# TP1 – ACP - Classification et reconstruction

Dans ce TP, nous allons utiliser une partie de la base de visages “Labeled Faces in the Wild” provenant de http://vis-www.cs.umass.edu/lfw/. Cette base contient 5749 personnes et 13233 images de taille 62 x 47 pixels. Certaines personnes ne sont représentées qu’une seule fois tandis que d’autres sont représentées très souvent (plus de 80 fois). Nous utiliserons ici seulement 7 personnes représentées 1288 fois.
Nous utiliserons google colab pour travailler. Pour cela, aller dans google colab et taper nouveau notebook en python 3.

Ouvrez le volet de gauche et cliquer sur fichier. Attendre quelques instants et cliquer sur importer pour charger le fichier de données TP1.npy.

## I. Chargement des données

Charger les données avec les commandes :

```
[X, y, name]=np.load("TP1.npy", allow_pickle=True)
```

**Question**
Sachant que X resprésente les features, y les labels et name le nom des classes, déterminer la taille des images , le nombre d’images et le nombre de classes.

Partitionner la base en une base d’apprentissage et une base de test en mettant 25% des données en test (fonction train_test_split()) pour obtenir les variables X_train, X_test, y_train et y_test.

**Question**
Combien y a-t-il d’images en train et en test ?

## II. Classification avec les kppv

Mettre en forme les données (train et test) en utilisant la fonction classe StandardScaler.

**Question**
A quoi sert cette fonction, en quoi consiste la mise en forme des données ?

Définir le classifieur 1PPV en utilisant la classe KNeighborsClassifier(). On souhaite utiliser la distance euclidienne dans le 1PPV.
Réaliser la classification des exemples de test en utilisant la méthode predict().
Afficher la matrice de confusion (fonction confusion_matrix()) et le rapport de classification (fonction classification_report())

**Questions**
Que représente la matrice de confusion ? Que vaut sa somme ? Est-ce que les classes sont équilibrées ?

Que représente le rapport de classification ? Retrouver chacun de ses éléments à partir de la matrice de confusion

Faire varier le K des KPPV et tracer l’évolution du taux de bonne reconnaissance.

**Questions**
Conclusion ? Interpréter l'évolution des résultats en fonction de K

Réaliser les mêmes tests avec la distance de Manhattan.

**Questions**
Conclusion ? Interpréter l'évolution des résultats en fonction de K

## III. Analyse en composantes principales et classification

Définissez la décomposition en utilisant la fonction PCA() et l'appliquer sur les données en utilisant la méthode fit().

Utiliser la méthode pca.explained_variance_ratio_() et tracer les variances.
Faire varier le nombre nb de composantes préservées et réaliser la classification en utilisant le 1PPV et la distance de Manhattan. Conclure.

**Questions**

Que représentent les valeurs renvoyées par pca.explained_variance_ratio_ ?

Combien de composantes sont nécessaire pour avoir une bonne classification

Comment varient les temps de calcul en fonction du nombre de composantes ?

## IV. Analyse en composantes principales et reconstruction

Définissez la décomposition en utilisant la fonction PCA() en conservant 50 composantes et l'appliquer sur les données en utilisant la méthode fit().
Récupérer les vecteurs propres en utilisant une méthode de PCA(). Redimensionner les vecteurs propres en images propres (np.reshape()) de manière à pourvoir les visualiser sous forme d'images (array de taille 50x62x47). On utilisera la fonction plot_gallery() pour la visualisation.

**Questions**
Que représentent les vecteurs propres ? Quelle est leur taille ?

On souhaite transmettre ces images en utilisant le moins de bande passante possible. Pour cela, les 50 images propres sont transmises une fois. Pour chaque nouvelle image, on transmet uniquement ses composantes dans le nouveau système d'axe de dimension 50. L'image est ensuite reconstruite à l'arrivée.
Reconstruisez les images de test à partir d'une des méthodes de PCA() puis remettre les données dans leur forme initiale en utilisant une des méthodes de StandardScaler. Afficher les images reconstruites et les comparer aux images initiales.

**Questions**
Lorsque l'on conserve 50 composantes, quel est le taux de compression, pourquoi ?
Quel est le principe de la reconstruction des images ?

Faire varier le nombre de composantes conservées et calculer l'erreur de reconstruction (norme L2). Afficher l'erreur de reconstruction en fonction du nombre de composantes.

**Questions**
- Que représente l'erreur de reconstruction ?
- Comment varie-t-elle en fonction du nombre de composantes ?

## V. Annexe

```
def plot_gallery(images):
    #       Affiche les 12 premières images contenues dans images
    #       images est de taille Nb image*Ny*Nx
   plt.figure(figsize=(7.2, 7.2))
   plt.subplots_adjust(bottom=0, left=.01, right=.99, top=.90, hspace=.35)
   for i in range(12):
      plt.subplot(3, 4, i + 1)
      plt.imshow(images[i], cmap=plt.cm.gray)
      plt.xticks(())
      plt.yticks(())
```