Interpretable Boosted GLM

Karol Gawłowski[1] **Patricia Wang**[2]

Version of 30 August 2025

Abstract

Bridging the gap between models' predictive power and interpretability is one of the key problems in modern predictive analytics specifically in insurance. Despite the availability of more performant Machine Learning (ML) tree-based models, less predictive GLMs are still a go-to method due to their explainable nature. We propose a novel method for ensembling GLMs and GBMs and transform the state-of-the-art interpretability technique – SHAP. The resulting ensemble model, Interpretable Boosted GLM (IBLM), retains the linear formulaic representation and provides a set of per-observation parameter corrections. These corrections help modelers understand how the ensemble deviates from the underlying GLM while improving its performance. The linear architecture of IBLM allows insurers to easily implement it into the existing rating structures, reducing or even eliminating friction costs of its implementation. The SHAP-corrected coefficients enable familiar interpretation of the rates for customers and stakeholders. Most importantly, the transparent nature of IBLM allows insurers better assessment of the risk they are exposed to.

Keywords: Generalized Linear Models, Regression Models, Interaction Effects, Ensemble Methods, Shapley Values, SHapley Additive exPlanations (SHAP), Interpretability, Explainable AI (XAI), XGBoost, Tabular Data

1 Introduction

Gradient Boosting Machines (GBMs), in particular eXtreme Gradient Boost (XGBoost), have consistently outperformed traditional statistical models and neural networks on tabular data.[3][4] Despite the maturity of reliable interpretability techniques such as SHapley Additive exPlanations (SHAP), the adoption of these more performant models in insurance remains slow, with transparency frequently being a key barrier. In insurance, tabular datasets are the

[1] EY UK, karol.gawlowski@bayes.city.ac.uk
[2] Convex, patricia.wang@convexin.com
[3] Why do tree-based models still outperform deep learning on tabular data
[4] XGBoost: A Scalable Tree Boosting System

norm, yet the adoption of advanced Machine Learning (ML) models such as tree ensembles or neural networks remains limited. Despite their success in other domains, neural networks in particular, often fall short in tabular data tasks, where tree-based models like GBMs excel.[1] This performance gap is largely due to the inherent properties of tabular data, which presents challenges—such as irregular patterns and the presence of uninformative features—that tree-based methods are naturally adept at handling.[1] In contrast, neural networks struggle with these complexities, making tree-based models the more reliable choice in many financial applications. On the other hand, Generalized Linear Models (GLMs) have long been a cornerstone for predictive modelling for financial applications due to their simplicity, interpretability, and relatively strong performance. They are widely used because they offer a clear formulaic structure that is easy to explain, aligning well with the regulatory requirements and professional standards in the industry.[5] Furthermore, GLMs are a familiar tool in actuarial science, often featured in professional exams and applied in real-world risk assessment models. However, while GLMs offer great transparency, their predictive power can be limited by their rigid structure. Non-linearity is typically modelled through variable transformation before estimating the coefficients of the linear equation so the scope of capturing non-linearity is not straightforward.

In actuarial data science, there has been a marked increase in interest in the application of machine learning, with commercial adoption often facilitated by software vendors (e.g., Akur8, Quantee, Optalitix) offering either AI-augmented GLM methodologies or fully machine learning–based models aimed at improving current modelling practices. The problem of bridging the performance–interpretability gap between transparent statistical methods and more powerful but opaque machine learning models has received sustained attention from both practitioners and academics.[6,7 8]

In summary, both GLM and ML have several weaknesses and strengths: GLM is easy to construct and interpret and fits well into the existing rating structure that stakeholders are familiar with. However, GLM struggles with non-linear data patterns without prior data transformation, often resulting in lower performance compared to ML models. Neural Networks may offer superior performance compared to GLM, though NNs are not adept at handling tabular datasets and may fall short in comparison to Gradient Boosting Machines

[5] Wuthrich et al AI Tools For Actuaries
[6] Merz et al. Interpreting Deep Learning Models with Marginal Attribution by Conditioning on Quantiles
[7] Meyer et al. SHAP for Actuaries: Explain any Model
[8] FRC - TAS - Models

(GBM).[1] The standard NN is opaque and difficult to interpret but some architectures, such as localGLMNet, are more interpretable.[9] Despite recent developments, all NNs variants suffer a major practical weakness: they require significant effort to configure to realise their full performance potential, and the associated cost and effort of implementing them often negates the performance improvements, rendering this model class less favoured in actuarial and insurance applications. GBM, in particular, XGBoost, has become a popular tool in recent years, due to its relative ease of implementation, and superior performance – especially with tabular data.[1,2] The only potential drawback of the standard GBM is its need for separate explanatory AI methods to attain interpretability.

This paper proposes a novel model Interpretable Boosted Linear Model (IBLM) that retains the strengths of GLM and GBM, while addressing their limitations: IBLM retains the interpretable nature of GLMs but adds flexibility to improve predictive accuracy. We build on the formulaic structure of GLMs by allowing the coefficients to vary locally for each observation, akin to the concept behind LocalGLMnet where output can be interpreted as an individual linear model adapted to different data points.[6] However, unlike the approach in LocalGLMnet, our architecture starts with a set of coefficients from a pre-existing GLM and generates a set of deviations (beta-corrections) that offer a better overall fit, while ensuring the combined model remains grounded in a familiar and linear closed form expression.

To achieve that, we introduce a residual modelling approach where a more flexible, non-linear model predicts the errors of the underlying GLM. This is the principle underlying boosting ensembles, thus we will refer to this residual model as a booster. While residual modelling itself is not new, our key contribution lies in how we utilize SHAP values within this framework. Rather than using SHAP exclusively to explain the residual predictions of the booster, we leverage it to correct the GLM coefficients. This allows us to combine the transparency of GLMs with the superior predictive power of machine learning models, addressing the long-standing trade-off in financial data science between model interpretability and accuracy. Our proposition leverages GBM in place of the residual model.

As we will demonstrate in more detail, our proposition allows for a smooth transition from statistical modelling to building ML models, which is a key consideration in practice. Our method can be used as a standalone predictive model or in a model development cycle, where

[9] LocalGLMnet: interpretable deep learning for tabular data

the residual model informs the modeler about non-linear effects and interactions that may not have been captured in the underlying GLM.

Organization of this manuscript. Section 2 is a formal introduction of the building blocks of the IBLM architecture – GLM, GBM, and SHAP. We then introduce IBLM along with beta-corrections and their interpretations in section 3. In section 4 we demonstrate real-world applications of IBLM and compare it to the underlying GLM to assess the uplift in performance, and to XGBoost as the top performing predictive ML model. We also discuss the practical interpretation of beta-corrections and how these insights can help insurers manage risk in their portfolios. Finally, we conclude in section 5.

2. Constituent models and SHAP introduction

This section is arranged as follows: section 2.1 introduces the starting point of IBLM – GLM. Section 2.2 introduces the basic GBM, and an improved implementation of GBM, i.e. XGBoost. Then section 2.3 introduces SHAP which is the explanatory technique that produces Shapley values to correct the coefficients of IBLM.

2.1 GLM

The GLM was first introduced in 1972 as a class of models to predict a response variable with a distribution from the exponential family including Normal, Binomial, Poisson, and Gamma distribution.[10] GLMs have three core components:

(1) A random response variable y whose distribution belongs to the exponential family, characterized by a canonical parameter θ and a dispersion parameter ϕ with $\alpha(\phi) > 0$ For a fixed ϕ, the distribution is fully specified by θ. The parameter θ is the canonical parameter.

(2) A linear model where the response variable y can be linearly represented by a set of independent variables or features $x_1, \ldots, x_d$ in the form of:

$$\eta = \beta_0 + \sum_{j=1}^{d} \beta_j x_j$$

where $\beta_1, \ldots, \beta_d$ are coefficients to be estimated and β_0 is the intercept.

(3) A link function g that connects the expected value of the response variable in (1) to the linear predictor η in (2) :

[10] Nelder - Generalized Linear Models

$$g(\mu) = \eta$$

When g corresponds to the canonical link of the exponential family distribution, μ is the mean of the random variable y in (1) and the linear predictor η is directly related to the canonical parameter θ. The choice of link function depends on the distribution of y with log-link being the most popular choice.[11] For example:

- Poisson and exponential distributions often use the log link.
- Binomial distribution often uses the logit link.
- Gamma distribution often uses the reciprocal link.

It is worth pointing out the resemblance between the linear equation describing the local accuracy property of SHAP we introduce next and the linear equation describing the second component of GLM. This resemblance allows the coefficients of the Interpretable Boosted Linear Model to be corrected by SHAP. This point will be discussed in section 2.3.

2.2 Gradient Boosting Machine and Extreme Gradient Boosting

Gradient Boosting Machines (GBMs), introduced by Friedman, is a flexible framework for constructing predictive models through stagewise functional optimization.[12] In this setting, we consider a response Y with conditional mean $\mu(x) = E[Y|x]$, modeled via a predictor function $F(x)$. The GBM framework aims to approximate $F(x)$ by iteratively combining base learners, typically regression trees, in order to minimize a specified loss function.

Formally, let $L(y, F(x))$ denote a convex loss function. The gradient boosting algorithm constructs the predictor as an additive model

$$F_M(x) = \beta_0 + \sum_{m=1}^{M} \nu f_m(x)$$

where f_m are base learners from a restricted function class F, $\nu \in (0, 1]$ is a shrinkage parameter (learning rate), and M is the number of boosting iterations. At iteration m, the algorithm fits a base learner f_m to the negative gradient of the loss with respect to the current model:

$$r_{im} = -\frac{\partial}{\partial F} L\left(y_i, F\left(x_i\right)\right)\Big|_{F=F_{m-1}\left(x_i\right)},$$

[11] Wutrich et al - AI Tools for Actuaries
[12] Friedman - Greedy Function Approximation: A Gradient Boosting Machine

for observed data (y_i, x_i) The learner f_m is chosen to approximate these residuals, and the model is updated via

$$F_m(x) = F_{m-1}(x) + \nu f_m(x).$$

When the loss is the squared error, the updates reduce to fitting regression trees to residuals, which gives gradient boosting its interpretation as a sequential residual fitting procedure. For distributions commonly used in actuarial modeling—such as Poisson, Gamma, or Tweedie—the natural choice is to use the corresponding likelihood loss together with the canonical log link. In this setting the mean satisfies $\mu_m(x) = exp\left(F_m(x)\right)$ and the update takes the multiplicative form

$$\mu_m(x) = \mu_{m-1}(x) exp\left(\nu f_m(x)\right)$$

Thus, gradient boosting can be viewed as either an additive ensembling scheme on the canonical scale $F(x)$, or equivalently as a multiplicative ensembling scheme on the mean scale $\mu(x)$. This dual perspective highlights its flexibility in capturing nonlinearities and interactions beyond the scope of generalized linear models and is fundamental for IBLM architecture.

XGBoost is an implementation of GBM with a few additional technical advances to reduce overfitting, to handle missing data efficiently, and to improve computational performance.

One of the key improvements to the GBM is that XGBoost adds a regularisation term to the loss function that penalizes complexity of the model so the new loss function $L'(y, F(x))$ becomes:

$$L'(y, F(x)) = L(y, F(x)) + \gamma T + \frac{1}{2}\lambda \|w\|^2.$$

where T is the number of leaves in a tree; w the leaf weights; γ and λ are user-defined hyperparameters of XGBoost. If γ and λ are both zero, the loss function is effectively the same as the basic GBM.[2]

In addition to the regularisation term, XGBoost uses several other techniques such as shrinkage and column sampling to reduce overfitting. While each technique on its own may be used in other ML algorithms, XGBoost uniquely employs a combination of all three that reduce overfitting more efficiently than the basic GBM. XGboost also improves performance through the optimal split when dealing with missing and scarce data that are often present in

real datasets. Lastly, XGBoost has several computational advantages such as parallel computation that makes the model converge faster. These design and technical advantages allow XGBoost to outperform other machine learning algorithms on many large, high-dimensional datasets with missing values, so we chose XGBoost as a constituent model of IBLM. That said, we want to emphasize that XGBoost could be easily substituted with another booster model, such as the basic GBM or even NN, if deemed more suitable.

2.3 SHAP

SHAP stands for SHapley Additive exPlanations. It is based on cooperative game theory that guarantees a unique attribution of contributions of each input variable. The SHAP technique has become one of the most popular explanatory methods since its introduction in 2017.[13] These contributions can be summed to produce an interpretable decomposition of the original prediction. For a simple linear model such as GLM, the outputs are easy to interpret, so the best explanation is the model itself. For complex non-linear models such as XGBoost or Neural Networks, the outputs are not easy to interpret, so a simpler explanation framework such as SHAP is needed.

SHAP is the adaptation of Shapley values. In a regression context, the Shapley value of feature x_j is the weighted average of the difference between conditional expectation on subset of features including x_j and conditional expectations on all subsets excluding x_j. It is calculated as follows: let S represent a feature subset $S \subseteq F$ where F is the set of all features. A model $f_{S\cup\{j\}}$ is trained with feature x_j present, and model f_S is trained with the feature x_j withheld, i.e. a subset S excluding x_j. Then, the difference between predictions from model $f_{S\cup\{j\}}$ and f_S is computed. Since the effect of feature x_j depends on other features in subset S, we need to compute all the differences between $f_{S\cup\{j\}}$ and f_S conditioning on all the subsets excluding x_j, that is $S \subseteq F \setminus \{j\}$. Shapley value of feature x_j denoted as ϕ_j is then the weighted average of all possible differences. [10] Formally, Shapley value of a covariate x_j is computed as:

$$\phi_j = \sum_{S\subseteq F\{j\}} \frac{|S|!(|F|-|S|-1)!}{|F|!} \left[f_{S\cup\{j\}}\left(x_{S\cup\{j\}}\right) - f_S(x_S)\right]$$

[13] A Unified Approach to Interpreting Model Predictions

where F is the set of all features and S represents a subset $S \subseteq F$ and f_S is a model f trained only on the subset S.

Retraining f_S for all possible permutations of S is computationally inefficient, and SHAP is a way to approximate ϕ_j without loss of its desirable properties. In practice, SHAP decomposes an output of any model into the contributions of its input features, where their directionality and magnitude indicate their impact on the overall prediction.

$$g(x) = \phi_0 + \sum_{j=1}^{d} \phi_j(x)$$

Where $g(x)$ is the models' prediction at the link function level and $\phi_j(x)$ is the contribution of feature x_j.

Now we list the properties of SHAP that make it a desirable and reliable model explanation method:

1. **Local Accuracy:** the decomposition is exact, i.e. the prediction equals the sum of all contributions plus the baseline.
2. **Symmetry:** if two features contribute identically across all coalitions, they receive equal attribution.
3. **Dummy (Nullity):** a feature with no effect on the prediction has zero contribution.
4. **Additivity**: attributions are consistent across models, allowing linearity of explanations.
5. **Model-agnostic**: SHAP values can be computed for any predictive model by evaluating the contributions of each feature across all of their possible subsets. In practice, exact computation is combinatorially expensive, but specialized algorithms exploit the model structure to improve efficiency. For instance, TreeSHAP leverages the hierarchical structure of decision trees to compute SHAP values in polynomial time, making it particularly suitable for tree-based ensembles such as GBMs. This allows the additive decomposition to be efficiently obtained even for complex models while retaining the interpretability guarantees of the SHAP framework.

It has been demonstrated that SHAP is more accurate and consistent than other explanatory techniques.[14] As discussed in section 3, under the Interpretable Boosted Linear Model, SHAP values are used to correct coefficients of GLM, so these desirable properties are especially important.

3. IBLM

3.1 IBLM definition

The main aim of the Interpretable Boosted Linear Model is to achieve superior performance compared to a pure GLM, whilst maintaining a linear architecture, making the coefficients of the ensemble model easy to interpret and explain.

In this paper, we propose the Interpretable Boosted Linear Model (IBLM), which combines a GLM with a tree-based booster such as XGB, leveraging the superior performance of tree ensembles on structured data. The GLM serves as the primary predictor, and the booster is trained to model its residuals using the same input features. The ensemble output is obtained by aggregating the predictions of the GLM and the residual model, either additively or multiplicatively, depending on the response variable. SHAP values from the residual model are transformed into adjustments to the GLM coefficients, referred to as beta-corrections. When the residual model is predictive, this results in an ensemble that is both highly performant and interpretable within a familiar linear-model framework.

Lemma (SHAP decomposition as a linear predictor)

Let y_B *be the residual predicted by the booster model, and let*

$$y_B = \varphi_0(x) + \sum_{j=1}^{d} \varphi_j(x)$$

be its SHAP decomposition, where $\varphi_j(x)$ *is the SHAP contribution of feature* x_j*.*

This can be approximated in a locally linear form as

$$y_B \approx \varphi_0(x) + \sum_{j=1}^{d} \frac{\varphi_j(x)}{x_j+\delta} x_j = \varphi_0(x) + \sum_{j=1}^{d} \alpha_j(x) x_j$$

[14] Monotonic solutions of cooperative games

Where $\delta > 0$ *guarantees numerical stability if* $|x_j| < \varepsilon$. *This representation highlights that SHAP can be interpreted as a locally linear model, with feature-specific coefficients* $\alpha_j(x)$ *that vary with the feature vector* x.

Definition (Interpretable Boosted Linear Model)

Let $x = (x_1, ..., x_d)$ *be the feature vector, g the link function,* $\mu = E[Y|x]$, *and let the GLM and booster components* $\mu_G(x)$ *and* $\mu_B(x)$, *respectively. Then the IBLM models the canonical-scale predictor as:*

$$g(\mu(x)) = g(\mu_G(x)\mu_B(x)) = \underbrace{\lambda(x)}_{baseline} + \sum_{j=1}^{d} \underbrace{(\beta_j + \alpha_j(x))}_{adjusted\ coef.} x_j$$

Which can be written as:

$$g(\mu(x)) = \lambda(x) + \sum_{j=1}^{d} \alpha_j'(x) x_j$$

where:

- $\lambda(x) = \beta_0 + \phi_0(x)$ *combines the GLM intercept* β_0 *with the baseline adjustment* $\phi_0(x)$ *from the booster,*
- β_j *are the original GLM coefficients,*
- $\alpha_j(x)$ *are feature-wise corrections learned by the residual model.*
- $\alpha_j'(x) = \beta_j + \alpha_j(x)$ *is a final corrected beta for a given observation*

This formulation preserves the additive structure of the GLM on the canonical scale, while allowing flexible, nonlinear adjustments through $\alpha_j(x)$. For links such as the log, this also corresponds to multiplicative adjustments on the mean scale, giving a unified and interpretable representation of the combined model.

The IBLM architecture and crucially $\alpha_j(x)$ can be obtained using any state-of-the-art predictive model, given φ_j estimation techniques that are model agnostic like kernel-SHAP.[15] We comment on the practical aspects and interpretation in the section that follows.

3.2 Beta-corrections interpretation and practical consideration

In many ways the interpretation of beta-corrections will be similar to that of regression attentions of LocalGLMNet. The main difference here is that $\alpha_j(x)$ are anchored in β_j thus they can give feedback as to the fit quality of the underlying model and inform of signal not captured therein. Thus, IBLM can be used either as an interpretable architecture enhancing a pre-existing GLM, or in a modelling cycle where the booster informs about areas of underperformance of the base model. As the GLM iterations improve, the interpretable component explains more variance, leaving only complex patterns for the machine learning booster to capture while transparency is maintained.

We list properties of beta-corrections below:

(1) Goodness of fit: if $\alpha_j(x) < \varepsilon$ or similarly $SE\left(\beta_j\right) > \alpha_j(x)$ indicates that the underlying GLM captures the signal from x_j sufficiently well and that there are no strong interactions effects involving x_j.

(2) Nonlinearity: $\frac{\partial \alpha_j(x)}{\partial x_j} \approx 0$ over the range of x_j, indicates a linear relationship of the target variable with x_j,. Otherwise, the non-linear signal has been captured by the booster.

(3) Interaction effects: $\frac{\partial \alpha_j(x)}{\partial x_k} \approx 0$ for $k \neq j$ indicates lack of interaction effects or otherwise, they are captured by the booster.

(4) Treatment of the intercept: When any categorical or binary variables are not used, term $\lambda(x)$ can be dropped and replaced with a constant λ as a sum of β_0 and a constant φ_0 SHAP bias. Otherwise, $\lambda_i(x)$ for an observation x_i is the sum of its $\alpha_{i,j}(x)$ for reference levels of categorical/binary predictors.

[15] Molnar (2019)

Points 2 and 3 are specifically interesting when utilizing libraries like XGBoost which let the modeler control interactions and introduce monotonicity constraints. We will expand on that in the next section.

The proposed architecture utilizes XGBoost implementation of the GBM as the booster for the underlying GLM, however this need not be the case. We emphasize that the same formulaic representation can be obtained using any model to predict the residuals, as long as it is of satisfactory predictive performance and utilizes the same predictors. The beta-corrections can be obtained in the same fashion. However, when it comes to estimating SHAP, tree-based models are preferable because the computation time required is much less than for e.g. neural networks. Which combined with tree ensembles outperforming NN in tabular tasks, make the XGBoost an optimal choice for IBLM. What is more, XGBoost implementation offers additional features useful in practice like monotonicity constraints, efficient and scalable compute as well as easily retrievable SHAP values. These design features further make IBLM applicable in practice and their impact can be viewed through beta-corrections. We also note that standard SHAP-based variable importance can still be used if need be.

Extending the approach to alternative residual models, including neural-network–based variants, lies beyond the scope of the present study but offers potential for further investigation. A promising direction for future research is to place LocalGLMnet in place of the booster and interpret its learned regression attentions directly as $\alpha_j(x)$ in the IBLM framework.

4 Real Data Example

4.1 Data overview

We consider the application of our model architecture to the FreMTPL2freq dataset, available through the CASdatasets R package.[16] This dataset is widely used in actuarial data science literature, and we provide only a brief overview here. For a detailed analysis, see the case study by Noll et al (2020).

The dataset consists of $n = 678'013$ observations and includes 12 variables, 11 of which are predictors. Our observations are (Y_i, v_i, x_i) where we have the claims number $Y_i \in N_0$,

[16] Dutang–Charpentier

exposure $v_i \in (0, 2.01]$ and predictors x_j outlined in listing 4.1.1. The total number of reported claims is $\sum_{i=1}^{n} y_i = 36'012$ with a combined exposure of $\sum_{i=1}^{n} v_i = 358'499.4$ yielding an overall claim frequency of 10%.

Listing 4.1.1

- 5 numerical features: Bonus-Malus Level, Density, Driver's Age, Vehicle Age, Vehicle Power;
- 1 binary feature: Vehicle Gas;
- 3 categorical features: Area Code, Vehicle Brand and Region

4.2 Model Performance benchmarks

The modelling task is to estimate the expected claim frequency. Model performance is assessed using Poisson deviance D and to compare the performance uplift against a homogenous model we use the Pinball Score R^*.

$$R^* = 1 - \frac{D_m}{D_0}$$

Where D_m indicates Poisson deviance of a predictive model of interest and D_0 Poisson Deviance of a null model. We split the data into Train, Validation and Test sets consisting of 70%, 15% and 15% of the total number of records respectively. Train set L consists of $n_L = 474'609$ records and both test T and validation V of $n_T = n_V = 101'701$

We perform minimal data processing and feature engineering and will demonstrate how the latter can be achieved further in this section, utilizing outputs of IBLM. We remark that categorical variables for the GLM were dummy encoded and their treatment for XGB is through one-hot encoding.

Firstly, we compare the performance of our proposed architecture against two widely used interpretable and high-performing models – GLM and XGBoost. L is used to learn the parameters of the GLM, XGBoost, and IBLM models. The validation set is employed for early stopping and to prevent overfitting in the boosted models, ensuring robust out-of-sample generalization.

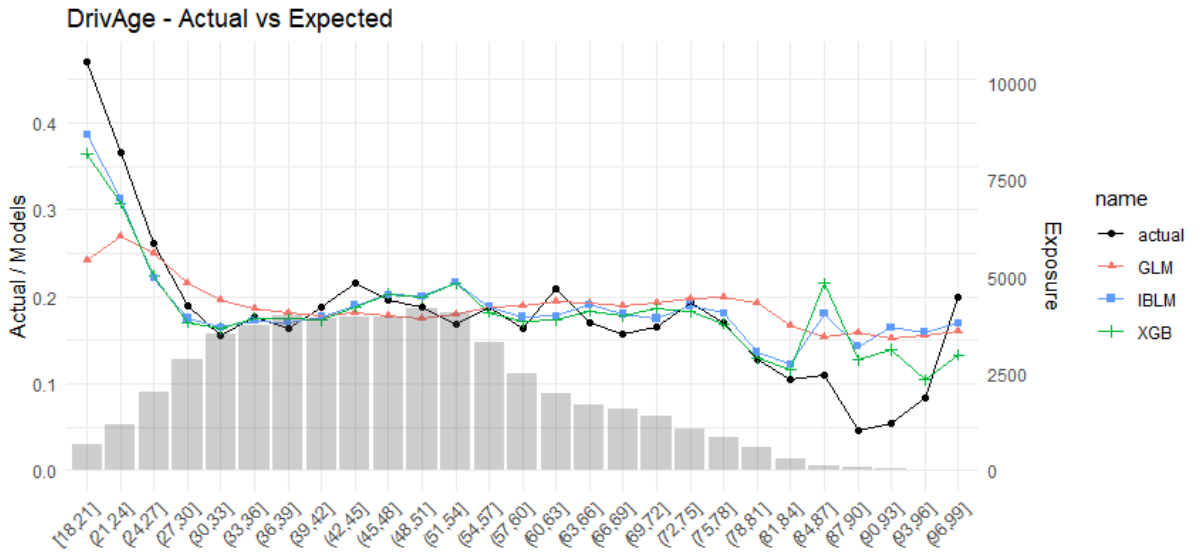
For the GLM, we use the canonical log link. Both XGBoost and the tree-based booster component within IBLM are trained using default hyperparameters, optimization of which is beyond the scope of this study. We train the models for up to 300 rounds with early stopping set to 25. In practice, we find that the default XGBoost parameters already yield strong performance, which is appealing for practitioners who may be less experienced with hyperparameter tuning; while more advanced users could pursue further gains through optimization algorithms. The key hyperparameter values used are $learn\,rate = 0.3$; $max\_depth = 6$.

Table 4.2,1 presents the results of the model benchmarking exercise. We observe that while XGBoost achieves the highest predictive accuracy, IBLM follows closely. Both models outperform the baseline GLM by a factor of approximately three in terms of improvement over a homogeneous model, as measured by the Pinball Score.

Table 4.2.1

	null	**GLM**	**XGBoost**	**IBLM**
Poisson Deviance	1.4195	1.3606	1.2386	1.2475
Pinball Score	0.00%	4.15%	12.74%	12.12%

Figure 4.2.2



We also look at an example actual vs expected chart based on T for DrivAge in Figure 4.2.2. Unsurprisingly, we see how on average IBLM is closer to XGBoost while diverging from the GLM, in most of the cases fitting better to the actual observations.

The ages were only binned for better legibility. We note that for ages 84 and above, there is very little exposure thus none of the models is very performant. It is worth observing that in those low exposure segments, IBLMs predictions remained much closer to its baseline GLM as compared to the XGB.

From an implementation standpoint if predictions vastly different from the underlying GLM are undesirable a post hoc capping can be applied. Details of this approach are covered in Appendix 1.

4.3 Interpretability per observation

We show how to investigate individual instances of IBLM. This view would mostly be useful from an underwriters' standpoint, or for explainability for insureds requirements.

Let's look at an observation from T where $y_{GLM} = 0.2942$ and $y_{IBLM} = 0.1114$ yielding a correction from the booster of $y_B = 0.3784$. For this instance, the correction from the GLM is in the right direction as the true target value $y_i = 0$.

In absolute terms the biggest beta-corrections are for VehAge and VehBrand. However, in relation to the β_j the greatest impact is obtained for VehPower. In most cases the booster proposes a decrease in the GLM parameters whereas an increase is suggested only for Area and VehPower, which results in an overall decrease of the predicted frequency.

Table 4.3.1

	Predictors	**GLM coef.**	**Beta-corr.**	**Lin. Pred.**	**Lin. Pred.**	**IBLM**
	x	β	α	$x(\beta + \alpha)$	$\sum x(\beta + \alpha)$	$exp\left(\sum x(\beta + \alpha)\right)$
Intercept	1	-2.6174	-0.0053	-2.6228	-2.195	0.111
Region	R73	-0.1669	-0.0285	-0.1955		
VehAge	2	-0.0601	-0.2940	-0.7082		
VehPower	10	-0.0003	0.0072	0.0691		
Density	3301	0.0000	0.0000	0.0146		
DrivAge	61	0.0008	-0.0017	-0.0593		
Area	E	0.0132	0.0003	0.0135		
BonusMalus	50	0.0156	-0.0007	0.7434		
VehGas	Regular	0.3337	-0.1244	0.2093		
VehBrand	B12	0.5318	-0.1909	0.3410		

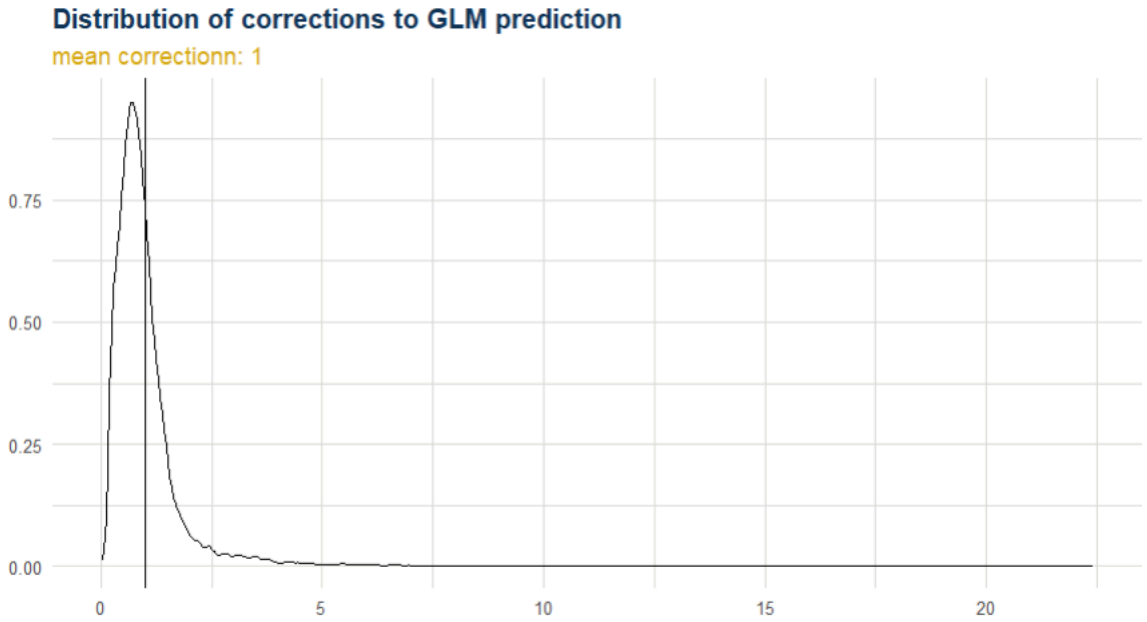
4.4 Global Interpretability

Given the structure of the IBLM, the flow of information from the booster can be examined in multiple ways, which we outline in this section based on T, unseen during training.

Figure 4.1.1. presents the distribution of overall corrections stemming from the booster, which have a mean close to 1. This aligns with the balancing property of the GLM component—on average, no systematic correction is required. This property provides a safeguard against booster bias that could otherwise degrade the IBLM fit. A pronounced long tail in this distribution may be undesirable in practice; the model can be fitted with this in mind by applying a transformation to the residuals, as described in Appendix 1. Alternatively, focusing on instances with the largest corrections can be instructive, as it highlights where the GLM exhibits its largest errors relative to the highly performant XGBoost. Such analysis can act as an additional validation step and help identify model points warranting further scrutiny,

for example to determine whether they should be included in training or excluded due to being genuine outliers or originating from sparse regions of the predictor space.

Figure 4.4.1.



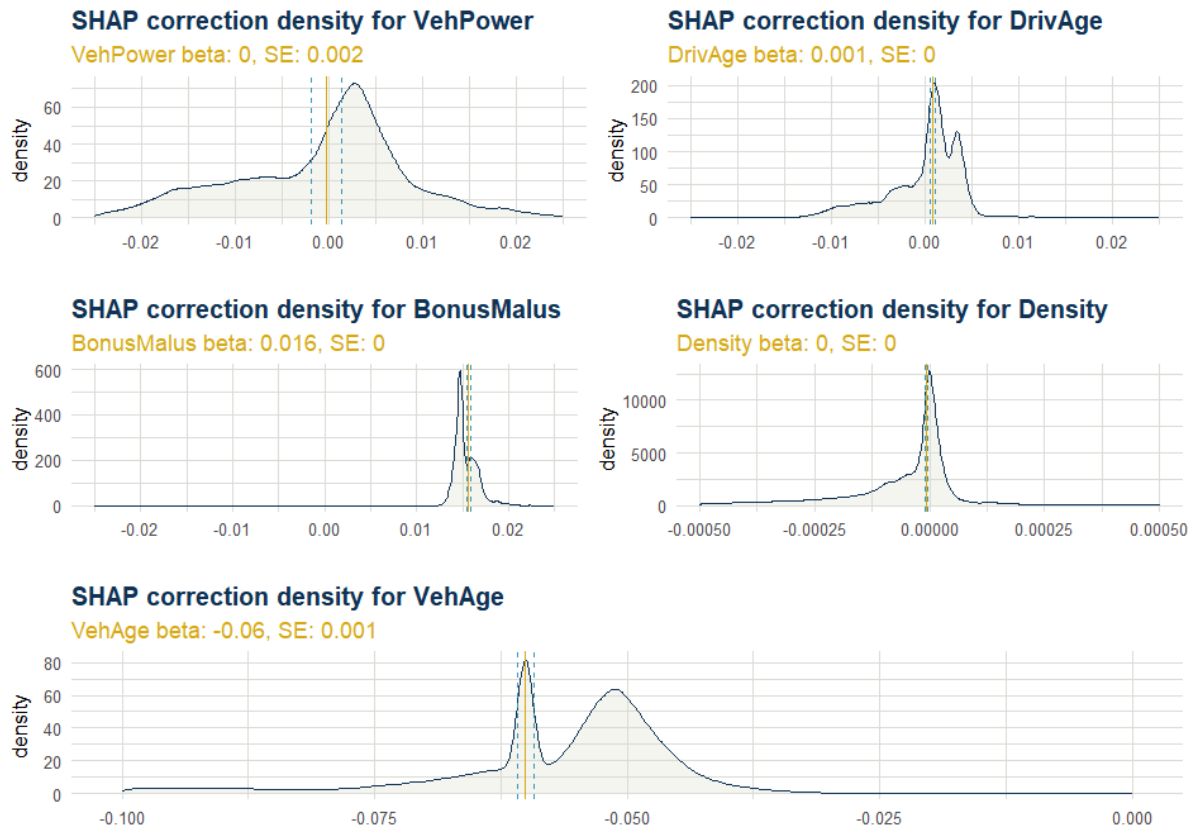
Let's look at the distributions of the beta-corrections to better understand which variables still had signal left to be utilized by the booster after being processed by the GLM. We will investigate the location and spread of these distributions and how it compares with the β_j as well as their standard errors.

We start by assessing the distributions for continuous predictors. These can reveal the extent of corrections made by the booster, their directionality and potential for identified interaction effects. We observe that $\alpha'_{VehAge}(x)$ is bimodal and one of the modes lays on the β_{VehAge} with a relatively tight spread, only slightly falling outside the standard error of β_{VehAge}. The second mode prompts us to investigate this predictor further for non-linear relationships with the target variable or potential interaction effects with other predictors. Density of $\alpha'_{VehPower}(x)$ is not centred around $\beta_{VehPower}$ and importantly, it is spread widely outside the standard error interval, suggesting a reversal in final IBLM coefficients for many observations. The dataset we are analysing is well studied and it has been shown that treatment of this variable as ordinal is preferable, however without this prior knowledge,

$\alpha'_{VehPower}(x)$ would give the modeler the feedback needed for feature engineering in this case.[17] In section 4.5 we will look more closely at the DrivAge variable. From its beta-correction distribution in 4.4.2 we can see an asymmetrical signal found by the booster as well as bimodality and we will show what explains these phenomena.

In practice, these insights from IBLM can aid a modelling actuary in building a better underlying GLM. A portfolio manager will also benefit from these insights, as the shape of the beta-correction distributions informs about the potential of profitability, low-risk segments or anti-selection. For instance, VehAge bimodality suggests that there are two subgroups that differ in the impact this variable has on the risk. The mode not aligned with β_{VehAge} should see an increase in the estimated frequency - not captured by the base GLM. A similar effect is seen for DrivAge.

Figure 4.4.2.



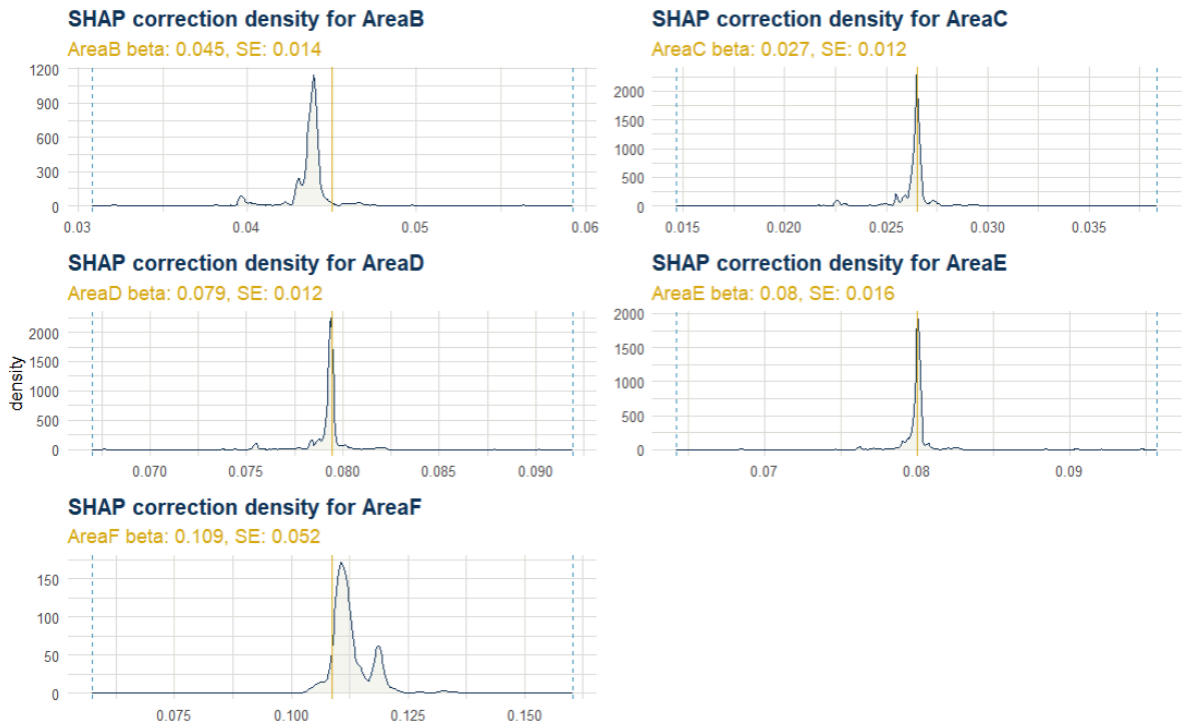
The location of these distributions may also reveal interaction effects. It could serve as a prompt to study a given predictor more closely and perform feature engineering leading to a

[17] Noll et al *Case Study: French Motor Third-Party Liability Claims*. March 4, 2020.

better fitting GLM e.g. through splitting continuous predictors into buckets or grouping categories to higher order classes.

We will now analyse two cases of categorical variables, starting with Area. The distributions of $\alpha'_{Area}(x)$ for each of the levels of this variable suggest that there is no more signal left for the booster to use after this the data has been processed by the GLM. We note that because the densities lay within the standard error bands of GLM parameters and have minimal dispersion. We notice that the modes of the corrections sometimes don't fall exactly on the β_j. Given their magnitude and the width of standard error interval, this should not have a material impact on the final model predictions. The modeler can decide whether to include these $\alpha_j(x)$ in the final IBLM for improved interpretability. We refrain from interpreting the bimodality of the beta corrections for AreaF, as this distribution is contained within the standard error bands as well.

Figure 4.4.3.

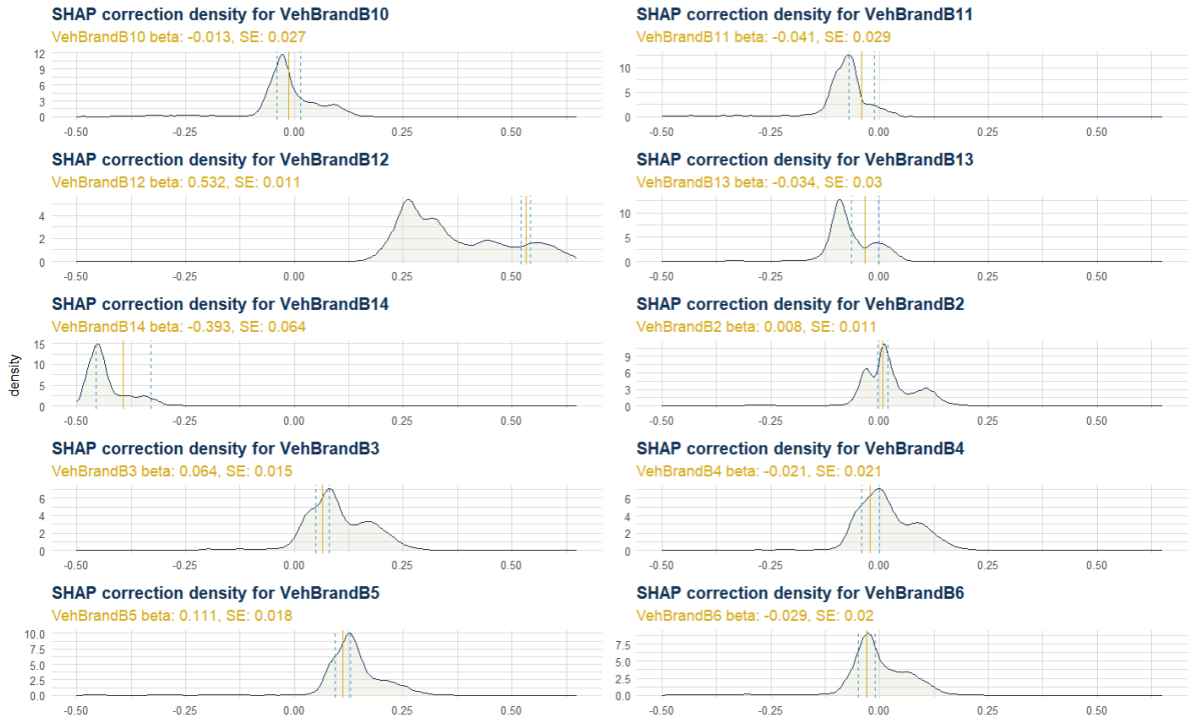


Focusing on the VehBrand variable next, we can see how the booster finds a predictive signal not captured by the GLM. This is indicated by beta-correction distributions falling outside the standard error bands. Interestingly, the modes of some of them are not aligned with β_j. For example, VehBrandB12 sees the greatest correction as compared to other levels. We also note

that only $\beta_{VehBrandB12}$, $\beta_{VehBrandB14}$, $\beta_{VehBrandB3}$, $\beta_{VehBrandB5}$ estimates were significant and we can see that reflected in the corresponding beta-corrections as well, as they are centred around zero.

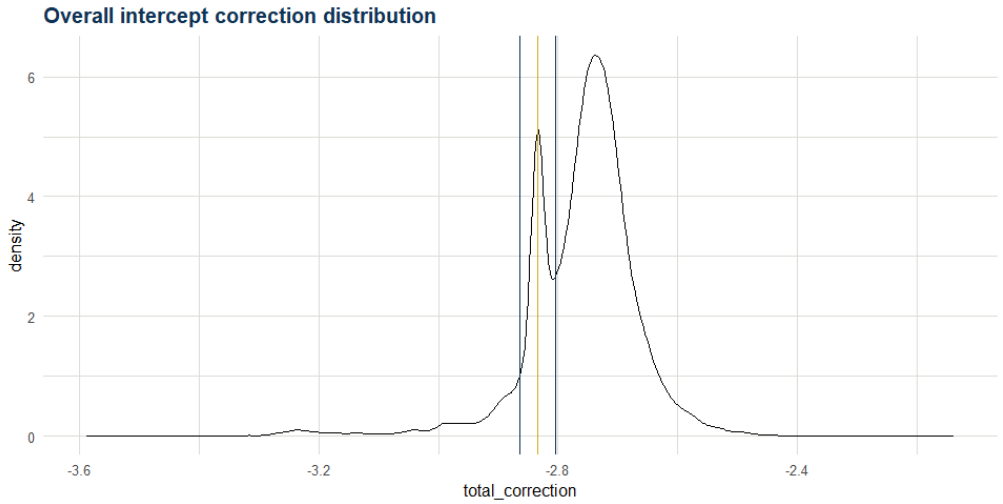
If the IBLM was fitted with the underlying GLM fixed and deployed into production, this set of distributions gives valuable feedback to the risk of under or overpricing. This is especially notable for VehBrandB12 or B13 where the $\alpha'_{VehBrand}(x)$ mode is below the GLM estimate.

Figure 4.4.4.



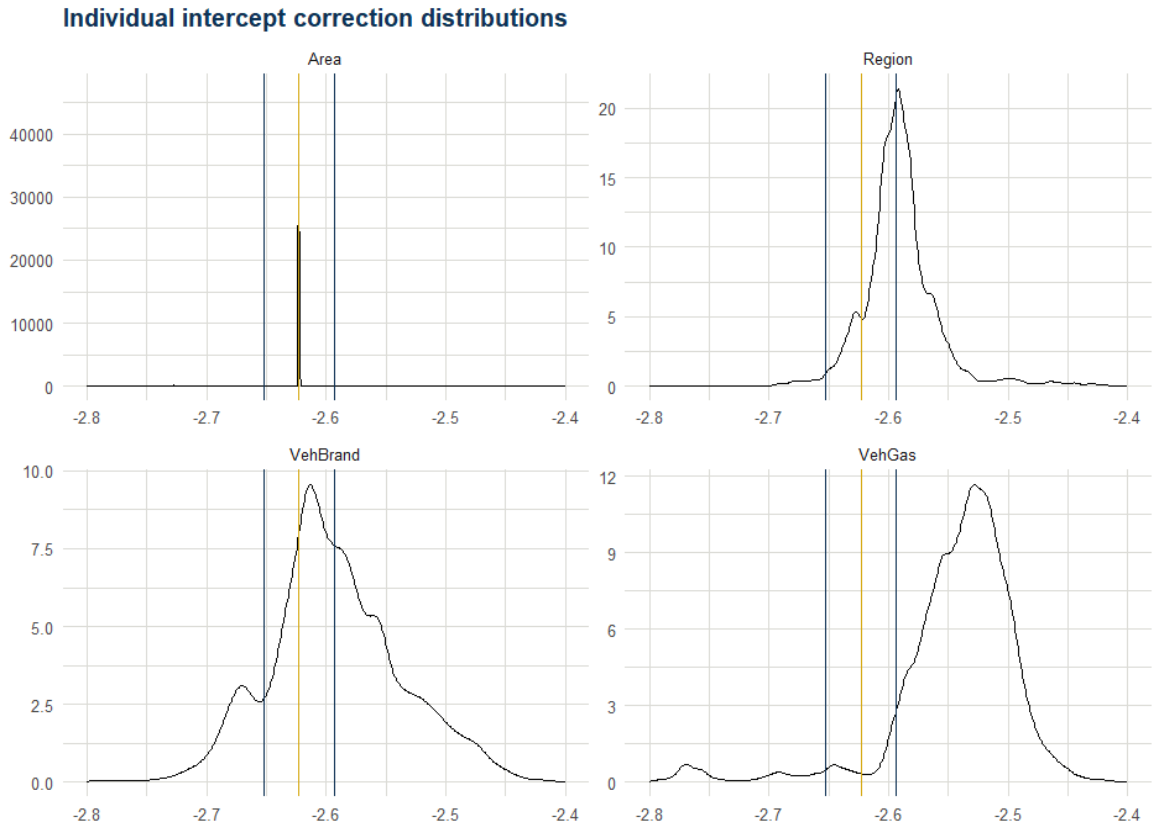
We now analyse the obtained $\lambda(x)$. Figure 4.4.5. shows the distribution of $\lambda(x)$, which we note is bimodal thus prompting us to analyse the individual $\lambda_j(x)$ components for reference levels of categorical variables in 4.4.6.

Figure 4.4.5.



We observe that $\alpha_{AreaA}(x)$ has practically no impact on the location of the distribution which is in line with what we have observed earlier for the other levels of this covariate. The greatest shift is noticed in the $\alpha_{VehGas}(x)$ as it falls outside $SE(\beta_0)$. Mode of $\alpha_{VevBrand}(x)$ is the closest to the original intercept but its spread is also wider than the standard error bands.

Figure 4.4.6.



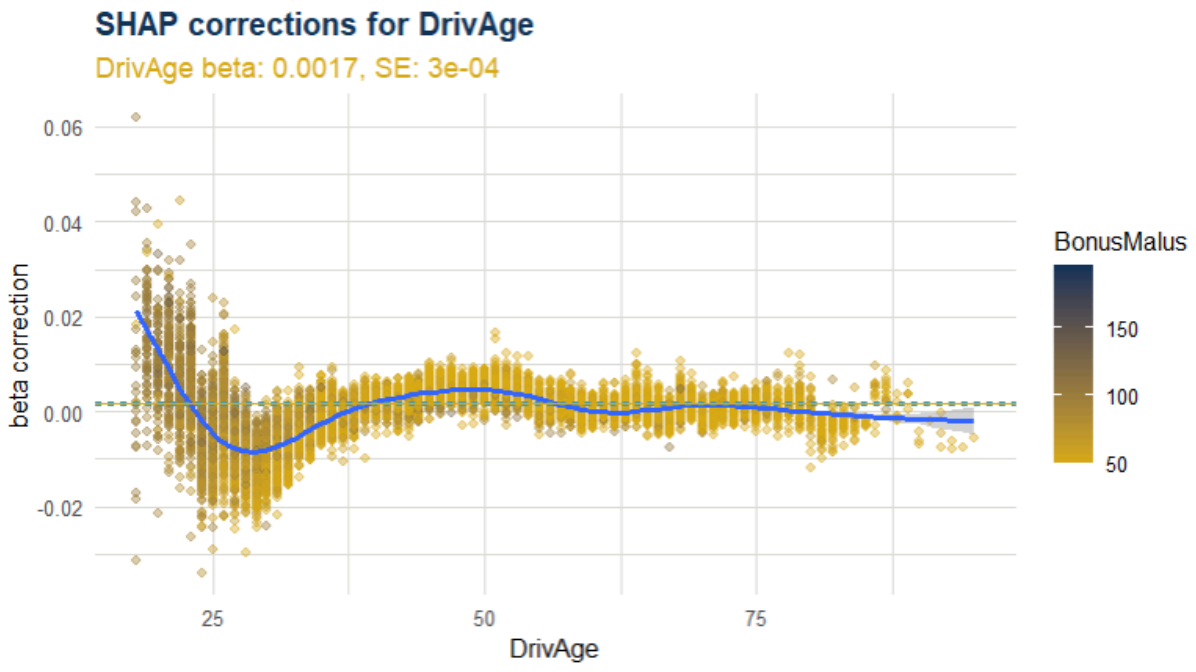
4.5 Interaction effects

Now we use the booster as a source of information on how to improve the underlying GLM and understand the risk portfolio. Here we will look for beta-corrections to reveal potential feature engineering opportunities and interaction effects between predictors.

We plot the individual $\alpha'_{i,j}(x)$ against $x_{i,j}$ to inspect dependence between the two, which would reveal non-linear patterns uncaught by the GLM. We plot average $\alpha_j(x)$ for each value in the domain of $x_{i,j}$. This is indicated by the blue line. The scatter can then be coloured by the level of another predictor to identify interaction effects.

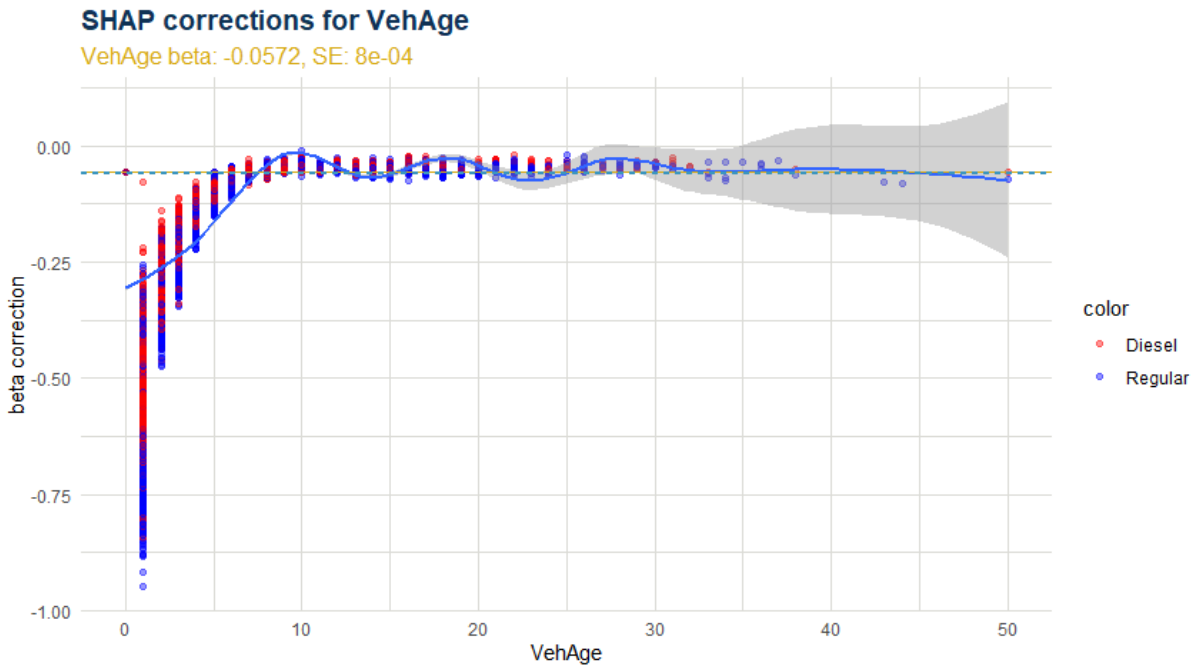
Looking at DrivAge, the change in directionality and dispersion of beta corrections suggests the underlying GLM does not capture the non-linear signal of the covariate. We note that for younger drivers $\beta_{DrivAge}$ should be increased on average, while drivers in the segment between 22 and 38 should see a reversal in the base model parameter and only for the older drivers the GLM fits well. This gives us an indication for potential to split this feature into three distinct buckets where the experience seems to be different. Furthermore, we note that there is an interaction between DrivAge and BonusMalus – young drivers tend to have higher scores, which is in line with intuition and is revealed by the darker coloured scatter.

Figure 4.5.1.



Now we investigate a potential interaction between VehAge and VehGas by plotting $\alpha'_{i,VehAge}(x)$ against $x_{i,VehAge}$ in anticipation of discovering clear patterns. We note that the greatest corrections for the newest vehicles coincide with those having Regular Vehgas. That interaction also explains the bimodal shape of VehAge in Figure 4.4.2. We also observe that the magnitude of the beta-corrections varies by VehAge thus suggesting both the interaction effect and a non-linear relationship between the predictor and the target variable.

Figure 4.5.2.



5. Conclusions

This paper introduced the Interpretable Boosted Linear Model (IBLM) as a method for closing the interpretability–performance gap in financial predictive modelling. The proposed approach preserves the parametric structure of a GLM while achieving predictive performance approaching that of state-of-the-art gradient boosted tree models. The method combines a base GLM with a residual learner, thereby enhancing predictive accuracy, and leverages transformed SHAP values to recover parameter-level corrections to the underlying linear model. We further demonstrated how the IBLM can be incorporated into an iterative modelling process aimed at maximising the variance explained by the GLM component.

IBLM improves on GLMs by delivering higher predictive accuracy, and on neural networks and other ML models by being easier to fit and operationalize, while retaining a transparent linear representation compatible with existing rating structures. SHAP-corrected betas provide risk insights, supporting more accurate pricing for established products as well as data-driven pricing in new or niche markets where prior actuarial knowledge may be limited.

References

1. Chen, Tianqi & Guestrin, Carlos. *XGBoost: A Scalable Tree Boosting System*. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, 13-17 August 2016, 785-794. https://doi.org/10.1145/2939672.2939785
2. Friedman, J. H. *Greedy Function Approximation: A Gradient Boosting Machine*. *Annals of Statistics*, 1189–1232, October 2001.
3. Grinsztajn, Léo, Oyallon, Edouard & Varoquaux, Gaël. *Why do tree-based models still outperform deep learning on tabular data?* 36th Conference on Neural Information Processing Systems (NeurIPS 2022) Track on Datasets and Benchmarks, Nov 2022, New Orleans, United States.
4. Lundberg, S. M. & Lee, S.-I. *A Unified Approach to Interpreting Model Predictions*. Proceedings of the 31st International Conference on Neural Information Processing Systems, Long Beach, 4-9 December 2017, 4766–4777.
5. Mayer, Michael & Meier, Daniel & Wüthrich, Mario V. *SHAP for Actuaries: Explain any Model* (March 15, 2023). SSRN: https://ssrn.com/abstract=4389797 or http://dx.doi.org/10.2139/ssrn.4389797
6. Merz, Michael, Richman, Ronald, Tsanakas, Andreas & Wüthrich, Mario V. *Interpreting Deep Learning Models with Marginal Attribution by Conditioning on Quantiles* (June 18, 2022). *Data Mining and Knowledge Discovery*, Volume 36, pp. 1335–1370. https://doi.org/10.1007/s10618-022-00841-4
7. Molnar, C. (2019). *Interpretable machine learning: A guide for making black box models explainable*. Leanpub. http://leanpub.com/interpretable-machine-learning
8. Nelder, A. J. & Wedderburn, R. W. M. *Generalized Linear Models*. *Journal of the Royal Statistical Society. Series A (General)*, 1972, Vol. 135, No. 3, pp. 370–384.
9. Noll, Alexander, Salzmann, Robert & Wüthrich, Mario V. *Case Study: French Motor Third-Party Liability Claims*. March 4, 2020.

10. Richman, R. & Wüthrich, M. V. *LocalGLMnet: interpretable deep learning for tabular data*. *Scandinavian Actuarial Journal*, 2023(1), 71–95. https://doi.org/10.1080/03461238.2022.2081816
11. Wuthrich, Mario V., Richman, Ronald, Avanzi, Benjamin, Lindholm, Mathias, Maggi, Marco, Mayer, Michael, Schelldorfer, Jürg & Scognamiglio, Salvatore. *AI Tools for Actuaries* (August 08, 2025). SSRN: https://ssrn.com/abstract=5162304 or http://dx.doi.org/10.2139/ssrn.5162304
12. Young, H. Peyton. *Monotonic solutions of cooperative games*. *International Journal of Game Theory* 14.2 (1985), pp. 65–72.
13. Financial Reporting Council. Technical Actuarial Guidance: Models. Guidance document, published 7 October 2024. The Financial Reporting Council Limited,

Appendices

Appendix 1 – Post-hoc capping and affine correction

If the full IBLM were deployed, large departures from the underlying GLM would not be desirable. We therefore cap the booster's multiplicative correction after training. For a trim level $p \in [0, 1)$ we define a clip function:

$$clip_p(x) := min\{max\{x, 1 - p\}, 1 + p\}$$

The clip function can be extended so that the left tail and right tail clipping are asymmetrical.

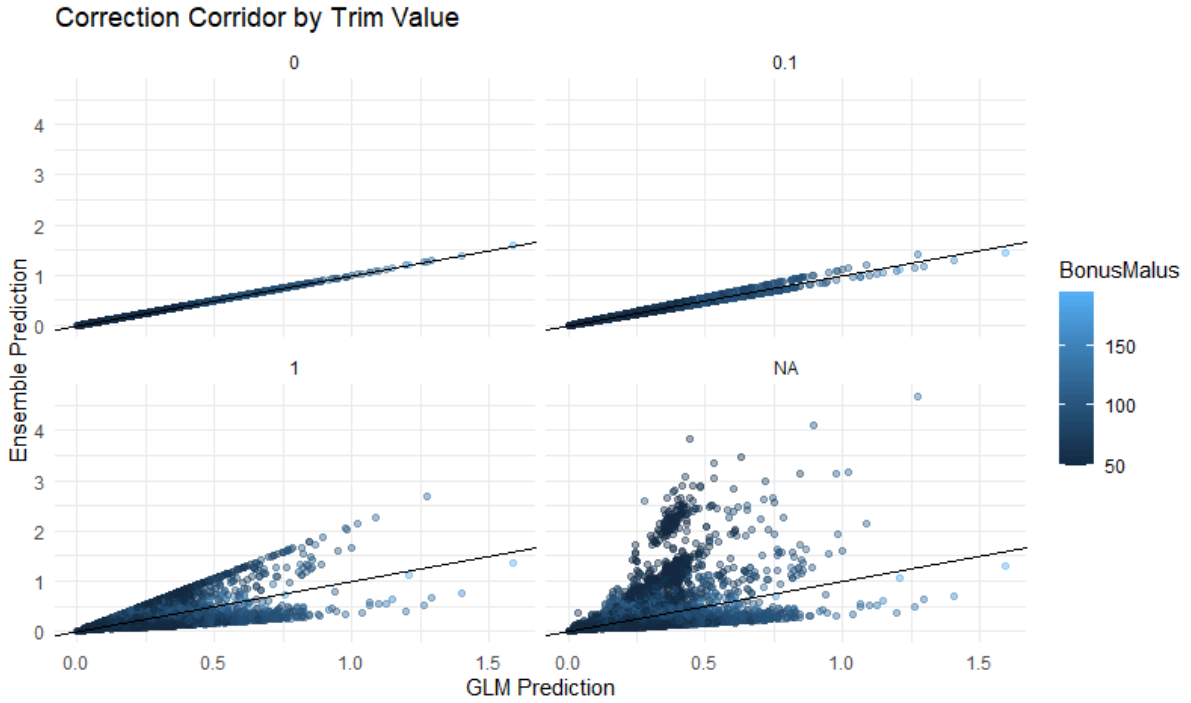
Then we can apply the post-hoc capping and ensure the average correction of the booster has a mean of 1 in the following way:

$$\tilde{y}_{booster}(x_k) = \frac{clip_p(x)}{\frac{1}{2}\sum_{i=1}^{n} clip_p(x_i)}$$

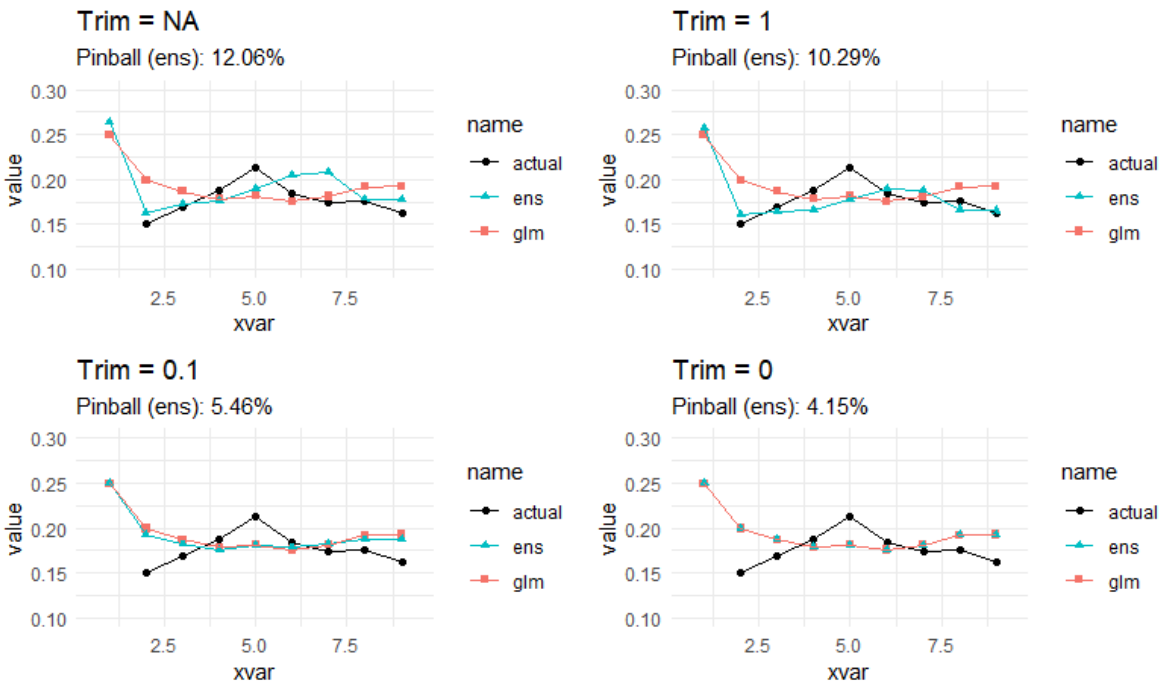
The normalizing constant in the denominator can be trained as an additional model parameter from L and used at inference time. We note that the closer the trim value is to 0, the worse the predictive performance of IBLM is, ultimately matching GLMs performance at $p = 0$.

We can visualize a relationship between the base GLM and the clipped ensemble with a scatter plot. As p decreases, points converge toward the diagonal forming a *funnel* shape, allowing less deviation from the GLM. The scatter can also be colored by a predictor which

will reveal which one of them was the biggest driver of the deviations from the base model. (NA indicates a case of unrestricted deviations from the base model)



The series of AvE charts show that as p gets closer to zero, the closer IBLM predictions converge to the GLMs. This is a desirable property from a practical perspective because it allows a smooth transition from the GLM to IBLM by increasing p through time as stakeholders gain confidence in application of the ML model.



Instead of applying post-hoc capping, one could achieve the same effect during training by modifying the booster's target through an affine transformation, thereby constraining its deviations from the GLM's predictions.

Appendix 2 - R code

In order to reproduce the results and graphs from this manuscript, please refer to this GitHub repository github.com/Karol-Gawlowski/IBLM_BHP and the script paper_code.R