

WEBINAIRE

REPRODUCTIBILITÉ EN APPRENTISSAGE AUTOMATIQUE

30 OCTOBRE 2020



OBJECTIFS DE LA PRÉSENTATION

- Sensibiliser sur les enjeux de la reproductibilité.
 - Inciter l'intégration des solutions permettant une meilleure reproductibilité dans vos solutions d'affaires ou académiques.
 - Améliorer votre productivité.
- 



VOTRE CONFÉRENCIER



DAVID BEAUCHEMIN

Candidat au doctorat

Département d'informa-
tique et de génie logiciel

- Introduit à la recherche reproductible en 2016 (R Markdown et Git)
- Participation à REPROLANG de la conférence LREC [Garneau et al., 2020]
- Membre actif dans le développement d'une librairie facilitant la reproductibilité ([Poutyne](#))



Introduction



C'EST QUOI LA REPRODUCTIBILITÉ?

La reproductibilité est le principe qu'on ne peut tirer de conclusions que d'un événement bien décrit, qui est apparu plusieurs fois, provoqué par des **personnes différentes**.

Toutefois, ont utilise souvent ce terme pour spécifiquement désigner la **réplicabilité**. Soit la réplication (reproduction) des résultats d'un article dans des environnements pas (toujours) différents [Drummond, 2009, Pineau et al., 2020].





AU MENU



Répliquer





AU MENU



Répliquer



Jeux de données





AU MENU



Répliquer



Jeux de données



Procédure
d'entraînement





AU MENU



Répliquer



Jeux de données



Procédure
d'entraînement



Code



POURQUOI S'Y INTÉRESSER ?

70 %¹

1. [Baker, 2016]





POURQUOI S'Y INTÉRESSÉ?

50 %¹

1. [Baker, 2016]





POURQUOI S'Y INTÉRESSÉ?

40 %²

2. [Raff, 2019]





MOTIVATION



Réutilisation





MOTIVATION



Réutilisation



Productivité





MOTIVATION



Réutilisation



Productivité



Transfert





MOTIVATION



Réutilisation



Productivité



Transfert



Se faire connaître



Les barrières à la réplicabilité



Non-disponibilité des
données



Non-disponibilité des
données



Mauvaise
spécification



Non-disponibilité des données



Mauvaise spécification



Non-disponibilité du code



Non-disponibilité des données



Mauvaise spécification



Non-disponibilité du code



Non-disponibilité des configurations

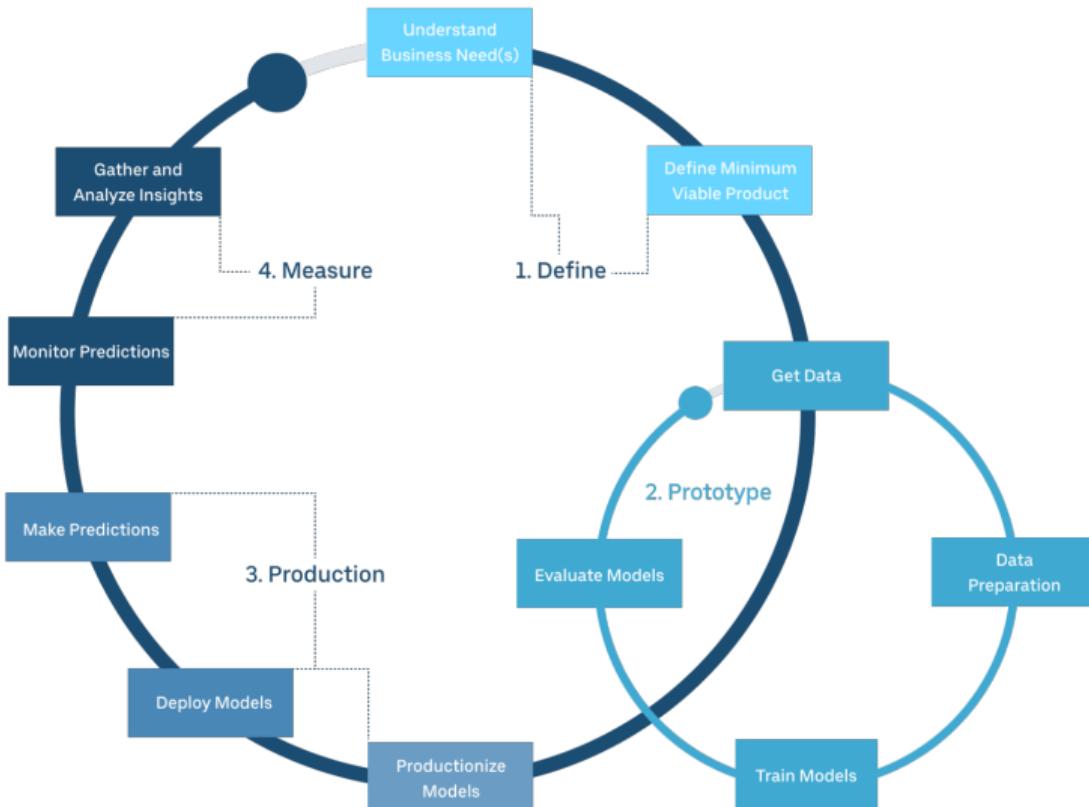


Figure 1 – From Uber Engineering

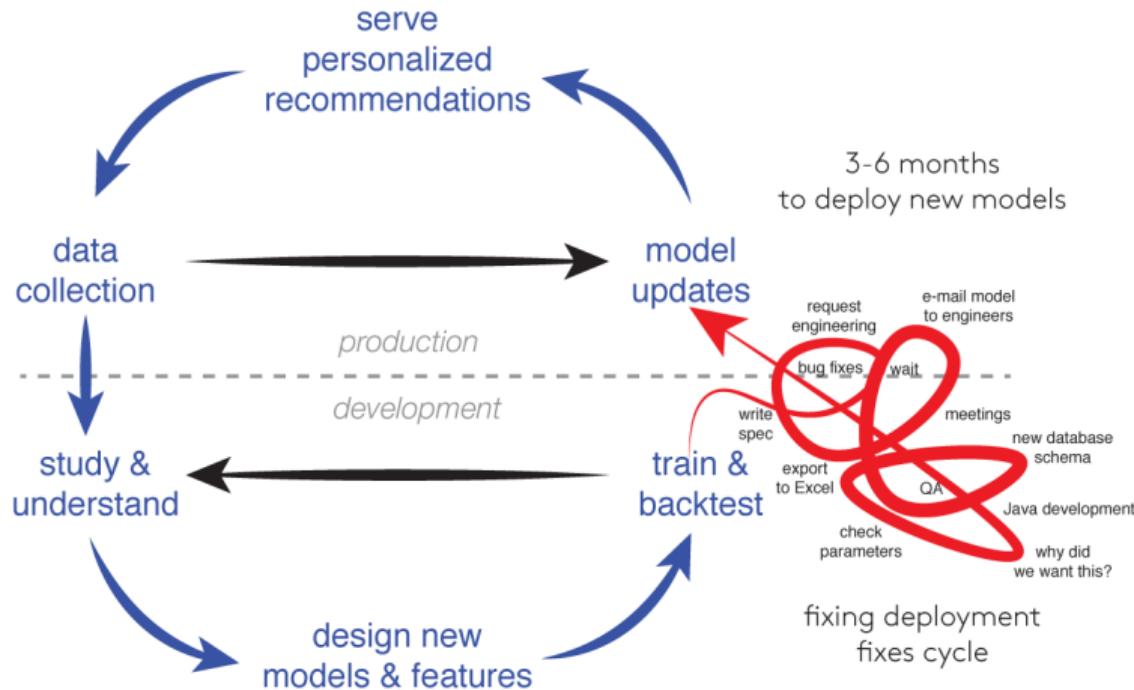


Figure 2 – The need for Agile machine learning

OK, mais comment?

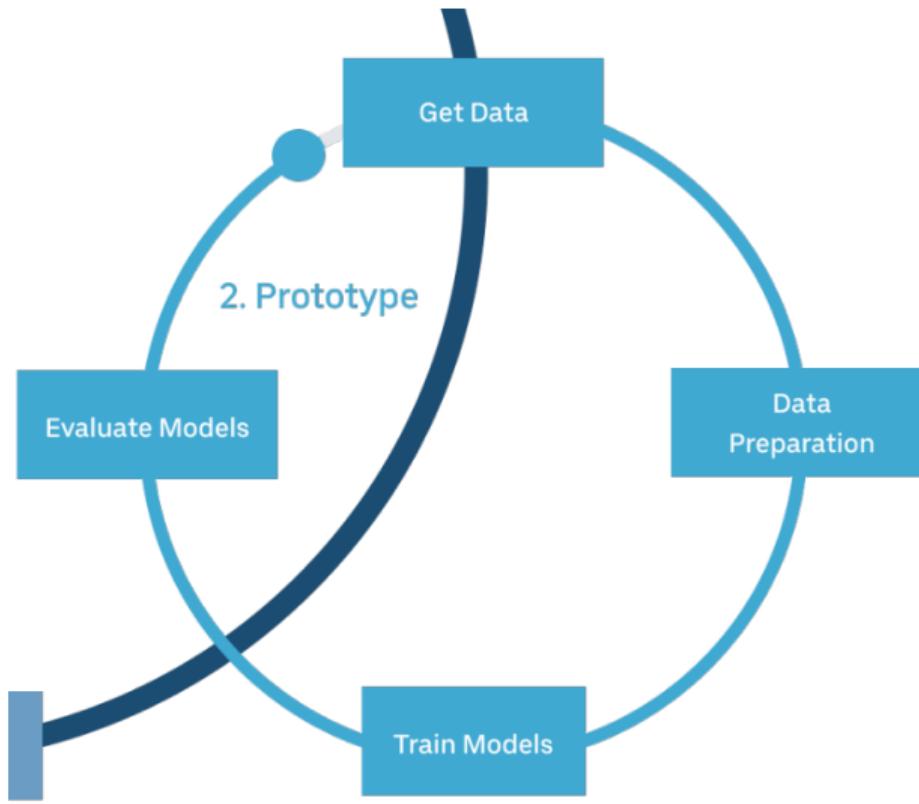


Figure 3 – From Uber Engineering



VERSION DES DONNÉES & ÉTAPES DE PRÉTRAITEMENT





VERSION DES DONNÉES & ÉTAPES DE PRÉTRAITEMENT



Gestion des versions





VERSION DES DONNÉES & ÉTAPES DE PRÉTRAITEMENT



Gestion des versions



Étapes prétraitement



VERSION DES DONNÉES & ÉTAPES DE PRÉ PROCESSION

- Data Version Control (**DVC**)
 - Dask [Dask Development Team, 2016]
- 



CODE



Version





CODE



Version



Différence





CODE



Version



Différence



Divergences



CODE

- Git





DÉVELOPPEMENT DES MODÈLES



Réinventer





DÉVELOPPEMENT DES MODÈLES



Réinventer



Simplification





DÉVELOPPEMENT DES MODÈLES



Réinventer



Simplification



Facilite





DÉVELOPPEMENT DES MODÈLES

- Poutyne [Paradis et al., 2020]
 - PyTorch Lightning [Falcon, 2019]
 - Scikit-Learn [Buitinck et al., 2013]
 - Gensim [Řehůřek and Sojka, 2010]
 - Allen NLP [Gardner et al., 2017]
- 



ENTRAÎNEMENT, CONFIGURATION ET RÉSULTATS



Version de
l'entraînement





ENTRAÎNEMENT, CONFIGURATION ET RÉSULTATS



Version de
l'entraînement



Résultats





ENTRAÎNEMENT, CONFIGURATION ET RÉSULTATS



Version de
l'entraînement



Résultats



Visualisation





ENTRAÎNEMENT, CONFIGURATION ET RÉSULTATS



Version de
l'entraînement



Résultats



Visualisation



Erreurs
d'entraînement



ENTRAÎNEMENT, CONFIGURATION ET RÉSULTATS

- MLFlow [Zaharia et al., 2018]
 - Hydra [Yadan, 2019]
 - Sacred [Greff et al., 2017]
 - Notif [Beauchemin, 2019]
- 



RAPPORT ET ANALYSE DES RÉSULTATS



Tableau des résultats





RAPPORT ET ANALYSE DES RÉSULTATS



Tableau des résultats



Mise à jour





RAPPORT ET ANALYSE DES RÉSULTATS



Tableau des résultats



Mise à jour



Visualisation configuration



RAPPORT ET ANALYSE DES RÉSULTATS

- Python2LaTeX
- TensorBoard
- Markdown





ENVIRONNEMENT



Différents environnements





ENVIRONNEMENT



Différents environnements



Réutilisation





ENVIRONNEMENT

- Docker
- 

La suite



Itérations d'expérimentations



POUR ALLER PLUS LOIN

- Clean code
 - Continuous Machine Learning
 - Faire des tests!
 - Writing Code for NLP Research [Gardner et al., 2018]
 - SOLID
 - Cet article [Pineau et al., 2020]
- 



PÉRIODE DE QUESTIONS



WEBINAIRE

MERCI DE VOTRE ÉCOUTE !



REFERENCES i

-  Baker, M. (2016).
1,500 Scientists Lift the Lid on Reproducibility.
Nature News, 533(7604) :452.
 -  Beauchemin, D. (2019).
Notif - The notification package.
<https://notificationdoc.ca/>.
- 



REFERENCES ii

-  Buitinck, L., Louppe, G., Blondel, M., Pedregosa, F., Mueller, A., Grisel, O., Niculae, V., Prettenhofer, P., Gramfort, A., Grobler, J., Layton, R., VanderPlas, J., Joly, A., Holt, B., and Varoquaux, G. (2013).
API design for machine learning software : experiences from the scikit-learn project.
In *ECML PKDD Workshop : Languages for Data Mining and Machine Learning*, pages 108–122.
 -  Dask Development Team (2016).
Dask : Library for dynamic task scheduling.
 -  Drummond, C. (2009).
Replicability Is Not Reproducibility : Nor Is It Good Science.
Evaluation Methods for Machine Learning Workshop.
- 



REFERENCES iii

-  Falcon, W. (2019).
PyTorch Lightning.
GitHub. Note : <https://github.com/PyTorchLightning/pytorch-lightning>, 3.
 -  Gardner, M., Grus, J., Neumann, M., Tafjord, O., Dasigi, P., Liu, N. F., Peters, M., Schmitz, M., and Zettlemoyer, L. S. (2017).
AllenNLP : A Deep Semantic Natural Language Processing Platform.
 -  Gardner, M., Neumann, M., Grus, J., and Lourie, N. (2018).
Writing Code for NLP Research.
In *Conference on Empirical Methods in Natural Language Processing : Tutorial Abstracts*.
- 



REFERENCES iv

- Garneau, N., Godbout, M., Beauchemin, D., Durand, A., and Lamontagne, L. (2020).
A Robust Self-Learning Method for Fully Unsupervised Cross-Lingual Mappings of Word Embeddings : Making the Method Robustly Reproducible as Well.
- Greff, K., Klein, A., Chovanec, M., Hutter, F., and Schmidhuber, J. (2017).
The Sacred Infrastructure for Computational Research.
pages 49–56.
- Paradis, F., Beauchemin, D., Godbout, M., Alain, M., Garneau, N., Otte, S., Tremblay, A., Bélanger, M.-A., and Laviolette, F. (2020).
Poutyne : A Simplified Framework for Deep Learning.
<https://poutyne.org>.



REFERENCES v

-  Pineau, J., Vincent-Lamarre, P., Sinha, K., Larivière, V., Beygelzimer, A., d'Alché Buc, F., Fox, E., and Larochelle, H. (2020).
Improving Reproducibility in Machine Learning Research (A Report from the NeurIPS 2019 Reproducibility Program).
 -  Raff, E. (2019).
A Step Toward Quantifying Independently Reproducible Machine Learning Research.
 -  Řehůřek, R. and Sojka, P. (2010).
Software Framework for Topic Modelling with Large Corpora.
In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50.
- 



REFERENCES vi

-  Yadan, O. (2019).
Hydra - A framework for elegantly configuring complex applications.
 -  Zaharia, M., Chen, A., Davidson, A., Ghodsi, A., Hong, S., Konwinski, A., Murching, S., Nykodym, T., Ogilvie, P., Parkhe, M., Xie, F., and Zumar, C. (2018).
Accelerating the Machine Learning Lifecycle with MLflow.
IEEE Data Engineering Bulletin, 41:39–45.
- 