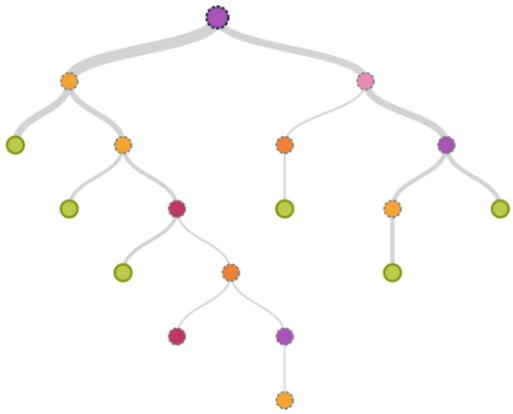
**Machine learning:des réseaux de neurones aux big data [INFO-Y112]**



Rapport

# Project 1: Overfitting, Underfitting and Metaparameters

Auteur:
Siéwé Kouéta Anicet : 00364245

Professeur:
Benoît Frénay

UNIVERSITÉ LIBRE DE BRUXELLES ULB

October 26, 2019

# Contents

# 1 Goal of the Project

Le fichier pdf qui a ete utiliser pour reponde au **task 2** se trouve dans le repertoire **doc/Q2_Graphe_tree_(max_feat**

# 2 Training and Test Datasets

les jeux de données utiliser se trouvent dans **Data/Adult_test.csv** et **Data/Adult_train.csv**

# 3 Training Decision Trees

## 3.1 task 1

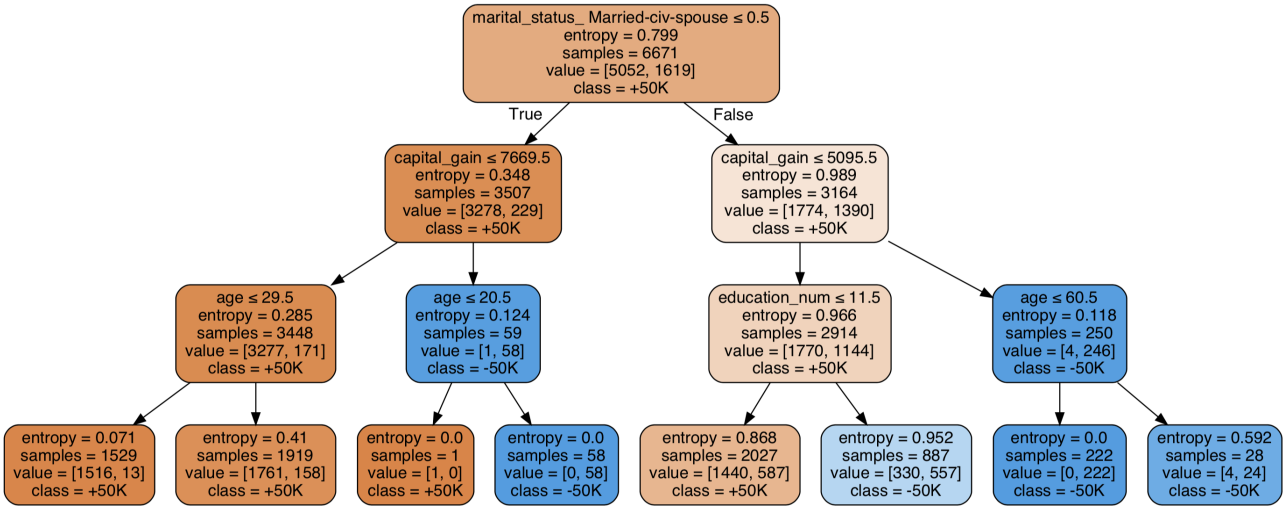



Figure 1: Abre de decision de profondeur 3

Au niveau des feuilles, on constate que les valeurs de l'entropie sont relativement partagée. Quand cette dernière prend une valeur nulle (*entropy=0.0*), nous sommes en presence de classes pures. Ce qui suppose que nous avons une bonne classification. Quand par contre l'entropie est élévée, on remarque qu'il y'a encore de l'incertitude (des zones d'ombre) dans les classes. Par conséquent, on peut clairement enduire que la profondeur de l'arbre n'est pas assez large car certaine feuilles (avec forte entropie) nécessitent encore un traitement supplémentaire.
Nous sommes clairement en underfitting car notre modele a une profondeur relativement faible par rapport au nombre de données a disposition.
l'erreur de d'entrainement obtenue est = **0.163** (soit une performence de *0.836* ) l'erreur de de teste(generalisation) obtenue est = **0.162** (soit une performence de *0.837* )

sur la courbe de la figure 2 ,Après avoir utilisé toutes nos données , le modèle utilise à peu près la même chose sur le jeu de generalisation que sur le jeu de d'entrainement - cela signifie que nous n'avons pas de surapprentissage (le modèle est très bien généralisé). En revanche, en termes absolus, l'erreur est élevée pour la generalisation mais les courbes s'aplatissent très rapidement, ce qui signifie que l'ajout de plus en plus de donnee ne change rien.

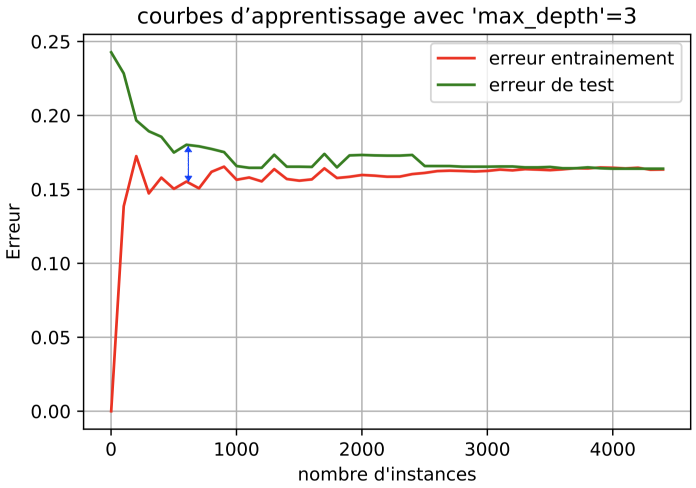



Figure 2: courbes d'apprentissage

## 3.2 task 2

Le metaparametre choisi pour cette partie a été le ***max_features*** (qui limite le nombre max de *features* a utiliser). Dans l'arbre generer avec ce metaparametre (voir *doc/Q2_ Graphe_ tree_ (max_ features).pdf*), Très vite on constate qu'indépendamment des valeurs qui lui était attribuée, on se retrouvait toujours avec des classes pures au niveau des feuilles (entropie nulle, taux de precision systématiquement égal=1, erreur d'entrainement toujours nulle). Nous sommes clairement en overfitting.
Notre 1er DT (voir figure 1), bien qu'imparfait, était néanmoins accessible (moins complexe). Tandis que celui de la figure *doc/Q2_ Graphe_ tree_ (max_ features).pdf*) est très peu accessible, trop profond et difficile à comprendre (plus complexe).

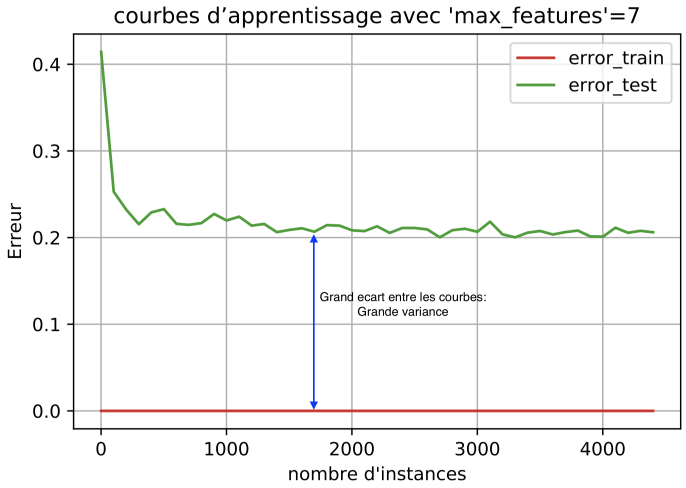



Figure 3: courbes pour max_features=7

Le nouvel écart entre les deux courbes pour le cas du metaparametre ***max_features*** suggère une augmentation substantielle de la *variance*. Le grand écart et la faible erreur d'entraînement

(Train_Error=0.0) indiquent également un problème d' *overfitting*.

# 4 Comparing Models of Increasing Complexity

task 3

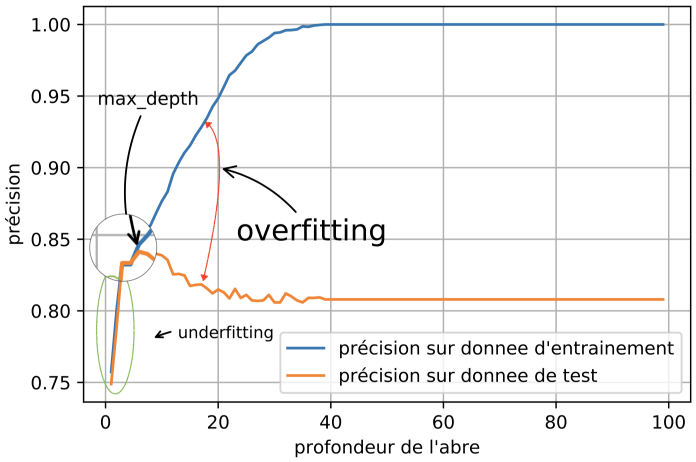



Figure 4: Evolution des Courbes precision pour max_depth=[1;100]

dans la premier portion(portion encercler en vert) de la figure 4 nous sommes en underfitting car nous avons un modele peut complexe (du a sa faible faible profondeur=[ 1, 4]) pour énormément de données. On a par consequent un taux de precision assez bas (soit pour l'entrainement que pour la generalisation).
la partir ZOUMER sur la figure 4 nous montre notre meilleurs modele car le taux de precision sur les données de test est a son pique le plus eleve, par consequent l'erreur de généralisation est a son niveau le plus bas.
apres ce pique, nous entrons progressivement en overfitting car avec la profondeur qui s'accroit, le modele tend a complexifier et surprendre des données. plus la profondeur de l'abre augmente, plus la precision sur les donnees d'entrainemet augmente jusqu'a provoquer un *overfitting*.
les resultal obtenuent respectes parfaitement le principe de la théorie vue en cour. cela montre ainsi que plus la complexiter du model augmente (Croissance de la Profondeur *max_depth* ), plus il a tentance a *overfitter*.

## 4.1 task 4

comme il a ete dit a la Q2, faisant varier la valeurs du metaparametre **max_features**, on se retrouve toujours avec un taux de precision pour les données d'entrainement égal a 1 par consequent une erreur toujours nulle.
Par contre notre taux de precision pour les données de test est toujours reste relativement bas [0.78 ; 0.81].
pour toutes les valeurs attribuée a notre metaparametre (**max_features**) Nous sommes en overfitting .

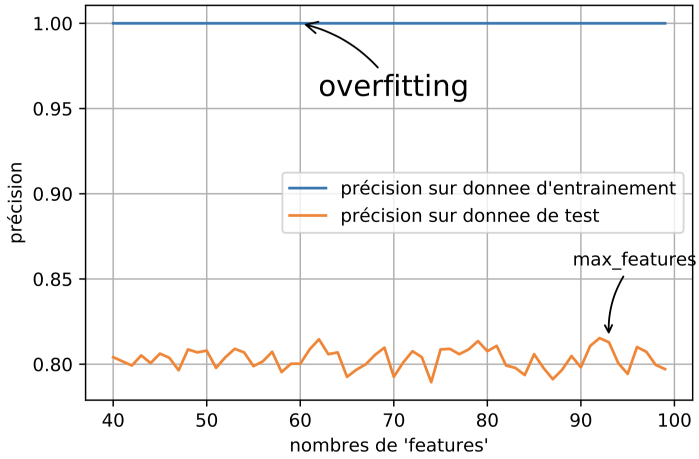



Figure 5: Evolution des Courbes precision pour max_feature=[40;100]

# 5 Choose your Model

## 5.1 task 5

consernant l'analyse des courbes de la Q3 (voir figure 4) on constate qu'a partir d'une profondeur de l'abre compris entre [ 6,  7] le model obtient de meilleur performence (*précision*). La meilleur profondeur de l'abre est **5**, car elle est celle qui correspond a la meilleur performance.

consernant l'analyse des courbes de la Q4 (voir figure 5), il est asser dificile d'identier a vue d'oeil une meilleur pour le metaparametre **max_features**, car qu'importe la valeur qu'on donnera a ce dernier, nous seront toujour en **overfit**. Néanmoins , si devait absoluman avoir une valeur, elle serais de max_features=93, car elle correspond a la precision max lors de la generalisation(testing).

```
MEILLEUR PROFONDEUR DE L'ABRE:

+ ------------------------------------------- +
|              Training |              Testing |
+ ------------------------------------------- +
| Profondeur  |      39 |                    5 |
+ ------------------------------------------- +
| precision   |     1.0 |   0.8422525358516963 |
| du Model    |         |                      |
+ ------------------------------------------- +
```

Figure 6: resulta de Algo pour la meilleur profondeur de l'abre