Rapport de RODD
TP n°6

Antonio Tavares

29 avril 2022

1 Jeux de données

— `ecoli.txt` : 7 attributs par instance, 327 instances, 5 classes. La classe regroupant le nombre minimal d'instances en possède 20, le choix a été fait de supprimer les instances étant dans des classes de petites tailles ($\leq$ 5 instances par classe).
— `prnn.txt` : 2 attributs par instance, 250 instances, 2 classes. Autant d'instances dans chaque classe (125).

Ces deux jeux de données proviennent du site `www.openml.org`.

2 Résultats : `main()`

La fonction `main()` n'utilise pas de regroupement, les performances sur les 5 jeux de données considérés sont les suivantes :

	Séparation	Temps (s)	gap	Erreurs train/test
D = 2	Univarié	1.3s	0.0%	5/1
	Multivarié	0.9s	0.0%	1/0
D = 3	Univarié	8.0s	0.0%	0/4
	Multivarié	0.8s	0.0%	0/2
D = 4	Univarié	10.5s	0.0%	0/4
	Multivarié	4.2s	0.0%	0/1

TABLE 1 – Résultats jeu de données `iris` (train size 120, test size 30, features count : 4)

	Séparation	Temps (s)	gap	Erreurs train/test
D = 2	Univarié	12.4s	0.0%	10/2
	Multivarié	1.4s	0.0%	0/2
D = 3	Univarié	30.2s	3.7%	6/2
	Multivarié	3.2s	0.0%	0/1
D = 4	Univarié	30.5s	7.7%	11/2
	Multivarié	18.9s	0.0%	0/2

TABLE 2 – Résultats jeu de données `seeds`

	Séparation	Temps (s)	gap	Erreurs train/test
D = 2	Univarié	14.5s	0.0%	5/2
	Multivarié	0.3s	0.0%	0/2
D = 3	Univarié	27.1s	3.7%	0/2
	Multivarié	1.2s	0.0%	0/1
D = 4	Univarié	18.9s	7.7%	0/1
	Multivarié	3.7s	0.0%	0/3

TABLE 3 – Résultats jeu de données `wine`

3 Question d'ouverture

Nous avons décidé de tester d'autres méthode de clustering pour les comparer aux méthodes déjà implémentées ; nous avons implémenter les méthodes classiques DBscan et Kmean, bien documentées sur internet.