

Exercice : Reconnaissance des nombres

Dans cet exercice, nous allons à nouveau utiliser Scikit-learn qui est une bibliothèque libre Python destinée à l'apprentissage automatique, et notamment le jeu de données [digits](#) : nous disposons de 1 797 images de 8×8 pixels en niveaux de gris représentant les chiffres de 0 à 9.

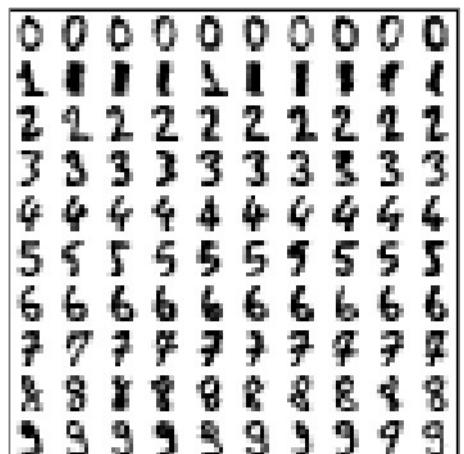
Figure : Extrait du jeu de données digits

Ces données nous sont fournies par l'intermédiaire de 2 tableaux :

- X est une matrice de :
 - 1797 lignes pour les images,
 - 64 colonnes pour les caractéristiques des images.

Pour tout $k \in [0, 1796]$, $X[k]$ est un vecteur de \mathbb{R}^{64} qui représente l'image digitalisée d'un chiffre.

- Y est un vecteur de taille 1 797.



Pour tout $k \in [0, 1796]$, $Y[k]$ correspond à l'étiquette de $X[k]$, ie l'entier compris entre 0 et 9 représenté par $X[k]$.

Le fichier [IA_Chiffres_Eleves.py](#) est à votre disposition : il comporte des fonctions permettant d'afficher les chiffres, ainsi que le squelette des fonctions que vous aurez à développer

Apprentissage supervisé

On envisage l'algorithme des k plus proches voisins.

1. Écrire la **fonction `distance(im1, im2)`** qui renvoie la distance euclidienne entre les images `im1` et `im2`.
2. Écrire la **fonction `PlusProchesVoisins(X, Y, im, k)`** qui renvoie la ou les étiquettes les plus présentes parmi les k voisins de `im` (image de 8×8 pixels représentée sous la forme d'une liste).
3. Proposer des instructions pour **partager les données en 2 groupes** : un groupe pour l'apprentissage et l'autre pour le test.
4. Écrire des instructions pour calculer **le taux de bonnes prédictions** sur les données de tests.

Apprentissage non supervisé

On s'intéresse à nouveau à l'ensemble des images de chiffres de 0 à 9, mais cette fois-ci sans prendre en compte leur étiquette.

Nous allons appliquer l'algorithme des k moyennes avec $k = 10$, de manière à obtenir un regroupement de ces images en 10 classes, que l'on espère correspondre aux différents chiffres représentés. Pour rappel, ces images sont regroupées dans une liste `X`. Chaque image `X[i]` est représentée par un vecteur de \mathbb{R}^{64} .

5. Expliquer **l'algorithme des k -moyennes**.
6. Écrire une **fonction `barycentre(s)`** qui calcule le barycentre d'un ensemble d'images représenté par la liste `s`. Ce dernier est un vecteur de \mathbb{R}^{64} (il représente lui aussi une image 8×8).
7. Écrire la **fonction `kmoyennes(X,k)`** qui prend en argument une liste `X` d'images et un entier `k` et renvoie une liste de `k` classes où sont regroupées les images. Pour cela on choisit aléatoirement `k` centres initiaux `lesCentres`. Pour chaque image `Xi`, on calcule la valeur de `lesCentres` la plus proche de `Xi` pour ranger ce dernier dans la classe. On calcule les barycentres de ces classes. On s'arrête quand ces derniers ne sont pas modifiés.