Estimation of Individual Claim Liabilities

A comparison of Traditional and Machine Learning Methodologies

Marco De Virgilis Pierluigi Cerqueti

Abstract

The intent of this paper is to show how to implement machine learning (ML) algorithms in the framework of calculating Claim Liabilities.
Traditionally, in order to estimate future losses, actuaries have been using methodologies based on aggregated data in the form of run-off triangles.
This paper outlines the limitations of such methodologies and proposes more sophisticated tools and models based on ML algorithms that are capable of overcoming drawbacks of standard approaches, namely, accuracy and timeliness of estimates.
We propose a new framework that could enhance traditional estimates in providing an additional set of evaluations that could be used by actuaries as another term of comparison.

Keywords. Run-off Triangles, Chain-ladder, GLM, GAM, MARS, KNN, CART, Gradient Boosting, Neural Network, Classification, Regression, Claim Liabilities, Ultimate Losses, IBNR, RBNS.

Introduction

Estimating future claim payments is a central task performed by actuaries on a daily basis. Such estimates are of high value for the insurance companies because they constitute one of the main entries of their balance sheet.
The accuracy of these figures is, therefore, of primary concern for all of the stakeholders. Producing reliable numbers could really make the difference between a company operating in a safe and sound way or being put in receivership, rehabilitation or liquidation status.
Unfortunately, real world data is subject to inherent fluctuations and systematic distortions that makes this process very complicated and, often, expert judgment is needed. An additional layer of uncertainty is also introduced by the delay between the actual occurrence of the claims, the notification to the insurance company and the actual payment.
Ideally it is in the insurance company's stakeholders' interests to be able to know, as soon as possible, the final claim losses in order to quantify the liabilities and hence, take strategic decisions in a timely manner.
Traditional methods of estimating claim liabilities, such as the development technique, all require stable pattern and company practices in order to produce

reliable estimates; the major drawback is that a precise evaluation can only be calculated when sufficient time has been allowed for the fluctuations to stabilize.

Company management, however, needs to take decisions in a timely fashion. Allowing time before acting could mean losing competitive advantages within the market. This situation is also exacerbated by the fact that initial estimates have to be reviewed and checked periodically in order to ensure they are still valid.

It is also very difficult to compensate aggressive decisions, since insurance companies are not allowed to charge future customers more to make up for past losses. In fact, according to the CAS *Statement of Principles Regarding Property and Casualty Insurance Ratemaking*, "A rate is an estimate of the expected value of future cost."

It is clear that this definition does not grant the rights to overcharge future customers due to past choices that led to losses.

Traditional methodologies are based on run-off triangles - a nice and easy way to aggregate claims that allows to gauge both the time and the materiality development of the claims.

This, however, comes at a cost both in terms of accuracy and timeliness of reliable results. In order to overcome these disadvantages, this paper will show how to address this problem in the context of ML techniques, comparing the benefits and the difficulties of switching to such a framework.

This paper is set out as follows:

Section 1 will present the terminology and notations used throughout the paper.

In Section 2, this paper will address in more detail how traditional methodologies work, providing context and efforts made to enhance results obtained through this methodology.

Section 3 will provide a gentle introduction to the subject of ML algorithms, describing the fundamental concepts behind the models used later on.

Section 4 will present a case study based on real data discussing the results achieved, the issues encountered, and how to increase the quality of the estimates.

In Section 5, there will be a comparison of results obtained alongside a description of the performance indicators used.

In Section 6, we will present some considerations and proposed methodologies regarding the estimate of IBNYR.

In Section 7, we will state some conclusions.

All statistical analyses were performed using the free and open source statistical software R.[1]

[1] R Core Team (2018). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.

Confidentiality

Due to data confidentiality constraint, sensitive quantitative information has been intentionally hidden in the following study.
It is, however, possible to appreciate the value of the conclusions stated.
The focus of the paper is practical, and more emphasis will be given to the implementation and practical aspects. The main attention, therefore, will be on the algorithms themselves, rather than on the specific data.

1 Terminology and Notation

This paper will make extensive use of special terminology and notation.
We will start with a description of the claim process and the relative technical terms that describe each individual stage, then these concepts will be expressed in mathematical form.
As already described, the full claim payment process takes time from the moment of the accident until the claim is paid and therefore closed.
Moreover, claims are not always paid in full at the moment they are reported, but they can generate a series of small payments.
Once a claim is reported from the policyholder to the insurance company, the claim is known, and a case outstanding is initially estimated.
The term "case outstanding" refers to the amount that the insurance company, at any given time, expects to pay in addition to what has already been paid for each specific claim. The sum of these two components is called "reported claims."[2]
In addition to known claims, the insurance company will have to pay claims which have occurred but are not yet known. In fact, it can take some time between the time an accident actually happens and the moment the policyholder reports it.
The amounts that the company expects to pay on these claims is called "estimated pure IBNR" (incurred but not reported).[3]
The last component that the company needs to consider is the development on known claims; the company, in fact, could end up paying more (or less) than what it had previously anticipated.
This component is called IBNER, incurred but not enough reported. The sum of pure IBNR and IBNER make up the broad definition of IBNR.
The IBNR, i.e. the sum of future outflows which are not yet known, is therefore a random variable that needs to be estimated.
On the overall level, the sum of money that the company has to set aside to meet future liabilities is called reserve and, as we have seen, it is made of several components:[4]

- Case Outstanding on known claims
- IBNER
- IBNYR

The sum of these three components plus the payments already made is called ultimate claims amount, and it refers to the total losses that the company experiences.

[2] Sometimes it possible to find the term "incurred claims." The two definitions indicate the same concept.

[3] Other accepted definitions are INBYR, incurred but not yet reported, or more simplistically, IBNR.

[4] We are deliberating ignoring the expense component as we want to focus the discussion purely on the claim perspective.

The following mathematical notation will be used to identify the concepts explained:

- $r(w, d)$: Reported claim amount at time $w+d$ with respect to claims occurred at time w.
- $p(w, d)$: Amount paid at time $w+d$ with respect to claims occurred at time w.
- $os(w, d)$: Case outstanding at time $w+d$ with respect to claims occurred at time w.
- $R(w, d)$: Cumulative reported claim amount at time $w+d$ with respect to claims occurred at time w.

$$R(w, \hat{d}) = \sum_{d=0}^{d=\hat{d}} r(w, d)$$

- $P(w, d)$: Cumulative paid claim amount at time $w+d$ with respect to claims occurred at time w.

$$P(w, \hat{d}) = \sum_{d=0}^{d=\hat{d}} p(w, d)$$

- It is possible to derive the following relation: $R(w, d) = P(w, d) + os(w, d)$

2 Traditional Methodologies

The development technique, also known as the Chain Ladder technique, is one of the most frequently used methodologies for estimating unpaid claims.
The main assumption behind this methodology is that future claims will develop in a consistent way to previous claims already recorded.
Before explaining how such a technique works, it is fundamental to introduce the concept of a *run-off triangle*.[5]
A development triangle is a table that shows the value of claims (paid, reported or outstanding) of various cohorts over time.
It looks like the following:

				d			
		0	1	2	3	4	5
w	0	$R(0,0)$	$R(0,1)$	$R(0,2)$	$R(0,3)$	$R(0,4)$	$R(0,5)$
	1	$R(1,0)$	$R(1,1)$	$R(1,2)$	$R(1,3)$	$R(1,4)$	
	2	$R(2,0)$	$R(2,1)$	$R(2,2)$	$R(2,3)$		
	3	$R(3,0)$	$R(3,1)$	$R(3,2)$			
	4	$R(4,0)$	$R(4,1)$				
	5	$R(5,0)$					

Table 1: Run-off Triangle

The objective of estimating future claim liabilities is, therefore, estimating the lower part of the triangle shown in Table 1.
As already stated, the primary assumption of this technique is that future claims will follow past developing patterns. In order to represent this concept from a mathematical point of view the idea of development (age-to-age) factor is introduced.
For every development year $d \in (1, \dots, n)$, each development factor is defined as:[6]

$$f_d = \frac{\sum_{w=0}^{n-d} R(w,d)}{\sum_{w=0}^{n-d} R(w,d-1)}$$

It is therefore possible to define the ultimate cost of claims for each Accident Year w as:

$$R(w,d) = R(w,d-1) \prod_{k=n-w+1}^{d} f_d$$

[5] Also referred as *Development triangle*.

[6] Here the development factors are defined based on reported claims, however, they could also be defined based on paid claims.

It is possible to demonstrate that these definitions are subject to the following assumptions:[7]

- $E[R(w, d+1)|R(w, 1), \dots, R(w, d)] = R(w, d)f_d$
- $Var[R(w, d{+}1)|R(w, 1), \dots, R(w, d)] = R(w, d)\alpha_d^2$ with a proportionality constant α_d^2
- $\{R(i, 1), \dots, R(i, d)\}$ and $\{R(j, 1), \dots, R(j, d)\}$ are independent

2.1 Development Technique Example

In this section, we present an example of such methodology.
We consider the following run off triangle for cumulative claim amounts:[8]

		Development									
		0	1	2	3	4	5	6	7	8	9
Origin	1981	5,012	8,269	10,907	11,805	13,539	16,181	18,009	18,608	18,662	18,834
	1982	106	4,285	5,396	10,666	13,782	15,599	15,496	16,169	16,704	
	1983	3,410	8,992	13,873	16,141	18,735	22,214	22,863	23,466		
	1984	5,655	11,555	15,766	21,266	23,425	26,083	27,067			
	1985	1,092	9,565	15,836	22,169	25,955	26,180				
	1986	1,513	6,445	11,702	12,935	15,852					
	1987	557	4,020	10,946	12,314						
	1988	1,351	6,947	13,112							
	1989	3,133	5,395								
	1990	2,063									
	f_d	2.999	1.624	1.271	1.172	1.113	1.042	1.033	1.017	1.009	1.000

Table 2: RAA Triangle

[7] We will not prove these equations here. For further details, please see [11].

[8] Historical Loss Development, Reinsurance Association of America (RAA), 1991, p.96.

Using the previous equations we can obtain the following results:

AY	Reported	Ultimate	IBNR
1981	18,834	18,834	0
1982	16,704	16,858	154
1983	23,466	24,083	617
1984	27,067	28,703	1,636
1985	26,180	28,927	2,747
1986	15,852	19,501	3,649
1987	12,314	17,749	5,435
1988	13,112	24,019	10,907
1989	5,395	16,045	10,650
1990	2,063	18,402	16,339
Total	160,987	213,122	52,135

Table 3: Development Technique Results

2.1.1 Advantages and Disadvantages

This technique has been proven to be a powerful methodology for actuaries and it has been widely used and implemented. In fact, several reserving algorithms are based on this approach.

It has the advantage of being of easy implementation and it does not involve difficult calculations. Moreover, with the aid of some assumptions, it can be used to implement stochastic approaches.[9]

Such procedures allow the estimate of uncertainty around the point estimate, producing distributions of the IBNR (or reserve).

The main critique of this method is the compression of data and the subsequent loss of information. In fact, compacting all the data from different years, and possibly millions of different claims, into a relatively small triangle, a considerable amount of information is inevitably lost.

It would be more effective to make extensive use of all the information and data that is available, in order to produce a more accurate projection of ultimate claims.

Since all of the data is usually already recorded by the claim department, it would be clever to actually use it, thus improving actuarial projections.

Another limitation of this methodology is regarding the timeframe in which the estimates are calculated.

As it is possible to note from Table 3, the IBNR is greater for more recent years as they are less developed.

This carries a higher level of uncertainty in the projections, and estimates need to be revised as these accident years develop.

[9] Two of the most famous and used non-proprietary models are those of Mack [11] and England and Verrall [6].

Ultimate projections, when limited data is available, are greatly influenced by the level of the first data points, and random fluctuations could severely distort the ultimate claim estimates.
This aspect could carry greater implications as company executives need to take decisions in a timely fashion. Allowing time before acting could mean losing competitive advantages within the market.

3 Machine Learning Techniques

Recent advances in computer power have paved the way for more sophisticated algorithms that make more extensive use of data in a wide variety of areas.

Such algorithms can improve data and computational capabilities leading to situations in which traditional actuarial tasks can be tackled with increasingly sophisticated approaches.

In the remainder of this paper, we will focus on several algorithms and exploit their capabilities in the area of individual claim reserving.

3.1 General Framework

The target of the exercise is to predict the ultimate cost of the claims when they are initially reported. At this stage, when $d = 0$, there is no paid amount and an initial case reserve is established.

These claims, usually called RBNS, Reported But Not Settled, can follow two different paths.

In one case they will be paid and therefore, at $d = \infty$, they will be fully settled and there will be a paid amount $R(w, \infty) > 0$. This is the amount that needs to be estimated.

On the other hand, claims could be closed with no payment (CNP), and therefore, $R(w, \infty) = 0$.

In order to model the previous relations we will build two different frameworks: a classification and a regression structure.

First the probability of each claim to be closed with no payment will be computed, and then, if this first process will have a negative outcome (i.e. the claim will be paid), an amount will be calculated.

In addition, a third model will estimate the time that this process will take, from the moment the claim is reported until it is closed, either with payment or not.

3.2 Modeling Framework

In this section we present the various modeling techniques that have been used to fit the data.

Each modeling framework is introduced and explained. Moreover, for further details, all the necessary references are provided.

3.2.1 Generalized Additive Model (GAM)

A Generalized Additive Model is an additive linear model in which the target variable depends linearly on some function of the predictor variables.

The basic idea is to replace $\sum x_{ij}\beta_j$, the linear component of the model, with an additive component $\sum f_j(x_{ij})$.

The mathematical specification of the model is as follows:

$$y_i = Exponential(\mu_i, \theta)$$
$$g(\mu_i) = \beta_0 + f_1(x_{i1}) + f_2(x_{21}) + \cdots + f_j(x_{ij})$$

GAMs, like GLMs, assume the random component of the outcome to follow an exponential distribution.

In addition to standard GLMs, such a framework allows the addends of the linear model to be any arbitrary functions of the predictors. These functions, $f_1(\cdot), f_2(\cdot), \ldots, f_j(\cdot)$, will be splines,[10] producing a smoothing effect on the predictors.

CNP Classification

As previously described, it is necessary to define a distribution that belongs to the exponential family for the response variable.

The logistic regression is a universally accepted modeling framework for classification tasks. From a mathematical perspective, the model will predict the probability that each claim will result in a payment. Therefore, the model will return a number between 0 and 1.

This could be interpreted, for each claim, as:

$$\text{Claim Status} = \begin{cases} \text{Claim CNP} & \text{if p} \leq 0.5 \\ \text{Paid Claim} & \text{if p} > 0.5 \end{cases}$$

Payment Amount

The response variable is assumed to be distributed according to a Gamma distribution.[11]

The corresponding probability density function is:

$$f(x; \alpha, \beta) = \frac{\beta^\alpha x^{\alpha-1} e^{-\beta x}}{\Gamma(\alpha)} \quad \text{for } x > 0 \text{ and } \alpha,\, \beta > 0$$

It is worth noting that the Gamma distribution is only defined for positive values of x. In this context, x represents the paid amount, and therefore, we expect these quantities to be always positive.[12]

Payment Lag

The Payment Lag represents the number of days between the moment the claim has been reported and the moment it has been closed either with or without payment.

This quantity can therefore be greater than or equal to zero (if the claim has been closed the same day as received), and it will always be an integer.

[10] Splines are defined piecewise by polinomials. For further details, please see [7].

[11] Other choices of this distribution could also be possible, e.g. log-normal or inverse gaussian.

[12] For simplicity purposes, we are excluding the case of salvage and subrogation.

A Poisson distribution could be used to fit the data.[13] The probability of closing a claim in n days is, therefore, given by the equation:

$$P(n) = e^{-\lambda}\frac{\lambda^n}{n!}$$

3.2.2 Multivariate Adaptive Regression Spline (MARS)

Another GLM variant that allows to handle non-linearity natively is called Multivariate Adaptive Regression Splines, or MARS.[14]
Instead of fitting smooth functions, as the GAM discussed in the previous section, MARS models incorporate piecewise linear functions, or *hinge functions*, into a regular GLM.
Moreover such models allow to model significant interactions between the predictors, as well as interactions among the piecewise linear functions.
The response variables chosen would be the same as the ones described in the previous sections, i.e. binomial, gamma and poisson.

3.2.3 K Nearest Neighbor

K Nearest Neighbor (KNN) is perhaps the most straightforward algorithm among all machine learning techniques. The mechanism is very simple, examples are classified based on the nearest neighbors.
In KNN classification, the output is a class. An object is classified according to its neighbors, with the object being assigned to the class most common among its k nearest neighbors.
In KNN regression, the output is a value. This value is the average of the values of its k nearest neighbors.
Let us assume that we have a training data set X made up of $x_i, 1 < i < N$ training examples.
The examples are described by a set of features F. Each training example is classified with a class $y_i, 1 < i < N$. The objective is to classify an unknown sample k.
For each $x_i \in N$ it is possible to calculate the distance between k and x_i as follows:

$$d(k, x_i) = \sum_{f \in F} w_f \delta(k_f, x_{if})$$

where w_f is the weight assigned to the feature f and δ is the chosen distance metric.
The k nearest neighbors are selected based on this distance metric.

3.2.4 Gradient Boosting

Boosting is one of the most powerful learning methodologies in the field of statistical learning. It was originally introduced in classification tasks, but it

[13] Another choice could also be the negative binomial distribution.

[14] For further details, please see [9].

can be nicely extended to regression problems as well.
The basic idea is to combine a set of *weak* learners to produce a powerful model. A *weak* learner is a model whose error is only slightly better than random guessing.
The purpose of boosting is to sequentially apply the *weak* learners to repeatedly modified versions of the data.
The final prediction is then produced, combining all of the *weak* learners.
In the remainder of this paper, such algorithms have been applied considering classification and regression trees (CART).

Regression and Classification Trees (CART)

The basic idea behind trees is that they produce disjoint regions of the space, R_i, as represented by the terminal nodes of the tree.

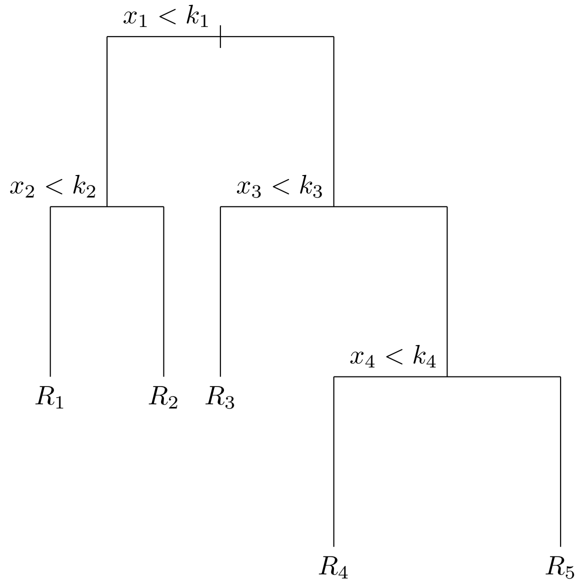


Figure 1: Classification Tree diagram

In classification tasks, the regions R_i would represent the class which the sample x_i belongs to. For our purposes, two classes have been identified, *Claim CNP* or *Paid Claim.*
In regression tasks, the regions R_i would represent the average value for each region of the target value. For our purposes, two models have been built: one modeling the claim amount and one modeling the closing lag.

Boosting Trees

The procedure of boosting consists in iteratively fitting trees to the data in order to produce a powerful learner.

At each stage, n $(1 < n < N)$, it can be assumed that a weak model (i.e. simple mean), T_n, has been estimated from the data.
The algorithm then improves the previous estimates by introducing a new model f that yields better results, $T_{n+1}(x) = T_n(x) + f(x)$.
The optimal model f, would, therefore, be the one that satisfies the following equation:

$$T_{n+1}(x) = T_n(x) + f(x) = y$$

which yields:

$$f(x) = y - T_n(x)$$

The gradient boosting algorithm, therefore, will fit a new model on the residuals from the previous model. This means that each tree, T_{n+1}, will correct the output of the previous tree T_n.
In order to mathematically implement the algorithm it is also necessary to define a loss function that allows the computation of the gradient in order to optimize the model parameters; this makes the gradient boosting model a gradient descent algorithm[15].

3.2.5 Artificial Neural Network

Artificial neural networks represent a very powerful class of machine learning models, inspired by the brain structure. Each net is made up of several interconnected neurons, organized in layers, that compute mathematical functions. Artificial neural networks have been proven to be very effective in bot classification and regression tasks.

[15] Further information available at [8].

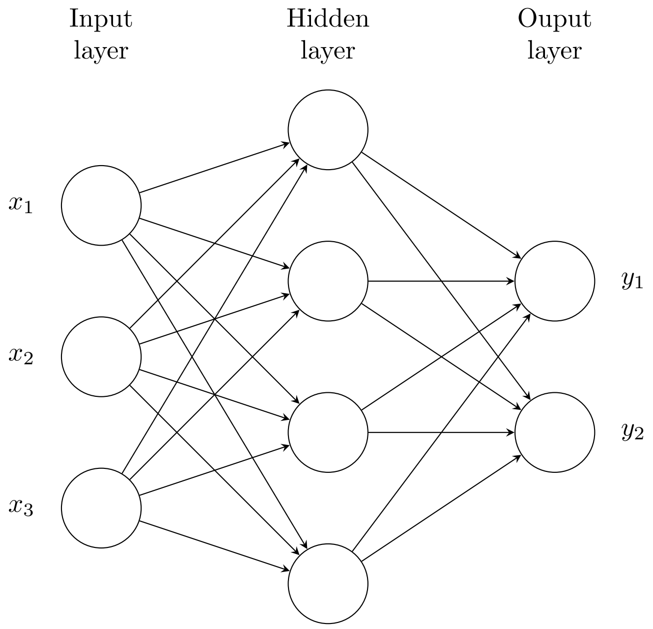


Figure 2: Neural Network structure

The figure above represents a neural network which takes three input features, it has one hidden layer with four units (neurons), and two output units. Neural networks with several hidden layers are defined as *deep* networks, as opposed to *shallow* networks, which don't have very many hidden layers.
The main idea behind this structure is that each neuron applies a linear transformation to the output of a previous unit and computes a function, defined *activation* function, before passing the result to the next layer.
If we assume that x is a feature of the data set, a single neuron, will compute the following transformation:

$$f(Wx + b)$$

Where f represents the *activation* function (e.g. sigmoid or ReLU), W and b are respectively the weights and the biases of the model that needs to be optimized. In order to find the best parameter, an optimization algorithm based on stochastic gradient descent could be implemented.[16]
For our modeling framework, three different neural networks with different tasks have been fitted.
One network is responsible for the Payment/No Payment classification, one network is responsible for the paid amount estimate, and one for the closing lag.

[16] Further information at [14].

4 Case Study

In order to compare the different algorithms we have selected a series of claims that have been fully closed, either with payment or not, and we will compare the actual observed values with the predicted values from the models.
The previously described modeling frameworks have been applied to the data: results and comparisons will be discussed and analyzed in the next session.

4.1 Database Description and Pre-processing

The database is made of about 800,000 physical liability damage closed claims from standard Personal Auto Policies with Accident Year ranging from 2008 to 2018.
The variables used as target variables are the following:

- *CNP Indicator*: This variable indicates if the claim has been paid or closed with no payment.
- *Payment Lag*: Time between the reporting date and the payment date.
- *Paid Amount*: Claim amount paid to the insured.

The variables used as predictors are listed below (in alphabetical order):

- *Accident Month*: Accident month of occurrence.
- *Accident Location*: Where the accident has occurred.
- *Age*: Age of the insured driver.
- *Attorney*: Whether or not the insured has an attorney.
- *Authorities Intervention*: Whether or not any authorities have intervened.
- *BAC*: Blood Alcohol Concentration of the insured.
- *City*: City of occurrence.
- *Fraud Indicator*: Whether or not the claim has been flagged as fraud.
- *Garage Location*: Where the car is usually parked overnight.
- *Gender*: Gender of the Insured.
- *Reported Amount*: Reported amount at $t = 0$.
- *Reporting Lag*: Time difference between the accident date and the reporting date.
- *Reporting Method*: How the accident has been reported, eg. phone, email.
- *Reporting Month*: Reporting month to the insurer.
- *Vehicle Manufacturer*: Insured's vehicle manufacturer.
- *Witnesses*: Presence of witnesses to the accident.

4.1.1 Accident Year Considerations and On Leveling of Claims

The Accident Year has been purposely removed from the predictor variables because the main intent of the modeling exercises is to build models that will work on any claims, possibly also on future claims, rather than only on ones which have already happened or on a specific accident year.

In doing this, the actuarial department can build a model that, once put into production, can estimate the ultimate cost of claims in an automated way as they are reported.

Since the claim database goes back to 2008, it was necessary to put all of the claims on level in order to take into account inflation, court awards, or change in legislation.

This allows to treat each claim as it happened in the same period and under the same condition or regulatory framework.

4.1.2 Train and Test Split

The database has been split into train and test set, according to an 80% and 20% proportion. All the models have been trained on the train set and evaluated on the test set.

This gives the opportunity to check and evaluate the performance of the models on data never seen before, essentially replicating a real word scenario.

More precisely, once the model has been built and evaluated it can been used on new claims as they are reported to the insurer.

5 Results Comparison

In this section we will present the results obtained alongside a description of the performance indicators used.
We start by introducing the main metrics implemented to assess the models.

5.1 Regression Performance

As far as regression is concerned, we evaluate the goodness of fit using the Normalized Root Mean Square Error (*NRMSE*).
This metric can be calculated as follows:

$$NRMSE = \frac{RMSE}{y_{max} - y_{min}}$$

where the *RMSE* is the so-called Root Mean Square Error:

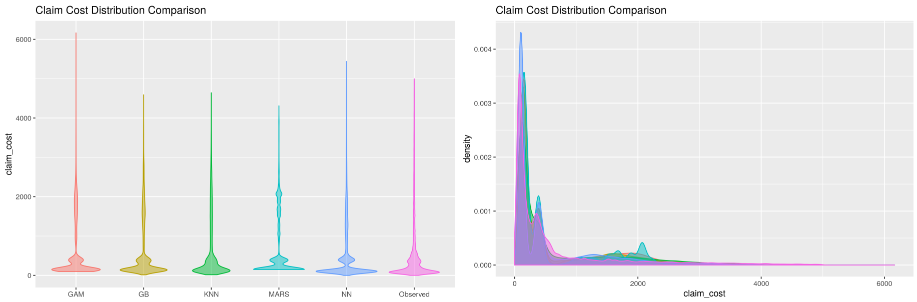
$$RMSE = \sqrt{\frac{\sum_{t=1}^{T}(\hat{y}_t - y_t)^2}{T}}$$

Here we report a summary table of the regression performances for each modeling approach implemented for both the target variables: claim cost and closing lag.

NRMSE	GAM	MARS	KNN	GB	NN
Claim Cost	0.1383	0.1380	0.2228	0.1337	**0.1291**
Closing Lag	0.0988	0.1018	0.1014	**0.0793**	0.0990

Table 4: NRMSE Results

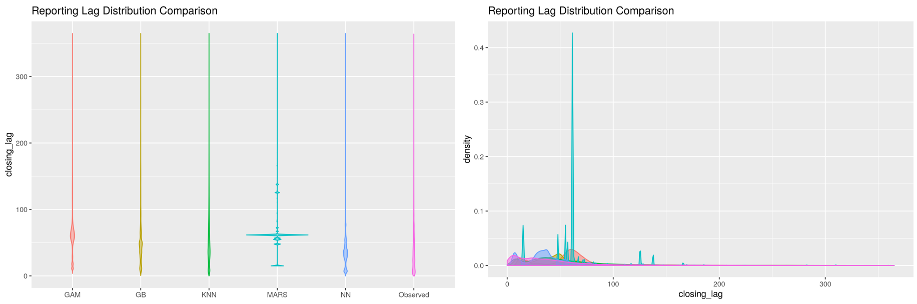
Here are the violin plots and the comparisons of the distributions for both the claim cost and the payment lag.



(a) Violin Plot

(b) Distribution Comparison

Figure 3: Claim Cost Estimates



(a) Violin Plot

(b) Distribution Comparison

Figure 4: Closing Lag Estimates

As it is possible to observe, the Neural Network approach and the Gradient Boosting approach yield the best results for, respectively, Claim Cost and Closing Lag.

As far as the estimation of the claim costs is concerned GAM, MARS and GB lead to very similar results, only slightly worse than the NN performance. In a situation in which computing power would be limited, going for such estimates could be perfectly reasonable.

Performances of the models estimating the closing lag are all very close to each other, apart from the Gradient Boosting approach, which is deemed to be the best model for the task.

5.2 Classification Performance

The performance of a classification task could be evaluated according to the area under the Receiver Operating Characteristic curve (ROC curve) and the Precision-Recall F measure.

A confusion matrix is the starting point to evaluate the performance of a binary classifier. This matrix has the following notation:

	Reference Value	
Predicted Value	True Positive Rate	False Negative Rate
	False Positive Rate	True Negative Rate

Table 5: Confusion Matrix

For each cutoff point, all of these metrics can be evaluated and plotted on a graph. The ROC is created by plotting the true positive rate (TPR) against the false positive rate (FPR).
The Area Under the Curve (AUC) is equal to the probability that the classifier will rank a randomly chosen positive instance higher than a randomly chosen negative one.
AUC ranges in value from 0 to 1. A model whose predictions are all wrong has an AUC of 0; one whose predictions are always correct has an AUC of 1.
It follows that the greater the area, the better the classifier.
The next graph plots this curve for all the classification algorithms implemented:

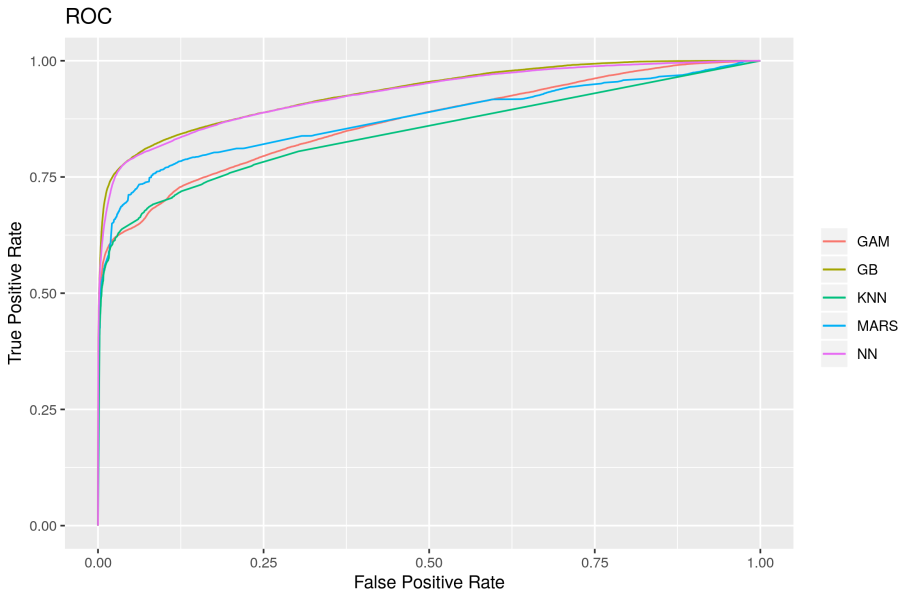


Figure 5: ROC

And the relative AUC Values:

AUC	GAM	MARS	KNN	GB	NN
Paid/CNP	0.8661	0.8748	0.8457	**0.9311**	0.9273

Table 6: AUC

Based on the previous metrics it is possible to calculate, for each value of the cutoff point, the so called F1 score.
If we define the *Precision* as:

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

The F1 score could be computed as:

$$F1 = 2 * \frac{Precision * Recall}{Precision + Recall}$$

Where the *Recall* is defined in the same way as the *True Positive Rate*.
The F1 score could be interpreted as a measure of accuracy. Here are the values (at the optimal cutoff points) for all the models implemented:

F1 Score	GAM	MARS	KNN	GB	NN
Paid/CNP	0.7140	0.7387	0.7060	**0.8074**	0.7963

Table 7: F1 Score

From the evidence previously presented, we can deduce that the best classification model is the Gradient Boosting approach. However, Artificial Neural Networks, also lead to good performances.
It is possible to observe how these two algorithms are notably better than all the other methodologies implemented, at least as far as classification is concerned.

5.3 Overall Performance

Finally, bringing everything together, we have evaluated the results considering all of the three modeling steps together.
The model works in a sequence fashion as described below:

1. The first algorithm estimates the probability of whether a claim will be paid or not, and it classifies accordingly.

2. Claims classified as being paid are fed into the second model which estimates the ultimate cost.

3. The third and last model will estimate, for each claim, the closing lag.

Following the framework previously described, it is possible to treat each claim individually as they are reported, and then aggregate amounts, in order to produce outflows.
This process has the advantage of producing, for each term, the expected aggregate claim amounts, allowing the calculation of discounted liabilities, if permitted by the regulatory framework.
Similarly to the previous sections, we present a summary comparison between observed and estimated outflows.
Here is a summary table with the respective NRMSE values:

NRMSE	GAM	MARS	KNN	GB	NN
Outflows	0.0478	0.0558	0.0557	**0.0194**	0.0213

Table 8: Comprehensive NRMSE Results

We continue the analysis presenting a graph that compares the outflows for each payment quarter:

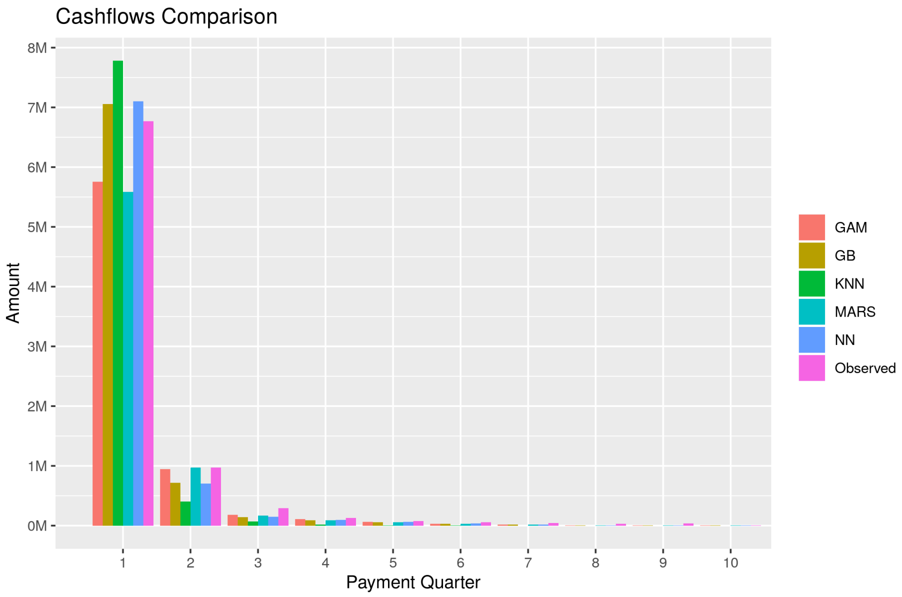
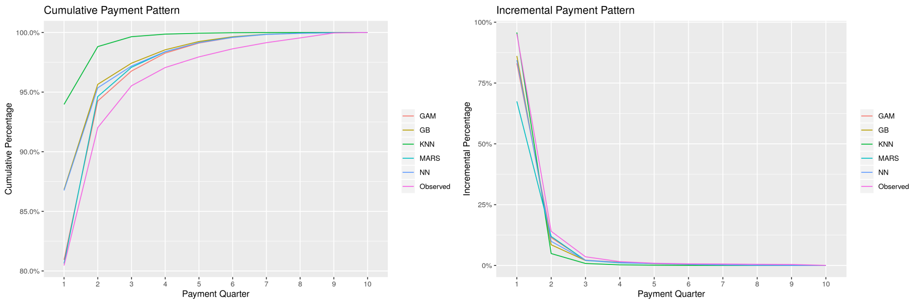


Figure 6: Outflows Comparison

Another visual tool that can help compare the model estimates is the plot that parallels the payment patterns (cumulative and incremental) produced by each approach:



(a) Cumulative Payment Pattern

(b) Incremental Payment Pattern

Figure 7: Payment Pattern Comparison

Observing the previous results, it is possible to appreciate that the model that produces the estimates that most closely match the actual experience is the Gradient Boosting. Neural Networks, however, perform in a very satisfactory way as well, and they can also be a viable option.
The choice of the model, nonetheless, depends also on several other factors such as availability of data and computing power.

6 IBNYR Considerations

The estimations performed up to this point have allowed us to compute the values of the ultimate cost for reported but not settled claims (RBNS).

The objective of this first analysis was, in fact, to predict the level of the IBNER. However, this is only one component of the total amount of future claim liabilities.

Total IBNR is, indeed, the sum of two factors: IBNER and IBNYR.

IBNYR is defined as the sum of claims that have occurred but are not yet known. It follows from this definition that it cannot be determined from predictive models applied to known claims, as in the previous exercise.

At this point it is also necessary to understand that the level of IBNYR depends on the evaluation date. This is because the more an Accident Year is developed, the lesser IBNYR is expected.

For example, if we are evaluating a given accident year at two different evaluation dates, the level of IBNYR at the later evaluation date will be less than the IBNYR level at the previous evaluation date; this depends on the fact that more time has been allowed and therefore more claims have been reported.

If we assume we are performing a reserve estimation at the end of year N, the IBNYR (with respect to year N only) will be equal to the sum of all the claims occurred during year N but that will be reported from year $N+1$ onward.

6.1 Case Study

In order to evaluate the level of IBNYR for our case study we followed a simple but effective methodology. Let's consider that the evaluation date is December 31, *N*.

1. We have considered the observed ultimate value of all the claims occurred in all the previous years and reported by year end.

2. We have considered the observed ultimate value of all the claims occurred in all the previous years and reported after year end.

3. Computing the ratios of these quantities, IBNYR/(RBNS + IBNER), we can have a time series that gives us the proportion of IBNYR at year end, compared to the ultimate value of claims reported by year end.

4. If we compute such ratios for all the years up to year *N-1* and estimate a value for year *N*, we could, then, multiply this estimate by the level of ultimate amounts predicted using the methodology already described to obtain an estimated value for the IBNYR.

The only drawback encountered in carrying out this algorithm is that we didn't have older accident years that were developed enough to have statistically stable data.

In order to overcome this issue, we have simulated a large amount of claims using the Neural Network already trained to enlarge the dataset.

Of course, in a context where more data is available, it would not be necessary to simulate claim amounts.
Here are the ratios obtained:

Accident Year	Ratios
2008	0.0654
2009	0.0642
2010	0.0698
2011	0.0711
2012	0.0575
2013	0.0562
2014	0.0557
2015	0.0614
2016	0.0519
2017	0.0668

Table 9: Ratio Series

At this point it is sufficient to estimate the value for the following year, and then estimate the level of the IBNYR.
This can be done with various techniques such as Local Polynomial Regression or Spline Interpolation.
We present the output of the implementation of such techniques:

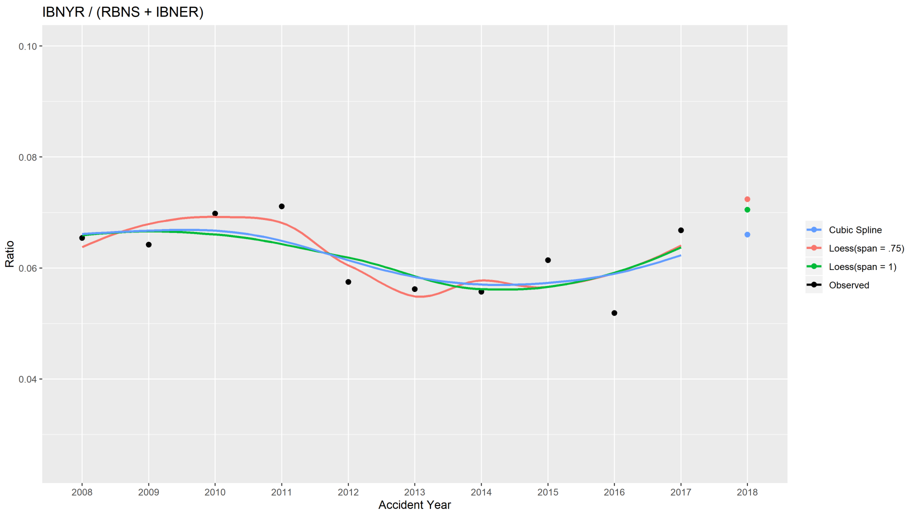


Figure 8: Ratios estimation

Having selected a sensible ratio for the following year, it is possible to mul-

tiply this figure by the amount of ultimate claim amounts estimated following the procedure described previously.
Once again, since the most recent year is not fully developed, we compared the estimates with the claim level obtained from the simulation exercise.
These are the results in terms of percentage deviation for each of the three possible estimation techniques presented:

2018 IBNYR	LOESS, span = .75	LOESS, span = 1	Cubic Spline
% Error	2.677%	2.646%	**2.475**%

Table 10: 2018 IBNYR Estimation Error

As it is possible to observe from the previous table, we obtained fairly accurate estimates.
It is, however, important to keep in mind that such comparisons have been produced against simulated data and not real observed paid claims.

7 Conclusions

In this paper we have presented how machine learning methodologies could be implemented in the context of estimating claim liabilities.
This workflow is based on the construction of subsequent models with different targets in order to capture the main features of insurance claims.
Moreover the full analysis is divided into two steps: first we have projected RBNS claims, and then we predicted the level of IBNYR to produce the full level of ultimate claims for any given year.
The results produced carry a high level of accuracy and they also have the advantage of an early evaluation, i.e., it is not necessary to wait for claim development as it is in standard techniques, e.g. Chain Ladder.
The framework presented also has the benefit of not completely relying on the accuracy of individual point estimates.
This is because, even if we allow small errors on the individual claim evaluations, we are mainly interested in the reserve amount on the aggregate level.
In this scenario, small discrepancies on individual claims can easily compensate each other and still produce an accurate and precise aggregate claim reserve.
A prompt claim reserve estimate, also, has the valuable advantage of allowing early decisions from the management, such as investment strategy, mix of business, market expansion, or mergers and acquisitions. There are, however, some inconveniences to consider before implementing a similar framework.
One of the main difficulties is the availability of data; complex machine learning methodologies, to be properly trained, require a considerable amount of data.
Another complication that may arise is related to the technological infrastructure. Computational power is one of the keys to success in the field of Machine Learning.
Whereas this research is focused on estimating the ultimate amount of claims, future studies could explore the possibilities of applying machine learning algorithms to predict individual claim development.

References

[1] Antonio, K., Plat, R., *Micro-level stochastic loss reserving for general insurance*. Scandinavian Actuarial Journal, 2014/7.

[2] Bett, L., *Machine Learning with R*. Packt Publishing Ltd, 2014.

[3] Bornhuetter R.L., Ferguson, R.E., *The actuary and IBNR*. Proceedings of the Casualty Actuarial Society 59, 1972.

[4] Breiman, L., Friedman, J.H., Olshen, R.A., Stone, C.J., *Classification and Regression Trees*. Wadsworth Statistics and Probability Series, 1984.

[5] Charpentier A., *Computational Actuarial Science with R*. CRC Press, 2015.

[6] England, P.D., Verrall, R.J., *Stochastic Claims Reserving in General Insurance*. British Actuarial Journal, 8, issue 3, 2002.

[7] Ferguson, James C., *Multi-variable curve interpolation*. J. ACM, vol. 11, no. 2, pp. 221-228, Apr. 1964.

[8] Friedman, J. H., *Greedy Function Approximation: A Gradient Boosting Machine*. The Annals of Statistics, Vol. 29, No. 5, Oct. 2001.

[9] Friedman, J. H., *Multivariate Adaptive Regression Splines*. The Annals of Statistics, Vol. 19, No. 1, Mar. 1991.

[10] Hastie, T., Tibshirani, R., Friedman, J., *The Elements of Statistical Learning. Data Mining, Inference, and Prediction*. Springer Series in Statistics, 2009.

[11] Mack, T., *Measuring the Variability of Chain Ladder Reserve Estimates*. Casualty Actuarial Society Forum, Spring 1994.

[12] Pigeon, M., Antonio, K., Denuit, M., *Individual loss reserving with the multivariate skew normal framework*. ASTIN Bulletin 43/3, 2013.

[13] R Core Team, R *A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing, 2018.

[14] Robbins H, Monro S., *A Stochastic Approximation Method*. The Annals of Mathematical Statistics, Vol. 22, No. 3. Sep. 1951.

[15] Wiley, J. F., *R Deep Learning Essentials*. Packt Publishing Ltd, 2016.

[16] Wüthrich, M.V., *Machine learning in individual claims reserving*. Scandinavian Actuarial Journal, 2018/6.