IHDCM037 - Machine Learning
Travaux pratiques
Séance 4 : K-means

Remarque : Pour réaliser les travaux pratiques de ce cours, nous travaillerons avec le langage `Python` sur des `Jupyter Notebook`. Ceux-ci peuvent être utilisés en ouvrant la plateforme `Anaconda Navigator` et en lançant `Jupyter Notebook`.

Première partie

Cette séance est consacrée à l'algorithme de *clustering* **non supervisé** *K-means*. Le jeu de données à manipuler sera à nouveau le jeu de données `iris` mais il y aura une grande différence par rapport à la séance précédente car cette fois, les *targets* des données ne seront **pas** considérées. Voici les différentes étapes à suivre pour former des *clusters* à partir des données du *dataset* `iris` :

1. Importez le *dataset* `iris`.
2. Sauvegardez la partie *data* du *dataset* dans un tableau et extrayez uniquement les données concernant la longueur et la largeur des pétales des iris.
3. Générez un modèle *K-means* grâce à la fonction `KMeans` du module `cluster` de la librairie `sklearn`. Utilisez les paramètres `n_clusters = 2` et `init = 'random'`.
   - Qu'induit le paramètre `n_clusters = 2` ? .....
   - Qu'induit le paramètre `init = 'random'` ? .....

     .....
4. Entraînez ce modèle sur les données et sauvegardez le *label* (= numéro du *cluster*) prédit par le modèle pour chaque donnée.

   Remarque : Allez jeter un oeil à la documentation du modèle *K-means* pour déterminer quels sont les attributs de celui-ci qui pourraient vous aider.
5. Sauvegardez les coordonnées des centres des *clusters* dans un tableau.
   - Comment sont déterminés les centres des *clusters* ? .....

     .....

     .....
6. Affichez les données et les *clusters* grâce à la fonction `plot_clusters` fournie sur Webcampus. Cette fonction prend en argument les données, les *labels* prédits par l'algorithme *K-means*, ainsi que le tableau contenant les centres des *clusters* (par exemple : `plot_clusters(x,y,centers)`).
7. Calculez la valeur de la fonction objectif pour ce modèle.
   - Quelle est la formule de la fonction objectif ? .....
   - Quelle est la valeur de celle-ci pour ce modèle ? .....

Deuxième partie

Effectuez les mêmes étapes, mais fixez la valeur du paramètre `n_clusters` à 10 lors de la génération du modèle *K-means*.

- Quelle est la valeur de la fonction objectif pour ce modèle ? .....
- Comparez cette valeur avec celle du modèle précédent. Comment pouvez-vous expliquer une telle différence ? .....

  .....
- À votre avis, quel est le meilleur des deux modèles ? .....

Troisième partie

Tout comme pour l'apprentissage supervisé, les paramètres ont un rôle très important à jouer dans les modèles non supervisés de Machine Learning. Cependant, dans l'apprentissage non supervisé, il n'est pas possible de se baser sur des métriques pour savoir quelle valeur d'un paramètre rendra le modèle plus performant. Il est donc important de trouver d'autres alternatives pour déterminer la valeur "optimale" de ces paramètres.

Pour l'algorithme de *clustering* non supervisé *K-means*, il existe une méthode permettant de déterminer le nombre "optimal" de clusters : la méthode du coude. Pour appliquer cette méthode, générez des modèles *K-means* en faisant varier la valeur du paramètre `n_clusters` entre 1 et 10 (9 algorithmes de *clustering* seront donc générés et entraînés). Pour chaque modèle, calculez la valeur de la fonction objectif et affichez ces valeurs en fonction du nombre de *clusters* considérés. La valeur "optimale" se trouve dans le coude du graphique.

- Quelle est la valeur "optimale" pour le paramètre `n_clusters` ? .....