Intelligence artificielle

Projet – détection d'outliers par arbres de décisions

Sarah Moali

L3 S6 Informatique

Rapport

II- Données:

1- Nombre de données dans chaque classe:

-Inlier: 250 données.

-Outliers: 80 données

2- Capture d'écran du résultat:

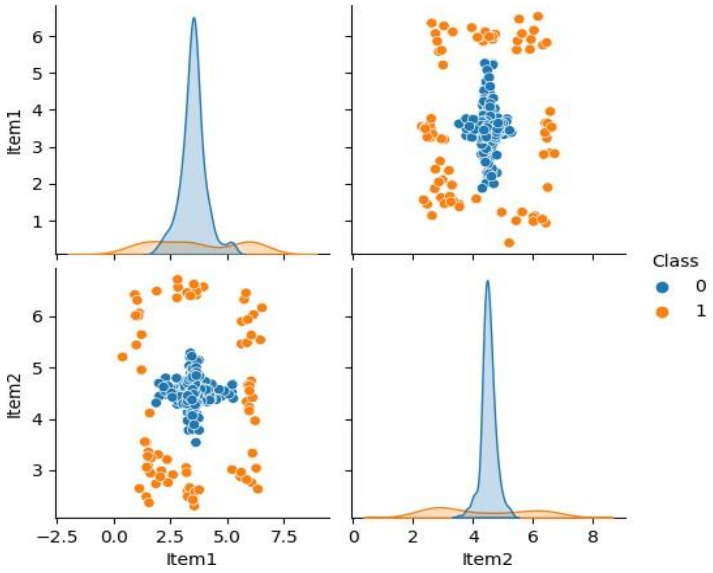

Figure1: Visualisation de données. Les outliers sont représentés en orange, les inliers en blue

3-la forme des données: les données ne nous permettent pas de dresser une fonction pour distinguer ou si j'ose dire classifier les données par une fonction qui sépare les différentes données.

Intelligence artificielle

Projet – détection d'outliers par arbres de décisions

III-Evaluation:

1. À la seule lecture des coefficients, le modèle nous semble pas bon parce qu' il a prédit que 5 valeurs vraies positive sur 35, donc les outliers prédit comme outliers est considérablement inférieurs aux valeurs fausses négatives (outliers prédits comme inliers) donc le modèle ne prédit pas bien les outliers car il a classé 30 valeurs sur 35 en tant que inliers. Par contre le modèle est bon par rapport à la prédiction des inliers car il a prédit 1000 valeurs qui sont vraies négatives qui sont des inliers, ces derniers sont largement supérieurs aux 2 valeurs inliers prédit comme outliers.

2- Calcule et comparaison de l'exactitude et l'exactitude pondérée de la matrice confusion:

Exactitude=(T N + T P)/(T N + F P + F N + T P)=(1000+5)/(1000+2+30+5)= 0.969= 96,9%

Exactitude pondéré= (TN/(TN+FP)+TP/(FN+TP))/2= ((1000)/ (1000+2)+(5)/(30+5))/2

=0.57= 57% .

L'exactitude est 96,9 se rapproche des 100% est largement supérieur de l'exactitude pondéré qui est de 56% .

3- L'exactitude donne un score aussi bon car le modèle prédit bien les inliers qui sont largement supérieur en nombre par rapport aux outliers qui sont que 35 outliers dans le modèle ,malgré le modèle ne prédit pas bien les outliers mais tellement le modèle prédit bien les inliers, aussi y a une grande différence dans le nombre des inliers par rapport aux outliers 1002 inliers>35 outliers donc l'exactitude est bon.

4- Elle n'est pas pertinente dans notre cas car nous cherchons à bien prédire les outliers et les différencier des inliers. Le modèle prédit 30 outliers comme inliers mais que 5 outliers comme outliers. Le modèle fait le contraire cela dit il prédit bien les inliers ce qui pas n'est pas pertinent dans notre cas car nous cherchons à bien prédire les outliers.

IV- Algorithme :

1- Arbre réduit à une feuille (classe DecisionLeaf):

Pour implémenter la feuille de décision Leaf on a créé la classe "directdecision" qui indique si une donnée est un inlier ou un outlier.

On a créé une fonction "getSpiltParameters" qui détermine l'attribut qui a le plus grand écart-type et les seuils a et b.

Intelligence artificielle

Projet – détection d'outliers par arbres de décisions

Dans la classe "decisionleaf" on devise les données en trois dataframes (L,M,R) qui représentent respectivement les données dont les données qui sont plus petites que le seuil a, les données comprises entre a et b et les données qui sont plus grandes que le seuil b.

- Evaluation de l'arbre de décision réduit à une feuille

: La matrice de confusion est: 204 46

24 56

L'exactitude pondéré est de :75.8%

L'exactitude est de :78.78787878787878%

La précision est de :54.90196078431373%

Le rappel est de :70.0%

L'apprentissage avec l'arbre réduit a une feuille nous semble qui donne des résultats raisonnable car il nous donnee exactitude de 75% car le modèle prédit plus de valeurs vraies positives (outliers comme outliers) que des valeurs fausses négatives(outliers comme inliers) 56>24, il prédit aussi plus d'inliers comme inliers que de inliers comme outliers 204>46

2- Arbre superficiel (buildFinalDecisionTree):

Pour la structure de donnée utilisée pour implémenter cet arbre on s'est servi de tout ce qui a été implémenté pour la première partie en rajoutant la classe Node qui est le noeud de notre arbre.

- Evaluation de l'arbre superficielle :

La matrice de confusion du deuxième algorithme est: 219 31

23 57

L'exactitude est de :83.63636363636363%

L'exactitude pondéré est de :79.425%

La précision est de :64.77272727272727%

Le rappel est de :71.25%

On constate une remarquable amélioration par rapport L'apprentissage avec l'arbre réduit a une feuille, l'exactitude est passée aussi de 78% jusqu'à 83%, car dans notre matrice le nombre des valeurs fausses négative à diminuer d'une valeurs 24 passée à 23 car le modèle a prédit une valeurs vraies positives en plus(56 passée à57), ainsi le nombre des valeurs

fausses positives a diminuée jusqu'à 31 cela en passant de 46 à 31, et ces valeurs diminuer sont augmentées dans les valeurs vraies positives cela en passant de 204 à 219

3- Le tableau reprenant l'ensemble des métriques:

	exactitude	exactitudepondere	rappel	precision
hauteur max=1	0.787879	0.75800	0.7000	0.549020
hauteur max=2	0.796970	0.79375	0.7875	0.557522
hauteur max=3	0.560606	0.55700	0.5500	0.287582
hauteur max=4	0.445455	0.42150	0.3750	0.184049

On constate bien que l'arbre avec hauteur max =1 est identique à notre arbre de décision réduit à une feuille car on a les même coefficient

4.3

avec hauteur maximum de 2 ont un bon résultat car c'est lui qui a le coefficient d'exactitude le plus grand en le dépassant on est en sur-apprentissage .