

Analyse de Données – Développeur

C5-160512-INFO

1. Préambule

Pr. Carl FRÉLICOT

FST/PAS221 – carl.frelicot@univ-lr.fr



Licence d'Informatique 3ème année

Automne 2021 - La Rochelle



1. Préambule
2. Tableaux et Espaces
3. Réduction de la Dimensionnalité
4. Classification Non Supervisée
5. Classification Supervisée



1. Préambule
 - 1.1 Objectifs et Moyens
 - 1.2 NoteBook
 - 1.3 Python 3
2. Tableaux et Espaces
3. Réduction de la Dimensionnalité
4. Classification Non Supervisée
5. Classification Supervisée



EC

- Intérêts et objectifs
 - ① continuum Licence 2 – Licence 3 – Master (parcours *Données*)
importance de la *Data Science*
Jupyter NoteBook, IPython
 - ② de la programmation
 - ③ mieux maîtriser les concepts vus en Licence 2
quid des étudiants qui n'ont pas suivi AD-Utilisateur ?
- Aspects organisationnels
 - maquette CM/TD/TP/TEA = 7.5h/9h/9h/0h
en pratique 7.5h/7.5h/7.5h/7.5h
TD/TP programmés en dernière semaine vs des créneaux de "réunion"
 - CM
problème ouvert
 - CC Intégral et pas d'évaluation des TP ni des TEA
~~2ème session~~
2ème chance

Des questions ?



1. Préambule

- 1.1 Objectifs et Moyens
- 1.2 NoteBook
- 1.3 Python 3

2. Tableaux et Espaces

3. Réduction de la Dimensionnalité

4. Classification Non Supervisée

5. Classification Supervisée

NoteBook

Environnement de développement fondé sur :

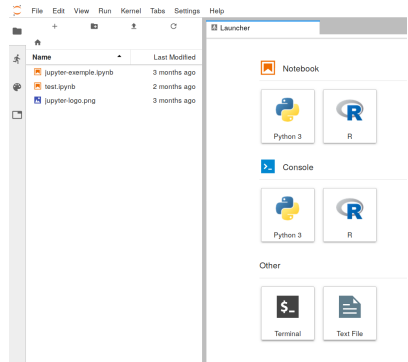
- un navigateur web où on code
- un noyau Python qui tourne dans un terminal à qui on soumet le code

Jupyter Notebook

- disponible dans *Anaconda*,
sinon `> pip install jupyter`

- énoncés de TP en Markdown

- alternative : JupyterLab
`> jupyter lab`



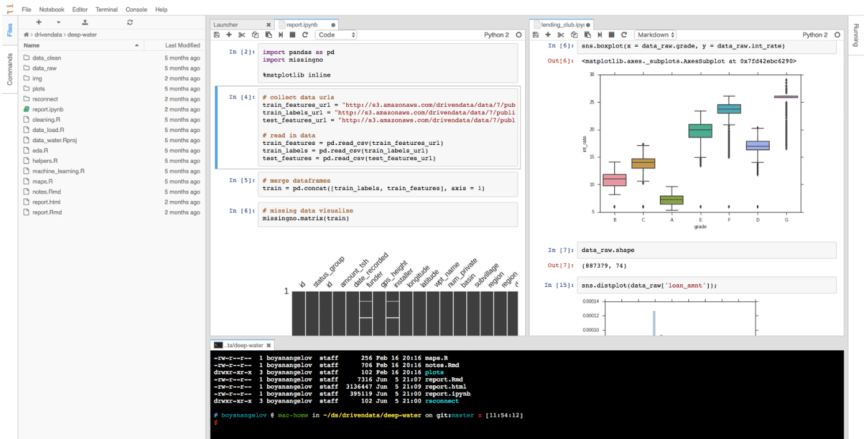
NoteBook

Environnement de développement fondé sur :

- un navigateur web où on code
- un noyau Python qui tourne dans un terminal à qui on soumet

JupyterLab

<https://python.guillod.org/python001.html>





1. Préambule

- 1.1 Objectifs et Moyens
- 1.2 NoteBook
- 1.3 Python 3

2. Tableaux et Espaces

3. Réduction de la Dimensionnalité

4. Classification Non Supervisée

5. Classification Supervisée



Caractéristiques <https://python.sdv.univ-paris-diderot.fr/cours-python.pdf>

- portable et gratuit
- syntaxe simple, interprété .ipynb
- typage dynamique interpréteur
ex. de types natifs : float, str, bool, ... , tuple, list, dict, set conversion
- orienté objet
tout est objet (instance de classe) y compris les classes encapsulation
 - attributs ou propriétés, variables
 - méthodes ou fonctions
- modules : fichiers .py contenant des classes, des fonctions, des variables import
- paquets : répertoires modules spécialisés pouvant contenir des modules

Types structurés

- un **tuple** rassemble plusieurs objets dans un ordre indexé numériquement, objet non modifiable
- une **list** est un **tuple** modifiable
- dans un **dict** les éléments sont indexés par des clés
- les **comprehension list** sont construites de manière itérative

```
In [1]: tuple = (1,3,True,7.5,'tuple')  
tuple[2]
```

```
Out[1]: True
```

```
In [2]: liste = [1,3,True,7.5,'liste']  
liste[2]=2  
liste.insert(3,4)  
print(liste)
```

```
[1, 3, 2, 4, 7.5, 'liste']
```

```
In [3]: numbers = [1,2,3,7,10]  
carres = [x**2 for x in numbers if x%2==1]  
print(carres)
```

```
[1, 9, 49]
```



Modules et Paquets

- **pyplot** de **matplotlib**
- **scipy.linalg**, **.stats**, etc
- **numpy**

graphiques
méthodes numériques, statistiques, etc
tableaux numériques multi-dimensionnels

Tableaux, Matrices et Vecteurs

- **numpy.array**
.shape, **.zeros**, **.eye**, **.diag**, ... , **.T**, **.dot**

```
In [5]: print(A.mean(axis=0))      [3. 2.]
         print(np.cov(A,ddof=1))   [[0.5 0.5]
                                   [0.5 0.5]]
```

```
In [4]: import numpy as np
         A = np.array([[2, 1],[4, 3]])
         x = np.array([2,1]).T
         Ax = np.dot(A,x)
         print(Ax)

[ 5 11]
```

Types structurés

- un **tuple** rassemble plusieurs objets dans un ordre indexé numériquement, objet non modifiable
- une **list** est un **tuple** modifiable
- dans un **dict** les éléments sont indexés par des clés
- les **comprehension list** sont construites de manière itérative

```
In [1]: tuple = (1,3,True,7.5,'tuple')
         tuple[2]
```

Out[1]: True

```
In [2]: liste = [1,3,True,7.5,'liste']
         liste[2]=2
         liste.insert(3,4)
         print(liste)
```

[1, 3, 2, 4, 7.5, 'liste']

```
In [3]: numbers = [1,2,3,7,10]
         carres = [x**2 for x in numbers if x%2==1]
         print(carres)
```

[1, 9, 49]



1. Préambule
 - 1.1 Objectifs et Moyens
 - 1.2 NoteBook
 - 1.3 Python 3
2. Tableaux et Espaces
3. Réduction de la Dimensionnalité
4. Classification Non Supervisée
5. Classification Supervisée