AU : 2024-2025

École Nationale d'Ingénieurs de Sousse

Rapport du projet : Random Forest

Réalisé par :
Khelifi Rafik
Abdesalem Iheb

Filière :
Informatique Appliquée
Groupe : IA.3.1

Enseignant : Mr. Walid Chainbi

Table des matières

Chapitre 1

Introduction

Dans le cadre du projet module d'intelligence artificielle, nous explorons les concepts fondamentaux et les algorithmes phares utilisés pour résoudre des problèmes complexes. L'intelligence artificielle, omniprésente dans de nombreux domaines, vise à concevoir des systèmes capables de traiter des données, de détecter des schémas complexes et de prendre des décisions intelligentes, imitant ou surpassant parfois les capacités humaines.

Parmi les nombreuses techniques développées en apprentissage supervisé, Random Forest se distingue par sa robustesse et sa polyvalence. Cet algorithme, proposé par Leo Breiman en 2001, repose sur la méthode d'ensemble et combine plusieurs arbres de décision pour produire des résultats généralisables. Random Forest, qui signifie "forêt aléatoire", est intuitif à comprendre, rapide à entraîner, et offre des modèles performants. Cependant, en dépit de son efficacité, il est parfois considéré comme une "boîte noire", en raison de la difficulté d'interpréter ses résultats de manière détaillée.

Chapitre 2

Fonctionnement de Random Forest

2.1 Introduction au fonctionnement

Random Forest est une méthode d'ensemble qui combine plusieurs arbres de décision pour obtenir des résultats robustes. Cet algorithme repose sur deux concepts fondamentaux : le Tree Bagging et le Feature Sampling, qui introduisent de la diversité dans la forêt et réduisent les risques d'overfitting.

2.2 Formule fondamentale

L'essence de Random Forest peut être résumée par la formule suivante :

$$\textbf{Random Forest} = \textbf{Tree Bagging} + \textbf{Feature Sampling}$$

Cette formule met en évidence les deux mécanismes clés :
— Tree Bagging (Bootstrap Aggregating) : Tirages aléatoires d'échantillons de données (lignes) avec remise pour entraîner chaque arbre.
— Feature Sampling : Sélection aléatoire d'un sous-ensemble de caractéristiques (colonnes) à chaque division de nœud dans chaque arbre.

Ces mécanismes permettent à Random Forest d'exploiter pleinement la puissance de la diversité pour produire un modèle collectif robuste et fiable.

2.3 Principe de base

— Bagging (Bootstrap Aggregating) : Création de sous-échantillons de données pour entraîner chaque arbre. Chaque sous-échantillon est obtenu par échantillonnage avec remise, ce qui introduit de la variabilité entre les arbres.
— Sélection aléatoire des caractéristiques : Lors de chaque division de nœud, seules quelques caractéristiques sont prises en compte pour déterminer la meilleure division, ce qui limite les corrélations entre les arbres.
— Combinaison des prédictions : Une fois les arbres construits, leurs prédictions sont agrégées :
  — En classification : les arbres votent, et la classe majoritaire est choisie.
  — En régression : les prédictions des arbres sont moyennées.

2.4 Étapes principales

Le fonctionnement détaillé de Random Forest peut être décomposé en trois étapes clés :

1. Génération aléatoire de sous-échantillons des données À partir du jeu de données d'origine, plusieurs sous-échantillons sont créés par échantillonnage avec remise. Chaque sous-échantillon est utilisé pour entraîner un arbre de décision distinct. Cette étape garantit que chaque arbre a une vision partielle et unique des données.
2. Construction d'un arbre de décision pour chaque sous-échantillon Chaque arbre est construit à partir d'un sous-échantillon. Lors de la construction, à chaque nœud, un sous-ensemble aléatoire de caractéristiques est sélectionné pour déterminer la meilleure division. Cela introduit davantage de diversité dans la forêt.
3. Agrégation des prédictions des arbres Une fois tous les arbres construits, leurs prédictions sont combinées :
   — En classification : les arbres votent, et la classe majoritaire est retenue.
   — En régression : les prédictions des arbres sont moyennées.

2.5 Genèse de l'algorithme

L'arbre de décision, bien qu'intuitif et interprétable, souffre d'un problème majeur : sa performance dépend fortement de l'échantillon de départ, ce qui le rend sujet à l'overfitting. Pour pallier ce problème, Leo Breiman a introduit Random Forest en 2001. En combinant plusieurs arbres indépendants grâce à des tirages aléatoires sur les données et les caractéristiques, Random Forest réduit significativement le risque d'overfitting tout en maintenant une grande flexibilité.

2.6 Résumé

En résumé, Random Forest est une méthode d'ensemble puissante qui utilise :
— Tree Bagging : Pour réduire la variance.
— Feature Sampling : Pour introduire de la diversité entre les arbres.
— Agrégation : Pour obtenir une prédiction robuste et généralisable.
Cette approche, bien que simple dans son concept, permet de construire des modèles à la fois précis, robustes et adaptables à divers types de données.

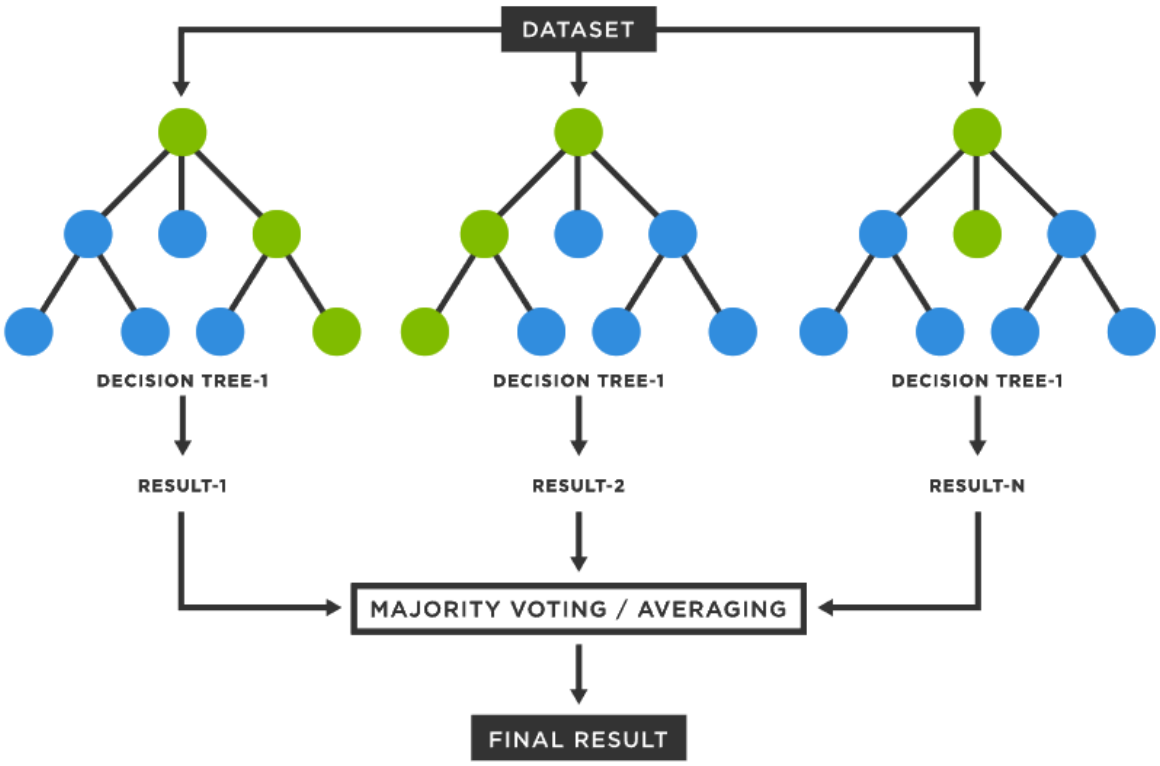

FIGURE 2.1 – Principe de Random Forest

Chapitre 3

Comparaison avec d'autres algorithmes

3.1 Arbre de décision

— Principe : L'arbre de décision est une structure hiérarchique où chaque nœud représente une décision basée sur une caractéristique, conduisant à une prédiction dans les feuilles.
— Avantages :
  — Facile à comprendre et à interpréter, même pour des utilisateurs non experts.
  — Rapide à entraîner sur des ensembles de données de petite taille.
— Limites :
  — Très sujet à l'overfitting, surtout avec des jeux de données petits ou bruités.
  — Sensible à des variations mineures dans les données d'entrée, ce qui peut changer la structure de l'arbre.

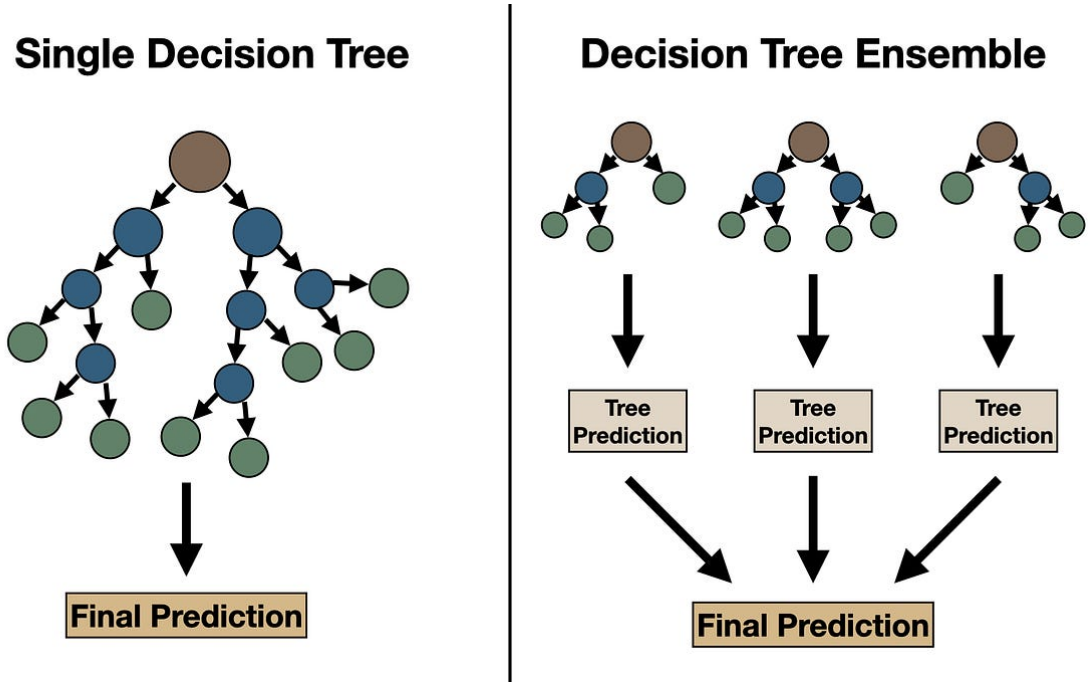

FIGURE 3.1 – Arbre de décision vs Random Forest

3.2 Bagging (Bootstrap Aggregating)

- Principe : Combinaison de plusieurs modèles indépendants entraînés sur des sous-échantillons aléatoires des données d'origine (échantillonnage avec remise).
- Avantages :
  - Réduit la variance en agrégeant les prédictions des modèles.
  - Permet une meilleure généralisation que l'utilisation d'un seul modèle.
- Exemple : Random Forest est une extension du bagging, améliorée par la sélection aléatoire des caractéristiques, augmentant la diversité des modèles.
- Limites :
  - Peut être coûteux en termes de calcul pour des jeux de données très volumineux, car il nécessite la construction de plusieurs modèles.

3.3 Boosting

- **Principe :** Construire des modèles de manière séquentielle, où chaque modèle corrige les erreurs des précédents.
- **Avantages :**
  - Réduit le biais en se concentrant sur les erreurs difficiles à corriger.
  - Donne de très bons résultats sur des données complexes.
- **Exemples :**
  - **AdaBoost :** Met l'accent sur les exemples mal classés en leur attribuant un poids plus élevé.
  - **Gradient Boosting :** Optimise une fonction de perte grâce à une approche de descente de gradient.
- **Limites :**
  - Plus sensible au surapprentissage (overfitting) que Random Forest.
  - Plus lent à entraîner car les modèles sont construits séquentiellement.

3.4 Pourquoi Random Forest ?

- Équilibre biais-variance : Réduit efficacement la variance (grâce au bagging) tout en maintenant un biais acceptable.
- Performance : Comparable à Boosting dans de nombreux cas mais avec un entraînement plus rapide et une implémentation plus simple.
- Robustesse : Moins sensible au bruit et à l'overfitting que les arbres de décision simples.
- Flexibilité :
  - Applicable à des problèmes de classification et de régression.
  - Capable de gérer des ensembles de données avec des valeurs manquantes ou des caractéristiques non pertinentes.
- Limites par rapport à Boosting :
  - Moins performant que Boosting dans des contextes où l'erreur est principalement due au biais.
  - Peut nécessiter un grand nombre d'arbres pour atteindre de très hautes performances, augmentant ainsi le coût en calcul.

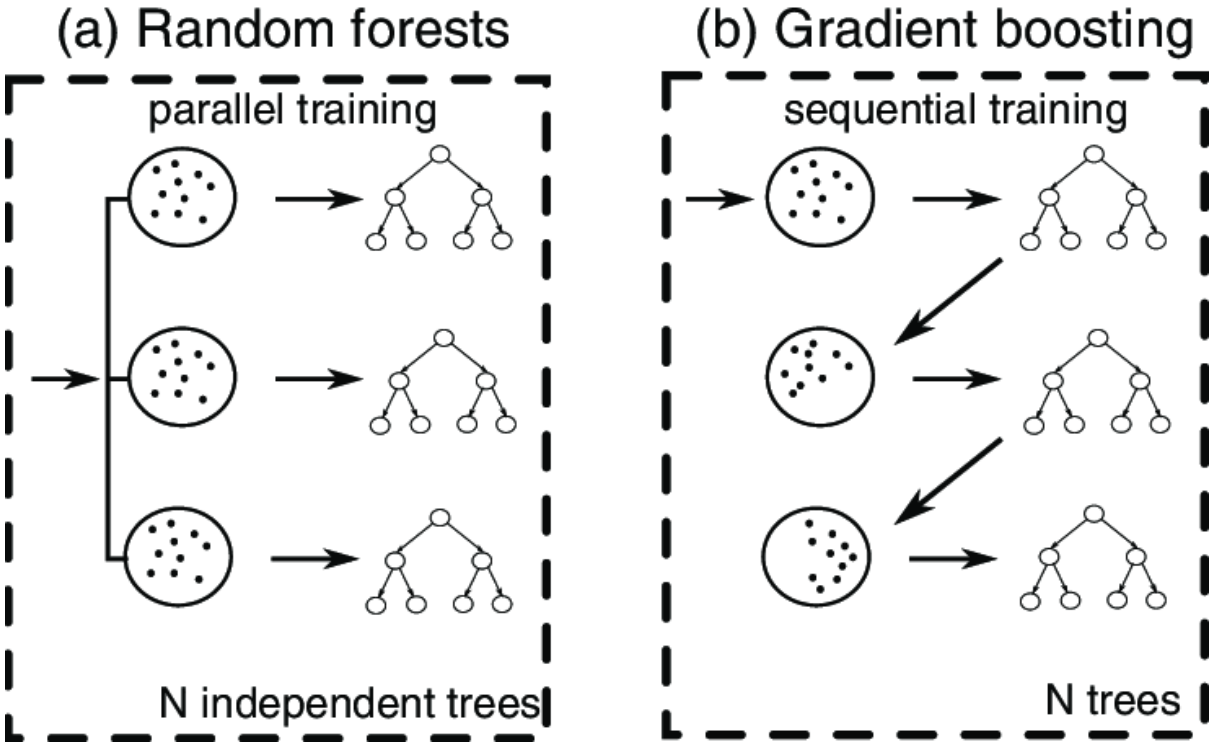

FIGURE 3.2 – Gradient Boosting vs Random Forest

Chapitre 4

Avantages et Désavantages

4.1 Avantages

Random Forest présente plusieurs avantages qui en font un algorithme très populaire en apprentissage supervisé :

- Capacité à gérer les valeurs manquantes : Les arbres de la forêt peuvent traiter des échantillons incomplets en utilisant des approches comme des règles basées sur la fréquence des caractéristiques.
- Réduction du risque d'overfitting : Grâce à l'utilisation du bagging et à la sélection aléatoire des caractéristiques, Random Forest diminue la variance des prédictions sans compromettre significativement le biais.
- Robustesse face au bruit dans les données : Random Forest reste performant même en présence de données bruitées ou de caractéristiques non pertinentes.
- Polyvalence : Cet algorithme est adapté à la fois pour des problèmes de classification (exemple : prédiction de classes) et de régression (exemple : prévision de valeurs continues).
- Explicabilité : Bien que le modèle complet soit complexe, chaque arbre individuel peut être analysé pour comprendre la logique derrière une prédiction spécifique.
- Bonne performance avec les grands ensembles de données : Random Forest peut gérer des bases de données volumineuses et de haute dimensionnalité, ce qui le rend utile dans des applications réelles variées.
- Capacité à mesurer l'importance des caractéristiques : Grâce à des métriques comme la réduction de l'impureté ou la permutation, Random Forest identifie quelles caractéristiques influencent le plus les prédictions.

4.2 Désavantages

Malgré ses nombreux atouts, Random Forest présente certaines limites :

- Complexité calculatoire : En raison de la construction de plusieurs arbres de décision et de la combinaison des prédictions, Random Forest peut être coûteux en temps et en ressources pour des ensembles de données très volumineux.
- Moins interprétable qu'un seul arbre de décision : Bien que chaque arbre individuel soit explicable, la forêt dans son ensemble est plus difficile à interpréter, notamment pour des utilisateurs non techniques.
- Sensibilité à l'hyperparamétrage : Si des hyperparamètres comme le nombre d'arbres

ou la profondeur maximale ne sont pas bien configurés, le modèle peut soit sous-apprendre, soit sur-apprendre (overfitting).
— Mémoire élevée : La construction et l'utilisation de multiples arbres nécessitent une mémoire importante, ce qui peut poser des problèmes pour des systèmes aux ressources limitées.
— Performances limitées sur des données biaisées : Random Forest peut souffrir lorsque les données d'apprentissage sont fortement déséquilibrées, par exemple, avec des classes majoritaires et minoritaires.
— Problèmes de temps d'inférence : Lorsqu'un grand nombre d'arbres est utilisé, le temps de prédiction pour chaque nouvel échantillon peut être relativement long.

Chapitre 5

Conclusion et Perspectives

5.1 Résumé

Random Forest s'est imposé comme une méthode d'ensemble puissante et flexible, particulièrement efficace pour résoudre des problèmes complexes en classification et en régression. En surmontant les limites des arbres de décision simples grâce au bagging et à la sélection aléatoire des caractéristiques, il offre des prédictions robustes, tout en réduisant le surapprentissage. Ses capacités à gérer les données bruitées, les valeurs manquantes, et à mesurer l'importance des caractéristiques en font un choix privilégié dans de nombreuses applications pratiques. Cependant, son coût en calcul et sa relative complexité en termes d'interprétabilité restent des défis à considérer.

5.2 Perspectives

Afin de tirer parti des forces de Random Forest et d'en élargir les applications, plusieurs pistes d'amélioration et d'exploration méritent d'être envisagées :

- Amélioration des performances calculatoires :
  - Intégrer des techniques d'optimisation comme la réduction des caractéristiques redondantes pour diminuer les temps d'entraînement.
  - Utiliser des approches parallélisées ou des outils comme Hadoop ou Spark pour traiter de grands ensembles de données.
- Applications à grande échelle :
  - Tester l'algorithme sur des jeux de données volumineux (**Big Data**), en exploitant les technologies distribuées pour un traitement efficace.
  - Développer des modèles Random Forest adaptés à des domaines émergents comme la médecine personnalisée, la détection de fraude, ou encore l'intelligence artificielle embarquée.

En conclusion, Random Forest reste un outil incontournable en science des données et apprentissage automatique, offrant un excellent compromis entre robustesse et précision. Les recherches futures devraient se concentrer sur l'amélioration de ses performances computationnelles et son intégration avec des systèmes intelligents de nouvelle génération.