M2 Traitement de l'Information et Exploitation des Données (TRIED)

Rapport de stage de fin d'études

Optimisation du couple production risque en fonction des données web

Karim ASSAAD

Tuteur de Stage :
M^me^ Sylvie THIRIA

Encadrants :
M. Régis LOMET
M. Anas EL KHALOUI

Membres du Jury :
M^me^ Cecile MALLET
M. Anastase CHARANTONIS
M. Etienne HUOT

01/04/2017 - 29/09/2017

Résumé — L'un des objectifs de Crédit Agricole Consumer Finance est le développement d'une expertise sur les méthodologies de traitement des données provenant du Web. Exploiter ces données représente un véritable défi sur différents aspects : leur volumétrie, leur forme (données non structurées), la complexité des outils matériels et logiciels utilisés tout au long du projet (de la récolte des données jusqu'à leur traitement pour ensuite les analyser). L'élaboration d'un tel projet met ainsi en évidence deux phases nécessaires à sa mise en oeuvre. Dans un premier temps, se familiariser avec les outils en place et l'environnement, avec la terminologie des données du Web, et les méthodes permettant de les exploiter. Dans un second temps, il s'agit de préparer les données pour utiliser les méthodes étudiées. L'analyse de ces données par rapport aux différentes problématiques de l'équipe sera ainsi rendue possible.

Mots clés : Risque, Scoring, Big-Data, Weblog, Modeling, Web-Scraping, Sentiment Analysis, Text-Mining, Python, Spark, Semi-Supervised-Learning, Co-Training.

CACF
1 Rue Victor Basch
91300 Massy
France

Remerciements

Mes vifs remerciements vont aux corps administratif et enseignant du Master TRIED ainsi qu'à ceux de l'École Nationale Supérieure de l'Industrie et l'Entreprise pour les diverses disciplines grâce auxquelles j'ai pu réussir ce stage et, particulièrement, à mon tuteur, Madame Sylvie THIRIA, mentor digne de son rang, pour sa présence constante, ses recommandations enrichissantes et ses instructions fructueuses.

Je tiens aussi à remercier toute l'équipe de la direction OCF de CACF, spécialement Monsieur Régis LOMET, Responsable de la direction Octroi de Crédit et Fraude, pour m'avoir permis d'effectuer mon stage au sein de OCF et à Monsieur Anas EL KHALOUI (Chargé d'études statistiques) pour son dévouement, son sens de l'encadrement et le temps précieux qu'il m'a consacré. Grâce à leur confiance, nous avons pu nous impliquer totalement dans l'étude.

Je profite de cette occasion pour remercier du plus profond de mon cœur les membres du jury, Madame Cecile MALLET, Monsieur Anastase CHARANTONIS et Monsieur Etienne HUOT pour avoir bien voulu faire partie de l'examen du projet.

Sans oublier d'adresser mes sincères remerciements à Madame Damla CIMTOSUN, Monsieur Anis GUENINECHE et Monsieur Jean-Luc VASSENT pour la relecture du rapport et pour leurs remarques pertinentes, ainsi qu'à toutes les personnes qui m'ont été d'un énorme soutien moral et d'une grande aide durant cette brève expérience qui fut enrichissante et agréable.

Ces remerciements ne peuvent s'achever sans une pensée particulière à tous les membres de ma famille. Leur soutien et leur encouragement sont pour moi les piliers fondateurs de ce que je suis et de ce que je fais.

Table des matières

Table des figures

Liste des tableaux

Liste des sigles et acronymes

DMP	*Plateforme de Gestion d'Audience "Data Management Plate-forme"*
API	*Interface de Programmation Applicative "Application Programming Interface"*
RMSE	*Erreur Moyenne Quadratique "Root-Mean-Square Error"*
SVM	*Machine à vecteurs de support "Support Vector Machine"*
NCC	*Cluster le Plus Proche "Nearest Cluster Classifier"*
kNN	*k Plus Proches Voisins "k-Nearest Neighbors"*
NLP	*Traitement du Langage Naturel "Natural Language Processing"*

Introduction

La direction OCF a pour objectif de réduire le nombre de prêts non remboursés tout en maximisant la production. En moyenne, il faudrait 70 dossiers remboursés pour couvrir les pertes liées à un dossier non remboursé. Des indicateurs permettent d'évaluer la qualité, la probabilité de défaut d'un client. Ces indicateurs évaluent le risque de crédit. Réduire les pertes consiste donc, en grande partie, à trouver la meilleure méthodologie pour définir et calculer le risque et aussi à adopter une stratégie optimale consistant à accorder ou non un crédit à une personne. Traditionnellement, le risque est calculé à partir des données sur le client : catégorie socio-professionnelle, revenu, statut résidentiel, etc.

Une source d'information est cependant mal exploitée : le web. De manière générale, le web regorge d'informations qui peuvent être utilisées par l'entreprise : le service Marketing peut se servir des données provenant des réseaux sociaux sur lesquels est connectée une grande partie de la population. Ainsi, on pourrait par exemple analyser le comportement et le sentiment des clients pour établir une campagne marketing ciblée. Nous pourrions également analyser les données que laisse un client sur le web pour détecter les tentatives de fraudes, les incohérences rencontrées en croisant les informations provenant de diverses sources. Nous pourrions aussi utiliser ces nouvelles données pour tenter d'améliorer le modèle de score. Auparavant, les ressources matérielles ne permettaient pas d'exploiter une si grande volumétrie de données. L'arrivée de nouvelles technologies matérielles et logicielles capables de traiter toutes ces informations en un temps raisonnable rend les questions suivantes légitimes :

— Est-ce que les données provenantes du web permettent d'augmenter de manière non négligeable les performances de l'entreprise ?
— Quels sont le coût et le temps nécessaires à la mise en place des technologies requises ?
— Quel est le coût lié à la formation du personnel à l'utilisation de ces technologies et à l'exploitation de données de type complètement différent ?

Le stage permet d'éclairer ces questions, sans nécessairement y répondre directement. On se concentre principalement sur l'exploration et l'exploitation de ces données, on apporte également une approche de calcul du risque de crédit en se basant sur les données de log de navigation web et de définir son comportement.

Le présent rapport s'étale sur trois chapitres. On l'aborde par un premier chapitre consacré à mettre le projet dans son cadre général par une présentation de l'organisme d'accueil, des métiers et des enjeux de ce dernier et une problématique pour mettre en exergue le contexte et les objectifs de la mission. Le deuxième chapitre fait l'objet d'une étude approfondie des données et des méthodes mises en œuvre pour leurs l'enrichissement. Le troisième chapitre s'intéresse à détailler les algorithmes de modélisation pour déterminer le pré-score. Enfin, le dernier chapitre présente la conclusion générale qui inclut les différentes perspectives du stage

Chapitre 1

Contexte

On entame ce premier chapitre par une brève présentation de l'entreprise au sein de laquelle on a conçu et développé la globalité du projet de fin d'études, ainsi que la définition de quelques notions nécessaires à la clarté de la stratégie développée. Le chapitre propose aussi une critique des méthodes existantes ainsi que l'environnement logiciel utilisé pour la mise en place de la solution envisagée.

1.1 Organisme d'accueil

Crédit Agricole Consumer Finance (CACF) est une filiale du Crédit Agricole, spécialisée dans le crédit à la consommation. Elle est un acteur majeur dans ce domaine au niveau européen avec 9540 collaborateurs dont 3400 en France, elle est ainsi présente dans 21 pays du monde. La stratégie de Crédit Agricole Consumer Finance est construite autour de la qualité de la relation client. Les enjeux de l'entreprise sont les suivants :

— Permettre aux clients de financer leurs projets en proposant une large gamme de produits (prêts automobile, travaux, prêts personnels . . .)
— Permettre une distribution multicanale : vente directe via les agences, e-commerce, partenariats et courtage pour atteindre la plus large clientèle possible.
— Investir dans des méthodes innovantes pour la simplification de l'expérience client et améliorer l'efficacité opérationnelle.

1.1.1 Direction Octroi de Crédit et Fraude

Le stage s'est déroulé dans la direction Octroi de Crédit et Fraude (OCF). Rattachée à la Direction Développement Crédit France, l'entité Octroi de Crédit et Fraude est en charge de la maîtrise du risque d'impayé et de fraude à l'octroi de crédit pour l'ensemble

des marchés. Le périmètre de la direction s'étend à deux types de circuits principaux : le circuit long qui correspond à l'octroi de crédits aux clients d'entreprises extérieurs (CACF agit alors comme fournisseur de service) et le circuit court chargé des clients directs de CACF. L'entité OCF se doit d'assurer la maîtrise du risque au regard des objectifs fixés par la Direction Générale, en passant par des politiques d'acceptation et de gestion de la fraude. C'est dans ce contexte que la mission du stage prend son sens, elle consiste à développer des méthodes sûres et nouvelles de gestion du risque et de crédit par le biais de scores et de plusieurs procédures bancaires, en s'assurant que ces méthodes respectent les finalités commerciales de l'entreprise.

1.2 Cadre du stage

La mission principale de l'équipe dans laquelle s'est déroulée le stage consiste à mettre en place des stratégies d'acceptation à travers des outils de décision d'octroi de crédit pour la marque commerciale Sofinco et pour les partenaires bancaires (Crédit Agricole et LCL). D'autre part, elle doit proposer des méthodes innovantes pour le calcul des scores, l'implémentation des modèles statistiques, les processus d'acceptation et les méthodes de détection de la fraude. L'objectif principal consiste à créer et maintenir un équilibre entre le niveau du risque et la production.

1.2.1 Objectif

CACF dispose de plusieurs méthodes pour estimer le risque, qui sont essentiellement basées sur les données renseignés par des clients et de leur historique de transactions et d'actions. L'équipe a vu de l'intérêt dans l'exploitation des données indépendant aux biais de la véracité afin de mettre en place une nouvelle méthode d'estimation du risque par un pré-score. Cela se fait par le biais des données provenant du web et plus précisément les données provenant de la DMP, destinée à stocker les données décrivant les actions des visiteurs du site web Sofinco [Dongre 2015]. La figure 1.1 schématise la nouvelle stratégie que nous expérimentons.

Traditionnellement, la construction de scores est basée sur les données déclaratives des clients sans vérification préalable de la véracité des informations reçues. En effet, quelques données peuvent être fausses ou inexactes, et auront une influence directe sur l'estimation du risque. La solution que l'on propose se fonde sur l'utilisation de données qui proviennent de la navigation internet et qui sont récoltées grâce aux cookies. Chaque site web met en place des cookies qui récupère des informations sur les internautes de passage. Ils permettent entre autre d'identifier l'appareil (aussi appelé "device") avec lequel l'internaute est connecté. Ainsi les cookies permettent de rattacher des données de navigation à un client dont le risque est connu. On peut donc construire des bases de données d'apprentis-

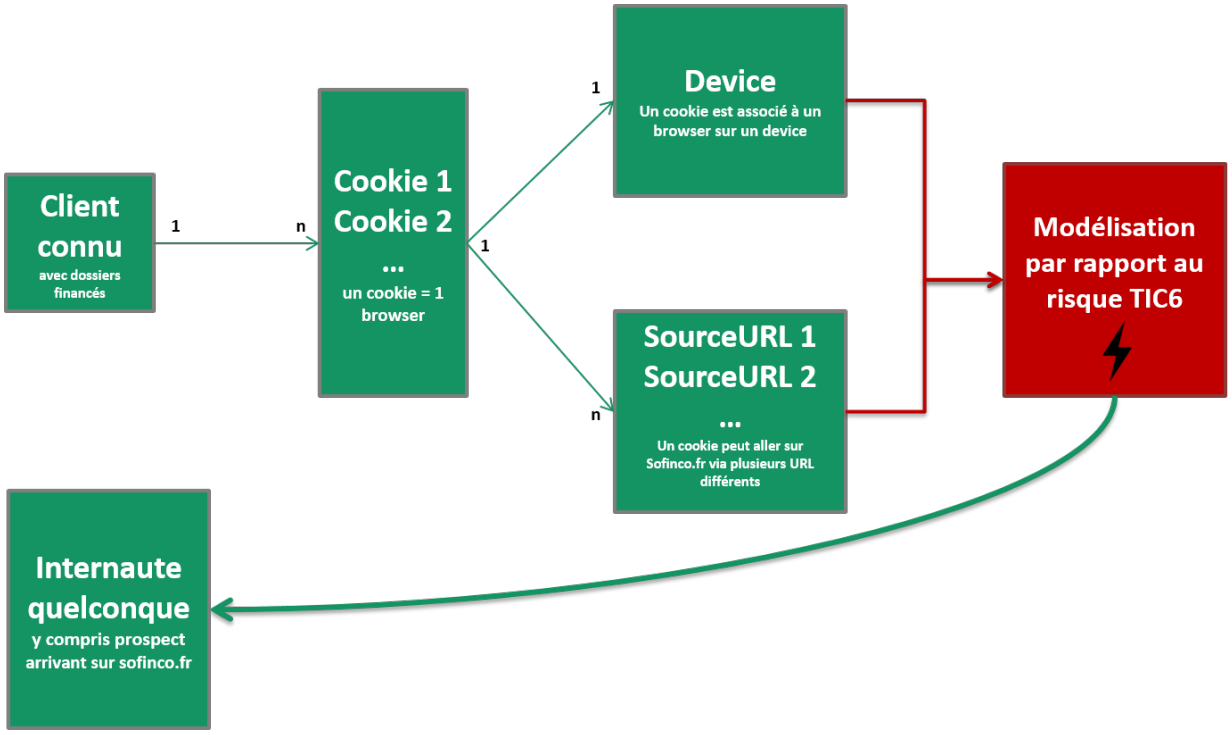

FIGURE 1.1 – Usage à l'octroi

sage pour la modélisation. Grâce à cela, il devient possible de pré-scorer des internautes quelconques qui peuvent être des potentiels clients pour appliquer la stratégie.

1.2.2 Environnement de travail

La réalisation des modèles de calcul a nécessité plusieurs logiciels et langages de programmation.

Spark : C'est un outil distribué, dédié au traitement rapide des données volumineuses, grâces aux mécanismes de parallélisme et *Map Reduce* qu'il propose [MapR 2017]. Il est donc indispensable pour exploiter la richesse des données disponibles sur la DMP. L'accès à Spark se fait par le biais d'une interface web nommée *HUE* facile à utiliser, mais elle permet de travailler sur un seul noeud du cluster à la fois. On travaille donc plutôt sur le terminal pour avoir accès à toute la puissance de l'outil. L'une des principales tâches durant le nouveau processus a été d'adapter le code Spark existant et le mettre à jour pour qu'il soit exploité sous sa nouvelle version utilisant les DataFrame au lieu des RDD (Resilient Distributed Dataset). Cette nouvelle structure de données de Spark est en effet plus adaptée à nos besoins [Karau 2013]. Le tableau 1.1 fait l'objet d'une petite comparaison, pour justifier le choix des Dataframe au regard des RDD.

Les restrictions qui sont en place dans le cluster pour des raisons de sécurité des données

RDD	Dataframe
Ne peut être accédé pour modification	Modifiable à souhait
Ne peut pas être optimisées	Optimisation automatique des opérations
Lenteur à l'usage	Rapidité d'utilisation

TABLE 1.1 – Rdd Vs Dataframe

rendait impossible d'accéder à internet pour chercher des données provenant du web ou même de télécharger des librairies, ce qui complexifie le déroulement du travail.

Python : C'est le langage utilisé pour compléter les manipulations non réalisables sur le cluster, il est très riche en librairies et convenable pour le traitement des données.

Hive : C'est un système d'entrepôt de données pour Hadoop. Il est utilisé pour la gestion des grands ensembles de données, il propose des fonctionnalités de requête, de sauvegarde et de synthèse des données.

1.3 État de l'art

1.3.1 La variable cible

Pour la création de la variable cible qui représente le risque d'octroi de crédit, il a fallu extraire plusieurs informations stockées dans l'entrepôt de données SAS de l'entreprise. En effet les données extraites nous procurent les dossiers de souscription de chaque client, un client pouvant posséder plusieurs dossiers. Si l'un d'entre eux est tombé en impayé, le client en question sera considéré comme risqué.

Cette stratégie est assez stricte et rigide, elle ne prend pas en considération les imprévus de la vie, si l'on prend l'exemple d'un bon client dont le dossier est passé en impayé à cause d'un imprévu, le client sera identifié comme risqué alors que ce n'est pas le cas, mais cela ne pose pas de vrai problème car le risque calculé par cette méthode est destiné à servir un processus de pré-score intermédiaire et non exclusif dont le but est seulement d'identifier en amont les profils les plus susceptibles d'être risqués. Ce pré-score constitue une couche supplémentaire dans la gestion du risque et ne remplacera pas les scores et les règles déjà en place. La donnée de risque qui servira pour le pré-score est donc une variable binaire qui prend la valeur 1 si un client donné a eu au moins un dossier impayé et 0 sinon.

1.3.2 Variables de la DMP

Les variables explicatives qui composent le jeu de données sont issues de la DMP et sont originalement fournis par la société prestataire *1000mercis* qui s'occupe de la collecte des données générées par les tags déposés sur les pages du site web de l'entreprise. On va utiliser principalement deux jeux de données de la DMP que sont "Sofinco" et "Media" et qui peuvent être joints à travers une clé unique de session.

Le premier jeu de données contient l'ensemble des appels de tags collectés sur le site *sofinco.fr*, le site mobile *m.sofinco.fr*, les parcours sur la vente déportée et les remontées des devis acceptés ou en attente des comparateurs de prêts. Le deuxième jeu de données contient des informations sur les bannières publicitaires indiquant le site de provenances des clients.

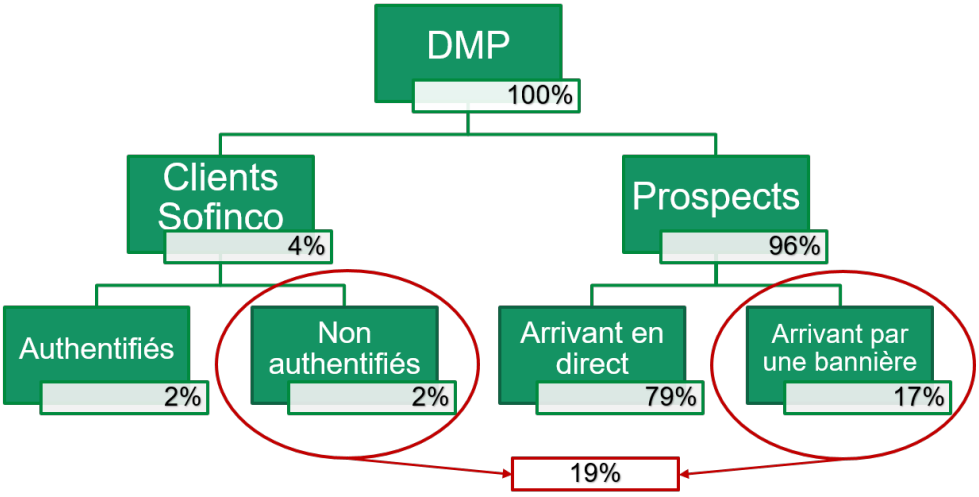

FIGURE 1.2 – Répartition de la DMP

1.4 Étude descriptive

Durant la phase d'analyse, le nettoyage et le remplacement des valeurs manquantes est primordial, en effet le jeu de données dispose de plusieurs variables contenant un nombre important de valeurs manquantes allant jusqu'à 97%.

Le jeu de données n'est pas structuré et manque d'informations. Entre la faible variation des variables et le nombre important de données manquantes, il est devenu nécessaire d'enrichir la base de données par des informations extérieures.

Nous disposons par ailleurs, dans les données, de variables avec un taux de complétude très élevé (Moins de 3% de données manquantes) :

— L'URL de la page sur laquelle se trouve l'utilisateur.
— L'ID de cookie qui permet de reconnaître de manière unique les utilisateurs et qui sert dans les jointures.
— Le User-Agent qui contient les informations sur l'équipement logiciel (Navigateur et OS avec informations de version) et le matériel (Appareil, modèle) de l'internaute.
— L'URL du site contenant la bannière publicitaire sur laquelle a cliqué l'internaute et qui l'a amené sur le site Sofinco.

Conclusion

D'après les analyses descriptives, il est clair que les données à disposition ne sont pas suffisantes pour fournir une bonne explication de la variable cible. De ce fait, de nouvelles variables sont requises pour pouvoir implémenter des modèles prédictifs utiles. La première piste sera d'utiliser la variable **User Agent** potentiellement exploitable pour la modélisation dans la mesure où elle donne des informations sur l'équipement du client. De même, il sera intéressant d'utiliser les URL des bannières publicitaires pour la création de nouvelles variables renseignant les fréquentations virtuelles de la clientèle.

Nous avons mis au point une démarche bien précise. Elle prendra en compte les différentes étapes, décrites ci-dessous, qui permettront à la fin d'alimenter et de qualifier le jeu de données.

— L'exploration des données de navigations web en vue de les utiliser dans la modélisation.
— L'exploitation de nouvelles sources de données qui ne sont pas nécessairement structurées et de méthodes de fouille de données.
— La construction d'un modèle de pré-score qui servira à identifier les internautes prospects qui représentent un risque potentiel.
— La recherche d'une méthode de validation adéquate au problème et l'approbation du pré-score afin de créer un prototype qui sera mis en production. En fonction de la performance du prototype, on pourra vérifier si les données DMP/WEB ont un véritable intérêt pour la modélisation du risque, et comment on peut les utiliser.

Chapitre 2

Traitement et Exploitation

La détection automatique des clients de la banque ainsi que l'enrichissement des données et leur prétraitement fera l'objet de ce chapitre. Les différentes méthodes d'alimentation du jeu de données seront aussi exposés d'une manière détaillée.

2.1 Reconnaissance des clients non authentifiés

Le premier travail consiste en une reconnaissance des clients non authentifiés grâce aux cookies. Le tableau[2.1] représente un affichage à titre d'exemple permettant de relier les clients en fonction de leurs ID dans la base de la banque avec leurs cookies et les appareils qu'ils utilisent lors de leurs connexion. De cette façon, on arrive à détecter les clients qui accèdent au site de la banque même s'ils ne sont pas authentifiés au préalable et on peut leurs associer directement leurs risques bancaires.

Client	Cookie	Appareil
Client A	Cookie 1	iPhone S5-IOS 10-Safari 15
Client A	Cookie 2	Tablette Samsung Galaxy-Andoid 6.0.1-Mozilla firefox 9
Client A	Cookie 3	Ordinateur portable HP-Windows 10-IE 10
Client B	Cookie 1	MacBook Pro-MacOS X Sierra-Chrome 9.2
Client B	Cookie 2	Sony Xperia Z4-Android 7.1-Chrome 5

TABLE 2.1 – Exemple du tableau référant aux cookies et aux ID clients

Cette méthode offre l'avantage de répliquer le taux de clients identifiés par deux.

2.2 Variable User Agent

Le User-Agent est un élément central des réseaux web et joue un rôle important dans la reconnaissance des utilisateurs du réseau. Cette variable permet, par exemple, à un site internet de reconnaître les robots de moteurs de recherche ou encore d'identifier le système d'exploitation et le navigateur utilisés par une machine s'y connectant.

La variable **User Agent** est la seule variable exploitable pour la modélisation du risque, elle comprend des données sous format *Json* et demande un traitement d'extraction d'informations et de décodage pour pouvoir en créer de nouvelles variables intelligibles. A l'aide des données contenues dans **User Agent**, nous sommes parvenus à discerner plusieurs informations relatives aux périphériques utilisées par les utilisateurs, le système d'exploitation ainsi que le navigateur. A première vue, les variables tirées ne présentent pas un lien direct avec le risque d'octroi de crédit et ne seront peut-être pas très bénéfiques pour l'étape de la modélisation. De ce fait, le besoin de trouver de nouvelles sources d'informations vient se confirmer. De même, **User Agent** permet de détecter les robots qui accèdent au site, mais cela ne sera pas très pertinent dans un premier temps car l'apprentissage sera focalisé sur des clients réels de la banque. Dans un deuxième temps, on pourra exploiter la détection des robots pour prévenir contre la prédiction du risque sur ces robots et par la suite optimiser les ressources. Le modèle sera par la suite appliqué sur de vrais internautes qui pourraient être de futurs clients.

2.2.1 Analyse tri à plat

Variable device : D'après les analyses de fréquence par rapport au risque pour chaque variable, la majorité des profils classés comme risqués correspondent à des périphériques inconnus. En effet, la variable périphérique est qualitative avec 18 modalités et 90% des individus sont classés comme utilisant des périphériques inconnus donc les valeurs de cette variable ne seront pas très intéressantes pour la prédiction du risque.

Variable système d'exploitation : On remarque que les individus sont répartis entre deux classes dominantes *Windows 8* et *Windows 7*.

Variable browser : L'analyse de cette variable débouche sur une constatation intéressante, les individus identifiés comme risqués utilisent des navigateurs internet majoritairement *Google Chrome* ou *Firefox*, par contre ceux qui utilisent *Internet Explorer* ne sont pas considérés comme risqués. D'un autre côté, les classes *Firefox* et *Chrome* correspondent aussi à des individus non risqués avec un pourcentage de 99%, donc le fait d'avoir des individus risqués éparpillé sur deux classes ne permet pas tout seul de déduire le comportement de ces individus. Il faudra compléter ces variables par des informations plus discriminantes.

2.3 Extraction des données du web

Les données ne sont pas significatives donc, on va utiliser les URL des bannières d'où les clients sont venus au site Sofinco pour étudier leurs profils. L'idée est d'associer à chaque session des données web en utilisant l'URL de la bannière associée, puis combiner les sessions d'un même utilisateur afin d'obtenir un résultat qui pourra nous donner une vison de son profil.

2.3.1 Web scrapping

Le web scrapping est une technique d'extraction de données à partir des contenus disponibles sur les sites web, cette opération s'effectue par le biais d'un script et a pour objectif l'exploitation des données extraites et leur réutilisation dans un autre contexte. Dans ce cas, il s'agit d'alimenter le jeu de données initial et de l'enrichir par des informations supplémentaires [Severance 2017].

On a employé le site Alexa qui repose sur des techniques d'analyses avancées de trafic, ce qui permet de le considérer comme site fiable et source riche en information. Il fournit des études statistiques importantes sur les URLs des sites donnés en entrée : le classement en terme de nombre de visites, l'audience sociodémographique, la qualité du site en terme de vitesse, les sites semblables, l'engagement des visiteurs envers le site et plusieurs indicateurs sur la confiance générale des visiteurs envers le site.

Pour récupérer des données pertinentes, on a implémenté une librairie de web scrapping qui fait l'extraction variable par variable et utilise différentes méthodes selon la nature de la donnée recherchée. Le processus est assez délicat, il faut d'abord récupérer le contenu du site web sous format *HTML* et le nettoyer en éliminant les commentaires et les fonctions *Java Script* et *CSS*. L'étape suivante consiste à utiliser les expressions régulières pour identifier les bonnes balises et obtenir la donnée spécifique voulue.

Les données sont très importantes pour qualifier le comportement des clients de la banque, en effet, connaître le type de site fréquenté par les clients renseigne sur les tendances clients et leurs niveaux de confiance. Le jeu de données initial est constitué de sessions de plusieurs clients. De ce fait, un client apparaît dans plusieurs lignes en tant qu'utilisateur de différentes sessions. En effet, cette notion de session garantit qu'on aura, à la fin, plusieurs sites et l'ensemble de leurs URLs permettra de qualifier les clients et d'identifier leurs profils. De cette façon, on garantit que le résultat final ne sera pas affecté et que le profil d'un client ne sera pas déterminé par la qualité d'un seul site.

Afin d'effectuer le web scrapping, il faut respecter quelques réglementations imposées par les sites scrappés. Cela permet de lever toute illégalité sur cette activité et garder la confidentialité des utilisateurs des sites. D'un autre côté, le code doit s'adapter aux contraintes des serveurs pour ne pas les saturer et garder une performance élevée. Le

respect de ces conditions est coûteux en terme de temps, en effet, elles engendrent des arrêts aléatoires du scrappeur qui marque des pauses suivies de redémarrages de manière aléatoire. Cela est dû au fait que le scrappeur ne doit pas produire de pattern, au risque d'être banni.

2.3.2 Extraction par API

Tout comme le web scrapping, l'extraction par API permet de récolter des données web, mais d'une manière plus élaborée et guidée par les créateurs des sites, les données seront présentées sous forme d'un fichier log prêtes à être utilisées. Le site utilisé par cette méthode est MyWot qui permet de fournir deux informations clés qui sont la confiance associée au site ainsi que sa sécurité pour les enfants.

2.3.3 Nouvelles variables

Grâce aux méthodes mentionnées précédemment, on a pu créer plusieurs nouvelles variables pour enrichir le jeu de données.

- rank : Le classement mondial du site en terme de nombre de visites.
- country : Le pays majoritaire des visiteurs.
- rank country : Le classement dans le pays majoritaire en terme de nombre de visites.
- keywords : Les mots-clés de recherche qui envoient le trafic vers ce site.
- keywords percent : Le pourcentage de trafic de recherche attribué à chaque mot clé.
- bounce rate : Le taux de rebond du site qui présente l'engagement des visiteurs envers un certain site. Plus il est élevé, moins les utilisateurs sont engagés.
- total sites linking : Le total des sites qui pointent vers le site.
- sites linking : Les 5 sites pointant vers le site qui génèrent le plus de trafic.
- upstream sites : Les sites visités par les utilisateurs avant d'accéder au site en question.
- related sites : Les sites similaires par chevauchement d'audience à ce site.
- similar sites : Les sites avec des noms similaires au site.
- categories : Les thèmes associés aux mots clés.
- male : Un score de similitude du public de ce site avec la population générale des internautes en terme de proportion d'hommes.
- female : Un score de similitude du public de ce site avec la population générale des internautes en terme de proportion de femmes.
- no college : Un score de similitude du public de ce site avec la population générale des internautes en terme de proportion d'éducation supérieure terminée.
- some college : Un score de similitude du public de ce site avec la population générale des internautes en terme de proportion d'éducation supérieure entamée.

— graduate school : Un score de similitude du public de ce site avec la population générale des internautes en terme d'éducation secondaire.
— college : Un score de similitude du public de ce site avec la population générale des internautes en terme d'accès depuis une institution supérieure.
— home : Un score de similitude du public de ce site avec la population générale des internautes en terme d'accès depuis le domicile.
— school : Un score de similitude du public de ce site avec la population générale des internautes en terme d'accès depuis une école.
— work : Un score de similitude du public de ce site avec la population générale des internautes en terme d'accès depuis un lieu de travail.
— child safety : Un taux qui indique la sûreté du site pour les enfants.
— trust worthiness : Un taux qui présente la fiabilité du site.

Il y a, en plus, une variable qui indique le taux de confiance associé à chacune des variables sociodémographiques et des variables de sécurité, ce qui permet d'obtenir au final un total de 34 variables explicatives.

2.3.4 Imputation des valeurs manquantes

Le pré-traitement des données est essentiel pour la suite du travail, il est constitué d'une partie remplacement de valeurs manquantes et d'une partie transformation de variables. A première vue, notre jeu de données présente plusieurs valeurs manquantes, certaines colonnes comme **catégories**, **similar sites**, **no college**, **school** contiennent plus de 50% de données manquantes qu'il a fallu les supprimer. De même, les individus qui présentent plusieurs valeurs manquantes en terme de variables (plus de 50%) seront éliminés du jeu de données.

On a réussi à scrapper 29584 URLs en tenant compte des données manquantes et de leur suppression, le nombre d'URLs a baissé de 33%. Néanmoins, certaines variables supprimées seront récupérées par le biais de variables appelées **related sites** et qui seront expliquées par la suite. A l'issue de ces traitements, le taux maximal de données manquantes par colonne est de 22%. De plus, on supprime les individus qui ont des données manquantes sur des variables non imputables comme le **rank** et **bounce rate**. On peut ainsi mettre en place des méthodes d'imputation pour compléter le jeu de données : on s'intéresse au reste des variables, et on va expliquer leur imputation cas par cas.

Variable pays

Pour pouvoir remplir les cases manquantes de la variable **country**, on a utilisé la librairie "*whois*" qui permet d'avoir le pays d'origine du site. En effet, "*whois*" envoie une requête sur le serveur hébergeant le site web pour recueillir des données relatives au domaine

visité tel que le code du pays, variable qui nous intéresse ainsi que d'autres informations comme le propriétaire, les contacts associés, les statuts et les prestataires.

A l'aide d'une deuxième libraire "*pycountry*" comportant les dictionnaire d'extensions de nom de domaines, on arrive à identifier les codes pays extraits dans l'étape précédente et les faire correspondre aux pays adéquats. On s'assure de remplacer les valeurs manquantes par le nom complet des pays au lieu de mettre l'indicatif pays pour des raisons d'homogénéité de la base. En effet, la variable est de type chaîne de caractère et indique le nom des pays.

De cette façon, on arrive à remplir les valeurs vides sans avoir recours à des méthodes de prédiction, néanmoins cette méthode laisse place à une petite marge d'erreur car les valeurs initiales de pays sont prises par rapport à l'origine des visiteurs et non par rapport au site lui-même.

Variable rang national

Pour remplacer les valeurs manquantes de cette variable, on a créé une méthode dédiée schématisée par la figure [2.1]. On commence par chercher le pays d'origine de l'URL du site ainsi que le rang mondial du site. Pour le pays en question, on recherche les rangs mondiaux des sites de ce pays, on compare le rang mondial du site voulu avec les valeurs des rangs mondiaux des sites de son pays. Pour l'URL avec le rang mondial le plus proche et inférieur à celui de notre site, on incrémente de 1 son rang national et on l'associe à la valeur manquante. Ainsi, on remplace les valeurs manquantes avec une approche qui garantit l'ordre des rangs au sein d'un même pays, mais qui peut avoir un intervalle d'erreur en terme de précision. Si on prenait un exemple simple : 3 sites d'un même pays ayant consécutivement les rangs mondiaux 8 , 11 , 15 et le rang national du premier et troisième site sont 5 et 10, il est évident qu'on pourra classer le rang du deuxième site dans le même sens en le mettant à un rang 6 donc supérieur au premier rang et inférieur aux troisième rang national, mais cette valeur n'est pas tout à fait exacte, en effet le rang national se trouve certainement dans l'intervalle 5-10 mais peut prendre n'importe quelle valeur continue de cet intervalle.

Imputation par la variable la plus corrélée

Pour remplacer les valeurs manquantes dans les variables qui présentent une forte corrélation avec d'autres, on impute chacune des variables à partir de l'autre grâce à un modèle polynomial. Pour rappel, les valeurs traitées (**male-female** et **home-work**) ne sont pas complémentaires pour un site donné, mais elles présentent quand même de forts coefficients de corrélation.

Le coefficient de corrélation entre les variables **male** et **female** est égale à -0.95, On

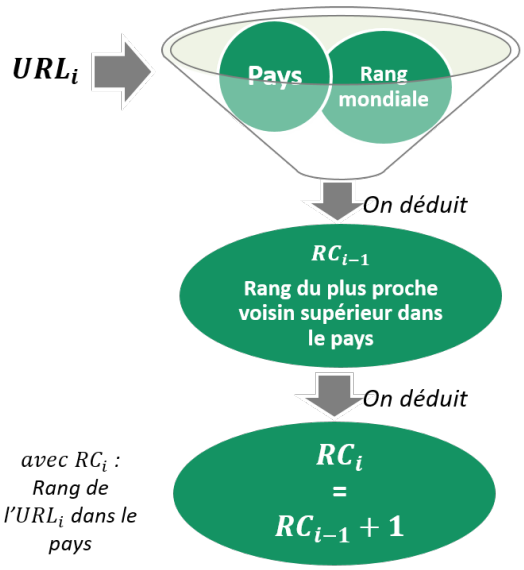

FIGURE 2.1 – Procédure de calcul des données manquantes du rang national

effectue donc un modèle de régression sur les scores de ces variables en précisant une borne qui est égale à (-50) comme présenté sur la courbe à gauche de la figure[2.2], si une valeur tombe en dessous de (-50) on lui affectera automatiquement (100). C'est un modèle polynomial de degré 3, et a un RMSE de 1.8 sur un intervalle $\left[-100, 100\right]$. Le modèle inverse ayant les mêmes paramètres admet une RMSE réciproque de 1.16.

Le coefficient de corrélation pour les variables **maison** et **travail** atteint –0.86. Le modèle qui va servir pour l'imputation est polynomial de degré 3 avec une borne aux alentours de (-70) comme le démontre la courbe à droite de la figure[2.2]. Le RMSE pour ce modèle est de 5.8 sur un intervalle de $\left[-100, 100\right]$ et le RMSE réciproque du modèle inverse est de 9.46.

Variables de confiance

Pour traiter les données manquantes dans les valeurs de confiance qui sont associées à d'autres variables, on a effectué une transformation en changeant leurs modalités en variables binaires. Cette méthode permet de détecter les variables qui sont rattachées aux taux de confiance et dont la valeur a été imputée. En effet, pour améliorer la performance du modèle dans le futur, il serait intéressant de diminuer le poids des variables aux valeurs imputées. Pour mieux illustrer le traitement effectué sur les variables de confiance, on prendra une variable avec les modalités confiance élevée, confiance moyenne et pas de confiance, qu'on transforme en trois variables binaires. Dans le cas où ces trois variables

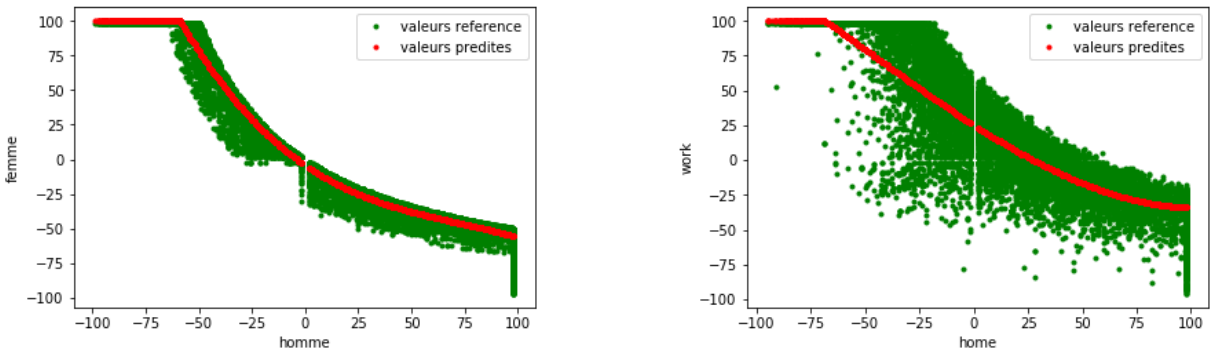

FIGURE 2.2 – Régression polynomiale entre homme-femme et maison-travail

sont à zéro, cela veut dire que l'URL rattachée a une valeur imputée.

Imputation par arbre de décision des variables issues de MyWot

Pour les variables MyWOT, on utilise des arbres de décision pour l'imputation des valeurs. La profondeur des arbres est optimisée. L'erreur du modèle utilisé pour la prédiction des valeurs manquantes de **child safety** est de 0.27, tandis que pour la variable **trust worthisiness**, l'erreur est de 0.2.

Imputation par kNN des variables sociodémographiques

Pour les variables sociodémographiques, on utilise le modèle kNN pour l'imputation des valeurs. La validation croisée a permis de choisir un k=49. L'erreur pour les variables **male**, **female**, **graduate school**, **college**, **home** et **work** sont consécutivement de 1, 1.4, 0.8, 1.3, 0.9 et 1.1.

Traitement supplémentaires

Une étape importante à réaliser est de vérifier par une analyse descriptive que les méthodes d'imputation de données n'ont pas modifié ou biaisé la structure initiale des données. A priori, presque toutes les variables ,sauf quelque unes, ont gardé la même distribution en terme de moyenne et de variance à très peu de choses près. On peut détecter une différence de 0.9 dans la variance qui passe de 27.2 à 28.1.

Les variables qui subissent une différence remarquable sont **rank**, **rank country** et **total sites liking**. Il faut noter que les valeurs manquantes des variables **rang** et **total sites linking** n'ont pas été imputées et que, pour le cas de la variable **rank country**, elles ont été traitées selon un algorithme spécifique figure[2.1]. De ce fait, on arrive à la déduction que la différence de variance vient certainement du fait qu'on a éliminé au début

du processus des individus qui ont plus que 50% de leurs variables en données manquantes. La différence au niveau de la variance reste quand même négligeable pour ces variables aux alentours de 2, on passe par exemple de 52 à 54.

On effectue une différence en valeur absolue des données imputées et des données initiales et on les visualise sur la carte thermique de la figure[2.3]. On remarque qu'il n'y a pas de différence de structure entre les données modifiées et celles de départ, en effet la corrélation est restée inchangée. La différence maximale de corrélation est la même, elle est entre **bounce rate** et **rank** de l'ordre de 0.2, puis on trouve que la corrélation entre **rank** et **child safety** de 0.14. La valeur de corrélation pour les variables **bounce rate** et **child safety** est moins importante mais non négligeable, aux alentours de 0.09. Les variables **rank** et **rank country** ont une corrélation de 0.07 tandis que la corrélation entre **bounce rate** et **trust worthiness** est de 0.06. Entre **bounce rate** et **childe safety**, on trouve une corrélation de 0.05, suivie par une corrélation à l'ordre de 0.04 entre **rank** et **trust worthiness**. La corrélation entre les couples de variables **childe safety** et **trust worthiness**, **child safety** et **child safety confidence**, **rank** et **trust worthiness** est de 0.03. Les paires de variables restantes admettent une corrélation négligeable inférieure à 0.02.

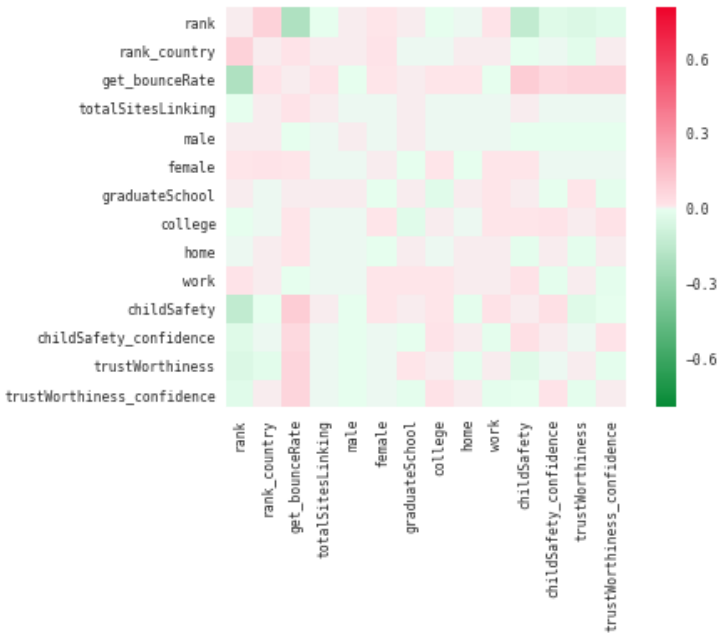

FIGURE 2.3 – Différence entre les cartes thermique des données avant et après le nettoyage

2.4 Analyse des sentiments

L'analyse des sentiments reste une science inexacte, en effet le langage humain est très complexe et l'apprendre à une machine est une tâche assez délicate. Le processus inclut l'analyse des nuances grammaticales et/ou culturelles ainsi que l'argot et les fautes d'orthographe qui sont très fréquents. De plus, enseigner à une machine l'influence que peut engendrer le contexte sur la tonalité est encore plus difficile. D'autre part, la communication humaine ne peut pas se limiter à quelques catégories pour la caractériser, certes elle peut être positive, négative et neutre mais la notion de sentiment va au-delà de ces critères et peut s'avérer bien plus complexe [Nasukawa 2003].

Mais avant de s'intéresser à la partie analyse des sentiments, il est essentiel d'effectuer un traitement du langage naturel et de le nettoyer [Bird 2009] afin de pouvoir l'exploiter et dégager les sentiments des site des bannières. Cette procédure est représentée par la figure[2.4].

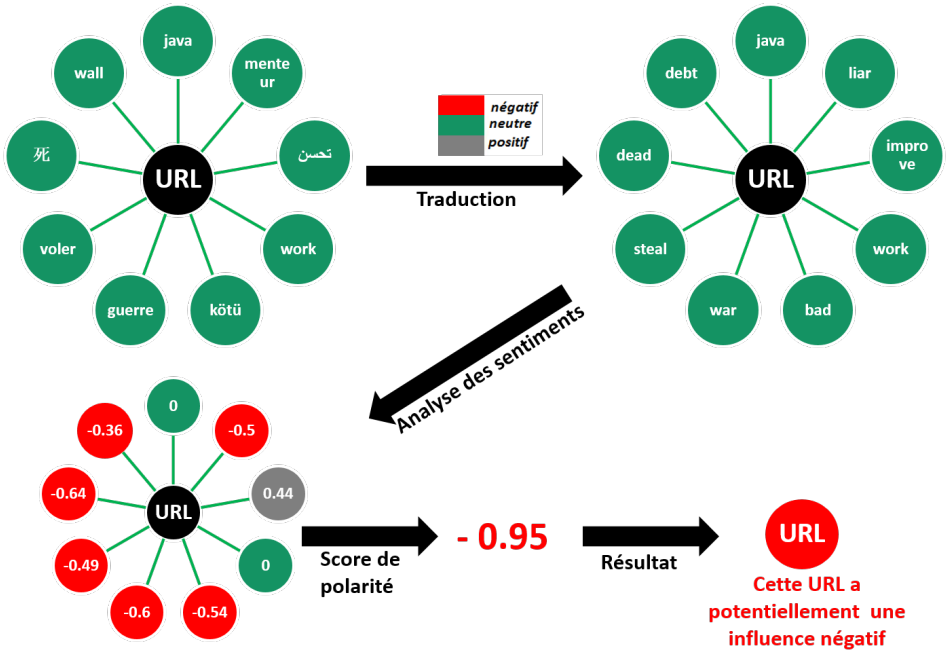

FIGURE 2.4 – Procédure NLP

2.4.1 Traitement du langage naturel

Les données sont fournies en plusieurs langues, donc un travail de traduction est nécessaire. Pour ce faire, on a créé un script exécutable qui se charge de la traduction en anglais de tous les textes collectés peu importe la langue. Le choix de l'anglais est justifié par le

faite que la librairie de NLP la plus réputée (*NLTK*) utilise des dictionnaires d'anglais dans le traitement du texte. L'API utilisée pour la traduction est celle de *Microsoft Bing*. Un des obstacles rencontrés au cours de ce processus est la présence des caractères spéciaux, en effet la version 2 de *Python* transforme les données à traduire en un codage qui traite les caractères spéciaux en les remplaçant par des caractères *utf-8*, de ce fait les mots à traduire sont difficilement repérables, il a donc fallu basculer vers *Python 3* qui arrive correctement à gérer ce genre de caractères.

Le nettoyage des textes bruts a nécessité les étapes suivantes :
— Élimination de la ponctuation.
— Élimination des mots qui contiennent des chiffres.
— Élimination des mots de lisaison et assimilés.
— Convertir en lettres minuscules.
— Lemmatisation.
— Élimination des mots qui commencent par ‘.
— Élimination des mots qui contiennent moins que 4 lettres.
— Élimination des mots qui contiennent plus que 14 lettres.
— Élimination des mots qui contient des caractères spéciaux.
— Association des URL qui n'ont pas de mots clés à ‘Unknown’.

2.4.2 Variable sentiment

Tous les traitements effectués ont permis de créer une nouvelle variable sentiment qui est issue du calcul d'un indicateur de polarité grâce à un dictionnaire d'anglais. Cette variable qui représente le sentiment d'une URL donnée prend des valeurs comprise entre -1 et 1 ou l'intervalle $[-1,-0.25]$ réfère aux sentiments négatifs, l'intervalle $]-0.25, 0.25[$ présente les sentiments neutres et l'intervalle $[0.25, 1]$ présente les sentiments positifs.

2.5 Classification selon les mots clés

La première étape de la classification des mots clés consiste en la création de la matrice *IF-IDF*. En effet, cette méthode est largement utilisée pour la fouille de textes. C'est une technique qui permet d'afficher le nombre d'occurrences d'un mot donné contenu dans un document texte et d'évaluer ainsi son importance. Cette méthode se base sur une pondération des mots, plus le mot est fréquent et plus son poids augmente.

Pour la classification des URL, on ne considérera que les mots qui se répètent au moins deux fois, cela nous a permis de réduire le nombre de mots de 42037 à 17088. Même avec l'élimination des mots sans doublons, la matrice est assez volumineuse et compte 19172 lignes et 17088 colonnes, il était donc impossible d'effectuer une réduction de dimension

par le biais d'une analyse des composantes principales, de calculer la matrice de covariance ou même de faire un clustering.

Pour arriver à une solution, il fallait utiliser une méthode de factorisation pour traiter la matrice. Pour ce faire, on a employé la méthode *Sparse Random Projection* [Li 2006] qui consiste à multiplier la matrice initiale A de dimension (n x D) avec une matrice R(D x k) avec k<D. La matrice R contient les valeurs -1, 0, 1 avec des probabilités égales à

$$\frac{1}{2\sqrt{D}} \quad ; \quad 1 - \frac{1}{\sqrt{D}} \quad ; \quad \frac{1}{2\sqrt{D}}$$

qui étaient initialement $\frac{1}{6}, \frac{2}{3}, \frac{1}{6}$, mais on a utilisé $\sqrt{D}$ pour la rapidité contre une petite perte d'information. Le résultat obtenu à l'issue de cette technique nous a permis de transformer la matrice de dimension (19172, 17088) en une matrice (19172, 8452) et de réduire ainsi considérablement le nombre de variables. On peut maintenant appliquer l'*ACP* et réduire encore plus les dimensions pour avoir à la fin une matrice de taille (19172, 3900). La figure[2.5] permet de visualiser le pourcentages d'informations fournies par les axes, en effet 3900 dimensions expliquent 100% de l'information.

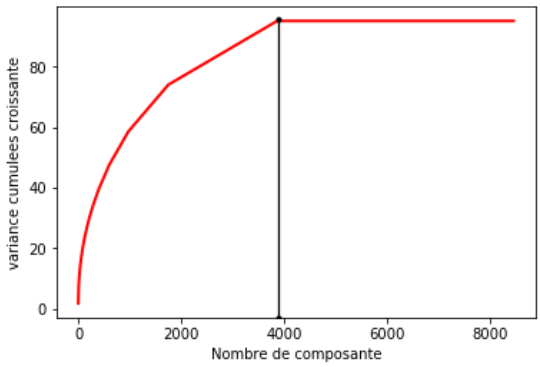

FIGURE 2.5 – Choix du nombre de composantes principales

A présent, on pourra effectuer la classification des URL dont le critère sera le nombre d'occurrence des mots.

2.5.1 Variable segmentation par mots clés

Tous les traitements effectués ont permis de créer une nouvelle variable qui segmente les URLs par rapport à leurs mots clés.

D'après la figure[2.6] on voit par lecture graphique que 75 classes semblent être un choix optimal, la courbe admet en effet une allure qui descend exponentiellement et qui se stabilise aux alentours de 75. La méthode se base sur des calculs de distances et de variances intra-classes.

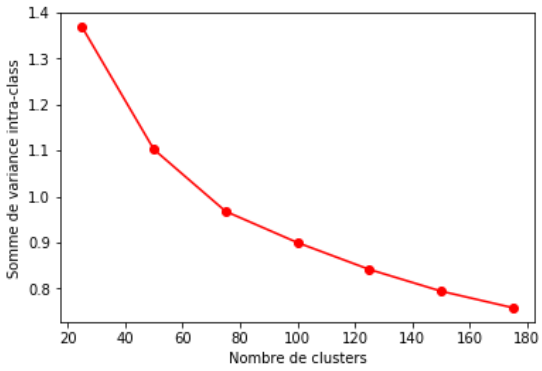

FIGURE 2.6 – Choix du nombre de clusters

Par la suite, on applique un apprentissage non supervisé *les k-moyennes* pour segmenter les URLs.

2.6 Enrichissement de la base

Pour avoir des données plus volumineuses, on a utilisé les données de *Alexa* et *myWot*, spécifiquement les URL des sites reliés à la variable **related sites**. Les URLs similaires qui sont rattachés à plusieurs URL du jeu de données seront traitées selon la méthode suivante : Pour chaque site similaire, on identifie toutes les URLs de la base qui lui correspondent. Pour remplir ces données qualitatives, on prend le maximum de fréquence des données qualitatives des URLs reliées selon la formule suivante :

$$Y_j = \max X_{.j}$$

Pour remplir ces données quantitatives, on calcule les moyennes des variables quantitatives des sites reliés selon la formule qui suit :

$$Y_j = \frac{\sum_{i=1}^{n} X_{ij}}{n}$$

Cette technique a permis de passer de 19 000 URLs à 90 000 URLs. La figure[2.7] illustre le principe de la méthode, si l'on suppose que l'on a deux URL 1 et URL 2, qui sont reliées à des URLs similaires dont l'une est l'URL lambda qui se répète pour les deux URLs 1 et 2. Pour pouvoir créer les données relatives au site lambda, il suffit de s'intéresser à l'ensemble des URL auxquelles il est rattaché et d'appliquer le traitement nécessaire selon la nature des variables.

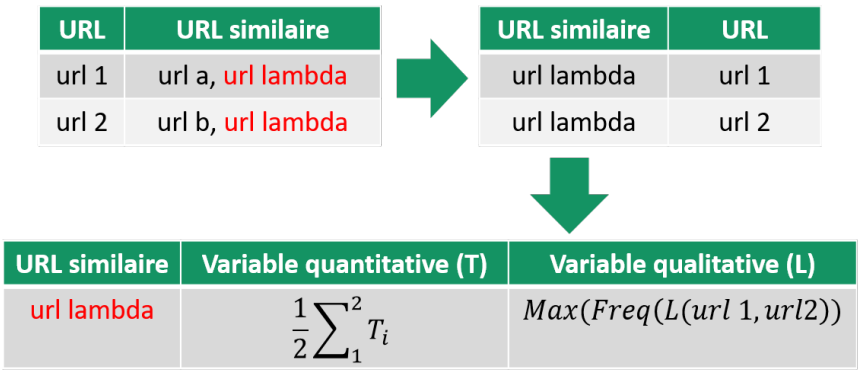

FIGURE 2.7 – Exemple d'enrichissement des données

2.7 Profiling

Afin de dresser le profil de fréquentation web d'un client donné, on utilisera l'ensemble des URLs auxquels a accédé ce client. Pour chaque internaute, les informations provenant des URLs qu'il a visités seront combinées selon l'exemple de la figure[2.8]. Dans notre jeu de données initial, chaque observation présente une session, et un internaute peut avoir une ou plusieurs sessions. Pour ce faire, on utilise le même processus employé dans l'enrichissement des données, cela veut dire qu'on combine les données disponibles à partir des sessions et on les utilise pour obtenir le profil de l'internaute. Cela sert à qualifier les internautes moyennant des informations obtenues de leurs sessions

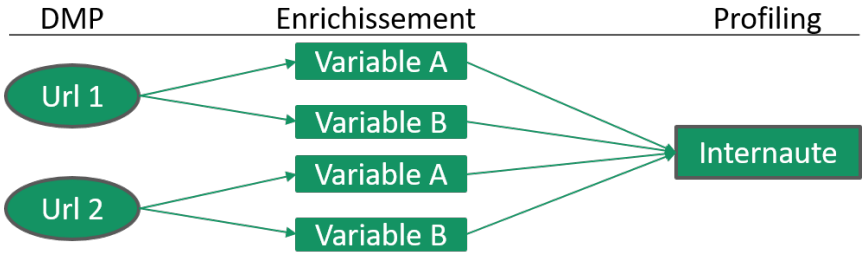

FIGURE 2.8 – Exemple de profiling à partir des URLs

Conclusion

Cette partie consistait principalement à enrichir les données DMP à partir du web et à compléter les valeurs manquantes des nouvelles variables ainsi que d'à ajouter de l'information supplémentaire par le biais de plusieurs analyses linguistiques et méthodes calculatoires. Maintenant, on a un jeu de données complet et donc, on peut passer à la phase de modélisation.

Chapitre 3

Modélisation

Une fois la phase d'enrichissement et de nettoyage des données établies, on passe à l'implémentation des modèles prédictifs. Ce chapitre s'intéresse donc à la modélisation en terme d'apprentissage automatique par des méthodes classiques et d'autres innovantes.

3.1 Algorithmes utilisés

Régression Logistique : La *régression logistique* est très répandue dans plusieurs domaines. C'est un modèle très utilisé pour le calcul des scores bancaires.

NCC : Le principe de l'algorithme du *NCC* est inspiré du celui du *kNN* traditionnel, l'idée principale est de classer des observations non étiquetées en fonction de l'étiquette de leurs voisins le plus proche. En obtenant un certain nombre de partitions en utilisant un algorithme de clustering simple, l'algorithme *NCC* extrait un grand nombre de clusters hors des partitions. Ensuite, l'étiquette de chaque centre de cluster est déterminée en utilisant le mécanisme de vote majoritaire entre les étiquettes des observations dans le cluster. Le *NCC* est en quelque sorte un *kNN* où les voisins sont des centres de clusters plutôt que des individus. *NCC* est environ k fois plus rapide que *kNN* [Parvin 2012].

SVM : Les *SVM* sont une généralisation des classifieurs linéaires. Les séparateurs à vaste marge ont été développés dans les années 1990 à partir des considérations théoriques de Vladimir Vapnik sur le développement d'une théorie statistique de l'apprentissage : la théorie de Vapnik-Chervonenkis. Ils ont rapidement été adoptés pour leur capacité à travailler avec des données de grande dimension, le faible nombre d'hyper-paramètres, leurs garanties théoriques, et leurs bons résultats en pratique [Vapnik 1999].

Forêts aléatoires : Cet algorithme combine les concepts de sous-espaces aléatoires et de *Bagging*. L'algorithme des forêts d'arbres décisionnels effectue un apprentissage sur de multiples *arbres de décision* entraînés sur des sous-ensembles de données légèrement

différents qui prennent en considération différents échantillons et variables. La décision finale sera un vote majoritaire des différents *arbres de décisions* entraînés.

3.2 Méthode de validation

Pour mieux exploiter les données dans l'apprentissage, on a utilisé la méthode de validation croisée et plus spécifiquement la méthode *k-fold cross-validation* [Biernat 2014].

Pour pouvoir mieux expliquer les modèles et faire la différenciation entre leurs modes de fonctionnement, il est très important de choisir la métrique la plus adéquate. La performance générale du modèle se calcule selon le pourcentage du nombre d'individus correctement prédit par rapport au nombre total d'individus prédit. Dans ce cas, cette mesure de performance n'a pas trop de sens, elle n'indique pas réellement la qualité des modèles, en effet les profils non risqués sont très nombreux et largement en avantage par rapport aux profils risqués, ce qui fait qu'on a toujours une très bonne performance puisque l'algorithme arrive facilement à identifier les individus non risqués. On déduit qu'il n'est probablement pas très pertinent d'utiliser la performance comme méthode d'évaluation des algorithmes.

D'autre part il s'avère que la précision et le rappel sont des mesures plus significatives puisqu'elles se focalisent principalement sur la prédiction des profils risqués. On rappelle que la précision et le rappel ont pour expression les formules suivantes :

$$Precision = \frac{VP}{VP + FP} \qquad Rappel = \frac{VP}{VP + VN}$$

Où VP signifie les vrais positifs qui réfèrent aux individus risqués, FP signifie les faux positifs et VN les vrais négatifs.

Dans le but de simplifier l'interprétation de la performance, on utilisera plutôt une seule mesure qui est une moyenne harmonique de la précision et du rappel. Cette mesure s'appelle le *F-score* ou F1.

$$F - score = \frac{2 * Precision * Rappel}{Precision + Rappel}$$

3.3 Modélisation du risque

Une fois les nouvelles variables explicatives prétraitées, il faut les injecter dans les algorithmes de prédiction pour voir si elles ont un effet positif sur la performance ou s'il aurait fallu exploiter seulement la variable **User Agent**. D'après les résultats et les comparaisons, on a remarqué que l'utilisation de la variable **User Agent** seule ne permet pas d'identifier les profils risqués, par contre l'ajout des nouvelles données a aidé à bien

classifier les profils à risque et non risqués. Concernant les erreurs de généralisation et d'apprentissage, on en est venu à la conclusion que tous les algorithmes ont réalisé un progrès sur l'ensemble d'apprentissage. Néanmoins, l'algorithme de *NCC* se démarque en validation. Il est passé à un *F-score* égale à 0.11.

	Régression Logistique	SVM	Forêts aléatoires	NCC
F1 de validation	0	0	0.08	0.11
F1 d'apprentissage	0.06	0.5	0.8	0.15

TABLE 3.1 – F-score en fonction des différents algorithmes

Le tableau[3.1] montre que les algorithmes de *SVM* et de *Forêts aléatoires* sont en sur-apprentissage, en effet le *F-score* pour le *SVM* pour l'échantillon d'apprentissage est de 0.5 tandis que le *F-score* de validation est de 0. De même, le modèle de *Forêts aléatoires* a un *F-score* de 0.8 pour l'apprentissage et 0.08 pour l'ensemble de validation.

Pour essayer de venir à bout de ce problème, on envisage de mettre en œuvre deux méthodes d'amélioration des résultats. La première consiste à faire une optimisation des paramètres du modèle et la deuxième consiste en une réduction de nombre de variables. Ces techniques seront appliquées sur tous les algorithmes même ceux qui ne subissent pas de sur-apprentissage dans le but d'améliorer leurs performances.

3.3.1 Optimisation des paramètres

Dans cette partie, on va optimiser les paramètres des différents modèles à l'aide de la validation croisée.

Régression logistique

On modifie le seuil de probabilité (0.5 par défaut) en essayant plusieurs valeurs, le résultat obtenu est légèrement meilleur. La courbe à gauche de la figure[3.1] démontre la performance en fonction des poids affectés pour les classes risquée et non risquée. Ces poids varient entre 0 et 1 et sont complémentaires, si par exemple on fixe le poids de la classe 0 à 0.3, le poids de la classe 1 sera de 0.7. On remarque que la courbe de validation est stable entre les poids 0.1 et 0.2 pour la classe non risquée et puis elle décroît au-delà de ces valeurs. En contrepartie, la courbe d'apprentissage est croissante entre 0.1 et 0.2 pour la classe non risquée puis elle décroît pour les valeurs supérieures. Pour stabiliser le modèle d'une façon plus efficace, on va considérer la valeur de validation maximale qui correspond à la valeur d'apprentissage minimale, donc les poids consécutifs seront de 0.9 et 0.1 pour les risqués et non risqués.

Pour régulariser le paramètre λ, on fait varier λ entre 0.5 et 30. Les deux courbes de la partie droite de la figure[3.1] sont fluctuantes et irrégulières avec de grand pics. La valeur choisie est de 1.5.

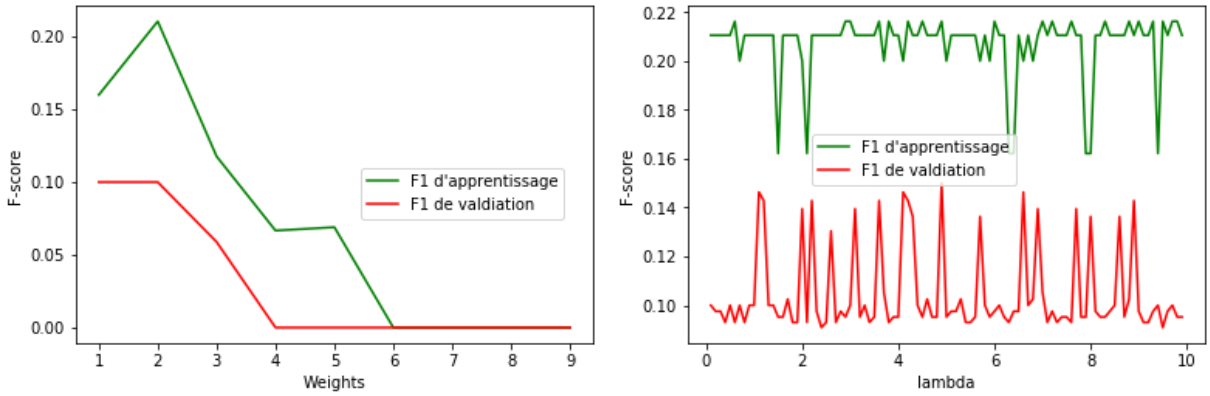

FIGURE 3.1 – Paramétrage de la régression logistique

Après ces modifications, le *F-score* de validation atteint les 0.1 et celui de l'apprentissage atteint les 0.16.

SVM

Il est clair que les solutions de ce problème ne sont pas naturellement séparables, donc on écarte le choix du noyau linéaire dès le début et on utilise un noyau de type *Gaussien radial* :

$$K(x, x') = e^{-\sigma||x-x'||^2}$$

.

Afin de réduire la sensibilité de *SVM* à la distribution de la classe de formation, on varie le paramètre coût [Forman 2007]. On commence par répliquer le protocole d'expérience complet du *coût* allant de 0.5 à 50 avec un pas de 1. Au-delà de ces valeurs, la performance s'est considérablement dégradée. D'après la courbe à gauche de la figure[3.2], on remarque que plus on augmente la valeur du *coût* plus la performance augmente, mais on n'arrive toujours pas à éliminer le problème de sur-apprentissage, on choisit toutefois un *coût* égal à 41, cette valeur est la meilleure obtenue sur l'ensemble de validation qui donne un *F-score* égal à 0.11.

Ayant fixé le paramètre *coût* à 41, on fait le même processus pour le paramètre *gamma*. Selon la courbe de la figure[3.2] à droite, on remarque qu'une petite augmentation de *gamma* améliore le *F-score* de validation tandis qu'une importante augmentation dans la valeur de *gamma* ne fait que dégrader la performance. D'après la matrice de confusion une valeur élevée de *gamma* correspond à une variance faible. Ceci implique que le vecteur de

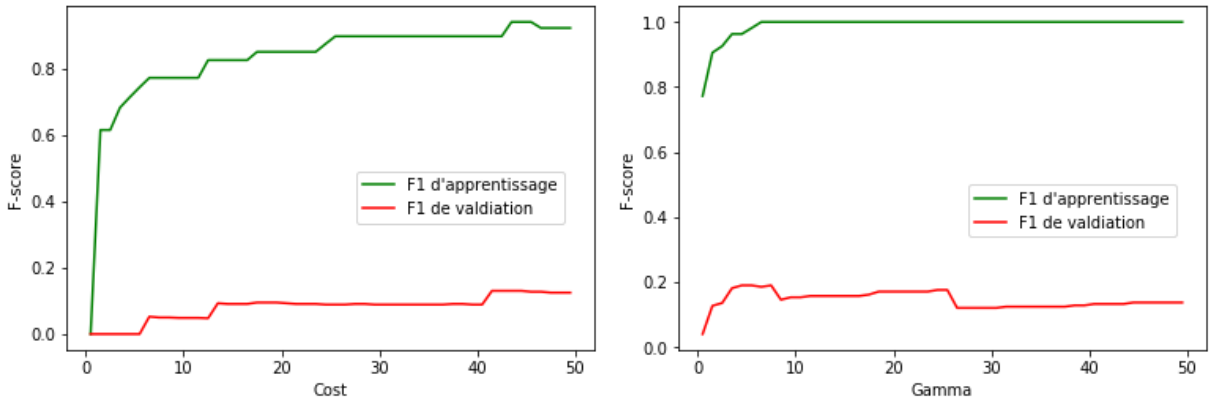

FIGURE 3.2 – Paramétrage du SVM

support n'a pas d'influence étendue. Au final, on n'a pas obtenu un *gamma* optimal qui permet de réduire le sur-apprentissage. On choisit quand même de faire un compromis entre une performance assez élevée et un sur-apprentissage moins important avec un *gamma* égal à 9.

Ces différents paramétrages ont permis d'améliorer légèrement les résultats obtenus sur l'échantillon de validation, en effet le *F-score* se stabilise à 0.19. Pour l'ensemble d'apprentissage, le *F-score* est de 0.99, par suite le problème du sur-apprentissage n'est pas résolu.

Forêts aléatoires

Un plus grand nombre d'arbres donne de meilleures performances, mais rend l'exécution plus lente. Afin d'améliorer la robustesse du modèle tout en tenant compte de la capacité du processeur on a pu élever le nombre d'arbres à 64.

Selon la courbe de gauche en figure[3.3], on déduit que les poids qui permettent de réaliser le balance entre le sur-apprentissage et la performance sont aux alentours de 0.2 et 0.8, le *F-score* obtenu pour un tel paramétrage est de 0.2 pour la validation et de 0.81 pour l'ensemble d'apprentissage.

De même, on démontre selon la courbe à droite de la figure [3.3] que l'augmentation du *F-score* suit l'augmentation de la valeur de la *profondeur des arbres* et le sur-apprentissage, donc on choisit une valeur égale à 13 qui permet d'avoir un *F-score* de 0.22 pour la validation et un *F-score* de 0.79 pour l'échantillon d'apprentissage.

Dans ce cas, on réussit à diminuer l'effet de sur-apprentissage et à améliorer la robustesse du modèle.

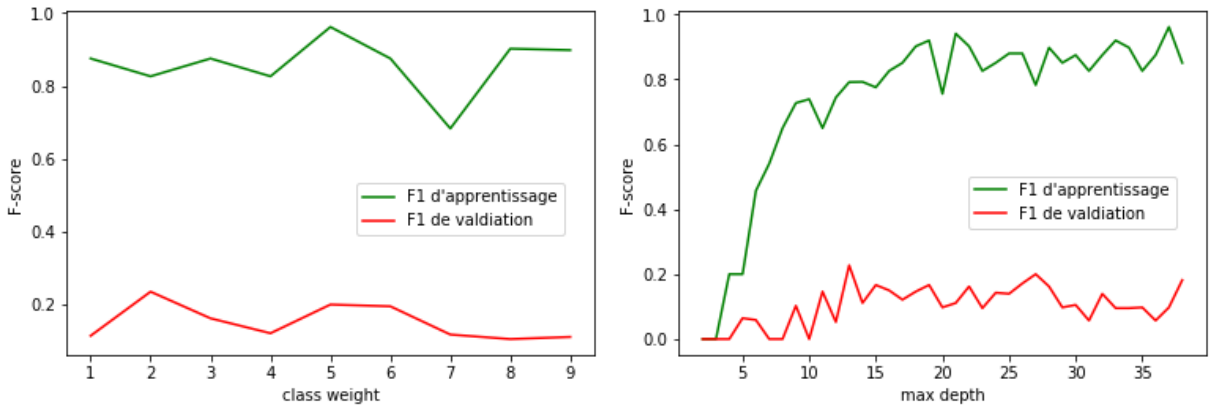

FIGURE 3.3 – Paramétrage des forêts aléatoires

NCC

La figure[3.4] met en exergue le *F-score* obtenu pour différents types de distances. On remarque que les distances *sqeuclidean*, *minkowski* et *chebyshev* sont les plus robustes. On choisit d'utiliser la distance de *chebyshev*, car c'est la plus simple. Entre deux points donnés A et B, de coordonnées respectives $(A_0, \ldots, A_n)$ et $(B_0, \ldots, B_n)$, la distance de *chebychev* est définie par :

$d(A, B) = \max_{i \in [[0,n]]} (|A_i - B_i|)$. $d(A, B) = \max_{i \in [[0,n]]} (|A_i - B_i|)$.

Autrement dit : c'est la distance associée à la norme « infinie ».

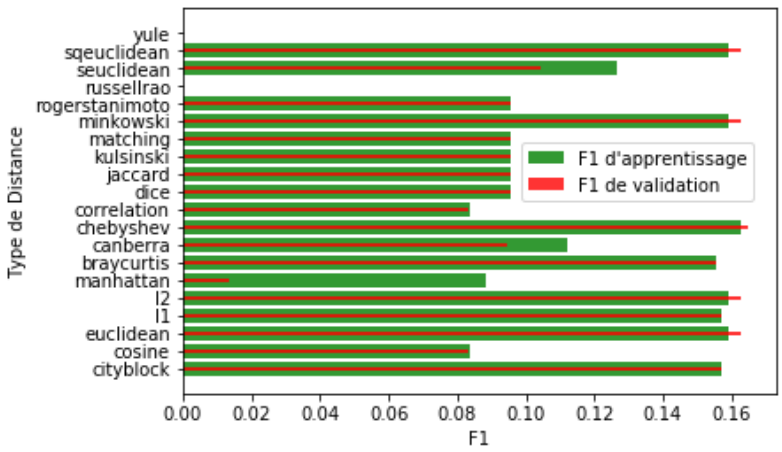

FIGURE 3.4 – Paramétrage du NCC

Après ce changement le F-score de validation augmente jusqu'à 0.162 et le *F-score* d'apprentissage devient 0.16.

	Régression Logistique	SVM	Forêts aléatoires	NCC
F1 de validation	0.1	0.19	0.22	0.16
F1 d'apprentissage	0.16	0.99	0.79	0.16

TABLE 3.2 – Impact de l'optimisation des paramètres sur le F-score

3.3.2 Réduction du nombre de variables

La réduction du nombre de variables est importante pour diminuer le sur-apprentissage et elle sera effectuée de deux manière différentes. La première méthode consiste à utiliser un apprentissage non supervisé pour combiner les variables quantitatives et la deuxième méthode consiste en une sélection de variables en fonction de leur importance.

Clustering des variable quantitatives

Une méthode de clustering sera appliquée principalement sur les variables quantitatives afin de les segmenter et de les remplacer par une seule variable. On commence par appliquer une analyse des composantes principales sur le sous échantillon composés des variables quantitatives. La figure[3.5] des variances cumulées démontre que 5 dimensions suffissent pour expliquer 70% de l'information.

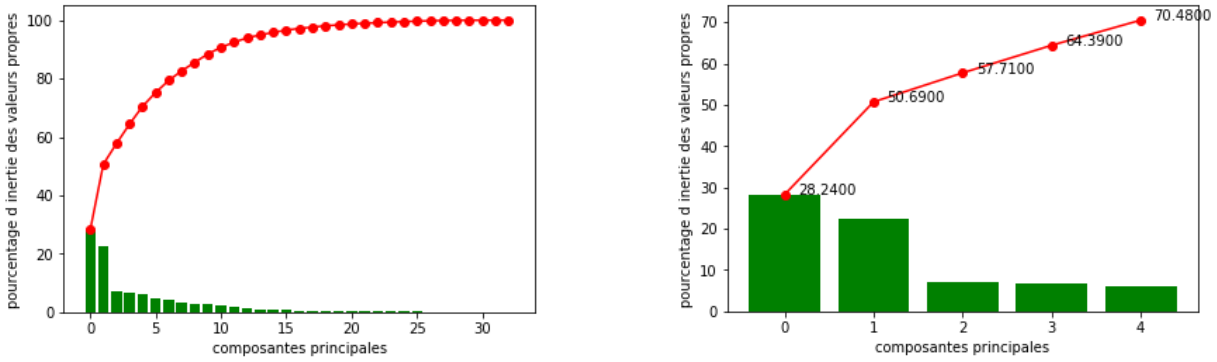

FIGURE 3.5 – Choix des dimensions

La deuxième étape consiste à exploiter les résultats retournés par l'analyse factorielle et à les injecter dans un algorithme de k-Means. La méthode *elbow* qui figure sur la courbe [3.6] démontre que le choix de 11 classes est très raisonnable, en effet cette technique permet de calculer les distances entre les individus ainsi que leur variance intra-classe.

Cette méthode a diminué le sur-apprentissage, mais elle a également dégradé la performance, elle est donc à rejeter.

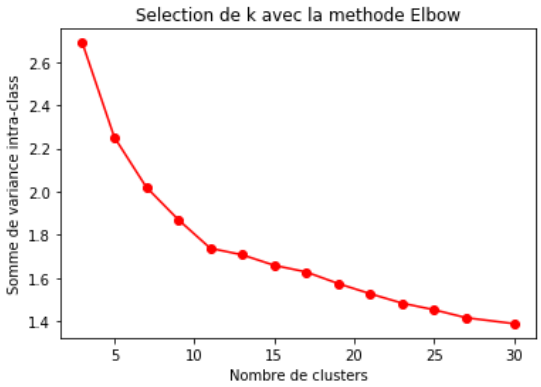

FIGURE 3.6 – Méthode Elbow pour le choix du nombre de cluster

Sélection des variables

La deuxième technique qui a été utilisé est la sélection des variables par le biais d'un modèle qui fournit un critère d'importance de variables. On utilise donc le modèle des *forêts aléatoires* qui s'est démarqué des autres modèles par sa performance et qui permet de choisir les variables les plus pertinentes grâce au critère de l'indice de *Gini*. Les variables dont les importances sont inférieures à la moyenne sont éliminées.

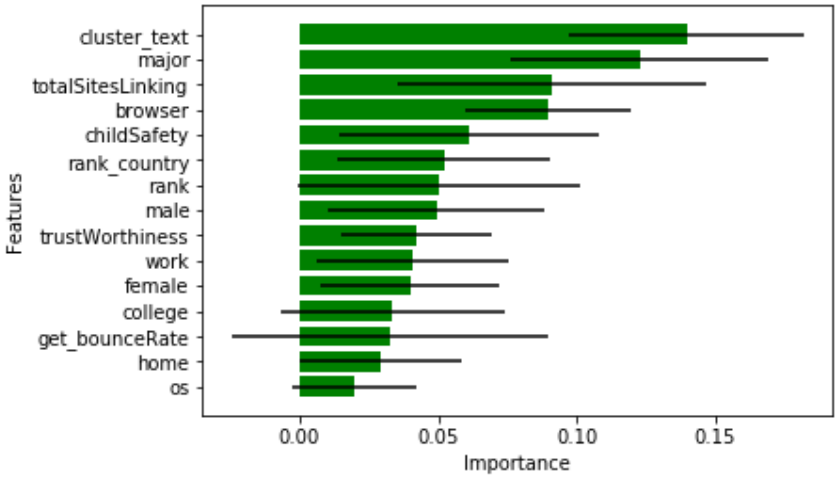

FIGURE 3.7 – Importance des variables

On a réduit le nombre de variable à 15. D'après la figure[3.7] on conclut que les variables les plus significatives sont :

— **cluster text** qui est issue de la classification des mots clés.
— **major** qui fait partie des variables de **User Agent** et qui signifie le version du navigateur.

Cette sélection a permis de modifier les résultats des différents algorithmes. Pour la *régression logistique*, pour le *NCC* et pour le *SVM* les *F-scores* se dégradent. En contrepartie, le *F-score* selon les *forêts aléatoires* s'est amélioré. Pour l'ensemble de validation, il est de 0.25 et de 0.74 pour l'échantillon d'apprentissage.

On arrive donc à la conclusion que le *NCC* est le meilleur en terme de stabilité mais l'algorithme des forets aléatoires se démarque d'un point de vue de performance même s'il n'élimine pas totalement le sur-apprentissage.

3.4 Co-apprentissage

Le *co-apprentissage* est une technique d'apprentissage *semi-supervisée* qui nécessite deux types de données. Elle se base sur une supposition qui énonce que chaque élément est décrit en utilisant deux méthodes de fonctionnalités différentes. Elles fournissent des informations différentes et complémentaires sur l'instance. Le co-apprentissage apprend d'abord d'un classifieur distinct pour chaque vue contenant des exemples étiquetés. Les prédictions les plus fiables de chaque classificateur sur les données non étiquetées sont ensuite utilisées pour construire itérativement des données [Zhu 2008].

Dans notre cas, on prendra deux échantillons différents qui serviront à l'apprentissage. Le premier échantillon contient 50% d'individus non risqués et 50% d'individus risqués. Pour ce qui en est du deuxième, il y a deux cas envisageables qu'on traitera ultérieurement. Pour chaque échantillon, la modélisation sera faite avec un classifieur différent.

Le principe de base de l'algorithme est inspirée de l'idée de (Blum et Mitchell, 1998) qui supposes que les variables peuvent être divisées en deux ensembles ; les deux ensembles sont conditionnellement indépendants, compte tenu de la classe. Initialement, deux classifieurs distincts sont formés avec les données marquées. Chaque classifieur classe les données non marquées et enseigne l'autre classifieur avec les quelques exemples non étiquetés et ainsi ce processus se répète. On va utiliser cette méthodologie, mais d'une manière moins stricte comme l'ont suggérée Goldman et Zhou (2000). De cette façon on évitera les fortes hypothèses sur la répartition des fonctionnalités.

D'autre part, on ajoutera une étape intitulée les modèles d'initialisation où la première itération aura ses propres modèles, puis à partir de la deuxième itération d'autres modèles seront introduits et le processus de la méthode co-apprentissage de Goldman and Zhou (2000) se poursuivra automatiquement.

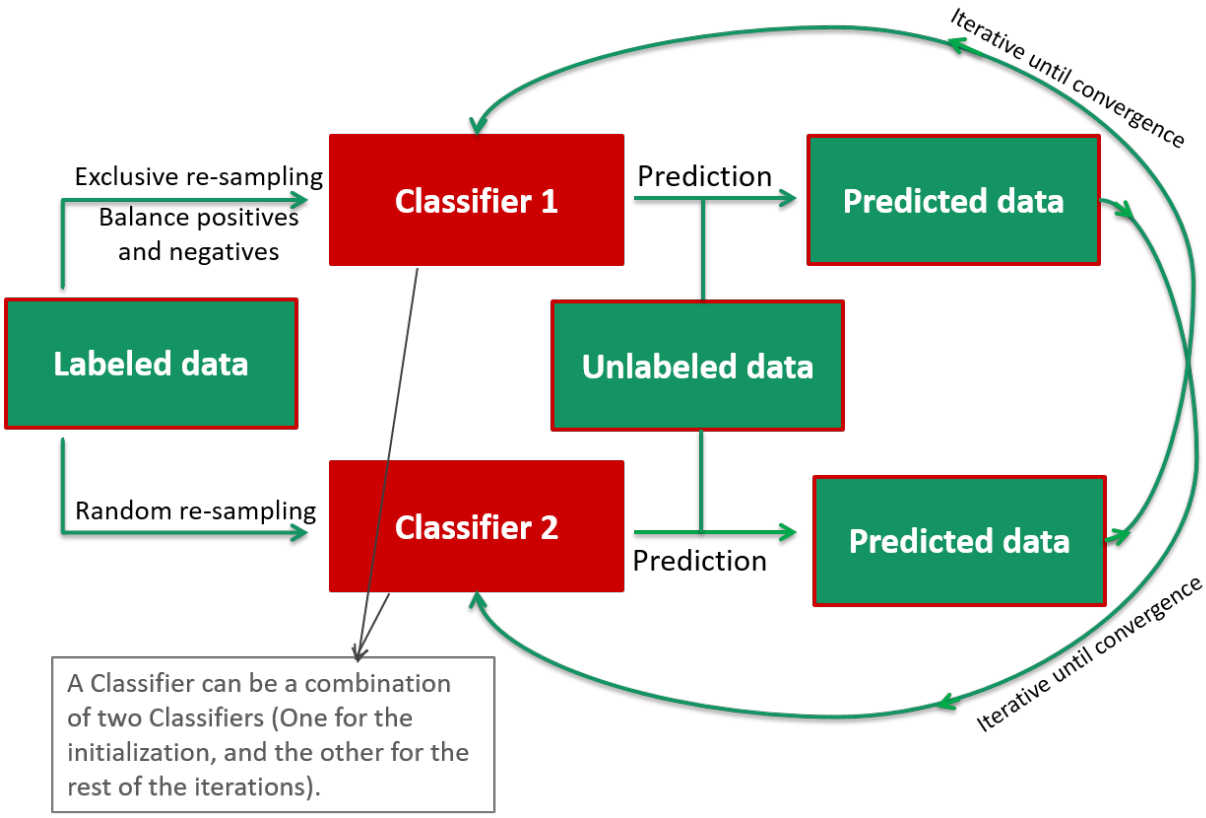

FIGURE 3.8 – Procédure du co-apprentissage

3.4.1 Choix des classifieurs

Classifieur 1

L'échantillon des clients de la banque n'est pas proportionnellement distribué. En effet les individus à risque constituent une minorité et pour les mettre en évidence, on utilise une technique de ré-échantillonage qui devra améliorer les résultats des algorithmes de prédiction. L'échantillonnage est basé sur une sélection aléatoire des profils non risqués, qu'on additionne aux profils risqués en construisant un nouvel ensemble équilibré de données. Les *F-scores* de validation pour les différents algorithmes sont respectivement de 0.2 pour la *régression logistique*, 0.3 pour le *SVM*, 0.28 pour le *NCC* et 0.60 pour les *forêts aléatoires*. A noter que le *SVM* est le seul à subir du sur-apprentissage. Il est vrai que cette méthode a permis de retourner des résultats performants , mais cette dernière reste toutefois limitée, car elle n'utilise pas l'ensemble des données disponibles. On l'a appliquée seulement pour l'englober dans le co-apprentissage.

Ici, il est clair que l'algorithme des *forets aléatoires* surclasse tous les autres algorithmes. On va donc le choisir comme classifieur 1.

Classifieur 2

D'après les résultats obtenus avant le re-échantillonage, on a vu que le *NCC* est l'algorithme le plus stable, donc on va le choisir pour le classifieur 2.

3.4.2 Expérimentations

Maintenant qu'on a fixé les algorithmes des classifieurs 1 et 2, on va faire des expériences pour choisir les algorithmes utiliser dans les initialisations. En plus dans la troisième expérimentation on va proposer une hypothese en plus qui q pour but de diminuer le bruit et augmenter la performance.

Première expérimentation : Concernant le deuxième échantillon, on commencera par prendre en considération des individus non risqués, car ils sont beaucoup plus nombreux, et donc on utilise un one class classifier pour l'initialisation du classifieur 2. On choisit le one class *SVM* comme modèle d'initialisation du classifieur 2, et un NCC pour le classifieur 2. Une fois testés sur les données simulées, on obtient une meilleure performance qui s'est amélioré de 5%. D'autre part, le modèle ne retourne pas de bonnes performances pour les données réelles avec un *F-score* de 0.08.

Deuxième expérimentation : Cette fois-ci, on prendra le *NCC* aussi comme modèle d'initialisation du deuxième classifieur. Après cette modification, on obtient de meilleures performances. En moyenne et après 100 itérations, le *F-score* obtenu est de 0.49 pour l'ensemble de validation et de 0.71 pour l'ensemble d'apprentissage.

A noter que le co-apprentissage a nécessité de paramétrer le nombre d'itération à 3, car au delà de ce nombre, la performance commence à se dégrader, ici il s'agit d'un problème de filtre. En effet l'augmentation du nombre d'itération engendre une augmentation des erreurs cumulées.

A titre de rappel, la détermination d'un pre-score est une approche innovante pour la banque. Dans le but de le calculer, on a utilisé des données qui sont principalement issues du web et ne figurent pas dans les informations bancaires. De ce fait, on n'a pas vraiment un pré-score référant pour réaliser une comparaison exacte. On va donc comparer le résultat du pré-score avec le score bancaire dont les données utilisées sont principalement renseignés par des clients et sont directement liées au risque. Pour cela, on introduit une nouvelle métrique, il s'agit de l'indicateur de *Gini* qui est utilisé traditionnellement pour la validation des scores dans la banque. La valeur de *Gini* référent selon la détermination des scores avec les méthodes bancaires est de 0.4.

La valeur de l'indice de *Gini* obtenue à l'issu des nouvelles données est de 0.42 ce qui est égale au *Gini* référent.

Pour vérifier que la capacité du modèle à détecter les profils non risqués est encore

	Régression Logistique	SVM	Forêts aléatoires	NCC
F1 Validation	0.41	0.37	0.49	0.44
F1 Apprentissage	0.43	0.65	0.72	0.43
Gini Validation	0.3	0.26	0.42	0.32
Gini Apprentissage	0.32	0.51	0.76	0.32

TABLE 3.3 – F-score et Gini après le co-apprentissage

élevée, on s'intéresse à la performance générale des *forêts aléatoires* qui est égale à 86%. On peut donc valider le modèle.

Troisième expérimentation : Cette expérimentation est basée sur la deuxième, mais cette fois ci, on va ajouter l'hypothèse que le classifieur d'initialisation du classifieur 1 est un classifieur robuste.

En pratique le classifieur d'initialisation ne va pas changer, mais la différenciation survient au niveau des données injectées dans ce classifieur. Cette hypothèse garantit que les données sont bien discriminantes et que le classfieur sera capable de mieux les identifier.

Cette modification a permis d'obtenir un *F-score* de 0.58 pour les *forêts aléatoires* sur l'ensemble de validation et une valeur de *Gini* de 0.45 soit 112% du *Gini* référent.

	Régression Logistique	SVM	Forêts aléatoires	NCC
F1 Validation	0.39	0.41	0.52	0.43
F1 Apprentissage	0.43	0.66	0.72	0.43
Gini Validation	0.27	0.31	0.45	0.31
Gini Apprentissage	0.31	0.52	0.8	0.32

TABLE 3.4 – F-score et Gini après le co-apprentissage avec hypothèse

Suite à l'hypothèse, la performance générale des *forêts aléatoires* devient aux alentours de 88%.

Conclusion

On peut confirmer que l'enrichissement des données était bénéfique pour la modélisation du risque. De plus, l'apprentissage semi-supervisé semble être une piste intéressante dans le cas de données de la DMP. Les diverses méthodes avancées, présentées durant ce volet et appliquées sur des données provenant du web ont permis d'obtenir un pré-score plus performant que le score calculé à partir des données renseignées par des clients.

Conclusion et Perspectives

Dans le cadre du stage de fin d'études, on a eu l'opportunité d'appliquer l'apprentissage automatique sur des données de log de navigation web dans le domaine bancaire. La mission principale a été de dresser le profil du risque des prospects provenant des bannières publicitaires en fonction des données issues de leur navigation web. Durant ce projet, plusieurs méthodes innovantes de prédiction et de prétraitement de données ont été mises en place, plus particulièrement la phase d'enrichissement des données qui a constitué une grande partie du projet. Pour arriver à nos fins, on a suivi une démarche bien spécifique : d'abord on a étudié et analysé les données existantes, ensuite on les a enrichi à l'aide de méthodes de web scrapping et de text mining. Enfin on a abordé l'étape de modélisation au cours de laquelle on a exploité les nouvelles données ainsi obtenues à l'aide de différentes technologies et techniques. Il y a eu aussi un profond travail de documentation facilitant la compréhension du code, qui permettra à toute personne prenant la suite des travaux effectués de continuer la tâche sans encombre. On a par ailleurs créé deux librairies Python qui serviront pour le web scrapping et pour le co-apprentissgae.

Ce travail a été très instructif pour plusieurs raisons. Techniquement, il nous a procuré l'opportunité à acquérir des connaissances dans le développement Spark et Python et de toucher à plusieurs aspects de la Data Science. De plus, il a été très enrichissant puisqu'il a donné l'occasion de développer des compétences sur des outils orientés « Big Data ». On a pu, ainsi, acquérir des connaissances dans des domaines transverses et liées au traitement des données volumineuses et les appliquer dans un cadre strict et réglementé qui est le milieu bancaire.

Dans ce stage d'après les différentes expérimentations on a pu prouver que les visites internet d'un prospect servent bien à prédire son profil risque. On a aussi prouvé que dans le cadre de la DMP, il est intéressant d'utiliser les prospects qui n'ont pas encore un profil risque pour améliorer la modélisation par le biais de l'apprentissage semi-supervisé.

Hormis le côté technique et fonctionnel, ce projet a été une occasion pour découvrir le travail dans une structure professionnelle au sein d'une grande société et les difficultés inhérentes comme la gestion du temps et la collaboration avec les services informatiques. En participant à la mise en place de la nouvelle stratégie de calcul de pré-score, on a pu découvrir la rigueur nécessaire à la réalisation d'un produit de qualité.

Le prototype du pré-score développé constitue une nouvelle référence pour l'estimation du risque des clients. A l'avenir, on envisage comme perspectives d'améliorer la modélisation du risque par le biais de plusieurs moyens. On envisage donc de réaliser un long web scrapping avec des pauses aléatoires, car si on extrait les données d'une façon très rapide, le scrappeur risque de tomber dans la catégorie des robots, souvent bannis. De plus, on propose dans le futur d'utiliser un dictionnaire spécifique à l'analyse de sentiments pour mieux cerner le sens du texte et proposer des indicateurs reliés au risque de crédit. S'ajoute à cela le choix de passer à l'étape d'enrichissement avant d'appliquer les analyses des sentiments et la classification des mots clés. On n'a pas pu faire cela à cause de limitations techniques dues à un manque de mémoire vive. En outre, dans la partie apprentissage, il est conseillé d'ajouter une phase de filtrage pour réduire le nombre de données qu'un classifeur injecte dans un deuxième. Cela pourra être réalisé en rajoutant à la sortie de chacun des classifieurs une condition d'appartenance à un domaine précis. De plus, les deux classifieurs pourront être utilisés aussi dans la phase de prédiction au lieu de les utiliser seulement pour labelliser les données à disposition.

Bibliographie

[Biernat 2014] Eric Biernat et Michel Lutz. Data science fondamentaux et études. 2014.

[Bird 2009] Steven Bird, Edward Loper et Ewan Klein. *Natural Language Processing with Python*. O'Reilly, Media Inc, pages 27–51, 2009.

[Dongre 2015] Jagdish Raikwale Dongre Vedpriya. *An Improved User Browsing Behavior Prediction Using Web Log Analysis*. 2015.

[Forman 2007] George Forman et Ira Cohen. *Learning from Little : Comparison of Classifiers Given Little Training*. 2007.

[Karau 2013] Holden Karau. *Fast Data Processing with Spark*. 2013.

[Li 2006] Trevor J. Hastie Li Ping et Kenneth W. Church. *Very Sparse Random Projections*. 2006.

[MapR 2017] Academy MapR. *DEV 360 - Introduction to Apache Spark and DEV 361 - Build and Monitor Apache Spark Applications*. 2017.

[Nasukawa 2003] Jeonghee Yi Nasukawa Tetsuya. *Sentiment Analysis : Capturing Favorability Using Natural Language Processing*. 2003.

[Parvin 2012] Moslem Mohamadi Sajad Parvin Zahra Rezaei and Behrouz Minaei Parvin Hamid. *Nearest Cluster Classifier*. 2012.

[Severance 2017] Charles Severance. *Using Python to Access Web Data*. 2017.

[Vapnik 1999] Vladimir Vapnik. The nature of statistical learning theory. 1999.

[Zhu 2008] Xiaojin Zhu. *Semi-Supervised Learning Literature Survey*. 2008.