

DÉPARTEMENT DE MATHÉMATIQUES, D'INFORMATIQUE ET DE GÉNIE

**8INF29 – Séminaire en informatique I**

**ING76215 – Lectures dirigées**

**Automne 2021**

**Projet 1**

**Professeur : Ismaïl Khriss**

**Forets aléatoires**

**Par : Ayman Chafni**

Table des matières

[Introduction : 3](#_Toc88687949)

[Arbres de décision 3](#_Toc88687950)

[Définition : 3](#_Toc88687951)

[Concepts : 4](#_Toc88687952)

[Limitations**:** 5](#_Toc88687953)

[Concept : 6](#_Toc88687954)

[Un échantillonnage aléatoire : 6](#_Toc88687955)

[Un sous-ensemble aléatoire d’attributs : 6](#_Toc88687956)

[Code : 7](#_Toc88687957)

[Librairies : 7](#_Toc88687958)

[Préparation : 8](#_Toc88687959)

[Entrainement et prédiction : 8](#_Toc88687960)

[Références : 10](#_Toc88687961)

# Introduction :

« Francis Galton, un statisticien, voulait initialement prouver qu’un groupe du « bas peuple », même en grand effectif, ne pouvait même pas réaliser une tâche aussi simple qu'estimer le poids d'un bœuf dans une foire agricole. Au cas par cas, ce fut effectivement mauvais, mais en faisant la moyenne des estimations, on atteint un résultat qui est qualifié d'excellent. » [1]

Ce phénomène est connu sous le nom de la sagesse de la foule. En analogie avec les forêts aléatoires, elles sont basées ainsi sur le même phénomène où les individus du « bas du peuple » ici sont les arbres de décision qui ne sont pas très intelligentes en elles-mêmes (elles peuvent être très utiles parfois mais parfois trouvent de la difficulté à s’adapter comme on va voir à la partie II) mais en combinant plusieurs on obtient de meilleurs résultats. Encore, plus notre population est diverse, plus on garantit moins d’erreur sur la prédiction, « Cela s'explique par la formule suivante : **erreur du groupe = erreur individuelle moyenne - diversité des prédictions**»[1] C’est pour cette raison que les arbres de décision dans les forêts aléatoires sont construits d’une certaine manière à maximiser la diversité entre eux (ce qui va être détaillé dans la partie III)

Pour définir alors les forêts aléatoires, on peut dire que c’est une technique d’apprentissage d’ensemble qui combinent plusieurs arbres de décision pour avoir une meilleure performance prédictive. Les forêts aléatoires comme les arbres de décision sont des techniques supervisées de Machine Learning et peuvent être utilisées pour la classification de même que la régression.

# Arbres de décision

## Définition :

La technique des arbres de décision est l’une des techniques de Machine Learning les plus populaires et en même temps facile à utiliser. Elle peut être utilisée en classification ou en régression. Comme le nom le suggère les arbres de décision consistent en des nœuds et des branches. On dérive de chaque nœud plusieurs sous-nœuds, à partir de quelques règles simples sur les attributs, et on itère jusqu’à la satisfaction d’une certaine condition garantissant que le nœud résultant est « pur ». En obtenant plusieurs nœuds « purs » à la fin, on va pouvoir prédire facilement la classe d’un nouvel élément en suivant juste les règles établies jusqu’à l’atteinte de l’un de ces nœuds.

## Concepts :

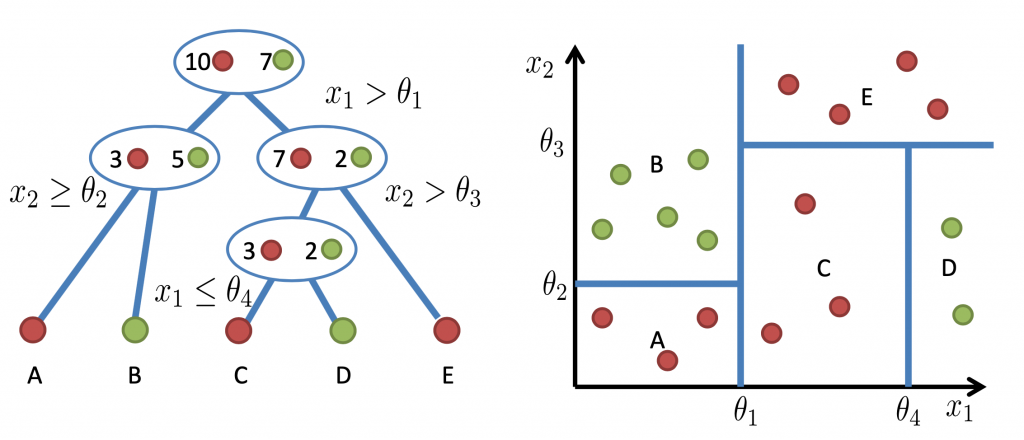


Figure : Exemple d'arbre de décision

Le principe est très simple car c’est très intuitif et ressemble beaucoup au raisonnement humain. Comme cité dans la définition et comme le montre la figure 1, l’algorithme des arbres de décision extrait des sous nœuds à partir d’un nœud en appliquant des règles simples sur les attributs. On itère jusqu’à ce qu’on arrive à un nœud « pure » (qui vérifie une certaine condition)

\*\*L’algorithme des arbres de décision dans plusieurs librairies populaires de Machine Learning comme sci-kit learn est conçu comme étant un arbre binaire, c.à.d., chaque nœud se divise au plus en deux sous-nœuds, et ce pour des raisons de performance.

Dans chaque étape, on doit diviser le nœud a la base d’un attribut, mais cet attribut n’est pas choisi arbitrairement (sinon on aura de très mal résultats). Il y a une certaine approche statistique que les arbres de décision suivent pour déterminer l’attribut dans chaque nœud. Pour cela il y a des différents algorithmes qui peuvent être utilisées :

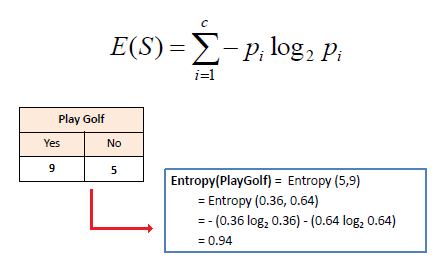
* **ID3** → (extension de D3)
* **C4.5** → (successeur de ID3)
* **CART** → (Classification et Régression)
* **CHAID** → (Chi-carrée)
* **MARS** → (multivariate adaptive regression splines)

On peut voir clairement que l’utilisation d’un algorithme ou l’autre pour la sélection d’un attribut dépend du type de notre apprentissage (classification ou régression), ainsi que nos préférences et la nature de nos données.

Tous ces algorithmes utilisent des critères statistiques pour effectuer la sélection, ces critères sont :

* **Entropie**,
* **Gain d’information,**
* **Indice de Gini,**
* **Rapport de gain,**
* **Reduction de la Variance,**
* **Chi-Carrée**

Chaque algorithme parmi les algorithmes déjà cités utilisent une ou plusieurs de ces critères. Par exemple : l’algorithme ID3 utilisent l’entropie ainsi que le gain d’information pour prendre sa décision. Ces derniers peuvent se calculer de la manière suivante :



Figure

Une image contenant texte

Description générée automatiquement

Figure

On ne va pas détailler le principe de fonctionnement de chaque critère et chaque algorithme utilisé par les arbres de décision parce que ce n’est pas le but ici.

## Limitations**:**

Comme indiqué dans l’introduction, les arbres de décision ne présentent pas une technologie très solide bien que l’on utilise encore elle présente plusieurs limitations, notamment le problème du surapprentissage. Pour régler ce problème, on peut utiliser des techniques comme l’élagage où on élimine des branches de l’arbre de telle façon à ne pas influencer la précision de la prédiction. Ou encore mieux, on peut utiliser une autre technique dérivée des arbres de décision qui est les forets aléatoires.

# Concept :

La technique des forêts aléatoires est une technique d’apprentissage d’ensemble qui, comme les arbres de décision, s’applique en cas de classification ainsi que la régression. Dans le cas d’une classification, la forêt aléatoire rassemble toutes les prédictions des arbres qu’elle a utilisés et la classe avec le plus grand nombre de « votes » serait la classe prédite par la forêt. Dans l’autre cas de régression, la foret aléatoire calcul la moyenne des prédictions de tous les arbres.

La clé à une prédiction plus précise dans les forêts aléatoires est d’avoir une diversité de résultats des différents arbres de décision, ce qui implique une faible corrélation entre ces arbres.[2]

Le mot « aléatoires » dans le nom vient du fait que les forets aléatoires utilisent :

* Un échantillonnage aléatoire des données d’entrainement pour chaque arbre de décision
* Un sous-ensemble aléatoire d’attributs utilisés pour chaque arbre de décision

Ces deux méthodes sont utilisées pour assurer la faible corrélation entre les arbres de décision.

## Un échantillonnage aléatoire :

Les forets aléatoires n’utilisent pas toutes les données d’apprentissage pour chaque arbre, mais plutôt chaque arbre prend en entrée un échantillon aléatoire de ces données. Cela ne veut pas dire que la taille des données utilisées pour chaque arbre est inferieure a la taille des données originale car on remplace les données enlevées dans l’échantillon par des duplications des valeurs présentes. Par exemple, si nos données d’entrainement sont [1, 2, 3, 4, 5, 6] l’échantillon va être quelque chose comme [1, 2, 3, 4, 5, 6]. Cette technique est connue sous le nom du ‘bagging’ (Bootstrapping Aggregation) et ça permet d’avoir des arbres moins corrélés les uns les autres. En fait, les forets aléatoires profitent du fait que les arbres aléatoires sont très sensibles aux données d’entrainement, et pour cela une simple modification dans les données d’entrée peut générer des résultats très différents.

## Un sous-ensemble aléatoire d’attributs :

Pour chaque arbre dans notre foret, on utilise juste un sous-ensemble d’attributs choisi aléatoirement et pas tous les attributs. Ceci permet ainsi d’agrandir le gap entre nos arbres, c.à.d., moins de corrélation et par la suite une meilleure capacite prédictive.

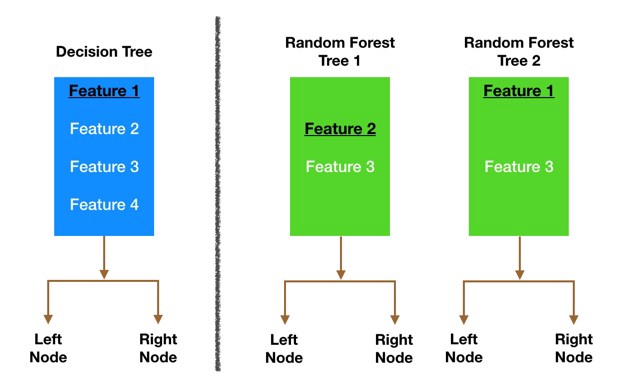


Figure  : Différence entre arbre de décision et foret aléatoires dans l’utilisation des attributs

S’assurer que les différents arbres ne sont pas corrélés et sont différents les uns les autres.

# Code :

J’ai essayé de faire une petite démonstration des forêts aléatoires à l’aide de la librairie de Machine Learning sur Python SciKit-Learn.

## Librairies :

Les librairies utilisées pour cette démonstration sont :

* Pandas : C’est une librairie très utile pour la manipulation et l’analyse des données
* Scikit-learn (package sklearn) : Cette librairie comporte plusieurs utilités de Machine learning. On utilise de ces utilités la classe RandomForestClassifier du package sklearn.ensemble, la classe qui permet d’utiliser la technique des forêts aléatoires, la classe metrics qui offre plusieurs outils pour évaluer la performance du modèle.
* Subprocess : Cette librairie permet de lancer des commandes du système dans notre programme
* La classe Image de Ipython.display qui va nous permettre d’afficher des images dans un notebook Jupyter.

Une image contenant texte

Description générée automatiquement

Figure

## Préparation :

Le dataset utilisé est un dataset importé du site web Kaggle.com et présente des enregistrements sur des patients (Age, sexe, niveau de cholestérol, etc.) et est-ce qu’ils présentent des maladies cardiaques ou pas.

Une image contenant table

Description générée automatiquement

Figure

On commence alors par la lecture de ces données à partir d’un fichier csv à l’aide de la fonction read\_csv() de pandas. Deuxièmement, on effectue un petit traitement de données en éliminant toute ligne contenant une valeur manquante avec la fonction dropna() de pandas.

Avant d’utiliser l’algorithme des forets aléatoires de sklearn, il faut que toutes les valeurs dans notre dataset soient des réels, on remplace alors les chaines de caractères par des nombres réels comme ‘M’ pour masculin devient 1 et ‘F’ devient 0.

On procédera après a la division du dataset en données d’entrainement et des données de test qui constitueront dans notre cas 20% du total. On sépare après la colonne des classes des autres colonnes.

Une image contenant texte

Description générée automatiquement

Figure

## Entrainement et prédiction :

Après la préparation des données, on initialise la classe RandomForestClassifier avec nos préférences. Ici, j’ai laisse les valeurs de paramètres par défaut, sauf pour le max\_depth qui est la profondeur max de chaque arbre fixé à 2 pour une bonne visualisation, ainsi que le random\_state=0 pour pouvoir contrôler l’état aléatoire et pouvoir régénérer les résultats.

Ensuite on applique la fonctions fit() qui permettra de faire l’entrainement. On prédit les classes des données test avec la fonction predict() et on utilise la fonction metrics.accuracy\_score() pour déterminer la précision de notre prédiction.

Une image contenant texte

Description générée automatiquement

Figure

Pour visualiser les résultats, j’ai choisi de visualiser 4 arbres de décision de notre foret aux positions : 0,10,90 et 99. La fonction export\_graphviz nous permet de faire cela en générant des fichiers .dot qu’il faut convertir en png, et on fait cela en lançant une commande système a l’aide de la fonction call(). Mais il faut faire attention et installer Graphviz et les utilités (voir notes.txt)

Une image contenant texte

Description générée automatiquement

Figure

En manipulant un peu les paramètres du constructeur de RandomForestClassifier, sous forme de petits tests en changeant a chaque fois un paramètre pour voir la combinaison qui permet une meilleure performance et moins de temps d’exécution. Par exemple, j’ai refait l’entrainement en remplaçant le critère de Gini par l’entropie, et en ne limitant pas la profondeur de l’arbre comme dans la première exécution et j’ai eu de meilleurs résultats comme le montre la figure 11.

Une image contenant texte

Description générée automatiquement

Figure

Une image contenant texte

Description générée automatiquement

Figure

Résultats :

Pour les résultats complets, veuillez exécuter le fichier python ou le fichier Jupyter joints à ce rapport. Des fichiers png seront produits dans le même chemin représentant des arbres de décision.

# 

# Références :

[1] <https://fr.wikipedia.org/wiki/Sagesse_de_la_foule>

[2] <https://towardsdatascience.com/understanding-random-forest-58381e0602d2>

[3] <https://www.kdnuggets.com/2020/01/decision-tree-algorithm-explained.html>

[4] <https://www.kaggle.com/fedesoriano/heart-failure-prediction>

[5] <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html>

[6] <https://towardsdatascience.com/how-to-visualize-a-decision-tree-from-a-random-forest-in-python-using-scikit-learn-38ad2d75f21c>