

# Python pour l'IA

## TP2 : La librairie « NumPy »

Mme I.Yahiaoui

### Exercice 1 :

- Créer un tableau depuis la liste [1.6, 0.5, 7.4, 22.3] et stocker le dans la variable « vec ».
  - Afficher le contenu de la variable « vec »
  - Afficher le type de données du tableau « vec », sa forme, sa dimension et sa longueur.
  - Afficher le troisième élément du tableau.
  - Modifier la valeur « 0.5 » par « 3.8 ».
  - Calculer la taille de la mémoire du tableau.
  - Convertir le type du tableau vers « int »
  - Calculer la taille de la mémoire du tableau après la conversion du type.
  - Sauvegarder le tableau dans un fichier texte.
- Créer une matrice « mat » depuis la liste [[0, 3, 6, 9], [5, 10, 15, 20], [30, 40, 50, 60]]
  - Afficher le contenu de la variable « mat »
  - Afficher le type de données de la matrice « mat », sa forme, sa dimension et sa longueur.
  - Pouvez-vous obtenir la dimension et la longueur à partir de la forme ?
  - Créer la matrice « mat\_1 » extraite de « mat » telles que  $i \geq 1, j \leq 2$ .
  - Modifier la matrice « mat » afin que sa dernière ligne soit divisée par 10 et que sa première colonne soit multipliée par 2.
- Créer un tableau multidimensionnel « nd\_vec » d'une dimension > 2
  - Afficher le contenu de « nd\_vec »
  - Afficher le type de données du tableau « nd\_vec », sa forme, sa dimension et sa longueur.
  - Pouvez-vous obtenir la dimension et la longueur à partir de la forme ?
  - Afficher un élément de ce tableau ainsi qu'un sous tableau.

### Exercice 2 :

- Créer un tableau 1D vide de taille 10 et de type « int » ; observer les valeurs du tableau.
- Créer un tableau 2D vide de dimension (3\*3) et de type « float » ; observer les valeurs du tableau.
- Créer un tableau de 3 dimensions (3\*3\*3) initialisé avec des zéros « 0 ».
- Créer une matrice de taille (5\*5) comportant des uns « 1 ».
- Créer un tableau de taille (10\*10) initialisé avec la valeur « np.inf ».
- Créer une matrice d'identité de taille (4\*4).
- Créer une matrice dont la diagonale est égale à [2, 3, 4, 5] et des 0 ailleurs.

### Exercice 3 :

- Créer un tableau comportant la série d'entiers de 5 à 25 : [5, 6, 7, ..., 23,24,25]
- Créer un tableau contenant tous les entiers pairs entre 100 et 150 inclus.
- Créer un tableau qui comporte les dates du mois de Février 2019, ce tableau est de type « datetime64[D] »
- Créer un tableau de dix éléments équidistants entre 2 et 5 inclus.
- Soit la fonction  $y = f(x) = (e^x - e^{-x}) / (e^x + e^{-x})$ . Créer des tableaux x, y qui contiennent respectivement 50 valeurs réparties uniformément sur [-5, 5] et l'évaluation de la fonction sur ces valeurs
- Écrire un programme NumPy qui permet de générer un nombre aléatoire entre 0 et 1.
- Créer un tableau de la dimension et de la taille de votre choix contenant des valeurs entre 34 et 76.
- Faites un tableau de forme (3, 5) avec des nombres aléatoires à partir d'une distribution normale standard (une distribution normale avec une moyenne 0 et une variance 1).
- Créer une matrice (10x5) comportant des entiers aléatoires entre 0 et 100.
  - Standardiser cette matrice telle que chaque colonne aura une moyenne nulle et un écart type égal à 1.

### Exercice 4 :

- Créer une matrice aléatoire 10x10.
  - Sélectionner l'élément dans la première ligne et la première colonne.
  - Mettre l'élément de la dernière ligne et de la dernière colonne à -1.
  - Sélectionner les éléments de la deuxième ligne.
  - Sélectionner les éléments de la troisième colonne.
  - Remplacer tous les éléments de la deuxième ligne par 0.
- Écrire un programme NumPy pour inverser un tableau (le premier élément devient le dernier).
- Écrire un programme NumPy pour créer un tableau 2d avec 1 sur la bordure et 0 à l'intérieur.
- Écrire un programme NumPy pour créer une matrice 8x8 et remplissez-la avec un motif en damier.

### Exercice 5 :

- Écrire un programme NumPy dans lequel vous allez :
  - Définir un tableau à une dimension de la taille de votre choix.
  - Calculer l'écart type des valeurs présentes dans ce tableau.
  - Chercher les valeurs et les indices des éléments qui sont supérieurs à « la moyenne des valeurs + 1.96 l'écart type ».
  - Remplacer toutes ces valeurs par la valeur maximale du tableau.
  - Chercher les valeurs et les indices des éléments qui sont inférieurs à « la moyenne des valeurs - 1.96 l'écart type ».
  - Remplacer toutes ces valeurs par la valeur minimale du tableau
  - Compter le nombre d'éléments égaux au maximum et le nombre d'éléments égaux au minimum.
  - Compter le nombre d'éléments différents de min ou de max.

#### Exercice 6 :

- Écrire un programme NumPy pour créer une matrice 3x3 avec des valeurs allant de 2 à 10
- Créer la matricer suivante sans saisir explicitement ses valeurs :

```
[ [ 1,  6, 11],  
  [ 2,  7, 12],  
  [ 3,  8, 13],  
  [ 4,  9, 14],  
  [ 5, 10, 15] ]
```

- Diviser le tableau « s = np.arange(25).reshape(5,5) » afin de séparer chaque colonne à part.
- Créer la matrice suivante en concaténant 3 tableaux unidimensionnel. Utiliser np.arange et np.ones pour la création des tableaux.

```
[ [  7,  8,  9],  
  [  1,  1,  1],  
  [12, 18,  2] ]
```

#### Exercice 7 :

- Écrire un programme qui trouve les positions des éléments non nuls dans un tableau.
- Écrire un programme qui teste si aucun des éléments d'un tableau donné n'est nul.
- Utiliser la fonction numpy « np.pad » pour ajouter une bordure (remplie de 0) autour d'un tableau existant.
- Utiliser la fonction « np.asarray » pour créer un tableau à partir d'un tuple ou une liste.
- Tester la fonction « np.append » pour ajouter des éléments à la fin d'un tableau.
- Utiliser la fonction « np.in1d » pour tester si chaque élément d'un tableau 1-D est également présent dans un second tableau.
- Utiliser la fonction « np.intersect1d » pour trouver des valeurs communes entre deux tableaux.
- Utiliser la fonction « np.tile » pour construire un tableau en répétant les éléments d'un autre tableau.
- Utiliser la fonction « bincount » pour compter le nombre d'occurrences de chaque valeur d'un tableau de valeurs positives.

#### Exercice 8 :

Utiliser la librairie numpy pour lire un fichier représentant une « dataset ». Utiliser la fonction numpy.genfromtxt() pour lire le fichier « titanic.csv » dans un tableau qu'on nommera donnees»

- Afficher le type de la variable donnees en utilisant la fonction type()
- Afficher sa dimension, sa forme, etc
- Apporter des modifications à votre base et sauvegarder la nouvelle base dans un fichier « titanic\_mod.csv »