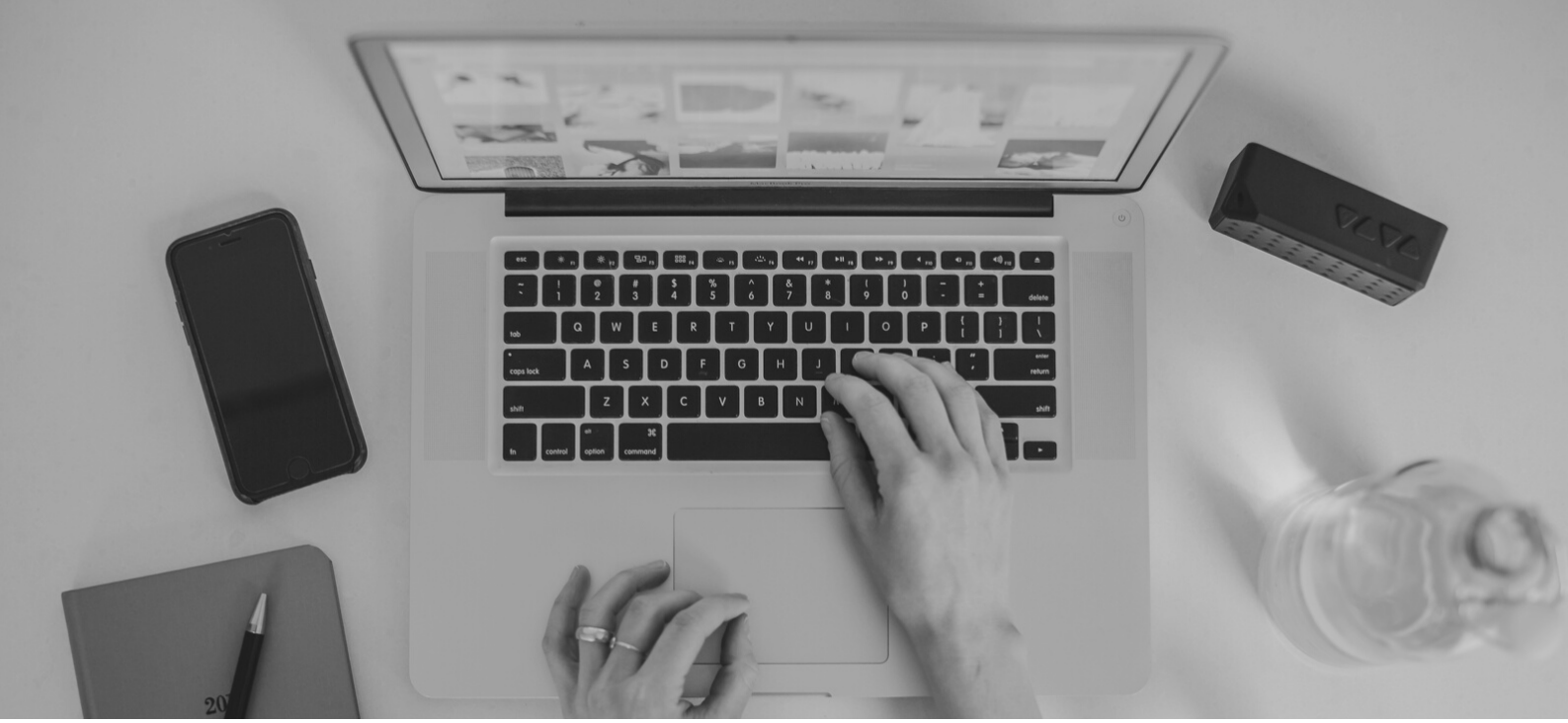
OCT 2022

Guide Jupyter Notebook

Mode d'emploie technique + Bonus

PRÉPARÉ PAR : IBRAHIM LAHLOU



Sommaire

- Introduction
- Avantages de l'utilisation de jupyter
- Installation
- L'interface de Jupyter notebook
- Raccourcie Clavier
- Bonus (Thèmes, Extension,Latex)
- Alternative online (Colab,kaggle...)
- Extraction des données
- Conda Commands
- Sécurité des Notebooks

Introduction

Jupyter Notebook et JupyterLab sont des **applications Web** open source développées par Project Jupyter. Jupyter Notebooks est centré sur la documentation alors que JupyterLab est un environnement de développement. Selon la page Web officielle,

Jupyter Notebook vs Jupyter Lab

- JupyterLab est le dernier environnement de développement interactif basé sur le Web pour les blocs-notes, le code et les données. Son interface flexible permet aux utilisateurs de configurer et d'organiser des flux de travail dans les domaines de la science des données, du calcul scientifique, du journalisme informatique et de l'apprentissage automatique.
- Le Jupyter Notebook est l'application Web originale pour la création et le partage de documents informatiques. Il offre une expérience simple, rationalisée et centrée sur le document.
- Dans ce documents on parlera de **Jupyter Notebook**

Avantage de l'utilisation de jupyter Notebooks

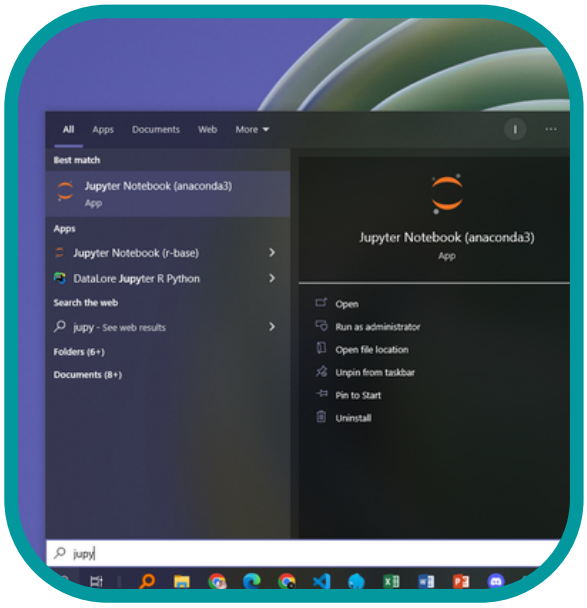
- Il s'agit d'une application Web qui s'exécute sur n'importe quel navigateur populaire.
- Vous pouvez facilement combiner du texte, des graphiques, de l'audio et de la vidéo dans vos documents.
- Il vous permet de documenter facilement en utilisant la syntaxe markdown pour la mise en forme du texte brut et la composition LaTeX pour le formatage mathématique.
- JupyterLab peut être utilisé comme outil de prototypage rapide.
- La fonctionnalité de codage interactif vous permet d'écrire, d'exécuter et de déboguer rapidement des codes.
- C'est un outil efficace pour préparer et partager des notes de cours pour un enseignement interactif.
- Il vous permet de partager efficacement vos résultats de recherche.
- Il assure la reproductibilité.
- Il peut être facilement hébergé via GitHub ou sur votre site Web.

Anaconda offers the easiest way to perform Python/R data science and machine learning on a single machine. Start working with thousands of open-source packages and libraries today.

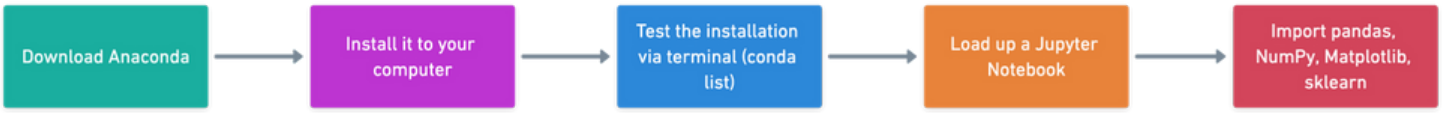
Download

Installation via Anaconda

- L'installation est simple et ce fait en exécutant le fichier d'installation pour tous les OS
- Vous pouvez l'ouvrir en tapant jupyter dans la barre de recherche
- L'utilisation de jupyter sur anaconda est mieux pour une meilleur installation de package par exemple le noyau irkernel pour utiliser R



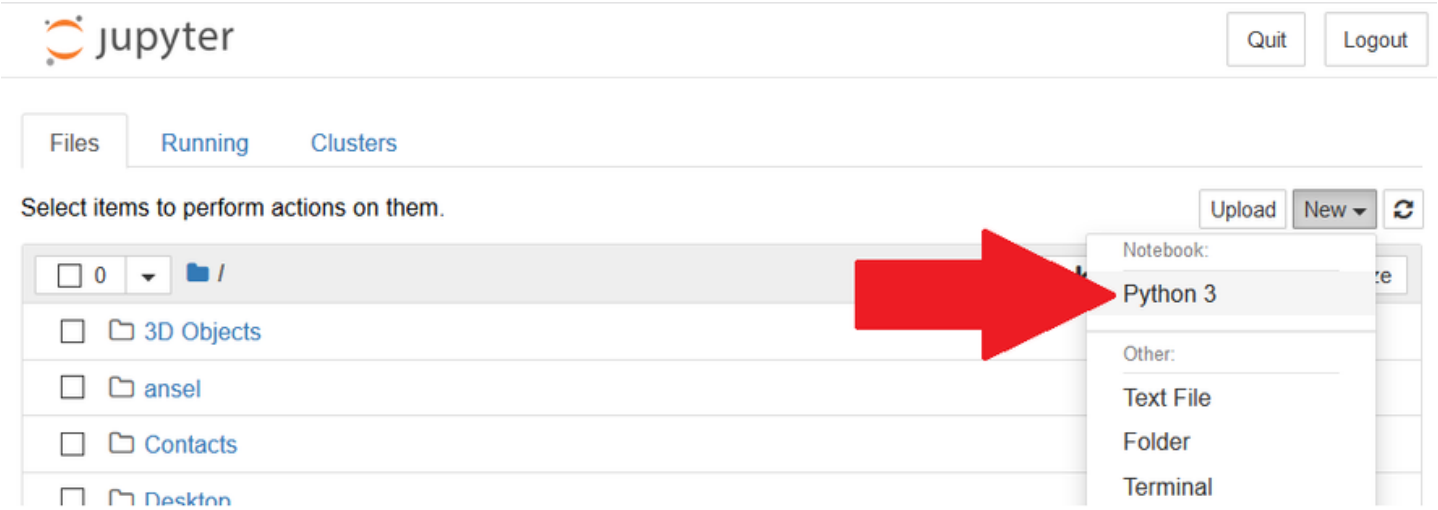
ANACONDA



- Anaconda est une distribution des langages de programmation Python et R pour le calcul scientifique qui vise à simplifier la gestion et le déploiement des paquets. La distribution inclut des packages de science des données adaptés à Windows, Linux et macOS
- Il existe également une petite version bootstrap d'Anaconda appelée Miniconda, qui inclut uniquement conda, Python, les paquets dont ils dépendent, et un petit nombre d'autres paquets. si vous n'avez pas au moins 3Go et d'ajouter vous même quelque configuration supplémentaire

L'interface de jupyter notebook

- Le **IPython Notebook** est maintenant connu sous le nom de Jupyter Notebook.
- L'environnement Jupyter Notebook permet en effet de prendre en charge plus de 50 langages de programmation différents,*(mais leurs version ne sont pas tous mis a jour)*
- Le terme "**notebook**" est lié au caractère intrinsèque de l'outil qui permet d'écrire des petits bouts de code exécutable (appelés "**cellules**"), de les documenter pour expliquer ce qu'ils font et d'afficher les données résultant de leur exécution.
- Tout cela est stocké dans un document partageable avec d'autres utilisateurs.
- Alors après l'installation de celle-ci ouvrez Jupyter Notebook : cela va ouvrir un nouvel onglet dans votre navigateur Internet appelé Home.
- Commencez par créer un nouveau notebook en cliquant sur New puis Python 3 Dans un répertoire que vous avez créé vous même (pour l 'organisation):



https://nbviewer.org/

Raccourcie Clavier

Jupyter: Modes & Cell Types		
2 Jupyter Modes		
	Edition Mode	[Intro] to Edition mode
	Command Mode	[Esc] to Command Mode
3 Cell Types		
	Markdown *texts or formats*	
	Raw text, no format	
	Program code e.g. Python, grey cell, `In [n]` before	

Command Mode [ESC] key	
↩	Go to **Edition Mode**
^ ↩ [ctrl] ↩	Run Cell
⇧ ↩	Run Cell, Select Cell Under
⌥↩ [alt]↩	Run Cell, Insert Cell Below
m	**Markdown** Cell
y	**Code** Cell
r	**Raw** Cell
1	Header 1
2	Header 2, and so on.. 3,4,5,6
↑ / ↓	Select before / after cell
k / m	Select before / after cell
a / b	Insert cell below / under
x	Cut Cell
c	Copy Cell
v	Paste Cell Under
⇧ v	Paste Cell Below
d d	Erase Cell
s / ⌘ S	Save and Checkpoint
L	Toggle Cell Numbers

Raccourcie Clavier

Command Mode [ESC] key (cont)	
o	Toggle Outputs
h	Show Help
␣ / ⇧␣	Scroll Screen Down / Up

Mac / Windows Keyboards Equivalences			
Mac	*Windows*	Mac	*Windows*
⌘	⊞	[ctrl]	Control / ^
⌥	**Option** / [alt]	⇧	[Shift]
↩	**Enter** / Return	↩	[Intro]
⇥	**Tabulation**/ [Tab]	␣	[Space]
⌘ ⊞	[ctrl] Control ^		

Note: The shortcuts may differ depending on your system configuration.

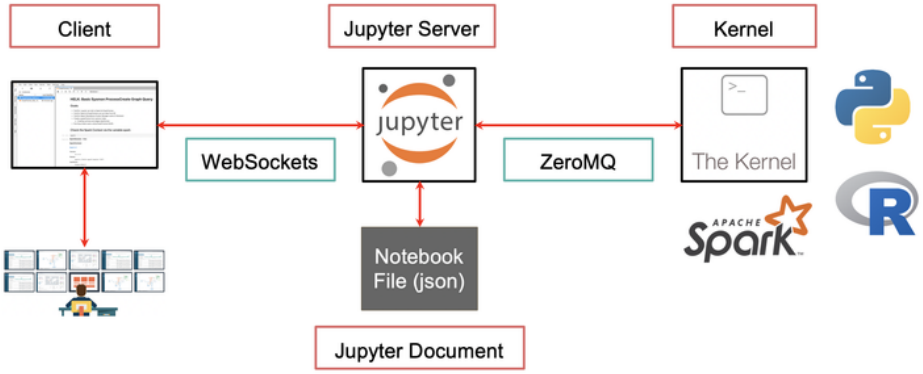
Command History Shorcuts	
[Ctrl] p or ↑	Access previous command in history
[Ctrl] n or ↓	Access next command in history
[Ctrl] r	Reverse-search through command history

Miscellaneous Shortcuts	
[Ctrl] L	Clear terminal screen
[Ctrl] C	Interrupt current Python comman
[Ctrl] D	Exit IPython session

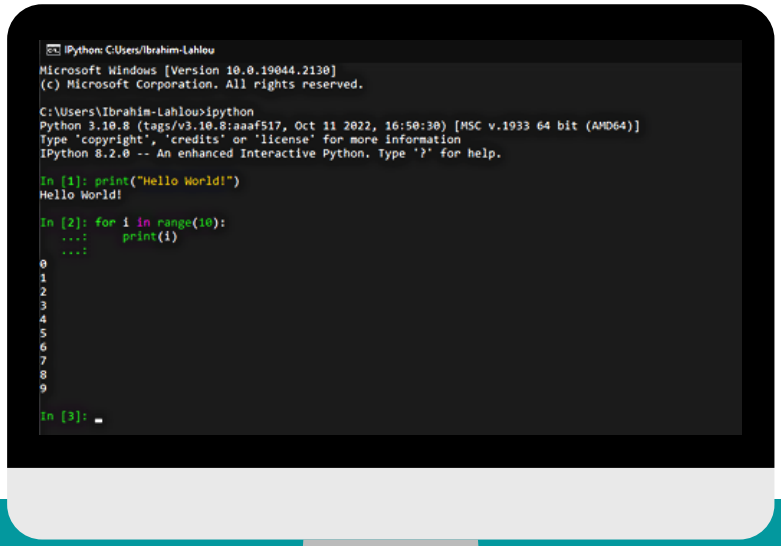
Edition Mode [Enter] / doble click		
Mac	Windows	Keys, Meaning
[Esc]	[Esc]	**Command Mode**
⌘ m	[ctrl] m	**Command Mode**
Indentation		
⇥	[tab]	Indent , Complete Code
⇧ ⇥	⇧ [tab]	Outdent
⌘]	[ctrl]]	Indent
⌘ [	[ctrl] [	Outdent
Navigation Shorcuts inside cell		
⌥ ←	[alt] ←	← 1 word to left
⌥ →	[alt] →	→ 1 word to right
⌥ ⌫	[alt] ⌫	← Erase left word

Comment fonctionne jupyter notebook pour éxecuter du code

- Les blocs-notes Jupyter fonctionnent avec ce qu'on appelle un modèle à deux processus basé sur une **infrastructure noyau-client**. Ce modèle applique un concept similaire à l'environnement de programmation REPL (Read-Evaluate-Print Loop) qui prend les entrées d'un seul utilisateur, les évalue et renvoie le résultat à l'utilisateur.
- Sur la base du concept de modèle à deux processus, nous pouvons expliquer les principaux composants de Jupyter de la manière suivante:

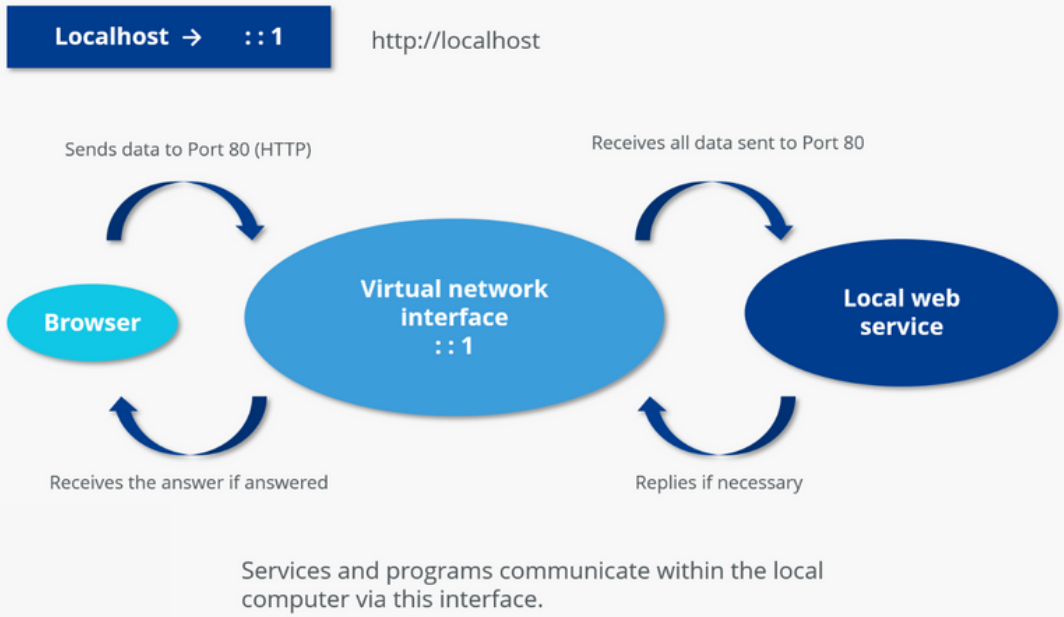


- Dans le coté client on envoie le code sous la forme d'une console QT ou dans le navigateur
- Dans le coté Noyau (le noyau IPYTHON) reçoit le code l'exécute et renvoie les résultats au client pour affichage. Un processus du noyau peut avoir plusieurs clients communiquant avec lui, c'est pourquoi ce modèle est également appelé modèle à deux processus découplé.
- Vous pouvez testez le noyau **ipython** sur le terminal



Comment fonctionne jupyter coté serveur

Lorsque vous lancez une application via un serveur, cela se fait généralement à l'aide d'un serveur web, tel qu'Apache ou Nginx. Le serveur web reçoit les demandes des clients (par exemple, les navigateurs web) et les transmet à l'application. L'application traite alors la demande et renvoie une réponse au serveur web, qui à son tour renvoie la réponse au client.



Dans le cas d'Anaconda, certaines des applications incluses dans la distribution, comme Jupyter Notebook, peuvent être exécutées en tant que serveur web. Lorsque vous lancez Jupyter Notebook, il démarre un serveur web sur votre machine locale, et l'interface du notebook est servie par ce serveur web. Vous pouvez ensuite vous connecter au serveur du notebook à l'aide de votre navigateur Web en visitant l'URL http://localhost:8888 (ou http://127.0.0.1:8888) .

Si vous souhaitez exécuter une application sur un serveur distant et y accéder depuis une autre machine, vous pouvez le faire en configurant le serveur Web pour qu'il écoute sur une adresse IP publique. Par exemple, vous pouvez configurer Jupyter Notebook pour écouter sur une adresse IP publique en le démarrant avec la commande suivante dans l'interface de commande d'anaconda :

```
jupyter notebook --ip=0.0.0.0
```

Comment fonctionne jupyter coté serveur

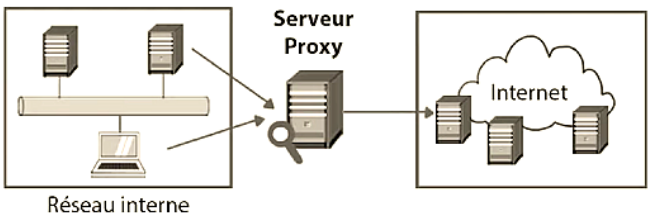
Cela fera en sorte que le serveur du bloc-notes écoute sur toutes les interfaces réseau disponibles, y compris toute adresse IP publique attribuée au serveur. Vous pouvez ensuite accéder au bloc-notes à partir d'une autre machine en visitant l'URL L http//serverip-address:8888 dans un navigateur Web.

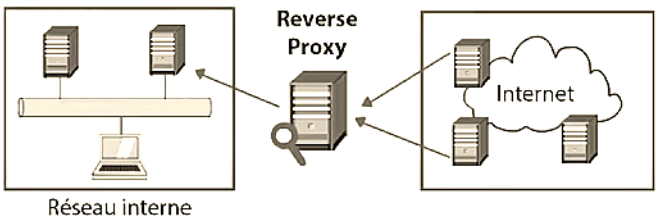
Une autre option consiste à utiliser un proxy inverse, c'est-à-dire un serveur qui se trouve devant votre serveur d'application et qui transmet les demandes entrantes à l'application. Cela peut étre utile pour acheminer les demandes vers différentes applications en fonction de l'URL, sinsi que pour fournir des fonctionnalités supplémentaires telles que l'équilibrage de charge (Load balancer) et la terminaison SSL

Bonus c'est quoi le proxy

Un proxy est un serveur qui agit en tant qu'intermédiaire entre un ordinateur client et les serveurs distants auxquels il souhaite accéder. Il peut être utilisé pour des raisons de sécurité, de performance, de filtrage de contenu et de contrôle d'accès.

Un data scientist peut utiliser un proxy pour Sécurité ,Performance,Filtrage de contenu ,Contrôle d'accès et Accès à des données géographiquement restreintes
Il est important de noter que l'utilisation d'un proxy peut nécessiter une configuration supplémentaire et une maintenance supplémentaire, il est donc important de consulter la documentation du proxy et de bien comprendre les risques et les avantages liés à son utilisation.





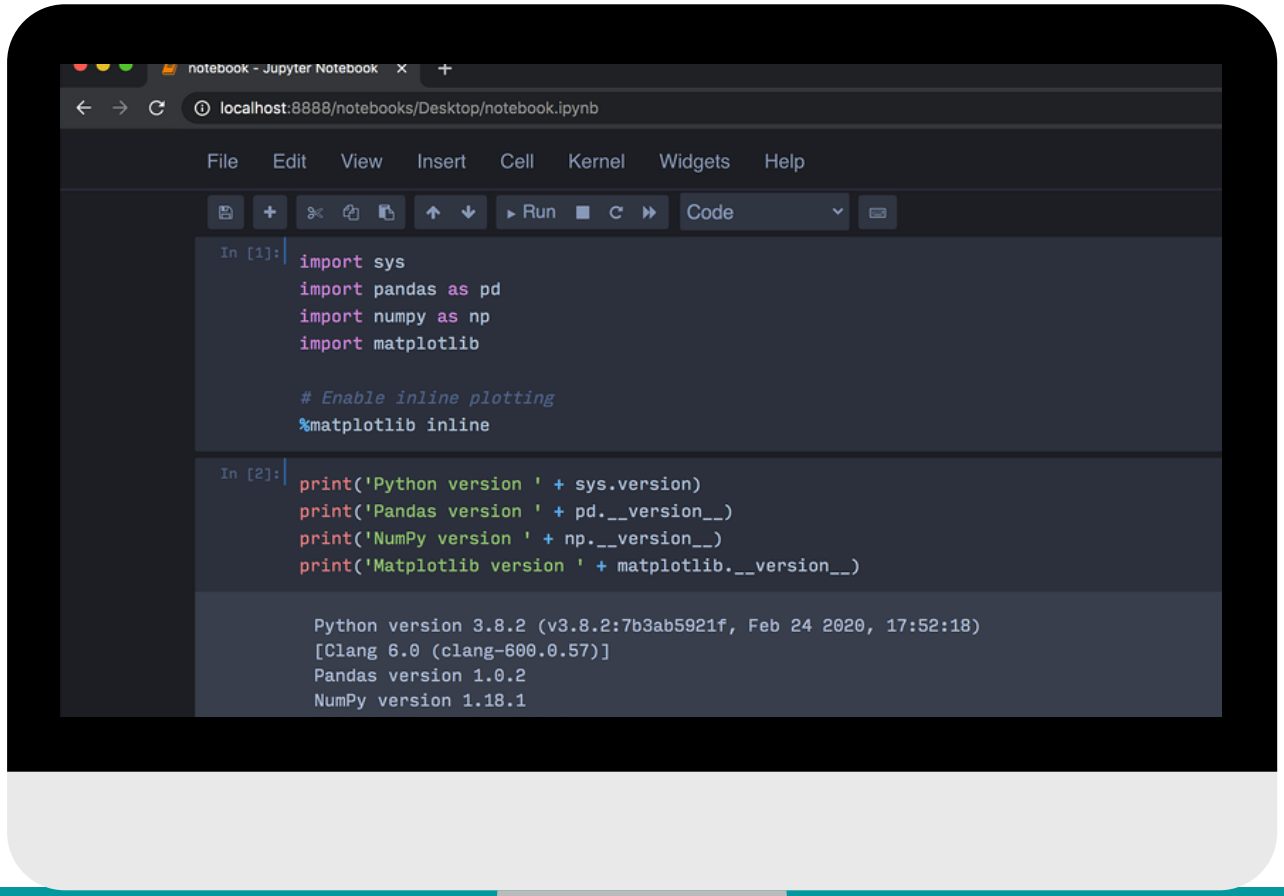
Les thèmes dans Jupyter notebook (Bonus)

- Vous pouvez créer des thèmes personnalisés pour Jupyter Notebook à l'aide de jupyther-themes.
- Installer avec pip ou conda

```
# conda
conda install -c conda-forge jupyterthemes
```

- Personnalisez le thème avec des options données. Ma configuration :

```
jt -t onedork -f inputmono -fs 11 -ofs 11 -cellw 88% -T
```



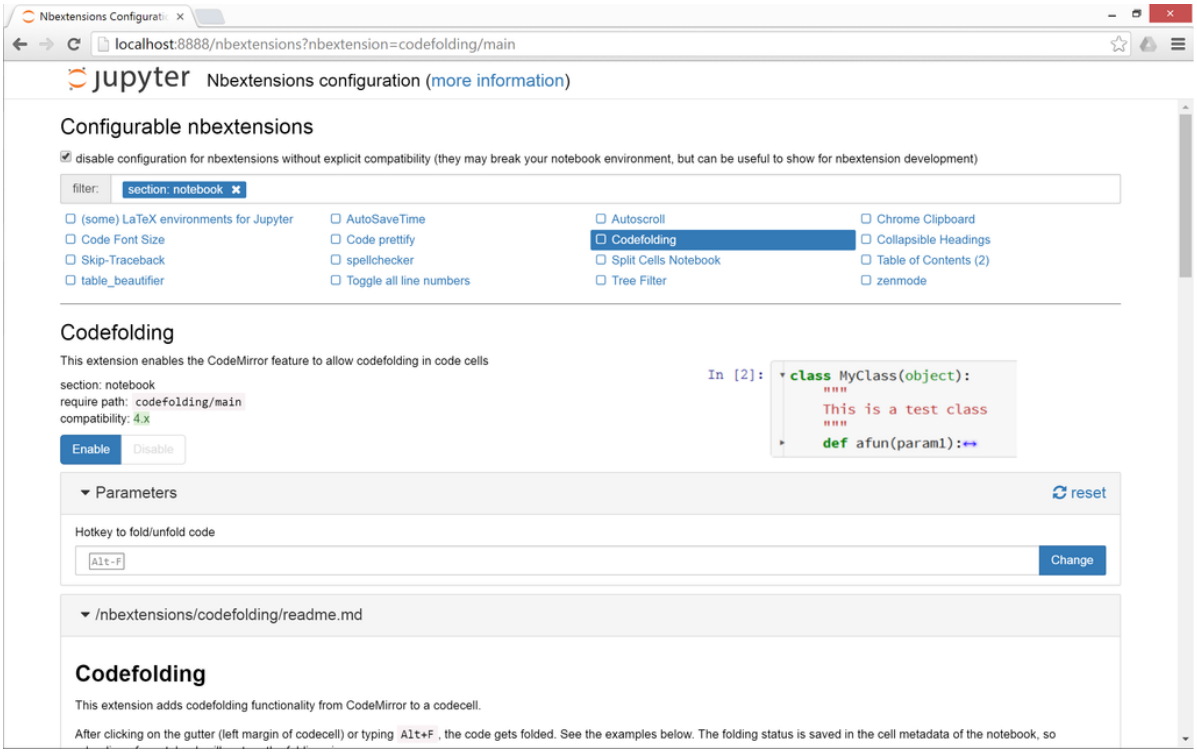
Les extensions dans Jupyter notebook (Bonus)

- Il existe des packages conda pour les notebook extensions et les jupyter_nbextensions_configurators disponibles auprès de conda-forge. Vous pouvez installer les deux à l'aide de ces commandes vérfier que vous n'avez pas un antivirus qui ne laisse pas l'installation s'execute

```
pip install jupyter_contrib_nbextensions
pip install jupyter_nbextensions_configurator
```

- Cela installe également automatiquement les fichiers Javascript et CSS (en utilisant), de sorte que la deuxième étape d'installation ci-dessous peut donc être ignorée.

```
jupyter contrib nbextension install --user
jupyter nbextension_configurator enable --user
```



- Pour plus d'informations sur les extension vous pouvez voir le site ci-dessous:

https://www.tabnine.com/blog/top-12-jupyter-notebook-extensions/

Astuce jupyter notebook's Markdown

`#␣Header 1`
 	
`Header 1`
`========`
	# Header 1

`##␣Header 2`
 	
`Header 2`
`--------`
	## Header 2

`###␣Header 3`	### Header 3
`####␣Header 4`	#### Header 4
`#####␣Header 5`	##### *Header 5*
`*italics*` `_italics_`	*italics*
`\*literal asterisks\*`	*literal asterisks*
`**bold**` `__bold__`	**bold**
`~~strikethrough~~`	~~strikethrough~~

`1.␣First item`
`2.␣Second item`
`␣1.␣Subitem`
	1. First item
	2. Second item
	  A. Subitem

`*␣Item 1`
`␣Indent`
`-␣Item 2`
`␣+␣Item 3`
	• Item 1
	  Indent
	• Item 2
	  ▪ Item 3

`- [x] Done` `- [ ] To do`	• ☑ Done • ☐ To do

`A`
`Line␣␣`
`Break`
	A Line
	Break

`---` `* * *`	---

Astuce jupyter notebook's Markdown

<a id="anchor"></a> [Go to anchor](#anchor) # Top Header [Go to header](#Top-Header)	Go to anchor
https://sqlbak.com [Link](https://sqlbak.com "optional title") Click [here][id] [id]:https://sqlbak.com	Link
> blockquote text	blockquote text

``python
print('hello');
``
 	
`inline_code();`
	`print('hello');`

|Left |Center|Right| |:-----|:----:|----:| |1 |A |C | |2 |B |D |	Left Center Right 1 A C 2 B D
![alt text](logo.png "Title") ![][id] [id]:logo.png "Title"	
$$\sqrt{k}$$ Inline: $\sqrt{k}$	$\sqrt{k}$
[![Img Alt Text](http://img.youtube.com/vi/aZCXOw707nc/0.jpg)](https://youtu.be/aZCXOw707nc "Video Title")	YouTube

l'utilisation de Latex dans jupyter notebook (Bonus)

- LaTeX est un logiciel de haute qualité pour la préparation de documentations scientifiques. Il se compose de macros TeX et de programmes pour traiter les documents techniques en latex.
- Maintenant, Latex est disponible gratuitement dans la plupart des éditeurs de code source en ligne. L'un des plus populaires est Jupyter Notebook qui utilise MathJax pour rendre la syntaxe Latex à l'intérieur du markdown/HTML.
- Pour utiliser LaTeX dans le bloc-notes Jupyter, placez le contenu mathématique Latex à l'intérieur des symboles '$ ... $' double '$$ ... $$'. Le fichier latex est enregistré avec l'extension (.tex).

- MathJax est une bibliothèque multi-navigateurs JavaScript qui affiche des notations mathématiques ou scientifiques dans les navigateurs Web en utilisant le balisage Latex et MathML.
- Vous n'avez pas besoin d'installer ou de télécharger des plugins supplémentaires pour vous aider à écrire des documents Web et à inclure des mathématiques à l'aide de JavaScript.
- Pour l'installer :

```
conda install mathjax
```

- Pour ajouter des équation mathématique tout d'abord utilisez les symboles dollar $$
- Les notations suivantes vous aident à écrire des équations mathématiques de base dans Latex jupyter notebook:
  - On mode mathématique, utilisez le symbole **\,** pour ajouter un peu d'espacement.
  - Pour ajouter une nouvelle ligne, utilisez ****
  - Pour afficher les valeurs de fraction, utilisez **\frac{value 1}{value 2}**
  - Pour les exposants ou l'alimentation, utilisez **^{}**
  - Pour ajouter des indices ou des index, utilisez **_{}**
  - Pour prendre la racine nième d'une valeur, utilisez **\sqrt{n}{value}**

- Pour ajouter des lettres grecques utilisez \ puis :

```
$$\sigma, \Sigma \\
\delta, \Delta \\
\omega, \Omega \\
\gamma, \Gamma \\
\lambda, \Lambda \\
\theta, \Theta$$
```

$$\sigma, \Sigma \\ \delta, \Delta \\ \omega, \Omega \\ \gamma, \Gamma \\ \lambda, \Lambda \\ \theta, \Theta$$

- Pour ajouter des dérivés

```
$$f'(x) \\
f''(x) \\
f^{(k)}(x) \\
\frac{\partial^{k} f}{\partial x^k}$$
```

$$f'(x) \\ f''(x) \\ f^{(k)}(x) \\ \frac{\partial^{k} f}{\partial x^k}$$

- Pour ajouter les sommes et les intégrales

```
$$\sum \\
\sum\limits_{j=1}^k A_{\alpha_j} \\
\sum_{j=1}^k A_{\alpha_j} \\
\sum^k_{i=1}\sum^l_{j=1}\,q_i q_j \\
\int_{a}^b f(x)dx \\
\int_{a}^b\int_{c}^d f(x,y)dxdy
$$
```

$$\sum \\ \sum\limits_{j=1}^k A_{\alpha_j} \\ \sum_{j=1}^k A_{\alpha_j} \\ \sum^k_{i=1}\sum^l_{j=1}\,q_i q_j \\ \int_{a}^b f(x)dx \\ \int_{a}^b\int_{c}^d f(x,y)dxdy$$

- Pour ajouter des accolades pour les système d'équation par exemple

```
$$
sign(y) = \left\{
    \begin{array}\\
        2 & \mbox{if } \ y \in \mathbf{Z}^* \\
        0 & \mbox{if } \ y = 0 \\
        -2 & \mbox{else.}
    \end{array}
\right.
$$
```

$$sign(y) = \left\{ \begin{array}\\ 2 & \mbox{if } \ y \in \mathbf{Z}^* \\ 0 & \mbox{if } \ y = 0 \\ -2 & \mbox{else.} \end{array} \right.$$

- Pour ajouter les matrices

```
$\begin{matrix} a&b\\ c&d \end{matrix} \\
\begin{pmatrix} a&b\\ c&d \end{pmatrix} \\
\begin{vmatrix} a&b\\ c&d \end{vmatrix} \\
\begin{Vmatrix} a&b\\ c&d \end{Vmatrix} \\
\begin{bmatrix} a&b\\ c&d \end{bmatrix} \\
\begin{Bmatrix} a&b\\ c&d \end{Bmatrix}$
```

$\begin{matrix} a&b\\ c&d \end{matrix} \\ \begin{pmatrix} a&b\\ c&d \end{pmatrix} \\ \begin{vmatrix} a&b\\ c&d \end{vmatrix} \\ \begin{Vmatrix} a&b\\ c&d \end{Vmatrix} \\ \begin{bmatrix} a&b\\ c&d \end{bmatrix} \\ \begin{Bmatrix} a&b\\ c&d \end{Bmatrix}$

- Pour ajouter des probabilités

```
$A \cup B \\
A \cap B = \varnothing \\
\overline{A} \\
\mathrm{P}(A)=0,15 \\
\mathrm{P}_B(A)=0,5$A \ B $A \setminus B \\
p(A\cap B)=p(A)\times p_A(B) \\
\displaystyle\binom{n}{p} \\
\dbinom{n}{p} \\
p(B)=p(A)\times p_A(B)+% \\
p(\overline{A})\times p_{\overline{A}}(B)$
```

$A \cup B \\ A \cap B = \varnothing \\ \overline{A} \\ \mathrm{P}(A)=0,15 \\ \mathrm{P}_B(A)=0,5$A \ B $A \setminus B \\ p(A\cap B)=p(A)\times p_A(B) \\ \displaystyle\binom{n}{p} \\ \dbinom{n}{p} \\ p(B)=p(A)\times p_A(B)+% \\ p(\overline{A})\times p_{\overline{A}}(B)$

- **Logic**

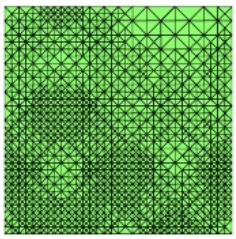
description	command	output
not	\sim	$\sim$
and	\land	$\land$
or	\lor	$\lor$
if...then	\to	$\to$
if and only if	\leftrightarrow	$\leftrightarrow$
logical equivalence	\equiv	$\equiv$
therefore	\therefore	$\therefore$
there exists	\exists	$\exists$
for all	\forall	$\forall$
implies	\Rightarrow	$\Rightarrow$
equivalent	\Leftrightarrow	$\Leftrightarrow$

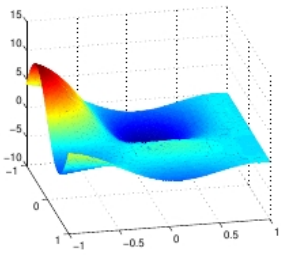
- **Set Theory**
- Pour ajouter les opérateur ensembliste et logique : (les ensembles s'annote ainsi \mathbb{...})

description	command	output
element of	\in	$\in$
not an element of	\not\in	$\notin$
subset of	\subset	$\subset$
subset of	\subseteq	$\subseteq$
not a subset of	\not\subset	$\not\subset$
contains	\supset	$\supset$
contains	\supseteq	$\supseteq$
union	\cup	$\cup$
intersection	\cap	$\cap$
big union	\bigcup_{n=1}^{10}A_n	$\bigcup_{n=1}^{10} A_n$
big intersection	\bigcap_{n=1}^{10}A_n	$\bigcap_{n=1}^{10} A_n$
empty set	\emptyset	$\emptyset$
power set	\mathcal{P}	$\mathcal{P}$

- **Insert figure**

```
\begin{figure}[H]
\begin{center}
\includegraphics[width=0.2\textwidth]{fig.eps}
\caption{simulation result }\label{fig:pro2}
\end{center}
\end{figure}
```





- **Draw Figure**

```
\begin{figure}[H]
\begin{center}
\begin{tikzpicture}[scale=0.5,>=latex]
\shade[ball color=gray,opacity=0.50] (0,0) circle (2cm);
\draw[->] (0,0,0) -- (3,0,0) node (xaxis) [right] {$x$};
\draw[->] (0,0,0) -- (0,3,0) node (zaxis) [right] {$z$};
\draw[->] (0,0,0) -- (0,0,5) node (yaxis) [left] {$y$};
\draw [->] (0,0,0)-- (2,0,2)node[right] {$_{r\sin\phi}$};
\draw [blue,->] (0,0,0)-- (2,2,2)node[right] {$r$};
\draw [blue,->] (0.5,0.5,0.5)to[bend right](0,0.5,0);
\node[] at (0.3,0.6,0)   {$_\phi$};
\draw[blue,densely dashed] (2,0,2)-- (2,2,2)--(0,1.8,0);
\draw[blue,densely dashed](2,0,2)--(1.7,0,0)node[above]{$x$};
\draw[blue,densely dashed](2,0,2)--(0,0,1.9)node[left]{$y$};
\coordinate (x0) at (2,0,2);
\draw [blue,->] (0,0,0.5)to[bend right](0.4,0,0.4);
\node[] at (0.3,0,1)   {$_\theta$};
\end{tikzpicture}
\caption{Spherical coordinate in $\mathbb{R}^3$}
\end{center}
\end{figure}
```

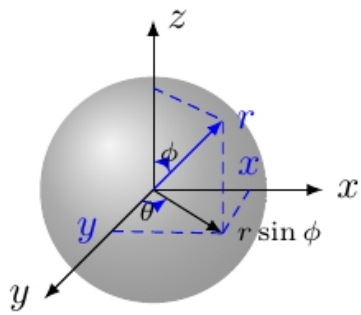


Figure 2: Spherical coordinate in $\mathbb{R}^3$

- **Table**

```
\begin{table}[H]
\caption{Errors of FEM for Poisson Equation}\label{tab:p2}
\begin{center}
\begin{tabular}{ccc}
\hline $h$&$\norm{u-u_h}_{L^2}$ & $\abs{u-u_h}_{H^1}$\\
\hline $1/4$& $7.62\times 10^{-3}$ &$1.02\times10^{-1}$
\\$1/8$& $2.02\times 10^{-3}$ &$5.58\times10^{-2}$
\\ $1/16$ & $5.32\times 10^{-4}$ &$3.01\times10^{-2}$
\\ $1/32$& $1.37\times 10^{-4}$ &$1.60\times10^{-2}$
\\ $1/64$& $3.52\times 10^{-5}$ &$8.51\times10^{-3}$
\\ $1/128$& $ 8.87\times 10^{-6}$ &$4.48\times10^{-3}$
\\ $1/256$& $2.20\times 10^{-6}$ &$2.35\times10^{-3}$
\\\hline
\end{tabular}
\end{center}
\end{table}
```

Table 1: Errors of FEM for Poisson Equation

h	$\|u-u_h\|_{L^2}$	$\|u-u_h\|_{H^1}$
$1/4$	7.62×10^{-3}	1.02×10^{-1}
$1/8$	2.02×10^{-3}	5.58×10^{-2}
$1/16$	5.32×10^{-4}	3.01×10^{-2}
$1/32$	1.37×10^{-4}	1.60×10^{-2}
$1/64$	3.52×10^{-5}	8.51×10^{-3}
$1/128$	8.87×10^{-6}	4.48×10^{-3}
$1/256$	2.20×10^{-6}	2.35×10^{-3}

- **Algorithms**

```
\begin{algorithm} [H]
\caption{Bisection method}
\begin{algorithmic}[1]
\State $a_0\gets a, b_0\gets b$
\While{$k >0$}
   \State $c_k\gets \frac{a_{k-1}+b_{k-1}}{2}$
\If {$f(a_k)f(c_k)<0$}
    \State $a_k \gets a_{k-1}$
    \State $b_k\gets c_k$
 \EndIf
\If {$f(b_k)f(c_k)<0$}
    \State $a_k \gets c_k$
    \State $b_k\gets b_{k-1}$
 \EndIf
\State $x^k\gets c^k\gets \frac{a_{k}+b_{k}}{2}$
\EndWhile
\end{algorithmic}
\end{algorithm}
```

Algorithm 1 Bisection method

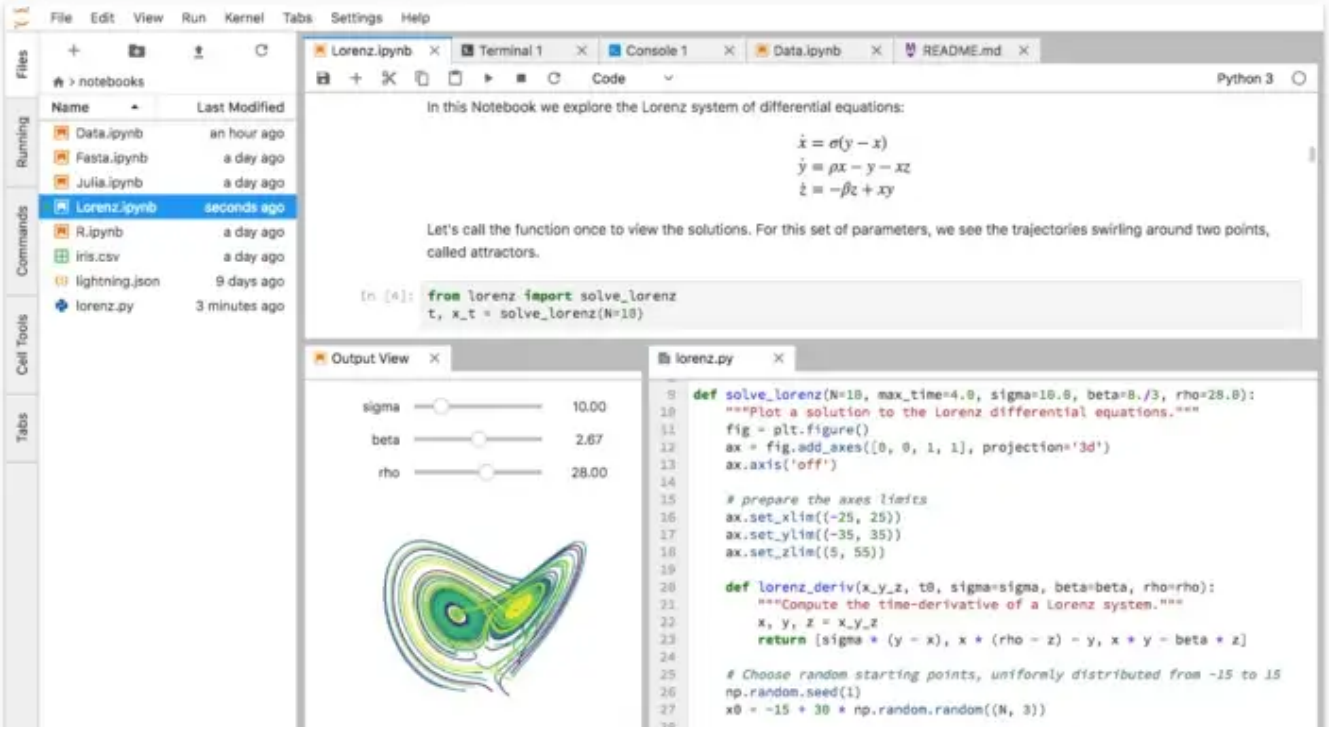
```
a_0 ← a, b_0 ← b
while k > 0 do
    c_k ← (a_{k-1}+b_{k-1})/2
    if f(a_k)f(c_k) < 0 then
        a_k ← a_{k-1}
        b_k ← c_k
    end if
    if f(b_k)f(c_k) < 0 then
        a_k ← c_k
        b_k ← b_{k-1}
    end if
    x^k ← c^k ← (a_k+b_k)/2
end while
```

- Pour plus de details concernant l'utilisation de latex voir le pdf :

http://math.univ-lyon1.fr/irem/IMG/pdf/LatexPourLeProfDeMaths.pdf

Technologie plus général que jupyter notebook

JupyterLab est un environnement de développement interactif permettant de travailler avec des notebooks, du code et des données. Plus important encore, JupyterLab prend entièrement en charge les notebooks Jupyter. En outre, JupyterLab vous permet d'utiliser des éditeurs de texte, des terminaux, des visualiseurs de fichiers de données et d'autres composants personnalisés en même temps que les notebooks dans une zone de travail à onglets.



JupyterLab offre un haut niveau d'intégration entre les notebooks, les documents et les activités :

Glissez-déposez pour réorganiser les cellules des notebooks et les copier entre les notebooks.
Exécutez des blocs de code de manière interactive à partir de fichiers texte (.py, .R, .md, .tex, etc.).
Liez une console de code à un noyau de bloc-notes pour explorer le code de manière interactive sans encombrer le bloc-notes avec du travail temporaire.
Modifiez les formats de fichiers les plus courants avec un aperçu en direct, tels que Markdown, JSON, CSV, Vega, VegaLite, etc.
La création de JupyterLab a duré plus de trois ans, avec plus de 11 000 commits et 2 000 versions des paquets npm et Python. Plus de 100 contributeurs de la communauté au sens large ont participé à la construction de JupyterLab, en plus de nos développeurs JupyterLab.

Pour commencer, consultez la documentation de JupyterLab pour obtenir des instructions d'installation et une présentation détaillée, ou essayez JupyterLab avec Binder. Vous pouvez également configurer JupyterHub pour utiliser JupyterLab.

Visual Studio Code

VS Code IDE recommandé

- Visual Studio Code permet de travailler avec les Notebooks Jupyter de manière native et par le biais de fichiers de code Python. Cette rubrique couvre la prise en charge native disponible pour les Notebooks Jupyter et montre comment :
  - créer, ouvrir et enregistrer des notebooks Jupyter
  - Travailler avec des cellules de code Jupyter
  - Afficher, inspecter et filtrer des variables à l'aide de l'Explorateur de variables et du Visualiseur de données
  - Se connecter à un serveur Jupyter distant
  - Déboguer un bloc-notes Jupyter
- Et ainsi développer des notebooks Jupyter dans VS Code peut se faire entièrement via une interface web utilisant GitHub Codespaces, un environnement de développement hébergé dans le cloud qui est sécurisé et configurable avec des ressources de calcul gratuites (plus d'informations sur les quotas d'utilisation mensuels de Codespaces). Cela signifie que vous pouvez profiter de toutes les fonctionnalités de VS Code, y compris la prise en charge des notebooks Jupyter, sans rien installer sur votre machine.

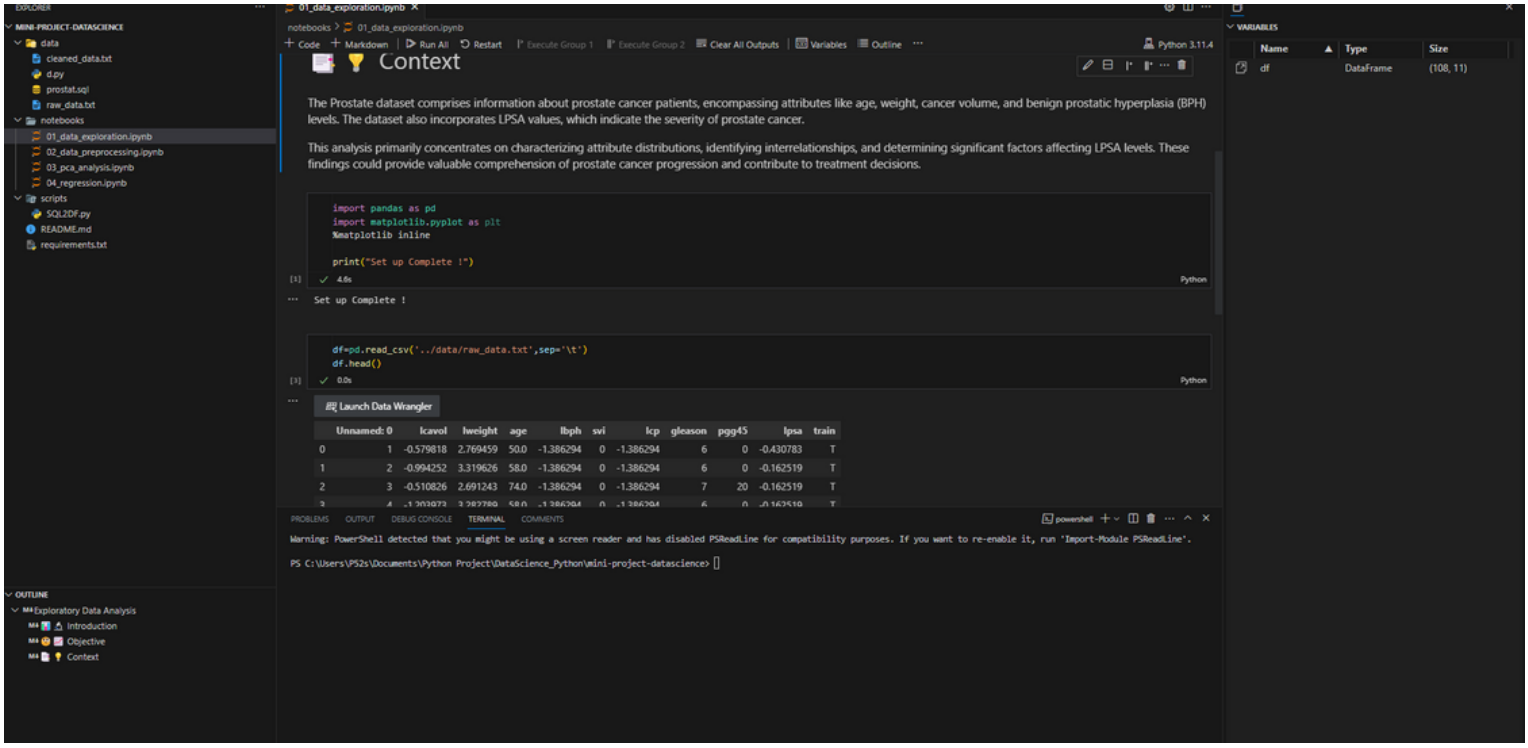
Visual Studio Code

VS Code IDE recommandé

Lorsque vous travaillez sur des projets d'apprentissage automatique avec les notebooks Jupyter dans VSCode, il existe plusieurs extensions et techniques qui peuvent grandement améliorer votre productivité et votre expérience de développement. Voici quelques étapes et conseils pour maximiser vos résultats :

Extensions Essentielles pour VSCode Notebooks :

- **Extensions de Machine Learning** : Explorez les extensions conçues spécifiquement pour les projets de machine learning tels que "Jupyter" ou "Python". Elles apportent des fonctionnalités supplémentaires et une meilleure intégration avec les notebooks.
- **Tabnine pour l'Autocomplétion** : Intégrez l'extension TabNine pour bénéficier d'une autocomplétion intelligente. Cette extension s'appuie sur des modèles d'apprentissage automatique pour anticiper et suggérer intelligemment du code, accélérant ainsi votre flux de travail.
- **Extensions de Thème :** Personnalisez l'apparence de VSCode en utilisant des extensions de thème. Choisissez un thème visuel tel que "Material Theme" pour donner à VSCode une allure moderne et agréable.



Visual Studio Code

VS Code IDE recommandé

Flexibilité des Langages et Gestion des Versions :

- **Multi-Language** : Profitez de la polyvalence offerte par VSCode. Vous n'êtes pas limité à un seul langage de programmation. Vous pouvez facilement travailler avec différents langages, notamment R et Python, dans le même environnement.
- **Gestion des Versions via venv ou Anaconda** : Pour maintenir des environnements de développement propres, utilisez des environnements virtuels (venv) ou, de manière plus avancée, Anaconda. Anaconda vous permet de créer des environnements isolés avec des versions spécifiques de packages pour chaque projet.

Intégration Efficace avec Anaconda :

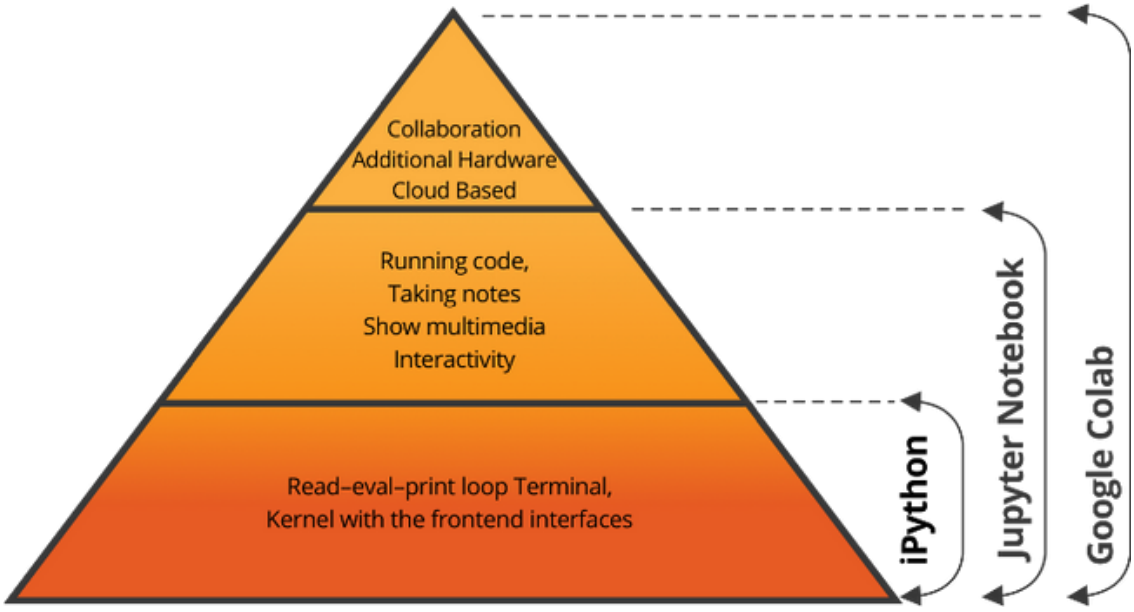
1. **Installer Anaconda** : Si ce n'est pas déjà fait, installez Anaconda, un outil puissant de gestion de packages et d'environnements.
2. **Naviguer vers le Répertoire de Projet** : Ouvrez un terminal et utilisez la commande **cd** pour accéder au répertoire où se trouve votre projet ML.
3. **Activer l'Environnement** : Utilisez la commande **conda activate <nom_de_l'environnement>** pour activer l'environnement conda spécifique à votre projet.
4. **Lancer VSCode** : Dans le même terminal, exécutez la commande **code .** pour ouvrir VSCode avec l'environnement conda actif.

ANACONDA. tabnine

Colaboratory

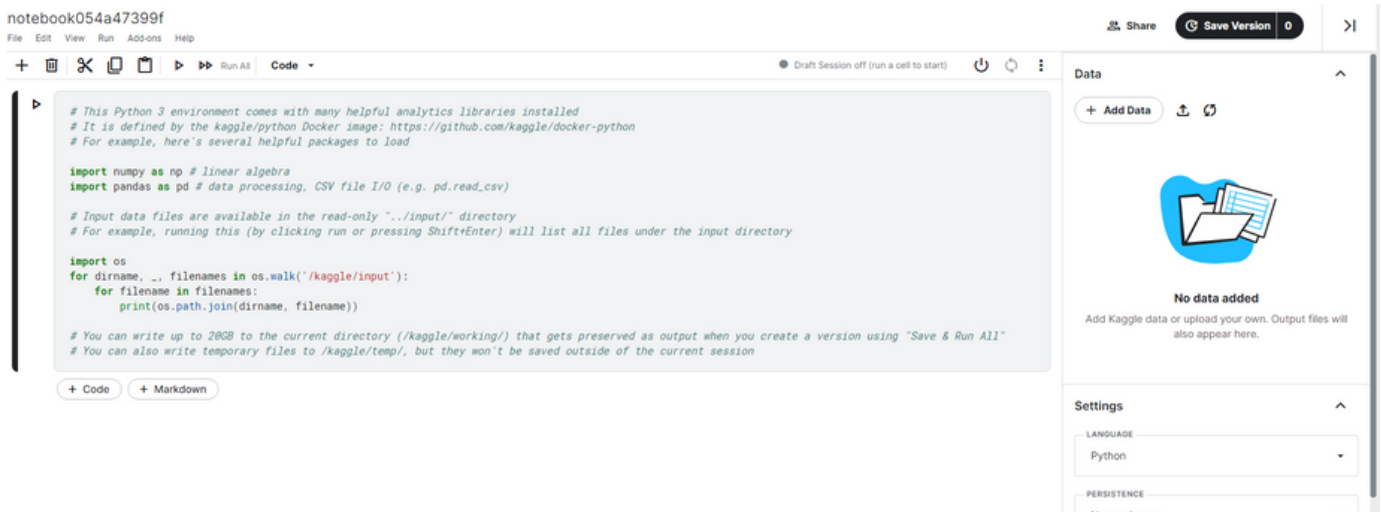
Google Colab une solution en ligne

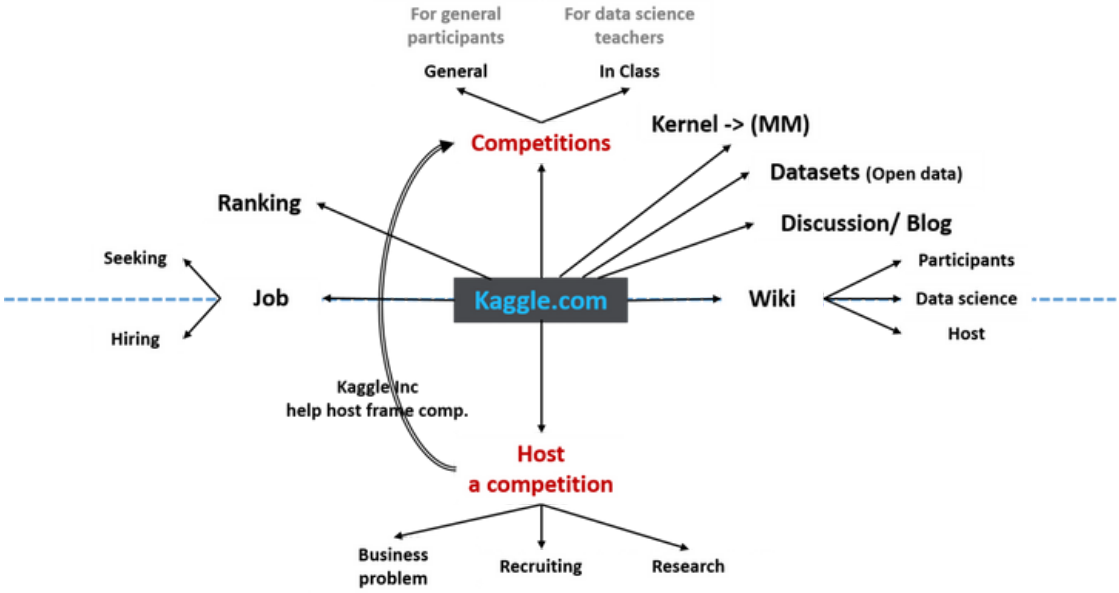
- Colaboratory, ou Colab en abrégé, est un produit de Google Research Open source, qui permet aux développeurs d'écrire et d'exécuter du code Python via leur navigateur. Google Colab est un excellent outil pour les tâches d'apprentissage en profondeur.
- Il s'agit d'un bloc-notes Jupyter hébergé qui ne nécessite aucune configuration et dispose d'une excellente version gratuite, qui donne un accès gratuit aux ressources informatiques de Google telles que les GPU et les TPU.



Kaggle

- Kaggle est une plateforme web organisant des compétitions en science des données appartenant à Google. Sur cette plateforme, les entreprises proposent des problèmes en science des données et offrent un prix aux datalogistes obtenant les meilleures performances. L'entreprise a été fondée en 2010 par Anthony Goldbloom





Datalore

Collaborative no-code analytics joins forces with scalable infrastructure

Datalore

- Datalore est une plateforme collaborative de Data Science .
- Elle est disponible dans un cloud géré pour une utilisation personnelle sur les plans Community et Professional, ou en tant que service hébergé privé pour les entreprises sur le plan Enterprise.

Smart Python, Kotlin, Scala, and R notebooks

Datalore is Jupyter-compatible and comes with smart coding assistance for Python, Kotlin, Scala, and R. Each notebook has an isolated environment, integrated environment manager, internal versioning, and integrated storage.

Learn more

Database connections and SQL cells

Enterprise Professional

In Datalore you can connect to a wide range of databases from the interface, browse database schema, and then query your data with native SQL cells. You can start working with SQL and then seamlessly continue to work with the query result using Python in the same notebook.

Learn more

Teamwork

In Datalore you can collaborate on code in real time. You can share notebooks simply using a link or email invitation and set the level of access as either view or edit. You can also organize team projects in workspaces and share the data with your whole team using workspace files.

Learn more

Reporting

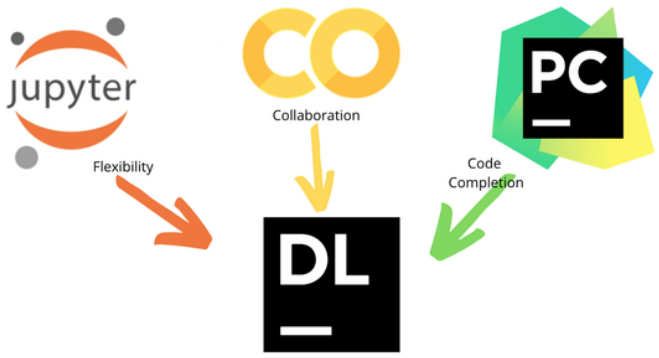
Turn your notebooks into interactive data stories with a single click. You can hide code cells from viewers and share static or interactive reports with your stakeholders by simply providing them with a link to the report.

Learn more

On-premises installations Enterprise

Datalore can be hosted in a private cloud and on-premises by companies that require extra customization and data security. The Enterprise plan includes custom configurations for SSO, environment configuration and computational resources.

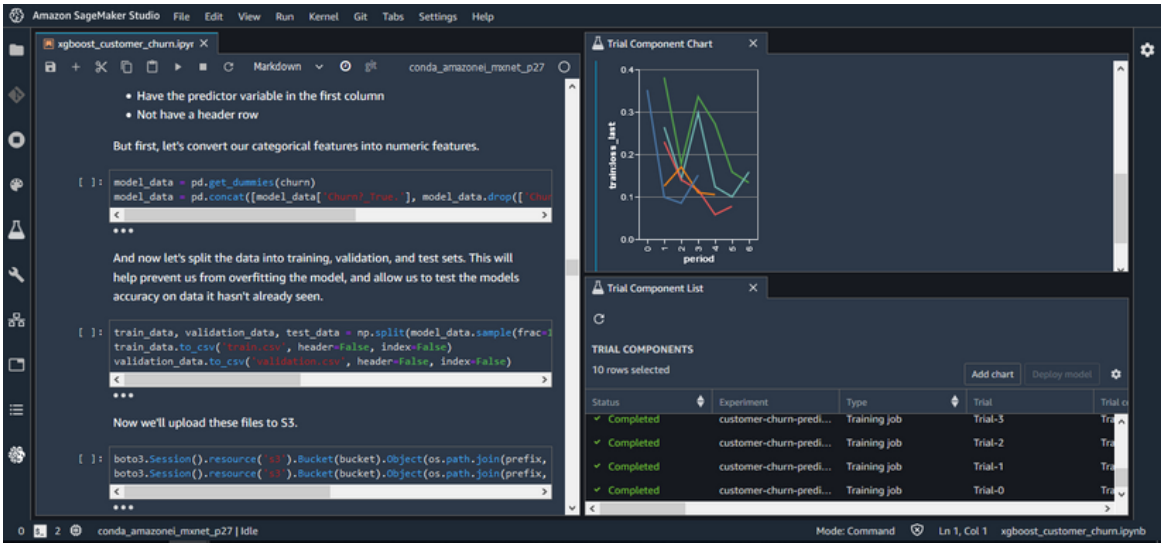
Learn more

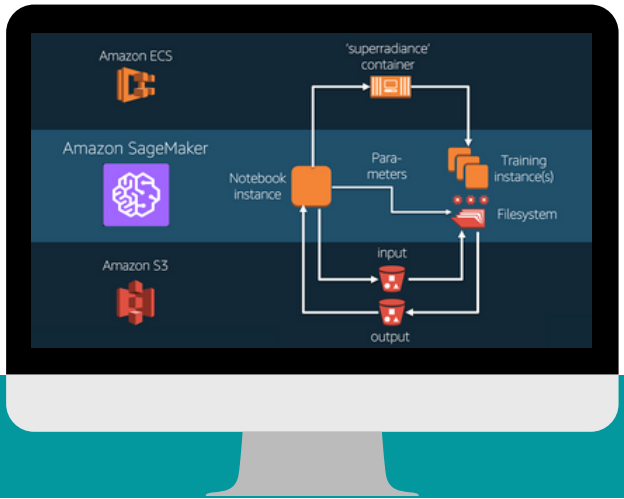


Amazon SageMaker

SageMaker une autre alternative pour les utilisateurs d'aws

- Amazon Web Services (AWS), le service de cloud computing de Amazon.com, est le plus grand fournisseur de services cloud au monde. À partir de ses centres de données, l'entreprise fournit plus de 200 services complets, notamment le calcul, le stockage et la base de données.
- Amazon SageMaker est une **plateforme de machine learning cloud** lancée en novembre 2017.
- Il fournit une instance de bloc-notes de création Jupyter intégrée pour un accès facile à vos sources de données à des fins d'exploration et d'analyse, ce qui vous évite d'avoir à gérer les serveurs. Il fournit également des algorithmes d'apprentissage automatique courants optimisés pour s'exécuter efficacement sur des données extrêmement volumineuses dans un environnement distribué.





Amazon SageMaker

Home / Products / Azure Machine Learning

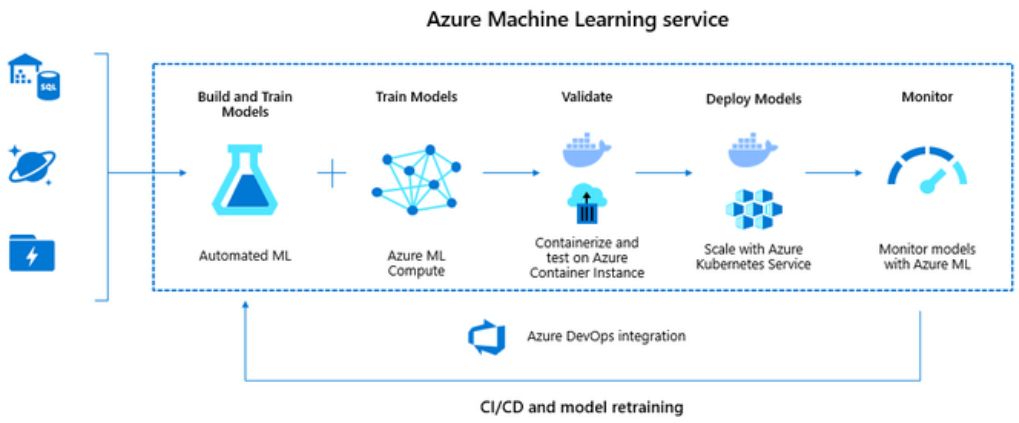
Azure Machine Learning

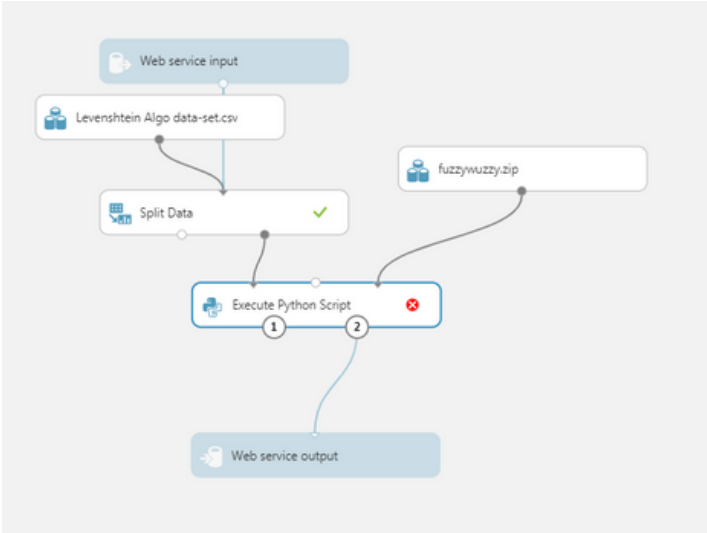
Use an enterprise-grade service for the end-to-end machine learning lifecycle

Try Machine Learning free

Azure ML une autre alternative pour les utilisateurs de Microsoft

- Azure Machine Learning permet aux scientifiques des données et aux développeurs de créer, déployer et gérer des modèles de haute qualité plus rapidement et en toute confiance. Il accélère le délai de rentabilisation grâce à des opérations d'apprentissage automatique (MLOps) de pointe, à l'interopérabilité open source et à des outils intégrés. Cette plateforme de confiance est conçue pour les applications d'IA responsables en apprentissage automatique.





Azure Machine Learning

Different façon pour extraire les Données with Jupyter notebook

File.txt

File.csv

File.xlsx

```
with open('File.txt') as f:
    lines = f.readlines()

import csv
with open('File.csv', 'r') as f:
    csv_reader = csv.reader(f)

import xlwings as xw

ws = xw.Book("File.xlsx").sheets['Sheet1']
```

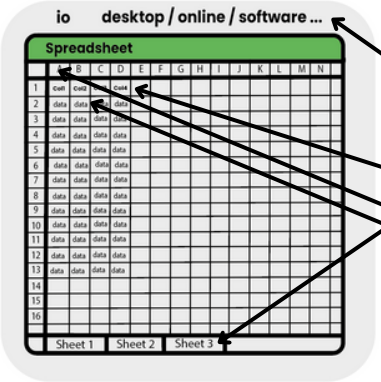
```
import pandas as pd

df1 = pd.read_csv('File.csv')
```

io : str, bytes, csvfile path object, or file-like object
sep: délimiteur de champ, par défaut ','
header : int default (l'entête du fichier csv)
squeeze : bool, default False (pas conseillé)
parse_dates : indique quelles colonnes doivent être converties en objets de type date.
date_parser : pour spécifier quelles colonnes doivent être converties en objets de type date

```
df2 = pd.read_excel('File.xlsx')
```

io : str, bytes, ExcelFile, xlrd.Book, path object, or file-like object
sheet_name : str, int, list, or None, default 0 (c'est le nome de la feuille)
header : int, list of int, default (l'entête de la feuille)
index_col : int, list of int, default None
usecols : str, list-like, or callable, default None
squeeze : bool, default False (pas conseillé)
converters : dict, default None



Vous pouvez utiliser la fonction **pd.read_sql()** pour lire des données d'une base de données directement dans un DataFrame, puis utiliser les fonctions de Pandas pour manipuler les données.

Pour se connecter à une base de données, vous pouvez utiliser **SQLAlchemy** qui est une bibliothèque de mappage objet-relationnel pour Python. Il permet de créer une connexion à une base de données en utilisant un connecteur adaptateur pour un type de base de données spécifique.

```
from sqlalchemy import create_engine
import pandas as pd

# Create a connection to the database
engine = create_engine("mysql+pymysql://user:password@host/database")

# Extract data from the database and store it in a DataFrame
df = pd.read_sql("SELECT * FROM table", engine)

# Modify the data in the DataFrame
df['column'] = df['column'] * 2

# Write the modified data back to the database
df.to_sql("table",engine, if_exists='replace')
```

1. verifier que vous avez git installer dans le système ou le contenaire
2. Effectuer la commande sur git bash:

```
git clone https://github.com/username/repository.git
cd repository
```

3. Apres vous pouvez extraire le fichier file.csv avec pandas

Si votre fichier se trouve dans un server Http

```
import requests

url = "http://example.com/path/to/file.csv"
response = requests.get(url, auth=('user', 'pass'))
with open("file.csv", "wb") as f:
 f.write(response.content)
```

IP

Vous pouvez faire a la place du nom de domaine une adresse ip
Apres vous pouvez extraire le fichier file.csv avec pandas

Si votre fichier se trouve dans un server FTP

```
import ftplib

ftp = ftplib.FTP("ftp.example.com")
ftp.login("username", "password")
ftp.cwd("/path/to/file")

with open("file.csv", "wb") as f:
 ftp.retrbinary("RETR file.csv", f.write)

ftp.quit()
```

Apres vous pouvez extraire le fichier file.csv avec pandas

```
import zipfile
import pandas as pd

with zipfile.ZipFile('archive.zip', 'r') as archive:
    if'file.csv'inarchive.namelist():
         archive.extract('file.csv')
         with archive.open('file.csv') as file:
             data = pd.read_csv(file)
    else:
         print('file.csv not found')
```

Si votre file.csv se trouve dans un contenaire
Créer in dockerfile et un image docker
chercher le fichier qui sera surement ziper le déziper

```
FROM python:3.8-slim
RUN apt-get update && apt-get install -y unzip
RUN pip install pandas

docker build -t my_image .

docker run -v $(pwd)/folder:/app -it my_image

unzip file.zip
```

Apres vous pouvez extraire le fichier file.csv avec pandas dans le contenaire lui meme

Les 11 Commandes a connaître avec conda

Conda est un gestionnaire de paquets et un système de gestion de l'environnement pour Python et d'autres langages. Voici quelques commandes couramment utilisées qui peuvent être utiles lorsque vous travaillez avec Conda :

conda create : Crée un nouvel environnement conda avec un nom spécifié et éventuellement une version spécifique de Python.

```
conda create --name myenv python=3.8
```

conda activate : Active un environnement conda spécifique, le transformant en environnement courant.

```
conda activate myenv
```

conda deactivate : Désactive l'environnement Conda actuel, le rendant inactif.

```
conda deactivate
```

conda list : Liste tous les paquets installés dans l'environnement actuel.

```
conda list
```

conda install : Installe un paquet dans l'environnement actuel.

```
conda install numpy
```

conda update : Met à jour un paquet à la dernière version dans l'environnement actuel.

```
conda update numpy
```

conda update : Met à jour un paquet à la dernière version dans l'environnement actuel.

```
conda remove numpy
```

conda update : Met à jour un paquet à la dernière version dans l'environnement actuel.

```
conda info
```

conda update : Met à jour un paquet à la dernière version dans l'environnement actuel.

```
conda env export --name myenv > environment.yml
```

Débuter avec Jupyter Notebook : Astuces de Commande

Lorsque vous travaillez avec Jupyter Notebook, il existe des astuces simples pour exécuter des commandes sans chercher le terminal. Utilisez le point d'exclamation (!) suivi de chaque commande, comme ceci :

1. Installer des Packages :
Au lieu de naviguer vers le terminal, installez des packages directement dans une cellule en ajoutant !`avant la commande. Par exemple :

```
!pip install numpy
```

2. Lister des Fichiers :
Observez les fichiers dans un répertoire sans quitter votre notebook. Utilisez !ls (Linux) ou !dir (Windows) :

```
!pip install numpy
```

3. Vérifier les Versions :
Vérifiez les versions des packages installés sans ouvrir le terminal. Par exemple :

```
!pip show pandas
```

4. Git Commandes :
Pour exécuter des commandes Git, utilisez !`pour interagir avec Git depuis vos cellules :

```
!git status
```

5. Exécution de Scripts :
Exécutez des scripts en utilisant le point d'exclamation, par exemple pour Python :

```
!python script.py
```

Ces astuces vous permettent d'exécuter des commandes sans quitter votre notebook, simplifiant ainsi votre flux de travail avec Jupyter Notebook.

Sécurité dans jupyter notebook

Risques d'Exécution de Code :

- Code Vulnérable : Exécuter un code non fiable dans un Jupyter Notebook peut créer des risques de sécurité, surtout lors de partages ou d'utilisations de sources non fiables.
- Mesures : Évitez d'exécuter du code non vérifié, préférez des notebooks fiables. Isoler les environnements via des environnements virtuels ou des conteneurs peut être utile.

Exécution de Code à Distance :

- Configurations Vulnérables : Un serveur Jupyter exposé sur Internet, sans solide authentification, peut permettre l'exécution de code malveillant.
- Mesures : Privilégiez une authentification robuste. Utilisez HTTPS, des pare-feu pour limiter l'accès externe. JupyterHub est recommandé pour les environnements multi-utilisateurs.

Exposition de Données Sensibles :

- Sortie Vulnérable : Les sorties de cellules peuvent accidentellement afficher des données sensibles, comme des clés API ou des mots de passe.
- Mesures : Avant de partager, assurez-vous de nettoyer les sorties. Utilisez des outils comme nbstripout pour supprimer les sorties avant la validation.

Vulnérabilités de Paquets Non Intentiionnelles :

- Paquets Vulnérables : Les Jupyter Notebooks dépendent souvent de paquets externes. Les vulnérabilités dans ces paquets peuvent être exploitées.
- Mesures : Maintenez vos paquets à jour pour appliquer les correctifs de sécurité. Utilisez des gestionnaires de paquets comme conda ou pip.

Déni de Service (DoS) :

- Épuisement des Ressources : Des calculs complexes dans un Jupyter Notebook peuvent consommer beaucoup de ressources, entraînant un déni de service.
- Mesures : Limitez les ressources pour les processus. Des outils comme jupyter-resource-usage peuvent aider à surveiller et à contrôler l'utilisation des ressources.

Permissions Mal Configurées :

- Accès Insecure aux Fichiers : Des permissions incorrectes pour les notebooks peuvent permettre un accès non autorisé aux fichiers.
- Mesures : Configurez correctement les autorisations pour garantir un accès restreint. Évitez de partager dans des répertoires publics sans contrôles d'accès.

Mises à Jour Manquantes :

- Logiciels Obsolètes : Ignorer les mises à jour expose aux vulnérabilités connues.
- Mesures : Mettez à jour Jupyter Notebook et ses dépendances régulièrement. Abonnez-vous aux bulletins de sécurité pour rester informé des vulnérabilités.

Commande Sécurité dans jupyter notebook

1. Mise à Jour des Packages : Gardez vos packages à jour pour éviter les vulnérabilités.
2. Vérification des Packages Vulnérables : Utilisez des outils comme 'safety' pour vérifier les vulnérabilités dans vos packages.
3. Vérification de la Configuration du Serveur : Vérifiez les paramètres de configuration de votre serveur Jupyter pour des problèmes de sécurité potentiels.
4. Suppression des Sorties de Cellules : Avant de partager votre notebook, supprimez les sorties de cellules qui peuvent contenir des informations sensibles.
5. Analyse des Autorisations de Fichiers : Utilisez des commandes pour vérifier et modifier les autorisations de fichiers sensibles.

```
# Mise à Jour des Packages
pip install --upgrade package_name

# Vérification des Packages Vulnérables
pip install safety
safety check

# Vérification de la Configuration du Serveur
jupyter notebook --config-dir
cat ~/.jupyter/jupyter_notebook_config.py

# Suppression des Sorties de Cellules
jupyter nbconvert --ClearOutputPreprocessor.enabled=True --inplace
notebook.ipynb

# Analyse des Autorisations de Fichiers
ls -l
chmod 600 sensitive_file.txt
```