



Gestion de stock intelligente

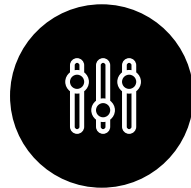
 10h11

William Brojanigo
Chief Data Scientist





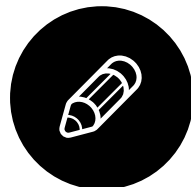
Société de conseil dans l'optimisation et l'exploitation de la data.



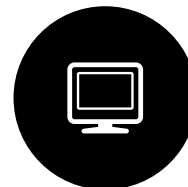
Gestion de projet



Data Science



Design



Développement
web



DEMARCHE DE TRAVAIL : DESIGN THINKING

DECOUVERTE

Découvrir l'environnement d'étude, comprendre les enjeux du projet et entrer dans une **phase de dialogue** avec les différentes équipes.

DEFINITION

Définir les **problématiques** soulevées par les objectifs du projet comme l'existant, et **décrire les scénarios potentiels** permettant d'y répondre.

CREATION

Choisir le scénario le plus pertinent et prioritaire, produire **la brique technologique correspondante** et le document de **spécifications fonctionnelles** associé.

LIVRAISON

Finaliser la conception par **l'implémentation** de la brique en production.

Cette méthode en 4 étapes provient de la démarche *Double Diamond* du *Design Council*. Adaptée par nos équipes aux problématiques de la data, elle permet de valider chaque étape de rendu et de co-construire avec notre équipe chaque livrable. Après un premier sprint de production d'une brique, nous reprenons la même démarche afin de construire la seconde. Allez plus loin : <https://www.designcouncil.org.uk/news-opinion/design-process-what-double-diamond>



10h11 dispose d'un **programme de R&D labellisé par le Ministère de la Recherche** sur les 4 axes suivants :

L'HARMONISATION DES BASES DE DONNEES

Faire parler entre elles des bases de données qui n'ont pas la même structure ou le même format d'origine : concevoir des systèmes capables de **comprendre des flux de données multiples et hétérogènes**.

L'INTELLIGENCE ARTIFICIELLE

Concevoir des **algorithmes et modèles statistiques** qui répondent à différentes problématiques.

LA MISE A JOUR ET LE MAINTIEN DES TECHNOLOGIES

Outils capables d'**évoluer dans le temps**, de s'adapter facilement aux mises à jour, aux changements de formats, aux standards qu'imposent Internet aujourd'hui.

L'ERGONOMIE DES TABLEAUX DE BORD

Prendre en compte **les mécanismes de la pensée humaine** : produire des outils qui soient facilement pris en main par le futur utilisateur entraînant une adhésion et une fidélisation plus fortes.



Exemples de briques technologiques issues de notre R&D à la disposition de nos clients :

- Bering : *scoring de crédit*
- Bil : *indice cartographié*
- Flare : *reconnaissance de tonalité sémantique*
- Karibu : *moteur de profilage*
- Master Draft : *sélection optimisée des composantes*
- Planotec : *planification multi-user*
- Vasco : *trading bot*
- Vigio : *dashboard de pilotage*
- ...

SAY HI TO NEMO

Le premier Chatbot prédictif
lié à la prise de commande

ME CONNECTER AVEC

EMAIL

FACEBOOK



BeMyApp France @BeMyAppFR
Quoi de mieux que de commencer la semaine #MondayMotivator, en félicitant nos deux gagnants de #SmartWarehouse !
E... <https://twitter.com/i/web/status/1061936299521777664>
Twitter | 12 nov. 2018 (160 kB)

BeMyApp France @BeMyAppFR
#frictionlessupplies deux équipes coup de cœur #walteretcarriot @10h11 rendez-vous en janvier avec @Lyrecorecruite encore bravo 🎉 à tous
<https://t.co/6No50ufhnT>
Twitter | 18 déc. 2018 (221 kB)





Enjeux de nos clients :

- optimiser la gestion du stock ;
- éviter les pénuries de fournitures ;
- automatiser les commandes.

Besoin général ressenti :

anticiper l'activité afin d'optimiser les ressources humaines et matérielles.





Focus Nemo : le moteur prédictif

But :

Prédire d'une semaine à l'autre (fréquence adaptable selon le client) les quantités commandées de chaque référence produit pour chacun des clients.

Data :

- historique de commandes
- prévisionnels de commandes
- caractéristiques physiques des produits
- open data (exemples : historique météorologique, données boursières, caractéristiques du calendrier)

Comment :

Trois méthodes testées : ARIMA, MLP et XGBoost.



1 – ARIMA

Limitations principales identifiées :

- données manquantes ou corrompues sont généralement compliquées à gérer ;
- assomption de relation linéaire ;
- dépendance temporelle fixe ;
- données univariées.

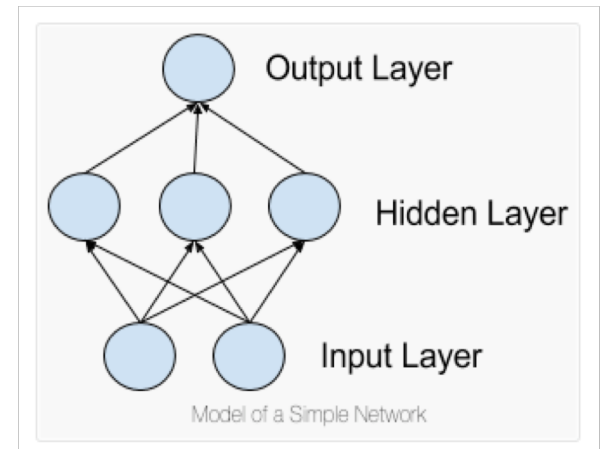


2 – MLPs : MULTILAYER PERCEPTRONS

- typologie classique de réseaux de neurones ;
- recommandé pour image data, text data, time series data ;
- testé dans le même terrain d'application : Ferreira, R. P., Martiniano, A., Ferreira, A., Ferreira, A., & Sassi, R. J. (2016). *Study on daily demand forecasting orders using artificial neural network*. IEEE Latin America Transactions, 14(3), 1519-1525.

Avantages principaux :

- robustes aux bruits et aux données manquantes ;
- rapide apprentissage des relations non-linéaires ;
- *multiple time steps* et *multivariate forecasting* ;
- pas d'assomption sur la stationnarité de la série temporelle (pas de traitements nécessaires).



Pour plus de détails :

- <https://machinelearningmastery.com/time-series-prediction-with-deep-learning-in-python-with-keras/>



3 – XGBoost (Extreme Gradient Boosting)

Bibliothèque logicielle open source créée par Tianqi Chen (*PhD Student, University of Washington*) permettant de mettre en œuvre des méthodes de *Gradient boosting* afin de résoudre des problèmes de ML supervisé.

Avantages principaux :

- *parallel computing* : par défaut, utilise tous les *cores* de la machine ;
- *regularization* : permet la régularisation (évite des problèmes de *overfitting*) ;
- *enabled cross validation* : fonction de CV déjà implementée dans l'algorithme ;
- *missing values* : traite automatiquement les données manquantes ;
- *flexibility* : supporte des fonctions objectifs définies par l'utilisateur ;
- *tree pruning* : contrairement aux GBMs classiques, où le pruning s'arrete dès qu'il y a une perte d'information, XGBoost fait croître l'arbre jusqu'à la profondeur max et l'elague jusqu'à que la perte soit au-dessous d'un seuil défini ;
- *availability* : actuellement disponible pour R, Python, Java, Julia, et Scala.

Pour plus de détails :

- <https://www.hackerearth.com/fr/practice/machine-learning/machine-learning-algorithms/beginners-tutorial-on-xgboost-parameter-tuning-r/tutorial/>



Résultats :

- métrique d'évaluation utilisée : **RMSE** ;
- comparé au ARIMA, MLP est beaucoup plus lent d'un point de vue computationnel, mais il permet d'obtenir de meilleures prédictions ;
- l'efficacité du MLP dépend d'un grand nombre de paramètres à optimiser (la taille de la fenêtre de prédiction, le nombre de couches et de neurones, ...) lesquels augmenteront notablement les temps computationnels ;
- Aux vues des RMSEs obtenus, il en résulte que le XGBoost est le modèle le plus performant (sur les données utilisées).



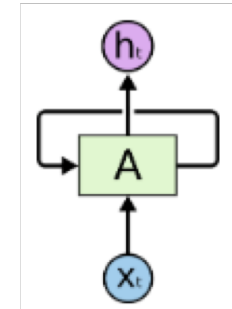
Freins à lever :

- Le RMSE nous permet parfaitement de comparer la capacité prédictive de plusieurs modèles, mais nous ne donne pas de % d'erreur pour définir plus globalement la fiabilité du modèle choisi.
- Limite dans la vitesse computationnelle due à l'optimisation des hyperparamètres et la sélection des variables.
- Les produits qui ont été très peu commandés ne garantissent pas des prédictions fiables.
- Tests inachevés avec LSTM (voir slide suivante).



LSTM : LONG SHORT-TERM MEMORY

- typologie de *Recurrent Neural Networks* (RNNs) ;
- *long-term dependency* : par défaut, leur comportement est de se souvenir des informations pendant de longues périodes ;
- surmonte le problème de stabilité des gradients pendant la phase d'apprentissage des RNNs classiques (*vanishing gradient problem*).



Problème rencontré :

En supposant que nous connaissons $X_{(t)}$ (prévisionnel), il y a des limitations dans la modélisation de $X_{(t-n)}, \dots, X_{(t)}$ et $Y_{(t-n)}, \dots, Y_{(t-1)}$ afin de prédire $Y_{(t)}$:

- les variables n'ont pas la même dimension temporelle ;
- solution possible : assigner -1 à $Y_{(t)}$ et appliquer *masking function* (inachevée sur Python).

Pour plus de détails :

- <http://colah.github.io/posts/2015-08-Understanding-LSTMs/>
- <https://machinelearningmastery.com/when-to-use-mlp-cnn-and-rnn-neural-networks/>
- <https://machinelearningmastery.com/promise-recurrent-neural-networks-time-series-forecasting/>
- <https://machinelearningmastery.com/multivariate-time-series-forecasting-lstms-keras/>



Merci pour votre écoute !



Des questions ?