

k-moyennes fleuries

Marquer comme terminé

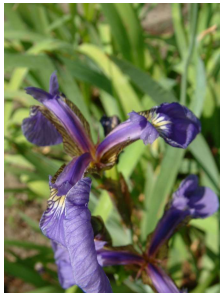
Ouvert le : lundi 23 janvier 2023, 00:00

À remettre : lundi 22 janvier 2024, 16:30

Il s'agit d'implémenter l'algorithme des k-moyennes à une base de données classique en apprentissage automatique : les iris. La base de données des iris est présentée ci-dessous, puis le travail est décrit. Enfin, l'algorithme est rappelé.

Les iris (150 données)

Classes :



Iris setosa



Iris versicolor



Iris virginica

Attributs :

1. Longueur des sépales
2. Largeur des sépales
3. Longueur des pétales
4. Largeur des pétales

SÉPALE, subst. masc. (<https://cnrtl.fr/definition/sépale>)

BOT. Chacun des éléments foliacés, généralement verts, dont la réunion compose le calice et supporte la corolle de la fleur.

PÉTALE, subst. masc. (<https://cnrtl.fr/definition/pétale>)

A. – BOT. Chacun des éléments foliacés, généralement colorés, qui composent la corolle d'une fleur.

Répartition de la population, selon les attributs fournis :



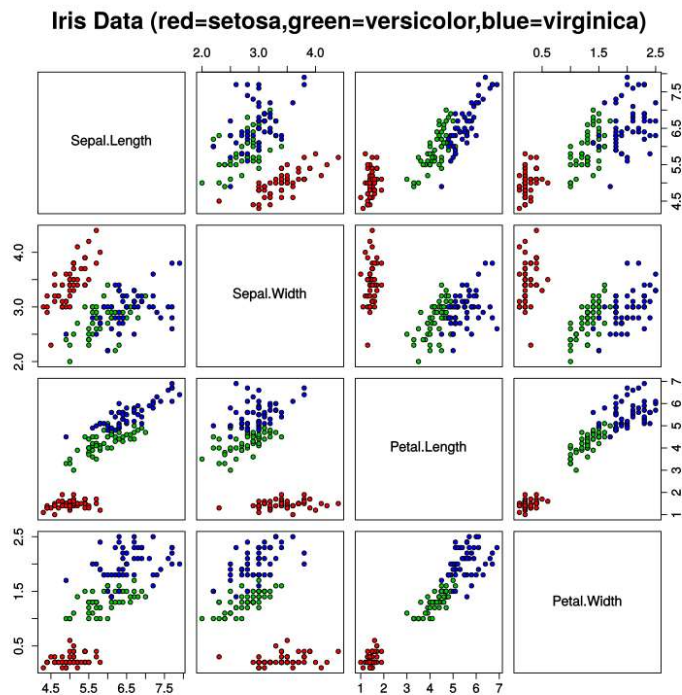


image tirée de https://en.wikipedia.org/wiki/Iris_flower_data_set

k-moyennes et iris

Le programme devra :

1 - Prendre en entrée deux fichiers, inclus dans l'archive fournie :

```
- iris_data
- iris_label
```

Le fichier `iris_data` est composé d'une donnée par ligne sous la forme :

```
5.0;3.3;1.4;0.2;
```

```
7.0;3.2;4.7;1.4;
```

```
6.5;3.2;5.1;2.0;
```

Le fichier `label` a le même nombre de lignes que le fichier `data`, chaque ligne indiquant la classe de la ligne correspondante dans le fichier `data`. Avec :

- 0 désigne la classe des iris setosa
- 1 désigne la classe des iris versicolor
- 2 désigne la classe des iris virginica

2 - Programmer, en python, l'algorithme des k-moyennes, idéalement de manière à pouvoir l'appliquer à différents jeux de données.

3 - Appliquer l'algorithme aux iris, en faisant varier la valeur de k (i.e. le nombre de clusters) et donnez le k minimal permettant d'obtenir des clusters contenant un unique type d'iris. Les classes (fichier `label`) seront utilisées pour vérifier la "pureté" des clusters.

4 - Tester son implémentation sur les clusters gaussiens fournis dans le fichier `clusters.py` joint (sans utiliser les centres comme centroïdes). Comparer les résultats pour 2 à 6 clusters, que vous pourrez éloigner, rapprocher ou étendre (en taille).

Rappel de l'algorithme des k-moyennes

INIT choisir alétoirement k points comme centres

1.

pour chaque point, affecter au cluster dont le centre est le plus proche
2.

pour chaque cluster C , calculer son nouveau centre m
$$m = \frac{1}{|C|} \sum_{x \in C} x$$
3.

tant que affectations changent, itérer 1.

Le déroulement de l'algorithme est sur le principe suivant :

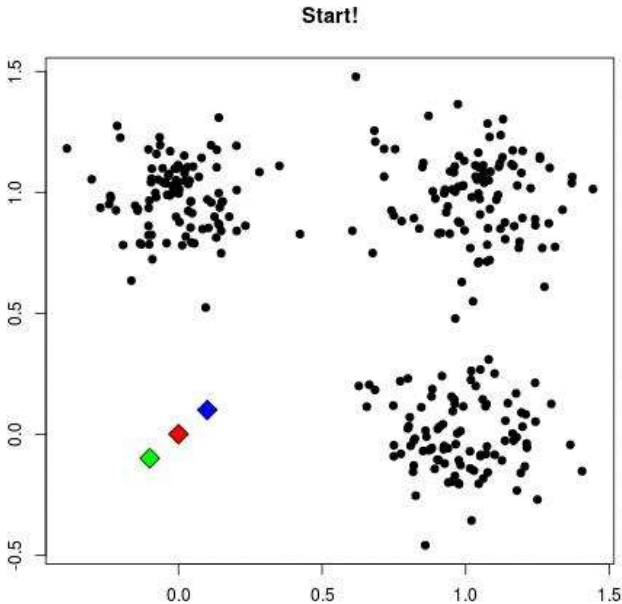


image tirée de <https://mubaris.com/posts/kmeans-clustering/>

clusters.py

iris.zip

25 janvier 2023, 16:43

23 janvier 2023, 11:24

Ajouter un travail

Statut de remise

Numéro de tentative	Ceci est la tentative 1.
Statut des travaux remis	Aucun devoir n'a encore été remis
Statut de l'évaluation	Non évalué
Temps restant	2 heures 40 min restants

Contactez-nous



Suivez-nous



Obtenir l'app mobile

Université Paris 8 - Vincennes - Saint-Denis
2 rue de la Liberté - 93526 Saint-Denis cedex

[Charte de l'université Paris 8](#) / [CGU - Données personnelles - Mentions légales](#)