

Data Mining

IRC

John Samuel

CPE Lyon

Year: 2020-2021

Email: john(dot)samuel(at)cpe(dot)fr



Objectifs

- Représentation, manipulation, prétraitement et visualisation de données
- Traitement de données
- Construction des modèles de traitement

Composition du module:

- Cours: 12h
- Travaux pratiques et projet: 16h

Environnement de

- Système d'exploitation: Linux
 - Ubuntu
 - Machine virtuelle (VirtualBox) + Ubuntu
- Éditeur: Jupyter
- Logiciels: pandas, matplotlib, scikit-learn

Cours:

- Interactifs
- Les questions: chaque 20-30 mins
- Devoir surveillé: 60%

Travaux pratiques et Projet

- Projet: 40%
- 3 travaux pratiques et projet
- Programmation en binôme
- Soumission en ligne

Cours	Dates
Cours 1 (4h)	1 mars
Cours 2 (4h)	2 mars
Cours 3 (2h)	3 mars
Cours 4 (2h)	3 mars

Travaux pratiques	Dates
TP 1	4 mars
TP 2 et Projet	15 mars
TP 3 et Projet	16 mars
TP 4 et Projet	17 mars

Travaux pratiques



Soumission: Travaux pratiques et Projet

TP	Points
TP 1	✗
TP 2	✗
TP 3	✗
Projet	✓





Travaux pratiques

Chaque exercice a un niveau de difficulté

- ★: Facile
- ★★: Difficulté moyenne
- ★★★: Difficile

Liste de contrôle

Avant de déposer votre projet, vérifiez si vous respectez la liste de contrôle suivante:

-  Les noms (prénom et noms) de la binôme sont présents dans le fichier CONTRIBUTORS
-  Votre rapport est complet avec toutes les sections requises.
-  Votre code est bien commenté.
-  Votre code peut être exécuté sans aucune erreur (et si possible, sans aucun avertissement).

Modèle de code

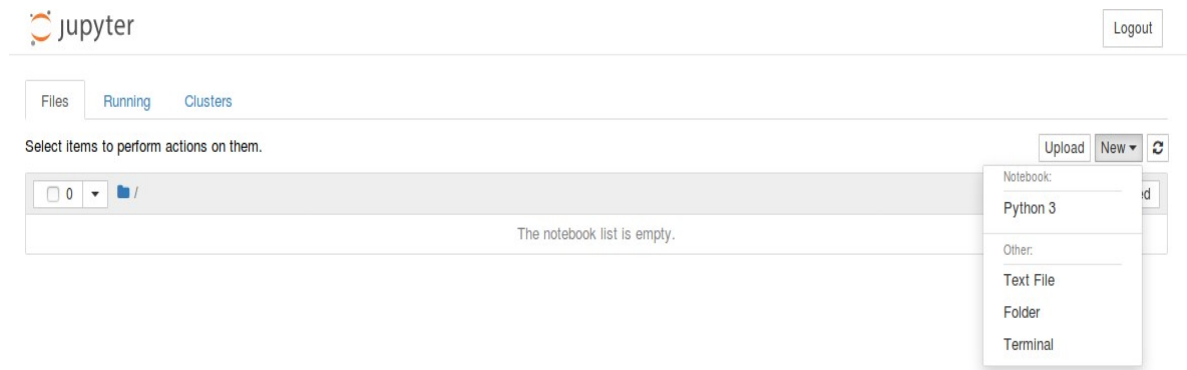
Vous pouvez consulter <https://github.com/johnsamuelwrites/MachineLearning> en ligne ou le cloner sur votre machine à l'aide du terminal en utilisant les commandes suivantes.

```
$ git clone https://github.com/johnsamuelwrites/MachineLear
$ cd MachineLearning
$ ls
```

Et pour les dernières modifications:

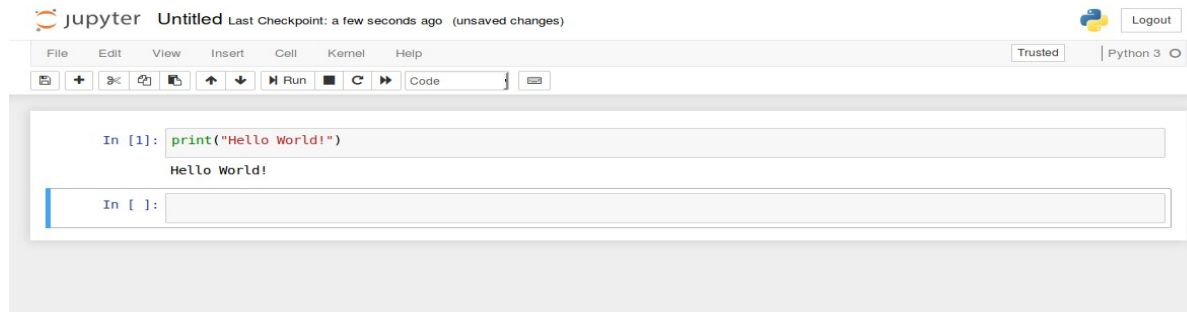
```
$ git pull
```

Travaux pratiques: Notebooks Jupyter



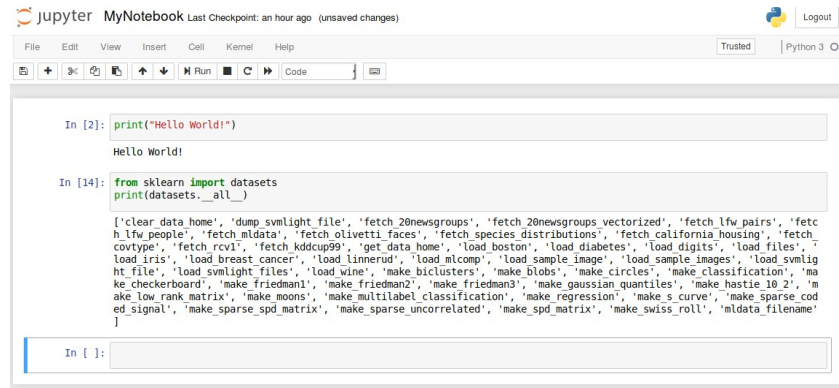
Data Mining: Notebooks Jupyter

Travaux pratiques: Notebooks Jupyter



Data Mining: Notebooks Jupyter

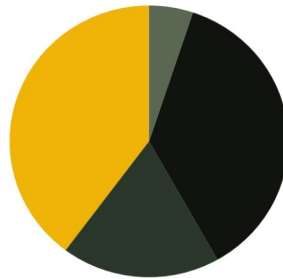
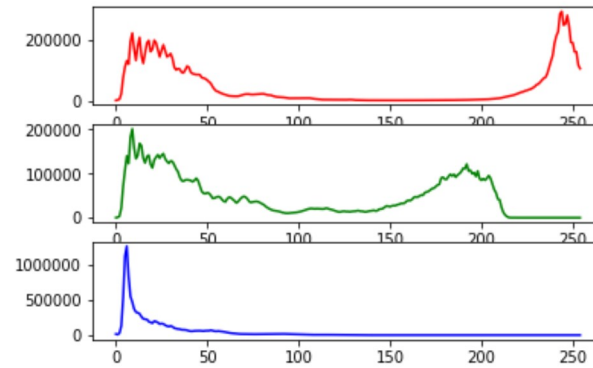
Travaux pratiques: notebook Jupyter



The screenshot shows a Jupyter Notebook titled "MyNotebook" with a status bar indicating "Last Checkpoint: an hour ago (unsaved changes)". The interface includes a menu bar (File, Edit, View, Insert, Cell, Kernel, Help) and a toolbar with icons for file operations, cell navigation, and execution. The notebook contains two code cells:

```
In [2]: print("Hello World!")  
  
Hello World!  
  
In [14]: from sklearn import datasets  
         print(datasets.__all__)  
  
['clear_data_home', 'dump_svmlight_file', 'fetch_20newsgroups', 'fetch_20newsgroups_vectorized', 'fetch_lfw_pairs', 'fetch_lfw_people', 'fetch_mldata', 'fetch_olivetti_faces', 'fetch_species_distributions', 'fetch_california_housing', 'fetch_covtype', 'fetch_rcv1', 'fetch_kddcup99', 'get_data_home', 'load_boston', 'load_diabetes', 'load_digits', 'load_files', 'load_iris', 'load_breast_cancer', 'load_linnerud', 'load_mldata', 'load_sample_image', 'load_sample_images', 'load_svmlight_file', 'load_svmlight_files', 'load_wine', 'make_biclusters', 'make_blobs', 'make_circles', 'make_classification', 'make_checkerboard', 'make_friedman1', 'make_friedman2', 'make_friedman3', 'make_gaussian_quantiles', 'make_hastie_10_2', 'make_low_rank_matrix', 'make_moons', 'make_multilabel_classification', 'make_regression', 'make_s_curve', 'make_sparse_coded_signal', 'make_sparse_spd_matrix', 'make_sparse_uncorrelated', 'make_spd_matrix', 'make_swiss_roll', 'mldata_filename']
```

Travaux pratiques: Visualisation et notebook Jupyter



Travaux pratiques: Visualisation et notebook Jupyter



Travaux pratiques: Wikidata (Open Data)

The screenshot shows the Wikidata Query interface. At the top, there's a header with the Wikidata logo, 'Wikidata Query', and buttons for 'Examples', 'Help', and 'Tools'. Below this is a 'Query Helper' sidebar on the left with a 'Filter' section set to 'instance of' and 'programming language', and a 'Show' section with 'Limit 100'. The main area displays a SPARQL query:

```
1 SELECT ?languageLabel (YEAR(?inception) as ?year)
2 WHERE
3 {
4   #instances of programming language
5   ?language wdt:P31 wd:Q9143;
6   wdt:P571 ?inception;
7   rdfs:label ?languageLabel.
8   FILTER(lang(?languageLabel) = "en")
9 }
10 ORDER BY ?year
11 LIMIT 100
```

Below the query editor, there's a status bar indicating '100 results in 123 ms' and buttons for '</> Code', 'Download', and 'Link'. The results table shows the following data:

languageLabel
ENIAC coding system
ENIAC Short Code
Von Neumann and Goldstine graphing system

A 'Download' dropdown menu is open, showing options: 'JSON file', 'JSON file (verbose)', 'TSV file', 'TSV file (verbose)', and 'CSV file'.

Sites web

- <https://jupyter.org/>
- <https://www.wikidata.org/>

Couleurs

- [Color Tool - Material Design](#)

Images

- [Wikimedia Commons](#)