Individual claims reserving in Credit insurance using GLM and Machine Learning

Damiano Ticconi

Sapienza, University of Rome,
Department of Statistical Sciences
damiano.ticconi@uniroma1.it

Abstract. This paper is inspired by the work of Wüthrich, who introduced the use of machine learning techniques in non-life claims reserving.
Machine Learning techniques, born to channel data complexity and able to deal with highly non-linear dependencies, are presently expected to provide a valid alternative to traditional reserving techniques. Moreover, in a framework where insurance undertakings are collecting an increasing amount of data on policyholders and claims, it seems natural investigating the potentialities of these algorithms.
We focused on salary-backed loan insurance, a peculiar branch of credit insurance providing coverage for the outstanding debts in case of policyholder's unemployment. A particular feature of these contracts is that they are characterised by a claim frequency very sensitive to credit cycle trends, with strong variability in time.
The aim of this work is to investigate whether machine learning techniques can deal efficiently with such variability, exploiting information of macro-economic indices.
We offer a comparison of three different statistical techniques: Generalized Linear Models (GLM), which represent the benchmark of the analysis, Artificial Neural Networks (ANN), whose popularity is spreading in actuarial sciences, and Support Vector Machines (SVM), that are known for their good generalisation capabilities but are new in this field.

Key words: Individual claims reserving, granular reserving, machine learning, GLM, Artificial Neural Networks, Support Vector Machines, credit insurance

Introduction

Claims reserving is one of the main issues for non-life actuaries, founding its roots in loss development triangles such as Chain Ladder. Despite their effectiveness depends on the hypothesis of claim homogeneity, which is rarely met in reality, their use is mainly motivated by computational efficiency. The broadening of computational limits in recent times and the increasing amount of information collected by insurance companies have however encouraged researchers in exploring the use of new techniques that could exploit these achievements and improve the reserving process by including individual information on claims.
In this direction, many authors like Zhou [21] and Taylor [16] have proposed the use of GLMs, which are now considered standard actuarial tools, while Wüthrich [19], [20] showed the benefits of machine learning techniques, opening a new line of research. Support Vector Machines, introduced by Vapnik [3], represent our main proposal in this context, as they represent a fairly new technique in actuarial sciences which has mainly been proposed for fraud detection, see e.g. Kirlidog [8].
Following this new path, we hence explored the use of machine learning in a peculiar branch of credit insurance, namely insurance covers linked to *Cessione del quinto* (*CQ* for short), which is a form of salary-backed loan unique to Italy. Being known the cyclic nature of credit insurance, we used these techniques to consider during the reserve estimation process, in addition to individual claim information, macro-economical variables such as GDP.
This work is organised in 7 main sections: in the first part we explain the peculiarities of salary-backed loan insurance, providing then a quick review of claims evolution process. In Section 3 we recall the claim reservation principles and we show how we declined the estimation process. In Section 4 we show the essentials of the methodologies applied in the

case study and then we dedicate two sections to the results of the case study. The work closes with a section concerning computational issues, followed by our conclusions.

1 Salary-backed loan insurance

A salary-backed loan is a peculiar type of employee individual loan, unique to Italy and fitting among consumer's credit operations.
Their distinctive characteristic consists in the payment of the mortgage through a direct transfer of a percentage, up to a fifth, of the worker wage/salary.
This determines a strong decrease in terms of insolvency rate with respect to the market average but, nonetheless, the Italian Regulator requires an additional assurance on these operations which is to be given by means of an insurance contract.
This contract is dual in nature, covering the outstanding debt in both the cases of *death* or *unemployment* of the Transferee. Due to this duality, it has to be distinguished in two different risks, referring respectively to Term Life insurance and Non-Life insurance, the latter being the main interest of this work.

We hence identify four different actors on the operation:

1. A *Transferor*, granting the loan, underwriter and beneficiary of the insurance coverage;
2. A *Transferee*, requesting the loan and insured one;
3. The *Insurance Company*, providing coverage on the loan;
4. The Transferee's *Employer*, withholding a part of the Transferee's salary, used to cover each mortgage rate.

2 Claim timeline

We now provide a quick summary of the evolution timeline for a CQ unemployment claim, as shown in Figure 1, to allow a better understanding of the claim reservation process.

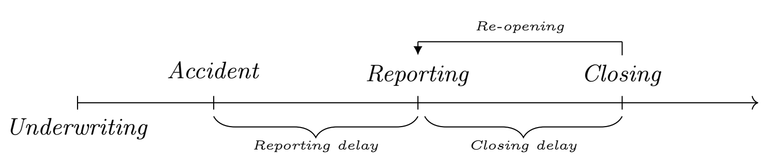


Fig. 1: Example of claim timeline.

- A claim takes place when the Transferee becomes unemployed, either for resignation or termination, causing the inability of the Employer to pay mortgage rates by withdrawing from the worker's salary. This moment is usually called *accident date* and will be referred to as a in the following.
  In this phase only the Transferor has knowledge on the accident and has to execute standard recovery procedures, such as monitoring whether the insured one can find a new job.
- If recovery is impossible, the loan is defaulted and the Transferor involves the Insurance Company, marking the *reporting date* and a consequent *reporting delay*, which we denote as r, i.e. the time gap between accident and reporting dates.
  It has to be noticed that it is not rare to observe relevant reporting delays, even larger than a year, because of the nature of the monitoring procedure and a prescription period equal to two years. This phenomenon is quite significant for a sound management of risks, as it translates in a higher percentage of late reporting claims than standard insurance branches.

– At this point, the Company gathers all information necessary to file the claim on its systems, to open a case reserve and to operate additional verifications on the claim's validity. If the outcome of these procedures is positive, the Insurer dispenses a reimbursement to the Transferor, otherwise the claim is closed with *no follow-up*. In both cases, this marks the *closing date* and a resulting *closing delay*, denoted as c.
– Though rare in practice for these kind of contracts, it is possible for any counterparty to request a *re-opening* of the claim, setting a *re-opening date* and the start of a second verification procedure which will result once again in a payment or in a closure with no follow-up.

3 Claims Reserve

Solvency II Regulation [4] requires the set up of Technical Provisions on a market-consistent basis, splitting the total amount for non hedgeable risks in the sum of Best Estimate and Risk Margin. The calculation of Claims Best Estimate is often based on the local Claim Reserve, given by

$$\mathbb{E}^P\left[X_{t,R}\middle|\mathcal{F}_t\right], \tag{1}$$

where t denotes the evaluation date, R the maximum admissible reporting delay according to empirical observation, P the natural probability measure, $\mathcal{F}_t$ the information available in t and $X_{t,R}$ the total liabilities for claims outstanding in t.
Excluding both the possibilities for claims to be settled with more than one payment and to be re-opened once they are closed, due to scarce materiality of these phenomena, we can further expand

$$X_{t,R} = \sum_{a=0}^{t}\sum_{r=0}^{R} X_{a,r}(c), \tag{2}$$

where $X_{a,r}(c)$ is the outstanding liability for a claim occurred in a, varying from the birth of the Company to the evaluation date, having reporting delay r and that will be settled in $a+r+c$.
Furthermore, the total local reserve can be divided in two main components that need separate evaluation because, as we can see in Fig. 2, each claim can be in one of two different states at a certain time t:

1. *Reported But Not Settled*, or *RBNS*, if the claim has been reported to the Insurer, but has not been closed yet (with or without follow-up);
2. *Incurred But Not Reported*, or *IBNR*, if the claim has already taken place but is yet to be reported to the Insurer.

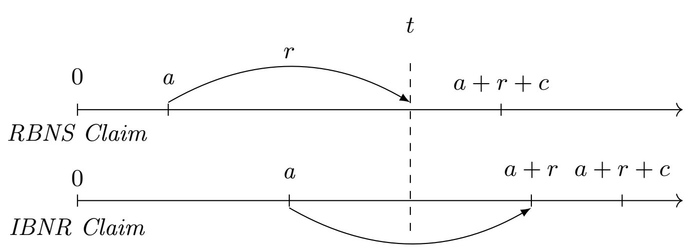


Fig. 2: Two different states for claims at a generic evaluation date t.

We can hence expand (1) as

$$\begin{aligned}\mathbb{E}^P\left[X_{t,R}\middle|\mathcal{F}_t\right] &= \underbrace{\sum_{a=0}^{t}\mathbb{E}^P\left[\mathbb{1}_{[a<t,a+r\leq t]}X_{a,r}(c)\middle|\mathcal{F}_t\right]}_{RBNS} + \\ &+ \underbrace{\sum_{n=0}^{N}\mathbb{E}^P\left[\mathbb{1}_{[n,\, r\leq R,\, a<t,\, a+r>t]}X_{n,r}(c)\middle|\mathcal{F}_t\right]}_{IBNR},\end{aligned} \tag{3}$$

where the two addendums represent RBNS and IBNR claims respectively, identified by the indicator functions accordingly to Fig. 2.
It can be noticed that IBNR liabilities $X_{n,r}(c)$ have a different subscripts from (2) as they need to be evaluated for every contract in force $n = 1, \dots, N$ at the evaluation date.

For our purposes we supposed that historical data $\mathbf{x}(t)$ collected by the Company contains all available information in t, even when we drop the index for simplicity, and that this is sufficient to give a proper evaluation of claim amounts.
Concerning RBNS component, we need an estimation of the closing delay for every claim, because of its strong influence on the final claim amount. It is in fact plausible presuming that an onerous claim will be subjected to additional verifications and, viceversa, a claim needing exceptional procedures will be more likely to result in a higher liability.
We hence need a proper classification model to esteem probabilities for the random variable closing delay c of each opened claim

$$\hat{P}\Big(c = c^* \Big| \mathcal{F}_t\Big) = \hat{P}\Big(c = c^* \Big| \mathbf{x}(t)\Big). \tag{4}$$

Once this model is tuned we can provide an estimation of the vector of closing delays $\mathbf{c}$, selecting for every claim the c^* value with higher estimated probability, completing the information $\tilde{\mathbf{x}}(t) = \{\mathbf{x}(t), \mathbf{c}^*\}$.
At this point we have everything we need to calibrate a statistical learning tool capable of approximating the relationship described by observed data between final payments and claim characteristics. We will use this as a regression function to provide an estimation of outstanding liabilities for RBNS claims

$$\hat{X}_{a,r}(c) = f_1\Big(\tilde{\mathbf{x}}(t)\Big). \tag{5}$$

For IBNR appraisal, where we dispose of less information as we could not observe claim characteristics, we opted for a *frequency-severity* approach, where we discern among in force contracts the ones that are more likely to produce an IBNR claim and then we produce an estimation of the outstanding liability for each of these.
For the first task, we used a classification tool predicting the probabilities that each existing risk will cause a late reporting claim in a fixed time interval r^*

$$\hat{P}\Big(\mathbb{1}_{[n,\, r \leq r^*,\, a<t,\, a+r>t]} \Big| \mathcal{F}_t\Big) = \hat{P}\Big(\mathbb{1}_{[n,\, r \leq r^*,\, a<t,\, a+r>t]} \Big| \mathbf{x}_n(t)\Big) = p_{\mathbb{1}}\big(\mathbf{x}_n(t)\big). \tag{6}$$

Selecting a proper cut-off value k, we discerned IBNR claims from the rest of the portfolio according to the probability estimates produced by (6)

$$\mathbb{1}_{[n,\, r \leq r^*,\, a<t,\, a+r>t]} = \begin{cases} 1 & if\ p_{\mathbb{1}}\big(\mathbf{x}_n(t)\big) \geq k \\ 0 & if\ p_{\mathbb{1}}\big(\mathbf{x}_n(t)\big) < k \end{cases}. \tag{7}$$

We were hence able to provide an estimation of the outstanding liabilities for each risk marked as IBNR by means of a second regression function, similar to (5) but ignoring information on the claim

$$\hat{X}_{a,r} = f_2\Big(\mathbf{x}_n(t) \Big| \mathbb{1}_{[n,\, r \leq r^*,\, a<t,\, a+r>t]}\Big). \tag{8}$$

4 Statistical learning tools

In this section we provide a review of the statistical models used in this paper: Generalised Linear Models (GLM), Artificial Neural Networks (ANN) and Support Vector Machines (SVM).

4.1 Generalised Linear Models

GLMs can be considered a standard tool for actuaries, as they are commonly used and have already been proposed for granular reservation processes, see e.g. Zhou [21] and Taylor [16],

hence they can represent a valid benchmark for the analysis.
The main reasons for the popularity of GLMs are their effectiveness and simplicity in modeling different kinds of relationships underlying data.

Their essential idea consists in explaining the variability of data through a generalisation of the linear regression model as follows

$$\mathbf{Y} = g^{-1}(\mathbf{X}\boldsymbol{\beta}) + \boldsymbol{\varepsilon} = g^{-1}(\boldsymbol{\eta}) + \boldsymbol{\varepsilon} \tag{9}$$

where $\boldsymbol{\varepsilon}$ is an error term and $g(\cdot)$ a link function, linking the response variable $\mathbf{Y}$ to the linear predictor $\boldsymbol{\eta} = \mathbf{X}\boldsymbol{\beta}$, while $\boldsymbol{\beta}$ represents the vector of regression parameters to be estimated.

It is known that GLMs are based on two main assumptions:

1. $Y_1, \ldots, Y_n$ are i.i.d. and their distribution function belongs to exponential family, i.e. it can be expressed as
$$f(y, \theta, \lambda) = d(y, \lambda) \exp\left[\frac{y\theta - b(\theta)}{\lambda}\right],$$
where $y, \theta, \lambda \in \mathbb{R}$ and $b(\cdot), c(\cdot)$ are real-valued functions;

2. $\mathbb{E}(\boldsymbol{\varepsilon}) = 0$ and $\mathbb{V}(\boldsymbol{\varepsilon}) = \sigma^2 I$, implying that
$$g(\mathbb{E}(\mathbf{Y})) = g(\boldsymbol{\mu}) = \mathbf{X}\boldsymbol{\beta} = \boldsymbol{\eta}.$$

When using these techniques, our main concern is choosing a proper distribution function for $\mathbf{Y}$ and a link function $g(\cdot)$, as they shape the kind of relationship that will be captured by the model.
As for the link function, it is common to opt for a logarithmic link, because it induces convenient transformations on data, or for the *canonical link*, which assumes a known form for every distribution of the exponential family and benefits useful properties.
Regarding the distribution, the choice depends on the kind of problem we need to solve and the characteristics of the response variable. For our purposes, as reported in Table 1, we relied on Binomial distribution for classification problems and Gamma or Inverse Gaussian distributions for cost-regression problems.

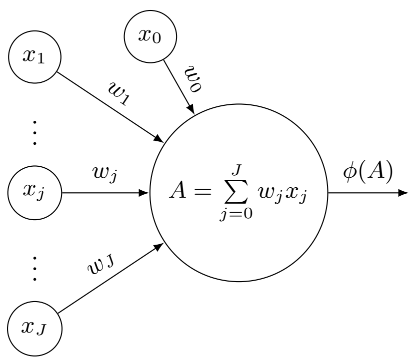
Distribution	**Canonical Link**	**Mean function**
Binomial	$\eta = \ln\left(\frac{\mu}{1-\mu}\right)$	$\mu = (1 + \exp(-\eta))^{-1}$
Gamma	$\eta = -\mu^{-1}$	$\mu = -\eta^{-1}$
Inverse Gaussian	$\eta = \frac{1}{\mu^2}$	$\mu = \eta^{-\frac{1}{2}}$

Table 1: Distribution choices and related canonical links.

4.2 Neural Networks

Though the fascinating biological inspiration, Artificial Neural Networks (ANN) are complex algorithms offering a valid alternative for problems where GLM hypothesis could be too restrictive.
Despite harder intepretation, their popularity is probably due to the *universal approximation theorem* [6], stating that a well-structured, feed-forward neural network can always approximate to any degree of accuracy any continuous function, even though it does not provide a way to find the best network configuration.

The elementary unit of an ANN is the *neuron*, or *node*, transforming an input $\mathbf{x} = x_1, \ldots, x_J$ to an output $\phi(A)$ as follows



1. Calculate the entity of each input $x_j \cdot w_j$, $\forall j$, considering a bias unit $x_0 = -1$;
2. Calculate the *Activation* as the sum of each input $A = \sum_{j=0}^{J} w_j x_j$;
3. Transform A through *activation function* $\phi(\cdot)$;
4. Transmits the output $\phi(A)$ to following neuron.

Fig. 3: Schema of a neuron.

A neural network is the result of the combination of several neurons in multiple *layers*. Usually all of the units on a layer are assigned with the same activation function, but it is possible choosing different functions for each layer. The assortment for $\phi(\cdot)$ is more than vast, being possibile to use any kind of function but, to speed up the calibration procedure, it is recommended selecting a function whose derivative is simple to compute. Some common choices are reported in Table 2.

Type	**Function**
Sigmoid	$\phi(A) = \frac{1}{1+e^{-kA}}$
Tanh	$\phi(A) = \frac{2}{1+e^{-kA}} - 1$
ReLu	$\phi(A) = \max[0, A]$
Softplus	$\phi(A) = \ln(1 + e^A)$

Table 2: Common choices for activation functions.

Special considerations are necessary for the output layer, whose activation function determines the shape of the final result of the net: for example, in a classification problem a Sigmoid output unit is desirable, as it produces results in [0,1], while for a regression problem admitting values in $\mathbb{R}$ a linear output unit could be preferred.

We have restricted our analysis to three-layered, feed-forward networks, as shown in Fig. 4, where information flows from input layer towards output layer with no possibility for recurrent connections.
The choice of such configuration is motivated for both the strength of universal approximation theorem and simplicity, as empirical evidence shows that such configuration is sufficient to solve most problems.

It has to be noticed that, while the numbers of input and output nodes J and O always equal the number of explicating and response variables respectively, choosing a proper amount of hidden units H is crucial for a good fitting.
Furthermore, we denoted as $h = 0$ and $o = 0$ two bias units, commonly set equal to -1 to help containing over-fitting and, indicating with $\phi_1(\cdot)$ and $\phi_2(\cdot)$ the activation functions chosen for hidden and output layers respectively, the final result of the net is given by

$$o(\mathbf{x}, \mathbf{w}, H) = \phi_2 \left(\sum_{h=0}^{H} w_h \phi_1 \left(\sum_{j=0}^{J} w_{hj} x_j \right) \right) \tag{10}$$

where w_{hj} denotes the weight connecting j-th input node and h-th hidden node, while w_h the weights between h-th hidden neuron and the output one.

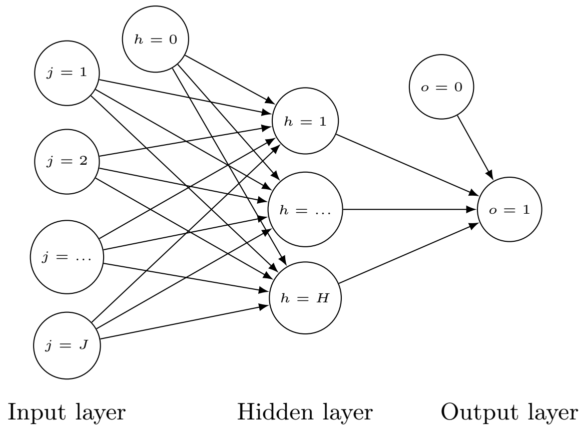


Fig. 4: Example of three-layered, feed-forward ANN.

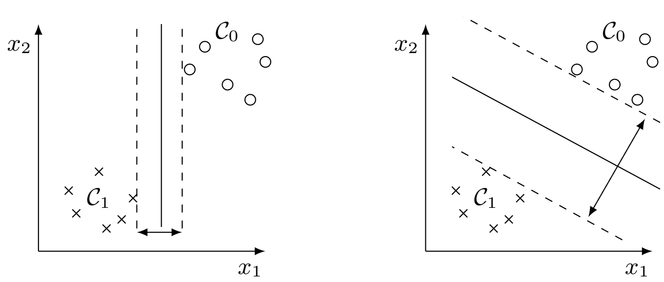
Now, through supervised learning algorithms, we can teach the network to replicate a phenomenon according to a set of observations $\mathbf{x}_n, n = 1, \ldots, N$ where the correct response $y_n, n = 1, \ldots, N$ is known.
This is accomplished by gradually correcting the weight vector $\mathbf{w}$ until the one that minimises a chosen loss function L is found. A common choice is a quadratic loss considering an L2 regularisation term, governed in its intensity by a parameter α, i.e.

$$\min_{\mathbf{w},H,\phi(\cdot),\alpha} \quad L = \sum_{n=1}^{N} (o_n(\mathbf{w}, H, \mathbf{x}_n) - y_n)^2 + \frac{1}{2}\alpha||\mathbf{w}||^2. \tag{11}$$

The first learning algorithm taylored for ANN's learning is Back-propagation [12], a gradient descent method proposing to correct weights of the hidden layer proportionally to their contribution to the final output, but numerous improved variants have been introduced in time, e.g. Quick-prop [5] or Resilient-prop [11].

4.3 Support Vector Machines

Introduced by Vapnik [3], Support Vector Machines (SVM) represent a solid alternative to common machine learning tools.
The idea underlying this technique is to find, among all possible linear solutions to the problem, the one being the farthest from observed data, as it should be more resilient against noise. In Fig. 5 we can see how this is translated in a linearly separable, binary classification problem.



(a) Smallest margin (b) Largest margin

Fig. 5: Optimal solution for SVM.

In such illustrative case, our objective is to find a linear decision boundary d_n maximising the so-called *margin*, i.e. the "tube" between observed data, shaped in Fig. 5 by two dotted lines.
Given two classes $\mathcal{C}_0, \mathcal{C}_1$, a convenient choice is $d_n = sign(\mathbf{w}^T\mathbf{x}_n + b)$, implying that

$$\begin{cases} d_n = +1 & if\, \mathbf{x}_n \in \mathcal{C}_1 \Leftrightarrow y_n = +1 \\ d_n = -1 & if\, \mathbf{x}_n \in \mathcal{C}_0 \Leftrightarrow y_n = -1 \end{cases}. \tag{12}$$

It can be shown that the margin equals $M = \frac{2}{||\mathbf{w}||}$, hence maximising the margin equals minimising

$$\begin{aligned} \min \quad & \frac{1}{2}\mathbf{w}^T\mathbf{w} \\ sub \quad & y_n[\mathbf{w}^T\mathbf{x}_n + b] \geq 1 \quad \forall n = 1, \ldots, N \end{aligned} \tag{13}$$

which is a quadratic optimum problem with n convex constraints - requiring that each data point is correctly classified - that admits a global optimum.
Problem (13) can be solved by means of Lagrange multipliers approach, which in dual space assumes the following form

$$\begin{aligned} \max_{\boldsymbol{\lambda},\mathbf{w},\mathbf{b}} \quad & \mathcal{L} = \sum_{n=1}^{N} \lambda_n - \frac{1}{2}\sum_{n,m=1}^{N} y_n y_m \lambda_n \lambda_m \mathbf{x}_n^T\mathbf{x}_m \\ sub \quad & \lambda_n \geq 0 \quad \forall n = 1, \ldots, N \\ & \sum_{n=1}^{N} \lambda_n y_n = 0 \end{aligned} \tag{14}$$

where λ_n are the multipliers and whose solution is given by

$$\begin{aligned} \mathbf{w}^* &= \sum_{n=1}^{N} \lambda_n^* \, y_n \, \mathbf{x}_n \\ b^* &= \frac{1}{N_{SV}} \cdot \sum_{k=1}^{N_{SV}} \left(y_k - \mathbf{x}_k^T\mathbf{w}^*\right). \end{aligned} \tag{15}$$

Equation (15) gives the name to the algorithm: the *support vectors* are the only input vectors whose multiplier is different from 0, hence contributing to the final solution.
It is worth noticing that, while for ANN one has to choose *ex-ante* a proper configuration for the network and then optimise it, when using SVM the optimal configuration is automatically derived during learning procedure.
Furthermore, (14) only depends on the dot product of couples of input vectors $(\mathbf{x}_n\mathbf{x}_m)$ and this can provide insight on both how inputs contribute to final solution and how to deal with more complex problems. It can be in fact shown that, if we transformed data through a function $\phi(\cdot)$, the Lagrangian in (14) would depend only on $\phi(\mathbf{x}_n)\phi(\mathbf{x}_m)$. This could prove useful because a non-linear problem could result linearly separable in a transformed space, as we can glimpse in Fig. 6.

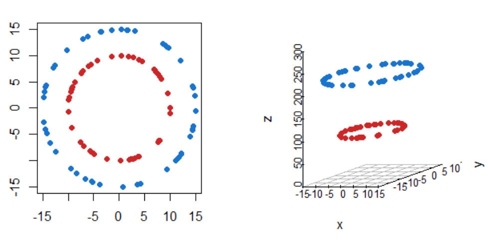


Fig. 6: Example of linear separability in a transformed space

This transformation can then be optimised by means of the so-called *kernel trick*, i.e. employing in equation (14) *kernel functions* $K((\mathbf{x}_n, \mathbf{x}_m)) = \phi(\mathbf{x}_n)\phi(\mathbf{x}_m)$, that allow the quantification of vector similarity in transformed spaces without computing dot products.
Common kernel choices are reported in Table 3 and, even though it is impossible to detect *ex-ante* the best kernel transformation, radial kernel has shown empirically the best performances overall.

Kernel	Function
Linear	$K(\mathbf{x}_n, \mathbf{x}_m) = (\mathbf{x}_n^T \mathbf{x}_m)$
Polynomial	$K(\mathbf{x}_n, \mathbf{x}_m) = [(\mathbf{x}_n^T \mathbf{x}_m) + 1]^k$
Sigmoid	$K(\mathbf{x}_n, \mathbf{x}_m) = \tanh[(\mathbf{x}_n^T \mathbf{x}_m) + b]$
Gaussian RBF	$K(\mathbf{x}_n, \mathbf{x}_m) = \exp\left[\frac{\lvert\lvert\mathbf{x}_n - \mathbf{x}_m\rvert\rvert^2}{2\sigma^2}\right]$

Table 3: List of most common kernel functions.

Generalisation for non perfectly-separable problems, that are more common in practice, is easily made by introducing a parameter C to delimit the influence of misclassified training points. Adopting a radial kernel, this translates (14) into

$$\begin{aligned} \max_{\mathbf{w},b,C,\sigma^2} \quad & \mathcal{L} = \sum_{n=1}^{N} \lambda_n - \frac{1}{2} \sum_{n,m=1}^{N} \lambda_n \lambda_m y_n y_m \exp\left[\frac{||\mathbf{x}_n - \mathbf{x}_m||^2}{2\sigma^2}\right] \\ sub \quad & 0 \leq \lambda_n \leq C \quad \forall n = 1, \ldots, N \\ & \sum_{n=1}^{N} \lambda_n y_n = 0 \end{aligned} \tag{16}$$

For further insight and for regression-SVM we refer to Kecman [7].

5 RBNS reserve estimation

We considered real data provided by an Italian company, considering 15 years of historical information collected up to 31.12.2016. Explicating variables deduced from available information can be divided in the first three of the following groups:

Policyholder

- Age in years
- Working seniority in years
- Type of administration
- Accrued severance pay
- Premium paid

Loan

- Duration in year
- Rate amount in euros
- Net annual percentage rate
- Number of mortgage rates paid
- Loan type

Claim

- Event type, such as *resignation*, *death*, etc.
- Re-opening, a flag {0-1}
- Reporting delay in years

Macro-economic

- GDP value at underwriting
- Loan default rate at underwriting
- GDP value at accident date
- Loan default rate at accident date

We decided to include macro-economic variables to link claim information to economic trends, known to have a significant influence on credit insurance. We considered for simplicity only two indexes, namely Italian GDP and Italian loan default rates for consumer credit, sourcing data respectively from ISTAT[1] and Bank of Italy [2], but presented methodologies can easily be adapted to include additional information.

As general calibration settings, we opted for a 80%-20% train-test split, applying a 5-fold cross validation strategy [9] on train set for parameter tuning, choosing a Sigmoid activation for hidden units of neural networks and a Gaussian RBF kernel for SVM, employing respectively the set ups described in (11) and (16).
To avoid numerical dominance phoenomena, we standardised every numerical input to have 0 mean and unitary variance.
For RBNS estimation, as shown in Section 3, our first task is to complete the information on claims by estimating closing delays. Because CQ insurance is characterised by quick liquidation processes, we have very few observation whose closing delay is larger than 3 years,

hence we considered for simplicity only annual time intervals, grouping all claims having closing delay $c \geq 3$, restricting hence the variable's domain in $C \in \{0, 1, 2, \geq 3\}$.
This configures as a multi-classification procedure with unbalanced classes, so we opted for *mean per class error* (*MPCE*) as metric to tune and evaluate each of the three different algorithms presented in Section 4. The final performances on the test set are reported in Table 4.

(a) GLM Performance

Actual \ Predicted	0	1	2	3+	*error*
0	75.5%	0.2%	0.3%	0.2%	*0.9%*
1	13.9%	0.4%	0.5%	0.3%	*97.3%*
2	4.7%	0.3%	0.6%	0.4%	*89.8%*
3+	1.7%	0.1%	0.4%	0.6%	*79.4%*
					MPCE = 66.85%

(b) ANN Performance

Actual \ Predicted	0	1	2	3+	*error*
0	74.5%	0.9%	0.4%	0.4%	*2.3%*
1	12.6%	1.7%	0.6%	0.2%	*88.9%*
2	3.7%	0.5%	1.4%	0.4%	*76.5.%*
3+	1.3%	0.1%	0.2%	1.1%	*60.1%*
					MPCE = 56.90%

(c) SVM Performance

Actual \ Predicted	0	1	2	3+	*error*
0	74.8%	1.0%	0.3%	0.1%	*1.8%*
1	13.7%	1.3%	0.0%	0.0%	*91.6%*
2	0.3%	0.0%	5.8%	0.0%	*4.5%*
3+	0.1%	0.0%	0.0%	2.7%	*3.0%*
					MPCE = 25.23%

Table 4: Confusion matrices on test set, expressed as percentage of total test claim number.

The selected models have hence been used to produce an estimation of closing delay for every RBNS claim, which we used among other explicating variables to regress on claim cost. Parameter tuning of (5) has been conducted using *Root Mean Squared Error* (*RMSE*) as metric and we reported the final performances on the test set in Table 5, alongside with the final estimation of RBNS reserve. It should be noticed that both RMSE and reserve amounts are expressed in comparison with the actual RBNS reserve registered in the Company's balance sheet at 31.12.2016.

Model	**MPCE** (Closing delay)	**RMSE** (Claim cost)	**Reserve** (Amount)	**Capacity** (Closed claims)
Internal	NA	NA	100%	148.4%
GLM (InvG)	66.85%	0.0617%	84.04%	104.3%
GLM (Gamma)	66.85%	0.0731%	98.07%	125.3%
ANN	56.90%	0.0619%	84.82%	118.3%
SVM	25.23%	0.0553%	86.40%	112.1%

Table 5: Estimation result for RBNS reserve, where reserve amounts and RMSE are expressed in comparison with the actual amount registered on the balance sheet.

In this table we can easily deduce that SVM shows the best performances both in terms of MPCE and RMSE, and that all of the calibrated models suggest a slight decrease in reserve amount with respect to the internal methodology, which proves to be the most prudent.
Sufficiency to cover future actual payments has been checked in comparison with final payments for every claim that has been closed at 31.12.2017, expressing *capacity* as reserved amounts on actual payments ratio.
For further inspection, we decided to conduct a back-testing to check the reliability of these methodologies, repeating the same analysis for RBNS claims at 31.12.2015 and 31.12.2014, comparing the results with actual paids at 31.12.2017 for every claim that has been closed in this time lapse. The outcome of the back-testing, shown in Table 6, has proven positive even for 2014, a particularly onerous year for the Company.

(a) RBNS 2015

Model	MPCE	RMSE	Capacity
Internal	NA	NA	148.6%
GLM (InvG)	68.79%	0.0589%	138.7%
GLM (Gamma)	68.79%	0.0804%	156.3%
ANN	59.23%	0.0586%	140.3%
SVM	24.94%	0.0533%	133.9%

(b) RBNS 2014

Model	MPCE	RMSE	Capacity
Internal	NA	NA	123.4%
GLM (InvG)	71.44%	0.0615%	100.1%
GLM (Gamma)	71.44%	0.0919%	117.2%
ANN	64.97%	0.0634%	100.4%
SVM	25.77%	0.0555%	103.2%

Table 6: Summary of back-testing results.

6 IBNR reserve estimation

Concerning IBNR component, it is known that claim frequency in credit insurance is *non-stationary* and highly volatile, hence the temporal dimension acquires fundamental value for an accurate estimation of future liabilities.
As shown in Fig.7, the dataset used for calibration is composed by the aggregation of the portfolios observed at the three most recent evaluation dates for which all of the IBNR claims have been observed, considering a time cap for late reporting equal to $r^* = 3$ years, one year larger than the prescription limit for CQ claims.

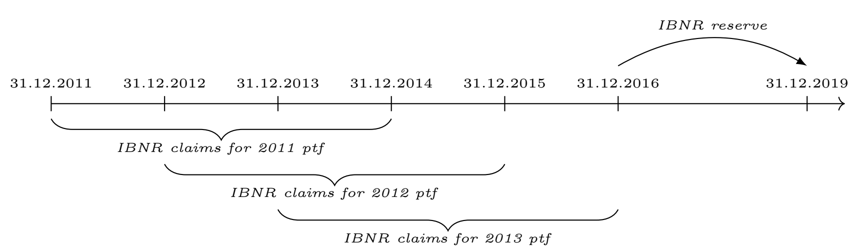
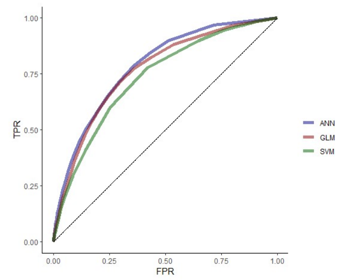


Fig. 7: Dataset construction schema for IBNR reserving.

Referring to explanatory variables presented in Section 5, to calibrate tools described in (6) and (8) we cannot rely on claim information, needing hence to concentrate on policy-holder, loan and macroeconomic groups to explain the variability of IBNR claims.
Macroeconimic indices have been observed at each contract underwriting and each evaluation date and, being the calibration unique for the three portfolios, they serve to outline time trends in data.

Calibration settings are similar to RBNS component, applying an 80%-20% train-test split and a 5-fold cross validation strategy on train set for parameter tuning.
Performance on test set and parameter tuning has been evaluated in terms of area under curve (AUC) described by Receiver Operating Characteristic (ROC) for (6) and RMSE for (8).
Concerning IBNR claims classification, due to strong class unbalance, we opted for a cut-off value maximising F1 score [14], which is the result of an average between precision and recall. The chosen cut-off values are reported in Table 7, alongside with the final performances on test set.
For regression on claim cost, data used is the same from Section 5, excluding information on claim characteristics. Final results of the reserve estimation for each IBNR-classified contract are reported in Table 8 where we included Chain Ladder for comparison with standard actuarial tools. As for RBNS estimation, we reported RMSE and IBNR reserve amounts in comparison with the amount registered on the balance sheet at 31.12.2016 of the Company, while the number of claims is always expressed proportionally to the number estimated by means of the internal estimation technique, denoted as N_c.

(a) ROC comparison



(b) Cut-offs maximising F1 score

Model	Cut-off	AUC
GLM	0.03	0.7707
ANN	0.04	0.7884
SVM	0.04	0.7370

Table 7: Principal metrics for IBNR classification.

	Test Performances		**Reserve Estimation**	
Model	**AUC** (Frequency)	**RMSE** (Severity)	**Claim Number**	**Reserve Amount (,000)**
Internal	NA	NA	N_c	100%
Chain Ladder	NA	NA	$0.97 \cdot N_c$	82.74%
GLM (InvG)	77.07%	0.0129%	$1.07 \cdot N_c$	119.54%
GLM (Gamma)	77.07%	0.0276%	$1.07 \cdot N_c$	121.56%
ANN	78.84%	0.0128%	$0.77 \cdot N_c$	94.90%
SVM	73.70%	0.0124%	$1.10 \cdot N_c$	81.48%

Table 8: IBNR estimation comparison.

7 Computational Effort

All of the analysis has been conducted via R software [10], using *H2O* [17] package for GLM, *nnet* [18] package for neural networks, and *liquidSVM* [15] for SVM.

We chose *nnet* for its simplicity and quickness, as it implements only feed-forward networks with sigmoidal activation functions for hidden neurons and a loss function with L2 regularisation which is solved by means of BFGS algorithm [13]. This translates in a simple grid search over only 2 hyper-parameters: hidden neurons amount H and the intensity α of the regularisation parameter.

As for *liquidSVM*, it provides one of the fastest SVM calibration tools but only implements radial gaussian kernel. Because it solves the optimum problem in the dual space, the only hyper-parameters to be tuned are the shape parameter of kernels σ^2 and the C parameter to delimit training error.

In Table 9 we report computational timings for IBNR parameter tuning, which requires the heaviest computations. Timings are expressed in minutes and have been estimated using a computer having Intel i7-8550U CPU, 2 GHz and 16 GB RAM. The grid for ANN tuning in frequency classification has been reduced with respect to others to contain time expense. Final models are described in Table 10, where their complexity is explained in terms of number of weights and we report their respective computational timings.

	Frequency				**Severity**			
Model	Grid size	Data size (,000)	Total timing	Average timing	Grid size	Data size (,000)	Total timing	Average timing
ANN	55	854	391.37	7.12	130	55	90.55	0.70
SVM	64	854	247.37	3.87	64	55	19.58	0.31

Table 9: Timing summary for IBNR parameter tuning in minutes.

	Frequency		Severity	
Model	Weights	Timing	Weights	Timing
ANN	96	3.38	571	1.59
SVM	533	1.23	475	0.06

Table 10: Summary of calibration timing in minutes and complexity of selected models.

Conclusion and outlook

We have shown how machine learning techniques can exploit individual and macroeconomic information for claims reserving in credit insurance, producing an estimation of both RBNS and IBNR components of the local Claims Reserve.
GLMs have represented our benchmark, due to their proven effectiveness in several actuarial problems. Though simple and easy to understand, they have several restrictions and are heavily affected by outliers or collinearity in explicating variables.
Though harder to explain, Neural Networks could represent a valid alternative when the structural hypothesis of GLMs are too restrictive, as they are able to detect even higly non-linear relationships. Furthermore they can be arbitrarily complex, at the cost of increasing over-fitting risk and computational timings.
On the other hand, SVMs can be easier to interpret, are granted with a structure more stable and resilient, and do not require architecture designing. As downside, kernel matrix computation can often be quite challenging for most calculators, resulting in heavy tuning timings.
While all of the presented methodologies have achieved both comparable results and capacity with respect to actual payments, SVMS seem to have shown the best training metrics overall. However, to express a final preference for any of these estimators we need to explore their variability, which will be object of further developments alongside with a sharpening of tuning procedure. This further step will allow to appraise both the Best Estimate and Risk Margin, necessary to the evaluation of Technical Provision in a Solvency II framework.

Acknowledgments

The author wishes to thank Professor Luca Passalacqua for providing precious feedbacks on this paper.

References

[1] Base dati ISTAT, `http://dati.istat.it/`.
[2] Base dati statistica Banca d'Italia, `https://www.bancaditalia.it/statistiche/basi-dati/bds/index.html`.
[3] Cortes, C. and Vapnik, V. (1995), *Support-vector networks.*, In Machine learning, Vol.20, N. 3, pp. 273-297, Springer.
[4] EU Parliament (2009), *Directive 2009/138/EC*, European Union Official Journal.
[5] Fahlman, S. E. et al. (1988), *An empirical study of learning speed in back-propagation networks.*, Carnegie Mellon University, Computer Science Department.
[6] Hornik, K., Stinchcombe, M. and White, H. (1990), *Universal Approximation of an Unknown Mapping and Its Derivatives Using Multilayer Feedforward Networks*, In Neural Networks, Vol. 5, pp. 551-560.
[7] Kecman, V. (2005), *Support Vector Machines - An Introduction.*, In Wang L. (eds) Support Vector Machines: Theory and Applications. Studies in Fuzziness and Soft Computing, Vol. 177, pp. 1-47, Springer.
[8] Kirlidog, M. and Asuk, C. (2012), *A fraud detection approach with data mining in health insurance*, In Procedia-Social and Behavioral Sciences, Vol. 62, pp. 989–994, Elsevier.
[9] Stone, M. (1974), *Cross-Validatory Choice and Assessment of Statistical Predictions.*, In Journal of the Royal Statistical Society. Series B, Vol. 36, N. 2, pp. 111-147.

[10] R Foundation for Statistical Computing, *R: A Language and Environment for Statistical Computing.*, https://www.R-project.org/.

[11] Riedmiller, M. and Braun, H. (1993), *A direct adaptive method for faster backpropagation learning: The RPROP algorithm.*, In Neural Networks, IEEE International Conference on, pp. 586-591.

[12] Rumelhart, D. E. and Hinton, G. E and Williams, R. J. (1986), *Learning representations by back-propagating errors.*, In Nature, Vol. 323, pp. 533-536.

[13] Shanno, D. F. (1970), *Conditioning of quasi-Newton methods for function minimization*, In Mathematics of computation, Vol. 24, N. 111, pp. 647-656.

[14] Sokolova, M. and Japkowicz, N. and Szpakowicz, S. (1970), *Beyond Accuracy, F-Score and ROC: A Family of Discriminant Measures for Performance Evaluation.*, In AI 2006: Advances in Artificial Intelligence, Lecture Notes in Computer Science. Vol. 4304, pp. 1015-1021.

[15] Steinwart, I. and Thomann, P. (2017), *liquidSVM: A Fast and Versatile SVM package*, ArXiv e-prints 1702.06899, http://www.isa.uni-stuttgart.de/software.

[16] Taylor, G. and McGuire, G. (2004), *Loss reserving with GLMs: a case study*, Meeting of the Casualty Actuarial Society, Colorado Springs, Colorado.

[17] The H2O.ai team, *h2o: R Interface for H2O*, https://CRAN.R-project.org/package=h2o.

[18] Venables W. N. and Ripley B. D. (2002), *Modern Applied Statistics with S.*, Springer, http://www.stats.ox.ac.uk/pub/MASS4.

[19] Wüthrich, M. V. (2018), *Machine learning in individual claims reserving*, In Scandinavian Actuarial Journal, pp. 1-16, Taylor & Francis.

[20] Wüthrich, M. V. (2018), *Neural networks applied to chain-ladder reserving.*

[21] Zhou, J. and Garrido, J. (2009), *A loss reserving method based on generalized linear models.*, In Society of Actuaries.