Assignment 1 - Machine Learning Methods and Data Analytics

Roberto Carminati

July 17, 2024

Contents

1 Introduction

In this report, we aim to develop a predictive model for the frequency of claims in a motor insurance portfolio. The dataset provided includes the feature vector $\mathbf{x}$, exposure volume v_i, and the claims count $N_i \sim \text{Poisson}(v_i\lambda(\mathbf{x_i}))$ for each policy. The feature vector consists of five covariates: car weight (x_1), annually driven distance (x_2), age of the driver (x_3), age of the car (x_4), and gender of the driver (x_5). The primary objective is to estimate $\lambda(\mathbf{x})$, the expected claim frequency, using various statistical and machine learning techniques.

2 Data Description

The dataset "A2Dataset14.csv" contains the following variables:

- **Car Weight (x_1)**: Weight of the car.
- **Annually Driven Distance (x_2)**: Distance driven annually.
- **Age of Driver (x_3)**: Age of the driver.
- **Age of Car (x_4)**: Age of the car.

- **Gender of Driver** (x_5): Gender of the driver (binary).
- **Exposure Volume** (v_i): Volume or exposure.
- **Claims Count** (N_i): Number of claims.

We plot the histograms of the numerical variables ("Sex" excluded) and we note that it seems realistic that "Counts" follows a Poisson distribution. (Figure 1)

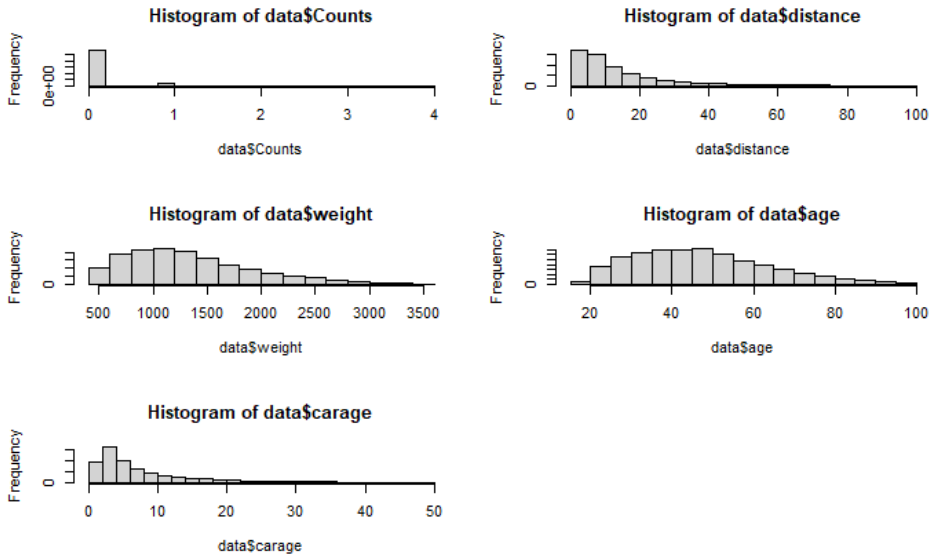

Figure 1: Histograms

We don't have highly correlated variable as we can see from the correlation plot. (Figure 2)

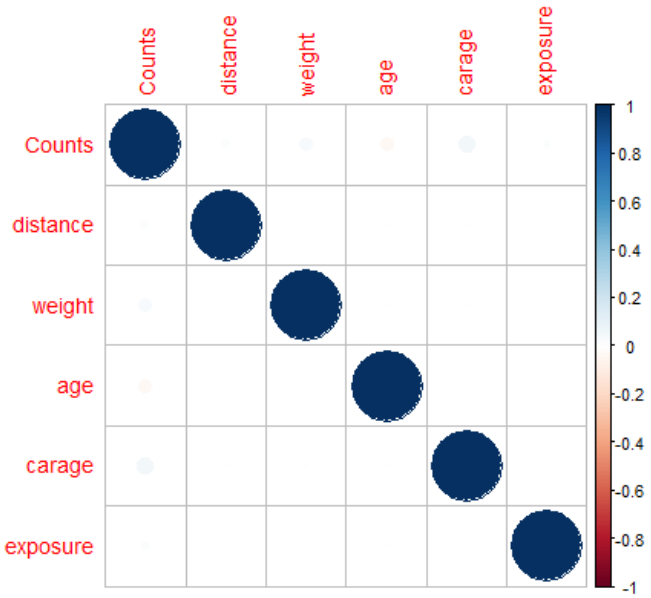

Figure 2: Correlation Plot

3 Models

3.1 Poisson Regression Model

We fit a Poisson Generalized Linear Model for the estimation of $\lambda(x)$ considering linear, quadratic and mixed terms:

$$\begin{aligned}\log(\lambda) = \beta_0 &+ \beta_1 \text{weight} + \beta_2 \text{distance} + \beta_3 \text{age} + \beta_4 \text{carage} + \beta_5 \text{sex} \\ &+ \beta_6 \text{weight}^2 + \beta_7 \text{distance}^2 + \beta_8 \text{age}^2 + \beta_9 \text{carage}^2 \\ &+ \beta_{10}(\text{weight} \cdot \text{distance}) + \beta_{11}(\text{weight} \cdot \text{age}) + \beta_{12}(\text{weight} \cdot \text{carage}) \\ &+ \beta_{13}(\text{distance} \cdot \text{age}) + \beta_{14}(\text{distance} \cdot \text{carage}) + \beta_{15}(\text{age} \cdot \text{carage}) \\ &+ \log(\text{exposure})\end{aligned}$$

While doing diagnostic of our model, we see that the mixed terms are not significant, the linear term are all extremely significant. Looking at the quadratic terms, age^2 is extremely significant, weight^2 just at level 5%, while distance^2 and carage^2 are not significant.

We also performed a backwards selection with Bayesian Information Criterion BIC to penalize more the coefficients. Following this approach we decide to remove also weight^2, which was not bringing a big improvement even in deviance.

So we compute our final GLM model :

$$\begin{aligned}\log(\lambda) = \beta_0 &+ \beta_1 \text{weight} + \beta_2 \text{distance} + \beta_3 \text{age} + \beta_4 \text{carage} + \beta_5 \text{sex} + \\ &\beta_6 \text{age}^2 + \log(\text{exposure}).\end{aligned}$$

The regression coefficients for the selected model terms are reported.(Table 1)

Additionally, the estimated $\lambda(x)$ for the benchmark settings (**Weight** = 1000, **Distance** = 10, **Age** = 27, **CarAge** = 5, **Sex** = male) is 3.328414. And in (Figure 3) we can see the different values of $\lambda(x)$ vs **Age** when the other variables are set to the benchmark.

Table 1: Regression Coefficients for Poisson Regression Model

Variable	**Coefficient**
(Intercept)	-2.331262
distance	0.003941
weight	0.000222
sexmale	-0.068362
carage	0.020286
age	-0.019445
I(ageˆ2)	0.000114

3.2 Poisson Regression Tree

We compute a model with a minimum of 8000 observations in any terminal node, the maximum depth of any node of the final tree as 8. Then, we performed a 10-fold cross

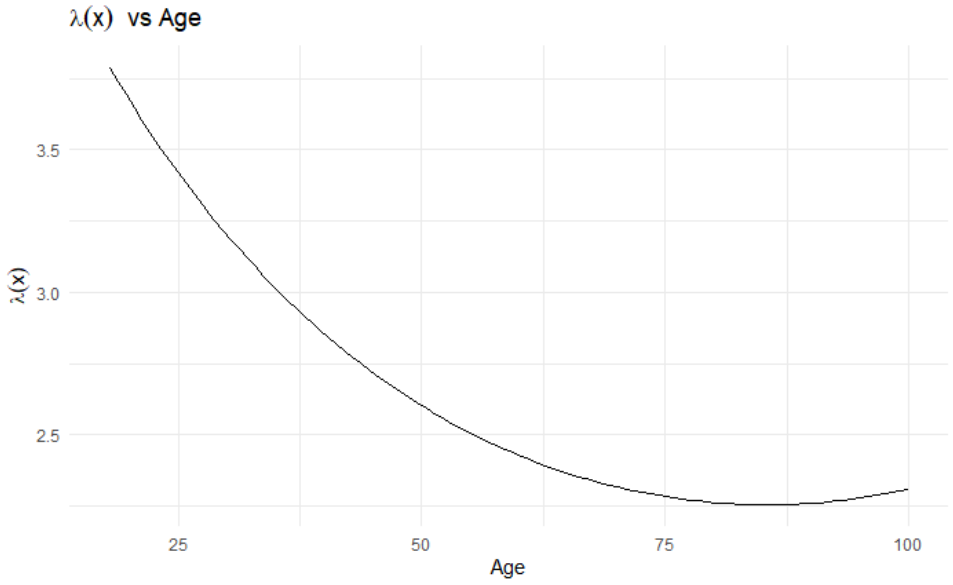

Figure 3: Plot of GLM $\lambda(x)$ versus Age when other predictors are $Weight = 1000$, $Distance = 10$, $CarAge = 5$, $x_5 =$ male.

validation to tune the complexity parameters and, using the one standart error rule, we found the optimal value and we use it to prune our tree to a smaller and simplier one. (Figure 4)

The final tree structure is visualized in (Figure 4). The estimated $\lambda(x)$ for the benchmark settings is 1.086871 and is plotted vs **Age**. (Figure 5)

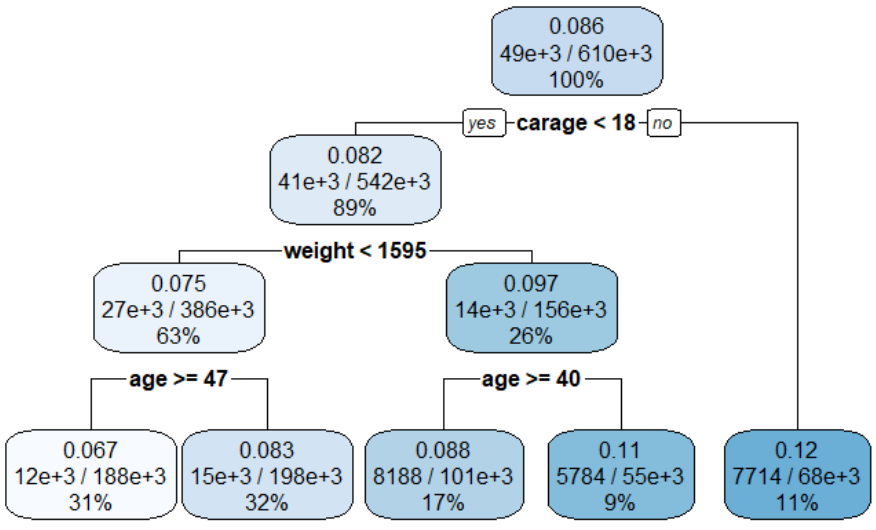

Figure 4: Optimal Poisson Regression Tree structure

3.3 Poisson Boosting Tree (No Base Model)

Now, we split the data into training and test sets to tune parameters using the Poisson boosting tree method. We conduct a grid search with "shrinkage" $[0.3, 0.5, 0.7]$ and

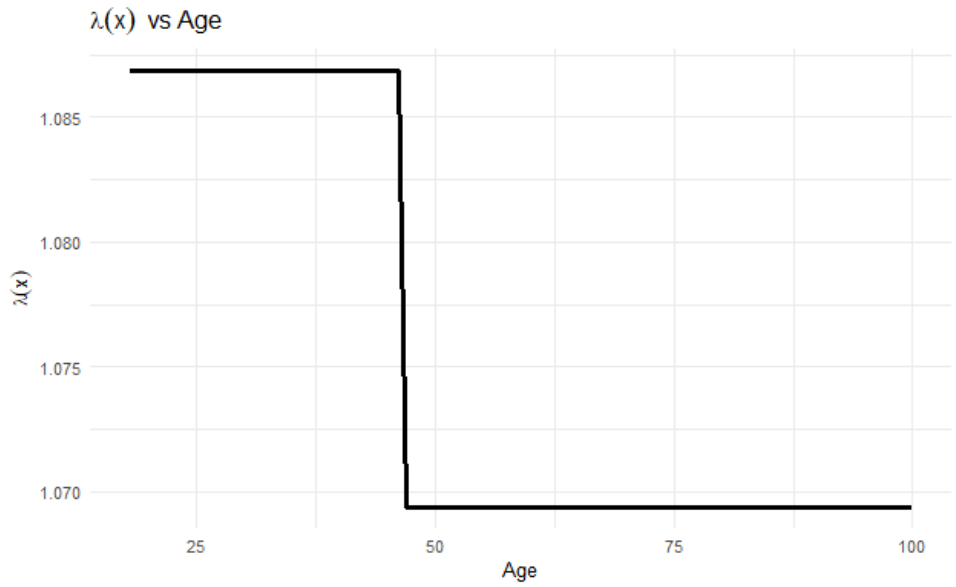

Figure 5: Plot of regression tree $\lambda(x)$ versus Age when other predictors are $Weight = 1000$, $Distance = 10$, $CarAge = 5$, $x_5 =$ male.

"maximum depth" $[2, 4]$: for each parameter combination, we calculate the validation error over 5 iterations and record the optimal number of trees and the validation errors.

Now we determine the combination of parameters that minimize the average validation error. The lowest average error achieved is 0.4076747, with an average number of trees of 33.6.

Next, we build the boosting tree model without a base model using the optimal parameters identified: "shrinkage" $= 0.3$ and "maximum depth" $= 2$. Initially, we assess the model with 50 trees to validate our findings and then we refine the model using the number of trees that minimizes the out-of-sample error, which is 37. (Figure 6)

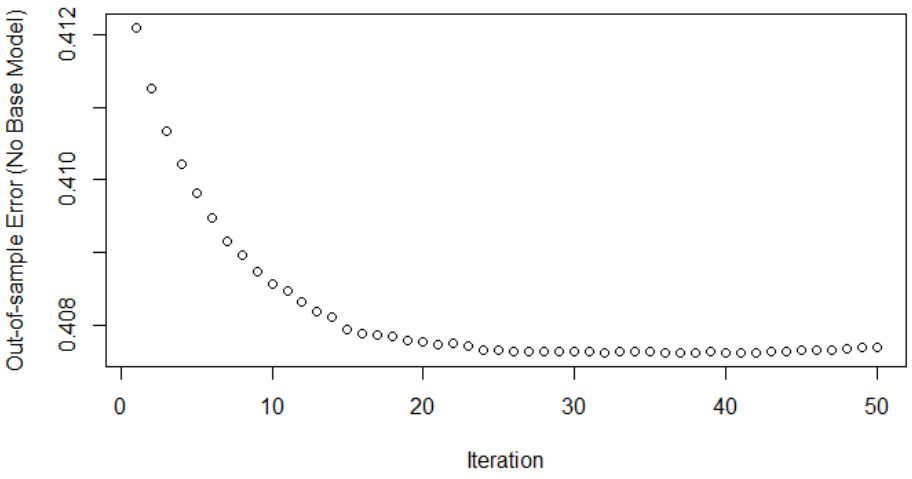

Figure 6: Boosting Poisson regression tree with "shrinkage" $= 0.3$ and "maximum depth" $= 2$ vs different numbers of trees

Using the Poisson boosting tree method with no base model the estimated $\lambda(x)$ for the benchmark settings is 0.06851 and is plotted vs **Age**. (Figure 7)

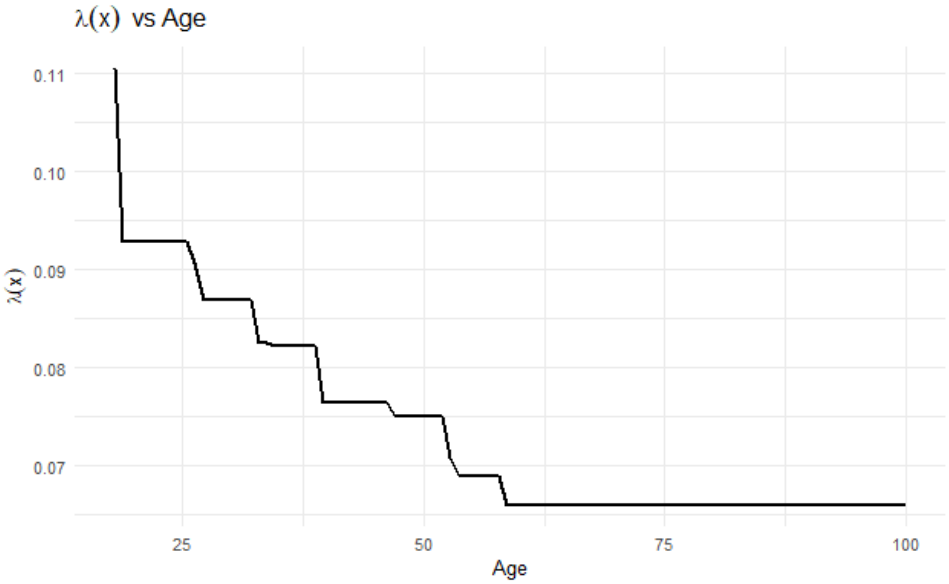

Figure 7: Plot of Boosting Poisson Regression Tree $\lambda(x)$ versus Age when other predictors are $Weight = 1000$, $Distance = 10$, $CarAge = 5$, $x_5 =$ male.

3.4 Poisson Boosting Tree (With Base Model)

Now, we repeat the same operations using the Poisson boosting tree method but with a base model established by a generalized linear model (GLM) fitted earlier in 3.1.

Utilizing a grid search with "shrinkage" $[0.3, 0.5, 0.7]$ and "maximum depth" $[2, 4]$, we evaluate each parameter combination by computing the validation error over 5 iterations. This process allows us to identify the optimal combination of parameters corresponding to the minimum average validation errors. ("shrinkage" = 0.3) and "maximum depth" = 4

After analyzing the average out-of-sample errors, we determine the parameters that yield the lowest average error of 0.4099607, along with an average number of trees amounting to 21.

Subsequently, we construct the boosting tree model with a base model derived from the GLM. We start by verifying our findings with an initial assessment using 50 trees and proceed to refine the model using 32 trees, which minimizes the out-of-sample error. (Figure 8)

This iterative approach ensures that both the base model from the GLM and the subsequent boosting tree model are finely tuned to predict motor insurance claim frequencies accurately.

Using the Poisson boosting tree method with as base model the generalized linear model (GLM) fitted earlier in 3.1, the estimated $\lambda(x)$ for the benchmark settings is 0.07976411 and is plotted vs **Age**. (Figure 9)

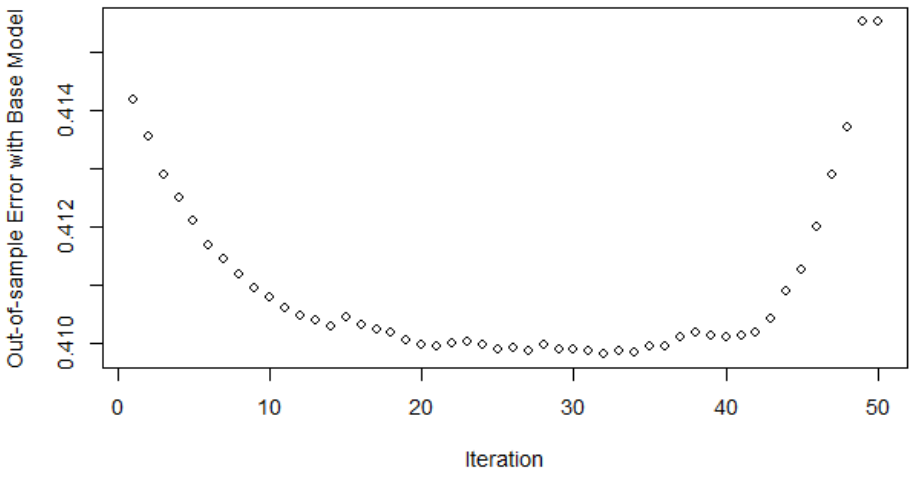

Figure 8: Boosting Poisson regression tree with base model with "shrinkage" = 0.3 and "maximum depth" = 2 vs different numbers of trees

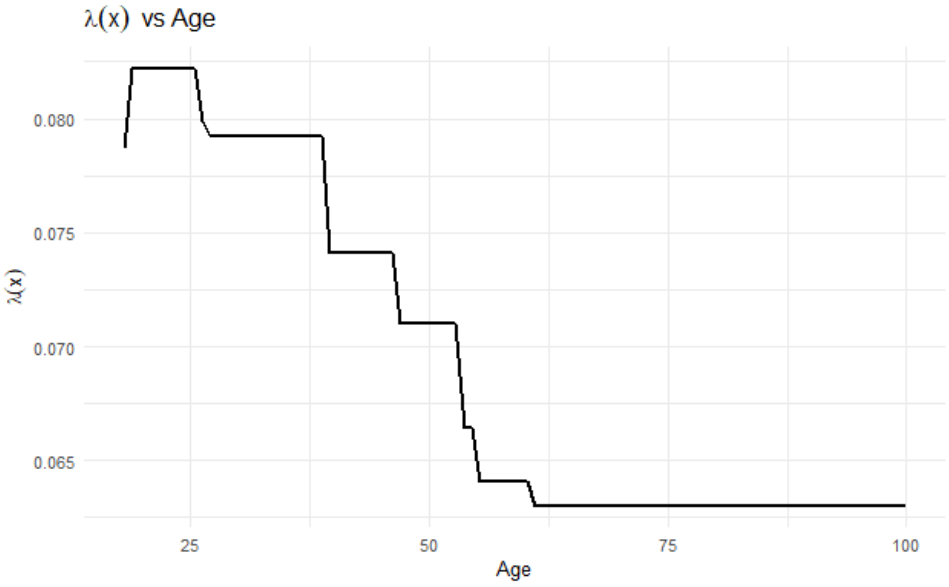

Figure 9: Plot of Boosting Poisson Regression Tree with Base Model $\lambda(x)$ versus Age when other predictors are $Weight = 1000$, $Distance = 10$, $CarAge = 5$, $x_5 =$ male.

3.5 Neural Network Model

In addition, we estimate $\lambda(x)$ using a one-layer Neural Network model with 10 neurons (Figure 10), implemented in TensorFlow. This model accounts for the fact that exposure is treated as a given factor. For simplicity, we use only two integer covariates, Age and Distance.

Initially, we train the neural network over 70 epochs to visually tune the number of epochs based on the validation error. We 20 epochs the error seems to not decrease anymore. In the next part of the project, when comparing our models using 10-fold cross-validation, we use the value of 20 epochs. This choice strikes a good balance between

```
Model: "model_13"
__________________________________________________________________________________________
 Layer (type)                Output Shape              Param #   Connected to
==========================================================================================
 Design (InputLayer)         [(None, 2)]               0         []
 Layer1 (Dense)              (None, 10)                30        ['Design[0][0]']
 Network (Dense)             (None, 1)                 11        ['Layer1[0][0]']
 LogVol (InputLayer)         [(None, 1)]               0         []
 multiply_13 (Multiply)      (None, 1)                 0         ['Network[0][0]',
                                                                  'LogVol[0][0]']
==========================================================================================
Total params: 41
Trainable params: 41
Non-trainable params: 0
__________________________________________________________________________________________
```

Figure 10: Architecture and parameters of our Neural Model

minimizing the validation error and managing the computational cost.(Figure 11)

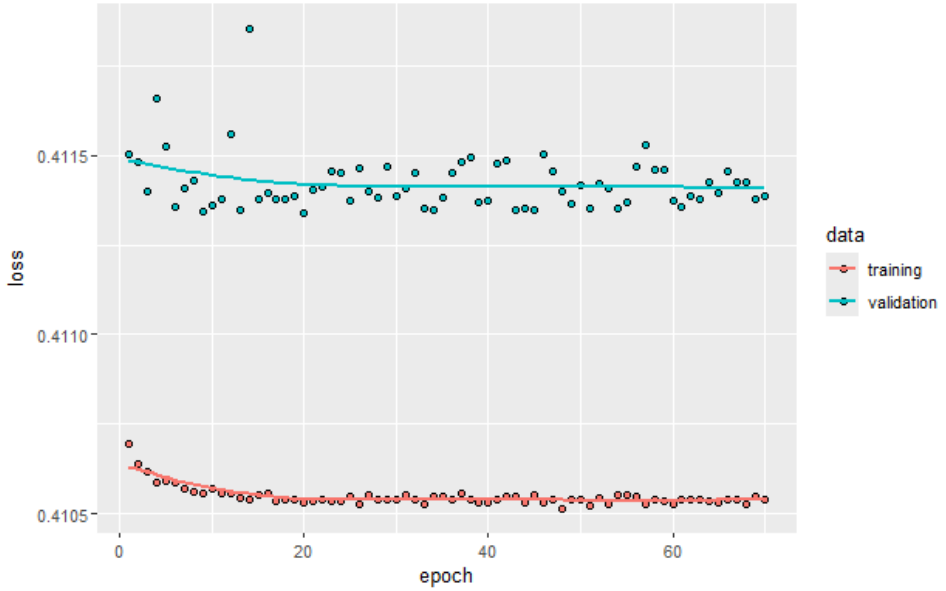

Figure 11: Training and validation epoch all over 70 epochs of our Neural Network with 1 layer and 10 neurons

As a benchmark, we estimate $\lambda(x)$ for $Distance = 10$ and $Age = 27$ obtaining a value of 6354572827. Furthermore, we present a plot of $\lambda(x)$ versus Age while holding $Distance = 10$. This neural network approach offers a different perspective on modeling the expected claim frequency, potentially capturing non-linear relationships between the predictors and the response variable.(Figure 12)

(When used knit to html in R-markdown we occure problem on plotting the (Figure 12), but this can be found in 12302504CarminatiRobertoDataset14.Rmd)

3.6 Comparison of the models

We compare the performance of all our models—fitted in sections a), b), c), d), and e) using 10-fold cross-validation to determine the best model. The models compared are: the generalized linear model (GLM), the Poisson regression tree model, the Poisson boosting tree model without a base model, the Poisson boosting tree model with the GLM as a

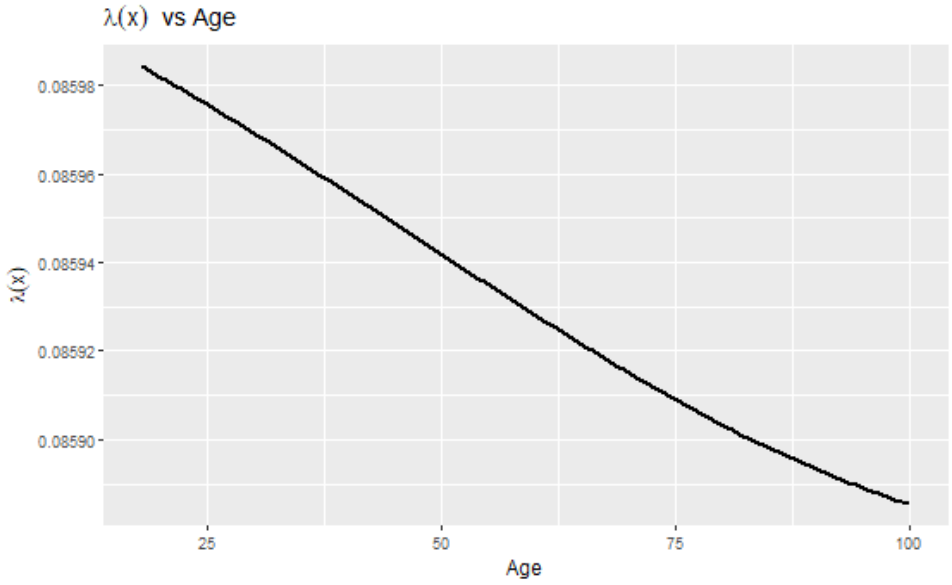

Figure 12: Plot of $\lambda(x)$, estimate with Neural Network, versus Age when other predictors are $Weight = 1000$, $Distance = 10$, $CarAge = 5$, $x_5 =$ male.

base model and the neural network model. The results of the 10-fold cross-validation errors are presented in Table 2.

Table 2: 10-Fold Cross-Validation Errors for Different Models

Model	**10-Fold Cross-Validation Error**
Model_glm	0.4067965
Model_tree	0.4097227
Model_tree_boosting_no_base_model	0.4071204
Model_tree_boosting_base_model	0.4070122
Model_nn	0.4107265

The Poisson regression tree model appears to be too simplistic, as indicated by its relatively higher validation error. Incorporating boosting improves the model's performance, as seen in the lower error of the Poisson boosting tree model without a base model. However, adding a base model to the boosting process does not lead to further improvements. The one-layer neural network model performs worse than the GLM and boosting models, but this result is reasonable given its simple architecture and the use of only two variables. A more complex neural network architecture might yield better results. The GLM shows us the better performance with a validation error of 0.4068.

To compute the out-of-sample error, we use the Poisson deviance, defined as:

$$D(y, \hat{y}) = \frac{2}{n}\left(\sum_{i=1}^{n} y_i \log\left(\frac{y_i}{\hat{y}_i}\right) - \sum_{i=1}^{n}(y_i - \hat{y}_i)\right)$$

where y_i represents the observed counts and $\hat{y}_i$ represents the predicted counts. This function calculates the Poisson deviance, providing a measure of model performance by comparing observed and predicted counts.

When examining the estimated λ versus the age of the customer, we observe a decreasing trend in λ as age increases, which aligns with our expectations. However, at very advanced ages, there is a slight increase in λ, which may initially seem unexpected. This anomaly can be attributed to the smaller number of customers in this age group, resulting in less robust estimates.

Overall, our analysis suggests that older customers tend to have lower frequencies of claims from policies within this portfolio. This pattern is generally consistent with the understanding that older individuals may drive less frequently or may have safer driving habits compared to younger drivers.

3.7 Logistic Regression for High Claims

Furthermore, we continue the analysis by introducing a categorical variable, High, into the dataset. High is defined as "Yes" if the claims count of a policy $N \geq 2$, and "No" otherwise. Initially, we applied a logistic regression model using all available predictors, resulting in a residual deviance of 25648. After we selected the model by including only the significant predictors Weight, Age, CarAge, Sex, and Age^2 obtained a residual deviance of 25682.

We develop a logistic regression classifier to predict the response variable High based on the predictors Weight, Distance, Age, CarAge, Sex and Age^2.

For the covariate benchmark as described in section 3.1 (Weight = 1000, Distance = 10, Age = 27, CarAge = 5, Sex = male), we calculate the predicted probability of High being "Yes" obtaining 0.00289. This probability is relatively low, which aligns with our data characterized by many zeros and ones. (Table 3)

Table 3: Frequencies for Response Variable

Response Variable (High)	**Frequency**
0	563399
1	44661
2	1865
3	72
4	3

Additionally, we plot the probability Pr[High = Yes] versus Age while holding other predictors constant at Weight = 1000, Distance = 10, CarAge = 5, and Sex = male.(Figure 13

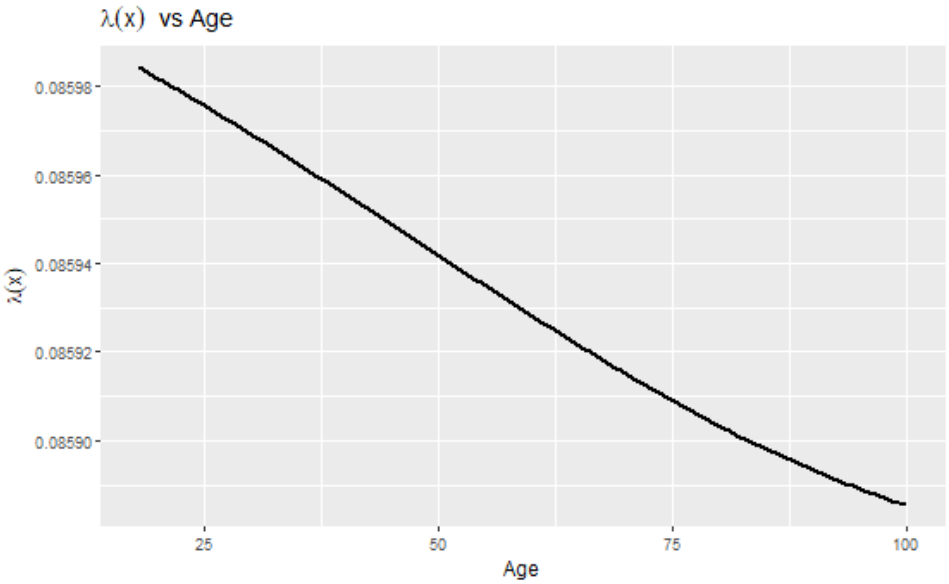

Figure 13: Predicted Probability of High = Yes versus Age.