How Actuaries and Data Scientists could learn from each other

Xavier Conort
Chief Data Scientist @DataRobot
September 2018

1991-1998:

ENSAE ParisTech

université PARIS DIDEROT PARIS 7

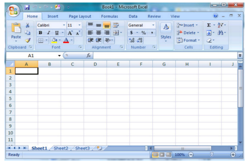
INSTITUT DES ACTUAIRES

sas

INSEE

1999-2007:

CNP ASSURANCES



Hyperion

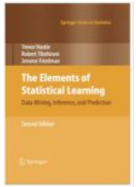
2008-2011:

CERA
Certified Enterprise Risk Actuary

AXA

BASIC RATEMAKING

2011-2013:


Trevor Hastie
Robert Tibshirani
Jerome Friedman
The Elements of Statistical Learning
Data Mining, Inference, and Prediction
Second Edition
Springer

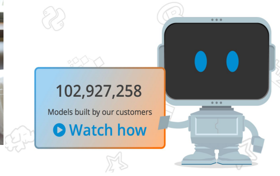
R

coursera
education for everyone

kaggle

A*STAR I²R

2013-now:



GitHub

scikit learn

dmlc XGBoost

TensorFlow

fastTEXT

DataRobot

AGENDA

- What are GLMs and why do Actuaries like to use them?
- Why do Data Scientists prefer Machine Learning? and what do they fail to learn from GLMs?
- Machine Learning initiatives, that successfully incorporate GLMs features
- How can Actuaries benefit from Machine Learning
- How to further improve Regularized Generalized Linear Models

DataRobot Confidential. Copyright © DataRobot, Inc. - All Rights Reserved

What are GLMs and why do Actuaries like to use them?

DataRobot

What Actuaries needs

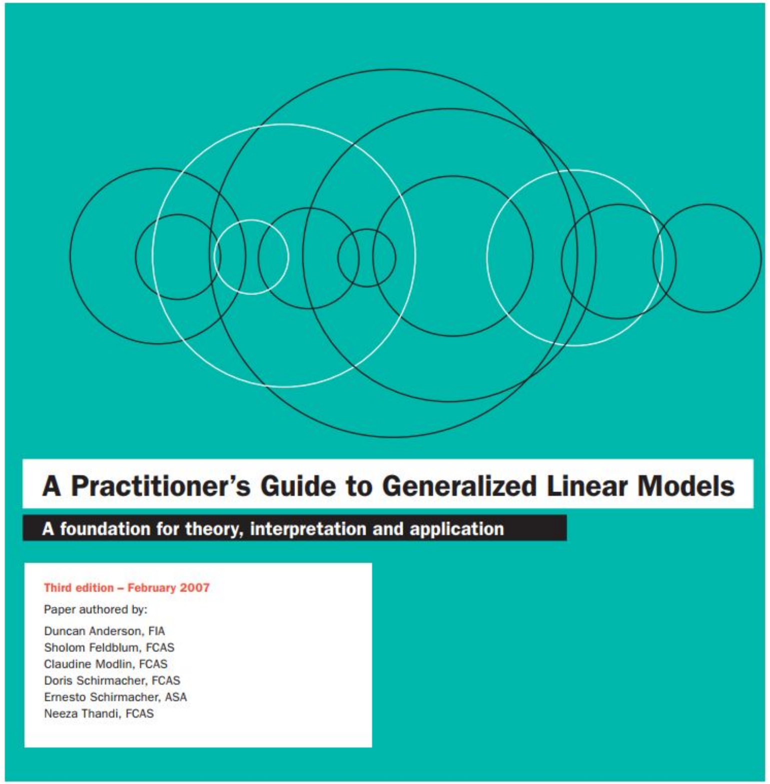
Actuaries have to deal with

- Risks with severity that follows **skewed distributions**
- Risks that tend to vary **multiplicatively** with rating factors
- **Pricing constraints**:
  - pricing that is pro-rata to the policy duration
  - commercial discounts
  - small changes vs previous pricing
- **Regulatory constraints**:
  - transparency
  - known relationships between risk and risk factors
    - example: risk expectation should always increase with the sum insured

We will see that GLMs structure is very relevant to Actuaries needs

DataRobot Confidential. Copyright © DataRobot, Inc. - All Rights Reserved

What are GLMs



A Practitioner's Guide to Generalized Linear Models

A foundation for theory, interpretation and application

Third edition – February 2007

Paper authored by:

Duncan Anderson, FIA
Sholom Feldblum, FCAS
Claudine Modlin, FCAS
Doris Schirmacher, FCAS
Ernesto Schirmacher, ASA
Neeza Thandi, FCAS

In summary, the assumed structure of a GLM can be specified as:

$$\mu_i = E[Y_i] = g^{-1}\left(\sum_j X_{ij}\beta_j + \xi_i\right)$$

$$Var[Y_i] = \frac{\phi V(\mu_i)}{\omega_i}$$

where

Y_i	is the vector of responses
$g(x)$	is the link function: a specified (invertible) function which relates the expected response to the linear combination of observed factors
X_{ij}	is a matrix (the "design matrix") produced from the factors
β_j	is a vector of model parameters, which is to be estimated
ξ_i	is a vector of known effects or "offsets"
ϕ	is a parameter to scale the function $V(x)$
$V(x)$	is the variance function
ω_i	is the prior weight that assigns a credibility or weight to each observation

GLMs variance function

$$Var(Y_i) = \frac{\phi . V(\mu_i)}{\omega_i},$$

	$V(x)$
Normal	1
Poisson	x
Gamma	x^2

In practice, Actuaries use GLMs with a poisson (frequency modeling), gamma (severity modeling) or tweedie (cost modeling) distribution to deal with risk severity that follows **skewed distributions.**

By using a **Gamma distribution**, Actuaries inform the model that the expected variance increases with the **square of the expected value** of each observation.

This is important because:

- This first reflects a reality: larger a risk is, larger its dispersion is.
- It second **prevents Actuaries from overfitting observations with large values**.
    - Indeed more tolerance is given to the deviation of the observed value with its expected value if this expected value is high.

DataRobot **Confidential**. Copyright © DataRobot, Inc. - All Rights Reserved

GLMs link function

$$\mu_i = E[Y_i] = g^{-1}(\sum_j X_{ij}\beta_j + \xi_i)$$

Log link function is also very much used in practice by actuaries. This helps:

- ensure that predicted value is non negative
- build a model that takes into account the fact that most insurance risk vary **multiplicatively** with rating factors

$$\mu_i = g^{-1}(\beta_1 x_{i1} + \ldots + \beta_p x_{ip}) = \exp(\beta_1 x_{i1}) \cdot \exp(\beta_2 x_{i2}) \ldots \exp(\beta_p x_{ip})$$

Important note: applying a linear model to the log of Y values (instead of a GLM with a log link function) is not recommended as this leads to biased predicted values. Indeed, log(E(Y)) is different to E(log(Y)). If you do this, don't forget to adjust your predictions or you will predict the median of the cost instead of the mean

DataRobot Confidential. Copyright © DataRobot, Inc. - All Rights Reserved

Offset

Offset is an awesome feature that allows Actuaries to **incorporate constraints or strengthen their modelling strategy**:

- they can ensure that predicted value is proportional to the exposure

$$E[Y_i] = g^{-1}\left(\sum_i X_{ij}\beta_j + \xi_i\right) = \exp\left(\sum_i X_{ij}\beta_j + \ln(e_i)\right) = \exp\left(\sum_j X_{ij}\beta_j\right).e_i$$

where e_i = the exposure for observation i.

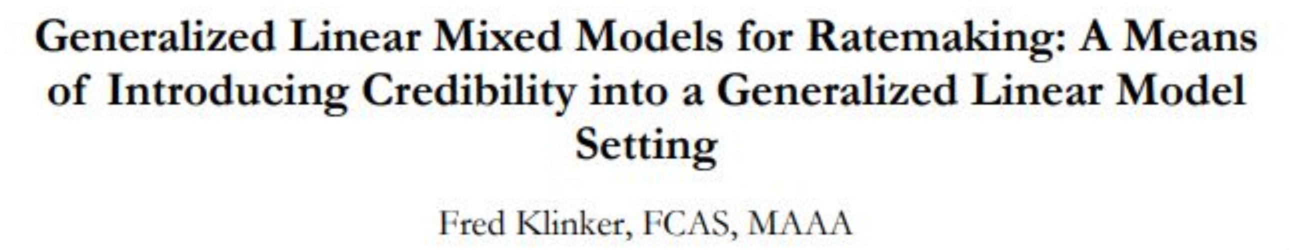
- they can apply arbitrary discount to some part of the population
- they can include a priori effects derived from:
  - other larger, similar products
  - market practice
  - previous pricing
- they can build a model in multiple stages:
  - first stage will focus on primary features that are fully trusted
  - second stage will capture the marginal effects of features that are less trusted and less commonly available

DataRobot Confidential. Copyright © DataRobot, Inc. - All Rights Reserved

Additional GLMs features

Some actuaries use:

- splines to learn non-linearity
  - those need to be manually defined and can computationally expensive
- binning to capture non-linearity
  - bins should not be too small such that they contain enough statistical material
- interactions to learn complex relationships
- mixed models to handle categorical features with high cardinality


Generalized Linear Mixed Models for Ratemaking: A Means of Introducing Credibility into a Generalized Linear Model Setting

Fred Klinker, FCAS, MAAA

Confidential. Copyright © DataRobot, Inc. - All Rights Reserved

Why do Data Scientists prefer Machine Learning? Are they discarding GLM features too fast?

DataRobot

Data Scientists work with too many features for GLMs

Building a **GLMs is very time consuming**. Actuaries need to manually check the statistical significance of each factor (via p-values) and then **remove the undesirable factors from the model**. This is not practical in presence of large number of features or unstructured data such as text.

Data Scientists prefer **automated regularization embedded in Machine Learning algorithms**.
One of the most popular ML algorithms is the Regularized GLMs (a very close cousin to GLM!) where a penalty is added to the GLMs loss function. This forces the model to automatically shrink to 0 coefficients of undesirable factors

Regularized loss function = GLM loss function + $\lambda_2 ||\beta||^2 + \lambda_1 ||\beta||_1$

If lambda2 is 0, it is a LASSO penalty. If lambda1 is 0, it is a RIDGE penalty. If none are zeros, it is an elastic-net penalty

DataRobot Confidential. Copyright © DataRobot, Inc. - All Rights Reserved

Data Scientists want to learn automatically complex signal

Data Scientists love **Machine Learning** (ML) algorithms that **can automatically capture non-linearity and interactions between factors**.
This helps them be very productive and get strong accuracy without much effort while building a complex GLMs requires Actuaries strong familiarity with the data, significant effort and expertise.
It is common to hear that ML helps speed up projects from X months to X weeks.
With Automated ML (automated preprocessing, automated hyper-parameters tuning, automated model selection), this can be further reduced to X days or even less.
Popular ML algorithms that are good in catching complexity:

- Gradient Boosting Machine
- Random Forest
- Neural Network
- Support Vector Machine

DataRobot **Confidential**. Copyright © DataRobot, Inc. - All Rights Reserved

GLMs features Data Scientists should know better

well known

little known

$$\mu_i = E[Y_i] = g^{-1}\left(\sum_j X_{ij}\beta_j + \xi_i\right)$$

$$Var[Y_i] = \frac{\phi V(\mu_i)}{\omega_i}$$

By discarding too fast GLMs from their toolbox, Data Scientists often fail to learn the benefits that some GLMs features can offer:

- **less risk to overfit large values** thanks to the use of poisson, gamma, tweedie loss functions
- **multiplicative structure** thanks to the log link function that can help the algorithm find more easily the signal
- ability to **incorporate known effects, control bias or do boosting** via the use of offset

Confidential. Copyright © DataRobot, Inc. - All Rights Reserved

Example of offset to control bias

The use of offset was critical in my solution to win the GE Flight Quest, a competition where I had to predict flights delays in USA.

I wanted the machine:

- to learn a flight is late because of bad traffic and bad weather
- and not to learn that one airport will never have delays in the future because there have not been any delay in this airport for the past 3 months

To achieve this, I used a 2 stages modeling (approach I learnt from actuaries!):

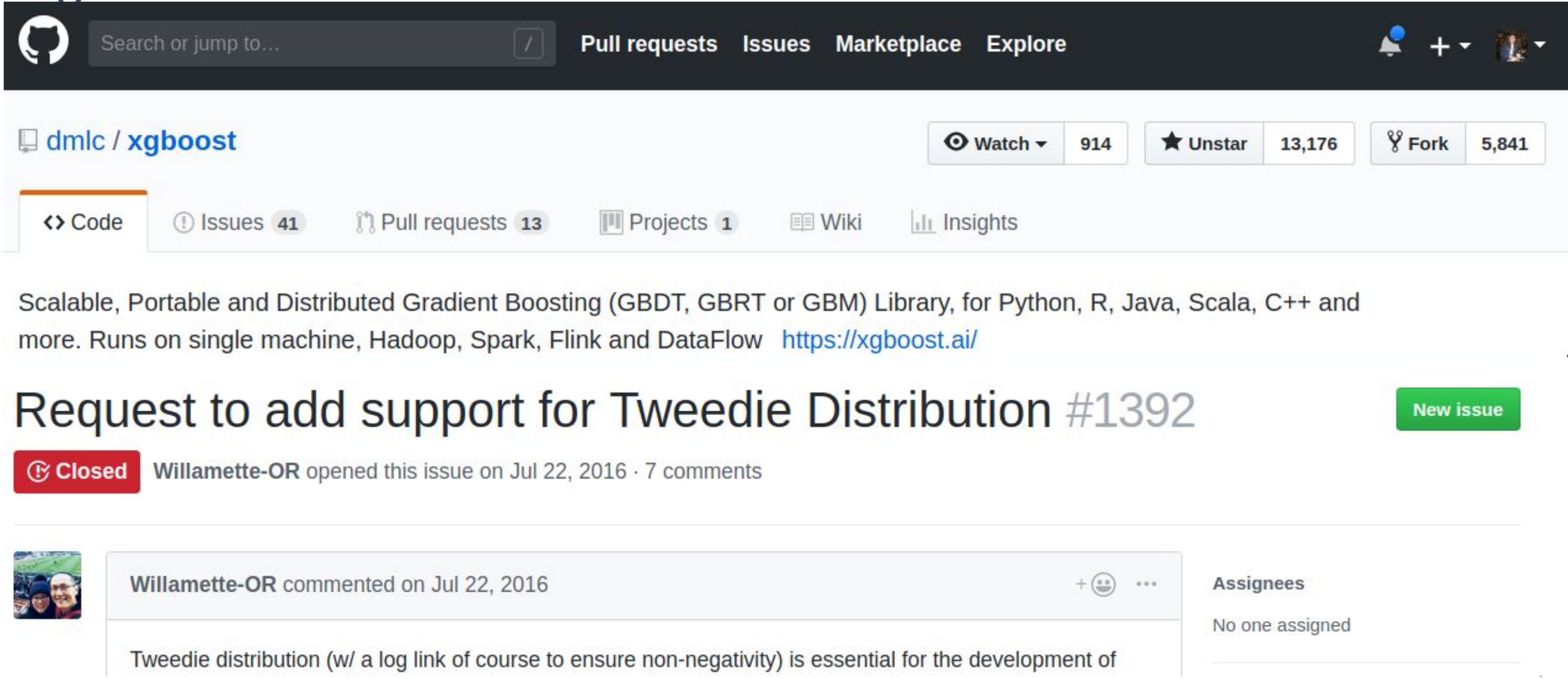
- first I fitted one GBM with features that I believed strongly related to delays
- then, I fitted one Regularized GLM with the GBM predictions as offset and other variables such as the name of the airport as features

DataRobot Confidential. Copyright © DataRobot, Inc. - All Rights Reserved

Machine Learning initiatives, that successfully incorporate GLMs features

DataRobot

Xgboost



Scalable, Portable and Distributed Gradient Boosting (GBDT, GBRT or GBM) Library, for Python, R, Java, Scala, C++ and more. Runs on single machine, Hadoop, Spark, Flink and DataFlow https://xgboost.ai/

Request to add support for Tweedie Distribution #1392

Willamette-OR opened this issue on Jul 22, 2016 · 7 comments

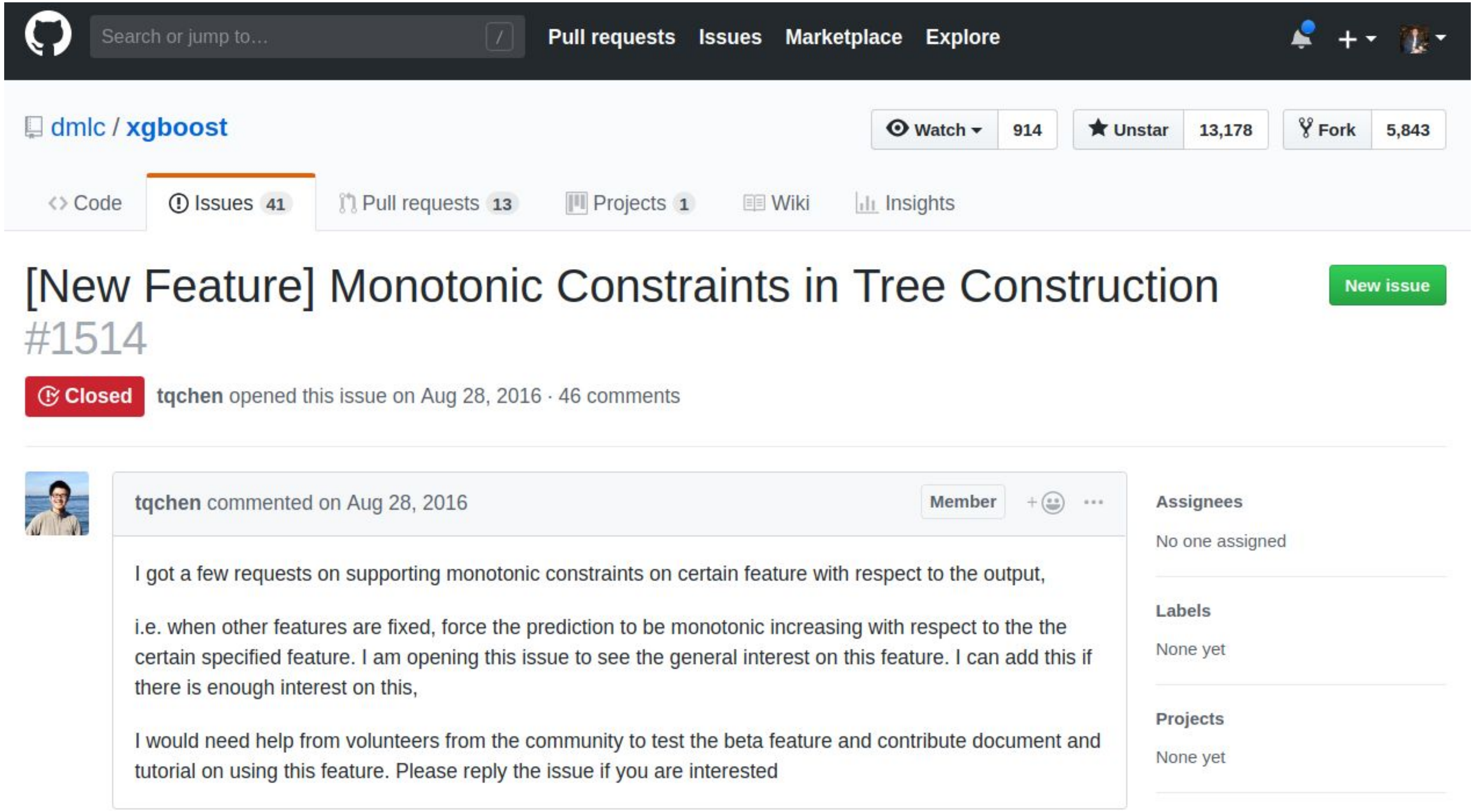
Willamette-OR commented on Jul 22, 2016

Tweedie distribution (w/ a log link of course to ensure non-negativity) is essential for the development of

Assignees

No one assigned

Xgboost



Pull requests Issues Marketplace Explore

dmlc / xgboost

Watch 914 Unstar 13,178 Fork 5,843

Code Issues 41 Pull requests 13 Projects 1 Wiki Insights

[New Feature] Monotonic Constraints in Tree Construction #1514

New issue

Closed **tqchen** opened this issue on Aug 28, 2016 · 46 comments

tqchen commented on Aug 28, 2016 Member

I got a few requests on supporting monotonic constraints on certain feature with respect to the output,

i.e. when other features are fixed, force the prediction to be monotonic increasing with respect to the the certain specified feature. I am opening this issue to see the general interest on this feature. I can add this if there is enough interest on this,

I would need help from volunteers from the community to test the beta feature and contribute document and tutorial on using this feature. Please reply the issue if you are interested

Assignees

No one assigned

Labels

None yet

Projects

None yet

DataRobot Confidential. Copyright © DataRobot, Inc. - All Rights Reserved

Why Actuaries and Data Scientists should be excited

$$\mu_i = E[Y_i] = g^{-1}(\sum_j X_{ij}\beta_j + \xi_i)$$

Gradient Boosting Machine (implemented by xgboost) is a close cousin to GLMs. The only difference is:

- the design matrix X is not defined by the user but is a set of thousands of rules found in a stagewise manner by the Machine.
- the coefficients beta are learnt slowly by the Machine.
- early stopping is used when accuracy improvements are too low

Actuaries can apply GBM to their risk modeling in a very similar way to what they do with GLMs
On the other hand, Data Scientists can add Actuaries tricks to their toolbox (exponential distributions, link function, offset...).

DataRobot Confidential. Copyright © DataRobot, Inc. - All Rights Reserved

Other Machine Learning initiatives

Offset support

Xgboost (GBM), H20 (GBM, ElasticNet, NN), DataRobot (GBM, ElasticNet, SVM), R gbm, R glmnet (ElasticNet)

Poisson support for claim frequencies

Xgboost (GBM), LightGBM (GBM), H20 (GBM, ElasticNet, NN), DataRobot (GBM, ElasticNet, NN, SVM), R glmnet (ElasticNet), pyglmnet (ElasticNet)

Gamma support for claims severities

Xgboost (GBM), LightGBM (GBM), H20 (GBM, ElasticNet, NN), DataRobot (GBM, ElasticNet, NN, SVM), pyglmnet (ElasticNet)

Tweedie support for total claim cost (frequency x severity)

Xgboost (GBM), LightGBM (GBM), H20 (GBM, ElasticNet, NN), DataRobot (GBM, ElasticNet, NN, SVM),

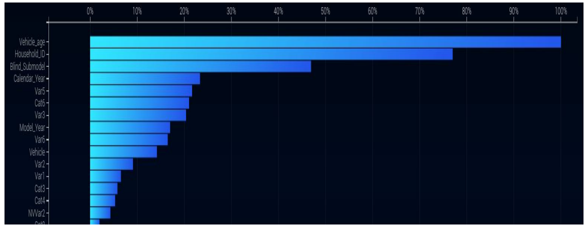
DataRobot Confidential. Copyright © DataRobot, Inc. - All Rights Reserved

How Actuaries can benefit from Machine Learning

Machine Learning (ML) benefits for Actuaries

Thanks to ML, Actuaries can

- Be more productive => More use cases
- Benchmark existing models
- Explore faster new data to enrich existing solutions.
- Improve existing models structure thanks to ML insights



explore new data value thanks to ML feature impact

How can you improve existing models thanks to ML?

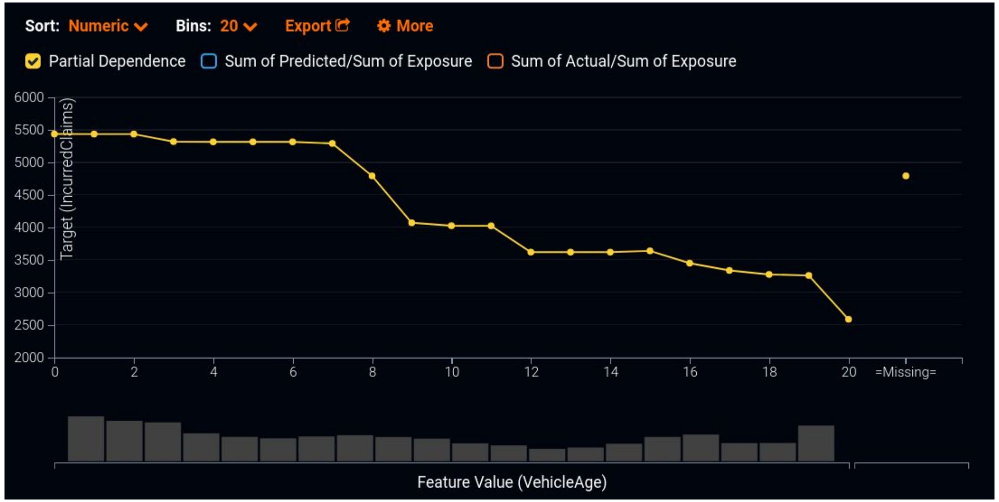
- learn optimal boundaries (from partial dependence plots) when you do binning to capture non-linearity
- add most impactful interactions found by ML
- improve modelling of categorical variables

Challenge: adding higher complexity to a GLM structure increases the risk of overfitting

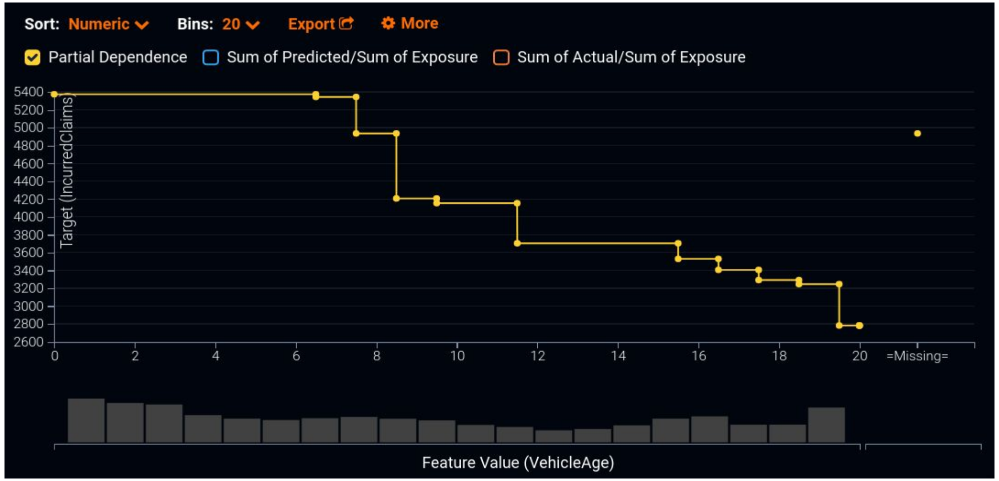
DataRobot **Confidential**. Copyright © DataRobot, Inc. - All Rights Reserved

Automated Binning

Learn from xgboost Partial Dependence



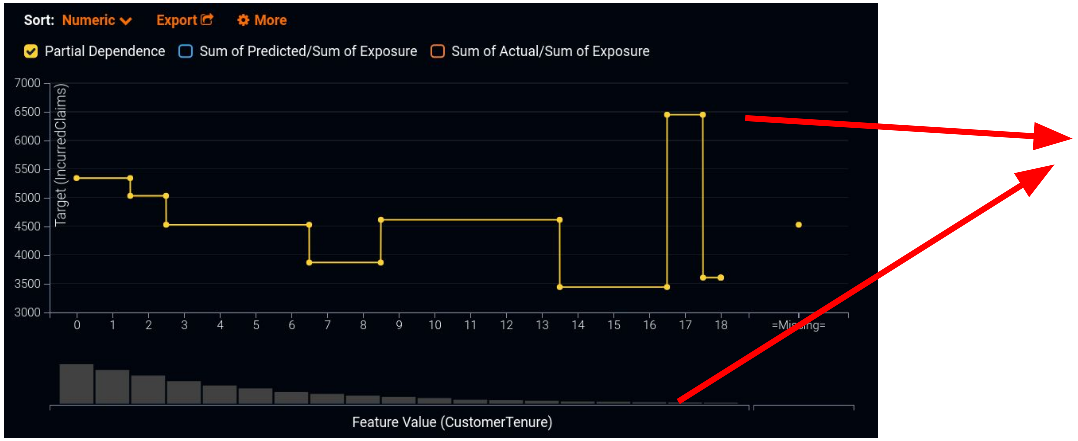
Find boundaries that best approximate the xgboost Partial Dependence



Risk of overfitting a complex structure

Creating (granular) bins from numeric features, adding levels of categorical features and adding interactions increase the risk of overfitting

Unfortunately, using traditional (Ridge or Lasso) GLMs won't help as information on the ordinal nature of bins is lost. For small bins with little statistical information, the Ridge or Lasso penalty will fail to assign a value close to the coefficients of adjacent bins.



Predictions for this bin seem very noisy. And there is too little data to make the bin's results credible.

How to reduce risk of overfitting of a complex structure

Here is the solution I developed for DataRobot's Generalized Additive Model to reduce the risk of overfitting small bins

Instead of fitting a GLM with

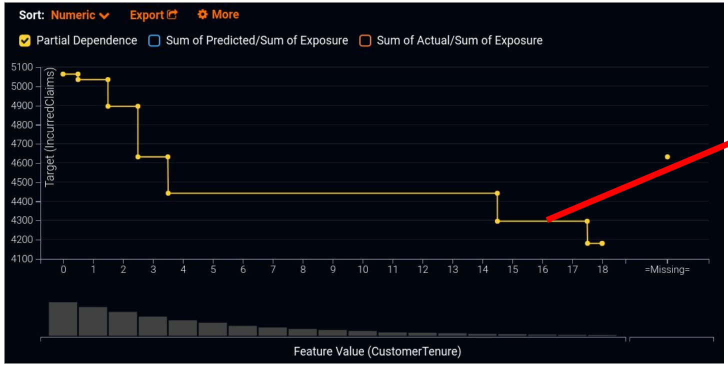
- Target ~ complex model structure, offset=O, link_function=L, distribution=d

I use a surrogate model as the strategy to combat overfitting.

I fit first a GBM with the same target, offset, link_function and distribution.

Then fit a Linear Model with

- GBM_margin_predictions ~ complex model structure



Results look much less noisy.
Indeed, GBM didn't capture the noise in the data thanks to regularisation and the ordinal nature of the feature that is exploited by GBM

Build your own surrogate model

- Fit your favorite ML algorithm:
  - Use one that supports link function, exponential distributions and offset if you need to
- Learn from the algorithm's insights for feature engineering:
  - feature impact to select factors
  - partial dependence plots for each factor to bin numeric features or decide on the formula
- Fit a "main effects" model
  - Apply the link function to the in-sample predictions of your ML algorithm (margin predictions) and use this as a target
  - Fit a linear model with the features you derived from the ML insights
  - Apply the inverse link function to the linear model predictions
- Find interactions:
  - use interactions reported by the ML algorithm if it exists (supported soon in DataRobot)
  - or look for interactions that best explain the residuals between your "main effects" solution and the predictions of your favorite ML algo
- Fit selected interactions on the residuals

DataRobot Confidential. Copyright © DataRobot, Inc. - All Rights Reserved

How to further improve Regularized Generalized Linear Models

This is an emerging hot topic in insurance

Last year during my visit in Japan, Iwasawa-san, a famous actuary in Japan, convinced me that the potential of Regularized Generalized Linear Models has not been fully exploited yet and **Fused Lasso** could be of great interest to Actuaries.
Thanks to him, I discovered that Fused Lasso can allow **data driven risk factor binning, levels grouping and spatial (or interactions) modeling within a GLM framework and combat the risk of overfitting small bins**!
In the meantime, my brother in France shared with me very good slides from Belgium actuarial researchers that say exactly the same thing
https://rininsurance17.sciencesconf.org/data/Devriendt.pdf

KU LEUVEN LRISK

Sparse modeling of risk factors in insurance analytics

Sander Devriendt
Joint work with Katrien Antonio, Edward Frees and Roel Verbelen

R in Insurance Conference
Paris, June 8, 2017

Sparse modeling of risk factors in insurance analytics 1/23

According to the researchers, magic comes from new penalties

- Ordinal risk factors (e.g. age): Fused Lasso

$$\lambda \sum_i w_i |\beta_{i+1} - \beta_i|.$$

- Nominal risk factors (e.g. car brand and model): Generalized Fused Lasso

$$\lambda \sum_{i>k} w_{i,k} |\beta_i - \beta_k|.$$

- Spatial risk factors (e.g. postal code): Graph Guided Fused Lasso

$$\lambda \sum_{(i,k)\in G} w_{i,k} |\beta_i - \beta_k|.$$

Unfortunately, no implementation is available yet except for the R genlasso package that supports only the gaussian distribution

Takeaways

DataRobot

Takeaways

Data Science took time to embrace Actuaries practices.
On the other end, Actuaries have resisted to Machine Learning innovations.

We can now observe interest on both sides and that more Machine Learning algorithms support features that are essential for Actuaries but also useful for Data Scientists.

More innovations are expected in the future.
New approaches such as surrogate models and Fused Lasso are good examples of emerging techniques that could change how both Actuaries and Data Scientists work.

DataRobot Confidential. Copyright © DataRobot, Inc. - All Rights Reserved

Questions