Fondation ubs: Université Bretagne Sud

CHAIRE D-CC "DÉCISIONNEL-CONNAISSANCE CLIENT" - IUT de Vannes - 8 rue Montaigne - 56017 Vannes Cx / 02 97 62 64 64 / chaire.d-cc@univ-ubs.fr

Rapport d'utilisation de Model Producer, le logiciel de Predicsis pour la production rapide de scores comportementaux

Matthieu GOUSSEFF

January 30, 2017

Contents

Abstract

Le logiciel Model Producer propose un compromis entre la rapidité de traitement, la simplification du data management et la qualité prédictive de scores de comportement. La scalabilité permise par le moyennage optimisé de prédicteurs bayésiens naïfs produit des scores de bonne qualité dans des délais raisonnables. La construction d'agrégats à partir de tables contenant plusieurs mesures d'une variable par individu est une fonctionnalité susceptible d'améliorer la productivité des data-miners lors de la production de scores.

1 Contexte

1.1 Productions de scores comportementaux

La relation entre l'entreprise et le client repose en grande partie sur la capacité de proposer le bon produit au bon client au bon moment. La production de scores d'appétences consiste à estimer, à partir des variables qu'on peut mesurer sur un individu, la probabilité qu'il a de présenter un comportement d'intérêt. L'exemple typique est de mesurer quel appétence un client peut avoir à un produit ou à un service proposé par l'entreprise. On peut alors cibler les individus les plus appétents à l'offre qu'on souhaite promouvoir.

Il existe un grand nombre d'approches prédictives pour élaborer de tels scores, régressions logistiques, arbres de décisions, forêts aléatoires... Le Model Producer (MP) est l'implémentation commerciale d'une méthode originale de combinaisons de classifieurs bayésiens naïfs.

1.2 Predicsis

1.2.1 L'entreprise

Predicsis a été fondée en 2013 afin de valoriser la technologie Khiops développée au sein d'Orange Labs sur une durée de 15 ans. Soutenue par le fonds d'investissement Innovacom VC, Predicsis a vu ses revenus tripler entre 2014 et 2015. Elle compte aujourd'hui 25 salariés et revendique une approche disruptive dans la production de scores comportementaux.

1.2.2 Model Producer

Les classifieurs bayésiens naïfs sont une méthode classique de classification supervisée. L'approche proposée par Predicsis est de combiner des classifieurs bayésiens naïfs en optimisant un critère original, le taux de compression. Les atouts revendiqués par Predicsis sont :

- Une simplification du data management par la création d'indicateurs (agrégats)
- Une scalabilité permettant de traiter à la fois un grand nombre d'individus et un grand nombre de variables prédictrices

- Un score de très bonne qualité au vu des objectifs métiers visés

Il est à noter que l'algorithme d'optimisation utilisé dans la discrétisation des variables et l'agrégation de modèles fait l'objet d'un brevet.

2 Méthodologie sous-jacente

Le Model Producer de Predicsis repose sur la combinaison de classifieurs bayésiens naïfs. Cette combinaison utilise un critère original qui est appelée compression. Sans entrer dans le détail mathématique de ces méthodes (disponibles dans les publications [Boullé, 2004] et [Boullé, 2006] par exemple) les paragraphes suivants permettent d'avoir une idée générale du fonctionnement de l'outil.

2.1 Combinaison de classifieurs bayésiens naïfs selectifs

2.1.1 Classifieurs bayésiens naïfs

Les prédicteurs bayésiens naïfs sont une classe de modèles de classification supervisée.

On parle d'apprentissage supervisé quand on dispose d'une base d'apprentissage. Dans notre cas, la base d'apprentissage est un ensemble de clients pour lesquels on sait s'ils ont acheté ou non le produit, et pour lesquels on a mesuré en plus un certain nombre de variables prédictrices (âge, revenus, autres produits achetés, etc.).

Le terme bayésien vient de la façon d'estimer la probabilité d'acheter en utilisant les propriétés des probabilités conditionnelles.

On parle de prédicteur bayésien naïf quand on fait l'hypothèse que les variables prédictrices sont indépendantes conditionnellement à la variable à prédire. L'avantage majeur de cette hypothèse d'indépendance est qu'elle permet de traiter les variables prédictrices une à une, et donc de traiter des jeux de données contenant un grand nombre de variables, même pour un grand nombre d'individus.

Hélas, dans la réalité, cette hypothèse d'indépendance est souvent contredite par les données réelles, ce qui peut dégrader la valeur prédictive du modèle construit.

2.1.2 Classifieurs bayésiens naïfs sélectifs

Pour éviter la dégradation des performances dues à la dépendance entre les variables prédictrices, il est possible de sélectionner des sous-ensembles de variables compatibles avec cette hypothèse d'indépendance conditionnelle. Cette méthode permet la sélection ou la non-sélection d'une variable, pas la pondération de multiples prédictrices et est réputée sensible au sur-ajustement : si on change d'échantillon d'apprentissage, on est susceptible de sélectionner un autre sous-ensemble de prédictrices.

Plusieurs critères sont utilisés pour la sélection de modèles. La maximisation de la probabilité a posteriori (MAP), la précision (accuracy), l'aire sous la courbe ROC (AUC)...

2.1.3 Combinaison de modèles

La combinaison (on parle aussi de moyennage, de l'anglais averaging) de modèles permet, au lieu de sélectionner un modèle unique, d'effectuer une moyenne pondérée de plusieurs modèles. Dans le cas des classifieurs bayésiens naïfs, la question du moyennage des modèles revient à une pondération des variables retenues. On recherche des pondérations qui optimisent des critères de qualité de modèles, par exemple ceux évoqués dans le paragraphe précédent.

Les méthodes habituelles de moyennage de classifieurs bayésiens naïfs sélectifs peuvent donner des résultats assez proches d'un classifieur bayésien naïf sélectif ordinaire.

2.2 Combinaisons de classifieurs bayésiens par optimisation du taux de compression

La spécificité du Model Producer de Predicsis est qu'il utilise un critère original pour la combinaison de modèles, **le taux de compression.** Les classifieurs présentés jusqu'ici s'inscrivent dans la théorie de statistique bayésienne qui s'appuie sur certaines hypothèses de distribution des données.

2.2.1 Compression de code et théorie de l'information

De manière indépendante, dans les années 70-80, les chercheurs en informatique définissent des méthodes pour recoder une information de la manière la plus synthétique possible. Il s'agit de passer, à l'aide d'un recodage judicieusement défini, d'une séquence (par exemple de 0 et de 1) à une autre, de taille plus petite, mais contenant la même information.

Le rapport entre la longueur du code initial décrivant la séquence et du code final décrivant la même séquence est appelé le taux de compression.

Cette notion, au départ définie pour des reconstruction parfaites, a été étendue à la reconstruction de séquences les plus proches possibles de l'originale, de la même façon qu'un modèle statistique cherche à apporter la meilleure approximation de la réalité. On parle alors de complexité stochastique de modèles.

À nouveau, il est difficile d'affiner la notion sans entrer dans des formalisations mathématiques ardues, mais l'idée générale est de faire moins d'hypothèses sur les processus susceptibles d'avoir généré les données et de définir une façon d'agréger les modèles la plus robuste face à la situation la plus défavorable.

2.2.2 En pratique

En pratique, la combinaison de bayésiens naïfs basée sur l'optimisation du taux de compression aboutit à prendre en compte l'influence d'un plus grand nombre de variables, en réalisant un compromis entre la nécessité de sélectionner

des variables et la robustesse du modèle. D'autre part, chacun des modèles qu'on combine est toujours un classifieur bayésien naïf sélectif, donc l'hypothèse d'indépendance des variables prédictrices permet de traiter les variables séquentiellement. Ce qui permet le passage à l'échelle sans rencontrer de problèmes en termes de mémoire nécessaire et de conserver des temps de calcul raisonnables.

[Boullé, 2004] montre avec des cas pratiques le comportement de ces combinaisons de modèles.

Ainsi, d'autres méthodes concurrentes peuvent apporter des résultats meilleurs, mais au prix d'une complexité et d'un temps de calcul pas forcément compatibles avec les exigences d'une activité économique.

2.3 Discrétisation et construction d'agrégats

2.3.1 Discrétisation des variables

Les classifieurs bayésiens naïfs décrits plus haut reposent sur l'utilisation de variables prédictrices catégorielles, mais l'outil Model Producer accepte également des variables quantitatives comme prédictrices potentielles. Il les discrétise de manière quasi optimale (voir [Boullé, 2004]).

En évitant à nouveau les formalisations mathématiques, on peut retenir que l'algorithme de discrétisation exige une étape de tri des données selon la valeur de la variable à discrétiser, puis des algorithmes de recherche d'intervalles, avec des stratégies de regroupement/éclatement d'intervalles qui permettent un compromis entre le temps de calcul et l'optimalité de la solution retenue.

Cette phase de discrétisation est préliminaire à la constructions des modèles de combinaisons de classifieurs bayésiens, et c'est d'expérience la phase la plus longue lors de l'utilisation du Model Producer. Il faut noter que cette phase n'est nécessaire qu'une fois et qu'elle n'est pas effectuée à chaque raffinage des modèles (voir section test pratique).

2.3.2 Construction d'agrégats

Il existe des cas ou certaines variables sont mesurées plusieurs fois pour un individu. Dans le cas de multiples commandes passées par un client unique, une variable de montant est mesurée pour chaque commande. Ces données sont en général stockées dans des tables périphériques.

Afin de réduire le temps de data management préliminaire à la construction de modèles, l'outil propose de construire un certains nombre de variables récapitulatives qui seront intégrées à l'ensemble des variables prédictrices candidates (pour la procédure de construction de ces variables, voir [Boullé, 2014]).

Pour une table périphérique, les opérateurs d'agrégation intégrés à la version actuelle de Model Producer sont les suivants (les exemples sont entre parenthèses).

- Nombre de mesures (nombre de commandes d'un client)
- Nombre de catégories (le nombre de produits différents achetés)

- Mode (le produit acheté le plus souvent par le client)
- Moyenne (le montant moyen de commandes pour ce client)
- Minimum (le coût de l'article le moins cher acheté par le client)
- Maximum (le montant de la plus grosse commande du client)
- Somme (la somme des montants de toutes les commandes d'un client)

Ces variables sont ensuite traitées comme les autres prédictrices du modèle.

On notera qu'on peut paramétrer l'utilisation de ces tables périphériques ainsi que le nombre de variables reconstruites qu'on accepte d'intégrer au modèle final.

Conclusion sur la méthodologie

L'outil Model Producer de Predicsis repose sur une **combinaison de classifieurs bayésiens naïfs** (bagging) qui permet de trouver un **compromis entre la scalabilité (temps de calcul, mémoire nécessaire), la facilité de construction d'un modèle et la qualité de prédiction**. Les mesures répétées sur un individu peuvent être prises en compte à partir de tables périphériques sans data management supplémentaire.

3 Test Utilisateur

3.1 Installation

Le serveur utilisé pour tester Model Producer est équipé d'un processeur unique à 10 cœurs cadencé à 3.1 GHz, il est équipé de 256 Go de RAM. Les données sont stockées sur des disques de 1,2 To à 10 000 tours minutes. Le système d'exploitation du serveur est une Debian 8.

Le Model Producer est hébergé à l'intérieur d'une machine virtuelle qui présente un serveur accessible par le biais d'un navigateur (par défaut, par le port 8080 du localhost).

L'installation nécessite vagrant et virtual box.

Predicsis fournit un fichier Vagrantfile qui permet de configurer la machine virtuelle et un fichier provision.sh qui permet le téléchargement du contenu de la machine virtuelle (le Model Producer proprement dit). La licence est gérée par le biais d'un système de token.

On peut se demander si la solution technique par vagrant, c'est à dire par virtualisation est vraiment pertinente par rapport à une solution de type docker. Si la sécurité est renforcée, la consommation en ressource est plus importante. Cependant, sur notre serveur, cela 'na pas posé de problèmes.

L'accès se fait par un navigateur (localhost sur le port 8080).

Avantage

- L'installation est aisée si l'on dispose des droits nécessaires et d'une connexion suffisante.
- Le lancement et la fermeture du Model Producer se font par une simple commande vagrant. Un système de clé permet le téléchargement sécurisé du contenu de la machine virtuelle.
- Le Model Producer (MP) est acccessible depuis tous les postes qui ont accès au serveur qui l'héberge, par le biais d'un simple navigateur web.
- Si le port 8080 est indisponible, MP propose d'autre ports qui sont affichés dans le terminal à partir duquel est fait le lancement.

Inconvénients

- Des problèmes de stabilité ont été rencontrés. Nécessité de fermer et relancer la machine virtuelle.
- Il faut être vigilant sur les paramètres systèmes relatifs à virtual box et s'assurer que le repertoire par défaut contient l'espace suffisant à la duplication des données.
- La mise en place de la machine virtuelle demande un certain temps. Ainsi, il est parfois nécessaire de rafraîchir plusieurs fois avant d'accéder au MP.
- Pas de message d'erreur en cas de fermeture inopinée du Model Producer, nécessité de consulter les logs (via l'interface, donc peu simple si la machine ne se lance pas).

Conclusion sur l'installation L'installation par machine vagrant est en principe simple et sécurisée. L'accès au Model Producer par un navigateur peut être vu comme un avantage. L'interrogation du modèle par une API à l'aide d'un SDK n'a pas pu être testée.

3.2 Préparation des données

Les données doivent être déposées dans un répertoire créé au sein du répertoire mp-data d'où est lancé le Model Producer. Le nom du répertoire sera le nom du projet associé. Une nomenclature est également exigée pour le nommage des variables, bien que les dernières versions aient apporté un peu plus de souplesse.

Le nom du fichier contenant la variable à prédire doit se terminer par la chaîne de caractères CENTRAL. Les tables périphériques doivent reprendre le nom de la table centrale sans le suffixe CENTRAL. MP ne propose pas d'échantillonner pour l'utilisateur un sous-échantillon d'apprentissage et un sous-échantillon de test. La structure de fichier doit donc être dupliquée avec un suffixe TRAIN et un suffixe TEST pour les deux sous-échantillons. Cette

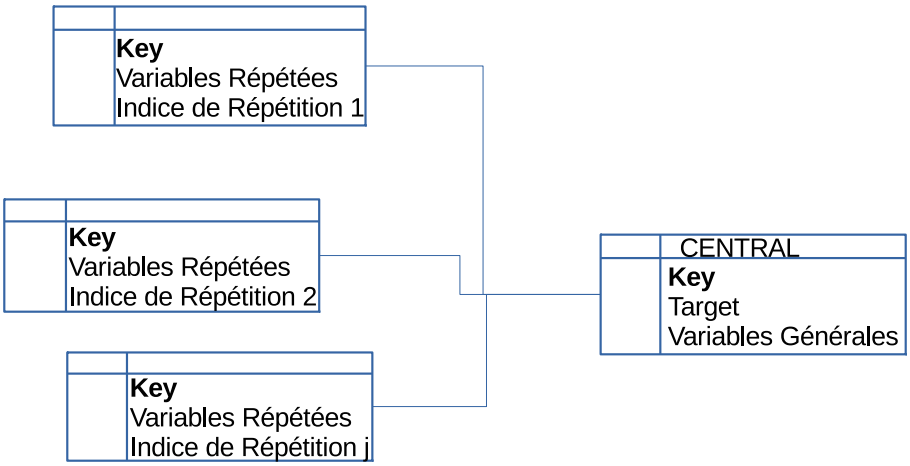

Figure 1: Structure en étoile pour la construction automatique d'agrégats

structuration contraignante peut cependant être aisément automatisée dans la phase d'extraction des données.

A noter que le séparateur de champs est spécifié au niveau de l'outil et non au niveau du projet. Ainsi, si vous avez traité des fichiers csv séparés par un ";" et que les fichiers de votre prochain modèle sont séparés par un "|" il faudra modifier le séparateur par défaut avant de lancer le modèle, sans quoi une erreur sera renvoyée. Enfin, un répertoire tobedeployed doit être créé pour le jeu de données sur lequel l'équation de prédiction sera appliquée.

Avantages

- Une convention de nommage explicite.
- Toutes les données traitées par le MP sont dans le même répertoire.
- Une plus grande souplesse depuis les dernières versions.
- La possibilité d'utiliser des tables périphériques.

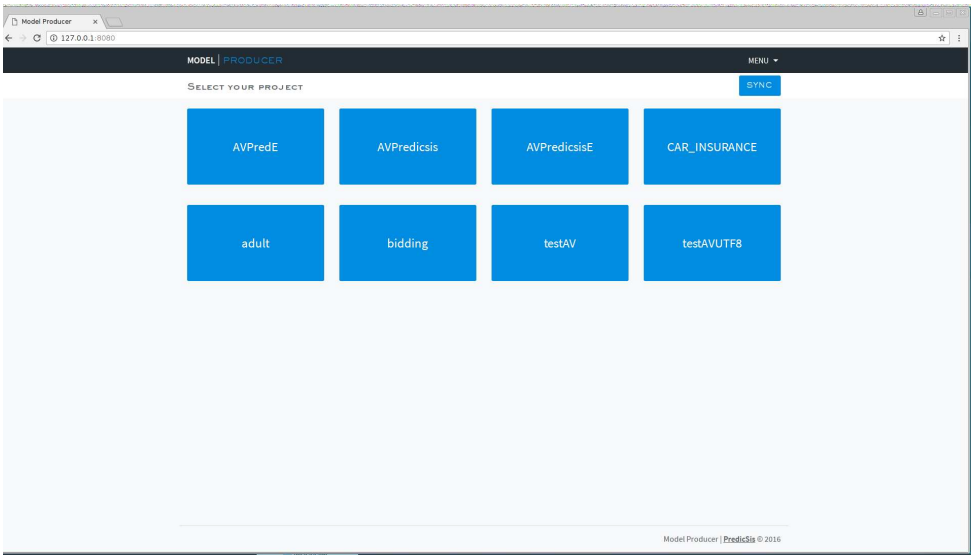

Figure 2: Page d'accueil du Model Producer

Inconvénients

- Pas d'import convivial de données.
- Pas de possibilité de créer les échantillons test et apprentissage dans l'outil.
- Ergonomie du choix du séparateur, des variables d'identification et cible au niveau outil et non au niveau projet.
- Les données doivent obligatoirement être encodées en UTF-8.

3.3 Production d'un score

La page d'accueil de MP présente autant de tuiles qu'il y a de répertoires contenant des données à scorer dans le répertoire mp-data (voir la figure 2, page 10). On accède à la construction de modèles en cliquant sur la tuile correspondant aux données qu'on veut traiter.

3.3.1 Premier score

Lors de la création du premier score, MP effectue un certain de nombre de tâches préliminaires. Ces tâches ne sont effectuées qu'une fois, mais ce sont elles qui demandent le plus de temps de calcul.

- Analyse des noms de fichier
- Recherche du séparateur

- Identification des variables d'identification et à prédire
- Tri et discrétisation des données

La dernière étape prend le plus de temps (algorithme en $O(nlog(n))$), mais elle n'est effectuée qu'une seule fois et n'est plus nécessaire pour les raffinements ultérieurs de modèles.

Enfin, un premier score est produit, qui, par défaut, ne prend pas en compte les tables périphériques. Ce premier modèle intègre la liste des modèles disponibles, qui est incrémentée d'une ligne à chaque raffinement du modèle. Pour chaque modèle il est possible de raffiner le modèle, de produire un rapport, ou de déployer le score sur un autre jeu de données.

3.4 Rapport d'analyse

Le rapport d'analyse d'un score est présenté à l'utilisateur comme une page web interactive, qui peut ensuite être exportée et distribuée au format pdf. On peut voir un exemple de la page principale sur la figure 3, page 12.

L'indicateur synthétique par défaut est l'indicateur de Gini (aire sous la courbe ROC). Il permet de comparer la qualité des différents modèles créés. La robustesse du modèle est estimée par le rapport entre cet indice de Gini mesuré sur l'échantillon test et sur l'échantillon d'apprentissage. Le taux de répondants, c'est à dire le pourcentage d'individus ayant la modalité d'intérêt pour la variable cible est présenté, afin de servir de référence pour l'effet des différentes modalités.

Le gain et et le levier sont affichés par tranche de 5% de l'échantillon. Les tranches pour lesquelles le levier cumulé est supérieur à 2 sont indiquées par un surlignage de couleur. Cette représentation aide à la décision quant à la taille de la cible à retenir, en fonction des objectifs de concentration qu'on cherche à atteindre ou de ressources disponibles.

La liste des prédictrices retenues est affichée. Cliquer sur une variable permet d'accéder à un graphique représentant l'évolution du pourcentage de répondants par modalité, ainsi que la fréquence de chaque modalité.

Le niveau représente la corrélation entre la variable cible et la variable prédictrice.

Le poids indique dans quelle mesure la variable prédictrice améliore la qualité de prédiction du modèle.

La contribution de la variable au modèle est estimée le produit du poids par le niveau divisé par la somme de ces produits pour l'ensemble des prédictrices du modèle. Enfin pour chacune des variables, un graphique montrant l'évolution du taux de répondants selon les modalités de la prédictrice est disponible.

Il est possible de choisir les prédictrices qu'on veut voir apparaître dans le rapport exporté en pdf, et on peut associer à chacune un commentaire.

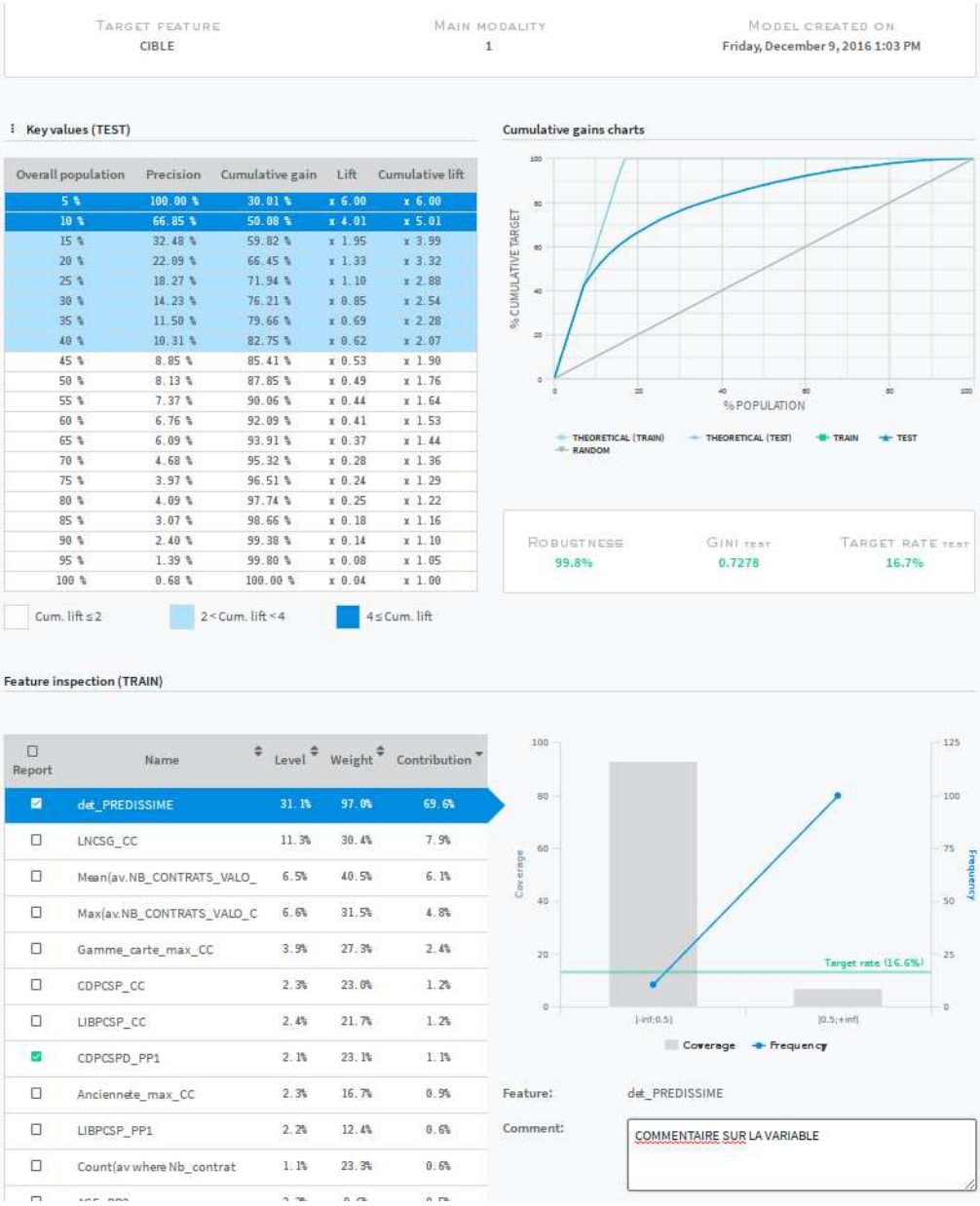

Figure 3: Exemple de rapport produit par Model Producer.

La production de ce rapport est rapide et aisée. Elle est un avantage certain de l'outil et permettra de communiquer avec les experts métiers sur la base de documents intelligibles et appropriables.

3.4.1 Raffinage de modèle

Le premier modèle construit ne prend pas en compte les variables créées à partir des tables périphériques. **La force de l'outil est indéniablement la facilité avec laquelle il est possible de raffiner le modèle initial.** L'utilisateur peut de manière très intuitive forcer la prise en compte ou le rejet de variables prédictrices, fixer un nombre maximal de variables et un nombre maximal de variables issues des tables périphériques.

Ces raffinements sont pris en compte très rapidement pour la construction d'un nouveau modèle, car ils ont lieu après le tri et la discrétisation des variables. Le travail de construction d'un modèle prédictif est donc rapide et intuitif **pour qui connaît bien les variables d'entrée.**

3.5 Un exemple concret : score d'appétence assurance vie

Nous avons testé le Model Producer sur un jeu de données ayant servi à la construction d'un score d'appétence à un produit d'assurance-vie.

3.6 Déroulement de la production du score

Les données ont été mises en forme et exportées dans le répertoire mp-data. Les premières versions du Model Producer ont exhibé des erreurs internes qui ont été corrigées par la suite. **Le support technique a été réactif et efficace dans la correction des ces bugs.**

Le jeu de données comportait 120 000 lignes et 30 variables dans la table centrale. La table périphérique contenait 42 variables mesurées à 4 périodes différentes sur les mêmes 120 000 individus.

Ce jeu de données a ensuite été scindé en un échantillon d'apprentissage et un échantillon de test (tirage aléatoire sans remise). Il a été nécessaire de ré-encoder le jeu de données en UTF-8 pour que le MP accepte de le traiter.

3.6.1 Comparaison score régression logistique

Parallèlement à la construction du score avec le Model Producer de Predicsis, un score a été construit à l'aide d'une régression logistique. La plupart des variables continues avaient été discrétisées au préalable à l'aide de la procédure d'optimal binning de SAS Enterprise Miner. Pour les variables répétées dans le temps, différents indicateurs agrégés ont été construits "à la main".

On a ensuite construit les courbes ROC correspondant aux deux modèles.

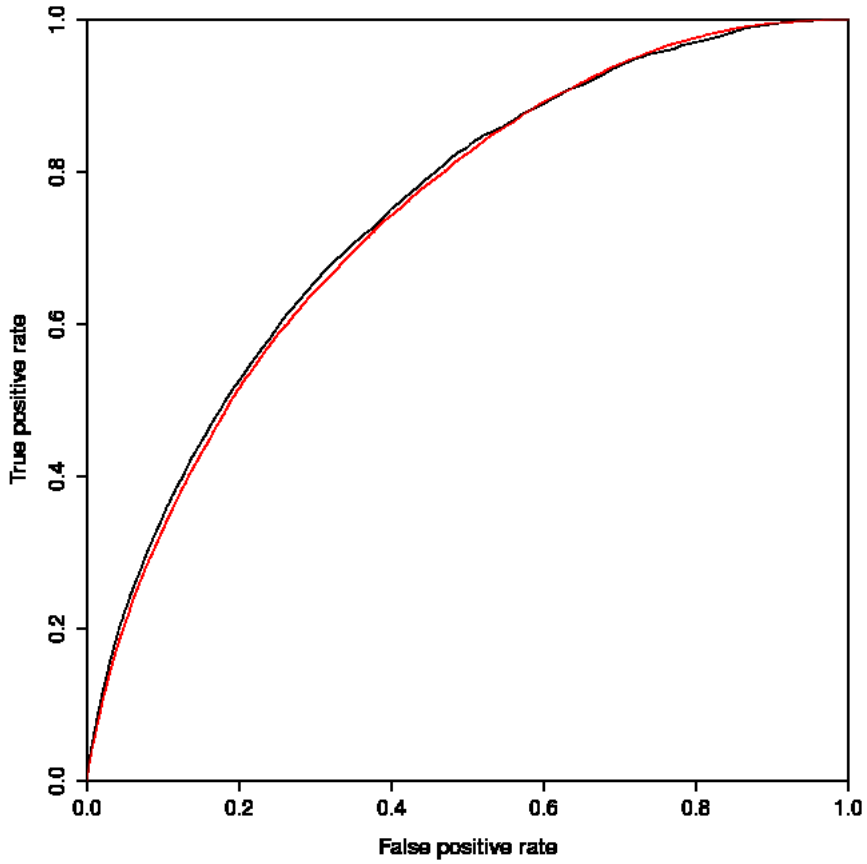

Figure 4: Courbes Roc des scores produits par régression logistique et par Model Producer.

On constate avec une certaine surprise que **sur notre jeu de données, la régression logistique et le score produit par le Model Producer présentent des performances tout à fait similaires,** avec des courbes ROC quasiment superposables.

Mais il est nécessaire de prendre en compte le temps nécessaire à la production de ces deux modèles. Dans le cas de la régression logistique, la durée de construction du modèle de score peut être évaluée à environ une douzaine de jours homme. **Dans le cas de Model Producer, la mise en forme des données et la production du score final ont pu être effectuée dans la même journée** (après la prise en main de l'outil et la préparation sommaire des données).

Le Model Producer est donc un outil performant pour améliorer la productivité d'une équipe de data-miners, même s'il ne garantit pas l'optimalité d'un modèle selon le critère de l'aire sous la courbe ROC. Dans le cadre de la production fréquente de scores servant à cibler les individus les plus disposés à présenter un comportement donné, Model Producer est donc une solution performante et ergonomique.

3.6.2 Scalabilité

Le jeu de données d'origine a été échantillonné avec remise pour produire des jeux de données de plus grande taille. Aucun problème n'a été rencontré jusqu'au million de lignes pour la table centrale. Cependant, nous rencontrons actuellement des soucis pour des jeux de données de 10 millions de lignes, soit un volume total d'environ 7 Go. La recherche des paramètres à régler est toujours en cours.

Conclusion

L'approche par compression de code est le fruit d'une convergence entre les recherches en statistique bayésienne et les recherches en théorie de l'information. Le score obtenu est d'une qualité tout à fait satisfaisante, mais ne surpasse pas nécessairement ce qu'on peut obtenir avec d'autres méthodes prédictives (par exemple une régression logistique ou une forêt aléatoire). La construction du modèle est cependant très largement plus rapide.

La discrétisation automatique des variables et la construction d'agrégats à partir de tables périphériques permettent un gain de productivité conséquent et simplifient les tâches de data management préalable.

L'interface ergonomique aide à produire des rapports aux formats html et pdf afin de transmettre une information appropriable par des experts métiers non spécialisés en data science.

Les procédures d'import de fichier et les conventions de nommage facilitent l'automatisation des traitements mais peuvent sembler contraignantes. La restriction au format UTF-8 peut exiger des conversions préalables et fastidieuses.

La technologie d'installation via le chargement d'une machine vagrant à partir d'un serveur extérieur a montré quelques instabilités sur un serveur équipé du système d'exploitation Debian Jessie. Elle exige aussi une attention particulière aux paramètres de virtual box pour le choix des répertoires d'installation. Elle conduit cependant une installation rapide et sécurisée.

La scalabilité a été démontrée pour quelques millions de lignes, mais on rencontre toujours un problème d'erreur interne au-delà de 10 millions de ligne pour la table centrale (pour des volumes de l'ordre de la dizaine de gigaoctets).

Enfin, l'utilisation de l'API Model Producer à l'aide d'un SDK Python afin d'automatiser des scores dans des process asynchrones n'a pas encore été testée.

En conclusion, Le Model Producer permet d'obtenir des scores de qualité tout à fait satisfaisante en termes d'indice de Gini et de robustesse, et dans des délais qui améliorent la productivité d'un facteur parfois supérieur à 10 en présence de nombreuses prédictrices et/ou de mesures répétées de certaines variables prédictrices.

References

M. Boullé. Khiops: a statistical discretization method of continuous attributes. *Machine Learning*, 55(1):53–69, 2004.

M. Boullé. Modl: a bayes optimal discretization method for continuous attributes. *Machine Learning*, 65(1):131–165, 2006.

M. Boullé. Towards automatic feature construction for supervised classification. In *ECML/PKDD 2014*, pages 181–196. Springer-Verlag, 2014.

Remerciements

Les travaux présentés dans ce document ont été financés, dans le cadre de la chaire Décisionnel-Connaissance Client de la Fondation UBS, par les entreprises suivantes : Crédit Agricole de Bretagne, Groupe Avril, Cofilmo et Business&Decision. La chaire D-CC remercie Predicsis pour la mise à disposition d'une licence ayant permis de tester le logiciel Model Producer.