# Travail pratique 3

## Clustering… et KNN

En théorie, à ce point-ci, vous devriez déjà avoir fait un bout de chemin pour le partitionnement des données. Maintenant, nous allons utiliser l’algorithme du KNN pour inférer la catégorie grammaticale (nom, verbe, adjectif, …) des centroïdes! En d’autres mots, nous allons poser l’hypothèse suivante :

* Si les coordonnées d’un centroïde tombaient par hasard sur un mot, quelle serait sa catégorie grammaticale?

Tout juste avant l’affichage des résultats après l’itération finale du partitionnement, votre application devrait être en possession des distances de chaque mot avec le centroïde qui lui est le plus proche. Vous pouvez utiliser ces distances pour faire « voter » les « k » mots les plus proches du centroïde pour la catégorie grammaticale de ce mot.

Vous devez pouvoir afficher la catégorie grammaticale prédite de chaque centroïde ainsi que celle de chaque mot dans le cluster.

*Voir le fichier res\_4textes\_t5\_n5\_k5\_m10\_normaliser*

Comme vous pouvez le voir dans le nom du fichier, vous allez devoir laisser à l’utilisateur le choix, non seulement du nombre de voisins qui pourront voter, « k », mais aussi de pouvoir normaliser la matrice, tel que vu en classe.

L’appel pour produire le fichier ci-haut devra prendre l’allure suivante :

C:\> main.py -c -t 5 -n 5 -k 5 -m 10 --normaliser

Le KNN et la normalisation doivent être implémentées. À la ligne de commande, le KNN n’est pas optionnel, mais la normalisation oui.