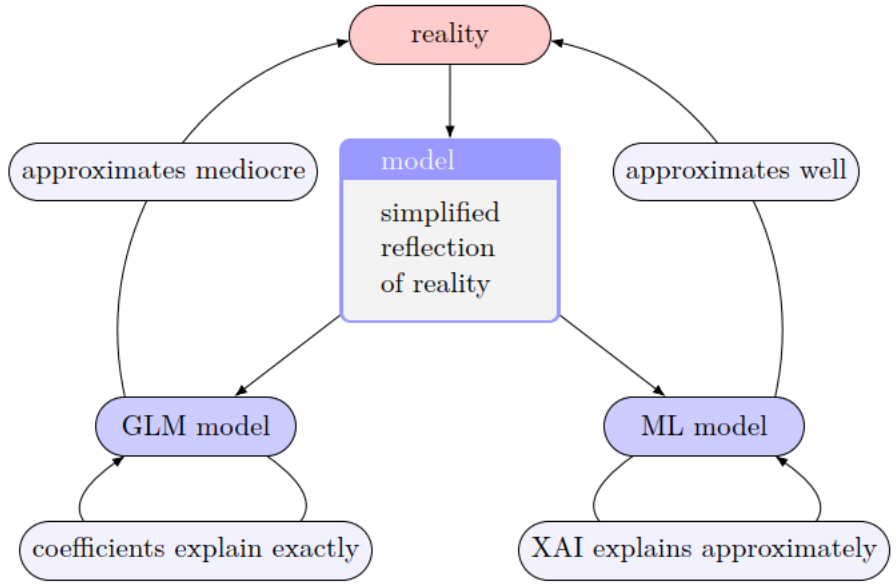
Peeking into the Black Box

An Actuarial Case Study for Interpretable Machine Learning

Christian Lorentzen
christian.lorentzen@mobiliar.ch

Michael Mayer
michael.mayer@mobiliar.ch

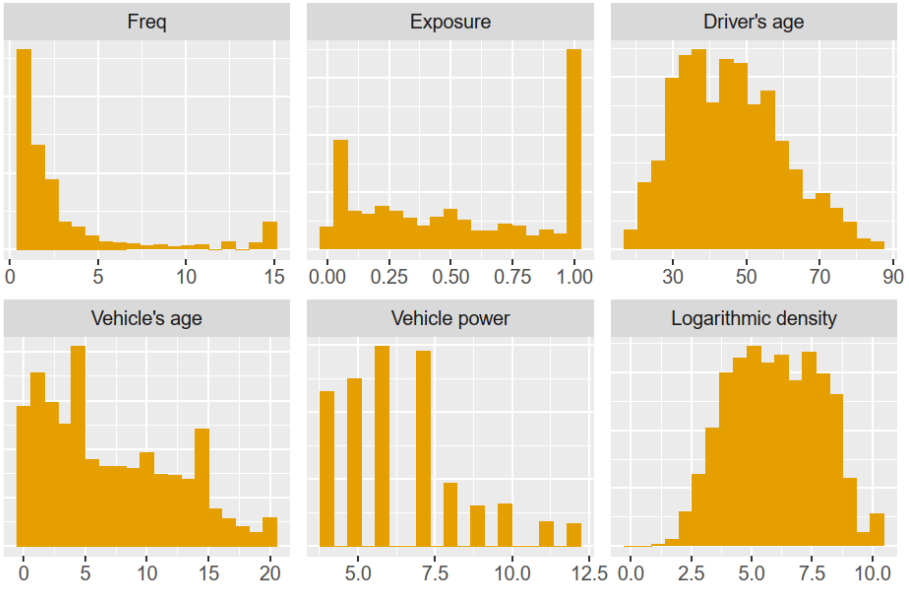
Prepared for:
Fachgruppe “Data Science”
Swiss Association of Actuaries SAV



Data

```
> head(freMTPL2freq, 9)
```

	IDpol	ClaimNb	Exposure	Area	VehPower	VehAge	DrivAge	BonusMalus	VehBrand	VehGas	Density	Region
1	1	1	0.10	D	5	0	55	50	B12	Regular	1217	R82
2	3	1	0.77	D	5	0	55	50	B12	Regular	1217	R82
3	5	1	0.75	B	6	2	52	50	B12	Diesel	54	R22
4	10	1	0.09	B	7	0	46	50	B12	Diesel	76	R72
5	11	1	0.84	B	7	0	46	50	B12	Diesel	76	R72
6	13	1	0.52	E	6	2	38	50	B12	Regular	3003	R31
7	15	1	0.45	E	6	2	38	50	B12	Regular	3003	R31
8	17	1	0.27	C	7	0	33	68	B12	Diesel	137	R91
9	18	1	0.71	C	7	0	33	68	B12	Diesel	137	R91



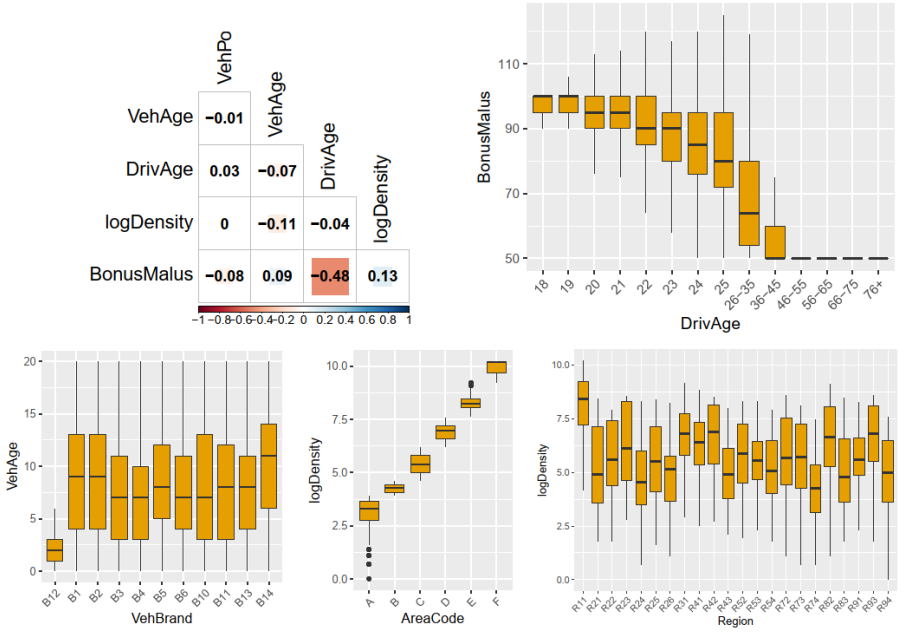


Figure 1: Histograms of selected variables.

Three models

1. Poisson GLM without interactions.
2. Deep neural network fitted by the R interface to Keras and TensorFlow
3. Gradient boosted trees using XGBoost

Global model **agnostic** methods R packages

- Flashlight
- DALEX
- IML

Defining explainers

1

```
# Setting up explainers
>fl_glm <- flashlight(
>  model = fit_glm, label = "GLM",
>  predict_function = function(fit, X) predict(fit, X, type = "response")
>)
```

2

```
>fl_nn <- flashlight(
>  model = fit_nn, label = "NNet",
>  predict_function = function(fit, X)
>    predict(fit, prep_nn_calib(X, x), type = "response")
>)
```

3

```
>fl_xgb <- flashlight(
>  model = fit_xgb, label = "XGBoost",
>  predict_function = function(fit, X) predict(fit, prep_xgb(X, x))
>)
```

4

```
# Combine them and add common elements like reference data
>metrics <- list(‘Average deviance‘ = deviance_poisson,
>                ‘Relative deviance reduction‘ = r_squared_poisson)
>fls <- multiflashlight(list(fl_glm, fl_nn, fl_xgb), data = test,
>                       y = y, w = w, metrics = metrics)

# Version on canonical scale
>fls_log <- multiflashlight(fls, linkinv = log)
```

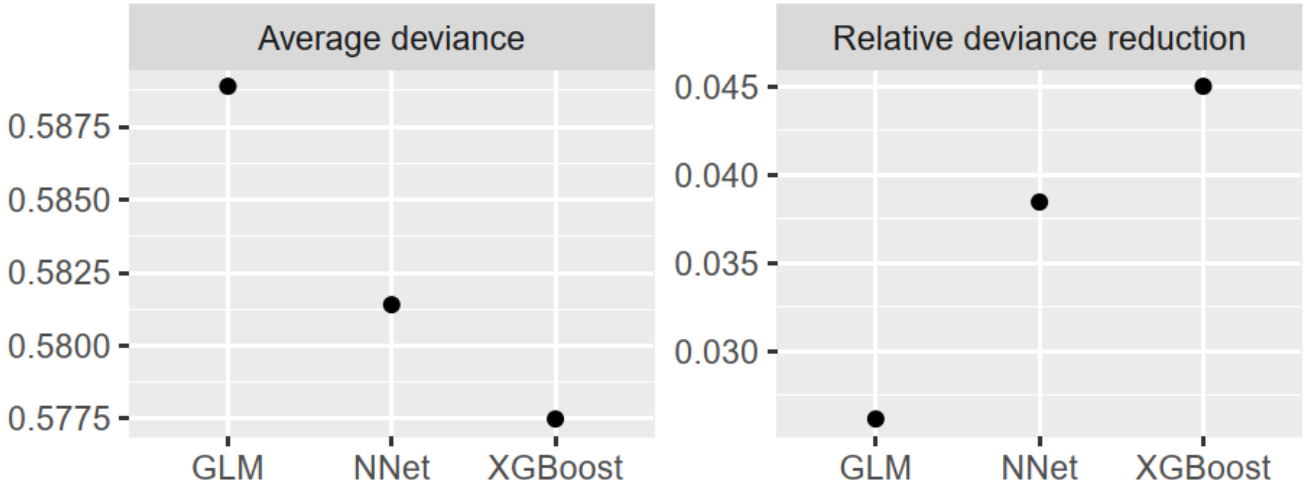
Performance

```
>(perf <- light_performance(fls))

  metric                          value label
1 Average deviance              0.589  GLM
2 Relative deviance reduction   0.0261 GLM

3 Average deviance              0.581  NNet
4 Relative deviance reduction   0.0385 NNet
5 Average deviance              0.577  XGBoost
6 Relative deviance reduction   0.0450 XGBoost

# Plot
>plot(perf, geom = "point") +
>  labs(x = element_blank(), y = element_blank())
```

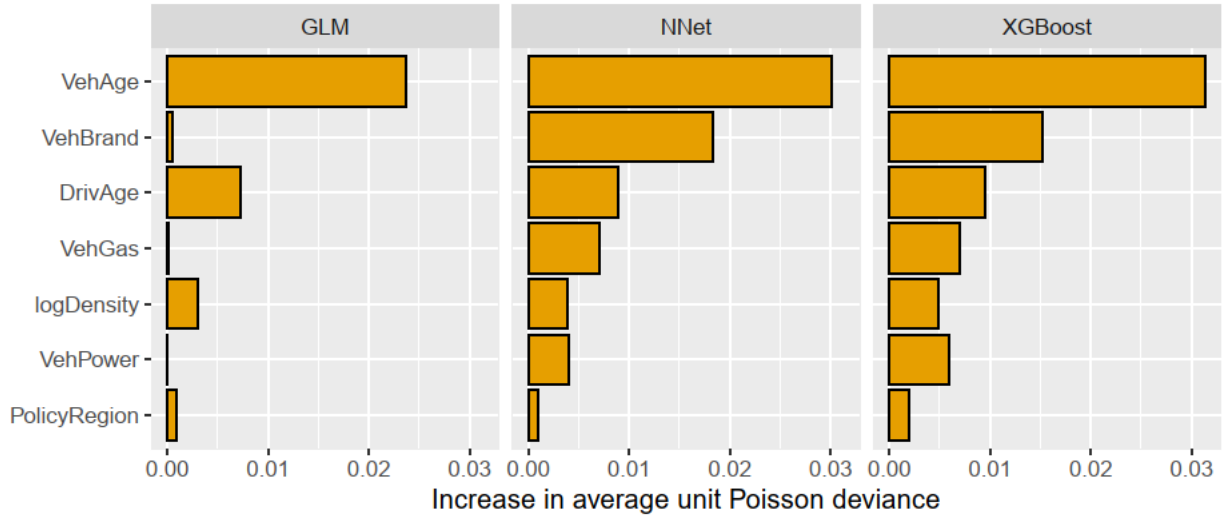


$$D(y, \hat{y}) = \sum_{i=1}^{n} w_i S(y_i, \hat{y}_i) / \sum_{i=1}^{n} w_i$$

$$S(y, \hat{y}) = 2\left(y \log \frac{y}{\hat{y}} - (y - \hat{y})\right)$$

Figure 3: Average unit Poisson deviance (left) and its relative improvement (right). Both scoring functions are evaluated on the 20 % hold-out data.

Variable importance

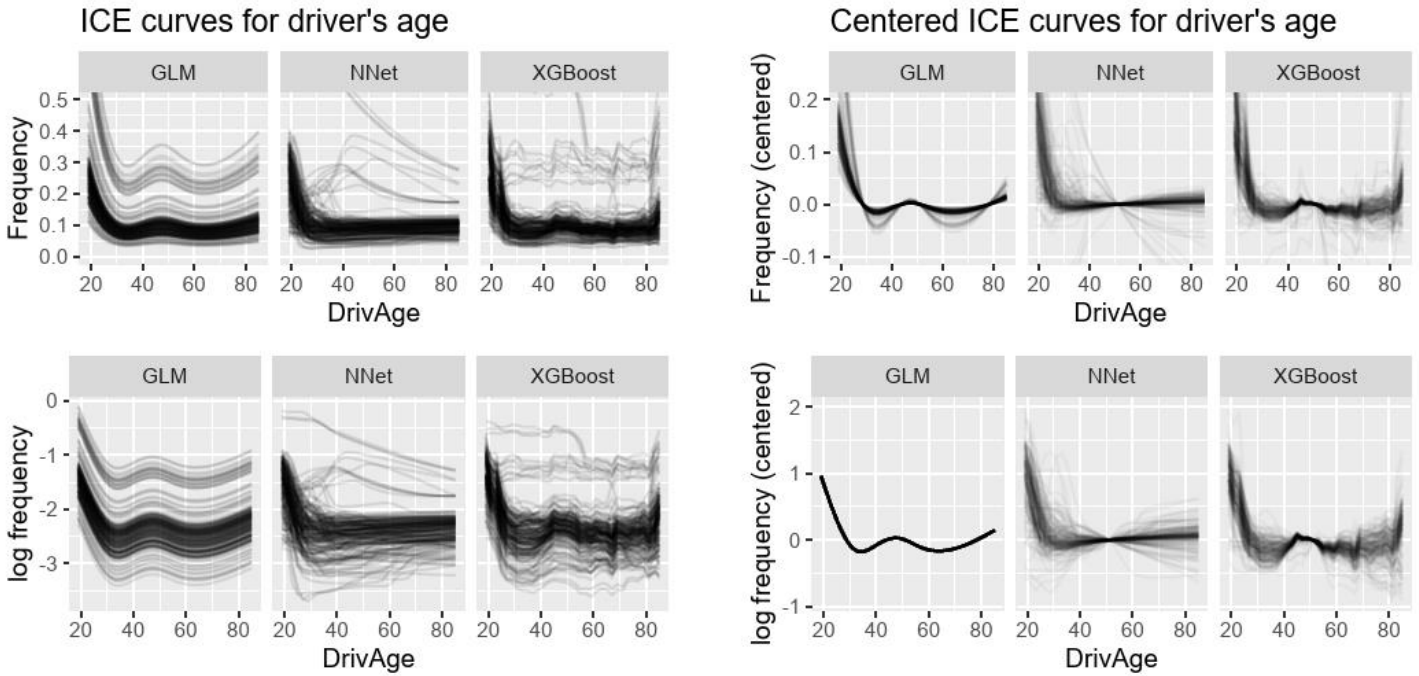



```
>imp <- light_importance(fls, v = x)
>plot(imp, fill = "#E69F00", color = "black")
```

Algorithm 1: Permutation importance

scoreOriginal ← performance on data
for *x in variables* **do**
  dataShuffled ← data with permuted column x
  scoreShuffled ← performance on dataShuffled
  importance[x] ← scoreShuffled − scoreOriginal
end
output : importance

Individual Conditional Expectation (ICE)



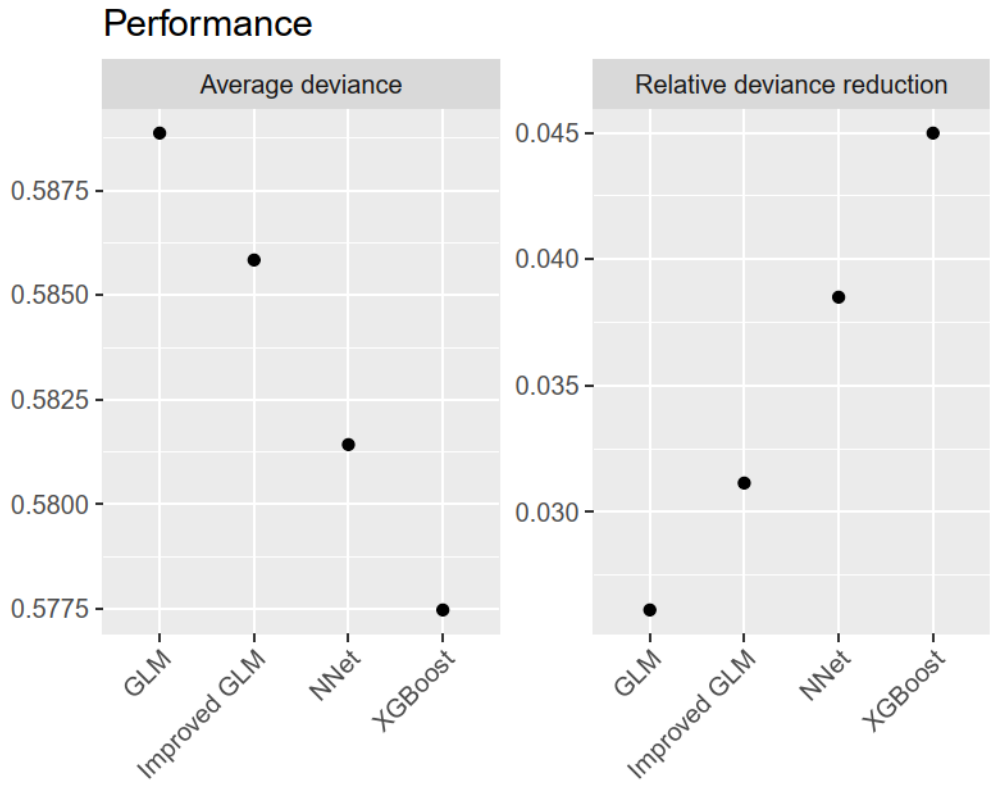
Algorithm 2: ICE for variable x and one observation

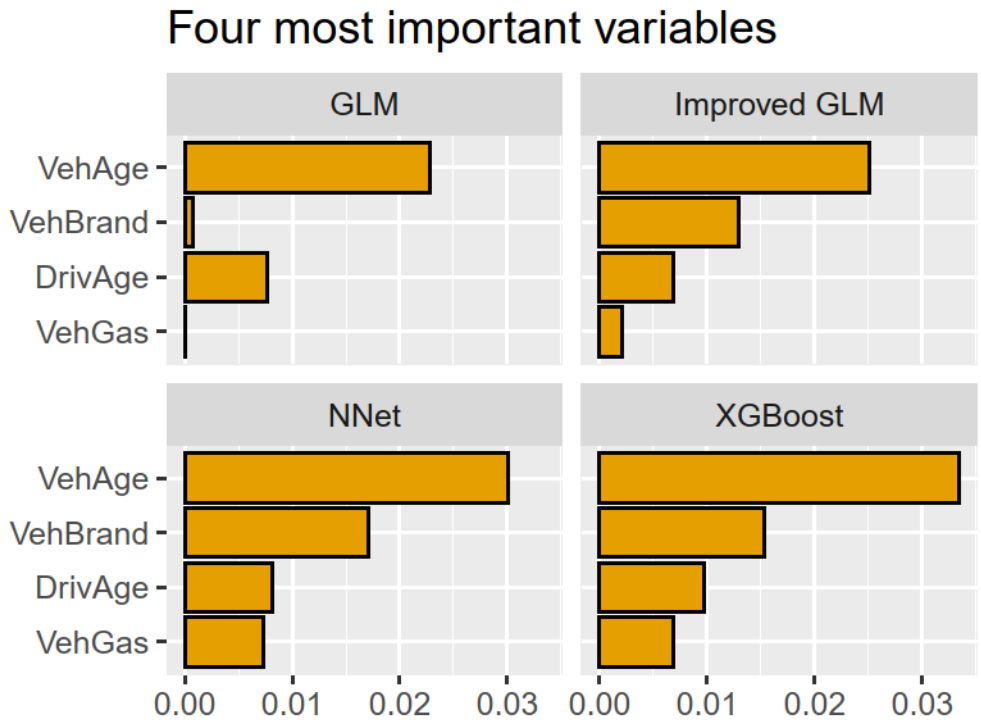
obs $\leftarrow$ data row
for v *in grid of values* **do**
  obs$[x] \leftarrow v$
  ice$[v] \leftarrow$ prediction for obs
end
output : ice

Improving the GLM by interpretable machine learning

1. Run a simple GLM and a tuned black box ML model as benchmark.
2. Compare performance: How much room is left for improvement of the GLM? If the GLM does almost as good as the modern ML model, stop. Otherwise, use XAI to improve the GLM by following the next steps.
3. Study variable importance: Are the same variables important for the GLM and the ML benchmark model? Can certain variables be dropped?
4. Go through (main) effects plots for the strongest predictors in the ML benchmark. What numerical predictor needs more flexibility, e.g. by adding squared terms or splines? Can certain categorical classes be reasonably collapsed? Etc.
5. Study interaction strength.
6. Use these findings to improve the GLM. If performance is acceptably close to the ML benchmark, stop. Otherwise, iterate.

Improved GLM





The end

The end