TP n=°5 : K moyennes

Informatique tronc commun 2ère année

1) Exécuter l'instruction suivante `afficher_liste_points(li)`. La liste points est une liste de couples de flottants, et c'est cette liste qu'on va essayer de couper en différentes classe.

2) Écrire une fonction `barycentre(liste_p)` qui prend en entrée une liste de couples de flottants nommé `liste_p` et qui renvoie un couple correspondant au barycentre des points de `liste_p`.

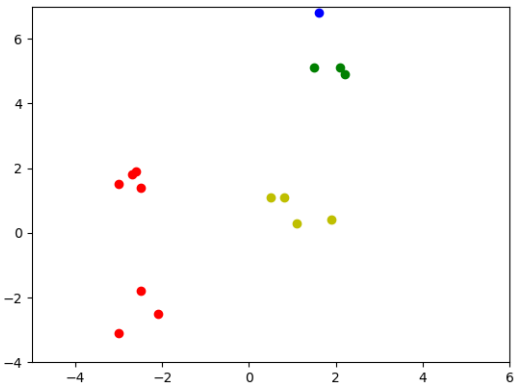
3) Écrire une fonction `barycentre_classe(liste_classe` qui prend en entrée une liste de liste de couples de flottants nommé `liste_classe` et renvoie la liste des barycentres de chaque sous-listes de `liste_classe`.

4) Écrire une fonction `distance_euclidienne(p1,p2)` qui prend deux couples `p1` et `p2` et renvoie la distance euclidienne entre `p1` et `p2`.

5) Écrire une fonction `plus_proche_indice(liste_centre,point)` qui prend en entrée une liste de points `liste_centre` et un point `point`. La fonction renvoie l'indice du centre le plus proche du point donné en entrée.
Par exemple `plus_proche_indice([(1.0,0.0),(1.5,2.0),(-1.0,0.5),(2.0,-1.0)],(-1.2,0.7))` renvoie 2 car le point (-1.2,0.7) est le plus proche de (-1.0,0.5) en position 2.

6) Observer la boucle while dans la fonction préécrite `k_moyenne(liste,k)`. Quelle est le risque d'écrire `while True`? Grâce à quelle instruction les appels à la fonction `k_moyenne(liste,k)` vont terminer?
Le danger d'avoir `while` True est d'écrire une boucle qui ne termine pas.

7) Faites un premier appel à la fonction `k_moyenne` avec la liste `l` et `k=4`. Est-ce que l'appel est satisfaisant?



La répartition des classes ainsi obtenue est à gauche. On peut voir qu'il y a quatre grands groupes de points. Cependant, la répartition par couleurs ne distingue pas ces quatre groupes de points. Les points rouges sont sur 2 grands groupes et les points bleus et verts se partagent un groupe.

8) Modifier la manière d'initialiser la liste `liste_centre` dans la fonction `k_moyenne(liste,k)` pour utiliser la fonction `choix_aléatoire(liste,k)`.

9) Décommenter la ligne `liste_centre = [(1.1,0.3),(1.6,6.8),(-2.7,1.8),(0.8,1.1)]`, et exécuter un appel de `k_moyenne(l,k)` avec k = 4. Que se passe t-il? Pourquoi?
On obtient une division par zéro, car il y a un barycentre à un moment de l'exécution qui n'a aucun point le plus proche. Une classe se retrouve alors vide, et en fonction de où on a mis la division dans la fonction `barycentre(liste_p)`, on divise par 0.

10) Proposer une correction de la fonction `k_moyenne(liste,k)` et/ou des sous fonctions qui composent `k_moyenne(liste,k)` pour prendre en compte ce problème.

11) Écrire une fonction `eps_moyen(classe)` qui prend en entrée une liste de points `classe` et calcule la distance moyenne d'écart entre chaque point de la classe et le barycentre de la classe.

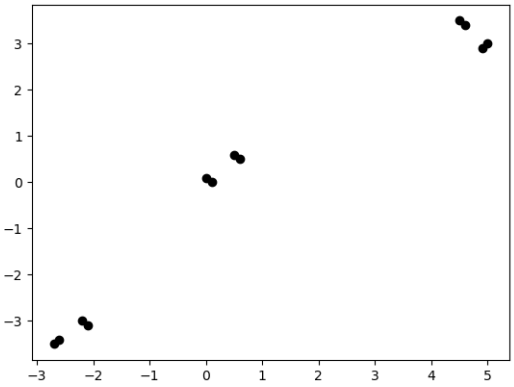
12) Écrire une fonction `eps_max(cls)` qui prend en entrée `cls` une liste de classe et renvoie la valeur du maximum de la fonction `eps_moyen(cl)` sur les classes `cl` de la liste `cls`.

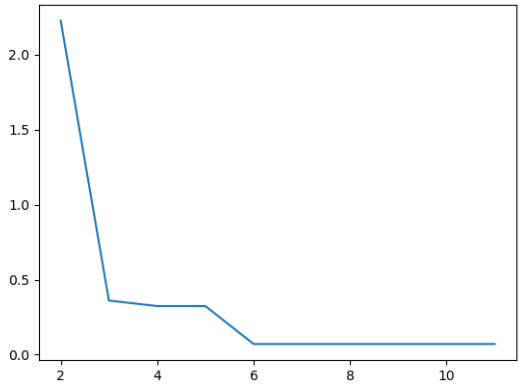
13) En utilisant la fonction `k_moyenne_sans_aff(liste,k)` qui est l'implémentation de l'algorithme des k-moyennes sans affichage, écrire une fonction `k_moyenne_multiple(liste,k,nb)` qui est l'implémentation de l'algorithme présenté précédemment.

14) Écrire une fonction `tracer_k(liste,nb,max_k)` prenant en entrée une liste `liste` de points et deux entiers `nb,max_k` et trace l'erreur moyenne renvoyée par `k_moyenne_multiple(liste,k,nb)` en fonction de `k` pour `k` variant de 2 à `max_k`.

15) Quelle valeur de k voudrait-on choisir pour la liste `li` ? Quelle critère peut-on choisir sur le graphique tracé par l'appel de `tracer_k(li,50,10` pour trouver ce k?

16) Quelle valeur de k voudrait-on choisir pour la liste `li` ? Quelle critère peut-on choisir sur le graphique tracé par l'appel de `tracer_k(li,50,10` pour trouver ce k?





Pour trouver la forme du graphe de droite, il faut repérer sur la figure de gauche les points suivants :

- Les points forment trois grands groupes → On s'attend à une forte chute de l'erreur quand k = 3.
- Chaque grand groupe peut être vu comme deux groupes de deux points. Ainsi, on aurait 6 classes différentes au total si on veut être plus fin → On s'attend à une chute plus modéré de l'erreur pour k = 6.

17) Écrire une fonction `tracer_nb(liste,nb_max,k,rep)` prenant en entrée une liste `liste` de points et trois entiers `nb_max,k,rep` et trace l'erreur moyenne renvoyée par rep appels de `k_moyenne_multiple(liste,k,nb)` en fonction de `nb` pour `nb` variant de 2 à `max_nb`.