GBMs as Factor Tables: Achieving Both Transparency and Interpretability Without Approximation

Lucas Muzynoski
Avenue Analytics

June 18, 2025

Abstract

The increasing adoption of machine learning in regulated industries is hindered by the competing demands of predictive accuracy and regulatory compliance. While Gradient Boosting Machines (GBMs) offer superior predictive performance, they lack the transparency and interpretability that regulators require. Conversely, traditional Generalized Linear Models (GLMs) provide transparency but often require extensive manual feature engineering to capture complex relationships.

This paper presents a novel methodology that bridges this gap by transforming GBMs into factor tables—a format fully compatible with regulatory requirements in insurance pricing, criminal justice, healthcare, and financial services. Unlike previous approaches that approximate complex models with simpler ones, we demonstrate mathematically that any GBM can be exactly represented as a set of factor tables without information loss, maintaining complete prediction equivalence.

To enhance interpretability while preserving predictive power, we introduce novel L0-like regularization penalties that discourage unnecessary feature interactions both ensemble-wide and within individual trees. These techniques effectively control model complexity while maintaining performance.

We further propose the use of multi-objective tuning that explicitly generates a Pareto frontier of models balancing interpretability against predictive performance. In case studies on recidivism prediction and insurance pricing, our models matched or outperformed benchmark approaches (including EBM and Random Forest) while maintaining complete transparency.

The resulting approach enables practitioners to deploy sophisticated machine learning models that are both interpretable (calculations are understandable) and transparent (the underlying formula is fully disclosed). This addresses a critical gap in current methods, which typically provide either post-hoc explanations without transparency or interpretable models with limited predictive power. Our methodology satisfies regulatory requirements while retaining the predictive advantages of GBMs.

Keywords: Insurance pricing, gradient boosting, model transparency, factor tables, interpretability

1 Introduction

Highly regulated industries face a common challenge: balancing the predictive power of advanced machine learning models with stringent requirements for transparency and interpretability. In domains such as insurance pricing [Goldburd et al., 2016], financial services [Rudin, 2019], healthcare decision-making [Caruana et al., 2015], and public sector resource allocation, regulators demand models that are not only accurate but also fully transparent and auditable.

Traditionally, Generalized Linear Models (GLMs) have been the standard in these regulated environments due to their inherent transparency and interpretability [Rudin, 2019, Garrett and Rudin, 2024], despite their limitations in capturing complex non-linear relationships without extensive manual feature engineering. The emergence of machine learning algorithms, particularly

Gradient Boosting Machines (GBMs), has demonstrated significant improvements in predictive accuracy across multiple domains [Friedman, 2001]. However, their wider adoption in regulated industries has been limited because they lack both transparency (their underlying formula is complex and difficult to share) and interpretability (their calculations are not inherently understandable) [Lipton, 2018, Garrett and Rudin, 2024].

While post-hoc explanation methods such as SHAP values [Lundberg and Lee, 2017] provide some interpretability by explaining individual predictions, they fail to deliver the transparency that regulators increasingly demand, as the underlying model remains a black box [Garrett and Rudin, 2024]. This creates a significant dilemma for practitioners in regulated industries: either sacrifice predictive performance for transparency and interpretability by using traditional statistical models, or employ sophisticated machine learning models with superior predictive power but struggle with regulatory compliance and stakeholder trust. Current solutions addressing this problem through model distillation [Tan et al., 2018] or simplified approximations [Lou et al., 2013] typically result in information loss and reduced accuracy.

In this paper, we present a novel methodology that bridges this gap by transforming GBMs into fully transparent and interpretable factor tables—a format compatible with regulatory requirements in multiple industries—while explicitly managing the trade-off between interpretability and predictive performance.

First, we provide a theoretical foundation showing that any GBM can be exactly represented as a set of factor tables, establishing mathematical equivalence rather than approximation. Second, we introduce modifications to LightGBM that add ensemble-wise and tree-wise L0-like regularization to enhance the interpretability of the resulting factor tables. Third, we develop a multi-objective tuning framework that generates a pareto frontier of optimal models, allowing practitioners to choose parameters that effectively balance predictive performance and interpretability based on their specific requirements.

The resulting solution enables practitioners to deploy sophisticated machine learning models with transparency and interpretability guarantees while retaining much of the predictive advantage of GBMs. These qualities are valuable not only for regulatory compliance but also for building stakeholder trust, facilitating model debugging, enabling knowledge discovery, and supporting ethical decision-making. While our empirical validation includes insurance pricing applications, the methodology is applicable to any domain where understanding model decisions is important, including healthcare [Caruana et al., 2015], finance [Wüthrich and Merz, 2019], criminal justice [Garrett and Rudin, 2024], education, and public policy [Rudin, 2019]. Our experimental results demonstrate that this approach produces factor tables that are intuitive and interpretable for diverse stakeholders—from technical experts to domain specialists to end users—with controllable trade-offs between simplicity and predictive power.

2 Related Work

2.1 Defining Interpretability and Transparency

In machine learning, terms like "interpretability" and "explainability" are often used interchangeably, but recent work has emphasized important distinctions [Lipton, 2018, Garrett and Rudin, 2024]. Following Garrett and Rudin [2024], we distinguish between two key concepts:

Interpretability refers to predictive models whose calculations are inherently understandable. For an interpretable AI system, a person can see how the system works and what information it relies upon in a particular instance. The inner workings of the model are accessible to users, providing clear information about the factors used and how they combine to produce results.

Transparency, in contrast, refers to sharing the underlying formula for the model. This enables independent researchers to conduct evaluations and assess the accuracy of the model. While transparency often facilitates interpretability, a model can be transparent yet not inter-

pretable (when the formula is available but too complex to understand), or interpretable but not transparent (when reasoning for individual predictions is provided without access to the full model).

These distinctions are crucial for our work, as we aim to achieve both interpretability and transparency—creating models whose calculations are understandable while also fully disclosing their underlying structure.

2.2 Current Approaches to Machine Learning Interpretability

2.2.1 Post-hoc Explanation Methods: Interpretability Without Complete Transparency

Several widely-used techniques provide explanations for black-box models after training. SHAP (SHapley Additive exPlanations) values [Lundberg and Lee, 2017] and LIME (Local Interpretable Model-agnostic Explanations) [Ribeiro et al., 2016] explain individual predictions by assigning importance values to each feature, offering a form of local interpretability. Partial Dependence Plots (PDPs) [Friedman, 2001] and Accumulated Local Effects (ALE) plots [Apley and Zhu, 2020] visualize average feature effects across the dataset, providing global insights.

While valuable, these methods provide interpretability without transparency—they explain predictions without fully revealing the model's underlying structure. This creates several limitations: explanations may not faithfully represent model behavior [Rudin, 2019], can be inconsistent across instances, and do not enable stakeholders to fully understand the model's complete decision logic.

2.2.2 Inherently Interpretable Models: Achieving Both Interpretability and Transparency

Generalized Linear Models (GLMs) [Nelder and Wedderburn, 1972, McCullagh and Nelder, 1989] have been the historical standard for both interpretability and transparency. Their parameters are directly interpretable as feature effects, and their structure is completely transparent. However, they often require extensive manual feature engineering to capture complex relationships.

Explainable Boosting Machines (EBMs) [Nori et al., 2019] represent a more recent approach, using generalized additive models with boosting techniques to maintain interpretability while improving predictive power. EBMs train on one feature at a time and can include pre-specified pairwise interactions, achieving accuracy comparable to ensemble methods. Although EBMs are interpretable, full transparency is somewhat limited by the inability to efficiently display full model details. Manual calculation of predictions is not easily done.

2.2.3 Complex Models with Published Details: Transparency Without Interpretability

Some researchers publish comprehensive details of complex models, including architectures, hyperparameters, and sometimes even weights, providing transparency. However, these models often remain practically uninterpretable due to their complexity—the formula is available but incomprehensible to human understanding. As Garrett and Rudin [2024] note, a model can be transparent but not interpretable when its formula is too complicated to understand.

2.3 Approaches to Enhanced Interpretability

2.3.1 Enhancing Linear Models

Researchers have developed various techniques to improve the predictive power of inherently transparent and interpretable models like GLMs while preserving their interpretability. LASSO [Tibshirani, 1996] and Elastic Net [Zou and Hastie, 2005] regularization implement penalties

that encourage sparsity, enabling GLMs to perform automatic feature selection and handle high-dimensional data more effectively. These approaches maintain the transparency and interpretability advantages of linear models while addressing some of their limitations.

Despite these improvements, enhanced linear models still face fundamental constraints in modeling complex non-linear relationships and interactions. They require extensive manual basis expansion to capture such patterns, often demanding significant domain expertise and experimentation. This manual basis expansion process—where modelers explicitly create bins, splines, interaction terms, and other transformations—is time-consuming and may miss important patterns in the data. This creates a gap where even advanced linear methods struggle to match the predictive performance of more complex models.

Our approach takes a different direction—instead of enhancing simpler models to improve accuracy, we transform complex, high-performing models to achieve transparency and interpretability without sacrificing their predictive advantages, while automatically discovering an effective basis expansion.

2.3.2 Model Distillation Approaches

Model distillation techniques attempt to transfer knowledge from complex models to simpler, more interpretable ones. Recent work has focused on distilling gradient boosting machines into generalized additive models (GAMs) [Lou et al., 2012, 2013] or traditional GLMs [Tan et al., 2018, Maillart and Robert, 2024]. Hara and Hayashi [2018] addressed tree ensemble complexity through Bayesian model selection. These approaches improve interpretability but typically sacrifice some predictive accuracy and only approximate the original model behavior rather than providing exact equivalence.

2.4 Interpretability and Transparency Requirements

Many domains require models that are both interpretable and transparent. Insurance pricing exemplifies this through established practices using GLMs and factor tables [Goldburd et al., 2016]. Actuarial literature emphasizes the importance of interpretable models for pricing, underwriting, and reserving [Wüthrich and Merz, 2019]. Similar requirements exist in healthcare [Caruana et al., 2015] and criminal justice [Garrett and Rudin, 2024].

The need for both interpretability and transparency arises from several concerns: ensuring fairness, enabling stakeholder trust, facilitating model debugging, supporting knowledge discovery, and enabling ethical decision-making. While post-hoc explanation methods offer some degree of interpretability, they fall short of providing the full transparency needed in many applications [Rudin, 2019].

2.5 Gap in Current Approaches

Despite numerous advances in machine learning interpretability, a significant gap remains. Current approaches either: 1. Provide post-hoc interpretability without full transparency (SHAP, LIME) 2. Offer both interpretability and transparency but with limited predictive power (GLMs) 3. Achieve better prediction accuracy but sacrifice interpretability, transparency, or both (complex ML models) 4. Approximate complex models with simpler ones, losing information in the process (distillation approaches)

Our work addresses this gap by developing a method that transforms GBMs into factor tables that are both interpretable (calculations are understandable) and transparent (the underlying formula is fully shared), without sacrificing the predictive advantages of the original model. Unlike previous approaches, our method provides an exact representation rather than an approximation, preserving the full predictive power of the GBM while making it accessible to stakeholders across technical, business, and regulatory domains.

3 Background

3.1 Generalized Linear Models and Factor Tables

Generalized Linear Models (GLMs) [Nelder and Wedderburn, 1972] are widely used in regulated industries due to their transparency. Let $V = \{v_1, v_2, \ldots, v_p\}$ represent the original raw features (such as driver age, vehicle type, etc.). In practice, these raw features are rarely used directly in GLMs. Instead, a basis expansion transforms them into model inputs $X = \{x_1, x_2, \ldots, x_m\}$ through operations such as binning, one-hot encoding, splines, polynomials, or other transformations. In a GLM, the expected value of the response variable is related to a linear combination of these expanded features through a link function:

$$g(\mathbb{E}[Y]) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_m x_m \tag{1}$$

where $g(\cdot)$ is the link function.

Colloquially, factor tables are a table of multiplicative factors as part of a rating algorithm. Here we use a more general and formal definition.

Definition 1 (Factor Table)**.** A factor table is a function $F(V)$ that maps from a subset of features $S \subseteq V$ to a set of coefficients, where the mapping creates mutually exclusive and exhaustive categories. Each unique combination of feature values in S maps to exactly one coefficient value.

In insurance pricing, GLMs are converted into multiplicative factor tables for implementation and regulatory filing. With a log link function, additive factor tables are typically exponentiated, allowing predictions to be calculated by multiplying factors rather than adding them and applying the inverse link function.

Proposition 1. Any GLM with categorical features (or discretized numeric features) can be exactly represented as a sum of factor tables, subject to the same link function.

Formally, a GLM can be rewritten as:

$$g(\mathbb{E}[Y]) = \beta_0 + \beta_1 x_1 + \ldots + \beta_m x_m = \beta_0 + F_1(V) + F_2(V) + \ldots + F_k(V) \tag{2}$$

Because we required that X consists of categorical or discretized numeric features, X consists of one-hot encoded features. This can be re-written as a linear combination of indicator functions, showing the equivalence.

The methodology in the next section bridges the gap between GLMs and GBMs by showing that we can also transform a GBM into transparent factor tables without affecting the predictions.

4 Methodology

4.1 Converting a GBM into Factor Tables

4.1.1 Definitions

Definition 2 (Decision Tree)**.** A decision tree $f(V)$ is a recursive partitioning of the feature space that creates a set of mutually exclusive and exhaustive regions $R_1, R_2, \ldots, R_m$. Each region R_j is associated with a constant prediction value γ_j. The tree's output for any input V is given by:

$$f(V) = \sum_{j=1}^{m} \gamma_j \cdot \mathbb{1}(V \in R_j) \tag{3}$$

where $\mathbb{1}(\cdot)$ is the indicator function that equals 1 when V belongs to region R_j and 0 otherwise.

Definition 3 (Gradient Boosting Machine)**.** A gradient boosting machine (GBM) is an ensemble model that sequentially builds decision trees to minimize a loss function. For a prediction task with input features $V = \{v_1, v_2, \ldots, v_p\}$ and target variable Y, a GBM with n trees can be expressed as:

$$g(\mathbb{E}[Y]) = \beta_0 + \alpha \sum_{i=1}^{n} f_i(V) \tag{4}$$

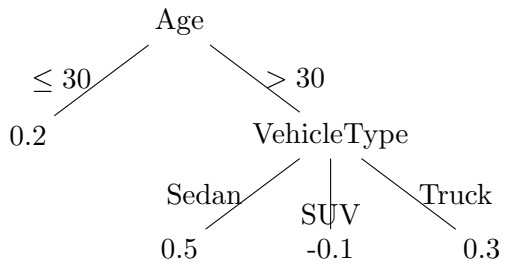
where $g(\cdot)$ is the link function, β_0 is the initial prediction, α is the learning rate, and each $f_i(V)$ is a decision tree. In the original gradient boosting formulation (Friedman, 2001), each tree is fitted to the negative gradient of the loss function with respect to the current model prediction. Modern implementations like XGBoost [Chen and Guestrin, 2016] and LightGBM [Ke et al., 2017] additionally utilize second-order approximations with Hessians to improve optimization.

4.1.2 Equivalence of Trees and Factor Tables

Proposition 2. Every decision tree can be exactly represented as a factor table.

A decision tree partitions the feature space into mutually exclusive and exhaustive regions (leaf nodes), with each region defined by a unique combination of feature conditions along its path from root to leaf. Each leaf node maps to exactly one prediction value. This structure directly satisfies our definition of a factor table: a function mapping from feature combinations to coefficients, where the mapping creates mutually exclusive and exhaustive categories. The factor table simply records which combination of feature values leads to which leaf node, preserving the tree's exact predictive behavior.

Example 1. Consider a simple decision tree for car insurance pricing with two features: Age (continuous) and VehicleType (categorical):



This tree can be exactly represented as the following factor table:

Age Upper Bound	VehicleType	Factor Value
30	Sedan	0.2
30	SUV	0.2
30	Truck	0.2
∞	Sedan	0.5
∞	SUV	-0.1
∞	Truck	0.3

For any input combination of Age and VehicleType, exactly one row in this table applies, giving the same prediction as would be obtained by traversing the decision tree.

4.1.3 Tree and Factor Table Operations

We now formalize the fundamental operations for decomposing trees into factor tables and combining them into consolidated representations.

Definition 4 (Tree Decomposition)**.** A decision tree $f(V)$ with nodes $\{N_1, N_2, \ldots, N_m\}$ (including both internal and leaf nodes) can be decomposed into m factor tables $\{F_1, F_2, \ldots, F_m\}$, where each F_j corresponds to a node N_j and captures its contribution:

$$f(V) = \sum_{j=1}^{m} F_j(S_j) \tag{5}$$

where $S_j \subseteq V$ is the subset of features used in the path to node N_j, and F_j outputs the node's contribution value when all path conditions to that node are satisfied and 0 otherwise. Internal nodes capture main effects of features, while leaf nodes capture remaining effects after accounting for all decision splits.

Definition 5 (Factor Table Consolidation)**.** Factor table consolidation is the process of merging multiple factor tables into a more compact set of tables that preserves the original model's behavior. This operation acts as the inverse of decomposition by recombining tables based on their feature dependencies.

Given a set of factor tables $\{F_1(S_1), F_2(S_2), \ldots, F_k(S_k)\}$ from multiple nodes across trees, consolidation proceeds by:

1. *Grouping by feature sets*: Tables are grouped according to the features they depend on.
2. *Value aggregation*: Within each group, factor values are aggregated by summing contributions that apply to the same feature value combinations:

$$F_{\text{consolidated}}(s) = \sum_{i:S_i \subseteq \text{features}(s)} F_i(s[S_i]) \tag{6}$$

where $s[S_i]$ represents the subset of feature values from s that correspond to features in S_i. This operation extracts only the relevant feature values needed by each factor table.

These fundamental operations provide the mathematical foundation for transforming complex tree-based models into interpretable factor tables while maintaining exact functional equivalence.

4.1.4 From GBM to Factor Tables

Corollary 1. Any gradient boosting machine can be exactly represented as a sum of factor tables.

This corollary follows directly from our earlier definitions. Since a GBM is an ensemble of decision trees, and each decision tree can be transformed into factor tables, the entire GBM can be expressed as a sum of these factor tables.

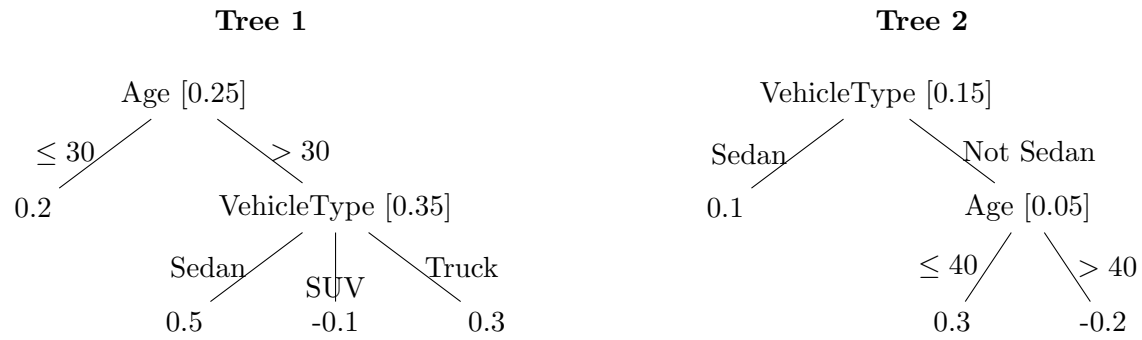
While this direct conversion establishes mathematical equivalence, the resulting collection might contain hundreds or thousands of tables, hindering interpretability. To address this, we apply our decomposition and consolidation operations with two specific strategies:

- **ANOVA-style consolidation**: Group and combine factor tables that share exactly the same feature sets. This approach separates main effects from interaction effects, creating distinct tables for:
  - *Main effects*: One table per feature, showing its isolated impact
  - *Interaction effects*: Separate tables for each specific feature combination
- **Full consolidation**: Combine factor tables hierarchically, where one table's feature set can be contained within another's. This produces fewer, more comprehensive tables that integrate main effects with their interactions.

Both strategies preserve mathematical equivalence to the original GBM while offering different interpretability benefits. ANOVA-style consolidation provides clearer insights into feature contributions and interactions, making it ideal for model analysis and stakeholder explanations. Full consolidation creates a more compact representation, beneficial for implementation and regulatory filings.

To illustrate how this conversion works in practice, consider the following example of converting a simple two-tree ensemble into factor tables:

Example 2. Consider a simple model as the sum of the following two trees, where each node is labeled with its internal value in brackets, and leaf nodes contain their leaf values:



Using the internal node values, we can decompose each tree into main effects and interactions:
First Tree Decomposition:

Age Main Effect

Age	Factor Value
≤ 30	0.2
> 30	0.35

Age × VehicleType Interaction Effect

Age	VehicleType	Factor Value
> 30	Sedan	0.5 - 0.35 = 0.15
> 30	SUV	-0.1 - 0.35 = -0.45
> 30	Truck	0.3 - 0.35 = -0.05
≤ 30	*	0

Second Tree Decomposition:

VehicleType Main Effect

VehicleType	Factor Value
Sedan	0.1
Not Sedan (SUV/Truck)	0.05

Age × VehicleType Interaction Effect

Age	VehicleType	Factor Value
≤ 40	Sedan	0
≤ 40	SUV/Truck	0.3 - 0.05 = 0.25
> 40	Sedan	0
> 40	SUV/Truck	-0.2 - 0.05 = -0.25

Now we consolidate these tables across both trees to create our final factor tables:

Consolidated Age Main Effect

Age	Factor Value
≤ 30	0.2
$30 < \text{Age} \leq 40$	0.35
> 40	0.35

Consolidated VehicleType Main Effect

VehicleType	Factor Value
Sedan	0.1
SUV	0.05
Truck	0.05

Consolidated Age × VehicleType Interaction

Age	VehicleType	Factor Value
≤ 30	Sedan	0
≤ 30	SUV	0.25
≤ 30	Truck	0.25
$30 < \text{Age} \leq 40$	Sedan	0.15
$30 < \text{Age} \leq 40$	SUV	-0.45 + 0.25 = -0.2
$30 < \text{Age} \leq 40$	Truck	-0.05 + 0.25 = 0.2
> 40	Sedan	0.15
> 40	SUV	-0.45 + (-0.25) = -0.7
> 40	Truck	-0.05 + (-0.25) = -0.3

These factor tables together exactly reproduce the predictions of the original two-tree ensemble. For any input, we simply look up the appropriate values in each table and sum them. For example, for a 25-year-old with an SUV: - From Age table: 0.2 - From Vehicle table: 0.05 - From Interaction table: 0.25 Yielding a total prediction of 0.2 + 0.05 + 0.25 = 0.5

Similarly, for a 35-year-old with an SUV: - From Age table: 0.35 - From Vehicle table: 0.05 - From Interaction table: -0.2 Yielding a total prediction of 0.35 + 0.05 + (-0.2) = 0.2

We can also create a fully consolidated table that combines main effects with interactions:

Fully Consolidated Factor Table

Age	VehicleType	Factor Value
≤ 30	Sedan	0.2 + 0.1 + 0 = 0.3
≤ 30	SUV	0.2 + 0.05 + 0.25 = 0.5
≤ 30	Truck	0.2 + 0.05 + 0.25 = 0.5
$30 < \text{Age} \leq 40$	Sedan	0.35 + 0.1 + 0.15 = 0.6
$30 < \text{Age} \leq 40$	SUV	0.35 + 0.05 + (-0.2) = 0.2
$30 < \text{Age} \leq 40$	Truck	0.35 + 0.05 + 0.2 = 0.6
> 40	Sedan	0.35 + 0.1 + 0.15 = 0.6
> 40	SUV	0.35 + 0.05 + (-0.7) = -0.3
> 40	Truck	0.35 + 0.05 + (-0.3) = 0.1

This single table completely captures the behavior of the original two-tree ensemble. For any input, only one row applies, producing the exact same prediction as the separate tables or the original trees.

4.1.5 Intuitive Understanding of GLM and GBM Equivalency

The equivalence between GBMs and factor tables and GLMs can be understood by examining how both GBMs and GLMs parameterize features:

In a standard GLM workflow, two distinct steps occur:

1. The modeler manually creates a basis (model matrix) of features (e.g. binning, splines, interactions) [McCullagh and Nelder, 1989, Hastie et al., 2009].

2. Model parameters are estimated via maximum likelihood.

A GBM such as XGBoost or LightGBM effectively automates this first step:

1. Trees adaptively partition the feature space, greedily expanding the basis [Friedman, 2001, Zhang and Yu, 2005].

2. Leaf values are estimated through a second-order (Newton–Raphson) update with explicit regularization [Chen and Guestrin, 2016].

Each split in a GBM creates categorical boundaries—discovering useful feature transformations that a modeler might otherwise specify by hand. Converting a GBM to factor tables simply extracts this learned basis into a transparent, regulatory-compliant format.

This explains why GBMs often outperform manually specified GLMs: GBMs both discover and fit effective basis functions in one coherent procedure. Our conversion method preserves the full predictive power of the GBM while exposing its automatically learned basis.

4.2 Controlling Model Complexity

Although we can use the techniques described this far to convert any GBM into transparent factor tables, to increase the interpretability of those tables, we need to control model complexity.

Our implementation, as discussed more in Experimental Setup, is based on LightGBM. To control model complexity and create a more interpretable model, we leverage both existing LightGBM hyperparameters and introduce two new parameters with L0-like penalization variations.

4.2.1 Using Existing LightGBM Parameters

Before introducing specialized L0-like regularization, we can leverage LightGBM's built-in hyperparameters to control model complexity. While these parameters were not specifically designed for interpretability, they significantly impact the structure of the resulting factor tables:

Early stopping further controls model size by preventing unnecessary trees once performance plateaus on cross-validation. While these parameters help create more interpretable models, they don't specifically target feature interactions across the ensemble—a limitation addressed by our L0-like regularization techniques in the next section.

4.2.2 New L0-like Penalization Parameters

To effectively control model complexity, we implement two complementary L0-like regularization mechanisms:

First, we introduce a feature interaction penalty that discourages the model from creating new feature combinations that haven't appeared in previous trees:

$$\mathcal{L}_{split} = \mathcal{L}_{original} + \lambda_0 \cdot N_f \cdot \nVdash_{new\_ensemble} \tag{7}$$

where $\mathcal{L}_{original}$ is the original split criterion, λ_0 is the regularization strength, N_f is the number of features used in the current tree path including the candidate feature, and $\nVdash_{new\_ensemble}$ is an indicator that equals 1 if the resulting feature combination hasn't been used in any previous tree. Importantly, this penalty is zero if (1) the feature is already used in the current tree path, or (2) the resulting feature combination is a subset of previously used feature combinations across the ensemble.

Parameter	Effect on Factor Tables	Recommended Setting
`max_depth`	Directly limits interaction order to `max_depth-1`	3-5
`num_leaves`	Limits unique paths per tree	4-16 (well below $2^{\texttt{max\_depth}}$)
`learning_rate`	Higher rates reduce number of trees needed	0.1-0.5 to minimize tree count
`colsample_bytree`	Reduces chance of complex interactions	0.7-1.0 to limit feature combinations
`min_data_in_leaf`	Prevents overly specific interaction patterns	Higher values (50+) for smoother factors
`min_gain_to_split`	Prevents splits with minimal improvement, reducing weak interactions	0.1-1.0 to eliminate insignificant branches
`max_bin`	Sets maximum number of splits per feature, creating smaller factor tables	15-255

Table 1: LightGBM parameters affecting factor table complexity

Second, we implement a tree complexity penalty that discourages increasing the number of unique features within each individual tree. This results in the model preferring main-effects and lower order interactions before using higher order interactions, even if they've been previously introduced into the ensemble:

$$\mathcal{L}_{split} = \mathcal{L}_{original} + \lambda_c \cdot N_f \cdot \mathbb{1}_{new\_in\_tree} \tag{8}$$

where λ_c is the complexity regularization strength, N_f is again the number of features in the tree path after adding the candidate feature, and $\mathbb{1}_{new\_in\_tree}$ indicates whether the candidate feature is new to the current tree only. This penalty applies only when introducing features not previously used in the current tree, encouraging feature reuse within individual trees.

Together, these penalties, along with LightGBM's built-in parameters, produce more interpretable GBMs, but require carefully balancing complexity reduction against predictive performance.

4.3 Multi-objective Tuning

To systematically navigate this interpretability-performance trade-off, we formalize a bi-objective optimization problem:

$$\min_{\theta \in \Theta} \{-\text{CV}(\theta), C(\theta)\} \tag{9}$$

where θ represents the hyperparameter configuration, $\text{CV}(\theta)$ is the cross-validation performance metric (e.g., log-likelihood or Gini coefficient), and $C(\theta)$ quantifies model complexity or the inverse of interpretability. This formulation explicitly acknowledges that maximizing predictive performance (minimizing $-\text{CV}(\theta)$) and maximizing interpretability (minimizing $C(\theta)$) are competing objectives.

We chose the median number of resulting factor tables during cross-validation (using full consolidation) as our primary metric to approximate interpretability. Alternative proxies for

interpretability could include the total parameter count, the maximum interaction order, or L1/L2 norms of model parameters. The framework is flexible enough to accommodate these alternative formulations based on domain-specific interpretability requirements. Which metric best approximates interpretability is an open research question.

We implement this optimization using Optuna [Akiba et al., 2019], which employs Bayesian optimization with a Tree-structured Parzen Estimator to efficiently explore the hyperparameter space. Our implementation leverages Optuna's multi-objective capabilities to construct a Pareto frontier of non-dominated solutions—configurations where improving one objective necessarily degrades the other.

From this set of Pareto-optimal hyperparameter configurations, the practitioner can choose the configuration that best meets their goals before fitting their final model.

5 Case Studies

5.1 Broward County Recidivism

We first present our methodology on the Broward County Recidivism dataset released by ProPublica [Angwin et al., 2016]. Preprocessing of the data follows Rudin [Rudin et al., 2020] to create a filtered dataset with derived features such as age at screening and prior offense counts. The dataset includes a set of features, the decile of the COMPAS recidivism score, and the actual two-year recidivism status. Our target variable is two-year recidivism status (binary classification), which allows direct comparison with the COMPAS score that was designed to predict this outcome [Northpointe Inc., 2009]. This dataset provides an ideal test case for our approach as it involves high-stakes decision making where model transparency is paramount [Garrett and Rudin, 2024].

5.1.1 Modeling Setup

The recidivism prediction models were constructed using a 70/30 train/test split. We included seven features in our analysis: age, juvenile felony count (`p_juv_fel_count`), juvenile misdemeanor count (`p_juv_misd_count`), other juvenile offense count (`p_juv_other_count`), prior charge type (`p_charge`), sex, and current charge degree (`c_charge_degree`). Model hyperparameters were optimized through 100 trials using 3-fold cross-validation.

5.1.2 Model Fit

The model output, with full consolidation, is an intercept of -0.185044 and two relatively simple factor tables. To create a probability prediction, we sum the intercept with the corresponding factor from each table, and then apply the inverse logistic link function.

Using our methodology, we convert a LightGBM model with 123 trees to this simple two-table model with no change in predictions.

5.1.3 Broward Results on Test Set

While EBM demonstrates marginally superior AUC compared to our proposed approach, it presents limitations in terms of comprehensive interpretability. Specifically, EBM incorporates all available features without implementing feature selection methodology and utilizes seven interaction terms. Although EBM provides visualization capabilities through interactive plotting utilities for examining feature and interaction contributions, it lacks the capacity for straightforward manual prediction calculation. In contrast, our method achieves both feature and interaction selection through L0-like regularization penalties optimized via cross-validation, resulting in a more parsimonious and computationally transparent model structure.

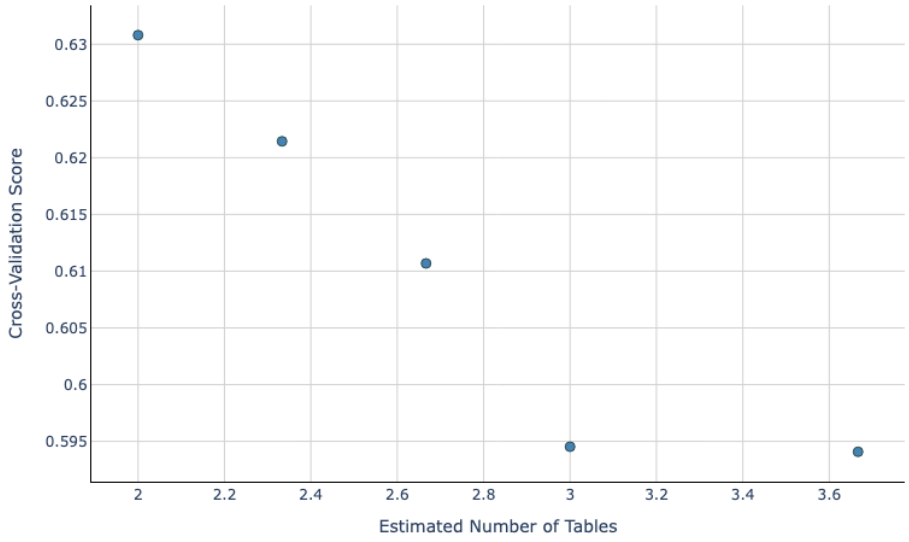


Figure 1: Broward Recidivism Tuning Results: Pareto frontier showing the trade-off between cross-validation log-loss and the median number of consolidated tables produced.

Table 2: Factor table for Prior Charge Count and Sex interaction

Prior Charge Count (Upper Bound)	**Sex**	**Factor**
0	F	-1.1001
1.5	F	-0.4956
2.5	F	0.0441
3.5	F	0.4723
5.5	F	0.5567
6.5	F	0.8860
7.5	F	1.1617
9.5	F	1.2109
14.5	F	1.4425
17.5	F	1.4603
∞	F	1.5309
0	M	-0.7750
1.5	M	-0.3211
2.5	M	0.0972
3.5	M	0.5723
5.5	M	0.6568
6.5	M	0.9860
7.5	M	1.2617
9.5	M	1.2617
14.5	M	1.5314
17.5	M	1.5492
∞	M	1.6198

Table 3: Factor table for Sex and Age interaction

Sex	**Age**	**Factor**
F	20.5	1.4170
F	21.5	1.0817
F	22.5	0.8567
F	24.5	0.3637
F	27.5	0.2214
F	29.5	0.1139
F	33.5	0.0772
F	34.5	-0.1892
F	44.5	-0.4018
F	48.5	-0.5613
F	50.5	-0.5589
F	52.5	-0.6382
F	∞	-0.9604
M	20.5	1.4702
M	21.5	1.1362
M	22.5	0.9111
M	24.5	0.4181
M	27.5	0.2758
M	29.5	0.1683
M	33.5	0.0702
M	34.5	-0.1962
M	44.5	-0.4088
M	48.5	-0.5682
M	50.5	-0.5658
M	52.5	-0.6452
M	∞	-0.9674

Table 4: Comparison of AUC by model on Test Data

Metric	**COMPAS Model**	**Random Forest**	**EBM**	**Our Model**
AUC	0.6961	0.6757	0.7278	0.7258

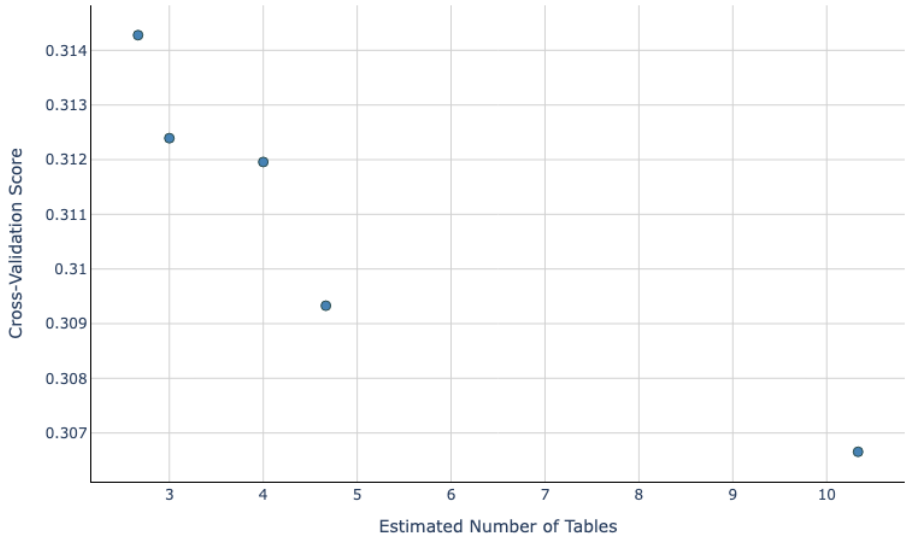


Figure 2: French Motor Data Tuning Results: Pareto frontier showing the trade-off between cross-validation Poisson negative log-likelihood and median number of consolidated tables.

5.1.4 Broward Study Discussion

With our method, we produced a model that uses just three features– Age, Sex, and Prior Charges. When consolidated, it uses just two factor tables, but it produced AUC results that were near the best-in-class, and outperformed both the black-box random forest model as well as the proprietary COMPAS model which also requires the results of a questionnaire with 137 questions [Northpointe Inc., 2009]. At least for this test dataset, a judge could have hypothetically looked up two numbers on the factor tables, added them together, and got prediction that is at least as good as the secret COMPAS model.

5.2 French Insurance Dataset

Insurance pricing models are generally subject to regulator approval, and they are typically filed as sets of factor tables. These models must balance the competing goals of interpretability and predictive performance. We demonstrate our methodology using the French Motor Third-Party Liability (MTPL) insurance dataset, commonly known as "freMTPL2freq" Charpentier [2014].

This dataset contains information on 678,013 insurance policies including policyholder characteristics, vehicle details, and resulting claim counts, representing a standard benchmark in actuarial literature for claim frequency modeling.

5.2.1 Model Fit

From our tuning results, we have the choice of hyperparameter specification from the Pareto frontier. We chose specification 2 (second from left on Pareto plot), which has 543 trees and produces a model that effectively uses only Vehicle Age, Vehicle Gas Type (Diesel or Gasoline), Bonus-Malus coefficient, and Driver Age.

As another point of comparison, we also fit the best CV model (right-most on Pareto plot). We show test set results for each in the following section.

Table 5: Insurance Study Results Comparison

Model	MAE	MSE	Mean Poisson Deviance
Random Forest	0.2491	1.0851	0.6898
EBM	0.1850	0.5326	0.5994
Ours (More Interpretable)	0.1856	0.5350	0.5934
Ours (Best CV)	0.1839	0.5316	0.5834

5.2.2 Insurance Results on Test Set

Our best cross-validation model achieves superior out-of-sample performance compared to all benchmark models, including other transparent approaches and the black box random forest. While this model offers the highest predictive power, it sacrifices some degree of interpretability due to having a larger number of tables and factors. In contrast, our alternative model, selected to prioritize interpretability effectively utilizes only a subset of features while maintaining CV scores comparable to the EBM. This more parsimonious solution enables straightforward manual calculation, making it, in our assessment, more interpretable than the EBM in this particular application.

5.2.3 Insurance Study Discussion

Despite this public dataset being simpler and containing fewer features than proprietary commercial insurance data, our results demonstrate the viability of creating fully transparent and interpretable models that maintain competitive predictive performance.

Our approach offers two key advantages for practical implementation. First, by extending the powerful and widely used LightGBM package, our method readily scales to commercial applications with larger, more complex datasets. Second, the resulting output is already structured as factor tables—the standard format most carriers use when filing their models. This substantially reduces translation errors during implementation.

Furthermore, beyond the Poisson regression demonstrated here, LightGBM also implements Tweedie regression, which is particularly well-suited for insurance pure premium modeling Goldburd et al. [2016].

6 Discussion

This paper presents an approach to convert Gradient Boosting Machines into factor tables, creating models that achieve high predictive performance while maintaining full transparency and interpretability. Our findings have several important implications:

The experimental results demonstrate that it is possible to achieve state-of-the-art predictive performance with models that are both transparent and interpretable. However, we acknowledge that as models incorporate large numbers of features or complex interactions, some interpretability may be sacrificed, though transparency is preserved.

From a computational perspective, our approach efficiently handles large numbers of trees. However, exploring deep interactions (particularly with `num_leaves` $\gg 4$ and/or `max_depth` $\gg 3$) can become computationally intensive. This presents a practical limitation that future algorithmic improvements may address.

For regulated industries such as insurance, healthcare, and finance, our approach provides a direct pathway for interpreting and filing GBM-based models National Association of Insurance Commissioners [2020], Garrett and Rudin [2024]. This addresses a critical need in domains where model decisions must be explainable to regulators and stakeholders.

A notable advantage of our method is that it does not require bespoke modeling code. By leveraging proven libraries like LightGBM with minor modifications, practitioners can implement our approach with minimal additional development effort. The primary requirement is the conversion mechanism from the trained model into factor table format.

Despite these advantages, our approach cannot yet fully replace traditional statistical modeling for inference purposes. The coefficients are already regularized (containing bias), and there is not a clear pathway to confidence intervals or prediction intervals, which limits certain statistical applications McCullagh and Nelder [1989].

7 Conclusion and Future Work

This paper demonstrates that GBMs can be transformed into factor tables to create models that maintain high predictive performance while providing full transparency and interpretability. Our approach bridges the gap between black-box models and interpretable models, offering a promising solution for domains where both performance and explainability are critical.

Several directions for future research emerge from this work:

First, the development of better quantitative definitions of interpretability would help standardize evaluation across different approaches Lipton [2018]. Currently, interpretability remains somewhat subjective, making it difficult to compare different methods objectively.

Second, a full exploration of alternative working definitions for model complexity (such as total parameters, interaction depth, or feature importance distribution) would provide more nuanced ways to balance performance and interpretability.

Third, theoretical justification for optimal ways of regularizing GBMs specifically for interpretability and performance would strengthen the foundation of this approach. This includes investigating how different regularization techniques affect both the predictive performance and the resulting factor table structure.

Finally, developing methods for confidence intervals and prediction intervals would address the current limitations regarding statistical inference Hastie et al. [2009]. This includes exploring bootstrapping techniques and other approaches that could better unify statistical modeling and machine learning paradigms.

By addressing these challenges, future work can further enhance the practical utility of transparent models in high-stakes decision-making contexts, where interpretability is not merely desirable but essential Caruana et al. [2015], Rudin [2019].

Acknowledgments

Carolyn Olsen provided helpful feedback on the manuscript. Jerry Claghorn for an insightful discussion about the potential interpretability of GBMs.

References

Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2623–2631. ACM, 2019.

Julia Angwin, Jeff Larson, Surya Mattu, and Lauren Kirchner. Machine bias: Compas recidivism algorithm data. `https://github.com/propublica/compas-analysis`, 2016. URL `https://github.com/propublica/compas-analysis/blob/master/compas-scores-two-years.csv`. Retrieved May 20, 2025.

Daniel W Apley and Jingyu Zhu. Visualizing the effects of predictor variables in black box supervised learning models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 82(4):1059–1086, 2020.

Rich Caruana, Yin Lou, Johannes Gehrke, Paul Koch, Marc Sturm, and Noemie Elhadad. Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1721–1730, 2015.

Arthur Charpentier. Computational actuarial science with r. *CRC Press*, 2014.

Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 785–794, 2016.

Jerome H Friedman. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232, 2001.

Brandon L. Garrett and Cynthia Rudin. The right to a glass box: Rethinking the use of artificial intelligence in criminal justice. *Cornell Law Review*, 2024.

Mark Goldburd, Anand Khare, and Dan Tevet. *Generalized linear models for insurance rating*. Casualty Actuarial Society, 2016.

Satoshi Hara and Kohei Hayashi. Making tree ensembles interpretable: A bayesian model selection approach. In Amos Storkey and Fernando Perez-Cruz, editors, *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, volume 84 of *Proceedings of Machine Learning Research*, pages 77–85. PMLR, 09–11 Apr 2018. URL `https://proceedings.mlr.press/v84/hara18a.html`.

Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer, 2nd edition, 2009.

Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. Lightgbm: A highly efficient gradient boosting decision tree. In *Advances in Neural Information Processing Systems*, pages 3146–3154, 2017.

Zachary C Lipton. The mythos of model interpretability. *Queue*, 16(3):31–57, 2018.

Yin Lou, Rich Caruana, and Johannes Gehrke. Intelligible models for classification and regression. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 150–158, 2012.

Yin Lou, Rich Caruana, Johannes Gehrke, and Giles Hooker. Accurate intelligible models with pairwise interactions. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 623–631, 2013.

Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems*, pages 4765–4774, 2017.

Arthur Maillart and Christian Robert. Distill knowledge of additive tree models into generalized linear models: a new learning approach for non-smooth generalized additive models. *Annals of Actuarial Science*, 18(3):692–711, 2024. doi: 10.1017/S1748499524000241.

Peter McCullagh and John A. Nelder. *Generalized Linear Models*. Chapman and Hall, 1989.

National Association of Insurance Commissioners. Principles for artificial intelligence (ai). Technical report, NAIC, 2020.

John A Nelder and Robert WM Wedderburn. Generalized linear models. *Journal of the Royal Statistical Society: Series A (General)*, 135(3):370–384, 1972.

Harsha Nori, Samuel Jenkins, Paul Koch, and Rich Caruana. Interpretml: A unified framework for machine learning interpretability, 2019. URL `https://arxiv.org/abs/1909.09223`.

Northpointe Inc. Measurement & treatment implications of COMPAS core scales. Technical report, Northpointe Inc., 2009. URL `https://www.michigan.gov/documents/corrections/Timothy_Brenne_Ph.D.__Meaning_and_Treatment_Implications_of_COMPA_Core_Scales_297495_7.pdf`. Retrieved May 20, 2025.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Why should i trust you?: Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144, 2016.

Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1(5):206–215, 2019.

Cynthia Rudin, Caroline Wang, and Beau Coker. The Age of Secrecy and Unfairness in Recidivism Prediction. *Harvard Data Science Review*, 2(1), mar 31 2020. https://hdsr.mitpress.mit.edu/pub/7z10o269.

Sarah Tan, Rich Caruana, Giles Hooker, and Yin Lou. Distill-and-compare: Auditing black-box models using transparent model distillation. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, pages 303–310, 2018.

Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 1996.

Mario V Wüthrich and Michael Merz. From generalized linear models to neural networks, and back. *SSRN Electronic Journal*, 2019.

Tong Zhang and Bin Yu. Boosting with early stopping: Convergence and consistency. *Annals of Statistics*, 33(4):1538–1579, 2005.

Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the royal statistical society: series B (statistical methodology)*, 67(2):301–320, 2005.

A Implementation Details

We extended LightGBM v4.5.0.99 with custom C++ implementations of our L0-like regularization mechanisms. The GBM-to-factor-table conversion pipeline was implemented in Rust using the Polars dataframe library. Multi-objective hyperparameter tuning was performed using Optuna. All experiments were conducted on a MacBook Air M1 chip with 16GB RAM.

B French Motor Supplement

B.1 Intereptable Model Details

Model intercept is -2.30193.

Table 6: Insurance Model 1: Factor table for Vehicle Gas Type and Vehicle Age interaction

VehAge (Upper Bound)	**VehGas**	**Factor**
0	Diesel	0.116
1.5	Diesel	-0.0634
2.5	Diesel	-0.0286
3.5	Diesel	-0.0444
4.5	Diesel	0.0199
5.5	Diesel	0.0278
6.5	Diesel	0.0286
7.5	Diesel	0.0245
8.5	Diesel	-0.0305
9.5	Diesel	-0.0312
10.5	Diesel	-0.0473
12.5	Diesel	-0.0458
13.5	Diesel	-0.1367
14.5	Diesel	-0.1278
15.5	Diesel	-0.1493
16.5	Diesel	-0.1045
17.5	Diesel	-0.0897
19.5	Diesel	-0.1961
20.5	Diesel	-0.2135
21.5	Diesel	-0.2596
24.5	Diesel	-0.2288
25.5	Diesel	-0.0089
27.5	Diesel	-0.1501
33.5	Diesel	-0.2602
∞	Diesel	-0.3694
0	Gasoline	1.0175
1.5	Gasoline	-0.0937
2.5	Gasoline	-0.0883
3.5	Gasoline	-0.1011
4.5	Gasoline	-0.0368
5.5	Gasoline	-0.0709
6.5	Gasoline	-0.0213
7.5	Gasoline	-0.0097
8.5	Gasoline	-0.0306
9.5	Gasoline	-0.0273
10.5	Gasoline	-0.0434
12.5	Gasoline	-0.0459
13.5	Gasoline	-0.1368
14.5	Gasoline	-0.147
15.5	Gasoline	-0.1684
16.5	Gasoline	-0.1684
17.5	Gasoline	-0.1823
19.5	Gasoline	-0.3927
20.5	Gasoline	-0.3544
21.5	Gasoline	-0.3772
24.5	Gasoline	-0.3464
25.5	Gasoline	-0.1013
27.5	Gasoline	-0.2425
33.5	Gasoline	-0.3526
∞	Gasoline	-0.4619

Table 7: Insurance Model 1: Factor table for Driver Age

DrivAge (Upper Bound)	**Factor**
23.5	0.0167
26.5	-0.2698
28.5	-0.3227
30.5	-0.3103
32.5	-0.2741
34.5	-0.1778
36.5	-0.1435
38.5	-0.086
40.5	-0.0453
42.5	0.0455
44.5	0.1233
47.5	0.1954
50.5	0.1942
51.5	0.1845
52.5	0.1814
53.5	0.1821
54.5	0.1492
58.5	0.0575
60.5	-0.0445
62.5	0.0263
64.5	0.0578
66.5	0.0283
68.5	0.0007
70.5	0.1126
∞	0.1014

Table 8: Factor table for Bonus-Malus and Vehicle Age interaction

BonusMalus (Upper Bound)	**VehAge (Upper Bound)**	**Factor**
50.5	0.0	0.3148
50.5	1.5	-0.2938
50.5	2.5	-0.2961
50.5	3.5	-0.3476
50.5	4.5	-0.313
50.5	5.5	-0.3255
50.5	6.5	-0.3189
50.5	7.5	-0.3189
50.5	8.5	-0.3615
50.5	9.5	-0.3635
50.5	10.5	-0.3538
50.5	11.5	-0.381
50.5	12.5	-0.3805
50.5	13.5	-0.4688
50.5	14.5	-0.4955
50.5	15.5	-0.519
50.5	16.5	-0.5561
50.5	17.5	-0.6288
50.5	19.5	-0.6861
50.5	20.5	-0.6926
50.5	21.5	-0.7442
50.5	24.5	-0.6875
50.5	25.5	-0.4977
50.5	27.5	-0.5821
50.5	33.5	-0.737
50.5	∞	-0.7786
51.5	0.0	0.3936
51.5	1.5	-0.2151
51.5	2.5	-0.2174
51.5	3.5	-0.3046
51.5	4.5	-0.3349
51.5	5.5	-0.354
51.5	6.5	-0.3474
51.5	7.5	-0.3474
51.5	8.5	-0.39
51.5	9.5	-0.4175
51.5	10.5	-0.4175
51.5	11.5	-0.4483
51.5	12.5	-0.4506
51.5	13.5	-0.5389
51.5	14.5	-0.5656
51.5	15.5	-0.6449
51.5	16.5	-0.6707
51.5	17.5	-0.7671
51.5	19.5	-0.8245

Continued on next page

Table 8: Factor table for Bonus-Malus and Vehicle Age interaction

BonusMalus (Upper Bound)	**VehAge (Upper Bound)**	**Factor**
51.5	20.5	-0.8309
51.5	21.5	-0.8825
51.5	24.5	-0.8259
51.5	25.5	-0.6361
51.5	27.5	-0.7204
51.5	33.5	-0.8753
51.5	∞	-0.917
54.5	0.0	0.543
54.5	1.5	-0.0187
54.5	2.5	-0.0135
54.5	3.5	-0.0863
54.5	4.5	-0.1166
54.5	5.5	-0.1357
54.5	6.5	-0.1323
54.5	7.5	-0.1323
54.5	8.5	-0.1541
54.5	9.5	-0.1816
54.5	10.5	-0.1996
54.5	11.5	-0.2399
54.5	12.5	-0.2422
54.5	13.5	-0.2827
54.5	14.5	-0.2876
54.5	15.5	-0.3873
54.5	16.5	-0.3813
54.5	17.5	-0.4777
54.5	19.5	-0.5183
54.5	20.5	-0.5247
54.5	21.5	-0.5763
54.5	24.5	-0.5197
54.5	25.5	-0.3299
54.5	27.5	-0.4142
54.5	33.5	-0.5691
54.5	∞	-0.6108
56.5	0.0	0.7577
56.5	1.5	0.2586
56.5	2.5	0.2684
56.5	3.5	0.2375
56.5	4.5	0.2722
56.5	5.5	0.2531
56.5	6.5	0.2565
56.5	7.5	0.2565
56.5	8.5	0.2347
56.5	9.5	0.2043
56.5	10.5	0.1796
56.5	11.5	0.1393
56.5	12.5	0.1369

Continued on next page

Table 8: Factor table for Bonus-Malus and Vehicle Age interaction

BonusMalus (Upper Bound)	**VehAge (Upper Bound)**	**Factor**
56.5	13.5	0.0965
56.5	14.5	0.0915
56.5	15.5	-0.0082
56.5	16.5	-0.0022
56.5	17.5	-0.0986
56.5	19.5	-0.1391
56.5	20.5	-0.1456
56.5	21.5	-0.1972
56.5	24.5	-0.1405
56.5	25.5	0.0493
56.5	27.5	-0.0351
56.5	33.5	-0.19
56.5	∞	-0.2317
57.5	0.0	0.4115
57.5	1.5	-0.1775
57.5	2.5	-0.1609
57.5	3.5	-0.1918
57.5	4.5	-0.1571
57.5	5.5	-0.1763
57.5	6.5	-0.1728
57.5	7.5	-0.1728
57.5	8.5	-0.1946
57.5	9.5	-0.225
57.5	10.5	-0.2394
57.5	11.5	-0.2702
57.5	12.5	-0.2726
57.5	13.5	-0.313
57.5	14.5	-0.3179
57.5	15.5	-0.4177
57.5	16.5	-0.4116
57.5	17.5	-0.6499
57.5	19.5	-0.8686
57.5	20.5	-0.8751
57.5	21.5	-0.9267
57.5	24.5	-0.87
57.5	25.5	-0.6802
57.5	27.5	-0.7646
57.5	33.5	-0.9195
57.5	∞	-0.9611
60.5	0.0	0.6854
60.5	1.5	0.1995
60.5	2.5	0.216
60.5	3.5	0.2179
60.5	4.5	0.2525
60.5	5.5	0.2409
60.5	6.5	0.2686

Continued on next page

Table 8: Factor table for Bonus-Malus and Vehicle Age interaction

BonusMalus (Upper Bound)	**VehAge (Upper Bound)**	**Factor**
60.5	7.5	0.2686
60.5	8.5	0.2467
60.5	9.5	0.2467
60.5	10.5	0.2323
60.5	11.5	0.2015
60.5	12.5	0.1992
60.5	13.5	0.0997
60.5	14.5	0.0947
60.5	15.5	0.0551
60.5	16.5	0.0611
60.5	17.5	0.1012
60.5	19.5	0.1054
60.5	20.5	0.1467
60.5	21.5	0.0951
60.5	24.5	0.1518
60.5	25.5	0.3416
60.5	27.5	0.2572
60.5	33.5	0.1023
60.5	∞	0.0607
62.5	0.0	0.9825
62.5	1.5	0.8514
62.5	2.5	0.868
62.5	3.5	0.8698
62.5	4.5	0.9045
62.5	5.5	0.8929
62.5	6.5	1.0372
62.5	7.5	1.0372
62.5	8.5	1.0238
62.5	9.5	1.0839
62.5	10.5	1.0695
62.5	11.5	1.157
62.5	12.5	1.1546
62.5	13.5	1.0551
62.5	14.5	0.9941
62.5	15.5	0.9247
62.5	16.5	0.9307
62.5	17.5	0.9708
62.5	19.5	0.975
62.5	20.5	1.0917
62.5	21.5	1.0401
62.5	24.5	1.0967
62.5	25.5	1.2865
62.5	27.5	1.2022
62.5	33.5	1.0473
62.5	∞	1.0056
64.5	0.0	0.515

Continued on next page

Table 8: Factor table for Bonus-Malus and Vehicle Age interaction

BonusMalus (Upper Bound)	**VehAge (Upper Bound)**	**Factor**
64.5	1.5	0.0897
64.5	2.5	0.1062
64.5	3.5	0.1081
64.5	4.5	0.0986
64.5	5.5	0.0779
64.5	6.5	0.1314
64.5	7.5	0.1351
64.5	8.5	0.1217
64.5	9.5	0.1219
64.5	10.5	0.0929
64.5	11.5	0.087
64.5	12.5	0.0847
64.5	13.5	-0.0064
64.5	14.5	-0.0674
64.5	15.5	-0.1369
64.5	16.5	-0.0914
64.5	17.5	-0.0513
64.5	19.5	-0.0471
64.5	20.5	0.0696
64.5	21.5	0.018
64.5	24.5	0.0746
64.5	25.5	0.2644
64.5	27.5	0.1801
64.5	33.5	0.0252
64.5	∞	-0.0165
68.5	0.0	0.6405
68.5	1.5	0.2052
68.5	2.5	0.2218
68.5	3.5	0.2236
68.5	4.5	0.2583
68.5	5.5	0.2375
68.5	6.5	0.2692
68.5	7.5	0.2729
68.5	8.5	0.2596
68.5	9.5	0.2678
68.5	10.5	0.2144
68.5	11.5	0.2086
68.5	12.5	0.2119
68.5	13.5	0.1208
68.5	14.5	0.0598
68.5	15.5	-0.0277
68.5	16.5	-0.0217
68.5	17.5	0.0185
68.5	19.5	0.0226
68.5	20.5	0.1393
68.5	21.5	0.0877

Continued on next page

Table 8: Factor table for Bonus-Malus and Vehicle Age interaction

BonusMalus (Upper Bound)	**VehAge (Upper Bound)**	**Factor**
68.5	24.5	0.1444
68.5	25.5	0.3342
68.5	27.5	0.2498
68.5	33.5	0.0949
68.5	∞	0.0533
72.5	0.0	0.6983
72.5	1.5	0.263
72.5	2.5	0.2796
72.5	3.5	0.2814
72.5	4.5	0.3254
72.5	5.5	0.3479
72.5	6.5	0.3796
72.5	7.5	0.3833
72.5	8.5	0.3699
72.5	9.5	0.3782
72.5	10.5	0.3248
72.5	11.5	0.319
72.5	12.5	0.3223
72.5	13.5	0.2312
72.5	14.5	0.1702
72.5	15.5	0.0827
72.5	16.5	0.0887
72.5	17.5	0.1496
72.5	19.5	0.1538
72.5	20.5	0.3176
72.5	21.5	0.266
72.5	24.5	0.3226
72.5	25.5	0.5124
72.5	27.5	0.428
72.5	33.5	0.2732
72.5	∞	0.2315
76.5	0.0	0.6862
76.5	1.5	0.263
76.5	2.5	0.2796
76.5	3.5	0.2814
76.5	4.5	0.3254
76.5	5.5	0.3479
76.5	6.5	0.3796
76.5	7.5	0.3833
76.5	8.5	0.352
76.5	9.5	0.3782
76.5	10.5	0.3248
76.5	11.5	0.319
76.5	12.5	0.3223
76.5	13.5	0.2312
76.5	14.5	0.1702

Continued on next page

Table 8: Factor table for Bonus-Malus and Vehicle Age interaction

BonusMalus (Upper Bound)	**VehAge (Upper Bound)**	**Factor**
76.5	15.5	0.0827
76.5	16.5	0.0887
76.5	17.5	0.1496
76.5	19.5	0.1538
76.5	20.5	0.3176
76.5	21.5	0.266
76.5	24.5	0.3226
76.5	25.5	0.5124
76.5	27.5	0.428
76.5	33.5	0.2732
76.5	∞	0.2315
80.5	0.0	0.813
80.5	1.5	0.4059
80.5	2.5	0.4225
80.5	3.5	0.4243
80.5	4.5	0.4683
80.5	5.5	0.4908
80.5	6.5	0.5225
80.5	7.5	0.5263
80.5	8.5	0.4949
80.5	9.5	0.5211
80.5	10.5	0.4678
80.5	11.5	0.4619
80.5	12.5	0.4652
80.5	13.5	0.3741
80.5	14.5	0.3131
80.5	15.5	0.2256
80.5	16.5	0.2317
80.5	17.5	0.2925
80.5	19.5	0.2967
80.5	20.5	0.5811
80.5	21.5	0.5295
80.5	24.5	0.5861
80.5	25.5	0.7759
80.5	27.5	0.6916
80.5	33.5	0.5367
80.5	∞	0.495
85.5	0.0	0.8086
85.5	1.5	0.4016
85.5	2.5	0.4182
85.5	3.5	0.42
85.5	4.5	0.464
85.5	5.5	0.4865
85.5	6.5	0.5182
85.5	7.5	0.5219
85.5	8.5	0.4906

Continued on next page

Table 8: Factor table for Bonus-Malus and Vehicle Age interaction

BonusMalus (Upper Bound)	**VehAge (Upper Bound)**	**Factor**
85.5	9.5	0.5168
85.5	10.5	0.4634
85.5	11.5	0.4576
85.5	12.5	0.4609
85.5	13.5	0.3698
85.5	14.5	0.3088
85.5	15.5	0.2213
85.5	16.5	0.2323
85.5	17.5	0.2932
85.5	19.5	0.2973
85.5	20.5	0.4611
85.5	21.5	0.4095
85.5	24.5	0.4661
85.5	25.5	0.656
85.5	27.5	0.5716
85.5	33.5	0.4167
85.5	∞	0.375
90.5	0.0	0.7709
90.5	1.5	0.3638
90.5	2.5	0.3934
90.5	3.5	0.3952
90.5	4.5	0.4392
90.5	5.5	0.4252
90.5	6.5	0.4569
90.5	7.5	0.4606
90.5	8.5	0.4293
90.5	9.5	0.4516
90.5	10.5	0.3982
90.5	11.5	0.3923
90.5	12.5	0.3956
90.5	13.5	0.318
90.5	14.5	0.257
90.5	15.5	0.1696
90.5	16.5	0.1806
90.5	17.5	0.1489
90.5	19.5	0.153
90.5	20.5	0.3168
90.5	21.5	0.2652
90.5	24.5	0.3218
90.5	25.5	0.5116
90.5	27.5	0.4273
90.5	33.5	0.2724
90.5	∞	0.2307
95.5	0.0	0.9771
95.5	1.5	0.5701
95.5	2.5	0.5996

Continued on next page

Table 8: Factor table for Bonus-Malus and Vehicle Age interaction

BonusMalus (Upper Bound)	**VehAge (Upper Bound)**	**Factor**
95.5	3.5	0.6014
95.5	4.5	0.6454
95.5	5.5	0.6523
95.5	6.5	0.684
95.5	7.5	0.6877
95.5	8.5	0.6563
95.5	9.5	0.6786
95.5	10.5	0.613
95.5	11.5	0.6071
95.5	12.5	0.6104
95.5	13.5	0.5328
95.5	14.5	0.4718
95.5	15.5	0.3843
95.5	16.5	0.3953
95.5	17.5	0.5002
95.5	19.5	0.5044
95.5	20.5	0.6682
95.5	21.5	0.6166
95.5	24.5	0.6732
95.5	25.5	0.863
95.5	27.5	0.7786
95.5	33.5	0.6237
95.5	∞	0.5821
100.5	0.0	1.0308
100.5	1.5	0.808
100.5	2.5	0.946
100.5	3.5	0.9355
100.5	4.5	0.9795
100.5	5.5	1.1205
100.5	6.5	1.1522
100.5	7.5	1.1712
100.5	8.5	1.1562
100.5	9.5	1.1872
100.5	10.5	1.1501
100.5	11.5	1.2215
100.5	12.5	1.2401
100.5	13.5	1.1324
100.5	14.5	1.1196
100.5	15.5	1.0322
100.5	16.5	1.0838
100.5	17.5	1.1886
100.5	19.5	1.1928
100.5	20.5	1.3566
100.5	21.5	1.4778
100.5	24.5	1.5344
100.5	25.5	1.7242

Continued on next page

Table 8: Factor table for Bonus-Malus and Vehicle Age interaction

BonusMalus (Upper Bound)	**VehAge (Upper Bound)**	**Factor**
100.5	27.5	1.6399
100.5	33.5	1.485
100.5	∞	1.4433
106.5	0.0	1.1353
106.5	1.5	0.9125
106.5	2.5	1.0504
106.5	3.5	1.04
106.5	4.5	1.084
106.5	5.5	1.225
106.5	6.5	1.2567
106.5	7.5	1.2756
106.5	8.5	1.2607
106.5	9.5	1.3239
106.5	10.5	1.2868
106.5	11.5	1.3999
106.5	12.5	1.4185
106.5	13.5	1.3108
106.5	14.5	1.298
106.5	15.5	1.2105
106.5	16.5	1.2621
106.5	17.5	1.4384
106.5	19.5	1.4425
106.5	20.5	1.6063
106.5	21.5	1.7275
106.5	24.5	1.7842
106.5	25.5	1.974
106.5	27.5	1.8896
106.5	33.5	1.7347
106.5	∞	1.693
112.5	0.0	1.2688
112.5	1.5	1.0459
112.5	2.5	1.1839
112.5	3.5	1.1734
112.5	4.5	1.2024
112.5	5.5	1.3585
112.5	6.5	1.3902
112.5	7.5	1.4091
112.5	8.5	1.3941
112.5	9.5	1.4619
112.5	10.5	1.4248
112.5	11.5	1.5379
112.5	12.5	1.5565
112.5	13.5	1.4488
112.5	14.5	1.436
112.5	15.5	1.344
112.5	16.5	1.3956

Continued on next page

Table 8: Factor table for Bonus-Malus and Vehicle Age interaction

BonusMalus (Upper Bound)	**VehAge (Upper Bound)**	**Factor**
112.5	17.5	1.5718
112.5	19.5	1.576
112.5	20.5	1.7398
112.5	21.5	1.8892
112.5	24.5	1.9458
112.5	25.5	2.1356
112.5	27.5	2.0513
112.5	33.5	1.8964
112.5	∞	1.8547
118.5	0.0	1.3392
118.5	1.5	1.1164
118.5	2.5	1.2544
118.5	3.5	1.2439
118.5	4.5	1.2729
118.5	5.5	1.4289
118.5	6.5	1.4606
118.5	7.5	1.4795
118.5	8.5	1.4646
118.5	9.5	1.539
118.5	10.5	1.5019
118.5	11.5	1.615
118.5	12.5	1.6336
118.5	13.5	1.5259
118.5	14.5	1.5131
118.5	15.5	1.4211
118.5	16.5	1.4727
118.5	17.5	1.649
118.5	19.5	1.6531
118.5	20.5	1.8169
118.5	21.5	1.9663
118.5	24.5	2.0229
118.5	25.5	2.2128
118.5	27.5	2.1284
118.5	33.5	1.9735
118.5	∞	1.9318
125.5	0.0	1.546
125.5	1.5	1.3231
125.5	2.5	1.4611
125.5	3.5	1.4506
125.5	4.5	1.4796
125.5	5.5	1.6357
125.5	6.5	1.6674
125.5	7.5	1.6863
125.5	8.5	1.6713
125.5	9.5	1.7458
125.5	10.5	1.7087

Continued on next page

Table 8: Factor table for Bonus-Malus and Vehicle Age interaction

BonusMalus (Upper Bound)	**VehAge (Upper Bound)**	**Factor**
125.5	11.5	1.8218
125.5	12.5	1.8404
125.5	13.5	1.7327
125.5	14.5	1.7199
125.5	15.5	1.6279
125.5	16.5	1.6795
125.5	17.5	1.8557
125.5	19.5	1.8599
125.5	20.5	2.0237
125.5	21.5	2.1731
125.5	24.5	2.2297
125.5	25.5	2.4195
125.5	27.5	2.3352
125.5	33.5	2.1803
125.5	∞	2.1386
135.5	0.0	1.363
135.5	1.5	1.1402
135.5	2.5	1.2781
135.5	3.5	1.2677
135.5	4.5	1.2967
135.5	5.5	1.4527
135.5	6.5	1.4844
135.5	7.5	1.5033
135.5	8.5	1.4884
135.5	9.5	1.5628
135.5	10.5	1.5257
135.5	11.5	1.6388
135.5	12.5	1.6574
135.5	13.5	1.5497
135.5	14.5	1.5369
135.5	15.5	1.4449
135.5	16.5	1.4965
135.5	17.5	1.6728
135.5	19.5	1.6769
135.5	20.5	1.8407
135.5	21.5	1.9901
135.5	24.5	2.0467
135.5	25.5	2.2366
135.5	27.5	2.1522
135.5	33.5	1.9973
135.5	∞	1.9556
140.5	0.0	1.4927
140.5	1.5	1.2698
140.5	2.5	1.4078
140.5	3.5	1.3973
140.5	4.5	1.4263

Continued on next page

Table 8: Factor table for Bonus-Malus and Vehicle Age interaction

BonusMalus (Upper Bound)	**VehAge (Upper Bound)**	**Factor**
140.5	5.5	1.5823
140.5	6.5	1.6141
140.5	7.5	1.633
140.5	8.5	1.618
140.5	9.5	1.6925
140.5	10.5	1.6554
140.5	11.5	1.7685
140.5	12.5	1.7871
140.5	13.5	1.6794
140.5	14.5	1.6666
140.5	15.5	1.5746
140.5	16.5	1.6261
140.5	17.5	1.8024
140.5	19.5	1.8066
140.5	20.5	1.9704
140.5	21.5	2.1198
140.5	24.5	2.1764
140.5	25.5	2.3662
140.5	27.5	2.2819
140.5	33.5	2.127
140.5	∞	2.0853
147.5	0.0	1.595
147.5	1.5	1.3722
147.5	2.5	1.5102
147.5	3.5	1.4997
147.5	4.5	1.5287
147.5	5.5	1.6847
147.5	6.5	1.7164
147.5	7.5	1.7354
147.5	8.5	1.7204
147.5	9.5	1.7949
147.5	10.5	1.7577
147.5	11.5	1.8709
147.5	12.5	1.8894
147.5	13.5	1.7818
147.5	14.5	1.7689
147.5	15.5	1.6769
147.5	16.5	1.7285
147.5	17.5	1.9048
147.5	19.5	1.909
147.5	20.5	2.0727
147.5	21.5	2.2221
147.5	24.5	2.2788
147.5	25.5	2.4686
147.5	27.5	2.3842
147.5	33.5	2.2293

Continued on next page

Table 8: Factor table for Bonus-Malus and Vehicle Age interaction

BonusMalus (Upper Bound)	**VehAge (Upper Bound)**	**Factor**
147.5	∞	2.1877
152	0.0	1.6307
152	1.5	1.4079
152	2.5	1.5459
152	3.5	1.5354
152	4.5	1.5644
152	5.5	1.7204
152	6.5	1.7522
152	7.5	1.7711
152	8.5	1.7561
152	9.5	1.7949
152	10.5	1.7577
152	11.5	1.8709
152	12.5	1.8894
152	13.5	1.7818
152	14.5	1.7689
152	15.5	1.6769
152	16.5	1.7285
152	17.5	1.9048
152	19.5	1.909
152	20.5	2.0727
152	21.5	2.2221
152	24.5	2.2788
152	25.5	2.4686
152	27.5	2.3842
152	33.5	2.2293
152	∞	2.1877
157	0.0	1.7318
157	1.5	1.509
157	2.5	1.647
157	3.5	1.6365
157	4.5	1.6655
157	5.5	1.7958
157	6.5	1.8276
157	7.5	1.7711
157	8.5	1.7561
157	9.5	1.7949
157	10.5	1.7577
157	11.5	1.8709
157	12.5	1.8894
157	13.5	1.7818
157	14.5	1.7689
157	15.5	1.6769
157	16.5	1.7285
157	17.5	1.9048
157	19.5	1.909

Continued on next page

Table 8: Factor table for Bonus-Malus and Vehicle Age interaction

BonusMalus (Upper Bound)	**VehAge (Upper Bound)**	**Factor**
157	20.5	2.0727
157	21.5	2.2221
157	24.5	2.2788
157	25.5	2.4686
157	27.5	2.3842
157	33.5	2.2293
157	∞	2.1877
∞	0.0	1.9696
∞	1.5	1.7468
∞	2.5	1.8773
∞	3.5	1.8668
∞	4.5	1.8958
∞	5.5	2.0261
∞	6.5	2.0579
∞	7.5	2.0014
∞	8.5	1.9864
∞	9.5	2.0252
∞	10.5	1.988
∞	11.5	2.1012
∞	12.5	2.1197
∞	13.5	2.0121
∞	14.5	1.9992
∞	15.5	1.9072
∞	16.5	1.9588
∞	17.5	2.1351
∞	19.5	2.1393
∞	20.5	2.303
∞	21.5	2.4524
∞	24.5	2.5091
∞	25.5	2.6989
∞	27.5	2.6145
∞	33.5	2.4596
∞	∞	2.418

Due to the size and number of the tables, the "best CV" model tables are omitted from this paper but are available in a supplemental file.