe Non} = 3 \quad (\text{Soleil} : 2, \text{Nuage} : 1)$$

$$p(\text{Oui}) = \frac{5}{8} = 0.625, \quad p(\text{Non}) = \frac{3}{8} = 0.375$$

$$Gini(S) = 1 - \left(0.625^2 + 0.375^2\right) = 0.46875$$

Impureté initiale

Le Gini vaut ≈ 0.47. Nous devons **réduire** cette impureté en choisissant un bon attribut.

Étape 2 : Division par la Météo (1/2)

Sous-ensembles :

- **Soleil** (3 exemples) :

$$\{(35, Non), (28, Non), (20, Oui)\} \Rightarrow 2 \text{ Non}, 1 \text{ Oui}$$

- **Pluie** (3 exemples) :

$$\{(18, Oui), (22, Oui), (16, Oui)\} \Rightarrow 3 \text{ Oui}, 0 \text{ Non}$$

- **Nuage** (2 exemples) :

$$\{(19, Oui), (25, Non)\} \Rightarrow 1 \text{ Oui}, 1 \text{ Non}$$

Étape 2 : Division par la Météo (2/2)

Calcul des Gini de chaque sous-ensemble de l'attribut Météo :

$$Gini(\text{Soleil}) = 1 - \left(\left(\tfrac{1}{3}\right)^2 + \left(\tfrac{2}{3}\right)^2\right) = 1 - (0.11 + 0.44) = 0.45$$

$$Gini(\text{Pluie}) = 1 - (1^2 + 0^2) = 0$$

$$Gini(\text{Nuage}) = 1 - \left(\left(\tfrac{1}{2}\right)^2 + \left(\tfrac{1}{2}\right)^2\right) = 1 - (0.25 + 0.25) = 0.5$$

$$\boxed{Gini_{\text{après}}(\text{Météo}) = \frac{3}{8} \times 0.45 + \frac{3}{8} \times 0 + \frac{2}{8} \times 0.5 = 0.29375}$$

Gain(Météo) = 0.46875 - 0.29375 = 0.175

Conclusion : Le Gini baisse à ≈ 0.29. Le gain est de 0.175.

Étape 3 : Division par la Température (1/2)

Testons un seuil **Température** < 21, on aura deux groupes S_1 et S_2 :

$$S_1 = \{(Soleil, 20, Oui), (Pluie, 18, Oui), (Pluie, 16, Oui), (Nuage, 19, Oui)\}$$

$$S_2 = \{(Soleil, 35, Non), (Soleil, 28, Non), (Pluie, 22, Oui), (Nuage, 25, Non)\}$$

Étape 3 : Division par la Température (2/2)

$$Gini(S_1) = 1 - \left(\left(\frac{4}{4}\right)^2 + \left(\frac{0}{4}\right)^2\right) = 0$$

$$Gini(S_2) = 1 - \left(\left(\frac{3}{4}\right)^2 + \left(\frac{1}{4}\right)^2\right) = 0.375$$

$$\boxed{Gini_{\text{après}}(\text{Temp}) = \frac{4}{8} \times 0.375 + \frac{4}{8} \times 0 = 0.1875}$$

Gain(Temp<21) = 0.46875 - 0.1875 = 0.2812

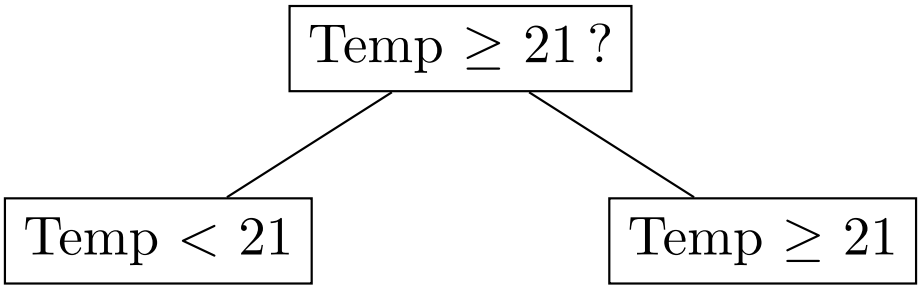
Conclusion : Le gain est plus grand (0.2812) qu'avec Météo (0.175).

Meilleur attribut pour la racine

$$\textbf{Gain(Mété)} = 0.175, \quad \textbf{Gain(Temp<21) = 0.2812}$$

Décision

Température maximise le gain en Gini : c'est donc l'attribut choisi pour la **racine** de l'arbre.



Branche "temp < 21"

- $\mathbf{S_1}$ (temp < 21) :

  $\{(\text{Soleil}, 20, \mathit{Oui}), (\text{Pluie}, 18, \mathit{Oui}), (\text{Pluie}, 16, \mathit{Oui}), (\text{Nuage}, 19, \mathit{Oui})\}$

  $\Rightarrow 100\,\%$ «Oui» $\Rightarrow$ Feuille = «Oui».

$$\mathit{Gini}(S_1) = 1 - \left(\left(\tfrac{4}{4}\right)^2 + \left(\tfrac{0}{4}\right)^2\right) = 0$$

Remarque

Les feuilles pures, pas besoin de séparation par Météo !

Branche "temp ≥ 21"

$\mathbf{S_2}$ (temp ≥ 21) :

$\{(\text{Soleil}, 35, Non), (\text{Soleil}, 28, Non), (\text{Pluie}, 22, Oui), (\text{Nuage}, 25, Non)\}$

$$Gini(S_2) = 1 - \left(\left(\tfrac{3}{4}\right)^2 + \left(\tfrac{1}{4}\right)^2\right) = 0.375$$

Remarque

On doit **poursuivre** la séparation par la Météo.

Branche «temp ≥ 21» : séparation par Météo

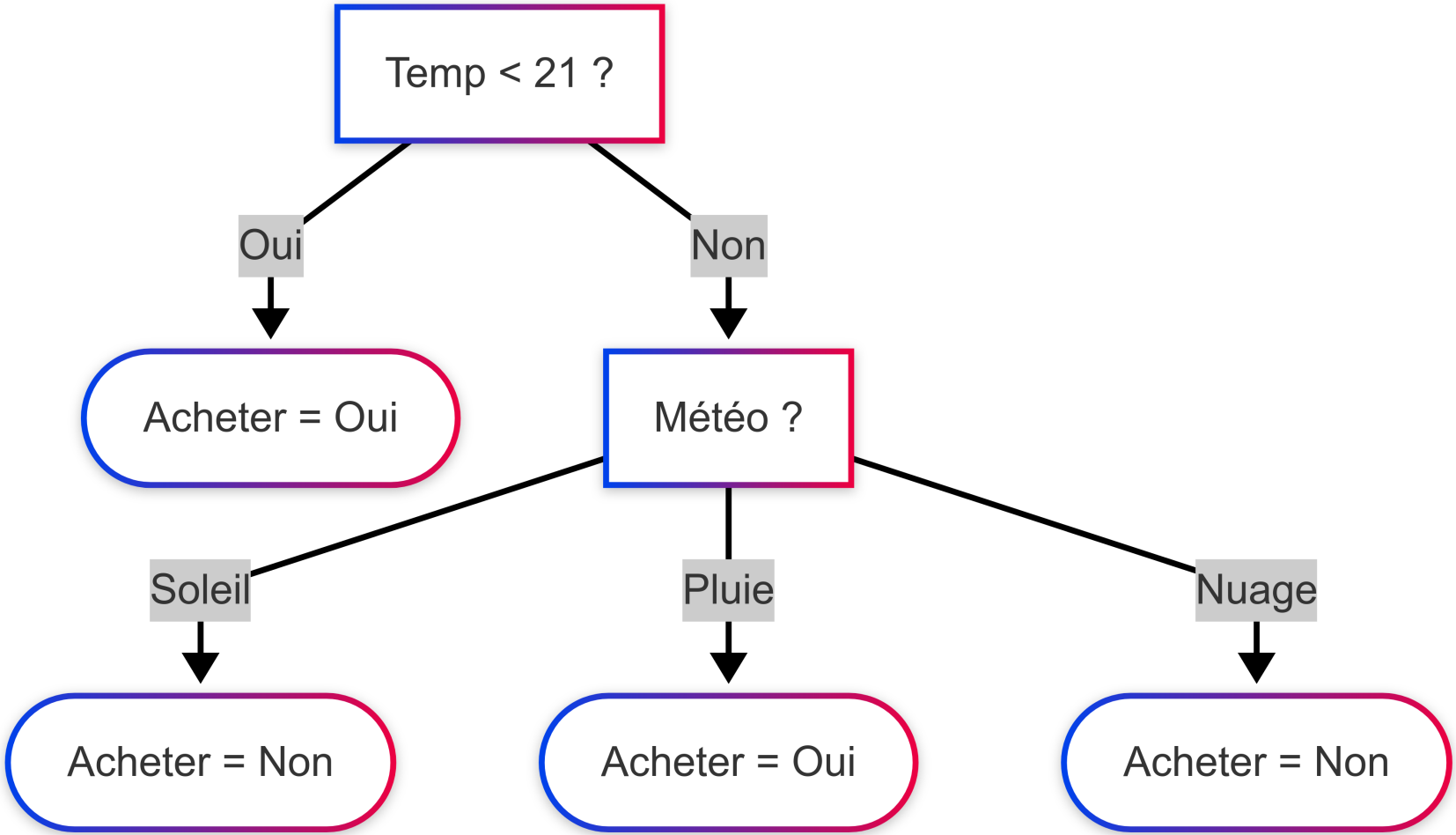
S_2 contient 4 exemples :

$(\text{Soleil}, 35, Non),\ (\text{Soleil}, 28, Non),\ (\text{Pluie}, 22, Oui),\ (\text{Nuage}, 25, Non).$

Test Météo :

- Soleil : 2 exemples → 2 Non ⇒ Gini=0, Feuille=«Non».
- Pluie : 1 exemple → 1 Oui ⇒ Gini=0, Feuille=«Oui».
- Nuage : 1 exemple → 1 Non ⇒ Gini=0, Feuille=«Non».

Arbre de Décision Final



Conclusion

- Les **arbres de décision** classent en testant successivement les attributs (Météo, Température, etc.).
- L'**Indice de Gini** mesure la pureté : on choisit à chaque nœud l'attribut qui **réduit** le plus l'impureté (maximisation du gain).
- Lorsque toutes les données d'une branche appartiennent à la même classe (Gini=0), on obtient une **feuille** et on arrête la division.

Bilan de cet exemple :

- La température est le premier attribut choisi (meilleur gain).
- Météo affine la séparation dans la branche "temp ≥ 21".

Exercice : Construire un Arbre de Décision

Objectif : Construire un arbre de décision basé sur un jeu de données simplifié en utilisant l'**indice de Gini** pour choisir les meilleurs attributs de séparation.

Données :

Temps	**Vent**	**Sortie Vélo ?**
Ensoleillé	Faible	Oui
Nuageux	Fort	Non
Pluie	Faible	Oui
Pluie	Fort	Non
Nuageux	Faible	Oui
Ensoleillé	Fort	Oui
Ensoleillé	Faible	Oui
Pluie	Fort	Non

Table – Données d'observation météo et décision de sortie en vélo

II : Apprentissage par Ensemble : Random Forest

Qu'est-ce qu'une Random Forest ? (1/2)

Définition : Une **Random Forest** est un ensemble de **plusieurs arbres de décision** construits à partir de :

- **Bootstrap** (Bagging) :
  - Création de plusieurs échantillons de taille égale à celle de l'ensemble initial, tirés **avec remise**.
  - Chaque échantillon peut ainsi contenir des doublons et ignorer certains exemples originaux.
- **Sélection aléatoire d'attributs** à chaque division (Random Subspace) :
  - Au lieu de tester tous les attributs, on en choisit **un sous-ensemble** aléatoire pour chaque nœud.
  - Cela augmente la **diversité** entre les arbres.

Qu'est-ce qu'une Random Forest ? (2/2)

Vote majoritaire ou moyenne

- **Classification** : la prédiction finale est le *vote majoritaire* de tous les arbres.
- **Régression** : la prédiction finale est la *moyenne* des prédictions.

Pourquoi utiliser une Random Forest ? (1/2)

Avantages :

- **Meilleure robustesse** : la variance du modèle est réduite par le vote/moyenne.
- **Réduction du risque de sur-apprentissage** (overfitting) par rapport à un arbre unique.
- **Facile à utiliser** : peu d'hyperparamètres critiques (nombre d'arbres, nombre d'attributs aléatoires, etc.).

Pourquoi utiliser une Random Forest ? (2/2)

Limites :

- **Moins interprétable** qu'un arbre unique (il est plus complexe de visualiser une "forêt").
- Peut être **coûteux en mémoire** et en temps de calcul pour un très grand nombre d'arbres.

Idée générale

Plus on a d'arbres *indépendants*, plus la **moyenne** de leurs erreurs se compense, améliorant la qualité globale.

Bagging : Bootstrap Aggregation (1/2)

Étapes clés pour construire une Random Forest (exemple de classification) :

1. **Échantillons Bootstrap** :
   - À partir de l'ensemble d'origine (taille N), on forme M sous-échantillons également de taille N, mais tirés **avec remise**.
   - Chaque sous-échantillon est utilisé pour entraîner **un arbre de décision**.
2. **Arbres aléatoires** :
   - À chaque nœud, au lieu de tester **tous les attributs**, on en sélectionne **un sous-ensemble** (exemple : $\sqrt{d}$ parmi d attributs).
   - On choisit l'attribut qui maximise la réduction du Gini **parmi ceux sélectionnés**.

Bagging : Bootstrap Aggregation (2/2)

3. **Vote majoritaire** :
   - Pour prédire une classe, on **combine les prédictions de chaque arbre** par un vote.
   - En régression, on prend la **moyenne** des valeurs prédites.

Note

Le *Bagging* seul réduit déjà la variance. La **sélection aléatoire d'attributs** *ajoutée* évite que tous les arbres se ressemblent trop (cas de Bagging pur).

Exemple : Mini-données (construction d'une Forêt) (1/2)

Données simplifiées (8 exemples) :

ID	Taille (m)	Poids (kg)	Jouer (Oui/Non)
1	1.50	60	Oui
2	1.80	80	Non
3	1.65	70	Oui
4	1.70	75	Non
5	1.55	62	Oui
6	1.90	90	Non
7	1.60	68	Oui
8	1.75	72	Non

Table – Jeu de données fictif

- **Attributs :** Taille, Poids.

Exemple : Mini-données (construction d'une Forêt) (2/2)

Étape 1 : Échantillons Bootstrap
On forme 3 échantillons (puisqu'on veut 3 arbres). Chaque échantillon est obtenu par **tirage avec remise** de 8 exemples :

$$\textbf{Échantillon \#1 :} \{1, 2, 2, 3, 5, 5, 7, 8\}$$
$$\textbf{Échantillon \#2 :} \{2, 4, 4, 5, 6, 6, 7, 8\}$$
$$\textbf{Échantillon \#3 :} \{1, 1, 2, 3, 6, 7, 8, 8\}$$

Remarque

Chaque échantillon est de taille 8 (même que l'ensemble initial), mais contient des **doublons** et éventuellement **omet** certains exemples (OOB – Out Of Bag).

Exemple : Construction des Arbres (Étape 2)

- Pour chaque échantillon, on construit un **arbre de décision** en utilisant la **sélection aléatoire d'attributs**.
- Si on a 2 attributs (Taille, Poids) :
  - À chaque nœud, on peut en tirer 1 au hasard (ou parfois les 2).
  - On choisit celui qui **maximise** la réduction de l'impureté (Gini).
- Ainsi, on obtient 3 arbres différents, chacun "surapprenant" potentiellement à sa portion de données, **mais** de façon distincte.

Conséquence

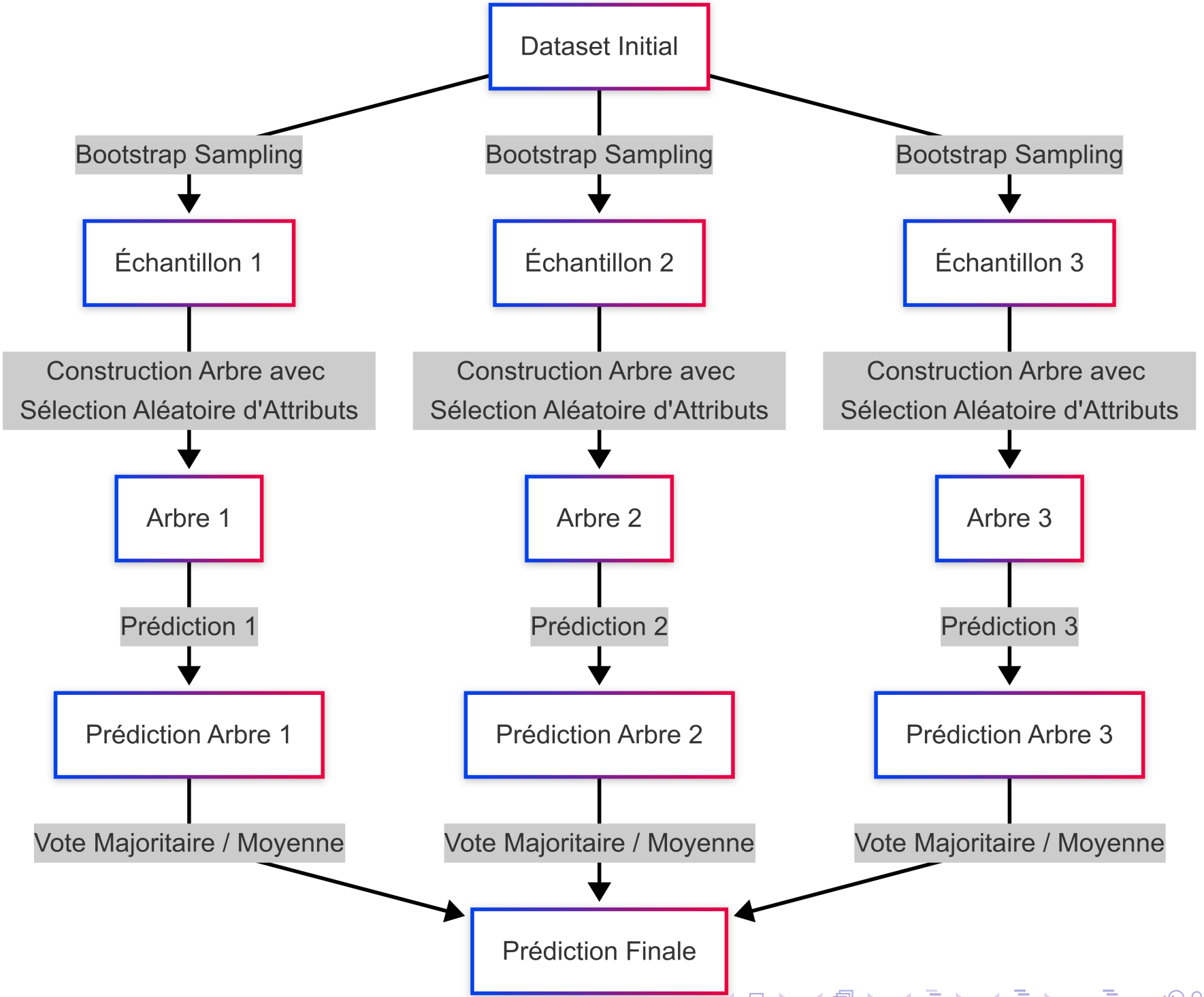
Les **corrélations** entre arbres diminuent (ils ne sont pas "clones"), améliorant la robustesse du vote final.

Exemple : Prédiction (Étape 3)

Pour un **nouvel exemple** ($Taille = 1.65, Poids = 72$) :

- **Arbre 1** prédit "Oui".
- **Arbre 2** prédit "Non".
- **Arbre 3** prédit "Oui".

Vote majoritaire = Oui (2votes sur 3)


Dataset Initial
Bootstrap Sampling
Bootstrap Sampling
Bootstrap Sampling
Échantillon 1
Échantillon 2
Échantillon 3
Construction Arbre avec Sélection Aléatoire d'Attributs
Construction Arbre avec Sélection Aléatoire d'Attributs
Construction Arbre avec Sélection Aléatoire d'Attributs
Arbre 1
Arbre 2
Arbre 3
Prédiction 1
Prédiction 2
Prédiction 3
Prédiction Arbre 1
Prédiction Arbre 2
Prédiction Arbre 3
Vote Majoritaire / Moyenne
Vote Majoritaire / Moyenne
Vote Majoritaire / Moyenne
Prédiction Finale

Conclusion sur les Random Forests

- Une **Random Forest** est une **forêt** d'arbres de décision entraînés sur des **échantillons bootstrap**, avec une **sélection aléatoire d'attributs**.
- Chaque arbre est "instable" mais le **vote** ou la **moyenne** confère une grande **stabilité** à la forêt.
- Réduction de l'overfitting, bonne performance pratique.

Exercice : Construire Trois Arbres d'une Forêt Aléatoire

Objectif : Comprendre le fonctionnement des forêts aléatoires en construisant **trois arbres** à partir de sous-échantillons tirés par Bootstrap.

Données initiales :

ID	Revenu mensuel	Historique de crédit	Décision
1	Faible	Mauvais	Refusée
2	Moyen	Bon	Approuvée
3	Élevé	Mauvais	Refusée
4	Faible	Bon	Approuvée
5	Élevé	Bon	Approuvée
6	Moyen	Mauvais	Refusée
7	Faible	Mauvais	Refusée
8	Élevé	Bon	Approuvée

Table – Données simplifiées pour une demande de prêt (8 exemples, 2 attributs).

Trois Échantillons Bootstrap

Échantillon 1 = {1, 2, 3, 4, 4, 7, 8, 8}

ID	Revenu mensuel	Historique de crédit	Décision
1	Faible	Mauvais	Refusée
2	Moyen	Bon	Approuvée
3	Élevé	Mauvais	Refusée
4	Faible	Bon	Approuvée
4	Faible	Bon	Approuvée
7	Faible	Mauvais	Refusée
8	Élevé	Bon	Approuvée
8	Élevé	Bon	Approuvée

Table – Échantillon 1 (tirage avec remise)

Échantillon 2 = {1, 2, 2, 5, 5, 5, 6, 8}

ID	Revenu mensuel	Historique de crédit	Décision
1	Faible	Mauvais	Refusée
2	Moyen	Bon	Approuvée
2	Moyen	Bon	Approuvée
5	Élevé	Bon	Approuvée
5	Élevé	Bon	Approuvée
5	Élevé	Bon	Approuvée
6	Moyen	Mauvais	Refusée
8	Élevé	Bon	Approuvée

Table – Échantillon 2 (tirage avec remise)

Échantillon 3 = {1, 3, 3, 4, 6, 6, 7, 8}

ID	Revenu mensuel	Historique de crédit	Décision
1	Faible	Mauvais	Refusée
3	Élevé	Mauvais	Refusée
3	Élevé	Mauvais	Refusée
4	Faible	Bon	Approuvée
6	Moyen	Mauvais	Refusée
6	Moyen	Mauvais	Refusée
7	Faible	Mauvais	Refusée
8	Élevé	Bon	Approuvée

Table – Échantillon 3 (tirage avec remise)

Étape 2 : Construction de 3 Arbres de Décision

1. **Indice de Gini initial** : Calculez le Gini de chacun des trois échantillons 1, 2, 3 (présentés auparavant).
2. **Sélection Aléatoire d'Attributs** :
   - À chaque nœud, choisissez **au hasard** l'un des deux attributs (*Revenu mensuel* ou *Historique de crédit*).
   - Calculez le **Gain de Gini** et effectuez la division si elle **réduit** l'impureté.
3. **Compléter les trois arbres** : Continuez les divisions jusqu'à obtenir des feuilles pures (classe "Approuvée" ou "Refusée") ou presque pures.

Rappel

La Random Forest utilise **Bagging** (tirages avec remise) et la **sélection aléatoire d'attributs** pour construire des arbres variés (réduisant le sur-apprentissage).

Étape 3 : Décision Majoritaire

Nouveau point à prédire :

(**Revenu mensuel** = Moyen, **Historique de crédit** = Bon)

1. **Arbre 1, Arbre 2, Arbre 3** : Déterminez la classe prédite **(Approuvée ou Refusée)** par chacun des trois arbres.
2. **Vote majoritaire** :

   Décision finale = *majorité*(votes "Approuvée", votes "Refusée").

3. Comparez la décision finale à ce que donnerait **un seul arbre** pris isolément.

Note

La Random Forest combine les prédictions pour **réduire l'instabilité** d'un arbre unique et améliorer la **robustesse** du modèle.