cea | CENTRE D'ÉTUDES ACTUARIELLES

INSTITUT DES ACTUAIRES

Mémoire présenté devant le Centre d'Etudes Actuarielles pour la validation du cursus à la Formation d'Actuaire du Centre d'Etudes Actuarielles et l'admission à l'Institut des Actuaires le 20 MARS 2017

Par : Bergère Matthieu

Titre : Modélisation de transformation de devis par apprentissage automatique

Confidentialité : ☐ NON ☒ OUI (Durée : ☒ 1an ☐ 2 ans)
Les signataires s'engagent à respecter la confidentialité indiquée ci-dessus

Membre présent du jury de l'Institut des Actuaires :

Membres présents du jury du Centre d'Etudes Actuarielles :

Secrétariat :

Bibliothèque :

Entreprise :AXA Insurance UK

Nom : Barry Hawkins

Signature / Cachet :

Directeur de mémoire en entreprise :

Nom : John Jones

Signature :

Invité :

Nom :

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(s) du candidat(s)

SME Conversion modelling – a machine learning approach

Résumé

Mot clefs: model de transformation, optimisation tarifaire, apprentissage automatique

L'optimisation tarifaire est nécessaire pour maintenir la profitabilité d'un portefeuille d'assurance dans un environnement concurrentiel. Cela nécessite des modèles complexes qui deviennent désuets dès que la concurrence ajuste ses prix. En conséquence, la charge de travail requis pour maintenir ces modèles à jour ne peut être soutenable que pour un grand portefeuille d'assurance à la personne.

Pour mettre en place une optimisation tarifaire sur un portefeuille plus petit, il est essentiel d'automatiser le plus possible les éléments de cette optimisation. Deux activités résistent aux efforts d'automatisation : la construction et la calibration des modèles de transformation de devis. Il est possible de déléguer ces tâches à un ordinateur avec un algorithme d'apprentissage automatique. Bien qu'il existe une large gamme d'algorithmes, seulement deux ont été testé pour ce prototype : *Gradient Boosting* et *Random Forest*. Ces deux algorithmes ont été choisis pour leur simplicité à implémenter en R et pour leur différence avec le modèle linéaire généralisé qui est la pratique courante actuelle. Ils permettent de tester l'hypothèse de linéarité dans les modèles de transformation de devis.

Le pouvoir prédictif de chaque méthode a été comparé au modèle linéaire généralisé sur deux produits différents pour tirer des enseignements en matière d'implémentation et d'efficacité. Les deux modèles ont obtenu de meilleurs résultats que le modèle actuel avec un temps de production beaucoup plus court.

Abstract

Key words: conversion modelling, price optimisation, machine learning

Price optimisation is a key component to maintain a profitable portfolio in a competitive environment. This requires complex models which are only valid until the competition adjusts its prices again. As a result, the workload required to maintain these models in a fit state can only be supported by large portfolios in the mass market of personal insurance.

To implement price optimisation on a smaller portfolio, it is important to automate as much of the process as possible. Two key areas that have resisted any significant automation are the model construction and calibration. It is possible to delegate these tasks to a computer by letting the machine learn from previous data. Although there is a wide range of machine learning algorithms, only two were tested for this proof of concept: Gradient Boosting Machine and Random Forest. These two algorithms were chosen for their simplicity of implementation (in R) and their marked difference with linear models which represent the current industry practice. This challenge of the linearity assumption is an additional benefit proposed by some machine learning algorithms.

The performance of each method has been assessed against generalised linear models on two different products giving some insight on best practice for implementation and showing competitive performance for a much shorter development time.

Table of contents

Note de synthèse

Une approche rigoureuse pour déterminer la prime commerciale est essentielle sur un marché concurrentiel. Les modèles de transformation de devis en police d'assurance sont l'un des outils clefs des équipes pour déterminer cette prime commerciale. Cependant, leur développement et maintenance demandent beaucoup de ressources qui ne sont pas disponibles sur des marchés de taille réduite. Les techniques d'apprentissage automatique peuvent potentiellement offrir une solution moins onéreuse pour aider à déterminer la prime commerciale. Apres une mise en contexte du marché de l'assurance au Royaume-Uni et une introduction à l'optimisation tarifaire, trois exemples d'application d'algorithme d'apprentissage automatique sont présentés avec les leçons qui en ont été tirées.

Le marché de l'assurance aux petites et moyennes entreprises (PME) en Royaume-Uni commence à suivre les mêmes évolutions que celles observées sur le marché de l'assurance à la personne. Sur ce dernier, le développement des sites de comparaison a dynamisé le secteur du courtage en focalisant l'attention des consommateurs sur le prix et, par conséquence, les commissions perçues. L'évolution nécessaire des courtiers pour survivre à ces profondes modifications a pris deux formes : l'optimisation et automatisation de leur processus de vente pour réduire les coûts de fonctionnement et la refonte de leur business model avec des sources de revenue autres que les commissions.

Bien que plus lente à se développer dans le secteur commercial, cette automatisation grandissante change la donne pour les assureurs aux entreprises : le marché devient plus compétitif et transparent avec les courtiers demandant des devis à un plus grand nombre d'assureurs. Avec l'optimisation tarifaire organisée par les courtiers, le devis le moins cher remporte le risque tandis que le courtier empoche ou débourse la différence entre la prime d'assurance et le prix offert au client. Le portefeuille de l'assureur est donc défini par les risques les pour lequel il est le moins cher. Jusqu'à présent, beaucoup d'assureurs commerciaux géraient leur portefeuille de risques avec une stratégie tarifaire simple au niveau du portefeuille et peu de différentiation des primes entre risques. Cette stratégie permettait de fournir un prix raisonnable pour tout risque et ainsi soutenir ses partenaires courtiers en acceptant la majorité des risques. Les portefeuilles de risques comportaient d'importantes subventions internes pour contre balancer des segments non rentables. Cette approche permettait aux courtiers de placer tous leurs risques avec une poignée d'assureurs partenaires aux critères d'acceptation larges plutôt que de dépendre d'une longue liste d'assureurs pour placer les risques inhabituels. Dans un environnement plus concurrentiel, cette gestion de portefeuille expose l'assureur à une anti-sélection. Il est donc important d'adapter la tarification et sélection des risques au fur et à mesure que le marché migre vers un modèle plus compétitif.

Avec l'automatisation des devis et transactions, la capture d'information s'améliore. Le commerce électronique de polices d'assurance commerciales s'ouvre au-delà de la poignée de courtiers qui avaient développé leur propre système et dévient disponible pour les assureurs et les petits courtiers grâce à l'apparition de plateformes électroniques indépendantes (*software house*). Les données de devis deviennent accessibles en grande quantité offrant aux assureurs une vue du marché plus

complète. Cette amélioration rapproche le marché de l'assurance aux PMEs de celui de l'assurance à la personne.

Avec ces nouvelles informations sur la composition du marché et de la part de l'assureur sur chacun des segments, il est possible de raffiner la structure tarifaire pour améliorer la profitabilité du portefeuille. La théorie de gestion de portefeuille d'actif financier peut aussi être appliquée à un portefeuille de risques en considérant les deux mesures qui étaient le risque et le retour sur investissement pour un portefeuille financier équivalentes à la taille du portefeuille et le profit attendu. De la même manière qu'il existe un portefeuille financier optimal pour un niveau de risque donné, il existe un portefeuille d'assurance optimal pour un volume donné. Pour un actif d'assurance, le niveau de la prime d'assurance défini le niveau de profitabilité (l'attrait de l'actif) ainsi que le volume d'actif détenu (part de marché possédé). Pour cette modélisation, la prime pure (cout des sinistres) est supposée connue pour chaque segment du marché. Dans la pratique, cette donnée est estimée et peut représenter un risque. Il est possible d'ajouter une prime d'incertitude à la prime pure pour refléter le risque supplémentaire et éviter qu'une résolution mathématique n'aboutisse à un portefeuille donc la profitabilité est douteuse.

La résolution des équations pour identifier un portefeuille d'assurance optimal fait entrer en jeu la sensitivité du client au prix du contrat d'assurance. Ce paramètre permet de comparer les deux effets du prix contraire présentés précédemment: une augmentation du prix améliorant la marge par police d'assurance mais réduisant le nombre de polices vendues. Son estimation est donc l'élément clef de l'optimisation tarifaire. Au-delà de l'optimisation théorique, l'implémentation présente quelques défis : la sensitivité au prix peut être estimé aux alentours du prix actuel mais est complètement inconnue pour des variations importantes de prix. Aussi, les résultats obtenus ne sont valides qu'aux alentours du point prix actuel. Une méthode d'optimisation locale permet de rester dans la zone connue. La méthode suggérée est une méthode de gradient permettant de d'améliorer le portefeuille par petites itérations. Il est important de noter qu'une optimisation de proche en proche avec contrainte ne garantit qu'un maximum local et non absolu. Sans contrainte de volume ou de tarification (en supposant que chaque prix pour chaque profil de risque peut être fixé indépendamment), la contribution de chaque risque est une fonction convexe donc il existe un maximum absolu. Néanmoins, une contrainte sur le volume réduit l'espace de prix possibles et peut introduire des effets de bords. De même, une dépendance entre les prix (par le biais d'une structure tarifaire multiplicative par exemple) complexifie la relation entre variables et profit. Ces deux phénomènes peuvent nuire à la convexité de la fonction dont on recherche le maximum et créer des maxima locaux. Par conséquent, le résultat de l'optimisation locale mathématique ne peut remplacer une revue complète du positionnement stratégique du portefeuille. Cela étant dit, l'optimisation est un outil précieux pour améliorer un portefeuille en identifiant des opportunités à la marge du portefeuille existant. Ces résultats sont néanmoins sujets à une estimation correcte de la sensitivité du client au prix.

Sans hypothèse supplémentaire, mesurer l'élasticité tarifaire nécessite un grand nombre de devis. En effet, l'élasticité est le ratio entre deux taux de transformation de devis qu'il faut estimer avec suffisamment de précision pour que le ratio soit stable. Ceci requière un grand nombre de devis identiques (de l'ordre de 10^5) qu'il est impossible d'obtenir. Il est alors nécessaire d'autoriser des variations entre devis dont l'effet est expliqué par un modèle. La méthode d'estimation de l'élasticité

tarifaire répandue au Royaume-Uni est sur la base d'un modèle linéaire généralisé utilisant pour lien une fonction logistique et une structure d'erreur suivant une distribution de Bernoulli. Cette méthode suppose une indépendance d'effet entre les différentes variables explicatives et une contribution linéaire (après transformation par la fonction logistique). De plus, re-construire ce modèle sur de nouvelles données est un travail important et manuel où chaque variable doit être à nouveau testée. En effet, le pouvoir de prédiction d'une variable dépend d'une différence de tarification entre l'assureur et le reste du marché. Si l'un des deux change sa structure, une variable peut passer de prédictive à neutre et inversement. De plus, la validité de ces modèles ne dure que tant que les prix restent similaires. Avec un changement de prix par mois, ces modèles doivent être entièrement revus tous les deux ou trois mois. Ceci génère une masse de travail qui requière une équipe importante. Dû à la taille du marché de l'assurance pour les PMEs, de telles dépenses ne sont pas soutenables.

Les algorithmes d'apprentissage automatique fournissent une alternative pour réduire cette charge de travail et d'aider à établir une prime commerciale à moindre coût. Plusieurs algorithmes ont été considérés sans être retenus. Le premier groupe est celui de régressions pénalisées. Ces algorithmes répliquent le même modèle que celui utilisé par la majorité de l'industrie mais ajoute un processus automatique de sélection des variables explicatives. Cette méthode est la plus directe d'automatisation et demande un investissement important dans la construction de l'algorithme de sélection de variable : il faut d'abord reconstruire le modèle actuel (ou équivalent) dans un langage de programmation capable de supporter un module de régression pénalisée. Pour que la sélection automatique soit capable de recréer des résultats en ligne avec un opérateur humain, la pénalisation doit être choisie de manière précise pour refléter le critère humain. Comme d'autres algorithmes sont disponibles dans le domaine public déjà prêt à l'emploi, les régressions pénalisées ont été jugées trop lourdes pour une implémentation rapide. Les autres classes d'algorithmes considérés comprennent les réseaux neuronaux et les machines à vecteurs de support. Néanmoins, ces deux classes ont été abandonnées dû à leur aspect boite noire. Au final, deux algorithmes ont été testés en termes de temps de calcul et de performance. Les algorithmes choisis sont le *Gradient Boosting Machine* (GBM) et la *Random Forest*. Ces modèles bénéficient d'une implémentation simple dans le langage de programmation R et sont issue d'arbres de classification qui ne sont pas soumis à la contrainte linéaire des modèles linéaires généralisés.

Ces deux modèles sont testés contre un modèle linéaire généralisé sur un premier produit de responsabilité civile. Les deux modèles sont calibrés sur des données d'apprentissage puis valider contre un échantillon gardé à part. Les résultats montrent que les deux algorithmes sont plus prédictifs que le modèle standard (critère de courbe ROC). Le GBM est facile à calibrer avec des critères automatiques de terminaison d'apprentissage et obtient la meilleure performance. La *Random Forest* obtient aussi de meilleurs résultats que le modèle linéaire généralisé mais a des difficultés à gérer un grand nombre de variables sans signal ce qui crée une distorsion de la prédiction. Celle-ci nécessite d'être re-calibrée avant d'être utilisable. De plus, l'algorithme est plus lent que le GBM. Ceci peut être compensé par une parallélisassion des calculs plus difficile à mettre en place. Certaines versions de GBM peuvent aussi être parallélisées (XGBoost) mais le volume de données utilisées n'a pas nécessité cette implémentation.

Le GBM a aussi été testé contre un modèle linéaire généralisé sur un autre produit d'assurance propriétaire loueur. Une fois encore, le GBM surpasse le modèle standard. Cependant, cette analyse a montré qu'une segmentation capable d'améliorer le modèle linéaire généralisé peut aussi améliorer le GBM. Ainsi, les algorithmes d'apprentissage automatique ont toujours besoin d'un travail en amont de segmentation et de création de variables explicatives recombinées. Ils peuvent être reconstruits rapidement et indépendamment mais une analyse manuelle reste nécessaire pour obtenir les meilleurs résultats possibles.

Le GBM est un bon candidat pour fournir rapidement un modèle prédictif mais manque de transparence pour une analyse détaillée des causes. Certains outils d'analyse d'arbres tels que l'importance et les effets marginaux pallient partiellement à ce manque de visibilité mais nécessitent plus de développement avant d'être concluant. La première mise en application du modèle GBM a été de valider un modèle linéaire généralisé et ainsi trouver des pistes d'amélioration. En second plan, il a été utilisé comme outil d'aide à la décision en modélisant l'impact d'un changement de structure tarifaire. Enfin, la troisième utilisation fut pour normaliser différents devis au niveau de leur taux de transformation pour une analyse d'élasticité tarifaire. La compréhension du modèle n'est pas encore suffisante pour être directement utilisée dans une optimisation mathématique. Néanmoins, à plus long terme, l'amélioration de la prédiction par rapport au modèle standard peut être un avantage concurrentiel précieux lorsque la compétition s'intensifie.

Executive summary

Pitching an insurance policy at the right price compared to the market is essential in a competitive marketplace. Conversion models are one of the main tools used by market pricing teams. However, their development and maintenance require extensive resources which are not available beyond the large markets. Machine learning algorithms can potentially provide a cheaper solution to support market pricing teams. After providing context to the trends seen on the Commercial Lines market in the UK and a brief introduction to price optimisation, three practical examples of machine learning are presented with their lesson learnt.

The insurance market targeted at Small and Medium Enterprise (SME) in the United Kingdom (UK) shows signs of following the same path as the personal lines insurance. On that segment, the development of price comparison websites has drastically changed how insurance policies are sold, focusing the customer on price, and therefore, on the commission added by intermediaries. The brokers' response to this market disruption was two folds: first, to automate and optimise the sale process to reduce costs and second, to change their business model with alternative stream of income.

Although slower to pick up in commercial lines, this growing automation is challenging commercial insurers with increased competition: brokers are requesting more insurers to quote to drive prices down. With the commission optimisation done by the brokers, the cheapest quote wins the risk while the broker take (or pay) the difference between what was offered to the customer and the insurer's price. The insurer's portfolio is therefore defined as the risks for which it was the cheapest. Up to then, most insurers managed their commercial book as a portfolio with simple rating structures and little price differentiation between risks. This strategy was providing reasonable prices for all risks with broad acceptance in order to support brokers who could face occasionally an unusual risk to place. The portfolio therefore included significant cross-subsidies to compensate for loss making risks. This approach was enabling brokers to place all their risks with a handful of non-selective insurers rather than having to rely on a long list of insurers place atypical risks. In a competitive market, this strategy exposes the insurer to anti-selection. As a result, it is crucial to adapt the pricing strategy for commercial risks as the market becomes more competitive.

With the automation of quote and transactions, the quality and quantity of data available increased. Electronic trading of commercial insurance spread from the handful of early adopters (mainly large brokers on their own systems) to most brokers thanks to software houses building bridges between brokers and insurers. Quote data became widely accessible to insurers who finally have a sight of the full market.

Using the newly available market information and what is the current share of an insurer, it is possible to tweak the pricing strategy to improve profitability. Portfolio management theory, originally designed for financial assets, can also be applied to insurance portfolio by replacing return and risk by expected profit and portfolio size. As there is an optimal financial portfolio for a given return, there is an optimal insurance portfolio given a volume of written premium. For an insurance asset, the premium defines both its profitability (attractiveness) and the volume owned (market share). For this modelling, the burning cost is considered known for all rating cells (identical risks). In practice, this is estimated and can present some uncertainties. It is possible to add a risk premium to

the burning cost to reflect the uncertainty (and therefore risk) and avoid that a mathematical optimisation ends up with a portfolio of uncertain profitability.

The solution to the optimisation equations brings in the price elasticity of the customer. This parameter allows quantifying the two conflicting effects presented earlier: a price increase which adds to the profitability of each policy but reduces the number of policy sold. Estimating the price elasticity is the corner stone of price optimisation. Beyond the theoretical optimisation, the implementation presents some difficulties: the price elasticity can be assessed around the current price point but is totally unknown for prices far from the current offering. Therefore, the results of optimisation are only valid locally, within a range of prices and the optimisation needs to be performed within this space. The suggested method is a gradient descent improving the portfolio by small increments. It is important to note that an optimisation by small steps and under constraints can only certify of a local maximum, not necessarily a global maximum. Without any constraints, both on volume and pricing structure (supposing that the price for each rating cell can be set independently), the contribution of each risk is a convex function of the price and therefore there is a global maximum. Nevertheless, a volume constraint restricts the space of acceptable prices and can introduce some border effect. Similarly, a dependency between prices (by the intermediary of a multiplicative rating structure for example) forces the relation between parameters and profit to be more complex. These two elements mean that the function and the space on which the maximisation is performed may not be regular enough to ensure the unicity of a maximum. As a result, a mathematical optimisation cannot replace a strategic review of the portfolio positioning. Having said that, the local optimisation is a useful tool to identify marginal opportunities close to the current portfolio. All these results are nevertheless subject to a correct estimation of the price elasticity.

Without additional assumptions, measuring the customers' price sensitivity requires a large number of identical quotes. The price elasticity is defined as the ratio between two conversion rates that need to be estimated with enough precision for the ratio to be stable. Simulations put the number of identical quotes in the order of tens of thousands which is impossible to gather. It is therefore mandatory to allow some variations between quotes with the effect of these variations explained by a model. Current market practice in the UK to estimate the price elasticity is via a Generalised Linear Model (GLM) with a logistic link function and a binomial error structure. This method implies an independence of effect between explanatory variables and a linear contribution (subject to the transformation by the link function). In addition, re-calibrating this model on new data requires a lot of manual work as each variable need to be re-tested. Indeed, the predictive power of each variable depends on the difference in rates between the insurer and the rest of the market. If either moves, a variable would lose or gain predictive power. In addition, the validity of such model lasts only as long as prices stay stable. With a price change each month, these models need to be completely reviewed every two or three months. This generates a workload requiring significant staff resources. Due to the limited market size of any one product in commercial lines, such costs are not sustainable.

Machine-learning algorithms provide an alternative to reduce workload and enable a low-cost market pricing. A few algorithms have been considered without being implemented. The first group includes penalised regressions. These algorithms replicate the industry-standard model but add an automatic process of variable selection. This is the most natural automatization method targeting the part of the process that used to be manual. However, it requires some significant upfront

investment: the ability to recreate the current model (or similar) in a programming language that can support a module to perform the penalised regression on top of it. For the automated selection to be able to compete with a human operator, the penalisation function needs to be chosen carefully to reflect the human criteria. Compared to other algorithms available off the shelf, the penalised regression has been judged too heavy for this proof of concept. The other groups of algorithm considered were neural networks and support vector machine. However, both groups have been dropped due to their black box behaviour. In the end, two algorithms have been tested for performance and speed: Gradient Boosting Machine (GBM) and Random Forest. These algorithms benefit from a simple implementation in R and are based on tree classifiers which are not subject to the linear condition present in GLMs.

These two algorithms have first been tested against a GLM on a public liability product. Both models are calibrated on a training sample and tested on a validation sample. The results show that both models out-perform the GLM (based on a ROC curve). The GBM is easy to calibrate with an automatic criteria to stop the training and has the best performance. The Random Forest also out-performs the GLM but has difficulties handling large number of variables without signal resulting in a distorted prediction. This, however, can be corrected via an isotonic regression. Nevertheless, the algorithm is slower than the GBM, unless the calculation is parallelised which more difficult to implement. Some implementation of GBM can also be parallelised (XGBoost) but the volume of data in this study did not require the added complexity.

The GBM has also been tested against a GLM on a second insurance product: a landlord insurance. Once again, the GBM is more predictive than the standard model. However, this analysis has shown that a segmentation which was able to improve the GLM was also able to improve the GBM. Thus, machine-learning algorithms still requires work of segmentation and feature engineering ahead of the modelling. Despite being calibrated on new data really quickly, an in-depth analysis is still required regularly to ensure the best possible results.

The GBM is an excellent candidate to provide quickly a predictive model but is lacking the transparency needed for a detailed analysis of causation. Some tree-based methods can provide a partial insight, such as variable importance or marginal effects, but need further development. The first implementation of the model was to validate the build of a GLM with little added work and find some leads for improvement. Secondly, the model has been used to support pricing decision by modelling the impact of a price change. Finally, the third use was to normalise quotes for their conversion ahead of an elasticity analysis. A better understanding of the model and its results is still needed before using it directly in an automated optimisation program. On a longer term, the improvement in predictive power compared to the industry standard can provide a welcome edge as the market become more competitive.

General Introduction

This paper is looking into improving conversion modelling for auto-rated products targeted at Small and Medium Enterprises (SME) companies in the UK market. The problematic is to understand what machine learning algorithms can bring to this analysis with an emphasis on workload for setting up and maintaining such models as well as performance. The underlying challenge is to provide good-quality market pricing tools responding to the Commercial Lines emerging requirements.

The first part is looking at a brief history of the SME market in parallel to the Personal Lines market. This is to help understand the recent trends on the Commercial Lines and the radical transformation of the market place that commercial lines insurance is going through.

As the SME market moves closer to the Personal Lines model with an intense competition on price, the theoretical basis for price optimisation will be covered in the second part of this paper. This introduces the theory behind portfolio management, optimisation and highlight the key concept of price elasticity.

The third part will focus on how to measure conversion and elasticity and the necessity of a model to do so. The market-standard model will be described with the level of resourcing required to support such a model. Then, two additional models, both machine learning algorithms, will be introduced with some of the theoretical aspects of these models described to support the understanding of the results from the two case studies that form the part 4 and 5 of this paper where the three models are put in competition.

I) Background and trends on the UK SME market

Evolution in personal lines: standardisation and rise of the aggregators.

The personal motor market in the UK was historically controlled by brokers. In 1991, Direct Line revolutionized the sector by cutting the middle man and dealing directly with the customer. The interaction with the customer moved from face-to-face to over the phone. This virtualisation continued with the increased access to internet. The first online aggregator was launched in 2002. With the shift of customer behaviours toward online shopping, the aggregators were able to impose a standardisation of the insurers' offering. The commoditisation of the products led to an increased price competition. By 2008, more than 50% of motor and home policies were bough online (either via aggregators or direct website). With most listing on aggregator sites ordered by price, the competition intensified on this single attribute. As the aggregator industry heavily relied on their own branding to attract customers to their website, these new players engaged in marketing wars that lead to a significant concentration in the sector. Four aggregators control the majority of the market with three of them being owned by a large insurance group: Confused (Admiral), Compare the market (BGL Group), Go Compare (Esure) and Money Supermarket (public company). A fifth player is worth mentioning due to its backing by Google (Beat That Quote) although it remains a smaller player at the moment.

Brokers have adapted to the rise of the aggregators. They are offering their products directly on the aggregators under their own branding but supported by a panel of insurer. To be competitive, brokers negotiate preferential rates with insurers thanks to their size and flex or wave their commission. As the pressure on price increased, brokers have diversified their stream of income: the reduced upfront commission is supported by other incomes. Brokers have become specialists at extracting value from the relationship with the client. It started with the opportunity of up-selling as soon as the customer left the aggregator website and arrived on the broker website. As the competition on the aggregator is on the main cover, the add-ons were under less scrutiny and brought some significant profits. The add-ons and services provided as such included premium finance (monthly payment instead of annual payment: charged with significant interest rates), roadside covers, legal protection services or uninsured loss recovery. The second opportunity was for the broker to cross-sell other insurance products. Beyond the simple sales, the administration of the policy was also an opportunity for additional income. Mid-term adjustments or cancellation could be subject to a fee. However, the largest source of income was linked to claim notification: lawyers and claim management companies were paying referral fees to get access to individuals involved in accident. The compensation process for victims of Road Traffic Accident (RTA) was generating large legal fees for the lawyers. As these were ultimately paid by the insurers and pushed the price of motor insurance to unsustainable levels, the referral fees were banned by the government in 2014. Although reduced, this stream of income still exists in two different forms. The replacement car hire can be done on credit (so the victim sends the bill directly to the at-fault insurer) with credit terms being inflated. More recently, a court judgement has approved the practice done by one insurer (RSA) of charging normal reparation rates to at-fault insurer while they only paid preferential rates to their garage network. The difference is kept as profit by the insurer. Therefore, sending any accident victim to a garage with a partnership can bring significant income. For brokers, all claims could be a source of income if they have their own garage agreement. In aggregate, the possibility of additional incomes for the broker pushes commission to negative territory on the most competitive segments (sales via an aggregator). As a consequence and thanks to their ability to extract value ignored by insurers, brokers are able to compete on price in an aggregator world and derive a large part of their business from these platforms.

For the broker to play on an electronic market, they needed to be able to get the price of various insurers automatically and with a single query. The biggest brokers had no difficulty to convince insurers to join their panel and build their pricing algorithm directly on the broker system in exchange of the possibility to access their large customer base. However, smaller brokers could not justify the development and maintenance cost for their small volume of premium. Software house companies developed as a hub between brokers and insurers with a single development on the software house giving access to all brokers with a contract with the given company. A market standard for communications and rating engines was developed by the industry and standard question sets (iMarket) introduced between software houses to help compatibility between various providers (and therefore increase competition between software house companies where size benefits would have forced a concentration of the players to the expense of the end customer). As pricing sophistication continued, insurers started to bring back in-house their pricing algorithm deployed on the broker systems, replacing it with a live connexion between the insurer and the broker system. In addition to perform a pricing algorithm as complex as the insurer wishes,

Insurance-Hosted Pricing (IHP) gives the insurer complete control of the rates (and when they are deployed) and access to quote data.

From Personal Lines to other segments: roll out of mass-market techniques to commercial motor

The share of aggregators started to plateau around 60% of the market in 2012. To pursue further growth, the aggregators moved to additional markets. Some extensions were outside the insurance industry (saving accounts, credit cards, energy provider) while other aggregators focused on the commercial world with company vehicles and micro-SME insurance. However, outside motor insurance, commercial customers have not shown yet the same change of shopping behaviour and online trading is still a niche market, albeit growing.

With the arrival of aggregators on the Commercial Vehicle market, a full view of the market prices became easily available without the intervention of a broker. This supported the development of the direct channel and pricing structures are getting more sophisticated to respond to the increased price competition. As a result, commercial lines underwriters are looking to replicate some of the innovations found in personal lines. The low conversion environment with large volume of data and a commoditised product bring a market highly driven by price which requires significant monitoring: volume can drop or skyrocket with a simple deviation from the market price. The capacity to change rates at a short notice becomes primordial in controlling the mix of business written and therefore the performance. From a necessity to avoid adverse risk selection, this capability has been increasingly looked as an opportunity for price optimisation. Predictive impact tools and price elasticity are being developed to support the commercial vehicle motor market.

From a simple optimisation of sales volume, personal lines and commercial lines have moved to a view of profit of multiple years. The online presence and the intermediation of the aggregators have brought back some explicit acquisition costs. To maintain or promote an online presence, advertising on a per-click basis is usual to drive traffic to the insurer website. Although each click is cheap, the low number of sales compared to the number of visit on the website makes it a significant cost per policy. On the other side, the business model of aggregators is based on a fee paid for each redirection to the insurer website leading to a sale. This cost is fixed and can be significant compared to low-risk policy premium. The number of years the insured will stay on the newly purchased contract has a significant influence on the amortisation of the acquisition cost and therefore the planning of marketing spend to attract traffic. Renewal model and life-time value of a policy are developed to optimise the new business pricing strategy and the one of the subsequent renewals.

From commercial motor to non-motor

With electronic trading having entered the commercial lines world via Good Carrying Vehicles (GCV), commercial brokers have realised that it is an efficient way of dealing quickly with simple cases. As brokers are paid by commission in proportion to the insurance premium, the smaller cases can only justify a light work from the broker. In personal lines and commercial motor, brokers with a heavy reliance on commission (by opposition to other sources of income) have been penalized by the rise of aggregators. At the same time, the costs of regulation and compliance have increased following actions by the Financial Conduct Authority (FCA). The FCA was responding to a series of failures in the industry: client money not segregated, mis-selling of add-ons, failure to act in the client best

interest... The sector has therefore gone through a significant restructuring and consolidation to tackle the sophistication of the source of income and an improved value for money.

Part of this is delivered via cost reduction which leads brokers to push for solutions to trade faster and easier, especially on the smaller cases which would not be profitable otherwise. Electronic trading is a natural solution, having shown its success in motor insurance. After entering the risk details on the system, the latest price is directly available and the sale is only a few clicks away. This is a lot easier than sending requests to various offices and waiting for manual quotes to come back (which can take a few working days) or having to rely on published rates (which requires calculation and are disliked by insurers for their lack of control). With a direct and reliable response, the sale can be agreed in the same interaction with the client, saving any cost or risk involved with a follow up meeting to conclude the sale.

Electronic trading was set up by some insurers in the mid-2000s. Specific products moved to auto-rated and made available directly to the broker. The preferred approach was an extranet website: each insurer has its own site available requiring manual inputs to get a price instantaneously. Although with a good response time, it requires individually populating each site and therefore limits the number of quotes that can be generated while the client is seating in the broker office. Some brokers have developed automatic population between their own system and the extranet websites to get multiple quotes at the same time. However, these developments were marginal as they required significant work and investment with no standardisation of the extranet sites between insurers. As before, larger broker pushed insurers to build their product on a broker system to facilitate integration. However, on commercial lines, only a handful of brokers and products had enough volume to provide a return on investment that could attract insurers.

Establishment of software houses in commercial lines non-motor

Part of the success of the personal lines came from the integration of electronic trading to the broker system as it was saving the broker a significant amount of time. As in personal lines, this development had to be grouped to share the cost for the smaller portfolios. Some software houses have identified this as an opportunity to grow and have extended their offering to commercial products. The market is building momentum with more insurers joining software houses, bringing more interest from brokers to join. However, the licencing costs charged by software houses can be prohibitive for small brokers and therefore extranet websites are still a good option for small scale brokers.

Historically, extranet systems were hosted by the insurers themselves, meaning they have access to quote data. This was extremely useful information on a market ridged with cross-subsidies. However, the data harvested on extranet sites was only showing a specific segment of market: the one identified by the broker as the target market of the insurer. If a broker identified a risk as not part of the footprint of a given insurer, it would not waste time processing the quote on that insurer extranet site. The insurer is therefore blind to some segments of the market. On a broker panel, the insurer would be automatically exposed to the full market as all risks seen by the broker would go through. Unfortunately, panels hosted by brokers are notorious for not sharing quote data (brokers are wary of price optimisation and prefer full blind competition to push the prices downward). However, software houses are less reticent to share quote data as they do not have any vested

interest in keeping the premium low. Instead, they identified this data as an additional service and income stream. In this way, they are more alike to a third party provider of IHP services (Insurer-Hosted Pricing).

Software houses provide more than a pricing platform for insurers: they are offering a full back-end solution to brokers. The back-end system is used by many brokers to capture the declaration from customers, compile communications and quotes received from insurers, administer the policy, bill customers and transfer the money to the insurer. On commercial risks, the data capture being done on a system is a good first step toward more automated pricing, which would provide opportunities for software houses to get more work from insurers.

From the insurer point of view, the manual process of underwriting small commercial risk requires a significant workforce and organisation leading to high expense ratio. In the current context of soft market (difficulty to carry high rates), there is a real emphasis on reducing the expense ratio to be able to follow the market in a price war without reducing profitability. To do so, the focus is on improving productivity of the underwriting team via automation of their work and re-directing the resources toward larger and more complex risks. Auto-rated products are expensive to set up but there is little influence of the volume written on the running cost. For an insurer looking to grow, it is a great opportunity to tackle larger volumes with identical resources. Therefore, there is the same drive on the insurer side to achieve higher automation of the underwriting for the smaller end of the commercial risks. This is likely to occur through a gradual increase of the underwriting footprint of auto-rated products.

As both insurers and software houses are willing to get more automation, this trend can build up momentum. However, the real driver of change would be the brokers and how they would like to do business. Auto-rated products tend to be inflexible with hard rules to cater for 90% of the risks and ignore the 10% remaining (one size fits all approach). This makes sense for a volume player but would alienate a local broker looking to place a risk in the 10% not suitable. The relation with a broker is a commercial one: both sides are looking to reach an agreement which is mutually beneficial. This requires compromises and flexibility that an algorithm struggles to implement effectively. As the proportion of not suitable risks increases, manual work is needed to cover these to keep a good relationship with brokers.

Moreover, the heterogeneity of the risks increases with their size. A tradesman working on his own can summarize fairly accurately his activity with one or two trades. The difference in risk between two tradesmen answering identically the same question set is limited. For a more complex risk with £10m turnover, 20 permanent employees and range of subcontractors, 2 or 3 activities are unlikely to provide the full picture. From a 90% that can be done automatically on small risks, the proportion drops drastically with more diverse risks. The situation to avoid is the one where the algorithm accept automatically a risk that a manual underwriter would have considered below average. Auto-rated products tend to focus on what is described as “clean risks”: risks without any marker of potentially worse performance. This could be poor claim history, unusual activity or inconsistency in the declaration of facts. As the quantity of information available on a risk grows, so is the risk of a warning flag to be raised and the risk to drop out. Current implementations of auto-rated product targeted to larger risks aim at 60% automatically quoted. Insurers need to set up a solution for the 40% dropping out to keep brokers on board of electronic trading.

Bearing this in mind, a realistic implementation has to be a mix of auto-rated and manual underwriting if we want to attract larger risks to e-trading. These products are being deployed to the market but the up-take from the brokering community is low: current practice of receiving a tailor-made manual quote is still engrained in the business. As for online purchase of commercial insurance, the expected shift in customer behaviour has not happened yet. This is potentially due to the quality of the offering which is currently being improved or it could be the customer/broker wish to know that the quote was designed for his specific risk and not a generic price. More personal products require more information that is only relevant to a few specific cases. A revolution in question sets is the next step with more free-form text and less mandatory questions. Brokers are already producing packs to send to insurers for quotes. These packs contains all information deemed relevant by the broker and should be considered as the inputs for a quote rather than a rigid question set. Extracting information out of these packs can be automated to a large extend, saving significant amount of time for the underwriter.

Consequences for SME pricing

The competition in price on SME market has significantly increased in the new e-traded environment that is appearing for the micro-SME. Profit margins have reduced and the products, which used to include significant cross-subsidies, are now exposed to anti-selection. The market is transforming from portfolio pricing to specific risk pricing but this transformation is slow. The technical price is an indicator of where the market will eventually settle but rushing to charge this price would give away premium unnecessarily: a position between the market premium and the true price would ensure a better risk selection than the competitors while still charging the maximum premium for the risk. A good portfolio management approach would utilise the discrepancies between risk pricing (what it costs to insure) and market pricing (what the market charges) to come up with an optimised portfolio.

The focus is now to fully understand these opportunities and how to build a successful pricing strategy on these markets. The quantity of information available has increased recently: quotes data are being shared by software houses and information related to the risk is becoming more and more detailed and consistent between brokers. These are the building blocks for conversion analysis and targeted pricing decision. The micro-SME market is ready to follow the Personal Lines market in the detailed analysis of market pricing and optimised price decision.

II) Portfolio management theory

Portfolio management: efficient frontier

The theory comes from asset portfolio management. Every asset/portfolio has an expected return and a risk attached to it. Between two portfolios of same risk, an investor will always prefer the one with the highest return. Similarly, between two portfolios of same return, an investor will always prefer the one with lower risk.

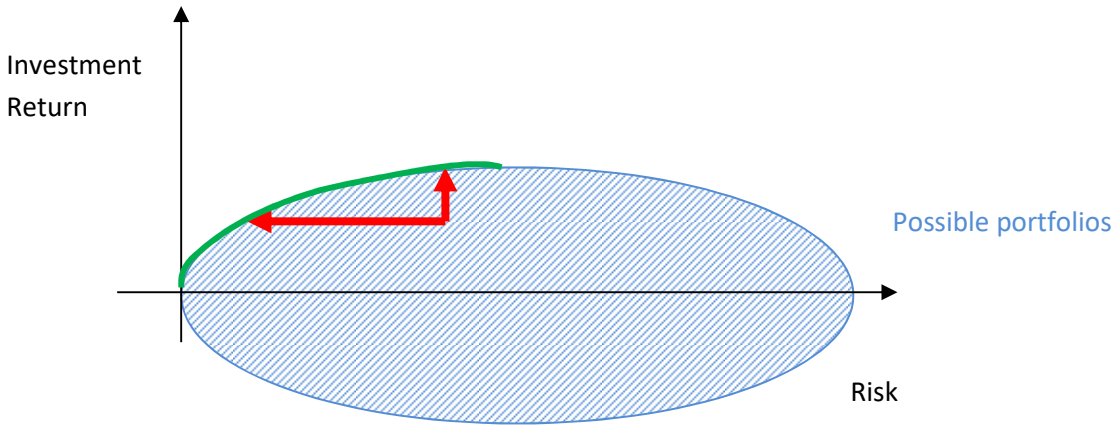


Figure 1 – Financial assets portfolio and efficient frontier

The current portfolio (at the start of the red arrows in Figure 1) can be improved as the investor has the choice among a range of portfolios with a higher or equal return than the current one for a risk lower or equal than the current level. All choices are strictly better than the original portfolio. However, the preference within the range of improved portfolios depends on the risk profile of the investor and his trade-off between risk and return. A portfolio that cannot be improved further is called optimal.

The efficient frontier is the set of portfolios that are optimal. The overall problem can be considered as a double optimisation: one on maximal return, the other one on minimum risk. On the figure 1 above, the maximised return is on the upper border of the possible portfolios. The minimised risk is verified on the left border of the possible portfolios. Therefore, the efficient border is the upper left quarter of the border (in green on the graph).

Case for insurance portfolio

In the case of insurance portfolio, the trade-off will be between volume and profit. The first part of the work will be to maximise the profit for a given volume. For the insurance portfolio, it is difficult to judge whether the volume should be minimised (lower exposure and lower capital requirement) or maximised (fixed expense dilution and higher model reliability).

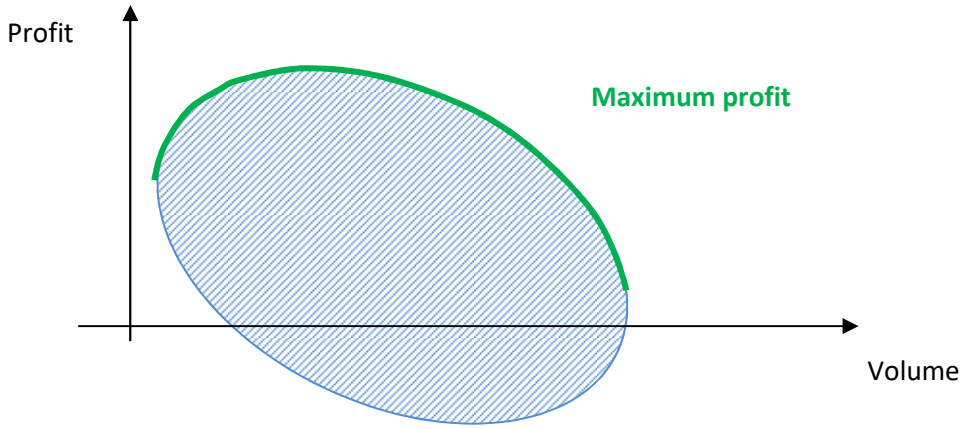


Figure 2 – Insurance portfolio

As the commercial insurance market has become more homogenous in term of offering on the SME segment, prices have become a major factor influencing the size of an insurance book. The current economic environment has put an even stronger emphasis on the pricing. The following work looks at the short term optimisation focused on price and considers all other elements constant. Other influences on the size of an insurance book include (not exhaustively) company brand, quality of product, sales channel and competitors' activity. All these elements have an influence on the volume written but take longer to change. Over the short-term view of the optimisation, these are considered fixed.

Mathematic formulation

$$\underset{V = V_0}{max} \sum_i (p_i - BC_i) n_i(p_i) \quad (1)$$

$$\text{With } V = \sum_i p_i * n_i(p_i) \quad (2)$$

Definition: V is the total volume of premium. V_0 is the planned volume of premium.
The index i defines a rating cell (group of identical risk) with its price p_i, its burning cost BC_i and its number of policy sold n_i.

Equation (1) describes the **maximisation of the profit**: the sum on each rating cell of the premium minus the burning cost, multiplied by the number of policy converted. This maximisation is done under the constraint of the volume of premium being equal to set level V_0. Equation (2) gives the link between the price and the volume of premium written, trivially the number of policy sold times the price they are sold at.

Comments:

- In this model, we consider that the number of policy converted is only a function of the price for the specific rating cell, i.e. all other factors are fixed and the conversion rate of a rating cell is independent of the price of the other rating cells. In reality, it is a good assumption for the conversion rate (all product characteristics and competitors staying the same). However, prices of other rating cells have also an influence on the traffic on the quoting platform, influencing the pool of quotes from which to convert. In the short term, we can consider the pool of quote to be independent of our prices.
- One element of optimisation is regarding the expense ratio. We can divide the expenses into variable costs (cost per policy) which can be included in the burning cost and the fixed

expenses which is a constant term in the equation bringing the profit down. This second term has no influence in the optimisation.

- The optimisation process requires a view on the cost of each policy (burning cost and expense model) and an estimation of the conversion rate depending on the price. The later information is dependent on the current market conditions and leads to the optimisation being only a short-term tool.

To transform the optimisation under constraint into a classic optimisation, a Lagrange multiplier is used. Instead of maximising over the set of prices (p_i) under a constraint, the maximisation is done over p_i and λ. The theory says that the function can only be maximised if the volume is equal to V_0 (constraint verified).

$$\boxed{max\ \sum_i (p_i - BC_i) n_i(p_i) - \lambda(\sum_i p_i * n_i(p_i) - V_0)} \quad (3)$$

The variable λ is known as the Lagrange multiplier. Its value is linked to the constraint on the volume of premium (which is to be equal to the targeted level, see equation (6)) and it carries the information that all the p_i are linked to produce the volume of premium required.

For the search of an extreme value, the differentiates by all variables must be null.

$$\forall i,\ (p_i - BC_i)\, n_i'(p_i) + n_i(p_i) - \lambda\, n_i(p_i) - \lambda\, p_i\, n_i'(p_i) = 0 \quad (4)$$

$$\forall i,\quad (1-\lambda)\,[n_i(p_i) + p_i\, n_i'(p_i)] = BC_i\, n_i'(p_i)$$

$$\forall i,\quad (1-\lambda)\left[\frac{n_i(p_i)}{n_i'(p_i)} + p_i\right] = BC_i$$

For ease of notation, let $(1-\lambda) = C$. C acts exactly as the Lagrange multiplier and is the same for all rating cells.

Let $\frac{n_i'(p_i)}{n_i(p_i)} = -e_i$ with $e_i > 0$. e_i is the percentage of increase in number of policy converted by pound of price decrease. For ease of understanding of the resulting formulae, $e_i$is set as a positive value. $e_i$is the sign opposite of the classic elasticity definition.

$$\boxed{\forall i,\quad p_i = \frac{BC_i}{C} + \frac{1}{e_i}} \quad (5)$$

Equation (5) shows that the optimal price is composed of two elements: the premium to target a fixed loss ratio across the whole book (C) and a premium linked to the elasticity of the market. The less our customers are affected by the price, the more we can charge on top of the common loss ratio. Due to this additional premium, the achieved loss ratio is lower than the "parameter" loss ratio (C).

Extreme cases and examples

In order to get a full understanding of the equation, here are a few examples derived from equation (5).

- In an extremely competitive market (example A), customer will react strongly to a change in price (e_i very large). Therefore the equation (5) can be seen as $p_i = \frac{BC_i}{C} \Leftrightarrow C = \frac{BC_i}{p_i}$. All the rating cells are priced to achieve the same loss ratio (C).
- Taking two rating cells with a burning cost of £50 and a base loss ratio C of 50%. The base premium for both risks will be £100. If the first risk is extremely competitive and a change of £1 in price would reduce the number of clients by 5%, the additional charge is only £20 (1/0.05, example B) on top of the risk premium. On the second risk, the sensitivity is only 1% for £1 price increase. The optimal price is then £100 (1/0.01) on top of the base premium (for a total of £200, example C)

	A	B	C
Burning Cost	£50	£50	£50
base LR C	50%	50%	50%
Base Premium	£100	£100	£100
e_i	+∞	5%	1%
Elasticity Premium	£0	£20	£100
Total Premium	£100	£120	£200
Achieved LR	50%	42%	25%

Table 1 – Examples of optimum prices depending on elasticity

The base loss ratio C is the link with the total volume of written premium. By differentiating the equation (3) in respect of λ and replacing the p_i by their solution given by (5), one gets:

$$\sum_i \left(\frac{BC_i}{C} + \frac{1}{e_i}\right) * n_i\left(\frac{BC_i}{C} + \frac{1}{e_i}\right) = V_0 \qquad (6)$$

The price is decreasing in C while the number of policies converted is increasing in C. As the volume is given by the number of policy multiplied by the average premium, the relation between C and the volume is complex. Intuitively, the book will be growing as a higher loss ratios is targeted (effect of improved conversion being dominant) until a point where the reduction of premium will be the dominant effect (giving away premium with only small change in conversion). There is therefore a maximum on the volume that can be written, which is linked to the price elasticity.

Additional comments:

- The current equation is not fully solved as e_i is an implied function of p_i. To continue in the resolution of the equation, either of the following is needed:
  - express e_i in function of p_i.
  - restrict the movement of p_i to intervals where e_i can be considered constant.
- Once e_i is fixed or expressed as a function of p_i, it is possible to resolve the equation numerically. It is possible to fix the value of C, resolve numerically the equation in $p_i/e_i$to get the prices charged and then calculating V_0. As C is the base loss ratio, it is possible to restrict C to a reasonable interval (0.05 to 2 for example). Under the assumption of known elasticity, the problem can be solved by numerical method.

The following development is looking at the case where only a point estimate of the elasticity is known. In addition, the theory considers real-time pricing and the possibility to set prices for each risk independently. This may not be the case depending on the rating structure (for example, multiplicative structure). In this simple model, the optimised price has two components, burning cost

and elasticity premium (more complex models would consider life-time value, add-ons and expenses as other component of the price). Because of these two additive components having to be merged in a single rating structure (usually multiplicative), there can be a loss of benefits. Work undertaken by the market pricing team on the direct channel has shown that optimised rating tables harvest only 36% of the optimisation benefits achieved by the full flexibility. Elasticity scoring and add-on profit scoring to further differentiate customers with the same risk profile but different elasticity can restate some of the benefits (48%) even with a multiplicative structure. However, the preferred option is to build a rating structure that follows the price components defined in the optimisation solution.

Local optimisation

It is likely that only a point estimate of the elasticity will be known, by opposition to the whole link between price and elasticity. In this case, we should restrict the modification to a small interval around the original price and move progressively toward the efficient frontier.

We can rewrite the equation to allow price modification from an original price.

$$p_i = p_i^0(1 + c_i)$$

Definition: p_i^0 is the starting price of the rating cell i, at which the elasticity is known.
c_i is the price change applied to the rating cell i.

To avoid confusion due to the change of variable from p_i to c_i, the number of policies (and related functions) are redefined as follows:

$$n_i(p_i) = \eta_i(c_i)$$

$$\varepsilon_i = -\frac{\eta_i'(c_i)}{\eta_i(c_i)}$$

$$\boxed{\forall i, \quad C\left(p_i^0(1 + c_i) - \frac{p_i^0}{\varepsilon_i}\right) = BC_i} \tag{7}$$

$$\forall i, \quad c_i = \frac{LR_i^0}{C} + \frac{1}{\varepsilon_i} - 1$$

With the equation for C becoming:

$$\sum_i \left(\frac{BC_i}{C} + \frac{p_i^0}{\varepsilon_i}\right) * \eta_i\left(\frac{LR_i^0}{C} + \frac{1}{\varepsilon_i} - 1\right) = V_0 \tag{8}$$

A quick solution to this problem would be to limit the movement of c_i to an interval (for example +/- 10%). However, the solution given by this method will not match the targeted premium volume and is likely to be non optimal for the volume achieved. On the graph below, the optimisation process stops on the border of the square (portfolio reachable by small adjustments) and produce the blue portfolio while the optimal portfolio within the reach is the top left corner of the square (portfolio pink).

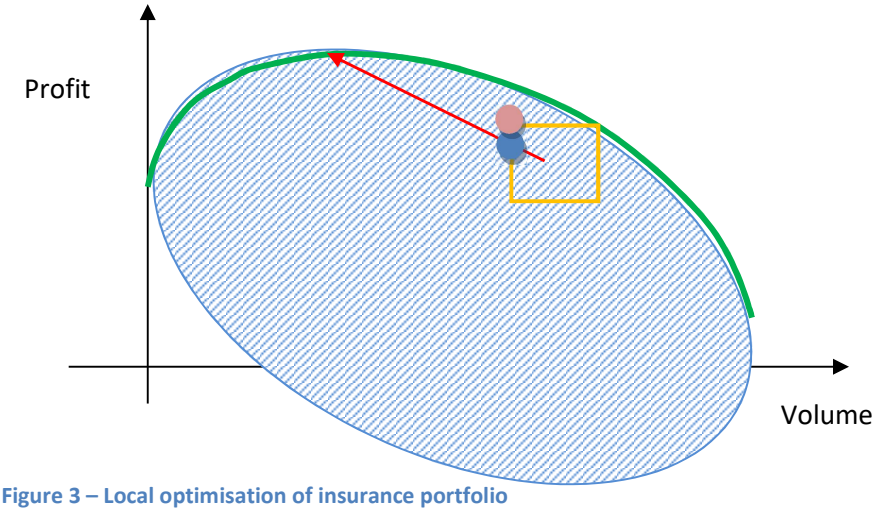


Figure 3 – Local optimisation of insurance portfolio

In the case of limited modifications of the price, it is better to look for a local maximum using a gradient method. Alternatively, a "locally efficient frontier" can be defined within the price range where the elasticity is considered known. As they are limited variations, systematic numerical methods can be implemented (in the style of "valuation on a grid"). Under this constraint, the definition of the efficient frontier will change as the starting position and the range of known elasticity evolves. As illustrated on the figure 3, there is a difference between the gradient descent with a capped variation and an efficient frontier. The correct implementation would be with additional constraints on the prices which can be added as Lagrange multipliers. A proxy for this resolution could be an iterative gradient descent with variables removed once they reach saturation.

It is supposed that successive optimisations of local efficient frontier will get the book closer to the absolute efficient frontier. However, this is not proven and depends on the shape of the elasticity function for all rating cells. A book composed mainly of young professional is unlikely to evolve into, for example, a mid-50s dominated book despite successive optimisation, even if the mid-50s segment is the most profitable.

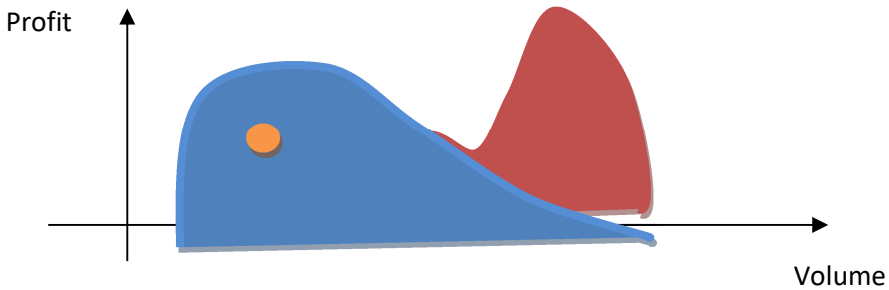


Figure 4 – non convex space of possible portfolios

In the diagram above, the area of potential portfolio dominated by young professional is represented in blue, in red for mid-50s. Here, we are describing a niche market with high profitability but low volume (young professional) compared to the larger market with low profitability of the older and safer drivers. The orange dot is the current portfolio and the starting point of the iterative optimisation. It is unlikely that local optimisation will provide the maximum profit (obtained via the mid-50s portfolio). Instead, a local maximum may be found in the young professionals segment. Optimisation is a short-term improvement and does not replace long term strategy which must be considered in every modelling review of the optimised portfolio.

Necessity of cross-subsidies on a market with fixed costs

Fixed costs can be seen in the optimisation program as starting with a negative profit before writing any business. When considering pure profit (leaving aside consideration of volume for a moment), any business where the premium collected is higher than the sum of claim and variable costs is contributing to the profit. However, when using an average expense ratio to spread the fixed costs, segments of the optimised portfolio would be reported as loss making. The criterion for a true contribution to underwriting profit is a net loss ratio plus variable expenses below 1. When the fixed costs are spread back as a percentage across the whole portfolio, segments included in the optimal portfolio would display a ratio above 1. This is where conventional Management Information would identify the segment as loss making rather than recognising its contribution prior to fixed expense.

This criterion can be adapted if the requirement is expressed as a profitability margin (for example, 95% COR) by adding the profit margin to the variable cost. Fixed expenses are in reality the majority of the expense loading on a policy: they represent the permanent structure required to be in business (offices, computers, employees…). As these costs are significant, there are a significant number of policies in any optimal portfolio that would appear to perform below target.

III) Assessing elasticity

Simplistic approach and the need of a model

The price elasticity is defined as below:

$$e_i = \frac{\delta n_i}{\delta p_i}(p_i)\frac{p_i}{n_i(p_i)} = \frac{p_i\frac{\delta n_i}{\delta p_i}(p_i)}{\frac{n_i(p_i)}{N_i}} = p_i\frac{\delta C_i}{\delta p_i} = \frac{\delta C_i}{C_i}\frac{p_i}{\delta p_i}$$

The elasticity is linked to the differential of the conversion rate. As a result, a point estimate of the conversion is not enough and the focus is on the evolution of the conversion rate in function of the price.

For illustration, we can assume a large volume (2n) of strictly identical quotes with only a difference in price: for half the cases, the price is increased by 1% compared to the base level. Let's call n_0 and n_1 the number of contracts converted on the base and increased price respectively.

A natural estimator of elasticity is then:

$$\widehat{e_i} = \frac{\frac{n_1}{n} - \frac{n_0}{n}}{\frac{n_0}{n}}\frac{p}{\Delta p} = \frac{n_1 - n_0}{n_0}\frac{p}{\Delta p} = \frac{p}{\Delta p}\left(\frac{n_1}{n_0} - 1\right)$$

In order to have the estimator defined for all outcomes (including $n_0 = 0$), it can be slightly changed. Although the change introduces a bias in the estimation, it allows the first and second moments to be defined so that some performance indicators can be calculated numerically:

$$\widehat{e_i} = \frac{p}{\Delta p}\left(\frac{n_1}{n_0 + 1} - 1\right)$$

The first step is to check that the bias introduced reduces with the volume of data to get an estimator with asymptotic convergence.

$$E[\widehat{e_i}] = \frac{p}{\Delta p}\left(E[n_1].E\left[\frac{1}{n_0 + 1}\right] - 1\right)$$

The term $E\left[\frac{1}{n_0+1}\right]$ is more complex so it is calculated on its own below:

$$E\left[\frac{1}{n_0 + 1}\right] = \sum_{i=0}^{n}\binom{n}{i}p^i(1-p)^{n-i}\frac{1}{1+i} = \frac{1}{n+1}\frac{1}{p}\sum_{i=0}^{n}\binom{n+1}{i+1}p^{i+1}(1-p)^{n+1-(i+1)}$$
$$= \frac{1}{n+1}\frac{1}{p}[1 - (1-p)^{n+1}]$$

Back in the original formula and using the conversion rate to express the expected values of the number of converted quotes:

$$E[\widehat{e_i}] = \frac{p}{\Delta p}\left(\frac{n}{n+1}\frac{C_1}{C_0}[1 - (1 - C_0)^{n+1}] - 1\right)$$

$$E[\widehat{e_i}] = \frac{p}{\Delta p}\left(\frac{C_1}{C_0} - 1\right) + \frac{p}{\Delta p}\frac{C_1}{C_0}\left(\frac{n}{n+1}[1-(1-C_0)^{n+1}] - 1\right)$$

$$E[\widehat{e_i}] = \frac{p}{\Delta p}\left(\frac{C_1}{C_0} - 1\right) + \frac{p}{\Delta p}\frac{C_1}{C_0}\left((1-C_0)^{n+1} - \frac{1-(1-C_0)^{n+1}}{n+1}\right)$$

As the second term in this equation has a limit in 0 when n tends to infinity (with $C_0 > 0$), this estimator is asymptotic convergent.

The two underlying variables follow a binomial distribution. It is therefore possible to simulate the estimator and its standard deviation. For the simulation, a conversion rate of respectively $C_0 = 40\%$ and $C_1 = 39.6\%$ are used for illustration and should produce an elasticity of -1.

Number of quotes	Expected value	Standard error
200	- 1.49	12.18
500	- 1.20	7.70
1000	- 1.10	5.45
2000	- 1.05	3.85
4000	- 1.02	2.72
10000	- 1.01	1.72
20000	- 1.00	1.22
50000	- 1.00	0.77

Table 2 – First and second moment of the simple estimator of elasticity

The table 1 above shows that the bias introduced is slow to be overcome, requiring 20,000 identical quotes in each category. However, the bias is known and could be adjusted for (or reduced by changing the "+1" into "+0.01" in its definition). More importantly, the standard error of the parameter is significant. This is due to its construction as a ratio where the denominator can come close to 0. This triggers some extreme values that create volatility. The volume required to have a reliable estimation (especially when we are trying to assess if the elasticity is greater or lower than -1) is beyond the 50,000 identical quotes in each category. Gathering this volume of identical quotes is impossible, especially in commercial insurance.

In consequence, variations between quotes need to be allowed and recognized in the estimation process. Doing so would enable to use similar quotes rather than strictly identical quotes. This is required to have enough volume to assess elasticity. From a point estimate of the price sensitivity, the model is now trying to assess a full multi-dimension function with non-price related characteristics feeding in. As the problem is now drastically more complex with a full multi-dimension space, some assumptions need to be made to simplify it again.

The choice of the model is one of the main challenges of this paper. The current state of play in the UK General Insurance can be assessed via the working parties and publication of the Institute of Actuaries. The General Insurance Research Organisation (GIRO) set up in the late 90s a working party on selection and retention of risks. The results of the working party were presented to the 1997

General Insurance Convention and set the basis of a GLM model for conversion and retention modelling[1].

The complexity of the modelling improved in the following decade but the underlying assumptions of the model were still considered relevant in a presentation to the 2010 General Insurance Convention. The focus was more on "super factors" to improve the predictive power and how to implement an optimised pricing decision[2].

Although machine learning is a current hot topic in General Insurance pricing seminars as of 2016, the actual implementation is considered commercially sensitive which limits the documentation available on the most recent practices. It is generally mentioned but there is no detail on the method or even the scope of utilisation. Verbal updates during pricing seminar highlight benefits in risk pricing with a focus on understanding the profitability rather than calculating prices. However, insurers are now deploying additional processing power to their rating engine to support more complex calculations, including machine learning models. Crucially, the type of model used is never communicated. As a sign of change in the industry, Willis Tower Watson has updated its optimisation software, historically only able to support their proprietor General Linear Models, to be compatible with R models[3].

There is a wide range of potential models that could be considered in this review. The classic approach of a generalised linear model with a logistic link function will be used as a benchmark. This is quite prescriptive of the shape of the surface and the independence of effect of each factor. The independence of effect can be removed via the introduction of interactions but this solution cannot be applied systematically: the GLM relies on restricted freedom to avoid over-fitting so interactions can only be used parsimoniously.

Within the machine learning algorithms available, two have been considered for this review: Gradient Boosting Machine and Random Forest. Both algorithms are based on tree-classifiers. There are other types that have been considered for inclusion in this paper but were not used due to time and technical constraints.

The first category of models which have been left on the side is the penalized regressions. These algorithms are close to the human process of building a regression and result in models that can be easily explained due to their similarity to the current models built. Although penalised regressions fit the requirement of speeding up the process, they do not provide an alternative to the linear constraints of a GLM.

The second category are support vector machine algorithms which are also related to a linear model (subject to the kernel transformation). Due to the kernel transformation being difficult to visualize, these models operate more as a black box which was considered detrimental for their acceptance by the business. Two additional facts further undermine the use of support vector machine: the data

[1] https://www.actuaries.org.uk/documents/**customer-selection-and-retention**

[2] https://www.actuaries.org.uk/documents/b05-price-**optimisation**-20

[3] https://www.willistowerswatson.com/en/press/2016/07/Speed-and-scale-integral-to-insurance-pricing-evolution

not being linearly separable and the influence of the choice of the kernel transformation on the outcome.

The third algorithm only considered is Neural Network. This algorithm is usually quoted by consultants as part of the machine learning revolution. This category includes a wide range of sub models with different architectures and learning process. The number of neurone and their organisation (especially the number of layers) is a key element of performance and would require multiple runs to assess the impact. Finally, the interpretability of the results is difficult as each neurone creates its own recombined variable that feeds into the next layer.

The last category considered is the class of predictive clustering algorithms. This category has the advantage of producing an output that can be understood and implemented. However, no algorithm was readily and easily available in R for implementation. Additional work would have been required to develop a working algorithm in another system which is beyond the simple proof of concept covered by the paper.

Overall principles of the models used

General linear/additive models

Generalized Linear Models are well understood and documented in the insurance industry and actuarial sciences so they will not be covered here beyond highlighting the differences with the other models presented.

The GLM assumes each observation follows a Bernoulli distribution with a probability defined as $P(x) = \frac{1}{1+e^{-\sum_{\alpha,i} c_\alpha \delta x_i = \alpha}}$ with the coefficients c_α calibrated by maximum of likelihood. The restriction of the degree of freedom limits the possibility of over-fitting and differentiates underlying trends from noise.

The main difficulty in a conversion modelling is that the list of factors of importance changes as the competitors adapt their pricing strategy. A predictive factor in a model may no longer be predictive at the next iteration if the pricing strategies of the insurer and its competitors are aligned. Conversely, a factor ruled out previously can become predictive. As a result, the full list of rating factors need to be review at each refresh of a conversion model. This creates a significant workload to refresh a GLM model when new data is available.

One point of note on the GLM used in this paper and more widely in the industry: the GLMs are usually performed using a specific proprietary software (EMBlem) with an algorithm converging extremely quickly. However, the EMBlem implementation of GLM forces a discretisation of variables which adds a lot of freedom as continuous variables have one parameter per discreet class rather than one for the whole range of value. To rebalance, EMBlem can build polynomial approximations to reduce the degrees of freedom on a continuous discretised variable. In many ways, this behaves similarly to a spline in a General Additive Model. Although it will be described as GLM in the rest of the paper, this refers to the GAM-like implementation in EMBlem with splines (which are engineered features in a GLM models and therefore can be considered as a GLM).

Ensemble methods

By opposition to the GLM, both machine-learning algorithms studied here are some kind of ensemble predictor. The idea of ensemble models comes from Condorcet's work on the Probability of Majority Decisions. Under the following assumptions, the probability of error tends toward zero as the number of judges tends to infinity:

- Each judge has a probability of taking the right decision better than a random process (probability of being right higher than 50%)
- Each judge is independent of the others
- The judgement is given by the majority of votes

Although all judges are equal in Condorcet's principle, the concept still holds with different or conditional weighting between predictors. For example, let consider a bag with apples (green or red) and pears (green only). Two classifications of fruits into apples and pears are built: one mainly picking up the colour, the other one responding mainly to shape. On a red fruit, the colour algorithm will perform really well at identifying an apple. On a green fruit, the shape algorithm is likely to work better. By combining these two models, we can produce a model more predictive than each one individually. How to combine these to get the best result is a complex problem without a single solution and the quality of the combination is usually the difference between the best performing models (illustrated in the outcome of Kaggle competitions).

Bagging

A second concept used by both methods is bagging (Bootstrap Aggregation): the idea is to generate a random set of observation on which to train the model and aggregate (average) the resulting models generated on multiple instances. This has the advantage of limiting the leverage of individual observations. As a result, the additional flexibility given to non-parametric models does not always result in over-fitting to the original data. This also brings some kind of independence between the predictors as they are trained on different set of data, helping to verify the Condorcet principle.

Out-Of-Bag (OOB) and cross validation

With the idea of bagging, some data is left aside from the training at each step. This data "out-of-the-bag" (OOB) can be used to assess the performance of the fit (without compromising the real out sample). Therefore the error on the OOB is a useful tool in assessing the improvement of the model beyond over-fitting. However, the OOB sample is quite small and therefore the prediction of the model is volatile. If this is used to assess improvement from the previous step, the noise related to the model performance can give the wrong message, showing a deterioration of fit when the model improvement is real but small compared to the volatility.

One way to stabilize the OOB performance measure is by cross validation: A 5-fold cross validation would divide the data into 5, train the model on 4 fifths, test on the last one and repeat the operation with each fifth of the data being used as a validation sample. This would moderate the results from the OOB as each part of the data play in turn the role of the OOB, bringing more stability to the test.

Importance of variable in tree classifiers

The importance of each variable in a tree is calculated using a permutation test. At each node, a feature is used to split the total population into 2 sub populations. These populations show different levels for the average response variable. The permutation test is assessing the likelihood that this separation could have been achieved by a random choice. If it is unlikely, the variable is bringing some valuable information. If it is likely, the feature may just provide a lucky split.

For example, let's consider a bag with 100 marbles, 10 white and 90 black. The next step is a feature that isolates 5 marbles. If the 5 marbles are white, the split in packs provide a pack of 5/5 and another of 5/95. This is an unlikely event done by chance. There is, as a result, a strong confidence that the feature is predictive. On the other side, if the 5 marbles are black, we have still improved our prediction (0/5 and 10/95) but the likelihood that a random cut would have achieved the same element is high. Therefore, it is difficult to credit the feature for the improvement. By summing and averaging the results of this test over all nodes and all trees for each feature, it is possible to rank the features in order of contribution to the prediction.

Gradient Boosting Machines

The Gradient Boosting Machine is a machine learning algorithm relying on the accumulation of weak learners (decision trees). At each iteration, the weak learner that improves the most the predictor is selected (gradient descent) and the weight of the observations is changed to emphasize the one which have been misclassified (boosting). As the weights are different at the next step, the next weak predictor to improve the model is different from the previous ones selected. This model focuses on the previous blind spot which means that it performs targeted learning and tends to learn faster than other algorithms. Left with enough time and iterations, a GBM would be able to recreate any separable data and therefore over-fit.

There are two tests to assess when to stop a GBM: the out-of-the bag (OOB) error arising from the bagging step and a cross validation. The OOB classification error is calculated using, for each data point, the prediction of only the predictors which were calibrated while the data point was excluded. The result at each step is dependent on the random sample and therefore can be quite volatile. The cross validation is looking at performing this on multiple samples (number of folds) and averaging the results. There is also a smoothing of the error improvement/deterioration over consecutive iterations to improve stability of the cross-validation test. This increases the computation time but give a more stable measure of the loss of generalization power at each step thanks to the averaging over the multiple folds and iteration by opposition to the OOB which would stop at the first deterioration.

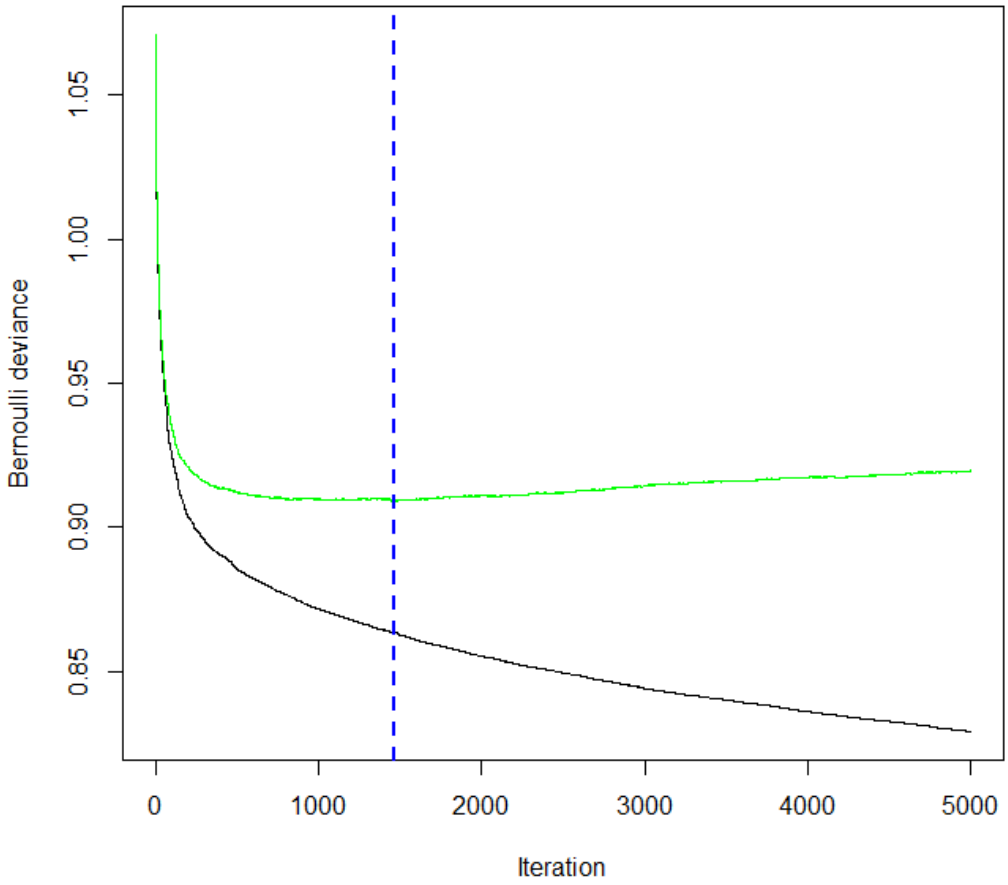


Figure 5 - Example of cross validation test

The figure 5 is an example of the evolution of the Bernoulli deviance in a GBM over various iterations. In black is the curve of the overall model which is constantly decreasing as predictors are selected toward this goal. In green is the result of the cross validation. It is apparent that, beyond a certain point (here, about 1500 trees), the model starts losing its generalization power and only explains the training data. This is a way to select the optimal number of iteration (number of trees built). There is a word of caution needed on the implementation in R: despite the averaging of the cross validation, the deviance is still volatile. Therefore, the curve is smoothed to identify the inflexion point. To ensure that the smoothing is appropriate, it is important to have enough data on either side of the inflexion point. Therefore, the number of trees grown should be largely in excess of the cut-off point. A declared inflexion point in the last 5% of the trees grown may be linked to a defect of the smoothing based on the absence of data points in the tail.

The GBM has been presented as an ensemble method as it pulls together weak learners to create a better predictor. Here, the ensemble process to recombine all the weak learners is based on a decreasing weight (which is a parameter of the model): each subsequent weak learner receive a smaller weight ensuring convergence of the model. Smaller step will increase the computation time, the size of the finished model and the time to apply it. If the step is really small, two consecutive weak learners selected may be identical (and therefore stored in duplicate for no reason). Larger steps may miss some level of details. There is a trade-off between accuracy and computing time that is left to the programmer to decide. The constraint for this paper was to run under 30 minutes to fit

with the desired outcome of quick production of models. This limit is arbitrary but provides a level-playing field for the comparison of machine-learning models.

Random Forests

By opposition to GBM, Random Forests[4] are a classic application of Condorcet principle with all predictors voting to a majority with the same weight. This ensemble method is based on classification trees like the GBM. To ensure independence between the various trees, each tree is grown out of a sample of the observations (bagging) with only a sample of the features available for building the tree. This is set to generate enough diversity in the trees to emulate the independence between predictors.

A random forest algorithm has a few parameters that can be tweaked: the sampling proportion of observation for each tree (with or without replacement), the sampling proportion of the features and some limitations to the tree complexity. The first two are to control the diversity of the trees grown. Higher proportions give better individual predictors but increase the correlation between them and remove most benefits of the ensemble stage. Lower proportions create weaker but more diverse learners and require a higher number of trees to be grown. The parameters related to the tree complexity can take the shape of a maximum number of nodes or a minimum number of observations in each node. Finally, the number of trees grown is technically a parameter, although, in practice, it is set as high as affordable in terms of time, computing power and speed of application.

Each tree has a risk of over-fitting the data on which it is trained. The feature selection is a first moderator as the data is aggregated in a smaller number of rating cells (group of observations with identical values for all features available) with some observations that cannot be distinguished with the restricted features (data not separable). The algorithm also supports usual restrictions for tree classifier such as maximum number of nodes or minimum number of observation in a leaf if required. No restriction on each tree would mean that every feature selected for a tree will be used to get as granular a tree as possible with as many leafs as rating cells. As each tree is stored for the application of the model, complex trees can create models of a significant size and slow down their use. With the additional protection of the bagging (aggregation of trees grown on bootstrapped data), the Random Forests have good protections against over-fitting and are considered incapable of over-fitting when reasonable parameters are used.

Following Condorcet's principle (adding predictors better than a random one), a Random Forest with an infinity number of trees could completely recreate a training sample with perfect accuracy (as long as the data is separable). This would appear like over-fitting the data as the prediction power on

---

[4] The Random Forest algorithm was developed by Leo Breiman and documented in the following publication for more information:

Breiman, L. (2001), *Random Forests*, Machine Learning 45(1), 5-32.

Breiman, L (2002), "Manual On Setting Up, Using, And Understanding Random Forests V3.1",http://oz.berkeley.edu/users/breiman/Using_random_forests_V3.1.pdf.

data never seen before would be significantly lower than the predictive power on the training data. Over-fitting tends to create this kind of behaviours between in and out samples but it is not the definition of over-fitting: a model is over-fitting when its performance improves on the training sample while deteriorated on new data. In the case of random forest, the performance on the out sample still improves, but at a slower speed than on the in sample, creating this difference of performance frequent in over-fitting problems. The assertion that random forests do not over-fit under reasonable parameters is correct. However, the performance on the training sample is not reflective of the performance on new data. A better measure of performance is using the Out-Of-Bag (OOB) error and will be discussed in the implementation section.

The calculation of each tree is performed independently of all the others. Unless an OOB error needs to be calculated (in which case, the information of which observation trained which tree is needed), the building of trees can be parallelized to speed up the algorithm. Multiple computers can train on the same data, generating forests that can be then recombined into one model. This enables to grow very large forests in a set time by pooling more computing resources together

However, here is a final note of caution for random forest users emboldened by the impossibility to over-fit: there is a limit to the number of trees that can be grown and considered independent. The number of permutation of features is limited and the bootstrapping of a training sample leaves some correlations between the predictors. This limit is high but not infinite and users should not aim to produce millions of trees in the quest of the perfect model.

IV) First example: Casualty insurance

Product description

The product targets tradesmen and professionals who do not need a property cover. The bulk of the market is sole traders working on their own in construction, plumbing or electric systems. For these, the only cover required is Public Liability. For slightly larger businesses, an Employers' Liability cover is available. In addition to these two main covers, a range of small add-ons are available for tools or personal accident. These are unlikely to influence conversion as they are marginal, both in terms of premium and frequency of quote.

The two main covers are rated on only 5 factors: occupation, number of people, limit of indemnity, claim experience and geographic area. Most business will show the same claim experience (3 years claim free) as the claim frequency on this type of product is extremely low (about 1%). The main differentiator is the occupation which presents thousands of levels (which can be grouped into less than 200 categories generating the same premium). The premium is calculated via a multiplicative structure within each cover and additive between covers.

In addition to the risk factors, the distribution environment has an important influence on the conversion: commission level, special discount (override), distribution channel and month (to capture changes done by the competition).

Data Description

The conversion model is built on a subset of quotes received over a period of 3 months at the end of 2015. The quotes have been treated to remove duplicates and represents unique opportunities of sale. Over the period, 2 rating series were active and all prices have been normalized as if provided by the latest rating model (base premium), with all differences gathered in one variable as the price variation from base.

The subset of data analysed (around 65,000 quotes) has been split into a training sample and a validation sample using a random variable so they both cover the whole period. The training sample was set as 9 tenth of the data and the validation sample at the remaining one tenth. 32 explanatory variables were kept, most of them were categorical. Only 4 variables were truly continuous: company turnover, commission ratio, base premium and premium variation. Other numerical variables were only present at set levels (for example, number of people covered (1 to 10)). The response variable was a flag for confirmed sale (excluding sales cancelled during the cool-off period) with the average conversion rate between 20% and 30%.

Models description and analysis

Three models have been built on the in-sample of the data: a Generalized Linear Model, a Gradient Boosting Method and a Random Forest. The table below summarize the factor selection for each model (factor selection for GBM and random forest are expressed as importance). One of the most predictive factors (Occupation) could not be used in the GLM due to the degree of freedom

associated to it leading to over-fitting issues. A grouped version (Occupation family) was used instead. The random forest algorithm gets contribution from a larger number of factors due to the algorithm selecting only a subset of the features to build each tree (see table 3).

Variable	Selection in GLM	Importance in GBM	Importance in RF
Occupation	Proxy (cf. occ fam)	36.98	13.53
Portal	Yes	14.41	9.28
Commission ratio	Yes	11.68	10.52
Base premium		7.55	10.60
Month	Yes	4.61	5.08
Premium Change	Yes	2.94	4.47
EL rated area	Yes (interaction)	2.75	5.39
Number of people (PL)	Yes	2.38	2.24
Rating version		2.34	1.01
Annual turnover		1.96	6.22
Max height of work		1.91	3.49
Flexed commission		1.4	2.68
PL rated area	Yes	1.25	5.04
Override type		1.22	0.45
Total premium		0.96	3.55
ERF	Yes	0.76	1.63
PA cover		0.75	0.98
Company status		0.68	1.77
Max depth of work		0.53	1.37
Tool cover		0.53	0.87
Occupation family	Yes (cf. occupation)	0.43	4.35
EL cover	Yes (interaction)	0.41	0.73
PL Limit of Indemnity		0.39	1.67
Hired-plant cover		0.28	0.48
Gas work cover		0.21	0.37
Premium override		0.18	0.33
Stock cover		0.17	0.28
Electric work cover		0.13	0.33
Contract cover		0.13	0.29
Plants cover		0.09	0.35
Tradesman / Profession	Proxy (cf. occ fam)		0.50
Business Equipment cover			0.16

Table 3 – Influence of variables in the 3 fitted model on Casualty Insurance

The second comparison of the models is via the comparison of their ROC curves: All observations are ordered by score (probability for GBM, proportion of votes for RF) from high to low. The line for a model is parametrized by a cut-off score: Amount the population with a score above the cut-off, which proportion of the total converted population is present (displayed on the vertical axis) and which proportion of the not converted are included (displayed on the horizontal axis). The bottom left point is a cut-off above the maximum score allocated (therefore 0% and 0%) while the top right is a cut-off lower than any score given by the model (therefore full population with 100% converted

and 100% not converted). Ideally, one wants a model that identifies most of the converted first without capturing the not converted (avoiding false positive): this is represented by a steep rise from the origin point. Models can perform better over different segments of the curve so a ROC curve is not an absolute ranking of models but can be useful in simple situations.

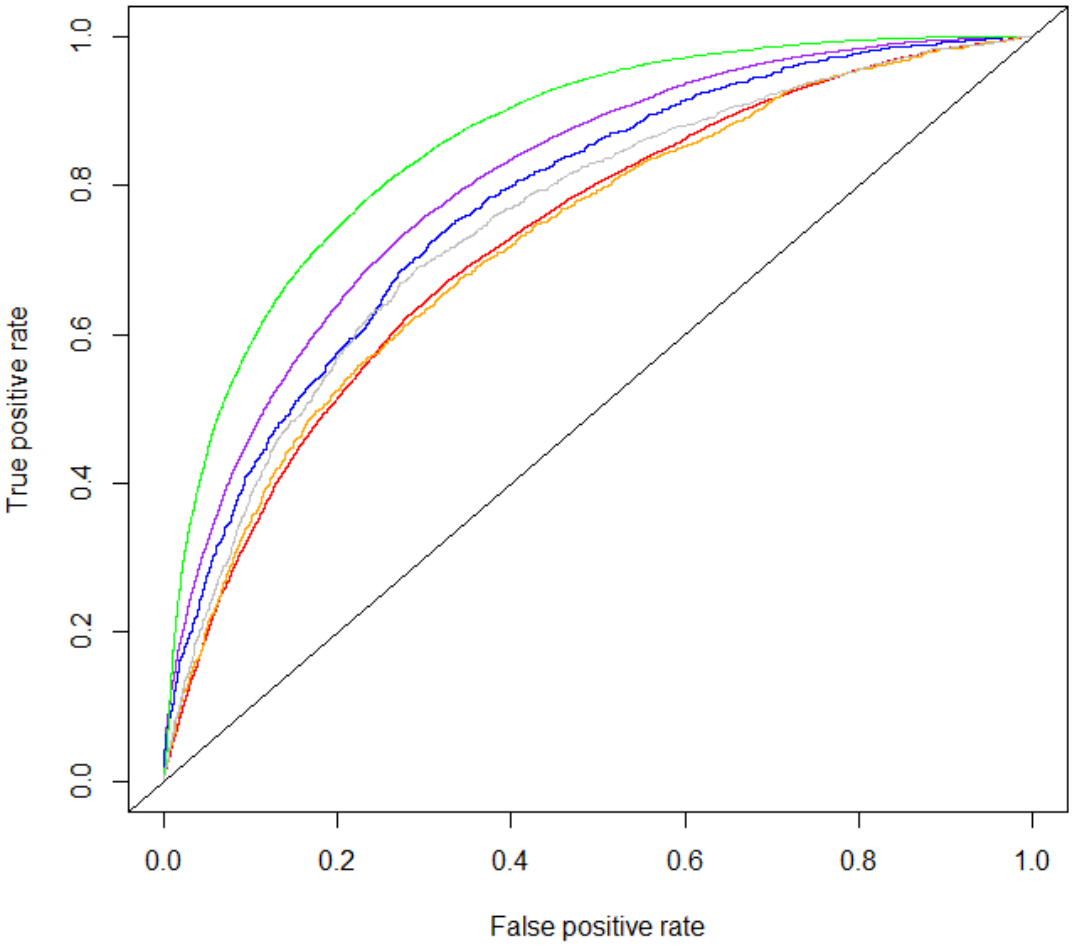


Figure 1 - ROC curve (In and Out samples) on Casualty Insurance

Red = GLM in-sample	Orange = GLM out-sample
Purple = GBM in-sample	Blue = GBM out-sample
Green = RF in-sample	Grey = RF out-sample

The GLM model shows great stability between in and out samples. However, the predictive power is lagging behind both machine-learning algorithms. The Random Forest prediction score on the in sample is calculated here using all trees and not just the OOB ones. This leads to an overstated predictive power on the in sample. Although there is a loss of predictive power for the GBM on the out sample, it is still the most predictive of all 3 models.

Recalibration of the random forest

Interestingly, the random forest provides a good score to identify conversion (as shown by the ROC curve) but does not reconcile in absolute probability. The mean conversion rate of the training set is 23.6% while the mean probability predicted by the random forest is only 8.7%. As the model is

correctly ranking the quotes in order of probability of converting, the predicted value can be used as a score to predict a real probability.

The link between score and probability needs to be monotonously increasing with value into [0;1]. To avoid making more assumptions on the shape of the link between score and conversion, an isotonic regression with a PAVA algorithm is used to derive the link.

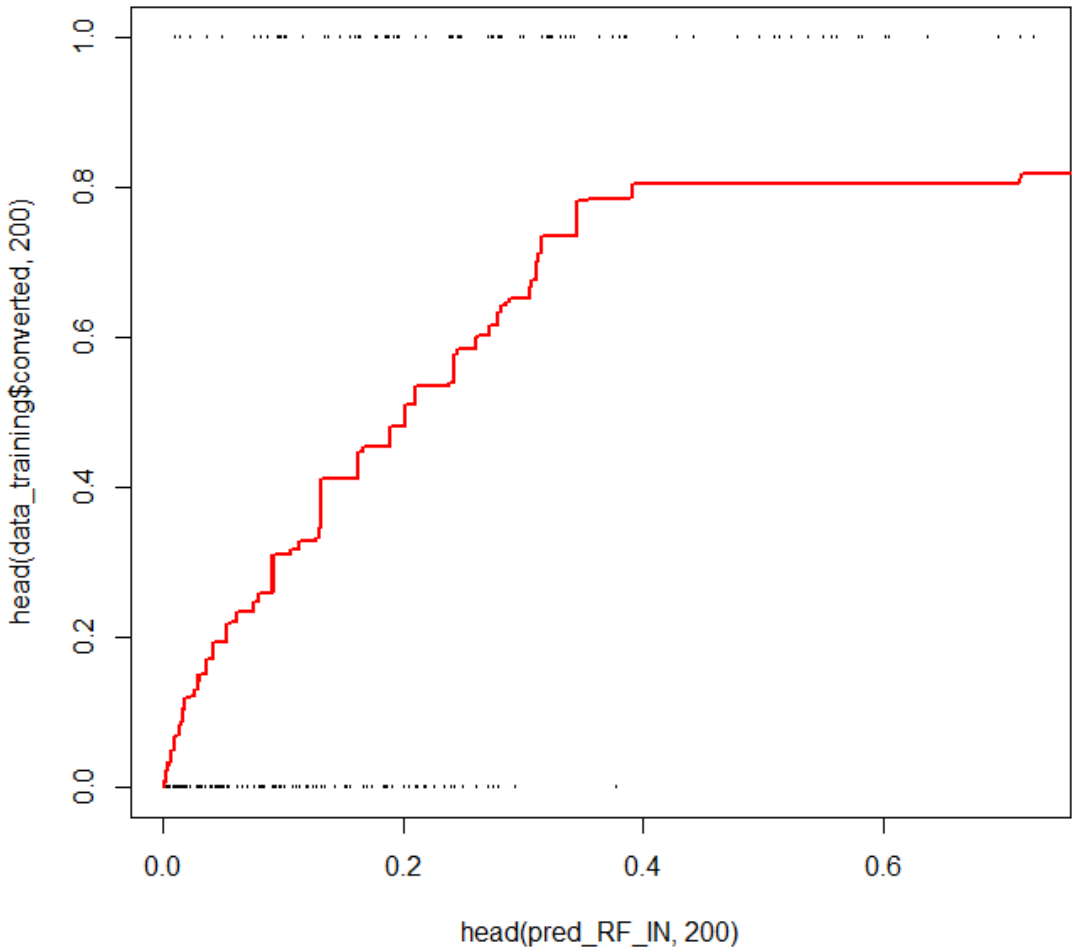


Figure 3 – Isotonic regression to recalibrate Random Forest prediction

This has no real impact on the ROC curve (apart from a small loss of granularity) as shown on the diagram below

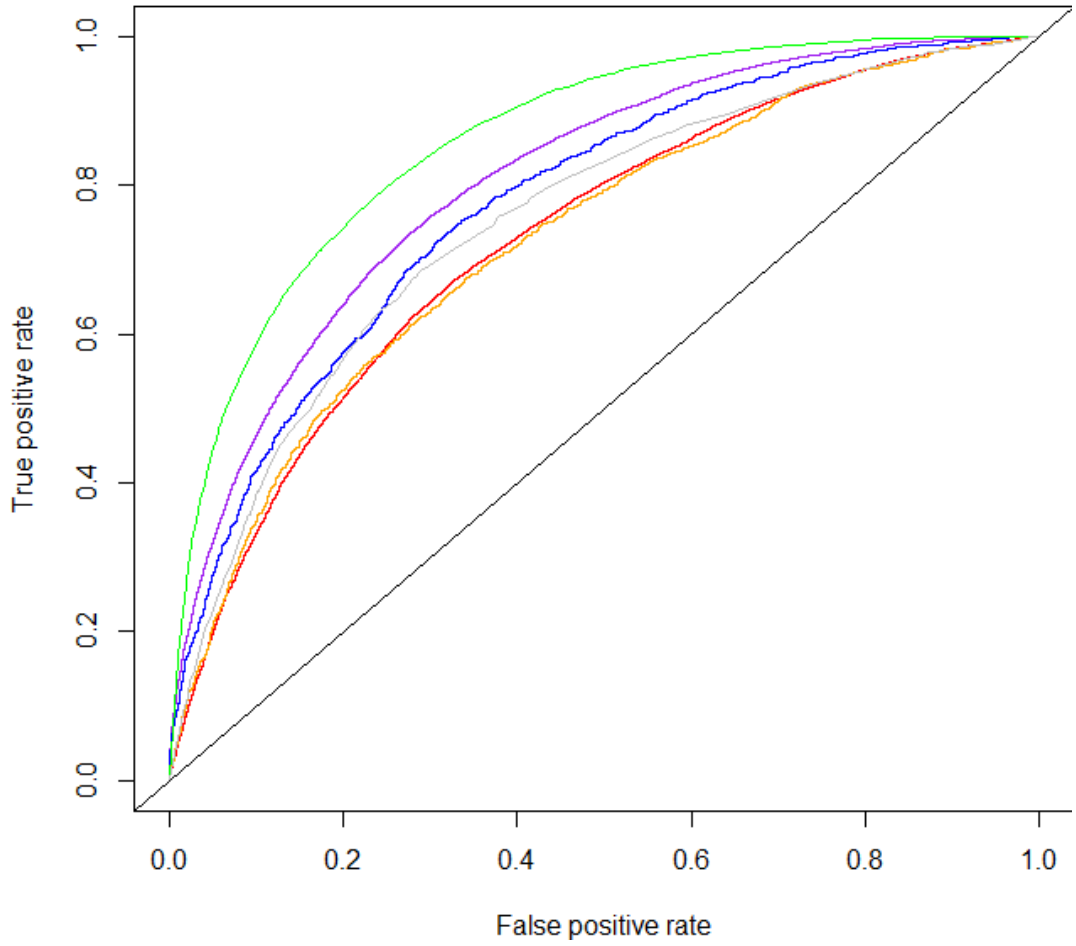


Figure 4 - ROC curves (RF recalibrated)

Red = GLM in-sample	Orange = GLM out-sample
Purple = GBM in-sample	Blue = GBM out-sample
Green = RF in-sample	Grey = RF out-sample

The underlying reason for the Random Forest not providing the correct prediction can be seen in the OOB error close to the conversion rate: the Random Forest prefers to predict that no policy convert and get an error rate close to the conversion rate rather than attempting a guess. This is due to the large number of explanatory variable without signal fed to the model. The trees grown using these variables cannot make a real prediction, sticking to the most observed value. Alternatives to fix this issue can be to reduce the number of variables without strong signal or rebalancing the quotes to conversion ratio to 50% (a posteriori sampling) to force the Random Forest to make a decision even in presence of low signal. The first option reduces the number of explanatory variable tested which is not optimal for our automatic process. The second option adds a layer of complexity to the implementation with potential consequences on the quality of the prediction that would need to be assessed. For these reasons, the GBM has been deemed better suited for the use as an automatic model builder.

Implementation and use

The first use of the machine learning models was to check that nothing obvious was missed from the GLM selection. This was done via a comparison of the importance of factors: some factors could not be included in the GLM due to the large number of levels and low exposure but a proxy should at least be present. For example, Occupation could not be fitted but Occupation family was in the GLM to work as a proxy for occupation. This highlighted some important factors missing from the original GLM model.

The second step was to compare the prediction of the GLM and GBM models and build a decision tree on the difference. If the decision identified good split rules, these should be investigated in the GLM. In the decision tree below, the top node is a split by number of people covered (strictly less than 4 on the right, more on the left). Within the left sub tree, further splits appear by occupation, distribution channel and commission with a pivotal value of 27.5%. This value was not identified as a pivot in the GLM which had a banded commission ratio (as the software cannot deal with continuous variables) while the GBM had access to the raw data. It was identified afterward that a specific broker was trading at this commission rate due to a specific agreement with the insurer. This fact was lost for the GLM with a commission band of 25%-35%. The GLM model has been updated to recognise this specific relationship. It is the final and improved GLM model that is compared in the section above.

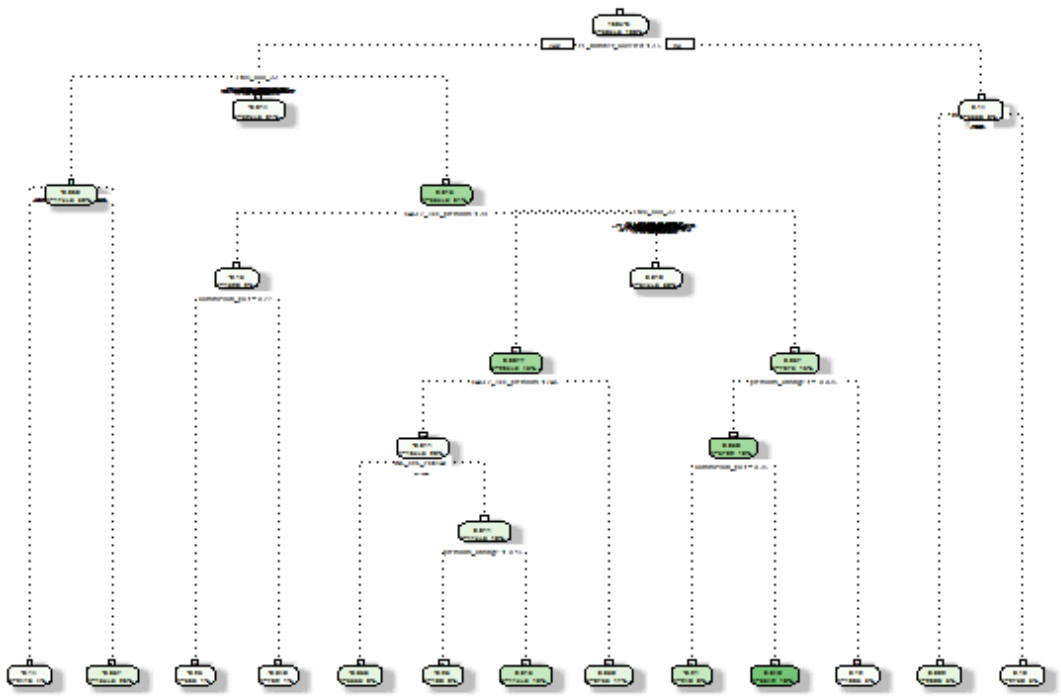


Figure 5 – Tree classifier of the difference between GBM and GLM

Beyond supporting the improvement of the GLM, machine-learning algorithms can and will be used directly as a conversion predictor. This implementation need to be phased for the business to grow in confidence in using these new tools which operates like black boxes. For a first implementation, the models have been considered for support rather than take the final decision so that human moderation could pick up any deficiency of the model. The two applications considered are for impact analysis and elasticity measuring.

In an impact analysis, a price change can be calculated on all quotes. However, the real impact on the portfolio is the price change on policy sold, not quoted. To give a good proxy, each quote can be weighted by their probability of conversion given by a conversion model. Post-implementation analysis would be able to verify that the weighting provided by the conversion model is a good proxy. Providing an up-to-date conversion model for each price change with a GLM has proved to be challenging due to resource constraints. As a decent machine-learning model can be built within a short space of time using the same code as the previous time without intervention from an analyst, this is a convenient alternative to out-of-date GLM models.

The second use considered is an intermediary step of elasticity modelling: a conversion model needs to be built to normalize the conversion of each quote before assessing the impact of the price change. The accuracy of the normalization will have an impact of the elasticity reading. Therefore, the better the prediction is, the better the assessment of the elasticity will be. From what we have seen on the out sample, machine learning algorithms can deliver better and faster results compared to GLM.

The predicted conversion rate is transformed into a linear predictor for a GLM model and used to normalise for the risk via an offset. Below is the graph on a one-way showing the normalised conversion against the actual observed. From there, the variation from the expected conversion can be attributed to the random price changes which have been put on the quotes to assess price sensitivity.

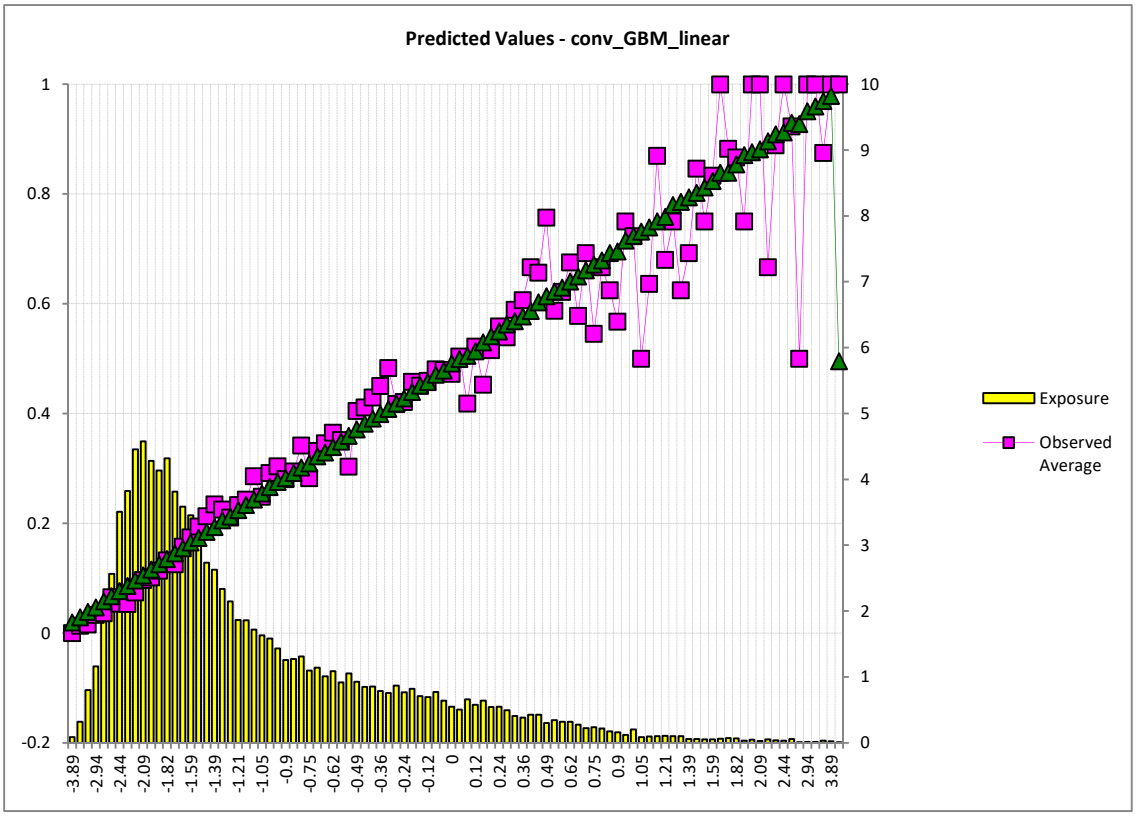


Figure 6 – GBM score in an GLM elasticity model

The model then assesses the impact of the price variation on the conversion via a 3-way interaction.

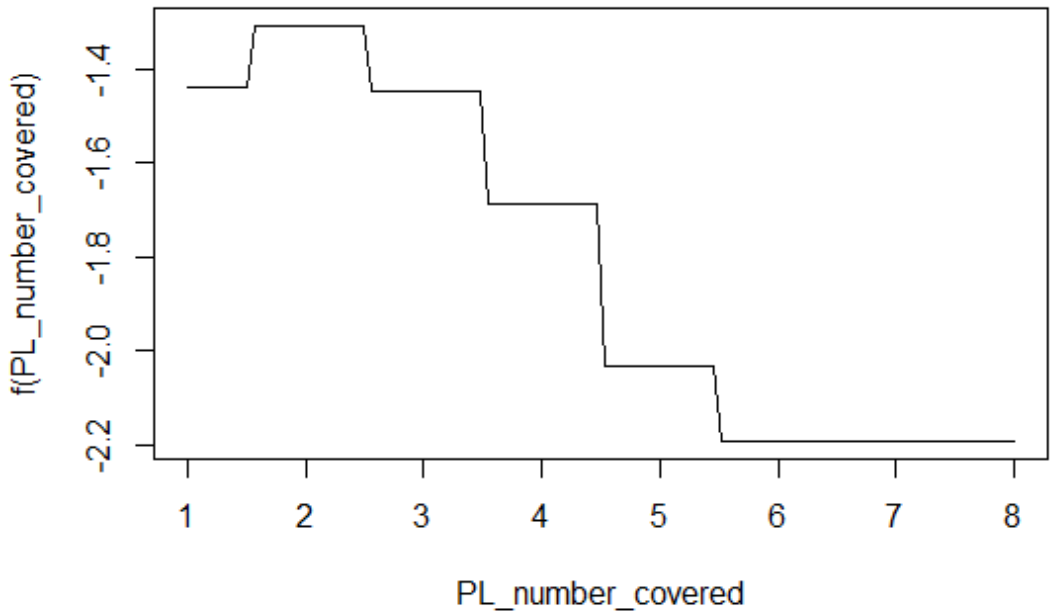
Category	Distribution	Elasticity (increase)	Elasticity (decrease)
PCT	SWH	- 3.71	- 2.74
PCT	Extranet	- 2.42	- 0.16
PCP	SWH	- 2.60	- 0.80
PCP	Extranet	- 2.63	- 0.16

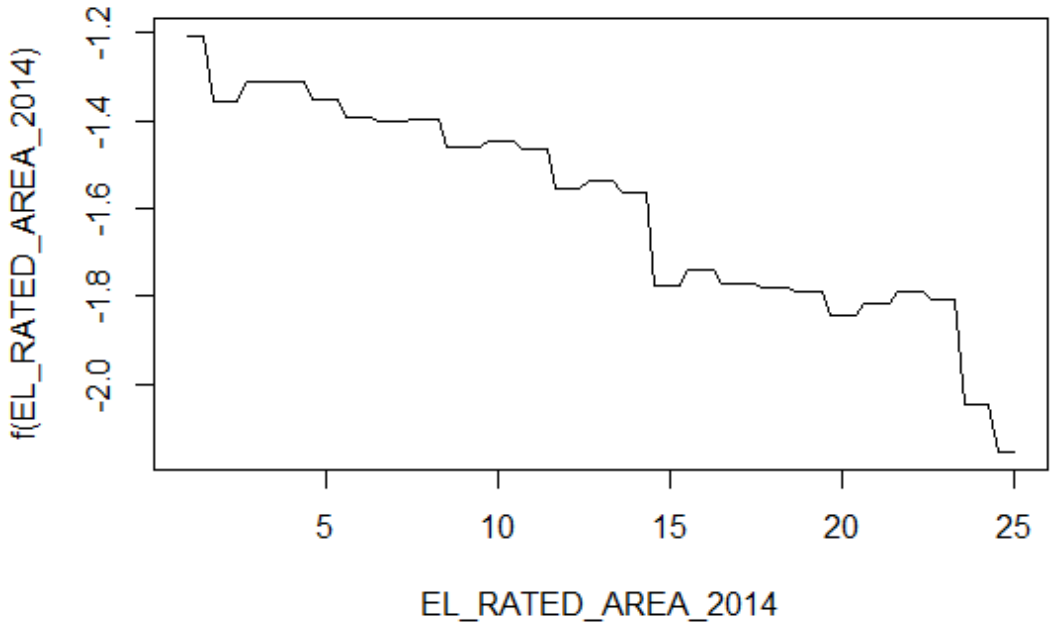
Table 4 – Elasticity reading from a GLM model using GBM normalisation

The result shows that a price reduction on the Extranet distribution channel would not bring any additional business (elasticity close to 0 for price reduction) while intense competition on all distribution channel means that a price increase would lead to significant loss of business (elasticity above 1). The asymmetry of the effect shows that the product is leading in price and would not gain from further price reduction while the competition is close behind. The exception is the Tradesman product on software house where brokers are looking for lower prices on this higher risk segment.

Each application feeds into a manual decision process (price decision by underwriters and estimation of the elasticity) so there is a sense-check before going live. It is important to build a better understanding of the models and some diagnostic tools before plugging the result directly into a mathematical optimisation. Any irregularity of the conversion rate would be exploited in full by an algorithm searching for an extreme value, potentially leading to erroneous price recommendations.

Some diagnosis tools are already available in R. These can be helpful to verify some trends but lack meaningful scales to support decision making. The graphs below validate the trends seen in the GLM but show also some volatility (especially on PL area).





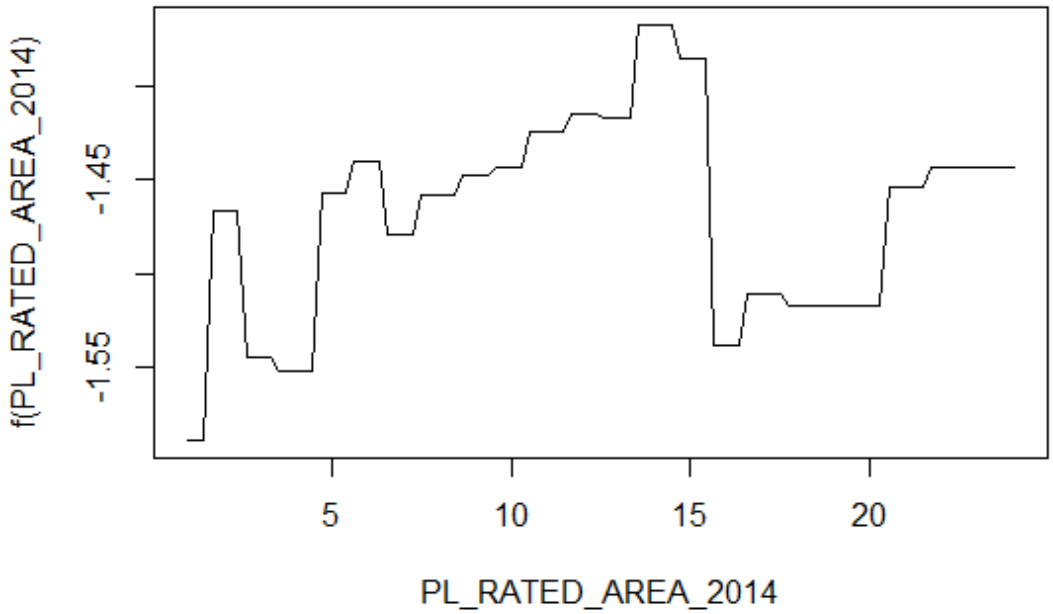


Figure 7 – Marginal contribution by factor in the GBM

The first graph (number of people covered) is in line with the pricing strategy which is to be attractive to low number of people (identified as the profitable market). The second one is also in line with the strategy of charging more for areas where companies specialised in claim farming are present. The third one on the other side does not show any significant trend. The scale seems to indicate far less predictive power then the two previous factors, with a factor 10 in terms of variations. However, the importance of this third factor was half of the first two showing still some predictive power but more likely through interactions rather than one-way.

As mentioned above, the difficulty is to transform these marginal contribution graphs into pricing decisions. This is usually done via measuring the change in conversion and dividing by the price elasticity to get a distance to the market price. An option considered was to attach the GBM model to a batch of quote multiple times, each time forcing on all quotes the studied factor to be at one specific value. By ranging through the various set of quotes, the impact of that factor can be quantified. However, the outcome of such a run is dependent on the batch of quote selected in the first place. Unfortunately, the creation and testing of diagnosis tools for GBMs is out of the scope of this report and is an area identified for future work.

V) Second example: Landlord insurance

Product description

This product covers properties not occupied by their owners. The mandatory covers are Building and Property Owner Liability. Optional covers such as rental income, contents, accidental damage or employers' liability can be a significant addition to the risk. More importantly, the policy can cover multiple locations which weaken the link between one location risk details and the price seen by the client. Each cover and location is purely additive for the premium. Therefore, the impact of one rating factor on the conversion can vary significantly between risks. For example, the building sum insured can be the main driver of the conversion of a single location with building cover only. However, on a risk with 4 locations and rental income cover, the influence of the first building sum insured on the final price and conversion is only marginal. This lack of consistent marginal effect is a challenge to the usual independent contributions of each factor in a GLM that may need interactions.

Data description

The conversion model is built on a subset of quotes received over a period of 9 months to the end of 2015. Using the same selection rules as for the Casualty product, the longer period was required to gather a similar volume (around 65,000 quotes). The quotes have been treated to remove duplicates and represents unique opportunities of sale. Over the period, 6 rating series were active and all prices have been normalized as if provided by the latest rating model (base premium), with all differences gathered in one variable as the price variation from base.

The data has been split into a training sample and a validation sample using a random variable so they both cover the whole period. The training sample was set as 9 tenth of the data and the validation sample at the remaining one tenth. 34 explanatory variables were kept. Only 4 variables were truly continuous: building sum insured, commission ratio, base premium and premium variation. Other numerical variables were only present at set levels (for example, number of occupants). The response variable was a flag for confirmed sale (excluding sales cancelled during the cool-off period) with an average conversion rate between 10% and 20%.

Further analysis (see below) highlighted the need to further segment the sample. Commercial risks represent 25% of quotes with a conversion rate 50% higher than the residential risks (75% of the data).

Models description and analysis

Following the analysis on the Casualty product, only two models have been built on the Property products: a GLM for reference and a Gradient Boosting model as a challenger.

During the build of the GLM model, it was found that the product operates across two very different markets. Policies where all the tenancies are residential present characteristics similar to a personal household risk. As a result, most insurers playing on the personal insurance market would accept these risks. On the other side, commercial tenants change significantly the risk and only a few insurers have the expertise to underwrite these properties. The result is a market with a lot less competition, higher profit margin and less price sensitivities. The differences between the two markets are such that a model benefits in being split into two rather than putting all the data together.

As the GBM has full flexibility, a single GBM was tested against the two segmented GLMs. The GBM shows a better ROC curve than the segmented GLMs. This illustrates the flexibility of the GBM which allows it to identify variations between different segments and break the link on an effect that the linear model would have carried across (unless an interaction was added). However, once the same segmentation is applied to the GBM, it is important to notice that the performance still improves: although the GBM did not need the segmentation to perform better than the GLM, the additional input is valuable and improves the model.

This illustrates the fact that machine learning algorithms do not produce the perfect model in which the computer takes all the decisions, performs the calculations and ends up with the perfect answer. There is still a significant work of segmentation and feature engineering that has the capability to improve modelling. Although one would challenge that features can be created automatically and therefore could be automated, the number of transformation involving one or more variables that can be created is astronomic compared to the number of useful ones. To illustrate the diversity of transformations, we can look at some examples of successful features created in various models. These range from simple operations (addition, multiplication, ratios…) to more advanced elements with scoring or spatial information. A list of examples is provided to illustrate the complexity.

- Sum of variables - sums insured (building, fixtures, furniture)
- Sum of variables with different weight – risk weighted sums insured (stock of alcohol, tobacco…)
- Ratio of variables - average salary by employee covered, historical claim frequency
- Scoring – flood risk with frequency and severity of events, riskiest driver
- Statistical test - credibility of the historical claim frequency
- Spatial information – travel time between risk address and home address

While the operations could be applied to any pair of continuous variable (and the variable they generated), the more advanced feature engineering are harder to automate as, for example, a score can take any form. While the computing power takes on the challenge of generating and testing a near infinite number of combinations, human insights are still valuable and a thorough review of the models is required on a regular basis. In between reviews with a specific segmentation and set of features, machine learning can provide a reliable and quick answer.

In the current modelling practice, the challenge of the model is usually done at the same time as the calibration when limitations are identified (via, for example, segments under fitted or correlation / interactions between features). The fact that the calibration is performed by the computer should not take away the opportunity of improvement. The review of the model and feature designs needs to be an independent step from the calibration with specific focus. The design stage is not a requirement for the recalibration but should be added to the maintenance work of the model. Once or twice a year for a model that could be refreshed on a weekly basis would still deliver significant saving in terms of workload.

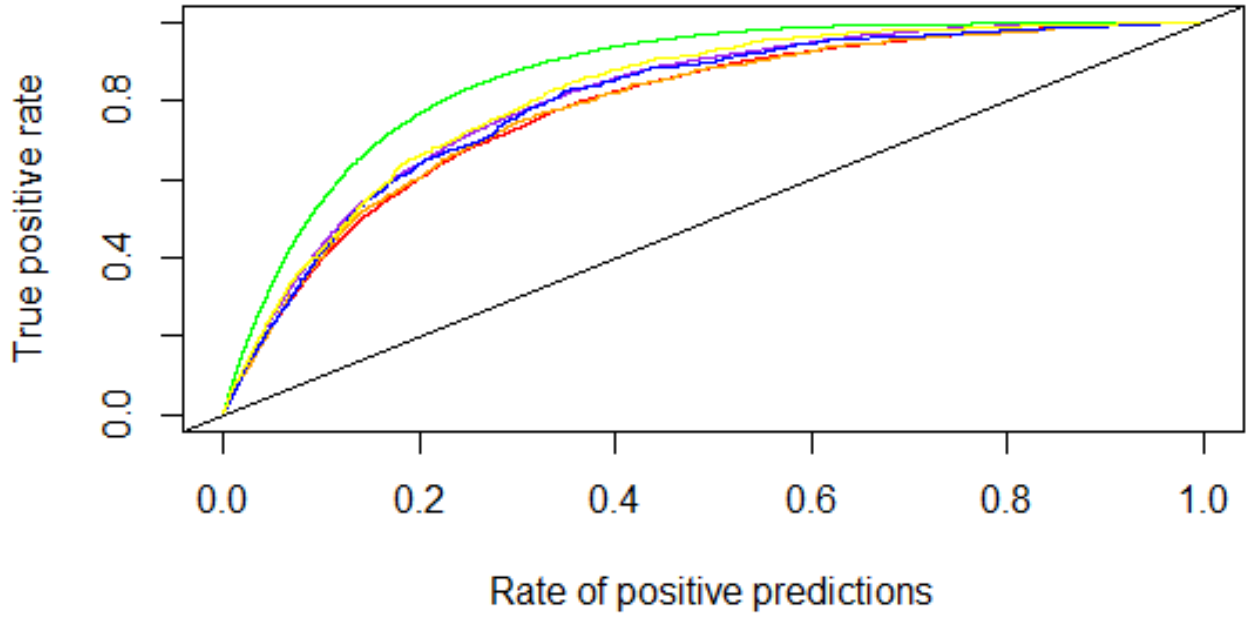


Figure 7 - ROC curves

Red = Segmented GLM in-sample
Orange = Segmented GLM out-sample
Purple = GBM in-sample
Blue = GBM out-sample
Green = Segmented GBM in-sample
Yellow = Segmented RF out-sample

On the graph above, the segmented GLMs are represented in red (in sample) and orange (out sample). The GBM without segmentation is represented in purple and blue for the in and out samples. Finally, the segmented GBM is represented in green and yellow for the two samples.

There is a significant gap between in sample and out sample on the segmented GBM model. This could be a sign of over-fitting which is dangerous for a model. Despite this, the segmented GBM out sample is the best performer of all the out samples. Although the performance is good and the model can be used, it suggests some caution should be taken. The gap between in sample and out sample is an example of how differently the model operates on different samples. As the reliability of the prediction fluctuates between samples, making decisions on the model output can be more risky. Features that are deemed extremely predictive on one sample can be less convincing on another one. As an optimisation algorithm will look for this kind of niches and irregularities to maximise the business outcome, confidence in the model to identify the true drivers of conversion is mandatory. From this example, the model seems to be the best predictor of quote conversion but the full and correct identification of the cause of the conversion is still to be proven.

VI) Conclusion

On each comparison performed, the machine-learning algorithms have been able to compete with the industry standard Generalised Linear Models in terms of performance. The implementation and analysis of the results have highlighted that the machine-learning algorithm can be produced and maintained with significantly less resources which make them adapted to the original task of providing market pricing tools for Commercial Lines. However, this comes with a price on interpretability of the models insights. This leads to the conclusion that machine-learning algorithms have a role to play but cannot, in the current state, completely replace GLMs in providing market insights. In addition to this general proof of concept, some valuation lessons have been derived from the implementation of the two algorithms.

The Gradient Boosting Machine has shown some good results on the prediction front, outperforming the GLM in both cases studied. The ease of implementation in R and speed at which it can be calibrated makes it a good contender to set the mark for expected predictive power. The existence of criteria to stop the learning phase though cross validation is a significant help to the implementation. The algorithm has shown resilience to the number of variable without signal being inputted so it is useful at identifying variables of interest. As highlighted by the improvement achieved by segmenting, GBM can still benefit from human intelligence in segmenting and feature engineering. Within a regular process, times need to be allocated regularly to review segmentation and features. On the limitation side, the workings of a GBM model are difficult to analyse which means little help for investigating and improving the model. The diagnosis tools available in R are just good enough to validate trends but not strong enough to challenge or indicate shortfalls in the model. Further work is required to improve these diagnostic tools to widen the use of the model. Due to the possible irregularities in the prediction of the model, as this is not constraint by a linear evolution, using the outcome of the model for direct mathematical optimisation would lead to prices making full use of these model imperfections. Until more experience is gained on the stability of the prediction, GBM can be used to support market pricing decisions rather than making them.

The Random Forest algorithm has out-performed the industry standard, although its use has been more problematic. In comparison to the GBM which has displayed good resilience to the quality of the input, the Random Forest prediction was drastically weakened by the large quantity of meaningless input. Despite the prediction being affected, its score was still valid and could give correct prediction after transformation. A Random Forest can be more resilient to low signal features if the prediction classes are balanced. However, this requires some sampling preparation which makes the implementation more difficult but should be the next development if one wishes to pursue further with this algorithm. Its structure of trees with all the same weight offers more possibilities of investigations into the link between variables. On the speed of implementation, the algorithm does not benefit from any boosting and is slow to converge. The parallelisation can help remediate to this but at the cost of losing information related to the Out-Of-Bag error. In an era of Cloud processing and large scale database, growing the trees for the Random Forest has become increasingly easy for equipped insurers. Overall, the Random Forest has a good predictive power and offers possibilities of investigation but requires further investment in setting up the parallelisation and a proper sampling to rebalance the prediction classes.

The Generalised Linear Models have still some advantages ahead of the two machine learning algorithms considered in this paper over the interpretability of the results. The full transparency of its working and the ability to show the effect of each factor ensure the buy-in of the business. In addition, it is possible to verify, sense check and moderate each finding before implementation. However, despite the insight gained from the machine-learning algorithms, it was not possible to improve the GLM further and therefore the GLM lagged behind in terms of predictive power. This is possibly due to the linear constraints imposed on the model while the conversion process has no obvious reason to be linear.

While the machine-learning algorithms reviewed are not able to fully replace the GLM for all conversion analysis, they have shown better performance than the GLM for modelling conversion. With more work needed to ensure that the models are producing the right answers for the right reasons, the proof of concept to deliver predictive models at a fraction of the maintenance cost can be considered a success. In addition of saving resources, the models have shown a marginal improvement in performance that could deliver an advantage needed in the competitive personal lines.

Appendix

R code for simulation of estimator

```
p0 = 0.4
p1 = 0.396

simul = function(n) {
  moment1 = 0
  moment2 = 0
  test=0


  for(i in (1:n)){
    for(j in (1:n)){
      prob = exp(
        i * log(p0) + lchoose(n,i) + log(1-p0)*(n-i)
        +  j * log(p1) + lchoose(n,j) + log(1-p1)*(n-j) )
      value = j / (i+1) - 1
      moment1 = moment1 + prob * value
      moment2 = moment2 + prob * (value **2)
      test = test + prob
    }
  }
  return(list(moment1 * 100,(moment2 - moment1**2)**(0.5) * 100))
}
```

TRM conversion - machine learning

Matthieu Bergère

Thursday, May 19, 2016

This document covers the investigations done in R on machine learning algorithms applied to conversion models for the Acturis TRM product.

Data source

```
## IN-sample data
data_source = read.csv("c:\\temp\\GB conversion.csv")


# OUT-sample data (DO NOT USE)
data_test_raw = read.csv("c:\\temp\\GB conversion OUT.csv")
```

All character strings in this import are transformed into factors (numerical values with a label on it - similar to EMBlem treatment of the data). This reduces the size of the database as numbers are easier to store than character strings. As the OUT sample is imported on its own, the coding for each level of a factor might differ (especially as one level does not appear in the out sample). The lines below correct it for the company status factor.

```
# Correct the mapping of the levels of SRC_COMPANY_STATUS
data_test_raw[,'SRC_COMPANY_STATUS']=factor(as.character(data_test_raw[,'SR
C_COMPANY_STATUS']),levels(data_source[,'SRC_COMPANY_STATUS']))
```

Now, we can trim the data to only the columns that will be needed in the model. Then, we can generate the formula of the model which is the response variable “converted” as a function of all other column left in the database.

```
select_col = c([…edited..], 'converted')
data_training = data_source[,select_col]
data_test = data_test_raw[,select_col]


# Create formula for model
response_column <- which(colnames(data_training) == "converted")
formula <- as.formula(paste0("converted ~ ",
                             paste(colnames(data_training[, -response_colum
n]),collapse = " + ")
))
```

GLM models

To understand the added value of the machine learning algorithms, we need to compare to the classic approach of GLM. Two versions of a GLM model have been produced, the second one after some feedback from the comparison with machine learning. Here, we import the outcome of the modelling done in EMBlem.

```
# GLM model - (done in EMBlem - second version after correction of the obvi
ous)
AK_CONV_IN1 = read.csv("c:\\temp\\AK conversion.csv")
AK_CONV_OUT1 = read.csv("c:\\temp\\AK conversion out.csv")
AK_CONV_IN2 = read.csv("c:\\temp\\AK conversion 2.csv")
AK_CONV_OUT2 = read.csv("c:\\temp\\AK conversion out 2.csv")
```

Gradient Boosting Method

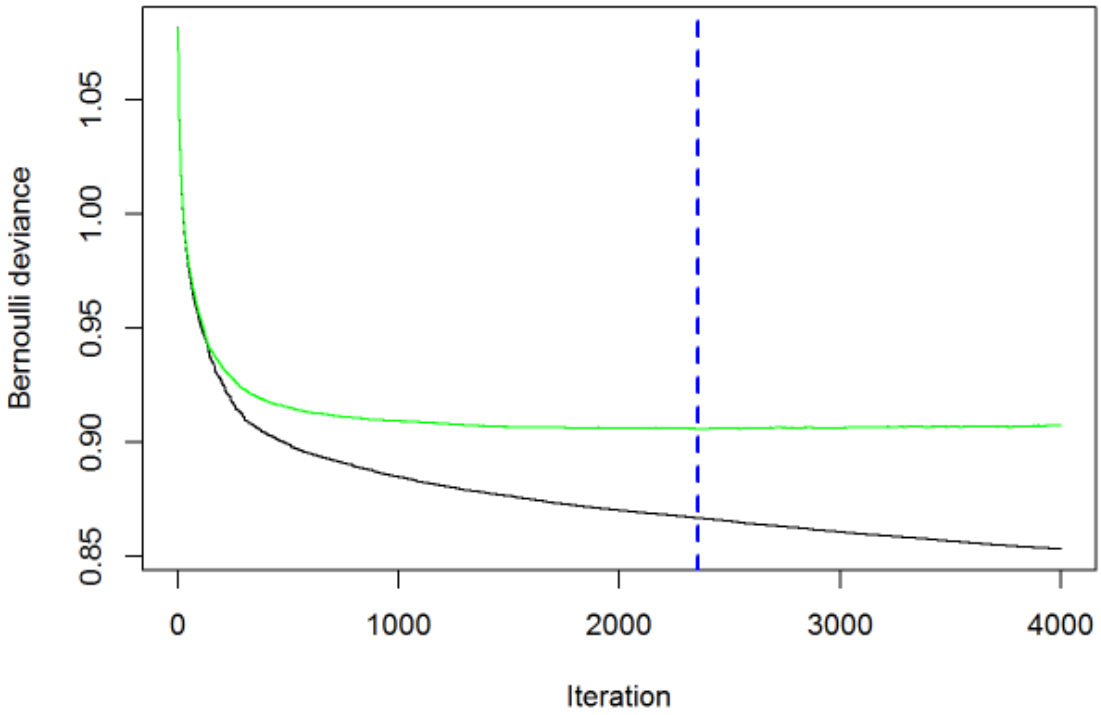
Training

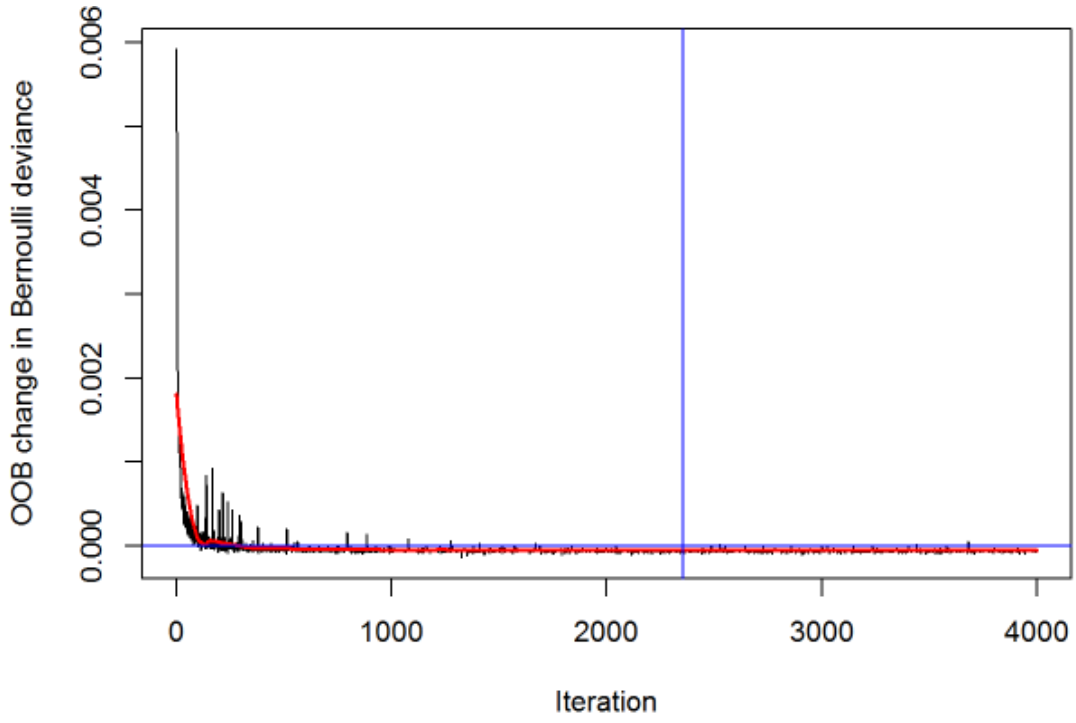
```
require(gbm)
## GBM model
# Training
temp = Sys.time()
gbm_model_R <- gbm(formula,
                   data_training,
                   distribution = "bernoulli",
                   n.trees = 4000,
                   bag.fraction = 0.6,
                   cv.folds = 5,
                   interaction.depth = 2,
                   shrinkage = 0.1)
# running time
Sys.time() - temp
## Time difference of 13.36257 mins
#save(gbm_model_R,file='c:\\temp\\model GBM CV')
#load('c:\\temp\\model GBM CV')
```

Parameters optimization

Once the GBM is generated, the optimal number of trees to use need to be defined. The plots for the Out-of-Bag error and cross validation illustrate the choice (here, I decided to use the cross validation one)

```
gbm_perf <- gbm.perf(gbm_model_R,oobag.curve=TRUE,overlay=FALSE,method='cv'
)
```





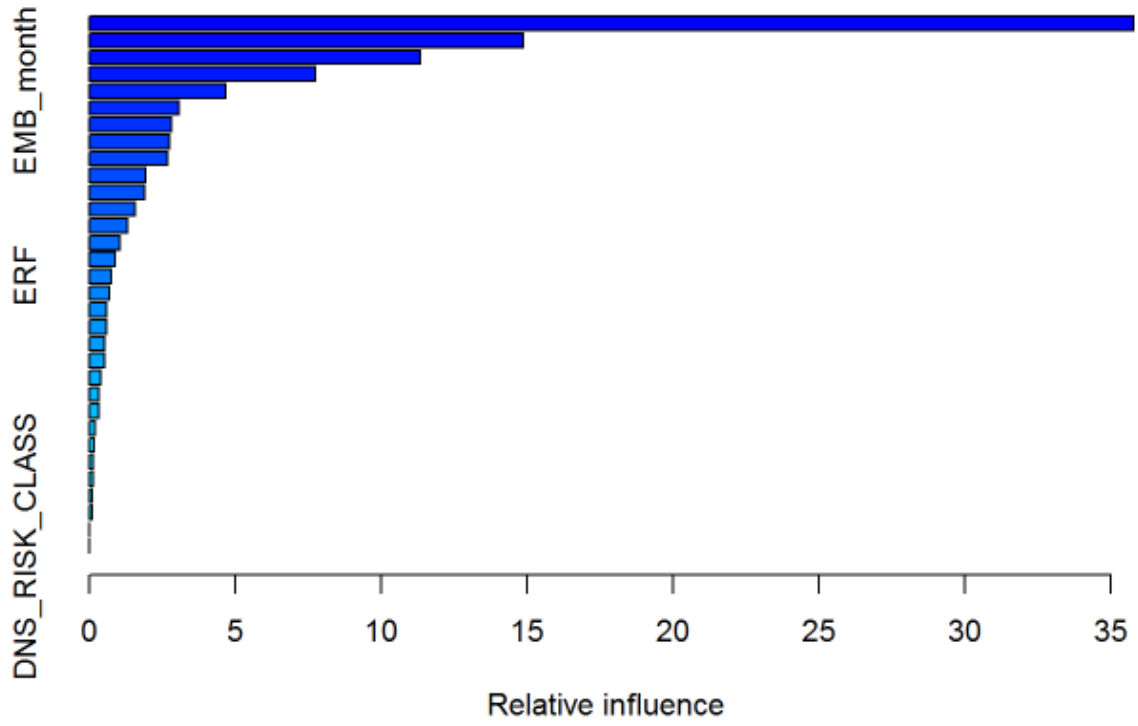
```
# Number of trees used for the model
gbm_perf
```

```
## [1] 2357
```

Analysis

We can now look at the importance of each variable in the GBM to compare with the GLM choice of variables.

```
summary.gbm(gbm_model_R,n.trees=gbm_perf)
```

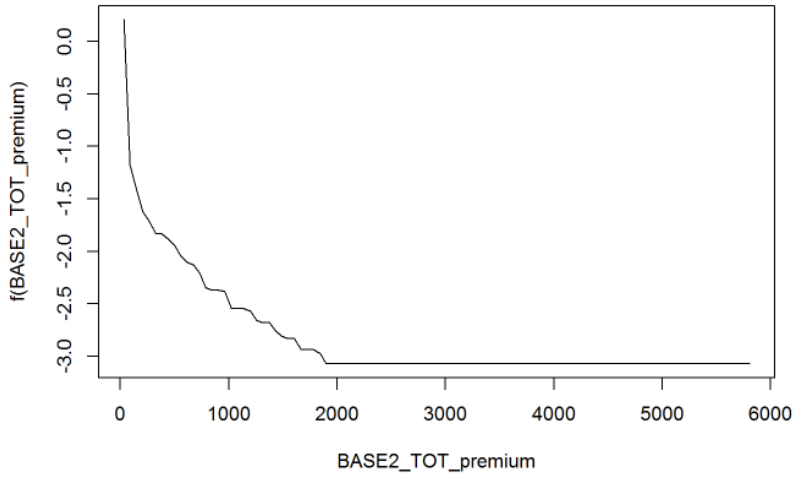



```
##                                          var      rel.inf
## EMB_occ_JJ                       EMB_occ_JJ 35.781845748
## INS_BUS_PORTAL                 INS_BUS_PORTAL 14.875387086
## commision_pc                     commision_pc 11.339187086
## BASE2_TOT_premium       BASE2_TOT_premium  7.732266644
## EMB_month                         EMB_month  4.664923205
## PL_number_covered         PL_number_covered  3.074623931
## premium_change               premium_change  2.817685771
## EL_RATED_AREA_2014       EL_RATED_AREA_2014  2.758944291
## RATING_VERSION_REF       RATING_VERSION_REF  2.694405920
## LOAD_max_height             LOAD_max_height  1.920292300
```

```
## ANNUAL_TURNOVER                       ANNUAL_TURNOVER  1.891677217
## commision_flex                         commision_flex  1.561651500
## SRC_OVERRIDE_TYPE                   SRC_OVERRIDE_TYPE  1.315266019
## PL_RATED_AREA_2014                 PL_RATED_AREA_2014  1.039112898
## EMB_Premium                               EMB_Premium  0.876450323
## ERF                                               ERF  0.755440607
## SRC_COMPANY_STATUS                 SRC_COMPANY_STATUS  0.696987427
## OWN_T_cover                               OWN_T_cover  0.606856480
## PAI_cover                                   PAI_cover  0.581004704
## LOAD_max_depth                         LOAD_max_depth  0.536289614
## EMB_Family                                 EMB_Family  0.525582289
## ELI_cover                                   ELI_cover  0.396861060
## HIP_cover                                   HIP_cover  0.341774083
## PLI_SI                                         PLI_SI  0.337649215
## LOAD_gas_work                       LOAD_gas_work  0.214108607
## OWN_S_cover                               OWN_S_cover  0.180120609
## PREMIUM_OVERRIDE_PERCENT2 PREMIUM_OVERRIDE_PERCENT2  0.137272247
## LOAD_electric                       LOAD_electric  0.118199844
## OWN_P_cover                               OWN_P_cover  0.110794791
## CAR_cover                                   CAR_cover  0.109772956
## BEQ_cover                                   BEQ_cover  0.007565526
## DNS_RISK_CLASS                     DNS_RISK_CLASS  0.000000000
```

We can also look at a marginal effect of each variable. The code below illustrate this for the base premium.

```
plot.gbm(gbm_model_R,'BASE2_TOT_premium',n.trees=gbm_perf)
```



Results

For the model comparison, we can create two vectors with the prediction of the GBM on the IN and OUT samples.

```
# Predict responses

pred_gbm_IN <- predict(gbm_model_R, n.trees = gbm_perf,newdata = data_train
ing[, -response_column], type = "response")

pred_gbm_OUT <- predict(gbm_model_R, n.trees = gbm_perf,newdata = data_test
[, -response_column], type = "response")
```

Random Forests model

Data preparation

The Random Forest algorithm takes a factor as a response (in the classification form) and cannot handle well any missing data. As a result, some recoding of the input data is required:

```
# Change response variable to factor type for classification algorithm + co
rrect missing EL area

data_training_RF = data_training

data_training_RF[,response_column]=as.factor(data_training_RF[,response_col
umn])

data_training_RF <- within(data_training_RF,

                           EL_RATED_AREA_2014 <- ifelse(!is.na(EL_RATED_ARE
A_2014),EL_RATED_AREA_2014,26))

data_test_RF <- within(data_test,

                           EL_RATED_AREA_2014 <- ifelse(!is.na(EL_RATED_ARE
A_2014),EL_RATED_AREA_2014,26))
```

Training

The training of the Random Forest can be done on a single core with the OOB error calculated or on parallel cores to speed up the process. The first code shows single core with an OOB output every

```
require(randomForest)

temp = Sys.time()

model_rf <- randomForest(formula,

                         data=data_training_RF,

                         ntree =1000,

                         nodesize = 25,

                         replace = TRUE,
```

```
                        do.trace = 200
)
## ntree      OOB      1      2
[…edited…]
Sys.time() - temp
## Time difference of 5.350168 mins
```

The second code use a distributed processing over 4 cores to create a larger forest.

```
require(foreach)
require(doSNOW)
cluster <- makeCluster(4,type="SOCK")
registerDoSNOW(cluster)


temp = Sys.time()
model_rf_par2 <- foreach( ntree = rep(1250, 4), .combine = combine, .packag
es = "randomForest") %dopar%
  randomForest(formula,
               data=data_training_RF,
               nodesize = 25,
               replace = TRUE,
               ntree =ntree)


Sys.time() - temp
## Time difference of 8.972297 mins
stopCluster(cluster)
```

Parameters optimization

Only two parameters have been looked into for this first application of Random Forest: number of trees and minimum observations by node. The first one is a fairly simple cut as the larger number of trees, the better the model. The second one is unfortunately difficult to decide: tests have been run on 400, 50, 25 and 1. With a 400 minimum observation, the models stay high level and struggle to compete with other models on the IN sample. As soon as reduced to 50, the model is performing better but with an OOB error increasing on the first 1000 observations. The model with 1 minimum observation creates some really complex trees that takes time to generate, space to store (7Gb) and more time to access. Although the model is extremely predictive on the IN sample (as expected), the OOB does not drop significantly to justify such a large model. A middle ground solution of 25 is displayed in this markdown. Other factors that could have been considered are the bagging fraction (with or without replacement) and the sampling of features, especially as a significant number of features were not predictive as highlighted by the analysis and calibration sections.

Analysis

As per the GBM model, it is interesting to see which variable has been picked up by the Random Forest. As the algorithm select only a limited number of features at each step and build a tree out of these, it is expected to have a wider selection of variables.

```
# Diagnosis - High level
resultRF = importance(model_rf_par2)
as.data.frame(resultRF[order(resultRF, decreasing=TRUE),])
```

```
##                            resultRF[order(resultRF, decreasing = TRUE), ]
## EMB_occ_JJ                                                     1172.86648
## commision_pc                                                    879.28958
## BASE2_TOT_premium                                               876.09220
## INS_BUS_PORTAL                                                  868.80584
## ANNUAL_TURNOVER                                                 495.37806
## EL_RATED_AREA_2014                                              423.71364
## EMB_month                                                       409.24830
## premium_change                                                  389.49250
## PL_RATED_AREA_2014                                              389.00642
## EMB_Family                                                      371.20784
## EMB_Premium                                                     308.78104
## LOAD_max_height                                                 287.53369
## commision_flex                                                  249.62795
## PL_number_covered                                               199.64873
## SRC_COMPANY_STATUS                                              143.72343
## ERF                                                             136.28210
## PLI_SI                                                          132.77013
## LOAD_max_depth                                                  111.72518
## PAI_cover                                                        84.53695
## RATING_VERSION_REF                                               83.28499
## OWN_T_cover                                                      72.85331
## ELI_cover                                                        61.43458
## DNS_RISK_CLASS                                                   43.73351
## SRC_OVERRIDE_TYPE                                                42.49894
## HIP_cover                                                        40.48185
## LOAD_gas_work                                                    31.24385
## PREMIUM_OVERRIDE_PERCENT2                                        31.01828
## OWN_P_cover                                                      28.48283
```

```
## LOAD_electric                                    25.79236
## CAR_cover                                        24.25237
## OWN_S_cover                                      23.38030
## BEQ_cover                                        13.40995
```

Calibration

We can now have a look at the prediction of the Random Forest

```
pred_RF_IN <- as.data.frame(predict(model_rf_par2,data_training_RF[,(colnames(data_training) != "converted")], type="prob"))[,2]
mean(pred_RF_IN)
## [1] [… edited… - result only half the original data]
mean(data_training$converted)
## [1] [… edited …]
```

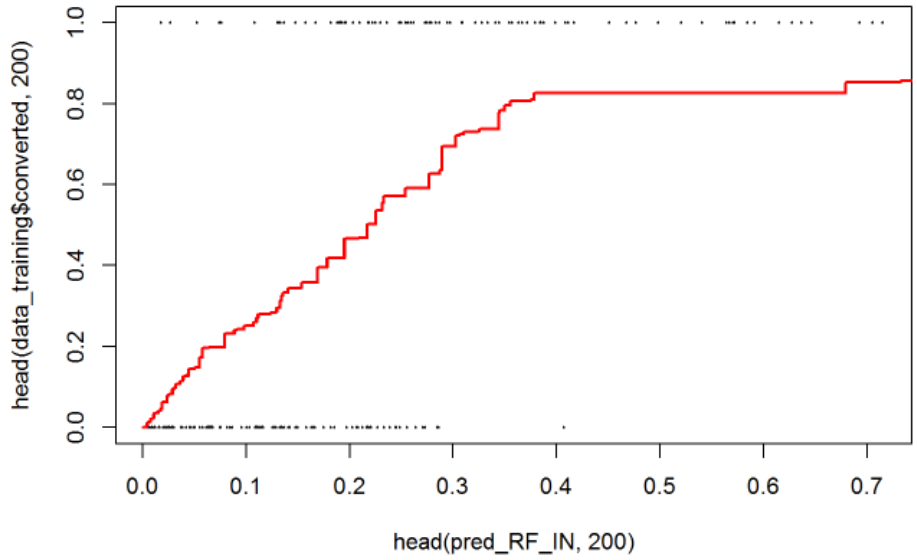
As it appears, the mean of the prediction is quite far from the the mean of the training sample. There was a sign of something similar when looking at the OOB during the training phase: the OOB error is similar to the conversion rate. This shows that the random forest is behaving in a "lazy" mode: classify everything as not converted and having errors of the same scale as the conversion rate rather than trying. There are 2 situations known to create this behaviour: when the cases to be modelled are rare and when a significant number of non-predictive features are in the model. Trees grown out of non-predictive features are unlikely to be predictive and would therefore always vote for the majority class. This leads to a dilution of the votes of the predictive trees and a bias toward non conversion. Despite this bias, the order of scores is still valid and therefore the ranking of the model is useable. To recalibrate the probability, I suggest an isotonic regression: this regression uses monotonous step functions to recreate any shape of relation between the RF score and the actual probability.

```
# Calibration of probability
require(isotone)
prob_calibration = gpava(pred_RF_IN,data_training$converted)
mean(data_training$converted)
## [1] [… edited … same number as below]
mean(prob_calibration$x)
## [1] [… edited …]
```

To use this calibration on the OUT sample data, it can be simplified to find all the breaking points of the step functions and only keep the breaking points

```
# Simplify calibration by finding breaking points
cal_simplified = data.frame(x=pred_RF_IN,y=prob_calibration$x)
cal_simplified = cal_simplified[order(cal_simplified$x),]
cal_simplified = cal_simplified[!duplicated(cal_simplified$y),]
```

```
# Visualisation of probability calibration
cal_visual = data.frame(x=pred_RF_IN,y=prob_calibration$x)
cal_visual = cal_visual[order(cal_visual$x),]
plot(head(pred_RF_IN,200),head(data_training$converted,200),cex=0.2);
lines(cal_visual$x,cal_visual$y,lwd=2,col=2)
```



Having created a link between the RF score and the conversion rate, it is possible to come up with the vector of our predictions.

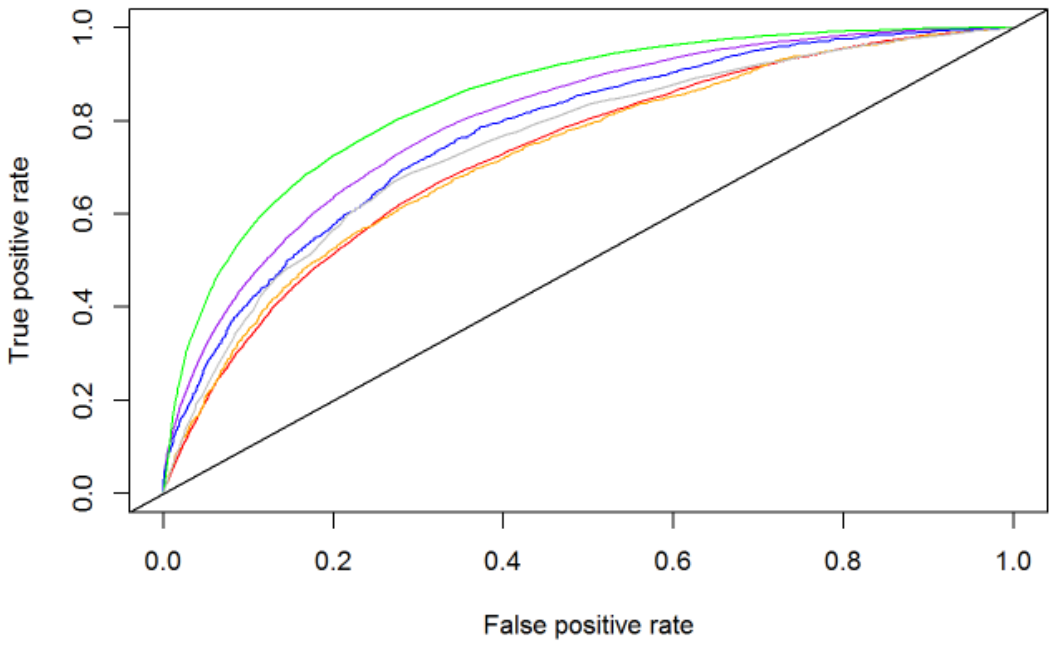
Results

```
# Predict responses with recalibration
pred_RF_IN <- as.data.frame(predict(model_rf_par2,data_training_RF[,(colnam
es(data_training) != "converted")], type="prob"))[,2]
index = findInterval(pred_RF_IN,cal_simplified$x)
pred_RF_IN_cal = cal_simplified$y[index]
pred_RF_OUT <- as.data.frame(predict(model_rf_par2,data_test_RF[,(colnames(
data_training) != "converted")], type="prob"))[,2]
index = findInterval(pred_RF_OUT,cal_simplified$x)
pred_RF_OUT_cal = cal_simplified$y[index]
```

Models comparison

In this section, we will be looking at the ROC curves to understand the performance of each model. For this, we create one object with our prediction and the correct label associated to it.

```r
## Model comparison
require(ROCR)
model_checks =list(
  pred = list(AK_CONV_IN2$ADAM_CR,
              AK_CONV_OUT2$ADAM_CR,
              pred_gbm_IN,
              pred_gbm_OUT,
              pred_RF_IN_cal,
              pred_RF_OUT_cal),
  label = list(data_training[,response_column],
               data_test[, response_column],
               data_training[,response_column],
               data_test[, response_column],
               data_training[,response_column],
               data_test[, response_column])
)
pred_group = prediction(model_checks$pred,model_checks$label)
perf = performance(pred_group, measure = 'tpr', x.measure = 'fpr') # ROC cu
rve
plot(perf,col=list('red','orange','purple','blue','green','grey'))
abline(a=0, b= 1)
```



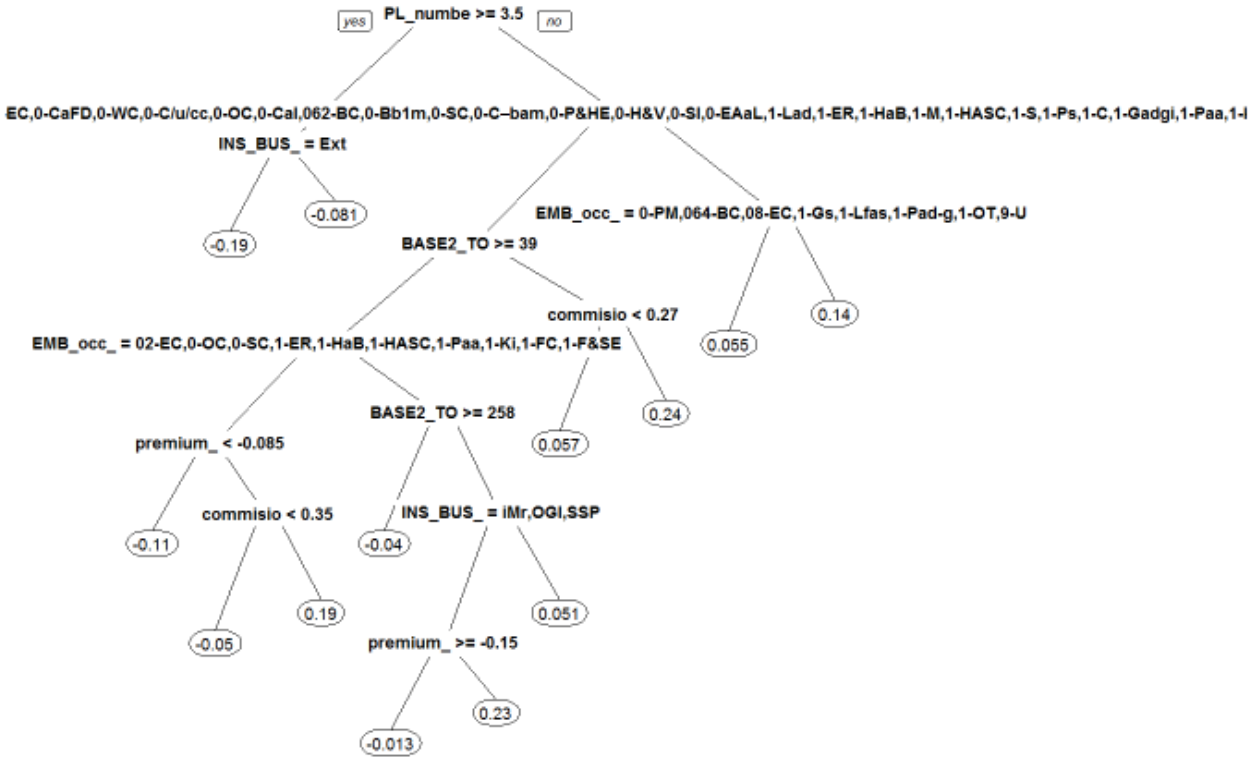
The best performing model on the ROC curve is the Random Forest on the IN sample (green). This is an over-optimistic view as shown by the OOB error. The performance on the OUT sample (grey) is more representative of the performance of the model on unseen data (more in line with the OOB). This is however still out-performing the GLM, both on IN and OUT samples (red and orange respectively). The GBM shows the best predictive power on the OUT sample (blue) despite a loss compared to the IN sample (purple).

Use of machine learning algorithm

Improving classic modelling - validation and error spotting

Then, we can compare the differences between the GBM and the first cut GLM model

```
require(rpart)
require(rpart.plot)
data_prep = as.data.frame(cbind(data_source,AK_CONV_IN1,pred_gbm_IN))[,c('pred_gbm_IN','ADAM_CR',select_col)]
data_prep$results = data_prep$pred_gbm_IN - data_prep$ADAM_CR
index = !(colnames(data_prep) %in% c('pred_gbm_IN','ADAM_CR','converted'))
data_prep = data_prep[,index]
formula_comp <- as.formula(paste0("results ~ ",
                                  paste(colnames(data_prep[, -(ncol(data_prep))]),collapse = " + "))
))
diff = rpart(formula_comp,data_prep)
prp(diff)
```



The tree shows a few elements:

- The number of people was not fitted in the original GLM
- Occupation could be reviewed more accurately
- Commission banding putting 25% to 35% may not be adapted as 27% comes as a breaking point (revealed afterward to be linked to a specific broker charging 27.5%)

Publish the markdown

```
require(rmarkdown)

render("c:\\temp\\TRM markdown.Rmd", output_file = "C:\\temp\\TRM documenta
tion.html")
```