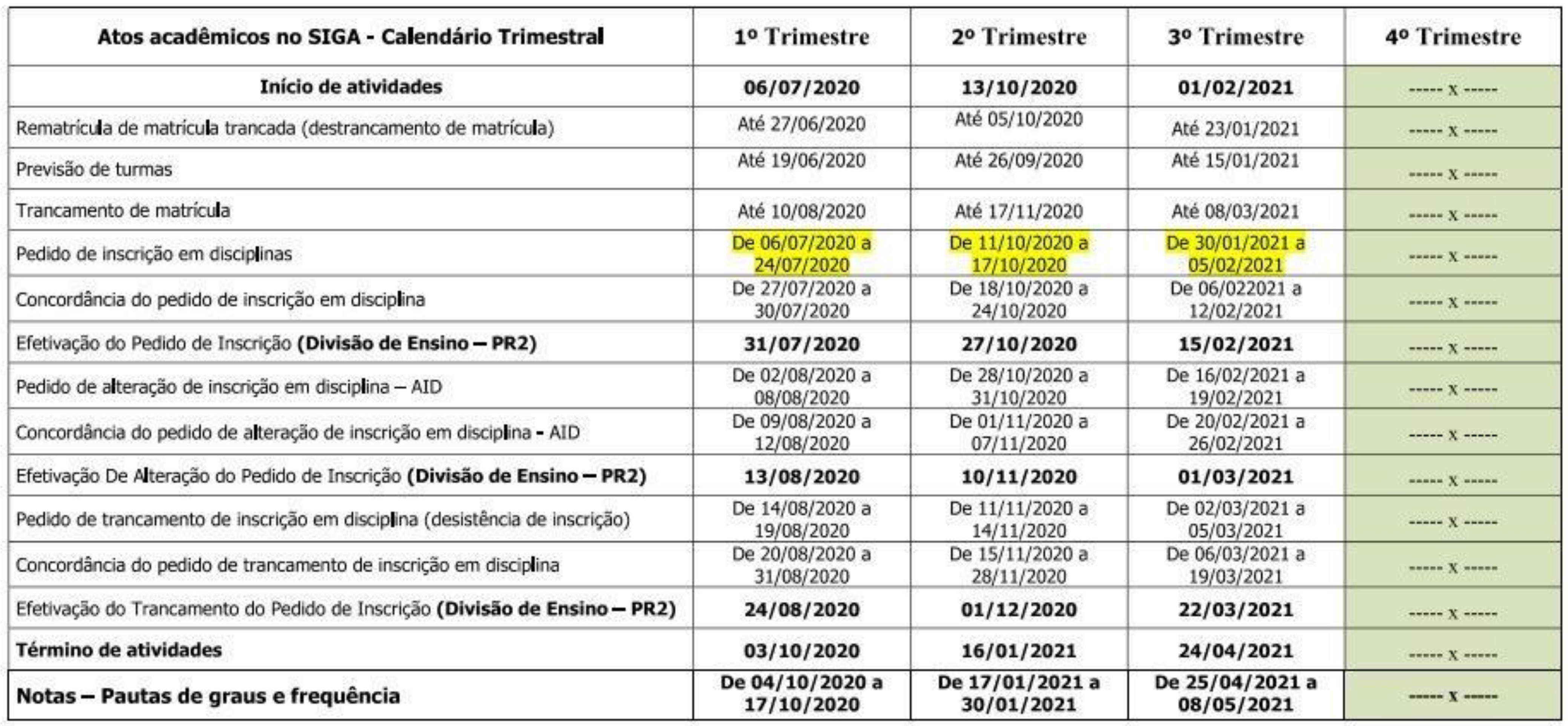
# Datas Importantes



| Atos acadêmicos no SIGA - Calendário Trimestral | 1º Trimestre | 2º Trimestre | 3º Trimestre | 4º Trimestre |
|---|---|---|---|---|
| **Início de atividades** | **06/07/2020** | **13/10/2020** | **01/02/2021** | ----- x ----- |
| Rematrícula de matrícula trancada (destrancamento de matrícula) | Até 27/06/2020 | Até 05/10/2020 | Até 23/01/2021 | ----- x ----- |
| Previsão de turmas | Até 19/06/2020 | Até 26/09/2020 | Até 15/01/2021 | ----- x ----- |
| Trancamento de matrícula | Até 10/08/2020 | Até 17/11/2020 | Até 08/03/2021 | ----- x ----- |
| Pedido de inscrição em disciplinas | De 06/07/2020 a 24/07/2020 | De 11/10/2020 a 17/10/2020 | De 30/01/2021 a 05/02/2021 | ----- x ----- |
| Concordância do pedido de inscrição em disciplina | De 27/07/2020 a 30/07/2020 | De 18/10/2020 a 24/10/2020 | De 06/022021 a 12/02/2021 | ----- x ----- |
| Efetivação do Pedido de Inscrição **(Divisão de Ensino – PR2)** | **31/07/2020** | **27/10/2020** | **15/02/2021** | ----- x ----- |
| Pedido de alteração de inscrição em disciplina – AID | De 02/08/2020 a 08/08/2020 | De 28/10/2020 a 31/10/2020 | De 16/02/2021 a 19/02/2021 | ----- x ----- |
| Concordância do pedido de alteração de inscrição em disciplina - AID | De 09/08/2020 a 12/08/2020 | De 01/11/2020 a 07/11/2020 | De 20/02/2021 a 26/02/2021 | ----- x ----- |
| Efetivação De Alteração do Pedido de Inscrição **(Divisão de Ensino – PR2)** | **13/08/2020** | **10/11/2020** | **01/03/2021** | ----- x ----- |
| Pedido de trancamento de inscrição em disciplina (desistência de inscrição) | De 14/08/2020 a 19/08/2020 | De 11/11/2020 a 14/11/2020 | De 02/03/2021 a 05/03/2021 | ----- x ----- |
| Concordância do pedido de trancamento de inscrição em disciplina | De 20/08/2020 a 31/08/2020 | De 15/11/2020 a 28/11/2020 | De 06/03/2021 a 19/03/2021 | ----- x ----- |
| Efetivação do Trancamento do Pedido de Inscrição **(Divisão de Ensino – PR2)** | **24/08/2020** | **01/12/2020** | **22/03/2021** | ----- x ----- |
| **Término de atividades** | **03/10/2020** | **16/01/2021** | **24/04/2021** | ----- x ----- |
| **Notas – Pautas de graus e frequência** | **De 04/10/2020 a 17/10/2020** | **De 17/01/2021 a 30/01/2021** | **De 25/04/2021 a 08/05/2021** | **----- x -----** |

## Avaliação e Atendimento

Critérios de aprovação são os do PPGI/UFRJ.

A avaliação da disciplina consiste em participação em sala de aula (P); exercícios e/ou protótipos desenvolvidos (E); apresentações/ /escritas de artigos (A).

$$MF = 0.2 * P + 0.2 * E + 0.6 * A$$

O aluno que desejar atendimento deverá requisitar o mesmo por e-mail e um horário será agendado pelos responsáveis para o atendimento.

serra@pet-si.ufrrj.br

zavaleta@pet-si.ufrrj.br

# Entregáveis – Março

4/3 - Aula + Definição dos grupos + PIT 5 min (Apresentação geral em PPT - DataSet + Problema + Objetivo)

11/3 - Aula + Descrição do projeto de DS - Tudo é via GIT!

- Entregar o projeto contendo (V1): Detalhamento e descrição textual da definição do problema a ser explorado pela equipe; descrever tecnicamente o dataset e sua fonte, o que pretendem fazer e o que vão e como extrair (plano dos experimentos). Apresentar os objetivos geral e específico do projeto de DS; apresentar métodos de data cleaning/tratamento de dados que serão usados, apresentar a proposta de modelo de extração de conhecimento e visualização de dados que será adotado.

18/3 - Aula + Refinamento do projeto de DS

- Agregar ao projeto (V2) : Plano do experimento a ser executado (scripts no Colab x Jupyter x IDE), projeto de coleta de metadados da proveniência dos experimentos, projeto para tornar seu experimento reprodutível, adicionar qualquer outro melhoria ou diferencial que julguem necessário

25/3 – Aula + entregar da 1a versão do artigo – Recomenda-se usar o template da LNCS e suas regras de formatação.

- Texto final (15-20) páginas com referências (pode ser em português ou inglês, escolha do grupo)

# Entregáveis – Abril

1/4 - Aula

8/4 - Aula + Sorteio ordem de apresentação dos grupos

15/4 - Entregar da 2a versão do artigo + Apresentação trabalho alunos (1ª. Parte dos grupos)

22/4 - Apresentação - trabalho alunos (2ª. Parte dos grupos)

10/5 (?) - Entrega da versão final do artigo + projeto completo (V3) + scripts com provenance + datasets anotados+ resultados de DS+ Depósito do notebook reprodutível e executáveis no GIT

PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA
UNIVERSIDADE FEDERAL DO RIO DE JANEIRO

# Introduction to Data Science

## MODULE IV – PART II

Scientific in Silico Experiments as Workflows and Scripts

Prof Sergio Serra e Jorge Zavaleta

# What? The Modeling Process

Ask an interesting question & learn reproducibility

Get the Data

Explore the Data

Model the Data

Communicate/Visualize the Results

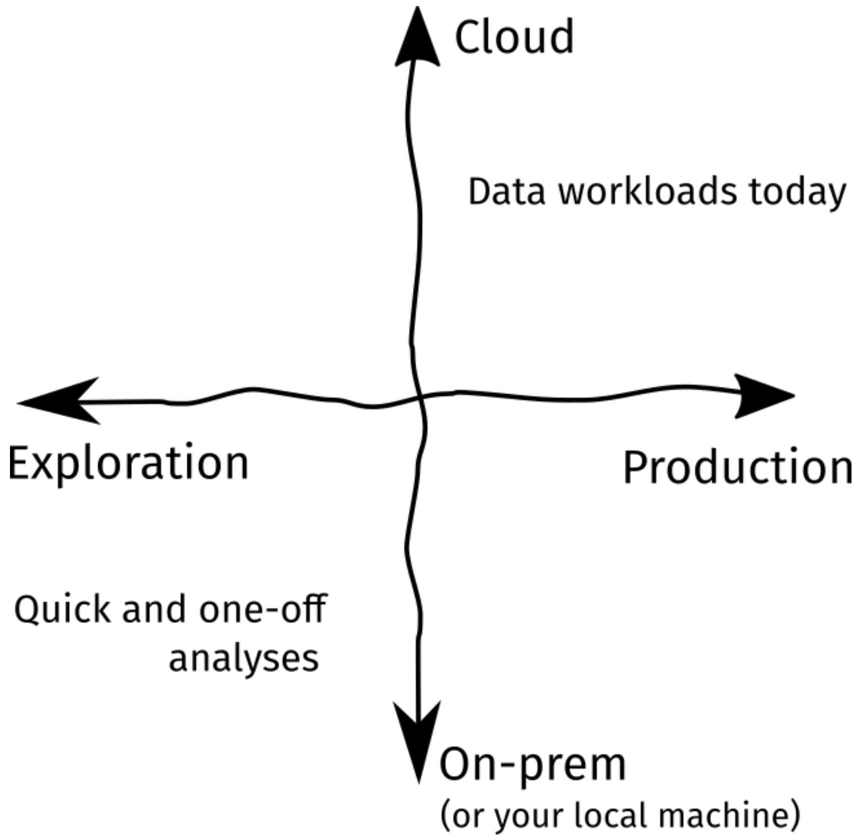
How were the model made?

Which model is relevant?

Who made it and when/how its executed?

Module III and IV

# The data science landscape today



**Data science is rapidly changing...**

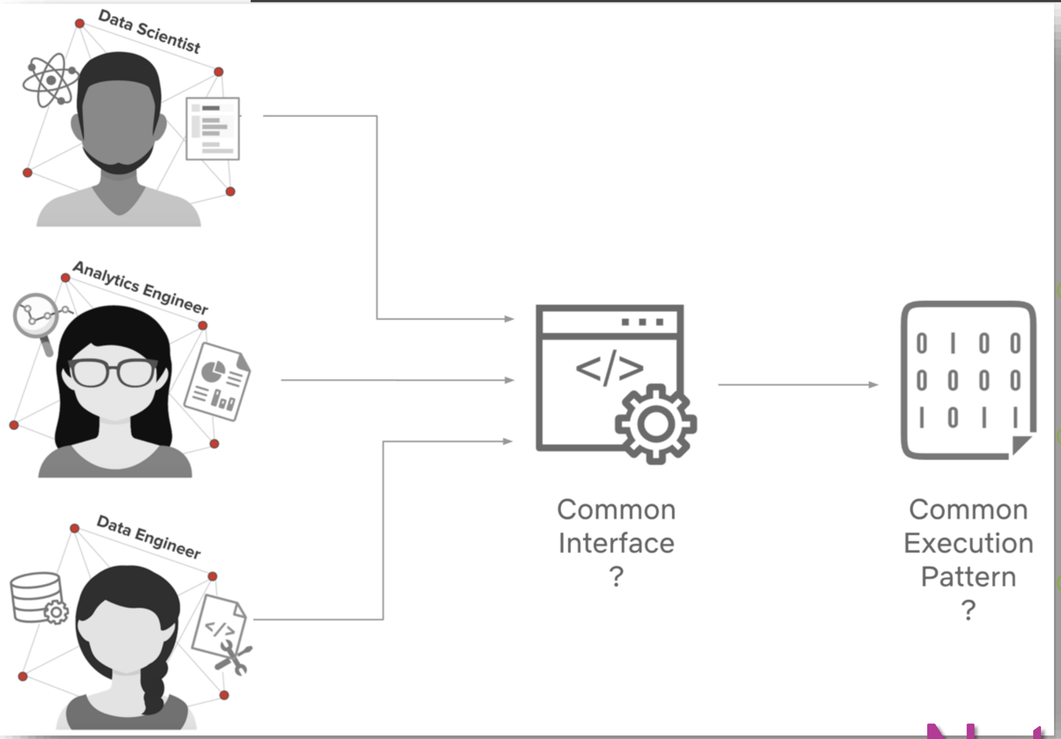
Phrases like “sexiest job of the 21st century” and “data is the new oil” have become old and replaced by more realistic **business problems and grounded technical challenges.**

The changes are three-fold: **We need to support both the**

(1) Demand for productionizing analyses and experiments,
(2) Rapid adoption of the Cloud,
(3) Different data user needs

A simplified framework on how to think about the advancements in the data science process for the past years

# Our (future) data users



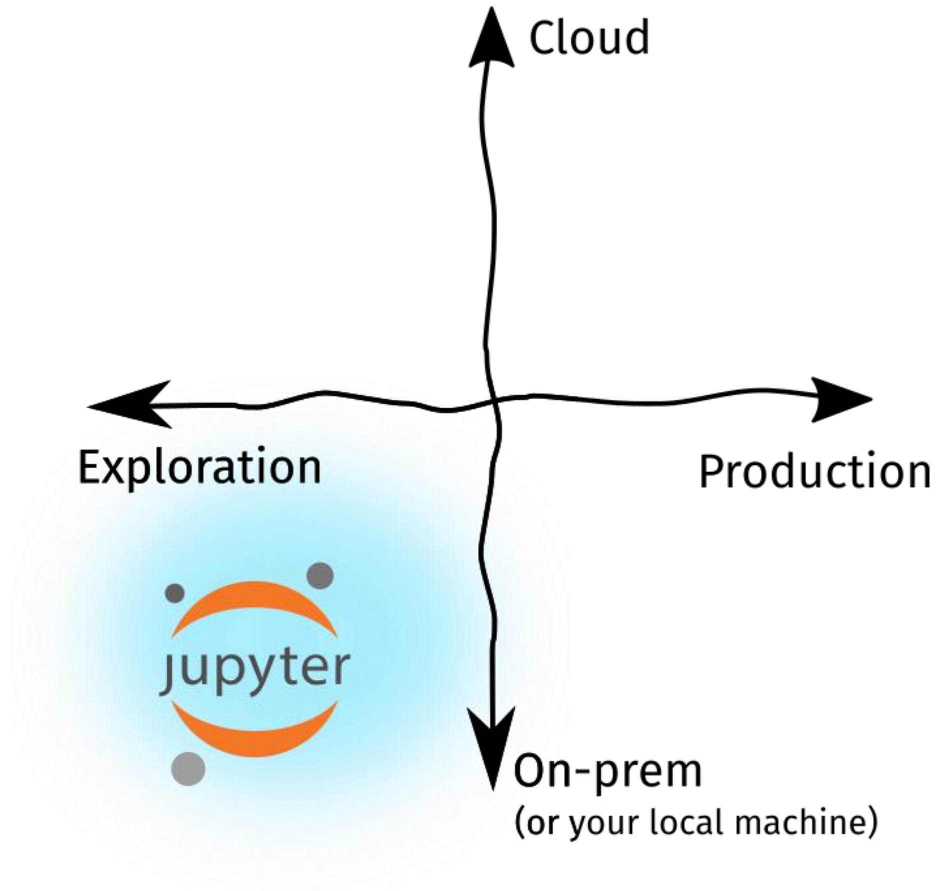
Data science teams grow (confluence of users). Unfortunately, their toolsets and workstyles vary a lot. To look forward at (future) needs, we ask a few basic questions:

- What interface will a **data analyst** use to chart a few combined queries into a business report?
- How will a **data engineer** write code that a reliability engineer can help ensure runs every hour?
- How will a **machine learning developer** encapsulate a model iteration their colleagues can reuse?

**Notebooks.**

- Shareable
- Easy to Read
- Documentation with
- Code
- Outputs as Reports
- Familiar Interface
- Multi-Language

# Notebooks: Woes To Wins...



The Good

- Quick iteration cycles
- Recorded outputs
- Easy to modify

The Bad

- Lack of history (version and provenance)
- Easy to modify

The Ugly

- Concurrent edits
- Browser state

## ...production gaps to fill

Things to preserve:

- Results linked to code
- Good visuals
- Easy to share

Things to improve:

- Not versioned
- Mutable state
- Templating
- Provenance

The Notebooks we know only cover a small domain of the data science ecosystem

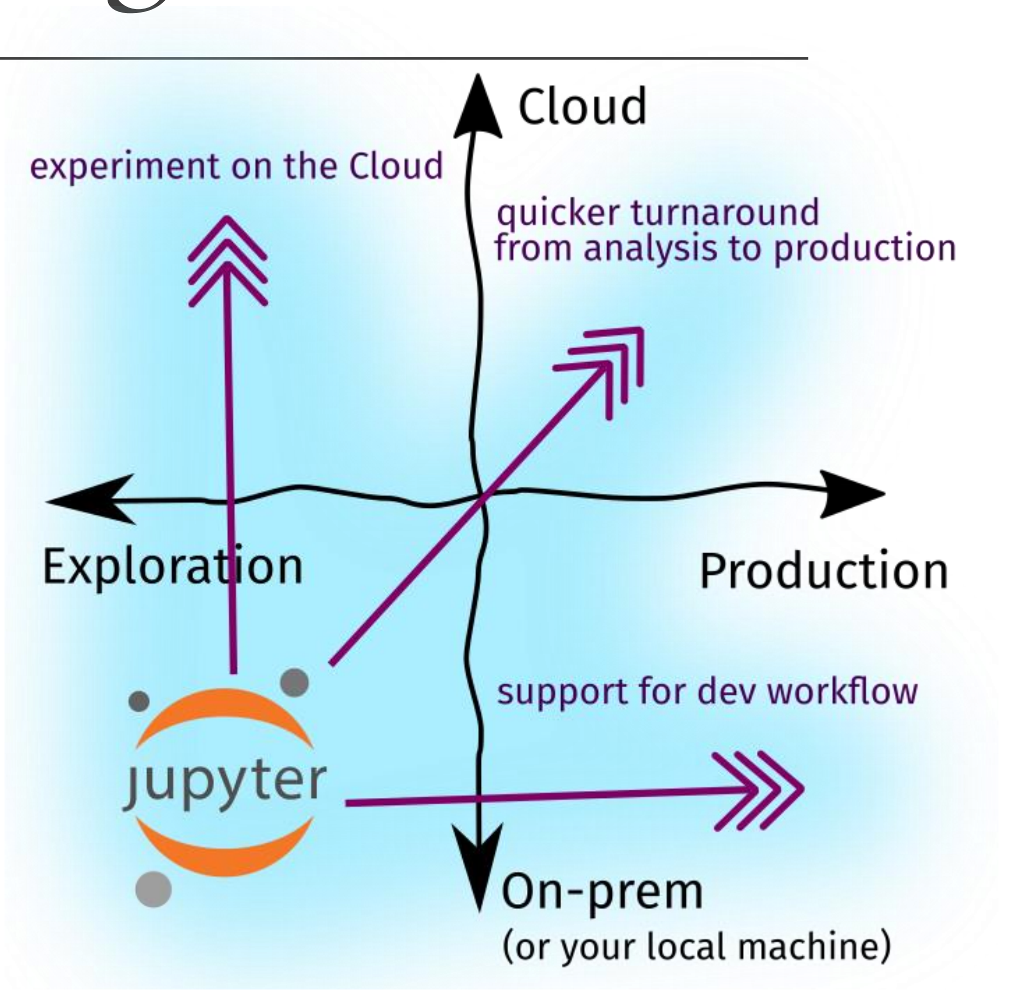
# The three forces of change

Jupyter Notebook ecosystem is growing, three forces:

1- Experiment on the Cloud: big data demands large compute and storage, something that your average consumer-grade machine will not always be capable of.

2- Support for developer workflow: more and more data science teams are starting to adopt software engineering best practices—version-control, gitfow, pull requests, and more.

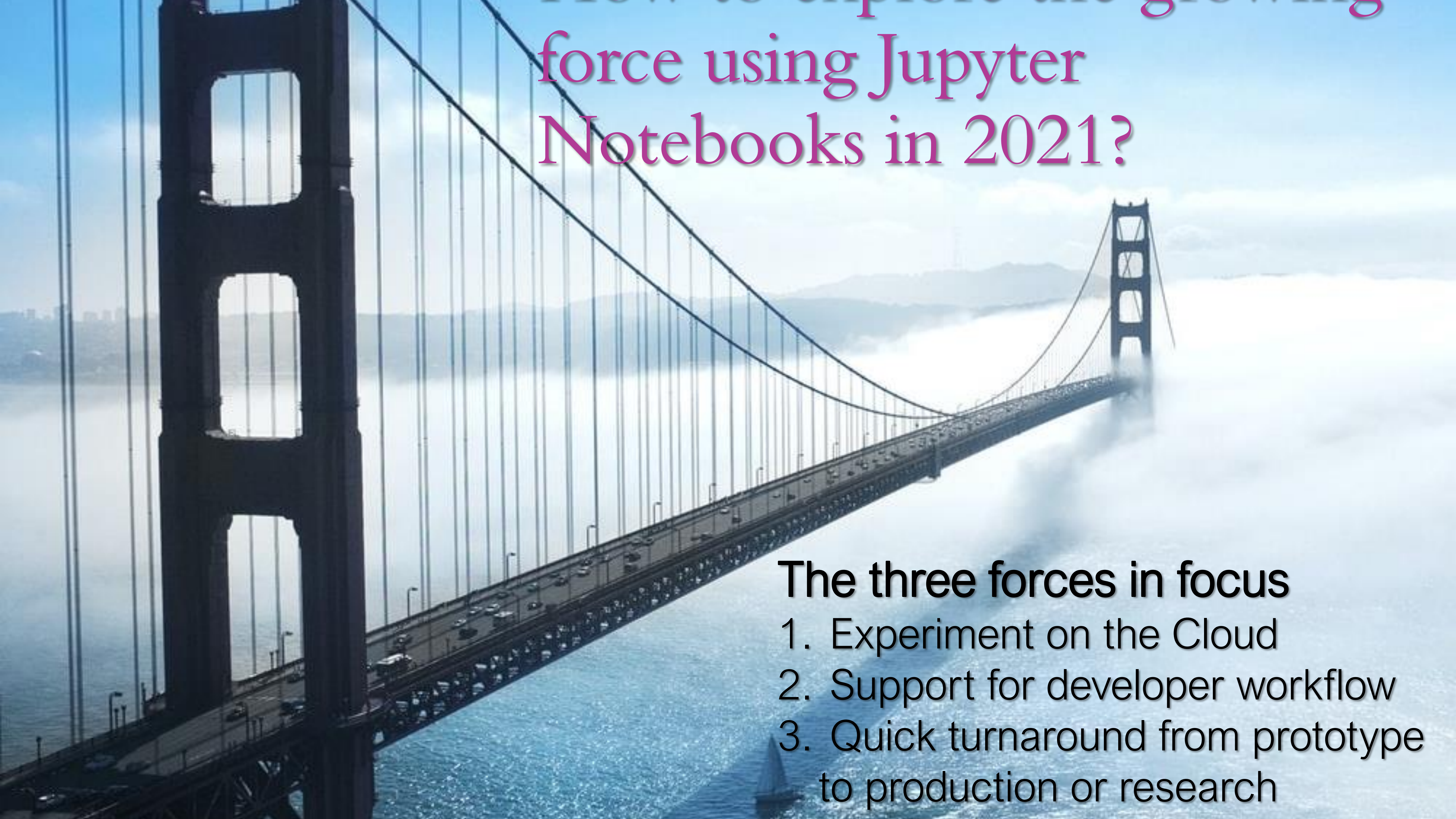
3- Quick turnaround from analysis to production/research: it's not enough to test hypotheses under controlled environments. Software written for analysis should be easily reused for prod.



The growth of the Notebook ecosystem is driven by these forces

## The three forces of change

- Growing towards a more Cloud-first environment means → perform Notebook-based tasks in machines more powerful than our own.
  - For example, managed notebook instances enabled us to run Jupyter notebooks from a remote server with no-ops and setup.
- Growing towards a more production workflow provides us with a set of tools to endow our notebook-based tasks with software engineering practices.
  - the growth of a tool doesn't depend on a single entity or organization. Filling these gaps may stem from individuals who contribute third-party plugins or organizations offering managed services from notebooks.

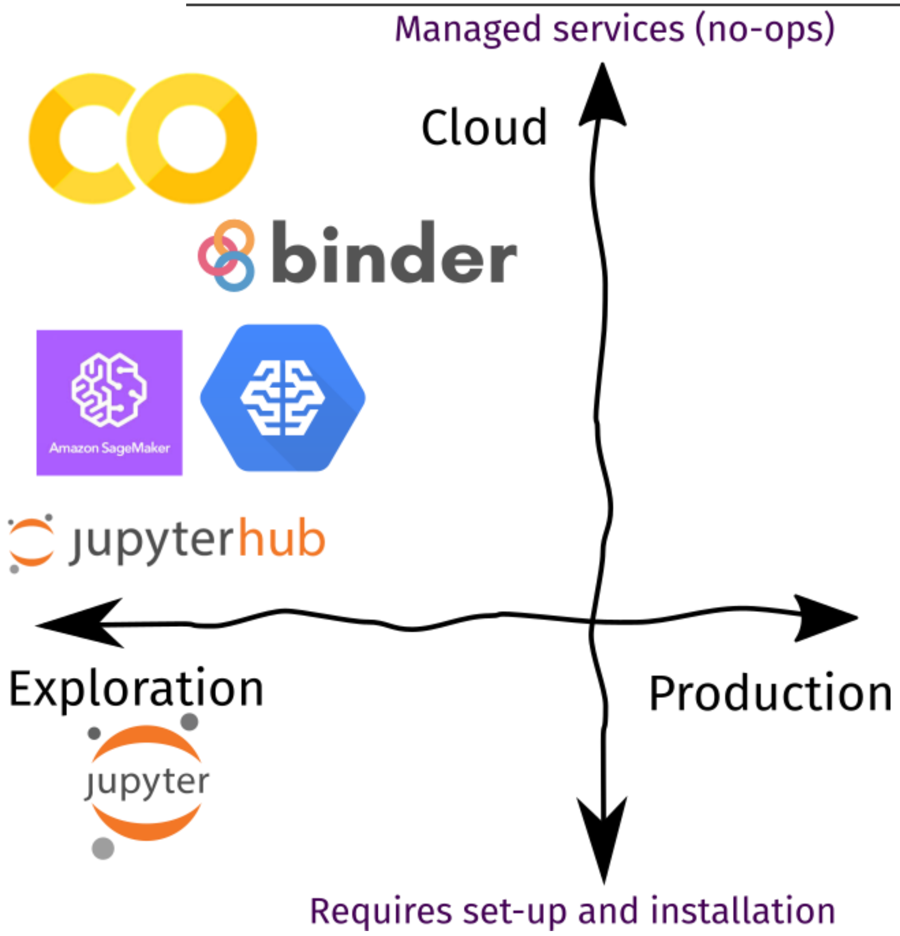


# How to explore the growing force using Jupyter Notebooks in 2021?

## The three forces in focus

1. Experiment on the Cloud
2. Support for developer workflow
3. Quick turnaround from prototype to production or research

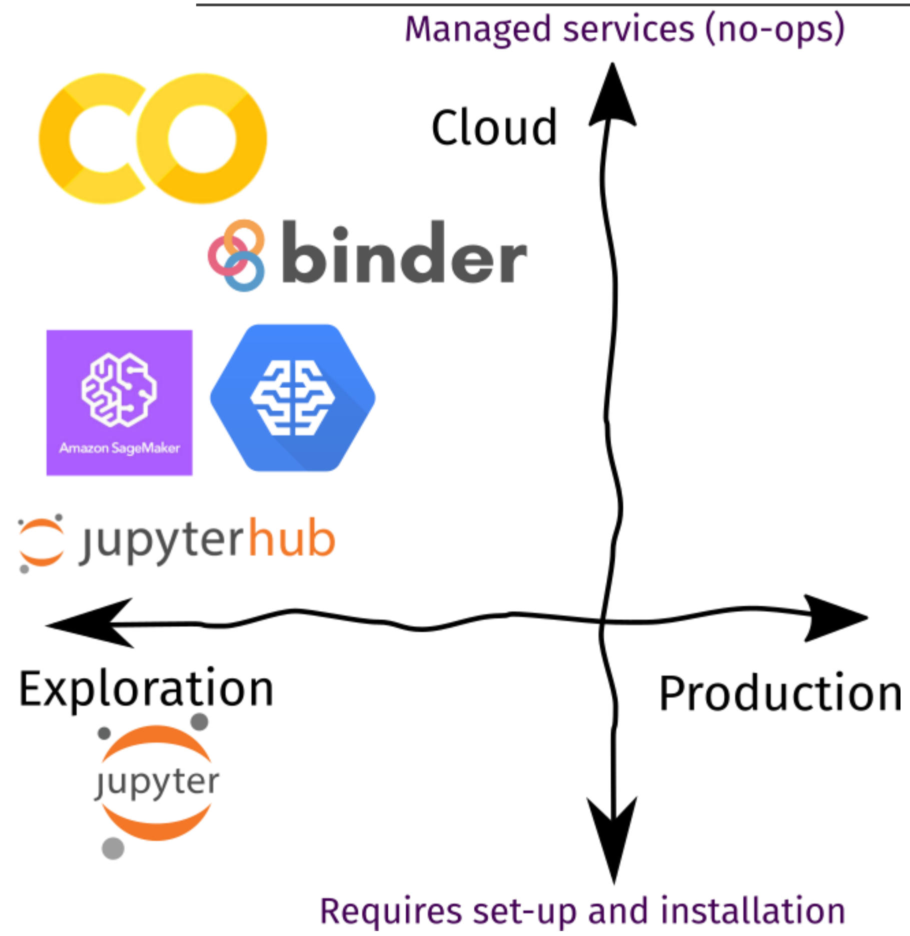
# 1- Experiment on the Cloud



A huge component of the **data science workflow is experimentation**. It includes engineering, hyperparameter optimization, and testing-out models.

As data gets bigger and models get more compute-intensive, it is not enough to do everything on your laptop. Experimentation workloads are often delegated to the Cloud.

# 1- Experiment on the Cloud



| | Recommendation |
|---|---|
| Best Tool of Choice | Google Colaboratory. With Colab, you have immediate access to powerful hardware for your experiments. |
| Runner-up | If you are on a Cloud Platform. Check-out managed notebook instances in AWS Sagemaker or Google Cloud's AI Platform. |
| Check out | Binder for "publishing" finished work, and JupyterHub if you wish to setup your own managed notebook instances. |

## 2- Support for developer workflow

As data science teams grow, we see a confluence of researchers and engineers working together. Unfortunately, their toolsets and workstyles vary a lot.

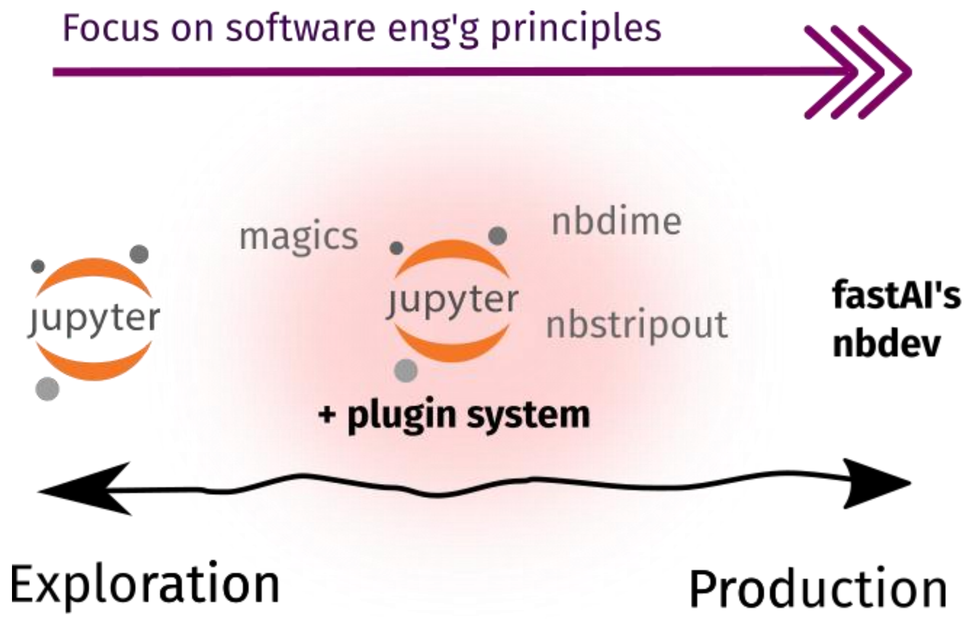
- Engineers are comfortable with text editors and IDEs, while some researchers find Jupyter Notebooks convenient. Sometimes engineers disdain notebooks, researchers who aren't familiar with concepts like DRY or KISS ☺☺☺

In an ideal world, everyone uses the same thing and we're all happy. However, in practice…..

(1) Engineers supporting the tools comfortable for researchers, and

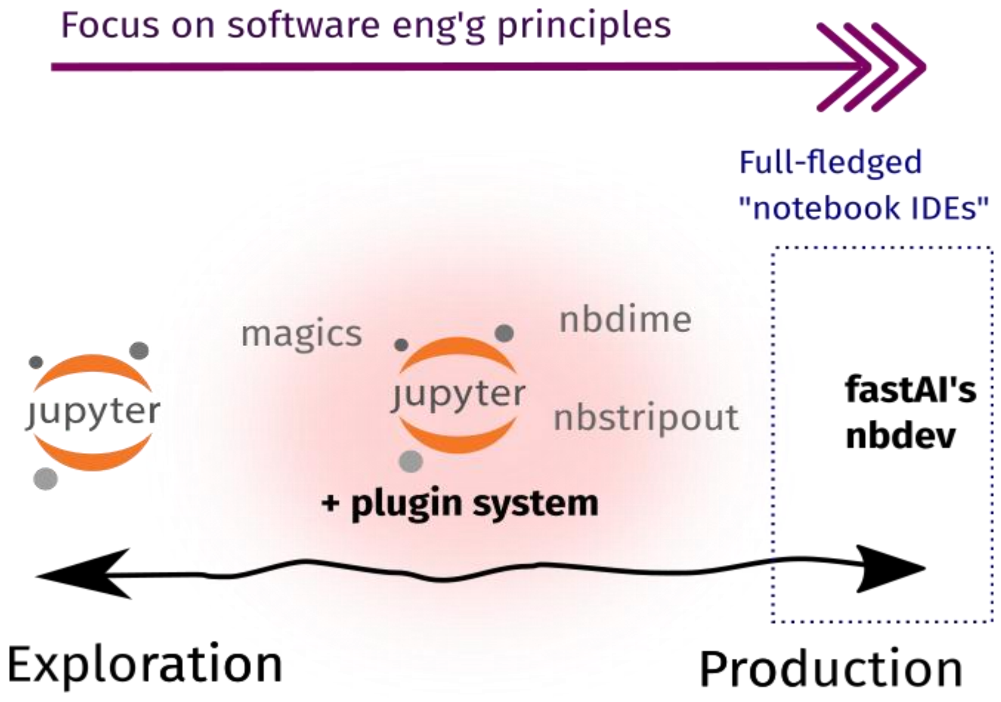
(2) Researchers learning basic software engineering principles.

# 2- Support for developer workflow



| | In Jupyter-ecosystem | External |
|---|---|---|
| Best Tool of Choice | nbdime and nbstripout emula te the Git developer workflow | cookiecutter-datascience to set up a more organized project structure |
| Runner-up | nbconvert commit your files into a more human-friendly text format | data-version control (DVC) so that you won't "pollute" your Git repo with model files (can reach GBs) or non-essential doc types (Excel sheets, PDFs, etc.) |
| Check-out | fast.AI's nbdev as a highly-opinonated way of developing everything in the context of notebooks. | |

## 2- Support for developer workflow



it encourages two things: modularization and documentation.

**Modularization** means that if you have copy-pasted functions scattered across multiple notebooks, then don't repeat yourself, and put them in a module where you can define them once and import anywhere.

**Documentation,** is the practice of writing text to accompany your analyses and code.

- **My suggestion is to document aggressively: learn about Python docstrings, Sphinx, or simple Markdown.**

Jupyter Notebooks aren't suited for a Git-like workflow. Even if they're just a JSON text file, they become quite unruly when checking for diffs or *gulps* resolving conflicts.

- Try a combination of **nbstripout** and **nbdime**.

# 3- Quick turnaround from prototype to production or research

This means that an engineer/researcher can treat Notebooks as **blackboxes** with expected inputs and outputs, then run it as she would any other process.

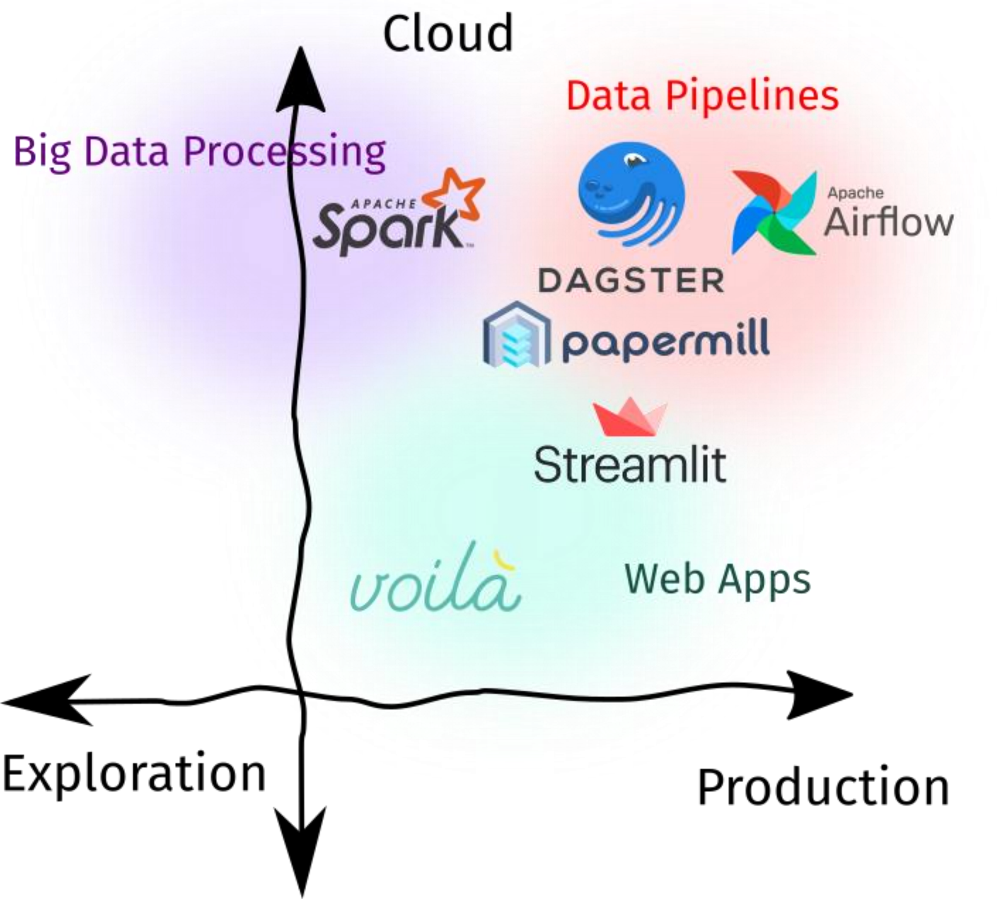
- How notebooks behave as software components? and how it integrates with some standard engineering tools?

Using notebooks in production or research is a highly-debated topic. There may be merits in using Notebooks, but it is important for data science teams to evaluate if a Notebook or the standard stack is the right tool for the job.

There are lots of tools that support bringing Notebooks into production/reproducible research.

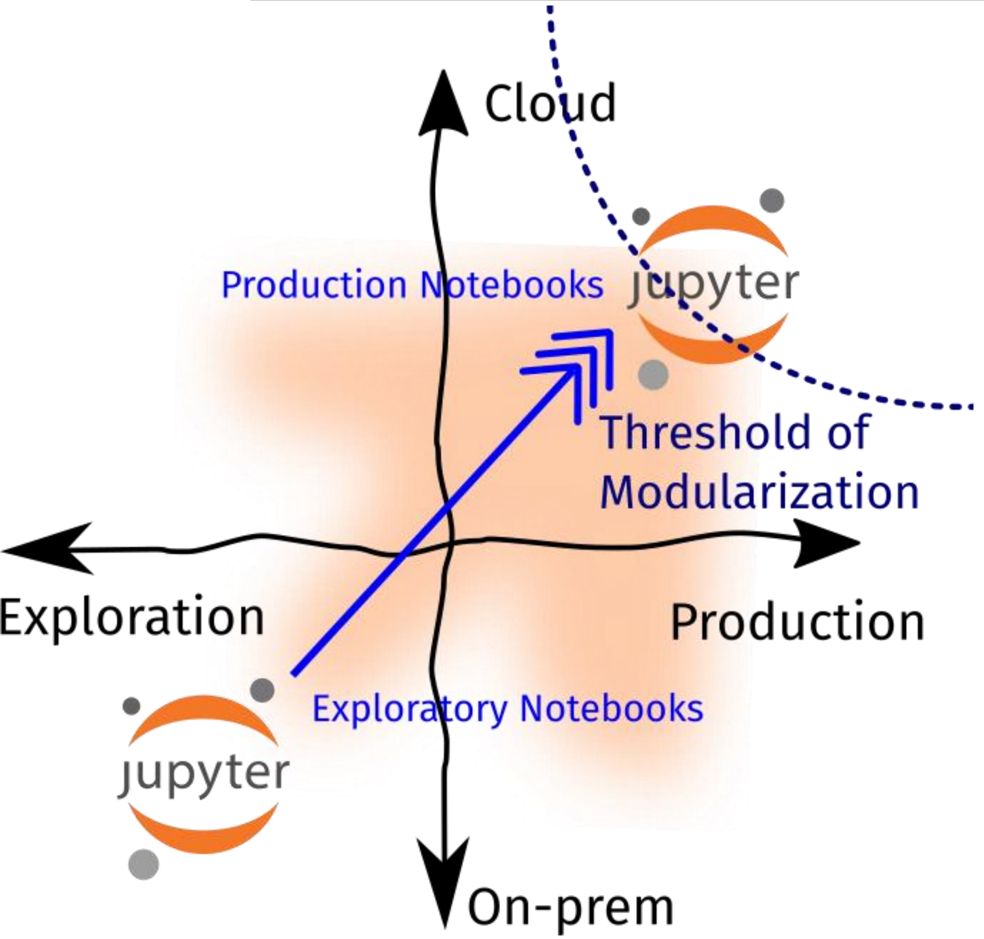
- **Components of a data pipeline:** Notebooks can be thought of as a function that takes in an input, transforms it, and returns an output.
  - For example, it can be used to obtain embeddings from an image before storing it into a distributed filesystem, a final layer in a data pipeline to generate templated reports, or a batch ML service that outputs some predictions for downstream tasks.
- **Web-application:** Interactivity is one of the main features of Jupyter Notebooks.
  - On one end you can intersperse code with text, then on the other, you can put widgets for control. Nowadays, there are tools that convert Notebooks into a fully-fledged web application or reproductible research
- **Big data processing interface**: We used to analyze Big Data using SQL, Hive, or some other DSL.
  - We can replace this interface with a Notebook-like point-of-contact. This gives enough expressiveness (since we're using Python) to quickly analye petabyte-scale datasets.

## 3- Quick turnaround from prototype to production or research



| | Recommendation |
|---|---|
| Best Tool of Choice | Papermill for parametrizing and running notebooks as you'd do for Python scripts. Streamlit, although not Jupyter-native, can be used to create web-apps in-tandem with nbconvert. |
| Runner-up | Voila for a Jupyter-native experience of building web-apps, and Apache Spark for Big Data Analysis. |
| Check out | Dagster (Dagstermill) and Airflow (PapermillOperator) for Notebook-based data pipelines. Although I recommend that for critical ETLs, traditional Python scripts should be considered. |

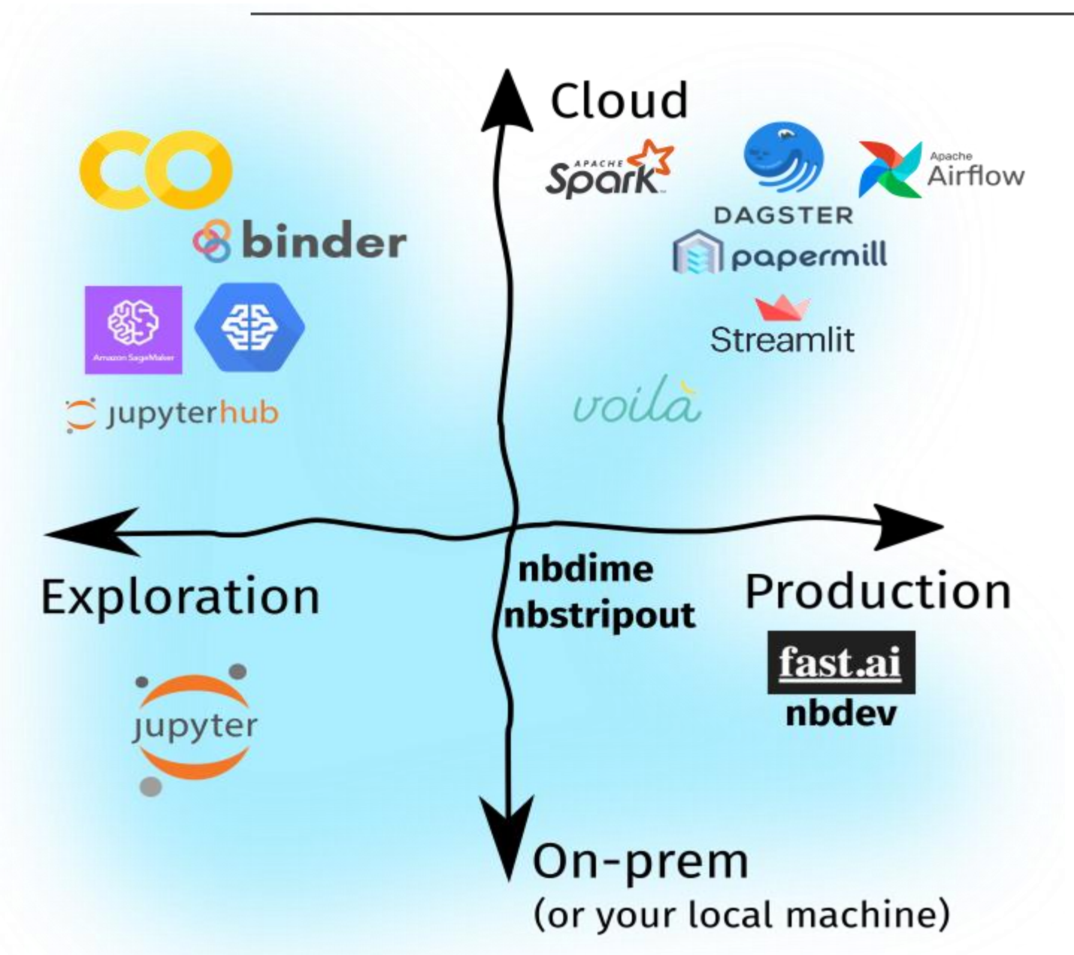
# Putting-it together



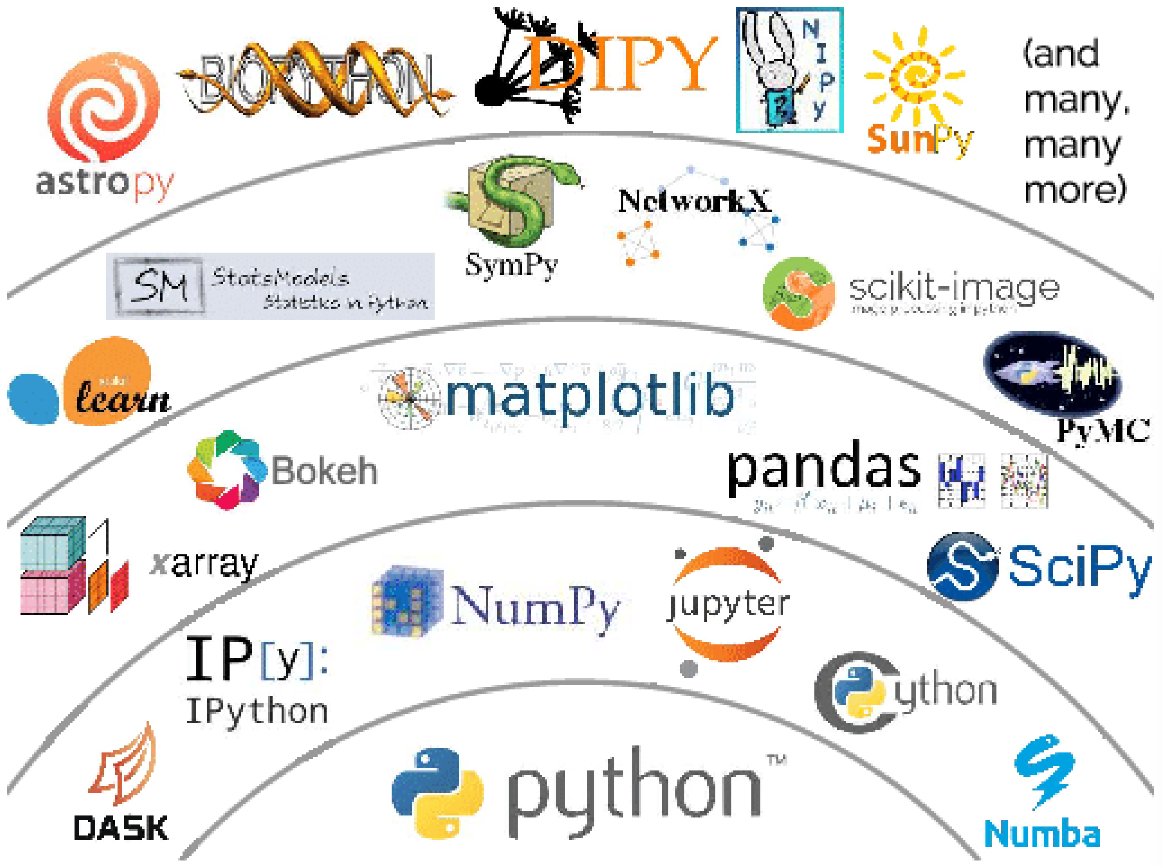
This means… even if you start with some **ad-hoc and dirty analyses**, You **always need to update** them so that you (or other people) can **rerun** them seamlessly and understand my methods in the future…

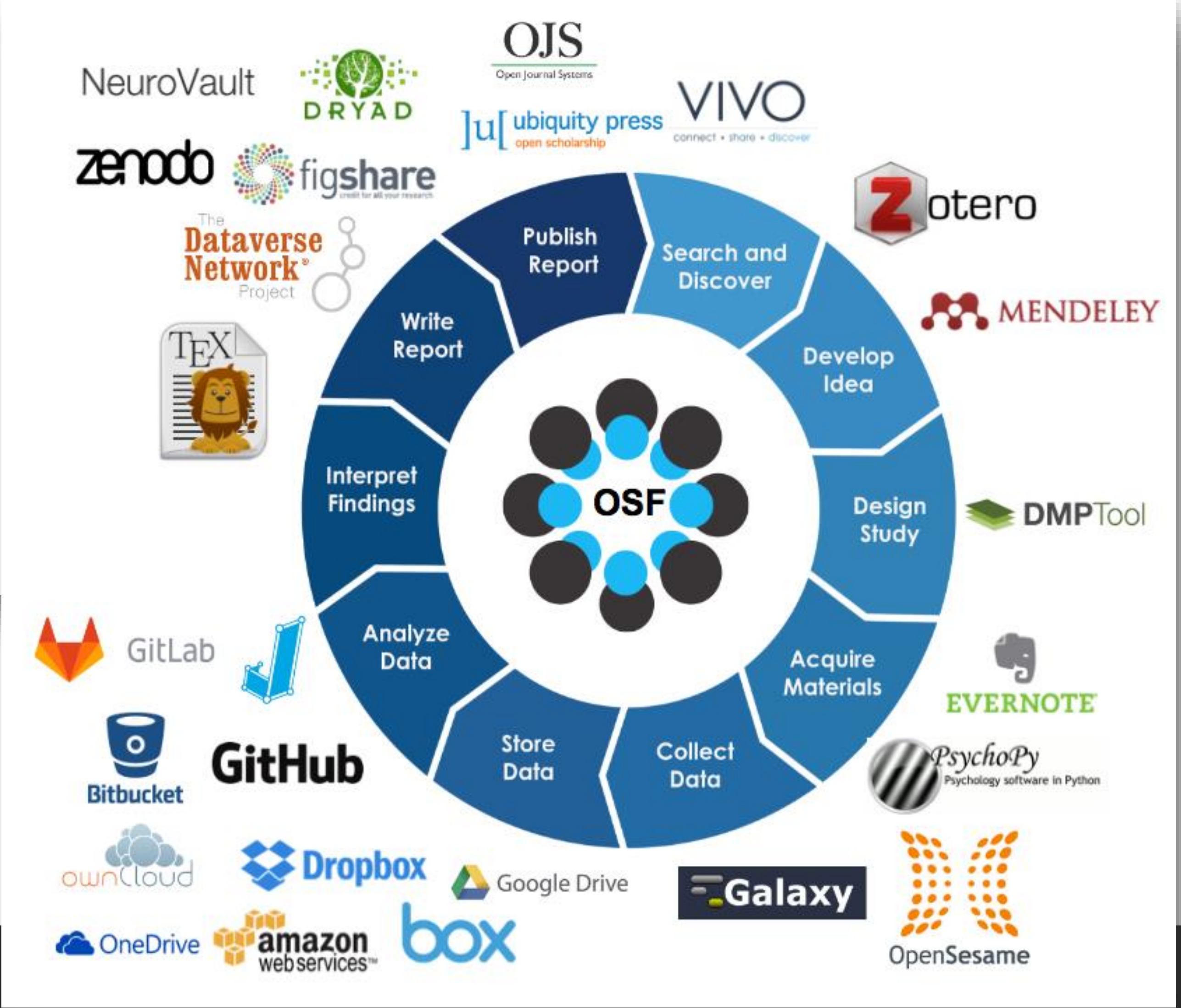
How to e convert exploratory Notebooks to Production/Reproductible research ones?

# Putting-it together – The Principles

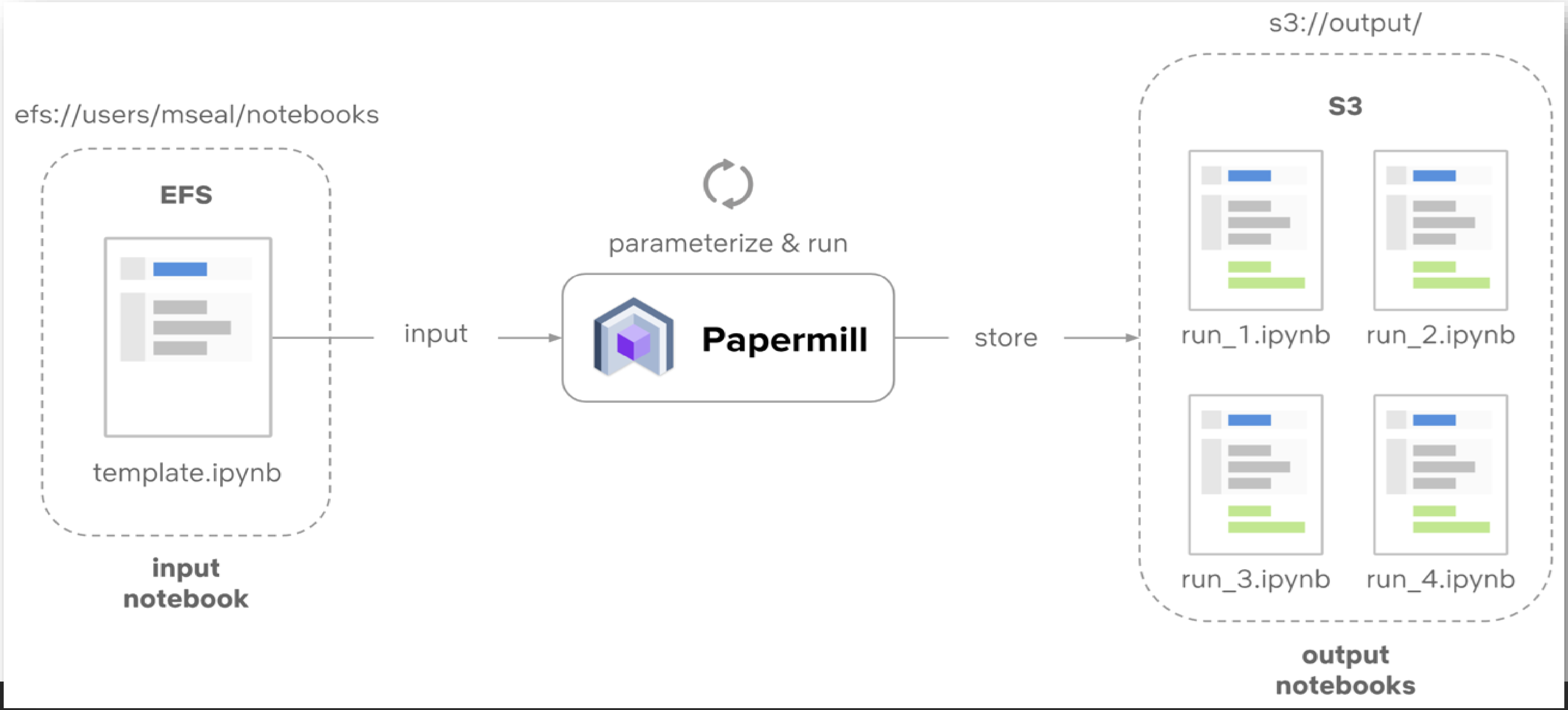


1. Keep a standard project structure and gitflow.
2. Refactor oft-repeated functions into Python modules.
3. Ensure that your notebook runs from top-to-bottom.
4. For data pipelines, use papermill ( et al.. ) and configure your notebook to take advantage of it.

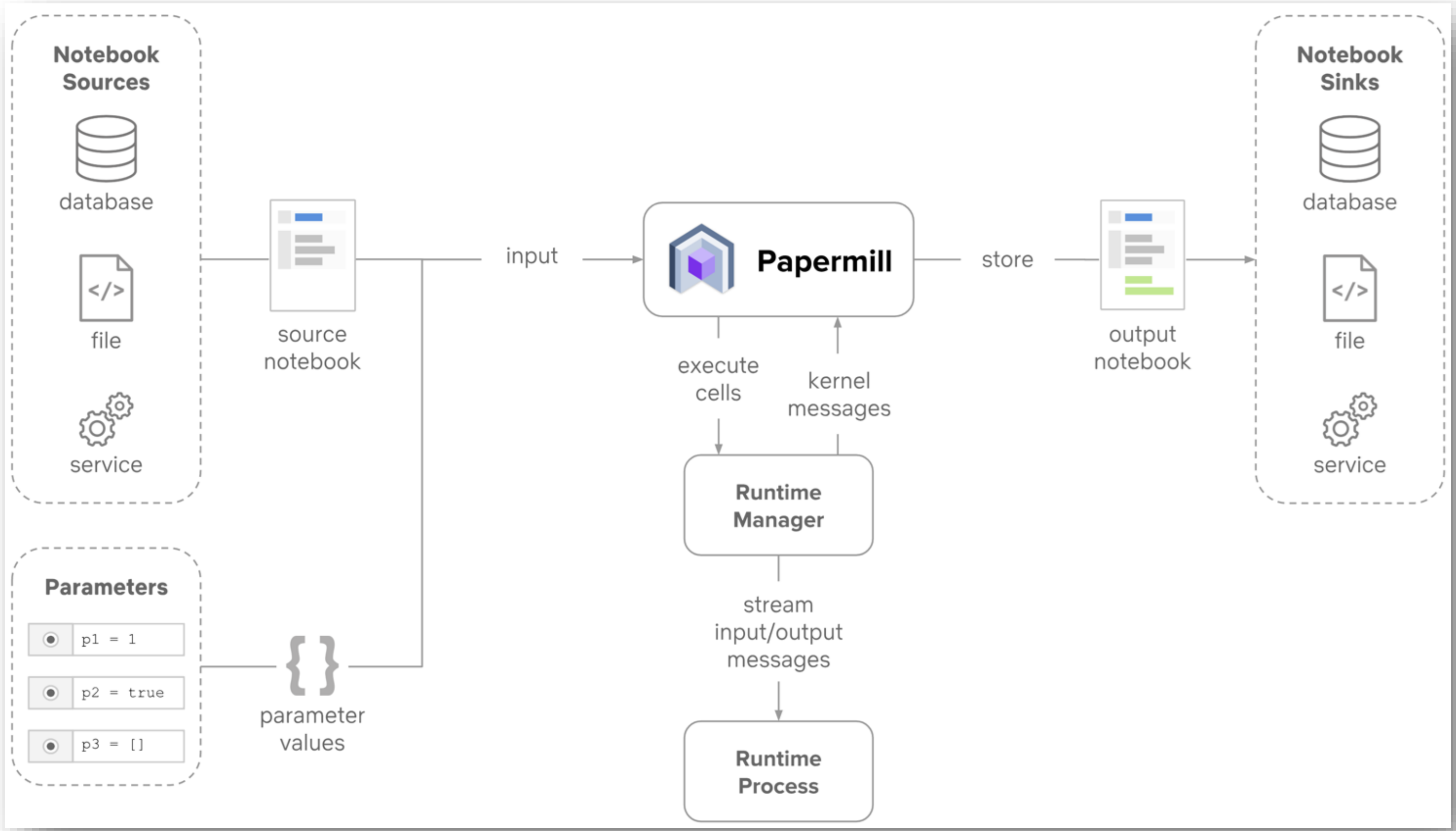

astropy
BIOPYTHON
DIPY
NIPY
SunPy
(and many, many more)
SymPy
NetworkX
StatsModels Statistics in Python
scikit-image
learn
matplotlib
PyMC
Bokeh
pandas
xarray
NumPy
jupyter
SciPy
IP[y]: IPython
Cython
DASK
python™
Numba


NeuroVault
DRYAD
OJS Open Journal Systems
ubiquity press open scholarship
VIVO connect • share • discover
zenodo
figshare
Zotero
The Dataverse Network Project
MENDELEY
TEX
Publish Report
Search and Discover
Write Report
Develop Idea
Interpret Findings
OSF
Design Study
DMPTool
Analyze Data
Acquire Materials
Store Data
Collect Data
GitLab
EVERNOTE
Bitbucket
GitHub
PsychoPy Psychology software in Python
ownCloud
Dropbox
Google Drive
Galaxy
OpenSesame
OneDrive
amazon web services
box

# Papermill


s3://output/
efs://users/mseal/notebooks
EFS
S3
template.ipynb
input notebook
input
parameterize & run
Papermill
store
run_1.ipynb
run_2.ipynb
run_3.ipynb
run_4.ipynb
output notebooks

# How it works

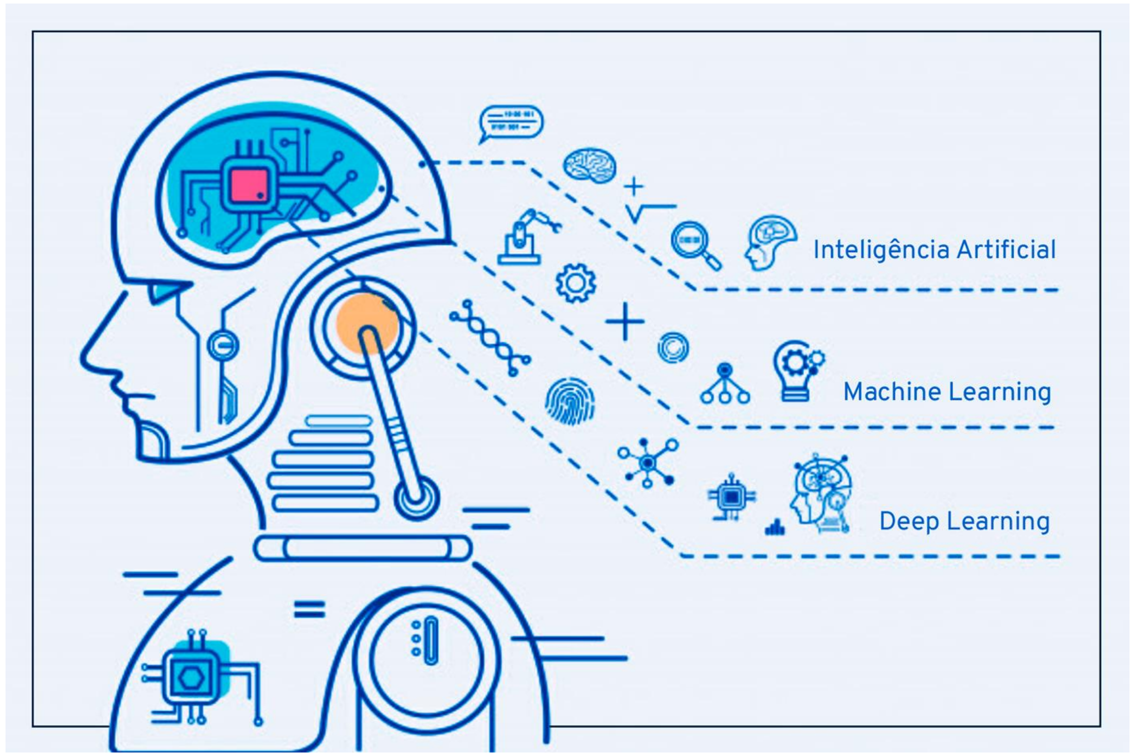


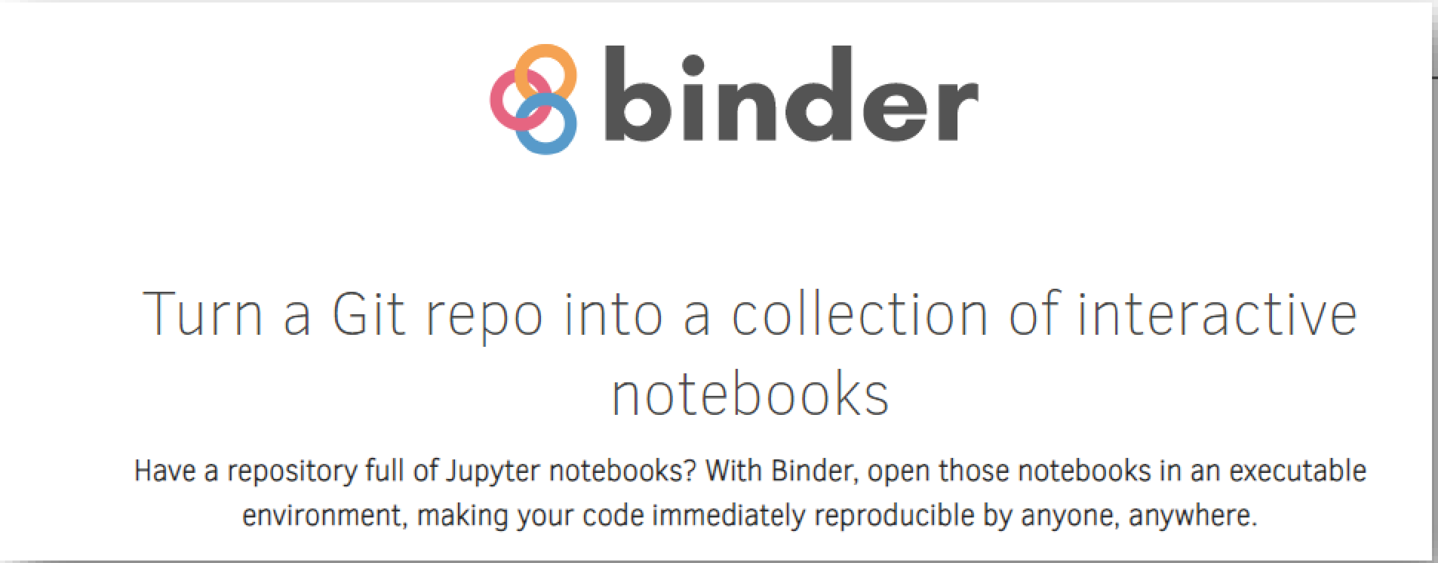
Reads from a source

Injects a parameters cell to the JSON document

Launches a runtime manager which runs the notebook kernel and collects messages.

Outputs to a sink




binder

Turn a Git repo into a collection of interactive notebooks

Have a repository full of Jupyter notebooks? With Binder, open those notebooks in an executable environment, making your code immediately reproducible by anyone, anywhere.

# Hands on...

NOTEBOOK:

MACHINE LEARNING

Scikit learn

# References

https://towardsdatascience.com/workflow-tools-for-model-pipelines-45030a93e9e0

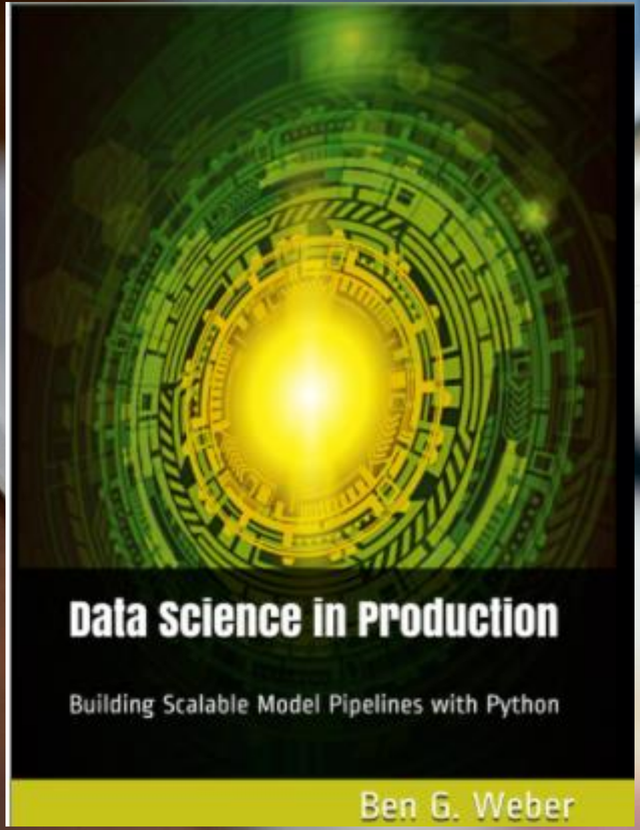
https://elifesciences.org/labs/d42fe2b9/integrating-binder-and-stencila-the-building-blocks-to-increased-open-communication-and-transparency

https://zenodo.org/

https://mybinder.org/

https://stenci.la/

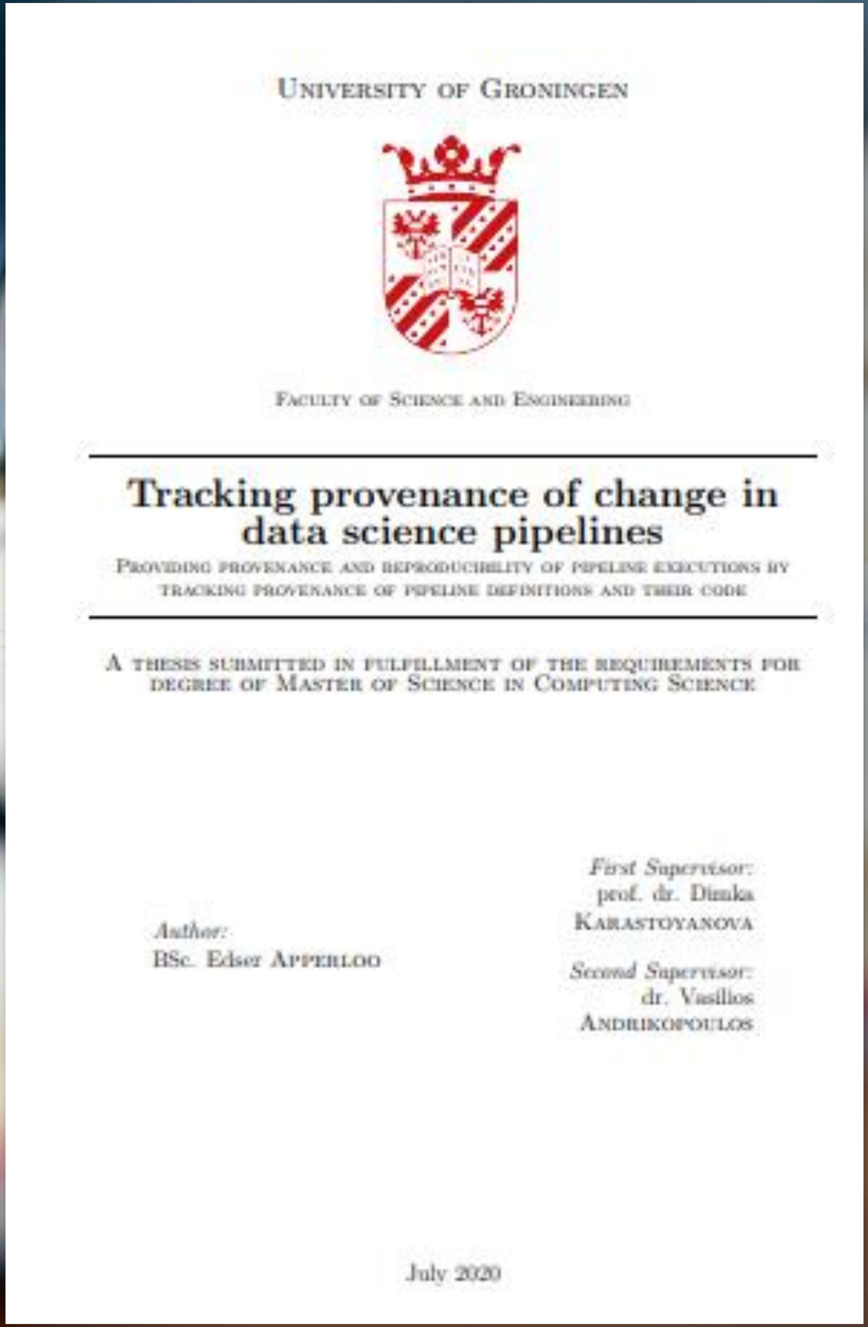
https://www.mysciencework.com/



Data Science in Production

Building Scalable Model Pipelines with Python

Ben G. Weber



University of Groningen

Faculty of Science and Engineering

Tracking provenance of change in data science pipelines

Providing provenance and reproducibility of pipeline executions by tracking provenance of pipeline definitions and their code

A thesis submitted in fulfillment of the requirements for degree of Master of Science in Computing Science

*Author:*
BSc. Edser Apperloo

*First Supervisor:*
prof. dr. Dimka Karastoyanova

*Second Supervisor:*
dr. Vasilios Andrikopoulos

July 2020