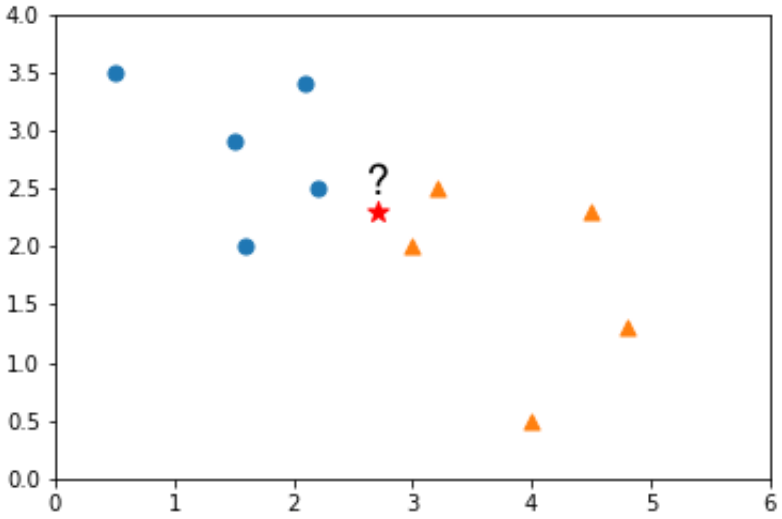
IHDCM037 - Machine Learning
Travaux pratiques
Séance 3 : K-nearest neighbors

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

Voici les différentes étapes à suivre pour créer un premier modèle *K-nearest neighbor* sur *dataset* iris :

1. Importez le *dataset* `iris` de la librairie `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. Formez un ensemble d'entraînement et un ensemble de test. Pour avoir les mêmes résultats, formez un ensemble d'entraînement qui contient 67% des données initiales et utilisez l'hyperparamètre `random_state = 42`.
5. Entraînez un modèle *K-nearest neighbors* sur l'ensemble d'entraînement. Fixez l'hyperparamètre `n_neighbors = 3`.
   - Quelle fonction utilisez-vous pour créer le type de modèle *K-nn* ? .....
   - Qu'induit l'hyperparamètre `n_neighbors = 3` ? .....

     .....
6. Calculez l'*accuracy* pour l'ensemble d'entraînement et pour l'ensemble de test.
   - Pourquoi l'*accuracy* est-elle une métrique fiable dans ce cas particulier ? .....

     .....
   - Que vaut *l'accuracy* pour l'ensemble de test ? .....

Deuxième partie

Dans la partie précédente, le *dataset* a été divisé en un seul ensemble d'entraînement et de test. Cependant, il existe un processus qui exécute cette division plusieurs fois et qui permet d'éviter l'*overfitting* : ..... Voici les différentes étapes à suivre pour appliquer ce processus dans le cas d'une classification des données contenues dans le *dataset* iris selon l'algorithme des *K-nearest neighbors* :

1. Importez le *dataset* `iris` de la librairie `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. Créez une *cross-validation* avec 5 séparations entraînement/test différentes.
   - Quel est le principe d'une *cross-validation* ? Comment procédez-vous pour la créer ?

     .....

     .....

     .....
   - Comment calculer les métriques de performance quand on effectue une cross-validation ?

     .....

     .....

4. Pour chaque séparation, générez un classificateur *K-nearest neighbors* avec l'hyperparamètre `n_neighbors = 3`. Sauvegardez la valeur de l'*accuracy* pour l'ensemble d'entraînement et de test pour chaque séparation, puis calculez l'*accuracy* finale pour chaque ensemble.
    - Quelle différence observez-vous avec les *accuracy* calculées lors de la première partie ?

      .....

      .....

Troisième partie

Effectuez les mêmes étapes que celles 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 ? .....

  .....