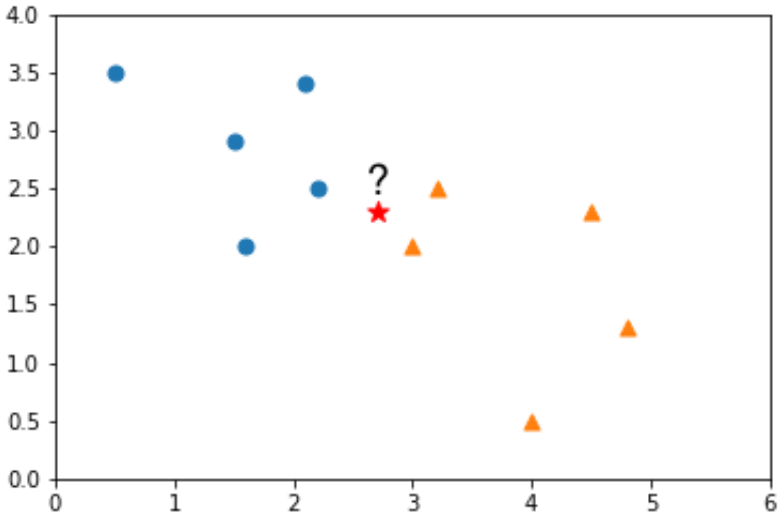
IHDCM037 - Machine Learning
Travaux pratiques
Séance 3 : K-nearest neighbors

Remarque : Pour réaliser les travaux pratiques de ce cours, nous travaillerons avec le langage `Python` sur des `Jupyter Notebook`. Ceux-ci peuvent être utilisés en ouvrant la plateforme `Anaconda Navigator` et en lançant `Jupyter Notebook`.

Première partie

Cette séance est dédiée à l'algorithme de classification **supervisée** *K-nearest neighbors* et plus particulièrement à son application à un jeu de données constitué de 3 sortes d'iris différentes.

Avant de passer à l'application du modèle de Machine Learning sur les données concernant les iris, voici un exemple simple. Le graphique suivant représente des données appartenant à 2 classes différentes (les ronds et les triangles). La donnée représentée par une étoile est une donnée qu'il faut classer selon l'algorithme des *K-nearest neighbors*.



- Si le paramètre K est fixé à 3, quelle sera la classe prédite pour la donnée "étoile" par le classificateur *K-nearest neighbors* ? .....
- Si maintenant le paramètre K est fixé à 6, quelle sera la classe prédite pour la donnée "étoile" par le classificateur *K-nearest neighbors* ? .....

Durant les séances précédentes, seuls un ensemble d'entraînement et un ensemble de test ont été considérés pour entraîner et tester la performance de différents modèles de Machine Learning. Dans cette séance, un nouvel ensemble sera introduit : l'ensemble de validation, car ce nouvel ensemble permet d'appliquer un processus appelé *cross-validation*. Voici les différentes étapes à suivre pour appliquer une *cross-validation* et une classification selon l'algorithme des *K-nearest neighbors* aux données présentées précédemment :

1. Importez le *dataset* `iris` de la librairie `sklearn`. Pour cela, vous devez d'abord importer le module `datasets` de `sklearn`.
2. Sauvegardez la partie *data* du *dataset* dans un premier tableau et la partie *target* dans un deuxième. Afin de pouvoir visualiser les données, extrayez dans un premier temps uniquement les données concernant la longueur et la largeur des sépales des iris.
3. Visualisez les données (largeur des sépales en fonction de leur longueur) grâce à la fonction `plot_dataset` fournie sur Webcampus. Cette fonction prend en argument **tout** le jeu de données, ainsi que les **indices** des deux *features* qui aideront à représenter les données sur le graphique (par exemple : `plot_dataset(diabetes,5,8)`).
4. Effectuez une première séparation des données en un ensemble contenant les données d'entraînement et de validation, et un autre ensemble contenant les données de test. L'ensemble de test doit contenir 10% des données initiales.

   Remarque : Faites attention à la façon dont les données sont ordonnées dans le *dataset* initial et utilisez une manière adéquate pour séparer des données.
5. À partir de l'ensemble contenant les données d'entraînement et de validation, créez une *cross-validation*. Pour cela, effectuez une deuxième division et séparez cet ensemble en deux nouveaux ensembles : un pour l'entraînement et un pour la validation, et faites une boucle pour effectuer 5 séparations différentes. L'ensemble de validation doit contenir 25% des données contenues dans l'ensemble entraînement-validation.
   - Quelle fonction allez-vous utiliser pour séparer les données en un ensemble d'entraînement et un ensemble de validation ? .....
   - Quel paramètre de cette fonction allez-vous modifier pour obtenir 5 séparations différentes ?

     .....
6. Pour chaque séparation, entraînez le classificateur *K-nearest neighbors* sur l'ensemble d'entraînement grâce à la fonction `KNeighborsClassifier` du module `neighbors` de la librairie `sklearn`. Fixez le paramètre `n_neighbors` à 3.
   - Qu'induit le paramètre `n_neighbors = 3` ? .....

     .....
7. Pour chaque séparation, calculez *l'accuracy* pour l'ensemble de validation et stockez-le dans une liste.
8. Retenez la séparation avec laquelle la plus haute *accuracy* a été obtenue, ré-entraînez un *"best"* modèle avec cette séparation et calculez *l'accuracy* pour l'ensemble de test.
   - Que vaut *l'accuracy* pour l'ensemble de test ? .....

Deuxième partie

Effectuez les mêmes étapes que celle de la première partie en ne conservant que la longueur et la largeur des pétales des iris (largeur des pétales en fonction de leur longueur).

- Que vaut *l'accuracy* pour l'ensemble de test ? .....
- Comment pouvez-vous expliquer cette valeur, ainsi que la différence avec celle de la première partie ? .....

  .....