ENSAE – Spécialisation Actuariat

Mémoire présenté devant l'ENSAE

pour l'obtention du diplôme d'Actuaire ENSAE

et l'admission à l'Institut des Actuaires

le 4/11/2015

Par : Arthur Lucchino

Titre: Optimisation du ciblage client dans un centre d'appel

Confidentialité : ☐ NON ☑ OUI (Durée : ☐ 1 an ☑ 2 ans)

Les signataires s'engagent à respecter la confidentialité indiquée ci-dessus

Membre présent du jury de l'Institut des Actuaires :

Signature :

Entreprise :

Nom : GIE AXA

Signature :

Directeur de mémoire en entreprise :

Nom : Lionel Cassier

Signature :

Membres présents du jury de l'ENSAE :

Autorisation de publication et de mise en ligne sur un site de diffusion de documents actuariels *(après expiration de l'éventuel délai de confidentialité)*

Signature du responsable entreprise :

Secrétariat :

Bibliothèque :

Signature du candidat :

ENSAE ParisTech, 3 Avenue Pierre Larousse, 92245 Malakoff cedex

Résumé

La richesse des données dont disposent les compagnies d'assurances, l'essor des technologies digitales couplé à la possibilité d'utiliser de nouveaux algorithmes permettant d'en tirer une valeur tangible obligent les compagnies d'assurance à se transformer.
Au sein du Data Innovation Lab, catalyseur de cette transformation au sein d'AXA, cette étude tente d'amorcer un changement dans la façon de piloter un processus marketing de Direct Assurance de relance téléphonique suite à un devis, autrefois piloté par une expertise métier, à travers plusieurs axes.
D'abord, les données utilisées par les systèmes d'informations existants ne permettent pas de prendre en compte le risque et la prévision des bénéfices : ainsi, le bénéfice réel induit par une souscription influait peu sur la décision de contacter un prospect.
Grâce aux nouveaux algorithmes de *machine learning*, la modélisation de l'impact de l'action marketing (l'*uplift*) sur les prospects peut être estimée de façon précise. Les algorithmes d'*uplift* permettent de cibler les prospects sur lesquels l'effet de la relance est le plus important.
Enfin, cette transformation s'opère à travers des changements dans l'évaluation des performances des actions marketing remettant les statistiques et les données au centre du jeu, notamment via des A/B tests.
L'objectif de ce mémoire est de montrer l'impact de ces axes de transformation à travers un *backtesting* des modèles produits pour attirer l'attention sur ces nouvelles méthodes. Néanmoins, la transformation digitale est large et requiert des changements massifs des systèmes d'information, de production et de pilotage de l'entreprise qui dépassent largement le cadre de l'étude.

Abstract

The wealth of data now available to insurance companies and the rise of digital technologies together with the possibility to use innovative algorithms allowing the extraction of some tangible value, force insurance companies to quickly transform.
In the Data Innovation Lab, catalyst of this transformation within AXA, this study attempts to initiate change in how to pilot a Direct Assurance marketing process of outbound calls following an insurance quote, formerly driven by business expertise, through several axes.
First, the data available in the existing information systems does not allow to take into account the risk and forecast profits: thus, the real benefit induced by a new contract had little influence on the decision to contact a prospect.
With the new machine learning algorithms, modeling of the impact of marketing action (known as uplift) on prospects can be estimated in an accurate way. The uplift algorithms allow to target prospects on which the effect of the outbound call is the highest.
Finally, this transformation needs changes in the performance evaluation of marketing actions and put statistics and data in a central position, for example through A / B testing.
The objective of this thesis is to show the impact of these transformation axes through backtesting of new models, in order to draw attention to these new methods. However, digital transformation is broad and requires massive changes of information systems, production and management of the company well beyond the framework of the study.

Remerciements

Je tiens à remercier avant tout mon tuteur Lionel Cassier, pour m'avoir donné l'occasion d'effectuer mon stage de fin d'étude au sein du Data Innovation Lab et aux côtés duquel j'ai pris beaucoup de plaisir à travailler. Ses précieux conseils et sa disponibilité m'ont été utiles au quotidien dans mon travail et pour la rédaction de ce mémoire.

Je remercie également Philippe Marie-Jeanne, directeur du DIL, et Emmanuel Néré, directeur de l'équipe Business Transformation du Data Innovation Lab pour m'avoir accueilli dans une équipe dynamique et compétente au sein de laquelle j'ai énormément appris durant ces six mois.

Merci à Julien Iris, chef de projet au DIL avec qui j'ai beaucoup apprécié travailler.

Merci à mes équipiers *Data Scientists* qui m'ont épaulé au quotidien : Nicolas Thiébaut, Yse Wanono et Paul-Arthur Oddon

Merci également aux *Data Scientists* d'AGD, Ahmed Besbes pour ses conseils et sa disponibilité et Mamadou Diaby pour son aide précieuse.

Enfin, je tiens à remercier mes collègues du "garage" et mes co-stagiaires, ainsi que toute la "famille" du Data Innovation Lab pour son accueil chaleureux.

Note

Les chiffres contenus dans ce rapport ont été volontairement modifiés pour des raisons de confidentialité.

Sommaire

Résumé **iii**

Remerciements **v**

Note **vii**

Sommaire **xi**

Introduction **1**

1 Contexte **3**

2 Problématique *business* du projet *Hot Lead Management* **11**

3 Traitement et exploration des données grâce aux outils *Big Data* **25**

4 Outils de *machine learning* utilisés **37**

5 Premiers résultats et impact *business* **57**

Conclusion **73**

Bibliographie **75**

Table des figures **78**

Table des matières **81**

Introduction

Ce mémoire traite d'un projet mené au sein du Data Innovation Lab d'AXA, entité chargée de promouvoir les technologies *Big Data* et la *Data science* à l'échelle du groupe. La naissance du Data Innovation Lab il y a deux ans est le fruit d'une volonté du groupe AXA de se transformer en profondeur et d'exploiter au rythme des avancées technologiques et théoriques de la *Data science* la multitude de données dont il dispose. Le Data Innovation Lab travaille en synergie avec toutes les entités du groupe en proposant des solutions *Data Science* innovantes et adaptées à leur besoins.

Le projet décrit dans ces pages a été conduit conjointement avec Direct Assurance, filiale d'AXA Global Direct dont le cœur de métier est l'assurance directe. Il a pour objectif l'optimisation des ressources des centres d'appels de Direct Assurance, avec en premier lieu une focalisation sur le processus de relance téléphonique suite à des devis édités en ligne.

L'optimisation des campagnes marketing possède également une dimension actuarielle, dans le sens où elle permettent par l'utilisation de variables nouvelles d'identifier des profils de client particuliers. Ces campagnes intègrent notamment des considérations de valeur client, indicateur clé dans la mesure des bénéfices. Certaines informations déterminantes en assurance directe, telles que les prix de la concurrence, permettent également de mieux comprendre les comportements des assurés sous un angle microéconomique.

Ce projet s'inscrit parfaitement dans le processus de transformation d'AXA, dans la mesure où les résultats attendus, s'ils se révèlent concluants, permettront peut-être de changer en profondeur la méthodologie du marketing chez AXA et de décliner les modèles à d'autres cas d'usage. En plus de reposer sur un indicateur particulièrement adapté au problème mais peu utilisé jusque là, l'*uplift*, l'essor des technologies digitales permettra de *tracker* ces modèles et de les raffiner "en continu". Cela permettra de fournir des éléments concrets en faveur d'une approche randomisée du marketing (A/B *testing*), qui permet de construire des modèles performants et de s'affranchir des biais de sélection.

La méthode choisie repose sur trois briques élémentaires : la modélisation de la mise en relation téléphonique, la modélisation de l'effet produit par la relance sur la souscription et une analyse coûts-bénéfices, prenant en compte les coûts du centre d'appel et le modèle de valeur client élaboré par Direct Assurance.

Après quelques éléments de contexte *business*, décrivant les enjeux et difficultés de la

mission du Data Innovation Lab, de sa structure et de son mode de fonctionnement, on présentera les enjeux métier du projet *Hot Lead Management* et l'approche de modélisation retenue en décomposant les trois leviers identifiés (joignabilité, *uplift* et valeur client). Un chapitre détaillera le processus existant, les contraintes et biais qu'il implique pour la modélisation. L'accent sera ensuite mis sur les technologies *Big Data* utilisées au cours du projet, notamment le calcul distribué et les bibliothèques de *machine learning* des langages de programmation classiques de *Data Science*, qui constituent le socle nécessaire à l'élaboration de modèles performants. Une brève description des bases de données utilisées sera faite, avec notamment des sources nouvelles encore très peu utilisées par le métier.

Une analyse approfondie de la modélisation suivra, avec notamment le cadre théorique et les hypothèses effectuées concernant les biais liés aux données existantes. On consacrera une partie du mémoire aux aspects plus techniques du *machine learning*, notamment les problématiques de la classification binaire et de la mesure des effets de traitement (*uplift*), avec un état de l'art des algorithmes utilisés.

On exposera ensuite les résultats de la phase exploratoire du modèle, permettant d'identifier certaines variables d'intérêt et de confirmer ou d'infirmer certaines considérations du métier, puis les premiers résultats de modélisation et le développement futur à apporter au modèle. Une dernière partie d'ouverture sera consacrée aux méthodes d'*active learning* applicables à notre problématique qui permettent d'améliorer en continu des modèles en se basant sur un arbitrage exploration/exploitation (algorithme des bandits contextuels).

Chapitre 1

Contexte

1.1 Le groupe AXA : une somme d'entités

1.1.1 Le groupe AXA

AXA est une entreprise multinationale française dont le cœur d'activité est l'assurance et la gestion d'actifs.

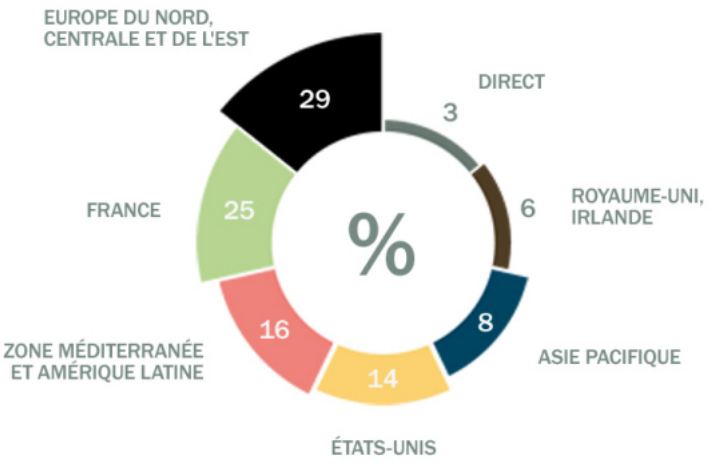


FIGURE 1.1: Répartition du chiffre d'affaire du groupe AXA à travers le monde

Les activités du groupe s'étendent dans le monde entier. C'est en France et en Europe qu'AXA reste le plus actif. AXA est une structure très décentralisée, dans laquelle la direction du groupe coordonne la stratégie globale sur le long terme. Elle est chargée de définir le modèle économique du groupe, ses objectifs, sa politique de développement mais également son image. Ces décisions stratégiques prises à l'échelle du groupe concernent à la fois les différents cœurs de métier (AXA P&C, AXA Life, AXA Santé), mais aussi les activités transverses gérées par des équipes dédiées : le marketing, l'informatique (IT), la stratégie digitale. Ces activités sont structurantes dans l'évolution du groupe, et nécessitent

une stratégie commune et globale.

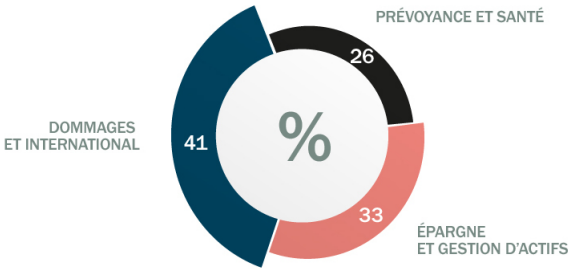


FIGURE 1.2: Résultat du groupe par secteurs d'activité

Un des exemples qui illustre ces prises de décision stratégiques est la transformation *Big Data* du groupe, sur laquelle nous reviendrons plus en détail. A travers la création d'équipes de *Data Scientists* et d'experts mondialement reconnus au sein des entités, la volonté du groupe est de cibler grâce aux données les processus métiers susceptibles de contribuer à la croissance du groupe.

Le Data Innovation Lab, au sein duquel j'ai effectué mon stage, est également un produit de cette stratégie de transformation, destiné à promouvoir l'utilisation de la *Data Science* et la transformation digitale chez les entités.

1.1.2 Les entités du groupe AXA

La structure du groupe AXA laisse une grande part d'autonomie à ses entités. Parmi celles les plus importantes en terme de chiffre d'affaires, citons AXA Global P&C, AXA Global Life, AXA Investment Managers.

Il appartient à chacune de ces entités de mener sa propre politique à court et moyen terme. Le groupe n'interviendra qu'au sommet de sa hiérarchie pour fixer des objectifs de résultat. Ses décisions n'affecteront a priori que leur stratégie de développement sur le long terme ou leur gouvernance.

1.1.3 AXA et la révolution *Big Data*

Par le terme "Big Data", on regroupe toutes les technologies, méthodes mathématiques et informatiques qui traitent de l'accès, du stockage et du traitement des données massives. La *Data Science* regroupe l'ensemble des techniques mathématiques et statistiques, en particulier de *machine learning*, utilisées pour en tirer de l'information.

L'avènement de l'informatique, puis d'Internet et des technologies connectées a permis d'accéder à une quantité innombrable de données, que l'on est désormais capables de stocker et de traiter.

Pour les grandes entreprises en particulier, l'enjeu est de taille : par diverses méthodes

statistiques, il est possible d'extraire de ces données de précieuses informations qui pourront être utilisées pour comprendre et anticiper le comportement et les besoins de leur clients.

Le métier de l'assurance est particulièrement concerné par l'exploitation des données massives, à la fois pour acquérir et fidéliser sa clientèle mais aussi pour mieux évaluer son exposition au risque. Le modèle même de l'assurance dommages est amené à évoluer vers le digital grâce aux objets connectés, qui permettent de récolter en temps réel des données comportementales sur les assurés. Le processus de souscription est également modifié avec le développement de l'assurance directe, qui s'affranchit des intermédiaires "physiques" (agents, courtiers).

Cette révolution peut être analysée sous deux angles. D'abord, elle permet d'améliorer des processus existants basés jusque là sur des règles issues d'avis d'experts, à la fois grâce aux outils de *Data Science* qui élaborent des nouvelles règles à partir d'algorithmes calibrés sur les données, et aux outils de *processing* des données qui permettent de mesurer la performance de ces processus et de les adapter pour améliorer leur efficacité, en étirant un processus en trois étapes : *tracking*, évaluer, raffiner.

D'autre part, les objets connectés modifient l'interface assureur/assuré et facilitent la communication grâce à l'immédiateté de l'information (exemple de l'assurance santé où certains objets connectés accélère certaines interventions). Ils sont ainsi une porte ouverte à l'innovation en terme d'offre de produits et de prévention. Citons l'exemple de l'application télématique développée par le DIL, outil de localisation et d'analyse de la conduite qui permet à la fois d'adapter la prime de l'assuré en analysant son comportement et d'individualiser la politique de prévention.

Le marketing fait également partie des grands bénéficiaires des objets connectés, par lesquels on peut à la fois récolter de l'information sur les clients, mieux les connaître et leur proposer les offres les plus adaptées.

Pour faire face à cette évolution, AXA a pour objectif d'inculquer progressivement la culture Big Data à ses entités pour qu'elles deviennent, à terme, *data-driven*, c'est-à-dire tirant parti l'information issue des données pour faciliter la prise de décision.

1.2 Le Data Innovation Lab

1.2.1 Présentation

C'est dans cette optique qu'a vu le jour en janvier 2014 le Data Innovation Lab (DIL), en tant que centre d'expertise pour les technologies Big Data et pour les projets à composante Data Science au sein du groupe AXA.

Le DIL a pour vocation d'accompagner les entités d'AXA dans leur transformation. Il mène en collaboration avec les entités des projets dans lesquels l'utilisation des données apporte une valeur ajoutée par rapport aux processus existants. En mettant au service

du groupe ses compétences à travers plusieurs cas d'usage, le DIL livre des solutions sur mesure suivant les besoins particuliers des entités, mais doit également remplir sa mission d'aide à la transformation en créant des solutions réutilisables à l'échelle du groupe. Le DIL est ainsi soumis à des exigences de performance court terme au même titre qu'un prestataire de service, tout en adaptant ses solutions à une problématique globale inscrite sur le long terme.

1.2.2 Mission et défis du Data Innovation Lab

La mission de transformation du DIL se décline en plusieurs parties.

Le DIL doit d'abord interagir au mieux avec le métier pour lancer des projets, à la fois en prospectant auprès des entités, en diffusant son catalogue d'offre de services et en étant réceptif à de nouveaux besoins de la part des entités. Le lancement et la réalisation d'un projet requièrent d'assimiler rapidement les processus qui régissent le *business* et de s'adapter à son fonctionnement interne.

L'IT joue un rôle essentiel dans le *Big Data* et donc dans la transformation. Aux côtés d'AXA Tech, entité dédiée aux ressources informatiques d'AXA, le DIL doit participer à la création et au développement des nouvelles plateformes de travail IT (notamment le développement du calcul distribué) en apportant, en tant qu'utilisateur privilégié, un *feedback* et des suggestions pour permettre à l'environnement IT d'évoluer pour satisfaire les ambitions digitales du groupe.

En tant que centre d'expertise en *Data Science*, une des missions premières du DIL est de réfléchir aux possibles solutions du volet *Data Science* d'un projet et de proposer des modèles et des algorithmes innovants adaptés aux données. Ainsi, les projets comportent une composante importante de *brainstorming* auxquels participent également les experts des entités.

Cette expertise est également mise à profit pour élaborer des programmes de formation en *Data Science* destinés aux entités. La sensibilisation des entités à la *Data Science* a pour but de les accompagner dans leur transformation.

Les compétences du DIL se prêtent bien aux projets orientés marketing, dans lesquels l'utilisation des données apporte une grande valeur ajoutée. Une étude de l'existant couplée à un travail de modélisation permet de repenser certains processus métier. Ceci s'applique notamment au ciblage d'offres ou de réductions et aux canaux de communication à utiliser.

Amené à travailler avec les entités sans en faire véritablement partie, le DIL se heurte fréquemment aux difficultés de communication entre les entités impliquées dans un projet, lui faisant parfois jouer un rôle de médiateur.

Un des principaux écueils auxquels est confronté le DIL est l'héritage laissé des entités. La transformation *data-driven* doit souvent passer par une refonte de certaines règles métier, ce que ne peut se permettre l'entité qui doit se tenir à des obligations de performance. Le projet *Hot Lead Management*, sur lequel j'ai travaillé, est un exemple de nombreux projets

pour lesquels les données disponibles sont issues des processus existants qui biaisent dans une certaine mesure les modèles. Convaincre les différents acteurs qu'il faut parfois "reculer pour mieux avancer" en relâchant certaines contraintes du métier, (et éventuellement sacrifier des bénéfices immédiats) pour pouvoir en tirer profit a posteriori est un travail que doit mener le DIL. L'exemple des campagnes marketing aléatoires (lié au projet *Hot Lead Management*), nécessaires pour tester au mieux leur efficacité, illustre cette problématique.

Le même genre de problème se transpose à l'IT : les structures et l'allocation des ressources ne sont pas toujours adaptées aux plateformes de travail, en particulier le travail collaboratif avec les entités. La mise en place d'une architecture plus flexible se heurte aussi à l'inertie des processus, inhérente aux structures informatiques à grande échelle (AXA Tech intervient de manière transverse chez toutes les entités). Ceci se traduit par exemple dans le quotidien des projets par la difficulté d'implémenter du code collaboratif, et la difficulté de s'adapter aux technologies informatiques qui évoluent très vite (en particulier le web).

1.2.3 Organisation et gouvernance

En tant qu'entité transversale au groupe AXA, le DIL est rattaché au GIE AXA (Groupe d'Intérêts Economique), groupement qui pilote la politique du groupe et dont fait partie la direction générale. Le DIL dépend directement de la COO (Chief Operating Officer) du groupe, Véronique Weil.

Le DIL est dirigé par Philippe Marie-Jeanne, dirigeant historique de la branche P&C d'AXA France. Son expertise technique et son niveau de séniorité au sein du groupe lui permettent de "catalyser" la transformation digitale aux côtés de la COO. Il est composé de quatre équipes :

— **L'équipe Business Transformation**, chargée d'identifier les besoins des entités et de proposer des solutions adéquates. Pour les entités du groupe, c'est l'interface par laquelle les nouveaux projets sont lancés en collaboration avec le DIL. Je suis moi-même rattaché à cette équipe.
L'équipe BT, dirigée par Emmanuel Néré, est composée :

- de chef de projets, au profil commercial, chargés du *management* du projet et de communiquer avec l'entité cliente.
- de *data scientists*, au profil scientifique (essentiellement des ingénieurs) qui travaillent sur la partie modélisation des projets et l'implémentation des algorithmes.

— **L'équipe Recherche et Développement** (R&D), qui travaille autour des objets connectés (télématique, maison connectée, santé connectée,...) et à l'élaboration de produits innovants sur le moyen et long terme. Elle est composée de *project designers*, d'universitaires qui apportent leur expertise dans le domain de la recherche, et de *data scientists*

— **L'équipe *Engineering***, chargée de mettre en place l'architecture IT nécessaire à la mise en place des projets et de développer les outils réutilisables à la fois pour le DIL et les entités. Elle est formée par des développeurs au profil très IT et des *data science engineers* qui font l'interface entre les technologies utilisées par le DIL et la

data science, et dont les compétences sont mises à profit pour la mise en production des projets.

— **L'équipe transversale**, chargée des problématiques légales, d'harmoniser les relations avec les entités et de structurer de façon plus générale le fonctionnement interne du DIL. Une de ses missions principales est la gestion de la confidentialité des données (*data privacy*) qui est un enjeu de taille pour le DIL. Par ailleurs, elle met en place et développe en collaboration avec les *data scientists* des différentes entités des formations en *data science* au sein d'AXA.

La taille et la structure du DIL en font une entreprise à taille humaine, semblable dans son fonctionnement à une *start-up* au sein d'AXA. La proximité des différentes équipes rend leur collaboration facile, alors que l'on est très souvent amené à consulter les *Data Engineers* et *Data Scientist* des autres équipes. Du fait de la relative jeunesse de ces métiers, une émulation permanente autour de la *Data Science* et des outils *Big Data* favorise la créativité et la montée en compétence.

1.2.4 Cycle de vie d'un projet

Le DIL fonctionne à la fois en mode *process* pour l'élaboration de solutions long terme (essentiellement développées par l'équipe R&D) et en mode projet pour tester à court terme des solutions et déterminer si elle possèdent une réelle valeur ajoutée pour l'entité. Un projet qui implique le Data Innovation Lab se décompose généralement en trois étapes :

— L'avant-projet : Les besoins de l'entité sont formalisés. Le DIL s'approprie les processus et les règles existantes du métier, un dialogue avec l'entité est installé afin de cerner si le projet contient une valeur *business*, et d'envisager le cas échéant les solutions possibles en termes de *machine learning* et d'architecture IT.

— Le POC (Proof of concept) : Les *Data scientists* et *Data engineers* du DIL développent, en collaboration avec ceux de l'entité si celle-ci possède les compétences nécessaires, les outils et algorithmes nécessaires pour mener à bien les objectifs du projet. C'est la phase dans laquelle se trouve actuellement le projet *Hot Lead Management* sur lequel je travaille depuis la phase d'avant-projet, dont le POC contenait également une phase exploratoire plus descriptive. A l'issue du POC, une décision est prise sur la mise en production ou non du projet.

— La mise en production : Cette phase correspond au déploiement du projet chez les entités, et comporte une grande partie de développement IT pour intégrer les outils et algorithmes du POC aux systèmes utilisés par les entités. C'est une étape clé du projet, qui comporte des difficultés de réconciliation des outils utilisés en POC et en production, qui ne sont généralement pas les mêmes. Un travail de développement IT est nécessaire pour traduire des parties du code utilisé. Plusieurs scenarii doivent être envisagés en fonction des capacités de déploiement des entités.

1.3 AXA Global Direct

1.3.1 Présentation

Le projet auquel j'ai contribué impliquait AXA Global Direct (AGD), qui regroupe les activités d'assurance directe du groupe à travers le monde, et plus particulièrement Direct

Assurance, sa filiale française, qui propose des contrats auto et habitation.

1.3.2 Le *Business Model* de l'assurance directe

L'assurance directe se distingue des activités d'assurance classiques par l'absence d'intermédiaire entre le client et l'assureur.

Ces intermédiaires sont traditionnellement des agents généraux, affiliés à une compagnie d'assurance sans en être salariés et rémunérés par commission sur les contrats souscrits, ou des courtiers en assurance mandatés par les clients pour proposer une offre de contrats provenant éventuellement de divers assureurs.

L'assurance directe permet de souscrire des contrats immédiatement par téléphone ou par internet : ce nouveau modèle s'adapte aux nouvelles technologies digitales, ainsi qu'au mode de vie et attentes d'une partie de la clientèle qui privilégient davantage les prix réduits et les produits simples qu'une relation personnalisé avec leur agent. Ce comportement est de plus en plus fréquent pour les produits dits de "commodités" (assurance auto, habitation,...) où les produits sont plus simples et plus standardisés que peuvent l'être les produits d'assurance vie et d'épargne. C'est la raison pour laquelle Direct Assurance se focalise sur ce type de produit où l'interaction humaine est moins valorisée.

L'aspect essentiellement digital de l'assurance directe et la maturité technologique et technique d'AGD en font un terrain idéal pour tester et développer des solutions Big Data. En effet, en l'absence de réseau de distribution, le marché de l'assurance directe est beaucoup plus liquide et donc soumis à une plus grande concurrence à travers les comparateurs (cyber-courtiers). Les primes affichées doivent être calculées de façon très fine car le classement et l'écart tarifaire par rapport à la concurrence sont des variables clefs dans la décision du client. Une mise à jour permanente est nécessaire sous peine de voir sa part de marché diminuer. Les primes doivent aussi être réduites en contrepartie de l'absence de relation client personnalisée via l'agent général. Enfin, le marketing joue un rôle primordial, d'autant que l'accès à la donnée se trouve facilité par les canaux digitaux (en ligne et par téléphone), qui permettent aussi un *tracking* permanent des processus.

Vu sous un autre angle, la flexibilité des tarifs en assurance directe (pas de diffusion nécessaire dans le réseau de distribution) est un avantage lorsqu'il s'agit de mettre en place rapidement de nouvelles stratégies de *pricing* et de marketing. Ces stratégies peuvent ensuite être facilement évaluées et corrigées grâce au *tracking* des prospections et des souscriptions sur les canaux digitaux, favorisé par des outils informatiques plus mûrs que ceux du métier traditionnel de l'assurance.

Chapitre 2

Problématique *business* du projet *Hot Lead Management*

2.1 Fonctionnement des centres d'appels Direct Assurance

2.1.1 Canaux de souscription

Direct Assurance propose des devis gratuits par téléphone ou par internet à ses clients. Ces derniers peuvent contacter directement un conseiller en assurance (CEA) opérant sur les plateaux téléphoniques, ou accéder au site internet DA, mais peuvent également choisir d'être redirigés sur ce site suite à une recherche en ligne de devis via un comparateur d'assurance (aussi appelé agrégateur ou cybercourtier). Ces comparateurs font appels aux web service des différents assureurs en lice pour un même produit et proposent ensuite les différents tarifs à l'internaute.

Le parcours d'un client peut se décomposer en quatre phases :

— Le tarif vu (ou *quote*) sur internet : cette phase ne concerne que les devis en ligne via les comparateurs. A ce stade, le devis est incomplet et les informations personnelles du prospect ne sont pas communiquées aux assureurs. Celui-ci peut choisir d'être redirigé vers le site DA s'il souhaite poursuivre son devis et communiquer ses informations personnelles, ou d'être mis en relation téléphonique avec un CEA

— Le tarif édité en ligne ou par téléphone : suite à un devis, le client transmet ses informations personnelles (identité, numéro de téléphone) et peut dès lors être recontacté. Si le devis provient des comparateurs, cette étape correspond au moment où ce dernier clique sur le tarif DA proposé et est redirigé vers le site internet DA.

— La mise en relation téléphonique avec un CEA, suite à un appel de la part du client ou une relance téléphonique. Cette phase n'a pas lieu si le client a immédiatement souscrit en ligne.

— La souscription, qui se matérialise par un premier règlement d'une partie de la prime. Le client dispose alors d'un délai de 30 jours au cours duquel il peut soit se rétracter, soit voir son contrat annulé s'il n'a pas fourni les pièces justificatives nécessaires. Une fois que le contrat est validé à l'issue du délai de 30 jours, on qualifie le contrat d'affaire nouvelle nette.

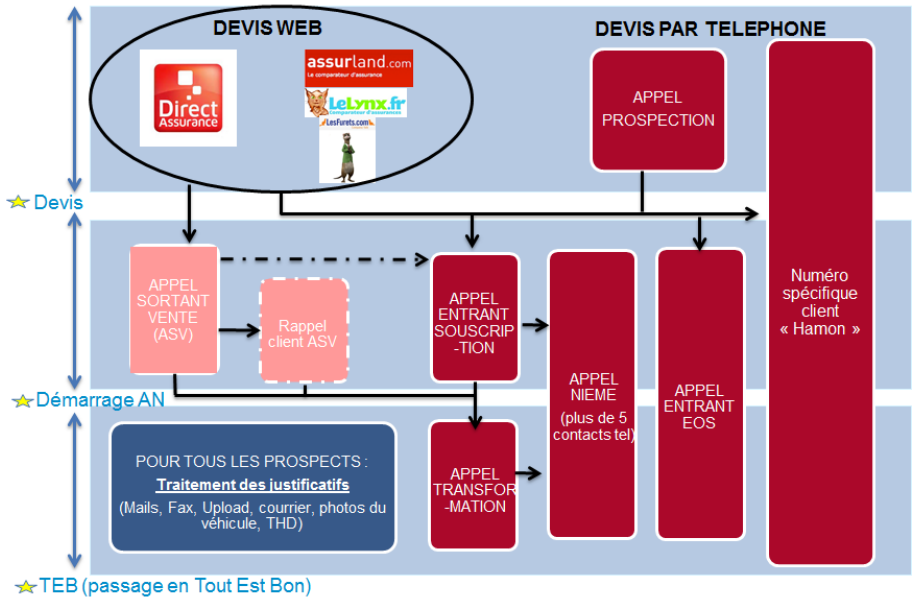


FIGURE 2.1: Processus de souscription

Le parcours d'un prospect depuis son premier devis jusqu'à, le cas échéant, la souscription et la validation, n'est pas toujours facile à tracer. La difficulté est de rattacher plusieurs actions de nature diverses, effectuées sur des canaux multiples. Certains clients peuvent prospecter dans un premier temps en passant par un agrégateur, puis se renseigner davantage par téléphone, éditer à nouveau leur devis sur le site DA, rappeler à nouveau pour obtenir des informations supplémentaires, souscrire en ligne puis fournir leur pièces justificatives en plusieurs étapes.

Ce parcours peut être une source d'information précieuse pour identifier certains profils et les traiter de façon optimale. Il est notamment intéressant de repérer certains comportements lors de la phase de prospection en ligne, ou d'identifier quels types de clients souscrivent rapidement ou au contraire nécessitent un travail supplémentaire. A travers l'étude du processus d'appels sortants vente décrit dans la partie suivante, le projet a pour objectif d'utiliser ces informations de parcours pour mieux cibler les campagnes marketing.

2.1.2 Les appels sortants vente

Un des leviers marketing utilisés par Direct Assurance est la relance téléphonique à chaud suite à l'édition d'un devis en ligne (deuxième étape décrite dans la partie précédente), appelée "appel sortant vente" (ASV).

Ces relances ont pour but d'inciter le client à souscrire en le mettant le plus rapidement possible en relation avec un CEA, qui pourra alors développer un argumentaire adapté. Un outil de rappel automatique paramétrable (ViaDialog) est utilisé pour déclencher les ASV. Le processus mobilisant une partie des ressources des plateaux téléphoniques, un algorithme de priorisation basé sur l'expérience métier est mis en place pour trier les devis édités et les placer dans des files de rappel (voir ci-dessous). Les relances suivent alors un processus, lui aussi priorisé.

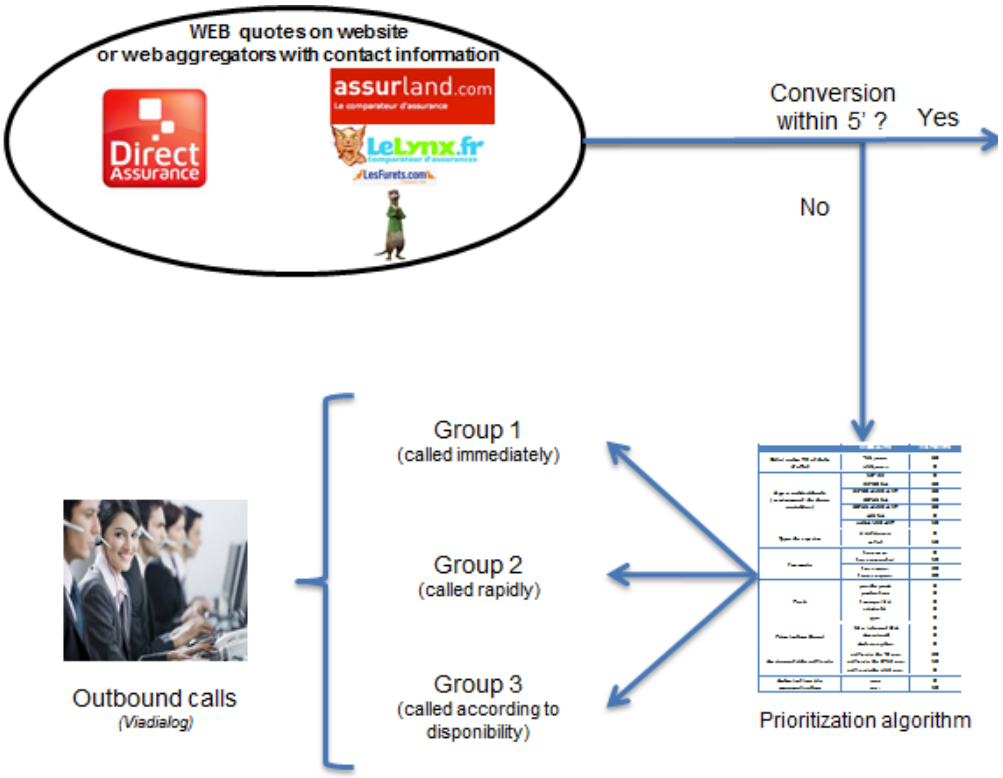


FIGURE 2.2: Processus d'appel sortant vente

Les ASV constituent un levier d'action pour améliorer le taux de souscription, comme le prouve une étude menée par les équipes marketing sur le premier semestre 2014.

Devant les résultats concluants de cette première étude, le projet Hot Lead Management vise à refondre l'algorithme utilisé par le métier, basé sur des règles simples, pour tirer davantage profit des ASV grâce aux données à disposition sur les devis et le parcours client.

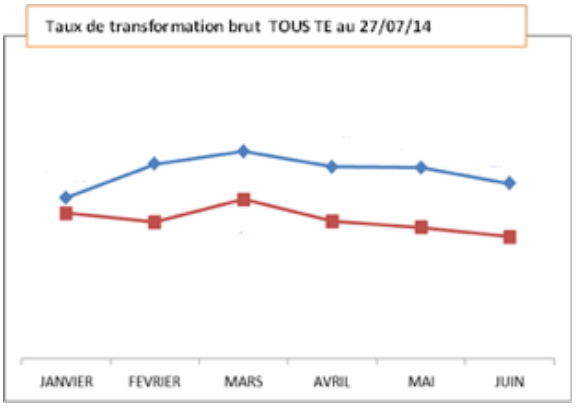


FIGURE 2.3: Conversion brute avec et sans ASV

2.2 Objectifs du projet

2.2.1 Amélioration du score existant

L'objectif initial du projet Hot Lead Management concerne l'algorithme de priorisation des appels sortants. En utilisant des sources de données variées (données tarifaires, données de la concurrence, données de navigation internet, données socio-démographiques,...) et des méthodes de *machine learning*, on cherche à construire un score pour déterminer l'effet de la relance téléphonique sur un prospect et remplacer l'algorithme déjà en place. Nous reviendrons plus en détail sur les modèles utilisés, basés notamment sur la notion d'uplift qui mesure l'effet d'un traitement sur une variable cible.

Une fois établi le score de priorisation, on pourra l'intégrer au processus et rechercher le seuil optimal de devis à rappeler en intégrant les coûts du plateau téléphonique. Les aspects de mise en production et l'implémentation, si possible en temps réel, de ce score dans le processus métier seront discutés dans un second temps.

2.2.2 Périmètre élargi

De façon plus macroscopique, on peut s'intéresser à l'allocation des ressources des plateaux téléphoniques par canal d'activité (appels entrants prospection, ASV, travail sur dossier,...). Il s'agit d'un problème d'optimisation dynamique qui requiert d'évaluer les coûts et les bénéfices des différents types d'activités pour rechercher à chaque instant l'optimum d'allocation.

La mise en équation de ce problème est complexe, car certains de ces coûts et bénéfices sont difficilement mesurables : par exemple, le bénéfice tiré du traitement d'un appel entrant, vu également comme un effet de traitement, ne peut être directement calculé puisqu'on peut difficilement savoir ce qui se serait produit par la suite si l'appel n'avait pas été pris (nous reviendrons plus tard sur les effets de traitement, qui nécessitent dans l'idéal d'avoir recours à des phases exploratoires ou essais randomisés). La mesure non biaisée de l'efficacité d'une stratégie d'allocation par rapport à une autre passe par des A/B tests, similaires aux essais randomisés.

2.3 Comment évaluer la performance du processus ?

2.3.1 Limites du score métier

Afin d'identifier de possibles améliorations immédiates (*quick wins*) ou plus fines (grâce à des modèles de *machine learning*) qui apporteraient de la valeur au processus d'ASV, examinons de plus près l'algorithme de priorisation existant.

FIGURE 2.4: Algorithme existant

Variable	Modalité	Score
Délai TE / date d'effet	A	x1
	B	x2
Age * antecedents	A	x1
	B	x2
	C	x3
	D	x4
Type de reprise	A	x1
	B	x2
	C	x3
Formule	A	x1
	B	x2
	C	x3
	D	x4
Pack	A	x1
	B	x2
	C	x3
	D	x4
	E	x5
Canal d'origine	A	x1
	B	x2
	C	x3
	D	x4
Ancienneté véhic.	A	x1
	B	x2
	C	x3

Le score mis en place par les équipes marketing de DA se base sur l'expérience métier. La notion de valeur client y est présente, mais sous sa forme la plus simple :

— Les devis prenant effet le jour J sont d'office exclus, car ils sont susceptibles d'être faits sur des véhicules déjà sinistrés pour obtenir une garantie rapide. Le délai entre le devis et la date d'effet est d'ailleurs le critère principal : on priorise les dates d'effet à moins de 15 jours, qui représentent les prospects "pressés".

— Vient ensuite le canal âge/antécédents, une des principales variables de la segmentation actuelle de la valeur client.

— L'ancienneté du véhicule joue également comme variable segmentante. La priorisation des formules souscrites s'explique par une marge en moyenne supérieure sur les formules complètes.

On note que le canal de souscription rentre peu en compte dans le score, alors que les coûts d'acquisition ne sont pas les mêmes selon le canal. La notion de marge n'est pas vraiment présente dans le score, et pourtant varie beaucoup suivant les packs, formules.

D'autre part, on ne considère à aucun moment la joignabilité dans ce score (i.e. la probabilité que le prospect décroche). Or pour bénéficier de la valeur ajoutée de la relance téléphonique, il faut que le prospect soit effectivement joint.

On constate également qu'environ 95% des prospects sont rappelés, ce qui soulève deux questions : d'abord, ceci empêche une analyse de l'effet du rappel (puisque le groupe témoin des prospects non rappelés est trop faible en effectif avec seulement 5%, voir plus bas). Ensuite, le process ne semble pas prendre en compte les coûts générés par le centre d'appel : il est possible que certains appels ne valent pas la peine d'être passés si le gain potentiel n'est pas supérieur au coût marginal de l'appel. C'est également une composante que nous tenterons d'optimiser au cours du projet.

Même si le score prend en compte la valeur client, il est possible de la raffiner en l'intégrant de façon plus granulaire (considérer plus de segments, voire calculer une valeur individuelle).

Avant de considérer des modèles de *machine learning* avancés, une première étape du projet consistera à produire une série de statistiques descriptives sur la joignabilité et la conversion, afin d'obtenir des *insights* sur la significativité de certaines variables et éventuellement identifier des améliorations immédiatement implémentables dans le processus actuel.

2.3.2 Indicateurs de performance du processus (KPI)

Le travail de modélisation s'articule autour de l'optimisation de certains indicateurs de performances (*Key Performance Indicators*), mis en évidence à travers l'analyse du score existant.

Un premier indicateur, le plus réducteur, serait le taux de souscription nette (pas d'annulation ni de rejet dans les 30 jours). On ne s'intéresse alors qu'au nombre d'affaires nouvelles, sans prendre en compte les bénéfices ni les coûts. Cet indicateur très simple sera la cible d'un premier modèle d'évaluation de l'impact de la communication sur la conversion.

Pour construire un indicateur de performance plus fin, il est nécessaire d'ajouter à ce modèle une étude de coût du centre d'appel, une étude des bénéfices générés suivant le profil de client via des modèles de valeur client, mais également un second modèle de joignabilité des prospects par téléphone.

Composantes du KPI

L'effet de notre campagne marketing d'ASV sur un individu i se mesure par le produit de trois facteurs :

$$G_i = \mathbb{P}(communication_i) * uplift_i * marge_i - cout\,marginal_i \tag{2.1}$$

La probabilité de communication L'effet de la communication ne s'applique qu'aux prospects qui ont effectivement décroché. Il faut donc tenir compte de la probabilité que le prospect réponde à l'appel, probabilité qui est susceptible de varier suivant l'heure de la journée, la CSP,... comme nous l'avons constaté dans le chapitre précédent. Elle sera estimée de façon plus précise grâce à un modèle de joignabilité.

L'*uplift* La composante d'*uplift* mesure l'effet de la communication sur la souscription : il s'agit du gain que rapporte la communication en terme de probabilité de conversion.

Il constitue l'élément essentiel du KPI puisque c'est celui qui est directement relié à la communication et par lequel on mesure la valeur ajoutée du CEA dans le processus de souscription. Nous reviendrons sur cette grandeur dans la suite de ce chapitre.

La marge La marge associée au contrat que le prospect est susceptible de souscrire est également un élément décisif qui peut contrebalancer l'*uplift*. Bien qu'un appel puisse avoir un *uplift* élevé sur un prospect, une marge espérée trop faible sur son contrat le placera derrière des devis à *uplift* plus faible mais à marge beaucoup plus élevée. Cette marge est principalment associée à un modèle de valeur client (qui sert déjà de référence pour le calcul de la prime).

Cette marge contient également les coûts d'acquisition du devis qui diffèrent essentiellement selon sa provenance (site DA ou agrégateurs).

Le coût marginal associé à un appel Ce terme résume le coût marginal d'un ASV, estimé à partir du coût horaire d'un CEA et du temps cumulé passé par les CEA en activité d'ASV.

Bénéfices agrégés

L'implémentation du processus de priorisation des ASV à partir de l'estimation des gains individuels se fait en optimisant la fonction de coût agrégée.

Si le gain espéré sur chaque prospect est positif, il est clair qu'il faudra en rappeler un maximum, sachant que le processus existant permet de rappeler la quasi-totalité des prospects ($\simeq 95\%$), pour la plupart assez rapidement. Si ce gain n'est pas toujours positif, alors le problème consiste à rechercher le seuil t qui maximise :

$$\sum_{i \in S(t)} G_i \tag{2.2}$$

où $S(t)$ est l'ensemble des individus à rappeler, ce qui revient, après classement des individus par gain espéré décroissant, à maximiser :

$$\sum_{i=0}^{t} G_{(i)} \tag{2.3}$$

ce qui revient à chercher l'indice maximal (i) tel que $G_{(i)} > 0$

2.4 Premier levier : la joignabilité

On ne peut retirer le bénéfice d'une conversation que si le prospect est effectivement joint. Comme la seule action du centre d'appel est le rappel, un modèle de joignabilité est nécessaire à la fois pour pouvoir évaluer le bénéfice de l'algorithme existant et pour que notre nouveau score tienne compte de la joignabilité. Il s'agit d'un problème de classification binaire classique où on cherche à prédire quels prospects vont décrocher.

On ne considère ici que la population des prospects sur lesquels il y a eu au moins une

tentative de rappel. Si l'on désigne par T la variable binaire correspondant à la communication effective (1 si le prospect décroche, 0 sinon), on recherche un classificateur binaire g :

$$\mathbb{P}(T = 1|X) = g(X) \tag{2.4}$$

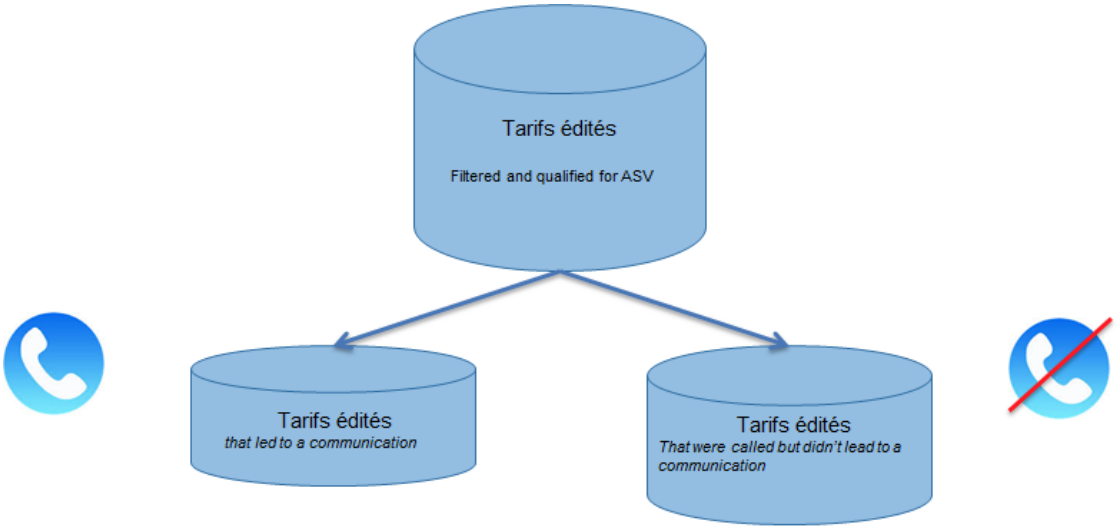


FIGURE 2.5: Modèle de mise en relation

2.5 Second levier : l'effet de la communication

2.5.1 Mesure de l'effet de traitement (*True Lift*)

Lors d'une campagne marketing, la méthode de ciblage fréquemment utilisée consiste à construire un score d'appétence à la variable cible (chiffre d'affaire, bénéfice,...) qui permettra sélectionner une sous-population sur laquelle on effectuera la campagne. Dans notre contexte, cela consisterait à rappeler les prospects ayant les plus grandes chances de souscrire.

Cette méthode à l'avantage d'être simple à mettre en oeuvre, de se prêter au techniques de *scoring* classiques et d'être assez intuitive. Cependant, elle ne mesure pas réellement l'effet de la campagne (le traitement) : un prospect ayant de grandes chances de souscrire n'est pas forcément sensible à la campagne, et souscrirait certainement sans incitations. Pire, la campagne marketing pourrait avoir un effet négatif et dissuaderait ce dernier de convertir.

La bonne grandeur à considérer est ici l'effet de traitement : quel gain rapporte une campagne marketing sur un client particulier ? La mesure de l'effet de traitement, évoquée dans un article fondateur de Rubin (Rubin, 1974) est un problème rencontré dans d'autres disciplines, en particulier l'épidémiologie clinique (effet d'un médicament) ou encore en économétrie (évaluation des politiques publiques). Dans le cadre marketing, on appellera

True Lift ou *uplift* cet effet de traitement. Dans le cas présent, le traitement est la communication suite à un ASV et la variable cible est dans un premier temps la souscription.

La problématique de l'*uplift* peut être résumée par un schéma simple, qui identifie les quatre grandes catégories de clients suivant leur réaction à une campagne marketing :

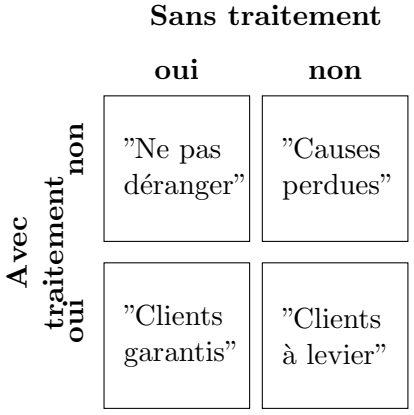


FIGURE 2.6: Catégories de clients

— **Les prospects à "ne pas déranger"** : Ce sont les clients que l'on veut à tout prix éviter de cibler dans une campagne marketing, ceux sur lesquels l'*uplift* est négatif. La campagne marketing aura l'effet inverse et diminuera leur appétence à la souscription.

— **- Les "causes perdues"** : ce sont les prospects qui convertissent mal et sur lesquels la campagne n'a aucun effet. Leur appétence de base est faible et leur *uplift* est nul. Ce sont des clients que l'on ne souhaite pas cibler.

— **Les "clients garantis"** : ce sont les prospects qui convertissent bien et sur lesquels la campagne n'a pas d'effet. Parce que leur *uplift* est nul, on ne souhaite pas non plus cibler ces clients. La campagne marketing n'a pas d'impact sur eux et serait un coût inutile.

— **Les "clients à levier"** : ce sont les clients que l'on souhaite cibler : leur appétence est "boostée" par la campagne marketing qui a dans ce cas un impact fort.

Une seule de ces quatre catégories doit être ciblée. Or un score d'appétence classique, qui ne s'intéresse qu'à la conversion, se tromperait en ciblant en priorité les "clients garantis" et les clients à "ne pas déranger".
Il aurait été intéressant de décomposer le traitement en deux étapes : rappel ou non, puis communication ou non, pour également mesurer l'effet du simple rappel. Ceci n'est pas réalisable car sur l'historique disponible, il y a une tentative de rappel pour environ 95% des prospects.

Le cadre idéal pour mesurer le *True Lift* est l'essai randomisé : une partie aléatoire de la population reçoit le traitement tandis que l'autre partie sert de groupe témoin (aussi appelé groupe de contrôle). Les populations étant homogènes (puisque choisies aléatoirement), on peut alors comparer la variable d'intérêt sur lequel le traitement est censé agir, entre les deux groupes.

En pratique, il est rare de pouvoir mener des essais randomisés. On dispose de données historiques pour lesquelles les groupes traités et témoins n'ont pas été choisis aléatoirement (typiquement selon un modèle élaboré au préalable, contenant d'autres variables). On fait alors face à un biais de sélection dont il faut s'affranchir pour évaluer l'effet de traitement.

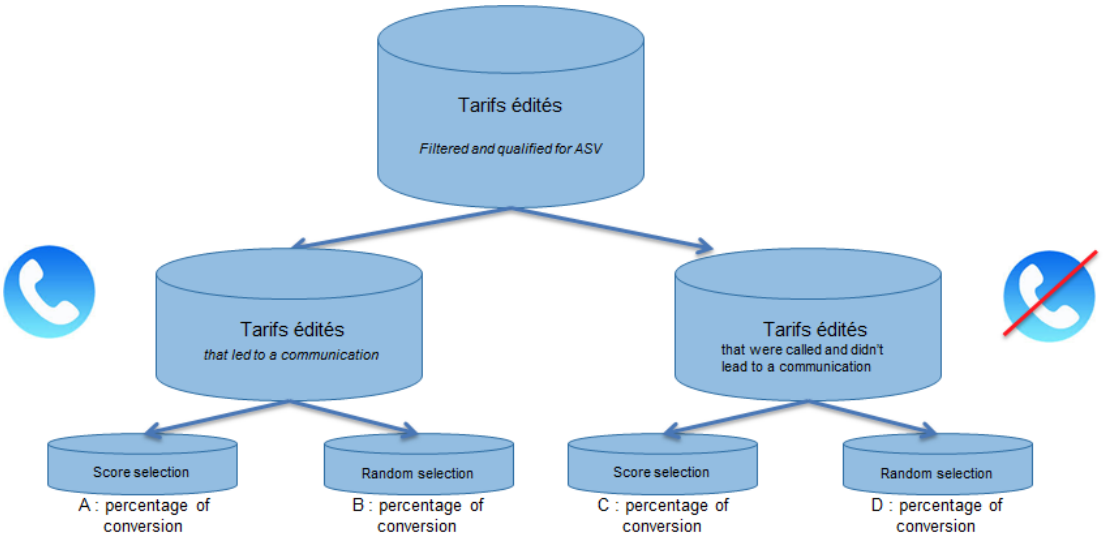


FIGURE 2.7: Modèle d'*uplift*

La figure ci-dessus schématise l'évaluation d'un score de sélection sur l'*uplift*. On sépare la population en un groupe traité et un groupe de contrôle (communication ou non), puis on regarde la différence de conversion parmi les individus sélectionnés par notre score, ici $A - C$. Si cet *uplift* est positif et significatif et qu'il est supérieur l'*uplift* constaté en sélectionnant au hasard des individus, ici $B - D$ (qui estime l'effet moyen du traitement sur la population), alors le score atteint son objectif.

2.5.2 Cadre théorique

Formellement, si on désigne par Y_0 et Y_1 les variables cibles potentielles qui correspondent respectivement à la variable cible sous l'événement $\{T = 0\}$ et sous l'événement $\{T = 1\}$ (heuristiquement, $Y_0 = Y|T = 0$ et $Y_1 = Y|T = 1$), notre variable cible est en fait la différence $TL = Y_1 - Y_0$, appelé *uplift*.

La difficulté dans ce contexte tient au fait que l'on n'observe pour un individu qu'une seule de ces deux variables (un individu reçoit ou non le traitement). La véritable variable cible n'est donc jamais observée. On observe seulement la variable $Y = Y_0 + T * (Y_1 - Y_0)$.

Pour notre étude, la nomenclature des variables est la suivante :

— Y : souscription

— T : communication

— X : données explicatives disponibles (tarifaires, navigation,...)

Bien que pour un individu on ne puisse observer l'*uplift*, il est cependant possible d'estimer la quantité :

$$TL(X) = \mathbb{E}[Y|X, T = 1] - \mathbb{E}[Y|X, T = 0] \tag{2.5}$$

Comme mentionné, pour que cette estimation soit convergente, l'hypothèse centrale est l'indépendance du traitement et de la variable cible conditionnellement à l'ensemble des variables X (CIA, *conditional independance assumption*), qui s'écrit :

$$Y_0, Y_1 \quad \perp \quad T|X \tag{2.6}$$

Lorsque l'hypothèse n'est pas vérifiée (i.e. qu'on ne dispose pas dans X de l'ensemble des variables qui dépendent à la fois de Y_0, Y_1 et de T), on parle de biais de sélection.

2.5.3 Hétérogénéité et biais de sélection

Hétérogénéité et biais historique

Dans notre étude, nous sommes confrontés à la fois à une hétérogénéité dans les données historiques et à un éventuel biais de sélection.

D'abord, l'éligibilité au ASV est déterminée par des règles métier : les devis à date d'effet jour J et supérieure à J+60 et les devis par téléphone ne sont pas éligibles. On doit donc nécessairement se priver de cette sous-population sur laquelle on ne pourra estimer l'*uplift*. Il existe aussi une hétérogénéité historique dans les données puisque les ASV sont déjà priorisés par un score métier existant : le traitement appliqué n'est pas le même selon la file de priorité. Si la rapidité du rappel influe sur la probabilité de souscription, la file la plus prioritaire aura plus de chances d'être rappelée à chaud et donc de souscrire.

Pour contrecarrer ce biais, on peut contrôler par le délai entre le devis et la mise en relation. Pour une première itération du modèle, on regroupera indifféremment toutes les files de priorité et on intégrera dans les variables explicatives ce délai pour mettre en évidence ou non sa significativité. D'après les premières statistiques descriptives, le délai de rappel ne semble pas avoir d'effet fort sur la souscription, ce qui justifie cette première approche.

Hypothèses d'indépendance conditionnelle

Par ailleurs, il est nécessaire dans notre modèle de faire l'hypothèse que, pour le client, le fait de recevoir un ASV n'influence pas l'appétence à la souscription. Rappelons qu'un ASV ne débouche pas forcément sur une communication.

On peut voir le rappel comme une variable binaire R qui vaut 1 pour la quasi-totalité de la population, et dont dépend évidemment la communication ($T = 0$ si $R = 0$). Dès lors, si on considère que des prospects rappelés qui ne répondent pas sont moins susceptibles de souscrire, on fait face à un biais de sélection. Ignorer ce biais est est une hypothèse forte mais justifiée par le fait que les ASV sont passés avec un numéro inconnu par le prospect.

En revanche, ce biais de sélection peut se manifester davantage après que ce dernier a décroché : un prospect qui raccroche immédiatement après avoir été mis en relation avec le serveur vocal préliminaire à la conversation est probablement moins susceptible de souscrire. Or cette variable n'est pas observable au moment du rappel, et biaisera la conversion Y_0 (Y_1 n'est jamais observé pour ces prospects). En incluant ces individus, on aura tendance à sous-estimer Y_0 et donc à surestimer l'effet de traitement.

2.6 Troisième levier : le concept de valeur client

En assurance, la valeur client (NBV, *New Business Value*) permet de mesurer le bénéfice dégagé par une affaire nouvelle en tenant compte de plusieurs paramètres qui dépendent du souscripteur et des paramètres exogènes de marché.

Elle est calculée dans le but d'identifier les segments de marché les plus profitables à une période donnée. Elle sert ainsi à optimiser le *pricing* des polices et la prospection en ciblant ces segments dans les campagnes marketing.

Son calcul consiste essentiellement à déterminer les bénéfices futurs générés par les contrats souscrits à une période donnée, amputés des coûts fixes générés à la souscription.

2.6.1 Méthodologie de calcul

Les composantes du calcul de la NBV sont :

— La prime commerciale (PC) qui représente le montant annuel payé par l'assuré, incluant les frais de chargement de l'assureur. C'est sur cette prime que se répercutent entre autres les cycles de souscription et l'évolution du risque

— La prime pure, qui représente le montant théorique minimal permettant de couvrir le risque de l'assuré. L'évolution de cette prime reflète l'amélioration de la conduite, la tendance de sinistralité et est corrigée par le taux d'inflation.

— Les coûts d'acquisition d'une affaire nouvelle. En assurance directe, ces coûts sont essentiellement reversés aux agrégateurs qui perçoivent une part de la prime en cas d'affaire nouvelle provenant de leur site.

— Les coûts de administratifs et les coûts de gestion de sinistres.

— Les produits financiers, diminués du coût du capital immobilisé rapporté au contrat.

— Le taux de résiliation annuel, qui est lié à la stratégie de *pricing* et à certaines tendances (exemple de la loi Hamon, qui stipule qu'en cas de changement d'assureur le processus de résiliation est à la charge du nouvel assureur).

Les simulations permettant le calcul de la NBV requièrent un certain nombre d'hypothèses. En particulier, les taux de résiliation suivent une dynamique basée sur les trois derniers mois. La dynamique de *pricing* est supposée suivre le marché de l'assurance auto (indice INSEE).
La NBV est calculée de manière individuelle. On peut ensuite les regrouper par les principaux segments (i.e les variables les plus significatives).

FIGURE 2.8: Exemple de calcul de NBV

Year	1	2	3	4	5	6	7	8	9	10
Commercial Premium (index)	100	98	104	103	102	102	101	105	107	106
Pure Premium (index)	60	65	63	62	62	61	60	58	59	59
Loss ratio	85%	75%	70%	70%	65%	60%	63%	60%	55%	60%
Lapse rate	15%	20%	15%	14%	21%	21%	15%	17%	15%	20%
Exposure	85%	90%	95%	90%	80%	90%	85%	95%	90%	90%
Survival rate	100%	90%	70%	65%	65%	50%	45%	40%	30%	25%
Acquisition costs	230									
Management cost	50	55	50	70	80	65		65	65	70
Financial incomes	25	23	30	32	32	28	29	20	25	30
Cost of capital	12%	13%	13%	12%	13%	13%	14%	12%	13%	13%
Discounted cash flows	-108	16	52	38	41	36	35	23	16	23
Cumulative cash flows	-108	-92	-40	-2	39	75	110	133	149	172
NBV	172									

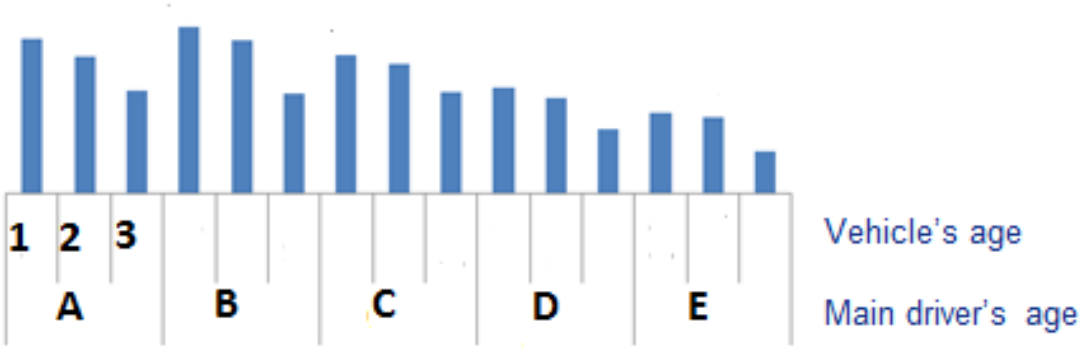


FIGURE 2.9: Segments de NBV

2.7 Problématique actuarielle

La problématique actuarielle sous-jacente à ce projet est double :

— A travers l'optimisation d'un processus marketing, il s'agit d'identifier les variables qui déterminent l'appétence des prospects aux produits d'assurance auto directe et ainsi cibler les profils sur lesquels l'assureur dispose d'un levier d'action. L'intégration de données autres que tarifaires, telles que les informations de navigation internet permettront de mieux comprendre le parcours client. Les données sur les tarifs de la concurrence ajoutent des considérations microéconomiques sur le comportement des prospects.

— La première variable cible des campagnes marketing est le chiffre d'affaire. On cherche alors à acquérir davantage de parts de marché, sans considérer les bénéfices : bien que chaque affaire nouvelle est censée dégager une marge positive, celle-ci peut-être proche de zéro dans des secteurs aussi concurrentiels que l'assurance auto pour les produits de base sans *anciliairies* (garanties complémentaires). L'approche choisie par le projet va plus loin : on souhaite maximiser les bénéfices en considérant la marge potentielle dégagée sur un client. Celle-ci intègre les possibilités de *cross-sell*, d'*upsell*, et le profil de risque du client, à travers un modèle de valeur client.

2.8 Limites liées aux processus en place

Comme nous l'avons évoqué plus haut, le processus existant génère un certain nombre de biais qu'il est parfois possible d'atténuer par certains filtres ou certaines techniques statistiques, mais qui restent difficiles à gérer.

Cette étude permet de toucher du doigt un problème récurrent auquel on fait face lors de la modélisation d'un *uplift*. La solution est ici assez claire : pour se doter de données de qualité adaptées à la modélisation, il faut dans un premier temps effectuer un essai randomisé sur la population lors du rappel pour évaluer sans biais son impact. Pour ensuite permettre une amélioration continue du modèle, une zone de hasard ("zone blanche") doit être conservée dans le processus de rappel, tout en exploitant le modèle sur le reste de la population.

Cette idée de compromis entre exploration et exploitation se retrouve en *active learning*, notamment dans les algorithmes de type "bandits".

Cette méthodologie est évidemment très difficile à mettre en œuvre pour plusieurs raisons, à commencer par les exigences de résultats immédiates qui peuvent souffrir de cette phase exploratoire, et de certaines contraintes techniques du call center.

Chapitre 3

Traitement et exploration des données grâce aux outils *Big Data*

3.1 Contexte sur l'architecture existante

3.1.1 Environnement IT

La mise en œuvre de solutions *Big Data* n'est possible que si l'on dispose d'un environnement adapté. En effet, l'estimation et la calibration des modèles nécessitent d'importantes ressources à la fois pour le stockage des données (mémoire dure) et pour leur traitement (mémoire RAM et processeurs).

A travers AXA Tech, entité gérant l'IT du groupe, le DIL s'est doté de ces ressources, par deux interfaces, partagés par le DIL et d'autres entités disposant de compétences de *Data Science* :

— Trois "super-serveurs", appelés *Big Box* 1, 2 et 3 disposant de multiples cœurs qui permettent le calcul massif.

— Un *cluster* de serveurs utilisant la plateforme Hadoop, administré grâce aux outils Cloudera, qui utilise le calcul massivement parallélisé.

Chaque utilisateur peut se connecter aux *Big Box* et au *cluster* via le protocole SSH, et gérer ses données directement sur ces serveurs.

Ces deux plateformes de travail sont nécessaires : même si le calcul parallélisé se prête naturellement au *Big Data*, certains calculs ne sont par nature pas parallelisables.

3.1.2 Le *datalake*

En tant qu'entité au service de la transformation *Big Data* du groupe, le DIL est amené à travailler avec des données issues d'une multitude d'entités, générées par une multitude de processus différents. Ces données doivent être récupérées, façonnées et fusionnées et pour les besoins spécifiques d'un projet.

Le travail d'extraction des données auprès des entités est généralement long et fastidieux, car il se heurte à des obstacles de sécurité et à l'inertie inhérente au transfert d'information.

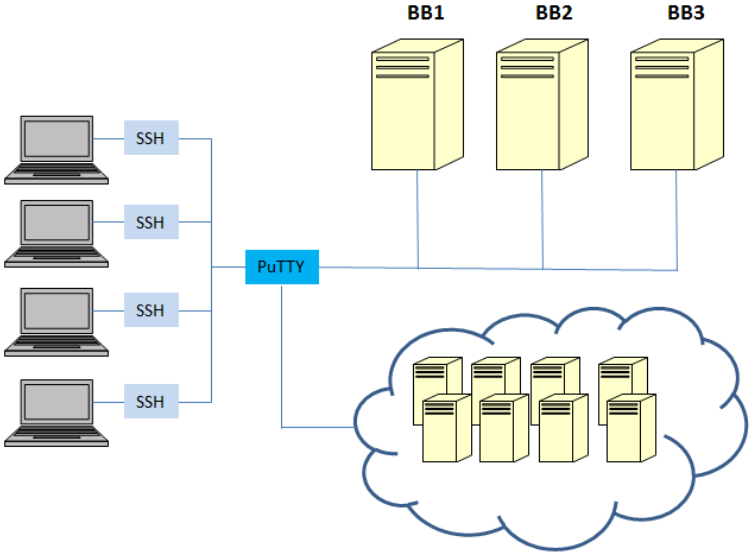


FIGURE 3.1: Schéma simplifié de la plateforme IT

Pour y remédier, le DIL, en collaboration avec les entités du groupe, a contribué à la création d'un *datalake*, un système de stockage de fichiers bruts rassemblant toutes les sources de données accessibles, structurées ou non, de formats très divers (texte, photos, enregistrements audios, ...). Ces fichiers bruts peuvent ainsi être manipulés au gré des projets, afin d'en extraire l'information nécessaire et de la formater de façon adéquate. Ceci est rendu possible par le système de fichiers HDFS (voir plus bas).

Le *datalake* représente un gain majeur en terme d'accessibilité aux données (donc de temps) et de flexibilité dans leur traitement. Il permet d'entamer le travail de formatage et de *feature engineering* très en amont et de ne pas brider le travail du *Data Scientist*.

Chaque projet dispose d'un répertoire commun dans le *datalake*, dans lequel il entrepose et manipule les données nécessaires.

3.2 La plateforme Hadoop et le calcul distribué

3.2.1 Le système de fichier Hadoop

Le système de fichier Hadoop (HDFS) est un système de fichier distribué qui sert à stocker des données volumineuses sur plusieurs disques durs classiques et permettant un accès en *streaming* à ces données. Le système de fichier Hadoop se base sur un *Name Node* qui gère les métadonnées des fichiers (arborescence des répertoires, espace des noms) et plusieurs noeuds de données (*Data Nodes*) qui contiennent des blocs de données. Ces blocs sont répliqués en plusieurs exemplaires sur les différents noeuds, ce qui rend le HDFS très tolérant aux défaillances des noeuds. L'architecture du HDFS a été pensée pour stocker des données sur de très nombreux serveurs peu coûteux qui ont chacun une probabilité non négligeable de défaillance, de sorte qu'un système comportant des milliers de noeuds doit constamment travailler avec des dysfonctionnements de ses éléments. La détection de ces défaillances et la récupération rapide des données sont l'une des forces du HDFS. En résumé, le système HDFS peut se voir comme des ordinateurs s'échangeant de la mémoire,

très peu coûteuse.

L'environnement HDFS est donc particulièrement adapté au *Big Data*, qui traite des données trop volumineuses pour être contenues et manipulées sur une seule machine. Il permet, couplé à l'utilisation de la fonction MapReduce, de passer à l'échelle de données massives les algorithmes parallélisables.

FIGURE 3.2: *Cluster* tournant sur Hadoop

3.2.2 La fonction MapReduce

Lorsqu'on manipule des données volumineuses, les calculs ne peuvent plus être menés de front et requièrent plusieurs machines. La fonction MapReduce est l'un des premiers outils développés pour le calcul distribué. C'est une fonction abstraite qui comporte essentiellement trois étapes :

— **Map** : Des noeuds du *cluster* (ensemble des machines utilisées pour le calcul) sont sollicités pour construire des couples $(clef, valeur)$ à partir des données en entrées et de la requête initiale.

— **Shuffle** : Les noeuds *mappers* répartissent ensuite les couples $(clef, valeur)$ obtenus par $clef$ vers autant de noeuds *reducers* qu'il y a de clés.

— **Reduce** : Ces derniers appliquent en parallèle la fonction *Reduce* pour calculer un résultat associé à leur $clef$

Ce schéma induit une parallélisation des calculs, et s'applique particulièrement aux systèmes de fichiers distribués tels que le HDFS. Dès lors qu'un système est capable de traduire un problème en *MapReduce*, aucune connaissance particulière en calcul distribué n'est requise du programmeur, qui pourra alors utiliser sans problème les ressources d'un *cluster*. Le schéma suivant résume le processus *MapReduce*.

3.2.3 Un exemple type

L'exemple type qui illustre les étapes de la fonction *MapReduce* est le comptage de mots dans un ensemble de documents.

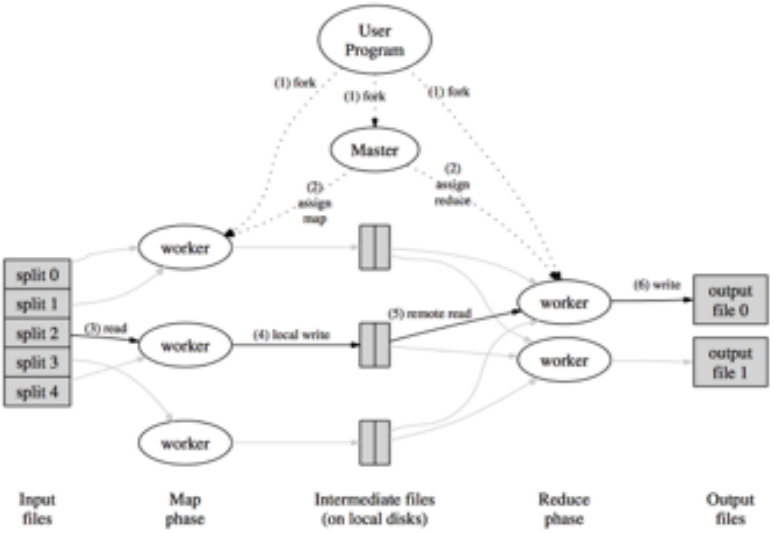


FIGURE 3.3: Schéma de principe du processus *MapReduce*

Chaque document (*splits*) sur le schéma passe par la fonction *map*, qui associe à tous les mots (les clefs) la valeur 1. L'étape de *shuffle* regroupe par clé les couples et les envoie aux noeuds *reducers* qui appliquent la fonction *reduce*, en l'occurrence une simple somme des valeurs, pour obtenir un résultat intermédiaire. Les résultats intermédiaires sont ensuite combinés pour avoir le résultat final.

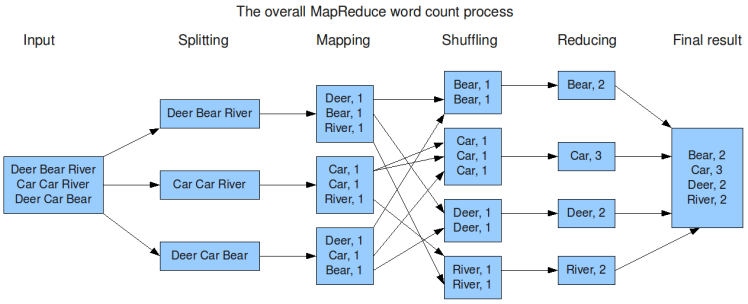


FIGURE 3.4: Exemple du *wordcount*

3.3 Nettoyage et agrégation des données

3.3.1 Le SQL distribué : Hive, Impala

Pour le traitement de données, les bases de données sont généralement structurées en tables relationnelles. Elles peuvent ainsi être interrogées et manipulées via le langage SQL (Structured Query Language), exploitant l'algèbre relationnelle. SQL fonctionne selon le paradigme client/serveur : le client formule une requête, le serveur renvoie le résultat de cette requête.

Le traitement des données massives dépassant les possibilités des tables SQL standards

en terme de capacité, des langages similaires au SQL adaptés au calcul distribué ont été développés. Entre autres, Hive est un langage traduisant les requêtes SQL en tâche MapReduce ou Spark en les partitionnant, selon les versions. La syntaxe y est quasi-identique, bien que certaines fonctionnalités diffèrent.

Basé également sur une syntaxe similaire au SQL, Impala est un outil développé par Cloudera sur le même principe que Hive et utilisant le même type de tables (tables Hive). En revanche, contrairement à Hive, Impala exécute directement les requêtes en distribuant instantanément aux noeuds du cluster et en évitant de traduire les requêtes en MapReduce. Cela lui permet un traitement plus efficace des données. Impala charge les tables en RAM dès qu'il les traite, ce qui accélère en particulier les requêtes impliquant une lecture répétée des mêmes bases de données.

3.3.2 L'outil Thetis

L'une des étapes les plus pénibles mais indispensable à un projet de *Data science* est le nettoyage des données, qui inclut le ré-encodage, le formatage et le *parsing*, la concaténation de colonnes et la suppression de colonnes superflues.

Ce nettoyage doit généralement se faire au cas par cas et prend un temps considérable dès lors que le nombre et les types des variables augmentent et que la qualité des données se dégrade, d'autant plus que la plupart des données sont récupérées sous forme brute (fichiers "plats").

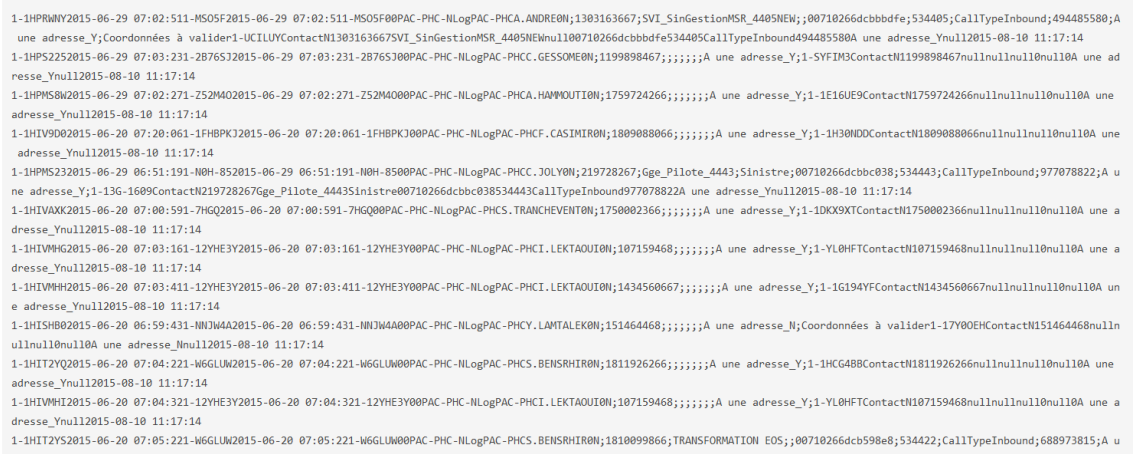


FIGURE 3.5: Exemple de données brutes

C'est pour résoudre ce genre de difficultés communes à tous les projets que l'équipe *engineering* du DIL développe des outils réutilisables. Ces outils s'appliquent tant au nettoyage et au *preprocessing* des données qu'à la création d'interfaces utilisateur et l'amélioration des algorithmes de *machine learning*.

Afin d'éviter le goulot d'étranglement que constitue le nettoyage des données "manuel", un outil de pré-traitement des données, Thétis, a été développé dans le but d'automatiser la création des tables Hive et de fournir des statistiques descriptives sur la qualité de données.

Cet outil permet de réaliser en quelques jours un travail qui pouvait auparavant prendre un mois entier !

Thétis fonctionne en plusieurs étapes :

1. Thétis infère le schéma de la table à partir d'un échantillon réduit des données brutes. Le schéma contient les noms des colonnes, leur format de stockage et d'autres métadonnées.
2. Une table Hive (vide) est ensuite créee à partir du schéma
3. Un nettoyage des données est lancé afin de formater ces dernières au schéma Hive et remplir la table.
4. Thetis parcourt ensuite la table remplie dans son intégralité et fournit un rapport de vérification, en particulier sur les conflits de format entre les données et le schéma inféré. Selon les résultats de ce rapport, on peut retoucher manuellement le schéma et recréer la table.
5. Il est possible de produire des statistiques sur les différentes colonnes de la table (encodage, distributions,...) pour une première exploration des données.

3.4 Données utilisées

Les différentes bases de données utilisables, présentées par source de provenance, sont récapitulées dans la grille ci-dessous.

— **Les données contrats** [otocnt] :
Ces tables répertorient tous les contrats DA ainsi que les devis édités. Dès lors que le devis est édité, il rentre dans les tables contrats sous le statut de devis, amené à changer si celui-ci devient un contrat. Chaque modification apportée à un contrat donne lieu à la création d'une nouvelle version du contrat, de sorte que chaque contrat est répliqué en n versions.

Les tables contrats contiennent essentiellement les informations tarifaires du devis, sur les garanties du contrat, les conducteurs, les véhicules ainsi que les informations personnelles du souscripteur (en particulier le numéro de téléphone). Ces tables constitueront le socle du *dataflow* pour la construction de la table finale.

— **Les données de flux de devis (tarifs vus)** [bfmxml] :
Ces tables enregistrent les tarifs vus en ligne qui ont fait appel au web service de DA, en particulier les tarifs vus sur les comparateurs. Elles se révèlent néanmoins inutilisables en l'état car l'identifiant d'un devis effectué sur un comparateur change au moment de l'édition, ce qui rend le matching difficile. De plus, elles sont sujettes à certains bugs de production qui nécessitent d'être précisément identifiés avant leur utilisation.

— **Les données d'appels téléphoniques** [CTI] :
Ces tables enregistrent les métadonnées des appels téléphoniques (routage, attente, durée, étapes). En plus de repérer les ASV pour lesquels il y a eu communication effective, elles sont utiles pour comptabiliser le temps passé au téléphone par les CEA et éventuellement cibler les profils générant les plus longs temps de communication ou rejetant systématiquement les relances.

— **Les tables de flux ViaDialog** : Cette table est un journal de flux permettant de récupérer les informations relatives aux ASV : heure d'entrée dans le processus, files de priorité, nombres de relances, aboutissement ou non des relances. Elles sont complétées, si une communication a lieu, par les tables CTI.

— **Les données socio démographiques** : En utilisant le code INSEE ou la maille IRIS, ces données utilisées pour la tarification peuvent compléter les tables contrats.

— **Les données de navigation** [navig] :
Ces données sont récoltées par un prestataire via les cookies du site internet DA. Elles apportent des informations supplémentaires concernant le parcours client.

— **Les données comparateurs** [comp] :
Ces données fournies par un comparateur en ligne apportent des informations qui peuvent potentiellement avoir un fort pouvoir prédictif : y figurent le rang de DA parmi les devis, le tarif du concurrent immédiatement au-dessus et du meilleur concurrent.

Les données internes sont généralement très accessibles et utilisables en temps réel.

En revanche, les données de navigation ne sont récupérées que sont forme d'extraits à des intervalles de temps espacés. De même, les données de la concurrence ne proviennent que d'un seul agrégateur sous forme d'extraits journaliers.

En ce qui concerne ces données dont l'utilisation est relativement nouvelle, un des objectifs contingents au projet est de mesurer la valeur ajoutée qu'elles contiennent, afin d'évaluer les possibilités d'extraction en temps réel pour le développement futur du projet.

3.5 *Dataflow*

3.5.1 Création d'une table finale

Une fois les différentes tables mise à disposition et nettoyées, il s'agit de créer une table finale qui agrège toutes les variables d'intérêt contenues et en crée éventuellement de nouvelles. Le fil conducteur dans la construction de cette table est de conserver une ligne par contrat/devis. Etant donné que de nombreuses tables présentent des correspondances multiples (plusieurs lignes associées à un même identifiant), certaines colonnes devront être agrégées pour en tirer une information unique par identifiant. Cette construction nécessite un long travail de récupération et de compréhension des tables. Le processus de construction de la table finale, appelé *dataflow* est schématisé ci-dessous. On note respectivement les correspondances uniques et multiples 1 :1 et 1 :N, et les correspondances incomplètes 1 :0.

3.5.2 L'outil Dataiku

Pour mettre en place le *Dataflow* envisagé et mieux visualiser le parcours des données jusqu'à la table finale, le logiciel Dataiku a été utilisé. Dataiku est une interface visuelle destinée à la création de modèles de *Data Science* qui possède de nombreuses fonctionnalités intéressantes. Il permet de schématiser le *Dataflow*, d'explorer rapidement des échantillons de tables et d'analyser d'un rapide "coup d'oeil" certaines caractéristiques des colonnes

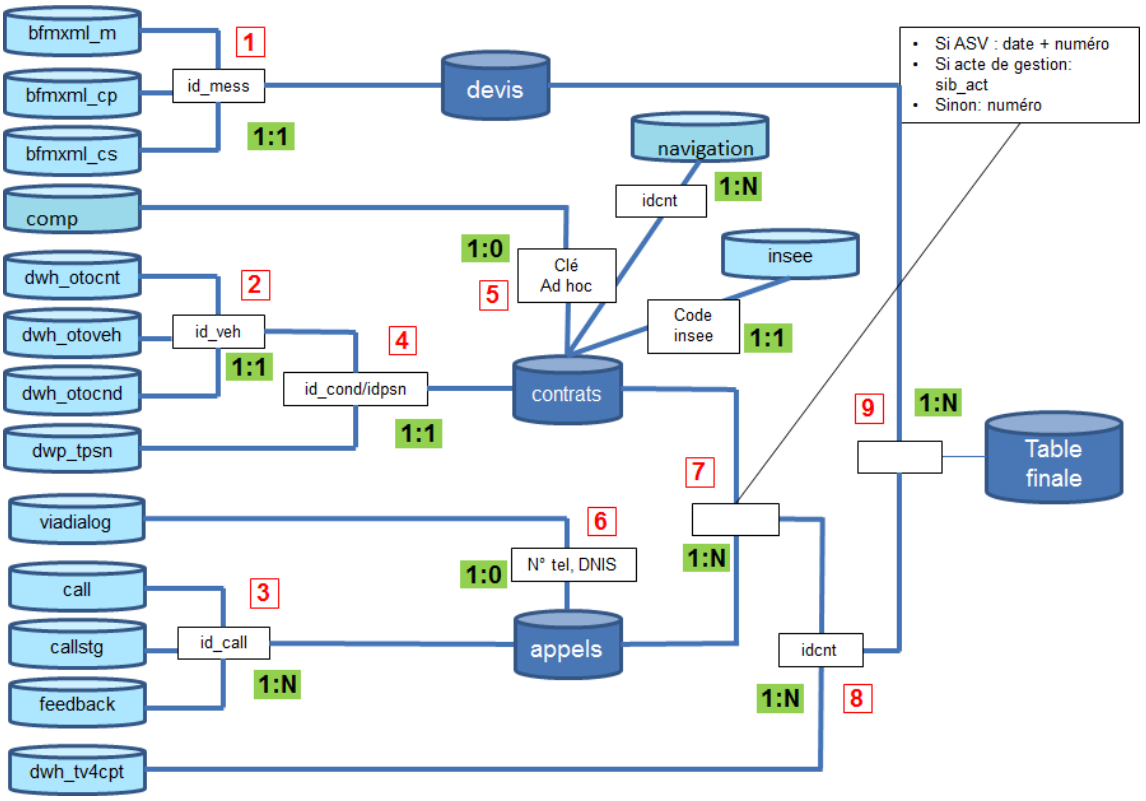


FIGURE 3.6: *Dataflow*

(valeurs manquantes, distributions) et d'implémenter facilement certaines transformations des données qui facilitent le *feature engineering*. Des outils de modélisation sont également disponibles bien que leur usage ne soit pas adapté à la mise en production (on choisira plutôt un *framework* séparé pour la modélisation, voir plus bas).

L'avantage de Dataiku est sa capacité à concilier les environnements distribués et non distribués dans un projet collaboratif.

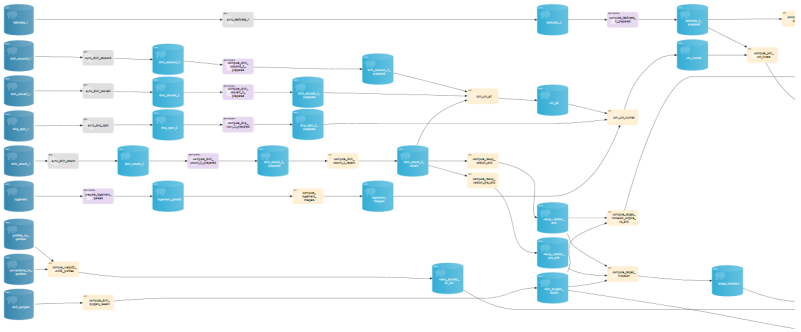


FIGURE 3.7: L'interface Dataiku

3.5.3 *Data Quality assessment*

Les jointures successives ont permis d'identifier certaines difficultés sur les différentes clés. Le *Data Quality Assessment* résume la qualité des données et le niveau de réussite de chaque jointure et propose des recommandations pour remédier à ces difficultés. Ces recommandations aux entités sont en général destinées à être mises en place sur le long terme, et font partie des missions du DIL. Si certaines d'entre elles avaient été mises en place, elle auraient permis un gain de temps énorme dans la construction du *dataflow*.

— Les tables des tarifs vus (*bfmxml*) n'ont pas été utilisées en première approche : sujettes à des bugs de production, elles ne peuvent pas être directement reliées aux devis édités des tables contrats car les identifiants des devis provenant des comparateurs changent une fois le devis édité.

— Certains champs des tables contrats ne sont totalement fiables, en particulier celui de l'origine du devis qui est une variable essentielle de notre étude. Ceci est dû à un versionnage incomplet des contrats (certaines informations sont écrasées suite à des modifications). Il est alors nécessaire de recourir à d'autres tables pour compléter et corriger ces champs. Un des champs essentiel affecté par ce problème est l'origine du devis, pour lequel un *patch* a été implémenté pour retrouver cette origine, mais dont certaines valeurs seront écartées de la modélisation car source de biais (l'écrasement du champ indique potentiellement une information liée à la conversion, ce qui est confirmé par les premières statistiques descriptives présentées plus bas).

— Les jointures avec les appels entrants des tables CTI sur les numéros de téléphone sont polluées par les faux numéros renseignés en ligne. Il est nécessaire de filtrer ces numéros pour éviter les doublons.

— Une des jointures particulièrement délicate est celles des données comparateurs aux tables contrats. En effet, on ne dispose ni d'un identifiant commun ni de l'heure exacte du devis. L'exercice est d'autant plus difficile que la table comparateur contient également les tarifs vus. Cette jointure est donc réalisée uniquement pour les devis provenant de ce comparateur qui correspondent de façon unique à une clé de jointure *ad hoc*.

Les données de navigation, disponibles sous forme d'*extracts*, ont nécessité un travail de compréhension approfondi en plusieurs étapes :

— Création d'un dictionnaire pour comprendre la signification de chaque colonne

— Déconcaténation de certaines colonnes pour extraire des champs ayant une signification.

— Recherche des clés de jointures non disponibles directement (en particulier l'identifiant d'un devis) dans certains champs (URL).

Les tables de navigation comportent environ 500 colonnes. Avant de les joindre, un travail d'exploration de ces données sera mené en parallèle pour en extraire l'information essentielle à travers des *features* agrégées. On peut ajouter, d'un point de vue pratique, que la structure en flux de la table ViaDialog ne se prête pas facilement au SQL. De nombreuses requêtes intermédiaires sont nécessaires pour obtenir une ligne par contrat.

Le *Data Quality Assessment* est avant tout un moyen de remonter les difficultés de récupération des variables et d'agrégation des tables au métier, afin que les processus de stockage des données puissent évoluer dans le sens des besoins des projets. C'est grâce à ces recommandations que les systèmes d'information pourront être améliorés.

3.6 Implémentations possibles du projet

A ce stade, la mise en production n'est pas véritablement instruite, car le projet en est à sa phase exploratoire/POC. Cependant des solutions peuvent d'ores et déjà être envisagées afin de se préparer au déploiement futur, mais aussi pour orienter certaines approches de modélisation concernant les variables à utiliser.

En effet, ce projet comporte une dimension "temps réel" : on souhaite dans l'idéal être en mesure de produire un score mis à jour régulièrement, car les données arrivent sous forme de flux et comportent une dimension temporelle. Néanmoins, à l'heure actuelle, certaines données ne sont pas disponibles en temps réel voire nécessitent une extraction sur des intervalles de temps très espacés (données de navigation par exemple).

Un premier modèle pourra être construit à partir des données facilement intégrables

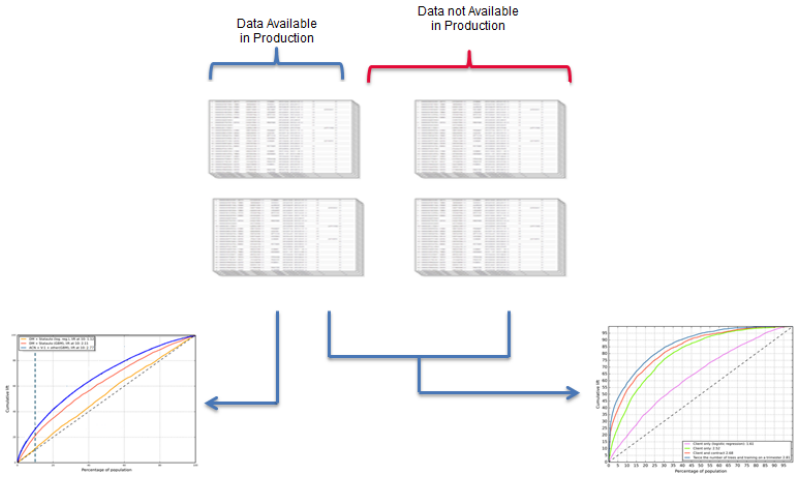


FIGURE 3.8: Scnenarii de production

aux outils du métier (ViaDialog), pour évaluer si ce modèle est effectivement profitable.

Un second modèle, ajoutant les données sur lesquelles on ne dispose que d'échantillons extraits et qui ne sont pas pour l'instant disponibles en production, permettra de mesurer le gain par rapport au premier modèle, et vérifier si ce gain est suffisant pour les acquérir en temps réel.

3.7 Outils de modélisation

Après avoir agrégé les données et construit des *features* d'intérêt en Hive dans une table finale, le choix a été fait de programmer la partie modélisation en Python, langage de programmation interprété et orienté objet qui possède une bibliothèque dédiée au *machine*

learning, *scikit-learn*. Cette librairie est une référence parmi la communauté des *data scientists* et implémente tous les algorithmes couramment utilisés.

R est un autre langage offrant de nombreuses possibilités dans le domaine du *machine learning*. Plus "obscur" que Python dans sa conception (le langage n'est pas réellement orienté objet), il est moins flexible car ses fonctions sont plus difficiles d'accès, contrairement aux classes *scikit-learn* qui peuvent être remodelées suivant les besoins. Il se révèle néanmoins être un bon complément à Python car il dispose de plusieurs bibliothèques utiles que *scikit-learn* ne propose pas.

3.8 Schéma du *Workflow*

A travers les paragraphes précédents, on a résumé l'architecture du projet (le *workflow*) qui se décompose en trois grande phases : nettoyage, *preprocessing* et *machine learning*.

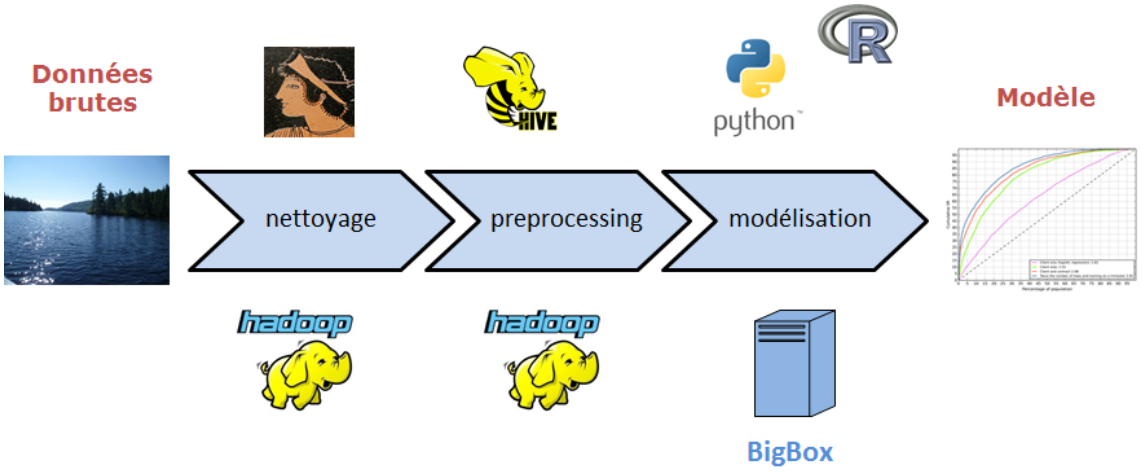


FIGURE 3.9: Schema du *workflow*

Chapitre 4

Outils de *machine learning* utilisés

L'estimation des modèles développés dans le chapitre suivant fait appel à l'apprentissage automatique (*machine learning*). L'apprentissage automatique développe des algorithmes à but essentiellement prédictif, et diffère des méthodes statistiques classiques par sa capacité à exécuter des algorithmes gourmands en terme de ressources de calcul.

4.1 *Preprocessing* des données

La table finale utilisée pour la modélisation nécessite un pré-traitement avant d'être 'ingérée' par les algorithmes.

4.1.1 Traitement des valeurs manquantes

Un problème quasi-systématique dans tout jeu de données est la présence de valeurs manquantes (champs vides pour certaines observations). La solution naïve consistant à retirer du modèle les observations ou les colonnes comportant des valeurs manquantes est trop coûteuse en terme d'information et peut engendrer des biais suivant le mécanisme d'apparition des valeurs manquantes, surtout lorsque ces mécanismes sont structurels (un champ vide contient en fait de l'information). Il existe toute une littérature sur le traitement des données manquantes. Pour ne pas complexifier les premiers lancements de modèles, on peut traiter les valeurs manquantes de plusieurs manières simples, selon le type de la variable (catégorielle ou numérique) :

— Pour les variables numériques, on peut les remplacer par la moyenne, la médiane, ou par un tirage aléatoire dans la distribution empirique (discrète).

— Pour les variables catégorielles, on peut les remplacer par le mode, la médiane (s'il existe une relation d'ordre) ou par un tirage aléatoire dans la distribution empirique (multinomiale).

Un traitement plus fin consisterait à faire de la "prédiction inversée" des variables manquantes par les autres variables (procédure d'imputation multiple). Nécessitant plus de temps de calcul, cette méthode ne sera pas développée en première approche.

Le traitement à appliquer dépend également du modèle utilisé : par exemple, dans les modèles d'arbres, remplacer les variables catégorielles par leur médiane conduirait à mettre

toutes ces variables dans la même feuille au moment du *split* (voir plus bas pour une explication détaillée des méthodes d'arbres).

4.1.2 Traitement des variables catégorielles

Certains algorithmes utilisés ne prennent en entrée que des données numériques, or de nombreuses variables d'intérêt sont des variables catégorielles non-numériques. D'autres algorithmes, en particulier les méthodes d'arbres, peuvent prendre en charge ces variables catégorielles pourvu que le nombre de modalités reste suffisamment faible. En effet, comme décrit plus bas, les arbres opèrent des partitions en deux ensembles des données suivant la variable. Une partition suivant une variable catégorielle à K modalités revient à étudier toutes les combinaisons possibles, soient $\sum_{i=1}^{K} \binom{K}{i} = 2^K$. La bibliothèque *scikit-learn* ne gère malheureusement pas les variables catégorielles.

Pour les intégrer aux algorithmes, on peut leur appliquer plusieurs transformations, décrites ci-dessous.

Encodage numérique

L'encodage numérique consiste à remplacer les K modalités de la variables par autant d'entiers $1, \ldots, K$. L'inconvénient majeur de ce type d'encodage est qu'une relation d'ordre arbitraire est créée. Cela pose un problème pour les modèles linéaires, car l'écart est le même entre les niveaux de la variable, qui n'est pas forcément souhaitable. Pour les modèles d'arbres, l'ordre va contraindre le partitionnement et empêchera l'exploration de l'ensemble des *splits*.

Encodage binaire

Le principe de l'encodage binaire des modalités de la variable (*dummyfication*) est le suivant : si la variable possède K modalités distinctes $C_1, \ldots, C_K$, on lui substitue $K - 1$ variables indicatrices $\mathbb{1}_{i \in C_1}, \ldots, \mathbb{1}_{i \in C_K}$. Lorsque le modèle comporte une constante, il est nécessaire de prendre une catégorie de référence pour ne pas rendre les variables colinéaires et donc se contenter de $K - 1$ indicatrices.

Cet encodage est particulièrement adapté aux modèles linéaires généralisés (GLM), même s'il présente l'inconvénient de gonfler l'ensemble des variables d'autant de modalités de la variable initiale.

En revanche, il se prête moins au méthodes d'arbres (voir plus bas), très utilisées en *machine learning* qui se basent sur un partitionnement récursif de l'espace des *features*, lorsque les variables présentent un grand nombre de modalités.

Prenons l'exemple suivant : une variable à $K = 10$ modalités équiprobables est encodée par 10 variables binaires. Les modalités 1 à 5 contiennent 90% de 1, tandis que les modalités 6 à 10 contiennent 90% de 0. Un partitionnement sur la variable intégrale conduirait à une séparation binaire optimale entre les modalités 1 à 5 et 6 à 10 avec une nette amélioration de la pureté (I_1). En revanche, si on partitionne suivant les variables binaires $\mathbb{1}_{i \in C_1}, \ldots, \mathbb{1}_{i \in C_{10}}$, on constate qu'aucune séparation n'améliore véritablement la pureté, du

fait du trop faible effectif par modalité (I_2) :

$$\begin{aligned} I_1 &= 0.5 * (0.9 * 0.1) + 0.5 * (0.9 * 0.1) \\ &= 0.09 \\ I_2 &= 0.1 * (0.9 * 0.1) + 0.9 * (\frac{0.9 * 5 + 0.1 * 4}{9}) * (1 - \frac{0.9 * 5 + 0.1 * 4}{9}) \\ &= 0.26 \end{aligned}$$

Encodage par impact

L'encodage par impact sur la variable cible (*impact coding*) est très utile dans le cas de modèles d'arbres, pour lesquels l'encodage binaire disperse trop l'information pour mener à des *splits* efficaces sur les variables binaires. Il ne s'applique qu'à la classification binaire. Dans ce cas, on peut encoder la variable par la proportion de 1 contenue dans la modalité, ce qui permet d'effectuer la séparation de façon optimale grâce à l'ordre crée par l'encodage. On peut ainsi montrer que la recherche parmi les 2^K possibilités se limite à K possibilités. Intuitivement, ce codage permet de séparer d'un côté les catégories possédant la plus grande proportion de 1, ce qui mène au partitionnement optimal.

$$X = C_k \Longleftrightarrow \tilde{X} = \frac{\#\{X = C_k, Y = 1\}}{\#\{X = C_k\}}$$

Pour être rigoureusement équivalent à l'exploration de tous les splits catégoriels à chaque étape de l'arbre, cette opération doit être répétée après chaque *split* dans les nouvelles feuilles de l'arbre (Louppe et al., 2013).

Catégories résiduelles

Lorsque le nombre de modalités est trop élevée (typiquement des villes, des codes postaux,...), certaines occurrences deviennent trop faibles et ne sont plus significatives. Pour contourner ce problème, on regroupe les modalités à trop faible occurrence en une modalité résiduelle. Le seuil d'occurrence est un paramètre à déterminer.

4.1.3 Elaboration d'un *pipeline* de transformations

Afin d'automatiser le pré-traitement des variables, on construit un *pipeline* qui appliquera successivement ces transformations, de manière suffisamment générale pour être réutilisé dans le cadre d'autres projets. Une encapsulation du code sur le modèle de *scikit-learn* permet davantage de flexibilité. Le *pipeline* présente également l'avantage d'être répété intégralement à chaque *Grid search* (voir plus bas), ce qui se révèle utile pour les transformations de type *impact coding* pour lesquels les échantillons d'entraînement et de test doivent être traités séparément pour éviter le surapprentissage.

4.2 *Preprocessing* des données de navigation

Les données de navigation arrivent de façon brute, et sans documentation, ce qui a nécessité un long travail de traduction des *headers*.

Les colonnes de ces tables ne sont pas exploitables telles quelles : la plupart d'entre elles sont textuelles, et doivent être déconcaténées et analysées (opérations de *parsing*). *Scikit-learn*

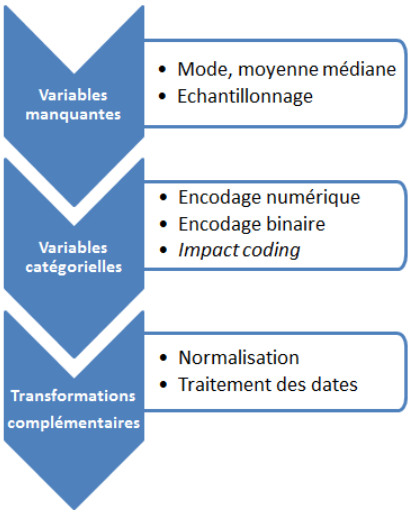


FIGURE 4.1: *Pipeline* des transformations

dispose de fonctionnalités intéressantes pour l'analyse textuelle. Le travail d'extraction de *features* d'intérêt se fait en plusieurs étapes :

— L'éclatement des chaînes de caractères (description, adresses URL) en plusieurs éléments, appelés *tokens*. Ceux-ci peuvent être déterminés par un ou plusieurs séparateurs. On doit éliminer les motifs uniques, typiquement des numéros de devis, de session qui sont propres à la ligne.

— On crée une nouvelle matrice de *features* textuelles avec en colonnes le "dictionnaire" des *tokens*, remplis par les occurrences de ceux-ci dans les lignes : c'est la représentation en "sac de mots".

— Une sélection des *tokens* pertinents est faite par *impact coding*

Les features textuelles ainsi obtenues sont ajoutées. Pour avoir une idée de la valeur ajoutée qu'elles peuvent apporter aux modèles, on réalise une prédiction de la variable cible avec des algorithme de classification binaire. Bien que l'importance des variables textuelles soit moindre par rapport aux principales variables numériques tirées de ces tables (nombre de visites, ...), elles apportent une information supplémentaire et améliorent la prédiction.

4.3 Principe du *machine learning*

L'estimation de modèles par des algorithmes de *machine learning* se fait généralement en trois étapes :

— **Entraînement** : On sélectionne un échantillon d'observations appelé échantillon d'entraînement, sur lesquels on applique l'algorithme qui produira une fonction de prédiction (classificateur, régresseur).

— **Test** : On teste la fonction de prédiction sur un échantillon de test, qui n'a pas été utilisé pour l'entraînement. On simule ainsi l'acquisition de nouvelles données sur lesquelles on souhaite faire une prédiction. On peut alors mesurer l'erreur de prédiction commise sur la variable cible, et évaluer la qualité d'un modèle. En changeant les paramètres de l'algorithme, on cherche à ajuster le meilleur modèle.

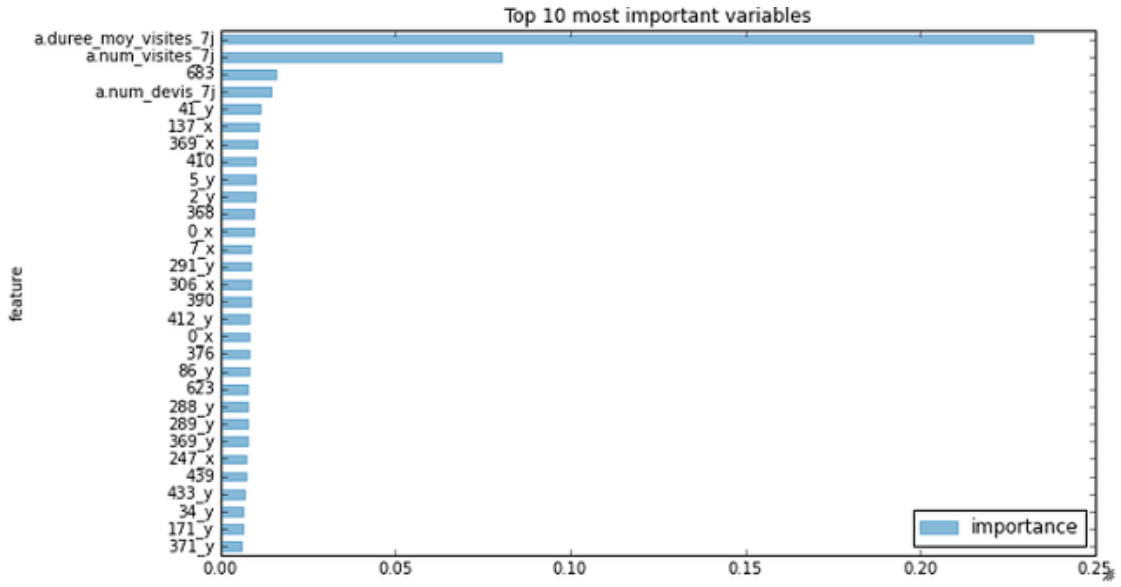


FIGURE 4.2: Importance relative des variables textuelles

— **Validation** : On sélectionne le meilleur modèle à l'issue de la phase de test en le testant sur un échantillon de validation.

4.3.1 Arbitrage Biais-Variance et surapprentissage

Une fois un modèle prédictif entraîné, l'erreur de prédiction sur des échantillon de test peut se décomposer en deux termes de biais et variance.

— Le biais correspond à l'erreur moyenne commise sur plusieurs échantillons i.i.d des données (généralisation de la prédiction). Si l'on a pas intégré suffisamment d'information à notre modèle, ce biais sera élevé. On parle alors de sous-apprentissage

— La variance de l'estimation correspond à la répercussion dans l'estimation du modèle de la variabilité de l'échantillon d'entraînement : elle est élevée si une légère fluctuation dans l'échantillon d'entraînement entraîne une grande fluctuation dans la prédiction. Ceci se produit lorsqu'on intègre trop d'information au modèle. On parle alors de surapprentissage.

4.3.2 Validation croisée

La validation croisée en k-folds consiste à subdiviser les données en k échantillons de tailles égales, puis à répéter la procédure entraînement/test en prenant successivement chacun des k échantillons comme échantillon de test. Les performances à chaque itération sont ensuite moyennées pour obtenir la performance finale du modèle. Il s'agit d'une sorte de *boostrap* sur les échantillons de test et d'entraînement.

La validation croisée dépend du nombres k de plis que l'on se donne. La version extrême à N plis est appelée *leave one out* (LOOCV).

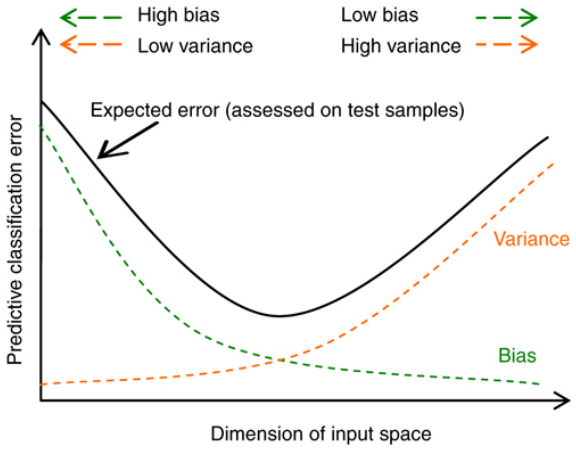


FIGURE 4.3: Arbitrage biais-variance

4.3.3 *Grid Search*

La dernière étape de la modélisation est l'optimisation des hyperparamètres du modèle.

Les hyperparamètres sont toutes les variables de calibration d'un modèle qui sont fixées arbitrairement en première approche. Pour les algorithmes classiques évoqués, voici quelque uns de ces paramètres :

— Régression logistique : la norme de pénalité à appliquer, les paramètres de l'algorithme d'optimisation de la vraisemblance, le type d'algorithme, le paramètre de régularisation...

— SVM : le paramètre de pénalité, la fonction noyau,...

— Forêts aléatoires : le nombre d'arbres, leur profondeur maximale, le nombre de variables à tirer avant chaque *split*, le critère d'impureté,...

Le terme *grid search* fait référence à une recherche brute sur l'ensemble des valeurs possibles pour les hyperparamètres spécifiés. Le nombre de combinaisons à tester augmente exponentiellement avec le nombre de paramètres choisis. Comme chaque combinaison implique une nouvelle validation croisée, on se borne à un petit nombre de paramètres pour limiter le temps de calcul.

D'autres méthodes existent pour l'optimisation des hyperparamètres, des *random searches* qui ne calculent pas chaque combinaison, mais procèdent à une exploration plus intelligente de l'espace des hyperparamètres (par des méthodes bayésiennes par exemple). C'est le cas du package *hyperopt* de Python.

4.4 Modélisation de la mise en relation

La cible est ici la communication qui est une variable binaire.

Avec les notations précédentes, on recherche un classificateur binaire g :

$$\mathbb{P}(T = 1|X) = g(X) \tag{4.1}$$

Pour rappel, ce score constitue la première "brique" de notre KPI, et sera couplé au score d'*uplift* et à la valeur client pour obtenir le score final.

Il existe de nombreux algorithmes de classification binaire, dont les plus importants sont expliqués ci-après.

4.5 Revue des algorithmes utilisés

4.5.1 Régression logistique

Soient $(x_i)_{1\leq i\leq N} \in \mathbb{R}^p$ et $(y_i) \in \{0,1\}$ une variable cible binaire. L'hypothèse du modèle de régression logistique suppose que $Y|X$ suit une distribution de Bernoulli de paramètre $\theta = \sigma(w^T x)$:

$$\mathbb{P}(Y=1|X=x) = \sigma(w^T x) = \frac{1}{1+\exp(-w^T x)} = \frac{1}{1+\exp(-\sum_{i=1}^{p} w_i x_i)}$$

où σ est la fonction sigmoïde définie par $\sigma(z) = \frac{1}{1+\exp(-z)}, \forall z \in \mathbb{R}$ et $(w_i)_{1\leq i\leq p}$ sont des coefficients à déterminer. Il est possible d'ajouter une constante $w^T x + w_0$.

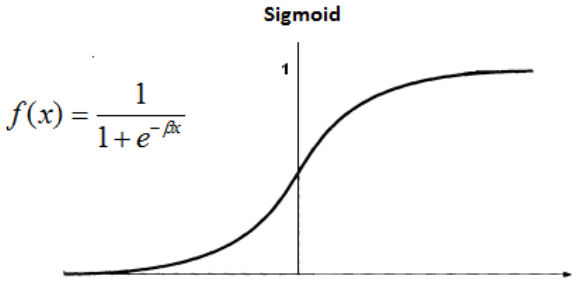


FIGURE 4.4: fonction sigmoïde

Les coefficients de la régression logistique s'interprètent en termes de *log odds ratios*, en écrivant

$$\log(\frac{\mathbb{P}(Y=1|X=x)}{\mathbb{P}(Y=0|X=x)}) = w^T x + w_0$$

L'objectif de la régression logistique est de modéliser les probabilités conditionnelles par des fonctions linéaires de X, tout en s'assurant que ces probabilités restent comprises entre 0 et 1.
Les (w_i) sont estimés par maximum de vraisemblance. La log vraisemblance s'écrit :

$$\begin{aligned} l(w) &= \sum_{i=1}^{N} \log(\mathbb{P}(Y=y_i|X=x_i)) \\ &= \sum_{i=1}^{N} \log(\theta^{y_i}(1-\theta)^{1-y_i}) \\ &= \sum_{i=1}^{N} y_i \log(\sigma(w^T x_i)) + (1-y_i)\log(1-\sigma(w^T x_i)) \end{aligned}$$

C'est une fonction convexe de w, on annule donc son gradient pour la maximiser :

$$\nabla_w l(w) = \sum_{i=1}^{N} x_i(y_i - \sigma(w^T x_i) = 0$$

Cette équation n'étant pas linéaire, elle doit être résolue par des méthodes d'optimisation itératives telles que la descente de gradient ou l'algorithme de Newton-Raphson.

4.5.2 Support Vector Machine

On se donne des observations (x_i, y_i) pour $i = 1, \ldots, N$, avec $x_i \in \mathbb{R}^p$ et $y_i \in \{-1, 1\}$. On cherche à entraîner un classificateur binaire $f(x) = w^T x + b$ tel que :

$$f(x_i) = \begin{cases} \geq 0 & \text{si } y_i = 1 \\ < 0 & \text{si } y_i = -1 \end{cases} \tag{4.2}$$

i.e $y_i f(x_i) \geq 0$ lorsque la classification est correcte.
La machine à support de vecteurs (SVM) cherche l'hyperplan optimal $f(x) = 0$ qui sépare les deux classes et maximise la distance au point le plus proche de chacune des classes. Ceci garantit l'existence d'une unique solution et améliore la performance sur l'échantillon de test.
Soit $d_+(d_-)$ les distances les plus courtes de l'hyperplan séparateur aux observations respectivement positives et négatives. On définit la marge de l'hyperplan par $M = d_+ + d_-$. Supposons que les données sont linéairement séparables et que l'échantillon d'entraînement vérifie :

$$w^T x_i + b \geq 1 \text{ si } y_i = 1$$
$$w^T x_i + b \leq -1 \text{ si } y_i = -1$$

En supposant que ces égalités sont vérifiées pour certains points, $d_+(d_-)$ est alors la distance entre l'hyperplan $H_1 : w^T x + b = 1$ ($H_{-1} : w^T x + b = -1$) et l'hyperplan séparateur $H : w^T x + b = 0$. Donc, $d_+ = d_- = 1/|||w|||$, et la marge est $2/|||w|||$. L'hyperplan séparateur est alors obtenu en minimisant $||w||^2$ sous la contrainte $y_i(w^T x_i + b) \geq 1, \forall i$:

$$\min_w ||w||^2 \text{ sous contrainte } y_i(w^T x_i + b) \geq 1, \forall i$$

Soft margin Supposons que les données ne sont pas linéairement séparables, i.e. que les classes se superposent. Pour contourner ce problème, on maximise M mais en autorisant certains points à se situer du mauvais côté de l'hyperplan. On définit des variables 'lâches' $\zeta = (\zeta_1, \zeta_2, \ldots, \zeta_n)$ et on modifie les contraintes de la façon suivante :

$$y_i(w^T x_i + b) \geq 1 - \zeta_i$$
$$\zeta_i \geq 0$$
$$\sum_{i=1}^{n} \zeta_i \leq \text{ constante}$$

L'erreur de classification a lieu lorsque $\zeta_i > \frac{1}{||w||}$. On borne $\sum_{i=1}^{n} \zeta_i$, l'erreur de classification totale autorisée. On peut traduire cette dernière contrainte par les multiplicateurs de Lagrange. Le problème devient alors :

$$\min_{w,\zeta} \frac{1}{2}||w||^2 + C\sum_{i=1}^{n} \zeta_i$$

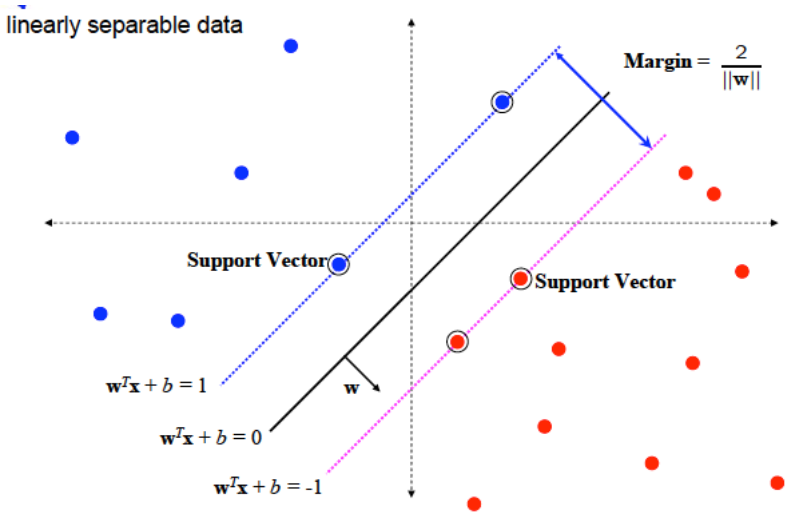


FIGURE 4.5: Support vector machine

$$\text{sous contrainte } y_i(w^T x_i + b) \geq 1 - \zeta_i \text{ et } \zeta_i \geq 0$$

En écrivant la formulation duale du problème via le Lagrangien, on constate que les données d'entraînement n'apparaissent que dans des produits scalaires entre vecteurs, ce qui permet de généraliser la procédure au cas non-linéaire. Le Langragien est :

$$L(w, \zeta, \alpha, \beta) = \frac{1}{2}||w||^2 + C\sum_{i=1}^{n}\zeta_i - \sum_{i=1}^{n}\alpha_i(y_i(w^T x_i + b) - 1 + \zeta_i) - \sum_{i=1}^{n}\beta_i\zeta_i$$

Le problème dual s'écrit : $\max_{\alpha,\beta} \min_{w,\zeta} L(w, \zeta, \alpha, \beta)$:

$$\max_{\alpha} \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i,j}^{n}\alpha_i\alpha_j y_i y_j x_i^T x_j$$

$$\text{sous contrainte } \sum_{i=1}^{n}\alpha_i y_i = 0 \text{ et } 0 \leq \alpha_i \leq C$$

et la solution est :

$$f(x) = w^T x + b = \sum_{i=1}^{n}\alpha_i y_i x_i^T x + b$$

Kernel tricks La machine à support de vecteurs est jusque là un classificateur linéaire, que l'on peut adapter au données non linéairement séparables par la *soft margin*. Une autre façon d'aborder ce problème est de plonger les vecteurs de données dans des espaces de plus grande dimension (éventuellement infinie) afin de les rendre linéairement séparables dans cet espace. On intègre simplement à la formulation précédente du SVM une fonction de *mapping* en remplaçant x_i par $\phi(x_i)$.

$$\min_{w,\zeta} \frac{1}{2}||w||^2 + C\sum_{i=1}^{n}\zeta_i$$

$$\text{sous contrainte } y_i(< w, \phi(x_i) > + b) \geq 1 - \zeta_i \text{ et } \zeta_i \geq 0$$

Le problème dual devient :

$$\max_{\alpha} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j}^{n} \alpha_i \alpha_j y_i y_j < \phi(x_i), \phi(x_j) >$$

$$\text{sous contrainte } \sum_{i=1}^{n} \alpha_i y_i = 0 \text{ et } 0 \leq \alpha_i \leq C$$

et la solution est :

$$f(x) = \sum_{i=1}^{n} \alpha_i y_i < \phi(x_i), \phi(x) > + b$$

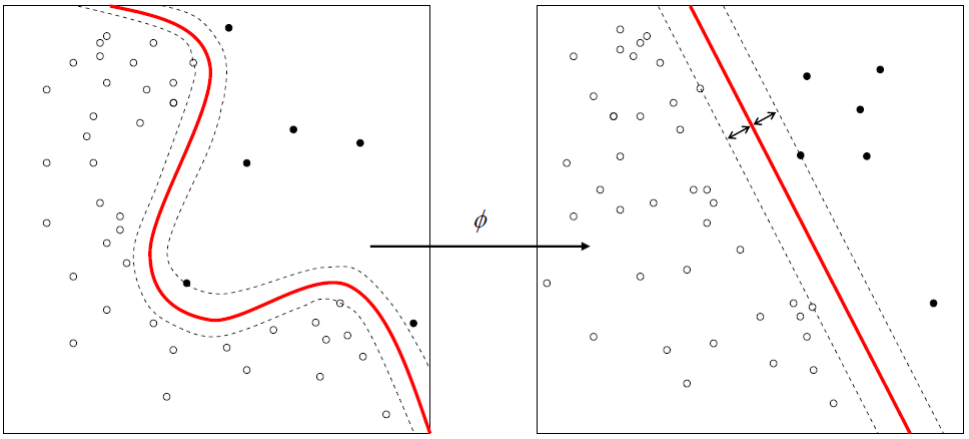


FIGURE 4.6: Kernel machine

On remarque que la formulation duale du problème ainsi que sa solution ne font intervenir $\phi(x)$ que par des produits scalaires. La fonction de *mapping* n'a pas besoin d'être explicitement spécifiée, on a seulement besoin de connaître la fonction noyau qui représente le produit scalaire dans le nouvel espace :

$$K(x, y) =< \phi(x), \phi(y) >$$

La solution devient : $f(x) = \sum_{i=1}^{n} \alpha_i y_i K(x_i, x) + b$.
Se pose alors la question du choix de la fonction noyau, qui doit vérifier les propriétés d'un produit scalaire dans le nouvel espace sans connaître la fonction de transformation. La réponse est données par le théorème de Mercer :

Théorème 4.5.1 *SI $K(x, x')$ est*

- *continue*
- *symétrique : $K(x, x') = K(x', x)$*
- *positive : $\sum_i \sum_j K(x_i, x_j) c_i c_j \geq 0, \forall x_i \in \mathbb{R}^p, \forall c \in \mathbb{R}$*

alors il existe $\phi : \mathbb{R}^p \to H$ un espace de Hilbert tel que :

$$K(x, y) =< \phi(x), \phi(y) >$$

Exemples classiques de noyaux :

— Polynomial : $K(x, x') = (1 + x^T x')^d$

— Radial : $K(x, x') = \exp(-\rho||x - x'||^2)$

— Réseau de neurones : $K(x, x') = \tanh(\alpha x^T x' + \beta)$

4.5.3 Méthodes d'arbres

Les méthodes d'arbres partagent l'espace des *features* en un ensemble de rectangles et entraînent un modèle simple (e.g. une constante) à l'intérieur de chaque rectangle. Le partitionnement récursif est un outil fondamental en *data mining*. Il permet à la fois d'explorer la structure des données, tout en développant des règles de décisions simples et visuelles pour prédire des variables catégorielles (arbres de classification) ou continues (arbre de régression).

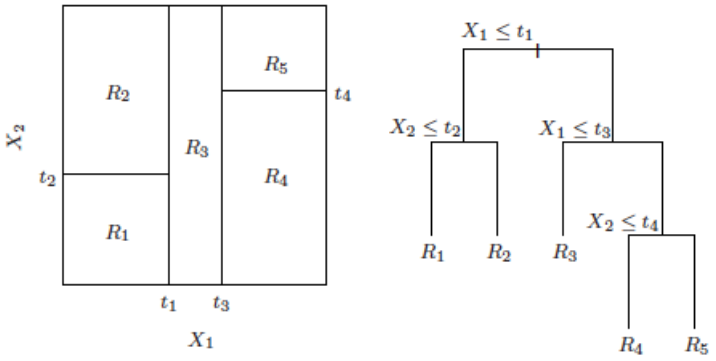


FIGURE 4.7: Partitionnement binaire récursif

Arbre de régression

Bien que l'arbre de régression n'entre pas dans le cadre de la classification, on présente son principe qui s'adapte ensuite à l'arbre de décision.
On considère N observations $(x_i, y_i)_{0 \leq i \leq N}$ où $x_i \in \mathbb{R}^p$ et y_i sont des variables cibles continues. On souhaite partitionner l'espace en M régions $R_1, R_2 \dots R_M$ sur lesquelles on ajuste une constante :

$$f(x) = \sum_{m=1}^{M} c_m I(x \in R_m)$$

En minimisant l'erreur quadratique $\sum (y_i - f(x_i))^2$, la meilleure constante c_m est la moyenne des y_i dans la région R_m :

$$c_m = \frac{\sum_{x_i \in R_m} y_i}{\#\{x_i \in R_m\}}$$

Il faut également décider de la partition qui minimise cette erreur. Le problème tel quel est de complexité combinatoire, mais il existe un algorithme glouton qui le résout. En partant des données complètes, on choisit à chaque itération une variable j sur laquelle on effectue

le partage, et le point s qui partage l'espace suivant cette variable.

$$\min_{j,s}[\min_{c_1} \sum_{x_i \in R_1(j,s)} (y_i - c_1)^2 + \min_{c_2} \sum_{x_i \in R_2(j,s)} (y_i - c_2)^2]$$

où R_1 et R_2 sont les demi-plans définis par :

$$R_1(j,s) = \{X | X_j \leq s\} \quad \text{et} \quad R_2(j,s) = \{X | X_j > s\}$$

Comme mentionné précédemment, la meilleure constante qui résout le sous-problème de minimisation dans la région pour un couple (j,s) donné est la moyenne des y_i sur cette région.

$$c_1 = \frac{\sum_{x_i \in R_1(j,s)} y_i}{\#\{x_i \in R_1(j,s)\}} \quad \text{et} \quad c_2 = \frac{\sum_{x_i \in R_2(j,s)} y_i}{\#\{x_i \in R_2(j,s)\}}$$

La paire optimale (j,s) est obtenue à chaque itération en testant toutes les possibilités (d'où le caractère glouton de l'algorithme). L'opération est ensuite répétée jusqu'à épuisement des variables.

Arbre de décision

Lorsque la cible est une variable catégorielle à K classes, on procède de la même façon qu'avec l'arbre de régression, à ceci près que la mesure d'impureté n'est plus l'erreur quadratique. On introduit une nouvelle métrique adaptée au problème de la classification (voir plus bas).
Dans chaque région finale (appelée feuille), on prédit la classe d'une observation au vote majoritaire (classe associée à la plus grande moyenne empirique d'occurrences) :

$$k_m = \text{argmax}_{k \in \{1,2 \ldots K\}} p_{m,k} \quad \text{où} \quad p_{m,k} = \frac{\sum_{x_i \in R_m} I(y_i = k)}{\#\{x_i \in R_m\}}$$

Les mesures d'impureté traditionnellement utilisées pour la classification sont :

— **L'erreur de classification 0/1** : $\frac{\sum_{x_i \in R_m} I(y_i \neq k)}{\#\{x_i \in R_m\}} = 1 - p_{m,k}$

— **L'indice de Gini** : $\sum_{k=1}^{K} p_{m,k}(1 - p_{m,k})$

— **L'entropie croisée** : $-\sum_{k=1}^{K} p_{m,k} \log p_{m,k}$

Forêts aléatoires

Bien que les arbres de décision produisent des prédictions à faible biais et des règles de décisions simples, leur variance reste élevée. Les forêts aléatoires permettent de réduire cette variance en conservant la même méthodologie. Les forêts aléatoires reposent sur le *bagging*, ou agrégation par *bootstrap* qui consiste à ajuster le même classificateur sur des échantillons *bootstrap* du jeu d'entraînement et moyenne les résultats ou effectue un vote majoritaire dans le cas d'une régression ou d'une classification. Un échantillon *bootstrap* est obtenu par tirage aléatoire avec remise dans l'échantillon d'origine. Comme les arbres sont identiquement distribués, ils ont la même variance intrinsèque notée σ^2. Soit ρ la

Algorithm 1 Forêts aléatoires

for $b = 1 : B$ **do**

Tirer un échantillon *bootstrap* de taille N dans l'échantillon d'entraînement

Ajuster un arbre de décision T_b sur l'échantillon *bootstrap* en itérant la procédure suivante dans chaque région obtenue :

— Choisir aléatoirement m variables parmi l'ensemble des variables.

— Choisir la meilleure variable de partage et le meilleur point de partage en minimisant la fonction d'erreur/d'impureté retenue.

end for

Dans le cas de la régression : la prédiction est égale à $\frac{1}{B}\sum_{b=1}^{B} T_b(x))$

Dans le cas de la classification : la prédiction est obtenue par vote majoritaire.

corrélation entre deux arbres. Calculons la variance de la prédiction de la forêt aléatoire.

$$\begin{aligned}
\mathrm{Var}(\frac{1}{B}\sum_{b=1}^{B} T_b(x)) &= \frac{1}{B^2}\Big(\sum_{b=1}^{B}\mathrm{Var}(T_b(x)) + \sum_{b\neq b'}\mathrm{Cov}(T_b(x), T_{b'}(x))\Big) \\
&= \frac{1}{B^2}\Big(B(B-1)\rho\sigma^2 + B\sigma^2\Big) \\
&= \rho\sigma^2 + \frac{1-\rho}{B}\sigma^2 \\
&\leq \sigma^2 \quad \text{si} \quad 0 \leq \rho \leq 1
\end{aligned}$$

La variance décroît avec B le nombre d'arbres et croît si la corrélation ρ décroît (i.e si le nombre de variables m présélectionnées à chaque partage décroît).

Importance des variables A l'instar des modèles linéaires, on souhaite évaluer l'importance de chaque variable dans la prédiction finale. A chaque partage de l'arbre, l'amélioration de l'impureté correspond à l'importance attribuée à la variable de partage, cumulée sur tous les arbres de la forêt.

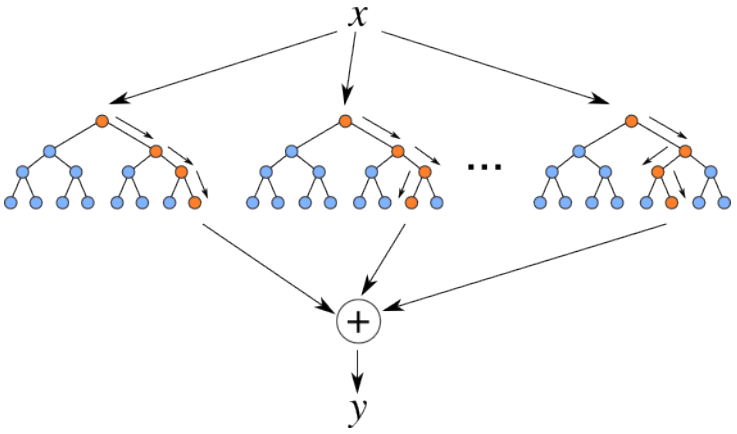


FIGURE 4.8: Forêt aléatoire

4.5.4 Méthodes de *boosting*

Le *boosting* est une approche du *machine learning* qui consiste à créer un classificateur performant à partir d'une multitude de classificateurs simples et peu performants. On peut

assimiler cette idée au *boostrap*, utilisé également dans les *random forests*, mais la théorie sous-jacente est fondamentalement différente.

Gradient boosting tree

Les *Gradient boosting trees* considèrent des modèles de la forme :

$$f(x) = \sum_{m=1}^{M} h_m(x) \quad \text{où} \quad h_m(x) = \sum_j \gamma_{jm} I(x \in R_{jm}) \quad \text{est un arbre de décision}$$

et où L est une fonction de perte arbitraire.
Le *Gradient boosting tree* va construire un modèle additif de façon récursive, suivant un cheminement *forward* :

$$f_m(x) = f_{m-1}(x) + \rho_m h_m(x) \quad \text{où} \quad h_m(x), \rho_m = \text{argmin}_{h,\rho} \sum_{i=1}^{N} L(y_i, f_{m-1}(x_i) + \rho h(x_i))$$

Algorithme de descente de gradient

Afin de résoudre le problème ci-dessus pour n'importe quelle fonction différentiable L, on utilise l'algorithme d'optimisation bien connu de la descente de gradient en effectuant une analogie numérique : la fonctionnelle $L(y, f)$ peut être vue comme une fonction vectorielle prenant comme argument non pas f mais $f(x_1), f(x_2), \dots f(x_N)$. En ignorant la contrainte de forme de f (qui doit être une somme d'arbres), le problème s'écrit alors :

$$\mathbf{f} = \text{argmin}_{\mathbf{f} \in \mathbb{R}^N} L(y, \mathbf{f})$$

où $\mathbf{f}$ est le vecteur des valeurs de f à chacune des N observations. Les techniques d'optimisation numérique résolvent ce problème comme une somme de vecteurs composantes $\mathbf{f} = \sum_{m=1}^{M} \mathbf{h}_m$ où $\mathbf{h}_0$ est la valeur initiale à chaque itération m, f_m est calculé à partir du vecteur courant f_{m-1} qui est la somme des mises à jour précédentes.
La descente de gradient utilisé pour l'algorithme de *gradient boosting* choisit $\mathbf{h}_m = \rho_m g_m$ avec $\rho_m \in \mathbb{R}$ et $g_m \in \mathbb{R}^N$ le gradient de $\mathbf{f} \to L(y, \mathbf{f})$ évalué en $\mathbf{f} = \mathbf{f}_{m-1}$:

$$g_{im} = \Big[\frac{\partial L(y_i, f(x_i))}{\partial f(x_i)}\Big]_{f(x_i)=f_{m-1}(x_i)}$$

Le pas est solution de :

$$\rho_m = \text{argmin}_\rho L(\mathbf{f}_{m-1} - \rho g_m)$$

et permet de mettre à jour le développement :

$$\mathbf{f}_m = \mathbf{f}_{m-1} - \rho_m g_m$$

Cependant, le gradient est seulement défini aux points $(x_i)_{1 \le i \le N}$ et ne peut être calculé en d'autres points, alors que le problème présente une contrainte de "direction" : $h_m(x)$ doit être un arbre de décision. On produit donc un arbre de décision $h_m(x)$ qui prédit un $h_m(x_i)$ aussi proche que possible du gradient $-g_m(x_i)$. On peut faire cette prédiction en ajustant un arbre de régression par les moindres carrés :

$$h_m = \text{argmin}_h \sum_{i=1}^{N} [-g_m(x_i) - h(x_i)]^2$$

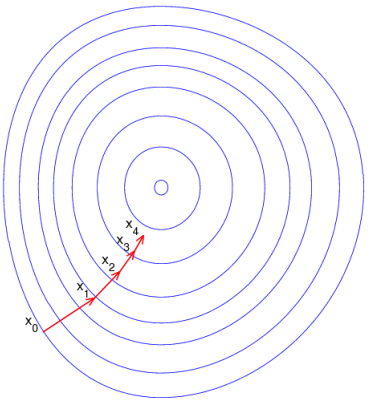


FIGURE 4.9: Descente de gradient : on parcourt la fonction dans le sens opposé du gradient

Idéalement, nous voudrions ajuster l'arbre h_m en minimisant la fonction de perte d'origine L. Comme cela n'est pas toujours possible, on utilise la métrique des moindres carrés qui permet un calcul simple. On utilise les feuilles finales de l'arbre de régression par les moindres carrés mais en prenant dans chacune de ces feuilles la constante qui minimise :

$$\gamma_{jm} = \text{argmin}_\gamma \sum_{x_i \in R_{jm}} L(y_i, f_{m-1} + \gamma)$$

Algorithm 2 Algorithme *Gradient Boosting tree*

Initialiser $f_0(x) = \text{argmin}_\gamma \sum_{i=1}^{N} L(y_i, \gamma)$
for m =1 :M **do**
Pour $i = 1, 2, \ldots, N$, calculer

$$r_{im} = \left[\frac{\partial L(y_i, f(x_i))}{\partial f(x_i)}\right]_{f=f_{m-1}}$$

Ajuster un arbre de régression avec erreur quadratique sur la cible r_{im} pour obtenir les partitions $R_{jm} \forall j$
Pour chaque partition R_{jm}, calculer

$$\gamma_{jm} = \text{argmin}_\gamma \sum_{x_i \in R_{jm}} L(y_i, f_{m-1}(x_i) + \gamma)$$

Mettre à jour $f_m(x) = f_{m-1}(x) + \sum_j \gamma_{jm} I(x \in R_{jm})$
end for
retourner $f_M(X)$

Régularisation

Pour éviter le surapprentissage, on peut régulariser l'étape d'entraînement de l'arbre en limitant le nombre de terme dans son développement. Une autre méthode consiste simplement à "rétrécir" l'arbre, en imitant la contribution de chaque arbre d'un facteur arbitraire

compris entre 0 et 1. A chaque itération, on met à jour le développement comme suit :

$$f_m(x) = f_{m-1}(x) + \nu\rho_m h_m(x), \quad 0 \leq \nu \leq 1$$

Le *shrinkage* ainsi opéré donne généralement de meilleurs résultats que la simple troncature du nombre de termes.

4.5.5 Performance des modèles de classification binaire

Il existe plusieurs métriques (associées à des fonctions de perte) pour calculer la performance des algorithme de *machine learning*.

En particulier, pour la classification binaire, la plupart des métriques sont évaluées à partir de la matrice de confusion, qui résume les résultats d'un modèle sur l'échantillon de test. Un classificateur binaire peut s'écrire en toute généralité

$$\widehat{g}(X) = \mathbb{1}\{s(X) > s_0\} \tag{4.3}$$

où $s(X)$ est le score associé au classificateur (en général assimilé à une probabilité).

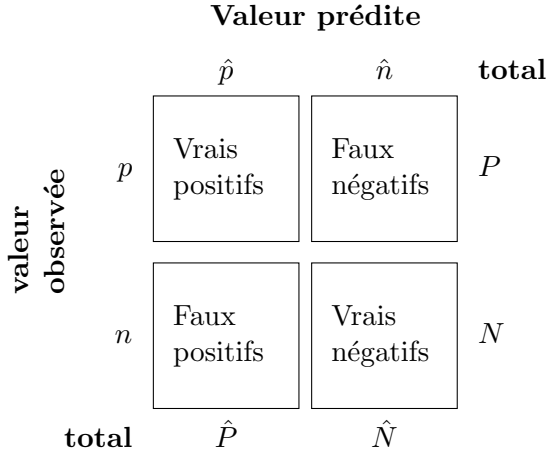


FIGURE 4.10: Matrice de confusion

On trouve parmi les métriques classiques :

— Le score d'exactitude absolue :

$$acc = \frac{p \cap \hat{p} \ + \ n \cap \hat{n}}{P + N}$$

— Le score de précision, qui mesure le taux de vrais positifs retrouvés :

$$prec = \frac{p \cap \hat{p}}{P}$$

— Le score de rappel, qui mesure le taux de vrais positifs parmi les positifs :

$$rappel = \frac{p \cap \hat{p}}{\hat{P}}$$

— Le score f1, une combinaison des scores précédents :

$$prec = \frac{2 * prec * recall}{prec + recall}$$

— La courbe ROC, qui trace le taux de vrais positifs en fonction du taux de faux positifs lorsque le seuil s_0 du classificateur varie. On peut tirer de cette courbe un autre score, appelé AUC ROC, qui correspond à l'aire sous la courbe ROC. Un classificateur aléatoire aura une courbe ROC égale à la première bissectrice, et un AUC ROC de 0,5.

— La courbe précision-rappel, qui trace la précision en fonction du rappel

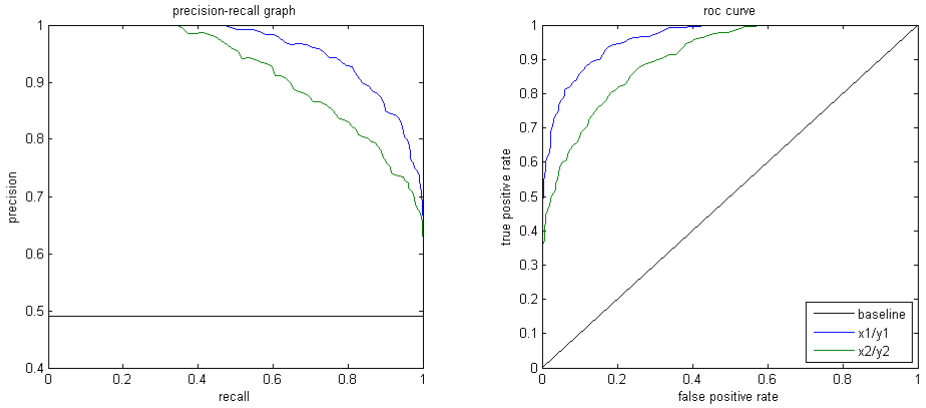


FIGURE 4.11: Courbes ROC et précision-rappel

— La courbe de *lift*, qui trace le pourcentage de vrais positifs en fonction du pourcentage de population sélectionné, trié par score décroissant. La courbe de lift d'un classificateur aléatoire est la première bissectrice.

4.6 Modélisation de l'effet de la communication

On cherche ici à évaluer l'impact d'une communication sur la souscription en considérant l'*uplift*. L'estimation de l'*uplift* sort du contexte de la classification binaire, car on ne cherche plus à prédire une variable cible mais la différence des variables cibles potentielles Y_0 et Y_1.

On sépare la population en groupes traité et témoin : d'un côté les prospects qui ont été mis en relation avec un CEA, de l'autre le reste des devis éligibles. On estime ensuite au sein de chacun de ces groupes la probabilité de souscription nette en fonction des caractéristiques du devis. L'*uplift* sera la différence de ces deux probabilités prédites, calculable pour tous les devis.

On rencontre dans la littérature plusieurs méthodes permettant de calculer l'*uplift* (également appelé *True Lift*) :

— soit de façon indirecte, comme différence de deux probabilités que l'on estime séparément.

— soit en modélisant directement l'*uplift*, en une seule étape.

4.6.1 Performance d'un modèle d'*uplift*

L'entraînement d'un modèle d'estimation d'*uplift* se fait en une ou deux étapes, suivant l'algorithme utilisé (voir plus bas). En revanche, la phase de test du modèle est plus délicate. On ne peut se contenter d'une métrique standard point par point $L(\widehat{TL}, TL)$ pour mesurer la qualité du modèle puisqu'on ne dispose d'aucune variable observée.

Une solution consiste à ranger les observations suivant leur *uplift* estimé, puis se fixer des quantiles et de calculer dans ces quantiles la différence des moyennes empiriques dans les groupes traités et témoins. Cette métrique dépend des quantiles choisis, qui ne doivent être ni trop petits pour disposer de moyennes suffisamment fiables, ni trop grands pour ne pas perdre la finesse du score.

Formellement, si on note $q_1 = 0 < q_2 < ...q_{n-1} < q_n = 1$ les n quantiles arbitraires choisis (pas nécessairement réguliers) et $Q_1, ..., Q_{n-1}$ les intervalles associés, notre mesure de performance du modèle sur un échantillon de test sera :

$$E(q) = \sum_{i=1}^{n} [(\bar{Y}_{i,1} - \bar{Y}_{i,0}) - \bar{TL}_i]^2$$

4.6.2 Estimation par différence de score

La méthode d'estimation la plus intuitive consiste à séparer les groupes traités et témoins et à ajuster le même modèle sur chacun d'entre eux, puis de soustraire les deux estimations pour obtenir l'*uplift*. Formellement, on se donne deux classificateurs f_1 et f_0 et on estime successivement :

1. $\mathbb{P}(Y = 1|X, T = 1) = f_1(X)$ sur la population $\{T = 1\}$
2. $\mathbb{P}(Y = 1|X, T = 0) = f_0(X)$ sur la population $\{T = 0\}$
3. $TL(X) = f_1(X) - f_0(X)$

L'estimation par différence de score permet de se ramener à la classification binaire évoquée plus haut, même si la mesure de la performance ne se fait pas suivant les critères de la classification binaire.

4.6.3 Estimation par effets croisés

Cette estimation (Lo, 2002) se fait en une étape. On donne un classificateur f et on intègre à l'ensemble des variables explicatives la variable T de traitement et les variables croisées $X * T$, de sorte qu'un individu traité voit ses variables explicatives dupliquées et qu'un individu non traité se voit rajouter des variables nulles

$$\mathbb{P}(Y = 1|X, T) = f(X, T, X * T) \tag{4.4}$$

$$\widehat{TL} = f(X, T, X * T|T = 1) - f(X, T, X * T|T = 0) \tag{4.5}$$

On peut montrer dans le cas des modèles linéaires généralisés que cette technique est équivalente à la précédente à ceci près qu'on peut tester sur la dernière la significativité des coefficients associés aux variables croisées. En effet, ces modèles sont estimés par maximum de vraisemblance.

$$g(\mathbb{P}(Y = 1|X, T, X * T)) = \theta_1.X + \theta_2.X * T \tag{4.6}$$

La log-vraisemblance est ici séparable :

$$\begin{aligned} l(X_i, Y_i, \theta_1, \theta_2) = & \sum_i [Y_i \log(g^{-1}(\theta_1.X_i + \theta_2.X_i * T_i)) \\ & + (1 - Y_i) \log(1 - g^{-1}(\theta_1.X_i + \theta_2.X_i * T_i))] \\ = & \sum_{i \in \{T_i = 0\}} \log(1 - g^{-1}(\theta_1.X_i)) + (1 - Y_i) \log(1 - g^{-1}(\theta_1.X_i)) \\ & + \sum_{i \in \{T_i = 1\}} \log(1 - g^{-1}((\theta_1 + \theta_2).X_i)) + (1 - Y_i) \log(1 - g^{-1}((\theta_1 + \theta_2).X_i)) \end{aligned}$$

En posant $\theta_1 + \theta_2 = \theta_3$, les deux termes peuvent être maximisés séparément et on obtient le même résultat que le modèle par différence.
Tout comme les modèles à différence de score, ce procédé n'est en fait qu'une transformation préalable des données et se prête théoriquement à n'importe quel type de modèle (GLM, arbres, Gradient Boosting,...).

Estimation par l'*uplift random forest*

Cet algorithme (Guelman et al., 2015b) fait partie des méthodes directes d'estimation de l'*uplift*. Il s'agit, comme son nom l'indique, d'un algorithme de forêt aléatoire associé à un critère de partitionnement bien choisi. Posons $Y_1 \sim p_1$ et $Y_0 \sim p_0$. A chaque partitionnement (binaire) d'un nœud des arbres en deux régions R et $\bar{R}$, on cherche à minimiser non plus l'impureté par rapport à la variable cible Y, mais à maximiser une distance D entre les deux lois de probabilité p_1 et p_0, pondérée par les effectifs :

$$C = \frac{M_R}{M} * D(p_1||p_0|R) + \frac{M_{\bar{R}}}{M} * D(p_1||p_0|\bar{R}) \tag{4.7}$$

où M_R, $M_{\bar{R}}$ et M représentent les effectifs associés à chaque région et total (respectivement).
Les distances utilisées peuvent être :

- Le contraste de Kullback-Leibler : $K(p||q) = \sum_y p(y) log(\frac{p(y)}{q(y)})$
- La distance euclidienne : $L2(p||q) = \sum_y (p(y) - q(y))^2$
- La distance khi-deux : $\chi^2(p||q) = \sum_y \frac{(p(y)-q(y))^2}{p(y)}$
- La distance $L1$: $L1(p||q) = \sum_y |p(y) - q(y)|$

Afin de pénaliser les partitionnements qui génèrent trop de déséquilibre entre les effectifs traité et non traité dans les nouveaux nœuds, on corrige cette distance dans le critère de partitionnement par un facteur D_{corr} :

$$C = \frac{D(p_1||p_0)}{D_{corr}}$$

$$D_{corr} = H(\frac{M_1}{M}, \frac{M_0}{M}) * D(p_1||p_0) + \frac{M_1}{M} * H(p_1) + \frac{M_0}{M} * H(p_0)$$

où $H(p) = -p(R) * \log(p(R)) - p(\bar{R}) * \log(p(\bar{R}))$ est la fonction d'entropie et $H(x, y) = -x \log(x) - y \log(y)$
On choisit alors le partitionnement qui maximise le critère C.

La *Causal conditional inference forest*

Cet algorithme (Guelman et al., 2015a) est similaire à celui de l'*uplift random forest*, mais y apporte certaines améliorations.

Les *uplift random forest* ont une tendance au surapprentissage, car tous les arbres sont développés jusqu'à une profondeur qui dépend d'un même critère, à savoir une taille minimale de nœud. Comme l'*uplift* est une quantité de second ordre (différence de probabilités estimées), le modèle doit être capable de distinguer l'impact relatif du traitement et des autres variables sur la cible. La variabilité de l'effet de traitement sur la cible est généralement plus faible que la variabilité due à l'ensemble des autres variables, de sorte que le surapprentissage (synonyme de grande variabilité) aura un effet bien plus fort sur l'*uplift*, dont l'estimation sera très instable.

Dans le cadre des *random forest* classiques, le surapprentissage est géré par une étape d'élagage postérieure à l'entraînement, durant laquelle on teste en partant des nœuds terminaux si la séparation améliore les performances du modèle sur l'échantillon de test.

Dans le cas de l'*uplift*, les auteurs de l'algorithme proposent une méthode permettant, à chaque nœud, de tester une hypothèse de significativité de l'interaction entre le traitement et chacune des n variables tirées au niveau du nœud. Pour ce faire, on définit la variable :

$$W = \begin{cases} 1, & \text{si } T = 1 \text{ et } Y = 1 \\ 1, & \text{si } T = 0 \text{ et } Y = 0 \\ 0, & \text{sinon} \end{cases}$$

Ceci permet de définir des statistiques de test (complexes) pour tester l'hypothèse nulle jointe :

$$H_0 = \cap_{j=1}^{n} H_{0,j}$$
$$\text{avec } H_{0,j} : \ \mathbb{E}[W|X_j] = \mathbb{E}[W]$$

Si cette hypothèse n'est pas rejetée, alors on arrête la croissance de la branche. Dans le cas contraire, on choisit la variable pour laquelle l'interaction avec T est la plus significative, puis on effectue le *split* en maximisant le critère suivant, qui représente la statistique d'un test du chi-deux de l'effet d'interaction entre la variable X_{j*} et T :

$$G^2(\Omega) = \frac{(L-4)[(\bar{Y_{1,l}} - \bar{Y_{0,l}}) - (\bar{Y_{1,r}} - \bar{Y_{0,r}})]}{\hat{\sigma}^2(L_{1,l}^{-1} + L_{0,l}^{-1} + L_{1,r}^{-1} + L_{0,r}^{-1})} \tag{4.8}$$

où les indices l et r représentent les nœuds respectivement à gauche et à droite du *split*, L_1 et L_0 les effectifs traité et non traité et $\bar{Y_1}$ et $\bar{Y_0}$ la proportion de variables cibles au sein des groupes traité et non traité.

Chapitre 5

Premiers résultats et impact *business*

5.1 Exploration des données

La première phase du projet consiste à produire une série de statistiques descriptives liées aux ASV pour retrouver les ordres de grandeur des études métiers et ainsi s'assurer la cohérence des données utilisées. Cette phase permettra également d'identifier des variables d'intérêt, de valider certaines intuitions et d'orienter le travail de modélisation. Elle permettra également de vérifier la pertinence des nouvelles données utilisées (en particulier des données sur les prix de la concurrence).

La fenêtre temporelle de l'étude s'étale de janvier 2014 à mai 2015, soit la plus large période sur laquelle on dispose des données de l'outil ViaDialog qui gère les ASV.
Pour des raisons de confidentialité, les valeurs ne sont pas indiquées. On retiendra de ces graphes les proportions et les tendances.

5.1.1 Statistiques sur les TE et sur la communication

Flux de tarifs édités Le flux de tarifs édités correspond aux grandeurs métiers, il est relativement régulier au cours de la période. Parmi ces devis, environ la moitié sont éligibles aux ASV. On constate que la majorité des devis sont faits en heures ouvrées et sont donc susceptibles d'être rappelés à chaud (s'ils sont éligibles).

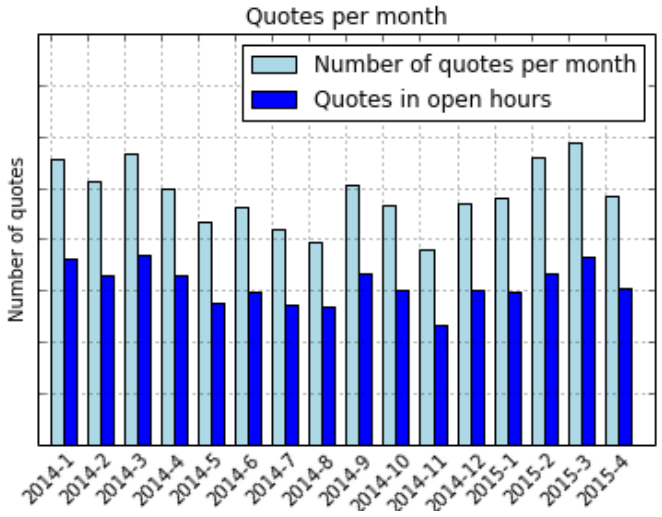


FIGURE 5.1: Flux de tarifs édités

Canal d'origine des devis Cette statistique, qui n'est pas affichée ici, permet de vérifier les chiffres métiers sur la part de tarifs édités par canal d'origine, répartis entre agrégateurs et site DA. Un des canaux correspond à des devis par téléphone, théoriquement non éligible aux ASV. Le problème, évoqué dans le *Data Quality Assessment*, provient du non-versionnage systématique des images du contrats qui provoque l'écrasement de certaines valeurs, en particulier de l'origine. Ces contrats dont l'origine est inexacte seront retirés de la table pour la modélisation.

Parts de devis par file de priorité On constate que la file 1 (la plus prioritaire) est la plus peuplée, et que la file 3 ne représente que 10% des devis. L'algorithme existant n'est donc pas très granulaire dans son filtrage de la file 1. Cette répartition des volumes peut éventuellement permettre, si des biais de files sont identifiés, de mener l'étude uniquement sur la file 1 (60% des données).

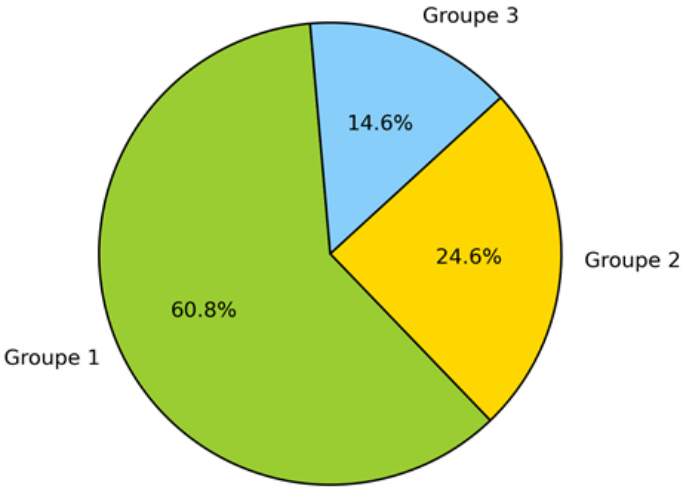


FIGURE 5.2: Parts de TE par file de priorité

Délai de rappel par heure d'édition du devis Cette statistique révèle qu'en heure ouvrées, la moitié des devis sont rappelés presque immédiatement (après le délai incompressible). Ceci est du au fait que la file 1 est la plus représentée, comme l'indique la statistique par groupe.
En croisant les délais moyen et médian de rappel par heure de la journée (heure d'édition du devis), on constate que le processus d'ASV est régulier (pas d'horaire privilégié pour les ASV).
Le délai moyen est impacté par les devis faits en heures non ouvrées. Il est ici moins intéressant que le délai médian.

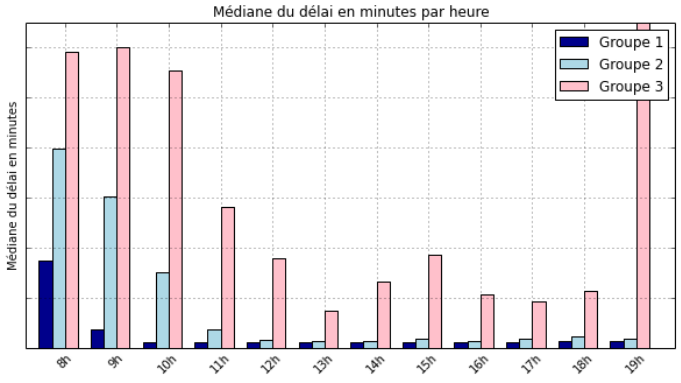


FIGURE 5.3: Délai médian par heure

Taux de réponse La taux de réponse est légèrement décroissant par file de priorité. Ici, on considère les communications effectives (d'une durée supérieure à 20 secondes), donc filtrées des éventuelles boîtes vocales non détectées par l'outil.

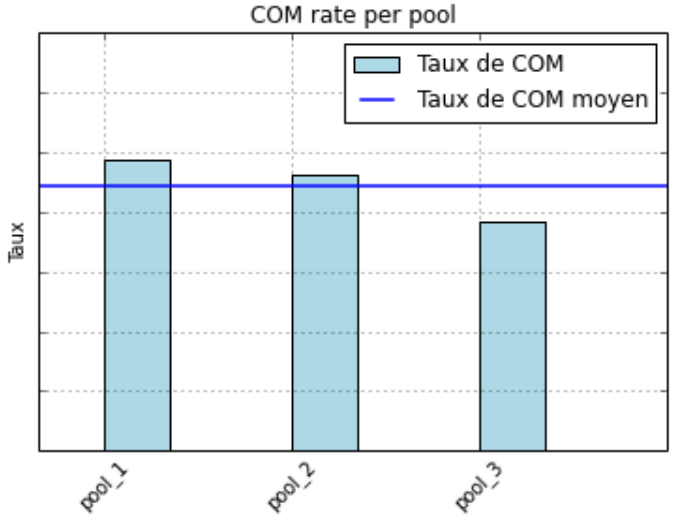


FIGURE 5.4: Taux de communication moyen par files de priorité

Taux de communication par CSP suivant l'heure de la journée Une des intuitions du métier est le lien entre joignabilité et catégorie socio-professionnelles. Hormis quelques fluctuations et des intuitions simples (meilleure joignabilité des retraités, heures creuses pour les agriculteurs), la tendance est à peu près la même pour toutes les CSP.

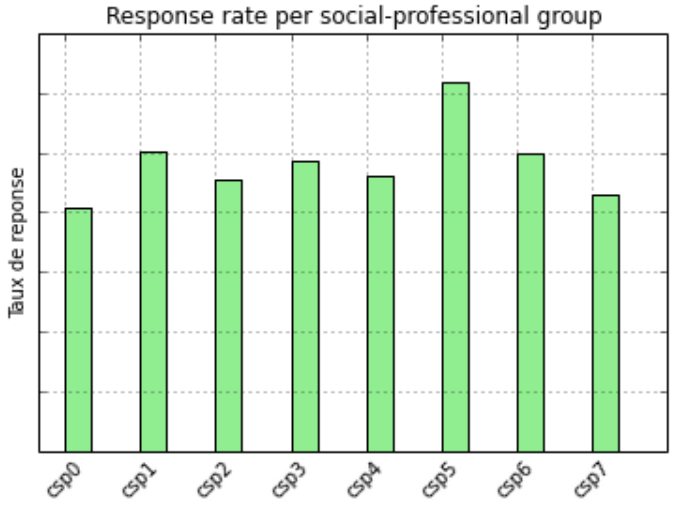


FIGURE 5.5: Taux de communication par CSP

5.1.2 Statistiques sur la conversion

Pour les statistiques concernant la conversion, certains filtres doivent être appliqués :

— on exclut les souscriptions immédiates faites en ligne (dans les 5 minutes qui suivent le devis), sur lesquelles la communication ne peut influer. Pour ces prospects, il n'y a pas de distinction entre les étapes devis et souscription.

— on ne considère que les conversions nettes : on exclut les annulations dans les 30 jours suite à la souscription (*cool-off*).

— on ne considère que les communications effectives (d'une durée supérieure à 20 secondes, pour filtrer les répondeurs non détectés par ViaDialog et les appels non décrochés par le CEA), c'est à dire les communication pour lesquelles le CEA a eu un réel impact.

Taux de conversion avec ou sans communication Il s'agit d'une statistique descriptive importante pour notre étude. On constate que l'ASV a un effet positif très net sur la conversion moyenne. Ces chiffres sont néanmoins à prendre avec précaution car ils ne tiennent pas compte des biais de sélection éventuels sur la communication. Ces biais seront étudiés dans la phase de modélisation.

***Uplift* par files de priorité** L'impact de la communication sur la conversion est la même pour les files 1 et 2 et légèrement supérieure pour la file 3. Une conclusion intéressante de cette statistique est que la priorisation actuelle n'est pas optimale puisqu'elle ne cible pas les prospects sur lesquels la communication a le plus d'effet. Il est intéressant de croiser cette statistique avec l'impact du délai de rappel sur la conversion, sachant que ce délai est croissant par file de priorité.

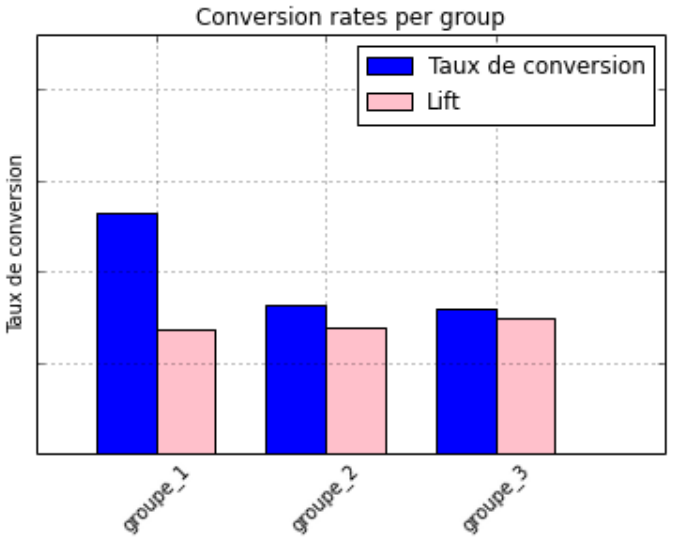


FIGURE 5.6: *uplift* par files de priorité

Conversion vs nombre de relances Le processus en place permet de relancer jusqu'à 5 fois le prospect suite à son devis. On constate que l'efficacité de la communication décroît avec le nombre de tentatives, ce qui pose la question de l'utilité des dernières relances.

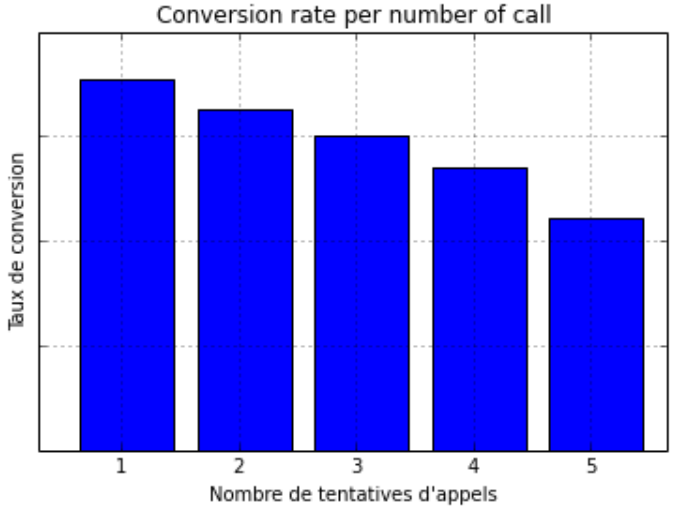


FIGURE 5.7: Conversions vs nombre de relances

Conversion vs délai entre devis et communication La chaleur du rappel influe légèrement sur la conversion pour la file 1, mais ne semble pas jouer sur les files 2 et 3. Ce graphe justifie le fait qu'on puisse mener dans un premier temps l'étude sur l'ensemble des files de priorité, car la différence de traitement entre les files est principalement résumée par le délai entre devis et rappel.

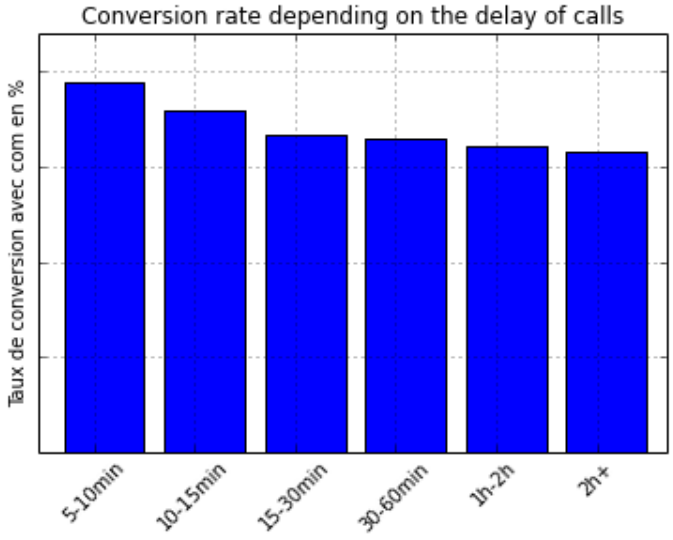


FIGURE 5.8: Conversions vs délai de rappel

Taux de conversion suivant le canal d'origine Cette statistique révèle deux choses : d'une part que le taux de conversion est meilleur pour les devis provenant du site DA, ce qui s'explique par le fait que ces devis ont directement choisi DA. D'autre part, on constate que l'*uplift* est légèrement meilleur sur les devis DA, alors que la priorisation actuelle n'intègre pas le canal d'origine dans son score. De plus, ces devis ont un coût d'acquisition quasi nul, ce qui accentuerait davantage la différence d'*uplift* en terme de marge.

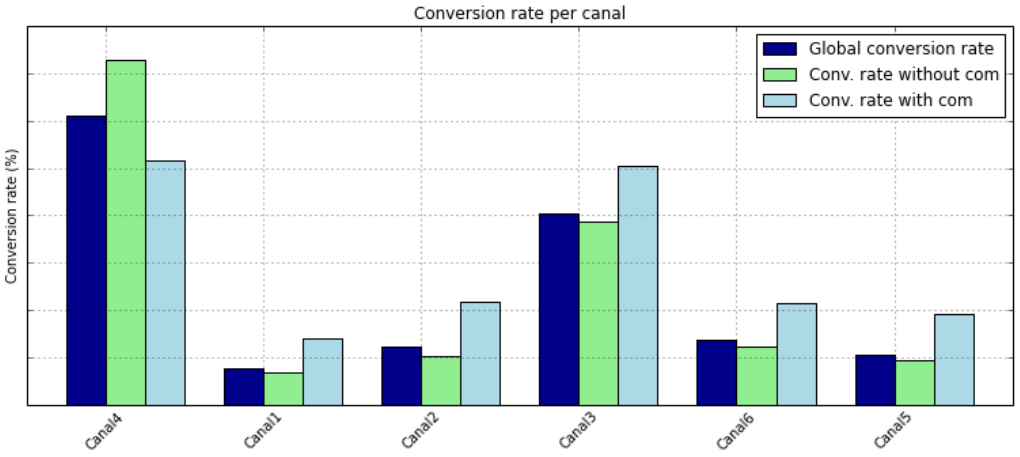


FIGURE 5.9: Conversions par canaux

Taux de conversion suivant le rang parmi la concurrence Sans surprise, le rang influence notablement le taux de conversion. En revanche, il est intéressant de constater (notamment sur les premiers rangs) que l'*uplift* décroît avec le rang : ce sont les devis sur lesquels DA est premier que l'on a le plus de levier. L'effet du rang sur la conversion n'est donc pas compensé par l'effet de la communication.

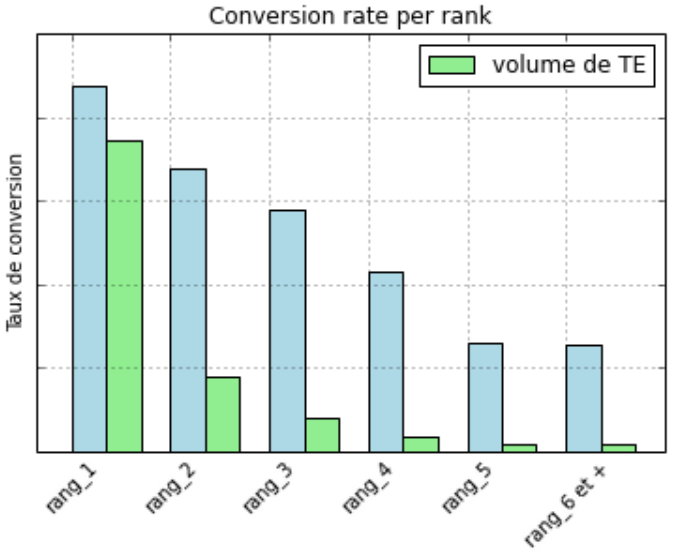


FIGURE 5.10: Conversions vs rang

5.2 Evaluation de l'algorithme existant (*business case*)

A partir des premières statistiques descriptives, il est possible d'évaluer le score de priorisation existant.

Pour évaluer les trois grandeurs de notre KPI, à savoir la joignabilité, l'*uplift*, et la valeur client, on se base sur les statistiques univariées du paragraphe précédent. Le score existant

se basant sur certains segments de valeur client, on attribue à chacun des groupe de priorité une valeur moyenne de NBV.

Les résultats décrits dans ce paragraphe sont présentés sous la forme d'un graphe où l'abscisse représente la part de population rappelée (rangée par file de priorité) et l'ordonnée représente le bénéfice cumulé. L'exemple se base sur des valeurs clients arbitraires par souci de confidentialité, simplement pour illustrer la méthodologie utilisée.

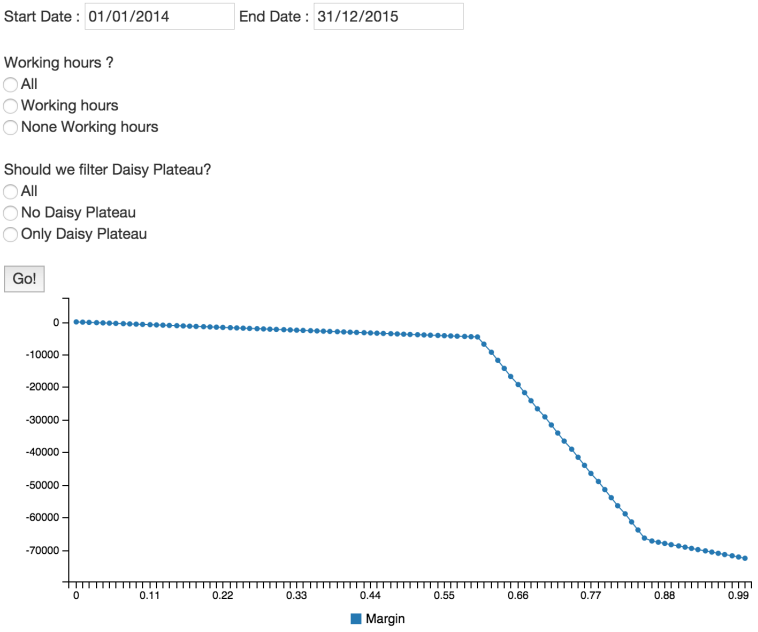


FIGURE 5.11: Méthodologie de calcul de la performance

5.3 Premiers résultats des modèles

Après avoir obtenu les *insights* descriptifs sur les données grâce aux statistiques descriptives, les premières itérations du modèle ont été lancées.

5.3.1 Caractéristiques de la table finale

Une fois le *preprocessing* effectué et les principaux filtres appliqués, la table finale comporte environ 750000 lignes pour 250 colonnes, pour une taille totale d'environ 1 Go. Plusieurs filtres seront ensuite appliqués selon les besoins du modèle (variable, cible, traitement, variables non pertinentes).

A ce stade, les données de navigation n'ont pas encore été jointes. Leur *preprocessing* est en cours de finalisation, et cela évite aussi de surcharger les tables pour les premiers tests.
Le rang parmi la concurrence n'étant disponible que pour un agrégateur et associé aux devis via une clé de jointure *ad hoc* incomplète, elles sont très peu remplies à l'échelle de la table. Les résultats du modèle de prédiction du rang de DA n'étant pas encore complètement disponibles, ils ne seront pas intégrés dans les premiers tests des modèles.

5.3.2 Modèle de mise en relation

La première itération du modèle de mise en relation a été faite sur l'ensemble des variables de la table. Les résultats très positifs ont permis de mettre en évidence le fait que certaines des variables issues des tables contrats étaient parfois renseignées ou écrasées après la communication, ce qui leur confère un pouvoir prédictif fallacieux. Un exemple de telles variables sont des données renseignées durant la communication sur l'ancienne compagnie d'assurance et l'ancienne prime versée par le prospect.

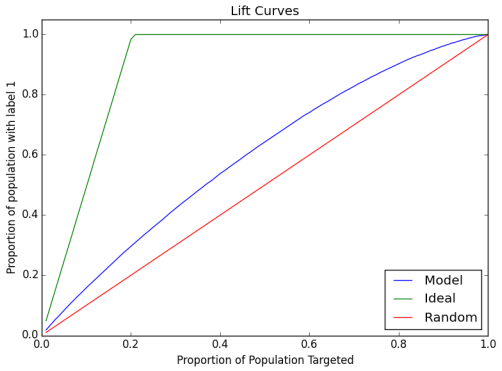
Ce problème fait écho au *Data Quality Assessment* du chapitre 3, où l'on mentionnait le "piège" majeur que constituait l'absence d'historisation systématique des versions du contrat. Ces variables sont identifiées par itération successives des modèles (elles apparaissent comme très prédictives sans explication apparente), et confirmées par les contacts du métier.

Une approche *forward* a finalement été choisie, en sélectionnant les variables fiables, ce qui réduit l'ensemble des *features* à 140 variables. Des vérifications avec le métier sont en cours concernant les variables douteuses.
Les résultats complets sont à ce jour disponibles pour les *Random Forest* et le et *Gradient Boosting*. Les modèles de SVM sont actuellement en cours de test, car ils nécessitent un calibration plus fine.

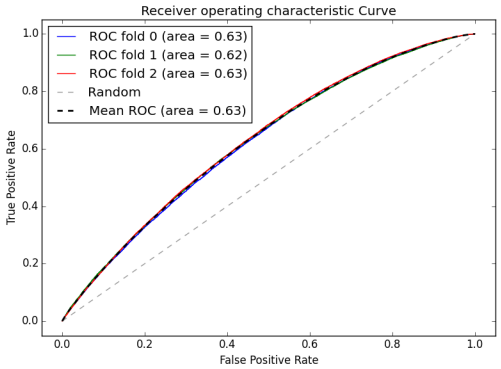
Random Forest

Les résultats des *Random Forest* sont résumés par les indicateurs usuels décrits dans le chapitre précédent :

Performance Cette première itération du modèle n'est pas très performante. L'AUC est relativement faible, ce qui s'explique par le choix d'un ensemble restreint de variables qui ne capture pas toute l'information disponible. On attend une meilleure performance lorsque l'intégralité des variables pertinentes sera ajoutée.



(a) Courbe de lift



(b) Courbes ROC

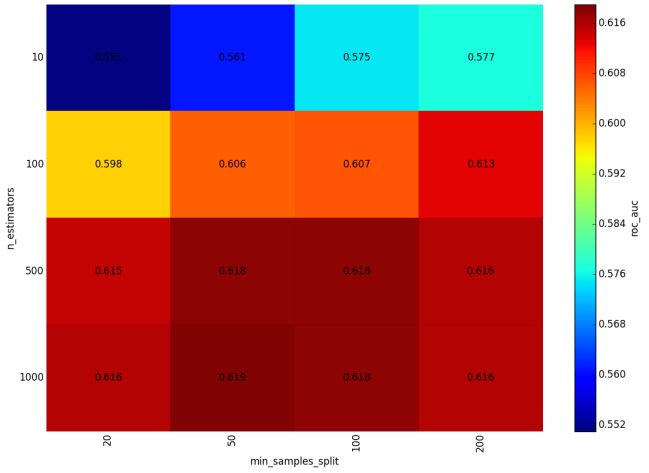
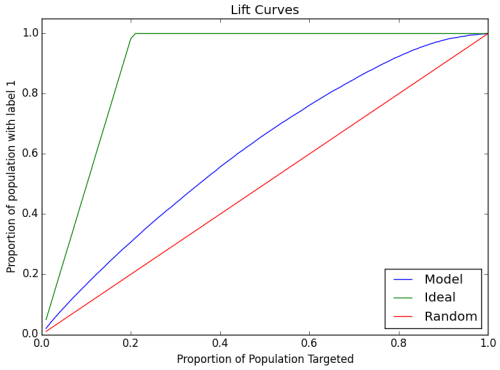


FIGURE 5.13: *Grid search* pour la *Random Forest*

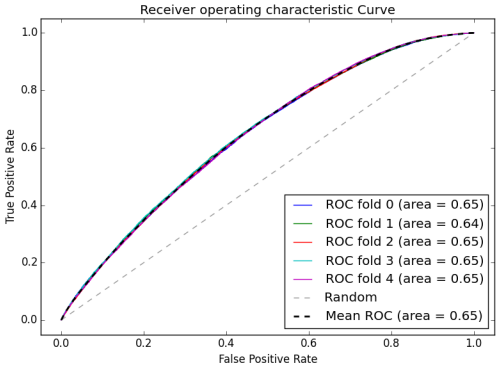
Gradient Boosting

Les résultats du *Gradient Boosting* sont résumés par les mêmes indicateurs.

Performance Cette première itération du modèle n'est pas très performante. L'AUC est relativement faible, ce qui s'explique par le choix d'un ensemble restreint de variables qui ne capture pas toute l'information disponible. On attend une meilleure performance lorsque l'intégralité des variables pertinentes sera ajoutée.



(a) Courbe de lift



(b) Courbes ROC

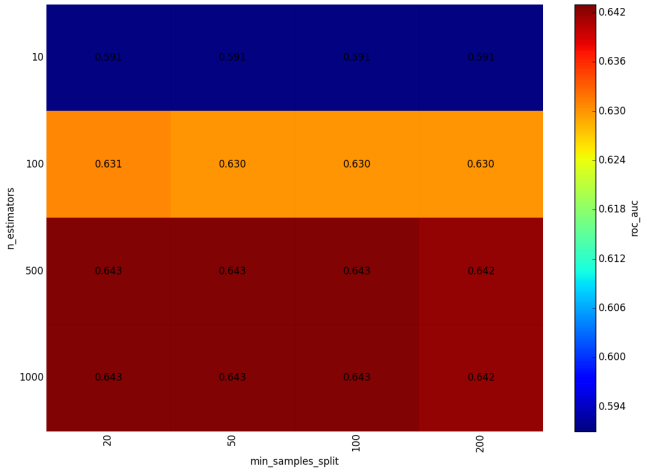


FIGURE 5.15: *Grid search* pour le *Gradient Boosting*

Importance des variables L'estimation et l'analyse de l'importance des variables n'est pas indispensable à un modèle de *machine learning* purement prédictif. Elles ajoutent cependant un caractère explicatif au modèle, lorsque les variables sont interprétables. C'est le cas dans cette première itération puisqu'aucune manipulation préalable des variables n'a été faite, en dehors de leur *preprocessing*.

On constate que les variables les plus prédictives sont celles liées à la prime et aux caractéristiques du prospect, en particulier son âge. L'ancienneté du véhicule joue également, ainsi que certaines données socio-démographiques, où par exemple la population est moins active.
Ces variables se retrouvent dans les deux modèles. Cependant, le *Gradient Boosting* a tendance à sélectionner de façon plus nette les variables : leur importance relative décroît plus vite que pour les *Random Forest*.

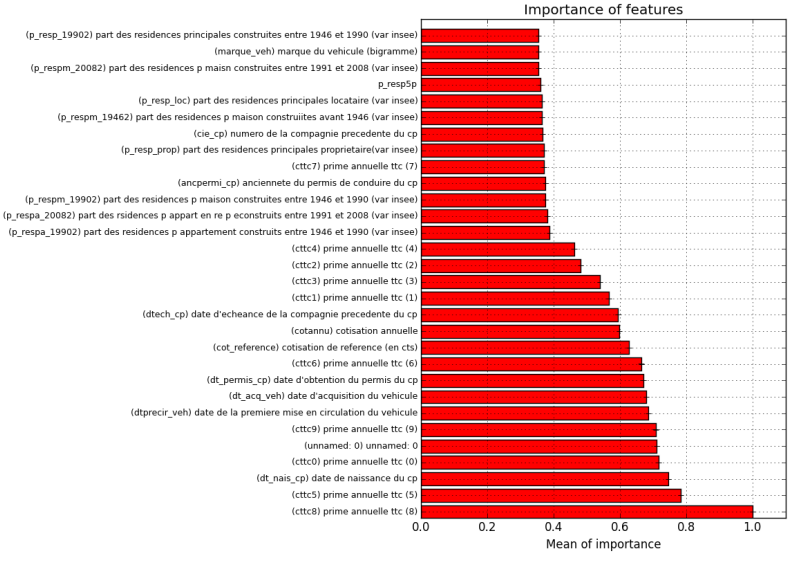


FIGURE 5.16: Importances des variables pour la *Random Forest*

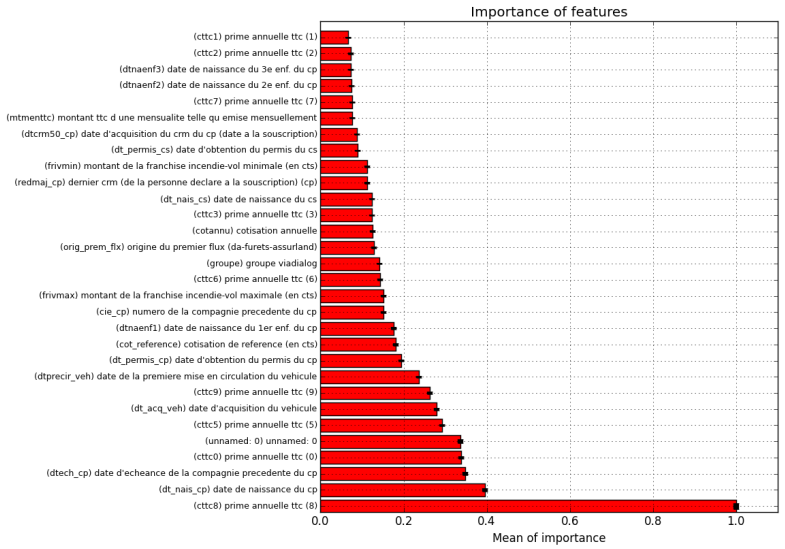


FIGURE 5.17: Importances des variables pour le *Gradient Boosting*

5.3.3 Modèle d'*uplift*

La première itération du modèle d'*uplift* a été faite un ensemble de variables dont certaines sont en cours de validation.

Pour rappel, la performance du modèle d'*uplift* est mesurée par l'histogramme suivant, qui prend en abscisse les quantiles de score estimé et qui mesure le lift observé dans chacun de ces quantiles par différence de moyenne parmi les individus appartenant à ce quantile entre les traités et les témoins.

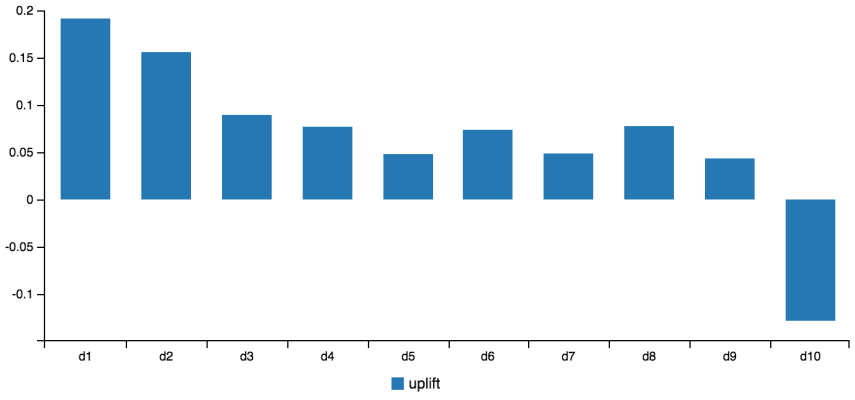


FIGURE 5.18: Modèle d'uplift *ccif*

L'algorithme utilisé ici est le *Causal conditional inference forest*, qui nécessitera d'être optimisé par *Grid search*. Les premiers résultats sont assez positifs, mais doivent être considérés avec précaution.

On constate que l'*uplift* est globalement décroissant, et devient même négatif dans le

dernier décile, ce qui montre *a priori* qu'une partie de la population sont des clients "à ne pas déranger", pour lesquels l'ASV à un impact négatif sur la conversion.

5.4 *Backtesting* du modèle

Une fois les modèles calibrés et la méthodologie validée par le métier, l'étape suivante sera le *backtesting* complet des scores produits, sur le modèle du *business case* décrit plus haut.

Les trois scores individuels, à savoir la joignabilité, l'*uplift* et la valeur client seront testés sur les données pour déterminer plus précisément la rentabilité actuelle du processus d'ASV. Le même graphe sera produit sur un échantillon de validation en classant par déciles les individus suivant chacun des scores, puis suivant le score final ($G = \mathbb{P}(communication) * uplift * NBV$), en examinant le bénéfice observé par déciles.

5.5 Ouverture : utilisation de l'*active learning*

La classification semi-supervisée est la branche du *machine learning* qui traite des problèmes de classification dans lesquels les données ne sont que partiellement labélisées. L'*active learning* s'intéresse à la classification semi-supervisée dans le cas où on choisit la partie des données que l'on veut labeliser (c'est à dire pour laquelle on souhaite observer la variable cible), ce qui induit une recherche d'un optimum de labels à donner parmi les données.

5.5.1 Les ASV vus comme un problème d'optimisation

On peut replacer les ASV, et plus généralement le problème de ciblage dans des campagnes marketing et de l'effet de traitement, dans un contexte d'*active learning*. Il s'agit d'un problème de décision dans lequel on recherche l'action optimale à déclencher face à des prospects. En l'occurrence, deux actions sont possibles : rappeler ou non le prospect. Le problème peut se généraliser à n actions (coup de téléphone, mail, sms,...)

Comme évoqué dans le chapitre 2, ce problème revient à arbitrer entre le rappel des prospects qu'on estime être appétents, que l'on détermine via un score entraîné sur les données passées (phase d'exploitation), et de nouveaux essais de rappels sur des individus a priori écartés par le score, mais potentiellement à tort car on ne dispose pas de suffisamment de données pour estimer précisément ce score (phase d'exploration).

5.5.2 Algorithme des bandits contextuels

Les algorithmes des bandits sont des algorithmes d'*active learning* où le problème est vu comme le choix d'une action parmi plusieurs possibles. On observe ensuite le résultat (*payoff*) de l'action choisie uniquement. L'objectif est de minimiser le regret cumulé (différence par rapport au payoff optimal, a priori inconnu), par un compromis exploitation/exploration qui permettra à la fois d'exploiter suffisamment le bras que l'on estime le plus rémunérateur (exploitation) et de tester tous les bras suffisamment de fois pour identifier le bras optimal (exploration).

A chaque itération t de l'algorithme :

— On observe $\mathcal{A}_t$, l'ensemble des actions, ou bras souvent constant (indépendant de t).

— On observe $x_{t,a}$ le "contexte" associé à l'itération t et au bras a, i.e. l'ensemble des variables explicatives.

— On choisit un bras a, et on constate le rendement de celui-ci, $r_{t,a}$

— Grâce à l'observation du rendement, l'algorithme met à jour sa stratégie de choix.

Généralement, la stratégie de l'algorithme consiste à optimiser le regret en rendement de la stratégie $((a_t), t = 1, \ldots, T)$ par rapport au choix optimaux, défini par :

$$R_A(T) = \mathbb{E}[\sum_{t=1}^{T} r_{t,a_t^*}] - \mathbb{E}[\sum_{t=1}^{T} r_{t,a_t}]$$

Le regret d'une stratégie aléatoire est linéaire en T. Une bonne stratégie doit donc aboutir à un regret en $O(T)$.

Il existe plusieurs variantes à l'algorithme des bandits contextuels :

— **Algorithme *epsilon-greedy*** : Cette stratégie consiste à équilibrer exploration et exploitation en choisissant alternativement le bras "gourmand" (celui qu'on estime jusque-là être le meilleur) et un bras aléatoire (exploration), avec probabilité $1 - \epsilon$ et ϵ respectivement. Il s'agit de la stratégie d'exploration la plus simple. Cette stratégie nécessite d'adapter la fraction d'exploration ϵ au cours du temps pour être sous-linéaire.

— **L'algorithme *epoch-greedy*** (Langford and Zhang, 2008) propose une méthode d'adaptation de ϵ au cours du temps. Son regret est en $O(T^{2/3})$ et peut atteindre $O(\log T)$ avec des hypothèses supplémentaires.

— **Les algorithmes UCB** (*Upper Confidence Bound*) font partie des meilleurs algorithmes dans le cas général. Cette stratégie consiste à produire des majorants (*Upper Confidence Bound*, la borne supérieure d'un intervalle de confiance de probabilité $1-d$) pour les payoff estimés pour choisir le bras tel que la quantité (payoff estimé + UCB) est maximale. On tient ainsi compte de la précision de l'estimation du payoff de chaque bras, pour réaliser un compromis exploration/exploitation (les bras qui ne sont pas estimés avec une précision suffisante ont plus de chances d'être rejoués).
Leur regret est en $O(\sqrt{T})$. La version linéaire de cet algorithme (Li et al., 2010) suppose un regret de la forme :

$$\mathbb{E}[r_{t,a}|x_{t,a}] = x'_{t,a}\theta^*_a \tag{5.1}$$

La méthode consiste à estimer θ par une régression *ridge* à partir de la matrice D_a du contexte et le vecteur c_a des rendements historiques observés jusque là pour le bras a.

$$\hat{\theta}_a = (D'_a D_a + I_d)^{-1} D'_a c_a = A_a^{-1} D'_a c_a$$

puis d'utiliser l'intervalle de confiance de probabilité $1 - \delta$:

$$\left|x'_{t,a}\theta_a - \mathbb{E}[r_{t,a}|x_{t,a}]\right| < \alpha\sqrt{x'_{t,a}A_a^{-1}x_{t,a}}$$

avec $\alpha = 1 + \sqrt{\log(2/\delta)/2}$, pour choisir le bras :

$$a_t = \text{argmax}_{a\in\mathcal{A}_t}\left(x'_{t,a}\hat{\theta}_a + \alpha\sqrt{x'_{t,a}A_a^{-1}x_{t,a}}\right)$$

Cette quantité est le rendement estimé, augmenté de l'incertitude d'estimation.

Conclusion

En assurance directe, les informations que les assureurs sont capables de récolter, notamment via les technologies digitales, permettent de cibler suivant certains critères (rentabilité, profil de risque,...) les clients susceptibles de réagir positivement à une campagne marketing. Le projet Hot Lead Management s'intéresse en particulier à l'optimisation des relances téléphoniques après des devis effectués en ligne.
Le travail effectué jusqu'ici a d'abord permis de constater les limites du processus de relance téléphonique actuel. Pour améliorer le score existant, uniquement basé sur des considérations métier, trois leviers ont été identifiés : la joignabilité des prospects, l'impact de la conversation téléphonique sur la souscription (*uplift*) et la valeur client, cette dernière mesurant les coûts et bénéfices liés à la souscription de nouveaux contrats.
Jusque là peu utilisé dans les campagnes marketing, l'*uplift* est pourtant la brique fondamentale du modèle : on ne cherche pas à cibler directement les prospects appétents, mais ceux sur lesquels la conversation téléphonique aura le plus d'impact. L'estimation de l'*uplift* se heurte néanmoins aux biais liés induits par le processus existant. Un travail de réflexion a été mené pour identifier ces biais et formuler certaines hypothèses pour la modélisation.
Le nettoyage et l'agrégation des différentes bases de données utilisées grâce aux technologies *Big Data* de calcul distribué (Hive, Impala) a permis de formuler certaines recommandations quant au *processing* de certaines base de données, mal adapté à l'utilisation qui en est faite dans le cadre du projet. En particulier, l'absence de certaines clefs de jointures et l'absence de versionnage systématique des données contrats impactent considérablement la construction de la table finale utilisée pour le modèle.
Une première phase exploratoire de statistiques descriptives macroscopiques a permis de confronter les données aux chiffres du métier pour s'assurer de leur qualité, de valider certaines intuitions et d'obtenir des *insights* supplémentaires pour orienter le travail de modélisation. Les premiers modèles ont été entraînés en utilisant des algorithmes de *machine learning* classiques, mais aussi des algorithmes plus innovants pour la modélisation de l'*uplift*. Les premiers résultats, bien que perfectibles et à prendre avec précaution, sont prometteurs. Une fois cette première étape de modélisation finalisée, des méthodes d'*active learning* seront testées, en particulier l'algorithme des bandits contextuels, qui sont particulièrement adaptées à notre problématique mais nécessitent une partie d'exploration aléatoire qui n'est pas encore supportée par les contraintes métier à ce jour.

Bibliographie

Friedman, J., Hastie, T., and Tibshirani, R. (2001). *The elements of statistical learning*, volume 1. Springer series in statistics Springer, Berlin.

Guelman, L. et al. (2015a). Optimal personalized treatment learning models with insurance applications.

Guelman, L., Guillén, M., and Pérez-Marín, A. M. (2012). Random forests for uplift modeling : an insurance customer retention case. In *Modeling and Simulation in Engineering, Economics and Management*, pages 123–133. Springer.

Guelman, L., Guillén, M., and Pérez-Marín, A. M. (2015b). Uplift random forests. *Cybernetics and Systems*, 46(3-4) :230–248.

Langford, J. and Zhang, T. (2008). The epoch-greedy algorithm for multi-armed bandits with side information. In *Advances in neural information processing systems*, pages 817–824.

Li, L., Chu, W., Langford, J., and Schapire, R. E. (2010). A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*, pages 661–670. ACM.

Lo, V. S. (2002). The true lift model : a novel data mining approach to response modeling in database marketing. *ACM SIGKDD Explorations Newsletter*, 4(2) :78–86.

Louppe, G., Wehenkel, L., Sutera, A., and Geurts, P. (2013). Understanding variable importances in forests of randomized trees. In *Advances in Neural Information Processing Systems*, pages 431–439.

Rubin, D. B. (1974). Estimating causal effects of treatments in randomized and nonrandomized studies. *Journal of educational Psychology*, 66(5) :688.

Table des figures

1.1 Répartition du chiffre d'affaire du groupe AXA à travers le monde 3
1.2 Résultat du groupe par secteurs d'activité 4

2.1 Processus de souscription 12
2.2 Processus d'appel sortant vente 13
2.3 Conversion brute avec et sans ASV 13
2.4 Algorithme existant 15
2.5 Modèle de mise en relation 18
2.6 Catégories de clients 19
2.7 Modèle d'*uplift* 20
2.8 Exemple de calcul de NBV 23
2.9 Segments de NBV 23

3.1 Schéma simplifié de la plateforme IT 26
3.2 *Cluster* tournant sur Hadoop 27
3.3 Schéma de principe du processus *MapReduce* 28
3.4 Exemple du *wordcount* 28
3.5 Exemple de données brutes 29
3.6 *Dataflow* 32
3.7 L'interface Dataiku 32
3.8 Scnenarii de production 34
3.9 Schema du *workflow* 35

4.1 *Pipeline* des transformations 40
4.2 Importance relative des variables textuelles 41
4.3 Arbitrage biais-variance 42
4.4 fonction sigmoïde 43
4.5 Support vector machine 45
4.6 Kernel machine 46
4.7 Partitionnement binaire récursif 47
4.8 Forêt aléatoire 49
4.9 Descente de gradient : on parcourt la fonction dans le sens opposé du gradient 51
4.10 Matrice de confusion 52
4.11 Courbes ROC et précision-rappel 53

5.1 Flux de tarifs édités 58
5.2 Parts de TE par file de priorité 59

5.3 Délai médian par heure 59
5.4 Taux de communication moyen par files de priorité 60
5.5 Taux de communication par CSP 60
5.6 *uplift* par files de priorité 61
5.7 Conversions vs nombre de relances 62
5.8 Conversions vs délai de rappel 62
5.9 Conversions par canaux 63
5.10 Conversions vs rang 63
5.11 Méthodologie de calcul de la performance 64
5.13 *Grid search* pour la *Random Forest* 66
5.15 *Grid search* pour le *Gradient Boosting* 67
5.16 Importances des variables pour la *Random Forest* 67
5.17 Importances des variables pour le *Gradient Boosting* 68
5.18 Modèle d'uplift *ccif* 68

Table des matières

Résumé **iii**

Remerciements **v**

Note **vii**

Sommaire **xi**

Introduction **1**

1 Contexte **3**
1.1 Le groupe AXA : une somme d'entités 3
1.1.1 Le groupe AXA 3
1.1.2 Les entités du groupe AXA 4
1.1.3 AXA et la révolution *Big Data* 4
1.2 Le Data Innovation Lab 5
1.2.1 Présentation 5
1.2.2 Mission et défis du Data Innovation Lab 6
1.2.3 Organisation et gouvernance 7
1.2.4 Cycle de vie d'un projet 8
1.3 AXA Global Direct 8
1.3.1 Présentation 8
1.3.2 Le *Business Model* de l'assurance directe 9

2 Problématique *business* du projet *Hot Lead Management* **11**
2.1 Fonctionnement des centres d'appels Direct Assurance 11
2.1.1 Canaux de souscription 11
2.1.2 Les appels sortants vente 12
2.2 Objectifs du projet 14
2.2.1 Amélioration du score existant 14
2.2.2 Périmètre élargi 14
2.3 Comment évaluer la performance du processus ? 14
2.3.1 Limites du score métier 14
2.3.2 Indicateurs de performance du processus (KPI) 16
2.4 Premier levier : la joignabilité 17
2.5 Second levier : l'effet de la communication 18
2.5.1 Mesure de l'effet de traitement (*True Lift*) 18

2.5.2 Cadre théorique 20
2.5.3 Hétérogénéité et biais de sélection 21
2.6 Troisième levier : le concept de valeur client 22
2.6.1 Méthodologie de calcul 22
2.7 Problématique actuarielle 23
2.8 Limites liées aux processus en place 24

3 Traitement et exploration des données grâce aux outils *Big Data* 25
3.1 Contexte sur l'architecture existante 25
3.1.1 Environnement IT 25
3.1.2 Le *datalake* 25
3.2 La plateforme Hadoop et le calcul distribué 26
3.2.1 Le système de fichier Hadoop 26
3.2.2 La fonction MapReduce 27
3.2.3 Un exemple type 27
3.3 Nettoyage et agrégation des données 28
3.3.1 Le SQL distribué : Hive, Impala 28
3.3.2 L'outil Thetis 29
3.4 Données utilisées 30
3.5 *Dataflow* 31
3.5.1 Création d'une table finale 31
3.5.2 L'outil Dataiku 31
3.5.3 *Data Quality assessment* 33
3.6 Implémentations possibles du projet 34
3.7 Outils de modélisation 34
3.8 Schéma du *Workflow* 35

4 Outils de *machine learning* utilisés 37
4.1 *Preprocessing* des données 37
4.1.1 Traitement des valeurs manquantes 37
4.1.2 Traitement des variables catégorielles 38
4.1.3 Elaboration d'un *pipeline* de transformations 39
4.2 *Preprocessing* des données de navigation 39
4.3 Principe du *machine learning* 40
4.3.1 Arbitrage Biais-Variance et surapprentissage 41
4.3.2 Validation croisée 41
4.3.3 *Grid Search* 42
4.4 Modélisation de la mise en relation 42
4.5 Revue des algorithmes utilisés 43
4.5.1 Régression logistique 43
4.5.2 Support Vector Machine 44
4.5.3 Méthodes d'arbres 47
4.5.4 Méthodes de *boosting* 49
4.5.5 Performance des modèles de classification binaire 52
4.6 Modélisation de l'effet de la communication 53
4.6.1 Performance d'un modèle d'*uplift* 54
4.6.2 Estimation par différence de score 54
4.6.3 Estimation par effets croisés 54

5 Premiers résultats et impact *business* **57**
5.1 Exploration des données 57
5.1.1 Statistiques sur les TE et sur la communication 57
5.1.2 Statistiques sur la conversion 60
5.2 Evaluation de l'algorithme existant (*business case*) 63
5.3 Premiers résultats des modèles 64
5.3.1 Caractéristiques de la table finale 64
5.3.2 Modèle de mise en relation 65
5.3.3 Modèle d'*uplift* 68
5.4 *Backtesting* du modèle 69
5.5 Ouverture : utilisation de l'*active learning* 69
5.5.1 Les ASV vus comme un problème d'optimisation 69
5.5.2 Algorithme des bandits contextuels 69

Conclusion **73**

Bibliographie **75**

Table des figures **78**

Table des matières **81**