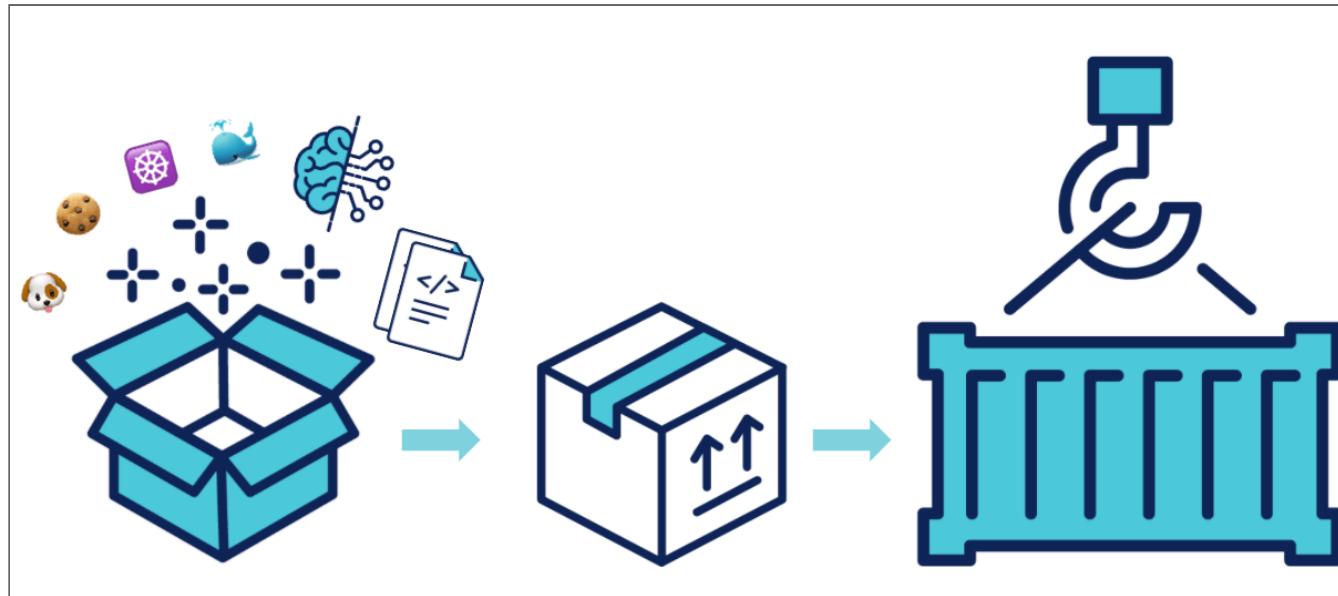


PACKAGING D'UNE APPLICATION DE MACHINE LEARNING



\$> WHOAMI ?



Consultant @ OCTO depuis 2017



Instructeur de formations : 🧠, 🔧, ☁



Co-organisateur de ce Meetup

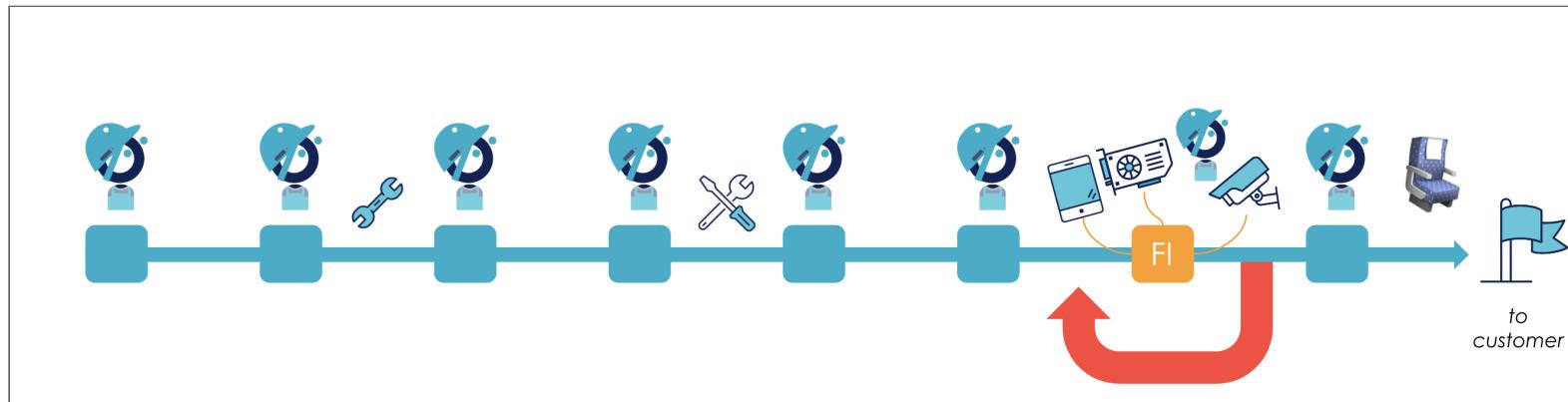
AVEZ-VOUS DÉJÀ ...

... mis en prod' une application de ML ?



DÉTECTION DE DÉFAUTS DE QUALITÉ

$$\text{🧠} + \text{📸} = \text{💰}$$

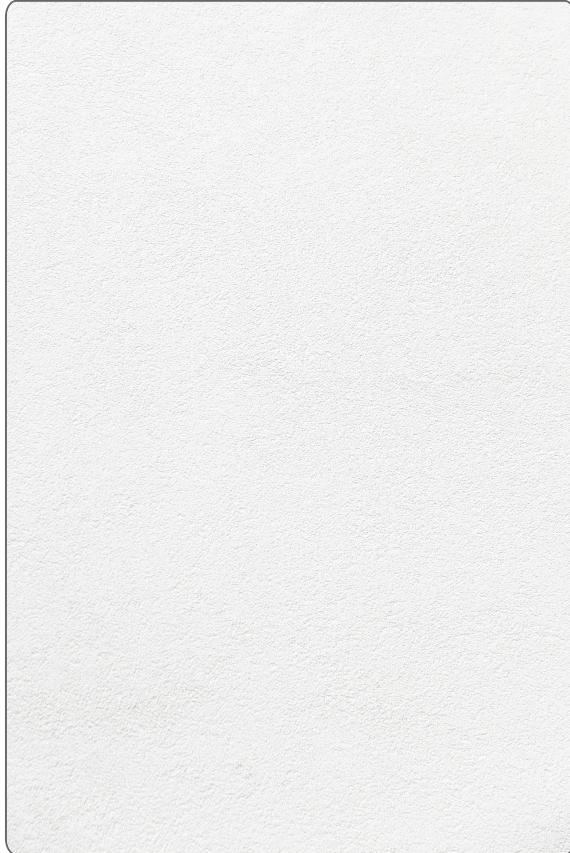


QUELQUES CAS D'USAGES



DÉFAUTS SUR TEXTILE

un bon pli



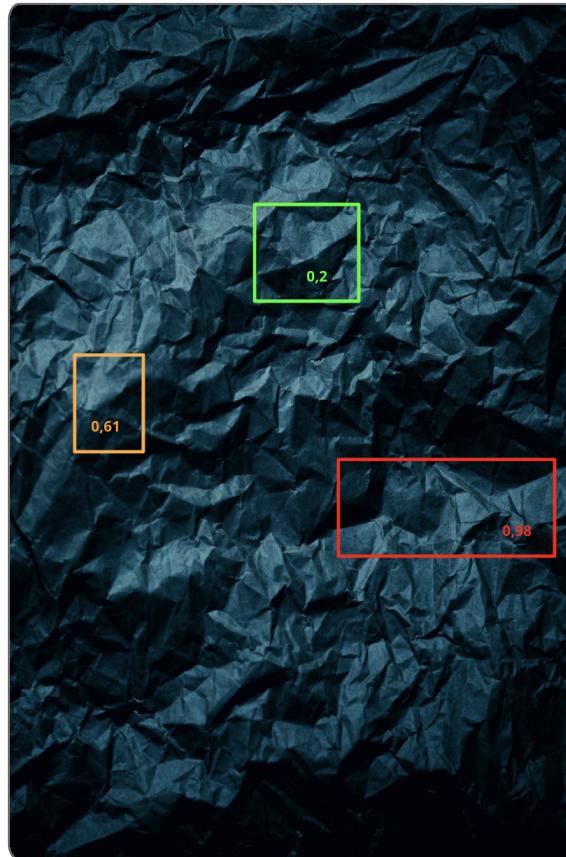
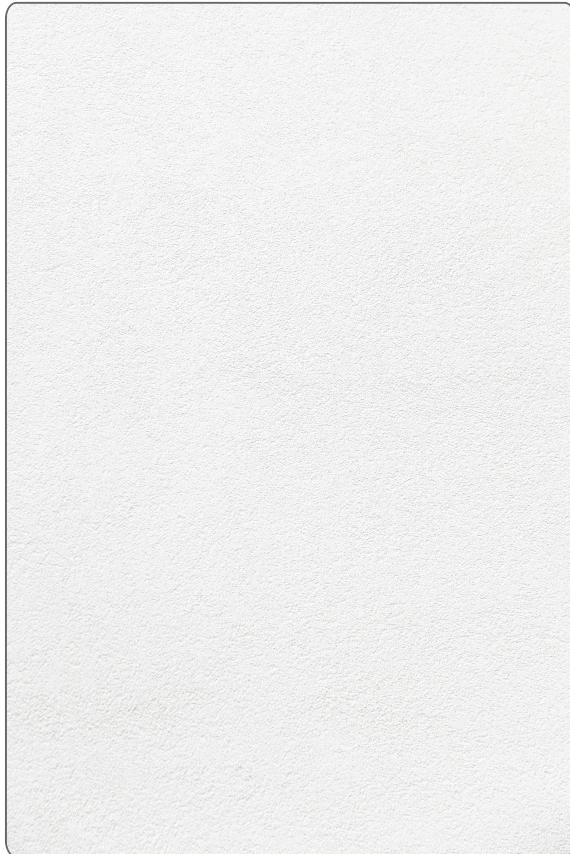
un mauvais pli



DÉFAUTS SUR TEXTILE

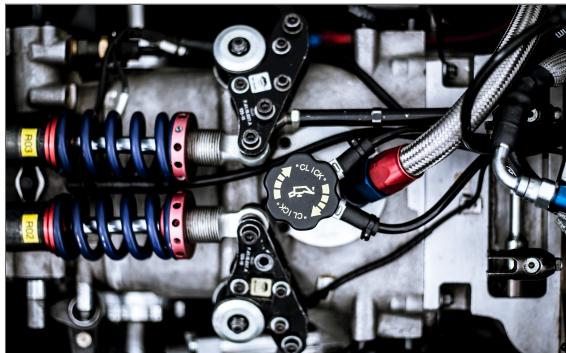
un bon pli

un mauvais pli

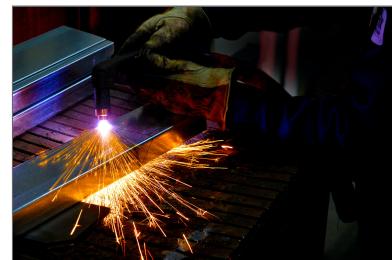


ET EN VRAC ...

Présence de composants



Etat de soudures



Détection de rayures



LA MODÉLISATION DURE PLUS D'1 AN

CE QU'ON AURAIT PEUT-ÊTRE DÛ FAIRE 😊

GOOGLE RULE OF ML #4

*Keep the first model simple
and get the infrastructure right*

AU BOUT D'1 AN

ON FAIT ENFIN NOTRE PREMIÈRE DÉMO EN USINE 💪

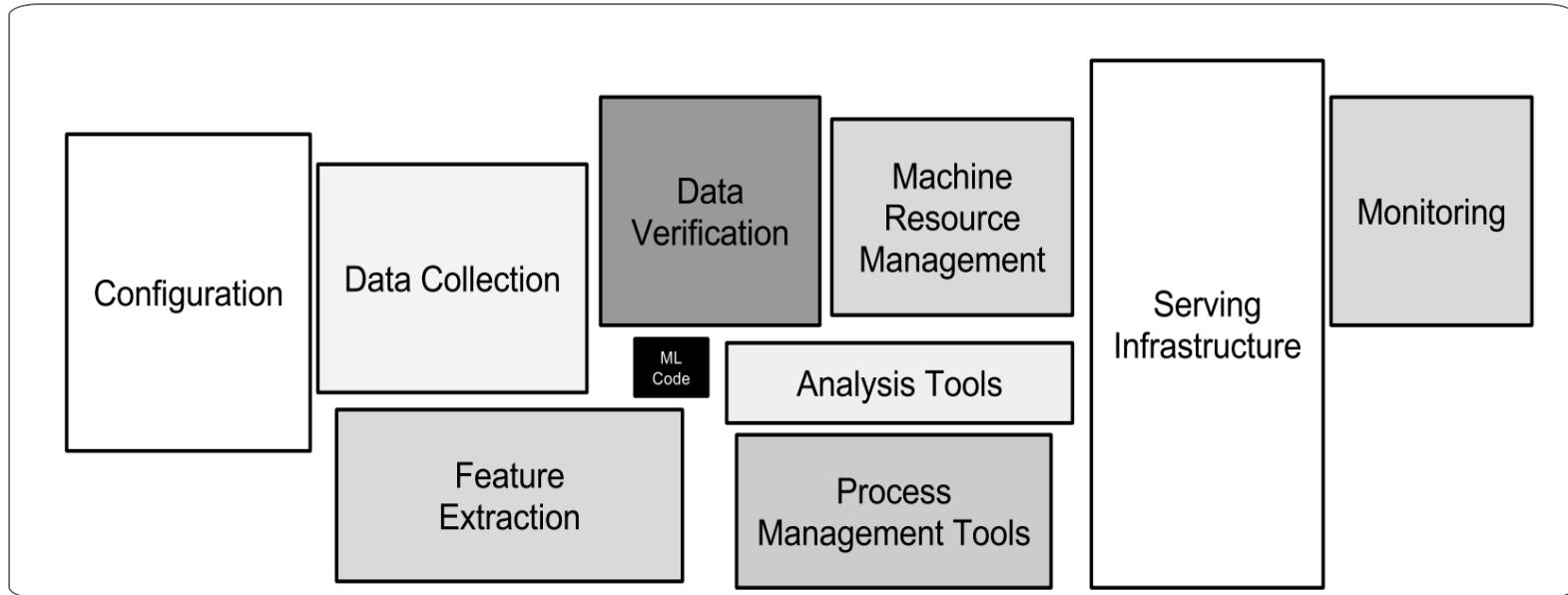


LA DÉMO CONVAINCT 

ON A LE GO POUR ALLER EN
PROD !  

ET MÊME POUR PASSER À
L'ÉCHELLE !  

PRENONS DU RECUL



Il nous a fallu plus d'1 an pour adresser la petite boîte noire 😱

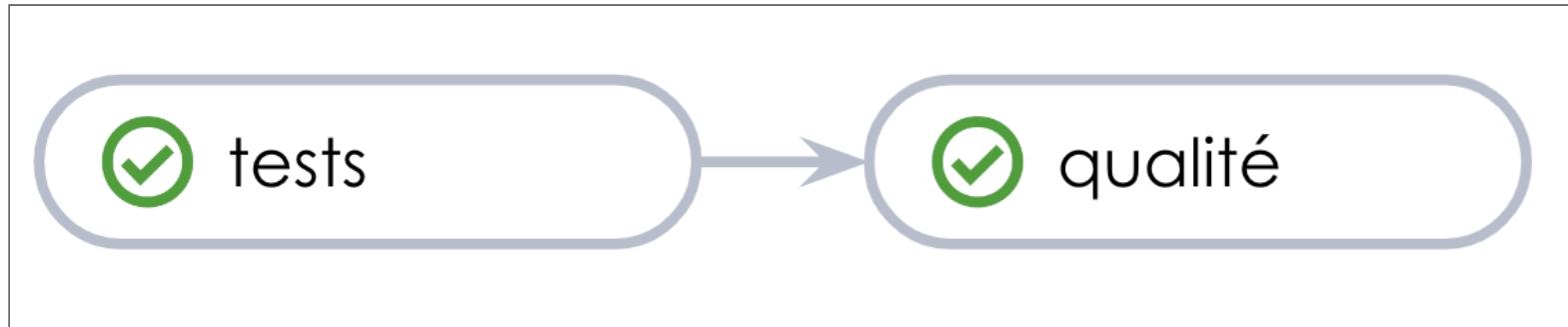
C'est du temps qu'on aurait pu consacrer à autre chose que la modélisation

(Source : [Hidden Technical Debt in Machine Learning Systems](#))

CE QU'ON FAIT BIEN

À CE MOMENT-LÀ

-  Tests automatisés avec Pytest,
-  Bonnes métriques de qualité de code,
 - *Code coverage, Pyramide de tests, ...*
-  Pipeline Gitlab-CI de tests auto.



LÀ OÙ ÇA PÊCHE

MODE D'EMPLOI POUR DÉPLOYER

-  Prendre l'avion dans l'usine cible
 - 24h avant l'échéance contractuelle

LÀ OÙ ÇA PÊCHE

MODE D'EMPLOI POUR DÉPLOYER

-  Passer la journée à configurer le "PC de prod"
 -  Clean install' Linux
 -  Bonne version de Python,
 -  Bons drivers GPU

LÀ OÙ ÇA PÊCHE

MODE D'EMPLOI POUR DÉPLOYER

- \$> git clone ; du repo où sont versionnés :
 - le code de ML, le code métier
 -  et le modèle

CONSTAT

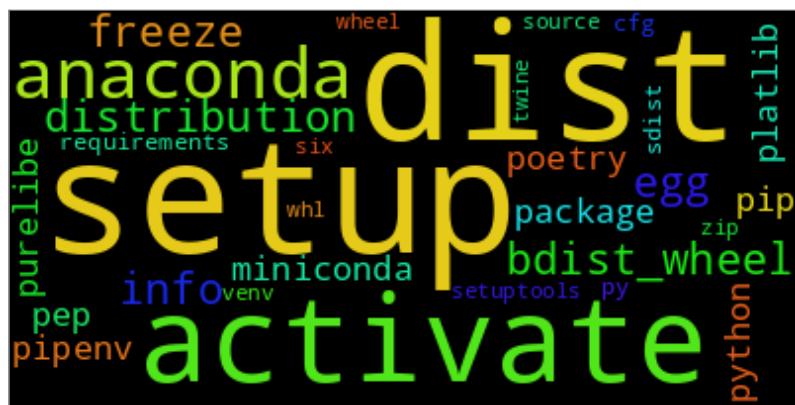


Sans investir dans l'automatisation,
L'équipe passera son temps
en déplacement pour déployer



ON COMMENCE À AUTOMATISER

- J'ai un mois pour automatiser le setup
 - 🐦 Clean install' Linux
 - 🐍 Bonne version de Python,
 - 💡 Bons drivers GPU
- Je ne comprends rien au packaging en Python 🤔



REPARTONS DES BASES DU PACKAGING

C'EST QUOI UNE APP DE MACHINE LEARNING ?

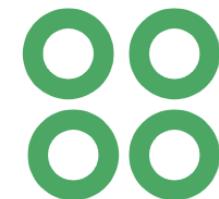


Data

Schema

Sampling over Time

Volume

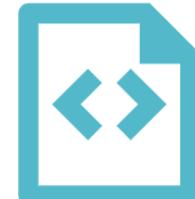


Model

Algorithms

More Training

Experiments



Code

Business Needs

Bug Fixes

Configuration

Source : [Continuous Delivery for Machine Learning](#)

J'AI COMMENCÉ SIMPLE ...

J'AI COMMENCÉ PAR UN ZIP

```
.PHONY: package ## 📦 packaging de l'application en zip
package:
    zip -r object-detector.zip \
        ${path_to_code} ${path_to_model} ${path_to_data}
```

```
.PHONY: install-app ## ⚙️ Code arbitraire à exécuter chez
vous si vous voulez vous servir de cette application
install-app: MY_OS := 'centOS'
    unzip object-detector.zip -d
${SITE_PACKAGES_DESTINATION_PATH}
# Here be dragons 🐉🐉 ...
mv ${SERVICE_FILE} ${SERVICE_PATH}
$(MAKE) compilation_en_c OS=${MY_OS}
```

PACKAGING AVEC UN ZIP

-  C'est simple, pratique, tout tient dans 1 archive
MAIS
-  Exécution de code arbitraire
-  C'est de la maintenance en +
-  On se passe des outils standards proposés par l'éco-système Python



BATTERIES INCLUDED

L'éco-système Python se veut "auto-portant", en mettant à disposition des outils prêt à l'emploi pour permettre son plein usage



LE FORMAT WHEEL



POUR ÉVITER DE RÉINVENTER LA ROUE



Wheel est le seul format officiellement préconisé à ce jour par la [Python Packaging Authority](#) (PyPA).

 Le format Wheel est un format de *packaging*,
et un package c'est :

Propriété

Example

Une **version** ecbec48 | v1.4.1

Le **code** code ML | métier

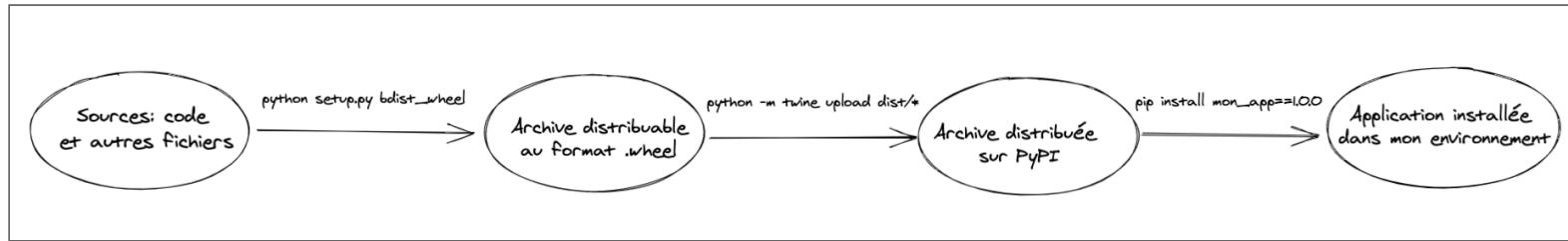
Des **metadata** pip | apt | yum
pour le *package manager*

Le tout dans une **archive** zip | tar.gz | bzip2

POINT VOCABULAIRE 😎

*Dans la littérature,
ces éléments de packaging
forment un artefact, ou un paquet*

👉 Mais en Python, on parle de **distribution**



- Les distributions Wheel se publient dans un dépôt :
 - le *Python Package Index (PyPI)*,
- Elles se génèrent avec la librairie `setuptools`,
- Elles se publient dans PyPI via la librairie `twine`
- Et se téléchargent avec la librairie `pip`.

LE SETUP, AVEC `setup.py`

```
from setuptools import setup

setup(
    name='Distutils',
    version='1.0',
    description='🧠 + 📦 = ❤️',
    author='Crafting Datascience',
    author_email='crafting.datascience@gmail.com',
    url='https://www.meetup.com/fr-FR/crafting-datascience/',
    packages=['distutils', 'distutils.command', ...],
)
```

LE SETUP, AVEC `setup.cfg`

Parce que décrire de la configuration avec du code exécutable, c'est la porte ouverte à toutes les fenêtres



-- Mehdi H. 2022

LE SETUP, AVEC `setup.cfg`

```
# setup.py
from setuptools import setup

setup()
```

```
# setup.cfg
[metadata]
name = presentation-packaging-machine-learning
version = 1.0
author = Mehdi Houacine
author_email = mehdi.houacine@octo.com
home-page = https://github.com/Mehdi-H/presentation-
packaging-machine-learning
license = <une licence in-croy-able>
description = <une description surprenante>
platform = any
classifiers =
    Programming Language :: Python :: 3
    Intended Audience :: Developers, DataScientists,
```

MLEngineers

Operating System :: OS Independent

Bug Tracker = <https://github.com/Mehdi-H/presentation-packaging-machine-learning/issues>

[options]

zip_safe = false

packages = find:

install_requires =

 numpy==1.19.4

 tensorflow==2.4.0

include_package_data = True

python_requires = >=3.8

[options.entry_points]

console_scripts =

 pkg-ml =

presentation_packaging_machine_learning.__main__:main

LE SAVIEZ-TU ?

Le `setup` peut générer un package
dans d'autres formats que .whl

```
$> python setup.py sdist --formats=gztar,zip;
```

- zip
- gztar
- bztar
- xztar
- ztar
- tar

LE SAVIEZ-TU ?

 Le format `wheel`,
c'est un `zip` avec une moustache !

A wheel is a ZIP-format archive with a specially formatted file name and the .whl extension

-- [PEP-0427](#)

LE SAVIEZ-TU ?

 Le format `wheel`,
c'est un `zip` avec une moustache !

```
$> unzip -vl pandas-1.4.3-cp310-cp310-win_amd64.whl;
```

```
Archive:  pandas-1.4.3-cp310-cp310-win_amd64.whl
      Length      Method      Size      Cmpr      Date    Time      CRC-32
      Name
-----  -----  -----  -----  -----  -----  -----
      --
      10643  Defl:N      3493   67%  06-23-2022 12:34  d5770ab4
pandas/_init__.py
      8812  Defl:N      2956   67%  06-23-2022 12:34  cc32e406
pandas/_typing.py
      518   Defl:N       357   31%  06-23-2022 12:40  b1fb299d
pandas/_version.py
     46176  Defl:N     11329   76%  06-23-2022 12:34  3b2fefd0
pandas/conftest.py
      331   Defl:N       151   54%  06-23-2022 12:33  da86e406
```

pandas/testing.py

...

PACKAGER DU ML AU FORMAT WHEEL

DÉMONSTRATION 😳

- 🧠 Un modèle de ML, c'est un fichier (.pkl, .h5, ...)
- 🛠️ `Setuptools` permet d'embarquer des fichiers dans un Wheel
- ⇒ `Setuptools` permet de packager du ML au format Wheel 🎉

STEP 1

SETUP.{PY,CFG}



```
[metadata]
name = presentation-packaging-machine-learning
version = 1.0
...
[options]
...
include_package_data = True # ➔
```

STEP 2

MANIFEST.IN



```
# MANIFEST.in

## ⤵
include
models/densenet121_weights_tf_dim_ordering_tf_kernels.h5
```

STEP 3

BUILD

```
$> python setup.py bdist_wheel;
```

ou même mieux, si vous rencontrez un
SetupToolsDeprecationWarning

```
$> python -m build --wheel;
```

TEST DE RECETTE #1

- Présence du modèle dans le .whl

```
$> unzip -vl dist/presentation_packaging_model-1.0-py3-none-any.whl
```

```
Archive: dist/presentation_packaging_model-1.0-py3-none-any.whl
```

Name	Length	Method	Size	Cmpr	Date	Time	CRC-32
	-----	-----	-----	-----	-----	-----	-----

presentation_pkg_ml/__init__.py	0	Defl:N	2	0%	07-03-2022	11:58	00000000
presentation_pkg_ml/densenet121_weights_tf_dim_ordering_tf_kernels.h5	34115984	Defl:N	30329718	11%	07-01-2022	19:53	b0ae32e5
presentation_pkg_ml/main.py	509	Defl:N	287	44%	07-03-2022	12:10	65037a2e
presentation_packaging_model-1.0.dist-info/METADATA	559	Defl:N	315	44%	07-03-2022	12:11	a9a8970e

92	Defl:N	92	0%	07-03-2022 12:11	801a68e9
presentation_packaging_model-1.0.dist-info/WHEEL					
79	Defl:N	68	14%	07-03-2022 12:11	494ecc87
presentation_packaging_model-1.0.dist-info/entry_points.txt					
20	Defl:N	22	-10%	07-03-2022 12:11	5faa91fa
presentation_packaging_model-1.0.dist-info/top_level.txt					
785	Defl:N	452	42%	07-03-2022 12:11	89e36031
presentation_packaging_model-1.0.dist-info/RECORD					
-----	-----	---			-

34118028		30330956	11%		8
files					

TEST DE RECETTE #2

- Installation en local

```
$> pip install dist/presentation_packaging_model-1.0-py3-
none-any.whl;

$> pip freeze;

keras==2.9.0
presentation-packaging-model @
file:///Users/.../presentation_packaging_model-1.0-py3-
none-any.whl
tensorflow==2.9.1
```

TEST DE RECETTE #3

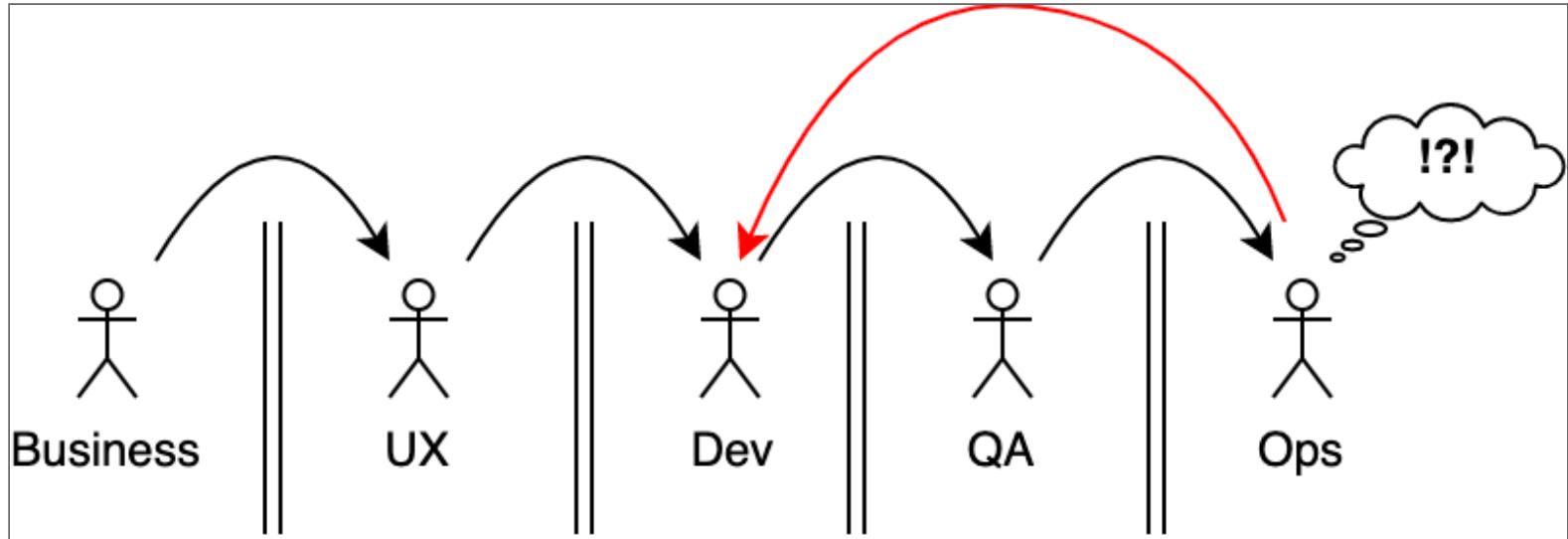
- Utilisation du point d'entrée en local

```
$> pkg-ml; ## ➡ Setup.cfg entry_point

🛠 Let's craft some ML 🧠 ...
Loading densenet121_weights_tf_dim_ordering_tf_kernels.h5
...
# print(f"👀 Printing model summary ...")
...
['conv5_block16_1_bn[0][0]']
conv5_block16_2_conv (Conv2D)      (None, 7, 7, 32)      36864
...
✅ Done ! 🎉
```

EST-CE QU'ON PEUT
FAIRE MIEUX ?





***The Wall of Confusion** : one group in a value stream approaches their job as complete when they've passed it onto the next group*

Sources : [S. Kawaguchi - The Wall of Confusion](#)

CE MUR EXISTE AUSSI
DANS LES PROJETS DE ML



LE WHEEL SEUL NE SUFFIT PAS

IL FAUDRAIT L'ACCOMPAGNER D'UNE PROCÉDURE D'INSTALLATION



- Dépendances systèmes, spécifiques à l'OS
 - Drivers GPU, paquets à installer avec apt ou yum
- Version de Python bien précise
 - python 3 à partir de la version 3.6.5
- Configuration, fichiers de service, ...

ON PEUT SE PASSER DE PROCÉDURES



AVEC DES OUTILS DE CONTENEURISATION COMME DOCKER



UNE IMAGE DOCKER



"SIMPLE"

```
FROM python:3.8.0-slim # ➡️ définition de la version de Python nécessaire, tirée du dockerhub
```

```
# ➡️ Installation de dépendances sur le système
```

```
RUN apt-get update \  
    && apt-get clean \  
    && rm -rf /var/lib/apt/lists/*
```

```
# ➡️ Packaging du modèle de machine learning
```

```
COPY densenet121_weights_tf_dim_ordering_tf_kernels.h5  
/app/
```

```
# ➡️ Packaging des sources python
```

```
COPY setup.py setup.cfg MANIFEST.in main.py /app/
```

```
# ➡️ Installation de dépendances Python via pip
```

```
WORKDIR /app
```

```
RUN pip install --user -U pip && pip install .
CMD pkg-ml
```

UNE IMAGE DOCKER 🐋

UN PEU PLUS ÉVOLUÉE

```
# 👇 étape de production du modèle :  
## il peut être généré ici ou téléchargé depuis un dépôt de  
modèles  
FROM python:3.8.0-slim as model-builder  
...  
# 👇 Génération des data  
FROM python:3.8.0-slim as data-builder  
...  
# 👇 Génération de la Wheel  
FROM python:3.8.0-slim as app-builder  
...  
# 👇 Génération de l'image résultante à publier  
FROM python:3.8.0-slim as app  
...
```

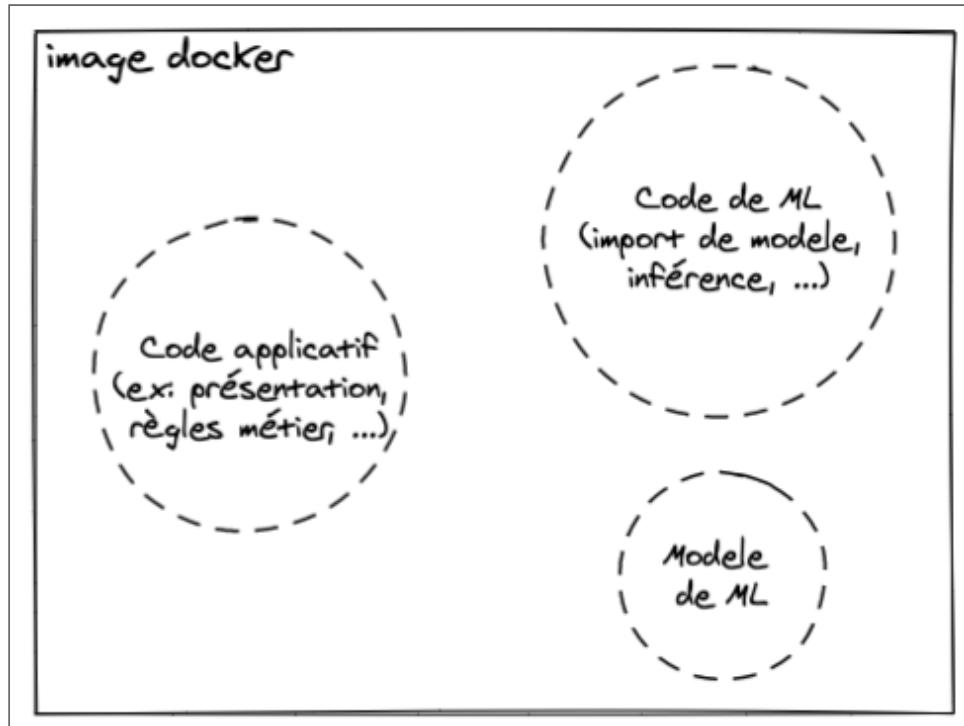
(avec *multi-stage build*)

LES STRATÉGIES DE DÉPLOIEMENT

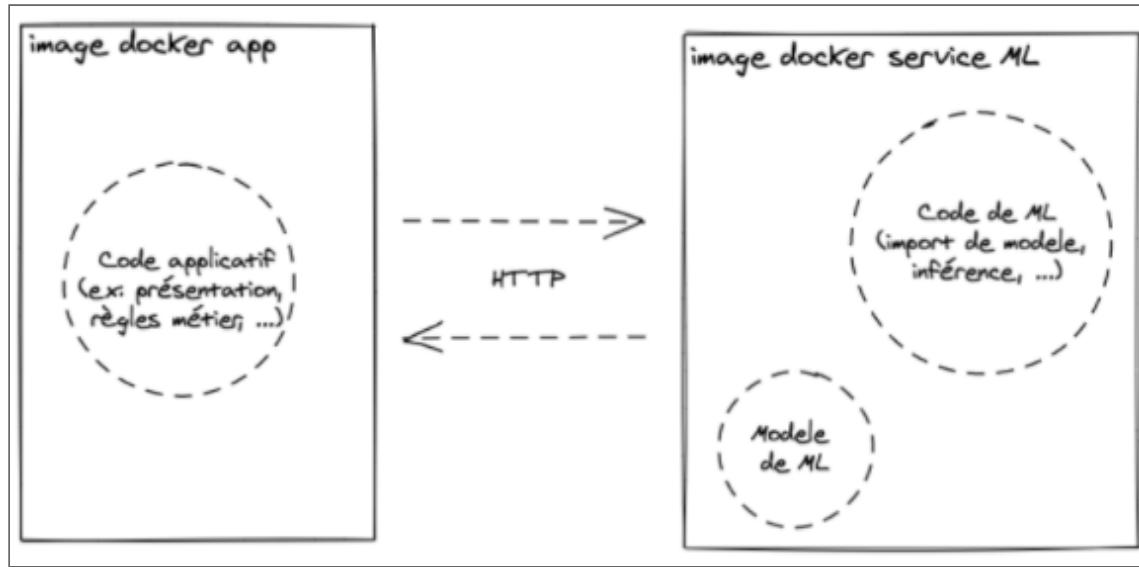


IMPACTENT LA FAÇON DE PACKAGER





Déploiement avec *modèle* embarqué



Déploiement avec *modèle isolé en tant que service*

TAKE AWAY



**VOUS N'AVEZ PAS ENVIE DE PACKAGER VOTRE APPLICATION DE ML
DANS UN ZIP 😎**

Il y a des outils dans l'éco-système Python fait pour ça



UTILISEZ LE FORMAT WHEEL
POUR ÉVITER DE RÉINVENTER LA ROUE 🐍





LE FORMAT WHEEL,

C'EST UN ZIP AVEC UNE MOUSTACHE 😎

LA STRATÉGIE DE PACKAGING

INFLUE SUR LA STRATÉGIE DE DÉPLOIEMENT

- Modèle embarqué
- Modèle isolé en tant que service

Lisez CD4ML, c'est bon pour la santé 



The screenshot shows the header of the martinFowler.com website. The logo "martinFowler.com" is on the left, followed by a search bar with a magnifying glass icon and a menu icon. Below the header, a navigation bar includes links for Refactoring, Agile, Architecture, About, Thoughtworks, a feed icon, and a Twitter icon. The main content area features a large title "Continuous Delivery for Machine Learning" in a serif font, followed by a subtitle "Automating the end-to-end lifecycle of Machine Learning applications". Below the subtitle is a detailed paragraph about the challenges of developing ML applications and how CD4ML addresses them. At the bottom of the content area, the date "19 September 2019" is visible.

Continuous Delivery for Machine Learning

Automating the end-to-end lifecycle of Machine Learning applications

Machine Learning applications are becoming popular in our industry, however the process for developing, deploying, and continuously improving them is more complex compared to more traditional software, such as a web service or a mobile application. They are subject to change in three axis: the code itself, the model, and the data. Their behaviour is often complex and hard to predict, and they are harder to test, harder to explain, and harder to improve. Continuous Delivery for Machine Learning (CD4ML) is the discipline of bringing Continuous Delivery principles and practices to Machine Learning applications.

19 September 2019

J'AI RÉDIGÉ UN ARTICLE DE BLOG SUR LE SUJET



Le Blog des Octos

STRATÉGIE SI

ARCHI & TECHNO

MÉTHODE

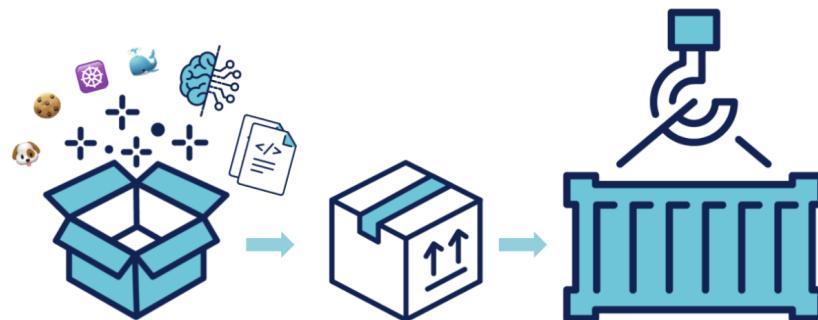
DIGITALISATION

DATA

ÉVÈNEMENT

Amener son projet de machine learning jusqu'en production avec Wheel et Docker

Posté le 02/03/2021 par Mehdi Houacine



Ca parle de packaging, de muffins et de chihuahua 🐶