Outils informatiques pour la science des données : Python

ÉCOLE MATHÉMATIQUE AFRICAINE :
Bases mathématiques de l'intelligence artificielle

1 – 19 Juillet 2019, Université de Yaoundé I, Yaoundé, Cameroun

Présentation Ivan Keller - keller.ivan@gmail.com

- Master de Maths Proba/Stats (Paris) + Master d'IA (KULeuven-Belgique)
- Collaboration avec Pr. Emmanuel Viennet comme Ingénie de recherche
  - Analyse de réseaux sociaux
  - Recommandation dans les musées
- "Data scientist" chez AXA Assurance Belgique depuis 2016
  - Classification automatique d'emails
  - Modèles de risques
  - Développement d'outils d'aides à la décision basés sur des données
  - Speech-to-Text
- Technologies :
  - Python, Scala, Spark
  - Data lake (cluster Hadoop), Cloud (AWS)

Plan pour les deux demi-journées

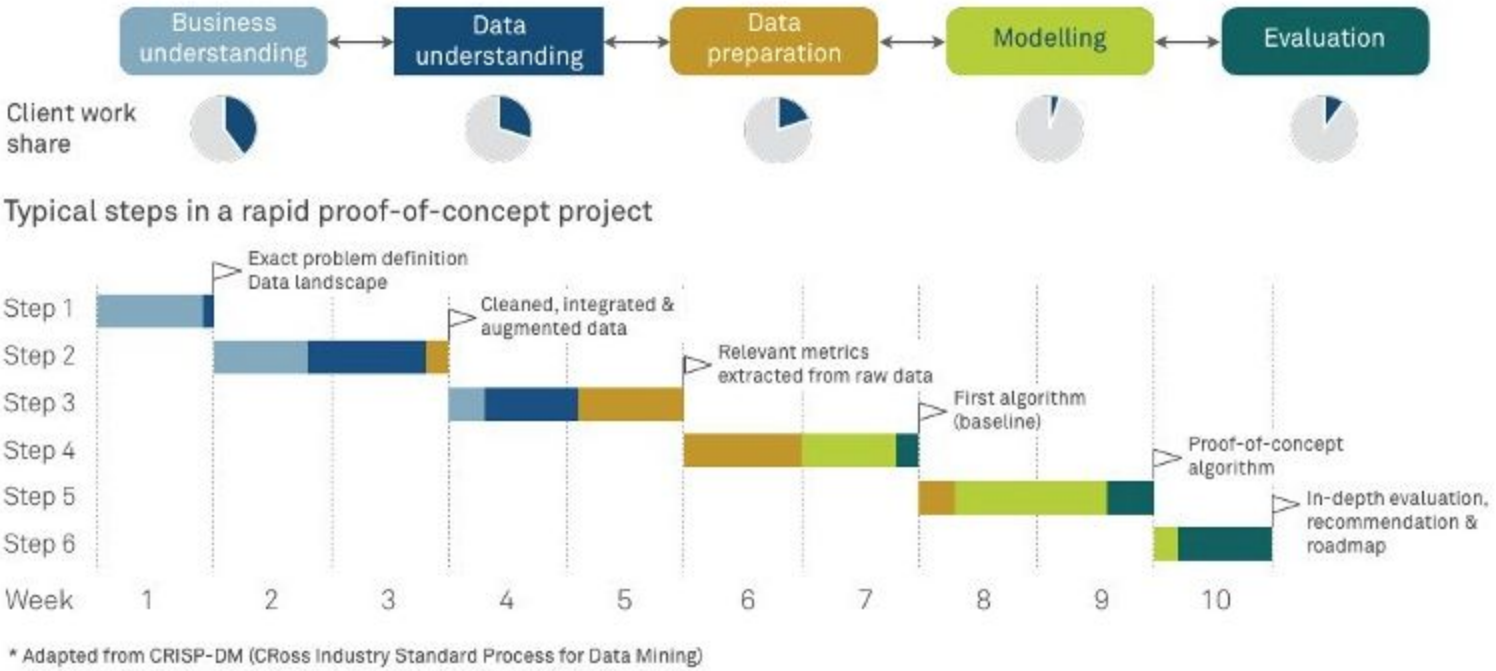
- Mon expérience de la data science dans l'industrie
- Python pour la data-science
  - Introduction
  - Rappels de Python
  - NumPy
  - Visualisation de données : Matplotlib
  - **TP1**: Numpy et visualisation
  - Pandas
  - **TP2**: Pandas
  - Scikit-Learn
  - **TP3**: Scikit-Learn
- Conclusion

Mon expérience de la data science dans l'industrie
AXA Assurance (Belgique)

Méthodes

1. Idée, question à laquelle répondre, problème à résoudre avec des données.
2. Première étude de faisabilité : on essaye rapidement si possible.
3. PoC: “Proof of Concept”: démonstration de faisabilité (prototypage)
4. Si ça marche on industrialise par itérations (méthode CRISP-DM, Agile)
    a. Développement itératif
    b. Déploiement et intégration
    c. Maintenance
5. Dans la pratique on fait souvent un peu de tout (dépend de la maturité de l’entreprise et du domaine avec ces technologies)

Déroulement d'un PoC



source: https://medium.com/@liwdai/data-science-rapid-proof-of-concept-projects-5e8414e10777

Obtenir et préparer des données

Difficile d'obtenir des données:

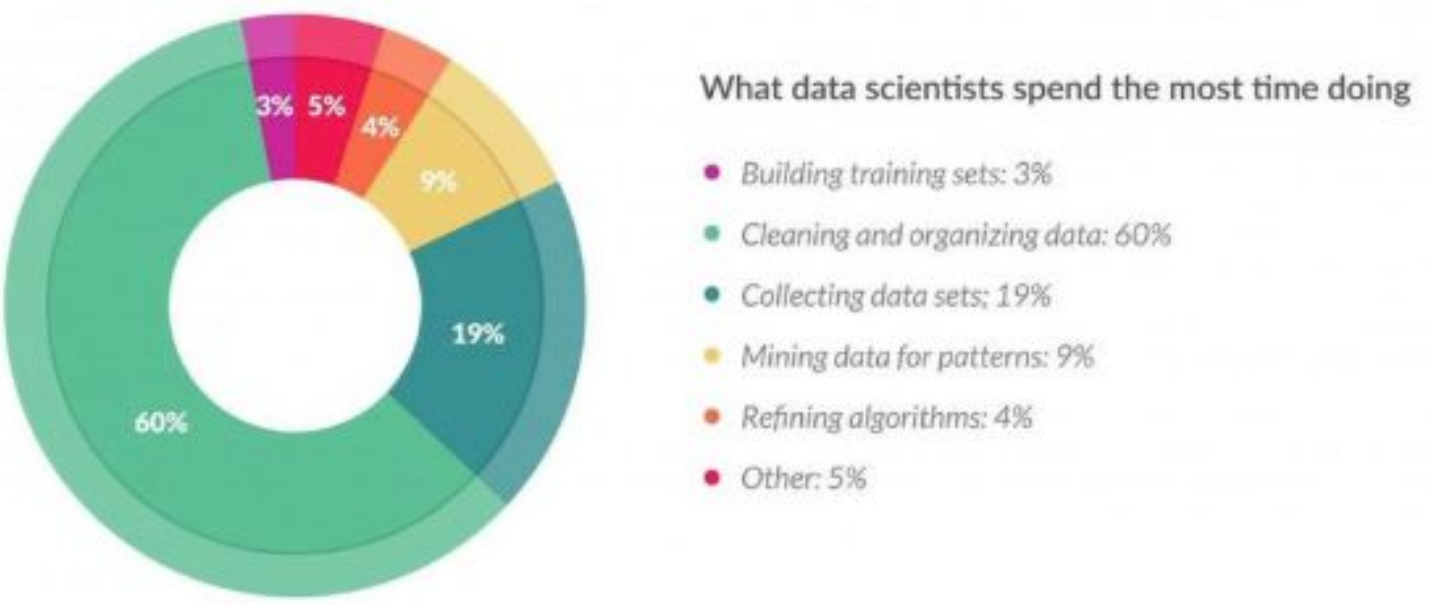
- Lenteurs des processus internes
- Contraintes légales (RGDP)
- Technologies variées et données peu documentées
- Données sales

Donc on passe beaucoup de temps à avoir un bon jeu de données avant de pouvoir l'utiliser. On construit des processus de transformation des données :

ETL = "Extract Transform Load"

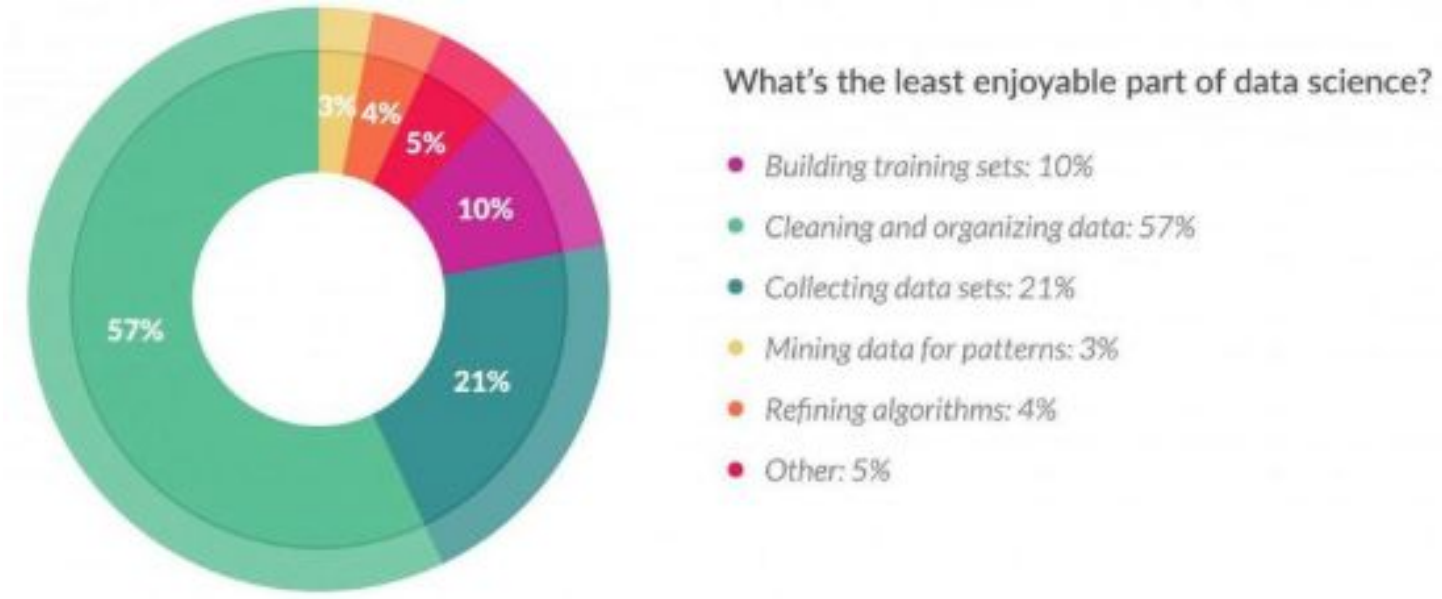
Les jobs recherchés : **data-engineer, machine-learning engineer**

À quoi les data-scientists passe-t-ils leur temps ?

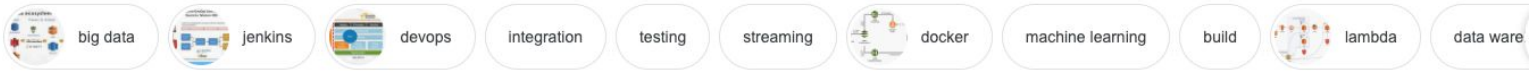


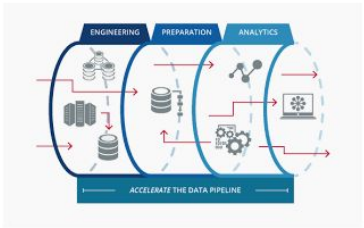
Source:
https://www.forbes.com/sites/gilpress/2016/03/23/data-preparation-most-time-consuming-least-enjoyable-data-science-task-survey-says/#1b963786f637

Qu'est-ce qu'ils apprécient le moins de faire ?

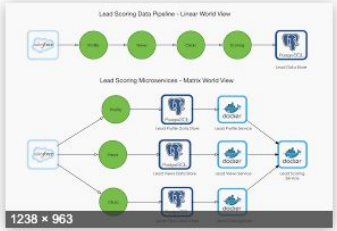


Chercher “Data Pipeline” dans une moteur de recherche



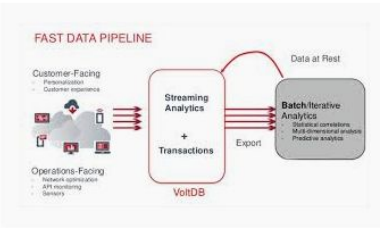


Building a Data Pipeline from Scratch ...
medium.com

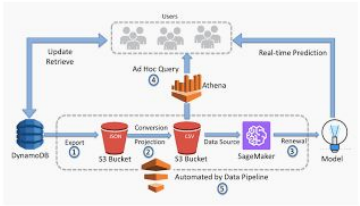


let friends build data pipelines
abhishek-tiwari.com

Building a Cloud Data Pipeline ...
youtube.com

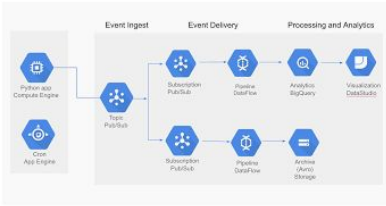


Fast Data Pipeline Design: Updating Per ...
dzone.com



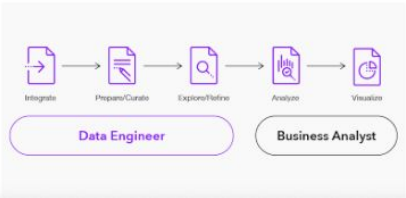
AWS Data Pipeline | AWS Big Data Blog
aws.amazon.com

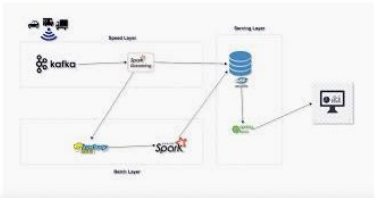
Data Pipeline with Databricks Delta ...
databricks.com

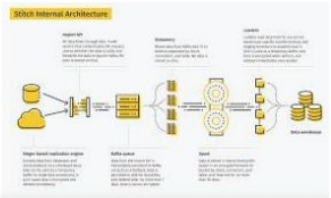


Building a data pipeline on GCP — noob ...
medium.com

A fast, serverless, big data pipeline ...
azure.microsoft.com







Python pour la science des données

Planning

- 1ére demi-journée :
  - Une très brève histoire de Python
  - Rappels Python (rapidement)
  - Manipuler des données (numériques) tabulaires : **Numpy**
  - Visualisation: **Matplotlib**
  - **TP1 :** Numpy/Matplotlib
- 2ème demi-journée :
  - Données structurées variées tabulaires : **Pandas**
  - **TP2** : Pandas
  - Modélisation : **Scikit-learn**
  - **TP3 :** scikit-learn

Brève histoire de Python

Guido van Rossum
ex-Python's Benevolent Dictator For Life (BDFL)

- 1989 : Création par Guido van Rossum (CWI Université d'Amsterdam) comme alternative entre C et un langage de script (genre bash pour Linux).
- 2000 : Python 2
- 2008 : Python 3 pas vraiment compatible avec Python 2
- 2018 : changement de gouvernance dans le développement du langage
- 2020 : fin de maintenance de Python 2.7

Python pour la science des données

- 1995 – **NumPy**: fast vector and matrix algebra
- 2001 – SciPy: optimization, integration, FFT, ODEs, …
- 2007 – **Matplotlib**: MATLAB-style plotting
- 2007 – **Scikit-Learn**: machine learning
- 2007 – **IPython**: improved REPL, notebooks for sharing & presenting
- 2007 – **Pandas**: data frames for data wrangling
- 2012 – pySpark: big data / distributed computing
- 2015 – **Jupyter** notebook, lab, hub

Python et R, Matlab, SAS, Scala (Spark), bases de données (SQL)

Aller à https://github.com/ivankeller/ema

En conclusion

- Nous avons vu introduit les principaux outils Python couramment utilisés en Data-science : Numpy, Matplotlib, Pandas, Scikit-Learn
- Domaine en évolution constante, très dynamique, tenez-vous au courant !
- Python est facile à apprendre et la communauté en ligne est riche en ressources
- Pour apprendre à faire bon usage des notebooks : https://jakevdp.github.io/blog/2017/03/03/reproducible-data-analysis-in-jupyter/