Do you trust your complex Machine Learning Model ?

Przemysław Biecek
Warsaw University of Technology

Complexity Institute
Singapore 2018

RENOIR - 691152 - H2020/2016-2019

Background

MSc in Software Engineering (2003)
and Mathematical Statistics (2003)
PhD in Mathematical Statistics (2007)
Habilitation in Medical Statistics (2013)

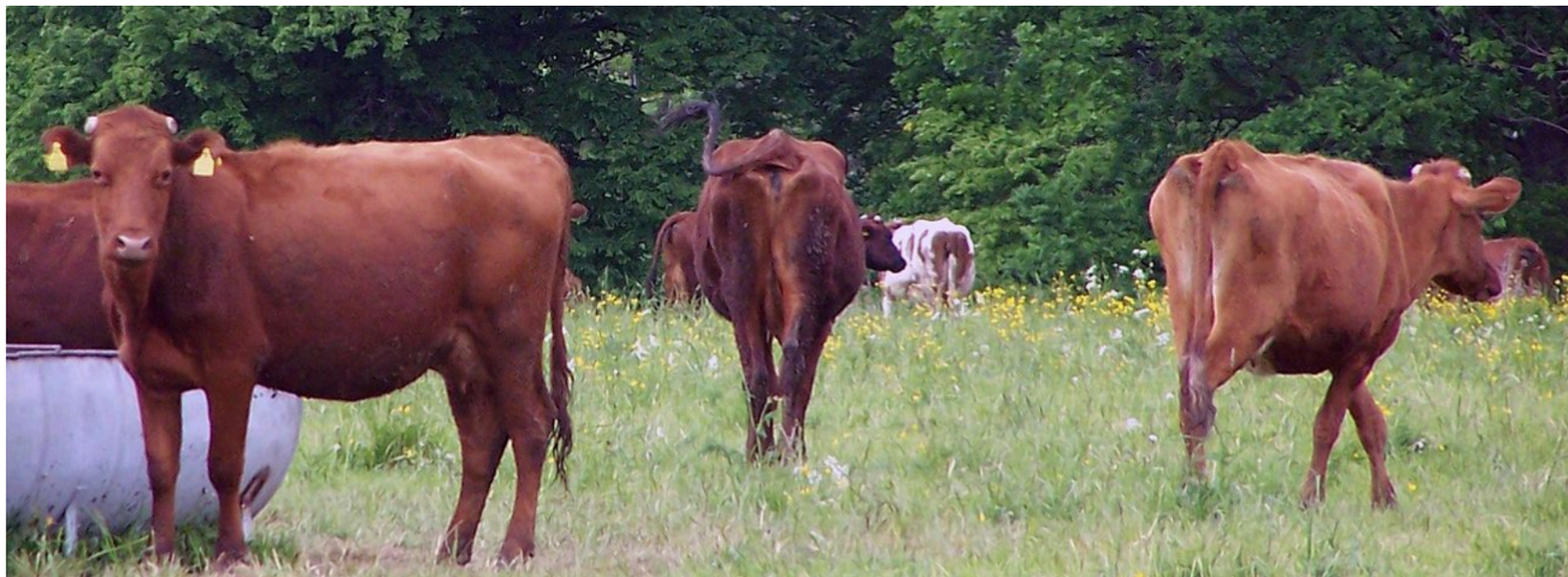
Research area

Machine Learning / Predictive Modelling
in molecular biology/medicine, mostly in oncology
Data visualisation
Model interpretability

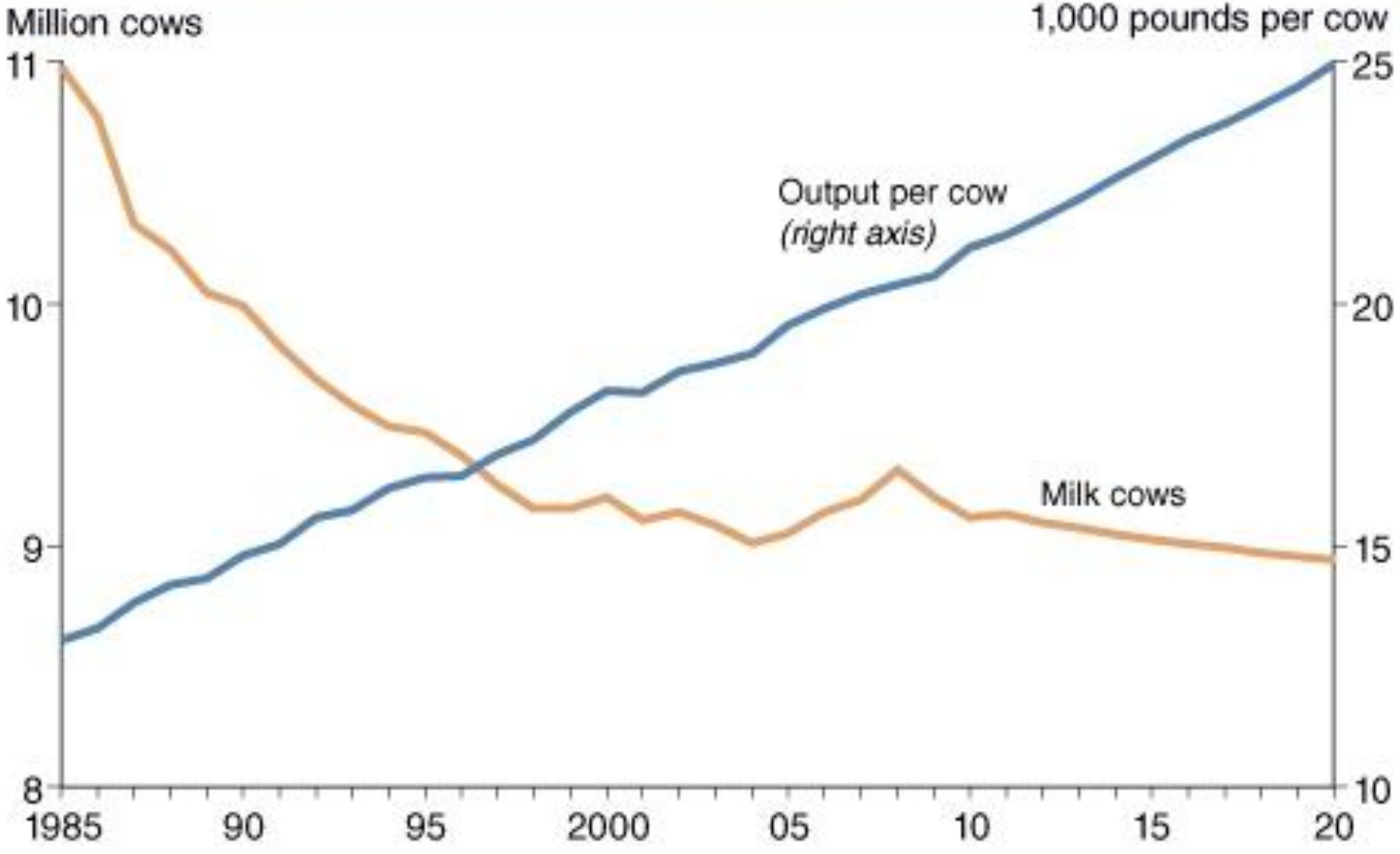
Head of MI^2 DataLab

Group of data enthusiasts (MSc and PhD students) from University of Warsaw and Warsaw University of Technology
https://github.com/MI2DataLab/

How large was the increase in the milk yield per cow during last 40 years?



U.S. dairy herd and milk production per cow



Source: USDA, Economic Research Service using USDA Agricultural Projections to 2020.

Possible due to selective breading! Machine learning is here.

BB2083
RACHID DE REMICHAMPAGNE

DOB: 05-APR-2013
HERD BOOK NO.: BBLBELM000857527143

CALVING DIFFICULTY		GESTATION (DAYS)	
PTA	REL.	PTA	REL.
6.3%	16%	-0.8d	42%

ICBF APR 2016

- ACCORD DE WIHOGNE
  - JUBILAIRE DE LA CLAIE
    - CARESSE DE LA CLAIE
- DEBORDANT ET DE BIOURGE
  - GOUTTE DE REMICHAMPAGNE
    - BOUTURE DE REMICHAMPAGNE

- First calves were born easily & good quality
- Easy calving on cows
- Widely used in Holland on dairy cows

BB4015
SIRE DE LA COUE

DOB: 01-NOV-2014
HERD BOOK NO.: BBLBELM000451780488

BELGIAN BLUE

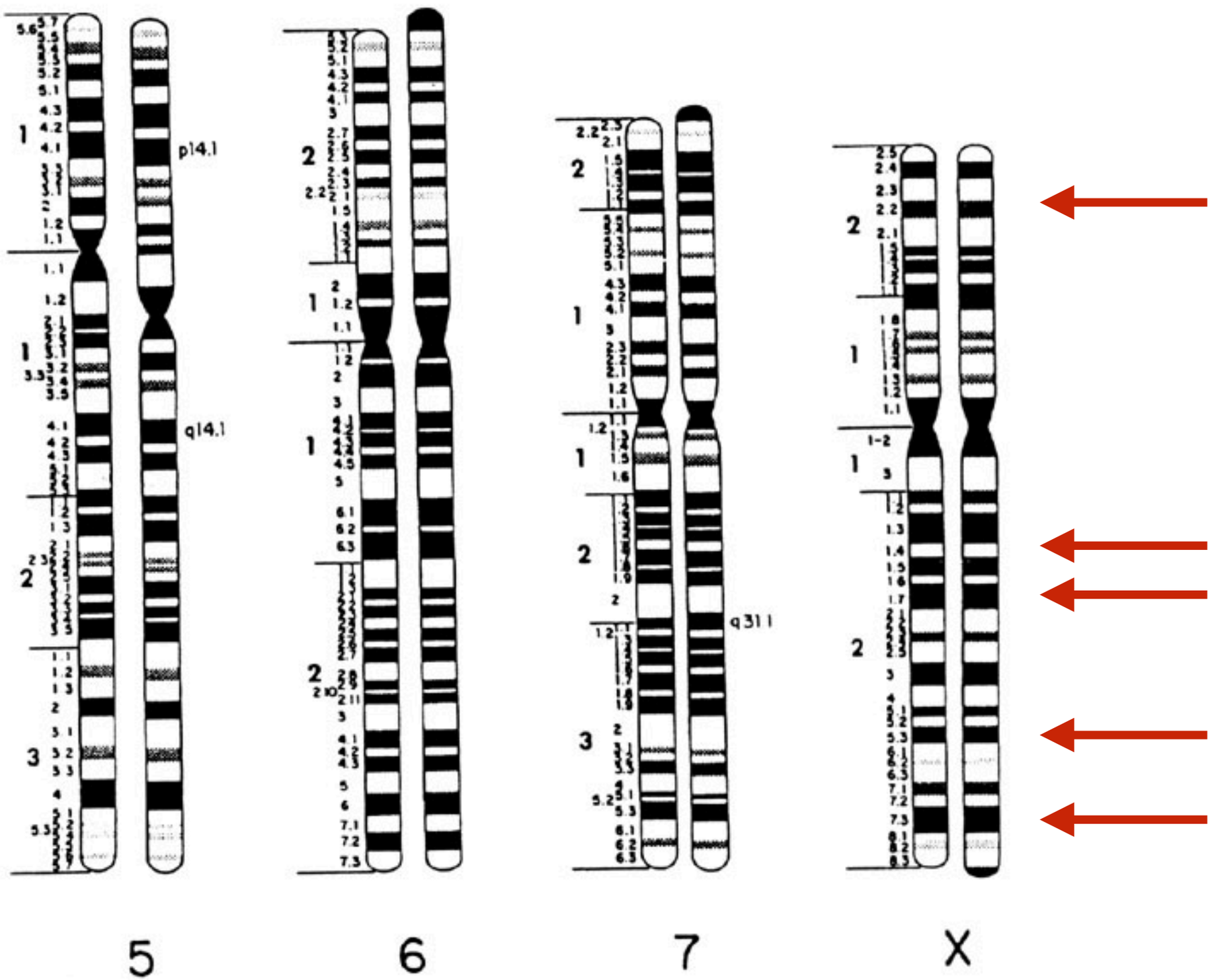
CALVING DIFFICULTY		GESTATION (DAYS)	
PTA	REL.	PTA	REL.
11.47%	14%	-0.88d	14%

ICBF APR 2016

- ORME DE SOMME
  - CAPPUCCINO DES AMANDIERS
    - LAETITIA ET DE CENTFONTAINE
- HERISSON DE LA COUE
  - LOQUACE DE LA COUE
    - DELICIEUSE DE LA COUE

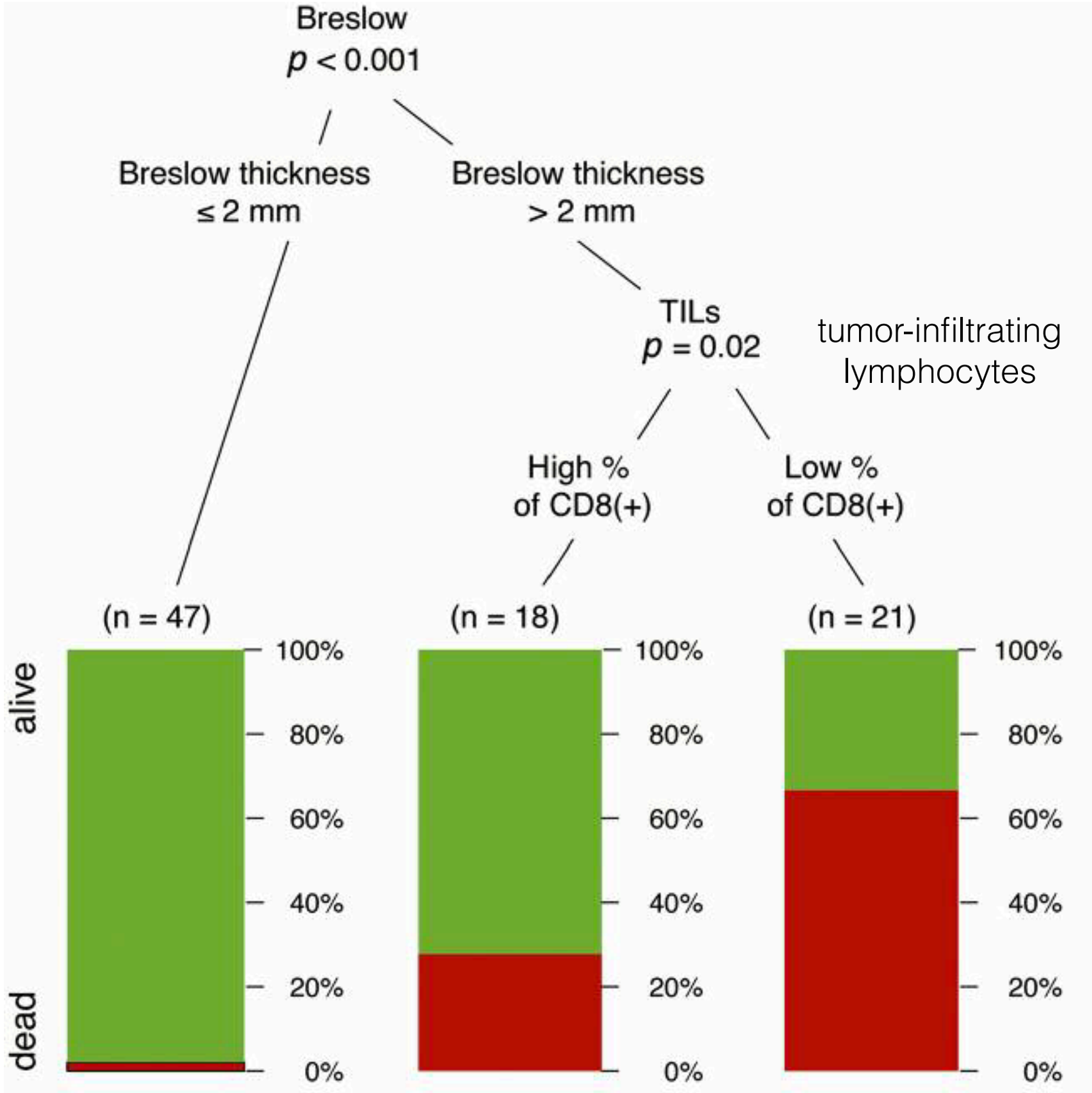
- Selected for crossing on dairy cows

These features are predicted based on genetic biomarkers like QTL = Quantitative Trait Loci

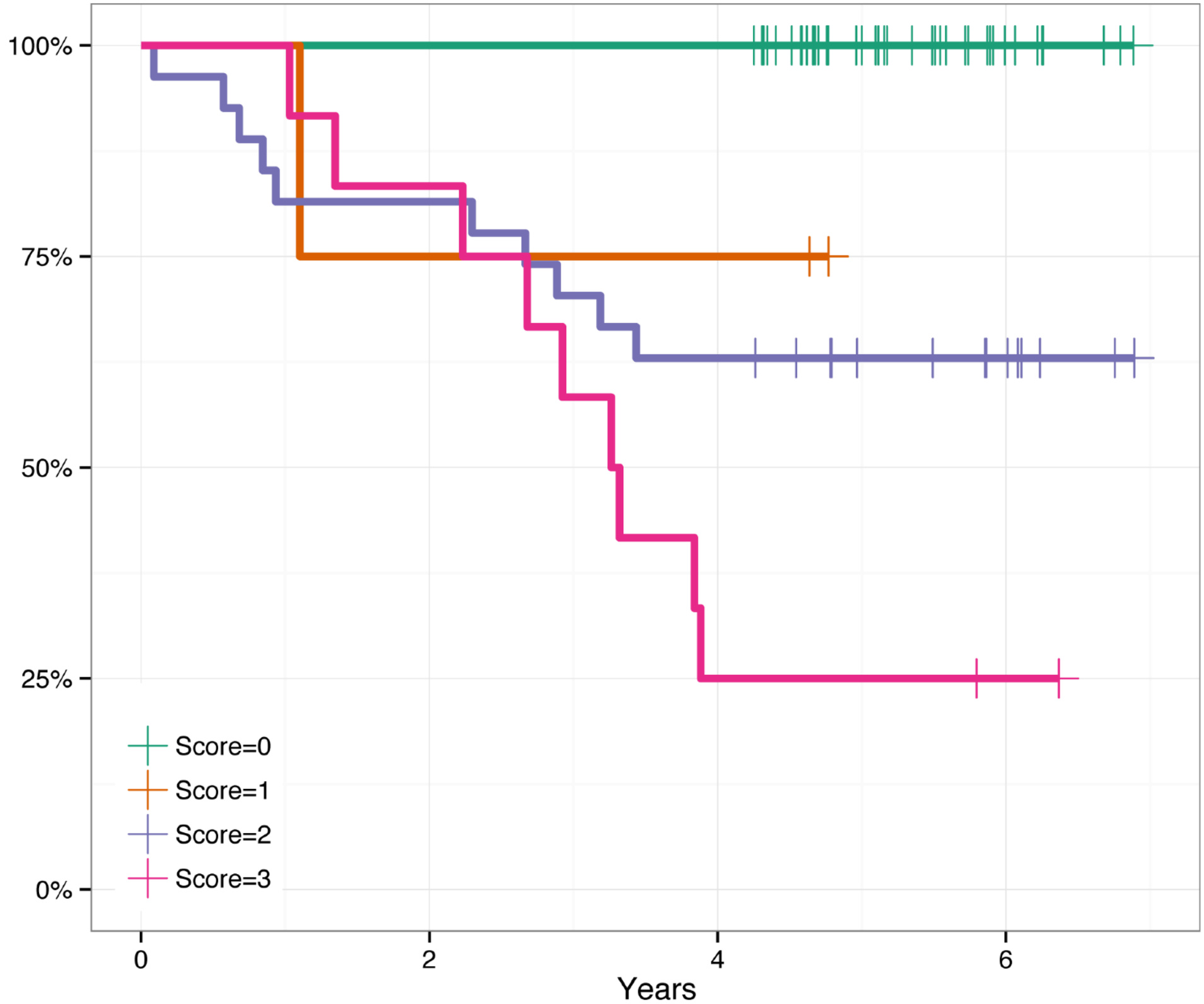


,,Small data” and
Machine Learning models

Collaboration with
Lower-Silesian Oncology Centre



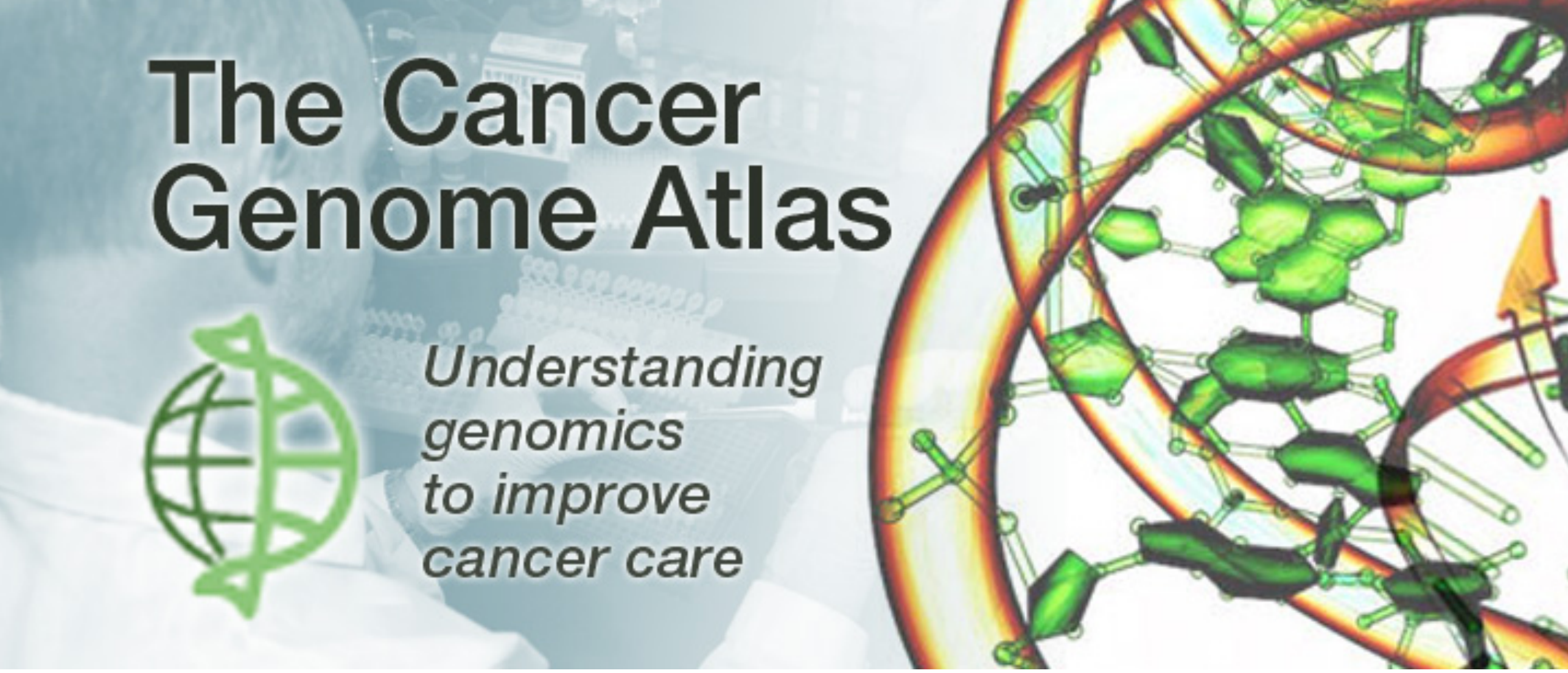
BILLCD8 - A Multivariable Survival Model as a Simple and Clinically Useful Prognostic Tool to Identify High-risk Cutaneous Melanoma Patients. Donizy P, Biecek P, Halon A, Matkowski R. ***Anticancer Res. 2016 Sep;36(9):4739-47.***



BILLCD8 - A Multivariable Survival Model as a Simple and Clinically Useful Prognostic Tool to Identify High-risk Cutaneous Melanoma Patients. Donizy P, Biecek P, Halon A, Matkowski R. ***Anticancer Res. 2016 Sep;36(9):4739-47.***

,,Big data'' and Machine Learning models

Collaboration with
The Greater Poland Cancer Center



The Cancer Genome Atlas (TCGA)

Multi-dimensional maps of the key genomic changes in **33 types of cancer**. **2.5 petabytes of data** describing tumour tissue and matched normal tissues from more than **11,000 patients** is **publicly available**. The data have contributed to more than a thousand studies of cancer by independent researchers.

Source: http://cancergenome.nih.gov/abouttcga/overview

Leukemia (LAML)

Lung adenocarcinoma (LUAD)

Lung squamous (LUSC)

Kidney (KIRC)

Bladder (BLCA)

Endometrial (UCEC)

Glioblastoma (GBM)

Head and neck (HNSC)

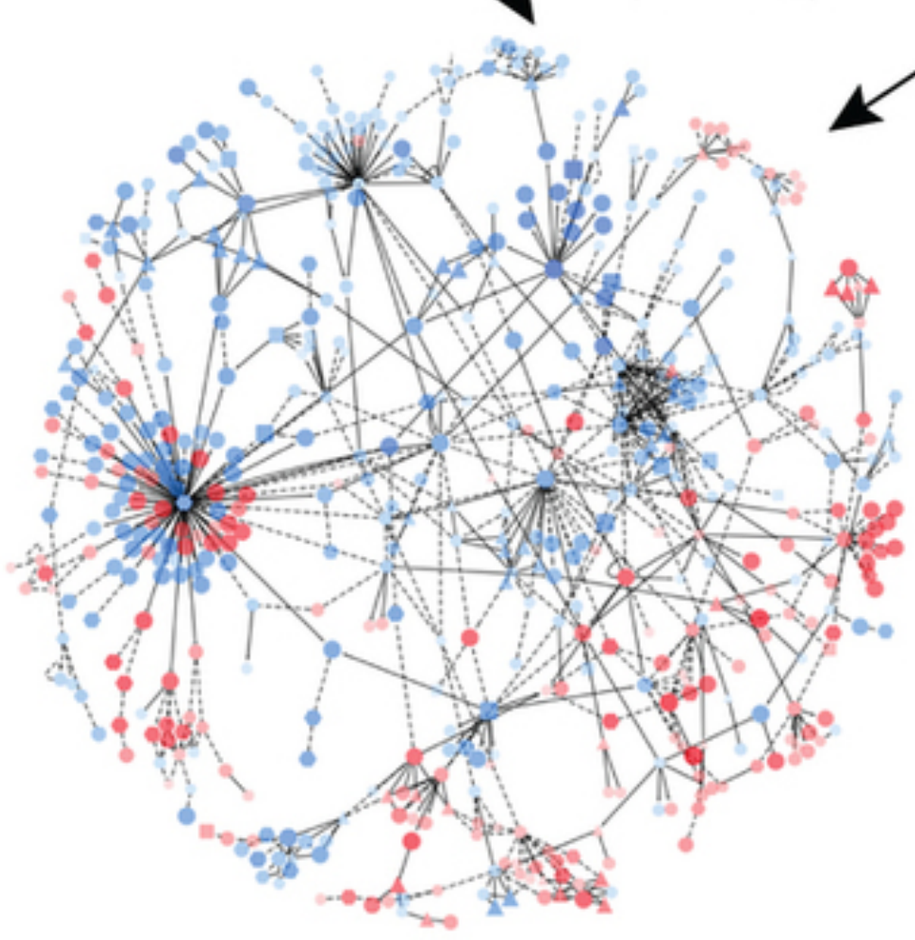
Breast (BRCA)

Ovarian (OV)

Colon (COAD)

Rectum (READ)

Thematic pathways



Omics characterizations

Platforms

Mutation

Copy number

Gene expression

DNA methylation

MicroRNA

RPPA

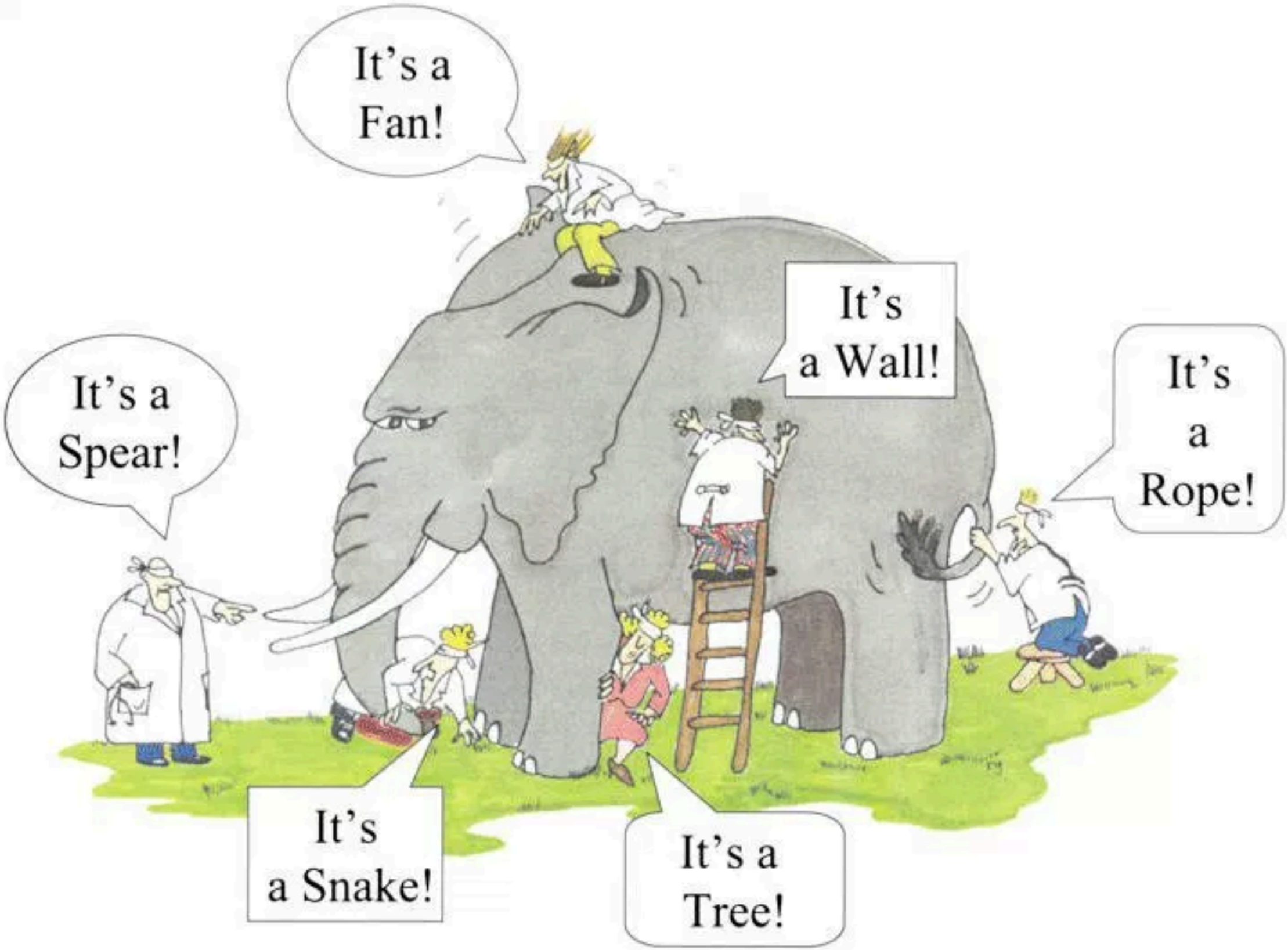
Clinical data

BRCA
BLCA
COAD
GBM
HNSC
KIRC
LAML
LUAD
LUSC
OV
READ
UCEC

Samples

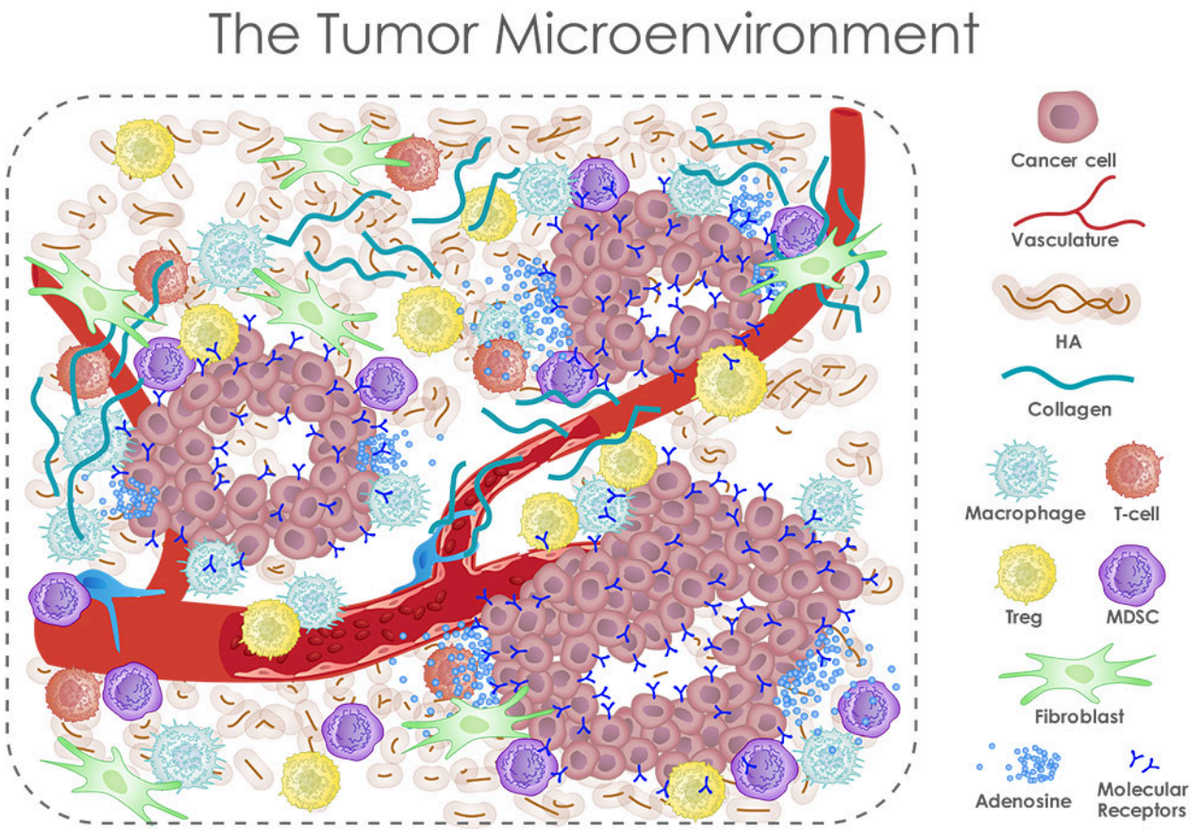
Genes/loci

Kyle Chang et al, The Cancer Genome Atlas Pan-Cancer analysis project, Article in Nature Genetics 45(10):1113-20

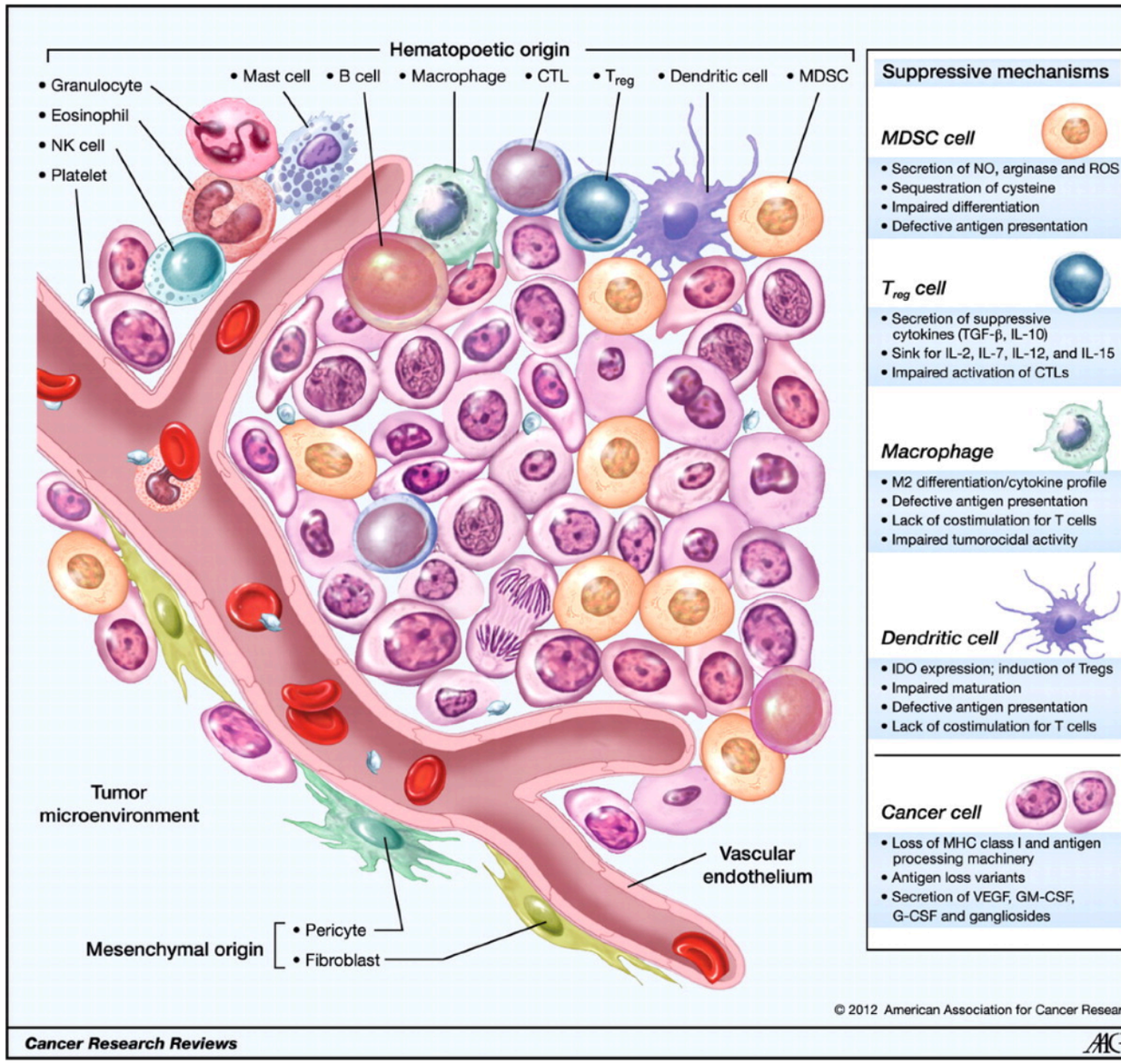


http://www.philipchircop.com/post/25783275888/seeing-the-full-elephant-its-a-tree-its-a

Can we use ML to recognise that cancer is more heterogeneous? And therefore more drug resistant?

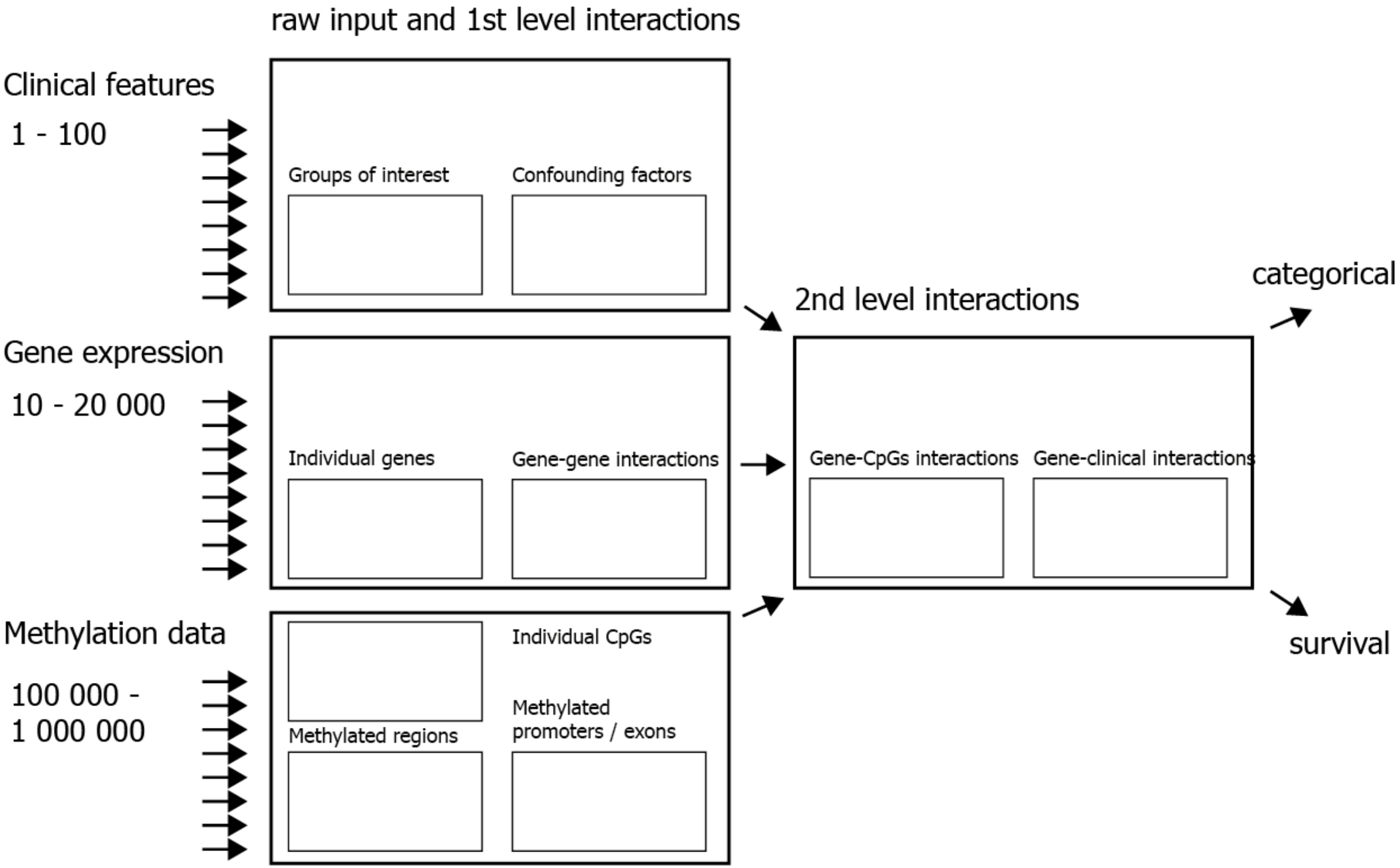


Whatcott et al.
Clin Cancer Res. 21:15 (2015)



Cellular Constituents of Immune Escape within the Tumor Microenvironment, Cancer Research

Learning with Structure: MLGenSig (Machine Learning Genetic Signatures)



Top 150 most important genes
(useful for domain specific validation)

DPPA5, SYCP1, CCR8, CCDC172, FCRL5, DRD3, UCMA, IL17C, CUZD1, HTR1A, CR2, SLCO4A1, PTPRM, SULF2, NPPB, ACSBG1, ITPK1, MYOF, PARP1, PPARGC1B, NUTM1, COLEC12, BEND4, DHX58, RNF19A, SIPA1L2, TAP2, TERT, BTG2, PARVA, PKIG, TIMP3, GPC6, ARHGAP24, DAB2, RIMS4, CPE, FLRT3, KRT8, SAMHD1, CAMK2D, GLB1L3, CMTM3, NLE1, TNRC18, DAZL, FAM24B-CUZD1, KCNG3, SMPD1, TNC, AHNAK, RAB39A, SLTM, ITGA5, FYTTD1, RRP1B, AGPAT5, EPB41L4B, TDRG1, DLC1, PLXNA2, CAMKV, GYLTL1B, WDR62, HES3, ROR2, CCDC6, COL5A2, ICMT, ALPK2, MYL4, ZDHHC22, LPAR3, HLA-DOA, HGSNAT, KRT19, SLC30A3, EFEMP2, LRIG3, ANKRD1, NCAPG2, YPEL5, ERVMER34-1, RRAGB, TTLL12, CKMT1B, CLIP1, PELP1, RABGAP1L, RASGRP2, UGT8, C20orf27, TOP1MT, SIT1, NDEL1, DKK1, ARHGAP29, POLR3G, SMAD6, PYDC1, PPAPDC1A, FLRT2, SCLY, TERF1, TNFAIP1, BX248253, DAZAP1, DIO3, PRIM1, MAGED2, STX5, ACOX3, PNPLA5, CTSB, TEX15, CDH2, RTN4RL2, SH2D5, ATP2A3, HHAT, SNCB, PFAS, LOXL4, ADAMTS9, PLCE1, SIRT1, BCAR3, EARS2, PALM3, WLS, ARL4D, NEK7, CAP2, TBC1D9, CCDC92, PPP1R15A, STEAP3, LCK, KIFAP3, RBM20, PLCXD3, CRIP3, DSCC1, EPSTI1, CPNE9, RARB, S1PR3, AKAP1, VARS

Complex models, ensembles, are often based on hundreds or thousands of simpler models.

Techniques like boosting or bagging turns multiple weak classifiers into a single stronger classifier.

Can we trust this model?

1. This model uses lots of inputs, variables, features, too many to find a single common pattern.
2. This model uses many small submodels, here 100 000 of decision trees that are doing the majority voting.
3. Assessment of the performance is based on a single number - accuracy (here =100%)

   Sensitivity and specificity is also 100%. Results are cross validated with different techniques but still there may be some bias in the data.

How we can validate this model against domain knowledge?

There are life-death decisions to be made, we need to trust this model before anyone will use it.

The Good, the Bad and the Ugly

some applications of Machine Learning Models

nature biomedical engineering

ARTICLES

https://doi.org/10.1038/s41551-018-0195-0

Prediction of cardiovascular risk factors from retinal fundus photographs via deep learning

Ryan Poplin[1,4], Avinash V. Varadarajan[1,4], Katy Blumer[1], Yun Liu[1], Michael V. McConnell[2,3], Greg S. Corrado[1], Lily Peng[1,4]* and Dale R. Webster[1,4]

Traditionally, medical discoveries are made by observing associations, making hypotheses from them and then designing and running experiments to test the hypotheses. However, with medical images, observing and quantifying associations can often be difficult because of the wide variety of features, patterns, colours, values and shapes that are present in real data. Here, we show that deep learning can extract new knowledge from retinal fundus images. Using deep-learning models trained on data from 284,335 patients and validated on two independent datasets of 12,026 and 999 patients, we predicted cardiovascular risk factors not previously thought to be present or quantifiable in retinal images, such as age (mean absolute error within 3.26 years), gender (area under the receiver operating characteristic curve (AUC) = 0.97), smoking status (AUC = 0.71), systolic blood pressure (mean absolute error within 11.23 mmHg) and major adverse cardiac events (AUC = 0.70). We also show that the trained deep-learning models used anatomical features, such as the optic disc or blood vessels, to generate each prediction.

Risk stratification is central to identifying and managing groups at risk for cardiovascular disease, which remains the leading cause of death globally[1]. Although the availability of cardiovascular disease risk calculators, such as the Pooled Cohort equations[2], Framingham[3–5] and Systematic COronary Risk Evaluation (SCORE)[6,7], is widespread, there are many efforts to improve risk predictions. Phenotypic information, particularly of vascu-

changes[22,23] ar
temic health c
clinical utility
work, we dem
cardiovascula

Machine le
ety of classifi

SCIENTIFIC REPORTS

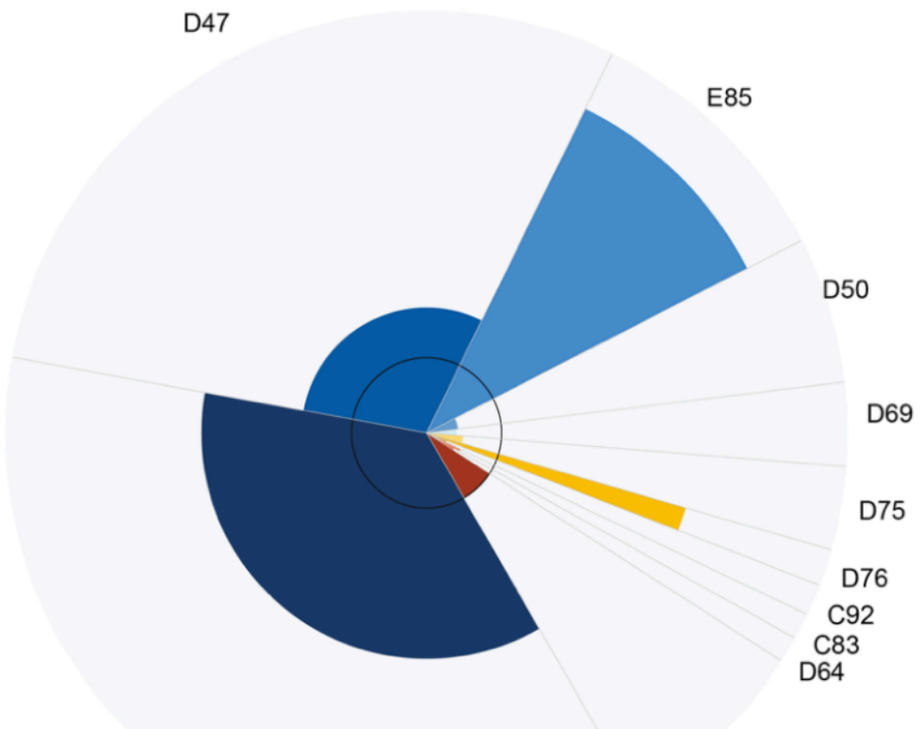
OPEN

An application of machine learning to haematological diagnosis

Gregor Gunčar[1], Matjaž Kukar[1], Mateja Notar[1], Miran Brvar[2], Peter Černelč[3], Manca Notar[1] & Marko Notar[1]

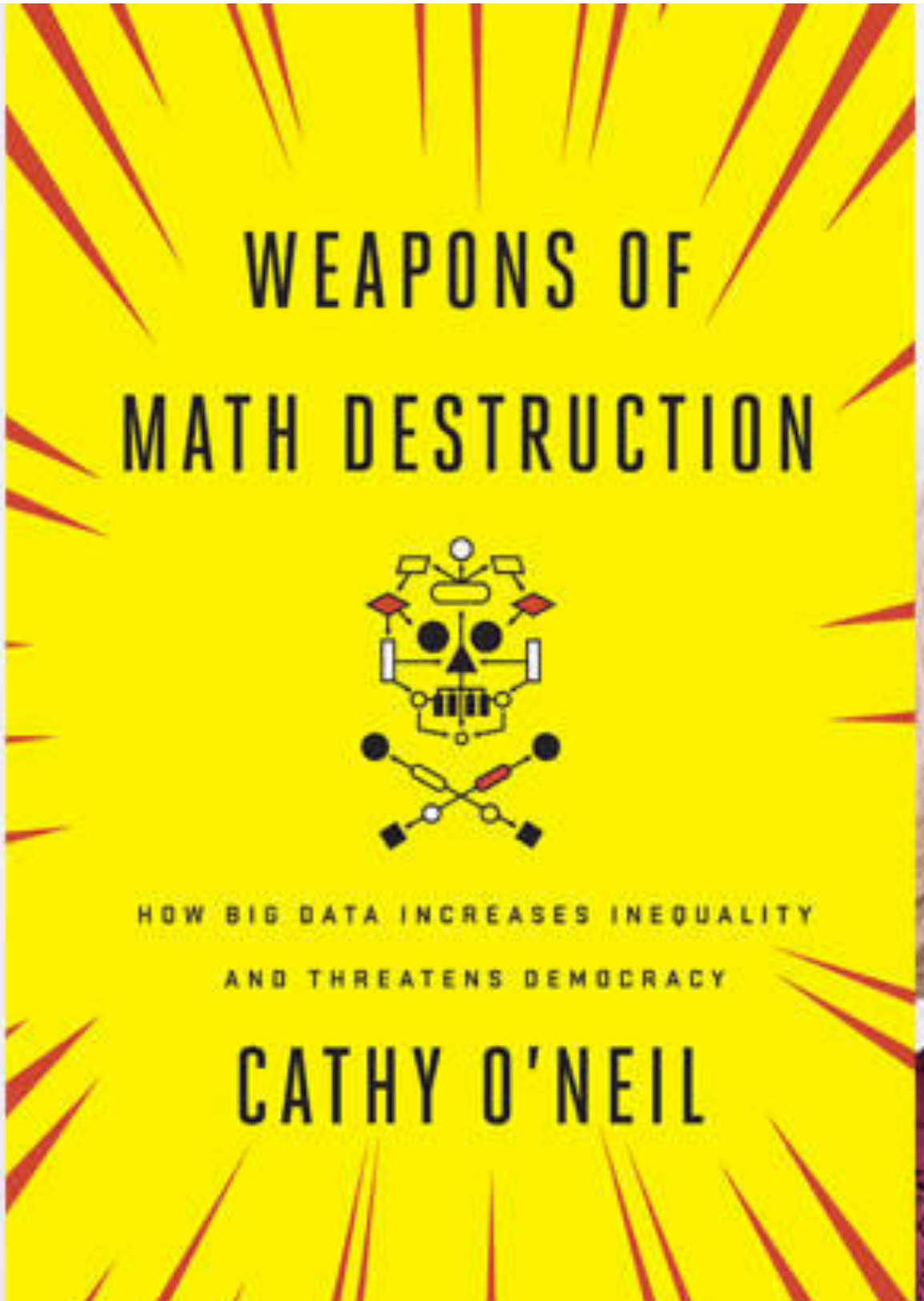
ved: 2 August 2017
ted: 14 December 2017
hed online: 11 January 2018

Quick and accurate medical diagnoses are crucial for the successful treatment of diseases. Using machine learning algorithms and based on laboratory blood test results, we have built two models to predict a haematologic disease. One predictive model used all the available blood test parameters and the other used only a reduced set that is usually measured upon patient admittance. Both models produced good results, obtaining prediction accuracies of 0.88 and 0.86 when considering the list of five most likely diseases and 0.59 and 0.57 when considering only the most likely disease. The models did not "finger and inc clinical special tests al unprec



ICD code	Prediction	Information score	Disease category
C90	36.20%	2.01	Multiple myeloma and malignant plasma cell neoplasms
D47	29.40%	0.67	Other neoplasms of uncertain or unknown behaviour of lymphoid. haematopoietic and related tissue
E85	10.20%	3.81	Amyloidosis
D50	5.60%	-1.37	Iron deficiency anaemia
D69	3.20%	-1.50	Purpura and other haemorrhagic conditions
D75	3.20%	-1.05	Other diseases of blood and blood-forming organs
D76	1.40%	2.60	Certain diseases involving lymphoreticular tissue and reticulohistiocytic system
C92	1.20%	-2.55	Myeloid leukaemia
C83	1.00%	-1.01	Diffuse non-Hodgkin lymphoma
D64	1.00%	-1.41	Other anaemias

Cathy O'Neil: The era of blind faith in big data must end

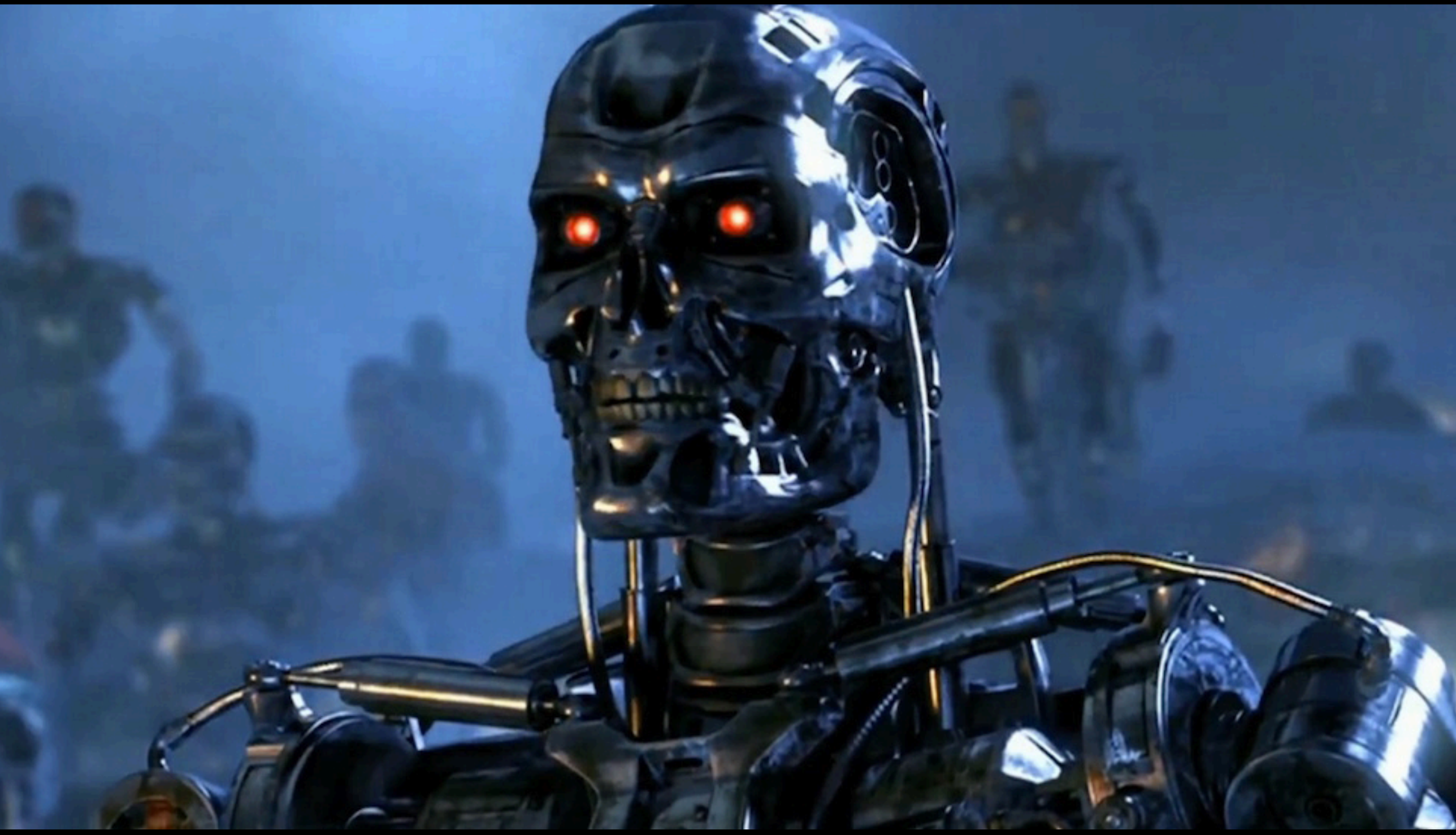


- “You don’t see a lot of skepticism,” she says. “The algorithms are like shiny new toys that we can’t resist using. We trust them so much that we project meaning on to them.”

- Ultimately algorithms, according to O’Neil, reinforce discrimination and widen inequality, “using people’s fear and trust of mathematics to prevent them from asking questions”.

https://www.theguardian.com/books/2016/oct/27/cathy-oneil-weapons-of-math-destruction-algorithms-big-data

Machine Learning Models will replace some humans



Machine Learning Models will empower humans

What do we need?

"Why Should I Trust You?" Explaining the Predictions of Any Classifier

Marco Tulio Ribeiro
University of Washington
Seattle, WA 98105, USA
marcotcr@cs.uw.edu

Sameer Singh
University of Washington
Seattle, WA 98105, USA
sameer@cs.uw.edu

Carlos Guestrin
University of Washington
Seattle, WA 98105, USA
guestrin@cs.uw.edu

ABSTRACT

Despite widespread adoption, machine learning models remain mostly black boxes. Understanding the reasons behind predictions is, however, quite important in assessing *trust*, which is fundamental if one plans to take action based on a prediction, or when choosing whether to deploy a new model. Such understanding also provides insights into the model, which can be used to transform an untrustworthy model or prediction into a trustworthy one.

In this work, we propose LIME, a novel explanation technique that explains the predictions of *any* classifier in an interpretable and faithful manner, by learning an interpretable

how much the human understands a model's behaviour, as opposed to seeing it as a black box.

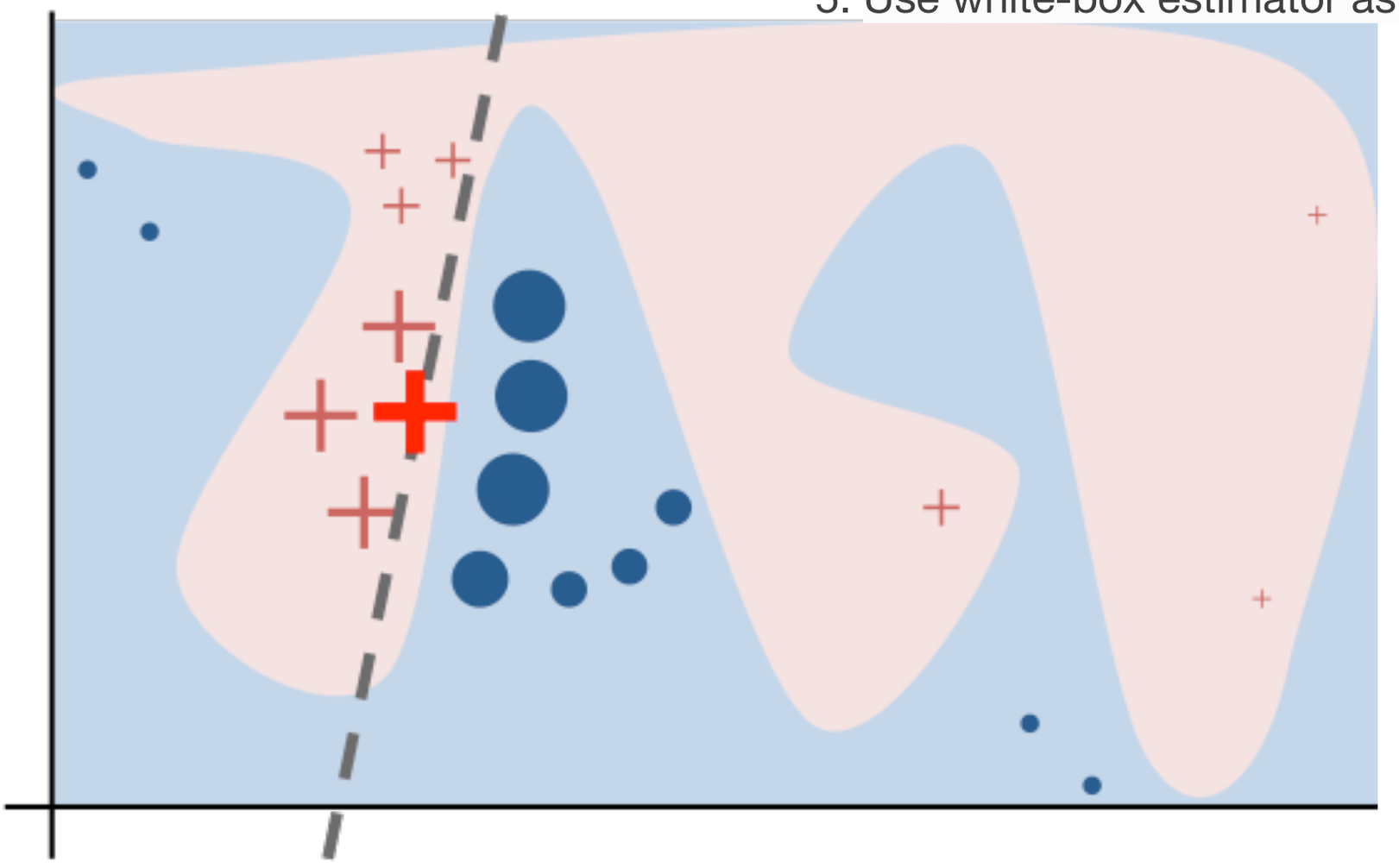
Determining trust in individual predictions is an important problem when the model is used for decision making. When using machine learning for medical diagnosis [6] or terrorism detection, for example, predictions cannot be acted upon on blind faith, as the consequences may be catastrophic.

Apart from trusting individual predictions, there is also a need to evaluate the model as a whole before deploying it "in the wild". To make this decision, users need to be confident that the model will perform well on real-world data, according to the metrics of interest. Currently, models are evaluated

More than 250 citations in last 18 months.

LIME: Local Interpretable Model-agnostic Explanations

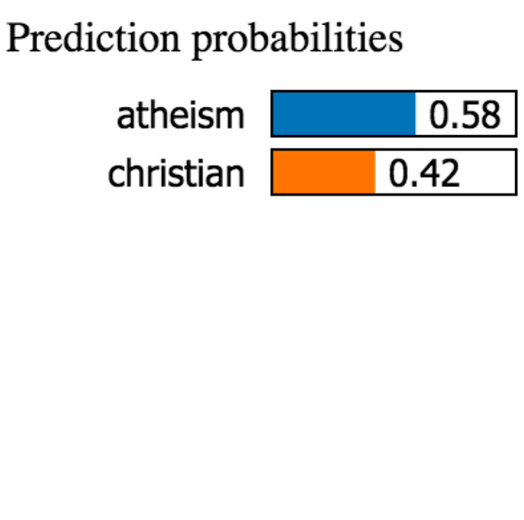
1. Generate a fake dataset around x.
2. Use black-box estimator to get target values y.
3. Train a new white-box estimator for (y,x).
4. Check prediction quality of a white-box classifier.
5. Use white-box estimator as an explanation of black-box model.

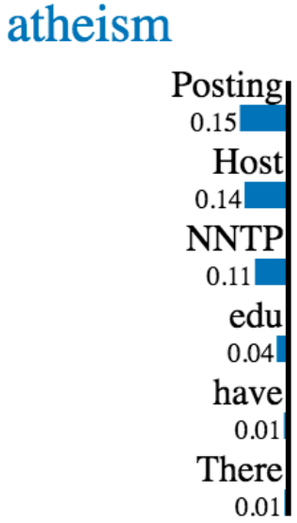


"Why Should I Trust You?" Explaining the Predictions of Any Classifier.
Marco Tulio Ribeiro, Sameer Singh, Carlos Guestrin (2016). https://arxiv.org/pdf/1602.04938.pdf

Port to R: Thomas Lin Pedersen (2017) https://github.com/thomasp85/lime

Explainers for text





Text with highlighted words

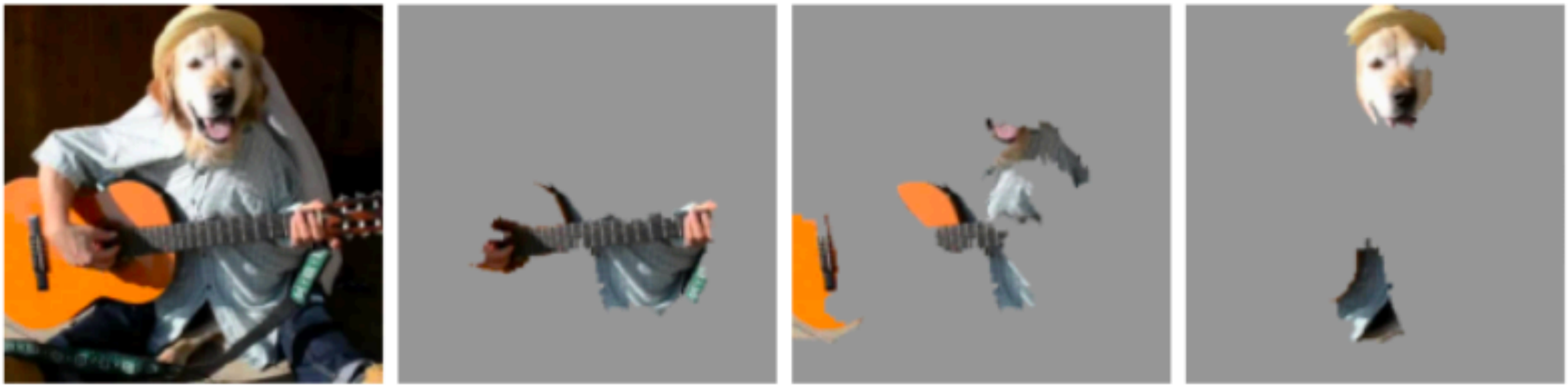
From: johnchad@triton.unm.edu (jchadwic)
Subject: Another request for Darwin Fish
Organization: University of New Mexico, Albuquerque
Lines: 11
NNTP-Posting-Host: triton.unm.edu

Hello Gang,

There have been some notes recently asking where to obtain the DARWIN fish.
This is the same question I have and I have not seen an answer on the
net. If anyone has a contact please post on the net or email me.

"Why Should I Trust You?" Explaining the Predictions of Any Classifier.
Marco Tulio Ribeiro, Sameer Singh, Carlos Guestrin (2016). https://arxiv.org/pdf/1602.04938.pdf

Explainers for images

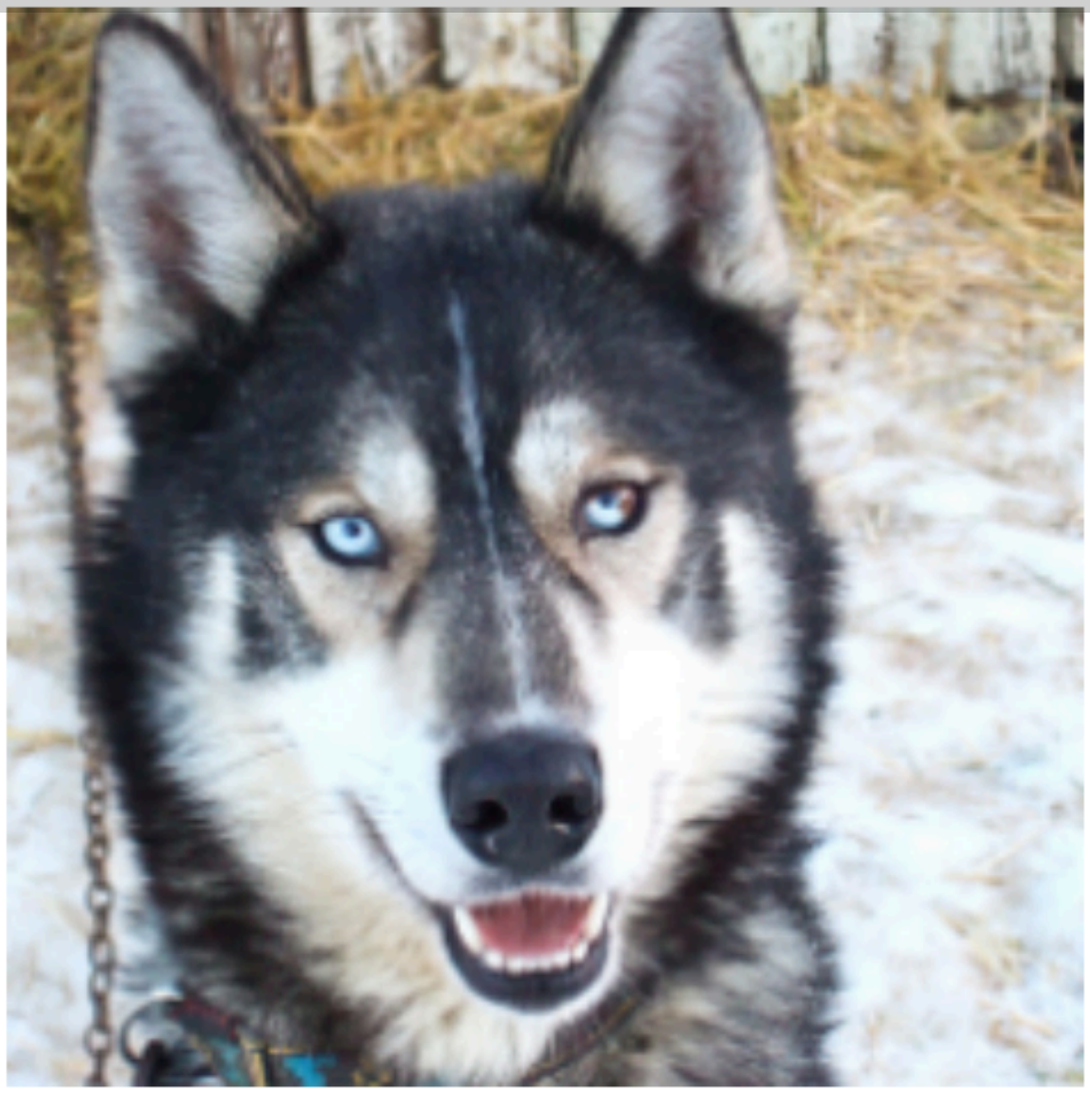


(a) Original Image (b) Explaining *Electric guitar* (c) Explaining *Acoustic guitar* (d) Explaining *Labrador*

Figure 4: Explaining an image classification prediction made by Google's Inception network, highlighting positive pixels. The top 3 classes predicted are "Electric Guitar" ($p = 0.32$), "Acoustic guitar" ($p = 0.24$) and "Labrador" ($p = 0.21$)

"Why Should I Trust You?" Explaining the Predictions of Any Classifier.
Marco Tulio Ribeiro, Sameer Singh, Carlos Guestrin (2016). https://arxiv.org/pdf/1602.04938.pdf

What went wrong?



(a) Husky classified as wolf

(b) Explanation

"Why Should I Trust You?" Explaining the Predictions of Any Classifier.
Marco Tulio Ribeiro, Sameer Singh, Carlos Guestrin (2016). https://arxiv.org/pdf/1602.04938.pdf

DALEX:

Descriptive mAchine LEarning eXplanations

Springer Series in Statistics

Trevor Hastie
Robert Tibshirani
Jerome Friedman

The Elements of Statistical Learning

Data Mining, Inference, and Prediction

Springer

MI DATA LAB

EXPLAIN!

How to attribute parts of model prediction to features?

Goal of this explainer:

- identify key features (reasons) that influence model prediction for a single observation,
- compare reasons across different models.

Use case:

- Prediction of treatment for a patient and reasons behind this prediction.

How to attribute parts of model prediction to features?

- Shapley values (*Additive feature attribution methods*) Lundberg, Lee 2017

- live (*Local Interpretable Visual Explanations*), Staniak, Biecek 2018

Definition **(Distance to relaxed model prediction)** *For a set of indexes indSet let us define the distance between model prediction and relaxed model prediction.*

$$d(x^{new}, indSet) := |f^{IndSet}(x^{new}) - f^{(}x^{new})|.$$

Definition **(Relaxed model prediction)** *Let $f^{IndSet}(x^{new})$ denote an expected model prediction for x^{new} relaxed on the set of indexes $IndSet \subset \{1, \ldots, p\}$.*

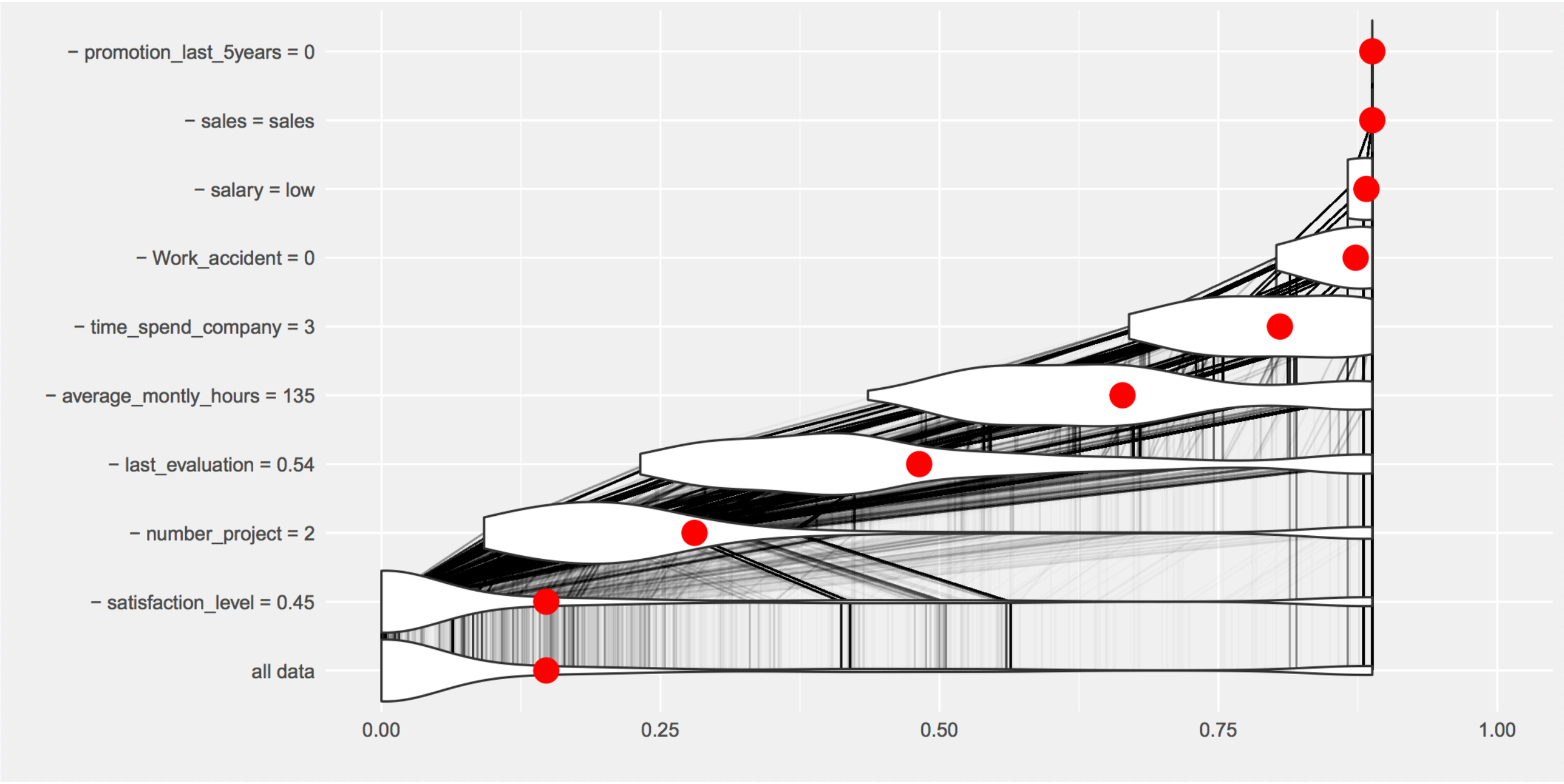
$$f^{IndSet}(x^{new}) = E[f(x)|x_{indSet} = x^{new}_{indSet}].$$

It is an expected value for model response conditioned on variables from set indSet in a way, that $\forall_{i \in indSet} x_i = x_i^{new}$.

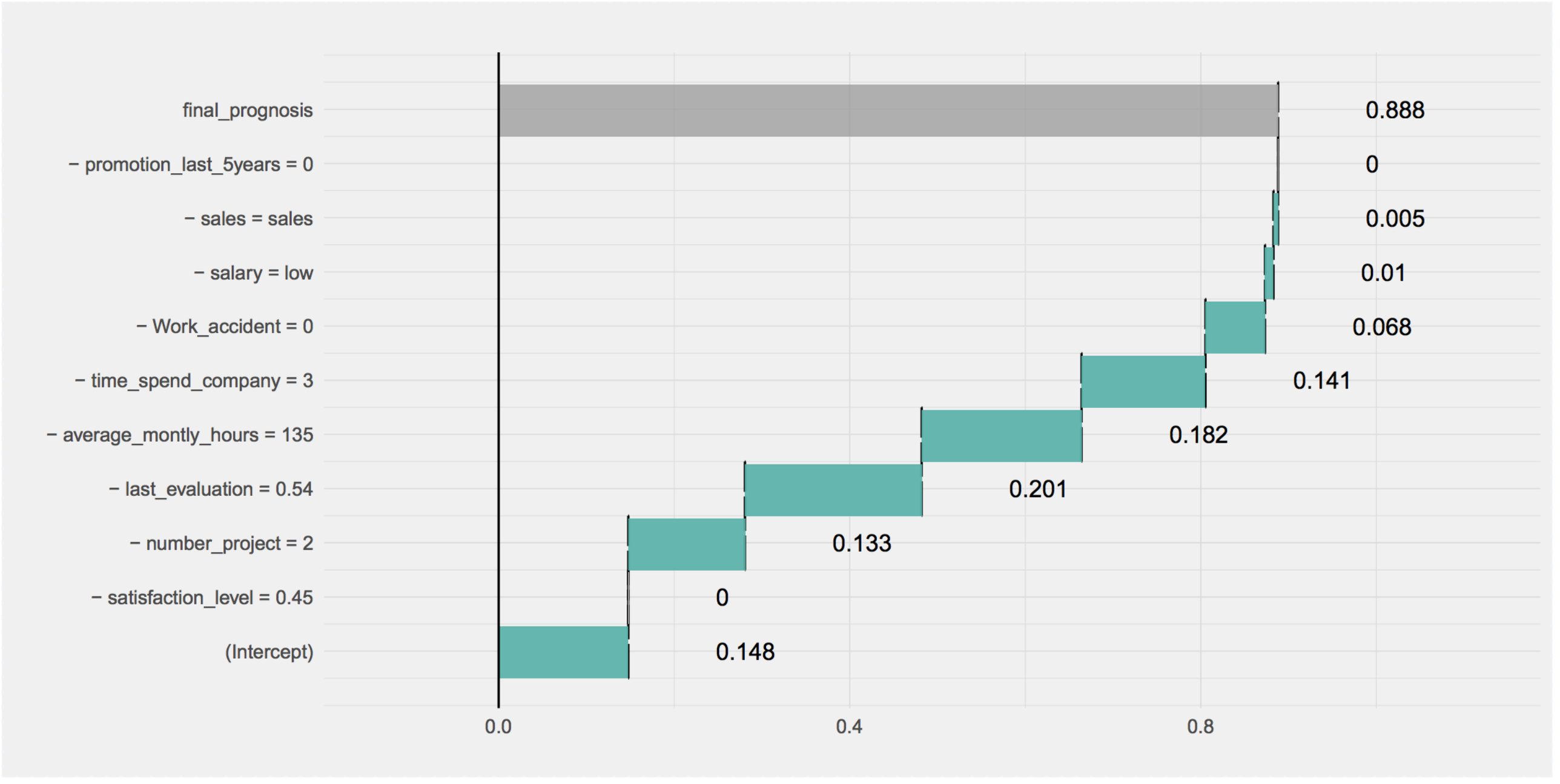
Algorithm Model agnostic break down of model predictions. The *step-down* approach

$p \leftarrow$ number of variables
$IndSet \leftarrow \{1, \ldots, p\}$ set of indexes of all variables
for i in $\{1, \ldots, p\}$ **do**
    Find new variable that can be relaxed with small loss in relaxed distance to $f(x^{new})$
    for j in $IndSet$ **do**
        Calculate relaxed distance with j removed
        $dist(j) \leftarrow d(x^{new}, IndSet \setminus \{j\})$
    end for
    Find and remove j that minimizes loss
    $j_{min} \leftarrow argmin_j dist(j)$
    $Contributions(i) \leftarrow f^{IndSet}(x^{new}) - f^{IndSet \setminus \{j_{min}\}}(x^{new})$
    $Variables(i) \leftarrow j_{min}$
    $IndSet \leftarrow IndSet \setminus \{j_{min}\}$
end for

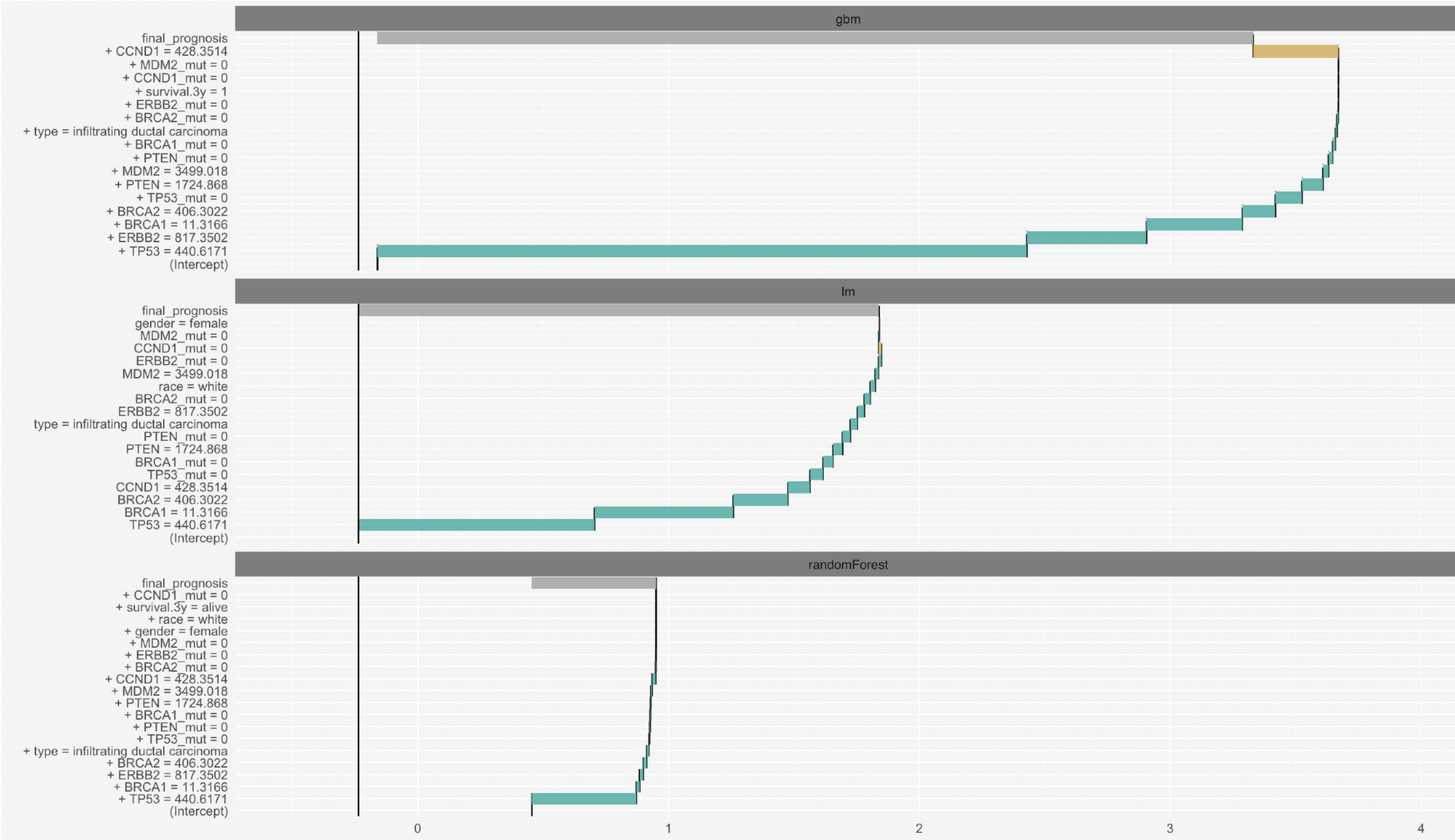
Relaxed decomposition of a single prediction



Relaxed decomposition of a single prediction

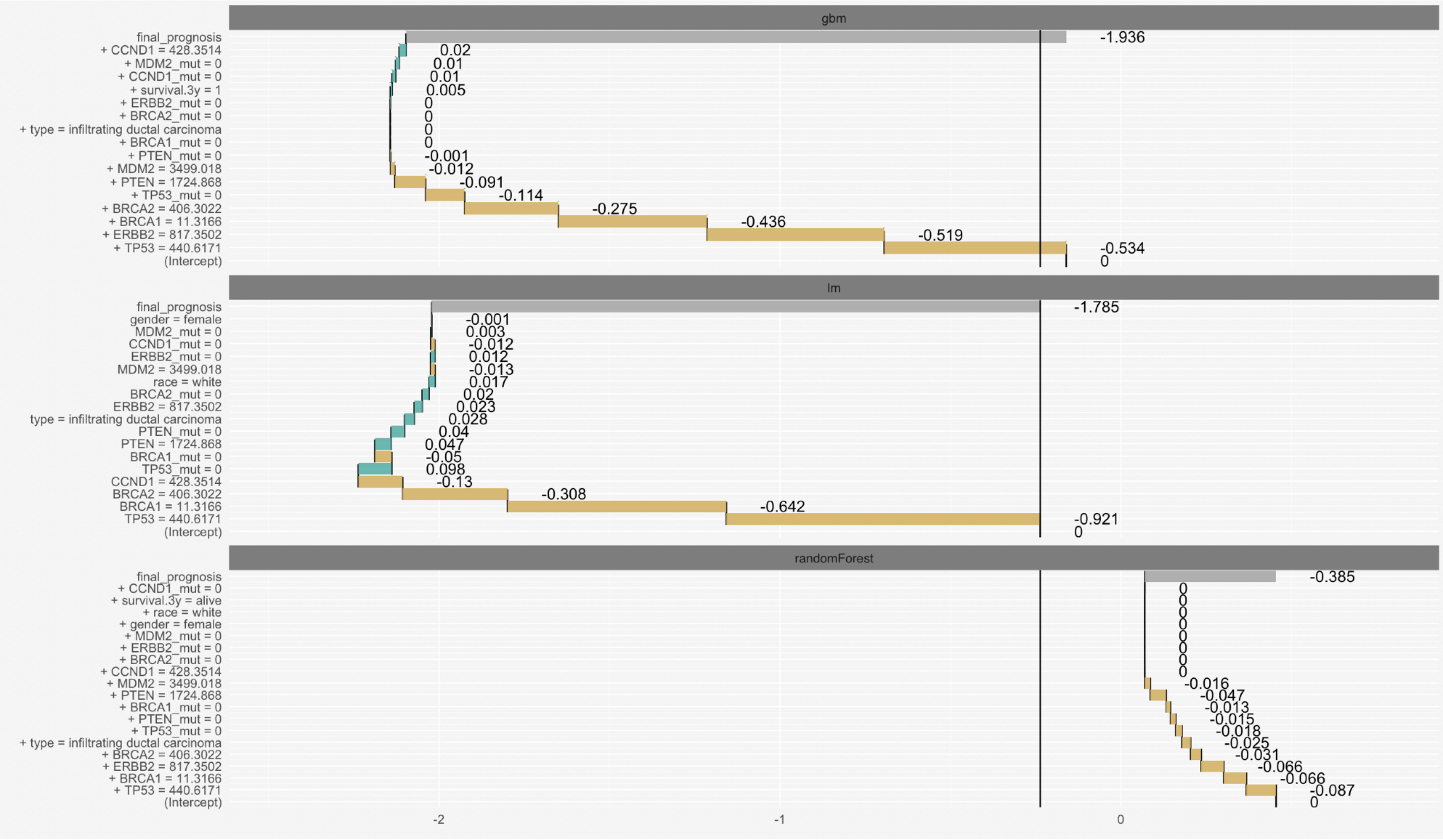


RandomForest / Gradient boosting / Logistic regression for TCGA data



log odds of being alive

RandomForest / Gradient boosting / Logistic regression for TCGA data



log odds of being alive

DALEX: : single_prediction() - Explanations for a Single Prediction

explain()

Basics

Black-box models, like random forest or extreme gradient boosting machines, are commonly used due to their high performance. They are very flexible, what often results in high accuracy.

The problem is, that due to their complicated structure it is hard to understand which variables were the most influential for a particular model prediction

Single prediction explainers are designed to decompose model prediction into parts that may be attributed for separate variables.

How does it work?
For linear models the idea is simple. Having the estimated of model coefficients the variable contributions are calculated as

$$\begin{aligned} \hat{y}^{new} &= baseline + \\ &(x_1^{new} - \bar{x}_1)\hat{\beta}_1 + \\ &... + \\ &(x_p^{new} - \bar{x}_p)\hat{\beta}_p \end{aligned}$$

where

$$baseline = \hat{\mu} + \bar{x}_1\hat{\beta}_1 + ... + \bar{x}_p\hat{\beta}_p$$

Note that variables are centred in order to have contributions calculated against the total average.

For generalised linear models we explain the linear component of the model.

For non linear models we used a method implemented in the breakDown package that assess the variable contribution based on relaxed distances between model predictions. See more details in the breakDown package vignettes.

References

- Ribeiro, Marco Tulio, Sameer Singh, and Carlos Guestrin. 2016. *"'Why Should I Trust You?': Explaining the Predictions of Any Classifier."* In, 1135–44. ACM Press.
- Staniak, Mateusz, and Przemysław Biecek. 2017. *Live: Local Interpretable (Model-Agnostic) Visual Explanations.* https://github.com/MI2DataLab/live
- Biecek, Przemyslaw. 2017. *BreakDown: BreakDown Plots.* https://CRAN.R-project.org/package=breakDown.

Use-Case

Why do you think that this will be a good wine? Let's use explanations for models prediction to explain factors that influences quality of wine. Data from UCI repository *https://archive.ics.uci.edu/ml/machine-learning-databases/wine-quality/* .

Here we are using linear model to link physicochemical properties of a wine with it's quality. Then the model is used to predict quality for a wine with selected properties.

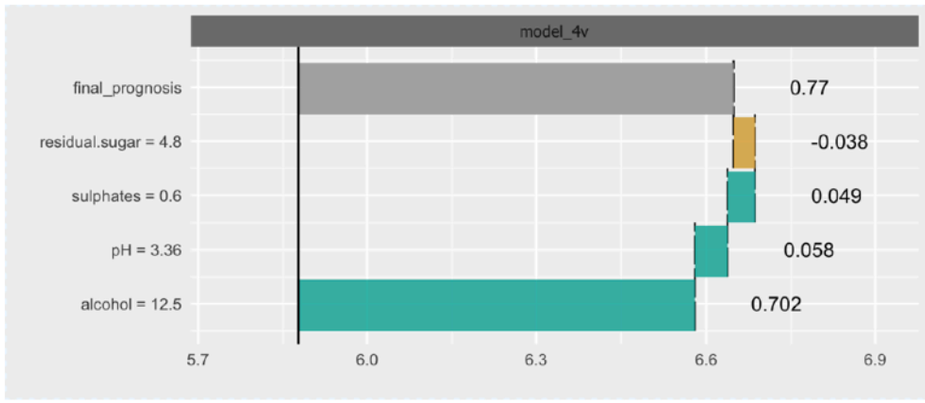
```
library("breakDown")
library("DALEX")
new.wine <- data.frame(citric.acid = 0.35,
                       sulphates = 0.6,
                       alcohol = 12.5,
                       pH = 3.36,
                       residual.sugar = 4.8)
wine_lm_model4 <- lm(quality ~ pH + residual.sugar +
                       sulphates + alcohol, data = wine)
predict(wine_lm_model4, newdata = new.wine)
## 6.648226
```

The predicted quality is high, but which properties of the wine was key for this prediction? Let's use the DALEX package to check this.
First we need a general model explainer for *wine_lm_model4* model.

```
wine_lm_explainer4 <- explain(wine_lm_model4,
                              data = wine, label = "model_4v")
```

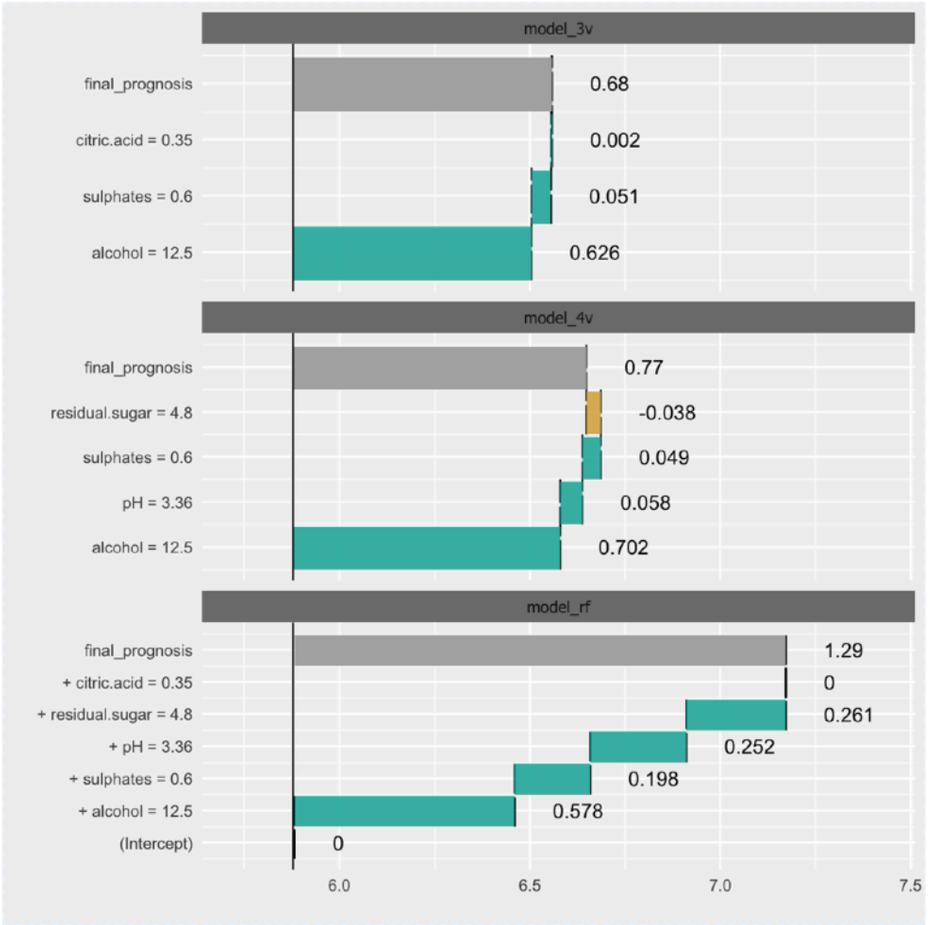
Now we can use the **single_prediction()** function to decompose model prediction into a parts that can be attributed to consecutive variables.
For linear models it's easy. In this case it's just a wrapper over the *predict* function with *type = "terms"* parameter. For nonlinear model we are using the breakDown package.
Once the decomposition is made one can use the *plot* function to present key components with the Break Down Plot.
Here we see that high alcohol concentration was crucial for model prediction.

```
wine_lm_predict4 <- single_prediction(wine_lm_explainer4,
                                      observation = new.wine)
plot(wine_lm_predict4)
```



A very useful feature of the DALEX package is the capability to overlay responses from different models in a single plot.
Below we present results for a Random Forest model and two linear models.

```
lm_model3 <- lm(quality ~ citric.acid + sulphates +
                  alcohol, data = wine)
lm_explainer3 <- explain(lm_model3, data = wine)
lm_predict3 <- single_prediction(lm_explainer3,
                   observation = new.wine)
rf_model4 <- randomForest(quality ~ pH + residual.sugar +
                   sulphates + alcohol, data = wine)
rf_explainer4 <- explain(rf_model4, data = wine)
rf_predict4 <- single_prediction(rf_explainer4,
                  observation = new.wine)
plot(rf_predict4, wine_lm_predict4, lm_predict3)
```



What is the conditional model response as a function of a single feature?

Goal of this explainer:

- plot the relation between single feature and average model output,
- compare relations across different models.

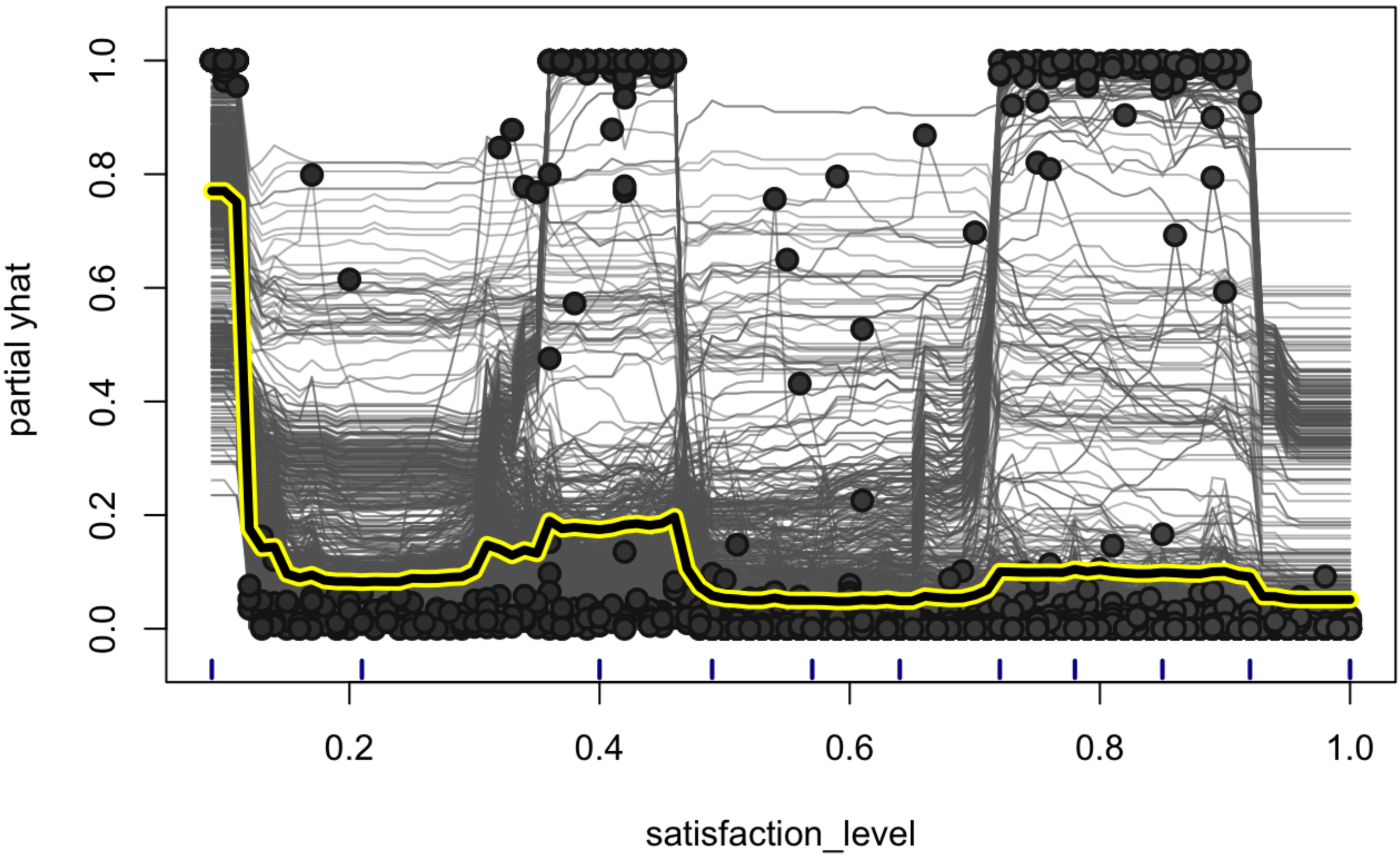
Use case:

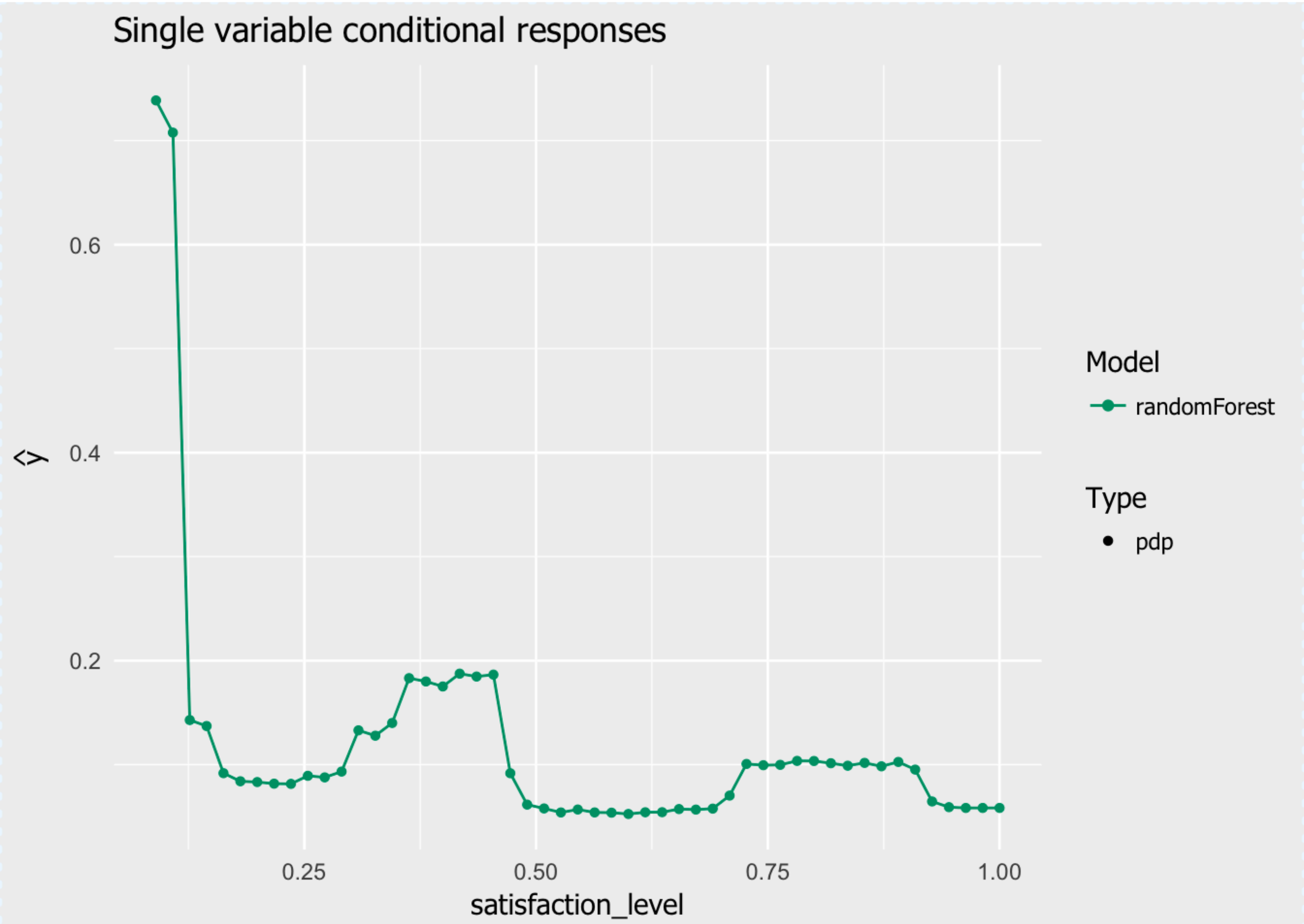
- Feature engineering.

What is the conditional model response as a function of a single feature?

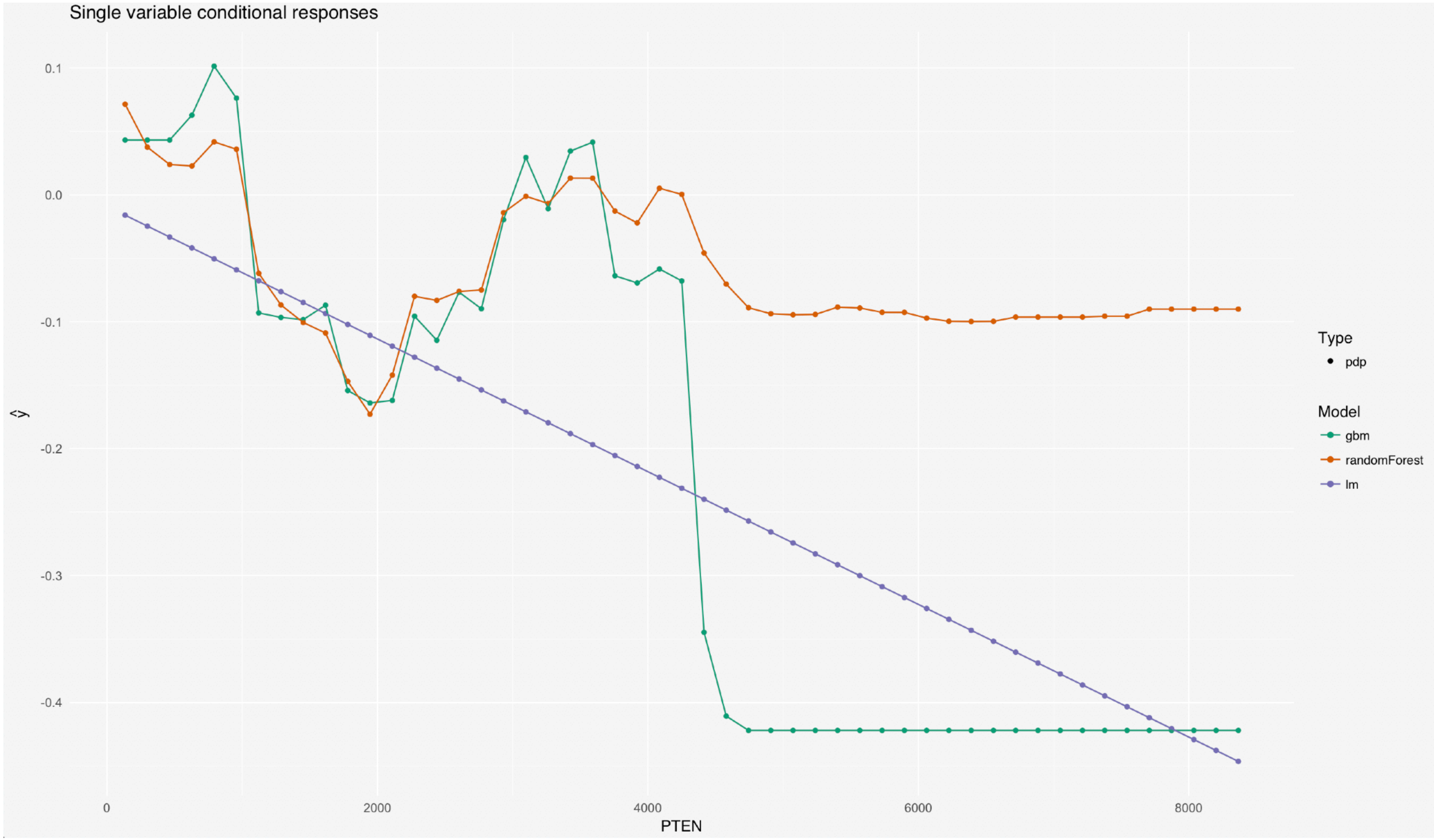
- *Partial Dependence Plots*, Greenwell 2017
- *Accumulated Local Effects Plots*, Apley 2017
- *Individual Conditional Expectations*, Goldstein , Kapelner, Bleich, Pitkin 2015
- *FactorMerger: Hierarchical Algorithm for Post-Hoc Testing,* Sitko Biecek 2017

What is the conditional model response as a function of a single feature?




Single variable conditional responses
0.6
0.4
0.2
<>
0.25
0.50
0.75
1.00
satisfaction_level
Model
randomForest
Type
pdp

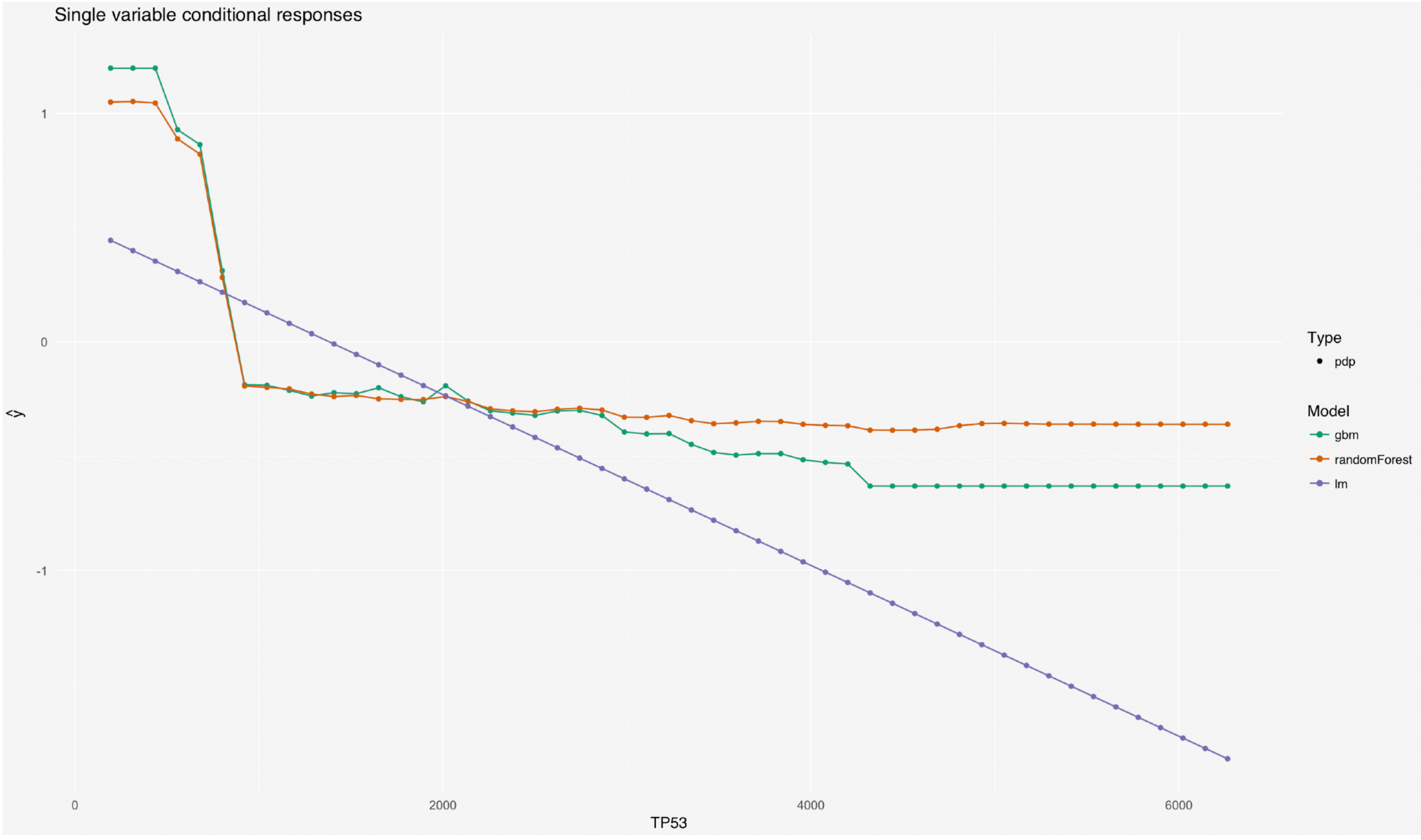
RandomForest / Gradient boosting / Logistic regression for TCGA data



log odds of being alive

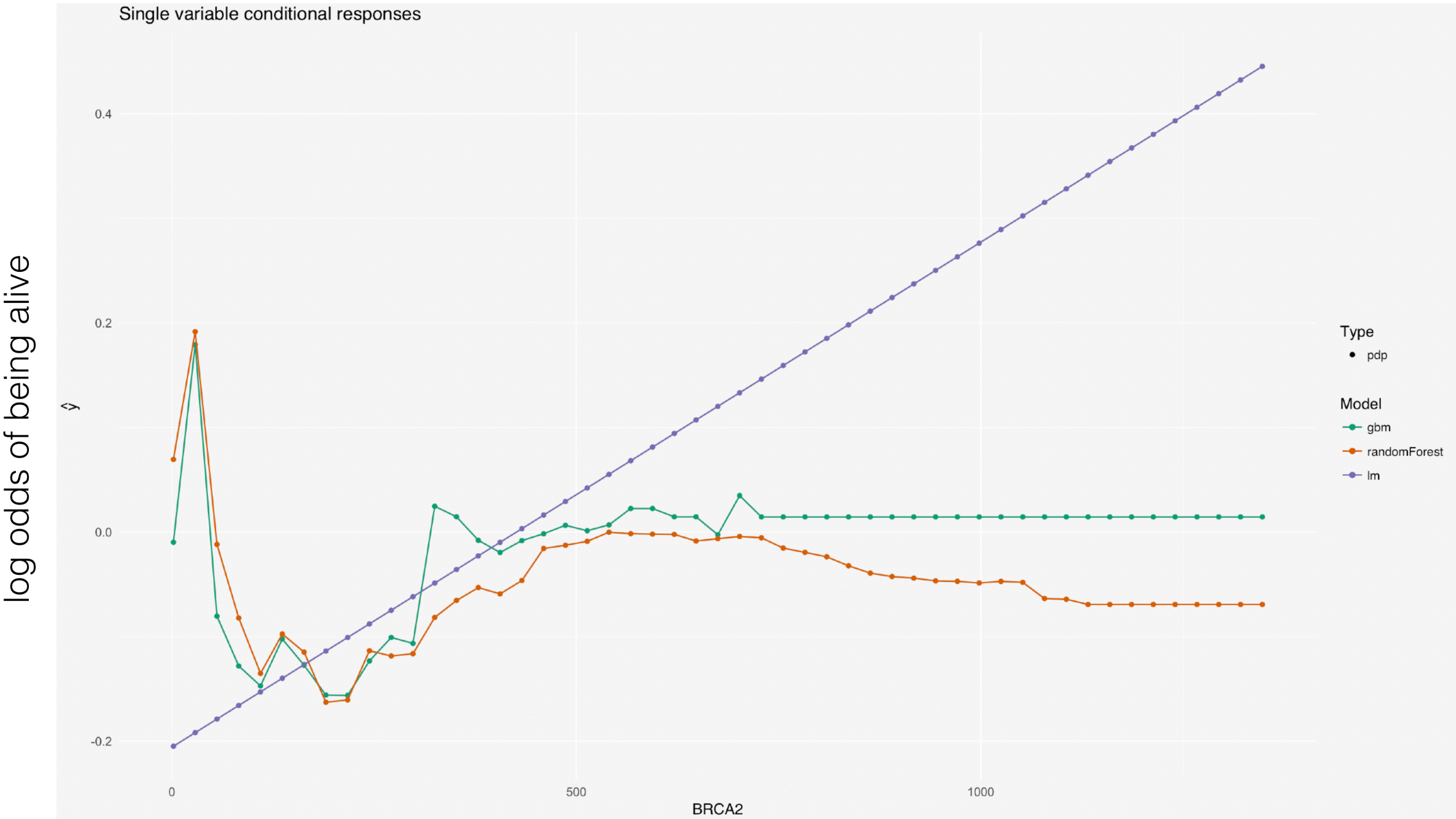
PTEN expression

RandomForest / Gradient boosting / Logistic regression for TCGA data



TP53 expression

RandomForest / Gradient boosting / Logistic regression for TCGA data



BRCA2 expression

DALEX: : single_variable() - Single Variable Conditional Model Responses

explain()

Basics

Black-box models, like random forest or extreme gradient boosting machines, are commonly used due to their high performance. They are very flexible, what often results in high accuracy.

The problem is, that due to their complicated structure it is hard to understand if and how a single variable affect model response.

Single variable explainers are designed to explain or at least interpolate the conditional effect of a single variable.

How does it work?
The idea is simple, for a selected variable x^i we would like to know what is the average output of the model as a function of this variable

$$E_{x_{-i}}[f(x^i, x^{-i}; \theta)]$$

But since we cannot calculate the expected value directly we need to estimate it.

1. Partial Dependency Plots [PDP] estimate this response as an average model response on all data points with overridden x^i.
2. Accumulated Local Effects Plots [ALEP] estimate the response as a cumulative sum of model responses calculated on data points similar to x^i under consideration.
3. Individual Conditional Expectations are model responses calculated for data points created from single observation with replaced x^i values.

Right now there are two types of single variable explainers implemented in the DALEX package: Partial Dependency Plots and Accumulated Local Effects Plots.

References

- Brandon Greenwell, *pdp: An R Package for Constructing Partial Dependence Plots*, The R Journal 9(2017), no. 1
- Apley, Dan. *ALEPlot: Accumulated Local Effects (Ale) Plots and Partial Dependence (Pd) Plots* (2017)
- Goldstein, Alex, Adam Kapelner, Justin Bleich, and Emil Pitkin. *Peeking Inside the Black Box: Visualizing Statistical Learning with Plots of Individual Conditional Expectation* Journal of Computational and Graphical Statistics (2015) 24 (1): 44–65.
- Sitko, Agnieszka, and Przemyslaw Biecek. *FactorMerger: Hierarchical Algorithm for Post-Hoc Testing (2017)* https://github.com/MI2DataLab/factorMerger.

Use-Case

Why are our best and most experienced employees leaving prematurely? Let's see with a dataset from Kaggle Human Resources competition https://www.kaggle.com/ludobenistant/hr-analytics/data.

Below we will explain how a single variable *satisfaction_level* affect the model outcome. First we will train a Random Forest classifier

```
library("randomForest")
library("breakDown")
HR_rf_model <- randomForest(left~., data = HR_data,
                            ntree = 500)
# getit: archivist::aread("pbiecek/DALEX/arepo/a79f3c7e")
```

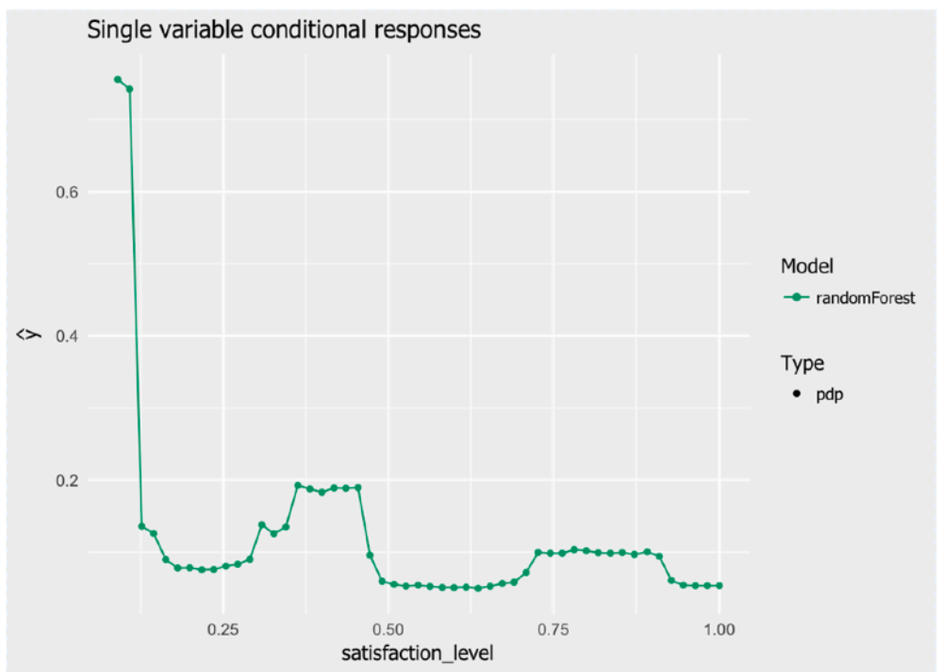
In order to generate explanations we need to convert the model into a uniform structure readable by the DALEX package. To do so we need to use the **explain()** function

```
library("DALEX")
explainer_rf  <- explain(HR_rf_model, data = HR_data)
```

Single variable explainer can be created with the **single_variable()** function. It's result can be printed or plotted. Use the type= parameter to choose PDP/ ALEP effects.

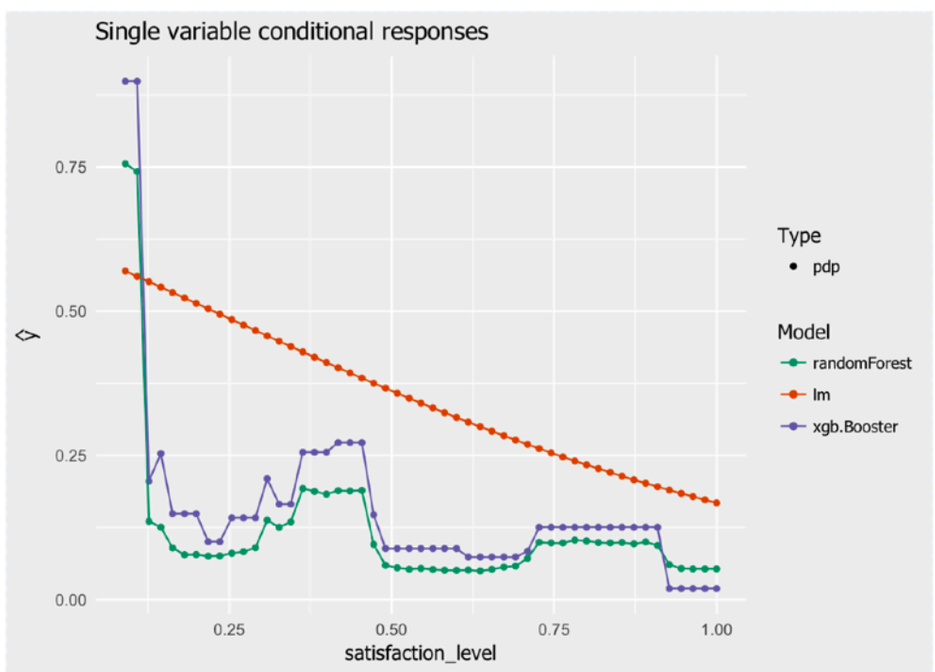
```
expl_rf  <- single_variable(explainer_rf,
            variable = "satisfaction_level",type = "pdp")
plot(expl_rf)
# getit: archivist::aread("pbiecek/DALEX/arepo/ad0f1699")
```

Result is a ggplot2 plot which present conditional profile of model responses as a function of satisfaction_level variable.



A very useful feature of the DALEX package is the possibility to overlay responses from different models in a single plot.
Below we present results for Random Forest, XGBoost model and Logistic regression model in a single plot. As we see XGB and RF are close to each other while GLM tries to find linear link.

```
library("xgboost")
model_martix_train <- model.matrix(left~.-1, HR_data)
data_train <- xgb.DMatrix(model_martix_train,
                label = HR_data$left)
param <- list(max_depth=2, objective="binary:logistic")
HR_xgb_model <- xgb.train(param, data_train, nrounds=50)
# GLM model
explainer_glm <- explain(HR_glm_model, data=HR_data)
expl_glm <- single_variable(explainer_glm,
              "satisfaction_level", "pdp")
#XGB model
explainer_xgb <- explain(HR_xgb_model,
                  data = model_martix_train)
expl_xgb <- single_variable(explainer_xgb,
              "satisfaction_level", "pdp")
# plot results
plot(expl_rf, expl_glm, expl_xgb)
```

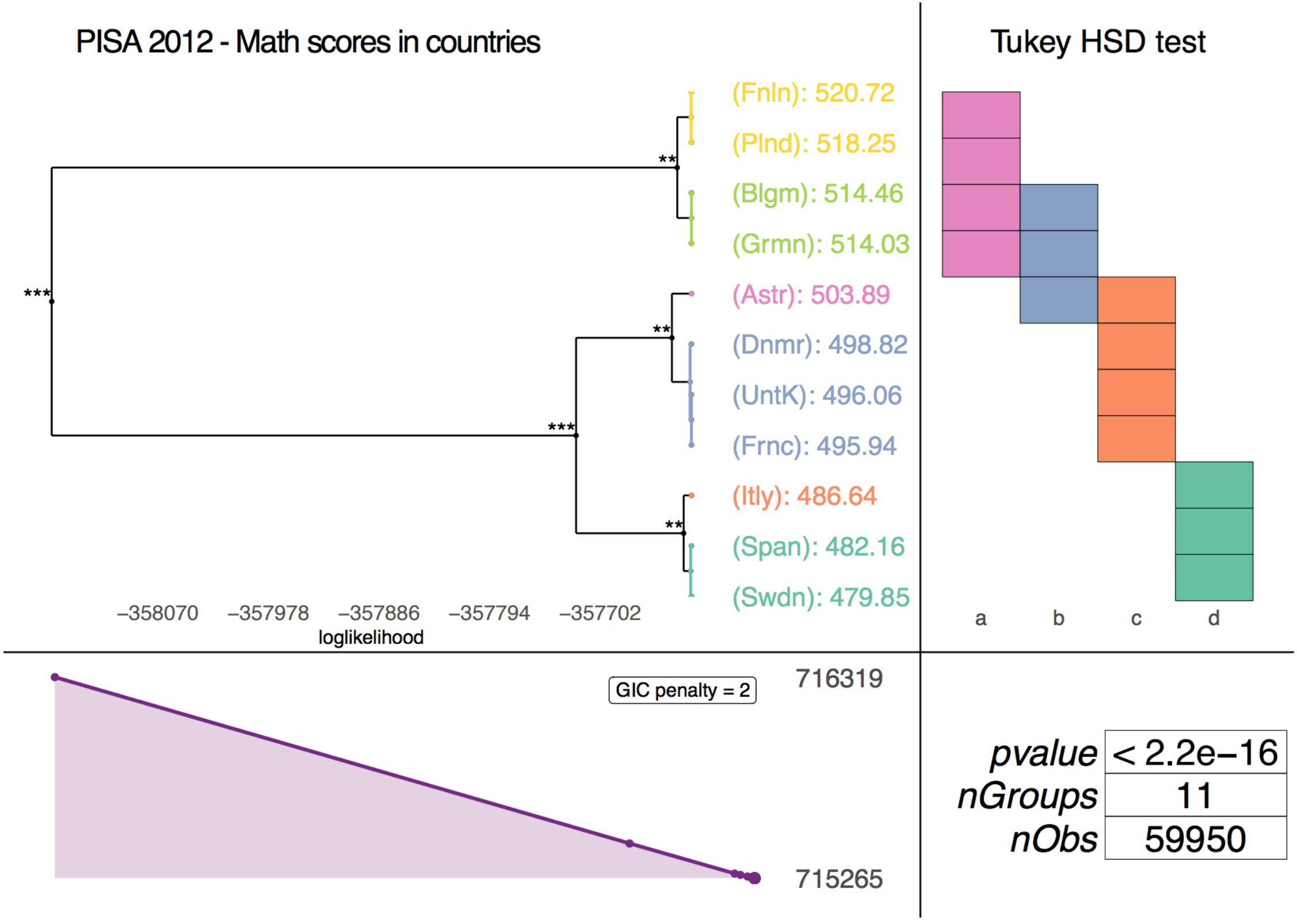


What about categorical dependent variable?

Algorithm 1 The outline of the Merging Path Plot algorithm implemented in **factorMerger**

function MERGEFACTORS($responseVariable, groupingVariable, adjacent$)
  $currentModel := createModel(responseVariable, groupingVariable)$
  $mergingPath := list(currentModel)$
  while $|levels(groupingVariable)| \geq 1$ **do**
    $pairsSet := generatePairs(groupingVariable, responseVariable, adjacent)$
    $selectedPair := \text{argmax}_{\text{pair} \in \text{pairsSet}} objectiveFunction(pair, responseVariable, groupingVariable)$
    $groupingVariable := mergeLevels(groupingVariable, selectedPair)$
    $currentModel := createModel(responseVariable, groupingVariable)$
    $mergingPath := add(mergingPath, currentModel)$
  end while
  return(mergingPath)
end function

The Merging Path Plot: adaptive fusing of k-groups with likelihood-based model selection
Agnieszka Sitko, Przemysław Biecek (2017)
https://arxiv.org/abs/1709.04412
https://github.com/geneticsMiNIng/FactorMerger



The Merging Path Plot: adaptive fusing of k-groups with likelihood-based model selection
Agnieszka Sitko, Przemysław Biecek (2017)
https://arxiv.org/abs/1709.04412
https://github.com/geneticsMiNIng/FactorMerger

factorMerger
Cheat Sheet

Agnieszka Sitko [aut, cre]
Przemysław Biecek [aut, ths]
University of Warsaw

Introduction

How to check if averages are different among k groups? Use **ANOVA**!

How to visualise how these groups are different? Use **factorMerger**!

The aim of **factorMerger** is to provide informative and easy to understand visualisations of *post-hoc* comparisons. It gives consistent and non-overlapping adaptive fusing of groups based on likelihood ratio test (LRT). The package **factorMerger** works for wide spectrum of families like Gaussian, binomial or survival.

Results from the adaptive fusing are presented with the *Merging Paths Plots* - a hierarchical representation of LRT-based distances among groups.

In addition, the *Generalized Information Criterion (GIC)* is presented for fused models. This criterion may be used to choose the optimal segmentation of groups.

Graphical summary of the variable of interest in each group is presented in the right panel.

Find more in *https://arxiv.org/abs/1709.04412*

Example

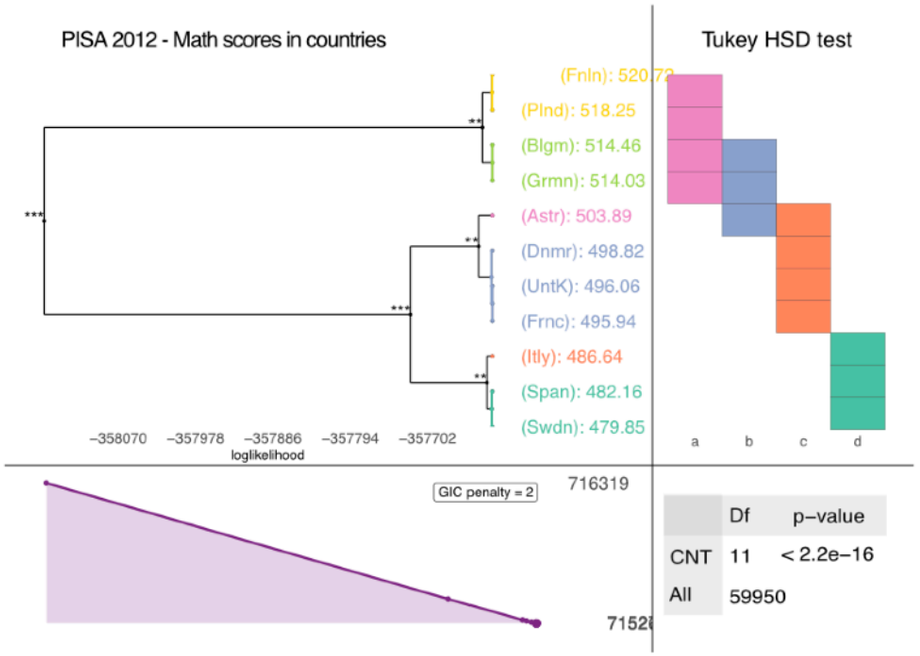
```
library(factorMerger)

fmAll <- mergeFactors(
    response  = pisaEuro$math,
    factor = pisaEuro$country,
    method = "fast-adaptive",
    family = "gaussian")

print(fmAll)

plot(fmAll,
    panel = "all",
    responsePanel = "tukey")
```

Set of tools to support post-hoc segmentation of groups

Four panels to see it all

A. The Merging Paths Plot panel
Shows the hierarchical structure of similarity between groups.
Stars presents how significant are differences between two clusters.
Height of the join corresponds to the likelihood of the combined model.

B. The response panel
Shows graphical summaries for group separately.
Use the responsePanel argument to select the desired summary.



C. The GIC panel
Shows the *Generalized Information Criterion* for all models in MPP panel.
Default *penalty=2* corresponds to the AIC criterion.
Colours in the MPP panel corresponds to the optimal model.

D. The summary panel
Shows the no. groups, no. observations,
and the p-value for global hypothesis
that parameter of interest is equall in all groups.

Key arguments of factorMerger

The **factorMerger** package is highly customisable.

The **panel** argument selects panels to be plotted.

The **nodesSpacing** argument selects how MPP nodes shall be spaced.

The **responsePanel** argument selects which summary shall be presented in panel B.

The **responsePanelPalette** argument selects the colouring palette.

The **colorClusters** argument selects if segments of groups shall be colour coded.

The **penalty** argument selects penalty for GIC criterion.

Argument	Summary
`panel = "all"`	All panels
`panel = "response"`	Panels A and B
`panel = "GIC"`	Panels A and C
`panel = "tree"`	Only panel A

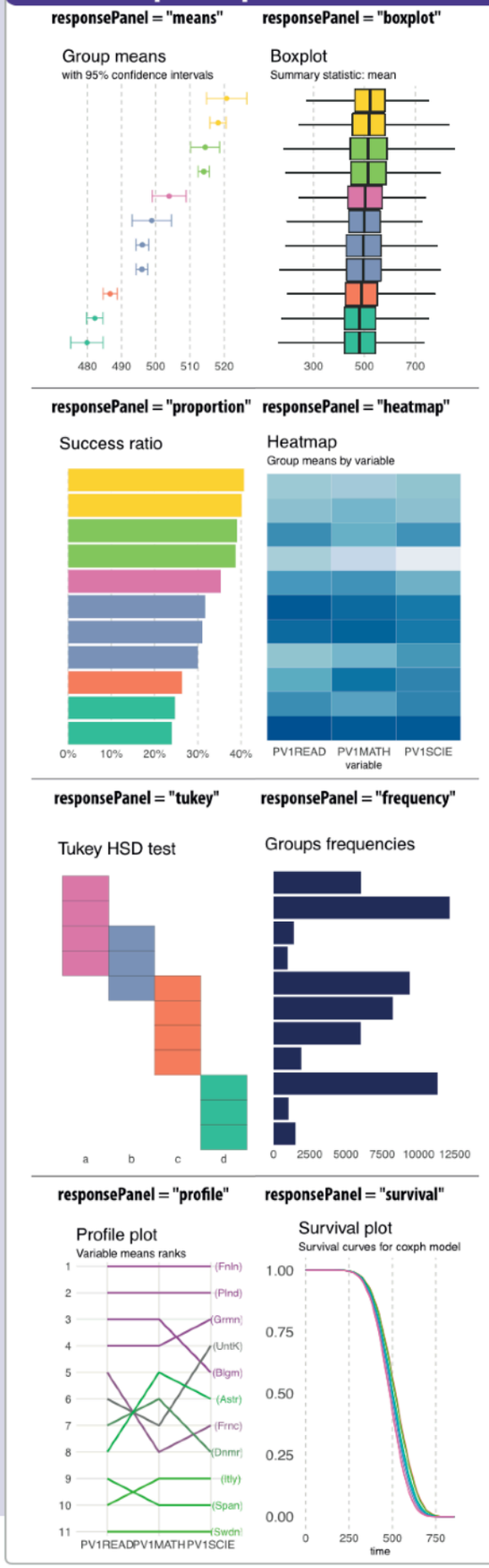
The **method** argument determines how pairs of groups are considered for grouping. The most computational demanding is *adaptive*. For larger number of groups one may consider only pairs of groups with closest averages or use the *fast-fixed* heuristic (fastest solution).

Argument	Summary
`method = "adaptive"`	Compare all pairs of groups
`method = "fast-adapive"`	Compare nearby groups
`method = "fixed"`	Delete or merge heuristic
`method = "fast-fixed"`	Reduced DMR method

The **family** argument determines what models are to be compared with likelihood ratio test. Either linear models, generalised linear models or Cox regression models.

Argument	Summary
`family = "gaussian"`	For 1/n-dimensional Gaussian
`family = "binomial"`	For binomial regression
`family = "survival"`	For Cox regression

The response panel summaries



Which features are the most influential?

Goal of this explainer:

- show feature importance,
- compare feature importance across different models.

Use case:

- Feature selection.

Which features are the most influential?

- Permutation based feature importance, Breiman 2001

- *Model Class Reliance: Variable Importance Measures,* Fisher, Rudin, Dominici 2018

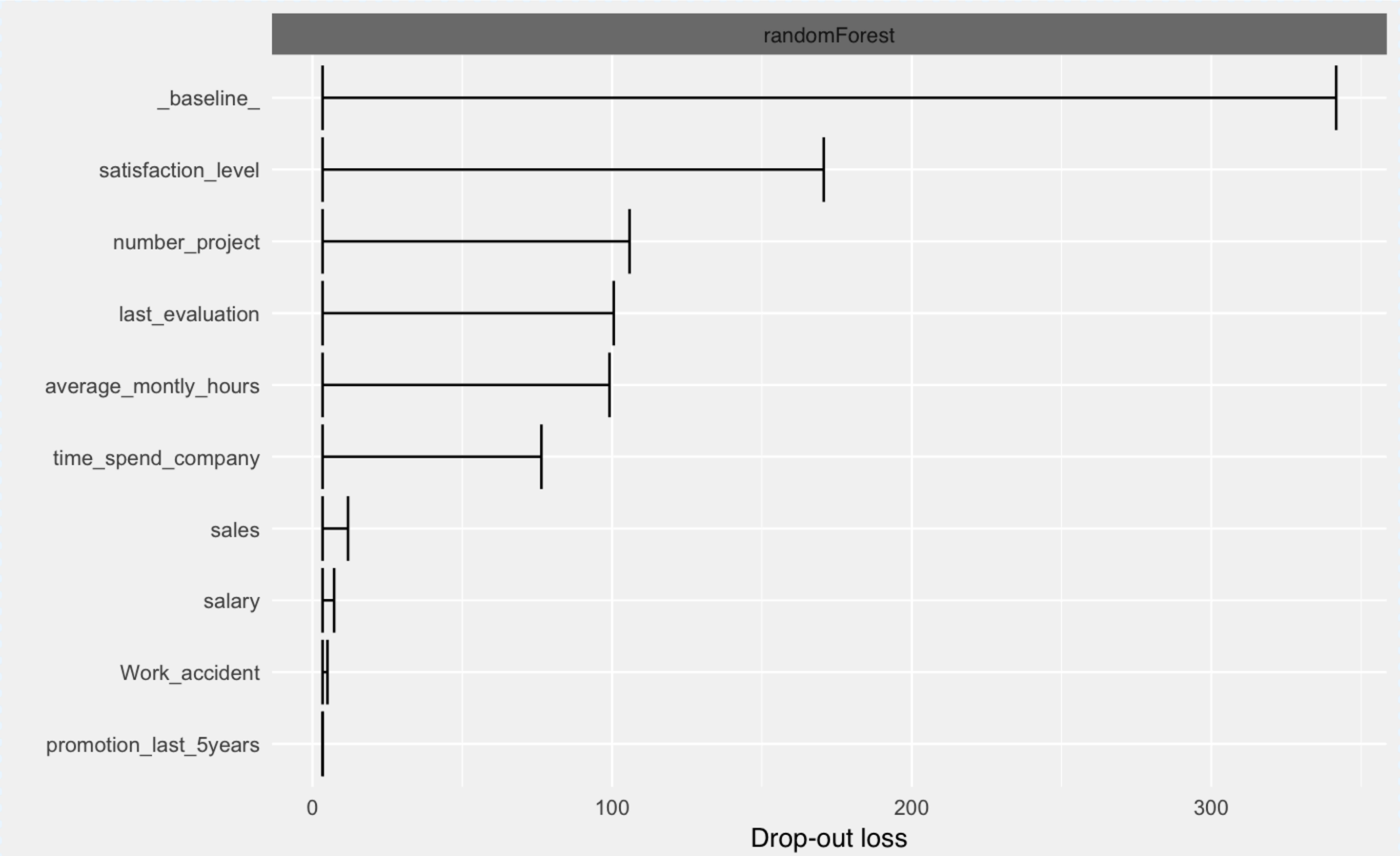
Which features are the most influential?

$$\hat{e}_{\text{orig}}(f) := \frac{1}{n}\sum_{i=1}^{n} L(f, \mathbf{Z}_{[i,\cdot]}) = \frac{1}{n}\sum_{i=1}^{n} L\{f, (\mathbf{y}_{[i]}, \mathbf{X}_{1[i,\cdot]}, \mathbf{X}_{2[i,\cdot]})\} = \hat{\mathbb{E}} L(f, Z),$$

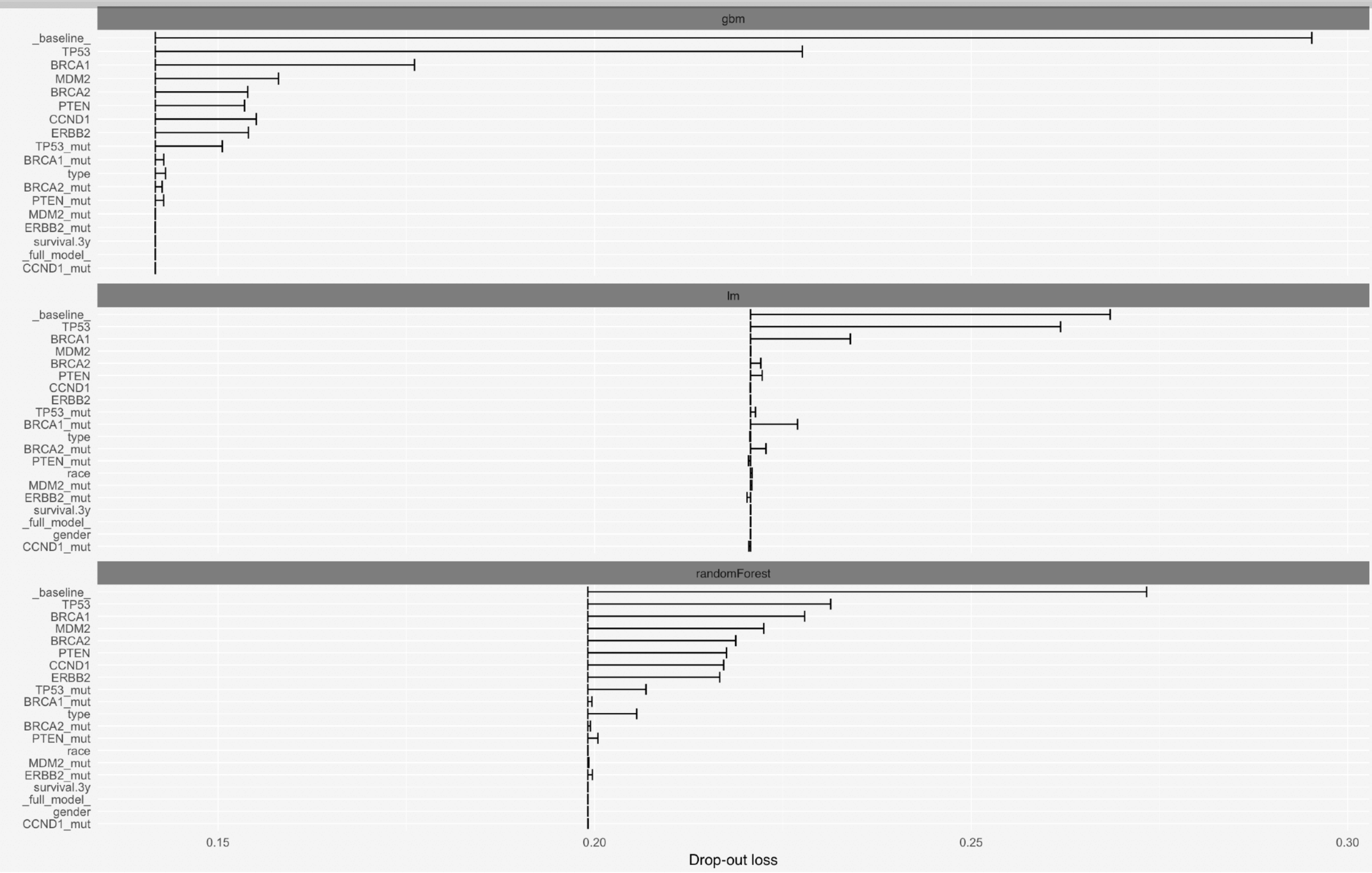
$$\begin{aligned}\hat{e}_{\text{switch}}(f) &:= \frac{1}{n(n-1)}\sum_{i=1}^{n}\sum_{j\neq i} h_f(\mathbf{Z}_{[i,\cdot]}, \mathbf{Z}_{[j,\cdot]}) \\ &= \frac{1}{n(n-1)}\sum_{i=1}^{n}\sum_{j\neq i} L\{f, (\mathbf{y}_{[j]}, \mathbf{X}_{1[i,\cdot]}, \mathbf{X}_{2[j,\cdot]})\}.\end{aligned}$$

Model Class Reliance $:= \dfrac{\hat{e}_{\text{switch}}(f)}{\hat{e}_{\text{orig}}(f)}$,

Model Class Reliance: Variable Importance Measures for any Machine Learning Model Class, from the “Rashomon” Perspective
Aaron Fisher, Cynthia Rudin, Francesca Dominici


randomForest
baseline
satisfaction_level
number_project
last_evaluation
average_montly_hours
time_spend_company
sales
salary
Work_accident
promotion_last_5years
0
100
200
300
Drop-out loss

RandomForest / Gradient boosting / Logistic regression for TCGA data



DALEX: : variable_dropout() - Explanations for Variable Importance

Basics

Black-box models, like random forest or extreme gradient boosting machines, are commonly used due to their high performance. They are very flexible, what often results in high accuracy.

The problem is, that due to their complicated structure it is hard to understand which variables were the most influential for a particular model prediction

Variable Dropouts are designed to assess the influence of a single variable on the final model accuracy.

How does it work?
The concept is straightforward, calculate how much we will loose on accuracy is a selected variable is perturbed.

For a selected loss function the model loss is calculated for the trained model applied to the original dataset against the original target. Then the model drop out loss is calculated for each variable in the dataset. It is calculated as the loos for a model applied to a dataset with selected column being permuted.

The variable dropout plots shows the initial loss of the trained model, size of the additional losses that come from permutations of selected variables and the 'baseline' which is the loss for permuted model responses.

It is useful when different model are compared, since on a single plot we see the initial model performance and also drops that come from permutations of a single variable in the selected model.

References

- Molnar, Christoph. 2018. *Interpretable Machine Learning*. https://christophm.github.io/
- Breiman, Leo. 2001. *Random Forests.* Machine Learning 45 (1). Springer: 5–32.
- Fisher, Aaron, Cynthia Rudin, and Francesca Dominici. 2018. *Model Class Reliance: Variable Importance Measures for any Machine Learning Model Class, from the 'Rashomon' Perspective.* http://arxiv.org/abs/1801.01489.

Use-Case

Why are our best and most experienced employees leaving prematurely? Let's see with a dataset from Kaggle Human Resources competition https://www.kaggle.com/ludobenistant/hr-analytics/data.

Here we are building a random forest model. The nice thing about this model is that it embeds some variable importance measure.

```
library("randomForest")
HR_rf_model <- randomForest(left~., data =
breakDown::HR_data, ntree = 100)
importance(HR_rf_model)
                     IncNodePurity
satisfaction_level       896.584411
last_evaluation          294.195269
number_project           496.397115
average_montly_hours     398.495343
```

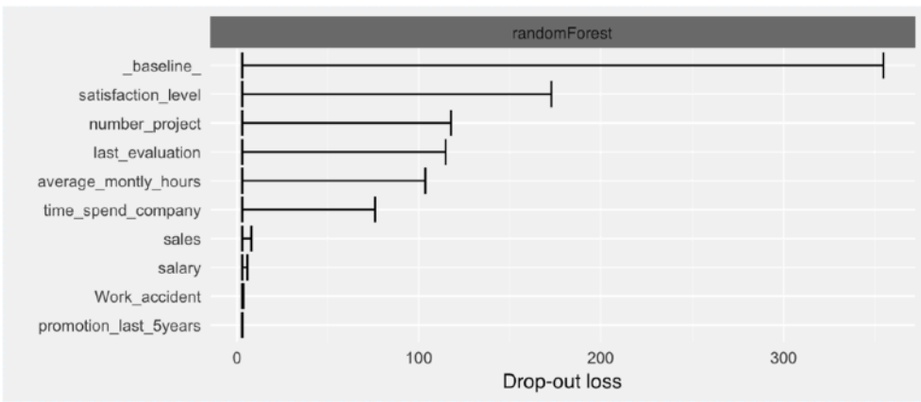
So, now let's build an explainer and use it to calculate our model agnostic variable importance measure with **variable_dropout()** function.

```
explainer_rf  <- explain(HR_rf_model, data = HR_data,
                         y = HR_data$left)
vd_rf <- variable_dropout(explainer_rf, type = "raw")
vd_rf
                 variable dropout_loss        label
                _baseline_   335.556063 randomForest
      satisfaction_level   168.001733 randomForest
         last_evaluation   100.714534 randomForest
          number_project   100.568214 randomForest
       time_spend_company    90.349207 randomForest
```

Values are different because different measures of importance are being calculated, but the order is more or less the same. Now we are ready to plot the variable importance with the generic **plot**() function.

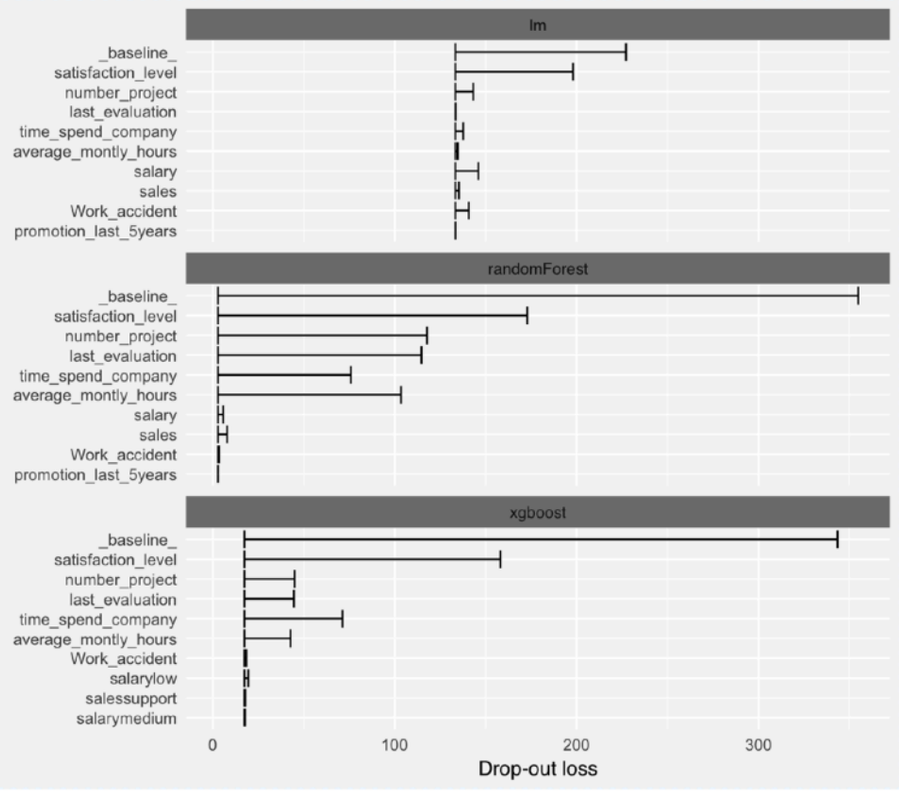
Note that in addition to *type="raw"* one can also use *"difference"* or *"ratio"* as suggested in the Fisher et all 2018 article.

```
plot(vd_rf)
```



A very useful feature of the DALEX package is the capability to overlay responses from different models in a single plot.
Below we present results for a Random Forest, GLM and XGboost.

```
library("xgboost")
mm_train <- model.matrix(left~.-1, HR_data)
data_train <- xgb.DMatrix(model_martix_train,
                label = HR_data$left)
param <- list(max_depth = 2,
      objective = "binary:logistic", eval_metric = "auc")
HR_xgb <- xgb.train(param, data_train, nrounds = 50)
ex_xgb <- explain(HR_xgb,  data = mm_train,
                  y = HR_data$left, label = "xgboost")
vd_xgb <- variable_dropout(ex_xgb, type = "raw")

HR_glm <- glm(left~., data=HR_data, family = "binomial")
ex_glm <- explain(HR_glm, data=HR_data, y = HR_data$left)
logit <- function(x) exp(x)/(1+exp(x))
vd_glm <- variable_dropout(ex_glm, type = "raw")
plot(vd_rf, vd_glm, vd_xgb)
```



Key points:

* Machine Learning Models have lots of very interesting applications,
* We need better tools to validate and understand ML models, otherwise we should not trust their results,
* DALEX is a model agnostic library that helps to better understand factors that influence ML models responses.

Thank you!

References:

A. Altmann, L. Tolo ̧si, O. Sander, and T. Lengauer. Permutation importance: a corrected feature importance measure. Bioinformatics, 26(10):1340–1347, May 2010.

D. W. Apley. Visualizing the Effects of Predictor Variables in Black Box Supervised Learning Models. ArXiv e-prints, Dec. 2016

B. Bischl, M. Lang, L. Kotthoff, J. Schiffner, J. Richter, E. Studerus, G. Casalicchio, and Z. M. Jones. mlr: Machine learning in r. Journal of Machine Learning Research, 17(170):1–5, 2016.

J. H. Friedman. Greedy function approximation: A gradient boosting machine. Ann. Statist., 29(5):1189– 1232, 10 2001

A. Goldstein, A. Kapelner, J. Bleich, and E. Pitkin. Peeking Inside the Black Box: Visualizing Statistical Learning with Plots of Individual Conditional Expectation. ArXiv e-prints, Sept. 2013.

A. Goldstein, A. Kapelner, J. Bleich, and E. Pitkin. Peeking Inside the Black Box: Visualizing Statistical
Learning With Plots of Individual Conditional Expectation. Journal of Computational and Graphical Statistics, 24(1):44–65, Jan. 2015.

B. M. Greenwell. pdp: An R Package for Constructing Partial Dependence Plots. The R Journal, 9(1): 421–436, 2017.

S. Marco Tulio Ribeiro and Carlos Guestrin. "why should i trust you?" explaining the predictions of any classifier. KDD ’16 Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 1135–1144, 2016.

T. L. Pedersen and M. Benesty. lime: Local Interpretable Model-Agnostic Explanations, 2017. URL https://CRAN.R-project.org/package=lime. R package version 0.3.1.

N. Puri, P. Gupta, P. Agarwal, S. Verma, and B. Krishnamurthy. MAGIX: Model Agnostic Globally
Interpretable Explanations. ArXiv e-prints, June 2017.

C. Strobl, A.-L. Boulesteix, T. Kneib, T. Augustin, and A. Zeileis. Conditional variable importance for random forests. BMC Bioinformatics, 9(307), 2008.

M. Tulio Ribeiro, S. Singh, and C. Guestrin. Nothing Else Matters: Model-Agnostic Explanations By Identifying Prediction Invariance. ArXiv e-prints, Nov. 2016a.

E. Štrumbelj and I. Kononenko. A general method for visualizing and explaining black-box regression models. In Proceedings of the 10th International Conference on Adaptive and Natural Computing Algorithms - Volume Part II, ICANNGA’11, pages 21–30, Berlin, Heidelberg, 2011. Springer-Verlag.

archivist : : record, restore and govern your R objects

Everything that exists in R is an object. **archivist** is an R package that stores copies of all objects along with their metadata. It helps to manage and recreate objects with final or partial results from data analysis.

Use the archivist to record every result, to share these results with future you or with others, to search through repository of objects created in the past but needed now.

Key functionalities include:

i) management of local and remote repositories which contain R objects and their meta-data (objects' properties and relations between them);
ii) archiving R objects to repositories;
iii) sharing and retrieving objects (and their pedigree) by their unique hooks;
iv) searching for objects with specific properties or relations to other objects;
v) verification of object's identity and context of its creation.

Record artifacts

R objects with partial or final results are called artifacts. They may be either tables, models, plots or any other structures.
Artifacts are stored in repositories. One repositories may be either local or remote.

- local repository is a folder with write/read access,
- remote repositories are usually available through http and have only read access.

Use the **createLocalRepo()** function to create an empty local repository.

```
library(archivist)
createLocalRepo("arepo")
```

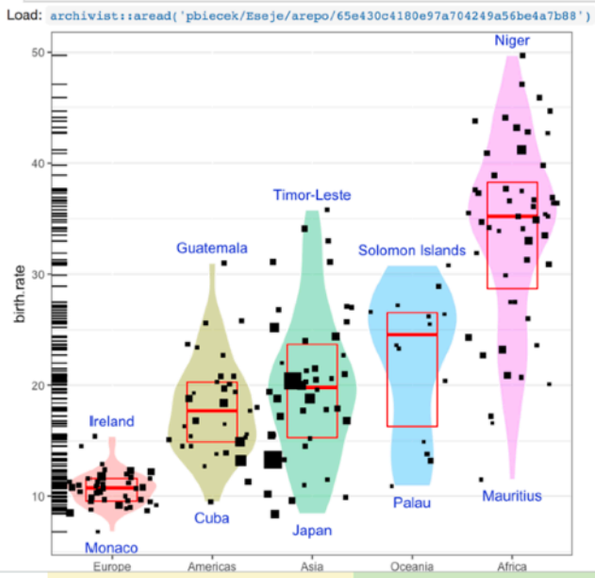
Use the **saveToRepo()** function to store selected R objects to a local repository. Each object is stored along with rich metadata (class, date of archivisation) and unique MR5 hash. All related objects, like data, are also stored in the repository.

```
model <- lm(len ~ supp+dose,
            data = ToothGrowth)
saveToRepo(model, repoDir = "arepo")
[1] "7c6b38019d2e028b0fb38fc0ed44a175"
attr(,"data")
[1] "1563e36bc621d97b9a736c63f1ccf5cb"
```

Share artifacts

Use **loadFromLocalRepo()** or **loadFromRemote Repo()** or more compact **aread()** functions to retrieve artifacts from repositories.
It's a good idea to attach hooks to key artifacts in reports or articles. The command below restores a ggplot2 object from GitHub `pbiecek/Eseje`.

```
aread('pbiecek/Eseje/arepo/65e430c41')
```



Browse artifacts

Use **searchInLocalRepo()** or **searchInRemoteRepo**() to filter out artifacts that matches selected citeria. Any meta-data collected for the artifact may be used for searching.
The more compact function with smaller number of arguments is **asearch()**.

The code below downloads and calculates BIC scores for all artifacts with tag `class:lm` (linear models) from the remote GitHub repository `pbiecek/graphGallery`.

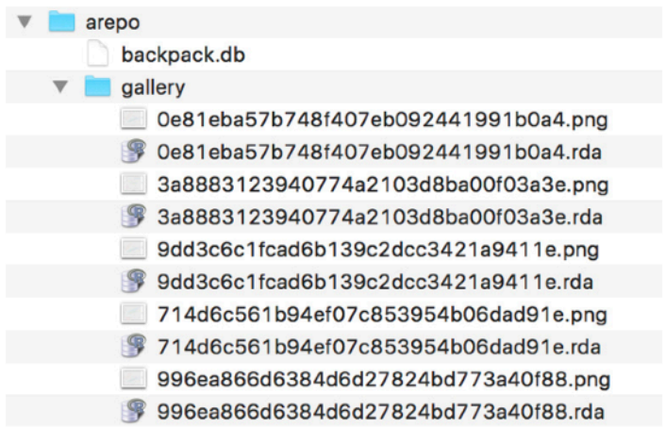
```
models <- asearch("pbiecek/graphGallery",
          patterns = "class:lm")
modelsBIC <- sapply(models, BIC)
sort(modelsBIC)
990861c7c27812ee959f10e5f76fe2c3
                        39.05577
2a6e492cb6982f230e48cf46023e2e4f
                        67.52735
0a82efeb8250a47718cea9d7f64e5ae7
                       189.73593
```

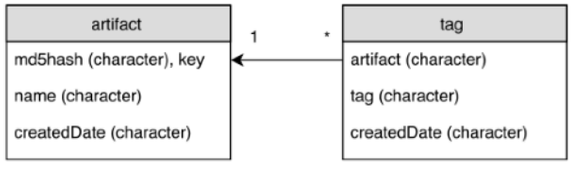
Use **asession()** to retrieve a session info object related to the selected artifact. Use **ahistory()** to retrieve list of commands used to create the artifact.

Architecture

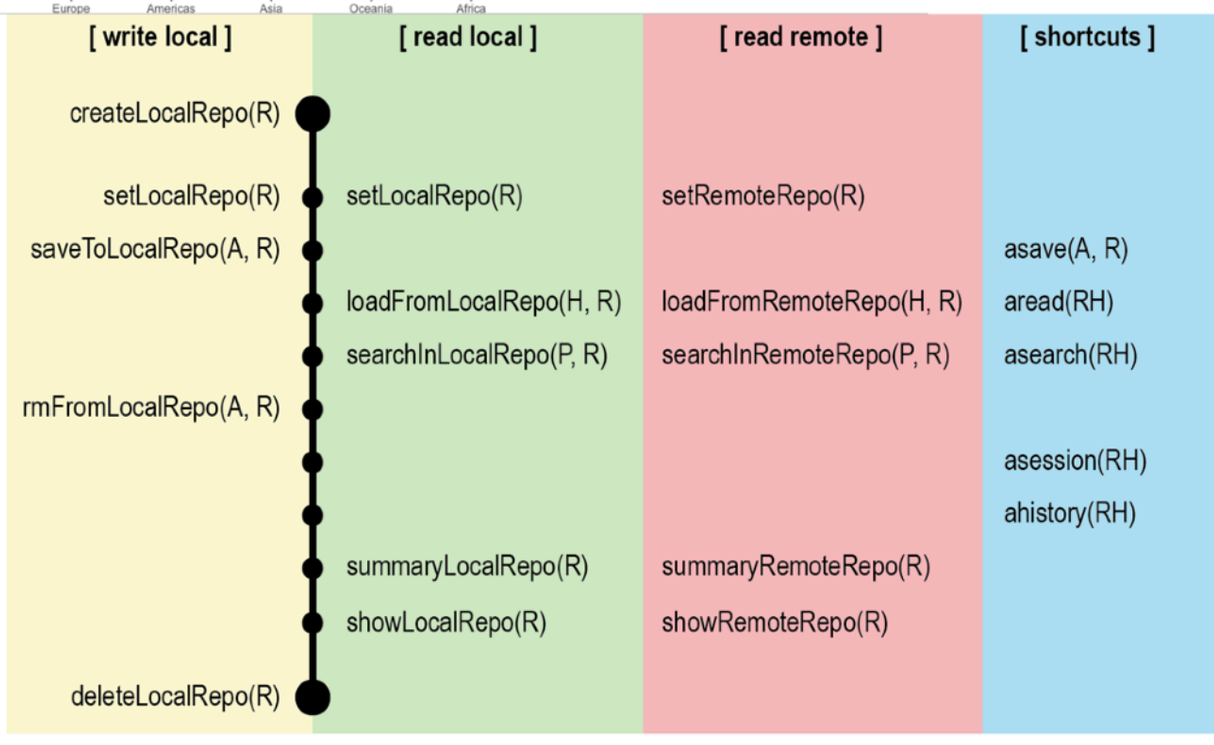
An archivist repository is a folder with following components:

- a **backpack.db** file with SQLite database that stores artifacts metadata, tags and relations between artifacts. One can also use other databases like PostgreSQL.
- a subfolder **gallery** with binary copies of artifacts. Each archivist is stored as a raw rda file with name that corresponds to it's MD5 hash
- small graphical or text miniatures that summaries each artifact. These are stored as txt or png files.





Meta-data about artifacts is stored in the database in two tables: *artifact* and *tag*. Various tags are attached to an artifact. These tags store information about class, variable names, session info, relation among artifacts and other predefined or user defined tags. They are useful to trace and recover artifacts.



A - artifact, any R object, like data.frame, ggplot, lm
H - md5hash, cryptographical hash of arbitrary R object
P - pattern, used to find artifacts with suitable tags

R - repository, a local repository is a folder, a remote repo is a based on git or hg. Repository contains rda dumps, miniatures and data base with object's tags.

Przemysław Biecek and Marcin Kosinski (2017). "archivist: An R Package for Managing, Recording and Restoring Data Analysis Results." *Journal of Statistical Software*, **82** (11), pp. 1-28. doi: 10.18637/jss.v082.i11 • Learn more at **http://pbiecek.github.io/archivist/**

```
ggplot(countries, aes(x=continent, y=birth.rate, label=country)) +
  geom_violin(scale="width", aes(fill=continent), color="white", alpha=0.4) +
  stat_summary(fun.data = "q3", geom = "crossbar",
               colour = "red", width = 0.5) +
  geom_jitter(aes(size=(population)^0.9),position=position_jitter(width = .45, height = 0),
              shape=15) +
  geom_rug(sides = "l") +
  geom_text(data=countriesMin, vjust=2, color="blue3") +
  geom_text(data=countriesMax, vjust=-1, color="blue3") +
  theme_bw() + xlab("") + theme(legend.position="none", panel.grid.major.x = element_line(color="whi
te"))
```

Load: `archivist::aread('pbiecek/Eseje/arepo/ba7f58fafe7373420e3ddce039558140')`