Reconstrução de métricas atuariais através do *stacking* de modelos de aprendizagem de máquina

Data reconstruction of two actuarial metrics by staking machine learning models

Reconstrucción de dos métricas actuariales mediante el *stacking* de modelos de aprendizaje de máquina

Amaury de Souza Amaral[1], Jardel Marques Monti[2] e Segundo Parra Milián[3]
[1]Pontifícia Universidade Católica de São Paulo
[2]Pontifícia Universidade Católica de São Paulo
[3]Instituto de Física Teórica – IFT - Universidade Estadual Paulista – São Paulo

Autor correspondente: Amaury de Souza Amaral

E-mail: amaury.s.amaral@gmail.com

Resumo

Objetivo: Grande parte da saúde dos brasileiros é financiada pelos planos de saúde cujos reajustes têm sido motivo de questionamentos nos tribunais. Dada a dificuldade em se obter informações nem sempre disponíveis nos processos judiciais, para a reconstrução dos dados, elaboramos uma métrica por meio de técnicas de *Deep Learning* para obtermos tais informações. **Método:** Após analisar os dados obtidos através do Órgão Regulador, treinamos três diferentes algoritmos de aprendizagem supervisionada objetivando obter informações por meio de um problema de otimização. Utilizamos o método Lagrangiano Aumentado com o objetivo de incluir as restrições na função de custo e do *Simulated Annealing* para minimizá-la. **Resultados:** Consistente como era de esperar, o desempenho do empilhamento superou o desempenho dos aprendizados de base. **Conclusões:** Com os resultados obtidos foi possível obter as informações de custo médio por sinistro e frequência retroativos, buscados do "passado do plano de saúde".

Descritores: Inteligência Artificial, Otimização de Processos, Saúde Suplementar.

Abstract

Objective: A large part of Brazilian's health care is financed by health insurance plans, which readjustments have been questioned in the courts. The data from court cases tends to not be readily available. Therefore, in order to reconstruct the data,

we developed a metric using Deep Learning techniques to obtain data estimations. **Method:** After analyzing the data obtained from the Regulatory Agency, we trained three different supervised learning algorithms aiming to obtain information through an optimization problem. We used the Augmented Lagrangian method aiming to include the constraints into the cost function and Simulated Annealing to minimize it. **Results:** Consistent as expected, the stacking performance outperformed the base learners. **Conclusions**: With the results obtained it was possible to obtain the retroactive average cost per claim and frequency information, fetched from the "health plan's past".

Keywords: Artificial Intelligence, Process Optimization, Supplementary Health

Resumen

Objetivo: Gran parte de la asistencia sanitaria de los brasileños se financia con planes de seguro médico cuyos reajustes han sido cuestionados en los tribunales. Dada la dificultad de obtener información que no siempre está disponible en las demandas, para la reconstrucción de datos, desarrollamos una métrica mediante técnicas de *Deep Learning* para obtener dicha información. **Método**: Tras analizar los datos obtenidos del Organismo Regulador, entrenamos tres algoritmos diferentes de aprendizaje supervisado con el objetivo de obtener información a través de un problema de optimización. Utilizamos el método Lagrangiano Aumentado para incluir las restricciones en la función de costo y el Recocido Simulado para minimizarla. **Resultados:** Tal y como se esperaba, el rendimiento del apilamiento superó a los aprendizajes de base. **Conclusiones:** Con los resultados obtenidos fue posible obtener la información de costo médio por siniestro y frecuencias retroactivas, obtenida del "pasado del plan de salud".

Descriptores: Inteligencia Artificial, Optimización de Procesos, Salud Complementaria.

Introduction

Health care services in Brazil are provided in two ways: the publicly funded Unified Health System (*SUS, Sistema Único de Saúde*) and the Supplementary Health Service Operators. To the best of our knowledge, there is no clear decoupling between the public and supplementary private systems. Medical cases that might

have been initially treated under private operators might be transferred to a public hospital.

The operators in the supplementary private system, in order to avoid serious disruption to the financial system, need to track over time information expressed in metrics, at least in the context of a stable business cycle, in which policyholders pay a fair and accurate premium. Two metrics are important to our work. The first one is the frequency rate (frequency, for short), which measures the number of expected claims, based in the exposure number. Moreover, frequency is important to determine the likelihood of paying out the medical costs in healthcare plans and subsequently planning for financial viability. The second one of interest is the average cost per claim (in Portuguese, *Custo Médio por Evento,* CMEV), which measures how much the provider pays out for each claim filled by its customers. Both frequency and CMEV are categorized according to the age group, the different types of hiring procedures and medical coverage, etc.

The above metrics are organized in the so-called actuarial technical reports (in Portuguese, *Notas Técnicas Atuariais,* NTA), these reports are the kernel of healthcare insurance plans. In order to maintain transparency, these actuarial technical reports are expected to be kept secure and available, according to the guidelines of the regulatory agencies.

The problem at hand

The NTAs are control and price-setting mechanisms of the product being sold. The NTAs forecast the costs and frequencies from which it will be possible to set the product's selling price, i.e., the premium. According to the evolutionary forecast of the NTA, it is possible to estimate the movement of the age group and to forecast the costs according to their users' aging.

In most situations, we do not have access to the required actuarial technical report. In 2020, however, an initiative from the regulatory agency made available a panel in Power BI format, giving information about the state of the healthcare insurance market from August 2015 to July 2020. The available data included metrics of frequencies and CMEVs. Still, there were many gaps in the data.

Approach to reconstruct the data

To the best of our knowledge, there is no dynamical equation describing healthcare insurance market dynamics. We only have data for five years. Indeed, the healthcare insurance market does not evolve on its own. It may depend on several factors (e.g., the development of demographic trends, migration, fertility, mortality, life expectancy, etc. indexes).

To reconstruct the values of frequency and CMEV before August 2015, we first reconstruct the frequency using the probability of survival $p_{x,n}$. To do that, we assume that the map f from the probability of survival to the frequency exists, i.e., $f(p_{x,n}) = Freq$ and assume that the map g from the frequency (Freq) to the CMEV, i.e., $g(Freq) = CMEV$ exists too. And both maps can be found in the previously mentioned data from August 2015 to July 2020. We used both maps to reconstruct the frequency and the CMEV before August 2015. However, it is not enough to find out the maps *f* and *g* by hand. Instead, we use three different regression machine learning algorithms: Neural Networks (NN), Xtreme Gradient Boosting (XGBoost) [(1)], and Support Vector Regressor (SVR, this is a regression version of Support Vector Machine [(2)]). Instead of fine-tuning the hyper-parameters on each model, we use stacked generalization (or stacking), which is a type of ensemble method to combine the models. See Ganaie H, et al [(3)] for more details on this method

Stacking is a way of combining multiple models (base learners) to achieve higher performance. It was proposed by Wolpert [(4)] and used for regression tasks [(5)]. The general idea is to have a set of m different predictors whose task is to generate predictions based on different partitions of the dataset. These predictions are then used as inputs and the best responses as outputs for training higher-level learners. We focus on how to obtain the optimal weights for combining the base learners, where we have used Simulated Annealing together with Augmented Lagrangian.

The remainder of the paper is organized as follows: in section II we provide an overlook of the collected data, describing the methodology used to reconstruct the metrics frequency and CMEV. Also, we present the base learners used in this work. Then, we describe how the stacking was achieved and present the equations for the integrated models. In section III we show results by using just Simulated Annealing

and using both Simulated Annealing and Augmented Lagrangian. We conclude in section IV, with discussions on the main results and possible future work.

Methods

About the data

In July 2020, the regulatory agency released a panel dataset in Power BI format on actuarial technical reports. We worked with data obtained from its 26th page. The data consist of categorical and numerical features. The categorical ones are Age groups, Coverages and Hirings. They are described in the following lines, where subscript i stands for the categorical variable that have more than one value.

- Age groups (Ag_i) consisting of ten values and they are 00 to 18 yrs, 19 a 23yrs, 24 to 28 yrs, 29 to 33 yrs, 34 to 38 yrs, 39 to 43 yrs, 44 to 48 yrs, 49 to 53 yrs, 54 to 58 yrs. 59 yrs or more.
- Coverages (C_i) consisting of six values. Admissions, Doctor's appointments, Exams, Other assistance expenses, other outpatient cares and Therapies.
- Hirings (H_i) consisting of three values. Business Collective, Individuals or families and Membership Collective.

The metrics are frequency and CMEV. The feature Frequency (Freq) is defined by

$$Freq = \frac{claims\ number}{exposure\ quantity} \qquad (1)$$

and stands for how many times health insurance was activated (claims number) by a population who signed up for health insurance (exposure quantity). The CMEV variable is defined by

$$CMEV = \frac{claims\ amount}{claims\ number} \qquad (2)$$

The data is organized monthly August 2015 to July 2020. In Table 1 we sum up the skewness, the maximum and minimum values of the collected data per every value of coverage C_i.

Table 1: Description of numerical features by coverage

Coverages	Frequency		CMEV[1]	
	Skewness	[Min., Max] (%)	Skewness	[Min., Max.]
C_1	0.9404	[34.7, 69.2]	-0.2122	[57,87]
C_2	0.4924	[43.1, 254]	-0.3664	[19, 38]
C_3	1.715	[0.8, 5]	1.293	[2656, 12255]
C_4	4.002	[12.1, 381.6]	0;9866	[45, 212]
C_5	1.156	[6.5, 31.3]	1.130	[64, 209]
C_6	0.8498	[4.5, 32.8]	1.034	[31, 161]

Note: (1) From the Portuguese abbreviation of *Custo Médio por Evento* (see p.3)

About the maps

The general idea for solving the problem at hand is to translate it into a regression one. For every coverage value (C_i) and a given age group (Ag_i), the following statements hold

- There is a function $F_{coverage}$ which maps the probability of survival $p_{x,n} = {}^{l_{x+n}}/_{l_x}$ (where l_m is the number of living people at age m) to the frequency (Freq.) defined in Equation (1)

$$F_{coverage}(p_{x,n},*) = Freq. \qquad (3)$$

  The probability of survival was collected from the Brazilian Institute of Geography and Statistics (in Portuguese, abbreviation for *Instituto Brasileiro de Geografia e Estatística,* IGBE).

- Also, there is a function $G_{coverage}$ which maps Freq. in Equation (1) to CMEV in Equation (2), i.e.

$$G_{coverage}(Freq.,*) = CMEV \qquad (4)$$

  where * in Equations. (3) and (4) stands for other values, such as Hirings and/or Age groups.

To find the functions in Equations (3) and (4), we used an optimization criterion, i.e., we aimed to minimize the loss function of the difference between the true value ϕ_{true} and the computed one ϕ_T, i.e.

$$\min \mathcal{L}(\phi_{true}, \phi_T) + CSTRS. \quad (5)$$

where $CSTRS.$ stands for possible constraints and/or bounds on the problem.

The base learners (BL)

To configure the Neural Network (NN) models we used Tensorflow [6] and the Keras packages. We used the scikit-learn library sklearn.svm.SVR and the XGBoost Python packages for the other two models.

In addition, we use NumPy [7] for linear algebra procedures and various tools available in the package scikit-learn for data processing. We split our dataset into 80% for training and 20% for testing.

The base learner: Neural Network

For the NN models, we transformed the categorical features into numerical ones using the entity embedding [8], which transforms an r-dimensional categorical variable into a vector of dimension specified by the user, in our case we set the dimension number given by

$$\dim_{number} = int(\sqrt{r+1}) \quad (6)$$

In Table 3 we show the transformations applied on the target values to reduce the skewness on the target distribution. In our NN models, we use four fully connected layers (128, 900, 256 and 64 neurons respectively). The fully connected layer uses the ReLU (Rectified Linear Unit) activation function. The output layer contains one neuron with sigmoid activation function. No dropout and regularization techniques were used in our NN models. We resize the transformed target $\tilde{t}$ with respect to its maximum $\widetilde{t_{max}}$. The Adam optimizer [9] was used when training each NN model with batch size 106.

Table 3: Transformations applied on the target values for coverage value

	Frequency		**CMEV[1]**	
Coverages	**NN[2]**	**XGBoost and SVR[3]**	**NN**	**XGBoost and SVR**
C_1	$\ln(t)$	$\ln(t+1)$	$identity$	$identity$
C_2	$\sqrt{t}$	$\ln(t+1)$	$identity$	$identity$

C_3	$\sqrt[3]{t}$	$\sqrt[3]{t + shift}$	$\sqrt{t}$	$\ln(t)$
C_4	$\ln(t)$	$\sqrt[10]{\ln(t+1)}$	$\sqrt[3]{t}$	$\ln(t)$
C_5	$\ln(t)$	$\ln(t+1)$	$\sqrt{t}$	$\ln(t)$
C_6	$\ln(t)$	$\ln(t+1)$	$\sqrt[3]{t}$	$\ln(t)$

Note: (1) From the portuguese abbreviation of *Custo Médio por Evento*. (2) Abbreviation for Neural Network and (3) Abbreviation for Support Vector Regressor.

The base learner: XGBoost Regressor

We trained four different XGBoost models. In all the four models we kept the learning rate fixed at 0.05 and used as parameters *max_depth*, *min_child_weight* and *n_estimators*. In our XGBoost models, we used label encoding to encode categorical features.

- When the target under consideration was Freq., we used max_dept = [6, 12], n_estimators= [500, 700] and min_child_weight = 7.
- And for the half cost per claims (CMEV) target, we used max_depth = [6, 12], n_estimators= [400, 700] and min_child_weight = 10.

The base Learner: Support Vector Machine and its Regressor

Also, we prepared four SVR models. In the four cases we used the radial basis function(rbf) as kernel function. When the frequency was the target under consideration the parameters C, γ and ϵ took the values [10, 30], 0.2 and $[3 \cdot 10^{-4}, 10^{-5}]$, respectively. And for the CMEV target [5,30], 0.1 and $[10^{-4}, 9 \cdot 10^{-4}]$, respectively.

Stacking the base learners

When training at zero level, i.e., at the base learners discussed previously, we divided the training data into six folds. We trained a base learner on each of the five folds and obtained the fold predictions on the remaining one. We averaged the predictions on the test set. It took about eight hours per target to train the base learners.

Predictions generated at the zero-level training were used as features for the stacked generalization models. As the initial stacked model, we used the weighted average $\omega_i \cdot M_i$(where repeated indices stands for summation) where ω_i are the

weights to optimize and M_i are the m models to be stacked. The weights ω_i are subject to $\sum_{i=1}^{m} \omega_i = 1$ and bounded to the interval [0, 1].

In addition to the weighted average stacking, we have explored two additional ways for integrating the models Equations (7) and (8)

$$\varphi_{STK}(\omega) = \omega_1 \varphi_{NN}^{\omega_2} \{\sum_{i=5}^{8} \omega_i \varphi_i^{xgb}\}^{\omega_3} + \omega_4 \{\sum_{i=9}^{12} \omega_i \varphi_i^{svr}\} \quad (7)$$

$$\varphi_{STK}(\omega) = \omega_1 \varphi_{NN} + \omega_2 \prod_{i=4}^{7} \{\varphi_i^{xgb}\}^{\omega_i} + \omega_3 \prod_{i=8}^{11} \{\varphi_i^{svr}\}^{\omega_i} \quad (8)$$

where the constraints on the weights ω_i are given by the vector functions $h(\omega)$ in Equations (9) and (10)

$$h(\omega) = (-1 + \sum_{i=5}^{8} \omega_i, -1 + \sum_{i=9}^{12} \omega_i, -1 + \omega_1 + \omega_4, \omega_2 + \omega_3 - 1)^T \quad (9)$$

$$h(\omega) = (-1 + \sum_{i=4}^{7} \omega_i, -1 + \sum_{i=8}^{11} \omega_i, -1 + \sum_{i=1}^{3} \omega_i)^T \quad (10)$$

Also, the weights are bounded to the interval [0, 1]. In fact, we are dealing with constrained optimization problems and to consider the constraints $h(\omega)$, we have used a common technique called the Augmented Lagrangian (AL) [10, 11]. Here, instead of minimizing the cost function $\mathcal{L}(\omega)$, the new objective to minimize is $\widetilde{\mathcal{L}_c}(\omega, \lambda)$ and is given by

$$\widetilde{\mathcal{L}_c}(\omega, \lambda) = \mathcal{L}(\omega) + \sum_j \lambda_j h_j(\omega) + \frac{1}{2} \sum_j c_j h_j^2(\omega) \quad (11)$$

Here $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_m)^T$ is a set of Lagrange multipliers. $c_j > 0, j = 1, \ldots, m$ are finite weighting factors. The introduction of the quadratic term with $h_i(\omega) = 0$ does not alter the optimal solution of the system and makes the right-hand side of Equation (11) strongly convex if c_j is larger. The method looks for an optimal solution ω_* by iteration, where the multiplier λ_j is updated by $\lambda_k = \lambda_{k-1} - c_k h_{k-1}(\omega_k)$ where ω_k is the solution to the unconstrained problem, Equation (11) at the kth step.

Then, we used the Simulated Annealing (SA) [12, 13]. The SA algorithm is inspired by the annealing technique in metallurgy, in which, the metal is heated to a high temperature and then is slowly cooled down. The resultant metal will be a resilient and malleable one.

The SA algorithm accepts all the solutions, the ones that decrease the cost function $\widetilde{\mathcal{L}_c}(\omega, \lambda)$ solutions that increase $\widetilde{\mathcal{L}_c}(\omega, \lambda)$, however, are also achieved under the condition

$$p = e^{-\frac{\Delta\widetilde{\mathcal{L}_c}(\omega,\lambda)}{T}} < random\ number \quad (12)$$

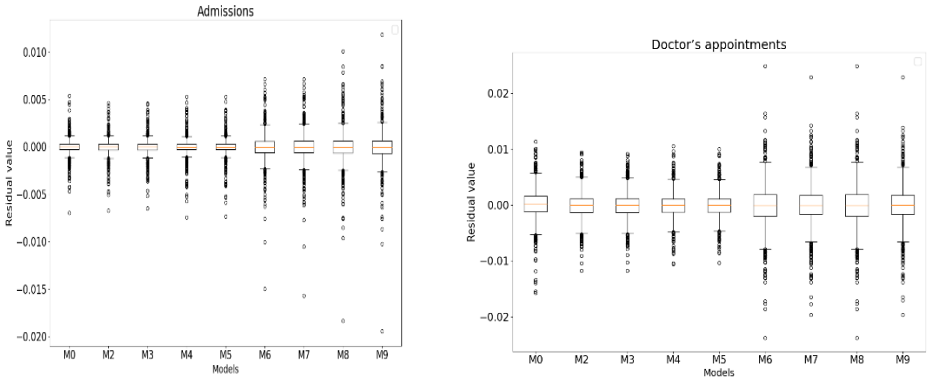
where $\Delta\widetilde{\mathcal{L}_c}(\omega, \lambda)$ is the variation in the objective function. T is the temperature at the beginning of the search process, being set to a high value. The lower the temperature the smaller the probability of transition. The temperature is updated following the cooling schedule, given by

$$T_{new} = \alpha T_{old} \quad (13)$$

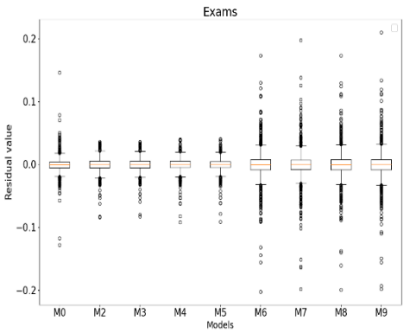
with $0 < \alpha < 1$. To guarantee convergence to a global minimum, the cooling schedule must be very slow. Indeed, the process can be seen as the following: Consider the function to be minimized as the energy function of a physical system, when cooled sufficiently, converges to a state of minimum energy slowly. This state represents the desired solution.

Results

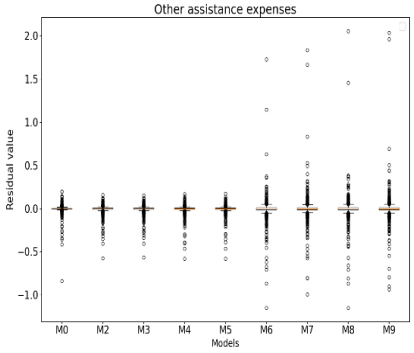
In figures 1 and 2 we show the distribution of residuals per each Coverage value at each base learner, we had nine base learners. At both figures M_i denotes the models. The objective when doing stacked generalization is to get a model which improves the performance of the base learner.



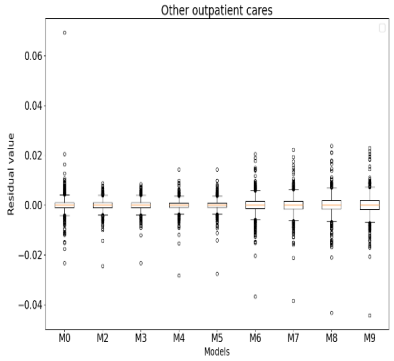
(a) C_1 (b) C_2



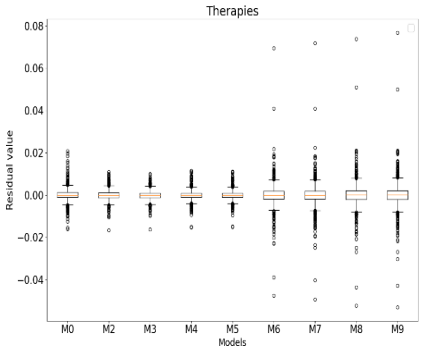
(c) C_3



(d) C_4

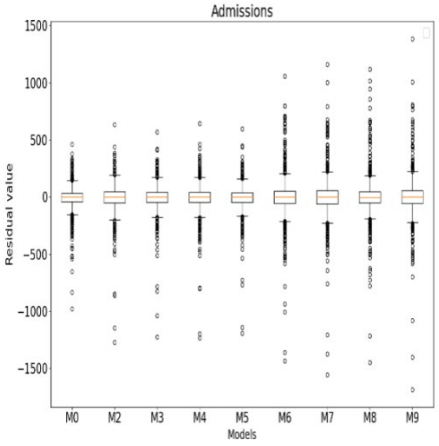


(e) C_5

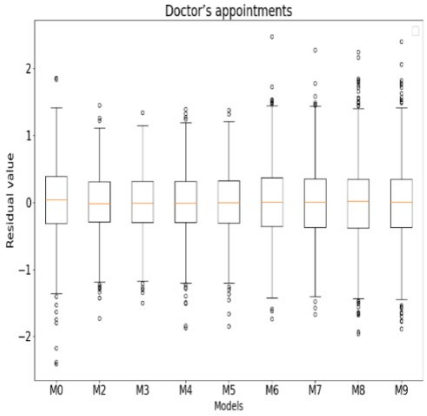


(f) C_6

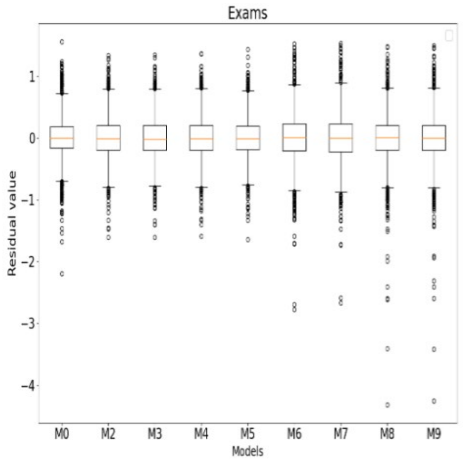
Figure 1: Distribution of Residual Values for each base learner when the target is Freq at the training set.



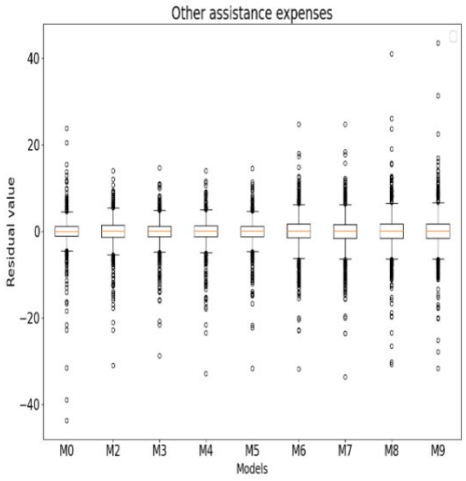
(a) C_1



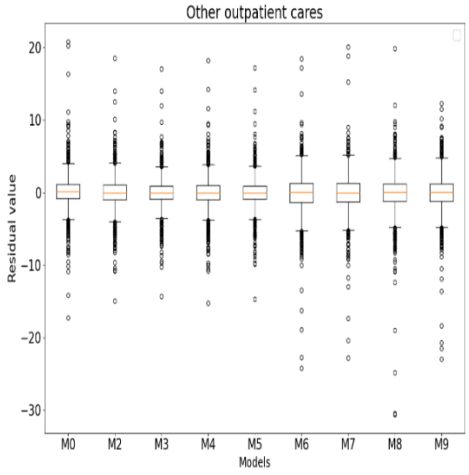
(b) C_2



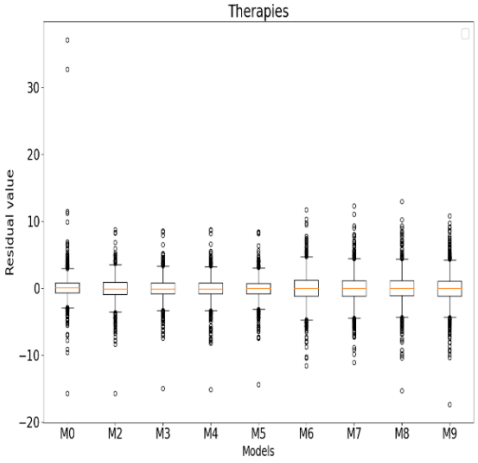
(c) C_3



(d) C_4



(e) C_5



(f) C_6

Figure 2: Distribution of Residual Values for each base learner when the target is CMEV at the training set.

Results for Freq. and CMEV

The initial loss function $\mathcal{L}(\omega)$ was RMSE. We used the Simulated Annealing technique together with the Augmented Lagrangian Equation 11 to obtain the weights ω_i. Figure 3 show the comparison between the true values and the predicted ones by the models given by the Equations 7 and 8 for both Freq and CMEV

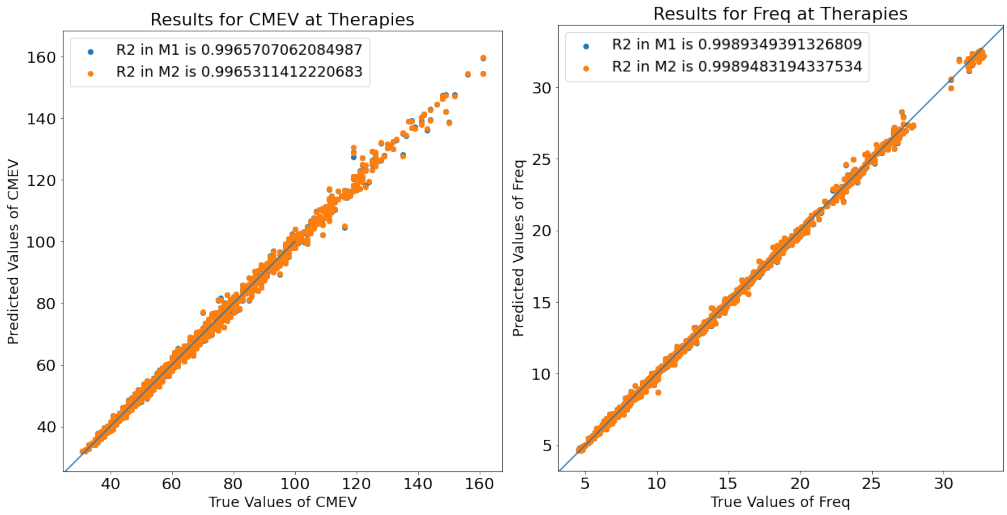


Figure 3: Comparison between true and predicted values for both targets at Therapies coverages

Our goal was to reconstruct both the frequency and the average cost per claims of the Brazilian insurance healthcare actuarial technical reports, from August 2015 to July 2020. To do that, we created coverage-based models for both targets, which are, according to our calculations, approximate to the true unknown functions.

We created our models by training Neural Networks, XGBoost and SVR models. Instead of deciding which model would be the best one, we stacked them to improve the performance of a single model (or the best learner). When overfitting was present, we dropped them from stacking.

By the end, we were left with several optimization problems. In the cases we needed to find out the optimal weights by minimizing the root mean squared error cost function and to consider the constraints on the weights, we modified the original function to be minimized by using the Augmented Lagrangian method. Then, we used the Simulated Annealing method to solve the optimization problems in hand.

Nevertheless, along this work we have assumed two hypotheses: the strong dependence of the probability of survival $p_{x,n}$ when computing the frequency and the second issue was the operator's size. Future work might explore other indexers that we have ignored in this work and the influence of operator's size.

Acknowledgments

The authors are very thankful to the organizers of the Second Congress of Artificial Intelligence at the Pontifical University of São Paulo held in November 2021

for encouraging us to finish this work. We thank to the referees for the comments about the first version of this paper.

I. References

1. Chen T, Guestrin C. Xgboost: A scalable tree boosting system. In Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining. 2016: 785-794.
2. Cortes C, Vapnik V. Support-Vector networks. Machine Learning. 1995; 20 (3): 273– 297.
3. Ganaie M, Hu M, et al. Ensemble deep learning: A review. arXiv preprint arXiv:2104.02395. 2021
4. Wolpert D H. Stacked generalization. Neural Networks. 1992; 5 (2): 241–259.
5. Breiman L. Stacked regressions. Mach. Learn. 1996; 24 (1): 49–64.
6. Abadi M, Agarwal A, Barham P, Brevdo E, Chen Z, Citro C, et al. TensorFlow: Large-scale machine learning on heterogeneous distributed systems. arXiv preprint arXiv:1603.04467. 2016.
7. Harris CR, Millman KJ, van der Walt SJ, Gommers R, Virtanen P, Cournapeau D, Wieser E, Taylor J, Berg S, Smith NJ, Kern R, Picus M, Hoyer S, van Kerkwijk MH, Brett M, Haldane A, Del Río JF, Wiebe M, Peterson P, Gérard-Marchant P, Sheppard K, Reddy T, Weckesser W, Abbasi H, Gohlke C, Oliphant TE. Array programming with NumPy. Nature. 2020 Sep;585(7825):357-362
8. Guo C, Berkhahn F. Entity embeddings of categorical variables. *ArXiv, abs/1604.06737*. 2016.
9. Kingma DP, Ba J. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980. 2014.
10. Hestenes MR. Multiplier and gradient methods. Journal of optimization theory and applications. 1969; 4 (5): 303–320.
11. Powell MJ. A method for nonlinear constraints in minimization problems. Opti mization. 1969: 283–298.
12. Bohachevsky IO, Johnson ME, Stein M L. Generalized simulated annealing for function optimization. Technometrics.1986; 28 (3): 209–217.
13. Romeo F, Sangiovanni-Vincentelli A. A theoretical framework for simulated annealing. Algorithmica. 1991; 6 (1): 302–345.