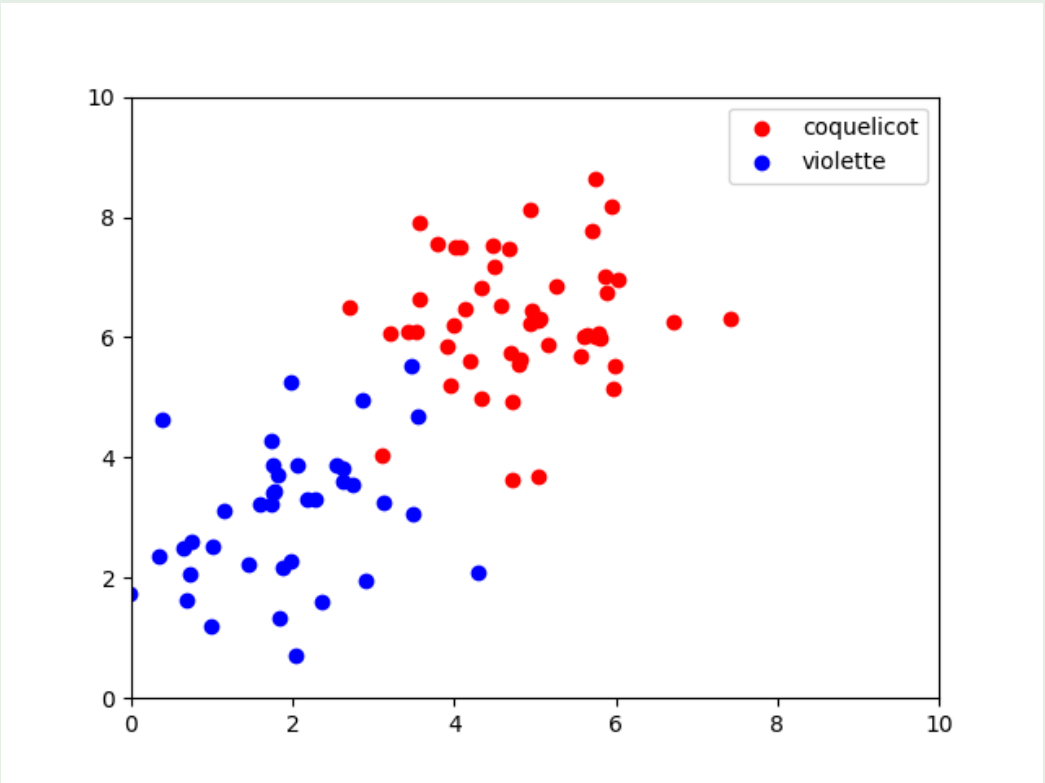
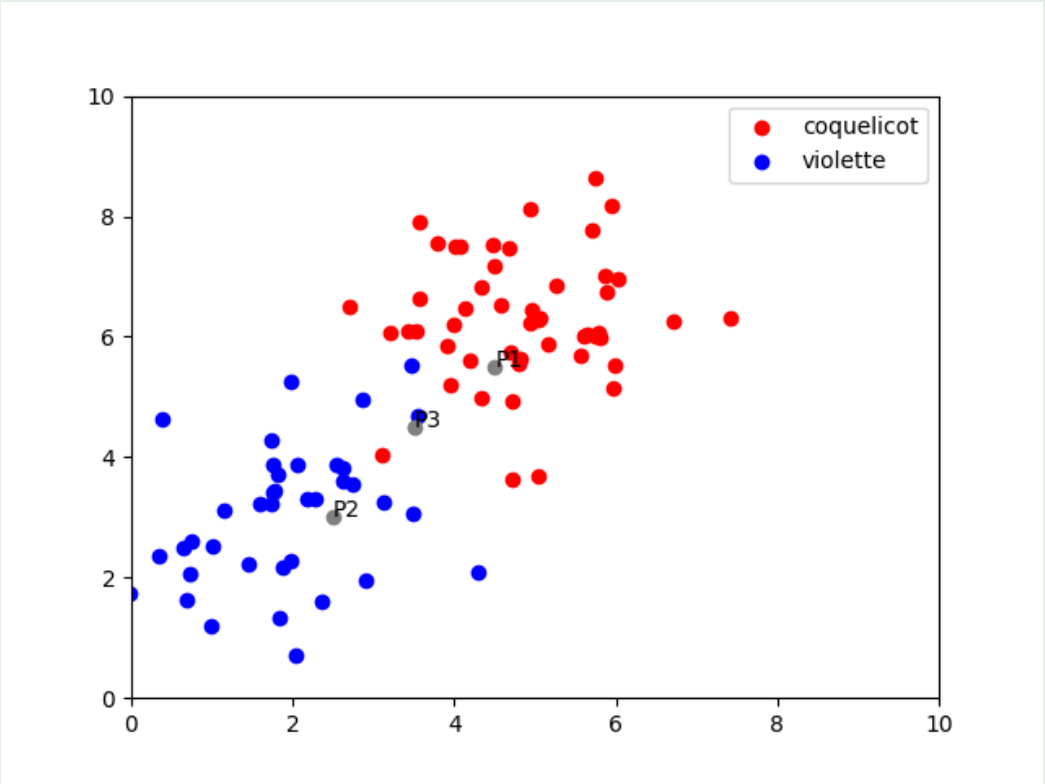
Un champ de fleurs

Dans un champ, à l'état sauvage deux types de fleurs ont poussés : des coquelicots et des violettes. On a représenté ci-dessous par un schéma la position de ces fleurs dans le champ



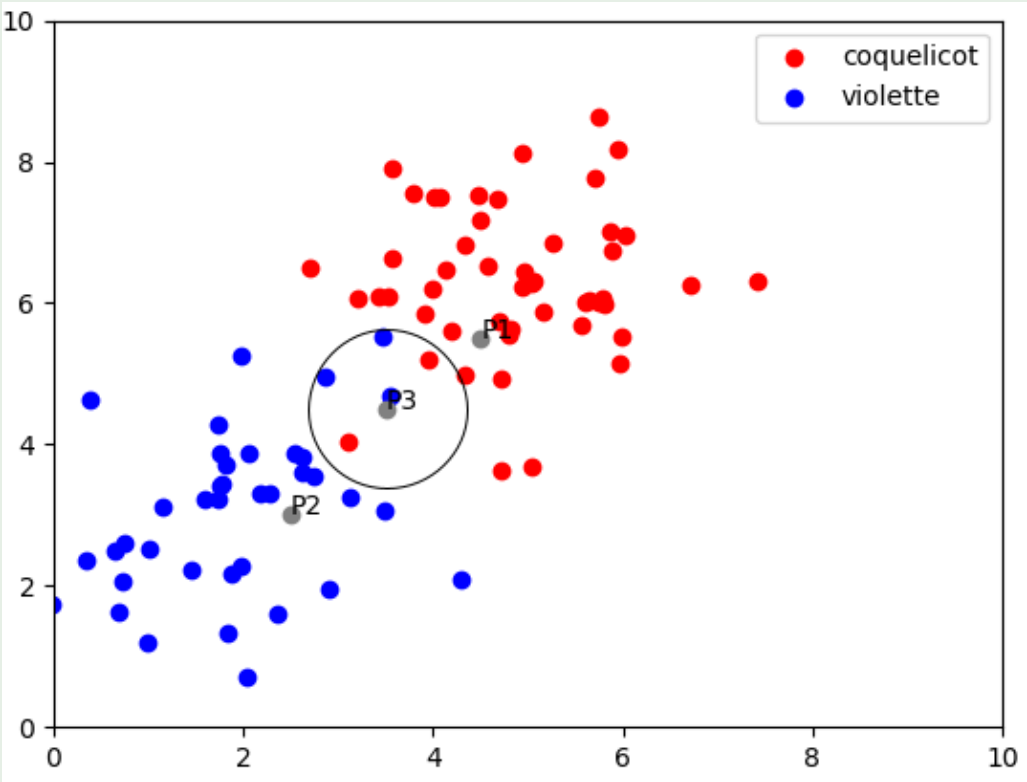
Un champ de fleurs

Trois nouvelles pousses, notées P_1, P_2 et P_3 (en gris sur le schéma) font leur apparition. Et on cherche à prédire si ces pousses sont des coquelicots ou des violettes.



Un champ de fleurs

On a tracé ci-dessous un cercle de façon apparaître les 5 voisins les plus proches de P_3. Choisir l'espèce majoritaire de ce cercle pour classer la nouvelle pousse P_3 est un exemple de l'application des 5 plus proches voisins (*nearest neighbours* en anglais, abrégé en *nn*)



Principe de l'algorithme

- L'algorithme des k plus proches voisins est un algorithme de classification des données appartenant à la famille des algorithmes d'apprentissage *supervisé*.

Principe de l'algorithme

- L'algorithme des k plus proches voisins est un algorithme de classification des données appartenant à la famille des algorithmes d'apprentissage *supervisé*.
- On dispose d'un jeu de données qui associe chaque donnée à une classe.

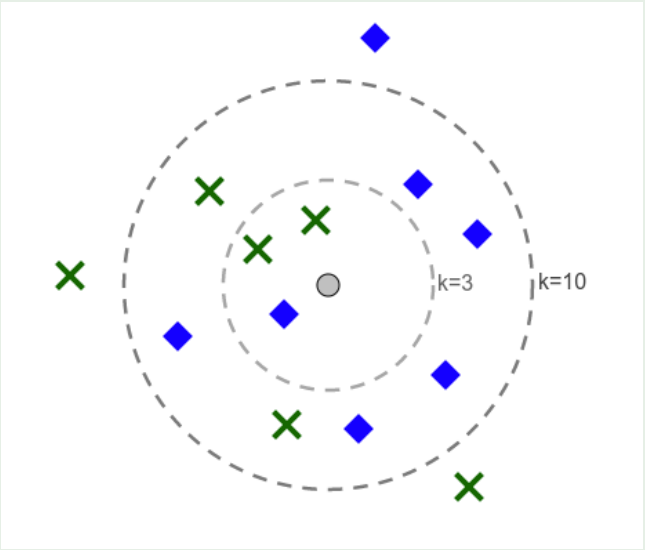
Principe de l'algorithme

- L'algorithme des k plus proches voisins est un algorithme de classification des données appartenant à la famille des algorithmes d'apprentissage *supervisé*.
- On dispose d'un jeu de données qui associe chaque donnée à une classe.
- L'algorithme attribut à une nouvelle donnée d non classée la classe majoritaire de ses k plus proches voisins.

Principe de l'algorithme

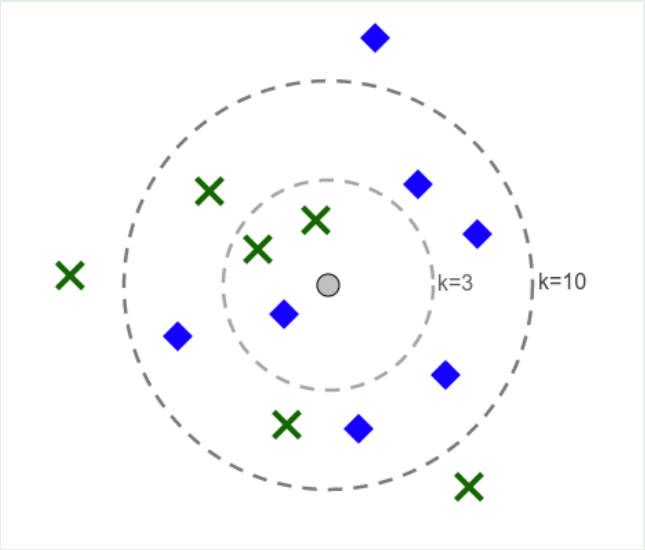
- L'algorithme des k plus proches voisins est un algorithme de classification des données appartenant à la famille des algorithmes d'apprentissage *supervisé*.
- On dispose d'un jeu de données qui associe chaque donnée à une classe.
- L'algorithme attribut à une nouvelle donnée d non classée la classe majoritaire de ses k plus proches voisins.
- On doit donc utiliser une distance sur l'ensemble des données (par exemple la distance euclidienne)

Exemple



Le point gris central est la donnée à classer. Quel sera le résultat de l'algorithme :

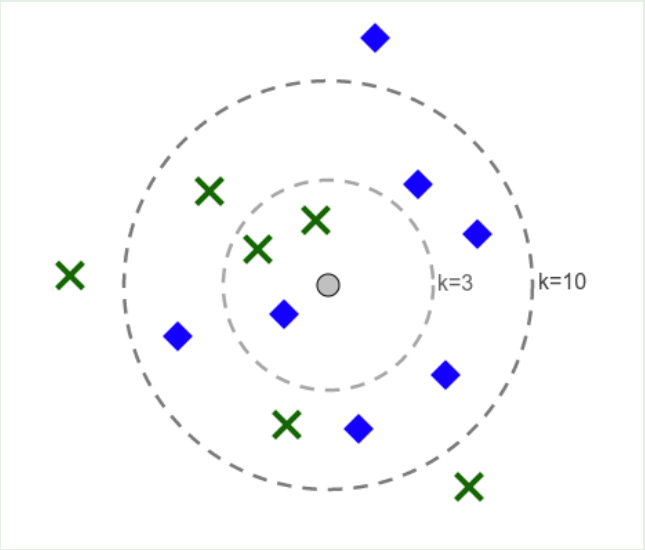
Exemple



Le point gris central est la donnée à classer. Quel sera le résultat de l'algorithme :

- Pour $k = 3$?

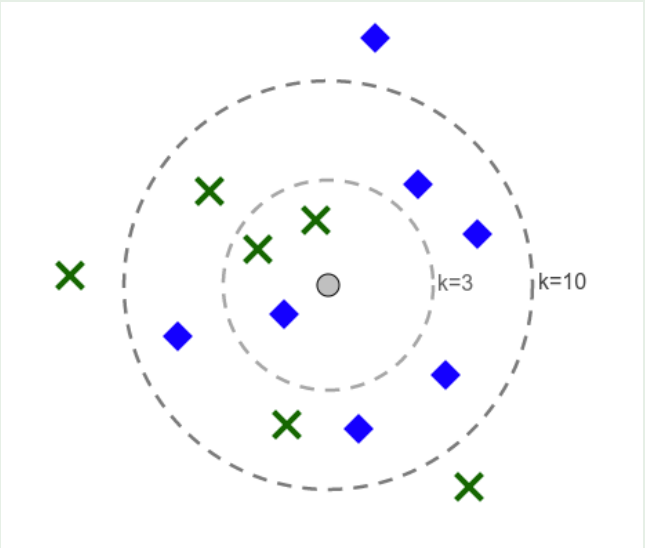
Exemple



Le point gris central est la donnée à classer. Quel sera le résultat de l'algorithme :

- Pour $k = 3$?
- Pour $k = 10$?

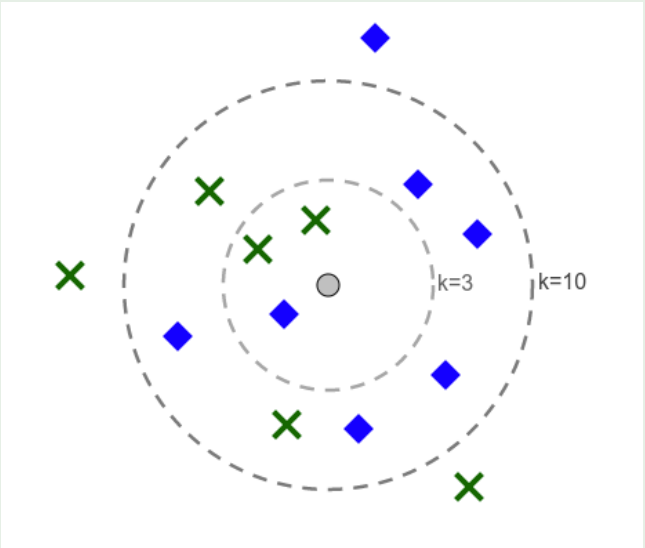
Exemple



Le point gris central est la donnée à classer. Quel sera le résultat de l'algorithme :

- Pour $k = 3$? Il y 2 croix et un losange dans les 3 plus prochains voisins, la classe majoritaire est donc la croix et l'algorithme classe la donnée comme une croix.
- Pour $k = 10$?

Exemple



Le point gris central est la donnée à classer. Quel sera le résultat de l'algorithme :

- Pour $k = 3$? Il y 2 croix et un losange dans les 3 plus prochains voisins, la classe majoritaire est donc la croix et l'algorithme classe la donnée comme une croix.
- Pour $k = 10$? Cette fois il y a 6 losanges et 4 croix parmi les 10 plus proches voisins, la donnée est donc classée parmi les losanges.

Test d'efficacité

- Pour tester l'efficacité de l'algorithme, on réserve une partie des données déjà classées afin d'effectuer des tests.

Test d'efficacité

- Pour tester l'efficacité de l'algorithme, on réserve une partie des données déjà classées afin d'effectuer des tests.
- Sur ces données, on peut comparer la classe effective de la donnée avec celle fournie par l'algorithme

Test d'efficacité

- Pour tester l'efficacité de l'algorithme, on réserve une partie des données déjà classées afin d'effectuer des tests.
- Sur ces données, on peut comparer la classe effective de la donnée avec celle fournie par l'algorithme
- En particulier, le calcul de la matrice de confusion, dont le coefficient M_{ij} est le nombre de données de la classe i affectés à la classe j permet de mesurer le taux d'erreur de l'algorithme.

Test d'efficacité

- Pour tester l'efficacité de l'algorithme, on réserve une partie des données déjà classées afin d'effectuer des tests.
- Sur ces données, on peut comparer la classe effective de la donnée avec celle fournie par l'algorithme
- En particulier, le calcul de la matrice de confusion, dont le coefficient M_{ij} est le nombre de données de la classe i affectés à la classe j permet de mesurer le taux d'erreur de l'algorithme.
- Ces tests permettent de déterminer une « bonne » valeur de k

Exemple

En utilisant l'algorithme des k plus proches voisins pour classer des données tests on a obtenu les matrices de confusion suivantes :

pour $k = 4$

$$\begin{pmatrix} 9 & 1 & 0 \\ 2 & 10 & 3 \\ 1 & 2 & 12 \end{pmatrix}$$

pour $k = 7$

$$\begin{pmatrix} 10 & 0 & 0 \\ 1 & 11 & 1 \\ 0 & 1 & 14 \end{pmatrix}$$

Exemple

En utilisant l'algorithme des k plus proches voisins pour classer des données tests on a obtenu les matrices de confusion suivantes :

pour $k = 4$

$$\begin{pmatrix} 9 & 1 & 0 \\ 2 & 10 & 3 \\ 1 & 2 & 12 \end{pmatrix}$$

pour $k = 7$

$$\begin{pmatrix} 10 & 0 & 0 \\ 1 & 11 & 1 \\ 0 & 1 & 14 \end{pmatrix}$$

- Donner le nombre de classes, et le nombre de données tests par classe.

Exemple

En utilisant l'algorithme des k plus proches voisins pour classer des données tests on a obtenu les matrices de confusion suivantes :

pour $k = 4$

$$\begin{pmatrix} 9 & 1 & 0 \\ 2 & 10 & 3 \\ 1 & 2 & 12 \end{pmatrix}$$

pour $k = 7$

$$\begin{pmatrix} 10 & 0 & 0 \\ 1 & 11 & 1 \\ 0 & 1 & 14 \end{pmatrix}$$

- Donner le nombre de classes, et le nombre de données tests par classe.

Exemple

En utilisant l'algorithme des k plus proches voisins pour classer des données tests on a obtenu les matrices de confusion suivantes :

pour $k = 4$

$$\begin{pmatrix} 9 & 1 & 0 \\ 2 & 10 & 3 \\ 1 & 2 & 12 \end{pmatrix}$$

pour $k = 7$

$$\begin{pmatrix} 10 & 0 & 0 \\ 1 & 11 & 1 \\ 0 & 1 & 14 \end{pmatrix}$$

- Donner le nombre de classes, et le nombre de données tests par classe.
  Il y a 3 classes, 10 données sont dans la première classe, et 15 dans chacune des deux autres.

Exemple

En utilisant l'algorithme des k plus proches voisins pour classer des données tests on a obtenu les matrices de confusion suivantes :

pour $k = 4$

$$\begin{pmatrix} 9 & 1 & 0 \\ 2 & 10 & 3 \\ 1 & 2 & 12 \end{pmatrix}$$

pour $k = 7$

$$\begin{pmatrix} 10 & 0 & 0 \\ 1 & 11 & 1 \\ 0 & 1 & 14 \end{pmatrix}$$

- Donner le nombre de classes, et le nombre de données tests par classe.
  Il y a 3 classes, 10 données sont dans la première classe, et 15 dans chacune des deux autres.
- Calculer le pourcentage d'erreur commis pour chacune des valeurs de k.
  Conclure

Exemple

En utilisant l'algorithme des k plus proches voisins pour classer des données tests on a obtenu les matrices de confusion suivantes :

pour $k = 4$

$$\begin{pmatrix} 9 & 1 & 0 \\ 2 & 10 & 3 \\ 1 & 2 & 12 \end{pmatrix}$$

pour $k = 7$

$$\begin{pmatrix} 10 & 0 & 0 \\ 1 & 11 & 1 \\ 0 & 1 & 14 \end{pmatrix}$$

- Donner le nombre de classes, et le nombre de données tests par classe.
  Il y a 3 classes, 10 données sont dans la première classe, et 15 dans chacune des deux autres.
- Calculer le pourcentage d'erreur commis pour chacune des valeurs de k.
  Conclure

Exemple

En utilisant l'algorithme des k plus proches voisins pour classer des données tests on a obtenu les matrices de confusion suivantes :

pour $k = 4$

$$\begin{pmatrix} 9 & 1 & 0 \\ 2 & 10 & 3 \\ 1 & 2 & 12 \end{pmatrix}$$

pour $k = 7$

$$\begin{pmatrix} 10 & 0 & 0 \\ 1 & 11 & 1 \\ 0 & 1 & 14 \end{pmatrix}$$

- Donner le nombre de classes, et le nombre de données tests par classe.
  Il y a 3 classes, 10 données sont dans la première classe, et 15 dans chacune des deux autres.
- Calculer le pourcentage d'erreur commis pour chacune des valeurs de k. Conclure
  - Pour $k = 4$, le pourcentage d'erreur est $9/40 = 22,5\%$.

Exemple

En utilisant l'algorithme des k plus proches voisins pour classer des données tests on a obtenu les matrices de confusion suivantes :

pour $k = 4$

$$\begin{pmatrix} 9 & 1 & 0 \\ 2 & 10 & 3 \\ 1 & 2 & 12 \end{pmatrix}$$

pour $k = 7$

$$\begin{pmatrix} 10 & 0 & 0 \\ 1 & 11 & 1 \\ 0 & 1 & 14 \end{pmatrix}$$

- Donner le nombre de classes, et le nombre de données tests par classe.
  Il y a 3 classes, 10 données sont dans la première classe, et 15 dans chacune des deux autres.
- Calculer le pourcentage d'erreur commis pour chacune des valeurs de k. Conclure
  - Pour $k = 4$, le pourcentage d'erreur est $9/40 = 22,5\%$.
  - Pour $k = 7$, le pourcentage d'erreur est $3/40 = 7,5\%$.

Exemple

En utilisant l'algorithme des k plus proches voisins pour classer des données tests on a obtenu les matrices de confusion suivantes :

pour $k = 4$

$$\begin{pmatrix} 9 & 1 & 0 \\ 2 & 10 & 3 \\ 1 & 2 & 12 \end{pmatrix}$$

pour $k = 7$

$$\begin{pmatrix} 10 & 0 & 0 \\ 1 & 11 & 1 \\ 0 & 1 & 14 \end{pmatrix}$$

- Donner le nombre de classes, et le nombre de données tests par classe.
  Il y a 3 classes, 10 données sont dans la première classe, et 15 dans chacune des deux autres.
- Calculer le pourcentage d'erreur commis pour chacune des valeurs de k. Conclure
  - Pour $k = 4$, le pourcentage d'erreur est $9/40 = 22,5\%$.
  - Pour $k = 7$, le pourcentage d'erreur est $3/40 = 7,5\%$.

  La valeur $k = 7$ semble être un bon choix.

Synthèse

La mise en oeuvre de l'algorithme demande donc à :

- Disposer d'un jeu de données $d = (d_0, \dots d_{n-1})$ déjà classées, c'est à dire attribuées à des classes $c_0, \dots c_{m-1}$

Synthèse

La mise en oeuvre de l'algorithme demande donc à :

- Disposer d'un jeu de données $d = (d_0, \dots d_{n-1})$ déjà classées, c'est à dire attribuées à des classes $c_0, \dots c_{m-1}$
- D'une distance entre deux données de façon à quantifier la notion de proximité.

Synthèse

La mise en oeuvre de l'algorithme demande donc à :

- Disposer d'un jeu de données $d = (d_0, \dots d_{n-1})$ déjà classées, c'est à dire attribuées à des classes $c_0, \dots c_{m-1}$
- D'une distance entre deux données de façon à quantifier la notion de proximité.
- Choisir un nombre k de voisins à considérer. La valeur de k influence la prédiction de l'algorithme (voir exemple précédent). En pratique, on teste plusieurs valeurs de k et on choisit celle qui donne les meilleurs résultats.

Synthèse

La mise en oeuvre de l'algorithme demande donc à :

- Disposer d'un jeu de données $d = (d_0, \dots d_{n-1})$ déjà classées, c'est à dire attribuées à des classes $c_0, \dots c_{m-1}$
- D'une distance entre deux données de façon à quantifier la notion de proximité.
- Choisir un nombre k de voisins à considérer. La valeur de k influence la prédiction de l'algorithme (voir exemple précédent). En pratique, on teste plusieurs valeurs de k et on choisit celle qui donne les meilleurs résultats.
- Une nouvelle donnée d_n est alors affectée à la classe de ses k plus proches voisins.

Synthèse

La mise en oeuvre de l'algorithme demande donc à :

- Disposer d'un jeu de données $d = (d_0, \dots d_{n-1})$ déjà classées, c'est à dire attribuées à des classes $c_0, \dots c_{m-1}$
- D'une distance entre deux données de façon à quantifier la notion de proximité.
- Choisir un nombre k de voisins à considérer. La valeur de k influence la prédiction de l'algorithme (voir exemple précédent). En pratique, on teste plusieurs valeurs de k et on choisit celle qui donne les meilleurs résultats.
- Une nouvelle donnée d_n est alors affectée à la classe de ses k plus proches voisins.
- La matrice de confusion donne une appréciation de l'efficacité de l'algorithme.

Synthèse

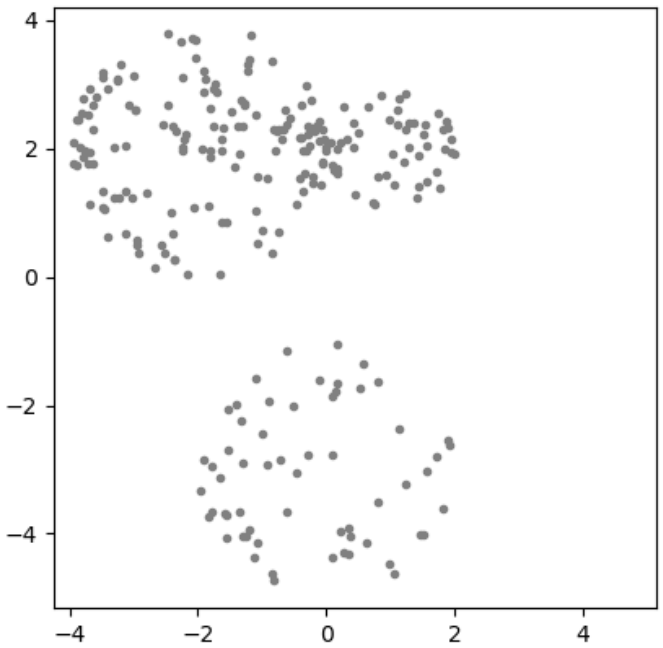
La mise en oeuvre de l'algorithme demande donc à :

- Disposer d'un jeu de données $d = (d_0, \dots d_{n-1})$ déjà classées, c'est à dire attribuées à des classes $c_0, \dots c_{m-1}$
- D'une distance entre deux données de façon à quantifier la notion de proximité.
- Choisir un nombre k de voisins à considérer. La valeur de k influence la prédiction de l'algorithme (voir exemple précédent). En pratique, on teste plusieurs valeurs de k et on choisit celle qui donne les meilleurs résultats.
- Une nouvelle donnée d_n est alors affectée à la classe de ses k plus proches voisins.
- La matrice de confusion donne une appréciation de l'efficacité de l'algorithme.

Voir le TP sur les iris de Fischer pour une mise en oeuvre d'un exemple en Python.

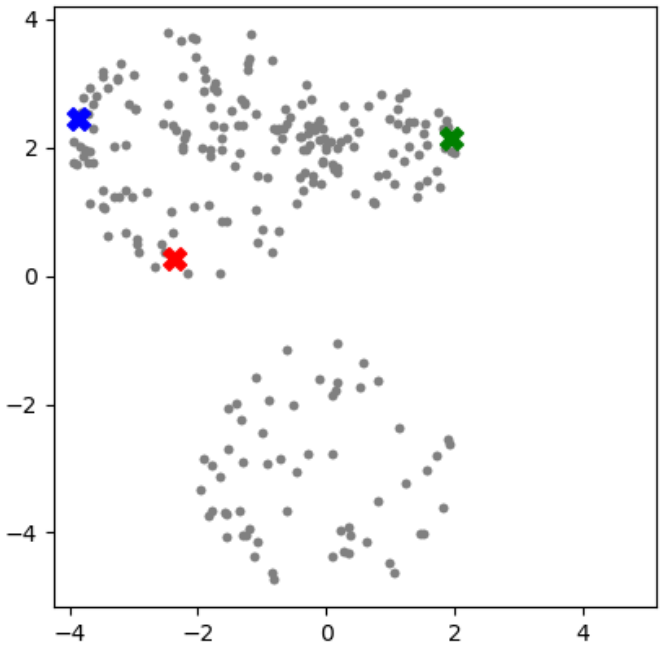
Un (autre) champ de fleurs

Dans un champ, à l'état sauvage trois espèces de fleurs ont poussées, chaque point gris indique l'emplacement d'une fleur et on souhaite prédire l'espèce de chacune des fleurs (en supposant qu'une espèce donnée pousse de façon préférentielle dans une certaine zone).



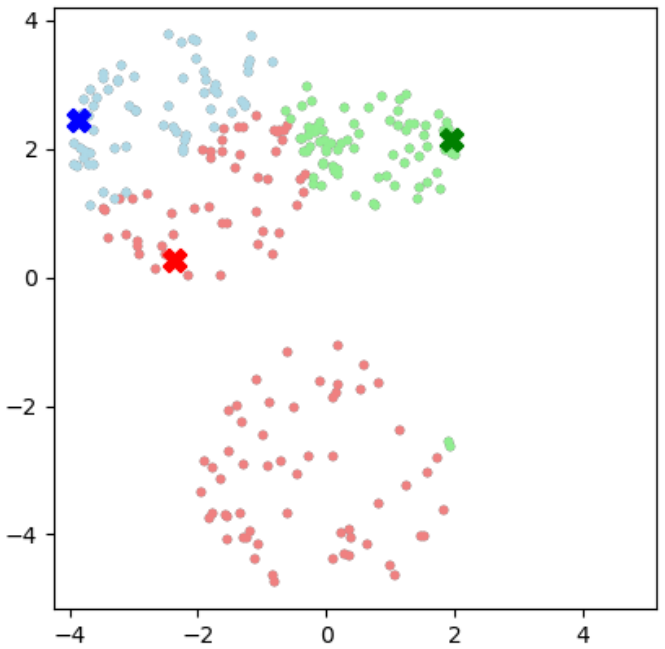
Un (autre) champ de fleurs

On commence par choisir au hasard trois fleurs et on considère qu'elles représentent chacune des trois espèces.



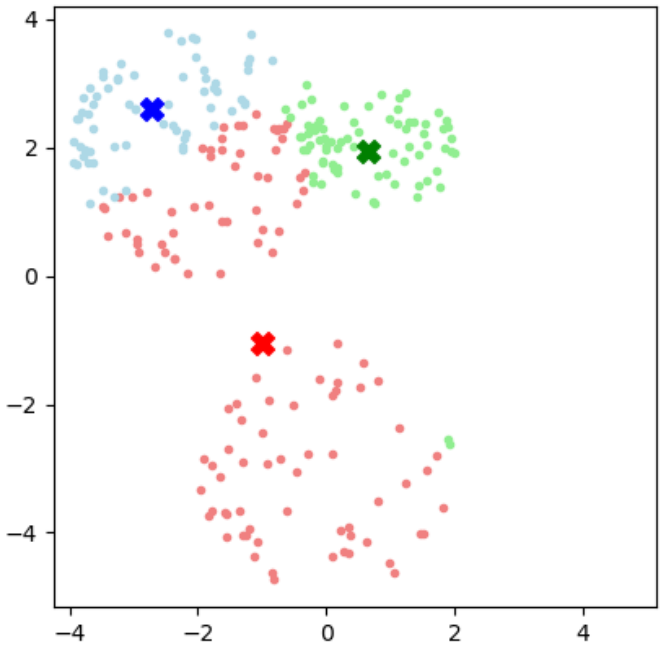
Un (autre) champ de fleurs

On attribut alors à chaque fleur une espèce en utilisant la proximité avec les trois fleurs choisies au hasard initialement.



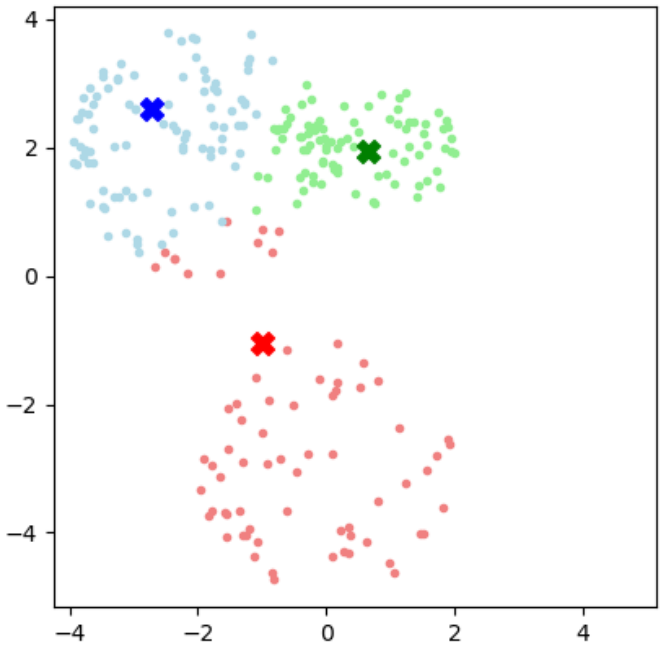
Un (autre) champ de fleurs

Dans chaque espèce on calcule alors la position du centre du nuage de points en faisant la moyenne des fleurs appartenant à cette espèce.



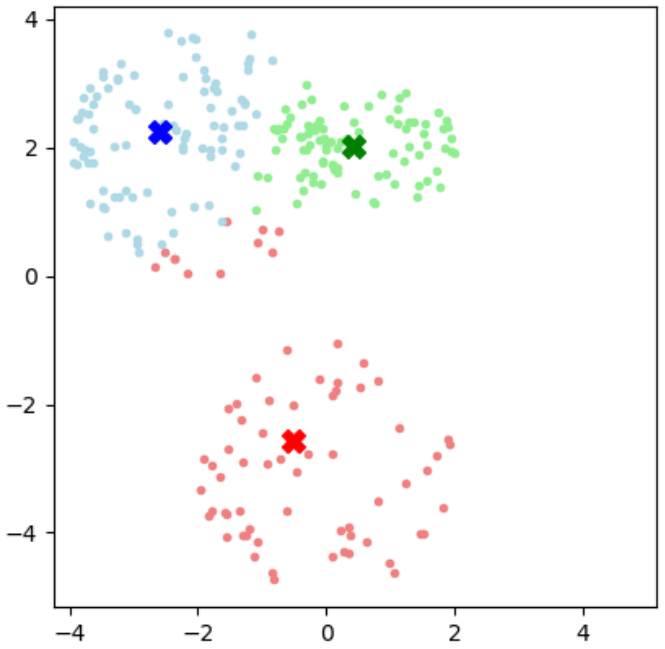
Un (autre) champ de fleurs

Comme précédemment, on affecte de nouveau les fleurs à une espèce par rapport à leur proximité aux nouveaux centres.



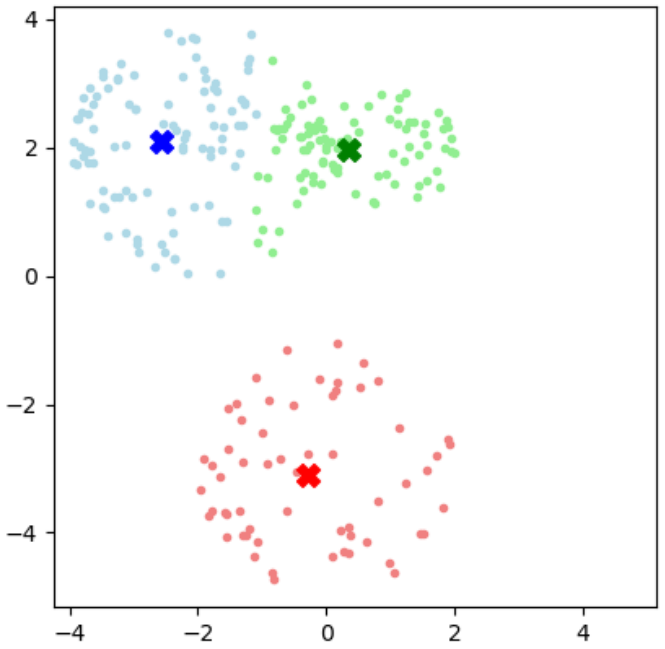
Un (autre) champ de fleurs

On réitère les étapes précédentes : calcul des nouveaux centres puis affectation suivant la proximité.



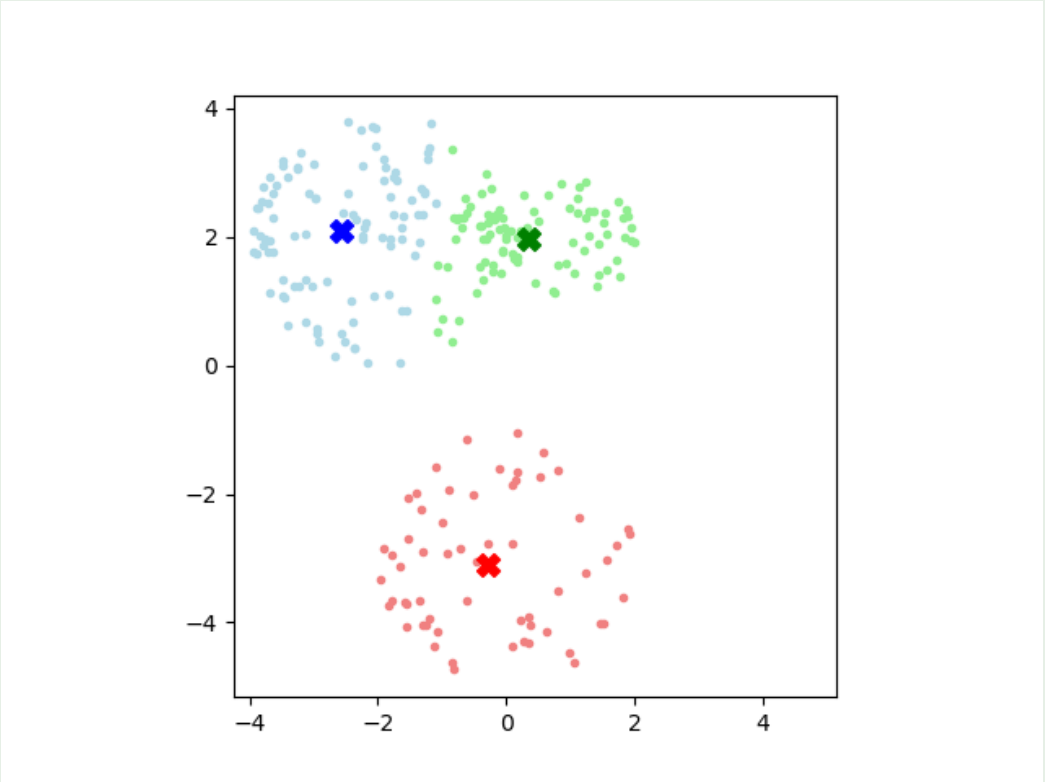
Un (autre) champ de fleurs

On réitère les étapes précédentes : calcul des nouveaux centres puis affectation suivant la proximité.



Un (autre) champ de fleurs

Après un certain nombre d'étapes, la répartition se stabilise.



Principe de l'algorithme

- L'algorithme des k-moyennes est un algorithme de classification des données appartenant à la famille des algorithmes d'apprentissage *non supervisé*.

Principe de l'algorithme

- L'algorithme des k-moyennes est un algorithme de classification des données appartenant à la famille des algorithmes d'apprentissage *non supervisé*.
- Les étapes de l'algorithme sont :

Principe de l'algorithme

- L'algorithme des k-moyennes est un algorithme de classification des données appartenant à la famille des algorithmes d'apprentissage *non supervisé*.
- Les étapes de l'algorithme sont :
  1. Choisir un entier k (le nombre de clusters à former) et choisir aléatoirement k points de données comme centres des clusters

Principe de l'algorithme

- L'algorithme des k-moyennes est un algorithme de classification des données appartenant à la famille des algorithmes d'apprentissage *non supervisé*.
- Les étapes de l'algorithme sont :
  1. Choisir un entier k (le nombre de clusters à former) et choisir aléatoirement k points de données comme centres des clusters
  2. Affecter chaque donnée au cluster dont le centre est le plus proches

Principe de l'algorithme

- L'algorithme des k-moyennes est un algorithme de classification des données appartenant à la famille des algorithmes d'apprentissage *non supervisé*.
- Les étapes de l'algorithme sont :
  1. Choisir un entier k (le nombre de clusters à former) et choisir aléatoirement k points de données comme centres des clusters
  2. Affecter chaque donnée au cluster dont le centre est le plus proches
  3. Mettre à jour les centres de chaque cluster en prenant comme nouveau centre la moyenne des points du cluster

Principe de l'algorithme

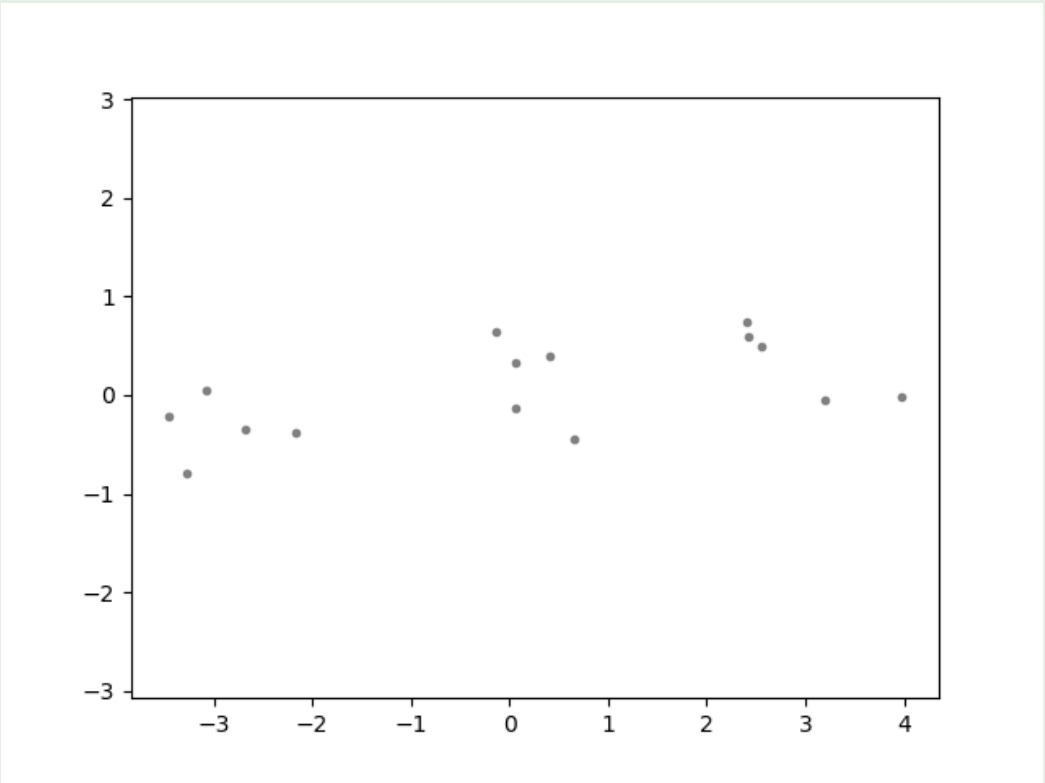
- L'algorithme des k-moyennes est un algorithme de classification des données appartenant à la famille des algorithmes d'apprentissage *non supervisé*.
- Les étapes de l'algorithme sont :
  1. Choisir un entier k (le nombre de clusters à former) et choisir aléatoirement k points de données comme centres des clusters
  2. Affecter chaque donnée au cluster dont le centre est le plus proches
  3. Mettre à jour les centres de chaque cluster en prenant comme nouveau centre la moyenne des points du cluster
  4. Répéter les étapes 2 et 3 jusqu'à atteindre un critère d'arrêt (stabilisation ou nombre maximal d'itérations)

Principe de l'algorithme

- L'algorithme des k-moyennes est un algorithme de classification des données appartenant à la famille des algorithmes d'apprentissage *non supervisé*.
- Les étapes de l'algorithme sont :
  1. Choisir un entier k (le nombre de clusters à former) et choisir aléatoirement k points de données comme centres des clusters
  2. Affecter chaque donnée au cluster dont le centre est le plus proches
  3. Mettre à jour les centres de chaque cluster en prenant comme nouveau centre la moyenne des points du cluster
  4. Répéter les étapes 2 et 3 jusqu'à atteindre un critère d'arrêt (stabilisation ou nombre maximal d'itérations)
- On montre (hors programme) que cet algorithme converge toujours.

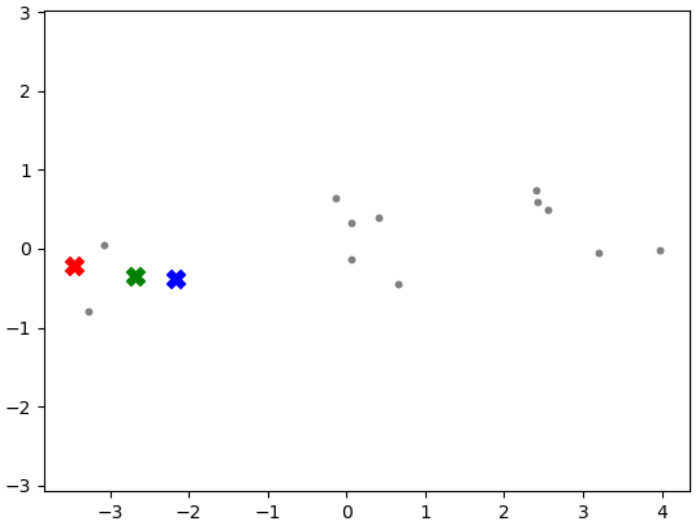
Exemple

Le choix des centres initiaux influence les résultats de l'algorithme. Sur l'exemple ci-dessous on « voit » bien trois cluster bien séparés contenant chacun cinq points mais ils ne seront par forcément détecté correctement.

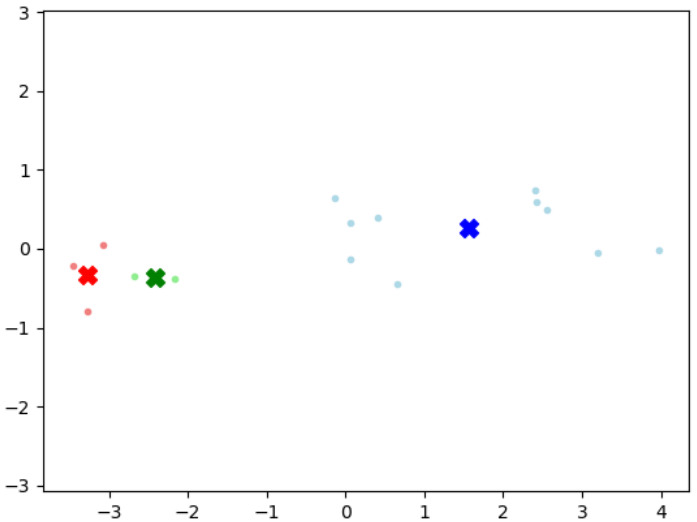


Exemple

Centres initiaux

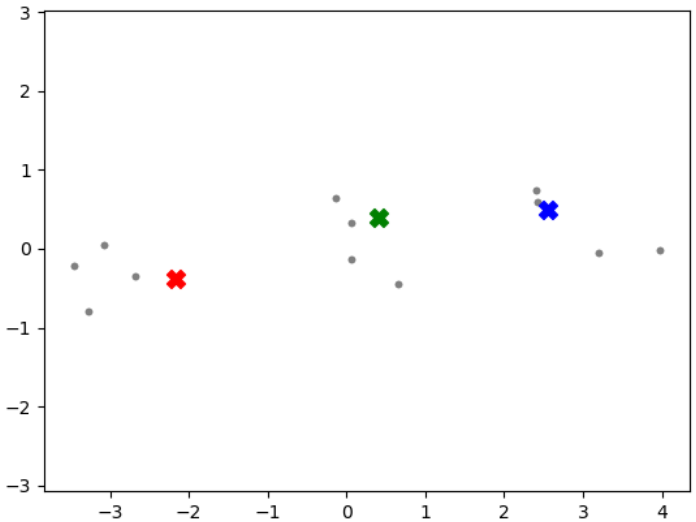


Résultats de la classification



Exemple

Centres initiaux



Résultats de la classification