Non life pricing: empirical comparison of classical GLM with tree based Gradient Boosted Models
Innovative approach to pure premium estimation

Leonardo Petrini

Paris, 8th June 2017
Outline

1 Methodologies and Tools
- GAM: A better GLM
- Tree Based Gradient Boosted Models

2 Empirical Results
- Adopted Datasets
- Performance Results
Motivation: Why should we bother?

- A correct and accurate pricing
- A better understanding of the risk components
- Number of Claims (NB) and Claim Severity (CS)

Key Quantity

\[ BurningCost = NB \times CS \]
Beyond GLM: Generalised Additive Models

\[ g(E[y|x]) = \beta_0 + f_1(x_1) + \ldots + f_p(x_p) \]

- **Advantages**
  - Effective in treating non-linearity
  - Can adapt to a large variety of scenarios

- **Disadvantages**
  - Can easily lead to overfitting
  - Computationally intensive

- **The 'mgcv' package:**
  - Define a formula
  - Create a parallel cluster
  - Run the 'mgcv::bam(...)' function
library(mgcv)
library(parallel)

ctrl <- list(nthreads = ...)
cl <- makeCluster(...) 

gamNB <- bam(formula = ... , data = ... , family = ..., 
          cluster = ...)

stopCluster(cl)
• Decision Tree based models
• Proven to work in Insurance
• XGBoost: The Kaggle "to-go" model
• Actively used by companies as ...
eXtreme Gradient Boosting: The State of Art

- Ensemble of Decision Trees
- Boosting Algorithm
- Active community
- Computationally attractive
- 10x Faster than GBM
library(xgboost)

train <- xgb.DMatrix(data = ..., label = ...)

test <- xgb.DMatrix(data = ..., label = ...)

watchlist <- list(train = ..., test = ...)

model <- xgb.train(params = list(...) ,
    data = ..., nround = rounds_eta,
    objective = ... ,
    eval_metric = ... )
Adopted Datasets

- CAS Dataset: 'freMTPL'
- Private Dataset: 'Actuarial Pricing Game'
- Pre-Processing
- Cross-Validation
- Metrics Used:
  - Number of Claims: Poisson Log-Loss
  - Claim Severity: Root Mean Square Error
  - Burning Cost: Normalised Gini Index
## CAS Dataset: GAM vs XGBoost

<table>
<thead>
<tr>
<th>Model</th>
<th>LogLoss</th>
<th>RMSE</th>
<th>Gini</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>GAM</strong></td>
<td>0.16</td>
<td>3852</td>
<td>0.24</td>
</tr>
<tr>
<td><strong>XGB</strong></td>
<td>0.17</td>
<td>1980</td>
<td>0.30</td>
</tr>
<tr>
<td><strong>XGB Tweedie</strong></td>
<td>-</td>
<td>-</td>
<td>0.25</td>
</tr>
</tbody>
</table>

**Gain**: 25%

![CAS Data: Pure Premium Gini Plot](image-url)
### Private Dataset: GAM vs XGBoost

<table>
<thead>
<tr>
<th>Model</th>
<th>LogLoss</th>
<th>RMSE</th>
<th>Gini</th>
</tr>
</thead>
<tbody>
<tr>
<td>GAM</td>
<td>0.38</td>
<td>2530</td>
<td>0.35</td>
</tr>
<tr>
<td>XGB</td>
<td>0.39</td>
<td>990</td>
<td>0.44</td>
</tr>
<tr>
<td>XGB Tweedie</td>
<td>-</td>
<td>-</td>
<td>0.38</td>
</tr>
</tbody>
</table>

**Gain**

- 26%
Thank You!

leopetrini@mail.com  |  GitHub: XGBoost-in-Insurance-2017