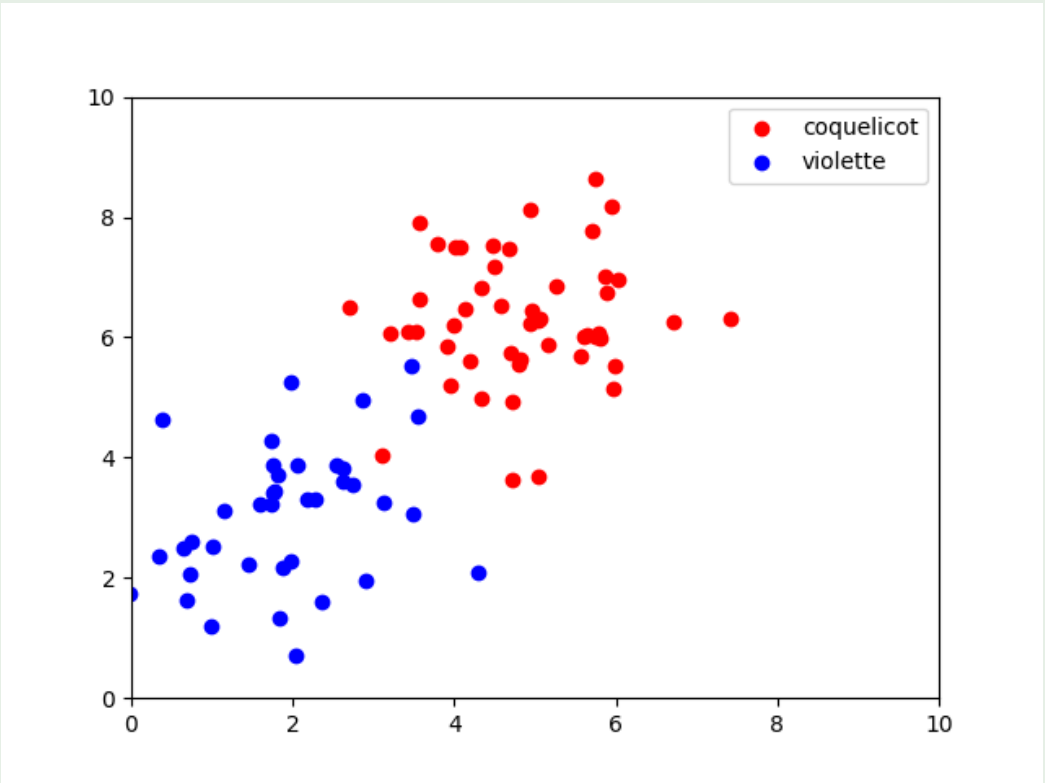
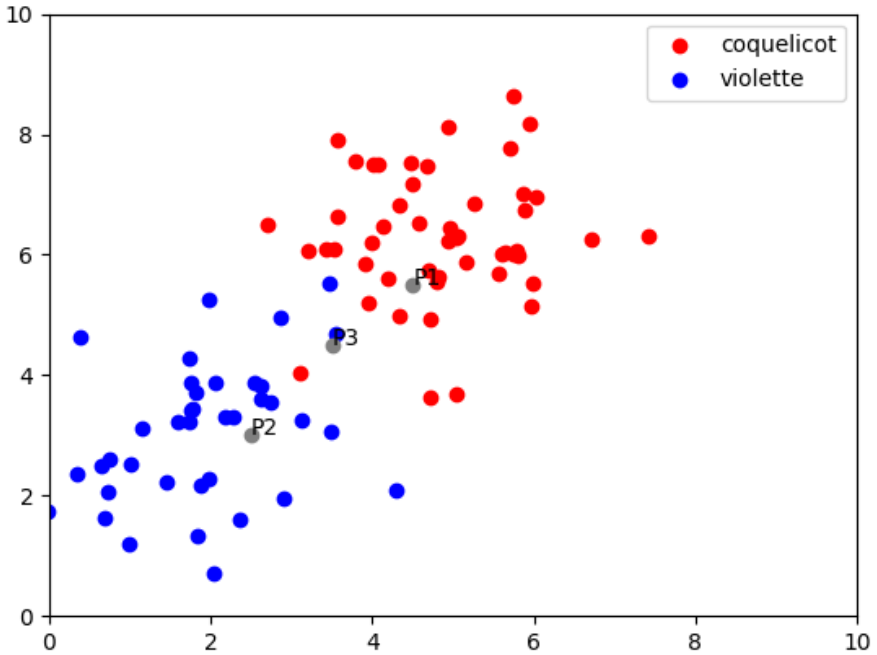
Un champ de fleurs

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



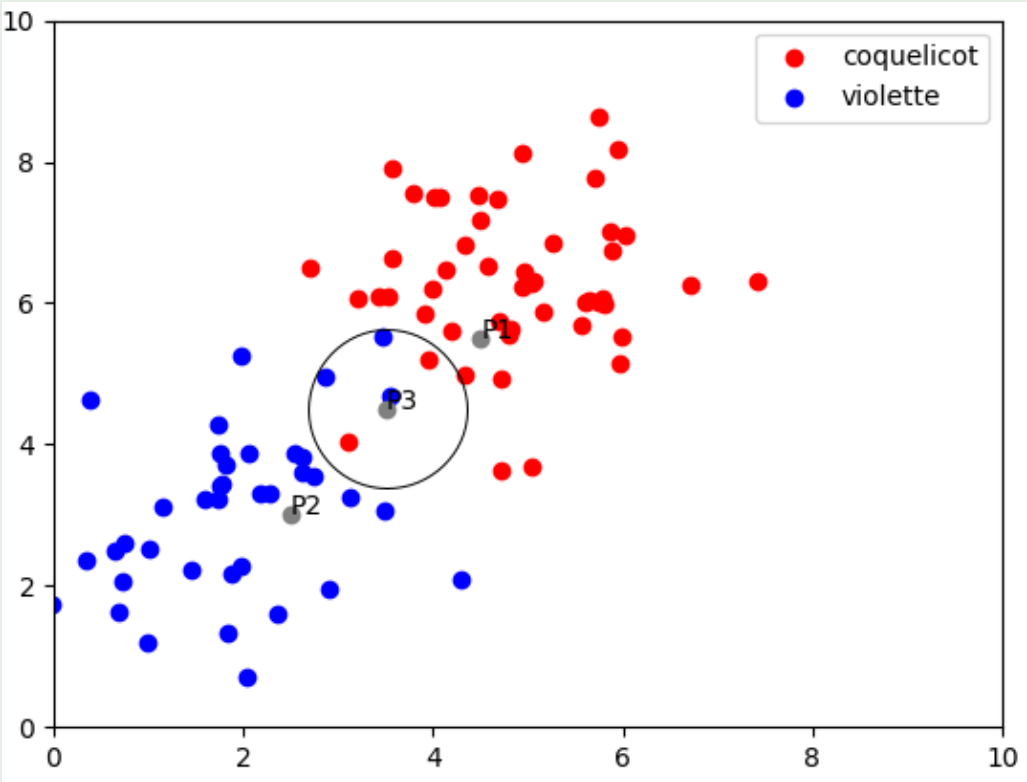
Un champ de fleurs

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.

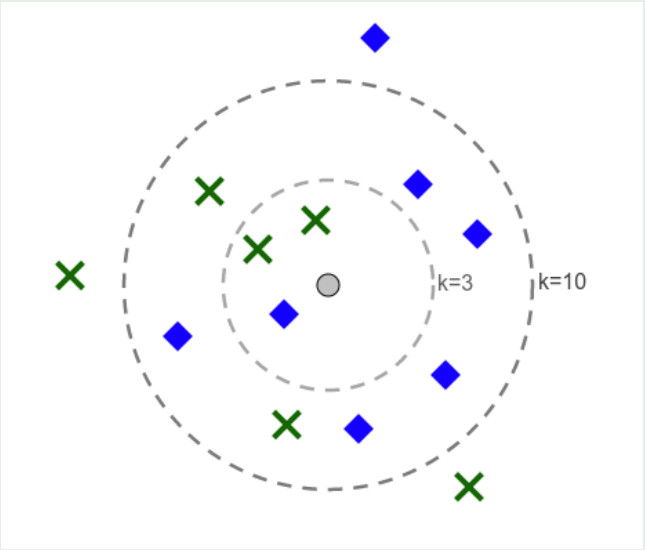
Principe de l'algorithme

- 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

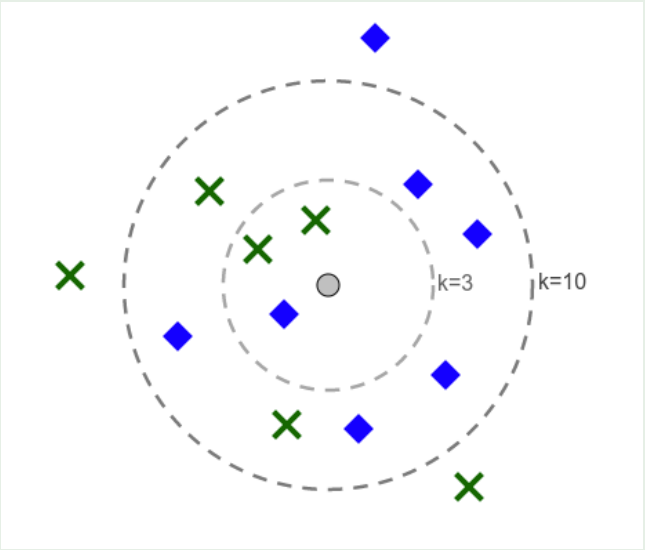
- L'algorithme des k plus proches voisins est un algorithme de classification des données appartenant à la famille des algorithmes d'apprentissage supervisé.
- On dispose d'un jeu de données qui associe chaque donnée à une classe.
- L'algorithme attribut à une nouvelle donnée d non classée la classe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

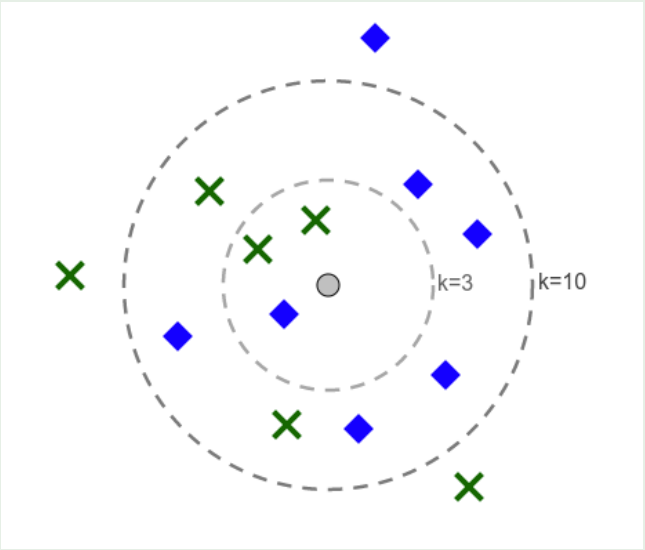
Exemple



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

- Pour $k = 3$?

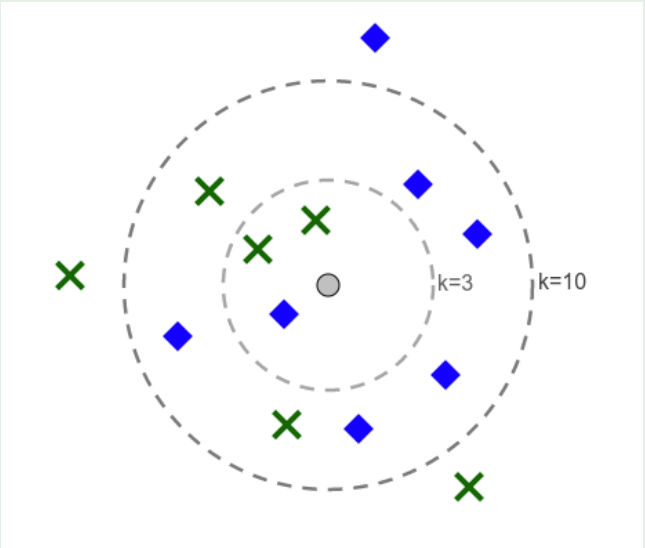
Exemple



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

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

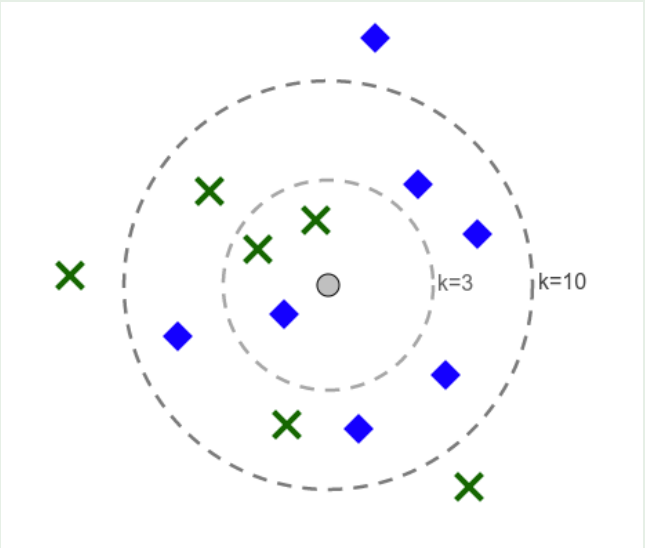
Exemple



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.

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.

On suppose qu'on dispose :

On suppose qu'on dispose :

- d'un jeu de données $(d_0, \dots d_{n-1})$

On suppose qu'on dispose :

- d'un jeu de données $(d_0, \dots d_{n-1})$
- d'une fonction `distance` prenant en argument deux données d_1 et d_2 et calculant la distance qui les sépare

On suppose qu'on dispose :

- d'un jeu de données $(d_0, \dots d_{n-1})$
- d'une fonction `distance` prenant en argument deux données d_1 et d_2 et calculant la distance qui les sépare
- d'une nouvelle donnée e non encore classée