UNIVERSITÉ DE STRASBOURG

Rapport – projet IA -2022

MARMAR Rayan
&
EIZOUKI Issam

Intoduction :

Partie 1 :

1. Ces données comportes 14 attributs (A → N).
2. Les instances sont catégorisés sous 4 différentes classes 0, 1, 2 et 3.
3. Classe 0 : 674 instances
   Classe 1 : 908 instances
   Classe 2 : 472 instances
   Classe 3 : 244 instances
4. D'après la distribution d'un échantillon de 100 instances selon les attributs A et B dans la figure 1, les données sont inséparables.
5. Oui, pour le modele du resau de neurones car, en effet les classes sont attribuées avec un id numérique, cela posera un problème durant l'apprentissage automatique, et c'est que le modèle peut entrainer un biais à cause de la relation d'ordre entre les differentes classes. Pour cette raison nous avons choisi de representer les classe en un encodage binaire **'one hot'** vue en tp.
6. Pour éviter d'avoir un mauvais biais ou variance, on va séparer les données, une partie pour le jeu d'entrainement (80%) et l'autre partie (20%) pour le jeu de test. En plus on divise la première partie en jeu d'entrainement et en jeu de validation pour pouvoir à tout moment détécter une forte augmentation de l'erreur (le taux de la validation s'éloigne du taux d'entrainement) et appliquer la méthode de **"l'early stopping"** .

Partie 2 :

– Arbre de décision :

Les quartiles sont les 3 indices avec lesquelles on peut diviser le jeu de données en 4 différentes parties tel que chacune contient le quart des données.
On les a obtenus en utilisant la fonction quantile de Pandas avec les attributs suivants:

- 0.25 pour le premier quartile.
- 0.5 pour le deuxième quartile (médiane).
- 0.75 pour le troisème quartile.

Partie 3 :

Modèle (decision tree 4)

Classes	**C0**	**C1**	**C2**	**C3**
Accuracy	0.83	0.86	0.84	0.9
Precision	0.71	0.76	0.59	0.67
Recall	0.83	0.88	0.49	0.125
F1-score	0.77	0.82	0.54	0.21

Modèle (decision tree 4)

		0	**1**	**2**	**3**
True label	**0**	130	19	6	1
	1	15	147	2	2
	2	25	20	44	0
	3	12	8	22	6

Predicted label

Modèle(Neural network relu 10-8-6)

Classes	**C0**	**C1**	**C2**	**C3**
Accuracy	0.94	0.94	0.94	0.92
Precision	0.97	0.9	0.8	0.63
Recall	0.86	0.93	0.9	0.65
F1-score	0.91	0.92	0.85	0.64

Modèle (Neural network relu 10-8-6)

		0	**1**	**2**	**3**
True label	**0**	135	9	5	7
	1	0	155	4	7
	2	2	3	80	4
	3	2	5	10	31

Predicted label

Modèle(Neural network relu 6-4)

Classes	C0	C1	C2	C3
Accuracy	0.94	0.94	0.94	0.92
Precision	0.94	0.91	0.9	0.65
Recall	0.87	0.95	0.92	0.5
F1-score	0.9	0.93	0.85	0.56

Modèle (Neural network relu 6-4)

		0	**1**	**2**	**3**
True label	**0**	136	6	5	9
	1	3	157	5	1
	2	2	2	82	3
	3	4	8	12	24

Predicted label

Modèle(Neural network tanh 10-8-4)

Classes	C0	C1	C2	C3
Accuracy	0.94	0.94	0.92	0.9
Precision	0.9	0.87	0.73	0
Recall	0.91	0.98	0.94	0
F1-score	0.91	0.92	0.83	0

Modèle (Neural network tanh 10-8-4)

		0	**1**	**2**	**3**
True label	**0**	142	9	5	0
	1	0	163	3	0
	2	2	3	84	0
	3	13	13	22	0

Predicted label

Partie 4 :

Comparaison des modèles :
* Tout depend de ce qu'on cherche à prédire
Par exemple, si on souhaite faire un test de maladie, on doit se concentrer sur l'identfication des cas positif, dans ce cas nous cherchons le modèle ayant le meilleur **rappel → meilleur modèle dans ce cas est y_pred_NN_tanh_10-8-6.csv.**
- En ne pas oubliant que que le rappel s'approche de 0 (pas de vrai positive)

* Si maintenant nous souhaitons identifier seulement à identifier uniquement les points de données pertinents, on regarde **l'exactitude** du modèle → **meilleur modèle dans notre cas :y_pred_NN_relu_10-8-4.csv**.

*Pour avoir une meilleur balance entre le rappel et la précision, nous devons dans ce cas, regarder le meilleur F1-score (la moyenne harmonique entre le rappel et la precision), dans ce cas notre **meilleur modèle sera y_pred_NN_tanh_10-8-6.csv**