

TP 1 Mesures de performance

Rahiche Messaoud | Groupe 3

Krizou Amani | Groupe 3

Adda Redouane | Groupe 3

Mendil Yousra | Groupe 3

Table des matières

1	KNN	3
1.1	L'implémentation de l'Algorithme	3
1.2	Matrice de confusion :	4
1.3	Le rappel, La precision et Le taux de faux positif :	4
2	MVS	5
2.1	Courbe ROC :	5
2.2	Matrice de confusion :	5
2.3	Le rappel, La precision et Le taux de faux positif :	5
3	Arbres de décision	6
3.1	Courbe ROC :	6
3.2	Matrice de confusion :	6
3.3	Le rappel, La precision et Le taux de faux positif :	6
4	Réseaux de neurones Perceptron	7
4.1	Courbe ROC :	7
4.2	Matrice de confusion :	7
4.3	Le rappel, La precision et Le taux de faux positif :	7

1 KNN

1.1 L'implémentation de l'Algorithme

1. Pseudo code :

- Charger les données
- Initialiser la valeur de k.
- Calculez la distance entre les données de test et chaque ligne de données d'entraînement.
- Trier les distances calculées par ordre croissant en fonction des valeurs de distance.
- Obtenir les k premières lignes du tableau trié
- Obtenir la classe la plus fréquente de ces lignes
- Renvoie la classe prédite

2. Python code :

```
1 def euc_dist(x1, x2):
2     return np.sqrt(np.sum(x1-x2)**2)
3
4 def KPP(x,Xt,Yt):
5     predicted_labels = [predict(x)]
6     return np.array(predicted_labels)
7
8 def predict(x):
9     k=10
10    distances = np.array([euc_dist(x, xt) for xt in Xt])
11
12    #get the k nearest neighbors and labels:
13    kdist = distances.argsort()[0:k]
14    knear = [Yt[i] for i in kdist]
15
16    #Illustrating the k-nearest neighbor:
17    plt.figure(figsize=(20,10))
18    for i in range(10):
19        d= Xt[kdist[i],:].reshape((20, 20))
20        d=np.transpose(d)
21        plt.subplot(1,10,i+1)
22        plt.title('label ' + str(Yt[kdist[i]]))
23        plt.imshow(d,cmap='gray')
24
25    #majority vote:
26    majority = Counter(knear).most_common(1)
27    return majority
```

1.2 Matrice de confusion :

On a fait l'affichage de la matrice de confusion de chaque classe dans le fichier jupyter, alors ici on a cree un tableau qui les regroupe :

Classe	0	1	2	3	4	5	6	7	8	9
VP	196	266	122	126	140	127	145	142	135	150
VN	2765	2878	2841	2854	2799	2859	2792	2847	2777	2801
FP	241	112	169	151	206	140	193	157	217	198
FN	131	77	201	202	188	207	203	187	204	184

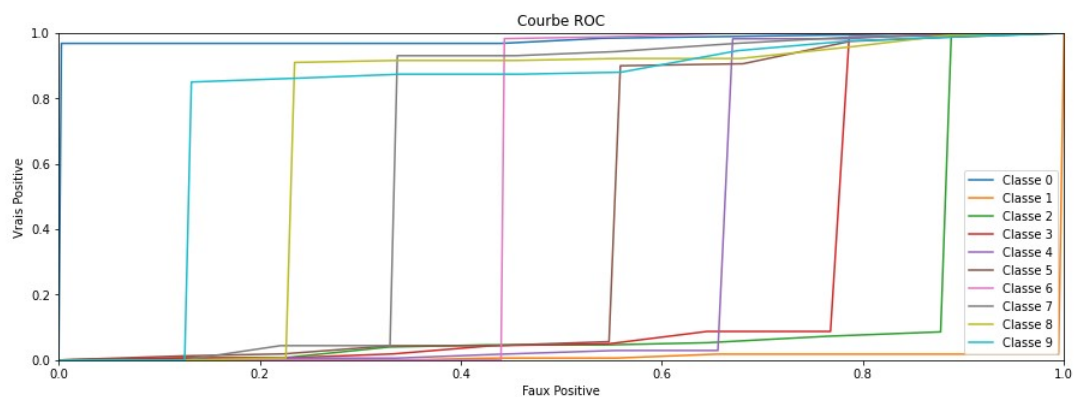
1.3 Le rappel, La precision et Le taux de faux positif :

Le tableau suivant englobe les mesures de performance de chaque classe :

Classe	0	1	2	3	4	5	6	7	8	9
Rappel	0.599	0.775	0.377	0.384	0.426	0.380	0.416	0.431	0.398	0.449
Précision	0.448	0.703	0.419	0.454	0.404	0.475	0.428	0.474	0.383	0.431
Taux de FP	0.448	0.592	0.456	0.427	0.522	0.403	0.487	0.456	0.515	0.518

2 MVS

2.1 Courbe ROC :



2.2 Matrice de confusion :

On a fait l'affichage de la matrice de confusion de chaque classe dans le fichier jupyter, alors ici on a cree un tableau qui les regroupe :

Classe	0	1	2	3	4	5	6	7	8	9
VP	184	163	137	144	163	135	173	141	151	142
VN	1473	1493	1500	1479	1474	1490	1486	1497	1487	1490
FP	6	3	14	16	8	25	3	18	16	25
FN	4	8	16	28	22	17	5	11	13	10

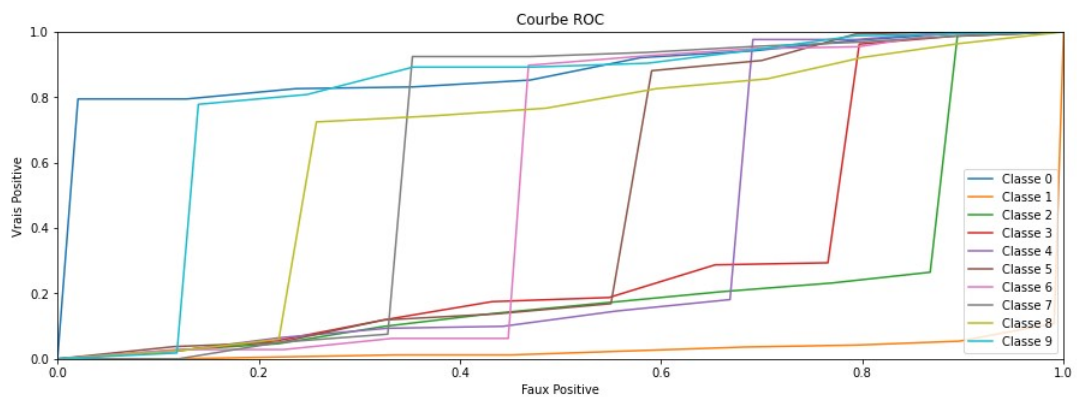
2.3 Le rappel, La precision et Le taux de faux positif :

Le tableau suivant englobe les mesures de performance de chaque classe :

Classe	0	1	2	3	4	5	6	7	8	9
Rappel	0.968	0.981	0.907	0.9	0.953	0.843	0.982	0.886	0.904	0.850
Précision	0.978	0.953	0.895	0.837	0.881	0.888	0.971	0.927	0.920	0.934
Taux de FP	0.4	0.727	0.533	0.636	0.733	0.404	0.625	0.379	0.448	0.285

3 Arbres de décision

3.1 Courbe ROC :



3.2 Matrice de confusion :

On a fait l'affichage de la matrice de confusion de chaque classe dans le fichier jupyter, alors ici on a cree un tableau qui les regroupe :

Classe	0	1	2	3	4	5	6	7	8	9
VP	151	148	109	107	136	114	147	135	111	127
VN	1447	1487	1475	1460	1462	1446	1461	1471	1444	1468
FP	39	18	42	53	35	46	29	24	56	40
FN	30	14	41	47	34	61	30	37	56	32

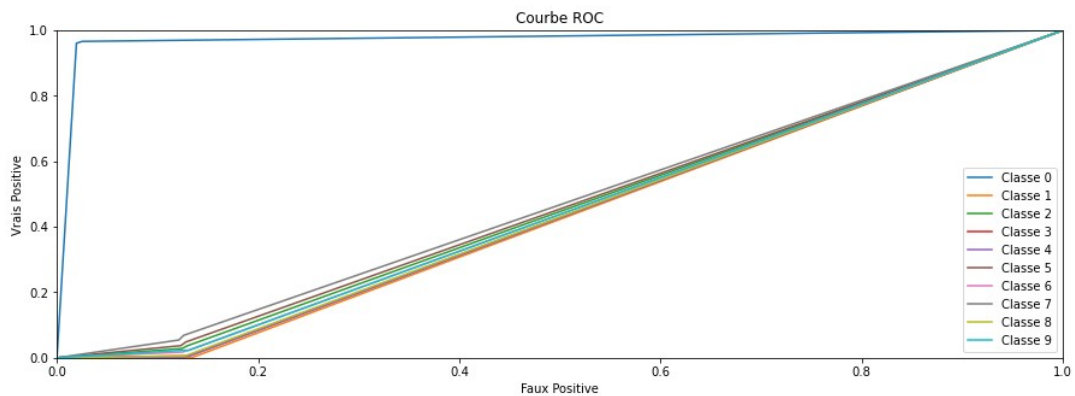
3.3 Le rappel, La precision et Le taux de faux positif :

Le tableau suivant englobe les mesures de performance de chaque classe :

Classe	0	1	2	3	4	5	6	7	8	9
Rappel	0.794	0.891	0.721	0.668	0.795	0.7125	0.835	0.849	0.664	0.760
Précision	0.834	0.913	0.726	0.694	0.8	0.651	0.830	0.784	0.664	0.798
Taux de FP	0.434	0.4375	0.493	0.47	0.492	0.570	0.508	0.606	0.5	0.444

4 Réseaux de neurones Perceptron

4.1 Courbe ROC :



4.2 Matrice de confusion :

On a fait l'affichage de la matrice de confusion de chaque classe dans le fichier jupyter, alors ici on a cree un tableau qui les regroupe :

Classe	0	1	2	3	4	5	6	7	8	9
VP	146	0	0	0	156	0	0	3	0	0
VN	1491	1503	1504	1497	179	1485	1505	1491	1481	1505
FP	6	164	163	170	1	182	162	166	186	162
FN	24	0	0	0	1331	0	0	7	0	0

4.3 Le rappel, La precision et Le taux de faux positif :

Le tableau suivant englobe les mesures de performance de chaque classe :

Classe	0	1	2	3	4	5	6	7	8	9
Rappel	0.960	0.0	0.0	0.0	0.993	0.0	0.0	0.017	0.0	0.0
Précision	0.858	/	/	/	0.104	/	/	0.3	/	/
Taux de FP	0.8	0.0	0.0	0.0	0.999	0.0	0.0	0.040	0.0	0.0