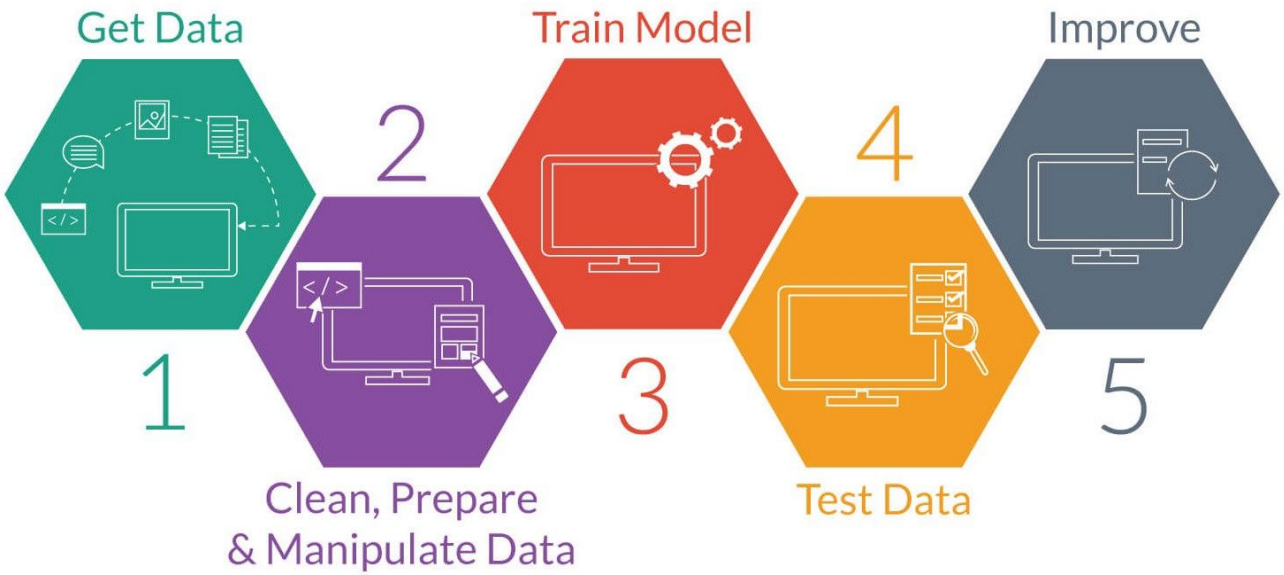
Mettre en place un transition IA

Développer un projet en Machine Learning

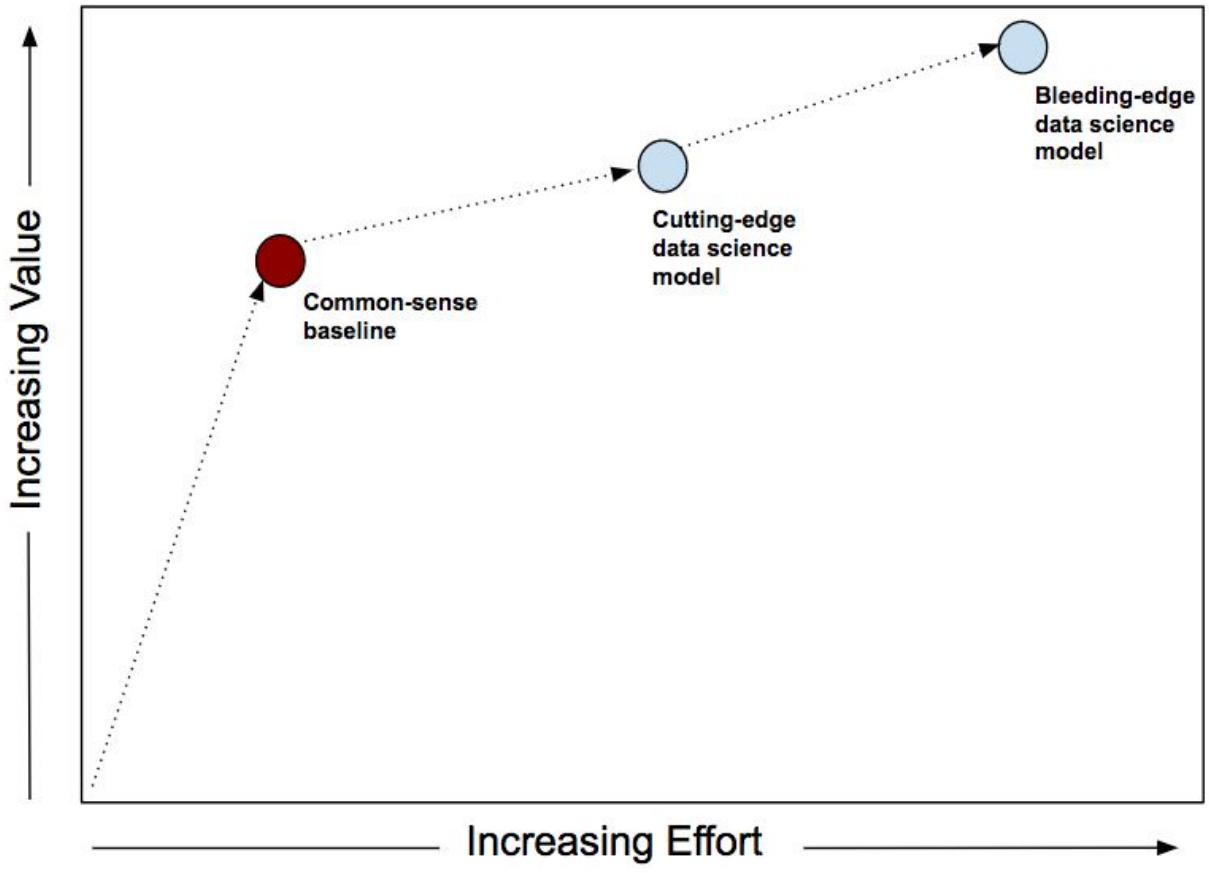
Développer un projet en Machine Learning



Développer un projet en Machine Learning

- Séparer les données en TRAIN/VALIDATION/TEST (i.e 60/20/20)
- Apprendre sur **TRAIN**
- Optimizer les hyperparamètres sur **VALIDATION**
- Observer la performance finale sur **TEST**

Développer un projet en Machine Learning



Projet académiques Vs Industriels

- $\neq$ Développement logiciel
- $\neq$ Infrastructure
- $\neq$ Performances

Développement logiciel

Académique

- Pile de scripts
- Peu de documentation
- Fonctionne le temps de l'expérience
- "Fair use"

Industriel

- Code hiérarchisé et déployable en production
- Documentation
- Code maintenable et robuste
- Galaxies de licences à respecter

Développer un projet en Machine Learning

Infrastructure

Académique	Industriel
▪ Données = un fichier	▪ Données = cloud
▪ Hardware limité	▪ Cloud computing
▪ Performance = Précision	▪ Performance = Plus-value

Développer un projet en Machine Learning

Un problème d'ingénérie avant d'être un problème de machine learning :

Données et prétraitements de qualité $>$ algorithme de qualité

Développer un projet en Machine Learning

Une approche en 4 étapes :

- Créer un pipeline robuste de bout en bout (sans ML)
- Intégrer du ML simple
- Ajouter des caractéristiques sensées
- Conserver un pipeline robuste

Développer un projet en Machine Learning

Créer un pipeline robuste de bout en bout (sans ML) :

- Une baseline avec une heuristique
- Mettre en place des statistiques d'évaluation

Développer un projet en Machine Learning

Intégrer du ML simple :

- Obtenir des données
- Définir UNE métrique d'évaluation facile à observer
- Définir des caractéristiques sensées et faciles à obtenir
- Considérer les heuristiques comme des caractéristiques
- Documenter TOUTES les caractéristiques utilisées

Développer un projet en Machine Learning

Intégrer du ML simple :

- Apprendre un modèle tous les n-jours
- Évaluer la dégradation des performances en fonction de l'âge du modèle
- Vérifier les performances en test avant de déployer en production
- Modèle appris sur des données jusqu'au jour N, tester sur les données après le jour N
- Mesurer la différence entre performance en apprentissage et test
- Plateau de performance $\Rightarrow$ trouver des nouvelles caractéristiques/augmenter la puissance du modèle
- Supprimer des caractéristiques pas déterminantes

Ajouter des caractérstiques sensées :

- Beaucoup de caractéristiques simples $>$ peu de caractéristiques complexes
- Des caractéristiques répandues plutôt que rares
- Regarder les erreurs pour imaginer les caractéristiques qui aideraient
- Communiquer avec les experts métiers

Des questions à garder en tête :

- Ajouter des statistiques d'évaluation ?
- Revoir/Complexifier la métrique d'évaluation ?
- Les données sont-elle "stables" ?