

# Rapport de projet

## Exploration algorithmique d'un problème

### SAE2.02

#### Objectif du projet

Le but du projet est de comprendre et reproduire l'algorithme d'apprentissage automatique KNN et Kmeans qui s'utilisent pour résoudre des problèmes de classification et de régression.

#### Contexte du projet

Ces algorithmes utilisent des données de type donnée-étiquetée, au départ ils possèdent plusieurs données et lorsqu'une nouvelle donnée non-étiquetée arrive on recherche les voisins les plus proches de cette nouvelle donnée en fonction des caractéristiques pour lui attribuer une étiquette.

#### Type des données choisis

Pour ce projet, j'ai décidé de travailler sur des points associés à un plan. Chaque point est une instance de la classe Point avec une coordonnée d'abscisse et une coordonnée d'ordonnée sur un plan. Ce point est associé à un nom, un groupe et une couleur qui représentent un objet.

J'ai créé pour ça des classes en Python pour mieux manipuler les données en les regroupant en tant qu'attributs d'un objet.

L'utilisateur peut manipuler une liste de points qu'il peut créer et déplacer sur un plan pour utiliser les algorithmes.

#### Points à améliorer

Il y a énormément de points qui pourraient être à améliorer sur le programme :

- Offrir plus d'options / possibilités à l'utilisateur pour manipuler les données,
- Améliorer l'interface sur le terminal en optimisant l'affichage,
- Implémentation de test dans un fichier externe avec des assertions,
- Kmeans pourrait itérer jusqu'à ce qu'il n'y ait plus de changement entre les différentes itérations,
- Les données choisies pourraient avoir de nouvelles données à travailler (ex : taille, poids... ),
- Ajouter plus de sécurité au programme.

## Programme Principal

Le programme propose différentes options à l'utilisateur :

- lister les options du programme,
- initialiser le plan avec des points aléatoires,
- afficher les objets sur le plan,
- effacer le plan,
- créer un point en particulier et le mettre dans la liste des points  
le programme demande à l'utilisateur des coordonnées et un nom,
- créer plusieurs points et les mettre dans la liste des points  
le programme demande un nombre de point à créer et un rayon de génération autour de l'origine ( ex : créer 10 points entre -10 et 10 en x et -10 et 10 en y ),
- afficher la liste des points et leurs attributs,
- utiliser l'algorithme KNN sur un point en particulier de la liste  
le programme demande l'indice du point et le nombre d'instances à retenir,
- utiliser l'algorithme KNN sur plusieurs points de la liste  
le programme demande l'indice du point de départ et du point d'arrivée et le nombre d'instances à retenir,
- utiliser l'algorithme KNN sur tous les points de la liste  
le programme demande le nombre d'instances à retenir,
- utiliser l'algorithme Kmeans sur le plan  
le programme demande le nombre d'instances à retenir, le nombre d'itérations à faire et le rayon de génération des centres autour de l'origine,
- ajouter un point de la liste sur le plan,
- ajouter plusieurs points de la liste sur le plan,
- ajouter tous les points de la liste sur le plan,
- Quitter le programme.

## Déroulement des fonctions

Seule les fonctions les plus importantes seront expliquée et déroulée.

### Fonction KNN

La fonction KNN prend en paramètre un objet non-étiquetée, une liste de données sur laquelle l'algorithme va travailler et un entier k qui est le nombre d'instance à retenir.

La fonction commence d'abord à faire une collection de toutes les distances puis trie cette collection dans l'ordre croissant. L'algorithme garde ensuite les k premiers points les plus proches et retient l'étiquette la plus courante chez ces k points. Une fois que l'algorithme à trouvé l'étiquette, il la place sur l'objet.

### Fonction Kmeans

La fonction Kmeans prend en paramètre une liste d'objets, un nombre d'instances à retenir, un nombre d'itération et un rayon de génération des centres.

La fonction commence par générer des centres temporaires sur le plan. Pour un certain nombre d'itération et pour chaque point sur le plan, l'algorithme regarde les centres les plus proches du point ajoute le point au groupe du centre. Ensuite l'algorithme calcule les nouveaux centres avec la moyenne des coordonnées des points de leurs groupes.

### Fonction distance

La fonction calcul une distance euclidienne entre deux points par rapport à leurs coordonnées sur un plan et retourne la valeur sous la forme d'un réel.