SAE Python

0. Expliquer brièvement l'objectif de Knn et K Means (ce que l'on souhaite faire et pourquoi).

Knn est un algorithme d'apprentissage automatique supervisé. Il sert à prédire la valeur d'un point en fonction des *k* points les plus proches dans l'espace des caractéristiques. Il repose sur l'hypothèse que des objets similaires sont proches les uns des autres. Une fois ceci fait, on peut chercher à en faire plusieurs choses : une classification ou une régression. Une classification, c'est chercher à mettre les données en paquet, à les regrouper. Une régression, c'est chercher une droite dans le nuage de données, un lien dans le groupe de données que l'on a. Et knn peut servir à faire ces deux choses.

Pour ce qui est de k-means, il sert à trier des points qui ont des similarités majeures en groupes, appelés cluster. Ces clusters sont caractérisés par des centroïdes. Les clusters ont eux des différences notables entre eux.

1. Décrire les algorithmes Knn et K Means, le type de données à traiter, les structures de ces données.

1.1 - KNN

Nous traitons des données de type tuple, c'est-à- dire que nous avons une étiquette et une information, qui forme une donnée. Et ces données sont structurées en liste.
En bref, l'algorithme pars d'un nuage de points et calcule les plus proches voisins de ces points. Pour la mise en oeuvre, on donne en entrée les données, le nombre de plus proches voisins qu'on calcule et nomme k, un nombre p positif et non nul qui servira de mode, celui ci déterminera quelle méthode sera utilisée afin de calculer les distances entre les distances de Manhattan, Euclidienne,vet de Chebyshev, et enfin on choisira quel type de résultat on voudra entre une régression et une classification.

Pour le choix entre les calculs de distances, le choix s'avère complexe. En effet, de ce que nous avons compris, la distance euclidienne permet d'avoir la distance en ligne droite avec beaucoup de précision. La distance de manhattan, elle, est moins coûteuse que la première, et est plus performante pour des mouvements en "grille" (des déplacements horizontaux et verticaux a la suite). La distance de chebyshev , moins coûteuse que la première également, est efficace lorsque l'on veut calculer avec des diagonales. Et pour la dernière

distance, compromis entre performance et précision, regroupe les trois autres calculs de distances, en fonction de la distance. Elle permet donc de la flexibilité.
Pour réaliser cet algorithme nous l'avons divisé en deux fonctions: une pour le calcul de distances et une pour faire la régression et la classification. Dans la première partie, le calcul des distance des points est fait, en fonction du mode que l'on a choisi en entrée, donc en fonction de p. Le calcul des distance se fait en appelant une fonction avec les différents calculs de distance possible, dont on met en entrée deux listes de points, et p, pour choisir le mode de calcul.

Dans la deuxième partie, en fonction de si nous avons choisis de faire une régression ou une classification, le calcul est fait afin de renvoyer la chose désirée.

1.2 - K-Means

Pour ce qui est de K-means, nous utilisons une liste de points, ainsi que un entier k qui sera le nombre de centroïdes
Pour l'algorithme, nous prenons un ensemble de points de dimensions quelconque, ainsi qu'un nombre de points k a choisir. Ces points seront des "centroïdes". Ils seront les centres de leur groupe de points. Ensuite, pour les autres points, on les affilient au centroïde le plus proche, en les mettant dans des clusters, des groupes de points. Pour chaque point, on les compare à tous les centroïdes et on met le point dans le cluster qui possède le centroïde le plus proche. Après, pour chaque cluster, on calcul le point moyen (la moyenne des points), et le point du cluster qui se rapproche le plus de la moyenne, devient le nouveau centroïde du cluster. Puis on réaffecte tous les autres points au centroïdes le plus proches (dans le bon cluster).

Pour ce qui est des cas d'arrêts du programme, il y en a 2 :

- Soit la liste des centroïdes converge, c'est-à-dire que les points des centroïdes stagnent.
- Soit, on a atteint le maximum d'itérations

Plus précisément, nous faisons la sauvegarde des centroïdes, et nous la comparons avec la liste des nouveaux centroïdes. Si les nouveaux centroïdes sont identiques aux anciens, alors nous incrémentons la variable "convergence".
Donc, pour que l'algorithme s'arrête, il faut que soit la variable "convergence" soit supérieur ou égal à 3 (aucun changement de centroïdes 3 fois) ou, qu'il y a eu 100 000 modifications de centroïdes et de clusters (c'est une valeur choisie arbitrairement).

2. Les limitations ou problèmes que vous avez rencontrés, les recherches (explorations) effectuées pour résoudre ces problèmes (ou les contourner).

Pour ce qui est des limitations et des problèmes rencontrés, nous allons voir d'abord ceux concernant d'abord Knn. Pour KNN, nous n'avions aucune idée de ce que régression et classification voulaient dire, nous avons donc dû faire des recherches dessus. Et donc nous ne savions pas comment faire le choix entre les deux lors du programme, donc nous avons décidé que ce serait une variable rentrée au moment du lancement. C'est le seul gros problème que nous avons rencontré, qui était donc un problème de vocabulaire technique.

Pour ce qui est de k-means, le plus gros souci c'était la compréhension de l'algorithme et de devoir ensuite trouver une solution pour le reproduire. Nous n'avions pas compris le concept de cluster et de centroïdes, nous poussant donc à devoir faire des recherches plus précises dessus. Et en dernier point, le fait de devoir faire en sorte que le programme s'adapte au nombre de dimensions des points choisie nous a aussi posé un peux problème, mais nous avons finalement trouvé la solution en cherchant.

Et plus globalement, concernant les difficultés, pour un des membres du groupe, un manque global de connaissance de python, qui a entrainer un des membre du groupe à faire beaucoup de recherches, à la place de se concentrer sur les algorithmes en eux-même. Et sinon, si nous avions des questions concernant les consignes qui parfois nous semblaient un peu floue, nous demandions à l'enseignant.

Mais en général, ce travail, notamment la dimension de recherche en autonomie, nous a beaucoup apporté, que ce soit en connaissance de python pure, ou en compréhension de l'algorithme. Enfin, nous avons aussi appris en explication d'algorithme, ce que nous n'avions pas l'habitude de faire, notamment sur notre propre travail.