

## Séance de TP 7 Cartes Laplaciennes

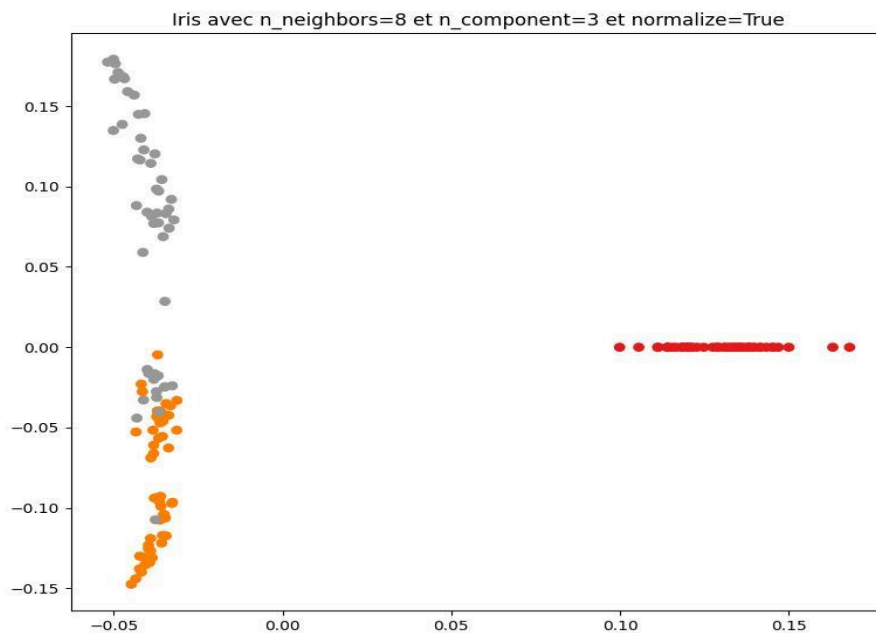
---

### 1- Carte Laplacienne

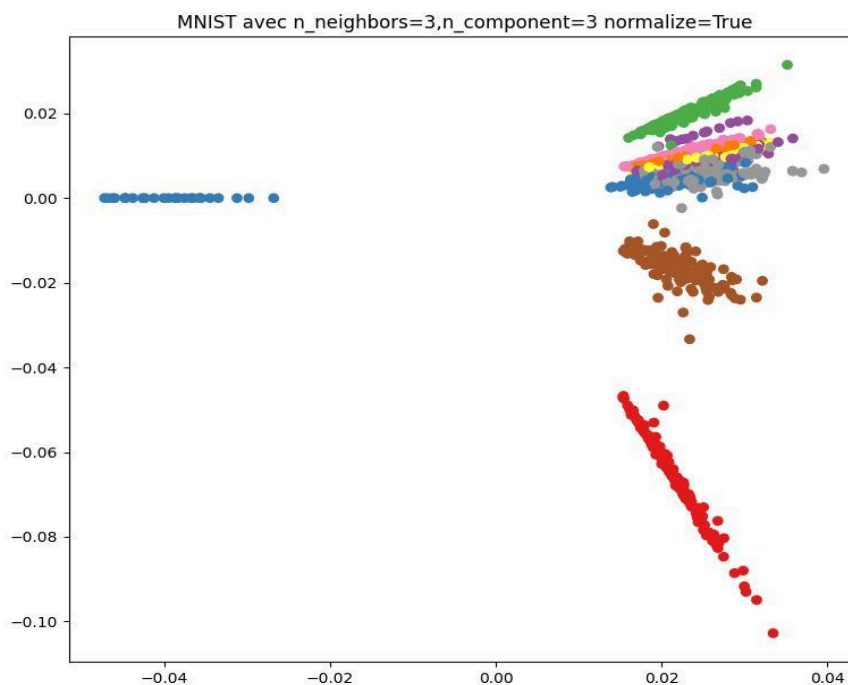
- a. Programmer la méthode `laplacian_map` en vous conformant au modèle proposé.

```
def my_laplace(X,y,n_neighbors,n_component,i,j,normalize=False):  
    m=X.shape[0]  
    W = kneighbors_graph(X, n_neighbors,mode='distance')  
    W=(W+W.T)*0.5  
    D=sparse.diags(np.asarray(W.sum(axis=1)).flatten())  
    if normalize==True:  
  
L=np.eye(m)-np.sqrt(sparse.linalg.inv(D))@W@np.sqrt(sparse.linalg.inv(D  
) )  
    else:  
        L=D-W  
    [yl,YL]=sparse.linalg.eigsh(L,k=n_component,which='SM')  
    plt.figure(figsize=(9,8))  
    plt.scatter(YL[:,i],YL[:,j],c=y,cmap=plt.cm.Set1)  
    plt.show()
```

- b. Visualiser le résultat de Cartes Laplaciennes sur les deux dataset IRIS et MNIST.



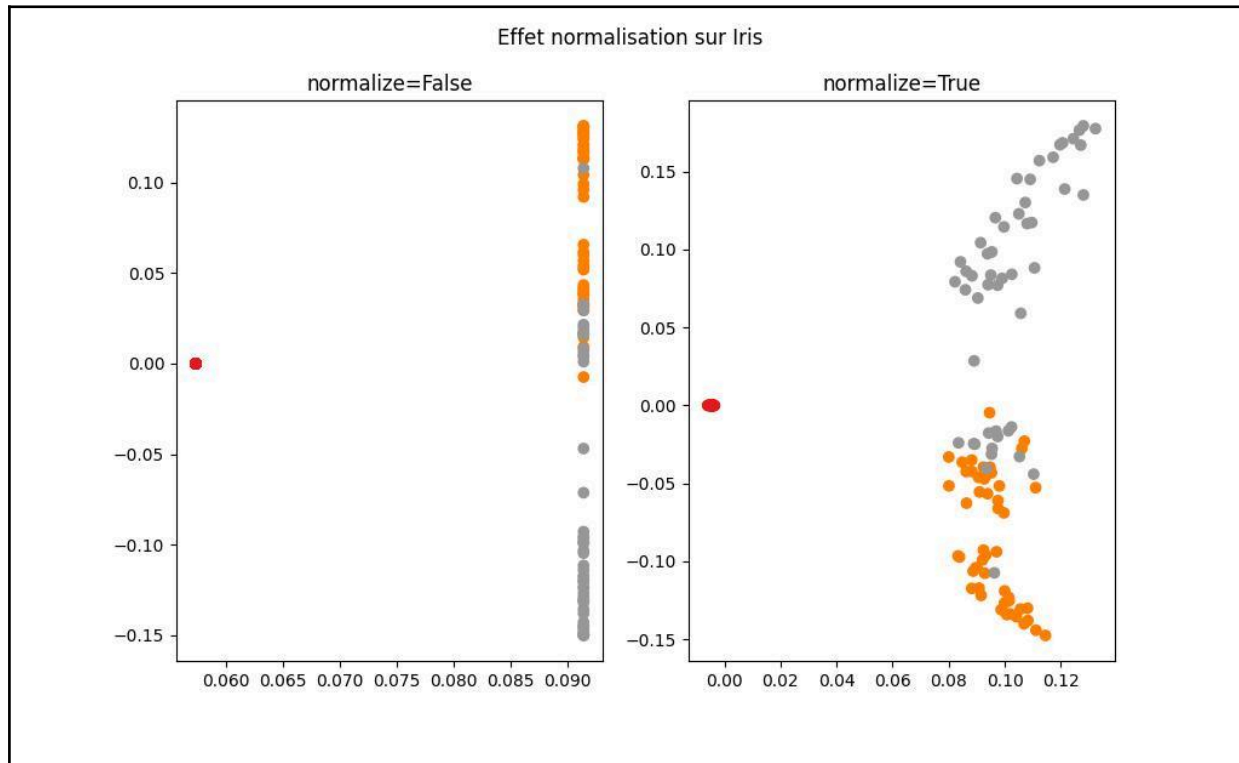
***Pour iris on voit que les 3 classes bien sont représentés avec trois composantes principales les données de même classe sont restées ensemble***



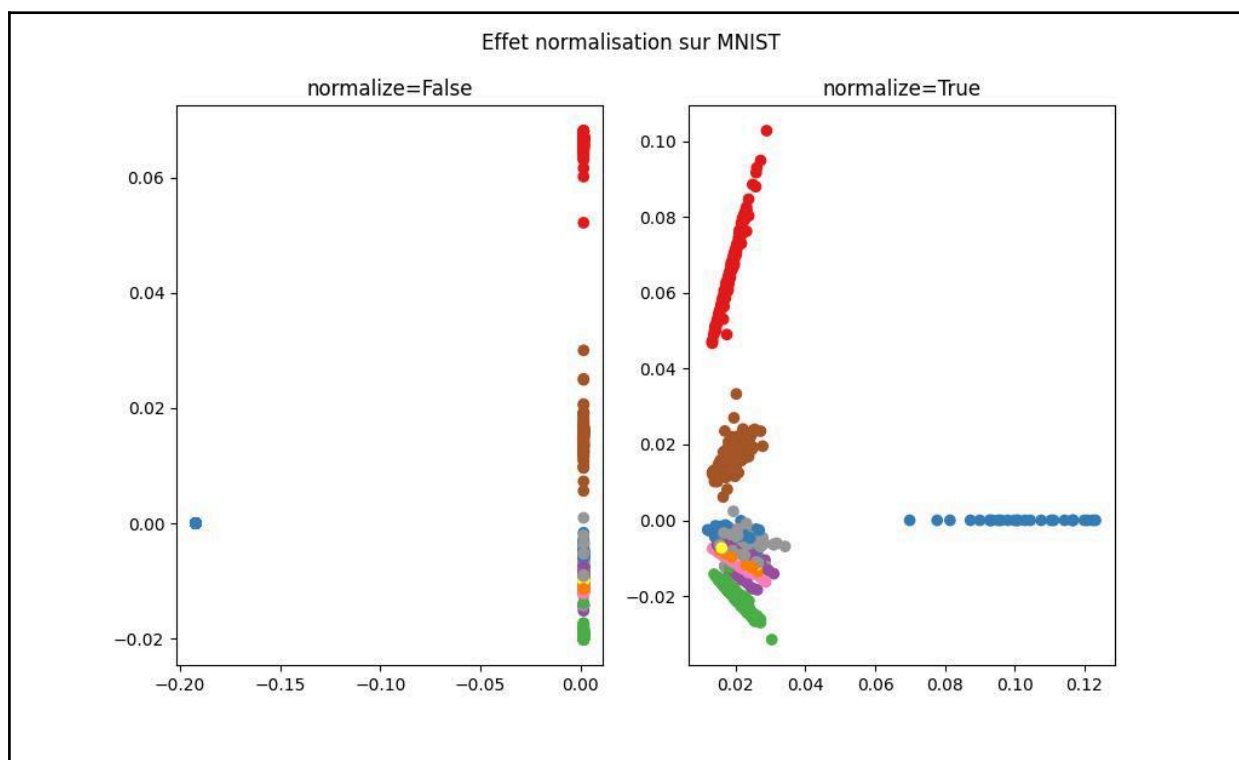
***Pour MNIST on voit aussi que les classes sont représentés avec trois composantes principales***

**Mais le model est plus performant sur les données Iris**

c. Déterminer l'influence de la normalisation du Laplacien sur les résultats obtenus.



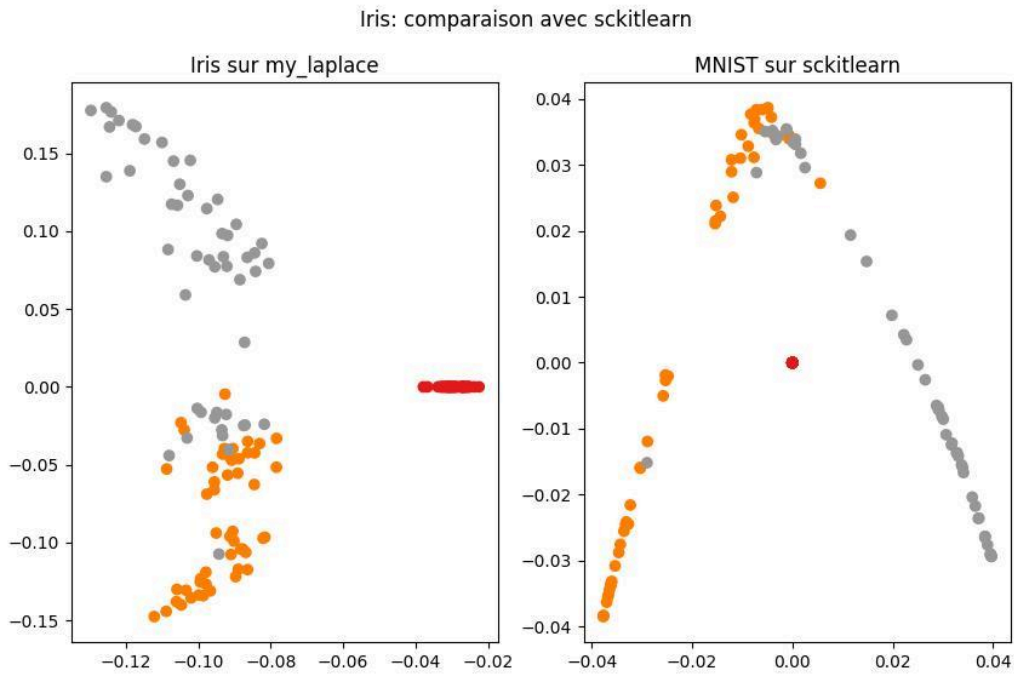
d. Déterminer l'influence du nombre de voisins sur les résultats obtenus.



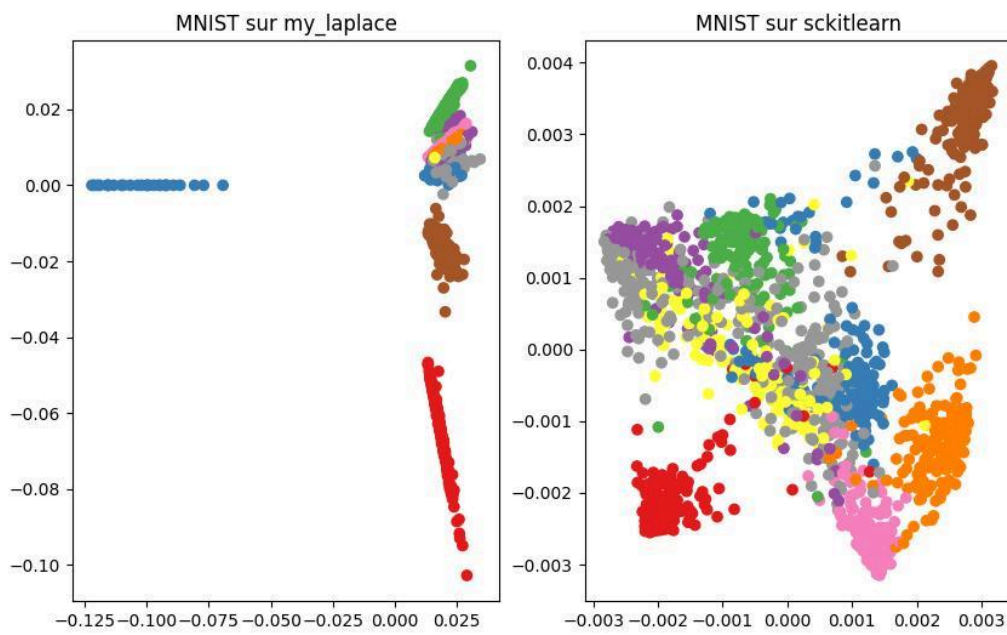
***Avec la normalization on représente bien la répartition des classes et on a plus de variance et plus de points projetés***

## 2- Comparaison avec Scikitlearn

- Comparer vos résultats à ceux obtenus avec scikitlearn pour les mêmes hyper-paramètres.



MNIST: comparaison avec sckitlearn



***Avec scikit-learn on presque le même résultat pour les deux dataset c'est sure juste que peut être on a pas pris le même nombre de voisins***