Mémoire présenté le :
pour l'obtention du diplôme
de Statisticien Mention Actuariat
et l'admission à l'Institut des Actuaires

Par : Omar El Farouk Guiz

Titre du mémoire : *Contribution de données externes dans la tarification et suivi du risque*

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

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

Membres présents du jury de l'Institut des Actuaires Signature

Membres présents du jury de la filière

Entreprise :

Nom :
BNP PARIBAS CARDIF

Signature :

Directeur de mémoire en entreprise :

Nom : Clément Saillard

Signature :

Invité :

Nom : Maria Aguilera

Signature :

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

Signature du responsable entreprise

Signature du candidat

Résumé

Aujourd'hui, nous comptons dans le monde plus de 2,87 milliards d'utilisateurs de téléphones mobiles. Peu à peu, leur importance dans la vie de chacun s'accentue. Au cours des années, les consommateurs ont vu le prix des téléphones augmenter occupant ainsi une part plus importante dans leurs budgets au point de faire de l'assurance contre le vol ou la casse une nécessité. En conséquence, le monde de l'assurance téléphonique est devenu un marché majeur en constante progression. Ce marché est distribué principalement par trois acteurs : les opérateurs téléphoniques, les chaînes de magasins et les banques ; où la digitalisation de la souscription à une offre devient une attente courante pour le client en comparaison à la souscription en point de vente.

L'utilisation par les différents constructeurs téléphoniques de composants communs intégrés à leurs produits nous permet une meilleure compréhension des facteurs responsables d'une fréquence élevée de casse. En enrichissant les variables tarifantes principalement basées aujourd'hui sur les tranches de prix des téléphones, il en résulte une prime de risque assuré plus adaptée. Pour une étude optimale de riches données sont donc nécessaires telles que les données du portefeuille mais également des données additionnelles dites externes. Dans ce contexte, l'objectif assigné à ce travail est de définir pertinemment les principaux facteurs qui auraient une incidence sur la fréquence de sinistralité. Ainsi, circonscrites et consignées, les facteurs en question seraient pris en considération dans la tarification de ce produit dans le futur.

Dans un premier temps, une analyse de la modélisation de l'apport de données physiques d'une base externe à nos données internes grâce aux plus récentes méthodes statistiques et de *Machine Learning* (utilisation de GLM, XGBoost, LGBM) est effectuée pour obtenir la meilleure prédiction de la fréquence de sinistralité.

Dans un second temps, une amélioration de la calibration des modèles en passant par différentes techniques telles que la *Cross Validation* ou méthodes de sélection est proposée.

Mots-clés : Assurance affinitaire, Téléphone mobile, Méthode d'apprentissage, XGBoost, Tarification, Arbres de décisions, Vol-Casse de téléphone, P&C, IARD, Modèles Linéaires, Shapley.

Abstract

Today, there are more than 2.87 billion smartphone users worldwide. Slowly and steadily, the smartphone is occupying an important place in the life of every user. Over the years, consumers have seen the prices of phones increase as a larger part of their budgets to the point where insurance against theft or breakage almost become a necessity. As a result, the world of mobile insurance is a large and growing market. Such product is mainly distributed by three players besides the direct insurers : telephone operators, retailers and banks; where the digitalization of the underwriting of an offer is increasing significantly compared to the underwriting at the point of sales.

The use by the various telephone manufacturers of common components integrated into their products allows a better understanding of the factors responsible for a high frequency of breakage. By enriching the pricing variables, which are today mainly based on telephone price brackets, the outcome is a more appropriate assessment of the premium for the insured risk. For an optimal study, rich dataset is therefore required besides portfolio data based on experience but also additional external data. In this context, the aim of this work is to define the main factors that would have an impact on the frequency of claims. Thus, identified and recorded, the factors in question could be taken into account in the pricing segmentation of this product in the future.

As a first stage, we realized an analysis of the impact of external data on the prediction of claims frequencies by using the latest statistical and Machine Learning methods (use of GLM, XGBoost, LGBM) to obtain the best prediction of the claims frequency.

In a second stage, the models were streamlined by using different techniques such as Cross Validation or selection methods.

Keywords: Mobile phone insurance, Smartphone, Machine Learning methods, XGBoost, Pricing, Decision Trees, Theft, P&C Insurance, Linear Models, Shapley.

Note de Synthèse

Le marché de la téléphonie mobile voit l'apport d'un flux conséquent de nouveaux modèles dans un secteur hautement concurrentiel dont la progression tarifaire et la démocratisation de l'utilisation des téléphones sont en forte progression.

En effet, en 2020, le nombre de téléphones en circulation dans la France est de 60 millions pour un prix moyen de 420€ soit 128€ de plus que le prix moyen du téléphone en Europe.

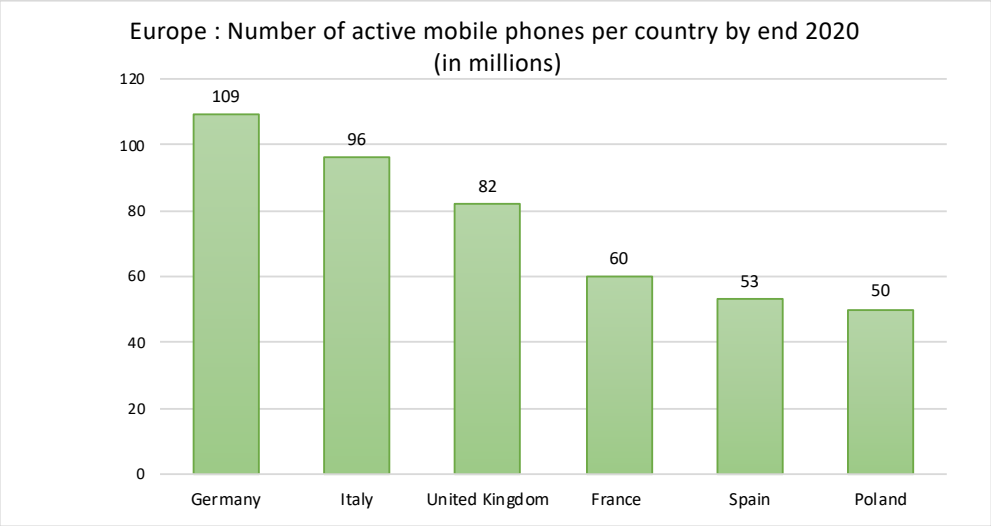


FIGURE 1 : NOMBRE DE TELEPHONES MOBILES (EN MILLIONS) EN EUROPE EN 2020

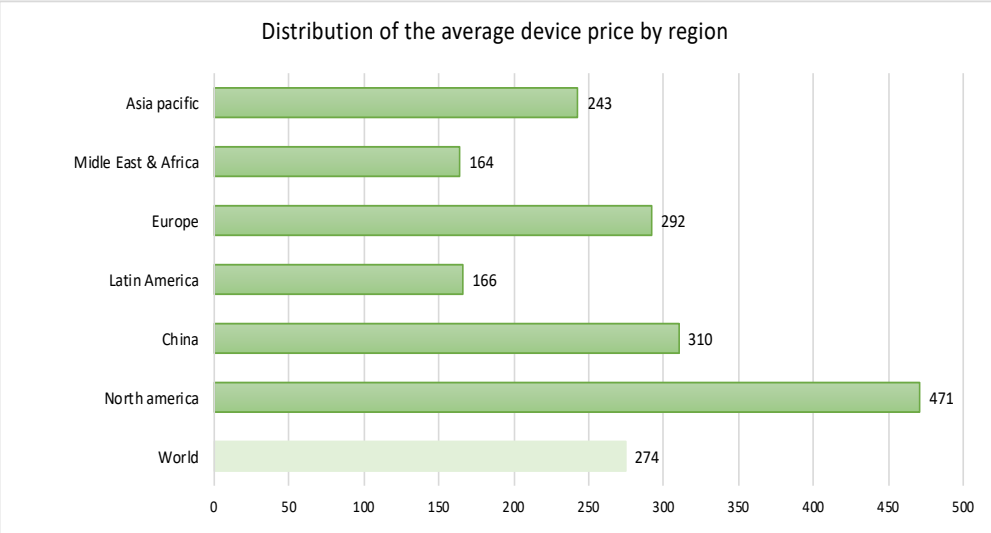


FIGURE 2 : PRIX MOYEN D'UN TELEPHONE PAR REGION DANS LE MONDE

Le périmètre d'étude est défini sur un portefeuille d'assurance vol-casse de téléphones portables interne issu principalement des pays nordiques, limité aux marques Apple, Huawei et Samsung qui représentent plus de 90% de ce portefeuille. En 2020, ces dernières représentent 81% de la vente mondiale dont 6 modèles appartenant au top 10 des meilleures ventes au cours du premier trimestre.

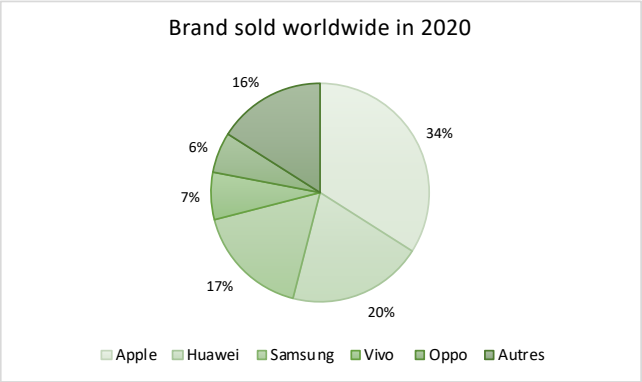


FIGURE 3 : MARQUES VENDUES DANS LE MONDE EN 2020

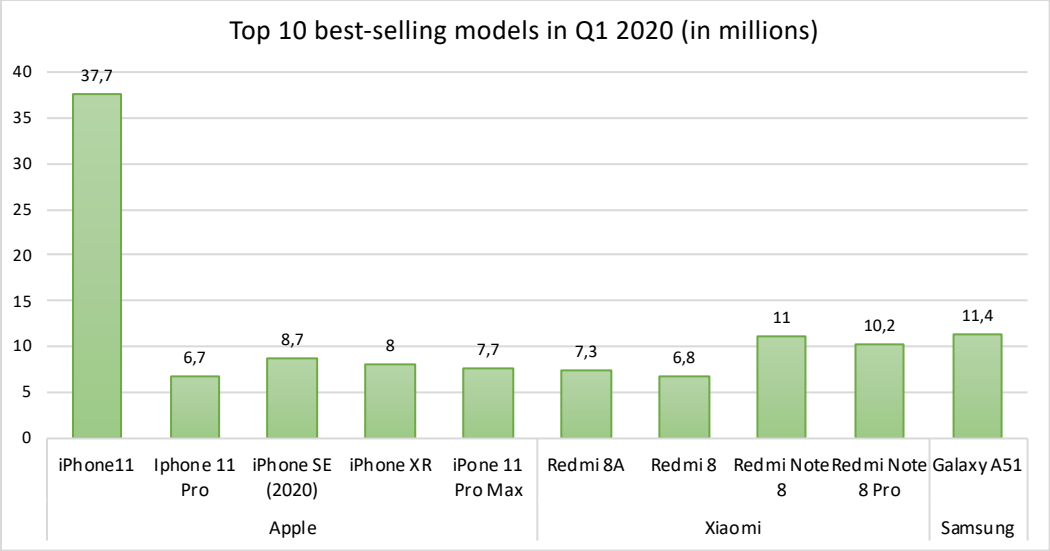


FIGURE 4 : TOP 10 DES MODELES LES PLUS VENDUS EN Q1 2020 (EN MILLIONS)

Le premier trimestre de l'année 2020 a été dominé par la vente d'Apple, suivi par Samsung et Xiaomi, une marque en pleine expansion. Huawei reprendra le dessus lors du quatrième trimestre avec la sortie du Huawei P30 Pro.

L'objectif de ce mémoire est de tester un modèle *Analytics* de prédiction des fréquences de sinistralité tout en analysant l'impact de l'utilisation de variables externes concernant les caractéristiques physiques des téléphones. Le risque sous-jacent est le dommage accidentel faisant partie des assurances affinitaires pour les téléphones portables.

L'étude a soulevé des questions concernant l'utilisation de données externes : quels seraient les apports de ces données et en particulier quels seraient leurs impacts sur la prédiction de la fréquence de sinistralité ? Les données externes utilisées dans l'étude sont exclusivement relatives aux caractéristiques physiques des téléphones mobiles retenus. De nouvelles variables plus intuitives et aisément interprétables ont également été créées à partir des données externes, telles que le ratio de la taille d'écran sur la taille de téléphone.

L'analyse de ces données constitue une part non négligeable du mémoire et a été réalisée sur la garantie principale en assurance téléphonie mobile : la garantie dommage accidentel.

Ce projet au sein de l'Actuariat Corporate a trois buts principaux :

- Expliquer : mieux comprendre les facteurs impactant les fréquences de sinistres par marque et modèle de téléphone portable
- Prédire : améliorer nos modèles de *Pricing* en étant capable de prédire plus précisément la fréquence de sinistres pour les lancements de nouveaux téléphones
- Tester : participer à la recette métier par l'Actuariat de librairies Python créées par le Datalab Cardif telle que l'*Actuarial Pricing Package*

Les données externes nous apportent une connaissance additionnelle physique du produit, il faut donc les exploiter de la meilleure manière possible.

Des modèles XGBoost et LGBM (les deux meilleurs algorithmes de prédiction d'après le magazine Science) sont réalisés afin de mesurer l'apport de ces données externes. Nous mesurons cette apport grâce à la métrique de la variance de Poisson expliquée.

		XGBoost	LGBM
Variance de Poisson expliquée	Modèle basique	0	0
	Modèle tranches de prix	0,7%	0,7%
	Modèle variables internes	6,18%	5,15%
	Modèle variables internes + externes	7,28%	6,92%

Tableau 1 : Comparatif des mesures de predictions par modeles et methodes

L'ajout des données externes à notre portefeuille améliore notre pouvoir de prédiction pour les deux algorithmes analysés. Or nous disposons d'une quantité très importante de variables externes pour analyser notre risque. La sélection des variables à retenir est donc une étape importante.

Deux tris sont réalisés parmi les données internes et externes par la méthode de Shapley puis par une méthode ascendante mesurée par la Variance de Poisson expliquée. Ces derniers nous ont permis de réaliser un modèle dit réduit aussi performant que le modèle contenant toutes les variables internes et externes par l'algorithme XGBoost.

Nous nous lançons par la suite dans le combat contre le sur-apprentissage fréquemment présent pour ces algorithmes. Des modèles de validations croisées sont conçus pour :

- Combattre le sur-apprentissage et évaluer la réelle puissance de prédiction de nos modèles *Machine Learning*
- Tarification de produit en anticipant la sortie d'un nouveau modèle de téléphone

En effet par une méthode précise de validation croisée, nous nous mettons en situation de sortie d'un téléphone déjà disponible dans notre table de données. Nous tentons donc de prédire sa fréquence de sinistralité en écartant ce même modèle de téléphone de notre table d'apprentissage et en utilisant tous nos champs de données internes et externes disponibles afin de tarifer ce produit au mieux en vue de son risque de sinistralité.

Cependant la performance des prédictions n'est pas suffisamment significative afin de tarifer un nouveau produit adéquatement au vu de la fréquence de sinistralité prédite. En effet, dans la situation hypothétique du calcul de la prime pure pour un IPhone 12 prochainement annoncé, il est impossible pour le moment avec les données disponibles dans notre portefeuille et les données externes ajoutées de prédire justement sa future fréquence de sinistralité.

Par la suite, nous réalisons une nouvelle méthode de sélection de variables et un *benchmark* de nos modèles *Machine Learning* par des modèles linéaires généralisés, modèles communément utilisés en actuariat, pour la tarification des risques P&C.

		XGBoost	GLM Lasso
Variance de Poisson expliqué	Modèle variables internes	6,18%	5,14%
	Modèle variables internes + externes	7,28%	7,13%

TABLEAU 2 : COMPARATIF DES MESURES DE PREDICTIONS PAR MODELES ET METHODES

Selon les résultats d'une analyse comparative entre le modèle linéaire généralisé log-Poissonnien (Generalized Linear Model ou GLM) et le modèle d'Extreme Gradient Boosting, les performances des deux modèles restent relativement équivalentes lorsqu'ils

comportent uniquement les données internes, et légèrement meilleures pour les modèles XGBoost lorsque l'on y ajoute les données externes.

Étant donné la quantité importante de variables dans l'étude, l'utilisation d'un XGBoost a semblé plus appropriée car il dispose d'une sélection automatique des variables tout en assurant une bonne gestion des valeurs manquantes. Cette sélection automatique des variables réalisée par l'algorithme XGBoost a été confrontée à d'autres méthodes de sélection dans l'étude ce qui a permis d'attester de sa qualité.

Les variables internes et externes conservées après les différentes étapes de sélection sont incorporées directement dans le modèle XGBoost et permettent de prédire la fréquence de sinistralité des appareils mobiles aussi bien que le modèle composé par toutes les variables.

Le résultat obtenu grâce à la métrique de somme des erreurs absolues est que le gain en précision après incorporation des données internes et externes sélectionnées est de 45% face au modèle basique, 37% face au modèle par tranches de prix et 13% face au modèle composé uniquement de données internes.

		Variance de Poisson expliquée	Somme des erreurs absolues	Gain (%)
XGBoost	Modèle basique	0%	10,4	0%
	Modèle tranches de prix	0,7%	9,5	8%
	Modèle variables internes	6,18%	7,1	32%
	Modèle réduit variables internes + externes	7,29%	5,7	45%

Tableau 3 : Comparatif des mesures de predictions par modeles

Ce premier modèle réalisé fournit des résultats plutôt encourageants puisqu'il est d'ores et déjà plus stable et plus performant que le modèle GLM utilisant uniquement les données internes liées au risque, au partenaire et à l'assuré lui-même. Néanmoins l'apport de données externes dans nos modèles de prédiction n'est pas assez significatif.

A travers les différentes méthodes énoncées précédemment, nous avons pu mieux comprendre les facteurs explicatifs du risque et participer à la recette métier par l'utilisation de différents packages. Finalement en considérant les résultats des études, la variable ayant le plus d'impact sur la prédiction des fréquences de sinistralité est la variable « opérateur téléphonique ». C'est en effet cette variable qui contient les

différentes données comportementales sur l'assuré, telles que son lieu d'habitation, son âge, sa catégorie socioprofessionnelle, parmi d'autres.

Une personne ayant par exemple un profil « Geek » aurait tendance à plus souvent utiliser son téléphone qu'une personne âgée et changerait donc par la même occasion le risque de sinistralité sur l'appareil. Parmi les données externes, le ratio entre la surface de l'écran/la surface de l'appareil, le poids du téléphone et le type de protection de l'écran auraient le plus d'impact sur la fréquence de sinistralité.

Le schéma récapitulatif ci-dessous permet de visualiser les étapes réalisées dans cette étude afin de construire le processus d'intégration des données auxiliaires dans le modèle de prédiction :

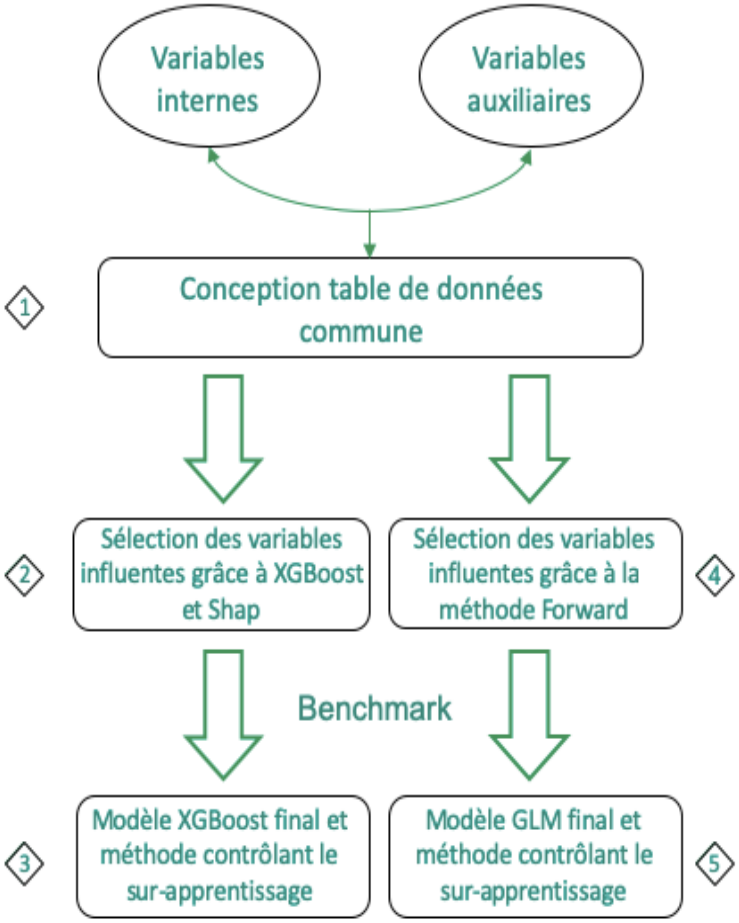


FIGURE 5 : STRUCTURE DU MÉMOIRE

Synthesis

The mobile market is seeing a substantial flow of new models in a highly competitive sector where the price increase and democratization of telephone use are growing strongly.

Indeed, in 2020, the number of mobile phones in circulation in France is 60 million for an average price of €420, i.e. €128 more than the average price of a mobile phone in Europe.

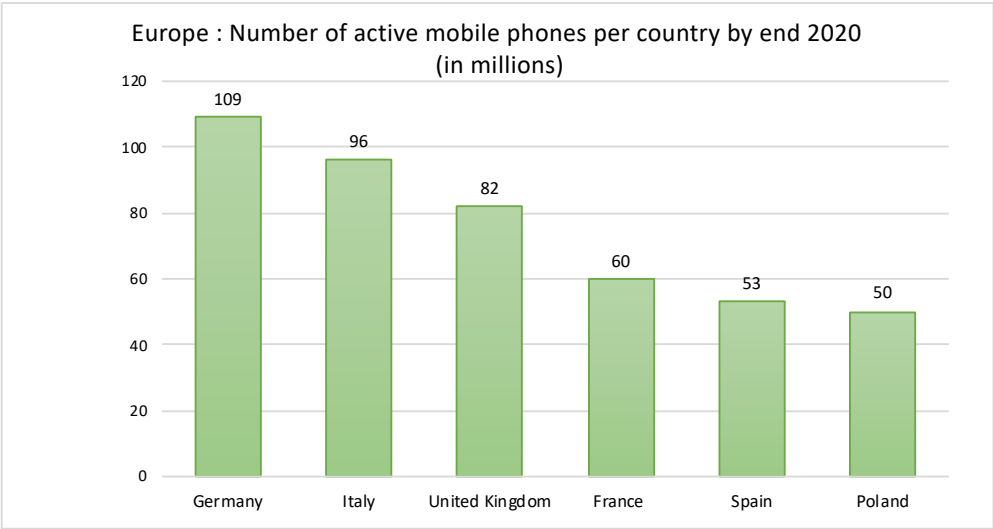


Figure 6 : Number of mobile phones (in millions) in Europe in 2020

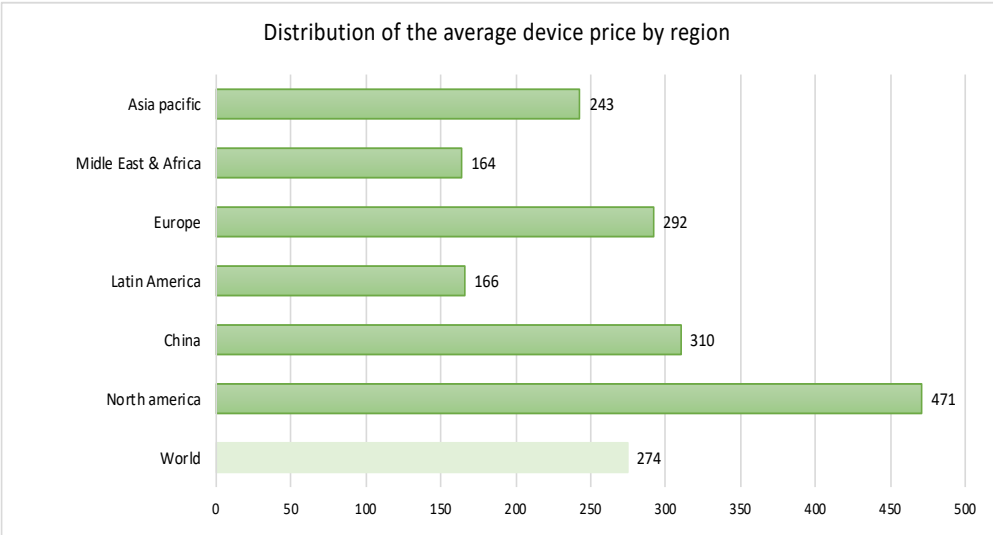


Figure 7 : Average price of a phone per region in the world

We define our scope of study as an internal portfolio mainly from the Nordic countries, limited to Apple, Huawei and Samsung brands, which represent more than 90% of this portfolio. In 2020, these brands will account for 81% of worldwide sales including 6 models in the top 10 best-selling models in the first quarter.

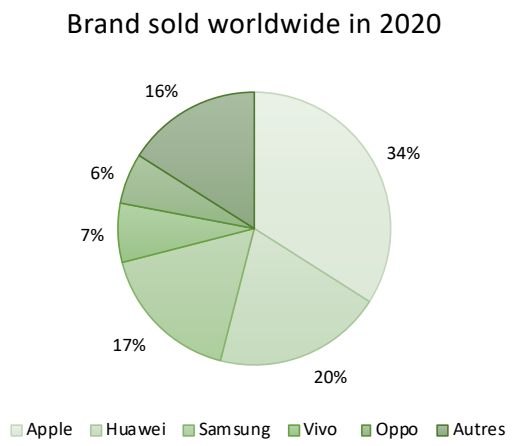


FIGURE 8 : BRAND SOLD WORLDWIDE IN 2020

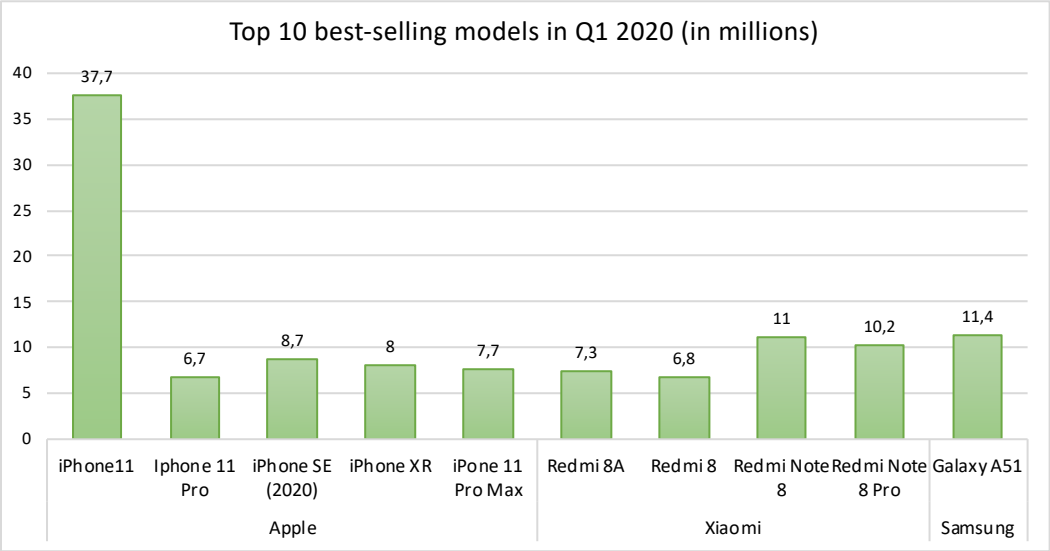


FIGURE 9 : TOP 10 BEST-SELLING MODELS IN Q1 2020 (IN MILLIONS)

The first quarter of the year 2020 was dominated by the sale of Apple, followed by Samsung and Xiaomi which is in full expansion. Huawei regain the upper hand in the fourth quarter with the release of the Huawei P30 Pro.

The objective of this thesis is to test an analytics model for predicting claims frequencies while analyzing the impact of the use of external variables on the physical characteristics of mobile phones. The underlying risk is the accidental damage which is part of the affinity insurance for mobile phones.

The study raised questions about the use of external data : what would be the contributions of these external data and in particular what would be their impact on the prediction of claims frequency ? The external data used in the study are exclusively related to the physical characteristics of the selected mobile phones. New, more intuitive and easily interpretable variables were also created from the external data, such as the ratio of screen size to phone size.

The analysis of these data constitutes a non-negligible part of the paper and was carried out on the main covered risk in mobile insurance : the accidental damage risk.

This project within Corporate Actuarial has three main goals :

- Explain : to better understand the factors impacting claim frequencies by brand and model of mobile phone
- Predict : Improve our pricing models by being able to more accurately predict the frequency of claim for new phone launches.
- Test : participate in the business cycle by Actuarial of Python libraries created by Datalab Cardif such as the Actuarial Pricing Package

External data provides us with additional physical knowledge of the product, so we need to exploit it in the best possible way.

XGBoost and LGBM models (the two best prediction algorithms according to Science magazine) are used to measure the contribution of these external data. We measure their contribution by using the explained Poisson variance metric.

		XGBoost	LGBM
Explained Poisson Variance	Dummy model	0	0
	Price brackets model	0,7%	0,7%
	Internal features model	6,18%	5,15%
	Internal & external features model	7,28%	6,92%

Tableau 4 : Comparative table of prediction scores by models and methods

Adding external data to our portfolio improves our predictive power for both algorithms. However, we have a very large number of external variables to analyze our risk. The selection of the variables to be retained is therefore an important step.

Two sorting are carried out among the internal and external data by the Shapley method and then by a bottom-up method measured by the explained Poisson Variance.

These last ones allowed us to realize a so-called reduced model as efficient as the model containing all the internal and external variables by the XGBoost algorithm.

We then embark on the fight against the overfitting frequently present in Machine Learning algorithms. Cross Validation models are designed for :

- Avoid overfitting and evaluate the real predictive power of our Machine Learning models.
- Product pricing by anticipating the release of a new phone model

Indeed, by a precise method of Cross Validation, we simulate a new device launching by withdrawing it from the available data and then we try to predict its frequency of claims by removing this same model of telephone from our learning table and by using all our available internal and external data in order to price this product the best possible way in view of its risk of claims.

However, the predicted results are not significant enough to price a new product adequately in view of the predicted frequency of claims. Indeed, for the soon to be launched IPhone 12, it is not possible at this time with the data available in our portfolio and the external data added to predict its frequency in an efficient way.

Subsequently, we are implementing a new variable selection method and benchmarking of our Machine Learning models with generalized linear models, models commonly used in actuarial science, on P&C risks.

		XGBoost	GLM Lasso
Explained Poisson Variance	Internal features model	6,18%	5,14%
	Internal & external features model	7,28%	7,13%

Tableau 5 : Comparison of prediction scores by models and methods

According to the results of a comparative analysis between the log-Poissonnian Generalized Linear Model (GLM) and the Extreme Gradient Boosting model, the performance of the two models remains relatively equivalent when only internal data is included, and slightly better for the XGBoost models when external data is added.

Given the large number of variables in the study, the use of an XGBoost seemed more appropriate due to the automatic selection of variables while ensuring good management of missing values. This automatic selection of the variables carried out by the XGBoost algorithm was compared against other selection methods in the study, which made it possible to certify its quality.

The internal and external variables chosen after the different selection steps are added directly into the XGBoost model and allow the prediction of the frequency of claims of mobile devices as well as the model composed by all variables.

The result in term of using absolute error sum metrics is that the gain in accuracy after incorporating the selected internal and external data is 45% compared to the dummy model, 37% compared to the price brackets model and 13% compared to the model composed only of internal data.

		Explained Poisson Variance	Sum of absolute errors	Gain (%)
XGBoost	Dummy model	0%	10,4	0%
	Price brackets model	0,7%	9,5	8%
	Internal features model	6,18%	7,1	32%
	Internal & external features reduced model	7,29%	5,7	45%

Tableau 6 : Comparison of prediction scores by models

This first model produced provides rather encouraging results since it is already more stable and more efficient than the GLM model using only internal data related to the risk, the partner and the insured. Although the external data showed a better performance of the models, such improvement would not be significant enough to justify the operational costs.

Through the different methods outlined above, we have been able to improve our pricing models and participate in the business cycle by using different packages. Finally, the behavioural variables such as the distributor name would show the best performance on the models. It is indeed this variable that contains the various behavioral data on the insured, such as his place of residence, his age, his socioprofessional category...

A person with a "geek" profile, for example, would tend to use their smartphone more often than an elderly person and thus the risk exposure would be different for both users. Among the external data, the ratio between the screen area/phone surface area, the weight of the device and the type of screen protection would have the greatest impact on the frequency of claims.

The summary diagram below illustrates the steps performed in this study to build the process of integrating the auxiliary data into the prediction model :

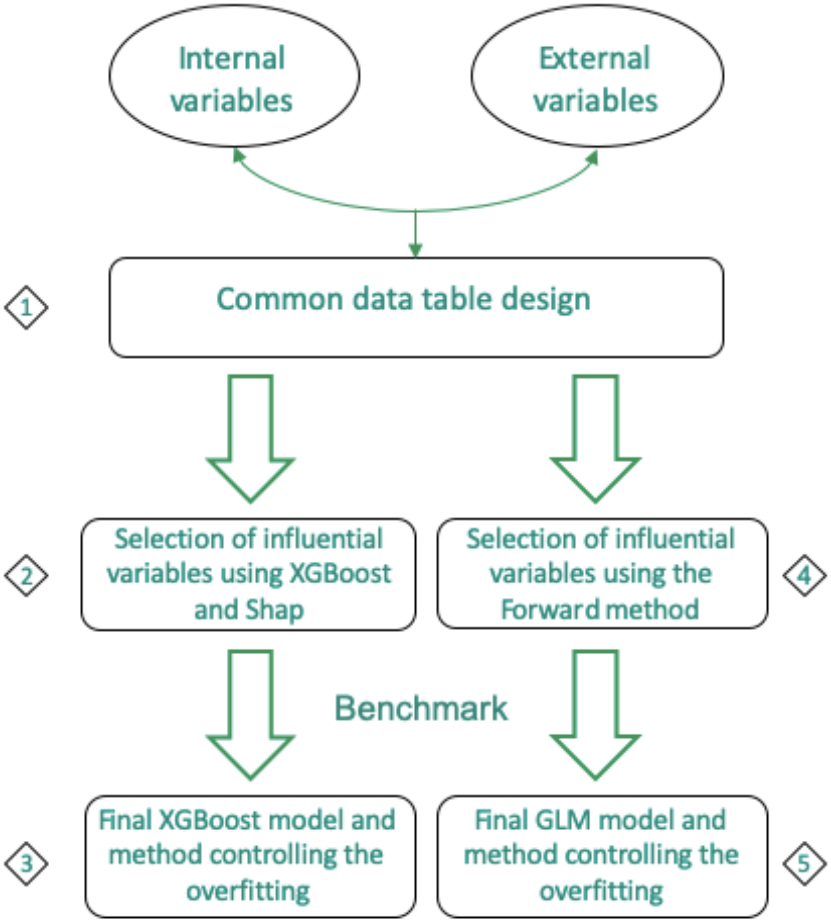


FIGURE 10 : STUDY STRUCTURE

Remerciements

Je tiens à remercier toutes les personnes qui ont contribué au succès de mon alternance et qui m'ont aidé lors de la rédaction de ce rapport. Dans un premier temps, j'adresse mes remerciements à ma tutrice pédagogique Mme Orbert Ben-Taïeb qui m'a beaucoup aidé dans ma recherche d'alternance et m'a aidé dans la préparation aux entretiens pour cette entreprise. Son écoute et ses conseils m'ont permis de cibler mes candidatures et de trouver ce stage qui était en totale adéquation avec mes attentes.

Je tiens à remercier vivement mon maitre d'alternance M. Clément SAILLARD, Head of *Pricing* P&C pour son accueil, le temps passé ensemble et le partage de son expertise au quotidien. Grâce à sa confiance j'ai pu m'accomplir totalement dans mes missions.

Je remercie également toute l'équipe P&C, leur esprit d'équipe et en particulier Maria AGUILERA, qui m'a beaucoup aidé à comprendre les problématiques d'analyse des données. Elle fut d'une aide précieuse dans les moments les plus délicats.

Merci, à Nicolas VASSEUR « manager adjoint » de Clément, pour ses qualités professionnelles mais tout aussi très humaines tout au long de cette alternance.

Enfin, je tiens à remercier toutes les personnes qui m'ont conseillé et relu lors de la rédaction de ce rapport de stage : ma famille, mes amis et camarades de promotion.

Table des matières

Résumé 2
Abstract 3
Note de Synthèse 4
Synthesis 10
Remerciements 16
Introduction 20
Cadre et objectifs de l'étude 22
1.1 BNP Paribas Cardif 22
1.1.1 Création du groupe 22
1.1.2 Ouverture internationale 22
1.1.3 Quelques chiffres 23
1.1.4 L'actuariat Corporate 24
1.2 L'assurance des téléphones portables (MPI selon l'EIOPA) 24
1.3 Présentation du mémoire 26
1.4 Problématiques soulevées 27
1.5 Résultats obtenus 27
Traitement et agrégation des données brutes pour l'étude 29
2.1 Périmètre de modélisation 29
2.2 Partenaire et données internes 30
2.2.1 Présentation générale 30
2.2.2 Description du portefeuille de notre partenaire. 31
2.2.3 Inventaire et données internes 32
2.3 Information auxiliaire considérée 33
2.4 Traitements des données 36
2.4.1 Données internes appartenant à BNP Cardif 36
2.4.2 Information auxiliaire 39
2.5 Jeu de données final 40
Analyse des données et sélection des variables 42
3.1 Le Machine Learning 42

3.1.1 Qu'est-ce que le Machine Learning ? 42
3.1.2 L'apprentissage supervisé ?....44
3.1.3 Le lien Biais – Variance ? 45
3.1.4 Le Sur-apprentissage ? 47
3.1.5 Utilisation de la validation croisée dans les méthodes d'apprentissage supervisé....49
3.2 Méthode d'apprentissage supervisé retenu 52
3.2.1 Les arbres décisionnels....52
3.2.2 Gradient Boosting....54
3.2.3 Extreme Gradient Boosting 56
3.2.4 LightGBM....57
3.2.5 Avantages et inconvénients des modèles 57
3.2.6 Optimisation des paramètres de modèles 59
3.3 Mesure de l'apport des données extérieures par rapport aux données internes 61
3.3.1 Les mesures de performance 61
3.3.2 Mesure de l'apport des données externes....65
3.3.3 Un modèle composé uniquement de variables externes ? 74
3.4 Sélection et description des variables les plus influentes 75
3.4.1 Sélection des variables par l'algorithme XGBoost....76
3.4.2 Mise en œuvre d'une sélection de variables par le diagramme de Shapley 77
3.5 Modèle final et sur-apprentissage 79
3.5.1 Mise en œuvre d'une sélection des variables 79
3.5.2 Cross Validation 83
3.5.2 Ajout de Bruit 85
Benchmark avec la régression des variables influentes et analyse 89
4.1 Modèles linéaires généralisés 89
4.1.1 Modèles linéaires 89
4.1.2 Modèles linéaires généralisés 90
4.2 Pénalisation de la régression....95
4.3 Mise en œuvre d'une sélection de variables....96
4.3.1 Benchmark des modèles Machine Learning....97
4.3.2 Limites du GLM 102
4.4 Mesure de l'apport des données externes....102
Conclusion....105

Bibliographie 107

Liste des figures 109

Liste des tables 111

Glossaire 112

Introduction

Un contrat d'assurance téléphonie couvre plusieurs risques tels que le dommage accidentel, le vol, l'oxydation ou encore le bris des glaces. Afin de faciliter la vente des produits par les distributeurs, la granularité de tarification d'un tel contrat est généralement la tranche de prix du téléphone. Les variables telles que les informations sur les assurés, ainsi que les caractéristiques physiques du téléphone ne sont pas prises en compte dans la segmentation du tarif.

L'étude produite dans ce mémoire a été réalisée au sein de la Direction Actuariat Corporate de BNP Paribas Cardif, dans l'équipe Pricing P&C, le sujet concerne l'assurance téléphonie mobile.

Le présent mémoire a pour ambition de créer et de tester un modèle de prédiction de fréquences de sinistralités tout en analysant l'impact de l'utilisation de nouvelles variables externes concernant les caractéristiques physiques des téléphones.

L'étude se décompose alors en quatre parties qui constituent les différents chapitres :

- Le premier chapitre permet de donner au lecteur une vision claire du contexte dans lequel est réalisé le mémoire en détaillant notamment les étapes de réalisation des modèles *Machine Learning*, GLM et différentes méthodes de calibration des modèles afin d'éviter le sur-apprentissage.

- Le second chapitre traite de la conception de la table de données utilisée pour réaliser cette étude, tout en définissant les variables qui la composent, l'échelle à laquelle elles sont observées et leur provenance. Cette table résultant de l'agrégation, après retraitements, d'une table de données internes et d'une table de données externes.

- Dans le troisième chapitre, l'analyse des données externes est réalisée. L'accent est notamment mis sur l'étude de l'apport des caractéristiques physiques des téléphones mobiles sur la prédiction de la fréquence de sinistralité. L'ensemble des analyses des données externes est réalisé sur la garantie de dommage accidentel. Ce chapitre permet également d'analyser et de présélectionner les variables influant sur la survenance d'un sinistre dommage accidentel. Ces variables sont ensuite intégrées dans le modèle final.

- Finalement dans le quatrième et dernier chapitre, une comparaison entre le modèle linéaire généralisé log-Poissonnien (Generalized Linear Model ou GLM) et le modèle d'Extreme Gradient Tree Boosting (XGBoost) est réalisée. Le modèle linéaire généralisé est le modèle communément utilisé en assurances de dommages, mais le modèle d'Extreme Gradient Tree Boosting représente une bonne alternative car il permet de faire face à une quantité massive de variables disponibles pour réaliser l'étude. C'est pourquoi ici le modèle XGBoost est utilisé en tant que modèle de prédiction. Les variables internes et externes présélectionnées sont donc intégrées dans le modèle afin de prédire la fréquence de survenance d'un sinistre dommage accidentel sur les appareils mobiles.

Nous tenons à préciser que pour la suite de l'étude tous les chiffres ont été transformés ou bien non énoncés pour des questions de confidentialité. Nous ne parlerons pas de fréquence mais d'indicateur de fréquence. Cette transformation ne détériore pas la validité des résultats et n'affecte pas la conclusion de l'étude

Cadre et objectifs de l'étude

1.1 BNP Paribas Cardif

L'étude faisant l'objet de ce mémoire a été réalisé par la direction Actuariat Corporate Protection de BNP Paribas Cardif. Ce chapitre a pour but de présenter brièvement BNP Paribas Cardif.

1.1.1 Création du groupe

En 1973, la Compagnie Bancaire obtient l'agrément et crée une société d'assurance vie, Cardif, spécialisée dans l'assurance de personnes, l'épargne et la retraite. L'activité de Cardif se table sur la création d'un réseau de distribution externe. Les premiers partenaires de Cardif sont Cetelem, Crédit du Nord ou encore Cortal. Cardif débute son développement à l'international dès 1989 avec son implantation en Italie en Belgique et en Espagne.

Ce développement rapide à l'international via le développement de partenariats se poursuit les années suivantes, Cardif est implanté dans 20 pays en 1997. Après la fusion de la Compagnie Bancaire et de Paribas en 1999 et puis celle de BNP et Paribas en 2000, le nouveau groupe décide de regrouper ses sociétés d'assurances sous le nom de BNP Paribas Assurance. La filiale regroupe Cardif, Natio Vie et Natio Assurance, les deux assurances IARD et Vie créées par BNP. En 2010, les différentes entités de BNP Paribas Assurance sont regroupées sous la marque Bno Paribas Cardif.

1.1.2 Ouverture internationale

Les activités de Cardif sont géographiquement diversifiées. En effet, avec des implantations dans 33 pays, BNP Paribas Cardif est présent en Europe, Asie et Amérique latine. BNP Paribas Cardif est le deuxième assureur vie français le plus international après Axa. Possédant un important réseau de distribution avec plus de 500 partenaires à travers le monde, BNP Paribas Cardif est l'assureur comptant le plus de partenariats en banque-assurance dans le monde.

FIGURE 11 : PRESENCE INTERNATIONALE DE CARDIF

1.1.3 Quelques chiffres

Nous avons ci-dessous une carte de monde donnant la répartition du chiffre d'affaires selon les zones géographiques où est implanté BNP Paribas Cardif. L'activité "Epargne" regroupe la retraite et l'épargne tandis que l'activité "Protection" regroupe la prévoyance, la santé et le IARD (incendies, accidents et risques divers).

On remarque le poids important du marché français. En effet, la France représente à elle seule 42% du chiffre d'affaires du groupe. Nous pouvons voir ci-dessous la ventilation du chiffre d'affaires du groupe par activité et par réseau de distribution :

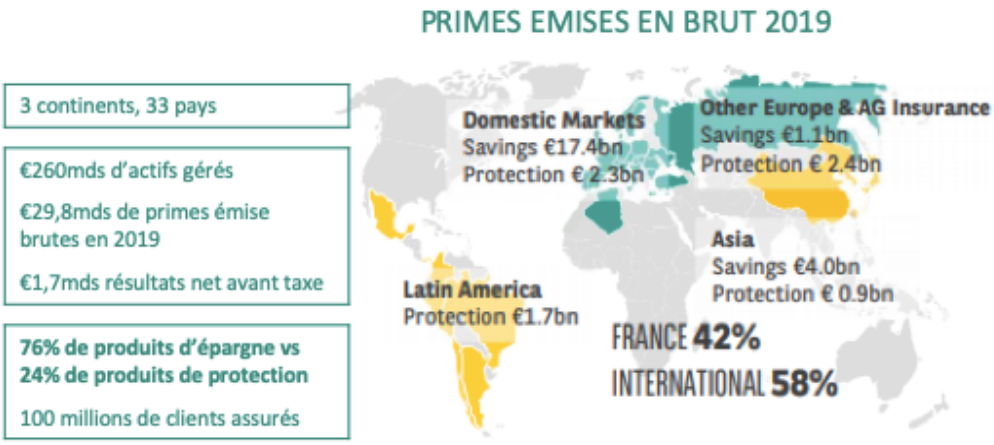


FIGURE 12 : CHIFFRE D'AFFAIRES PAR ACTIVITE

`

1.1.4 L'actuariat Corporate

L'actuariat Corporate du groupe BNP Paribas Cardif est principalement divisé en quatre directions :

- Protection : produits prévoyance, santé et IARD
- Transfert de risques
- M&A : Fusions et acquisitions
- Savings : ALM et produits épargne/retraite

La direction Protection est quant à elle divisée en 5 équipes :

- Clôture et Surveillance des Risques ;
- Règlementation, Reporting, Solvabilité II ;
- Transverse Actuariat Protection ;
- Tarification Prévoyance et Santé ;
- Tarification P&C : c'est dans cette dernière que j'ai pu effectuer l'étude sur laquelle porte ce mémoire.

Cette équipe a pour rôle de superviser et d'assister tous les pays du groupe BNP Paribas Cardif.

Les principales responsabilités de cette équipe sont :

- Définir et garantir la politique de souscription ainsi que la rentabilité technique de chaque pays ;
- Contribuer à la création de nouveaux produits et à leur validation ;
- Assurer le bon suivi des *Guidelines* ;
- Soutien dans la tarification des accords dit « *Out of the box* » ou jugés stratégique par le groupe.

1.2 L'assurance des téléphones portables (MPI selon l'EIOPA)

Les garanties de l'assurance des appareils mobiles proposent pour la plupart une garantie casse, une garantie vol et une garantie oxydation. En cas de sinistre, la prise en charge varie selon les conditions générales de la notice d'assurance. L'assuré peut ainsi bénéficier d'une réparation, d'un possible prêt de smartphone, d'une indemnisation, du remplacement du téléphone par un téléphone neuf ou bien reconditionné.

Pour cela, en France, il est généralement nécessaire de déclarer le sinistre auprès de l'assureur dans les 5 jours qui suivent le sinistre, ou dans les 2 jours s'il s'agit d'un vol.

Ce mémoire traite uniquement de l'assurance de dommage accidentel pour les téléphones portables. L'assurance dommage accidentel des téléphones portables couvre les sinistres non intentionnels du type : bris d'écran, oxydation, les casses affectant les boutons du téléphone, celles survenant au niveau du micro ou encore celles impactant le port de charge du téléphone portable. Parfois, la maladresse et la perte peuvent aussi être incluses.

FIGURE 13: COUVERTURE DOMMAGE ACCIDENTEL

Un code python sur notre table de données internes et plus précisément sur les champs référents de description des sinistres nous a permis de créer un « nuage de mots » reflétant les causes de sinistres et les types de sinistres les plus récurrents avec une taille de police variant d'un mot à un autre selon sa fréquence d'apparition : la taille du mot dans le nuage reflète son apparition sur ce champ, mettant en avant les causes les plus communes de sinistralité.

Les résultats de cet exemple montrent les divers types et causes de sinistres susceptibles de survenir avec la mise en relief du scénario le plus courant qui est, dans ce cas-là, la casse et le bris d'écran.

FIGURE 14 : WORD CLOUD DES CAUSES DE SINISTRE DU PORTEFEUILLE

1.3 Présentation du mémoire

L'étude a été réalisée dans la Direction Corporate de BNP Paribas Cardif, au sein de la branche Protection dans l'équipe P&C sur une problématique orientée assurance de téléphones portables.

L'objectif de ce mémoire est d'établir une méthode alternative de construction d'un modèle *Machine Learning*, qui réponde à l'exigence de suivi du risque de sinistralité par l'apport de données externes. L'étude est réalisée sur un ensemble de données d'environ X polices relatives aux contrats téléphonique présents dans le portefeuille Nordique depuis Octobre 2014. Le périmètre des données utilisées pour cette étude est intégralement détaillé dans le chapitre 2.

Par ailleurs, ce mémoire de *Data Science* au profit de l'actuariat porte une ambition, outre de répondre à la problématique métier de construction d'un modèle prédictif, de définir une méthode permettant l'exploitation de l'information externe qui est amenée à être développée par les autres entités de BNP Paribas Cardif.

Un des traitements des données apporté a permis d'évaluer l'ensemble des champs auxiliaires à disposition, de les traiter et de les joindre avec l'ensemble de données des contrats du partenaire. L'ensemble de données final contient plus de 50 champs d'informations à exploiter.

La démarche consiste ensuite à :

1. Établir un modèle prédictif du risque en fonction des données observées sur les fréquences de sinistralité. Dans cette étude, les caractéristiques des téléphones sont intégrées dans les modèles tarifaires de façon directe dans la régression (sélection des variables les plus influentes par Gradient Boosting). Une mesure est faite sur l'apport de ces données par rapport aux données agrégées.
2. Calibrer les modèles *Machine Learning* afin de combattre le phénomène de sur-apprentissage.
3. Benchmark des modèles *Machine Learning* par les modèles linéaires généralisés auxquels nous appliquons une pénalisation par la méthode de Lasso.

1.4 Problématiques soulevées

Durant la réalisation de l'étude faite dans le présent mémoire, plusieurs problématiques ont été soulevées. Ainsi, nous tentons de répondre à ces interrogations parmi lesquelles nous trouvons notamment :

- Quels facteurs pourraient avoir un impact sur la fréquence de casse de téléphones portables ?

- L'utilisation d'informations auxiliaires contribuerait-elle à une meilleure estimation de la fréquence de sinistralité ?

- Y-a-t-il un gain à l'utilisation de méthodes non linéaires de *Machine Learning* pour la tarification par rapport aux techniques usuellement utilisées ?

- Comment identifier, éviter ou bien combattre le sur-apprentissage des modèles non-linéaires ?

- Sommes-nous capables de prédire la fréquence de sinistralité d'un nouveau modèle de téléphone avant sa mise sur le marché ?

1.5 Résultats obtenus

La réalisation de ce mémoire nous permet de mettre en évidence le pouvoir prédictif de la fréquence de sinistralité de téléphone portable. En effet, le pouvoir de

prédiction utilisant les méthodes de *Machine Learning* est meilleur que celui des modèles linéaires généralisés.

Les modèles proposés dans ce mémoire suggèrent une approche différente dans la tarification du produit. Tarifer selon la tranche du prix de l'appareil n'est pas le meilleur modèle à appliquer.

Les informations auxiliaires ayant le plus d'impact sur nos modèles sont le ratio calculé par la taille d'écran sur la taille du téléphone, le poids du téléphone et le type de protection dont le téléphone dispose. Même si ces variables externes améliorent le pouvoir de prédiction, elles ne sont pas suffisantes pour effectuer la tarification du portefeuille.

Par ailleurs, la variable la plus importante de nos modèles reste une donnée interne et correspond au partenaire de distribution. L'information sur le partenaire de distribution résulte de l'information comportementale de l'assuré, tel que son lieu d'habitation, son âge, sa catégorie socioprofessionnelle parmi d'autres.

Ainsi l'étude révèle que des données comportementales (pays, partenaire de distribution) ont plus d'impact sur la prédiction de fréquence de sinistralité que les données auxiliaires représentant les caractéristiques physiques du téléphone portable. Elles permettent de nettement améliorer la performance de nos modèles prédictif.

On suppose que l'apport de nouvelles données comportementales de clients (âge, activité, métier etc.) permettrait d'améliorer significativement le pouvoir prédictif du modèle.

Traitement et agrégation des données brutes pour l'étude

2.1 Périmètre de modélisation

Lors de la tarification d'un contrat, plusieurs étapes sont nécessaires pour calculer la prime réellement payée par l'assuré. La première étape consiste à calculer l'espérance du risque que l'on appelle prime pure. Une fois la prime pure obtenue, la prime technique est obtenue en ajoutant les chargements de gestion et d'acquisition permettant de financer les coûts d'acquisition et d'administration supportés par l'assureur.

Un chargement de sécurité peut également être intégré, ce qui permet de pouvoir résister à la volatilité naturelle des sinistres. C'est pourquoi ce chargement peut, par exemple, être proportionnel à la variance du sinistre. Finalement, la prime technique peut être modifiée en fonction de la politique commerciale de la compagnie d'assurance pour aboutir à la prime commerciale. L'assurance non-vie se détache particulièrement de l'assurance vie de par la survenance même du sinistre mais également par le coût de ce sinistre.

Nous considérons donc le modèle collectif suivant :

$$\mathrm{S} = \begin{cases} \sum_{i=1}^{N} X_i & si\ N > 0, \\ 0\ sinon \end{cases}$$

Avec :

- S la charge de sinistres,
- N la variable aléatoire à valeurs entières représentant le nombre de sinistres,
- La suite des variables aléatoires réelles X_i pour $i = 1, \dots, N$ représentant les coûts des sinistres.

Ce modèle repose sur les hypothèses suivantes :

- Les risques considérés sont des risques homogènes,
- Les variables N et les $(X_i)_{\ i\geq 1}$ sont indépendantes,
- Les variables aléatoires $(X_i)_{\ i\geq 1}$ suivent la même loi.

Si les hypothèses sont respectées, ce modèle à l'avantage de fournir une prime pure relativement simple à calculer puisque nous avons alors l'égalité suivante :

$$\boldsymbol{E}[S] = \boldsymbol{E}[N] \times \boldsymbol{E}[X]$$

L'étude proposée dans ce mémoire porte sur la modélisation du nombre de sinistres N sur des contrats d'assurance appareils électroniques (téléphones, tablettes, e-books). Seuls les contrats concernant les téléphones portables sont conservés afin d'avoir des contrats ayant un risque homogène. Les passages à la prime pure et aux primes techniques et commerciales ne seront pas abordés dans ce mémoire qui traite uniquement du suivi du risque de sinistralité afin d'intégrer des données externes dans la modélisation de la fréquence de survenance d'un sinistre, analyser la performance des modèles basés sur des méthodes de *Machine Learning* et d'autres problématiques soulevées durant l'étude.

La fréquence de survenance d'un sinistre correspond au nombre de sinistres avérés survenant durant la période où le contrat est effectif (par exemple, si le client est assuré depuis 3 mois, la période d'exposition est égale à 0.25, soit un quart d'année). Elle se calcule de la manière suivante :

$$Fréquence = \frac{Nombre\ de\ sinistres}{Période\ d'exposition\ (années)}$$

La garantie qui sera abordée dans l'étude sera uniquement la garantie de dommage accidentel. Celle-ci permettra de mettre en évidence le gain de l'apport des données externes et de comparer les performances de différentes méthodes de prédiction.

2.2 Partenaire et données internes

Dans cette partie nous allons introduire le portefeuille à travers lequel nous avons effectué notre étude générale. Par accord confidentiel, il n'est pas possible de citer le nom du partenaire ni les chiffres relatifs à celui-ci.

2.2.1 Présentation générale

Notre partenaire est un gestionnaire de sinistres spécialisé en assurance dommage accidentel des téléphones portables, c'est un broker Américain implanté récemment en Europe. Sur son site officiel, il propose des couvertures adaptées à différents appareils. Nous retrouvons des téléphones portables, ordinateurs et tablettes.

L'assurance proposée couvre le dommage accidentel pour un téléphone neuf comme utilisé auparavant. Deux couvertures sont proposées : une première où la sinistralité est liée à un unique produit, la seconde concerne une couverture familiale lié à un maximum de 4 téléphones portables et coûte plus cher que la première.

Le type de service proposé par ce dernier est le remplacement du téléphone portable et la proposition d'un téléphone de remplacement reconditionné à son assuré dans un délai dépendant du type de contrat après déclaration du sinistre par le client.

Pour un processus classique, en France il faudrait que le client attende de 5 à 7 jours, à compter du moment d'envoi de son téléphone, le temps que ce dernier soit réparé.

Voici ci-dessous tout le cycle entre notre partenaire et l'assuré en cas de survenance d'un sinistre ou dommage accidentel du téléphone portable :

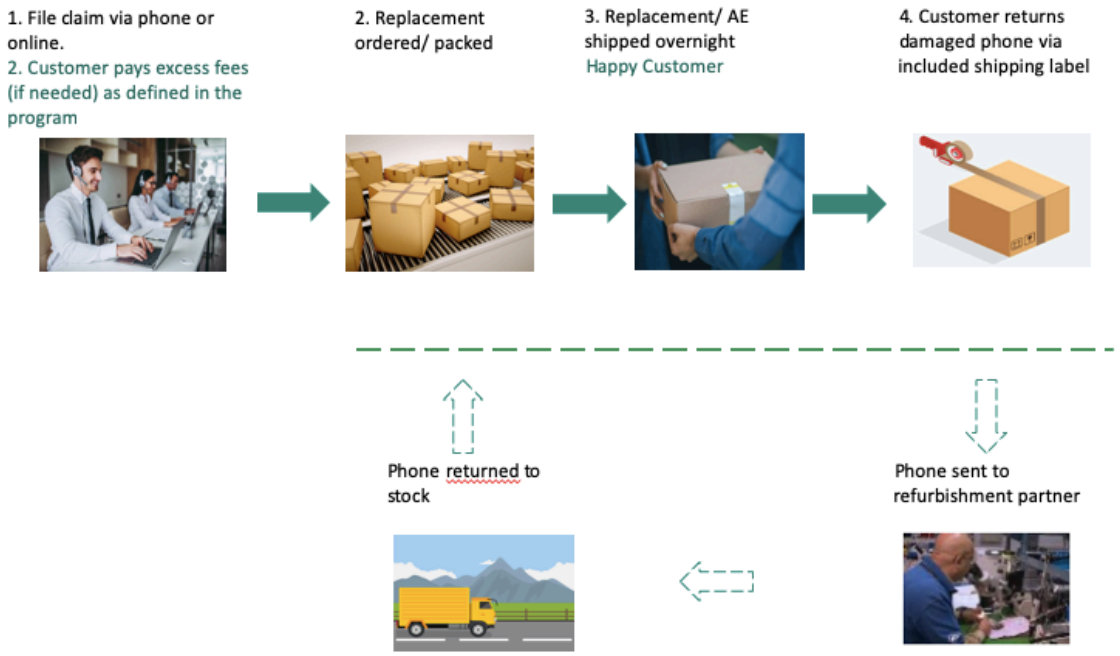
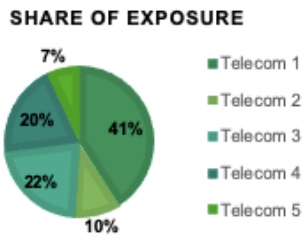


FIGURE 15 : CYCLE D'UN SINISTRE ENTRE UN ASSURE ET NOTRE PARTENAIRE

2.2.2 Description du portefeuille de notre partenaire.

Le périmètre de notre partenariat se centre autour de l'Europe avec trois principaux pays la Finlande, le Danemark et la Suède ; avec 5 opérateurs Télécom 1, Télécom 2, Télécom 3, Télécom 4 et Télécom 5 ; d'où la consistance du portefeuille sur lequel l'étude va être menée avec environ X contrats de 2 ans avec des primes mensuelles. Ci-dessous quelques premières statistiques sur ce portefeuille :



Country	Program	Start of production
Finland	Telecom 1	Dec-14
	Telecom 2	Dec-16
Denmark	Telecom 3	Dec-15
	Telecom 4	Dec-15
Sweden	Telecom 5	Dec-17

FIGURE 16 : EXPOSITION ET FREQUENCE DE SINISTRALITE PAR TELECOM

Environ 90% de l'exposition totale est répartie sur 3 marques :

- ✓ Apple (45%), Samsung (27%), Huawei (15%)
    - 45 modèles représentent 90% de ces 3 marques (environ 80% de l'ensemble total)
    - 24 modèles représentent 80% de ces 3 marques (environ 75% de l'ensemble total)

2.2.3 Inventaire et données internes

Les données internes proviennent de notre partenaire ayant un portefeuille représenté par quatre ensembles de données sous format csv.

Ce sont les tables des :

- Primes
- Sinistres
- Ventes
- Annulations

Entre les champs disponibles, nous y trouvons principalement :

- Opérateur : représente les cinq différents opérateurs décrits antérieurement.
- Marque : il s'agit de la marque de téléphone, la majorité étant composé par Apple, Samsung, Huawei.
- Modèle : ce champ est du type « Free text » rempli par le vendeur. Par conséquent, un modèle unique de téléphone pouvait être renseigné de différentes manières (Par exemple un IPhone 6 peut être écrit : Iphones 6, Iphhone 6, Ipohne 6).
- Inception date : date du début du contrat.
- Prime : montant de la prime mensuelle.
- Devises : $\begin{cases} EUR \\ Couronnes\ dannoises\ Suédoise \end{cases}$

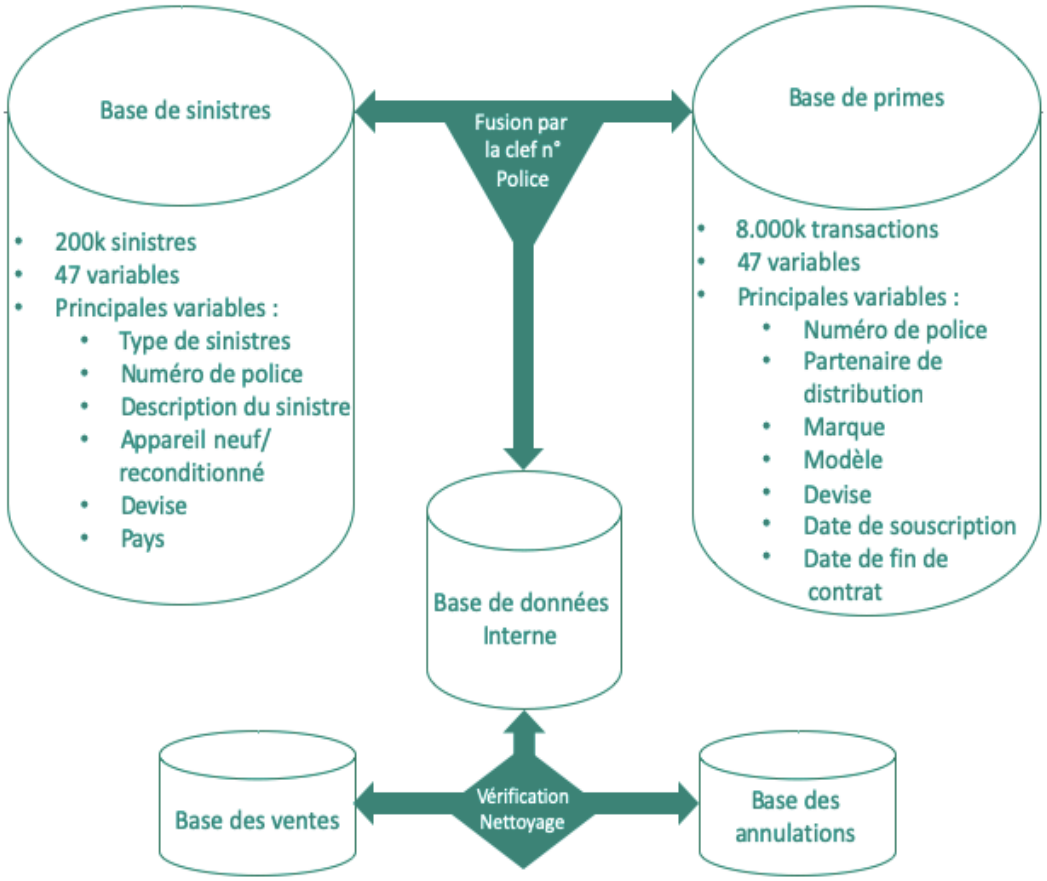


FIGURE 17 : CONSTRUCTION DE LA TABLE INTERNE FINALE

Comme le montre le schéma ci-dessus, la jointure effectuée entre la table des sinistres et la table des primes avec le numéro de police comme clé de jointure a permis de constituer la table interne finale de l'étude.

Concernant les deux tables restantes : celle des ventes et celle des annulations, elles ont permis de détecter certaines incohérences et de faire certaines vérifications dans cette table interne finale (le détail de ces opérations sera explicité dans une autre partie de ce rapport).

2.3 Information auxiliaire considérée

GSMArena est un site de comparateur de téléphones portables, tablettes, ordinateurs portables et montres connectées. L'ensemble des caractéristiques physiques de ceux-ci a été extrait de ce site jusqu'en 2020, cette base externe contient donc aussi tous les modèles récents de téléphones portables.

Le jeu de données contenant des informations auxiliaires aux téléphones provient donc de GSMArena et est disponible sur un site nommé Kaggle, un site de compétition et d'échange principalement autour des problématiques *Machine Learning*.

Cette table contient à peu près 8 000 appareils dont la grande majorité est composé de téléphones portables avec 86 caractéristiques différentes dont les grands volets sont présents dans le schéma ci-dessous.

GSMArena.com	
NETWORK	2G, GPRS, Edge
LAUNCH	Announced Status
BODY	Dimensions Weight Build SIM
DISPLAY	Type Size Resolution Protection
PLATFORM	OS Chipset CPU GPU
MEMORY	Card slot Internal
MAIN/SELFIE CAMERA	Triple Features Video
SOUND	Loudspeaker 3.5mm jack
COMMS	WLAN Bluetooth GPS NFC Radio USB
MISC	Colors Price
TESTS	Performance Camera Loudspeaker Audio quality Battery life

TABLEAU 7 : CARACTERISTIQUES DE LA BASE GSMARENA

Dans un premier temps nous décidons de traiter les valeurs manquantes. Ainsi toute variable ayant un pourcentage de valeurs manquantes dites « NaN » supérieures à 60% est effacée de notre table de données.

Afin de simplifier notre tâche nous décidons de visualiser les valeurs manquantes à travers un package disponible en *Open Source* du nom de Seaborn.

Voici-ci-dessous le graphique nous affichant les valeurs manquantes de la table de données :

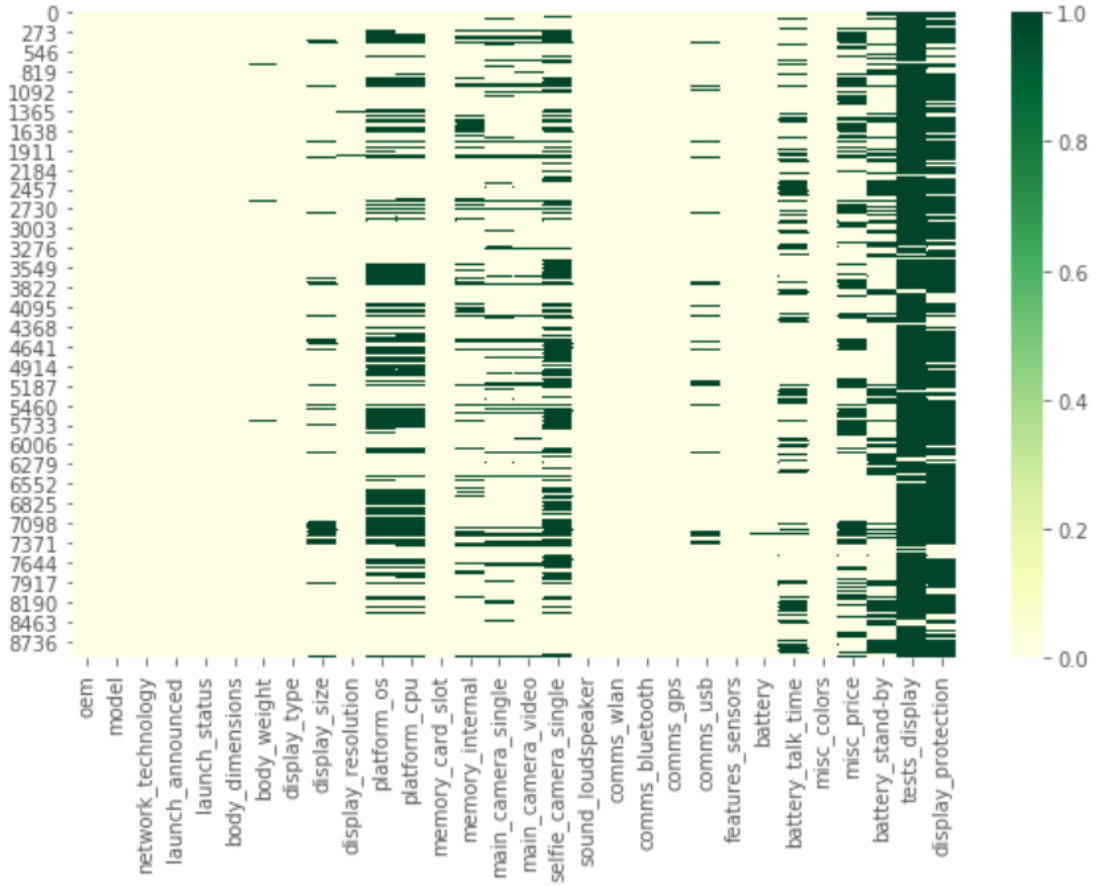


FIGURE 18 : GRAPHIQUE DES VALEURS MANQUANTES DE LA TABLE EXTERNE

Ce graphique reste très simple à lire, nous pouvons voir que chaque nuance de vert correspond à un champ manquant. La variable tests_display est majoritairement colorié en vert, son taux de champs manquants est élevé et bien supérieur à 60%.

La variable display_protection est jugé importante dans notre étude de la fréquence de sinistralité, nous décidons ainsi de prendre le temps de remplir les champs manquant par type de modèle et marque de téléphone portable.

Les variables retenues et considérées intéressantes pour l'analyse sont :

- Launch announced : correspond à l'année d'annonce du téléphone.
- Body Dimensions : correspond aux dimensions du téléphone portable.
- Body Weight : correspond au poids du téléphone en grammes.
- Display type: correspond au type d'écran (LCD, AMOLED, RETINA).
- Display size : correspond à la taille de l'écran en inches.

- Display resolution : correspond à la résolution de l'écran en pixels.
- Display protection: correspond aux différentes protections d'écran (ion-strengthened glass, corning gorilla glass, scratch resistant glass).
- Battery : batterie amovible ou non.

2.4 Traitements des données

2.4.1 Données internes appartenant à BNP Cardif

Dans cette partie, nous allons décrire les différentes étapes et modifications apportés aux données issues du partenaire récemment implanté dans la zone Europe.

2.4.1.1 Création d'un tableau de correspondances par modèle de téléphone

Uniformiser tous les noms des marques et modèles des téléphones portables afin de retrouver un modèle unique par ligne est la première étape de notre étude. En effet, ce champ est clé pour faire la jointure avec les données GSMArena.

Je tiens à rappeler qu'une précédente étude a été effectuée au sein de Cardif sur ce Mapping, le périmètre est restreint sur les trois marques Apple, Huawei et Samsung (90% du portefeuille) uniquement sur les téléphones portables. C'est-à-dire que ce tableau a été établi uniquement sur ces trois marques. Pour les autres modèles et marques, nous les avons classifiés comme Others.

De ce fait, la création de ce tableau de correspondances a requis une quantité considérable de temps, sachant qu'un modèle de téléphone peut être renseigné de 200 façons différentes.

Une fois ce travail effectué sous EXCEL, cette table a de nouveau été importée sous python et jointe à la table interne (Primes + Sinistres) par le biais des variables « Brand » et « Model » afin que pour chaque police soit approprié de la marque, le modèle corrigé plus la variable Brand1 – Model1. Finalement, nous avons 4 Marques : APPLE, HUAWEI, SAMSUNG et OTHERS avec tous les modèles des 4 premières marques uniformisés comme l'exemple ci-dessous :

	A	B	C	D	E	F
1	Brand	Model	Brand1	Model1	Brand1 - Model1	
2	APPLE	IPHONE 6	APPLE	IPHONE 6	APPLE - IPHONE 6	
3	APPLE	IPHONE 6 GOLD	APPLE	IPHONE 6	APPLE - IPHONE 6	
4	APPLE	APPLE IPHONE 7 32 GB	APPLE	IPHONE 7	APPLE - IPHONE 7	
5	APPLE	IPHONE 6 16GB	APPLE	IPHONE 6	APPLE - IPHONE 6	
6	APPLE/APPLE	MOBILNÃ TELEFON APPLE IPHONE 7 PLUS	APPLE	IPHONE 7 PL	APPLE - IPHONE 7 PLUS	
7	APPLE/APPLE	MOBILNÃ TELEFON APPLE IPHONE 6S 32G	APPLE	IPHONE 6S	APPLE - IPHONE 6S	

FIGURE 19 : TABLE DE CORRESPONDANCES DITE MAPPING

2.4.1.2 Traitements et création de variables

Chercher à créer de nouvelles variables à partir des données à notre disposition permet d'exploiter pleinement la richesse des données. Pour profiter entièrement de toutes les données à disposition, il est nécessaire de prendre possession de la table et de travailler les données de manière optimale.

L'étude portant sur le suivi de la fréquence de sinistralité pour chaque contrat, il nous fallait créer une variable définissant notre fréquence de sinistralité. Pour l'obtenir, nous avions besoin de renseignements sur le nombre de sinistres et sur l'exposition.

Afin de calculer la fréquence de sinistralité par contrat, il fallait calculer en amont l'exposition qui représente la durée de vie du contrat.

Start date précédemment citée dans l'inventaire des données internes (provenant de la table des primes) est une variable créé en émettant l'hypothèse que la date de début du contrat est le minimum entre trois variables :

- Inception date.
- Transaction date.
- Effective date.

Ce choix étant dû aux incohérences rencontrées lorsqu'une de ces variables était datée après la date de fin de contrat. Par la suite, il s'est avéré qu'une partie de ces incohérences correspondait à des contrats annulés qui furent écartés de notre étude.

La variable End date est égale au minimum entre le champ date termination et date evalutation.

Finalement l'exposition est égale à :

$$Exposition \ = \frac{(\ end\ date - \ start\ date\ )}{365{,}25}$$

Une fois l'obtention de dates cohérentes et traitées en format de dates sous Python, nous créons la variable d'exposition annuelle du contrat qui correspond à la durée totale de l'engagement de l'assureur envers l'assuré. Ainsi un contrat qui dura 22 mois, aura une exposition égale à environ 1,81.

Le deuxième traitement concerne le nombre de sinistres. Ce champ a été calculé comme l'agrégation des sinistres individuels

Ainsi, après création de ces deux dernières variables, nous sommes en capacité de calculer notre fréquence de sinistralité par contrat :

$$\textit{Fréquence de sinistralité} \ = \frac{\textit{Nombre de sinistre}}{\textit{Exposition}}$$

Finalement, le coût moyen des sinistres par modèle de téléphone a été calculé en suivant les étapes ci-dessous :

- Nous commençons par recoder la variable devise, en couronne danoise, couronne suédoise ou euro selon la provenance géographique de l'opérateur (variable disponible pour toutes les observations).

- Nous créons une variable paid qui représente le maximum entre la variable paid_this_time et 0 afin de corriger cette variable qui contenait des valeurs négatives.

- Avec le traitement de la variable paid, nous créons la variable claim_cost et calcule le coût du sinistre qui représente la somme de « shiping_cost », « paid » et « excess_value_applied_to_the_claim ».

L'information sur le coût du sinistre étant disponible, nous avons uniformisé les coûts en créant une autre variable nommée claim_cost_eur. Nous avons obtenu cette dernière en multipliant tous les coûts de sinistres par le taux de change adéquat.

2.4.2 Information auxiliaire

Cette partie traite de l'ensemble des données externes disponibles en fournissant un inventaire des variables dont l'influence est testée dans ce mémoire.

2.4.2.1 Tableau de correspondances

Un travail similaire à celui décrit dans la section 2.4.1.1 sur les données GSMArena recueillies par le biais de Kaggle a été effectué afin de le préparer à la jointure avec les données du portefeuille.

Le premier travail qui a été effectué sur cette table externe, est l'uniformisation de toutes les marques et modèles des téléphones portables de telle sorte qu'on ne perde pas d'informations dans la phase de conception de la table finale.

Cette dernière sera justement une jointure de la table interne et de la table externe par une variable contenant deux informations : la marque et le modèle.

De la même manière que pour le Mapping de la table de données interne, il a été établi que sur ces trois marques majeures : Apple, Samsung, Huawei. Pour les autres modèles et marques, ils étaient marqués comme Others.

2.4.2.2 Traitement et création de variables

Certes GSMArena contient 86 caractéristiques pour 8 000 appareils électroniques majoritairement composé de téléphones portables, mais ces caractéristiques doivent être traitées afin d'optimiser notre étude.

En effet, l'information qui nous intéresse est souvent en format texte libre dans laquelle il faut faire plusieurs opérations :

- Batterie
- Mémoire interne
- Taille d'écran en pouces
- Résolution de l'écran
- Poids du téléphone
- Protection de l'écran

Parmi ces opérations, nous attirons l'attention sur le champs « body_dimensions », lequel a été transformé en 4 variables :

- Longueur du téléphone
- Largeur du téléphone
- Epaisseur du téléphone
- Le ratio taille d'écran sur taille du téléphone = taille de l'écran / longueur du téléphone

2.5 Jeu de données final

Les manipulations des tables de données ont été réalisées sous Python dans la plateforme du nom de Domino. Les données de primes sont jointes aux sinistres par numéro de police. La table de données externe est, quant à elle, jointe à la table d'information du portefeuille dit interne complète avec une clé de jointure (annoncé plus haut Marque – Model1) :

[Marque] X [Modèle]

Notre table finale fusionnant à la fois des données internes et externes contient plus de 50 champs. Nous avons réussi à récupérer 100% des modèles de notre portefeuille (restreint au périmètre de l'étude) dans la table GSMArena (aucune perte d'information).

La suite du mémoire amène à étudier l'ensemble de ces données disponibles en testant plusieurs modèles du suivi de la fréquence de sinistralité avec et sans l'apport de données externes. Cette étude est réalisée sur la garantie principale du produit, c'est-à-dire le dommage accidentel : la garantie vol n'est pas étudiée dans ce mémoire.

Une fois cette analyse réalisée, nous pourrons par la suite, lors de futures études développer nos idées par la création d'un module *Analytics* T&D transversale et centrer notre étude sur les facteurs ayant un impact majeur.

Ci-dessous un schéma récapitulatif du travail effectué sur les tables de données :

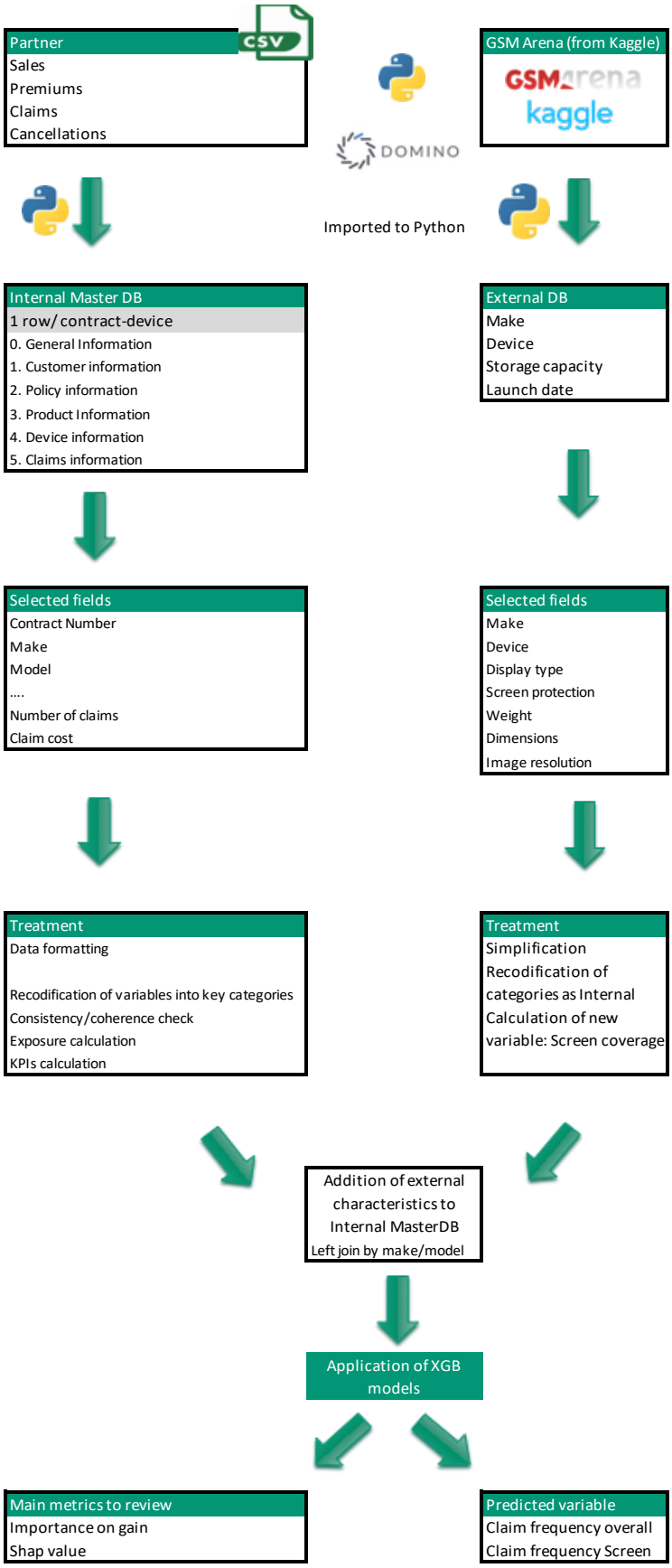


FIGURE 20 : CONSTRUCTION DE TABLE DE DONNEES FINALE

Analyse des données et sélection des variables

Dans cette section, une analyse des données disponibles est réalisée. La méthode d'apprentissage automatique de l'Extreme Gradient Boosting permet de réaliser un tri sur l'ensemble des variables et sert également en tant que modèle de régression pour prédire la variable réponse (la fréquence de sinistralité). Ainsi, dans un premier temps la théorie du Gradient Boosting est développée, cette technique nous permettra par la suite de tester l'apport des données externes puis servira à analyser et présélectionner les variables les plus influentes du modèle.

3.1 Le Machine Learning

3.1.1 Qu'est-ce que le Machine Learning ?

Arthur Samuel définissait en 1959 le *Machine Learning* comme étude ayant pour but de donner la capacité à une machine d'apprendre sans être explicitement programmée. Il aura fallu quarante ans afin que Tom Mitchell ajoute à cette première définition une suite explicité par : « On dit qu'un programme informatique apprend de l'expérience E en ce qui concerne une certaine classe de tâches T et la mesure de performance P, si sa performance dans les tâches T, telle que mesurée par P, s'améliore avec l'expérience E ».

Le processus d'apprentissage commence par des observations ou des données, telles que des exemples, une expérience directe ou des instructions, afin de rechercher des modèles dans les données et de prendre de meilleures décisions à l'avenir sur la base des exemples que nous fournissons. L'objectif premier est de permettre aux ordinateurs d'apprendre automatiquement sans intervention ou assistance humaine et d'ajuster les actions en conséquence.

- **Les données :**

Le *Machine Learning* consiste à apprendre, sans être explicitement programmé, à partir de données. Parmi les données il existe généralement une variable à expliquer ainsi que des variables explicatives. Les modèles de *Machine Learning* apprennent la variable à expliquer en fonction des variables explicatives.

Une fois la phase d'apprentissage effectuée, les modèles sont capables de prédire la variable à expliquer sur une nouvelle table de données contenant les variables explicatives seules.

Il existe une possibilité d'explication de données très variées. Il peut s'agir, par exemple, de déterminer la possibilité qu'un assuré commette une fraude, quel est le risque qu'un assuré ait un accident de la route ou encore la possibilité de survie d'un naufragé.

Nous distinguons deux principaux types de données :

- Les données quantitatives.
- Les données qualitatives.

Les données **quantitatives** sont les variables qui ont une certaine représentation numérique et qui contiennent des informations numériques. La taille, l'âge, le poids sont les types qui entrent dans cette catégorie.

Les variables quantitatives sont classées comme suit :

- Numériques.
- Continues.

Les variables numériques sont les variables qui sont de nature discrète.
Disons que nous avons une variable comme le nombre de chambres dans une maison, les valeurs possibles peuvent être 0,1,2,3 et ainsi de suite. Comme vous pouvez le voir, les valeurs associées à cette variable sont discrètes par nature, car le nombre de chambres ne peut pas être égal à 1,5.

Les variables continues sont celles dont les valeurs sont de nature continue comme la hauteur puisque les valeurs sont de nature continue. La valeur de la hauteur peut être de 175,5 cm.

Les données **qualitatives** sont les variables qui sont de nature catégorielle ou qui n'ont pas de représentation numérique. Par exemple, les variables comme le sexe, la couleur des yeux sont des variables qualitatives par nature catégorielles car la valeur possible d'une variable de sexe est masculine, féminine et autres et elles n'ont pas de représentation numérique ou ne peuvent pas être représentées numériquement.
Les variables qualitatives sont classées comme suit :

- Ordinales
- Nominales

Les variables ordinales sont celles qui peuvent être rangées dans un ordre. Par exemple, considérons une variable comme la taille d'un T-shirt. Les valeurs possibles sont XXL, XL, L, M et S. Nous pouvons donc classer les données dans l'ordre car XXL est

supérieure à XL, etc. Ce type de variable qualitative fait donc partie des variables ordinales.

Les variables nominales sont les variables où l'ordre n'a aucune importance. Le sexe est une variable nominale parce qu'il n'y a pas d'ordre chez les hommes, les femmes ou autres.

La qualité des modèles de *Machine Learning* dépend fortement de la qualité des données utilisées lors de l'apprentissage. Des études préalables quant à la qualité des données sont ainsi faites, il faut notamment fixer les erreurs, mais également veiller au fait d'avoir une table de données suffisamment représentative de la réalité que l'on souhaite modéliser. La proposition des variables explicatives pertinentes au modèle et l'abstention des informations redondantes ou inutiles sont à prendre en compte. Le traitement de données est considéré comme le premier pilier du *Machine Learning*.

3.1.2 L'apprentissage supervisé ?

On distingue deux types d'algorithmes : les algorithmes supervisés et non supervisés.

Les algorithmes supervisés extraient de la connaissance à partir de couples variables explicatives ou variables à expliquer, ou plus simplement entrées/sorties. Les sorties sont donc déjà connues. Le fonctionnement de ces modèles suppose un lien entre les valeurs d'entrée et de sortie. L'algorithme supervisé vise à apprendre ce lien via différents exemples, pour ensuite être capable de prédire une valeur de sortie à partir des données d'entrées seules.

Il est appelé apprentissage supervisé car le processus d'apprentissage d'un algorithme à partir de l'ensemble des données de formation peut être considéré comme un enseignant supervisant le processus d'apprentissage. Nous connaissons les bonnes réponses, l'algorithme fait itérativement des prédictions sur les données de formation et est corrigé par l'enseignant. L'apprentissage s'arrête lorsque l'algorithme atteint un niveau de performance acceptable.

Les algorithmes non supervisés n'intègrent pas cette notion d'entrée/sortie. L'objectif de l'apprentissage non supervisé est de modéliser la structure ou la distribution sous-jacente des données afin d'en apprendre davantage sur les données.

On les appelle apprentissage non supervisé car, contrairement à l'apprentissage supervisé ci-dessus, il n'y a pas de réponses correctes et il n'y a pas d'enseignant. Les

algorithmes sont laissés à eux-mêmes pour découvrir et présenter la structure intéressante dans les données.

Dans ce mémoire, nous allons nous intéresser exclusivement à des méthodes d'apprentissage supervisé.

Le principe des algorithmes d'apprentissage supervisé est de supposer qu'il existe une fonction lien f entre les données de sorties Y et les données d'entrées $X = (Xi)_0 \leq i \leq n$ avec $\forall\ i, X_i$ un vecteur de même dimension que Y. Nous avons alors :
$Y = f(X) + \varepsilon$ avec ε un bruit aléatoire centré.

Nous avons notamment la relation :

$$E[Y\ |X1, \ldots Xn] = f(X1, \ldots Xn).$$

Les algorithmes d'apprentissage supervisé cherchent à modéliser la fonction lien f. Une des difficultés de l'apprentissage supervisé est d'apprendre la fonction lien, sans apprendre le bruit aléatoire, c'est ce que nous allons présenter avec le dilemme biais-variance.

3.1.3 Le lien Biais – Variance ?

Les modèles d'estimation commettent deux types d'erreurs lorsqu'ils cherchent à prédire une variable de sortie :

- Le biais dit erreur du modèle : représente la différence entre le paramètre réel et son espérance
- La variance dite erreur de processus : lié au fait qu'il s'agit d'un processus stochastique/aléatoire

Le biais est la différence entre la prédiction moyenne de notre modèle et la valeur correcte que nous essayons de prédire.

Le modèle avec un biais élevé accorde très peu d'attention aux données de formation et simplifie trop le modèle. Il conduit toujours à une erreur élevée sur les données de formation et de test.

La variance est la variabilité de la prédiction du modèle pour un point de données donné ou une valeur qui nous indique la dispersion de nos données.

Un modèle à forte variance accorde beaucoup d'attention aux données de formation et ne généralise pas sur les données qu'il n'a pas encore vues. Par conséquent,

de tels modèles sont très performants sur les données de formation mais présentent des taux d'erreur élevés sur les données de test.

À titre d'illustration nous proposons le schéma issu de l'excellent article « Understanding the Bias Variance Tradeoff » [13] :

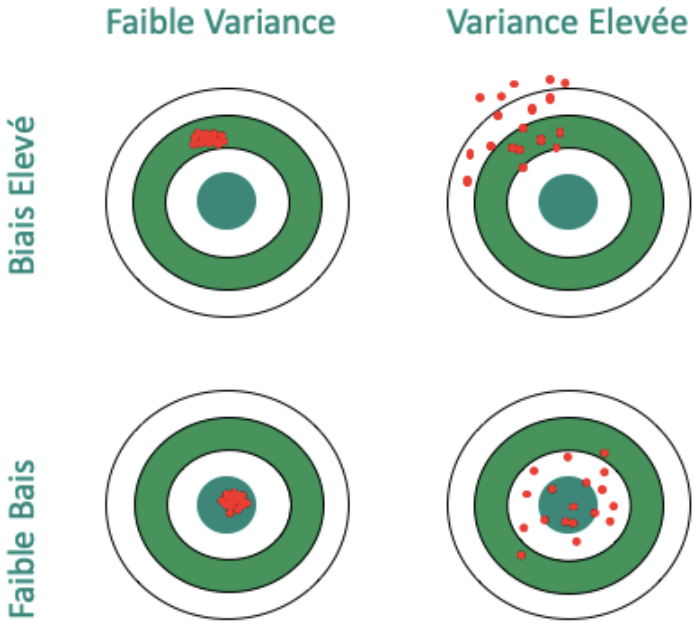


FIGURE 21 : REPRESENTATION DU BIAIS ET DE LA VARIANCE D'UN MODELE

Dans le diagramme ci-dessus, le centre de la cible est un modèle qui prédit parfaitement les valeurs correctes. Plus nous nous éloignons de la cible, plus nos prédictions deviennent mauvaises. Nous pouvons répéter notre processus de construction de modèle pour obtenir des résultats distincts sur la cible.

Cette problématique est centrale dans les modèles d'apprentissage supervisé. Généralement, diminuer la variance se fait au prix d'une augmentation du biais et inversement. Il s'agit alors de trouver un compromis acceptable entre les deux : on parle de « compromis biais-variance ». Supposons qu'il existe une fonction f liant la variable de sortie Y aux données d'entrées X de la manière suivante :

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

Avec : ε un vecteur aléatoire centré de variance $Id \times \sigma^2$.

Notons $\hat{f}$ la fonction de prédiction obtenue par un modèle de *Machine Learning*. Si nous donnons une matrice d'entrée X au modèle, il nous prédit ainsi la valeur $\hat{f}(X)$

en sortie. L'erreur commise par le modèle est l'écart entre cette valeur et $f(\mathrm{X})$. Mathématiquement, nous obtenons la relation suivante :

$$\begin{aligned} E\left[\left(f(X) - \hat{f}(X)\right)^2\right] &= E\left[f(X)^2 - 2\times f(X)\times\hat{f}(X) + \hat{f}(X)^2\right] \\ &= E\left[f(X)^2\right] + E\left[\hat{f}(X)^2\right] - 2\times f(X)\times E[\hat{f}(X)]\ \Big] \\ &= Var\big(f(X)\big) + E[\hat{f}(X)]^2 + Var(\hat{f}(X)) + E[\hat{f}(X)]^2 - 2\times f(X)\times E[\hat{f}(X)] \\ &= \sigma^2 + Var(\hat{f}(X)) + Biais(\hat{f}(X))^2 \end{aligned}$$

Avec :

- $Biais\left(\hat{f}(X)\right) = E\left(f(X) - \hat{f}(X)\right)$
- $Var\left(\hat{f}(X)\right) = E\left[(\hat{f}(X) - E[\hat{f}(X)])^2\right] = E[\hat{f}(X)^2] - E[\hat{f}(X)]^2$

Par conséquent, l'erreur quadratique moyenne générée par le modèle dépend bien d'une variance et d'un biais décrits précédemment. A cela s'ajoute σ^2 qui est « l'erreur incompressible » que nous allons ignorer.

Nous allons présenter plus en détail le phénomène de « sur-apprentissage ». Cette problématique apparaît souvent lorsque l'on cherche à diminuer le biais des modèles. Nous présenterons également les solutions pour faire face au compromis biais-variance.

3.1.4 Le Sur-apprentissage ?

Nous parlons de sur-apprentissage ou d'« *Overfitting* » pour un modèle supervisé lorsque le modèle obtient d'excellents résultats de prédiction sur les données d'apprentissage fournies, mais qu'il s'avère être mauvais lors de la prédiction de nouvelles données.

Dans l'apprentissage supervisé, le sur-apprentissage se produit lorsque notre modèle capture le bruit en même temps que le modèle sous-jacent dans les données. Cela se produit lorsque nous entraînons beaucoup notre modèle sur un ensemble de données bruyantes. Ces modèles ont un faible biais et une forte variance. Ces modèles sont très complexes, comme les arbres de décision qui sont sujets au surdimensionnement.

Dans l'apprentissage supervisé, il y a sous-apprentissage lorsqu'un modèle est incapable de saisir le modèle sous-jacent des données. Ces modèles présentent généralement un biais élevé et une faible variance. Cela se produit lorsque nous disposons de très peu de données pour construire un modèle précis ou lorsque nous essayons de construire un modèle linéaire avec des données non linéaires. De plus, ces types de modèles

sont très simples pour saisir les modèles complexes des données comme la régression linéaire et logistique.

Si notre modèle est trop simple et comporte très peu de paramètres, il peut présenter un biais élevé et une faible variance. En revanche, si notre modèle comporte un grand nombre de paramètres, il aura une variance élevée et un faible biais. Nous devons donc trouver le bon équilibre sans sur-ajuster ou sous-ajuster les données.

C'est ce compromis de complexité qui explique le compromis entre biais et variance. Un algorithme ne peut pas être plus complexe et moins complexe en même temps.

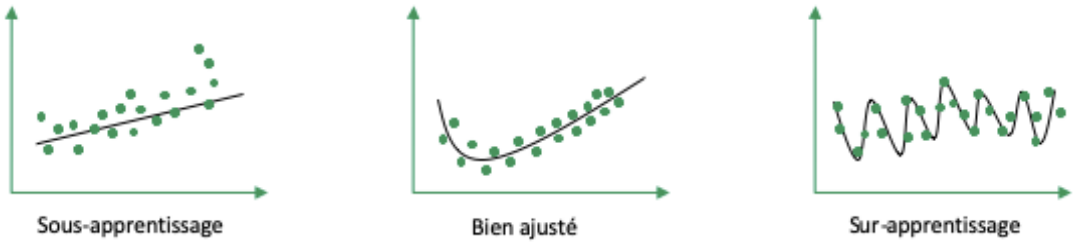


FIGURE 22 : LE SUR-APPRENTISSAGE

Il s'agit alors de trouver un bon compromis entre la performance obtenue lors de l'apprentissage et la possibilité de généraliser le modèle. Pour construire un bon modèle, nous devons trouver un bon équilibre entre le biais et la variance de manière à minimiser l'erreur totale.

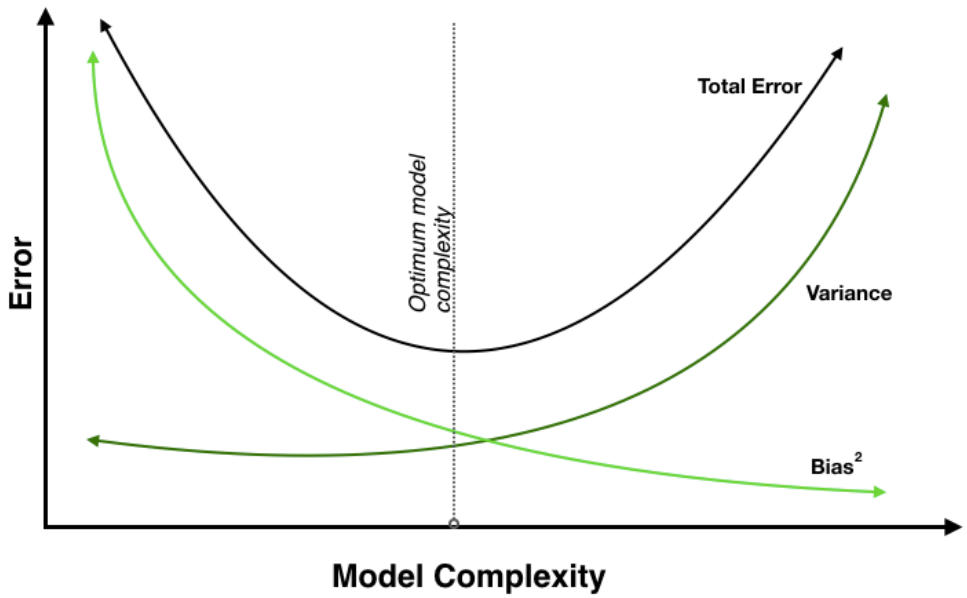


FIGURE 23 : LE COMPROMIS BIAIS VARIANCE

Pour trouver le point de complexité optimal, il faut tester le modèle sur d'autres données que celles utilisées lors de l'apprentissage afin d'en déterminer sa variance. Le biais peut en effet être mesuré sur les données d'apprentissage et peut généralement être rendu très faible en augmentant la complexité du modèle.

L'étude porte sur le portefeuille d'un broker Américain pour les contrats d'assurance téléphonique sur l'année 2014 à 2019. La table de données est ensuite découpée aléatoirement en deux échantillons :

Un échantillon d'apprentissage contenant 70 % de la table et sur lequel les méthodes sont appliquées et les algorithmes apprennent. Il permet d'ajuster le modèle aux données.

Un échantillon test qui permet de valider le modèle sur une table de données totalement indépendante de la table sur laquelle le modèle a appris et ainsi simuler la réception de nouvelles données. Cet échantillon permet d'évaluer objectivement l'erreur réelle de prédiction du modèle en comparant valeurs prédites et valeurs observées.

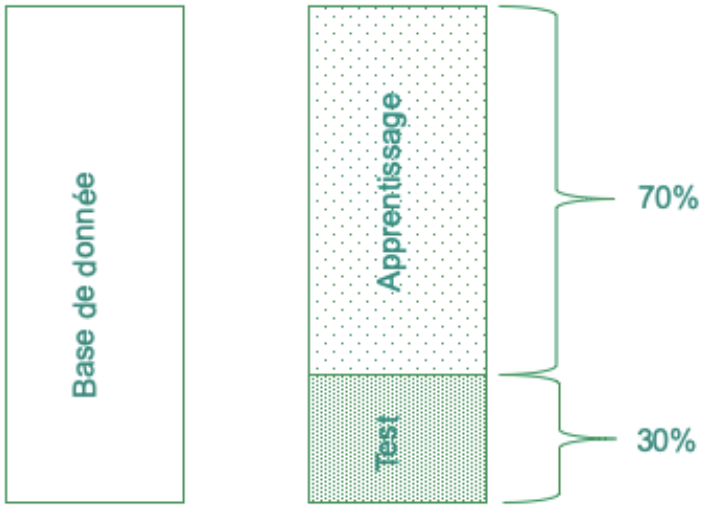


FIGURE 24 : SEPARATION TRAIN-TEST DE LA TABLE DE DONNEES

3.1.5 Utilisation de la validation croisée dans les méthodes d'apprentissage supervisé

La validation croisée reprend cette idée pour le choix du jeu de paramètres. Cependant, celle-ci est plus exhaustive. Au lieu de considérer une table d'entraînement et de validation fixée, la validation croisée considère différentes tables d'entraînement et de validation. Cette approche vise à optimiser la généralisation du modèle.

Il existe différents algorithmes de validation croisée, nous allons maintenant en présenter quelques-uns. Nous supposons que la table d'apprentissage contient m couples entrée/sortie. Voici quelques méthodes de validation croisée.

• La méthode LPOCV (Leave P Out Cross Validation) :

La table de validation contiendra k couples entrée/sortie et la table d'entraînement correspondante sera la table d'apprentissage à laquelle nous aurons retiré les k couples de la table de validation. Le processus sera répété jusqu'à avoir obtenu toutes les tables d'entraînements et de validations possibles en retirant k éléments.

Cette approche laisse p points de données en dehors des données d'entrainement, c'est-à-dire que s'il y a n points de données dans l'échantillon original, alors, n-p échantillons sont utilisés pour former le modèle et p points sont utilisés comme ensemble de validation. Cette démarche est répétée pour toutes les combinaisons dans lesquelles l'échantillon original peut être séparé de cette manière, puis l'erreur est calculée en moyenne pour tous les essais, pour donner l'efficacité globale.

Cette méthode est exhaustive dans le sens où elle doit entraîner et valider le modèle pour toutes les combinaisons possibles, et pour un p modérément grand, elle peut devenir irréalisable par calcul.

Un cas particulier de cette méthode est lorsque p = 1. Cette méthode est connue sous le nom de validation croisée "Leave One Out".

• La méthode LOOCV (Leave One Out Cross Validation) :

Cette méthode consiste à construire m bases d'entraînement et de validation. La base de validation ne contiendra qu'un unique couple entrée/sortie. Ce couple va successivement être chacun des éléments de la base d'apprentissage. La base d'entraînement sera constituée de la base d'apprentissage à laquelle nous aurons retiré l'élément de la base de validation. L'erreur du modèle est calculée sur chaque base de validation. Ces m erreurs peuvent être utilisées pour juger de la performance du modèle en validation croisée.

Voici un schéma représentant la méthode LOOCV, qui est un cas particulier de LKOCV pour k=1 :

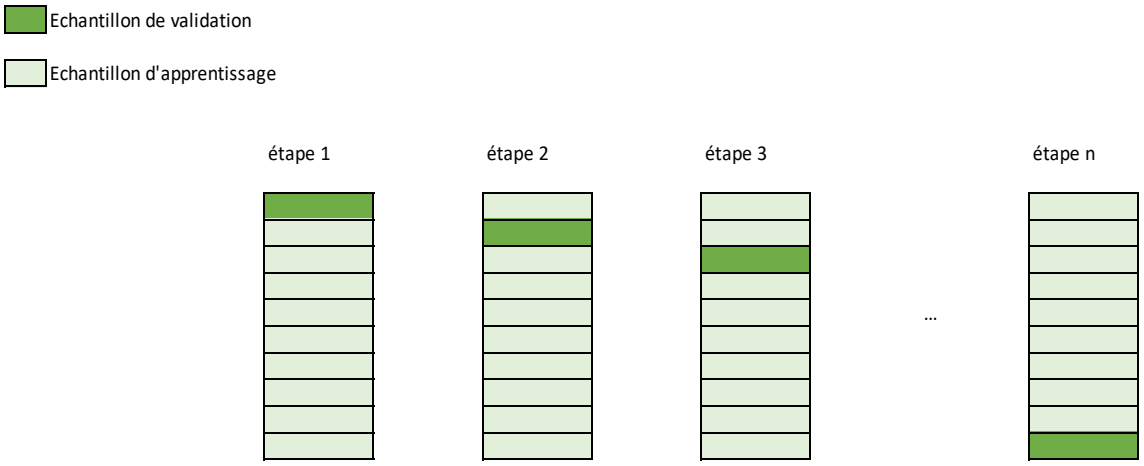


FIGURE 25 : METHODE LOOCV

• La méthode k-fold Cross Validation :

Dans la validation croisée k-fold, les données sont divisées en k sous-ensembles de tailles égales de façon aléatoire. La méthode d'attente est alors répétée k fois, de sorte qu'à chaque fois, l'un des k sous-ensembles est utilisé comme ensemble de test / validation et les autres k-1 sous-ensembles sont réunis pour former un ensemble d'entraînement.

L'estimation de l'erreur est moyennée sur l'ensemble des k essais pour obtenir l'efficacité totale de notre modèle. Comme nous pouvons le voir, chaque point de données doit se trouver dans un ensemble de validation exactement une fois, et doit se trouver dans un ensemble d'entraînement k-1 fois. Cela réduit significativement le biais car nous utilisons la plupart des données pour l'ajustement, et réduit tout aussi significativement la variance car la plupart des données sont également utilisées dans un ensemble de validation. L'échange des ensembles d'entraînement et de test renforce également l'efficacité de cette méthode. En règle générale et selon les données empiriques, k = 5 ou 10 est généralement préféré, mais rien n'est fixe et il peut prendre n'importe quelle valeur.

La méthode LPOCV (notons que LOOCV est un cas particulier de LPOCV) est exhaustive, c'est-à-dire qu'une fois terminée, la méthode a divisé le jeu de données en fonction de toutes les combinaisons possibles. Alors que la méthode k-fold n'est pas exhaustive, car elle ne sépare le jeu de données qu'en un nombre limité de sous-échantillons possibles. La méthode k-fold présente néanmoins l'intérêt d'être plus rapide à mettre en place qu'une méthode exhaustive.

3.2 Méthode d'apprentissage supervisé retenu

Tout d'abord, nous allons présenter un modèle très simple : les arbres de décision. En effet, il est utile de comprendre leur fonctionnement afin de comprendre par la suite le modèle implémenté qui est lui-même tiré des arbres de décision.

3.2.1 Les arbres décisionnels

Les arbres de décision sont parmi les méthodes de classification les plus populaires et les plus simples à ce jour. Leur popularité est principalement due à leur mise en œuvre très simple et efficace, à leur capacité à saisir des relations non linéaires entre les différents intrants et enfin à leur assez bon niveau d'interprétabilité.

Cette méthode est instable malgré ses divers avantages, car toute modification mineure de l'ensemble des données relatives à la formation entraînera probablement des changements dans l'arborescence.

Les arbres de décision peuvent expliquer les variables quantitatives et qualitatives. Pour les variables quantitatives, nous parlons d'arbres de régression et pour les variables qualitatives : d'arbres de classification.

L'arbre de décision permet de diviser les variables explicatives en groupes homogènes en fonction des variables à prédire. Ces groupes sont formés en considérant une hiérarchie basée sur le pouvoir prédictif des variables explicatives.

L'algorithme d'un arbre de décision fonctionne selon le **principe** suivant :

- On divise les individus en k classes (souvent k=2) pour expliquer la variable de sortie. La première division est obtenue en choisissant la variable explicative qui fournit la meilleure explication de la variable de sortie. L'échantillon initial est alors divisé en k sous-échantillons de populations. Ces sous-populations définissent des nœuds de l'arbre. À chaque nœud est associé une mesure de proportion, qui permet d'expliquer l'appartenance à une classe ou la signification d'une variable de sortie.

- On recommence : chaque sous-population est à nouveau divisée selon la variable la plus pertinente.

- On continue jusqu'à ce que plus aucune séparation ne soit possible. Nous obtenons alors des nœuds terminaux, appelés « feuilles » de l'arbre.

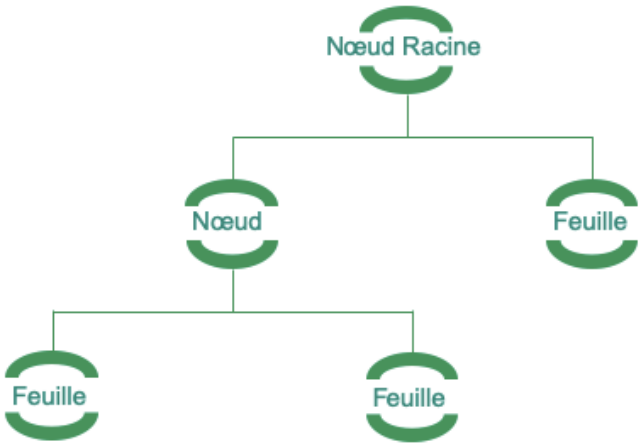


FIGURE 26 : SCHEMA SIMPLIFIE D'UN ARBRE BINAIRE CART

L'arbre CART (Classification And Regression Tree) est une méthode d'apprentissage statistique supervisé visant à construire un modèle de prédiction. L'idée principale de l'arbre CART est de partitionner récursivement et de manière binaire l'espace des variables explicatives afin d'obtenir toutes les valeurs possibles de la variable à prédire.

Supposons que nous disposons d'un n-échantillon d'une variable Y avec p prédicteurs $x_1, \ldots, x_p$. L'objectif est de pouvoir prédire les valeurs de Y pour de nouvelles valeurs $x_1, \ldots, x_p$.

Soit t_p, le nœud parent et t_g , t_d les nœuds fils respectivement gauche et droit. À partir de l'ensemble des observations sur la variable Y, l'arbre CART sélectionne à chaque itération et donc à chaque nœud t_p, la variable xj et sa valeur x ∗ j qui segmentent le mieux l'espace des variables en deux sous-espaces les plus homogènes possible dans les nœuds t_g et t_d .

À titre d'illustration nous proposons ce schéma issu de la thèse « Classification and Regression Trees » [17] :

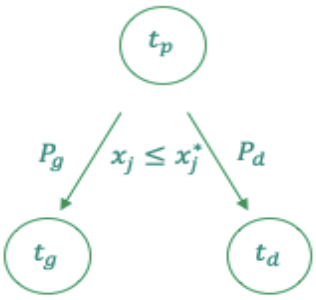


FIGURE 27 : SCHEMA DE FRACTIONNEMENT DE L'ALGORITHME CART

Cette notion d'homogénéité se traduit par une fonction d'impureté notée i(t). Il existe plusieurs fonctions qui permettent de définir l'impureté i(t) d'un nœud t, mais dans le cas de l'étude d'une variable aléatoire Y continue, comme c'est le cas ici, la variance du nœud peut représenter un bon indicateur de l'hétérogénéité présente dans le nœud.

L'impureté d'un nœud parent restant constante indépendamment du découpage réalisé pour créer les nœuds fils, maximiser l'homogénéité dans les nœuds fils est alors équivalent à maximiser la variation d'impureté Δi défini comme :

$$\Delta \mathrm{i} \ = \ \mathrm{i}(\mathrm{t_p}) \ - \mathrm{E}(\mathrm{i}(\mathrm{t_{fils}}))$$

Le problème de maximisation de l'impureté Δi définie dans l'équation précédente peut également s'écrire :

$$\max_{x_j < x_j^*, j=1,\ldots,p} [i(t_p) - P_g \times i(t_g) - P_d \times i(t_d)]$$

Où P_g et P_d sont respectivement les probabilités des nœuds gauche et droit.

Ce problème traduit la manière de procéder de l'algorithme : il parcourt toutes les valeurs possibles pour toutes les variables x_j pour j = 1, ..., p afin de trouver le meilleur découpage qui permettra de maximiser la variation d'impureté Δi.

Dans le cas où la fonction d'impureté est représentée par la variance du nœud, le problème de maximisation énoncé au-dessus pour alors se réécrire :

$$\max_{x_j < x_j^*, j=1,\ldots,p} [P_g \times \ Var(t_g) + P_d \times \ Var(t_d)]$$

3.2.2 Gradient Boosting

Les méthodes de prédiction Boosting comme le Random Forest (autrement dit forêts aléatoires), sont basées sur l'agrégation d'arbres de décisions. La différence avec le Random Forest consiste dans le fait que la génération des arbres se fait en série.

En effet, le Boosting génère des arbres reliés, c'est-à-dire que chaque arbre a accès à son prédécesseur. La création des arbres a donc pour objectif d'améliorer la prédiction de l'arbre précédent en affectant plus de poids aux observations mal classées de son prédécesseur. Ces méthodes s'occupent mieux des observations difficiles à classer et évitent en même temps le problème de sur-apprentissage en agrégeant les arbres.

Différents algorithmes de Boosting existent. Ils sont utilisés pour agréger des modèles ou agréger les arbres. Ils diffèrent selon la méthode d'agrégation d'arbres, la méthode de traitement des observations mal classées ou leur fonction de perte. Nous utilisons dans notre étude la méthode de XGBoost (Extreme Gradient Boosting) et LightGBM qui sont deux variantes de l'algorithme de Gradient Boosting.

<u>Nous présentons ci-dessous la méthode de Gradient Boosting :</u>

Le modèle Gradient Boosting est appelé ainsi puisqu'un algorithme de descente du gradient est utilisé pour minimiser la fonction de perte lors de la génération des arbres. Nous considérons la base d'apprentissage composée des couples entrées/sorties (X,Y).

Soit $\mathfrak{I}_k$ un arbre construit à l'étape b. Soit f_k le modèle à l'étape b :

$$f_k = \sum_{i=1}^{k} \mathfrak{I}_i$$

Soit E la fonction de perte du modèle. Nous considérons comme fonction de perte la fonction des moindres carrées E défini comme suit :

$$E = \sum_{m=1}^{n} \frac{(Y_m - f(X_m))^2}{2}$$

Le gradient de E par rapport à $f(X_m)$ est :

$$\frac{\partial E}{\partial f(X_m)} = \frac{1}{2} \sum_{i=1}^{n} \frac{\partial (Y_i - f(X_i))^2}{\partial f(X_m)} = f(X_m) - Y_m$$

L'algorithme du Gradient Boosting se déroule de la manière suivante :

- Nous construisons un premier arbre de décision $\mathfrak{I}_1$ visant à prédire Y en fonction de X. Le résidu de la valeur observée Y_m est $res_1 = Y_m - f_1(X_m) = -\frac{\partial E}{\partial f_1(X_m)}$

- Pour k = 2, ..., k itérations, les étapes suivantes sont itérées. Afin de combler les manques de l'arbre $\mathfrak{I}_{k-1}$ l'algorithme construit l'arbre $\mathfrak{I}_k$ qui prédit le résidu de l'étape précédente. Le modèle construit à l'étape k est donc $f_k(X) = \mathfrak{I}_{k-1}(X) + \mathfrak{I}_k(X)$. L'arbre $\mathfrak{I}_k$ est égal à $\mathfrak{I}_k\ (X_m) = Y_m - f_{k-1}(X)) + (f_k(X) - Y_m)$. Le résidu est donc : $res_k = Y_m - f_k(X_m) = \frac{-\partial E}{\partial f_k(X_m)}$

3.2.3 Extreme Gradient Boosting

L'algorithme Extreme Gradient Boosting est une version dite « extrême » de l'algorithme Gradient Boosting. Il s'agit d'un algorithme particulièrement populaire à l'heure actuelle. Cet algorithme est connu par sa capacité prédictive et sa vitesse d'exécution. Tout en contrôlant la variance, comme les méthodes Boosting, XGBoost vise à réduire le biais.

Afin de mesurer la robustesse de l'algorithme, les algorithmes de prédiction utilisent une fonction appelée la fonction objective. Cette fonction est la somme de deux fonctions : la fonction de perte et la fonction de régularisation.

La fonction de perte mesure la capacité prédictive du modèle tandis que la fonction de régularisation contrôle la complexité du modèle et est utilisée afin d'éviter le problème de sur-apprentissage.

L'algorithme XGBoost construit des arbres de manière séquentielle. Si le modèle final contient k arbres, la prédiction finale de Y pour une observation x_m est :

$$\widehat{y_m} = \sum_{i=1}^{K} \mathfrak{I}_i(x_m)$$

L'algorithme initialise la valeur à prédire à 0.

$$\widehat{y_m}^0 = 0$$

De même que pour le Gradient Boosting la prédiction pour l'étape 1 est :

$$\widehat{y_m}^1 = \widehat{y_m}^0 + f_1(x_m)$$

Arrivant à l'itération k, nous avons :

$$\widehat{y_m}^k = \widehat{y_m}^{k-1} + \mathfrak{I}_k(x_m) = \sum_{i=1}^{K} \mathfrak{I}_i(x_m)$$

À chaque étape, nous déterminons $\mathfrak{I}_k(x_m)$. L'arbre à choisir pour chaque étape est celui qui minimise la fonction d'objectif. L'arbre choisi pour chaque étape dépend de forme de la fonction de régularisation et de la fonction de perte choisie.

La méthode construit des arbres horizontalement dite de manière Level-wise :

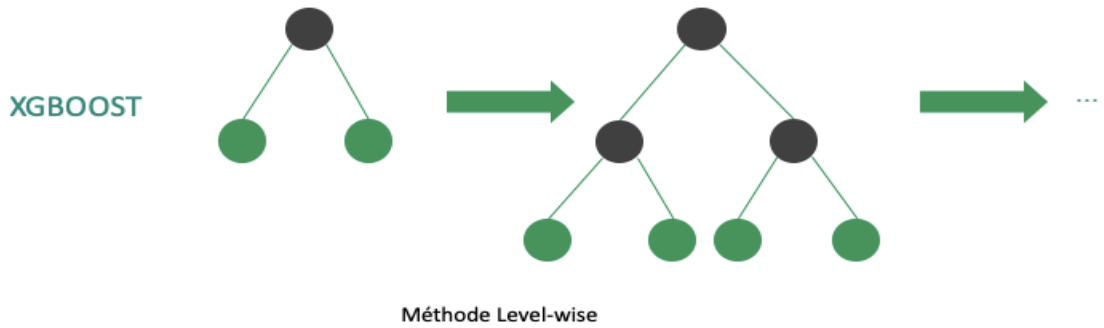


FIGURE 28 : SCHEMA DE LA METHODE LEVEL-WISE

3.2.4 LightGBM

La méthode LightGBM est elle aussi une méthode de Gradient Boosting basée sur des arbres de décisions. Cependant, elle se démarque en construisant des arbres verticalement (*Leaf-wise*) et non horizontalement (*Level-wise*). En fait, l'algorithme va choisir la feuille avec le plus grand delta de perte pour agrandir l'arbre.

Ainsi, en poursuivant la construction sur la même branche, cette méthode va réduire les pertes de façon plus significative que les méthodes évoluant horizontalement. Son avantage par rapport aux autres méthodes est qu'elle peut gérer des données volumineuses de façon beaucoup plus rapide.

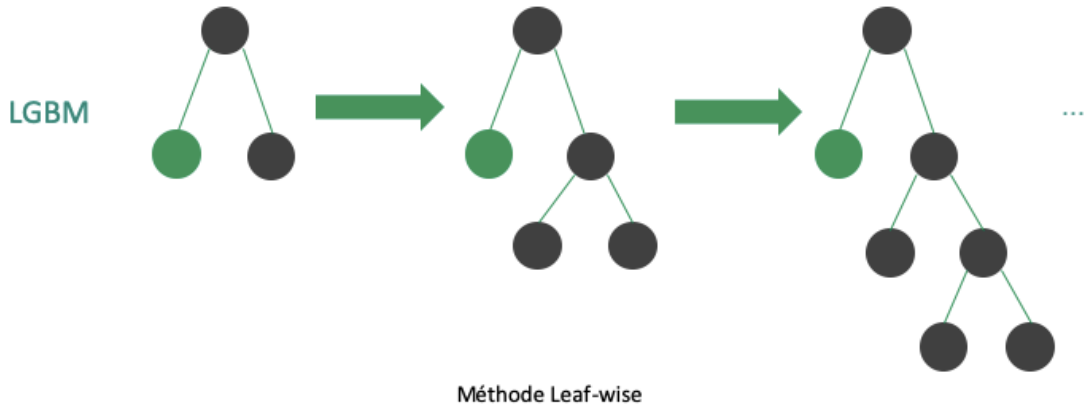


FIGURE 29 : SCHEMA DE LA METHODE LEAF-WISE

3.2.5 Avantages et inconvénients des modèles

Avantages de la méthode

XGBoost est un algorithme efficace et facile à utiliser qui offre des performances et une précision élevée par rapport à d'autres algorithmes. XGBoost est également connu

sous le nom de version régularisée du GBM. XGBoost propose une régularisation L1 (régression de Lasso) et L2 (régression de Ridge) intégrée qui empêche le modèle d'être surdimensionné. C'est pourquoi, XGBoost est également appelé forme régularisée de GBM (Gradient Boosting Machine). XGBoost utilise la puissance du traitement parallèle et c'est pourquoi il est beaucoup plus rapide que le GBM. Il utilise plusieurs cœurs de CPU pour exécuter le modèle.

Il a aussi une capacité intégrée de traitement des valeurs manquantes. Lorsque XGBoost rencontre une valeur manquante à un nœud, il essaie de diviser à la fois à gauche et à droite et apprend le chemin menant à une perte plus élevée pour chaque nœud. Il fait ensuite la même chose lorsqu'il travaille sur les données de test. XGBoost permet à l'utilisateur d'effectuer une validation croisée à chaque itération du processus de boosting et il est donc facile d'obtenir le nombre optimal exact d'itérations de boosting en une seule fois. C'est la différence avec GBM, où il faut effectuer une recherche par grille et où seules des valeurs limitées peuvent être testées. Finalement GBM arrêtera de diviser un nœud lorsqu'il rencontrera une perte négative dans la division. Il s'agit donc plutôt d'un algorithme gourmand. XGBoost, en revanche, effectue des scissions jusqu'à la profondeur maximale spécifiée, puis commence à élaguer l'arbre à l'envers et supprime les scissions au-delà desquelles il n'y a pas de gain positif.

Par exemple : il peut arriver qu'un fractionnement de perte négative, disons -4, soit suivi d'un fractionnement de perte positive +13. GBM s'arrêtera au moment où il rencontrera -4. Mais XGBoost ira plus loin et verra un effet combiné de +9 du fractionnement et conservera les deux.

Limites de la méthode

L'algorithme XGBoost dépend d'un très grand nombre de paramètres et est basé sur la méthode de Boosting présentant plusieurs limites. En effet, XGBoost ne peut pas gérer les caractéristiques catégorielles, nous devons binariser manuellement ces variables alors que l'algorithme LightGBM les traites automatiquement. LightGBM montre ensuite expérimentalement qu'il s'entraîne en une fraction de temps comme XGBoost avec une précision comparable. En outre, il existe des ensembles de données pour lesquels XGBoost ne peut pas fonctionner car les données ne peuvent pas être stockées dans la mémoire de travail, c'est un algorithme assez gourmand. LightGBM ne doit pas stocker autant de mémoire de travail. Les deux documents mentionnent les avantages de la compression des données et des accès au cache. LightGBM utilise XGBoost comme base de référence et le surpasse en termes de vitesse d'apprentissage et de taille des ensembles de données qu'il peut traiter. Les précisions sont comparables. Dans certains cas, le LightGBM atteint sa précision maximale en moins d'une minute et en ne lisant qu'une fraction de l'ensemble des données. Cela démontre la puissance des algorithmes d'approximation et

de l'échantillonnage intelligent d'un ensemble de données pour extraire le maximum d'informations, le plus rapidement possible.

3.2.6 Optimisation des paramètres de modèles

Les algorithmes de ML les plus puissants sont connus pour détecter des modèles et des régularités dans les données en réglant automatiquement des milliers (voire des millions) de paramètres pouvant être appris. Par exemple, dans les modèles basés sur les arbres (arbres de décision, XGBoost, LGBM), les paramètres pouvant être appris sont le choix des variables de décision à chaque nœud et les seuils numériques utilisés pour décider de prendre la branche gauche ou droite lors de la génération des prédictions.

Dans les modèles basés sur les arbres, les hyper-paramétrages comprennent des éléments comme la profondeur maximale de l'arbre, le nombre d'arbres à faire pousser, le nombre de variables à prendre en compte lors de la construction de chaque arbre, le nombre minimum d'échantillons sur une feuille, la fraction des observations utilisées pour construire un arbre, et quelques autres.

Étant donné le nombre de paramètres des algorithmes nous n'optimiserons que les paramètres suivants dont chaque nom correspond à celui utilisé par Python :

- Max_depth : profondeur maximale d'un arbre. Utilisée pour contrôler le sur-ajustement car une profondeur plus élevée permettra au modèle d'apprendre des relations très spécifiques à un échantillon particulier.

- Min_child_weight : définit la somme minimale des poids de toutes les observations requises chez un enfant.

- Eta : désigne la réduction de la taille d'étape utilisée lors de l'étape de mise à jour afin d'éviter le sur-ajustement.

- Nthread : désigne le nombre de cœurs utilisés pour l'exécution en parallèle.

- Subsample : désigne la fraction des observations qui doivent être des échantillons aléatoires pour chaque arbre. Des valeurs plus faibles rendent l'algorithme plus conservateur et empêchent le sur-ajustement, mais des valeurs trop faibles pourraient entraîner un sous-ajustement.

- Colsample_bytree : désigne la fraction des colonnes qui doivent être des échantillons aléatoires pour chaque arbre.

- Table_score : le score de prédiction initial de tous les cas, biais global.

- Objective : paramètre qui spécifie le type de fonction objectif utilisé. Par exemple objective = "binary : logistic" est spécifié pour la régression logistique pour la classification binaire.

- Metrics : désigne la métrique utilisée pour évaluer la validation du modèle. La métrique peut prendre les valeurs : "auc", "mse", "LogLoss", "rmse", etc.

- Num_boost_round : correspond au nombre de tours de Boosting ou d'arbres à construire. Sa valeur optimale dépend fortement des autres paramètres, et il doit donc être réajusté chaque fois que vous mettez à jour un paramètre. Vous pourriez le faire en l'ajustant avec tous les paramètres dans une recherche par grille, mais cela demande beaucoup de travail de calcul.

- Early_stopping_rounds : si les performances ne se sont pas améliorées pour n tours (n est défini par la variable early_stopping_round), nous arrêtons l'entraînement et gardons le meilleur nombre de tours de relance.

Ainsi, si nous voulons tester 5 paramètres en leur accordant 5 valeurs différentes afin de trouver les paramètres optimaux du model nous avons 5 x 5 x 5 x 5 x 5 = 3125 combinaisons possibles d'hyper-paramètres.
Ce sous-espace discret de tous les hyper-paramètres possibles est appelé la grille d'hyper-paramètres (**Grid Search**). Nous utiliserons le symbole de notation vectorielle $h = [h_0, h_1, \ldots, h_p]$ afin de désigner toute combinaison de ce type, c'est-à-dire tout point de la grille.

La recherche exhaustive de grille (GS) n'est rien d'autre que l'approche de force brute qui scanne toute la grille de combinaisons d'hyper-paramétrages h dans un certain ordre, calcule la perte de validation croisée pour chacune d'entre elles et trouve ainsi le **h*** optimal.

En général, l'utilisation de l'ordre lexicographique, c'est-à-dire l'ordre du dictionnaire imposé aux vecteurs hyper-paramétrages, est déconseillée et un ordre différent devrait être envisagé. La raison est qu'avec l'ordre lexicographique, il y a de fortes chances que la recherche se concentre sur une partie peu intéressante de l'espace de recherche pendant un temps assez long.

Une alternative intéressante consiste à scanner l'ensemble de la grille de manière totalement aléatoire, c'est-à-dire selon une permutation aléatoire de l'ensemble de la grille. Avec ce type de recherche, il est probable que l'on rencontre très tôt des régions proches de l'optimum dans l'espace hyper-paramétrique.

Cette démarche demande énormément en temps de compilation et est parfois quasi impossible à exécuter.

3.3 Mesure de l'apport des données extérieures par rapport aux données internes

L'apprentissage des modèles supervisés se fait sur la table d'apprentissage. Ensuite les performances sont testées sur la table de test. La table de test contient également des variables entrées/sorties, il s'agit alors de comparer la sortie prédite par le modèle à la variable sortie de la table de test. La mesure de la performance du modèle peut être faite de différentes manières.

3.3.1 Les mesures de performance

3.3.1.1 Variance expliquée de Poisson

L'étude des erreurs de prédiction permet d'évaluer la qualité des valeurs prédites. Il existe plusieurs indicateurs permettant de mesurer la qualité de la prévision d'une variable. Ils permettent également de déterminer la précision de la prévision par rapport à une méthode alternative. Dans un premier temps nous avons eu l'idée de comparer, par exemple, l'erreur quadratique moyenne obtenue pour chacune des méthodes de prévision.

L'erreur quadratique moyenne, notée MSE (Mean Square Error), permet de quantifier les écarts entre les valeurs prédites par un modèle et les valeurs réellement observées. Plus précisément la MSE est égale à la moyenne des erreurs individuelles élevées au carré calculée à partir des résidus issus de la différence entre les valeurs prédites par le modèle $\widehat{y_i}$ et les valeurs observées y_i,

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \widehat{y_i})^2 ,$$

D'autre part, le coefficient de détermination dit R^2 représente la proportion de la variance y_i qui a été expliquée par les variables indépendantes du modèle. Il fournit une indication de la qualité de l'ajustement et donc une mesure de la qualité des échantillons

non vus qui sont susceptibles d'être prédits par le modèle, grâce à la proportion de variance expliquée. R^2 est une version normalisée de MSE.

Comme cette variance dépend de l'ensemble des données, R^2 peut ne pas être comparable de manière significative entre les différents ensembles de données. Le meilleur score possible est de 1,0 et il peut être négatif (parce que le modèle peut être arbitrairement pire). Un modèle constant qui prédit toujours la valeur attendue de y_i, sans tenir compte des caractéristiques d'entrée, obtiendrait un score R^2 de 0,0.
Si $\widehat{y_i}$ est la valeur prédite du -ième échantillon et y_i est la valeur réelle correspondante pour les échantillons totaux, le R^2 estimé est défini comme suit :

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \widehat{y_i})^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2},$$

Finalement, le DataLab (équipe de *Data Science* consacrée à créer des outils afin de simplifier différentes études) de Cardif, m'a proposé une méthode alternative : en probabilité et en statistique, les distributions de Tweedie sont une famille de distributions de probabilité qui comprennent les distributions normales, Gamma et Gaussiennes inverses purement continues, la distribution de Poisson à échelle purement discrète et la classe des distributions de Poisson-Gamma composées qui ont une masse positive à zéro. Ainsi, nous calculons l'erreur de déviation moyenne Tweedie avec un paramètre de puissance (p). Il s'agit d'une mesure qui permet d'obtenir les valeurs attendues des objectifs de régression.

Les cas spéciaux suivants existent :

- Lorsque la puissance = 0, elle est équivalente à l'erreur quadratique moyenne.
- Lorsque la puissance = 1, elle équivaut à la moyenne de la déviation de la Poisson.
- Lorsque la puissance = 2, cela équivaut à l'écart moyen de Gamma.

Si $\widehat{y_i}$ est la valeur prédite du -ième échantillon, et y_i est la valeur réelle correspondante, alors l'erreur de déviation moyenne de Tweedie (D) pour la puissance p, estimée est définie comme :

$$Deviance(y, \hat{y}) = \frac{1}{n}\sum_{n=1}^{n}\begin{cases} (y_i - \widehat{y_i})^2, & pour\ p = 0\ (Normal) \\ 2(y_i \log\left(\frac{y_i}{\widehat{y_i}}\right) + \widehat{y_i} - y_i)\ , & pour\ p = 1\ (Poisson) \\ 2(y_i \log\left(\frac{\widehat{y_i}}{y_i}\right) + \frac{\widehat{y_i}}{y_i} - 1)\ , & pour\ p = 2\ (Gamma) \\ 2(\frac{\max\ (y_i, 0)^{2-p}}{(1-p)(2-p)} + \frac{y_i\widehat{y_i}^{1-p}}{1-p} - \frac{\widehat{y_i}^{2-p}}{2-p})\ , & sinon \end{cases}$$

Pour une distribution de Poisson avec puissance = 1, la déviance s'échelonne linéairement, et pour une distribution normale (puissance=0), quadratiquement. En général, plus la puissance est élevée, moins nous accordons de poids aux déviations extrêmes entre les cibles réelles et prédites.

La variance expliquée est la généralisation du R^2 appliqué à d'autres loi de probabilité que la loi normale. Ainsi pour notre étude, afin de mesurer les déviations extrêmes de prédiction de fréquence, nous avons une distribution de Poisson avec une puissance égale à 1.

On définit la variance de Poisson expliquée comme :

$$\text{Variance de Poisson expliquée} \;=\; 1 - \frac{\textit{Deviance poisson}}{\textit{Deviance de poisson en moyenne}},$$

Finalement afin de mesurer le gain de performance entre chaque modèle final, nous décidons de tout aussi calculer la somme des erreurs absolues (SAE). Plus précisément la SAE est égale à la somme des erreurs absolues individuelles calculé à partir des résidus issus de la différence entre les valeurs prédites par le modèle $\widehat{y_i}$ et les valeurs observées y_i,

$$SAE \;=\; \sum_{i=1}^{n} |y_i \;-\; \widehat{y_i}|$$

3.3.1.2 Courbe Lift

La *Lift Curve* permet de mesurer la performance en moyenne d'un modèle prédictif en l'opposant au valeurs réelles observées. En effet, un package Python contenant la modélisation de cette courbe par le DataLab de Cardif est disponible à l'utilisation. Cette dernière représente sur l'axe des abscisses nos prédictions coupées par bins (intervalle de valeurs), et sur l'axe des ordonnées une moyenne (de prédiction ou bien de valeurs réelles). Cette représentation nous permet de savoir si notre modèle est bien calibré en moyenne :

- Si la courbe de prédiction est au-dessus de l'actuel réel, notre modèle surestime

- Si la courbe de prédiction est en-dessous de l'actuel réel, notre modèle sous-estime

- Si la courbe de prédiction est superposée de l'actuel réel, notre modèle est bien prédit et bien calibré en moyenne

La double *Lift Curve* permet de mesurer la performance en moyenne de deux modèles prédictifs, comparé au modèle réel. Cette double courbe modélisée par le DataLab de Cardif, représente sur l'axe des abscisses la différence de prédictions pour

nos deux modèles choisis, coupé par bins (intervalle de valeurs), et sur l'axe des ordonnées une moyenne (de prédiction ou bien de valeurs réelles). Cette représentation, nous permet de savoir lequel de nos deux modèles est le mieux calibré en moyenne.

3.3.1.3 Valeur de Shapley

La méthode de Shapley repose sur la théorie des jeux coopératifs. Elle est utilisée pour allouer le gain total entre joueurs en fonction des valeurs de gains des coalitions possibles. Michel Denault (2001) a adapté cette méthode pour introduire la méthode d'allocation du capital de Shapley.

Nous présentons tout d'abord le principe de la valeur de Shapley dans son contexte d'origine.

<u>Principe de la méthode de Shapley :</u>

Dans un jeu coopératif, il est important de pouvoir trouver une répartition des gains pour une coopération. Il est souhaitable que cette répartition soit équitable. Une des solutions proposées dans la littérature de la théorie des jeux est la valeur de Shapley. La méthode de Shapley s'applique dans le cas d'une utilité transférable. Par exemple si plusieurs agriculteurs décident de former une coopérative dans l'objectif de maximiser la production et le profit, et ensuite répartir ce profit entre ses membres, nous pouvons parler de solution avec utilité transférable.

Un jeu coopératif est en général défini par sa fonction caractéristique, qui donne les valeurs des gains de chaque coalition. La description du jeu se fait à l'aide des valeurs de sa fonction caractéristique qu'on va noter φ, par exemple si la coalition composée des joueurs 1 et 5 obtient un gain de 20, la valeur de la fonction φ sera 20 pour cette coalition $\varphi(1, 5) = 20$. Le jeu est souvent appelé le jeu φ, et toutes les coalitions possibles ont une valeur, notamment les coalitions composées d'un seul joueur, la valeur de φ est par définition égale à 0 pour une coalition vide.

Dans un jeu coopératif de n joueurs, le nombre de coalitions non vides possibles est $2^n - 1$. La fonction caractéristique doit en principe vérifier la super-additivité pour les unions des coalitions disjointes, cela se traduit mathématiquement pour deux coalitions possibles C1 et C2 par :

$$(C1 \cap C2 = \emptyset) \Rightarrow \varphi(C1 \cup C2) \geq \varphi(C1) + \varphi(C2)$$

Dans le cas d'un jeu à utilité transférable, Shapley a développé un ensemble d'axiomes qui engendrent une solution unique, appelée la valeur de Shapley dans la

littérature. Soit $N = \{1, \ldots, n\}$, l'ensemble des joueurs, et $\boldsymbol{\phi_i(\varphi)}$ la valeur de Shapley dans un jeu coopératif de n joueurs et de fonction caractéristique φ. Les axiomes utilisés par Shapley sont les suivants :

1. Symétrie : Pour que le jeu soit équitable, la permutation des joueurs ne doit pas changer leurs valeurs de Shapley. C'est-à-dire, si le joueur i devient le joueur j, alors leurs valeurs de Shapley sont aussi permutées. Mathématiquement, pour toute permutation π, et pour tout joueur i ∈ N,

$$\phi_\pi(i)(\pi\varphi) = \phi_i(\varphi);$$

2. Optimalité de Pareto : La valeur des gains d'une coalition doit forcément être égale à la somme des valeurs de Shapley de ses membres : X i ∈ N

$$\phi_i(\varphi) = \varphi(N);$$

3. Additivité : Dans le cas d'un joueur qui participe parallèlement à deux jeux coopératifs, de fonctions caractéristiques φ1 et φ2, et si nous considérons ces deux jeux comme un seul jeu, donc de fonction caractéristique $\varphi = \varphi_1 + \varphi_2$, alors la valeur de Shapley du joueur i est tout simplement la somme de ses valeurs dans ces deux jeux :

$$\phi_i(\varphi) \equiv \phi_i(\varphi_1 + \varphi_2) = \phi_i(\varphi_1) + \phi_i(\varphi_2).$$

Shapley a montré que l'unique valeur du jeu φ qui vérifie ces axiomes, est la valeur de Shapley, donnée pour chaque joueur i par l'expression suivante :

$$\phi i(\varphi) = \sum_{\substack{Z \subset N \\ i \in Z}} \frac{(n - |Z|)!\,(|Z| - 1)!}{n!} \left[\varphi(Z) - \varphi(Z \backslash \{i\})\right]$$

Où |Z| est le nombre de joueurs de la coalition Z.

La solution de Shapley représente donc une valeur moyenne d'impacts marginaux du joueur i dans les différentes coalitions possibles. En effet, si nous considérons que chaque joueur entre aléatoirement dans une coalition, son impact sera mesuré par l'accroissement réalisé au niveau du gain de la coalition suite à son entrée.

3.3.2 Mesure de l'apport des données externes

Pour tester l'apport des données externes provenant de GSM Arena, par rapport aux autres données, les variables sont ajoutées au XGBoost et LGBM au fur et à mesure. Plusieurs modèles sont donc réalisés :

- Modèle 1 : Fréquence uniforme
- Modèle 2 : Fréquence par tranches de prix
- Modèle 3 : Prédiction XGBoost/LGBM par variables internes
- Modèle 4 : Prédiction XGBoost/LGBM par variable internes + externes

Modèle 1 :

Dans un premier temps, nous construisons un modèle dit « trivial », où la fréquence de sinistralité des téléphones du portefeuille correspond à la fréquence moyenne de tous les téléphones du portefeuille.

$$Fréquence_{flat} = \frac{\sum sinistres}{\sum exposition}$$

Ainsi, la variance de Poisson expliquée pour les deux algorithmes est égale à 0 car les valeurs prédites par le modèle $X_{modèle}$ et les valeurs observées X_{obs} sont égales (*dans ce cas* $X_{modèle} = X_{obs}$).

Modèle 2 :

Dans un second temps, nous avons considéré une situation réelle de tarification c'est-à-dire par tranches de prix. Nous avons simulé 5 tranches de prix adaptées au marché de consommations et calculé leurs fréquences de sinistralité.

$$Fréquence_{tranches} = \frac{\sum_{Tranche} sinistres}{\sum_{Tranche} exposition}$$

La première tranche de prix est définie entre 0 et 100€, la seconde entre 100 et 250€, la troisième entre 250 et 400€, la quatrième entre 400 et 800€ et la cinquième tranche pour les téléphones qui coutent plus de 800€.

La variance de Poisson expliquée pour ce modèle est égale à 0,7%. La déviance entre les fréquences calculées par tranches de prix et fréquences réelles est importante.

Modèle 3 :

Le 3ème modèle conçu nous permet de prédire la fréquence de sinistralité grâce aux variables internes de notre portefeuille :

- 2 variables qualitatives : marque et partenaire de distribution
- 1 variable quantitative : prix d'achat du téléphone

On prend l'exposition du contrat comme poids dans notre modèle de prédiction. La variance de Poisson expliquée nous affiche :

	Variance de Poisson expliquée	Temps d'exécution
XGBoost	6,18%	00 :02 :23
LightGBM	5,15%	00 :00 :00.0001

TABLEAU 8 : RESULTAT DE PREDICTION DU MODELE INTERNE PAR LES METHODES DE MACHINE LEARNING

Ainsi, notre première interprétation est que le pouvoir de prédiction de notre fréquence de sinistralité est plus efficace face à celui des tranches de prix.
Nous pouvons aussi remarquer que la variance de Poisson expliquée est relativement faible, mais que l'algorithme XGBoost propose une faible meilleure prédiction que le LightGBM pour un temps d'exécution bien plus lent. L'algorithme LightGBM prédit quasi-instantanément.

Dans un premier temps, nous décidons de représenter grâce à la *Lift Curve*, la représentation des fréquences prédites faces aux actuelles pour notre modèle composé de données uniquement internes à notre portefeuille modélisée par l'algorithme XGBoost. Ensuite nous reproduisons notre représentation modélisée par l'algorithme LightGBM.

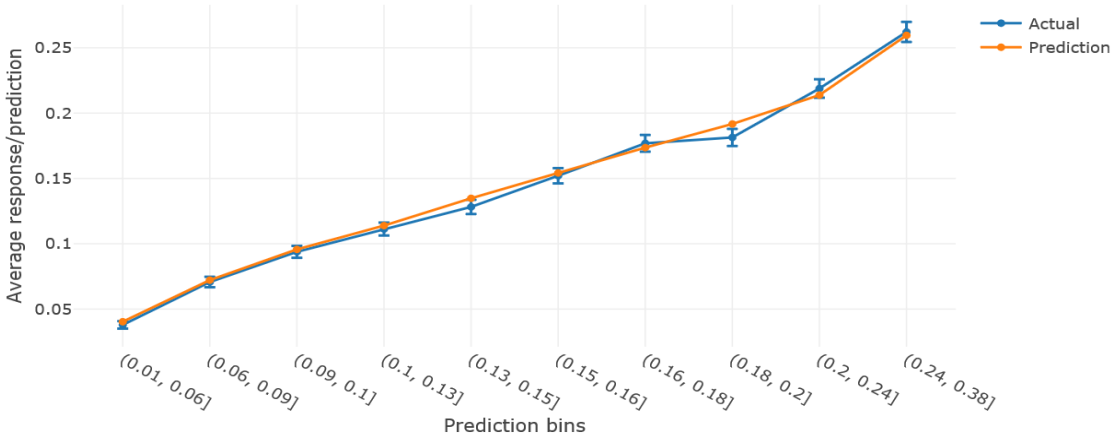


FIGURE 30 : LIFT CURVE DES PREDICTIONS[1] MODELISEES PAR XGB POUR LE MODELE INTERNE

La *Lift Curve* nous montre sur l'axe des abscisses une segmentation en 10 subdivisions de fréquences (ici [0,01 ;0,06], [0,06 ;0,09] ...), présentant en comparaison sur

[1] Indicateur de fréquence à titre visuel

l'axe des ordonnées les moyennes des fréquences prédites et réelles accompagnées d'un intervalle de confiance à 97,5%.

Les fréquences prédites sont généralement superposées aux réelles et appartiennent globalement aux intervalles de confiance (8 cas sur 10). Ainsi, le modèle en moyenne prédit bien et est en moyenne est bien calibré.

Dans un second temps, nous représentons grâce à la *Lift Curve*, les fréquences prédites faces aux actuelles pour notre modèle composé de données uniquement internes à notre portefeuille modélisée par l'algorithme LightGBM.

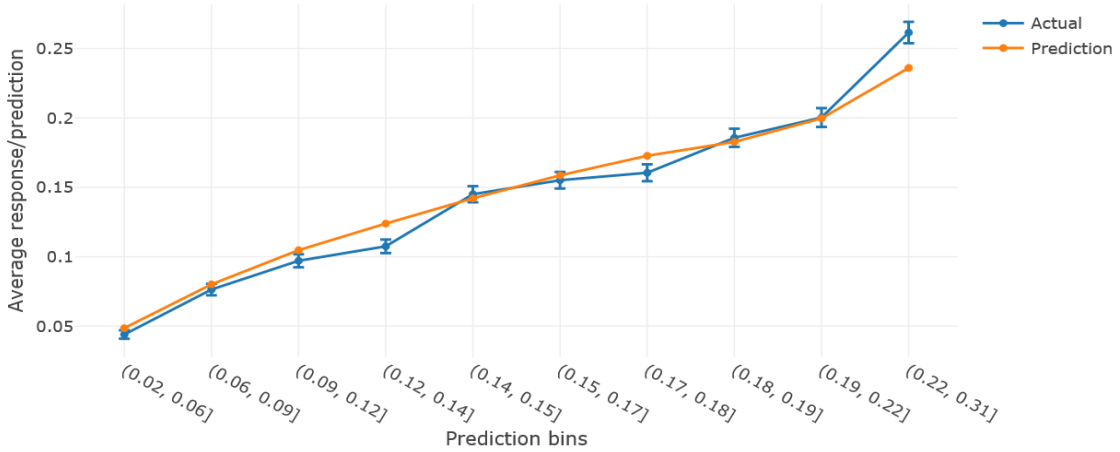


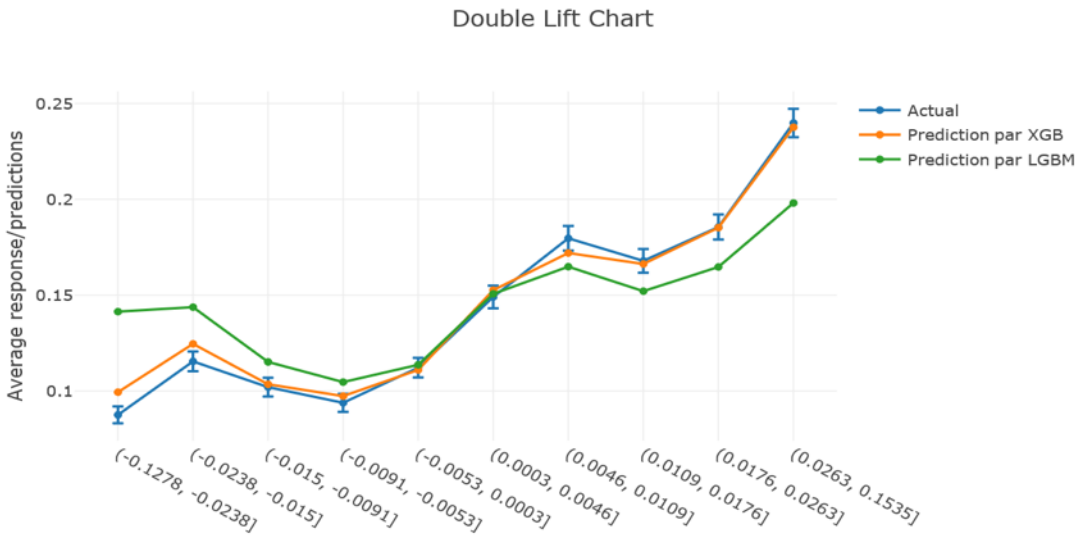
FIGURE 31 : LIFT CURVE DES PREDICTIONS[2] MODELISEES PAR LGBM POUR LE MODELE INTERNE

La *Lift Curve* nous montre sur l'axe des abscisses une segmentation en 10 subdivisions de fréquences (ici [0,02 ;0,06], [0,06 ;0,09] ...), présentant en comparaison sur l'axe des ordonnées les moyenne des fréquences prédites et fréquences observées accompagnées d'un intervalle de confiance à 0,975.

Les fréquences prédites en moyenne sont généralement superposées aux réelles et appartiennent globalement aux intervalles de confiance (6 cas sur 10 ici appartiennent aux intervalles). Ainsi, les prédictions sont assez bien prédites et moyennement calibré.

On décide finalement de représenter grâce à la double *Lift Curve* les fréquences prédites faces aux actuelles composées de données uniquement internes de notre portefeuille modélisées par les deux différents algorithmes décrits précédemment. L'avantage est que nous pourrons comparer les deux prédictions pour les mêmes subdivisions d'intervalles de fréquences tout en sachant que la variance de Poisson expliquée par le modèle XGBoost est supérieure à celle du modèle LightGBM.

[2] Indicateur de fréquence à titre visuel



FIGURE[3] 32 : DOUBLE LIFT CURVE COMPARANT XGB ET LGBM POUR LE MODELE INTERNE

La double *Lift Curve* nous montre sur l'axe des abscisses une segmentation en 10 subdivisions de différences des deux fréquence prédites (ici [-0,12 ; -0,02], [-0,02 ; -0,01] ...), présentant en comparaison sur l'axe des ordonnées les moyenne des fréquences prédites et fréquences réelles accompagnées d'un intervalle de confiance à 0,975.

Les fréquences prédites par XGBoost sont en général superposées aux fréquences actuelles (appartenant à 7 intervalles de confiance sur 10), face à des prédictions par LGBM éloignées des réelles et qui n'appartiennent globalement pas aux intervalles de confiance (8 bins sur 10 ici). Ainsi, les prédictions sont moins bonnes par l'algorithme LightGBM.

Modèle 4 :

Le 3ème modèle conçu nous permet de prédire la fréquence de sinistralité grâce aux variables internes de notre portefeuille, auxquelles nous ajoutons les variables externes de notre table de données externes :

- 5 variables qualitatives : marque, partenaire de distribution, protection d'écran, batterie, type de résolution
- 6 variables quantitatives : prix d'achat, poids, rapport d'écran sur téléphone, diagonale d'écran, résolution

On prend l'exposition du contrat comme poids dans notre modèle de prédiction.

[3] Indicateur de fréquence à titre visuel

La variance de Poisson expliquée nous affiche :

	Variance de Poisson expliquée	Temps d'exécution
XGBoost	7,28%	00 :04 :38
LightGBM	6,92%	00 :00 :00.0001

TABLEAU 9: RESULTAT DE PREDICTION DU MODELE EXTERNE PAR LES METHODES DE MACHINE LEARNING

Quant à la variance de Poisson expliquée pour ce modèle, la différence est relativement faible, mais il existe une claire amélioration lors de l'ajout de variables externes.

Nous décidons dans un premier temps de représenter grâce à la *Lift Curve*, les fréquences prédites faces aux actuelles pour notre modèle composé de données internes auxquels nous avons ajouté les données externes par l'algorithme XGBoost.

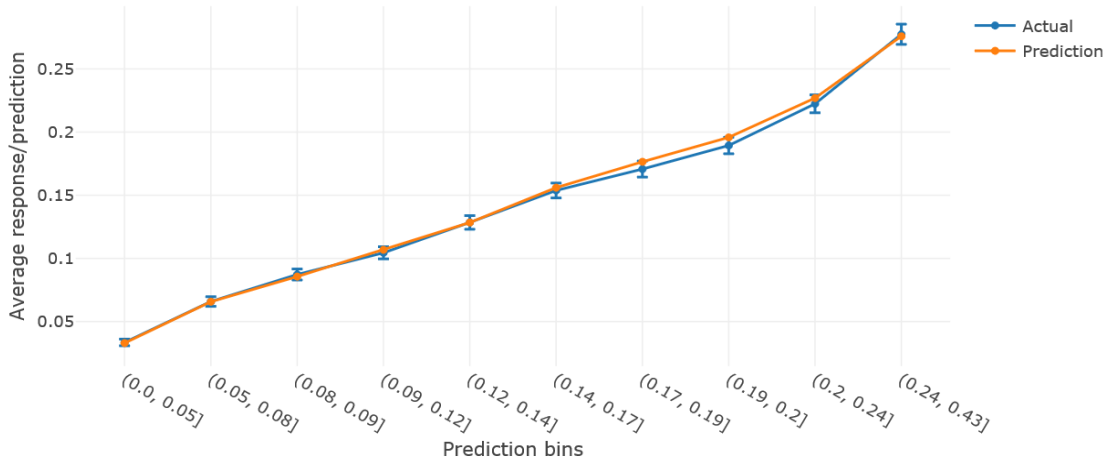


FIGURE 33 : LIFT CURVE DES PREDICTIONS[4] MODELISEES PAR XGB POUR LE MODELE EXTERNE

La *Lift Curve* nous montre une segmentation en 10 subdivisions de prédictions de fréquences, présentant en comparaison les fréquences réelles accompagnées d'un intervalle de confiance à 97,5%.

Les fréquences prédites sont superposées aux réelles et appartiennent à tous les intervalles de confiance. Ainsi, la représentation des prédictions est meilleure avec l'ajout de données externes.

[4] Indicateur de fréquence à titre visuel

Dans un second temps, nous représentons grâce à la *Lift Curve*, les fréquences prédites faces aux actuelles pour notre modèle composé de données internes auxquelles nous avons ajouté les données externes modélisées par l'algorithme LightGBM.

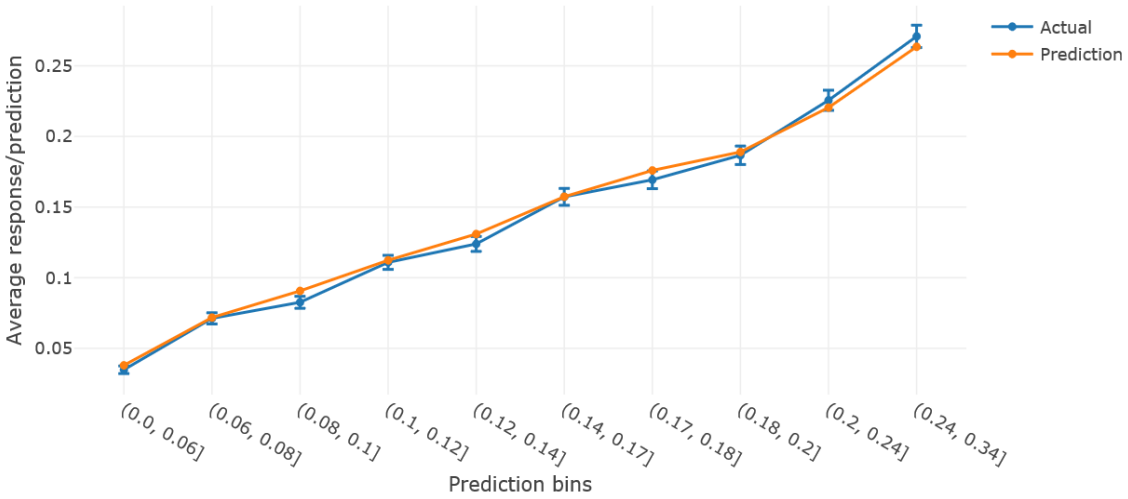


FIGURE 34 : LIFT CURVE DES PREDICTIONS[5] MODELISEES PAR LGBM POUR LE MODELE EXTERNE

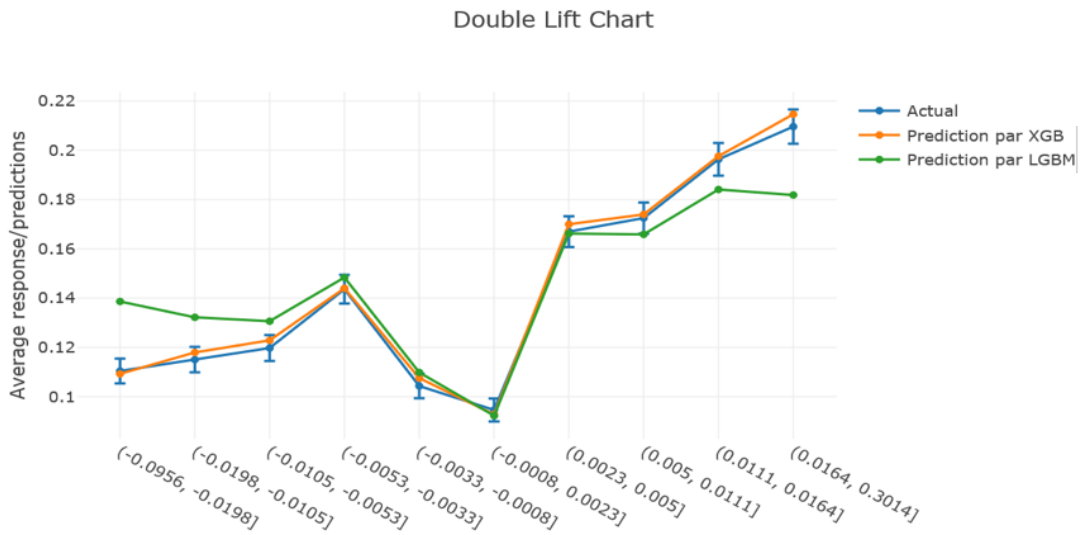
La *Lift Curve* nous montre une segmentation en 10 subdivisions de prédiction de fréquences, présentant en comparaison les fréquences réelles accompagnées d'un intervalle de confiance à 0,975.

Les fréquences prédites sont généralement proches des réelles et appartiennent globalement aux intervalles de confiance (7 sur 10 ici). Ainsi, les prédictions sont tout aussi bonnes par l'algorithme LightGBM que pour l'algorithme XGBoost.

On décide finalement de représenter grâce à la double *Lift Curve* le modèle de fréquences prédites faces aux actuelles composé de données internes auxquelles nous avons ajouté les données externes modélisées par les deux algorithmes décrits précédemment.

L'avantage est que nous pourrons comparer les deux prédictions pour les mêmes subdivisions d'intervalles tout en sachant que la variance de Poisson expliquée par le modèle XGBoost était très légèrement supérieure à celle du modèle LightGBM.

[5] Indicateur de fréquence à titre visuel



FIGURE[6] 35 : DOUBLE LIFT CURVE COMPARANT XGB ET LGBM POUR LE MODELE EXTERNE

La double *Lift Curve* nous montre une segmentation en 10 subdivisions des deux différentes prédictions de fréquences par les deux algorithmes. Elle nous représente en comparaison les fréquences réelles accompagnées d'un intervalle de confiance à 0,975.

Les fréquences prédites par XGBoost sont en général superposées aux fréquences actuelles (appartenant à tous les intervalles de confiance), face à des prédictions par LGBM éloignées des réelles et qui n'appartiennent globalement pas aux intervalles de confiance (6 bins sur 10 ici). Ainsi, les prédictions sont moins bonnes par l'algorithme LightGBM.

<u>Tableau comparatif des performances des modèles</u>

		XGBoost	LGBM
Variance de Poisson expliquée	Modèle basique	0	0
	Modèle tranches de prix	0,7%	0,7%
	Modèle variable internes	6,18%	5,15%
	Modèle variables internes + externes	7,28%	6,92%

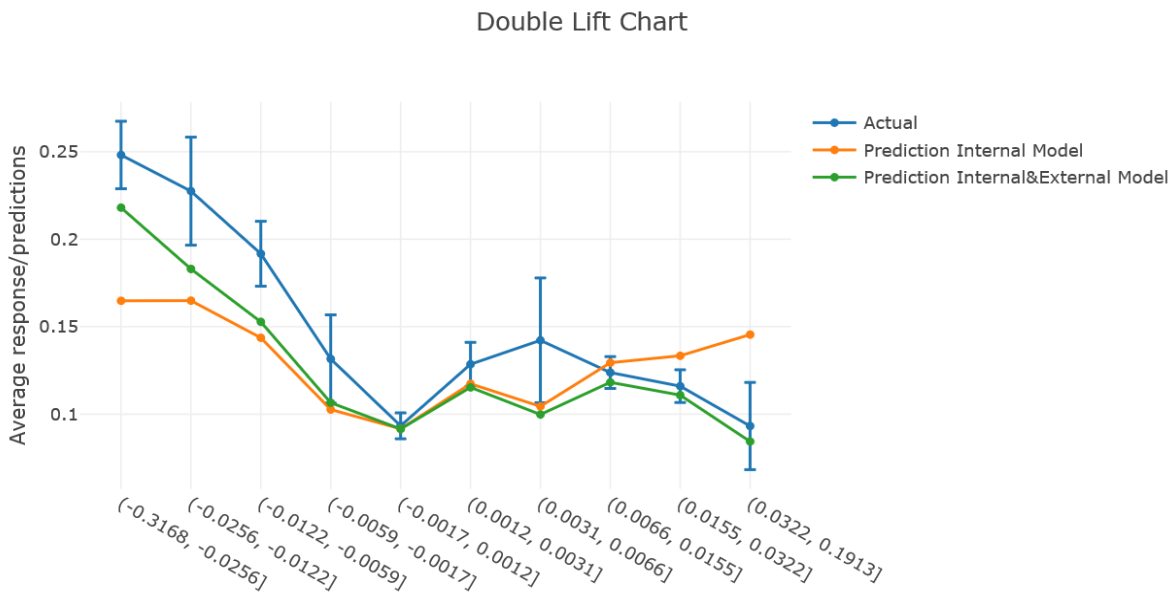
TABLEAU 10 : COMPARATIF DES MESURE DE PREDICTIONS PAR MODELES ET METHODES

Comme nous avons pu apercevoir précédemment, les prédictions par l'algorithme XGBoost sont meilleures que celles du LightGBM.

Ainsi, nous décidons de représenter la double *Lift Curve* comparant les prédictions des fréquences de sinistralités par le modèle composé de données internes auxquelles nous

[6] Indicateur de fréquence à titre visuel

avons ajouté les données externes modélisées par XGBoost face au modèle composé de données uniquement internes modélisée par le même algorithme. L'avantage est que nous pourrons comparer les deux prédictions pour les mêmes subdivisions d'intervalles.



FIGURE[7] 36 : DOUBLE LIFT COMPARANT LE MODELE INTERNE ET FINAL PAR XGB

La double *Lift Curve* nous montre une segmentation en 10 subdivisions des deux différentes prédictions de fréquences par les deux algorithmes. Elle nous représente en comparaison les fréquences réelles accompagnées d'un intervalle de confiance à 0,975.

Les fréquences prédites par XGBoost pour le modèle interne et externe sont en moyenne mieux prédites que celles comportant uniquement les données du modèle interne. Nous pouvons donc conclure que l'ajout de données exogènes dans notre modèle est plus avantageux en termes de prédiction de fréquence de sinistralité.

Nous décidons de représenter un graphique des prédictions des 10 modèles de téléphones les plus importants en termes d'exposition dans notre portefeuille :

[7] Indicateur de fréquence à titre visuel

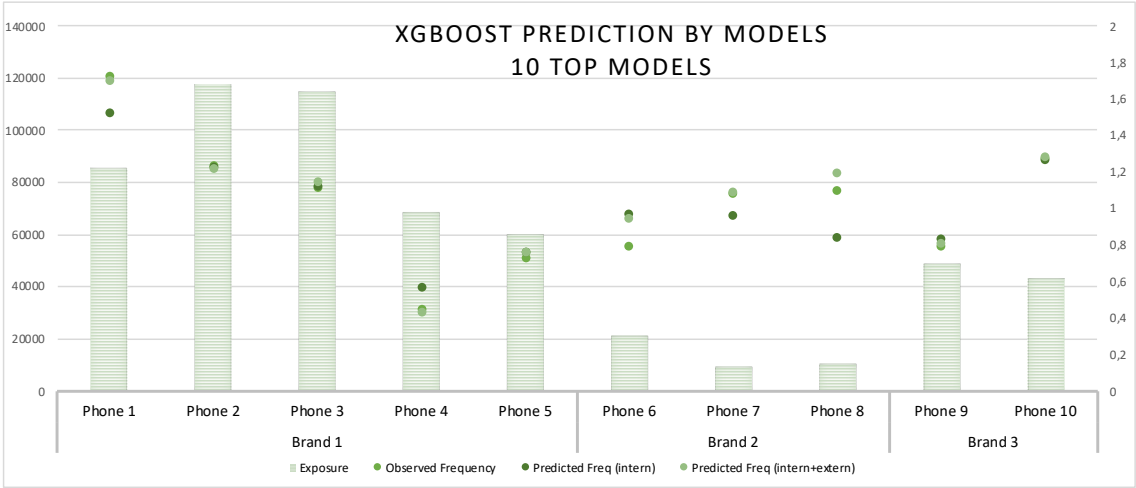


FIGURE 37 : GRAPHIQUE DE PREDICTIONS[8] DES TOP 10 MODELES DE TELEPHONES

On peut voir facilement que pour le modèle de téléphone 1 de la première marque, son indicateur de fréquence de sinistralité observé est de 1,8. L'indicateur de la fréquence prédite pour le même téléphone par le modèle XGBoost comportant uniquement les données internes (Predicted Freq (Intern)) est égal à environ 1,5 comparé à 1,81 pour le modèle XGBoost comportant les données internes et externes.

Il en est de même pour les autres téléphones représentés sauf pour le téléphone 6 appartenant à la marque 2, ainsi cela confirme encore une fois qu'en moyenne les données externes ajoutées nous permettent de mieux prédire la sinistralité.

3.3.3 Un modèle composé uniquement de variables externes ?

Dans le cadre de prédiction de sinistralité d'un téléphone qui sera mis en vente, il était intéressant de tester un modèle composé uniquement de variables externes et ainsi répondre à la problématique : « Est-il possible d'évaluer le risque de sinistre à partir de ses caractéristiques physiques ? ». A cette fin, une analyse comparative a été mené entre un modèle constitué uniquement des données fournies (variables internes) et un modèle composé uniquement de données externes, obtenues à partir de données de GSMArena. Nous avons prédit les fréquences de sinistralité à travers les deux précédents algorithmes (XGBoost, LightGBM).

[8] Indicateur de fréquence à titre visuel

Tableau comparaison des performances des modèles avec l'intégration de variables internes uniquement ou de variables externes uniquement.

		Variance de Poisson expliquée	Temps d'exécution
Externes uniquement	XGBoost	5,53%	00 :00 :49
	LightGBM	5,52%	00 :00 :00.0002
Internes uniquement	XGBoost	6,18%	00 :02 :23
	LightGBM	5,15%	00 :00 :00.0001

TABLEAU 11 : SECOND COMPARATIF DES MESURES DE PREDICTIONS PAR METHODES ET MODELES

Le tableau laisse supposer que les données externes fournissent autant d'informations que les seules données internes puisque les niveaux des variances expliquées de Poisson sont sensiblement similaires.

Ainsi, un modèle qui n'intègre que des données externes est performant car il est pratiquement équivalent à un modèle contenant uniquement des données internes. Ce constat montre le pouvoir prédictif des données externes. D'un point de vue opérationnel, cela permettrait par exemple d'établir un tarif de table au client reposant simplement sur peu de critères comme les dimensions du téléphone acheté par la personne, plus quelques autres critères.

Nous avons pu comparer jusqu'à présent deux modèles issus du Gradient Boosting et qui sont très semblable. Pour la suite du mémoire, nous utiliserons que l'algorithme XGBoost pour sa meilleure performance de prédiction.

3.4 Sélection et description des variables les plus influentes

Au regard de la quantité importante de variables intégrées dans nos modèles, plusieurs XGBoost sont utilisés afin de présélectionner les variables influentes sur la variable réponse (fréquence de sinistralité). Ces variables sont ensuite analysées avant d'être intégrées dans le modèle de prédiction. En effet, bien qu'un maximum d'informations aide toujours à mieux expliquer le risque à modéliser, il faut tout de même éviter la redondance d'information afin d'aider le modèle à mieux performer. Une analyse des corrélations est donc faite en amont avant de réaliser le modèle de prédiction.

Par ailleurs, bien que le XGBoost soit supposé intégrer la sélection des variables dans son algorithme, nous décidons tout de même de procéder à un tri des variables de

manière à pouvoir confronter l'efficacité de l'algorithme lorsque toutes les variables sont intégrées sans traitement préalable avec un modèle où les variables intégrées sont sélectionnées par d'autres méthodes.

3.4.1 Sélection des variables par l'algorithme XGBoost

Afin de réduire l'incorporation de bruit dans le modèle, nous tentons d'éviter d'intégrer des variables inutiles. Pour cela une présélection des variables est réalisée pour chaque ensemble de données provenant de sources différentes. Pour rappel, les sources de données sont les suivantes :

- Les données internes relatives au contrat, au risque et à l'assuré.
- Les données externes comportant les données physiques des appareils.

Ainsi, pour chaque ensemble de données, un modèle a été réalisé pour sélectionner les variables influentes provenant de cette source. Une fois que l'ensemble des XGBoost aura été réalisé, toutes les variables seront ensuite intégrées ensemble dans un modèle de XGBoost final.

XGBoost nous montre les variables les plus influentes dans les données internes et externes.

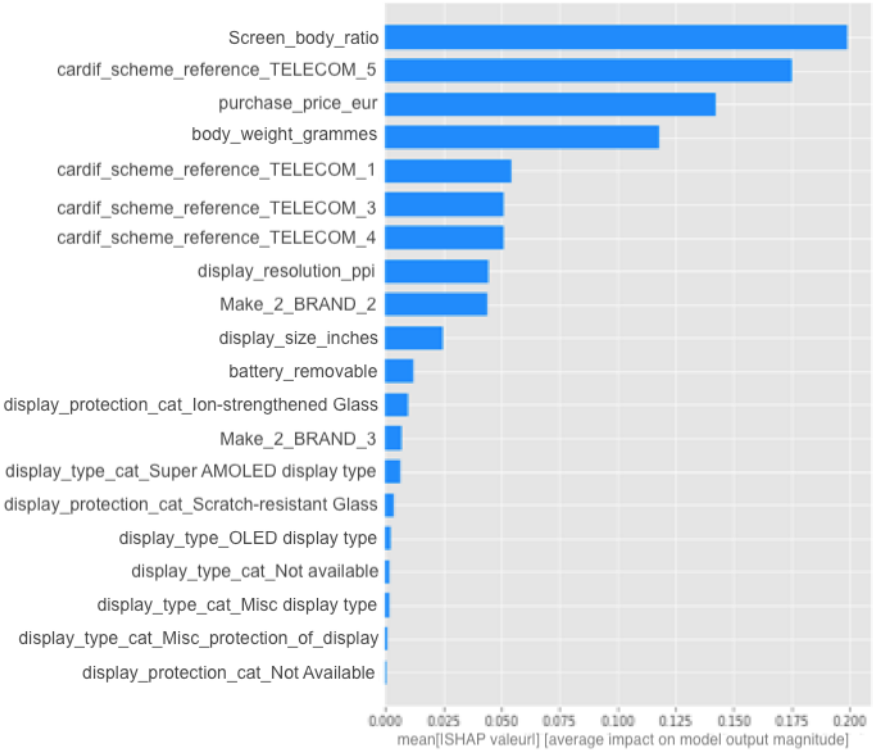


FIGURE 38 : DIAGRAMME XGB DES VARIABLES INFLUENTES DANS LE MODELE

Ainsi, nous pouvons lire sur le graphique ci-dessus que les variables les plus influentes sont :

- Screen_body_ratio : le ratio taille d'écran sur taille du téléphone est la variable la plus influente dans notre modèle.
- Cardif_scheme_reference_TELECOM_5 : la variable représentant le 5ème opérateur de notre portefeuille est aussi très influente dans notre modèle.
- Purchase_price_eur : le prix d'achat du téléphone.
- Body_weight_grammes : le poids du téléphone.

3.4.2 Mise en œuvre d'une sélection de variables par le diagramme de Shapley

Nous utilisons la méthode de Shapley pour confirmer notre précédente sélection des variables les plus influentes du modèle de jeu de données. Le diagramme de valeur de Shapley peut en outre montrer les relations positives et négatives des prédicteurs avec la variable cible. Nous allons donc visualiser ce graphique pour le modèle XGBoost réalisé avec les données internes et externes

Diagramme de Shapley :

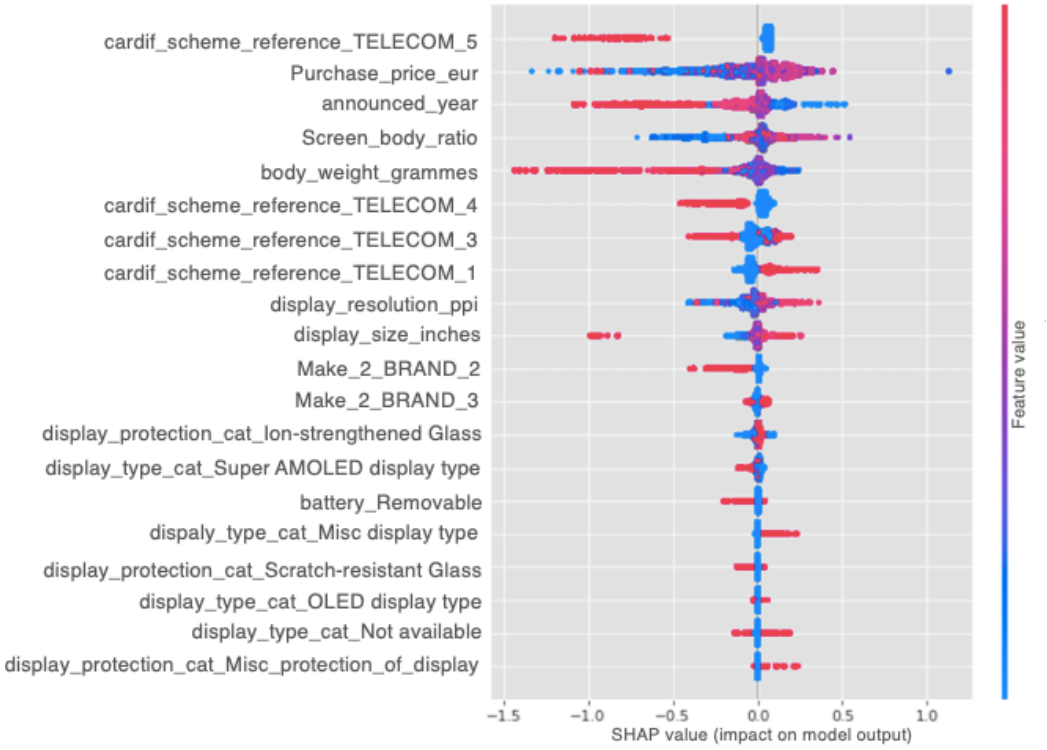


FIGURE 39 : DIAGRAMME DE SHAPLEY REPRESENTANT L'IMPORTANCE DES VARIABLES DANS LE MODELE

Ce graphique est constitué de tous les points des données du train. Il montre les informations suivantes :

- L'importance de la caractéristique : les variables sont classées par ordre décroissant.
- Impact : l'emplacement horizontal montre si l'effet de cette valeur est associé à une prédiction supérieure ou inférieure.
- Valeur originale : la couleur indique si cette variable est élevée (en rouge) ou faible (en bleu) pour cette observation.
- Corrélation : un niveau élevé du ratio d'écran a un impact haut et positif sur la fréquence de sinistralité. Le "haut" provient de la couleur rouge, et l'impact "positif" est indiqué sur l'axe des abscisses. Il en est de même pour le prix d'achat du téléphone. De même, nous dirons que le poids du téléphone est négativement corrélé avec la variable cible.

- Description des variables influentes :

Les variables les plus importantes sont classées par ordre décroissant, ainsi les variables les plus influentes du modèle sont : Cardif_scheme_reference_TELECOM_5, Purchase_price_eur, Announced_year, Screen_body_ratio et body_weight_grammes.

Il faut ensuite essayer d'interpréter l'impact et la corrélation à la variable cible : la fréquence de sinistralité de l'appareil.

- Le prix d'achat du téléphone est la seconde variable la plus importante du modèle, de plus cette variable a un impact haut et positif sur la fréquence de sinistralité (rouge et positif sur l'axe des abscisses). Ainsi nous pouvons dire que plus un téléphone est cher plus sa probabilité d'être sinistré est élevée.

- Le ratio taille d'écran sur taille du téléphone est tout aussi une variable influente dans notre modèle. De même que pour le prix d'achat, plus ce ratio d'écran est grand plus le téléphone risque d'être sinistré. Cela relève tout aussi de la logique, plus les bordures du tactile sont petites, plus le risque d'un bris d'écran et dysfonctionnement d'écran est courant.

- Le poids du téléphone est aussi une variable influente du modèle mais à l'inverse du prix d'achat et du ratio taille d'écran sur taille de téléphone, la variable a un impact haut et négatif. Ainsi plus un téléphone est lourd plus la fréquence de sinistralité sera faible.

3.5 Modèle final et sur-apprentissage

3.5.1 Mise en œuvre d'une sélection des variables

Après la présélection des variables, via le graphique d'importance d'XGBoost et le diagramme de Shapley, il est possible que certaines des variables explicatives utilisées dans le modèle ne soient pas significatives. Elles n'apportent pas d'informations sur la variable à expliquer et ne doivent donc pas figurer dans le modèle.

Il faut alors procéder à une deuxième sélection : le but étant de réduire notre modèle aux variables apportant une valeur au modèle final et permettant ainsi de mieux prédire notre fréquence de sinistralité.

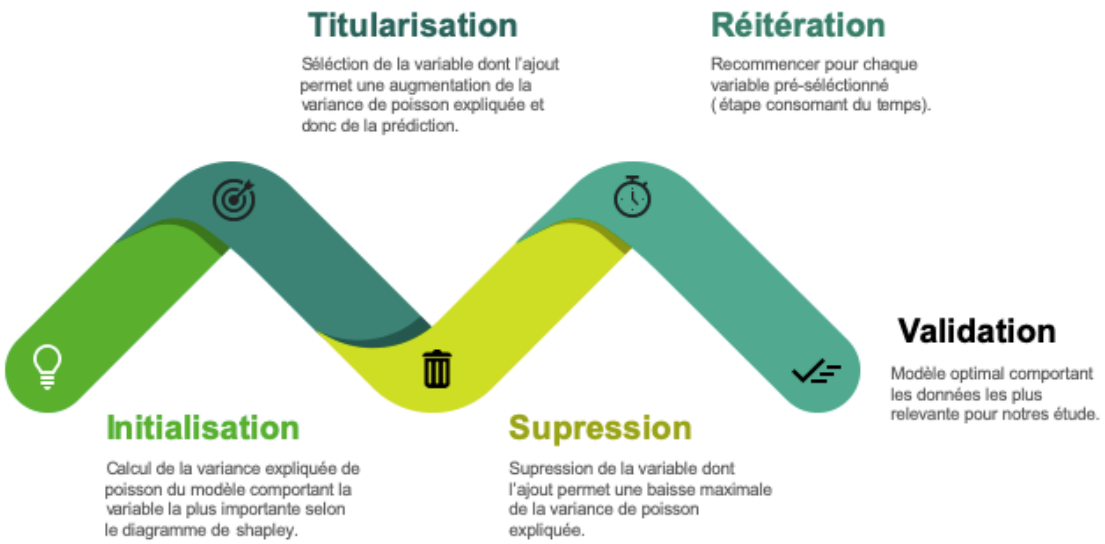


FIGURE 40 : METHODE DE SELECTION DE VARIABLES

Nous commençons donc notre modèle simplifié en ajoutant une à une les variables qui étaient classées parmi les plus importantes sur le diagramme de Shapley :

Variable	Variance de Poisson Expliquée
Make	0,002
Make + Cardif_scheme_reference	0,027
Make + Cardif_scheme_reference + Purchase Price	0,0617
Make + Cardif_scheme_reference + Purchase Price + Screen Ratio	0,0704

Make + Cardif_scheme_reference + Purchase Price + Screen Ratio + Body Weight	0,0723
Make + Cardif_scheme_reference + Purchase Price + Screen Ratio + Body Weight + Display protection	0,072755
Make + Cardif_scheme_reference + Purchase Price + Screen Ratio + Body Weight + Display protection + Announced year	0,0729
Make + Cardif_scheme_reference + Purchase Price + Screen Ratio + Body Weight + Display protection + Announced year + Display type	0,07274
Make + Cardif_scheme_reference + Purchase Price + Screen Ratio + Body Weight + Display protection + Announced year + Display Resolution	0,07262
Make + Cardif_scheme_reference + Purchase Price + Screen Ratio + Body Weight + Display protection + Announced year + Display Inches	0,0726
Make + Cardif_scheme_reference + Purchase Price + Screen Ratio + Body Weight + Display protection + Announced year + Battery	0,07265

Tableau 12 : Selection ascendante de variables

Suite à l'utilisation de la méthode précédente, les variables retenues pour le modèle simplifié sont donc :

Code Variable	Définition
Make	Marque du téléphone
Cardif_scheme_reference	Partenaire de vente du téléphone
Purchase_price	Prix d'achat du téléphone
Screen_body_ratio	Ratio taille écran sur taille téléphone
Body_weight_grammes	Poids du téléphone
Display_protection_cat	Catégorie de protection du téléphone
Announced_year	Date d'annonce du téléphone

Tableau 13 : Variables selectionnees

Notre modèle simplifié nous permet de passer du modèle précédant le plus optimal comportant 11 variables à ce modèle simplifié comportant uniquement 7 variables, soit 4

de moins pour un gain de 0,01% en termes de prédiction (on rappelle que la prédiction du modèle 4 était de 7,28% et celle du modèle simplifié est de 7,29%).

Comme nous avons pu apercevoir précédemment, les prédictions par l'algorithme XGBoost sont meilleures que celles du LightGBM. Ainsi, nous décidons de représenter la *Lift Curve* comparant les prédictions des fréquences de sinistralités par le modèle composé de données internes auxquelles nous avons ajouté les données externes modélisées et réduites comme expliqué plus haut par XGBoost. L'avantage est que nous pourrons comparer les deux prédictions pour les mêmes subdivisions d'intervalles.

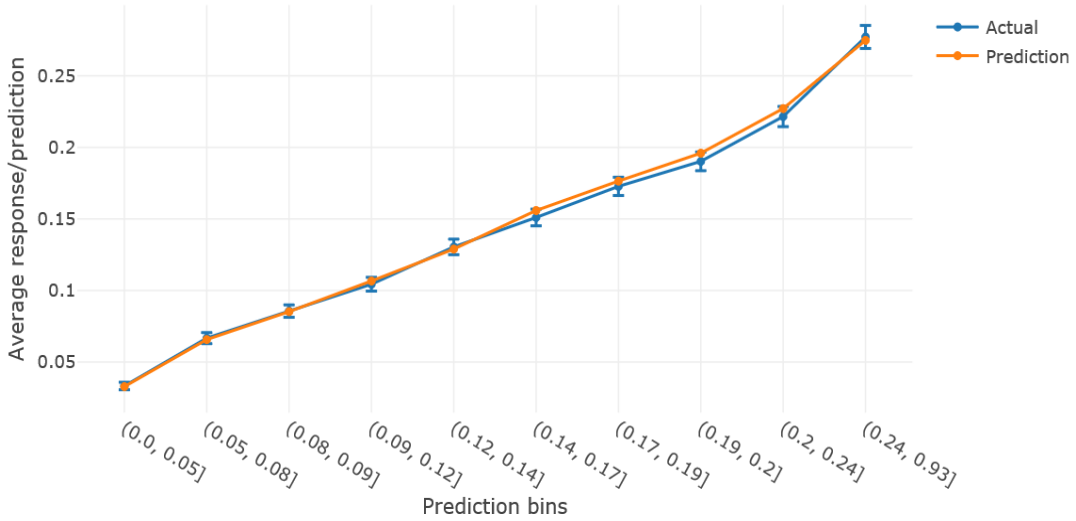


FIGURE 41 : LIFT CURVE DES PREDICTIONS[9] MODELISEES PAR XGB POUR LE MODELE REDUIT

La *Lift Curve* nous montre une segmentation en 10 subdivisions de prédiction de fréquences, présentant en comparaison les fréquences réelles accompagnées d'un intervalle de confiance à 0,975.

Les fréquences prédites sont superposées aux réelles et appartiennent à tous les intervalles de confiance. Ainsi, la représentation des prédictions est tout aussi bonne en ayant réduit notre modèle passant de 11 variable à uniquement 7 variables.

Ainsi, nous pouvons interpréter que l'ajout de données auxiliaires dans notre modèle réduit est toujours autant bénéfique en termes de prédiction dans nos modèles.

Nous utilisons encore une fois la méthode du diagramme de Shapley pour projeter les variables les plus influentes du modèle de jeu de données et leurs impacts.

[9] Indicateur de fréquence à titre visuel

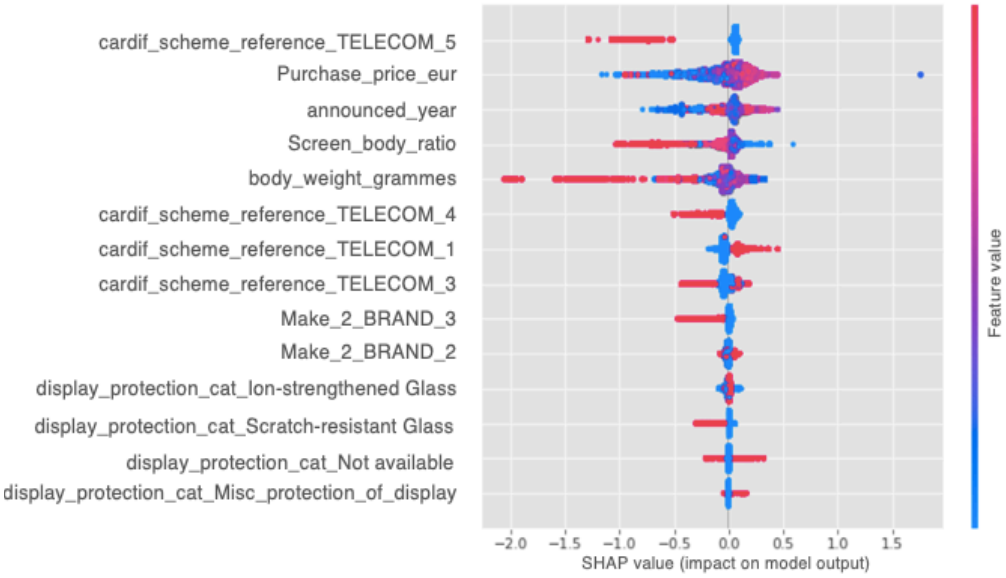


FIGURE 42 : DIAGRAMME DE SHAPLEY REPRESENTANT L'IMPORTANCE DES VARIABLES DANS LE MODELE REDUIT

Le diagramme de valeur de Shapley nous montre que la variable la plus importante du modèle est une variable interne sur l'opérateur TELECOM 5, par la suite les variables les plus importantes de notre modèle sont bien les variables externes : purchase_price, screen_body_ratio, announced_year, body_weigth etc.

Finalement nous décidons de mesurer le gain d'apport des données externes pour le modèle XGBoost final réduit grâce à la métrique de somme des erreurs absolues.

		Variance de Poisson expliquée	Somme des erreurs absolues	Gain (%)
XGBoost	Modèle basique	0%	10,4	0%
	Modèle tranches de prix	0,7%	9,5	8%
	Modèle variables internes	6,18%	7,1	32%
	Modèle réduit variables internes + externes	7,29%	5,7	45%

TABLEAU 14 : COMPARATIF DES MESURES DE PREDICTIONS PAR MODELES

Le résultat obtenu grâce à la métrique de somme des erreurs absolues est que le gain en précision après incorporation des données internes et externes sélectionnées est de 45% face au modèle basique, 37% face au modèle par tranches de prix et 13% face au modèle composé uniquement de données internes.

3.5.2 Cross Validation

Afin de vérifier par la suite si nos modèles présentent du sur-apprentissage, nous appliquons sur le modèle le plus performant et simplifié comportant les données les plus importantes une validation croisée.

Dans un premier temps la table d'apprentissage est divisée en 10 sous échantillons de taille égale, de façon aléatoire. Un de ces échantillons est utilisé en table de validation et les 9 autres servent de table d'entraînement. Nous répétons ce processus 10 fois. L'erreur du modèle est calculée sur chacune des 10 tables de validation utilisées.

La moyenne des variances de Poisson expliquées pour ce modèle est égale à 6,2%, soit 1,1% de moins que pour le modèle de découpage de la table de données simple (70% pour l'apprentissage et 30% pour le test). Le modèle simplifié présente ainsi du sur-apprentissage. Sa puissance réelle de prédiction est donc inférieure au résultat précédemment annoncé.

- Tarification d'un nouveau modèle de téléphone

La principale étude basée sur les *Cross Validation* doit nous mettre en situation de sortie d'un nouveau téléphone. Nous tentons de prédire la fréquence de sinistralité de ce dernier en l'écartant de notre table d'apprentissage. Nous utilisons tous nos champs de données internes et externes disponibles afin de tarifer ce produit au mieux au vu de son risque de sinistralité.

Dans notre étude, les modèles sont choisis en fonction des expositions les plus élevées dans notre portefeuille parmi les 3 marques : Apple, Samsung, Huawei.
Nous utilisons la méthode LKOCV (Leave k Out Cross Validation), nous rappelons que cette méthode est la même que la précédente, sauf que la table de validation contiendra k couples entrée/sortie et la table d'entraînement correspondante sera la table d'apprentissage à laquelle nous aurons retiré les k couples de la table de la validation. Le processus sera répété jusqu'à avoir obtenu toutes les tables d'entraînements et de validations possibles en retirant k éléments.

Ici nous choisissons k=1, tel qu'on soit dans la situation où le modèle écarté est le futur modèle de téléphone portable annoncé et prochainement disponible à l'achat.
Voici un schéma représentant la méthode LOOCV, qui est un cas particulier de LKOCV pour k=1 avec nos 10 modèles :

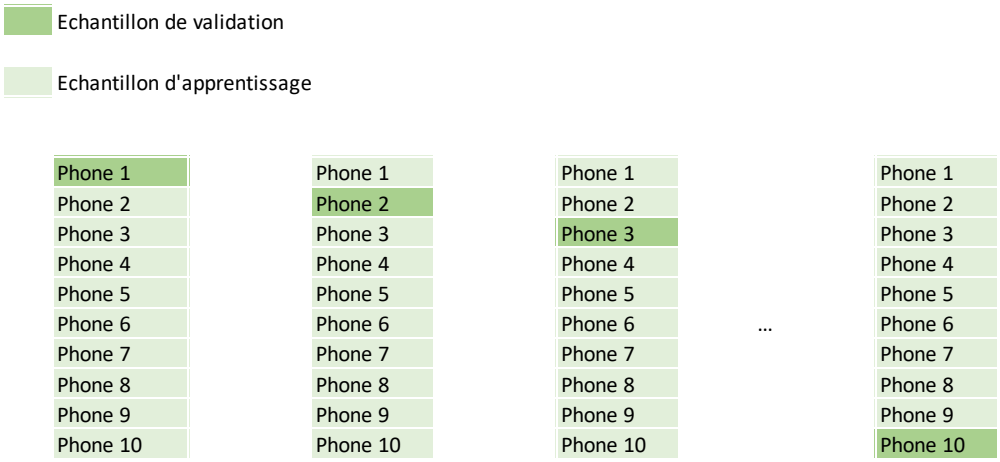


FIGURE 43 : METHODE LOOCV APPLIQUE A NOTRE TOP 10 DE MODELES DE TELEPHONES

Ainsi dans notre première simulation, nous considérons que le téléphone 1 est celui qui sortira dans le futur et nous essayons de prédire sa fréquence de sinistralité. Nous faisons de même pour la suite des téléphones portables.

Les résultats des fréquences prédites face aux fréquences observées dans notre portefeuille sont représentés dans le graphique ci-dessous :

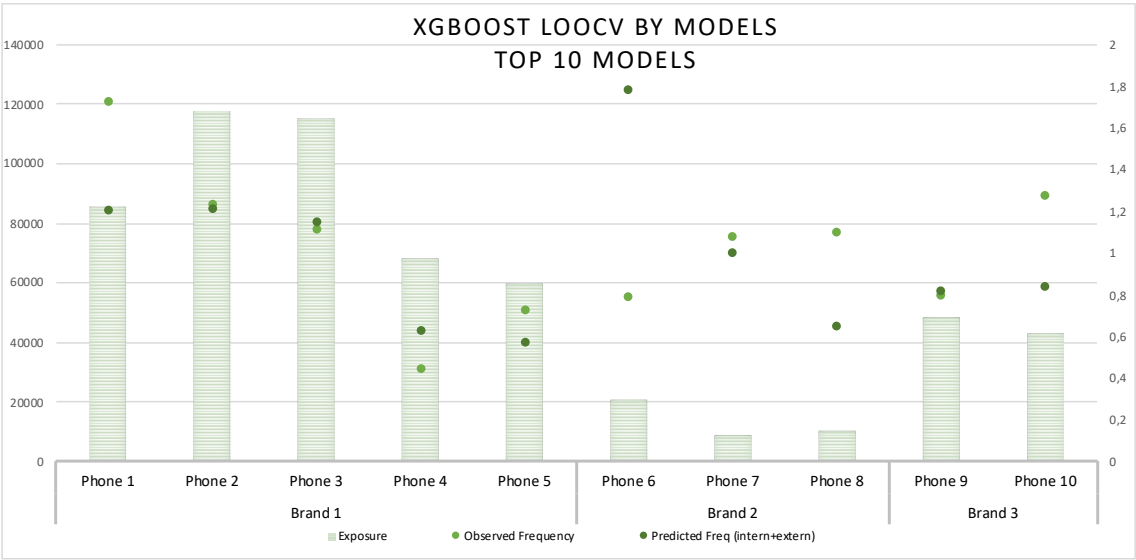


FIGURE 44 : RESULTAT DE PREDICTIONS[10] APRES METHODE LOOCV

Dans l'exemple du premier modèle de téléphone appartenant à la première marque, grâce aux informations des 9 autres téléphones portables, l'indicateur de

[10] Indicateur de fréquence à titre visuel

fréquence de sinistralité prédite est de 1,25. Or en réalité l'indicateur de fréquence observée pour ce même téléphone portable est égal à 1,8. Alors que pour le téléphone 2, l'indicateur de fréquence prédite est superposée à l'indicateur de fréquence observée. Le modèle prédit parfaitement la fréquence de sinistralité de ce téléphone portable.

Néanmoins, le modèle prédit bien uniquement pour 3 téléphones portables sur les 10 présents, dont les 2 ayant les expositions les plus élevés (téléphone 2, téléphone 3). Pour les autres téléphones portables les prédictions ne sont pas en accord avec la réalité. Nous ne pouvons donc malheureusement pas nous baser sur ce type de modèle afin de tarifier de futurs produits dès leurs annonces.

3.5.2 Ajout de Bruit

C'est un défi d'entrainer un modèle *Machine Learning* qui fonctionnera bien sur des données non renseignées, et pas seulement sur celles sur lesquelles notre modèle a été formé. Cette caractéristique est appelée la généralisation et entraine bien sur des entrées non observées. En effet, il existe des méthodes telles que la séparation test/apprentissage ou la validation croisée pour mesurer le degré de généralisation de notre modèle.

Nous pouvons classer la performance du modèle dans 3 cas :

- Le modèle est peu performant sur l'ensemble des données d'apprentissage et sur les nouvelles données - modèle d'ajustement insuffisant (Underfitting).

- Le modèle est performant sur l'ensemble des données d'entraînement et peu performant sur les données invisibles - modèle d'ajustement excessif (Overfitting).

- Le modèle apprend notre ensemble de données d'entraînement et fonctionne bien sur les données invisibles, il est capable de généraliser – modèle bien ajusté.

Dans notre cas, nous sommes confrontés à des modèles d'ajustement excessif ; il est donc important de surveiller les performances pendant l'entrainement pour détecter s'il y a sur-apprentissage.

La régularisation consiste à modifier notre algorithme d'apprentissage pour réduire son erreur de généralisation mais pas son erreur d'entraînement. Une des méthodes de régularisation les plus courantes dans le *Machine Learning* est le bruit : il s'agit d'introduire un bruit stochastique dans le processus d'entraînement.

- Qu'est-ce que le bruit Gaussien ?

Le bruit Gaussien est un bruit statistique ayant une fonction de densité de probabilité égale à celle de la distribution de la loi normale. Il est également connu sous le nom de distribution Gaussienne. La fonction de densité de probabilité p d'une variable aléatoire Gaussienne z est donnée par :

$$p_G(z) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(z-\mu)^2}{2\sigma^2}}$$

où z représente le niveau de gris, μ la valeur moyenne et σ l'écart-type. Dans notre cas, nous choisissons μ=0. En résumé, les valeurs que le bruit peut prendre sont distribuées de manière Gaussienne.

L'ajout de bruit augmente la taille de notre ensemble de données. Lorsqu'un modèle s'entraîne, du bruit aléatoire est ajouté à chaque échantillon de formation, ce qui constitue une forme d'augmentation des données. De plus, lorsque nous utilisons du bruit, nous augmentons le caractère aléatoire de nos données et le modèle est moins capable d'apprendre des échantillons d'entraînement puisqu'ils changent à chaque itération. Par conséquent, le modèle apprend des caractéristiques plus vastes et présente moins d'erreurs de généralisation.

Lorsque nous appliquons du bruit, nous créons de nouveaux échantillons à proximité des échantillons d'apprentissage et la distribution des données d'entrées est ainsi lissée. Cela permet au modèle d'apprendre beaucoup plus facilement à partir de nos données. Ainsi, dans un premier temps nous multiplions notre table de données par 4. En effet, nous nous retrouvons dans un cas où notre table présente 2 millions de contrats.

Par la suite, pour chaque variable quantitative sélectionnée dans notre modèle final, nous calculons l'écart-type associé et informant notre modèle qui ajoutera ce bruit par conséquent.

Voici les écarts-types calculés pour les variables quantitatives :

	Standard deviation
Body weight (grammes)	40
Screen body ratio (%)	5,80
Purchase price (EUR)	307

TABLEAU 15 : STANDARD DEVIATION DES VARIABLES QUANTITATIVES SELECTIONNEES

Après ajout du bruit, nous mesurons la performance de notre modèle grâce à la variance de Poisson expliquée. Celle-ci nous donne un résultat de 5,95%.

Le modèle est tout aussi performant que celui de la *Cross Validation*. Cela nous confirme que la puissance réelle de prédiction sans sur-apprentissage est de 6%.

Nous représentons le graphique de la *Lift Curve* :

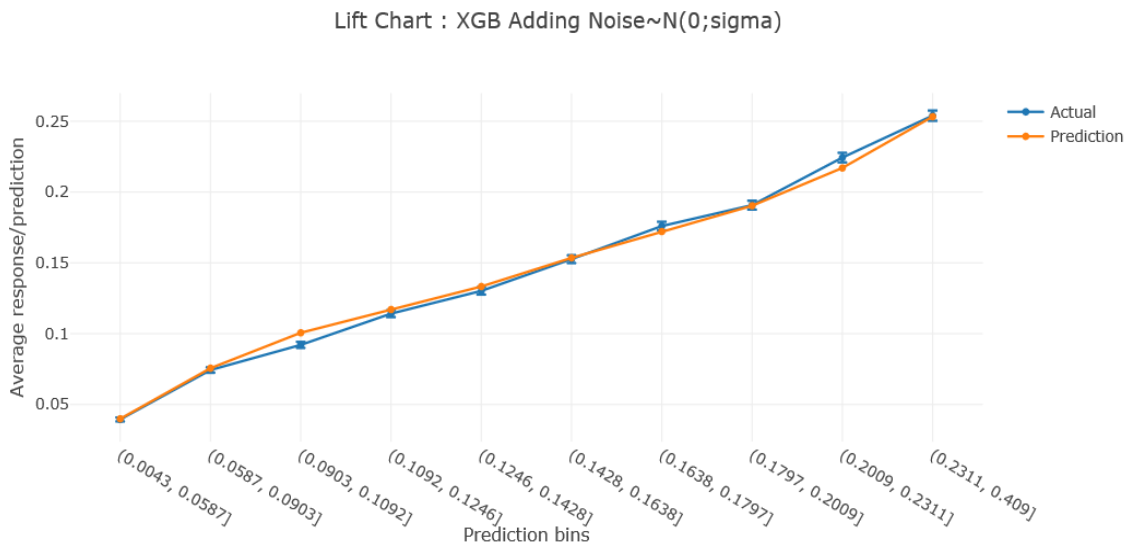


FIGURE 45 : LIFT CURVE DES PREDICTIONS[11] MODELISEES PAR XGB POUR LE MODELE AVEC AJOUT DE BRUIT

La *lift Curve* nous montre une segmentation en 10 subdivisions de prédictions de fréquences, présentant en comparaison les fréquences réelles accompagnées d'un intervalle de confiance à 0,975 du modèle d'ajout de bruit.

Les fréquences prédites sont superposées aux réelles et appartiennent à 8 bins sur 10 alors que tous les intervalles de confiance y était compris pour le modèle réduit présentant du sur-apprentissage. Ainsi, la représentation des prédictions est en moyenne bonne et représente la réalité du pouvoir de prédiction sans sur-apprentissage.

Nous représentons par la suite la méthode du diagramme de Shapley pour projeter les variables les plus influentes du modèle de jeu de données et leurs impacts après ajout de bruit.

[11] Indicateur de fréquence à titre visuel

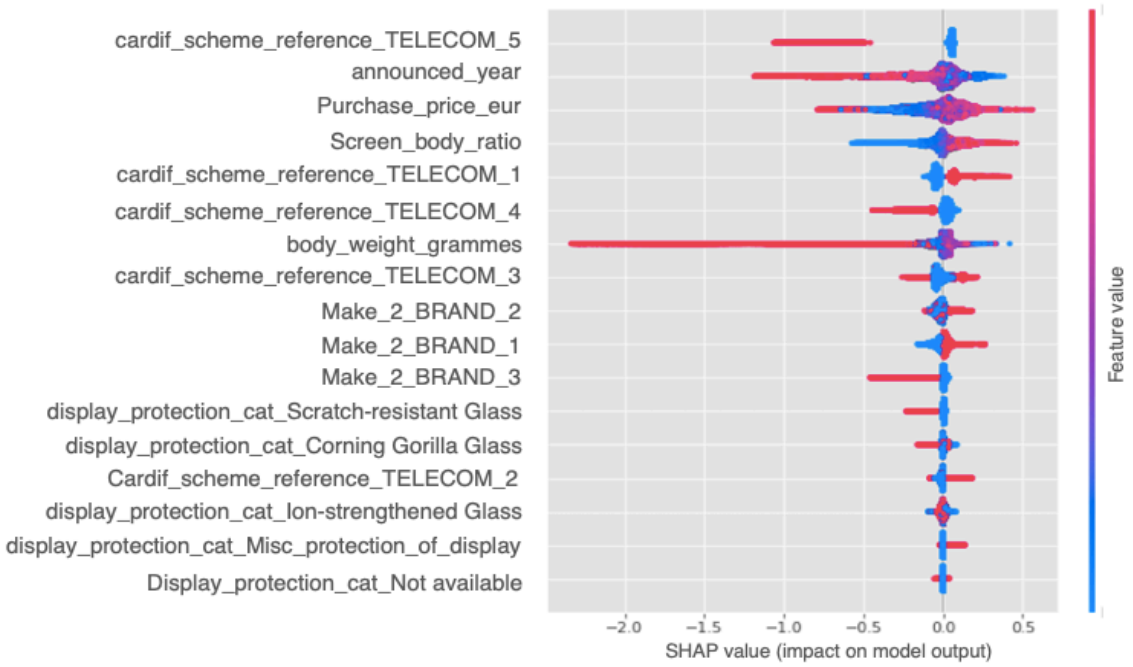


FIGURE 46 : DIAGRAMME DE SHAPLEY REPRESENTANT L'IMPORTANCE DES VARIABLES DU MODELE AVEC AJOUT DE BRUIT

Le diagramme de valeur de Shapley nous montre que les variables les plus importantes de notre modèle sont bien les variables externes : purchase price, screen body ratio, announced year, body weigth...

Pour les variables Screen body ratio et body weight grammes, l'ajout du bruit a impacté le modèle et est très visible sur le diagramme de Shapley.
L'impact négatif pour le poids du téléphone est décuplé, ainsi plus le téléphone est léger plus celui-ci aura sa fréquence de sinistralité élevé. Inversement pour la variable Screen_body_ratio qui a été impacté positivement, ainsi plus le ratio de la taille d'écran sur la taille de téléphone est élevé plus l'appareil mobile est sujet à plus grande fréquence de sinistralité.

Benchmark avec la régression des variables influentes et analyse

Ce chapitre met en œuvre l'intégration dans la régression des variables sélectionnées dans les étapes précédentes. Cette régression est réalisée par l'intermédiaire d'une étude XGBoost, qui en plus de posséder des qualités de sélection de variables, est également un modèle de prédiction. De plus, le graphique de Shapley nous a permis de confirmer nos variables choisies.

Les modèles de prédiction généralement utilisés en assurance dommage sont les modèles linéaires généralisés, c'est pourquoi nous commencerons par énoncer la théorie de ces modèles avant de réaliser une comparaison qui permettra de mettre en parallèle le modèle de XGBoost et les modèles linéaires généralisés. L'efficacité de la sélection automatique des variables réalisée par l'algorithme XGBoost sera ensuite remise en question en testant la performance de l'algorithme avec et sans les différents traitements de variables réalisés dans le chapitre précédent.

4.1 Modèles linéaires généralisés

4.1.1 Modèles linéaires

Les modèles linéaires sont conçus pour prédire et expliquer les valeurs d'une variable aléatoire Y dite variable à expliquer ou variable réponse à partir d'autres variables aléatoires $\mathrm{X}_1, \ldots, \mathrm{X}_\mathrm{p}$ dites variables explicatives. Nous cherchons ainsi une relation entre les n variables explicatives et la variable réponse Y. *Si* $Y \in L^1$ ($E(|Y|) < \infty$), nous définissons la fonction de régression de Y *sur* $X = (X_1, \ldots, X_p)$ *par* :

$$\begin{aligned} f : R^n &\rightarrow R \\ x &\rightarrow E(Y \mid X = x) \end{aligned}$$

Un modèle de régression linéaire est défini par une équation de la forme :

$$Y = X\beta + \varepsilon$$

Où :

- *Y est un vecteur aléatoire de dimension n,*
- *X est une matrice de taille $n \times p$ connue, appelée matrice du plan d'expérience,*
- *β est le vecteur de dimension p des paramètres inconnus du modèle,*
- *ε est le vecteur de dimension n des erreurs.*

Les hypothèses concernant le modèle sont :

$$(H) \begin{cases} (H1): \ rg(X) \ = \ p \\ (H2): \ E[\varepsilon] \ = \ 0, Var(\varepsilon) = \sigma^2 I_n \end{cases}$$

L'hypothèse (H2) signifie que les erreurs sont centrées, de même variance (homoscédasticité) et non corrélées entre elles. L'estimateur des moindres carrés $\hat{\beta}$ est défini comme suit :

$$\hat{\beta} = \ arg \min_{\beta \in R^p} \sum_{i=1}^{n} (y_i - \sum_{j=1}^{p} \beta_j x_{ij})^2 \ = \ arg \min_{\beta \in R^p} \|Y - X\beta\|^2$$

$\hat{\beta}$ est unique, sans biais et donné par la formule :

$$\hat{\beta} = \ (X'X)^{-1} \ X'Y$$

4.1.2 Modèles linéaires généralisés

Les modèles linéaires généralisés (GLM) sont une extension de la régression linéaire ordinaire dans laquelle la variable réponse Y_i peut être considérée comme une réalisation d'une distribution spécifique de la famille exponentielle. En régression linéaire classique, une variation constante d'un prédicteur entraîne une variation constante de la variable expliquée, peu importe le point de l'ensemble de définition considéré. A contrario, les GLM permettent de modéliser une dépendance non linéaire dans le sens où c'est l'image de la variable réponse par une fonction arbitraire g (appelée fonction de lien) qui dépend linéairement des variables explicatives :

$$g(E(Y|X)) \ = \ \beta^T X$$

Le modèle linéaire classique est retrouvé si g est définie comme fonction identité. Nous cherchons à estimer la fonction de régression :

$$E(Y|X) \ = \ g^{-1} \ (\beta^T X)$$

Un modèle GLM est donc composé de 3 composantes clés :

- une composante aléatoire spécifiant la distribution $Y|X$.
- une composante déterministe qui met en relation un paramètre η aux prédicteurs X.
- une fonction de lien g monotone et différentiable qui relie les deux précédentes composantes.

La composante aléatoire : la variable réponse

La variable réponse Y est la variable aléatoire que l'on cherche à expliquer. La composante aléatoire, spécifiant la distribution conditionnelle de la variable de réponse, Y_i (pour la ième des n observations échantillonnées indépendamment), étant donné les valeurs des variables explicatives dans le modèle.

Dans la formulation originale de Nelder et Wedderburn, la distribution de Y_i est un membre d'une famille exponentielle, telle que les familles de distributions Gaussiennes (normales), binomiales, de Poisson, Gamma ou Gaussiennes inverses. Des travaux ultérieurs ont cependant étendu les GLM aux familles exponentielles multivariées (comme la distribution multinomiale), à certaines familles non exponentielles (comme la distribution bi-négative-binomiale à deux paramètres) et à certaines situations dans lesquelles la distribution de Y_i n'est pas complètement spécifiée.

Dans cette étude nous nous concentrons sur le cas des familles exponentielles. Tout d'abord, la distribution de probabilité est choisie en accord avec la forme des données à modéliser. Voici un exemple de schéma non exhaustif en fonction du type de données à étudier.

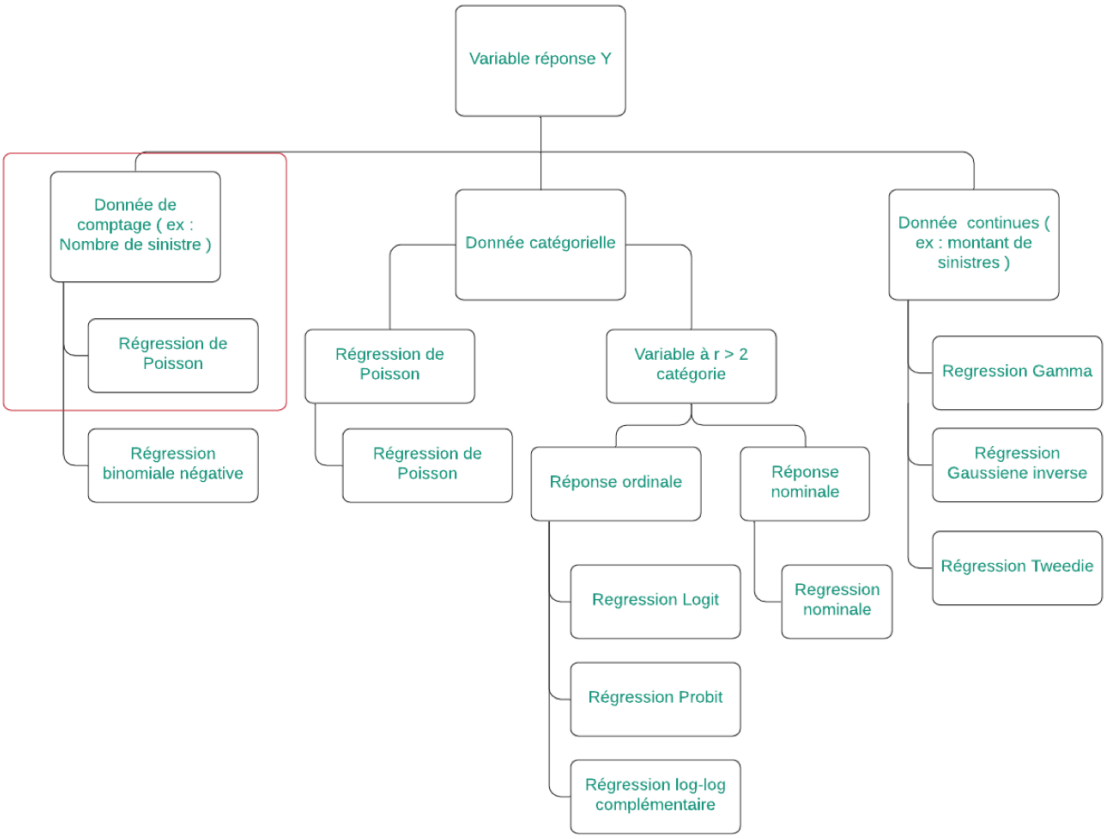


FIGURE 47 : SCHEMA DES TYPES DE REGRESSION EN FONCTION DES DONNEES

Pour les données de comptage, il convient donc de choisir une distribution discrète : la loi de Poisson. Ces lois doivent faire parties de la famille exponentielle qui correspond aux densités de la forme :

$$f_{\theta,\phi}(y) = \exp\left[\frac{y\theta - b(\theta)}{a(\phi)}\right] \times \exp\left[c(y,\phi)\right]$$

Avec :

- $b(\cdot)$ une fonction de classe C^3 de dérivée première inversible,
- $a(\cdot)$ et c$(\cdot)$ des fonctions dérivables,
- θ le paramètre naturel lié aux deux premiers moments de la loi,
- ϕ le paramètre de dispersion.

Deux propriétés concernant le paramètre naturel θ découlent de l'écriture de la densité pour les lois appartenant à la famille exponentielle :

- $E(Y) = b'(\theta)$.
- $V(Y) = b"(\theta)\phi$.

La composante déterministe :

La composante linéaire, autrement dit prédicteur linéaire est la quantité qui intègre les informations sur les variables indépendantes dans le modèle. Le symbole η désigne un prédicteur linéaire. Il est lié à la valeur attendue des données par le biais de la fonction de liaison.

η est exprimé comme une combinaison linéaire de paramètres inconnus β. Les coefficients de la combinaison linéaire sont représentés par la matrice des variables indépendantes X. η peut donc être exprimé comme :

$$\eta = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p$$

Les variables explicatives à intégrer dans le modèle sont à choisir avec soin. En effet, pour la cohérence et performance du modèle, le nombre de variables utilisées ne doit pas être trop important. C'est pourquoi, il est nécessaire de sélectionner parmi toutes les variables à disposition, les variables dont le pouvoir explicatif est le plus important afin de réussir à obtenir un juste équilibre.

La fonction de lien :

Il s'agit de la fonction qui va contenir le lien entre μ, l'espérance de $Y|X$, et η, tel que :

$$g(\mu) = \eta$$

En pratique, la fonction de lien souvent utilisée est la fonction logarithme car elle permet d'obtenir un modèle multiplicatif : g(x) = ln(x).

On obtient alors :

$$\ln\ (E[Y|X]) = \eta$$

$$E[Y|X]\ =\ g^{-1}\ (\beta_0\ +\beta_1 X_1\ +\ldots+\beta_p X_p)$$

$$E[Y|X] = exp(\beta_0)\ \times\ exp(\beta_1 X_1)\ \times\ldots\times\ exp(\beta_p X_p)$$

Estimation des paramètres de Poisson :

Dans l'étude réalisée, l'intérêt porte tout particulièrement sur la loi de Poisson de paramètre λ qui modélise le nombre de sinistres, de densité :

$$f_\lambda(y)\ =\ exp(-\lambda)\ \frac{\lambda^{\ y}}{y!}$$

$$= \exp(y\ln(\lambda) - \lambda - \ln y\,!)$$

Par rapport à la mesure de comptage, avec $y \in N, \theta = ln\lambda, a(\phi) = \phi = 1,$

$$b(\theta) = exp(\theta) = \lambda, \qquad et\ \ c(y,\varphi) = -ln\ y!$$

De plus,

$$E(Y) = \frac{\partial b(\theta)}{\partial\theta} = \frac{\partial exp(\theta)}{\partial\theta} = \exp(\theta) = \ b'(\theta)$$

$$V(Y) = a(\phi)\frac{\partial^2 b(\theta)}{\partial^2\theta} = \frac{\partial^2 exp(\theta)}{\partial^2\theta}\phi = \exp(\theta) = b"(\theta)\ .$$

Notre but est d'estimer la moyenne $\mu_i = \ E[y_i|x_i]$, pour $i = 1, \ldots, n$. Rappelons que nous avons la fonction de lien $g(\mu_i) = \eta_i$, qui lie l'espérance μ_i au paramètre $\eta_i = x_i^T\beta$. Nous pouvons donc d'abord estimer les coefficients β, comme $\hat{\beta}$, et ensuite utiliser ces estimations pour soutenir que :

$$g(\widehat{\mu_i}) = x_i^T\hat{\beta}, i = 1, \ldots, n$$

C'est-à-dire :

$$\widehat{\mu_i} = g^{-1}(x_i^T\hat{\beta}), i = 1, \ldots, n$$

Nous utilisons le maximum de vraisemblance. La vraisemblance des échantillons (x_i, y_i), i = 1, ... n, sous réserve de x_i, i = 1, ... n, est :

$$L(\theta) = \prod_{i=1}^{n} f(y_i;\theta_i,\phi)$$

Écrit en fonction de $\theta = (\theta_i, \ldots, \theta_n)$ pour indiquer la dépendance du paramètre naturel.

La log-vraisemblance du modèle est :

$$l(\theta) = \sum_{i=1}^{n} \ln\ (\ f(y_i;\theta_i,\phi)) = \ \sum_{i=1}^{n} \frac{y_i\theta_i - b(\theta_i)}{a(\phi)} + \sum_{i}^{n} c(y_i,\phi)$$

Nous voulons maximiser cette probabilité logarithmique sur tous les coefficients β ∈ $\mathbb{R}^p$; il s'agit véritablement d'une fonction de β, car chaque paramètre naturel θ_i peut en effet être écrit en termes de moyenne μ_i de la distribution exponentielle de la famille, et $\mu_i = \ x^T\beta$, i = 1, ... n. Nous pouvons donc écrire :

$$l(\beta) = \sum_{i=1}^{n} y_i\theta_i - b(\theta_i) - \ln(y_i)\,!$$

Pour être plus concret, avec la fonction de lien canonique g ; rappelons qu'il s'agit de la fonction de lien qui fixe $\theta_i = \eta_i = x_i^T\beta$, pour i = 1, ..., n.

Ensuite, le log vraisemblance afin de maximiser $\hat{\beta}$, est :

$$l(\beta) = \sum_{i=1}^{n} y_i x^T\beta - exp(x_i^T\beta) - \ln(y_i)\,!$$

La dérivée de l'équation précédente en fonction de β est égale à :

$$\frac{\partial l(\beta)}{\partial \beta} = \sum_{i=1}^{n} \Big(y_i - exp(x_i^T\beta)\Big)x_i$$

$$\frac{\partial l(\beta)}{\partial \beta} = 0$$

Dans le cas général, nous ne savons pas résoudre explicitement cette équation. L'estimateur du maximum de vraisemblance est calculé numériquement, souvent grâce l'algorithme de Newton-Ramphson ou l'algorithme IRLS (Iterative Reweighted Least Squares).

<u>Prise en compte de l'exposition :</u>

Les modèles linéaires généralisés proposent la prise en compte de deux types de variables :

- Les variables intégrées au modèle, qui se verront affecter des coefficients en fonction du résultat de la régression
- Des variables en « offset », c'est-à-dire des variables de tarification intégrées dans le modèle avec des coefficients contraints

Une variable dite « offset » peut être introduite dans l'équation du modèle, ce qui nous permet également de modéliser une fréquence de sinistralité (nombre de sinistres par unité d'exposition) au lieu de compter sans compromettre la distribution naturelle des données. Il est important de tenir compte de ce paramètre pour ne pas modéliser de la même façon deux risques similaires qui auraient été exposés sur des durées distinctes.

Les modèles de comptage tiennent compte de ces différences en déplaçant la variable d'exposition vers le côté droit de l'équation de régression et en prenant le log de cette variable dans le modèle avec un coefficient égal à un.

$$E[N|X, exposition] \;=\; e \times (exp(\beta_0) \times exp(\beta_1 X_1) \times \ldots \times exp(\beta_p X_p))$$
$$\ln(E[N|X, exposition]) = \; X\beta + ln(exposition).$$

Cette variable logarithmique, ln(exposition) ou une variable construite de manière similaire est souvent appelée la variable « offset ».

4.2 Pénalisation de la régression

Les estimations des moindres carrés sont obtenues en minimisant l'erreur quadratique résiduelle. Nous rappelons que l'estimateur est défini comme suit :

$$\hat{\beta} = \; arg \min_{\beta \in R^p} \sum_{i=1}^{n} (\, y_i - \sum_{j=1}^{p} \beta_j x_{ij})^2 \quad = \; arg \min_{\beta \in R^p} \|Y - X\beta\|^2$$

Il existe deux raisons pour lesquelles cette méthode n'est pas suffisante :

- L'interprétabilité du modèle : nous sommes amenés à sélectionner un sous-ensemble de variables explicatives afin de pouvoir ressortir l'ordre d'importance des variables explicatives.

- La précision des estimateurs : les estimations des moindres carrées présentent souvent un faible biais tandis que sa variance est assez importante. La précision des prédictions peut parfois être améliorée en réduisant certains coefficients. Ce faisant, nous sacrifions un peu de biais pour réduire la variance des valeurs prédites et pouvons donc améliorer la prédiction globale. La régression pénalisée est utilisée afin de contrôler le paramètre du modèle. Nous présentons dans ce qui suit la régression de Lasso.

- **Régression de Lasso**

Lasso (Least Absolute Shrinkage and Selection Operator) est une méthode de réduction des coefficients de régression appliquée aux modèles GLM. Elle consiste à appliquer une pénalisation proportionnelle à la norme L1 du vecteur de paramètres β lors du calcul de la fonction de vraisemblance ($\|\beta\|_{L_1} = \sum_i |\beta_i|$)

Avec la fonction de pénalité nous avons dans notre cas :

$$arg \min_{\beta \in R^p} \sum_{i=1}^{n} (- l(\beta) \ + \lambda \sum_{j=1}^{p} |\beta_j|$$

Avec λ (λ ≥ 0) un paramètre (coefficient de pénalité) qui permet de contrôler l'impact de la pénalité, il reste à fixer.

4.3 Mise en œuvre d'une sélection de variables

Afin de sélectionner les variables les plus explicatives parmi celles présélectionnées par le modèle d'Extrême Gradient Tree Boosting, une méthode Forward selection est utilisée. L'utilisation de cette méthode permettra de comparer l'efficacité de sélection automatique des variables de l'algorithme d'Extrême Gradient Tree Boosting et du diagramme de Shapley avec la sélection réalisée par cette procédure.

- **Critère de sélection**

Le critère AIC (Akaïke Information Criterion) est un critère qui repose fondamentalement sur la minimisation de la perte d'information de Kullback-Leibler (K-L). Cette dernière peut être appréhendé comme une distance entre la réalité des données et le modèle utilisé pour leur approximation. Akaike (1973) a trouvé une relation qui liait cette information et la log vraisemblance du modèle testé, en corrigeant cette dernière par le nombre de paramètres. Le critère AIC est alors défini de la manière suivante :

$$AIC \ = \ -2\ln(\mathcal{L}) \ + 2k$$

Où $ln(\mathcal{L}) = \ ln\left(\mathcal{L}(\hat{\theta})\right)$ est la fonction de vraisemblance maximisée, et k est le nombre de paramètres libres dans le modèle. Le modèle avec une valeur AIC minimale est choisi comme le meilleur modèle pour s'adapter aux données.

Le critère AIC sera utilisé pour la mise en application de la procédure de régression ascendante.

- **Méthode de régression**

Il existe deux types de méthodes pour les régressions pas à pas :

<u>The forward selection (la procédure ascendante) :</u>

Il s'agit d'une régression où le modèle initial possède une seule variable puis, à chaque itération, la variable qui permet de diminuer le critère AIC est intégrée au modèle. Ce processus est répété jusqu'à ce que l'ajout d'une variable entraîne une augmentation de l'AIC du modèle.

The backward selection (la procédure descendante) :

Cette procédure correspond au cheminement inverse de la Forward Selection. Il s'agit d'une régression où le modèle initial comprend toutes les variables candidates puis, à chaque itération, la variable qui dégrade le plus le critère AIC est retirée du modèle. Le processus est arrêté lorsque le retrait de toute variable du modèle implique une hausse de l'AIC.

4.3.1 Benchmark des modèles Machine Learning

Pour tester l'apport des données externes provenant de GSMArena, par rapport aux autres données, les variables sont ajoutées au GLM et sont comparées par la suite au modèle XGBoost et LGBM réalisé précédemment.

Nous décidons de comparer deux modèles et réalisons donc :

- Modèle 1 : prédiction des fréquences de sinistralité via un GLM par variables internes
- Modèle 2 : prédiction des fréquences de sinistralité via un GLM par variable internes + externes

Lors de notre modélisation nous décidons d'appliquer une pénalisation Lasso à nos modèles linéaires généralisés.

Modèle 1 :

Le 1er modèle de Benchmark à travers le GLM conçu nous permet de prédire la fréquence de sinistralité grâce aux variables internes dont notre portefeuille dispose :

- 2 variables qualitatives : marque et partenaire de distribution
- 1 variable quantitative : prix d'achat du téléphone

Nous prenons l'exposition du contrat comme poids dans notre modèle de prédiction.

La variance de Poisson expliquée est la suivante :

	Variance de Poisson expliquée	Temps d'exécution
GLM	5,14%	00 :00 :10

Tableau 16 : Resultat de prediction du modele interne par GLM LASSO

Le modèle GLM composé uniquement de données internes propose une prédiction au même niveau que celle de l'algorithme LGBM pour un temps d'exécution moins rapide.

Nous pouvons aussi remarquer que la variance de Poisson expliquée est relativement faible mais reste supérieure à un modèle basé sur les tranches de prix.

Puis, nous décidons de représenter grâce à la *Lift Curve* les fréquences prédites faces aux actuelles pour notre modèle composé de données uniquement internes à notre portefeuille modélisée par un GLM.

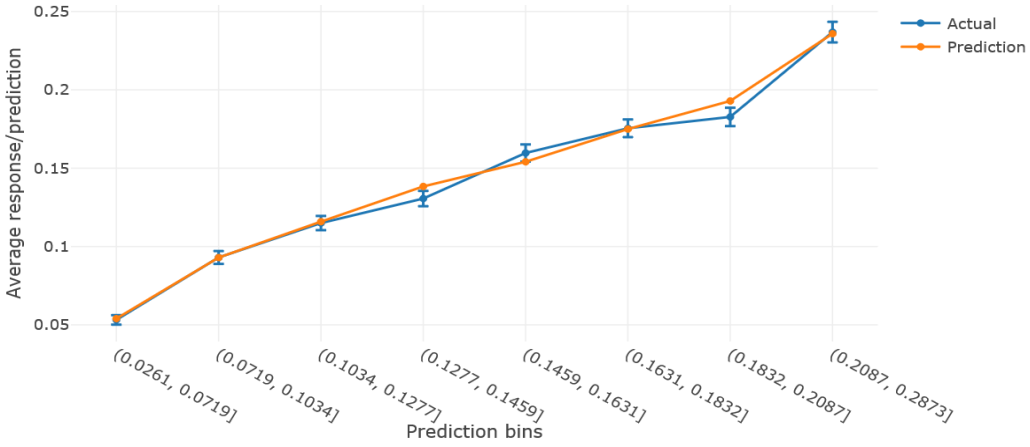


FIGURE 48 : LIFT CURVE DES PREDICTIONS[12] MODELISEES PAR GLM LASSO POUR LE MODELE INTERNE

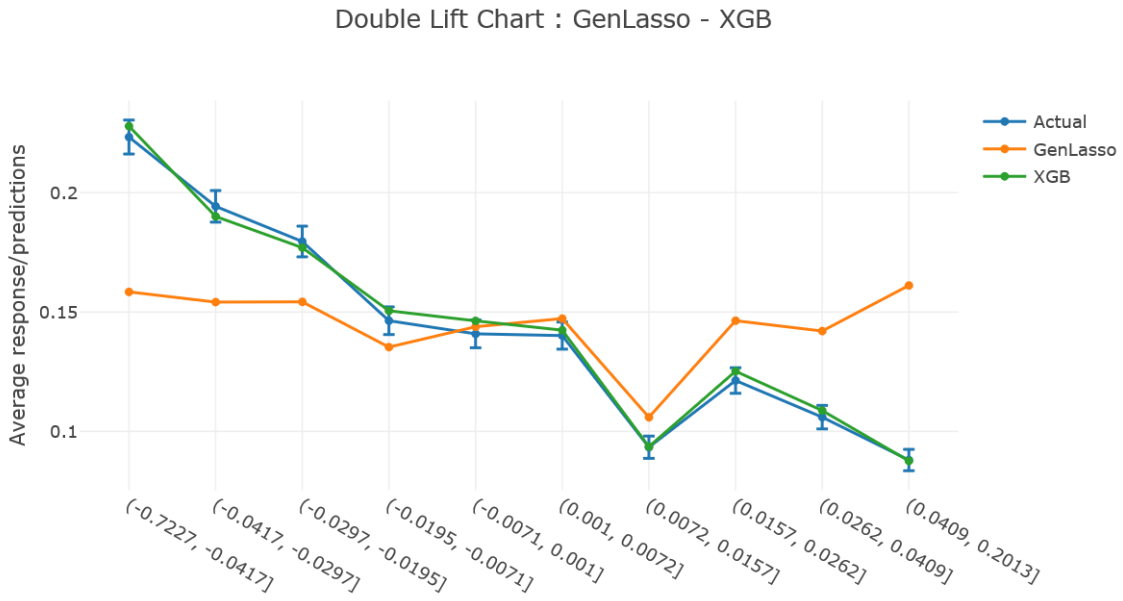
La *Lift Curve* nous montre sur l'axe des abscisses une segmentation en 8 subdivisions de fréquences (ici [0,02 ;0,07], [0,07 ;0,10] ...), présentant en comparaison sur l'axe des ordonnées les moyenne des fréquences prédites et réelles accompagnées d'un intervalle de confiance à 0,975.

Les fréquences prédites en moyenne sont proches des réelles et appartiennent globalement aux intervalles de confiance (6 cas sur 8 ici appartiennent aux intervalles). Ainsi, en moyenne les prédictions sont bonnes par GLM.

Finalement afin de comparer efficacement notre meilleur algorithme de prédiction : XGBoost, nous décidons de le confronter à notre modèle GLM pénalisé par la méthode Lasso grâce à la double *Lift Curve*. Les fréquences prédites faces aux actuelles composées de données uniquement internes de notre portefeuille modélisées par les deux différents algorithmes décrits précédemment.

[12] Indicateur de fréquence à titre visuel

L'avantage est que nous pourrons comparer les deux prédictions pour les mêmes subdivisions d'intervalles de fréquences tout en sachant que la variance de Poisson expliquée par le modèle XGBoost est supérieure à celui de modèle GLM pénalisé.



FIGURE[13] 49 : DOUBLE LIFT CURVE COMPARANT LES PREDICTIONS MODELISEES PAR GLM LASSO ET XGB POUR LE MODELE INTERNE

La double *Lift Curve* nous montre une segmentation en 10 subdivisions des deux différentes prédictions de fréquences par les deux algorithmes. Elle nous représente en comparaison les fréquences réelles accompagnées d'un intervalle de confiance à 97,5%.

Les fréquences prédites par XGBoost sont pratiquement toutes en moyenne superposées aux fréquences actuelles (appartenant à 9 intervalles de confiance sur 10), face à des prédictions par GLM éloignées des réelles et qui n'appartiennent globalement pas aux intervalles de confiance (uniquement 2 bins sur 10 ici). Ainsi, le modèle XGBoost est plus performant que le GLM en termes de prédiction.

Modèle 2 :

Le 2ème modèle benchmark nous permet de comparer cette fois-ci les prédictions de la fréquence de sinistralité grâce aux variables internes de notre portefeuille dispose auxquelles nous ajoutons les variables externes de notre table de données externe :

[13] Indicateur de fréquence à titre visuel

- 5 variables qualitatives : marque, partenaire de distribution, protection d'écran, batterie, type de résolution.
- 6 variable quantitative : prix d'achat, poids, rapport d'écran sur téléphone, diagonale d'écran, résolution.

On prend l'exposition du contrat comme poids dans notre modèle de prédiction. La variance de Poisson expliquée nous affiche :

	Variance de Poisson expliquée	Temps d'exécution
GLM	7,13%	00 :04 :81

TABLEAU 17 : RESULTAT DE PREDICTION DU MODELE FINAL PAR GLM

Nous gagnons 2% en termes de prédiction grâce à l'ajout des données externes. La variance de Poisson expliquée reste néanmoins inférieure à celle du modèle XGBoost réalisée plus haut.

Nous représentons encore une fois grâce à la *Lift Curve* les fréquences prédites faces aux actuelles pour notre modèle composé de données internes auxquels nous avons ajouté les données externes par l'algorithme GLM pénalisé par la méthode Lasso

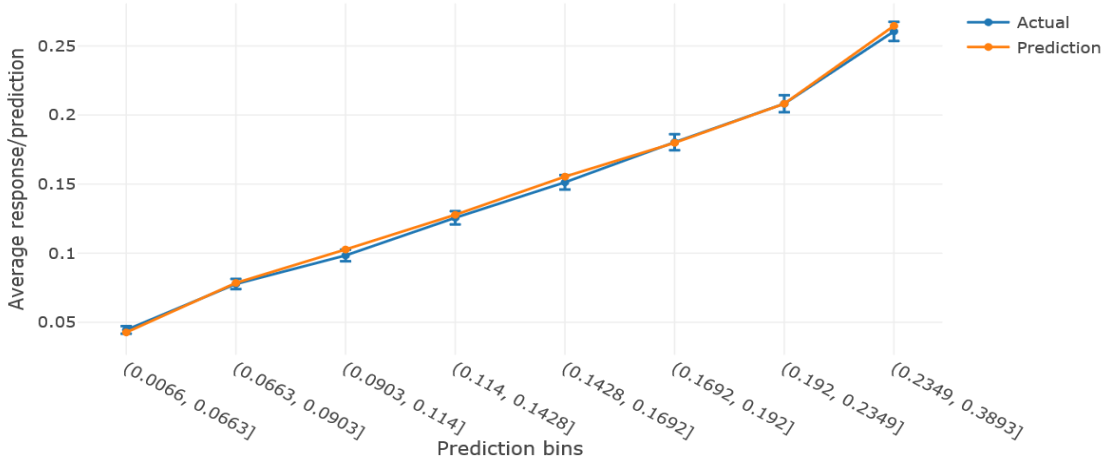


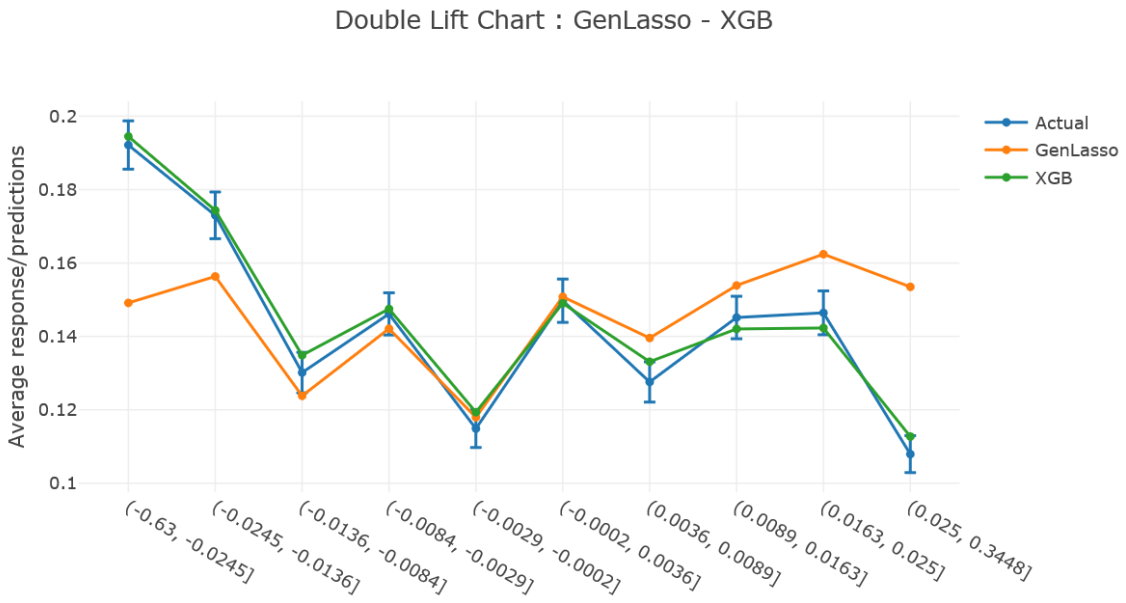
FIGURE 50 : LIFT CURVE DES PREDICTIONS[14] MODELISEES PAR GLM LASSO POUR LE MODELE FINAL

La *Lift Curve* nous montre une segmentation en 8 subdivisions de prédiction de fréquences, présentant en comparaison les fréquences réelles accompagnées d'un intervalle de confiance à 97,5%.

[14] Indicateur de fréquence à titre visuel

Les fréquences prédites sont superposées aux réelles et appartiennent à tous les intervalles de confiance. Ainsi, les prédictions sont meilleures avec l'ajout de données externes.

On décide finalement de représenter grâce à la double *Lift Curve* les fréquences prédites faces aux actuelles composées de données internes auxquelles nous avons ajouté les données externes modélisées par les deux algorithmes décrits précédemment. L'avantage est que nous pourrons comparer les deux prédictions pour les mêmes subdivisions d'intervalles tout en sachant que la variance de Poisson expliquée par le modèle XGBoost est très légèrement supérieure à celui du modèle GLM.



FIGURE[15] 51 : DOUBLE LIFT CURVE COMPARANT LES PREDICTIONS MODELISEES PAR GLM LASSO ET XGB POUR LE MODELE FINAL

La double *Lift Curve* nous montre une segmentation en 10 subdivisions des deux différentes prédictions de fréquences par les deux algorithmes. Elle nous représente en comparaison les fréquences réelles accompagnées d'un intervalle de confiance à 0,975.

Les fréquences prédites par XGBoost sont en général superposées aux fréquences actuelles (appartenant à 9 de confiance sur 10), face à des prédictions par GLM éloignées des réelles et qui n'appartiennent globalement pas aux intervalles de confiance (4 bins sur 10 ici). Ainsi, les prédictions sont moins bonnes par l'algorithme du GLM.

[15] Indicateur de fréquence à titre visuel

Tableau comparatif des performances des modèles

		XGBoost	GLM Lasso
Variance de Poisson expliqué	Modèle variable internes	6,18%	5,14%
	Modèle variable internes + externes	7,28%	7,13%

TABLEAU 18 : COMPARATIF DES MESURES DE PREDICTIONS PAR MODELES ET METHODES

Comme nous avons pu voir précédemment, les prédictions par l'algorithme XGBoost sont meilleures que celles du GLM Lasso mais l'écart reste limité. Ainsi l'apport de données externes reste toujours profitable à nos modèles en termes de prédiction.

4.3.2 Limites du GLM

Il existe plusieurs limites à l'utilisation des GLM. Nous pouvons citer la détection et modélisation d'interactions entre les variables, ainsi que l'utilisation de variables quantitatives continues.

Pour le premier cas, il convient à priori aux statisticiens de mesurer les interactions et les corrélations entre les différentes variables. L'insertion de deux variables fortement corrélées peut entraîner un effet redondant sur la variable à expliquer, et ainsi impliquer un effet de double pénalisation ou gratification pour certains individus. Des tests de corrélations doivent être effectués au préalable de façon à prévenir ces risques.

Pour le second cas, les valeurs prises par une variable quantitative continue sont souvent nombreuses. Ainsi, pour une variable de ce type, le coefficient de régression prendra en compte l'ensemble des valeurs prises par la variable. Il arrive souvent que certaines valeurs soient extrêmes et non représentatives à cause du faible nombre d'une modalité au sein de la table. Afin de pouvoir obtenir des estimateurs consistants pour les variables quantitatives, il est préférable de les transformer en variables qualitatives discrètes. Une seconde solution (généralement préférable à la première) qu'un actuaire pourrait envisager serait d'effectuer un découpage en classes des variables continues.

4.4 Mesure de l'apport des données externes

Après notre présélection des variables, via le graphique d'importance d'XGBoost et le diagramme de Shapley, il est possible que certaines des variables explicatives utilisées dans le modèle ne soient pas significatives. Ainsi, elles n'apportent pas d'information sur la variable à expliquer et ne doivent donc pas figurer dans le modèle.

Il faut alors procéder à une deuxième sélection : le but étant de réduire notre modèle aux variables apportant une valeur au modèle final et permettant ainsi de mieux prédire notre fréquence de sinistralité.

Nous décidons à présent d'appliquer la méthode Forward résumé dans le schéma ci-dessous afin de sélectionner nos plus importantes variables pour le modèle final GLM Lasso :

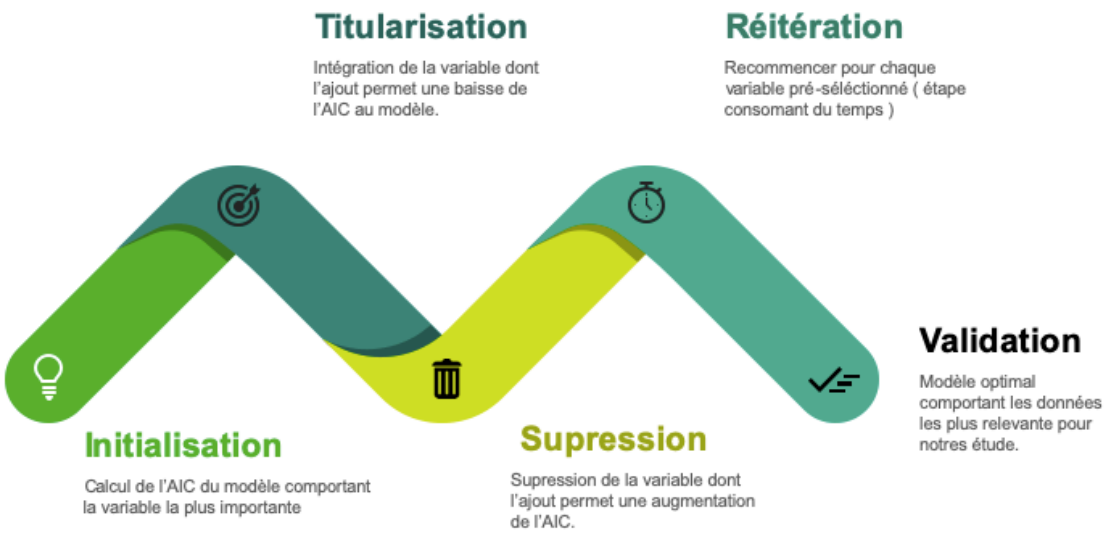


FIGURE 52 : METHODE DE SELECTION FORWARD

Nous commençons donc notre modèle simplifié en ajoutant une à une les variables qui étaient classés parmi les plus importantes :

Variable	AIC
Make	384 433
Make + Cardif_scheme_reference	381 451
Make + Cardif_scheme_reference + Purchase Price	381 451
Make + Cardif_scheme_reference + Purchase Price + Screen Ratio	378 158
Make + Cardif_scheme_reference + Purchase Price + Screen Ratio + Body Weight	376 127
Make + Cardif_scheme_reference + Purchase Price + Screen Ratio + Body Weight + Display protection	376 020
Make + Cardif_scheme_reference + Purchase Price + Screen Ratio + Body Weight + Display protection + Display type	376 012

Make + Cardif_scheme_reference + Purchase Price + Screen Ratio + Body Weight + Display protection + Display type + Display Resolution	375 972
Make + Cardif_scheme_reference + Purchase Price + Screen Ratio + Body Weight + Display protection + Display type + Display Resolution + Display Inches	375 923
Make + Cardif_scheme_reference + Purchase Price + Screen Ratio + Body Weight + Display protection + Display type + Display Resolution + Display Inches + Battery	375 940

TABLEAU 19 : METHODE DE SELECTION FORWARD APPLIQUEE AU PORTEFEUILLE

Les variables retenues pour le modèle simplifié sont donc :

Code Variable	Définition
Make	Marque du téléphone
Cardif_scheme_reference	Partenaire de vente du téléphone
Purchase_price	Prix d'achat du téléphone
Screen_body_ratio	Ratio taille écran sur taille téléphone
Body_weight_grammes	Poids du téléphone
Display_protection_cat	Catégorie de protection du téléphone
Display Resolution	Résolution de l'écran du téléphone
Display Inches	Taille du téléphone

TABLEAU 20 : VARIABLES SELECTIONNEES

Or ces variables représentent sans la batterie toutes celles utilisées lors du modèle précédent. Ainsi, nous décidons pour le GLM Lasso de garder toutes nos données internes et externes et par la même occasion de garder les résultats de la métrique variance de Poisson expliquée calculée auparavant égale à 7,13%.

Conclusion

Ce mémoire est la conclusion d'une année de travail au sein de BNP Paribas Cardif sur un modèle de *Pricing Analytics* pour l'assurance vol/casse de téléphone mobile. L'étude avait pour ambition de comprendre les facteurs impactant les fréquences de sinistres par marque et modèle de téléphone portable ainsi que d'améliorer la capacité de prédiction de la sinistralité de nos modèles de *Pricing* à partir des données disponibles (données internes/externes) tout en participant à la recette métier par l'Actuariat de librairies Python créées par le Datalab Cardif.

Deux grandes questions étaient soulevées : est-ce que l'apport des données externes améliorait significativement la modélisation de prédiction de fréquence de sinistralité ? Quelle méthode de modélisation faut-il utiliser pour obtenir le meilleur pouvoir de prédiction ?

Une première partie non négligeable se concentre sur l'exploitation de données externes récupérées. Il s'est avéré que l'apport des données sur les caractères physiques des téléphones n'était pas un axe d'amélioration significative des modèles. L'utilisation de la combinaison des données internes et externes en comparaison aux seules données internes fournissait des performances certes meilleures mais non significatives. Un gain de 13% par la métrique de somme des erreurs absolues a été constaté.

Cependant, cette étude répond à plusieurs questions. En particulier, elle nous permet de souligner la nécessité d'utiliser les données externes disponibles pour capter davantage d'informations sur les composants ayants un lien direct avec la casse de l'appareil.

Au regard de l'importante quantité de variables internes et celles récupérées et traitées d'une source extérieure, l'utilisation de méthodes d'apprentissage automatique telle que l'Extreme Gradient Tree Boosting s'est imposée comme indiscutable. Afin de garantir une forte capacité de prédiction tout en limitant le nombre de variables tarifantes, l'efficacité de l'algorithme de Gradient Tree Boosting dans sa sélection automatique de variables a également été remise en cause en réalisant une seconde sélection plus manuelle telles que l'étude des diagrammes de Shapley et une procédure de sélection Forward. Cette étape a permis de confirmer la qualité de la sélection automatique des variables réalisée par l'algorithme. Ainsi, le modèle XGBoost final dit réduit a été conçu à travers les variables sélectionnées.

Par la suite, les résultats d'une analyse comparative des performances d'un modèle linéaire généralisé log-Poissonnien avec le modèle d'Extreme Gradient Tree Boosting réduit a permis de montrer que le modèle XGBoost est plus approprié. En effet, en intégrant les variables externes sélectionnées qui constituent les caractéristiques physiques directement dans le modèle de XGBoost, le modèle produit s'est avéré à la fois plus performant et plus stable.

L'un des inconvénients observés associés aux méthodes de *Machine Learning* est le phénomène de sur-apprentissage. Le gain en prédiction de fréquence de sinistralité après ajout des variables externes n'était pas significatif, la puissance réelle de prédiction des modèles ML était équivalente à celle du GLM.

Par ailleurs, nous avons voulu tester le pouvoir de prédiction de notre modèle dans l'optique de la sortie d'un nouveau téléphone. Pour ce faire, nous avons utilisé la méthode de *Cross Validation* dite Leave One Out. Les résultats obtenus n'étaient pas assez performants et ne permettent pas de se passer d'une tarification à dire d'expert.

A travers les différentes méthodes énoncées précédemment, nous avons pu mieux comprendre les facteurs explicatifs du risque et participer à la recette métier par l'utilisation de différents packages. Finalement en considérant les résultats des études, parmi les données externes ajoutées, le ratio entre la surface de l'écran/la surface de l'appareil, le poids du téléphone et le type de protection de l'écran ont un fort impact sur la fréquence de sinistralité.

Mais la variable ayant le plus d'impact sur la prédiction des fréquences de sinistralité est la variable « opérateur téléphonique ». C'est en effet cette variable qui contient les différentes données comportementales sur l'assuré, telles que ses habitudes, *hobbies,* travail etc.

De ce résultat, nous pouvons en tirer le constat que la base de données externes constitue un axe à améliorer dans l'apport davantage d'informations personnelles et comportementales de l'assuré.

Bibliographie

[1] HASTIE T., Tibshirani R., Friedman J. (2017), «The Elements of Statistical Learning », 587-591.

[2] TIBSHIRANI R. (2007), « Regularization: Ridge Regression and the LASSO », 2-27.

[3] CORNILLON P.A., MATZNER-LOBER E. (2007), « Régression – Théorie et applications », 1-290.

[4] DURAND T. (2016), Évaluation et optimisation de la rentabilité d'un portefeuille automobile. Mémoire, Institut des Actuaires.

[5] https://scikit-learn.org/stable/modules/linear_model.html (Analyse des régression linéaires, pénalisation de Ridge et Lasso), site consulté le jeudi 12 novembre.

[6] Statistics Breiman, L (1996). Bagging. Machine Learning, 45 (1): 5 – 32.

[7] PLANCHET F., SERDECZNY. G. (2014), Modèles fréquence – coût : Quelles perspectives d'évolution ? Institut des Actuaires, Isfa.

[7] GAUVILLE R. (2017), Projection du ratio de solvabilité : des méthodes de machine learning pour contourner les contraintes opérationnelles de la méthode des SdS. Mémoire, Euria.

[9] COHEN W. (1995), Fast effective rule induction, 1-9.

[10] R. BELLINA. (2014), Méthodes d'apprentissage appliquées à la tarification non-vie. Mémoire Isfa.

[11] CHEN T. (2014), « Introduction to Boosted Trees», 1-41.

[12] CHEN T., GUESTRIN C. (2016), « XGBoost: A scalable Tree Boosting System », 1-13.

[13] FORTMANN-ROE S. (2012), « Understanding the Bias-Variance Tradeoff ».

[14] FORTMANN-ROE S. (2012), « Accurately Measuring Model Prediction Error ».

[15] PESNEAUD A. (2019), Création de zoniers en assurance habitation à l'aide de variables externes et de méthodes de Data Science. Mémoire Isup.

[16] https://www.math.univ-toulouse.fr/~besse/Wikistat/pdf/st-m-app-linSelect.pdf (Sélection de modèle en régression linéaire), site consulté le jeudi 12 novembre.

[17] TIMOFEEV R. (2004), Classification and Regression Trees. Master thesis, Humboldt University.

[18] HUN-KYUN B. (2010), « Classification and regression tree (CART) analysis for indicator bacterial concentration prediction for a Californian coastal area », 1-9.

[19] JEMLI H. (2019), Étude du risque de rachat de produits d'épargne italiens par des données agrégées et individuelles. Mémoire, Dauphine.

[20] CHARPENTIER A. (2010), « Statistique de l'assurance ». 3ème cycle. Université de Rennes 1 et Université de Montréal.

[21] Breiman L. (1994). Heuristics of instability in model selection, 2351-2382.

[22] http://wikistat.fr/pdf/st-m-app-agreg.pdf (Agrégation des modèles), site consulté le jeudi 12 novembre.

[23] Breiman, L (2001). Random forests. Machine Learning, 45 (1): 123 – 140.

[24] James, G., Witten, D., Hastie, T., and Tibshirani, R. (2013). An introduction to statistical learning. New York: Springer. 303-332

Liste des figures

Figure 1 : Nombre de téléphones mobiles (en millions) en Europe en 2020 4
Figure 2 : Prix moyen d'un téléphone par région dans le monde 4
Figure 3 : Marques vendues dans le monde en 2020 5
Figure 4 : Top 10 des modèles les plus vendus en Q1 2020 (en millions) 5
Figure 5 : Structure du Mémoire 9
Figure 6 : Number of mobile phones (in millions) in Europe in 2020 10
Figure 7 : Average price of a phone per region in the world 10
Figure 8 : Brand sold worldwide in 2020 11
Figure 9 : Top 10 best-selling models in Q1 2020 (in millions) 11
Figure 10 : Study structure 15
Figure 11 : Présence internationale de Cardif 23
Figure 12 : Chiffre d'affaires par activité 23
Figure 13: Couverture dommage accidentel 25
Figure 14 : Word cloud des causes de sinistre du portefeuille 26
Figure 15 : Cycle d'un sinistre entre un assuré et notre partenaire 31
Figure 16 : Exposition et fréquence de sinistralité par télécom 31
Figure 17 : Construction de la table interne finale 33
Figure 18 : Graphique des valeurs manquantes de la table externe 35
Figure 19 : Table de correspondances dite Mapping 37
Figure 20 : Construction de table de données finale 41
Figure 21 : Représentation du biais et de la variance d'un modèle 46
Figure 22 : Le sur-apprentissage 48
Figure 23 : Le compromis biais variance 48
Figure 24 : Séparation train-test de la table de données 49
Figure 25 : Méthode LOOCV 51
Figure 26 : Schéma simplifié d'un arbre binaire CART 53
Figure 27 : Schéma de fractionnement de l'algorithme CART 53
Figure 28 : Schéma de la méthode level-wise 57
Figure 29 : Schéma de la méthode Leaf-wise 57
Figure 30 : Lift curve des prédictions modélisées par XGB pour le modèle interne 67
Figure 31 : Lift curve des prédictions modélisées par LGBM pour le modèle interne 68
Figure 32 : Double lift curve comparant XGB et LGBM pour le modèle interne 69

Figure 33 : Lift curve des prédictions modélisées par XGB pour le modèle externe 70
Figure 34 : Lift curve des prédictions modélisées par LGBM pour le modèle externe 71
Figure 35 : Double lift curve comparant XGB et LGBM pour le modèle externe 72
Figure 36 : Double lift comparant le modèle interne et final par XGB 73
Figure 37 : Graphique de prédictions des top 10 modèles de téléphones 74
Figure 38 : Diagramme XGB des variables influentes dans le modèle 76
Figure 39 : Diagramme de Shapley représentant l'importance des variables dans le modèle 77
Figure 40 : Méthode de sélection de variables 79
Figure 41 : Lift curve des prédictions modélisées par XGB pour le modèle réduit 81
Figure 42 : Diagramme de Shapley représentant l'importance des variables dans le modèle réduit 82
Figure 43 : Méthode LOOCV appliqué à notre top 10 de modèles de téléphones 84
Figure 44 : Résultat de prédictions après méthode LOOCV 84
Figure 45 : Lift curve des prédictions modélisées par XGB pour le modèle avec ajout de bruit 87
Figure 46 : Diagramme de Shapley représentant l'importance des variables du modèle avec ajout de bruit 88
Figure 47 : Schéma des types de régression en fonction des données 91
Figure 48 : Lift curve des prédictions modélisées par GLM LASSO pour le modèle interne 98
Figure 49 : Double lift curve comparant les prédictions modélisées par GLM LASSO et XGB pour le modèle interne 99
Figure 50 : Lift curve des prédictions modélisées par GLM LASSO pour le modèle final 100
Figure 51 : Double lift curve comparant les prédictions modélisées par GLM LASSO et XGB pour le modèle final 101
Figure 52 : Méthode de sélection Forward 103

Liste des tables

Tableau 1 : Comparatif des mesures de prédictions par modèles et méthodes..... 6
Tableau 2 : Comparatif des mesures de prédictions par modèles et méthodes..... 7
Tableau 3 : Comparatif des mesures de prédictions par modèles 8
Tableau 4 : Comparative table of prediction scores by models and methods..... 12
Tableau 5 : Comparison of prediction scores by models and methods..... 13
Tableau 6 : Comparison of prediction scores by models 14
Tableau 7 : Caractéristiques de la base GSMARENA 34
Tableau 8 : Résultat de prédiction du modèle interne par les méthodes de machine learning..... 67
Tableau 9: Résultat de prédiction du modèle externe par les méthodes de machine learning..... 70
Tableau 10 : comparatif des mesure de prédictions par modèles et méthodes.... 72
Tableau 11 : Second comparatif des mesures de prédictions par méthodes et modèles..... 75
Tableau 12 : Sélection ascendante de variables 80
Tableau 13 : Variables sélectionnées 80
Tableau 14 : comparatif des mesures de prédictions par modèles 82
Tableau 15 : Standard déviation des variables quantitatives sélectionnées 86
Tableau 16 : Résultat de prédiction du modèle interne par GLM LASSO 97
Tableau 17 : Résultat de prédiction du modèle final par GLM 100
Tableau 18 : comparatif des mesures de prédictions par modèles et méthodes 102
Tableau 19 : Méthode de sélection Forward appliquée au portefeuille 104
Tableau 20 : Variables sélectionnées 104

Glossaire

IARD : Incendies, accidents et risques divers
MRH : Multirisques habitation
CART : Arbre de classification et de regression (Classification And Regression Tree)
XGB : Extreme Gradient Boosting
LGBM : Light Gradient Boosting
GLM : Modèles linéaires généralisés (Generalized linear model)
DT : Arbres décisionnels (Decision tree)
ML : Méthodes d'apprentissage (Machine Learning)
RF : Forêts aléatoires (Random Forest)
GS : Grille de recherche (Grid Search)
GBM : Gradient Boosting Machine
SVM : Machines à vecteurs de support (Support Vector Machine)
SL : Super apprenti (Super Learner)
AIC : critère d'information d'Akaike (Akaike information criterion)
BIC : critère d'information d'Akaike (Bayesian information criterion)
MSE : Erreur moyenne quadratique (Mean Square Error)
SAE : Somme des erreurs absolues (Sum of absolute errors)
IRLS : Iterative Reweighted Least Squares