Mémoire présenté devant le jury de l'EURIA en vue de l'obtention du Diplôme d'Actuaire EURIA et de l'admission à l'Institut des Actuaires

le 27 Septembre 2019

Par : ROLLAND Louis
Titre : Claims segmentation: A Machine Learning approach

Confidentialité: Non

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

Membre présent du jury de l'Institut des Actuaires :
CORRÉGÉ Pierre
SAID Khalil
GUÉLOU Sonia
GIBELLO Hélène
Signature:

Entreprise:
SCOR SE
Signature:

Membres présents du jury de l'EURIA :
VERMET Franck
LENCA Phillipe (IMT)

Directeur de mémoire en entreprise:
BOLLACHE Vincent
Signature:

Invité:
-
Signature:

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

Signature du responsable entreprise:

Signature du candidat:

EURIA
EURo Institut
d'Actuariat

6, avenue le Gorgeu
CS 93837
29238 Brest Cedex 3

T +33 (0)2 98 01 66 55
euria@univ-brest.fr

Abstract

The traditional methods used in Reserving are aggregated methods, based on a segmentation of contracts: in general, for each segment, a different pattern is fitted to the developments of the previous years to predict the claim amount of the next years.

Using a segmentation reduces the volatility of the estimation by aggregating claims with similar patterns together. Under the Solvency II regulation, this segmentation must build homogeneous risk groups.

The goal of this study is to challenge the current segmentation of claims used by SCOR P&C. This segmentation is based on expert judgements, contains technical limitations and is built on criteria regarding the contracts features. To challenge this segmentation, Machine Learning methods were used on a database containing variables describing both the contracts and the claims' features. The missing values of the database were filled using advanced techniques based on correlations between the variables.

The challenging segmentations are based on:

- Decision Tree regressors: fitted to durations of Incurred or Paid cash-flows, with each leaf corresponding to a class;
- Clustering methods: based on variables and/or indicators.

The sturdiness of these segmentations was tested using more advanced algorithms such as Neural Networks, Random forests or Gradient Boosting.

The results were compared to the current actuarial segmentation based on two criteria:

- Homogeneity: studied by measuring the intra-variance of indicators, regarding Incurred and Paid cash-flows, for sets of classes;
- Quality of prediction: using Chain-Ladder method to estimate the cash-flow N not knowing the last development period. As well as predicting IBNR and Reserves for the last 3 or 5 years of development on a database of closed-claims.

Once the most predictive segmentation selected, its composition will be analysed to study the relevance of its use and its limits.

Keywords:
Actuarial Sciences, General Insurance, Claims Reserving, Machine Learning, Patterns of Development, Chain-Ladder, Facultative contracts, Sklearn, Neural Networks, Decision Tree, Random Forest, Gradient Boosting, Dimension Reduction, MCA, Missing Values, missMDA, Clustering, K-means, K-modes, K-prototypes

Résumé

Les méthodes usuelles de provisionnement non-vie sont des méthodes agrégées, reposant sur une segmentation des contrats. Pour chaque segment, un développement est ajusté sur les développements historiques pour estimer le montant de sinistres à venir.

L'utilisation d'une segmentation permet de réduire la variance de l'estimation en regroupant des sinistres similaires ensemble. Sous Solvabilité II, cette segmentation doit former des groupes de risques homogènes.

L'objectif de cette étude est de challenger la segmentation des sinistres actuellement utilisée chez SCOR P&C, basée sur avis d'experts, contenant des limitations techniques et construite sur des critères relatifs aux contrats.

Pour ce faire, des méthodes de Machine Learning ont été utilisées sur une base de données contenant des variables au niveau contrat et au niveau sinistre. Le traitement des valeurs manquantes a été effectué en utilisant une méthode basée sur les corrélations entre les variables.

Les segmentations testées ont eté construites à partir de :

- Arbres de décision ajusté sur les duration de *cash-flows* d'*Incurred* ou de *Paid*: une classe sera attribuée à chaque feuille ;
- Clusterings : sur les variables et/ou indicateurs.

Pour mesurer la prédictibilité de ces classes, des algorithmes plus élaborés (Réseaux de neurones, Random Forest, Gradient Boosting) ont été utilisés.

Les résultats des nouvelles segmentations ont été comparés à la segmentation actuelle sur 2 critères :

- Homogénéité des classes : en mesurant l'erreur de prédiction sur les durations de cash-flows;
- Qualité de la prédiction : en estimant le montant de l'année N, en enlevant la dernière diagonale, et en estimant le montant d'IBNR et de Réserves, pour les trois ou cinq dernières années de développement.

Une fois la segmentation la plus prédictive sélectionnée, sa composition sera étudiée pour mesurer la cohérence de son éventuelle utilisation et ses limites.

Mots clefs :
Sciences actuarielles, Assurance Non-Vie, Provisionnement Non-Vie, Machine Learning, Type de développement, Chain-Ladder, Réassurance Facultative, Sklearn, Réseaux de neurones, Arbres de Décision, Forets Aléatoires, Gradient Boosting, Réduction de dimension, MCA, valeurs manquantes, missMDA, Clustering, K-means, K-modes, K-prototypes

Acknowledgement

First and foremost, I must address my sincere gratitude toward my supervisor, Mr. Vincent BOLLACHE, for his expert's advice, his support and commitment throughout this project and for giving me the opportunity to work on such an interesting subject, bringing innovative solutions to actuarial problematics. As well as my academic supervisor Mr. Romain LAILY for his time and his opinion on the conduct of this study.

I would like to address my appreciation toward Mr. Frederic SCHWACH, for his time and his implication in this project, as well as his experience regarding the writing of the actuarial thesis. I am grateful toward the whole P&C Reserving Project Team, for sharing with me their experts' points of view on this project, regarding its limits and its implications.

I would like to thank Mr. Hicham DAHER, IT Data Scientist, for his advice and expertise regarding algorithms theory and use, and for giving me the IT resources necessary for the leading of this project.

I would also like to thank SCOR's many specialists with whom I had the chance to discuss. Especially towards:

- Ms. Fatima NAJI, Pricing actuary and Data Scientist, for sharing with me her previous work on aggregating claims regarding their development ratios, my cluster on indicators are inspired by her work, and while her method did not work on my dataset, it gave me a new perspective on how to aggregated claims;
- Mr. Jean-Claude RAZAFINDRAKOTO, for his overall ideas on how to aggregate claims, as well as his business-application mindset and how to analyse the composition of the classes.

I am grateful toward Mr. Pape TOP Head of UK P&C Reserving and Ms. Saijal PATEL, P&C Reserving Actuary, for the information they gave me about UK P&C Reserving, which helped me get more background and interpretability regarding the results of my study.

I also thank EURIA and especially its director Mr. Franck VERMET, for the many discussions we've had and for sharing his knowledge on Machine Learning with me.

Finally, I thank my family and my friends, for their continuous love and support.

Executive summary

Keywords:
Actuarial Sciences, General Insurance, Claims reserving, Machine Learning, Patterns of development, Chain-Ladder, Facultative contracts, Sklearn, Neural Networks, Decision Tree, Random Forest, Gradient Boosting, dimensions reduction, MCA, missing values, missMDA, Clustering, K-means, K-modes, K-prototypes

Goal of the study

The traditional methods used in Reserving are aggregated methods, based on a segmentation of contracts. The amounts estimated depend on the segmentation: a segmentation aggregating claims with similar development patterns will enhance the prediction quality. However, a segmentation mixing claims with different patterns could compute amount not corresponding to the underlying risk: a short-tail claim assigned to a long-tail class will have its underlying risk overestimated, and vice versa.

The goal of this study is to find ways to create homogeneous classes of claims, for which the underlying will be correctly estimated. Using such a segmentation will enhance the prediction quality, as described hereafter:

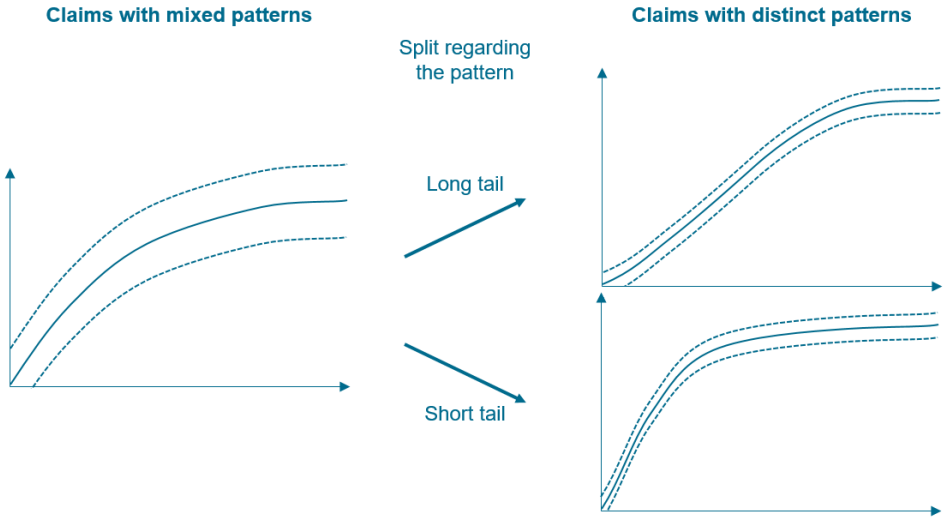


Figure 1: Impact of the segmentation regarding the estimation error

The current segmentation used by SCOR is partly based on expert judgements, contains technical limitations and is built on criteria regarding the contracts features. This study will analyse the relevance of this segmentation, by building new segmentations using Machine Learning algorithms on a database containing claim-level features.

Framework of the study

Building the development patterns indicators

In order to describe the different patterns of development, the undiscounted durations for Incurred and Paid cash-flows have been computed. The underlying hypothesis behind these indicators is that a small duration will reflect a short pattern of development (short tail), while higher values for that indicator will reflect longer pattern of development (long tail). Therefore, the models will revolve around building homogeneous classes regarding these indicators. Hereunder are examples to illustrate how the durations can describe the patterns:

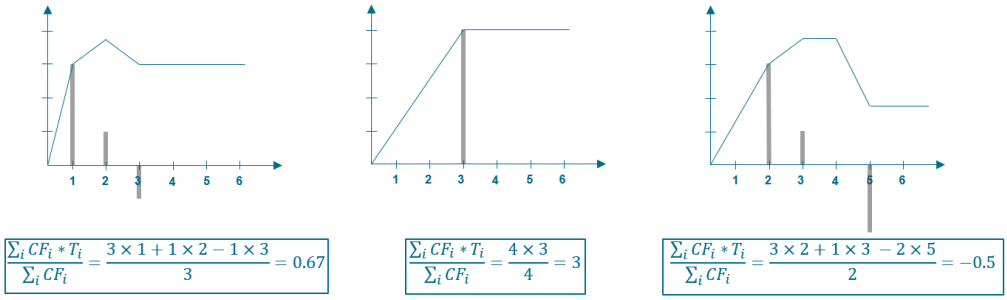


Figure 2: Impact of the development patterns on the values of the indicators

Presentation of the study database

A database composed of Facultatives contracts as for 4Q18, was extracted from SCOR central accounting system, gathering contracts and claims features. This database contains information related to the contract itself, such as type of policy, scope of business or type of cover, and also information related to the claims, such as the country in which the claim occurred or the type of event that led to the claim.

The database contain nearly 450,000 lines and 20 variables, as well as two tables of development for both Incurred and Paid.

The database quality was challenged on the three criteria put forward by the Solvency II legislation: appropriateness, completeness and accuracy.

The process was firstly applied to the Fire database, composed of the claims in the Fire line of business, representing 50% of the whole database, being short-tail and quite homogeneous. Once established, the process was generalized to the whole database.

Improving the quality of the study database

The database contained missing values that needed to be filled in order to use Machine Learning algorithms. The correlations between the variables were used in the missing values management process, via a method based on dimension reduction. Principal component analysis (PCA[1]) have been applied to the dataset variables, and the missing values were projected on the principal axes of the PCA.

Due to technical limitations, optimizing this process was necessary. To do so, clusters were used to reduce the size of the database and an other measure of the quality was put in practice, based on the study of the probabilities of belonging in each category for a missing value.

[1] A special kind of PCA (MCA) have been used in order to also consider categorical variables.

Building the challenging segmentations

The challenging segmentations were built following two different kinds of algorithms:

- Tree-based segmentations (supervised approach);
- Cluster-based segmentations (unsupervised approach);

Tree-based segmentations

The tree-based segmentations are built from a decision tree regressor fitted to an indicator (undiscounted duration of Incurred or Paid cash-flows). Once the regression tree built, each leaf will be considered as a class. The predicted value for each class is the mean of the indicator of the class. Hereunder is presented the process of the value assignment:

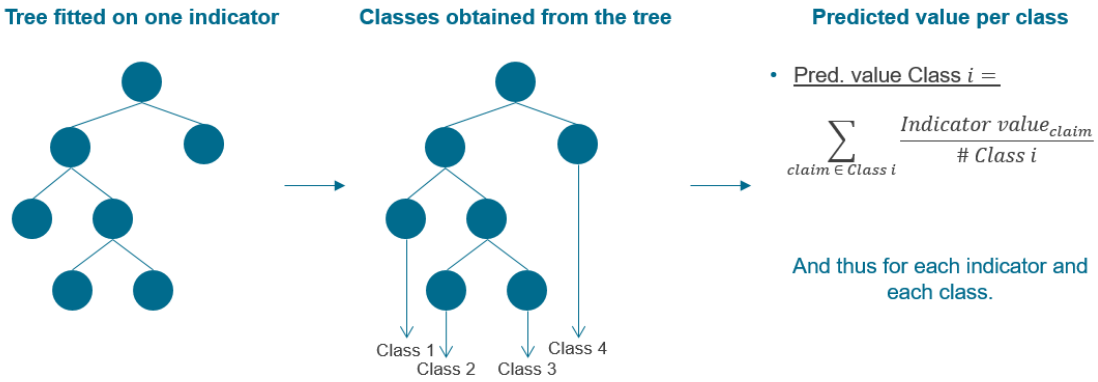


Figure 3: Value assignment process for tree-based segmentations

Cluster-based segmentations

The cluster-based segmentations are built using either k-means (if all variables are numerical) or k-prototype (if variables are both numerical and categorical). The predicted value assignment process follows the same principles as the classes of the tree-based segmentations. The predicted value of each class, for each indicator, will be the mean of that indicator, for the claims composing the class. Hereunder is presented the process of the value assignment:

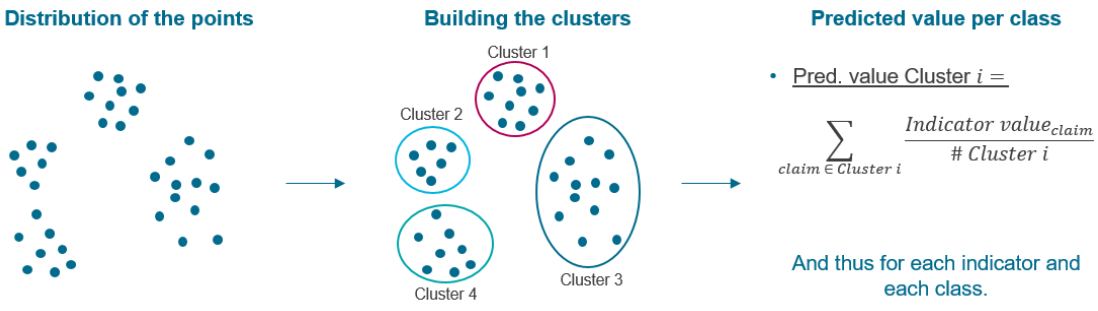


Figure 4: Value assignment process for cluster-based segmentations

In this study, three different kinds of cluster-based segmentations were studied, each based on a different selection of variables:

- Clusters based on the indicators only: this segmentation will aggregate claims having similar values for the indicator.
- Clusters based on variables only: this segmentation will build homogeneous classes regarding the variables.
- Clusters based on the variables and indicators: these segmentations provide a compromise between the two previous segmentations, by considering both the variables and the values of their indicators in the clustering process.

Selecting of the segmentations per method

In order to choose the best segmentation per method (trees fitted to Incurred/Paid, clusters based on indicators/variables/indicators & variables), it is necessary to compare these segmentations between each other. To allow a comparability between the segmentations based on supervised and unsupervised algorithms, a specific methodology had to be put in place. This methodology is composed of the following steps:

1. Reducing the size of the database to consider biases: since the claims with the most recent underwriting years had a bias in their indicator values (as they did not have enough time to be fully developed), they had to be dropped;
2. Random split of the database between a training set and a test set (70%/30%);
3. Fitting a segmentation process to the training set for both supervised and unsupervised methods;
4. Classifying the test set using either:
   - The decision tree used to build the classes for the tree-based segmentations;
   - Classification algorithms for the cluster-based segmentations (Decision Trees, Random Forests, Gradient Boosting and Neural Networks). The classification algorithms are fitted to the training set to predict the classes of the test set.
5. Once the whole test set is allocated to the classes of the new segmentations, the mean of the indicator for that class was assigned as the values for that class (and thus for each indicator).
6. A value being now available for each indicator and each class, the Mean Squared Error (MSE) of prediction between each segmentation could be computed and compared with each other.
7. After the study of the MSE, one set of parameters per method was kept. This choice was done by looking at the reduce of MSE and the complexity of the models (elbow method, study of the improvement, trade-off between improvement and complexity/number of classes).

The claims with the most recent underwriting years, that were not considered when building the classes (due to a bias in their indicators' value) needed to be reclassified in the newly built classes. While for tree-based segmentations, classes are clearly defined by sets of rules that compose the tree; the cluster-based segmentations are not so clearly defined. Therefore, classification algorithms were used to reclassify the most recent claims in the cluster-based segmentations.

Study of the segmentations

The prediction quality of the segmentations has been compared considering:

- The 2018 cash-flow of Incurred and Paid;
- IBNR / Reserves.

All these estimations were done using Chain-Ladder, as it highlights the natural development of the classes without experts judgements.

2018 cash-flow predictions

The most recent year of development (2018) has been taken out of the triangle and estimated using Chain-Ladder, as illustrated hereafter:

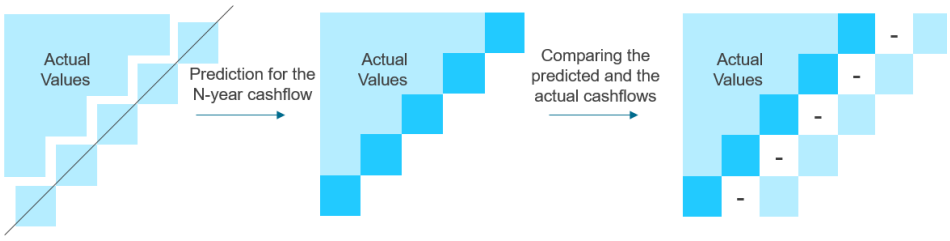


Figure 5: Predicting the 2018 cash-flow

Since the measure is only measuring the cash-flow of the next year, it is quite volatile.

IBNR and Reserves predictions

The last 3 or 5 years of development have been taken out of the triangles, then the Ultimate, of a database composed of closed claims, was estimated using Chain-Ladder. The Ultimate amounts being known, since the database used is made of fully developed claims, it is possible to know the IBNR/Reserves for 3/5 years prior. This process is illustrated hereafter:

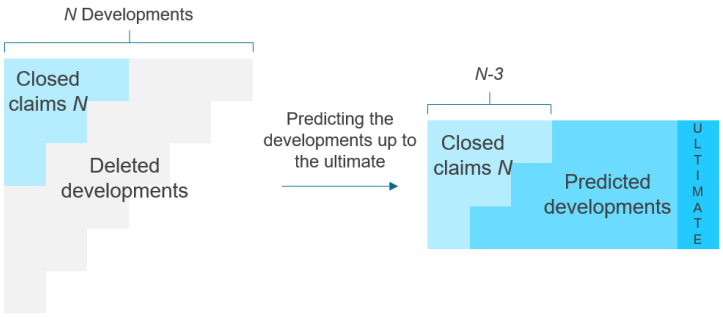


Figure 6: Prediction des IBNR / Reserves

Analysis of the results

The quality of prediction was quantified via two indicators:

- $Delta\ \% = \frac{Amount_{Predicted} - Amount_{Actual}}{Amount_{Actual}}$
  Measures how close to the real value the total prediction is, but errors can compensate;

- $|Error|/|CF| = \frac{\sum_{j \in UWY} |CF^j_{Predicted} - CF^j_{Actual}|}{\sum_{j \in UWY} |CF^j_{Actual}|}$
  With $CF^j_{\cdot}$ amount for the j-th underwriting year.
  Measures the sum of errors for each underwriting year, divided by the sum of the absolute values of each actual cash-flow.

These measures were used for each prediction: Cash-flow of 2018 for Incurred and Paid, IBNR/Reserves predictions with three or five years taken out.

On the Fire database, the prediction using the aggregated pattern (aggregating all the claims in the same class) was the best for almost all the indicators. This was due to the homogeneity of the Fire database and to the fact that the segmentations had too few claims per class to correctly evaluate the factors of development. The Actuarial Segment was outperformed by most of the segmentations; and especially by the segmentation based on clusters of variables.

A study of the composition of the segmentation based on clusters was then conducted. After analysing the composition of the classes, some risk-profiles could be associated to each class.

The process was then extended to the whole database. Due to the heterogeneity of the database, and the larger number of claims, the estimations for the whole database were more precise, and the Actuarial Segment was performing better than on the Fire Database. Even though the Actuarial segment was better on the whole database, it was still outperformed by the best segmentation (based on a tree fitted to Incurred durations).

The challenge of SCOR's current segmentation highlighted some of its flaws, especially regarding the volatility of small classes' developments. However, it is worth noting that these volatilities were only occurring due to the fact that only the natural developments were considered, with no experts judgement. Indeed, the business practices often contains corrections to reduce the volatility and experts judgement regarding the section of factors of development.

Conclusion

The goal of this study was to see how SCOR's current segmentation, containing experts judgements would compare with segmentations based on Machine Learning.

The study was conducted with a scientific approach. By justifying every choices and by limiting the experts judgements, in order to have a fair comparison between the segmentations based on algorithms and the current one.

To build the segmentations, indicators describing the development patterns were built, enabling a quantification of the homogeneity of the classes.

The segmentations with the best homogeneities regarding these indicators were then selected, and the natural developments of their classes were compared to those of SCOR's current segmentation.

The challenging segmentations could outperformed the current segmentation's. Therefore, the segmentations created are composed of more homogeneous classes, regarding the natural developments.

The whole process of the building of the classes was efficient as the classes obtained formed homogeneous groups and had more precise predictions than the Actuarial Segment. The use of indicators was justified, indeed, different kinds of patterns have been distinguished.

The challenge of SCOR's current segmentation also highlighted some of its flaws and ways to improve it, as well as the fragility of the natural pattern of development.

This study takes advantages of Machine Learning to compute efficient models, while still being interpretable and keeping in mind the business applications and its constraints: interpretability, transparency and control over the models.

Note de synthèse

Mots clefs :
Sciences actuarielles, Assurance Non-Vie, Provisionnement Non-Vie, Machine Learning, Type de développement, Chain-Ladder, Reassurance facultative, Sklearn, Réseaux de neurones, Arbres de Décision, Forets Aléatoires, Gradient Boosting, Réduction de dimension, MCA, valeurs manquantes, missMDA, Clustering, K-means, K-modes, K-prototypes

Objectif de l'étude

Les principales méthodes de provisionnement sont des méthodes agrégées, basées sur une segmentation des contrats. L'estimation des réserves est impactée par la qualité de la segmentation : une segmentation regroupant des sinistres ayant le même type de développement permettra d'augmenter la précision de prédiction; en revanche, une segmentation regroupant des sinistres avec différents types de dévelopements aura une estimation erronée des risques sous-jacents : un sinistre à developpement court placé dans une segmentation à développement long verra son risque sous-jacent sur-estimé, et inversement.

L'objectif de cette étude est d'élaborer une méthode permettant la construction de classes de sinistres homogènes. L'utilisation d'une telle segmentation permettra d'améliorer l'évaluation du risque sous-jacent, comme illustré ci-dessous:

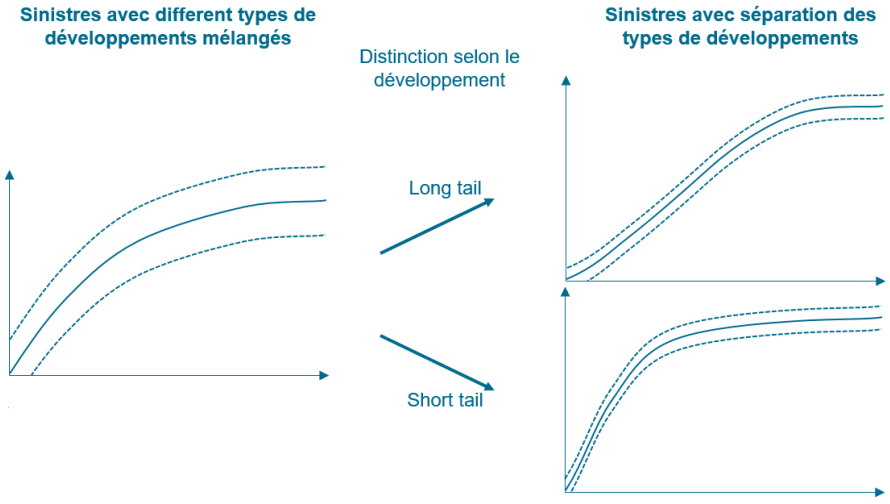


Figure 7: Impact d'une segmentation sur l'erreur d'estimation

La segmentation utilisée chez SCOR est basée en partie sur avis d'experts et sur des critères relatifs au contrat. Le but de cette étude est de challenger cette segmentation, en construisant de nouvelles segmentations à partir d'algorithmes de Machine Learning, appliqués sur une base de caractéristiques au niveau contrat et sinistre.

Cadre de l'étude

Mise en place d'indicateurs

Afin de décrire les types de développement, les durations non-actualisées pour les cash-flows *Incurred* et *Paid* ont été etudiées. L'hypothèse de la mise en place de ces indicateurs est qu'une duration faible traduira d'un dévelopement plus court (*short tail*) et qu'une duration longue d'un développement plus long (*long tail*). Ainsi, la construction des modèles se fera autour du regroupement de sinistres ayant des valeurs similaires pour ces indicateurs. Les exemples ci-dessous illustrent l'impact du développement sur les valeurs des indicateurs :

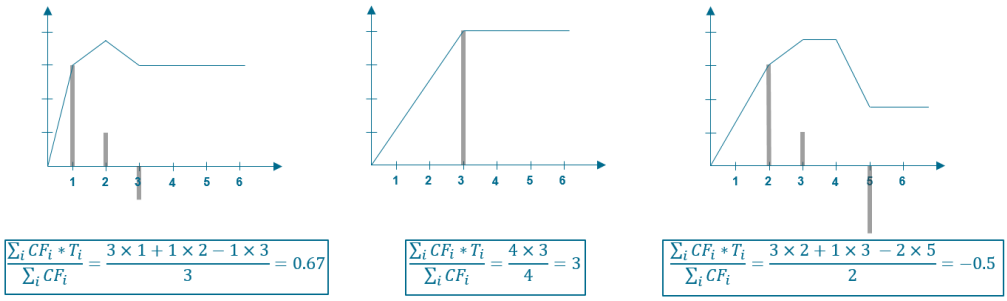


Figure 8: Impact du type de développement sur la valeur des indicateurs

Présentation de la base d'étude

La base de données est composée de contrats *Facultatives*, extraits du système interne de comptabilisation de SCOR, en vue quatrième trimestre 2018 (4Q18). Cette base de données contient des informations au niveau contrat: le type de police, le domaine d'activité ou la nature du contrat; ainsi que des caractéristiques au niveau sinistre: pays dans lequel le sinistre a eu lieu, type d'évenement qui a conduit au sinistre. la base d'étude contient près de 450,000 sinistres et 20 variables. Ainsi que deux tableau de développements pour *Incurred* et *Paid*.

La qualité de la base de données a été vérifiée sur les 3 critères renseignés dans Solvabilité II, vérifiant: le caractère approprié, l'exhaustivité et l'exactitude de la base de données.

Une première mise en place de processus nécessaire à la construction des prototypes a été ajusté sur la base contenant seulement les contrats de la *line of business* "*Fire*". Cette sous base contient 50% de la base complète et est connue comme étant *short-tail*. Une fois les processus calibrés et une analyse de la cohérence des resultats effectuée, l'ensemble de la méthodologie a été appliquée au reste de la base.

Amélioration de la qualité de la base de l'étude

La base contenant des données manquantes sur certaines variables, un remplissage de ces données est nécessaire pour appliquer nos algorithmes de Machine Learning. Pour prendre en compte les corrélations entre les autres variables, une methode basée sur la réduction des dimensions à été utilisée. Des analyses en composantes principales (ACPs[2]) ont été appliquées aux variables, et les données manquantes ont étés projetées sur les axes principaux de ces ACPs. Afin de contounrner des limitations techniques, des clusters ont été utilisés pour réduire la taille de la base de données. Une nouvelle mesure de la qualité du remplissage des valeurs manquantes a du être mise en place, basée sur l'étude des probabilités d'appartenance aux différentes catégories.

[2]Un certain type d'ACP, des MCAs, ont été utilisées pour prendre en compte les variables catégoriques.

Construction des prototypes de segmentations

Les segmentations sont construites selon deux types de méthodes:

- Les méthodes basées sur arbres de décision (approche supervisée);
- Les méthodes basées sur des clusters (approche non-supervisée).

Segmentations basées sur les arbres de décision

Les segmentations basées sur des arbres de décision sont formées à partir d'arbres de regression, ajustés sur un indicateur (duration non-actualisée d'*Incurred* ou de *Paid*). À partir des feuilles obtenues pour ces arbres de régression, une classe sera attribuée pour chaque feuille. Afin de déterminer une valeur de prédiction pour chaque indicateur, la moyenne des individus pour chaque classe sera considerée comme la valeur de prédiction.

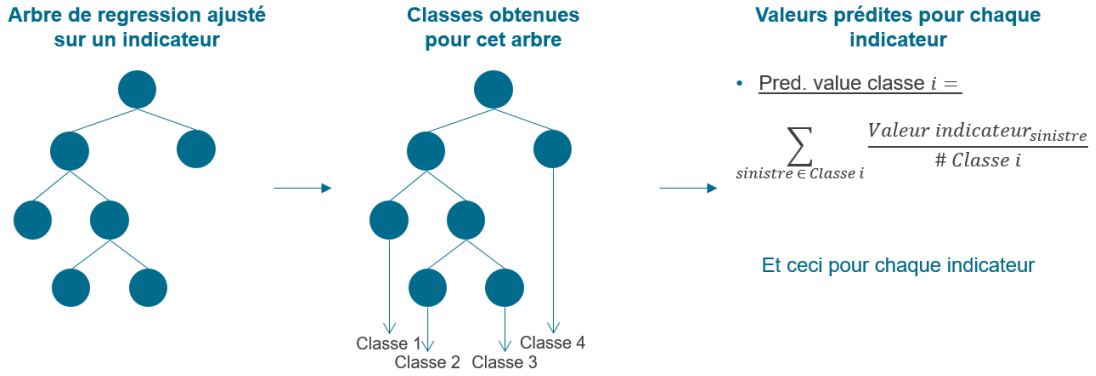


Figure 9: Attribution des valeurs de prédictions pour les arbres de decision

Segmentations basées sur les clusters

Les segmentations basées sur clusters sont formées à partir de k-means (si les données sont numériques) et k-prototypes (si les données sont à la fois numériques et catégorique). Le principe d'attribution des valeurs de prédictions pour chaque indicateur est similaire à celui utilisé pour les arbres : la valeur moyenne est attribuée pour chaque classe, et ce pour chaque indicateur.

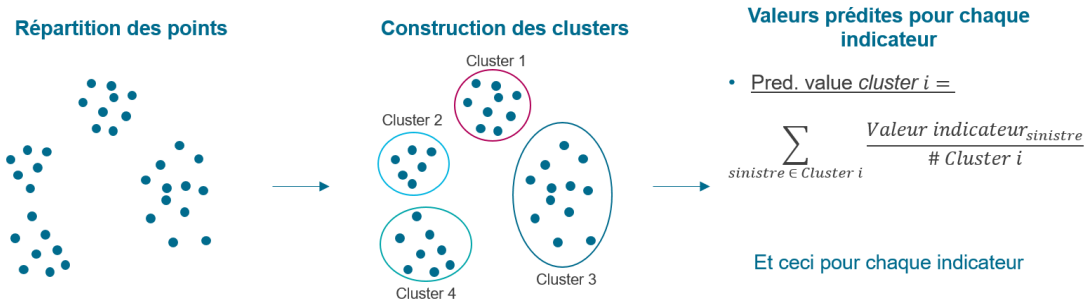


Figure 10: Attribution des valeurs de prédictions pour les clusters

Cette étude s'est concentrée sur trois modèles de segmentations sur clusters, chaque modèle étant basée sur une sélection différente des variables utilisées pour les clusters :

- Clusters basés sur les indicateurs seulement : ces segmentations vont créer différentes classes de valeurs pour les indicateurs, et allouer les sinistres dans les classes correspondantes ;
- Clusters basés sur les variables, ces segmentations vont créer des groupes homogènes au regard des variables ;
- Clusters basés sur les variables et les indicateurs, ces segmentations proposent un compromis entre les deux methodes précédentes en considérant à la fois les valeurs des indicateurs et les variables descriptives lors de la construction des classes.

Processus de sélections des segmentations par type de modèle

Afin de sélectionner une segmentation par modèle (arbres sur *incurred*/*paid*, clusters sur indicateurs/variables/indicateurs & variables), il est nécessaire de pouvoir comparer ces segmentations.

Pour permettre une comparabilité entre les modèles supervisés (basés sur les arbres) et non supervisés (basés sur clusters), une méthodologie spéciale a du être mise en place. Cette méthodologie est composée des étapes suivantes :

1. Réduction de la base de données pour prendre en considération un biais dans les valeurs des durations. En effet, pour les années les plus récentes (contrats souscrits en 2016/2017), les sinistres n'ont pas suffisement de temps pour se développer ;
2. Division aléatoire de la base en une base d'apprentissage et une base de test (70/30%) ;
3. Ajustement d'une segmentation sur la base d'apprentissage pour les modèles supervisés et non-supervisés ;
4. Classification de la base de test dans les nouvelles classes via :
   - Pour les segmentations basées sur les arbres : la réutilisation de l'arbre servant à construire les classes ;
   - Pour les segmentations basées sur les clusters : des algorithmes de classification ajusté sur la base d'apprentissage (arbres de décision/foréts aléatoires, réseaux de neurones et gradient boosting).
5. Une fois la base de test classée, les valeurs des indicateurs par classe ont été attribuées comme la moyenne de chaque indicateur par classe;
6. Une valeur par classe pour chaque indicateur étant maintenant disponibles, les intra-variances (erreurs moyennes quadratiques) peuvent-être estimées et comparer entres les différentes segmentations ;
7. Une fois les erreur de prédiction obtenues pour different paramètres, une segmentation par modèle a été choisie. Ce choix est basé sur l'étude de la réduction d'erreur et sur la compléxité des modèles (méthode du coude, compromis entre amélioration de la prédiction et complexité/nombre de classes).

Les sinistres les plus récents n'étant pas été pris en compte lors de la construction des classes, il est nécessaire de les reclasser dans les nouvelles classes. Les segmentation basées sur les arbres sont clairement definies par les règles composant les arbres; ce qui n'est pas le cas pour les segmentations basées sur les clusters. Ainsi, la classification des nouveaux sinistres dans les classes basées sur les clusters a été effectuée par des algorithmes de classification.

Étude de la prédictabilité des meilleurs segmentations

La qualité de prédiction des segmentations a été estimée en prédisant :

- Les *cash-flows* d'*Incurred* et de *Paid* ;
- Les montants d'IBNR et de Réserves.

Toutes les prédictions ont été estimées par Chain-Ladder, car cette méthode met en évidence les développements naturels de chaque segmentation, sans jugement d'expert.

Étude de la prédictabilité du *cash-flow* d'*Incurred* et de *Paid*

La dernière année de développement été enlévée du triangle, puis été évaluée par Chain-Ladder :

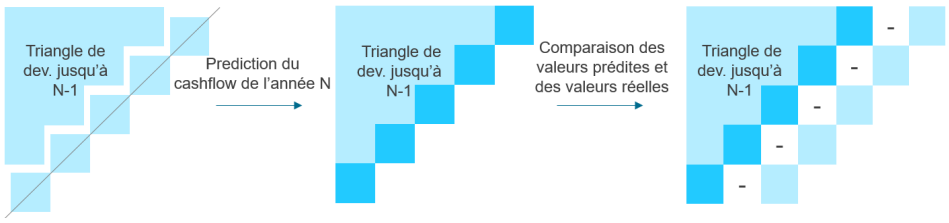


Figure 11: Prédiction du *cash-flow* de l'annee 2018

Étude de la prédictabilité d'IBNR et de Réserves

Les montant d'IBNR et de Réserves : seuls les sinistres clos ont été gardés. Pour ces sinistres, les 3 dernières années développement ont été enlevées et les montant d'IBNR/Réserves ont été estimés par Chain-Ladder.

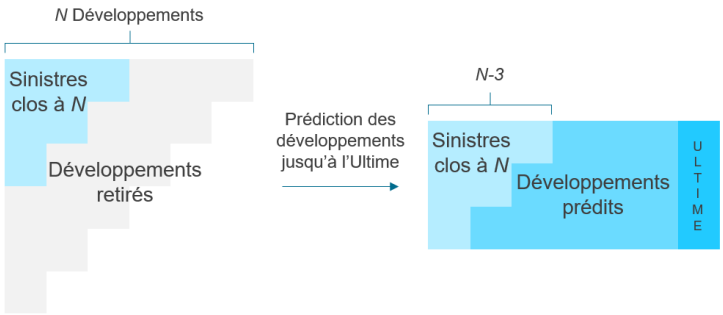


Figure 12: Prediction des IBNR / Reserves

Analyse des résultats

La qualité de prédiction a été mesurée selon deux indicateurs :

- $Delta\ \% = \frac{Montant_{Predit} - Montant_{Reel}}{Montant_{Reel}}$
  Mesure la distance entre la prédiction et le montant réel au total, des erreurs peuvent se compenser ;

- $|Erreur|/|CF| = \frac{\sum_{j \in annee\ de\ souscription} |CF^j_{Predit} - CF^j_{Reel}|}{\sum_{j \in annee\ de\ souscription} |CF^j_{Reel}|}$
  Avec $CF^j_{\cdot}$ montant pour l'année de souscription j.
  Mesure la somme des erreurs pour chaque année de souscription, divisée par la somme des *cash-flows* de toutes les années.

Ces mesures ont été utilisées pour chaque prédiction : estimations des *cash-flows* de 2018, prédictions pour IBNR/Réserves en enlevant 3 ou 5 années de développements.

Cette analyse a d'abord été effectuée sur la base de données *Fire*, représentant 50% de la base complète, réputée homogène et ayant un développement court. Sur cette base, les prédictions obtenues par la segmentation aggrégée (tous les sinistres dans la même classe) était la meilleure sur presque tous les indicateurs. Ceci étant du à l'homogénéité de la base *Fire*, et au fait que certaines classes des segmentations avait trop peu de sinistres pour pouvoir correctement estimer les facteurs de développement.

Sur la base *Fire*, la segmentation actuarielle avait de moins bonnes performances que la plupart des autres segmentations.

Afin de mieux percevoir les classes construites par les algorithmes, une analyse de la composition de chaque classe a été effectuée. Après analyse, des profils de risques distincts ont pu être identifiés pour chaque classe, ayant chacun des caractéristiques propres.

La généralisation du processus montre de meilleurs résultats, dus à l'hétérogénéité des nouvelles *lines of business* et au plus grand nombre de sinistres présent dans la base, permettant une meilleure estimation des facteurs de développement (et ce même pour les plus petites classes).

De maniere générale, les prédictions sont plus précises sur la base complète, et spécialement pour la segmentation actuarielle.

Le challenge de la segmentation actuarielle a mis en évidence certaines de ses faiblesses, notamment la volatilité de l'estimation des développements naturels pour les plus petites classes. Cependant, il est important de préciser que cette volatilité n'est présente que dans notre cadre d'étude. En effet, en pratique, des corrections sont mises en place limiter la volatilité des facteurs de développement, notamment via la non-sélection des ratios en dehors d'un certain intervalle.

Conclusion

Le but de cette étude est de challenger la segmentation actuelle, contenant des jugements d'experts, en la comparant avec des segmentations issues d'algorithmes de Machine Learning.

Pour construire les segmentations, des indicateurs décrivant le developpement des sinistres ont été mis en place. Afin de mesurer l'homogénéité au niveau de développements.

Les segmentations ayant les meilleurs homogénéités ont été sélectionnées. Leurs développements naturels ont été utilisés pour prédire des montants de provisions, et comparés à ceux obtenus pour la segmentation actuelle.

Les nouvelles segmentations obtenues sont capables de prédire plus précisement que la segmentation actuelle. Et sont donc plus homogènes au niveau des développements.

Cette étude a été construite en ayant une approche la plus scientifique possible. En justifiant les différents choix, afin de limiter les jugements d'experts, pour permettre une comparaison juste entre les résultat des algorithmes et la segmentation actuelle.

La démarche de la construction des classes est cohérente. En effet, les classes obtenues forment des groupes homogènes et ont une meilleure homogénéité au niveau des développement naturels. Les segmentations arrivent bien à séparer différents types de développement dans différentes classes.

Cette étude a mis en évidence certains défauts de la segmentation actuarielle, et souligné la fragilité des développements naturels.

Finalement, cette étude propose une méthode se basant sur du Machine Learning pour produire des modèles performants, tout en considérant l'importance des contraintes métier : interprétabilité, transparence et contrôle des modèles.

Contents

Introduction **1**

I Reserving and Machine Learning theory **4**

1 Traditional methods in Reserving **5**
1.1 Claim life cycle . . . 5
1.2 The Chain-Ladder method . . . 6
1.2.1 Underlying hypotheses . . . 6
1.2.2 Construction of the Chain-Ladder method . . . 6
1.3 Why is a segmentation needed? . . . 9
1.3.1 Regulatory aspects of the building of homogeneous risk groups . . . 11

2 Machine Learning **12**
2.1 Supervised Learning . . . 12
2.1.1 Artificial Neural Networks . . . 13
2.1.2 Decision Tree . . . 14
2.1.3 Random Forest . . . 15
2.1.4 Gradient Boosting . . . 17
2.2 Unsupervised Leaning . . . 19
2.2.1 Dimension reduction algorithms . . . 19
2.2.2 Clustering algorithms . . . 21

II Presentation of the study database **26**

3 Overview of the database **27**
3.1 Regulatory aspects of the building of the database . . . 28
3.1.1 GDPR . . . 28
3.1.2 Solvency II . . . 28
3.2 Original variables . . . 29
3.3 Selection of variables . . . 30
3.4 Constructed variables . . . 30
3.5 Indicators . . . 32

4 Missing value management **35**
4.1 Filling missing data using dimension reduction . . . 35
4.2 Application on the Fire database . . . 37
4.2.1 MCA for Sob & Top . . . 38
4.2.2 MCA for Country_Claim . . . 39
4.2.3 Risk_Nature & Claim_Cause . . . 43
4.3 Quality of the study database . . . 44

III Challenging the existing segmentation **45**

5 Building the new segmentations **46**
5.1 Selection of the Fire sub-database . . . 46
5.2 Construction of the Actuarial Segmentation . . . 48
5.3 Framework of building the challenging segmentations . . . 48
5.4 Segmentations based on Decision Trees . . . 50
5.4.1 Study of the MSE for trees fitted to TF_Incurred_N . . . 51
5.4.2 Study of the MSE for trees fitted to TF_Paid_N . . . 52
5.4.3 Conclusion of the MSE analysis for the tree-based models . . . 53
5.5 Segmentations based on Clustering . . . 54
5.5.1 Study of the MSE for clusters based on the indicators . . . 55
5.5.2 Study of the MSE for clusters based on variables . . . 56
5.5.3 Study of the MSE for cluster-based on both the variables and indicators . . . 57
5.6 Conclusion on the choice of the models . . . 58

6 Study of the quality of prediction for the segmentations **61**
6.1 Predicting the 2018 cash-flow . . . 62
6.1.1 Classifying the most recent years: 2016 and 2017 . . . 62
6.1.2 Distribution of the recent claims in the new segmentations . . . 64
6.1.3 Comparing the results . . . 65
6.2 Predicting the IBNR / Reserves amounts . . . 66
6.2.1 Predicting the IBNR / Reserves not knowing the last 3 years of development . . . 67
6.2.2 Predicting the IBNR / Reserves not knowing the last 5 years of development . . . 69
6.3 Selection of the best segmentation . . . 70
6.3.1 Presentation of the best segmentation . . . 70
6.3.2 Comparing the development patterns . . . 71
6.3.3 Risk profile of the classes . . . 73
6.3.4 Conclusion on the best segmentation . . . 76
6.4 Conclusion on the segmentation challenge . . . 77

7 Generalization on the whole database **78**
7.1 Prediction results on the whole database 80
7.2 Conclusion on the whole database 83

Conclusion **84**

Bibliography **85**

A Mathematical framework for Machine Learning algorithms **87**
A.1 Artificial Neural Network 87
A.2 Decision Tree 90
A.3 Random Forest 93

B Variables analysis of the Fire Database **95**
B.1 Analysis for numerical data 95
B.1.1 Densities and cumulative distribution for numerical variables . . . 95
B.1.2 Correlations between numerical variables 99
B.2 Analysis for categorical data 101
B.2.1 Frequencies of categories for categorical variables 101
B.2.2 Correlation between categorical variables 108
B.3 Analysis for indicators 111
B.3.1 Densities and cumulative distributions for indicators 111
B.4 Note on the reliability of the Indicators 115

C Analysis of the best segmentation **117**
C.1 Study of the numerical variables 117
C.2 Study of the categorical variables 119

Introduction

The huge increase in the amount of data is entirely reshaping our daily lives. The use of data to optimize processes of all industries is becoming a reality. In fact, progress related to Artificial Intelligence is appearing more and more frequently, and in every domain.

In statistical modelling, the use of this data is mainly done using Machine Learning. The use of Machine Learning in the insurance field is being thoroughly studied all over the world. Indeed, as some of the insurance field processes are heavily regulated due to the nature of their implications, a gap between what the new techniques can provide in terms of innovations and what is currently done has widened.

In P&C Reserving, Machine Learning has been used to improve the quality of prediction compared to more naive methods, for example, Mario Wuttrich in his paper: "Neural Networks applied to Chain-Ladder Reserving" (2017) [17] has used Machine Learning techniques to predict IBNR amounts, using aggregated methods (based on a segmentation of claims) with development factors computed using Neural Networks on information based at the claim level (individual claim reserving).

Examples of more complex aggregated methods to estimate development factors, resulting in an improvement of the quality of prediction, are flourishing. However, the use of Machine Learning to improve the quality of these segmentations is far less popular.

The goal of this paper is to challenge these segmentations with the use of Machine Learning, by building new segmentations of claims using data available at the contract and claim levels, and to measure the relevance of the new segmentations by comparing them to the existing one used by SCOR.

To evaluate the homogeneities of the segmentations, the developments obtain via Chain-Ladder have been used, as they clearly highlight the natural developments of the classes without any experts judgements.

SCOR's current segmentation process, called the Actuarial Segmentation, is as follows:

- The great majority of contracts are classified in different actuarial segments, based on what seems at first like arbitrary criteria (Subsidiary, Line of business or Type of policy for example);
- All the claims belonging to the same actuarial segment are considered to have the same development pattern;
- Therefore, all the claims belonging in the same segment will all be projected in the same way, inducing a certain amount of IBNR.

In order to challenge the existing segmentation with the use of Machine Learning, the following steps are followed:

- Building of the database, containing information that can explain the different types of underlying risks;
- Construction of indicators (target variables) that can describe the development patterns, for both Incurred and Paid cash-flows;
- Building the segmentations, using either decision trees or clustering methods;
- Defining measures of the quality of a classification and estimating the effects of using such classifications, essentially by estimating IBNR and Reserves amounts using Chain-Ladder;
- Selecting the most predictive segmentation, and compare its composition with the Actuarial Segment;
- Studying the relevance of using such models.

This thesis is organized in three parts:

1. Reserving and Machine Learning theory:

   This part will cover the theory behind the usual Chain-Ladder and the principle of a segmentation of claims. In this study, only Chain-Ladder based reserving has been used to compare the impact of the segmentations on the IBNR estimations. This part will also detail the Machine Learning algorithms used later in this paper: Decision Tree and Clustering to build the new segmentations, and Neural Networks, Random Forest and Gradient Boosting to study the predictability of the classes.

2. Presentation of the study database:

   In this part, the process of building the database will be detailed. Starting with the export of the raw database from Business Object. Some variables needed treatments to be used, it was necessary to create new variables. Among all the

variables, a selection was made in order to only keep the usable and relevant ones. All the kept variables will have their values displayed and the correlations between the variables are also detailed to prevent an eventual bias in the model.

An issue raised in this paper is the management of missing values. To solve that problem, methods based on dimension reduction have been used. However, due to technical limitations, this method must be complemented by the use of clustering to reduce the number of categories. Moreover, as the function to find the optimal parameters, was not usable on large datasets, another method, based on probabilities of belonging to the categories was used.

In a first step, a focus was made on a more homogeneous subset of the database, with claims in the "Fire" Line of Business, composing half of the whole database and being related to property risks, this database will be referred as the Fire Database. The reduction of the database was done so as to quicken the explanatory process. Once the whole process established for this subset, it will be generalized to the whole database.

3. Building of a new segmentation:

   To challenge the current segmentation, two types of segmentations were tested, based on either Decision Trees or Clusterings. The study of the errors on indicators, regarding Incurred and Paid cash-flows, was used to choose the right set of parameters for each type of approach.

   For all these segmentations, the 2018 cash-flow and IBNR / Reserves prediction was estimated and compared with the Actuarial Segment and a segmentation containing one class with all the claims:

   - The estimation of the cash-flow of Incurred/Paid for the year 2018;
   - The estimation of the IBNR / Reserves not knowing the last three years of development;
   - The estimation of the IBNR / Reserves not knowing the last five years of development.

From these results, the segmentation predicting the closest to the actual values was kept, and its composition was studied, as well as the patterns of each of its class.

Part I

Reserving and Machine Learning theory

Introduction

The study of this paper revolves around the relevance of new segmentations compared to others based on different criteria.

The processes used can be quite complex and involve different steps, sometimes juggling between Reserving and Machine Learning notions.

Defining precisely the framework of this comparison and the techniques used is essential. Therefore, defining the notions used in this study was done separately, as to not lose the reader by spreading new notions throughout this paper.

Chapter 1

Traditional methods in Reserving

This chapter details the Reserving methods used, as well as some concepts regarding the Reserving methodology that will later be used in this paper.

This study being based on the quality of the segmentation itself, it is adequate to use the Chain-Ladder with no expert judgements, as it highlights the natural development of the claims. Therefore, enabling a fair comparison between the segmentations, without introducing experts judgements.

1.1 Claim life cycle

The typical claim cycle is composed of many steps, displayed hereafter:

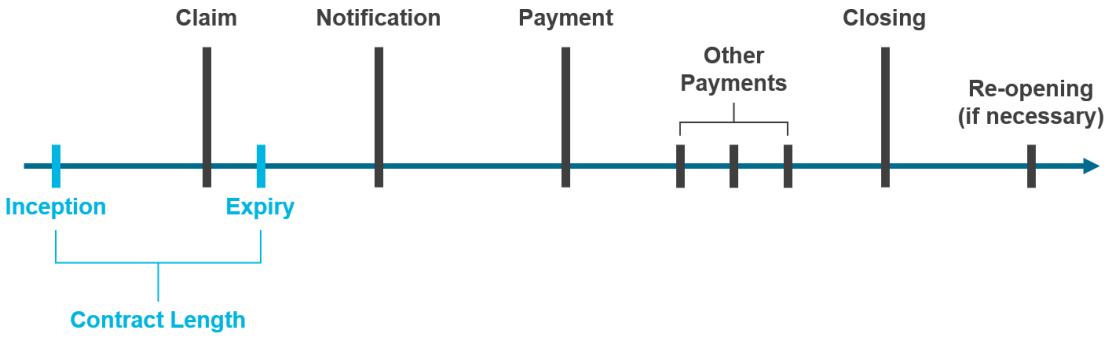


Figure 1.1: Claim life cycle

In Reserving, it is needed to estimate how a claim will develop in order to predict the necessary amount to set aside in order to cover the payments that will occur. A common way to do so is by using the Chain-Ladder method.

1.2 The Chain-Ladder method

In this section will be introduced the theory behind the Chain-Ladder method, its construction and the way it is used in P&C Reserving. This section will be focused on incurred amounts, please note that the methodology is the same for paid amounts.

1.2.1 Underlying hypotheses

The Chain-Ladder method is constructed around strong hypotheses:

1. Two adjacent development years are proportional;
2. The way the claims develop throughout the years is the same (independence of underwriting year).

From these hypotheses, by knowing the development of the claims for previous underwriting years, it is possible to estimate the Incurred/Paid amount for the calendar year using a proportional approach.

These hypotheses rely on the absence of changes in the underwriting policy or the claims management, if changes like these were to happen, the patterns of development could drastically change from one year to another, making the proportional approach irrelevant.

1.2.2 Construction of the Chain-Ladder method

If an upper triangle of incurred amount is known, it is possible to compute development factors for each period. A common way to do so is by considering the average of the development factors weighted by the incurred amount for each development period.

Let $C_i^j \ \forall i \times j \in [1:N] \times [1:P] \ \ i \leq j$ be an upper triangle of incurred cumulative amounts with:

- C_i^j cumulative amounts of incurred: $C_i^j = \sum_{k=1}^{i} c_k^j = C_{i-1}^j + c_i^j$ with c_i^j being the incremental amount of incurred for the i-th development period and the j-th underwriting year, $i \leq j$ by construction;
- N the number of development periods;
- P the number of underwriting years.

Let f_i^j be the development factor for the i-th period and the j-th underwriting year, f_i^j is the ratio of the cumulated amounts of the i-th and i-1-th period:

$$f_i^j = \frac{C_i^j}{C_{i-1}^j} \tag{1.1}$$

The underlying hypothesis states that supposedly the development factors for all the underwriting years are the same. Therefore: $\forall i \in [1:P] \;\; \exists f_i$ such that:

$$\forall j \in [1:N] : \;\; f_i^j = f_i \tag{1.2}$$

From that theoretical results, still under the hypothesis that all the underwriting years will have the same development pattern, it is possible, knowing the incurred amounts of the previous years to estimate the incurred amount of the years to come. To do so, estimating f_i is necessary, one common way to do so is to consider the mean of f_i^j weighted by their respective amounts of incurred:
$\forall i \times j \in [1:N] \times [1:P] \;\; i \leq j$

$$\hat{f}_i = \frac{\sum_{j=1}^{N-i} f_i^j \times C_{i-1}^j}{\sum_{j=1}^{N-i} C_{i-1}^j} = \frac{\sum_{j=1}^{N-i} C_i^j}{\sum_{j=1}^{N-i} C_{i-1}^j} \tag{1.3}$$

The development factors now built, filling the triangle of development is possible, to do so, the incurred amount of the second half of the triangle will be deduced from the previously computed development factors:
$\forall i \times j \in [1:N] \times [1:P] \;\; i > j$

$$\hat{C_i^j} = \hat{f}_i \times C_{i-1}^j \tag{1.4}$$

Hereunder is a visualization of the filling process:

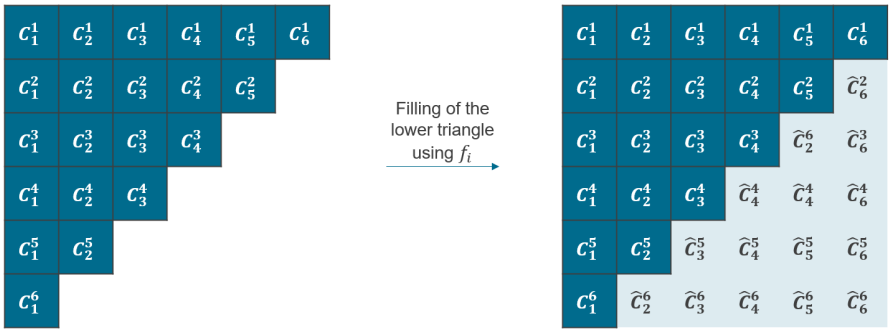


Figure 1.2: Filling the lower triangle of developments

Note: It is common practice to exclude some factors of development that can seem to have abnormal values, which can be due to the fact that a certain segment may have few claims, and therefore a high volatility. However, these corrections introduced a form of experts judgement. Therefore, the predicting process in this paper will not consider any form of correction, as to only focus on the natural underlying patterns, induced by the segmentations.

Convergence of the claims

The lower triangle now filled, either the claims are said to be fully developed, if that is the case, $\lim_{i \to N} f_i = 1$. If the claims are long tail, there might not be enough development periods available to consider the claims fully developed. If so, the development factors have to be estimated for the future development periods (using a parametric formula for example) and set to converge in a certain way that can be predicted, via either expert judgement or market value.

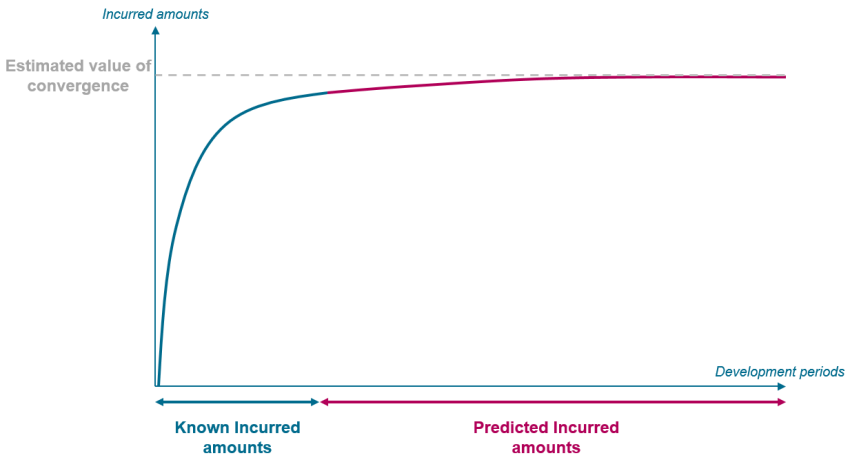


Figure 1.3: Predicting Incurred amounts

Knowing the pattern followed by the claims enables an insurance or reinsurance company to define the following amounts:

Definition 1.2.1. Ultimate amount: corresponds to the sum of the cash-flows for a claim (of either Paid or Incurred), it is defined as such:

$$Ultimate = \sum_{j \in UWY} Ultimate_j = \sum_{j \in UWY} \sum_{i=1}^{T} c_i^j \tag{1.5}$$

With:

- j the underwriting year;
- i the periods of development;
- c_i^j the incremental amount of Incurred or Paid per period;
- T horizon at which the claims are considered fully developed: $\forall j > T,\ c_i^j = 0$.

Definition 1.2.2. Incurred But Not Reported (IBNR): corresponds to the amount of reserve to set aside in order to match the amount necessary to cover the Ultimate cost of the claims:

$$IBNR = \sum_{j \in UWY} IBNR_j = \sum_{j \in UWY} (Ultimate_j - Actual_j) \tag{1.6}$$

With:

- j the underwriting year;
- $Ultimate_j$ the sum of all cash-flows;
- $Actual_J$ the amount already paid and in reserve.

Note: When using Paid cash-flows for Ultimate and Actual (instead of Incurred), this amount corresponds to the Reserves.
On the triangle of development, it is possible to visualize the Ultimate and IBNR amounts as such:

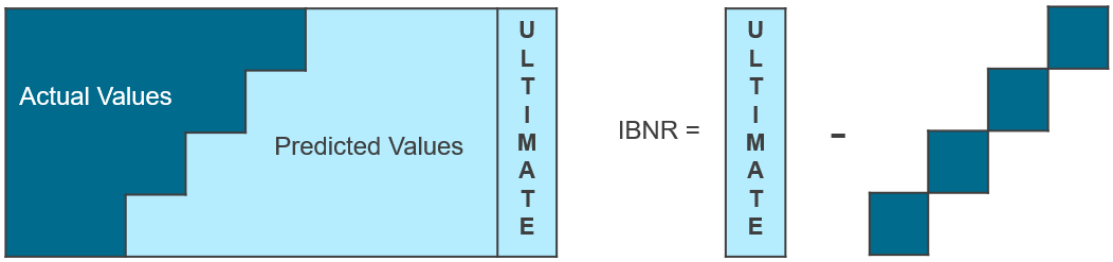


Figure 1.4: Visualizing the IBNR on a triangle of cumulated developments

1.3 Why is a segmentation needed?

For reserving projections, such as Chain-Ladder, a segmentation of claims is necessary to regroup claims with the same kinds of patterns and to exclude claims with different patterns in other segments.

If all the claims of an insurer portfolio are grouped into one single segment, some claims with different behaviours could be projected using the same pattern. This could result in overestimating or underestimating the amounts of reserve.

For example, if an insurance company decides to group two different type of claims, one being short-tail (the claims are quickly fully developed) and one being long-tail (the claims take longer to reach their full development), the development pattern of the portfolio containing both types of claims will be sensible to the proportion of each type of claims.

Hereunder is an example of the impact of the proportion of short-tail/long-tail claims in the development pattern:

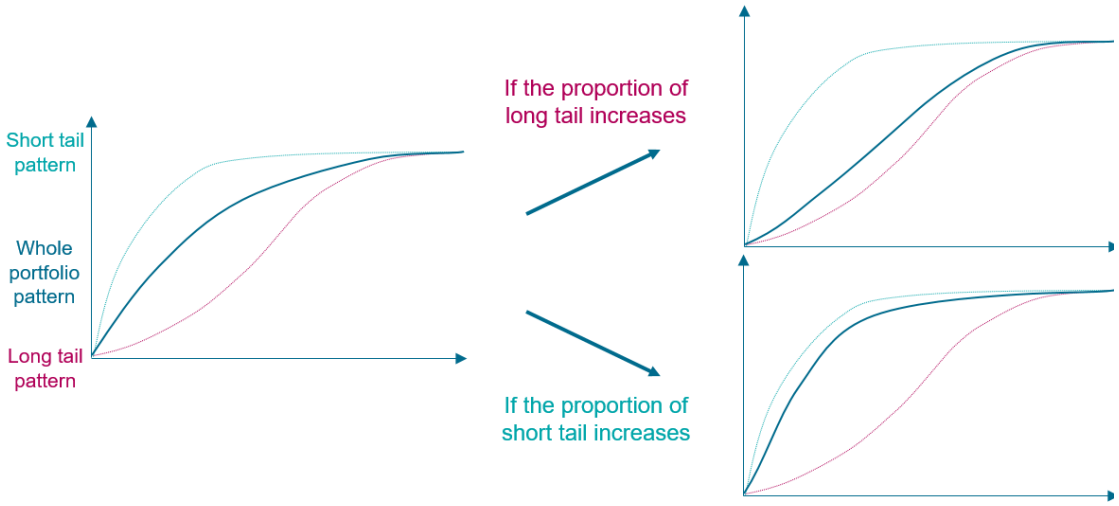


Figure 1.5: Impact of the proportion of claims on the pattern of development

While the pattern of the whole portfolio is sensitive to the proportion of different claims, the patterns for each type of claims remain the same. Therefore, it is possible that the portfolio merging short tail and long tail claims will not predict IBNR amounts corresponding to the underlying risks: if a short-tail claim falls in a long-tail class, the IBNR amount regarding that claim will be over-estimated, and vice-versa.

Through this simple example, the necessity of segmenting the claims with different patterns is put into evidence. On the other hand, having a too detailed segmentation, by reducing the number of claims in each segment, could also increase the volatility of the estimation of the development factor: there is a trade-off between homogeneity of the segment and the quality of the prediction.

1.3.1 Regulatory aspects of the building of homogeneous risk groups

The European Directive Solvency II, enforced since the 1st of January 2016, deeply restructured the Insurance regulations. The Solvency II regulation specifies the framework of the segmentation:
Directive 2009/138/EC, Section 2, Article 80:

Segmentation

> "Insurance and reinsurance undertakings shall segment their insurance and reinsurance obligations into homogeneous risk groups, and as a minimum by lines of business, when calculating their technical provisions."

References to segmentation policy are also present in the Official Journal of the European Union, L12, Volume 58, Article 34 (3):

Calculation methods

> [..] Where a calculation method is based on grouped policy data, insurance and reinsurance undertakings shall ensure that the grouping of policies creates homogeneous risk groups that appropriately reflect the risks of the individual policies included in these groups.

And in the the Official Journal of the European Union, page 9:

> The segmentation of insurance and reinsurance obligations into lines of business and homogeneous risk groups should reflect the nature of the risks underlying the obligation. The nature of the underlying risks may justify segmentation which differs from the allocation of insurance activities to life insurance activities and non-life insurance activities, from the classes of non-life insurance set out in Annex I of Directive 2009/138/EC and from the classes of life insurance set out in Annex II of Directive 2009/138/EC.

Respecting the framework imposed by the Solvency II regulation is mandatory for a segmentation to be approved by the regulator. Therefore, the construction of the new segmentations must consider and comply with these guidelines.

Conclusion on the reserving methods

The reserving methods presented in this chapter are fairly simple. Allowing us to focus on the study of the natural underlying patterns, that are highlighted when using a Chain-Ladder without any experts judgements. The quality of the segmentations will therefore be more easily interpretable.

Chapter 2

Machine Learning

This chapter focuses on the theory behind the Machine Learning algorithms used later in this paper.

Each parameter is briefly presented, to allow the reader to understand how the algorithm works and how the tuning of the parameters can modify its structure and its results. For more details on the parameters, as well as a proper mathematical framework, please refer to the appendix A.

Machine Learning algorithms can be divided into two major categories: supervised and unsupervised learning:

2.1 Supervised Learning

In Supervised Learning, the algorithms will fit a model on a training dataset in order to predict either a class (in Classification) or a numerical value (in Regression). Once the model has estimated either values or classes, it will compare the model's evaluation and the actual values.

In this section, the following algorithms will be explained:

- Artifical Neural Network;
- Decision Tree;
- Random Forest;
- Gradient Boosting.

By adjusting the model, the algorithm will get closer to predicting the "real" values. In the end, a complex enough model will have results matching perfectly the "real" values. However, when used to predict values or classes from another dataset with different data, called the test dataset, the model will not be as correct as it was on the training set, this phenomenon is called over-fitting. For this reason, only the error estimation on a dataset not used for fitting, called the test dataset, will be displayed. The test dataset is obtained using a random selection of the rows of the total database.

2.1.1 Artificial Neural Networks

Artificial Neural Networks are based on the communication processes of brain cells. An Artificial Neural Network is composed of different neurons arranged in layers, each neuron is connected to every neuron of the two layers next to it (the layer before and the layer after), by synapses:

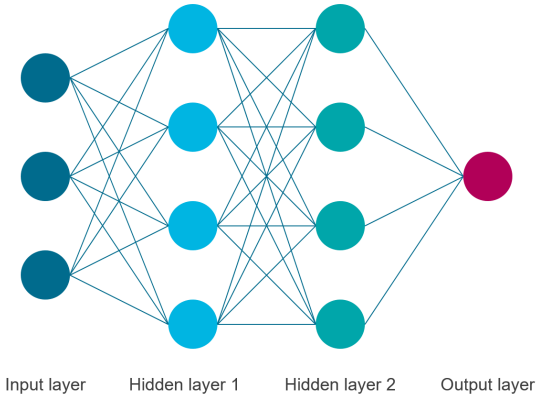


Figure 2.1: Example of Neural Network

The example above is a Neural Network with an input layer composed of 3 neurons (3 variables used for predictions), two hidden layers each composed of 4 neurons and finally the output layer composed of one final neuron.

The neurons of the Artificial Neural Network are each associated with a value:

$$Neuron\,(i,h)_{Value} = z_i^{(h)} = \phi\left(W_{i,0}^{(h)} + \sum_{j=1}^{n_{h-1}} W_{i,j}^{(h)} \times z_j^{(h)}\right) \tag{2.1}$$

With:

- $h \in \{1, 2, ..., H\}$ layer of the neuron;
- $z_i^{(h)}$ value of the $i \in \{1, 2, ..., n_h\}$ neuron of the h layer;
- ϕ the activation function: sigmoid, Rectified Linear Unit ($ReLU(x) = max(x, 0)$);
- $W_{i,j}^h$ weight of the value of the j-th neuron of the previous layer, in the value of the i-th neuron of the h layer.

Iteratively, from the equation above, the output value is a function of all the weights for each layer and of the input values. Therefore, the model will compute an optimisation algorithm (a gradient descent algorithm for example) to minimize the error between the actual data and the model's estimation.

Note: Neural Networks are based on calculus using numerical data, therefore the scale of the variables has an impact on the importance of the variables, to allocate each variable with the same weight, a scaling of variables is necessary (using a min-max method or standardisation).

Artificial Neural Network parameters The following parameters are available to customize the Artificial Neural Network. More parameters are available for fine algorithm tuning but the following are the most important:

Parameter	What is controls
Hidden layers	Controls the number of hidden layers, and the number of neurons composing each layer
Activation	The type of activation to use, can either be: an identity function, a logistic function, an hyperbolic tangent function or a ReLU
Solver	The optimisation algorithm to use: Newtonian, Gradient Descent...
Learning rate	The learning rate of the optimisation algorithm
Max iteration	The maximum number of iterations for the optimisation algorithm

Table 2.1: Artificial Neural Network parameters

To get more details about the Artificial Neural Network please refer to the Appendix A.

2.1.2 Decision Tree

A decision tree is a simple algorithm that separates the dataset by successive cuts in the span space of the dataset variables. At each node the algorithm will find the best variable to use and at which point to cut in order to obtain the best criteria that will cut the population into the two most homogeneous subsets (sub-populations) and thus until a certain condition is verified. In this paper, the assumption that all the Decision trees are binary is made, which means that for every split, only two subsets will be obtained from the original dataset.

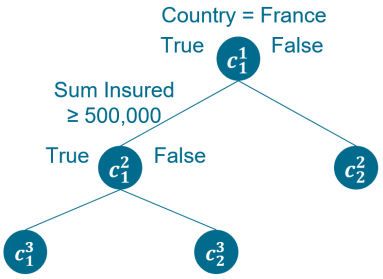


Figure 2.2: Example of a Decision Tree

Decision Tree parameters The following parameters are available to customize the Decision Tree. More parameters are available, but the following are the most important:

Parameter	What is controls
Criterion	The impurity function, used to measure the quality of a split. Commonly: Gini/Entropy in classifications, MSE in regressions
Max depth	The maximum depth of the tree
Min samples split	The minimum number of observations required for a node to be split
Min samples leaf	The minimum number of observations required to constitute a leaf
Max leaf nodes	Maximum number of leaves (cannot exceed $2^{Max\ Depth}$)
Min impurity decrease	Threshold of decrease of impurity to allow a node to be splitted
Min impurity split	If the impurity is below this threshold, the node will not split.

Table 2.2: Decision tree parameters

To get more details about the Decision Tree please refer to the Appendix A.

2.1.3 Random Forest

A Random Forest is a bagging method based on Decision trees:

$$\text{Bagging} = \text{Bootstrap} + \text{Aggregate}$$

- Bootstrap: A Bootstrap algorithm will enhance the sturdiness of an algorithm learning process; It re-samples the dataset using sampling with replacement, therefore the same line can appear multiple times; This sampling with replacement can be done on both observations and variables.
- Aggregate: Many decent classifiers together will have better results and a smaller volatility than a unique very good classifier. Many classifiers will be computed, and at each node, the split will be fitted on the re-sampled dataset, the final output will be the average over all the classifiers. This will reduce the variance of prediction.

The Random Forest algorithm will generate multiple decision trees, with each split fitted to different samples generated using Bootstrap and will either:

- For classification: Place the observations where they have been placed the most among all the trees composing the forest.
- For regression: Associate the observations with the mean value on all trees composing the forest.

Hereunder is an example of a Random Forest classification decision if the majority of trees have made the same decision as the first sample:

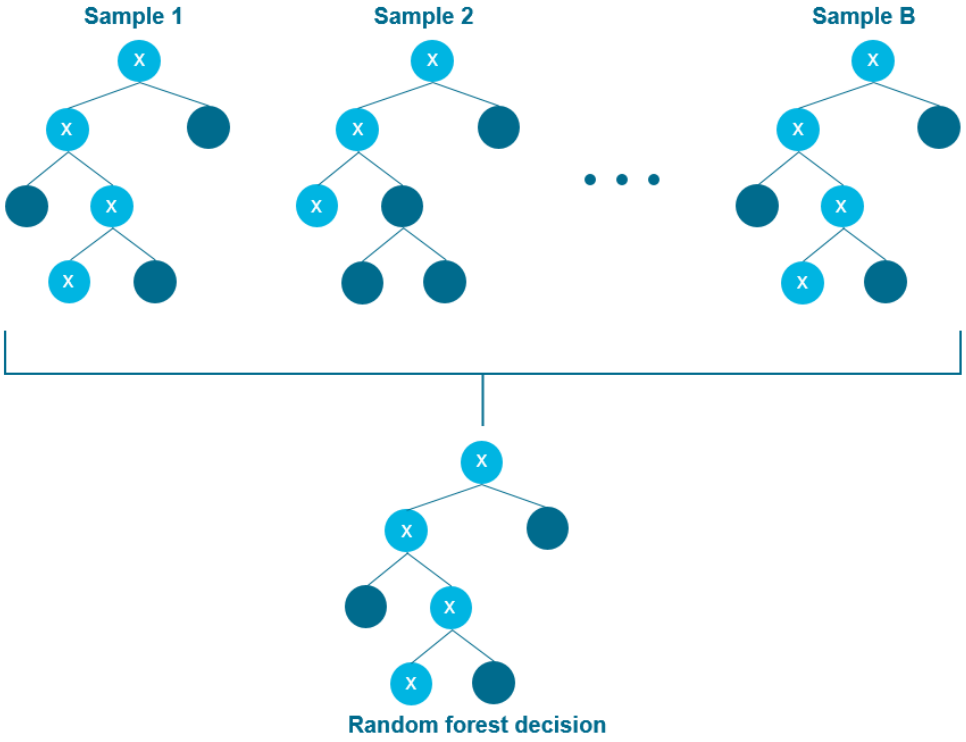


Figure 2.3: Random Forest decision making

Random Forest parameters The following parameters are available to customize the Random Forest. More parameters are available for fine algorithm tuning, but the following are the most important:

Parameter	What is controls
N estimators	Number of trees that compose the Random Forest
Max features	Proportion of variable available at each split for each tree
Bootstrap	Whether to use bootstrap samples of observations or not
+ Decision Tree parameters	Decision Tree parameters will be applied to each tree

Table 2.3: Random Forest parameters

To get more details about the Random Forest please refer to the Appendix A.

2.1.4 Gradient Boosting

Gradient Boosting is a method also based on Decision Trees. Gradient refers to the optimisation approach: the algorithm is based on using different models one after the other. Each model will be adjusted to the error of the previous models.

Boosting refers to the use of many "weak" models with high variance (models that taken alone have bad results) but that taken together end up with a better model having a small variance.

Hereunder is a theoretical example to better visualize how a Gradient Boosting algorithm works:

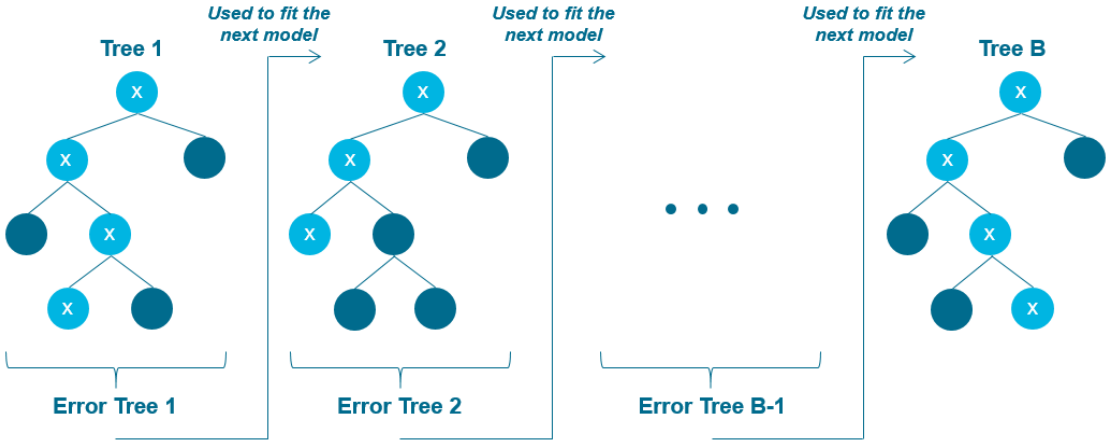


Figure 2.4: Gradient Boosting iterative process

Gradient Boosting parameters The following parameters are available to customize the Gradient Boosting. More parameters are available for fine algorithm tuning, but the following are the most important:

Parameter	What is controls
Loss	Loss function to optimize
Learning rate	Weights of the contribution of each successive tree, the smaller the learning the higher the number of trees,
N estimators	Number of steps to perform
Subsample	Fraction of lines to use in the samples (if $subsample < 1$: results in Stochastic Gradient Boosting)
Max_features	Proportion of variables available at each split for each tree
Bootstrap	Whether to use bootstrap samples of observations or not
+ Decision Tree parameters	Decision Tree parameters will be applied to each tree

Table 2.4: Gradient Boosting parameters

Advantages and disadvantages of each algorithm:

Hereunder is a table describing the usual advantages and disadvantages of each algorithm presented:

	Advantages	Disadvantages
Neural Networks	Takes into account the interaction between the variables	Takes a long time to compute
		Requires scaling of the variables
		Very hard to interpret
Decision Tree	Very easy to visualize	High variance
	The user can have a full control over the output	Prone to over-fitting
Random Forest	Still easy to visualize	Prone to over-fitting
	Good prediction in general for a rather simple model	Harder than the Decision Tree to visualize
	Reduce of the variance compared to a decision tree alone	Can be long to compute
Gradient Boosting	Good prediction	Harder than the Random Forest to visualize
	Resist to over fitting	Can be very long to compute

Table 2.5: Advantages and disadvantages of each algorithm

2.2 Unsupervised Leaning

In Unsupervised Learning, the model does not try to predict or to find the correct output values. Unsupervised algorithms only work on the dataset, in order to achieve a certain task.

Unlike supervised training, the algorithm will not compare its result with the reality, there is no correct answer, the algorithm is left alone and untouched until it reaches its goal.

Unsupervised Learning is mainly used to find patterns in the datasets, for example dividing the dataset in different parts or finding dependencies among the variables.

2.2.1 Dimension reduction algorithms

2.2.1.1 PCA theory

PCA (Principles Component Analysis) is a method used to reduce the dimensional complexity of a dataset while keeping its characteristics/diversity.

The PCA method, applied to a dataset $X = X_{i,j}$ with $i \in 1, 2, .., n$ row index and $j \in 1, 2, .., p$ column index, will compute a subset of lower dimension ($q < p$) that will maximize the variance of the dataset when projected on it.

Which translates into building the best subset of dimension $\forall i \in \{1, .., p\}$ minimizing the Mean Square Error between the real data and the projected data:

$$\|X_{real} - X_{projected}\|^2$$

The following steps need to be followed in order to compute the PCA:

1. Standardization of the data:

$$Z_j = \frac{X_j - \overline{X_j}}{\sqrt{n \times Var(X_j)}} \quad \text{with} \quad X_j = \begin{pmatrix} X_{1,j} \\ X_{1,j} \\ \vdots \\ X_{n,j} \end{pmatrix} \quad \forall j \in \{1, ..., p\}$$

   This step is necessary, otherwise the PCA will separate variables with high values from the ones with small values regardless of their correlations.

2. Once the dataset is standardised, constructing the covariance matrix is needed:

$$\Sigma(Z) = \begin{pmatrix} \sigma_{1,1} & \sigma_{1,2} & ... & \sigma_{1,p} \\ \sigma_{2,1} & \sigma_{2,2} & ... & \sigma_{2,p} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{n,1} & \sigma_{n,2} & ... & \sigma_{n,p} \end{pmatrix}$$

   With:

$$\sigma_{i,j} = \frac{1}{n-1} \times \sum_{k=1}^{n} (z_j^i)(z_k^i) \qquad \forall i, j \in \{1, ..., p\} \times \{1, ..., n\}$$

3. The eigenvectors of the correlation matrix and their corresponding eigenvalues can be obtained via diagonalization of the correlation matrix, they are the principal components (axis on which the projected variance will be maximum):

$$\Sigma(X) \times \nu_j = \lambda_j \times \nu_j \qquad \forall j \in \{1, 2, ..., p\}$$

The subset of dimension $q < p$ is equal to $Span(\nu_1, \nu_2, ..., \nu_q)$ and its explained variance is equal to:

$$\frac{\sum_{i=1}^{q} \lambda_i}{\sum_{i=1}^{n} \lambda_i} \tag{2.2}$$

Studying the evolution of the explained variance for different dimensions is necessary in order to determine which dimension to use. A dimension that is too low might not replicate well enough the characteristics of the database while a value that is too high will make the PCA closer to the database, making it less useful.

PCA can be used to visualize potential groups of observations or to explain the relations between the variables. By construction, PCA can only be done on numerical variables. Furthermore, as it is based on linear algebra, the structure of dependencies can only be linear.

2.2.1.2 MCA theory

MCA (Multiple Correspondence Analysis) is the generalization of PCA for categorical variables. In order to be used, categorical variables must be re-encoded as numerical ones. To so so, the One-Hot Encoding method has been used.

Definition 2.2.1. One-Hot Encoder

This method will create an indicator vector for each category of the variable:

	Sub
Claim 1	Sub 3
Claim 2	Sub 1
Claim 3	Sub 2
Claim 4	Sub 2

→

	Sub 1	Sub 2	Sub 3
Claim 1	0	0	1
Claim 2	1	0	0
Claim 3	0	1	0
Claim 4	0	1	0

Figure 2.5: One-Hot Encoder

When using One-Hot Encoder, the size of the dataset can increase significantly if it contains categorical variables with a lot of different categories, as for every category, a new column will be computed.

One-Hot Encoding can be done whether the variables are ordinal or not. Once the categorical variables are numerically encoded, the PCA will be applied to the following triplet of matrices:

$$\left(IXD_{\Sigma}^{-1}, M = \frac{1}{IJ}D_{\Sigma}, D = \frac{1}{I}\mathbb{1}_I\right) \tag{2.3}$$

With:

- X: the indicator matrix of category for each variable: Let a dataset with I observations and J variables, with each variable $j \in J$ having n_j different categories, X will be a matrix containing I rows and $\sum_{j=1}^{J} n_j$ columns, each column corresponding to the indicator vector for a category in a categorical variable (cf Fig. 2.5);
- $D_{\Sigma} = diag\left(1_{n_1}, 1_{n_2}, ..., 1_{n_J}\right)$ with $\forall j \in \{1, 2, .., J\} \quad 1_{n_j} = \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix}$ of length n_j;
- $M = \frac{1}{IJ}D_{\Sigma}$ the metric;
- $D = \frac{1}{I}\mathbb{1}_I$ matrix of the row weights.

Note:
While for a PCA the maximum number of dimensions is the number of variables, for a MCA the maximum number of dimensions is equal to the sum of all categories for all variables. Therefore, a higher number of dimensions can be required when using an MCA, especially if the variables have a lot of different categories.

2.2.2 Clustering algorithms

Many problems require the building of homogeneous classes, but computing the optimal classes is one of the most complex and time-consuming tasks, in fact it is a NP-hard problem, which means that the computing complexity of this problem is above any polynomial function. Clustering algorithms offer an alternative to this segmentation problem with far less costly method.

The following clustering algorithms are all based on the same principle, called the Lloyd's algorithm:

1. The number of clusters N is set by the user, each cluster will be associated with a center of gravity, also called centroids;
2. N points are randomly placed in the feature space of the database, these points will be the centroids of the clusters;
3. All the points of the dataset will be associated to the clusters with the closest centroids based on a distance measure.
4. The centroids are then set to be equal to the center of gravity of the points for each cluster;
5. The same process is done multiple times until the number of movements among the clusters is less than a defined number.

The main difference between clusters based on only numerical or only categorical or mixed type variables is the measure of the distance between points to determine in which cluster an observation belongs to.

While less costly, this algorithm can be stuck in local minima, for this reason it is common use to compute the algorithm multiple times with different starting centroids. To determine which segmentations is the best, the inertia can be studied.

Definition 2.2.2. Inertia: The sum of the squared distances between the observations and the centroids of the cluster they belong to:

$$Inertia = \sum_{k=1}^{K} \sum_{i=1}^{I^k} \left(x_i^k - \overline{x_k} \right) \tag{2.4}$$

With:

- K the number of clusters;
- I^k the number of observations in the cluster k;
- $\overline{x_k}$ the centroids (center of mass) of the k-th cluster;
- $x_\cdot^k$ observations belonging in the k-th cluster.

2.2.2.1 Clustering for only numerical variables: k-means

For only numerical variables, the measure used is the Euclidean distance.

Definition 2.2.3. Euclidean distance:

Let X and Y be two observations with N numerical variables: $X = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix}; Y = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{pmatrix}$.

$$Euclidian\ Distance(X, Y) = \sum_{j=1}^{N} \|x_j - y_j\|^2 \tag{2.5}$$

The algorithm will assign the observations to the cluster with the closest centroid and thus iteratively until all the observations are correctly classified.

Note: The k-means algorithm makes the assumption that the optimal subsets are convex. Therefore, when trying to predict clusters for supposedly non-convex subsets, other algorithm should be used (Affinity Propagation, Agglomerative clustering or Mean-shift for example). The different outputs of these methods are displayed hereafter, using an example available on the Sklearn website:

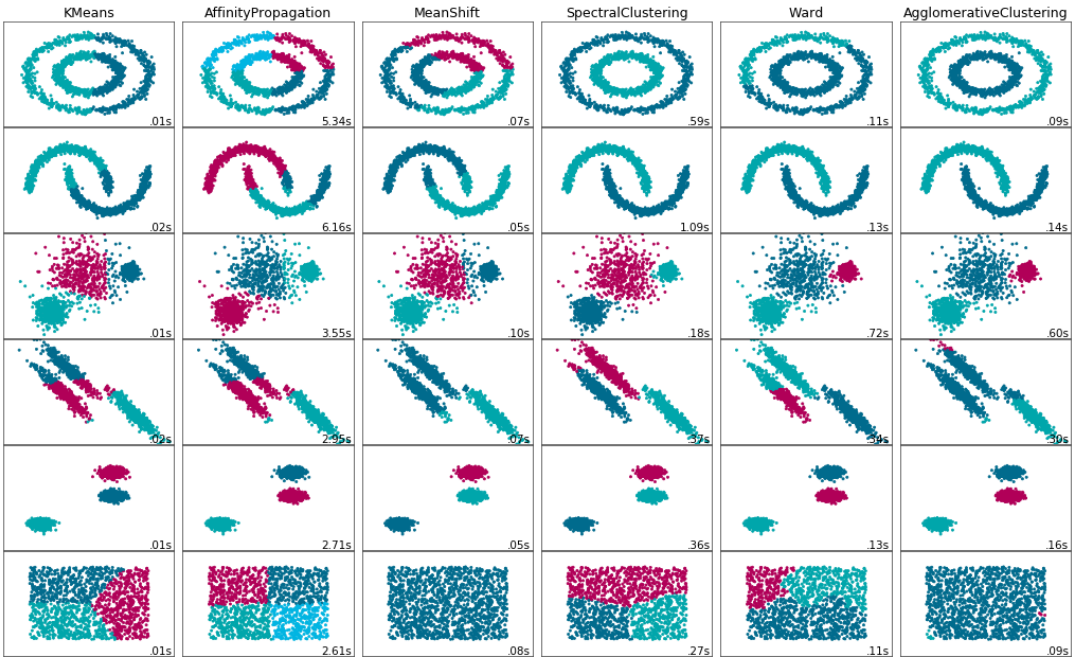


Figure 2.6: Comparison of the results using different clustering algorithms

In this study, after visualizing the distributions of the numerical variables used, the convexity of the subsets does not seem to be problematic, moreover, the other algorithms

are often more complex and require more computer power, due to the size of our database, testing some of the other clustering algorithms was not possible. For more information on this subject please refer to the Sklearn website [22].

2.2.2.2 Clustering for only categorical variables: k-modes

Introduced by Z. Huang (1998) [6], k-modes offer an alternative to the k-means algorithm for categorical variables.

The variables are encoded using One-Hot-Encoding and to measure the distance between two observations, the Hamming distance is used:

Definition 2.2.4. Hamming distance:

Let X and Y be two observations with P categorical variables: $X = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_P \end{pmatrix}; Y = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_P \end{pmatrix}$.

$$Hamming\ Distance(X,Y) = \sum_{j=1}^{P} \mathbb{1}\left(x_j \neq y_j\right) \tag{2.6}$$

The less categories two observations have in common, the farther they are. Using that distance, the clustering algorithm follows the same iterative principle as the k-means algorithm: the algorithm will assign the observations to the cluster with the closest centroid and thus iteratively until all the observations are correctly classified.

2.2.2.3 Clustering for both numerical and categorical variables: k-prototypes

While k-means were only able to cluster numerical variables and k-modes only categorical variables, k-prototype, also introduced by Z. Huang [6], enables the computation of clusters based on mixed variables (numerical and categorical). The distance used for k-prototypes is a mix of the two previous distances (2.5 and 2.2.4).

Definition 2.2.5. Distance used for k-prototypes:
Let X and Y be two observations with N numerical variables and $P - N$ categorical variables: $X = \begin{pmatrix} x_1 \\ \vdots \\ x_N \\ \vdots \\ x_P \end{pmatrix}; Y = \begin{pmatrix} y_1 \\ \vdots \\ y_N \\ \vdots \\ y_P \end{pmatrix}$.

$$K\text{-}prototype\ Distance(X,Y) = \sum_{j=1}^{N} \|x_j - y_j\|^2 + \gamma \sum_{j=N+1}^{P} \mathbb{1}\left(x_j \neq y_j\right) \tag{2.7}$$

With $(x_i)_{i=1}^{N}$, $(y_i)_{i=1}^{N}$ being the numerical attributes and $(x_i)_{i=N+1}^{P}$, $(y_i)_{i=N+1}^{P}$ being the categorical attributes. The distance for k-prototypes is the sum of the Euclidean distance for numerical attributes and the Hamming distance for categorical attributes weighted by a factor γ to even-out the weights of each type of variables.

Conclusion on the Reserving and Machine Learning Theory

The theoretical notions now defined, they will be put to use in the following two parts.

The Machine Learning algorithms will be used to:

- Fill the missing values of the dataset: unsupervised algorithms (PCA/MCA) are used to fill the missing values based on the correlations between the variables. Clusters are also used to reduce the number of different categories;
- Build the alternative segmentations: decision trees and clusters (k-means and k-prototypes);
- Evaluate the homogeneity of the newly built segmentations: supervised algorithms;
- Classify the recent claims: classification algorithms.

The reserving notions defined will be used to quantify the quality of prediction of the segmentations considering:

- The predictability of the 2018 cash-flow, for Incurred and Paid;
- IBNR and Reserves predictions on a database of closed claims.

The estimations will be done using Chain-ladder, as it gives the natural development patterns of each class without any experts judgements.

Part II

Presentation of the study database

Introduction

In this part will be presented the different variables contained in the database, the selection of variables and the reasons behind that selection.

Then, a presentation of the techniques used to improve the quality of the database will be detailed, especially via the filling of missing values based on dimension reduction.

The processes will firstly be applied to a subset of the database, called the Fire database, as it composed of the line of business Fire, composed of property contracts and representing 50% of the whole database. Once established, the processes will be generalized to the whole database.

An overview of the variables is available in the appendix B. In this overview, the distribution of each variable is presented for the Fire database, as well as a study of the correlations between variables.

The process of building the patterns indicators will then be detailed. A study of the relevance of these indicators is available at the end of the appendix B. It is not included in this part as it uses notions presented in the part three.

Chapter 3

Overview of the database

The database is composed of claims regarding Facultatives contracts.

Facultatives: A facultative contract provides an optional reinsurance protection related to one risk only. Whereas the common protections provide a protection to a group of risks, Facultatives can be used to cover unusual risks that fall outside of common covers' scope.

Most of the risk covered by the Facultatives are related to SCOR Business Solution, which is SCOR's major Specialty Insurance branch. Scor Business Solution, also refereed as SBS is the Corporate risk insurance arm of SCOR, it provides customized covers for clients exposed to risks with commercial high values.

The number of claims in classical reinsurance is low, by nature of the business. However, since each facultative contract provides cover single risks only, the number of facultatives contract is rather high. For these reasons, the choice of building a database composed of facultatives contracts was made.

The database is extracted using Business Object, from SCOR's internal sources, with data available at SCOR 2018 year-end. All amounts are in Euro as SCOR 4Q18 exchange rates.

The database contains information describing the contracts and the claims themselves. It has nearly $450,000$ claims, on all subsidiaries and with underwriting years ranging from 1950 to 2018

3.1 Regulatory aspects of the building of the database

Through this section, the compliance of the database construction with regulations on both data privacy (GPDR) and the data quality (Solvency II) will be detailed.

3.1.1 GDPR

The General Data Protection Regulation (GDPR) is a European Regulation enforced on the 25th of May 2018, replacing the Data Protection Directive 95/46/EC [21]. The GDPR regulates the use of personal data by observation, company or organization. Therefore, it has important impacts on the way companies deal with personal data. However, despite having completely reshaped personal data management, it is solely confined to the protection of personal data. The database of this study does not contain personal information regarding individuals, since it is only composed of Facultative contracts between companies and SCOR, the GDPR does not apply on the database of this study.

3.1.2 Solvency II

The Solvency II regulation brought up criteria regarding all fields of the insurance sectors, including data science.

The Solvency II regulation explicitly specifies three criteria, as stated on the Solvency II regulation documentation, article 82 (3): [19]:

> Data quality and application of approximations, including case-by-case approaches, for technical provisions
>
> Member States shall ensure that insurance and reinsurance undertakings have internal processes and procedures in place to ensure the appropriateness, completeness and accuracy of the data used in the calculation of their technical provisions."

In order to include the database in the IBNR estimation process and to comply with the Solvency II regulation when building the database, these three criteria: appropriateness, completeness and accuracy, must be verified.

A study of the database compliance with these three criteria will therefore be presented in the section 4.3: Quality of the study database.

3.2 Original variables

The following variables were directly extracted from BO:

Variable's name	Description	Type
CLM - Claim	Identification number of the claim	Key
Country_Claim	Country of the claim	Categorical
Country_Cedent	Country of the cedent	Categorical
Inception _Date	Date of the start of the cover	Date
Claim_Date	Date of the claim	Date
First_Notification_Date	Date at which SCOR knew about the claim	Date
Creation_Date	Date of creation of the claim in SCOR database	Date
Closing_Date	Date at which the contract was declared closed	Date
Contract_Expiry_Date	End date of coverage	Date
Lob	Contract's line of business	Categorical
Sob	Contract's scope of business	Categorical
Top	Type of policy	Categorical
Claim_Cause	Claim's type of cause	Categorical
Subsidiary_Event	Event that led to the claim	Categorical
Subsidiary_Event_Type	Type of event that led to the claim	Categorical
Fac_Sector	Sector of the contract	Categorical
Risk_Nature	Code transcribing the type of underlying risk	Categorical
Insured_Sector	Sector of the insured (internal classification)	Categorical
CLM_UWY*	Contract's underwriting year	Categorical
CLM_Subsidiary *	Classification among SCOR entities	Categorical
CLM_Subsidiary_Ledger*	Sub-classification of Subsidiary	Categorical
CLM_Cedent*	Contract's Cedent	Categorical
CLM_Cedent_group*	Group of Cedent	Categorical
CLM_Cedent_Ultimate_group*	Group of Cedent at ultimate	Categorical
Main_Currency	Currency of the contract	Categorical
CLM_Last_position*	Number of estimates done by SCOR on the claim	Numerical
Claim_latent	whether the contract is classified as latent or not	Binary
Nature	Nature of the coverage: QS, XL or XS	Categorical
Brokerage_rate	Brokerage rate in %	Numerical
Retrocession_rate	Retrocession rate in %	Numerical
Follow_up	Name of the procedure if there is one	Categorical
Sum_Insured	Amount under coverage	Numerical
PML_100%	Probable Maximum Loss in total	Numerical
PML_SCOR	Probable Maximum Loss for SCOR	Numerical
SCOR_EGPI	Estimated Gross Premium Income	Numerical
Annual_Limit	Limit of coverage	Numerical
Layer_cap	Limit of the layer	Numerical
Attachment_point	Deductible of the layer	Numerical
SCOR_Liab	SCOR's share of Liab_Amount	Numerical
Liab_Amount	Max liability for the contract	Numerical

Table 3.1: Original variables

The variables in gray were not kept, the reasons behind it is explained in the next section.

* The prefix CLM refers to the sub-universe "Claims" from which these variables originate from, however, they are variables related to the contract.

3.3 Selection of variables

This section will present the selection of variables, in order to obtain a database only containing variables that can describe the underlying risk.

Out of all the variables presented in the last section, some were not selected for various reasons:

- Used only to create new variables: CLM_Claim and all the dates;
- Too many missing values: Insured_Sector, Subsidiary_Event_Type (more than 75% of missing values);
- Too many different categories: Insured_Name, Subsidiary_Event, CLM_Cedent, CLM_Cedent_Ultimate_group, CLM_Cedent_group;
- They were not relevant to describe the underlying risk: Brokerage_Rate, Retrocession_Rate;
- Their use would bias the model as the information they contain is not supposed to be known when classifying the claim: CLM_Last_Position_Number, Closing_Date, Claim_Latent;
- Too many incorrect observations: Coutry_Cedent, Annual_Limit, Layer_cap, Attachment_point, SCOR_Liab, Liab_Amount.

Having too many different categories can be problematic. Indeed, when encoded using one-hot encoding, having a lot of different categories can make the database grow drastically bigger (one column per category).

3.4 Constructed variables

New variables needed to be constructed, in order to use other information that was not directly available from BO or needed treatments.

Variable's name	Desciption	Type of variable
Key	Concatenation of four other variables	Key
Geo_insured	Country of the insured	Categorical
Contract_Length	Time between inception date and expiry date (in years)	Numerical
SCOR_PML_%	Share of SCOR based on the PML	Numerical

- Key is the concatenation of: CLM_Claim, CLM_Subsidiary, CLM_Subsidiary_ledger and CLM_U/W_Year.
  Despite CLM - Claim being an individual Key in each subsidiary, it was not unique among the merge of all subsidiaries, therefore it had to be replaced by another Key, more precise, in order to identify each and every claim individually.

- Geo_insured: Combination of 2 other variables
  - Segment Geo N1, that is more precise globally but less precise in the USA;
  - Segment Geo SII, that is less precise globally but more precise in the USA.

  Geo_insured is equal to the most precise variable every time. Therefore, equal to: Segment Geo SII when Segment Geo N1 is equal to USA; and equal to Segment Geo N1 otherwise.

The variables with dates were replaced by lengths of time based on a claim life-cycle:

- Construction of Contract_Length:

$$Contract\_Length = \frac{DaysBetween(Inception\_Date, Expiry\_Date)}{365.25}$$

SCOR_PML_% is SCOR's share related to the Probable Maximum Loss, reconstructed using the PML variables:

- Construction of SCOR_PML_%:

$$SCOR\_PML\_\% = \frac{Probable\ Maximum\ Loss\ for\ SCOR}{Probable\ Maximum\ Loss\ total}$$

3.5 Indicators

In order to create homogeneous classes composed of claims with the same pattern of development, building indicators to describe these very patterns is necessary.

Are defined, for the following indicators:

- N is the year-end 2018;
- $development\ period_i$ is the i-th development period (on a quarterly basis);
- Inc or $Paid$ are the cash-flows for the incurred and the paid amounts respectively (equal to zero if there was no cash-flow for a certain development period);

The following indicators are candidates which can describe the risk:

- Incurred_N is the Amount of Incurred accumulated up to the year N:

$$Incurred\_N = \sum_{i=1}^{N} Inc_i$$

- TF_Incurred_N is the undiscounted duration for the incurred cash-flows (TF stands for "Time-Factor":) defined as:

$$TF\_Incurred\_N = \frac{\sum_{i=1}^{N} Inc_i \times development\ period_i}{\sum_{i=1}^{N} Inc_i}$$

Note: TF_Incurred_N could induce a division by zero if the settled amount of the incurred reaches zero. For example: An amount of incurred was registered, as a claim was notified. But after verification, the claim was not covered by SCOR, therefore the insurance company was not liable anymore, and the claims settles at 0. Which implies:

$$Incurred\_N = \sum_{i=1}^{N} Inc_i = 0$$

Moreover, for some claims, the settling process was not always exact, as some claim ended with a very low amount of incurred, resulting in a huge TF_Incurred_N. Therefore, to cancel that effect, all the TF_Incurred_N with an Incurred amount settling at less than 0.1€ (in absolute value) were forced to be equal to 0€.

- TF_Max_Incurred_N is a twist from the previous formula to capture the effects of claims developing and settling at zero. Instead of dividing by the sum of Incurred, the sum is divided by the maximum of cash-flows:

$$TF\_Max\_Incurred\_N = \frac{\sum_{i=1}^{N} Inc_i \times development\ period_i}{\max\limits_{1 \leq i \leq N} (Inc_i)}$$

– Paid_N is the Paid amount accumulated up to the year N:

$$Paid\_N = \sum_{i=1}^{N} Paid_i$$

– TF_Paid_N is the undiscounted duration of the Paid cash-flows, with the same adjustments made as for Incurred, defined as:

$$TF\_Paid\_N = \frac{\sum_{i=1}^{N} Paid_i \times development\ period_i}{\sum_{i=1}^{N} Paid_i}$$

– TF_Max_Paid_N is built to prevent division by zero from happening (cf: Incurred), instead of dividing by the sum of Paid, the sum is divided by the maximum of Paid cash-flows:

$$TF\_Max\_Paid\_N = \frac{\sum_{i=1}^{N} Paid_i \times development\ period_i}{\max\limits_{1 \leq i \leq N} (Paid_i)}$$

Hereunder is a fictive example to visualize how the Indicators values change regarding the development patterns of a claim:

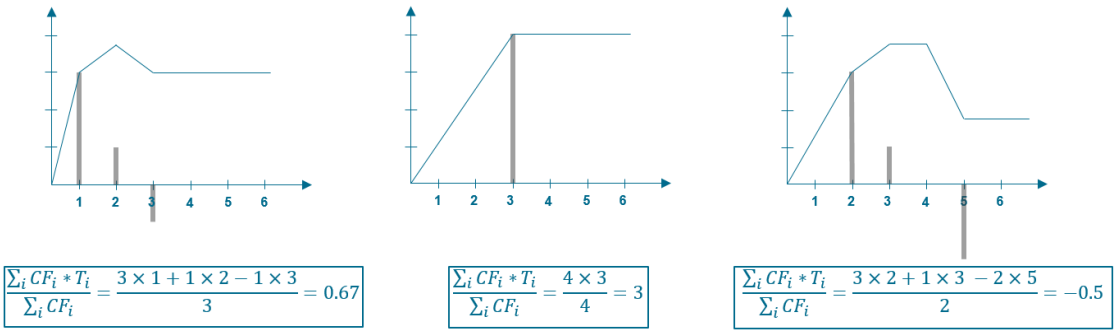


Figure 3.1: Impact of the development patterns on the values of the Indicators

Note: It is possible to see how a negative cash-flow, occurring after a long time, could make the duration negative.

Conclusion on the overview of the database

Throughout this chapter, the process of the building of the database was detailed. At this step the database is composed of 446345 lines, 19 variables + 1 Key, 4 Indicators, and two tables of development. Among these variables, 3 were created, 6 are numerical, 13 categorical and 1 (underwriting year) can be either used as categorical or numerical.

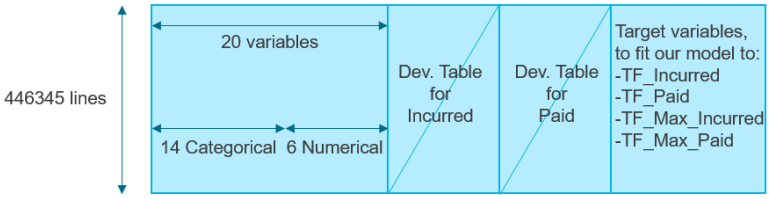


Figure 3.2: Composition of the whole database

Reduction of the database size: the Fire database

From now on, only the claims in the Line of Business (Lob) Fire will be kept. This line of business contains property related claims.

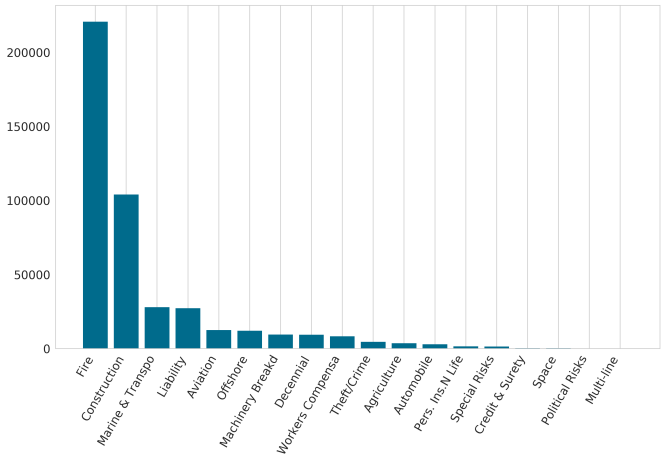


Figure 3.3: Distribution of the lines of business

Reducing the database was done in order to hasten the research process in both data pre-processing and algorithm-testing. The choice of the Fire Lob was made because it was the biggest Line of Business, containing 49.45% of the whole database (220659 out of 446345 lines). Once the whole process is established, it will be extended to the database containing all Lines of business.

Chapter 4

Missing value management

The data analysis showed that some variables had missing values. Most of the algorithms do not handle missing values, or fill them using basic techniques such as:

- Dropping the rows containing missing values;
- Replacing the missing values with the medians/means of the columns;
- Creating a new category 'Missing value' for categorical variables.

However, as the proportion of missing values is rather important and as some correlations between variables are strong, finding a way to use the correlations, could improve the quality of the database.

4.1 Filling missing data using dimension reduction

In order to find a better estimation of which category the missing data could have belonged to, the missMDA method, based on PCA and MCA, has been used. The idea behind the missMDA method is to perform successive PCA or MCA in order to adjust a missing value using the correlations between variables.

1. The algorithm will first replace the missing value by either the mean or the median (step 1.1), then it will perform a PCA/MCA on the filled dataset (step 1.2);
2. Using the result of the PCA/MCA it will adjust the value of the missing data by projecting it on the principal axis (step 2.1 & 2.2);
3. Then it will perform again a PCA/MCA on the adjusted dataset and so forth (step 3).

Herunder is an illustrated example of the missMDA algorithm:

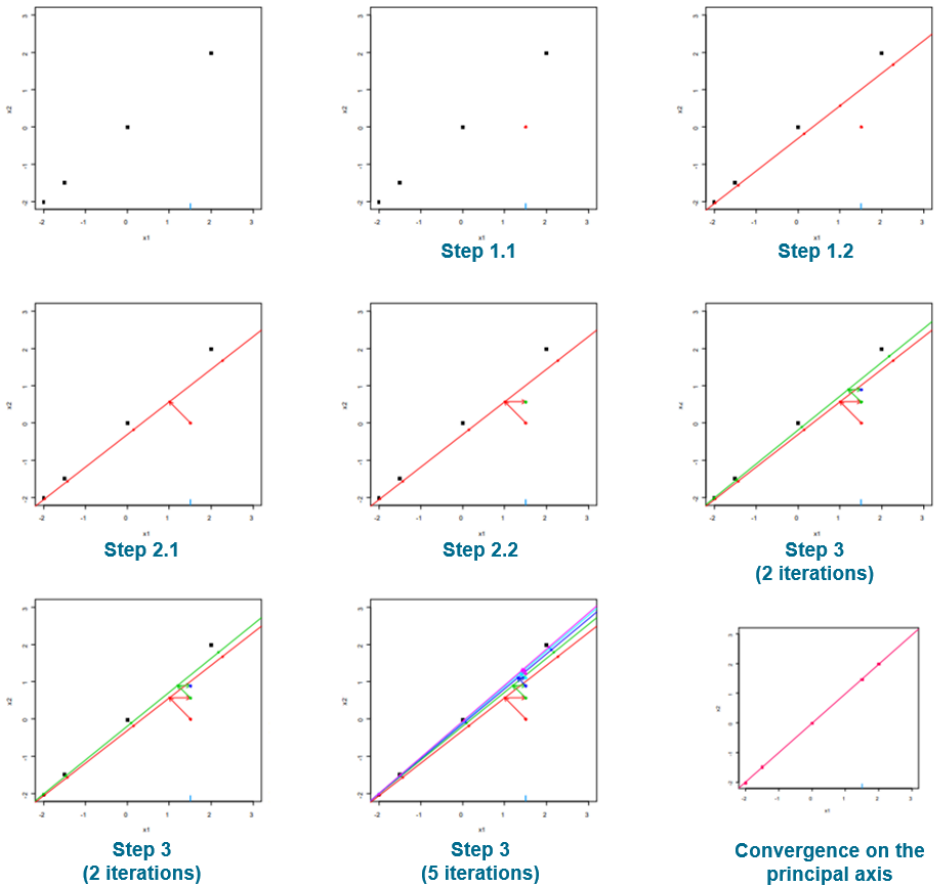


Figure 4.1: missMDA algorithm example

By projecting the missing data on the principal axis, it will artificially increase the explained variance of this axis. In order to reduce over-fitting, a method of cross-validation is implemented using either leave-one-out or k-fold validation. Hereunder is a graph showing the process for filling missing values, using the missMDA method, taken from the package's user guide:

The result of the MCA for missing data is a table of "estimated frequency", based on the correlation with other variables, of belonging in each category.

Note: It is worth noting here that the MCA will simply recreate structure of dependencies within the dataset. The only goal here is to fill the missing value using correlations, there is no direct link with the underlying risk!

4.2 Application on the Fire database

The MCA having been defined and having seen how it can be used to fill missing values, it will be used to fill the missing values present in the database. The database has several variables with missing values, the percentages of missing values are displayed hereafter:

1. Claim_Cause: 66.85%
2. Country_Claim: 0.87%
3. Risk_Nature: 55.33%
4. Sob: 10.19 %
5. Top: 22.44 %

The missMDA package, in R, has been used to fill Claim_Country, Sob and Top due to their low yet significant amount of missing values. Such processes could not be used to fill Risk_Nature and Claim_Cause as they had too many missing values. They were filled by a new category "Missing Value".

Hereunder is the plan of the database filling process for missing values:

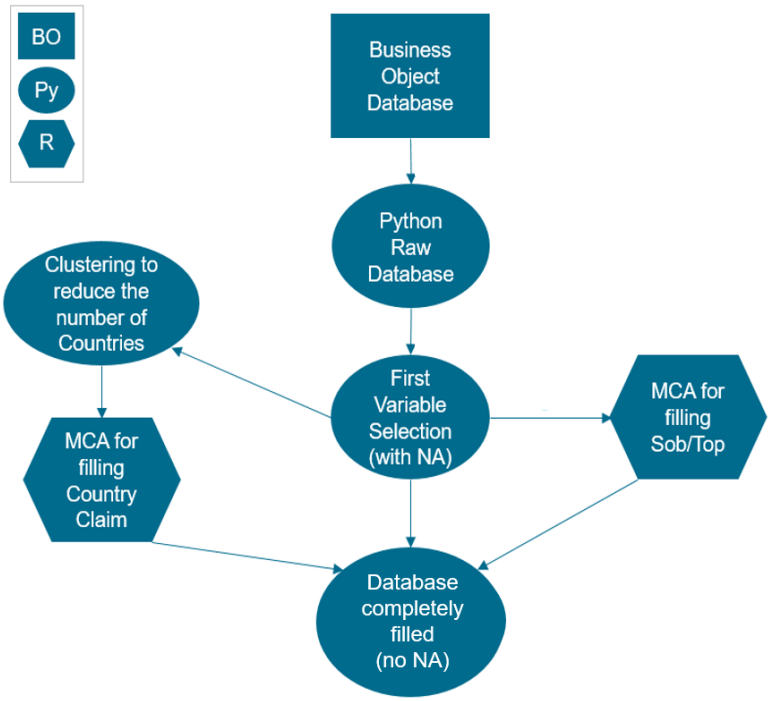


Figure 4.2: Process of filling missing Data

4.2.1 MCA for Sob & Top

To fill the missing data for both Sob and Top, an MCA on Sob, Top, Fac_Sector, Geo_Insured and Subsidiary was computed.

The number of categories is equal to:

$$11\ (Sob{+}1NA){+}12\ (Top{+}1NA){+}10\ (Fac){+}39\ (Geo\_Insured){+}9\ (Subsidiary) = 81$$

As seen in the previous section, a dimension in MCA does not correspond to a variable but rather to a category of a variable, since it is based on One Hot Encoding (as a column does not refer to a whole variable but just to one category for a variable). Therefore, the number of dimensions ranges from 1 to 79 (81 - 2 as the 'NA' categories for Sob and Top do not count as they will be filled by other categories). The percentages of explained variance for the MCA on this subset is as follows:

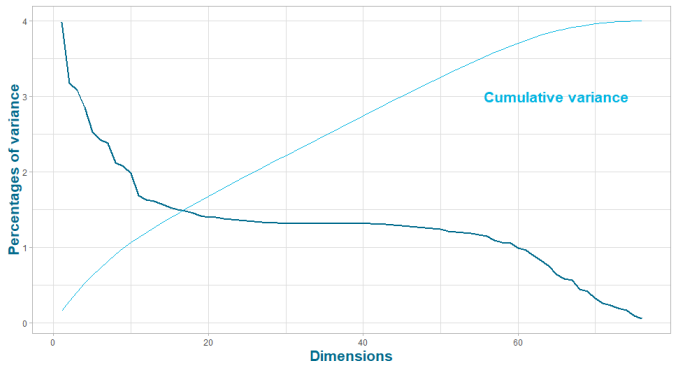


Figure 4.3: Percentages of explained variance for Sob/Top

Finding the optimal number of dimensions

The optimal number of dimensions is defined as the number of dimensions such that, when tested on cross-validation, the number of misclassifications is the lowest.

A function to find the best number of dimensions is available in the package missMDA (estim_ncpMCA). This method will randomly transform known data into missing values and predict its value using MCA. This is done in order to measure the sturdiness of the model and avoid over-fitting.

Unfortunately, this method could not be used to find the optimal number of dimensions for this very large dataset due to technical limitations. The maximum number of dimensions that could be computed is 10 and the best model with dimension ranging from 1 to 10 was the one with 10 dimensions.

As the function giving the optimal number of dimensions to use could not be used for Sob and Top, finding a new way to determine which number of dimensions to choose is necessary.

The method selected consists of computing tables of probabilities for a wide range of dimensions, and to select the one giving the highest maximum probability of belonging to a category, for both Sob and Top.

While the optimal number of dimensions function could not be computed for dimension higher than 10, the table of probabilities itself could be computed for dimension up to 54.

A table containing the minima, means and medians of maximum probabilities for different numbers of dimensions is available hereunder:

	Sob			Top		
Number of dimensions	**Min**	**Mean**	**Median**	**Min**	**Mean**	**Median**
10	26%	46%	45%	22%	43%	39%
20	27%	48%	43%	22%	47%	48%
40	28%	56%	48%	21%	50%	50%
54	26%	68%	64%	23%	54%	56%

Table 4.1: Sob/Top comparison between number of dimensions

As computing the MCA was not possible for numbers of dimensions higher than 54, due to technical limitations, the model using 54 dimensions has been chosen, as it is the one with the highest means and medians for the maximum probability.

4.2.2 MCA for Country_Claim

The Country_Claim variable has 201 categories, the MCA has been computed using Geo_Insured and Subsidiary, as they are all geographical variables and highly correlated to Country_Claim. The number of all categories is equal to:

$$201\ (Country\_Claim) + 39\ (Geo\_Subsi) + 9\ (Subsidiary) = 249$$

The percentage of explained variance for the MCA on this subset is as follows:

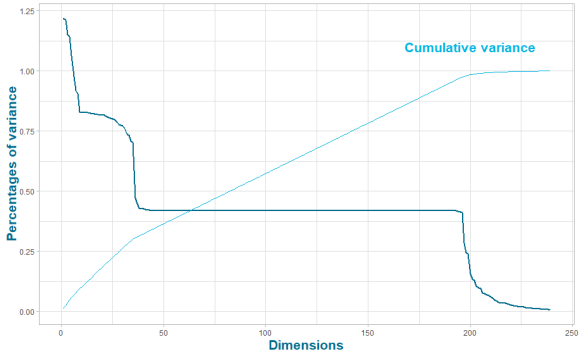


Figure 4.4: Percentages of explained variance for Country_Claim

With this current subset, the maximum number of dimensions cannot exceed 20, resulting in the highest probability for a missing value to belong in an existing class to be 20% on average.

In order to enable an MCA computation using more dimensions, it is necessary to reduce the total number of categories. To do so, the small countries will be grouped together among clusters based on the same 3 variables: Country_Claim, Geo_Insured and Subsidiary.

The small countries are defined as countries in Country_Claim appearing less than 500 times: it represents 158 countries out of 201 countries in total.

To predict the ideal number of clusters to merge the small countries into, clusters were built for number of clusters ranging from 1 to 30 and the inertia 2.2.3 was studied.

Below is a graph showing the decrease in total inertia as the numbers of clusters increases:

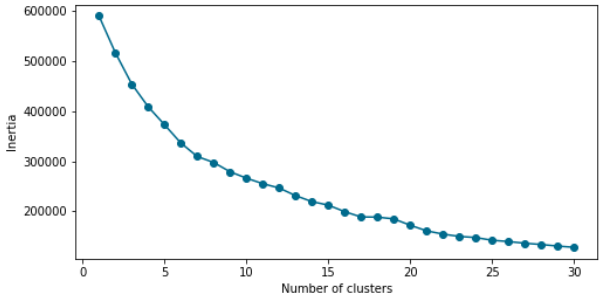


Figure 4.5: Clusters inertias on all countries

Using the Elbow Method, as defined by Sebastian Raschka in his Python Machine Learning manual [14], one way to find the best number of clusters is to choose the number after which the decrease in inertia is less important. In this case the decrease in inertia is less important after 5 clusters, therefore the chosen number of clusters to merge small countries into will be 5.

Once the optimal cluster built, a new database is created from the original dataset: for each line with a small country, their original country will be replaced by a cluster. By doing so, the new dataset contains 96 columns only.

$$96 = 249\ (original\ columns) - 158\ (number\ of\ small\ countries) + 5\ (number\ of\ clusters)$$

Now that the Country_Claim variable has been pre-processed; it is possible to compute the estim_ncpMCA with a maximum dimension of 90 (due to technical limitations). For a maximum number of dimensions of 90, the optimal number of dimensions obtained via cross-validation using the missMDA package is 65.

In order to justify the use of this number of dimensions, the same process used for Sob/Top will be applied. The minima, means and medians of the vector of maximum

probability, for different dimensions will be estimated, in order to confirm or infirm 65 as the best number of dimensions.

A table containing the minima, means and medians of maximum probabilities for different numbers of dimensions is available hereunder:

	Country_Claim		
Number of dimensions	**Min**	**Mean**	**Median**
20	14.4%	25.2%	24.0%
30	25.4%	37.0%	38.2%
35	28.5%	39.6%	41.0%
40	29.8%	41.6%	43.2%
45	31.4%	44.1%	45.8%
50	33.5%	47.3%	49.2%
55	32.5%	51.78%	54.1%
60	35.0%	52.1%	53.6%
65	30.0%	53.1%	54.5%
70	22.6%	37.6%	32.2%
75	14.9%	29.5%	21.0%
80	14.1%	28.6%	19.2%

Table 4.2: Country_Claim comparison between number of dimensions

The study of the vectors of maximum probabilities confirms that 65 is indeed a coherent choice for the number of dimensions, as it is the number of dimensions that has the highest median and the highest mean for maximum probabilities. Therefore, the table of frequencies obtained using 65 as the number of dimensions was kept to fill the missing values.

Moreover, Country_Claim also provides an example that the study of the maximum of probabilities for the table obtained via missMDA was coherent. The same number of dimensions was said to be the best for both methods.

Once the process of filling values is complete, a new variable, Country_Claim_no_NA, is created from the Country_Claim variable with the missing value replaced by the country associated with the highest probability. The small countries are untouched in the final variable, that way no cluster is contained in the Country_Claim_no_NA variable. The construction of the Country_Claim_no_NA variable is summarized hereunder:

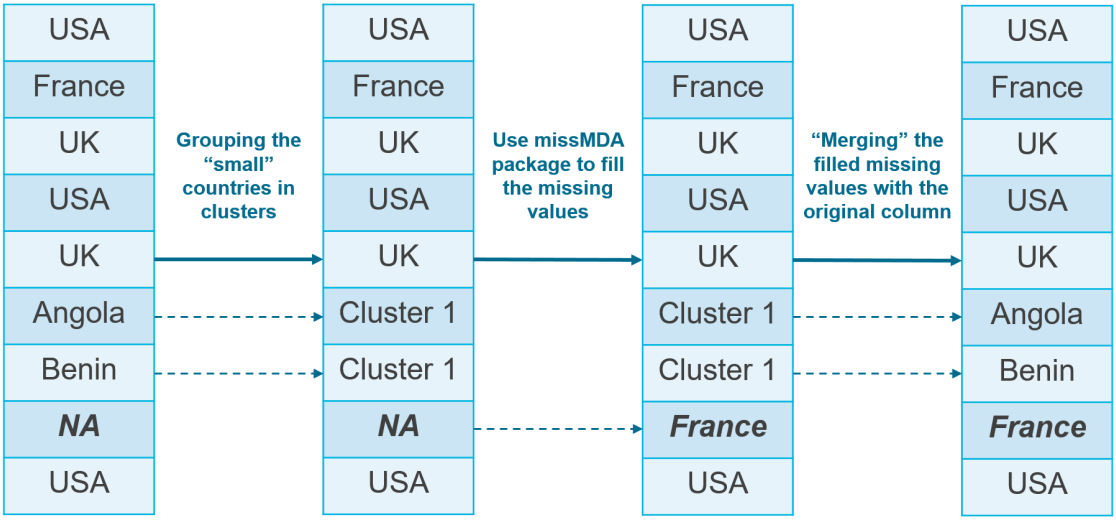


Figure 4.6: Construction of the Country_Claim_no_NA variable

Note: When applying the missMDA on the database containing clusters for small values, verifying that no missing value had its highest probability in a cluster was necessary. Otherwise, associating it to the most frequent country of the cluster could be a solution but it might induce a small bias in the filling process.

4.2.3 Risk_Nature & Claim_Cause

As Risk_Nature and Claim_Cause have a lot of missing values (54% and 67%), as well as a lot of different categories (117 and 103), computing a MCA to fill the missing data would not be efficient. In order to see whether a missing value actually holds some information for Risk_Nature and Claim_Cause, seeing its impact on the underlying risk is needed, here transcribed via TF_Incurred:

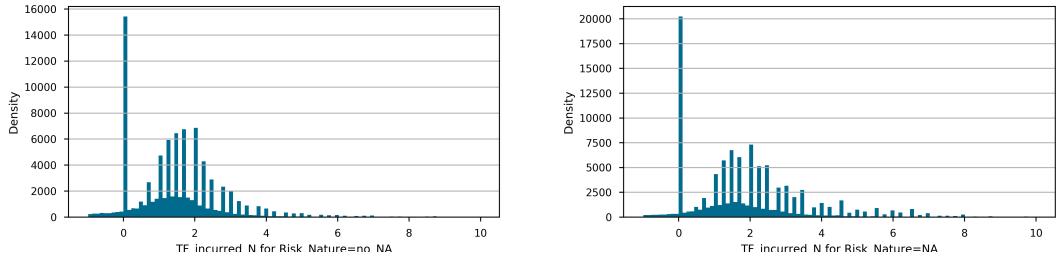


Figure 4.7: TF_Incurred_N density difference between Risk_Nature filled or NA

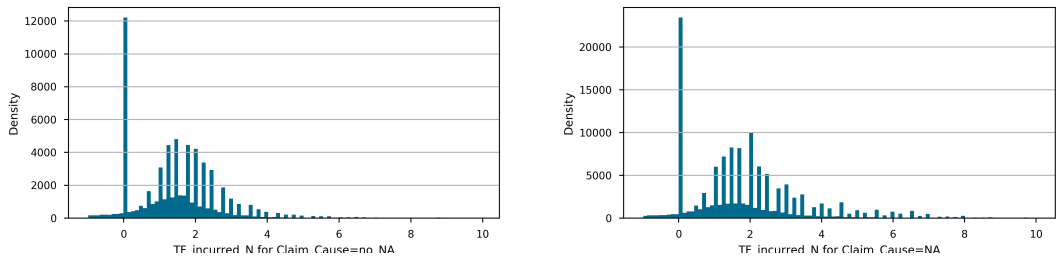


Figure 4.8: TF_Incurred_N density difference between Claim_Cause filled or NA

For both Claim_Cause and Risk_Nature, the Time Factor density's tail seems to be heavier for high values when there is missing data. In order to keep the behaviour of the missing data on the indicators, a new category: 'Missing_value' is created.

Conclusion on the missing value management

Different ways of dealing with missing values were detailed in this chapter, from simple ones, like the creation of a new category "Missing_value" to replace the missing values, up to more advanced methods using PCA/MCA to take into account correlations with other variables. A mix of clustering and dimension reduction was used to compensate the technical limitations of the missMDA package, making it usable on larger datasets.

4.3 Quality of the study database

A special care was taken when building the database to make sure that it contains no bias, this analysis was done regarding the three major axes stated in the Solvency II regulation documentation [19] and in the recommendations of the French Institute of Actuaries for the writing of the actuarial thesis [18]:

Accuracy The accuracy of the database was indeed necessary to justify the use and reliability of the database, each variable had to be carefully studied to verify that no bias could be left in the database. For example, different categories could, in fact, be related to the same category. Moreover, as the filling of some fields were mandatory to register a claim in SCOR's claims database, these mandatory fields were sometimes filled in a specific way, as a code for underwriters to specify that they did not know the required piece of information. Keeping these incorrectly filled fields could induce a bias in the algorithms, hence why a careful analysis of each variables was necessary.

Completeness At first, the database was far from being complete, indeed some variables contained a lot of missing values. The variables containing too many missing values were dropped, and the one containing few missing values were filled using the techniques described in the chapter regarding the missing values [4].

Appropriateness The appropriateness of the use of the variables was studied at different steps during the database construction. Firstly, the selection of variables was done looking at whether the variable could describe the underlying risk or not. Secondly, the selection of variables was presented to experts, to see whether they agreed or not on the reliability of the selection of variables and if indeed the selected variables could describe the underlying risk. By doing so, some variables were taken out of the selection because it was known that the way they were used at SCOR made them irrelevant regarding the underlying risk. Moreover, when building the new segmentations, making sure that no biases were contained in the variables used is mandatory.

Conclusion on the presentation of the study database

This part was focused on the presentation of the Fire database, its construction and how to improve its quality. After all these steps, the Fire database contains no more missing values and is ready to be used to build new segmentations.

Part III

Challenging the existing segmentation

Introduction

In this part will be detailed the different steps taken to build the challenging segmentations.

The segmentations are built using either supervised or unsupervised algorithms, five different types of segmentations were studied:

- Two segmentations based on decision trees fitted to either Incurred or Paid indicators;
- Three segmentations based on clustering on indicators, variables, or both.

From all these models, only one segmentation per type has been kept. The kept segmentation was chosen regarding the homogeneity of the classes, quantified by the MSE on the indicators.

Once chosen, the five segmentations are compared to the Actuarial Segmentation and the aggregated segmentation (segmentation composed of only one class) on their ability to predict the actual cash-flows and IBNR / Reserves amounts[a].

[a] All the amounts predicted have been rescaled for confidential reasons.

Chapter 5

Building the new segmentations

The new segmentations were built using two different methodologies:

1. Decision Tree regressors, in which each leaf value was associated to a distinct class;
2. Clustering methods based on either the variables or the indicators, or both.

For each type of segmentation (2 based on Trees and 3 based on Clustering methods), only one segmentation will be kept, this segmentation will be chosen regarding the errors of prediction for various ranges of parameters.

5.1 Selection of the Fire sub-database

The previous part focused on the Fire database, composed of claims belonging in the Line of Business "Fire" (220,659 out of 446,345 lines). Beyond the selection of the Lob=Fire database, a new selection of lines is necessary to prevent biases in the models:

1. Dropping the most recent underwriting years due to bias in the indicators values: As the indicators are homogeneous to durations, keeping the most recent years introduced a bias in the indicators. When looking at the aggregated triangle (the triangle containing all claims of the Fire database in the same segment), the claims could be considered fully developed after 3 years (just for the Fire Database, as the Fire Line of business is rather short-tail). Therefore, to prevent a bias in the value of the indicators, only the claims having underwriting years before 2015 (included) have been kept.

This effect can be seen on the median values for TF_Incurred_N and TF_Paid_N in the Fire dataset:

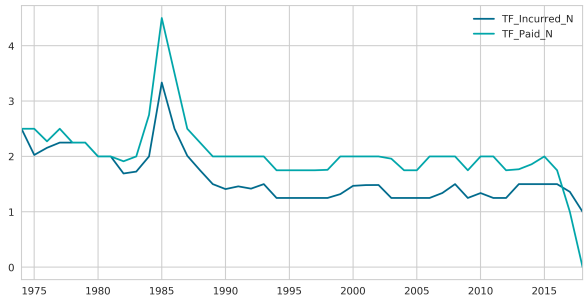


Figure 5.1: Underwriting year bias in the median values of the indicators

2. Dropping the underwriting years before 2003:
   Due to an eventual bias in the claims management after SCOR's downgrade, and the drop in the number of claims in 2002;

3. Dropping the extreme values for the indicators to prevent the whole error being due to only a few claims:
   The segmentations being built based on the value of the indicators, some adjustments needed to be made in order to better see the impact of the segmentation on the error. Without these adjustments, most of the error would be due to claims with very high values for the indicators, and the study of the homogeneity would be less clear.

Will be kept in the Database only the claims having:

- TF_Incurred $\in [-25; 25]$;
- TF_Paid $\in [-10; 25]$;
- TF_Max_Incurred $\in [-10; 25]$;
- TF_Max_Paid $\in [-10; 25]$;
- Underwriting years $\in [2003; 2015]$.

By doing so, the resulting database contains 38,605 lines:

	Number of lines
Fire Database	220,659
- Claims with UW_Y $<$ 2003	- 174,219
- Claims with UW_Y $>$ 2015	-5940
- Adjustment for TFs	-1895
= Final number of claims in the sub-database	**38,605**

5.2 Construction of the Actuarial Segmentation

The Actuarial Segmentation is constructed from variables available at the contract level and from expert judgement.

A segment could be composed of the contracts belonging in a precise Line of Business and a precise subsidiary. For example: MFPFIRE01 contains property related contracts underwritten in the UK.

There are 42 different classes in the Actuarial Segmentation (in the Fire Database), hereunder is the distribution of the classes:

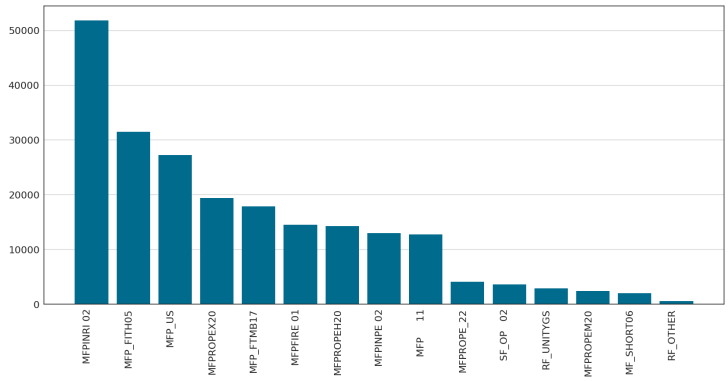


Figure 5.2: Actuarial Segmentation classes

Only the classes containing more than 500 claims are displayed (27 out of 42 categories containing 2613 lines have been omitted).

5.3 Framework of building the challenging segmentations

The Solvency II legislation states that the segmentation has to form groups of homogeneous risk. Moreover, as to better predict the development patterns of the claims, the classes should contain claims with the same kind of pattern of development. In order to compare the segmentations with each other, the following two criteria have to be taken into account:

1. Homogeneity regarding the indicators;
2. Similarity regarding the pattern of development.

The homogeneity of the classes will be quantified by studying the intra-variances of the classes. For each class and for each indicator (TF_Incurred, TF_Paid, TF_Max_Incurred and TF_Max_Paid), a measure of error will be computed.

The measure of error retained is as such:

$$Mean\ Square\ Error = MSE = \frac{\sum(Predicted\ Value - Value\ of\ the\ TF)^2}{Number\ of\ claims}$$

This measure will tell how homogeneous the classes are regarding the indicators: the lower the MSE, the lower the intra-variance groups.

The study of the MSE is the first way to compare the segmentations with each other. Since this error will be available on all four indicators, the choice will be made looking at the four errors. The choice will consider both the reduce of the MSE and the complexity of the model.

Once the model chosen, Incurred and Paid amounts will be estimated using a Chain-Ladder method, fitted to the previous years only. The estimated amounts will then be compared to the actual values. The comparison will be done on the segmentations' ability to predict the cash-flow of Incurred/Paid for the year 2018, as well as IBNR/Reserves for the three and five previous years.

Beyond the previously introduced criteria, based on easily quantifiable statistical results, the segmentations should also be coherent regarding claims management in general. To study this aspect, the composition of each class will have to be studied, to see whether it makes sense to use these classes as claims segmentations and if they form homogeneous risk groups.

Reminder: To prevent over-fitting, all the errors displayed in this paper are the errors obtained for a test database, on which the model has not been fitted. In order to have a fair comparability with the Actuarial Segmentation and the segmentation which has only one class, the error for these models will be the one regarding only the claims of the test database, composed of 30% of the claims, randomly selected.

5.4 Segmentations based on Decision Trees

For the classes based on Decision Trees, a regression tree was fitted to an indicator: either TF_Incurred_N or TF_Paid_N. No trees were fitted to TF_Max_Incurred_N and TF_Max_Paid_N but their MSE were used to choose the best parameters. The model will split the database in order to create leaves that contain claims with similar values for the fitted indicator.

The choice of the decision tree was made because it is the only algorithm which allows the user to have complete control over the number of classes. The goal of this study being to create new segmentations that will later be used to estimate reserves using Chain-Ladder, having control over the number of observations per class is necessary.

Indeed, a certain volumes of claims per underwriting year is necessary to compute a coherent Chain-Ladder. For the same reason, the variable "underwriting year" was not used, as claims from all underwriting years are necessary to compute the Chain-Ladder.

By choosing the maximum number of leaves, as well as the minimum number of observations per leaf, different trees were fitted for each indicator.

In order to estimate the error on all indicators, the following method was used:

1. Fit a regression Tree to an indicator and consider each leaf as a class;
2. The tree, by grouping claims with similar values for the indicator, will create homogeneous classes regarding this indicator.
3. Associate the means per class for the other indicators as the values for that class (therefore the same for all claims belonging to the same class);

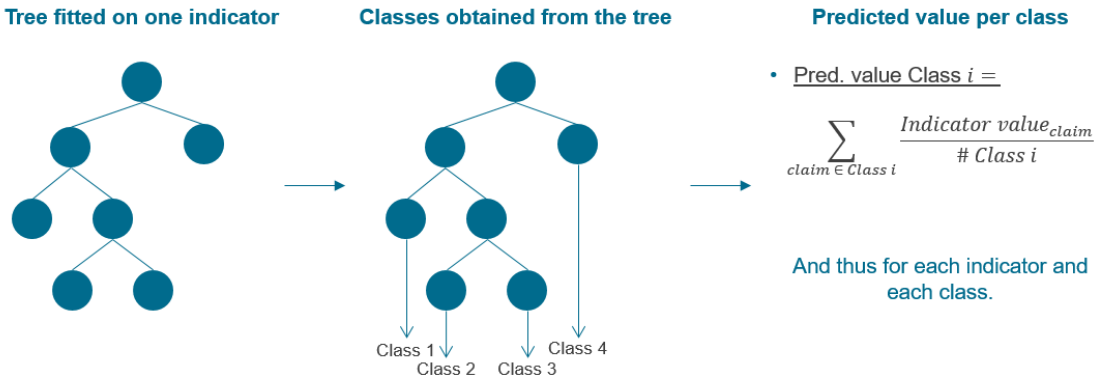


Figure 5.3: Predicted value assignment process for tree-based segmentations

Now that a predicted value is available for each indicator, all the MSE can be computed for the model.

In order to have a fair comparison between the new segmentations and the Actuarial Segment, having a somewhat similar number of classes is necessary. The limit was

set looking at the number of classes of the Actuarial Segment, for the subset of the Fire database: the classes of the Actuarial Segment having more than 100 claims is 15, therefore, the maximum number of classes for our models will be $15 \times 2 = 30$, to have the same order of magnitude.

Regarding the number of minimum observations per leaf, a minimum of 100 was fixed. Below this number, the number of claims per underwriting year might become too little to compute a Chain-Ladder. The upper limit was set at 5000, which was more than enough considering the size of our dataset and the range of the maximum number of leaves.

5.4.1 Study of the MSE for trees fitted to TF_Incurred_N

Hereunder are the errors for each indicator, for classes built by fitting a tree to TF_Incurred_N. In order to have a point of comparison with the Actuarial Segmentation, the MSE of the Actuarial Segmentation and the variance of the indicator are also displayed. On the X-axis are the maximum number of claims per leaf, the Y-axis is the negative MSE, and the different lines refer to different maximum number of classes:

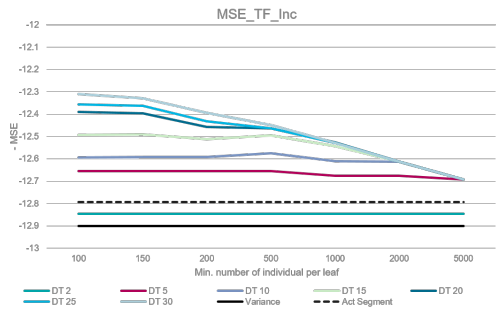


Figure 5.4: MSE for TF_Incurred

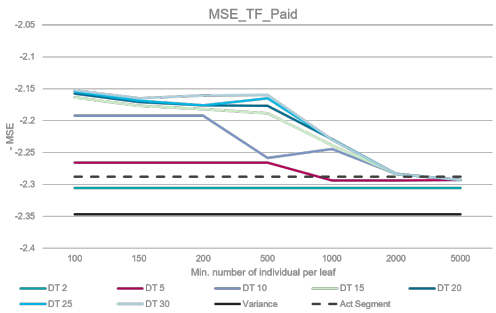


Figure 5.5: MSE for TF_Paid

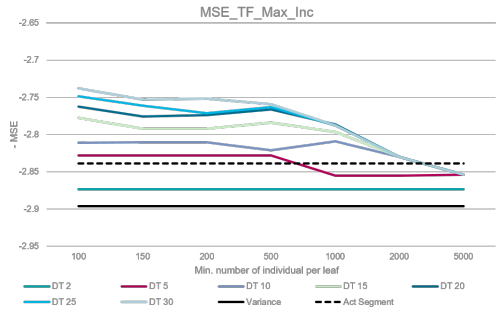


Figure 5.6: MSE for TF_Max_Incurred

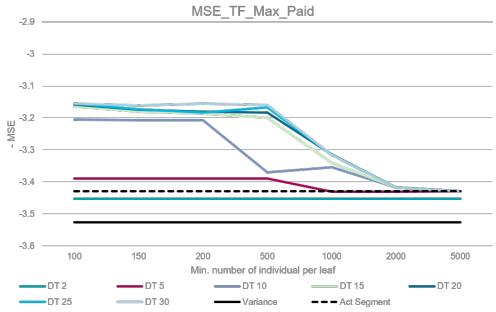


Figure 5.7: MSE for TF_Max_Paid

As the minimum number of claims per leaf grows higher, the differences between the models with different maximum number of classes grows shorter. Indeed, the constraint

on the minimum number of claims per leaf can reduce the number of different classes, up to the point where the maximum number of leaves have no impact on the final model. For example, the models with 15 and 30 classes with 2000 observations minimum per leaf are the same.

There is a general trend: the higher the maximum number of leaves and the lower the minimum number of observations per leaf the lower the MSE. However, the improvement brought by having a higher number of classes is less important after 20 classes.

For these reasons, the chosen model for the Tree fitted to TF_Incurred_N is the model with 20 as the maximum number of classes and 100 as minimum number of claims per classes. This model will be referred as: DT_20_100_Inc.

5.4.2 Study of the MSE for trees fitted to TF_Paid_N

Hereunder are the errors for each indicator, for classes built by fitting a tree to TF_Paid_N. In order to have a point of comparison with the Actuarial Segmentation, the MSE of the Actuarial Segmentation and the variance of the indicators are also displayed. On the X-axis are the maximum number of claims per leaf, the Y-axis is the negative MSE, and the different lines refer to different maximum number of classes:

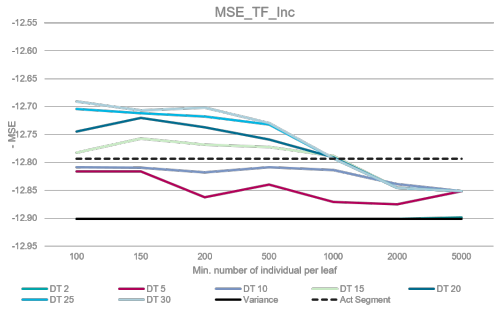


Figure 5.8: MSE for TF_Incurred

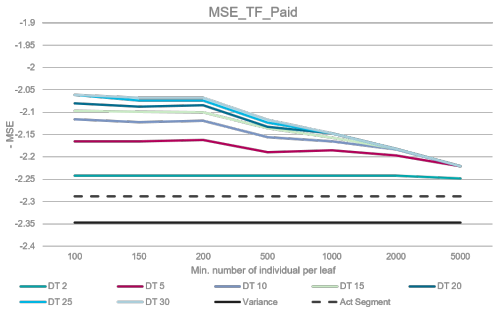


Figure 5.9: MSE for TF_Paid

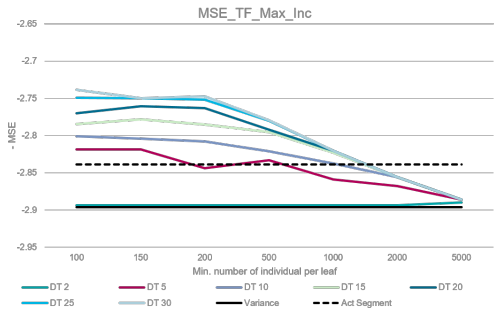


Figure 5.10: MSE for TF_Max_Incurred

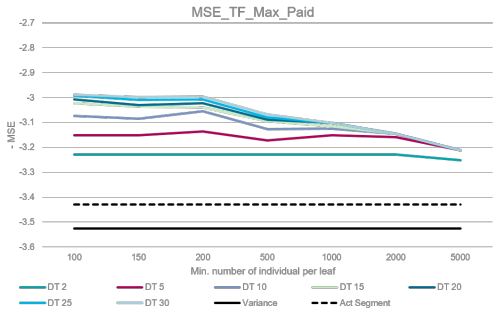


Figure 5.11: MSE for TF_Max_Paid

Same comments than for the Tree fitted to TF_Incurred_N, with the only difference that for this tree, an over-fitting seems to occur for the model with 20 classes, indeed, the model with 150 as the minimum of claims per leaf is better than the one with 100. For this reason, the chosen model for TF_Paid_N is the one with 20 classes and 150 claims minimum per leaf, it will be referred as: DT_20_150_Paid.

Same comment regarding the constraint on the minimum number of claims per leaf: the models with 15 and 30 classes with 2000 observations minimum per leaf are the same.

5.4.3 Conclusion of the MSE analysis for the tree-based models

The classes built by fitting a tree to an indicator also reduce the error for the other indicators. It is worth noting that the difference between the MSE of the Actuarial Segment and the variance (which correspond to an MSE for a segmentation having only 1 class) is very small, and that the actuarial segmentation is easily outperformed regarding the MSE, which could have been expected as these models are only built on these indicators. For the reasons stated previously, the chosen segmentations are:

- DT_20_100_Inc: 20 classes and 100 observations minimum per leaf for the Tree fitted to TF_Incurred_N;
- DT_20_150_Paid: 20 classes and 150 observations minimum per leaf for the tree fitted to TF_Paid_N.

5.5 Segmentations based on Clustering

In this section, the segmentations were built using clustering methods. Once the cluster built, each cluster will be associated to a class.

However, as the clustering methods are unsupervised algorithms with, by essence, no good or bad results, how can the quality of a cluster compared to another be measured? Indeed, clustering methods, will build a segmentation composed of homogeneous classes regarding the variables used to build the classification, but how to be sure that the classes are homogeneous risk-wise? And how to compare it with the Actuarial Segmentation?

To enable a kind a comparability between the segmentation based on clusters and those based on supervised methods, the following method has been followed:

1. Building of the classes using clustering on the train database;
2. Predicting the classes for the test database, only using the descriptive variables with four different algorithms[1]:
   - Neural Networks: with 1 hidden layer composed of 100 neurons;
   - Decision Tree: with no limit regarding the depth or the number of leaves;
   - Random Forest: composed of 10 trees, fitted to bootstrap samples, comparing $\sqrt{number\ of\ variables}$ at each split, with no limit regarding the depth or the number of leaves;
   - Gradient Boosting: with a learning rate of 1% and 100 iterative trees each having a maximum depth of 3.
3. Associating all the claims in each class, with the mean of the values of the class it belongs to, as predicted value, and thus for each indicator.

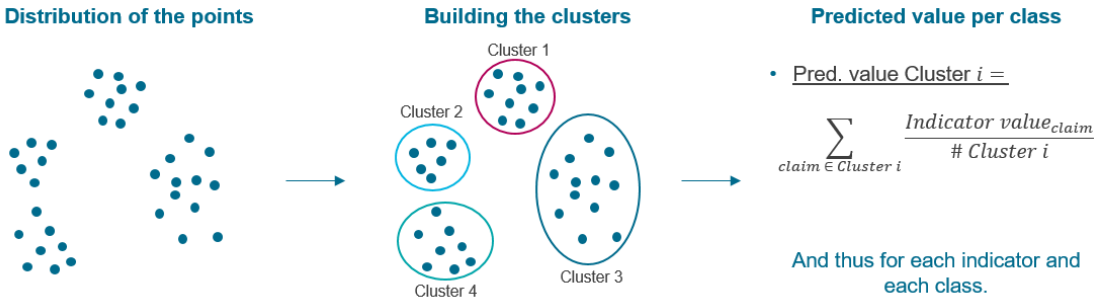


Figure 5.12: Predicted value assignment process for cluster-based segmentations

Now that a "predicted value" has been computed for all the indicators, it is possible to measure the MSE for these segmentations.

[1] The parameters chosen were simple, the goal was to see how the algorithms compared to each other, not to optimize each algorithm.

The segmentations based on clusters are based on either:

- All the indicators, only numerical values ⇒ k-means algorithm;
- All the variables, categorical and numerical values ⇒ k-prototype algorithm;
- Indicators + variables, categorical and numerical values ⇒ k-prototype algorithm.

5.5.1 Study of the MSE for clusters based on the indicators

In this section, the clusters will be based on the indicators only. The classes are built using k-means on TF_Incurred_N, TF_Paid_N, TF_Max_Incurred_N and TF_Max_Paid_N. Once the classes built, the MSE for each indicator was estimated. Hereunder are the MSE for these clusters, with the number of classes on the x-axis:

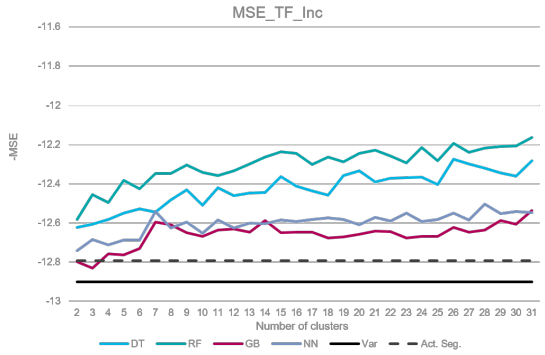


Figure 5.13: MSE for TF_Incurred

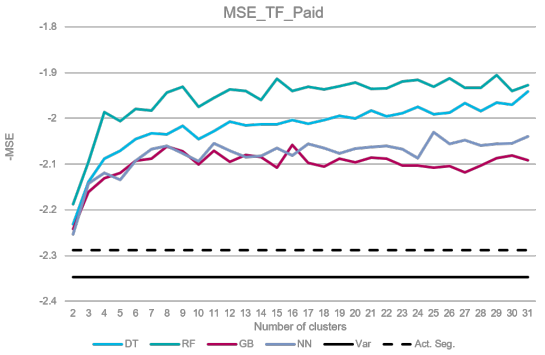


Figure 5.14: MSE for TF_Paid

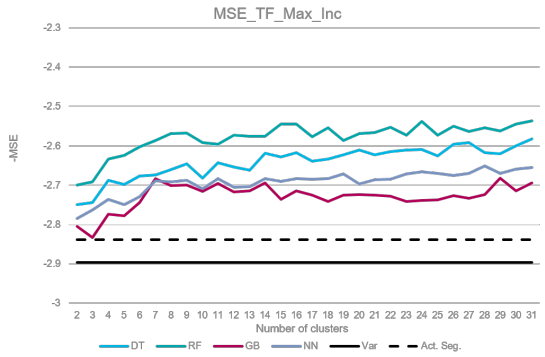


Figure 5.15: MSE for TF_Max_Incurred

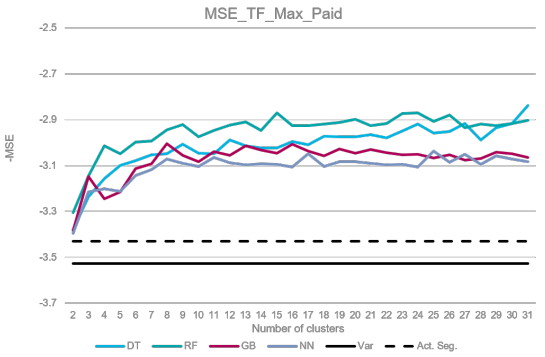


Figure 5.16: MSE for TF_Max_Paid

For every indicator, the best MSE was the one obtained via predicting the classes using Random Forest. The different algorithms used to predict the classes have a great

impact on the MSE values, which led to think that the clusters based on Indicators are hard to predict.

The decrease in error is important up to 8 clusters but seems to level out after that. A peak in improvement seems to occur between 3 and 4 clusters (especially for Paid indicators). For that reason, the number of clusters chosen for the clustering methods based on indicators is the segmentation with 4 classes.

5.5.2 Study of the MSE for clusters based on variables

In this section, the clusters will be based on the variables only. The classes are built using k-prototypes on all the dataset variables, once the classes built, the MSE for each indicator was estimated. Hereunder are the MSE for these clusters:

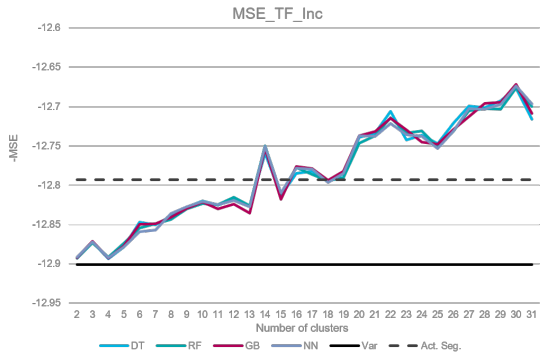


Figure 5.17: MSE for TF_Incurred

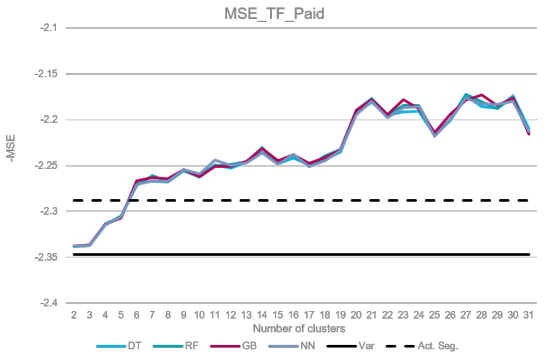


Figure 5.18: MSE for TF_Paid

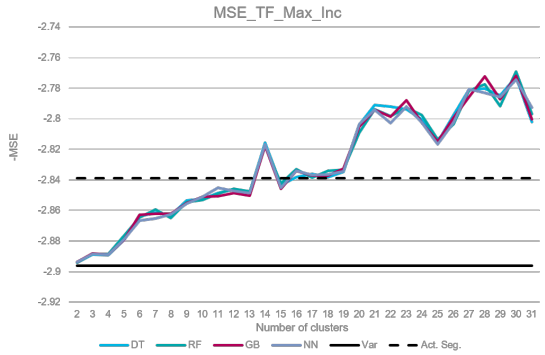


Figure 5.19: MSE for TF_Incurred

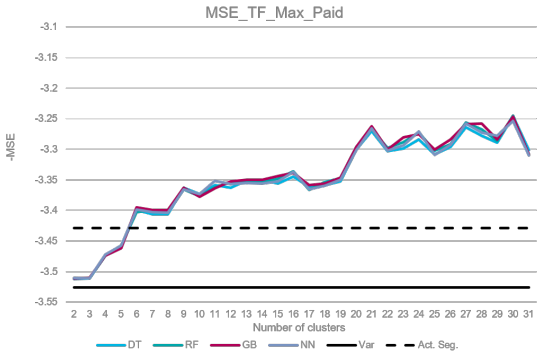


Figure 5.20: MSE for TF_Paid

Compared to the clusters based on the indicators, the error reduction is less important for the clusters based on the variables, which is logical, as they are not directly based on the indicators themselves. Interestingly, the clusters based on variables outperform the

actuarial segmentation at around 20 clusters, which is roughly the number of classes of the Actuarial Segmentation.

The MSE decreases steadily for classes up to 9, after that the reduce in MSE is lower (except a peak at 14 for TF_Incurred_N and TF_Max_Incurred_N). A great reduction is also present near 20.

The chosen number for this segmentation will be 9, while 14 provided a greater improvement for TF_Incurred_N and TF_Max_Incurred_N, the decrease was lower for TF_Paid_N and TF_Max_Paid_N.

Moreover, the impact of the algorithm is more important for this segmentation, all the algorithms end up having roughly the same MSE and thus for all the points, which seems to indicate that all the algorithms came up having the same results for the classification of the claims.

5.5.3 Study of the MSE for cluster-based on both the variables and indicators

In this section, the clusters will be based on the indicators and the variables. The classes are built using k-prototypes on all the dataset variables and the four indicators, once the classes built, the MSE for each indicator was estimated. Hereunder are the MSE for these clusters.

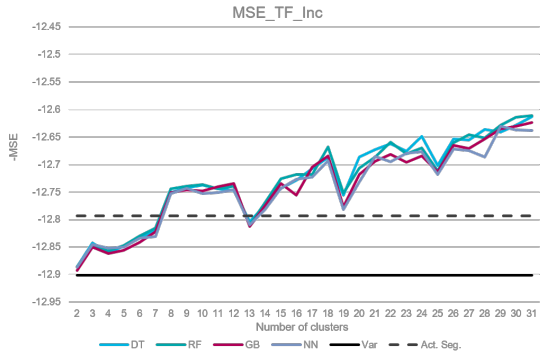


Figure 5.21: MSE for TF_Incurred

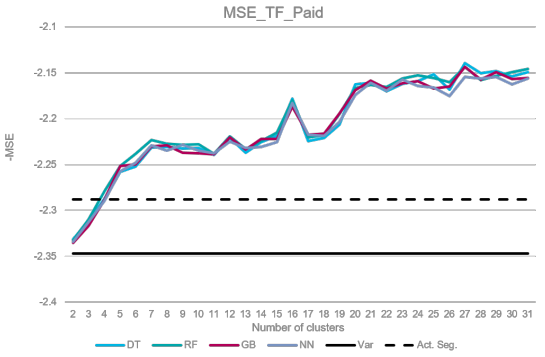


Figure 5.22: MSE for TF_Paid

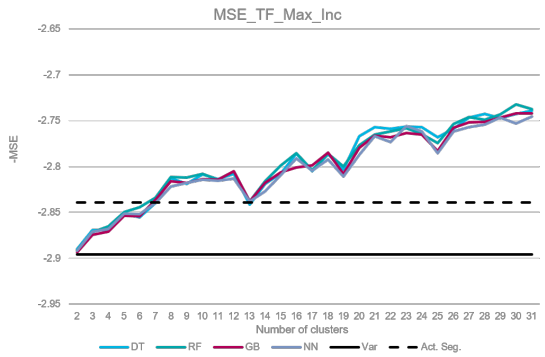


Figure 5.23: MSE for TF_Max_Incurred

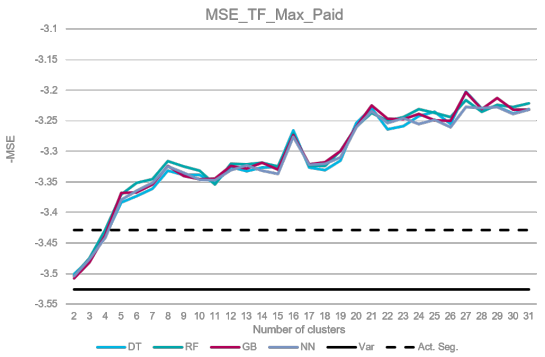


Figure 5.24: MSE for TF_Max_Paid

The clusters based on the variables and the indicators are very similar to the ones based on variables only. Overall, the clusters based on both slightly outperform the ones based solely on the variables, and the segmentations based on both seem to be less predictable as it is possible to see differences in the MSE due to the difference in classification of the different algorithms.

For the four indicators, the MSE decreases steadily up to around 23 classes, compared to the clusters based on variables only, the error seems to be less volatile, for that reason, and to have a wide range in the number of clusters, the number of classes chosen for the segmentation based on both variables and indicators is 23.

5.6 Conclusion on the choice of the models

The selected model will now be referred as:

- DT_20_100_Inc: Segmentation based on a Decision Tree fitted to Incurred, with 20 leaves and 100 observations minimum per leaf;
- DT_20_150_Paid: Segmentation based on a Decision Tree fitted to Paid, with 20 leaves and 150 observations minimum per leaf;
- 4_clusters_on_4_TFS: Segmentation based on clusters on the 4 indicators with 4 classes;
- 9_clusters_on_var: Segmentation based on clusters on the variables with 9 classes;
- 23_clusters_on_vartf: Segmentation based on clusters on variables and the 4 indicators with 23 classes;

Hereunder is the table of the MSE for each indicator and for each model on the test database. The percentages below the MSE for the new models are the improvements

regarding the Actuarial Segmentation MSE:

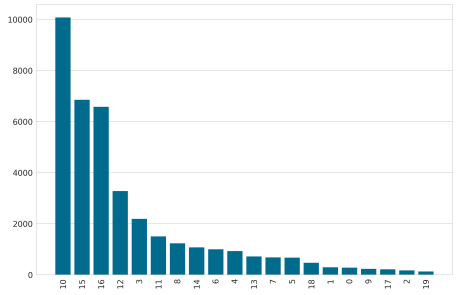
$$Improvement = \frac{New\ Seg_{MSE} - Act\ Seg_{MSE}}{Act\ Seg_{MSE}}$$

	TF_Incurred	TF_Paid	TF_Max_Inc.	TF_Max_Paid
One Segment (Variance)	12.901	2.347	2.896	3.526
MSE Actuarial Segment	12.793	2.288	2.839	3.429
DT: 20 leaves / 100 min fit on Inc.	12.390 (-3.15%)	2.157 (-5.73%)	2.763 (-2.68%)	3.158 (-7.90%)
DT: 20 leaves / 150 min fit Paid	12.720 (-0.57%)	2.088 (-8.74%)	2.761 (-2.75%)	3.030 (-11.64%)
Cluster on 4 TFs: 4	12.472 (-2.51%)	2.000 (-12.59%)	2.6278 (-7.44%)	3.0585 (-10.80%)
Clusters on variables: 9	12.776 (-0.13%)	2.241 (-2.05%)	2.833 (-0.21%)	3.342 (2.54%)
Clusters on variables + TFs: 23	12.664 (-1.01%)	2.186 (-4.46%)	2.771 (-2.40%)	3.280 (-4.35%)

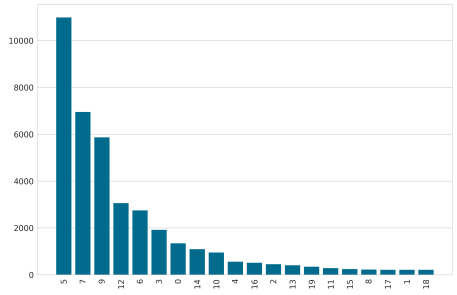
Table 5.1: Comparison of the MSE (test database) of the chosen models

Looking at this table it is possible to see that all the segmentations (except 9 cluster on variables for TF_Max_Paid) reduce the MSE for all the indicators, therefore, they should all compose more homogeneous risks groups than the Actuarial Segmentation, based on these 4 indicators.

Distribution of the classes in the new segmentations

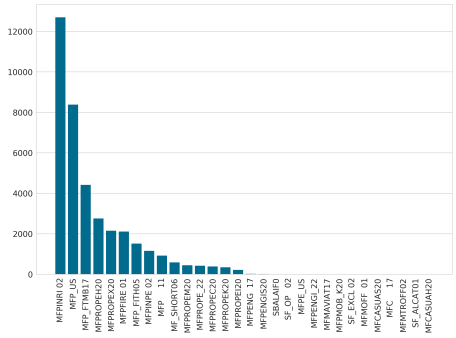


Classes of DT 20 100 Inc

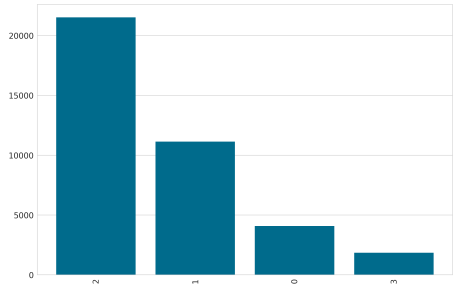


Classes of DT 20 150 Paid

For the tree-based segmentations, the three biggest classes contain respectively 60.92% and 61.73% of the whole database.

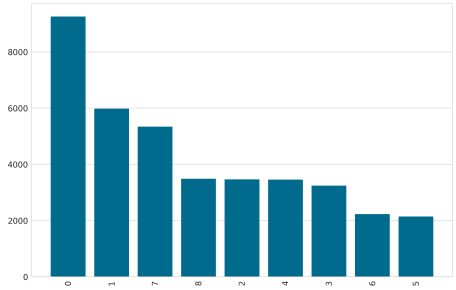


Classes of the Actuarial Segmentation

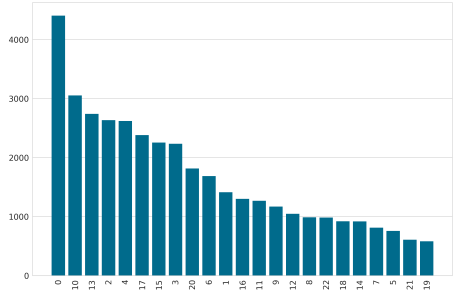


Classes of 4 Clusters on 4 TFs

For 4_Clusters_on_4_TFs: 55.75% of the claims are contained in the biggest class.



Classes of 9 Clusters on variables



Classes of 23 Clusters on variables and TFs

The classes based on clusters of variables are more evenly distributed compared to the other new segmentations.

Chapter 6

Study of the quality of prediction for the segmentations

This chapter detail the challenge the quality of prediction of the segmentations selected in the last chapter. This comparison challenges the segmentations on two criteria:

1. The segmentation's ability to predict 2018 cash-flow: a pattern is fitted per class, on the triangle as at year-end 2017, to predict the 2018 cash-flow;
2. The segmentation's prediction regarding the total amount of IBNR. To predict the Ultimates amounts: a pattern is fitted per class on a triangle made of closed claims, as at year-end 2015 (three years taken out) or year-end 2013 (five year taken out). From these Ultimate amounts, the IBNR and Reserves are obtained and compared to the actual values.

The results on these two criteria determine which segmentation is the best among all the models.

Once the best segmentation selected, a study of its composition will be presented, to analyse the coherence of the output of the algorithm.

6.1 Predicting the 2018 cash-flow

In this section, the segmentations will be compared to each other based on their ability to correctly predict the cash-flow of 2018: the year 2018 will be taken out and a pattern will be fitted for each class to predict the cash-flow for the year 2018. Finally, the predicted amounts will be compared to the actual values to see how good the predictions are.

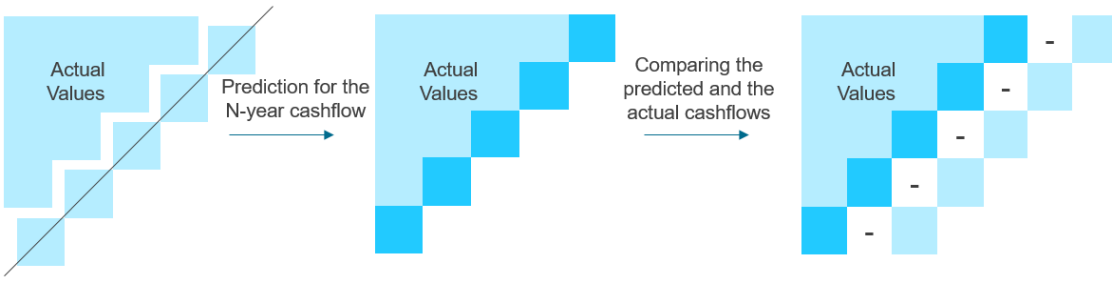


Figure 6.1: Process used to compare the quality of the N cash-flow prediction

6.1.1 Classifying the most recent years: 2016 and 2017

The segmentations were built on a database composed of claims with underwriting years ranging from 2003 to 2015. This was done so as to prevent a bias in the models, due to a "cut" in the possible values for the indicators of the recent claims.

In order to have a fair comparison of the models, as well as to include the predictability of the classes in the comparison process, classifying the new claims is necessary. To classify the most recent claims, the segmentation will be considered built, and the claims of the most recent contracts (underwritten in 2016 & 2017) will be classified in the new classes.

6.1.1.1 Classification for the tree-based models

The trees fitted on the years 2003-2015 give sets of rules to predict the classes of all claims. Therefore, the tree fitted to the years 2003-2015 will be used to classify the claims of the years 2016-2017.

6.1.1.2 Classification for the cluster-based models

Different algorithms were tested to see which one would have the best accuracy in predicting the classes, these tests were done on the 2003-2015 database, using a 10-fold cross-validation[1].

[1]The choice of keeping an unstratified k-fold and a non-weighted accuracy was made voluntarily, as to analyse the impact of the size of the classes "by hand".

An optimisation of the algorithm was done on the following parameters:

- Decision Tree: max depth of the tree;
- Random Forest: max depth of each tree, proportion of variables to use per tree and number of trees;
- Neural Network: number of neurons per layer and number of layers;
- Gradient Boosting: learning rate and depth of the tree.

The following accuracies were obtained for the segmentations based on clusters:

	Model	Average Accuracy	Standard deviation	Computing time
4 Clusters on 4 TFs	**Decision Tree**	45.85%	2.57%	50 seconds
	Random Forest	52.13%	3.27%	39 seconds
	Gradient Boosting	58.10%	3.35%	1h 23mins
	Neural Network	52.02%	5.21%	1h 13mins
9 Clusters on variables	**Decision Tree**	65.73%	12.88%	50 seconds
	Random Forest	68.02%	12.70%	39 seconds
	Gradient Boosting	64.39%	15.08%	4h 10mins
	Neural Network	67.72%	10.54%	50mins
23 Clusters on variables and TFs	**Decision Tree**	66.95%	13.11%	51 seconds
	Random Forest	69.76%	13.64%	42 seconds
	Gradient Boosting	73.74%	8.71%	8h 47mins
	Neural Network	73.44%	8.51%	17mins

Table 6.1: Algorithms accuracies for Cluster-based segmentations

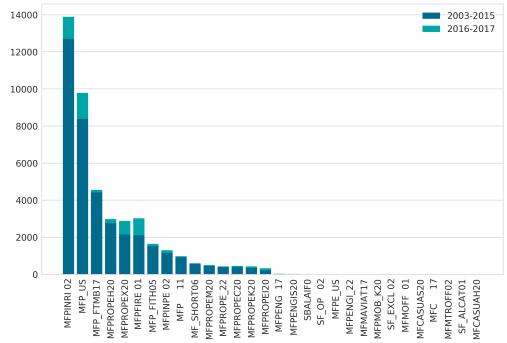
All the algorithms give somewhat close accuracies, a Gradient Boosting algorithm was selected for "4 Clusters on 4 TFs" and "23 Clusters on variables and TFs", with a learning rate of 0.05 and 100 estimators. A Random Forest was selected for "9 Clusters on variables", with 500 estimators, with $\frac{\sqrt{numberofvariables}}{numberofvariables}$ as the proportion of variables to use and no limit regarding the depth of the tree.

It is worth noting that, while the accuracies for the clusters on variables are rather correct, the ones for the clusters on TFs only are not great at all, indeed, as one class contains 55.75% of the database, one model classifying all the claims in that class would, on average, outperform the classification based on Decision Trees, Random Forests and Neural Networks.

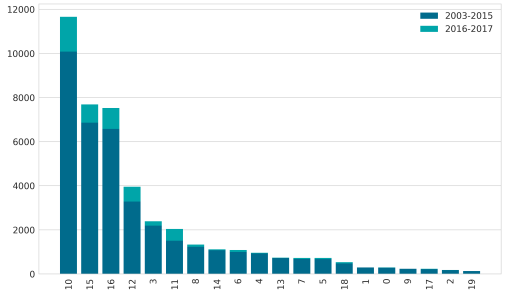
Moreover, the accuracies on the clusters based on variables only are worse than the accuracies based on both the indicators and variables, which is unexpected since the indicators values or not considered in the classification process and as the number of classes is higher for the segmentation based on both the variables and the indicators. This can be due to the clusters on variables being not distinct enough due to the small number of classes.

6.1.2 Distribution of the recent claims in the new segmentations

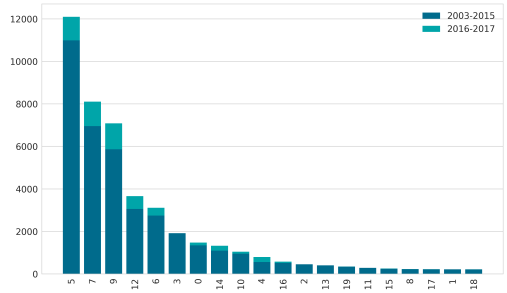
Hereunder are the distributions of the newly classified claims (2016-2017), as well as the claims on which the classes were built (2003-2015).



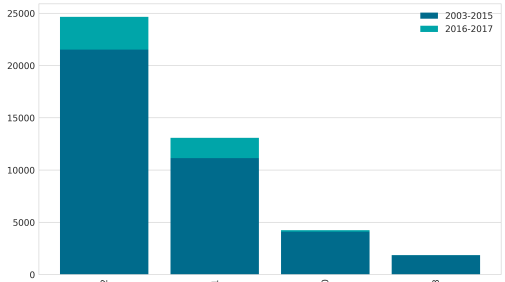
Distribution of the classes
of the Actuarial Segmentation



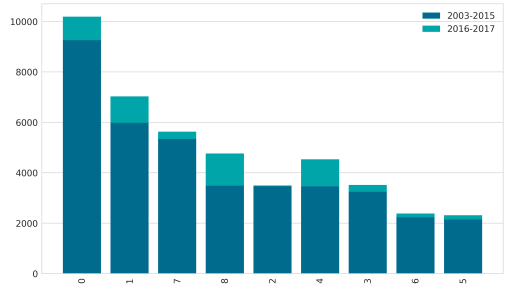
Distribution of the predicted
classes of DT 20 100 Inc



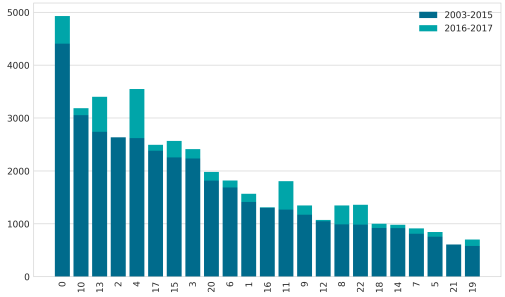
Distribution of the predicted
classes of DT 20 150 Paid



Distribution of the predicted
classes of 4 Clusters on 4 TFs



Distribution of the predicted
classes of 9 Clusters on variables



Distribution of the predicted
classes of 23 Clusters on variables and TFs

6.1.3 Comparing the results

The claims for 2016/2017 having been reclassified in the classes for each segmentation, the cash-flow for the year 2019 can be estimated for underwriting years ranging from 2003 to 2017.
For the following tables:

- $Delta\ \% = \frac{Amount_{Predicted} - Amount_{Actual}}{Amount_{Actual}}$
  Measures how close to the real value the total prediction is, but errors can compensate;
- $|Error|/|CF| = \frac{\sum_{j \in UWY} |CF^j_{Predicted} - CF^j_{Actual}|}{\sum_{j \in UWY} |CF^j_{Actual}|}$ With $CF^j_{\cdot}$ amount for the j-th underwriting year.
  Measures the sum of errors for each underwriting year, divided by the sum of the absolute values of each actual cash-flow.

Hereunder is the result table for the Incurred cash-flow of 2018:

Incurred 2018 cash-flow

	Amount	**Delta %**	**|Error| / |CF|**
Actual	368,875,264	//	//
One Segment	305,627,643	-17.15%	19.72%
Actuarial Segment	381,993,796	3.56%	29.29%
DT 20 100 Inc	316,914,859	-14.09%	17.65%
DT 20 150 Paid	335,585,277	-9.02%	19.64%
4 Clusters on 4 TFs	277,386,795	-24.80%	27.60%
9 Clusters on variables	328,666,982	-10.90%	16.67%
23 Clusters on Var & TFs	357,989,774	-2.95%	23.67%

Table 6.2: Incurred 2018 cash-flow predictions

Hereunder is the result table for the Paid cash-flow of 2018:

Paid 2018 cash-flow

	Amount	**Delta %**	**|Error| / |CF|**
Actual	317,757,021	//	//
One Segment	360,864,091	13.57%	53.45%
Actuarial Segment	446,393,658	40.48%	75.67%
DT 20 100 Inc	316,914,860	-0.27%	126.76%
DT 20 150 Paid	394,575,500	24.18%	62.92%
4 Clusters on 4 TFs	378,169,193	19.01%	67.73%
9 Clusters on variables	392,870,084	23.64%	60.43%
23 Clusters on Var & TFs	445,240,269	40.12%	78.28%

Table 6.3: Paid 2018 cash-flow predictions

The two tables of results are very sensitive to the classification of the new claims. Indeed, as the Fire Database is composed of very short tail claims, most of the 2018 cash-flow amount is due to the claims with an underwriting year of 2016 or 2017. However, as we have seen earlier, this classification is quite volatile for the segmentations based on clusters.

In general, the Incurred estimation is more accurate than the Paid estimation. The Actuarial Segmentation is also outperformed on 3 out of 4 measures of errors by the segmentation having only one class. This can be due to the homogeneity of the Lob=Fire database.

This first measure of quality of prediction is quite volatile because based on the prediction of an unique cash-flow.

6.2 Predicting the IBNR / Reserves amounts

In the last section, the prediction of the cash-flow of 2018 was done on the whole range of underwriting years (2003-2015) and compared to the actual cash-flow of 2018. However, when estimating IBNR or Reserves, a selection regarding the years of development and the underwriting years needs to be made.

Knowing that:

$$IBNR = \sum_{j \in UWY} (Ultimate_j - Actual_j)$$

We need to know the "real" values for $Ultimate_j$, $\forall j \in UWY$ in order to compare it with the predicted values. However, the only claims for which the ultimate is known are the closed claims: claims for which it is considered that no payment will be made.

Therefore, for the study of the IBNR / Reserves, our database must be only composed of closed claims, hence the dropping of the open claims for this study[2].

[2]The closed claims represented 35628 claims, 92.29% of the Fire sub-database.

6.2.1 Predicting the IBNR / Reserves not knowing the last 3 years of development

In this section, the IBNR / Reserves will be estimated for closed claims. The 3 most recent years will be deleted: the development for 2018, 2017 and 2016. From the remaining development, a pattern will be fitted for each class, and the Ultimate amount will be estimated.

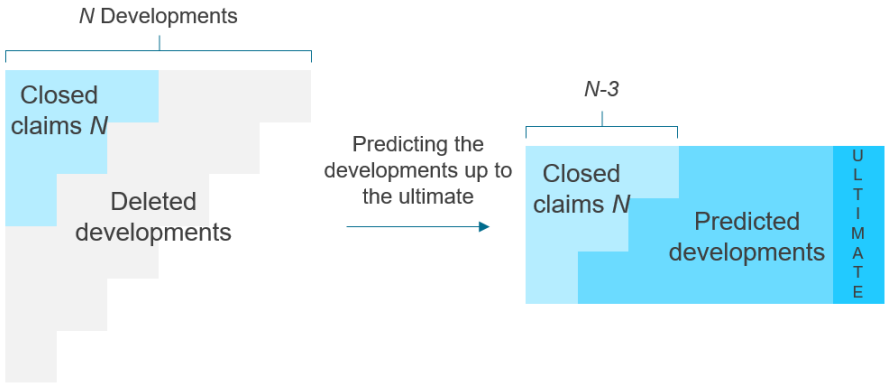


Figure 6.2: Process used to compare the quality of IBNR prediction

The ultimate amount being estimated, it is now possible to compare the estimation of IBNR / Reserves for each segmentation and compare it with the actual value:

$$IBNR_{N-3}^{Predicted} = Ultimate^{Predicted} - Actual_{N-3}$$

Knowing that the claims are closed for the year N, we have that $Actual_N$ is equal to $Ultimate$, therefore the actual IBNR for the year $N-3$ is as follows:

$$IBNR_{N-3}^{Actual} = Ultimate^{Actual} - Actual_{N-3} = Actual_N - Actual_{N-3}$$

The deletion of the last 3 years of development also reduced the number of underwriting years:

The underwriting years were initially ranged from 2003 up to 2017, now that 3 years of development have been deleted, the last 3 underwriting years have to be taken out: 2018, 2017 and 2016. Moreover, the deletion of the last 3 underwriting years means that no developments were available to calculate the development factors for the three oldest underwriting years: 2003, 2004 and 2005.

The comparison for the IBNR / Reserves amounts not knowing the 3 last years of development will therefore be made only considering the 9 years ranging from 2006 up to 2015.

Hereunder are the results of the IBNR estimations:

	IBNR predictions		
	Amount	**Delta %**	**|Error| / |CF|**
Actual value	25,306,093	//	//
One Segment	777,716	-96.63%	37.62%
Actuarial Segment	-33,272,204	-231.48%	95.50%
DT 20 100 Inc	7,829,356	-69.06%	37.99%
DT 20 150 Paid	33,719,195	33.25%	47.41%
4 Clusters on 4 TFs	28,350,728	12.03%	59.69%
9 Cluster on variables	12,796,456	-49.43%	38.50%
23 Clusters on Var & TFs	-7,298,270	-128.84%	56.40%

Table 6.4: IBNR prediction with 3 development years taken out

Hereunder are the results of the Reserves estimations:

	Reserves prediction		
	Amount	**Delta %**	**|Error| / |CF|**
Actual value	256,596,807	//	//
One Segment	331,198,458	29.07%	58.36%
Actuarial Segment	406,321,433	58.35%	69.11%
DT 20 100 Inc	333,818,542	30.09%	56.54%
DT 20 150 Paid	373,416,394	45.53%	75.79%
4 Clusters on 4 TFs	305,804,145	19.18%	55.78%
9 Cluster on variables	374,530,195	45.96%	75.94%
23 Clusters on Var & TFs	498,003,307	94.08%	101.26%

Table 6.5: Reserves prediction with 3 development years taken out

The Actuarial Segmentation, the segmentation that has only one class and the segmentations based on variables and TFs give overall bad results. However, the other segmentations give at least somewhat realistic IBNR amounts. Regarding the Reserves, the results are more stable.

It is worth noting that for both the IBNR and the Reserves, the Actuarial Segmentation is outperformed by the segmentation that only has one class.

6.2.2 Predicting the IBNR / Reserves not knowing the last 5 years of development

The process is the same as before, the only difference is that instead of taking 3 years of development out, 5 were taken out.

Hereunder are the results for the IBNR estimations:

	IBNR		
	Amount	**Delta %**	**|Error| / |CF|**
Actual value	45,218,528	//	//
One Segment	182,685,548	304.01%	91.26%
Actuarial Segment	177,245,190[3]	291.97%	86.05%
DT 20 100 Inc	186,178,519	311.73%	91.87%
DT 20 150 Paid	209,114,260	362.45%	106.82%
4 Clusters on 4 TFs	259,320,314	473.48%	139.54%
9 Cluster on variables	184,309,725	307.60%	91.54%
23 Clusters on Var & TFs	249,857,348	452.56%	133.98%

Table 6.6: Results of the IBNR prediction with 5 development years taken out

Hereunder are the results for the Reserves estimations:

	Reserves		
	Amount	**Delta %**	**|Error| / |CF|**
Actual value	452,195,462	//	//
One Segment	356,575,131	-21.15%	28.54%
Actuarial Segment	578,188,041	27.86%	47.44%
DT 20 100 Inc	366,375,145	-18.98%	32.34%
DT 20 150 Paid	390,978,738	-13.54%	23.64%
4 Clusters on 4 TFs	366,420,150	-18.97%%	32.84%
9 Cluster on variables	532,977,480	17.86%	19.22%
23 Clusters on Var & TFs	455,569,693	0.72%	18.17%

Table 6.7: Results of the Reserves prediction with 5 development years taken out

The prediction for the IBNR with 5 years taken out is overestimated for all segmentation. Indeed, the IBNR amount is volatile, and close to 45,000 while all the segmentations predict an IBNR amount around 185,000,000. The results on the Reserves are more coherent.

Taking out 2 more years in the study of the prediction of IBNR / Reserves means that the measure of error is done only focusing on the 5 underwriting years ranging from 2008 up to 2013. The small number of underwriting years studied makes the results even more volatile.

[3] A correction of 7.9 billion was made due to a small class having an abnormal development factor

6.3 Selection of the best segmentation

For all the indicators, the 5 segmentations were sorted from best to worst based on their predictabilities. Each got attributed a number corresponding to their rank, the sums of the ranks for all indicators were done to study which segmentation was the best overall.

			Segmentation				
Criteria	Cash-flow	Measure of error	DT 20 100 Inc	DT 20 150 Paid	4 Clusters on 4 TFs	9 Clusters on variables	23 Clusters on variables + TFs
Predicting 2018 cash-flow	Incurred	Delta %	4	2	5	3	1
		|Error|/|CF|	2	3	5	1	4
	Paid	Delta %	1	4	2	3	5
		|Error|/|CF|	5	2	3	1	4
IBNR/Reserves estimation (3 years taken out)	IBNR	Delta %	4	2	1	3	5
		|Error|/|CF|	1	3	5	2	4
	Reserves	Delta %	2	3	1	4	5
		|Error|/|CF|	2	3	5	1	4
IBNR/Reserves estimation (5 years taken out)	IBNR	Delta %	2	3	5	1	4
		|Error|/|CF|	4	2	5	3	1
	Reserves	Delta %	4	2	5	3	1
		|Error|/|CF|	4	3	5	2	1
Total			35	32	47	27	39

Table 6.8: Ranks of the segmentations

Considering all the measures of error for Incurred and Paid, the best segmentation is the one with 9 classes based on the clusters on variables (it is also the best considering only the Incurred or Paid related predictions separately).

6.3.1 Presentation of the best segmentation

The best segmentation is the one with 9 classes based on clusters of variables. Hereunder is the distribution of the classes for this segmentation, with the claims of the most recent contracts (underwritten in 2016 & 2017) in a distinctive colour:

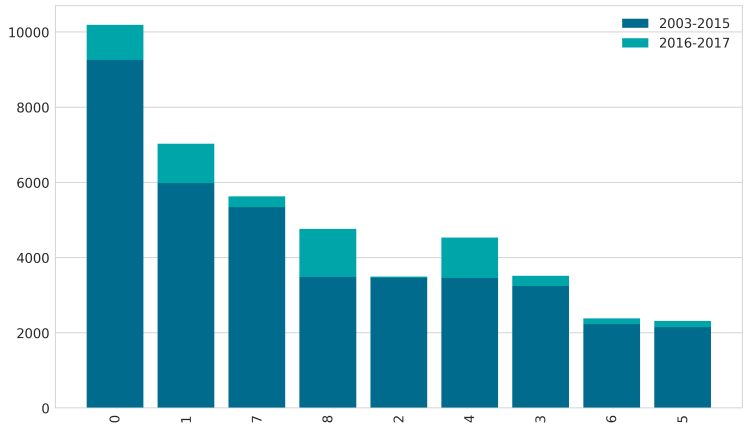


Figure 6.3: Distribution of the classes of 9 Clusters on variables

The two previous chapters focused on the homogeneity of classes regarding indicators of the development patterns and quality of prediction by studying the estimation of 2018 cash-flow, IBNR and Reserves.

However, beyond these statistical criteria, it is necessary to study the very composition of the classes to see whether they are coherent enough to be used by an insurance/reinsurance company and if they comply with regulations.

6.3.2 Comparing the development patterns

The development pattern of each class will be displayed for Incurred and Paid, to see whether the segmentation has created homogeneous and distinct groups regarding the development patterns (which should be the case since this segmentation is the one that has the best predictability).

Hereunder are the patterns of development of Incurred for each segmentation:

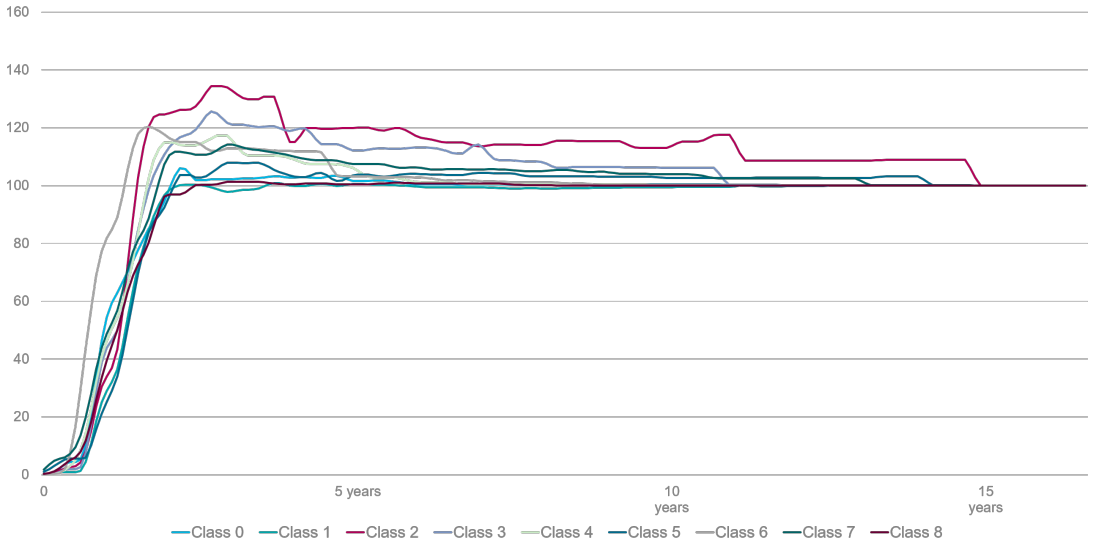


Figure 6.4: Incurred developments for 9 clusters on variables

The classes 2, 3, 4, 5 and 6, have a lot of positive developments on Incurred. On the other hand, 0, 1 and 8 have very little to no positive developments and have very similar patterns.

Hereunder are the patterns of development of Paid for each segmentation:

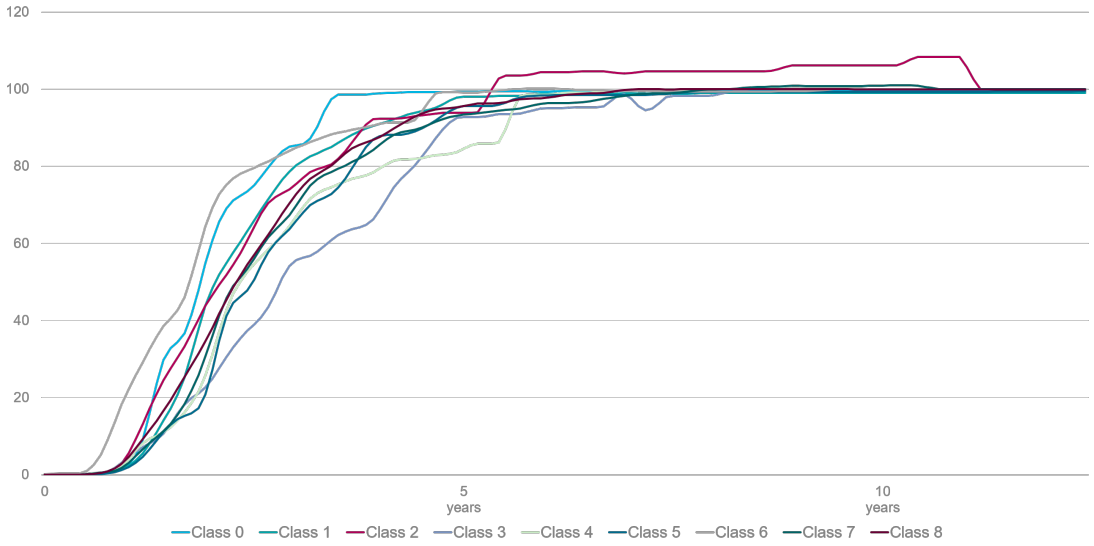


Figure 6.5: Paid developments for 9 clusters on variables

The classes 0 and 6 are the shortest tailed for Paid. While 3 and 4 are rather long-tailed. The class 2 even has positive developments, due to one claim[4] having a late positive development for a high amount, shifting the whole pattern.

Overall, even if some classes have distinct patterns, they remain close to each other. This was expected as the Fire database is homogeneous and short tail in general.

Hereunder are the undiscounted durations, in years, obtained from the patterns[5]:

	Class 0	Class 1	Class 2	Class 3	Class 4	Class 5	Class 6	Class 7	Class 8
Duration of Incurred	1.02	1.34	-0.99	0.05	0.69	0.89	0.25	0.39	1.18
Duration of Paid	2.03	2.34	2.25	3.19	2.89	2.76	1.95	2.72	2.56

Table 6.9: Undiscounted durations for 9 clusters on variables

[4] A claim for a mining company in Chile was firstly notified as a machinery breakdown, after a court decision, a faulty design was said to be the cause of the claim, reducing SCOR's liability by more than 2M€.

[5] These durations are calculated from the patterns of development, considering therefore the Incurred/Paid amount as a weight (for our indicators, TF_Incurred and TF_Paid, each row has the same weight.)

6.3.3 Risk profile of the classes

In this section, the risk profile of each class will be highlighted. This analysis was done regarding the previous studies on the numerical variables and the patterns, as well as a study on the categorical variables.

For the analysis of the categorical variables, a focus was made on the scope of business and the type of policy, as well as the country of the claim. Specific comments regarding other variables were made if necessary. A display of the composition of the classes regarding these three variables (Sob, Top, Country_Claim) is available in the appendix C.

Class 0 (23.24% of the database[6])

The claims of the class 0 are divided between: residential (all the residential business is contained in the class 0 (four times the average)), commercial, industrial (non-petrochemical), and infrastructure covers. Half of its policies are related to physical damage on named perils (highest proportion among all the classes, more than three times the average[7]), the second biggest kind of policy is physical damage and business interruption, also on named perils (two times the average). The claims of this class are related to small risks on which SCOR has a big share. This class contains all the activity related to the Italy entity (2.3%).

The claims are mainly from the south of Europe: 78.63% of Spain, 13.39% of Portugal, 2.59% of Italy and 1.58% from Andorra.

The underlying risk is especially short tail, with the second lowest duration for Paid, and shows no positive developments on Incurred.

Class 1 (13.14% of the database)

The business of the class 1 is in majority composed of industrial (non-petrochemical) (two times the average), 20% of the claims are related to food production (four times the average). Its policies are mainly related to "All risk" covers for physical damage and business interruption.

The claims are mainly from South America: 24.23% from Columbia, 17.69% from Ecuador, 13.10% from Mexico, 7% from Chile.

The class 1 has the longest duration for Incurred, since it has no positive developments, and a median duration for Paid.

Class 2 (7.96% of the database)

The business of the class 2 is composed at 41% of commercial covers (two times the average) and a third of Industrial (non-petrochemical) covers. Its policies are composed of 88% of physical damage and business interruptions on "All risk" (1.5 times the average).

[6] underwriting years ranging from 2003 and 2017, with a selection considering the TF extreme values.
[7] Compared to the proportion observed among all the classes.

The claims are mainly from the Americas and Israel: 21.63% from Chile, 10.92% from Israel, 8.60% from Canada and 6.50% from Mexico.

The class 2 has a negative duration for incurred, due to one claim[8] having a positive development eleven years later. Without this claim, the pattern would be located in the average of the other patterns.

Class 3 (7.99% of the database)

The class 3 has the highest share of financial institution covers (eight times the average) and the second highest share of infrastructure covers. Half of its policies are for physical damage covers, it also has the third highest share of covers for named perils. 23% of the claims cover risks related to real estate (10 times the average), 10% cover risks relate to telecommunications (3 times the average).

The claims are from South East Asia: 57.37% from Honk-Kong, 13.82% from China, 5.74% from Malaysia, 3.74% from Taiwan and 3.68% from South-Korea.

The class 3 is the shortest tail for Incurred, due to positive developments but the second most long-tail for Paid. Which seems to indicate that the risk is quickly estimated, but then re-evaluated downward. It also indicates that SCOR has a prudent approach in the management of these claims.

Class 4 (10.37% of the database)

The business of the class 4 is very similar to the business of the class 1, with larger risks: 57.90% is composed of industrial covers (non-petrochemical) (two times the average), a quarter of infrastructure covers, the rest is evenly distributed. It contains 10% of risks related to metallurgy (four times the average).

The claims are from South Asia (except South-Korea): 26.23% from Thailand, 15.78% from Indonesia, 9.21% from South-Korea, 8.02% from India, 4.96% from China.

The class 4 is the second most long-tail for Paid with a duration of 2.89 years but is rather short-tail for Incurred, with a duration of 0.69 years.

Class 5 (5.19% of the database)

Class 5 has the highest share of Commercial covers. Its policies are mainly related to physical damage and business interruption. 27% (12 times the average) of the contracts cover retailers and 19% cover governments (10 times the average). A fifth of the claims are related to theft acts.

The claims are mainly from South America: 28.19% from Brazil, 25.90% from Mexico, 19.79% from Columbia, 5.01% from Ecuador, 4.83% Venezuela.

[8] A claim for a mining company in Chile was firstly notified as a machinery breakdown, after a court decision, a faulty design was said to be the cause of the claim, reducing SCOR's liability by more than 2M€.

Class 6 (5.44% of the database)

The class 6 has the second highest share of commercial business and Financial institution business. Half of its policies are physical damage related, 30% is related to named perils.

The claims are mainly from Europe (and the United Arab Emirates): 41.43% from Germany, 13.15% from the United Arab Emirates, 8.17% from Italia, 8% from Austria, 6.11% from Malta.

The class 6 is very short tail with a duration of 0.25 years for Incurred and 1.95 years for Paid (smallest duration for Paid among all classes).

Class 7 (12.91% of the database)

The class 7 has the highest share of infrastructure and the second highest share of industrial covers (petrochemical). All its policies are for both Physical damage and business interruption: two thirds for all risks and a third for named perils. 16% of the claims are related to water treatment (8 times the average).

The claims are mainly from the south of Europe (although more diverse than the other classes): 25.95% from Portugal, 17.17% from Spain, 6.89% from France, 4.45% from Israel, 3.69% from Saudi Arabia.

The incurred duration for the class 7 is low due to positive developments.

Class 8 (10.76% of the database)

The class 8 is composed of large energy related contracts, it also has the highest median EGPI and by far the highest Incurred and Paid amounts.

Half of the class 8 business is related to petrochemical industry covers (almost all of petrochemical contracts are in the class 8), a third of it is for infrastructure. 90% of its policies are for physical damage and business interruption on all risks. 15% of the claims are related to power generation (excluding nuclear, thermal, hydro and renewable). The class 8 contains the highest rate of XL covers: 23%, 6 times the average rate.

The claims are mainly from North America: 33.44% from the USA, 6.27% from Canada, 5% from Germany, 4.69% from Austria and 3.96% from France.

The duration for Incurred is the second highest, indeed the class 8 has no positive developments.

6.3.4 Conclusion on the best segmentation

The Fire line of business being very homogeneous by definition, finding a way to build more homogeneous subgroups was a challenge.

The study of the development pattern of each class showed that the segmentation managed to distinguish different kinds of patterns. Moreover, the newly built classes outperform the Actuarial Segmentation, regarding the prediction quality, measured by comparing the prediction (obtained by the natural developments) and the actual values of 2018 cash-flows and IBNR / Reserves.

However, the errors studied are volatile, and a few big claims can distort an otherwise homogeneous class. Indeed, since there is no correction regarding the development factors, claims that have abnormal behaviours will corrupt the estimation for all the claims of its class. For example, the claim which has a late positive development in the class 2 makes all the claims of the class 2 have a late positive development, which would not have occurred if this claim was deleted or if a selection of development factors was made.

A study of the composition of each class showed that risk-profiles could be assigned per class. Therefore, the segmentation put forward is coherent and interpretable.

Overall, the selected segmentation separates the different types of patterns, as it is the best segmentation regarding the prediction errors, while clearly defining new sets of claims with different characteristics, as detailed in the risk profile of each class.

The results on the Fire database are quite volatile, due to the restricted numbers of claims per class. But in general, the Actuarial Segmentation is still outperformed in terms of homogeneity and predictability.

6.4 Conclusion on the segmentation challenge

The objective of this project is to study the relevance of the Actuarial Segmentation, and especially the homogeneity of the classes, regarding their development patterns.

To do so, new segmentations have been built. The construction of these segmentations follows a framework composed of different steps defined in this paper:

- Building of indicators to describe the patterns of development;
- Selecting the segmentations around the homogeneity of these indicators;
- Comparing the values predicted by the natural developments of each class, in order to quantify the predictability of the segmentation.

The results obtained by using this framework are coherent: the precisions of prediction are better than the ones obtained for SCOR's current segmentation, as the classes built are composed of more homogeneous claims.

Moreover, by considering a classification process for the recent underwriting years, the segmentations are not only better in a retrospective view but are also coherent when used for new underwriting years. That aspect is reflected by the different risk profiles that can be interpreted and linked to the newly built classes.

The classes built on trees and those built on clusters give good results of the same order of magnitude, despite being constructed using very different methods. Indeed, for tree-based segmentations, the classes are defined by rules that describe the interaction between the variables and the underlying risks (estimated by indicators). However, clusters are solely built on similarities between the claims, no causal links between the variables and the indicators are estimated.

Therefore, the fact that the best segmentation is the one based on clusters on variables tells us that:

- Groups with different behaviours are clearly present in the dataset. As the study of the classes have shown, some groups have specific characteristic that make them stand out.
- The variables chosen describe the underlying risk (or at least a majority of it). Indeed, if variables that are irrelevant to the underlying risk were considered in the building of the classes, then the classes would be homogeneous regarding those irrelevant variables but not regarding the risk.
- The indicators might not describe the pattern as much as expected, as the best segmentation is the one not taking into account the indicators. This can be due to the high number for which the indicators are equal to zero, but also to the fact that most of the claims had similar values, due to the homogeneity of the Fire database.

The homogeneity of the Fire database limited the improvement brought by the Machine Learning methods. Now that the method is established for the Fire database, it will be generalized to the whole database, and the improvements will be compared.

Chapter 7

Generalization on the whole database

The process of challenging the Actuarial Segmentation now established for the Fire database, it will be extended to the database containing all lines of business. The whole database contains different lines of business, and more diverse risks. Hereunder is the distribution of the lines of business:

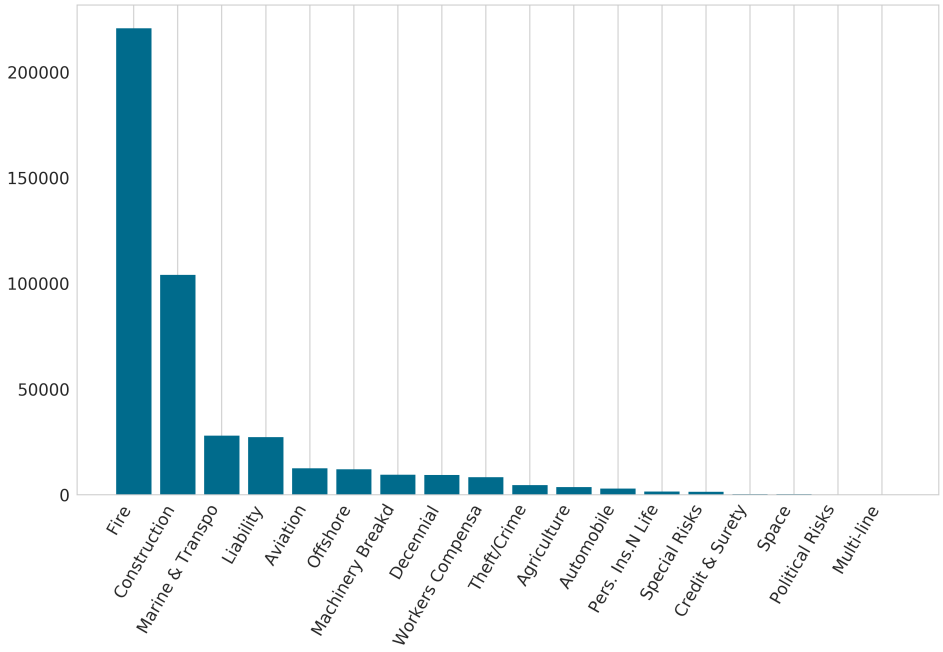


Figure 7.1: Distribution of the lines of business

The whole database is more heterogeneous than the Fire database. That effect can be seen in the means of indicators per line of business[1], displayed hereunder:

	Agriculture	Automobile	Aviation	Construction	Credit & Surety	Decennial	Fire	Liability	Machinery Breakdown
TF_Incurred	1.96	1.78	2.14	2.45	2.10	5.21	1.22	1.91	1.51
TF_Paid	2.02	2.30	3.51	3.56	2.87	7.21	2.12	2.98	2.37
Size %	0.80%	0.65%	2.79%	23.31%	0.08%	2.09%	49.44%	6.12%	2.13%

	Marine & Transport	Multi-line	Offshore	Pers. Ins.N Life	Political Risks	Space	Special Risks	Theft Crime	Workers Compensation
TF_Incurred	2.70	1.48	1.36	1.74	5.07	2.13	2.43	2.57	2.19
TF_Paid	3.02	2.58	2.24	2.00	2.21	3.13	2.12	2.40	2.64
Size %	6.26%	0.00%	2.70%	0.34%	0.03%	0.07%	0.31%	1.02%	1.85%

Table 7.1: Means of indicators per line of business

For that reason, it was more likely to have a greater improvement, since it would be easier for the algorithm to find classes with distinctive behaviours. Moreover, the segmentation will have more data to estimate the factors of development, making the prediction process sturdier.

The same process used for finding the best segmentations on the Fire database was applied to the whole database: the segmentations were built on a train database and the errors were compared on the test database.

Some technical limitations occurred when generalizing the process to the whole database:

- The python algorithm used in this study for computing the k-prototype is still in development and takes a very long time to compute on a big dataset. In order to reduce the computation time when measuring the MSE for the cluster, the number of iterations used for the clustering algorithm was lowered from 10 (for the Fire database) to 3, inducing more volatility in the MSE estimation. Even with only 3 iterations, it would often take more than 10 hours for a single set of parameters to be computed;

- The size of the database and the format of the import in ResQ required the merging of big datasets, and the limits of memory of the servers were sometimes reached. Despite the server being very powerful with 120 GB of ram and 12 CPUs, although shared between multiple users. Moreover, since the ResQ import files are very heavy (more than 6GB per type of cashflow per type of segmentation), a manual import in ResQ would take nearly half a day, and thus for each type of cash-flow and segmentation.

[1] Considering the values between -20 and 30 for the underwriting years ranging from 1950 to 2010.

From the study of these errors, one segmentation per model was selected. The parameters for the selected segmentation are as such:

- Segmentation on tree fitted to Incurred: 15 classes/1000 claims min. per class;
- Segmentation on tree fitted to Paid: 15 classes/1000 claims min.;
- Segmentation on clusters on indicators: 6 clusters;
- Segmentation on clusters on variables: 24 clusters;
- Segmentation on clusters on the variables and the indicators: 21 clusters.

The most recent claims were assigned to the existing classes. Then, all the segmentations were tested on their ability to predict the cash-flow of the most recent year for Incurred/Paid and the IBNR/Reserves prediction. The accuracies for the segmentations based on clusters are better on the whole database (due to more data available for the algorithm to learn).

7.1 Prediction results on the whole database

Some corrections were needed for the whole database:

- The years 1950-1973 contained volatile late developments that impacted the ultimate predictions, they were therefore taken out (representing 505 claims);
- A huge claim related to the World Trade Center attacks gave the underwriting year 2001 a bigger weight and shifted the whole development pattern, that claim was therefore taken out;

Hereunder are the results of predictions for the selected segmentations:

			Segmentations						
Criteria	**CF**	**Measures of error**	**Act. Seg.**	**One Seg.**	**DT 15 Inc**	**DT 15 Paid**	**6 clus. on TF**	**24 clus. variables**	**21 clus. variables & TF**
Predicting 2018 cah-flow	Inc.	Delta %	-6.4%	1.5%	-2.3%	-4.4%	10.7%	-5.5%	12.0%
		|Error|/|CF|	27.5%	22.7%	35.4%	27.0%	35.7%	28.3%	42.7%
	Paid	Delta %	13.8%	-6.5%	-15.9%	-9.6%	0.1%	7.2%	3.4%
		|Error|/|CF|	37.3%	25.6%	24.1%	25.5%	46.5%	34.4%	36.8%
IBNR Reserves estimation N-3	IBNR	Delta %	126.4%	155.9%	-17.6%	78.0%	83.0%	25.9%	2.6%
		|Error|/|CF|	277.2%	150.6%	66.2%	102.8%	139.7%	105.8%	100.5%
	Res.	Delta %	178.3%	90.5%	40.1%	74.5%	35.3%	34.4%	42.1%
		|Error|/|CF|	200.4%	121.2%	78.9%	103.3%	59.4%	67.0%	78.5%
IBNR/Reserves estimation N-5	IBNR	Delta %	122.4%	260.0%	-108.8%	170.8%	162.2%	60.8%	89.4%
		|Error|/|CF|	234.9%	244.5%	152.7%	169.5%	197.8%	99.8%	115.1%
	Res.	Delta %	58.0%	46.7%	12.3%	39.1%	30.3%	12.9%	5.2%
		|Error|/|CF|	101.3%	65.7%	52.1%	60.8%	52.7%	45.8%	45.7%

Table 7.2: Results of prediction for the whole database

The results on the whole database are overall better than the ones of the Fire database. The average error among all models on the cash-flow prediction went from 11.78% (for Fire) to 6.11% (for all lines of business) for Incurred an from 23.04% (for Fire) to 8.07% (for all lines of business) for Paid.

Moreover, the size of the dataset (despite making every step of the process much slower) made the estimation of the development factors sturdier. Indeed, the results on the whole database are less volatile than on the Fire database.

The Actuarial Segmentation is more precise, but is still outperformed by some challenging segmentations. The best segmentation overall is the one based on trees fitted to Incurred, containing 15 classes. On the next page is the tree from which the classes are obtained.

Hereunder are the development patterns of Incurred for each class of the segmentation based on tree fitted to Incurred:

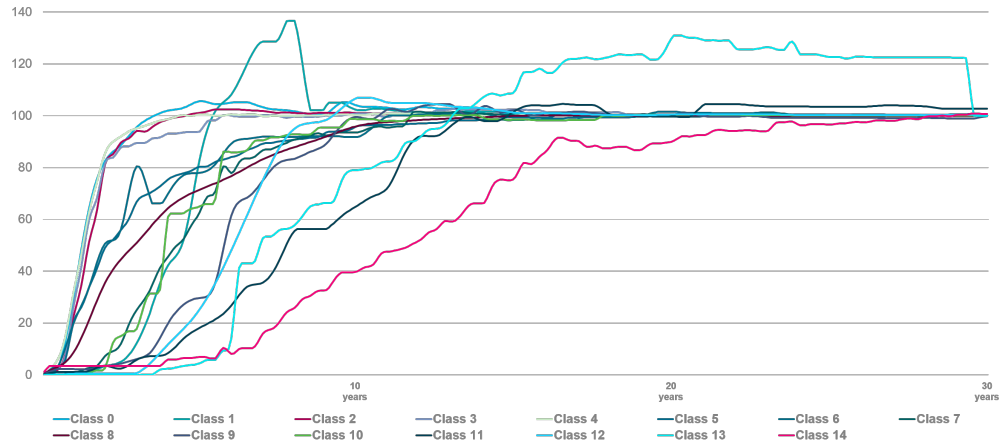


Figure 7.2: Incurred developments for segmentation based on tree fitted to Incurred

Hereunder are the development patterns of Paid for each class of the segmentation based on tree fitted to Incurred:

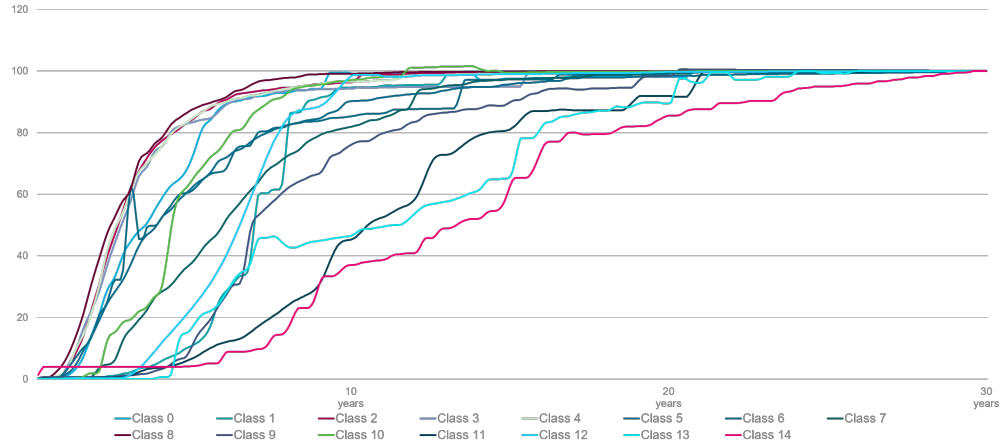


Figure 7.3: Paid developments for segmentation based on tree fitted to Incurred

Both the developments of Incurred and Paid are more spread out than for the Fire database. The positive development of Incurred for the class 1 is due to a large claim related to an off-shore petrol platform, and the one of the class 13 is due to a large claim related to a decennial contract.

Hereunder is the decision tree from which the classes are obtained:

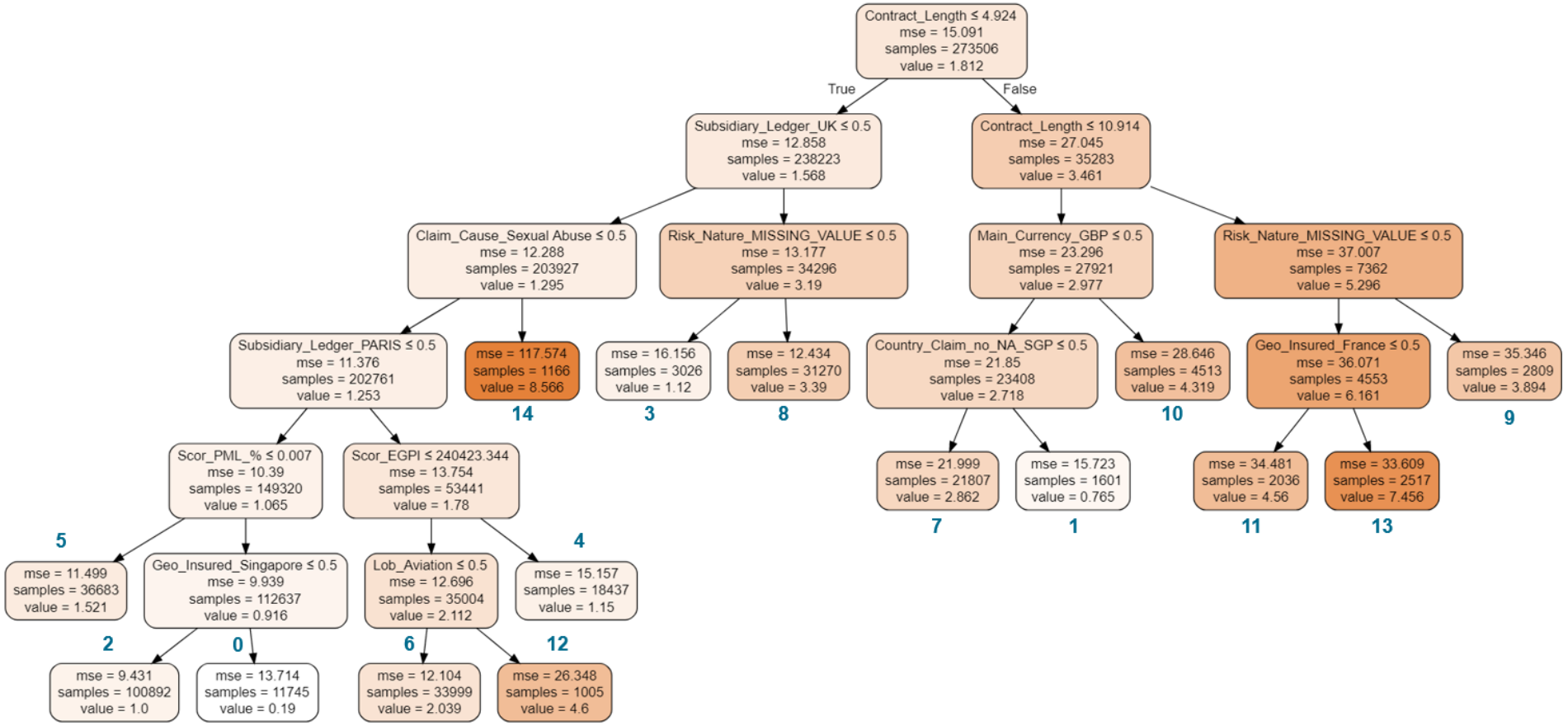


Figure 7.4: Decicion tree fitted to Incurred with: 15 classes / 1000 claims per class min.

The splits are done considering a value for the numerical variables, the claims that are below that value will go to the left and the others on the right. Regarding the categorical variables, as they are encoded in One-Hot, they are either equal to 1 or 0 whether the claim had that category for that variable or not.

For example $Main\_Currency\_GBP \leq 0.5$ will split on the left (True) the claims that are not in GBP and on the right (False) the claims that are in GBP

For the tree: mse refers the mean square error of the node, samples to the size of the node and value to the mean value of the node.

Some classes are easily interpretable, such as the long-tail claims related to sexual abuse, or the claims of the line of business Decennial (that is essentially contained in the classes 9, 11 and 13). As well as more short tail classes such as the classes 0 and 1 in Singapore, or the class 2, which is composed at 70% of claims of the Fire Line of business, showing once again the homogeneity of the Fire subset.

Compared to the segmentation based on clusters, the rules of this segmentation are clearly defined, making it easier to use, as no classification step is required to allocate the new claims to the classes.

7.2 Conclusion on the whole database

The results of the generalization confirm our expectation. Indeed the predictions are better due to both the heterogeneity of the base and the increase in the number of claims.

The fact that the best segmentation was based on clusters for the Fire database and on trees for the whole database indicates that:

- The homogeneity of the Fire database reduces the improvements brought by splits on durations;
- The heterogeneity of the complete database enables the dividing of sub-groups based on their duration values.

Dividing this study in two steps (first the study on the Fire database and then the generalization on the whole database) made the research process much quicker. Indeed, adjusting our models and doing the explanatory process directly on the whole database would have taken too much computation time and calibrations.

Furthermore, testing the process on both a homogeneous dataset and a heterogeneous dataset gave us confirmation on the reliability of the process.

This study also highlighted the fact that aggregating all the claims is, in terms of predictions, better than creating inhomogeneous classes. Especially if the claims are already homogeneous, which is the case for the Fire database.

Conclusion

This thesis revolved on the challenging of SCOR's current segmentation of claims. To do so, new segmentations were built using Machine Learning on a database containing both contracts and claims features.

To enhance the quality of the database and to enable the use of Machine Learning algorithms, the missing values of the study database were filled using methods based on the correlations between the variables and dimension reductions. Due to technical limitations, this method has to be complemented with clusters to reduce the computation time and enabled the use of a more complex model, giving better results. A method for choosing the parameters of the algorithm also had to be implemented for big datasets. This method gave results that were proven to match with the method originally used (based on cross-validation) and for cheaper computational costs.

Regarding the building of the classes, indicators (undiscounted durations) were built to describe the development patterns. These patterns were proven to describe the development pattern, as the segmentations built around these indicators were able to differentiate different kinds of developments.

Challenging segmentations were built around the homogeneity of these indicators, using either decision tree or clustering methods. The challenging segmentations showed overall good results regarding the homogeneity of the claims and the quality of the prediction. The best segmentation was thoroughly studied and its composition made sense regarding claims management.

The framework in itself is interesting and can be generalized to create indicator-related segmentations in other domain. Alternatives to these methods were also tested but had limited results, such as clusters on the development ratios for certain periods or mixing clusters and decision trees (for example: using a tree to split a cluster in two). And more advanced methods could have also been used, such as finding the optimal segmentation by using genetic algorithms.

Machine Learning innovations often come with lack of interpretability and concrete application. This study, by focusing on simple algorithms used at different steps of the process, put forward a method that takes advantage of Machine Learning while remaining relevant to the current practices.

Bibliography

[1] Vincent Audigier, Francois Husson and Julie Josse, *MIMCA: Multiple imputation for categorical variables with multiple correspondence analysis.* arXiv:1505.08116 [stat.ME], (May 2015).

[2] Hanen Borchani, Gherardo Varando, Concha Bielza and Pedro Larranaga, *A survey on multi-output regression.* Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 216-233, (2015).

[3] Alexandre Boumezoued and Laurent Devineau, *Individual claims reserving: a survey.* (August 2018).

[4] Leo Breiman, *Random Forest.* Statistics Department, University of California (January 2001).

[5] Jerome H. Freidman, *Greedy Function Approximation: A Gradient Boosting Machine.* IMS 1999 Reitz Lecture (February 1999).

[6] Zhexue Huang, *Extensions to the k-Means Algorithm for Clustering Large Data Sets with Categorical Values.* Data Mining and Knowledge Discovery2, 283-304, (1998).

[7] Francois Husson and Julie Josse, *missMDA: A Package for Handling Missing Values in Multivariate Data Analysis.* Journal of Statistical Software, Volume 70, Issue 1, (April 2016).

[8] Salma Jamal, et al., *Machine Learning and Traditional Methods Synergy in Non-Life Reserving.* ASTIN working party, 2018.

[9] Julie Josse, *Distributed multilevel matrix completion for medical databases.* Chairs Day: Insurance, Actuarial Science, Data and Models, June 2018.

[10] Kevin Kuo, *DeepTriangle: A Deep Learning Approach to Loss Reserving.* Insurance: Mathematics and Economics, (May 2018).

[11] Len Llaguno, Manolis Bardis, Robert Chin, Tina Gwilliam, Julie Hagerstrand, Evan Petzoldt, *Reserving with Machine Learning: Applications for Loyalty Programs and Individual Insurance Claims.* Casualty Actuarial Society E-Forum, (2017).

[12] Gráinne McGuire, *Self-assembling insurance claim models using regularized regression and machine learning.* August 2018.

[13] Laura Elena Raileanu and Kilian Stoffel, *Theoretical comparison between the Gini Index and Information Gain criteria.* Annals of Mathematics and Artificial Intelligence 41: 77–93, (2004).

[14] Sebastian Raschka, Vahid Mirjalili. *Python Machine Learning.* Packtub, 2017.

[15] Ronald Richman, *AI in Actuarial Science.* Actuarial Society of South Africa's Convention, 2018.

[16] Franck Vermet, *Apprentissage statistique : une approche connexionniste.* Support de cours EURIA (M1), (2016).

[17] Mario V. Wüttrich *Neural Networks Applied to Chain-Ladder Reserving.* May 2017.

[18] Institut des actuaires *Recommandations à l'usage des étudiants, des filières académiques et des membres du jury en vue de la préparation de mémoire d'actuariat.* March 2019.

[19] *Directive 2009/138/EC of the European Parliament and of the Council.* 2009.

[20] *Official Journal of the European Union.* L12, Volume 58, 17 January 2015.

[21] *Directive 95/46/EC of the European Parliament and of the Council.* Official Journal of the European Communities, No L 281/31, 24 October 1995.

[22] Sklearn community, *2.3. Clustering.* Sklearn website, https://scikit-learn.org/stable/modules/clustering.html

Appendix A

Mathematical framework for Machine Learning algorithms

A.1 Artificial Neural Network

Let χ be a d-dimensional feature space, d being the number of variables used for predictions in the dataset and $x \in \chi$ the vector containing features values for an observation. The output of each neuron depends on three parameters:

1. The output of the previous layer (or the variables themselves for the first layer);
2. A vector of weights containing an intercept that will be applied to the output of the previous layer.
3. An activation function ϕ that translates the rate of "firing-up" for each cell:
   (a) Binary function that activates above a threshold (θ):

$$\phi(z) = \begin{cases} 1, & \text{if } z \geq \theta. \\ 0, & \text{if } z < \theta. \end{cases} \tag{A.1}$$

   (b) Linear function / Rectified Linear Unit(ReLU):

$$\phi(z) = z \qquad \text{or} \qquad Max(z, 0) \tag{A.2}$$

   (c) Logistic function:

$$\phi(z) = \frac{1}{1 + e^{-z}} \in [0, 1] \tag{A.3}$$

   (d) Hyperbolic tangent

$$\phi(z) = tanh(z) = \frac{sinh(z)}{cosh(z)} = \frac{e^{z} - e^{-z}}{e^{z} + e^{-z}} \in [-1, 1] \tag{A.4}$$

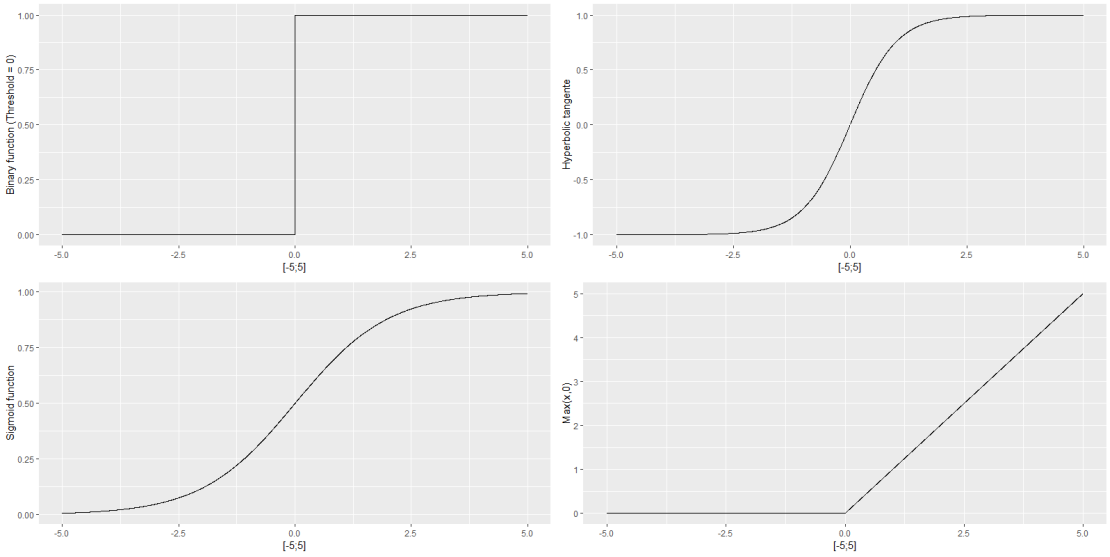


Figure A.1: Activation functions

The means of the activation functions have little to no importance since they can be adjusted by the intercepts introduced by the weights.

Mathematical notations

Let a Neural Network composed of H hidden layers, each hidden layer noted $h \in \{1, 2, ..., H\}$ will be composed of n_h neurons.
$\forall x \in \chi$:
The output for the first hidden layer ($h = 1$) is given by:

$$z_i^{(1)}(x) = \phi\left(W_{i,0}^{(1)} + \sum_{j=1}^{d} W_{i,j}^{(1)} \times x_j\right) \tag{A.5}$$

The output of the $i \in \{1, .., n_h\}$ neuron in the $h \in 2, ..., H$ hidden layer is given by:

$$z_i^{(h)}(x) = \phi\left(W_{i,0}^{(h)} + \sum_{j=1}^{n_{h-1}} W_{i,j}^{(h)} \times z_j^{(h-1)}(x)\right) \tag{A.6}$$

Eventually, the output of the Neural Network, noted Out, is given by:

$$Out(x) = \phi\left(\beta_0 + \sum_{j=1}^{n_H} \beta_j \times z_j^{(H)}(x)\right) \tag{A.7}$$

The output in this case is a function of both the features of $x \in \chi$ and of all the weights that composed the Neural Network.

The $z_j^{(h)}(x)$ notation can be generalized by declaring that $z_j^{(0)}(x) = x_j$ and the n_h notation by declaring that $n_0 = d$.

$$\forall h \times i \times j \in \{1 : H\} \times \{1 : n_h\} \times \{0 : n_{h-1}\}$$

$$Out(x) = Out_\alpha(x) \qquad \alpha = \left(\left(W_{i,j}^h \right)_{h=\{1:H\};i=\{1:n_h\};j=\{0:n_{h-1}\}} ; (\beta_j)_{j=\{0:n_H\}} \right) \tag{A.8}$$

In this example, the model was built to solve a regression problem, please note that in the case of a classification problem, the whole process is similar, it is just needed to add a final activation function right before the output layer to determine the most adequate class. This is usually done with a binary function (cf. EqA.1).

Algorithm optimization

To study the convergence and the efficiency of the Neural Network it is possible to compare the model's outputs with the actual data, a common way to do so is by computing a Loss Function. For example:

$$Loss(\alpha) = \frac{1}{2} \times \sum_{x \in \chi} \left(Out_\alpha(x) - Real\ data(x) \right)^2 \tag{A.9}$$

In order to optimize the Neural Network minimizing this Loss Function is needed. Since all the weights are continuous and if all the activation functions are continuous, a gradient descent algorithm can be used in order to find a minimum.

The gradient descent method is based on the Taylor development of order 1 (higher orders can be used; however, the derivatives can quickly become too complicate to calculate).

Let $Loss(.) \in L1$ function of α, using the Taylor for α near α and for $\nabla_\alpha Loss(\alpha)$ gradient of $Loss(\alpha)$:

Algorithm 1 Gradient Descent Algorithm

Require: $\eta > 0$: learning rate ; α^0 Initial value of weights; N number of iterations
  $\alpha = \alpha_0$
  $L(\alpha) = Loss(\alpha_0)$
  $\nabla = \nabla_\alpha Loss(\alpha_0)$
  for $n \in [1 : N]$ **do**
    $\alpha = \alpha - \nabla_\alpha Loss(\alpha) \times \eta$
    $L(\alpha) = Loss(\alpha)$
    $\nabla = \nabla_\alpha Loss(\alpha)$
  end for
  return $L(\alpha); \alpha$

A.2 Decision Tree

Let χ be a d-dimensional feature space, d being the number of variables used for predictions in the dataset and $x \in \chi$ the vector containing feature values for an observation.
Let C^k be the set of classes for the k-step in the tree.

$\forall k \in \{1, 2, ..., K\}$

$$C^k = \left(c_i^k\right)_{i=1}^{i=I^k}$$

With:

1. K being the depth of the tree;
2. c_i^k being the i-class for the k-step;
3. I^k being the number of nodes of the k-step;

Each c_i^k has an associated value v_i^k, which can be a class for classification or a number in regression.

Note: Decision trees are often presented with final nodes present in different layers. However, in this framework, when a node does not split, it will still be present in the next layer as itself just one layer deeper, in fact, all final nodes will be present in the final layer:
Although the model is exactly the same, this hypothesis lightens the notation as only the final layer needs to be referred to.

From this framework, a decision tree can be defined as a function:

$$\begin{aligned} T : x \in \chi &\to \left(v_i^K\right)_{i=1}^{I^K} \\ T(x) &= \textstyle\sum_{i=1}^{I^K} v_i^K \mathbb{1}\left(x \in c_i^K\right) \end{aligned} \tag{A.10}$$

Probabilities of belonging in each class can de defined by declaring that: $\forall i \times k \times x \in \{1, 2, ..., I^k\} \times \{1, 2, ..., K\} \times \chi$

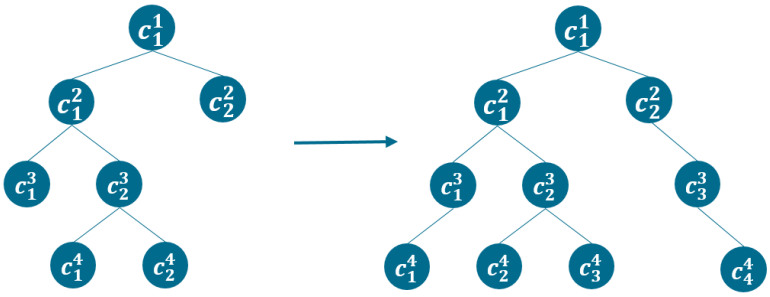


Figure A.2: Framework hypothesis regarding final nodes

$$p(c_i^k) = \frac{\|x \in c_i^k\|}{\|x \in \chi\|}$$

How to measure the Purity

Defining an Impurity Function is needed in order to measure the purity of the set of subsets:

$$\Pi\left(p(c_1^k), p(c_2^k), ..., p(c_{I^k}^k)\right)$$

Such that:

1. $\Pi\left(p(c_1^k), p(c_2^k), ..., p(c_{I^k}^k)\right)$ achieves its maximum at the point $\left(\frac{1}{I^k}, \frac{1}{I^k}, ..., \frac{1}{I^k}\right)$
2. $\Pi\left(p(c_1^k), p(c_2^k), ..., p(c_{I^k}^k)\right)$ achieves its minimum at the points: $(1, 0, 0, ..., 0)$, $(0, 1, 0, ..., 0)$, $(0, 0, 1, ..., 0)$,..., $(0, 0, 0, ..., 1)$.
3. $\Pi\left(p(c_1^k), p(c_2^k), ..., p(c_{I^k}^k)\right)$ is a symmetric function of $(p(c_1^k), p(c_2^k), ..., p(c_{I^k}^k))$

The impurity for a split at a node is given by: $\forall k \times i \in \{1, 2, ..., K\} \times \{1, 2, ..., I^K\}$

$$Impurity\ of\ c_i^k = I(c_i^k) = \Pi\left((p(c_1^{k+1}|c_i^k), p(c_2^{k+1}|c_i^k), ..., p(c_{I^{k+1}}^{k+1}|c_i^k)\right)$$

Although the impurity function has been defined for splits with any number of sub-classes, it is worth noting that when using decision trees, in most cases, the impurity function is only used for splits in two child-leaves, and thus at every node.

For example: the split of the c_i^k node divides the c_i^k class in two sub-classes c_j^{k+1} and c_{j+1}^{k+1} with $j \in \{1, 2, I^{k+1} - 1\}$.

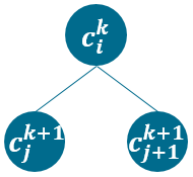


Figure A.3: Split for the c_i^k node.

Two main impurity functions are used for decision trees: the Gini Index and the Entropy:

Gini Index

The Gini Index is based on the decrease of impurity, in this case, the decrease of impurity induced by a split: for example, the decrease of impurity induced by the split on the c_i^k node.

The test will split a node in two sub-classes: c_j^{k+1} and c_{j+1}^{k+1}.

The impurity is then given by:

$$I_{Gini}\left(c_i^k\right) = \sum_{t=(j,j+1)} p_t \times (1 - p_t) = 1 - p_j^2 - p_{j+1}^2 \tag{A.11}$$

Entropy

Entropy is based on the information theory, the goal of entropy is to reduce the randomness of the choice.

For the same example, the decrease of impurity given by entropy is:

$$I_{Entropy}\left(c_i^k\right) = -\sum_{t=(j,j+1)} p_t \times \log_2(p_t) = -p_j \times \log_2(p_j) - p_{j+1} \times \log_2(p_{j+1}) \tag{A.12}$$

Although the Gini Index seems different from the Entropy, in application it has almost no impact on the model's output. Laura Elena Raileanu and Kilian Stoffel showed that, in 98% of the cases, this choice had no effect on the output: *Theoretical comparison between the Gini Index and Information Gain criteria* [13] that 98% of the trees tested were exactly the same using either Gini Index or Entropy as impurity functions. Hereunder is a comparison of the impurities obtained by Gini and Entropy[1] functions:

[1] A scaling was made to better visualize the similarities between the two functions.

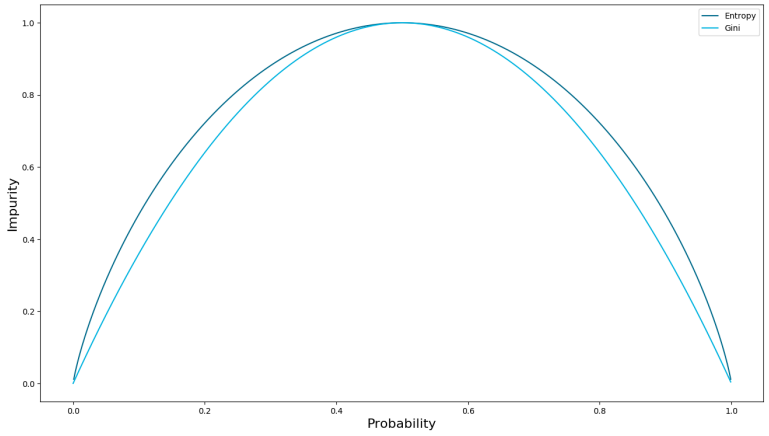


Figure A.4: Gini Index and Entropy comparison for a split in two child nodes (scaled)

A.3 Random Forest

In order to better understand the strength of the bootstrap method, let's study the variance of the Random Forest prediction:

$$\sigma^2 \left(T_{Random\ \ Forest} \left(x \right) \right) = \sigma^2 \sum_{b=1}^{B} \left(\frac{T_b \left(x \right)}{B} \right)$$

Under the hypothesis that trees have small dependencies, which can be done using random Bootstrap samples:

$$\sigma^2 \sum_{b=1}^{B} \left(\frac{T_b \left(x \right)}{B} \right) \approx \sum_{b=1}^{B} \sigma^2 \left(\frac{T_b \left(x \right)}{B} \right) = \frac{1}{B^2} \sum_{b=1}^{B} \sigma^2 \left(T_b \left(x \right) \right) = \frac{1}{B} \overline{\sigma^2 \left(T_b \left(x \right) \right)} \tag{A.13}$$

Under the hypothesis that every variance for each tree is bounded, the limit of the random forest variance converges to 0 as the number of perfectly independent Bootstrap samples tends to infinity.

Although, it is clear that the hypothesis of the Bootstrap samples being perfectly independent is far from being true, since they are based on the same dataset; as long as they are not 100% correlated and as long as the prediction models are coherent, the variance will tend to decrease as the number of Bootstrap samples increases.

Algorithm 2 Random Forest Algorithm

Require: χ: training dataset; B: number of trees in the forest;
p_{row} size of the bootstrap sample for observations; p_{col} size of the bootstrap sample for variables.
for $b \in [1:B]$ **do**
-Draw a bootstrap sample of size p_{row} of observations from the training data
-Select p_{col} variables at random from the N variables
-Fit a Decision Tree T_b to the bootstrapped data:

$$T_b(x) = \sum_{i=1}^{I^K} v_{i,b}^K \mathbb{1}\left(x \in c_{i,b}^K\right)$$

end for
Output the ensemble of trees, to make prediction at a new point:
return

$$T_{Random\ Forest}\left(x\right) = \frac{1}{B}\sum_{b=1}^{B} T_b\left(x\right)$$

Appendix B

Variables analysis of the Fire Database

Through this section will be presented all the Fire Database variables. The values of each variable will be displayed and a study of correlations will be presented, in order to have a more concrete look at the relations between the variables in the database.

B.1 Analysis for numerical data

This section will present the numerical variables, first by displaying the densities of every variable and then by displaying the correlation between these variables.

B.1.1 Densities and cumulative distribution for numerical variables

In this section, densities and cumulative distributions will be presented for all the numerical variables. Some of the variables having extremely high values, they needed to be displayed in logarithmic scale.

Sum_insured The distribution of the variable Sum_insured is displayed hereunder in logarithmic scale for more visibility as it has very high values.

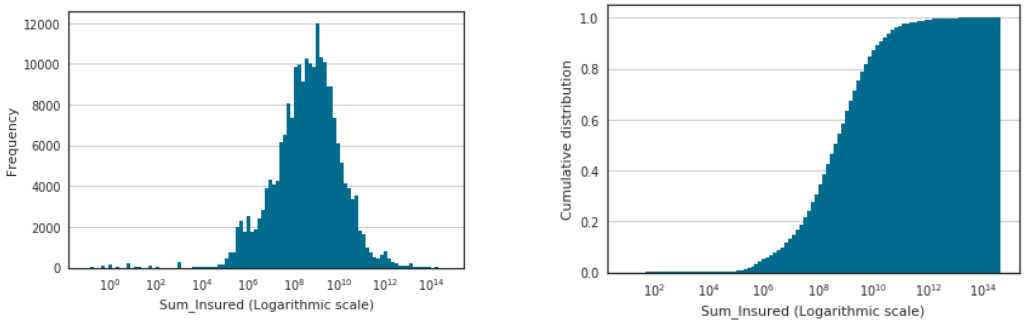


Figure B.1: Density and cumulative distribution for Sum_Insured

Level	10%	30%	50%	70%	90%
Quantile	4.0×10^6	8.6×10^7	4.6×10^8	2.1×10^9	1.8×10^{10}

Since the logarithmic scale is used, 3169 lines (1.43%) were not displayed as their values were equal to zero.

PML_100% The distribution of the variable PML_100% is displayed hereunder in logarithmic scale for more visibility as it has very high values.

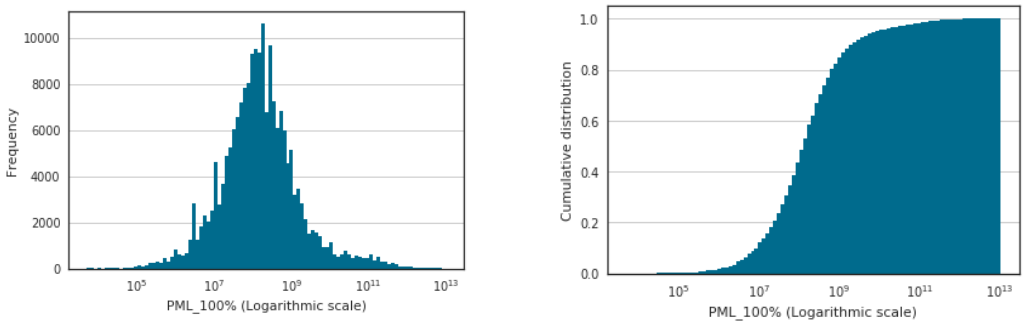


Figure B.2: Density and cumulative distribution for PML_100%

Level	10%	30%	50%	70%	90%
Quantile	0.0	30,051 k	108,239 k	304,898 k	1,895,758 k

Since the logarithmic scale is used, 23853 lines (10.81%) were not displayed as their values were not in the plot range (either < 5000 or $> 10^{13}$).

Contract_Length The distribution of the variable Contract_Length is displayed hereunder:

$$Contract\_Length = \frac{DaysBetween(Inception\_Date, Expiry\_Date)}{365.25}$$

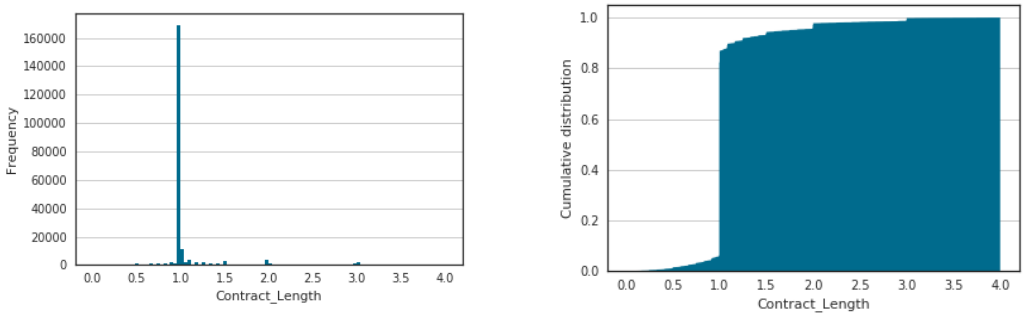


Figure B.3: Density and cumulative distribution for Contract_Length

Level	10%	30%	50%	70%	90%
Quantile	1.00	1.00	1.00	1.00	1.16

In order to make the graph more readable, 583 lines (0.26%), were not displayed, as their values were above 4.
Peaks for integers were predictable as most of the contracts are yearly covers with 85.12% of Contract_Length values are between 0.9 and 1.1.

PML_SCOR The distribution of the variable PML_SCOR is displayed hereunder in logarithmic scale for more visibility as it has very high values.

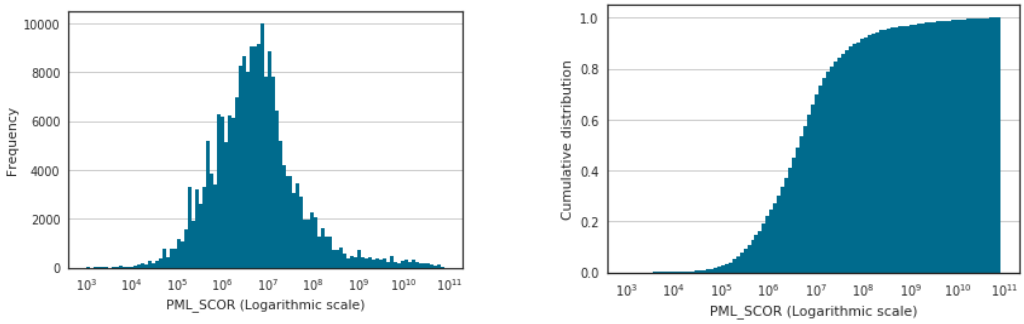


Figure B.4: Density and cumulative distribution for PML_SCOR

Level	10%	30%	50%	70%	90%
Quantile	3.2×10^5	1.7×10^6	4.6×10^6	1.1×10^7	7.8×10^7

Since the logarithmic scale is used 2993 lines (1.36%), were not displayed as their values were equal to zero.
In logarithmic scale, SCOR's Probable Maximum Loss shows a somewhat symmetrical density around its median, with a heavier tail for high values.

SCOR_EGPI The distribution of the variable SCOR_EGPI is displayed hereunder in logarithmic scale for more visibility as it has very high values.

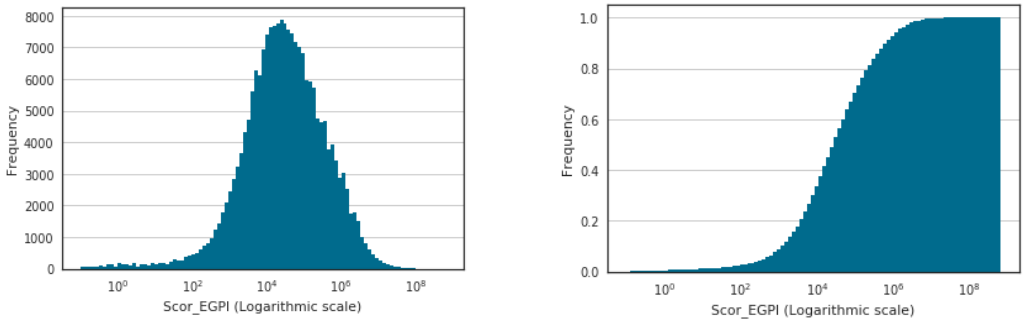


Figure B.5: Density and cumulative distribution for SCOR_EGPI

Level	10%	30%	50%	70%	90%
Quantile	123	5,894	23,243	90,411	637,479

Since the logarithmic scale is used, 16236 lines (7.36%) were not displayed as their values were equal to zero.
When plotted in the logarithmic scale, SCOR's Expected Gross Premium Income of the contract shows a skewed Gaussian-like density with a heavier tail for small values.

SCOR_PML_% The distribution of the variable SCOR_PML_% is displayed hereunder.

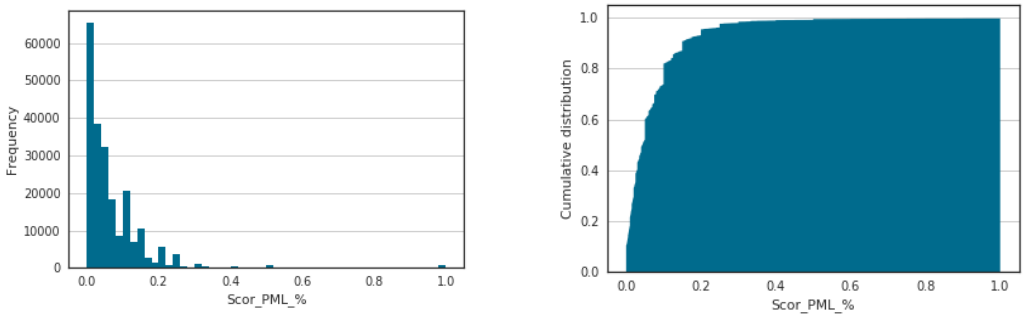


Figure B.6: Density and cumulative distribution for SCOR_PML_share

Level	10%	30%	50%	70%	90%
Quantile	0%	2%	4.3%	8%	15%

The peak at zero is due to contracts on which SCOR has a very little share, as well as contracts on which the Probable Maximum Loss was either equal to zero (1.24%) or missing (10.23%), in both cases, it was considered that the Probable Maximum Loss was equal to zero, therefore SCOR's share of Probable Maximum Loss also had to be equal to zero.

B.1.2 Correlations between numerical variables

Having highly correlated variables can be problematic for many reasons:

- Having two "similar" variables will increase the weight of these variables, compared to others, when using only samples of variables, which can bias the model by overestimating the importance of certain variables (cf. Bootstrap in the Random forest section 2.1.3);
- Operations on the database, such as diagonalization or inversion, can take much longer as the matrix can have a rank lower than its number of columns.

For these reasons, the study of the correlation between variables is necessary. In order to build a matrix of correlations, Pearson's correlation estimator is introduced:

Definition B.1.1. Correlation coefficient: Let X and Y, two random variables. The correlation between X and Y is:

$$\rho_{X,Y} = \frac{cov\left(X,Y\right)}{\sqrt{Var\left(X\right) \times Var\left(Y\right)}} \tag{B.1}$$

Definition B.1.2. Person's correlation estimator:

Let $x = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}$ and $y = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}$ be two n-sized random samples of X and Y.

Let Pearson's estimator for correlation:

$$\boxed{r_{x,y} = \frac{\sum_{i=1}^{n}\left(x_i - \overline{x}\right)\left(y_i - \overline{y}\right)}{\sqrt{\sum_{i=1}^{n}\left(x_i - \overline{x}\right)^2 \times \sum_{i=1}^{n}\left(y_i - \overline{y}\right)^2}}} \quad \text{with} \quad \overline{x} = \frac{\sum_{i=1}^{n} x_i}{n},\ \overline{y} = \frac{\sum_{i=1}^{n} y_i}{n} \tag{B.2}$$

Hereunder is the correlation matrix obtained using Pearson's estimator for correlation:

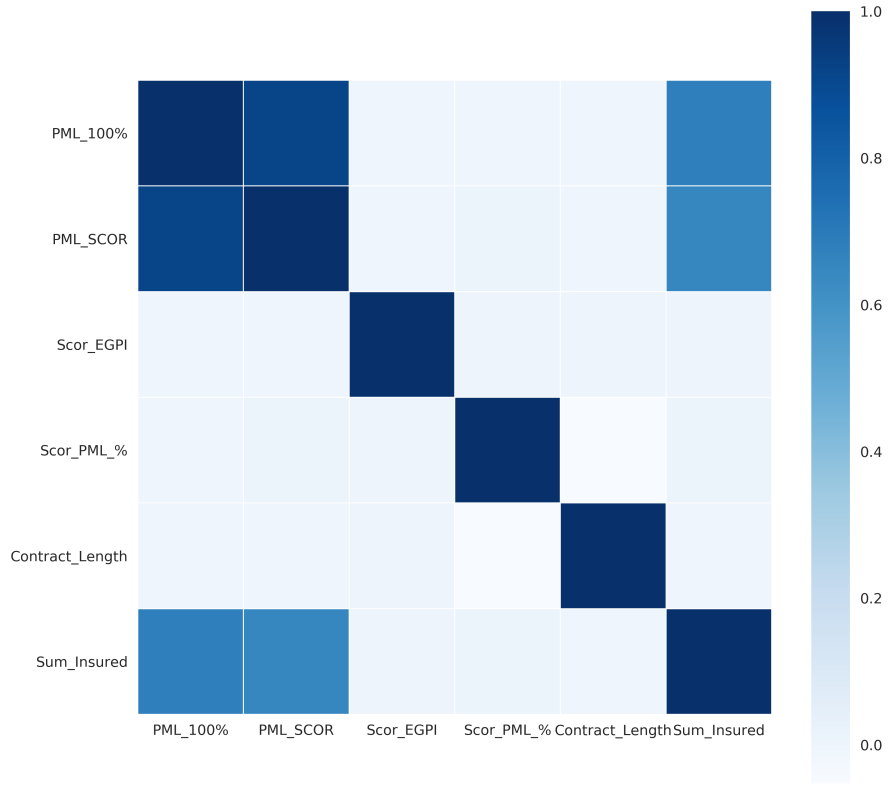


Figure B.7: Correlations between Numerical variables

In general, all the numerical variables have low correlations between each other, with two exceptions:

1. PML_100% and PML_SCOR have a correlation of 91.83%. Indeed, the higher the total probable loss is, the higher SCOR's probable loss will be. The variable SCOR_PML_% has been created in order to take into account SCOR's share without replicating the volume effect that cause the high correlation. Therefore, only the variable PML_100& and SCOR_PML_% are kept in the study database.

2. Sum_Insured with PML_SCOR and PML_100% having a correlation of 65.44% and 68.07% respectively, which seems logical, the more risks SCOR insures, the more it can loose.

Considering these high correlations when building the variables samples is necessary in order to allocate the same weight to all variables. For example, if both PML_100% and PML_SCOR were to be kept when building the variable samples, since they are extremely correlated, it would be almost the same as one of these two variables having a probability of belonging in the variable samples two times higher than the other variables.

B.2 Analysis for categorical data

This section will present the categorical variables, first by displaying the frequencies of the categories for each variable and then by displaying the correlation between these variables.

B.2.1 Frequencies of categories for categorical variables

This section will go through all the categorical variables[1], displaying the frequency for each of their categories.

Actuarial_segment Hereunder are the frequencies for Actuarial Segments:

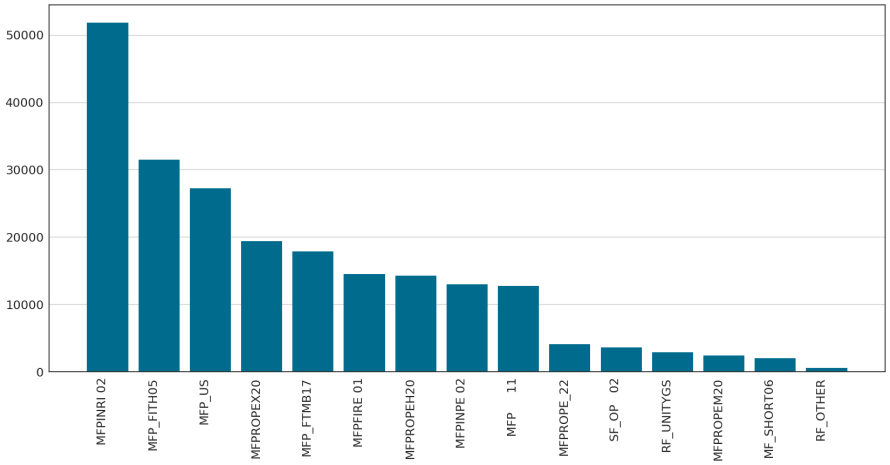


Figure B.8: Actuarial_Segment

For more clarity only the categories that are present more than 500 times are displayed (27 out of 42 categories containing 2613 lines have been omitted).
The actuarial segment variable contains the actuarial segment to which the claim belongs, it is built mainly from the Lob and Contract Nature, based on arbitrary criteria. The segmentation being challenged in this paper will not be used to predict the new classes, but to compare the new models to the existing one.

Reminder: These data analyses are done on the Fire Database (claims with Lob=Fire).

[1] The variables Subsidiary and Subsidiary_Ledger were not displayed for confidentiality reasons

Country_Claim Hereunder are the frequencies for Country_Claim:

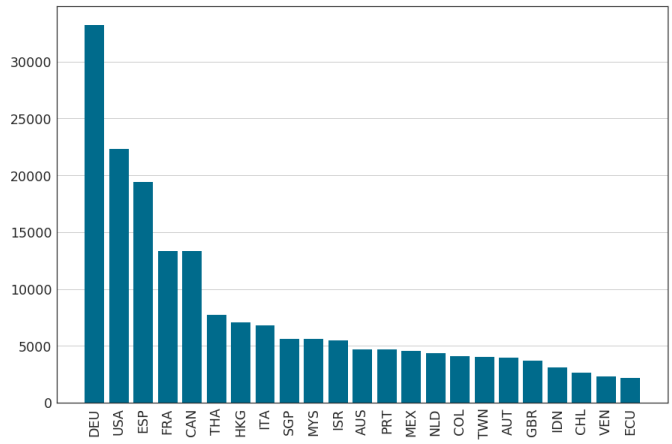


Figure B.9: Country_Claim

For more clarity only the categories that are present more than 2000 times are displayed (178 out of 201 categories containing 34578 lines have been omitted). Country_Claim is the country in which the claim happened.

Sob Hereunder are the frequencies for Sob:

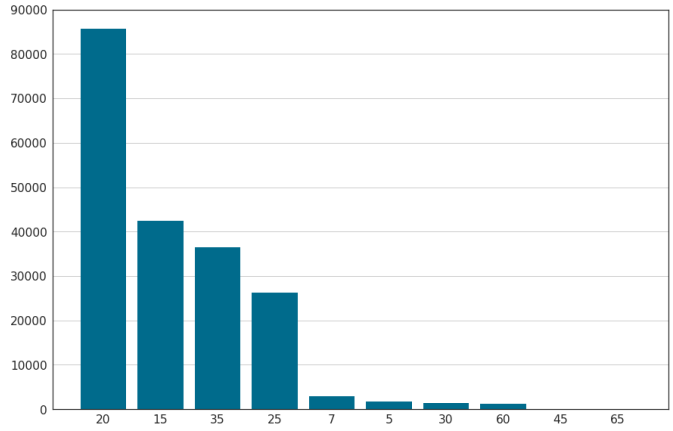


Figure B.10: Sob

For more clarity only the categories that are present more than 500 times are displayed (27 categories containing 2613 lines have been omitted).

Sob refers to the Scope of Business, for example:

Code	Label
5	Residential/Personal
7	Multi-Family Residential
15	Commercial
20	Industrial Non-petrochemical
25	Industrial Petrochemical
30	Financial Institutions
35	Infrastructures/Civil Works
45	Professional Services/Trades
60	Transport
65	Aviation / Space

Table B.1: Sob categories

Top Hereunder are the frequencies for Top:

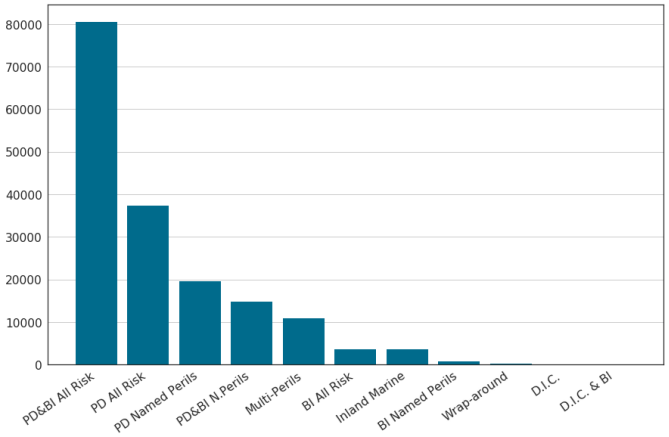


Figure B.11: Top

Top refers to the Type of Policy, for example:

- PD stands for Physical damage;
- BI stands for Business Interruption;
- DIC
- wrap around

Claim_Cause Hereunder are the frequencies for Claim_Cause:

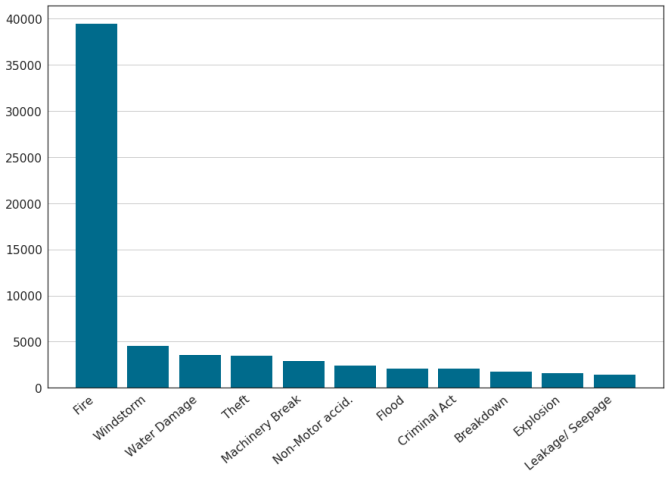


Figure B.12: Claim_Cause

For more clarity only the categories that are present more than 500 times are displayed (87 out of 102 categories containing 5243 lines have been omitted). Also, the category corresponding to missing value is not displayed, it is the biggest category with 147532 lines (66.86% of the Fire Database).

Fac_Sector Hereunder is the frequencies for Fac_Sector.

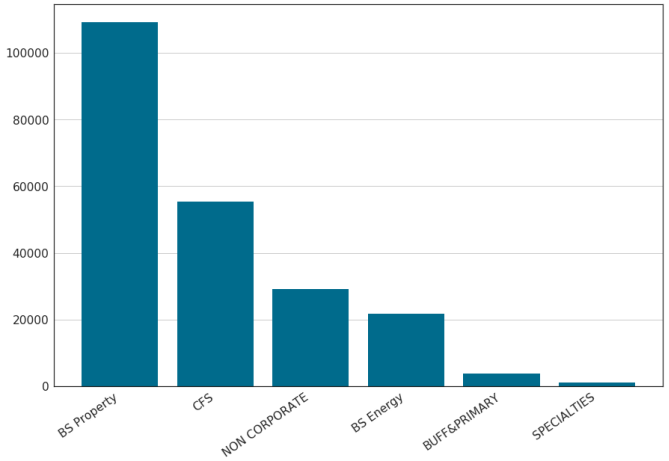


Figure B.13: Fac_Sector

For more clarity only the categories that are present more than 500 times are displayed (5 out of 10 categories containing 1281 lines have been omitted). Fac_Sector is an internal

segmentation for Facultative contracts. BS stands for Business Solutions and CFS for Corporate Functions Solutions.

Risk_Nature Hereunder are the frequencies for Risk_Nature:

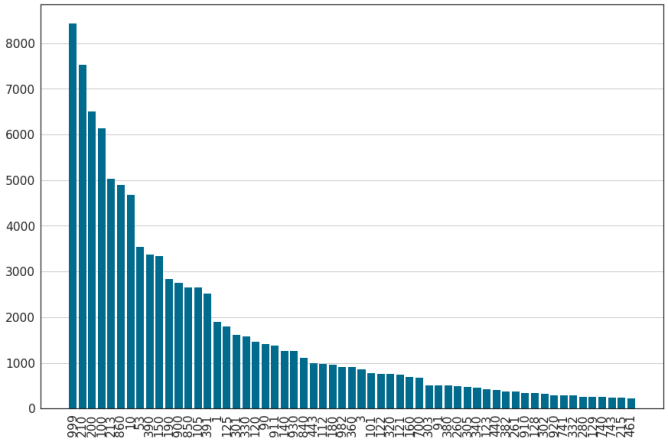


Figure B.14: Risk_Nature

For more clarity only the categories that are present more than 500 times are displayed (102 out of 117 categories containing 35007 lines have been omitted). Risk_Nature refers to the underlying risk. For example 100 refers to food products and 200 to office buildings.

CLM_UW_Y Hereunder are the frequencies for CLM_UW_Y:

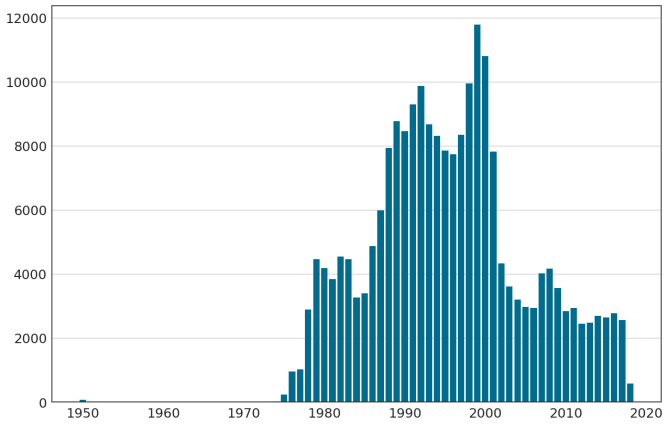


Figure B.15: CLM_UW_Y

The years from 1950 to 1974, containing 352 claims have been omitted. A fall in the number of claims right after SCOR's downgrade in the end of 2002 can be perceived.

Such a drop in the number of claims can induce a bias in the model. For example: a change in the underwriting policy after 2002.

Main_currency Hereunder are the frequencies for Main_currency:

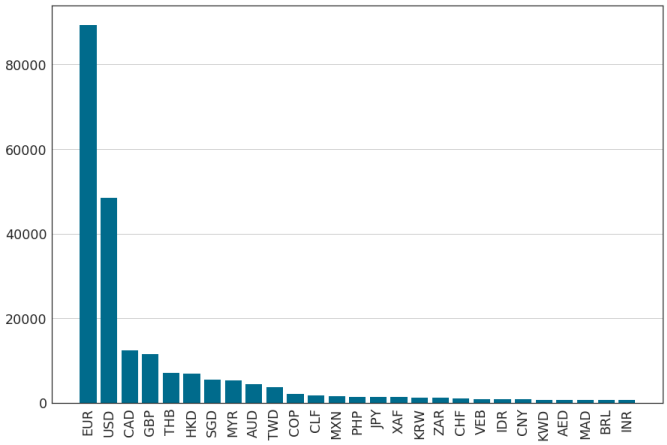


Figure B.16: Main_currency

For more clarity only the categories that are present more than 500 times are displayed (101 out of 128 categories containing 6618 lines have been omitted).

Nature Hereunder are the frequencies for Nature:

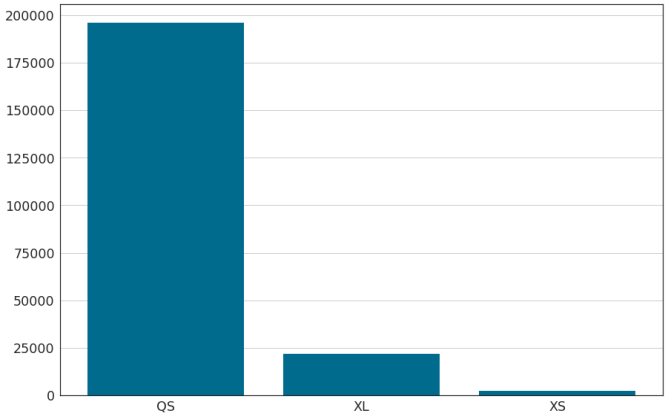


Figure B.17: Nature

Nature of the contract refers to the type of covers.

- QS: Quota-share;
- XL, XS: Excess of Loss.

Geo_Insured Hereunder are the frequencies for Geo_insured:

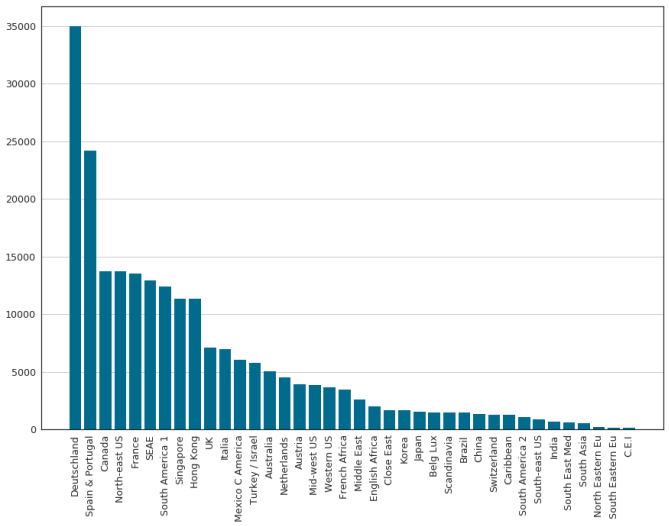


Figure B.18: Geo_Insured

For more clarity only the categories that are present more than 5000 times are displayed (25 out of 39 categories containing 41538 lines have been omitted). Geo_Insured is the geographical zone in which the contract is related to.

Follow_Up The Follow_Up variable is not displayed as 99.4% of the lines belong in the same category, hereunder are the details for each category:

Label of the category	Number of occurrences	Percentage
No_Follow_Up	219336	99.40%
Deleg Underw.	919	0.42%
Ponct. Follow-up	191	0.09%
Annual visit	108	0.05%
Permanent f.u.	52	0.02%
Ponct. no F.U.	27	0.01%
Rev 12-36 months	25	0.01%
Annuity	1	$\approx 0\%$

Table B.2: Frequency of categories for Follow_Up

B.2.2 Correlation between categorical variables

Computing correlations between categorical variables is necessary for the same reasons as for numerical variables. The usual correlations based on Pearson's estimator being only valid for numerical data, a new form of correlation between categorical data must be introduced.

In this analysis, Cramer's V was used as it considers frequencies of categories to compute a form a dependence between categorical variables.

Cramer's V

Cramer's V is based on Pearson's Chi-squared test of independence: χ^2

Definition B.2.1. χ^2: Let:

- X and Y samples of length n for two categorical variables with respectively r and s different categories;
- $n_{i,.}$ the random variable of the frequency of the i-th category for the variable X;
- $n_{.,j}$ the random variable of the frequency of the j-th category for the variable Y;
- $n_{i,j}$ the random variable of the frequency of i-th category for the variable X and j-th category for the variable Y.

$$\boxed{\chi^2 = \sum_{i=1}^{r}\sum_{j=1}^{s} \frac{\left(n_{i,j} - \frac{n_{i,.}n_{.,j}}{n}\right)^2}{\frac{n_{i,.}n_{.,j}}{n}}} \tag{B.3}$$

The χ^2 measures the dependence between two categorical variables, X and Y, by studying the distribution in each category for each variable. Although the χ^2 statistic is able to determine whether two categorical variables are independent or not, it does not compute a form of correlation, as its values are not contained in $[-1, 1]$. Moreover, its value heavily relies on the sample size and the number of categories for each variable, which should not be the case for a dependence measure.

In order to obtain an equivalence of correlation for categorical variables, finding a way to normalize the χ^2 statistic is needed. To do so, the limits in value of χ^2 have to be studied.

Study of the value of χ^2 for $X \perp\!\!\!\perp Y$: From the previously introduced notation:

$$P(X = i) = \frac{n_{i,.}}{n} \quad \forall i \in \{1, 2, ..., r\} \quad \text{and} \quad P(Y = j) = \frac{n_{.,j}}{n} \quad \forall i \in \{1, 2, ..., s\}$$

The hypothesis $X \perp\!\!\!\perp Y$ gives $P(X \cap Y) = P(X) \times P(Y)$, therefore:

$$\frac{n_{i,.}}{n} \times \frac{n_{.,j}}{n} = P(X = i) \times P(Y = j) = P(X = i \cap Y = j) = \frac{n_{i,j}}{n}$$

thus

$$n_{i,j} = \frac{n_{.,j} \times n_{i,.}}{n} \qquad \forall i \times j \in \{1, 2, ..., r\} \times \{1, 2, ..., s\} \tag{B.4}$$

Hence, by using B.4 the following result can be obtained:

$$\chi^2 = \sum_{i=1}^{r} \sum_{j=1}^{s} \frac{\left(n_{i,j} - \frac{n_{i,.} n_{.,j}}{n}\right)^2}{\frac{n_{i,.} n_{.,j}}{n}} = 0 \tag{B.5}$$

Thus, if $X \perp\!\!\!\perp Y$: $\chi^2 = 0$.

As χ^2 is a sum of squares, it is non-negative. Therefore, the minimum of χ^2 is reached when used to measure the dependency between two independent variables.

Now that the minimum value of χ^2 has been found. It is necessary to study the boundness of χ^2.

Study of the maximum of χ^2:

To do so, the division of the χ^2 formula by $\frac{n_{i,.} n_{.,j}}{n}$ can be studied:

$$\chi^2 = \sum_{i=1}^{r} \sum_{j=1}^{s} \frac{\left(n_{i,j} - \frac{n_{i,.} n_{.,j}}{n}\right)^2}{\frac{n_{i,.} n_{.,j}}{n}} = n \times \sum_{i=1}^{r} \sum_{j=1}^{s} \left(\frac{n_{i,j}^2}{n_{i,.} n_{.,j}} - 1 \right) \tag{B.6}$$

As $0 < n_{i,j} \leq n_{i,.} = \sum_{j=1}^{s} n_{i,j}$:

$$\sum_{i=1}^{r} \sum_{j=1}^{s} \frac{n_{i,j}^2}{n_{i,.} n_{.,j}} \leq \frac{n_{i,j}}{n_{.,j}} = \sum_{j=1}^{s} \frac{n_{.,j}}{n_{.,j}} = s \tag{B.7}$$

By injecting this inequality in B.6 the following inequality can be obtained:

$$\chi^2 \leq n(s-1)$$

Using the symmetry between s and r another inequality can be obtained:

$$\chi^2 \leq n(r-1)$$

Both of these inequalities imply:

$$\boxed{\chi^2 \leq n \times min((s-1), (r-1))}$$

.

By dividing χ^2 by its maximum, a measure of dependence between two categorical variables in $[0, 1]$ is computed, this newly defined measure is Cramer's V:

Definition B.2.2. χ^2:

$$\boxed{V_{cramer} = \sqrt{\frac{\chi^2}{n \times min(s-1, r-1)}} \in [0, 1]} \tag{B.8}$$

With:

– χ^2 Pearson's statistics;

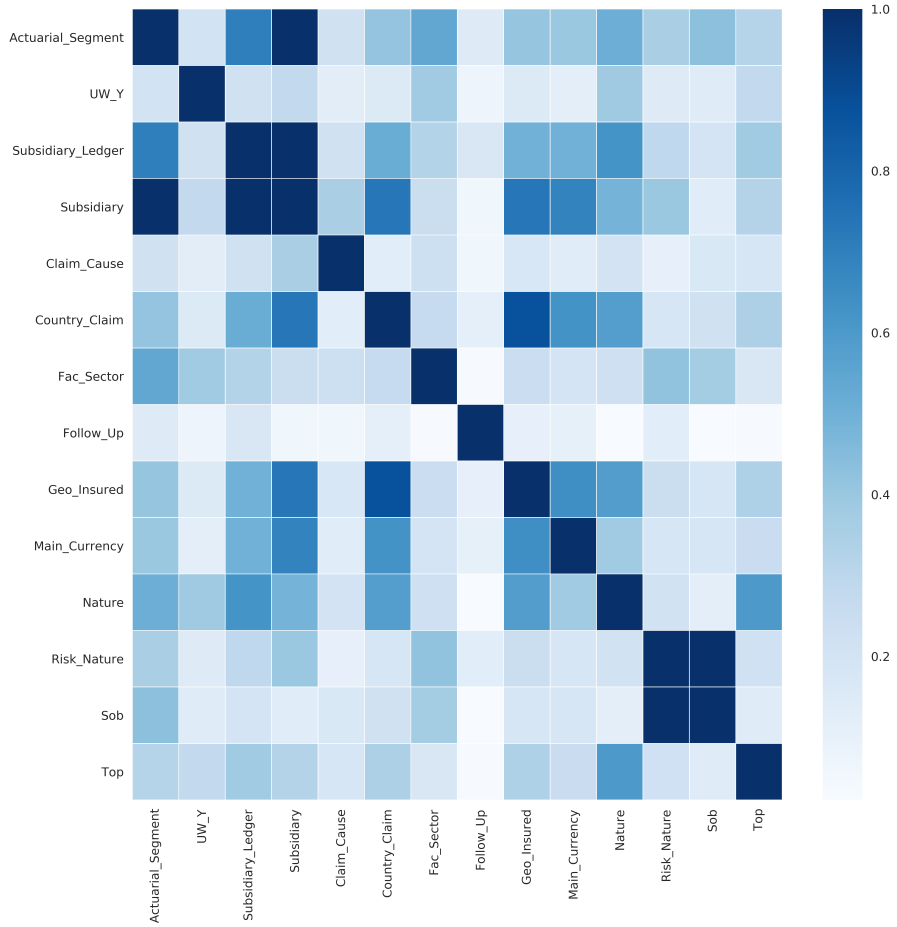
– n sample size;

– s the number of columns;

– r the number of rows.

If the variables are independent: $V_{cramer} = 0$ while $V_{cramer} = 1$ for entirely correlated variables.

Hereunder is the correlation matrix obtained for categorical variables using Cramer's V:



It is clear that some categorical variables are highly correlated with others:

– Subsidiary with Actuarial_Segment: 99.99%. By construction, the Actuarial Segment is different for every Subsidiary. Therefore, the Subsidiary can be deduced from the Actuarial_Segment, which is translated by a Cramer's V ≈ 1.

- Subsidiary with Subsidiary_Ledger: 99.99%. The Subsidiary_Ledger is a subclass of the Subsidiary, therefore, the Subsidiary can be deduced when knowing the Subsidiary_Ledger, resulting in a Cramer's V ≈ 1.
- Risk_Nature with Sob: 99.81%. Also, by construction, the Sob can be deduced from the Risk_Nature, resulting in a Cramer's V ≈ 1.

In order to allocate the same weight to all variables when using bootstrap on variables, a choice must be made between taking Subsidiary or Actuarial_Segment, Subsidiary or Subsidiary_Ledger, Risk_Nature or Sob.

B.3 Analysis for indicators

B.3.1 Densities and cumulative distributions for indicators

In this section, the density and cumulative distribution are displayed for each indicator. The Incurred and Paid amounts are displayed in logarithmic scale as they had a wide range of values.

Reminder:

- Incurred_N is the Amount of Incurred accumulated up to the year N:

$$Incurred\_N = \sum_{i=1}^{N} Inc_i$$

- TF_Incurred_N is the Duration for the incurred cash-flows, defined as:

$$TF\_Incurred\_N = \frac{\sum_{i=1}^{N} Inc_i \times development\ period_i}{\sum_{i=1}^{N} Inc_i}$$

- TF_Max_Incurred_N is a twist from the previous formula to capture the effects of claims developing and settling at zero.

$$TF\_Max\_Incurred\_N = \frac{\sum_{i=1}^{N} Inc_i \times development\ period_i}{\max\limits_{1 \leq i \leq N} (Inc_i)}$$

Incurred_N The distribution of the indicator Incurred_N is displayed hereunder in logarithmic scale for more visibility as it has very high values.

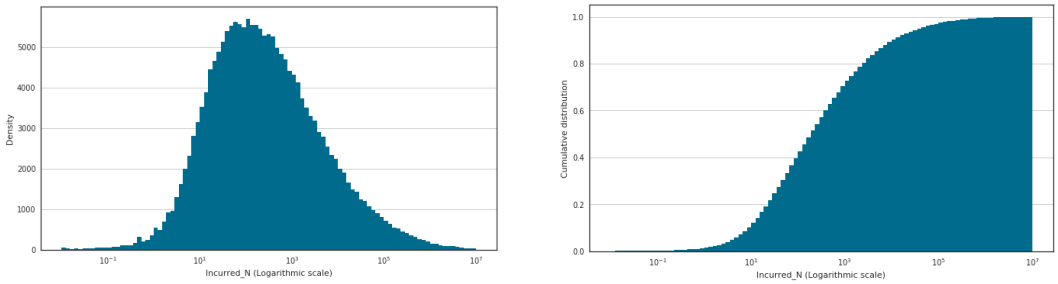


Figure B.19: Incurred_N density and cumulative distribution

Level	10%	30%	50%	70%	90%
Quantile	0.00	18.59	106.47	602.01	8353.50

Since the logarithmic scale is used, 34965 (15.85%) lines were not displayed as their Incurred amount was either negative (0.76%) or null (15.09%).

In logarithmic scale, the Incurred amount seems to have a skewed Gaussian density, with a heavier tail for high values.

TF_Incurred_N The distribution of the indicator TF_Incurred_N is displayed hereunder.

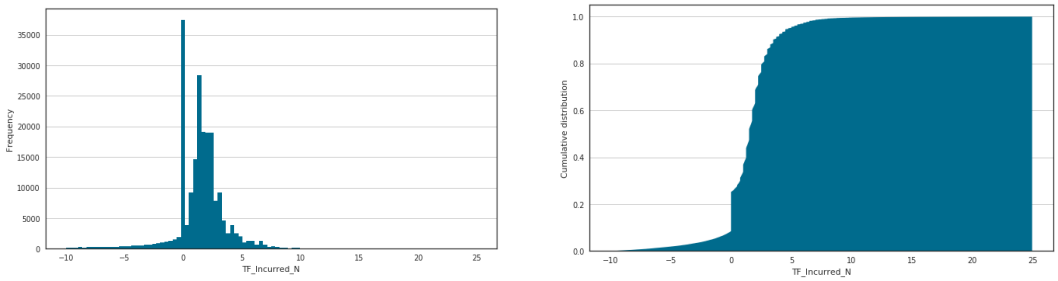


Figure B.20: TF_Incurred_N density and cumulative distribution

Level	10%	30%	50%	70%	90%
Quantile	-1.67	0.20	1.49	2.00	3.50

In order to make the graph more readable 12355 lines (5.56%) were not displayed as their values were either above 25 (0.16%) or below −10 (5.40%).

TF_Max_Incurred_N The distribution of the indicator TF_Max_Incurred_N is displayed hereunder.

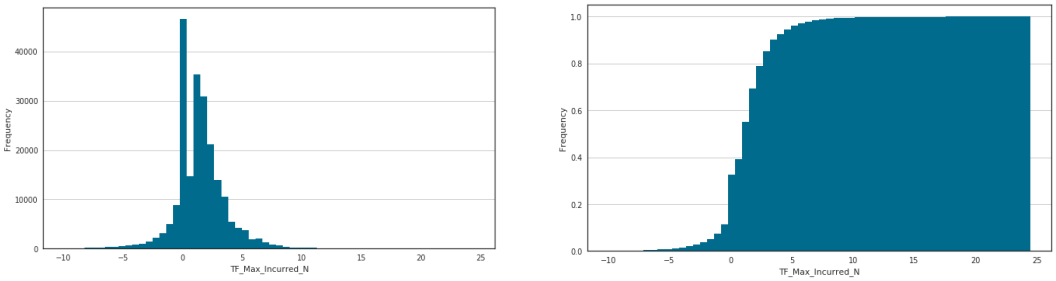


Figure B.21: TF_Max_Incurred_N density and cumulative distribution

Level	10%	30%	50%	70%	90%
Quantile	-0.42	0.05	1.38	2.18	3.78

In order to make the graph more readable 559 lines (0.25%) were not displayed as their values were either above 25 (0.03%) or below −10 (0.22%).

Paid_N The distribution of the indicator Paid_N is displayed hereunder in logarithmic scale for more visibility as it has very high values.

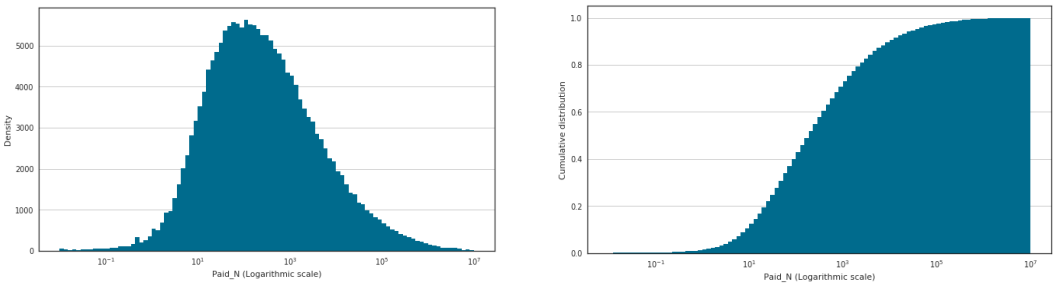


Figure B.22: Paid_N density and cumulative distribution

Level	10%	30%	50%	70%	90%
Quantile	0.00	16.63	98.17	556.32	7459.56

Since the logarithmic scale is used 37682 (17.1%) lines were not displayed as their Paid amount was either negative (0.61%) or null (16.5%). The high number of Paid amounts being equal to zero can be explained as some claims have an incurred, and are therefore present in the database but for which no payment has yet been made.

In logarithmic scale, the Paid amount seems to have a Gaussian-like density, with a heavier tail for high values.

TF_Paid_N The distribution of the indicator TF_Paid_N is displayed hereunder.

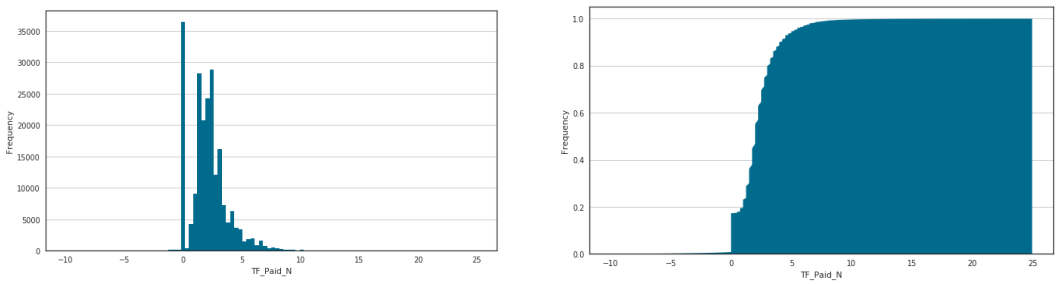


Figure B.23: TF_Paid_N density and cumulative distribution

Level	10%	30%	50%	70%	90%
Quantile	0.00	1.46	2.00	2.55	4.06

In order to make the graph more readable 1110 lines (0.50%) were not displayed as their values were either above 25 (0.13%) or below −10 (0.37%).

TF_Max_Paid_N The distribution of the indicator TF_Max_Paid_N is displayed hereunder.

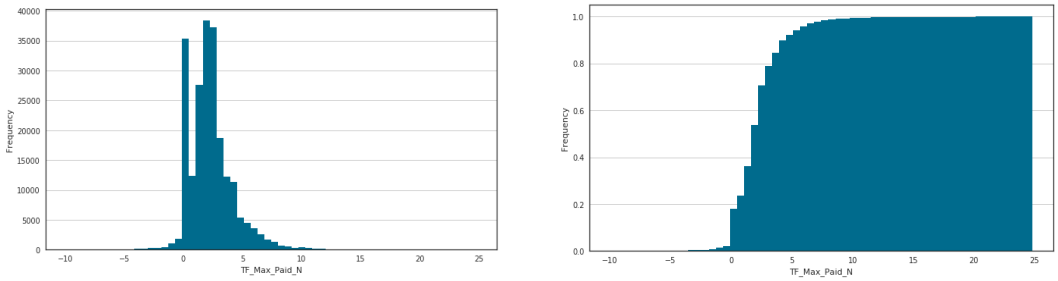


Figure B.24: TF_Max_Paid_N density and cumulative distribution

Level	10%	30%	50%	70%	90%
Quantile	0.00	1.50	2.00	2.75	4.65

In order to make the graph more readable 294 lines (0.13%) were not displayed as their values were either above 25 (0.004%) or below −10 (0.010%).

The Indicators for the years up to $N-1$ are not displayed as they are almost the same as the ones displayed just before.

B.4 Note on the reliability of the Indicators

This note will study the reliability of the indicators regarding their descriptiveness of the development patterns. This study was done for the Fire database.

The segmentation based on clusters of indicators builds homogeneous classes regarding the value of the 4 indicators: TF_Incurred, TF_Paid, TF_Max_Incurred, TF_Max_Paid. Hereunder are the distributions of the 4 classes of the segmentation selected in the previous section, a PCA was used to display the values in two-dimension[2]:

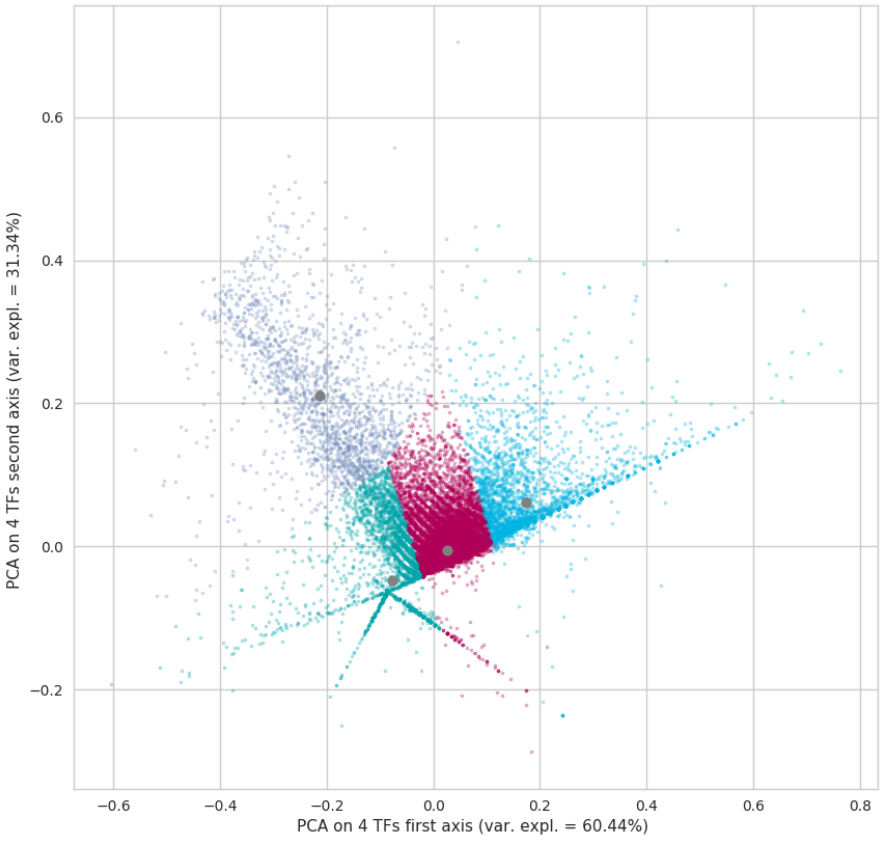


Figure B.25: Values of the four clusters for the four indicators

[2] $First\ Axis = 0.64{*}TF\_Incurred{+}0.44{*}TF\_Paid{+}0.51{*}TF\_Max\_Incurred{+}0.38{*}TF\_Max\_Paid$ (explained variance = 60.44%).
$Second\ Axis = -0.65 * TF\_Incurred + 0.55 * TF\_Paid - 0.06 * TF\_Max\_Incurred + 0.52 * TF\_Max\_Paid$ (explained variance = 31.46%).

Hereunder are the mean values for each class:

	Size	TF_Incurred_N	TF_Paid_N	TF_Max_Incurred_N	TF_Max_Paid_N
Class 0	4,089	3.65	4.57	4.23	5.53
Class 1	11,143	-0.17	0.35	0.12	0.34
Class 2	21,521	1.55	2.18	1.65	2.35
Class 3	1,852	-12.74	2.41	-1.60	2.73

Table B.3: Centroids for each class

The classes have distinct profiles, Class 0 reflects long-tail claims, class 1 claims for which they are no positive developments on Incurred, claims 2 is made of short tail claims, and class 3 contains the claims with a late positive development for Incurred. A study of the development patterns is necessary to confirm or infirm the reliability of the indicators. Hereunder are the development patterns per class:

Figure B.26: Incurred patterns for each class (4 clusters on 4 TFs)

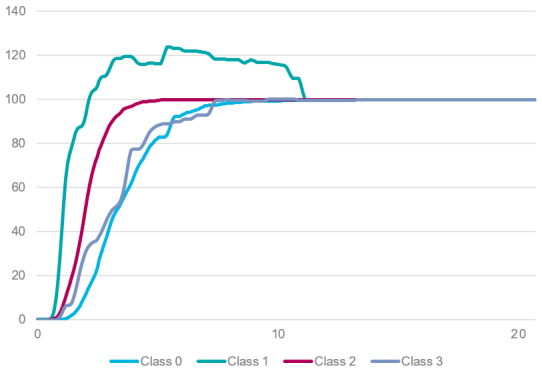


Figure B.27: Paid patterns for each class (4 clusters on 4 TFs)

The patterns for each class are very different and follow the analysis made on the indicators values. Therefore, the indicators are good indicators to split different types of development patterns.

Appendix C

Analysis of the best segmentation

This chapter provides a concrete look at the compositions of the different classes, regarding both the numerical and categorical variables.

C.1 Study of the numerical variables

Hereunder is a graph to compare the distribution of the standardized median values for numerical variables[1] and the Incurred and Paid amounts among the clusters.

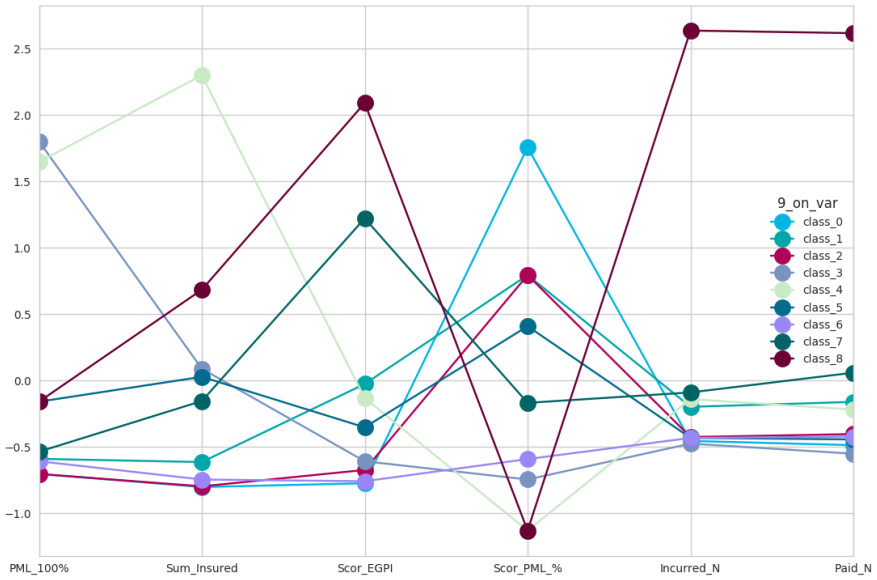


Figure C.1: Median values for numerical variables among the classes

– 0, 2 and 6 contain smaller risks that have roughly the same values for the first 3

[1] Contract_Length is not displayed because the median value of 1 was the same for every classes.

variables (those related to the size of the contract);

- 1 and 5 are clearly above 0, 2 and 6 regarding the median size of the risks;
- The class 3 contains claims related to medium risks (as it has an medium Sum_Insured) but for which the probable maximum loss is very large (reflected by the high PML). SCOR only has a small share on these claims;
- 4 has very high PML total and Sum_Insured, the claims it contains are therefore related to large risks;
- 8 on the contrary, has a medium PML total but very high Sum Insured and EGPI. Which indicates that the claims are related to large risks (regarding the sum insured) but on which the maximum loss estimated is limited (reflected by the small values for PML_100%);
- 7 is a smaller version of 8, but with a much higher PML share.

General trends can also be observed, for example the fact that SCOR tends to have big shares only on small risks.

Despite not being considered in the clustering process, the Incurred and Paid amounts are different among the classes. With the class 8 having very high median values, followed by the classes 1, 4 and 7, and then, the classes: 0, 2, 3, 5, 6.

C.2 Study of the categorical variables

The following graphs represent the percentages of each category among the classes obtained from the clusters on variables (9 clusters).

Country_Claim

Hereunder are the distributions of the country in which the claim occurred for the 9 classes:

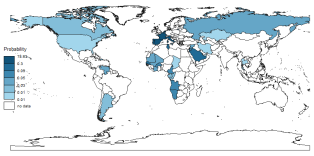


Figure C.2: Class 0

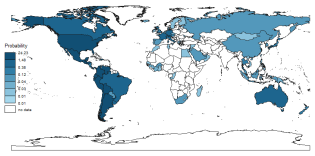


Figure C.3: Class 1

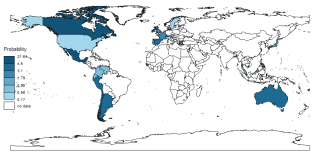


Figure C.4: Class 2

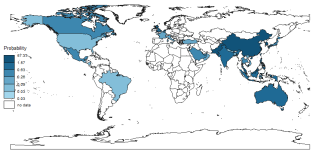


Figure C.5: Class 3

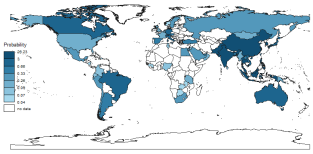


Figure C.6: Class 4

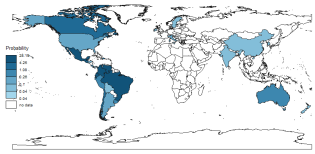


Figure C.7: Class 5

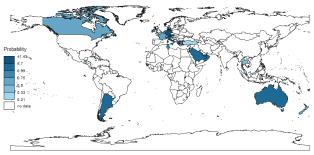


Figure C.8: Class 6

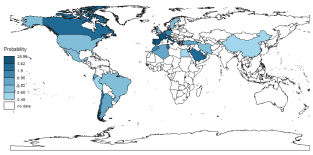


Figure C.9: Class 7

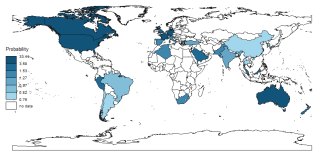
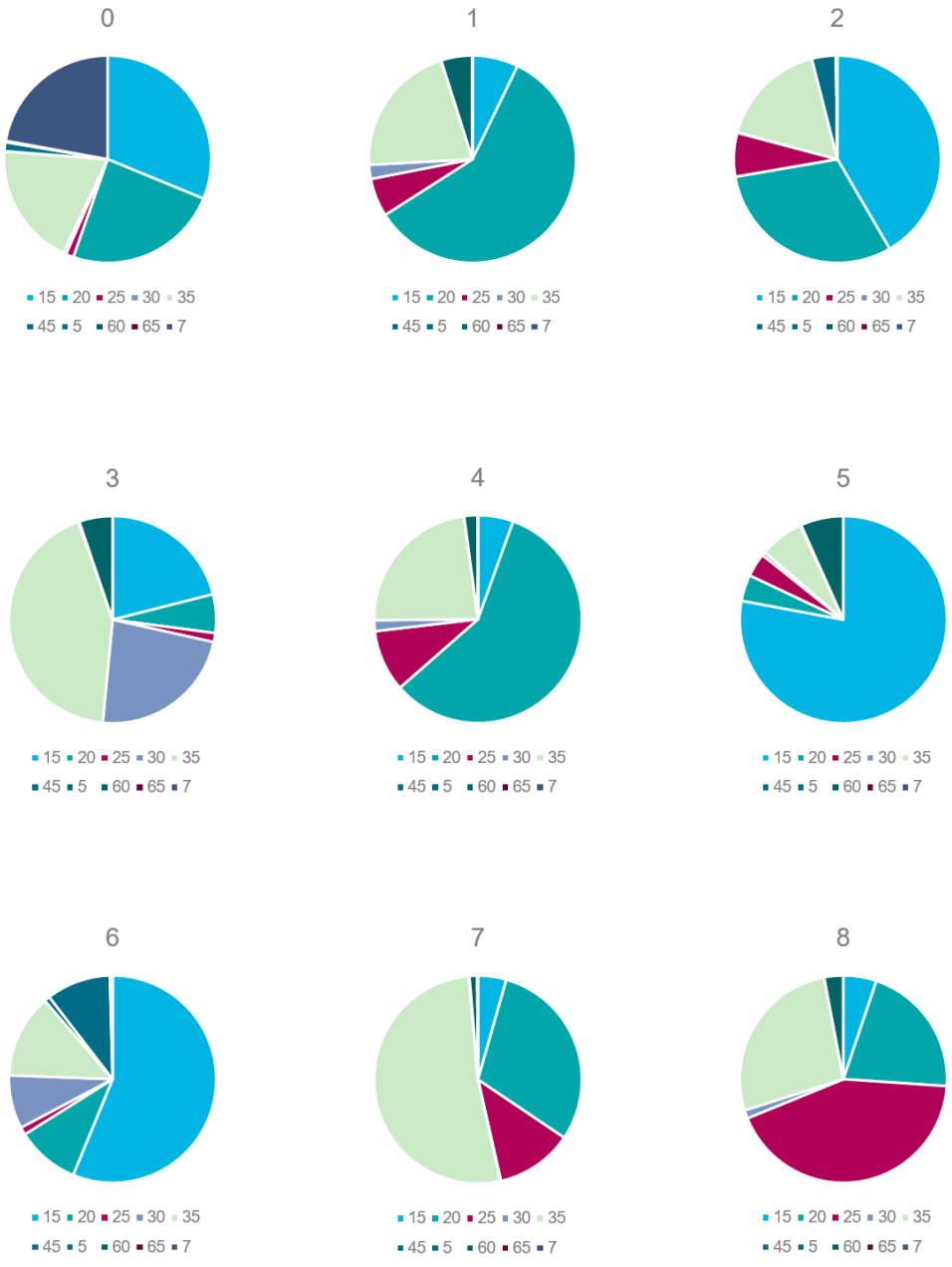


Figure C.10: Class 8

Sob

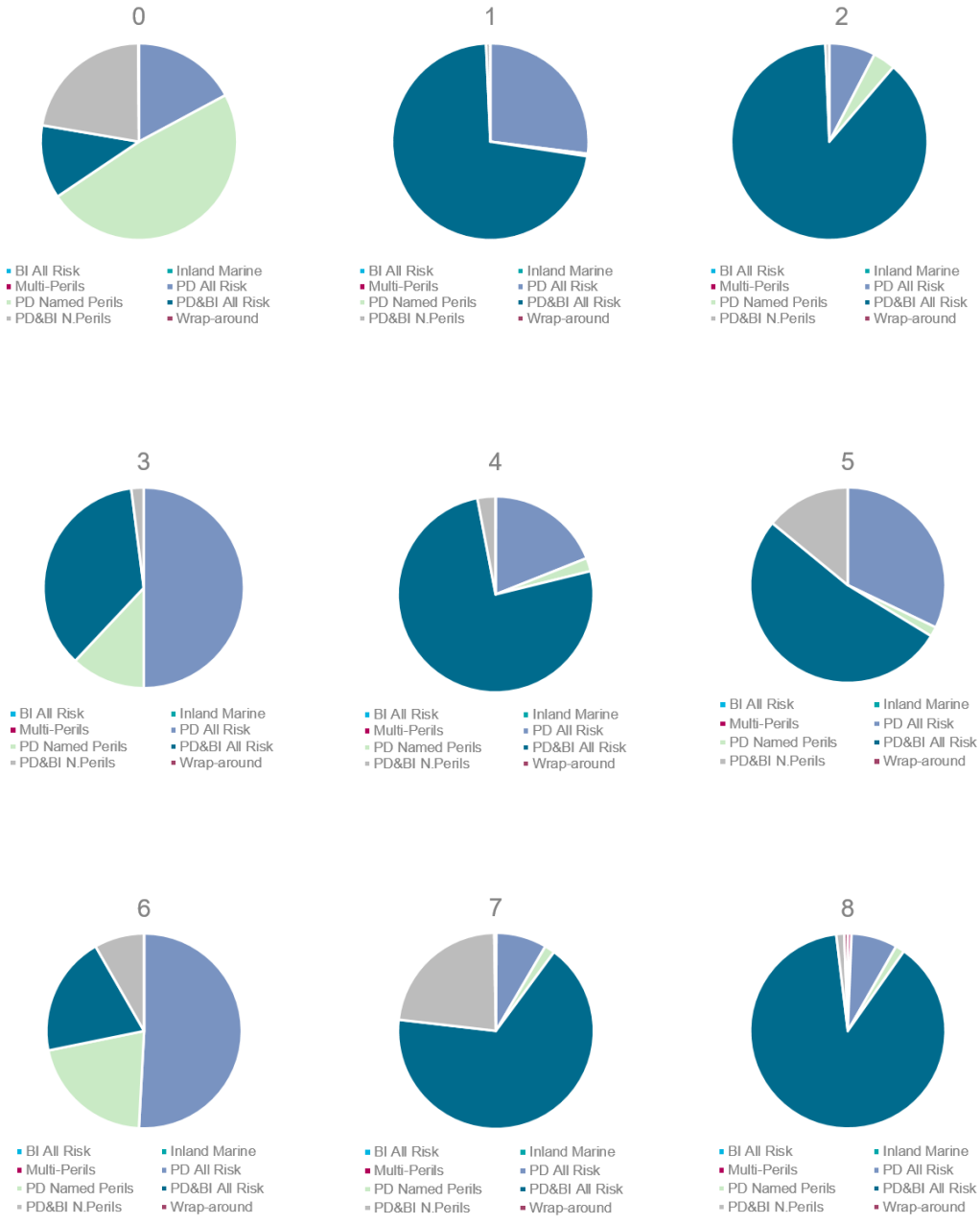
Hereunder are the distribution of Scope of business among the 9 classes:



Code	Label	Code	Label
5	Residential/Personnal	30	Financial Institutions
7	Multi-Family Residential	35	Infrastructures/Civil Works
15	Commercial	45	Professional Services/Trades
20	Industrial Non-Petrochemical	60	Transport
25	Industrial Petrochemical	65	Aviation/Space

Table C.1: Sob descriptions

Top



Top refers to the Type of Policy, for example:

- PD stands for Physical damage;
- BI stands for Business Interruption;
- DIC stands for differences in conditions (cover gaps between policies);
- wrap around: liability protection against being sued.

List of Figures

1 Impact of the segmentation regarding the estimation error iv
2 Impact of the development patterns on the values of the indicators v
3 Value assignment process for tree-based segmentations vi
4 Value assignment process for cluster-based segmentations vi
5 Predicting the 2018 cash-flow viii
6 Prediction des IBNR / Reserves viii
7 Impact d'une segmentation sur l'erreur d'estimation xi
8 Impact du type de développement sur la valeur des indicateurs xii
9 Attribution des valeurs de prédictions pour les arbres de decision xiii
10 Attribution des valeurs de prédictions pour les clusters xiii
11 Prédiction du *cash-flow* de l'annee 2018 xv
12 Prediction des IBNR / Reserves xv

1.1 Claim life cycle 5
1.2 Filling the lower triangle of developments 7
1.3 Predicting Incurred amounts 8
1.4 Visualizing the IBNR on a triangle of cumulated developments 9
1.5 Impact of the proportion of claims on the pattern of development 10

2.1 Example of Neural Network 13
2.2 Example of a Decision Tree 14
2.3 Random Forest decision making 16
2.4 Gradient Boosting iterative process 17
2.5 One-Hot Encoder 20
2.6 Comparison of the results using different clustering algorithms 23

3.1 Impact of the development patterns on the values of the Indicators 33
3.2 Composition of the whole database 34
3.3 Distribution of the lines of business 34

4.1 missMDA algorithm example 36
4.2 Process of filling missing Data 37
4.3 Percentages of explained variance for Sob/Top 38
4.4 Percentages of explained variance for Country_Claim 39

4.5 Clusters inertias on all countries . . . 40
4.6 Construction of the Country_Claim_no_NA variable . . . 42
4.7 TF_Incurred_N density difference between Risk_Nature filled or NA . . 43
4.8 TF_Incurred_N density difference between Claim_Cause filled or NA . . 43

5.1 Underwriting year bias in the median values of the indicators . . . 47
5.2 Actuarial Segmentation classes . . . 48
5.3 Predicted value assignment process for tree-based segmentations . . . 50
5.4 MSE for TF_Incurred . . . 51
5.5 MSE for TF_Paid . . . 51
5.6 MSE for TF_Max_Incurred . . . 51
5.7 MSE for TF_Max_Paid . . . 51
5.8 MSE for TF_Incurred . . . 52
5.9 MSE for TF_Paid . . . 52
5.10 MSE for TF_Max_Incurred . . . 52
5.11 MSE for TF_Max_Paid . . . 52
5.12 Predicted value assignment process for cluster-based segmentations . . . 54
5.13 MSE for TF_Incurred . . . 55
5.14 MSE for TF_Paid . . . 55
5.15 MSE for TF_Max_Incurred . . . 55
5.16 MSE for TF_Max_Paid . . . 55
5.17 MSE for TF_Incurred . . . 56
5.18 MSE for TF_Paid . . . 56
5.19 MSE for TF_Incurred . . . 56
5.20 MSE for TF_Paid . . . 56
5.21 MSE for TF_Incurred . . . 57
5.22 MSE for TF_Paid . . . 57
5.23 MSE for TF_Max_Incurred . . . 58
5.24 MSE for TF_Max_Paid . . . 58

6.1 Process used to compare the quality of the N cash-flow prediction . . . 62
6.2 Process used to compare the quality of IBNR prediction . . . 67
6.3 Distribution of the classes of 9 Clusters on variables . . . 70
6.4 Incurred developments for 9 clusters on variables . . . 71
6.5 Paid developments for 9 clusters on variables . . . 72

7.1 Distribution of the lines of business . . . 78
7.2 Incurred developments for segmentation based on tree fitted to Incurred . 81
7.3 Paid developments for segmentation based on tree fitted to Incurred . . . 81
7.4 Decicion tree fitted to Incurred with: 15 classes / 1000 claims per class min. 82

A.1 Activation functions . . . 88
A.2 Framework hypothesis regarding final nodes . . . 91
A.3 Split for the c_i^k node. . . . 92

A.4 Gini Index and Entropy comparison for a split in two child nodes (scaled) 93

B.1 Density and cumulative distribution for Sum_Insured 96
B.2 Density and cumulative distribution for PML_100% 96
B.3 Density and cumulative distribution for Contract_Length 97
B.4 Density and cumulative distribution for PML_SCOR 97
B.5 Density and cumulative distribution for SCOR_EGPI 98
B.6 Density and cumulative distribution for SCOR_PML_share 98
B.7 Correlations between Numerical variables 100
B.8 Actuarial_Segment 101
B.9 Country_Claim 102
B.10 Sob 102
B.11 Top 103
B.12 Claim_Cause 104
B.13 Fac_Sector 104
B.14 Risk_Nature 105
B.15 CLM_UW_Y 105
B.16 Main_currency 106
B.17 Nature 106
B.18 Geo_Insured 107
B.19 Incurred_N density and cumulative distribution 112
B.20 TF_Incurred_N density and cumulative distribution 112
B.21 TF_Max_Incurred_N density and cumulative distribution 113
B.22 Paid_N density and cumulative distribution 113
B.23 TF_Paid_N density and cumulative distribution 114
B.24 TF_Max_Paid_N density and cumulative distribution 114
B.25 Values of the four clusters for the four indicators 115
B.26 Incurred patterns for each class (4 clusters on 4 TFs) 116
B.27 Paid patterns for each class (4 clusters on 4 TFs) 116

C.1 Median values for numerical variables among the classes 117
C.2 Class 0 119
C.3 Class 1 119
C.4 Class 2 119
C.5 Class 3 119
C.6 Class 4 119
C.7 Class 5 119
C.8 Class 6 119
C.9 Class 7 119
C.10 Class 8 119

List of Tables

2.1 Artificial Neural Network parameters 14
2.2 Decision tree parameters 15
2.3 Random Forest parameters 16
2.4 Gradient Boosting parameters 17
2.5 Advantages and disadvantages of each algorithm 18

3.1 Original variables 29

4.1 Sob/Top comparison between number of dimensions 39
4.2 Country_Claim comparison between number of dimensions 41

5.1 Comparison of the MSE (test database) of the chosen models 59

6.1 Algorithms accuracies for Cluster-based segmentations 63
6.2 Incurred 2018 cash-flow predictions 65
6.3 Paid 2018 cash-flow predictions 65
6.4 IBNR prediction with 3 development years taken out 68
6.5 Reserves prediction with 3 development years taken out 68
6.6 Results of the IBNR prediction with 5 development years taken out 69
6.7 Results of the Reserves prediction with 5 development years taken out . . 69
6.8 Ranks of the segmentations 70
6.9 Undiscounted durations for 9 clusters on variables 72

7.1 Means of indicators per line of business 79
7.2 Results of prediction for the whole database 80

B.1 Sob categories 103
B.2 Frequency of categories for Follow_Up 107
B.3 Centroids for each class 116

C.1 Sob descriptions 120