Dávid Burka – László Kovács – László Szepesváry

Modelling MTPL insurance claim events: Can machine learning methods overperform the traditional GLM approach?*

DÁVID BURKA
Assistant Professor
Corvinus University of Budapest,
Hungary
Email: david.burka@uni-corvinus.hu

LÁSZLÓ KOVÁCS
Assistant Lecturer
Corvinus University of Budapest,
Hungary
Email: laszlo.kovacs2@uni-corvinus.hu

LÁSZLÓ SZEPESVÁRY
PhD Student
Corvinus University of Budapest,
Hungary;
Chief Actuary
Magyar Posta Életbiztosító Zrt. (Hungarian Post Life Insurance Company),
Hungary
Email: szepesvary.laszlo@mpb.hu

Pricing an insurance product covering motor third-party liability is a major challenge for actuaries. Comprehensive statistical modelling and modern computational power are necessary to solve this problem. The generalised linear and additive modelling approaches have been widely used by insurance companies for a long time. Modelling with modern machine learning methods has recently started, but applying them properly with relevant features is a great issue for pricing experts. This study analyses the claim-causing probability by fitting generalised linear modelling, generalised additive modelling, random forest, and neural network models. Several evaluation measures are used to compare these techniques. The best model is a mixture of the base methods. The authors' hypothesis about the existence of significant interactions between feature variables is proved by the models. A simplified classification and visualisation is performed on the final model, which can support tariff applications later.

KEYWORDS: motor third-party liability pricing, interpretable machine learning, claim frequency modelling

* *Dávid Burka* participated in this study within the framework of the GINOP-2.2.1-18-2018-00010 project of Corvinus University of Budapest, titled 'Automated, life situation based, real time decision support framework'.

Motor third-party liability (MTPL) insurance covers the risk of causing damage to a third party with a motor vehicle. The liability is owned by the faulty driver, but the MTPL insurance transfers the financial risk from the individual to the insurance company. While CASCO- (casualty and collision) type constructions pay insurance benefits to the owner of the insured car, MTPL insurance pay to third parties, called claimants.

The types and amounts of damage can spread over a wide range. Property damage and bodily injuries are typically separated. Claims from small crushes during parking to large accidents with many broken cars and other damage to the environment can also cause property damage. Serious claims usually couple with bodily injuries varying from suffering minor bruising to becoming disabled or deceased in an accident. In some cases, the claimant loses the ability to work or lifetime extra costs are incurred due to the accident, indicating a demand for life-long annuities. The financial consequences of the latter examples can be enormous: multi-million-euro claims are possible.

Consequently, it is not surprising that in many countries it is compulsory to have MTPL for vehicle owners. Vehicle insurance (i.e. relevant legislation, insurance coverage, and structural specifics) may vary from country to country. The modelling described in this study is based on Hungarian data, and the application is made for the Hungarian MTPL structure; however, the results can be applied to other countries as well.

1. MTPL pricing and tariffs in general

The risk of causing an accident with a motor vehicle (in other terms, having an MTPL claim) depends on several factors related to the driver's profile and the characteristics of the insured car. For example, young drivers with high-performance cars or drivers living in suburbs with high annual mileage usually cause claims more frequently, so they are more risky drivers according to MTPL insurances. Therefore, the price of the MTPL insurance should be differentiated based on several factors for which the use of statistical modelling is essential in the pricing process.

Some important variables are described below that can influence the claim-causing risk, hence, they are usually tariff elements of MTPL insurance. As powerful

cars are viewed as riskier, the performance (kW) and the cubic capacity (cc) of the vehicle are usually taken into consideration in pricing. Insurance companies have their own data about risks, and published tariffs show that the manufacturing year, brand, and fuel type of the car also influence the claim-causing probability. Various characteristics of the driver, namely, age, residence, driving experience, and expected annual mileage are also important and can be potential tariff factors as well. Finally, the bonus-malus class of the policyholder must also be mentioned. The bonus-malus system is used to classify the drivers according to their individual claim history. If a driver does not have a claim in one year of insurance coverage, it results in a forward step in the system. In case of causing a claim, a negative step is applied. The rules of these systems are usually defined by country laws and insurance companies incorporate the bonus-malus class as a tariffing factor. The above list is not complete, and other influencing variables can also be used. The results of this study verify the use of several mentioned variables.

In the pricing process, the goal of statistical modelling is to determine the connection between the mentioned variables and MTPL claims. Usually, one-year long periods are analysed with the premium also being calculated for one year. The following two features are essential for profiling insurance contracts (*Ohlsson–Johansson* [2010]):

– claim frequency: the number of claims divided by the in-force years of insurance coverage (cumulative data of individual policies),
– claim severity: the total claim amount divided by the number of claims.

The pure premium, known as the net premium, is the product of the claim frequency and the claim severity. This amount covers the cost of claims of the group in the analysed period (per policy and per year). The net premium, however, does not contain other expenses and profits of the company, which indicates the next step of pricing, called gross premium calculation.

Statistical modelling can usually be performed separately for claim frequency and claim severity, with the goal of determining the dependencies with the tariffing factors (variables). For such problems, regression models are often used in the statistics. According to *Ohlsson* and *Johansson* [2010], linear regression is not suitable for this problem because the assumed normally distributed errors are not reasonable for either the pieces or the amounts of insurance claims. Multiplicative models usually fit the insurance claim experience better than linear ones. A generalised linear model (GLM) can be a more sophisticated framework that can handle the aforementioned problems. Instead of the normal distribution, GLM uses a more general class

of distributions and a more general dependency structure can be modelled among variables than in the case of simple linear regression (*Ohlsson–Johansson* [2010]).

It is not surprising that GLMs and the slightly more general, generalised additive models (GAMs; discussed later) are frequently used in MTPL pricing; similar techniques are applied worldwide for insurance tariff calculations. There are specialised software programs for actuarial pricing that are capable of this type of modelling and defining the optimal MTPL tariff structure according to the chosen model.

GLMs and GAMs provide a general and widely applied framework for tariff optimisation; however, owing to the fixed probability distributions and the lack of automatically handled interactions, they carry the risk of underfitting. Therefore, it can happen that the GLMs and GAMs do not program results in the best model for later claim experiences. This study analyses the possibility of providing a better fitting approach with machine learning (ML) methods, such as random forests (RFs) or neural networks (NNs). This is a new approach in insurance pricing that has been examined by some authors and has been applied by some companies; however, the widespread use of these techniques is not yet common in the sector.

It is important to note that the understanding and formalisation of the models can also be important viewpoints since the main application is tariff making. For example, in Hungary, the MTPL tariffs must be published by companies, so black box algorithms are not applicable. This study only analyses claim-frequency modelling; other elements of tariffing could form part of future research.

2. Data and applicable data transformation for claim modelling

The data used for the analysis were derived from a Hungarian non-life insurance company's MTPL database. Contractual and claim data from two calendar years[1] were used as the initial database. Contractual data were identified using contract numbers. The start, end, and cancel dates of the contract, car data, driver data, and discount data (discounts the policyholder demands from the premium tariff) were collected. Claim data were identified using the claim number linked to the contract number. Data about the date of the claim and the date of the reporting, and data about the total of the claims (divided into paid and outstanding parts) were collected.

The most important variables are listed below from the total 20 features, with their abbreviations in italics. Some data were distorted, and values of some variables

[1] The calendar years are not disclosed due to business secrecy.

were exchanged because of business-secret considerations; however, this did not influence the application of the models.

The target variable for our analysis is binomial: the policy caused claim in the analysed policy year or not (binomial) – *claimBool*. This target variable will be predicted by selected ML models that consider several client-, policy-, and vehicle-related factors:

– age group of the client (multinomial) – *AgeGroup*,
– region code of residency (multinomial) – *RegionCode*,
– actual bonus-malus class of the client (multinomial) – *BM*,
– performance (kW) class of the car (multinomial) – *KWclass*,
– brand of the car (multinomial) – *VehicleType*,
– cubic capacity (cc) of the car (numeric) – *VehicleCubicCapacity*,
– fuel type of the car (multinomial) – *VehicleFuel*,
– manufacturing year of the car (numeric) – *VehicleManufactureYear*,
– sales channel of the contract (multinomial) – *VehicleSalesChannel*,
– demanding child discount (binomial) – *Child*,
– demanding multiple cars in family discount (binomial) – *MultipleCarsinFamily*,
– demanding experienced driver discount (binomial) – *ExperiencedDriver*.

In the case of some variables, it was useful to transform them into a limited range. Categorical variables, such as the region codes, were classified into higher-level regions, and the age of the customer was transformed into age groups. These classifications were done in line with the company's analysed tariff practices.

In the initial dataset, applications of several data cleansing methods were needed. The missing and unrealistic values make up less than 0.3% of the observations; these policies were excluded from the dataset. Extreme outliers were filtered via Tukey's outer fences as a cut-off value (upper quartile ± 3 * interquartile range) (*Abzalov* [2016]).

The goal of the net premium calculation is to determine the expected yearly claim volume of a contract which can be considered as the net yearly price of the insurance (without the company's costs and profit). Consequently, for further application, contractual and claim data must be transferred to a yearly basis as follows: the first year of insurance coverage lasts from the start date of the contract to the first anniversary, the second year lasts from the first to the second anniversary, and so on. These periods are called policy years.

The contractual data used were cut into policy-year pieces per contract, so in the transformed database, each record contains data about one policy year of a

contract. Policy data can change from one policy year to another, for example, bonus-malus classes and age are constantly changing, which must be handled to have an unbiased database for pricing. Records with policy years starting in a fixed calendar year were used for later analysis.

After obtaining the policy-year-based contractual database, claim data can be joined to it. It must be examined with the help of claim number and claim date, whether a policy caused a claim in a policy year or not. If no claim corresponds to the period, then the actual policy year record obtains a claim figure of zero, otherwise, the corresponding claim data are shown.

It can also occur that, in the case of a policy, the number of elapsed years is not an integer (e.g. because the observation period is over or the policy was cancelled). The insurance risk for a period shorter than a year is naturally lower than the risk for a whole year, so the length of the period from a policy year spent in insurance coverage must be taken into account in the premium calculation. This is referred to as a yearly unit. It is assumed that insurance risk is linearly proportional to yearly units in the case of periods shorter than one year.

With the application of these transformations, we obtained a dataset of more than 200,000 observations and more than 20 variables for the examined period. The database described above is a suitable tool for claim-frequency modelling.

3. Methodology for the claim modelling analysis

This section describes the methodology for the analysis: the applied supervised learning models, feature selection, and evaluation methods.

3.1. MTPL claim modelling as a supervised learning problem

ML is a group of algorithms and statistical models that can perform a specific task without using explicit instructions, relying only on patterns and inference (*Neal* [2007]). It is seen as a subset of artificial intelligence. ML algorithms build a mathematical model based on sample data, known as 'training data', to make predictions without being explicitly programmed to perform the task (*Koza et al.* [1996]).

ML tasks are classified into several broad categories. In supervised learning, the algorithm builds a mathematical model from a set of data that contains both the inputs and the desired outputs (*Russell–Norvig* [2010]).

In the premium calculation for MTPL insurance, the most important factors for the net premium are the probability of a policy to cause claims in the insurance peri-

od and the expected total of the claims. Determining whether a policy causes claim in a policy year is an analogous problem to object recognition in images or filtering emails, as in all of these cases, we have a well-specified output target variable that is binary: Does the image contain the object? Is this email a spam? Does this policy cause claims? Similar problems arise in other areas of the insurance business as well. For instance, *Boodhun* and *Jayabalan* [2018] also apply supervised learning methods for underwriting in life insurance.

Binary output labels or target variables can be predicted using a classification model, which is a type of supervised learning. The standard output for classification models is a probability vector that gives the $\mathbf{P}(Y = True)$ probabilities for a Boolean Y target variable based on a set of feature variables for the observations (*Mohri–Rostamizadeh–Talwalkar* [2012]). Therefore, if we encode the claim information for a portfolio of MTPL policies into a *claimBool* variable where *True = the policy caused claim in a given year* and *False = the policy did not cause claim in a given year*, then we can easily fit a supervised learning model to this target variable based on some features of the policies. In this way, we can obtain the yearly claim probabilities for each individual policy in our portfolio, which can be utilised for individual-level pricing by the insurance company.

It is important to note that more than one claim can be made in each period. According to our data, we had only a few of this kind of event ($z = 4.61\%$ where $z = \sum claim\ events / \sum claimBool - 1$).

The first modelling trials led us to realise that using models with binomial target variables is much better than multinomial ones in terms of the goodness of fit and simplicity. In the later analysis, we used binomial models, so if the *claimBool* variable is true, it can also sign more than one claim. In the final step of tariff making, we propose adjusting the probabilities of $\mathbf{P}(claimBool = 1)$ with a $(1 + z)$ multiplicator.

There can also be claims incurred but not reported (IBNR). A similar technique can be used to handle these events in the tariff-making step by estimating the y% of IBNR claims. A simple method for estimating IBNR claim events is the chain ladder method (*Matvejevs–Malyarenko–Matvejevs* [2014]).

3.2. Description of the applied supervised learning models

In this study, we apply GLMs and GAMs that are widely used methods in actuarial claim modelling. We also utilise two other ML methods (RF and NN models), which are not yet widespread among insurance companies, but in recent literature, there are some promising results in the sector. Most notably, *Noll*,

Salzmann, and *Wuthrich* [2018] showed on French MTPL claim data that a simple GLM does not capture interactions of feature components appropriately, whereas tree-based methods and NNs are able to address these interactions more successfully.

Our goal is to test the robustness of *Noll*, *Salzmann*, and *Wuthrich* [2018] results on claim data from a very different region and to propose a stacking or voting model that provides a better fit than the individual ones. A technique for obtaining the results of a more complicated supervised learning model is also introduced. Further methodological techniques for later analysis are also described in this section.

3.2.1. GLMs and GAMs

In selecting the exact classification models to apply, we first considered the most recent literature on the subject. Supervised learning is commonly applied by actuaries when estimating claim probabilities with respect to some feature variables of the GLM. Some recent examples of the application of GLMs in MTPL pricing are mentioned by *Kafková* and *Krivánková* [2014], *Giancaterino* [2016], and *Henckaerts et al*. [2018]. General insurance applications of GLMs are demonstrated by *Gray* and *Kovács* [2001], in addition to a motor claim analysis with application for mortality modelling.

In this subsubsection, we provide a summary of the main characteristics of GLMs. For a broad introduction to GLMs, we refer to *Harris*, *Hilbe*, and *Hardin* [2014]. The main attribute of GLMs is the generalisation of the probability distribution of the target variable. GLMs extend the framework of linear regression models with a normal distribution to the class of distributions from the exponential family. It allows the modelling of many variable types (counts, frequencies, etc.). A link function makes a connection between the mean and a linear function of the feature variables. The link function $g(\mu)$ is a monotonic differentiable function of the form $g(\mu) = \mathbf{x}'\boldsymbol{\beta}$ where $\mathbf{x}'$ is the vector of feature variables and $\boldsymbol{\beta}$ is the vector of regression parameters. For our target variable with a binomial distribution, we use the logit link function $g(\mu) = \ln(\mu/(1-\mu))$.

In a binomial distribution $\mu = \mathbf{P}(Y = True) = p$ from the GLM with a logit link, we can get the desired probabilities for claim events by $p = \exp(\mathbf{x}'\boldsymbol{\beta}) \Big/ (1 + \exp(\mathbf{x}'\boldsymbol{\beta}))$. Thus, the only parameter of the binomial distribution to be estimated is the $\boldsymbol{\beta}$ vector. This can be solved through maximum likelihood estimation with the application of the Newton-Raphson method for numerical optimisation, introduced by *Harris*, *Hilbe*, and *Hardin* [2014].

The major appeal of GLMs for actuaries is that marginal feature effects can be obtained very easily by obtaining the values from the $\exp(\boldsymbol{\beta})$ vector. This information can be used in pricing to set premium multipliers based on the features of the policies. Furthermore, parameter significance can also be easily tested using a classic Wald test.

The performance of a GLM, however, in predicting claim event probabilities can be low, as using only linear combinations of features can be restrictive. Therefore, we should extend the GLM framework to consider non-linear effects and interactions as well. Interactions can be represented in a GLM by multiplying the variables that we wish to interact with. Interactions in the case of categorical features can be considered via cross-products between dummy variable encodings. In contrast, dealing with non-linear effects of continuous features leads to GAMs, introduced by *Hastie* and *Tibshirani* [1987].

For the estimation of the smooth functions for each feature, we utilise the thin plate spline framework proposed by *Wood* [2003] as this solution does not require the choice of the number and form of basis functions in the non-linear spline modelling. For the implementation of the GAMs, we applied the *mgcv* (mixed GAM computing vehicle) R package developed by *Wood* [2017].

3.2.2. RF method

In the very recent literature, some promising applications of tree-based ML models in insurance pricing have appeared. We were mainly inspired by the results of *Henckaerts et al*. [2019], who applied RF models for claim event prediction of Belgian MTPL policies with great success. We aimed to test whether the RF method's good prediction performance was preserved on our Hungarian data.

Decision trees partition data based on yes-no questions can predict the same target value for each member of the constructed subsets. This value is usually the mean of the target variable in a subset. In our case, as we average a Boolean target, this method results in an empirical $\mathbf{P}(Y = True) = p$ estimate. A popular approach to construct decision trees is the classification and regression tree (CART) algorithm, introduced by *Breiman et al*. [1984]. In the CART algorithm, two parameters control the construction of a tree. The *maxdepth* parameter sets the maximum depth of any node of the final tree with the root node counted at a depth of zero. The complexity parameter (*cp*) informs the program that any split which does not improve the fit by *cp* will likely be pruned off by cross-validation, hence, the program does not need to pursue it. A lower *cp* results in more complex decision trees.

RFs (*Breiman* [2001]) are ensemble techniques that combine multiple decision trees. They reduce the variance of a single tree by averaging the forecasts of multiple trees on bootstrapped samples of the original data. This stabilises the prediction and improves the predictive performance compared with a single decision tree.

The number of used trees (T) was chosen to be 500 at first and increased further until an additional decrease in the cross-validated loss function was achieved (*Breiman* [2001]). In our study, we apply a loss function, called deviance, as suggested by *Venables* and *Ripley* [2002]. The deviance is defined as $D\left(y,\hat{y}\right)=-2\ln\left[L\left(\hat{y}\right)/L\left(y\right)\right]$ likelihood ratio where $L\left(\hat{y}\right)$ is the model likelihood and $L\left(y\right)$ is the likelihood of the saturated model (i.e. the model with the number of parameters equal to the number of observations). For competing models of fit, the best model obtains the lowest deviance value on the holdout data. The holdout data were selected via 10-fold cross-validation in our study. In 10-fold cross-validation, the original sample was randomly partitioned into 10 equal-sized subsamples. Of the 10 subsamples, a single subsample was retained as the validation data for testing the model and the remaining nine subsamples were used as training data. The cross-validation process was then repeated 10 times, with each of the 10 subsamples used exactly once as the validation data.

The CART algorithm was implemented in the *rpart* (recursive partitioning and regression trees) R package developed by *Therneau et al*. [2015]. The *randomForest* package, developed by *Liaw* and *Wiener* [2002], is readily available to fit RFs for standard regression and classification problems.

The RF algorithm does not make any assumptions regarding the relationship between features and targets, which usually results in good prediction capabilities. However, it was not possible to obtain the marginal effects of each feature.

We can still rank the features according to their importance in the model's prediction. We can measure the importance of a specific feature x_l in a decision tree by summing the improvements in the loss function over all the splits on x_l. We normalise these variable importance values such that they sum to 100%, giving a clear idea about the relative contribution of each variable in the prediction. We can generalise this approach to ensemble techniques by averaging the importance of variable x_l over the different trees of the RF.

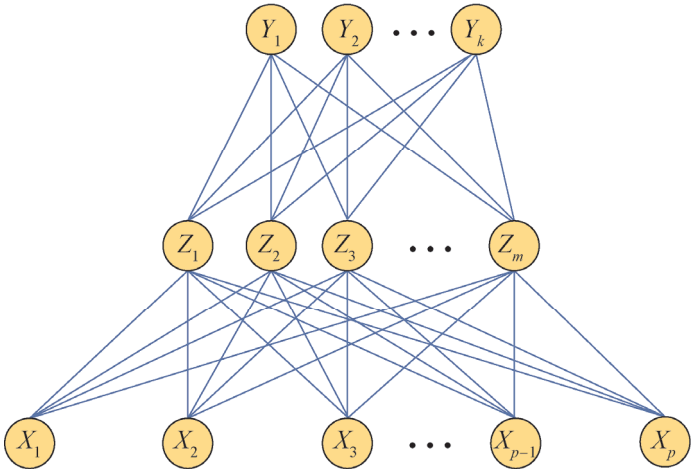
3.2.3. NN models

The next type of supervised learning model to be applied is NNs. These kinds of models are becoming increasingly popular for solving image, handwriting, and other pattern-recognition tasks (*Lecun–Bengio–Hinton* [2015]). They also had

some successful applications in estimating claim probabilities for MTPL policies on US (United States) data, especially with pay-as-you-go solutions (*Yeo* [2011]). *Boodhun* and *Jayabalan* [2018] successfully applied NNs for life insurance underwriting. The ability of NNs to handle successfully complex patterns between feature sets and the target might also be useful in detecting complicated interactions between our features that other models might miss. (For a detailed description of NNs, see *Hastie–Tibshirani–Friedman* [2009], *Hajek* [2005].)

A multi-layer perceptron (MLP) is used for introducing the concept of NNs for a K-class classification problem where the k^{th} target class is represented by a dummy variable Y_k, but the principle is the same for a single Boolean target variable. (See Figure 1.) Derived or hidden features Z_m are created from linear combinations of the inputs (**X** is the matrix of the p feature variables) with a sigmoid link function $\left(\sigma(v) = 1/\left(1 + e^{(-v)}\right)\right)$. The target Y_k is modelled as a GLM of the Z_m variables with a logit link function.

Figure 1. A schematic single hidden layer NN



Note. $X_1, \ldots, X_p$ are input variables, $Z_1, \ldots, Z_m$ are derived or hidden features, and $Y_1, \ldots, Y_k$ are the dummy variables representing the categories of the target variables.

Source: *Hastie–Tibshirani–Friedman* [2009].

The NN model has unknown parameters called weights (the coefficients of the GLMs that define the network), and we seek values for them that make the model fit the training data well. We used a backpropagation algorithm for the training (*Hastie–Tibshirani–Friedman* [2009]).

NNs are quite sensitive to the unit of measurement of continuous features (*Hajek* [2005]). To handle this, every continuous feature is normalised to a (0, 1) scale before applying the NN model.

The most important parameters of MLPs are the number of hidden layers and the number of latent variables in each layer. There is no well-defined or heuristic algorithm for the choice of network structure. Usually, two or three competing network structures are tested on a separate validation set, and the one that yields the minimal prediction error is chosen (*Hastie–Tibshirani–Friedman* [2009]).

The exact marginal effects of the features cannot be expressed even in a single hidden layer network because of the large number of interactions between the feature variables and the latent variables in the hidden layers. However, we can obtain a measure of predictor importance by recording, during training, how much the contribution of feature x_l to the loss-function gradient updates in an average iteration during the run of the backpropagation algorithm (*Hajek* [2005]).

We apply an R implementation of NNs in the *mlp* function from the RSNNS (Stuttgart Neural Network Simulator in R) package, developed by *Bergmeir* and *Benítez* [2012].

3.3. Feature selection

To boost model performance by avoiding overfitting, we execute feature selection. As in NNs and RF models, the marginal effects of features cannot be determined. We apply a simple heuristic algorithm called recursive feature elimination (RFE). RFE, described in Algorithm 1, only requires an importance measure to be assigned to each feature. Both RFs and NNs can provide a measure of variable importance in prediction.

The algorithm has an optional step (line 4.4), where the predictor rankings are recomputed in the model based on the reduced feature set. *Svetnik et al*. [2004] showed that, for RF models, there was a decrease in performance when the rankings were recomputed at every step. For this reason, in our case, line 4.4 was not executed for RFs. We skip line 4.4 for NNs as well, because its nature is similarly complex to that of RFs. Furthermore, according to *Svetnik et al.* [2004], the recalculation can only improve performance in cases where the initial rankings are not adequate, like in the case of GLMs or GAMs with highly collinear features, as the Wald test has a biased *p*-value estimate (which is used for feature ranking in these models).

We use our own implementation of RFE in R language to skip the step at line 4.4.

Algorithm 1 (*Kuhn* [2011])

1. Tune/train the model on the training set using all predictors.
2. Calculate model performance.
3. Calculate variable importance or rankings.
4. For each subset size S_i, $i = 1, \ldots, p$, **do**
4.1 keep the S_i most important variables,
4.2 pre-process data (optional),
4.3 tune/train the model on the training set using S_i predictors,
4.4 recalculate the rankings for each predictor (optional).
end
5. Calculate the performance profile (which is deviance in our case) over the S_i.
6. Determine the appropriate number of predictors.
7. Use the model corresponding the optimal S_i.

In the case of GAMs, we can take advantage of the marginal effects of the features from the models. With this extra information, we can use more sophisticated solutions for feature selection in GLMs. We chose to apply sure independence screening (SIS), as it is the most suitable for large datasets with a large number of features or with $n > 10^5$ (*Fan–Lv* [2018]). The SIS method for feature selection is presented in Algorithm 2.

Algorithm 2

1. Let d be the number of all possible features in the GLM. Let m be the number of features to be retained in a SIS iteration. m should be chosen based on the number of records in the data matrix and the RAM capacity of the computer.
2. Estimate the model containing all d features.
3. Obtain the Wald test statistic for each feature denoted by $\hat{L}_j$ for the j^{th} feature.
4. Determine the $\hat{\mathbb{M}} = \left\{ j \in [\,1, d\,] : \left|\hat{L}_j\right| \text{is among the top } m \text{ ones} \right\}$ set.
5. Estimate the model using only the features from $\hat{\mathbb{M}}$ with a lasso-like regularisation method (*Tibshirani* [1996]). This estimates the coefficients of irrelevant features to be zero.

6. Form the $\hat{\mathbb{F}}$ set by selecting the features with non-zero coefficients in the model estimated in step 5.

7. Estimate the model containing all features from $\hat{\mathbb{F}}$.

8. Repeat steps 2–6 until the elements in $\hat{\mathbb{F}}$ do not change.

We used the SIS algorithm's implementation in the R package (see also *Saldana–Feng* [2018]).

3.4. Methodology for model evaluation and comparison

While examining the most recent literature on non-life insurance pricing with supervised ML models, we found that the models were only assessed through the deviance measure. This is the main methodology of evaluation in *Henckaerts et al.* [2018], [2019]; *Yeo* [2011]; *Kafková–Krivánková* [2014]; *Giancaterino* [2016]; *Verbelen–Antonio–Claeskens* [2018]; and *Noll–Salzmann–Wuthrich* [2018]. The listed authors compared deviances on a separate test set or used an average deviance from a 10-fold cross-validation. However, there are some other approaches used in the literature: supervised learning models are fitted on French MTPL data and model performance is analysed with concentration and Lorenz-curves by *Denuit*, *Sznajder*, and *Trufin* [2019].

Comparing the deviances of different models works well when the task is to compare different setups for the same model family and to select the best alternative. For example, deviance is an appropriate measure for selecting the number of trees in RF models, the interactions to use in GAMs, or the number of hidden layers in NNs. However, the deviance does not illustrate how well the $\mathbf{P}\left(Y = True\right) = p$ probabilities obtained from the ML model can be used to predict $\left\{Y = True\right\}$ events.

Furthermore, the deviance is unable to compare different kinds of models, as the saturated model is always interpreted in the given GAM, RF, or NN framework. Therefore, the deviance enables us to only compare different parameterisations of the same model family, as the deviance from the saturated model in the case of GAMs or RFs can be on a completely different likelihood ratio scale, even on the same dataset.

We propose evaluation measures that are commonly used to evaluate classifier models. We aim to measure on a uniform, model family-independent scale how well the $\mathbf{P}\left(Y = True\right)$ probabilities can be used to predict actual $\left\{Y = True\right\}$ events (*Mohri–Rostamizadeh–Talwalkar* [2012]).

First, we split our MTPL policy data to separate a 20% test set. The remaining 80% will be the 'training data'. This set of data can be split further into training and validation sets if the ML algorithm demands parameter value selection (e.g. for

choosing the number of trees in a RF). The training data can also be split multiple times during the 10-fold cross-validation steps in the algorithms (e.g. for selecting spline function parameters in GAMs).

The point is that the 20% test set that has been completely disregarded by the ML models up until now will only come into play during the evaluation phase. We create the $\mathbf{P}\left(Y = True\right) = p$ prediction vectors from every trained model. As we assumed earlier, the insurance risk is linearly proportional to yearly units in the case of periods shorter than one year, and the probabilities obtained from the models were adjusted with the inverse of the average yearly unit. This technique is in line with that of *De Jong* and *Heller* [2008]. In the following sections, we will show and use this type of probability. Adjustments according to more than one claim case and the IBNR cases ([1 + y] and [1 + z]) multiplicators) were not used here; these can be done at the later tariff calculation.

We define cut-off values in such a way that $\mathbf{P}\left(Y = True\right) = \mathbf{P}\left(claim = 1\right) >$ cut-off $\rightarrow claim = 1$ is set for the policy. After predicting 0 and 1 values for each policy based on the probabilities and the cut-off value, we can construct the so-called confusion matrix by grouping the actual and predicted classifications in a 2 × 2 matrix. (See Table 1.) Naturally, the cut-off value influences the values in the confusion matrix.

Table 1

Structure of a confusion matrix

Actual value	Model value	
	0	1
0	*a*	*b*
1	*c*	*d*

Note. *a*, *b*, *c*, and *d* denote the number of observations in each cell.

At many cut-off points, we calculate:

– the proportion of correct classifications (true positive rate, $TPR = d/\left(c + d\right)$) and

– the proportion of misclassifications (false positive rate, $FPR = b/\left(a + b\right)$).

These values are used to create a curve, called the receiver operating characteristic (ROC) curve, in a coordinate system. The principles of the ROC curves and confusion matrices are described based on (*Bradley* [1997]). The curve's *x* axis represents the *FPR* values and the *y* axis represents the *TPR* values at the cut-offs used. The cut-off values used are the percentiles of the empirical distribution of the $\mathbf{P}(claim = 1)$ vector from each evaluated model.

It is easy to see that the perfect classifier will be at the (0, 1) coordinate as this model would have 0 *FPR* and 1 *TPR*. Thus, we grade every model based on the area under its ROC curve. This measure is called the area under the curve (*AUC*). A useful model would have an $AUC > 0.5$ as for a $\mathbf{P}(claim = 1)$ vector containing 0.5 for every policy (the completely random guess model) would obtain an *AUC* of 0.5. Naturally, a higher *AUC* indicates a better classification performance.

Two other important aspects from an insurance company's point of view are mentioned that will lead us to define new measures to the claim-modelling problem:

– Misclassification in the case of contracts with claims and contracts without claims is asymmetric from the financial aspect. The company has profit on contracts without claims and usually has loss on contracts with claims, but the latter loss is usually much higher than the former profit.

– The company is mainly interested in maximising profits. However, the points of the ROC curve are only classification rates, which cannot support the profit-maximising decision in the asymmetric financial situation.

Based on these two aspects, we introduce a utility function, $U(\alpha)$ as a new assessment tool for this problem. Let us assume that the company uses the predicted probabilities and the cut-off value as an underwriting tool: if the predicted probability is not higher than the cut-off value, then the contract is accepted, otherwise, it is rejected. Thus, the company accepts $a + c$ number of contracts. (See Table 1.) Let us suppose that the company has one unit of 'profit' on each contract without claim and *L* unit of 'loss' ($L < 0$) on each contract with claim. The company's goal is to maximise $U(\alpha) = a + L \cdot c$ utility function with the optimal α cut-off value. The idea of using the general assessment function of a confusion matrix was also used by *Figini* and *Uberti* [2010].

The motivation to define $U(\alpha)$ is to model a utility that is similar to the company's expected profit. As this study only deals with claim-frequency modelling

and no other parts of the tariff calculation (e.g. claim amount, expense loadings, gross premiums), uniform premiums per contract, uniform claim amounts per claim, and uniform expenses are assumed. In this way, uniform profits and losses can be assumed among the two types of contracts. Of course, the formula can be generalised with other model elements, but at this stage of modelling, it provides a simple optimisation tool that can handle the asymmetric financial situation.

In addition to the optimal $U(\alpha)$ and α, we also show the ratio of the retained (kept) portfolio $(a + c)/(a + b + c + d)$. (See Table 1.) The defined measures can be used to realistically assess a supervised learning model's ability to detect policies that cause claim events for insurance companies, considering the financial asymmetry of an insurance company.

4. Claim-modelling results

In this section, we show the results of the described methods on our sample data and our conclusions regarding the examined models.

4.1. Exploratory data analysis

First, we examined the distribution of the policies with claims (*claimBool* = 1) and no claims (*claimBool* = 0) with respect to the values of different feature variables. The relative frequency of the policies with claims is below 5% in our data, so we need to predict an extremely rare event.

By examining the distribution of the feature values inside the claim and no-claim groups separately, we can quickly discover some important relationships between our features and the claim profile of the policies.

For example, the proportion of policyholders under the age of 30 and above the age of 70 is noticeably higher in terms of policies with claims than in the no-claim category. (See Figure 2.) Similarly, it is more likely that a driver from the capital city (Budapest) has a claim compared to a rural driver. (See Figure 3.)

After these preliminary conclusions from simple visualisation, we could run our more complex models. First, we separated a randomly chosen 80% of our data for model training, and the remaining 20% was used to calculate all the performance measures.

Figure 2. Age distribution of clients holding policies without and with claim

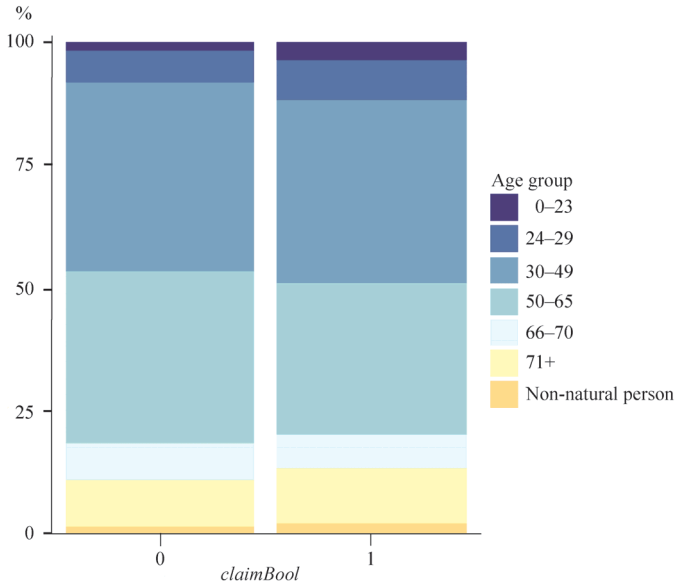
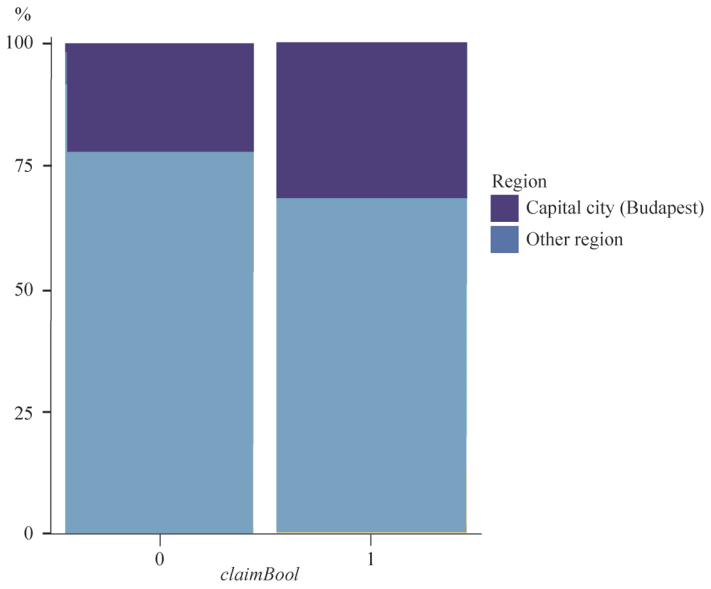


Figure 3. Geographical distribution of policies without and with claim



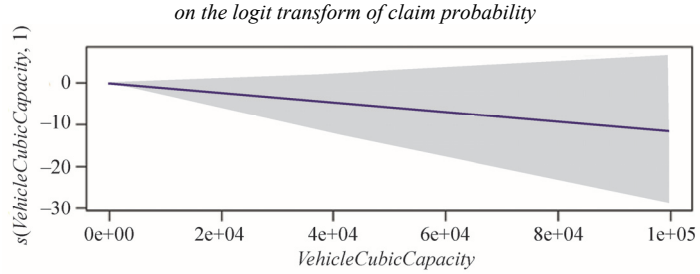
Note. *claimBool*: binary target variable denoting whether the policy caused claim in a policy year (1) or not (0).

4.2. Results of the GAM and GLM

First, we ran a GAM with all 20 features included to test for the necessity of non-linear effects. We considered interactions up until three-way combinations to avoid cases that are not present in the observed sample.

None of our continuous features had non-linear effects on the logit transform of the *claimBool* target variable. A fitted thin plate spline function for the *VehicleCubicCapacity* feature is shown in Figure 4. This feature has a linear function as its spline representation (with its 95% confidence bars marked by grey area). This can also be seen in the second parameter of the spline function *s* that denotes the degree of the basis in the spline.

Figure 4. Spline function describing the effect of vehicle cubic capacity on the logit transform of claim probability



None of the fitted spline functions significantly differs from a linear form. Therefore, further investigations were conducted in a GLM framework to improve the run time.

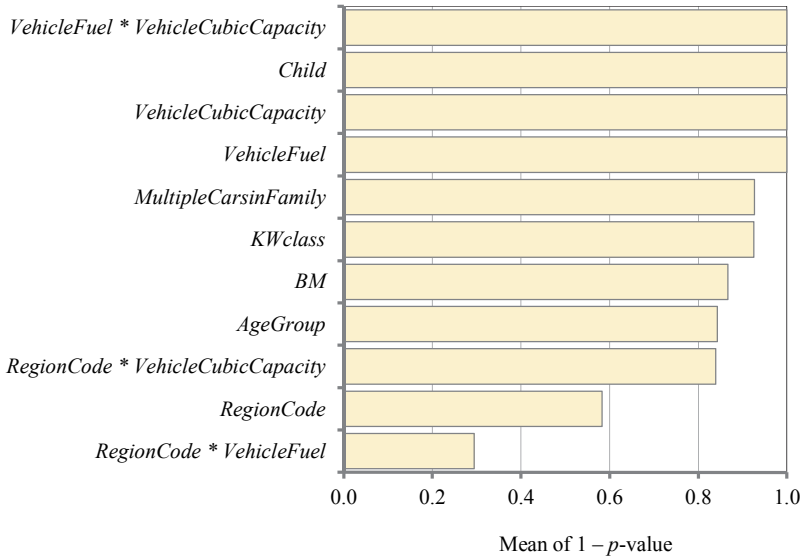
Using the iterative SIS feature selection algorithm, we obtained the following model:

$$AgeGroup + BM + RegionCode + KWclass + VehicleFuel + Child + \\ + VehicleCubicCapacity + MultiplieCarsinFamily + VehicleFuel * RegionCode + \\ + VehicleFuel * VehicleCubicCapacity + VehicleCubicCapacity * RegionCode.$$

Feature importance is defined via the partial Wald test: the smaller the *p*-value for the Wald test of the coefficient, the more significant the effect of that feature. The effects are ordered according to importance (in decreasing order by their $1 - p$-value measures), as shown in Figure 5. For categorical features or interactions containing categorical features represented by dummy variables, the means of their $1 - p$-value measures are considered.

The parameters from the model indicate that the most important predictor for claim events is the interaction between the fuel type and the cubic capacity of the vehicle (*VehicleFuel* * *VehicleCubicCapacity*). The features are also individually important, along with the feature of having a child discount.

Figure 5. Feature importance in the GLM



The model contains coefficients that are not significant according to the Wald test at even 10%, but removing the features associated with these coefficients did not improve the 10-fold cross-validated deviance of the model, so we kept the solution of the iterative SIS algorithm. The reason for this is the presence of two- and three-way interactions. For example, if we have some non-significant region codes and we remove the dummy of these codes (we merge them into the reference category), they can affect the coefficients of the interactions as well.

4.3. Results of the RF model

First, we experimented with the application of the CART algorithm with different *cp*-s. The default value in the *rpart* package is $cp = 0.01$, which results in a null model; for every policy, the tree predicts the relative frequency of claims in the training set. This is because of the low frequency of claim events in the portfolio. We gradually decreased the value of *cp* until an actual decision tree was built by

utilising some features. The *maxdepth* parameter was fixed at its maximal value of 30. However, the resulting trees become too wide rather quickly with *AUC* on the test set, which is worse than the GLM proposed in Sub-section 3.2. The detailed results are presented in Table 2. With 63 leaves, the decision tree loses its advantage of easy interpretability, and it still underperforms the GLM with interactions on the test set in *AUC*.

Table 2

AUC on the test set for CART decision trees with different cp-s

Model	Number of leaves	*AUC*
GLM	–	0.6446
CART, *cp* = 0.010000	1	0.5000
CART, *cp* = 0.000305	1	0.5000
CART, *cp* = 0.000304	13	0.5515
CART, *cp* = 0.000301	13	0.5515
CART, *cp* = 0.000300	63	0.6341

After unsuccessful trials with the CART algorithm, we applied the RF algorithm with $T = 500$ to address the challenge posed by the low claim frequency. Neither the increase nor the decrease in the number of trees improved the 10-fold cross-validated deviance.

The RFE algorithm proposed the following RF model:

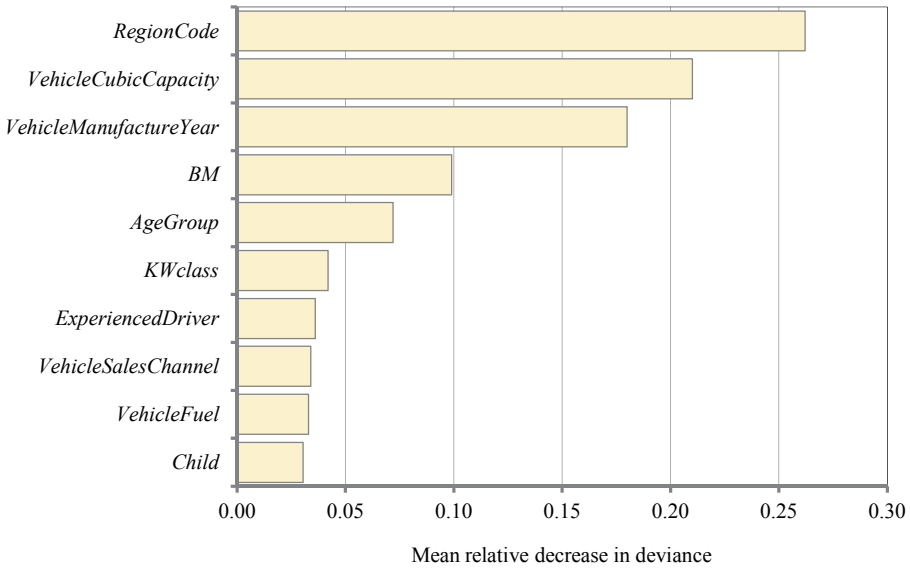
$$\begin{aligned} &\textit{VehicleCubicCapacity} + \textit{VehicleManufactureYear} + \textit{RegionCode} + \textit{BM} + \\ &+ \textit{AgeGroup} + \textit{KWclass} + \textit{VehicleSalesChannel} + \textit{ExperiencedDriver} + \\ &+ \textit{VehicleFuel} + \textit{Child}. \end{aligned}$$

In a RF model, the importance of features can be estimated by calculating the mean relative decrease they cause in our loss function (deviance in our case) during model training.

It is important to note that in the RF model, the importance of features is slightly rearranged with respect to the GLM. The most important feature of this model is the *RegionCode*, which is also included in the GLM, but not in the group with significant features at 5%. The *Child* and *VehicleFuel* features, which can be considered some of the most important stand-alone features in the GLM, rank lower in the RF model. The *MultipleCarsinFamily* Boolean feature, which was significant at 10% in the GLM, was excluded by the RFE algorithm. The RF algorithm seems to handle the continuous features (*VehicleCubicCapacity* and *VehicleManufactureYear*)

better, as they rank higher on its list. As we could not find significant non-linear main effects for these features with GAM, the RF algorithm seems to be more potent in detecting complex interactions between these features, as in the case of GAM.

Figure 6. Feature importance in the RF model

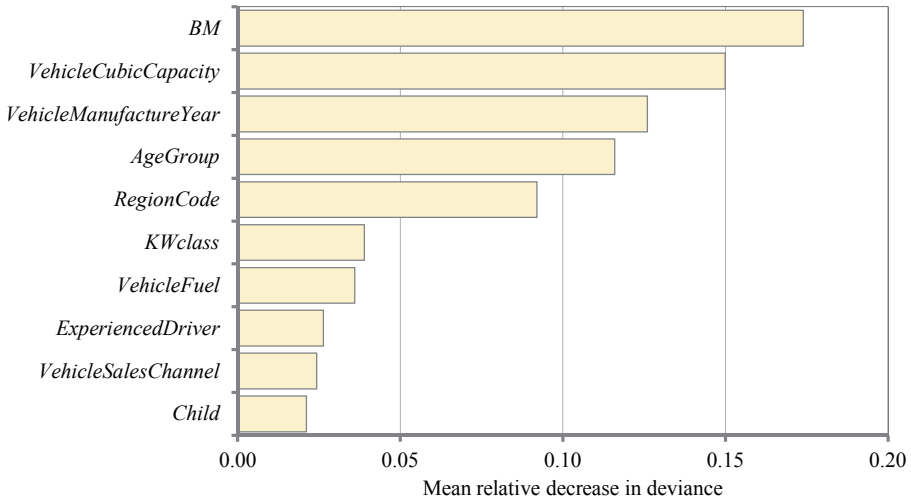


4.4. Results of the NN model

We constructed an MLP NN with three hidden layers with 64 latent variables in each layer, following the suggestions of *Lecun*, *Bengio*, and *Hinton* [2015]. The RFE algorithm suggested leaving the full set of features in the model. It is not surprising as a NN is the most efficient when it can form many connections in its hidden layers to detect complex interactions (*Hastie–Tibshirani–Friedman* [2009]). This means that the model prefers a large number of input features and a complex hidden layer structure. As a result, the so-called 'way deep learning' structures are preferred in the cases of complex pattern recognition tasks (*Lecun–Bengio–Hinton* [2015]).

The importance of features can be estimated in a manner similar to that of the RF model. We can obtain the mean relative decrease they cause in our loss function (deviance). The results are in Figure 7. Only the top 10 features are shown.

Figure 7. Feature importance in the NN model



In our NN, the *BM* class is the most important feature. However, the *RegionCode* (the most important feature in the RF model) was also high on the list. Furthermore, the NN prefers continuous features, such as RF. This suggests that the latent variables of the NN can capture more complex interactions than those in the GLM. These characteristics make the NN model more similar to RF behaviour. However, owing to the large number of input features, the NN is more likely to over-fit the original dataset.

4.5. Evaluation and extension of supervised learning models

After analysing the behaviour and feature importance in the applied models, we assessed their predictive performance in a separate test set. The *AUC* measures for each model are presented in Table 3.

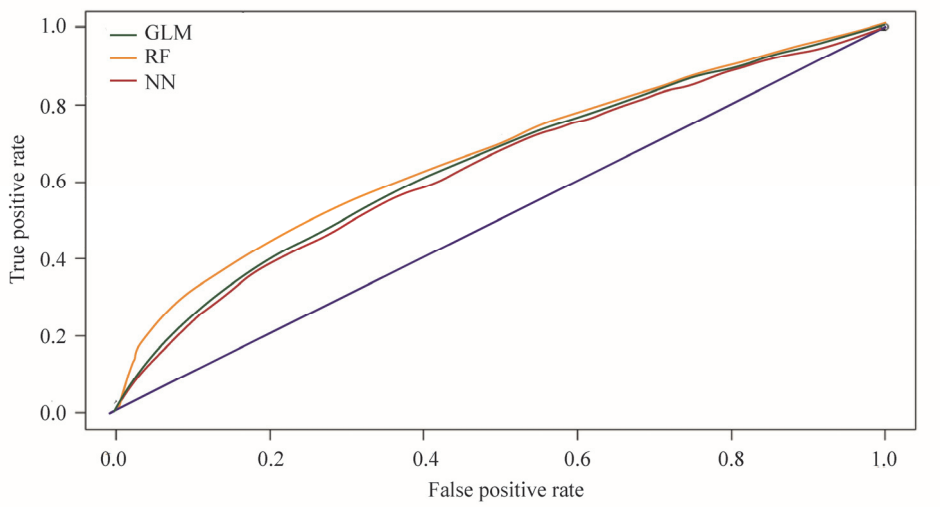
Table 3

AUC measures for the three examined models

Model	*AUC*
GLM	0.6446
RF	0.6657
NN	0.6347

From the ROC curve, we can easily observe that all three models are better than the random guess model. However, the GLM and NN are similar in performance, whereas, the RF outperforms them. This conclusion is supported by the *AUC* measures of the three models. In contrast, from the ROC curves, we can see that the advantage of the RF model disappears at lower cut-off values (the top right corner of the graph). The NN has an advantage in these regions, so it seems that this model is slightly better at estimating higher claim probabilities.

Figure 8. ROC curves for the three examined models



The respective *AUC* values are not too high for the first sight, as they do not reach the favourable 0.7 value. However, owing to the prediction of a very rare event (the relative frequency of the policies with claims is lower than 5%) and the fact that which drivers cause claim is a random incidence, it is not surprising that *AUC* is at this scale.

Overall, we can determine that the performance of the RF at certain cut-off values can be improved by considering the results from the other two applied models. Based on this conclusion, we found that it would be useful to create a voting model that averages the predicted $\mathbf{P}(claim = 1)$ probabilities from the three models with some weights. In the literature, these kinds of models are usually described as stacking models as they stack the results of different ML models (*Smyth–Wolpert* [1999]). Recently, these models have been used to identify Higgs bosons with great success (*Alves* [2017]).

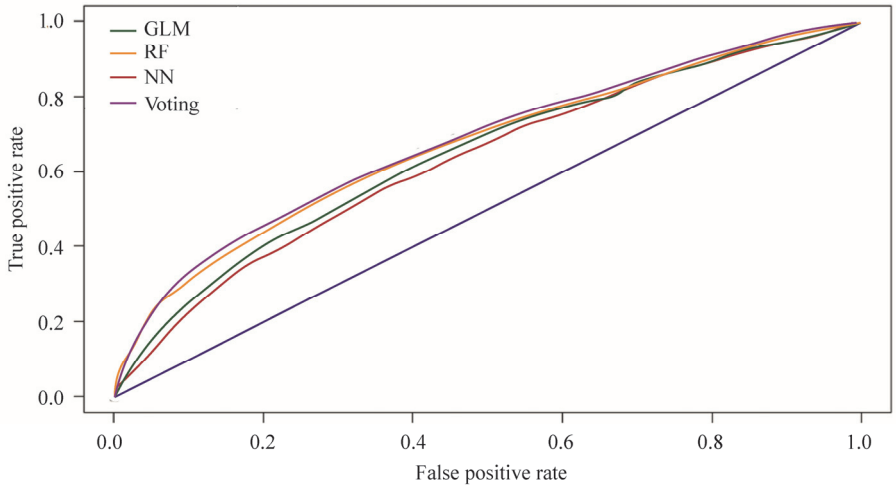
We set the weight combinations on the training set via a grid search, where each weight could take an integer value in the range of [1, 10]. The objective function was a 10-fold cross-validated deviance of the voting model. The final weight combinations were 1 *GLM* – 2 *RF* – 1 *NN*. This is not surprising, since the RF model produced the best *AUC* measure, it makes sense to give this model a higher weight in a voting of $\mathbf{P}(claim = 1)$ predictions. The *AUC* measures extended with the voting model are introduced in Table 4.

Table 4

AUC measures for the three primary models and the voting model

Model	*AUC*
GLM	0.6446
RF	0.6657
NN	0.6347
Voting	0.6791

Figure 9. ROC curve for the three primary models and the voting model



The voting model can correct the low performance of the RF at the lower cut-off points on the ROC curve, so this new model produces the best *AUC* measure.

However, because of the uniformly poor performance of these models when predicting claim events, the models could not be used as underwriting machines.

We also assessed the models using the $U(\alpha)$ utility function. The results for $L = -10, -20, -30$, and -50 are shown in Tables 5, 6, 7, and 8, respectively. The rank is based on the maximum utility value. According to the volume of the average premium per contract and the average claim amounts from public Hungarian data, $-20 \leq L \leq -10$ can be reasonable for the Hungarian MTPL business, but for assessing the models, we show some more extreme values of the L loss parameter.

Table 5

Analysis of the utility function with L = –10

Denomination	GLM	RF	NN	Voting
Maximum utility	27,978	28,894	27,936	28,675
Kept portfolio (%)	95.7	95.5	99.9	93.8
Optimal cut-off value (%)	8.4	15.0	14.0	8.7
Rank	3.	1.	4.	2.

Table 6

Analysis of the utility function with L = –20

Denomination	GLM	RF	NN	Voting
Maximum utility	16,927	18,495	16,257	19,043
Kept portfolio (%)	81.6	88.0	82.0	85.0
Optimal cut-off value (%)	5.0	8.6	4.7	5.9
Rank	3.	2.	4.	1.

Table 7

Analysis of the utility function with L = –30

Denomination	GLM	RF	NN	Voting
Maximum utility	9,046	10,422	8,297	11,009
Kept portfolio (%)	56.2	72.4	64.8	71.8
Optimal cut-off value (%)	3.4	4.5	3.3	4.2
Rank	3.	2.	4.	1.

Table 8

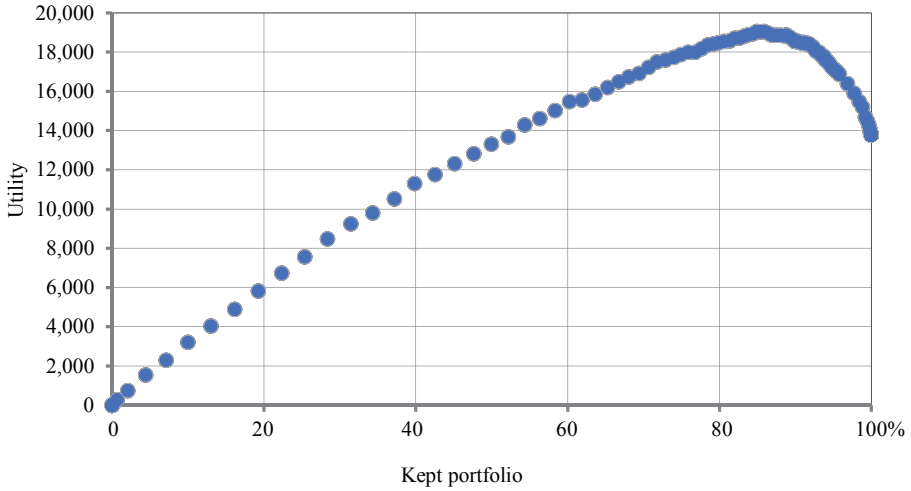
Analysis of the utility function with L = –50

Denomination	GLM	RF	NN	Voting
Maximum utility	1,174	1,440	1,540	3,023
Kept portfolio (%)	34.1	26.1	25.3	28.4
Optimal cut-off value (%)	2.6	0.7	2.3	2.1
Rank	4.	3.	2.	1.

Tables 5–8 show that the voting model produces the best results in each segment of the L loss parameter, except for the $L = -10$ case. It is important to note that the NN performs poorly in the case of moderate L values and has good results in the case of extreme L values, whereas the RF method behaves in the opposite manner. This characteristic is consistent with the tendencies of the ROC curves of the two models. The voting model has good overall results, confirming again that this is our most beneficial model.

Figure 10 shows the chart of the kept portfolio and the utility value for the voting model with $L = -20$ for different cut-off values. Eighty-five percent of the portfolio is kept, and at the optimal α value, the *TPR* is 39.8%, while the *FPR* is 14.2%. Taking into consideration the fact that claim causing is a rare and random incidence, we can state that the model can identify a significant proportion of clients with claims, in addition to incorrectly classifying too many no-claim policies.

Figure 10. Kept portfolio and utility in the case of different cut-off values (voting model, L = –20)



In general, we can declare that in the case of non-extreme values of the loss parameter, it is worth accepting almost all the policies, even with the assumed uniform premium. This implies that in such cases, the best application of the model will be to accept most of the policies but differentiate the premium according to the predicted claim probability for each policy. The possibilities of this type of tariff concept will be explained in the next section.

5. Pricing application – building an explainer model for the voting predictions

For later pricing applications, there are two main possible approaches. The first is to use the probabilities obtained from the voting model without any modification. It is possible to calculate a unique probability with the model for each combination of the feature variables, and the collection of these probabilities can provide the first pillar of the tariff. It can be supplemented later with claim amount modelling and other tariff elements. The advantage of this approach is that it maintains the total variance of the model. Disadvantages include the complex structure, lack of visualisation, and the fact that it is extremely difficult to publish or describe. There are countries in which national laws require MTPL tariff publishing. In the latter case, formalising the predicted probabilities of the model with the tariffing variables can be an important aspect, despite losing some variance. In the present section, we examine this second approach.

We can attempt to identify homogeneous groups in our policies that have similar $\mathbf{P}(claim = 1)$ predictions in order to handle these groups similarly during later pricing. For this, we must understand how each feature affects these predictions, and based on this understanding, we can try to create groups from our policies along the features that are the most important in predicting the claims. We use the $\mathbf{P}(claim = 1)$ probabilities from the voting model to create homogenous policy clusters.

For a classification with k clusters, for each group $j = 1, \ldots, k$ we determine the formula

$$\widetilde{Y}_j = \left(\sum_{i=1}^{n_j} claimBool_{ij} \right) \Big/ n_j$$

where n_j denotes the number of observations in cluster *j* and $claimBool_{ij}$ is the binary target variable for the i^{th} policy in cluster *j*. After adjusting $\tilde{Y}_j$ with the inverse of the average yearly unit, we obtain Y_j, the actual claim frequency of the group. Furthermore, we calculated the average predicted $\mathbf{P}(claim = 1)$ probability for each cluster. X_j denotes the average for cluster *j*.

The performance of each classification model was assessed using two measures. First, for each model, we calculated Spearman's rank correlation between the vectors of Y_j and X_j. In this way, we can see how well each model captures the order of the clusters according to their actual claim frequency. This is important information, as we need to give higher premiums for groups with higher $\mathbf{P}(claim = 1)$ probabilities.

Second, we calculate a variant of the sum of squared errors within our clusters (*SSW*). In this *SSW* measure, we aim to capture the variability of the binary target variable around the X_j average predicted claim probabilities. During later pricing, the tariff for each cluster can be based on the average prediction of a ML model. Therefore, it is important to see how the actual claim experience differs from the average prediction within a cluster. The *SSW* is defined by the following formula:

$$SSW = \sum_{j=1}^{k} \sum_{i=1}^{n_j} \left(claimBool_{ij} - X_j \right)^2 .$$

The smaller *SSW* values indicate that the predicted claim probabilities for the clusters better fit the actual claim experience within the clusters.

To create homogenous clusters with respect to the predicted $\mathbf{P}(claim = 1)$ probabilities from the voting model, we apply the CART decision tree algorithm to the predicted $\mathbf{P}(claim = 1)$ probabilities from the voting model with a squared error loss function. These features can be the original features that we used for the three elementary models. In this way, we can use the leaves from the decision tree to create homogenous policy clusters with respect to the predicted $\mathbf{P}(claim = 1)$ probability. We examined decision trees with maximum depths ranging from 1 to 27. The exit value is selected based on the rank correlation measure, as the tendency shows that more clusters usually result in a lower rank correlation. Figure 11 shows the details from the 1 to 15 depth case.

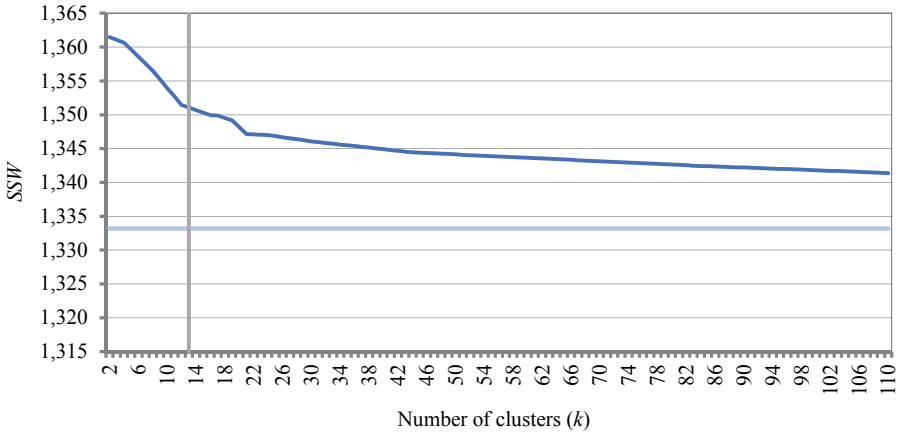
We also examined the *SSW* measure as a function of cluster number. (See Figure 12.) The horizontal line shows the sum of the squared deviations for the probabilities of the voting model without clustering. We can see that after 12 clus-

ters, which is equivalent to a tree depth of four, the decrease in the within-group variability is not considerable.

Based on the 'elbow' present at 12 clusters in Figure 12, we can consider the decision tree with depth four as optimal. This decision tree has acceptable within-cluster claim experience variability, and the predicted average claim probability of the clusters reflects the order of the actual relative claim frequencies remarkably well, moreover, the simplicity is also beneficial. Twenty-one clusters with a tree depth of nine could also be an optimal choice; however, it implies a more complex tree structure.

Figure 11. Rank correlation between actual and predicted claim probabilities on clusters made by CART on voting model results

Figure 12. SSW of predicted claim probabilities on clusters made by CART on the voting model results

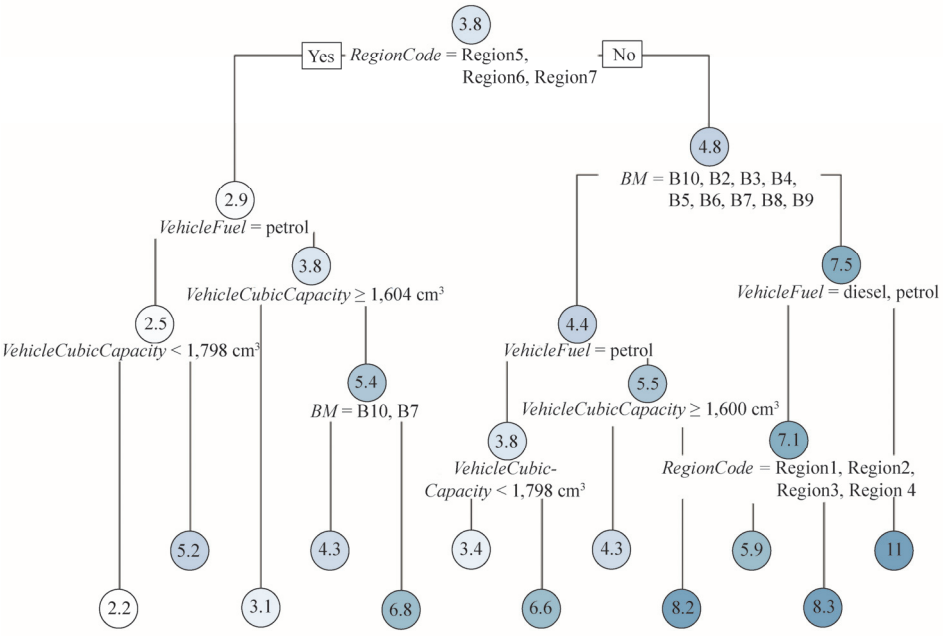


Note. The horizontal line shows the sum of the squared deviations for the probabilities of the voting model without clustering.

We will use the $k = 12$ clusters in the actual case and show further results in detail with this setting. Some results for $k = 21$ case are also shown. It is possible to increase further the number of clusters to develop the *SSW* measure and maintain a high rank correlation, but this makes the tree much more complex.

The resulting tree for $k = 12$ is shown in Figure 13. In each node, we can observe the average $\mathbf{P}(claim = 1)$ value in percent (number of nodes). The splitting condition is always given as a label to each node that is not a leaf. The tree is split at each node in such a way that policies that satisfy the current node's splitting conditions will be arranged to the left, and policies that do not satisfy the condition will be arranged to the right. With the average predicted claim probabilities inside the formed policy clusters (leaves), we can reproduce the variance of the predicted probabilities from the voting model with 64% accuracy.

Figure 13. Simplified decision tree with 12 clusters on the voting model results



Note. In each node, the average $\mathbf{P}(claim = 1)$ value can be seen in percent. Policies that satisfy the actual node's splitting conditions are arranged to the left (Yes label), and the others are arranged to the right (No label). A darkening of the blue colour indicates an increase in the average predicted claim probability values.

This is not an exceptionally good ratio for preserved variance with a regression tree, but the resulting model is simple compared to the quite complicated calculation

method for the claim probabilities from the voting model (the weighted average of claim probabilities from three, already quite complex ML models), so we could achieve our goal of simplifying. Furthermore, the decision tree can be used to identify homogenous policy groups with respect to claim probabilities.

We can observe that the decision tree built from the voting model reflects some characteristics of each original model. The most important split is based on the *RegionCode*, and this feature is quite important for RF and NN models. In Figure 13, we can see the *VehicleFuel* feature on the second and third levels of the tree, which was preserved in each model by every feature selection algorithm. At deeper levels, the tree is split in order to reflect the interaction of vehicle cubic capacity (*VehicleCubicCapacity*) and fuel type (*VehicleFuel*), which was the most important effect in the GLM.

In the last step, we tested the simplified models obtained with some of the previously used evaluation measures. Evidently, due to variance loss, these models probably have worse predictive performance than the original voting model on a contract level. However, it is important to quantify the difference between voting and other models. Table 9 shows the respective *AUC* values and Table 10 displays the earlier defined indicators regarding the $U(\alpha)$ utility function for each model.

Table 9

AUC measures extended with CART simplified voting models

Model	*AUC* (rank)
GLM	0.6446 (4.)
RF	0.6657 (2.)
NN	0.6347 (6.)
Voting	0.6791 (1.)
Voting (simplified with CART, 12 clusters)	0.6361 (5.)
Voting (simplified with CART, 21 clusters)	0.6523 (3.)

Table 10

Analysis of the utility function with L = –20, extended with CART simplified voting models

Denomination	GLM	RF	NN	Voting	Voting (simplified with CART, 12 clusters)	Voting (simplified with CART, 21 clusters)
Maximum utility	16,927	18,495	16,257	19,043	17,280	17,877
Kept portfolio (%)	81.6	88.0	82.0	85.0	82.1	80.0
Optimal cut-off value (%)	5.0	8.6	4.7	5.9	4.4	4.2
Rank	5.	2.	6.	1.	4.	3.

The results show that the models obtained from the voting model's CART classification have worse efficiency than the voting and RF techniques, but they perform similarly or better than the GLM and NN models. In particular, the 21-cluster case has quite good figures. These results are particularly beneficial for our simplified voting model, as it has only 12 or 21 different groups with simple classification rules, compared to the elementary models with quite difficult structures. Although our GLM model includes several feature variables and interactions, other models are much more complex and difficult or impossible to interpret.

Overall, the policies most at risk can be described by the interactions of several features. The ML models that culminate in the voting model can identify these complicated interactions between features. Furthermore, by building an explainer decision tree on the predicted $\mathbf{P}(claim = 1)$ vector, we can identify clusters of policies with homogenous risk profiles with an acceptable loss of variance and predictive performance; moreover, we can easily get interpretable segmentation rules based on known features of the policies.

Regarding the question about the possible overperformance of the GLM approach, it can be stated that the best model was not the GLM or GAM or any of the ML methods, but a mixture of these. From the actual experiment, we can conclude that the best solution is to build several separate models using ML techniques and stack them to integrate their advantages.

6. Conclusion

This study presented the application of the GLM, GAM, RF, and NN techniques for modelling the yearly MTPL claim event probabilities. Several evaluation measures were described or defined: the *AUC*, the loss function, the rank correlation, and the *SSW* between the actual claim experience and predicted probabilities. All the separate models showed good figures; however, the best model was a mixture of them, the so-called voting method, which is a weighted average of the predictions from the elementary models. In the last step, the pricing application of the model is described. In addition to the application of the full voting model, a simplified clustering method was used based on the CART algorithm, which can be a tool for transparent tariff making. This approach can help to visualise and publish our results in exchange for losing some of the complete model's variance, but with still tolerable performance, and it can also detect some of the interactions made by the model.

References

ABZALOV, M. [2016]: Exploratory data analysis. *Modern Approaches in Solid Earth Sciences*. Vol. 12. July. pp. 207–219. https://doi.org/10.1007/978-3-319-39264-6_15

ALVES, A. [2017]: Stacking machine learning classifiers to identify Higgs bosons at the LHC. *Journal of Instrumentation.* Vol. 12. No. 05. Article No. T05005. http://dx.doi.org/10.1088/1748-0221/12/05/T05005

BERGMEIR, C. – BENÍTEZ, J. M. [2012]: Neural networks in R using the Stuttgart neural network simulator: RSNNS. *Journal of Statistical Software*. Vol. 46. No. 7. pp. 1–26. https://doi.org/10.18637/jss.v046.i07

BOODHUN, N. – JAYABALAN, M. [2018]: Risk prediction in life insurance industry using supervised learning algorithms. *Complex & Intelligent Systems*. Vol. 4. No. 2. pp. 145–154. https://doi.org/10.1007/s40747-018-0072-1

BRADLEY, A. P. [1997]: The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern Recognition.* Vol. 30. No. 7. pp. 1145–1159. https://doi.org/10.1016/S0031-3203(96)00142-2

BREIMAN, L. [2001]: Random forests. *Machine Learning*. Vol. 45. Issue 1. pp. 5–32. https://doi.org/10.1023/A:1010933404324

BREIMAN, L. – FRIEDMAN, J. H. – OLSHEN, R. A. – STONE, C. J. [1984]: *Classification and Regression Trees*. Routledge. Boca Raton. https://doi.org/10.1201/9781315139470

DE JONG, P. – HELLER, G. Z. [2008]: Generalized linear models for insurance data. *International Statistical Review*. Vol. 76. Issue 2. pp. 300–328. https://doi.org/10.1111/j.1751-5823.2008.00054_17.x

DENUIT, M. – SZNAJDER, D. – TRUFIN, J. [2019]: Model selection based on Lorenz and concentration curves, Gini indices and convex order. *Insurance: Mathematics and Economics*. Vol. 89. pp. 128–139. https://doi.org/10.1016/j.insmatheco.2019.09.001

FAN, J. – LV, J. [2018]: Sure independence screening. *Wiley StatsRef: Statistics Reference Online*. pp. 1–8. https://doi.org/10.1002/9781118445112.stat08043

FIGINI, S. – UBERTI, P. [2010]: Model assessment for predictive classification models. *Communications in Statistics – Theory and Methods*. Vol. 39. No. 18. pp. 3238–3244. https://doi.org/10.1080/03610920903243751

GIANCATERINO, C. G. [2016]: GLM, GNM and GAM approaches on MTPL pricing. *SSPub*. pp. 427–481. http://www.ss-pub.org/wp-content/uploads/2016/08/JMSS16040502.pdf

GRAY, R. – KOVÁCS E. [2001]: Az általánosított lineáris modell és biztosítási alkalmazásai *Statisztikai Szemle*. 79. évf. 8. sz. 689–702. old.

HAJEK, M. [2005]: *Neural Networks*. University of KwaZulu-Nata. Durban.

HARRIS, T. – HILBE, J. M. – HARDIN, J. W. [2014]: Modeling count data with generalized distributions. *Stata Journal*. Vol. 14. No. 3. pp. 562–579. https://doi.org/10.1177/1536867x1401400306

HASTIE, T. – TIBSHIRANI, R. [1987]: Generalized additive models: Some applications. *Journal of the American Statistical Association*. Vol. 82. No. 398. pp. 371–386. https://doi.org/10.1080/01621459.1987.10478440

HASTIE, T. – TIBSHIRANI, R. – FRIEDMAN, J. [2009]: *The Elements of Statistical Learning*. Springer. New York. https://doi.org/10.1007/978-0-387-84858-7

HENCKAERTS, R. – ANTONIO, K. – CLIJSTERS, M. – VERBELEN, R. [2018]: A data driven binning strategy for the construction of insurance tariff classes. *Scandinavian Actuarial Journal*. Vol. 2018. No. 8. pp. 681–705. https://doi.org/10.1080/03461238.2018.1429300

HENCKAERTS, R. – CÔTÉ, M.-P. – ANTONIO, K. – VERBELEN, R. [2019]: Boosting insights in insurance tariff plans with tree-based machine learning methods. *North American Actuarial Journal*. Vol. 25. Issue 2. pp. 255–285. https://doi.org/10.1080/10920277.2020.1745656

KAFKOVÁ, S. – KRIVÁNKOVÁ, L. [2014]: Generalized linear models in vehicle insurance. *Acta Universitatis Agriculturae et Silviculturae Mendelianae Brunensis*. Vol. 62. No. 2. pp. 383–388. https://doi.org/10.11118/actaun201462020383

KOZA, J. R. – BENNETT, F. H. – ANDRE, D. – KEANE, M. A. [1996]: Automated design of both the topology and sizing of analog electrical circuits using genetic programming. In: *Gero, J. S. – Sudweeks, D*. (eds.): *Artificial Intelligence in Design '96*. Springer. Dodrecht. pp. 151–170. https://doi.org/10.1007/978-94-009-0279-4_9

KUHN, M. [2011]: The caret package. *A Unified Interface for Predictive Models*. pp. 1–27. https://www.r-project.org/conferences/useR-2010/slides/Kuhn.pdf

LECUN, Y. – BENGIO, Y. – HINTON, G. [2015]: Deep learning. *Nature*. Vol. 521. No. 7553. pp. 436–444. https://doi.org/10.1038/nature14539

LIAW, A. – WIENER, M. [2002]: Classification and regression by random forest. *R News*. Vol. 2. No. 3. pp. 18–22. https://cran.r-project.org/doc/Rnews/

MATVEJEVS, A. – MALYARENKO, A. – MATVEJEVS, A. [2014]: Estimation and calculation procedures of the technical provisions for outstanding insurance claims. *Applied Computer Systems*. Vol. 15. No. 1. pp. 14–21. https://doi.org/10.2478/acss-2014-0002

MOHRI, M. – ROSTAMIZADEH, A. – TALWALKAR, A. [2012]: *Foundation of Machine Learning. Second Edition*. MIT Press. Cambridge. https://doi.org/10.1007/978-3-642-34106-9_15

NEAL, R. M. [2007]: Pattern recognition and machine learning. *Technometrics*. Vol. 49. No. 3. pp. 366–366. https://doi.org/10.1198/tech.2007.s518

NOLL, A. – SALZMANN, R. – WUTHRICH, M. V. [2018]: Case study: French motor third-party liability claims. *SSRN Electronic Journal*. 4 March. https://doi.org/10.2139/ssrn.3164764

OHLSSON, E. – JOHANSSON, B. [2010]: *Non-Life Insurance Pricing with Generalized Linear Models*. Springer. Berlin, Heidelberg.

RUSSELL, S. – NORVIG, P. [2010]: *Artificial Intelligence – A Modern Approach. Third Edition*. Cambridge University Press. Cambridge. https://doi.org/10.1017/S0269888900007724

SALDANA, D. F. – FENG, Y. [2018]: SIS: An R package for sure independence screening in ultrahigh-dimensional statistical models. *Journal of Statistical Software*. Vol. 83. No. 2. pp. 1–25. https://doi.org/10.18637/jss.v083.i02

SMYTH, P. – WOLPERT, D. [1999]: Linearly combining density estimators via stacking. *Machine Learning*. Vol. 36. No. 1. pp. 59–83. https://doi.org/10.1023/A:1007511322260

SVETNIK, V. – LIAW, A. – TONG, C. – WANG, T. [2004]: Application of Breiman's random forest to modeling structure-activity relationships of pharmaceutical molecules. In: *Roli, F. – Kittler, J. – Windeatt, T.* (eds.): *Multiple Classifier Systems. 5th International Workshop, MCS 2004,*

Cagliari, Italy, June 9–11, 2004. Proceedings. Springer. Berlin, Heidelberg. pp. 334–343. https://doi.org/10.1007/978-3-540-25966-4_33

THERNEAU, T. – ATKINSON, B. – RIPLEY, B. – RIPLEY, M. B. [2015]: *rpart: Recursive Partitioning and Regression Trees. R Package version 4.1-10.* http://cran.ma.ic.ac.uk/web/packages/rpart/rpart.pdf%5Cnhttps://cran.r-project.org/web/packages/rpart/rpart.pdf

TIBSHIRANI, R. [1996]: Regression shrinkage and selection via the Lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*. Vol. 58. Issue 1. pp. 267–288. https://doi.org/10.1111/j.2517-6161.1996.tb02080.x

VENABLES, W. N. – RIPLEY, B. D. [2002]: Tree-based methods. In: *Venables, W. N. – Ripley, B. D.* (eds.): *Modern Applied Statistics with S*. Springer. New York. pp. 251–269. https://doi.org/10.1007/978-0-387-21706-2_9

VERBELEN, R. – ANTONIO, K. – CLAESKENS, G. [2018]: Unravelling the predictive power of telematics data in car insurance pricing. *Journal of the Royal Statistical Society. Series C: Applied Statistics*. Vol. 67. No. 5. pp. 1275–1304. https://doi.org/10.1111/rssc.12283

WOOD, S. N. [2003]: Thin plate regression splines. *Journal of the Royal Statistical Society. Series B: Statistical Methodology*. Vol. 65. No. 1. pp. 95–114. https://doi.org/10.1111/1467-9868.00374

WOOD, S. N. [2017]: *Generalized Additive Models: An Introduction with R. Second Edition*. Chapman and Hall/CRC. Boca Raton. https://doi.org/10.1201/9781315370279

YEO, A. C. [2011]: Neural networks for automobile insurance pricing. In: *Khosrow-Pour, M. D. B. A.* (ed.): *Encyclopedia of Information Science and Technology, Second Edition*. pp. 2794–2799. https://doi.org/10.4018/978-1-60566-026-4.ch446