UC 2A : Big data et sciences du vivant

Apprentissage supervisé par inférence d'**arbres de décisions**

Ce TP a pour objectif de vous faire expérimenter l'apprentissage par arbres de décisions en faisant varier les paramètres.

Les arbres de décision peuvent s'accommoder aussi bien de variables catégorielles que numériques.

Les arbres de décision ont plusieurs **avantages** qui les rendent intéressants dans des contextes où il est utile de comprendre la séquence de décisions prise par le modèle :

- Ils sont simples à comprendre et à visualiser.
- Ils nécessitent peu de préparation des données (normalisation, etc.).
- Le coût d'utilisation des arbres est logarithmique.
- Ils sont capables d'utiliser des données catégorielles et numériques.
- Ils sont capables de traiter des problèmes multi-classe.
- Modèle en boîte blanche : le résultat est facile à conceptualiser et à visualiser.

Ces modèles présentent néanmoins des risques :

- *Sur-apprentissage* : parfois les arbres générés sont trop complexes et généralisent mal à des données non vues. Choisir des bonnes valeurs pour les paramètres *profondeur maximale* (max_depth) et *nombre minimal d'exemples par feuille* (min_samples_leaf) permet d'éviter ce problème.
- Il faut éviter d'apprendre avec des classes très déséquilibrées. Il est donc recommandé d'ajuster la base de données avant la construction, pour éviter qu'une classe domine largement les autres (en termes de nombre d'exemples d'apprentissage).

Vous trouverez sur le site du cours l'archive '**tp_arbre_decision.zip**' contenant tous les fichiers nécessaires pour les expériences.

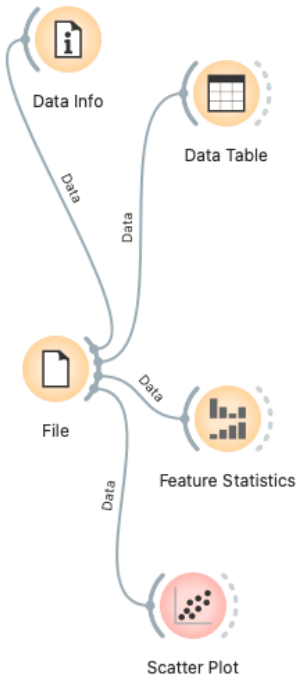
L'apprentissage se fera en utilisant **le logiciel « Orange »** qui est une boîte à outil très riche pour les méthodes de sciences des données, avec programmation graphique, ce qui la rend facile à utiliser.

Le logiciel est accessible soit par Anaconda-Navigator, soit directement en le chargeant préalablement.

Vous utiliserez en particulier les widgets '*Tree*', 'C4.5' pour l'apprentissage, ainsi que les widgets '*Classification Tree Viewer*' et '*Test and Score*' pour la visualisation des résultats.

1- Apprentissage d'arbres de décision avec le logiciel Orange

Question 1 : Vous allez commencer par réaliser le schéma d'analyse suivant :



Question 2 : En double cliquant sur le widget « file », chargez le fichier « iris.tab ».

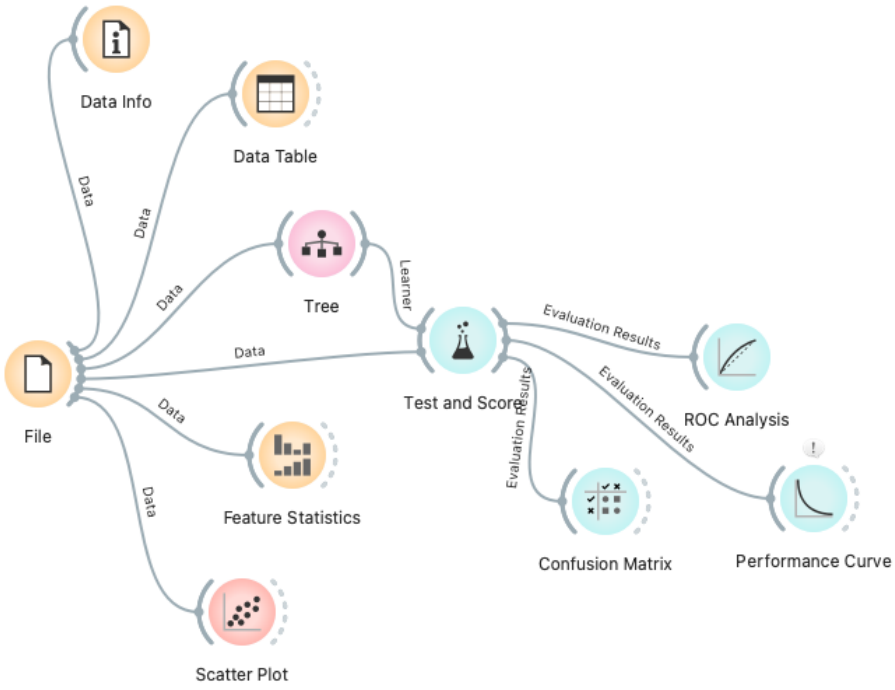
- Double-cliquez sur le widget « Data Info ». Quelles informations avez-vous sur la base de données ?
- Double-cliquez sur le widget « Data Table ». Regardez comment sont décrites les données. Quel est le type des descripteurs ?
- Double-cliquez sur le widget « Feature Statistics ». Observez la distribution des classes pour chacun des 4 descripteurs. Lequel vous paraît le plus discriminant ?
- Double-cliquez sur le widget « Scatter Plot ». Essayez plusieurs couples d'attributs. Quels sont ceux qui apparaissent les plus discriminants pour prédire la classe des iris ?

Question 3 : Nous allons maintenant étudier un fichier de données très célèbre : le fichier « iris.tab » qui compte 150 descriptions de fleurs de type « iris ». Les descriptions se font ici selon 4 descripteurs : la *longueur* et la *largeur* des *sépales*, la *longueur* et la *largeur* des *pétales*. Ces iris appartiennent à trois sous-classes : « iris-setosa », « iris-versicolor » et « iris-verginica ».

Le problème est ici d'essayer de trouver comment prédire la classe des iris en fonction des 4 descripteurs mesurables. Dans le cas présent, on cherche une fonction de prédiction prenant la forme d'un arbre de décision.

Complétez le schéma d'analyse selon la figure de la page suivante.

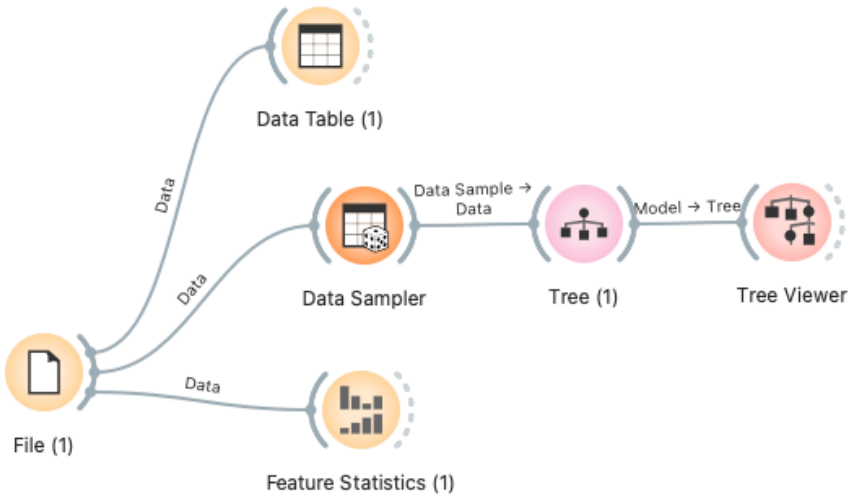
Ici, vous allez contrôler l'apprentissage dans le widget « Test and Score ». Dans un premier temps, nous allons réaliser un apprentissage par « Random sampling » avec un training size de 70%, le test s'opérant sur les 30% de données restantes. Les résultats affichés correspondent à la moyenne des résultats sur 10 répétitions. Ce nombre peut être changé.



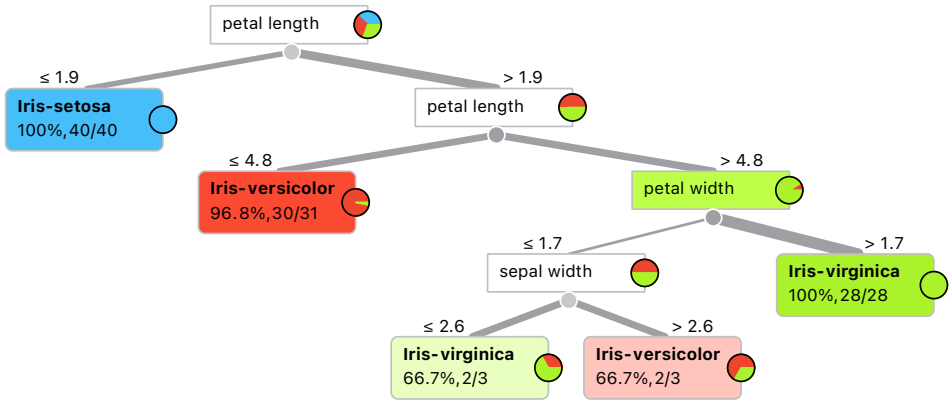
Il est intéressant d'examiner la matrice de confusion. Que constatez-vous ?

Question 4 : Nous allons maintenant étudier l'impact du nombre d'exemples d'apprentissage.

Il nous faut d'abord être capable d'afficher un arbre appris. Pour cela, nous allons réaliser le schéma suivant.



Le widget « Data Sampler » permet de sélectionner au hasard une proportion des exemples de la base de données. Par exemple 70%. Une fois l'apprentissage réalisé, le widget « Tree Viewer » permet de visualiser l'arbre de décisions appris. Vous devriez obtenir quelque chose comme la figure suivante.



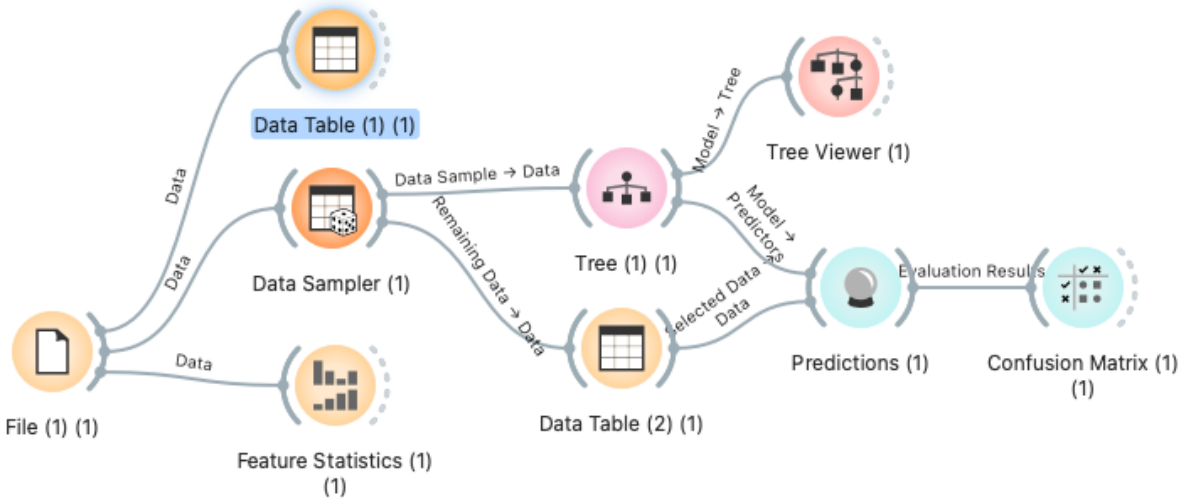
Est-ce que l'arbre appris par l'algorithme correspond à vos attentes ? Retrouvez-vous l'attribut qui vous semblait le plus discriminant en haut de l'arbre ?

Comment interprétez-vous la matrice de confusion ?

Et les courbes ROC pour chacune des trois classes d'iris ?

Vous pouvez examiner les exemples mal classés en double-cliquant sur le widget « Data Table(1) » à placer après « Confusion Matrix » et en sélectionnant les mal classés ('mis-classified'). Est-ce que ce sont des exemples difficiles à classer ?

Question 5 : Nous voulons maintenant voir quelle est la **performance en généralisation**. Nous allons pour cela examiner la performance sur les 30% de données non sélectionnées pour l'apprentissage. Cela se fait en prenant une deuxième sortie du « Data Sampler ». Attention, il faut bien prendre « Remaining data ». Voir le schéma suivant.



Quelles sont les performances obtenues ?

Que se passe-t-il si on demande au « Data Sampler » de sélectionner **seulement 5% des données** pour l'apprentissage ? Quel arbre obtenez-vous ? Quelle est maintenant la performance ?

Étude d'une base de données plus grosse : heart.csv

Question 6 :

Chargez la base 'heart.csv'.

Combien a-t-elle de lignes ? Combien de colonnes ?

Essayez un apprentissage par arbre de décisions dessus ? Que se passe-t-il ?

Que dit le message d'erreur ?

Faites ce qu'il faut pour pouvoir réaliser l'apprentissage par arbre de décision.

Visualisez l'arbre appris sur toutes les données.

Quelle est la performance observée pour une validation croisée à 5 plis ?

Question 7 : Contrôle du sur-apprentissage.

Jouez sur les paramètres 'Min.number of instances in leaves' et 'maximal tree depth' pour essayer d'obtenir une performance optimale.

2- Apprentissage d'arbres de régression

Nous allons maintenant étudier la tâche de régression (prédiction d'une variable continue en fonction de variables de description continues). Pour cela, on utilisera les widgets 'Regression Tree' et 'Regression Tree Graph' de l'onglet 'Regression'.

Afin de faciliter l'interprétation des résultats, on examinera des tâches de régression à 2 variables.

Question 8 : Utilisez les fichiers de données déjà créés : 'bd_2d_regression_droite.tab' et 'bd_2d_regression_courbe.tab', et réalisez un apprentissage avec arbre de régression. Examinez le résultat obtenu grâce au widget 'Regression Tree Graph'. Modifiez les réglages par défaut de 'Regression Tree'. Qu'en conclure sur l'interprétabilité du résultat ?

Question 9 : Vous utiliserez le widget 'Widget Generate Data' pour créer vos propres bases de données à deux variables. Il faudra ensuite utiliser le widget 'select Attributes' pour ramener le fichier à un fichier avec deux attributs continus, dont l'un est la variable à prédire.

3- Apprentissage de forêts aléatoires (*random forests*)

Dans cette partie, nous examinons l'apprentissage par méthode d'ensemble, c'est-à-dire en combinant plusieurs apprentissages.

Question 10 : refaites des expériences de **classification** avec les bases de données que vous avez créées dans la première partie en utilisant cette fois la classification par 'random forest'.

Question 11 : Faites de même pour les arbres de régression.