

TP5

Dans ce TP vous aurez besoin des fichiers suivant:

- **learning.py** (que vous pouvez télécharger à partir du lien suivant: [lien](#))
- **utils.py** (que vous pouvez télécharger à partir du lien suivant: [lien](#))
- **iris.csv** (que vous pouvez télécharger à partir du lien suivant: [lien](#))
- **restaurant.csv** (que vous pouvez télécharger à partir du lien suivant: [lien](#))
- **orings.csv** (que vous pouvez télécharger à partir du lien suivant: [lien](#))
- **zoo.csv** (que vous pouvez télécharger à partir du lien suivant: [lien](#))

Vous mettez les fichiers learning.py et utils.py dans le dossier **TP5**. Vous créez un dossier nommé: **aima-data** dans le répertoire **TP5**. Vous mettez dans ce dossier les fichiers **iris.csv**, **restaurant.csv**, **orings.csv** et **zoo.csv**

Vous modifiez le fichier **learning.py** en transformant en commentaire les instruction suivantes:

- `from qpsolvers import solve_qp` (transformez en `#from qpsolvers import solve_qp`)
- `from probabilistic_learning import NaiveBayesLearner` (transformez en `#from probabilistic_learning import NaiveBayesLearner`)

Créez un nouveau fichier sous le nom de **tp5.py** dans le dossier **TP5**.

But du TP :

Utiliser deux méthodes d'apprentissage machine : **réseaux de neurones feedforward** et **arbres de décision** pour pouvoir déterminer la classification d'une « plante **iris** » particulière selon les trois types suivants: **setosa**, **versicolor**, ou **virginica**. Les attributs pris en considération pour la classification sont dans cet ordre (valeurs en cm):

1. La longueur de la sépale
2. La largeur de la sépale
3. La longueur de la pétale
4. La largeur de la pétale

La plante à classifiez à les valeurs d'attribut suivants (valeurs en cm) :

5. La longueur de la sépale : 6.7, la largeur de la sépale: 3.0 , la longueur de la pétale: 4.7 et la largeur de la pétale : 1.4

Les données d'apprentissage sont dans le fichier **iris.csv**. Ce fichier sera utilisé pour créer un objet de type **DataSet** qui sera utilisé pour l'apprentissage machine. La table correspondante contient pour chaque instance de plante :

- Les 4 attributs de classification cités précédemment.
- La classe correspondante.

Étapes du TP :

Dans le fichier **TP5.py**, écrire les instructions qui permettent de :

1. Importer toutes les définitions du fichier **learning**.

2. Créer une instance de l'objet **DataSet** , avec un seul paramètre nommé : **name**. Le paramètre **name** devra avoir la valeur du nom du fichier contenant les données (sans l'extension). Autrement dit **name='iris'**
3. Utiliser la fonction **DecisionTreeLearner** avec comme paramètre l'instance de l'objet **DataSet** créée précédemment. L'appel de la fonction avec ces données va retourner un objet de type **DecisionFork** qui représentera l'**arbre de décision** généré à partir des données en paramètre. L'objet **DecisionFork** en question pourra être utilisé plus tard comme fonction pour prédire la classe d'une plante donnée en paramètre (ses attributs seront donnés en paramètre).
4. Créer une liste des valeurs des attributs suivants dans cet ordre (seulement les valeurs numériques) :
 1. La longueur de la sépale : 6.7
 2. La largeur de la sépale: 3.0
 3. La longueur de la pétale: 4.7
 4. La largeur de la pétale : 1.4Les attributs correspondant à la plante dont on veut prédire la classe. Et la liste aura donc **4** valeurs.
5. Utiliser l'instance de l'objet **DecisionFork** créée dans l'étape 3 comme fonction pour prédire la classe de la plante décrite par la liste créée précédemment. La fonction a besoin d'un seul paramètre : la liste des attributs de la plante (la liste créée à l'étape 4). Afficher ensuite la classe prédite.
6. Convertir les valeurs des classes de l'instance de l'objet **DataSet** créée à l'étape 2 en valeurs numériques en appelant la méthode **classes_to_numbers** (sans paramètres) à partir de cette même instance de l'objet **DataSet** créée à l'étape 2. Les classes seront automatiquement converties en valeurs numériques :
 1. **0** correspondra à **Setosa**
 2. **1** correspondra à **Versicolor**
 3. **2** correspondra à **Virginica**
7. Appeler la fonction **NeuralNetLearner** avec comme paramètre l'objet **DataSet** créé à l'étape 2 et les paramètres nommés **learning_rate=0.04** et **epochs=500**. La fonction va générer un réseau de neurones **FeedForward** et retourner une autre fonction à mettre dans une variable pour pouvoir appeler plus tard. Cette fonction permettra de prédire la classification d'une plante donnée en paramètre (par ces attributs) en utilisant le réseau de neurones généré.
8. Appeler la fonction retournée à l'étape 7 avec comme paramètre la liste créée à l'étape 4 pour prédire la classe correspondante à ces attributs.
9. Afficher la classe prédite à l'étape 8 :
 1. en valeur numérique (la valeur retournée à l'étape 8)
 2. en chaîne de caractère représentant le nom de la classe correspondante à la valeur retournée à l'étape 8 en utilisant les correspondances décrites dans l'étape 6. Par exemple : si la valeur retournée à l'étape 8 est **0**, vous devrez afficher **0** et la classe correspondante qui est **Setosa**.