Mémoire d'alternance

Master 2 Ingénierie Statistique et Data Science

Mission : Résolution de problématiques métier à l'aide de méthodes de statistique et de science des données

Maître d'apprentissage :
Paul Coudret
Manager du département Modélisation et Intelligence des Données

Tuteur pédagogique :
Michel Broniatowski
Responsable pédagogique du département ISDS de l'ISUP

Alternant :
Vincent Le Goualher
Master 2 ISDS

Paris La Défense, 17/06/2019 - 31/08/2020

Remerciements

Je voudrais tout d'abord exprimer toute ma gratitude envers Paul Coudret pour la confiance qu'il m'a accordée et pour la manière dont il m'a intégré à l'équipe de Modélisation et Intelligence des Données. La qualité et la constance de son accompagnement tout au long de cette alternance ont été extrêmement précieuses, tant sur le plan humain que sur le plan technique. Je suis très heureux de pouvoir le remercier aussi pour le renouvellement de sa confiance et le soutien déterminant qu'il m'a apporté dans le cadre de mon embauche.

Je tiens également à remercier mes collègues Lauretta Djemi Tchatat et Solange Klein pour l'accueil très chaleureux qu'elles m'ont réservé, ainsi que l'aide qu'elles m'ont apportée à de nombreuses occasions. C'est un plaisir de pouvoir continuer à travailler avec elles. Merci à mon directeur Marc-Olivier Munsch et à tous les collègues avec qui j'ai travaillé cette année : Aurélie Saje, Julien Navarro, Xavier Bres, Bénédicte Guérin, Françoise Tenailleau, Philippe Bedin, Olivier Guelzec, Michel Ehret, Sébastien Coquelet, Franck Troesch, Oleg Popa, Nathalie Jounel, Olivier Charrette, Rita Da Costa, Hugues Desbarats, Laurent Clautour, Yasmine Mahli.

Merci à Michel Broniatowski d'avoir bien voulu m'accueillir à l'ISUP. Je le remercie également pour son cours de statistique de haut niveau et pour sa supervision du parcours ISDS dont j'ai particulièrement apprécié certains cours, parfois en dépit de leur difficulté : calcul stochastique avec Zhan Shi, *machine learning* avec Claire Boyer, *C++* avec Raphaël Roux, processus markoviens avec Olivier Bardou, réseaux de neurones avec Annick Valibouze et fiabilité avec Emmanuel Rémy, sans oublier l'anglais avec Cyprien Zitoun. Je remercie l'ensemble des professeurs et plus particulièrement ceux qui viennent d'être cités.

Je remercie Nathalie Obert-Ben Taieb pour l'accueil chaleureux qu'elle m'a réservé, son accompagnement et sa disponibilité.

Je souhaite également adresser des remerciements particuliers à mes camarades Cyprien Ferraris, Clémence Léguillette, Paul Malfrait, Margaux Morin et Raphaël Mignot pour leur aide, leur gentillesse et leur compétence qui m'a souvent laissé admiratif.

Enfin je remercie ma famille. C'est grâce à son soutien que j'ai pu être à la hauteur des exigences académiques de cette formation et suivre le rythme de travail soutenu imposé par l'alternance.

Table des matières

1 Contexte

1.1 *Crédit Mutuel*

Le *Crédit Mutuel* est l'un des acteurs financiers majeurs en France depuis sa création en 1882. C'est une banque coopérative et mutualiste, non une société par actions. À ce titre, le *Crédit Mutuel* appartient à ses huit millions de clients-sociétaires et c'est à eux qu'elle rend des comptes. Le groupe compte environ 80 000 salariés et 30 millions de clients pour un résultat net de 3,5 Md€ en 2019. Principalement implanté en Europe de l'Ouest, ses activités couvrent surtout les secteurs de la banque de détail et d'investissement : on peut citer les marques *Crédit Mutuel*, *CIC* et *Cofidis*. Comme le montre l'organigramme présenté en annexe A, le groupe s'illustre aussi dans bien d'autres domaines comme l'assurance ou la presse mais aussi de la technologie en général via sa filière informatique Euro-Information : télésurveillance, monétique (*Monetico Paiement*) ou téléphonie (*NRJ Mobile*) par exemple. Sur la page Wikipédia de Yann Le Cun [1], considéré comme l'un des inventeurs des réseaux de neurones à convolution, on lit : "Dans les années 1990, Yann Le Cun développe la technique des réseaux convolutifs pour la reconnaissance d'image, technologie mise en application rapidement par le *Crédit mutuel de Bretagne* pour la lecture optique de chèques."

Le *Crédit Mutuel* est lui-même composé de plusieurs groupes : *Crédit Mutuel Alliance Fédérale* (groupe principal), *Crédit Mutuel Arkéa* et *Crédit Mutuel Nord Europe* notamment.

À noter que le groupe *Crédit Mutuel* s'investit aussi dans le monde de la musique à travers son organisation *Le Crédit Mutuel donne le LA*, qui parraine et finance de nombreux événements au niveau local comme national.

1.2 *Centre de métier Leasing*

Le *Centre de métier Leasing* du groupe *Crédit Mutuel Alliance Fédérale* se spécialise dans le crédit-bail et les services associés à cette activité. Il s'agit de financement de matériel de tout type : bureautique, informatique, bâtiment, construction, agriculture, médical, etc... Les clients sont très majoritairement des professionnels – et plus marginalement des particuliers. L'entreprise propose plusieurs produits bancaires selon le type de bien financé et les besoins du client. Ces produits correspondent pour la plupart à des contrats de location, souvent avec option d'achat. C'est donc le *Centre de métier Leasing* qui est propriétaire des biens financés et non le client, comme c'est le cas dans le cadre des prêts bancaires "classiques".

Le *Centre de métier Leasing* compte environ 600 salariés pour une production — c'est-à-dire un montant de biens financés — de 6 milliards d'euros en 2019. C'est le leader sur le marché du leasing français. À l'échelle européenne, le secteur est dominé par *Société Générale Leasing Solutions* et *BNP Paribas Leasing Solutions*.

L'activité du *Centre de métier Leasing* se décompose en deux entités et deux *business model* bien spécifique :

- *Crédit Mutuel Leasing*
- *CCLS*

L'entité *Crédit Mutuel Leasing* s'appuie sur le réseau d'agences bancaires du groupe, soit près de 6 000 points de vente. Environ 125 000 contrats sont signés chaque année par cette entité pour une production de 5 milliards d'euros et 250 employés hors réseau d'agences bancaires. On peut souligner le fait que la majorité de la production — 60% — concerne le financement de véhicules. *Crédit Mutuel Leasing* travaille avec des entreprises (85% du chiffre d'affaires) et dans une moindre mesure avec des particuliers.

L'entité *CCLS* représente quant à elle 350 salariés, 1 milliard d'euros de production et 50 000 contrats annuels. Elle opère uniquement auprès des professionnels. Elle s'appuie sur ses commerciaux ainsi que son *Espace Partenaires*, qui permet de réaliser des demandes de financement en ligne et d'obtenir une réponse automatique donc rapide. Contrairement à *Crédit Mutuel Leasing*, *CCLS* ne traite pas directement avec le client final. Cette entité travaille en étroite collaboration avec des entreprises appelées "apporteurs" ou "partenaires". Ce sont souvent les distributeurs ou producteurs du bien financé.

Donnons un exemple pour illustrer cet activité. Le cabinet d'avocats **A** souhaite installer des copieurs commercialisés par l'entreprise **K**. **A** ne souhaite pas acheter les copieurs au comptant et préfère financer ce matériel par *leasing*. Le *Centre de métier Leasing* achète comptant le bien à **K** et en devient donc propriétaire. **A** paie ensuite des loyers au *Centre de métier Leasing*. Ces loyers peuvent inclure, en plus du remboursement du bien en lui-même et des intérêts, de la maintenance ou de l'assurance. L'entreprise **K** fait alors appel au *Centre de métier Leasing* pour ses solutions de financement. L'entreprise **K** devient un "partenaire" ou un "apporteur" du *Centre de métier Leasing*. Il faut noter que le client final **A** rembourse le bien auprès du *Centre de métier Leasing*. À ce titre, c'est la banque qui assume le risque en cas de défaut du client final.

Les avantages pour l'apporteur sont notamment la perception du paiement dès la livraison du matériel, l'offre de financement complète et clé en main qu'elle est en mesure de proposer à ses clients, la fidélisation client via le suivi spécifique des échéances de contrats et plus généralement le développement de l'activité. Côté client, les avantages du leasing sont la maîtrise du budget et la préservation de la trésorerie, la simplicité de la gestion du matériel (entretien, renouvellement) et la réduction d'actif immobilisé au bilan, les loyers versés étant comptabilisés comme des charges.

L'entité *CCLS* qui gère ce type d'opérations compte environ 1500 apporteurs pour 100 000 clients. Elle reçoit en moyenne une demande de financement chaque minute, qui sont traitées à 80% par l'outil interne de décision automatique. Il s'agit d'un programme qui évalue les demandes de financement sur la base de règles métier et de modèles statistiques. Le résultat de cette évaluation est une acceptation automatique, un refus automatique ou un transfert aux Risques pour un traitement manuel de la demande de financement.

Il est important de préciser que *CCLS* faisait partie du groupe *General Electric* jusqu'à son rachat en 2016 par *Crédit Mutuel Alliance Fédérale*, alors que *Crédit Mutuel Leasing* a toujours été rattaché au *Crédit Mutuel*. Par conséquent, ces deux structures diffèrent par leur culture d'entreprise et pas seulement par leur *business model*. C'est un point important à prendre en compte dans le cadre de leur rapprochement au sein d'un unique *Centre de métier Leasing*. Par ailleurs, les systèmes d'information relatifs à chaque entité sont distincts.

1.3 Le département MID

L'alternance s'est déroulée au sein du département **M**odélisation et **I**ntelligence des **D**onnées, sous la direction de Paul Coudret qui en est le manager. Il compte quatre personnes. Les missions du département concernent principalement l'analyse statistique, la modélisation et la gouvernance des données. Les problématiques métier associées sont par exemple la détection des fraudes, la prévision de la décote du matériel, l'étude des probabilités de défaut, la cotation et l'octroi de financement ou le contrôle automatique de la qualité des données.

Le département MID travaille en étroite collaboration avec le département de *Suivi de la Performance Entreprise* qui compte six personnes et qui s'attelle à l'extraction de données, au *reporting* et à l'analyse de données. Les deux équipes sont réunies au sein du même open space.

Actuellement, les missions du département MID sont également réparties sur les deux antennes du *Centre de métier Leasing*. Néanmoins, il faut souligner que cette situation est récente et que MID faisait initialement partie de *CCLS*. Par conséquent, la maîtrise des problématiques métier spécifiques à chacune des entités et de leur système d'information est hétérogène.

Enfin, il faut noter que le département MID est amené à travailler avec presque tous les services de l'entreprise. À ce titre, les attentes des uns et des autres vis-à-vis d'un même modèle sont parfois contradictoires. Par exemple, le Commerce a tendance à favoriser la production et donc l'exposition au risque, alors que les Engagements (gestion des risques) souhaitent plutôt minimiser la prise de risque si l'impact sur la production n'est pas jugé trop important.

2 Missions

Le travail effectué au cours de cette alternance se décompose en quatre missions principales qui sont chacune associée à une problématique métier spécifique :

1. Classification des apporteurs
2. Animation d'une communauté de données / Détection d'anomalies
3. Tarification des forfaits d'entretien automobile
4. Modélisation de la valeur résiduelle des biens financés

Les deux premières missions sont terminées. Les deux autres sont toujours en cours de réalisation et doivent être terminées avant la fin de l'année 2020.

La méthode de résolution qui a été employée peut être résumée de la façon suivante :

- Réunions de travail avec le département concerné par la problématique métier
- Interprétation de la problématique en un phénomène statistique à modéliser
- Recherche bibliographique : modèles existants, données nécessaires, méthodes de calcul, estimation des performances, etc...
- Déploiement des méthodes appropriées pour la construction du modèle souhaité.
- Comparaison des différentes méthodes mise en œuvre : performances, temps de calcul, ergonomie, etc...
- Présentation des résultats auprès du département dont émane la problématique traitée par le modèle
- Ajustement du modèle & conclusion de la mission

Les outils logiciels utilisés pour la mise en œuvre des méthodes statistiques et la construction des modèles sont présentés en annexe B.

2.1 Classification des apporteurs

2.1.1 Données du problèmes

Les apporteurs sont les entreprises qui fournissent le matériel au client final. Le rôle du *Centre de métier Leasing* est de permettre au client final de financer le bien loué - éventuellement avec option d'achat - auprès de l'apporteur d'affaires. Ce processus a été expliqué plus en détails dans la partie 1.2. Dans ce contexte, le *Centre de métier Leasing* attribue une cote à chacun de ses apporteurs, soit environ 2000 entreprises. L'objectif de cette note qualitative est de synthétiser la qualité des apporteurs : les clients de l'apporteur sont-ils globalement fiables ? Font-ils peu défaut ? Le volume de contrats soumis par l'apporteur est-il important ?

Actuellement, la cote d'un apporteur est attribuée manuellement. Il s'agit d'une variable qualitative ordinale qui peut prendre six valeurs distinctes : très bon apporteur / bon apporteur / apporteur standard / apporteur restreint /apporteur à risque / apporteur à haut risque. Le choix d'un nombre de modalités égal à six est arbitraire et correspond aux besoins des Engagements. À ce titre, il doit être conservé lors de la modélisation.

L'objectif de la mission est de prédire et d'actualiser la cote des apporteurs du *Centre de métier Leasing*. Ce processus est appelé à être effectué tous les mois. Les cotes ainsi déterminées seront toujours vérifiées par le service des Engagements. Ainsi, le rôle du modèle est de faciliter la phase d'évaluation des apporteurs en confirmant des changements de cotes effectuées par les humains, ou en suggérant de nouveaux.

Il s'agit donc ici d'un problème de classification supervisée, où la variable à expliquer est la cote des apporteurs. Les variables explicatives qui ont pu être obtenues ne sont pas présentées pour des raisons de confidentialité.

Un modèle est construit chaque mois. L'échantillon d'apprentissage est constitué par les données sur les apporteurs — cotes et variables explicatives — collectées sur les n mois précédant. Le choix de l'entier $n \in \mathbb{N}^*$ est un enjeu important de l'étude : s'il est trop petit, alors la quantité de données est insuffisante pour obtenir un modèle prédictif. Si n est trop grand, alors l'information est potentiellement redondante et on observe un phénomène de surapprentissage qui dégrade les performances de la classification sur l'échantillon de test, i.e le mois en cours. Dans cette perspective, les différentes méthode utilisées pour élaborer la classification ont été testées pour n mois d'apprentissage, $n \in [\![1; 6]\!]$.

2.1.2 Méthodes de résolution

Arbre de décision. Les variables explicatives disponibles étant de nature qualitative pour certaines et quantitative pour d'autres, une première expérimentation a été menée en utilisant l'algorithme CART [2] pour construire un arbre de décision binaire. L'avantage de cette méthode d'estimation non-linéaire est sa simplicité et sa facilité d'interprétation.

On rappelle qu'un arbre de décision binaire est constitué d'un ensemble de nœuds. Tous les nœuds, à l'exception des nœuds terminaux de l'arbre, sont caractérisés par une condition binaire et exactement deux nœuds fils. Il s'agit donc d'un partionnement binaire récursif de l'espace des variables explicatives. Quant aux nœuds terminaux, ils sont étiquetés par une modalité de la variable à expliquer. Un exemple d'un tel arbre est donné à la page suivante.

L'algorithme CART se décompose en deux étapes. D'abord, on procède à la construction d'un arbre dit maximal au sens du nombre de nœuds terminaux par exemple, ou du nombre d'individus par nœuds terminaux. Ensuite, on extrait une sous-partition optimale du point de vue de la variable à expliquer.

La détermination des nœuds de l'arbre de décision, qui séparent récursivement l'espace des variables explicatives en deux, consiste à choisir une variable explicative et y associer une condition binaire : une valeur seuil dans le cas d'une variable quantitative et un partitionnement en deux dans le cas d'une variable qualitative. Pour être en mesure de faire ce choix, on définit l'impureté de Gini :

$$\Phi_t = \sum_{n=1}^{C} {p_t}^c (1 - {p_t}^c)$$

où C désigne le nombre de modalités de la variable à expliquer, t un nœud de l'arbre et ${p_t}^c$ la proportion d'individus de classe $c \in [\![1; C]\!]$. Un nœud a une impureté de Gini nulle s'il ne contient que des observations de la même classe.

La détermination d'un nœud se fait alors par le choix d'une variable explicative et d'un partitionnement / valeur seuil qui maximise la différence d'impureté de Gini entre le nœud et

ses fils. Cette différence d'impureté s'écrit :

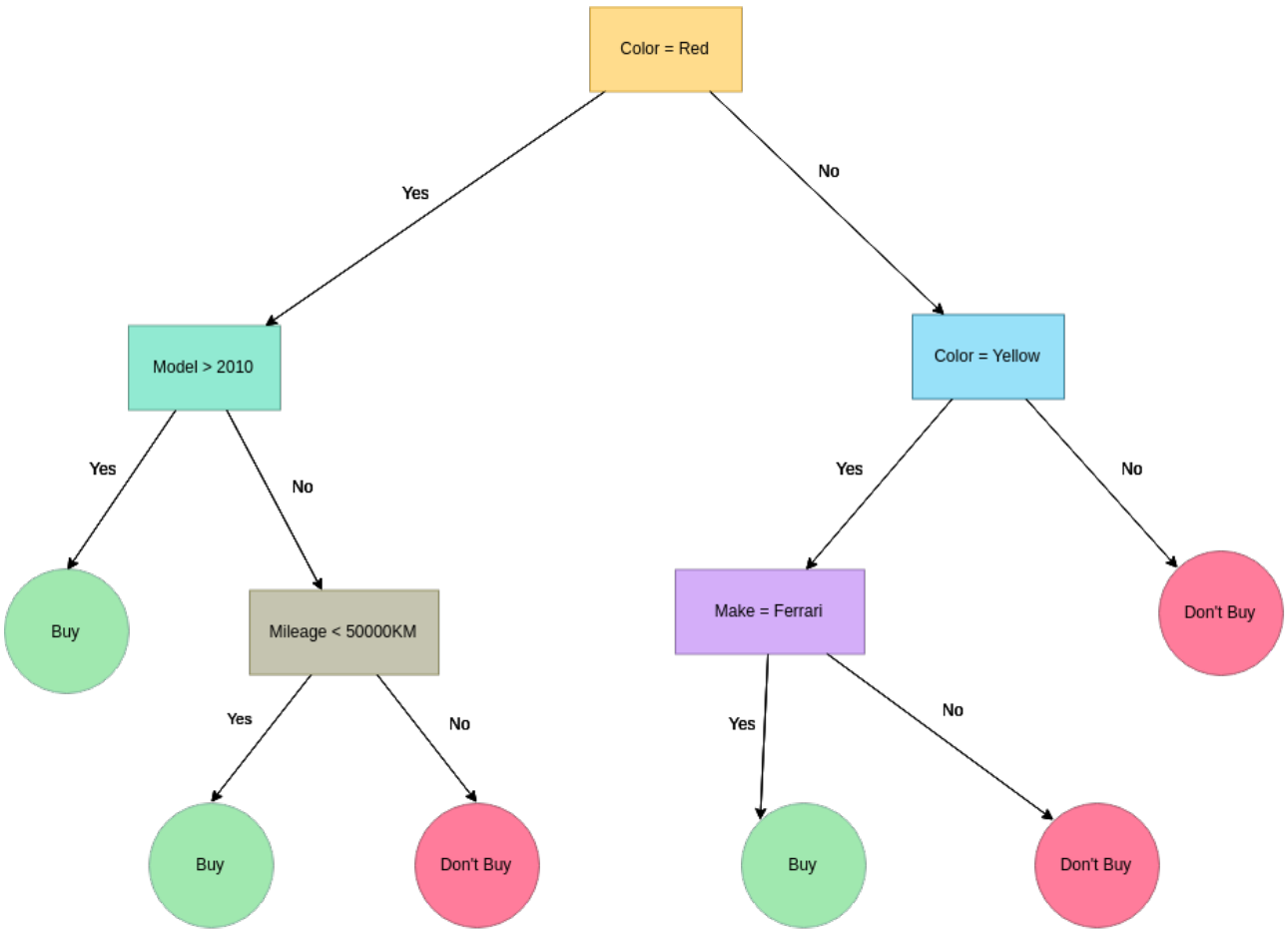
$$\Phi_t - (\frac{\#t_1}{\#t}\Phi_{t_1} - \frac{\#t_2}{\#t}\Phi_{t_2})$$

où t_1 et t_2 désignent les nœud fils du nœud t.

Ce procédé est répété récursivement sur chacun des nœuds fils d'un nœuds donné. L'arbre est ainsi développé jusqu'à atteindre une condition d'arrêt. La condition la plus simple est de ne pas découper les nœuds purs, i.e ceux qui ne contiennent que des individus de la même classe. La construction d'un arbre de décision aboutit nécessairement à l'existence de tels nœuds, quitte à ce qu'ils ne contiennent qu'un seul individu. Toutefois, cette pratique conduit généralement à des arbres inutilement "profonds" et complexes à calculer.

D'autres critères d'arrêt sont habituellement préférés : ne pas procéder au découpage d'un nœud s'il contient un nombre d'individus inférieur à un seuil donné, ou s'il entraîne une diminution d'impureté de Gini inférieure à une certaine quantité. Dans le cas de la classification des apporteurs, le seuil de vingt apporteurs par nœud terminal a été implémenté.

FIGURE 1 – Arbre de décision modélisant l'achat d'une voiture



L'arbre maximal — au sens de la condition d'arrêt — présente une variance importante et un faible biais ; il est sujet au phénomène de surapprentissage. Une deuxième phase dite d'élagage ou *pruning* est donc nécessaire pour en extraire un sous-arbre, meilleur que l'arbre maximal du point de vue de l'erreur de généralisation. Ce processus, qui correspond en fait à une sélection de modèle, est effectué par validation croisée [3].

Le package R *rpart* [3] a été utilisé pour l'arbre de décision de la cote des apporteurs. Le taux de précision sur échantillon test (prédictions correctes des cotes des apporteurs) obtenues avec cette méthode se situe entre 65 % et 70 % selon le mois testé.

Méthodes d'ensembles Deux algorithmes d'agrégation d'arbres de décisions ont été implémentés pour la classification des apporteurs :

- *Adaboost-SAMME*
- Forêts aléatoires

Adaboost-SAMME [4] est un algorithme de boosting dont les classifieurs faibles sont des arbres de décision simples. C'est une extension de la méthode de classification binaire *Adaboost* [5] adaptée aux classes multiples. L'algorithme *Adaboost* peut aussi être utilisé pour un nombre de classes $K > 2$, mais en décomposant le problème en plusieurs classifications binaires. Cela a pour conséquence de diminuer les performances de prédiction [4].

Soit $n \in \mathbb{N}$ et $(x_1, y_1), ..., (x_n, y_n)$ l'échantillon d'apprentissage. Pour tout $i \in [\![1; n]\!]$, x_i est le vecteur dont les coordonnées sont les valeurs des variables explicatives pour l'individu i et y_i sa classe : $x_i \in \mathcal{X}$, $y_i \in [\![1; K]\!]$. On présente ci-dessous l'algorithme Adaboost-SAMME [4] :

1. Initialiser les poids des observations $\omega_i = \frac{1}{n}, i \in [\![1; n]\!]$.

2. Pour $j \in [\![1; m]\!]$, $m \in \mathbb{N}$:
   (a) Entraîner un arbre de décision simple $\mathcal{A}^{(j)} : \mathcal{X} \to [\![1; K]\!]$ sur l'échantillon d'apprentissage *pondéré par les poids* w_i.
   (b) Calculer :
   $$\epsilon^{(j)} := \frac{\sum_{i=1}^{n} \omega_i \, \mathbb{I}[y_i \neq \mathcal{A}^{(j)}(x_i)]}{\sum_{i=1}^{n} \omega_i}$$
   (c) Calculer :
   $$\alpha(j) := \log(\frac{1 - \epsilon^{(j)}}{\epsilon^{(j)}})$$
   (d) Modifier les poids :
   $$\omega_i \leftarrow \omega_i \, e^{\alpha(j) \, \mathbb{I}[y_i \neq \mathcal{A}^{(j)}(x_i)]} \ , \ i \in [\![1; n]\!]$$
   (e) Normaliser les poids ω_i.

3. Renvoyer la valeur :
$$\tilde{y}_i = \underset{k \in [\![1;K]\!]}{\operatorname{argmax}} \sum_{j=1}^{m} \alpha^{(j)} \, \mathbb{I}[\mathcal{A}^{(j)}(x_i) = k] \ , \ i \in [\![1; n]\!]$$

À noter qu'il existe plusieurs méthodes pour pondérer les individus (étape 2.(a)). Cela peut être effectué directement sur l'échantillon d'apprentissage par duplication des individus associés à un poids important. La pondération peut également intervenir directement dans l'algorithme de calcul du classifieur faible. Dans le cas de l'arbre de décision simple, les poids interviennent dans le calcul de l'impureté de Gini (proportions d'individus d'une classe donnée dans un nœud).

Dans le contexte de la classification des apporteurs, le recours à l'algorithme *Adaboost-SAMME* a engendré une augmentation de la précision sur échantillon de test d'environ 5% par rapport aux arbres de décisions simples.

Une autre méthode d'ensemble d'arbres de décision, connue pour son efficacité dans un grand nombre de configurations, a été implémentée dans le but d'augmenter la précision des prédictions : l'algorithme des forêts aléatoires [6].

Le premier point consiste à entraîner un ensemble de $m \in \mathbb{N}$ arbres de décisions simples $\{\mathcal{A}^{(j)} : \mathcal{X} \to [\![1; K]\!] \mid j \in [\![1; m]\!]\}$ sur un échantillon boostrappé* de l'échantillon d'apprentissage (et de même taille). Le deuxième point est, pour chaque arbre, de sélectionner $mtry$ variables parmi les variables explicatives ($mtry \in [\![1; dim(\mathcal{X})]\!]$) lors de la construction d'un nœud donné. Cela permet notamment d'amoindrir la corrélation entre les arbres de décision simples construits sur échantillon boostrappé. En effet, si une poignée de variables explicatives sont des prédicteurs bien plus efficaces que d'autres de la variable de sortie, alors elles sont souvent sélectionnées pour déterminer un nœud : les arbres de décision simples qui en résultent sont similaires (forte corrélation) et l'information contenue dans les prédicteurs moins "puissants" est sous-utilisée.

La prédiction de la forêt aléatoire est le "vote majoritaire" des arbres de décision $\{\mathcal{A}^{(j)} : \mathcal{X} \to [\![1; K]\!] \mid j \in [\![1; m]\!]\}$ dans le cas d'une classification (La moyenne est généralement utilisée pour la régression).

Le choix du paramètre $mtry$ est une problématique importante. Dans le cas de la classification des apporteurs, le nombre de variables explicatives est faible (quinze), si bien que toutes les possibilités ont été testées par validation croisée. Plus précisément, la valeur retenue est celle qui minimise l'erreur de prédiction moyenne sur échantillon *Out Of Bag** : il s'agit des individus qui ne sont pas sélectionnés lors du boostrapping qui précède la construction de chaque arbre de décision simple.

Un autre paramètre important est le nombre d'arbres $ntree$ constituant la forêt aléatoire. L'augmentation de la valeur par défaut $ntree = 500$ n'a pas significativement diminué l'erreur sur échantillon de test, si bien que cette valeur a été conservée.

La précision des forêts aléatoires pour la classification des apporteurs est stable dans le temps (un modèle étant construit chaque mois) et se situe aux alentours des 85 %, soit 15 à 20 % de plus que l'arbre de décision simple. Cette performance a été jugée suffisante par les utilisateurs du modèle.

Une contrepartie majeure des bonnes performances des forêts aléatoires est l'absence — ou presque — d'explicabilité du modèle, inhérente aux méthodes d'ensembles. Une procédure existe néanmoins pour évaluer l'importance relative des variables explicatives [7]. Soit un arbre de décision $\mathcal{A}$ de la forêt aléatoire et V une variable explicative. L'erreur de prédiction ϵ_1 de l'arbre $\mathcal{A}$ est évaluée grâce aux individus $(x_1, y_1), ..., (x_p, y_p), 1 \leq p \leq n-1$ de l'échantillon *Out Of Bag**. L'erreur de prédiction ϵ_2 est évaluée une seconde fois après avoir permuté aléatoirement les valeurs de la variable V parmi les individus de l'échantillon *Out Of Bag*. La moyenne normalisée des écarts $|\epsilon_1 - \epsilon_2|$ de tous les arbres de la forêt aléatoires est calculée pour toutes les variables explicatives. Plus la valeur est importante, plus on estime que la variable explicative est importante dans la génération de la prédiction du modèle.

Utilisation du modèle Le fruit de la classification des apporteurs est un livrable produit mensuellement. Il s'agit d'une feuille de calcul qui présente la cote réelle et prédite de chaque apporteur sur les six derniers mois. Cette présentation permet d'identifier les suggestions de changement de cote proposées par le modèle ainsi que leur éventuelle récurrence au cours des mois précédents, ou au contraire les apporteurs pour lesquels le modèle valide la cote.

Le fichier présente également les valeurs des variables explicatives pour chaque apporteur. Une démarche empirique d'explication des décisions prises par le classifieur a également été

proposée. D'une part, les variables qui font l'objet d'une évolution importante par rapport au mois précédent sont signalées à l'utilisateur. D'autre part, on mesure l'écart entre les données de chaque apporteur et les valeurs moyennes de tous les apporteurs qui ont la même cote et appartiennent au même marché.

Pour chaque variable explicative, on présente la valeur ainsi que le 20-quantile calculé sur l'ensemble de la population. Ce sont en réalité les 20-quantiles qui ont été retenus pour construire le modèle. Cette décision est en partie liée à la pandémie de Covid-19. Le recours aux quantiles diminue légèrement les performances du modèle — 2 à 3 % — mais rend la classification plus stable dans le temps en cas de dégradation/amélioration uniforme du portefeuille. En effet ce type de phénomène n'a qu'un impact limité sur les quantiles relatifs aux individus.

2.2 Animation d'une communauté de données — Détection d'anomalies statistiques

2.2.1 DATAKO

Une communauté de données, DATAKO, a récemment été créée au *Centre de métier Leasing*. L'objectif est de réunir les personnes qui manipulent quotidiennement la donnée et qui en ont une connaissance spécifique, afin de faire de la veille technologique, du partage de connaissances, de bonnes pratiques et de mener des expérimentations autour de la statistique et de l'analyse de données.

La communauté a vocation à se réunir régulièrement lors de sessions mensuelles, ou plus régulièrement via le Réseau Social d'Entreprise. Le lancement du projet était initialement prévu le 1er avril 2020 et a dû être reporté. Néanmoins, l'équipe pilote s'est investi dans plusieurs projets liés à la visualisation des données et à la statistique.

2.2.2 Problématique

L'un des projets qui a été mené au sein de la communauté de données est la détection de potentielles anomalies parmi l'ensemble des contrats qui démarrent à un mois donné, en collaboration avec le département du Contrôle Permanent. On dispose d'environ une dizaine de variables sur chaque contrat : durée, taux, montant du bien, assurance, etc...

La détection de contrats en anomalie fait usuellement l'objet d'un jeu de règles qui est enrichi au fur et à mesure que de nouvelles anomalies sont identifiées manuellement. Ce processus est fastidieux et difficile à maintenir, sans être nécessairement fiable. Par ailleurs, il s'agit d'une démarche passive qui repose sur la détection d'anomalies existantes. L'approche statistique du problème a pour vocation d'apporter un éclairage objectif sur le sujet ainsi qu'une proactivité dans la recherche d'anomalies.

2.2.3 Recherche bibliographique

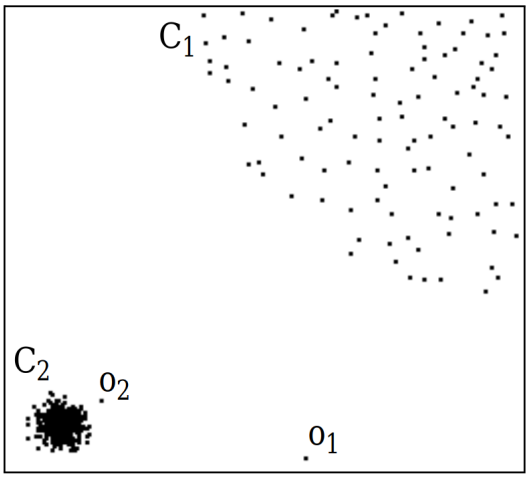
L'approche statistique qui a semblé la plus appropriée pour aborder la problématique de non-conformité d'un contrat est la détection d'*outliers*. Un *outlier* peut être défini de la façon suivante [8] : une observation (ou un ensemble d'observations) est qualifiée d'*outlier* lorsqu'elle *semble* incohérente par rapport au reste des données étudiées. L'utilisation du verbe "sembler" est importante car l'incohérence est seulement une éventualité : la donnée identifiée comme un *outlier* peut en effet provenir d'une distribution différente de celle qui a été choisie pour la modélisation, mais le modèle choisi peut tout aussi bien être inadapté au phénomène étudié.

On distingue plusieurs approches qui s'opposent selon différents axes dans le domaine de la détection d'anomalies [9]. On peut tout d'abord séparer les méthodes **globales** et **locales** [10], selon le périmètre des données sur lequel s'appuie une méthode pour évaluer l'anormalité d'un

individu. En effet, certains algorithmes prennent l'intégralité des données en compte, alors que d'autres n'utiliseront que les $k \in \mathbb{N}$ "plus proches" voisins. Il existe des méthodes qui attribuent un **label** aux individus — outlier / non-outlier — (méthode de clustering *DBSCAN* [11] par exemple) et d'autres qui génèrent un **score** d'anomalie pour chaque observation comme *Local Outlier Factor* [10] ou *Isolation Forest* [12]. Comme en statistique au sens large, on distingue les méthodes **paramétriques** et **non-paramétriques**, ainsi que les algorithmes **supervisés** et **non-supervisés** dans le même sens qu'en *Machine Learning*.

Dans le cadre de la problématique du Contrôle Permanent, plusieurs critères ont permis d'affiner la recherche d'un algorithme de détection d'anomalies. Tout d'abord, les méthodes **non-supervisées** ont été privilégiées car elles ne requierent pas de travail préliminaire sur les anomalies. De plus cet aspect rend possible la détection d'*outliers* qui n'auraient pas pu être détectés par des humains. D'autre part, l'algorithme devait être préférentiellement **dédié à la détection d'anomalies** et non au clustering, quand bien même cette technique peut servir à la détermination d'*outliers*. En effet, ces algorithmes sont optimisés pour la détermination des *clusters* et pas des *outliers* [10]. Ensuite, une méthode qui ne fournit pas un résultat binaire mais un **score d'anomalie** était souhaitée. Cela permet de générer une liste des contrats par *outlierness* décroissante, puis au métier d'examiner un nombre variables de contrats "suspects" selon leur besoin et leur disponibilité. De plus, les techniques **non-paramétriques** ont été ciblées. Le fait de ne pas avoir à déterminer la distribution des données a été perçu comme très avantageux dans la mesure où cette opération peut être ardue, et les résultats insatisfaisants d'autant plus que l'espace de variables explicatives est multidimensionnel. Enfin, il a semblé nécessaire de disposer d'une méthode **locale**. Cette caractéristique assure une détection satisfaisante dans des configurations comme celle-ci :

FIGURE 2 – Jeu de données en deux dimensions



Dans le cas présenté ci-dessus, C_1 / C_2 sont des *clusters* et O_1 / O_2 des *outliers*. Une méthode de détection d'anomalies globale déterminera que les outliers sont soit seulement O_1, soit O_1, O_2 et un grand nombre de points $\in C_1$ car la distance entre O_2 et C_2 n'est pas plus grande que la distance moyenne entre les points de C_2. En revanche, une méthode locale correctement paramétrée sera en mesure de détecter uniquement O_1 et O_2.

Ce point est important dans le contexte d'application de l'algorithme de détection d'anomalies, puisqu'une Analyse en Composantes Principales a mis en évidence des clusters nettement séparés. Cette typologie de contrat est liée entre autres à la nature des biens financés : le matériel informatique correspond à des montants et des produits financiers (taux, durée, etc...) bien différents de ceux des équipements de construction par exemple.

Plusieurs algorithmes satisfont l'ensemble des critères ci-dessus : Local Outlier factor [10], Isolation Forest [12] et réseaux de neurones récurrents [13]. LOF a été choisi pour des raisons de compatibilité logicielle, de simplicité d'utilisation, de puissance de calcul nécessaire et de facilité de paramétrage.

2.2.4 Local Outlier Factor [10]

Soient o, p et q des individus du jeu de données étudié, noté D. Dans le cas présent, ce sont des contrats caractérisés par les différentes variables explicatives disponibles. On suppose l'espace des variables explicatives muni d'une distance d.

Définition 1 : k-distance. Soient $k \in [\![1; \#D - 1]\!]$ et $p \in D$. Il existe $o \in D$ tel que :

1. $\#\{o' \in D \setminus \{p\} \,|\, d(p, o') \leq d(p, o)\} \geq k$
2. $\#\{o' \in D \setminus \{p\} \,|\, d(p, o') < d(p, o)\} \leq k - 1$

On définit alors la **k-distance** de p : $d_k(p) := d(p, o)$.

Remarque : la condition $k \in [\![1; \#D - 1]\!]$ assure l'existence de l'objet o dont il est question ci-dessus, mais en pratique $k << \#D$. Par ailleurs, l'objet o peut ne pas être unique. D'où la définition qui suit.

Définition 2 : k-voisinage. Soit $p \in D$. On définit le **k-voisinage** de p :

$$V_k(p) := \{q \in D \setminus \{p\} \,|\, d(p, q) \leq d_k(p)\}$$

Les objets q — tous à une distance inférieure à $d_k(p)$ du point p — sont alors appelés les k plus proches voisins de l'objet p.

Définition 3 : k-inaccessibilité. Soient $k \in [\![1; \#D - 1]\!]$ et $o \in D$. Pour tout $p \in D$, on définit la **k-inaccessibilité** du point p par rapport au point o :

$$\iota_k(p, o) := max\{d_k(o), d(p, o)\}$$

L'idée de la k-inaccessibilité est de "lisser" les fluctuations des distances entre le point o et ses points les plus proches, à savoir son k-voisinage. L'intensité de ce lissage est contrôlé par le paramètre k. Pour les points plus éloignés du point o, la k-inaccessibilité conserve la valeur réelle de la distance.

On fixe maintenant la valeur du paramètre $k \in [\![1; \#D - 1]\!]$, qu'on note $MinPts$. La notion de MinPts-inaccessibilité ne permet pas de comparer différents points entre eux car les ordres de grandeur peuvent grandement fluctuer sans rapport avec l'*outlierness* (cf figure 2 ci-dessus). Dans cette perspective, on définit la densité locale d'un point.

Définition 4 : Densité locale. Soit $p \in D$.

$$\rho_{MinPts}(p) := \left[\frac{\sum_{o \in V_{MinPts}(p)} \iota_{MinPts}(p, o)}{\# V_{MinPts}(p)} \right]^{-1}$$

La densité locale d'un point p, $\rho_{MinPts}(p)$, est donc l'inverse de la moyenne de sa MinPts-inaccessibilité par rapport aux points de son MinPts-voisinage $V_{MinPts}(p)$. Elle quantifie en quelque sorte l'accessibilité du point p par rapport à son voisinage.

Notons que la densité locale peut-être infinie si toutes les MinPts-inaccessibilités sont nulles dans la somme ci-dessus. Cela se produit lorsque $MinPts$ objets distincts entre eux et distincts du point p ont strictement les mêmes coordonnées que p, i.e si p possède au moins $MinPts$ doublons dans le jeu de données D. Ce problème n'est pas traité par la méthode LOF telle qu'elle est présentée ici. Ce problème peut être résolu en s'appuyant sur une autre définition de la k-distance (cf définition 1) qui exige que les objets o' soient tous de coordonnées distinctes dans la première condition.

Dans le cas où l'on souhaite conserver la définition initiale pour bénéficier des résultats ci-après, alors il convient de supprimer les doublons du jeu de données étudié ou d'introduire un bruit statistique pour dédoublonner les points concernés. C'est cette dernière méthode qui a été adoptée dans le cadre de l'application de la méthode LOF.

La dernière étape de l'algorithme LOF pour estimer l'*outlierness* d'un point p est de le comparer aux points de son MinPts-voisinage de point de vue de la densité locale.

Définition 5 : Local Outlier Factor.

$$\forall p \in D,\ LOF_{MinPts}(p) := \frac{\sum_{o \in V_{MinPts}(p)} \frac{\rho_{MinPts}(o)}{\rho_{MinPts}(p)}}{\#\, V_{MinPts}(p)}$$

Par exemple, si le point p est peu accessible par rapport aux points de son MinPts-voisinage, i.e sa densité est plus faible que celles des points de son voisinage, alors $LOF(p)$ est grand. On obtient bien une quantification locale de l'*outlierness* d'un point.

Théorème : Soit $C \in \mathcal{P}(D)$. Posons :

$$\epsilon := \frac{min\{\iota(p,q \,|\, p,\ q\ \in\ C)\}}{max\{\iota(p,q \,|\, p,\ q\ \in\ C)\}} - 1$$

Alors pour tout point $p \in C$ tel que les MinPts plus proches voisins de p et leurs MinPts plus proches voisins appartiennent à C :

$$\frac{1}{1+\epsilon} \leq\ LOF_{Minpts}(p)\ \leq 1+\epsilon$$

La preuve de ce théorème est disponible dans la publication originale [10]. Si C est un cluster "resserré", alors la valeur ϵ est faible si bien que pour les points qui sont profondément situés dans le cluster C, $LOF_{Minpts}(p) \approx 1$.

Détermination d'un sous-ensemble de valeurs pour le paramètre MinPts. Sur un échantillon gaussien $\mathcal{X}$ de grande taille, empiriquement la variance de la variable aléatoire LOF se stabilise à partir de MinPts = 10. D'autre part, on observe sur un échantillon uniforme que pour des valeurs inférieures à MinPts = 10, certains individus ont des LOF significativement supérieurs à 1. Cela signifie qu'ils sont caractérisés comme *outliers*. Or il est souhaitable que la méthode LOF ne mette en évidence aucun outlier si l'échantillon provient d'une distribution uniforme. Par conséquent, il est conseillé de sélectionner des valeurs $MinPts \geq 10$. Par ailleurs, on montre que la borne inférieure de MinPts peut-être vue comme le cardinal minimal d'un cluster [10]. D'une façon générale, $MinPts \geq m$ où $10 \leq m \leq 20$ fonctionne pour la plupart des applications [10].

La question de la borne supérieure pour la paramètre MinPts est plus difficile à trancher. Une interprétation de cette borne est le nombre maximal d'outliers vis-à-vis d'un cluster donné. En effet, si MinPts dépasse ce nombre, alors LOF caractérisera comme *outliers* les individus

d'un cluster adjacent. Il convient d'effectuer une analyse de données préliminaire pour faciliter la détermination de cette borne supérieure pour les valeurs de MinPts à tester.

Dans le cas présent, on estime que la quantité de contrats présentant des anormalités est faible a priori. D'autre part, la puissance de calcul limitée invite à sélectionner un nombre raisonnable de valeurs possibles pour MinPts. L'algorithme a donc été déployé pour $MinPts \in [\![10; 20]\!]$.

Le score d'*outlierness* final d'un individu est le maximum des LOF_{MinPts} des individus sur toutes les valeurs du paramètres MinPts qui ont été testées.

2.2.5 Conclusion

La production du mois de juin 2019 a été utilisée pour la phase de test. L'algorithme LOF a été déployé sur l'ensemble des contrats et les individus présentant les scores d'anormalité les plus importants ont été transmis au Contrôle Permanent. Cette expérimentation a été un succès puisqu'elle a permis d'identifier une non-conformité par rapport à la durée d'un contrat de leasing d'une pelle de chantier. Néanmoins, la plupart des contrats identifiés comme outliers étaient conformes. Il est probable que ce problème provienne de la période d'étude trop restreinte qui aboutit sur une diversité de contrats insuffisante. Par exemple, des contrats d'informatique ont obtenu un score LOF important en raison de leur montant élevé, sans qu'il n'y ait de problème de conformité associé. Cela tient au fait que ces gros contrats sont plus rares que les autres dans le domaine de l'informatique. Une perspective d'amélioration est d'augmenter la taille du jeu de données étudié pour densifier les différents clusters et rendre plus robuste la caractérisation des outliers. D'autre part, il est envisager de coupler la méthode purement statistique qui a été décrite ci-dessus à des règles métier existantes pour l'identification de contrats en anomalie.

2.3 Tarification des forfaits d'entretien automobile

2.3.1 Contexte de l'étude

Dans le cadre du leasing de véhicules, le *Centre de métier Leasing* propose au client un forfait d'entretien (payé mensuellement) qui peut comprendre plusieurs prestations : révision, changement de pneus, entretien de la climatisation, réparation en cas de panne, remplacement de véhicule, etc... Ce forfait est facturé mensuellement au client sur la base d'un tarif fixé lors de la signature du contrat.

Le point de départ de la tarification des forfaits d'entretien automobile est la diminution progressive de leur rentabilité. Cette tendance a été identifiée par une étude actuarielle qui a procédé par analyse comparée des charges et des loyers par année calendaire, sur la base des trois dernières années de production. Chez *Centre de métier Leasing*, une étude préliminaire a également été menée pour étudier l'évolution temporelle de la rentabilité. Une approche différente a été adoptée, à savoir l'étude des charges/loyers en fonction de l'âge contrat. Le problème qui se pose est l'inférence de la rentabilité à un âge de contrat a_c pour des véhicules qui sont financés depuis $a_v \leq a_c$ années.

Ces inférences ont été effectuées via des régressions linéaires multiples. Cette méthode a produit des résultats similaires à ceux de la première étude, venant confirmer une baisse de la rentabilité des forfaits d'entretien. Si aucune action n'est entreprise, alors la marge deviendra négative à l'horizon 2022 selon le modèle qui a été construit.

L'étude préliminaire a montré que le coût lié aux charges en fonction de l'âge du contrat est relativement stable dans le temps, i.e au fur et à mesure que les années calendaires se succèdent. Cela ne signifie pas pour autant que ces coûts sont stables. En effet, le portefeuille de *Crédit*

Mutuel Leasing est vieillissant, c'est-à-dire que la proportion de véhicules "âgés" augmente. Or, il a pu être vérifié que les charges et les coûts associés augmentent avec l'âge du véhicule, conformément à l'intuition. Ce phénomène provoque l'augmentation des charges d'entretien. Il peut en partie être expliqué par le caractère relativement récent de l'offre d'entretien automobile (environ dix ans) : la signature de nouveaux contrats sur des véhicules neufs n'a pas un potentiel illimité, et finit probablement par ne plus compenser le vieillissement des contrats signés précédemment.

En revanche, les loyers perçus par *Crédit Mutuel Leasing* à un âge de contrat fixé diminuent clairement en fonction du temps. Ce phénomène s'explique vraisemblablement par deux faits : d'une part le tarif des forfaits n'augmente plus depuis quelques années alors qu'il croissait d'environ 5 % par an auparavant, d'autre part des dérogations ont été accordées aux commerciaux ces trois dernières années pour abaisser les prix des forfaits d'entretien en dessous des tarifs en vigueur.

En réalité, l'objectif de cette mission est de modéliser les charges engendrées par le forfait d'entretien des véhicules en fonction de l'âge du contrat. Les experts métier s'appuieront ensuite sur cette étude pour déterminer une tarification appropriée en fonction de la marge souhaitée. Afin d'effectuer cette modélisation, on dispose d'environ 50 000 contrats signés entre 2010 et 2020. Les variables relatives aux interventions d'entretien sont le type (entretien au sens large / réparation / pneus), le coût et la date. D'autre part, des variables relatives aux contrats (données sur le véhicule, le client, etc...) sont utilisées pour construire le modèle :

Le forfait entretien vendu pouvant, ou non, couvrir différents types d'intervention (entretien/réparation/pneus), il est nécessaire de construire un modèle par type d'intervention pour être en mesure de les traiter indépendamment. L'objectif est de fournir à l'utilisateur une estimation des charges en fonction des valeurs des variables explicatives et en particulier de la durée du contrat. Une des problématiques majeures de cette mission est la variable explicative temporelle, à savoir l'âge du contrat. Il convient de prendre en compte la spécificité de cette information, ainsi que la présence de données censurées à droite* dans des proportions significatives. Il est inenvisageable que ces données ne soient pas prises en compte dans l'étude, puisqu'elles contiennent l'information la plus récente sur le portefeuille de *Crédit Mutuel leasing*. Dans ce contexte, deux méthodes sont mises en œuvre pour effectuer cette modélisation.

2.3.2 Première méthode : analyse d'événements récurrents

Une première stratégie est de scinder le problème en deux étapes :

1. Modélisation du nombre d'interventions en fonction du temps
2. Modélisation de la charge d'une seule intervention

Cette décomposition permet d'utiliser des méthodes d'analyse de survie [14] pour établir le premier modèle, particulièrement adaptée à la variable de nature temporelle ainsi qu'aux données censurées évoquées ci-dessus. Plus spécifiquement, le type de modèle adapté à la modélisation du nombre d'interventions en fonction du temps est dit "modèle d'événements récurrents" [15]. Ces méthodes estiment la fonction $t \to \mathbb{E}[N_t]$ où N_t est le nombre d'événements cumulés à la date $t \geq 0$.

Modèle de Cox. La première famille de modèles existants pour l'analyse d'événements récurrents se base sur le modèle de Cox [16] aussi appelé modèle à risques proportionnels. Destiné à modéliser la survenue d'un événement non-récurrent (mort, panne), il exprime le taux de défaillance instantané, à savoir $\lambda : t \to \lim_{\Delta t \to 0} \frac{1}{\Delta t} \mathbb{P}(T \leq t + \Delta t \,|\, T > t)$ où T est la

variable aléatoire de l'instant de défaillance, sous la forme :

$$\lambda(t, X_1, ..., X_n) = \lambda_0(t)\, e^{\sum_{i=1}^{n} \beta_i X_i}$$

où $t > 0$ et $X_1, ...X_n$ sont des covariables éventuellement dépendantes du temps. Les paramètres $\beta_1, ...\beta_n$ sont estimés par la méthode du maximum de vraissemblance [16]. Cette méthode est dite semi-paramétrique car aucune contrainte n'est imposée sur la forme de la fonction λ_0. Par contre, cette fonction est supposée être identique quelque soit l'individu considéré. il en découle que le rapport du taux de défaillance entre deux individus quelconques est indépendant du temps : c'est l'hypothèse des risques proportionnels. On vérifie que cette hypothèse est satisfaite par les données par la méthode des résidus de Schonfeld [17], définis pour la covariable $i \in [\![1; n]\!]$ et l'individu $j \in [\![1; m]\!]$, défaillant à l'instant t_j par :

$$R_{ij} := X_{ij}(t_j) - \sum_{k \in [\![1;m]\!] \setminus j, T_k > tj} X_{ik(t_j)}$$

.

En susbtance, il s'agit de comparer les caractéristique de l'individu j défaillant à l'instant t_j avec celles de tous les autres individus non-défaillants. Si l'hypothèse des risques proportionnels est vérifiée, alors ces résidus sont uniformément distribués en fonction du temps (ou plutôt des instants de défaillance de l'échantillon). Cette caractéristique peut être testée à l'aide d'un graphique ou d'un test d'adéquation du χ^2.

Le modèle de Cox, initialement conçu pour modéliser la survenue d'événements non-récurrents, a été adapté aux événements récurrents par Andersen & Gill [18]. Ce modèle fait l'hypothèse que la survenue d'événements multiples pour un individu donné est indépendante. Cette supposition est forte et sa vérification ardue ; malgré cela il a été décidé de mener une première expérimentation avec ce modèle en raison de sa facilité d'implémentation. Les résultats fournis par cette méthode sont satisfaisants en première approche. Ils sont présentés dans le tableau ci-dessous. L'analyse de cette application du modèle d'Andersen & Gill doit encore être approfondie, notamment concernant la distribution de l'erreur et l'identification des contrats qui sont mal pris en compte.

Nombre d'interventions

Type	Échantillon	Nombre	Erreur moyenne	Erreur relative moyenne
Entretien	App.	10367	0.01	0.30 %
Entretien	Test	2612	-0.03	-0.90 %
Réparation	App.	1586	0.04	2.90 %
Réparation	Test	401	0.04	2.94 %
Pneus	App.	870	0.16	5.57 %
Pneus	Test	220	0.15	-5.19 %

Modèles de fragilité [19]. Ce type de modèles introduit un effet aléatoire qui permet de prendre en compte d'une part l'hétérogénéité entre observations qui n'a pas été "capturée" par les covariables disponibles et d'autre part la dépendance entre événements successifs pour un individu donné. Pour cela, les observations sont séparées en groupes et le taux de défaillance est supposé s'écrire pour l'individu i du groupe j :

$$\lambda(t_{ij} \mid Z_i) = Z_i\, \lambda(t_{ij})$$

où Z_i est le facteur de fragilité pour le groupe i et λ le taux de défaillance de Cox explicité dans la partie précédente. Le facteur de fragilité est généralement modélisé de façon paramétrique, et le plus souvent par une distribution Gamma.

Un modèle de fragilité sera prochainement construit afin de confronter les résultats fournis par le modèle d'Andersen & Gill.

Modélisation de la charge unitaire d'une intervention. Suite à l'analyse d'événements récurrents, la modélisation du coût unitaire des interventions d'entretien, de réparation et de pneumatiques devra être effectuée. L'algorithme d'ensemble d'arbres de décision XGBoost [24] est envisagé pour ce problème de régression. La construction de ce modèle demande un travail de paramétrage important et inachevé à ce jour. Néanmoins, une première expérimentation a été menée. Mêmes si ses performances sont pour l'heure insuffisantes, ses résultats constitueront un point de départ pour évaluer la pertinence des méthodes qui seront utilisées utltérieurement. À noter que les deux modèles — nombre / coût — doivent être entraînés sur les mêmes échantillons d'apprentissage. On présente ci-dessous les estimations obtenues en combinant le modèle d'Andersen & Gill présenté précédemment et le prototype de modèle de coût unitaire d'intervention. On restreint le périmètre d'évaluation aux contrats **terminés**, car l'objectif du modèle est d'estimer les charges au terme de la durée de location.

Combinaison des deux modèles

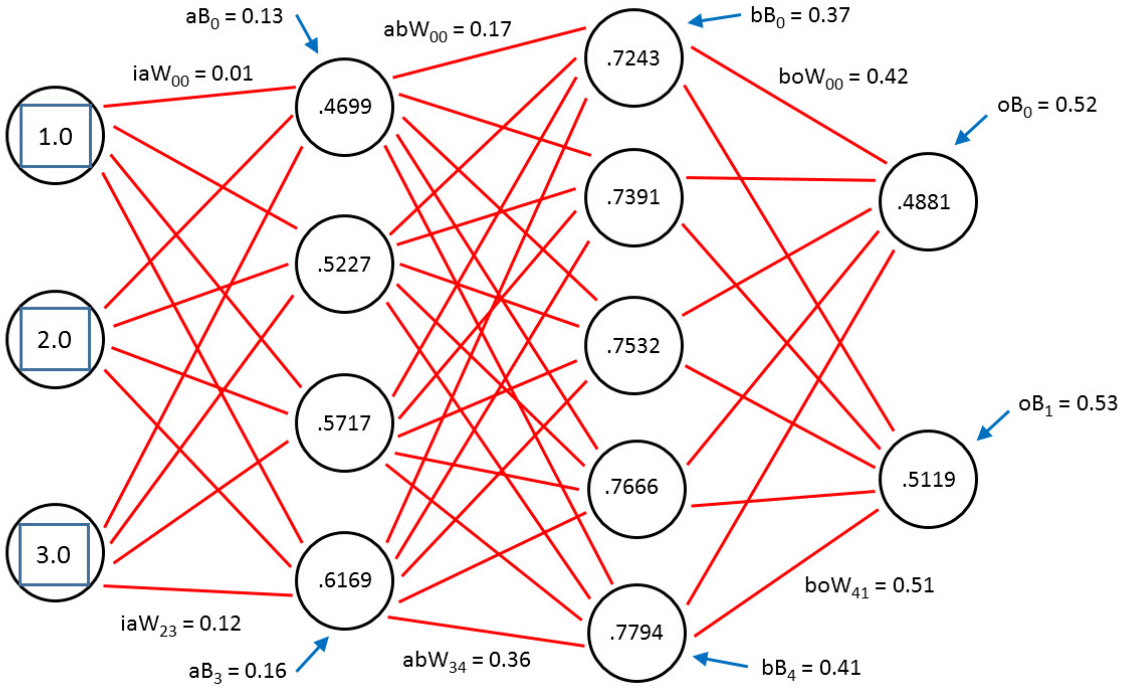
Type	Échantillon	Err. moy.	Err. rel. moy.	Err. moy. mensuelle
Entretien	App.	-19.40 €	-1.90 %	-0.10 €
Entretien	Test	-27.60 €	-2.70 %	-0.25 €
Réparation	App.	-15.20 €	-2.47 %	-0.12 €
Réparation	Test	-81.43 €	-12.28 %	-1.34 €
Pneus	App.	30.59 €	-5.66 %	-1.20 €
Pneus	Test	38.45 €	-6.84 %	-2.16 €

Néanmoins cette erreur n'est pas tout à fait centrée en 0 et surtout, sa distribution présente une variance importante. Il convient donc de comparer cette méthode à d'autres solutions envisagées.

2.3.3 Deuxième méthode : réseaux de neurones

Les réseaux de neurones artificiels, schématiquement inspirés de la structure cellulaire du cerveau, sont des ensembles d'unités de calculs appelées neurones formels. Les neurones sont reliés entre eux et leur état évolue en fonction du temps. Un réseau de neurones peut-être vu comme un graphe orienté pondéré dans lequel chaque noeud est un neurone formel :

FIGURE 3 – Représentation d'un réseau de neurones



Notons N le nombre de neurones du réseau et $w_{i,j}$ le poids de l'arc du neurone i vers le neurone j, où $i, j \in [\![1; N]\!]$. Le neurone $j \in [\![1; N]\!]$ est caractérisé à l'instant discret t par :

— Un signal d'entrée $in_j(t)$
— Une activation $a_j(t)$
— Un *seuil* θ_j
— Une *fonction d'activation* f_j
— Un signal de sortie $out_j(t)$. Le plus souvent, $out_j(t) = a_j(t)$.

Notons $W_j := (w_{1,j}, ..., w_{N,j})$ l'ensemble des poids des arcs vers le neurone j (certains d'entre eux peuvent être nuls s'il n'y a pas d'arc), $A(t)$ le N-uplet de toutes les activations du réseau et ϕ_j une valeur réelle éventuellement nulle appelée *biais* du neurone. Le **signal d'entrée** du neurone j à l'instant t s'écrit $in_j(t) := in(W_j, A(t), \phi_j)$. La quantité $in_j(t)$ peut donc éventuellement dépendre de l'activation du neurone j lui-même, $a_j(t)$. Si c'est le cas, les activations initiales des neurones concernés doivent être correctement initialisées pour que le calcul du signal d'entrée soit possible. La fonction in est souvent booléenne, linéaire ou affine si le biais $\phi_j \neq 0$.

La *fonction d'activation* f_j et le *seuil* θ_j jouent un rôle déterminant puisqu'ils permettent de déterminer l'**activation** du neurone j à l'instant $t+1$ via la relation : $a_j(t+1) = f_j(a_j(t), in_j(t), \theta_j)$.

À noter qu'un *neurone d'entrée* n'a aucun neurone qui le précède : son signal de sortie est la valeur que l'on souhaite communiquer au réseau. À l'inverse, un *neurone de sortie* n'a pas de successeur, son signal de sortie est la valeur (ou l'une des valeurs) calculée par le système.

D'après ce qui précède, un réseau de neurones peut être décrit comme une application $y = f(x, \Theta)$ où x est l'entrée, y la sortie et Θ l'ensemble des paramètres du réseau de neurones. Le plus souvent, l'entrée x est un vecteur de $\mathbb{R}^d$ où $d \in \mathbb{N}$: chacune de ses composantes est l'activation d'un neurone d'entrée. La sortie y, éventuellement multidimensionnelle, est la plupart du temps à valeurs réelles dans le cas d'une régression ou à valeurs discrètes dans le cas d'une classification. Cette méthode est donc adaptée au présent problème de la modélisation des charges engendrées par les forfaits d'entretien de véhicule. On peut noter qu'en choisissant

convenablement les fonctions d'activation de la couche de sortie, on peut imposer une contrainte sur la sortie y, comme par exemple $y \in [0;1]$ dans le cas du calcul d'une probabilité.

Θ est l'ensemble des paramètres du réseau de neurones, à savoir l'ensemble des poids $\{w_{i,j}|i,j \in [\![1;N]\!]\}$ et éventuellement des biais $\{\phi_j, j \in [\![1;N]\!]\}$. L'efficacité du réseau de neurones dans la tâche qu'on désire le voir accomplir est quantifiée par une fonction dite d'erreur ou de perte que l'on note $E(\Theta)$.

L'objet de l'*apprentissage* est la recherche d'un ensemble de paramètres Θ_0 tel que l'erreur $E(\Theta_0)$ soit la plus faible possible. Cette recherche s'effectue le plus souvent par un algorithme d'optimisation itératif :

1. Initialisation du paramètre Θ.
2. Calcul de l'erreur $E(\Theta)$ et des quantités relatives à la règle d'apprentissage.
3. Correction du paramètre Θ via la règle d'apprentissage.
4. Tant qu'un certain critère n'est pas satisfait, itération à l'étape 2.

Il existe de nombreuses règles d'apprentissage. L'une des plus connues est la méthode dite de rétropropagation du gradient qui cherche à minimiser le paramètre Θ par un algorithme de descente du gradient. D'autres règles s'appuient sur la méthode de Newton ainsi que ses variantes [20] ou encore l'ajout ou la suppression de neurones, la modification des fonctions d'activation, etc... Les règles d'apprentissage sont souvent spécifiques à un paradigme donné.

Dans le cadre de l'apprentissage supervisé, on dispose d'un ensemble de couple $\{(x_i, y_{d_i})|i \in [\![1;n]\!]$ où $n \in \mathbb{N}$, x_i est la valeur de la variable explicative et y_{d_i} la valeur de la variable à expliquer réelle/désirée associée à la valeur x_i. Soit $\mathcal{C}$ un sous-ensemble de $[\![1;n]\!]$. On définit la fonction d'erreur $E(\Theta, \mathcal{C})$ du réseau de neurones en comparant le signal de sortie $y_i = f(x_i, \Theta)$ à la sortie réelle/désirée y_{d_i} pour tout $i \in \mathcal{C}$. Formellement, la fonction de perte du réseau de neurones s'écrit $E : (\Theta, \mathcal{C}) \to E(\{(f(x_i, \Theta), y_{d_i})|i \in \mathcal{C}\})$. Il existe de nombreuses fonctions d'erreur, dont l'une des plus courantes est l'erreur quadratique dans le cas d'un signal de sortie réel.

Ce procédé peut conduire à un phénomène de sur-apprentissage : l'erreur $E(\Theta, \mathcal{C})$ est très faible mais $E(\Theta, [\![1;n]\!] \backslash \mathcal{C})$ est très élevée. Afin de garantir de bonnes capacités de généralisation, l'apprentissage doit être régulièrement suspendu afin d'estimer et de contrôler $E(\Theta, [\![1;n]\!] \setminus \mathcal{C})$. On parle de validation. Le réseau de neurone devra aussi être évalué en fin d'apprentissage sur un échantillon de test $\{(x'_i, y'_{d_i})|i \in [\![1;n']\!]\}$ différent de l'échantillon d'apprentissage.

Dans le cadre de l'application de cette méthode à la modélisation des charges d'entretien automobiles, il convient de prendre en compte plusieurs problématiques spécifiques à l'utilisation des réseaux de neurones.

Tout d'abord, en pratique il est difficile d'utiliser les variables explicatives catégorielles telles quelles compte tenu de la nature des fonction d'entrée des réseaux. Afin de contourner cette difficulté, on utilise la technique appelée *Hot One Encoding*. Cette méthode associe un axe réel à chaque modalité d'une variable qualitative donnée. Si l'individu possède l'attribut en question, alors il se voit affecter la valeur 1, sinon 0. Dans le cas présent, les variables comme la marque du véhicule possède un nombre considérables de modalités possibles, si bien que le *Hot One Encoding* augmente considérablement la dimension de l'espace des variables explicatives. Il est donc impératif d'anticiper ce qu'il est maintenant convenu d'appeler le "fléau de la dimension" (*curse of dimensionality*) [21].

Ensuite, il est nécessaire de normaliser les données, afin de supprimer l'influence du choix arbitraire des unités dans lesquelles sont exprimées les grandeurs physiques (âge, kilométrage, prix). Pour cela, après avoir séparé le jeu de données en un échantillon d'apprentissage et

un échantillon de test, la normalisation est effectuée en utilisant uniquement la moyenne et l'écart-type de l'échantillon d'apprentissage. De cette manière, aucune information provenant de l'échantillon de test n'est utilisée lors de la construction du modèle [22].

D'autre part, si les réseaux de neurones demandent des ressources de calcul importantes, ils s'appuient grandement sur l'algèbre linéaire qui se prête à la parallélisation [23]. Par conséquent, il convient de prêter une attention particulière à cet aspect. Des packages comme *doParallel* permettent de paralléliser les calculs sous R par exemple.

Enfin, les hyperparamètres* d'un réseau de neurones sont particulièrement nombreux : pas d'apprentissage, nombre de couches, nombre de neurones associés à chaque couche, fonctions d'activation et règle d'apprentissage. Par conséquent, la réalisation d'un *Grid Search** adapté est une étape déterminante pour assurer de bonnes performances prédictives.

Les résultats de cette méthode, lorsqu'ils seront disponibles, devront être mis en perspective avec ceux obtenus par l'approche fréquence-coût présentée précédemment.

2.4 Modélisation de la décote des matériels

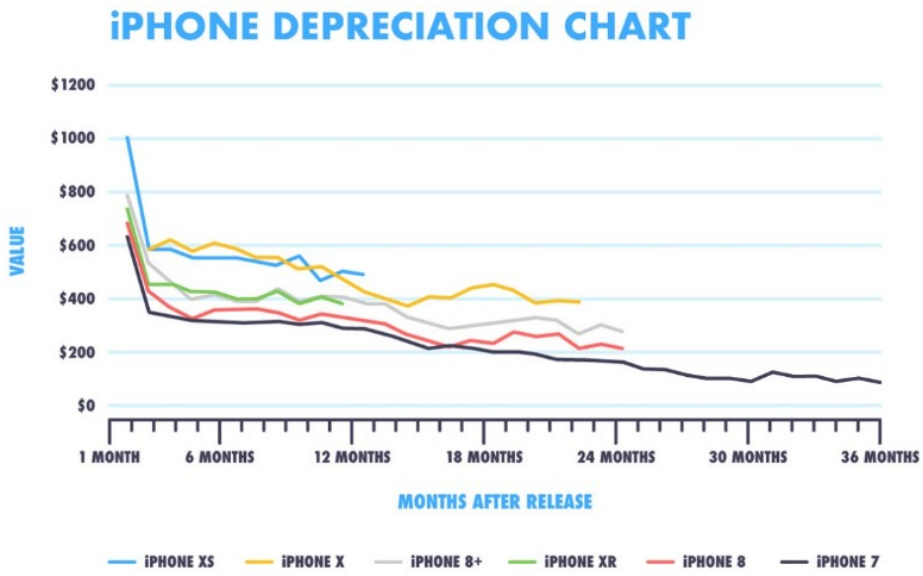
Première phase. Dans le secteur du *leasing*, un contrat est caractérisé notamment par sa durée, son taux d'intérêt, le montant des loyers (mensuels) et la valeur résiduelle . La valeur résiduelle correspond à la valeur contractuelle du bien en fin de contrat : si le client désire devenir propriétaire au terme de la durée fixée, il devra s'acquitter d'un montant égal à la valeur résiduelle. Par conséquent, plus la valeur résiduelle est élevée, moins le capital à rembourser par le client est important et donc, toutes choses égales par ailleurs, plus son loyer sera bas. La valeur résiduelle est donc une notion fondamentale lors de l'établissement d'une contrat en *leasing*.

Il est important de distinguer la valeur résiduelle de la valeur réelle. La valeur résiduelle est la valeur contractuelle du bien au terme du contrat, quelle que soit sa valeur réelle. Elle est fixée à l'avance, c'est-à-dire lors de la rédaction du contrat. Pour la déterminer, il est donc primordial de disposer d'une estimation convenable de la valeur réelle du bien au moment où le contrat arrivera à son terme. C'est ce point qui constitue l'enjeu majeur de cette mission : l'étude de la décote des matériels en fonction du temps.

Les matériels sont regroupés par familles de biens ayant un profil de décote similaire. La façon dont les matériels sont regroupés constitue en soi une problématique de clustering afférente à l'étude des décotes.

Pour débuter cette mission, un *backtesting* des décotes théoriques qui sont actuellement utilisées par *Crédit Mutuel Leasing* a été effectué. La base de travail est l'ensemble des ventes de matériels qui sont réalisées lorsque le contrat passe en contentieux, c'est-à-dire lorsque le client fait défaut de paiement. Ainsi, l'adéquation entre les décotes prévues par le *Centre de métier Leasing* et les décotes réellement observées a pu être évalué. Par ailleurs, un rapport sur ce sujet doit être remis deux fois par an aux autorités de régulation bancaire. Dans un premier temps, la rédaction de ce rapport réglementaire a été automatisée via l'outil *R Markdown* : le programme récupère les décotes théoriques ainsi que les cessions de matériels qui ont été faites et génère les données, graphiques et tableaux réglementaires permettant de jauger l'adéquation entre les valeurs réelles et théoriques.

FIGURE 4 – Exemple : fonctions de décote de plusieurs modèles d'*iPhone*



Cette étape est une première prise de contact avec les données du problème et la façon dont elles sont organisées au *Centre de métier Leasing*. D'autre part, ce document constitue maintenant le backtesting/suivi de performances des décotes théoriques. Cette étude a confirmé le besoin d'un nouveau modèle de décotes et formule les problématiques qu'il devra adresser :

— Le clustering sur le type de matériel : combien de types de décote et quelle classification ?
— Le découpage temporel du problème : année, mois voire jour ?
— La modélisation : modèle linéaire, non-linéaire, séries temporelles , algorithmes de Machine Learning ?

Décote des véhicules Ces questions vont d'abord être traitées dans un périmètre restreint au secteur de l'automobile. En effet, ce marché est spécifique puisqu'il concerne un volume particulièrement important et qu'il est d'une grande complexité en raison de la diversité des produits proposés. D'autre part, les processus métier du *Centre de Métier Leasing* sont distincts pour l'automobile et des difficultés sont actuellement rencontrées pour estimer correctement ce type de décotes en particulier. Enfin, l'accès aux données internes sur les véhicules n'est pas aisé car elles sont pour l'heure mal intégrées au système d'information du groupe. L'objectif est donc de proposer dans les meilleurs délais une estimation robuste de la valeur de décote réelle, en vue d'obtenir une meilleure visibilité sur la valeur résiduelle des contrats de véhicules neufs et d'occasion.

À l'heure actuelle, le *Centre de Métier Leasing* établit ses décotes sur la base de données achetées auprès de deux prestataires extérieurs. Le premier point sera donc la réalisation d'une étude comparative de ces données avec les ventes internes. L'enjeu est d'estimer l'adéquation de ces décotes avec le portefeuille de l'entreprise afin de décider s'il est souhaitable de conserver l'acquisition régulière de ces informations, et de quelle source le cas échéant. Ensuite, un modèle interne devra être construit en tenant compte des difficultés qui se présentent. La première d'entre elles est la question de la volumétrie des contrats qui doit être suffisante sur un modèle de véhicule donné pour pouvoir en déduire une fonction de décote. D'autre part, le biais de

sélection lié à une étude restreinte au portfeuille du *Centre de Métier Leasing* doit être pris en compte si des données venant de prestataires extérieurs sont injectées dans le modèle, car ces dernières sont représentatives du marché dans son ensemble. Enfin, l'intégration de véhicules qui n'existent pas actuellement (non-commercialisé à ce jour ou absent du portefeuille) nécessitera un examen approfondi des capacités prédictives du modèle de décote.

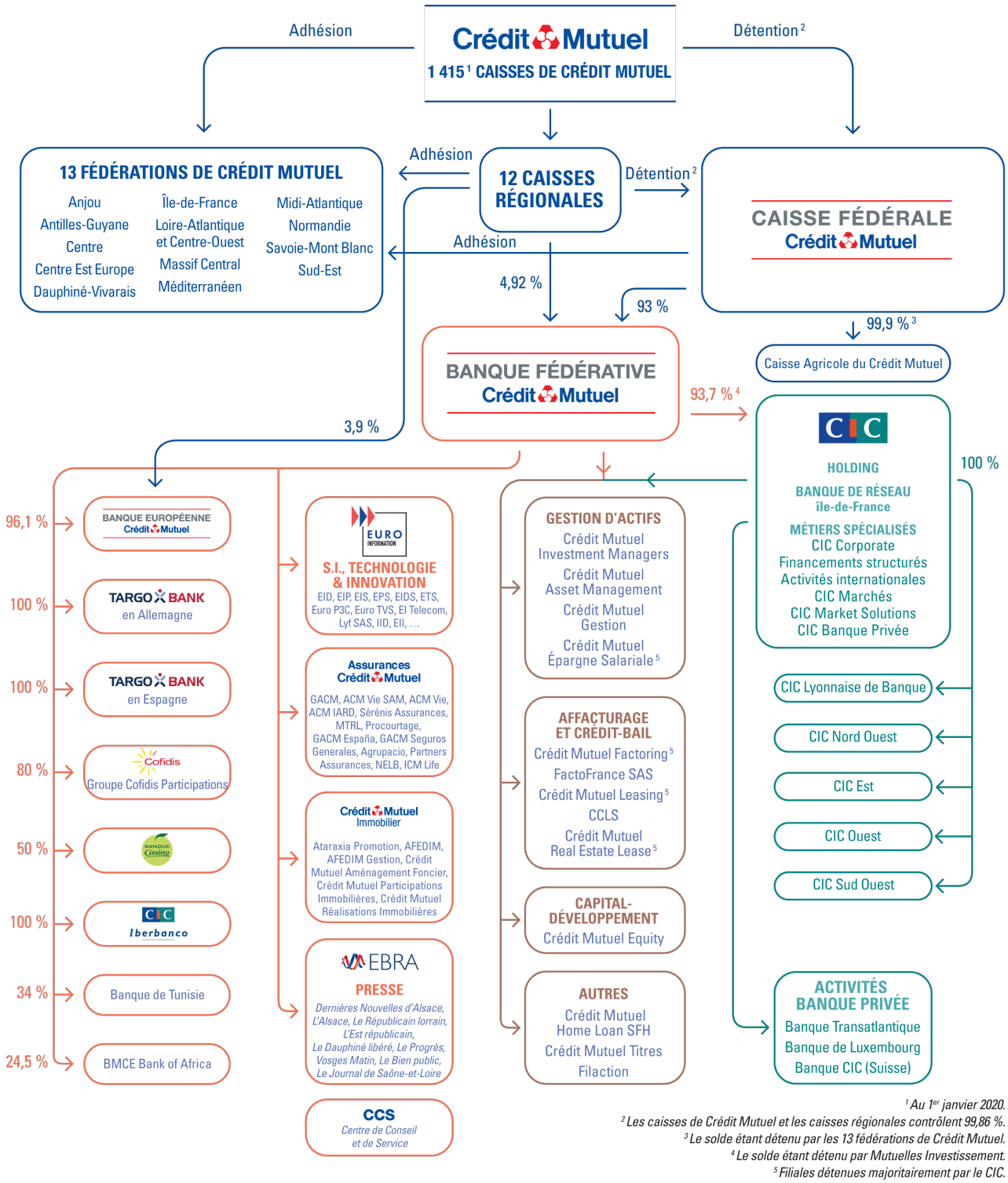
Des méthodes dites d'analyse de panel sont actuellement à l'étude pour procéder à cette modélisation. Ce sont des méthodes de régression adaptées aux données de nature simultanément transversale (plusieurs individus différents) et longitudinale (observations d'un même individu effectuées à plusieurs dates). On pourra également envisager des méthodes d'apprentissage automatiques. Néanmoins, ces méthodes ne sont pas nécessairement conçues pour prendre en compte la spécificité temporelle du problème étudié.

3 Conclusion

Les sujets abordés durant cette alternance portent sur des domaines de la statistiques très divers : détection d'anomalies, classification supervisée, clustering et problèmes de nature temporelle notamment. Dans ce contexte, cette année a été riche d'enseignements aussi bien à l'université qu'en entreprise et elle constitue une montée en compétence significative en tant qu'ingénieur statisticien. D'autre part, l'utilisation des outils liés aux langages *R* et *Python* a été l'occasion de progresser et de se mettre à jour sur le plan des technologies de la *Data Science*. Enfin, l'apprentissage du métier de la finance au sens large, de ses processus et de ses spécificités est un socle de connaissances qui sera très utile à l'avenir. Il faut également souligner le fait que cette alternance s'est déroulée dans un département qui comporte des profils très différents et complémentaires. Dans cette perspective, le fait d'intégrer l'équipe de Modélisation et Intelligence des Données a été une expérience particulièrement enrichissante sur le plan humain.

4 Annexes

4.1 Annexe A : Organigramme du groupe *Crédit Mutuel*



[1] *Au 1er janvier 2020.*
[2] *Les caisses de Crédit Mutuel et les caisses régionales contrôlent 99,86 %.*
[3] *Le solde étant détenu par les 13 fédérations de Crédit Mutuel.*
[4] *Le solde étant détenu par Mutuelles Investissement.*
[5] *Filiales détenues majoritairement par le CIC.*

4.2 Annexe B : Outils informatique

R. Une grande partie des travaux présentés dans ce mémoire ont été effectués avec le langage *R* et son environnement de développement *RStudio*. Il s'agit d'un logiciel gratuit et *open source* pour la statistique et la représentation graphique. Une de ses grandes forces est l'abandondance de *packages* (ensembles de fonctions) maintenus à jour et mis à disposition par le *Comprehensive R Archive Network* (CRAN) : *ggplot2* pour les graphiques, *nnet* pour le perceptron multi-couches, *rpart* pour les arbres de décision, *doParallel* pour la parallélisation des calculs sur plusieurs cœurs, *survival* pour l'analyse de survie, etc...

Les possibilités offertes par l'ensemble de packages *Tidyverse* ont été particulièrement exploitées pour traiter les données en amont du déploiement de méthodes de statistique.

FIGURE 5 – Le *Tidyverse*

Ces packages ont été conçu dans une optique d'ergonomie, de concision du code et de lisibilité. On peut citer le *pipe*, noté %>%, qui permet d'écrire lisiblement une succession de fonctions s'appliquant à un objet donné. Par exemple, l'application des fonctions f, g, et h à un objet x s'écrira :

```
x %>% f() %>% g() %>% h()
```

plutôt que :

```
h(g(f(x)))
```

Dans le cadre de la *Data Science* où de nombreuses opérations sont effectuées en cascade sur le jeu de données considérée, le recours au *pipe* améliore significativement la clarté du code. Toujours dans le *Tidyverse*, le package *purrr* permet d'appliquer des fonctions anonymes à la volée dans une séquence de traitement de données. Combiné à l'utilisation du *pipe*, cela ajoute également une grande concision au code. Le package *stringr* permet de traiter simplement les chaînes de caractères à tel point que ses fonctions ont été intégrées à *Base R*, l'ensemble des fonctionnalités de base du logiciel. Le package *forcats* est dédié à la gestion des variables catégorielles. On peut citer le regroupement de modalités insuffisamment représentées, le renommage de modalités ou le traitement des valeurs manquantes : le remplacement par une moyenne ou par un tirage selon la distribution observée se font très simplement.

Le recours au *Tidyverse* a ainsi permis de réduire la phase de traitement et de mise en forme des données qui précède nécessairement l'utilisation de méthodes et d'algorithmes statistiques. L'autre avantage qui a été constaté est la supériorité technique des fonctions du *Tidyverse* sur le plan de la rapidité d'exécution, que ce soit pouur l'import de fichiers au format .csv ou la manipulation de tableaux de données au sens large.

Python. *Python* est un langage de programmation particulièrement adapté à la statistique car il permet l'utilisation de quelques modules *open source* et de grande qualité. On peut citer le module libre *Scikit-learn* qui implémente un grand nombre d'algorithmes d'apprentissage automatique. Il est l'initiative de chercheurs de l'INRIA qui continuent de le développer dans une large mesure. Son objectif est d'harmoniser l'utilisation des fonctionnalités statistiques tout en s'appuyant sur les modules scientifiques *NumPy* et *SciPy*. Certaines méthodes comme *Isolation Forest* — adaptation de l'algorithme des forêts aléatoires pour la détection d'anomalies— sont disponibles seulement sous *Python* et pas sous *R*. D'autres modules comme *Keras* offrent des possibilités de paramétrage très riches pour les réseaux de neurone : nombre de couches, de neurones par couches, fonction d'activation, seuils, méthode d'apprentissage, etc...

Le *DataLab*. Il s'agit d'un serveur mis à disposition par la filière informatique du groupe *Crédit Mutuel*, *Euro Information*. Il propose un environnement de développement particulièrement pensé pour *Python* et les modules afférents à la statistique et au traitement de données, ainsi qu'une puissance de calcul importante. Cela permet de mettre à profit l'efficacité d'éxecution du langage *Python*, accompagné de l'outil *Jupyter Notebook* pour la présentation simultanée des résultats et du code. Le *DataLab* met régulièrement à jour l'offre de modules proposée, et proposera bientôt *PyTorch* pour les réseaux de neurones à structure dynamique.

Au-delà de la puissance des machines considérées individuellement, le *DataLab* donne la possibilité de distribuer les calculs sur de nombreuses machines pour le traitement de très gros volumes de données. Ce travail se fait via le framework *Spark* interfacé avec le système de fichiers distribué *Hadoop*.

5 Lexique

Échantillon boostrappé : échantillon constitué par des prélèvements avec remise de l'échantillon d'origine.

Out Of Bag : Les observations dites *out of bag* sont celles qui n'ont pas été sélectionnées lors de la constitution d'un échantillon boostrappé.

Donnée censurée à droite : pour un contrat de leasing par exemple, cela désigne le fait de ne pas disposer des informations au-delà d'une certaine date, alors que le contrat n'est pas terminé.

Hyperparamètre : paramètre d'un algorithme dont la valeur est fixée avant le début du processus d'apprentissage.

Grid Search : désigne la construction d'un ensemble de modèles du même type mais d'hyperparamètres différents afin de sélectionner le meilleur selon un critère d'évaluation donné.

Taux de charge : dans le contexte de la modélisation des charges d'entretien automobile, il s'agit du ratio $\frac{Charges\ d'entretien}{Prix\ du\ véhicule}$.

RMSE : *Root Mean Squared Error*. Dans le cadre d'un régression i.e la modélisation d'une variable quantitative Y, il s'agit de la quantité $\sqrt{\sum_{i=1}^{n}(\hat{Y}_i - Y_i)^2}$ où $\hat{Y}_i$ (resp. Y_i) est la valeur réelle (resp. prédite) associée à l'individu i , $1 \leq i \leq n$.

Fonction anonyme : fonction définie à l'endroit même où elle est appelée, n'ayant à ce titre nul besoin d'être nommée.

Références

[1] LE CUN Yann.
https://fr.wikipedia.org/wiki/Yann_Le_Cun

[2] BREIMAN, L., FRIEDMAN, J. H., OLSHEN, R. A., AND STONE, C. J. *Classification And Regression Trees*. Chapman & Hall, 1984.

[3] THERNEAU, Terry M., ATKINSON, Elizabeth J., *An Introduction to Recursive Partitioning Using the RPART Routines*. Mayo Foundation, 2019.
https://cran.r-project.org/web/packages/rpart/vignettes/longintro.pdf

[4] FREUND, Yoav, SCHAPIRE, Robert E., *Experiments with a New Boosting Algorithm*. AT&T Laboratories, 1996.

[5] ZHU, Ji, ROSSET, Saharon, ZOU, Hui, HASTIE, Trevor *Multi-class AdaBoost*. Stanford University, 2006.

[6] BREIMAN, L., *Manual On Setting Up, Using, And Understanding Random Forests V3.1*, 2002.
https://www.stat.berkeley.edu/ breiman/Using_random_forests_V3.1.pdf

[7] LOUPPE, G., WEHENKEL, L., SUTERA, A., GEURTS, P., *Understanding Variable Importances In Forests Of Randomized Trees*. Dept. of EE & CS, University of Liège, Belgium.

[8] BARNETT, V., LEWIS, T., *Outliers in Statistical Data*, 1994. Troisième édition, John Wiley & Sons.

[9] ZIMEK, Arthur ; FILZMOSER, Peter, *There and back again : Outlier detection between statistical reasoning and data mining algorithms*, 2018. Wiley Interdisciplinary Reviews : Data Mining and Knowledge Discovery.

[10] KNORR, E. M., NG, R. T. : *Algorithms for Mining Distance-Based Outliers in Large Datasets"*. Proc. 24th Int. Conf. on Very Large Data Bases, New York, NY, 1998, pp. 392-403.

[11] ESTER, Martin, KRIEGEL, Hans-Peter, SANDER, Jiirg, XU, Xiaowei : *A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise*. Institute for Computer Science, University of Munich, 1996.

[12] LIU, Fei Tony, TING, Kai Ming and ZHOU, Zhi-Hua : *Isolation forest*. Data Mining, 2008. ICDM 2008. Eighth IEEE International Conference on Data Mining.

[13] HAWKINS, Simon, HE, Hongxing, WILLIAMS, Graham and BAXTER, Rohan : *Outlier Detection Using Replicator Neural Networks*. CSIRO Mathematical and Information Sciences, 2002.

[14] KLEIN, John P., MOESCHberger, Melvin L. : *Survival Analysis. Techniques for Censored and Truncated Data*. Springer, 2003.

[15] COOK, Richard J., LAWLESS, Jerald : *The Statistical Analysis of Recurrent Events*. Springer, 2007.

[16] COX, D. R. : textitRegression Models and Life-Tables. Journal of the Royal Statistical Society. Series B (Methodological), Vol. 34, No. 2(1972), pp. 187-220

[17] SCHOENFELD, D. : *Partial Residuals for The Proportionnal Hazards Regression Model*, 1982. Biometrika, vol. 69, p. 239-241.

[18] ANDERSEN, P. K., GILL, R.D : *Cox's Regression Model For Counting Processes : A Large Sample Study*. Statistical Research Unit, Copenhagen, Mathematical Centre, Amsterdam, 1982.

[19] MONACO, J. V., GORFINE, M., HSU, L., *General Semiparametric Shared Frailty Model : Estimation and Simulation with* **frailtySurv**.. Jounal of Statistical Software, Volume 86, Issue 4, August 2018.

[20] Alberto Quesada, *neuraldesigner*.
https://www.neuraldesigner.com/blog/5_algorithms_to_train_a_neural_network

[21] YERESSIAN, K., *Overcoming The Curse Of Dimensionality In Neural Networks.*
https://arxiv.org/pdf/1809.00368.pdf

[22] Sebastian Raschka, Why do we need to re-use training parameters to transform test data ?
https://sebastianraschka.com/faq/docs/scale-training-test.html

[23] Richard P. Brent, 1991, Parallel Algorithms in Linear Algebra. *Australian National University*
https://arxiv.org/pdf/1004.5437.pdf