Technical report
Practitioner challenge

Elisabetta Sanasi, Simone Moawad, Michele Bergami, Anushka Agrawal, Barath Raaj

February 28, 2024

Contents

1 Introduction **2**

2 Data description and pre-processing **2**

3 Resampling techniques **4**

3.1 Random undersampling 4

3.2 Random oversampling 4

3.3 SMOTER 4

3.4 Gaussian noise 4

3.5 WERCS 4

4 Models **5**

4.1 Poisson regression 5

4.2 ZIP regression 5

4.2.1 Limitations 5

4.3 Hurdle model 6

4.3.1 Limitation 6

4.4 Random forest 6

4.5 Rebagg 7

4.6 Deep imbalanced regression 7

4.7 XG-Boost 7

5 Model fitting **7**

6 Results **8**

6.1 First comparison 8

6.2 Overall performance 9

6.3 Performance on the rare class 9

6.4 The comprise 10

7 Conclusion and further improvements **12**

8 References **13**

1 Introduction

In the field of insurance risk management, predicting the frequency and the severity of claims with a high level of accuracy is crucial for ensuring financial stability and efficient resource allocation. Usually, the number of claims is assumed to be distributed as a Poisson and a Poisson Regression model is used to make prediction, whereas the severity is predicted assuming a gamma distribution. In a second moment, multiplying the predictions of frequency and severity enables insurance companies to get an estimate of the claim cost, which is crucial information to fix the final price of the insurance. However, the distribution of the number of claims is heavily zero-inflated, as a result of a substantial portion of policyholders that does not file any claims within a given period (defined exposure). This results in a strongly right-skewed distribution and more sophisticated models are needed to address this peculiar characteristic. Hence, the main purpose of this report is to present, analyze and explore different models to predict the frequency of claims (namely, the number of claims per time of exposure). A first approach will take into account Poisson and Zero Inflated Poisson (ZIP) regressions; then we will move on to hurdles models and to the application of several resampling technique coupled with more sophisticated machine learning techniques, such as random forest, XG-boost and Rebagg.

The main metric of comparison is going to be the mean absolute error (MAE), with particular interest in its value for the normal class (frequency values below 1) and for the rare class (frequency values above 1) separately. Moreover, the F1 score is considered, a metric which combines precision and sensitivity into a single metric, providing a balanced measure that considers both false positives and false negatives. It ranges from 0 to 1, where a score of 1 indicates perfect precision and recall, and a score of 0 indicates poor performance in either precision or recall.

$$\begin{aligned}\text{F1 Score} &= \frac{2}{\frac{1}{\text{Precision}} + \frac{1}{\text{Recall}}} \\ &= \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}\end{aligned}$$

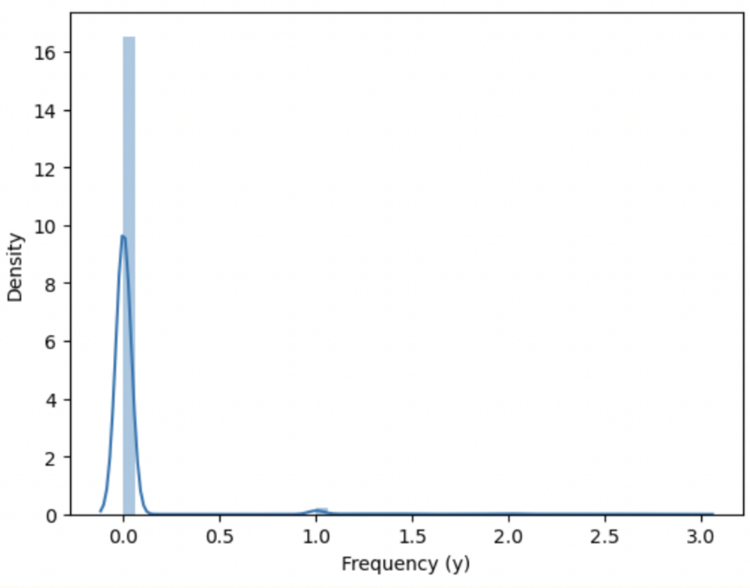
2 Data description and pre-processing

The dataset being used for this analysis contains about 680,000 motor insurance policies data along with some of their features. In particular, the dataset contains 11 columns:

- **IDpol**: the policy ID

- **ClaimNb**: the number of claims in a given exposure.
- **Exposure**: duration of the policy in year units
- **Area**: the area code (A-F)
- **VehPower**: the power of the car
- **VehAge**: the vehicle age, in years
- **DrivAge**: the driver age, in years
- **BonusMalus**: Bonus/malus, between 50 and 350. Values below 100 means bonus and above 100 means malus
- **VehBrand**: the car brand (unknown categories)
- **VehGas**: the car gas, either Diesel or regular
- **Density**: the density of inhabitants (number of inhabitants per km^2) in the city the driver of the car lives in
- **Region**: the policy regions in France (based on a standard French classification)

The graph below shows the distribution of the number of claims. The already mentioned problem of zero-inflation can easily be seen.



Before fitting the models, "VehBrand", "VehPower", "VehGas", "Region", "Area" are one hot encoded and “VehAge” and “DrivAge” are kept as they are, while the rest of the columns are dropped. Moreover, the number of claims are clipped to 4, the exposure to 1 and the amount of claims to 200000 from above to correct for unreasonable observations (that might be data error). We opted also for clipping the frequency to 4 from above to avoid extreme values for the frequency associated with individuals with a very short exposure.

3 Resampling techniques

Resampling is essential when dealing with imbalanced datasets in machine learning. Such datasets have unequal representation of different ranges in the dependent variable, which can mislead models. Resampling techniques, like oversampling or undersampling, help balance the dataset. Metrics such as precision, recall, and F1-score provide a better evaluation of model performance in these scenarios .

3.1 Random undersampling

Undersampling aims to address imbalance by reducing the number of instances from the majority represented part of the distribution.

3.2 Random oversampling

By oversampling, the normal domain is left untouched, while the rare domain is randomly oversampled by duplication.

3.3 SMOTER

In the SMOTER algorithm the normal domain is undersampled, and the rare domain is oversampled. Oversampling is achieved via generating synthetic cases through interpolation of rare values and a random selection of k-nearest neighbors.

3.4 Gaussian noise

With this strategy, the normal domain is undersampled and the rare domain is oversampled by generating synthetic cases. Synthetic cases are generated by adding Gaussian noise to the features (X) and to the target values (y) of rare domain instances.

3.5 WERCS

With this strategy, target values are not split into rare and normal domains. Rather, relevance values are used as weights (probabilities) to select samples for undersampling and oversampling. Instances with high relevance values have higher probabilities of being selected for oversampling and lower probabilities of being selected for undersampling.

4 Models

In this section the Models used are presented. In particular we present the results for each of them, highlighting the main characteristics along with their limitations.

4.1 Poisson regression

Poisson regression is a statistical technique used to model count data, particularly when the outcome variable represents the number of occurrences of an event in a fixed unit of time or space. It is an extension of the more familiar linear regression model, but tailored to handle count data that follow a Poisson distribution. So, while Poisson regression is well-suited for count data, it makes the strong assumption that the mean equals the variance and it will ignore more complicated non-linear relationships that may exist in real world data.

4.2 ZIP regression

A first refinement of Poisson regression is performed using ZIP regression. It extends the traditional Poisson regression model by accounting for both the structural component (modeled as a Poisson process) and the excess zeros component (modeled as a separate process).

4.2.1 Limitations

ZIP models can encounter some challenges when dealing with sparse predictors matrices, as in this case. Here are a few potential issues:

- Computational Complexity: ZIP models involve estimating two separate components, which are the inflation component (typically modeled using logistic regression) and the count component (modeled using Poisson regression). When dealing with a sparse predictors matrix, the computational complexity of fitting these models can increase significantly, especially if the matrix is very large.

- Model Interpretability: Sparse matrices can sometimes lead to overfitting in ZIP models, particularly if there is insufficient data to estimate the model parameters accurately. This can affect the interpretability of the model coefficients and make it challenging to draw meaningful insights from the model.

- Data Sparsity: In ZIP models, the inflation component captures excess zeros in the data, which can arise due to various reasons such as structural features or measurement errors. Sparse predictors matrices may exacerbate this issue, leading to difficulties in accurately modeling the excess zeros and potentially affecting the model's predictive performance.

- Convergence Issues: Sparse predictors matrices can sometimes lead to convergence problems during model estimation, particularly if the predictors are highly correlated or if there are collinearities present in the data. Convergence issues can manifest as warnings or errors during model fitting and may require additional preprocessing or regularization techniques to address.

4.3 Hurdle model

The hurdle model essentially separates the process generating zeros from the process generating positive counts, allowing for a more flexible and realistic modeling approach for datasets with excess zeros.

4.3.1 Limitation

Statistical limitations:

- Assumption violation: The hurdle model assumes independence between the processes generating zeros and those generating positive counts. If this assumption is violated, the model may produce biased estimates.

- Model complexity: The hurdle model introduces additional complexity compared to simpler count data models. This complexity can sometimes lead to difficulties in interpretation and may require more sophisticated statistical techniques for estimation and inference.

- Limited flexibility: While the hurdle model is useful for handling excess zeros, it may not adequately capture the full complexity of the data-generating process in some cases. For example, it may struggle with overdispersion or other patterns that are not well-captured by the two-stage process assumed by the model.

Computational limitations:

- Convergence Issues: Like many statistical models, the hurdle model may encounter convergence issues during estimation, particularly if the model is misspecified or if the optimization algorithm struggles to find the maximum likelihood estimates.

4.4 Random forest

Random forest is a very popular ensemble learning method that can be used both for classification and regression purposes. It basically builds multiple decision trees, each of which is trained on p predictors randomly chosen, and then combines together the predictions to improve accuracy and reduce overfitting. Hence, random forest provides a robust method against overfitting. Nevertheless, it will be more computationally complex compared to simpler linear models.

4.5 Rebagg

The Rebagg is a combination of resampling techniques and bagging. The main feature of bagging is that it aims at improving the accuracy of the model by combining several decision trees trained on different subsets of the training data. Hence, Rebagg is able to combine the benefits of bagging (reducing variance) with decision trees, providing improved stability and accuracy compared to individual decision trees. Nevertheless, exactly as random forest, it may also require more computational resources and time for training compared to individual decision trees.

4.6 Deep imbalanced regression

We explored the Deep Imbalanced Regression (DIR) framework, which addresses imbalanced data with continuous targets. The paper by Yang et al. introduces label distribution smoothing (LDS), a technique that explicitly considers nearby target values and calibrates both label and feature distributions. We incorporated LDS into our neural network architecture, which consisted of three hidden layers, each with 256 ReLU dense neurons. However, despite our efforts, the model predictions did not meet our expectations, leading us to exclude it from our final presentation. Unfortunately, time constraints prevented us from exploring other techniques proposed in the paper.

4.7 XG-Boost

XG-boost stands for extreme gradient boosting. Again, this is an ensemble technique, where several decision trees are combined together sequentially. The peculiarity of the method is that each new tree is trained to correct the errors made by the previous ones. It incorporates L1 and L2 penalty regularization to prevent overfitting and tree pruning is how the model complexity is reduced considerably. XGBoost is highly efficient and scalable, making it suitable for large datasets, but requires careful tuning of hyperparameters, and the optimal parameter settings may not be immediately obvious.

5 Model fitting

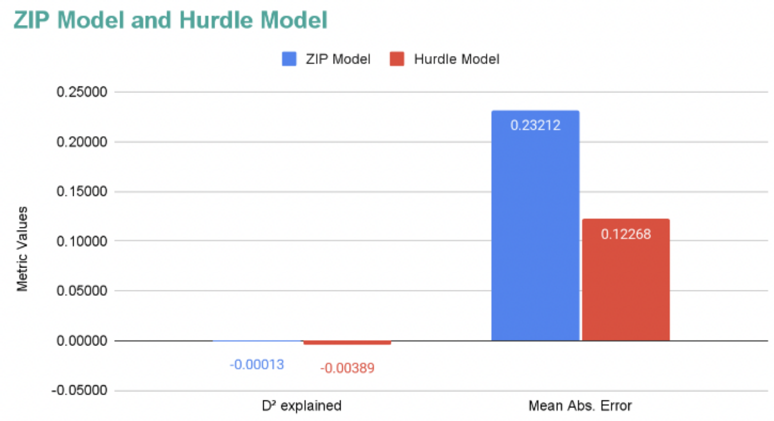
All the models were fitted and the final test performance was measured across five folds of the data. We performed cross validation to improve the performance of the model and make sure the results were not only due to a single random split of the training data. The best combination for hyperparameters for the resampling techniques have been done using a “grid search”. The code file with all the results has been attached with this report.

6 Results

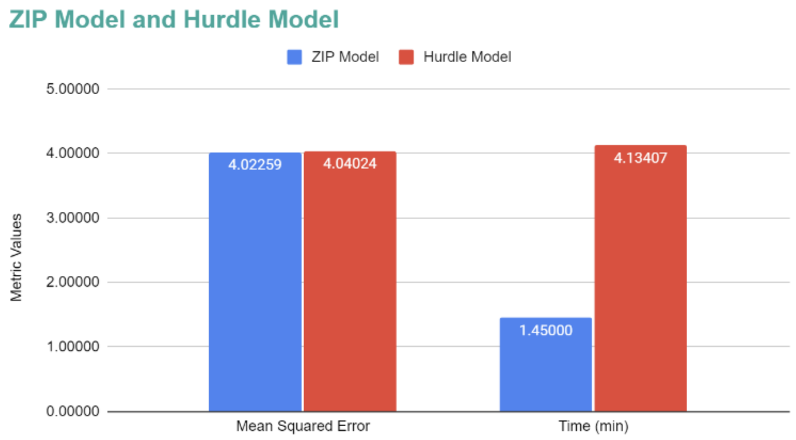
While analyzing the results, we found out that different models could be suggested as the best model according to three main objectives that they may address.

6.1 First comparison

Here we firstly report a first comparison between zero-inflated Poisson (ZIP) regression and Hurdle model with respect to the benchmark.



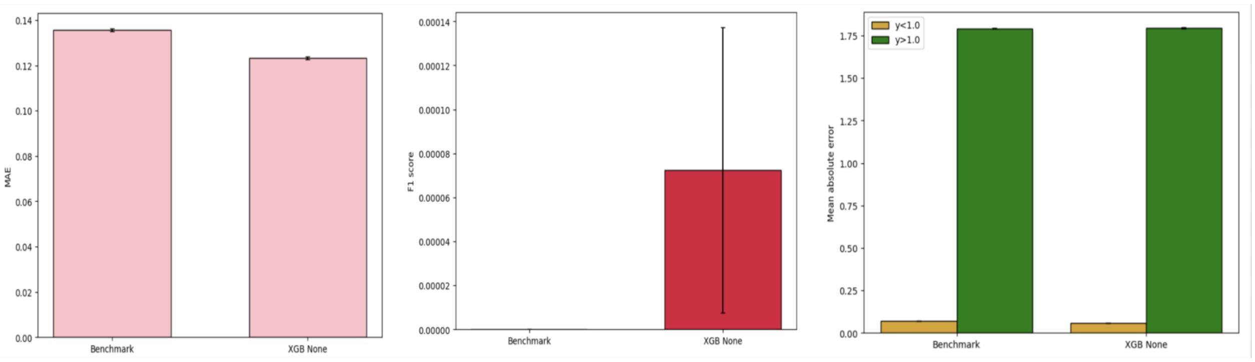
As we can see from the above chart, there is higher MAE for ZIP model as compared with the Hurdle model. Further, we can see the D2 explained is very similar.



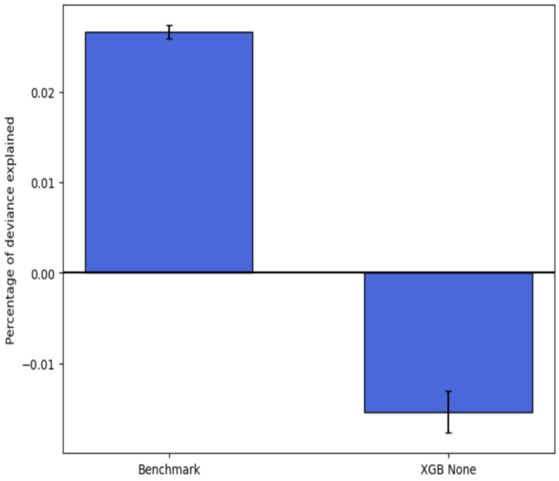
Further, on comparing the MSE and time taken by each of the model, Hurdle model takes more time than ZIP model. The higher time complexity comes with a benefit with lower MAE. Additionally, MSE is very close and comparable.

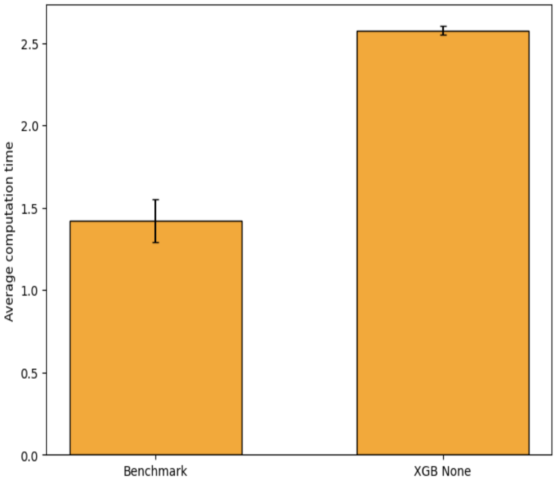
6.2 Overall performance

If interested in a model that overall performs better than the benchmark (the classic Poisson regression), the model to choose is the XG-boost without resampling techniques. As we can see from the plot below, this model has an overall mean absolute error (MAE) which is lower than the MAE of the benchmark and, at the same time, it also does a better job on the rare class with a slightly lower MAE.



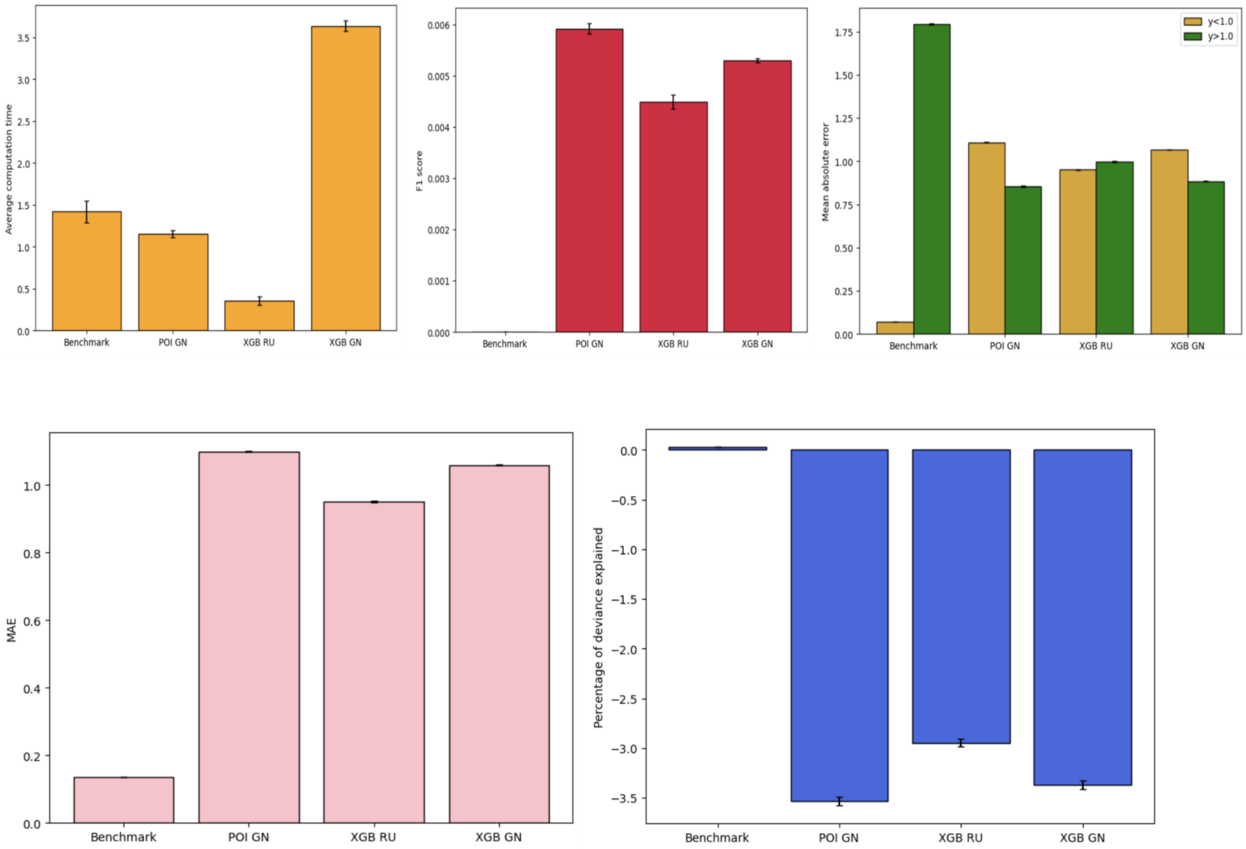
On the other hand, the proportion of deviance explained (PDE) is lower and the average computation time is higher, since it is more complex than the benchmark.





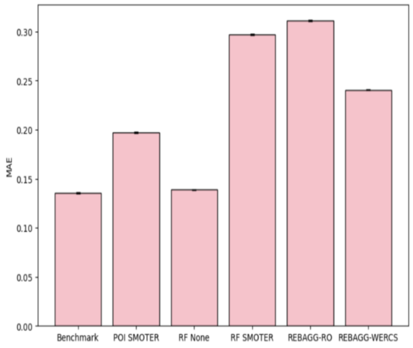
6.3 Performance on the rare class

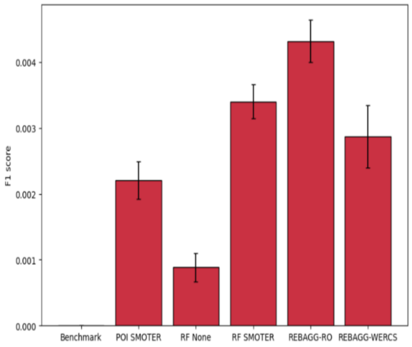
We highlighted how the main problem we tried to address is class imbalances. This is a major problem because the training data will contain few values of the frequency above 1 and, as a consequence, the model will struggle trying to predict them with the test set. So, if interested in finding a model that in particular does a better job in predicting the rare class, we recommend the Poisson regression with gaussian noise (POI GN), the XG-boost with gaussian noise (XGB GN) and the XG-boost with random undersampling (XGB RU) as resampling technique. With the exception of XGB GN, these models have a lower average computational time and introduce a drastic improvement in terms of MAE for the rare class: the values are almost half of the one for the benchmark. Nevertheless, the overall MAE is higher as a result of a higher MAE for the common class.

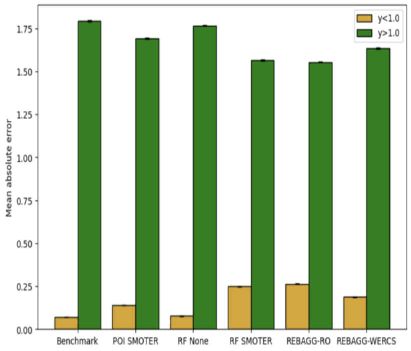


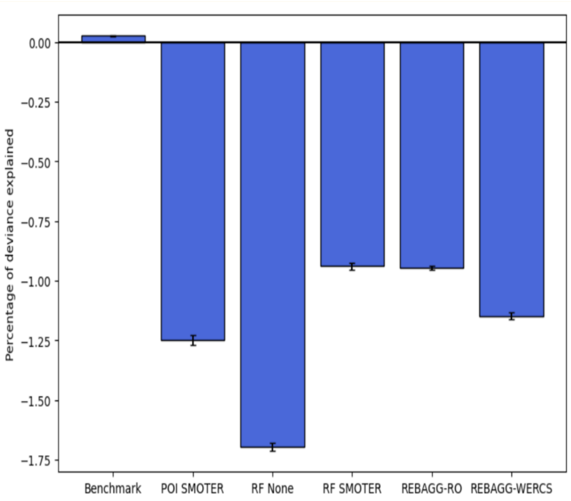
6.4 The comprise

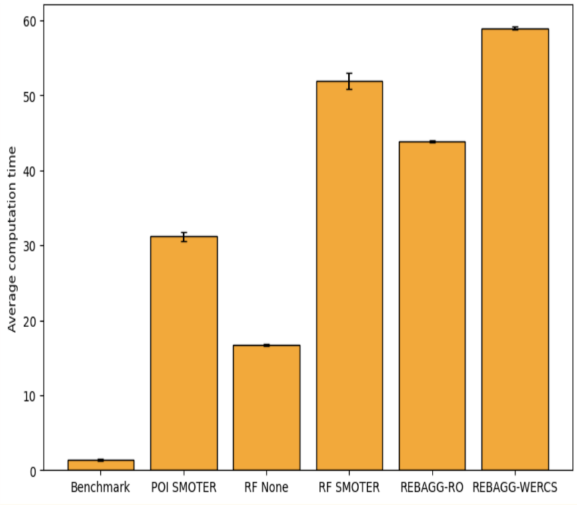
Finally, if interested in a kind of compromise we recommend a set of models: in particular the Poisson regression with SMOTER (POI SMOTER), the random forest without resampling (RF None), the random forest with SMOTER (RF SMOTER), the rebagg with random oversampling (REBAGG RO) and the rebagg with WERCS. We talk about a compromise in this case, because these models manage to improve the MAE for the rare class as before, but, at the same time, remains competitive and comparable with the benchmark in terms of overall MAE. It should come as no surprise that they all have an average computational time higher than the benchmark, since they are much more complex.



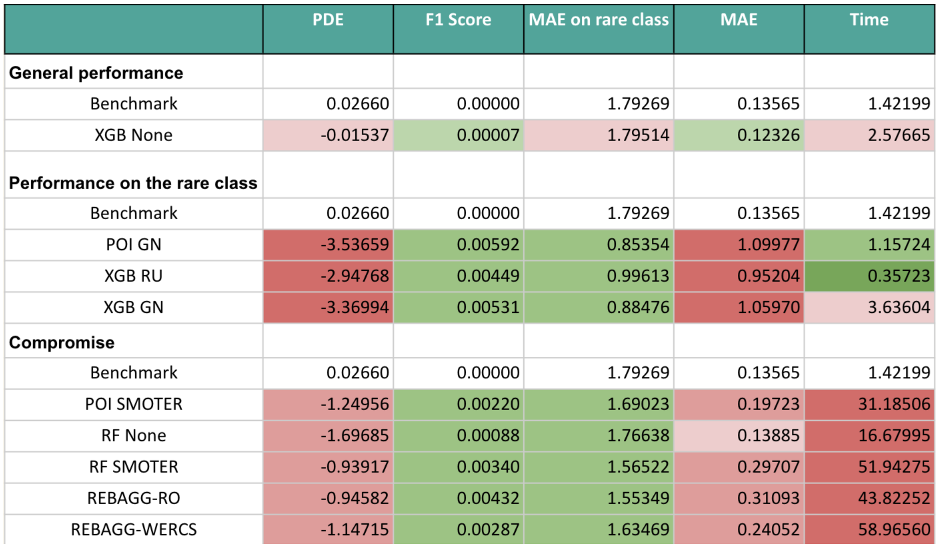








The table below summarizes all the values for the metrics used in the analysis for the proposed models.



	PDE	F1 Score	MAE on rare class	MAE	Time
General performance					
Benchmark	0.02660	0.00000	1.79269	0.13565	1.42199
XGB None	-0.01537	0.00007	1.79514	0.12326	2.57665
Performance on the rare class					
Benchmark	0.02660	0.00000	1.79269	0.13565	1.42199
POI GN	-3.53659	0.00592	0.85354	1.09977	1.15724
XGB RU	-2.94768	0.00449	0.99613	0.95204	0.35723
XGB GN	-3.36994	0.00531	0.88476	1.05970	3.63604
Compromise					
Benchmark	0.02660	0.00000	1.79269	0.13565	1.42199
POI SMOTER	-1.24956	0.00220	1.69023	0.19723	31.18506
RF None	-1.69685	0.00088	1.76638	0.13885	16.67995
RF SMOTER	-0.93917	0.00340	1.56522	0.29707	51.94275
REBAGG-RO	-0.94582	0.00432	1.55349	0.31093	43.82252
REBAGG-WERCS	-1.14715	0.00287	1.63469	0.24052	58.96560

7 Conclusion and further improvements

In this analysis we tried to address the problems of zero-inflation, which, in turns, results in class imbalances, in the context of predicting the frequency of the number of claims. We came up with some models that can be defined as “the best” according to three main criteria: the overall performance, the performance on the rare class and a compromise between the two classes. To further refine the analysis we suggest implementing a simultaneous gridsearch of resampling techniques and model hyperparameters to possibly achieve an optimal model.

8 References

- Yuzhe Yang, Kaiwen Zha, Ying-Cong Chen, Hao Wang, Dina Katabi. "Delving into Deep Imbalanced Regression" (2021). arXiv preprint arXiv:2102.09554.

- https://www.kaggle.com/datasets/floser/french-motor-claims-datasets-fremtpl2freq

- https://github.com/jafetgado/resreg/tree/master