



MÉMOIRE DE STAGE

---

# Analyse des données assurantielles

---

BOUSSENGUI FRANÇOIS

Promotion 2021

*Entreprise*

APRIL MOTO

M. BLANVILLAIN ADRIEN

*Université de Tours*

M. PERROLLAZ VINCENT

6 Avril 2021 au 30 Septembre 2021

# Note de synthèse

---

Cette synthèse présente les travaux réalisés dans le cadre d'un stage de six mois à APRIL Moto au sein du service informatique.

Les travaux réalisés durant ce stage sont divers. La première mission concerne l'amélioration de l'outil de veille concurrentielle présent dans l'entreprise. La deuxième mission quant à elle consiste à la mise en place d'un outil prédictif dans le cadre de la prévention de l'attrition.

La première partie de ce mémoire traite de l'amélioration de l'outil de veille concurrentielle en place dans l'entreprise. En effet, APRIL moto est une entreprise spécialisée dans la conception, la gestion et la distribution des produits d'assurance adaptés aux détenteurs de deux-roues. Elle exerce son activité dans un secteur très concurrentiel et qui possède ses particularités par rapport aux autres domaines que peut concerner l'assurance. Par voie de conséquence, il est indispensable pour APRIL Moto de rester aux faits de l'activité de ses concurrents. C'est l'objectif de cet outil de veille concurrentielle. La méthode principale utilisée pour le fonctionnement de cet outil est le web-scraping. C'est une technique permettant l'extraction de données sur un site internet. Dans le cas d'APRIL Moto, les données sont extraites de sites internet de comparateurs d'assurance tels que LesFurêts ou encore SollyAzar.

Cet outil est donc un robot de web-scraping qui fonctionne quotidiennement. Il permet le remplissage du formulaire de questions concernant le conducteur ainsi que son véhicule présent sur les sites des comparateurs. A la fin du remplissage de ce formulaire, des formules tarifaires sont proposées. Elles sont ensuite récupérées par la robot et sauvegardées sur un serveur dans un fichier au format CSV et mises en valeur grâce à une application de business intelligence.

L'objectif principal de la mission consiste donc à faire en sorte que le robot puisse récupérer les tarifs proposés par les concurrents pour tous les scénarios définis. Un scénario peut être défini comme un profil de personne. En d'autres termes, c'est un conducteur de deux-roues avec des caractéristiques définies au préalable. En effet, pour récupérer des tarifs pertinents pour la prise de décisions et le calcul d'indicateurs, 50 scénarios ont été définis. Ils représentent différents cas pertinents que l'on peut avoir sur les principaux segments du marché de l'assurance deux-roues à savoir : les scooters, les cyclomoteurs et les motos. Par conséquent, le bon remplissage des formulaires pour avoir les bons retours de tarifs est une chose cruciale. Pour que cela se face correctement, l'amélioration du script Python de web scraping permettant le remplissage du formulaire a été indispensable. Ce travail a constitué la majeure partie de la mission. Après avoir effectué ce travail, il a été question d'assurer l'automatisation de l'outil de sorte à ce qu'il récupère quotidiennement les tarifs que proposent les concurrents sur les sites des comparateurs.

Puis il a été question de l'utilisation de ces données. En effet, pour que ces données soient utiles à la prise de décision, un outil de Business Intelligence a été créé. Grâce au logiciel Qlik Sense, une application qui synthétise les informations récupérées permet d'avoir connaissance des évolutions tarifaires du marché. Cette application est alimentée quotidiennement avec les données récupérées par le robot de web scraping. Elle permet également d'avoir des informations sur la performance du robot.

S'agissant de la maintenance du robot, il faut donc s'assurer que ce dernier fonctionne correctement quotidiennement et dans le cas échéant, trouver ce qui n'a pas fonctionné grâce à des fichiers « .log » permettant de retracer le parcours effectué par le robot pour chaque scénario. Une erreur peut être par exemple, le mauvais remplissage d'une étape du formulaire avec une information erronée. Dans ce cas, soit les tarifs récupérés ne sont pas ceux attendus, soit le robot ne parvient pas à aller jusqu'à la fin du remplissage du formulaire. De même, des mails quotidien permettant d'attester le bon fonctionnement du

---

processus de récupération et de sauvegarde des données sont envoyés aux différentes personnes concernées. Ils permettent d'être tenu en alerte en cas de problème.

Les captchas constituent un obstacle au bon fonctionnement du robot. Pour limiter l'impact de ces derniers, une anomysation de l'adresse IP en utilisant le logiciel TOR a été mise en place.

Concernant la deuxième mission, il a été question d'établir un modèle prédictif dans le cadre la prévention de l'attrition. La prévention de l'attrition constitue un enjeu majeur pour les entreprises offrant des services sous forme de contrat liant l'entreprise au client sur une durée qui peut ne pas être au préalable défini. C'est le cas du secteur de l'assurance pour les deux-roues dans lequel exerce APRIL moto. En effet, quand un client souscrit à un contrat d'assurance chez APRIL Moto, ni l'assureur ni le client ne sait à quel moment et pourquoi prendra fin le contrat. Le point central est donc de pouvoir anticiper le départ des clients définis comme étant à fort risque de résiliation. Pour ce faire, il a été question d'utiliser une méthode classification supervisée à savoir l'algorithme de Boosting de Gradient disponible dans le package `tidymodels` du logiciel R.

Dans un premier temps, il a été question de se mettre d'accord sur les variables utiles à la modélisation du problème. Après cela, les données ont été récupérées grâce à des requêtes SQL sur les bases de données internes de l'entreprise. Nous avons donc récupérés les informations de 87042 contrats actifs ou résiliés sur la période du 1-06-2020 au 1-09-2020. Il a fallu ensuite nettoyer les données pour résoudre le problème des données manquantes, des données erronées ainsi que la transformation de certaines variables. Ce nettoyage nous a permis d'obtenir une base de données en bonne et due forme contenant 78019 contrats et 38 variables. Cela nous a permis d'effectuer une analyse exploratoire sur nos données avant de passer à la phase de modélisation. Durant la phase d'apprentissage nous avons donc entraîné le modèle sur 75 % des données et attesté les résultats sur les 25 % restant de l'échantillon. Après entraînement et validation des résultats de notre modèle, nous l'avons appliqué sur de nouvelles données. Nous avons donc pris un panel de contrats actifs au 1-06-2021 sur lequel nous avons appliqué le modèle prédictif. L'algorithme attribue une probabilité donnant le niveau de risque de résiliation du client. La finalité est donc de pouvoir confronter les résultats des prédictions à la réalité afin d'attester de la pertinence du modèle et voir quelles sont les perspectives d'amélioration.

## Remerciements

---

Avant de commencer le développement de cette expérience professionnelle, il me paraît tout naturel de commencer par remercier les personnes qui m'ont permis d'effectuer ce travail ainsi que ceux qui m'ont permis d'en faire un moment agréable et profitable.

Tout d'abord, je tiens à remercier Monsieur PERROLLAZ Vincent, enseignant à l'Université de Tours, pour avoir accepté d'être mon tuteur, pour son suivi tout au long de la période de stage ainsi que ses précieux conseils quant à la réalisation de ce rapport.

Ensuite, je remercie tout particulièrement Monsieur BLANVILLAIN Adrien, Lead Data-scientist au sein de APRIL MOTO pour m'avoir offert la possibilité de faire ce stage au sein de cette entreprise, de m'avoir fait confiance, de m'avoir transmis son savoir et d'avoir été présente tout au long de la durée de ce stage.

# Table des matières

---

|          |   |           |
|----------|---|-----------|
| <b>1</b> | <b>Introduction</b>                           | <b>6</b>  |
| <b>2</b> | <b>Le groupe APRIL</b>                        | <b>8</b>  |
| 2.1      | APRIL MOTO . . . . .                          | 8         |
| 2.2      | Le marché de l'assurance deux-roues . . . . . | 9         |
| 2.2.1    | Des facteurs différenciant . . . . .          | 9         |
| <b>3</b> | <b>Outil de veille concurrentielle</b>        | <b>11</b> |
| 3.1      | Présentation et objectifs . . . . .           | 11        |
| 3.2      | Processus d'exécution . . . . .               | 12        |
| 3.3      | Résultat . . . . .                            | 14        |
| 3.4      | Limites . . . . .                             | 14        |
| 3.4.1    | Les captchas . . . . .                        | 14        |
| 3.4.2    | Le débit internet . . . . .                   | 15        |
| 3.4.3    | Facteurs exogènes . . . . .                   | 15        |
| <b>4</b> | <b>Prévention de l'attrition</b>              | <b>16</b> |
| 4.1      | Problématique . . . . .                       | 16        |
| 4.2      | Les données . . . . .                         | 16        |
| 4.2.1    | Mise en forme de la base de données . . . . . | 16        |
| 4.3      | Analyse exploratoire . . . . .                | 17        |
| 4.4      | Modélisation . . . . .                        | 21        |
| 4.4.1    | Résultats . . . . .                           | 23        |
| 4.4.2    | Prédictions . . . . .                         | 25        |
| 4.5      | Actions concrètes . . . . .                   | 25        |
| 4.6      | Perspectives et améliorations . . . . .       | 25        |
| <b>5</b> | <b>Conclusion</b>                             | <b>27</b> |
| <b>A</b> | <b>Annexe</b>                                 | <b>28</b> |
| <b>B</b> | <b>Bibliographie / Sitographie</b>            | <b>34</b> |

# 1

## Introduction

---

Un automobiliste est un conducteur à peu près toute sa vie, du jour où il décroche son permis de conduire jusqu'à la vieillesse l'éloignant du volant. Cependant, l'activité d'un motocycle est moins linéaire : entre le cyclomoteur de son adolescence et la grosse cylindrée que l'on s'offre à 50 ans, il peut y avoir de nombreuses périodes sans deux-roues pour diverses raisons.

Concevoir des produits d'assurance adaptés à ce type de bien ne s'avère donc pas être une tâche facile.

Afin de rester compétitif sur le marché de l'assurance deux-roues, le grossiste courtier en assurance qu'est APRIL Moto s'est fixé des objectifs clairs qui sont les mêmes que celui du groupe APRIL dont elle est une filiale : devenir un acteur digital et champion de l'expérience client.

Ces objectifs s'inscrivent dans les exigences que le monde dans lequel nous vivons aujourd'hui nous impose. La transformation digitale désigne le passage d'une économie essentiellement matérielle, s'appuyant sur des points de ventes physiques à une économie dématérialisée s'appuyant des échanges massifs de données. Ce changement de paradigme permet de gommer les barrières physiques, facilite l'accès à l'information, multiplie le volume de données disponibles et change les modes de consommation et de communication.

L'expérience client quant à elle regroupe la perception et les émotions qu'une organisation procure à ses clients lorsqu'ils entrent en contact avec elle. Afin de rendre cette expérience unique et singulière, la bonne exploitation des données reste donc primordiale et ouvre la voie à l'élaboration d'une offre personnalisée.

La transformation digitale et l'amélioration de l'expérience client sont donc très liées. Exploiter la donnée pour la rendre accessible, créatrice de valeur ajoutée et être compétitif est un enjeu que APRIL Moto a su appréhender. Mieux connaître ses clients grâce à une utilisation optimale des données pour leur offrir la meilleure expérience client qui soit est le fil conducteur du travail effectué lors du stage et rapporté dans ce mémoire.

Dans un premier temps il s'agira de présenter l'environnement du stage, le groupe APRIL et son offre de manière générale puis plus particulièrement sa filiale APRIL Moto. Une définition de ce qu'est un courtier grossiste en assurance ainsi que les particularités du marché de l'assurance deux-roues seront présentées. Dans un deuxième temps, il s'agira de présenter le déroulement et les missions effectuées tout au long de ce stage.

## Contexte

Le stage s'est déroulé au sein du groupe APRIL et plus particulièrement dans sa filiale APRIL Moto basée à Tours. C'est une entreprise organisée en plusieurs départements parmi lesquels on retrouve le service informatique dans lequel j'ai effectué ce stage.

Le groupe APRIL est une entreprise française qui conçoit, gère et distribue des solutions d'assurance et des prestations d'assistance pour les particuliers, professionnels et les entreprises. Sa filiale APRIL Moto,

conçoit, gère et distribue des produits d'assurance destinés à des personnes possédant un véhicule deux-roues.

## Objectifs du stage

Les objectifs de ce stage sont divers. Il a tout d'abord été question d'appréhender l'environnement de l'entreprise. En d'autres termes se familiariser avec le secteur de l'assurance deux-roues. Par ailleurs, l'amélioration de l'outil de veille concurrentielle ainsi que l'élaboration d'un modèle prédictif afin d'anticiper les éventuels départs ont été au coeur du stage.

Avant de parler plus en détails de ces différentes missions, nous proposerons quelques définitions utiles.

## Courtier grossiste en assurance

Le courtier grossiste est un courtier d'assurances qui s'est spécialisé sur une ou plusieurs niches de marché pour proposer des offres d'assurances performantes à des clients qui ne peuvent pas trouver de réponses à leurs besoins en direct auprès des compagnies d'assurances.

C'est le cas pour les expatriés qui cherchent une assurance santé ou prévoyance. Les compagnies conventionnelles d'assurance françaises ne proposent pas de garanties de ce type en direct car elles ne disposent ni de services de gestion adaptés, ni des réseaux de distribution capables de leur apporter un nombre suffisant d'assurés pour rentabiliser leurs investissements compte tenu de leurs frais de structures élevés.

Ce sont donc les des courtiers d'assurances spécialisés qui négocient auprès de ces compagnies des conditions dites « de gros » pour assurer les expatriés. Ils définissent des gammes de formules de garanties et négocient les tarifs. Les offres sont ensuite distribuées dans les mêmes conditions de garanties et de tarifs par leurs équipes commerciales et par des courtiers (distributeurs) indépendants. C'est ce schéma que nous retrouvons pour la plupart des offres d'assurance expatrié, avec APRIL, Assur Travel, ASFE ou encore Indigo Expat.

Ce business model a des conséquences pour l'assuré. Ce montage permet à l'assuré de bénéficier de plusieurs avantages notables :

- **Une gestion efficace, à taille humaine** : La gestion des contrats et des remboursements est faite généralement par le courtier grossiste, par des équipes gestionnaires spécialisées et à taille humaine, de 10 à 100 personnes. L'assuré a ainsi la possibilité de joindre directement les personnes qui traitent les dossiers.
- **La sécurité financière des plus grands assureurs** : Les contrats sont assurés par des compagnies telles que Allianz, GAN, Swisslife. Ce sont elles qui sont garanties financièrement des engagements pris auprès des assurés. Elles n'interviennent pas dans la gestion des dossiers des individus. L'assureur n'a pas de contact direct avec l'assuré. Si un courtier grossiste venait à disparaître (ce qui n'est jamais arrivé sur ce marché jusqu'à présent), les assurés continueraient à être couverts par la compagnie d'assurance qui aurait alors pour tâche, soit de reprendre la gestion, soit de la confier à un autre courtier gestionnaire.
- **Plusieurs réglementations de protection des assurés se cumulent** : Les compagnies d'assurances françaises sont soumises à des réglementations strictes en matière de sécurité financière, de solvabilité et d'engagements à garantir les assurés quelle que soit l'évolution de leur état de santé ; d'un autre côté les courtiers grossistes d'assurances sont soumis à des règles strictes en matière de conseils et d'informations apportés aux clients ou de gestion de réclamations. Dans le cas des courtiers grossistes, ces règles se cumulent en apportant à l'assuré une protection maximum.

# 2

## Le groupe APRIL

Le groupe APRIL dont le siège social se trouve à Lyon est un courtier grossiste en assurance. Il a été fondé en 1988 par Bruno BRUSSET et Xavier Coquard. C'est un groupe qui a pour mission principale de concevoir, gérer et distribuer des produits d'assurance pour les particuliers, professionnels et entreprises. Le groupe compte environ 3 500 salariés, dont l'activité est répartie dans 22 pays et a réalisé un chiffre d'affaires de 1017,3 M€ en 2019.

Concernant les particuliers, le groupe APRIL propose des produits d'assurance pour : **La santé , l'emprunteur, prévoyance, Expat/Voyage, Moto, Bateau**

Concernant les professionnels et les entreprises : **Santé pro, Santé et prévoyance pro, Santé et prévoyance collective, Assurance BTP, Santé à l'étranger**

Le groupe APRIL s'organise comme suit :

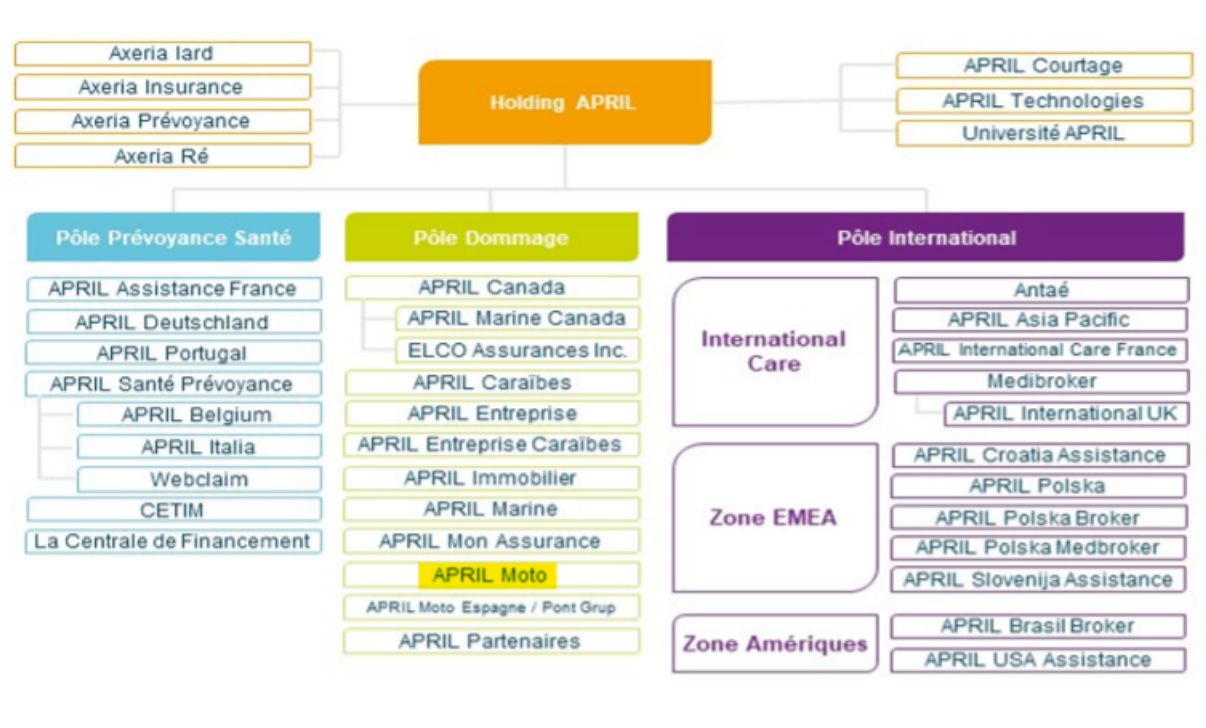


Figure 2.1 : Organigramme du groupe APRIL

### 2.1 APRIL MOTO

April Moto est la filiale du groupe APRIL qui a pour objectif de concevoir, gérer et distribuer des contrats d'assurance spécialement développés pour la pratique du deux-roues. Les clients sont des parti-



culiers et des courtiers en assurance. Elle compte environ une centaine de collaborateurs qui travaillent pour répondre aux objectifs de l'entreprise.

APRIL Moto s'organise comme suit :

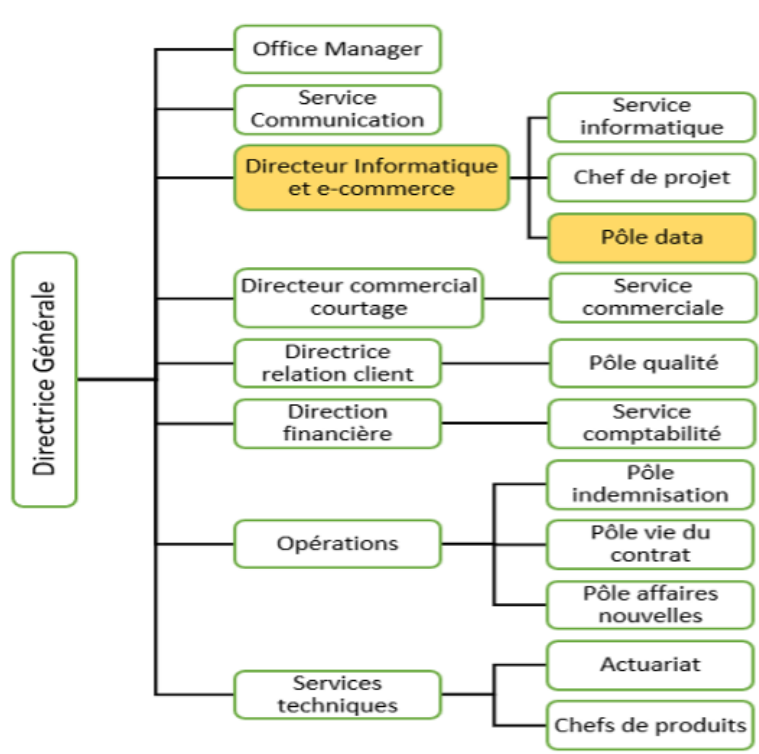


Figure 2.2 : Organigramme de APRIL Moto

## 2.2 Le marché de l'assurance deux-roues

Environ 3,6 millions de deux-roues composent le parc des véhicules français immatriculés en 2019. Autant de véhicule à assurer dans un contexte concurrentiel, à la fois attractif et en développement pour les acteurs qui se partagent le marché.

Historiquement réservé à une population de passionnés, le deux-roues est aujourd'hui la cible d'une population plurielle, qui assimile davantage le deux-roues à un mode de transport quotidien qu'à un loisir de plein air. Les deux-roues motorisés et assimilés composent un parc hétérogène de véhicule composé notamment de motos de route (38%), scooter (33%) et tricycles (13%).

A l'instar de l'automobile, les véhicules légers répondent à plusieurs obligations légales telles que le respect du code de la route et la souscription à une assurance. Ainsi, au même titre que l'assurance automobile, la seule assurance obligatoire est la responsabilité civile suivant l'article L211-1 du code des assurances.

Néanmoins, l'assurance moto est singulière et se démarque de l'assurance automobile de plusieurs façons.

### 2.2.1 Des facteurs différenciant

#### L'assurance moto : vers du sur-mesure

Un ensemble de paramètres définit une offre d'assurance deux-roues : le type et le modèle du véhicule, l'équipement associé, la couverture, l'usage, etc. Les assureurs l'ont bien compris et s'en servent pour appliquer leur stratégie de différenciation et proposer un socle classique de garanties avec des options à la

carte. De même que l'assurance automobile, la couverture varie de l'assurance au tiers jusqu'à l'offre tout risque avec options. L'assurance moto, déjà spécialisée, pousse très loin la personnalisation des offres allant jusqu'à une hyper-segmentation du marché pour répondre aux besoins de niche et attirer une clientèle friande de sur-mesure. Bien que permettant d'établir des offres aux prix les plus justes, ces pratiques ont l'inconvénient de complexifier les contrats d'assurance.

### **Des risques plus élevés**

La seconde particularité est le reflet d'une réalité plus dommageable : la sinistralité élevée des deux-roues. Du côté de l'accidentologie, les statistiques de la sécurité routière le prouvent. Les accidents impliquant les deux-roues ont provoqué 25,1 % des tués sur la route en 2019 et 30 % des blessés (21 000). Ces chiffres sont le reflet de la conduite spécifique du deux-roues où le pilote est en première ligne de tout accident. Les assurances proposent des offres adaptées à ces risques incluant des garanties corporelles importantes.

Du côté de la dégradation et du vol, les chiffres ne sont guères mieux. En 2019, 59 835 deux-roues ont été volés, soit 10 % des véhicules motorisés subtilisés en France selon l'Observatoire National de la Délinquance et des Réponses Pénales. Ces statistiques éloquentes influent sur les prix et contraignent les assureurs à exiger de l'assuré des équipements antivols mécaniques ou électroniques agréés.

### **Un marché unique**

Initialement peu disputé par les assureurs conventionnels pour les spécificités évoquées ci-dessus (marché de niche et sinistralité élevée), le marché de l'assurance deux-roues était monopolisé par les mutuelles d'assurance grand public telles que l'Assurance Mutuelle des Motards (ADMMDM), la MMA et la Macif. Ces acteurs répondaient alors au besoin des sociétaires tout en offrant, dans un second temps une couverture spécifique.

Désormais, l'on croise des acteurs divers, donnant un tournant concurrentiel au secteur. En plus des mutuelles, nous retrouvons les assureurs conventionnels, bancassureurs (tels que Axa et le Crédit Agricole), courtiers et grossistes courtiers. A l'inverse des deux premiers, qui se contentent d'une présence à minima, les courtiers ont percé récemment en introduisant des offres quasi-sur-mesure, en innovant sur les stratégies marketing et en utilisant des canaux de distribution novateurs. Pour cela, ils s'appuient sur des porteurs de risque, tel que Allianz pour APRIL Moto, l'un des leaders français du marché.

Les grossistes courtiers-grossistes ciblent tout particulièrement un marché de niche où il est possible de créer des produits innovants adaptés à des besoins spécifiques. Ils suivent une orientation « digital » en proposant des services en ligne dont notamment les simulateurs de tarifications en fonction des besoins, la souscription et l'édition de documents contractuels en ligne. L'assurance Moto Verte (AMV, filiale de Fihlet Allard) a également mis à disposition de ses clients le réseau social MotoSpot.

Voici une liste de principaux assureurs 2 roues :

AMV, APRIL, Assu 2000, Assur Bon Plan, Assuréo, AssurOnline, AXA, Caisse d'épargne, CIC, Crédit Agricole, Crédit Mutuel, EuroAssurance, Finaxy Moto, GMF, La Banque Postale, La Mutuelle des Motards, LCL, MAAF, MACIF, MAIF, Matmut, MMA, Quattro Assurances, RVA Assurances, Société Général.

# 3

## Outil de veille concurrentielle

---

### 3.1 Présentation et objectifs

Un robot de web-scraping permettant de récupérer quotidiennement les tarifs d'assurance des principaux concurrents de APRIL Moto sur les sites internet des comparateurs tels que « LesFurêts » ou « Sollyazar » a été mis en place. La mission a été d'assurer son amélioration et son bon fonctionnement.

Concrètement, le web-scraping (« racler le web » en français) est le processus d'extraction de données sur un site internet. Il permet de récolter une multitude d'informations précieuses et de les rassembler dans une base de données.

C'est une technique devenue de plus en plus populaire chez les entreprises ces dernières années. En effet, le scraping présente plusieurs avantages tels que :

- Automatiser le processus de collecte de données à grande échelle
- Débloquer des sources de données apportant une valeur ajoutée à l'entreprise
- Prendre des décisions éclairées, basées sur les données.

L'objectif principal de cet outil pour APRIL Moto est donc de pouvoir assurer une veille concurrentielle permettant à l'entreprise d'avoir une vision précise de son positionnement tarifaire sur le marché de l'assurance deux-roues.

Pour mener à bien cette mission, l'objectif a été d'améliorer le script de web-scraping déjà existant. La finalité de ce code est de remplir le formulaire des sites de comparateurs de tarifs d'assurance avec les informations des « scénarios » définis au préalable par l'actuaire et le directeur technique.

Un scénario est profil d'individu avec les caractéristiques principales suivantes : Le type de véhicule (Moto, Scooter, Cyclomoteur, NVEI<sup>1</sup>), la marque, le modèle, l'année de la version du modèle, le volume de la cylindrée, le code postale et la ville de résidence, l'âge du véhicule, l'âge du souscripteur, l'âge du permis A si renseigné, l'âge du permis A1 si renseigné, l'âge du permis A2 si renseigné, l'âge du permis AM équivalent au BSR si renseigné, l'âge du permis B si renseigné, le nombre de mois suivant le dernier antécédent moto, le nombre de mois suivant le dernier antécédent auto, le CRM auto et le CRM moto, l'âge du véhicule au moment de son achat et la zone de risque lié à l'emplacement géographique.

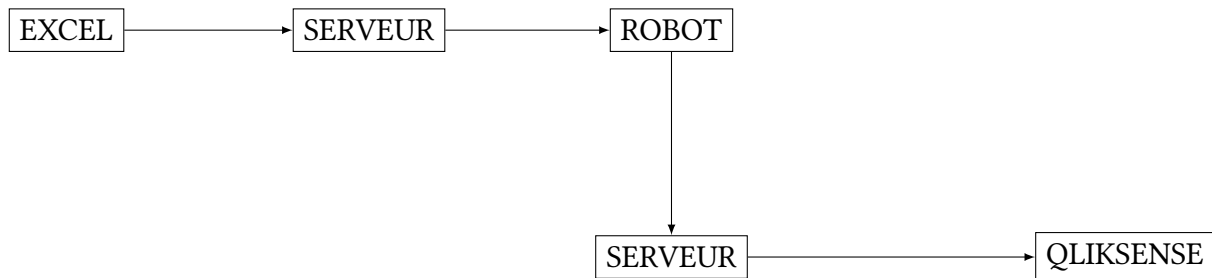
Un panel de 50 scénarios représentatifs du portefeuille client a été soigneusement défini dans un premier temps. Ce nombre de scénarios a été jugé raisonnable dans la mesure où il serait plus facile de déceler d'éventuelles erreurs qui feraient que le script ne parvienne pas à aller jusqu'au bout du remplissage des formulaires. C'est donc sur cette base de scénarios que le script de web-scraping a été amélioré avant de pouvoir le remettre en production.

---

1. Nouveaux Véhicules Electriques Individuels

## 3.2 Processus d'exécution

Le processus de fonctionnement de l'outil de veille concurrentielle comprend plusieurs étapes allant de l'importation des informations propres à chaque scénario à la visualisation des données récupérées. Sous forme de schéma, il peut s'illustrer comme suit :



De manière détaillée, le processus d'exécution est le suivant :

Tout d'abord les informations des 50 scénarios sont contenues dans un fichier Excel qui est déposé sur un serveur par le biais du logiciel *Talend*<sup>2</sup>. Ensuite ces informations sont déposées dans la base de données *SQL* que le robot de web-scraping récupère ensuite.

Plusieurs fichiers contenant des scripts *Python* sont nécessaires au bon fonctionnement du robot de web-scraping. Il y a trois scripts principaux qui sont des *classes*<sup>3</sup> qui interagissent entre elles. Ces fichiers sont les suivants :

- ***Robot.py*** : Ce fichier contient le script permettant le chargement et la gestion des caractéristiques des scénarios.
- ***Utils.py*** : Ce fichier contient le script permettant la transformation des informations dans le format adéquat.
- ***AMO\_lesfurêts.py*** : Ce fichier contient le script permettant le remplissage du formulaire. C'est le script qui a le plus nécessité d'amélioration car il dépend fortement de l'architecture du formulaire qui change au cours du temps.

Le fichier *Robot.py* permet donc la mise en forme des informations contenues dans le tableur Excel. Les caractéristiques des scénarios sont récupérées sur le serveur grâce à la fonction suivante :

```

def db_get_scenarios( self, start_id: Optional[int] = None, limit: Optional[int] = None):
    conn = self.conn
    where = ' ' if not start_id else ' and id >= ' + str(start_id)
    limit = ' ' if not limit else ' limit ' + str(limit)
    df = pd.read_sql_query("select * from scenarios where actif = 1"+where+limit+";", conn)
    return df
  
```

**Figure 3.1** : Fonction de connexion au serveur

Ensuite ces informations sont transformées et incorporées dans un dictionnaire<sup>4</sup> contenant les informations propres à chaque scénario grâce aux fonctions suivantes :

2. Outil ETL pour l'intégration de données.
3. Les classes sont un moyen de réunir des données et des fonctionnalités.
4. Un dictionnaire en python est une sorte de liste mais au lieu d'utiliser des index , des clés alphanumériques sont utilisées.

```

def scenario(self, scenario, scenarioId):
    return None

def initializeInput(self, scenario, scenarioId):
    today = datetime.today()

    input = scenario.to_dict()
    input['cylindree'] = str(input['cylindree'])
    input['age_veh'] = int(input['age_veh'])
    input['age_achat_veh'] = int(input['age_achat_veh'])
    input['cp'] = str(input['cp'])
    input['age_sous'] = int(input['age_sous'])

    if(not input['antecedents_moto']):
        input['nb_mois_assur_moto'] = '0'
    else:
        input['nb_mois_assur_moto'] = str(int(input['antecedents_moto']))

    input['crm_moto'] = str(input['crm_moto'])

    if(not input['antecedents_auto']):
        input['nb_mois_assur_auto'] = '0'
    else:
        input['nb_mois_assur_auto'] = str(int(input['antecedents_auto']))

    input['crm_auto'] = str(input['crm_auto'])
    input['dmec'] = Utils.getFixedDateFromAge(input['age_veh'])
    input['dachatv'] = max(Utils.getFixedDateFromAge(input['age_achat_veh']), input['dmec'])

    # input['ddn'] = Utils.getRandomDateFromAgeWithMax(input['age_sous'])
    input['ddn'] = Utils.getFixedDateFromAge(input['age_sous'])

    # input['pa2'] = False if (input['age_pa2'] == '' or not(input['age_pa2']) or math.isnan(input['age_pa2'])) else True
    input['pa2'] = False if (not 'age_pa2' in input) or (input['age_pa2'] == None or input['age_pa2'] == '' or math.isnan(input['age_pa2'])) else True
    #input['dpa2'] = max([Utils.getRandomDateFromAgeWithMax(int(input['age_pa2'])), input['ddn'] + timedelta(6576)]) if input['age_pa2'] != '' else None
    input['dpa2'] = Utils.getFixedDateFromAge(int(input['age_pa2'])) if input['pa2'] == True else None

    # input['pa'] = False if (input['age_pa'] == '' or not(input['age_pa']) or math.isnan(input['age_pa'])) else True
    input['pa'] = False if (not 'age_pa' in input) or (input['age_pa'] == None or input['age_pa'] == '' or math.isnan(input['age_pa'])) else True

```

Figure 3.2 : Fonction pour la création du dictionnaire

Après cela, le script contenu dans le fichier *AMO\_lesfurets.py* permet de remplir les informations des scénarios dans le formulaire du site « lesfurêts.com ». Le script est composé de plusieurs fonctions qui permettent de remplir chaque étape du formulaire à savoir les informations sur les caractéristiques du véhicules, sur l'identité du client, etc.

```

class AMOlesfurets(Robot):
    > def bandeau_cookies(self) : ...
    >
    > def step1(self, input) : ...
    >
    > def step2(self, input) : ...
    >
    > def step3(self, input) : ...
    >
    > def step4(self, input) : ...
    >
    > def scenario(self, scenario, scenarioId): ...
    >
    > def recupInfos(self, scenarioId, niveauId, scenario, version): ...
    >
    > def debutScenario(self): ...
    >
    > def retourPremierePage(self): ...
    >
    > def isResultPageOK(self): ...
    >
    > if __name__ == '__main__': ...

```

Figure 3.3 : Fonctions pour le remplissage du formulaire

En annexe se trouve des extraits de codes des fonctions permettant le remplissage du formulaire. A

A la fin du remplissage du formulaire, des formules tarifaires sont présentées et varient selon les caractéristiques du scénario. Ces formules sont ensuite sauvegardées et exportées sur un serveur dans un tableur Excel tout de suite exploitable.

L'étape finale consiste à la récupération de ces informations sur le serveur par le biais du logiciel *Talend* qui les transfère dans une l'application Qlik Sense <sup>5</sup> dédiée.

### 3.3 Résultat

Rappelons que l'objectif de cet outil est d'assurer une veille concurrentielle. Pour que celle-ci soit effective, les données récupérées sont intégrées dans une application Qlik Sense. Cette application contient des feuilles ayant chacune une utilité particulière.

- **Baromètre A** : Cette première feuille de l'application Qlik Sense présente des statistiques générales. On peut y trouver un graphique retraçant l'évolution des tarifs au fil du temps, le nombre de scénario ayant eu des tarifs à la hausse ou la baisse, de même que le nombre d'assureurs ayant fait varier leurs tarifs.
- **Baromètre (Avec possibilité de filtrer sur la date) A** : Cette feuille a exactement la même utilité que la première à la différence que l'on peut sélectionner une date précise afin d'avoir un comparatif avec les dernières données récupérées.
- **Analyse A** : Dans cette feuille, on retrouve des analyses plus détaillées comme par exemple l'évolution de la prime moyenne proposée par les assureurs. Par ailleurs, elle offre la possibilité de se focaliser sur un scénario en particulier.
- **Suivi technique A** : Sur cette feuille on peut trouver des informations relatives au bon fonctionnement du robot. En d'autres termes, elle donne des informations sur la performance du robot telles que le pourcentage de complétion des scénarios. On peut donc savoir si le robot a pu aller jusqu'au bout du formulaire pour un scénario quelconque.

Cette application Qlik Sense est donc un outil de Business Intelligence. En pratique, on parle de Business Intelligence lorsqu'une entreprise dispose d'une vue d'ensemble des données issues de sources internes ou externes et qu'elle utilise ces données pour favoriser le changement, gagner en efficacité et s'adapter aux évolutions du marché.

Cette application a donc une finalité d'aide à la décision pour les dirigeants de APRIL Moto. Grâce à cette dernière, les dirigeants ont une vision éclairée des fluctuations tarifaires qu'il peut y avoir sur le marché des assureurs deux-roues.

### 3.4 Limites

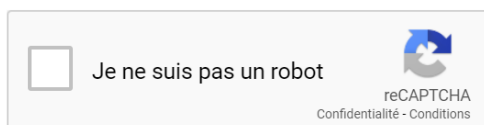
#### 3.4.1 Les captchas

Le fonctionnement d'un robot de web-scraping n'est pas sans embûches. Il est important de rappeler que le web-scraping est une technique devenue populaire auprès des entreprises durant ces dernières années. Par voie de conséquence les entreprises ont également mis en place d'autres techniques pour limiter ou empêcher de se faire sous-tirer des informations par le biais de ce genre de technique. Parmi celles-ci, la plus communément utilisée est celle de l'instauration de « Captcha ».

Les captchas acronyme de l'anglais « Completely Automated Public Turing test to tell Computers and Humans Apart », est une mesure de sécurité de type « authentification par question-réponse ». C'est une famille de tests de Turing permettant de différencier de manière automatisée un utilisateur humain d'un ordinateur. Ce test de défi-réponse est utilisé en informatique pour vérifier que l'utilisateur n'est pas un robot. Voici un exemple de Captcha que l'on peut rencontrer :

5. Logiciel de Business Intelligence.

Veuillez cocher la case ci-dessous pour continuer.



Je ne suis pas un robot

reCAPTCHA  
Confidentialité - Conditions

**Figure 3.4** : Exemple de captcha

C'est un élément qui s'est avéré problématique pour le bon fonctionnement du robot. En effet, étant donné que le robot de web-scraping effectue des tâches humaines à une vitesse d'exécution qui ne l'est pas forcément, les captchas apparaissent pour stopper immédiatement le robot. Afin de limiter l'impact des derniers dans la récupération des données, il existe des solutions. Parmi celles-ci, rendre anonyme l'adresse IP de l'ordinateur exécutant la tâche de récupération s'est avérée viable. Pour ce faire, nous avons utilisé le réseau d'anonymisation « **TOR** » qui permet de rendre « invisible » l'identité du serveur qui effectue la tâche. Solution efficace mais pas optimale car elle n'empêche totalement l'apparition des captchas mais diminue significativement leur impact en réduisant leur fréquence d'apparition.

Par ailleurs, il est possible d'instaurer des temps d'arrêt dans le code afin que le robot ne navigue pas à une vitesse surhumaine.

### 3.4.2 Le débit internet

Un autre point important pouvant entraver le bon fonctionnement du robot est le problème de débit internet. En effet, pour ce genre de pratique, il faut un débit internet stable et performant. Par conséquent, lorsque ce dernier rencontre des moments de moue, le robot ne parvient pas à faire correctement son travail et ne fournit pas les résultats escomptés.

### 3.4.3 Facteurs exogènes

La structure des formulaires que doit remplir le robot change au cours du temps. Etant donné que le robot tourne quotidiennement de manière automatique, on peut se rendre compte au jour le jour des éventuels soucis qui peuvent entraver son bon fonctionnement. Lorsque des modifications sur la structure du formulaire sont effectuées cela pose donc un véritable soucis étant donné que le script de remplissage est essentiellement basée sur la structure du formulaire. Cela demande donc des modifications du script afin de l'adapter aux divers changements effectués sur le formulaire.

# 4

## Prévention de l'attrition

---

### 4.1 Problématique

Le churn ou le taux d'attrition correspond pour une entreprise à la proportion de clients perdus au cours d'une période. Les entreprises doivent donc trouver des solutions pour lutter contre cette perte de clients qui représente par la même occasion une perte de revenus. Toute la difficulté réside dans la capacité à comprendre ce qui incite le client à renoncer aux services de l'entreprise.

A l'ère du Big Data et de l'analyse de données massives, plusieurs possibilités s'offrent aux entreprises. En effet, grâce à la collecte d'informations précieuses sur les clients au fil du temps, il est possible de mettre en place des techniques d'apprentissage automatique permettant de mettre la lumière sur les facteurs qui peuvent engendrer le départ de certains clients. Le secteur de l'assurance où le portefeuille de clients doit rester équilibré est un cas particulier de l'application de la prévention de l'attrition. C'est pourquoi dans l'optique de limiter et d'anticiper le départ de clients, nous avons mis en place une technique d'apprentissage automatique afin d'identifier les personnes ayant un risque de départ plus élevée que d'autres.

### 4.2 Les données

La première étape fut celle de la collecte de données utiles à la résolution de notre problème. Pour ce faire, nous avons procédé par un requêtage sur les bases de données internes à l'entreprise. Nous avons obtenu 3 bases de données que nous avons jointes sur la base du numéro de contrat. Ainsi, nous avons récupéré un échantillon de 87042 contrats actifs ou résiliés sur la période du 1-06-2020 au 1-09-2020 et 55 variables. S'agissant des données utilisées pour la prédiction, nous avons récupéré les informations d'un échantillon de 67826 contrats actifs au 01-06-2021.

#### 4.2.1 Mise en forme de la base de données

Avant de passer à la phase de modélisation, les données ont nécessité une exploration préliminaire qui nous a permis de mettre la lumière sur certains soucis que nous avons corrigés.

- **Les données manquantes** : Le jeu de données contenait des informations manquantes. Certaines variables n'étant pas renseignées pour certaines observations, nous avons donc décidé de supprimer ces observations qui représentaient une faible proportion de la base de données (2 % de l'effectif de départ). Par ailleurs, certaines variables contenaient des données manquantes pour des raisons connues telles que la non-saisie de l'information. Par conséquent, nous les avons remplacé par les valeurs adéquates. De même, certaines variables étant très peu renseignées, nous avons donc décidé de supprimer ces variables.
- **Les données erronées** : Certaines observations étant mal renseignées pour certaines variables, elles nécessitaient d'être traitées. Certaines observations avaient une valeur de prime négative par



exemple, chose incohérente. Par conséquent, nous avons conservé uniquement les individus ayant une prime positive ou nulle. Certaines valeurs des variables au format date étaient incohérentes. Nous avons supprimé ces observations.

- **Recodage et transformation des variables qualitatives** : Certaines variables contenant énormément de modalités ont été recoder de sorte à simplifier la compréhension du jeu de données.

Il se trouve que APRIL Moto propose des services pour les personnes ayant des voitures. Cependant, ce type d'individu représente une très faible part du portefeuille client. Par conséquent, nous avons supprimé ces observations afin de se focaliser uniquement sur les deux-roues. De même, nous avons supprimé les observations ayant un véhicule faisant partie de la catégorie Quad Moto Verte car elles représentent un part très faible du portefeuille.

Après avoir effectué toutes ces tâches préliminaires, nous obtenons une base de données exploitable contenant 78019 observations et 38 variables. Nous pouvons classer les variables caractérisant notre échantillon en 3 catégories :

- **Les variables individuelles** : La civilité, le numéro de département, la date de mise en circulation du véhicule, le(s) type(s) de permis que possède l'assuré (A, A1, A2, B, BSR), le bonus-malus auto initial, le bonus-malus moto initial, les antécédents auto ou moto si il y en a.
- **Les variables relatives aux caractéristiques du contrat de l'assuré** : Une variable indiquant l'état du contrat (actif ou résilié) qui sera notre variable d'intérêt. Le montant de la prime hors taxe associée, le nombre d'avenants au contrats, les options associées au contrat (PPC1, PCC2, PCC3, CASSE MECA, VALEUR MAJOREE, ASSISTANCE PLUS, ASSISTANCE VR, ACCESS EQUIP, RACHAT FR VI, RACHAT FR DTA), le canal de souscription, le nombre d'appels entrant, le nombre d'appels sortant, le nombre d'emails entrant, le nombre d'emails sortant, le nombre de courriers entrant et sortant,.
- **Les variables relatives aux caractéristiques du véhicule de l'assuré** : la catégorie du véhicule (Moto, Scooter/Cyclo 50, Scooter > 50), la classe, le groupe et le volume de la cylindrée.
- **Les variables relatives au canal d'acquisition** : Le type d'apporteur (comparateur, concessionnaire, direct, courtier), le nombre d'affaires nouvelles réalisées sur les 12 derniers mois ainsi que le nombre de résiliation sur les 12 derniers mois.

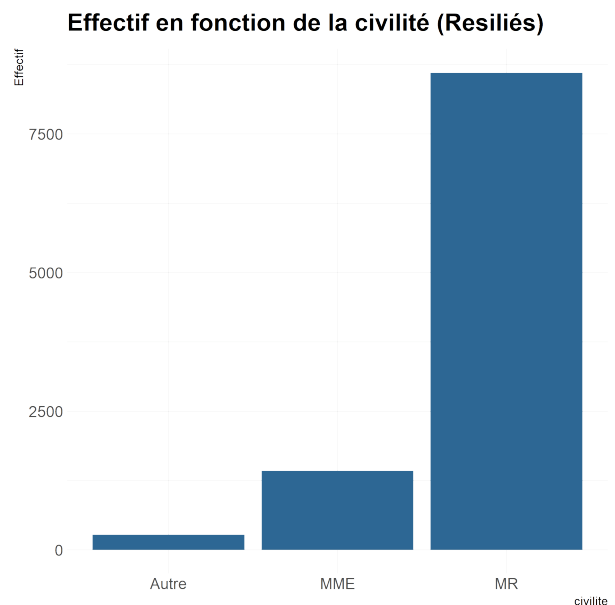
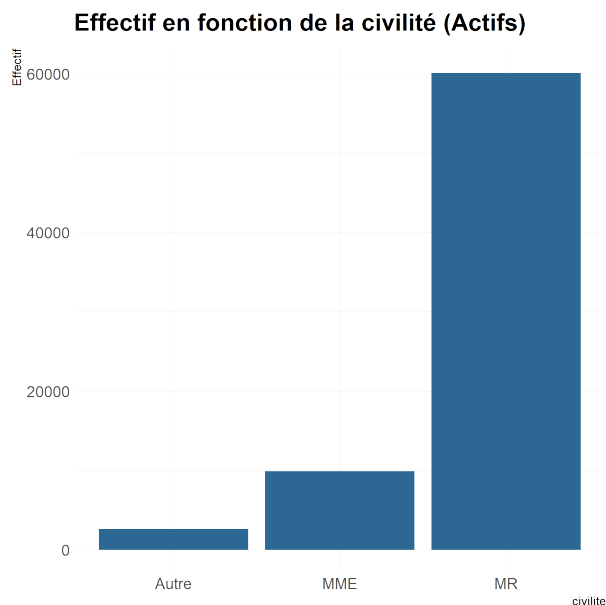
## 4.3 Analyse exploratoire

Notre objectif est d'établir un modèle de classification binaire capable de prédire le mieux possible si un assuré a plus de chance de résilié son contrat qu'un autre au vue de ses caractéristiques. Notre variable à expliquer sera celle en référence à l'état du contrat, indiquant si le contrat de l'individu est résilié ou non.

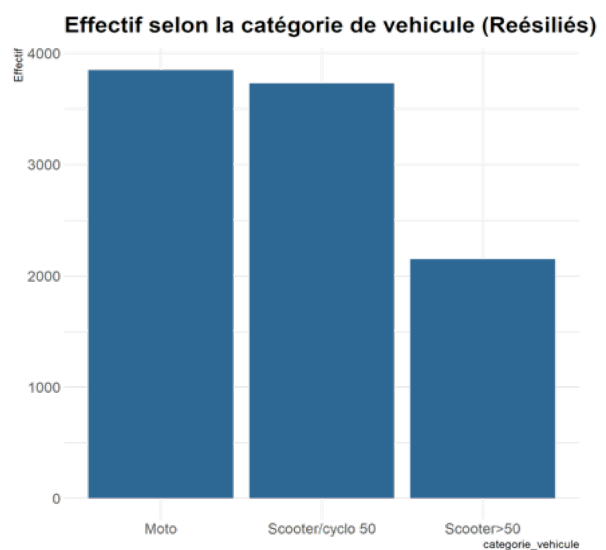
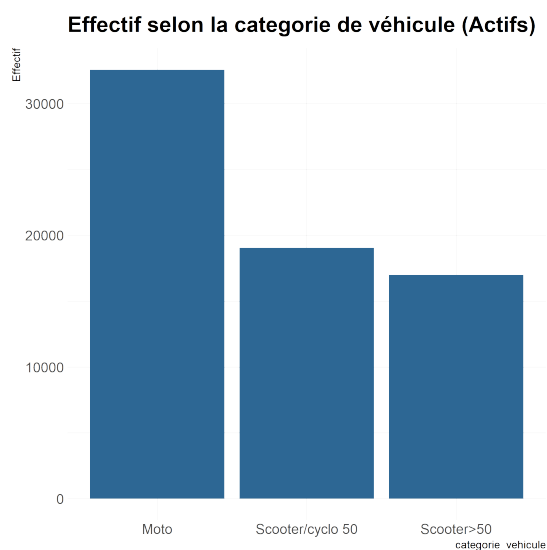
|          | Effectif | %    |
|----------|----------|------|
| Actifs   | 68338    | 87.6 |
| Résiliés | 9681     | 12.4 |

On se rend tout de suite compte que nous faisons face à un déséquilibre de classes important que nous devons prendre en compte dans la suite notre analyse.

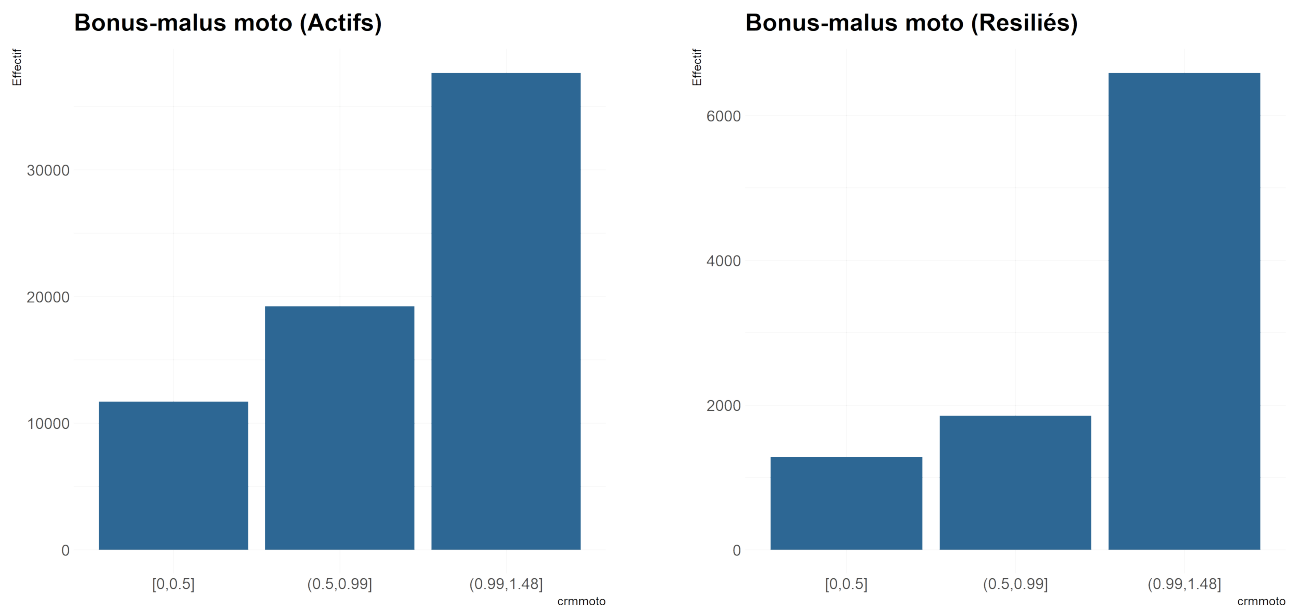
Par conséquent, si on s'intéresse aux caractéristiques de ces deux classes d'individus séparément, on a que les individus dont le contrat est toujours en cours ont une prime moyenne hors taxe de 393 € et ceux dont le contrat est résilié, une prime moyenne hors taxe de 404 €. Il n'y a donc pas de grande différence entre les niveaux de primes attribuées.



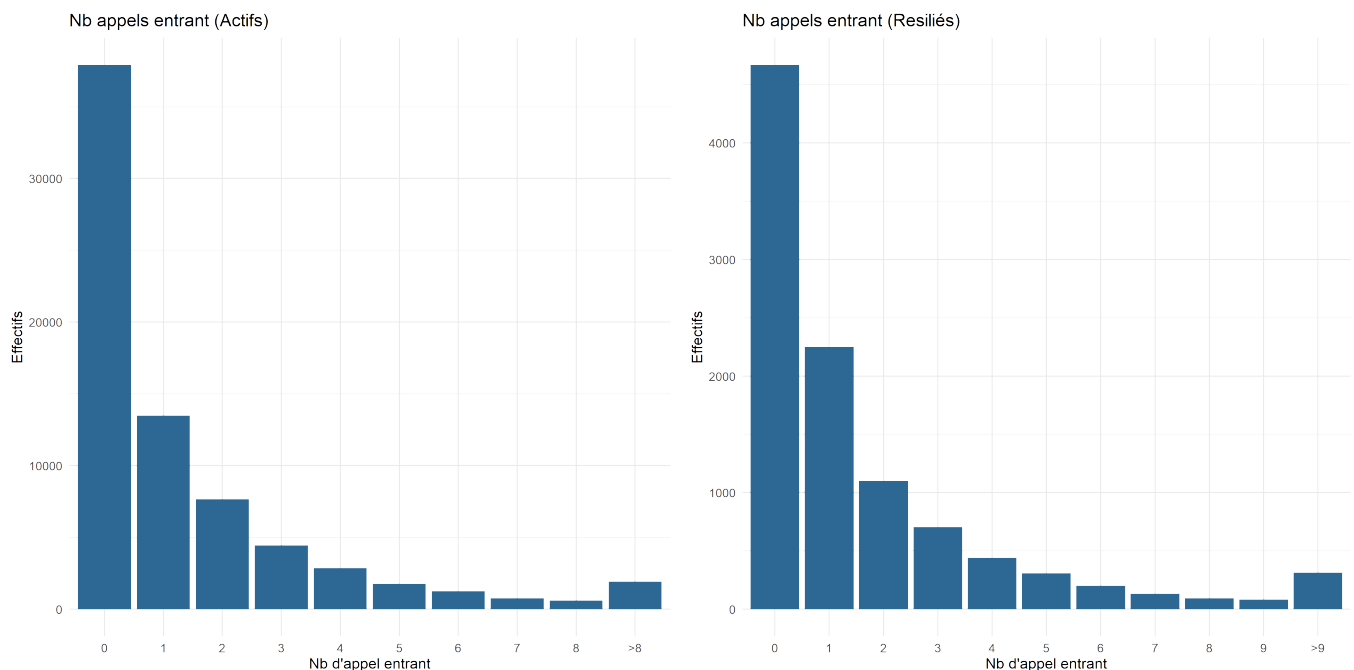
Que ce soit chez les individus ayant un contrat en cours ou chez les individus ayant un contrat résilié, il y a majoritairement des hommes.



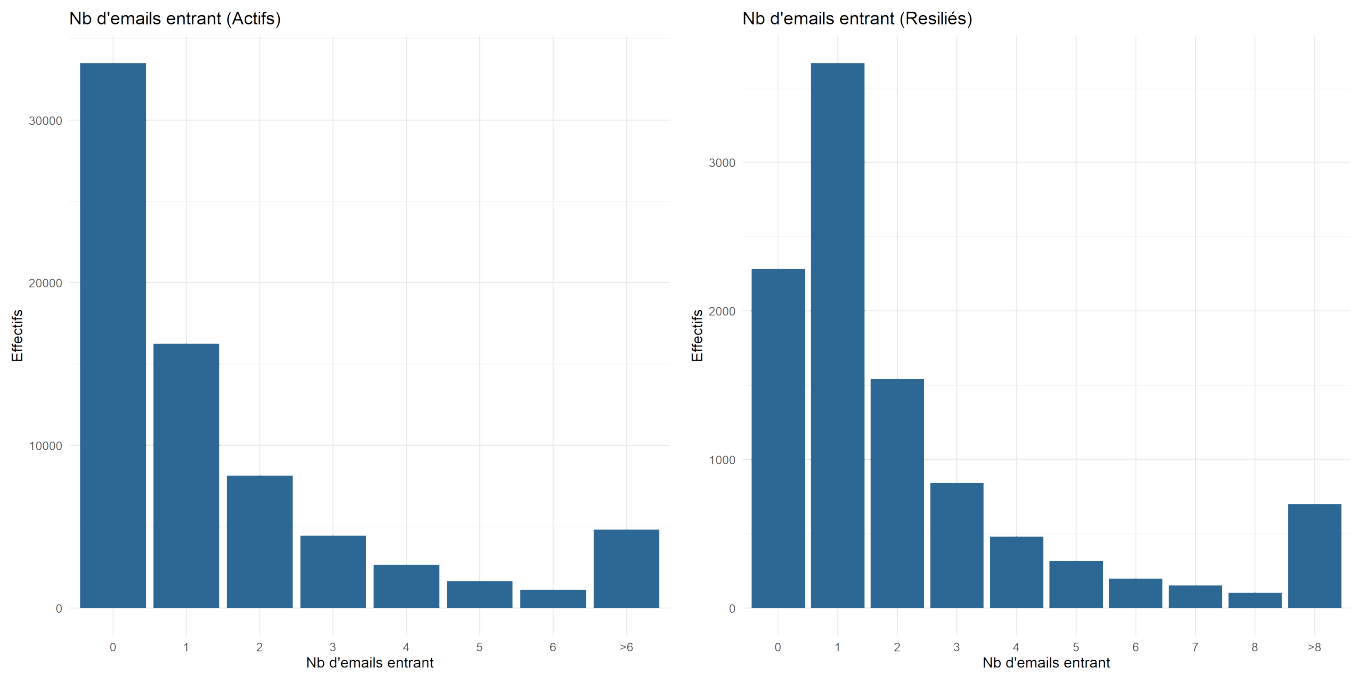
Chez les individus ayant un contrat en cours, la majorité possède une moto. Cependant, on voit que chez les personnes ayant un contrat résilié, la majorité possède également une moto mais aussi qu'une très grande part de ces individus possède un scooter/cyclo 50.



Le système du bonus-malus (ou coefficient de réduction) est un principe de réduction-majoration de la prime d'assurance moto lors de la souscription à un contrat. Elle varie selon le comportement du conducteur en fonction des sinistres et de la responsabilité de l'assuré. Si un motard a un bonus, le montant de l'assurance sera moins élevé comme une récompense à la bonne conduite. En revanche, s'il a un malus, ce montant sera plus important si l'assuré a été partiellement ou totalement responsable d'un sinistre. Lorsque la valeur du coefficient de majoration est supérieure à 1, nous sommes dans le cas d'un malus et inversement. Ainsi, on voit sur ce graphique que ce soit chez les individus ayant un contrat en cours ou ceux ayant un contrat résilié que la valeur de leur bonus-malus est comprise en 0.99 et 1.48.



Les appels entrant sont les appels que reçoivent l'entreprise de la part des clients pour des raisons diverses. Concernant les actifs et les individus ayant un contrat résilié, on voit que majoritairement il n'y a aucun appel entrant. Cependant, on voit que chez les personnes dont le contrat est résilié, malgré le fait qu'il n'y ait majoritairement pas eu d'appels entrant, bon nombre d'individus ont eu à passer des appels envers la boîte.



Comme les appels entrant, les emails entrant sont ceux que les clients ont envoyé à l'entreprise. On voit que pour les personnes dont le contrat est résilié, il y a eu plus d'envoi d'emails de la part des clients vers la boîte.

Après cette phase d'analyse exploratoire, plusieurs modélisations ont été essayées. Seule la plus pertinente est présentée dans ce mémoire.

Autres graphiques : voir annexe A

## 4.4 Modélisation

Nous avons scindé notre échantillon de départ en deux : 75 % dédiée à l'entraînement et 25 % conservé comme échantillon test. Nous avons retenu les variables suivantes pour la modélisation :

Le montant de la prime hors taxe, la date de mise en circulation du véhicule, la catégorie de véhicule, le bonus-malus auto et moto, nombre d'appels sortant, nombre d'appels entrant, nombre d'emails entrants, nombre d'emails sortant, nombre de courriers entrant, le nombre d'avenants au contrat (modifications effectuées), les différentes options, le canal de souscription, les antécédents auto et moto et la zone géographique.

Après avoir appliqué différentes modélisations, la plus convaincante a été celle de la méthode dite du « **Gradient Boosting** » se basant sur des arbres de décisions. Le Boosting de Gradient est un algorithme d'apprentissage supervisé dont le principe est de combiner les résultats d'un ensemble de modèles plus simples et ayant des performances faibles (arbres de décisions) afin de fournir une meilleure prédiction. On parle de méthode d'agrégation de modèles. L'idée est donc simple : au lieu d'utiliser un seul modèle, l'algorithme va en utiliser plusieurs qui seront ensuite combinés pour obtenir un seul résultat robuste.

Etant donné que nous sommes dans un cas de classes déséquilibrées, nous devons utiliser des métriques adaptées pour évaluer les performances de notre modèle. Aisni nous utiliserons la **précision** et le **recall**. La précision pour minimiser le taux d'erreurs parmi les exemples prédits positifs par le modèle et le recall pour détecter un maximum de positifs. Compte tenu du problème que nous essayons de résoudre, le **recall** sera plus approprié. L'objectif ici est d'identifier au mieux le nombre de clients qui sont effectivement enclins à résilier leur contrat même si certains « non-churners » sont identifiés à tort comme « churners ». C'est-à-dire, dans notre cas, il est préférable d'avoir un nombre de faux négatifs le plus faible possible.

De même, nous avons également utilisé la validation croisée pour obtenir de meilleurs résultats. Au lieu de simplement diviser les données en ensembles d'entraînement et de test, la validation croisée permet de découper nos données d'entraînement en un nombre  $K$  de sous échantillons.

Formellement, l'algorithme procède comme suit :

**Inputs** : Données d'entraînement  $(x_i, y_i)_{i=1}^n$ , une fonction de perte dérivable  $L(y, F(x))$ , nombre d'itération  $M$ .

L'algorithme :

1. Initialise le modèle avec une valeur constante :

$$F_0(x) = \operatorname{argmin}_{\gamma} \sum_{i=1}^N L(y_i, \gamma)$$

2. Pour  $m = 1$  à  $M$  :

- (a) Calcul les *pseudo-résidus* :  $r_{im} = \left[ \frac{\partial L(y_i, F(x_i))}{\partial F(x_i)} \right]_{F(x)=F_{m-1}(x)}$

- (b) Ajuste un apprenant de base (ou apprenant faible) fermé sous l'échelle  $h_m(x)$  aux pseudo-résidus, c'est-à-dire l'entraîner à l'aide de l'ensemble d'apprentissage  $(x_i, r_{im})_{i=1}^n$

3. Calcule le multiplicateur  $\gamma_m$  en résolvant le problème d'optimisation unidimensionnel suivant :

$$\gamma_m = \operatorname{argmin}_{\gamma} \sum_{i=1}^N L(y_i, F_{m-1}(x_i) + \gamma h_m(x_i))$$

4. Mets à jour le modèle :

$$F_m(x) = F_{m-1}(x) + \gamma_m h_m(x)$$

5. **Output** :  $F_M(x)$

Ce processus se fait de manière itérative.

Avant de pouvoir appliquer ce modèle à nos données, nous avons procédé au « future engineering » de nos variables. En d'autres termes, nous avons appliqué des transformations à nos variables afin qu'elles soient dans le format adéquat pour l'algorithme.

```

{r}
xgb_rec <- recipe(target ~ prime_ht + datecirc + categorie_vehicule +
  crmauto + crmmoto + nbtelexportant + nbtelexportant + nbemaie + nbemais + nbcoure + avenants + canal +
  PPC1 + PCC2 + PCC3 + ACCESS_EQUIP + ASSISTANCE_PLUS + ASSISTANCE_VR + RACHAT_FR_DTA +
  RACHAT_FR_VI + CASSE_MECA + VALEUR_MAJOREE + antecedents1 + antecedents2 + typetiers + new_zone, data = train)

%>%
  step_date(datecirc, keep_original_cols = FALSE) %>%
  step_other(all_nominal_predictors()) %>%
  step_corr(all_numeric_predictors(), threshold = 0.7) %>%
  step_dummy(all_nominal_predictors(), one_hot = TRUE) %>%
  step_nzv(all_predictors()) %>%
  step_smote(target)
  |
xgb_prep <- prep(xgb_rec)

```

Figure 4.1 : Future engineering

Dans un premier temps, grâce la fonction « **recipe** » on spécifie la formule du modèle que l'on va entraîner. Dans notre cas, on veut expliquer la variable « **target** » qui est notre variable cible qui fait référence à l'état du contrat. Ensuite dans les différents « **step** » ou étapes on procède aux transformations que l'on applique aux variables explicatives :

- On transforme la variable date de mise en circulation dans le format adéquat. C'est-à-dire en variable catégorielle.
- On crée une catégorie « autre » pour les modalités des variables qualitatives les moins représentées.
- On élimine les prédicteurs numériques ayant un coefficient de corrélation au minimum égale à 0.7.
- On transforme les variables catégorielles en variables numériques.
- On supprime tous les prédicteurs ayant une variance proche de zéro.
- On rééchantillonne les classes de la variable à prédire avec la méthode « smote ». Elle permet de générer des nouvelles observations de la classe minoritaire en utilisant la méthode des K plus proches voisins. Après cela, on se retrouve avec des classes équivalentes.

Le modèle que nous allons utiliser possède plusieurs hyperparamètres à optimiser. Techniquement, dans le logiciel *R* on spécifie le modèle avec les valeurs des hyperparamètres à optimiser comme suit :

```

{r}
xgb_spec <-
  boost_tree(
    trees = tune(),
    min_n = tune(),
    mtry = tune(),
    learn_rate = tune(),
  ) %>%
  set_engine("xgboost") %>%
  set_mode("classification")

```

Figure 4.2 : Spécification du modèle

Les paramètres à optimiser sont :

- **Trees** : Un entier pour le nombre d'arbres contenus dans l'ensemble.
- **Min\_n** : Un entier pour le nombre minimum d'observations requises dans un noeud.
- **Mtry** : Un nombre pour le nombre (ou la proportion) de prédicteurs qui seront échantillonnés au hasard à chaque division lors de la création des modèles d'arbres.
- **Learn rate** : Un nombre pour la vitesse à laquelle l'algorithme de boost s'adapte d'une itération à l'autre.

Après avoir spécifié le modèle et préparé nos variables, on agrège le tout comme suit :

```

xgb_wf <- workflow() %>%
  add_recipe(xgb_rec) %>%
  add_model(xgb_spec)

```

Figure 4.3 : Workflow

La fonction « **workflow** » nous permet d'agréger la spécification de notre modèle contenue dans l'objet « **xgb\_spec** » ainsi que le future engineering appliqué à nos variables contenu dans l'objet « **xgb\_rec** ». Pour finir, on incorpore le tout dans l'objet « **xgb\_wf** ».

Enfin, on optimise les hyperparamètres de notre modèle sur des sous échantillons des données d'entraînement obtenus par validation croisée comme suit :

```
xgb_res <- tune_race_anova(
  xgb_wf,
  folds,
  grid = 10,
  metrics = model_metrics,
  control = control_race(verbose = TRUE, verbose_elim = TRUE)
)
```

**Figure 4.4** : Optimisation des hyperparamètres

La fonction « **tune\_race\_anova** » calcule un ensemble de mesures de performance (par exemple, précision ou RMSE) pour un ensemble prédéfini d'hyperparamètres du modèle sur un ou plusieurs des sous échantillons des données d'entraînement issues de la validation croisée. Une fois qu'un nombre initial de sous échantillon a été évalué, le processus élimine les combinaisons d'hyperparamètres qui sont peu susceptibles d'être les meilleurs à l'aide d'un modèle ANOVA.

Ainsi dans cette fonction est contenu l'objet « **xgb\_wf** » que nous avons présenté plus haut, l'objet « **folds** » qui contient les échantillons obtenus par validation croisée sur lesquelles seront évalués les différents hyperparamètres à tester. En plus, cette fonction contient l'objet « **grid** » qui est le nombre de valeurs pour chaque hyperparamètre à tester et l'objet « **metrics** » utilisé pour définir les métriques de performance à utiliser.

#### 4.4.1 Résultats

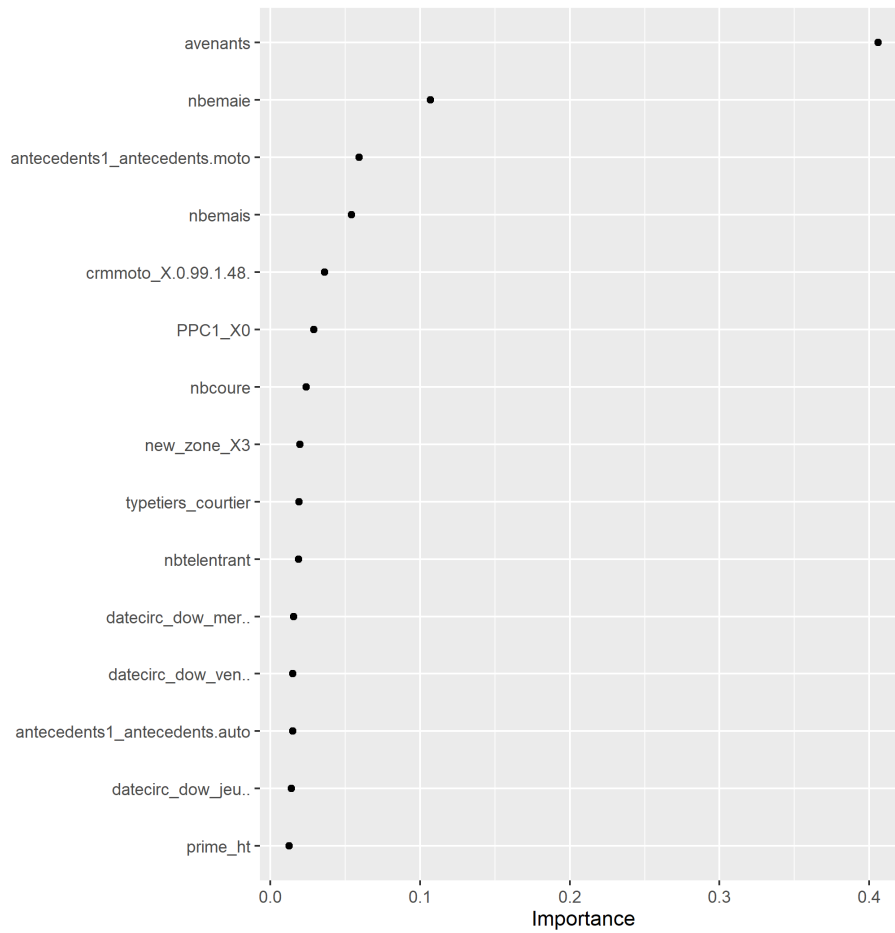
Après optimisation des hyperparamètres, on obtient les valeurs suivantes :

| mtry<br><int> | trees<br><int> | min_n<br><int> | learn_rate<br><dbl> |
|---------------|----------------|----------------|---------------------|
| 56            | 1726           | 13             | 0.0304494           |

**Figure 4.5** : Valeur des hyperparamètres après optimisation

Après cette phase d'optimisation des hyperparamètres, on applique à présent la spécification de notre modèle avec les valeurs optimisées des hyperparamètres sur nos données tests.

Les variables qui contribuent le plus à expliquer le phénomène sont les suivantes :



**Figure 4.6 : Variables importantes**

Le nombre d'avenants ou encore le nombre de modifications apportées au contrat est la variable qui contribue l'explication du modèle. Elle est suivie par le nombre d'emails entrant. Cette variable représente le nombre d'emails envoyé par le client à APRIL Moto. De même, la modalité « antecedents\_moto » de la variable « antecedents1 » indiquant que l'individu possède des antécédents moto contribue à expliquer notre phénomène. Ensuite nous avons la variable nombre d'emails sortant suivie de la modalité « 0.99 - 1.46 » de la variable indiquant le niveau de majoration de la prime hors taxe.

On voit que les variables relatives à la relation client telles que le nombre d'email entrant et sortant sont des éléments qui contribuent à la classification de nos individus. Ce sont sans doute des facteurs à prendre en compte de manière concrète pour tenter de résoudre le problème.

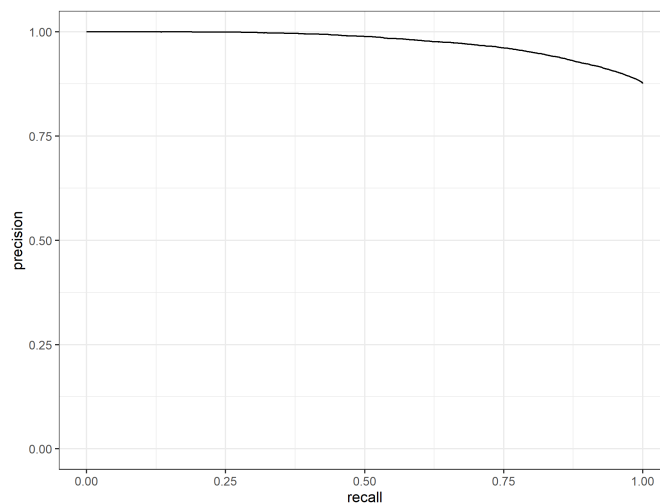
On réalise la matrice de confusion sur les données test :

|          | Actifs | Résiliés |
|----------|--------|----------|
| Actifs   | 16800  | 2086     |
| Résiliés | 285    | 335      |

Notre modèle a une précision de 0,88. En d'autres termes, lorsqu'il prédit qu'un individu a un contrat actif, il est correct 88 % du temps. S'agissant du recall, il est de 0,98 ce qui signifie qu'il identifie correctement 98 % des individus ayant un contrat actif.

La courbe de précision recall est la suivante :





**Figure 4.7** : Courbe Precision-Recall

#### 4.4.2 Prédictions

Comme évoqué plus tôt, pour réaliser les prédictions, nous avons récupéré les informations d'un échantillon de 67826 contrats actifs au 01-06-2021 avec les mêmes caractéristiques des données d'entraînement. Après avoir appliqué le modèle sur ces données, nous avons obtenu des scores de risques pour chaque assuré. Par conséquent nous avons effectué une classification des individus en fonction de leur niveau de risque :

| Niveau de probabilité     | Effectif | %    |
|---------------------------|----------|------|
| Probabilité < 25 %        | 64766    | 95.5 |
| 25 % ≤ Probabilité < 50 % | 2406     | 3.5  |
| 50 % ≤ Probabilité < 75 % | 588      | 0.9  |
| Probabilité > 75 %        | 66       | 0.1  |

### 4.5 Actions concrètes

Suite aux résultats obtenus, il a été décidé de mettre en place des actions concrètes. Dans un premier temps, il s'agira de confronter les résultats du modèle à la réalité à un horizon de 3 mois. C'est-à-dire qu'au 1-09-2021, les résultats des prédictions seront confrontées à la réalité. Ceci car nous avons entraîné le modèle sur des données d'entraînement situées dans une période de 3 mois une année plus tôt. C'est-à-dire du 01-06-2020 au 01-09-2020.

Par la suite, un échantillon regroupant les individus ayant une forte probabilité de résiliation sera établie afin d'entreprendre des actions concrètes sur ces derniers et voir quels seront leurs effets.

### 4.6 Perspectives et améliorations

Après avoir confronté les résultats des prédictions à ceux de la réalité, si les résultats s'avèrent concluants, le modèle sera mis en production et fonctionnera de manière automatique. L'objectif sera d'obtenir des prédictions à un horizon de 3 mois et à chaque fois effectué des actions marketing sur les clients à fort

risque de résiliation afin de limiter l'attrition. D'un point de vue technique, il sera toujours question de trouver comment améliorer le modèle en réfléchissant à l'inclusion ou l'exclusion de certaines variables par exemple. L'objectif étant de minimiser les faux négatifs.

# 5

## Conclusion

---

Exploiter au mieux les données est un objectif primordial pour APRIL Moto. A l'heure actuelle, les données représentent un enjeu stratégique pour les entreprises. Dans le but d'en tirer profit, il a donc été question de l'amélioration de l'outil de veille concurrentielle et la mise en place d'un outil dans le cadre de la prévention de l'attrition.

L'amélioration de l'outil de veille concurrentielle a donc permis l'automatisation de la récupération des tarifs proposés par les concurrents dans le but de connaître les fluctuations tarifaires de manière quotidienne. Grâce à cet outil, APRIL Moto a la possibilité de prendre des décisions rapides et adéquates dans le but de pouvoir s'ajuster sur le marché par rapport à ses concurrents.

S'agissant de l'outil de prévention de l'attrition, c'est un outil créé grâce à un processus d'apprentissage automatique. L'objectif est d'apprendre avec ce qui s'est déjà passé afin de mieux prédire l'avenir. En d'autres termes, comprendre les comportements des personnes assurées chez APRIL Moto, permettra à l'avenir de mieux anticiper et éventuellement limiter le départ de certains clients en mettant en place des actions marketing ciblées.

# A

## Annexe

---

### Outil de veille concurrentielle - Fonctions pour le remplissage du formulaire du site LesFurêts

```
def step1(self, input) :  
    #Première Partie : VOTRE DEMANDE  
  
    # Quelle est la cylindrée de votre véhicule ?  
    self.logger.info("Quelle est la cylindrée de votre véhicule")  
    try:  
        if(input['typeveh'] == 'Scooter/cyclo 50'):  
            id = 'CYLINDREE_INF-CYLINDREE'  
        else:  
            id = 'CYLINDREE_SUP-CYLINDREE'  
  
        if(id == 'CYLINDREE_INF-CYLINDREE'):  
            class_name = 'ButtonField-value-CYLINDREE_INF'  
            el = WebDriverWait(self.driver, self.delay).until(EC.element_to_be_clickable(class_name))  
            el.click()  
            self.logger.info('Clic sur la cylindrée OK')  
        else:  
            class_name = 'ButtonField-value-CYLINDREE_SUP'  
            el = WebDriverWait(self.driver, self.delay).until(EC.element_to_be_clickable(class_name))  
            el.click()  
            self.logger.info('Clic sur la cylindrée OK')  
    except Exception:
```

```

def step2(self, input) :
    #Deuxième partie : VOTRE VEHICULE
    # Quel est le modèle du véhicule à assurer ?
    try: ...

    except Exception:
        try:
            xpath = '//*[@class="VehiculeRecap-modifier"]/button'
            el = WebDriverWait(self.driver, self.delay).until(EC.element_to_be_clickable((By.XPATH, xpath)))
            el.click()
            try:
                xpath = '//*[@class="VehiculeMenu-item activate"]'
                ret = WebDriverWait(self.driver, self.delay).until(EC.element_to_be_clickable((By.XPATH, xpath)))
                ret.click()
                ret = WebDriverWait(self.driver, self.delay).until(EC.element_to_be_clickable((By.XPATH, xpath)))
                ret.click()
            except:
                pass
        except Exception:
            self.logger.error('Clic sur le bouton de choix du véhicule KO')
            raise Exception

    # Indiquez sa marque
    try:

```

```

def step3(self, input) :
    #Troisième partie : VOS DONNEES PERSONNELLES
    # Quelle est votre date de naissance ?
    try:
        self.logger.info("Quelle est votre date de naissance ?")
        id = 'Select-C_NAISSANCE-year'
        el = WebDriverWait(self.driver, self.delay).until(EC.element_to_be_clickable((By.ID, id)))
        el.click()
        xpath = '//*[@id="Select-C_NAISSANCE-year"]/div[*]/div/div[@data-value="'+datet
        el = WebDriverWait(self.driver, self.delay).until(EC.element_to_be_clickable((By.XPATH, xpath)))
        el.click()

        id = 'Select-C_NAISSANCE-month'
        el = WebDriverWait(self.driver, self.delay).until(EC.element_to_be_clickable((By.ID, id)))
        el.click()
        xpath = '//*[@id="Select-C_NAISSANCE-month"]/div[*]/div/div[@data-value="'+datet
        el = WebDriverWait(self.driver, self.delay).until(EC.element_to_be_clickable((By.XPATH, xpath)))
        el.click()

        id = 'Select-C_NAISSANCE-day'
        el = WebDriverWait(self.driver, self.delay).until(EC.element_to_be_clickable((By.ID, id)))
        el.click()

```

```

def step4(self, input) :
    #Quatrième partie : VOS INFORMATIONS
    # Vous êtes...
    # Choix du genre en random
    try:
        self.logger.info('Vous etes...')
        genre = random.choice(['homme', 'femme'])
        xpath = '//*[@id="C_SEXE-wrapper"]/div/div[2]/div/div/div[contains(concat(" ", @
        el = WebDriverWait(self.driver, 20).until(EC.element_to_be_clickable((By.XPATH, xpath)))
        el.click()
        self.logger.info('Clic genre '+genre+' OK')

    except Exception:
        self.logger.error('Clic genre KO')
        raise Exception

    # Pause
    time.sleep(2)

    # Quelle est votre profession ?

```

## Outil de veille concurrentielle - Application Qlik Sense

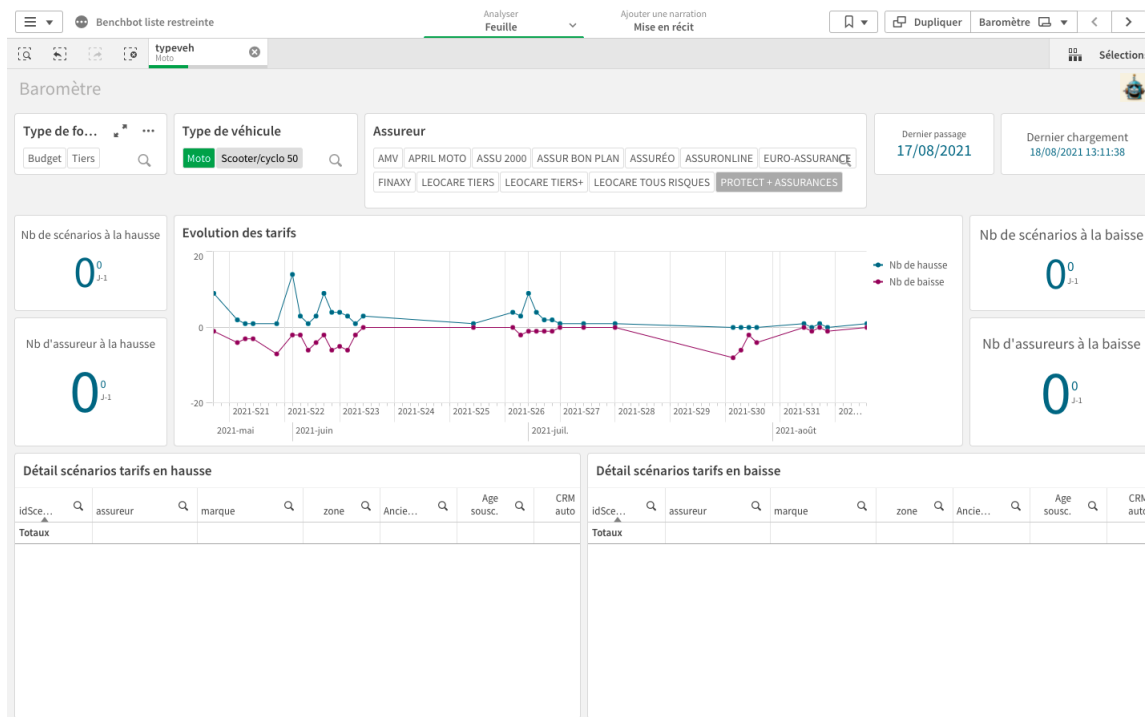


Figure A.1 : Feuille baromètre

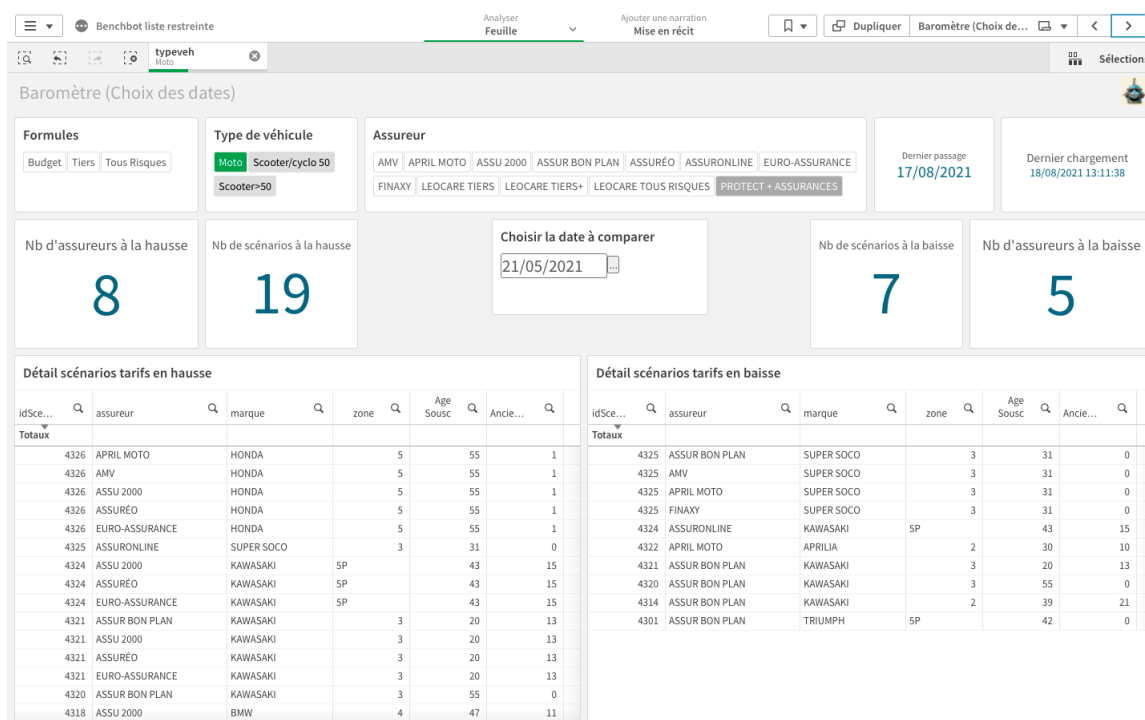


Figure A.2 : Feuille baromètre avec filtre sur la date

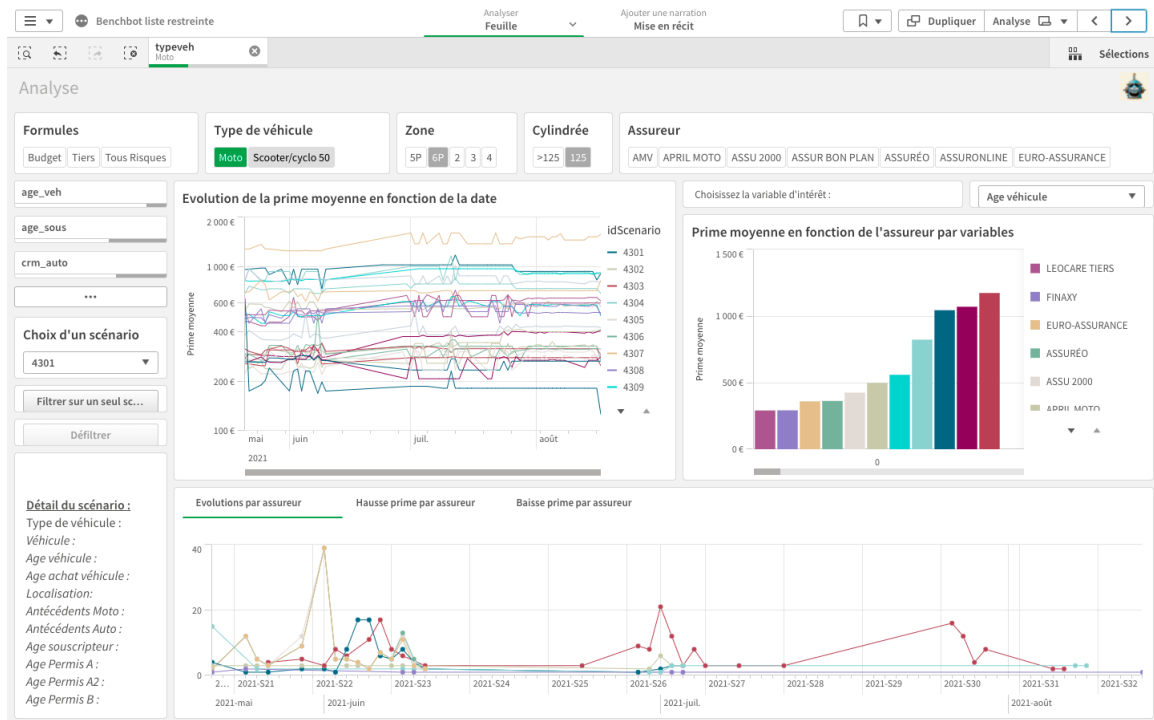


Figure A.3 : Feuille dédiée aux analyses

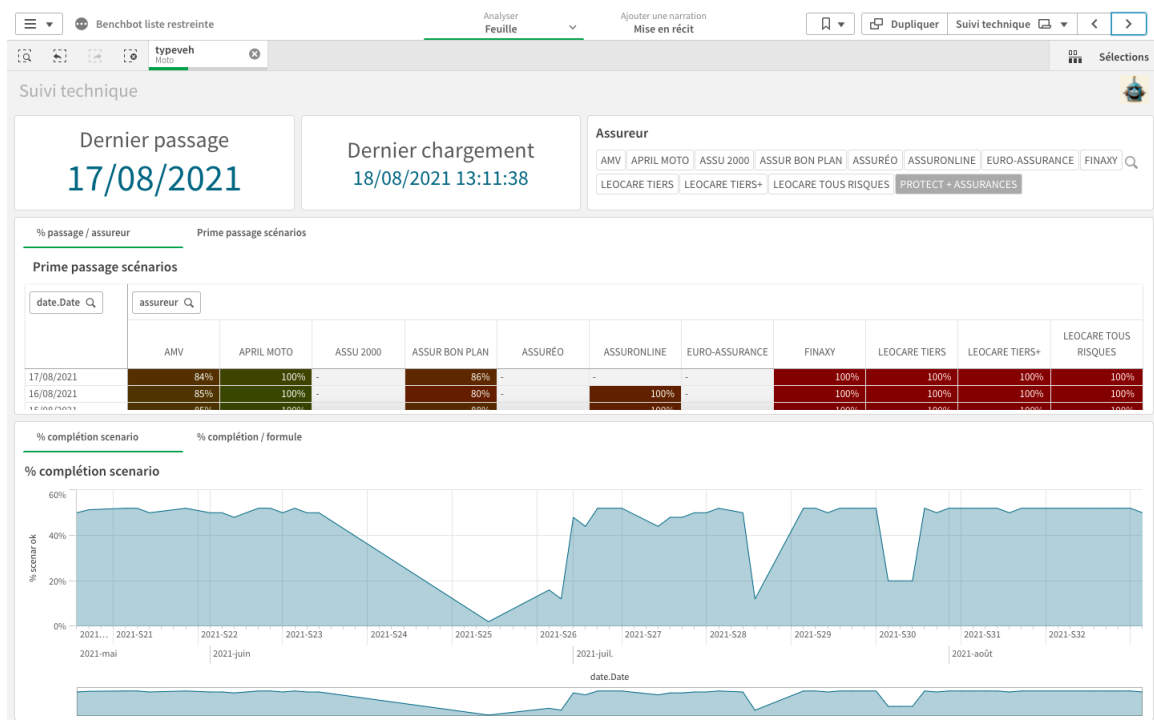
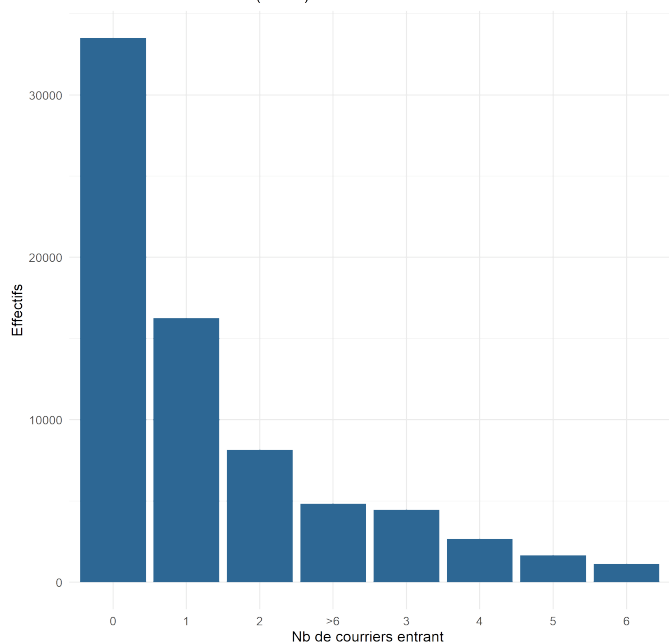


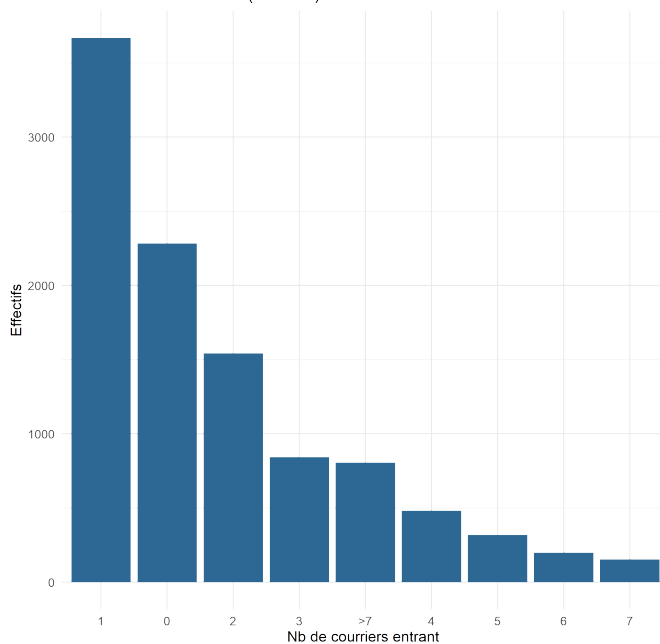
Figure A.4 : Feuille du suivi technique

## Prévention de l'attrition - Statistiques descriptives

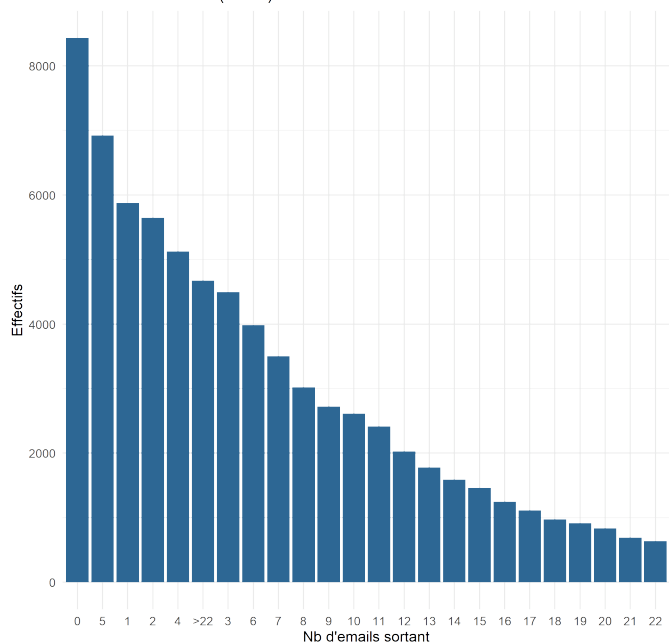
Nb de courriers entrant (Actifs)



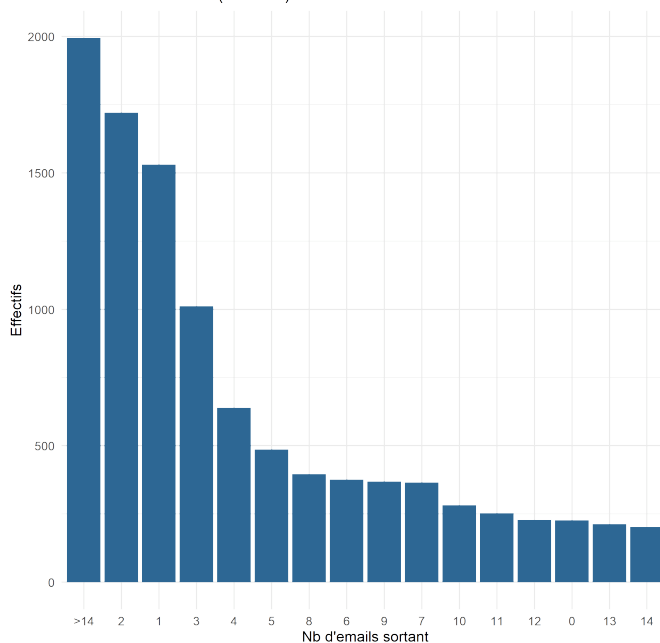
Nb de courriers entrant (Résiliés)



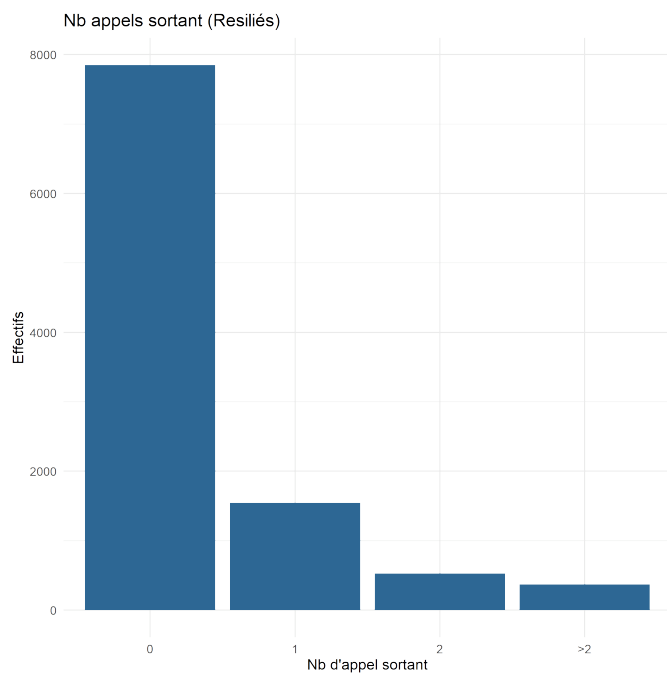
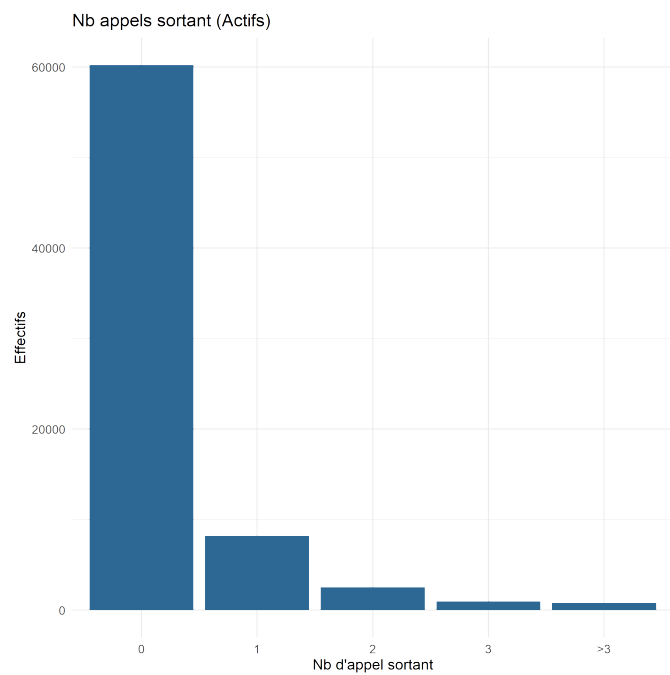
Nb d'emails sortant (Actifs)



Nb d'emails sortant (Resiliés)







# B

## Bibliographie / Sitographie

---

- <https://juliasilge.com/>
- <https://www.tidymodels.org/>
- <https://scikit-learn.org/stable/modules/ensemble.html>
- Ana Maria Perez Marin, Université de Barcelone, « Survival methods for analysis of customer life-time duration in insurance »