1. Introduction

BVA souhaite dans le cadre de son offre DmrP® (Data marketing research Platform) proposer à ses clients une solution de machine learning qui s'adapte au volume de données en accès distant. Cette solution ne doit nécessiter aucune programmation en n'utilisant que des interfaces WYSIWYG[2] accessibles à distance en mode SAS avec pour cela toutes les ressources du cloud. Elle doit pouvoir également pour les utilisateurs les plus chevronnés accepter la programmation (R, Python ou Spark[3]) mais uniquement si ces utilisateurs le souhaitent, que ce soit un input d'un modèle pour le finaliser ou en output : pour pouvoir intégrer la syntaxe du modèle dans une application tierce.

Dans ce cadre, des outils ont été retenus : Azure de Microsoft pour le stockage de la solution et Modeler d'IBM pour la partie Machine Learning (Lambert et al, 2015) sur la base de deux architectures standards d'éditeurs majeurs disponibles sur le marché selon le volume de données gérées :

- Sur les volumes moyens : en-dessous de 3 terras avec une donnée mise à jour de manière asynchrone : SSAS de SQL Server
- Sur des volumes infinis et/ou avec une mise à jour des données en temps réel, une architecture Hadoop/Spark (Big Data)

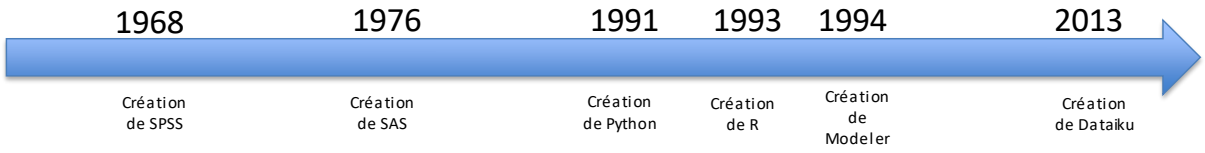
Les deux architectures devant fonctionner en toute transparence pour l'utilisateur marketing qui n'utilise pour cela que l'interface objet. L'outil auto-adaptant l'espace de calcul et le langage.

1.1. Méthodologie :

Le but de ce travail de recherche est de construire réellement un véritable démonstrateur de l'ensemble du processus et de le rendre opérationnel pour le proposer à des clients. Notre objectif n'est pas directement de montrer un cas de machine learning lui-même. Nos publications précédentes le font (Vallaud 2005, 2012). L'objectif est bien d'avoir une solution qui fonctionne et dont on puisse mesurer les performances.

1.2. La reprise en main des utilisateurs :

Toutes les données ne sont pas « big » mais quand elles le sont, si on veut les rendre accessibles à l'analyse par le plus grand nombre il faut des outils interfacés utilisateurs (WYSIWYG) capables de créer les « conditions » nécessaires à la manipulation et à l'analyse des données. Tous les utilisateurs ne seront pas tous des codeurs (Haris et al, 2013).



Le marché a pendant très longtemps créé des outils avec à la fois du code pour commencer - SAS (www.SAS.com) - et puis se sont rajoutés ensuite, avec la popularité de Windows, des interfaces « objet » permettant de manipuler les données et les modèles : SAS Guide, EM. SPSS racheté par IBM (http://www-03.ibm.com/software/products/fr/spss-modeler), un autre ancêtre de ces outils, permettant lui dans l'interface de compléter par du code, que ce soit dans sa solution d'analyse statistique plus classique ou bien de machine learning.

[2] https://fr.wikipedia.org/wiki/What_you_see_is_what_you_get
[3] R, Python et Spark sont des langages du big data

Il y a donc fort à penser que les outils de nouvelle génération aillent dans ce sens, avec Azure Machine Learning (https://azure.microsoft.com/fr-fr/services/machine-learning/) et Watson (http://www-05.ibm.com/fr/watson/). Si tant est que l'on arrive à bien définir son périmètre), Dataiku (http://www.dataiku.com/) dans une moindre mesure.

Le puriste prétend que les outils qui ne passent pas « que » par du code comme R et Python ne permettent pas de bien rentrer dans le modèle statistique lui-même. C'est un peu faire fi du passé car pendant des années, SAS et SPSS ont été les outils des « plus » grands spécialistes de la statistique du marché (Vallaud, 2015).

Pour démocratiser la data science sur des données volumineuses (Provost et Fawcett, 2013) il faut donc des solutions nouvelles qui vont créer à la fois les clusters Hadoop sur un serveur et dans le cloud, générer du code Spark pour l'utilisateur de manière transparente et lui permettre de faire du « machine learning » via son interface utilisateur. Tout cela étant généré pour lui.

Dans ces nouvelles solutions on peut imaginer deux extrêmes (Foreman, 2014) :

> 1) Une API packagée qui fait tout (par exemple : un modèle d'attrition) auquel l'utilisateur connecte ses données et en appuyant sur un bouton fait tourner le modèle et obtient uniquement les résultats dans un tableau de bord. Les individus en base étant alors scorés. C'est la solution la plus extrême que nous ne retenons pas dans notre présente analyse mais qui apparaît de plus en plus sur le marché.
> 2) L'autre solution que nous privilégions est une interface utilisateur qui laisse la main sur la modélisation à l'utilisateur mais qui fait tout le back office : création des clusters, ajustement de la mémoire, transcodification du modèle en Spark et restitution des résultats. Evidemment « révolution numérique » oblige, nous voulons cela sur un cloud sécurisé avec le meilleur rapport coût/simplicité/performance et une interface en ligne accessible de n'importe quel point, du bureau ou en Home Office.

Vers une typologie des outils :

Les outil d'advanced analytics selon le Gartner 2016[4]

[4] http://www.kdnuggets.com

Vers un outil distant de machine learning capable de gérer les données quelle que soit leur taille :

L'interface retenue est Modeler d'IBM : puissante solution de machine learning/data mining (Lemberger et al, 2015, Hyeans, 2016) relativement connue sur le marché.

Le cloud est celui d'Azure : un des plus simples à mettre en place et à gérer pour un utilisateur non informaticien. Notez que nous aurions pu prendre Azure ML (Mund, 2015) comme outil de machine learning pour rester cohérent avec notre choix Microsoft, mais la récence de l'outil, en constante évolution notamment depuis le rachat de Révolution (R sous Windows) ne le rend pas, selon nous, aussi puissant en WYSIWIG que Modeler qui a plus de 20 ans d'existence.

Le défi étant de faire travailler ensemble deux des meilleures technologies du marché d'éditeurs pour le moins concurrents. Cela peut aussi être un enjeu. Dans la suite du document, nous décrivons comment réaliser concrètement un tel outil.

1.3. Objectif du présent document :

Le but de ce document est de décrire précisément chacune des taches à effectuer pour permettre au lecteur de bien comprendre le travail réalisé.

2. Les outils SPSS Modeler + SSAS

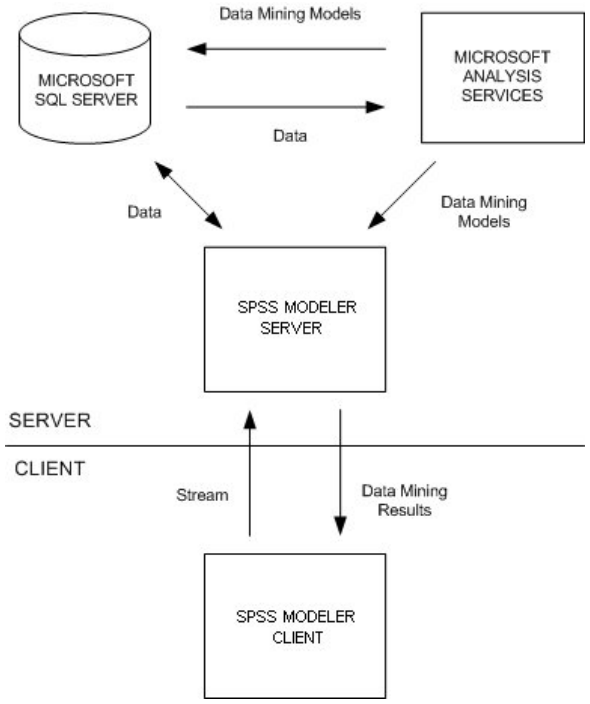
2.1. Architecture de SPSS Modeler + SSAS

SPSS Modeler est capable de déporter le calcul de ses modèles vers SQL Server Analysis Services. De cette manière, l'utilisateur peut bénéficier de la puissance de calcul de SQL Server en réduisant à un minimum les déplacements de la donnée. Cela permet ainsi d'optimiser des traitements de gros volumes, les calculs de machine learning se faisant dans la base de données elle-même (in data base).

La documentation officielle se trouve dans

http://www.ibm.com/support/knowledgecenter/en/SS3RA7_18.0.0/modeler_mainhelp_client_ddita/clementine/db_modeling_buildnode.html

Le diagramme suivant illustre l'architecture technique :



A noter que :

- SPSS Modeler Server est optionnel et donc utilisé chez BVA seulement sur des volumes très importants de données ;
- Le serveur SQL Server et SSAS doivent être sur la même machine.

2.2. Niveaux de performance

Le gain de performance avec SSAS provient du fait que les données ne sortent pas du serveur et que la taille de ce dernier est potentiellement plus importante qu'un poste de travail. On préconise l'utilisation de cette architecture pour des bases de données contenant entre 1 million et 100 millions de lignes par table.

2.3. Prérequis

- SPSS Modeler (client)
    - SPSS Modeler premium 17 ou +
    - Microsoft SQL Server Analysis Services 10.0 OLE DB (ou supérieur) (https://msdn.microsoft.com/en-us/library/dn141152.aspx)
    - Microsoft SQL Server Native Client (https://msdn.microsoft.com/library/mt654048.aspx)
    - Microsoft ADOMD.NET (https://msdn.microsoft.com/en-us/library/mt592624.aspx)
    - Le client doit pouvoir accéder au serveur SQL et à SSAS avec une authentification windows.
        - Il doit donc avoir une relation de confiance entre les domaines du client et le serveur.
- SQL Server Database 2012 ou +
- SQL Server Analysis Services (entreprise idéalement) 2012 ou +
    - Le service doit résider dans la même machine que la base SQL

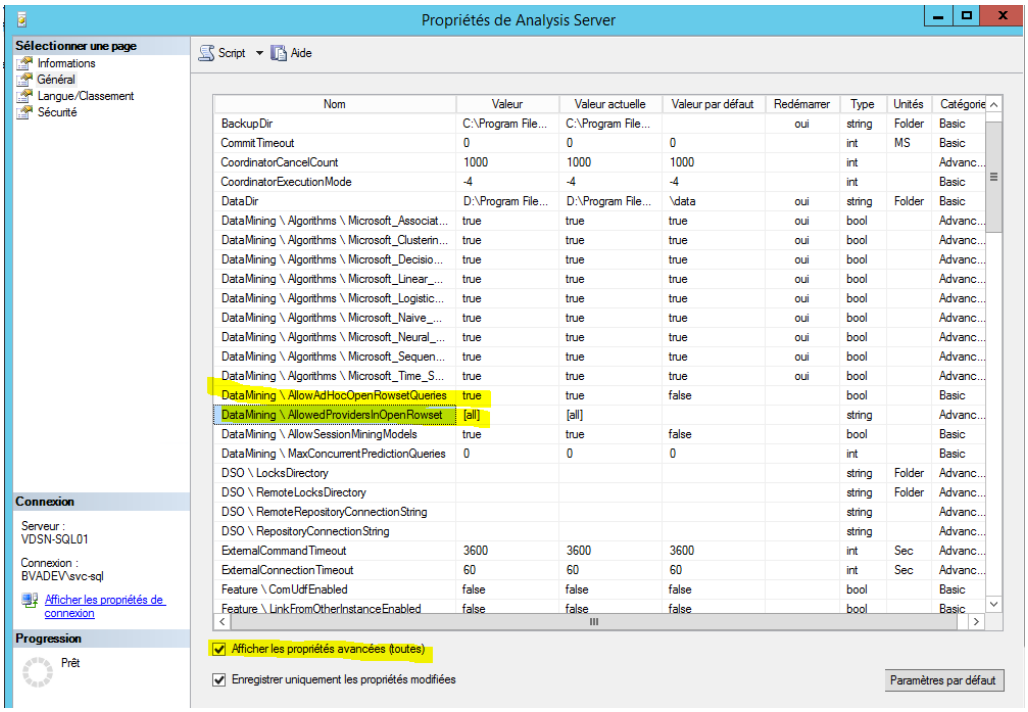
2.4. Configuration Connectivité

2.4.1. Pour SQL Server

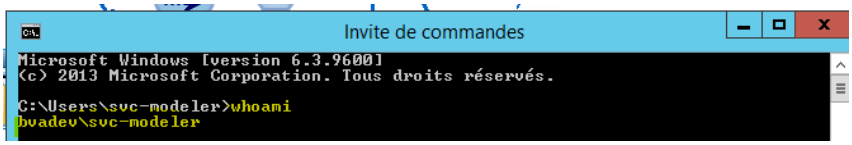
- Créer la clé de registre suivante :
  - HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSSQLServer\Providers\MSOLAP
- Rajouter le DWORD suivante :
  - AllowInProcess 1
- Redemarrer le serveur SQL.

2.4.2. Pour SQL Server Analysis Services

- Configurer les propriétés affichées dans l'image ci-dessous



- S'assurer que l'utilisateur qui exécute Modeler est administrateur du service SSAS (voir image ci-dessous)
  - Executer Whoami en local



  - Rajouter le même utilisateur parmi les administrateurs du serveur (une administration au niveau de la base SQL est également possible)