

Intelligence artificielle – TP n°1

Licence 3 Informatique
Année 2014-2015
TPs: Pierre Parrend
TP n°1

Objectifs de ce TP

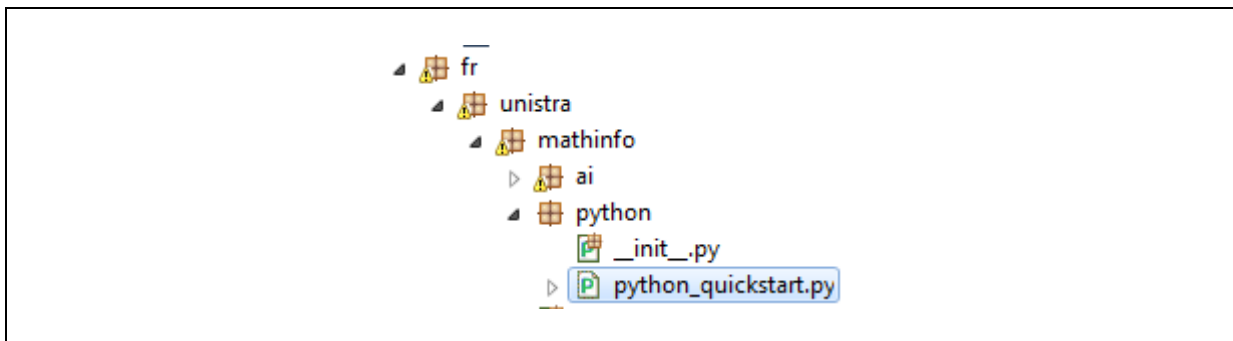
L'objectif de ce TP est :

- D'introduire l'usage du langage python
- de maîtriser une des opérations de bases utilisées dans le traitement des données en intelligence artificielle : la normalisation.

Le langage proposé est Python. Tout autre langage peut être utilisé

Quelques lignes de Python

1. Créer en python
 - 1.1. Un package fr.unistra.python
 - 1.2. Un module python_quickstart, dans le fichier python_quickstart.py dans ce package



2. Créer et exécuter la méthode 'main()' de votre module

```
5  
6  
7 if __name__ == '__main__':  
8     print("Hello main method")  
9  
10
```

Console

<terminated> D:\USERS\Pierre.Parrend\Google C
Hello main method

3. Créer une classe Square dans le module square.py, qui contiennent les méthodes
 - 3.1. setSize()
 - 3.2. getVolume()

Module square.py

```
class Square(object):  
    """  
    classdocs  
    """  
  
    def setSize(self, size):  
        self.size = size  
  
    def getSurface(self):  
        return self.size * self.size  
  
    def __init__(self):  
        """  
        Constructor  
        """  
        print("Square constructor")
```

Module python_quickstart.py :

```
from fr.unistra.mathinfo.python.square import Square  
  
if __name__ == '__main__':  
    print("Hello main method")  
  
    my_square = Square()  
    my_square.setSize(6)  
    print("square surface: " + str(my_square.getSurface()))
```

4. Créer une méthode run_array() qui enlève la dernière valeur d'un tableau, et y ajoute une nouvelle valeur

```
def run_array():
    my_array = [1, 2, 3, 4]
    print(my_array)
    last = my_array.pop()
    print(last)
    my_array.append(7)
    print(my_array)
```

5. Créer une méthode run_matrix() qui affiche une matrice ligne par ligne :

```
def run_matrix():
    my_matrix = [[1,2],[3,4],[5,6]]
    print(my_matrix)
    for line in my_matrix:
        print(line)
```

6. Créer un module python_test pour effectuer des tests unitaires

```
1'''
2Created on 19 mars 2015
3
4@author: Pierre.Parrend
5'''
6import unittest
7
8
9class Test(unittest.TestCase):
10
11
12    def setUp(self):
13        pass
14
15
16    def tearDown(self):
17        pass
18
19
20    def testHelloTest(self):
21        a = 1
22        b = 1
23        self.assertTrue(a == b, "a is b")
24
25
26if __name__ == "__main__":
27    #import sys;sys.argv = ['', 'Test.testHelloTest']
28    unittest.main()
```

```
<terminated> 0
Finding files... done.
Importing test modules ... done.
-----
Ran 1 test in 0.001s
OK
```

7. Créer une méthode load_csv qui charge et affiche un fichier csv

```
import csv

def load_csv():
    iris_file = "../../../datasets/iris.csv"
    iris_data = open(iris_file, 'r')
    data = csv.reader(iris_data)

    for line in data:
        print(line)
```

Analyse – Normalisation

8. Pour le dataset iris.csv, écrire l'algorithme permettant de normaliser les champs numériques entre -1 et 1
9. Pour le dataset iris.csv, écrire l'algorithme permettant de normaliser les classes sémantiques en mode 1-parmi-N

Normalisation – Exercice

10. Créer un module python fr.unistra.mathinfo.ai.labsession1.normalization
11. Créer un module unittest python test.normalization_test, avec

```
import unittest
from fr.unistra.mathinfo.ai.labsession1.normalization import Normalizer

class Test(unittest.TestCase):

    if __name__ == "__main__":
        #import sys;sys.argv = ['', 'Test.test_extract_classes_from_matrix']
        unittest.main()
```

12. Stocker les données du dataset sous forme de matrice en utilisant le module python csv
13. Afficher le dataset (1 entrée de données par ligne)
14. Extraire les classes correspondant aux espèces de fleurs
15. Donner l'expression permettant d'obtenir une valeur normalisée à partir du minimum réel, maximum réel, minimum de l'espace de normalisation, maximum de l'espace de normalisation, et valeur courante
16. Normaliser les valeurs des colonnes 1 à 4 entre -1 et 1
17. Normaliser les espèces en mode 1-parmi-n (1-of-n)

Exercice 2 : Classification

18. Développez, un arbre de décision permettant de classer les espèces d'iris selon la forme de leurs pétales et sépales
 - 18.1. Ecrire une méthode permettant d'obtenir la surface du pétale, et stocke le résultat dans un tableau [surface, classe]
 - 18.2. Ecrire une méthode permettant d'ordonner par ordre croissant tableau [surface, classe]
 - 18.3. Ecrire une méthode permettant d'extraire les déciles du tableau [surface, classe]
 - 18.4. Ecrire une méthode donnant les probabilités d'occurrence de chaque classe pour les différents déciles
 - 18.5. Représenter l'arbre de décision, en séparant les déciles selon la probabilité d'obtenir un résultat non ambigu