LIVE and breakDown: explainers for single prediction

Mateusz Staniak
Munich, 25.05.2018

breakDown

CRAN 0.1.5 | downloads 636/month | downloads 3050 | build passing Pending Pull-Requests Github Issues

Break Down Plots: Model Agnostic Explainers for Individual Predictions

`breakDown` package decomposes individual predictions into parts attributed to particular variables. See vignettes for more examples.

Bookdown website for `breakDown` package: https://pbiecek.github.io/breakDown/

Methodology behind prediction explainers implemented in `breakDown` and `live`: https://arxiv.org/abs/1804.01955

How to install

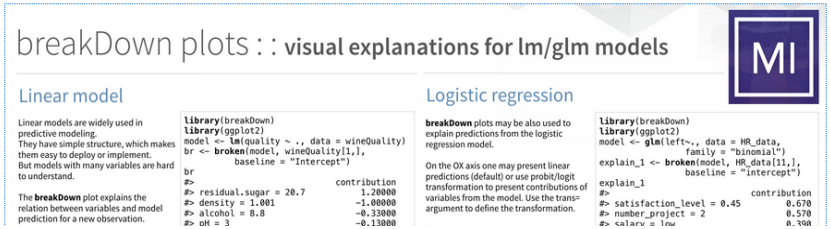
Install from GitHub

```
devtools::install_github("pbiecek/breakDown")
```

Install from CRAN

```
install.packages("breakDown")
```

Cheatsheets



DALEX

CRAN 0.2.2 | downloads 1731 | build passing Coverage Status

Descriptive mAchine Learning EXplanations

DALEX Stories

- A gentle introduction to DALEX with examples
- How to use DALEX with caret
- How to use DALEX with mlr
- How to use DALEX with xgboost package
- Talk about DALEX at Complexity Institute / NTU February 2018
- Talk about DALEX at SER / WTU April 2018
- How to use DALEX for teaching. Part 1

Install

From GitHub

```
# dependencies
devtools::install_github("MI2DataLab/factorMerger")
devtools::install_github("pbiecek/breakDown")

# DALEX package
devtools::install_github("pbiecek/DALEX")
```

or from CRAN

```
install.packages("DALEX")
```

broken() / single_prediction()

BreakDown: idea

For linear models:

$$f(x^{new}) = (1, x^{new})(\mu, \beta)^T = baseline + (x_1^{new} - \bar{x}_1)\beta_1 + ... + (x_p^{new} - \bar{x}_p)\beta_p$$

$$baseline = \mu + \bar{x}_1\beta_1 + ... + \bar{x}_p\beta_p.$$

Contribution: $(x_1^{new} - \bar{x}_1)\beta_1$

Model-agnostic contribution:

Definition 2.4.3 (Added feature contribution) *For j-th feature we define its contribution relative to a set of indexes IndSet (added contribution) as*

$$contribution^{IndSet}(j) = f^{IndSet \cup \{j\}}(x^{new}) - f^{IndSet}(x^{new}). \quad (8)$$

It is the change in model prediction for x^{new} after relaxation on j.

Model-agnostic breakDown

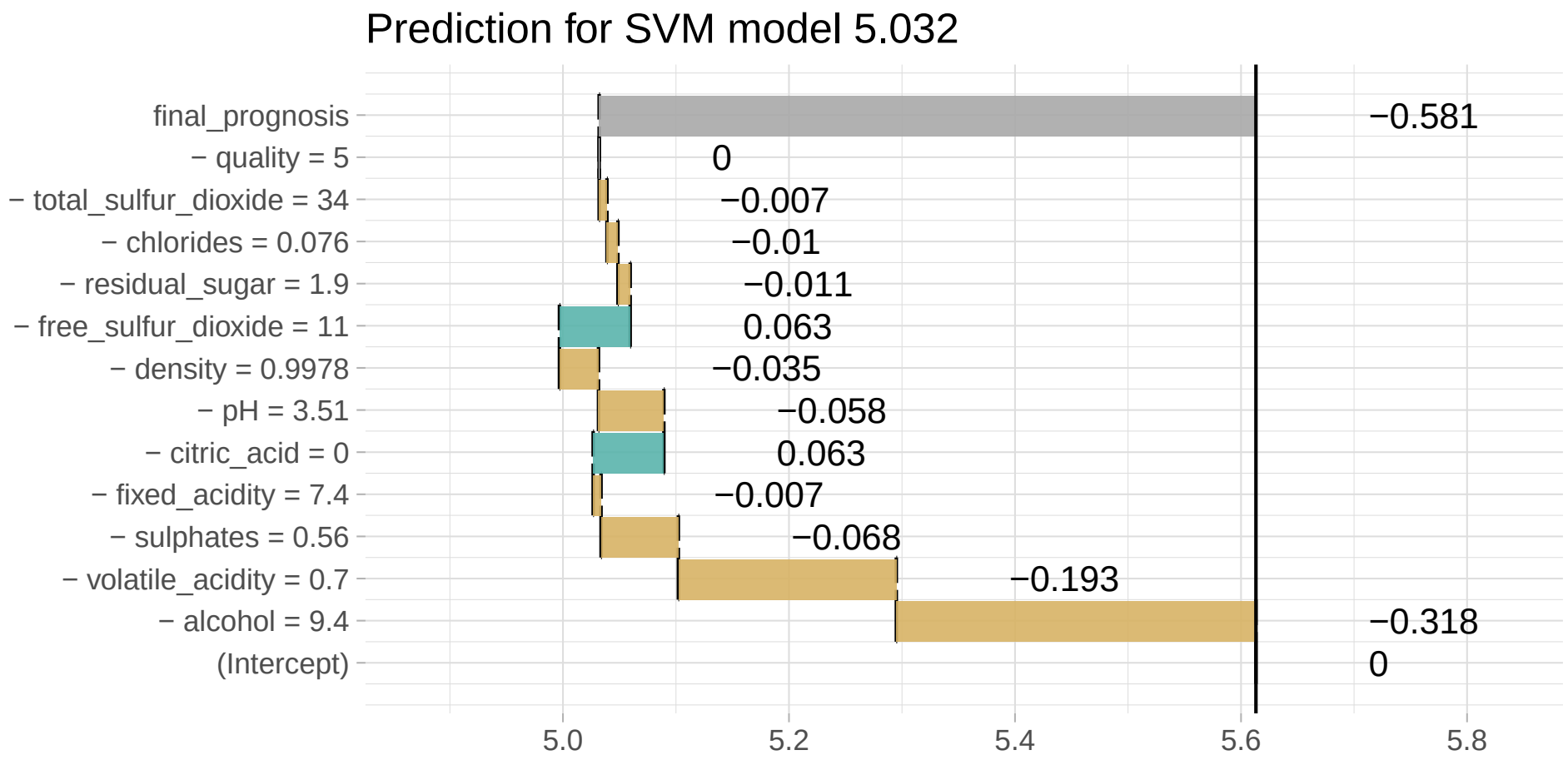
Step-up

$p \leftarrow$ 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
    $j_{min} \leftarrow \arg\min_j dist(j)$
    $Contribution^{IndSet}(i) \leftarrow f^{IndSet}(x^{new}) - f^{IndSet \setminus \{j_{min}\}}(x^{new})$
    $Variables(i) \leftarrow j_{min}$
    $IndSet \leftarrow IndSet \setminus \{j_{min}\}$
end for

Step-down

$p \leftarrow$ number of variables
$IndSet \leftarrow \emptyset$ empty set
for i in $\{1, \ldots, p\}$ **do**
    Find new variable that can be relaxed with large distance to $f^{\emptyset}(x^{new})$
    for j in $\{1, \ldots, p\} \setminus IndSet$ **do**
        Calculate relaxed distance with j added
        $dist(j) \leftarrow d(x^{new}, IndSet \cup \{j\})$
    end for
    Find and add j that maximize distance
    $j_{max} \leftarrow \arg\max_j dist(j)$
    $Contribution^{IndSet}(i) \leftarrow f^{IndSet \cup \{j_{max}\}}(x^{new}) - f^{IndSet}(x^{new})$
    $Variables(i) \leftarrow j_{max}$
    $IndSet \leftarrow IndSet \cup \{j_{max}\}$
end for

Waterfall plots



Future of breakDown

- ➔ Sparse explanations
- ➔ Non-additive contributions
- ➔ Going from local to global

LIVE

live: Local Interpretable (Model-agnostic) Visual Explanations

Installation

Install stable CRAN version:

```
install.packages("live")
```

or the development version:

```
devtools::install_github("MI2DataLab/live")
```

See the latest changes.

Features coming up next:

- more methods of sampling,
- better support for comparing explanations for different models / different instances,
- improved Shiny application (see `live_shiny` function in development version).

If you have any bug reports, feature requests or ideas to improve the methodology, feel free to leave an issue.

Materials

Find the paper about `live` and breakDown on arXiv.

Website: https://mi2datalab.github.io/live/

Conference talk on `live`: https://github.com/mstaniak/Berlin_2017

LIVE: idea

LIME for regression / tabular data

Focus on model visualization

LIME: advantages

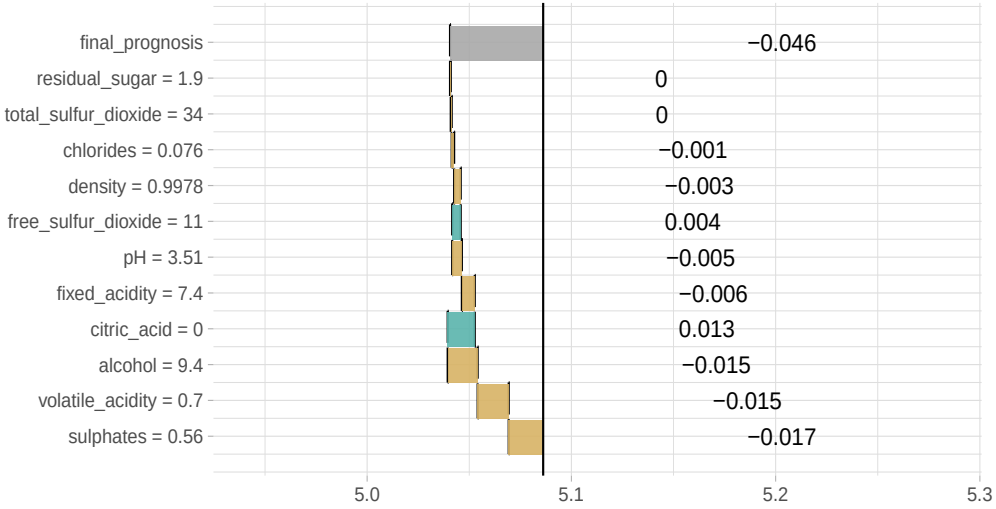
- Different methods of sampling
- High flexibility regarding e.g. explanation model
- Built-in visualization tools for linear explanation

LIVE: work flow

```
sample_locally() %>%
  add_predictions() %>%
    fit_explanation() %>%
      plot()
```

Variable	N	Estimate		p
fixed_acidity	500		0.10 (0.08, 0.12)	<0.001
volatile_acidity	500		−1.47 (−1.64, −1.29)	<0.001
citric_acid	500		−0.54 (−0.64, −0.44)	<0.001
residual_sugar	500		0.01 (−0.04, 0.06)	0.664
chlorides	500		1.12 (0.32, 1.91)	0.006
free_sulfur_dioxide	500		−0.01 (−0.01, −0.00)	<0.001
total_sulfur_dioxide	500		0.00 (−0.00, 0.00)	0.417
density	500		−27.45 (−41.49, −13.41)	<0.001
pH	500		−0.29 (−0.42, −0.16)	<0.001
sulphates	500		1.19 (1.08, 1.30)	<0.001
alcohol	500		0.23 (0.21, 0.26)	<0.001

−40 −30 −20 −10 0



Future of LIVE

- Going from local to global
- More theoretical background

  (E.g. How to pick sample size? How to generate neighbourhoods?)
- More visualization tools

Thank you for you attention!