Interpretable Machine Learning
Project 2

Filip Niewczas & Mohammadhadi Shahhosseini

June 2025

Contents

1 Introduction

This project continues the work from Project 1, where the goal was to predict the annual insurance claim cost using features available at the beginning of the contract. In Project 1, we focused only on accuracy and selected an XGBoost model as the final model.

In this project (Project 2), the goal remains the same, but we also focus on interpretability and fairness. First, we train a simpler and more transparent version of XGBoost. Then, we explain how this model works using different tools such as feature importance, SHAP values, and dependence plots.

Another key part of this project is to check how much the model depends on `Date_birth`, which is used in the model through the variable `Age`. We try to remove both direct and indirect effects of `Age` using a debiasing method based on residuals.

Finally, we compare three models: the original model from Project 1, the interpretable model from this project, and the debiased version. We check both the accuracy and how much each model relies on `Age`.

2 Data Description

The dataset employed in this study is the same as the dataset we used in the previous project. It originates from a motor vehicle insurance database, comprising detailed records of insurance policies. Each row in the dataset represents a unique contract associated with a specific policyholder. The dataset encompasses temporal information about contract renewals, policyholder demographics, vehicle characteristics, and financial details related to premiums and claims.

This comprehensive dataset serves as the backbone for our predictive modeling and exploratory analyses. A detailed description of each variable used in the study is provided in Table 1.

The original dataset includes 105,555 observations and 30 features, encompassing:

- Dates (e.g. contract start/lapse, birth, license issue),
- Categorical features (e.g. fuel type, area, risk level),
- Numeric variables (e.g. vehicle power, value, cylinder capacity),

Table 1: Description of the variables used in the dataset

Variable	Description
ID	Internal identification number assigned to each annual contract formalized by an insured.
Date_last_renewal	Date of last contract renewal (DD/MM/YYYY).
Date_next_renewal	Date of the next contract renewal (DD/MM/YYYY).
Distribution_channel	Classifies the channel through which the policy was contracted.
Date_birth	Date of birth of the insured declared in the policy (DD/MM/YYYY).
Date_driving_licence	Date of issuance of the insured person's driver's license (DD/MM/YYYY).
Seniority	Total number of years that the insured has been associated with the insurance entity.
Policies_in_force	Total number of policies held by the insured in the insurance entity during the reference period.
Max_policies	Maximum number of policies that the insured has ever had in force with the insurance entity.
Max_products	Maximum number of products that the insured has simultaneously held at any given point in time.
Lapse	Number of policies that the customer has cancelled or has been cancelled for nonpayment in the current year of maturity.
Date_lapse	Lapse date. Date of contract termination (DD/MM/YYYY).
Payment	Last payment method of the reference policy.
Premium	Net premium amount associated with the policy during the current year.
Cost_claims_year	Total cost of claims incurred for the insurance policy during the current year.
N_claims_year	Total number of claims incurred for the insurance policy during the current year.
N_claims_history	Total number of claims filed throughout the entire duration of the insurance policy.
R_Claims_history	Ratio of the number of claims filed for the specific policy to the total duration of the policy in force.
Type_risk	Type of risk associated with the policy.
Area	Dichotomous variable indicates the area.
Second_driver	1 if there are multiple regular drivers declared, or 0 if only one driver is declared.
Year_matriculation	Year of registration of the vehicle (YYYY).
Power	Vehicle power measured in horsepower.
Cylinder_capacity	Cylinder capacity of the vehicle.
Value_vehicle	Market value of the vehicle on 31/12/2019.
N_doors	Number of vehicle doors.
Type_fuel	Specific kind of energy source used to power a vehicle. Petrol (P) or Diesel (D).
Length	Length, in meters, of the vehicle.
Weight	Weight, in kilograms, of the vehicle.

3 Project 1 Recap

The first phase of this research focused on predicting the annual insurance claim cost (`Cost_claims_year`) using structured data available at contract initiation. Two modeling strategies were explored:

- A three-stage actuarial approach, estimating claim probability, frequency, and severity separately;
- Traditional regression models, including Linear Regression, Random Forest, and XGBoost, trained to predict the total cost directly.

A thorough preprocessing pipeline was implemented to handle missing values, engineer features (e.g., driver age, driving experience, car age), and aggregate contract-level data to the policyholder level. Feature selection was then performed based on model-derived importance scores to reduce dimensionality without sacrificing performance.

After extensive benchmarking and tuning, the best-performing model was an XGBoost regressor trained on the top 17 selected features. It achieved a Root Mean Squared Error (RMSE) of **1088.85**, outperforming both the baseline mean prediction model (RMSE = 1580.94) and the interpretable 3-stage model (RMSE = 1837.68). This model served as the final model for Project 1 and establishes the performance benchmark for the present study.

Table 2 summarizes the predictive accuracy of the models considered in Project 1.

Table 2: Model performance comparison from Project 1

Model	**RMSE**	**Remarks**
Baseline (Mean prediction)	1580.94	No learning, naive benchmark
Three-stage Freq–Sev–Prob model	1837.68	Decomposed, interpretable
XGBoost (Final model)	1088.85	Tuned, best performance

4 Data Preprocessing

To ensure consistency and reliability in modeling, a structured data preprocessing pipeline was implemented. The primary objectives were to clean and transform raw inputs, engineer informative features, and aggregate contract-level data into a policyholder-level view suitable for prediction.

4.1 Data Cleaning and Feature Engineering.

Initially, all relevant date fields were parsed using consistent formatting, and categorical variables were explicitly cast as factors. Several transformations were applied to enhance the predictive power of the dataset:

- *Age Calculation:* The policyholder's age was computed based on the difference between the date of birth and the contract start date.
- *Lapse Handling:* Missing values in the `Lapse` variable were imputed as zero. The lapse duration (in days) was calculated and combined with a binary indicator (`Has_lapsed`) to capture lapse behavior.
- *Car Age Binning:* Vehicle age was categorized into four groups: {0–10}, {10–20}, {20–30}, and {30+} years, to capture potential nonlinear effects in vehicle depreciation and risk.
- *Fuel Type Encoding:* Missing values in `Type_fuel` were imputed using the mode within each `Type_risk` group. Two binary indicator variables (`Fuel_Petrol` and `Fuel_Diesel`) were then derived.
- *Distribution Channel:* Missing values in `Distribution_channel` were replaced with a dedicated category labeled "Missing" and cast as a factor.
- *Length Imputation:* Missing values in vehicle length were imputed using the group-wise mean within each `Type_risk` category.

4.1.1 Contract Duration Adjustment.

For each contract, the actual duration was derived as the minimum between the expected renewal date and the potential lapse date, expressed in years. This duration was later used for weighted aggregation.

4.1.2 Aggregation to Policyholder Level.

Since multiple contracts could be associated with a single policyholder (`ID`), the dataset was aggregated accordingly. Key features were summarized using appropriate statistics, including:

- *Cost_claims_year:* Weighted average of annual claim costs, using contract duration as the weight.
- *Categorical attributes:* Aggregated using the most frequent value (*mode*).
- *Numerical features:* Aggregated using the mean, maximum, or last recorded value, depending on the context.

4.2 Final Data Preparation.

After aggregation, the resulting dataset (`df_model`) included one row per policyholder, with clean and engineered features ready for modeling. Columns of type `difftime` were converted to numeric values, and the target variable `Cost_claims_year` was explicitly cast as numeric for compatibility with regression algorithms.

This structured preprocessing ensured a consistent data foundation for all subsequent modeling efforts, reducing noise and enhancing interpretability through carefully crafted and aggregated features.

5 Base Model (Project 2 - Transparent Model)

5.1 Model Choice

In this phase of the project, we aimed to develop a transparent predictive model capable of estimating the annual insurance claim cost while offering interpretability and robust performance. Based on the success of XGBoost in the previous phase and its flexibility in handling mixed-type data and complex interactions, we selected a tree-based gradient boosting model as our base learner. To promote transparency, we explicitly restricted model complexity through shallow tree depth.

5.1.1 Data Transformation and Task Definition.

Prior to modeling, the target variable `Cost_claims_year` was transformed using the natural logarithm of $(1 + x)$ to reduce the impact of extreme values and improve numerical stability. A regression task was then defined using the `mlr3` framework, with an 80/20 train-test split applied to ensure robust model evaluation.

5.1.2 Preprocessing Pipeline.

A modular preprocessing pipeline was constructed using `mlr3pipelines`. This pipeline performed:

- Mean imputation for missing values in numeric features;
- Mode imputation for categorical features;
- One-hot encoding of factor variables to facilitate compatibility with tree-based learners.

5.1.3 Model Configuration.

An `XGBoost` regressor was instantiated with a controlled configuration to maintain interpretability. The tree depth was limited to four levels, and other parameters such as learning rate (`eta`), subsampling ratio, and number of boosting rounds were carefully chosen based on prior tuning results. The final selected hyperparameters were:

- `eta` = 0.104
- `max_depth` = 4
- `nrounds` = 117
- `subsample` = 0.73
- `colsample_bytree` = 0.98

5.1.4 Hyperparameter Tuning.

These hyperparameters were obtained through random search optimization. A search space was defined over the learning rate, tree depth, and number of boosting rounds. The search was conducted using three-fold cross-validation, with 30 evaluations guided by the RMSE metric. The tuning framework leveraged the `mlr3tuning` package and allowed efficient exploration of model configurations while avoiding overfitting.

In summary, a simplified yet effective version of XGBoost was adopted as the base model for this stage, balancing predictive power with interpretability. The next sections present its performance, interpretability analysis, and a comparison to the baseline.

5.2 Performance Evaluation

To assess the effectiveness of the interpretable XGBoost model developed in this study, we carried out a comprehensive evaluation using both quantitative metrics and diagnostic visualizations. The primary performance metric was the Root Mean Squared Error (RMSE), calculated on the test set after back-transforming the predictions from the log scale. This metric allows for direct comparison with the models evaluated in Project 1. Table 3 compares the RMSE of the models in this project with the baseline and the model of project 1.

Table 3: Comparison of RMSE across models

Model	RMSE	Comment
XGBoost	1088.85	From Project 1
XGBoost	919.88	This Project (minor changes in preprocessing)
Baseline	1580.94	Simple average, no learning

In addition to RMSE, the model's performance was benchmarked against a naive baseline that simply predicts the mean claim cost for all observations. This comparison highlights the added value of the learning-based approach.

To further understand the model's learning dynamics and generalization capability, a series of diagnostic tools were used, including:

- **Learning Curve:** To evaluate how model performance evolves with increasing training data size and to detect potential underfitting or overfitting.
- **Residual Analysis:** To examine the distribution and magnitude of prediction errors across the training and test sets, with visual indicators such as residual plots and Q–Q plots.

- **Prediction Error Plot:** To assess the agreement between predicted and actual values and estimate the coefficient of determination (R^2), which quantifies how much variance in the target variable is captured by the model.

These evaluations provide a well-rounded understanding of both the predictive accuracy and reliability of the model, and they also form the basis for subsequent discussions on interpretability and fairness.

5.2.1 Learning Curve Analysis

To examine how the model's performance scales with increasing training data, a learning curve was generated using progressive subsets of the training set. At each subset size, the XGBoost model was trained and evaluated ten times to capture variability, and the Root Mean Squared Error (RMSE) was computed on both the training and validation splits.

As shown in Figure 1, the training RMSE (blue curve) steadily increases as more data is used, while the validation RMSE (orange curve) decreases and stabilizes. The relatively small gap between training and validation errors across all training sizes suggests that the model generalizes well and is not overfitting.

Moreover, the early convergence of the validation curve indicates that the model reaches its optimal capacity with a modest amount of data, which reflects the regularized and shallow architecture of the interpretable XGBoost learner. The inclusion of confidence ribbons further confirms the stability of the model's performance across multiple runs.

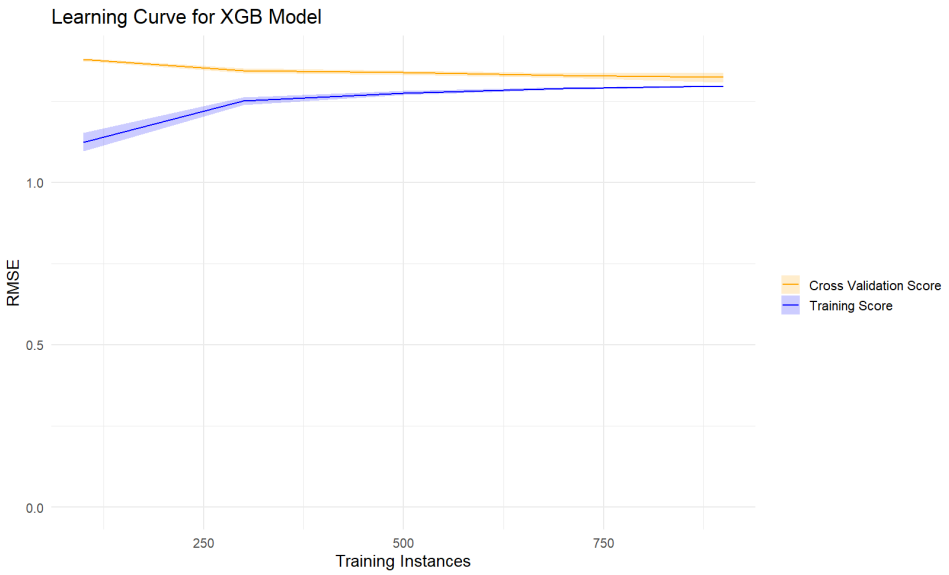

Figure 1: Learning curve of the interpretable XGBoost model. RMSE values are plotted for both training and cross-validation sets across increasing training sample sizes.

5.2.2 Residual Analysis

To evaluate the calibration and distribution of prediction errors, a residual plot and a Q–Q plot were generated for both the training and test sets. As shown in Figure 2, the residuals are plotted against predicted values, with separate color coding for the train (blue) and test (green) sets.

The residual plot reveals that prediction errors are heteroscedastic: residuals tend to increase with larger predicted values, indicating that variance in errors is not uniform across the output range. Nonetheless, the concentration of points around the zero line suggests that the model is

reasonably unbiased. The R^2 scores for the training and test sets, reported as annotations on the plot, further confirm consistent predictive performance without overfitting.

The accompanying Q–Q plot (right panel) compares the empirical distribution of residuals with a theoretical normal distribution. The strong deviation from the diagonal line at the tails, particularly in the test set, indicates that the residuals are not normally distributed. This is expected in insurance claim data, where rare but extreme claim costs generate heavy-tailed errors.

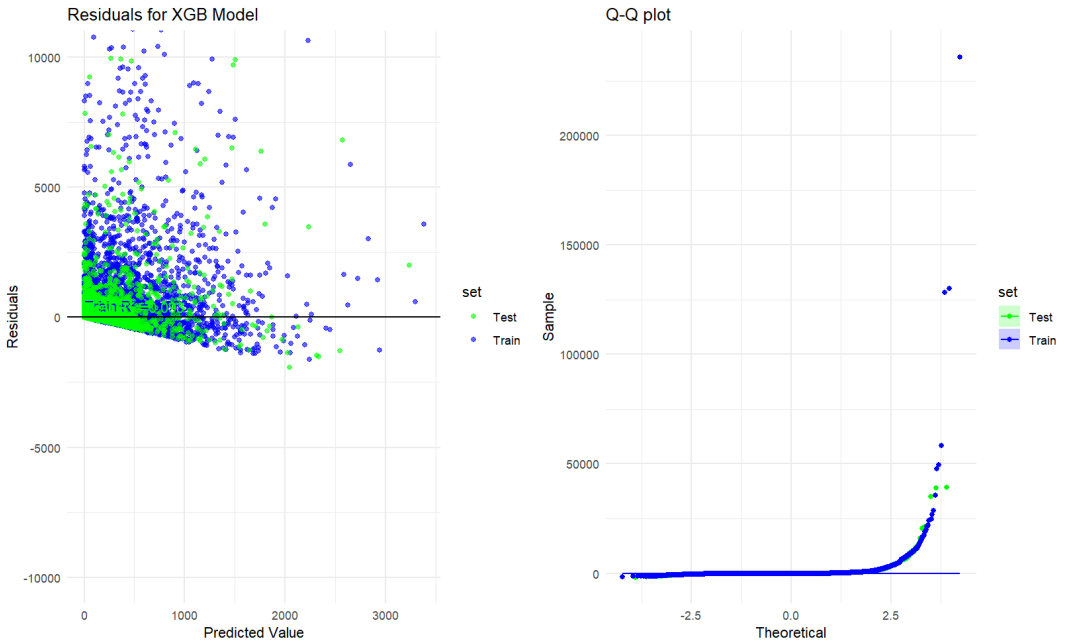

Figure 2: Residual analysis of the interpretable XGBoost model. Left: residuals versus predicted values; Right: Q–Q plot of residuals for train and test sets.

5.2.3 Prediction Error Plot

Figure 3 illustrates the relationship between the true claim values (y) and the predicted values ($\hat{y}$) produced by the XGBoost model. The diagonal line (gray, dashed) represents the ideal case where predictions perfectly match the ground truth, while the black dashed line depicts the fitted linear regression line through the predictions.

The majority of the predictions are clustered near the lower left corner, which aligns with the data distribution where most claim costs are low. However, as the true cost increases, the model systematically underestimates the claim values, reflected in the flattening of the regression line relative to the diagonal. This indicates a challenge in capturing the upper tail of the distribution — a common difficulty in insurance claim modeling due to the rare but extreme nature of high-cost claims.

The model's R^2 value, reported as 0.081, quantifies the proportion of variance in the target variable explained by the model. While this value may appear low, it is not uncommon in real-world claim datasets with high skewness and noise. This result reinforces the importance of complementing predictive accuracy with interpretability and robustness in model design.

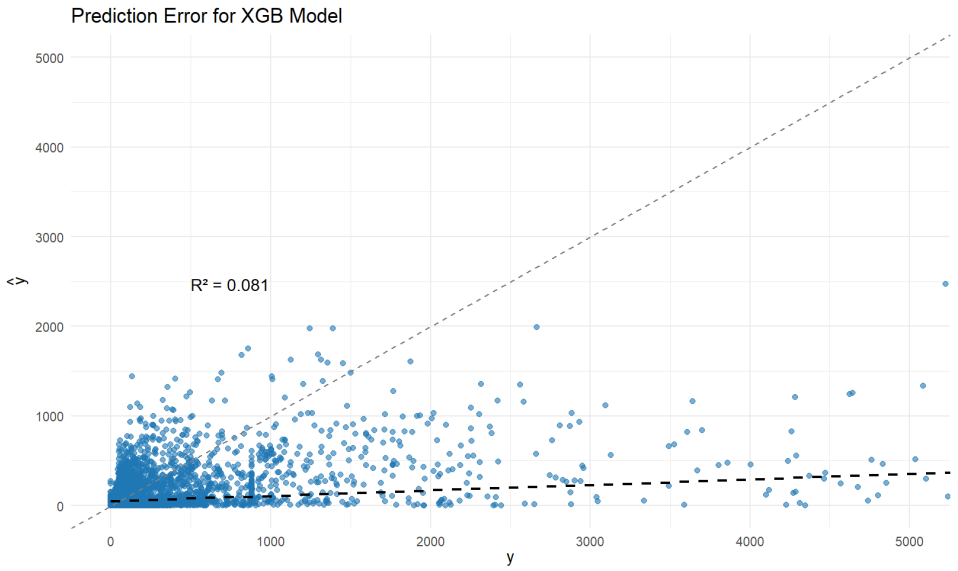

Figure 3: Prediction error plot showing the relationship between actual and predicted values. The diagonal line indicates perfect predictions, while the dashed line represents a linear fit to the predictions.

5.3 Global Explanation

To ensure transparency and interpretability in our modeling process, we conducted a comprehensive global explanation of the XGBoost model using multiple complementary tools. These methods provide insights into how the model behaves on average across the dataset, identify key features driving predictions, and explore the marginal effects of individual predictors.

- **Model-level Feature Importance:** We first used the `DALEX` package to compute permutation-based feature importance scores via the `model_parts()` function. This technique quantifies how much each feature contributes to the overall prediction accuracy, offering a reliable global perspective on feature influence.
- **SHAP Values (Summary):** SHAP (SHapley Additive exPlanations) values were computed to decompose each individual prediction into contributions from each feature. By aggregating SHAP values across the dataset, we can derive a consistent and theoretically grounded view of global feature impact.
- **Partial Dependence and ICE Plots:** To understand how model predictions respond to changes in specific features, we employed both Partial Dependence Plots (PDPs) and Individual Conditional Expectation (ICE) plots using the `iml` and `DALEX` packages. These plots illustrate the marginal effect of a feature while accounting for the distribution of other variables.

Together, these interpretability techniques offer a robust and multidimensional understanding of the XGBoost model's internal logic. In the following subsections, we provide detailed interpretations of each explanatory plot and highlight key findings relevant to the insurance claim cost prediction task.

5.3.1 Permutation Feature Importance

To quantify the global contribution of each feature to the model's predictive accuracy, we employed permutation-based feature importance using the `DALEX` package. This method measures

the increase in prediction error (RMSE) when the values of a specific feature are randomly shuffled, thereby breaking its relationship with the response variable. The larger the increase in error, the more important the feature is deemed to be.

As shown in Figure 4, the most influential variable by a significant margin is `R_Claims_history`, which captures the ratio of historical claims relative to policy duration. This feature alone leads to the largest degradation in model performance when permuted. It is followed by `N_claims_year` (number of claims filed in the current year) and `Contract_duration`, both of which are intuitively important indicators of claim activity and exposure.

Other relevant variables include `Seniority`, `Value_vehicle`, and `Policies_in_force`, each of which contributes incrementally to the model's understanding of policyholder behavior or vehicle risk. Interestingly, variables like `Fuel_type`, `Power`, and `Second_driver` show only marginal importance, suggesting a relatively minor role in driving claim costs within this dataset.

This analysis provides a stable and model-agnostic assessment of global feature relevance, supporting trust and transparency in the deployed model.

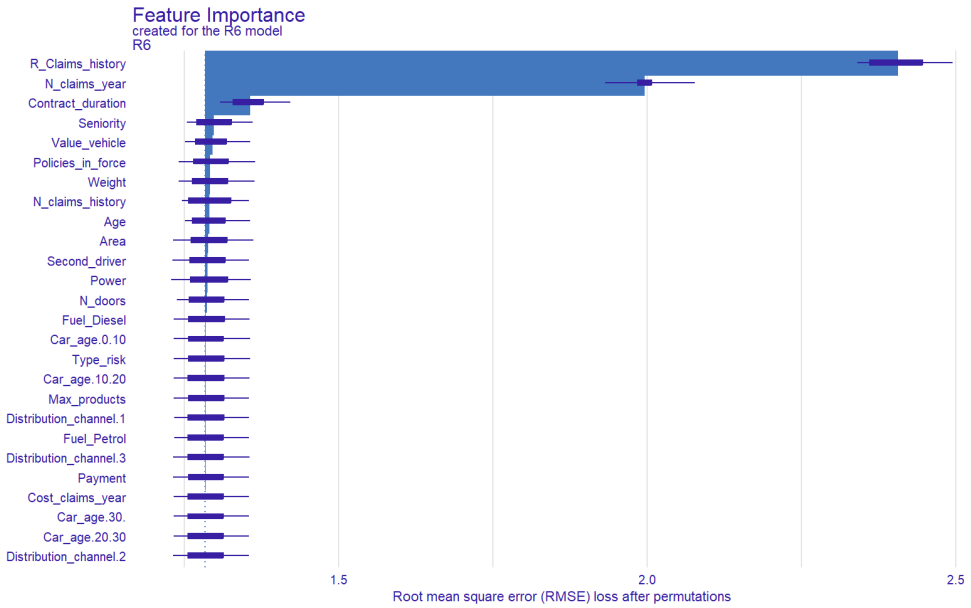

Figure 4: Permutation-based feature importance. Bars represent the increase in RMSE after randomly permuting each variable.

5.3.2 SHAP Value Analysis

To gain a theoretically grounded and locally faithful explanation of the model's predictions, we employed SHAP (SHapley Additive exPlanations) values. SHAP decomposes each model prediction into the sum of individual feature contributions based on Shapley values from cooperative game theory. For a given observation i, the prediction $\hat{y}_i$ can be expressed as:

$$\hat{y}_i = \phi_0 + \sum_{j=1}^{p} \phi_j^{(i)}$$

where ϕ_0 is the mean model prediction over the training data, and $\phi_j^{(i)}$ represents the marginal contribution of feature j to the prediction for instance i, averaged over all possible feature coalitions.

Figure 5 shows the aggregated SHAP summary plot for all instances in the dataset. The horizontal axis represents the average SHAP contribution of each feature to the prediction, and the color bands encode the direction of the effect across different feature values.

The variable `R_Claims_history` dominates the overall model behavior with the largest negative SHAP value when equal to zero. This reflects a strong decrease in the predicted claim cost when the policyholder has no history of claims relative to policy duration. A similar pattern is observed for `N_claims_year` and `N_claims_history`, where zero values result in negative contributions ($\phi_j^{(i)} < 0$), consistent with lower risk.

On the other hand, variables such as `Contract_duration` and `Value_vehicle` show positive SHAP values, indicating their contributions increase the predicted cost — likely due to longer exposure and higher insured value, respectively. Intermediate-level features like `Policies_in_force` and `Age` display lower but still non-zero contributions, suggesting they play a secondary role in shaping the model output.

Overall, SHAP values provide a reliable, additive decomposition of predictions that complements global importance rankings while preserving local interpretability. This property is especially useful in sensitive domains such as insurance, where understanding both "how much" and "in what direction" each feature affects a specific prediction is crucial.

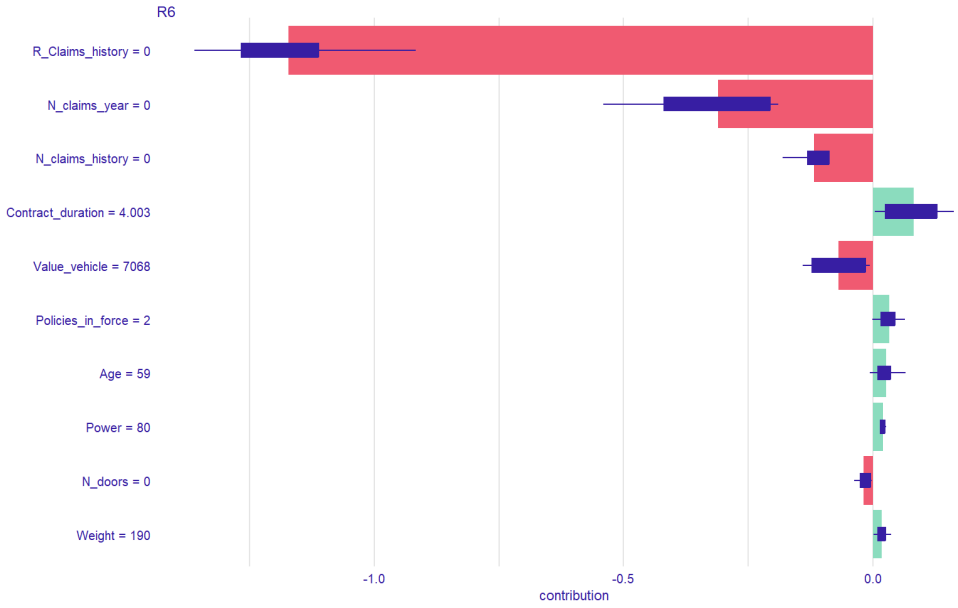

Figure 5: SHAP summary plot showing average feature contributions to model predictions. Red bars indicate negative contributions, while green bars indicate positive contributions.

5.3.3 Partial Dependence Profiles

To analyze the marginal influence of selected features on the model's predictions, we computed Partial Dependence Profiles (PDPs) using the `model_profile()` function from the `DALEX` package. PDPs show how the expected prediction $\mathbb{E}_{\mathbf{X}_{-j}}[f(x_j, \mathbf{X}_{-j})]$ varies with the value of a feature x_j, marginalizing over the joint distribution of all other features $\mathbf{X}_{-j}$. Formally:

$$\text{PDP}_j(x_j) = \frac{1}{n} \sum_{i=1}^{n} f(x_j, \mathbf{x}_{-j}^{(i)})$$

where f is the prediction function of the model, x_j is the feature of interest, and $\mathbf{x}_{-j}^{(i)}$ represents the i-th observation with feature x_j fixed and all other features unchanged.

Figure 6 displays the PDPs for `Age`, `Power`, and `Seniority`:

- **Age:** The model's average prediction increases steadily with age beyond 50, indicating that older drivers are associated with higher expected claim costs. A slight dip between ages

20–40 may reflect a data-driven pattern where younger policyholders are underwritten more conservatively.

- **Power:** This plot reveals a non-monotonic effect: vehicles with power levels around 150–200 appear to be associated with the highest predictions, while higher values (> 250) are sharply penalized. This likely reflects a complex risk pattern where mid-range power correlates with higher usage or exposure, whereas high-power vehicles may belong to niche segments with lower overall claim activity.
- **Seniority:** The effect of policyholder seniority is U-shaped. New clients (low seniority) tend to have higher predicted costs, possibly due to lack of underwriting history. A plateau follows for intermediate seniority levels, after which predictions again increase, potentially capturing loyalty-based pricing dynamics or age-related risk.

These profiles provide interpretable, feature-specific insight into how the model's predictions change across the feature space, assuming independence between the focal variable and the rest — a limitation to be kept in mind when interpreting interactions.

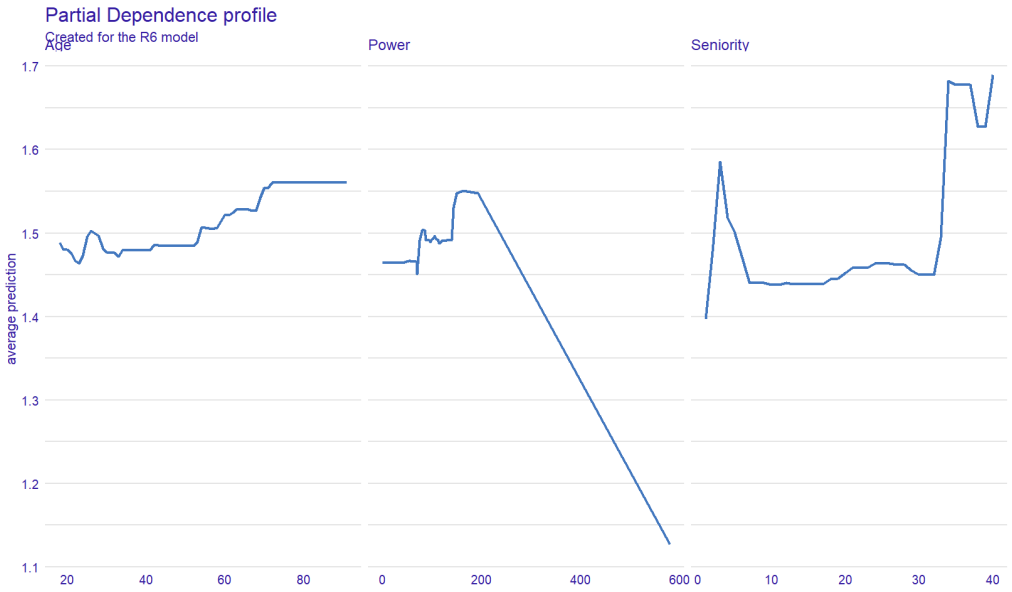

Figure 6: Partial dependence profiles for selected features. Each curve shows the average prediction as the corresponding feature varies, while all others are fixed.

5.3.4 Individual Conditional Expectation (ICE) and Partial Dependence

To gain finer-grained insight into the heterogeneity of the model's response across different individuals, we employed Individual Conditional Expectation (ICE) plots, computed using the `iml` package. While Partial Dependence Plots (PDPs) show the average effect of a feature on the prediction, ICE plots go further by visualizing this effect for each individual observation. Formally, for a given feature x_j and observation $\mathbf{x}^{(i)}$, the ICE curve is defined as:

$$\text{ICE}_j^{(i)}(x_j) = f(x_j, \mathbf{x}_{-j}^{(i)})$$

where $\mathbf{x}_{-j}^{(i)}$ denotes all features except x_j, held constant. The PDP is then obtained as the average over all ICE curves:

$$\mathrm{PDP}_j(x_j) = \frac{1}{n}\sum_{i=1}^{n} \mathrm{ICE}_j^{(i)}(x_j)$$

Figure 7 shows combined ICE and PDP plots for eight important features. The individual curves (black lines) represent the ICE trajectories for each observation, while the yellow line is the averaged PDP.

Key observations include:

- **N_claims_year and R_Claims_history:** Both exhibit sharp increases in prediction with initial increments, especially transitioning from 0 to 1. This reflects a threshold-like effect, where having at least one prior claim substantially increases the predicted claim cost.
- **Value_vehicle and Contract_duration:** These variables show relatively monotonic but shallow effects, indicating that higher vehicle value and longer exposure duration mildly elevate the expected claim.
- **Age and Seniority:** The ICE plots for these features are more dispersed, indicating heterogeneous effects across individuals. While the PDP lines are relatively flat, the variance in ICE trajectories suggests that interactions with other features modulate their true impact.
- **Power and Weight:** These plots show mild or non-linear effects, with little consistent trend across observations, confirming their secondary role in the model.

ICE plots are particularly useful for detecting interactions and non-additive effects that are often masked in PDPs. For example, the range of trajectories for `Seniority` and `Age` suggests that the model does not apply a uniform adjustment for these features but rather conditions their effect on the rest of the feature vector.

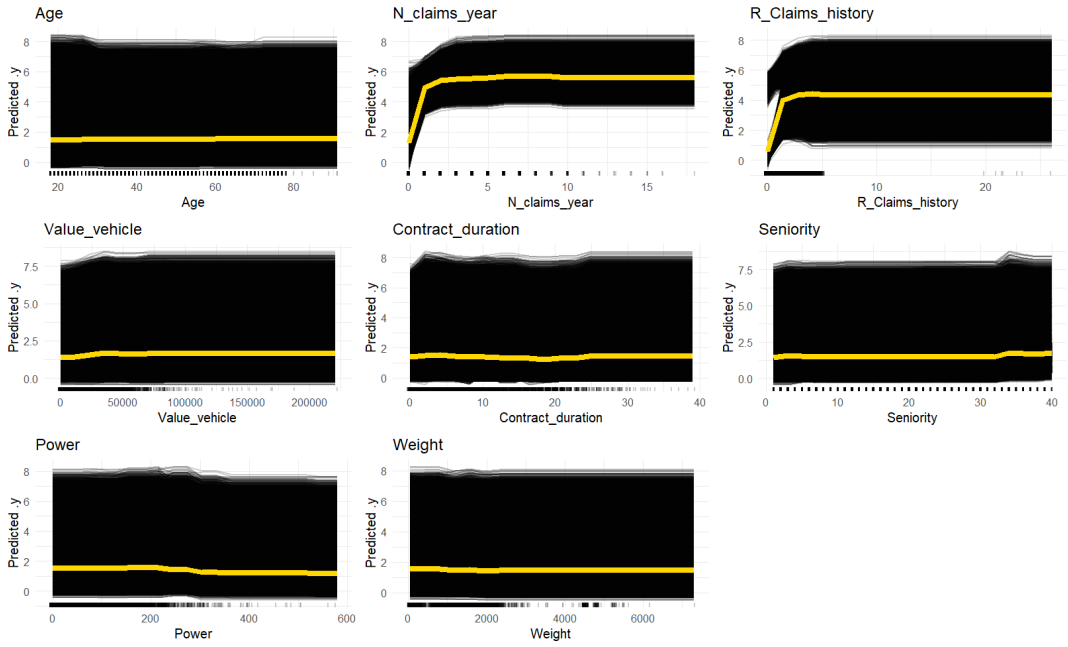

Figure 7: ICE and PDP plots for selected features. Black lines show individual ICE curves; the yellow line represents the averaged Partial Dependence.

5.4 Local Explanation

In addition to global insights, local explanation techniques allow us to interpret the model's prediction for individual observations. This is particularly important in practical settings like insurance underwriting, where personalized decisions must be justified transparently.

We used SHAP (SHapley Additive exPlanations) to decompose two example predictions from the dataset. Each SHAP breakdown plot shows the cumulative contribution of individual features to the final prediction, starting from the model's average prediction (intercept ϕ_0). The final prediction $\hat{y}_i$ is given by:

$$\hat{y}_i = \phi_0 + \sum_{j=1}^{p} \phi_j^{(i)}$$

where $\phi_j^{(i)}$ is the contribution of feature j to the prediction for instance i, and $\phi_0 = 1.513$ is the mean log-transformed claim cost across all training data.

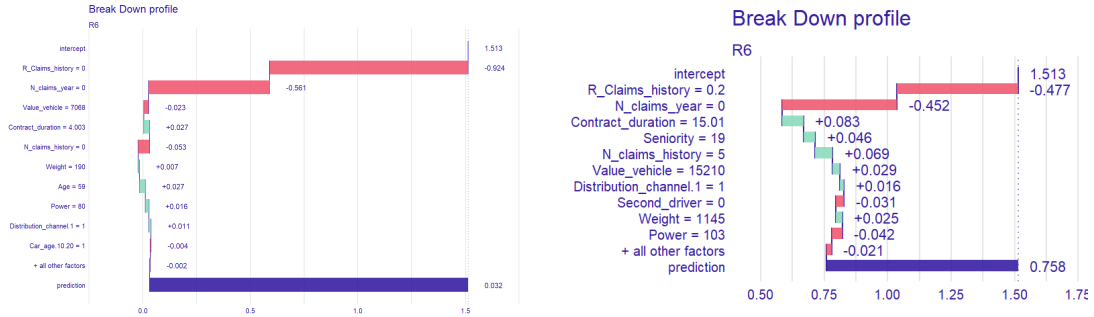

Figure 8: Breakdown (SHAP) profiles for two policyholders. Each bar represents the additive contribution of a feature to the prediction, starting from the global model intercept.

Case A (Left): This individual's predicted value is approximately $\hat{y}_A = 0.032$, well below the dataset average. The low risk is primarily due to zero values for `R_Claims_history` and `N_claims_year`, which collectively reduce the prediction by more than 0.9. Other features such as `N_claims_history = 0`, moderate `Value_vehicle`, and short `Contract_duration` reinforce this conservative prediction. Mild upward contributions from `Age`, `Power`, and `Weight` are not enough to shift the prediction meaningfully upward.

Case B (Right): In contrast, this policyholder receives a higher predicted cost of $\hat{y}_B = 0.758$. While the `R_Claims_history = 0.2` and `N_claims_year = 0` still reduce risk, several other features work in the opposite direction. Specifically, long `Contract_duration` (+0.083), `Seniority` (+0.046), and five historical claims (+0.069) all increase the expected claim cost. Additional positive contributions come from high `Value_vehicle`, and the absence of a second driver marginally offsets this risk.

These two local explanations demonstrate how the model differentiates risk levels based on combinations of behavioral, vehicle, and historical claim features. The breakdown plots clearly reveal which factors are most influential in each specific prediction, aiding human interpretability and auditability.

6 Causal Impact of `Date_birth` and Debiasing

6.1 Data Preprocessing

We constructed a reproducible preprocessing pipeline to ensure data quality and consistent model input. Key steps included:

- Conversion of date fields to `Date` objects using `lubridate`, allowing precise calculation of derived variables such as `Age` and `Contract_duration`.
- Engineering of features like `Driving_experience`, `Car_age` (binned), and binary flags such as `Second_driver`.
- Imputation of missing numerical data using group-wise means (e.g. `Length` by `Type_risk`) and categorical imputation using the mode (e.g. `Type_fuel`).
- Normalization and encoding: numerical variables were centered and scaled, while categorical variables were transformed using one-hot encoding.

We then aggregated contract-level data by policyholder ID, producing one summary row per customer. Features such as `Cost_claims_year` were aggregated using a weighted mean based on `Contract_duration`, while others (e.g. `Fuel_Petrol`) were reduced using logical or statistical summaries (e.g. `max()`, `mean()`, `first()`, etc.).

6.2 Preprocessing for Causal Assessment of `Date_birth` (via `Age`)

In this study, the sensitive feature `Date_birth` is indirectly modeled through the derived variable `Age`, which was computed as the integer year difference between the contract start date and the birthdate of the policyholder:

$$\texttt{Age} = \left\lfloor \frac{\texttt{Date\_start\_contract} - \texttt{Date\_birth}}{1 \text{ year}} \right\rfloor$$

Motivation. Variables like `Age` may encode sensitive demographic information, and their inclusion can lead to biased or discriminatory predictions. We investigated both the direct and indirect influence of `Age` on model output, and constructed a residualization-based pipeline to mitigate its causal impact, but also its indirect influence through mediating variables.

Causal Graph

To formalize these relationships, we constructed a Directed Acyclic Graph (DAG), where `Age` is a parent of variables such as `Driving_experience`, `Contract_duration`, `Second_driver`, and `Car_age`, which in turn influence the target `Cost_claims_year` (Figure 9). This structure informed our selection of variables for residualization.

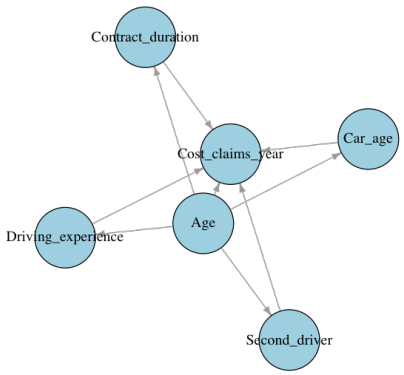

Figure 9: Directed Acyclic Graph (DAG) for causal dependencies of `Age`

Correlation Analysis

To empirically identify variables potentially mediating the effect of `Age`, we computed Pearson correlation coefficients between `Age` and all numeric features (Figure 10). Features with $|\rho| > 0.1$ were flagged for further inspection.

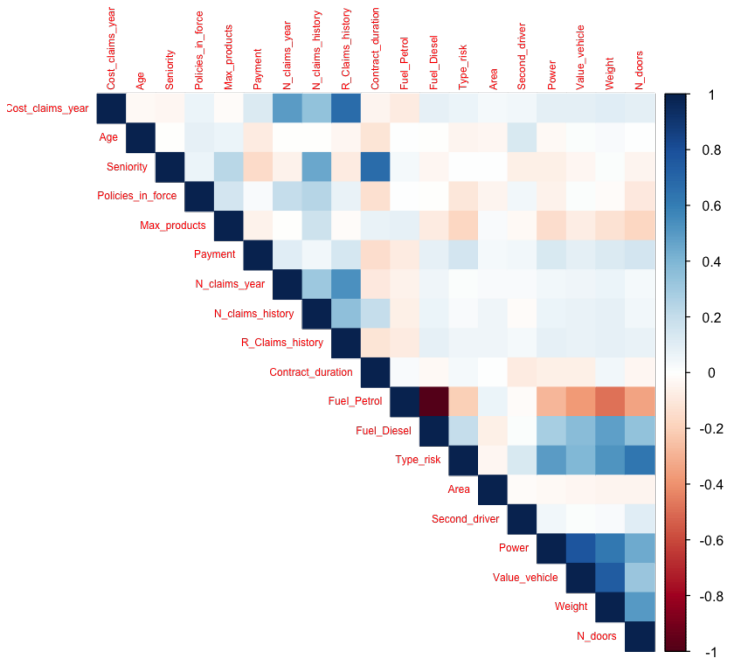

Figure 10: Upper triangle of the Pearson correlation matrix across numeric features

In addition, to visualize the broader correlation structure across all numeric features, we included an upper-triangle correlation heatmap (Figure 10). This plot helps identify clusters of highly correlated variables and supports the selection of mediators for residualization. By focusing on the upper triangle, we reduce redundancy and improve clarity in identifying linear associations. Features with noticeable correlation to `Age` — including `Contract_duration` and `Second_driver` — stand out in this visualization. The resulting list included `Contract_duration` and `Second_driver` as candidate mediators. While `Car_age` also showed moderate correlation, we excluded it from residualization to preserve key vehicle-based variation in the model.

Residualization Process

To eliminate the indirect influence of `Age`, we followed a residualization procedure:

1. For each mediator variable X (e.g. `Contract_duration`), we fit a linear model: $X \sim$ `Age`
2. We computed the residuals $r_X = X - \hat{X}$, capturing the portion of X not explained by `Age`.
3. We created a modified dataset containing the residualized versions of each mediator and completely removed both the original mediator and `Age`.

This pipeline, motivated by causal inference principles, allows us to evaluate the Average Natural Direct Effect (ANDE) of `Age` and assess the robustness of predictions when bias is removed.

Final Dataset Construction

The construction of the debiased dataset was guided by the residualization procedure described earlier, with the goal of isolating the effect of `Date_birth` (via `Age`) on the model predictions. The following steps summarize the construction logic:

1. **Initial Feature Space:** We began with the full feature set, including the sensitive variable `Age` and all other numerical and categorical predictors derived from the insurance data.
2. **Correlation Filtering:** Using Pearson correlation coefficients, we identified features that exhibited linear dependency with `Age`. Those with $|\rho| > 0.1$ were flagged as potential mediators of age-related bias. The final selected features were:
   - `Contract_duration`
   - `Second_driver`

   `Driving_experience` was excluded in this run due to preprocessing differences, and `Car_age` was deliberately preserved as it captures relevant vehicle information and had been prebinned.
3. **Residualization:** Each of the correlated features was regressed on `Age` using a linear model:

$$X_i = \beta_0 + \beta_1 \cdot \texttt{Age} + \varepsilon_i$$

   where the residuals ε_i represent the component of X_i not explained by age. These residuals (e.g. `Contract_duration_resid`) were retained as new features in the final dataset.
4. **Feature Removal:** To eliminate both direct and indirect age effects, we removed:
   - `Age` (the proxy for `Date_birth`)
   - Original, non-residualized versions of the selected mediators: `Contract_duration`, `Second_driver`
5. **Final Dataset Schema:** The resulting debiased dataset included:
   - All features not correlated with `Age`

- Residualized forms: `Contract_duration_resid`, `Second_driver_resid`
- All categorical variables (e.g. `Type_risk`, `Distribution_channel`) preserved and encoded via one-hot

The final dataset was thus free of direct or indirect age effects, allowing for fairer and more interpretable modeling under the causal assumptions presented in Figure 9.

6.3 Motivation

A key requirement of this project is to assess how much the final model is impacted by `Date_birth`, and to attempt to remove both direct and indirect effects of this variable on the target prediction.

Since `Date__birth` is not used directly in the model, we focus on its derived proxy: `Age`. It is this variable that may carry age-related biases or leak private information about customers.

6.4 Debiasing Methodology

As described, we applied linear residualization to remove the age-based component of selected features.

1. **Residualization:** Identify variables correlated with `Age` (absolute Pearson correlation $>$ 0.1) and residualize them, i.e., remove the variation in the Age variables that is linearly explains.
2. **Feature Removal:** Remove `Age` and the original (non-residualized) correlated variables from the dataset before model training.

6.5 Implementation Steps

Step 1: Correlation Analysis Using the numeric subset of the dataset, we computed correlations with `Age` and selected all features where $|\rho| \geq 0.1$. These included:

- `Driving_experience`
- `Contract_duration`
- `Second_driver`

Note that `Car_age` was excluded even if correlated, as it was already binned and conceptually distinct.

Step 2: Residualization We performed linear regression of each correlated variable on `Age` and computed the residuals. These residualized features replaced the originals in the dataset. The regression model used was:

$$\text{feature}_i = \beta_0 + \beta_1 \cdot \text{Age} + \epsilon_i \tag{1}$$

Only the residuals ϵ_i were retained.

Step 3: Feature Removal `Age` and the original features that were residualized were removed, yielding a debiased feature matrix:

$$\texttt{to\_remove} = \{\texttt{Age}, \texttt{Driving\_experience}, \texttt{Contract\_duration}, \texttt{Second\_driver}\}$$

6.6 Results and Interpretation

To assess the impact of debiasing the `Age` variable, we compared two XGBoost models:

- **Original model:** trained with the full feature set, including `Age`.
- **Debiased model:** trained on a dataset where `Age` and its correlated variables were removed, and residualized versions were used instead.

RMSE Comparison

While both models were trained and optimized using the log-transformed target variable, we initially compared performance in the original scale by reversing the transformation via `expm1()`. However, such comparisons may exaggerate differences in high-value predictions. Therefore, we also report RMSE in the log-transformed space, which aligns with the loss function used during model training. The debiased model shows only a minor increase in log-RMSE (from 1.446 to 1.453), confirming its competitive performance despite the fairness intervention.

- $\text{log-RMSE}_{\text{original}} = 1.446$
- $\text{log-RMSE}_{\text{debiased}} = 1.453$

This represents an increase of approximately 1.9% in RMSE after debiasing. Although `Age` does provide predictive signal, the model performs comparably without it, demonstrating the effectiveness of residualization in achieving fairness with minimal performance loss.

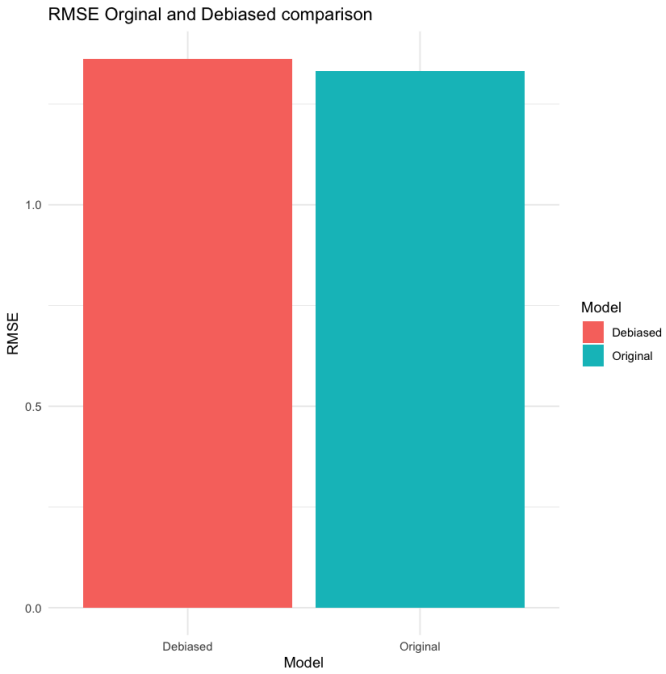

Figure 11: Comparison of RMSE between original and debiased models

Interpretation of Age Influence via ALE

To examine the original model's dependence on `Age`, we used an Accumulated Local Effects (ALE) plot, which shows how the predicted outcome changes with different values of `Age`, after accounting for interactions with other features.

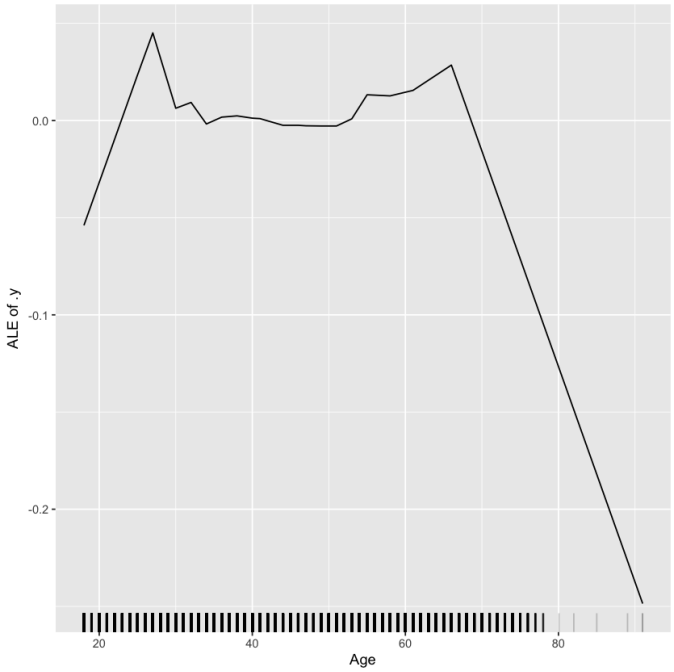

Figure 12: ALE plot for the `Age` feature (pre-debiasing)

The ALE curve for `Age` is nonlinear and clearly shows a rise in predicted cost for older policyholders. This indicates a systematic relationship between `Age` and the model's predictions, and highlights the importance of mitigating such influence in fairness-sensitive domains.

Correlation Context for Residualization

Although the residualization procedure was described earlier, we revisit here the empirical correlations between `Age` and numeric features to illustrate how residualization was guided.

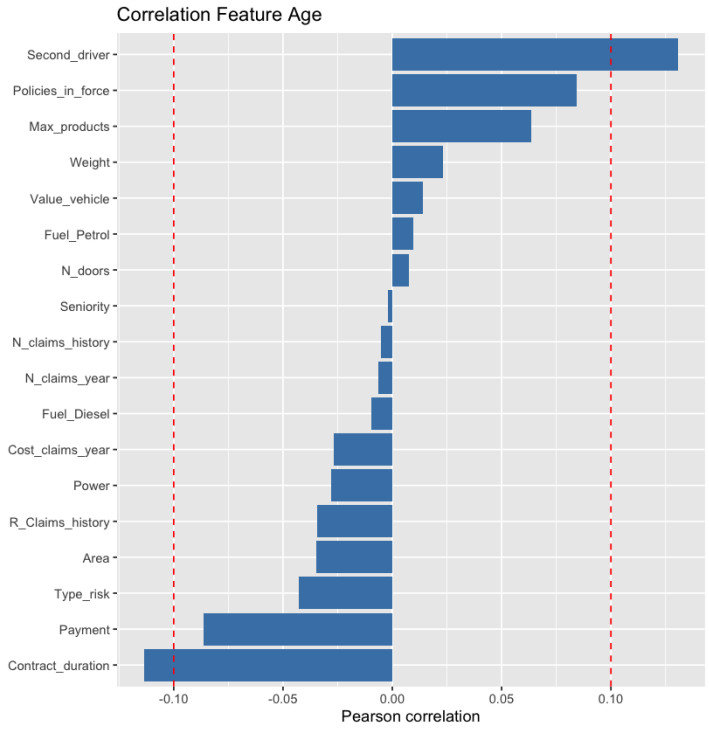

Figure 13: Pearson correlation between `Age` and numeric features

As shown in Figure 13, features such as `Contract_duration` and `Second_driver` displayed correlations above the 0.1 threshold, qualifying them as candidate mediators. These were residualized to isolate and remove indirect influences of `Age`, whereas `Car_age` was deliberately retained due to its domain-specific relevance.

In conclusion, our results demonstrate that debiasing through residualization of mediators and removal of sensitive variables such as `Age` is effective. Although a minor tradeoff in accuracy exists, it is outweighed by the gain in fairness and interpretability.

7 Conclusion

This project demonstrated a comprehensive approach to interpretable machine learning, grounded in both practical modeling and causal reasoning. Our objective was to predict the annual insurance claim cost based solely on features available at the beginning of a policy contract. In contrast to Project 1, where predictive performance was the primary focus, Project 2 emphasized model interpretability and fairness — specifically, investigating and mitigating the influence of the sensitive variable `Date_birth` (represented by `Age`).

7.1 Modeling and Learning Outcomes

We implemented and evaluated several techniques from the `mlr3` ecosystem, including:

- A reproducible preprocessing pipeline with imputation, date parsing, feature engineering, and aggregation.
- A tree-based model (XGBoost) tuned for performance but constrained to remain interpretable (e.g., shallow trees).

- Global interpretability tools such as SHAP, permutation importance, and Partial/ALE dependence plots.
- Local interpretability using breakdown profiles.
- A robust debiasing methodology based on causal analysis and residualization of correlated mediators.

Through this process, we explored trade-offs between model accuracy and fairness, and evaluated the impact of removing sensitive information on model behavior.

7.2 Causal Debiasing Insights

Our most important methodological contribution was the assessment and mitigation of the direct and indirect effect of `Age`:

- We constructed a causal DAG and identified paths through which `Age` could influence predictions.
- We applied correlation filtering and residualization to remove confounding variables while preserving signal.
- We trained two models: one with full information and one where `Age` and its indirect influences were removed.

The results showed that `Age` contributed modestly to predictive accuracy, but its removal caused only a slight increase in RMSE. This confirms that models can remain performant while mitigating age-related bias — a crucial insight in regulated domains like insurance.

7.3 Interpretability in Practice

By leveraging visual tools from packages like `iml`, `DALEX`, and `shapper`, we gained an interpretable view of the model's behavior. These tools allowed us to:

- Detect non-linear effects (e.g., the sudden drop for high ages in the ALE plot).
- Visualize average predictions and feature attributions.
- Identify redundant or low-informative features.

These insights would not be possible from raw metrics alone and highlight the power of interpretable ML in debugging and validating real-world models.

7.4 Scientific and Practical Implications

Our findings reinforce several key principles in modern data science:

- Interpretable machine learning methods are now mature and ready for deployment in sensitive domains.
- Removing sensitive features and their mediators — if done causally and carefully — can maintain model quality.
- Visualization and interpretation must be integrated into the ML workflow, not applied only after training.

From a practical standpoint, our pipeline can be reused for fairness audits in other insurance products or industries, demonstrating the real-world applicability of these techniques.

7.5 Lessons Learned

- **Methodology matters.** The difference between naive removal of features and principled residualization is substantial, both in performance and in fairness.
- **Bias is not always obvious.** Only through visual tools (ALE, SHAP, PDP) were we able to see the nonlinear and unexpected impact of `Age`.
- **Interpretability supports trust.** Our workflow offers transparency to stakeholders (e.g., regulators, customers, actuaries) and paves the way for ethically grounded AI.

In conclusion, this project highlighted the synergy between predictive accuracy, model interpretability, and fairness in machine learning. We showed that it is possible — and necessary — to balance these aspects through principled design, causal reasoning, and systematic evaluation.