Bilan Projet KMeans

**Sommaire :**

1. Présentation du projet et des objectifs
2. Déroulement du projet
3. Décisions importantes prises
4. Difficultés Rencontrés
5. Les + et les -

*1.Le projet*

L’objectif du projet était de réaliser l’algorithme du K Means à partir d’un fichier “.txt” où il fallait récupérer les données et de le représenter via un outil de visualisation de données

C’est un projet individuel et la durée pour le faire était d’environ 2 semaines

*2.Déroulement du projet*

J’ai choisi Intellij comme IDE pour développer car il me semblait le plus pratique et le plus complet, de plus il a son propre terminal intégré

Pour la visualisation du projet j’ai préfére Jfreechart aux autres suggestions données (plotly, Smile, …) car là encore je pense que cela est plus adapté aux objectifs du projet car il est possible via les librairies de représenter les données de beaucoup de façons différentes tout en implémentant une architecture objet (fonctionnel).

Ce qui m’amène aux codage, qui est orienté objet, le but étant de faire en sorte que le moins de données possible soit explicitements données par l’utilisateur et que si l’on souhaite rajouter de nouvelles fonctionnalités, il ne soit pas nécessaire de modifier le code déjà existant.

La première chose à faire après m’être assurer de bien avoir compris ce qui était demandé a été de faire un diagramme de classes (il y a plusieurs versions mais ici je ne parlerai que de 2 d’entres elles) tout d’abord j’ai crée un diagramme avec beaucoup de classes différentes (une pour Centroide, une autre pour Classes) mais le problème était que les classes était trop dépendante entres elles ce qui n’allait pas. La deuxième version est donc celle qui a pour but de relier les classes le moins possible entres elles (ce qui permet notamment de les tester séparément)

Après avoir fait le diagramme, j’ai commencé le codage, les classes ont été codées et testés dans l’ordre de dépendance croissant (de la moins dépendante à la plus dépendantes) c’est à dire : Données => Clusters => Matrice => Kmeans => … (les classes graphiques sont abordés plus tard)

(Les explications liés au codage dont pour la plupart mises directement en commentaires dans ce dernier pour faciliter la compréhension de l’utilisateur)

Pour plus d’explication sur le code, voir les commentaires des classes

*3.Décisions importantes importantes prises*

(Classés par ordre d’importance croissant)

* Dans le main je préfère lancer l’algo avec des valeurs par défauts plutôt que en demandant la saisie de l’utilisateur (mis en commentaire qui marche) pour montrer dans un premier temps le fonctionnement et ensuite permettre à l’utilisateur si il le souhaite d’activer la saisie (en enlevant les commentaires et les mettant sur”init”)
* Certaines données sont écrits en “brut” (exemple : this.donnée(i).get(4)) afin de faciliter le fonctionnement de l’algorithme
* Clusters ne contient pas les données (dans un premier temps j’avais mis un arraylist de données en paramètre) c’est le numéro du cluster qui est en paramètre
* Les méthodes ne doivent pas dépasser plus de 15 lignes, sauf pour les classes liés à l’interface graphique car l’architecture n’est pas totalement fini

*4. Difficultés rencontrés*

* Le fait de devoir utiliser des librairies java mais via le langage scala
* apprendre à utiliser Jfreechart pour implémentation graphique
* respecter une architecture objet

*5.Les + et les -*

(ce qui a était fait, ce qui n’a pas était fait et ce qu’il reste à faire)

les + :

* le fait d’avoir réussi à faire fonctionner le K Means et à l’implémenter graphiquement
* le codage est majoritairement orienté objet et peut donc être complété assez aisément
* les classes ne sont pas trop dépendantes chacune les unes des autres et ne sont pas trop longues

les - :

* il pourrait être judicieux de définir une condition d’arrêt qui calculerait le delta entre les anciens et nouveaux centres
* certaines données sont écrit en brute ce qui fait que si on choisit un différend fichier texte de base alors l’algo ne fonctionnera plus à part selon certaines conditions très précises (voir notice)
* il devrait pouvoir être possible de relier les couleurs de chacun des graphiques entre elles

Fait par Pierre-Louis NOHET, Groupe POUFSOUFFLE