STA2536 Home Work 3 Report

Hewei Ding
Julian Schady
Justice Kelvin Kwadzo Dzameshie

November 24, 2025

Part 1 : Poisson GLM vs. GAM for Claim Frequency

The objective of this section is to model the claim frequency, i.e. the expected number of claims per unit of exposure, as a function of rating factors and to assess the benefit of using non-linear effects for the main continuous predictors. Specifically, we aim to:

1. Fit a standard Poisson GLM with log link and log-exposure offset to model the expected claim count.
2. Fit a Poisson GAM using penalized splines for key continuous predictors while treating the categorical variables as factors.
3. Compare both models using deviance and information criteria (AIC), and interpret the estimated smooth functions in terms of claim risk.

Data Description

The claim frequency analysis is based on the `freMTPL2freq` dataset, which contains policy-level records from a French Motor insurance portfolio. Each row corresponds to one policy-year exposure and includes the following variables displayed in Table 1.

Table 1: Key variables in the `freMTPL2freq` dataset

Variable	Description
`ClaimNb`	Number of reported claims during the exposure period.
`Exposure`	Length of time the policy was in force (in years).
Rating factors	Driver age (`DrivAge`), vehicle age (`VehAge`), bonus–malus level (`BonusMalus`), vehicle power, fuel type (`VehGas`), vehicle brand (`VehBrand`), area and regional indicators (`Area`, `Region`), and population density (`Density`).

The dataset therefore combines information on claim experience, exposure, driver characteristics, vehicle characteristics, and geographic risk factors for a large number of policies.

Data Preparation and Exploratory Analysis

We first removed rows with non–positive exposure and missing values in key variables and construct the claim frequency per unit exposure defined as:

$$\text{ClaimFreq}_i \;=\; \frac{\text{ClaimNb}_i}{\text{Exposure}_i},$$

After this filtering step we retain 678,013 policies.

Continuous predictors and correlation

The main continuous predictors considered are `BonusMalus`, `Density`, `VehAge` and `DrivAge`. Figure 1 shows histograms and boxplots for each of these variables.

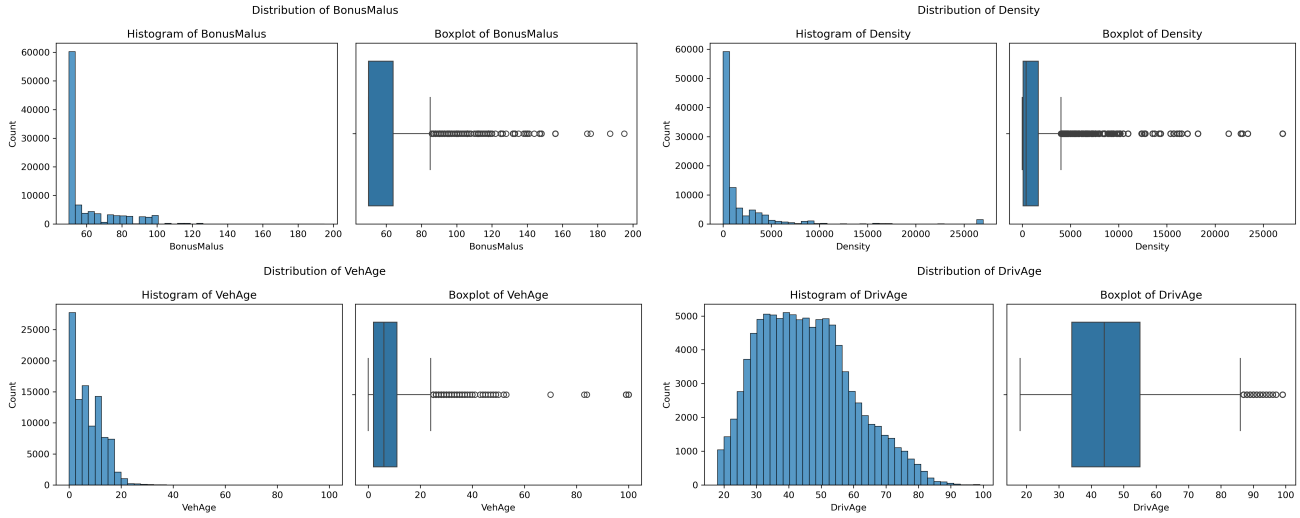

Figure 1: Distributions of key continuous predictors: BonusMalus, Density, VehAge and DrivAge.

The histograms confirm that `BonusMalus`, `Density` and `VehAge` are all strongly right–skewed with many outliers, while `DrivAge` is approximately unimodal with a long right tail at older ages. A correlation heatmap between these variables and `ClaimFreq` is shown in Figure 2.

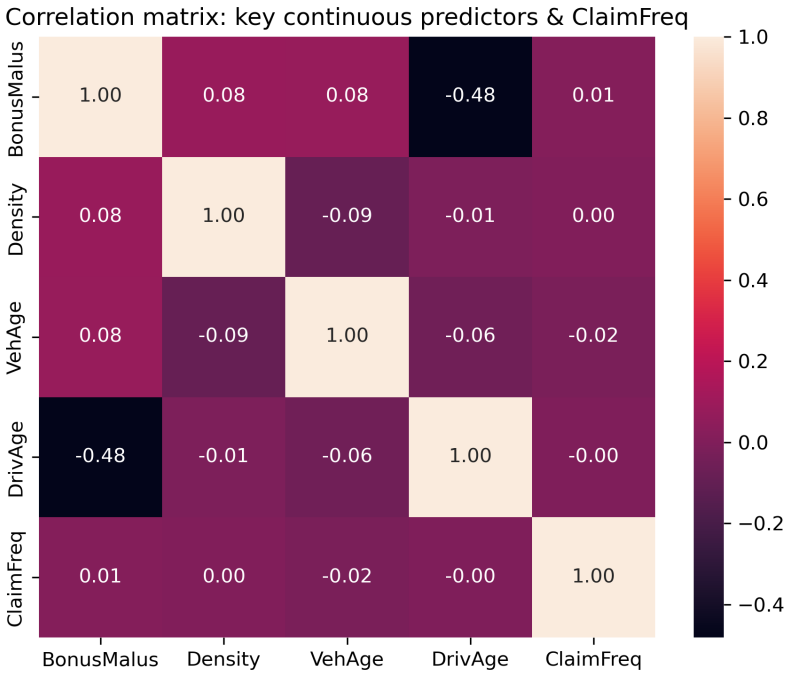

Figure 2: Correlation matrix for key continuous predictors and claim frequency.

Correlations are modest in magnitude, `BonusMalus` is moderately negatively correlated with `DrivAge`

($\rho \approx -0.48$) and only weakly correlated with `ClaimFreq` itself. This suggests limited multicollinearity and motivates modelling non–linear effects of individual predictors rather than strong interactions.

Univariate smoothing

To explore the functional form of the relationship between claim frequency and individual predictors, we fit univariate Poisson regressions with penalised B–spline terms. For each predictor x we fit

$$\text{ClaimNb}_i \mid x_i \sim \text{Poisson}(\mu_i), \qquad \log \mu_i = f(x_i) + \log(\text{Exposure}_i),$$

where f is represented by a B–spline basis and `log(Exposure)` is included as an offset. Binned empirical claim frequencies are overlaid with the fitted smooth.
The resulting curves for `BonusMalus` and `Density`, with piecewise linear (`degree=1`) and cubic (`degree=3`) splines, are shown in Figure 3.

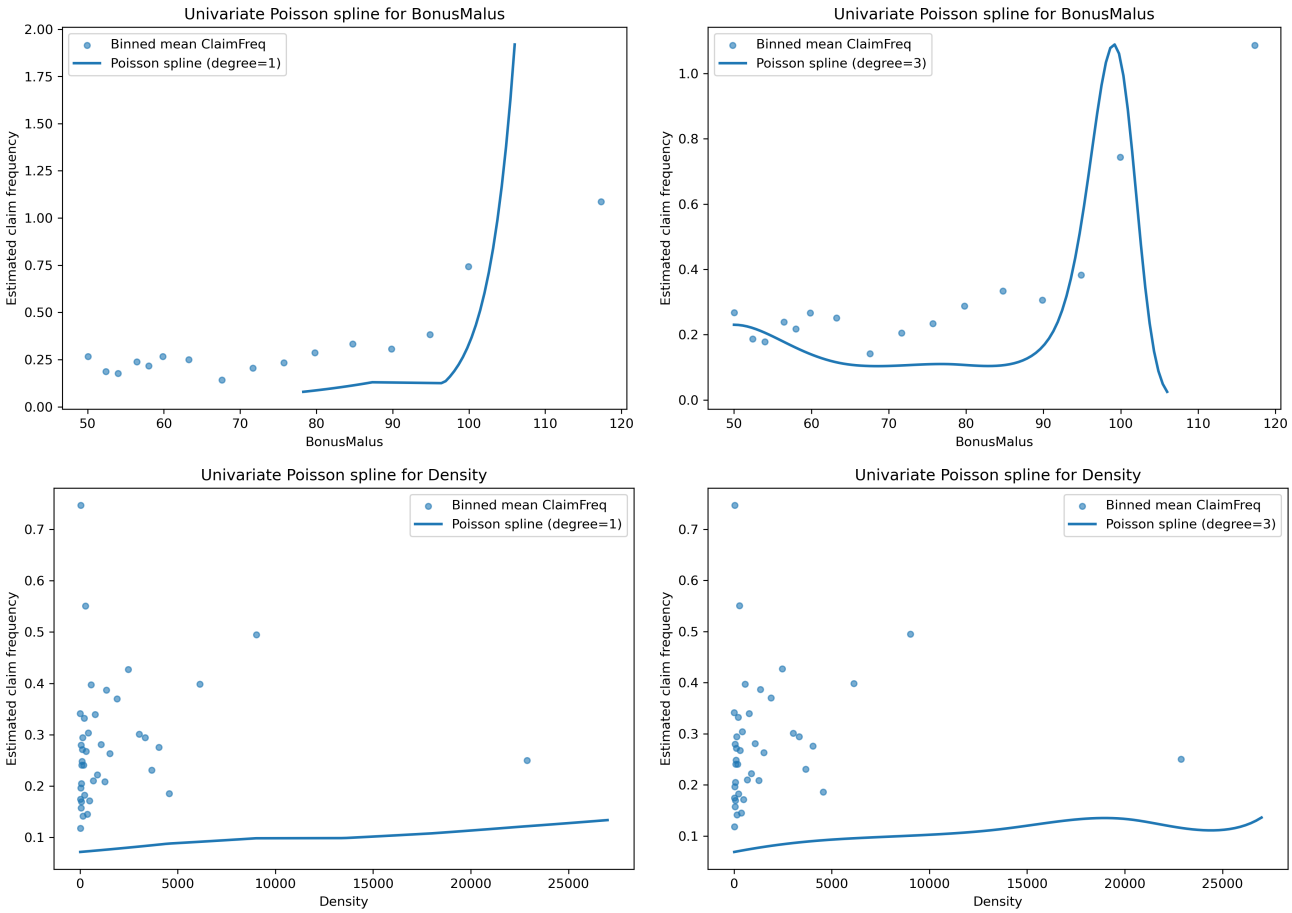

Figure 3: Univariate Poisson spline fits for claim frequency as a function of BonusMalus and Density

For `BonusMalus`, both spline specifications show a relatively flat frequency for low to moderate scores, followed by a sharp increase in claim frequency for the highest risk classes. For `Density`, the fitted curves indicate a mild but non–linear upward trend at high population densities. These patterns suggest that simple linear effects in a GLM may not be sufficient and motivate the use of non–linear smoothers in a GAM.

Model set-up

Let Y_i denote the claim count for policy i with exposure e_i and covariate vector x_i. For both models we assume

$$Y_i \mid x_i \sim \text{Poisson}(\mu_i), \qquad \log(\mu_i) = \log(e_i) + \eta_i,$$

where $\log(e_i)$ is included as an offset and η_i is the linear predictor.
For the Poisson GLM, η_i is specified as a linear combination of the continuous predictors and dummy variables for each categorical level. For the Poisson GAM, η_i is decomposed into smooth functions $f_j(\cdot)$ for the continuous covariates and factor effects for the categoricals:

$$\eta_i = \beta_0 + f_1(\text{BonusMalus}_i) + f_2(\text{Density}_i) + f_3(\text{VehAge}_i) + f_4(\text{DrivAge}_i) + \text{factor terms for categorical variables}$$

The analysis proceeds by preparing the data (computing claim frequency, exploring key predictors), fitting both models on the full dataset, and then comparing their fit and interpretability.

Model fitting

We now model the claim counts `ClaimNb` using a standard Poisson GLM and a Poisson GAM with penalised splines for the continuous predictors. Both models use the same response, offset and predictor set.

Poisson GLM

Let $Y_i = \text{ClaimNb}_i$ denote the number of claims for policy i, with exposure $e_i = \text{Exposure}_i$ and predictor vector $\mathbf{x}_i$. The Poisson GLM fitted assumes

$$Y_i \mid \mathbf{x}_i \sim \text{Poisson}(\mu_i),$$

$$\log \mu_i = \eta_i = \beta_0 + \beta_1 \text{BonusMalus}_i + \beta_2 \text{Density}_i + \beta_3 \text{VehAge}_i + \beta_4 \text{DrivAge}_i + \boldsymbol{\gamma}^\top \mathbf{z}_i + \log e_i,$$

where $\mathbf{z}_i$ are dummy variables for the categorical factors `VehBrand`, `VehGas` and `Region`, and $\log e_i$ is included as an offset. The link function is the canonical log-link, and the standard Poisson assumption $\text{Var}(Y_i \mid \mathbf{x}_i) = \mu_i$ is adopted.
The model is estimated by iteratively reweighted least squares (IRLS). The key results are: residual df $\text{df}_{\text{resid}} = 677\,976$, Deviance: 217,460.45; Pearson $\chi^2 = 1{,}792{,}930$, AIC: 286,857.48 and BIC: $-8{,}885{,}670$.

From the parameter estimation table, Table 17 in the appendix, the continuous predictors are all highly significant with the expected signs. `BonusMalus`: $\hat{\beta}_1 = 0.0226$ with p-value < 0.001 implies that a one–unit increase in BonusMalus multiplies the expected claim count by $\exp(0.0226) \approx 1.023$ (about a 2.3% increase). `Density`: $\hat{\beta}_2 = 6.125 \times 10^{-6}$ also with p-value < 0.001 indicates an increase of 1000 in population density multiplies μ_i by $\exp(0.0061) \approx 1.006$. `VehAge`: $\hat{\beta}_3 = -0.0387$ with a p-value of < 0.001) tell us that each extra year of vehicle age corresponds to a multiplicative factor $\exp(-0.0387) \approx 0.962$ (about a 3.8% decrease) in expected claims. Lastly, `DrivAge`: $\hat{\beta}_4 = 0.0065$ also with p-value < 0.001 implies each additional year of driver age increases expected claims by $\exp(0.0065) \approx 1.0065$.

Dispersion diagnostics are given by the ratios

$$\frac{\text{Deviance}}{\text{df}_{\text{resid}}} = 0.321, \qquad \frac{\text{Pearson } \chi^2}{\text{df}_{\text{resid}}} = 2.64.$$

The Pearson ratio substantially exceeds 1, indicating residual overdispersion relative to the Poisson variance assumption, so standard errors may be somewhat underestimated. For this part of the project we retain the Poisson GLM as a baseline, noting this limitation.

Poisson GAM with penalised splines

To relax the strict linearity assumption for the continuous predictors, we fit a Poisson GAM using penalised B–splines, thus the continuous predictors BonusMalus, Density, VehAge and DrivAge were modelled using cubic B-splines with up to 10 degrees of freedom each, implemented via a BSplines smoother in the GAM. The categorical predictors VehBrand, VehGas and Region were included as standard factor effects via a design matrix of dummy variables, entering linearly alongside the smooth terms in the GAM. The resulting model can be written as

$$Y_i \mid \mathbf{x}_i \sim \text{Poisson}(\mu_i),$$

$$\log \mu_i = \alpha_0 + f_1(\text{BonusMalus}_i) + f_2(\text{Density}_i) + f_3(\text{VehAge}_i) + f_4(\text{DrivAge}_i) + \boldsymbol{\delta}^\top \mathbf{z}_i + \log e_i,$$

where each f_j is represented by up to 10 cubic B–spline basis functions and estimated under a roughness penalty. The spline basis coefficients appear in the output (Table 19)as, `BonusMalus_s0`, ..., `BonusMalus_s8`, etc., so each smooth has a nominal basis dimension of about 9–10 parameters. The effective degrees of freedom (EDF) of each smooth are therefore at most 10 and are shrunk downwards by the penalty, so that the fitted curves are flexible but not overly wiggly.

The GAM is estimated via penalised IRLS using the same Poisson family and offset. The key summary quantities are: $\text{df}_{\text{resid}} = 677{,}947$ and Df Model $= 65$, reflecting the extra spline coefficients. Deviance: 212,569.69; Pearson $\chi^2 = 1{,}616{,}000$. AIC: 282,024.72; and BIC: $-8{,}890{,}172$. From Table 18 in appendix, the categorical coefficients for `VehBrand`, `VehGas` and `Region` remain broadly similar in sign and significance to the GLM, while the spline coefficients collectively capture non–linear patterns in the continuous predictors.

The spline coefficients themselves are not directly interpretable, but the shape of the smooth functions $f_j(\cdot)$ can be visualised by plotting the fitted μ_i over a grid of covariate values (analogous to the univariate spline plots in Figure 3). In practice these plots show the same sharp increase in claim frequency for very high `BonusMalus` and a modest non–linear effect of `Density`.

Dispersion ratios for the GAM are

$$\frac{\text{Deviance}}{\text{df}_{\text{resid}}} = 0.314, \qquad \frac{\text{Pearson } \chi^2}{\text{df}_{\text{resid}}} = 2.38.$$

Overdispersion is still present, but slightly reduced relative to the GLM, suggesting that some of the extra variability has been absorbed by the non–linear terms.

Model comparison and selection

Table 2 summarises the main goodness–of–fit and information–criterion metrics.

Table 2: Model comparison for Poisson GLM and Poisson GAM for claim frequency

Model	Deviance	df_{resid}	Dev/df	Pearson/df	AIC
Poisson GLM	217,460.45	677,976	0.321	2.64	286,857.48
Poisson GAM	212,569.69	677,947	0.314	2.38	282,024.72

The GAM achieves a noticeably lower deviance (a reduction of about 4,900), and a lower AIC by roughly 4,800, despite having additional parameters for the spline bases. This indicates that the improvement in fit easily compensates for the increased complexity in terms of the AIC criterion. The Pearson dispersion ratio is also closer to 1 in the GAM, consistent with a better description of the mean–variance relationship, although some overdispersion remains.

From a modelling perspective, the extra complexity arises from replacing four linear effects in the GLM with four smooth functions, each with up to about 10 effective degrees of freedom. Figure 3 showed clear non–linear behaviour in the relationship between claim frequency and both `BonusMalus` and `Density`, especially at high values. The GAM is specifically designed to capture such non–linearities, while retaining a transparent additive structure and the same Poisson–log link as the GLM.

Discussion of assumptions and justification for smoothers

Both models rely on the standard Poisson regression assumptions:

1. Conditional on the predictors and exposure, $Y_i \sim \text{Poisson}(\mu_i)$ with mean $\mu_i = e_i\lambda_i$.
2. The log–link is appropriate, so that $\log \lambda_i$ is an additive function of the covariates (linear in the GLM, smooth additive in the GAM).
3. Observations are conditionally independent given the predictors.

The exploratory data analysis and univariate spline plots support the additivity assumption but suggest that strict linearity is questionable for some covariates, in particular `BonusMalus` and `Density`. The GAM explicitly relaxes the linearity assumption via smooth $f_j(\cdot)$ while keeping the additive log–mean structure. The Pearson dispersion ratios indicate moderate overdispersion for both models, so the exact Poisson variance assumption is not fully met; however, the GAM reduces the dispersion relative to the GLM, suggesting a better mean specification.

Given the clear non–linear patterns in the univariate smoothing plots, the substantial improvements in deviance and AIC in Table 2, and the slight reduction in overdispersion, we conclude that the use of non–linear smoothing terms in a Poisson GAM is justified for this dataset. The GAM provides a better fit to the claim frequency data while preserving interpretability through smooth, additive effects of the continuous predictors and standard factor effects for the categorical variables.

Part 2: Credit Risk Modelling

The goal of this part is to use German Credit Data to compare three classification models that will predict high or low credit. This analysis will help highlight the key predictors that influence credit risk.

Data Processing and Exploratory Data Analysis

The dataset was downloaded from kaggle. It contains 1000 observations of 9 different variables. There are two continuous predictor variables, the age of the person, and the duration of the loan. There are 6 categorical predictors. The first is the sex of the person. The next is the type of job which can be unskilled and non-resident, unskilled and resident, skilled, and highly skilled. The next is the persons housing situation which can be own, rent, or free if they live with someone like a parent.The next is amount of money in the saving account categorized as little, moderate, quite rich, and rich. The next is checking account which uses the same categories as savings account. The last is the purpose of the loan, categorized as car, furniture/equipment, radio/TV, domestic appliances, repairs, education, business, vacation/others. All categorical variables were encoded into 1 hot encoding variables. The Target variable is the credit amount. In the dataset it is a continuous variable denoting the amount of the credit owed. This is transformed into a categorical target variable, where any amount above the median value is categorized as high risk and any amount below the median is categorized as low risk. The Median Credit amount was 2319.5. This means the target variable is evenly split between low risk and high risk.

The next step was to conduct an exploratory data analysis. There are 1000 observations in the dataset. The initial observation was that the savings account and checking account variables had many missing observations, as seen in Table 3. To deal with this the observations were not removed, as this would eliminate a large potion of valuable data. The meaning of the missing entries was investigated by looking at the original data. In the original data the missing entries actually represented no savings account, no checking account or unknown account status[1]. In theory this should be very useful information to predict credit amount, so instead all missing values were given their own encoding representing that no account exists or is unknown if one exists.

Table 3: Missing Values Analysis

Column	**Missing Count**	**Missing Percentage (%)**
Checking account	394	39.4
Saving accounts	183	18.3

Table 4 shows the statistics of the continuous variables. It includes the credit amount even though it was transformed into a categorical variable for the response. Some interesting information from this is that the average age of the loans is 35. The average duration of the loans are just below two years. The two continuous variables are not correlated with a correlation of -0.04, this can be seen in figure 15 in the appendix. Table 5 shows the distribution of each categorical variable. Some interesting information from this table is that people with skilled jobs makes up the most of the loans, 69% of loans are by men, the majority of the people own their home, the majority have little in their savings account, the majority of entries for the checking account data are missing, and that car loans are the most common loan with TV/Radio and furniture/equipment being close behind.

Table 4: Descriptive Statistics of Continuous Variables

Variable	**Mean**	**Std**	**Min**	**25%**	**Median**	**75%**	**Max**
Age	35.55	11.38	19.00	27.00	33.00	42.00	75.00
Credit Amount (DM)	3271.26	2822.74	250.00	1365.50	2319.50	3972.25	18424.00
Duration (months)	20.90	12.06	4.00	12.00	18.00	24.00	72.00

Table 5: Categorical Variables Summary

Variable	**Category**	**Count**	**Percentage (%)**
Job	unskilled & non-resident	22	2.2
	unskilled & resident	200	20.0
	skilled	630	63.0
	highly skilled	148	14.8
Sex	male	690	69.0
	female	310	31.0
Housing	own	713	71.3
	rent	179	17.9
	free	108	10.8
Saving accounts	little	603	60.3
	moderate	103	10.3
	quite rich	63	6.3
	rich	48	4.8
	missing	183	18.3
Checking account	little	274	27.4
	moderate	269	26.9
	rich	63	6.3
	missing	394	39.4
Purpose	car	337	33.7
	radio/TV	280	28.0
	furniture/equipment	181	18.1
	business	97	9.7
	education	59	5.9
	repairs	22	2.2
	domestic appliances	12	1.2
	vacation/others	12	1.2

Figure 4 visualizes the distribution of the continuous variables. Age is skewed to the right, meaning that cost loans are taken out by younger people with it decreasing with age. The duration of the loan is also skewed to the right with some time periods being very common choices. The credit amount is also skewed to the right, meaning that most loans are around the average with some large loans being outliers. However the response is transformed into a categorical variable of high and low risk so it is not so relevant. They have very low correlation. Figure 16 visualizes the Credit Risk based off each loans purpose and can be found in the appendix.

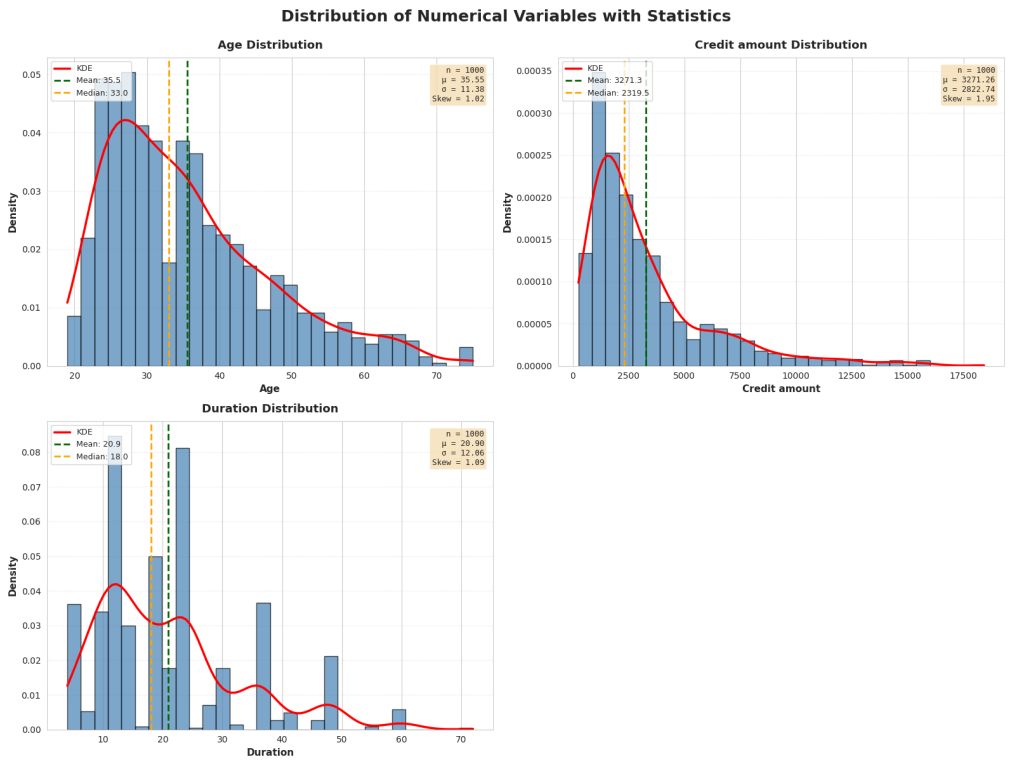

Figure 4

Model Fitting

The goal of the model fitting step is to fit models to classify credit as high or low based off the relevant predictors. We believe all the predictors in the dataset hold relavant information so we will be using all of them. The data set was split into an 80 20 percent train test split. So there is a total of 800 training observations and 200 test ones.

The first classification model that was fit was a logistic regression. The continuous variables age and duration were scaled using a standard scaler for logistic regression. The model was fit to the training set. The model performs well on the training set with a train accuracy of 0.7700, a train precision of 0.7842, a train recall of 0.7450 and a train F1-score of 0.7641. The coefficients of the model can be seen in Table 7, which will be discussed further on.

The next model that was fit was a logistic GAM. This was done by applying smooth functions to the continuous predictors age, and duration. We also included a bivariate smooth interaction between age and duration. This is because the interaction between age and duration of the loan could provide important information. The categorical variables which are added into the GAM using linear smooth function. The logistic GAM had a train accuracy of 0.7612, a train Precision 0.7817, a train Recall of 0.7250, and a train F1 Score of 0.7523. The coefficients of the fitted GAM model can be seen in Table 7. The Smooth functions of the continuous variables will be discussed in the next part.

Finally, the Random Forest Classifier was employed to predict credit risk. The model has 200 decision trees with a maximum depth of 10, a minimum of 20 samples required to split a node, and at least 10 samples per leaf. This configuration balances model complexity while helping to prevent overfitting. This was found using a hyperparameter tuning process. Hyperparameters such as the number of trees (n estimators), tree depth (max depth), minimum samples per split and leaf, and the number of features considered at each

split (max features) were tuned using RandomizedSearchCV, allowing for efficient exploration of parameter combinations. this method also uses GINI split criteria. The model was trained with out-of-bag (OOB) scoring enabled, providing an unbiased estimate of generalization performance without requiring a separate validation set. On the training set, the model achieved an accuracy of 0.7800, a precision of 0.7828, a recall of 0.7750, and an F1 score of 0.7789, indicating strong performance. After training, predictions on the test set were generated, and standard metrics including accuracy, precision, recall, F1 score, ROC AUC, and classification reports were calculated.

Model Evaluation and Comparison

All three models were evaluated on the test set, with results summarized in Table 6. Overall, the models generalize well and achieve comparable performance. The Random Forest model achieves the highest accuracy (0.770) and F1-score (0.7723), while Logistic Regression and Random Forest are tied in recall (0.780). Logistic Regression has slightly lower accuracy (0.765) but maintains competitive precision (0.7573) and the highest ROC-AUC (0.8363). Logistic GAM performs similarly, with an accuracy of 0.760 and balanced precision and recall values. The overall similarity in performance suggests that all models are capturing the key patterns in the data, indicating that the most predictive features are consistently identified across modeling approaches.

Table 6: Credit Risk Prediction - Model Comparison

Model	Accuracy	Precision	Recall	F1-Score	ROC-AUC
Logistic Regression	0.765	0.7573	0.780	0.7685	0.8363
Logistic GAM	0.760	0.7653	0.750	0.7576	0.8340
Random Forest	0.770	0.7647	0.780	0.7723	0.8313

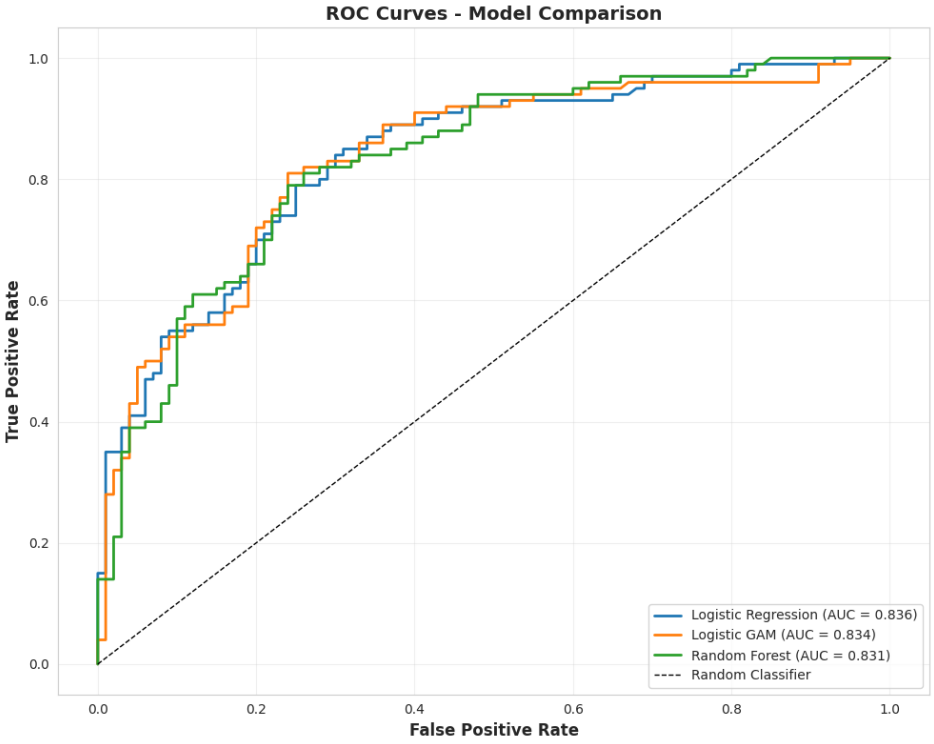

Figure 5: ROC curve for models

Table 7 shows the model coefficients for the logistic regression, and GAM model alongside with the Feature Importance values from the Random Forest Models. For the logistic regression the largest coefficient is the duration of the loan being 1.6572. This makes sense as a loan with a longer duration makes more sense to be high risk. Some features are missing in the table, that is because they are incorporated into the intercept value, therefore the coefficient values are in relationship to the intercept. Some other interesting we can see that the coefficient for male is 0.1981, meaning that the log odds that a loan for a male person are high risk is increases by 0.1981 compared to if it were to a female. The coefficient of age is almost zero, meaning there is little influence between age and wether the loan is high or low credit. Another interesting coefficient is that high skilled job being 0.4301. this has to do with the size of the loan taken out by a person with a high skilled job. In terms of the Logistic GAM it has some very similar trends in coefficients as the logistic regression model. The smooth functions of Age and Duration can be seen in Figure 6. It seems to show that as age increases there is less probability that the loan will be classified as high. Whereas as duration increases the probability increases that a loan will be classified as high which agrees with the intuition from durations coefficient in the logistic regression model. The bivariate smooth function can be seen in Figure 7. It seems to increase the odds of a high risk credit when age is low and duration is high. the Feature importance values also seem to agree with the previous two models, as the highest value is for the duration of the loan. This indicates that it is the most important variable for all three models for determining low and high risk loans.

Table 7: Coefficients and Feature Importance Across Models

Group	**Feature**	**Logistic Regression Coefficients**	**Logistic GAM Coefficients**	**Random Forest Feature Importance**
Gender	Male	0.1981	-0.1519	0.020238
Housing	Own	-0.0419	0.0833	0.018515
Housing	Rent	0.0573	0.0079	0.006312
Saving Accounts	None	0.1467	0.0534	0.019558
Saving Accounts	Moderate	0.0189	0.1822	0.002879
Saving Accounts	Quite Rich	0.0827	-0.0685	0.000905
Saving Accounts	Rich	-0.0467	0.1685	0.003161
Checking Account	None	0.0123	-0.0853	0.021980
Checking Account	Moderate	0.1322	0.0143	0.017072
Checking Account	Rich	-0.0495	0.3025	0.004820
Purpose	Car	0.1983	-0.1119	0.024028
Purpose	Furniture/Equipment	0.2532	-0.3432	0.017295
Purpose	Domestic appliances	-0.1943	-0.1927	0.000000
Purpose	Education	-0.1377	0.3985	0.005210
Purpose	Radio/TV	-0.1982	-0.0276	0.045277
Purpose	Repairs	-0.0246	-0.0066	0.000109
Purpose	Vacation/others	-0.0391	-0.2087	0.000228
Job	Unskilled and Resident	0.0361	-0.0745	0.052070
Job	Skilled	0.1381	0.3942	0.019862
Job	Highly Skilled	0.4301	1.5317	0.036090
Continuous Variables	Duration	**1.6572**	-	**0.610183**
Continuous Variables	Age	-0.0308	-	0.074209

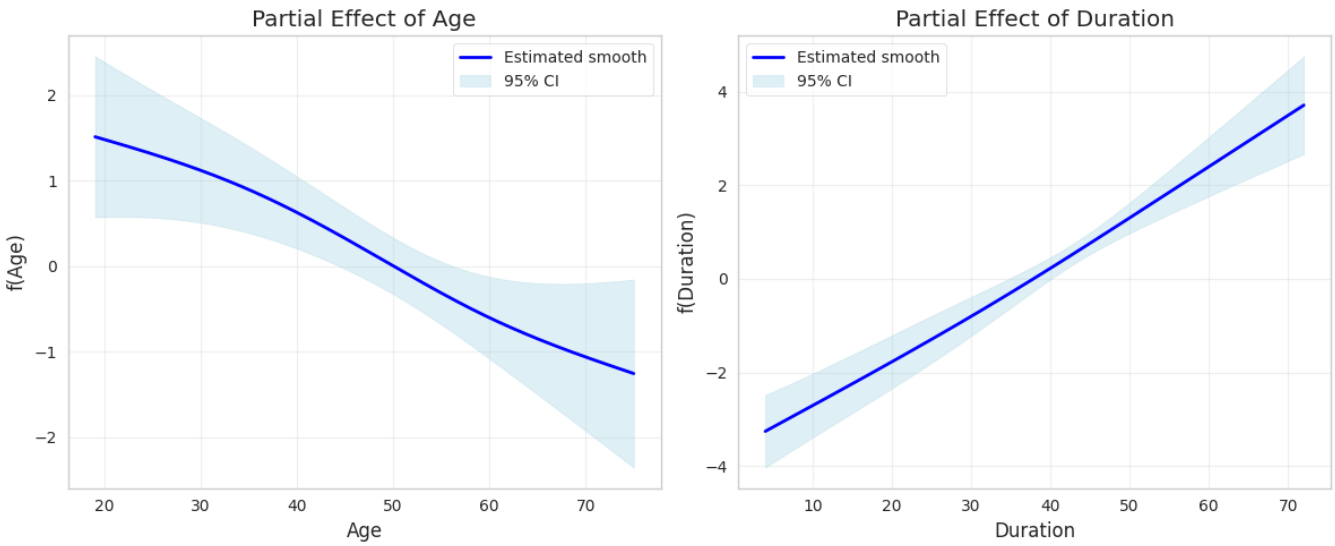

Figure 6: Smooth Function for Age and Duration

Figure 7: Bivariate Smoothing Function for Age and Duration

Insights an Conclusion

The model comparison indicates that all three models—Logistic Regression, Logistic GAM, and Random Forest—perform similarly on the test set, with comparable accuracy, precision, recall, F1-score, and ROC-AUC metrics. Logistic Regression achieves the highest recall (0.780) and competitive precision, suggesting it is slightly better at correctly identifying high-risk credit cases. The Random Forest achieves the best

overall accuracy (0.770) and provides robust estimates of feature importance, while the Logistic GAM captures non-linear relationships through smooth functions, providing additional interpretability of continuous predictors.therefore we believe the best performing model s the Random Forest model as it achieve the best accuracy, Recall, and F1-Score. The GAM interaction plots reveal meaningful relationships between the continuous predictors Age and Duration. Specifically, as Age increases, the probability of a high-risk loan decreases, whereas longer loan durations increase the probability of high-risk classification. The bivariate interaction indicates that younger applicants with longer-duration loans have the highest predicted credit risk, highlighting an important interplay that simple linear models may not fully capture. Across all models, Duration emerges as the most important predictor for credit risk, with consistently high coefficients and feature importance values. Categorical variables such as Job and Purpose also have meaningful contributions: high-skilled jobs and larger loan purposes tend to increase credit risk, while other categories have nuanced effects depending on the model. Gender has a modest effect, with male applicants showing slightly higher log-odds of high-risk credit in the Logistic Regression model. These insights reinforce the intuitive understanding of credit risk: longer loan commitments, certain occupations, and specific loan purposes are associated with increased risk, while Age and other demographic variables play more subtle roles.

Part 3: Regression Tree for Claim Severity

In this part; the severity analysis combines the `freMTPL2sev` and `freMTPL2freq` datasets. The `freMTPL2sev` file contains individual claim records with the total paid amount `ClaimAmount` and the policy identifier `IDpol`. The `freMTPL2freq` dataset provides the corresponding policy-level characteristics and claim counts.

Objective

The objective of Part 3 is to build and interpret a regression tree model for average claim severity. The main goals are to:

1. Identify combinations of policy characteristics associated with high and low average severities.
2. Use cost–complexity pruning with k-fold cross-validation to select an appropriate tree size that balances fit and interpretability.
3. Visualise the final pruned tree and describe its key splitting rules, terminal nodes, and most important predictors in terms of claim severity segmentation.

Data Preparation and Exploratory Analysis

Construction of the response

The `freMTPL2sev` dataset was first aggregated to the policy level by summing `ClaimAmount` and counting claims `ClaimNb_from_sev` per `IDpol`. The resulting severity dataset therefore contains, for each claim-bearing policy, its average claim size together with the same rating factors used in the frequency analysis (driver age, vehicle age, bonus–malus, vehicle brand and fuel type, density, and region). Policies with strictly positive `ClaimNb` were retained and the average claim severity was defined as

$$\text{AvgSeverity}_i = \frac{\text{ClaimAmount}_i}{\text{ClaimNb}_i},$$

for each policy i with at least one claim. This produced a working dataset of $n = 24{,}944$ policies.

Descriptive statistics

Summary statistics of the response are:

Table 8: Descriptive statistics for average claim severity (in currency units).

	Count	Mean	Std. dev.	Min	Q1	Median	Q3	Max
AvgSeverity	24 944	2 221.37	28 992.57	1.00	710.56	1 172.00	1 228.08	4 075 401.00

Most policies have relatively modest average severities (interquartile range roughly 700–1 230), but the maximum value exceeds 4 million, indicating the presence of extreme outliers.

Distributional shape and skewness

The raw average severity is extremely right-skewed, with skewness ≈ 116.6. The extremely high skewness of the raw average severities is driven by a small number of very large claims, and reflects the strong right–tail behaviour of the severity distribution rather than a computation error. The histogram in Figure 8 shows a large spike near zero with a very long right tail driven by a small number of very large claims.

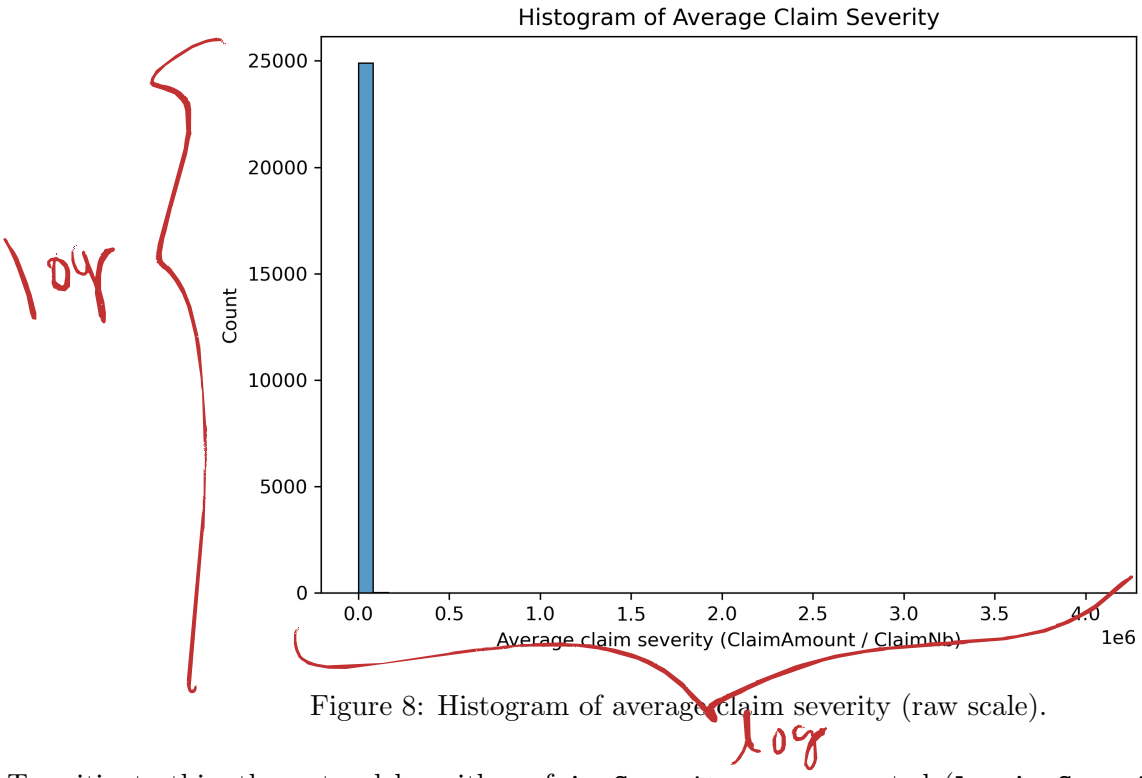

Figure 8: Histogram of average claim severity (raw scale).

To mitigate this, the natural logarithm of `AvgSeverity` was computed (`log_AvgSeverity`) and its histogram plotted (Figure 9). The log-transformed response has skewness ≈ -0.59 and is much closer to symmetric, although some tail behaviour remains.

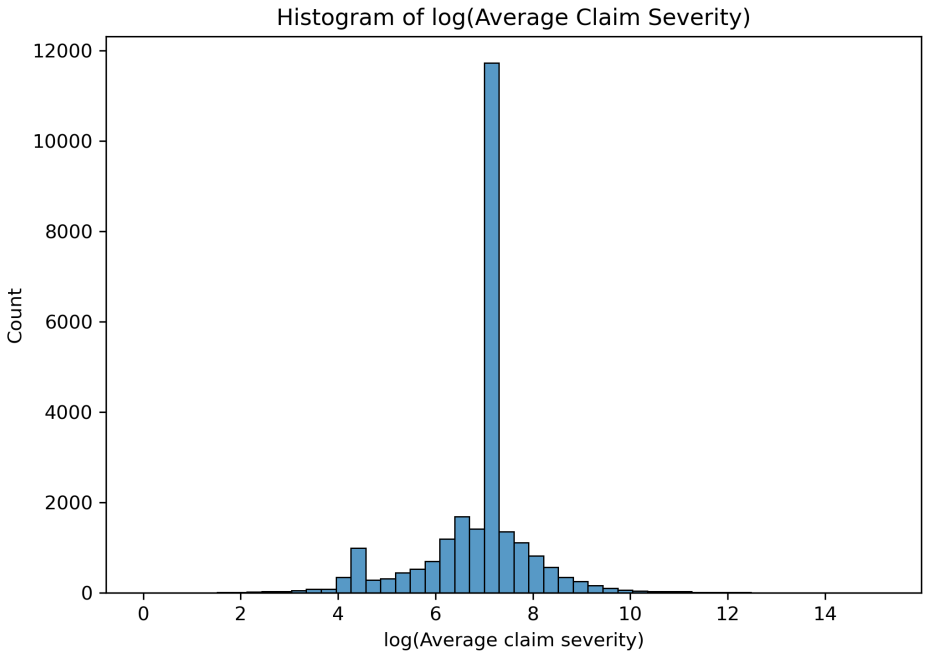

Figure 9: Histogram of log(AvgSeverity).

For the regression tree in this part of the project, we proceeded with the raw severity as target, but the plots and skewness clearly show that the outcome is heavy-tailed and dominated by a few very large claims.

Model set-up

Regression tree model

Let S_i denote the observed average claim severity for policy i and x_i its covariate vector. A regression tree partitions the predictor space into disjoint regions $R_1, \ldots, R_M$ and predicts a piecewise-constant mean severity,

$$\hat{S}(x) = \sum_{m=1}^{M} \mu_m \, \mathbb{I}(x \in R_m), \qquad \mu_m = \frac{1}{|R_m|} \sum_{i: x_i \in R_m} S_i,$$

where splits are chosen greedily to minimise the sum of squared errors and $\mathbb{I}(\cdot)$ is the indicator function.

In the implementation, a large initial tree is fitted on the training data using `BonusMalus`, `Density`, `VehAge`, `DrivAge`, `VehBrand`, `VehGas`, and `Region` as predictors (with categoricals one-hot encoded). A cost–complexity pruning path is then computed and a grid of pruning parameters α is evaluated via k-fold cross-validation using mean squared error. The selected α is used to prune the tree and obtain the final model, which is subsequently visualised and interpreted in terms of severity risk segments.

Model fitting

Predictors and design matrix

As in Part 1, the following predictors were used:

Table 9: Types of predictors used in the models

Type	Predictors
Continuous	`BonusMalus`, `Density`, `VehAge`, `DrivAge`
Categorical	`VehBrand`, `VehGas`, `Region`

A design matrix was created by applying one-hot encoding to the categorical variables (dropping the first level of each), resulting in 36 predictor columns. The final dataset for modelling had `X_full` of shape $(24\,944, 36)$ and a response vector `y_full` of length 24 944.

The data were randomly split into a training set (80%, 19 955 policies) and a test set (20%, 4 989 policies) with a fixed random seed for reproducibility. An initial "large" tree was fitted on the training data with no pre-pruning thus maximum depth was set to None. This tree had depth 53 and 18 035 leaves, which essentially overfit the training data.

Cost–complexity pruning and cross-validation

To control overfitting, cost–complexity pruning was used. For a tree T with leaves $|T|$, the cost–complexity criterion is

$$C_\alpha(T) = \mathrm{SSE}(T) + \alpha|T|,$$

where $\mathrm{SSE}(T)$ is the training SSE and $\alpha \geq 0$ is the cost–complexity parameter. Increasing α penalises large trees and induces pruning.

Using `cost_complexity_pruning_path` on the training sample of 19,955 observations, the initial depth-53 tree with 18,035 leaves yielded a sequence of 13,562 candidate values of the penalty parameter α, ranging from 0 to approximately 1.39×10^8. For computational efficiency, a representative grid of 40 α values from this range was selected and evaluated using 5-fold cross-validation (`KFold` with shuffling and a fixed random seed), with negative mean squared error as the scoring metric. For each α, we recorded the mean cross-validated MSE, its standard deviation, and the resulting tree depth and number of leaves; the first rows of these results show that very small α values correspond to extremely large trees (depth 53, roughly 18k leaves), while larger α values gradually reduce tree size. The chosen α from this grid was then used to prune the initial tree to the final model reported in Table 10.

Model selection and pruning

Selection of the pruning parameter

We first identified the α that minimised the mean cross-validated MSE. This unconstrained optimum corresponded to an almost completely pruned tree with a single leaf. To avoid such a trivial model, the selection was restricted to trees with more than one leaf. Among those, the chosen α was

$$\alpha^\star \approx 1.35 \times 10^3,$$

with cross-validated MSE $\approx 2.11 \times 10^9$ and standard deviation $\approx 2.14 \times 10^9$. On the tuning set, this α produced a tree with depth 32 and 1 135 leaves.

The relationship between cross-validated MSE and α is shown in Figure 10, where the selected $\alpha^\star$ is marked by a vertical line. The curve is relatively flat across a wide range of α values, reflecting the difficulty of predicting extremely variable claim severities.

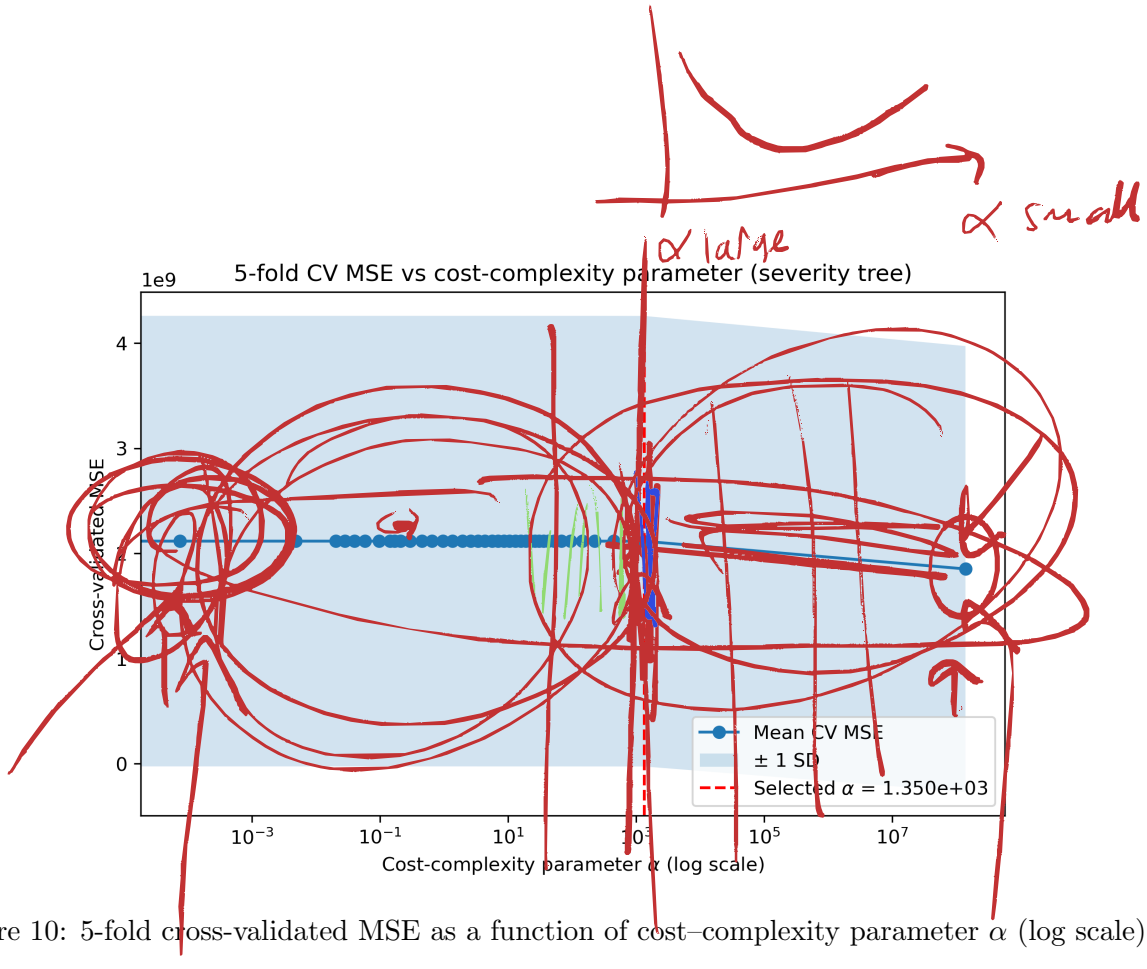

Figure 10: 5-fold cross-validated MSE as a function of cost–complexity parameter α (log scale).

Final pruned tree and performance

Using $\alpha^\star$, a pruned regression tree (`final_tree`) was fitted on the full training set. This tree still had substantial complexity (depth 32, 1 135 leaves), indicating that the algorithm required many splits to adapt to the highly irregular severity surface.

Its predictive performance was evaluated on both the training and test sets shown in Table 8:

Table 10: Performance of the final pruned regression tree.

Model	Depth	Leaves	Train MSE	Test MSE
Final pruned tree ($\alpha^\star$)	32	1 135	1.3432×10^6	4.8483×10^8

The train MSE is much smaller than the test MSE, which is expected given the extreme outliers and the large number of leaves; nonetheless the pruned tree is substantially simpler than the initial unpruned tree.

For interpretability, a truncated plot of the top four levels of `final_tree` was generated (Figure 11), focusing on the main upper-level splits rather than the full depth-32 structure.

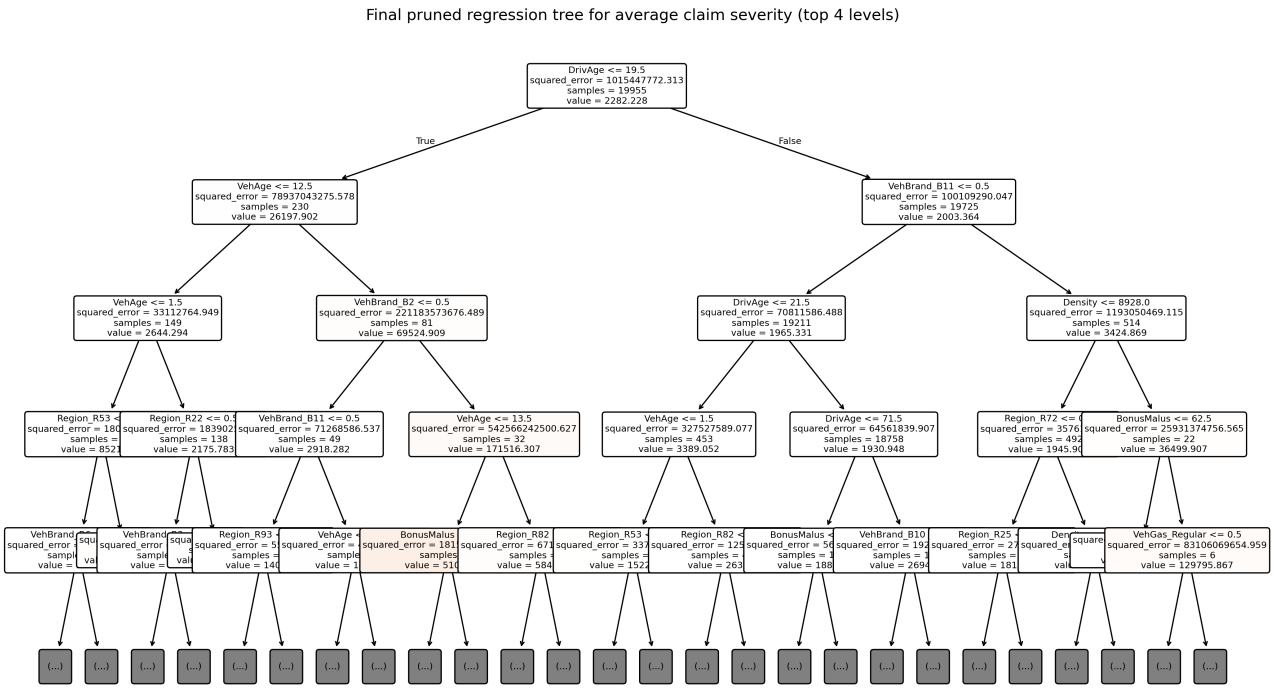

Figure 11: Top four levels of the final pruned regression tree for average claim severity.

Analysis and interpretation

Model assumptions

Regression trees make relatively weak distributional assumptions compared with parametric GLMs. The key assumptions are:

1. observations are approximately independent;
2. the conditional mean of the response is piecewise constant over rectangular regions in predictor space;
3. the quality of a split is assessed via squared error loss (implicitly assuming minimisation of MSE).

Given the aggregated policy-level data, the independence assumption is reasonable, although repeat policies across years could violate it. The extreme right tail of `AvgSeverity` means that variance is highly heterogeneous, but regression trees do not require constant variance. The very large depth and number of leaves in the pruned tree suggest that a simple piecewise-constant approximation struggles to capture the heavy-tailed severity distribution, but the model is still valid under its assumptions.

Key splitting variables in the upper tree

In the path-extraction, the predictors appearing in upper splits (depth 0–2) were counted. The most frequent were; `DrivAge` (appearing in two upper-level splits), `VehAge` (two splits), and, each appearing once, `VehBrand_B2`, `VehBrand_B11`, and `Density`.

This indicates that driver age, vehicle age, certain high-risk vehicle brands, and the population density of the policyholder's region are the most important variables in segmenting claim severity at the top of the tree.

Example risk segments (paths to terminal nodes)

Looking at all the enumerated 1,135 leaves we produced three illustrative risk segments based on their predicted severities.

High-severity segment

One extreme leaf (leaf 52) has a predicted average severity of approximately 4,075,401, which corresponds to the largest observed claim in the data. The leaf is defined by the following combination of risk factors (Table 11) and thus captures a very concentrated high-severity segment.

Table 11: Conditions defining the high-severity leaf (leaf 52)

Variable	**Condition**
DrivAge	≤ 19.5 years
VehAge	$12.5 < \text{VehAge} \leq 13.5$ years
VehBrand	B2
BonusMalus	> 97.5
Region	R24

This segment therefore corresponds to very young drivers with old vehicles of a specific brand, very poor past claim experience, and residence in Region R24. The extremely high predicted severity reflects the influence of a single catastrophic claim in this region of the predictor space.

Medium-severity segment

A more typical high-risk segment (leaf 45) has a predicted average severity of around 1,687. The leaf is defined by the following combination of risk factors (Table 12), representing very young drivers with relatively old vehicles of brand B11 using regular fuel. Their predicted severities are higher than the portfolio average but far below the catastrophic segment above.

Table 12: Conditions defining the medium high-severity leaf (leaf 45)

Variable	**Condition**
DrivAge	≤ 19.5 years
VehAge	$12.5 < \text{VehAge} \leq 15.5$ years
VehBrand	B11 (not B2)
VehGas	Regular

Low-severity segment

At the other end of the spectrum, leaf 1133 has a predicted average severity of about 1.74. The leaf is defined by the combination of risk factors in Table 13, corresponding to middle-aged drivers with moderate bonus–malus scores, mid-range density, vehicle brands other than B11 and B13, and residence in Region R52.

Table 13: Conditions defining the very low-severity leaf (leaf 1133)

Variable	Condition
DrivAge	$52.5 < \text{DrivAge} \leq 53.5$ years
BonusMalus	≤ 70.5
Density	$21.5 < \text{Density} \leq 159.5$
Region	R52 (not R21, R22 or R93)
VehBrand	not B11 and not B13

The predicted severity in this segment is essentially negligible, reflecting a combination of favourable driver age, bonus–malus and territorial factors that the tree associates with very low average claim sizes.

Overall assessment

The regression tree captures intuitive patterns:

1. severity is strongly influenced by driver age and vehicle age, with young drivers and older vehicles associated with higher severities;
2. certain vehicle brands and regions (e.g. B2, B11, R24) are implicated in high-severity segments;
3. bonus–malus level and regional density further refine high- and low-risk groups.

However, the heavy-tailed nature of the severity data leads to very deep and fragmented trees even after pruning, and the cross-validated MSE is largely driven by a small number of extremely large claims. The log-transformed histogram suggests that modelling log severity rather than the raw amount might yield smoother relationships and more stable trees, but this was beyond the scope of the current assignment.

In summary, the cost–complexity pruned regression tree provides a flexible, non-parametric segmentation of claim severity that highlights key factors (driver age, vehicle age, brand, region, and bonus–malus), but its complexity and sensitivity to outliers underline the challenges of modelling motor claim severities on the original monetary scale.

Part 4: Predicting Frequency of Insurance Claims

Data Analysis

We build on the cleaned dataset prepared in Part 1, which contains 678,013 policy-year observations from the `freMTPL2freq` portfolio. Each record includes exposure, claim count, driver and vehicle characteristics, and geographic risk factors. Since the data contain no missing values and all observations have strictly positive exposure, the dataset is suitable for frequency modelling without additional filtering.

From Part 1, histograms and boxplots showed that `VehAge`, `BonusMalus`, and `Density` are highly right-skewed, while `DrivAge` is more symmetric with a moderate right tail. Categorical factors (e.g. `VehBrand`, `Area`, `Region`) exhibit substantial variation in mean claim frequency, supporting their inclusion as rating factors.

Data Preprocessing

Missing data and data quality. The dataset used in Part 4 contains no missing values in any variable, and all rows have strictly positive exposure. Therefore no imputation is required. As a minor data-quality adjustment, categorical variables (`VehBrand`, `VehGas`, `Area`, `Region`, `IDpol`) are cast to categorical types, and integer features stored in nullable `Int32` format are converted to standard `int64`, which is necessary for downstream encoding and `scikit-learn` model fitting.

Feature engineering. Since Random Forests and Gradient Boosted Trees can automatically learn non-linear effects and interactions, extensive feature engineering is not required. All original predictors are retained, and the empirical claim frequency

$$\widehat{\text{ClaimFreq}} = \frac{\texttt{ClaimNb}}{\texttt{Exposure}}$$

is created only for exploratory plots. This derived feature is not used as a modelling target.

Scaling and normalisation. Tree-based learners do not require feature scaling. Random Forests and Gradient Boosted Trees split on thresholds of raw predictor values, so their behaviour is invariant to monotonic rescaling. For this reason no standardisation or normalisation is applied to continuous predictors in the main analysis.

Encoding. `Scikit-learn` implementations of Random Forests and Gradient Boosted Trees require numerical inputs, so all categorical predictors are converted to dummy (one-hot) indicators. After the train–test split, we apply `pandas.get_dummies` to the training predictors, expanding the original factors `VehBrand`, `VehGas`, `Area` and `Region` into a set of binary indicator variables. This increases the design matrix to 42 predictors in the training set: five continuous features (`VehPower`, `VehAge`, `DrivAge`, `BonusMalus`, `Density`) and 37 dummy variables for brand, fuel type, area, and region (e.g. `Region_Aquitaine`, `Region_Ile-de-France`, `VehBrand_B10`, `Area_C`). The resulting feature matrix is sparse in the categorical part, but still relatively high-dimensional given the large sample size ($\approx 5.4 \times 10^5$ rows).

Dummy encoding is applied *after* the train–test split: the dummy structure is learned from the training set only, and the test set is subsequently realigned to the same columns, with any missing levels filled by zeros. This avoids leakage of category information from the test data into the training step. In principle, tree-based models can handle such wide, dummy-encoded inputs; however, in practice the combination of a very large sample size and dozens of binary predictors leads to substantial computational cost. Training Random Forests and Gradient Boosted Trees on this design matrix is noticeably slower, and hyperparameter tuning must be restricted to a relatively small grid to keep runtimes manageable. This highlights a practical limitation of applying flexible tree ensembles to large-scale insurance frequency data with many categorical rating factors.

Summary. Given the high quality of the dataset and the robustness of tree-based methods, the required preprocessing is minimal: type cleaning, optional feature construction for exploration, and one-hot encoding of categorical variables. No imputation, outlier removal, or feature scaling is required for the models used in Part 4.

loss function?

Random Forest

To provide a flexible non-parametric benchmark to the Poisson GLM, we fit Random Forest regression models to the claim *frequency* rather than the raw claim count. For each policy i with claim count Y_i and exposure e_i we define

$$\lambda_i = \frac{Y_i}{e_i},$$

and use λ_i as the response in a Random Forest regressor, with e_i supplied as a sample weight. This weighting scheme mirrors the role of the offset $\log(e_i)$ in the Poisson GLM and concentrates more influence on policies with higher exposure. The model therefore learns a non-linear mapping from the rating factors to expected claim frequency, which can be converted back to an expected claim count $\widehat{\mu}_i = \widehat{\lambda}_i\, e_i$ if required.

Baseline specification. The baseline Random Forest model is a regression forest with n_estimators $=$ 100 trees, grown on bootstrap samples of the training data. Trees are allowed to grow to full depth (`max_depth=None`) but must contain at least 20 observations per leaf (`min_samples_leaf = 20`) to reduce variance and avoid overfitting to very small subsets of policies. At each split a random subset of predictors

of size `max_features="sqrt"` is considered, following standard Random Forest practice. Out-of-bag (OOB) sampling is enabled (`oob_score=True`), providing an internal estimate of the generalisation error without requiring a separate validation set.

Hyperparameter tuning and cross-validation. To assess the sensitivity of the results to the choice of hyperparameters and to obtain a tuned forest, we perform a `RandomizedSearchCV` over a reduced grid of settings:

$$n\_\text{estimators} \in \{100, 150, 200\}, \quad \texttt{max\_depth} \in \{\texttt{None}, 8, 12\},$$

$$\texttt{min\_samples\_split} \in \{2, 10, 50\}, \quad \texttt{min\_samples\_leaf} \in \{5, 20, 50\}, \quad \texttt{max\_features} \in \{\sqrt{p}, 0.5p\},$$

where p is the number of predictors. We draw 8 random combinations from this grid and, for each, perform three-fold cross-validation on the training data using the negative mean Poisson deviance as the scoring criterion. The model with the lowest cross-validated Poisson deviance is selected as the CV-tuned Random Forest and refitted on the full training set with exposure weights. For this final model we report both the OOB R^2 and the out-of-sample performance on the held-out test set.

Gradient Boosted Trees

As a second non-linear benchmark we use gradient-boosted decision trees tailored to Poisson outcomes. Rather than relying solely on external libraries such as XGBoost or LightGBM, we adopt the histogram-based gradient boosting implementation `HistGradientBoostingRegressor` from `scikit-learn` with a Poisson deviance loss. This framework is optimised for large tabular datasets, natively supports Poisson loss, and integrates smoothly with the existing Python preprocessing pipeline.

As with the Random Forest, we model policy-level claim frequency $\lambda_i = Y_i/e_i$ and supply the exposure e_i as a sample weight. The model is therefore trained to minimise a weighted Poisson deviance between observed and predicted frequencies, with higher-exposure policies receiving greater influence in the loss function.

Baseline specification. The baseline gradient-boosted Poisson model uses `loss="poisson"` and combines a sequence of shallow regression trees. The main hyperparameters are chosen to balance flexibility and regularisation:

- **Learning rate** (`learning_rate`): set to 0.05, providing a conservative step size so that each individual tree makes a small adjustment to the current prediction, reducing the risk of overfitting.
- **Number of iterations** (`max_iter`): set to 200, corresponding to at most 200 boosting stages (trees).
- **Tree complexity**: we restrict depth to `max_depth = 3` and impose `min_samples_leaf = 50` to ensure that each leaf is supported by a reasonably large number of policies, which smooths the fitted function.
- **Regularisation**: an ℓ_2 penalty on leaf values (`l2_regularization`) provides additional shrinkage of extreme predictions.

The algorithm also uses validation-based early stopping (`early_stopping=True`) with a small validation fraction. If the validation Poisson deviance does not improve for a fixed number of iterations, boosting terminates early, providing an automatic convergence criterion and an additional safeguard against overfitting.

Hyperparameter tuning and training procedure. To refine the baseline settings, we again employ a `RandomizedSearchCV` over a restricted set of hyperparameters:

$$\texttt{learning\_rate} \in \{0.03, 0.05, 0.10\}, \quad \texttt{max\_iter} \in \{100, 150, 200\},$$

$$\texttt{max\_depth} \in \{2, 3, 4\}, \quad \texttt{min\_samples\_leaf} \in \{20, 50, 100\},$$

$$\texttt{max\_leaf\_nodes} \in \{15, 31, 63\}, \quad \texttt{l2\_regularization} \in \{0.0, 0.5, 1.0, 2.0\}.$$

We randomly sample 10 combinations from this grid and perform three-fold cross-validation using the negative mean Poisson deviance as the scoring metric, again with exposure used as a sample weight. The configuration with the best cross-validated Poisson deviance is selected as the CV-tuned gradient-boosted Poisson model and refitted on the full training set. Early stopping ensures that the number of effective boosting iterations is determined adaptively by the validation performance, so the final model is typically simpler than the nominal `max_iter` would suggest.

In summary, the gradient-boosted Poisson trees provide a flexible semi-parametric alternative to the GLM and Random Forest, combining Poisson-appropriate loss, exposure weighting, and strong regularisation through shallow trees, shrinkage, and early stopping.

Model Comparison

Evaluation metrics

To assess and compare the performance of the different models, we focus on out-of-sample prediction of claim *frequency* on the test set. Let

$$\lambda_i = \frac{\text{ClaimNb}_i}{\text{Exposure}_i}$$

already done.

denote the empirical claim frequency for policy i, and $\widehat{\lambda}_i$ the corresponding model prediction. We consider the following metrics.

Mean squared error (MSE). The mean squared error of predicted frequencies is

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}\left(\widehat{\lambda}_i - \lambda_i\right)^2,$$

which penalises larger errors more heavily.

Mean absolute error (MAE). The mean absolute error is

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}\left|\widehat{\lambda}_i - \lambda_i\right|,$$

providing a scale-based measure of average absolute deviation between predicted and observed frequencies.

Poisson deviance. Because the underlying data are claim counts, we also use the Poisson deviance as a natural discrepancy measure. For non-negative observations y_i and strictly positive predictions $\hat{y}_i$, the Poisson deviance is defined as

$$D_{\text{Pois}}(y, \hat{y}) = 2\sum_{i=1}^{n}\left[y_i \log\left(\frac{y_i}{\hat{y}_i}\right) - (y_i - \hat{y}_i)\right],$$

earlier

where in our case $y_i = \lambda_i$ and $\hat{y}_i = \widehat{\lambda}_i$. Smaller values correspond to a better fit to the observed frequencies.

Out-of-bag R^2 (Random Forest). For the Random Forest models, we additionally report the out-of-bag (OOB) coefficient of determination R^2_{OOB}, which is an internal cross-validation estimate based on predictions for observations not included in the bootstrap sample of each tree.

Predictive performance

Table 14 summarises the test-set performance of all models in terms of RMSE, MAE and Poisson deviance for claim frequency. For the Random Forest variants we additionally report the out-of-bag R^2 as an internal validation measure.

Table 14: Test-set performance metrics for claim frequency models.

Model	RMSE	MAE	Poisson deviance	OOB R^2
Poisson GLM	2.0149	0.1912	1.0155	–
Random Forest (baseline)	2.0127	0.1938	1.0117	0.00093
Random Forest (CV-tuned)	2.0136	0.1913	0.9956	0.00094
HistGBM (Poisson, baseline)	2.0143	0.1911	1.0057	–
HistGBM (Poisson, CV-tuned)	2.0142	0.1910	1.0024	–
XGBoost (Poisson)	2.8307	0.5695	1.3982	–
LightGBM (Poisson)	2.9316	0.5660	1.3908	–

The Poisson GLM provides a strong baseline, with a deviance of approximately 1.02. The tree-based approaches based on Random Forests and histogram gradient boosting deliver modest but consistent improvements in deviance: the CV-tuned Random Forest achieves the lowest deviance (0.996), followed closely by the CV-tuned histogram gradient boosting model (1.002). In contrast, the RMSE and MAE of these models are very similar to the GLM, indicating that, in terms of average prediction error on the frequency scale, the gain from moving to more flexible methods is relatively small.

The out-of-bag R^2 values for the Random Forest models are very close to zero (around 0.001), suggesting that only a small fraction of the variation in individual policy frequencies is explained beyond the global mean once sampling variability is taken into account. This is consistent with the low overall claim frequencies and the high noise level at the policy level.

The XGBoost and LightGBM Poisson models perform noticeably worse on this dataset, with substantially higher RMSE, MAE and Poisson deviance. This likely reflects the greater sensitivity of these frameworks to hyperparameter choices and to the way exposure and frequency are incorporated; under the limited tuning budget used here, the simpler histogram-based gradient boosting implementation in `scikit-learn` produced more stable and accurate results.

Overall, the results indicate that the Poisson GLM already captures much of the signal in the data, while carefully tuned tree-based methods can achieve small additional gains in Poisson deviance at the expense of increased model complexity and training time.

Random Forest interpretability: feature importance and tree structure

An advantage of Random Forests is the ability to extract measures of variable importance. We compute impurity-based feature importance scores from the CV-tuned Random Forest and rank predictors by their contribution to reducing node impurity across the forest. The most important predictors are reported in Table 15.

Table 15: Top predictors by Random Forest feature importance.

Predictor	**Importance**
BonusMalus	0.5281
DrivAge	0.1425
Density	0.0905
VehAge	0.0721
VehPower	0.0365
VehBrand dummies (combined)	0.0552
Area_C + Area_D + Area_E	0.0134
VehGas_Regular	0.0110
Region_Rhone-Alpes	0.0093
Region_Languedoc-Roussillon	0.0090
Region_Ile-de-France	0.0020

The importance scores confirm that the bonus–malus level (`BonusMalus`) is by far the dominant predictor, accounting for more than half of the total impurity reduction. Driver age (`DrivAge`), population density (`Density`) and vehicle age (`VehAge`) also contribute substantially, which is consistent with actuarial intuition: young or very old drivers, newer vehicles and urban areas tend to be associated with higher risk. Vehicle power and fuel type (`VehPower`, `VehGas`) play a secondary but non-negligible role.

Although each individual vehicle-brand and geographical dummy has a relatively small importance, their combined contribution is meaningful. Aggregating all vehicle-brand indicators yields an importance of about 0.055, comparable to the effect of vehicle power, while the combined importance of the area dummies `Area_C`, `Area_D` and `Area_E` is around 0.013. Certain specific regions such as `Region_Rhone-Alpes` and `Region_Languedoc-Roussillon` stand out slightly within the regional factors, whereas `Region_Ile-de-France` has a relatively modest marginal importance once the other covariates are included.

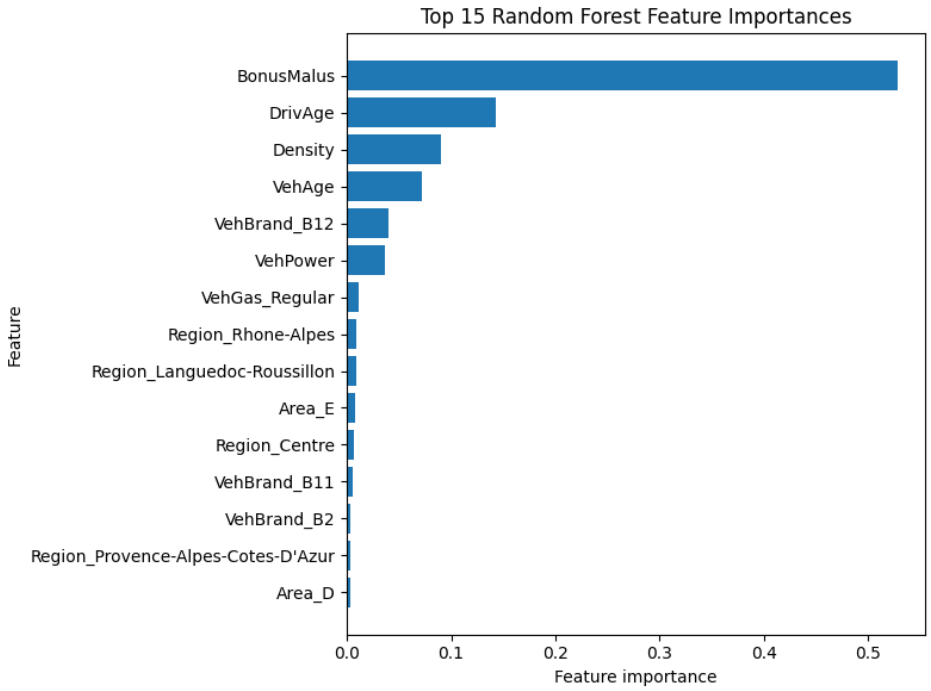

Figure 12: Top 15 feature importance scores for the CV-tuned Random Forest.

For additional interpretability, we also visualise a single representative tree from the forest. This highlights how key predictors such as `Density`, `BonusMalus` and `DrivAge` interact in the tree structure.

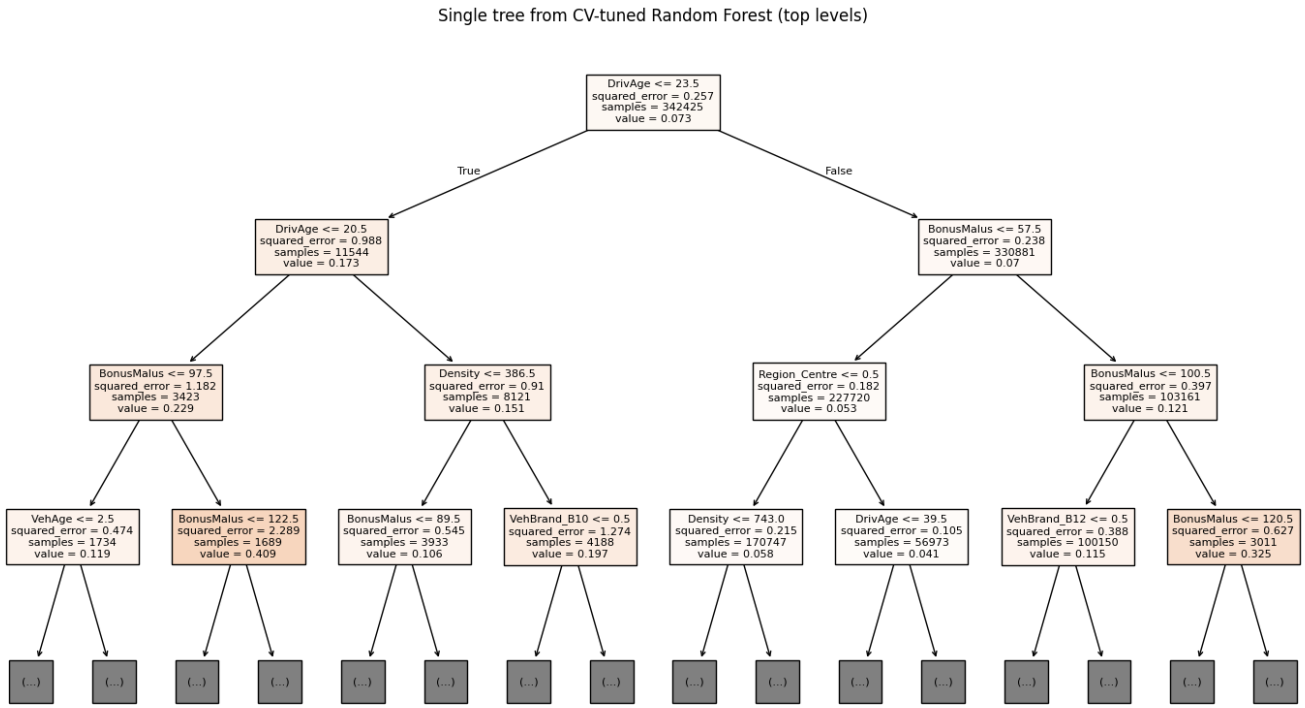

Figure 13: Illustrative tree highlighting key splits in the Random Forest model.

The structure of this tree highlights several important patterns. The root node splits on `DrivAge`, indicating that driver age is a key discriminator: younger drivers (those aged under 23.5) exhibit substantially higher predicted claim frequencies. Conditional on age, the `BonusMalus` score governs most subsequent splits, with higher malus classes consistently leading to markedly higher predicted frequencies. Population `Density` provides further refinement, especially among younger drivers, capturing the elevated risk typical of high-traffic urban environments. Regional and vehicle-brand indicators tend to appear only at deeper nodes, confirming their secondary importance relative to the main demographic and risk classification variables.

Overall, the tree-based visualisations reinforce the insights provided by the feature importance analysis: claim frequency risk is primarily driven by Bonus–Malus level, driver age and population density, with the remaining predictors contributing smaller but still informative adjustments in specific segments of the portfolio.

Training time and scalability

Finally, we compare the computational cost of each approach. For each model we record the approximate wall-clock training time on the full training set (about 5.4×10^5 policies) and provide qualitative comments on scalability. Table 16 summarises these results.

Table 16: Approximate training times and scalability comments.

Model	Training time	Scalability / comments
Poisson GLM	≈ 3 s	Fast IRLS on sparse design matrix
Random Forest (baseline)	≈ 7 min	100 deep trees, OOB enabled; slow for large n
Random Forest (CV-tuned)	≈ 39 min	8 candidates $\times$ 3-fold CV; computationally heavy
HistGBM (Poisson)	≈ 30 s	Histogram-based, typically faster than RF
HistGBM (Poisson, CV-tuned)	≈ 5 min	10 candidates $\times$ 3-fold CV; still cheaper than RF CV
XGBoost (Poisson)	≈ 20 s	Efficient but sensitive to hyperparameters; higher memory use
LightGBM (Poisson)	≈ 21 s	Good scalability with many dummies; requires careful regularisation

Qualitatively, the Poisson GLM trains almost instantaneously and scales well to even larger portfolios, making it attractive for frequent re-fitting or real-time pricing. Random Forests provide improved flexibility but become costly as both the sample size and the number of dummy-encoded predictors increase, and careful control of tree depth and forest size is required to maintain tractable runtimes. Histogram-based gradient boosting and modern libraries such as XGBoost and LightGBM offer a favourable trade-off between flexibility and speed on large tabular datasets, especially when combined with early stopping and appropriate regularisation.

Conclusion and recommendations

This study compared several modelling approaches for predicting claim frequency in a large French motor insurance portfolio. The benchmark Poisson GLM and GAM from Part 1 established a strong parametric baseline, with the GAM providing modest improvements by capturing non-linear effects in the continuous predictors. In Part 4 the analysis was extended to a range of machine-learning methods, including Random Forests, histogram-based gradient boosting, XGBoost and LightGBM.

Across all models, differences in RMSE and MAE were small, reflecting the high noise level inherent in policy-level claim frequencies. The most discriminating metric was the Poisson deviance. The CV-tuned Random Forest achieved the lowest deviance (0.996), followed closely by the CV-tuned histogram gradient boosting model (1.002). Both methods delivered only modest improvements relative to the Poisson GLM (1.015), indicating that a large part of the signal is already captured by a linear specification with log-link and exposure offset. The gradient-boosting implementations in XGBoost and LightGBM performed substantially worse on this dataset, likely due to the instability of their Poisson objectives, sensitivity to hyperparameters, and the large number of one-hot encoded features.

From an interpretability perspective, the GLM remains the most transparent model and may therefore be preferable in pricing or regulatory settings. The Random Forest retains partial interpretability through feature importance and tree visualisations, which highlight the dominant role of `BonusMalus`, `DrivAge` and `Density`. Histogram gradient boosting offers a favourable balance between flexibility and computational efficiency, training much faster than Random Forests while achieving similar predictive accuracy.

Overall, the CV-tuned Random Forest emerged as the best-performing model in terms of Poisson deviance, providing the strongest predictive accuracy while still offering a reasonable degree of interpretability. However, the relatively small performance gap between this model and the Poisson GLM suggests that the choice of model should depend on operational considerations. For applications requiring transparency, stability and ease of deployment, the Poisson GLM remains a competitive option. When non-linear effects and interactions are expected to be material, the Random Forest or histogram-based gradient boosting are more suitable alternatives.

Future work could include the use of exposure-weighted GAMs with spline interactions, regularised GLMs to stabilise estimation in high dimensions, or mixed-effects models to account for unobserved heterogeneity at the policy or regional level. Additional data on driving behaviour, telematics, socio-economic indicators or traffic patterns could further enhance predictive power. Feature engineering such as risk-score construction or clustering of categorical levels may also benefit the performance of tree-based methods.

Part 5: Comparative Analysis of GAMs and Random Forests

The goal of this section is to compare GAMs and Random Forest models by their assumptions, interaction between predictors, data prepossessing requirements, how models can be interpreted and how models can be regularized. GAMs are a sort of extension of GLMs, so they share some of the same assumptions. So we also assume that the response variable given the predictors follows a distribution from the exponential family.

Analysis of Assumptions: Linearity and Additivity

Generalized Additive Models (GAMs) rely on several structural assumptions. The first assumption is the additivity of predictors. This assumption states that the relationship between the predictors X and the response Y can be modeled as the sum of smooth functions of the predictors:

$$g\big(\mathbb{E}[Y \mid \mathbf{X}]\big) = \beta_0 + f_1(X_1) + f_2(X_2) + \cdots + f_p(X_p),$$

where $g(\cdot)$ is the link function and $f_j(\cdot)$ are smooth functions.The reason this is possible is provided by the Kolmogorov-Arnold representation theorem, introduced in Module 5 of our course, which states that "any continuous function can be represented as a sum of compositions of smooth functions" [2]. This assumes that interactions between predictors are either negligible or explicitly included as bivariate smooth terms. GAMs also assume that the smooth functions are sufficiently regularized to avoid overfitting. GAMs are an extension GLMs therefore they also share some of their assumptions. For example GAMs also assume that the errors are independent and identically distributed and they assume that the response variable is distributed according to one of the exponential distribution family. GAMs do not assume linearity, as they can be used to predict non linear data. These assumption help GAMs to be interpretable.

In contrast, Random Forests make far fewer structural assumptions. They are a nonparametric model, meaning they don't assume any distribution for the response. Random Forest models are ensembles of decision trees that can capture complex, nonlinear relationships and high-order interactions automatically. Random Forests do not assume additivity, linearity, or any specific functional form between predictors and the response. They only require that the training data is representative of the underlying distribution and that the features contain useful signal. In comparison to GAMs which are already pretty flexible, Random Forest models are even more flexible.

When applying these two models to data that has a linear relationship, the GAM would perform better. This is because Random Forest models do not assume any specific distribution of the response, or any specific relationship between the predictors and the response. However a linear model is a very special case of a GAM. It is the case where the smooth additive functions are just linear functions, and the link function is just an identity link. Meaning that the response is equal to the sum of linear models of the predictors. This would better capture a linear relationship, then a random forest model. To illustrate this, a simulated example will be used. We simulated a simple linear dataset with one predictor X and a response y. The simulation was defined as:

$$X_i \sim \text{Uniform}(-5, 5), \quad i = 1, \ldots, n$$

$$y_i = 2X_i + 3 + \epsilon_i$$

where the noise term $\epsilon_i \sim \mathcal{N}(0, 1)$ is independent and normally distributed, and n is the number of observations. In this setup, the true relationship between X and y is linear with a slope of 2 and an intercept of 3, and the random noise adds variability to the observed responses. We then fit a GAM with an identity link function and linear smoothing functions(Linear Regression) and a random forest model.

The models were evaluated on a test set of the same simulated data. The GAM obtained a test MSE of 0.8563 and the Random Forest model had a test MSE of 1.5874. The GAM had a smaller MSE as it assumed a linear relationship between the repsonse and predictor based off the choice of an identity link function and linear smoothing functions. The RF model was also overfit on the training set. This can be seen in Figure 14. The GAM generalizes much better to the new data, whereas the RF was fit too closely to the training data and is wiggling back and forth.

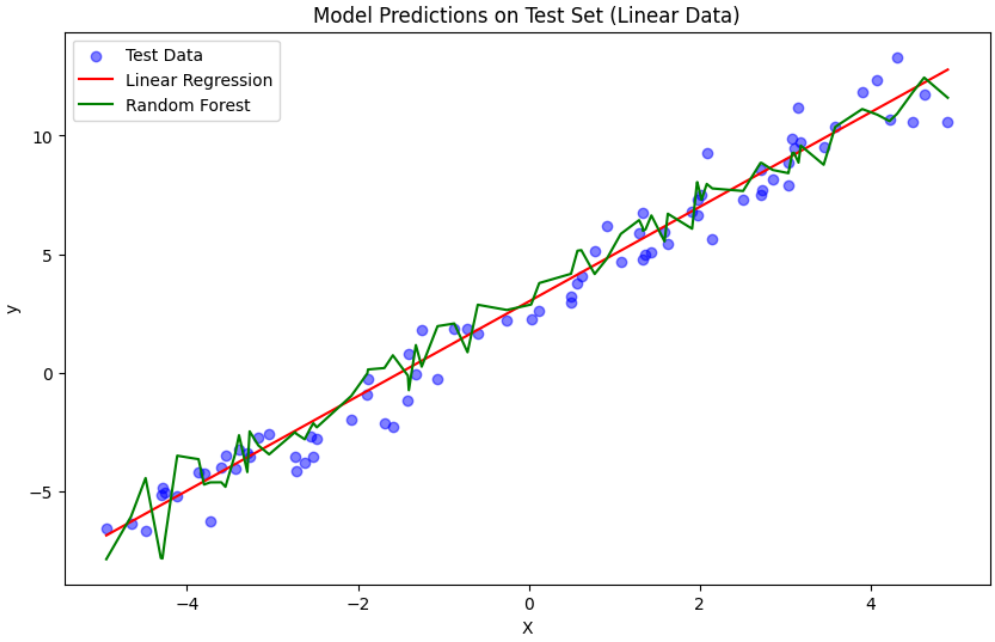

Figure 14: GAM vs RF on Linear Data

Interactions Between Predictors

Generalized Additive Models (GAMs) handle interactions between predictors by explicitly including **bivariate or higher-order smooth terms** in the model. In a standard GAM, the response is modeled as a sum of smooth functions of individual predictors:

$$g(\mathbb{E}[Y \mid X_1, \ldots, X_p]) = \beta_0 + f_1(X_1) + f_2(X_2) + \cdots + f_p(X_p),$$

which assumes additivity. To capture interactions, a term like $f_{12}(X_1, X_2)$ can be included, where f_{12} is a smooth function of the two predictors jointly. This allows GAMs to model nonlinear interactions in a controlled and interpretable way, but such interactions must be specified in advance. For example in part 2 of this homework, a bivarite smoothing function was included between the predictors age and duration.

In contrast, Random Forests inherently account for interactions through their tree-based structure. Each tree splits the feature space based on different predictors, and subsequent splits can depend on previous splits, automatically capturing complex interactions of any order. Unlike GAMs, Random Forests do not require the user to pre-specify interactions and can flexibly model nonlinear, high-order dependencies. However, this flexibility comes at the cost of interpretability, as the individual effects of predictors are harder to isolate.

Analysis of Data Preprocessing Requirements

Sensitivity to Feature Scaling Generalized Additive Models (GAMs) in general are sensitive to the scale of predictors when regularization or penalized smooth functions are used. For example, if one predictor has a much larger range than another, the smoothness penalty may disproportionately affect the larger-scale variable, potentially distorting the estimated functions. Standardizing or normalizing predictors can help ensure that the smoothness penalties are applied consistently and that the model fits are not biased by the scale of the variables. In contrast, Random Forests are largely insensitive to feature scaling because they are based on splitting the feature space at thresholds. Whether a feature ranges from 0 to 1 or 0 to 1000 does not change the tree's ability to find optimal splits.

Highly correlated predictors affect GAMs and Random Forests differently. In GAMs, multicollinearity can make it difficult to disentangle the individual effects of predictors because the estimated smooth functions may become unstable or poorly identified. This can reduce interpretability and increase variance in the estimated effects. Random Forests, on the other hand, are more robust to multicollinearity in terms of predictive performance: correlated predictors may be used interchangeably across different trees, so the overall prediction accuracy remains largely unaffected. However, feature importance measures in Random Forests can be biased in the presence of highly correlated variables, making it harder to assess the true contribution of each predictor.

0.1 Analysis of Interpretability an Regularization

In terms of Interpretability, GAMs are much easier to interpret than Random Forest models. When a Gam is fit the individual smoothing functions can be graphed. This can visualize the individual effect a predictor variable has on the response. —- Add some more (https://ecogambler.netlify.app/blog/interpreting-gams/) Random Forests, in contrast, are more difficult to interpret directly because predictions are aggregated over many decision trees. Interpretation tools for Random Forests typically include feature importance metrics and to estimate the contribution of each predictor, but these are less intuitive than the explicit functional form provided by GAMs.

Regularization in GAMs is typically achieved through penalization of the smooth functions. Each smooth term $f_i(X_i)$ is estimated with a penalty on its wiggliness (e.g., using a roughness penalty or a smoothing parameter λ), which prevents overfitting by controlling the flexibility of the function. In Random Forests, regularization is achieved implicitly through model averaging and limiting tree complexity. By building many trees on bootstrap samples and considering only a random subset of predictors at each split, Random Forests reduce variance and overfitting, and additional hyperparameters such as `max_depth` or `min_samples_leaf` further constrain the flexibility of the trees.

In Conclusion, GAMs are very powerful models, that can be used in a variety of scenario's, however they must meet certain assumptions. GAMs are also also easier to interpret. Random Forest models are very powerful models, and can be fit onto all sorts of data. However it is much harder to interpret the affects each predictor has on a result. Based off these conclusions, these two models are better fits for different scenarios. For a scenario in which decisions need to be justified an interpretable model such as a GAM is much better suited. Fo r eampl an insurance pricing model that is under specific regulation much be able to justify why prices are set at each rate. Therefore it is imperitive that an interpretative model is used. For a non scenario where the prediction is the most important result and the interpret ability is less important, random forest models are much better suited. One example of this would be in fraud detection. A Random forest model could be used to identify potential fraud, and then they could be investigated further. But the precision of that initial prediction is much more important then knowing why it predicted fraud.

The choice between GAMs and Random Forests depends on the trade-off between interpretability and predictive power. GAMs are particularly well-suited for contexts that require **transparent, regulatory-driven modelling** such as pricing in insurance, credit scoring, or compliance reporting, because the smooth functions provide clear explanations of predictor effects. Random Forests excel in **complex prediction tasks** where capturing nonlinearities and high-order interactions is critical, such as fraud detection or operational risk modeling, where predictive accuracy is prioritized over interpretability. In practice, GAMs offer a balance of flexibility and clarity, while Random Forests prioritize performance at the expense of transparency.

Credit Author Statement and Use of AI Tools

Below are our contributions to this homework;

Hewei Ding - Parts 4

Julian Schady - Parts 2 and 5

Justice Kelvin Kwadzo Dzameshie - Parts 1 and 3

Use of AI Tools (ChatGPT)
We used AI assistants only to accelerate drafting and code templating; skeletons for data-loading and plotting to suggest debugging tips, and to help with language polish. All code was executed locally, outputs were independently verified against our data, and any AI-generated text was edited for accuracy and specificity. No synthetic data or undisclosed model outputs were used. Final methodological decisions and all interpretations are the authors' own.

Appendix

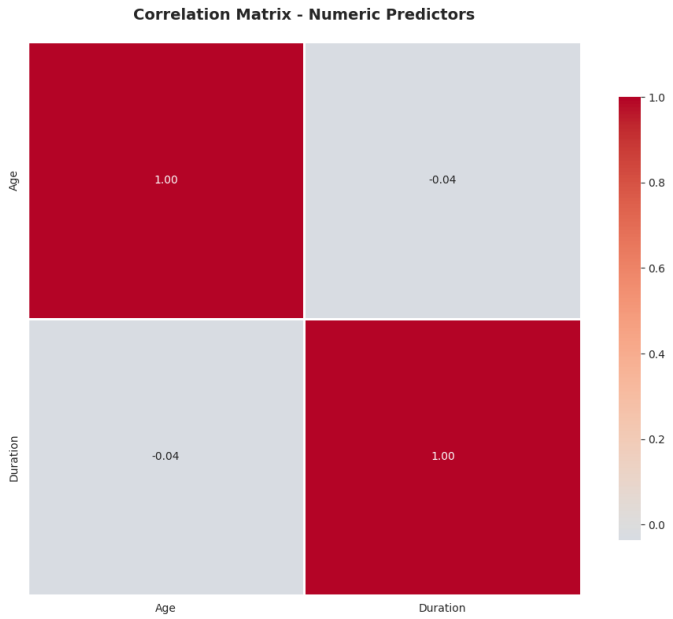

Figure 15: Heat Map of Age and Duration

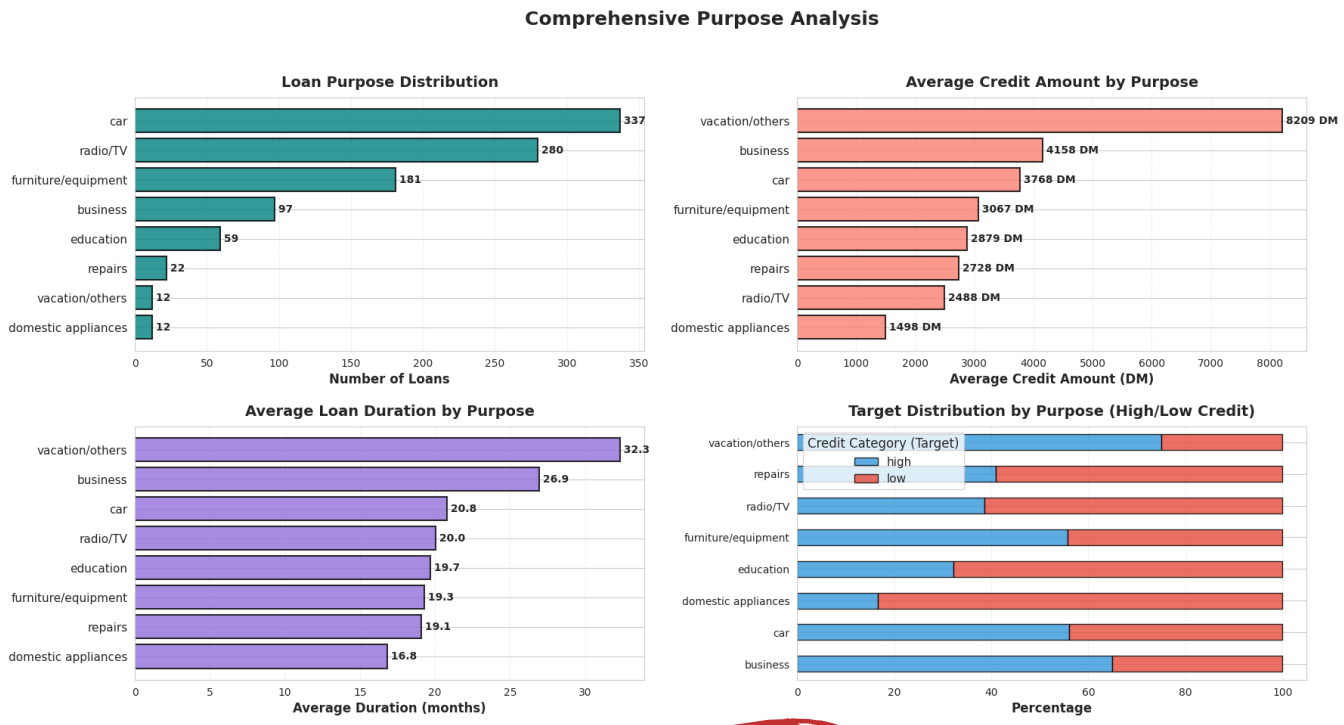

Figure 16: Enter Caption

Table 17: Poisson GLM coefficient estimates for claim frequency

Term	Coef	Std. err	z	p-value	CI 0.025	CI 0.975
Intercept	-3.7596	0.040	-93.334	0.000	-3.839	-3.681
C(VehBrand)[T.B10]	0.0406	0.036	1.128	0.259	-0.030	0.111
C(VehBrand)[T.B11]	0.1204	0.039	3.104	0.002	0.044	0.197
C(VehBrand)[T.B12]	0.1609	0.017	9.283	0.000	0.127	0.195
C(VehBrand)[T.B13]	0.0501	0.041	1.231	0.218	-0.030	0.130
C(VehBrand)[T.B14]	-0.1514	0.078	-1.931	0.053	-0.305	0.002
C(VehBrand)[T.B2]	-0.0191	0.015	-1.251	0.211	-0.049	0.011
C(VehBrand)[T.B3]	0.0020	0.022	0.091	0.927	-0.041	0.045
C(VehBrand)[T.B4]	-0.0337	0.030	-1.134	0.257	-0.092	0.025
C(VehBrand)[T.B5]	0.0705	0.025	2.846	0.004	0.022	0.119
C(VehBrand)[T.B6]	-0.0443	0.028	-1.562	0.118	-0.100	0.011
C(VehGas)[T.Regular]	0.0609	0.011	5.602	0.000	0.040	0.082
C(Region)[T.R21]	0.0814	0.082	0.998	0.318	-0.078	0.241
C(Region)[T.R22]	0.0426	0.052	0.823	0.411	-0.059	0.144
C(Region)[T.R23]	-0.0836	0.061	-1.373	0.170	-0.203	0.036
C(Region)[T.R24]	0.0054	0.024	0.224	0.823	-0.042	0.053
C(Region)[T.R25]	-0.0274	0.045	-0.611	0.541	-0.115	0.061
C(Region)[T.R26]	-0.0719	0.049	-1.475	0.140	-0.167	0.024
C(Region)[T.R31]	-0.1018	0.035	-2.893	0.004	-0.171	-0.033
C(Region)[T.R41]	-0.2480	0.045	-5.462	0.000	-0.337	-0.159
C(Region)[T.R42]	-0.0058	0.089	-0.065	0.948	-0.180	0.168
C(Region)[T.R43]	-0.1376	0.134	-1.027	0.304	-0.400	0.125
C(Region)[T.R52]	-0.0288	0.030	-0.949	0.343	-0.088	0.031
C(Region)[T.R53]	0.0472	0.029	1.644	0.100	-0.009	0.104
C(Region)[T.R54]	-0.0859	0.039	-2.228	0.026	-0.161	-0.010

Table 18: Poisson GAM coefficient estimates for claim frequency (factor effects only)

Term	Coef	Std. err	z	p-value	CI 0.025	CI 0.975
C(VehBrand)[B1]	-0.3972	0.074	-5.392	0.000	-0.542	-0.253
C(VehBrand)[B10]	-0.3742	0.080	-4.687	0.000	-0.531	-0.218
C(VehBrand)[B11]	-0.3023	0.081	-3.733	0.000	-0.461	-0.144
C(VehBrand)[B12]	-0.2704	0.073	-3.681	0.000	-0.414	-0.126
C(VehBrand)[B13]	-0.3517	0.082	-4.300	0.000	-0.512	-0.191
C(VehBrand)[B14]	-0.5737	0.103	-5.551	0.000	-0.776	-0.371
C(VehBrand)[B2]	-0.4112	0.074	-5.585	0.000	-0.555	-0.267
C(VehBrand)[B3]	-0.3907	0.075	-5.213	0.000	-0.538	-0.244
C(VehBrand)[B4]	-0.4342	0.077	-5.614	0.000	-0.586	-0.283
C(VehBrand)[B5]	-0.3312	0.076	-4.367	0.000	-0.480	-0.183
C(VehBrand)[B6]	-0.4461	0.077	-5.788	0.000	-0.597	-0.295
C(VehGas)[T.Regular]	0.0616	0.011	5.596	0.000	0.040	0.083
C(Region)[T.R21]	0.1512	0.082	1.852	0.064	-0.009	0.311
C(Region)[T.R22]	0.0818	0.052	1.577	0.115	-0.020	0.183
C(Region)[T.R23]	-0.0525	0.061	-0.863	0.388	-0.172	0.067
C(Region)[T.R24]	0.1035	0.025	4.205	0.000	0.055	0.152
C(Region)[T.R25]	0.0300	0.045	0.666	0.505	-0.058	0.118
C(Region)[T.R26]	0.0206	0.049	0.419	0.675	-0.076	0.117
C(Region)[T.R31]	-0.0846	0.035	-2.400	0.016	-0.154	-0.016
C(Region)[T.R41]	-0.2063	0.046	-4.527	0.000	-0.296	-0.117
C(Region)[T.R42]	0.0206	0.089	0.232	0.817	-0.154	0.195
C(Region)[T.R43]	-0.0623	0.134	-0.464	0.642	-0.325	0.201
C(Region)[T.R52]	0.0130	0.030	0.428	0.669	-0.047	0.073
C(Region)[T.R53]	0.1156	0.029	3.994	0.000	0.059	0.172
C(Region)[T.R54]	-0.0215	0.039	-0.555	0.579	-0.097	0.054
C(Region)[T.R72]	-0.0584	0.034	-1.727	0.084	-0.125	0.008
C(Region)[T.R73]	-0.0982	0.043	-2.281	0.023	-0.183	-0.014
C(Region)[T.R74]	0.2117	0.066	3.212	0.001	0.082	0.341
C(Region)[T.R82]	0.1004	0.024	4.208	0.000	0.054	0.147
C(Region)[T.R83]	-0.2817	0.076	-3.729	0.000	-0.430	-0.134
C(Region)[T.R91]	-0.0136	0.033	-0.415	0.678	-0.078	0.050
C(Region)[T.R93]	0.0008	0.025	0.033	0.974	-0.049	0.050
C(Region)[T.R94]	0.1348	0.067	2.012	0.044	0.003	0.266

Table 19: Poisson GAM spline basis coefficient estimates for continuous predictors

Term	Coef	Std. err	z	p-value	CI 0.025	CI 0.975
BonusMalus_s0	-1.454e-14	2.43e-15	-5.994	0.000	-1.93e-14	-9.79e-15
BonusMalus_s1	-2.167e-14	3.66e-15	-5.921	0.000	-2.88e-14	-1.45e-14
BonusMalus_s2	-0.9592	0.088	-10.932	0.000	-1.131	-0.787
BonusMalus_s3	-1.0879	0.100	-10.840	0.000	-1.285	-0.891
BonusMalus_s4	-0.4306	0.088	-4.886	0.000	-0.603	-0.258
BonusMalus_s5	-0.3266	0.100	-3.267	0.001	-0.523	-0.131
BonusMalus_s6	-0.2440	0.107	-2.279	0.023	-0.454	-0.034
BonusMalus_s7	5.6036	0.465	12.054	0.000	4.692	6.515
BonusMalus_s8	-6.8381	1.302	-5.250	0.000	-9.391	-4.285
Density_s0	0.0108	0.108	0.100	0.920	-0.201	0.222
Density_s1	0.1150	0.070	1.637	0.102	-0.023	0.253
Density_s2	0.1248	0.084	1.483	0.138	-0.040	0.290
Density_s3	0.1237	0.078	1.594	0.111	-0.028	0.276
Density_s4	0.2241	0.080	2.804	0.005	0.067	0.381
Density_s5	0.2707	0.078	3.449	0.001	0.117	0.424
Density_s6	0.2069	0.120	1.723	0.085	-0.028	0.442
Density_s7	0.2596	0.185	1.404	0.160	-0.103	0.622
Density_s8	0.2563	0.087	2.949	0.003	0.086	0.427
VehAge_s0	-1.3930	0.078	-17.906	0.000	-1.546	-1.241
VehAge_s1	-1.0838	0.061	-17.867	0.000	-1.203	-0.965
VehAge_s2	-1.2536	0.045	-27.728	0.000	-1.342	-1.165
VehAge_s3	-1.0345	0.035	-29.455	0.000	-1.103	-0.966
VehAge_s4	-1.0893	0.032	-34.250	0.000	-1.152	-1.027
VehAge_s5	-1.1099	0.027	-41.853	0.000	-1.162	-1.058
VehAge_s6	-3.2752	0.177	-18.549	0.000	-3.621	-2.929
VehAge_s7	0.7746	1.025	0.756	0.450	-1.235	2.784
VehAge_s8	-2.4840	1.011	-2.456	0.014	-4.466	-0.502
DrivAge_s0	-0.6037	0.119	-5.071	0.000	-0.837	-0.370
DrivAge_s1	-0.9355	0.066	-14.141	0.000	-1.065	-0.806
DrivAge_s2	-0.5822	0.080	-7.323	0.000	-0.738	-0.426
DrivAge_s3	-0.4267	0.070	-6.124	0.000	-0.563	-0.290
DrivAge_s4	-0.1001	0.073	-1.364	0.173	-0.244	0.044
DrivAge_s5	-0.3331	0.070	-4.733	0.000	-0.471	-0.195
DrivAge_s6	-0.4459	0.097	-4.603	0.000	-0.636	-0.256
DrivAge_s7	0.0732	0.153	0.478	0.632	-0.227	0.373
DrivAge_s8	-0.3031	0.275	-1.103	0.270	-0.842	0.236

References

[1] UCI Machine Learning Repository. *Statlog (German Credit Data) Dataset*. University of California, Irvine. Available at the UCI Machine Learning Repository: `https://archive.ics.uci.edu/dataset/144/statlog+german+credit+data`.

[2] Christopher Blier-Wong. *Module 5: Smoothing and Generalized Additive Models*. Lecture notes for STA2536: Statistical Learning, University of Toronto, Department of Statistical Sciences, 2025.

[3] Christopher Blier-Wong. *Module 6: Tree-based Methods*. Lecture notes for STA2536: Data Science for Risk Modelling, University of Toronto, Department of Statistical Sciences, 2025.