RÉPUBLIQUE ALGERIENNE DÉMOCRATIQUE ET POPULAIRE

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

École Nationale Polytechnique

Département Génie Industriel

Entreprise KPMG

Mémoire de projet de fin d'études

Pour l'obtention du diplôme d'ingénieur d'état en Génie Industriel (Management Industriel et Management de l'Innovation)

Prédiction du Churn Rate Par le Machine Learning dans le secteur des M&A Application au sein de KPMG

Présenté par

Mohamed Aimed HAMOUR (Management Industriel)

Nazim Malik BENHAMDINE (Management de l'Innovation)

Sous la direction de Mme Nadjwa NOUAL MAA

Présenté et soutenu publiquement le 06/07/2020

Composition du Jury :

Président	M. Iskander ZOUAGHI	MCB	ENP
Promotrice	Mme Nadjwa NOUAL	MAA	ENP
Examinateur	M. Ali BOUKABOUS	MAA	ENP
Invité	Mme Amina SADAOUI	Consultante	KPMG

ENP 2020

RÉPUBLIQUE ALGERIENNE DÉMOCRATIQUE ET POPULAIRE

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

École Nationale Polytechnique

Département Génie Industriel

Entreprise KPMG

Mémoire de projet de fin d'études

Pour l'obtention du diplôme d'ingénieur d'état en Génie Industriel (Management Industriel et Management de l'Innovation)

Prédiction du Churn Rate Par le Machine Learning dans le secteur des M&A Application au sein de KPMG

Présenté par

Mohamed Aimed HAMOUR (Management Industriel)

Nazim Malik BENHAMDINE (Management de l'Innovation)

Sous la direction de Mme Nadjwa NOUAL MAA

Présenté et soutenu publiquement le 06/07/2020

Composition du Jury :

Président	M. Iskander ZOUAGHI	MCB	ENP
Promotrice	Mme Nadjwa NOUAL	MAA	ENP
Examinateur	M. Ali BOUKABOUS	MAA	ENP
Invité	Mme Amina SADAOUI	Consultante	KPMG

ENP 2020

Dédicaces

Je dédie ce travail :

À mes précieux parents qui se sont toujours sacrifiés pour ma réussite, m'ont soutenu, m'ont épaulé, m'ont fait part de leur entière confiance et ont toujours eu foi en ma personne, en plus de m'avoir doté d'une éducation digne, leur amour a fait de moi ce que je suis aujourd'hui.

À ma sœur, à qui je souhaite toute la réussite, bonheur et succès du monde, que dieu lui procure bonne santé et longue vie.

À toute ma famille, mes grands-parents, mes tantes, mes oncles, cousins, cousine.

À tous mes amis, Amar en ces temps difficiles, Anes grand entrepreneur, Yanis, Anis, Madjid et Rostane.

À mon binôme Aimed et tous les efforts qu'il a suscités durant ce projet.

À tous les gens qui m'aiment.

BENHAMDINE Nazim

Dédicaces

Je dédie ce travail :

A la mémoire de ma grand-mère Houria, qui me considérait comme son fils, qui a toujours cru en moi en me souhaitant toute la réussite du monde ;
A la mémoire de mon grand-père Mohamed pour tout ce qu'il m'a apporté au cours de ma jeunesse ;
A la mémoire de mon grand-père Rabah pour tous ses conseils et pour m'avoir toujours poussé à me dépasser pour réussir ;
A ma grand-mère Tasaadit pour m'avoir toujours accueilli en son domicile et apporté sa bienveillance ;
A mes chers parents qui m'ont éduqué, m'ont transmis des valeurs que je considère hautement, m'ont fait confiance et ont toujours cru en ma réussite, pour tous leurs sacrifices, leur appui inconditionnel, leur amour et leur soutien considérable tout au long de mes études, et sans qui je ne serais jamais arrivé là où je suis aujourd'hui.
A ma sœur, pour son appui et ses encouragements dans les moments difficiles, et à qui je souhaite une longue vie pleine de santé, bonheur et réussite.
A tous les autres membres de ma famille, mes tantes, mes oncles, mes cousins et cousines.
A tous mes amis proches, Amar, Fayçal, Kaoua et mes collègues du GI que sont Yanis, Anes et Anis.
A mon binôme et ami de longue de longue date Nazim pour tous les efforts qu'il a consentis durant ce projet, ainsi que son appui durant toutes ces années.
A ce cher CAP qui m'a tant donné et grâce auquel j'ai rencontré des gens formidables.
A tous les gens qui m'aiment et que j'aime en retour.
Je dédie ce modeste travail.

Aimed

Remerciements

Nous remercions tout d'abord le Dieu tout puissant de nous avoir guidés sur le chemin sur lequel nous nous trouvons aujourd'hui, et de nous avoir accordé le courage, la détermination et la volonté de mener à bien nos études, ainsi que ce projet.

Nous aimerions ensuite remercier vivement notre promotrice, Madame Nadjwa NOUAL pour son encadrement de grande qualité lors de cette dernière étape de notre scolarité, ainsi que pour son dévouement sans faille et son aide précieuse à l'accomplissement de cette mission. Son mentoring et ses conseils bienveillants au cours de notre parcours en spécialité nous aurons permis d'apprendre et de faire évoluer nos compétences. Nous lui témoignons, de ce fait, de notre reconnaissance la plus sincère.

Nous tenons également à remercier très chaleureusement toute l'équipe Deal Advisory de KPMG Algérie et plus particulièrement l'équipe « Deal Analytics » et Amina DAOUD qui nous a donné l'opportunité de réaliser ce projet. Notre gratitude va également à Mohamed BOUREHIL, Hadia MOKRANE, Fatima Zohra AOUALI et Abdelkrim BAHMED qui ont beaucoup contribué à notre épanouissement au sein du département, tout en nous aidant à développer nos compétences en Deal Advisory ; nos remerciements vont également aux autres stagiaires du département que nous avons eu la chance de côtoyer et avec qui nous avons tissé des liens très forts.

Nous remercions également notre encadrante au département et notre ainée indus, la consultante Amina SADAOUI, qui a su nous guider tout au long de notre aventure, en nous prodiguant des conseils très instructifs, en contribuant à nous former aux rouages du métier de consultant et en suivant de manière régulière et attentionnée le travail que nous avons effectué, en étant constamment disponible pour nous aider.

Nous remercions également Mr Mathieu BEAUCOURT (PDG), Mehdi BETTAHAR (Manager du département) et la responsable RH, Rym SEMGHOUNI qui ont veillé au bon déroulement de notre stage.

De plus, nous aimerions remercier l'ensemble des enseignants du département du Génie Industriel qui nous ont formés au cours de nos années au sein de la spécialité, ainsi qu'aux membres du jury pour l'évaluation de notre travail.

Enfin, un grand merci à tous ceux qui ont contribué de près ou de loin à l'accomplissement de ce travail.

الملخص :

KPMGهي شركة متعددة الجنسيات متخصصة في الاستشارات والمحاسبة والخبرة المالي والمشاركة، إلى جانب أنشطتها في دعم الشركات في عملية الاندماج أو الاستحواذ، وهي مسؤولية قسم" استشارات الصفقات" نهج يهدف إلى تحسين خدماتها المقدمة فيما يتعلق بالعناية الواجبة للسمعة. يتم تنفيذ هذه العملية من قبل فريق "تحليلات الصفقات" الذي يهدف إلى استخدام أدوات تكنولوجيا المعلومات لتعزيز التحول في نموذج العناية الواجبة للسمعة لذلك قمنا بتنفيذ مشروع تعلُّم الآلي يهدف إلى تصميم نموذج تنبؤي لمؤشر "معدل زبونة العملاء" في شكل برنامج كمبيوتر باستخدام خوارزميات التعلم تحت الإشراف، مما يجعل من الممكن تبسيط عملية إتخاذ القرار لأبعاد زمنية مختلفة. تم تفصيل الخطوات المؤدية إلى تصميم الحل الخاص بنا، مع النتائج التي تم الحصول عليها واقتراح للتنفيذ في عمليات الإدارة.

الكلمات الدالة: عمليات الدمج والاستحواذ, العناية الواجبة للسمعة, تعلُّم الآلي تصميم, إلغاء الاشتراك زبناء.

Abstract :

KPMG is a multinational professional services network with three lines of services : financial audit, tax and advisory and is continuously taking on the challenge of improving its quality of service in business support and due diligence precisely, which is the responsibility of the « Deal Advisory » department.
This procedure has been initiated by the « Deal Analytics » team that uses analytical and computing tools in order to support the due diligence activities and help introduce a new paradigm of the way processes work.
In this context, we were tasked to carry out a Machine Learning project with an objective of conceiving a predictive model for the « Client Churn Rate » indicator, using supervised learning algorithms, capable of being used to facilitate decision making at different temporary horizons.
The steps leading to assemble our solution are specified, with the results presented that we used to help suggest a way of implementing our models in the business processes of the department.
Keywords: Mergers & acquisitions, Due Diligence, Machine Learning, Modelling, Client Churn.

Résumé :

KPMG est une multinationale spécialisée dans le conseil, l'expertise comptable et financière et l'audit et engage en marge de ses activités dans l'accompagnement d'entreprises en cours de fusion ou d'acquisition, une responsabilité du département « Deal Advisory », une démarche visant à l'amélioration de ses services proposés concernant la Due Diligence. Cette démarche est entreprise par l'équipe « Deal Analytics » qui vise à utiliser les outils informatiques afin de véhiculer un changement de paradigme dans les rapports de Due Diligence.
Nous avons de ce fait mené un projet Machine Learning visant à concevoir un modèle prédictif de l'indicateur « taux de désabonnement clients » sous forme de programme informatique à partir des algorithmes de l'apprentissage supervisé, permettant de fluidifier la prise de décision avec une dimension temporelle modulable.
Les étapes menant à la conception de notre solution sont détaillées, avec les résultats obtenus et une proposition d'implémentation dans les processus métier du département.
Mots clés : Fusions & Acquisitions, Due Diligence, Machine Learning, Modélisation, Désabonnement client.

Tables des matières

Liste des Figures

Liste des Tableaux

Liste des Annexes

Liste des abréviations

AAS: Accounting Advisory Services.
ACV: Annual Contract Value.
ACV: Annual Contract Value.
ANN: Artifical Neural Networks.
ARPA: Average Revenue Per Account.
ARPU: Average Revenue Per User.
ARR: Annual Recurring Revenue.
AUC: Area Under Curve.
B2B: Business to Business.
B2C: Business to Consumers.
BDD: Buyer Due Diligence.
BI: Business Intelligence.
BPMN: Business Process Model and Notation.
CA: Chiffre d'Affaires.
CAC: Customer Acquisition Cost.
CART: Classification And Regression Tree.
CF: Cash Flows.
CLTV: Customer Lifetime Value.
CRISP-DM: Cross Industry Standard Process for Data Mining.
CSAT : Customer Satisfaction Score.
CTS: Cost To Serve.
CV: Cross Validation.
D&A: Deal Analytics.
DBN: Deep Belief Networks.
DBSCAN: Density-Based Spatial Clustering of Applications with Noise.
DD: Due Diligence.
DM: Data Mining.
DNR: Dollar Net Rate.
DRR: Daily Recurring Revenue.
EBITDA: Earnings Before Interest, Taxes, Depreciation, and Amortization.
ETI: Entreprise A Taille Intermédiaire.
EY: Ernst & Young.
FAQ: Foire Aux Questions.
FDD: Financial Due Diligence.
FN: False Negatives.
FP: False Positives.
FPR: False Positive Rate.
FTE: Full-Time Equivalent.
FY: Fiscal Year.
GAAP: Generally Accepted Accounting Principles.
GE: Grande Entreprise.
HCA: Hierarchical Cluster Analysis.
IA: Intelligence Artificielle.
ID3 : Iterative Dichotomiser 3.
IPO: Initial Public Offering.
KNN: K-Nearest Neighbours.

KPI: Key Performance Indicator.
KPMG: Klynveld Peat Marwick Goerdeler.
LBFGS: Limited-memory Broyden–Fletcher–Goldfarb–Shanno.
LIME: Local Interpretable Model-Agnostic.
LVR: Lead Velocity Rate.
M&A: Mergers and Acquisitions.
MCO: Moindre Carré Ordinaire.
ML: Machine Learning.
MLP: Multi-Layer Perceptron.
MRR: Monthly Recurring Revenue.
MSE: Mean Squared Error.
NPS: Net Promoter Score.
ODD: Operational Due Diligence.
PCC: Pearson Correlation Coefficient.
PIB: Produit Intérieur Brut.
PME: Petite Ou Moyenne Entreprise.
PPMCC: Pearson Product-Moment Correlation Coefficient.
PwC: PricewaterhouseCoopers.
QRR: Quarterly Recurring Revenue.
RBF: Radial Basis Function.
RBMs: Restricted Boltzmann Machines.
RF: Random Forests.
RH: Ressources Humaines.
ROC: Receiver Operating Characteristics.
RR: Recurring Revenue.
SaaS: Software as a Service.
SAV: Service Après Vente.
SGD: Stochastic Gradient Descent.
SMOTE: Synthetic Minority Over-Sampling Technique.
SPA: Sales and Purchase Agreement.
SPA: Société Par Actions.
SPI: Strategic Profitability Insights.
SVM: Support Vector Machine.
TCV: Total Contract Value.
TLU: Threshold Logic Unit.
TN: True Negatives.
TP: True Positives.
TPE: Très Petite Entreprise.
TPR: True Positive Rate.
TS: Transaction Services.
TVP: Target Value Platform
UK : United Kingdom.
VDD: Vendor Due Diligence.
WC: Working Capital.
WCR: Working Capital Requirement.

Introduction Générale

Introduction Générale

Puissant vecteur de croissance et de transformation, l'activité de fusions-acquisitions où « fusac », basée sur la création d'une alliance entre deux firmes ou plus, a considérablement évolué ces dernières années. Les raisons invoquées pour enclencher un processus de fusion ou acquisition sont nombreuses. En effet, il arrive parfois qu'une entreprise soit forcée à mettre la clé sous le paillasson, après des déboires financiers ou juridiques, entre autre. Les avantages découlant d'un rapprochement entre entreprises sont nombreux, puisqu'un tel processus permet pour les entreprises concernées de créer plus de valeur, de diversifier ou de recentrer leurs activités, de créer des synergies réductrices de coûts, ainsi que d'accélérer leur croissance.

Le processus menant au bon déroulement de la fusion ou de l'acquisition est de par nature complexe, et intègre plusieurs parties prenantes, en plus de l'entreprise vendeuse et l'entreprise acheteuse, comme les banques d'investissement, les cabinets d'avocats ainsi que les cabinets d'audit et de conseil qui sont impliqués. C'est dans ce dernier volet que s'inscrit KPMG, qui compte parmi les leaders historiques du conseil financier et dont les équipes du département « Deal Advisory » qui sont expertes dans les transactions de types fusions et acquisitions.

Le principal service proposé prend la forme de la « due diligence », coeur de métier d'une « fusac », elle permet pour une entreprise vendeuse d'accélérer le processus de cession en communiquant une information financière, juridique ou opérationnelle homogène et optimiser ainsi la négociation avec les acheteurs potentiels. Elle comporte différents aspects à étudier, qui varient selon l'entreprise et le secteur dans lequel elle mène ses activités.

Parmi les secteurs en vogue dans les « fusac » au cours des dernières années, celui du SaaS (Software as a service), pour « logiciel comme un service » se démarque au coeur d'une ère où la dématérialisation règne sur les entreprises, qui se dotent de plus en plus de solutions informatiques déployées de manière numérique.

Ayant oeuvré sur un nombre croissant de missions pour le compte d'entreprises éditrices de ces types de logiciels au cours des dernières années, KPMG, en recherche constante d'amélioration de son portfolio de services proposés, a tenu à concevoir une solution permettant d'améliorer son accompagnement taillé à ce genre de clients. Mais sous quelles formes pourrait-on concrétiser cet objectif ? Et comment ce dernier pourrait être implémenté au sein des processus métier du département « Deal Advisory » de KPMG ?

De nos jours, plusieurs entreprises ont compris l'importance de garder une traçabilité de leurs activités sous forme de données organisées et appréhendent également que ces ensembles de données représentent des leviers puissants de création de valeur, en marge de l'amélioration continue de leurs processus métiers. Cela est rendu possible grâce à l'expansion de l'utilisation des techniques du Machine Learning par des programmes informatiques pour extraire de l'information concrète à partir de données brutes.

Ce travail consistera donc à tirer profit des données accumulées de par les précédentes missions effectuées par KPMG auprès de clients oeuvrant dans le secteur du « SaaS » puis de dérouler une démarche visant à construire un modèle prédictif basé sur les méthodes issues du Machine Learning visant à être utilisé pour prédire un indicateur de performance incontournable dans le secteur du « SaaS » nommé « Churn rate » ou « Taux de désabonnement des clients ». Cette démarche s'inscrit dans un processus d'amélioration continue du processus de Due Diligence du cabinet KPMG Algérie, dont la première avancée fut la création d'une équipe appelée « Deal Analytics » chargée des analyses de données des transactions réalisées en appui au département « Deal Advisory », et dont nous ferons partie intégrante lors de l'accomplissement de notre mission.

Compte tenu de la pauvreté des références bibliographiques traitant de la construction de modèles prédictifs visant à prédire cet indicateur en Business to Business (B2B) dans le secteur du SaaS, nous baserons notre travail sur une recherche extensive visant à nous diriger vers une connaissance étendue du secteur. Cela aura pour but de construire des modèles innovants et dotés d'un fort pouvoir prédictif, répondant aux attentes du cabinet, souhaitant prendre le pas sur une concurrence encore en retrait dans le domaine de l'utilisation de modèles prédictifs en marge de ses processus métiers.

Pour but de présenter le travail réalisé par nos soins, décrire les étapes par lesquelles nous sommes passés et présenter les résultats obtenus, nous avons structuré ce document en trois parties organisées ainsi :

La première partie consiste en une présentation des différents concepts théoriques auxquels nous avons eu recours afin de concrétiser nos objectifs, et présentées sous deux chapitres :

- Le premier chapitre vient apporter des éclaircissements théoriques en ce qui concerne le marché fusions et acquisitions.
- Le second chapitre tient à vulgariser les différents concepts liés aux algorithmes de Machine Learning et leur évaluation.

La seconde partie a pour objectif de définir le cadre environnemental lié à la réalisation de notre projet à travers un diagnostic de l'existant, toujours sous deux chapitres :

- Le premier chapitre comprend une présentation de l'organisme d'accueil et de ses activités principales.
- Le second chapitre permet de cerner le déroulement actuel des processus de due diligence dans le département en vue d'en extraire une problématique.

La troisième et dernière partie comporte, pour sa part, le déroulement des étapes de la démarche visant à la conception de la solution proposée sous forme modèle prédictif, de par l'utilisation des techniques du Machine Learning et de l'intelligence artificielle, puis de son implémentation dans le cadre des activités du cabinet.

Partie 1 : État de l’art

Partie 1 : État de l'art

Cette partie fera fi de base théorique sur laquelle repose notre travail. Elle permet d'introduire les différents concepts et enjeux liés au marché des fusions et acquisitions (M&A) avec les différentes définitions liées notamment à la due diligence. Nous expliquerons par la suite, à travers la description du paradigme de financement des ventes présent dans le secteur du Software as a service qui est l'objet de notre travail, les notions relatives à la rétention et l'attrition des clients (Client Churn) et leur importance dans la construction d'un business model durable.

Nous nous attarderons par la suite sur la définition de l'apprentissage automatique (Machine Learning) ainsi que sur la théorie derrière les différents algorithmes d'apprentissage supervisé que nous aurons à utiliser en marge de notre projet.

Chapitre 1 : Contexte et enjeux du marché des M&A

L'environnement fortement concurrentiel entre les entreprises mène certaines à déployer des stratégies de croissances externes afin de pouvoir s'imposer sur un marché et prendre le pas sur leurs concurrents. Ces stratégies sont souvent concrétisées par une fusion avec une autre firme, visant à former une entité ayant plus de poids sur le marché, ou alors une acquisition totale ou partielle des actifs d'une entreprise ciblée pour renforcer sa position concurrentielle. Nous prendrons soin, dans ce chapitre, de définir différentes notions relatives aux activités de fusions et acquisitions.

1. Fusions et acquisitions (M&A)

Au cours des dernières décennies, plusieurs exemples de compagnies, comme Général Electric, Google ou Cisco, ont démontré la capacité de la croissance par fusion/acquisition à générer des revenus importants et gagner des parts de marché.

1.1. Définition des M&A

En effet les fusions et acquisitions sont des opérations de regroupement ou de prises de contrôle d'entreprises cibles, réalisées par l'intermédiaire d'un achat ou d'un échange d'actions. Elles s'inscrivent dans le cadre d'une stratégie de croissance externe que peut développer une entreprise pour but de d'accès à un nouveau segment client ou d'expansion géographique entre autres. Il existe une distinction entre les deux termes fusions et acquisition, en effet on peut les définir comme suit :

- **Fusion** : Une combinaison de deux ou plusieurs sociétés dans laquelle les actifs et les passifs de la (ou des) entreprise(s) vendeuse(s) sont absorbés par l'entreprises acheteuse. Bien que l'acheteuse puisse devenir une organisation considérablement différente après la fusion, elle conserve son identité d'origine.
- **Acquisition** : concerne l'achat d'un actif comme une usine, une division ou alors toute une entreprise.[1]

1.2. Avantages et inconvénients des M&A

Nous pouvons résumer les principaux avantages et inconvénients des fusions et acquisitions dans le tableau suivant :

[1] Meier et Schier, 2009, P.7-9

Tableau 1 : Avantages et inconvénients des M&A[2]

Avantages	Inconvénients
• Accès rapide à de nouveaux domaines d'activités • Contrôler des ressources supplémentaires • Exploitation de synergies de coûts ou de complémentarités • Augmentation du pouvoir de marché de l'entreprise • Économie d'intégration verticale • Économies d'échelles • Élimination des inefficiences • Intégration de nouveaux marchés en marge d'une stratégie d'expansion (notamment à l'international)	• Besoins en capitaux élevés • Problèmes de coordination et contrôle des activités regroupées • Coût de l'intégration physique des activités • Impact psychologique de l'opération sur le climat social • Intégration culturelle et managériale des entités délicate • Valorisation de la cible délicate à déterminer • Asymétrie de l'information entre les parties • Due diligence peu exhaustive ou incomplète

1.3. Processus de M&A

Le processus de fusion-acquisition peut être caractérisé par 3 étapes principales qui sont :

- **La phase pre-merger :** Cette phase consiste en la mise en place d'une stratégie qui nécessiterait le recours à une croissance par fusion-acquisition. Les objectifs à long terme sont débattus et un brainstorming peut être effectué afin de cerner des cibles potentielles et leur valeur ajoutée.
- **La phase de transaction :** Cette phase se caractérise par un ciblage affiné d'entreprises potentielles à acquérir. La firme acheteuse contacte ses cibles primaires puis entame ses recherches plus approfondies sur les activités de ces dernières. Des négociations peuvent ensuite avoir lieu afin d'entrevoir la possibilité d'une transaction entre les deux parties. Une fois qu'un candidat sort du lot, l'entreprise acheteuse peut entamer la phase de due diligence qui permet la pondération exhaustive de toutes les facettes opérationnelles et financières de l'activité d'une organisation.
- **La phase post-merger :** Une fois que la due diligence aura donné des résultats satisfaisants quant à la compatibilité d'un rapprochement entre deux parties, les négociations finales peuvent avoir lieu entre les entreprises concernées. Celles-ci portent notamment sur l'évaluation financière, la logistique à mettre en place et les ententes à mettre en place en ce qui concerne les aspects culturels et opérationnels. S'en suit l'intégration qui permet de conclure l'alliance entre les deux parties, œuvrant à maximiser leur productivité en diminuant les risques de répulsion.

Nous pouvons illustrer les principales activités se rattachant aux phases essentielles sur le schéma suivant :

[2] Meier et Schier, 2009.

Figure 1 : Principales activités du processus de M&A[3]

2. La Due Diligence

Afin de pouvoir juger de la faisabilité d'une opération de fusion-acquisition, il convient de passer par le processus de due diligence, un processus permettant de connaître sa cible de manière exhaustive, à travers plusieurs types d'études.

2.1. Définition de la due diligence

La due diligence (DD), appelée également « audit d'acquisition » est une procédure qui permet à l'acquéreur de vérifier un certain nombre d'informations en un minimum de temps en ce qui concerne sa cible pour but de statuer sur la faisabilité d'une transaction. Ce dernier met à disposition de son acheteur potentiel un grand nombre de documents dans une salle d'information (data room) pour accomplir cette procédure. La due diligence est une étape primordiale dans l'accomplissement d'une fusion-acquisition. Elle couvre plusieurs facettes de l'activité de la cible, de l'aspect financier au juridique en passant par les composantes opérationnelles.

2.2. Types de due diligence

Plusieurs audits de due diligence sont opérées auprès d'une cible, on retrouve notamment :

- **Due diligence comptable et financière** : Elle représente la partie la plus importante d'une due diligence, et une variable prépondérante dans la décision d'accomplir une transaction. « *La DD comptable et financière permet de prendre connaissance de la réalité des pratiques comptables de l'entreprise, afin de s'assurer que les comptes annuels sont réguliers, sincères et qu'ils donnent une image fidèle du patrimoine de l'entreprise, de sa situation financière et du résultat de l'entreprise.* » [4]

Un nombre important de paramètres financiers est soumis à une analyse détaillée, entre la solvabilité, la liquidité, l'analyse des coûts fixes et variables, la dette à court et long terme ainsi que les flux de trésorerie. Une image fidèle de la santé financière de l'entreprise vient en sortie de ce processus, comprenant l'identification des zones à risques.

[3] Corporate Finance Insititute, 2019, P.34-42
[4] MEIER, SCHIER. Fusions acquisitions, Stratégie, Finance, Management. P.191

- **Due diligence commerciale** : Elle met l'accent sur l'analyse du marché dans lequel œuvre une entreprise. Ce type de DD couvre les aspects spécifiques à une entreprise en ce qui concerne les achats, les ventes, les fournisseurs, les types de contrats négociés ainsi que l'efficacité de la chaîne logistique et d'approvisionnement. Il convient également de se pencher sur les domaines de recherche et développement et la propension qu'a la cible à accorder de l'importance à l'innovation dans son domaine d'activité. Une analyse des principaux concurrents est également incluse, au même titre que les produits et services qui contribuent à leur succès.

- **Due diligence opérationnelle** : l'ODD permet d'analyser les processus de travail de l'entreprise cible. Cette forme de due diligence est notamment privilégiée par les acheteurs d'entreprises industrielles. Des experts analysent la compatibilité du business plan avec les données opérationnelles afin d'évaluer les risques associés et afin de cibler des axes d'amélioration pour but de créer de la valeur à partir de l'optimisation de la chaine logistique ou de l'automatisation.

- **Due diligence environnementale** : La due diligence environnementale sert à vérifier la conformité d'une entreprise aux réglementations environnementales en vigueur. Elle connait une importance croissante en marge de la gestion de l'environnement, en particulier chez les entreprises manufacturières. Un certain nombre de paramètres rentre dans le cadre d'un audit, comme les risques liés à l'emplacement, le taux de contamination ou de pollution des sites de production ou alors l'impact sur l'environnement de l'utilisation industrielle.

- **Due diligence légale** : Concerne l'examen des conditions légales d'une société. Cette démarche s'intéresse aux contrats de travail, droits de brevets et de propriété intellectuelle ainsi qu'à la structure de propriété de la société concernée et de ses filiales, qui peuvent avoir différentes formes juridiques. Une attention particulière est également portée aux litiges actuels et potentiels de la firme en question, en marge d'un examen juridique dans lequel les experts analysent les contrats d'achats ou les baux possédés par une entité.

- **Due diligence réputationnelle** : La DD réputationnelle concerne l'évaluation minutieuse des risques de réputation attachés à un partenaire commercial ou à une entreprise cible, y compris les questions relatives à l'intégrité et à la crédibilité des personnes qui la gouvernent ainsi que la fiabilité et la prévisibilité de l'environnement politique. A cet effet, la due diligence réputationnelle s'effectue principalement par l'analyse de la réputation en ligne des entreprises, également connue sous le nom de « e-reputation » ou « online reputation ».

2.3. La due diligence dans le secteur du Software as a Service (SaaS)

Les solutions « Software as a Service » ou « SaaS » se sont implantés dans le secteur de l'informatique. On peut regrouper les éditeurs de SaaS en trois catégories :

- **Fournisseurs de SaaS purs :** Ces entreprises ont conçu leur offre primaire à partir d'une solution cloud. Ce sont, entre autres, les pionniers du SaaS comme Salesforce ou NetSuite.
- **Fournisseurs de logiciels sur site :** Suivant le courant des entreprises cloud, certaines entreprises informatiques telles que Adobe, Oracle ou Intuit ont effectué une transition vers un modèle de services hybrides où l'offre de cloud cohabite avec l'offre traditionnelle de logiciels téléchargeables.
- **Sociétés intégrées de technologie et de produit :** Les géants du marché de la technologie comme HP, IBM ou Cisco intègrent des offres de SaaS au sein de leur palette de services.

Cette nouvelle branche de la technologie nécessite néanmoins un nouveau mode de management peu traditionnel, qui se caractérise par plusieurs indicateurs et métriques à surveiller tout au long du cycle de vie d'une solution SaaS. Ces métriques doivent être mesurées et interprétées de manière précise afin de prendre les bonnes décisions sur chaque phase du cycle de vie d'une solution de SaaS.

Ce dernier se compose de 3 phases : lancement, évolution et stabilisation. Une étude de croissance, profitabilité et durabilité doit être effectuée sur chacune de ces étapes. Ces études peuvent être résumées dans ce qui suit :

- **Croissance :** Mesure de performance importante auprès des éditeurs de SaaS, particulièrement durant les premières années de services. Le taux de croissance est intimement lié à la pérennité financière et est utilisé pour comparer le positionnement de son entreprise sur le marché. Parmi les facettes de la croissance qui sont mises à l'étude en marge d'une entreprise de SaaS, on retrouve la croissance client et la croissance de revenu. Chacune d'entre elles couvre un certain nombre de métriques à optimiser, comme on peut le visualiser sur le schéma ci-dessous :

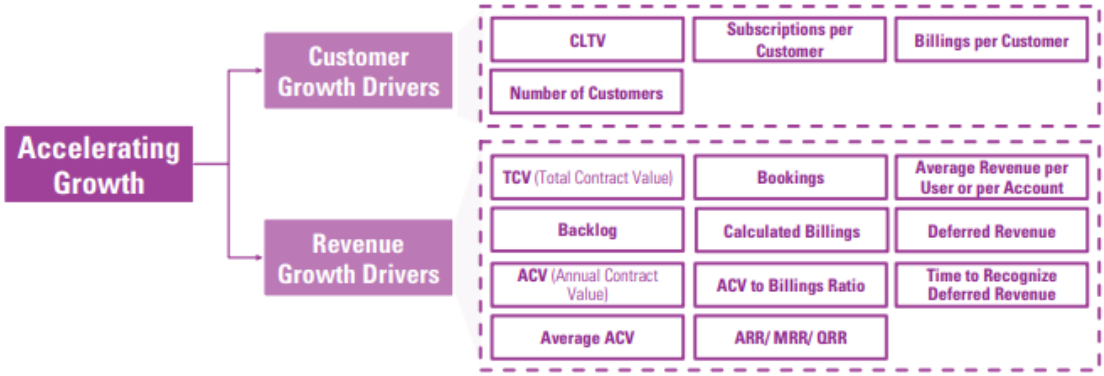


Figure 2 : Métriques permettant de mesurer la croissance [5]

- **Profitabilité :** Se mesure en s'intéressant aux dimensions coûts, marges et flux de trésorerie (Cash-Flow) en prenant en compte un certain nombre de métriques relatifs à chaque segment, comme illustré sur le schéma suivant :

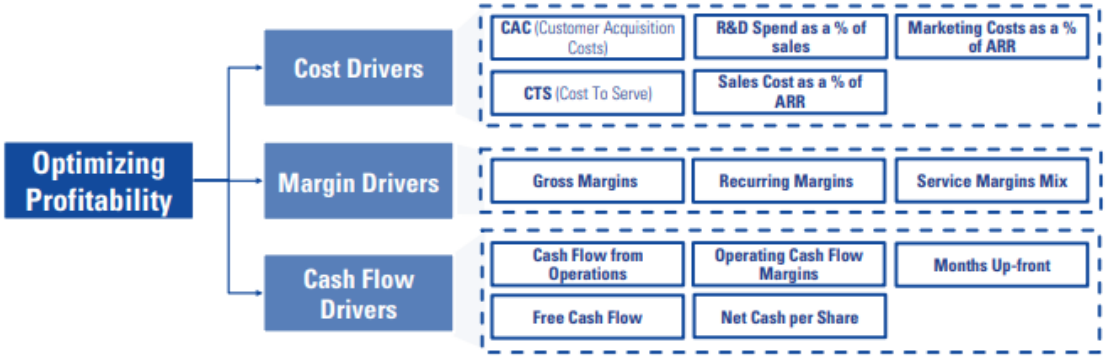


Figure 3 : Métriques permettant la mesure de la profitabilité [6]

[5] Document Interne KPMG
[6] Document Interne KPMG

- **Durabilité :** Surveiller la durabilité à long terme d'une entreprise de SaaS est un processus capital pour garantir son succès, en s'intéressant notamment à l'efficacité des ventes, la rétention client ainsi que l'expérience de l'utilisateur. Un nombre important de métriques est utilisé afin de mesurer la durabilité d'un éditeur de SaaS sur ces 3 dimensions, répartis suivant l'illustration ci-après :

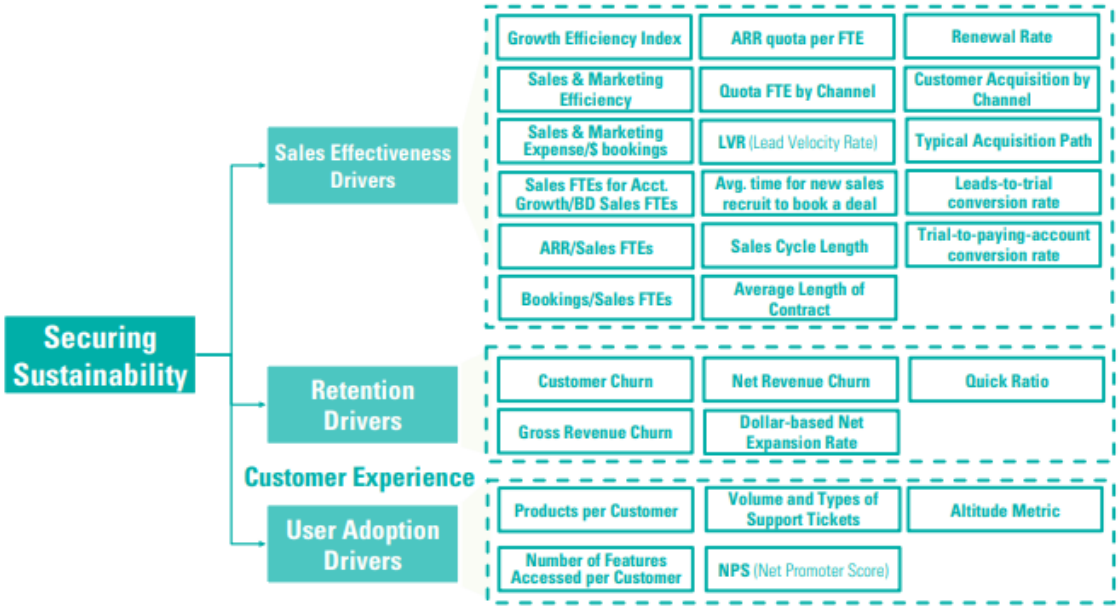


Figure 4 : Métriques de mesures de la durabilité [7]

Aujourd'hui, l'audit d'acquisition d'une entreprise du secteur SaaS comprend une étude basée sur ce précédent paradigme, assez représentatif des pratiques actuelles en termes de mesure de la performance économique et commerciale.

Le taux d'attrition ou Churn Rate est une métrique affiliée aux métriques de durabilité et est largement utilisée dans la mesure de la rétention client et constitue un levier important pour les entreprises vendeuses afin d'attirer les investisseurs.

3. Le Taux d'attrition (Churn Rate)

L'attrition des clients ou perte de clients ou de revenu durant une période de temps, peut impacter significativement la croissance et la durabilité d'une entreprise de SaaS. Opposée donc à la rétention de clients, l'attrition est notamment mesurée lors de la phase d'évolution du service lors du cycle de vie d'un éditeur de SaaS.

3.1. Définitions des métriques mesurant l'attrition des clients

Plusieurs métriques sont utilisées afin de traquer la rétention (et donc l'attrition) de sa clientèle, on les définit comme suit :

- **Monthly Reccuring Revenue (MRR)** : Le revenu mensuel récurrent est un revenu qu'une entreprise compte percevoir chaque mois, un revenu prévisible. Il est utilisé afin de suivre les revenus récurrents par client au fil du temps, par incréments mensuels. On peut aussi parler du « Annual Reccurring Revenue » ou ARR qui est la somme des MRR sur une année.

[7] Document Interne KPMG

- **Average Revenue per User (ARPU)** : Défini comme étant la moyenne des revenus mensuels gagnés par utilisateur. Son calcul s'effectue de la manière suivante :

$$ARPU = \frac{Total\ MRR}{Nombre\ de\ clients}$$

- **Customer LifeTime Value (CLTV)** : Cette métrique indique le revenu moyen qu'une entreprise peut attendre d'un seul compte client au cours de la durée de vie de ce dernier. Les entreprises utilisent cet indicateur pour identifier les segments de clientèle importants pour l'entreprise. On le calcule ainsi :

$$CLTV = \frac{Revenu\ total * Nombre\ de\ clients}{Nombre\ de\ commandes *\ Nombre\ de\ ventes * Customer\ Churn}$$

  On compare souvent cet indicateur au Customer Acquisition Cost.

- **Annual Contract Value (ACV)** : L'ACV est la valeur des revenus perçus de par les abonnements de chaque année pour les clients sous contrats, normalisés sur un an. Si par exemple un client signe un contrat de 5 ans avec une entreprise pour une somme de 50 000$, sur une seule année l'ACV sera de 10 000$. La formule de calcul de l'ACV est définie ainsi :

$$ACV\ = \frac{Valeur\ totale\ du\ contrat(sans\ les\ frais\ additionnels)}{Nombre\ total\ d'années\ du\ contrat}$$

- **Customer Acquisition Cost (CAC)** : Cet indicateur est défini comme étant la moyenne des coûts dépensés par nouveau client acquis. Ces coûts concernent notamment les coûts de marketing, de recueil des besoins ainsi que ceux de l'intégration, du déploiement de la solution et potentiellement de la formation nécessaire aux utilisateurs.

$$CAC = \frac{Somme\ des\ coûts\ dépensés\ pour\ acquérir\ de\ nouveaux\ clients}{Nombre\ de\ nouveaux\ clients\ acquis\ sur\ la\ période}$$

- **Customer Churn** : fait référence au taux de clients qui ont mis fin à leur souscription au cours d'une période de temps donnée. Il est mesuré comme étant le rapport du nombre de clients perdus sur le nombre de clients actifs au début d'une période donnée :

$$Customer\ Churn\ \% = \frac{Nombre\ de\ clients\ perdus}{Nombre\ de\ clients\ actifs\ au\ début\ d'une\ période} * 100$$

- **Gross MRR (Monthly Recurring Revenue) Churn** : cette métrique est utilisée afin de quantifier la perte de revenu encourue au cours d'une période due aux clients qui se sont désabonnés ou qui ont choisi un « downgrade » de leur offre, par le rapport suivant :

$$Gross\ MRR\ Churn\ \% = \frac{Somme\ des\ MRR\ perdus\ sur\ un\ mois}{Total\ du\ MRR\ au\ début\ du\ mois} * 10$$

- **Net MRR Churn** : cette métrique est incontournable afin de juger de la santé financière d'une entreprise SaaS, et représente un obstacle important à la croissance. En calculant le Net MRR Churn, une entreprise peut quantifier son taux de croissance par rapport à la proportion de sa clientèle qui a connu un « upsell » ou « Expansion MRR » et signifie littéralement « vendre plus » en perfectionnant l'expérience du client par rapport à son service et donc augmenter ses ventes. Le calcul du Net MRR Churn se fait par la formule suivante :

$$Net\ MRR\ Churn\ \% = \frac{Total\ MRR\ perdu - Expansion\ MRR(Upsells)}{Total\ MRR\ au\ début\ du\ mois} * 100$$

Les entreprises de SaaS visent généralement une valeur négative pour ce pourcentage, avec pour limite une valeur nulle.

- **Dollar Net Expansion Rate (DNR) :** Cette métrique permet de caractériser le revenu engrangé à partir de sa clientèle actuelle retenue en comparaison avec la période qui a précédé en tenant compte des upsells (gains) et des downsells et clients perdus (pertes). On peut utiliser cette métrique et l'appliquer sur différentes cohortes (segments de clients partageant plusieurs caractéristiques communes), afin d'avoir une idée sur la façon dont ces segments réagissent face aux stratégies de rétention. On peut visualiser la formule de calcul de cette métrique par la loi suivante :

$$Dollar\ Net\ Expansion\ Rate\ \% = \frac{Total\ MRR + upsells - downsells - churn}{Total\ MRR} * 100$$

La valeur de référence ciblée par une entreprise de SaaS, notamment lors des périodes plus avancées de son cycle de vie, est de 109% qui est la valeur médiane pour les entreprises qui ont connu une introduction en bourse (IPO). Un taux du DNR supérieur à 100% signifie qu'une entreprise acquiert plus d'argent en effectuant des upsells (et cross-sells qui représentent les clients qui changent d'offre) qu'elle n'en perd à cause de ses clients qui se désabonnent ou qui réduisent leur offre.

- **Quick Ratio** : Permet de cerner de manière rapide l'efficience de croissance d'un business de SaaS, en intégrant deux variables essentielles qui sont le churn et le revenu. Particulièrement utilisée lors de la phase de lancement du cycle de vie d'un éditeur de SaaS, sa formule de calcul est donnée comme suit :

$$Quick\ Ratio = \frac{MRR\ gagné + Expansion\ MRR}{MRR\ perdu + Downgrade\ MRR}$$

La référence retenue pour ce ratio est fixée à une valeur de 4. En effet pour chaque dollar $ perdu à cause d'un client, l'entreprise gagne 4$, faisant de ses revenus une valeur sure à terme.

Le graphe suivant illustre les phases du cycle de vie d'un éditeur de SaaS pendant lesquelles les métriques les plus importantes relativement à la rétention client sont mesurées :

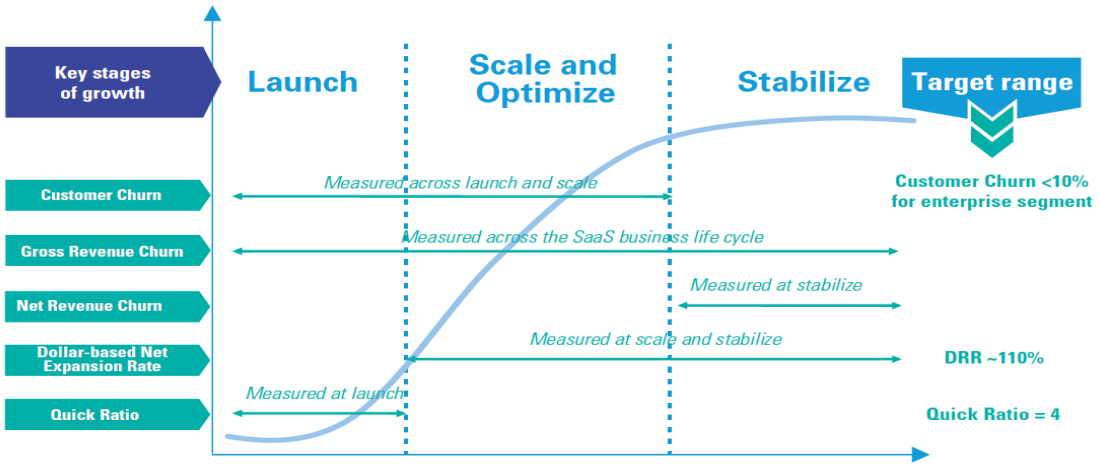


Figure 5 : Métriques de rétention durant la phase de vie d'un éditeur de SaaS [8]

3.2. Facteurs liés à l'attrition dans le secteur du SaaS

Afin d'analyser le churn, les entreprises ont besoin de comprendre les comportements de leurs clients les plus prompts à mettre fin à leur abonnement et donc de décrire ces comportements. Plusieurs facteurs peuvent impacter sur un client et dépendent de la nature du business considéré. Chaque firme a des facteurs spécifiques qui peuvent avoir une répercussion sur la variation du taux de Churn. Ahn et al. (2006) mettent en place un modèle conceptuel pour le Churn des clients, qui prend en compte 5 groupes de facteurs différents qui ont effet direct sur ce taux d'attrition : La satisfaction client, les variables propres au client, le statut du client, l'expérience utilisateur ainsi que les coûts de changement de fournisseur. On peut visualiser ces groupes de facteurs sur le schéma suivant [9] :

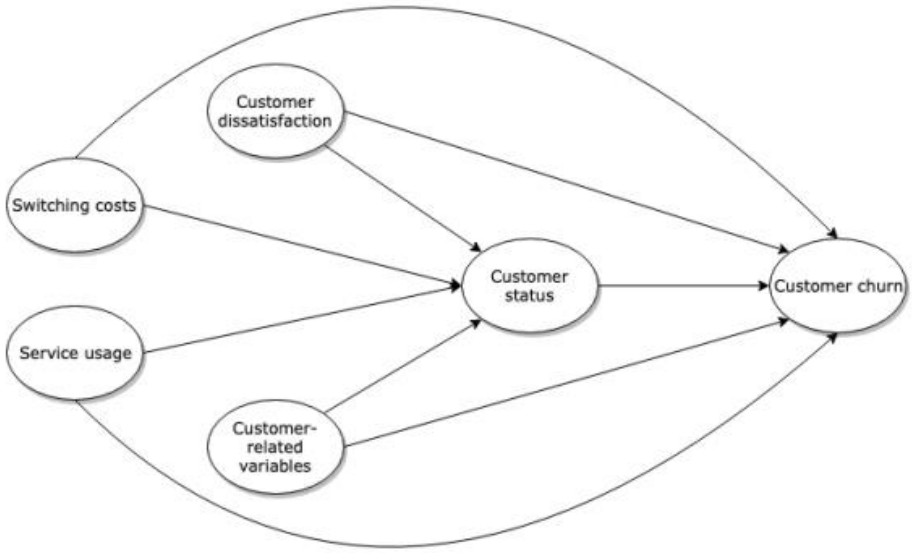


Figure 6 : Classes de facteurs menant à l'attrition client [10]

[8] Document Interne KPMG
[9] Anton Rautio, Churn Prediction in SaaS using Machine Learning, P.4-5
[10] Anton Rautio, Churn Prediction in SaaS using Machine Learning, P.05

Ces dimensions peuvent être déclinés en plusieurs facteurs, qui ont un impact direct et indirect sur la propension qu'a un client à se désabonner d'un service. Ces facteurs concernent aussi bien des clients d'une entreprise oeuvrant en B2C qu'une entreprise qui base ses activités sur un modèle de B2B. On retrouve notamment :

3.2.1. La satisfaction client

Représente une dimension essentielle dans une étude liée à la rétention client. En effet, un client non satisfait par un produit ou un service aura une probabilité plus importante de changer de service, en s'abonnant chez un concurrent. Un client satisfait, réagissant de manière positive à une stratégie de rétention, économisera des coûts important à une entreprise, en ce qui concerne l'acquisition notamment. Nous pouvons décliner cette dimension en un certain nombre de facteurs tels que :

- La confiance du client
- Le Customer Life Time Value (CLTV) défini précédemment
- La durée de vie moyenne d'un client
- Le Customer Satisfaction Score (CSAT) qui est un indicateur de satisfaction historique, le KPI le plus utilisé par les équipes marketing, construit à partir de la question « Êtes-vous satisfait de X ? ».

$$CSAT\% = \frac{Total\ de\ réponses\ positives}{Réponses\ totales} * 100$$

- Le Net Promoter Score (NPS) qui est un indicateur permettant de mesurer la propension des clients à recommander un service d'une marque. Il est calculé à partir d'un questionnaire où il est demandé d'évaluer sur une échelle généralement 0 à 10, quelle est la probabilité qu'un client recommande une entreprise à son entourage.

On obtient par la suite l'indicateur suivant :

$$NPS = \%promoteurs - \%détracteurs$$

Ces facteurs permettent donc de cerner plusieurs caractéristiques propres au niveau de satisfaction client.

3.2.2. Les coûts de changement de fournisseur

Concernent l'ensemble des coûts que supporte un client afin de pouvoir changer de fournisseur de service[11]. Ces coûts incluent par exemple :

- Coûts de désabonnement du service actuel
- Coûts de prospection d'un nouveau fournisseur
- Coûts de transaction
- Tarification du nouveau fournisseur pour le nouveau service choisi

L'ensemble de ces coûts peuvent influencer de manière assez considérable la décision d'un client à se désabonner d'un service donné.

3.2.3. Les variables propres au client

On retrouve dans cette catégorie différentes variables relatives au client affectant son comportement au sein du segment B2B. On retrouve comme facteurs :

[11] Heide, 1995

- Le Chiffre d'Affaire (CA) du client
- La tarification et la perception du client à son égard
- Les stratégies d'attraction engagées par les concurrents
- Les problèmes d'éthique auxquels a fait face un client
- Les problèmes liés à l'utilisation du service
- Les occurrences de querelles entre le client et un représentant du fournisseur de service
- Propension à adopter une nouvelle technologie

Ces variables, propres au client, vont faire pencher la balance dans un sens comme dans l'autre en ce qui concerne le désabonnement ou pas. Lorsque les problèmes d'un client lorsqu'il utilise un service sont récurrents, il aura tendance à se désabonner, par exemple.

3.2.4. Expérience utilisateur du service

Il est question dans ce cluster de classifier un nombre de facteurs relatifs à l'utilisation du service ainsi qu'au fournisseur de services, on peut citer les suivants :

- Fréquence et durée d'utilisation du service
- Nombre d'actions effectuées
- Image de marque du fournisseur de SaaS
- Taux d'utilisation des services offerts par l'éditeur de SaaS
- Monthly Reccurring Revenue (MRR)
- Nombres d'Upsells et de Downsells
- Présence d'un Service après vente (SAV)
- Qualité de l'intégration de la solution SaaS au sein des activités du client

Lorsque le fournisseur de SaaS propose un service de qualité, il aura tendance à garder ses clients.

3.2.5. Statut du client

Cette catégorie ne contient qu'une variable qui peut prendre plusieurs valeurs. Cette variable fait référence au statut d'utilisation actuel du service de la part d'un client. Ce dernier peut être un utilisateur actif, inactif ou alors suspendu.

Conclusion :

Dans le cadre de ce travail, il fut nécessaire de décrire le processus de fusion-acquisition, puis de caractériser les différentes analyses effectuées en marge de la due diligence. Il a ensuite été question de souligner le changement de paradigme qui a eu lieu dans la due diligence relative au secteur du Software as a service (SaaS) et de mettre en lumière les indicateurs de mesure de la performance d'une entreprise qui œuvre dans ce secteur. Dans ce contexte-là, il fut primordial de faire ressortir l'importance accordée par les investisseurs à la durabilité d'une entreprise de SaaS, concrétisée notamment par sa capacité à retenir ses clients, un phénomène expliqué par différents facteurs.

Chapitre 2 : Prédire à l'aide du machine learning

Un churn imprévu peut avoir des conséquences sur la croissance et les marges acquises par un éditeur de SaaS, il est donc primordial de cerner des comportements menant à l'attrition, afin de pouvoir concevoir un modèle permettant la prévision de l'attrition. Le Machine Learning permet d'effectuer des prévisions de manière optimale à l'aide des algorithmes d'apprentissage automatique supervisé et non supervisé que nous allons développer dans ce chapitre, puis nous allons décrire les méthodes utilisées afin d'évaluer leur performance.

1. L'intelligence artificielle

L'intelligence artificielle appelée IA, est un type d'intelligence propre aux machines, contrairement à l'intelligence naturelle dont font preuve les humains et les animaux. La discipline scientifique de l'IA, fondée en 1955, concerne l'ensemble des théories et des techniques mises en œuvre en vue de concevoir des machines capables de simuler l'intelligence. Cette imitation peut se faire dans le raisonnement comme dans les jeux, dans la pratique des mathématiques, dans la compréhension des langues naturelles, la perception visuelle ou auditive ou alors dans la commande d'un robot dans un milieu inconnu ou hostile. A partir des années 1980, et suite au développement des technologies informatiques et de la puissance de calcul des ordinateurs, la discipline de l'apprentissage automatique, plus connue sous le nom de Machine Learning (ML), voit le jour. Cette dernière se développe grâce aux ordinateurs qui commencent à déduire des règles à suivre en analysant un jeu de données. Des algorithmes « apprenants » sont créées et apparait donc l'apprentissage supervisé, non supervisé puis l'apprentissage par renforcement. Aujourd'hui, le Machine Learning est le meilleur moyen utilisé afin de simuler une intelligence artificielle.

2. Le Machine Learning

L'apprentissage automatique ou apprentissage machine dit Machine Learning est un champ d'étude de l'intelligence artificielle. Il se fonde sur les approches mathématiques et statistiques pour donner à des ordinateurs la capacité d'apprendre à partir de données. Ils font par la suite en sorte d'améliorer leurs performances à résoudre des tâches sans être explicitement programmés pour chacune. Plus largement, il concerne la conception, l'analyse, l'optimisation, le développement et l'implémentation de méthodes avec pour l'apprentissage machine comme résultat attendu.

2.1. Définition du Machine Learning

Le père fondateur du ML est Arthur Samuel, un mathématicien américain qui a développé un programme pouvant apprendre tout seul comment jouer aux Dames en 1959. Il définit le Machine Learning ainsi :

« Le Machine Learning consiste à laisser l'ordinateur apprendre quel calcul effectuer, plutôt que de lui donner ce calcul en le programmant de façon explicite ».

Une autre définition, plus utilisée dans les sciences de l'ingénieur est apportée par Tom Mitchell en 1997 :

« Un programme informatique apprend à partir d'une expérience E par rapport à une tâche T donnée et une mesure de la performance P, si cette dernière performance P sur la tâche T s'améliore avec l'expérience E ».

Par exemple, un filtre de spam utilisé par les boites de messagerie électronique, est un programme de Machine Learning qui apprend à signaler les spams en ayant comme source d'apprentissage plusieurs exemples de mails dits spams. Les exemples utilisés par le système sont appelés ou ensemble d'entrainement ou « training set ». Chaque exemple d'entrainement est appelé instance d'entrainement ou « training instance ». Dans ce cas, la tâche T est de signaler les spams pour les nouveaux emails, l'expérience E est l'entrainement à partir d'exemple ou « training data » et la performance P est à définir. On peut utiliser par exemple le ratio d'emails classifiés correctement. Cette performance est appelée précision ou « accuracy » et est souvent utilisée dans les tâches de classification[12].

2.2. Apports du Machine Learning

L'avènement du ML a enclenché un changement de paradigme dans la discipline de la science des données (data science). En effet, traiter des données avec les méthodes traditionnelles requiert une quantité importante d'effort afin de résoudre un problème, nécessitant à chaque itération l'intervention humaine. Cette approche traditionnelle peut être illustrée par le schéma suivant :

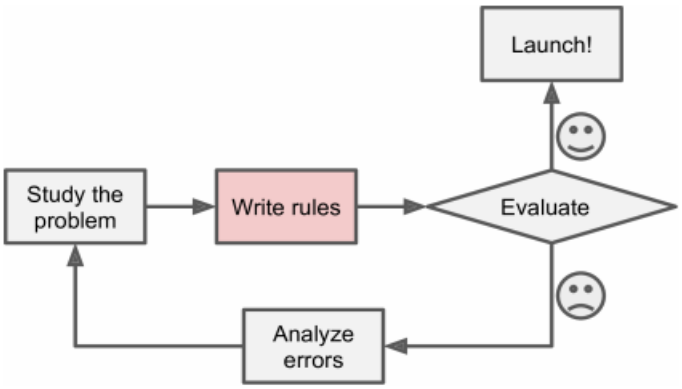


Figure 7 : Méthode de résolution traditionnelle d'un problème de science des données [13]

Pour un problème trivial, il convient de programmer une longue liste de règles à appliquer. Cependant, le contraste est saisissant lorsqu'on utilise les méthodes du Machine Learning pour résoudre un problème. En effet, on n'aura besoin que d'une phase d'entrainement de l'algorithme de ML pour arriver aux résultats attendus. L'approche de Machine Learning peut être résumée par le graphique suivant :

[12] Géron, 2019, P.87-104
[13] Géron, 2019, P.05

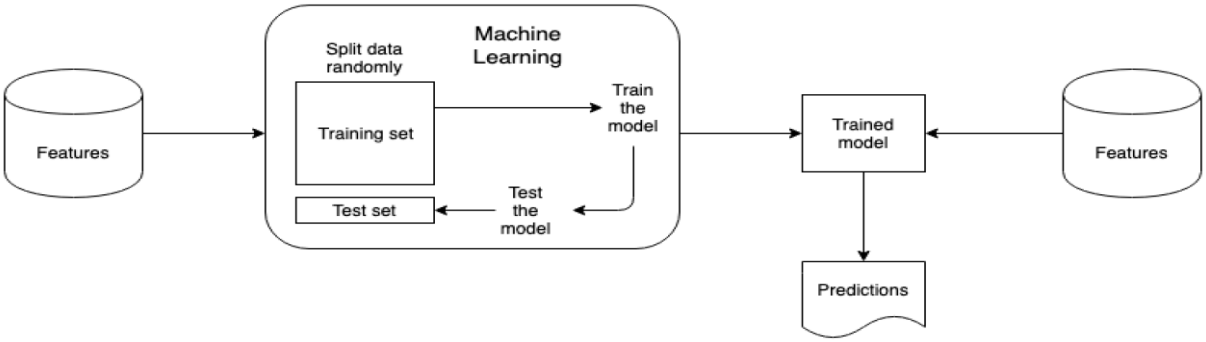


Figure 8 : Approche de résolution de problèmes de données grâce au ML [14]

La phase d'optimisation de l'algorithme est automatique, et permet de faire progresser l'algorithme en simulant un nombre infini de traitements. Le but est donc de déterminer un algorithme capable de prédire un résultat considéré juste, et ce, qu'importent les variables en entrée. Cette construction a lieu en 3 phases : la représentation, l'évaluation et l'optimisation :

- **Représentation** : Il faut tout d'abord définir quelles sont les règles qui régissent le modèle, en termes de variables retenues appelées « features » ainsi que l'algorithme qui transforme les données d'input en données output. Il convient également de procéder à la division de l'ensemble des données en 2 sous-ensembles qui sont le sous-ensemble d'entrainement « training set » ainsi que le sous ensemble d'évaluation « test set ».
- **Évaluation** : la marge d'erreur entre le modèle et les occurrences réelles est calculée, c'est-à-dire entre le modèle théorique et ce qui a été relevé dans la réalité.
- **Optimisation** : Réduire la marge en modifiant les paramètres de l'algorithme afin que la marge mesurée soit la plus faible possible.

Ce cycle est répété jusqu'à l'algorithme atteigne une précision acceptable, permettant de générer des prévisions correctes.

2.3. Types d'apprentissage automatique

Il existe 4 types de Machine Learning utilisés, qui sont : l'apprentissage supervisé, l'apprentissage non supervisé, l'apprentissage semi-supervisé ainsi que l'apprentissage par renforcement. Ils sont catégorisés suivant plusieurs paramètres relatifs à la façon dont la machine apprend :

2.3.1. L'apprentissage supervisé [15]

Dans l'apprentissage supervisé, les données d'entrainement en entrée de l'algorithme comprennent les solutions visées, appelées cible, étiquettes ou « labels ». La machine devra donc apprendre le processus nécessaire afin d'obtenir l'output désiré à partir de l'input. La figure disponible en **Annexe A** permet d'illustrer le fonctionnement d'un algorithme d'apprentissage supervisé.

Il existe 2 types de problèmes en apprentissage supervisé :

- **La régression :** La valeur cible à prédire dans ce cas est continue.
- **La classification :** La valeur cible à prédire est discrète.

[14] Rautio, 2019, P.16
[15] Géron, 2019, P.3-10

2.3.2. L'apprentissage non supervisé

Dans l'apprentissage non supervisé, on dispose d'un « dataset » d'observations notées dans une matrice X, cependant la variable cible n'est pas définie. Avec des étiquettes, la machine apprend à reconnaitre des structures dans les données X qu'on lui montre. En effectuant cela, elle pourra donc regrouper les données dans des clusters (méthode du Clustering), détecter des anomalies ou alors réduire la dimension de données. Ces résultats seront issus d'un étiquetage des solutions qu'accomplira la machine en trouvant des « patterns » communs aux données. On peut retrouver en **Annexe B** un schéma résumant les étapes de l'apprentissage non supervisé, ainsi que des exemples de problèmes utilisant ce type d'apprentissage[16].

Parmi les algorithmes les plus utilisés en marge de l'apprentissage non supervisé, on peut citer :

- Clustering :
  - K-Means
  - DBSCAN
  - Hierarchical Cluster Analysis (HCA)

- Détection d'anomalies :
  - Support Vector Machines à une classe
  - Isolation Forest

2.3.3. L'apprentissage semi supervisé

Quelques algorithmes traitent un jeu d'entrainement partiellement étiqueté, généralement avec une majorité d'occurrences non étiquetées. Ce type de problème rentre dans le cadre de l'apprentissage semi supervisé. Ces algorithmes sont notamment utilisés par des services d'hébergement de photos comme Google Photos qui reconnait des visages dans plusieurs photos d'un utilisateur et qui a besoin que ce dernier attribue une étiquette (un nom) unique à chaque visage, ce qui sert par la suite à la recherche de photos. Ces algorithmes sont souvent une combinaison de techniques supervisées et non supervisées. On retrouve notamment le *deep belief networks* (*DBN*) ainsi que les *restricted Boltzmann machines* (RBMs)[17].

2.3.4. L'apprentissage par renforcement

Ce type d'apprentissage est assez particulier puisque son paradigme est assez différent. En effet, le système d'apprentissage appelé *agent* dans ce contexte, observe son environnement puis est capable de sélectionner et réaliser des actions afin d'obtenir des *récompenses* ou des *pénalités*. L'agent apprend par la suite la meilleure stratégie à suivre, appelée *policy* afin de maximiser ses *récompenses* au fil du temps. Cette *policy* définit les actions que l'agent choisit au cours d'une situation donnée. Le « reinforcement learning » trouve notamment son utilisation dans la programmation de robots pour leur apprendre à se mouvoir. Un progrès assez conséquent a été relevé en 2017 lorsque l'agent AlphaGo a pu vaincre un champion du jeu de plateau Go en appliquant une *policy* apprise suite à l'analyse de millions de parties du jeu. Cette avancée a mené les développeurs de jeux vidéo comme EA Sports à utiliser ces algorithmes dans la programmation de ses opus de simulation de sports en tout genre. Un schéma résumant les étapes de ce type d'apprentissage est à retrouver en **Annexe C**[18].

[16] Saint-Cirgue, 2019, P.15-23
[17] Géron, 2019, P.10-31
[18] Géron, 2019, P.10-31

2.4. Méthodes de l'apprentissage supervisé

Au cours de ce travail, nous aurons à utiliser des algorithmes issus de l'apprentissage supervisé. Différentes méthodes existent ayant différents objectifs, on retient parmi les plus utilisés dans la classification [19]:

- Méthodes de base :
  - Support Vector Machine (SVM)
  - Arbres de décision
  - Régression logistique
  - Réseaux de neurones artificiels
  - K-Nearest Neighbours (KNN)
  - Forêts aléatoires

- Méthodes ensemblistes :
  - Gradient Boosting
  - Extreme Gradient Boosting
  - Adaptative Boosting

Nous aborderons dans cette partie les algorithmes de classification ainsi que leur formalisation mathématique :

2.4.1. Régression logistique

Cette méthode, malgré son nom portant le mot « régression » est une méthode de classification binaire. Nous avons toujours n points en p dimensions représentés par la matrice $X \in \mathbb{R}^{n \times p}$ mais leurs étiquettes sont elles représentées par un vecteur $y \in \{0,1\}^n$ qui représente l'appartenance (1) ou pas (0) à une classe.

Lorsque nous appliquons directement la régression linéaire, nous nous retrouvons dans une impasse car dans un problème de régression, les étiquettes $y^{(i)}$sont à valeurs dans $\mathbb{R}$ alors que 0 et 1 sont des éléments de $\mathbb{R}$. Il faudrait que beaucoup de points de coordonnées différentes aient la même étiquette $f\left(x^{(i)}\right) = 1$ pour tous les points $x^{(i)}$ positifs.

Une fonction linéaire n'est donc pas la meilleure approche dans ce cas. Nous aurions pu considérer de prendre en compte des probabilités proches de 0 et 1 mais comme les probabilités ne se comportent pas linéairement, ce n'est pas optimal non plus car les résultats varieraient entre -∞ et +∞ alors que les probabilités sont comprises entre 0 et 1.

Afin de résoudre ces problèmes, il faut utiliser une transformation logistique. Au lieu de prédire la probabilité que la classe à laquelle appartient un point est la classe « positive » , $p(Y = 1|x)$ directement comme la valeur d'une fonction linéaire en x, nous allons composer cette fonction linéaire avec la fonction logistique donnée ci-dessous :

$$\log istic : \mathbb{R} \to [0,1] , u \mapsto \frac{1}{1 + e^{-u}}$$

[19] Géron, 2019, P.37-41

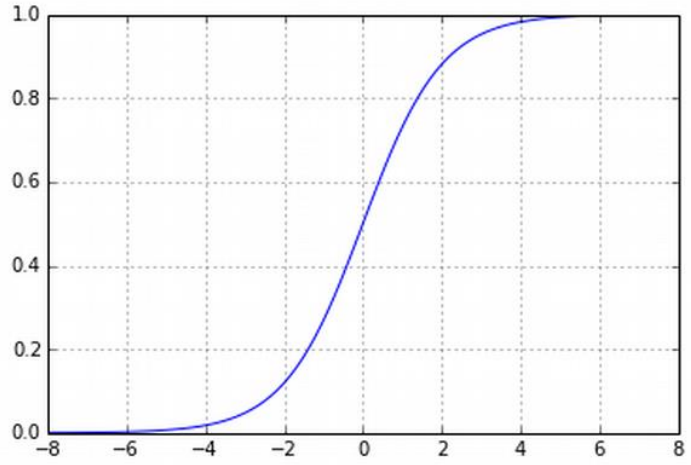


Figure 9 : Variations de la fonction logistique [20]

Nous aurons donc le modèle suivant de la régression logistique :

$$p(Y = 1|x) = \frac{1}{1 + exp\left(-\left(\beta_0 + \sum_{j=1}^{p} \beta_j x_j\right)\right)}$$

Où P(Y) représente la probabilité d'appartenance d'un point à une classe déterminée et β un vecteur regroupant les coefficients du modèle.
La régression logistique peut, tout comme la régression linéaire, s'apprendre par maximum de vraisemblance. Nous aurons donc à résoudre :

$$\arg \max_{\beta \in \mathbb{R}^{p+1}} \left(\sum_{i=1}^{n} \log p\left(Y = y^{(i)} | x^{(i)}, \beta\right) \right)$$

Il y a deux cas possibles pour le calcul de $p\left(Y = y^{(i)} | x^{(i)}, \beta\right)$:

- Soit $y^{(i)} = 1$, donc ce sera $p\left(Y = 1 | x^{(i)}, \beta\right)$
- Soit $y^{(i)} = 0$, ce sera $p\left(Y = 1 | x^{(i)}, \beta\right)$

On peut donc regrouper les deux cas dans le modèle suivant :

$$\log p\left(Y = y^{(i)} | x^{(i)}, \beta\right) = y^{(i)} \log p\left(Y = 1 | x^{(i)}, \beta\right) + \left(1 - y^{(i)}\right) \log\left(1 - p\left(Y = 1 | x^{(i)}, \beta\right)\right)$$

Il nous restera à remplacer $p\left(Y = 1 | x^{(i)}, \beta\right)$ par sa valeur et on obtiendra le problème ci-après :

$$arg \max_{\beta \in \mathbb{R}^{p+1}} \sum_{i=1}^{n} y^{(i)} \log\left(\frac{1}{1 + e^{-\left(\beta_0 + \sum_{j=1}^{p} \beta_j x_j\right)}}\right) + \left(1 - y^{(i)}\right) \log\left(1 - \frac{1}{1 + e^{-\left(\beta_0 + \sum_{j=1}^{p} \beta_j x_j\right)}}\right)$$

2.4.2. Arbres de décision

[20] Openclassrooms,2019

Cette classe d'apprentissage se base sur la représentation des choix sous la forme graphique d'un arbre avec les différentes décisions de classifications placées dans les feuilles.
En théorie des graphes, un arbre est un graphe non orienté, acyclique et connexe. L'ensemble des nœuds se divise en 3 catégories :

- **Nœud racine :** l'accès à l'arbre s'effectue par ce nœud
- **Nœud interne :** les nœuds qui ont des descendants qui sont à leur tour des nœuds
- **Nœuds terminaux (feuilles)** : nœuds qui n'ont pas de descendant
- **Branche :** définit le résultat d'un test effectué sur les nœuds internes

Le problème à résoudre grâce aux arbres de décision est de définir la manière de répartir une population d'individus en groupes homogènes selon un ensemble de variables discriminantes et en fonction d'un objectif fixé qui est la variable cible.

Formalisation : Il faut tout d'abord définir un concept principal : celui de l'homogénéité. En effet, dans un jeu de données, il peut exister plusieurs étiquettes. Lorsqu'on souhaite classifier ces données, plus l'information disponible est en abondance, plus il est difficile de prédire quelle sera la classe d'une occurrence prise au hasard. Lorsque l'information est maximale, il y a donc plusieurs classes (n) en jeu et « l'impureté » des données l'est également. On peut mesurer l'impureté par 2 indicateurs mathématiques qui sont l'entropie détaillé dans la partie **Annexe D** et l'index d'impureté de Gini définis sur n classes qu'on définit ainsi :

- **L'index d'impureté de Gini** : mesure l'inégalité dans un échantillon. Sa valeur est comprise entre 0 et 1 et son interprétation est la même que l'entropie. Sa formule de calcul est donnée par :

$$Gini\ index\ =\ 1\ -\sum_{i=1}^{n}(p_i)^2$$

Il existe trois types d'algorithmes très largement utilisés dans la conception d'arbres de décisions qui sont l'Iterative Dichotomiser 3 (ID3) et 4.5 (C4.5) qui sont détaillés en **Annexe E** ainsi que le Classification And Regression Tree (CART) :

Classification And Regression Tree (CART) : Cet algorithme est utilisé afin de générer des arbres de classification ainsi que de régression. Il utilise l'index de Gini comme métrique principale afin d'évaluer la division de branches dans la sélection des variables dans le cas de la classification. Il est utilisé principalement pour la classification binaire. Dans cet algorithme, il s'agit d'utiliser l'indice de Gini afin de calculer l'impureté existante dans chaque variable, suivant les fréquences d'apparition de ses catégories. C'est un processus itératif et inductif où on retient à chaque fois la variable qui a le minimum d'impureté jusqu'à l'obtention des variables cibles désirées.

Exemple d'un arbre de décision :

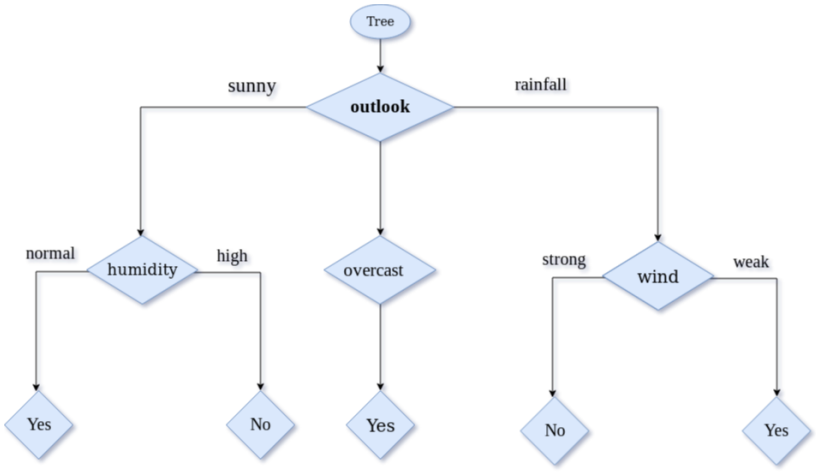


Figure 10 : Arbre de décision pour le problème du jeu de golf [21]

2.4.3. Forêts aléatoires (Random Forests)

Les Random Forests (RF) sont des classificateurs d'ensemble qui développent plusieurs arbres de classification. Chaque arbre est conçu sur un échantillon **bootstrap** de l'ensemble d'entraînement en utilisant une sélection aléatoire des nœuds. RF classifie une instance sur la base des classifications des arbres individuels. La classe qui reçoit le plus de votes est attribuée à cette instance. Les RF protègent contre le surapprentissage ou « overfitting », qui peut arriver parfois avec les arbres de décisions. La technique est capable de fournir une haute performance constante, est très robuste et a un temps de calcul raisonnable. Le seul paramètre à régler est le nombre des variables disponibles pour le fractionnement à chaque nœud.

Afin de formaliser les forêts aléatoires, il faut tout d'abord définir la notion de Bootstrap Aggregating (Bagging) :

La méthode **bootstrap** implique un rééchantillonnage itératif d'un ensemble de données en procédant à des tirages avec remises. Au lieu d'estimer notre statistique une seule fois sur les données complètes, nous pouvons le faire plusieurs fois sur un rééchantillonnage (avec remise) de l'échantillon d'origine. La répétition de ce rééchantillonnage plusieurs fois permet d'obtenir un vecteur d'estimations. Nous pouvons ensuite calculer la variance, la valeur attendue, la distribution empirique et d'autres statistiques pertinentes de ces estimations.

Formalisation : Cet algorithme construit plusieurs arbres de décisions en utilisant des échantillons bootstrappés à partir de l'ensemble d'entraînement, chaque arbre de décision ayant une variance élevée. L'agrégation des différents arbres permet de réduire la variance. Afin d'éviter qu'il y ait une corrélation entre les arbres, l'algorithme sélectionne aléatoirement un sous ensemble de variables à prendre en considération pour chaque arbre, généralement de l'ordre de $m = \sqrt{N}$, N étant le nombre de variables total. L'index de Gini présenté précédemment sert à mesurer l'importance des variables en marge du déroulement des arbres de décision.

On définit par la suite une fonction d'agrégation pour prédire un nouvel individu en utilisant la majorité des votes des arbres de décision dans le cas de la classification. Dans le cas de la régression, la moyenne des résultats obtenus par chaque arbre est calculée.

[21] Medium.com

$$G(x) = Vote\ majoritaire\big(G_1(x), \ldots, G_B(x)\big)$$

Sachant que B est le nombre d'échantillons obtenus aléatoirement à partir de l'ensemble d'entraînement.

On peut illustrer le fonctionnement d'un algorithme de Random Forest ainsi :

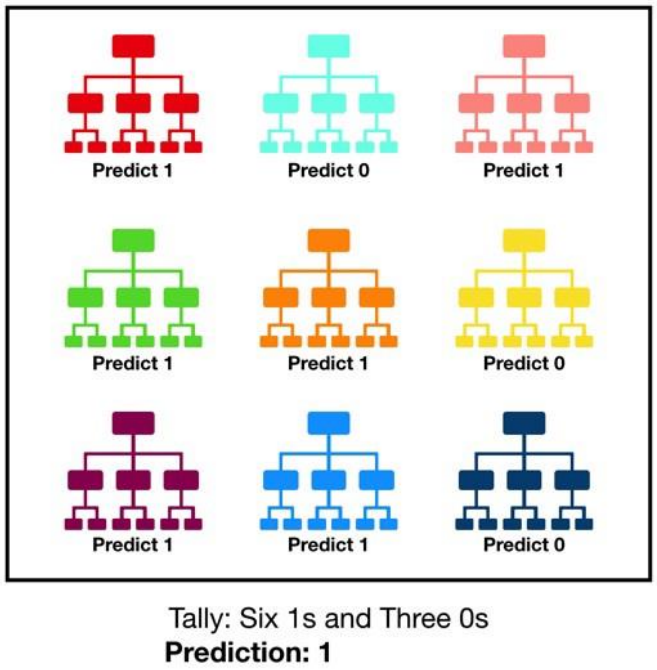


Figure 11 : Vote majoritaire des arbres de décisions pour le Random Forest [22]

2.4.4. Machines à Vecteurs Supports (Support Vector Machines)

Cet algorithme est utilisé afin de spécifier une frontière entre deux groupes. Cette frontière est également appelée « fonction de décision » ou « hyperplan ». Ce dernier est défini en localisant un point depuis lequel la distance à l'élément le plus proche des deux groupes de classification binaire est maximale. Lorsque cet hyperplan existe, il offre une grande marge de classification qui minimise les erreurs de prédiction. Ces marges font fi de vecteurs supports pour la frontière, d'où le nom Support Vector Machine. On peut illustrer le fonctionnement du modèle comme suit[23] :

[22] TowardsDataScience.com
[23] OpenClassrooms.com

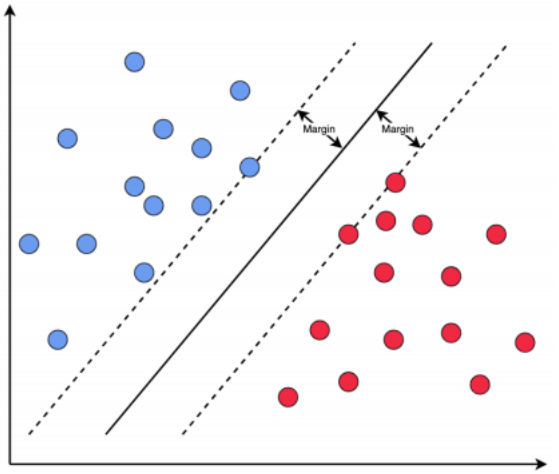


Figure 12 : Support Vector Machines en 2D [24]

Il existe une infinité d'hyperplans séparateurs qui classifient correctement les données. Il faut ainsi formaliser lequel convient le mieux, en définissant la marge d'un hyperplan séparateur H. Ce dernier est délimité par deux hyperplans parallèles à lui et H_+ et H_- qui sont « tangents » aux points les plus proches de chaque coté de la marge γ

Formalisation de l'algorithme : On considère nos données, n points en p dimensions, représentés par une matrice $X \in \mathbb{R}^{n \times p}$ avec des étiquettes représentées par un vecteur $y \in \{-1,1\}^n$, nous utilisons dans ce cas -1 au lieu de 0 pour la simplicité des calculs.
L'équation d'un hyperplan en dimension p est paramétrisée par les coordonnées du vecteur normal à cet hyperplan $w \in \mathbb{R}^p$ ainsi que par un scalaire $b \in \mathbb{R}$ ce qui nous permet d'écrire l'équation de l'hyperplan séparateur de marge maximale $H : <w, x> + b = 0$ tel que $<w, x>$ désigne le produit scalaire entre w *et* x. Donc H dispose d'une équation $\sum_{j=1}^{p} w_j x_j + b = 0$ et les w_j correspondent donc aux β_j utilisés précédemment, et b correspond à β_0.

Nous pouvons par la suite poser les équations correspondantes aux 2 hyperplans parallèles à la « frontière de décision » ainsi :

$H_+ : <w, x> + b = 1$ et $H_- : <w, x> + b = -1$

Afin que les données soient correctement classifiées, les points positifs d'étiquettes $y^{(i)} = 1$ vérifient $<w, x^{(i)}> + b \geq 1$ et les points négatifs d'étiquettes $y^{(i)} = -1$ vérifient $<w, x^{(i)}> + b \leq 1$. Ces 2 conditions peuvent être combinées en une seule :

$$y^{(i)}(<w, x^{(i)}> + b \geq 1) \geq 1$$

La marge γ étant donnée par l'expression suivante $\gamma = \frac{2}{\|w\|}$, nous auront donc le problème d'optimisation suivant :

$$arg \max_{w \in \mathbb{R}^p, b \in \mathbb{R}} \frac{2}{\|w\|_2} / y^{(i)}\left(<w, x^{(i)}> + b \geq 1\right) \geq 1 \forall i \in \{1, \ldots, n\}$$

Ce problème revient à optimiser le problème suivant :

[24] Rautio, 2019, P.23

$$arg \min_{w \in R^p, b \in R} \frac{1}{2} \|w\|_2{}^2 \text{ avec } y^{(i)}\left(<w, x^{(i)}> + b \geq 1\right) \geq 1 \forall i \in \{1, \dots, n\}$$

Ce problème est un problème quadratique primal. En utilisant la technique des multiplicateurs de Lagrange, nous pouvons le formuler en introduisant un scalaire α_i appelé multiplicateur de Lagrange, on aura alors :

$$arg \min_{w \in R^p, b \in R} \frac{1}{2} \|w\|_2{}^2 - \sum_{i=1}^{n} \alpha_i \left(y^{(i)}\left(< w, x^{(i)} > + b\right) - 1\right), avec\ \alpha_i \geq 0\ \forall i \in \{1, \dots, n\}$$

On peut par la suite définir la Lagrangien du problème :

$$\mathcal{L}_p(w, b) = \frac{1}{2} \|w\|_2{}^2 - \sum_{i=1}^{n} \alpha_i \left(y^{(i)}\left(< w, x^{(i)} > + b\right) - 1\right)$$

En annulant le gradient du lagrangien afin de trouver sa solution de minimisation(w^*, b^*) , on trouve un problème d'optimisation de la forme suivante, appelée forme duale :

$$arg \max_{\alpha \in \mathbb{R}^n} -\frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y^{(i)} y^{(j)} <x^{(i)}, x^{(j)}> + \sum_{i=1}^{n} \alpha_i$$

$$avec\ \alpha_i \geq 0\ et \sum_{i=1}^{n} \alpha_i y^{(i)} = 0$$

$$Tel\ que\ w^* = \sum_{i=1}^{n} \alpha_i y^{(i)} x^{(i)} \quad et \sum_{i=1}^{n} \alpha_i y^{(i)} = 0$$

Et donc, en trouvant la solution α^* du problème dual, on peut réécrire la fonction de décision en remplaçant w^* par sa valeur :

$$\sum_{i=1}^{n} \alpha_i \cdot y^{(i)} x^{(i)}\ \ et\ donc\ la\ fonction\ de\ décision\ f(x) = \sum_{i=1}^{n} \alpha_i \cdot y^{(i)} <x^{(i)}, x> + b$$

On peut alors effectuer l'interprétation suivante :

- Les points avec un $\alpha_i = 0$ se trouveront en dehors de la zone d'indécision qui se trouvent à l'intérieur de l'intervalle entre les deux hyperplans H_+ et H_-.
- Les points avec un α_i positif représentent les vecteurs supports de nos hyperplans.

A noter que la formulation du problème sous 2 façons différentes permet de faciliter le processus de résolution du problème. En effet, résoudre le primal est un problème d'optimisation en p dimensions alors que le dual est un problème d'optimisation en n dimensions et donc si p<<n, résoudre le primal sera plus efficace et si n<<p alors résoudre le dual est préconisé.

2.4.5. Réseaux de neurones artificiels (Artifical Neural Networks – ANN)

Les ANN sont des systèmes interconnectés visant à simuler le processus d'apprentissage des neurones biologiques comme ceux présents dans le cerveau humain. Les ANN sont des systèmes adaptatifs très puissants qui peuvent apprendre sans forcément comprendre le but de leur apprentissage. Les ANN sont construits à partir d'unités correspondantes aux données de départ. Elles sont reliées entre elles à travers un groupe de connexions pondérées. Les pondérations modifient l'output d'une couche afin que la couche suivante s'y adapte. L'apprentissage est concrétisé en ajustant les pondérations (poids) appelés « weights ». Une structure d'un réseau de neurone Multi-Layer Perceptron (MLP), qu'on définira par la suite, est présentée ci-après :

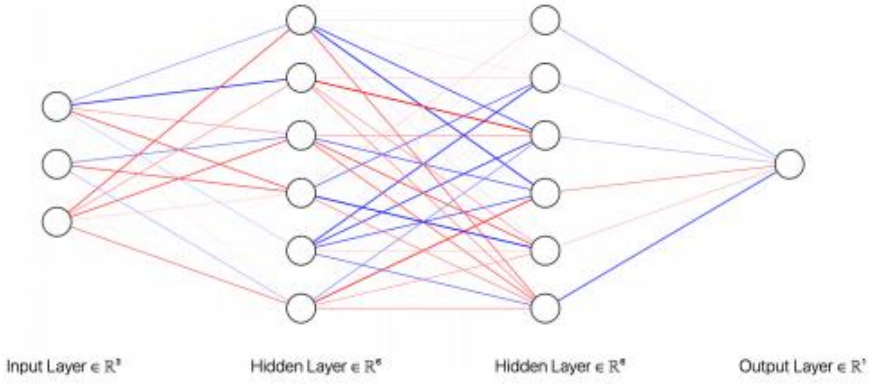


Figure 13 : Réseau de neurones Multi Layer Perceptron[25]

On peut y discerner 3 types de couches différentes : la couche input, les couches cachées ainsi que la couche output. Les nœuds qu'on peut distinguer dans cette représentation sont les neurones et chaque couche contient un certain nombre de neurones qui sont connectés avec les neurones des autres couches, les arcs rouges sont négatifs alors que les bleus sont positifs et leur épaisseur définit le poids de ces arcs.

- **Le Perceptron :**

L'architecture la plus simple des ANN est le « Perceptron », inventé en 1957 par Frank Rosenblatt et basé sur un neurone artificiel appelé « threshold logic unit » (TLU) où l'input et l'output représentent des nombres et chaque connexion de l'input est associée à un poids. Le TLU avec pour output z effectue une combinaison linéaire des inputs suivant la formule :

$$z = w_1 x_1 + w_2 x_2 + \ldots + x_n x_n = x^T w$$

Il applique ensuite une fonction « step » à la somme pour avoir un résultat : $h_w(x) = step(z)$ où $z = x^T w$.

[25] Rautio, 2019, P.20

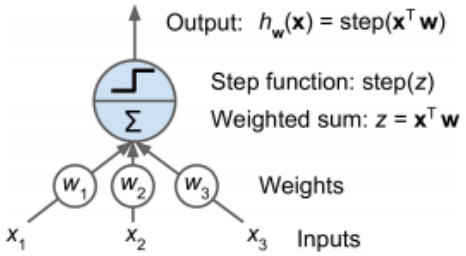


Figure 14 : Perceptron simple [26]

Parmi les fonctions « step » les plus utilisés dans les Perceptrons, on retrouve :

$$heaviside(z) = \begin{cases} 0 \; si \; z < 0 \\ 1 \; si \; z \geq 0 \end{cases} \quad sgn(z) = \begin{cases} -1 \; si \; z < 0 \\ 0 \; si \; z = 0 \\ +1 \; si \; z > 0 \end{cases}$$

On peut obtenir différents outputs pour plusieurs décisions différentes à partir d'une couche d'inputs singulière, en ajoutant un neurone de biais dans la couche d'inputs :

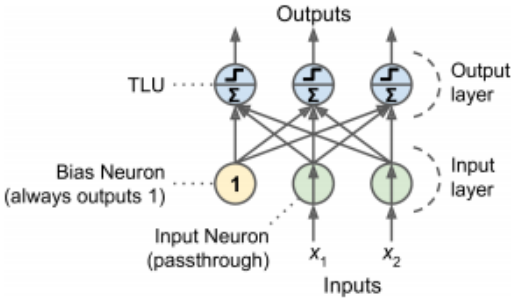


Figure 15 : Perceptron à plusieurs outputs [27]

On aura donc la fonction suivante pour la couche d'output, qui définit les résultats obtenus : $h_{W,b}(X) = \phi(XW + b)$
Où X représente la matrice des données en input, la matrice des poids W contient tous les poids sauf ceux du neurone de biais. Le vecteur b contient les poids entre le neurone de biais et les neurones outputs et la fonction ϕ est connue sous le nom de fonction d'activation. Quand il est question de TLU pour les neurones artificiels, celle-ci est la fonction « step » définie au préalable.

L'apprentissage du Perceptron et des autres ANN est défini par la loi suivante :

$$w_{i,j}{}^{étape\ suivante} = w_{i,j} + \eta \left(y_j - \hat{y}_j \right) x_i$$

Où $w_{i,j}$ est le poids entre le i-ème neurone input et je j-ème neurone output. x_i est la i-ème valeur input pour une instance d'apprentissage. $\hat{y}_j$ est l'output du j-ème neurone pour l'instance d'apprentissage, y_j est la variable cible pour le j-ème neurone output et η est défini comme étant le « learning rate », un hyperparamètre qui est modifié afin de converger vers la solution plus rapidement.

[26] Géron, 2019, P.282
[27] Géron, 2019, P.283

En effet, le poids de connexion entre 2 neurones est incrémenté lorsqu'ils ont le même output, en prenant en compte l'erreur générée par le réseau, ce qui renforce la connexion et réduit l'erreur. Il convient en effet d'insérer au Perceptron une seule instance d'entrainement à la fois et pour chaque instance, on obtient une prévision. Pour chaque neurone output qui génère une fausse prévision, le poids de connexion avec les neurones inputs qui donnent de bons résultats est renforcé.

Le Multi-Layer Perceptron (MLP) : Un réseau neurone plus avancé, avec des propriétés de calcul encore plus puissante est le Multi-Layer Perceptron, qui est décrit dans la figure suivante, composé de plusieurs couches (input, cachés, output), il contient un neurone de biais dans toutes les couches à part la couche output. Longtemps un casse-tête pour les scientifiques, il connait une avancée lorsque l'algorithme de rétropropagation est publié en 1986. Pour chaque instance d'apprentissage, on effectue une prévision de manière directe (forward pass), on mesure l'erreur puis on effectue un retour en arrière pour mesurer la contribution à cette erreur de la part de chaque connexion (reverse pass) puis les poids en sont modifiés afin de réduire l'erreur par la méthode d'optimisation de la descente de gradient « Gradient Descent ».

Afin de permettre à l'algorithme de fonctionner de manière optimale, la fonction d'activation « step » est remplacée par la fonction logistique :

$$\sigma(z) = \frac{1}{1 + exp(-z)}$$

La méthode d'optimisation du « Gradient Descent » donne également de très bons résultats avec 2 autres fonctions d'activation :

- La fonction tangente hyperbolique : $\tanh(z) = 2\sigma(2z) - 1$
- La fonction « Rectified Linear Unit » : $ReLU(z) = max(0, z)$

Le réseau de neurones MLP peut être utilisé pour la régression ainsi que pour la classification binaire en initialisant un seul neurone output avec une fonction d'activation qui est la fonction logistique pour déterminer la probabilité des 2 classes qui sont complémentaires.

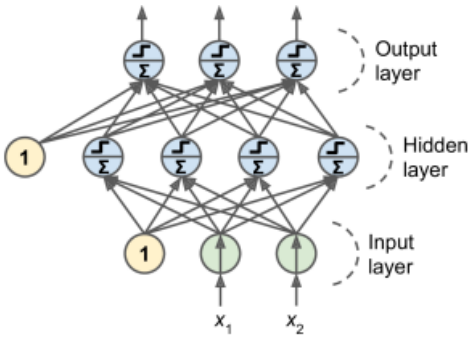


Figure 16 : Perceptron à plusieurs couches[28]

[28]Géron, 2019, P.286

2.4.6. K-Nearest Neighbors (KNN)

L'algorithme des K plus proches voisins ou K-Nearest Neighbors (KNN) est élémentaire mais très important en machine learning. Il peut être utilisé aussi bien pour les problèmes de régression que pour les problèmes de classification. L'attrait de l'utilisation de cette méthode consiste en son interprétation assez intuitive ainsi que pour son faible temps de calcul.

Fonctionnement de l'algorithme : Pour un problème de classification donné, supposons que l'on a deux étiquettes pour classer des points : rouge et bleu. Nous avons un point noir en input, l'algorithme tâchera alors de trouver ses K plus proches voisins et vérifier la couleur de ces voisins. Si la majorité est étiquetée « rouge » alors le point noir sera classé parmi les points rouges. Ces K points sont trouvés par l'algorithme à travers une métrique de distance. Pour les variables réelles en entrée, la plus populaire est la distance Euclidienne, aussi connue sous le nom de Norme 2 entre 2 points p et q :

$$d(p,q) = d(q,p) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + \ldots + (q_n - p_n)^2}$$

D'autres métriques peuvent être utilisées comme la distance de Manhattan qui calcule la distance entre 2 vecteurs en utilisant la somme de leurs différences absolues. On peut également citer la distance de Hamming ou alors la distance de Minkowski qui est une généralisation des distances Euclidienne et de Manhattan.

On peut représenter le processus d'apprentissage par la méthode KNN comme suit :

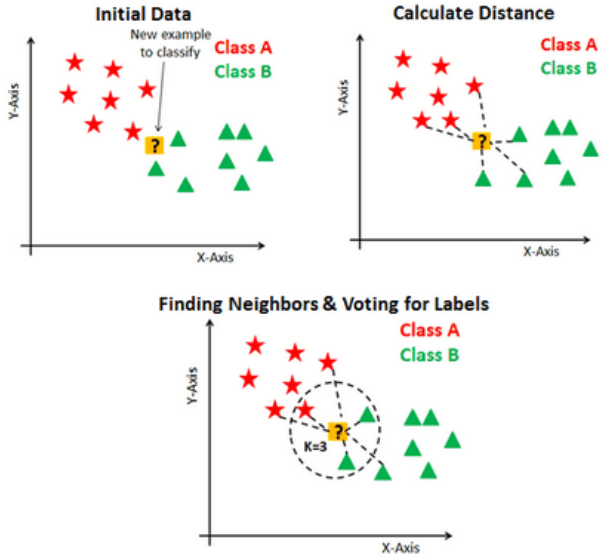


Figure 17 : Classification par la méthode KNN [29]

Un input x à classifier sera assigné à la classe avec la probabilité la plus importante suivante :

$$P(y = j|X = x) = \frac{1}{K}\sum_{i \in A} I\left(y^{(i)} = j\right)$$

[29] DataCamp.com

2.4.7. Apprentissage ensembliste

Le but des méthodes d'ensemble est de combiner les prédictions de plusieurs estimateurs de base construits avec un algorithme d'apprentissage donné afin d'obtenir de meilleures performances prédictives et améliorer la capacité de généralisations / la robustesse par rapport à un seul estimateur.

2.4.7.1. AdaBoost (Adaptive Boosting)

Adaboost est une technique de renforcement très populaire qui vise à combiner plusieurs classificateurs faibles pour construire un classificateur fort. L'article original d'AdaBoost[30] a été rédigé par Yoav Freund et Robert Schapire.

Une façon pour un nouveau prédicteur de corriger son prédécesseur est d'accorder un peu plus d'attention aux instances d'entraînement que le prédécesseur n'a pas pu classer. Ainsi, les nouveaux prédicteurs se concentrent de plus en plus sur les cas difficiles. C'est la technique utilisée par Ada-Boost.

Par exemple, pour construire un classificateur AdaBoost, un premier classificateur de base (tel qu'un arbre de décision) est formé et utilisé pour faire des prédictions sur l'ensemble de formation. Le poids relatif des instances de formation mal classées est alors augmenté. Un deuxième classificateur est formé en utilisant les poids mis à jour et fait à nouveau des prédictions sur l'ensemble de formation, les poids sont mis à jour, et ainsi de suite (voir **figure ci-dessous**).

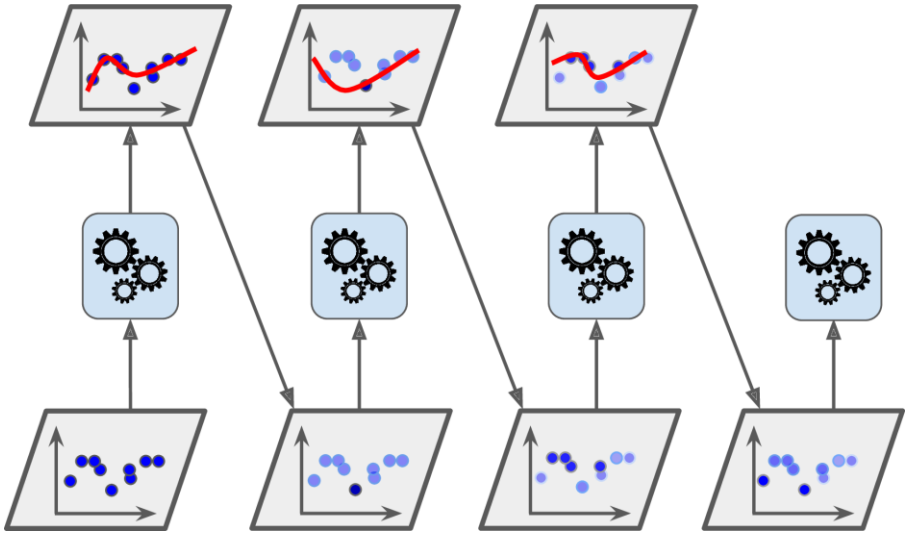


Figure 18 : Entraînement séquentiel AdaBoost avec mise à jour des poids des instances

Pour plus de détails sur les mathématiques derrière l'algorithme d'Adaboost voir **Annexe F**.

Avantages de l'algorithme AdaBoost :

- Très bonne utilisation des classificateurs faibles pour la mise en cascade.
- Différents algorithmes de classification peuvent être utilisés comme classificateurs faibles.
- AdaBoost a un haut degré de précision.

[30] Semantic Scholar, Experiments with a New Boosting Algorithm

Les inconvénients de l'algorithme Adaboost :

- Le nombre d'itérations d'AdaBoost est également un nombre mal défini de classificateurs faibles, qui peut être déterminé par une validation croisée.
- Le déséquilibre des données entraîne une diminution de la précision de la classification.
- Sujet à l'overfitting.

2.4.7.2. XGBoost

XGBoost est le principal modèle pour travailler avec des données tabulaires standard (le type de données stockées dans les DataFrames Pandas, par opposition à des types de données plus complexes comme les images et les vidéos).

Pour atteindre une précision maximale, les modèles XGBoost nécessitent plus de connaissances et de mise au point des modèles que des techniques comme Random Forest.

XGBoost est une implémentation de l'algorithme des arbres de décision renforcés par gradient. Qu'est-ce que les arbres décisionnels à gradient renforcé ? Nous allons parcourir le diagramme ci-dessous.

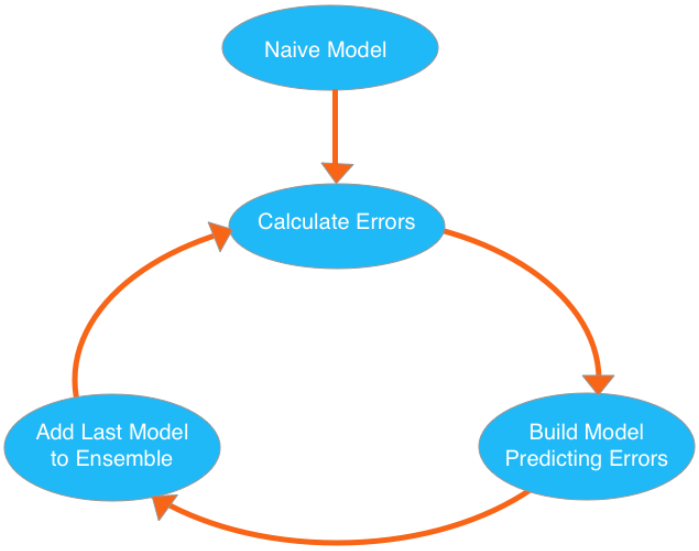


Figure 19 : Cycle XGBoost [31]

L'algorithme passe par des cycles qui construisent sans cesse de nouveaux modèles et les combinent en un modèle d'ensemble. Le cycle débute en calculant les erreurs pour chaque observation dans l'ensemble de données, pour ainsi construire ensuite un nouveau modèle pour les prévoir, enfin, celui-ci ajoute les prédictions de ce modèle de prédiction des erreurs à « l'ensemble des modèles ».

Pour faire une prédiction, XGBoost ajoute les prédictions de tous les modèles précédents et peut utiliser ces prédictions pour calculer de nouvelles erreurs, construire le modèle suivant et l'ajouter à l'ensemble.

Il y a une pièce en dehors de ce cycle. Nous avons besoin d'une prédiction de base pour commencer le cycle. En pratique, les prédictions initiales peuvent être assez naïves. Même si leurs prédictions sont très inexactes, les ajouts ultérieurs à l'ensemble permettront de corriger ces erreurs.

[31] Kaggle,XGBoost

Avantages de l'algorithme XGBoost :

La beauté de ce puissant algorithme réside dans son évolutivité, qui favorise un apprentissage rapide grâce au calcul parallèle et distribué et offre une utilisation efficace de la mémoire.

- Moins de feature engineering nécessaire (pas besoin de mise à l'échelle, normalisation des données, peut également bien gérer les valeurs manquantes).
- Bonne vitesse d'exécution.
- Moins sujet à l'overfitting.[32]

Les inconvénients de l'algorithme XGBoost :

- Overfitting possible si les paramètres ne sont pas réglés correctement.
- Plus difficile à régler car il y a trop d'hyper-paramètres.

2.4.7.3. Gradient Boosting

Le gradient boosting[33] a vu naissance lorsque des chercheurs ont replacé l'algorithme de l'Adaboost dans un cadre statistique plus formel dans lequel il n'était qu'une "simple" optimisation numérique visant à minimiser une fonction de perte de manière successive à l'instar d'une descente de gradient.

Tout comme AdaBoost, le Gradient Boost fonctionne en ajoutant séquentiellement des prédicteurs à un ensemble, chacun corrigeant son prédécesseur. Toutefois, au lieu de modifier les pondérations des instances à chaque itération comme le fait AdaBoost, cette méthode tente de faire correspondre le nouveau prédicteur aux erreurs résiduelles du prédicteur précédent.

Formulation : Pour résumer, l'algorithme du gradient boosting a besoin de 3 éléments principaux :

- Une fonction de perte à optimiser, qui doit être différentiable, qui permet de résoudre le problème.
- Un apprenant faible pour effectuer des prédictions. On peut ne plus utiliser uniquement des souches mais des arbres un peu plus grands, de 4 à 8 niveaux.
- Un modèle additif pour combiner nos apprenants faibles afin de minimiser notre fonction de perte. C'est à dire avancer dans notre fonction de perte en suivant le gradient, ce qui pourra être effectué en ajoutant un arbre de décision supplémentaire. On effectue cette procédure en paramétrant l'arbre, et ensuite en modifiant ses paramètres en allant dans la direction du gradient en diminuant la perte résiduelle que l'on a vu plus haut.

Utilisation du gradient boosting en classification : En classification, la première fonction de perte que l'on peut utiliser est celle originelle, c'est à dire la perte exponentielle, qui permet en fait d'obtenir l'algorithme de l'AdaBoost. Cependant, dans un environnement instable/bruyant, on peut préférer utiliser la fonction appelée déviance binomiale $\log(1 + \exp(-2yf))$, beaucoup moins sujette au variation du dataset. Ce sont les deux fonctions de pertes à utiliser pour une classification binaire.[34]

Avantages de l'algorithme gradient boosting :

- Offre souvent une précision prédictive imbattable.

---

[32] TowardDataScience
[33] ARCING THE EDGE Leo Breiman
[34] Openclassroom

- Beaucoup de flexibilité : peut optimiser sur différentes fonctions de perte et offre plusieurs options de réglage des hyperparamètres qui rendent l'ajustement de la fonction très souple.

Les inconvénients de l'algorithme gradient boosting :

- Coûteux en termes de calcul - les GBoost nécessitent souvent de nombreux arbres (>200) qui peuvent être exhaustifs en termes de temps et de mémoire.
- La grande flexibilité se traduit par de nombreux paramètres qui interagissent et influencent fortement le comportement de l'approche (nombre d'itérations, profondeur de l'arbre, paramètres de régularisation, etc.). Cela nécessite une grande recherche de grille pendant la régularisation.

3. Stratégie de ré-échantillonnage

Les algorithmes d'apprentissage machine ont du mal à apprendre lorsqu'une classe domine l'autre.

Il y a 4 façons de résoudre les problèmes de déséquilibre de classe comme ceux-ci :

- Synthèse de nouvelles instances de classes minoritaires
- Sur-échantillonnage de la classe minoritaire
- Sous-échantillonnage de la classe majoritaire
- Modifier la fonction de coût pour rendre la classification erronée des instances minoritaires plus importante que celle des instances majoritaires

3.1. Sur-échantillonnage : SMOTE

SMOTE signifie Synthetic Minority Oversampling Technique[35], Il s'agit d'une technique statistique permettant d'augmenter le nombre de cas dans l'ensemble de données de manière équilibrée. La technique fonctionne en générant de nouvelles instances à partir des cas minoritaires existants fournis en entrée. Cette implémentation de SMOTE ne modifie pas le nombre de cas majoritaires. (Voir algorithme SMOTE en **Annexe H**).

Les nouvelles instances ne sont pas simplement des copies de cas minoritaires existants ; à la place, l'algorithme prend des échantillons de l'espace de caractéristiques pour chaque classe cible et ses voisins les plus proches, et génère de nouveaux exemples qui combinent les caractéristiques de la cible avec celles de ses voisins. Cette approche augmente les caractéristiques disponibles pour chaque classe et rend les échantillons plus équilibrés.

SMOTE prend l'ensemble des données en entrée, mais il augmente le pourcentage des cas minoritaires seulement. Par exemple, supposons que vous ayez un ensemble de données déséquilibré où seulement 1% des cas ont la valeur cible A (la classe minoritaire), et 99% des cas ont la valeur B. Pour augmenter le pourcentage de cas minoritaires à deux fois le pourcentage précédent, vous devez entrer 200 pour le pourcentage SMOTE dans les propriétés du module.

[35] SMOTE Journal of Artificial Intelligence Research 16 (2002) 321–357

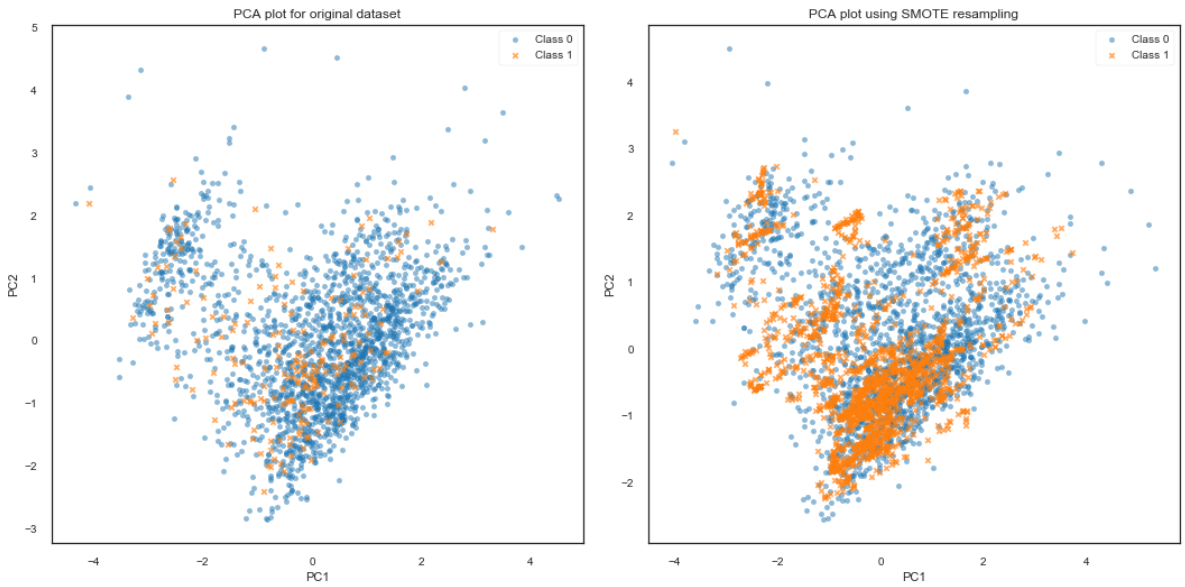


Figure 20 : Analyse en composantes principales sur les données avant et après SMOTE.

4. Evaluation des performances de l'apprentissage supervisé pour la classification

Après avoir mis en place un processus d'apprentissage supervisé pour la classification binaire, valider les résultats permet d'évaluer les performances de l'apprentissage. Pour ce faire, plusieurs critères sont utilisés, en fonction du problème.

Nous allons définir les différentes manières d'évaluation de la performance d'un modèle de classification dans ce qui suit :

4.1. La matrice de confusion

Prenons l'exemple de la classification des emails, différentes situations existent, prédire un spam qui n'en est pas un, ne pas prédire un spam qui en est un, prédire un spam qui en est un, … Ça devient tout de suite assez compliqué. Afin de clarifier les choses, on utilise un outil appelé matrice de confusion :

		PREDICTIVE VALUES	
		POSITIVE (1)	NEGATIVE (0)
ACTUAL VALUES	POSITIVE (1)	TP	FN
	NEGATIVE (0)	FP	TN

Figure 21 : Matrice de confusion

Les étiquettes TP, FP, FN et TN signifient respectivement True Positives (Vrais positifs), False Positives (faux positifs), False Negatives(faux négatifs) et True Negatives(vrais négatifs). Dans le cas de la classification des mails, on note la classe « spam » comme étant celle positive. Si on prédit un spam et que c'en est bien un, on fait une prédiction positive correcte, c'est donc un vrai positif.

Si par contre cette prédiction est incorrecte, ce sera un faux positif. On appelle « erreur de type 1 » les faux positifs et « erreur de type 2 » les faux négatifs.

4.2. Métriques d'évaluation

Ayant obtenu les 4 valeurs TP, FP, FN et TN, on peut dériver d'autres métriques pour évaluer la performance de notre modèle qui sont le rappel « recall », la précision ou alors la F-mesure « F1-score » que nous allons définir :

- **Rappel** : Le rappel ou sensibilité est le taux de vrais positifs, c'est-à-dire la proportion de positifs correctement identifiés. C'est dans notre exemple la capacité qu'a un modèle à détecter tous les spams :

$$Rappel = \frac{TP}{TP + FN}$$

- **Précision** : On peut avoir un bon rappel en ne ratant aucun spam, mais l'information étant incomplète, on définit le critère de précision qui est la proportion de prédictions correctes parmi les points que l'on a prédit positifs. Une forte précision nous indique qu'un qu'il y a peu de faux positifs classifiés par le modèle.

$$Precision = \frac{TP}{TP + FP}$$

- **La F-mesure** (F1-score) : Définie comme étant la moyenne harmonique de la précision et du rappel. Optimiser la F-mesure permet de trouver un compromis entre l'optimisation de la précision et du rappel, car en optimisant seulement le rappel, l'algorithme va prévoir la plupart des exemples reliés à la classe positive mais on aura par la suite beaucoup de faux positifs et donc une précision assez basse. De l'autre côté, optimiser la précision va mener le modèle à prévoir peu d'occurrences comme étant positifs, mais le rappel va être assez bas.

$$F - mesure = \frac{2 \times Rappel \times Precision}{Recall + Precision} = \frac{2TP}{2TP + FP + FN}$$

- **Spécificité** : Autre critère de performance défini comme étant le taux de vrais négatifs, ou la capacité à détecter toutes les situations où il n'y a pas de spam, cette mesure complète le rappel.

$$pécificité = \frac{TN}{FP + TN}$$

4.3. Courbes d'évaluation

Receiver Operating Characteristics Curve (ROC Curve) : La courbe ROC est une manière visuelle de représenter les performances d'un classificateur. On trace donc une courbe qui représente l'évolution du rappel (taux de vrais positifs) aussi appelé True Positive Rate (TPR) en fonction de 1-specificité (taux de faux positifs) aussi appelé False Positive Rate (FPR) qu'on définit par la loi suivante :

$$FPR = \frac{FP}{FP + TN}$$

La courbe ROC trace les valeurs du TPR et du FPR pour différents seuils S de classification. Diminuer la valeur du seuil de classification permet de classer plus d'éléments comme positifs, ce qui augmente le nombre de faux positifs et de vrais positifs. On peut visualiser une courbe ROC par l'illustration suivante :

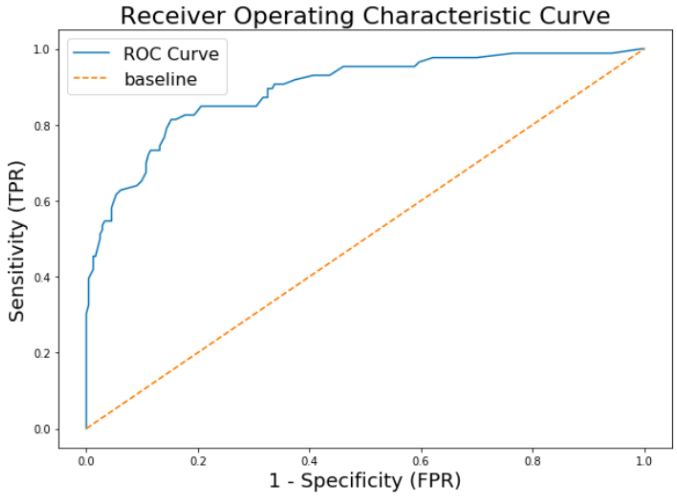


Figure 22 : Courbe ROC

Aire sous la courbe ROC (Area Under Curve AUC) : AUC signifie la courbe sous ROC. Cette valeur mesure l'intégralité de l'aire, à deux dimensions, située sous l'ensemble de la courbe ROC (par calcul d'intégrales) du point (0,0) à (1,1) défini par la fonction $f(x) = x$.

Cette mesure permet de quantifier le degré de séparabilité, en indiquant à quel point le modèle est capable de faire la distinction entre les classes. Plus l'AUC est importante, plus le modèle est prompt à prédire les positifs étant positifs et négatifs étant négatifs.

Concrètement parlant, l'AUC présente les avantages suivants :

- L'AUC est invariante d'échelle, elle mesure donc la qualité de la classification des prédictions, plutôt que leurs valeurs absolues
- L'AUC est indépendante des seuils de classification

Nous pouvons voir dans **l'Annexe G** une représentation de différentes classifications et leurs résultats en comparant leurs courbes ROC respectives, on peut s'apercevoir que plus la courbe de ROC est proche du coin en haut à gauche, plus l'AUC est important et la séparation des classes est fiable. On s'aperçoit également qu'une classification aléatoire donne des résultats qui sont sur la courbe de la première bissectrice du plan : $f(x) = x$.

5. Python

Python est un langage de programmation multi-paradigme : une sorte de couteau suisse pour le monde du codage. Il supporte entre autres la programmation orientée objet, la programmation structurée et les schémas de programmation fonctionnels. Il y a une blague dans la communauté Python qui dit que "Python est généralement le deuxième meilleur langage pour tout".

Python peut prendre en charge toutes les tâches, de l'exploration de données à la construction de sites web en passant par l'exécution de systèmes intégrés, le tout dans un langage unifié.

Facebook, selon un article paru en 2014 dans le magazine Fast Company, a choisi d'utiliser Python pour l'analyse des données parce qu'il était déjà largement utilisé dans d'autres secteurs de l'entreprise.

Python est un logiciel libre et open-source, et par conséquent n'importe qui peut écrire une bibliothèque pour étendre ses fonctionnalités. La science des données a été l'un des premiers bénéficiaires de ces extensions, en particulier Pandas, le patron de toutes ces extensions.

Pandas est la bibliothèque d'analyse de données Python, utilisée pour tout, de l'importation de données à partir de feuilles de calcul Excel au traitement d'ensembles pour l'analyse de séries chronologiques. Pandas met à votre disposition pratiquement tous les outils courants d'analyse de données. Cela signifie que le nettoyage de base et certaines manipulations avancées peuvent être effectuées avec les puissants « DataFrames » de Pandas.

Pandas est construit sur NumPy, l'une des premières bibliothèques à l'origine de la réussite de Python dans le domaine de la science des données. Les fonctions de NumPy sont exposées dans Pandas pour une analyse numérique avancée. Scikit-Learn est une bibliothèque d'apprentissage automatique qui fournit des modules pour la construction de modèles d'apprentissage supervisé et non supervisé ainsi que pour le prétraitement des données.

Tous ces éléments nous ont convaincu d'utiliser ce langage de programmation pour coder nos algorithmes de classification, en marge de la distribution Anaconda permettant d'avoir accès à toutes les bibliothèques nécessaires pour effectuer du Machine Learning.

Conclusion :

Pour conclure, ce chapitre a été l'occasion pour nous de présenter les concepts liés au Machine Learning puis de spécifier le type d'apprentissage utilisé dans le cadre de ce projet qui est l'apprentissage supervisé puis d'apporter quelques notions théoriques sur les algorithmes de classification et les méthodes utilisées pour évaluer les modèles qui en découlent.
La partie suivante abordera l'état des lieux qui inclut le contexte environnemental du projet et qui définira le cadre dans lequel la solution sera conçue.

Partie 2 : État des lieux

Partie 2 : Etat des lieux

Cette partie va aborder une analyse de l'environnement de KPMG et ses différents domaines d'activités et ainsi débouchera sur un diagnostic de la due diligence et les pistes d'amélioration dans l'objectif de définir le périmètre de notre étude et spécifier le cadre de celle-ci.

Chapitre 3 : Présentation de KPMG

Le conseil en stratégie vise à fournir aux dirigeants d'entreprises des conseils spécialisés pour la définition d'une stratégie d'entreprise. En pratique, les cabinets de conseil en stratégie ont une activité qui déborde largement de ce périmètre, et donne lieu à des recommandations sur les volets managériaux, organisationnels et performance.

Les cabinets de conseils en stratégie que sont les Big Four (KPMG, Deloitte, EY et PwC) ont une place grandissante dans ce secteur mais font valoir particulièrement une compétence en stratégie financière.

1. KPMG International

KPMG est un des cabinets leader de l'audit, du conseil et de l'expertise comptable. Aujourd'hui implanté en Algérie, celui-ci nous a permis d'effectuer notre projet dans sa filiale algérienne KPMG Algérie SPA.

KPMG a été fondée en 1987 et a ensuite fusionnée avec d'autres multinationales. Elle est une société commerciale multinationale qui offre des services professionnels à ses clients dans trois domaines spécialisés répartis en 2019 comme suit : le conseil (Advisory) (40 %), l'audit (38 %) et la fiscalité (22 %)[36], tel indiqué sur la figure ci-dessous :

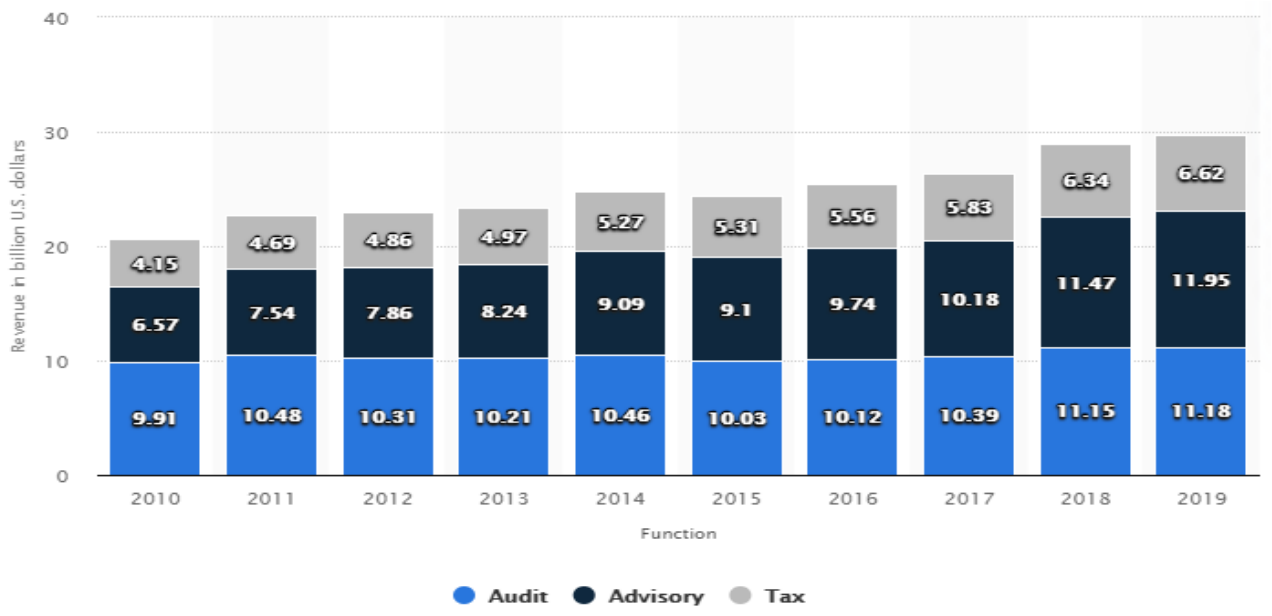


Figure 23 : Chiffre d'affaires de KPMG 2010 à 2019, par activité [37]

Chaque filiale de KPMG est une entité sans personnalité juridique et est membre de KPMG International Cooperative, une entité suisse enregistrée dans le canton suisse de Zoug. Présent dans 152 pays, KPMG dispose d'une connectivité mondiale avec un réseau de 188 982 employés dans le monde entier, comme le montre la figure ci-dessous :

[36] Bloomberg Business, Communiqué de presse 12/12/2019
[37] statista.com

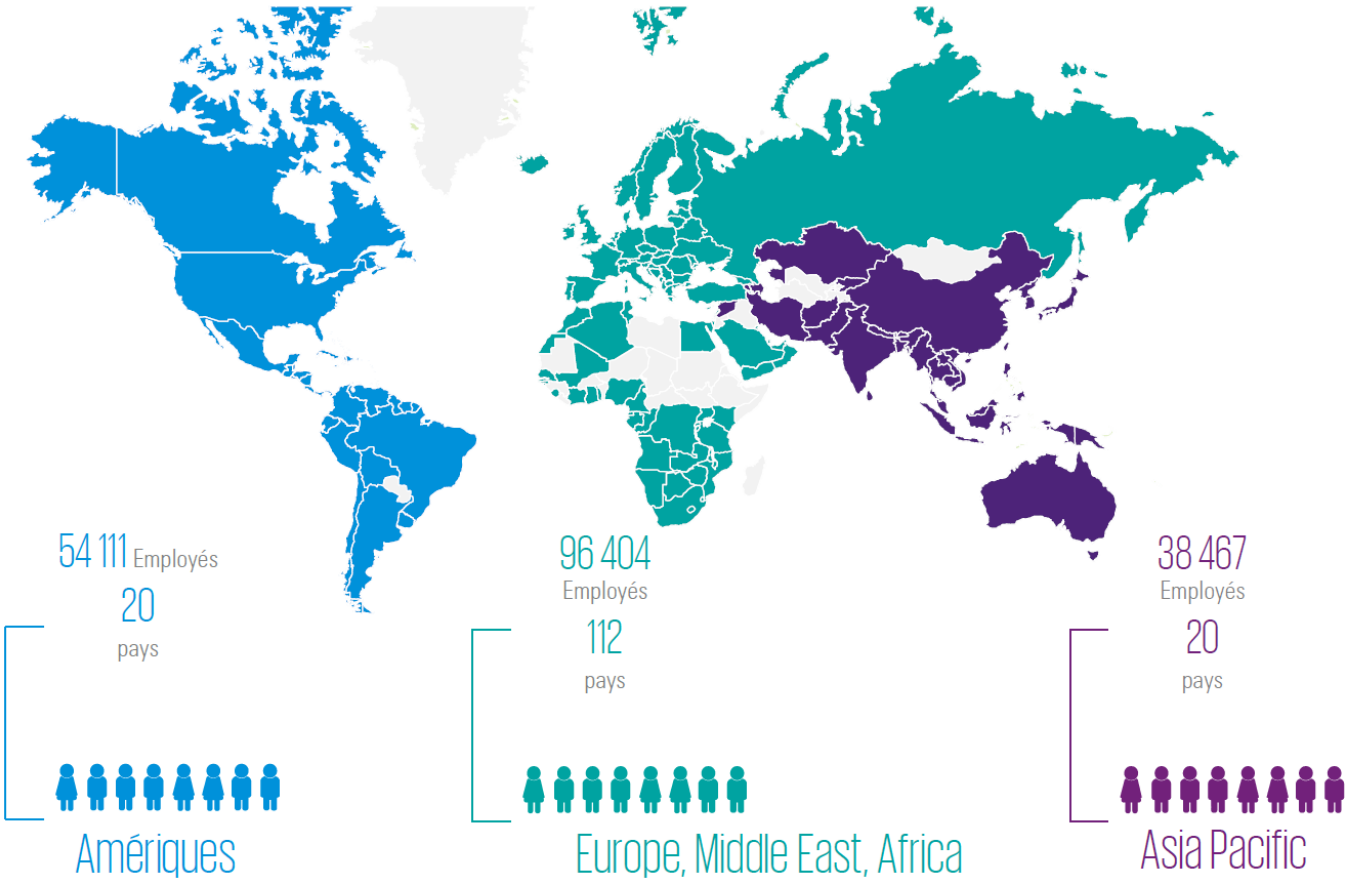


Figure 24 : Présence de KPMG dans le monde [38]

La société partage également son nom avec les "Big Four Auditors" et a été classée 2ème en 2019 dans le classement général des cabinets de conseil par Brand Finance. La société a également été classée 2ème en 2016 dans le classement des meilleurs conseillers en externalisation au monde en plus d'avoir obtenu le rang de 12e sur la liste « Fortune » des 100 entreprises possédant le meilleur environnement de travail (Best Companies To Work For), de 2017, basée sur le bonheur et les avantages des employés.[39]

2. Les domaines d'activités de KPMG International

En plus d'être spécialisé dans des opérations telles que les fusions & acquisitions, KPMG a su répartir son activité sur les trois lignes de services suivantes :

2.1. Audit

Un audit est l'examen du rapport financier d'une organisation par une personne indépendante de cette organisation. Le rapport financier comprend un bilan, un compte de résultat, un état des variations des capitaux propres, un tableau des flux de trésorerie et des notes comprenant un résumé des principales méthodes comptables et d'autres notes explicatives.

L'objectif d'un audit est de déterminer si les informations présentées dans le rapport financier, prises dans leur ensemble, reflètent la situation financière de l'organisation à une date donnée, par exemple :

[38] KPMG Intranet, 2019
[39] Brandfinance.com, fortune.com/best-companies.

- Les détails de ce qui appartient à l'organisation et de ce qu'elle doit aux parties prenantes sont-ils correctement enregistrés dans le bilan ?
- Les bénéfices ou les pertes sont-ils correctement évalués ?

Les audits d'états financiers donnent une assurance sur les informations utilisées par les investisseurs et les marchés de capitaux.

Les recettes d'audit de KPMG International pour l'année 2019 ont atteint 11,18 milliards de dollars, soit une augmentation de 3,7 %.

2.2. Tax & Legal Services

KPMG offre des conseils en matière de fiscalité et de droit, notamment avec des prestations fiscales tels que l'avis fiscal transfrontalier, la restructuration et l'audit fiscal, ou avec des accompagnements juridiques grâce à des services de type mise en œuvre juridique, gestion juridique de l'entreprise, et bien plus encore.

Les revenus des services fiscaux et juridiques ont augmenté de 7,8 % au cours de l'exercice 2019, passant de 6.14 milliards de dollars au cours de l'exercice 2018 à 6,62 milliards, grâce à la forte demande et croissance des services fiscaux multidisciplinaires et juridiques qui a été soutenue par l'élargissement continu de la couverture de l'offre de KPMG.

2.3. Advisory

KPMG offre des services de type « Advisory » qui vont aider ses clients à relever leurs défis et à saisir leurs opportunités cruciales et ainsi développer des solutions innovantes et technologiques afin de résoudre les défis commerciaux auxquels ils sont confrontés et obtenir ainsi des résultats financiers.

Ces services sont sont divisés en 4 sous-domaines :

- **Management consulting :**

Les professionnels du management consulting peuvent aider à identifier et à résoudre les défis d'ordre managérial qui font obstacle à la croissance et aux progrès, on peut citer notamment des défis de transformation numérique, d'extraction de connaissances (Knowledge discovery,), l'établissement de meilleures relations avec les clients, la réduction des coûts, dans le but de stimuler la croissance et de générer de la valeur.

- **Risk consulting :**

Les professionnels du risk consulting apportent l'expérience nécessaire pour aider les entreprises à rester sur la bonne voie et à faire face aux risques qui pourraient compromettre leur survie, tel que :

- Les risques liés aux technologies émergentes.
- Les risques liés à la gestion informatique.
- Les risques réputationnels et les pertes commerciales.
- Les risques financiers.

- **Deal advisory :**

Le consultant en deal advisory intervient tout le long du cycle de la transaction en matière de fusions & acquisitions (parfois aussi appelée « Fusac », ou en anglais M&A, un acronyme pour Mergers and Acquisitions) afin de minimiser tout risque d'échec pour ses clients, et en examinant comment les possibilités d'acheter, de vendre, de s'associer, de financer ou de restructurer une entreprise peuvent ajouter et préserver de la valeur.

- **Stratégie :**

KPMG aide les organisations et les équipes de direction à définir leur ambition et à élaborer des stratégies innovantes qui intègrent l'agilité, l'orientation client et l'excellence opérationnelle nécessaires pour prospérer sur des marchés dynamiques.

KPMG va aider les entreprises à atteindre un rendement supérieur à la norme grâce à des choix stratégiques d'investissement en capital, à la redéfinition de leurs portefeuilles, leurs activités et leurs modèles d'exploitation, et ainsi saisir et mettre en œuvre des opportunités de croissance qui correspondent à leurs ambitions financières.

En 2019, L'Advisory a atteint 11,5 milliards de dollars soit une augmentation de 7,9 % par rapport à 2018, en partie grâce à la forte demande des clients en matière de stratégie, de services liés aux transactions et de solutions de transformation numérique.

3. KPMG Algérie

Les domaines d'activités sont répartis autour de toutes les filiales de KPMG international dont KPMG Algérie que nous allons présenter.

3.1. Présentation de KPMG Algérie

KPMG Algérie SPA, qui est une filiale de KPMG France, opère en Algérie depuis mars 2002 et compte parmi ses clients les plus prestigieuses références locales et internationales.

En plus d'être la première des « Big Four » à s'implanter en Algérie, KPMG est le Leader incontestable dans son domaine d'activité comme le montre la figure ci-dessous :

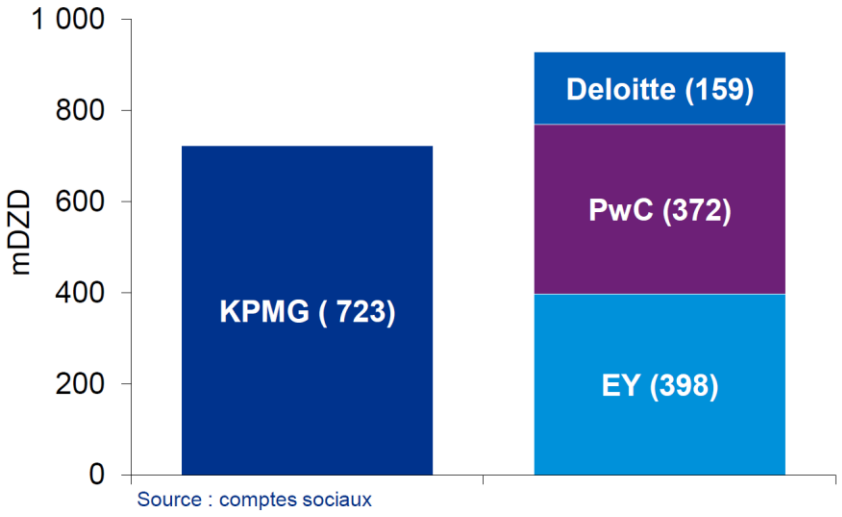


Figure 25 : Chiffre d'affaire KPMG Algérie et concurrents [40]

En 2019, les chiffres clés de KPMG Algérie se présentent comme suit :

- Un chiffre d'affaire qui s'élève à plus de 700m DZD contre un peu moins de 400m DZD de CA pour le 2ème.
- Des capitaux propres qui s'élèvent à plus de DZD 600m contre DZD 16m pour le 2ème cabinet.
- Des investissements de plus de DZD 1500m contre DZD 40m pour le 2ème cabinet.

[40] KPMG Intranet, 2019

KPMG Algérie dispose d'un portefeuille de plus de 100 références dans tous les grands secteurs d'activité économique et financière en Algérie en 2020.[41]

3.2. Structure de KPMG Algérie SPA

Le cabinet est structuré en 5 départements en plus de la division d'Oran comme le dévoile la figure ci-dessous :

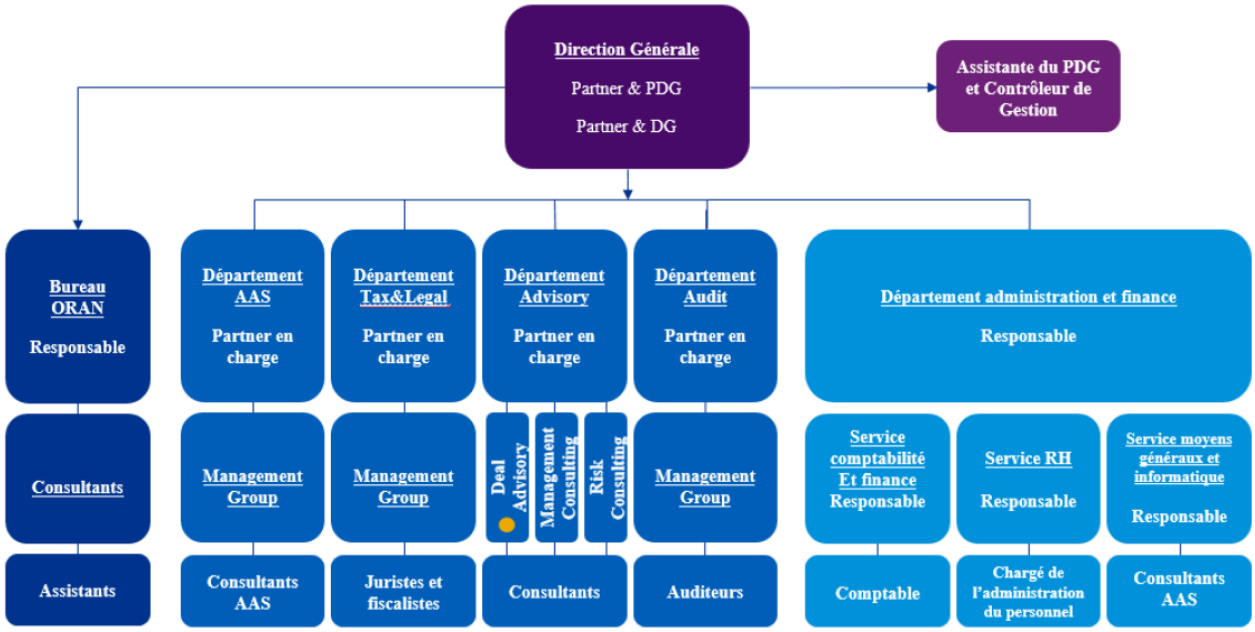


Figure 26 : Organigramme KPMG Algérie SPA [42]

- **Le département Accounting Advisory Services (AAS) :** Les services de conseil comptable des cabinets membres de KPMG se composent d'une équipe spécialisée qui fournit des conseils et un soutien en matière de comptabilité et d'information financière aux clients des cabinets membres, auditeurs et non-auditeurs, sur un large éventail de transactions et d'événements, y compris l'adhésion à des normes comptables nouvelles ou révisées et la gestion efficace des processus d'information financière, afin de les soutenir dans le cadre d'un processus d'introduction en bourse.
- **Département Tax&Legal** : C'est le département qui s'occupe des prestations fiscales et juridiques.
- **Département Audit** : C'est le département chargé des missions d'audits dans le domaine de la finance.
- **Département administration et finance** : C'est le département qui s'occupe des procédures administratives, des moyens généraux, de la comptabilité du cabinet, de la gestion des ressources humaines ainsi que du réseau informatique.
- **Département Advisory :** C'est le département qui s'occupe de l'activité Advisory tel indiqué précédemment. C'est au sein de ce département que nous avons menés notre projet et plus précisément au sein de la cellule Deal Advisory

[41] Guide Investir en Algérie 2020 Par KPMG Algérie
[42] Document Interne KPMG

3.3. Présentation du Deal Advisory

Apparu en Avril 2017, ce service est en expansion continue (Figure 5) et compte aujourd'hui une équipe de 39 collaborateurs, dont des collaborateurs spécialisés, issus pour la plupart de grandes écoles et rassemblant des profils d'exception.

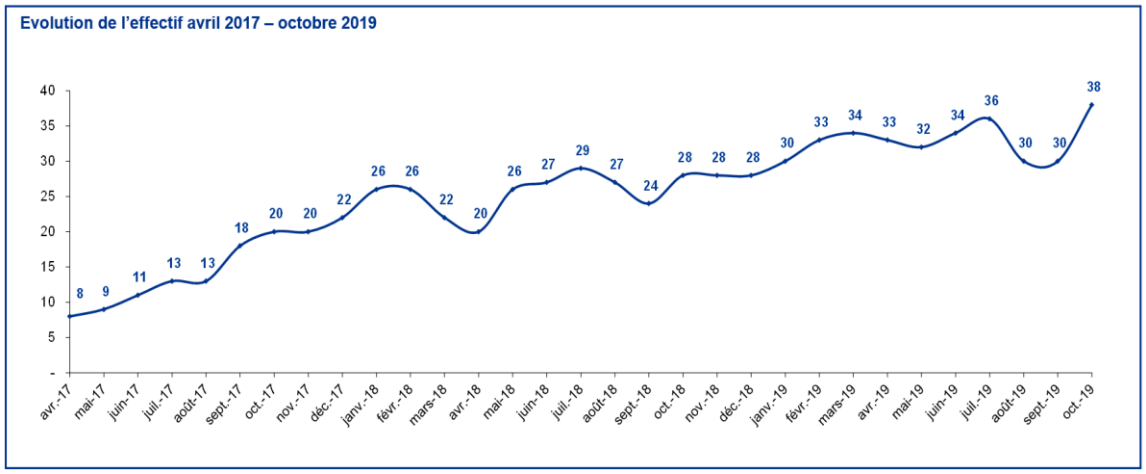


Figure 27 : L'évolution de l'effectif au sein du Deal Advisory [43]

Ce service est réparti en deux divisions Transaction Services (TS) et Deal Analytics (D&A) représentées dans la figure ci-dessous :

Figure 28 : Structure Deal Advisory Alger [44]

L'équipe Deal Advisory d'Alger intervient majoritairement pour le compte de Deal Advisory France et à la marge pour le marché local.
Il existe une double supervision de l'équipe :

- Responsabilité hiérarchique
- Responsabilité fonctionnelle (animation de l'équipe, formation, gestion du planning) par Deal Advisory France.

3.3.1. Le Transaction services

L'équipe « TS » intervient dans le cadre d'une transaction sous 3 domaines d'activités majeurs :

[43] Document Interne KPMG
[44] Document Interne KPMG

- **La due diligence :** est une enquête, un audit ou un examen effectué pour confirmer les faits d'une affaire en cours d'examen. Dans le monde de la finance, la diligence exige un examen des documents financiers avant de conclure une transaction proposée avec une autre partie.
- **L'élaboration des rapports de valorisation** : La valorisation c'est l'estimation de la valeur de l'entreprise dans le cadre de la transaction.
- **L'élaboration des Business Plan financiers / Planification financière** : Les Business Plan financiers sont des projections, sur les années à venir, des différents états financiers afin d'avoir une prévision sur l'évolution des agrégats financiers à moyen et long terme de l'entreprise cible.

3.3.2. Deal analytics

L'équipe « D&A » est une jeune équipe créée après l'équipe TS en fin 2017 et intervient comme activité support à la TS en facilitant le travail de celle-ci grâce à des outils d'automatisation, de programmation et de machine Learning, on peut citer notamment Power BI ou même AlteryX.

Conclusion :

Dans ce premier chapitre nous avons effectués une présentation de KPMG en mettant l'accent sur son environnement et ses domaines d'activités afin situer notre travail.

Nous effectuons notre stage au sein de l'équipe Deal Advisory qui est chargé de la réalisation de la Due Diligence, objet de notre projet. Nous allons donc poursuivre par un diagnostic interne détaillé du processus de la due diligence.

Chapitre 4 : Diagnostic interne et contexte de l'étude

KPMG s'est implantée depuis de nombreuses années comme étant une valeur sure du Deal Advisory, et met en place lorsqu'elle est contactée par un client, un processus complexe d'étude de la transaction dont le moteur est la « due diligence ».

Cherchant constamment à consolider sa place de leader sur le marché, KPMG vise à perfectionner le processus de due diligence en y intégrant de nouveaux indicateurs et des études plus approfondies permettant de faciliter la prise décision, en veillant à réduire le risque d'échec des transactions sur lesquelles elle est appelée à fournir son expertise.

C'est ainsi qu'est né le besoin de mettre en place de nouveaux outils permettant de soutenir la due diligence, notamment ceux basés sur l'utilisation des techniques du Machine Learning.

Nous allons présenter ci-dessous le processus de due diligence déployé par le département TS en marge de l'étude d'opportunité M&A puis nous allons spécifier le contexte de l'étude.

1. Diagnostic interne

La phase d'évaluation financière représente la majeure partie de l'étude d'opportunité d'une fusion ou une acquisition. Elle commence par le processus de due diligence qui se présente ainsi :

1.1. La due diligence chez KPMG

La due diligence effectuée par KPMG a pour objectif final de renseigner le client acquéreur sur ses potentielles cibles grâce à une étude menée sur les états financiers, la situation juridique, les opérations courantes ainsi que les risques inhérents à l'activité de celles-ci.

A la fin du processus, il deviendra possible de fixer une valeur à la cible qui dépend de différents facteurs, sur laquelle s'accorderont toutes les parties concernées afin de pouvoir finaliser la transaction.

KPMG effectue deux types de due diligence :

- Pour le compte des clients vendeurs appelée Vendor Due Diligence (VDD)
- Pour le compte de clients acquéreurs appelée Buyer Due Diligence (BDD) aussi appelée Financial Due Diligence (FDD)

1.1.1. La Vendor Due Diligence

Dans le cas de la VDD, l'analyse est effectuée sur un business et le processus est commissionné par le vendeur. La VDD est identifiée par les caractéristiques suivantes :

- La VDD est un processus objectif.
- Le rôle de KPMG est de mener sa mission à bien en concordance avec sa lettre d'engagement auprès du client vendeur.
- Une couverture des aspects critiques du business : financiers, commerciaux et opérationnels.
- La mission est effectuée avec pour but de satisfaire le besoin d'information des acquéreurs potentiels, il est donc primordial d'effectuer la due diligence du point de vue de l'acquéreur.
- Les rapports sont vérifiés et fiabilisés par le top management.
- Le vendeur est tenu informé en permanence de l'état d'avancement du processus ainsi que des potentielles anomalies trouvées.
- Les acheteurs potentiels sont alimentés avec les mêmes informations.

- Ce processus permet aux acquéreurs de proposer des offres d'acquisition formelles et donc ouvrir le champ aux acheteurs.

L'apport de la VDD aux clients vendeurs : On peut résumer les objectifs des entreprises vendeuses ayant recours au Deal Advisory de KPMG au nombre de 3 qui sont :

- Augmenter la valeur de sa firme
- Garder sous contrôle le processus de due diligence
- Réduire les écarts d'information

On peut alors décliner ces 3 objectifs en sous-objectifs du côté du vendeur, des sous objectifs qui seront remplis par l'approche de KPMG. Nous pouvons visualiser cette complémentarité par le tableau suivant :

Tableau 2 : Approche KPMG pour remplir les objectifs des clients [45]

	Objectifs du vendeur	**Approche KPMG**
Augmenter la valeur de l'entreprise	• Offres initiales élevées • Communiquer la valeur du business • Garder la tension compétitive à un niveau élevé • Mettre en avant un fonds de roulement élevé • Renforcer sa position en minimisant les négociations sur le prix	• Un rapport objectif et clair • Rapport orienté acquéreurs financiers et stratégiques • Quantification des opportunités et risques • Délivrer une information pertinente autour du business • Donner l'opportunité au vendeur de communiquer sur les zones critiques • L'avancée du processus est déterminée par le vendeur
Garder le processus de Due Diligence sous contrôle	• Une connaissance préétablie des points sensibles potentiels • Visibilité continue des résultats de la DD • Contrôle sur le flux informationnel • Communication constante avec les émetteurs d'offres	• Feedback continu sur les problèmes identifiés • Présentation régulière des problèmes de DD rencontrés • Liaison permanente avec le management et les actionnaires
Réduire les écarts d'information	• Réduire les conflits d'intérêts informationnels • Limiter les perturbations internes	• Explication brève du processus • Réunions régulières pour tenir compte du progrès du processus de DD

L'apport de la VDD pour les acquéreurs peut être mis en lumière au travers des points suivants :

[45] Document interne KPMG

- La VDD est un outil apprécié par les investisseurs car il limite le coût d'analyse et permet d'accélérer le processus d'acquisition.
- La VDD optimise le processus de cession en le fluidifiant, l'accélérant et en maximisant ses conditions de réalisation.
- Les informations financières présentées sont transparentes, rigoureuses et fiables ce qui facilite la prise de décision et le financement de l'opération car la VDD est accessible aux banques de financement.
- Optimise la qualité des négociations.

Parties prenantes de la VDD et leurs objectifs respectifs :

Nous pouvons visualiser à travers le tableau suivant les différentes parties prenantes d'une VDD ainsi que leurs objectifs respectifs :

Tableau 3 : Parties prenantes de la VDD et leurs objectifs [46]

Entreprise vendeuse	Management de la cible	KPMG	Autres conseillers	Entreprise acquéreuse
• Garder le contrôle du processus • Avoir un rapport de VDD complet permettant de cerner les problèmes adverses • Présentation formelle des résultats de la VDD aux acquéreurs • Pouvoir interagir avec la partie acheteuse à travers des sessions FAQ	• Comprendre le processus de VDD • Réduire l'impact sur le business • Booster la crédibilité de ses comptes • Résoudre les problèmes avant leur apparition • Atteindre les objectifs prévus	• Accompagner le vendeur dans sa transaction • Dresser un rapport formel sur les états financiers de la cible • Construire des relations avec les autres parties concernées • Améliorer sa réputation	• Alimenter la data room • Augmenter la valeur de la cible • Améliorer les garanties	• Obtenir des informations sur la cible • Obtenir les rapports KPMG • Avoir l'opportunité de faire des sessions de Q&A avec l'équipe chargée de la VDD

[46] Document interne KPMG

1.1.2. Buyer Due Diligence

Communément appelée FDD (Financial Due Diligence), elle représente pour un acquéreur (qui la commissionne auprès de KPMG) le meilleur moyen d'obtenir les informations clés à propos d'un business afin de l'acquérir.

L'importance d'une FDD est à définir sous 3 points essentiels :

1 – Valider une base pour l'évaluation financière :

Cette base couvre les points suivants :

- Elle apporte des informations sur les résultats financiers pour évaluer la cible.
- Elle donne une image fidèle des sources de revenu et de création de valeur de l'entreprise cible.
- Elle permet de cerner des métriques essentielles relatives à la cible comme la qualité des revenus (Quality of earnings) présentée par l'EBITDA entre autres.
- Elle met en liaison l'EBITDA et les flux de trésorerie de la cible afin d'opérer les ajustements nécessaires au modèle d'évaluation financière et finaliser les SPA.

2 – Juger la santé financière de la cible en interprétant ses sources de revenus.

3 – Comprendre la relation qui existe entre l'entreprise et ses capitaux propres, le matelas de sécurité d'une firme, pour juger de sa solvabilité.

Ces 3 points rentrent dans le cadre de la pondération des synergies existantes entre l'acquéreur et sa cible avec pour objectif de les rendre les plus explicites possibles afin de créer de la valeur pour les deux parties.

1.2. Visualisation du processus de due diligence

Nous pouvons résumer les activités mises en place par le département Transaction Services lors de la phase de due diligence par le processus suivant, que nous avons conçu grâce à l'outil Camunda qui permet de modéliser des processus d'affaires (Business Processes) avec la méthode BPMN 2.0 (Business Process Model and Notation).

Ce processus permet de visualiser les 2 types de due diligence effectuées et qui seront définies dans la suite de ce document :

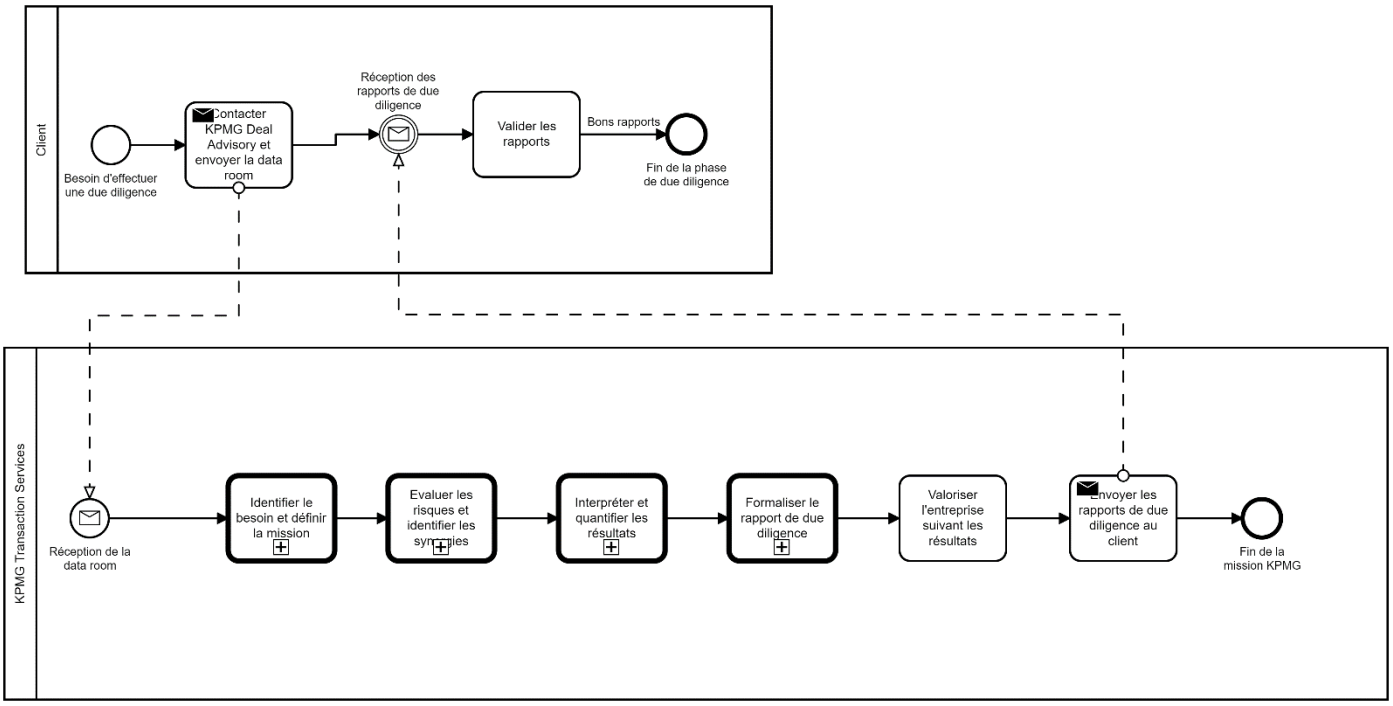


Figure 29 : Processus de due diligence de KPMG Transaction Services

Découlent de ce macro processus plusieurs processus mis en lumière dans la figure qui précède et qui concernent les parties suivantes du processus :

- Identifier le besoin et définir la mission
- Evaluer les risques et identifier les synergies
- Interpréter et quantifier les résultats
- Formaliser le rapport de due diligence

Pour chacun de ces sous processus nous avons dressé une modélisation spécifique, toujours à l'aide de la notation BPMN 2.0 grâce au logiciel Camunda. Nous allons détailler ces modèles dans ce qui suit :

A – Sous processus : Identifier le besoin et définir la mission :

Cette phase est particulièrement importante car elle permet de bien cadrer les limites de l'étude de la transaction afin d'avoir les résultats attendus par le client. Il s'agit de bien comprendre ce dernier, le marché dans lequel il œuvre ainsi que ses activités. Il convient ensuite de définir le contexte de la transaction puis celui de la mission afin de pouvoir organiser le travail en interne et répartir les tâches entre les différentes parties prenantes concernées.

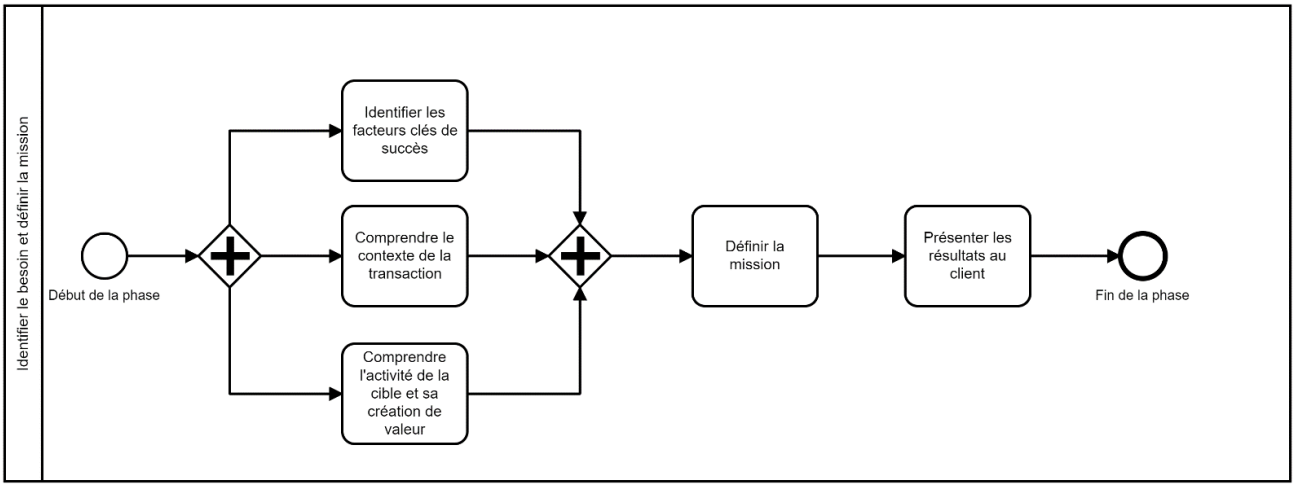


Figure 30 : Sous processus d'identification du besoin et définition de la mission

B – Sous processus : Evaluer les risques et quantifier les synergies :

Lorsque la phase préliminaire a été effectuée, avec des résultats validés auprès du client, il s'agira ensuite d'évaluer les risques et de quantifier les synergies attendues par les parties de la transaction. Il faut veiller ensuite à contrôler la qualité du travail en suivant les référentiels définis en interne par l'organisation KPMG.

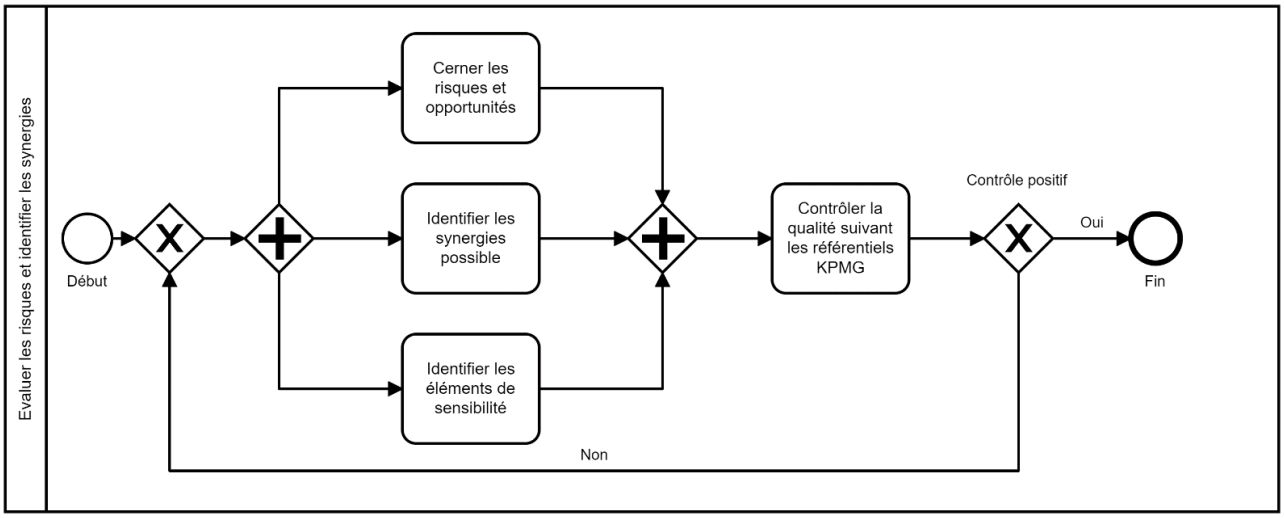


Figure 31 : Sous processus d'évaluation des risques et quantification des synergies

C – Sous processus : Interpréter et quantifier les résultats :

Lorsque la phase précédente est conclue, il faudra interpréter les résultats obtenus en mesurant de manière certaine l'impact qu'auront les rapprochements entre les deux entités, notamment à travers la mise en place de modèles prédictifs sur des indicateurs variés ainsi que sur des objectifs définis au préalable par les parties. Fort de son expérience, KPMG s'appuiera sur ses référentiels définis afin d'évaluer ces indicateurs et mettre en place des modèles qui suivent ses travaux préalables, et s'adaptant au contexte du marché.

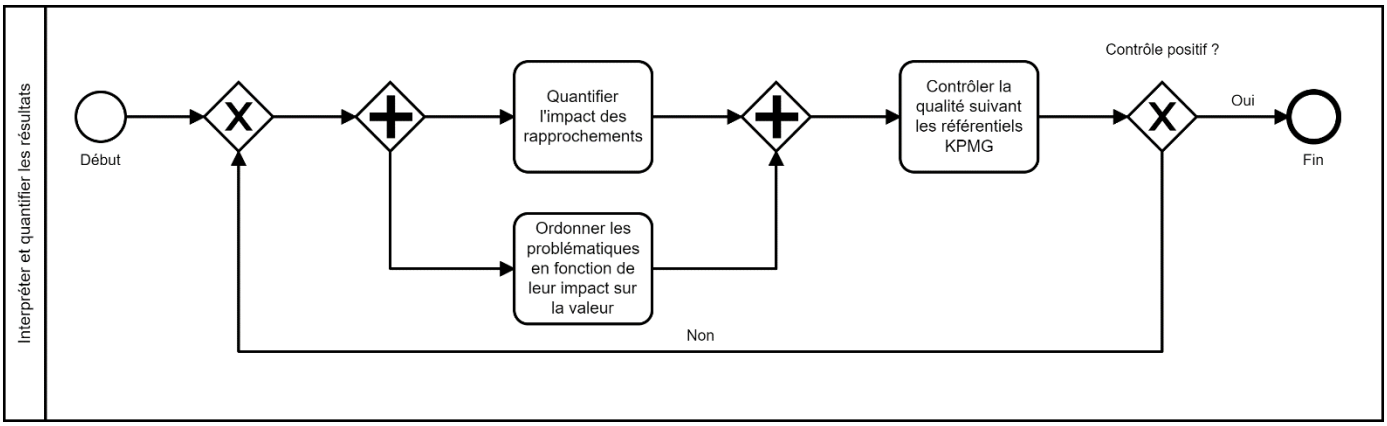


Figure 32 : Sous processus d'interprétation et quantification des résultats

D – Sous processus : Formaliser le rapport de due diligence :

Après avoir effectué plusieurs phases d'étude d'opportunité de l'investissement potentiel, si ce dernier est validé, la phase de due diligence peut commencer, elle couvre plusieurs aspects comme l'aspect commercial, environnemental, financier et fiscal entre autres. La due diligence financière est la partie la plus importante du processus, car elle permet notamment de connaitre des indicateurs financiers sur l'entreprise qui subit le processus et donc d'impacter sur la valuation ainsi que sur la décision d'acquisition, au vu de la santé financière de l'entité.

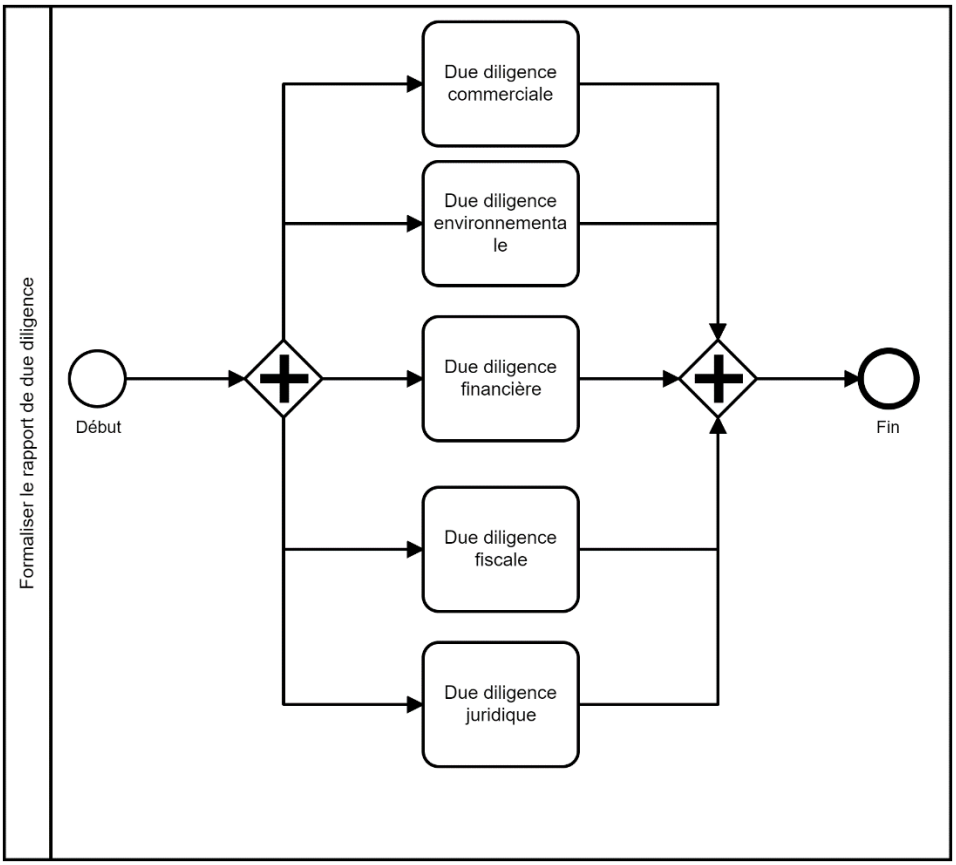


Figure 33 : Sous processus de formalisation du rapport de due diligence

1.3. La phase post due diligence

La finalisation de la phase de due diligence est une condition sine qua non à la fixation du prix de rachat de l'entreprise cible, se basant sur un ensemble d'indicateurs relatifs à la santé financière de l'entreprise cible dont :

- L'indicateur Earnings before interest, taxes, depreciation and amortization (EBITDA) qui renseigne sur la qualité des revenus d'une entreprise sans tenir compte des intérêts, taxes et amortissements.
- Les flux de trésorerie dits Cash Flows (CF)
- La dette nette
- Le fonds de roulement dit Working Capital (WC)
- Le besoin en fonds de roulement dit Working Capital Requirement (WCR)

Ces indicateurs permettent de définir les gains ajustés d'une entreprise. L'EBITDA par exemple, tel que rapporté par le vendeur, peut être impacté par des principes comptables agressifs, des erreurs de classification, des coûts ou bénéfices non récurrents ou inhabituels, etc… En conséquence, le client doit disposer d'une meilleure estimation de l'EBITDA. Dans le cadre des activités du TS, il y a un retraitement de l'EBITDA déclaré en EBITDA ajusté.

Le prix d'acquisition (ou de vente) est déterminé par la méthode des multiples par la formule suivante :

$$Prix\ d'acquisition = Gains\ Ajustés\ \times Multiple - Dette\ nette$$

Cette méthode est basée sur un multiple qui est un coefficient calculé selon le secteur d'activité de l'entreprise et qui prend en considération les concurrents sur le marché. Ce prix d'acquisition est ensuite sensible à un ajustement dont la méthode est définie dans le contrat liant les deux parties.

Les deux parties s'entendent ensuite sur les Sales and Purchase Agreement (SPA) : Contrat portant sur tous les aspects de l'acquisition : périmètre, prix d'acquisition, garanties, date de clôture, ajustement de prix. Ce contrat est généralement préparé par les avocats du vendeur, puis discuté avec l'acheteur. La déclaration de clôture est ensuite émise par l'une des deux parties et qui permet de valider définitivement les états financiers de la cible et définir l'ajustement du prix d'acquisition.

Enfin, la dernière étape qui est celle de l'ajustement du prix final appelée achèvement des comptes et qui est présentée comme une négociation entre les deux parties sur le prix final de la transaction suivant les comptes de la cible. Il existe également une autre méthode pour la détermination du prix final qui est celle de la « boite fermée » ou « locked box » et qui ne tient pas compte de l'ajustement du prix, car le prix final est calculé dès la phase de signature du SPA.

Pour résumer, voici le processus de la phase qui succède à la due diligence :

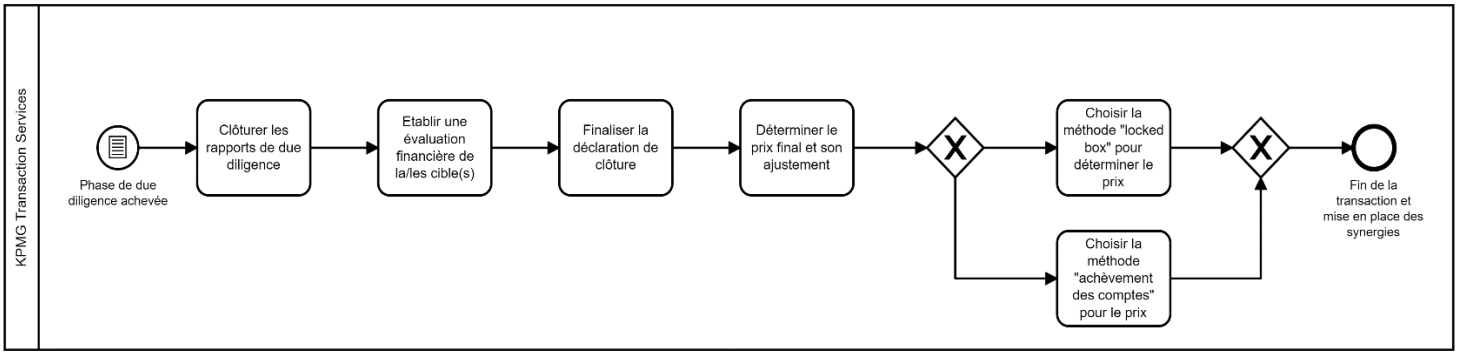


Figure 34 : Processus de la phase post due diligence [47]

1.4. Métriques et risques associés aux M&A

Les M&A sont par nature imprévisibles. En effet, les risques liés au manque d'information d'une des deux parties concernées sur l'activité globale de la cible laisse lieu à un facteur risque important qui a de fortes chances d'être préjudiciable pour l'acquéreur.

Afin de pouvoir éviter ces risques, un certain nombre de métriques est mis à l'étude lors de la phase de due diligence concernant une cible. Il existe 4 grandes classes de métriques, aussi appelés KPI (Key Performance Indicators) qui rentrent dans le cadre de l'étude d'opportunité d'une fusion ou acquisition. Nous pouvons citer ces classes, avec quelques exemples de KPI dans ce qui suit :

A – Les métriques relatives aux employés :

- **Taux de rétention des cadres de l'entreprise** : Ce taux se doit d'être sensiblement élevé afin de garantir que l'entreprise sait garder son personnel important.
- **Index de la santé des employés** : Permet de mettre en lumière des potentiels problèmes de santé liés à la surcharge de travail.

B – Les métriques relatives aux clients :

- **Taux de rétention des clients** : Un indicateur permettant de connaître la capacité d'une entreprise à conserver sa base de clients existante et mesurer son niveau par rapport à un référentiel donné.
- **Coût d'acquisition client** : Ensemble des coûts relatifs à l'acquisition de sa clientèle principalement concernant le marketing.
- **Net Promoter Score** : Le NPS est utilisé pour quantifier la satisfaction de sa clientèle.

C – Les métriques relatives aux processus :

- **Propension au changement** : Métrique relative notamment à la capacité d'une entreprise à s'adapter à la mise en place de bases de données, logiciels ou systèmes d'information.

D – Métriques relatives aux revenus :

- **Monthly Recurring Revenue** (MRR) ou ARR (Average Recurring Revenue) : Une estimation des revenus perçus chaque mois de la part des clients.
- **Burn Rate** : Un indicateur qui permet de quantifier la façon dont une entreprise utilise ses revenus.

[47] Scott Kupor and Preethi Kasireddy, 2014

- **Like for Like analysis** ou **Ventes à données comparables** est une méthode d'analyse financière utilisée pour identifier les produits, divisions ou magasins d'une entreprise qui contribuent à sa croissance et ceux qui accusent un retard.

Nous pouvons résumer ces métriques dans le tableau selon leur catégorie :

Tableau 4 : Métriques associées aux M&A chez KPMG

Catégorie	Employés	Clients	Processus	Revenus
Métriques	Taux de rétention des cadres de l'entreprise	Taux de rétention des clients	Propension au changement	Monthly Recurring Revenue
	Index de la santé des employés	Coût d'acquisition client		Burn Rate
		Net Promoter Score		Like for Like Analysis

L'analyse de ces métriques permet à KPMG de fournir des recommandations à ses clients permettant de limiter les risques relatifs à une transaction. Cependant, on constate un certain nombre de risques qui existent, et qui ne sont pas pris en considération par KPMG, pesant sur la mise en place d'une transaction comme :

- Une absence de prise en compte de l'aspect dynamique des KPI
- Surpayer une entreprise vendeuse lors de la transaction
- Sous-estimation du temps et des ressources nécessaires pour la synergie
- Rigueur insuffisante concernant la due diligence financière

Lorsque ces risques sont pris en compte, ils représentent un potentiel impact sur la réputation de KPMG si la transaction n'aboutit pas à une réussite.

1.5. Mise en place d'un nouveau paradigme de due diligence à KPMG

C'est dans ce contexte que KPMG a mis en place, dans une mesure pour pallier à l'existence de ces risques, une démarche visant à adopter les nouvelles technologies afin d'apporter une valeur ajoutée à ses études réalisées en marge d'une due diligence. Au temps où les bases de données et les data rooms sont de plus en plus conséquentes, les acheteurs sont actuellement sous une pression croissante afin de justifier la viabilité d'une acquisition, avec parfois des contraintes de temps handicapantes. Cependant, l'horizon changeant dans les M&A par la prolifération des outils d'analyse des données permet aux consultants d'affiner leurs analyses, aux acheteurs d'obtenir des informations plus poussées sur leurs cibles ainsi qu'aux vendeurs qui peuvent se mettre en avant plus facilement.

Les services de Deal Advisory de KPMG opèrent donc actuellement une mue vers l'automatisation des traitements de données avec l'implémentation de nouvelles méthodes au sein de leurs activités, ainsi que la mise en place de nouveaux outils de reporting permettant d'avoir une meilleure vue des résultats de la due diligence notamment. Ce besoin tire son origine notamment dans :

- Le manque d'information des acheteurs sur les business de leurs cibles.
- Le manque de temps pour cerner les risques et opportunités relatifs à l'acquisition d'une entreprise.
- La quantité massive de données à traiter.
- Suivre les tendances du marché qui opère une transformation digitale à grande échelle.

C'est pour remédier à ces problématiques qu'a été créée l'unité Deal Analytics (D&A) dont nous faisons partie, et qui a pour objectif de mettre ses connaissances en machine learning, analyse des données et business intelligence au service des activités du département Transaction Services.

L'intégration de nouveaux outils et méthodes se fait tout au long du cycle de la transaction et permet notamment :

- L'utilisation de l'outil Power BI afin de réaliser et visualiser les différentes analyses relatives à la due diligence (analyse Like for like, analyse de marge brute ou analyse de la qualité des revenus).
- La conception d'un script en langage de programmation Python permettant d'effectuer les retraitements des états financiers et les mettre en forme conformément à la charte de KPMG.
- L'utilisation de l'outil Alteryx afin de créer des modèles permettant d'effectuer des retraitements des états financiers suivant un format connu au préalable.
- Mise en place un outil appelé **KPMG TVP** qui permet de recueillir des données internes et externes à la cible afin de permettre au client d'identifier des pistes de création de valeur chez les cibles potentielles à travers des métriques déterminées (MRR, EBITDA par exemple). Cet outil intervient dès la phase de recherche d'opportunités pour concentrer les efforts rapidement et ainsi diriger le processus de due diligence. Son but est d'identifier, suivant un framework bien défini, des cibles à cash-flow positif dans l'environnement de transaction et permet de valider les hypothèses préalablement proposées.
- Implémentation de l'outil de reporting KPMG SPI afin de visualiser de larges quantités données, et effectuant des analyses standardisées relatives aux marges dégagées, à la position géographique et aux produits délivrés par les entreprises. On peut donc avoir accès à différentes métriques relatives, entre autres, aux clients et aux revenus. Parmi les questions auxquelles répond KPMG SPI :
    - Quels sont les clients les plus réguliers d'une entreprise et quelle tendance suivent-ils ?
    - Quels produits connaissent le plus de succès auprès des clients ?
    - Quel serait l'impact sur les marges et sur les ventes en s'implantant à un emplacement donné ?

Suite aux nombreuses avancées opérées dans le domaine de l'automatisation des processus au sein des équipes concernées par le Deal Advisory chez KPMG, un besoin interne a émergé, celui de pouvoir développer au sein de l'équipe Deal Analytics de KPMG Algérie la prédiction d'une métrique qui est le taux de churn ou d'attrition (Churn Rate). Ce besoin puise son origine de plusieurs facteurs :

- Nécessité de suivre les tendances du marché qui s'articule autour des modèles prédictifs

- Amélioration des services proposés par le département Transaction Services
- Absence de progrès significatif auprès des concurrents sur la prédiction de ce KPI
- Fiabilisation de la valorisation des entreprises cibles
- Définition d'un référentiel qui servira de base pour la prédiction d'autres KPI

Le secteur ayant été retenu pour faire cette étude est celui du software, plus communément appelé Saas (Software As A Service). Nous allons prendre soin, dans la partie suivante, de définir le contexte de ce marché des fusions et acquisitions, caractériser la place qu'y prennent les transactions relatives au secteur du software puis justifier l'importance ainsi que le besoin de la prévision de métriques dans ce secteur précisément.

2. Contexte de l'étude

Les M&A sont par nature imprévisibles. En effet, les risques liés au manque d'information d'une des deux parties concernées sur l'activité globale de la cible laisse lieu à un facteur risque important qui a de fortes chances d'être préjudiciable pour l'acquéreur. C'est dans ce contexte qu'a émergé au sein du département TS de KPMG le besoin de renforcer les rapports de due diligence par des métriques dynamiques en voulant adopter un outil prévisionnel permettant de prédire un KPI dans la phase de due diligence. Ce dernier, manquant à l'appel, concerne la rétention des clients de la cible. Nous allons, dans ce qui suit, définir le marché des fusions acquisitions, caractériser la place qu'y prend le software et expliquer pourquoi ce dernier a été retenu par le département dans lequel nous œuvrons (Deal Analytics), pour mener cette étude.

2.1. Secteur du Saas

Les Software as a service sont un nouveau mode de proposition de solution digitales aux entreprises. Contrairement aux éditeurs de logiciels qui proposent des offres de licences à leurs clients suivies d'installations de nouvelles versions ainsi que de la maintenance fréquentielle, les éditeurs de Saas se sont démarqués en proposant des abonnements à un service, qu'ils facturent de manière mensuelle à leurs clients. Cette distinction a conduit les spécialistes économistes à revoir leur modèle d'évaluation. En effet, ces derniers prédisent depuis des années l'explosion de la bulle du marché de Saas, chose qui n'a pas eu lieu du fait de la rentabilité certaine (mais différée) de ces derniers. Les spécialistes se concentrent en effet sur des métriques conventionnelles comme les revenus ou les bénéfices par action. Chose qui ne donne pas de résultats probants sur la santé financières des acteurs principaux de ce marché.

Les éditeurs de Saas constituent des clients récurrents auprès de KPMG Algérie, voulant principalement vendre leur entreprise à des entreprises plus huppées. Les équipes du TS sont donc chargées d'effectuer une due diligence sur ces entreprises, en tenant compte de leur principale force : leurs clients.

2.2. Evaluation de la santé financière d'un éditeur de Saas

Les règles comptables actuelles pêchent dans l'interprétation du mode de fonctionnement des Saas, ces derniers, proposant des abonnements sur plusieurs mois, ne pourront pas reconnaître un chiffre d'affaire annuel au démarrage de l'abonnement. Ce qui fait que l'analyse seule des comptes de résultats n'est pas suffisante pour évaluer un business Saas, dû au non alignement des dépenses (liées à l'acquisition client) et des revenus et cash flows qui sont différés.

L'évolution du cash-flow lié à un client présentée par le graphe ci-dessous permet de remarquer l'impact du non alignement des cash-flows parvenant d'un client. Ce dernier paiera un abonnement chaque mois mais l'éditeur doit satisfaire les dépenses liées à l'acquisition dès le début de

l'abonnement. Nous pouvons visualiser sur le graphe suivant l'évolution, à un horizon temporel déterminé, du cash-flow lié à un client d'un éditeur de Saas :

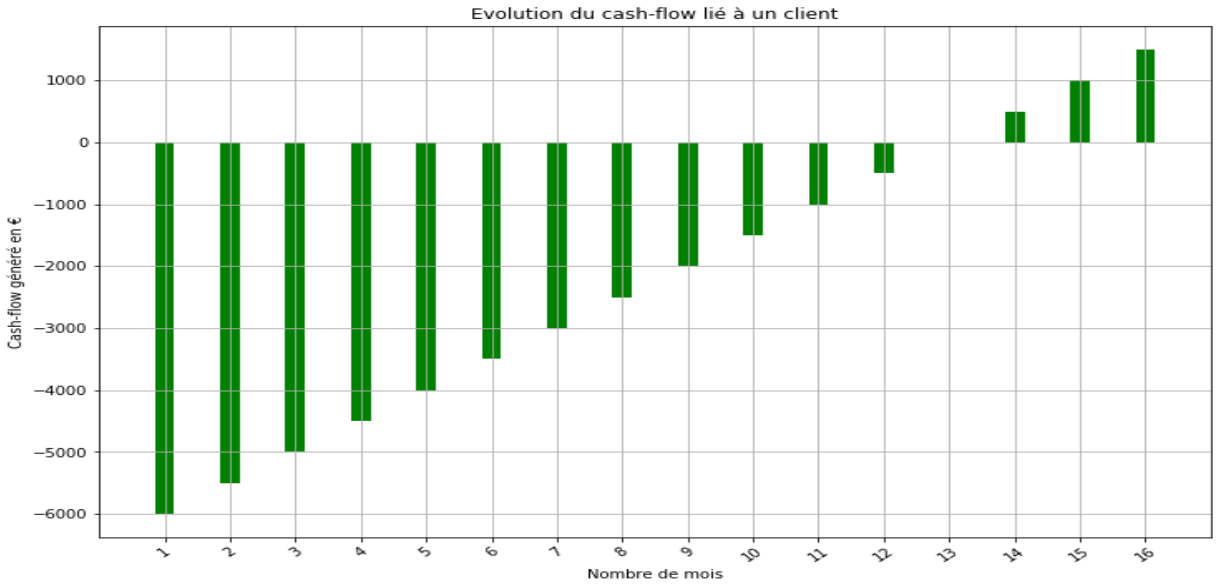


Figure 35 : Evolution du cash-flow lié à un client d'un éditeur de Saas [48]

Le **point d'équilibre** représente le mois où les dépenses liées à l'acquisition du client croiseront les revenus émanant de ce dernier. L'exemple étalé est relatif à un éditeur Saas qui a dépensé 6000 euros pour acquérir un client entre dépenses marketing et logicielles, et qui lui facture un service à raison de 500 euros le mois.

A partir du point d'équilibre, il deviendra nécessaire pour un éditeur de favoriser l'acquisition en amont d'un nombre important de clients. Le but étant de pouvoir dégager une marge de plus en plus croissante au fil du temps, ce qui représente le principal attrait de l'investissement dans les Saas. On peut voir ce phénomène expliqué par le graphe ci-dessous :

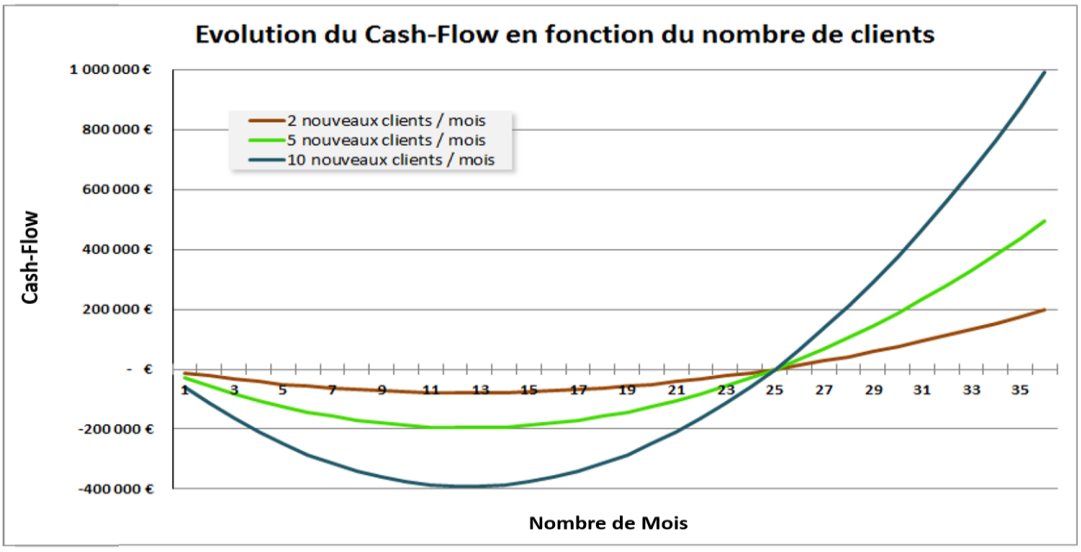


Figure 36 : Evaluation du Cash-flow en fonction du nombre de clients d'un éditeur SaaS [49]

[48] Scott Kupor and Preethi Kasireddy, 2014 (Webographie)
[49] Scott Kupor and Preethi Kasireddy, 2014 (Webographie)

2.3. Le marché des acquisitions de SaaS

De par la mise en lumière du contexte interne et externe à notre projet, nous pouvons déduire les résultats suivants :

- Les entreprises en marge du secteur du software représentent des cibles de plus en plus alléchantes pour les investisseurs au vu de leur propension à devenir des « cash machine » à terme, comme en atteste le graphe suivant, qui révèle un bon dans les transactions dans le secteur de puis l'année 2015 jusqu'au 3ème trimestre de 2019.

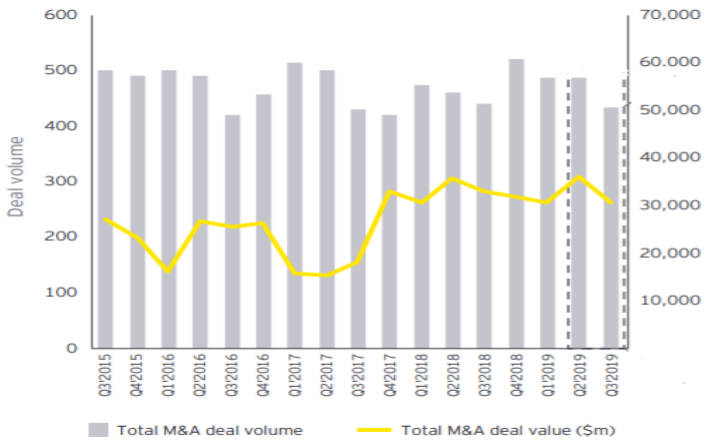


Figure 37 : Evolution du volume et valeur des deals réalisés dans le software [50]

- Le processus de due diligence au sein de KPMG n'inclut pas une mesure prévisionnelle dynamique du taux de churn pour les entreprises du secteur du Saas
- Les investisseurs ont tendance à ne pas fructifier leurs investissements, car leur prise de décision est biaisée par des indicateurs statiques et non dynamiques, ce qui leur permettrait d'entrevoir un gain ou une perte à long terme, et réaliser un arbitrage plus efficace.
- L'évaluation d'une entreprise œuvrant dans les Saas ne peut être effectuée sans tenir compte de ses clients, son principal levier de création de richesse. Cette dimension n'est pas prise en compte dans le processus de due diligence actuel et dans l'évaluation financière d'un éditeur de Saas, ce qui constitue un risque potentiel pour la transaction à terme.
- Le churn rate est une métrique importante permettant aux investisseurs de réaliser un arbitrage afin de délencher le processus de transaction pour acquérir une entreprise oeuvrant dans le secteur du Saas

C'est dans ce contexte que les SaaS (partie intégrante du marché du digital) proposant des services de qualité ont tendance à garder leur client suffisamment longtemps pour dégager une marge opérationnelle suffisante à perdurer et engendrer des profits conséquents après le point d'équilibre, leur permettant d'acquérir de nouveaux clients et ainsi de croître. C'est dans ce sens que les investisseurs s'intéressent fortement aux firmes de ce secteur car elles génèrent des revenus prévisibles permettant de mettre en place une stratégie durable. Les économies d'échelles dégagées par le fait de posséder une seule version exploitable par ses clients, contrairement aux éditeurs de logiciels, permet aux producteurs de Saas d'entrevoir une forte profitabilité.

[50] EY, 2019, (Webographie)

2.3.1. Facteurs d'acquisition

Les investisseurs s'attardent sur plusieurs paramètres qui permettent de faire pencher la balance dans le sens de l'acquisition des éditeurs de Saas. Il existe plus de 50 métriques sur les Saas dans la pratique mais il en existe principalement 4 qui sont récurrentes. On peut les diviser en 3 classes de paramètres qui sont les suivantes :

A- Trouver des clients :

CAC : Coût d'Acquisition d'un client : Il consiste à calculer le total des dépenses de vente et de marketing sur une période données engagées par l'éditeur de Saas et le diviser par le nombre de nouveaux clients signés sur cette même période.

MRR : Monthly Reccuring Revenue : Le revenu mensuel récurrent est la métrique SaaS la plus élémentaire. Il s'agit de votre source de revenus prévisible. Dès le début, vous suivrez cette métrique SaaS dans sa forme la plus basique lorsque des utilisateurs commenceront à s'inscrire. Mais elle deviendra de plus en plus complexe avec le temps. Comme le MRR est influencé par de nombreux facteurs, il faudra le ventiler pour refléter les revenus provenant des nouveaux clients et des renouvellements, et prendre en compte les changements de revenus dus à la mise à niveau, au downgrade ou à la perte de clients.

B – Gagner de l'argent avec ses clients :

LTV : Life Time Value : Le CAC seul ne suffit pas à réaliser un arbitrage efficace. En effet la valeur du CAC doit montrer qu'elle est acceptable et qu'engager un client permet de dégager des bénéfices dans un délai donné. Le LTV vient pallier à ce manque, en déterminant la limite supérieure du CAC comme étant la somme des profits actualisés attendus sur la durée de vie d'un client. La LTV doit être égale ou supérieure à 3 fois le CAC pour que l'entreprise ait suffisamment de temps de dégager des bénéfices sur la durée de vie d'un client.

C – Fidéliser ses clients :

Le Churn Rate : Il permet de renseigner sur le nombre de clients qui se sont désabonnés par rapport aux clients présents. Le Churn est une métrique pertinente pour prévoir la croissance d'une entreprise en faisant en sorte de pouvoir acquérir de nouveaux clients pour combler le vide laissés par ceux perdus. Il est intéressant de relever que conserver sa base de clients existante est plus intéressant pour une entreprise que de dépenser de l'argent pour en acquérir de nouveaux.

Il existe une corrélation établie entre ces paramètres, cependant le Churn Rate est la métrique la plus prépondérante par rapport aux éditeurs de Saas ainsi que pour les potentiels investisseurs. En effet, elle permet de cerner de manière directe la qualité du service proposé. Dans chaque secteur, il existe un taux de churn de référence et si celui d'un éditeur de Saas est plus important que la normale, cela renseigne donc sur sa piètre qualité de service et un investisseur aura donc tous les arguments de son côté pour aller chez un autre éditeur, plus performant.

De ce fait, lorsqu'un éditeur est en phase de croissance, il a intérêt à fidéliser ses clients pour éviter de dépenser plus qu'il n'en faut pour en acquérir de nouveaux, et devra donc s'intéresser principalement au Churn Rate et concentrer ses efforts pour le garder au plus bas niveau possible.

Afin de pouvoir mesurer ces indicateurs donc le Churn Rate il faut tenir compte d'une base de facturation : en effet, s'attarder sur les comptes de résultat ou sur les cash-flows d'un éditeur de Saas n'est pas pertinent. La solution est de s'intéresser à la facturation qui est un indicateur

prévisionnel plus intéressant. La facturation peut être vue comme étant la somme du CA du trimestre de l'année N augmenté des revenus différés du trimestre N-1. La facturation progressera au fur et à mesure que les nouveaux clients affluent.

3. Résumé des constats et justification de la problématique

De par les faits présentés, il nous parait évident que le processus de due diligence au sein de KPMG nécessite une amélioration incrémentale allant vers l'utilisation des outils d'intelligence artificielle et d'automatisation afin de pouvoir prédire des métriques et ainsi améliorer le niveau de service proposé à la clientèle en marge des fusions acquisitions.

D'autre part, et au vu nombre croissant de transactions réalisées dans le secteur du software, ce dernier peut être choisi pour réaliser une étude afin de dégager un modèle prédictif dans le cadre de ce projet pour évaluer sur un horizon déterminé le taux d'attrition (Churn Rate) d'une entreprise œuvrant en B2B.

Le travail que nous effectuerons dans le cadre de ce projet aura donc pour but de développer un modèle permettant la prévision du Churn Rate comme étant un problème de **classification binaire** des occurrences étudiées en marge de la due diligence d'une entreprise cible avec des données financières comme données d'entrées du modèle afin de :

- S'adapter à l'avancement actuel des multinationales KPMG à travers le monde dans le domaine de la digitalisation des processus.
- Profiter de l'absence de firmes locales et françaises spécialisées dans le consulting orienté vers les transactions dans le software en B2B pour développer un modèle.
- Améliorer la qualité de service du département Transaction Services en apportant une valeur ajoutée dynamique au processus de due diligence ce qui aura pour but de réduire le risque d'échec de la transaction.
- Inclure une nouvelle métrique permettant de mieux détailler l'évaluation financière et les ajustements du prix de la cible.

Conclusion :

Pour conclure, cette partie nous a tout d'abord permis de définir le périmètre de l'étude avec la présentation de l'organisme d'accueil, de ses activités principales en se concentrant sur la partie consulting financier.

Nous avons également pu découvrir une vue détaillée des processus mis en œuvre dans le département Transaction Services afin d'établir une due diligence. Nous en avons détecté une piste d'amélioration, qui fera l'objet de notre étude en marge de notre projet, avec un besoin latent existant, celui de suivre le progrès de KPMG vers l'automatisation en mettant en lumière le besoin de mettre en place un modèle prédictif en ce qui concerne le KPI qui est le Churn Rate en citant les facteurs conduisant vers ce choix.

Étant donné la complexité apparente des données, nous avons décidé de recourir à des méthodes issues de l'intelligence artificielle et de la business intelligence, afin de pouvoir cadrer nos ensembles de données selon leur pertinence et utilité, traiter ces données et en ressortir un modèle performant, puis visualiser les résultats.

Nous déroulerons donc dans le chapitre suivant notre démarche de résolution de la problématique, et donc de conception de modèles de Machine Learning.

Partie 3 : Solution proposée et son application

Partie 3 : Solution proposée et son application

L'objectif de cette partie est de répondre à la problématique formulée lors du chapitre précédent. Nous allons donc aborder les solutions taillées afin de répondre au besoin exprimé par KPMG, pour but d'améliorer la performance relative à sa due diligence en prédisant le Churn à travers des modèles informatiques basés sur l'apprentissage automatique (Machine Learning). Les solutions proposées seront détaillées et quantifiées afin de valider leurs performances, en concordance avec les référentiels cités dans les chapitres en amont.

Chapitre 5 : Modèles prédictifs proposés et apports

Afin d'atteindre cet objectif, nous avons choisi de suivre les étapes de la méthodologie de résolution de problème « data » appelée Cross Industry Standard Process for Data Mining (CRISP-DM). Cette méthodologie standardisée de gestion d'un projet de Data Mining (DM) ou de Data Science (DS) comporte l'avantage d'être intersectorielle, et peut donc être implémentée dans un projet de DM ou de DS sans tenir compte du domaine d'étude ou de l'objectif. Notre utilisation de cette méthodologie, très largement plébiscitée par les experts du domaine, revient également au fait de sa capacité à structurer le travail réalisé tout en détaillant chacune des étapes que nous avons effectuées.

Le schéma suivant permet d'expliquer les différentes étapes qui font partie de cette méthodologie, et que nous allons détailler dans ce chapitre afin de dérouler notre proposition de solution.

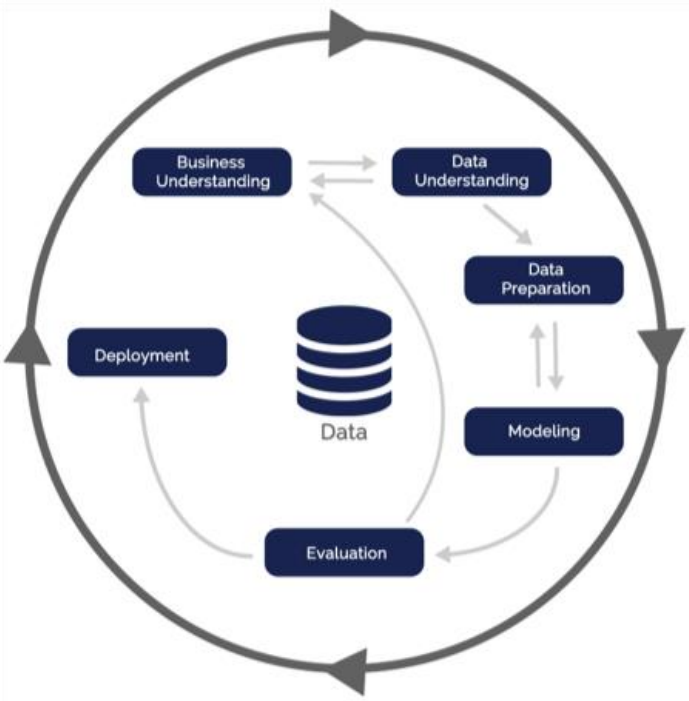


Figure 38 : Méthodologie CRISP-DM [51]

1. Compréhension du projet

Première étape de la méthodologie, son but est de donner un contexte à l'objectif visé par cette démarche de résolution. Celle-ci concerne, dans notre cas, d'apporter à la due diligence effectuée par KPMG dans le secteur du SaaS, une dimension supplémentaire, représentée par la prévision du taux de Churn en résolvant un problème de classification. En effet, dans ce cas, il s'agit pour la firme de se doter d'un outil supplémentaire dans la mise en place d'une due diligence pour le compte de ses clients qui œuvrent dans le secteur du SaaS, en Business to Business (B2B), et donc d'affiner l'étude, dont découlera une meilleure valorisation, menant à des décisions plus quantifiées.

Afin de mieux détailler la phase de compréhension du projet, nous pouvons relever les éléments suivants :

[51] Magrathealabs.com

1.1. Contexte de l'environnement du projet

- La base de données, issue d'une mission antérieure effectuée par le cabinet de conseil KPMG Algérie auprès d'un client éditeur de SaaS et utilisée afin d'effectuer notre projet est composée de 318 lignes, chacune contenant des données réparties sur des colonnes, qui représenteront les variables explicatives, que nous allons présenter dans une partie ultérieure. Ces lignes représentent des observations, occurrences ou enregistrements.
- Les différentes observations représentent des clients ayant un signé un contrat où étaient sous contrat avec l'éditeur de SaaS depuis le mois de Juin 2017 au mois de Septembre 2019.
- On note tout d'abord que la base de données en notre possession présente une série de variables de différents types (catégorielles et numériques) ainsi qu'une variable à prédire (ou expliquée) qui est appelée la variable cible. La cible ou l'objectif est donc d'utiliser nos modèles pour tenter de classer de manière correcte les entreprises qui font parties du groupe « Entreprises ayant churné » ainsi que ceux du groupe « Entreprises n'ayant pas churné ». On colle donc à ces entreprises une étiquette « Churn » ou « Non Churn » respectivement en marge d'une variable appelée « État de l'abonnement » qui prend ces deux étiquettes comme modalités.
- On dénote de par les occurrences à notre disposition que ce problème est à classes déséquilibrées, en effet les occurrences étiquetées « Churn » sont en nombre assez significativement inférieur à celui des occurrences étiquetées « Non Churn ».

1.2. Caractérisation technique du problème

L'objectif technique de notre problème est de concevoir un outil basé sur les algorithmes de l'apprentissage automatique afin de prédire un taux de Churn à partir de différentes variables. Ce problème est défini comme étant un problème de **classification binaire** en marge de l'apprentissage supervisé. En effet, notre cible, qui est la variable « État de l'abonnement », est définie au préalable avec les étiquettes données dans notre jeu de données et nous aurons donc la tâche d'implémenter par le biais de différents algorithmes d'apprentissage supervisé des règles de classification permettant de classer les éléments d'un ensemble donné en 2 groupes.

Nous tâcherons, afin de solutionner ce problème de suivre un plan de travail basé sur la méthodologie standardisée CRISP-DM visant à traiter un problème de classification binaire et donc utilisant des méthodes d'évaluation propres à ce type de problèmes, et spécifiées dans la section évaluation des modèles dans l'état de l'art.

1.3. Plan du projet

Les étapes de mise en place de notre modèle prédictif de Machine Learning seront conformes à celles en marge de la méthodologie CRISP-DM.

Il sera donc question d'abord d'importer notre base de données, de comprendre ses différentes composantes à travers plusieurs techniques de visualisation et d'analyse de fond et de forme. Nous procéderons ensuite au nettoyage des données suivant les remarques de l'étape précédente. L'étape suivante est celle de la modélisation où une série d'algorithmes de classification binaire seront appliquées sur nos jeux de données, leurs performances seront évaluées suivant les métriques mentionnées précédemment. Suivant les résultats offerts par ces métriques, les algorithmes verront leurs hyperparamètres être optimisées afin d'améliorer leurs performances. Une fois les meilleurs modèles choisis, ceux-ci seront soumis à une évaluation finale afin de juger leurs performances, puis soumis à la phase d'implémentation.

Afin de réaliser ces différentes tâches, nous utiliserons le langage de programmation Python, qui dispose d'une bibliothèque puissante afin de résoudre des problèmes de Machine Learning et qui est « Scikit Learn ». Le choix de ce langage est également justifié par la présence de plusieurs bibliothèques facilitant la manipulation, la visualisation et la modélisation des données ainsi que l'optimisation et l'évaluation des modèles construits.

Nous pouvons visualiser les différentes étapes de notre plan de résolution du problème proposé à nous sur le schéma suivant :

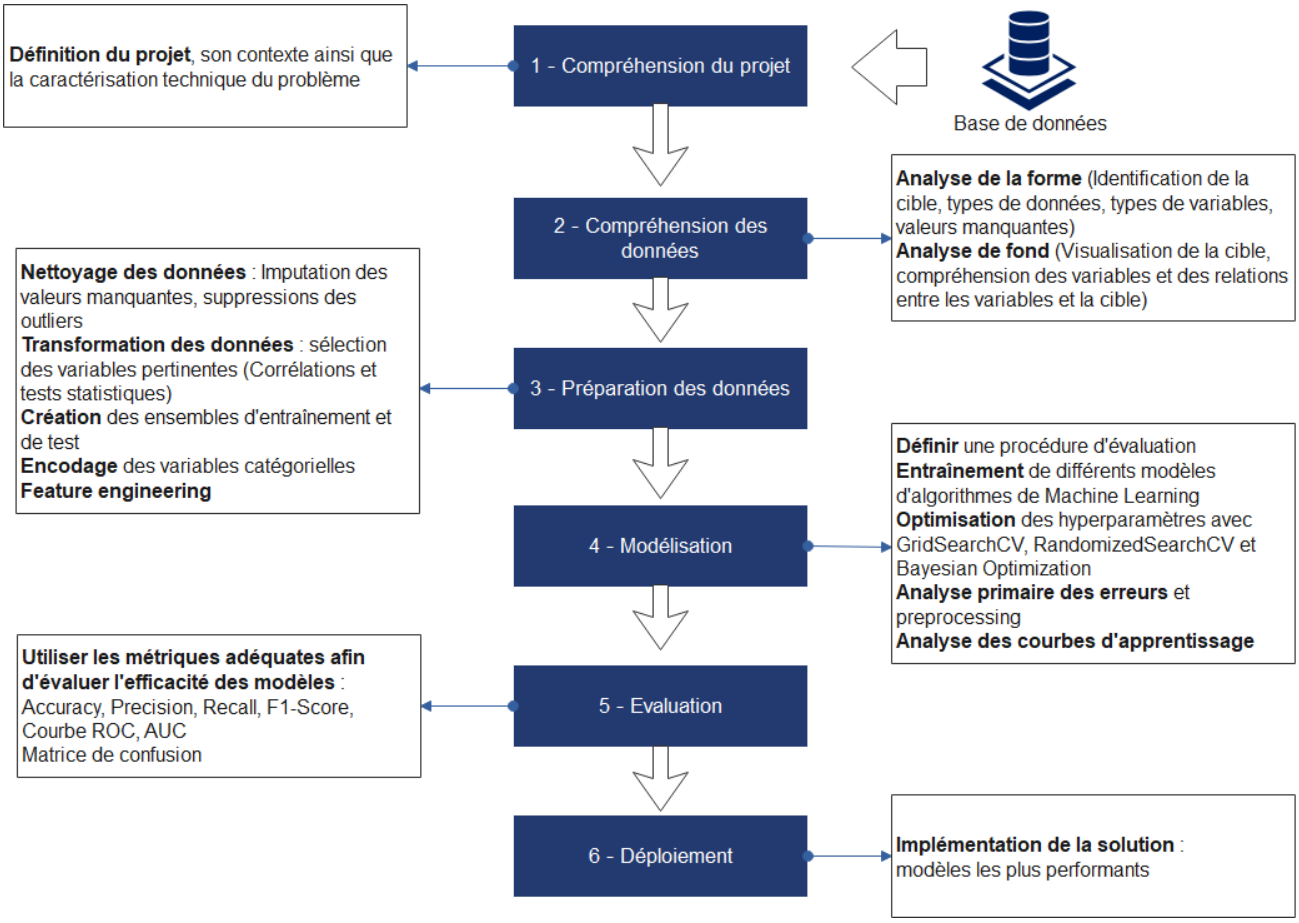


Figure 39 : Etapes détaillées de la méthodologie CRISP-DM

2. Compréhension des données

Cette étape va nous permettre d'appréhender la base de données brute pour mieux l'analyser après l'avoir épurée.

2.1. Base de données brute

Nous allons, dans cette étape, cerner les éléments constitutifs de la base de données à notre disposition. L'objectif de cette étape est de de savoir ce qui peut être attendu et réalisé à partir des données. Il s'agira de vérifier la qualité des données suivants différentes dimensions : la complétude, les distributions des variables ainsi que les relations existantes entre celles-ci et entre les variables et la cible étudiée.

Cette étape est cruciale pour le projet car elle permet d'émettre des hypothèses primaires sur les résultats attendus. Les hypothèses découlent des multiples observations qui sont effectuées à travers les visualisations de différents phénomènes, qui seront explicités par la suite. On peut alors avoir une idée primaire sur l'état actuel de données pour définir l'état désiré, et qui pourra servir d'input pour différents algorithmes de Machine Learning.

Les données mises à notre disposition sont sous format « Excel » et sont importées par le biais de la fonction de lecture des données de la bibliothèque de Python appelée « Pandas » qui transforme les données du format tableur Excel au format « DataFrame », qui facilite le traitement des données. L'étape de visualisation est fortement facilitée par l'utilisation des bibliothèques Python qui sont « MatPlotLib » ainsi que « Seaborn ».

Nos données se présentent sous la forme d'un nombre de lignes représentant des entreprises, organisées sous formes de colonnes représentant différentes variables, que nous avons pu obtenir avec l'aide de l'équipe Deal Analytics. Ces variables explicatives se présentent ainsi :

- **Customer** : Variable de type « chaine de caractère » correspond aux entreprises concernées par les enregistrements.
- **Product Types** : Variable de type catégorielle nominale qui correspond au type de logiciel concerné par le contrat entre les deux parties.
- **Product Types 2** : Variable de type catégorielle nominale, relative à l'utilisation effective du logiciel par l'entreprise.
- **Utilisateurs** : Variable de type numérique correspondant au nombre d'utilisateurs pouvant utiliser le logiciel dans une entreprise donnée.
- **Durée du contrat en cours** : Variable de type numérique nous fournissant le nombre de mois restants au contrat actuel entre les deux parties.
- **Date de signature du contrat** : Variable de type date qui nous indique la date de signature du contrat entre les deux parties.
- **Date de renouvellement du contrat** : Variable de type date indiquant la date à laquelle le renouvellement du contrat est prévu.
- **Région** : Cette variable de type catégorielle nominale nous fournit des informations sur la région de provenance de l'entreprise.
- **Tenancier :** Variable de type catégorielle qui nous indique le fournisseur du SaaS pour chaque entreprise.
- **MRR** : Il s'agit d'une variable numérique donnant les montants de Monthly Reccurring Revenue (MRR) dépensés par les entreprises chaque mois en marge de leur abonnement du mois de Janvier 2017 au mois de Septembre 2019.
- **Type de contractualisation** : Variable binaire qui nous renseigne si la contractualisation est effectuée de manière directe ou avec un intermédiaire.
- **Revendeur** : Variable « chaîne de caractère » qui nous renseigne sur le revendeur si la contractualisation se fait de manière indirecte.
- **ARR Estimate** : Variable numérique qui nous donne une estimation de l'Annual Reccuring Revenue (ARR) fournit par chaque entreprise, normalisée sur une année.

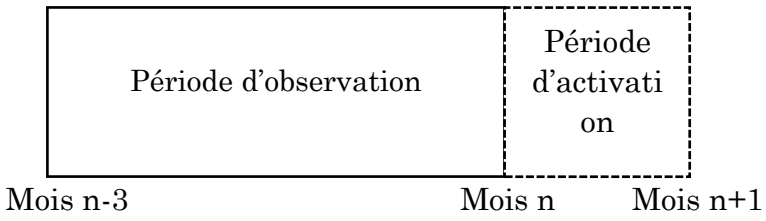
Les données fournies par ces différentes variables sont des données brutes et peuvent contenir plusieurs anomalies, valeurs manquantes ou variables non pertinentes, qu'il s'agira de modifier afin d'obtenir une base de données propice à servir d'input à notre processus de Machine Learning. La qualité des données étant primordiale pour obtenir des modélisations robustes, nous avons donc pu relever les informations suivantes des variables qui sont à notre disposition, et qui nous servirons à construire une base de données pertinentes, sujette par la suite aux analyses de fond et de forme :

- **Valeurs manquantes :** Les cellules vides de notre base de données ont été remplacées par des valeurs nulles, ce qui nous permet de visualiser les valeurs manquantes relatives à chaque variable. La fonction de Python « IsNa » nous permet d'obtenir un masque booléen qui nous retourne toutes les valeurs manquantes de notre dataset. En incluant ce masque booléen dans la fonction « heatmap » de la bibliothèque « Seaborn », nous obtenons la visualisation disponible en **Annexe I** :

Les valeurs manquantes étant représentées par des blancs, on peut remarquer leur forte présence, notamment concernant les MRR de chaque enregistrement. Ce problème peut être résolu en spécifiant une période d'observation pour chaque enregistrement. Cette période d'observation servira à cerner le comportement du client pendant celle-ci, en ce qui concerne le MRR. Nous avons choisi de prendre une période d'observation de 3 mois, pour une prédiction d'horizon de 1 mois. La logique derrière cette solution est que les clients ont un certain comportement, assez commun pendant le trimestre avant qu'ils ne se désabonnent. Choisir une période plus longue pourrait mener à des problèmes de saisonnalité.

L'horizon choisi, étant de 1 mois est appelé période d'activation, et nous allons donc relever les étiquettes des deux classes « Churn » et « Non Churn » sur ce mois-là.

Nous nous retrouvons donc avec le framework temporel suivant, qui sera commun à chaque enregistrement :



Nous procéderons donc en sélectionnant pour chaque entreprise les MRR des 3 derniers mois précédant leur désabonnement, et pour le cas des entreprises ne s'étant pas désabonnés, les 3 derniers mois disponibles (de Juin 2019 à Aout 2019). Quant à la période d'activation, elle nous indiquera la classe à prédire pour chaque entreprise et donc la cible que le modèle de classification binaire développé par nos soins essaiera de prédire.

D'autres variables comme la durée du contrat en cours, la date de renouvellement du contrat ainsi que les revendeurs connaissent également des valeurs manquantes.

- **Transformation, suppression et ajout de variables :** Toujours dans la logique de pouvoir obtenir une base de données pertinentes, nous pouvons remarquer au gré de la visualisation de notre base de données, dont une partie est présentée en **Annexe I** que la variable « Product Types 2 » ne contient qu'une seule catégorie, elle n'est donc sujette à aucune variabilité et n'a donc aucune influence sur la cible. Elle est donc éliminée en utilisant la fonction « Drop » de la bibliothèque « Pandas ».

La manipulation des dates n'étant pas possible pour les « Features » dans Python, nous avons transformé la variable « Date de signature du contrat » en une variable numérique qui indique le nombre de jours écoulés depuis la date de signature à une date fixée de manière aléatoire durant le mois de Mai 2020.

La présence de nombreuses valeurs manquantes pour la variable « date de renouvellement du contrat » nous amène à supprimer également cette variable dans notre traitement des données, toujours en utilisant la fonction « Drop ».

Voulant maximiser la représentativité des variables, et suivant la littérature qui renseigne sur certaines variables permettant d'expliquer le comportement d'une entreprise, nous avons choisi, en collaboration avec l'équipe « Deal Analytics » d'effectuer un « Scrapping » de données supplémentaires, qui nous serviront à modéliser le comportement des clients, à des fins de prédiction. Nous avons entamé ce travail en faisant quelques recherches sur les données disponibles et accessibles via des sites internet spécialisés. Après plusieurs itérations de l'ensemble de nouvelles variables à « Scrapper », nous avons convenu que les variables suivantes étaient susceptibles d'être accessibles en grand nombre pour une majeure partie de nos enregistrements :

- **Capital social** : Variable numérique représentant le montant du capital social de l'entreprise
- **Secteur d'activité** : Variable « chaîne de caractères » qui a été par la suite transformée en variable catégorielle nominale en suivant le référentiel suivant :
    - **Secteur primaire :** Entreprises œuvrant dans les extractions de ressources de la terre et l'agriculture ce qui prend en compte le pétrole.
    - **Secteur secondaire :** Ce secteur regroupe les activités liées à la transformation des matières premières avec par exemple les industries de l'ingénierie, de la production électrique et de l'aéronautique entre autres.
    - **Secteur tertiaire :** Ce secteur comprend toutes les autres activités ne faisant pas partie des deux premiers secteurs, comme les activités de l'assurance, de l'enseignement, de la grande distribution et du commerce entre autres.

- **Effectif moyen pour l'année qui précède le Churn** : Variable numérique qui indique le nombre d'employés de l'entreprise sur l'année précédant le Churn.
- **Chiffre d'affaires pour l'année précédant le Churn** : Variable numérique
- **La durée d'activité de l'entreprise :** Une variable de type date à la base représentant la date de début de l'activité de la firme en question, que nous avons transformé en variable numérique représentant le nombre de mois d'activité de l'entreprise depuis sa création.
- **Taille de l'entreprise** : Variable catégorielle ordinale qui représente la taille d'une entreprise. Nous nous sommes référés à la norme de taille des entreprises françaises concernant cette variable avec 4 catégories (TPE : très petite entreprise, PME : petite ou moyenne entreprise, ETI : entreprise à taille intermédiaire et GE : grande entreprise).
- **Forme juridique de l'entreprise** : Variable catégorielle nominale référant à la forme juridique de l'entreprise. Les entreprises avec un siège qui n'est pas sur le territoire français seront indiquée par une forme « Etranger ».

Nous avons effectué cette démarche de « Scrapping » avec un algorithme présenté en **Annexe J** sur plusieurs sites spécialisés tels que ZoomInfo ou societe.com.

Ajouté à cela, nous avons également pu utiliser les informations contenues dans les variables de MRR mensuelles, en effet pour ce faire nous avons choisi d'ajouter 4 variables supplémentaires :

- **Nombre de mois payés** : Variable numérique qui relate le nombre de mois où le client a payé son abonnement.
- **Nombre de Upsells** : Variable numérique qui nous donne le nombre de fois où un client a augmenté son montant d'abonnement, c'est-à-dire où son MRR a augmenté sur la période considérée.
- **Nombre de Downsells** : Variable numérique qui nous donne le nombre de fois où un client a réduit le montant de son abonnement, c'est-à-dire où son MRR a diminué sur la période considérée.
- **Moyenne revenue Client** : Variable numérique qui correspond à la moyenne des MRR sur la période considérée relativement au nombre de mois payés par le client.

L'ajout de ces variables permet de combler les pertes d'informations relatives à l'élimination de la dimension temporelle due aux différents MRR qu'on a dû retirer de notre base de données brutes.
Afin de pouvoir visualiser la structure de notre base de données avec les nouvelles variables ajoutées et des transformations effectuées, nous pouvons à nouveau visualiser les valeurs manquantes dans le graphique disponible en **Annexe K** :

Comme nous le voyons, nous disposons toujours de valeurs manquantes, et qui devront donc être soumises à une Imputation dans la partie Preprocessing puisque nous ne pouvons éliminer les occurrences qui connaissent des valeurs manquantes, au vu du nombre assez peu conséquent des enregistrements.

2.2. Analyses et visualisations de la base de données épurée

Après avoir extrait ces données nous nous sommes retrouvés avec une base de données plus pertinente, qu'on peut analyser et visualiser afin de se familiariser avec les différentes relations existantes entre les variables pour but d'émettre des hypothèses sur l'issue de nos modélisations.
Nous commençons tout d'abord par analyser les contours de notre base de données par des visualisations qui porteront sur les distributions des différentes variables, les relations entre les variables ainsi que les relations entre les variables et la cible qui est le Churn.
(Nous pouvons voir une partie de la base de données épurée en **Annexe K).**

- **Les types de variables :**

La fonction « dtypes.values_counts() » visualisée à l'aide de la fonction « plot.pie » de la bibliothèque « Matplotlib » nous permet d'obtenir le diagramme en camembert suivant des types de variables à notre disposition :

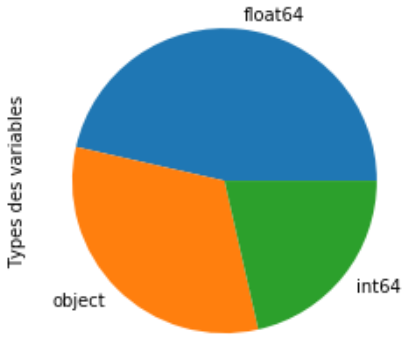


Figure 40 : Diagramme en camembert représentant la répartition des types de variables

En effet parmi les 28 variables de base dont nous disposons, 13 sont à valeurs réelles et donc de type « float », 9 de type « object » c'est-à-dire des variables catégorielles, et 6 sont des variables entières autrement dit de types « int ».

- **Distribution de la cible :**

Nous pouvons à travers la fonction « value_counts », spécifier que les 2 classes « Churn » et « Non Churn » sont déséquilibrées par la présence de nombre minoritaire des étiquettes « Churn » par rapport aux étiquettes « Non Churn ».

```
Entrée [263]: df['churn'].value_counts(normalize=True)

Out[263]: no     0.864353
          yes    0.135647
          Name: churn, dtype: float64
```

Figure 41 : *Distribution de la cible*

- **Distribution des variables :**

Afin de pouvoir mieux apprendre de nos variables pour ensuite quantifier leurs pouvoirs prédictifs. Le diagramme « distplot » de « Seaborn » nous permet de visualiser les distributions des variables numériques par des histogrammes, ce qui nous donne par exemple, pour la variable « Mois 1 » ainsi que pour la variable « Nombre de mois en activité ».

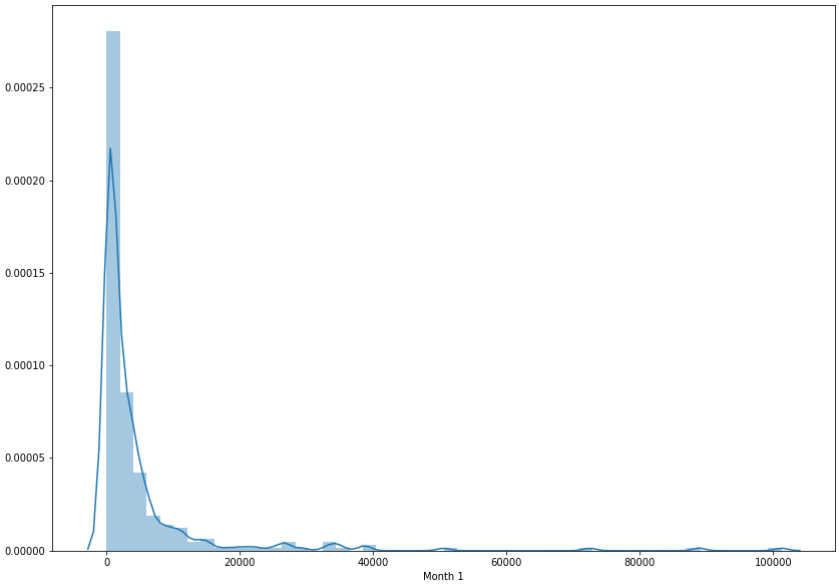


Figure 42 : Distribution de la variable « Mois 1 »

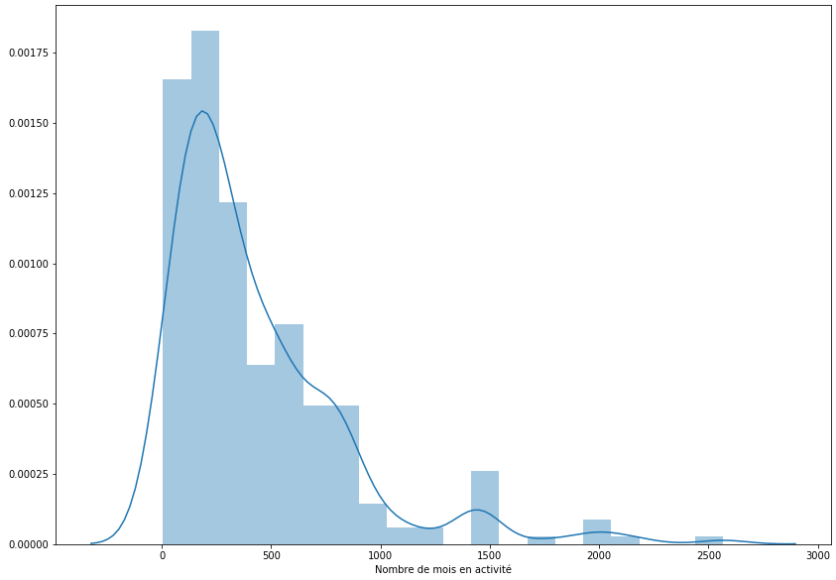


Figure 43 : Distribution de la variable « Nombre de mois en activité »

De ces 2 graphiques on peut comprendre que leurs distributions sont asymétriques ou « Skewed » ce qui nous laisse à penser que leur coefficient de « Skewness » doit être conséquemment élevé. Cette remarque s'applique également aux autres distributions des variables numériques de notre jeu de données.

On peut cependant observer une légère symétrie des distributions des deux variables « Nb de Downsells » et « Nb de Upsells », qui ressemble à une distribution normale comme on peut le visualiser sur l'histogramme suivant :

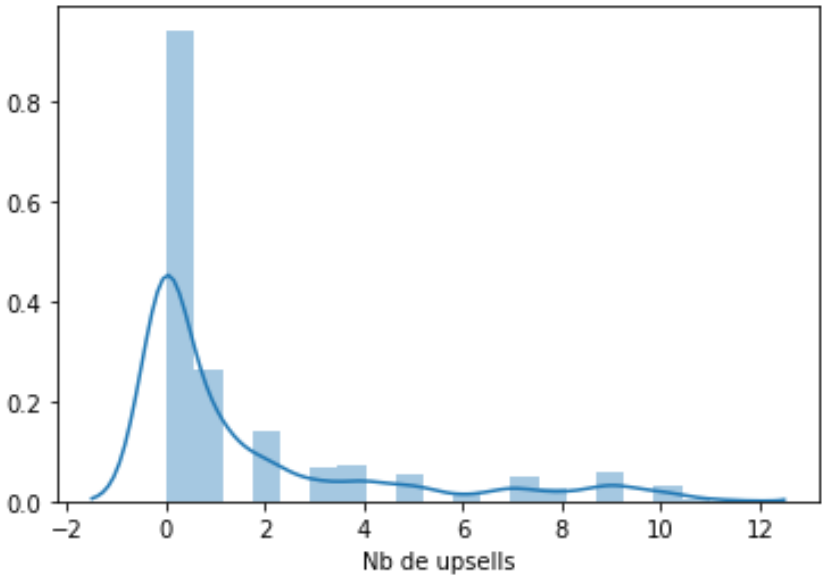


Figure 44 : Distribution de la variable « Nombre d'upsells »

Enfin la distribution de la variable « Nombre de mois payés » a une distribution proche de la distribution symétrique de Student comme affiché ci-après :

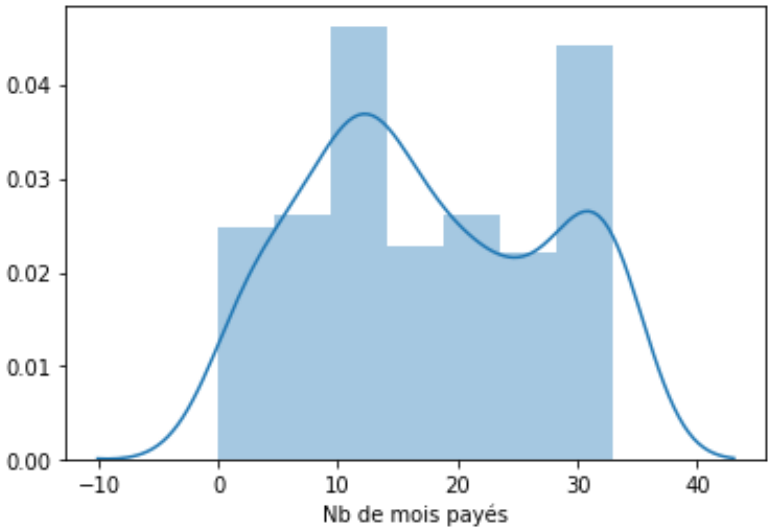


Figure 45 : Distribution de la variable « Nombre de mois payés »

Nous pouvons retrouver les visualisations des distributions des autres variables de type numérique en **Annexe L**.

Quant aux variables catégorielles, on a généré des diagrammes en camembert pour chacune d'elles grâce à la fonction « plot.pie » de « Matplotlib », qui nous serviront à distinguer les variables catégorielles avec un grand nombre de catégories pour les éliminer, ainsi que les variables catégorielles binaires ou possédant peu de modalités qu'on devra encoder. Ces diagrammes sont disponibles en **Annexe M** de ce document.

Nous clôturons cette analyse en visualisant la matrice de corrélation entre les variables numériques en calculant le coefficient de corrélation de Pearson (**Annexe N**) qui nous permet de repérer des redondances éventuelles dans l'explication de la cible. Nous avons pu visualiser cette matrice grâce à la fonction « corr » de « Pandas » appliqué à notre « DataFrame » qu'on a intégrée à une « heatmap » de « Seaborn » pour obtenir la carte suivante :

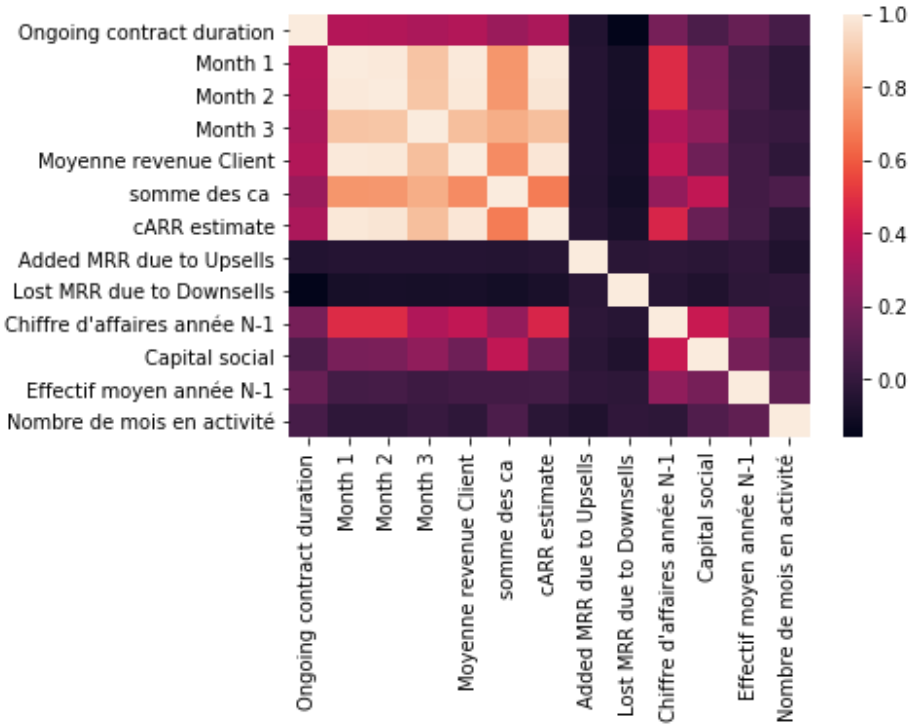


Figure 46 : Heatmap représentant les corrélations entre variables numériques

- **Relation entre les variables et la cible :** Dans ce qui suit, nous allons présenter les résultats de nos traitements en ce qui concerne les relations entre les variables et la cible c'est-à-dire nos classes « Churn » et « Non Churn ». Pour ce faire, nous avons divisé notre jeu de données 2, celui contenant les occurrences qui ont connu un « Churn » et l'autre avec les occurrences étiquetées « Non Churn ». On a ensuite, dans une boucle « for » passé l'action « distplot » de « Seaborn » pour chacun de ces 2 derniers ensembles de données.

  Pour les variables numériques continues, nous n'avons relevé qu'une seule distinction potentiellement significative entre les distributions, et qui concerne la variable « Nombre de mois en activité » comme l'explique le schéma suivant :

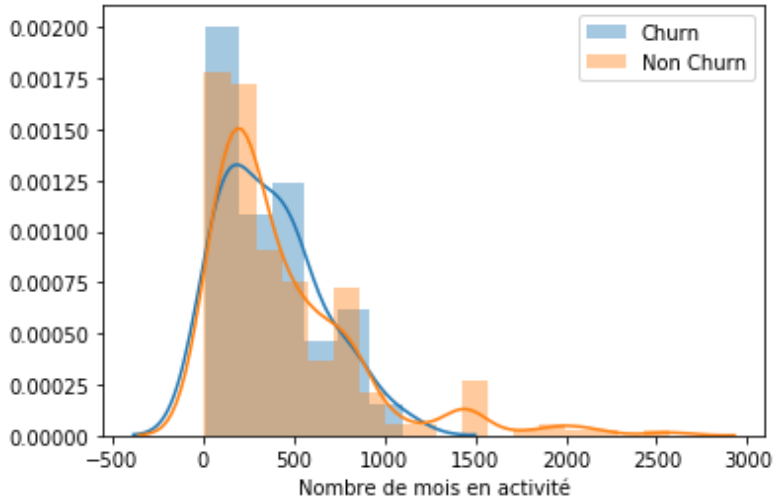


Figure 47 : Distribution de la variable « Nombre de mois en activité » selon chacune des classes

Cette hypothèse peut être vérifiée en effectuant le test de Student, aussi appelé T-test pour deux distributions indépendantes.
La mise en place de ce test est conditionnée par la normalité des distributions des variables numériques. Cette condition est remplie par la présence, dans chacune des distributions, de 318 occurrences, et donc de plus de 30 occurrences, ce qui remplit la condition du théorème central limite (**TCL**) et donc atteste de la **normalité** des échantillons considérés, qui peuvent être ainsi soumis au test de Student.
Ce dernier prend la forme suivante :

- **H0 :** Les deux distributions ont les mêmes valeurs prédites, en assumant que les populations ont des variances identiques.
- **H1 :** Les deux distributions des variables sont significativement distinctes par rapport à la variable cible.

Ce test est réalisé à l'aide d'une fonction « t-test » en utilisant la fonction « ttest_ind » de la bibliothèque « Scipy ». Nous avons implémenté une fonction, qu'on peut retrouver en **Annexe O**, qui réalise ce test sous un risque alpha de 5% et qui récupère la p-value du test pour chacune des variables considérées. Si cette dernière est inférieure au seuil alpha choisi alors H0 est rejetée. Les résultats nous révèlent que les variables « Mois 1 », « Mois 2 », « Mois3 » ainsi que « Moyenne revenue client », « Somme des CA » sont significativement distinctes par rapport au Churn alors que les autres variables numériques ne le sont pas.

Concernant les variables numériques discrètes, nous avons utilisé la fonction « Countplot » de « Seaborn » pour visualiser les relations existantes entre ces variables et la cible. Les résultats de cette visualisation concernant la variable « Users » sont ci-après :

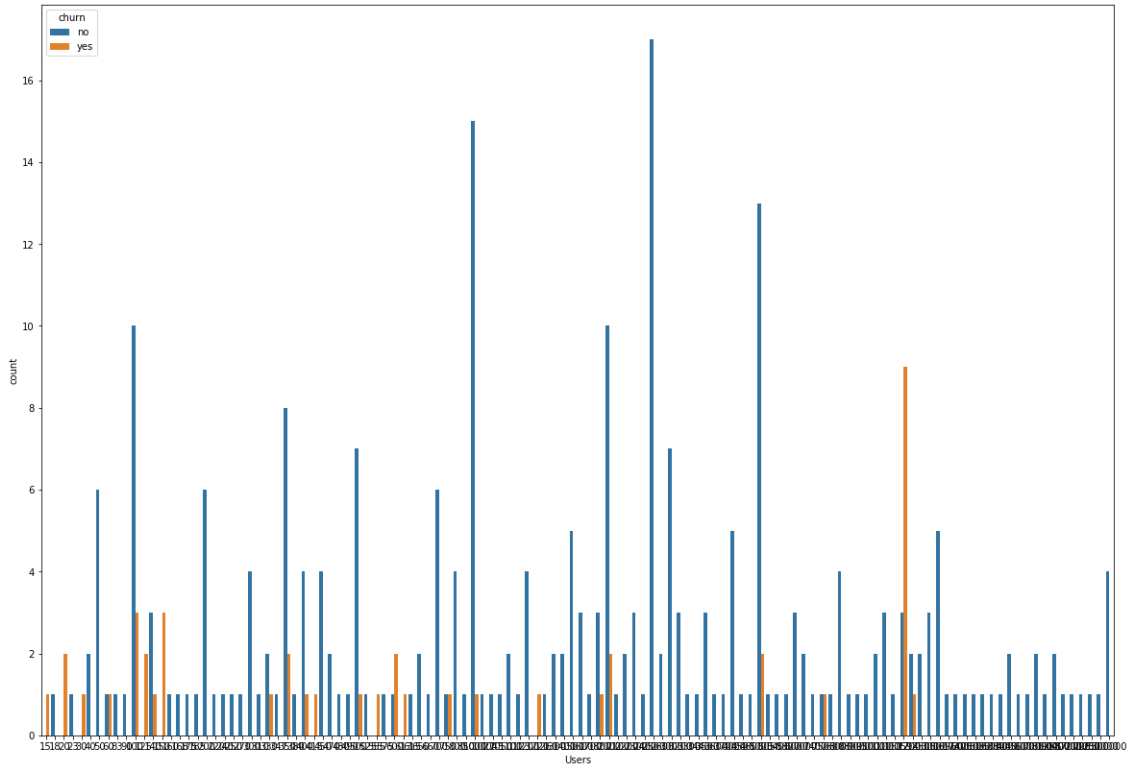


Figure 48 : Distribution de la variable « Users » selon les classes

On peut remarquer le fort nombre d'occurrences ayant un nombre d'utilisateurs élevé qui se sont désabonnés.

Nous pouvons retrouver les graphiques présentant les relations existantes entre les variables numériques et la variable cible « Etat de l'abonnement » dans l'**Annexe P**.

Enfin, pour les variables catégoriques, nous avons eu recours à la fonction « Crosstab » de « Pandas » pour obtenir les relations entre les modalités de ces variables et la variable cible. Une « heatmap » de « Seaborn » permet de visualiser nos résultats avec par exemple la variable « Secteur » dans le graphique suivant :

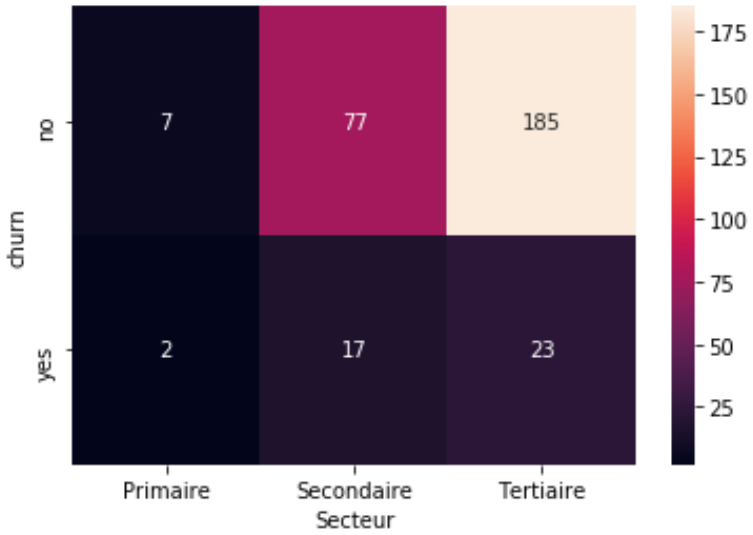


Figure 49 : *Distribution de la variable « Secteur » selon les classes*

On peut y remarquer que le taux d'entreprises étiquetées « Churn » est plus important pour les entreprises du secteur secondaire que pour les autres secteurs. Le reste des visualisations des relations entre les variables catégorielles et la variable cible sont disponibles en **Annexe Q**.

3. Préparation des données

Afin d'obtenir des performances de modélisations optimales, il convient de passer par l'étape de préparation des données, aussi connue sous le nom de pré-traitement ou de « Preprocessing ». Cette étape nous permet d'avoir un input à nos algorithmes de Machine Learning que la machine est en mesure de comprendre pour en tirer par la suite les règles mathématiques liant les variables à la cible selon l'algorithme implémenté. Ce format d'input doit notamment être numérique ou booléen, suite à un encodage des variables catégorielles. Nous déroulerons par conséquent les étapes de prétraitement suivantes :

3.1. Suppression de variables

Notre base de données comporte des variables catégorielles ayant un nombre de modalités trop élevées pour être transformées en variables numériques, ce qui est le cas des variables « Revendeur » ainsi que « Customer ID », et ces deux variables sont donc éliminées avec la fonction « drop » utilisée précédemment.

Au vu du faible nombre d'occurrences à notre disposition, nous n'effectuerons pas de suppression de variables supplémentaires, notamment celles qui possèdent des valeurs manquantes, nous choisissons d'effectuer une imputation de ces dernières, une procédure expliquée dans les prochaines étapes.

3.2. Création des ensembles d'entrainement et de test

Afin de pouvoir tester nos modélisations, il convient tout d'abord de créer le jeu d'entrainement, qui servira à entrainer nos modèles afin qu'ils obtiennent la meilleure performance possible, puis d'appliquer ce modèle sur un jeu que le modèle ne connaît pas encore afin de juger de sa performance sur des données qu'il n'a pas encore eu l'occasion de découvrir. Cette procédure est commune à tous les modèles de Machine Learning et nous permettra de juger de l'écart existant entre chaque modèle lorsqu'il sera amené à prédire à partir de données encore inconnues pour lui. Notre projet s'inscrit dans cette démarche puisqu'à partir de bases de données, on devra prédire quelles seront les occurrences qui se désabonneront à un horizon donné, et quelles sont celles qui ne le feront pas.

Nous effectuons la création de ces jeux en utilisant la fonction de « Pandas » appelée « train_test_split » prenant en paramètre l'ensemble de nos données de la base de données épurée, ainsi qu'une taille « test_size » pour le jeu de test qu'on a choisie étant égale à 20% pour avoir dans l'ensemble de test un nombre d'occurrences suffisant pour évaluer la performance de nos modèles, et un paramètre appelé « random_state » qui fige les données sélectionnées dans chaque ensemble, et qui est un entier naturel qu'on a choisi aléatoirement comme prenant une valeur de 2.

Le résultat de cette fonction est de retourner aléatoirement 2 sous-ensembles, l'un qui contiendra les observations qui subiront un entraînement à travers les différents modèles, et l'autre sur lequel

Nous obtenons les résultats suivants avec la fonction « value_counts » appliquée sur chaque ensemble :

```
trainset['churn'].value_counts()
```

```
no     221
yes     32
Name: churn, dtype: int64
```

```
testset['churn'].value_counts()
```

```
no     53
yes    11
Name: churn, dtype: int64
```

Figure 50 : Visualisation des dimensions des sous-ensembles d'entraînement et de test

Nous choisirons par la suite les 2 sous-ensembles de chacun des 2 ensembles créés, qui sont les sous-ensembles « X_train » et « y_train » ainsi que « X_test » et « y_test ». Les sous-ensembles « X_train » et « X_test » contiendront les données et leurs variables explicatives de notre jeu de données alors que « y_train » et « y_test » contiendront eux la variable expliquée les étiquettes qui nous réfèrent à quels groupes appartiennent les occurrences, qu'elles « Churn » et « Non Churn ». Nous effectuons cette subdivision des ensembles d'entraînement et de test en utilisant la méthode de sélection de colonne ainsi que la fonction « drop » toutes deux rendues possibles grâce à la bibliothèque « Pandas ».

3.3. Imputation des valeurs manquantes

Le traitement des valeurs manquantes est effectué selon 3 procédés distincts, la suppression des variables avec un nombre majoritaire de valeurs manquantes, la suppression des occurrences qui contiennent des valeurs manquantes ou alors l'imputation ou le remplacement des valeurs manquantes par une stratégie d'imputation adaptée à la situation et basée sur des règles mathématiques.

Au vu du nombre assez peu conséquent à notre portée, ainsi que l'absence de variables possédant majoritairement des valeurs manquantes, nous avons choisi d'appliquer différentes stratégies d'imputation selon le type de variable considéré ainsi que sa distribution.

Nous avons donc implémenté une fonction appelée « imputation », prenant en paramètres les ensembles d'entraînement et de test « X_train » et « X_test ». Cette fonction est présentée en **Annexe R** avec la façon dont elle opère sur les données.

3.4. Encodage des variables catégorielles

La partie suivante consiste en une mission primordiale dans le prétraitement des données et qui est l'encodage des variables catégorielles. Ces dernières étant sous format de « chaine de caractères » doivent être transformées en valeurs numériques que la machine peut comprendre. Cette opération a été réalisée en implémentant la fonction présentée en **Annexe S** appelée « encodage » qui fait suite à la fonction « imputation ».

Nous appliquons donc ces fonctions successivement pour obtenir des ensembles de données qui seront aptes à subir une modélisation par un entrainement de par les algorithmes de l'apprentissage supervisé. Une autre partie de prétraitement a lieu en marge de la modélisation, appelée standardisation, elle permet d'éliminer les « outliers » en plaçant toutes les occurrences des variables numériques sur une même échelle. Cette démarche sera expliquée dans la partie suivante.

4. Modélisation des données

Après avoir réalisé toutes les étapes de la préparation, nous passons à la partie de modélisation où plusieurs modèles d'apprentissage seront créés pour modéliser la capacité de nos données à prédire le Churn.

Nous mettrons en œuvre l'implémentation de plusieurs modèles, définis dans la partie état de l'art, car n'étant pas certains de ceux qui donneront les meilleurs résultats, et donc qui expliquerons le mieux la variable cible, nous effectuerons une démarche visant à choisir le meilleur modèle parmi tous les modèles essayés.

Nous entamerons ce travail en initialisant une procédure d'évaluation. Cette dernière prendra la forme d'une fonction que nous allons implémenter et qui sera appliquée sur chacun des modèles qui seront développés par la suite.

```
def evaluation(model):
    model.fit(X_train,y_train)
    y_pred = model.predict(X_test)
    print(confusion_matrix(y_test,y_pred))
    print(classification_report(y_test,y_pred))
    N, train_score, val_score = learning_curve(model,X_train,y_train,cv=4,scoring="f1", train_sizes =np.linspace(0.1,1,10))
    plt.figure(figsize=(12,8))
    plt.plot(N, train_score.mean(axis=1), label='training score')
    plt.plot(N, val_score.mean(axis=1),label='validation score')
    plt.legend()
```

Figure 51 : Algorithme de la procédure d'évaluation des modèles

Cette fonction prend en paramètre un modèle implémenté, qui sera entrainé sur les données d'entrainement avec la méthode « fit », puis qui sera utilisée pour effectuer des prédictions avec l'ensemble de test avec la méthode « predict ». Une fois les prédictions effectuées, nous visualisons tout d'abord la matrice de confusion des résultats en comparant les résultats de la prédiction « y_pred » avec l'ensemble de test de la cible « y_test » où la classe « Churn » prend la valeur 1 et la classe « Non Churn » la valeur 0.

Enfin, afin de visualiser les performances du modèle choisi, nous imprimons sur l'écran sa courbe d'apprentissage avec une métrique choisi étant le « F1 score » en fonction du nombre d'occurrences en abscisse. Nous implémentons une procédure de Cross-Validation, dans notre cas avec 4 sous-ensembles. Cette dernière est expliquée dans **l'Annexe T** et nous permet d'avoir un modèle plus robuste car ce dernier validera son entrainement d'abord sur un sous-ensemble de l'ensemble d'entraînement appelé ensemble de validation ou « Validation set ». Les deux scores étant affichés seront ceux de l'entrainement ainsi que de la validation.

Cette procédure nous permettra de cerner la performance de nos modèles et de juger 2 caractéristiques primordiales qui nous serviront dans la partie optimisation :

- **La performance du modèle :** nous évaluerons la performance du modèle relativement à sa courbe d'évaluation du « F1 score ».
- **La capacité du modèle à faire du surapprentissage :** Le surapprentissage ou « overfitting » est le phénomène qui est observé lorsque l'entraînement est parfaitement réalisé, avec une courbe d'entrainement étant une constante égale à 1, mais que la validation connait des résultats moindres.

Lorsqu'il y a « overfitting » ou une performance assez peu conséquente, nous soumettrons le modèle à une optimisation de ses hyperparamètres, une procédure qui sera expliquée dans la suite de ce chapitre. Nous allons dans ce qui suit présenter les résultats de nos différents modèles générés.

Nous allons implémenter 2 stratégies de modélisation, la première consiste à travailler avec notre base de données épurée, et nettoyée par la phase de prétraitement et lui appliquer des modèles de classification basiques puis des modèles de classification ensemblistes. En marge de la seconde stratégie, nous appliquerons un suréchantillonnage afin de rééquilibrées nos données et ainsi avoir un nombre égal entre les entreprises ayant connu un « Churn » et les entreprises ayant connu un « Non Churn ».

Les différents scripts qui nous ont permis d'implémenter ces modèles sur les bases des algorithmes définis dans les bibliothèques de « Scikit learn » sont disponibles **annexe U**, et les résultats sont à retrouver dans ce qui suit.

Les modèles implémentés ont subi une procédure d'optimisation de leurs hyperparamètres par des méthodes du type « GridSearchCV » et « Bayesian Optimization », ces notions sont à retrouver en **Annexe V**.

4.1. Première stratégie de modélisation

Les algorithmes utilisés pour cette première stratégie sont les suivants :

4.1.1. Arbres de décision

Le premier modèle entrainé est effectué en implémentant l'algorithme des arbres de décision. Ce classificateur, appelé « DecisionTreeClassifier » implémenté nous donne les résultats suivants en étant évalué avec la procédure d'évaluation présentée précédemment :

```
[[50  3]
 [ 7  4]]
              precision    recall  f1-score   support

           0       0.88      0.94      0.91        53
           1       0.57      0.36      0.44        11

    accuracy                           0.84        64
   macro avg       0.72      0.65      0.68        64
weighted avg       0.82      0.84      0.83        64
```

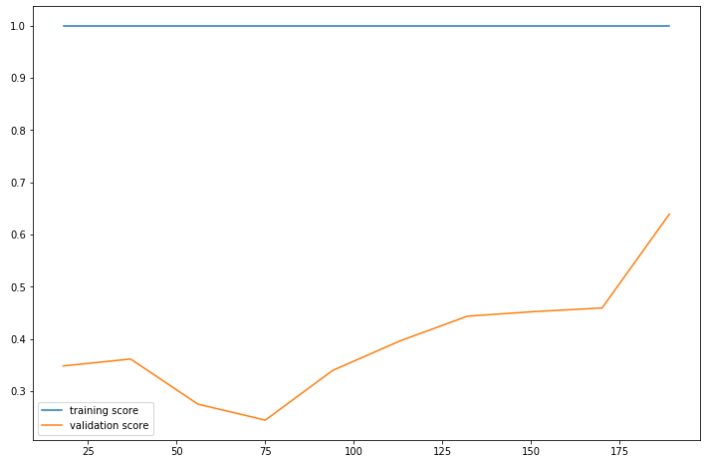


Figure 52 : Rapport de classification et courbe d'apprentissage du modèle « Arbres de décision » avant optimisation

On s'aperçoit que nous avons 4 occurrences bien prédites faisant partie de la classe « Churn » et 50 prédites correctement pour la classe « Non Churn », ce qui nous donne un « F1 score » moyen de 68%.

Ces résultats pouvant être améliorés, nous avons opéré une optimisation des hyperparamètres de ce modèle en en retenant 3 principaux qui sont :

- **Max_depth :** paramètre caractérisant la profondeur maximale d'un arbre de décision.
- **Min_samples_split :** le nombre minimal d'échantillons requis pour diviser un nœud interne.
- **Min_samples_leaf :** nombre minimal d'échantillons requis pour être le nœud feuille.

Pour trouver la combinaison optimale de ces paramètres nous avons implémenté un processus de « Grid Search » avec « cross-validation » appelé « GridSearchCV ». Ce dernier n'a pas amélioré de manière conséquente nos résultats, car si le modèle n'est plus en « overfitting », il perd quand même en recall dans la classification et donc en F1 score :

```
[[53  0]
 [ 9  2]]
              precision    recall  f1-score   support

           0       0.85      1.00      0.92        53
           1       1.00      0.18      0.31        11

    accuracy                           0.86        64
   macro avg       0.93      0.59      0.61        64
weighted avg       0.88      0.86      0.82        64
```

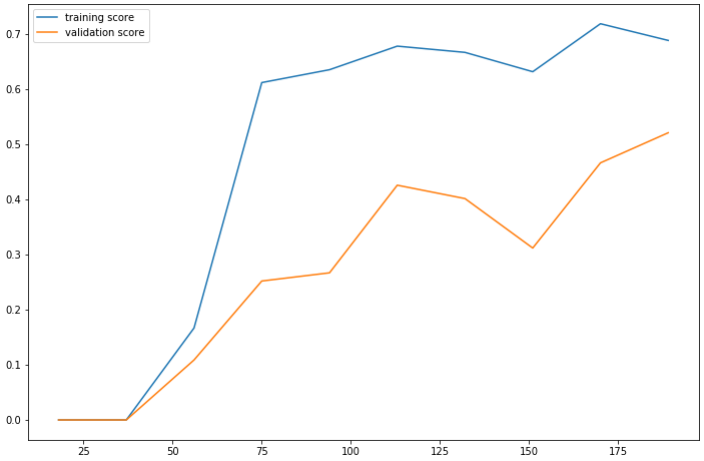


Figure 53 : Rapport de classification et courbe d'apprentissage du modèle « Arbres de décision » après optimisation

4.1.2. K-Nearest Neighbors

Nous avons par la suite utilisé le modèle de KNN en implémentant cet algorithme à l'aide du classifieur « KNeighborsClassifier », qui nous a donné les résultats suivant après une optimisation de l'hyperparamètre principal qui est K : nombre de voisins à l'aide de « GridSearchCV » avec une métrique de calcul de distance qui est la distance Euclidienne, ce qui nous a donné les résultats suivants :

```
[[51  2]
 [10  1]]
              precision    recall  f1-score   support

           0       0.84      0.96      0.89        53
           1       0.33      0.09      0.14        11

    accuracy                           0.81        64
   macro avg       0.58      0.53      0.52        64
weighted avg       0.75      0.81      0.77        64
```

Figure 54 : Rapport de classification « KNN »

On peut s'apercevoir que les résultats de ce modèle sont assez médiocres car bien que les occurrences « Non Churn » aient été bien prédites, ace qui a débouché sur un F1 score de 0.52.

La courbe d'apprentissage du modèle K-NN est à retrouver en **Annexe Y.**

4.1.3. Support Vector Machine

Nous avons effectué pour cet algorithme spécifique 3 types de modélisation, avec des fonctions noyaux appelés « kernels » différents. En effet après avoir testé le premier noyau appelé « Radial Basis Function » ou « rbf », ce dernier nous a donné des performances assez médiocres comme l'indique le rapport de classification suivant :

```
[[39 14]
 [ 9  2]]
              precision    recall  f1-score   support

           0       0.81      0.74      0.77        53
           1       0.12      0.18      0.15        11

    accuracy                           0.64        64
   macro avg       0.47      0.46      0.46        64
weighted avg       0.69      0.64      0.67        64
```

Figure 55 : Rapport de classification du modèle « Support Vector Machine » avec noyau « rbf »

Nous sommes donc passés au noyau « linéaire » dont les résultats sont ci-après, et qui nous montrent un léger accroissement de prédiction de la classe « Churn » :

```
[[43 10]
 [ 9  2]]
              precision    recall  f1-score   support

           0       0.83      0.81      0.82        53
           1       0.17      0.18      0.17        11

    accuracy                           0.70        64
   macro avg       0.50      0.50      0.50        64
weighted avg       0.71      0.70      0.71        64
```

Figure 56 : Rapport de classification du modèle « Support Vector Machine » avec noyau « linéaire »

Enfin, nous avons également construit un modèle basé sur la fonction noyau « sigmoid » avec les résultats suivants, où nous remarquons que la classe minoritaire a été assez bien prédite mais pas la classe majoritaire :

```
[[15 38]
 [ 5  6]]
              precision    recall  f1-score   support

           0       0.75      0.28      0.41        53
           1       0.14      0.55      0.22        11

    accuracy                           0.33        64
   macro avg       0.44      0.41      0.31        64
weighted avg       0.64      0.33      0.38        64
```

Figure 57 : Rapport de classification du modèle « Support Vector Machine » avec noyau « sigmoid »

Pour ces 3 algorithmes nous avons optimisé les hyperparamètres suivants :

- **C** : appelée valeur de régularisation, elle permet de contrôler la maximisation de l'hyperplan et doit être adéquate entre une valeur trop grande qui mènerait à un sur-apprentissage et une valeur trop faible qui affaiblirait les performances.
- **Gamma** : coefficient permettant d'adapter l'hyperplan aux données.

Les courbes d'apprentissage des différents modèles de SVM sont à retrouver en **Annexe Y**

4.1.4. Réseau de neurones artificiels

Nous avons par la suite implémenté un réseau de neurones avec plusieurs couches ou « Multi Layer Perceptron » aussi appelé MLP. Nous avons d'abord initialisé un classifieur « MLPClassifier » dont nous avons par la suite optimisé les hyperparamètres suivants grâce au « GridSearchCV » :

- **Hidden layer sizes :** Représente le nombre neurones présents à chaque couche input, output et cachée de notre Perceptron.
- **Activation :** Fonction d'activation d'une couche intermédiaire (ou cachée) à choisir entre « identité », « logistique », « tanh » et « relu ».
- **Solveur :** Le solveur utilisé pour l'optimisation à choisir entre :
  - **Lbfgs :** solveur d'optimisation de la famille des méthodes quasi-Newton.
  - **SGD :** réfère au stochastic gradient descent, une méthode d'optimisation alternative aux moindres carrés (MCO).
  - **Adam :** autre méthode basée sur le SGD.
- **Alpha :** terme de régularisation
- **Learning_rate :** utilisé seulement lorsque le solveur est de type SGD, il permet de contrôler la rapidité d'accès à la solution optimale.

Les résultats de cette démarche d'optimisation nous ont donné les résultats suivants à l'évaluation de ce modèle :

```
[[10 43]
 [ 3  8]]
              precision    recall  f1-score   support

           0       0.77      0.19      0.30        53
           1       0.16      0.73      0.26        11

    accuracy                           0.28        64
   macro avg       0.46      0.46      0.28        64
weighted avg       0.66      0.28      0.30        64
```

Figure 58 : Rapport de classification du modèle « Perceptron »

Nous remarquons que les résultats sont assez peu conventionnels, puisque la classe « Non Churn » est prédite de manière très correcte avec un Recall de 0.73 alors que la classe « Churn » connait des difficultés à être prédite correctement et accuse un Recall de 0.19 ce qui donne un F1 score de 0.28, ce qui est plus faible que les modèles précédents. (La courbe d'apprentissage du modèle Perceptron est à retrouver en **Annexe Y**).

4.1.5. Régression logistique

Les résultats de la régression logistique implémentée avec le classifieur « LogisticRegression » nous a fourni les résultats suivants :

```
[[45  8]
 [ 7  4]]
              precision    recall  f1-score   support

           0       0.87      0.85      0.86        53
           1       0.33      0.36      0.35        11

    accuracy                           0.77        64
   macro avg       0.60      0.61      0.60        64
weighted avg       0.77      0.77      0.77        64
```

Figure 59 : Rapport de classification du modèle « Régression Logistique »

Nous remarquons un F1 score de 60% pour un modèle qui a assez bien prédit les occurrences « Non Churn » mais qui connait des difficultés à prédire la classe « Churn ». Ce modèle a vu les hyperparamètres suivants être optimisés :

- **C :** valeur de régularisation semblable à celle des SVM.
- **Class_weight :** permet de spécifier le poids sur une échelle de 1 des différentes classes à prédire, dans notre cas il y en a deux.
- **Penalty :** ce paramètre permet de contrôler les valeurs extrêmes des autres paramètres dans la régularisation afin d'éviter le surapprentissage.
- **Solveur :** nous avons choisi le solveur « Liblinear » adapté aux datasets assez réduits.

La courbe d'apprentissage du modèle Régression logistique est à retrouver en **Annexe Y.**

4.1.6. Forets aléatoires

Après avoir testé les modèles basés sur les algorithmes de classifieurs dits « faibles », nous nous sommes attaqués aux algorithmes de l'apprentissage ensembliste dit « ensemble learning » en commençant par les forêts aléatoires avec le classifieur « RandomForestClassifier » optimisé avec une « GridSearchCV » pour les hyperparamètres cités précédemment pour le « DecisionTreeClassifier » en plus du nombre d'estimateurs « n_estimators » qui correspond au nombre d'arbres dans la forêt. Nous obtenons les résultats suivants :

```
[[53  0]
 [ 7  4]]
              precision    recall  f1-score   support

           0       0.88      1.00      0.94        53
           1       1.00      0.36      0.53        11

    accuracy                           0.89        64
   macro avg       0.94      0.68      0.74        64
weighted avg       0.90      0.89      0.87        64
```

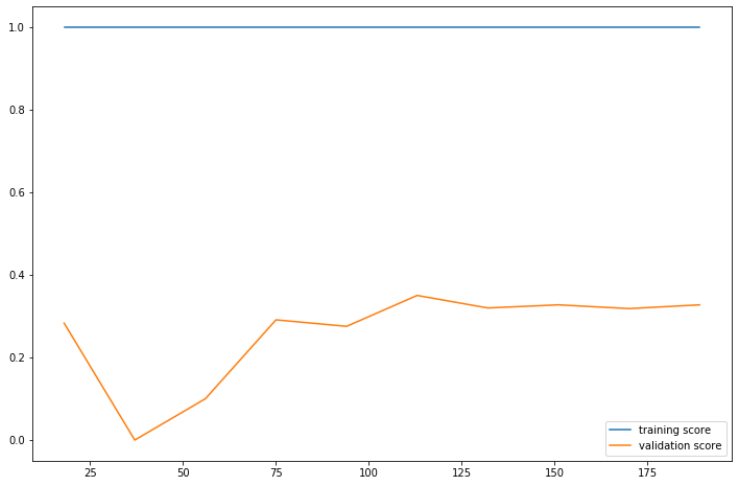


Figure 60 : Rapport de classification et courbe d'apprentissage du modèle «Forets aléatoires»

Nous remarquons que le modèle, bien qu'il possède un F1 score de 74%, est cependant en situation d'overfitting, et ce, malgré l'optimisation de ses hyperparamètres avec « GridSearchCV ».

4.1.7. Gradient Boosting

Ayant des modèles jusque-là assez peu performants ou alors en surapprentissage, nous avons implémenté un modèle basé sur le « GradientBoostingClassifier » qui est autre algorithme d'apprentissage ensembliste basé sur les arbres de décisions. Ce dernier a été soumis à une démarche de « GridSearchCV » pour les paramètres : « learning_rate », « max_depth » ainsi que « n_estimators ». Nous avons obtenu les meilleurs hyperparamètres et enclenché une procédure de validation qui nous a donné les résultats suivants :

```
[[52  1]
 [ 2  9]]
              precision    recall  f1-score   support

           0       0.96      0.98      0.97        53
           1       0.90      0.82      0.86        11

    accuracy                           0.95        64
   macro avg       0.93      0.90      0.91        64
weighted avg       0.95      0.95      0.95        64
```

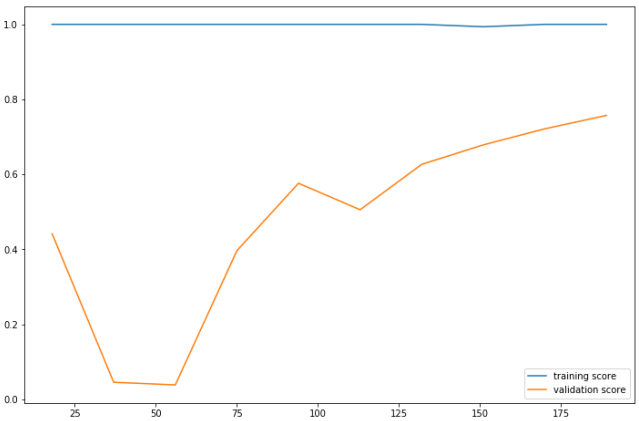


Figure 61 : Rapport de classification et courbe d'apprentissage du modèle « Gradient Boosting »

Nous pouvons remarquer que les résultats sont assez satisfaisants avec un F1 score de 91% ainsi qu'une courbe d'apprentissage qui nous révèle que le modèle n'est pas en overfitting.

4.1.8. Extreme Gradient Boosting

L'algorithme de XGBoost a également été entraîné en marge d'un modèle que nous avons mis en place. Cet algorithme basé sur les arbres de décision a été implémenté avec « XGBClassifier » de la bibliothèque « xgboost » et a été par la suite optimisé avec une méthode d'optimisation Bayesienne, reconnue comme étant beaucoup moins coûteuse en termes de complexité algorithmique et qui prend donc moins de temps à être implémentée en comparant avec « GridSearchCV » ainsi que « RandomizedSearchCV ».

Les hyperparamètres retenus pour cet algorithme furent le « learning_rate », le « max_depth », le « n_estimators » ainsi que « gamma » qui est un paramètre dit de régularisation, généralement entre 0 et 5 et qui évite le surapprentissage en étant bien optimisé. Notre démarche a abouti sur les résultats d'évaluation suivants :

```
[[53  0]
 [ 5  6]]
              precision    recall  f1-score   support

           0       0.91      1.00      0.95        53
           1       1.00      0.55      0.71        11

    accuracy                           0.92        64
   macro avg       0.96      0.77      0.83        64
weighted avg       0.93      0.92      0.91        64
```

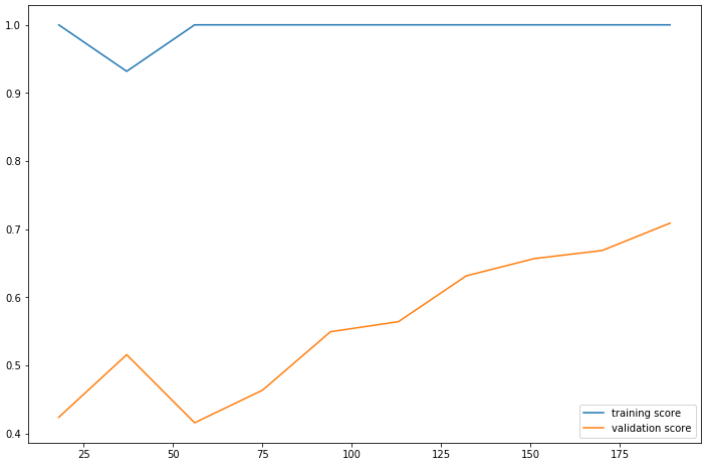


Figure 62 : Matrice de confusion et courbe d'apprentissage du modèle « Extreme Gradient Boosting »

Nous remarquons que le modèle nous donné des prévisions très correctes avec un F1 score de 83% et qui n'est pas en situation de sur-apprentissage.

4.1.9. Adaptative Boosting

Dernier modèle testé et basé sur le « AdaBoostClassifier » basé lui aussi sur les arbres de décision. Nous avons également suivi une démarche d'optimisation des hyperparamètres que sont le « n_estimators » ainsi que le « learning_rate ». Les résultats sont donnés ci-après, avec une courbe d'apprentissage qui nous apprend que le modèle est en sur-apprentissage mais qui donne de très bons résultats concernant le F1 score avec 94% :

```
[[53  0]
 [ 2  9]]
              precision    recall  f1-score   support

           0       0.96      1.00      0.98        53
           1       1.00      0.82      0.90        11

    accuracy                           0.97        64
   macro avg       0.98      0.91      0.94        64
weighted avg       0.97      0.97      0.97        64
```

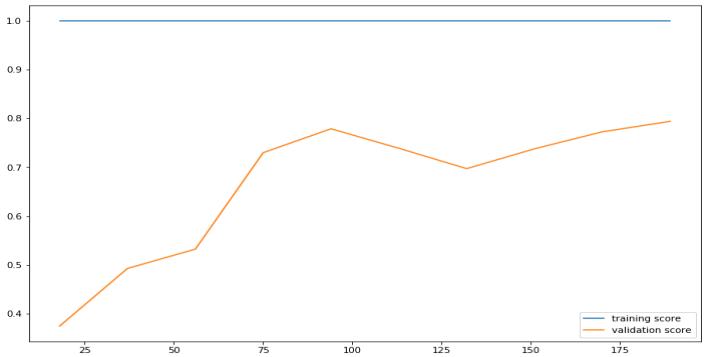


Figure 63 : Rapport de classification et courbe d'apprentissage du modèle « Adaptative Boosting »

Beaucoup d'algorithmes testés ont été réalisés avec une « Pipeline », qui crée un processus de prédiction en intégrant une partie de prétraitement désirée. Dans notre cas nous avons intégré à cette « Pipeline », pour les modèles n'étant pas basés sur les arbres que sont le SVM, KNN, Régression Logistique ainsi que Perceptron, une standardisation par la fonction « StandardScaler » pour le SVM et « RobustScaler » pour KNN, la régression logistique et Perceptron qui permettent de normaliser les distributions des colonnes relatives aux variables numériques pour éliminer les valeurs aberrantes ou « outliers ». L'utilisation d'une « Pipeline » permet de faire la distinction entre les modèles nécessitant un prétraitement supplémentaire, comme par exemple le « Feature Engineering » avec des méthodes comme « Polynomial Features » que nous avons utilisé et qui n'a pas donné de bons résultats ou alors la sélection de variables avec « SelectKBest » en se basant sur les tests statistiques (test de Fischer et du Chi-2) qui n'a pas non plus donné de bons résultats dans notre projet.

4.2. Seconde stratégie de modélisation

Au vu des résultats prometteurs des algorithmes testés précédemment, nous avons tenté une nouvelle approche sur notre base de données.

4.2.1. Sur-échantillonnage

En effet, ayant un ensemble de données déséquilibrés, nous avons tenté de comparer la performance des algorithmes précédents en tenant compte du F1 score, une métrique qui tient compte de ce déséquilibre. Cependant, une autre alternative s'offre à nous, qui est de rééquilibrer les occurrences de nos deux classes « Churn » et « Non Churn » dont la dernière est majoritaire devant la première.

Pour ce faire, deux techniques existent, qui sont le sous-échantillonnage qui consiste en l'élimination des occurrences en plus de la classe majoritaire par rapport à la classe minoritaire. Ayant un ensemble de données avec relativement peu d'occurrences, nous avons choisi d'opter pour la seconde option qui est le sur-échantillonnage qui vise à compléter le dataset original par des observations synthétiques de la classe minoritaire.

Nous avons opté pour la technique du « Synthetic Minority Over-Sampling Technique » connue sous le nom de SMOTE.

Pour cela, nous avons instancié un objet « oversample » de la classe « SMOTE » importée de la bibliothèque « imblearn » dont nous avons appelé la méthode « fit_resample » sur nos ensembles de données.

Nous avons par la suite testé 5 algorithmes sur ce nouvel ensemble de données, qui sont le « DecisionTreeClassifier », le « RandomForestClassifier », le « GradientBoostingClassifier » et le « AdaBoostClassifier » tous optimisés à l'aide d'un « GridSearchCV » ainsi que l'algorithme « XGBClassifier » optimisé par une « GridSearchCV » cette fois, portant sur les mêmes hyperparamètres choisis précédemment. Nous pouvons retrouver en **Annexe U** les différents scripts qui nous ont permis d'implémenter ces modèles sur les bases des algorithmes définis dans les bibliothèques de « Scikit learn ».

Les performances de ces modèles peuvent être jugées par les matrices de confusion et les rapports de classification suivants :

```
[[57  0]
 [ 2 51]]
              precision    recall  f1-score   support

           0       0.97      1.00      0.98        57
           1       1.00      0.96      0.98        53

    accuracy                           0.98       110
   macro avg       0.98      0.98      0.98       110
weighted avg       0.98      0.98      0.98       110
```

Figure 64 : Rapport de classification du modèle «Decision Tree»

```
[[57  0]
 [ 4 49]]
              precision    recall  f1-score   support

           0       0.93      1.00      0.97        57
           1       1.00      0.92      0.96        53

    accuracy                           0.96       110
   macro avg       0.97      0.96      0.96       110
weighted avg       0.97      0.96      0.96       110
```

Figure 65 : Rapport de classification du modèle «RandomForest»

```
[[57  0]
 [ 3 50]]
              precision    recall  f1-score   support

           0       0.95      1.00      0.97        57
           1       1.00      0.94      0.97        53

    accuracy                           0.97       110
   macro avg       0.97      0.97      0.97       110
weighted avg       0.97      0.97      0.97       110
```

Figure 66 : Rapport de classification du modèle «XGBoost»

```
[[57  0]
 [ 2 51]]
              precision    recall  f1-score   support

           0       0.97      1.00      0.98        57
           1       1.00      0.96      0.98        53

    accuracy                           0.98       110
   macro avg       0.98      0.98      0.98       110
weighted avg       0.98      0.98      0.98       110
```

Figure 67 : Rapport de classification du modèle «GBoost»

```
[[57  0]
 [ 1 52]]
              precision    recall  f1-score   support

           0       0.98      1.00      0.99        57
           1       1.00      0.98      0.99        53

    accuracy                           0.99       110
   macro avg       0.99      0.99      0.99       110
weighted avg       0.99      0.99      0.99       110
```

Figure 68 : Rapport de classification du modèle «AdaBoost»

Les courbes d'apprentissage de ces différents modèles sont à retrouver en **Annexe W** de ce document et nous apprennent que le « DecisionTreeClassifier » n'est pas en surapprentissage contrairement aux deux autres modèles, mais que le F1-score de AdaBoost est très largement devant les autres, puisque très proche des 100%.

5. Évaluation des modèles

Les modèles que nous avons construits vont être comparés afin d'en tirer les enseignements qui nous serviront à choisir les meilleurs d'entre eux.

Nous avons choisi de ne retenir que trois indices de performance qui sont le « Recall », le « F1 score » ainsi que le « AUC ». Nous pouvons dans ce qui suit visualiser les courbes de ROC ainsi que les scores AUC respectifs à chacun des modèles que nous avons testés avant d'effectuer le « sur échantillonnage » :

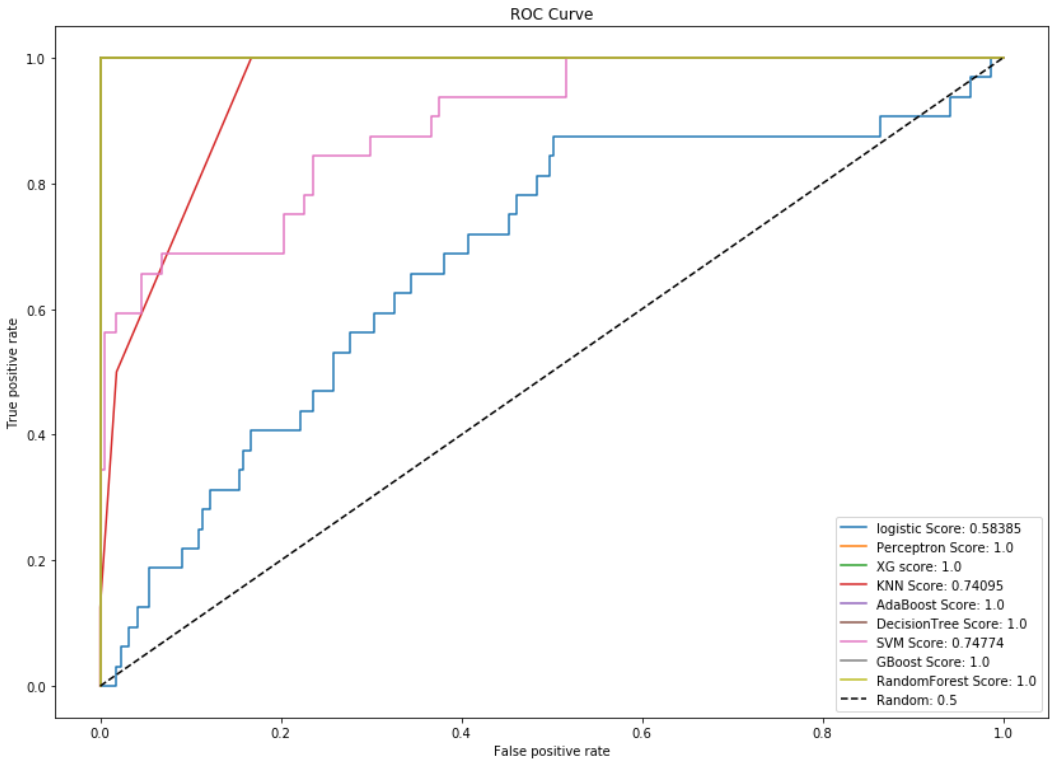


Figure 69 : Courbe ROC des modèles de la stratégie 1

Nous pouvons par conséquent comparer les performances de ces modèles sur un tableau résumant leurs différents indices de performance, afin de choisir le meilleur modèle entraîné :

Tableau 5 : Métriques des modèles de la stratégie 1

Modèle	**Recall**	**F1 score**	**AUC**
Arbres de décision	0.59	0.61	1.0
K-nearest neighbors	0.53	0.52	0.741
Support Vector Machine à noyau linéaire	0.5	0.5	0.748
Réseau de neurones artificiels	0.46	0.28	1.0
Régression Logistique	0.61	0.60	0.584
Forêts aléatoires	0.68	0.74	1.0
Gradient Boosting	0.90	0.91	1.0
Extreme Gradient Boosting	0.77	0.83	1.0
Adaptative Boosting	0.91	0.94	1.0

Nous en déduisant que les modèles basés sur le « Boosting » nous apportent des résultats très convaincants en termes de performance.

Afin de procéder à la sélection du modèle le plus apte à prédire nos classes, nous avons choisi d'établir un scoring relatif à chacun des modèles de « Boosting », sur une échelle de 1 à 10 avec une pondération pour les indices de performances suivante : que nous présentons dans le tableau suivant :

Tableau 6 : Evaluation des modèles de stratégie 1 sur une échelle de 10

Modèle	Recall	F1 score	AUC	Score global
Forêts aléatoires	6	7	10	7.6
Gradient Boosting	9	9	10	9.3
Extreme Gradient Boosting	8	8	10	8.6
Adaptative Boosting	9	9	10	9.3

Les algorithmes de Gradient Boosting et de l'Adaptative Boosting ayant des scores égaux, nous choisissons cependant le modèle de **Gradient Boosting** pour nos prédictions car il n'est pas sujet à un surapprentissage.

Quant aux modèles construits à partir de la base de données ayant connu un « suréchantillonnage », nous pouvons visualiser les courbes ROC et les scores AUC respectifs sur le graphique suivant :

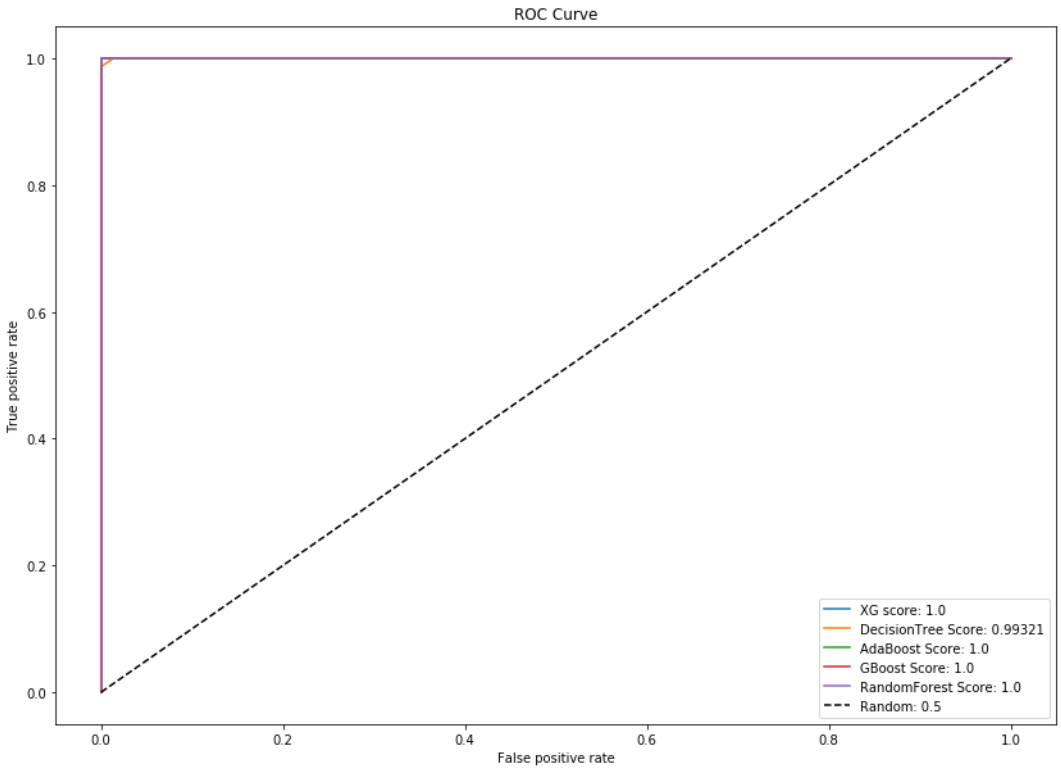


Figure 70 : Métriques des modèles de la stratégie 2

Nous pouvons par conséquent comparer les performances de ces modèles sur un tableau résumant les différents indices de performance de ces modèles, afin de choisir le meilleur modèle entraîné :

Tableau 7 : Evaluation des modèles de stratégie 2 sur une échelle de 10

Modèle	Recall	F1 score	AUC
Arbres de décision	0.98	0.98	0.993
Forêts aléatoires	0.96	0.96	1.0
Gradient Boosting	0.98	0.98	1.0
Extreme Gradient Boosting	0.97	0.97	1.0
Adaptative Boosting	0.99	0.99	1.0

Nous pouvons remarquer que le meilleur modèle est l'Adaptative Boosting, en effet ce dernier offre des résultats très satisfaisants se rapprochant de la perfection. Un léger surapprentissage est cependant à noter, un problème qui peut être réglé en effectuant une « RandomizedSearchCV » par exemple ou en choisissant d'autres hyperparamètres à optimiser.

Nous pouvons, de ce fait, utiliser ces 2 modèles prédictifs (le modèle Gradient Boosting ainsi que le modèle Adaptative Boosting) afin de calculer le taux de « Churn » prédit pour un horizon de 1 mois, pour le mois de « Septembre 2019 ». Nous aurons les résultats suivants :

Tableau 8 : Utilisation des meilleurs modèles pour prédire le Churn

Modèle	Taux de Churn prédit	Taux de Churn réel	Pourcentage d'erreur relative
Gradient Boosting	12,893%	13.564%	5.204%
Adaptative Boosting	13.427%	13.564%	1.021%

Nous remarquons que les résultats de l'AdaBoost sont sensiblement meilleurs par rapport à ceux du Gradient Boosting mais que les 2 modèles nous donnent des résultats très satisfaisants avec une erreur relative de 5% ou moins.

Nous pouvons utiliser les modèles retenus afin d'effectuer des prévisions à des horizons temporels mensuels modulables. Cela veut dire que par exemple lorsque nous voulons prédire le « Churn Rate » à des horizons de N mois, nous devrons reculer de N mois dans les MRR retenus puis retenir les 3 mois précédents en marge de la période d'observation. Il s'agira ensuite de prendre le mois d'après pour étiqueter les données.

6. Implémentation

Après avoir prédit le churn sur des périodes temporelles données (1, 3, 6 mois et 1 an), il faudra alors effectuer un benchmark et le situer sur un référentiel.

Pour ce faire nous avons mené une méta-analyse des études sur le churn pour avoir une idée sur à quel point le taux de résiliation est élevé ? Et est-ce le même pour toutes les entreprises ?

6.1. Churn Rate Benchmarks (Le seuil de 5%)

Les opinions sur le churn rate "idéal" du SaaS sont nombreuses et la plupart des commentateurs semblent partager le même point de vue :

« ...un taux d'attrition acceptable se situe entre 5 et 7 % **annuellement**, selon que l'on mesure les clients ou les revenus. » - Lincoln Murphy, Sixteen Ventures.[52]

Lorsque l'on examine les mathématiques en jeu, la logique qui sous-tend cet avis devient immédiatement évidente.

Si l'on suppose qu'une start-up compte 1 000 clients, un churn rate annuel de 5 % se traduirait par la perte de 50 clients au cours d'une seule année - ce qui n'est pas idéal, mais facile à compenser par l'acquisition de nouveaux clients.

Si nous comparons ce chiffre à un taux de désabonnement mensuel de 5 %, la même start-up perdrait 460 clients en un an, ce qui l'obligerait à remplacer près de la moitié de sa clientèle chaque année, juste pour atteindre son seuil de rentabilité.

[52] Sixteen Ventures SaaS Churn Rate

6.2. Churn Mensuel Vs Churn Annuel

La différence est frappante, car le taux d'attrition mensuel s'accroît avec le temps. Alors qu'un taux de désabonnement annuel de 5 % est mesuré sur l'ensemble de l'année.

$$1000 * 0.95 = 950$$

5% de désabonnement mensuel réduit le nombre de clients de 5% supplémentaires, chaque mois :

$$\text{Janvier} : 1000 * 0.95 = 950$$

$$\text{Février} : 950 * 0.95 = 903$$

$$\text{Mars} : 903 * 0.95 = 857$$

...

$$\text{Décembre} : 569 * 0.95 = 540$$

Le taux de désabonnement annuel et mensuel indique les mêmes informations (clients perdus), mais sur des périodes différentes. Pour ajouter à la confusion, il n'existe pas de méthode standardisée pour rendre compte des taux d'attrition : de nombreuses enquêtes ici font état d'une attrition annuelle des clients, tandis que d'autres font état d'une attrition mensuelle.

6.3. La Théorie Rencontre La Pratique

Pour simplifier les choses, nous pouvons utiliser les formules suivantes pour convertir le taux de désabonnement annuel en taux mensuel, et vice versa.

$$Churn\ rate\ mensuel = 1 - (1 - Chrun\ rate\ annuel)^{1/12}$$

$$Churn\ rate\ annuel = 1 - (1 - Churn\ rate\ mensuel)^{12}$$

Selon la formule ci-dessus, notre taux d'attrition annuel "idéal" de 5 à 7 % équivaut à un taux d'attrition mensuel de seulement 0,4 %, soit une perte d'environ 1 client sur 200.

$Churn\ rate\ mensuel = 1 - (1 - 0.05)^{12} = 0.4\%$

Mais demandez à n'importe quel fondateur de SaaS quel est son taux d'attrition, et il y a de fortes chances qu'il ait un taux d'attrition mensuel bien supérieur à 0,4 %. Cela signifie-t-il qu'il y a une épidémie de désabonnement dans le monde du SaaS ? Ou bien notre objectif "idéal" en matière d'attrition est-il tout simplement irréaliste ?

6.4. Les Problèmes Du Taux De Désabonnement

Une analyse sur six enquêtes a été réalisée en **Annexe X**, et nous avons une dichotomie claire en matière de résiliation : une série d'études semble suggérer qu'un taux de résiliation annuel de 5 à 10 % est courant ; une autre, un taux de résiliation mensuel de 5 à 10 %. Alors, que se passe-t-il ?

6.4.1. La Taille De L'entreprise

Les différents ensembles de données se répartissent en deux catégories : moins de 1 000 000 $ MRR (Baremetrics, Groove, Open Startups), et plus de 1 000 000 $ MRR (Pacific Crest, Totango, Blossom Ventures), ou plus généralement, grands et petits.

Le taux "idéal" de 5 à 7 % d'attrition annuelle semble se vérifier pour les grandes entreprises de SaaS, mais les petites entreprises semblent avoir un taux d'attrition beaucoup plus élevé.

Cela s'explique probablement par le fait que la plupart des "grandes" entreprises SaaS (et certainement la plupart des entreprises publiques) ciblent des clients professionnels, ce qui a un impact énorme sur le taux de résiliation :

- La facturation annuelle et la durée plus longue des contrats rendent le taux de résiliation plus difficile.
- Une ACV plus élevée signifie que les décisions sont généralement considérées comme étant à "long terme".
- Les entreprises sont moins sensibles au prix que les petites entreprises.

Comparez cela aux PME que la plupart des petites entreprises SaaS ciblent, et les énormes différences de taux de désabonnement deviennent compréhensibles :

- Une facturation mensuelle et des contrats plus courts facilitent grandement la résiliation.
- La facturation mensuelle et les contrats plus courts facilitent la résiliation.
- La volatilité des flux de trésorerie peut entraîner des annulations fréquentes.

6.4.2. Sensibilité Aux Prix Spécifique À L'industrie

De la même manière que certains types de clients sont plus enclins à la résiliation, différents types de logiciels auront des prédispositions différentes à la résiliation.

Si vous regardez votre propre pile technologique, vous verrez probablement certains produits que vous considérez comme essentiels et d'autres comme "agréables à avoir". Il est probable qu'un outil financier ou commercial sera moins sujet à la désaffection qu'un outil marketing, simplement parce qu'il est perçu comme étant plus directement responsable des revenus.

Il en va de même pour les outils de niche, ou ceux qui ont peu de concurrents - plus il serait coûteux de passer à un autre outil, plus votre taux de désabonnement sera faible.

6.4.3. Des Données Incohérentes

Il n'y a pas beaucoup de données disponibles sur le taux de désabonnement - et les informations qu'on a pu trouver ne sont pas toujours aussi claires.

La raison en est simple : les taux d'attrition élevés sont mauvais, car ils permettent littéralement d'enregistrer le nombre de clients qui quittent votre service chaque mois. Peu d'entreprises sont aussi courageuses que Buffer et HubStaff, et celles qui acceptent de partager leurs taux d'attrition ne le feront probablement que si elles sont anonymes et que les données exactes sont dissimulées à l'aide d'une fourchette.

6.4.4. L'obscurcissement Intentionnel

Les entreprises publiques ont encore plus intérêt à cacher les "mauvais" paramètres : cela peut avoir un impact sur le cours de leurs actions.

Selon la loi, les entreprises publiques doivent rendre compte de leurs performances. L'adoption de mesures standardisées (comme les GAAP) faciliterait la comparaison entre des entreprises similaires, ce qui a conduit beaucoup d'entre elles à développer des méthodes de reporting "propriétaires" qui rendent pratiquement impossible toute comparaison directe entre concurrents.

Et malgré son importance, il n'existe aucune obligation légale de rendre compte des taux d'attrition, ce qui explique le faible taux de réponse et les dizaines de calculs différents des taux d'attrition utilisés dans le rapport de Blossom Ventures.

6.4.5. Le Taux De Désabonnement Idéal

Dans le cas d'une grande entreprise SaaS bien établie, en voie d'introduction en bourse ou d'un autre type de sortie, les objectifs en matière de taux d'attrition sont très clairs : atteindre un taux d'attrition annuel de 5 à 7 %, c'est une caractéristique constante des grandes entreprises prospères.

Mais si, nous sommes dans un cas d'une entreprise à un stade plus précoce, les choses ne sont pas aussi claires. Même une entreprise SaaS prospère comme Buffer se bat encore avec un taux de résiliation de 5% par mois, et si c'est un cas d'une entreprise nouvelle dans le monde de l'adéquation produit/marché, il y a des raisons de croire que ce taux de résiliation sera plus élevé.

Bien qu'il soit difficile de donner un point de repère précis, les six études que nous avons analysées suggèrent la même chose : un taux d'attrition mensuel de 5 % est assez courant et, comme le montrent les études de Buffer, Baremetrics et Convertkit, il ne constitue pas un obstacle évident à la croissance.

Un "bon" taux d'attrition typique pour les entreprises SaaS qui ciblent les petites entreprises est de 3 à 5 % par mois. Plus les entreprises ciblées sont grandes, plus le taux d'attrition doit être faible car le marché est plus petit. Pour un produit d'entreprise (de l'ordre de X 000 à XX 000 dollars par mois), le taux d'attrition doit être inférieur à 1 % par mois. La plupart des premières entreprises SaaS que nous avons observées ont un taux de désabonnement d'environ 10 à 15 % la première année, car elles déterminent exactement ce que leur produit doit faire, puis elles sont capables de le réduire assez rapidement.

Points Clés :

- Le taux d'attrition est mauvais mais inévitable, il est donc important de le suivre et de l'améliorer au fil du temps.
- Un taux d'attrition annuel de 5 à 7 % est un excellent point de référence à viser - si l'entreprise SaaS établie et mature.
- Si c'est le cas d'une entreprise à un stade plus précoce, ou les PME, faudra s'attendre à ce que le taux de désabonnement soit plus proche de 5 % par mois.
- À mesure que le produit se développe et que le modèle d'entreprise mûrit, le taux de désabonnement devrait s'améliorer.
- Les taux d'attrition absolus ne sont pas aussi importants que les changements de taux d'attrition.

Si on pouvait tracer l'évolution du Churn mensuel idéal, celle-ci devrait ressembler à cela :

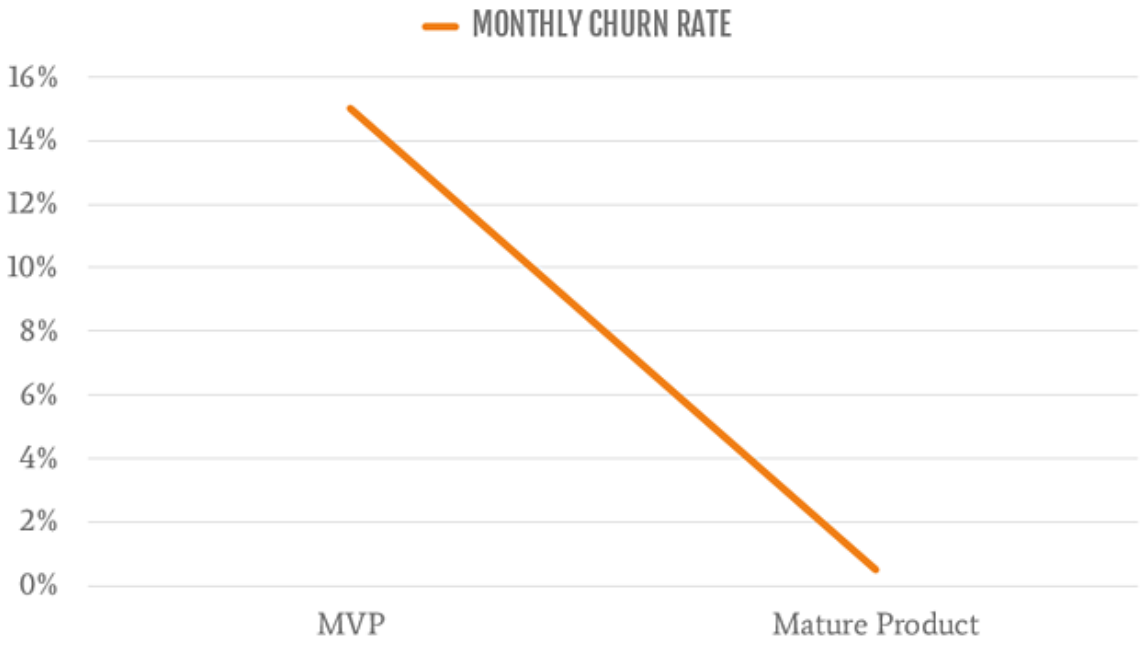


Figure 71 : Progression du Churn mensuel idéal

Conclusion :

Dans cette dernière partie, nous avons détaillé le processus de conception de notre modèle prédictif en suivant les étapes de la méthodologie de gestion de projet Machine Learning appelée CRISP-DM. En marge de cette dernière, nous avons commencé par comprendre l'objectif business concerné, puis après avoir préparé notre base de données, nous l'avons utilisé afin de construire des modèles prédictifs à partir des différents algorithmes de classification issus de l'apprentissage supervisé. Nous avons par la suite mis en place une procédure d'évaluation de ces modèles, ce qui nous a donné des résultats satisfaisants notamment pour les modèles de Gradient Boosting et Adaptative Boosting qui ont été sélectionnés comme étant les meilleurs modèles pour effectuer des prédictions pour différents horizons temporels. Ces derniers ont eu les meilleurs scores en marge de l'évaluation par le taux de réalité.

L'implémentation de cette solution a également été discutée, les modèles prédictifs seront soumis à un benchmark afin de situer les taux de « Churn » obtenus, et ainsi pouvoir prendre des décisions pondérées dans le secteur du SaaS.

Conclusion Générale :

Conclusion Générale

Pour conclure, ce présent document permet de décrire le travail que nous avons effectué auprès du cabinet de conseil, d'expertise comptable et d'audit KPMG, et plus précisément au sein du département « Deal Advisory », responsable de l'accompagnement d'entreprises en période de transaction de type fusion/acquisition ou de restructuration, comprenant l'équipe « Deal Analytics », chargée de missions d'appui analytique et informatique. L'ouvrage accompli consiste en la conception d'un programme informatique sous forme de modèle prédictif visant à utiliser les données de la mission effectuée afin de prédire le « Churn Rate » à des horizons temporels mensuels modulables, pour servir par la suite d'outil d'aide à la décision par son implémentation.

Après avoir défini les concepts théoriques liés au marché des fusions et acquisitions, ainsi qu'aux algorithmes de Machine Learning, nous avons effectué un diagnostic de l'existant permettant de situer l'apport de notre solution et son impact potentiel sur l'amélioration des processus de due diligence de KPMG.

L'étape suivante fut de dérouler les étapes décrites par la méthodologie de projet Machine Learning nommée CRISP-DM, qui a eu pour effet de structurer le travail accompli par nos soins. Ces étapes peuvent être résumées par les points suivants :

- Compréhension de l'objectif business que remplira l'aboutissement de ce projet ;
- Acquisition et compréhension de la base de données utilisée pour concevoir notre modèle ;
- Préparation de la base de données en y apportant les modifications nécessaires afin d'être apte à être considérée comme donnée d'entrée aux algorithmes informatiques du Machine Learning ;
- Conception de plusieurs modèles de classification binaires issues de l'apprentissage supervisé, sur la base de données initiale puis la base de données ayant subi un sur-échantillonnage ;
- Évaluation des différents modèles construits suivant les indicateurs de performances retenus que sont le « F1 score », le « AUC » ainsi que le « Recall » puis sélection des meilleurs modèles et prévision du « Churn » à horizon temporel d'un mois ;
- Proposition d'une implémentation de l'outil conçu en marge du processus de Due Diligence accompli par le département « Deal Advisory ».

Les résultats obtenus furent les suivants :

- Les modèles retenus ont été ceux basés sur l'apprentissage ensembliste, et qui ont obtenu des scores élevés en marge de notre évaluation.
- Les variables que nous avons « Scrappé » à l'aide d'un programme informatique que nous avons conçu en marge de l'accomplissement de la préparation des données ont permis d'apporter une dimension de réalité économique et financière supplémentaire aux variables dont nous disposions au départ.
- La partie implémentation permet de cerner l'apport concret de notre solution au sein d'une Due Diligence potentielle que pourrait accomplir le département auprès d'un éditeur de SaaS.

Le travail effectué a été accueilli de manière positive au sein de la division qui nous accueillait et permet, outre son utilisation en marge d'une Due Diligence future dans le secteur du SaaS, d'intégrer comme « perspectives » de nouvelles variables dans la phase de collecte de données sur la mission par des outils de « Scrapping » mais aussi d'ouvrir le champ à la conception de modèles prédictifs dans d'autres secteurs, en se basant sur les résultats satisfaisants obtenus par nos modèles.

En plus de nous avoir permis d'utiliser de manière concrète l'étendue des connaissances qui nous ont été inculqués au cours de notre formation en Génie Industriel, l'accomplissement de ce projet fut une opportunité unique pour nous de nous ouvrir au métier du consulting en cabinet de conseil, qui représente un objectif de carrière pour nous. Nous nous sommes également ouverts au vaste domaine de la science des données, en pleine expansion et qui représente un axe académique à fort potentiel pour les années à venir. Nous avons acquis, au cours de notre expérience, aussi bien des connaissances palpables telles que la comptabilité, la finance ainsi que l'informatique mais également des compétences plus abstraites telles que la rigueur, la communication, le travail d'équipe et l'importance d'un environnement de travail sain dans la performance des collaborateurs.

Bibliographie

Webographie :

- ABDULLAH Sammy. *109% net dollar retention is the new standard.* [en ligne]. [Consulté le 28/04/2020] Disponible sur : https://medium.com/@sammyabdullah/109-net-dollar-retention-is-the-new-standard-8c21685e5f99
- BROWNLEE, Jason. *SMOTE for imbalanced classification with Python*. [en ligne]. [Consulté le 09/06/2020] Disponible sur : https://machinelearningmastery.com/smote-oversampling-for-imbalanced-classification//
- CHAKURE, Afroz. *K-Nearest Neighbors (KNN) Algorithm*. [en ligne]. [Consulté le 29/04/2020] Disponible sur : https://towardsdatascience.com/k-nearest-neighbors-knn-algorithm-bd375d14eec7
- Corporate Finance Institute. Mergers Acquisitions M&A Process. [en ligne]. [Consulté le 29/04/2020] Disponible sur : https://corporatefinanceinstitute.com/resources/knowledge/deals/mergers-acquisitions-ma-process/
- DataScience Learner. *Gradient Boosting Hyperparameters Tuning : Classifier Example*. [en ligne]. [Consulté le 09/06/2020] Disponible sur : https://www.datasciencelearner.com/gradient-boosting-hyperparameters-tuning/
- Eurocloud France. *Pourquoi les experts financiers peinent à comprendre le business SaaS?* [en ligne] [Consulté le 09/04/2020]. Disponible à l'adresse : https://www.eurocloud.fr/les-experts-financiers-peinent-comprendre-business-saas/
- Google Developpers. *Classification : ROC et AUC*. [en ligne]. [Consulté le 17/05/2020] Disponible sur :https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc?hl=fr
- JACOBEUS, Nicolas. *Métriques SaaS, quelles sont celles qui comptent vraiment ?* [en ligne]. [Consulté le 12/04/2020] Disponible sur : https://www.belighted.com/fr/blog/metriques-saas
- KUPOR, Scott et KASIREDDY, Preethy. *Understanding SaaS. Why the pundits have it wrong?* [en ligne]. [Consulté le 10/04/2020]. Disponible à l'adresse : https://a16z.com/2014/05/13/understanding-saas-valuation-primer/
- LELOUP, Laurent. *Tendances 2019 en fusions*-acquisitions [en ligne]. [Consulté le 06/04/2020] Disponible à l'adresse: https://www.finyear.com/Tendances-2019-en-fusions-acquisitions_a40693.html
- MLMath.io. *Math Behind Decision Tree Algorithm*. [en ligne]. [Consulté le 18/05/2020] Disponible sur : https://medium.com/@ankitnitjsr13/math-behind-decision-tree-algorithm-2aa398561d6d
- NAIR, Amal. *Implemeting Baysian Optimization on XGBoost : A beginner's guide*. [en ligne]. [Consulté le 10/06/2020] Disponible sur : https://analyticsindiamag.com/implementing-bayesian-optimization-on-xgboost-a-beginners-guide/
- OpenClassrooms. *Entraînez un modèle prédictif linéaire*. [en ligne]. [Consulté le 12/05/2020] Disponible sur : https://openclassrooms.com/fr/courses/4444646-entrainez-un-modele-predictif-lineaire
- SAYAD, Saed. *Decision Tree-Classification*. [en ligne]. [Consulté le 15/05/2020] Disponible sur : https://www.saedsayad.com/decision_tree.htm
- SAYAD Saed. *Support Vector Machine*. [en ligne]. [Consulté le 12/05/2020] Disponible sur : https://www.saedsayad.com/support_vector_machine.htm
- YIU Tony. *Understanding Random Forest*. [en ligne]. [Consulté le 16/05/2020] Disponible sur : https://towardsdatascience.com/understanding-random-forest-58381e0602d2

Articles :

- KPMG Australia. Data Analytics in M&A: A new weapon in the modern deal maker's armoury [en ligne]. [Consulté le 29/03/2020]. Disponible à l'adresse :
- https://home.kpmg/content/dam/kpmg/au/pdf/2018/data-analytics-in-mergers-acquisitions.pdf
- KPMG Global : The race for game-changing transformation and strategic growth [en ligne]. [Consulté le 04/04/2020]. Disponible à l'adresse : https://assets.kpmg/content/dam/kpmg/xx/pdf/2019/03/consumer-and-retail-m-and-a-trends-2019.pdf
- QUINLAN, J.R. Induction of Decision trees, Machine Learning [en ligne]. [Consulté le 27/05/2020]. Disponible à l'adresse : https://link.springer.com/article/10.1007/BF00116251#citeas
- TROTT Orlando : Understanding SaaS M&A Due Diligence [en ligne]. [Consulté le 02/04/2020]. Disponible sur : <https://medium.com/data-dump/understanding-saas-m-a-due-diligence-f36216370e62>

Ouvrages :

- GERON Aurélien. Hands on Machine Learning with Scikit-Learn, Keras and TensorFlow. 2nd edition, O'reilly Media, USA, 2019, 483 p. IBSN 978-1-492-03264-9.
- HEIDE, J.B. and WEISS, A.M. Vendor Consideration and Switching Behavior for Buyers in High-Technology Markets. The Journal of Marketing, 1995, 59, pp- 30-43.
- MEIER, Olivier, SCHIER, Guillaume. Fusions acquisitions, Stratégie, Finance, Management, 3ème edition. DUNOD, Paris, 2009, 322 p. IBSN 978-2-10-053861.
- SAINT-CIRGUE, Guillaume. Apprendre le machine learning en une semaine. Edition : MachineLearnia. 2019, 100 p.
- SHERMAN, Andrew J. Mergers Acquisitions From A to Z 3rd Edition. AMACOM, USA, 2011, 318 p. IBSN 978-0-8144-1383-8.

Thèses :

- BUHLMANN, Peter and HOTHORN, Torsten. Boosting algorithms : regularization prediction and model fitting [en ligne]. Thèse : ETH Zurich and Universitat Erlangen-Nurnberg. 2005. [Consulté le 15/06/2020]. Disponible à l'adresse : https://wb.stanford.edu/~hastie/Papers/buehlmann.pdf
- RAUTIO, Anton. Churn Prediction in SaaS using Machine Learning. Mémoire de Master : Faculty of Management and Business. Tampere : Tampere University. 2019, 56 p.
- SERGUE, Marie. Customer Churn Analysis and Prediction using Machine Learning for a B2B company. Project in Engineering physics. Stockholm : KTH Royal Institute of Technology. 2020, 42 p.

Annexe

Annexe A : Supervised learning

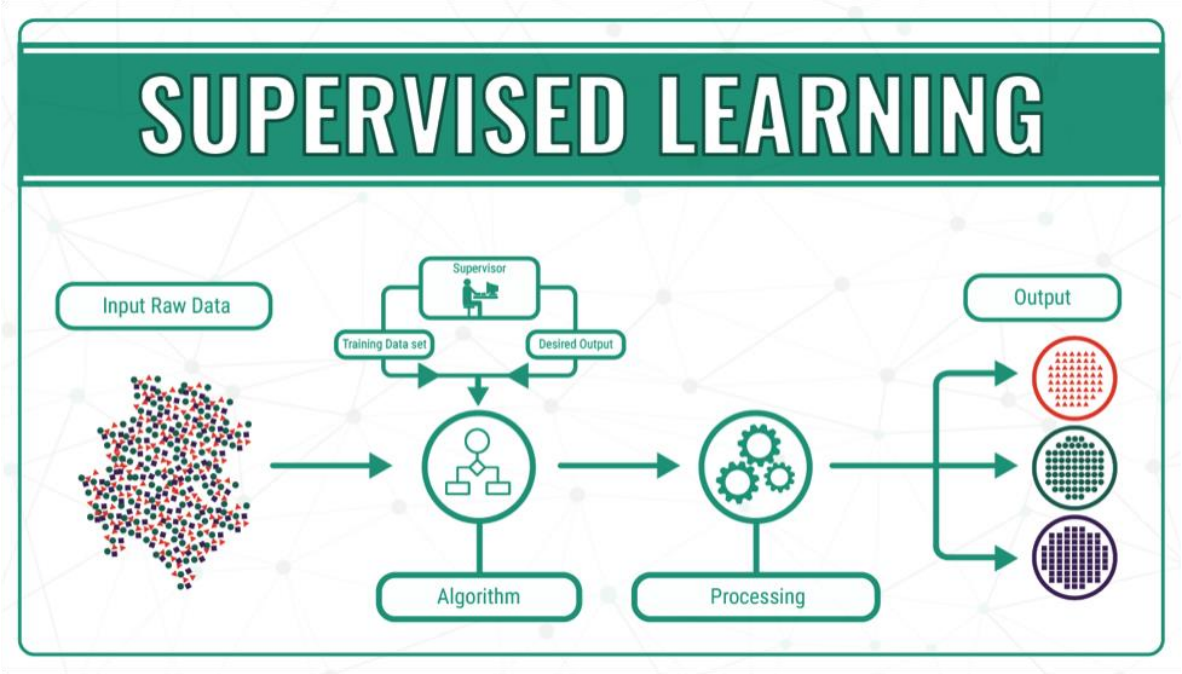


Figure A.1 : Apprentissage supervisé (mc.ai,2019)

Formalisation de l'apprentissage supervisé :

On suppose que les objets étudiés, pouvant être complexes à l'origine, sont représentés dans un format numérique structuré. On aura donc :

- L'objet X_i est représenté par un vecteur noté x_i défini par plusieurs variables
- On associe à chaque x_i une valeur cible notée y_i

Nous aurons comme données à notre disposition une matrice X des observations avec n lignes (occurrences) et p colonnes (variables) ainsi qu'un vecteur colonne (variable cible) y de n éléments :

$$X = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1p} \\ x_{21} & x_{22} & \cdots & x_{2p} \\ \vdots & \cdots & \cdots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{np} \end{pmatrix} \text{ et } y = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}$$

L'espace des couples observés $E = (x_1, y_1), \dots, (x_i, y_i), \dots, (x_n, y_n)$ est appelé ensemble d'entrainement ou d'apprentissage, qui sont les données étiquetées. Etant donne un ensemble d'entrainement E, on cherche à déterminer $f : \mathrm{X} \rightarrow Y$ une fonction modélisant la relation entre les objets décrits dans la matrice X et la variable cible Y.

Cependant, et en tenant compte du bruit pouvant existant dans les données, on ne supposera pas une relation déterministe, on posera donc le problème en les termes suivants :

$$f(X) = Y + \varepsilon$$

Où ε est l'erreur ou le résidu. Il s'agira d'onc d'approximer f en commettant le moins d'erreur possible sur E afin de faire de bonnes prédictions pour les valeurs de X non encore observées.

Le filtre de spam est un exemple de problème commun en ce qui concerne l'apprentissage supervisé. Ce dernier est de type classification puisque la variable cible est discrète avec 2 résultats (classes) possibles (Spam, Non Spam). Une illustration de ce problème est donnée ci-après :

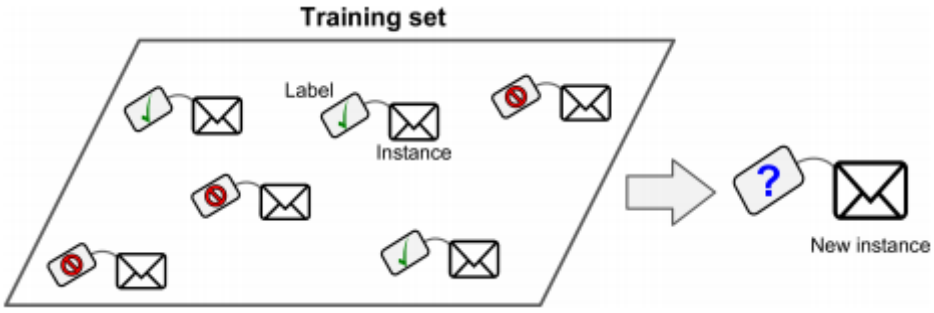


Figure A.2 : Visualisation du problème de classification de mails[53]

[53] Géron,2019

Annexe B : Unsupervised learning

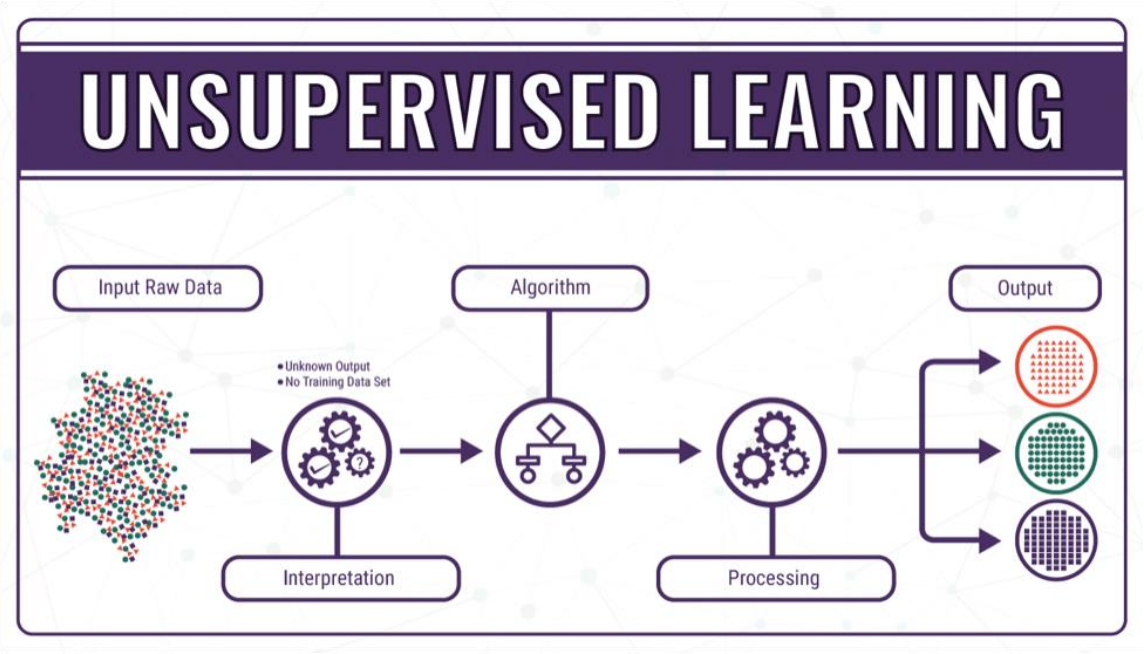


Figure B.1 : Apprentissage non supervisé[54]

L'apprentissage non supervisé n'est encore qu'à une étape assez reculée dans son utilisation pour résoudre des problèmes, cependant son potentiel est hors normes. Parmi les problèmes traités actuellement par le « unsupervised learning », on peut citer la détection d'anomalies, par exemple dans la détection de transactions suspectes relatives à une carte de crédit pour but de prévenir la fraude ou alors les défauts de fabrications. Un autre problème assez commun est le « rule learning » ou l'objectif est d'effectuer une étude approfondie dans de grands ensembles de données afin de déceler des relations entre différents attributs. A titre d'exemple, on peut effectuer des règles d'association entre ses ventes pour cerner les préférences communes des clients d'un supermarché et ainsi prendre des décisions en proposant des réductions sur des produits.

[54] Mc.ai 2019

Annexe C : Reinforced learning

La figure ci-dessous nous donne un schéma explicatif de l'apprentissage par renforcement :

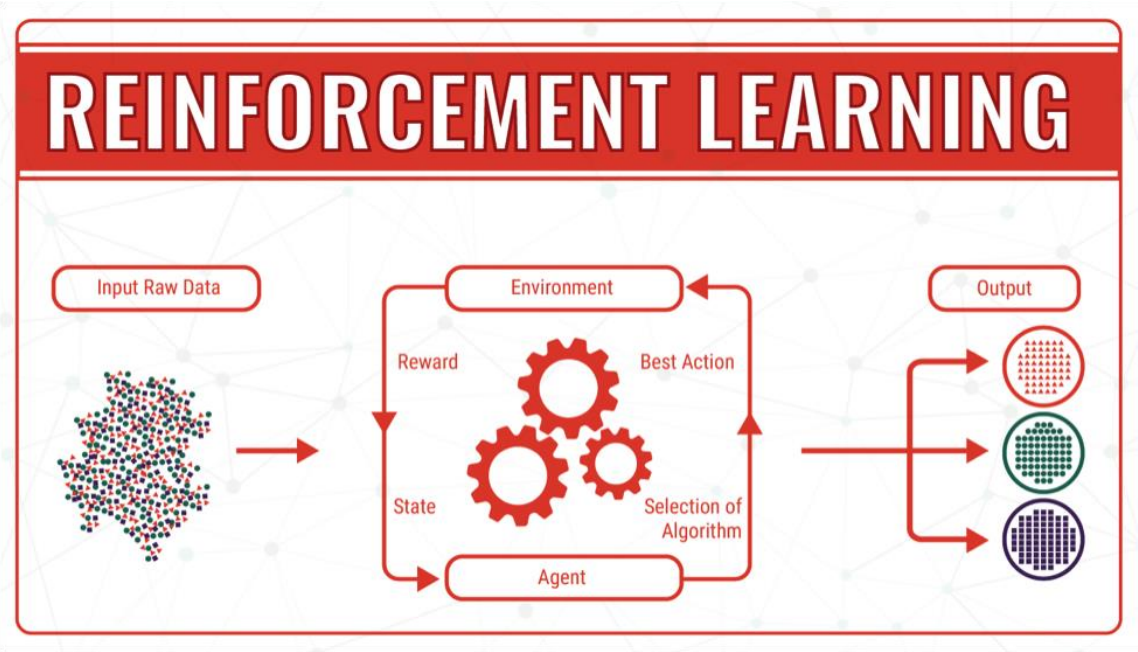


Figure C.1 : Apprentissage par renforcement (mc.ai,2019)

Annexe D : Arbre de décision entropie

L'entropie :

Définie comme étant la quantité d'information nécessaire afin de décrire un échantillon. Si ce dernier est homogène ne contenant que des éléments similaires, alors l'entropie est nulle. Sinon, si l'échantillon est équitablement réparti entre ses éléments alors l'entropie atteint son maximum qui est de 1.

La formule permettant de calculer l'entropie est définie ainsi :

$$Entropy = -\sum_{i=1}^{n} p_i * \log(p_i)$$

Annexe E : Arbre de décision autres types

Iterative Dichotomiser 3 (ID3)

Cet algorithme, destiné seulement à des problèmes de classification, utilise deux métriques, pour une variable (attribut) A et une décision S, on aura :

- L'entropie : $Entropy(S) = -\sum_{i=1}^{n} p_i * \log_2(p_i)$
- Le Gain d'Information : $Information\ Gain(S, A) = Entropy(S) - \sum_{i=1}^{n} P(S|A) \cdot Entropy(S|A)$

A partir de l'entropie des 2 classes cibles, on calcul à chaque fois le gain d'information obtenu par une variable expliquant la cible. Ce processus est également itératif et utilise à chaque fois le gain le plus important afin de construire les branches de l'arbre.

Iterative Dichotomiser 4.5 (C4.5)

Cet algorithme introduit en 1993 est une version améliorée du modèle ID3 prenant en compte, en plus des seules variables catégorielles que prenait en compte son prédécesseur, les variables numériques devant être devant subir cependant une discrétisation. On utilise pour le découpage l'entropie de Shannon définie par :

$$Shannon\ Entropy\ (y, K) = -\sum_{l \in Y} \Big(P(y = l) * \log_2\big(P(y = l)\big)\Big)$$

Avec K représentant l'ensemble d'apprentissage, Y l'ensemble des valeurs que peut prendre y et l qui est une classe donnée.

Soit : $(x_1, y_1), \ldots, (x_m, y_m)$ où $x_i \in \varphi$, $y_i \in \{-1,+1\}$.

Initialiser : $D_1(i) = \frac{1}{m}$ pour i = 1,...,m.

Pour t = 1,...,T :

- Entraîner le classificateur faible en utilisant la distribution D_t.
- Obtenir une hypothèse faible $h_t : \varphi \rightarrow \{-1,+1\}$.
- Objectif : sélectionner h_t avec une faible erreur pondérée :

$\varepsilon_t = Pr_{i\sim Dt}[h_t(x_i) \neq y_i]$.

- Choisir $\alpha_t = \frac{1}{2}\ln(\frac{1-\varepsilon t}{\varepsilon t})$.
- Actualiser, Pour i = 1,…,m:

$$D_{t+1}(i) = \frac{D_t(i)\exp(-\alpha_t y_i h_t(x_i))}{z_t}$$

Où Z_t est un facteur de normalisation (choisi pour que D_{t+1} soit une distribution).

Sortie de l'hypothèse finale :

$$H(x) = signe(\sum_{t=1}^{T} \alpha_t h_t(x))$$

Figure F.1 *L'algorithme de boosting AdaBoost.*

Commençons par considérer un ensemble de données avec N points, ou lignes, dans un ensemble de données.

$$x_i \in \mathbb{R}^n, y_i, \in \{-1,1\}$$

Dans le cas présent :

n est la dimension des nombres réels, ou le nombre d'attributs dans notre ensemble de données

x est l'ensemble des points de données

y est la variable cible qui est soit -1 soit 1 car il s'agit d'un problème de classification binaire, désignant la première ou la deuxième classe (par exemple, Fit vs Not Fit)

Nous calculons les échantillons pondérés pour chaque point de données.

AdaBoost attribue un poids à chaque exemple de formation afin de déterminer sa signification dans l'ensemble de données de formation. Lorsque les poids attribués sont élevés, cet ensemble de

points de données d'entraînement est susceptible d'avoir une plus grande influence sur l'ensemble d'entraînement. De même, lorsque les poids attribués sont faibles, ils ont une influence minimale sur l'ensemble des données d'entraînement.
Au départ, tous les points de données auront le même échantillon pondéré w :

$$\omega = 1/N \in [0,1]$$

Où N est le nombre total de points de données.

La somme des échantillons pondérés est toujours égale à 1, de sorte que la valeur de chaque poids individuel sera toujours comprise entre 0 et 1. Ensuite, nous calculons l'influence réelle de ce classificateur sur la classification des points de données à l'aide de la formule.

$$\alpha_t = \frac{1}{2}\ln\frac{(1 - TotalErreur)}{TotalErreur}$$

Alpha est le degré d'influence que cette souche aura dans la classification finale. L'erreur totale n'est rien d'autre que le nombre total de classifications erronées pour cet ensemble de formation divisé par la taille de l'ensemble de formation. Nous pouvons tracer un graphique pour Alpha en introduisant différentes valeurs d'erreur totale allant de 0 à 1.

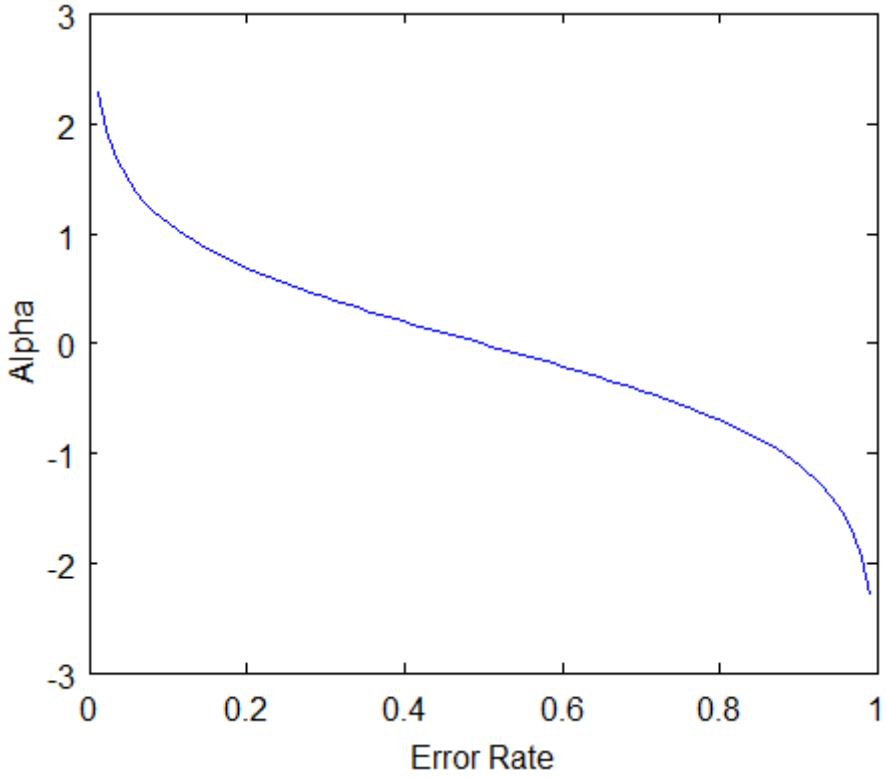


Figure F.1 : *Graphique de Aplha en function de l'erreur* [55]

Remarquez que lorsqu'une souche de décision fonctionne bien ou ne présente pas d'erreurs de classification, cela se traduit par un taux d'erreur de 0 et une valeur alpha positive relativement importante.

Si la souche se contente de classer la moitié correctement et l'autre moitié incorrectement (un taux d'erreur de 0,5, ce qui n'est pas mieux qu'une estimation aléatoire !), la valeur alpha sera de

[55] (Source: Chris McCormick) http://mccormickml.com/2013/12/13/adaboost-tutorial/

0. Enfin, si la souche ne cesse de donner des résultats mal classés (faites simplement le contraire de ce que dit la souche !), la valeur alpha sera alors négative et importante.

Après avoir saisi les valeurs réelles de l'erreur totale pour chaque souche, il est temps pour nous de mettre à jour les poids des échantillons que nous avions initialement pris pour 1/N pour chaque point de données. Pour ce faire, nous utiliserons la formule suivante :

$$\omega_i = \omega_{i-1} * e^{\pm\alpha}$$

En d'autres termes, le nouveau poids de l'échantillon sera égal à l'ancien poids de l'échantillon multiplié par le nombre d'Euler, porté à plus ou moins alpha (que nous venons de calculer à l'étape précédente).
Les deux cas pour l'alpha (positif ou négatif) indiquent :

L'alpha est positif lorsque la sortie prévue et la sortie réelle concordent (l'échantillon a été classé correctement). Dans ce cas, nous diminuons le poids de l'échantillon par rapport à ce qu'il était auparavant, puisque nous obtenons déjà de bons résultats.
Alpha est négatif lorsque la sortie prévue ne correspond pas à la classe réelle (c'est-à-dire que l'échantillon est mal classé). Dans ce cas, nous devons augmenter le poids de l'échantillon afin que la même erreur de classification ne se répète pas dans la souche suivante. C'est ainsi que les souches sont dépendantes de leurs prédécesseurs.

Annexe G : Comparaison courbes ROC

La figure suivante présente différentes courbes ROC permettant de visualiser des AUC différents et de juger de la performance du modèle de classification :

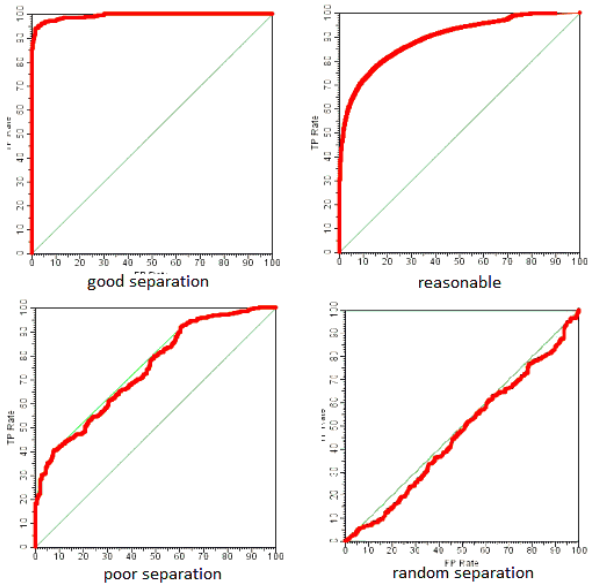


Figure G.1 : Visualisation de la disparité des modèles de classifications selon leur courbe ROC

Annexe H : Algorithme SMOTE

Algorithme *SMOTE* (T, N, k)

Input : Nombre d'échantillons de classes minoritaires T ; Quantité de SMOTE N% ; Nombre de voisins les plus proches k.

Output : (N/100) * *T échantillons synthétiques de la classe minoritaire.*

(∗ *Si N est inférieur à 100 %, répartissez au hasard les échantillons des classes minoritaires, car seul un pourcentage aléatoire d'entre eux sera SMOTEd.* ∗)

Si N < 100

Puis *répartir au hasard les échantillons de la classe minoritaire T.*

$T = (N/100) * T$

$N = 100$

Fin si

$N = (int)(N/100)$ (∗*Le montant de SMOTE est supposé être un multiple entier de 100.*∗)
k = Nombre de voisins les plus proches
$numattrs$ = Nombre d'attributs
$Sample[\,][\,]$: tableau pour les échantillons originaux des classes minoritaires
$newindex$: tient un compte du nombre d'échantillons synthétiques générés, initialisé à 0
$Synthetic[\,][\,]$: réseau pour les échantillons synthétiques

(∗ Calculer k voisins les plus proches pour chaque échantillon de classe minoritaire uniquement. ∗)

pour $i \leftarrow 1$ à T

Calculer k voisins les plus proches pour i, et enregistrer les indices dans le nnarray
Populate (N, i, nnarray)

Fin-pour

Populate(N, i, nnarray) *(∗Fonction permettant de générer les échantillons synthétiques.∗)*

Tant que $N \neq 0$

Choisissez un nombre aléatoire entre 1 et k, appelez-le nn. Cette étape permet de choisir l'un des k plus proches voisins de i.

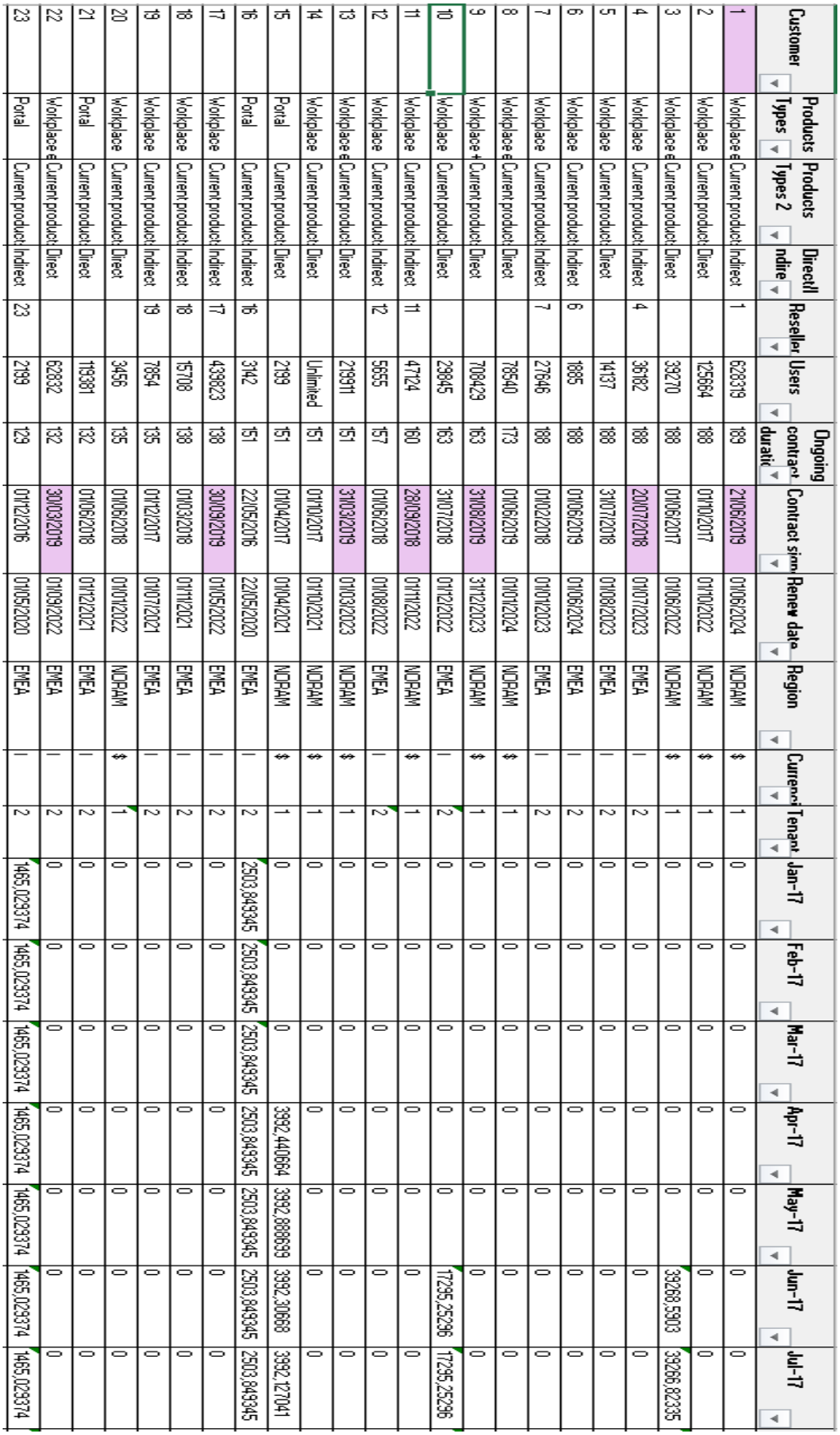
Pour attr ←1 **à** numattrs

Calculer : $dif = Sample[nnarray[nn]][attr] - Sample[i][attr]$
Caluler : $gap = random\ number\ between\ 0\ and\ 1$
$Synthetic[newindex][attr] = Sample[i][attr] + gap * dif$

Fin pour
$newindex++$
$N = N - 1$
Finir lorsque
Retour *(∗ Fin de la population. ∗)*
End of Pseudo-Code.

Annexe I : Base de données brute

Visualisation d'une partie de la base de données initiale dans la figure suivante.



Customer	Products Types	Products Types 2	Direct/Indire	Reseller	Users	Ongoing contract duratic	Contract sign	Renew date	Region	Currency	Tenant	Jan-17	Feb-17	Mar-17	Apr-17	May-17	Jun-17	Jul-17
1	Workplace a	Current product	Indirect	1	623319	189	21/06/2019	01/06/2024	NORAM	$	1	0	0	0	0	0	0	0
2	Workplace	Current product	Direct	125664	188	01/10/2017	01/10/2022	NORAM	$	1	0	0	0	0	0	0	0	0
3	Workplace a	Current product	Direct	39270	188	01/06/2017	01/06/2022	NORAM	$	1	0	0	0	0	0	0	39268,5903	39268,82335
4	Workplace	Current product	Indirect	36182	188	20/07/2018	01/07/2023	EMEA	I	2	0	0	0	0	0	0	0	0
5	Workplace	Current product	Direct	14137	188	31/07/2018	01/08/2023	EMEA	I	2	0	0	0	0	0	0	0	0
6	Workplace	Current product	Indirect	1885	188	01/06/2019	01/06/2024	EMEA	I	2	0	0	0	0	0	0	0	0
7	Workplace	Current product	Indirect	27846	188	01/02/2018	01/01/2023	EMEA	I	2	0	0	0	0	0	0	0	0
8	Workplace a	Current product	Direct	78540	173	01/06/2019	01/01/2024	NORAM	$	1	0	0	0	0	0	0	0	0
9	Workplace a	Current product	Direct	706429	163	31/08/2019	31/12/2023	NORAM	$	1	0	0	0	0	0	0	0	0
10	Workplace	Current product	Direct	29845	163	31/07/2018	01/12/2022	EMEA	I	2	0	0	0	0	0	0	17295,25296	17295,25296
11	Workplace	Current product	Indirect	47124	160	28/09/2018	01/11/2022	NORAM	$	1	0	0	0	0	0	0	0	0
12	Workplace	Current product	Indirect	5655	157	01/06/2018	01/08/2022	EMEA	I	2	0	0	0	0	0	0	0	0
13	Workplace a	Current product	Direct	219911	151	31/03/2019	01/03/2023	NORAM	$	1	0	0	0	0	0	0	0	0
14	Workplace	Current product	Direct	Unlimited	151	01/10/2017	01/10/2021	NORAM	$	1	0	0	0	0	0	0	0	0
15	Portal	Current product	Direct	2199	151	01/04/2017	01/04/2021	NORAM	$	1	0	0	0	0	3992,440664	3992,888639	3992,30668	3992,127041
16	Portal	Current product	Indirect	3142	151	22/05/2016	22/05/2020	EMEA	I	2	2503,849345	2503,849345	2503,849345	2503,849345	2503,849345	2503,849345	2503,849345	2503,849345
17	Workplace	Current product	Indirect	4338823	138	30/09/2019	01/05/2022	EMEA	I	2	0	0	0	0	0	0	0	0
18	Workplace	Current product	Indirect	15708	138	01/03/2018	01/11/2021	EMEA	I	2	0	0	0	0	0	0	0	0
19	Workplace	Current product	Indirect	7854	135	01/12/2017	01/07/2021	EMEA	I	2	0	0	0	0	0	0	0	0
20	Workplace	Current product	Direct	3456	135	01/06/2018	01/01/2022	NORAM	$	1	0	0	0	0	0	0	0	0
21	Portal	Current product	Direct	119381	132	01/06/2018	01/12/2021	EMEA	I	2	0	0	0	0	0	0	0	0
22	Workplace a	Current product	Direct	62832	132	30/09/2019	01/09/2022	EMEA	I	2	0	0	0	0	0	0	0	0
23	Portal	Current product	Indirect	2199	123	01/12/2016	01/05/2020	EMEA	I	2	1465,023374	1465,023374	1465,023374	1465,023374	1465,023374	1465,023374	1465,023374	1465,023374

Figure I.1 : Extrait de la base de données initiale fournie

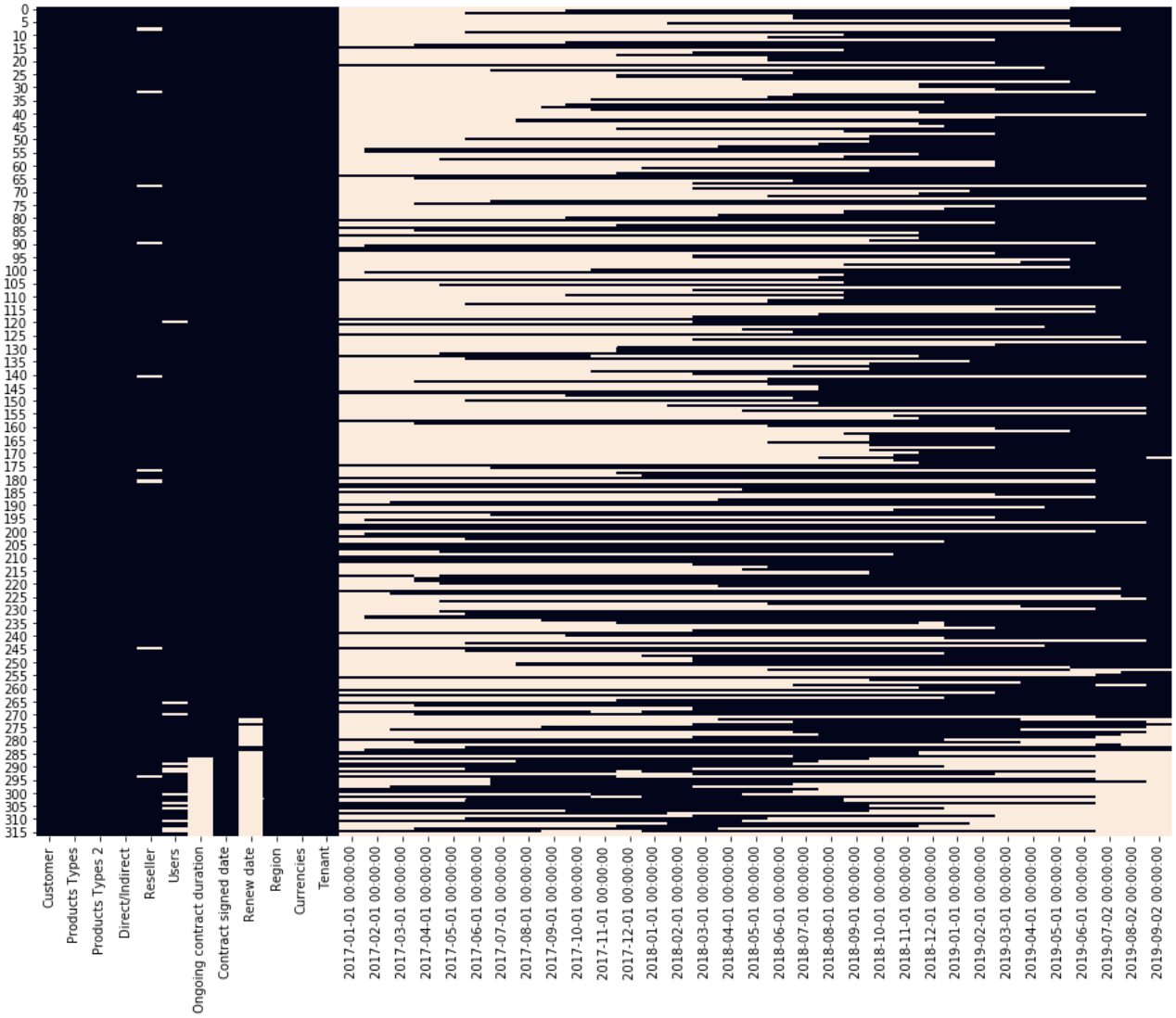


Figure I.2 : Graphique des valeurs manquantes pour notre base de données initiale

Annexe J : Outil de Scrapping

Pour extraire des données à l'aide du Scraping Web avec Python, vous devez suivre ces étapes de base :

- Trouver l'URL que vous voulez extraire
- Inspection de la page
- Trouver les données que vous souhaitez extraire
- Ecrire le code
- Exécuter le code et extraire les données
- Stocker les données dans le format requis

Bibliothèques utilisées pour le scraping Web :

- **Sélénium :** Sélénium est une bibliothèque de test pour le web. Elle est utilisée pour automatiser les activités des navigateurs.
- **BeautifulSoup :** BeautifulSoup est un paquet Python pour l'analyse de documents HTML et XML. Il crée des arbres d'analyse qui sont utiles pour extraire facilement les données.
- **Pandas :** Pandas est une bibliothèque utilisée pour la manipulation et l'analyse de données. Elle est utilisée pour extraire les données et les stocker dans le format souhaité.

Étape 1 : Trouver l'URL à scrapper

https://www.societe.com/

http://entreprises.lefigaro.fr/

https://www.zoominfo.com/

https://www.bloomberg.com/

Étape 2 : Inspecter la page

Les données sont généralement imbriquées dans des balises. Ainsi, nous examinons la page pour voir sous quelle balise sont imbriquées les données que nous voulons scraper.

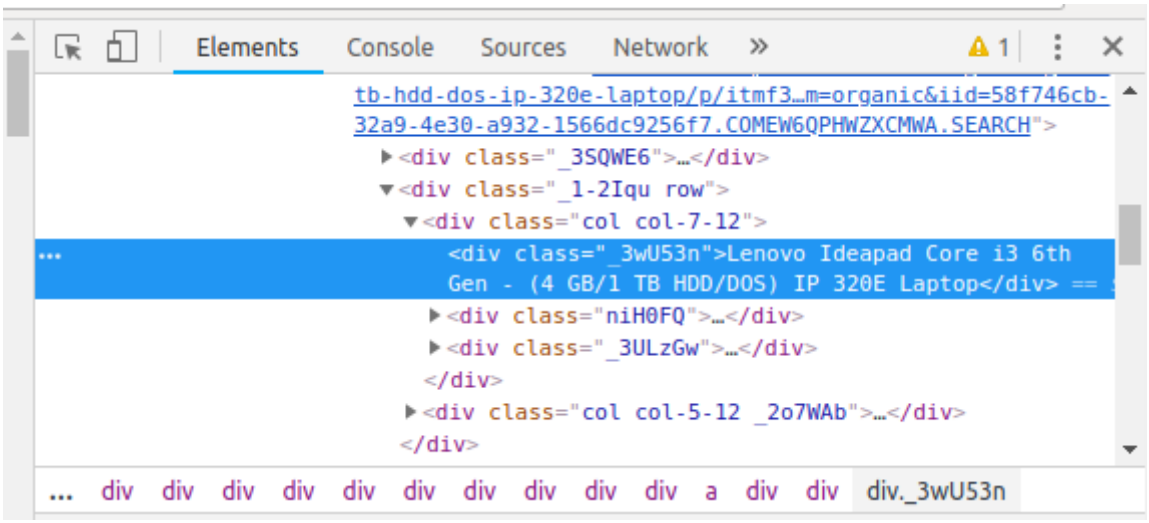


Figure J.1 : *Boîte d'inspection du navigateur Google Chrome*

Étape 3 : Trouvez les données que vous voulez extraire

Nous voulons extraire les données suivantes qui sont respectivement imbriqués dans la balise "div" :

- Capital_social
- Secteur_activite
- Effectif_moyen
- Chiffre_affaires
- Taille
- Forme
- Duree_activite

Etape 4 : Écrire le code

```
gedit web-s.py

from selenium import webdriver

from BeautifulSoup import BeautifulSoup

import pandas as pd

#Pour configurer le pilote web afin d'utiliser le navigateur Chrome, nous devon
s définir le chemin d'accès à Chrome (ouvrir un fenêtre).


Driver1 = webdriver.Chrome("/usr/lib/chromium-browser/chromedriver")

Driver2 = webdriver.Chrome("/usr/lib/chromium-browser/chromedriver")

Driver3 = webdriver.Chrome("/usr/lib/chromium-browser/chromedriver")

Driver4 = webdriver.Chrome("/usr/lib/chromium-browser/chromedriver")


Capital_social = [] #Liste des capitaux sociaux à stocker

Secteur_activite = [] #Liste des Secteurs d'activités à Stocker

Effectif_moyen = [] #Liste des Effectifs moyens à Stocker

Chiffre_affaires = [] #Liste des Chiffres d'affaires à Stocker

Taille = [] #Liste des Tailles des entreprises à Stocker

Forme = [] #Liste des Formes juridiques à Stocker

Durée = [] #Liste Les durées d'activités des entreprises à Stocker


Driver1.get("https://www.societe.com/q")

Driver2.get("http://entreprises.lefigaro.fr/q")

Driver3.get("https://www.zoominfo.com/q")

Driver4.get("https://www.bloomberg.com/q")


Content1 = Driver1.page_source
```

```
Content2 = Driver2.page_source
Content3 = Driver3.page_source
Content4 = Driver4.page_source

For i in range (1,4):
soup = BeautifulSoup(Contenti)
for a in soup.findAll('a',href=True, attrs={'class':'_31qSD5'}):

Capital_social=a.find('div', attrs={'class':'_3wU53n'})
Secteur_activite=a.find('div', attrs={'class':'_1vC4OE _2rQ-NK'})
Effectif_moyen=a.find('div', attrs={'class':'hGSR34 _2beYZw'})
Chiffre_affaires=a.find('div', attrs={'class':'YGSR34 _4beHlw'})
Taille=a.find('div', attrs={'class':'MGSR32 _2biKlw'})
Forme=a.find('div', attrs={'class':'PGSR16 _6beHlw'})
Durée=a.find('div', attrs={'class':'VGSR84 _7beHlw'})

Capital_social.append(cap.text)
Secteur_activite.append(secteur.text)
Effectif_moyen.append(effectif.text)
Chiffre_affaires.append(ca.text)
Taille.append(taille.text)
Forme.append(forme.text)
Durée.append(durée.text)
```

Étape 5 : Exécution du code et extraction des données

Exécuter le code et extraire les données en utilisant la commande ci-dessous :

```
python web-s.py
```

Étape 6 : Stocker les données dans un format requis

```
df = pd.DataFrame({'Capital social':Capital_social, 'Secteur activite' :
Secteur_activite, 'Effectif moyen' : Effectif_moyen, 'Chiffre affaires' :
Chiffre_affaires, 'Taille de l'entreprise':Taille, 'Forme juridique de
l'entreprise':Forme, 'durée d'activité de l'entreprise':Durée,})
df.to_csv('base_de_doonnées.csv', index=False, encoding='utf-8')
```

Annexe K : Extrait de la base de données épurée

ID	Products Types	Type_Transaction	Reseller	Users	Ongoing contract	Nb jours sous	Region	Tenant	Month 1	Month 2	Month 3	Moyenne revenu	somme des ca	eARR estimée	Nb de mois payé	Nb de downsells	Nb de upsell	Added MRR due to Upsell	Lost MRR due to Downsell	Chiffre d'affaires année N-1	Taille	Capital social	Secteur	Effectif moyen année	Nombre de mois en	Forme juri	churn
1	Workplace	Direct	1	54664	38	6519	EMEA	1	6844,483	7498,98	4742,234	5970,819	197037,026	82133,8	104	13	13	4384,616147	2670,877353	78997860,55	GE	116238,9282	Tertiaire	383,2743	757,1238	SPA	no
2	Portal	Direct		3833	0	6519	EMEA	1	1806,416	1806,42	1806,416	1806,416	34321,8997	68100,3	60	0	0	0	0	14836485,47	ETI	647973908,3	Secondaire	69,115038	848,23	SASU	yes
3	Portal + Lumdrive	Indirect		1885	0	6230	EMEA	1	1005,31	1005,31	1005,31	1005,31	21111,5026	37899,3	66	0	0	0	0	1007342260	ETI	27600773,28	Secondaire	4335,3979	1196,947	SASU	yes
4	Portal	Indirect	4	3142	38	6230	EMEA	2	1884,956	1884,96	1884,956	1884,956	62203,5345	22619,5	104	0	0	0	0	804420192,8	ETI	12871482,09	Primaire	2217,9644	1225,221	SA	no
5	Workplace	Direct		####	38	6230	EMEA	2	121736,7	121737	121736,7	100576,9	3319037,14	1460841	104	13	13	157507,1511	54619,46808	1648956010	GE	8910601958	Tertiaire	3399,2033	923,6282	SA	no
6	Support	Indirect	6	36424	0	6132	EMEA	2	1047,198	1047,2	1047,198	1047,198	12566,3706	39478,4	38	0	0	0	0	2307870512	ETI	61809264,66	Secondaire	2651,5042	1693,318	SPA	yes
7	Portal	Indirect	7	1539	38	6044	EMEA	2	207,8687	207,869	207,8687	205,4491	6779,81875	2494,42	104	0	3	14,39896634	0	55992605,86	PME	3141592,654	Secondaire	816,81409	1809,557	SASU	no
8	Portal	Direct		6283	75	5853	EMEA	1	6896,319	6896,32	6896,319	6694,036	220903,182	82755,8	104	0	3	351,3347795	0	1553513797	ETI	36709334,23	Tertiaire	8278,0966	1577,08	SPA	no
9	Support	Direct		36424	0	5774	EMEA	1	2217,964	2217,96	2217,964	10849,55	195291,871	83615,3	57	3	3	19421,84938	19421,32578	1648956010	GE	8910601958	Tertiaire	3399,2033	923,6282	SA	yes
10	Workplace	Direct		314	38	5711	EMEA	2	877,0279	877,028	877,0279	866,7146	28601,5831	10524,3	104	0	3	85,08480114	0	3697026,235	PME	1330876,037	Tertiaire	163,36282	556,0619	SPA	no
11	Portal	Direct	11	4398	38	5281	EMEA	1	2932,153	2932,15	2932,153	2932,153	96761,0537	35185,8	104	0	0	0	104719791	36428651,77	ETI	604884385,8	Tertiaire	122,52211	2318,495	SA	no
12	Portal	Direct	12	18850	38	5275	EMEA	2	5505,641	5505,64	5505,641	5227,087	172493,857	66067,7	104	0	3	437,7285764	0	76340701,48	ETI	36658427,86	Tertiaire	691,15038	1617,92	SPA	no
13	Portal	Indirect		15708	0	5275	EMEA	1	5026,548	5026,55	5026,548	7539,822	180955,737	189496	75	3	0	0	5026,548246	8782063082	GE	146618129,1	Tertiaire	291439,27	731,9911	SPA	yes
14	Portal	Indirect		1926	0	5180	EMEA	1	1604,83	1604,83	1604,83	1604,83	16048,3025	60500,7	31	0	0	0	0	0		0	Tertiaire	4241,1501	0	Etrange	yes
15	Portal	Indirect		1194	113	5180	EMEA	1	1053,219	1053,22	1053,219	978,8441	32301,8557	12638,6	104	0	9	239,5464398	0	248876341,7	ETI	30825926,01	Secondaire	1837,8317	2456,725	SPA	no
16	Portal	Direct	16	1037	38	5180	EMEA	2	775,9734	775,973	691,1504	696,2912	22977,6087	28187,6	104	0	3	84,82300165	0	283314480,4	ETI	18849555,92	Primaire	1746,7255	2434,734	SPA	no
17	Support	Indirect	17	36424	38	5180	EMEA	2	157,0796	157,08	157,0796	157,0796	5183,62788	5921,76	104	0	0	0	0	283314480,4	ETI	18849555,92	Primaire	1746,7255	2434,734	SPA	no
18	Portal	Indirect	18	1037	0	5152	EMEA	2	392,6991	392,699	392,6991	392,6991	9032,07888	11196,6	72	0	0	0	0	283314480,4	ETI	18849555,92		1746,7255	0	SPA	yes
19	Portal	Direct	19	16965	113	5083	EMEA	2	18335,91	18335,9	18335,91	13087,45	431885,738	220031	104	3	6	6227,683837	47,1238898	1686101888	GE	32803112,48	Secondaire	12707,742	1514,248	SASU	no
20	Portal	Direct		34558	113	5083	EMEA	1	13120,08	13120,1	13120,08	9457,979	312113,303	157441	104	0	3	4028,30718	0	112939941,7	ETI	737664804,6	Secondaire	819,95568	2466,15	SA	no
21	Portal	Indirect		440	38	5083	EMEA	2	471,2389	471,239	471,2389	246,9244	8148,50595	5654,87	104	0	6	282,21974	0	13008078,54	ETI	168043,791	Tertiaire	75,398224	637,7433	SPA	no
22	Portal	Direct		99274	38	5083	EMEA	2	10341,08	10341,1	10341,08	9436,678	311410,372	124093	104	0	3	2487,094185	0	7048685879	GE	691299622	Primaire	30869,289	2425,31	SA	no
23	Portal	Direct	23	18850	38	4986	EMEA	2	10210,18	10210,2	10210,18	11138,37	367566,34	122522	104	3	0	0	3403,392041	128750949,2	ETI	3352,079361	Tertiaire	455,53093	1916,372	SA	no

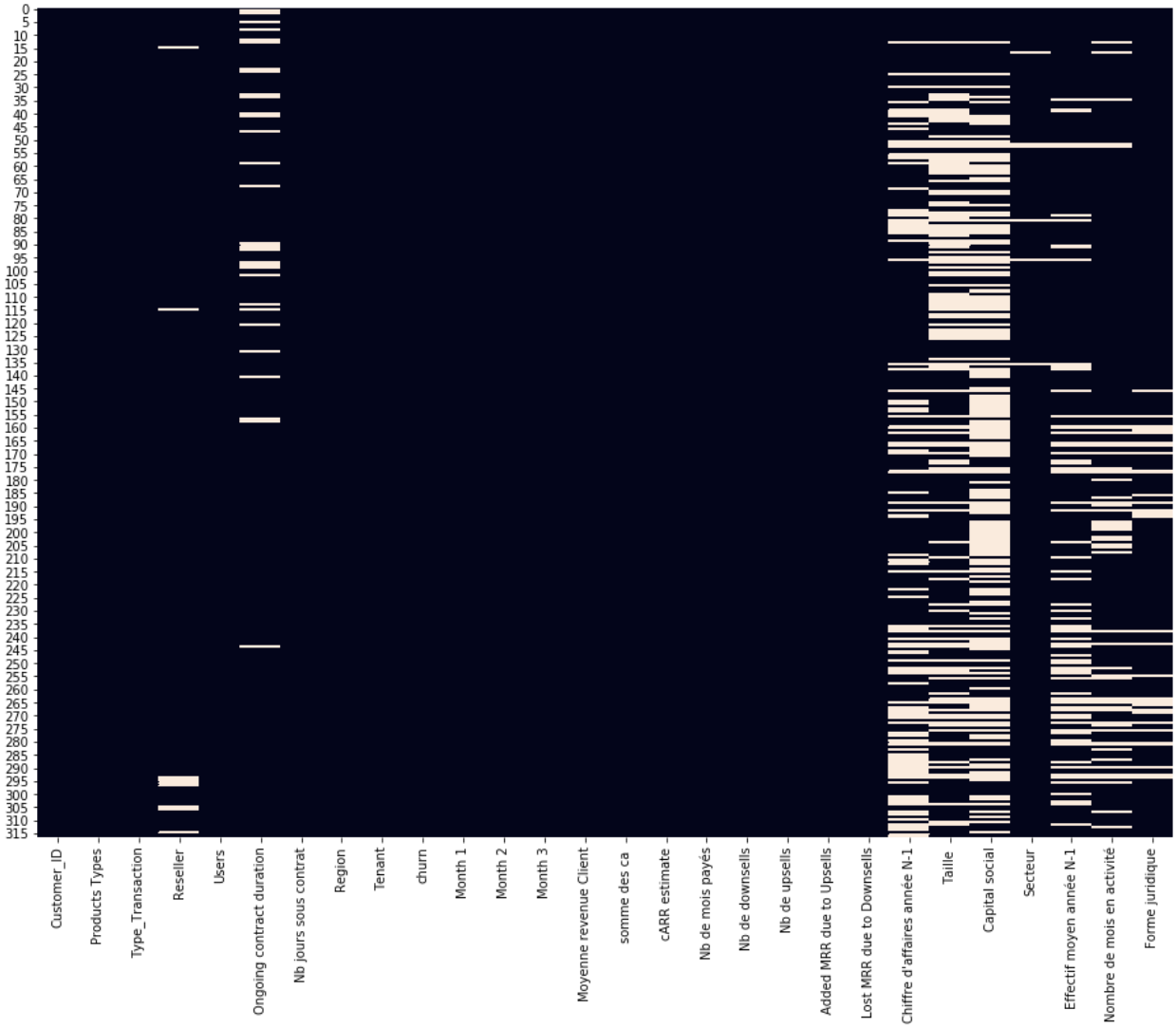
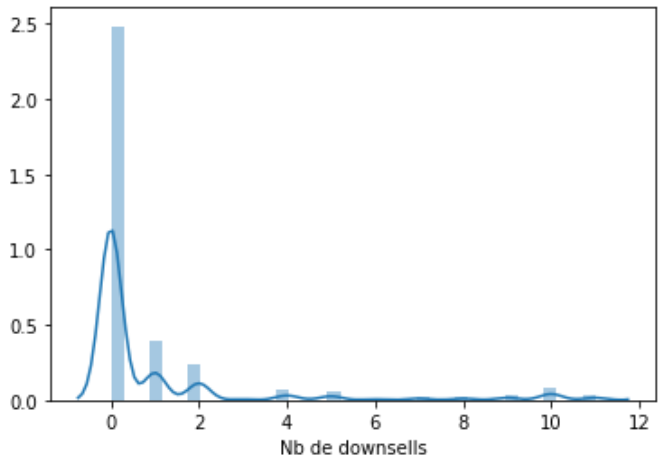
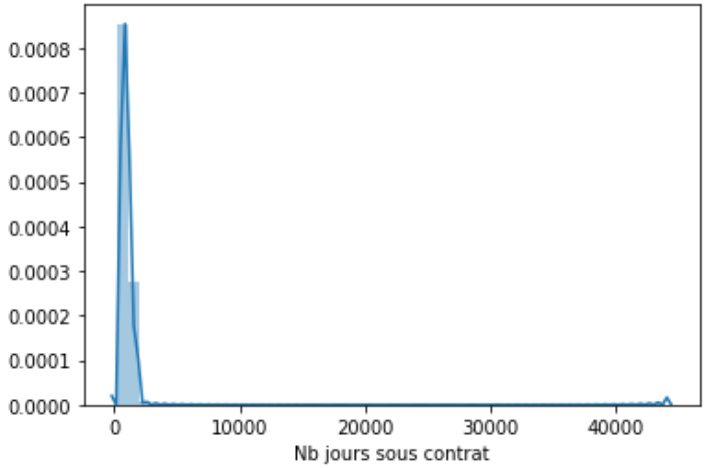


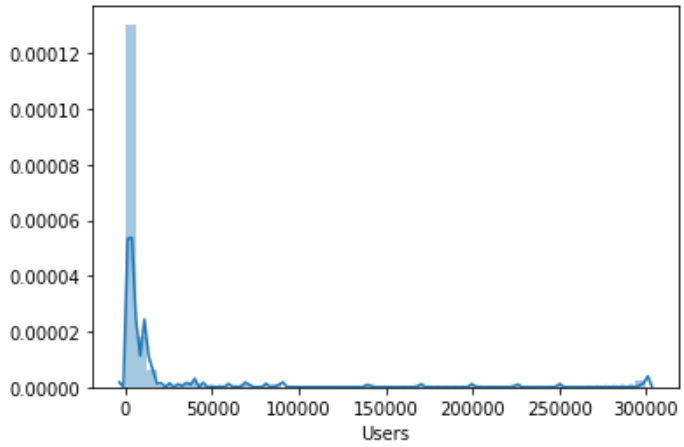
Figure K.1 : Diagramme des valeurs manquantes dans notre dataset épuré

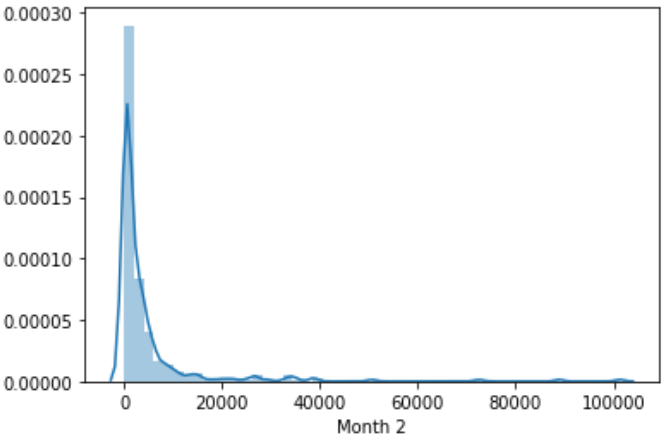
Annexe L : les distributions variables numériques

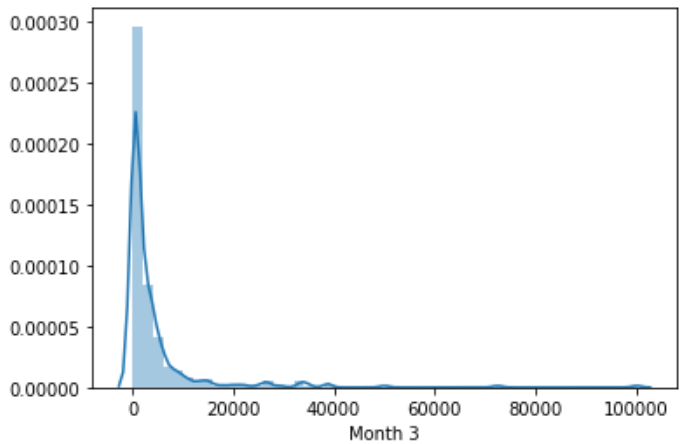
Cette annexe permet de présenter les distributions des différentes variables numériques de notre base de données.

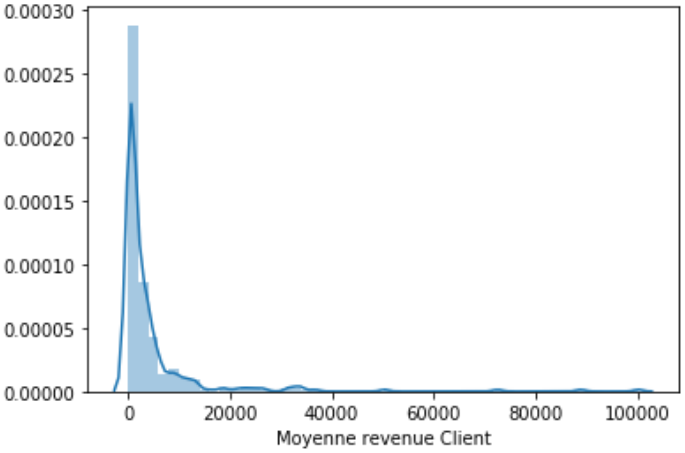


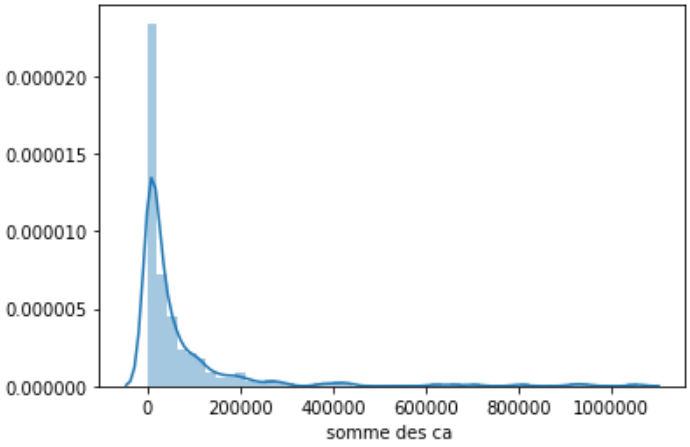


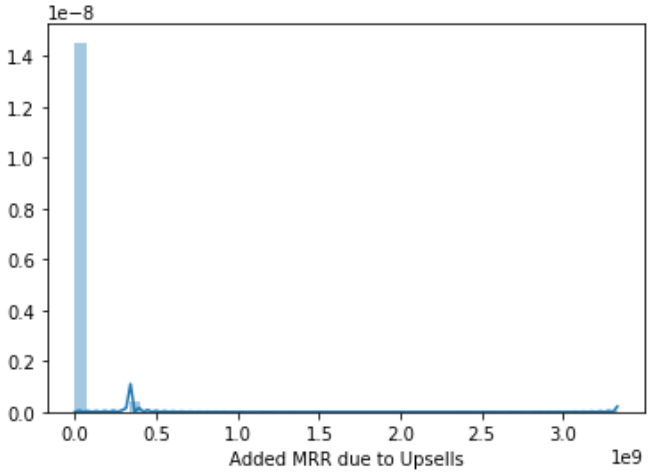


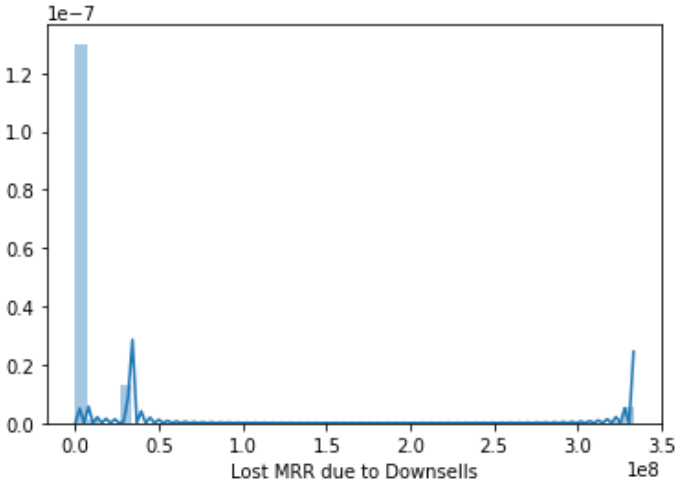


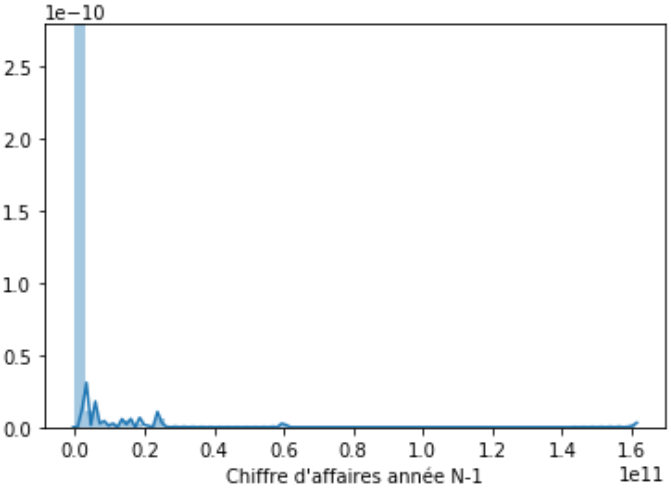


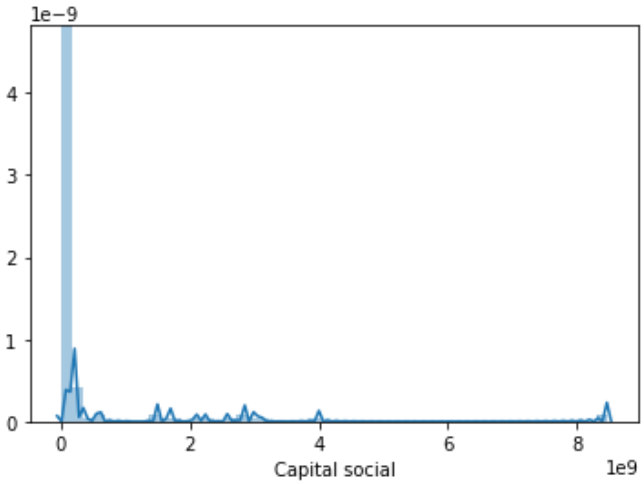


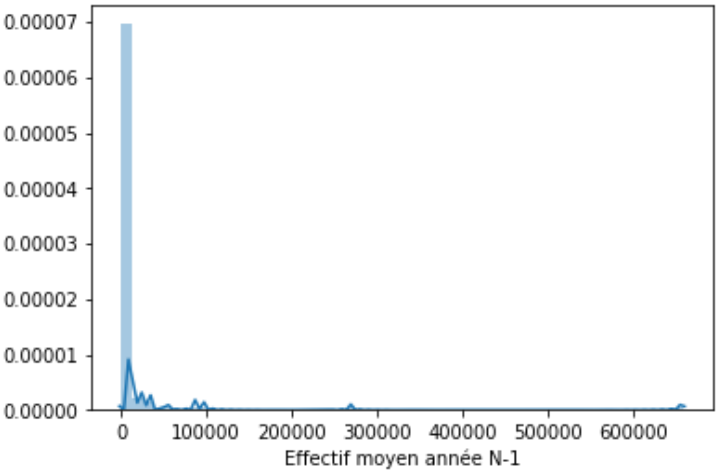

0.000020
0.000015
0.000010
0.000005
0.000000
0
200000
400000
600000
800000
1000000
somme des ca


1e−8
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.5
1.0
1.5
2.0
2.5
3.0
Added MRR due to Upsells
1e9

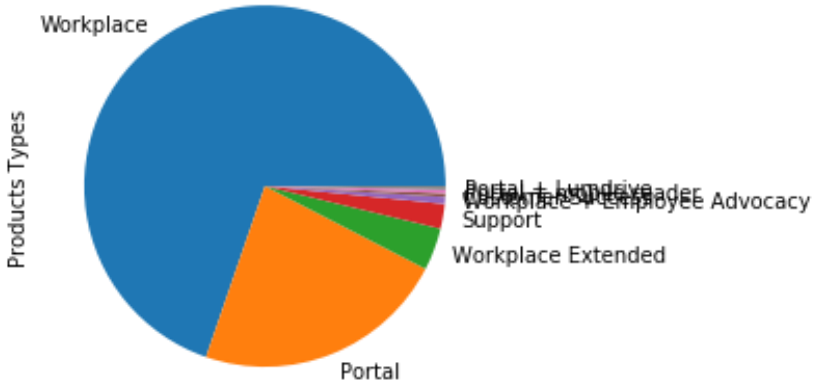

1e−7
1.2
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
Lost MRR due to Downsells
1e8

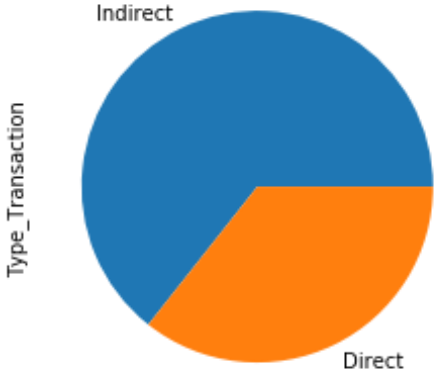

1e−10
2.5
2.0
1.5
1.0
0.5
0.0
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
1.6
Chiffre d'affaires année N-1
1e11

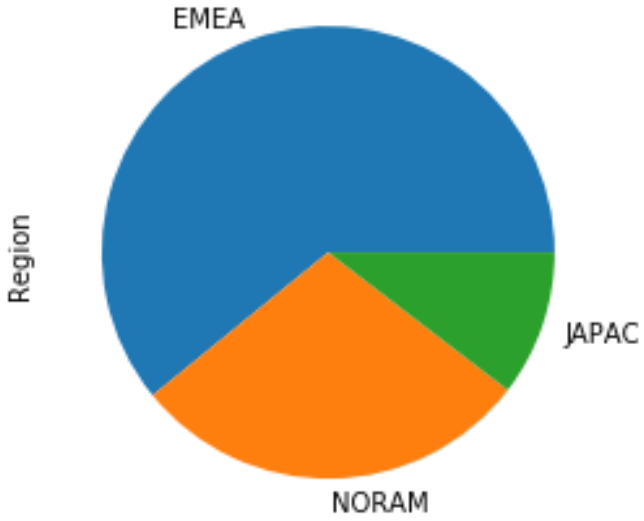

1e−9
4
3
2
1
0
0
2
4
6
8
Capital social
1e9

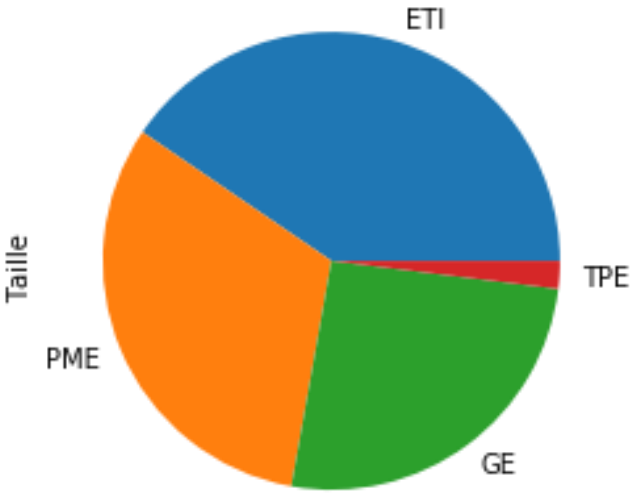

0.00007
0.00006
0.00005
0.00004
0.00003
0.00002
0.00001
0.00000
0
100000
200000
300000
400000
500000
600000
Effectif moyen année N-1

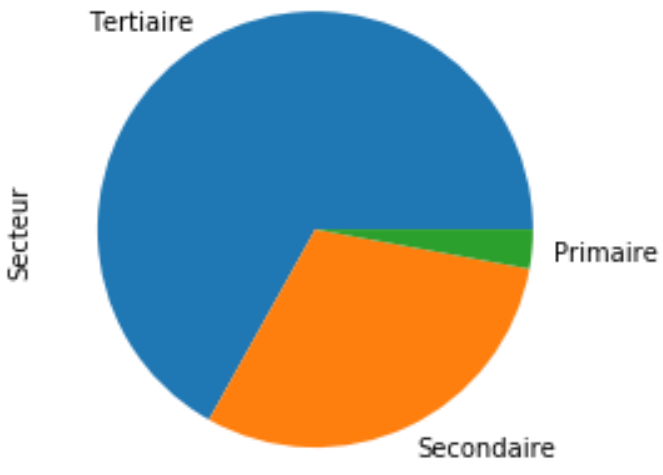
Annexe M : Diagrammes en camembert des variables catégorielles

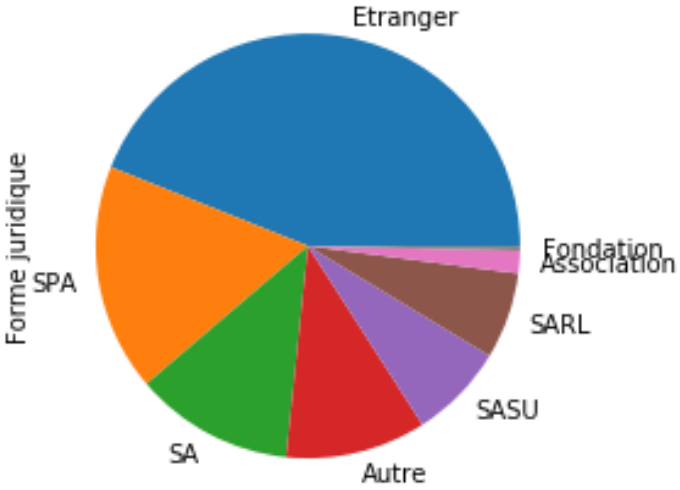












Annexe N : Coefficient de corrélation de Pearson

En statistique, le coefficient de corrélation de Pearson (PCC Pearson Correlation Coefficient), également appelé r de Pearson, le coefficient de corrélation produit-moment de Pearson (PPMCC Pearson Product-Moment Correlation Coefficient) ou la corrélation bivariée, est une statistique qui mesure la corrélation linéaire entre deux variables X et Y. Elle a une valeur comprise entre +1 et -1. Une valeur de +1 est une corrélation linéaire positive totale, 0 n'est pas une corrélation linéaire et -1 est une corrélation linéaire négative totale.

$$r = \frac{\mathrm{Cov}(X,Y)}{\sigma_X \sigma_Y}$$

Où :

$\mathrm{Cov}(X,Y)$ Désigne la covariance des variables X et Y,

Et $\boldsymbol{\sigma_X, \sigma_Y}$ désignent leurs écarts types respectifs.

Annexe O : Code t-test

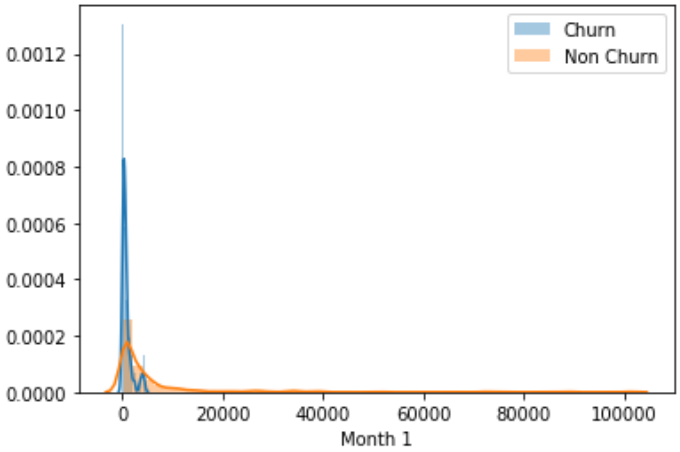
Le code suivant nous permet d'effectuer un test de Student pour visualiser les variables qui ont une distribution significativement différente relativement à chacune des classes.

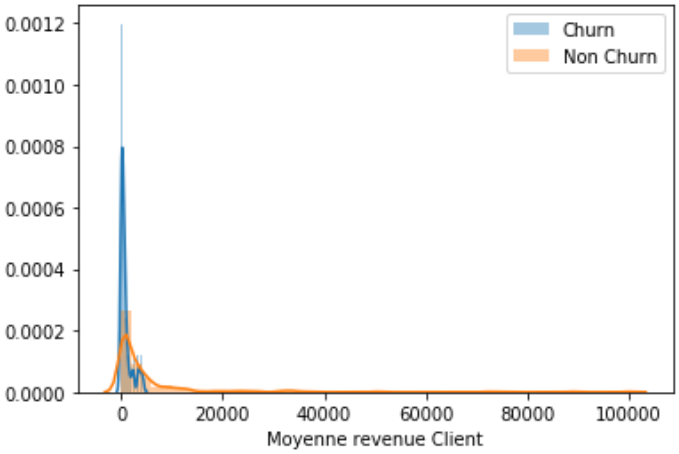
```
from scipy.stats import ttest_ind
balanced_nc = non_churn_df.sample(churn_df.shape[0])
def t_test(col):
    alpha = 0.05
    stat,p = ttest_ind(balanced_nc[col].dropna(),churn_df[col].dropna())
    if p<alpha :
        print('H0 rejetée')
    else:
        return 0
for col in colu :
    print(f'{col:-<70}{t_test(col)}')
```

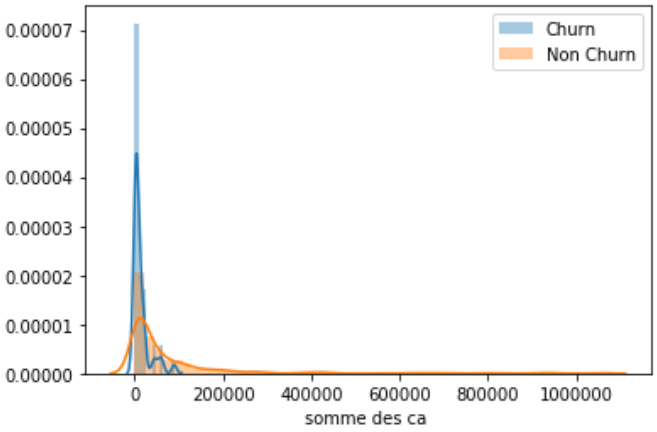
Figure O.1 : Visualisation du code permettant d'effectuer le t-test

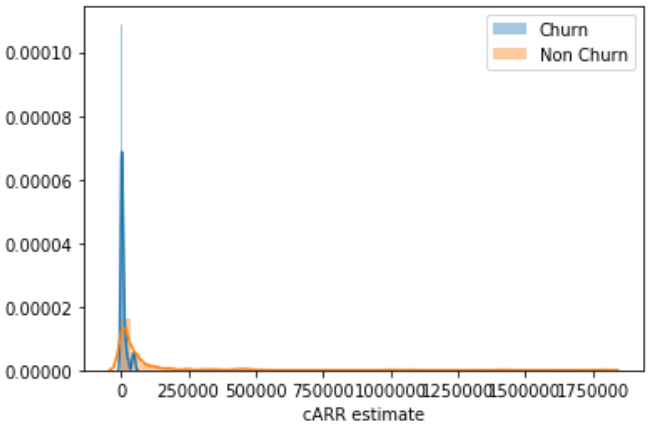
Annexe P : La distribution des variables numériques

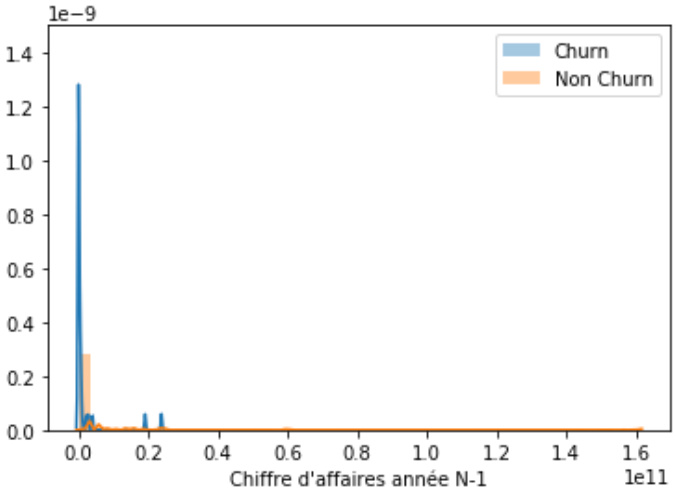
Les graphiques suivants permettent de représenter les distributions des différentes variables numériques selon les classes « Churn et « Non Churn » :

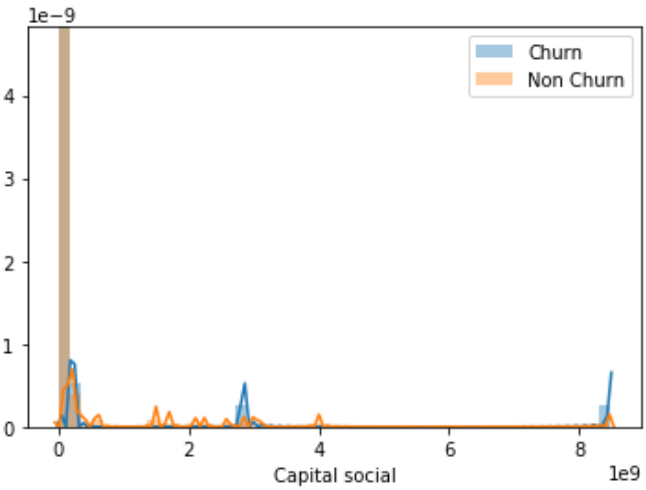


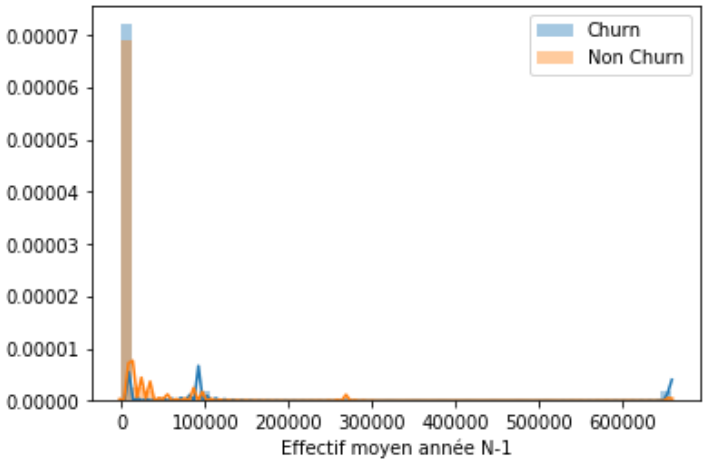


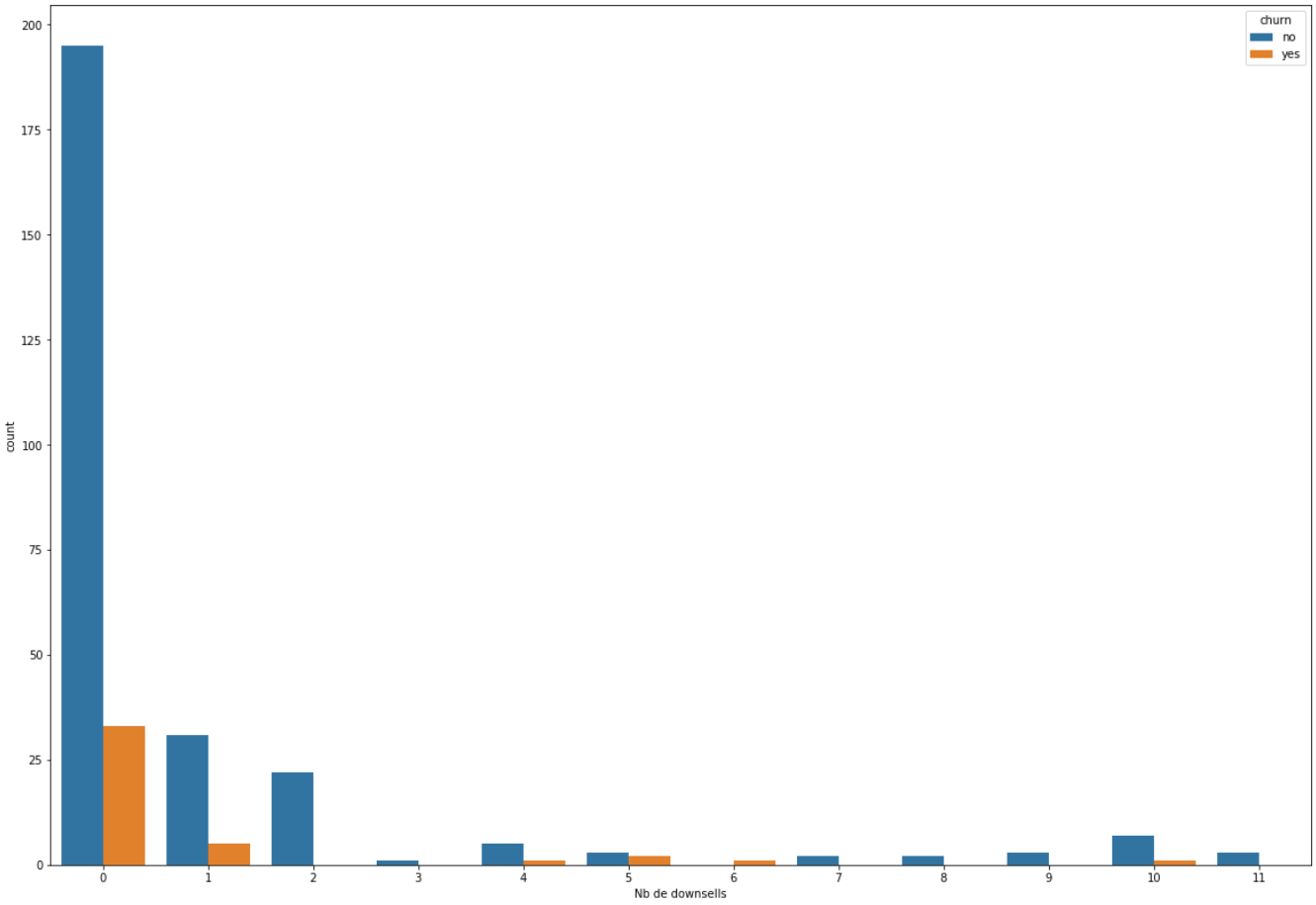


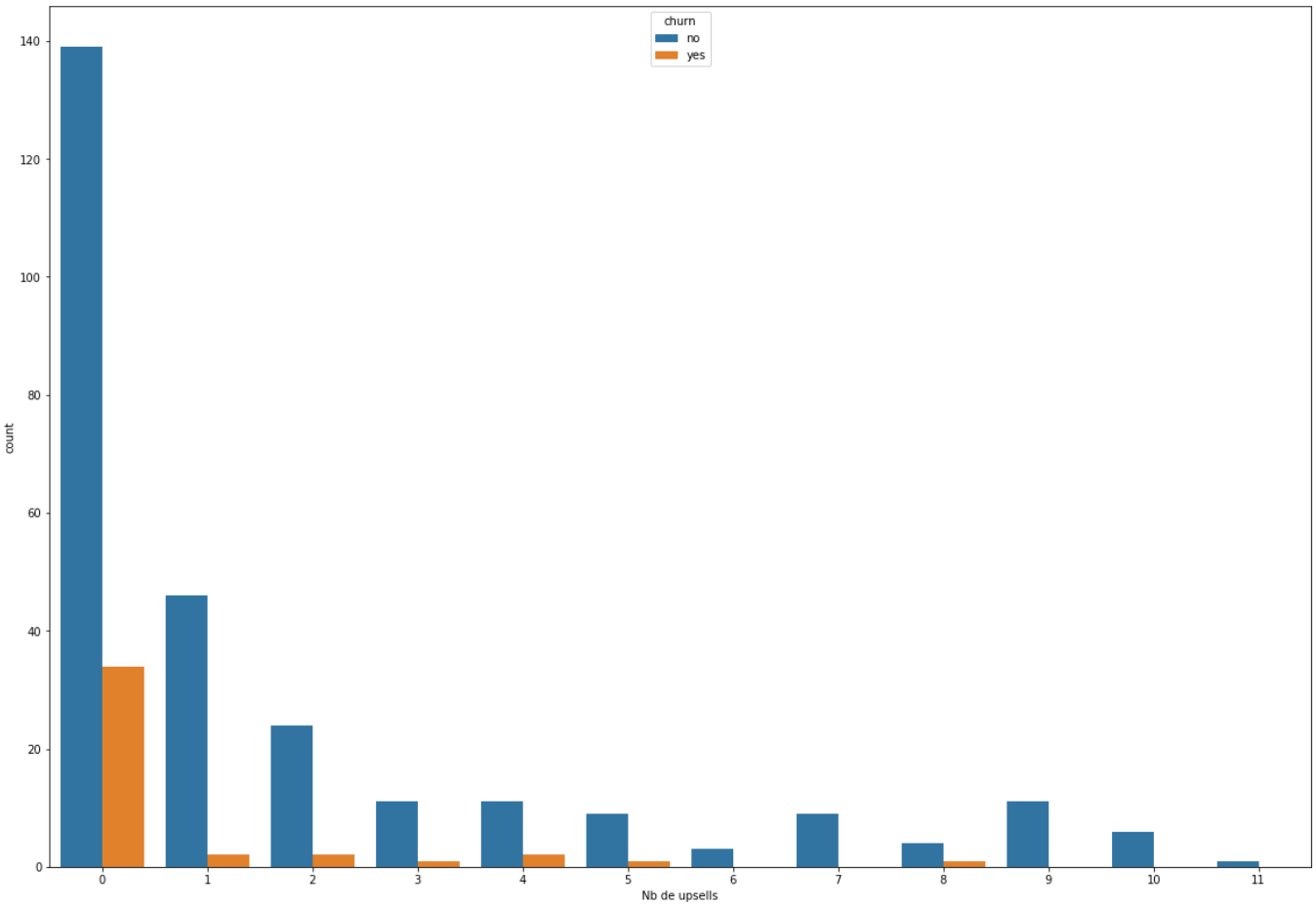


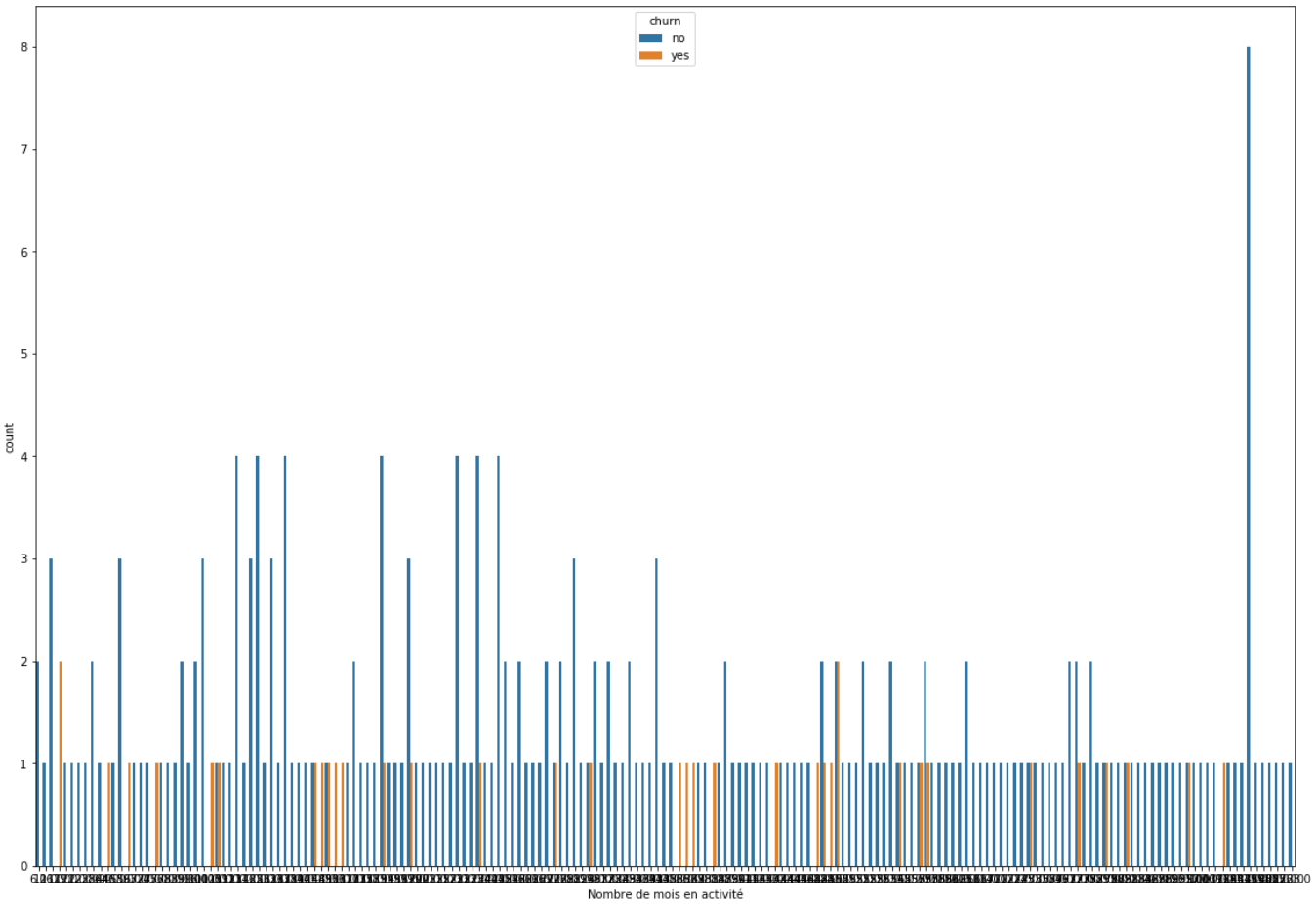


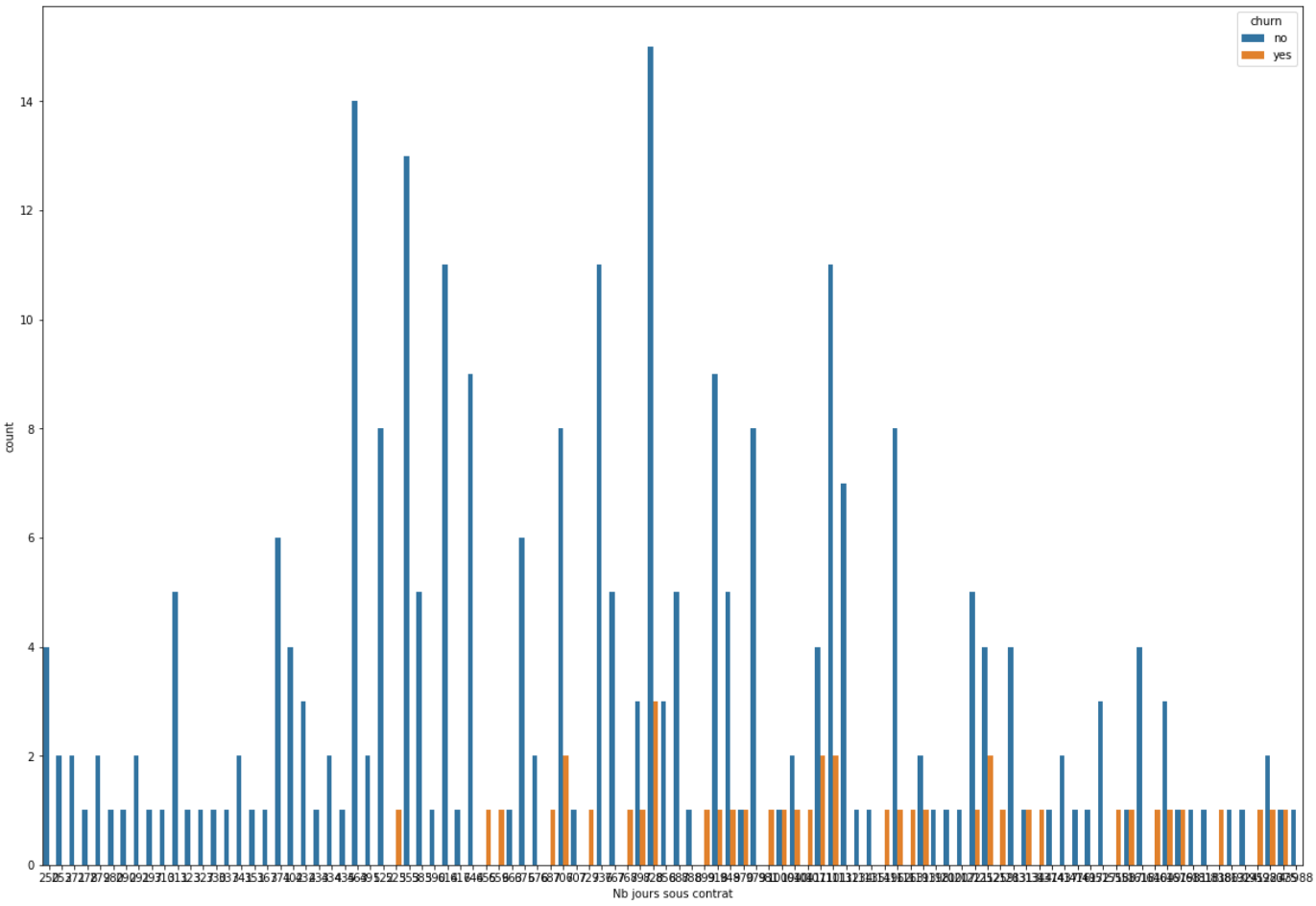



Churn
Non Churn
0.00007
0.00006
0.00005
0.00004
0.00003
0.00002
0.00001
0.00000
0
100000
200000
300000
400000
500000
600000
Effectif moyen année N-1


churn
no
yes
count
Nb de downsells

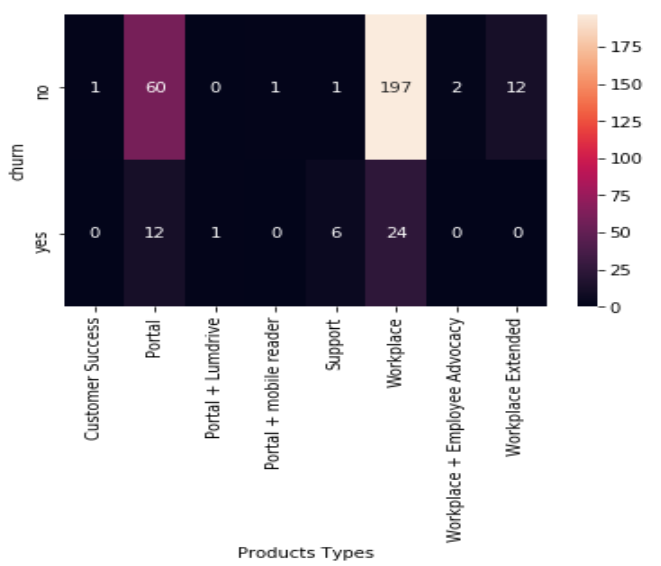

churn
no
yes
count
Nb de upsells

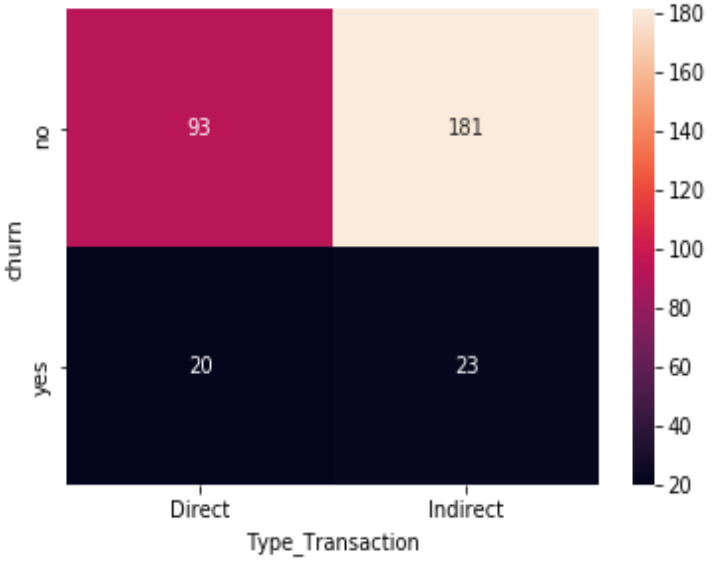

churn
no
yes
count
Nombre de mois en activité

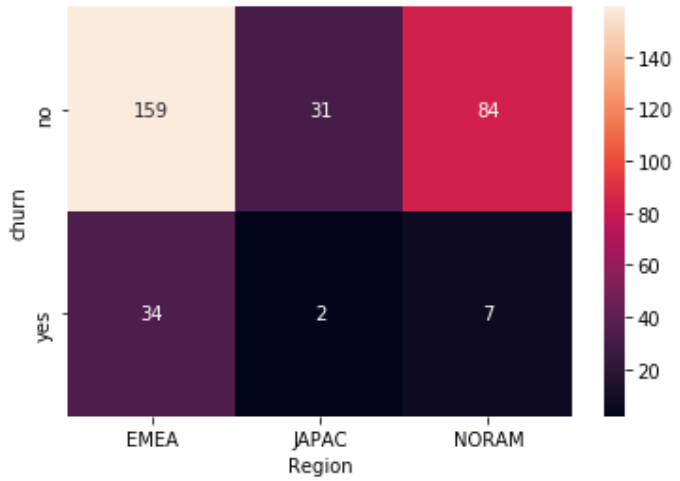

churn
no
yes
count
Nb jours sous contrat

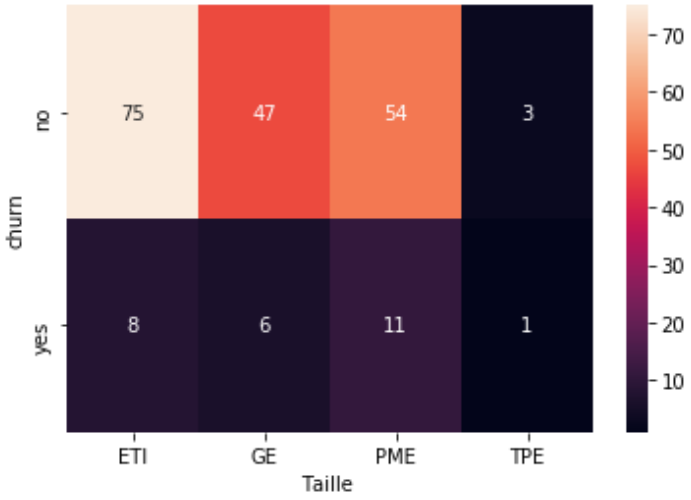
Annexe Q : la distribution des variables catégorielles

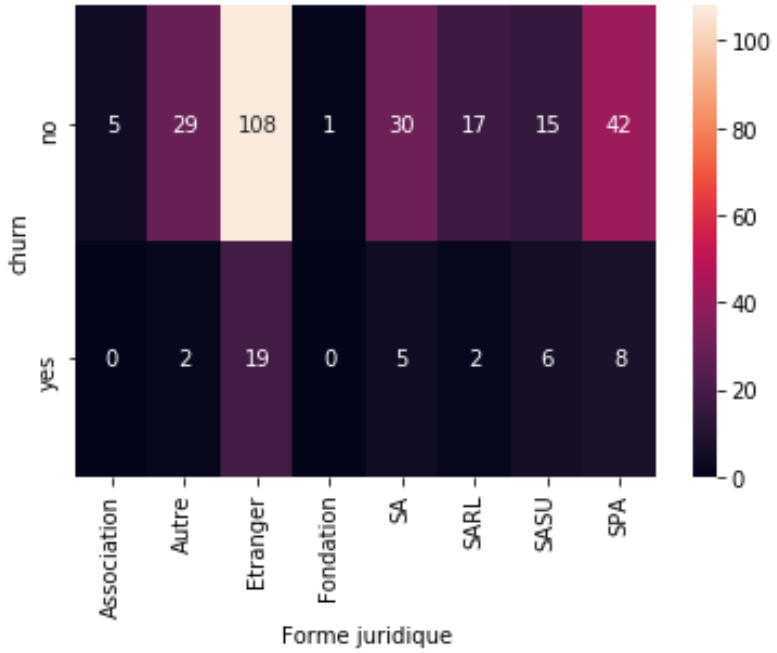
Les graphiques suivantes permettent de visualiser la distribution des variables catégorielles selon les classes « Churn » et « Non Churn » :










churn
no
yes
5
29
108
1
30
17
15
42
0
2
19
0
5
2
6
8
Association
Autre
Etranger
Fondation
SA
SARL
SASU
SPA
Forme juridique
100
80
60
40
20
0

Annexe R : Algorithme d'imputation des valeurs manquantes

```
from sklearn.impute import MissingIndicator
from sklearn.impute import SimpleImputer
def imputation (X_train, X_test)
    ip = SimpleImputer(strategy = "median")
    mis = MissingIndicator()
    impu = SimpleImputer(strategy = "most_frequent")
    X_train['Effectif moyen année N-1'] = ip.fit_transform(X_train[['Effectif moyen année N-1']])
    X_test['Effectif moyen année N-1'] = ip.transform(X_test[['Effectif moyen année N-1']])
    X_train["Chiffre d'affaires année N-1"] = ip.fit_transform(X_train[["Chiffre d'affaires année N-1"]])
    X_test["Chiffre d'affaires année N-1"] = ip.transform(X_test[["Chiffre d'affaires année N-1"]])
    X_train["Nombre de mois en activité"] = ip.fit_transform(X_train[["Nombre de mois en activité"]])
    X_test["Nombre de mois en activité"] = ip.transform(X_test[["Nombre de mois en activité"]])
    X_train['Capital social'] = mis.fit_transform(X_train[['Capital social']])
    X_test['Capital social'] = mis.transform(X_test[['Capital social']])
    X_train['Forme juridique'] = impu.fit_transform(X_train[['Forme juridique']])
    X_train['Taille'] = impu.fit_transform(X_train[['Taille']])
    X_train['Secteur'] = impu.fit_transform(X_train[['Secteur']])
    X_test['Forme juridique'] = impu.fit_transform(X_test[['Forme juridique']])
    X_test['Taille'] = impu.fit_transform(X_test[['Taille']])
    X_test['Secteur'] = impu.fit_transform(X_test[['Secteur']])
    return X_train,X_test
```

Figure R.1 : Algorithme d'imputation des valeurs manquantes

Cette fonction applique différentes stratégies d'imputation sur notre base de données. Nous avons tout d'abord le premier objet « Imputer » appelé « ip » qui est de la classe « Simple Imputer » avec une stratégie d'imputation basée sur la médiane de la variable passée en paramètre. Cet objet nous permet de remplacer les valeurs manquantes dans les colonnes « Effectif moyen année N-1 », « Chiffre d'affaires année N-1 » ainsi que « Nombre de mois en activité » par les médianes respectives de leurs distributions. Ce choix est justifié par le fait de la présence d'Outliers (Valeurs aberrantes) dans chacune d'entre elles, et donc que la moyenne est biaisée par ces dernières et donc trop élevée. Nous utilisons d'abord la méthode « fit_transform » de l'objet instancié pour calculer la médiane de l'ensemble d'entrainement « X_train » puis nous remplaçons les valeurs manquantes du jeu de test « X_test » par cette valeur, et ce afin de ne pas créer une fuite d'information du « X_test » vers le « X_train » en calculant une médiane sur l'ensemble de ces colonnes concaténées, ce qui aurait eu pour conséquence d'entrainer un ensemble d'entrainement avec des valeurs issues de l'ensemble de test, chose qui pourrait être préjudiciable au modèle implémenté. Cette stratégie a été appliquée également dans les autres imputers.

Nous avons également en marge de cette fonction instancié un objet « mis » de la classe « Missing Indicator » que nous avons utilisé pour transformer la colonne « Capital social » car cette dernière contient un nombre trop important de valeurs manquantes pour n'être remplacées que par la médiane de la distribution. Le résultat de cette transformation est d'avoir une colonne booléenne contenant « Vrai » si une valeur de « Capital social » et « Faux » sinon.

Le dernier « imputer » utilisé est l'objet « impu » de la classe « Simple Imputer » avec cette fois une stratégie « valeur la plus fréquente » spécifique aux valeurs catégorielles et qui est appliquée aux variables « Taille », « Secteur » et « Forme juridique » ayant été soumises à un scrapping de notre part.

Annexe S : Algorithme d'encodage des variables catégorielles

```
from sklearn.preprocessing import LabelEncoder
def encodage (X_train, X_test) :
    lb = LabelEncoder()
    col_bin = ['Type_Transaction','Tenant']
    for col in col_bin:
    X_train.loc[:,col] = lb.fit_transform(X_train[col])
    X_test.loc[:,col] = lb.fit_transform(X_test[col])
    xte = X_test[['Products Types','Region','Secteur','Forme juridique','Taille']]
    xte = pd.get_dummies(xt)
    X_train = pd.concat([X_train,xte],axis=1)
    xtr = X_train[['Products Types','Region','Secteur','Forme juridique','Taille']]
    xtr = pd.get_dummies(xtr)
    X_test = pd.concat([X_test,xtr],axis=1)
    X_train = X_train.drop('Products Types',axis=1)
    X_train = X_train.drop('Region',axis=1)
    X_train = X_train.drop('Secteur',axis=1)
    X_train = X_train.drop('Forme juridique',axis=1)
    X_train = X_train.drop('Taille',axis=1)
    X_test = X_test.drop('Products Types',axis=1)
    X_test = X_test.drop('Region',axis=1)
    X_test = X_test.drop('Secteur',axis=1)
    X_test = X_test.drop('Forme juridique',axis=1)
    X_test = X_test.drop('Taille',axis=1)
    X_test['Products Types_Portal + Lumdrive'] = 0
    X_test['Products Types_Customer Success'] = 0
    X_test['Products Types_Portal + mobile reader'] = 0
    X_test['Forme juridique_Association'] = 0
    X_train['Forme juridique_Fondation'] =0
    return X_train, X_test
```

Figure S.1 : Algorithme d'encodage des variables catégorielles

Dans cette fonction nous avons tout d'abord instancié un objet « lb » de la classe « LabelEncoder » qui est un encodeur binaire. Ce dernier transformera les colonnes sélectionnées en affectant des valeurs binaires à chacune des variables sélectionnées et possédant seulement 2 catégories. Quant aux autres variables qui possèdent plus de 2 catégories, elles subissent un encodage de type « One Hot Enconding » dont la procédure consiste en la création de nouvelles colonnes contenant des valeurs binaires pour chaque catégorie d'une variable concernée. C'est ce qui a été réalisée en marge de notre fonction où nous avons tout d'abord recueilli dans un « DataFrame » ces nouvelles colonnes grâce à la méthode « get_dummies » de « Pandas » puis nous avons effectué une concaténation de ces nouveaux « DataFrames » et ceux que l'on possèdé « X_train » et « X_test » tout en éliminant les variables de base concernées grâce à un « drop ».

Annexe T : Cross Validation

Validation croisée ou « Cross validation » :

Afin d'estimer l'erreur de test associée à l'ajustement d'un modèle particulier sur un ensemble d'observations, il est très courant de procéder à ce que l'on appelle une méthode de validation croisée. Dans cette méthode, l'ensemble de données est divisé de manière aléatoire en deux ensembles : un ensemble d'entraînement et un ensemble de test/validation. Le modèle est ensuite entraîné sur l'ensemble d'entraînement et évalué sur l'ensemble de test. Cette méthode n'est utilisée que lorsqu'il y a un modèle à évaluer et pas d'hyperparamètres à régler. Si l'on veut comparer plusieurs modèles et régler leurs hyperparamètres, une autre forme de validation croisée est utilisée. Celui-ci consiste à diviser les données non pas en deux mais en trois ensembles distincts. L'ensemble d'entraînement est divisé en ensemble d'entraînement et de validation, de sorte que l'ensemble original de données soit divisé en trois catégories : entraînement, validation et test, comme le montre la figure suivante :

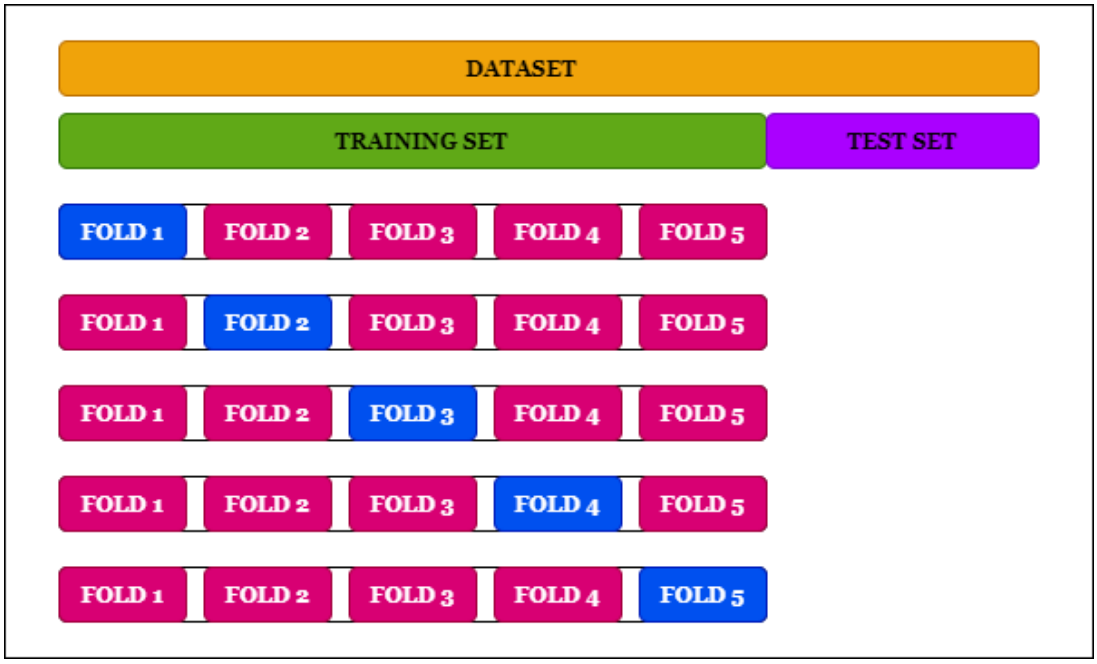


Figure T.1 : Méthode de Cross Valisation

Cette approche consiste à diviser l'ensemble des observations en k groupes, ou plis, de taille à peu près égale.

Le premier pli sert d'ensemble de validation, et la méthode est adaptée aux k-1 groupes restants. L'erreur quadratique moyenne est ensuite calculée sur la base des observations de l'ensemble de validation. La procédure globale est calculée k fois, avec à chaque fois un sous-ensemble différent d'observations jouant le rôle de jeu de validation. En conséquence, l'erreur de test est estimée k fois et en faisant la moyenne de ces valeurs, on obtient l'estimation de validation croisée (CV) avec k plis :

$$CV_{(k)} = \frac{1}{k}\sum_{i=1}^{k} MSE_i$$

L'avantage de la validation croisée est qu'elle peut être appliquée à presque toutes les méthodes d'apprentissage statistiques. Pour les modèles à forte intensité de calcul, on effectue généralement cette méthode utilisant k = 5 ou k = 10, ce qui nécessite respectivement d'adapter l'apprentissage seulement cinq à dix fois.

```
from sklearn.ensemble import AdaBoostClassifier, BaggingClassifier, GradientBoostingClassifier
from sklearn.svm import SVC
from sklearn.linear_model import LogisticRegression
from sklearn.neighbors import KNeighborsClassifier
from sklearn.preprocessing import StandardScaler
from sklearn.neural_network import MLPClassifier
from sklearn.preprocessing import RobustScaler
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from xgboost.xgb import XGBClassifier
from sklearn.model_selection import GridSearchCV

#Support Vector Machine
SVM = make_pipeline(StandardScaler(), SVC(random_state =0))
Cs = [0.001, 0.01, 0.1, 1, 10]
gammas = [0.001, 0.01, 0.1, 1]
param_grid = {'svc__C': Cs, 'svc__gamma' : gammas}
grid_search = GridSearchCV(SVM, param_grid, cv=4,scoring='f1')
grid_search.fit(X_train, y_train)
grid_search.best_params_
evaluation(grid.best_estimator_)

#KNN
KNN = make_pipeline(RobustScaler(), KNeighborsClassifier(n_neighbors=3))
nei = np.arange(1,20)
param_grid = {'kneighborsclassifier__n_neighbors': nei
              'metric' : ['euclidean','manhattan']}
grid_search = GridSearchCV(KNN, param_grid, cv=4,scoring='f1')
grid_search.fit(X_train, y_train)
evaluation(grid_search.best_estimator_)

#DecisionTree
DecisionTree = DecisionTreeClassifier(random_state=0)
max_depth = np.arange(1,20)
min_samples_split = np.arange(2,20)
min_samples_leaf = [9,10]

hyperDT = dict(max_depth = max_depth,
               min_samples_split = min_samples_split,
              min_samples_leaf = min_samples_leaf)

gridDT = GridSearchCV(DecisionTree, hyperDT,scoring='f1' cv = 3, verbose = 1,
                      n_jobs = -1)
bestDT = gridDT.fit(X_train, y_train)
evaluation(bestDT.best_estimator_)

#Perceptron
mlp = make_pipeline(StandardScaler(),MLPClassifier(max_iter=500))
mlp
parameter_space = {
    'mlpclassifier__hidden_layer_sizes': [(50,50,50), (50,100,50), (100,)],
    'mlpclassifier__activation': ['tanh', 'relu'],
    'mlpclassifier__solver': ['sgd', 'adam','lbfgs'],
    'mlpclassifier__alpha': [0.0001, 0.05],
    'mlpclassifier__learning_rate': ['constant','adaptive'],
}
clf = GridSearchCV(mlp, parameter_space, n_jobs=-1, cv=5, scoring='f1')
```

```
clf.fit(X_train,y_train)

#Logistic Regression
Logistic = make_pipeline(RobustScaler(),LogisticRegression())
penalty = ['l1', 'l2']
C = [0.0001, 0.001, 0.01, 0.1, 1, 10, 100, 1000]
class_weight = [{1:0.5, 0:0.5}, {1:0.4, 0:0.6}, {1:0.6, 0:0.4}, {1:0.7, 0:0.3}]
solver = ['liblinear', 'saga']

param_grid = dict(penalty=penalty,
                  C=C,
                  class_weight=class_weight,
                  solver=solver)

grid_1 = GridSearchCV(estimator=Logistic,
                    param_grid=param_grid,
                    scoring='f1',
                    cv = 4,
                    verbose=1,
                    n_jobs=-1)
grid_1.fit(X_train, y_train)
evaluation(grid_1)

#Random Forest
RandomForest = RandomForestClassifier(random_state=0)
n_estimators=[100,200,300,500,1000]
max_depth = [5, 8, 15, 25, 30]
min_samples_split = [2, 5, 10, 15, 100]
min_samples_leaf = [1, 2, 5, 10]

hyperF = dict(max_depth = max_depth,
              min_samples_split = min_samples_split,
             min_samples_leaf = min_samples_leaf)

gridF = GridSearchCV(RandomForest, hyperF, scoring='f1' cv = 3, verbose = 1,
                         n_jobs = -1)
gridF.fit(X_train, y_train)
evaluation(gridF.best_estimator_)

#Gradient Boosting
GBoost = GradientBoostingClassifier(random_state=0)
parameters = {
    "learning_rate" : np.linspace(0,2,50)
    "max_depth" : [2,3,4,5,7,9,10]
    "n_estimators" : [200,300,500,700,900,1000]
}
cv = GridSearchCV(GBoost,parameters,cv=4)
cv.fit(X_train, y_train)
evaluation(cv.best_estimator_)


#Extreme Gradient Boosting
from bayes_opt import BayesianOptimization
from sklearn.metrics import mean_squared_error
#Converting the dataframe into XGBoost's Dmatrix object
dtrain = xgb.DMatrix(X_train, y_train)
#Bayesian Optimization function for xgboost
def bo_tune_xgb(max_depth, gamma, n_estimators ,learning_rate):
     params = {'max_depth': int(max_depth),
               'gamma': gamma,
               'n_estimators': int(n_estimators),
               'learning_rate':learning_rate,
               'subsample': 0.8,
```

```
              'eta': 0.1,
              'eval_metric': 'rmse'}
     cv_result = xgb.cv(params, dtrain, num_boost_round=70, nfold=5)
     return -1.0 * cv_result['test-rmse-mean'].iloc[-1]

xgb_bo = BayesianOptimization(bo_tune_xgb, {'max_depth': (3, 10),
                                            'gamma': (0, 1),
                                            'learning_rate':(0,1),
                                            'n_estimators':(100,120)
                                           })

xgb_bo.maximize(n_iter=5, init_points=8, acq='ei')
params = xgb_bo.max['params']
print(params)

#Conversion des valeurs de max_depth et n_estimator de float à int
params['max_depth']= int(params['max_depth'])
params['n_estimators']= int(params['n_estimators'])

#Initialisation d'un XGBClassifier avec les paramètres entrainés
from xgboost import XGBClassifier
classifier2 = XGBClassifier(**params).fit(X_train, y_train)

predic_p2 = classifier2.predict(X_test)
print(classification_report(predic_p2, y_test))

#Adaptative Boosting
AdaBoost = AdaBoostClassifier(random_state=0)
param_grid = {'learning_rate':np.linspace(0,1,20)
              'n_estimators = [100,200,300,400,500]'}
grid_2 = GridSearchCV(AdaBoost,param_grid, cv=5, n_jobs=-1)
grid_2.fit(X_train, y_train)
evaluation(grid_2.best_estimator_)
```

Figure U.1 : Stratégie de modélisation 1

```
#Suréchantillonnage

from collections import Counter
from imblearn.over_sampling import SMOTE
import xgboost as xgb


#Création des jeux de données X et y
XX = pd.concat([X_train,X_test])
yy = pd.concat([y_train, y_test])

#Uitilisation de l'objet censé créer des occurrences synthétiques
oversample = SMOTE()
XX, yy = oversample.fit_resample(XX, yy)
#Compter le nombre d'occurrences pour chaque classe
Counter(yy)

#Division des sous ensemble de test et d'entraînement
yy_tr, yy_tes = train_test_split(yy,test_size=0.2, random_state=2)
XX_tr, XX_tes = train_test_split(XX, test_size=0.2, random_state=2)

#Définition de la procédure d'évaluation
def evaluationn(model) :
    model.fit(XX_tr,yy_tr)
```

```
    yy_pred = model.predict(XX_tes)
    print(confusion_matrix(yy_tes,yy_pred))
    print(classification_report(yy_tes,yy_pred))
    N, train_score, val_score = learning_curve(model,XX_tr,yy_tr,cv=4,scoring="f1", train_sizes
=np.linspace(0.1,1,10))
    plt.figure(figsize=(12,8))
    plt.plot(N, train_score.mean(axis=1), label='training score')
    plt.plot(N, val_score.mean(axis=1),label='validation score')
    plt.legend()

#Arbres de décision et optimisation
DecisionTree = DecisionTreeClassifier(random_state=0)
max_depth = np.arange(1,20)
min_samples_split = np.arange(2,20)
min_samples_leaf = [9,10]

hyperDT = dict(max_depth = max_depth,
               min_samples_split = min_samples_split,
              min_samples_leaf = min_samples_leaf)

DTgrid = GridSearchCV(DecisionTree, hyperDT,scoring='f1' cv = 3, verbose = 1, n_jobs = -1)
DTbest = DTgrid.fit(XX_tr, yy_tr)
evaluationn(DTbest.best_estimator_)

#Forêts aléatoires et optimisation
RandomForest = RandomForestClassifier(random_state = 0)
n_estimators=[100,200,300,500,1000]
max_depth = [5, 8, 15, 25, 30]
min_samples_split = [2, 5, 10, 15, 100]
min_samples_leaf = [1, 2, 5, 10]
hyperF = dict(max_depth = max_depth,
               min_samples_split = min_samples_split,
              min_samples_leaf = min_samples_leaf)

Fgrid = GridSearchCV(RandomForest, hyperF, scoring='f1' cv = 3, verbose = 1, n_jobs = -1)
Fgrid.fit(XX_tr, yy_tr)
evaluationn(Fgrid.best_estimator_)


#Extreme Gradient Boosting et optimisation
XGBoosting = make_pipeline(StandardScaler(),xgb.XGBClassifier())
parameters = {
    "gamma" : np.linspace(0,2,50)
    "learning_rate" : np.linspace(0,2,50)
    "max_depth" : [2,3,4,5,6,7,9,10]
    "n_estimators" : [100,200,300,500,700,900,1000]
}
XG_B = GridSearchCV(XGBoosting,parameters,cv=4)
XG_B.fit(XX_tr, yy_tr)
evaluationn(XG_B.best_estimator_)

#Gradient boosting et optimisation
GradientBoost = GradientBoostingClassifier(random_state=0)
parameters = {
    "learning_rate" : np.linspace(0,2,50)
    "max_depth" : [2,3,4,5,7,9,10]
    "n_estimators" : [200,300,500,700,900,1000]
}
GB = GridSearchCV(GradientBoost,parameters,cv=4)
GB.fit(XX_tr, yy_tr)
evaluationn(GB.best_estimator_)
```

```
#Adaptative boosting et optimisation
AdaptativeBoosting = AdaBoostClassifier(random_state = 0)
param_grid = {'learning_rate':np.linspace(0,1,20)
              'n_estimators = [100,200,300,400,500]'}
AB = GridSearchCV(AdaptativeBoosting,param_grid, cv=5, n_jobs=-1)
AB.fit(XX_tr, yy_tr)
evaluationn(AB.best_estimator_)
```

Figure U.2 : Stratégie de modélisation 2

Annexe V : Hyperparamètre et méthodes pour les optimiser

Hyperparamètre :

Dans l'apprentissage machine, un hyperparamètre est un paramètre dont la valeur est utilisée pour contrôler le processus d'apprentissage.

Les hyperparamètres peuvent être classés comme des hyperparamètres de modèle, qui ne peuvent pas être déduits lors de l'adaptation de la machine à l'ensemble d'apprentissage car ils se réfèrent à la tâche de sélection du modèle, ou des hyperparamètres d'algorithme, qui en principe n'ont pas d'influence sur la performance du modèle mais affectent la vitesse et la qualité du processus d'apprentissage. Un exemple d'hyperparamètre de modèle est la topologie et la taille d'un réseau de neurones.

Le processus de « GridSearchCV » est le processus qui consiste à effectuer un réglage des hyperparamètres afin de déterminer les valeurs optimales pour un modèle donné. Ceci est important car la performance de l'ensemble du modèle est basée sur les valeurs des hyperparamètres spécifiés. Pour un projet de Machine Learning, ça relève du cauchemar de fixer des valeurs pour les hyperparamètres. Il existe des bibliothèques qui ont été implémentées, comme « GridSearchCV » ou « RandomizedSearchCV » de la bibliothèque « Sklearn », afin d'automatiser ce processus et de rendre la vie un peu plus facile aux amateurs de ML. Ces processus sont expliqués par le schéma suivant :

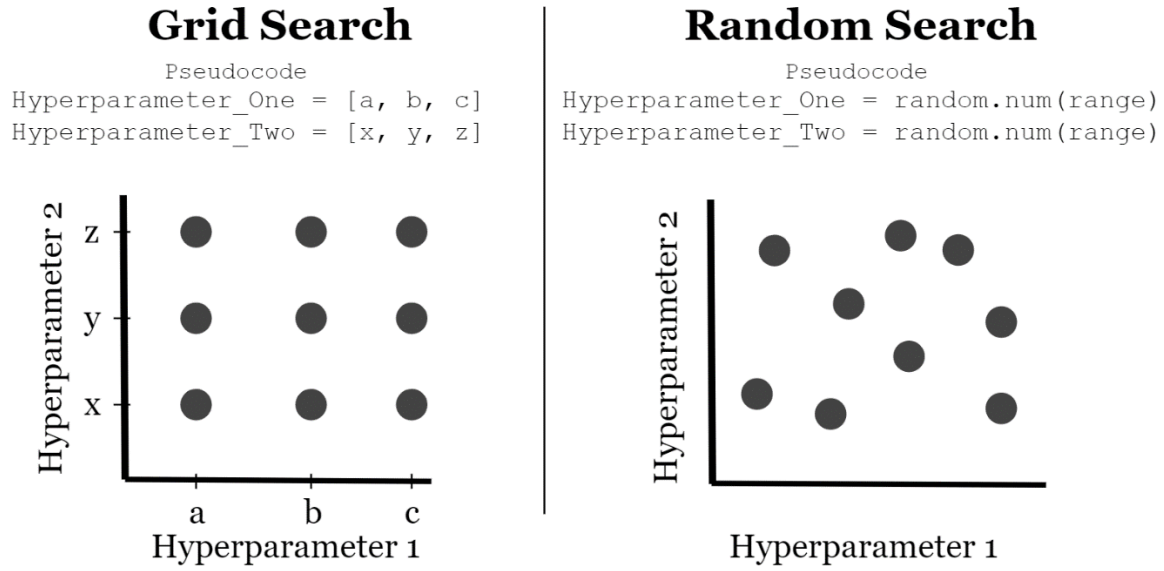


Figure V.1 : Schéma représentant les méthodes Grid Search et Random Search

Pour le « GridSearchCV » on spécifie des valeurs précises pour chaque hyperparamètre et l'algorithme effectuera une série de combinaisons de ces valeurs pour trouver la meilleure d'entre elles, optimisant une métrique définie en amont, dans notre cas c'est celle du F1 score. Une procédure de validation croisée (CV) est introduite en marge de cette optimisation afin de valider les résultats sur un sous-ensemble « validation » de l'ensemble d'entraînement.

Quant au « RandomizedSearchCV », on spécifie un intervalle de valeurs pour chaque hyperparamètre et l'algorithme va trouver la meilleure combinaison avec des valeurs inclues dans chacun de ces intervalles. Cette procédure est assez lente par rapport au « GridSearchCV »

Une troisième méthode d'optimisation, utilisée pour l'algorithme du « Extreme Gradient Boosting » est appelée optimisation Bayésienne.

Le processus gaussien est un modèle de substitution populaire pour l'optimisation bayésienne. Il définit une fonction préalable qui peut être utilisée pour tirer des enseignements des prédictions ou des croyances antérieures concernant la fonction objective.

La fonction d'acquisition, d'autre part, est responsable de la prédiction des points d'échantillonnage dans l'espace de recherche. La fonction d'acquisition suit le principe de l'exploration. C'est une fonction qui permet à l'optimiseur d'exploiter une région optimale jusqu'à ce qu'une meilleure valeur soit obtenue. L'objectif est de maximiser la fonction d'acquisition pour déterminer le prochain point d'échantillonnage. Les termes exploration et exploitation peuvent vous sembler familiers si vous avez entendu parler de l'échantillonnage de Thompson ou de la limite supérieure d'un intervalle confiance qui tourne autour du même principe. Ces termes sont également utilisés comme fonctions d'acquisition.

L'algorithme d'optimisation est présenté dans la figure ci-contre :

The optimization algorithm repeats the following steps for k = 1,2,3,..

- Find the next sampling point X_k by optimizing the acquisition function over the Gaussian Process :
$$X_k = argmax_X \; u(X|D_{1:k-1})$$
- Obtain the sample:
$y_k = f(x_k) + \varepsilon_k$ from the objective function f.
- Add above obtained sample to previous samples
$D1 : k = \{1 : k-1, (x_k, y_k)\}$ and update the Gaussian Process.

La fonction d'acquisition est donnée ainsi :

Expected improvement is defined as:

$$EI(x) = \xi max(f(x)-f(x^{best}),0)$$

Where:

- $f(x^{best})$ is the value of the best sample so far.
- $x^{best} = argmax_{x_i \varepsilon x_{1:k}} f(x_i)$ is the location of the best sample.

Annexe W : les courbes d'apprentissage des différents modèles

Les graphes suivants permettent de visualiser les courbes d'apprentissage des différents modèles appliqués à notre jeu de données ayant subi un sur-échantillonnage :

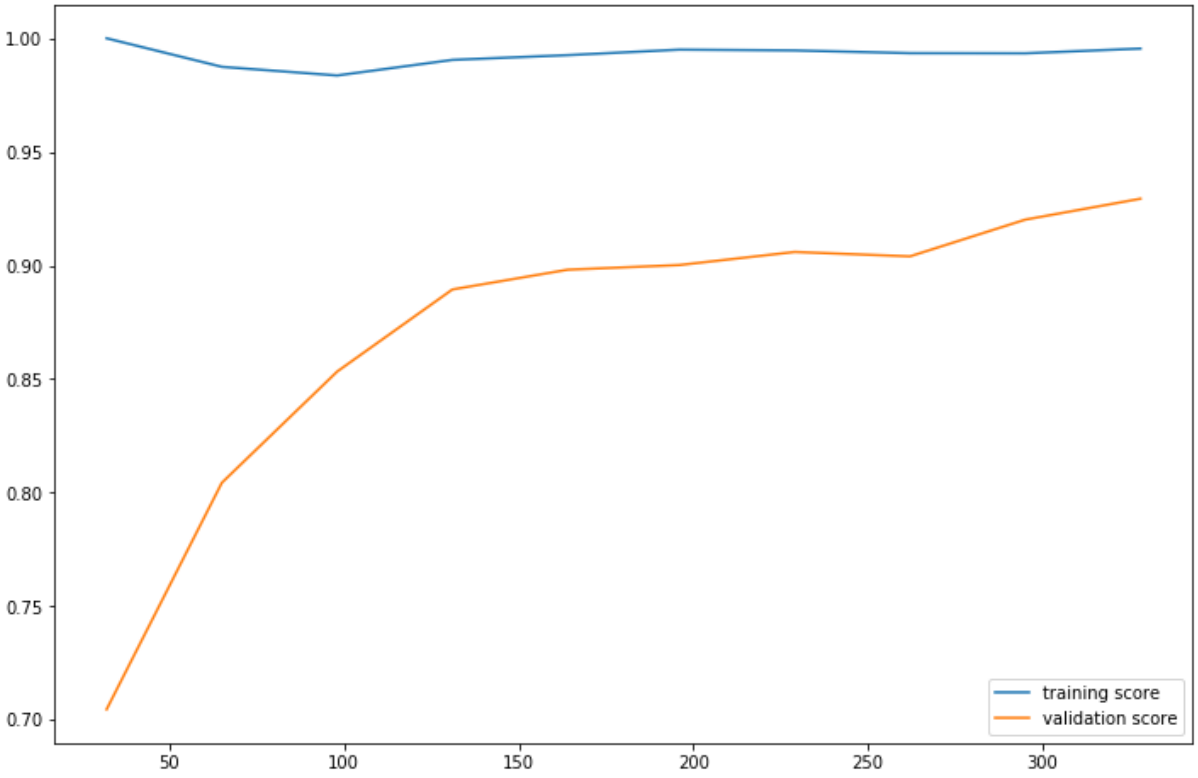


Figure W.1 : Arbres de décision

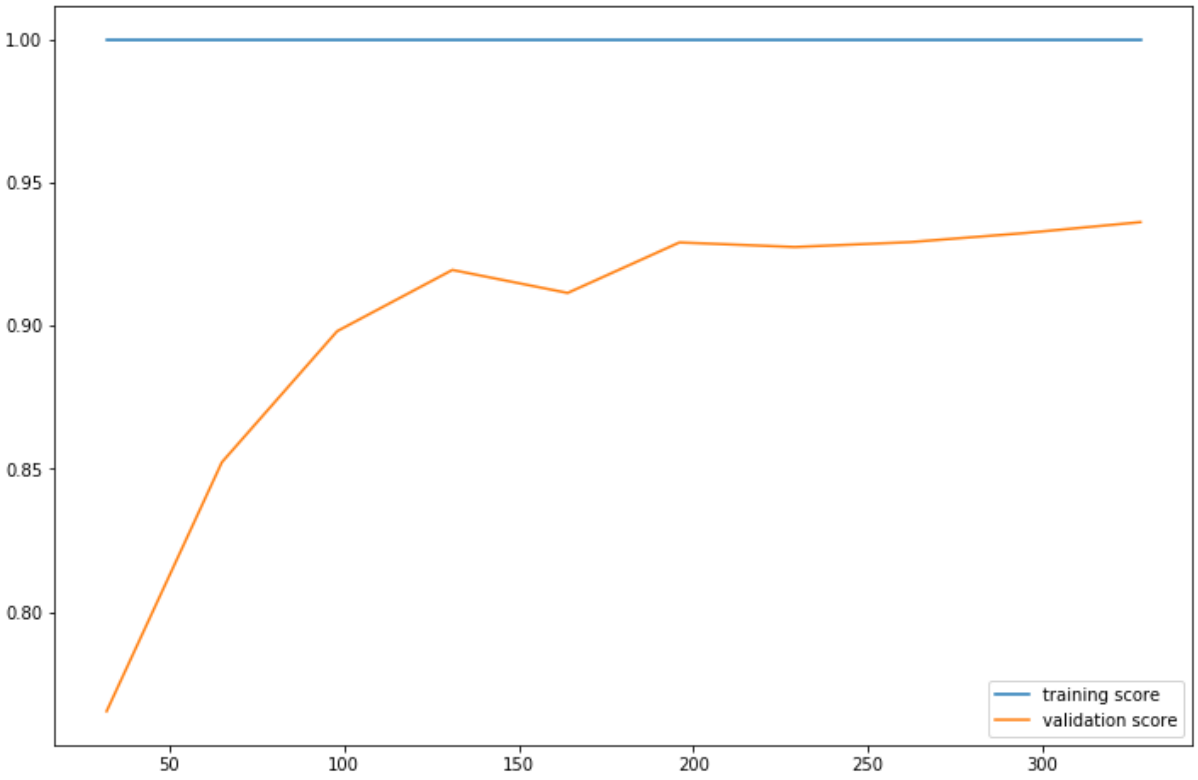


Figure W.2 : Forêts aléatoires

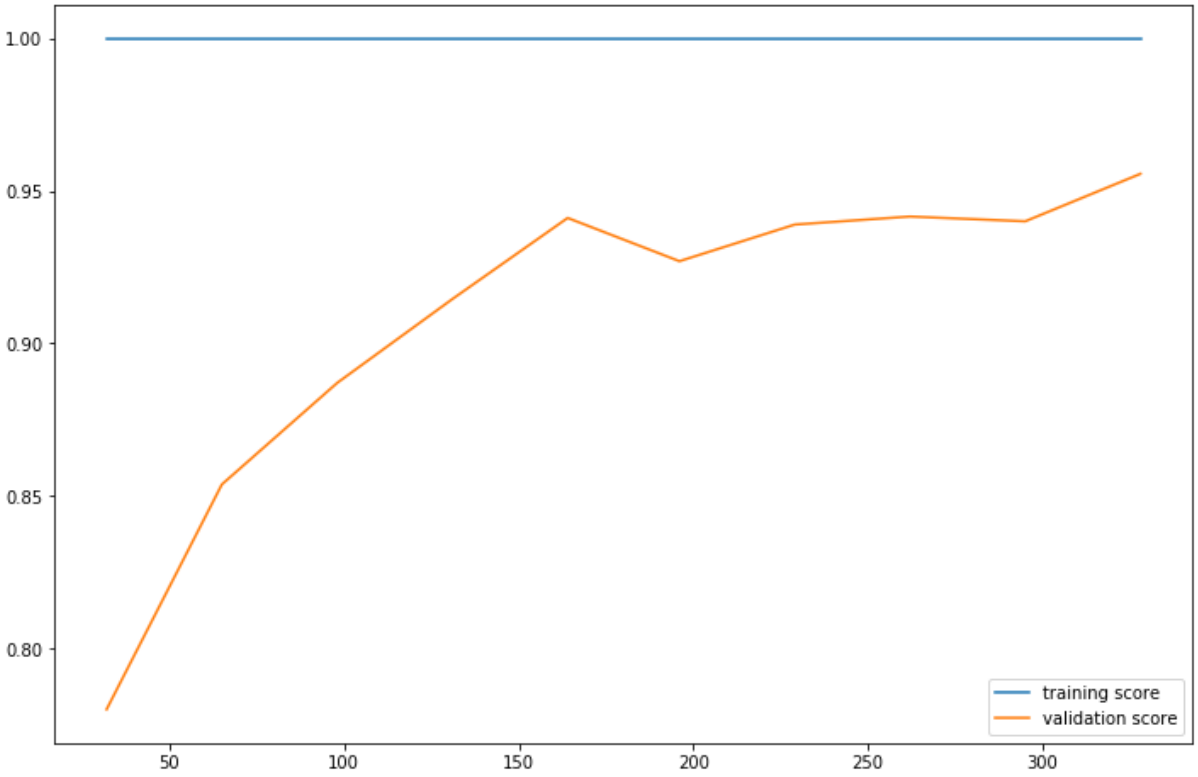


Figure W.3 : Extreme Gradient Boosting

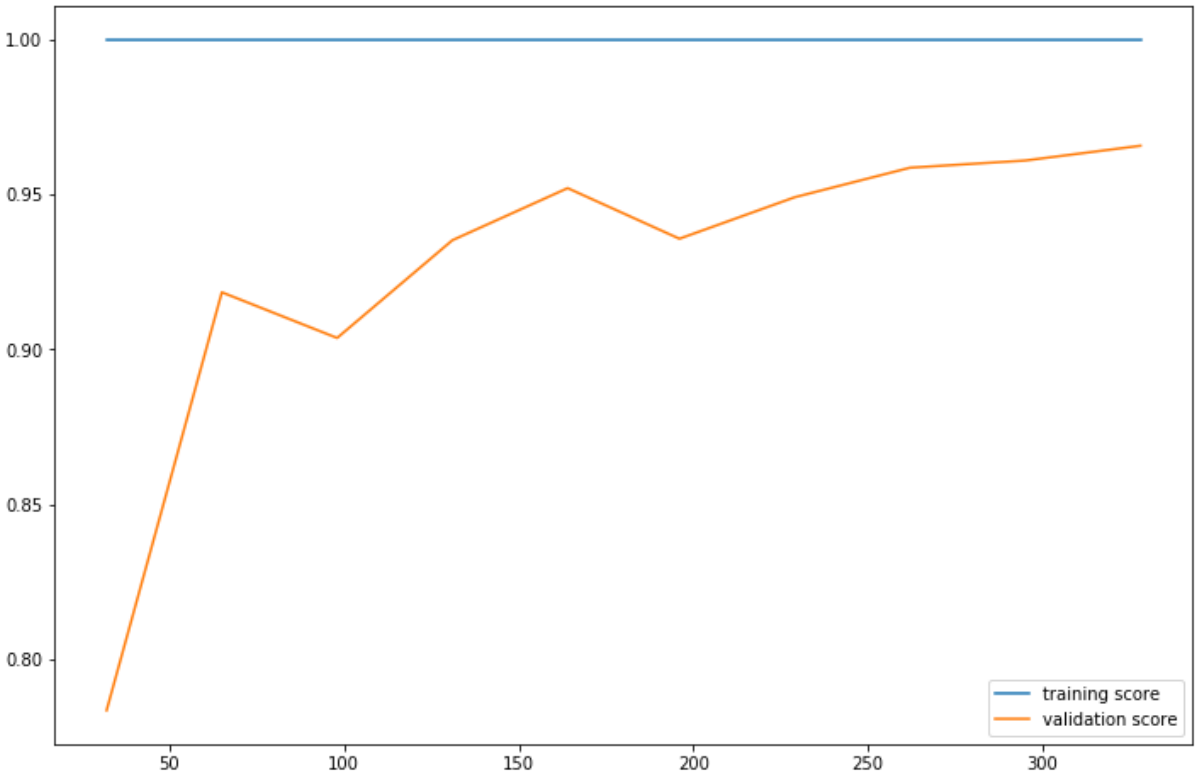


Figure W.4 : Gradient Boosting

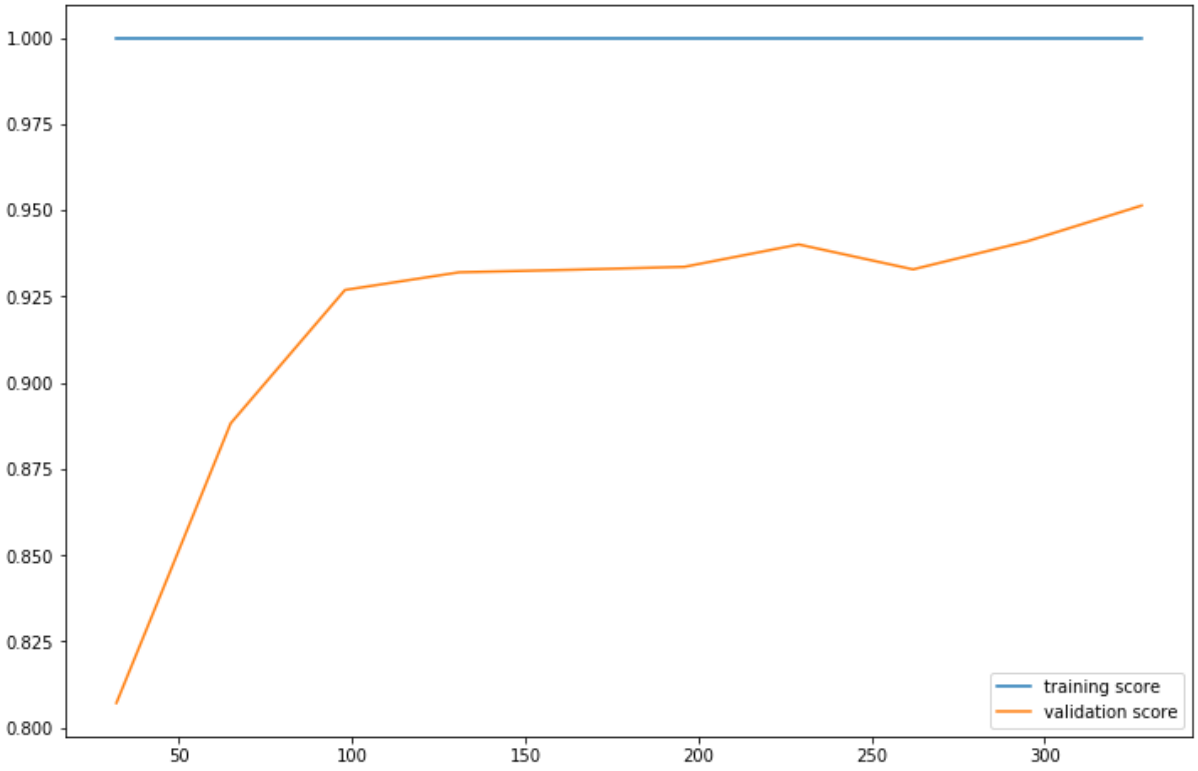


Figure W.5 : Adaptative Boosting (AdaBoost)

Annexe X : Enquêtes Churn

Annexe X.1 : Enquête Du "Pacific Crest"

L'enquête annuelle de Pacific Crest est l'ensemble de données de mesure du SaaS, et leur enquête 2016 offre un aperçu des mesures de performance de 336 entreprises SaaS - dont 177 ont déclaré leur taux de désabonnement.[56]

Pour avoir une idée des types de sociétés SaaS présentées, Pacific Crest fournit également quelques statistiques représentatives. Les entreprises de l'enquête ont un :

- Un chiffre d'affaires annuel médian de 5 millions de dollars.
- Médiane de 50 employés à temps plein.
- Valeur contractuelle moyenne (ACV) médiane de 25 000 $.[57]

L'étude elle-même montre que les taux de désabonnement des clients (appelés "désabonnement unitaire") sont répartis en différentes "tranches", allant de 1 à 3 % à plus de 15 %.

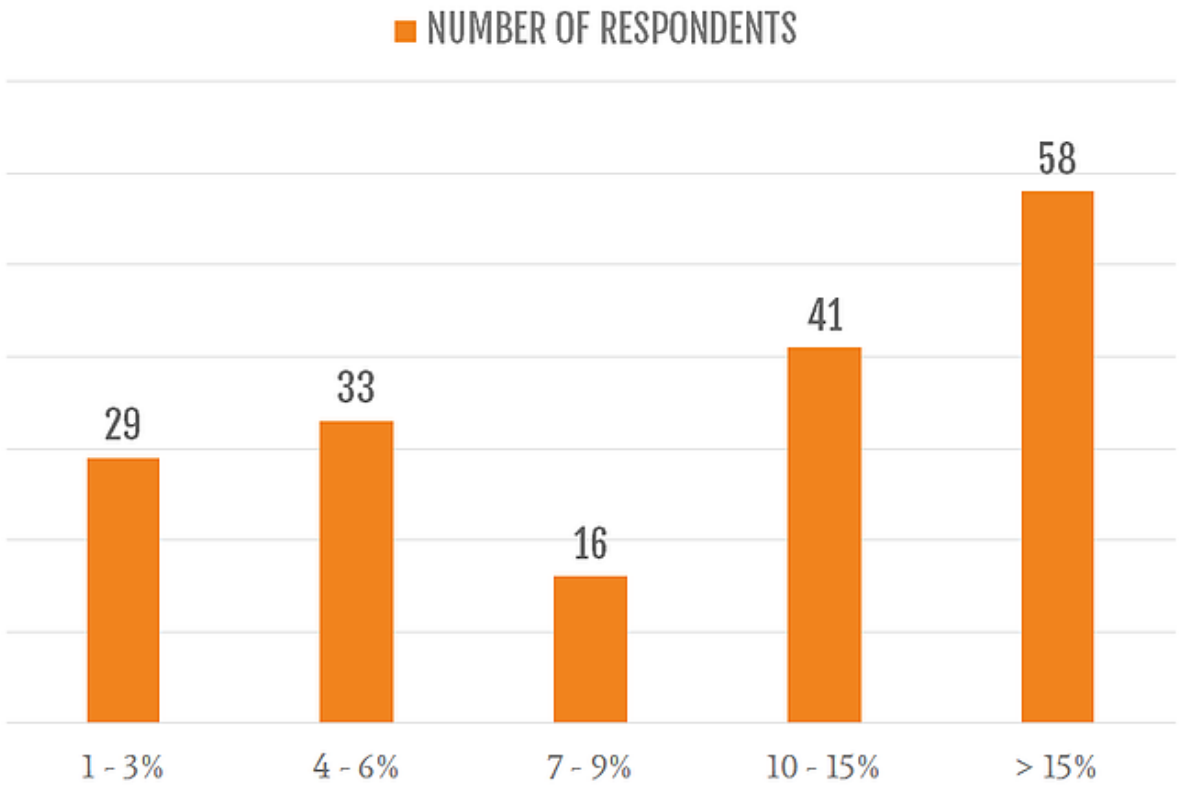


Figure X.1.1 : *La répartition du Churn Rate des participants*

Pacific Crest a indiqué que le taux d'attrition annuel médian pour l'ensemble de l'échantillon était de 10 %, soit 0,87 % par mois.

Ce chiffre est supérieur à notre taux d'attrition "idéal", mais c'est normal : un chiffre idéal est une référence que toutes les entreprises n'atteindront pas.

[56] Pacific Crest SaaS Survey
[57] Pacific Crest SaaS Survey 2016

Bien que l'utilisation de données par tranches rend difficile l'identification du nombre exact d'entreprises qui ont chuté sans notre fourchette idéale de 5 à 7 %, nous pouvons dire qu'un maximum de 78 des 177 participants (44 %) ont eu un taux de résiliation dans cette fourchette (ou mieux).

Bien que la fourchette "supérieure à 15 %" soit la plus importante, notre chiffre idéal semble toujours plausible.

Annexe X.2 : Rapport Sur Les Mesures De Totango

Comme Pacific Crest, Totango[58] a également publié un rapport annuel sur les mesures du SaaS. Leur enquête cible un éventail plus large d'entreprises ("allant des jeunes pousses aux entreprises établies ayant un chiffre d'affaire supérieur à 100 millions de dollars"), mais la majorité des participants (60 %) ont déclaré un chiffre d'affaires annuel compris entre 1 et 50 millions de dollars.

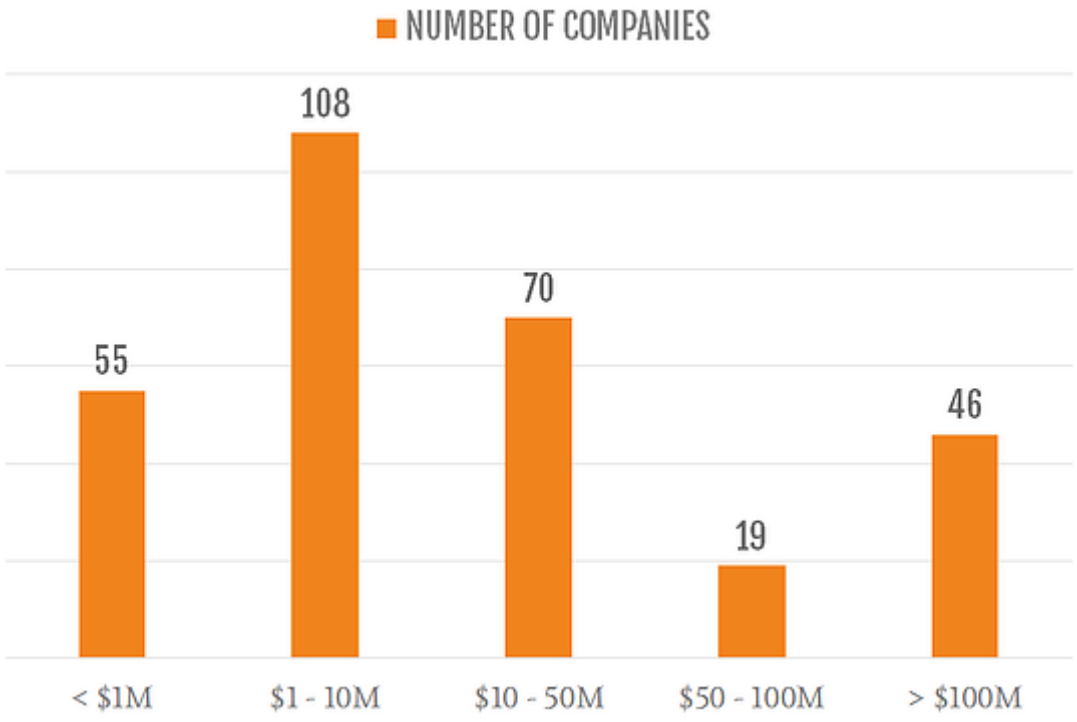


Figure X.2.1 : *La répartition du chiffre d'affaire des participants*

Totango a alors examiné les taux de résiliation annuels de trois "types" d'entreprises :

- Les entreprises à forte croissance (celles qui ont connu une augmentation de leur chiffre d'affaires de plus de 75 % d'une année sur l'autre).
- Croissance moyenne (25 à 75 %).
- Croissance faible (moins de 25 %).

Le taux de désabonnement a ensuite été rapporté pour trois grandes catégories :

[58] SaaS Metrics Report 2014

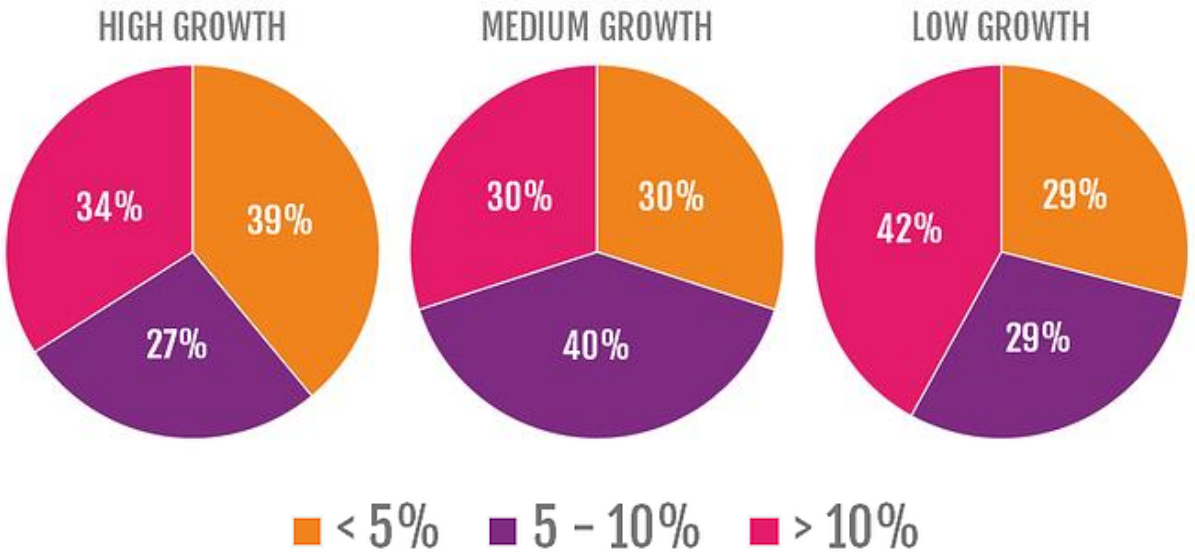


Figure X.2.2 : *La répartition du churn selon la croissance des participants*

Comme dans l'étude de Pacific Crest, les conclusions solides sont rares en raison de l'ampleur et du caractère vague des churn (11 % et 99 % des churn seraient classés dans la catégorie "plus de 10 %"), mais il est juste de dire cela :

Des taux de churn inférieurs à 5 % par an ne sont pas rares, en particulier dans les entreprises SaaS à forte croissance. Environ un tiers des personnes interrogées appartenaient à cette catégorie.

La majorité des sociétés SaaS (65 %) ont fait état d'un taux de résiliation de 10 % ou moins par an, ce qui est conforme aux conclusions de l'enquête de Pacific Crest.

Annexe X.3 : Enquête Blossom Street Ventures

Au début de l'année 2019, Blossom Street Ventures a analysé 40 sociétés SaaS cotées en bourse, afin de connaître leur taux de résiliation[59].

Ces entreprises, dont les revenus sur douze mois varient entre 75 et 382 millions de dollars, sont des sociétés comme HubSpot, Zendesk et Box (en d'autres termes, de grandes sociétés SaaS).

Seules 16 des entreprises choisies ont fait état du churn rate. Parmi celles qui l'ont fait, la majorité semble avoir fait état d'une perte de revenus (ce qui explique que la valeur médiane de rétention soit de 111 % - ce qui n'est possible qu'en raison d'une perte de revenus négative).

Sur ces 16 entreprises, seules 6 ont déclaré une rétention inférieure à 100 % (ce qui permet d'affirmer qu'elles faisaient référence à la perte de clientèle plutôt qu'aux revenus). Parmi celles-ci, le chiffre le plus bas rapporté était une rétention mensuelle de 99%. Si l'on suppose qu'il s'agit d'une perte de clientèle (difficile à déterminer à partir des seules données), l'entreprise en question aurait une perte de clientèle de 1% par mois, ou de 11,3% par an.

[59] Blossom Street Ventures Survey

Bien qu'il soit difficile de tirer des conclusions à partir de ces données, nous pouvons au moins affirmer que le taux d'attrition annuel le plus élevé possible est légèrement supérieur à 11 % - toutes les autres entreprises ayant déclaré un taux d'attrition inférieur. Cela semble corroborer les conclusions de Totango et de Pacific Crest.

Annexe X.4 : Les Points De Référence De Baremetrics

Jusqu'à présent, l'histoire a été relativement cohérente : la plupart des sociétés SaaS semblent avoir un taux de résiliation de l'ordre de 10% par an, ce qui équivaut à moins de 1% par mois.

Malheureusement, c'est là que les choses commencent à dévier.

Baremetrics est une plateforme d'analyse conçue pour les entreprises SaaS. Sur sa page Benchmarks[60], la société utilise ses connaissances pour regrouper les analyses en temps réel des tableaux de bord de plus de 600 petites et moyennes entreprises de SaaS - y compris les taux de désabonnement des clients.

Baremetrics se concentre sur le taux de désabonnement mensuel des clients et rend compte des taux de désabonnement mensuels moyens dans des cohortes d'entreprises ayant des valeurs ARPA (revenu moyen par compte) similaires.

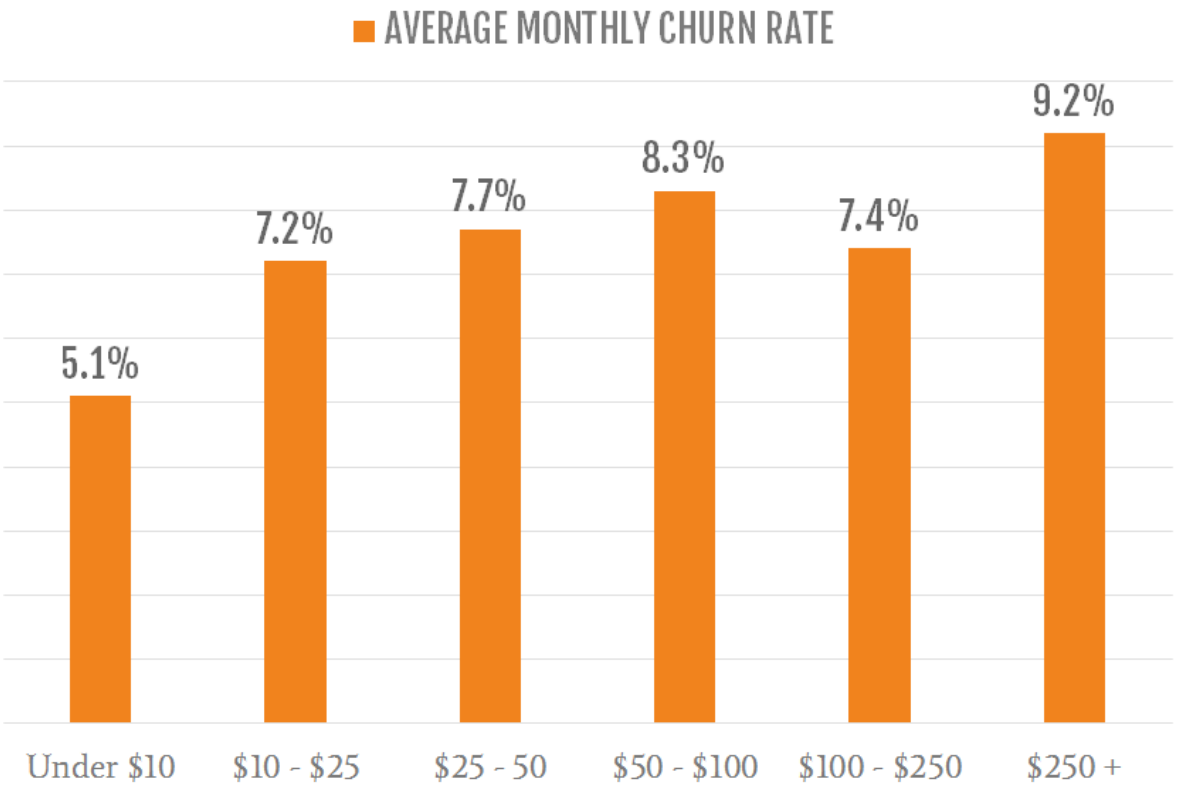


Figure X.4.1 : *La répartition churn selon l'ARPA des participants*

Si l'on fait la moyenne de ces cohortes, on constate un taux d'attrition mensuel de 7,5 %.

[60] Baremetrics Benchmarks

Sur une base annuelle, cela équivaut à un taux de désabonnement de 61 %, soit environ six fois plus que la moyenne enregistrée par Pacific Crest, Totango et Blossom Ventures.

C'est un écart énorme, qui va à l'encontre de l'objectif de 5 à 7 % d'attrition annuelle, mais il est important de noter que ces résultats sont confirmés par d'autres sources de données.

Annexe X.5 : Enquête De Groove's Saas

En 2013, Groove a mené ses propres recherches sur l'industrie du SaaS[61]. Leur enquête s'est concentrée sur les petites entreprises, recueillant les réponses de 712 sociétés SaaS, toutes en phase de post-production/marketing et entre 1 000 et 500 000 dollars en MRR.

Nous n'avons trouvé aucune mention des données brutes, mais Groove a rapporté le taux de rotation mensuel moyen sur l'ensemble de l'échantillon : 3,2 % de désabonnement mensuel, ou annualisé, soit un taux de désabonnement de 32,3 %.

Bien qu'il soit environ deux fois moins important que le taux de désabonnement moyen indiqué par Baremetrics, ce résultat est toujours du même ordre de grandeur, et très éloigné des taux de désabonnement mensuels inférieurs à 1 % que nous avons vus dans les autres ensembles de données.

Annexe X.6 : Les Startups Ouvertes

Pour aller plus loin, nous pouvons nous tourner vers les différentes startups SaaS qui utilisent la fonctionnalité de reporting de Baremetric.

Sur les 600 entreprises qui utilisent Baremetrics, 18 rendent actuellement publiques les données de leur tableau de bord, et leur taille varie de 447 $ en MRR (Helpman) à 1 078 560 $ (Buffer).

Les utilisateurs les plus connus de Baremetrics sont les tampons, et leurs derniers chiffres font état d'un taux de désabonnement mensuel de 5,1 % (ou d'un taux de désabonnement annuel de 46 %).

Buffer est une grande entreprise SaaS prospère, et l'idée qu'elle perde 46 clients sur 100 chaque année semble choquante, surtout si on la compare aux taux d'attrition moyens signalés par Pacific Crest et Totango. Mais c'est une statistique qui est confirmée par leur propre analyse de cohorte :

[61] Groove's SaaS Conversion Survey

Subscription		Months since account creation											
		0	1	2	3	4	5	6	7	8	9	10	11
January 2016	3485	99%	93%	87%	82%	78%	75%	72%	70%	68%	66%	64%	62%
February 2016	3275	99%	93%	87%	82%	79%	75%	72%	69%	67%	66%	63%	
March 2016	3259	99%	92%	86%	81%	77%	73%	71%	68%	66%	64%		
April 2016	3038	99%	94%	87%	82%	78%	75%	72%	69%	66%			
May 2016	2963	99%	94%	86%	81%	78%	74%	71%	68%				
June 2016	3205	99%	93%	87%	82%	78%	75%	72%					
July 2016	3256	99%	93%	87%	82%	78%	73%						
August 2016	3711	100%	94%	88%	83%	77%							
September 2016	3971	100%	93%	87%	80%								
October 2016	3822	99%	91%	84%									
November 2016	3274	99%	92%										
December 2016	3136	99%											

Figure X.6.1 : *Analyse de cohorte* de Baremetrics

Selon les données de leur cohorte actuelle, seuls 62 % des clients qui se sont inscrits en janvier 2016 étaient encore clients au début de 2017. Pour cette cohorte particulière, cela représente un taux d'attrition annuel de 38 % (et la cohorte de février semble être pire).

Ces chiffres se retrouvent partout, ce qui rend très plausible la moyenne de 7,5 % de désabonnement mensuel de Baremetrics. Les Baremetrics eux-mêmes font état d'un taux de désabonnement mensuel de 5,8 % (51 % par an). Le SaaS HubStaff montre un taux de désabonnement mensuel de 5,6 % (50 % par an). L'outil de marketing par courriel ConvertKit fait état d'un taux de désabonnement de 8 % par mois (63 % par an).

Annexe Y : les courbes d'apprentissage des modèles de la stratégie 1

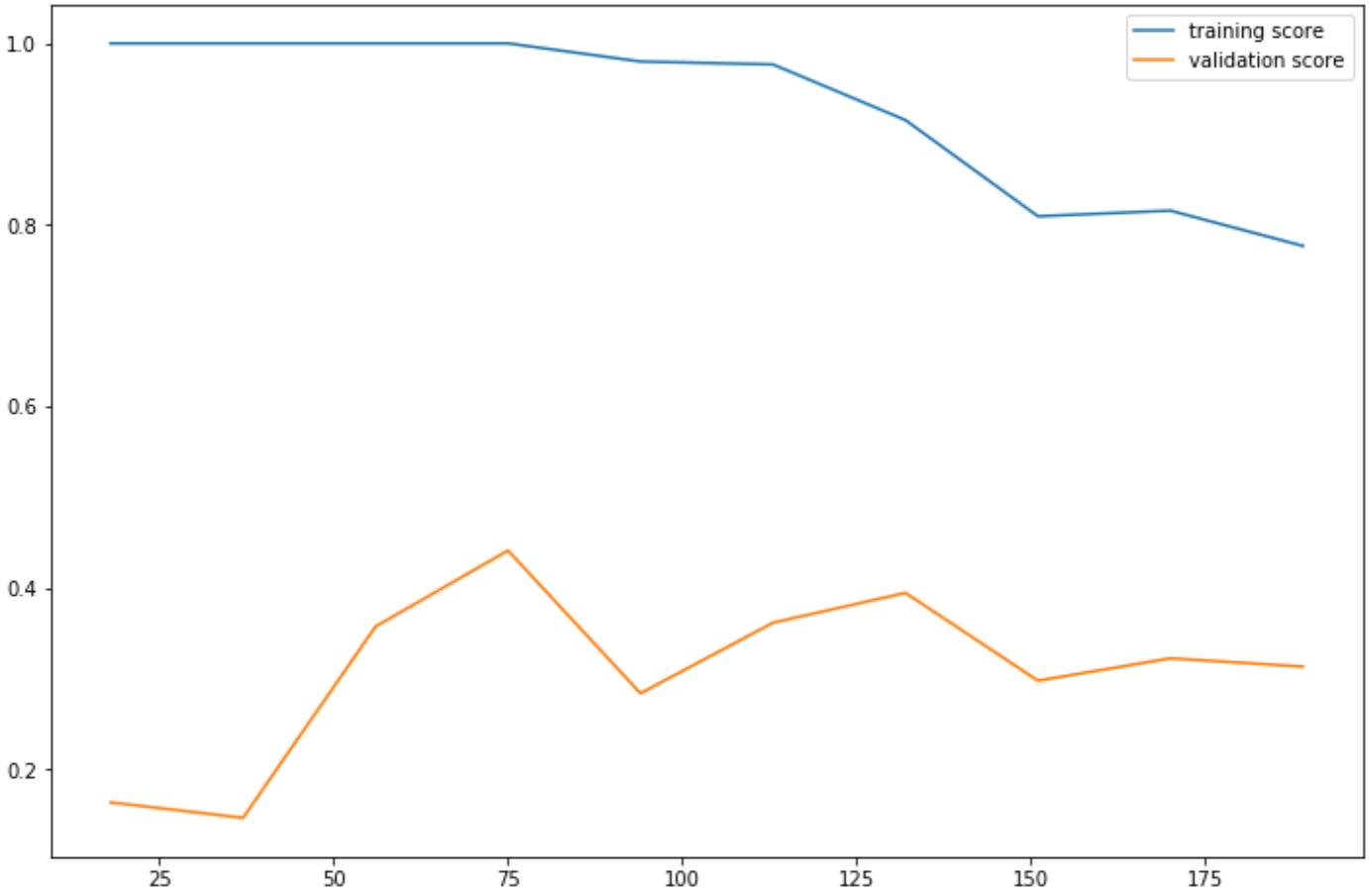


Figure Y.1 : *La courbe d'apprentissage du modèle SVM à noyau linéaire*

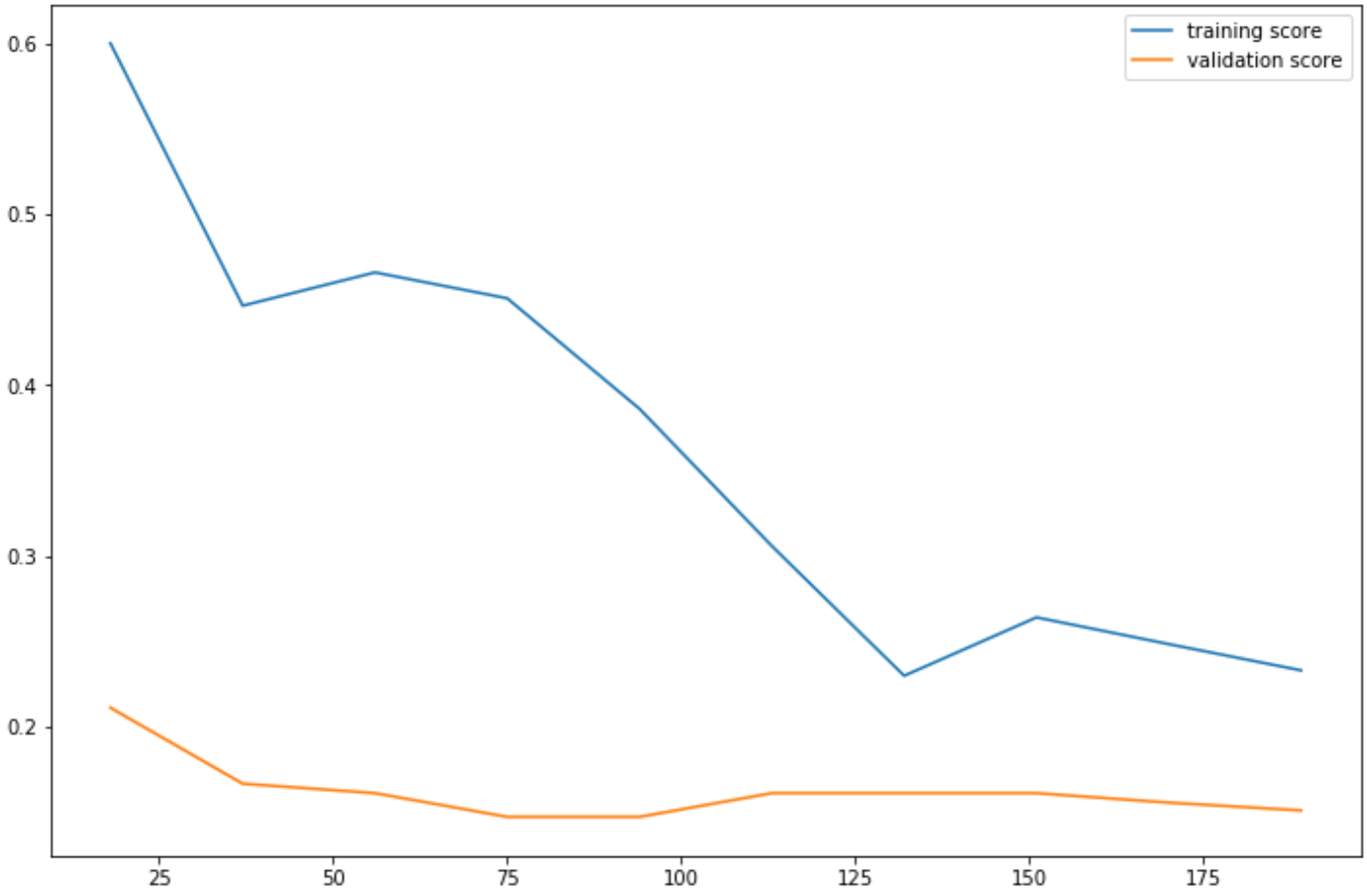


Figure Y.2 : *La courbe d'apprentissage du modèle SVM à noyau 'rbf'*

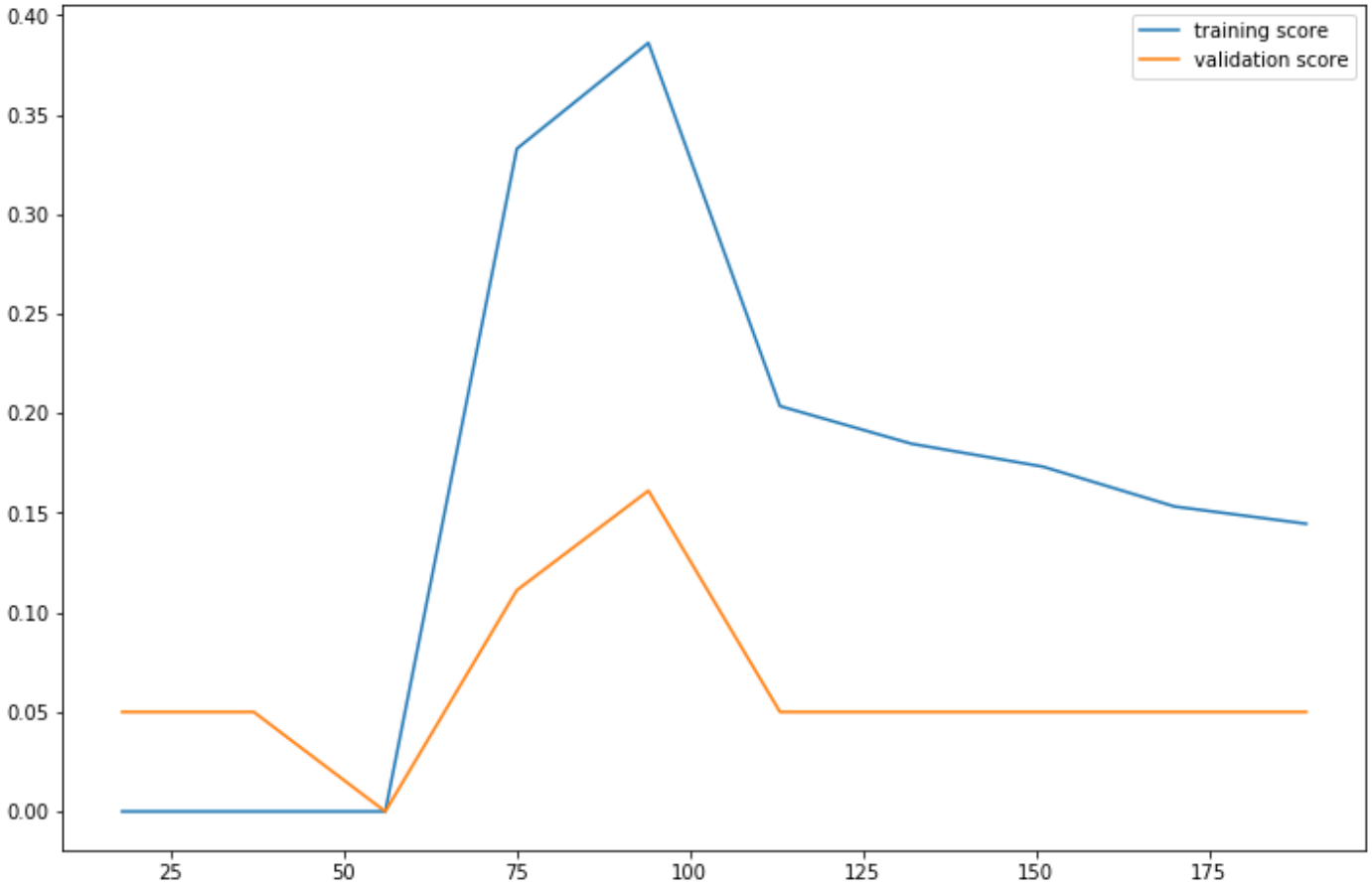


Figure Y.3 : *La courbe d'apprentissage du modèle SVM à noyau 'sigmoid'*

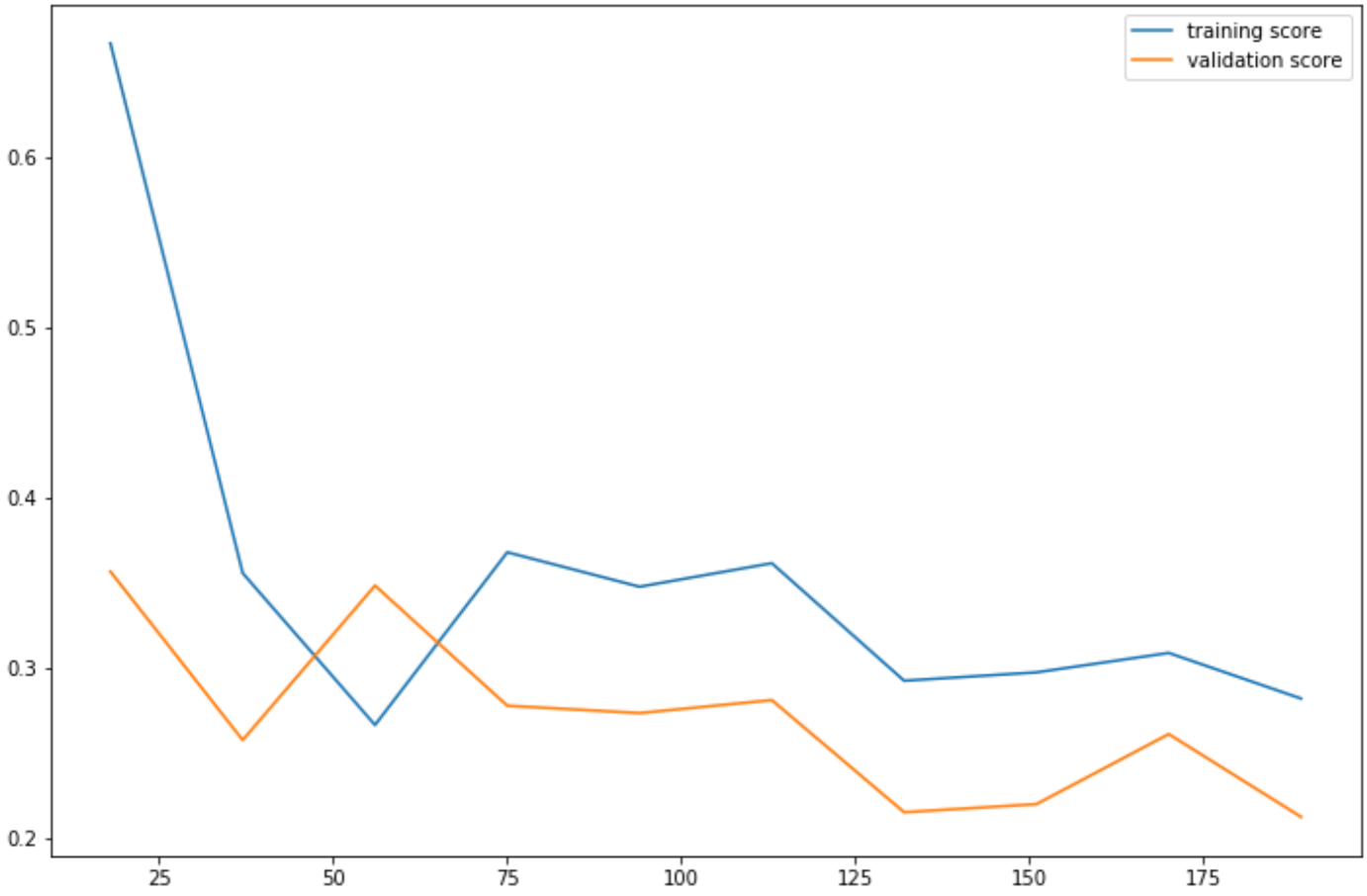


Figure Y.4 : *La courbe d'apprentissage du modèle Logistique régression*

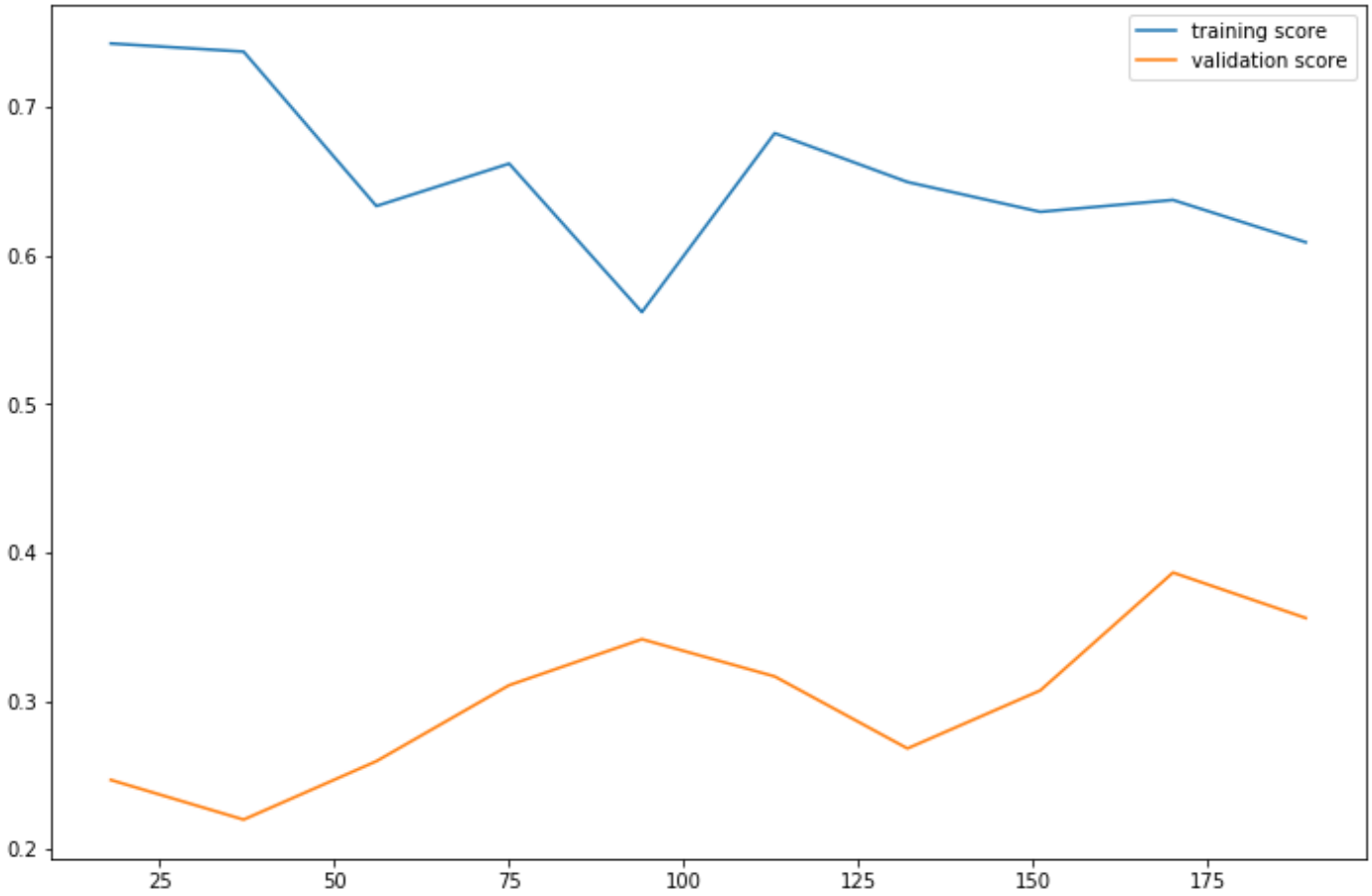


Figure Y.5 : *La courbe d'apprentissage du modèle KNN*

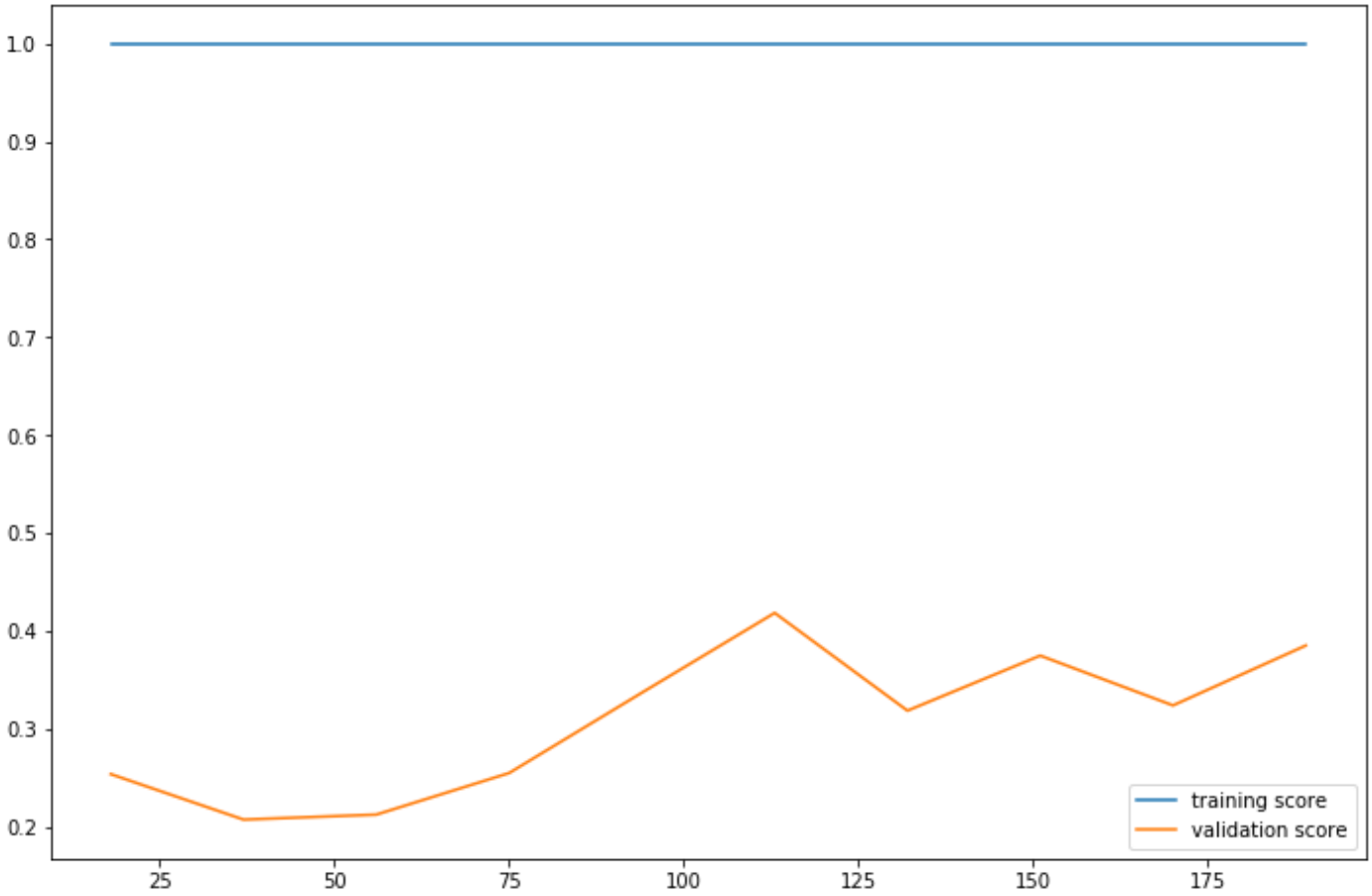


Figure Y.6 : *La courbe d'apprentissage du modèle Perceptron*