

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, on 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].





EN SOMME

- Être capable de **répliquer** les résultats d'un article/ d'un projet,
- 



EN SOMME

- Être capable de **répliquer** les résultats d'un article/ d'un projet,
 - à partir du **même jeu de données** ou un jeu de données différent (mais proche),
- 



EN SOMME

- Être capable de **répliquer** les résultats d'un article/ d'un projet,
 - à partir du **même jeu de données** ou un jeu de données différent (mais proche),
 - en utilisant la **procédure d'entraînement** de l'article ou en utilisant notre procédure d'entraînement et
- 



EN SOMME

- Être capable de **répliquer** les résultats d'un article/ d'un projet,
 - à partir du **même jeu de données** ou un jeu de données différent (mais proche),
 - en utilisant la **procédure d'entraînement** de l'article ou en utilisant notre procédure d'entraînement et
 - en utilisant le **code du projet**.
- 



POURQUOI S'Y INTÉRESSÉ?

- 70 % des chercheurs en science ont échoué dans leur tentative de reproduire un article d'un autre chercheur,
- 



POURQUOI S'Y INTÉRESSÉ?

- 70 % des chercheurs en science ont échoué dans leur tentative de reproduire un article d'un autre chercheur,
 - 50 % n'ont pas réussi à reproduire leurs **propres** expérimentations [Baker, 2016].
- 



POURQUOI S'Y INTÉRESSÉ?

- 70 % des chercheurs en science ont échoué dans leur tentative de reproduire un article d'un autre chercheur,
- 50 % n'ont pas réussi à reproduire leurs **propres** expérimentations [Baker, 2016].

L'informatique ne fait pas exception à cela malgré la simplicité (théorique) de réPLICATION des résultats. Selon une étude, sur 255 articles près de 40 % n'était pas réPLICABLE [Raff, 2019].





MOTIVATION

La reproductibilité facilite la réutilisation pour d'autres projets de recherche, améliorer votre productivité **et** permet le transfert vers l'industrie (plus facilement).



Les barrières à la réplicabilité

- Non-disponibilité du jeu de données ou **version** (pas clair) du jeu de données,

- Non-disponibilité du jeu de données ou **version** (pas clair) du jeu de données,
- **mauvaise spécification ou sous-spécification** du modèle et de la **procédure d'entraînement**,

- Non-disponibilité du jeu de données ou **version** (pas clair) du jeu de données,
- **mauvaise spécification ou sous-spécification** du modèle et de la **procédure d'entraînement**,
- manque de **disponibilité du code** nécessaire pour exécuter les expériences, ou **erreurs** dans le code,

- Non-disponibilité du jeu de données ou **version** (pas clair) du jeu de données,
- **mauvaise spécification ou sous-spécification** du modèle et de la **procédure d'entraînement**,
- manque de **disponibilité du code** nécessaire pour exécuter les expériences, ou **erreurs** dans le code,
- **configurations** du modèle déficientes [Pineau et al., 2020]¹.

1. Liste sélective

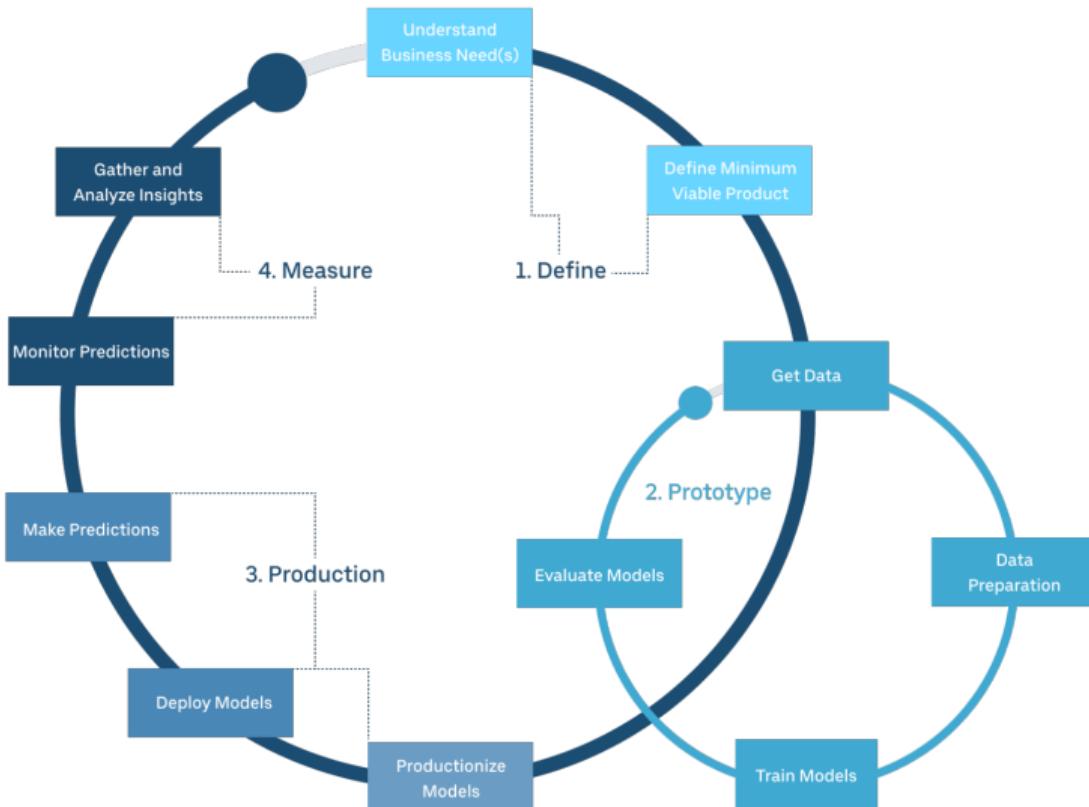


Figure 1 – From Uber Engineering

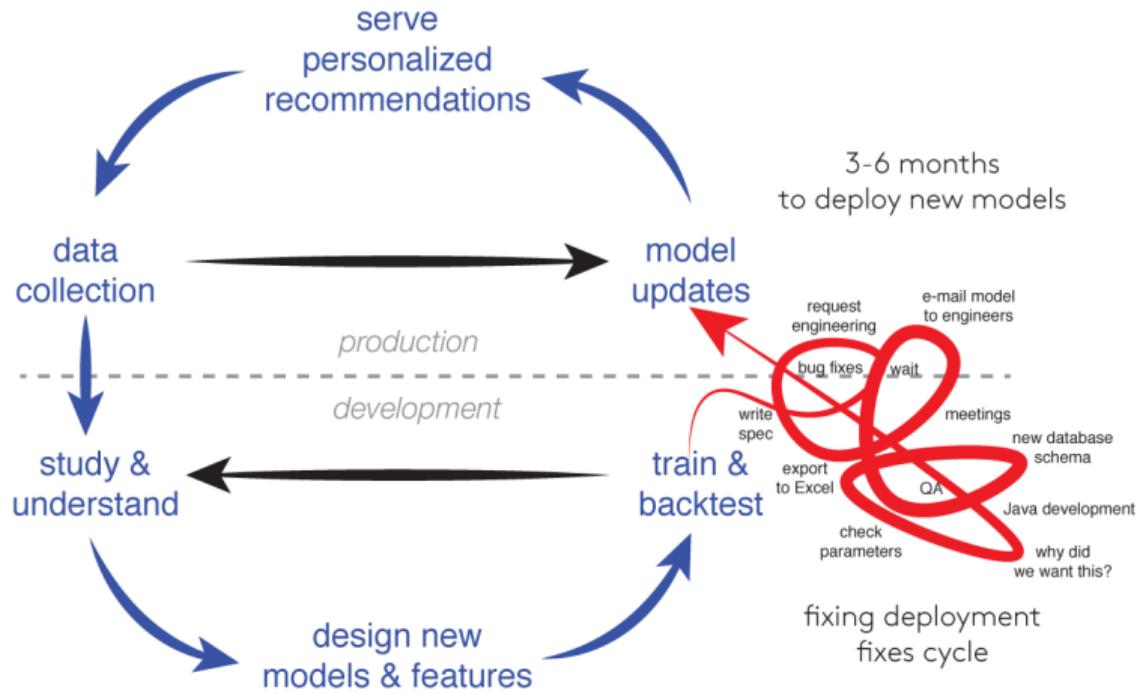


Figure 2 – The need for Agile machine learning

OK, mais comment?

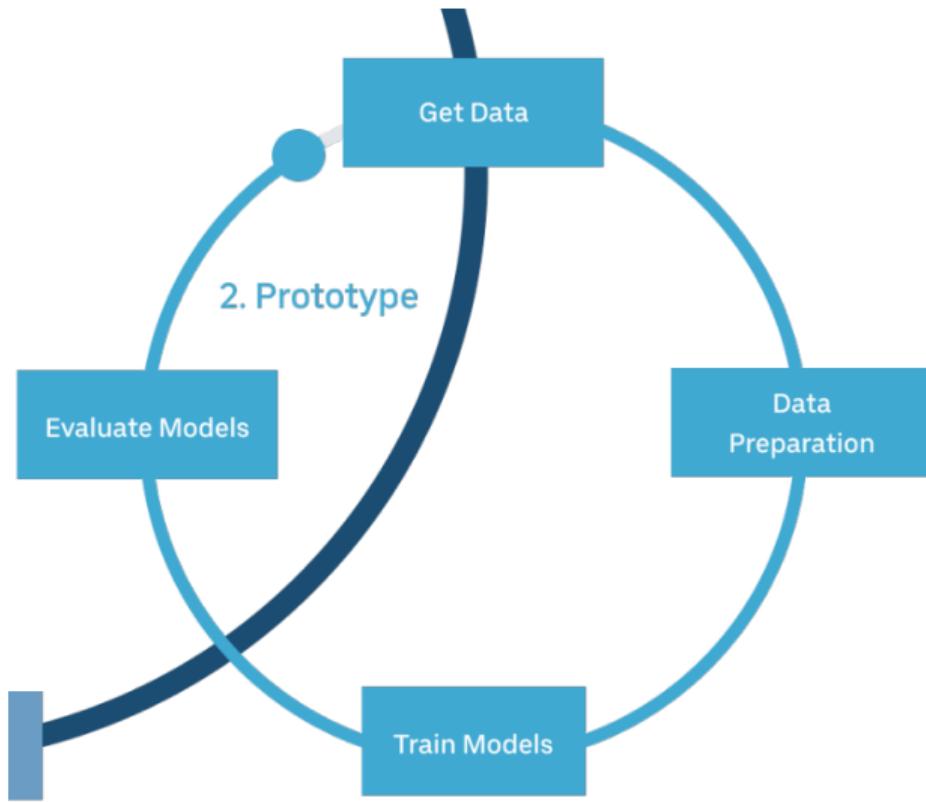


Figure 3 – From Uber Engineering



VERSION DES DONNÉES & ÉTAPES DE PRÉ PROCESSION

- Avec quelle version des données doit-on travailler?





VERSION DES DONNÉES & ÉTAPES DE PRÉ PROCESSION

- Avec qu'elle version des données doit-on travailler?
 - Comment gérer **facilement** plusieurs versions des données?
- 



VERSION DES DONNÉES & ÉTAPES DE PRÉ PROCESSION

- Avec qu'elle version des données doit-on travailler?
 - Comment gérer **facilement** plusieurs versions des données?
 - Comment définir **facilement** les étapes de pré procession des données?
- 



VERSION DES DONNÉES & ÉTAPES DE PRÉ PROCESSION

- Avec qu'elle version des données doit-on travailler?
- Comment gérer **facilement** plusieurs versions des données?
- Comment définir **facilement** les étapes de pré procession des données?

Il nous faut des ***data pipelines***, des tuyaux que nous pouvons raccorder **facilement** à nos modèles pour l'entraînement et la mise en production, par exemple, Data Version Control (**DVC**).





VERSION DU CODE

- Avec quelle version du code doit-on travailler?





VERSION DU CODE

- Avec qu'elle version du code doit-on travailler?
 - Comment savoir **rapidement** qu'elle est la différence d'implémentation entre deux versions du modèle?
- 



VERSION DU CODE

- Avec qu'elle version du code doit-on travailler?
 - Comment savoir **rapidement** qu'elle est la différence d'implémentation entre deux versions du modèle?
 - Comment gérer **facilement** les embranchements d'expérimentations?
- 



VERSION DU CODE

- Avec qu'elle version du code doit-on travailler?
- Comment savoir **rapidement** qu'elle est la différence d'implémentation entre deux versions du modèle?
- Comment gérer **facilement** les embranchements d'expérimentations?

Il nous faut un outil nous permettant de **visualiser** la différence entre des fichiers de code et nous permettant d'avoir **plusieurs** versions du code « **en même temps** », par exemple, **Git**.





DÉVELOPPEMENT DES MODÈLES

- Ne pas réinventer la roue.



DÉVELOPPEMENT DES MODÈLES

- Ne pas réinventer la roue.
- Simplifier l'écriture de code pour développer des modèles.





DÉVELOPPEMENT DES MODÈLES

- Ne pas réinventer la roue.
 - Simplifier l'écriture de code pour développer des modèles.
 - Qui facilite l'entraînement (GPU, multi-GPU/CPU).
- 



DÉVELOPPEMENT DES MODÈLES

- Ne pas réinventer la roue.
- Simplifier l'écriture de code pour développer des modèles.
- Qui facilite l'entraînement (GPU, multi-GPU/CPU).

Il nous faut des outils nous permettant de **simplifier le développement** de nos modèles, par exemple, Poutyne [Paradis et al., 2020], PyTorch Lightning [Falcon, 2019], Scikit-Learn [Buitinck et al., 2013], Gensim [Řehůřek and Sojka, 2010] et Allen NLP [Gardner et al., 2017].





ENTRAÎNEMENT, CONFIGURATION ET RÉSULTATS

- Avec quelle version du code, du modèle et des données avons-nous fait cet entraînement?





ENTRAÎNEMENT, CONFIGURATION ET RÉSULTATS

- Avec quelle version du code, du modèle et des données avons-nous fait cet entraînement?
 - Quels sont les résultats?
- 



ENTRAÎNEMENT, CONFIGURATION ET RÉSULTATS

- Avec quelle version du code, du modèle et des données avons-nous fait cet entraînement?
 - Quels sont les résultats?
 - Comment visualiser **rapidement** les résultats et les paramètres de configuration?
- 



ENTRAÎNEMENT, CONFIGURATION ET RÉSULTATS

- Avec quelle version du code, du modèle et des données avons-nous fait cet entraînement?
- Quels sont les résultats?
- Comment visualiser **rapidement** les résultats et les paramètres de configuration?

Il nous faut des outils nous permettant de **logger** les **paramètres** d'entraînement et les **résultats**, par exemple, MLFlow [Zaharia et al., 2018] et Sacred [Greff et al., 2017].





RAPPORT ET ANALYSE DES RÉSULTATS

- Comment créer des tableaux de résultats **facilement** (pas à la *mitaine*) ?
- 



RAPPORT ET ANALYSE DES RÉSULTATS

- Comment créer des tableaux de résultats **facilement** (pas à la *mitaine*) ?
 - Comment s'assurer **facilement** que les résultats sont à jour ?
- 



RAPPORT ET ANALYSE DES RÉSULTATS

- Comment créer des tableaux de résultats **facilement** (pas à la *mitaine*) ?
 - Comment s'assurer **facilement** que les résultats sont à jour ?
 - Comment visualiser **rapidement** les résultats et les paramètres de configuration ?
- 



RAPPORT ET ANALYSE DES RÉSULTATS

- Comment créer des tableaux de résultats **facilement** (pas à la *mitaine*) ?
- Comment s'assurer **facilement** que les résultats sont à jour ?
- Comment visualiser **rapidement** les résultats et les paramètres de configuration ?

Il nous faut des outils nous permettant de **créer** des tableaux de résultats **à même les résultats**, soit de diminuer le plus possible le travail manuel, par exemple, **Python2LaTeX** et **Markdown**.





DOCKERISATION

- Comment s'assurer que nos modèles fonctionnent sur d'autres environnements ?
- 



DOCKERISATION

- Comment s'assurer que nos modèles fonctionnent sur d'autres environnements ?
 - Comment faciliter la réutilisation de notre code ?
- 



DOCKERISATION

- Comment s'assurer que nos modèles fonctionnent sur d'autres environnements ?
- Comment faciliter la réutilisation de notre code ?

Docker!



La suite



Développer des processus rigoureux (par essai et erreur) et ne pas prendre tout ce qui a été discuté ici comme l'unique solution.



POUR ALLER PLUS LOIN

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



PÉRIODE DE QUESTIONS





REFERENCES i

-  Baker, M. (2016).
1,500 scientists lift the lid on reproducibility.
Nature News, 533(7604) :452.
 -  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.
- 



REFERENCES ii

-  Drummond, C. (2009).
Replicability is not reproducibility : Nor is it good science.
Proceedings of the Evaluation Methods for Machine Learning Workshop at the 26th ICML.
 -  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.
- 



REFERENCES iii

-  Gardner, M., Neumann, M., Grus, J., and Lourie, N. (2018).
Writing code for NLP research.
In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing : Tutorial Abstracts*, Melbourne, Australia. Association for Computational Linguistics.
 -  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.
- 



REFERENCES iv

-  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>.
 -  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.
- 

REFERENCES v

-  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, Valletta, Malta. ELRA.
<http://is.muni.cz/publication/884893/en>.
-  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 Eng. Bull., 41 :39–45.

WEBINAIRE

MERCI DE VOTRE ÉCOUTE !

