



Master Informatique EID2

Traitement numérique des données

TP3 Introduction à la classification

```
1 iris = datasets.load_iris()
2 X = iris.data
3 Y = iris.target
```

```
1 #####
2 # Plus Proche Voisin #
3 #####
4
5 #-----#
6 ## 1. Créez une fonction PPV(X,Y) ##
7 #-----#
8
9 def PPV(X,Y) :
10     w = metrics.pairwise.euclidean_distances(X, X)
11     t = np.argsort(w)
12     cmin = np.array(t[:,1])
13     pred = Y[cmin]
14     return pred
```

```
1 #-----#
2 ## 2- Modifiez la fonction pour calculer et renvoyer l'erreur de prédiction ##
3 #-----#
4
5 def erreur_PPV(Y1,Y2):
6     dif = Y1 == Y2
7     pct = 0.0
8     n = len(dif)
9     for i in range(n):
10         if dif[i] == 0 :
11             pct = pct + 1
12     pct = pct/n*100.0
13     return pct
```

```
1 #-----#
2 ## 3. Testez sur les données Iris ##
3 #-----#
4
5 P = PPV(X,Y)
6 print('le plus proche voisin :\n',PPV(X,Y))
7 print('Erreur de :',erreur_PPV(Y,P))
```

```
le plus proche voisin :  
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 2 1  
1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 1 2 2 2 2  
2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2  
2 2]  
Erreur de : 4.0
```

```
1 #-----#
2 ## 4. Testez la fonction des K Plus Proches Voisins de sklearn (avec ici K = 1) ##
3 #-----#
4
5 def kPPV(X,Y,k):
6     neigh = KNeighborsClassifier(n_neighbors=k)
7     neigh.fit(X, Y)
8     return neigh.predict(X)
```

[illegible][illegible]


```

1 #-----#
2 ## 2. Testez sur les données Iris et Fonction erreur ##
3 #-----#
4
5 irisCB = CBN(X,Y)
6 print(irisCB)
7 print('Erreur de :',erreur_PPV(irisCB,Y))

```

```

[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 1 2 2 2 2
2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2
2 2]
Erreur de : 4.0

```

```

1 #-----#
2 ## 3. Testez la fonction du Classifieur Bayesien Naïf inclut dans sklearn ##
3 #-----#
4
5 irisCBN = naive_bayes.GaussianNB().fit(X,Y).predict(X)
6 print(irisCBN)
7 print('Erreur de :',erreur_PPV(irisCBN,Y))

```

```

[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 1 2 2 2 2
2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2
2 2]
Erreur de : 4.0

```