Mémoire présenté devant l'ENSAE Paris
pour l'obtention du diplôme de la filière Actuariat
et l'admission à l'Institut des Actuaires
le 12/11/2020

Par : **Laurène MARTIN**

Titre : **Mortality Risk Modeling with Machine Learning**

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

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

Membres présents du jury de la filière

Entreprise : SCOR
Nom : Caroline Hillairet
Signature :

Membres présents du jury de l'Institut des Actuaires

Directeur du mémoire en entreprise :

Nom : Razvan Ionescu
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

Secrétariat:

Signature du candidat

Bibliothèque:

Ecole Nationale de la Statistique et de l'Administration Economique (ENSAE)
5, avenue Henry Le Chatelier - 91120 PALAISEAU, FRANCE

Résumé

L'avènement de l'intelligence artificielle en assurance ne se limite pas à l'automatisation de la souscription ou au développement de *chabots*. Cette dernière peut aussi être utilisée pour le cœur même du métier de l'assureur: l'amélioration de sa connaissance du risque. La consolidation des systèmes d'information permet aux assureurs et réassureurs d'accroitre leur efficacité en interne et d'analyser des bases de données plus riches.
L'industrie de l'assurance de personnes commercialise des contrats pour couvrir les accidents corporels, l'invalidité, la maladie ou encore le décès. Une bonne compréhension des risques biométriques est donc essentielle pour maximiser la prospérité d'une compagnie et proposer le tarif adéquat à chaque assuré. Pour cela, il est intéressant d'aller au-delà des méthodes statistiques traditionnelles et de faire appel aux modèles de *Machine Learnig*. Ces derniers ont souvent de bonnes performances par leur capacité à exploiter pleinement l'information contenue dans les bases de données volumineuses.

Si pour l'évaluation du risque mortalité, les modèles de durée ont prouvé leur efficacité, les modèles de *Machine Learnig* pourraient permettre d'aller plus loin. Ils n'ont pas été initialement conçus pour modéliser des durées, et ne peuvent pas être appliqués tels quels aux données de survie. Il est donc légitime de se demander dans quelle mesure les algorithmes de *Machine Learnig* pourraient prendre en compte les spécificités de l'étude de la survie.
Ce mémoire propose une étude théorique approfondie des adaptations de certains modèles de *Machine Learnig* pour l'évaluation du risque de mortalité. Les aspects métiers de l'assurance vie, des explications concrètes de sensibilité sur les produits, ainsi que des réflexions autour des avantages et inconvénients dans l'utilisation des modèles sur un marché d'assurance concurrentiel sont présentées. Les résultats de l'application numérique sont produits à partir de la base de mortalité NHANES.
L'ensemble des méthodes présentées est disponible dans une librairie développée en Python qui respecte les bonnes pratiques de développement: version, tests unitaires, documentation, tutoriels notebooks, intégration et déploiement continus.

Mots Clés: *Machine Learning*, Analyse de survie, Assurance-vie, Mortalité

Abstract

The use of artificial intelligence in insurance is not limited to automate underwriting or develop chatbots. It can also be used for the very heart of the insurer's business: improving the knowledge of risk. Indeed, consolidating information systems enables insurers and reinsurers to increase their internal efficiency and to analyze larger databases.
The life and health insurance industry sells contracts to cover individuals against bodily injury, disability, illness, or death. A good understanding of biometric risks is essential to maximize a company's prosperity and to offer the right price to each person insured. To achieve this, it is worth going beyond traditional statistical methods and using Machine Learning models. These have often good predictive performance due to their ability to fully exploit all the information contained in large databases.

While duration models have proven their effectiveness for assessing mortality risk, Machine Learning models could be used to go further. They have not initially been designed for survival modeling purposes and, contrary to other topics, they cannot be transposed as is to survival data. It is thus legitimate to wonder to what extent Machine Learning algorithms could be adapted and take into account the specificities of survival study.
This master thesis proposes an in-depth theoretical study of the adaptations of certain Machine Learning models for mortality risk assessment. Business aspects of life insurance are also addressed. Concrete explanations of product sensitivities as well as reflections on the advantages and disadvantages in the use of the models in a competitive insurance market are presented. The results of the numerical application are derived from the use of the American NHANES mortality database. All the methods presented in this master thesis are available in a library developed in Python that respects good development practices: versioning, unit testing, documentation, notebook tutorials, continuous integration, and deployment.

Keywords: Machine Learning, Survival Analysis, Life Insurance, Mortality

Note de Synthèse

Contexte

La modélisation du risque biométrique, qui caractérise l'ensemble des risques liés à la condition de vie humaine, est essentielle dans le secteur de l'assurance-vie. L'estimation impacte la tarification, le provisionnement et les ratios de solvabilité. Une connaissance approfondie du risque couvert est en ce sens essentielle pour garantir la prospérité d'un assureur. Pendant des décennies, les actuaires vie ont donc développé des méthodes statistiques pour estimer les risques biométriques et plus précisément la durée de vie. Entre-temps, les méthodes de *Machine Learning* se sont popularisées principalement car ces modèles reposent sur des hypothèses moins stricts et davantage sur les données.

Ce mémoire vise à évaluer les avantages de la modélisation de la mortalité à l'aide du *Machine Learning* sur un marché concurrentiel d'assurance-vie. Comme les algorithmes de *Machine Learning* n'ont pas été initialement conçus pour la modélisation de la durée, nous nous demanderons d'abord dans quelle mesure ils parviennent à traiter correctement les données de survie. Après une étude théorique approfondie des adaptations possibles pour l'évaluation des risques de mortalité, toutes les méthodes ont été implémentées au sein d'une bibliothèque Python, qui vise à répondre aux attentes opérationnelles d'utilisation. La bibliothèque Python a pour objectif de standardiser les modèles pour faciliter et automatiser l'étude de la mortalité d'un portefeuille. La tarification de produits d'assurance-vie repose en effet sur la modélisation de la mortalité. À partir des durées de vie estimées sur une base de données open-source, différentes stratégies de tarification ont été comparées sur un marché concurrentiel d'assurance-vie.

Modélisation de la survie

La première étape du processus de commercialisation d'une police d'assurance-vie est l'estimation du risque sous-jacent. Concrètement, les assureurs doivent estimer la durée de vie pour faire ressortir les facteurs de risque. Prédire des durées nécessite une technique de modélisation spécifique appelée *Analyse de survie*.
Lors de l'étude de la durée, les données sont sujettes à la *censure* : La plupart

du temps, une durée n'est que partiellement observée. De ce fait, les modèles statistiques ou de *Machine Learning* ne peuvent être transposés tels quels à des données de survie. Deux stratégies principales sont envisagées: l'application de modèles spécifiques aux données de survie: *modèles continus*, ou la modification de la structure des données pour permettre l'application de modèles standards: *les modèles discrets*. Dans ce mémoire, nous nous sommes concentrés sur les approches suivantes:

Modèles discrets	Modèles continus
Modèles linéaires généralisés	***Cox-ElasticNet***
modèles additifs généralisés	***Cox-XGBoost***
Gradient Boosting	***Arbre de survie***
Fôret aléatoire	***Fôret aléatoire de survie***

Table 1: Famille de modèles implémentés

Dans les deux cas, les aspects théoriques, les intuitions, les avantages et les inconvénients des modèles sont présentés et tous sont implémentés dans une bibliothèque Python.

Interprétation

Le secteur de l'assurance-vie est fortement réglementé, ce qui signifie que la commercialisation des produits doit respecter plusieurs contraintes. En effet, les assureurs doivent pouvoir justifier précisément le prix d'une police d'assurance et donc la mortalité estimée par un modèle. Cependant, certains de nos modèles, tels que le Random Forest ou Gradient Boosting, sont des modèles *"boîte noire"*, qui ne peuvent pas être directement interprétés.
Pour cette raison, en plus des modèles, trois méthodes d'interprétation complémentaires sont expliquées et ont été implémentées au sein de la bibliothèque Python: Importance des Variables, Dépendance partielle et SHAP (*SHapley Additive exPlanations*).

Présentation des données

Pour analyser l'impact sur un marché des différentes techniques de modélisation, il faut créer la demande pour les polices d'assurance-vie. Cette demande sera caractérisée par les observations de la base de données NHANES.
Il s'agit de la base de données open-source qui contient le plus d'informations disponibles sur les facteurs de risque et la mortalité. NHANES correspond à un

programme d'études conçu à l'origine pour évaluer la santé et le statut nutritionnel des adultes et des enfants aux États-Unis. Cette base de données est composée de 65.018 individus et de 106 variables, qui peuvent être classées en cinq catégories: démographie, alimentation, analyse médicale, visite médicale et questionnaire. Parmi les observations, 13,4% des individus sont morts au cours de la période de suivi de 2000 à 2014.
La base NHANES est pondérée de sorte à être représentative de la population américaine. Comme nous nous intéressons davantage à la méthodologie, nous utilisons uniquement les individus échantillonnés sans considération pour leur poids dans la population globale. Une réflexion sur la pondération afin de reproduire une population assurée permettrait cependant d'avoir des résultats plus consistants.

Estimation de la mortalité

L'estimation de la mortalité est nécessaire pour évaluer des primes d'assurance-vie. Pour un contrat d'assurance-vie donné, la variation de la prime pure entre deux assureurs dépend uniquement du choix de modélisation pour la mortalité. Ainsi, avant de simuler le résultat d'un assureur sur un marché concurrentiel, la mortalité de nos données doit être estimée à l'aide des différents modèles.
À cette fin, la bibliothèque Python, développée pendant ce mémoire, est utilisée pour estimer la mortalité de la base NHANES. Nous mettons également en évidence les principales étapes de la modélisation de la survie et de la détermination des facteurs de risque. Quelques aperçus sur le calibrage, la validation, les mesures de performance et la visualisation des modèles sont présentés.

Nous commençons d'abord par le prétraitement des données. La qualité des données est essentielle avant de commencer un projet. Nous reproduisons le processus d'une équipe de souscripteurs qui accepte et refuse les dossiers sur la base des informations médicales. Après ces considérations, nos modèles sont calibrés sur 29.870 individus et 27 facteurs de risque ont été pris en compte.
Sur la base de données NHANES, il semble que le modèle discret CatBoost soit le meilleur compromis en termes de performance prédictive, de temps de calcul et de calibration.

Toutefois, les données ont plus de valeur que le modèle lui-même. C'est-à-dire qu'en utilisant un autre jeu de données, une conclusion différente pourrait être tirée. À cette fin, la bibliothèque a été développée de manière à standardiser chaque

modèle, ce qui permet et facilite leur comparaison :

	Forces	Faiblesses
Regression Binomiale	• Pas de calibration requise • Interprétable • Possibilité d'ajout de régularisation	• Modélisation de structures linéaires • Problèmes de convergence
Regression Poisson	• Pas de calibration requise • Interprétable • Possibilité d'ajout de régularisation	• Modélisation de structures linéaires • Problèmes de convergence
GAM logistique	• Modélisation de l'interaction entre les variables et de structures non-linéaires • Interprétable • Possibilité d'ajout de régularisation	• Nécessité de spécifier les interactions • Problèmes de convergence
Forêt aléatoire	• Modélisation de l'interaction entre les variables et de structures non-linéaires • Parallélisable • Robustesse	• Sensible à l'overfitting • Sensible à la calibration des hyperparamètres • Difficulté de capter les effets de durée
LightGBM	• Modélisation de l'interaction entre les variables et de structures non-linéaires • Adapter aux grosses bases de données avec de nombreuses variables • Rapidité parmi les algorithmes de Gradient Boosting	• Sensible à la calibration des hyperparamètres • Pas interprétable
XGBoost	• Modélisation de l'interaction entre les variables et de structures non-linéaires • Robustesse parmi les algorithmes de Gradient Boosting	• Sensible à l'overfitting • Sensible à la calibration des hyperparamètres • Pas interprétable
Catboost	• Modélisation de l'interaction entre les variables et de structures non-linéaires • Supporte les variables catégorielles sans traitement à priori • Peu sensible à l'overfitting parmi les algorithmes de Gradient Boosting	• Sensible à la calibration des hyperparamètres • Pas interprétable

Table 2: Tableau récapitulatif des modèles discrets

	Forces	Faiblesses
Cox - Net	• Pas de calibration requise • Interprétable • Possibilité d'ajout de régularisation	• Modélisation de structures linéaires • Problèmes de convergence • Hypothèse de fonction de hasard proportionnelle
Cox - XGBoost	• Modélisation de l'interaction entre les variables et de structures non-linéaires • Robustesse • Bonne performance prédictive	• Sensible à l'overfitting • Sensible à la calibration des hyperparamètres • Pas interprétable
Arbre de Cox	• Temps de calcul rapide • Interprétable	• Hypothèse de fonction de hasard proportionnelle • Problèmes de convergence • Difficulté de capter les effets de durée
Arbre de survie	• Temps de calcul rapide • Ne repose sur aucune hypothèse • Interprétable	• Mesure d'hétérogénéité contestable • Difficulté de capter les effets de durée
Forêt aléatoire de survie	• Modélisation de l'interaction entre les variables et de structures non-linéaires • Parallélisable • Robustesse	• Sensible à l'overfitting • Sensible à la calibration des hyperparamètres • Difficulté de capter les effects de durée

Table 3: Tableau récapitulatif des modèles continus

Marché assurantiel

La mortalité des potentiels acheteurs sur le marché a été estimée à l'aide de différentes méthodes. Il devient alors possible, pour un produit d'assurance-vie, de type *assurance décès*, et sous diverses hypothèses économiques, de calculer une prime pure. La valeur de la prime pure de ce type de produits est en effet exprimée comme une fonction décroissante de la probabilité de survie à chaque période de temps.

On calcule enfin le ratio de sinistralité, qui est le rapport du montant global des sinistres sur le montant global des primes, afin de mettre en évidence le résultat d'un assureur en fonction de sa stratégie de tarification. Le tableau suivant montre

le ratio de sinistralité sur le marché total en fonction du modèle sous-jacent. Cette situation de monopole est principalement utilisée pour la comparaison avec le cas de concurrence. Les résultats sur le marché total ne permet pas une conclusion claire: Même si le loss ratio de certains modèles est légèrement plus proches de 100%, les effets ne sont pas significatifs à cause des erreurs d'estimation.

Models	*Loss Ratio*
Binomial Regression	101,20%
Poisson Regression	100,81%
Random Forest	97,50%
LightGBM	97,99%
XGBoost	96,74%
logistic GAM	95,92%
Catboost	99,09%
Cox	100,58%
Cox XGBoost	98,73%
Cox-Net	100,56%
Cox Tree	106,19%
Survival Tree	101,98%
Random Survival Forest	103,59%

Table 4: Loss Ratio sur le marché total

Nous disposons alors de toutes les informations nécessaires pour simuler un marché d'assurance-vie simplifié: Deux assureurs utilisant des stratégies de tarification différentes pour le même produit seront en concurrence. Cela permettra de mieux comprendre l'importance de la modélisation de la mortalité et de procéder à une comparaison concrète des modèles à des fins actuarielles. Le marché sera divisé entre les assureurs uniquement suivant les raisons économiques.

Une première expérience a mis en évidence que, toutes choses égales par ailleurs, utiliser un modèle de *Machine Learning* semble permettre de gagner des parts de marché et donc de battre un concurrent avec des méthodes de régression. L'assureur utilisant le *Machine Learning* parvient à obtenir un ratio de sinistralité proche de 100% tandis que l'assureur traditionnel réalise des pertes (Tableau 5).
La division du marché entre les deux assureurs est en effet bénéfique à celui qui dispose de la technologie la plus avancée car cet assureur semble pouvoir proposer des prix plus attractifs aux personnes moins risquées.

Assureur	***Loss Ratio***
Modèle de régression	143,9%
Modèle de Machine Learning	99,4%

Table 5: Comparaison des loss ratio

Une deuxième expérience a été menée pour évaluer la valeur de l'information pour deux assureurs utilisant la même méthode de modélisation. Les souscripteurs s'efforcent actuellement de mesurer l'avantage de l'inclusion de variables médicales. Cela empêche en effet les souscriptions en ligne que les clients pourraient faire eux-mêmes. En outre, l'obtention de ces informations exige des frais supplémentaires et peut décourager certaines personnes. Dans cette optique, nous avons décidé de comparer le résultat de deux assureurs utilisant le *Machine Learning* sur un marché concurrentiel si l'un d'entre eux décide de limiter sa tarification aux informations déclaratives.

Notre simulation a mis en évidence que les informations supplémentaires, malgré les frais de souscription, sont précieuses car elles permettent d'éviter l'antisélection.

Executive Summary

Context

Modeling biometric risk, which refers to all risks related to human life conditions, is essential in the life insurance industry. The estimation impacts pricing, reserving, and solvency assessments. In that sense, a deep knowledge of the risk covered is key to ensure the prosperity of an insurer. For decades, life actuaries have thus developed statistical methods to estimate biometric risks and more precisely the life duration. Meanwhile, the popularity of the Machine Learning methods is rising mainly as these models enable the use of more data with fewer constraints on assumption.

This master thesis aims at evaluating the benefit of modeling mortality with machine learning in a competitive life insurance market. As Machine Learning algorithms have not initially been designed for duration modeling purposes, we first wondered how they manage to handle survival data correctly. After an in-depth theoretical study of their possible adaptations for mortality risk assessments, all the methods have been implemented within a python library to meet operational usage expectations. The python library aims at standardizing several methods to facilitate and automate the study of the mortality of a portfolio. The pricing of life insurance products relies indeed on mortality modeling. Based on the estimated life duration of an open-source database, different pricing strategies were compared on a competitive life insurance market.

Survival Modeling

The first step of the marketing process of a life insurance policy is the estimation of the underlying risk. Concretely, insurers need to estimate life duration to bring out risk factors. Predicting time to event requires a specific modeling approach called *Survival Analysis*.
When studying duration, datas are subject to *censoring*: Most of the time survival duration is only partially observed. Due to this fact, statistical or machine learning models cannot be transposed as is to survival data. Two main strategies are considered: fitting specific models to raw survival data: *time-to-event models*, or

modifying the data structure to enable the application of standard models: *discrete models*. In this master thesis, we focused on the following approaches:

Discrete Models	Time-to-event
Generalized linear models	***Cox-ElasticNet***
Generalized Additive models	***Cox-XGBoost***
Gradient Boosting	***Survival tree***
Random Forest	***Random Survival Forest***

Table 6: Groups of implemented models

For both cases, the theoretical aspects, intuitions, advantages, and drawbacks of the models are presented and all have been implemented within a python library.

Interpretation

The life insurance industry is heavily regulated, which means that the commercialization of products should respect several constraints. Indeed, insurers must be able to justify precisely the price of an insurance policy and thus the mortality estimated by a model. However, some of our models, such as Random Forest or Gradient Boosting, are *"black box"* Machine Learning models, which cannot be directly interpreted.
For that purpose, besides the models, three complementary methods of interpretation are explained and have been implemented within the library: Variable Importance, Partial Dependence, and SHAP (*SHapley Additive exPlanations*).

Data presentation

To analyze the impact on a market of different mortality modeling approaches, we need demand for life insurance policies on the market. The demand will be characterized by the observations on the NHANES datasets.
NHANES is the open-source database with the most information available regarding risk factors and mortality. It is a program of studies originally designed to assess the health and nutritional status of adults and children in the United States. The dataset is composed of $65,018$ individuals, and 106 variables, which can be categorized into five classes: demography, dietary, laboratory, examination, and questionnaire. Among the observation, 13.4% of individuals were dead during the follow-up period from 2000 to 2014.
The NHANES database is weighted to be representative of the American population. Since we are more interested in the methodology, we use only sampled

individuals without consideration for their weight in the overall population. However, a reflection on the sample weights to reproduce an insured population would allow for more consistent results.

Mortality estimation

Modeling mortality is required to derive life insurance premiums. Given a life insurance product, the pure premium variation between two insurers will only rely on the mortality modeling choice. Thus, before simulating an insurer's result, the mortality of our dataset should be estimated with different models.
For that purpose, the python library, developed during this master thesis, is used to estimate the mortality of the NHANES dataset. We also highlight the main focus steps of survival modeling and determination of risk factors. Some insights into models calibration, validation, performance measurements, and visualization are presented.

We first start with the pre-processing of the data. The quality of the data is essential before starting a project. We replicate the process of an underwriting team that accepts and declines files based on medical information. After these considerations, our models were calibrated on $29,870$ individuals, and 27 risk factors were considered.
Based on the NHANES dataset, it seems that the discrete model CatBoost is the best agreement in terms of predictive performance, computation time, and calibration.

However, one may keep in mind that the data is more valuable than the model itself. That is to say that based on another dataset, one may derive a different conclusion. For that purpose, the library was developed in a way to standardize every model, which enables and facilitates their comparison:

	Strengths	Weaknesses
Binomial Regression	• No parameter calibration • Interpretability • Possibility to add regularization	• Model only linear patterns • May not converge
Poisson Regression	• No parameter calibration • Interpretability • Possibility to add regularization	• Model only linear patterns • May not converge
logistic GAM	• Model variable interactions and non linear patterns • Interpretability • Possibility to add regularization	• Need to specify the interactions • May not converge
Random Forest	• Model variable interactions and non linear patterns • May be parallalized • Robust	• Subject to overfitting • Sensitivity to hyperparameter • Difficulty to capture duration effect
LightGBM	• Model variable interaction and non linear pattern • Can adapt to large dataset with lots of features • Speeder gradient boosting algorithm	• Sensitivity to hyperparameters • No interpretability
XGBoost	• Model variable interactions and non linear patterns • More robust compared to other gradient boosting	• Subject to overfitting • Sensitive to hyperparameters • No interpretability
Catboost	• Model variable interactions and non linear patterns • Handle categorical variable without prior transformation • Less subject to overfitting compared to other gradient boosting	• Sensitivity to hyperparameters • No interpretability

Table 7: Discrete models summary table

	Strengths	Weaknesses
Cox - Net	• No parameter calibration • Interpretability • Regularization	• Model only linear patterns • May not converge • Rely on proportional hazard assumption
Cox - XGBoost	• Model variable interactions and non linear patterns • Robust • Good predictive performance	• Subject to overfitting • Sensitive to hyperparameters • No interpretability
Cox Tree	• Short computation time • Interpretability	• Rely on proportional hazard assumption • May not converge • Difficulty to capture duration effect
Survival Tree	• Short computation time • No underlying assumption • Interpretability	• Questionability of the heterogeneity measure • Difficulty to capture duration effect
Random Survival Forest	• Model variables interactions and non linear patterns • May be parallalized • Robust	• Subject to overfitting • Sensitivity to hyperparameter • Difficulty to capture duration effect

Table 8: Time-to-event models summary table

Insurance market

The mortality of all potential clients on the market have been estimated with different methods. It becomes then possible, given a life insurance product, such as a *death insurance policy*, and several economic hypotheses, to compute a pure premium based on each model estimation. The pure premium value of such products is indeed expressed as a decreasing function of the survival probability at every period of time.

We finally compute the loss ratio, which is the ratio of the global amount of the claims on the global amount of the premiums, to highlight the result of an insurer depending on the pricing strategy. The following table shows the loss ratio

depending on the underlying model. This monopoly situation is mainly used for comparison with the competitive case. The result on the global market does not lead to a clear conclusion: Even if the loss ratio of some models is slightly closer to 100%, there is no significant effect due to estimation errors.

Models	*Loss Ratio*
Binomial Regression	101.20%
Poisson Regression	100.81%
Random Forest	97.50%
LightGBM	97.99%
XGBoost	96.74%
logistic GAM	95.92%
Catboost	99.09%
Cox	100.58%
Cox XGBoost	98.73%
Cox-Net	100.56%
Cox Tree	106.19%
Survival Tree	101.98%
Random Survival Forest	103.59%

Table 9: Loss Ratio on the global market

We now dispose of all the information to simulate a simplified insurance market: Two insurers using different pricing strategies for the same product will compete. It will enable to give better insights into the importance of mortality modeling and to conduct a concrete comparison of the models for an actuarial purpose. The market will be split between insurers based only on economic reasons.

A first experiment highlighted that all other things being equal considering a Machine Learning model seems to allow to gain market shares and thus beat a contestant with regression methods. The insurer using a machine learning model manages to obtain a loss ratio close to 100% while the traditional one makes large losses (Table 10).
The division of the market between both insurers is indeed beneficial to the one with the most advanced technology as this insurer seems to be able to offer more attractive prices for less risky individuals.

Insurer	*Loss Ratio*
Regressions models	143.9%
Machine Learning models	99.4%

Table 10: Loss Ratio Comparison

A second experiment was conducted to assess information value for two insurers using the same modeling strategy. Underwriters are currently struggling to

measure the benefit of including different types of medical variables. It prevents indeed from a complete online underwriting that clients could make on their own. Besides, getting the information requires additional fees and may discourage some individuals. For that purpose, we decided to compare the impact on the competitive market between two insurers using Machine Learning if one decides to limit its underwriting to only declarative information.
Our simulation highlighted that despite the underwriting fees, additional information is valuable as it prevents for anti-selection.

Remerciements

Je tiens premièrement à remercier toutes les personnes qui ont contribué à la réalisation et au succès de mon stage, notamment Scor pour le maintien et l'adaptation de mon stage malgré le contexte sanitaire.

Je remercie en premier lieu Antoine Chancel, data scientist chez Scor Global Life, qui malgré la distance, m'a orientée pour mener à bien ce projet par ses conseils et le temps qu'il m'a accordé.

Je remercie ensuite Razvan Ionescu pour sa disponibilité et le partage de ses connaissances sur les modèles de durée, ainsi qu'Antoine Ly pour son expérience et sa connaissance du Machine Learning.

Enfin, je remercie l'ensemble des membres du service Knowledge et BRM pour le temps qu'ils m'ont consacré et leur bonne humeur. Leurs conseils ont permis de rendre mon stage plus enrichissant et motivant.

Contents

Introduction

The life insurance industry provides contracts that cover death or other health risks. As the premiums received by the insurer should be sufficient to cover future unknown claims, a deep knowledge of the risks covered is key to ensure the prosperity of the insurer. Besides, the higher the risk expertise of an insurer, the more accurate and fairer the risk premium is offered to clients.

For decades, life actuaries have developed statistical methods to estimate biometric risks and more precisely the life duration. Meanwhile, the popularity of the Machine Learning methods is rising mainly as these models enable the use of more data with fewer constraints on assumption. Indeed, Machine Learning algorithms allow capturing the information structure without relying on strong assumptions on the dependence or the distributions of variables, contrary to traditional statistical methods. Consequently, Machine Learning algorithms may be better to capture and model complex patterns.

Regarding previous points, we wonder to what extent Machine Learning algorithms manage to handle survival data correctly. As initially they have not been designed specifically for survival modeling purposes, we can wonder how such algorithms can be adapted to take into account survival analysis specificities.

This master thesis aims at evaluating the benefit of modeling mortality with Machine Learning in a competitive life insurance market. For that purpose, we will review and extend the traditional actuarial methodologies of survival study with Machine Learning models. A library in Python has been developed in parallel to adapt the existing open-source Machine Learning libraries to actuarial science needs. The library has been tested on a large survival database, recognized by the scientific community, named NHANES. It has been used to produce the results and illustrations presented in this master thesis.

Chapter 1

Study framework

Before considering the technical aspects, a good understanding of the life insurance industry and its evolution seems important. It will enable to focus on the biometric risk modeling stake for an insurer.

1.1 The life insurance industry

1.1.1 History

The Amicable Society for a Perpetual Assurance Office [32] founded in London in 1706 by William Talbot and Sir Thomas Allen was the first insurance company. The company offered the first life insurance scheme. It was working as follows. Each member between the ages of twelve to fifty-five paid a fixed annual payment, in exchange members earn shares, from one up to three shares depending on member age. At the end of the year a portion of the "amicable contribution", the dividends, was divided among the wives and children of deceased members. The sums paid were proportional to the shares owned by the deceased members. Amicable Society started with 2000 members.

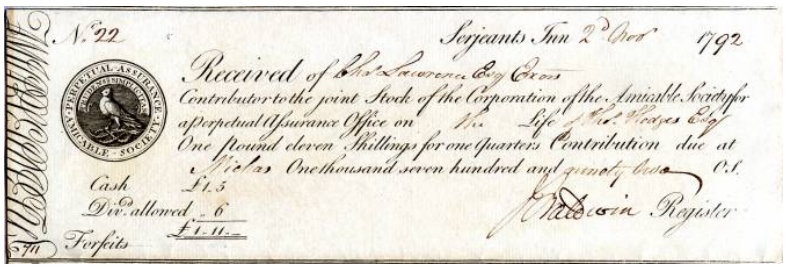


Figure 1.1: Receipt for payment of contribution (1792)

The life industry became more complex as the modern life insurance industry offers now all kinds of products covering death events or other hazards related to

the health condition of the insured. The main products are covering Mortality, Critical Illness, Medical Expenses, and Longevity (Pension). Besides, thanks to the technological improvement and data storage capacity, information considered for risk assessment increased a lot and is still increasing. For instance, nowadays life insurance applicants are asked to share, besides their age, part of their medical history, financial situation, and their profession. As the information available is increasing, new challenges for insurance companies and actuaries are arising, namely extracting and analyzing efficiently information from a large amount of data to better assess the risk.

1.1.2 Main departments and role in an Insurance company

The life insurance industry collects insurance premiums and invests in assets globally that can result in systemic risk for the economy. The recent regulations, such as Solvency II, force insurers, and reinsurers to increase their capital requirements and to increase their knowledge of the risks to reduce the chances of bankruptcy. In this spirit, the insurers' structure has been rethought. To ensure the economic prosperity of the company, four main departments described below work closely together.

Underwriting department role is to provide acceptance or rejection of an insurance application. To do so, underwriters assess the risk based on the information collected from the life insurance applicant. If the risk appears to be high but acceptable, an extra premium may be demanded. This department has a key role as it avoids the insurer to be anti-selected and insured the risk covered corresponds to premiums collected.
To grow its market share, an insurer endeavors to reduce the complexity of the underwriting process. Accessing information from a client is expensive. This cost includes the easiness to access to information. For example, asking for the age of a person is easier than collecting blood measurements. On the other hand, an additional accurate piece of information improves insurer knowledge of the risk.

The **Experience Analysis (EA)** department is in charge of monitoring the inforce portfolio. By comparing the observed claims with the expected ones, the team report about the health of the business. The team also provides insights into the assumptions that should be revised to better reflect the risk. More generally, the Experience Analysis team provides data and insights about the risk assumptions considered in pricing and valuation.

The **Research & Development** department is providing support to other teams for risk assessment. The team provides guidance and best practices to set assumptions for various products and risks. When data is not available internally to assess the risk, the R&D team often formulate recommendations based on external literature and data sources. In particular, R&D is responsible to estimate future trends and estimation of rare events (1-in-200 years scenarios).

The **Pricing** department works closely with the EA and R&D to set an adequate price for the insurance products. The pricing team relies on EA and R&D inputs on risk to forecast future claims. Considering other economic and financial assumptions the team determines the premiums need to meet the profitability targets. As the price competition between insurers or reinsurers is high, the pricing teams work closely with the underwriting teams to make sure that the underwritten business is profitable.

1.1.3 Context

Life actuaries working in the departments cited above use and adapt statistical methods to model the risk.
For many years, only a limited amount of information about the applicant was collected. For instance, only age, gender, and smoking status of an insured were available to assess the risk. Thus, simple models such as linear regressions or classifications were considered sufficient to grasp the risk.
The rapid development of technology since the 80s is a revolution. The miniaturization of computers leads to the development of connected objects that collect real-time data, such as the number of steps in a day, on distributed storage systems in the cloud. Thanks to the increase in computer power, the industry can use Machine Learning models to capture the information of the new larger datasets.

The life insurance industry is adopting these new technologies to continue offering the best services to its clients and robotize their internal processes. Also, it allows insurers to collect more information and built massive datasets. To fully take advantage of these large datasets, one could consider Machine Learning models that help to make decisions or give advanced recommendations for experts.
This master thesis is a review and an extension with Machine Learning techniques of the traditional actuarial methodologies for survival analysis.

1.2 Survival Analysis in life insurance

Survival Analysis is a branch of statistics for analyzing the duration of time until one or more events happen. Formally it is a collection of statistical techniques used to describe and quantify the time to a specific event such as death, disease incidence, termination of an insurance contract, recovery, or any designated event of interest that may happen to an individual. The time may be measured on a different scale such as years, months, or days from the beginning of the follow-up of an individual until the event occurs. The time variable is usually referred to as survival time because it gives the time that an individual has "survived" over some follow-up period.

Modeling biometric risk, i.e. modeling the duration until a claim, is essential in the life insurance industry as it impacts pricing, reserving, and solvency assessments. Insurers focus on modeling the duration before on three main events that are covered through different insurance products:

- Life spam, which is the duration before the death of the insured,
- Disability duration, which is the period an insured will remain disable i.e. duration before death or recovery,
- Autonomy (not disabled) duration, it is the duration before death or the loss of autonomy

For the computation of premiums and reserves, one may not limit the modeling to a binary classification whether the claim has occurred or not. Indeed, it is important to be able to compute the event occurrence probability at any time for the whole duration of the contract. In other words, we seek to predict the claim probability for a given period of time, which is the reason why it is important to master survival analysis theory.

One may also mention the underwriting field of a life company, that needs to classify insurance applicants from *bad* to *good* risk based on their risk profile. That means being able to order applicants based on their probability of claim occurrence. Besides, the development of more precise models contributes to being more inclusive, as we may derive a price even for a very risky individual. Thus it enables to sell products to clients, who would have been excluded from the portfolio in the past.

Chapter 2

Survival Analysis Theory

The first step of the marketing process of a life insurance policy is the estimation of the underlying risk. Concretely, insurers need to estimate life duration to bring out risk factors. Predicting time to event requires a specific modeling approach called *Survival Analysis*.
The main specificity of survival data is *censoring*. Indeed, most of the time survival duration is only partially observed. Because of it, different approaches must be considered to include this specificity. Two main modeling strategies exist to take censoring into account: fitting specific models to raw survival data or modifying the data structure to be able to apply standard models.
This chapter will introduce the survival analysis theory: the data challenges and the modeling strategies, on which Machine Learning modeling relies.

2.1 Censoring and truncation

Often, survival data contain a distinctive characteristic making it impossible to directly measure survival. While gathering the information, undesired events can occur during the observation period that pollutes the records and prevent to observe the full survival duration for all individuals. One may think of events such as Lapses, Hospital transfers, IT system failure during records, etc. Besides, the observation time is limited so the event of interest, such as death, may occur outside of the time window. These are called *Censoring* and *Truncation*.

2.1.1 Definitions

Censoring defines a situation in which the information is only partially known or observed, while Truncation corresponds to a situation when the information is totally unknown. There are two types for each:

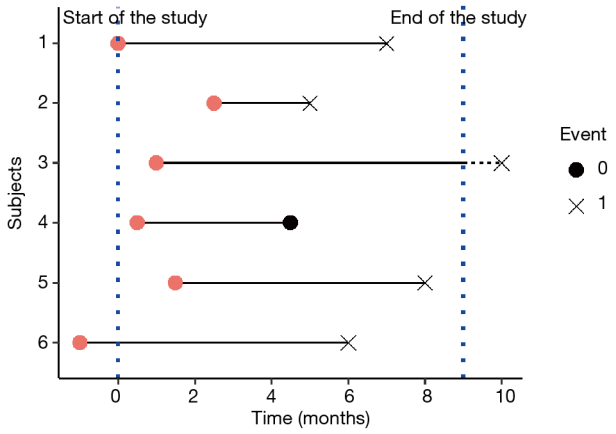


Figure 2.1: Illustration of censoring.

- **Right Censoring**: Right censoring refers to an event that occurs after the end of the observation period or to the loss of the trace of a subject due to other independent reasons. In Figure 2.1, subjects 3 and 4 are subject to right censoring. Even if the exact duration is not observed, right censoring still reveals partial information: the event of interest occurs after the observed time and thus the duration is at least as long as the censoring one.
  Let's consider the study of a life duration after purchasing a life insurance contract. If the insured ends her contract t years after the subscription, then for this observation, we can say only the insured survived at least t years. But her survival time, i.e. period until death occurred, is unknown.

- **Left Censoring** : Left censoring is quite the opposite of right censoring. It occurs when the trigger point of the duration measure is before the observation period as it is the case for the subject 6 in Figure 2.1. Once again, we only know that the real duration is bigger than the one observed.
  One may face left censoring when a study includes individuals who have already a contract at the beginning of the observation time. The purchasing time is thus unknown.

- **Right Truncation** Right truncation corresponds to individuals who are completely excluded from a study because the starting event that includes them in the study happens after the end of the observation period.
  Considering the same example as before, right truncation is observed when an individual buys insurance after the observation period. It is thus totally excluded.

- **Left Truncation** Left truncation is the opposite. An individual is excluded because the event of interest occurs before the beginning of the observation

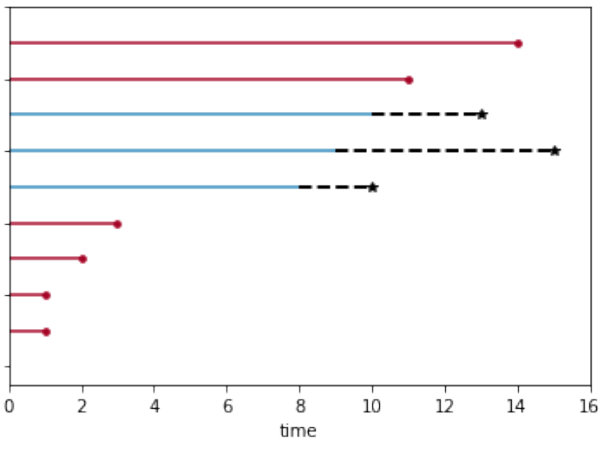
period.

All individuals who are insured and dead before the observation period are left truncated.

Most of the time, for studying biometric risks in life insurance, left censoring and right truncation does not occur.
Left truncation is more likely to happen when modeling life risks, but in the following, we will only deal with the right censoring, which is the most common scenario. It is worth noting that it is possible and quite easy to consider left truncation by enhancing a bit of the modeling.

2.1.2 Impact on duration estimation

When dealing with survival data, a common mistake could be to simply ignore any censor or truncation effects. This approach leads to an underestimation of the interest event probability. Another common mistake is to restrict the study to only observations that are complete by removing any censored or truncated records. Here as well, estimation is extremely biased. Let consider the ten following individuals to understand the intuition behind the importance of considering censoring:



	1	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*
Duration observed	10	9	1	3	0	2	8	1	14	11
Real duration	13	15	1	3	0	2	10	1	14	11
Status	C	C	D	D	C	D	C	D	D	D

Figure 2.2: Illustration to highlight the impact of censor

Based on these data, the mean survival duration period is 5.9 when considering censored time as death time and 5.3 when removing censored observation while the real one is 7. In both cases, in this example, life expectancy is underestimated

when the partial information coming from the censored individuals is ignored. In other words, the factor of risk will be overestimated.

2.2 Survival Data

Before introducing the different models, the main concepts of survival theory should be introduced.

The whole theory relies on two random variables: T the time to event and C the censoring time. They are assumed to be independent. The censoring time is a random variable modeling the observation period of an individual, that is to say, the time between the start of the observation of the individual and the time of withdrawal or loss of tracking. While the time to event is a random variable modeling the observation period between the start of observation and the studied event occurrence, for instance, the death.

However, in practice, the information available in the data sets is the stopping time of observation (because of death or censor) and an indicator of whether the observation is censored. That is to say:

$$\begin{cases} Y = \min(T, C) \\ \delta = 1_{\{T \leq C\}} \end{cases}$$

Considering p explanatory variables given at $t = 0$ and n subjects, a survival data table can be built as in Table 2.1. The columns are divided into three categories: the first p columns representing the risk factors, Y being the end of the follow-up period observed, and δ an indicator of death.

From this example, we can deduce that the first individual, who has the $x_{1.}$ characteristics, dies at time t_1 whereas the last individual is censored at time t_n.

Obs	X_1	...	X_p	Y	δ
1	x_{11}	...	x_{1p}	t_1	*1*
.	.	.	.	.	.
.	.	.	.	.	.
.	.	.	.	.	.
n	x_{n1}	...	x_{np}	t_n	*0*

Table 2.1: Example survival data table

2.3 Quantity of interest

The theory focuses on two functions as the quantity of interest to estimate: the **survival function** S and the **hazard function** h. Having an estimation of one of them allows to fully model the survival of an individual.

The **survival function** S represents the probability that the time to the event is not earlier than a specific time t:

$$S(t) = Pr(T \geqslant t) \tag{2.1}$$

The survival function is decreasing from 1 to 0. The meaning of a probability equals to 1 at the starting time is that 100% of the observed subjects are alive when the study starts: none of the events of interest have occurred. From this quantity, we can define the *cumulative death distribution function* $F(t) = 1 - S(t)$ and the *density function* $f(t) = \frac{dF(t)}{dt} = \frac{-dS(t)}{dt}$ for continuous cases and for discrete cases $f(t) = \frac{[F(t+\Delta t)-F(t)]}{\Delta t}$. The relationship among these functions is shown in Figure 2.3.

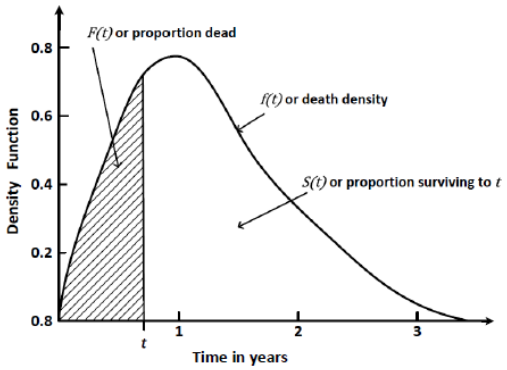


Figure 2.3: Relationship among $f(t)$, $F(t)$ and $S(t)$ (source [35])

The second quantity of interest is the **hazard function** h. It indicates the rate of event at time t, given that no event occurred before. Formally, the hazard rate function is defined as:

$$\begin{aligned} h(t) &= \lim_{\Delta t \to 0} \frac{Pr(t \leqslant T \leqslant t + \Delta t | T \geqslant t)}{\Delta t} \\ &= \lim_{\Delta t \to 0} \frac{F(t + \Delta t) - F(t)}{\Delta t S(t)} \\ &= -\frac{\mathrm{d} \log S(t)}{\mathrm{d}\, t} \end{aligned} \tag{2.2}$$

From this equation we can easily derive that

$$S(t) = exp(-\int_0^t h(s)\mathrm{d}s) = exp(-H(t))$$

where $H(t) = -\int_0^t h(s)\mathrm{d}s$ is called *the cumulative hazard function.*
Using the same notation as before, we can define a likelihood function taking into account censoring:

$$L = \prod_i P(T = t_i)^{\delta_i} P(T > t_i)^{1-\delta_i} = \prod_i h(t_i)^{\delta_i} S(t_i) \tag{2.3}$$

The intuition of the function comes from the contribution to the likelihood function between a censored and a full-observed individual:

- If an individual dies at time t_i, its contribution to the likelihood function is indeed the density that can be written as $S(t_i)h(t_i)$.
- If the individual is still alive at t_i, all we know is that the lifetime exceeds t_i, which means that the contribution to the likelihood function is $S(t_i)$.

2.4 Non-parametric survival model

2.4.1 Kaplan-Meier estimator

When we have no censor observations in the data, the empirical survival function is estimated by:

$$\hat{S}(t) = \frac{1}{n}\sum_{i=1}^{n} 1_{t_i \leq t} \tag{2.4}$$

This estimation is no longer viable in presence of censor as we do not observe the death time t_i but the end of observation time y_i. Thus Kaplan and Meier [21] extended the non-parametric estimation to censored data.
Kaplan-Meier estimator is the most widely used because of its simplicity to compute. It is implemented in many of survival libraries and packages of statistical and mathematical software. Besides, this estimator relies on no assumption and can thus easily be used as a reference model or to test hypothesis.
The main idea behind this estimator is that surviving after a given time t means being alive just before t and do not die at the given time t. Consequently, with $t_0 < t_1 < t$ we get :

$$
\begin{aligned}
S(t) &= \mathbb{P}(T > t) \\
&= \mathbb{P}(T > t_1 \ , \ T > t) \\
&= \mathbb{P}(T > t \mid T > t_1) \times \mathbb{P}(T > t_1) \\
&= \mathbb{P}(T > t \mid T > t_1) \times \mathbb{P}(T > t_1 \mid T > t_0) \times \mathbb{P}(T > t_0)
\end{aligned}
$$

In the end, by considering all the distinct times $t_i, (i = 1, ..., n)$ where an event occurred ranked by increasing order (whatever it is a death or censorship) we get: $S(t_j) = \mathbb{P}(T > t_j) = \prod_{i=1}^{j} \mathbb{P}(T > t_i \mid T > t_{i+1})$, with $t_0 = 0$.

Considering the following:

d_j the number of deaths that occurred in t_j

N_j the number of individuals alive just before t_j

The probability $q_j = \mathbb{P}(T \leq t_j \mid T > t_{j-1})$ of dying in the time interval $]t_{j-1}, t_j]$ knowing the individual was alive in t_{j-1} can be assessed by : $\hat{q}_j = \frac{d_j}{N_j}$
Let δ_i be the censorship indicator of each observation; the Kaplan-Meier estimator is then defined as:

$$\hat{S}(t) = \prod_{t_i \leq t} (1 - \frac{d_i}{N_i})^{\delta_i} \tag{2.5}$$

We finally obtain a step function for the survival function where the jumps are observed at the empirical observed death times.

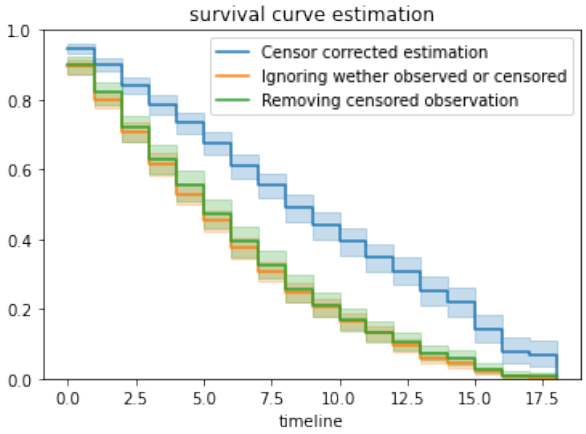


Figure 2.4: Impact of ignoring censoring in life duration study

As introduced before, ignoring censoring leads to an underestimation of the life duration. The Figure 2.4 highlights this underestimation. Three 'Kaplan-Meier' survival curves are plotted on different datasets: the real one, the one relying only on the fully observed individuals, and the one, for which censor and dead individuals are not distinguished. As the two last curves are below the real one, it means that at each time the survival probability is lower and thus that the risks have been overestimated.

Kaplan-Meier estimation is effective to get the survival curve of the global population. However, the precision of the estimation relies on the number of observations. If we want to take into account individuals' characteristics, we need to recompute the estimator for each chosen subset, which reduces the number of observations and thus the accuracy.
On the business side, it is indeed important to have a good prediction among different subgroups rather than on the global level. The insurer portfolio may indeed have an over-representation of some individuals compared to the population used to build the model, knowing that the insured population has lower mortality compared to the global population.

2.4.2 Nelson-Aalen estimator

Instead of estimating the survival function, another method has been developed by Aalen [1] and Nelson [27] to estimate the cumulative hazard function. Using the previous notation, it is defined as:

$$\hat{H}(t) = \sum_{t_i \leq t} \frac{d_i}{N_i} \delta_i$$

To get an estimator of the survival function, one only has to plug-in the cumulative hazard estimator into the formula $S(t) = e^{-H(t)}$

$$\hat{S}(t) = e^{-\hat{H}(t)} = \prod_{t_i \leq t} (e^{-\frac{d_i}{N_i}})^{\delta_i} \approx \prod_{t_i \leq t} (1 - \frac{d_i}{N_i})^{\delta_i}$$

If the number of deaths is small compared to the number of people at risks at each time, the Nelson-Aalen plug-in survival function can be approximated by the Kaplan-Meier estimator. The two estimators are thus numerically close, but they have different properties, which implies different confidence intervals or median times (see [30] for details on estimator properties).

2.5 Cox proportional hazard model

Cox's model [10] allows to take the effect of covariates and to measure their impacts through the estimation of the hazard function. It becomes then possible to rank people's risk according to their characteristics. This model may be considered as a regression model for survival data, which is a particular case of the proportional hazard models. Such models are expressed as a multiplicative effect of the covariates on the hazard function through the expression:

$$h(t \mid X) = h_0(t) \times g(\beta' X) \tag{2.6}$$

X the vector of covariates which must be time-independent

g a positive function

β the parameter of interest

h_0 the baseline hazard function for individuals with X=0.

Given two individuals, A and B having the covariates X_A and X_B respectively, the ratio of their hazard functions is assumed to be unchanged over time. That is the reason why such models are said to be proportional hazard models.
For the particular case of the Cox model, the function g is an exponential function so that the Cox model is defined by $h(t \mid X) = h_0(t) \times exp(\beta' X)$. The main interest of the model is the possibility to rank people on their risk level without computing the survival function. The relative risk is introduced to this end :

$$RR = \frac{h(t|X_A)}{h(t|X_B)} = exp(\beta(X_A - X_B)) \tag{2.7}$$

The estimation can be divided into two steps. Depending on the purpose of the study one may stop at the first one.

Estimation of the risk parameter β

We compute the estimator $\hat{\beta}$ by maximizing the partial likelihood function defined by Cox (cf Annexe A) :

$$L(\beta) = \prod_{i=1}^{m} \frac{e^{X'_{j_{(i)}}\beta}}{\sum_{j \in R_i} e^{X'_j \beta}} \tag{2.8}$$

m the total of uncensored individuals

$j_{(i)}$ the individuals who died at time $t_{(i)}$

$t_{(i)}, ..., t_{(m)}$the ordered time of observed death events

R_i the risk set, which is a set of indices of the subjects that still alive just before $t_{(i)} : R_i = \{j : t_j \leq t_{(i)}\}$

When one is only interested in comparing the survival curve to classify individuals according to their survival probabilities, only the estimation of the risk parameter β is needed. The baseline hazard $h_0(t)$ does not only have any effect on the relative risk.

Estimation of the baseline function

If one needs the survival function for every individuals, the hazard baseline $h_0(t)$ is needed in addition of the β parameters. The survival function can be computed as follows:

$$\hat{S}(t) = exp(-H_0(t)exp(X\hat{\beta})) = S_0(t)^{exp(t\hat{\beta})} \tag{2.9}$$

where $H_0(t)$ is the cumulative baseline hazard function, and $S_0(t) = exp(-H_0(t))$ represents the baseline survival function. The Breslow's estimator is the most widely used method to estimate $\hat{S}_0(t) = exp(-\hat{H}_0(t))$ where

$$\hat{H}_0(t) = \sum_{t_i<t} \hat{h}_0(t_i) \text{ with } \hat{h}_0(t_i) = \frac{1}{\sum_{j \in R_i} e^{X_j\beta}} \text{ if } t_i \text{ is a time event, 0 otherwise}$$

2.6 Exposures : an actuarial approach

Another method, widely used in actuarial science, specifically to build mortality tables, consists to discretize the data into small time intervals. The discretization enables to apply traditional methodologies to predict mortality, as it removes the lack of information due to the censor thanks to the exposure to risk.

Withdrawal from the study of the censored subjects introduces bias if we compute traditional estimators. The mortality rate, q_j, within a time interval $[\tau_j, \tau_j + 1]$, denoted interval j, can no longer be estimated with the ratio of the deaths, d_j, on the number of alive subjects at the beginning of the interval, l_j. The quantity $\frac{d_j}{l_j}$ is indeed an inaccurate estimation as deaths that occur after withdrawal will not be known. Therefore, withdrawing life is only exposed to the risk of death.
To compensate for the withdrawal, the number of alive subjects, l_j, is replaced by the number of subjects exposed to risk. Depending on the hypothesis made on mortality, several types of exposure can be considered.
Three exposures are considered by actuaries: *Distributed exposure*, *Initial exposure* and *Central exposure*. However, we will only focus on the last two as the

distributed exposure method relying on uniform distribution of deaths is not currently a widely-used one (see [4] for details).

2.6.1 Initial Exposure and Balducci hypothesis

We denote **initial exposure** the quantity, EI_j, which represents the global amount of time each life was exposed to the risk of death during the interval j. As the exposure is based on the lives at the start of the interval the exposure can be referred to as initial.
EI_j is the aggregation of the following individual exposure, ei_j:

- Alive at the start and the end of the interval are assigned 1
- Deaths during the time interval are assigned 1
- Censored are assigned the fraction of the interval they were observed

Formally, if we denote respectively $c_{i,j}$ and $t_{i,j}$ the censoring and death time of the individual i in interval j, w_j the number of withdrawals and l_j the number of alive subjects, the **initial exposure** is expressed as:

$$\begin{aligned} EI_j &= \sum_{i}^{l_j} 1_{\{t_{i,j}>1\}} \times 1_{\{c_{i,j}>1\}} + 1_{\{t_{i,j}<1\}} + c_{i,j} 1_{\{c_{i,j}<1\}} \\ &= \sum_{i}^{l_j} 1 - 1_{\{c_{i,j}<1\}} + c_{i,j} 1_{\{c_{i,j}<1\}} \\ &= l_j - w_j + \sum_{i=1}^{w_j} c_{i,j} \end{aligned}$$

To understand the idea behind this quantity, let define the two following notations for the rate of mortality,

- $q_j = P(T \leq \tau_j + 1 | T > \tau_j)$ in interval j
- ${}_{c_{i,j}}q_j = P(T \leq \tau_j + c_{i,j} | T > \tau_j)$ for the one in the interval $[\tau_j, c_{i,j}]$

The number of deaths can be expressed as the sum of the deaths observed within the interval and the deaths expected for the censored subjects. Formally:

$$d_j = (l_j - w_j) q_j + \sum_{i=1}^{w_j} {}_{c_{i,j}}q_j = l_j q_j - \sum_{i=1}^{w_j} {}_{1-c_{i,j}}q_{j+c_{i,j}} \tag{2.10}$$

The **Balducci hypothesis** supposes that mortality rates decrease over the inter-

val and are defined as:

$$\begin{aligned} {}_{1-c_{i,j}}q_{j+c_{i,j}} &= P(T_i \leq \tau_j + 1 | T_i > \tau_j + c_{i,j}) \\ &= (1 - c_{i,j})P(T_i \leq \tau_j + 1 | T_i > \tau_j) = (1 - c_{i,j})q_j \end{aligned}$$

Injecting it in the previous equation gives:

$$d_j = l_j q_j - q_j \sum_i^M (1 - c_{i,j})$$

Solving the formula for q_j:

$$\hat{q}_j = \frac{d_j}{l_j - \sum_{i=1}^{w_j}(1 - c_{i,j})} = \frac{d_j}{l_j - w_j + \sum_{i=1}^{w_j} c_{i,j}} = \frac{d_j}{EI_j}$$

We finally get the rate of mortality estimator corrected for censoring with the previous definition of **initial exposure** as expected. This approach relies on *Balducci assumtion*, which generally does not fit for mortality as mortality rates increase with time. However withdrawals are usually small compared to the population, which allows to ignore these errors.

2.6.2 Central Exposure and Constant hazard function

Depending on the mortality observed within a dataset, one may prefer to use another assumption: the constant hazard function over each time interval. In this case, another exposure should be used.
The **central exposure**, EC_j is the amount of time individuals are observed within the interval. The difference with the **initial exposure** is that only individuals who survived the whole time interval are assigned 1.

The *constant hazard function* assumption implies that the hazard is constant over each time interval.
For $e \in [0, 1]$, we denote h_j the hazard rate over the interval $[\tau_j, \tau_j + 1]$:

$$h(\tau_j + e) = h_j \tag{2.11}$$

As long as we consider time interval small enough this hypothesis is acceptable.
When h_j is known for each j, the survival function is easy to compute :

$$S(\tau_j + e) = \exp(-\int_0^{\tau_j+e} h(s)ds) = \exp(-\sum_{s=1}^{j-1} h_s + eh_j) \tag{2.12}$$

The goal is then to estimate each h_j.

Let $ec_{i,j}$ be the **individual central exposure**, it corresponds to the amount of time one is observed within an interval. In addition, $\delta_{i,j}$ is a death indicator in $[\tau_j, \tau_{j+1}]$ (1 if death is observed, 0 otherwise). The likelihood can then be written as:

$$L = \prod_i S(\tau_j + ec_{i,j})h(\tau_j + ec_{i,j})^{\delta_{i,j}} \tag{2.13}$$

Using the constant hazard function assumption and considering the logarithm of the likelihood we get:

$$\log(L) = \sum_i [ec_{i,j}h_j + \delta_{i,j}\log(h_j) - \sum_{s=1}^{j-1} h_s] \tag{2.14}$$

The maximum likelihood estimator $\hat{h}_j$, so that $\frac{d}{dh_j}log(L) = 0$, is then the ratio of the number of death observed within the interval divided by the exposure :

$$\hat{h}_j = \frac{\sum_i \delta_{i,j}}{\sum_i ec_{i,j}} = \frac{d_j}{EC_j} \tag{2.15}$$

By definition, we can write $\hat{q}_j = 1 - exp(-\hat{h}_j)$. As initial exposure, the central exposure is interesting as it can be expressed through a closed formula. However, it relies as well on a death distribution, which is generally not verified in practice.

2.6.3 Modeling using exposure

The main advantage of discretization is that it allows considering classical modeling approaches, by predicting the number of deaths for each time interval. In practice, we will model the random variable d_j describing the number of deaths using the exposure as weights or offset. Exposures are easy to compute and take into account censoring, however, this approach can generate a high number of lines in the dataset as we need to create *pseudo data table*, making the computation slow.

Pseudo data tables

Models can be applied to pseudo data tables, which are an alternative data structure in survival analysis modeling. Often, in the actuarial field, the information of a portfolio is directly presented in pseudo data tables. If not, we can easily transform traditional survival data tables into pseudo data tables.
In practice, we have to generate for each individual as many rows as time intervals and for each of them to compute individual exposure. The size of the intervals is fixed in advance: month, quarter, year, etc. The size choice depends on how granular and accurate the output is needed.

To illustrate, let's suppose that $t_1 \in [\tau_{j(1)}, \tau_{j(1)} + 1]$ and $t_2 \in [\tau_{j(2)}, \tau_{j(2)} + 1]$, $j(i)$ being the number of intervals considered for the individual i. The number of intervals varies between individuals as the observation period of each individual varies. The last time interval includes the time of death or censoring, which means that δ is always equal to 0 except for the last time interval. Regarding the exposure, it is always equal to 1 except for the last time interval, where it represents the amount of time observed in any case when we compute the **central exposure** or only in case of censoring when we compute the **initial exposure**.
Finally, we build the dataset for every individual in the study as illustrated for two observations in Table 2.2. It is worth noticing that this approach allows considering covariates that vary with the time interval. It is the case for time-varying covariates such as smoking habits. It is one advantage of this method in opposition to previous approaches considering only information at the start of observation.
The time interval j is added to the feature variables. That is to say, that the same individual is seen as two different ones depending on the time interval j considered.

Obs	X_1	...	X_p	$\boldsymbol{I}$	$\boldsymbol{ec}$	$\boldsymbol{ei}$	δ
O_{11}	x_{111}	...	x_{1p1}	*1*	*1*	*1*	*0*
O_{12}	x_{112}	...	x_{1p2}	*2*	*1*	*1*	*0*
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.
$O_{1j(1)}$	$x_{11j(1)}$	...	$x_{1pj(1)}$	*j(1)*	$t_1 - \tau_{j(1)}$	*1*	*1*
O_{21}	x_{211}	...	x_{2p1}	*1*	*1*	*1*	*0*
O_{22}	x_{212}	...	x_{2p2}	*2*	*1*	*1*	*0*
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.
$O_{2j(2)}$	$x_{21j(2)}$	...	$x_{2pj(2)}$	*j(2)*	$t_2 - \tau_{j(2)}$	$t_2 - \tau_{j(2)}$	*0*

Table 2.2: Survival pseudo data table

Chapter 3

Performance computation and model validation

Due to the very nature of the survival data, the classic metrics such as the Area Under the ROC curve (AUC) or the Mean Squared Error (MSE) might not be adapted to measure the model performances. Also, censorship prevents us to directly apply the usual metrics.
Statisticians proposed several metrics to deal with survival data along with estimators when the observed survivorship is censored. In this section, we describe some of these metrics.

3.1 Standardized Mortality Ratio

One of the most common and widely used metrics is the Standardized Mortality Ratio (SMR). Also known as Actual to Expected ratio in the actuarial field, the SMR can be used to measure the prediction accuracy of a model. It is defined as the ratio between the number of observed deaths divided by the number of predicted deaths by the model.

$$\text{SMR} = \frac{\sum_i \delta_i}{\sum_i pred_i} \tag{3.1}$$

with $\delta_i = 1$ if we observed deaths of individual i and $\delta_i = 0$ otherwise. The total expected number of deaths is obtained by summing $pred_i$ defined as the model predicted probability to observe the death of individual i.

A SMR close to 1 indicates that the model fits well the observations. Different values indicate that the model may have a bias. Also, a SMR lower than 1 shows that the model overestimates the mortality, while a SMR higher than 1 indicates that the model is underestimating the mortality.

The censorship must be taken into account when estimating the death probabilities $pred_i$. Indeed, it is unlikely to observe the death of an individual that has left the study after only a few days, while for an individual observed several years the probability should be higher.
Using the law of large numbers, the probability to observe a death within the study period can be approximated by the sum of the observed death divided by the number of observations. But we must add the number of non-observed deaths because of censorship. Let's start by the following equation:

$$\mathrm{P}(T \leq \tau) \simeq \frac{1}{N}\sum_{i=1}^{N}\delta_i + \frac{1}{N}\sum_{i=1}^{N}(1-\delta_i)\mathrm{P}(T \leq \tau | \, T > t_i) \tag{3.2}$$

with t_i the observation period for person i, τ the maximum observation period and T the random variable modelling the survival time. After few simplifications we end up with the following equality:

$$\sum_i \delta_i \simeq \sum_i (1 - S(\tau)) - (1-\delta_i)\left(1 - \frac{S(\tau)}{S(t_i)}\right) \tag{3.3}$$

We recall the survival function definition: $S(t) = \mathrm{P}(T > t)$. Considering $\hat{S}(t | \, X_i)$ the survival probability predicted by the model, we can define $pred_i$ as follows:

$$pred_i = 1 - \hat{S}(\tau | \, X_i) - (1-\delta_i)\left(1 - \frac{\hat{S}(\tau | \, X_i)}{\hat{S}(t_i | \, X_i)}\right) \tag{3.4}$$

3.2 Concordance Index

The Concordance Index, also called C-Index, has been introduced by Harrell et al. [18]. Mainly used to measure the relevancy of the bio-marker for the survival estimation, this metric is also used to assess the predictive performance of survival models.

This metric allows us to measure the model's ability to order individuals according to their survivorship. This metric is very relevant when the main goal of the model is to classify individuals according to their mortality risk, i.e. order individuals from the ones with the lowest mortality to the ones with the highest mortality. This metric measures the classification ability of the model, but it does not measure fit quality. Thus the potential bias of a model would not be detected by this metric, which is thus complementary of the SMR.

The C-Index is defined as a the conditional probability: the model survival predic-

tions (M_i, M_j) of the two individuals i and j are ordered in the same way as their respective survival observations (T_i, T_j).

$$\mathrm{C-Index} = \mathrm{P}\left(M_i < M_j \,|\, T_i < T_j\right) \tag{3.5}$$

As model survival prediction M_i, one could consider the predicted life span $M_i = \mathrm{E}[T \,|\, X_i]$ or the survival up to the end of the study period probability $M_i = \hat{S}(\tau \,|\, X_i)$. Note that, the classical definition is $\mathrm{C-Index} = \mathrm{P}\left(M_i > M_j \,|\, T_i < T_j\right)$ as the the model score M_i is considered. As the higher the score the higher mortality, the score and surival observation of the pairs must be ordered in opposite orders. However, in this study we found more convenient to consider the expected survival period.

A C-Index close to 1 indicates the good performance of the model while a C-Index close to 0.5 indicates poor performance.

The C-Index can be estimated only on the pairs of observations (i, j) that are comparable. Indeed, due to censorship, some pairs are not comparable. In Figure 3.1 we provide an illustration of this issue. In this example, in pair $(2, 3)$ both survival observations are censored, therefore we are not able to tell whom of the two individuals has the highest survival period. Similarly, in pair $(1, 2)$ we cannot tell who has the survived longer, as for individual 2 we are losing track after year 5.

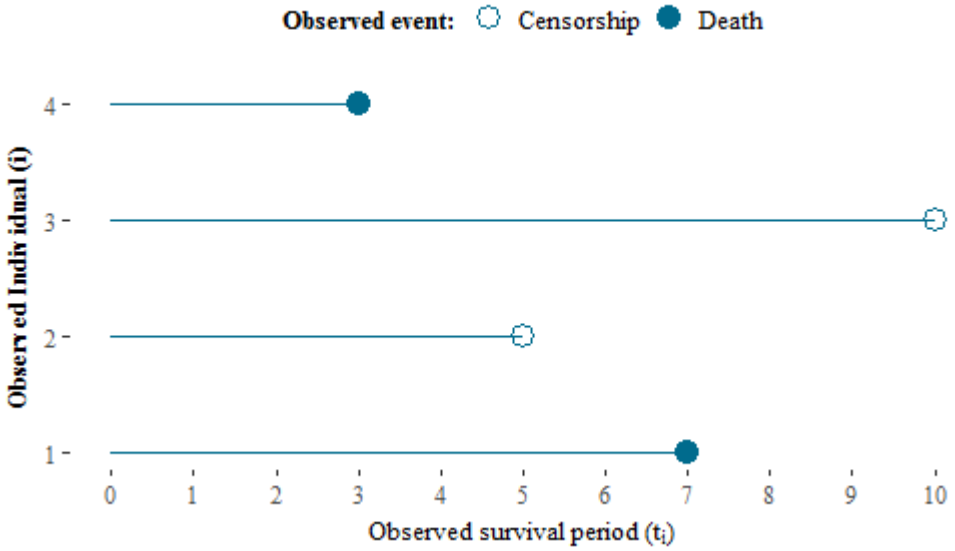


Figure 3.1: Censorship illustration, only pairs $(1,3), (1,4), (2,4)$ and $(3,4)$ are comparable.

Let Ω denotes the ensemble of comparable pairs (i, j) where $T_i < T_j$, then we can

estimate the C-Index as follows:

$$\text{C} - \text{Index} = \frac{1}{\text{Card}(\Omega)} \sum_{(i,j)\in\Omega} \mathbb{1}_{\{M_i < M_j\}} \tag{3.6}$$

However, this estimator depends on the study-specific censoring distribution. While this should have limited impact when using the C-Index compare model performances in a study, this prevents an accurate comparison of the C-Index from one study to another. In order to have a better estimation of the C-Index, Uno et al. [34] proposed an estimator based on the ICPW[1] approach. They proposed a free of the censoring distribution estimator as follows:

$$\text{C} - \text{Index} = \frac{\sum_i \sum_j \Delta_j G(t_j)^{-2} \mathbb{1}_{\{t_i < t_j\}} \mathbb{1}_{\{M_i < M_j\}}}{\sum_i \sum_j \Delta_j G(t_j)^{-2} \mathbb{1}_{\{t_i < t_j\}}} \tag{3.7}$$

where $G(t)$ denotes the probability of not having a censorship up to time t, and $\Delta_j = 1$ if no censorship, 0 otherwise.

3.3 Brier Score

Initially, the Brier Score has been introduced by Brier [8] to measure the accuracy of meteorological forecasts. Then, Graf et al. [17] proposed to use this metric in the bio-statistics field for assessing survival model performance. As the interpretation of itself is quite difficult, it is mainly used for model comparison.

The Brier Score, denoted BS, is defined as the average of squared difference between the survival probabilities and the survival observation a a given time t.

$$\text{BS}(t) = \frac{1}{N} \sum_i \left(\mathbb{1}_{\{T_i > t\}} - \hat{S}(t|\, X_i) \right)^2 \tag{3.8}$$

with $\hat{S}(t|\, X_i)$ the survival probability predicted by the model.

Because the censorship the BS cannot be estimated with the previous formula. In deed, if a censorship occurred before the fixed time t we cannot if the individual has survived longer than t. As for the C-Index, the authors considered an ICPW approach and they proposed the following estimator:

$$\text{BS}(t) = \frac{1}{N} \sum_i \frac{\hat{S}(t|\, X_i)^2}{G(t_i)} \mathbb{1}_{\{t_i \le t\,;\, d\delta_i = 1\}} + \frac{\left(1 - \hat{S}(t|\, X_i)\right)^2}{G(t)} \mathbb{1}_{\{t_i > t\}} \tag{3.9}$$

[1] Inverse Censorship Probability Weighting

where $G(t)$ is the probability of not observing a censorship up to time t.

Alike the Mean Squared Errors, the lower the BS the better. Usually, for model comparison the Brier Skill Score, BSS, is considered. It is defined as the reduction of the BS compared to the BS obtained on a reference model.

$$\text{BSS}(t) = 1 - \frac{\text{BS}(t)}{\text{BS}_{ref}(t)} \tag{3.10}$$

One needs to specify the time t to compute the Brier Score. Depending on the purposes a specific time t can be more relevant, for instance, we are studying the survival up to 5 years. Alternatively, the time-independent metric Integrated Brier Score, IBS, could be considered. It is defined as the average Brier Score:

$$\text{BSS} = \frac{1}{\tau}\int_0^\tau \text{BS}(t)\text{d}t \tag{3.11}$$

with τ the study period.

3.4 Exposure weighted AUC

When the time of observation is cut into intervals, the problem becomes a binary classification weighted by the exposure. In this case, the weighted AUC is a good measure of the performance of the model. This metric aims to evaluate the ability of the binary classification between dead and alive where the integration of the *initial exposure* as weight implies giving a bigger importance to the observed individuals rather to the censored ones. The importance given to a mistake on a censored subject increases with the observation time as the information increases as well. It is indeed worse to classify a censored individual as dead if he was observed 90% of the interval compared to one observed 10% of it because the first one is less likely to die in the resting time rather than the second one.

Let $I_d = \{i : \delta_i = 1\}$ and $I_a = \{i : \delta_i = 0\}$ be respectively the sets of dead and alive observations. Considering a threshold function f as follows:

$$f_\tau(\hat{\delta}) = \begin{cases} 1 & \text{if } \hat{\delta} \geq \tau \\ 0 & \text{if } \hat{\delta} < \tau \end{cases} \tag{3.12}$$

We then define the two following quantities:

Weighted true positive rate:

$$TPR(\tau) = \frac{1}{\sum_{i \in I_d} ei_i} \sum_{i \in I_d} 1_{\{f_\tau(\hat{\delta}_i) \neq 0\}} \times ei_i \tag{3.13}$$

Weighted false positive rate:

$$FPR(\tau) = \frac{1}{\sum_{i \in I_a} ei_i} \sum_{i \in I_a} 1_{\{f_\tau(\hat{\delta}_i) = 1\}} \times ei_i \tag{3.14}$$

A weighted ROC curve is drawn by plotting $FPR(\tau)$ and $TPR(\tau)$ for all thresholds $\tau \in R$.

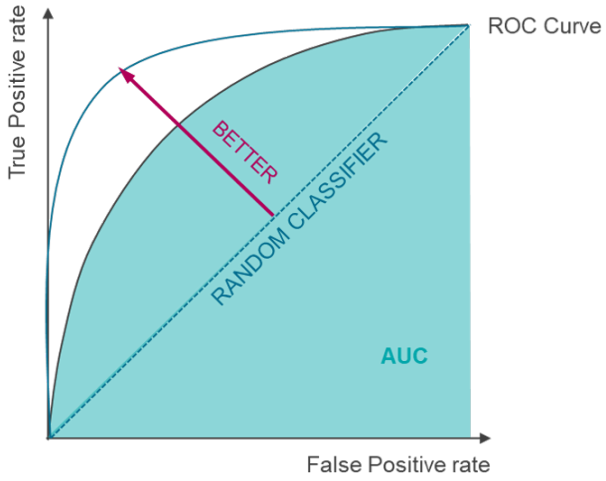


Figure 3.2: ROC Curve and AUC illustration

The weighted AUC is the value of the area under this ROC curve. The AUC ranges from 0.5 to 1 as the C-index, where 1 means that the model is perfect and 0.5 means that the prediction is equivalent to a random classification.

The interpretation is quite equivalent to the C-index metric in terms of model quality. However, thanks to the exposures every observation, even the censored ones, may be included to compute the AUC.

Chapter 4

Machine Learning to model mortality

Many of the current Machine Learning models have been adapted to survival analysis problems. In this chapter, the theory and the intuition behind the models that have been implemented in the Python library will be given. It is essential to review and understand the theory to make coherent implementation choices. Besides, there is almost no literature about the application of Machine Learning models to discrete data. A deep study of the theory is thus necessary to justify their correct adaptation to predict durations, which is required to derive insurance policy prices.

As mentioned before, we considered two approaches to model survival analysis: the models built on the survival dataset and the ones built on the dataset obtained after discretization. In the Python implementation, a main mother class *Model Discrete* and *Model Continuous* has been created for each approach, to gather the common functions of all models (cf Apendix B). These functions are evaluation metrics computation or prediction. Each model, defined in a specific class, inherits finally from the corresponding mother class. As the different implemented models rely on different existing Python packages, such as *statsmodel*, *lifelines* or *scikit-learn*, having a class for each enables to deal with all specific constraints. Besides through the different classes, a homogenization step is included, which contributes to simplify the library use. Each model can be fitted, evaluated, or can predict mortality thanks to the following pattern:

```
import scor_survival.models
#Training
model.train(X, event, exposition)
#Evaluation
model.auc(X, event, exposition)
model.ci(X, event, exposition)
#Predicting
model.predict_proba(X)
model.predict_surv(X)
```

4.1 Time to event framework

4.1.1 Cox-Model adaptation

Several Machine Learning methods have been adapted to Cox's Proportional-Hazard models such as Trees, Neural-Networks, Generalized Additive Models, etc. In this section, we present Elastic Net and Gradient Boosting Machine adaptation, as they are the most widely used in practice.

Cox-ElasticNet

Tibshirani et al. [33] proposed to apply the Elastic Net regularization to Cox proportional hazard model. In the Elastic Net approach, we add a penalization during the parameter estimation process. The goal is to put aside the less relevant features by penalizing the models with a high number of parameters. Decreasing the number of features allows to diminish the signal noise and consequently increase model accuracy.

This approach is very useful in high dimensions, i.e. when the number of features is close to the number of observations. This might occur during some epidemiological studies where a significant amount of information is available for each patient, but a limited number of patients are observed.

In practice, this approach is often used to quickly identify variables that have the biggest explanatory power and to put aside the non-relevant ones.

In this approach the model parameters β are estimated by optimizing the following:

$$\hat{\beta} = \underset{\beta}{\operatorname{argmax}} \left[\log(L(\beta)) - \lambda \left(\alpha \left\| \beta \right\|_1 + (1-\alpha)\frac{1}{2} \left\| \beta \right\|_2^2 \right) \right] \tag{4.1}$$

with L the likelihood of the Cox model (cf equation 2.8) and hyper-parameters λ and α.

The penalization intensity is controlled thanks to the hyper-parameter λ. If λ equals 0 then we are performing a classical Cox regression. The higher the value of λ the higher the penalization, the lower the number of non-null parameters.

When the hyper-parameter α equals 0 it is called LASSO[1] regression, when α equals 1 it is named a Ridge regression. Hyper-parameter α is in $[0, 1]$, it balances between LASSO and Ridge regression.

This approach has the same issues as the classical in Cox model. Namely, it relies heavily on the proportional force of mortality assumption that might not be verified. Besides, non-linear effects and unspecified interactions will not be captured

[1] LASSO stands for Least Absolute Shrinkage and Selection Operator

by this model.

Cox - Gradient Boosting Machine

To integrate non-linear effects within the Cox framework, a gradient boosting adaptation may be considered.
Gradient boosting consists in building a complex model thanks to the aggregation of several simple models called weak learners (Friedman [14]&[15]). The weak learners are all the same base learners though out the process, but they are successively trained on the residual errors made by the predecessor. Thus, each model relies on previous steps constructed models.
Gradient Boosting Machine is a mix between gradient boosting and gradient descent, which is an optimisation process to minimize a loss function. The adaptation of the GBM to a Cox's proportional hazard model [29] is possible by choosing the opposite of Cox partial likelihood as loss function :

$$LL(\beta) = -\sum_{i=1}^{m}[X'_{j_{(i)}}\beta - log(\sum_{j \in R_i} e^{X'_{j_{(i)}}\beta})] \tag{4.2}$$

Generally speaking, the algorithm is presented as the process below:

Initialisation : $F_0(x) = argmin_\beta LL(\beta)$

For $m = 1$ to M *(number of weak learners):*

- Computation of the pseudo-residuals: $r_m = -\frac{dL(F_{m-1}(X))}{F_{m-1}(X)}$
- Fitting a new weak learner on pseudo-residuals: $f_m(X) = r_m$
- Finding the best γ_m by solving $\gamma_m = argmin_\gamma L(F_{m-1}(X) + \gamma \times f_m(X))$
- Update the new model: $F_m = F_{m-1} + v \times \gamma_m f_m$

Thus, at each iteration, until the stopping condition is satisfied, we try to reduce the global error by fitting each specificity of the residuals. A learning rate, γ_m is introduced to control how much we adjust the weights of our base learner. This parameter may be constant and chosen at the beginning of the process or optimized at each step. A large value may reduce computing time but may cause divergence, a small one ensures convergence and getting an optimum but make learning time more consuming. The shrinkage parameter v, a scalar between 0 and 1, allows to regularize the model and ensure the convergence of the model.
The real advantage of gradient boosting is that it can adapt to any weak learners.

Most of the time trees are chosen. Cox Gradient Boosting Machine is a way of building classical regression trees by taking into account censoring within the loss function and assuming the proportional hazard hypothesis. In this case, trees are constructed consecutively, and the gradient shows the best path so that each tree is constructed on the previous one in such a way as it leads to the biggest error reduction.

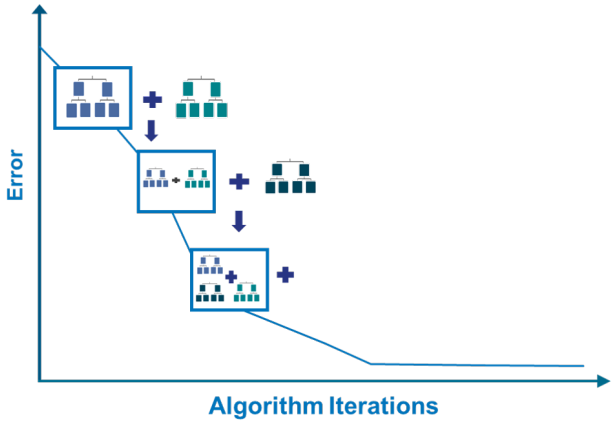


Figure 4.1: Gradient Tree Boosting

4.1.2 Survival Tree

Another method to build specific trees for survival analysis have been developed. Compared to Cox-Gradient Boosting, it enables to create predictor trees, which may be directly interpreted. The real advantage of trees is its simplicity compared to other Machine Learning techniques, which contributes to short computation time.
Survival trees have been explained by Bou-Hamad et al. [5] and Le Blanc and Crowley [23]. It is the direct adaption of decision trees to survival analysis. Traditional decision trees are also called CART (i.e. classification and regression tree), which is the fundamental algorithm of the tree-based methods family. The CART algorithm was developed by Breiman [7] and makes the use of trees popular to solve regression and multi-class classification problems.

Like CART, survival trees are binary trees grown by a recursive splitting of tree nodes. Starting from the root node composed of all the data, the observations are partitioned into two daughter nodes according to a predetermined criterion. Using a recursive process each node is split again into two nodes until reaching the *stopping criterion*. The best split for a node is found by searching over all possible variables and values, the one that maximizes survival difference.

The difference between *CART* and *Survival trees* relies on the **splitting criterion** used to build the tree. When dealing with survival data, the criterion must explicitly involve survival time and censoring information. Either it aims at maximizing the between-node heterogeneity or at minimizing the within-node homogeneity.

Log-rank criterion

The most widely used criterion is the maximization of the log-rank statistic (cf Annexe C) between the two sub-samples of the nodes, which contributes to creating splits that distinguish the most the mortality. As it is impossible to measure the similarity of the mortality within a group, the idea behind is that by sequentially creating splits with distinct mortality, we assume to obtain homogeneous groups at the end as the dissimilar cases become separated at each node.
Hyper-parameters should be introduced to optimize the number of splits: a minimum occurrence of events within a leaf or a lower threshold of the log-rank statistic to make a split. The intuition behind these *stopping criteria* is to ensure the quality of the split. The first one forces the splitting criterion to be computed on enough data to make sure that the log-rank statistic is consistent. The lower bound for the second one comes from the reject region bound of the underlying log-rank test, which means a node should not be split if the mortality is not statistically different with respect to any variable.

The main advantage of this method is that it does not rely on major assumptions to build the tree, even if the statistic considered to measure the difference in moralities between groups is questionable. Indeed, the log-rank test performance may be poor in some situations.
Once the tree is built, the model assumes that individuals within a leaf have the same common survival curve and thus a global survival curve is computed based on the individual within each final leaves. In open-source packages, the Nelson-Aalen estimator is used to compute the cumulative hazard function, from which we can deduce the survival curve or the expected lifetime duration. Experimental studies have shown that using the Kaplan-Meier estimator to directly estimate the survival curve gives similar results.

Thanks to the binary nature of survival trees, individuals with characteristics x_i fall into a unique leaf f composed of observations (x_i, δ_i) with $i \in \mathscr{I}_f$. The prediction of his cumulative hazard function is the estimator for x_i's terminal node:

$$\hat{H}(t|x_i) = \hat{H}_f(t) = \sum_{\substack{t_i<t \\ i\in I_f}} \frac{d_i}{N_i}\delta_i \tag{4.3}$$

Some other criteria have also been studied such as C-index maximization [31] or deviance minimization within one node.

Deviance criterion

The deviance minimization is based on a likelihood estimation relying on the proportional hazard function to partitions the observation. Under this hypothesis, the hazard function within a leaf f composed of observations (x_i, δ_i) with $i \in \mathscr{I}_f$, is expressed as follows:

$$h_f(t) = h_0(t) \times \theta_f$$

Using the formula 2.3, the likelihood can thus be rewritten :

$$L = \prod_f \prod_{i\in I_f} h_f(t_i)^{\delta_i} S_f(t_i) = \prod_f \prod_{i\in I_f} h_f(t_i)^{\delta_i} e^{-H_f(t_i)} = \prod_f \prod_{i\in I_f} (h_0(t)\theta_f)^{\delta_i} e^{-H_0(t_i)\theta_f} \tag{4.4}$$

Where $H_0(t)$ and $h_0(t)$ are respectively the baseline cumulative hazard function and the baseline hazard function, and θ_f is the parameter to estimate by likelihood maximisation. When H_0 is known, we can get the maximum likelihood estimator:

$$\hat{\theta_f} = \frac{\sum_{i\in I_f} \delta_i}{\sum_{i\in I_f} H_0(t_i)}$$

In practice, the cumulative hazard function is unknown and we plug in the breslow estimator

$$\hat{H}_0(t) = \sum_{i:t_i\leq t} \frac{\delta_i}{\sum_f \sum_{i:t_i\geq t; i\in I_f} \theta_f}$$

The deviance is finally defined as:

$$R(f) = 2[L_f(saturated) - L_f(\hat{\theta_f})] \tag{4.5}$$

where $L_f(saturated)$ is the log-likelihood for the saturated model that allows one parameter for each observation and $L_f(\hat{\theta_f})$ is the maximal log-likelihood.

The algorithm to build the tree adopts the principle of the CART algorithm: it will split the observation and covariate space into regions that maximize the reduction of the deviance realized by the split by testing all possible splits for each

of the covariates. In this approach a *stopping criterion* regarding the minimum size of a node is also considered since the likelihood estimation converges when it relies on a large amount of data.

Simulation experiments have shown that the performance is similar to the log-rank statistic. However, this method is not assumption-free and may not be applied to all data sets.
According to the trees built with a C-index maximization, the results are quite similar to the ones obtained with trees based on the log-rank statistic and are also assumption-free but the first one requires much more computation time. Thus, trees using the log-rank criterion should be privileged, which have been shown in several experimental studies.

4.1.3 Random Survival Forest

Random survival forest extends the random forest [6] method to right-censored survival data.

Random Forest

Random forest is an ensemble method inspired by the *bagging* of decision trees. Bagging, which means *Bootstrap Aggregating*, is an ensemble learning method that enables to create more robust predictor thanks to the aggregation of several ones trained on different subsets.
Bagging process consists in generating B random samples with replacement to train B trees $\hat{f}^b$ on these subsamples. Finally the prediction of a new input X is defined as:

$$\hat{f}_{Bagging}(X) = \frac{1}{B}\sum_{i=1}^{B}\hat{f}^b(X)$$

Random forest differs from simple bagging of trees as randomization is not only applied to drawn samples but also to select features. At certain nodes, rather than considering all the variables, a random subset of the attributes is selected to compute the splitting criterion. The introduction of randomization enables to reduce the correlation among the trees and to improve the predictive performance. The training process is illustrated in Figure 4.2.

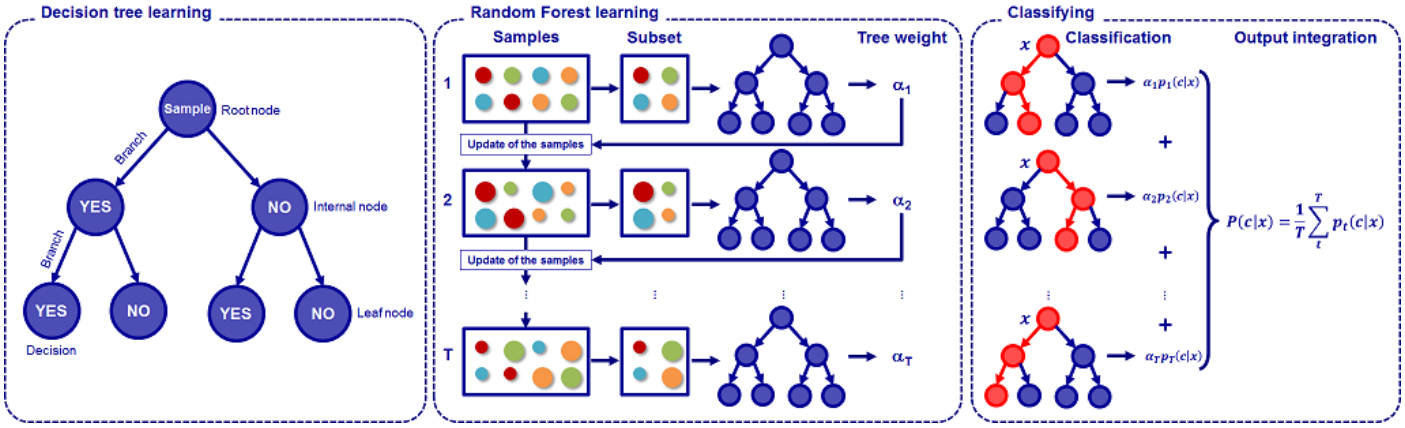


Figure 4.2: Random forest process illustration (source [26])

Random survival forest is an ensemble tree method developed by Ishwara et al. [20] that follows the same process but considers survival tree instead of traditional decision trees. The algorithm is processed as below:

- Draw B bootstrap sample from the original data that excludes on average 37% of the data, called out-of-bag data (OOB data)
- Grow a survival tree for each bootstrap sample, by selecting at each node p candidate variables. The split is chosen among the candidate variable that maximizes the survival difference between leaves.
- Calculate the cumulative hazard function for each tree, $\hat{H}^b(t|x_i)$ and average it over all the trees to obtain the ensemble cumulative hazard function:

$$\hat{H}(t|x_i) = \frac{1}{B}\sum_{b=1}^{B}\hat{H}^b(t|x_i)$$

The interpretation of the result may be questionable as we average several hazard functions to get the predicted one. However as $H(t) = \int_0^t h(s)\mathrm{d}s$ is already a sum of functions, averaging it returns still a sum : $\hat{H}(t) = \frac{1}{B}\sum_{b=1}^{B}\int_0^t \hat{h}^b(s|x_i)\mathrm{d}s = \int_0^t[\frac{\sum_{b=1}^{B}\hat{h}^b(s|x_i)}{B}]\mathrm{d}s$ and the prediction makes sense.

The main advantage is that forests can model non-linear effects without any prior transformation of the data and contrary to boosting, in bagging each tree is built independently and the process can thus be parallelized.

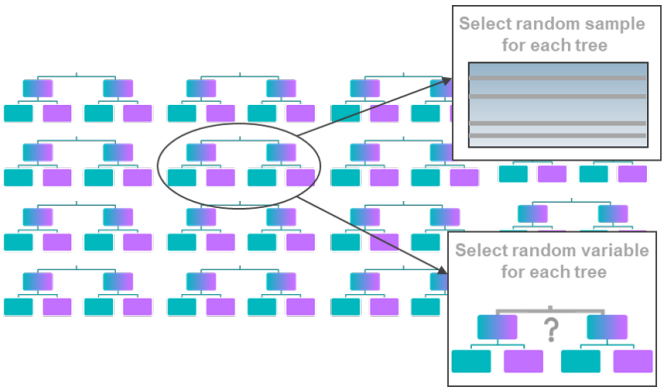


Figure 4.3: Bagging illustration

4.2 Discrete time modeling

We will now introduce several models that rely on the pseudo data table and give insights into the use of exposures within the modeling. These kinds of models are preferred within the actuarial field, as they may be applied to aggregated population data. Besides the conversion is only possible from a survival data table into pseudo data tables, which means that only discrete-time modeling is always possible.

As sometimes clients provide directly pseudo data tables with only one type of exposure available, it was essential to include in the Python library models based on the central and initial exposure.

4.2.1 Poisson regression

Poisson regression model assumes that the total number of deaths within the time interval j follows a Poisson distribution and is mainly based on the **central exposure**. That is to say:

$$d_j|X \sim \mathscr{P}(EC_j h_j(X)) \tag{4.6}$$

The idea behind the parameter used in the Poisson distribution comes from the constant hazard function hypothesis, as under this hypothesis $\hat{h}_j = \frac{d_j}{EC_j}$, so that the expected values match.

In literature, d_j corresponds to an aggregate number of deaths for all similar individuals (i.e. with the same vector X of characteristics for a specific interval). However due to the additive property of the Poisson distribution, it is equivalent to consider afterward the aggregation of the prediction of the death indicator of

everyone, which means considering a model as follows:

$$\delta_{i,j}|X \sim \mathscr{P}(ec_{i,j}h_{i,j}) \tag{4.7}$$

Using a log-link function, the model becomes equivalent to a classical Poisson regression model with the exposure in offset:

$$\log(E[d_j|X]) = \log(EC_j) + X'\beta = \log(EC_j) + \log(h_j) \tag{4.8}$$

which means

$$\log(h_j) = \log(\frac{E[d_j|X]}{EC_j}) = X'\beta \tag{4.9}$$

We then apply the classical generalized linear model with a Poisson distribution to a pseudo data table with exposure. Through the likelihood optimisation with respect to β, we get the risk parameters:

$$L(\beta|X, D, E) = \prod_j \frac{(EC_j e^{X_j'\beta})^{d_j} e^{-EC_j e^{X_j'\beta}}}{d_j!} \tag{4.10}$$

It is also possible to consider an extension and add a regularisation factor to only consider the variables with a high explanatory power. When the probabilities are small if the *central exposure* is not available in the data, approximating the model with *initial exposure* predicts similar results.
Using the maximum likelihood estimator, the hazard function is estimated $\hat{h}_j = \exp(X'\hat{\beta}_j)$, from which the rate of mortality is derived:

$$q_j = 1 - \exp(-\hat{h}_j) = 1 - \exp(-\exp(X'\hat{\beta}_j))$$

4.2.2 Binomial regression

In the case we try to model individually for each subject the time of his death and we can aggregate afterward, one may think of using a logistic regression instead of a Poisson one. Within each interval, j, we try to predict the indicator of death $\delta_{i,j}$. Considering a traditional binomial model $\delta_{i,j} \sim B(q_j)$, where $\delta_{i,j}$ means the death indicator that individual i will die before $\tau_j + 1$ given he has survived up to τ_j and $q_j = P(T \leq \tau_j + 1|T > \tau_j)$, will not enable to take censoring into account. Thus we will weight the model with the **individual initial exposure**, $ei_{i,j}$, representing the amount of time where i is observed between $[\tau_j, \tau_j + 1]$ equals to 1 if survival exceeds τ_j or if the death is observed. The intuition behind comes from the Balducci hypothesis to make the estimator matches, as under this assumption

$\hat{q}_j = \frac{\sum_i \delta_{i,j}}{\sum_i ei_{i,j}}$. Indeed the weighted likelihood of the model can be expressed as:

$$L(q_j) = \prod_{i=1}^{l_j} q_j^{\delta_{i,j} ei_{i,j}} (1 - q_j)^{(1-\delta_{i,j}) ei_{i,j}} \tag{4.11}$$

Then the log likelihood is:

$$l(q_j) = \sum_{i=1}^{l_j} \delta_{i,j} ei_{i,j} log(q_j) + ei_{i,j}(1 - \delta_{i,j}) log(1 - q_j) \tag{4.12}$$

Derivating the previous equation with respect to q_j:

$$\frac{dl(q_j)}{dq_j} = \sum_{i=1}^{l_j} \frac{\delta_{i,j} ei_{i,j}}{q_j} - \frac{ei_{i,j} - \delta_{i,j} ei_{i,j}}{1 - q_j} \tag{4.13}$$

Thus,

$$(1 - \hat{q}_j) \sum_{i=1}^{l_j} \delta_{i,j} ei_{i,j} = \hat{q}_j \sum_{i=1}^{l_j} ei_{i,j} - ei_{i,j} \delta_{i,j} \tag{4.14}$$

And finally we get the desired estimator $\frac{\sum_{i=1}^{l_j} \delta_{i,j}}{\sum_{i=1}^{l_j} ei_{i,j}} = \hat{q}_j$, as $ei_{i,j}$ equals 1 for dead subjects, ie those with $\delta_{i,j}$ equals 1.

The previous consideration leads us to consider the **initial exposure** as weight in the logistic regression to model the impact of the covariate vector X on the mortality rate. We will thus estimate a risk parameter β such as $logit(E[\delta|X]) = \frac{q(X)}{1-q(X)} = X'\beta$, which means $q(X) = \frac{1}{1+e^{-X'\beta}}$
Injecting it in the log-likelihood implies:

$$\begin{aligned} l(\beta) &= \sum_{i,j} \delta_{i,j} ei_{i,j} log(q_j(X_{i,j})) + ei_{i,j}(1 - \delta_{i,j}) log(1 - q_j(X_{i,j})) \\ &= \sum_{\substack{i,j \\ \{i,j|\delta_{i,j}=0\}}} (-ei_{i,j} log(1 + e^{X'_{i,j}\beta})) + \sum_{\substack{i,j \\ \{i,j|\delta_{i,j}=1\}}} (-ei_{i,j} log(1 + e^{-X'_{i,j}\beta})) \end{aligned}$$

If we change the feature space from $\{0, 1\}$ to $\{-1, 1\}$, we can write (by a simple variable change $\delta'_{i,j} = 2\delta_{i,j} - 1$) :

$$\begin{aligned} l(\beta) &= \sum_{\substack{i,j \\ \{i,j|\delta'_{i,j}=-1\}}} (-ei_{i,j} log(1 + e^{X'_{i,j}\beta})) + \sum_{\substack{i,j \\ \{i,j|\delta'_{i,j}=1\}}} (-ei_{i,j} log(1 + e^{-X'_{i,j}\beta})) \\ &= \sum_{i,j} (-ei_{i,j} log(1 + e^{-\delta'_{i,j} X'_{i,j}\beta})) \end{aligned}$$

It enables us to get the formula often used in Machine Learning literature and implemented in existing Python packages such as *scikit-learn*, on which our implementation can thus rely. This is then the formula to be maximized. Derivating this equation with respect to β :

$$\frac{dl(\beta)}{d\beta} = \sum_{i,j} \frac{ei_{i,j}\delta'_{i,j}X'_{i,j}e^{-\delta'_{i,j}X'_{i,j}\beta}}{1 + e^{-\delta'_{i,j}X'_{i,j}\beta}} \tag{4.15}$$

From this equation, we can deduce the first order condition by making it equals to zero. There is no explicit formula, but it can be solved numerically.

4.2.3 Generalized Additive Model

One limit of the two previous models is that only linear interactions are modeled. A first possibility to capture non-linear patterns is the use of generalized additive models (GAM). One strength of GAM is that it produces a regularized and interpretable solution. However, interpretability has a cost as interactions are not detected by the model. Indeed, one should specify the interactions of the variables to be considered. In other words, GAMs strikes a nice balance between the interpretable, yet biased, linear model, and the extremely flexible, "black box" learning algorithms.

GAM models may be seen as an extension of GLM ones. A Generalized Additive Model (semi-parametric GLM) is a GLM where the linear predictor depends linearly on unknown smooth functions, s. Any function could be considered, but in practice, splines are the most widely-used one as they perform well in such circumstances. (see Hastie et al. [19] for more details about GAM)

B-splines

B-splines are curves, made up of sections of polynomials of the degree of the splines, joined together so that they are continuous in value as well as first and second derivatives.
It is a piecewise function build from a polynomial of degree d, where d is a hyperparameter. To set it, one may use the grid-search method to find the one leading to the best prediction performance.
The points at which the sections join are named knots. The locations of the knots must be chosen, for example, the knots would either be evenly spaced through the range of observed x values or placed at quantiles of the distribution of unique x values.

Finally, B-splines are local functions, that is to say, they are zero everywhere outside the range of their knots.

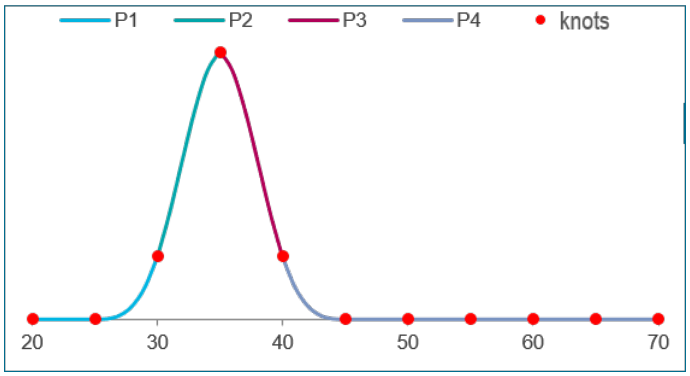


Figure 4.4: B-splines of degree 3

Figure 4.4 is an illustration when cubic splines of four polynomials, that is to say $s(x) = P_1(x) + P_2(x) + P_3(x) + P_4(x)$, with polynomials of degree three. The regularity conditions imply at each junction points that the value of the two polynomials and their first and second derivatives are equals. These constraints imply that a spline has only one degree of freedom to estimate.

In the same way, as previous GLM models discussed, Poisson and Binomial, considering exposures allows using GAM for survival time modeling.

Poisson GAM

The previous Poisson regression introduced before will thus be modified to capture the non-linear pattern. The model will still be stated as follows:

$$d_j|X \sim \mathscr{P}(EC_j h_j(X)) \tag{4.16}$$

A log-link function and the use exposure as offset is kept, however splines are applied to the covariates:

$$\log(E[d_j|X]) = log(EC_j) + \sum_{k=1}^{p} \theta_k S_k(X) = \log(EC_j) + \log(h_j) \tag{4.17}$$

which means

$$\log(h_j) = log(\frac{E[d_j|X]}{EC_j}) = \sum_{k=1}^{p} \theta_k S_k(X) = S'\theta \tag{4.18}$$

where $\theta = [\theta_1, .., \theta_p]'$ is the vector of regression coefficients and S is the regression matrix, which is the matrix of the B-spline transformation of the covariates :

$$S_j = [s'_1, ..., s'_n]', \text{ where } s'_i = (S_1(X_i), .., S_p(X_i))$$

The model becomes then equivalent to the Poisson linear regression if we consider the matrix S instead of the covariates. The θ parameter is once again obtained through a numerical likelihood maximization.

Binomial GAM

Based on the binomial regression section, the **initial exposure** will be introduced as weight in the binomial GAM to model the impact of the covariate vector X on the mortality rate. Using the same notation as in the Poisson GAM section, we want to estimate a risk parameter θ such as $logit(E[\delta|X]) = \sum_{k=1}^{p} \theta_k S_k(X) = S'\theta$, which means $q(X) = \frac{1}{1+e^{-S'\theta}}$
The problem is once again equivalent to the linear one as long as we consider the matrix of the B-spline transformation of the covariates. The weighted log-likelihood maximization with respect to θ enables to find an estimator of the risk parameter, from which it is possible to derive the mortality.

The estimation of Survival GAM was based on the package *PyGam*. However, this package requires a manual specification of all the variables, their interaction, and the associated spline level. An automatization step had to be introduced compared to the other implemented model to be called in the same manner.

4.2.4 Decision tree

The binomial regression aims to predict the death indicator, δ, using initial exposure as weights. As we are predicting a binary variable, it seems interesting to consider a weighted classification problem, such as a weighted decision tree for classification.

Classification Tree

For our purpose, we only need to focus on the intuition behind two-class classification trees. The goal of a binary classification tree is to successively divide observations into two groups with respect to the variable that creates the best split. The partition of the individuals is repeated on each subsample until reaching the *stopping criterion* or having only pure groups at the final step, that is to say,

groups in which all instances have the same label: δ.
The algorithm seeks to create at each step the two most homogeneous groups as possible by reducing the variance within a group, or equivalently to create two groups as different as possible by increasing the variance between groups. To determine the best split among the features and finding the partition rule, a *splitting criterion* is introduced to measure the quality of a split. In theory, all possible splits should be tested to find the best one, that is to say, the one giving the biggest reduction of variance. However, as it would be time-consuming some techniques of randomization are used in practice.
Two criterion are considered to build trees. The first one is the *entropy*, $H(x)$, which aims to measure the disorder quantity. The second one is the *Gini index*, $G(x)$, which measures the impurity within a group. To make a split from a given node composed of n_0 observations (x_i, δ_i) with $i \in \mathscr{I}_0$, we partition our data into two branches $\mathscr{I}_l$ and $\mathscr{I}_r$, with respectively n_l and n_l observations, in order to have the biggest information gain $I(x_0) - [\frac{n_r}{n_0} \times I(x_r) + \frac{n_l}{n_0} \times I(x_l)]$, where $I(x)$ is $G(x)$ or $H(x)$.

Weighted Tree

To prevent the bias due to censoring, we need to adapt the CART algorithm by injecting the initial exposure as weights in the *splitting criterion*.
Let define p_m, the proportion of deaths observed in node m, I_m the set of individuals of the node with characteristics x_m and $\hat{q}_m$ being the death rate estimator for individuals with x_m under the *Balducci hypothesis*:

$$p_m = \frac{1}{\sum_{i \in I_m} ei_i} \sum_{i \in I_m} ei_i \delta_i = \frac{\sum_{i \in I_m} \delta_i}{\sum_{i \in I_m} ei_i} = \hat{q}_m \text{ as } ei_i = 1 \text{ if } \delta_i \neq 0$$

Two splitting criterion may thus be adapted:
Gini index (cf annexe D):

$$G(x_m) = 2 \times \hat{q}_m \times (1 - \hat{q}_m)$$

Entropy:

$$H(x_m) = -\hat{q}_m \times \log_2(\hat{q}_m) - (1 - \hat{q}_m) \times \log_2(1 - \hat{q}_m)$$

One may also think to build a tree based on the deviance criterion such as described for the survival tree. In this case, the weighted log-likelihood of the binomial re-

gression should be considered to measure the deviance:

$$R(x_m) = 2[L_m(saturated) - L_m(\hat{\beta_m})] = 2\sum_{i \in I_m} ei_i log(\frac{1 + e^{-\delta_i X_i' \hat{\beta}_m}}{1 + e^{-\delta_i X_i' \beta_s}})$$

From this definition of the tree, a random forest algorithm may be derived as explained in the time-to-event section.

4.2.5 Gradient boosting

Using the Balducci hypothesis, we proved that the probability of dying in a specific interval j can be estimated as the ratio of the death indicator within this interval on the **initial exposure**:

$$P(T \leq \tau_j + 1 | T_i > \tau_j) = \frac{\sum_i^{l_j} \delta_{i,j}}{\sum_i^{l_j} ei_{i,j}} \tag{4.19}$$

From this result and the previous consideration on the logistic regression, we will thus consider traditional Machine Learning model to predict the probability of death within a time interval as the problem may become equivalent to a binary classification of the death indicator as long as **initial exposures** are used to weight the loss function in the model. We will focus on weighted-Gradient Boosting Machine (XGBoost, CatBoost, LightBoost).

Initialisation : $F_0(x) = argmin_\beta LL(\beta)$

For $m = 1$ to M *(number of weak learners):*

- Computation of the pseudo-residuals: $r_m = -\frac{dL(F_{m-1}(X))}{F_{m-1}(X)}$
- Fitting a new weak learner on pseudo-residuals: $f_m(X) = r_m$
- Finding the best γ_m by solving $\gamma_m = argmin_\gamma L(F_{m-1}(X) + \gamma \times f_m(X))$
- Update the new model: $F_m = F_{m-1} + v \times \gamma_m f_m$

Classification trees as used as weak learners, f_m and the opposite of the weighted-log-likelihood of the binomial regression as the loss function:

$$L(\beta) = \sum_i \delta_i ei_i log(q(X_i)) + ei_i(1 - \delta_i) log(1 - q(X_i)) \tag{4.20}$$

Where $q(X) = \frac{1}{1+e^{-X'\beta}}$

XGBoost

XGBoost (short of extreme gradient boosting) is an implementation of gradient boosted decision trees designed for speed and performance. It is initially created by Chen and Guestrin [9] and now is maintained by many developers. It offers a flexible framework to tree-based gradient boosting with randomly sub-sampling and regularization. But what is the most attraction of this method is that its training speed and model robustness are much better than traditional gradient boosting.

Using the original notation of the XGBoost paper [9], at time t, we have:

$$L^{(t)} = \sum_{i=1}^{n} l(y_i, \hat{y_i}^{(t-1)} + f_t(x_i)) + \Omega(f_t) \approx \sum_{i=1}^{n} l(y_i, \hat{y_i}^{(t-1)}) + g_i f_t(x_i) + \frac{1}{2} h_i f_t(x_i)^2 + \Omega(f_t) \tag{4.21}$$

In our case, we are interested at examining the gradient boosting for:

$$l(y_i, \hat{y_i}^{(t-1)} + f_t(x_i)) = ei_i y_i log(\frac{1}{1 + e^{-\hat{y_i}^{(t-1)} - f_t(x_i)}}) + ei_i(1 - y_i) log(1 - \frac{1}{1 + e^{-\hat{y_i}^{(t-1)} - f_t(x_i)}})$$

The gradient

$$\begin{aligned}
g_i &= \frac{\partial l(y_i, \hat{y_i}^{(t-1)})}{\partial \hat{y_i}^{(t-1)}} \\
&= -ei_i y_i \frac{e^{-\hat{y_i}^{(t-1)}}}{1 + e^{-\hat{y_i}^{(t-1)}}} + ei_i(1 - y_i) \frac{e^{\hat{y_i}^{(t-1)}}}{1 + e^{\hat{y_i}^{(t-1)}}} \\
&= -ei_i y_i \frac{e^{-\hat{y_i}^{(t-1)}}}{1 + e^{-\hat{y_i}^{(t-1)}}} + ei_i(1 - y_i) \frac{1}{1 + e^{-\hat{y_i}^{(t-1)}}} \\
&= -ei_i y_i + ei_i \frac{1}{1 + e^{-\hat{y_i}^{(t-1)}}} \\
&= ei_i \hat{q_x}^{(t-1)}(x_i) - ei_i y_i
\end{aligned}$$

With the log likelihood as the loss function, we see that the gradient for each instance i is proportional to the difference between the predicted dead and the observed ones weighted by the initial exposure. More importance is given to the observations with a bad mortality prediction.

The hessian

$$h_i = \frac{\partial^2 l(y_i, \hat{y_i}^{(t-1)})}{\partial \hat{y_i}^{(t-1)}} = ei_i \frac{e^{-\hat{y_i}^{(t-1)}}}{(1 + e^{-\hat{y_i}^{(t-1)})^2}} = ei_i \hat{q_x}^{(t-1)}(x_i)(1 - \hat{q_x}^{(t-1)}(x_i)) \tag{4.22}$$

The optimal weight

$$w_j^* = -\frac{\sum_{i\in I_j} g_i}{\sum_{i\in I_j} h_i + \lambda} = -\frac{\sum_{i\in I_j} ei_i \hat{q_x}^{(t-1)}(x_i) - ei_i y_i}{\sum_{i\in I_j} ei_i \hat{q_x}^{(t-1)}(x_i)(1-\hat{q_x}^{(t-1)}(x_i)) + \lambda} \tag{4.23}$$

The value of the weight w_j^* can be high and the initial second-order Taylor approximation can thus be no longer sustainable. A shrinkage factor η can be here introduced to diminish the importance of the current learner.

The scoring function for the weak learner t is

$$L^t(q) = -\frac{1}{2}\sum_{j=1}^{T} \frac{(\sum_{i\in I_j} ei_i \hat{q_x}^{(t-1)}(x_i) - ei_i y_i)^2}{\sum_{i\in I_j} ei_i \hat{q_x}^{(t-1)}(x_i)(1-\hat{q_x}^{(t-1)}(x_i)) + \lambda} + \gamma T \tag{4.24}$$

The literature also recommends using some subsampling when constructing the tree to diminish the greediness of the trees.
In a word, the XGBoost corrected with the initial exposure is a model that will correctly integrate the partial information for each duration where an individual is observed. The learners recursively segment the space with weights dependant on the performance of the precedent learners and the regularisation parameters. This dependency leads to difficulties in the analysis of the structure of the trees. However, the celebrity of the XGBoost is done and the model has demonstrated high performances in many data science competitions.

CatBoost

CatBoost is an open-source Machine Learning algorithm developed by Yandex [12] that uses gradient boosting on decision trees. The difference with other gradient boosting algorithms is that it successfully handles categorical features and uses a new schema for calculating leaf values when selecting the tree structure, which helps to reduce overfitting. Most popular implementations of gradient boosting use decision trees as base predictors. However, decision trees are convenient for numerical features, but, in practice, many datasets include categorical features, which are also important for prediction. Categorical features are not necessarily comparable with each other and most of the time to deal with categorical features in gradient boosting, a pre-processing is needed to convert the categorical value into numbers before training. With Catboost, this step is no longer needed but the categorical feature cannot contain missing values.

Within the developed library, a pre-processing step is thus still implemented to replace all missing value to the label *'Missing value'*, when some are contained in a categorical variable.

LightGBM

Microsoft researchers [22] have proposed a novel open-source gradient boosting algorithm called LightGBM. This new implementation aims at optimizing the process for large data sets with a high feature dimension. Contrary to other gradient boosting implementation, LightGBM does not scan all the data instances to estimate the information gain of all possible split points thanks to two novel techniques: *Gradient-based One-Side Sampling* and *Exclusive Feature Bundling* to deal with a large number of data instances and a large number of features respectively. The principle of *Gradient-based One-Side Sampling* is to exclude a significant proportion of data instances with small gradients as they play a less important role in the information gain, and only use the rest to estimate the information gain. The principle of *Exclusive Feature Bundling* is to reduce the number of features by bundle mutually exclusive features (i.e., they rarely take nonzero values simultaneously). The experimental results show that LightGBM can significantly outperform XGBoost in terms of computational speed and memory consumption while achieving almost the same accuracy.

Chapter 5

Interpretation

Some of the above-presented models are *"black box"* Machine Learning models, which cannot be directly interpretable. However, in many cases, understanding the model remains essential for different reasons: Respecting the regulatory requirement to justify every decision taken, keeping stakeholder trusts by understanding the results produced, etc.
For that purpose, three complementary methods of interpretation have been implemented within the library. The presented methods are post-hoc interpretation methods, i.e. once the model has been fitted, and agnostic to the model, i.e. independent of the algorithm to be explained. (see [11] for more details)

5.1 Permutation variable importance

Feature importance can be assessed using the permutation method which has been introduced by Fisher [13]. In this method, we measure the increase in the prediction error of the model after we permuted the values of the features. The permutation breaks the relationship between the feature and the true outcome thus it increases the prediction error. The higher the increase in the prediction errors, the higher the importance of the feature.
In practice, after having trained a model and assessed the model performance, for each feature j one can proceed as follows:

- For each observation replace the feature with a randomly generated variable. This enables to breaks the association between the feature and the true outcome to predict
- Produce model forecast and assess the loss in model performance L^j. It can either be the difference or the ratio between the original error of the model and the one after permutation.

High loss in performances shows that the model relied heavily on the feature to produce the predictions. If performances are unchanged after shuffling values of one feature, it means that the model ignored the feature for the prediction. Thus, the importance of the features is ranked by their descending L^j.

This approach provides a global insight into model behavior. By permuting a feature, we do not only break the relationship with the outcome variable but also the interaction effects with all the other variables. The inclusion of the interaction effect within the measure has not only advantages but also disadvantages. The importance of the interaction is indeed measured twice, in the two associated features.
The big advantage of this approach is that it is less time-consuming as we do not need to retrain the model. This method requires only the application of the trained model to different observations and an error computation.

Within the library, every model class inherits a permutation variable importance method. This method shuffles the modalities of a variable and assesses the loss in the performance of the model on the noisy data.

5.2 Partial Dependence

This method is the oldest one to interpret Machine Learning models, it has been introduced by Friedman [14]. The partial dependence plot shows the marginal effect of one or two features have on the predicted outcome. In practice, more features could be considered but we limit to one or two for representation purposes, i.e. producing 2D or 3D plots. The plot can show whether the relationship between the target and a feature is linear, monotonic, or more complex.
The *Partial Dependence function* is defined as the average model prediction for a given value of a feature. When dealing with numerical features, it can be defined as follows:

$$\hat{f}_{x_s}(x_s) = E_{x_c}[\hat{f}(x_s, x_c)] = \int \hat{f}(x_s, x_c)\partial P(x_c)$$

where x_s are the features for which the Partial Dependence function should be plotted and x_c the other features used to train the model. In practice, there are one or two features in the set S, those for which we want to know the effect. The *Partial Dependence function* is estimated thanks to:

$$\hat{f}_{x_s}(x_s) = \frac{1}{n}\sum_{i=1}^{n} \hat{f}(x_s, x_c^{(i)})$$

where $x_c^{(i)}$ are feature values from the data for the features in which we are not interested to measure the impact.

To produce the Partial Dependence curve in practice, after having trained a model, one can process as follows:

- Consider a data sample
- Select a feature
- Replace the feature values by x
- Compute the average model prediction for each value x of the feature thanks to the previous formula

When dealing with categorical features, the computation of the Partial Dependence is a bit different. For each of the categories, we get an estimate by forcing all data instances to have the same category. For example, if we are interested in the Partial Dependence plot for the gender, we get 2 numbers, one for each. To compute the value for "female", we replace the gender of all data instances with the number associated with "female" and average the predictions.

If the computation of this kind of plot is intuitive and has a causal interpretation, one has to keep in mind that Partial Dependence gives only the average trend. It can thus differ a lot when looking at subsets. Besides the Partial Dependence plot relies on the assumption that the features in C and S are not correlated. When the assumption holds, the *Partial Dependence Plot* perfectly represents how the feature influences the prediction on average. That is to say how the average prediction in the dataset changes when the j-th feature is changed. However, if this assumption is violated the averages calculated for the Partial Dependence plot will include data points that are very unlikely such as a two-meters tall person weighing thirty kilograms.

Every model class implements a Partial Dependence function. For the discrete models, mortality is predicted for different modalities of the variable. For survival models, the hazard rate is the indicator. The models non-able to handle categorical features transform the data after the variable has been set to a particular modality.

5.3 SHAP

SHAP (*SHAPley Additive exPlanations*) by Lundberg and Lee [24] is a method to explain individual predictions. The goal of SHAP is to explain the prediction of an instance x by computing the contribution of each feature to the prediction. It enables to quantify the role of each feature in the final decision of a model.
The principle comes from the game theory as we consider the prediction as the payout of a game in which each feature value of the instance is a "player". SHAP will then be based on the game theoretically optimal Shapley values as these values tell us how to fairly distribute the "payout" (i.e. the prediction) among the features.

Let's explain the general idea thanks to an example inspired by the one given in his book by Molnar [25]. We have trained a Machine Learning model to predict a life duration. For a certain individual, it predicts 10 years and we need to explain this prediction. The individual is 65 years old male and he is a current smoker with a past cancer history. The average prediction for all individuals is a remaining life of 8 years. SHAP tries to explain how much has each of the previous feature values contributed to the prediction compared to the average prediction thanks to the game theory intuition.
The "game" is to predict every single instance. The "players" are the feature values of the instances, in our example, they are the gender, age, smoking situation, and cancer history. Together these features implied a prediction of a 10-year life duration when the average prediction is 8 years. The goal is then to explain the difference of -2 years between the two. For example, the answer could be: The age contributed 15 years, the gender contributed 1 year, the smoking situation contributed -10 years, and the past cancer history -8 years. The contributions add up to -2 years of remaining life, which is the final prediction minus the average predicted remaining life duration.

Concretely the *Shapley value* is the average marginal contribution of a feature value across all possible coalitions. We will evaluate the contribution of the age when it is added to a coalition of gender and smoking status. We simulate that only these three features are in a coalition by randomly drawing another individual from the data and using its value for the cancer history. The value of cancer history is replaced by the randomly drawn, let's assume it is no past cancer. Then we predict the remaining life duration of an individual with this combination, assuming the model predicts x_1 years.
In a second step, we remove the age from the coalition by replacing it with the age

value from the randomly drawn individual which may be the same but may also be different as it is randomly drawn. We predict again the duration with this new age value and the prediction becomes x_2. Thus, the contribution of the new age was $x_1 - x_2$. This estimate strongly depends on the values of the randomly drawn observation that served as a "donor" for the cancer history and a feature value. To get the best estimate as possible the idea is to repeat this sampling step and averaging all the contributions.

Formally to approximate the Shapely estimation for a single feature value j of the instance of interest x, one can thus iterate M times the following m process:

- Draw random instance z from the data matrix X
- Choose a random permutation o of the feature values
- Order instance x: $x_o = (x_{(1)}, ..., x_{(j)}, ..., x_{(p)})$
- Order instance z: $z_o = (z_{(1)}, ..., z_{(j)}, ..., z_{(p)})$
- Construct two new instances:
  - With feature j: $x_{+j} = (x_{(1)}, ..., x_{(j-1)}, x_{(j)}, z(j+1)..., z_{(p)})$
  - Without feature j: $x_{-j} = (x_{(1)}, ..., x_{(j-1)}, z_{(j)}, z(j+1)..., z_{(p)})$
- Compute the marginal contribution $\phi_j^m(x) = \hat{f}(x_{+j}) - \hat{f}(x_{-j})$

The Shapley value is the average over the M iterations : $\phi_j(x) = \frac{1}{M} \sum_{m=1}^{M} \phi_j^m(x)$
The interpretation of the Shapley value for feature value j, ϕ_j, is: Given the current set of feature values, the contribution of the value of the j-th feature is ϕ_j to the the difference between the actual prediction of this particular instance and the average prediction for the data.
Then the procedure must be repeated for each of the features j, to get all Shapley values.
SHAP is thus quite time-consuming as this process has to be repeated for all possible coalitions. Computation time increases exponentially with the number of features, that is why most of the time only the contributions for a few numbers of samples of the possible coalitions are computed.

Chapter 6

Mortality modeling with the survival analysis library

The theoretical aspects of the methods implemented within the Python library have been presented in the previous sections.
In this section, we will focus on the library use and interest in a business matter to automate and facilitate the mortality modeling of an insurance portfolio. The calculations are performed on NHANES database as it is an open-source mortality experience database that contains a significant amount of information.

6.1 Dataset presentation

NHANES is the acronym of National Health and Nutrition Examination Survey, which is a study program designed to assess the health and nutritional status of adults and children in the United States. The survey combines interviews and physical examinations. This program began in the early 1960s and has been conducted as a series of surveys focusing on different population groups or health topics. In 1999, the survey became a continuous program that focuses on a variety of health and nutrition measurements to meet emerging needs. NHANES program gathers information that includes demographic, socioeconomic, dietary, and health-related questions. It gathers also information collected via examination consisting of medical, dental, and physiological measurements, as well as laboratory tests administered by trained medical personnel.

Considering the programs' periods, we use NHANES Continuous data, which refers to a survey between 1999 and 2014. There are five main categories of data set in the NHANES program: demography data set, dietary data set, laboratory data set, examination data set and questionnaire data set. Each of them includes more than ten sub-datasets and hundreds of candidates of explanatory variables.

In the NHANES program information is collected thanks to interviews and examinations. Persons complete an interview form and answer questions more about qualitative information, like life behavior and self-assessment of health status. While the physical examination of participants allows collecting objective quantitative information reflecting their health status, like height, weight, blood pressure, and cholesterol.
As NHANES is a national program, a sampling weight is provided in the dataset. It reflects the importance level in terms of population share in the USA of the same profile's people as this particular sampled individual. In the following, we will ignore the weights and consider each line as an individual. The purpose of our study is only to highlight how some factors may impact mortality. We do not have an interest in the representativity in terms of the American population.

Descriptive statistics

The dataset is composed of 65 018 individuals, and 106 variables. The follow-up time does not exceed sixteen and a half years. Among the observation, only 13.4% of individuals were dead during the follow-up period, and the remaining ones are all censored or are alive at the end of the observation period.

Among all the available variables, our analysis will only consider the most meaningful twenty-seven ones described in Annexe E. Some of the variables contain too many missing values or redundant information so we put them aside. We checked the correlation among the variables by computing the correlation matrix for the numerical variables and the Cramer's V between the categorical ones. The figure 6.4 highlights that only a few of our variables are subject to a high correlation. This correlation is mainly caused by similar information or causality effect.

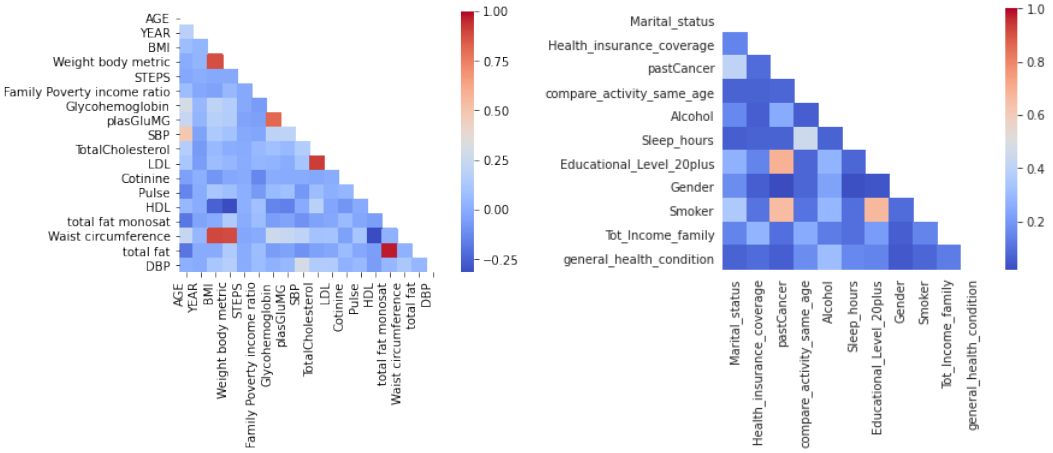


Figure 6.1: Correlation matrix and Cramer's V

Only the *Body Mass Index* will be conserved as it appears to gather the information of weight body metrics and waist circumference. We will not distinguish monounsaturated or the global fat amount as it highlights the same information. Regarding the categorical variables, even if correlation patterns are detected, we decided to conserve all variables as they contain valuable information.

We then check whether the observed mortality differs depending on the selected variables and their values. Including variables that have no impact on mortality will only contribute to add noise in the models and will not improve the predictive performance. The impact of smoking habits and insurance coverage on mortality is represented in Figure 6.2 as well as the distribution of the observations. As expected, some factors imply a higher chance of dying such as being a smoker.
Surprisingly, being covered by insurance seems to increase mortality. However, this deduction relies on the other factors, which means that it may be due to an over-representation of other risk factors among the insured individuals in our dataset. This highlights the need of going further in mortality analysis to avoid any misinterpretation.

```
from scor_survival.mortality import plot_univariate_mortality
plot_univariate_mortality(dataset,variable,event,exposure)
```

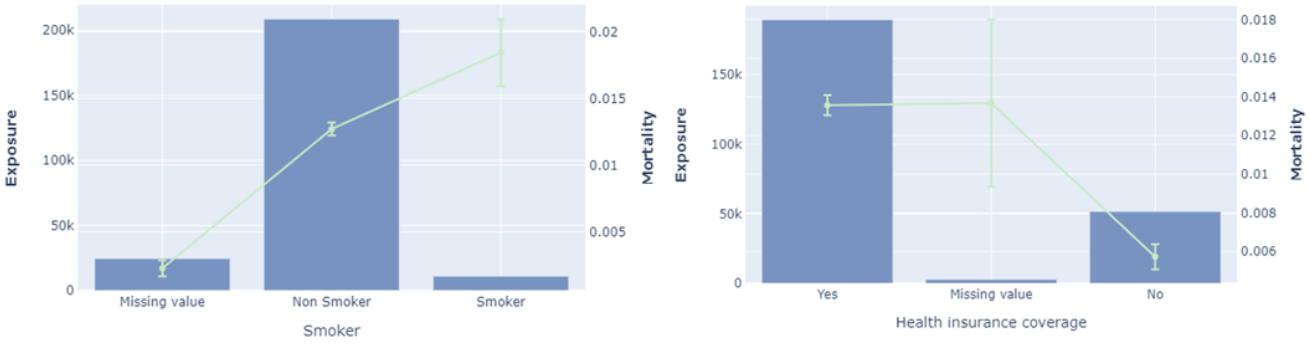


Figure 6.2: Impact of insurance coverage and cigarettes on mortality

6.2 Data pre-processing

Pseudo data table transformation

In the original data set, the followed-up time in months and a death indicator was the only information available, which means we disposed of a traditional survival data table. To apply our models based on the exposure approach, the first step was to compute the pseudo survival data table.

For our study, we decided to consider the survival time in years rather than in months. A yearly basis seems granular enough for life insurance mortality modeling purposes. Besides, this choice enables to considerably reduce the number of rows and thus the computation time. If one is interested in studying survival on monthly basis, more granular time intervals can be considered leading to a much bigger pseudo data table.
Considering the approach described in section 2.6.3, we created for each observation as many rows as the number of years individuals were observed. For each one of the created rows, we added an explanatory variable **year** to represent the time interval in which exposures *(the initial and the central one)* are computed.
The final step is to modify the time-varying variables. In this case, we only dispose of age. Thus, we only had to increment it by one in each interval, which means computing the *Atteined Age* by summing up the age at the start of observation and the survival year. Implicitly a hypothesis is made that the date of birth and the date of observation are the same.
A specific class has been created for the transformation of a survival data table into a pseudo data table:

```
from scor_survival.discretizer import exposure_expansion
pseudo_data = exposure_expansion(survival_data,
                                 time, event, indivdiual_key,
                                 entry_age, entry_year)
```

Missing value treatment

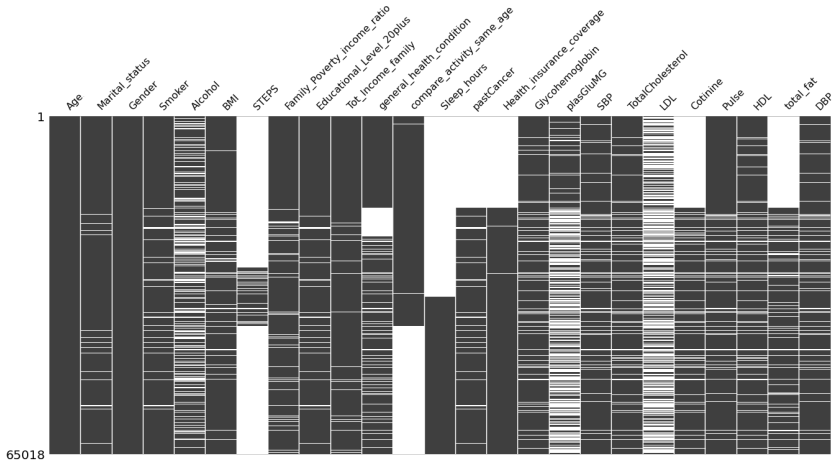


Figure 6.3: Missing value distribution among the variables

The missing value proportion for some of our variables is quite significant, which means we have to deal with a missing value imputation to train well our models. Two approaches have been considered depending on the variable type. For the numerical variables, the median of the series was assigned to all unspecified values. For the categorical variables, a new category has been created for each variable as *'Non available'*. This might enable to highlight some patterns. As shown in Figure 6.3, the variables steps, sleep hours, and compare activity same age are subject to a lot of missing value. However, we decided to keep these three variables in our models as other studies have shown their importance as a death factor.

Categorical feature

As some of our models cannot deal with categorical features, we apply one-hot encoding on all factor variables. One-hot encoding is a representation of categorical variables as binary vectors. This method produces a vector with length equal to the number of categories in the data set. If a data point belongs to the i^{th} category then components of this vector are assigned the value 0 except for the i^{th} component, which is assigned a value of 1. In this way, one can keep track of the categories in a numerically meaningful way.

Insurance portfolio reproduction

The main issue when dealing with insurance matters is that the global population is not representative of an insured portfolio. However, when data is not available for a specific business use case, insurers can build partnerships with insurtech, hospitals, universities, or governments to access new data sources. A model calibrated on those external data aims to replicate knowledge. Being used in another context can lead to poor performances.
On a business matter, this raises an issue as we cannot transfer or interpret a model trained on the global population to reveal the behavior of an insurer datasets. The real advantage of NHANES is the inclusion of the health insurance coverage variable, which specifies whether an observation is covered by an insurance policy and the overrepresentation of covered individuals among the observations.

We will not remove the non-covered individuals as it would be interesting to confirm the fact that the marginal effect of being covered by a health policy does reduce mortality contrary to the observed global effect (cf Figure 6.2). As highlighted in the figure below, old individuals are over-represented among insured people regardless of smoking habits, which contributes to bias the global effect.

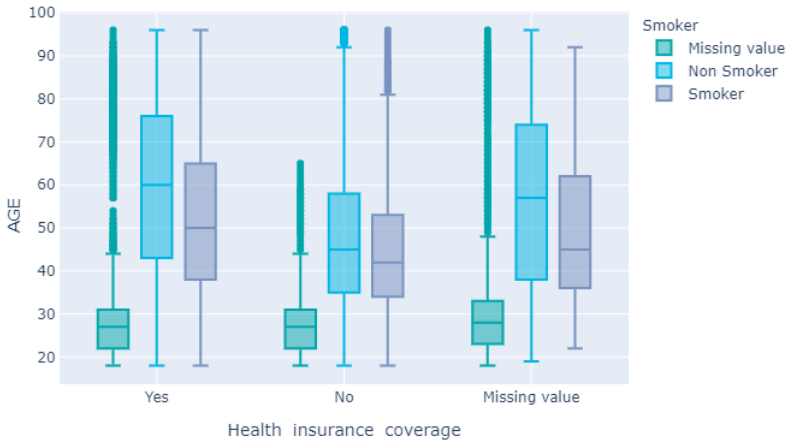


Figure 6.4: Insured population characteristics

We will remove or modify some instance values to make them match better to insurance conditions. The **quality of the data** should be correctly assessed before starting a project and collect business requirements. For instance, an underwriting team accepts and declines files based on medical information. This decision is based on the data produced by the applicant and the underwriter expert opinion. Based on medical considerations, we will thus remove all individuals, whose risk of death is too high. Insurers would indeed not accept to cover them as they may be too expensive clients.

Age

First, we will only consider the individuals, who are less than eighty years old as it is quite uncommon to sell an insurance policy to older people. It represents only 3% of our dataset so removing those rows is not a problem to calibrate the model.

Diabetes

One of the major risk factors of death is diabetes as it is a factor of cardiovascular accidents. According to the American Heart Association [3], at least 68% of people age sixty-five or older with diabetes die from some form of heart disease and 16% die of stroke. Diabetes disease occurs when the body cannot produce enough insulin. It is diagnosed by observing raised levels of glucose in the blood. Thus insurers often ask for *Glycohemoglobin* and *Fasting glucose* sample to test the presence of diabetes.
In general, 6.5% of glycohemoglobin can be adopted as a diagnosed threshold for diabetes. A 1% increase in absolute concentrations of glycohemoglobin is associated

with about a 10-20% increase in cardiovascular risk. Based on this consideration, we will only conserve individuals, whom glycohemoglobin concentration rate is less than 8%.
A fasting glucose rate between 70.27 and 97.3 mg/dl is regarded as normal, between 97.3 and 126.13 mg/dl is prediabetes and more than 126.13 mg/dl is considered as diabetes. Above 140 mg/dl, individuals will be removed from our dataset.

BMI

BMI is short of Body Mass Index. It is defined by the ratio between the weight in kilograms and the square of height in meters. For the adults over twenty years old, the *World Health Organization* [28] defines the nutritional status categories as follows:

BMI	Nutritious Score
Below 18.5	Under weight
18.5 - 24.9	Normal weight
25.0 - 29.9	Over weight
30.0 - 34.9	Obesity class I
35.0 - 39.9	Obesity class II
Above 40	Extreme obesity

Table 6.1: Body Mass Index - Nutritious Status

Based on these numbers and characteristics, we will only consider individuals with a BMI until forty. For an insurer, extreme obesity represents a risks too high to be covered. Indeed, calibrating a model on the global dataset, in which outliers are included, contribute to building less robust models. As we are not interested in their impact on mortality and they are rarely contained in an insurer portfolio, it is thus better to remove them for the calibration.

Dyslipidemia

Dyslipidaemia is an abnormal amount of lipids in the blood. This disease is thus tightly associated with total cholesterol volume. Usually we define *good* and *bad* cholesterol: *HDL*, which means high-density lipoprotein cholesterol and *LDL*, which means low-density lipoprotein cholesterol. To avoid Dyslipidaemia, one should have enough good cholesterol and not too much of the bad kind.
For *LDL*, a standard group is considered if the concentration is less than 2.5 mmol/L. For *HDL*, someone is healthy for a concentration of more than 1.5 mmol/L for the women and 1.1 mmol/L for the men. As in our dataset, the range values

of LDL and HDL seems to be normally distributed with a small variance around these healthy references, we will not apply any modification.

Blood pressure

Blood pressure categories can be constructed based on systolic and diastolic blood pressure measurements. Based on the Figure 6.5, individuals with systolic blood pressure above 170 and diastolic blood pressure above 100 will be removed.

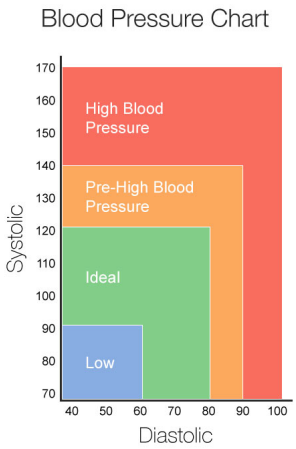


Figure 6.5: Blood pressure categories [2]

After all these considerations, we now have 29 870 individuals at disposal to calibrate the different models. Before calibrating them, we just split our data between a train and a test sample by making sure the split is stratified and the mortality contained within the two subsets is equivalent to the global mortality of 9%.

6.3 Models calibration

In data science, model hyper-parameters calibration is essential as it impacts the model performance. The calibration may be challenging as it often implies long and tedious calculations.
As illustrated in Figure 6.14 for the CatBoost model, the parameters choice implies very different performance results. We applied a grid search method to calibrate them. Based on the AUC, best model is obtained for a fit with 150 estimators of depth 8. To keep the SMR close to 100% we retained a learning rate of 0.07. This calibration appears to us to be a good balance as it provide good results on both metrics considered AUC and SMR.

```
cb = scor_survival.models.discrete.SurvivalCatBoostClassifier()
g = {'depth':[3,8,12], 'learning_rate':np.arange(0.01, 0.15, 0.01)
    , 'iterations':[50,100,150]}
gs = cb.gridsearch(g, X, y, w)
```

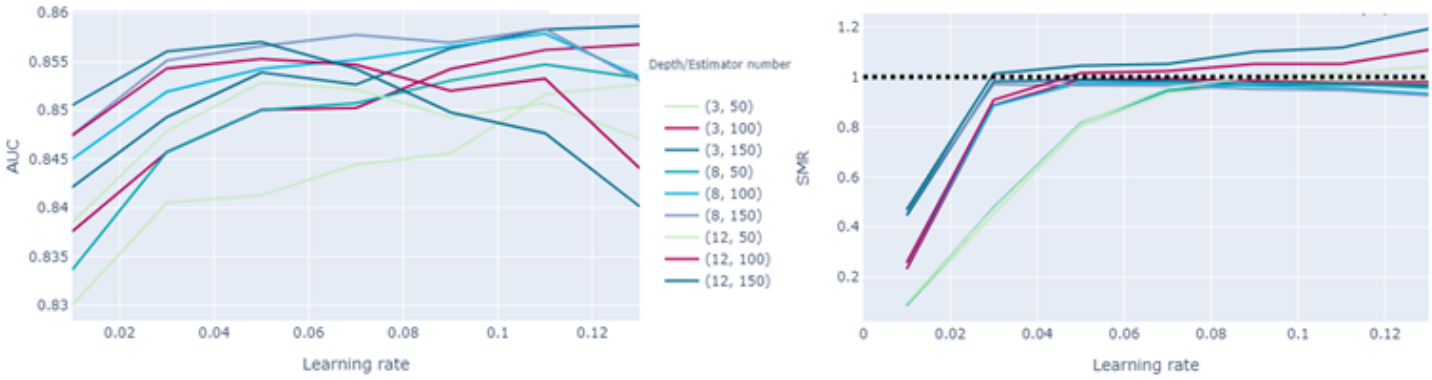


Figure 6.6: Catboost hyper-parameters selection

Besides, to avoid overfitting, it is important to tune a model with a similar AUC performance on the train and test data sets. The presence of over-fitting is highlighted in the figure as for many deep trees and high learning rates the AUC is decreasing on the test set.

For tree-based models, the research of hyper-parameters: *depth, leaves size, estimator numbers, learning rate, etc.* is the most complex. The research with k-fold cross-validation is advised with a separate hold out. The lack of computation resources to train a model may lead to a too tiny grid of parameters to test. This leads to over-fitting of the model. For instance, a data scientist could be tempted to train an XGBoost with an early stop on the number of estimators.

Based on this thought, within the library, every model is extended to implement a *scikit-learn* interface. This allows the parametrization before training a model. A default grid of parameters is also proposed as a first very parsimonious model. All the models presented in the following have been calibrated thanks to a k-fold cross-validation grid search based respectively on the weighted AUC or the C-Index for discrete or continuous models.

6.4 Models validation

A survival model is an estimator that should respect certain mathematical properties. The complexity of the models can frequently lead to the production of biased or inadequate models. The adoption of wrong models in insurance is a threat to

the business. Constant monitoring of the development process and the usage of the model is mandatory to reduce the risk. Our main focus is to obtain a non-biased model. Mathematically, the non-bias property for parametric models is:

$$E[E[Y|X]] = E[Y] \tag{6.1}$$

This equation states that the weighted average of the predictions is equal to the actual risk measured on the data. The model is expected to replicate the average mortality of the dataset. This equation consideration should always hold on the training set. This is the very first test to conduct after the calibration of a model. Within the library developed, a function implements for all discrete and continuous models this test. For continuous models, this test is the comparison of the average of the survival curves predicted by the model with the Kaplan-Meyer estimation (cf Figure 6.7). Even if it seems that on average the mortality is a bit overestimated by our models for the high duration, we will consider that all our models are acceptable. Only the Cox-XGBoost model is not contained within the Kaplan-Meier confidence interval for the last duration. The increase of the gap with duration is expected as the previous small errors are added successively.

```
from scor_survival import analysis
models = [coxnet, stree, coxtree, rsf, cox_xgboost, cox]
analysis.plot_average_survival(models, X, expo, event,
    max_duration)
```

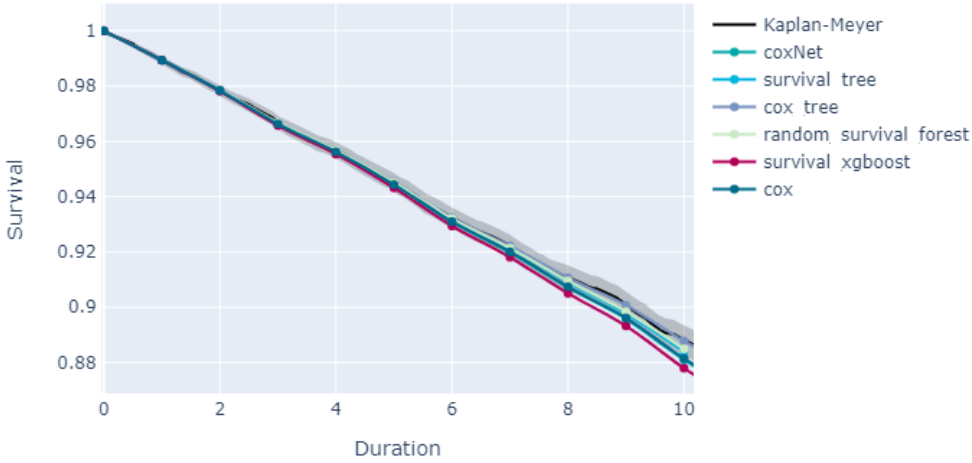


Figure 6.7: Validation for continuous model

For discrete models, the validation test is computing the ratio of the number of observed deaths and the expected number of deaths based on model predicted,

which is the SMR.

Models	*SMR Train*	*SMR Test*
Binomial Regression	0.99	0.99
Poisson Regression	1.00	0.98
logistic GAM	1.00	0.96
Random Forest	0.99	0.99
LightGBM	0.99	0.98
XGBoost	0.99	0.94
Catboost	1.01	0.99

Table 6.2: Validation metrics for discrete models

The SMR should be close to 1 on the train set if the model is well trained. Based on the global SMR, all our models are satisfying as the metric on the train and test set is close enough to 1. A deeper comparison will be conducted to define the one with the best predictive performance.

However, if the ratio is significantly different on the test set compared to the ratio on the train one, then we may conclude that our model is over-fitted.
The *XGBoost* model appears to be less effective than the other ones as it highly relies on the hyper-parameter choice. As it requires more computation capacity than the others, it is quite difficult to test a large grid of hyper-parameters.
The logistic GAM model seems to be over-fitted as well. Indeed, the model fits perfectly the train set but is biased on the test one. As the standard error on the test set is equal to 5.3%, this gap can be acceptable, as 100% is still contained within the confidence interval.

6.5 Models interpretation

6.5.1 SHAP

SHAP theory allows interpreting the model forecasts even if it is complex. A negative and high SHAP value means that the features impact largely the odds ratio negatively, which means that it reduces the dying probability. In the case of a lightGBM model, it seems that age has the highest impact on the predicted probability of dying. The prediction is going on the expected side, as the older an individual, the higher chance of dying.
As confirmed by the Figure 6.8, the use of the variables by the model to predict a

dying probability is coherent with medical evidence for almost all variables. Indeed, having financial difficulties, smoking, or having bad health habits and conditions have been proved to increase mortality.

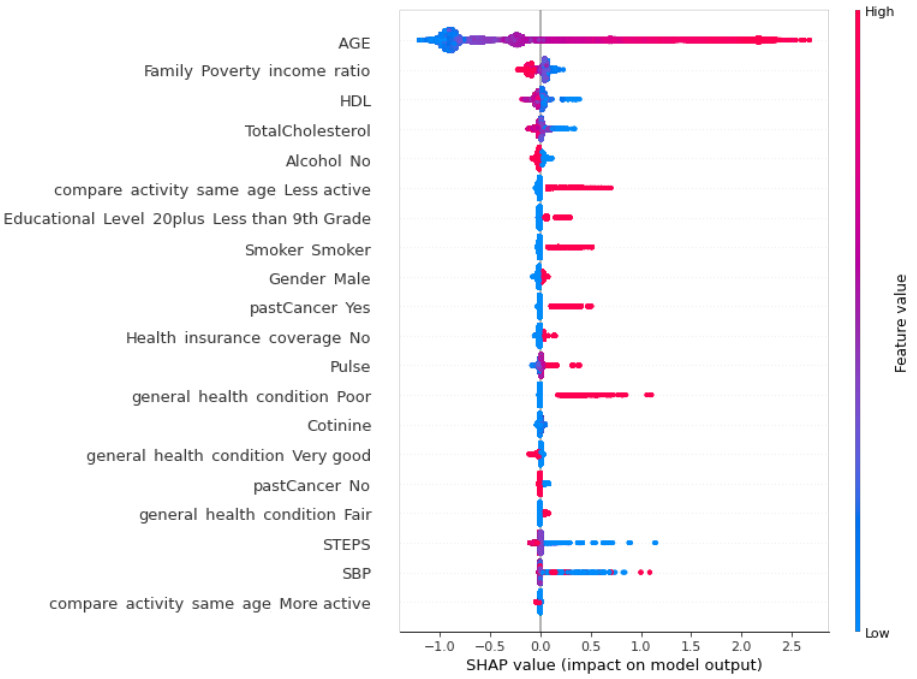


Figure 6.8: SHAP interpretation for the LightGBM model

The interpretation for each blood pressure is difficult as it is a combined effect, which is not explained with the only SHAP value. A dependence plot is a scatter plot that shows the effect a single feature has on the predictions made by the model. The color corresponds to a second feature that may have an interaction effect with the feature we are plotting. If an interaction effect is present between this other feature and the feature we are plotting it will show up as a distinct vertical pattern of coloring.

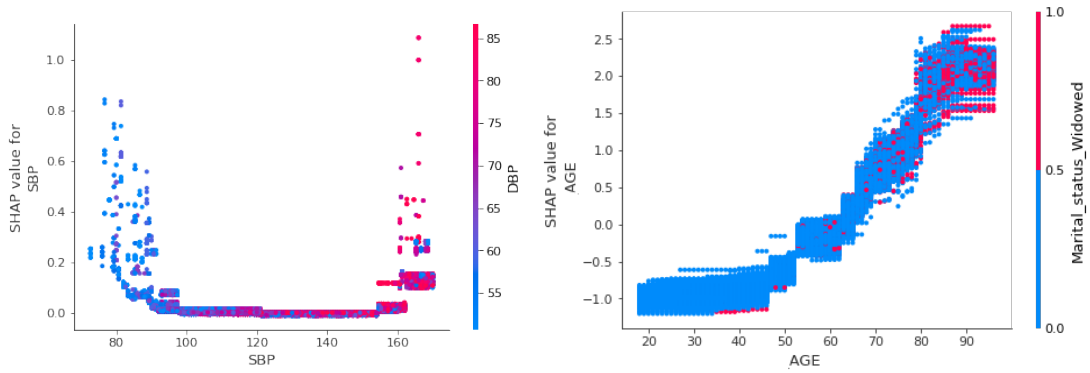


Figure 6.9: SHAP dependence for the LightGBM model

In the previous figure, the combined medical effect of blood pressure is visible. Indeed, having low values for both, which means hypotension, or high values for both, which means hypertension, implies a higher risk of dying as highlighted with high positive SHAP values in the figure 6.5. The second graph confirms the fact that there is an over-representation of widowed among old people, but that being a widow itself is not necessarily the death factor. There is indeed no distinct pattern of coloring for old people, as we can find widows for every SHAP value.

6.5.2 Partial Dependence

Visualizing the Partial Dependence is interesting to capture the causal effect of a variable on mortality and remove the hidden effects of other variables.

```
from scor_survival import analysis
model = [logistic, poisson, rf, lightgbm, xgboost, gam, catboost]
analysis.plot_partial_dependence(model, var, X, exposition, event)
```

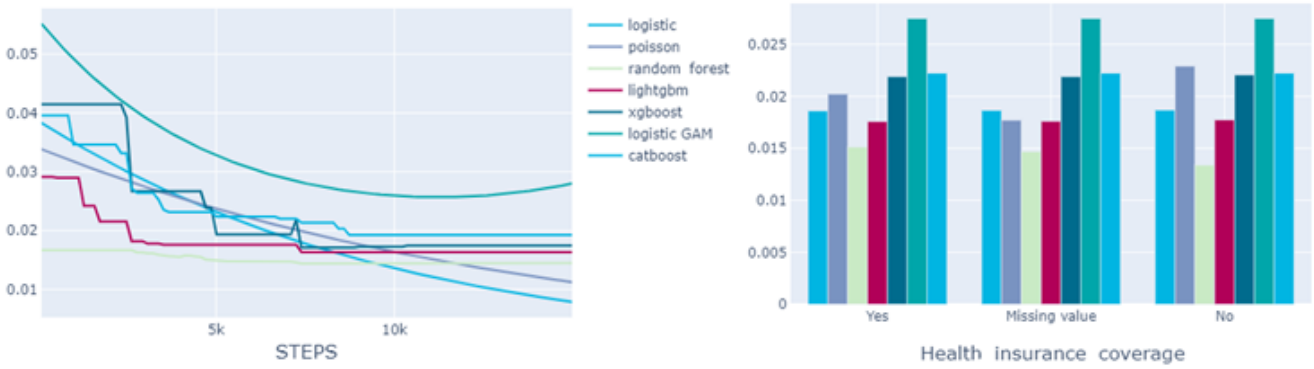


Figure 6.10: Partial Dependence for discrete models

The Partial Dependence above confirms indeed the decreasing marginal effect of the number of steps on the predicted mortality. It seems that the random forest model is less effective in capturing this trend as its Partial Dependence curve is almost flat. It highlights that the effect seems to be nonlinear, which contributes to conclude that the mortality will be better-modeled thanks to Machine Learning or non-linear model.

Concerning the marginal effect of the health insurance coverage, the Partial Dependence corrects the global effect of higher mortality among the insured individuals (cf Figure 6.2). Even if the effect does not seem significant, the previous figure shows slightly higher mortality among the non-covered individuals for all our models except the random forest one.

6.5.3 Variable Importance

For all our models, the variable *Age* is the one with the highest impact. As the impact of age is way higher than the other, we remove it from the charts for the sake of visibility. In the figure below, we plotted the Variable Importance for all our implemented models. In the different plots, we see that even if the value is a bit different between the models, their ranking is quite similar.
As the models are fitted on different data, *pseudo data tables* or *survival data tables*, two specific figures are provided for continuous and discrete models.
One remark regarding variable *YEAR*, it is considered only in discrete models as duration is embedded in the continuous model.

```
from scor_survival import analysis
models = [logistic, poisson, rf, lightgbm, xgboost, gam, catboost]
analysis.plot_var_importance(models, X, exposition, event)
```

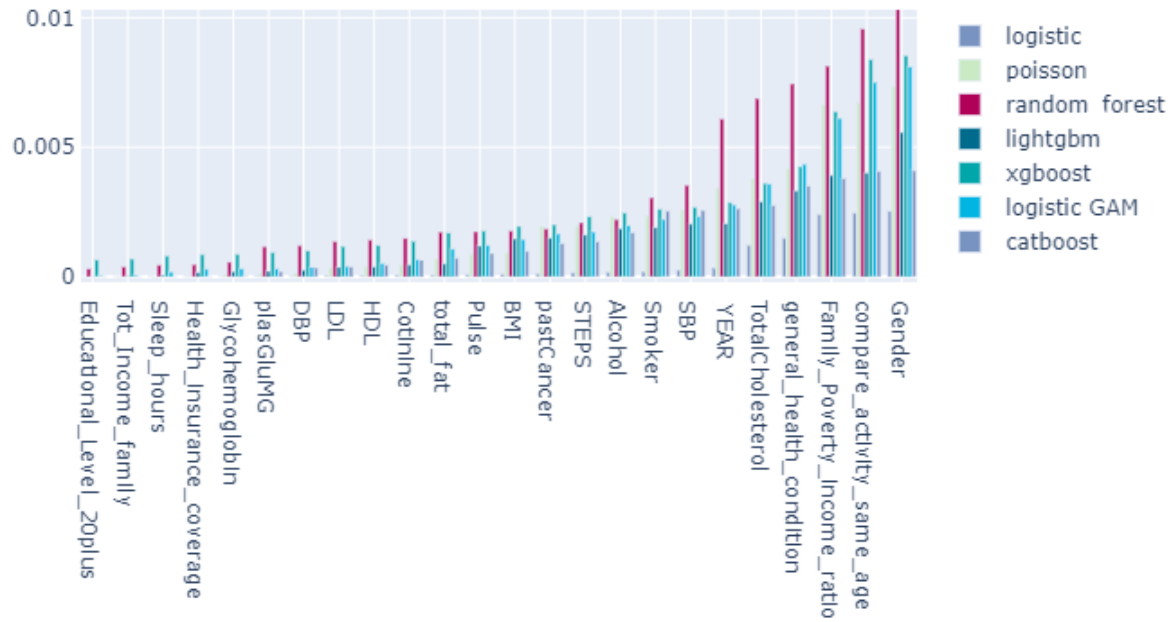


Figure 6.11: Variable Importance for discrete models

In both continuous and discrete models, methods based on the bagging of trees seem to be accentuating the importance of some risk factors compared to other models. Based on the figures, it seems that the highest impact of mortality comes from the social background, which may be seen through the importance of the family income and the general health conditions and habits, such as being active, consuming alcohol or cigarettes. This phenomenon has been highlighted in some

social science studies, such as the one published by Galea et al. [16], which concludes that the number of deaths attributable to social factors in the United States is comparable to the number attributed to pathophysiological causes.

```
from scor_survival import analysis
models = [coxnet, stree, coxtree, rsf, cox_xgboost, cox]
analysis.plot_var_importance(models, X, exposition, event)
```

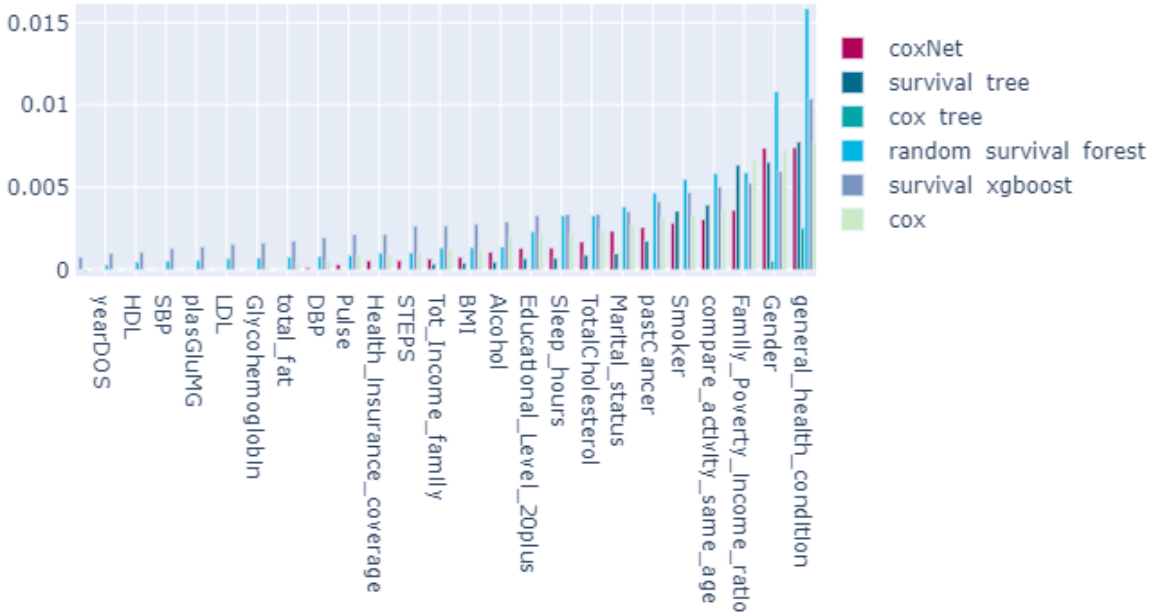


Figure 6.12: Variable Importance for continuous models

6.6 Models comparison

6.6.1 Performance metrics

ROC Curve

A comparison of multiple classifiers is usually straight-forward thanks to ROC curves, especially when no curves cross each other. Curves close to the perfect ROC curve have a better performance level than the ones close to the baseline. A classifier with the random performance level always shows a straight line from the origin to the top right corner. Two areas separated by this ROC curve indicates a simple estimation of the performance level.

The figure 6.13 highlights that all our models have quite good performance levels. As expected, all of them show a better performance on the train dataset rather than on the test one. However, interestingly the performance and so the ranking of our models depends on the data considered to evaluate them. If *XGBoost* model

seems to be the best one on the train set, it may be subject to over-fitting as it is no longer the case for the test set. Based on the test data set, *CatBoost* AUC curve is indeed the highest.

```
from scor_survival import analysis
models = [logistic, poisson, rf, lightgbm, xgboost, gam, catboost]
analysis.plot_roc_curve(models, X, exposition, event)
```

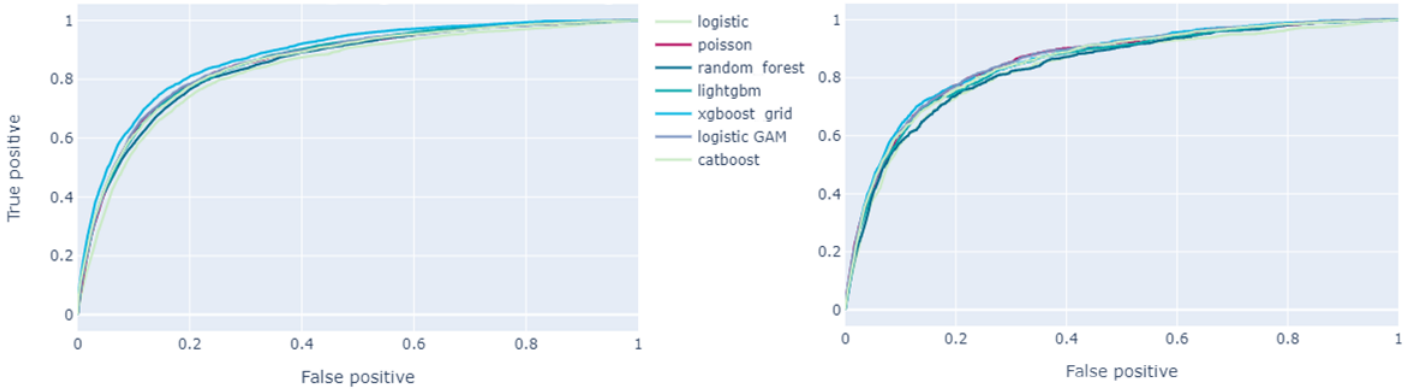


Figure 6.13: ROC Curve for the train (left) and test (right) datasets

C-Index

For continuous models, the C-index measure could be considered as an equivalent to the AUC as it gives as well insights on the risk ranking capacity of a model.

```
from scor_survival.models.continuous import model
model.ci(X, event, exposition)
```

Models	*C-Index Train*	*C-Index Test*
Cox	0.86	0.85
Cox-Net	0.86	0.85
Cox Tree	0.80	0.79
Cox XGBoost	0.89	0.85
Survival Tree	0.84	0.83
Random Survival Forest	0.85	0.83

Table 6.3: C-Index value

An important gap between the metric obtained on the train and test dataset indicates that the model is not perfectly calibrated. *Cox-XGBoost* model seems to be slightly over-fitted, even if the C-Index on the test set is still the best one. Based on this criterion a Cox-Net would be preferred as it seems to be more robust.

It may be interesting to compare the predictive capacity of our model on different subgroups rather than on the global level. As explained in the beginning, an insurer needs to produce consistent predictions for each category as the data used to calibrate the model is not necessarily representative of his portfolio.

```
from scor_survival import analysis
models = [logistic, poisson, rf, lightgbm, xgboost, gam, catboost]
analysis.plot_pred(variable, models, X, exposition, event)
```

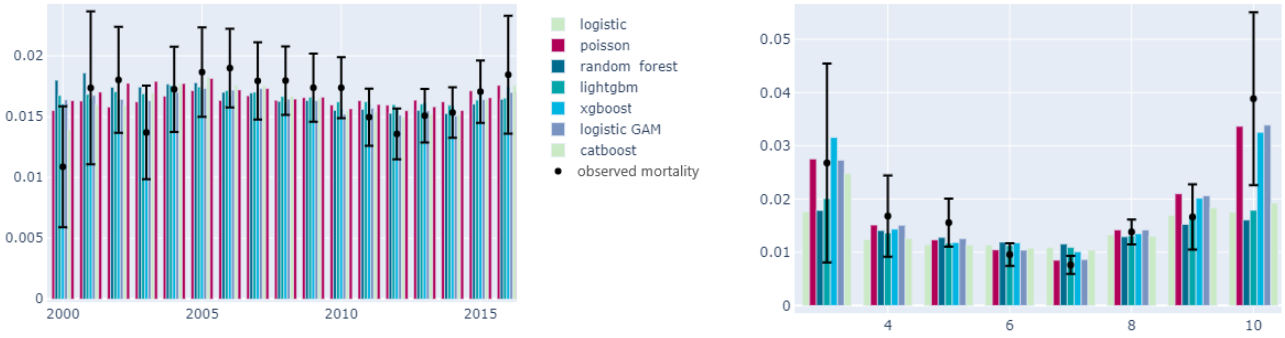


Figure 6.14: Mortality prediction by year and sleep hours

In other words, we expect that the prediction is indeed contained within the confidence interval of the observed mortality for each subgroup. According to the starting year of observation, it seems that it is the case for all our models. When focusing on sleep hours, it seems that some models, mainly the random forest and the lightgbm, may underestimate the observed mortality of the dataset.

However, the selection of a model may not only rely on the performance on some metrics as their performance highly depends on the dataset, on which they are evaluated. On a business mater, many other factors may also have an impact.

6.6.2 Convergence issue

For the models such as GAM or GLM, a gradient descent algorithm is used to estimate the parameters. If the objective function is not strictly convex, the optimization algorithm can descent into a local solution. If certain variables are correlated, the problem can be intractable, and the algorithm will return an error. A large number of features (and a high degree of freedom for GAM) increase the dimensions of the space of the parameters. The algorithm may struggle to find the optimal solution to the problem.

6.6.3 Edge effect issue

A model is efficient where the data have been observed. When extrapolating the data with a model, the user should be very careful. In survival analysis, when the survival dataset is converted into discrete data, the attended year is numerical information. If a GAM or a GLM can regress against this variable and propose an average trend, a classification tree will only locally approximate the mortality. Increasing the attended year observed in the dataset will produce very different behaviors: the decision tree will produce a flat shape.
This phenomenon is also visible on the Partial Dependence plot (cf Figure 6.10), as for tree-based methods, the impact of steps is flat on the edge.

6.6.4 Business constraints

The model calibration easiness, mainly the model sensitivity to hyper-parameters and the computation time, are important factors on the operational level. Models may be subject to frequent updates thus could be often re-calibrated. As seen above, Random Forest or XGBoost models require expensive computing resources to be very-well calibrated while regression models are generally straightforward.

Other important factors are the robustness and interpretability of the model. The regulation demands to insurers to be able to explain every decision and their determination of factor of risk. In this respect, a simple Binomial model could be preferred compared to the XGBoost algorithm. However, this could change in the future thanks to the development of interpretability methods.
More complex models, such as Random Forest or XGBoost, have better forecast performances as they can capture account variables interactions and non-linear patterns. But this comes with costs regarding the calibration easiness and interpretability. An insurer has to find the right balance between model precision and business constraints.

Based on these considerations, it seems that all our models have their strengths and weaknesses. On the NHANES dataset, it seems that the CatBoost model is the best agreement in terms of prediction performance, computation time, and calibration. Let indeed remind that a CatBoost model can deal with categorical feature without prior transformation features compared to all other models, which facilitates the process.
To give better insights into the comparison of the models, in the next chapter we will try to simulate the impact on an insurance market.

Chapter 7

Impact of modeling choice on a life insurance market

Survival models are used to forecast mortality rates for the insurance contract duration according to applicant characteristics. To illustrate the importance of a good mortality prediction for a life insurance company, we will consider the pricing of a policy on a competitive market.
Through a simulated competitive market between two insurers, we will give insights into the capacity of a model to price correctly simple products. For our purpose, we consider a simplified contract similar to products that can be found in the US market (cf Figure 7.1).

7.1 Pricing of life insurance policy

Given a premium π paid at time 0, the underwriting date, the insurer's beneficiary will earn C, fixed at \$ 100.000, in case of death in the following ten years. For simplicity matter, we will consider a constant interest rate $i = 1.5\%$ and we denote, v, the discount rate so that $v = \frac{1}{1+i}$. However, for business matters, an interest rate curve is considered.

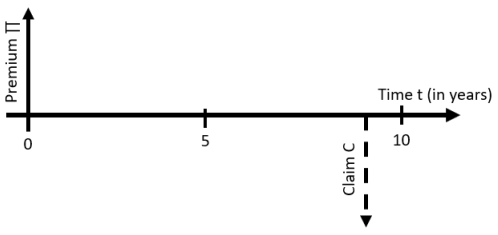


Figure 7.1: Life insurance contract

The pure premium of this life insurance contract is defined as the expected claim cost given one's characteristic, that is to say:

$$\pi(X) = E[Cv^T \mid X] = C \sum_{t=1}^{10} P(T = t \mid X) v^t$$
$$= C \sum_{t=1}^{10} S(t \mid X) h(t \mid X) \times v^t$$

Thus, the premium can be defined as a function of the survival curve: $\pi(X) = f_v(S(.|X))$. The function f_v is decreasing: The higher is the survival probability, the lower is the insurance price. The insurer is indeed less likely to pay the claim if the death probability is low.

From the previous formula, it becomes obvious that a binary classification (deaths VS alive) model is not enough to price life insurance contracts as we have to be able to predict survival at several periods. Also, insurers need to price precisely insurance products given life insurance purchaser characteristics.

To measure the performance of our models, we will compute the *Loss Ratio* on the test set. The Loss Ratio is a performance measure of the business result of an insurer. It is defined as the ratio of the total incurred claims on the total amount of premiums: $LR = \frac{Total\,Claims}{Total\,Premium}$. In practice, the loss ratio is computed at the end of a period to evaluate the result, thus the observed loss is considered. As we do not dispose of it in this fictive situation, the total claims amount will be estimated thanks to the previous formula where the death probability is replaced by the observed mortality.

Models	*Loss Ratio*
Binomial Regression	101.20%
Poisson Regression	100.81%
Random Forest	97.50%
LightGBM	97.99%
XGBoost	96.74%
logistic GAM	95.92%
Catboost	99.09%
Cox	100.58%
Cox XGBoost	98.73%
Cox-Net	100.56%
Cox Tree	106.19%
Survival Tree	101.98%
Random Survival Forest	103.59%

Table 7.1: Loss Ratio computation

A loss ratio below 100% indicates that the losses and thus the risk have been overestimated by the insurer. Considering that the discount rate has no effect, would lead to observe similar results as the global SMR. The Table 6.2 highlighted that all our discrete models overestimate mortality. However, for the Binomial and Poisson Regression models, the interpretation for the Loss Ratio is the opposite as we observe losses due to underestimation. This can be due to compensation effects through the aggregation. The loss ratio computation highlights the duration effect through the interest rate as more weight is given to recent years.

To compare the different premium estimations between the models, the figure below draws the mean premium for ages from twenty to eighty. Based on it, the age seems to impact the premium level exponentially in all models. The event of interest being death in the ten coming years is indeed highly dependant on one's age. The premium price increases almost until it reaches the sum insured. As for very old people surviving ten years, the insurance coverage period, is very unlikely, it means that the insurer is very likely to pay the claim.

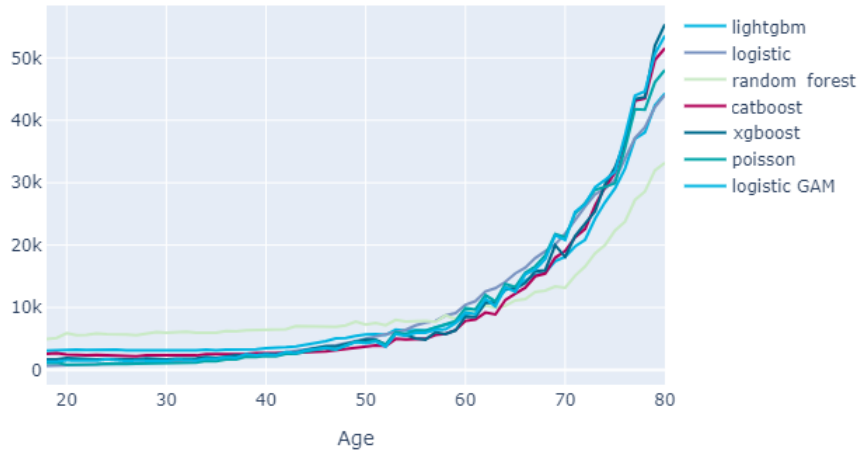


Figure 7.2: Premium evaluation based on individuals' age

7.2 Pricing game

Insurance companies were created before technological companies and have a deep knowledge of the risks with centuries of data accumulation. After decades of information system consolidation, the adoption of Machine Learning is key to analyze new data streams. An important matter for life insurers is to know whether using modern techniques for prediction based on a large number of variables and data is worthy. Insurers may face difficulties to assess the value of artificial intelligence

and underlying the complexity of the information systems.

To test the different impacts, in the following we will consider a pricing game on a market covered by two insurers A and B selling the life insurance coverage described previously. For easiness matters, each individual will select the insurer whose contract is the cheapest. However, in this situation, only the economic aspects are considered without any consideration for the behavior of each insured. For instance, if an applicant is asked to go to the doctor to get an insurance contract offer, even if it may reduce the price of its contract, the cost of opportunity may discourage her and she may choose the competitor. Formally, each individual's insurer choice can be written as:

$$A\,\mathbf{1}_{\{\pi_A \leq \pi_B\}} + B\,\mathbf{1}_{\{\pi_B < \pi_A\}}$$

7.2.1 Modeling choice

This first pricing game aims to test the impact of using a classical model versus a more advanced one. Insurer A is considered as a traditional insurer and its competitor B enters the market and price using a Machine Learning model. Both insurers will use all same amount of information available. However, they differ through their mortality modeling:

- Insurer A: Price the policy based on a *Binomial Regression*
- Insurer B: Price the policy based on a *CatBoost*

To have a fair game, we apply a small correction to the estimated premiums so that both insurers start with a Loss Ratio of 100%, which means perfect risk estimation. We will correct the price: $\pi_A^* = 1.012 \times \pi_A$ and $\pi_B^* = 0.9909 \times \pi_B$, where $\pi_\cdot$ is the premium estimated by the model.

Figure 7.3: Comparison of premium A and premium B

This highlights a division of the market between both insurers. Insurer B manages to obtain a loss ratio close to 100% while insurer A makes losses (cf Table 7.2). This experience highlights that all other things being equal considering a complex model seems to allow to gain market shares and thus beat a contestant with standard methods.

Insurer	*Loss Ratio*
A	143.9%
B	99.4%

Table 7.2: Loss Ratio Comparison

As proved in Appendix F, we cannot observe gains for both insurers, which means that if Insurer B succeeds to fit even better his premium estimation to its clients, Insurer A will still suffer from losses.

It is then interesting to compare the characteristics of the insured in portfolios A and B as it allows to understand the differences in losses ratios. It seems that Insurer B prices are more attractive for less risky individuals compared to Insurer A. The volume of premiums is larger for Insurer B compared to Insurer A for the categories which are less subject to mortality, such as being in a good health condition and not smoking.

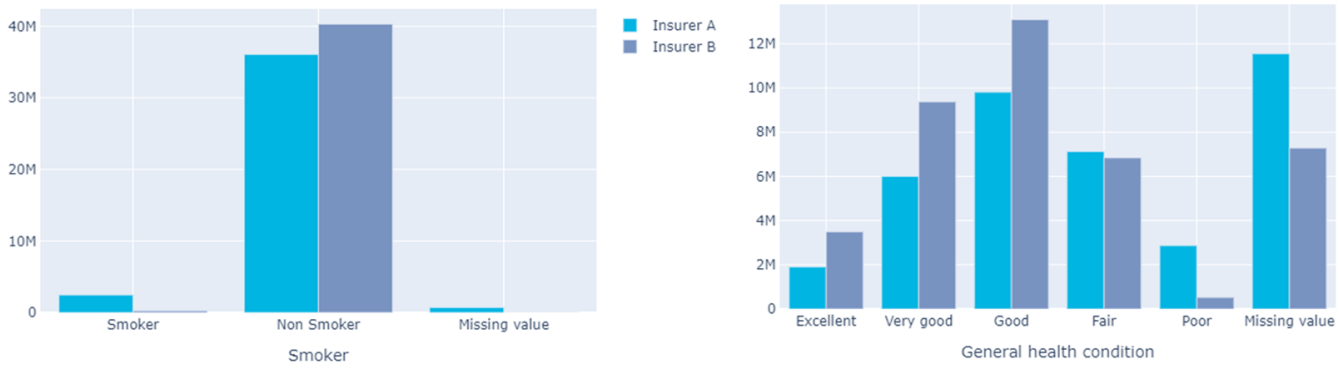


Figure 7.4: Global premium amount per category between insurers

7.2.2 Variable choice

Not only the techniques, the variables to include in the model may impact the result of an insurer. The acquisition cost of additional information on the applicant may be significant, thus an insurer may wonder whether there is a benefit of seeking them to price its products. Besides, the variable importance figure (cf Figure 6.11 and 6.12) highlighted that the social background was the most important

mortality factor. Based on our dataset, we focus on two main types of information levels:

- Declarative health information, which is free to collect
- Variable health information collected by a doctor or a health laboratory, which required medical analysis and thus are subject to fees assumed to be equal to $ 300.

As the previous steps highlighted that the *CatBoost* model seems to be the best one, both insurers on the market will model the mortality of their portfolio with a *CatBoost* model. However, they differ through their underwriting process:

- Insurer A: Collects only the information corresponding to the acquisition of free information, thus its model is calibrated using only the corresponding variables.
- Insurer B: Collects both types of information, thus its model is calibrated using all variables.

It is assumed that the medical fees are paid by the insured, which means that we will increase the estimated premium by $ 300 for each insured covered by B. If only the individuals that will choose B as an insurer will pay this extra $ 300 premium, insurer B has to pay the medical fees for every application. Concretely, we will have to add $ 300 for each individual in the total amount of loss.

Let n_B and n_A be the number of individuals for each insurer, $\pi_B(X)$ and $\pi_A(X)$ the premium and M_A and M_B be the price of the mortality observed in the portfolio. The loss ratio will thus be defined as :

$$LR_B = \frac{M_B + 300 \times (n_B + n_A)}{\sum_{i \in n_B} \pi_B(X_i) + 300 n_B} \text{ and } LR_A = \frac{M_A}{\sum_{i \in n_A} \pi_A(X_i)}$$

Insurer	Loss Ratio
A	142.6%
B	101.9%

Table 7.3: Loss Ratio Comparison

As we can see, considering additional information allows Insurer B to keep a Loss Ratio close to 100%. While Insurer A using less information makes large losses despite the lower underwriting fees.

An analysis of the portfolio characteristics of Insurer A and B is interesting to have a better understanding of the causes of the losses. Based on the number of insured by age, A seems to be more attractive for the younger individual on the market.
This result is at the first sight quite surprising as they are expecting to be less subject to mortality. A deeper analysis is needed to better explain the real deterioration of Insurer A's loss ratio.

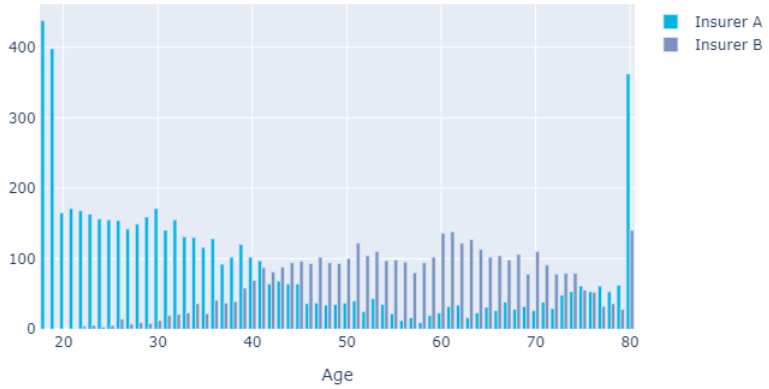


Figure 7.5: Age repartition between insurers

A first explanation can be deduced from the following figures. It seems that Insurer A is attracting the youngest individual as too much importance is given to one's age in the mortality prediction without any correction effect by the medical evidence. Indeed Insurer A does not distinguish for a given age, the impact of medical variables such as blood pressure or cholesterol amount. Based on this additional information, Insurer B succeeds to beat its competitor and to select the less risky individuals, i.e. the one with a larger amount of good cholesterol, HDL, and to avoid the one with a too high systolic blood pressure. On average, Insurer B capture indeed, for each age, individuals with higher HDL amount and lower SBP compared to Insurer A.

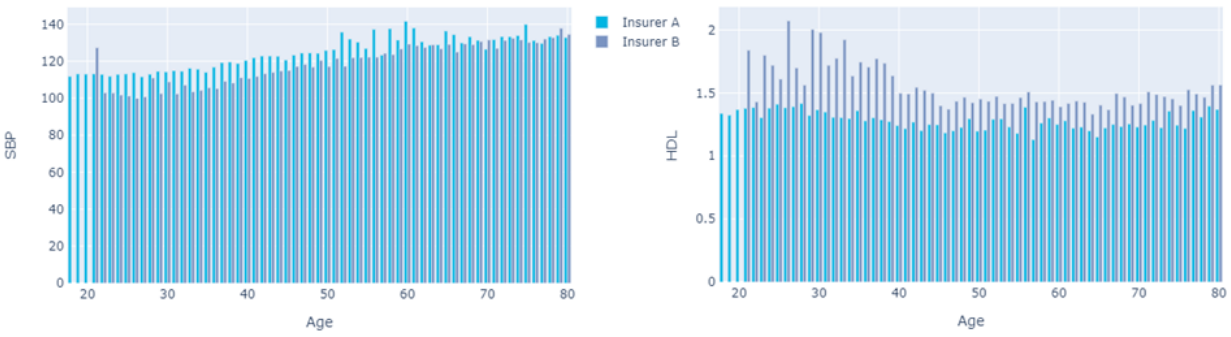


Figure 7.6: Comparison of medical information by age for each insurer

To confirm that the losses come from the asymmetries of information, we will focus on the impact of the split of the market on the portfolio of Insurer A for this two measures: the good cholesterol, HDL, amount and the Systolic Blood Pressure.
First, it is interesting to evaluate the bias in mortality estimation in different categories. On average on the global population, Insurer A manages to have a loss ratio of 100%. However, for a variable that is not used for the mortality modeling, we expect the SMR to vary randomly around 100%.
The figure below highlights that the mortality miss-estimation of Insurer A is larger than Insurer B. It is an expected result as the model retained by Insurer B uses this medical information. Insurer A underestimates the mortality of the riskiest individuals, that is to say, the one with extreme blood pressure or with a too low amount of HDL. In other words, for Cholesterol and Blood Pressure, Insurer A underestimates the mortality of *good risk* and overestimate the mortality risk of *bad risks*.

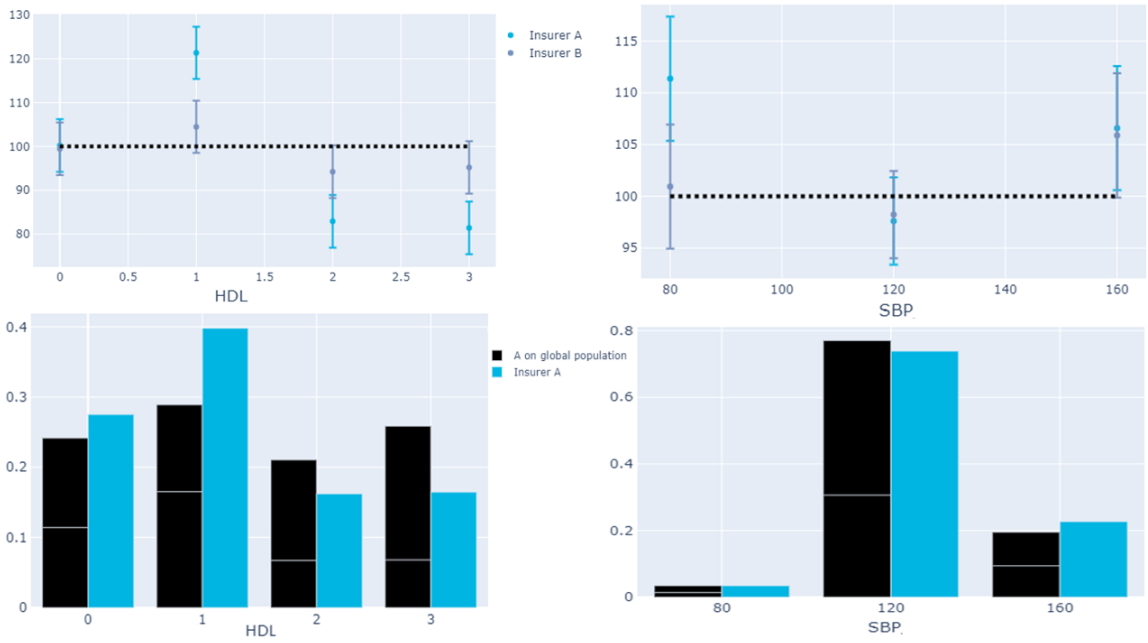


Figure 7.7: SMR and premium importance per medical profiles

Second, let's link the mortality error with the portfolio risk profile distribution. When competing with Insurer B, Insurer A's portfolio risk profile seems to have more weight towards categories where the mortality risk is underestimated. As categories, where the SMR is above 100%, are categories where the insurer is under-priced, the change in risk profile in Insurer A portfolio is causing losses and increase the Loss Ratio.
The competition with Insurer B implies that Insurer A is anti-selected. Indeed, it

looks like Blood Pressure and Cholesterol *good risk* are offered cheaper prices by Insurer B, and *bad risks* are offered cheaper prices by Insurer A. As *good risk* are not counterbalancing *bad risks*, the losses of Insurer A is increasing.

Based on the analysis of Insurer A's risk profile, it seems that the additional information collected by Insurer B is valuable as it prevents anti-selection. However, let's remind that we considered only the economic aspect on a closed market. Insurer B could still lose some good risk in detriment of Insurer A if the cost of opportunity was modeled.

This section highlighted the importance of good mortality risk modeling for an insurer. It indeed directly impacts its profit by causing losses in detriment to its competitor on the market. Our simulated market highlighted that not only a Machine Learning model implies gains when competing against a regression one. The information used to estimate mortality is also valuable regardless of the model choice.

To get more consistent results and confirm the real benefit of Machine Learning for mortality modeling, it would be better to conduct this experiment in an open market based on a representative insurer portfolio, in which new clients will ask for insurance products. Further work could be to resample the NHANES database to reflect the importance level in terms of population share in an insurer portfolio of the same profile's people as the particular sampled individual.

It would also be interesting to include the clients' behaviors. The study was indeed only based on the economic aspect without any consideration for the implied preferences such as the underwriting process, marketing action, or insurers' value.

Conclusion

This master thesis aimed at investigating to what extent Machine Learning algorithms manage to handle survival data correctly even if they have not initially been designed for survival modeling purposes. After an in-depth theoretical study of their possible adaptations for mortality risk assessment, all the methods have been implemented in a python library. This library aims to standardize several methods to facilitate and automate the study of the mortality of a portfolio.

Based on the open-source database NHANES, the application of a CatBoost model, which is an implementation of a gradient boosting able to handle categorical variables, leads to the best agreements between calibration ease, predictive performance, and computation speed.
This master thesis focused only on the Machine Learning adaptation for survival analysis. However, it would be interesting to include survival models based on deep learning techniques within the python library.

To give better insights into the importance of mortality modeling and the comparison of different models, we simulated a simplified life insurance market. The goal was to evaluate the economic impact of the competition between two insurers using different pricing strategies for the same life insurance product.
Based on our simplified example, we first highlighted that all other things being equal considering a Machine Learning model seems to allow to gain market shares and thus beat a contestant with regression methods.
Then, it seems that having the most information available is a real value as it prevents for anti-selection.
To get more consistent results and confirm the real benefit of Machine Learning for mortality modeling, it would be better to conduct this experiment in an open market based on a representative insurer portfolio, in which new clients will ask for insurance products. Further work could be to resample the NHANES database to reflect the importance level in terms of population share in an insurer portfolio. It would also be interesting to include clients' behaviors. The study was indeed only based on the economic aspect without any consideration for the implied preferences such as the underwriting process, marketing action, or insurers' value.

Bibliography

[1] O. Aalen. Nonparametric inference for a family of counting processes. *Annals of Statistics 6, 701–726*, 1978.

[2] American Heart Association. Understanding blood pressure readings. *American Heart Association*, 2011.

[3] American Heart Association. Cardiovascular disease and diabetes. *American Heart Association*, 2011.

[4] D. B. Atkinson and J. K. McGarry. Experience study calculations. *Society of Actuary*, 2016.

[5] I. Bou-Hamad, D. Larocque, and H. Ben-Ameur. A review of survival trees. *Statistics Surveys*, 2011.

[6] L. Breiman. Random forests. *Machine Learning, 45, 5–32*, 2001.

[7] L. Breiman, J. Friedman, R. Olshen, and C. Stone. Classification and regression trees. *Chapman Hall*, 1984.

[8] G. W. Brier. Verification of forecasts expressed in terms of probability. *Monthly Weather Review*, 1950.

[9] T. Chen and C. Guestrin. Xgboost: A scalable tree boosting system. *ArXiv e-prints*, 2016.

[10] D. R. Cox. Regression models and life tables. *Journal of the Royal Statistical Society*, 1972.

[11] D. Delcaillau, A. Ly, F. Vermet, and A. Papp. Interpretabilité des modèles : état des lieux des méthodes et application à l'assurance, 2020.

[12] A. V. Dorogush, V. Ershov, and A. Gulin. Catboost: gradient boosting with categorical features support. *ArXiv e-prints*, 2016.

[13] A. Fisher. All models are wrong but many are useful : Variable importance for black-box, proprietary, or misspecified prediction models, using model class reliance. *arXiv :1801.01489*, 2018.

[14] J. H. Friedman. Greedy function approximation: a gradient boosting machine. *Annals of Statistics*, 2001.

[15] J. H. Friedman. Stochastic gradient boosting. *Computational Statistics and Data Analysis*, 2002.

[16] S. Galea, M. Tracy, K.J. Hoggatt, C. Dimaggio, and A. Karpati. Estimated deaths attributable to social factors in the united states. *Am J Public Health*, 2011.

[17] E. Graf, C. Schmoor, W. Sauerbrei, and M. Schumacher. Assessment and comparison of prognostic classification schemes for survival data. *Statistics in Medicine*, 1999.

[18] F. Harrell, R. Califf, D. Pryor, K. Lee, and R. Rosati. Evaluating the yield of medical test. *American Medicine Association*, 1982.

[19] T. Hastie, R. Tibshirani, and J. Friedman. The elements of statistical learning. *Springer*, 2008.

[20] H. Ishwara, U. B. Kogalur, E. H. Blackstone, and M. S. Lauer. Random survival forests. *The Annals of Applied Statistics*, 2008.

[21] E. M. Kaplan and P. Meier. Nonparametric estimation from incomplete observations. *Journal of the American Statistical Association*, 1958.

[22] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T. Y. Liu. Lightgbm: A highly efficient gradient boosting decision tree. *Microsoft Research*, 2017.

[23] M. Le Blanc and J. Crowley. Relative risk trees for censored survival data. *Biometrics*, 1992.

[24] S. M. Lundberg and S. Lee. A unified approach to interpreting model predictions. *Advances in Neural Information Processing Systems*, 2017.

[25] C. Molnar. *Interpretable Machine Learning : A Guide for Making Black Box Models Explainable*. 2020.

[26] nanowiki. Random forest. *nanowiki*, 2016.

[27] W. Nelson. Theory and applications of hazard plotting for censored failure data. *Technometrics. 14: 945–965*, 1972.

[28] World Health Organization. Body mass index - bmi. *World Health Organization*, 2020.

[29] G. Ridgeway. The state of boosting. *Computing Science and Statistics*, 1999.

[30] G. Rodrìguez. Non-parametric estimation in survival models. *Spring*, 2001.

[31] M. Schmid, M. N. Wrigh, and A. Ziegler. On the use of harrell's c for clinical risk prediction via random survival forests. *Expert Systems with Applications*, 2016.

[32] Amicable Society. The charters, acts of parliament, and by-laws of the corporation of the amicable society for a perpetual assurance office. *Gilbert and Rivington*, 1854.

[33] R. Tibshirani, J. Friedman, T. Hastie, and N. Simon. Regularization paths for cox's proportional hazards model via coordinate descent. *Journal of Statistical Software*, 2011.

[34] H. Uno, T. Cai, and M.J. Pencina. On the c-statistics for evaluating overall adequacy of risk prediction procedures with censored survival data. *Statistical Medicine*, 2011.

[35] P. Wang, L. Yan, and R. K. Chandan. Machine learning for survival analysis: A survey. *arXiv:1708.04649*, 2017.

Annexe A

Cox likelihood

Cox model is defined as $h(t \mid X) = h_0(t) \times exp(\beta' X)$.
Let consider 3 individuals A, B, and C to give the intuition behind the formula. Considering that there is no tie and that $t_{(i)}$ is the ranked sequence of death time, for $i \in [1; 3]$.
Let R_i be the risk set, which is a set of indices of the subjects that are still alive just before $t_{(i)}$: $R_i = \{j : t_j \leq t_{(i)}\} = \{A, B, C\}$
The contribution to the likelihood will be the conditional probability :

$$P(\text{what happened at } t_{(i)} | \text{ one event occurs at } t_{(i)} \text{ and the information up to } t_{(i)})$$

Considering without loss of generality that A dies at $t_{(i)}$:

$$P_{t_i} = P(\text{A died; B,C survived} \mid \text{A,B,C in the risk set and one died})$$

$$= \frac{P(\text{A died; B,C survived})}{P(\text{A died; B,C survived}) + P(\text{B died; A,C survived}) + P(\text{C died; A,B survived})}$$

The events being independent using the previous notation:

$$\begin{aligned} P_A &= P(\text{A died; B,C survived}) \\ &= P(t_A = t_{(i)}, t_B > t_{(i)}, t_C > t_{(i)}) \\ &= f_A(t_{(i)}) \times S_B(t_{(i)}) \times S_C(t_{(i)}) \\ &= h_A(t_{(i)}) \times S_A(t_{(i)}) \times S_B(t_{(i)}) \times S_C(t_{(i)}) \\ &= h_A(t_{(i)}) \times C \text{ where } C = S_A(t_{(i)}) \times S_B(t_{(i)}) \times S_C(t_{(i)}) \end{aligned}$$

Injecting P_A in P_{t_i} and deducing the same formula for B and C :

$$P_{t_i} = \frac{h_A(t_{(i)}) \times C}{h_A(t_{(i)}) \times C + h_B(t_{(i)}) \times C + h_C(t_{(i)}) \times C}$$

$$= \frac{h_0(t_{(i)}) \times exp(\beta' X_A)}{h_0(t_{(i)}) \times exp(\beta' X_A) + h_0(t_{(i)}) \times exp(\beta' X_B) + h_0(t_{(i)}) \times exp(\beta' X_C)}$$

Finally we find :

$$P_{t_i} = \frac{exp(\beta' X_A)}{\sum_{j \in R_i} exp(\beta' X_j)}$$

The partial likelihood is then the multiplication over all the time of events:

$$L = \prod_{i=1}^{3} P_{t_i} = \prod_{i=1}^{3} \frac{exp(\beta' X_{j_{(i)}})}{\sum_{j \in R_i} exp(\beta' X_j)}$$

The general formula for n individuals is obtained with the same method.

Annexe B

UML Diagram

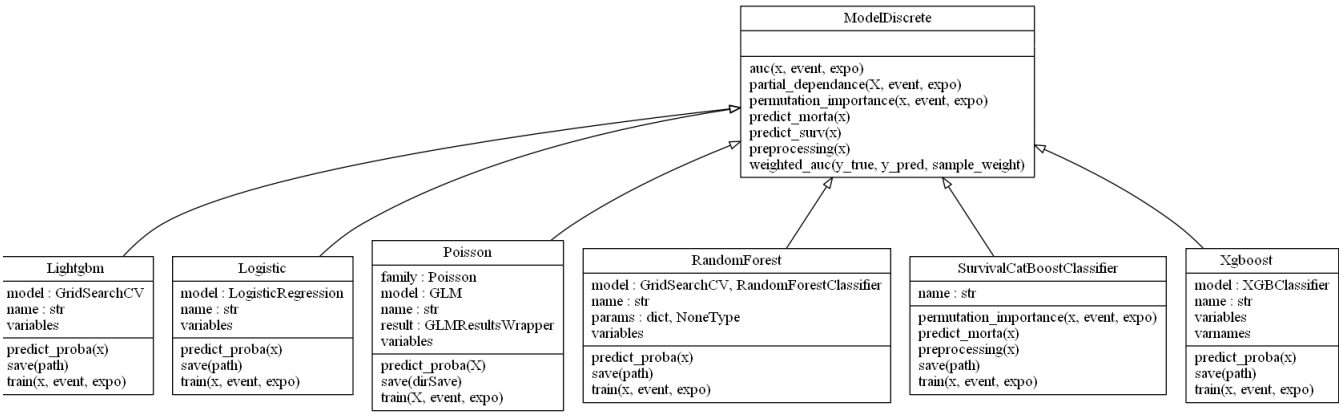


Figure B.1: Discrete model classes

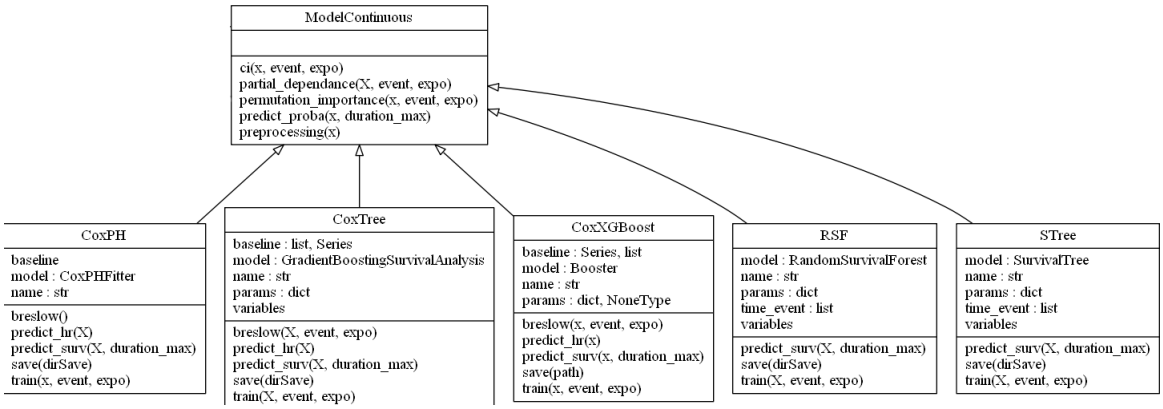


Figure B.2: Time-to-event model classes

Annexe C

Logrank statistics

The logrank test is the most widely used test to compare survival curves. It is a non-parametric test. This test can be generalized for any number of groups but for the survival tree purpose only the two groups version is used.

In this test, we state the null hypothesis : $H_0 : S_A(t) = S_B(t)\ \forall t$

Within each group, we compute for each observed time i, the number of expected death :

$$e_{Ai} = \frac{n_{Ai} d_i}{n_i} \text{ and } e_{Bi} = \frac{n_{Bi} d_i}{n_i}$$

We finally aggregate for each time to obtain the total number of expected death $E_{\cdot} = \sum_i e_{\cdot i}$ and the observed number of death $O_{\cdot} = \sum_i d_{\cdot i}$.

	Group A	**Group B**	**Total**
Death	d_{Ai}	d_{Bi}	d_i
Survivorship	$n_{Ai} - d_{Ai}$	$n_{Bi} - d_{Bi}$	$n_i - d_i$
Total	n_{Ai}	n_{Bi}	n_i

Table C.1: Notation used at time i

Finally the logrank statistics is given by $X^2 = \frac{(O_A - E_A)^2}{E_A} + \frac{(O_B - E_B)^2}{E_B}$
Under H_0 this statistics follows a chi-square distribution with one degree of freedom, we can then compute the p-value.

As the whole survival theory, the validity of this test relies on the independence assumption between the observed event and the censoring.
The main limit is the difficulty to reveal the difference of mortality between two groups when their survival curve cross. The power of the test is indeed maximum for proportional curves.

Annexe D

Gini index

Let consider the ten following individuals within a node m to understand the intuition behind the Gini index:

	1	***2***	***3***	***4***	***5***	***6***	***7***	***8***	***9***	***10***
δ	1	1	1	1	1	0	0	0	0	0
ei	1	1	1	1	1	1	1	0.2	0.8	0.5

Using the formulas, we have : $\hat{q}_m = \frac{5}{8.5} = 0.59$ and $G = 2\hat{q}_m(1 - \hat{q}_m) = 0.48$.

First split

Considering a perfect split where we have on the left side the first 5 individuals and the 5 others on the right side : $\hat{q}_l = 1$ and $\hat{q}_r = 0$ thus $G_l = 0$ and $G_r = 0$
The information gain is thus $I = 0.48$

Second split

Considering a variable that splits on the left side the dead and censored individuals and the survivor on the right : $\hat{q}_l = 0.77$ and $\hat{q}_r = 0$ thus $G_l = 0.36$ and $G_r = 0$
The information gain is thus $I = 0.20$

Third split

Considering a variable that splits on the left side the dead and alive individuals and the censored on the right : $\hat{q}_l = 0.71$ and $\hat{q}_r = 0$ thus $G_l = 0.40$ and $G_r = 0$
The information gain is thus $I = 0.19$

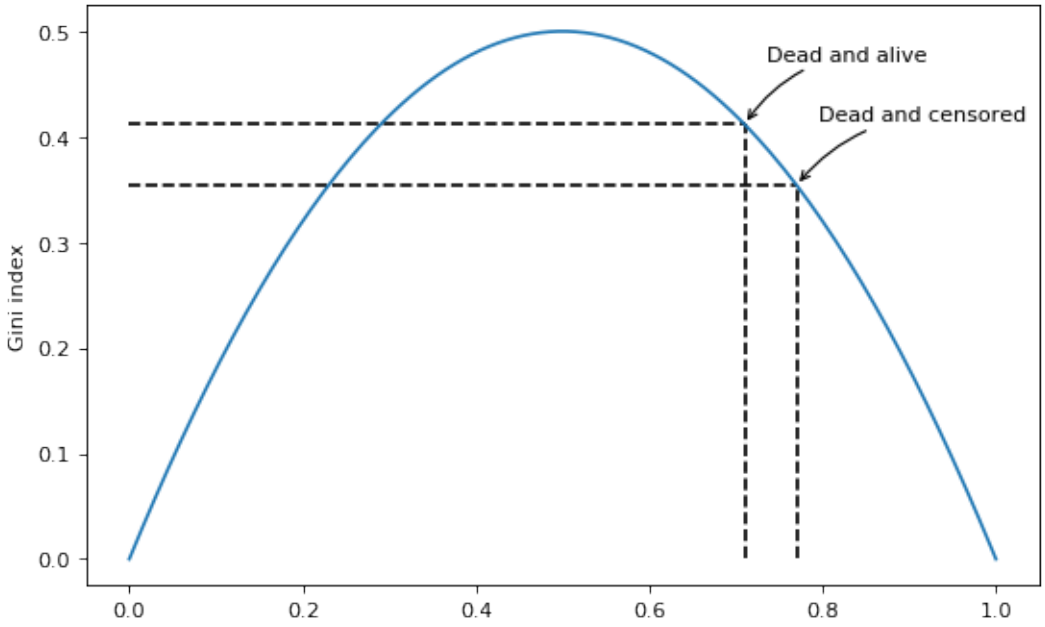


Figure D.1: Impurity curve

Thus the information gain is indeed the maximum for the first split as it creates two pure leaves.

As shown with the second and the third splits, the censor is indeed taken into account thanks to the exposure. Without adjusting the probability with the initial exposure, both splits would have been equivalent (as it becomes equivalent to consider all exposures equal to one and thus there is no difference between censored and alive people for the Gini index).

However, in our case mixing censored and dead observations within a leaf is better than mixing alive and dead ones. This can be explained as the Gini index increases if we reduce the global exposure keeping the number of dead observations equals and as the dominant label of the leaf. We can deduce the same conclusion on the ascendant side of the Gini curve by interchanging dead and alive observations.

Censored and dead VS alive splits are indeed preferable compared to alive and dead VS censored. For censored people, we know for sure that they survive until a certain point of time. However, the probability of dying before the end of the observation period is not null compared to the survivors. Thus the mistake is less certain.

Annexe E

NHANES variables dictionary

Target variables

- **Surv**: followed-up time in month
- **Death**: mortality status at the end of the observation

Categorical variables

- **Gender**: male or female
- **Alcohol**: regular alcohol consumer?
- **Smoker**: regular smoker?
- **Marital status**: marital status, in 6 categories : *Never Married/ Married/ Living With Partner/ Separated/ Divorced/ Widowed*
- **Educational Level 20plus**: education level for individual more than 20 years old, in 5 categories : *High School Grad GED or equivalent/ Some College or AA Degree/ College Graduate or Above/ 9-11th Grade/ Less Than 9th Grade*
- **Health insurance coverage**: covered by a health insurance?
- **general health condition**: self-evaluation of one's health status, in 6 categories : *Good/ Very Good/ Fair/ Excellent/ Poor*
- **Tot Income family**: total family income, on income amount category
- **pastCancer**: has cancer history?
- **compare activity same age**: self-evaluation for activity level compared with peers, in 3 categories *As Active/ More Active/ Less Active*

- **Sleep hours**: daily average sleep hours, on hours category

Numerical variables

- **Age**: one's age when he is interviewed
- **BMI**: body mass index
- **STEPS**: The step count recorded by the physical activity monitor
- **Family Poverty income ratio**: ratio of family income to poverty threshold
- **Glycohemoglobin**: glycohemoglobin %
- **plasGluMG**: fasting glucose (mg/dL)
- **SBP**: systolic blood pressure
- **TotalCholesterol**: total cholesterol (mmol/L)
- **LDL**: LDL cholesterol (mmol/L)
- **Cotinine**: cotinine by serum test (ng/mL)
- **Pulse**: 60s heart pulse
- **HDL**: HDL cholesterol (mmol/L)
- **total fat**: total fat (gm)
- **DBP**: diastolic blood pressure

Annexe F

Pricing Game theory

We tried to demonstrate that the empirical results are indeed the expected one. Formally, an insurance premium is defined as:

$$\pi(X_i) = \mathbb{E}[C_i|X_i]$$

We supposed that the population is divided between the two insurers only on the premium price basis, that is to say they will be covered by the cheapest one. The two subsets of individuals are thus defined as:

$$\Omega_A = \left\{i|\pi^A(X_i) < \pi^B(X_i)\right\} \text{ and } \Omega_B = \left\{i|\pi^B(X_i) < \pi^A(X_i)\right\} = \bar{\Omega}_A$$

Then let $Y_i^{\cdot}$ be the result of an insurer for individual i. The result is defined as the difference between the premium and the claim:

$$Y_i^A = \pi^A(X_i) - C_i \text{ and } Y_i^B = \pi^B(X_i) - C_i$$

Based on the formulas, we can deduce that:

$$Y_i^A = Y_i^B + \pi^A(X_i) - \pi^B(X_i)$$

Noticing that by construction :

$$\sum_{i\in\Omega} \mathbb{E}[Y_i^B] = \sum_{i\in\Omega} \mathbb{E}[\pi^B(X_i)] - \mathbb{E}[C_i] = 0$$

If we compute the global result of insurer A, $\mathbb{E}[\sum_{i\in\Omega_A} Y_i^A]$, we can derive a first result:

$$\begin{aligned}
\mathbb{E}[\sum_{i\in\Omega_A} Y_i^A] &= \sum_{i\in\Omega_A} \mathbb{E}[Y_i^A] \\
&= \sum_{i\in\Omega_A} \mathbb{E}[Y_i^B] + \sum_{i\in\Omega_A} \pi^A(X_i) - \pi^B(X_i) \\
&= \sum_{i\in\Omega} \mathbb{E}[Y_i^B] - \sum_{i\in\Omega_B} \mathbb{E}[Y_i^B] + \sum_{i\in\Omega_A} \pi^A(X_i) - \pi^B(X_i) \\
&= -\sum_{i\in\Omega_B} \mathbb{E}[Y_i^B] + \sum_{i\in\Omega_A} \pi^A(X_i) - \pi^B(X_i)
\end{aligned}$$

On the set Ω_A, we have $\pi^A(X_i) \leq \pi^B(X_i)$, which means that the second sum is negative. We can thus derive the following:

$$\mathbb{E}[\sum_{i\in\Omega_A} Y_i^A] + \sum_{i\in\Omega_B} \mathbb{E}[Y_i^B] \leq 0$$

From this equation, one should understand that partitioning individuals between two insurers implies either losses for both insurers, either gains for one and losses for the second one. It is indeed impossible to get gains for both.