Julien DUPRAT
Hicheme BEN GAIED
L3 Informatique S6

Rapport Projet IA

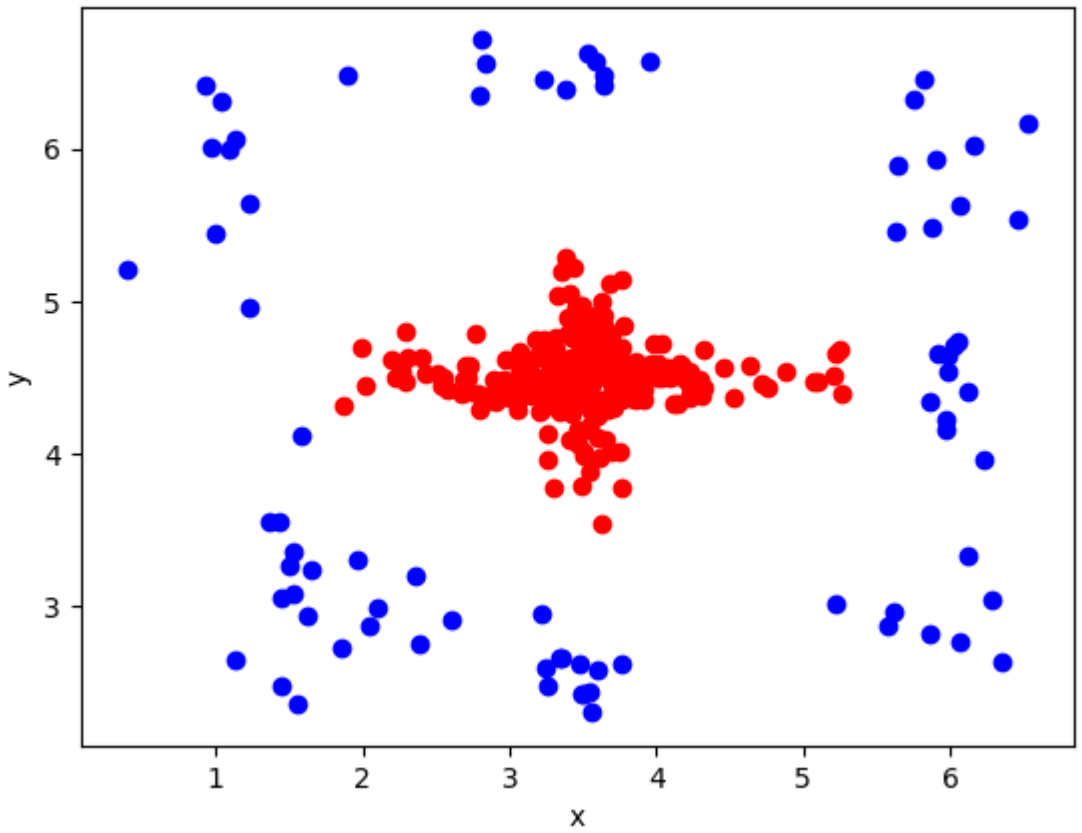
TABLE DES MATIÈRES

2 Données

Question 1 :
Dans le fichier data.csv il y a 330 données réparties de la manière suivante:

- 250 inlier (0)
- 80 outlier (1)

Question 2 :



Visualisation des données du fichier data.csv

Question 3 :
A première vue les outlier semblent être dans un rang A et les inliers dans un rang B

3 Evaluation

Question 1 :

$$\begin{pmatrix} 1000 & 2 \\ 30 & 5 \end{pmatrix}$$

A la lecture des coefficients, on remarque déjà que la matrice de confusion est très orientée pour trouver les true negative (TN) (différence de facteurs entre 30 et 500 avec le reste de la matrice). Les false negative (FN) sont aussi un peu représentés tandis que le reste est presque négligeable.
On prédit une exactitude ≃ 1 et l'exactitude pondérée ≃ 0,5 a vue de nez.
Les outliers étant les positives, on a une matrice qui semble cohérente car les outliers sont bien en nombre insignificatif. Ce qui prouve bien que ce sont des outliers à traiter.
Au total on a plus d'outliers non détectés que d'outliers apparents (vrais ou non) donc ça limitera quand même notre traitement. Cela dit les outliers non détectés ne devraient, par conséquent, pas être les pires et au final le traitement des outliers améliorera quand même sensiblement les résultats en principe.

Question 2 :
(on vous épargnera les calculs)

Exactitude pondérée = $\frac{\frac{TN}{TN+FP} + \frac{TP}{FN+TP}}{2}$ = 4001 / 7014

Exactitude = $\frac{TN + TP}{TN + FP + FN + TP}$ = 1005 / 1031

Précision = $\frac{TP}{TP + FP}$ = 5 / 7

Rappel = $\frac{TP}{FN + TP}$ = 1 / 7

Question 3 :

L'exactitude donne un score aussi bon, car TN >> TP ou FP ou FN
Par conséquent si on approxime on a Exactitude ≃ TN / TN = 1

Dans la formule de l'exactitude, si on a une des valeurs de la matrice de confusion beaucoup plus grande que les autres, elle fera pencher la balance soit vers 1 (TN ou TP) soit vers 0 (FN ou FP).
Ce qui est plutôt logique.

Question 4 :

Déjà une valeur qui est autant influencée par les différences de valeurs ne peut pas être pertinente lorsqu'on a une valeur grande devant les autres (comme on prend une échelle logarithmique pour les graphes si les valeurs sont trop écartées).
Mais dans notre cas précis, supposons qu'on arrive à supprimer tous les FP et les TP, la valeur changera peu, voire peu baisser car on supprime des TP.

4 Algorithmes

4.1 Decision Leaf

2) L'algorithme du k-mean a été implémenté à la main en choisissant aléatoirement les positions de départ et en les triant dans l'ordre (ex : si on obtient aléatoirement [2, 3, 1] on le transforme en [1, 2, 3] pour le manipuler plus facilement (cad centreL = 1, centreM = 2, centreR = 3))

3) lancez la commande python `python DecisionLeaf.py` pour voir les résultats sinon ils sont ci-dessous :

Real confusion Matrix:
[[225 25]
 [22 57]]

exactitude = 0.8571428571428571
exactitude pondérée = 0.8107594936708861
précision = 0.6951219512195121
rappel = 0.7215189873417721

voir graphes fournis

4.2 Superficial Tree

1) On a transformé la structure DecisionLeaf de la première partie qui nous a donné un nœud pour notre nouvelle structure et on a créé une nouvelle structure leaf qui contient les données en bout ainsi que si c'est une feuille d'inliers ou d'outliers rien de particulier.

2) Rien de particulier nous avons suivi l'algorithme.
Le seul point à ajouter est que les fichiers ST_toutes_donnees sont des fichiers que j'avais mis au point avant de me rendre compte que je n'utiliserais pas en fait les feuilles et que mon travail était à côté de ce qui m'était demandé.

3) Le modèle a l'air plutôt correct au vu de la matrice de confusion obtenue :

We have 329 entries with the Superficial Tree and k-mean coupled algorithms when there is really 329

Real confusion Matrix:
[[240 10]
 [7 72]]

exactitude = 0.9483282674772037
exactitude pondérée = 0.9356962025316455
precision = 0.8780487804878049
rappel = 0.9113924050632911

Une limite du modèle est que selon le critère choisi - pour séparer inliers et outliers - choisi en premier, on aura des résultats différents. En effet seule la dernière feuille compte et donc une entrée qui passerait haut la main le premier test mais serait un peu a la limite pour le second

pourrait être considéré comme outlier alors qu'il est en réalité inlier.
Une autre limite est le fait qu'on se repose sur l'algorithme de k-mean qui donne des résultats fluctuants de manière inhérente, d'autant plus si les données sont difficiles à séparer. Bien que les résultats restent globalement bons à chaque fois.

4.3 Generalized Tree

2)

Métriques \ Profondeur	1	2	3	4
TN	225	249	134	100
FN	22	13	21	30
TP	25	1	116	150
FP	57	66	58	49
Exactitude	0.8571	0.9574	0.5836	0.4529
Exactitude pondérée	0.8108	0.9157	0.6351	0.5101
Précision	0.6951	0.9851	0.3333	0.2462
Rappel	0.7215	0.8354	0.7342	0.6203

Plus on va en profondeur, moins on est précis. Ce qui est dû à la partie decision directe qui de manière plutôt bourine décide si des données sont inliers ou outliers. Le fait de faire tourner aussi le k-mean plusieurs fois sur des sets de données déjà épurés en grande majorité d'outliers fait qu'on finit par enlever des inliers de leur groupe et inversement. C'est l'algorithme même si on fait au minimum 3 groupes avec un outlier gauche, un inlier droit et le reste des inliers même si ce sont tous des inliers.

Au final les profondeur suivantes ne font que détériorer la performance de l'arbre. Ces inconvénients seraient probablement plus mitigés sur un set de données bien plus grandes (augmentant la chance que des outliers subsistent dans les inliers et soit éliminés plus nous allons profondément et inversement).

Sinon pour ce qui est de la profondeur 1 et 2 on retrouve globalement les mêmes résultats ce qui valide notre algorithme. Les variations peuvent être mises sur le dos du k-mean (il a bon dos).

Comparaison profondeur 1 (en premier) et DecisionLeaf (en second) :

The selected attribute is 0 with a standard variation of 1.0622874020253326
seuilA = 2.6453027037894183
seuilB = 4.578704326995005

The selected attribute is 0 with a standard variation of 1.0622874020253326
seuilA = 2.6453027037894183
seuilB = 4.578704326995005

3) Pour une profondeur égale à 2 nous obtenons les meilleurs résultats sur cet algorithme et ce set de données.