

DrWhy.AI - Tools for Explainable Artificial Intelligence

Hubert Baniecki

MI2 DataLab, Warsaw University of Technology

AISC, 11.11.2020



DALEX

moDel Agnostic Language for Exploration and eXplanation

 Python  663  100

DrWhy

DrWhy is the collection of tools for eXplainable AI (XAI). It's based on shared principles and simple grammar for exploration, explanation and visualisation of predictive models.

 R  398  51

modelStudio

 Interactive Studio for Explanatory Model Analysis

 R  138  17



```
hbaniecki:~$ whoami
```

Research Software Engineer at Data Lab lead by Przemysław Biecek

Data Science Student at Warsaw University of Technology

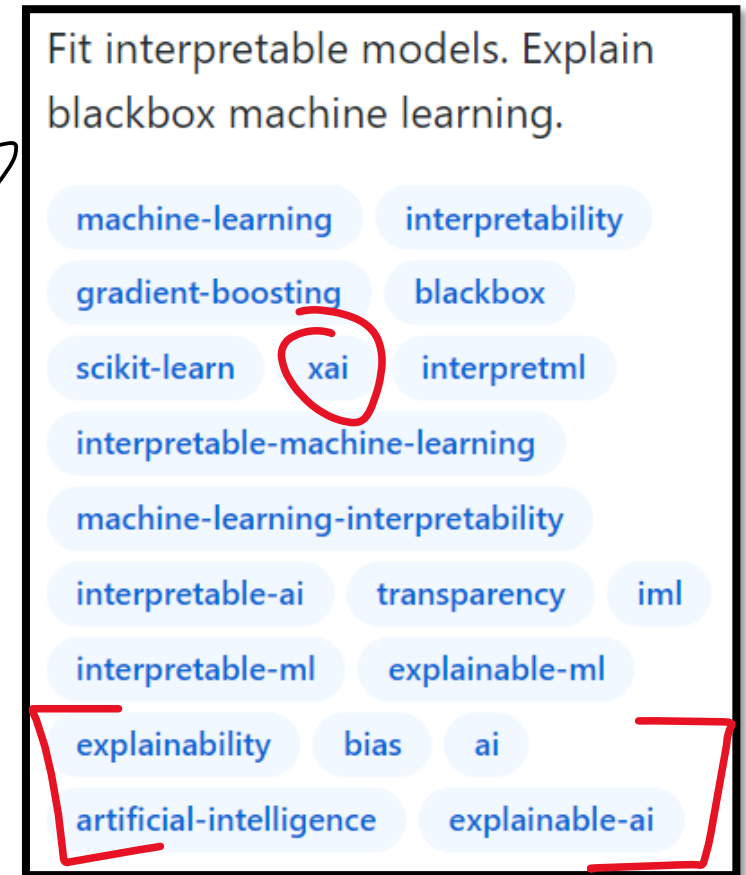
Interested in Explainable AI and model-human interaction

Developing and maintaining the DrWhy.AI universe

Packages: DALEX & modelStudio & more

The semantics of Explainable AI (XAI)

- **IBM:** A set of capabilities and methods used to describe an AI model, its expected impact and potential biases.
- **Microsoft:** *model interpretability*



Explainable AI (XAI)

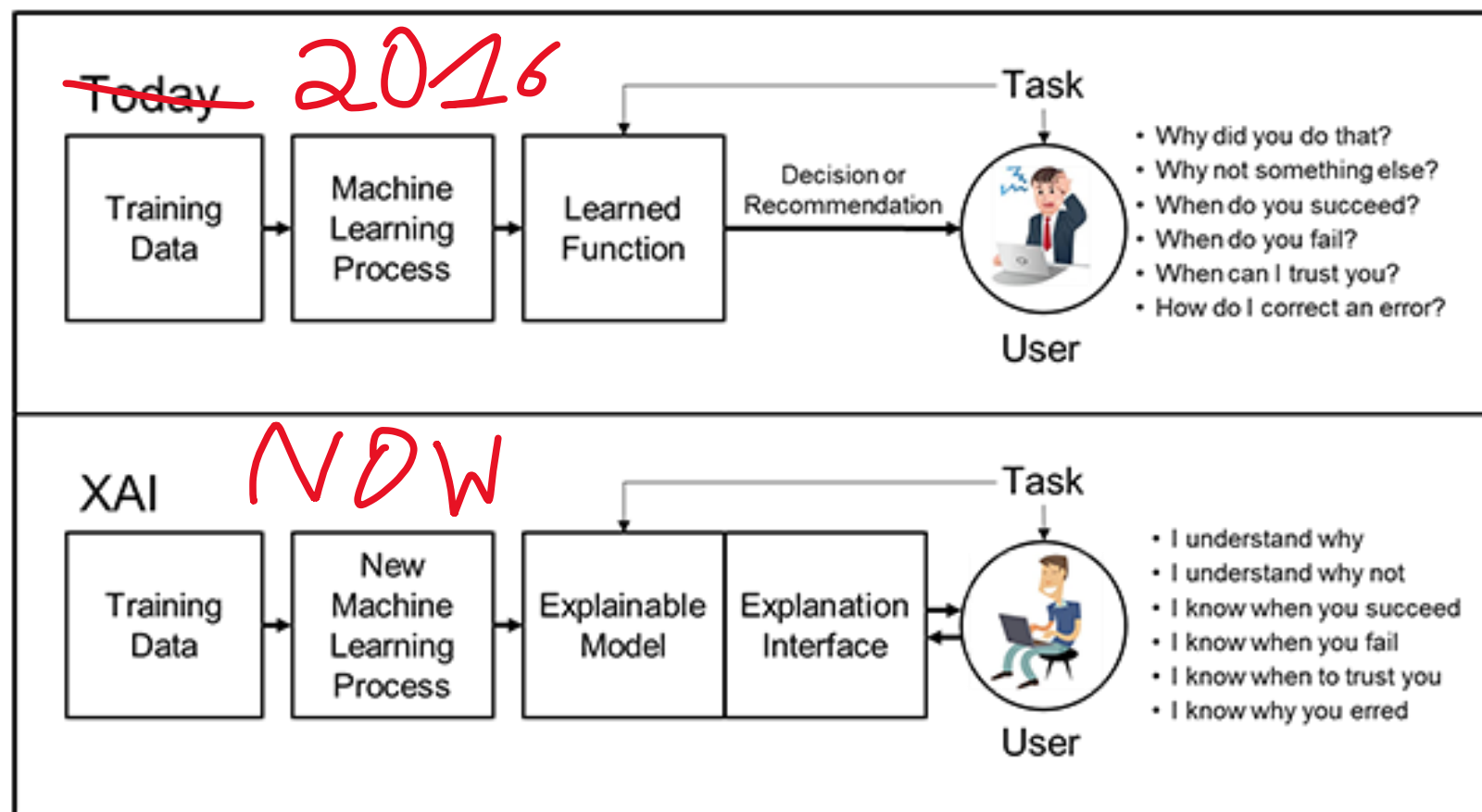
Google:

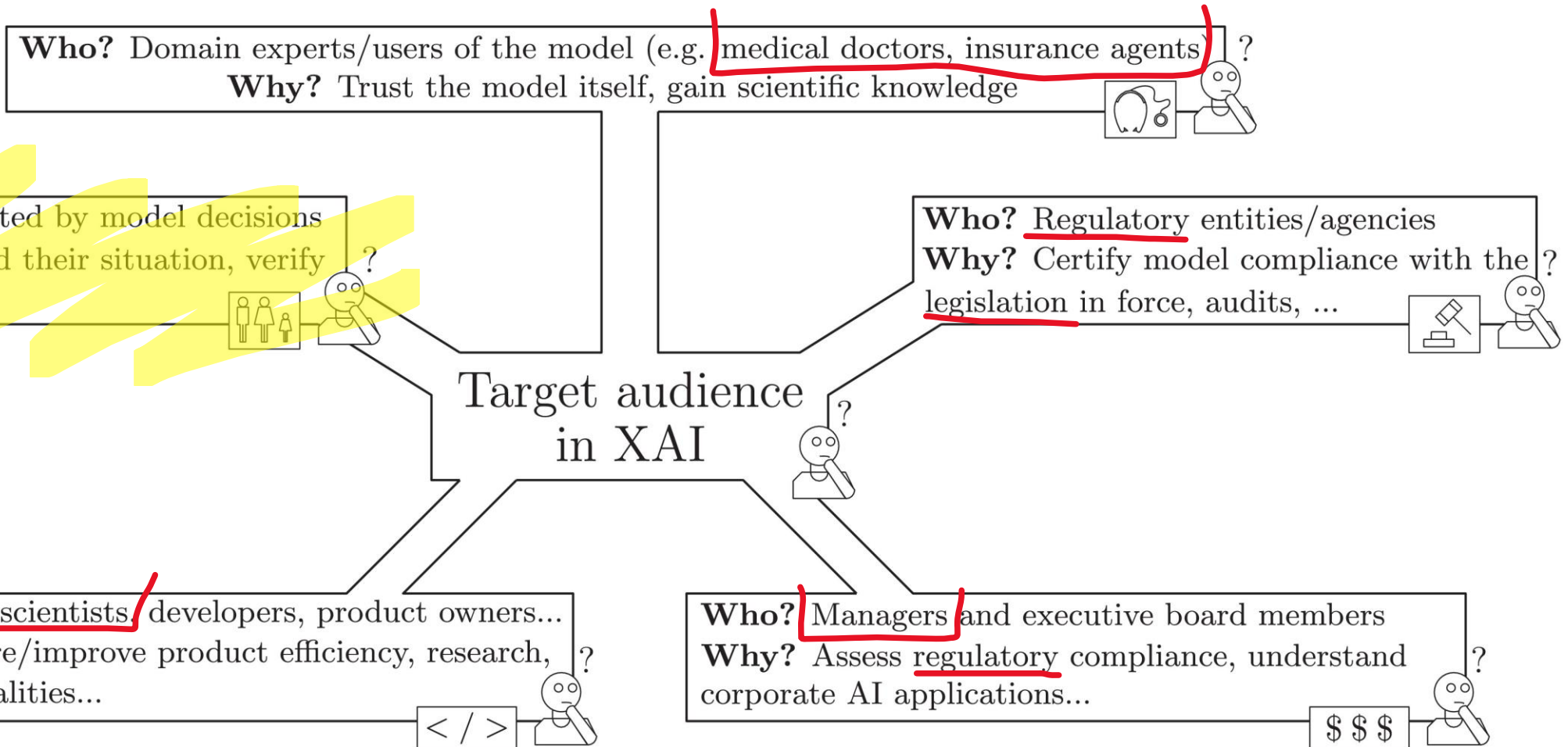
Tools and frameworks to understand and interpret your machine learning models.

Invest in XAI research 2017-21

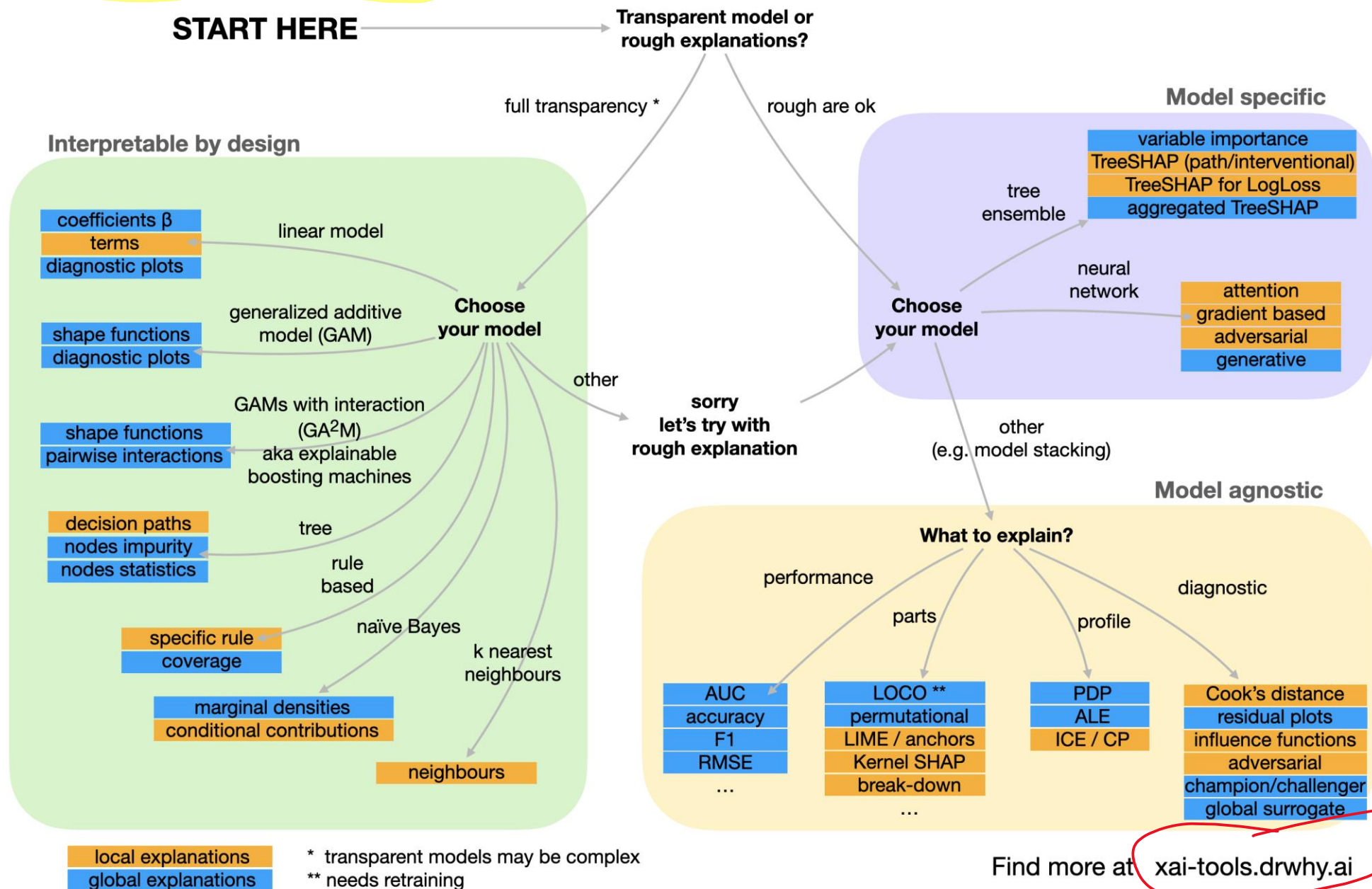
DARPA (Defense
Advanced Research
Projects Agency)

Government agency

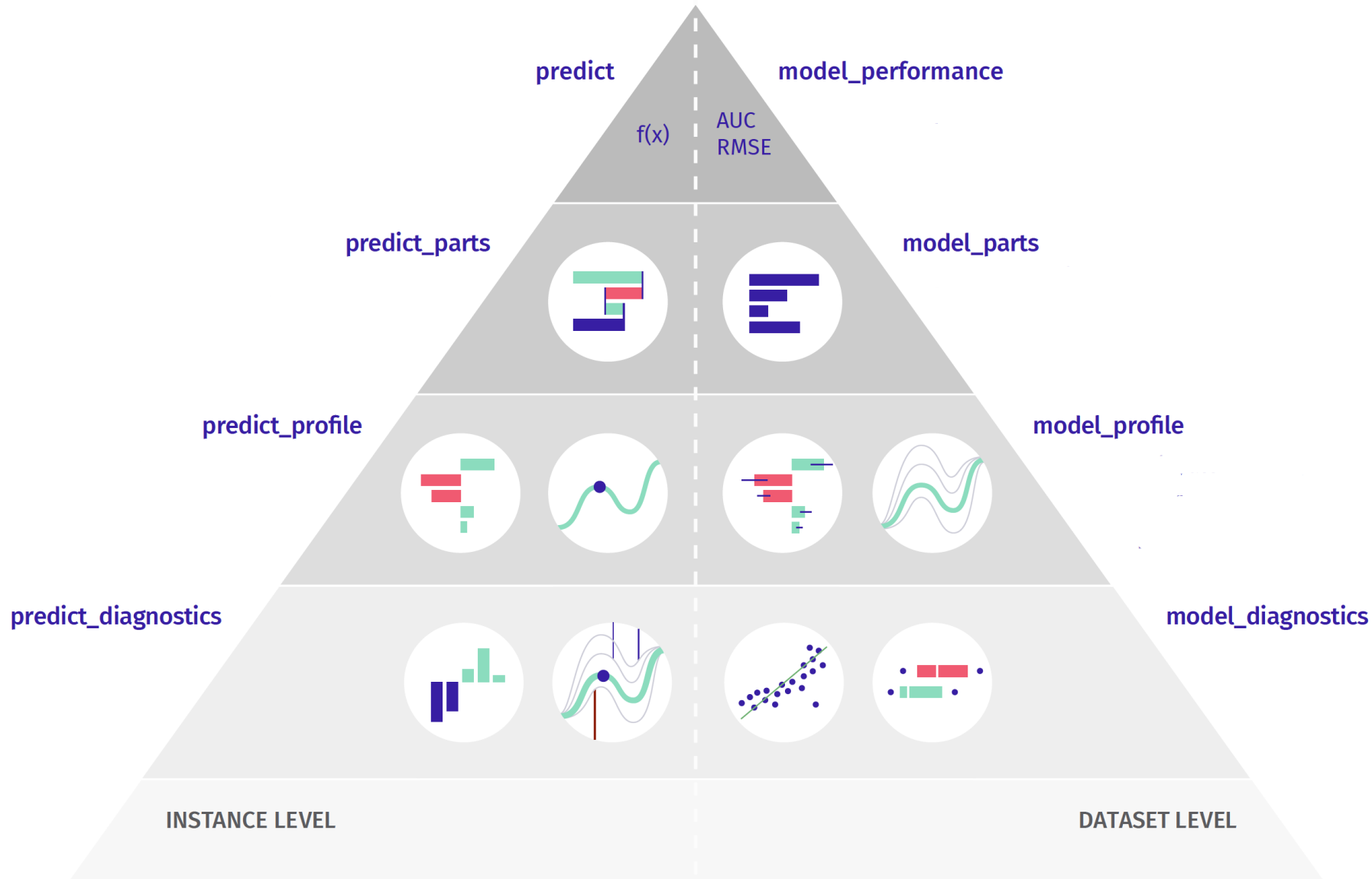




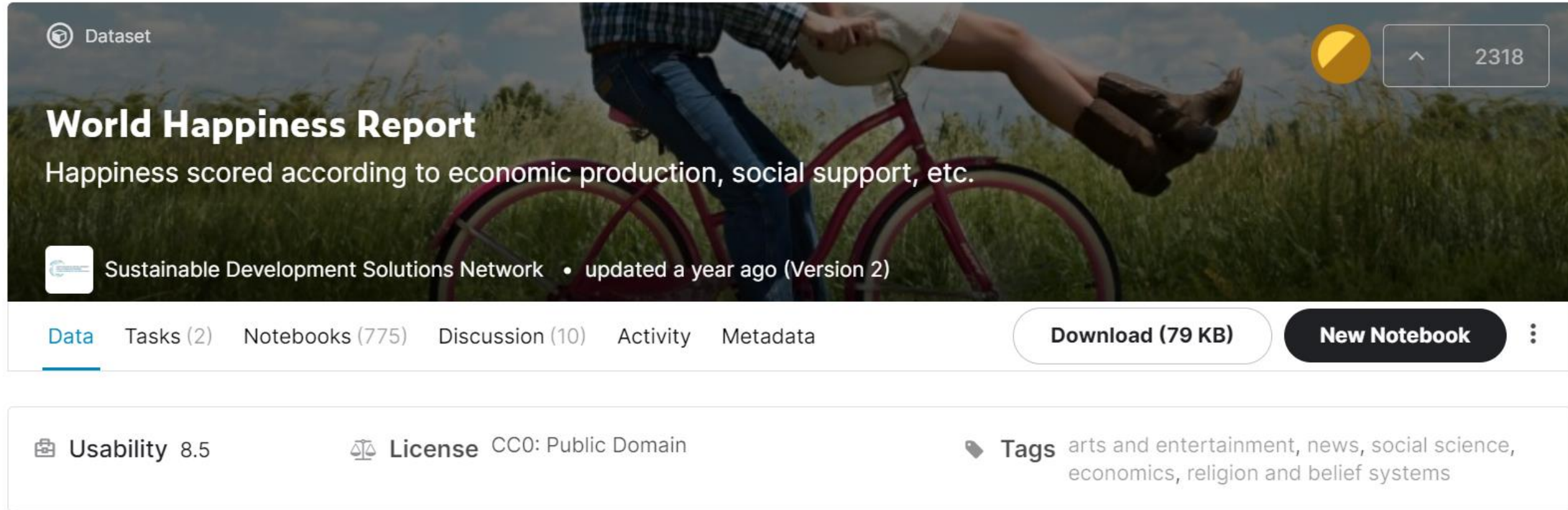
Arrieta, A. B. et al. (2020). Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Information Fusion*.



DALEX: moDeL Agnostic Language for Exploration and eXplanation



Machine learning predictive task



The image shows a screenshot of the Kaggle dataset page for the 'World Happiness Report'. The background is a photograph of a person riding a red bicycle through a field of tall grass. The page layout includes a header with the dataset name, a description, and the creator's name. Below this is a navigation bar with tabs for Data, Tasks, Notebooks, Discussion, Activity, and Metadata. To the right of the navigation bar are buttons for 'Download (79 KB)' and 'New Notebook'. At the bottom, there is a section for 'Usability' (8.5), 'License' (CC0: Public Domain), and 'Tags' (arts and entertainment, news, social science, economics, religion and belief systems).

Dataset

World Happiness Report

Happiness scored according to economic production, social support, etc.

Sustainable Development Solutions Network • updated a year ago (Version 2)

Data Tasks (2) Notebooks (775) Discussion (10) Activity Metadata

Download (79 KB) New Notebook

Usability 8.5 License CC0: Public Domain

Tags arts and entertainment, news, social science, economics, religion and belief systems

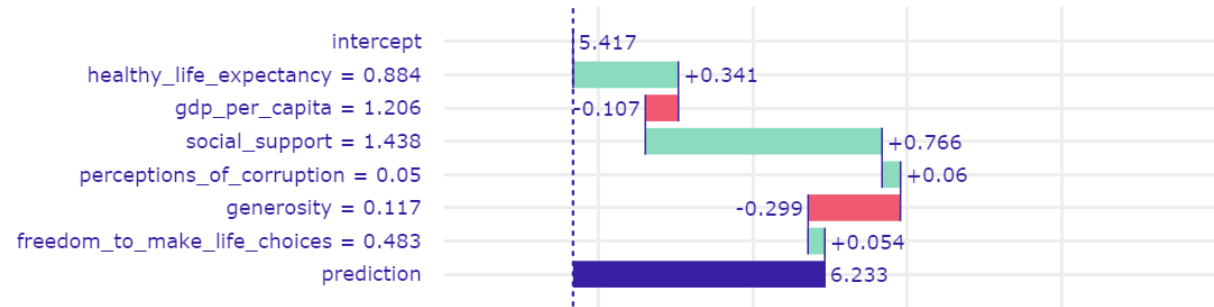
GDP, life expectancy, freedom, social => country happiness score [0, 10]

parts

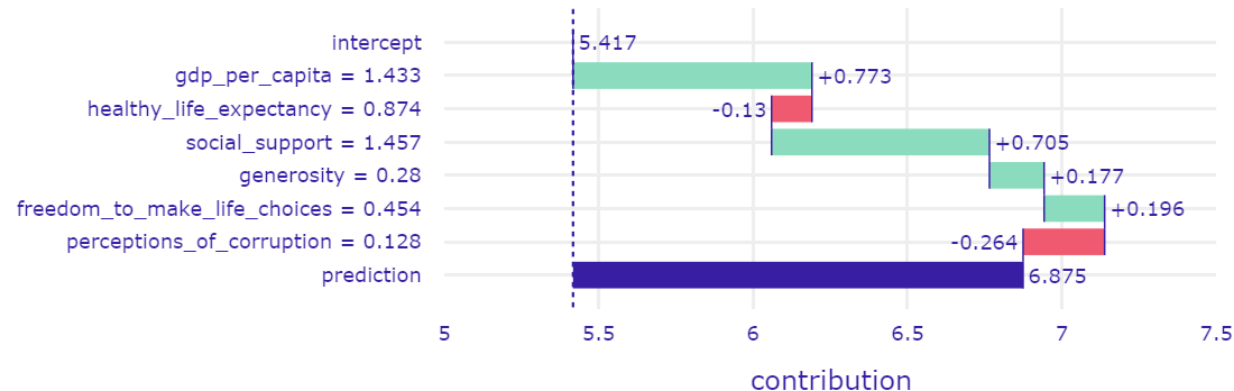
predict_parts

Break Down

Poland

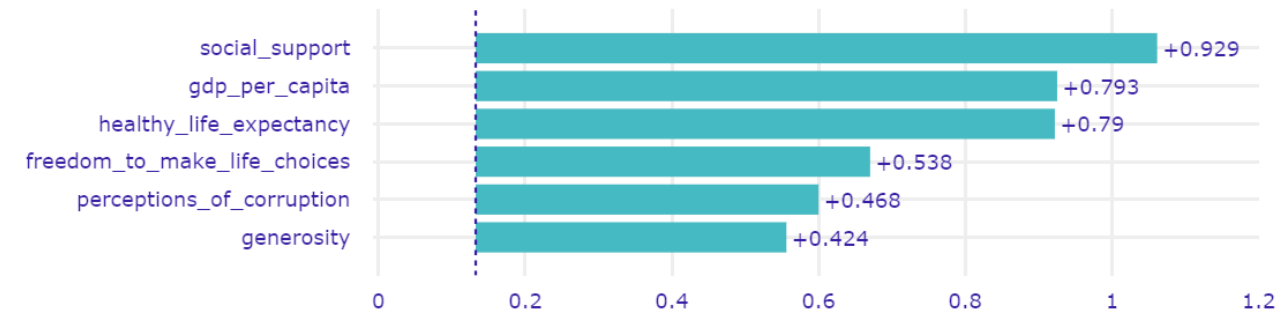


United States

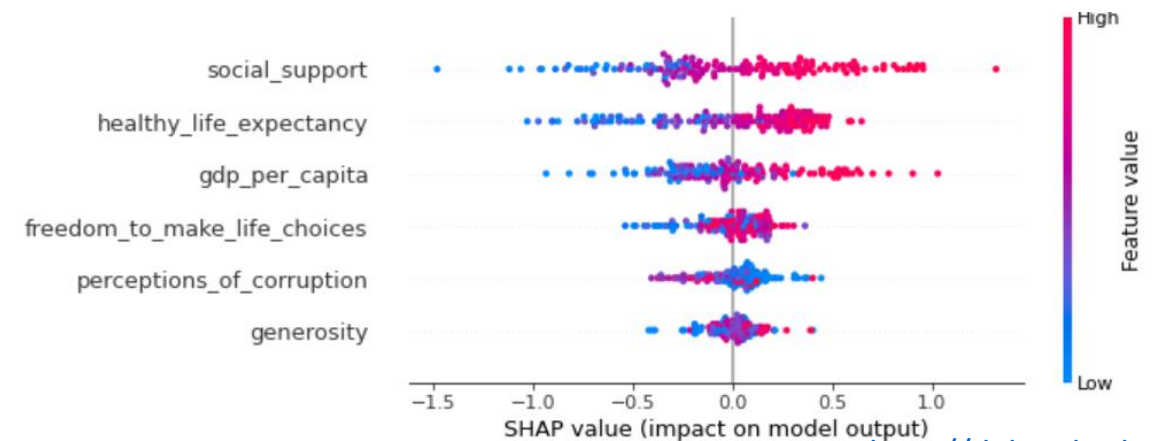


model_parts

Permutational Importance

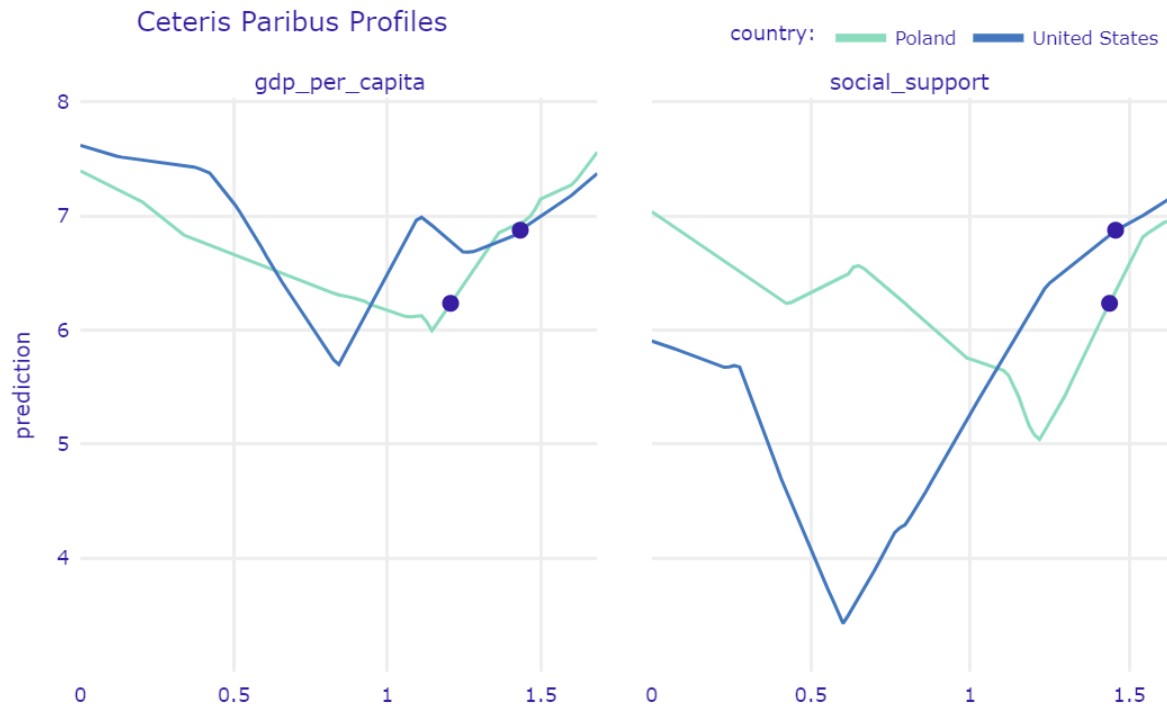


drop-out loss



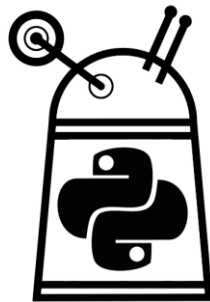
profile

predict_profile



model_profile





MODEL

- scikit-learn
- tensorflow, keras
- xgboost, lightgbm
- ANY

DATA

- pandas
- numpy

pip install dalex

import dalex as dx

dx.Explainer

EXPLANATIONS

- **result** attribute (pandas)
- **plot** method (plotly)

METHODS

predict/model + parts/profile/diagnostics
/surrogate/performance

Explainer



```
# 0. package
import dalex as dx
```

```
# 1. data
X, y = ...
```

```
# 2. model
model = ...
model.fit(X, y)
```

```
# 3. explainer
explainer = dx.Explainer(model, X, y)
```

Preparation of a new explainer is initiated

```
-> data           : 156 rows 6 cols
-> target variable : Argument 'y' was a pandas.Series. Converted to a numpy.ndarray.
-> target variable : 156 values
-> model_class     : tensorflow.python.keras.engine.sequential.Sequential (default)
-> label          : custom label
-> predict function : <function yhat_tf_regression at 0x000001D7649554C0> will be used
-> predict function : accepts pandas.DataFrame and numpy.ndarray
-> predicted values : min = 2.86, mean = 5.42, max = 7.73
-> model type      : regression will be used (default)
-> residual function : difference between y and yhat (default)
-> residuals       : min = -0.616, mean = -0.0103, max = 0.555
-> model_info      : package tensorflow
```

A new explainer has been created!

model



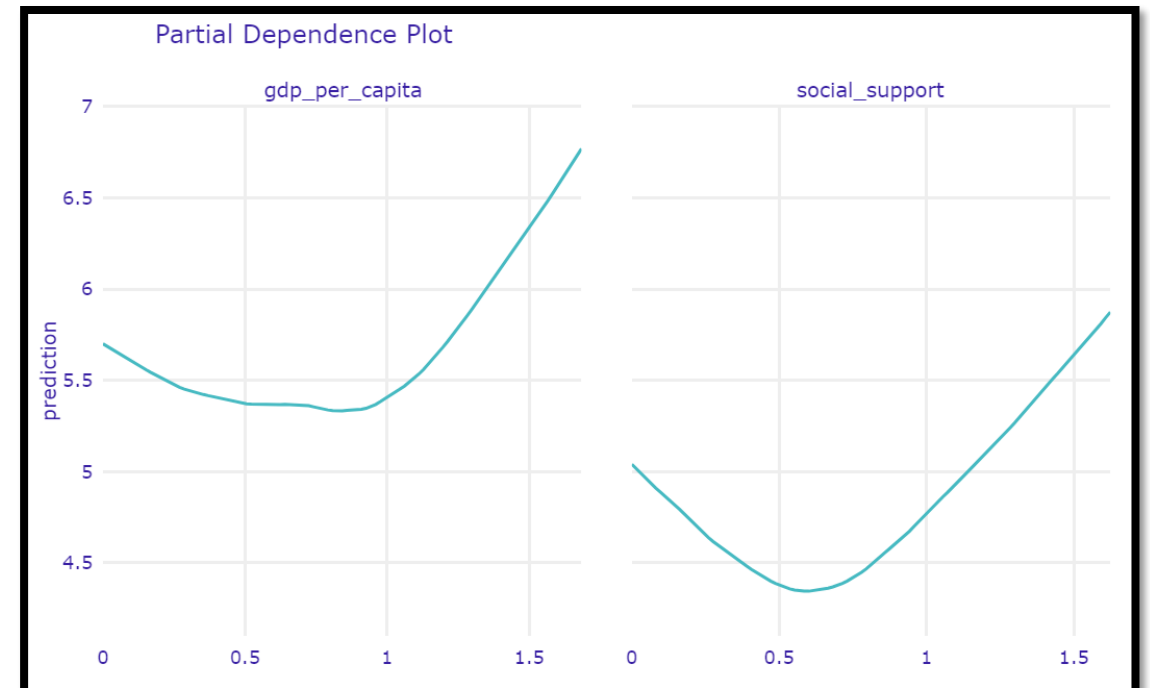
```
# 4. examine
explainer.model_performance()

# 5. explain
explainer.model_parts().result

# 6. explore
explainer.model_profile().plot()
```

mse	rmse	r2	mae	mad
0.017569	0.132549	0.985729	0.072329	0.03636

	variable	dropout_loss	label
0	_full_model_	0.132549	custom label
1	generosity	0.567029	custom label
2	perceptions_of_corruption	0.572801	custom label
3	freedom_to_make_life_choices	0.665235	custom label
4	gdp_per_capita	0.888245	custom label
5	healthy_life_expectancy	0.917414	custom label
6	social_support	1.046778	custom label
7	_baseline_	1.557307	custom label



predict



7. observation

```
obs = ...  
explainer.predict(obs)
```

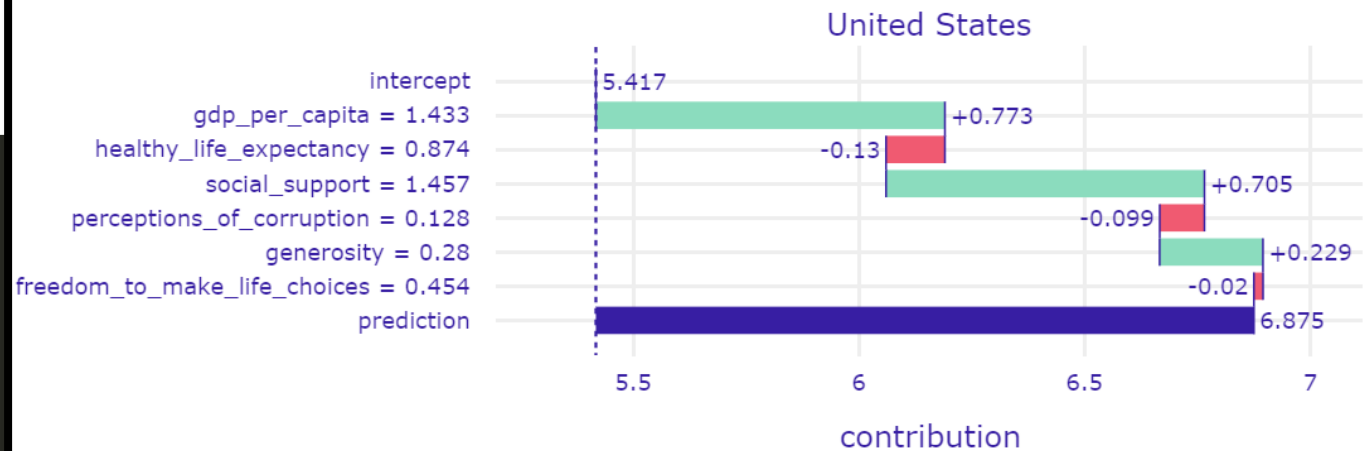
8. why?

```
explanation = explainer.predict_parts(obs)  
explanation.result  
explanation.plot()
```

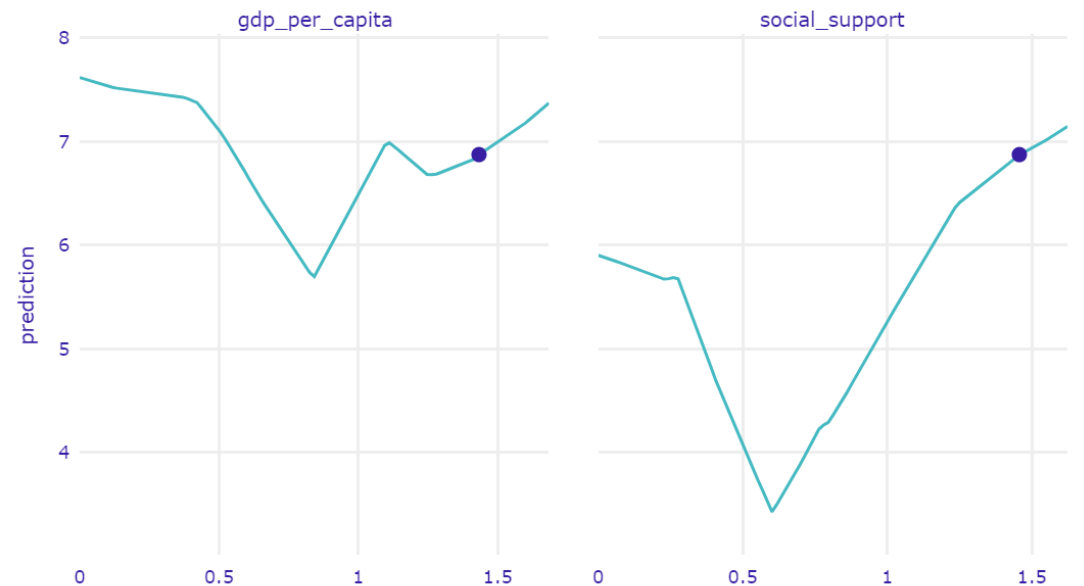
9. what if?

```
explainer.predict_profile(obs).plot()
```

Break Down



Ceteris Paribus Profiles



	variable_name	variable_value	variable	cumulative	contribution	sign	position	label
0	Intercept	1	Intercept	5.417360	5.417360	1.0	7	custom label
1	gdp_per_capita	1.433	gdp_per_capita = 1.433	6.189979	0.772619	1.0	6	custom label
2	healthy_life_expectancy	0.874	healthy_life_expectancy = 0.874	6.059744	-0.130235	-1.0	5	custom label
3	social_support	1.457	social_support = 1.457	6.764811	0.705067	1.0	4	custom label
4	perceptions_of_corruption	0.128	perceptions_of_corruption = 0.128	6.666029	-0.098782	-1.0	3	custom label
5	generosity	0.28	generosity = 0.28	6.894894	0.228865	1.0	2	custom label
6	freedom_to_make_life_choices	0.454	freedom_to_make_life_choices = 0.454	6.874513	-0.020381	-1.0	1	custom label
7			prediction	6.874512	6.874512	1.0	0	custom label

more!



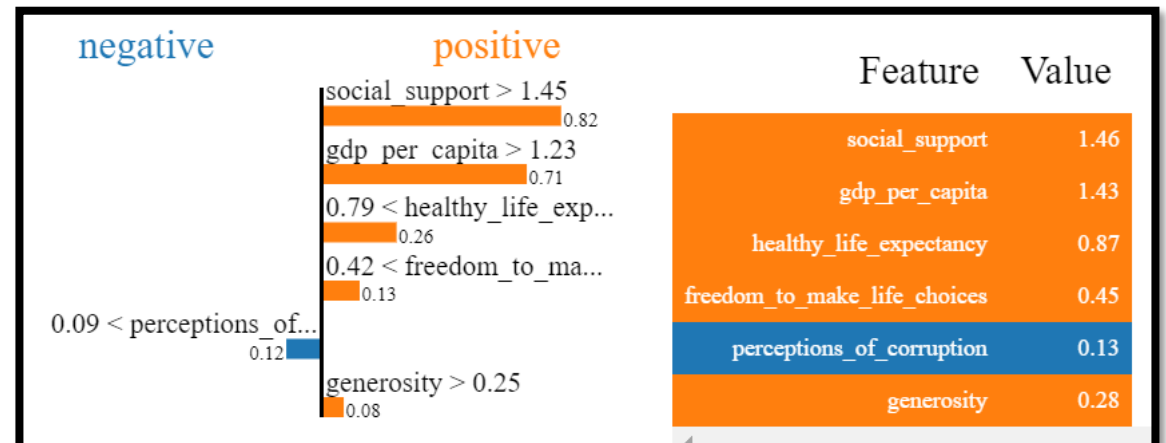
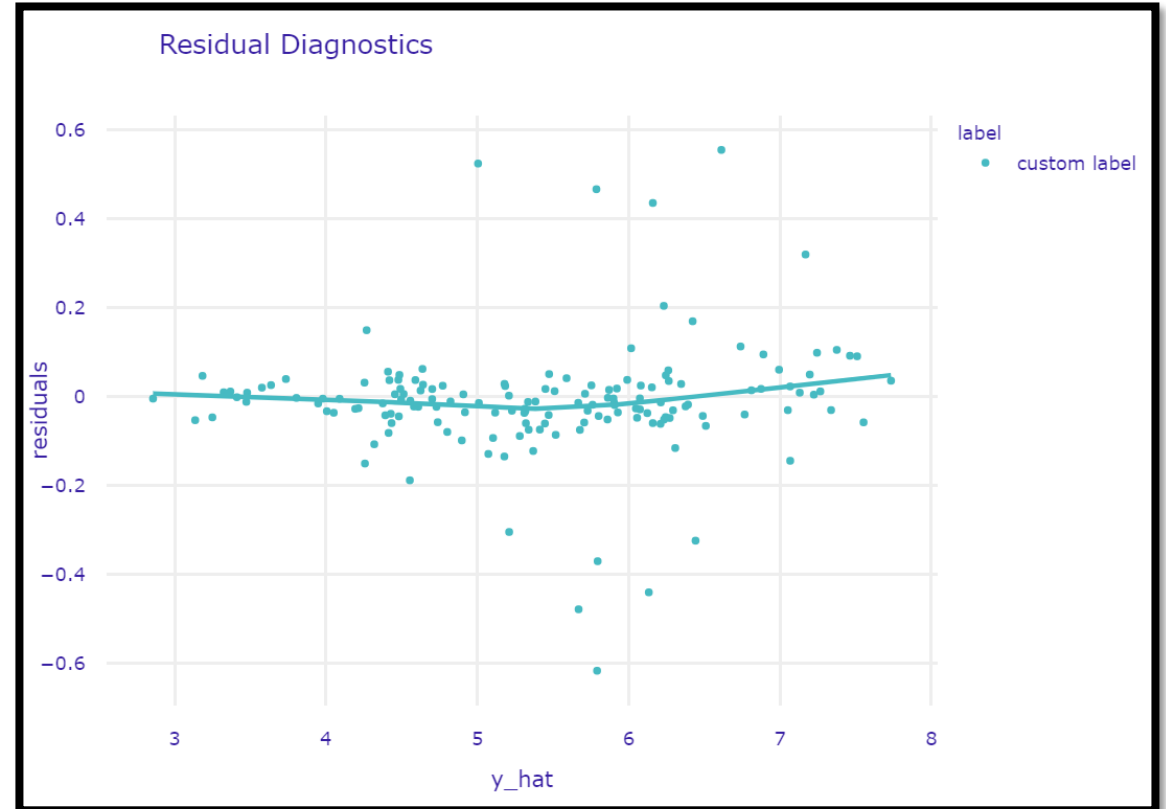
```
# 10. residuals
explainer.model_diagnostics().plot()

# 11. surrogate
tree = explainer.model_surrogate()
tree.plot()

# 13. types
explainer.model_profile(type='accumulated')

# 14. shap
explainer.model_parts(type='shap_wrapper')

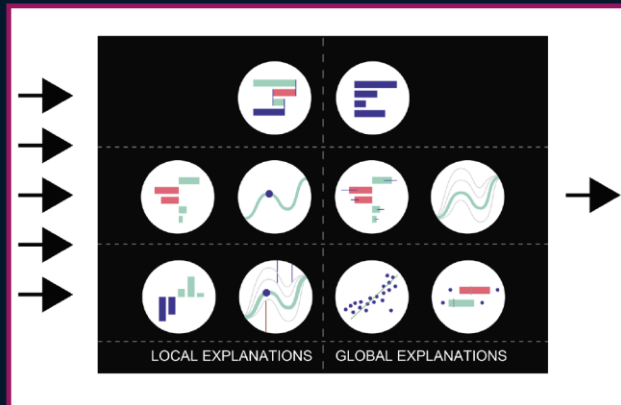
# 15. lime
explainer.predict_surrogate(obs)
```



DATA SCIENCE SERIES

EXPLANATORY MODEL ANALYSIS

Explore, Explain, and
Examine Predictive Models



PRZEMYSŁAW BIECEK
TOMASZ BURZYKOWSKI

 **CRC Press**
Taylor & Francis Group
A CHAPMAN & HALL BOOK

<https://pbiecek.github.io/ema/>

Model Exploration Stack

What is the model prediction
for the selected instance?

$f(x)$ AUC
RMSE

How good is the model?

ROC curve
LIFT, Gain charts
Chapter 15

Which variables contribute to
the selected prediction?

Break Down
SHAP, LIME
Chapters 6, 7, 8, 9

Which variables are important
to the model?

Permutational
Variable Importance
Chapter 16

How does a variable
affect the prediction?

Ceteris Paribus
Chapters 10, 11

How does a variable affect
the average prediction?

Partial Dependence Profile
Accumulated Local Effects
Chapters 17, 18

Does the model
fit well around
the prediction?

Chapter 12

Does the model
fit well in
general?

Chapter 19

PREDICTION LEVEL
LOCAL EXPLANATIONS

MODEL LEVEL
GLOBAL EXPLANATIONS

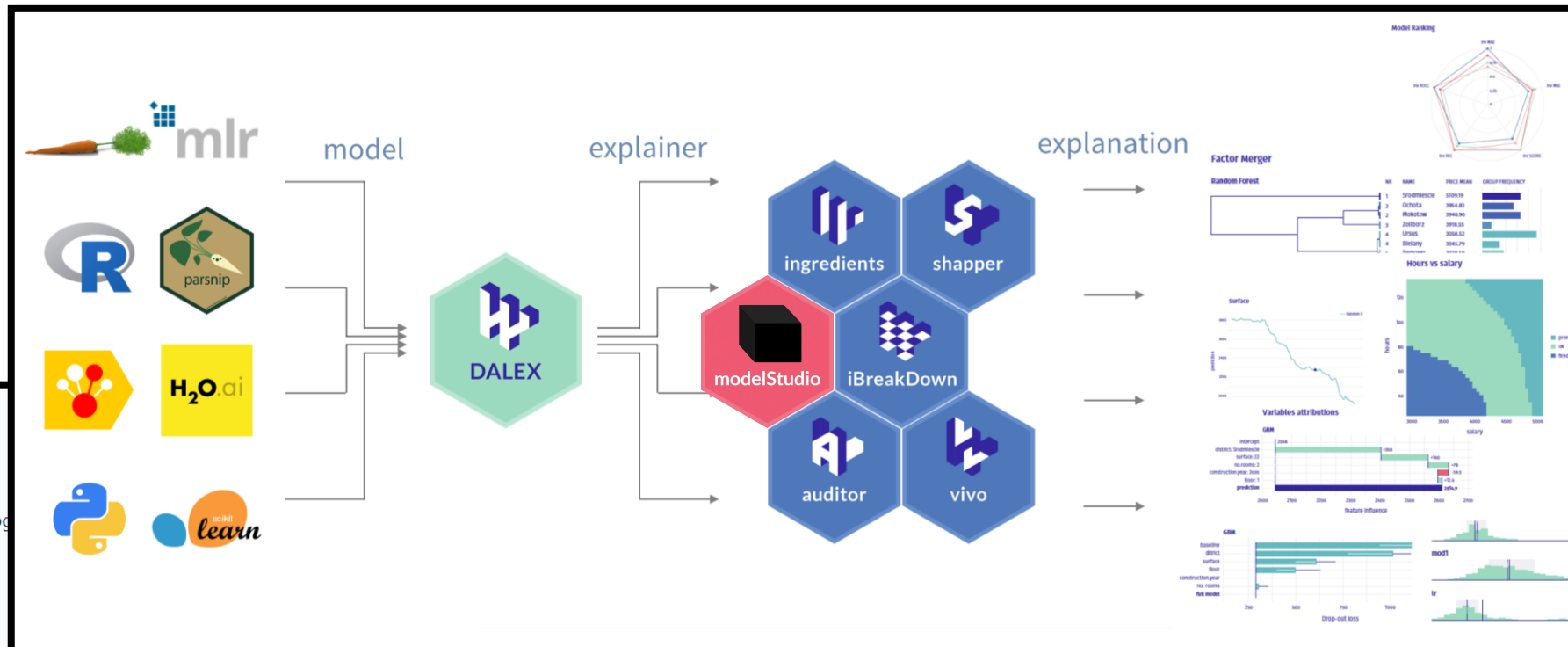
DrWhy.AI



Model Oriented

MI2DataLab @ Warsaw University of Technology

Repositories 46 Packages People 21



DALEX

model Agnostic Language for Exploration and eXplanation

Python 663 100

DrWhy

DrWhy is the collection of tools for eXplainable AI (XAI). It's based on shared principles and simple grammar for exploration, explanation and visualisation of predictive models.

R 398 51

randomForestExplainer

A set of tools to understand what is happening inside a Random Forest

R 166 25

modelStudio

Interactive Studio for Explanatory Model Analysis

R 138 17

modelDown

modelDown generates a website with HTML summaries for predictive models

R 101 12

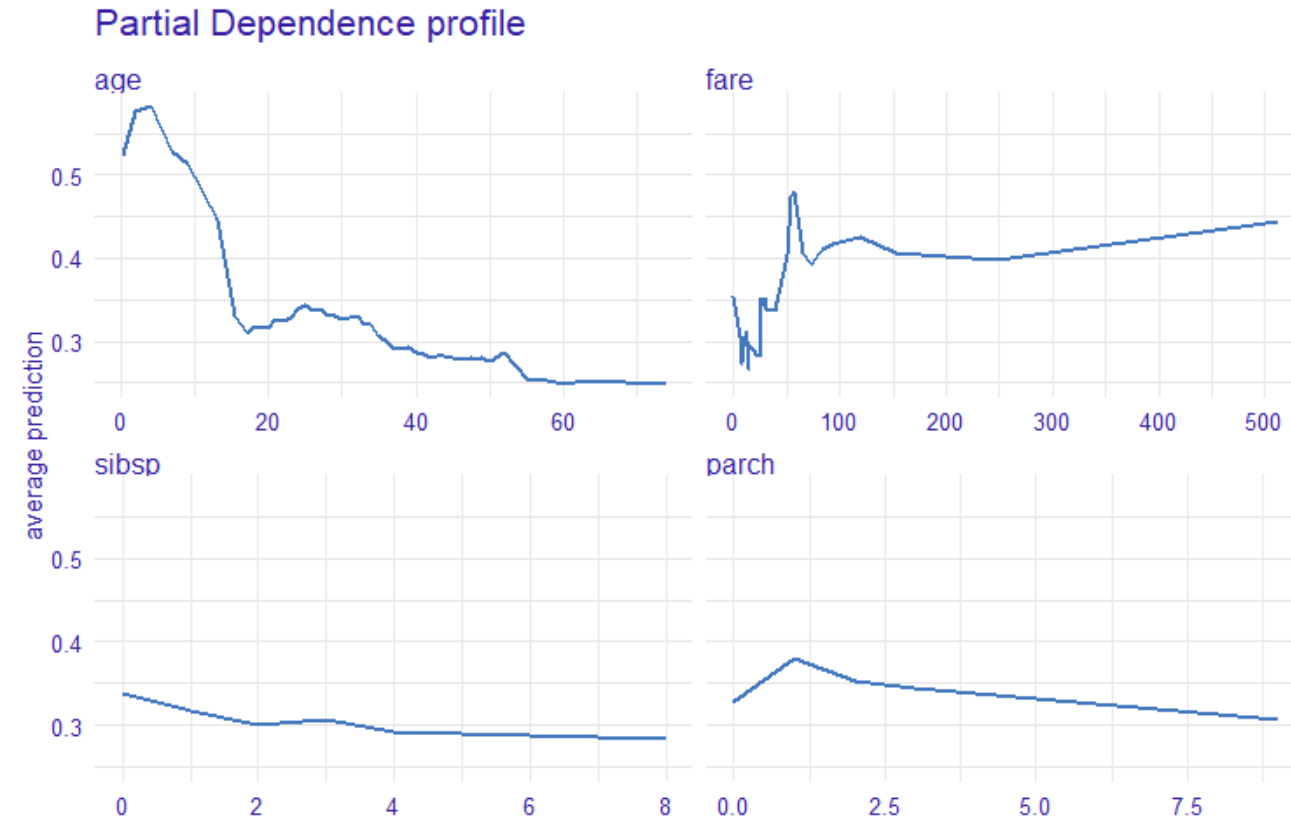
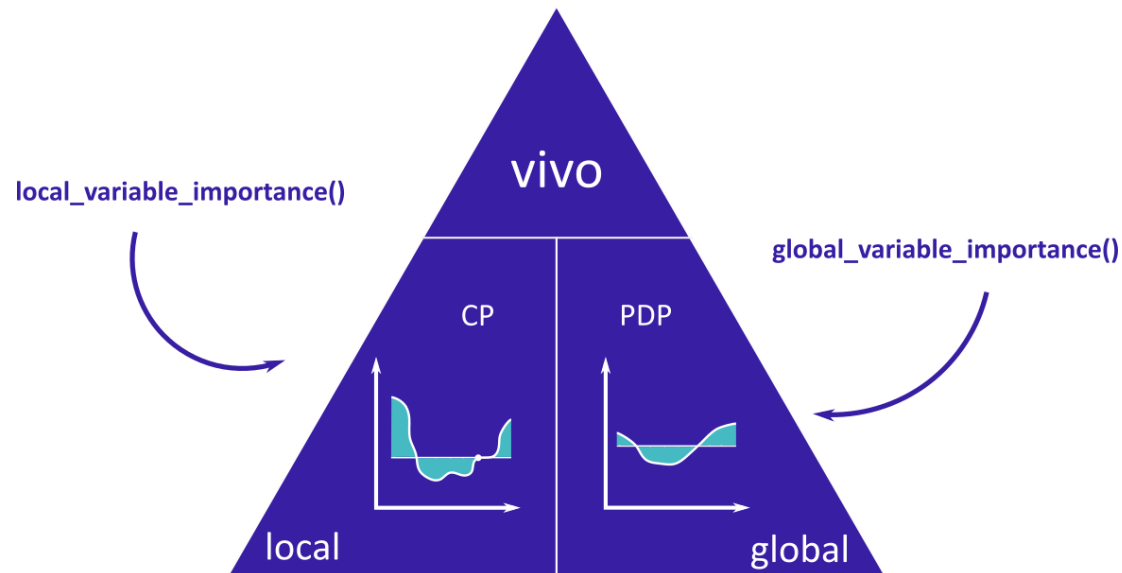
iBreakDown

Break Down with interactions for local explanations (SHAP, BreakDown, iBreakDown)

R 54 9

vivo

- alternative, model-agnostic way of calculating variable importance
- based on the Ceteris Paribus and Partial Dependence Profiles
- faster, no random component



Interactive XAI

How to explain?

parts

profile

distribution

*prediction
(local)*

**Break Down
Shapley Values**

1.

Ceteris Paribus

2.

**Feature
Importance**

4.

**Partial
Dependence**

6.

**Residuals
Distribution**

9.

**Pairwise
Correlations**

10.

**Scatter
Plots**

8.

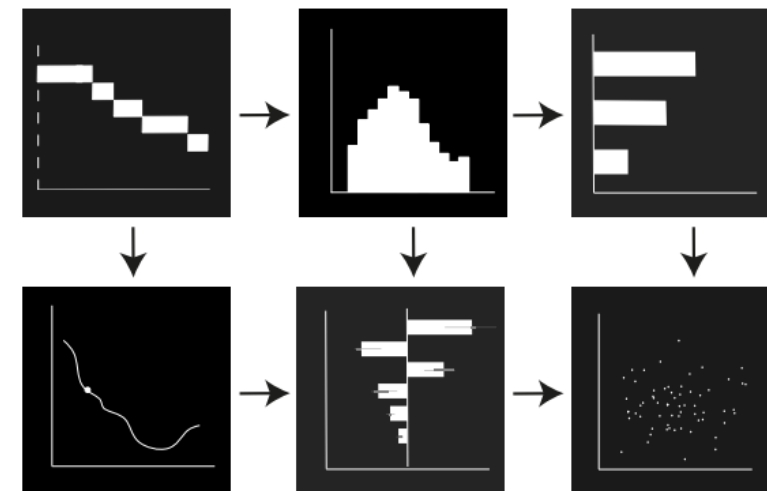
**Feature
Distributions**

What to explain?

*model
(global)*

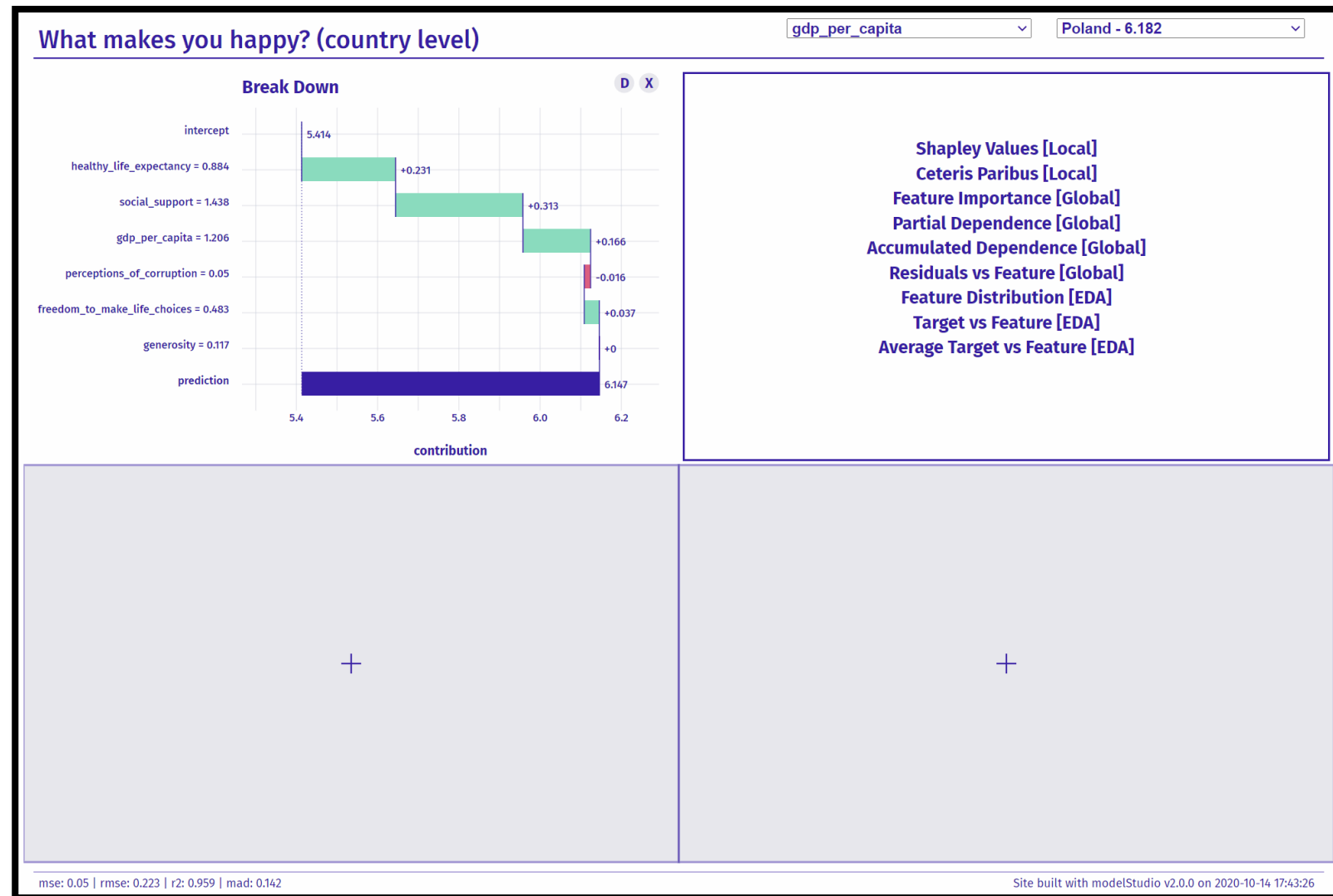
data

II generation explanations
(interactive explanatory
model analysis)



modelStudio

- creates a dashboard for interactive Explainable AI
- model explanation and data exploration
- automated calculations
- save & share your analysis



convenient

```
# 0. package
library("DALEX")
library("modelStudio")

# 1. data
X <- ...
y <- ...

# 2. model
model <- ...

# 3. explainer
explainer <- DALEX::explain(model, X, y)

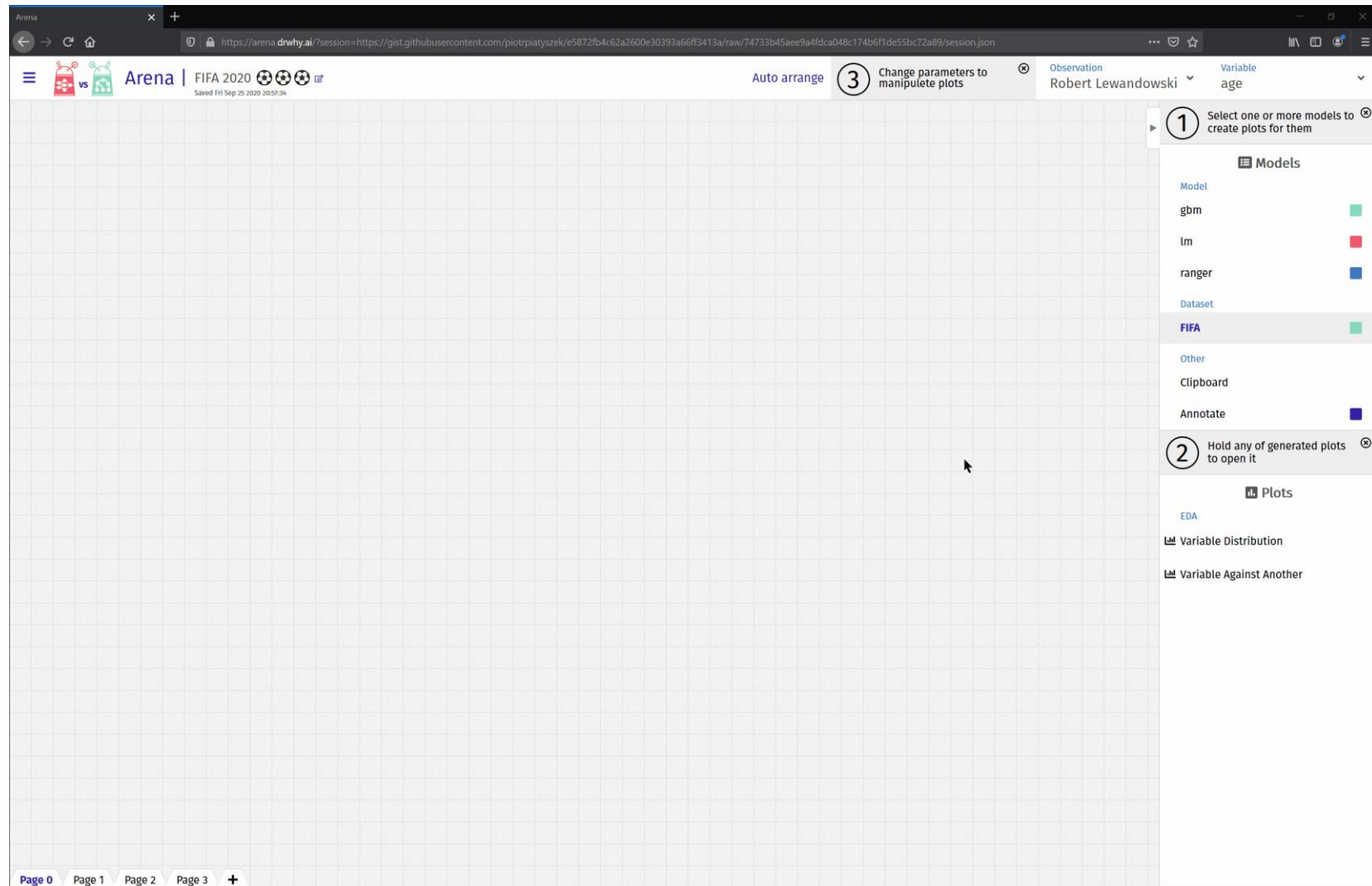
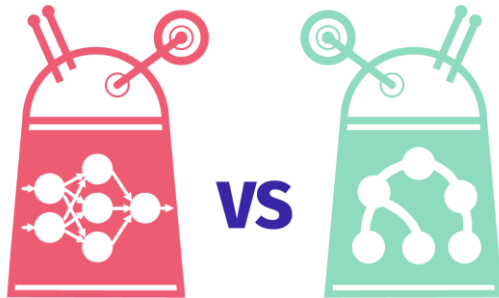
# 4. dashboard
ms <- modelStudio::modelStudio(explainer)
ms
```

DEMO:

<https://pbiecek.github.io/xai-happiness>

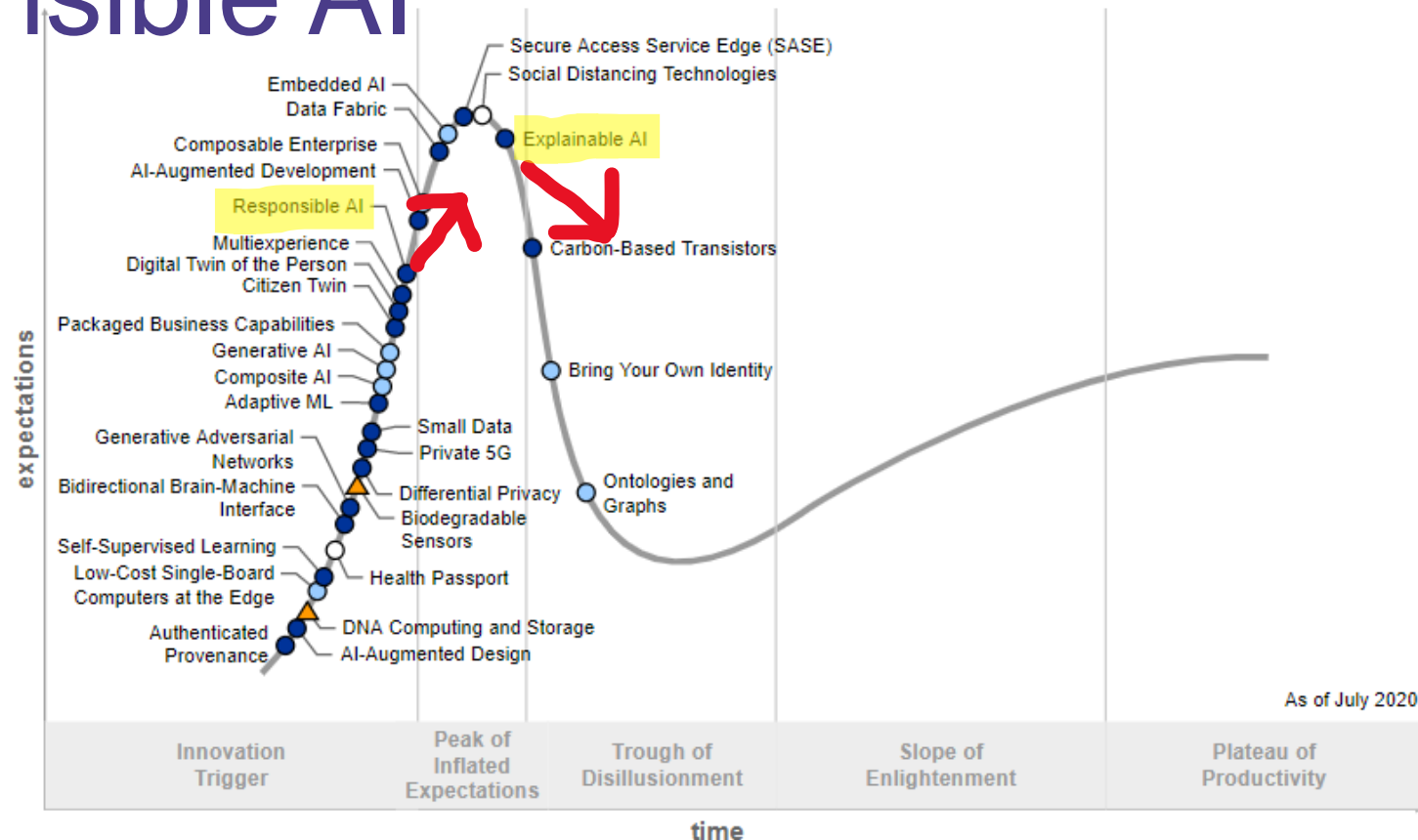
Arena

- multiple models!
- multiple datasets!!
- more plots (e.g. fairness)
- pages & cache & ...
- R & Python (next version)



Hype Cycle for Artificial Intelligence, 2020

Responsible AI (RAI)



Plateau will be reached:

○ less than 2 years ● 2 to 5 years ● 5 to 10 years ▲ more than 10 years ⊗ obsolete before plateau

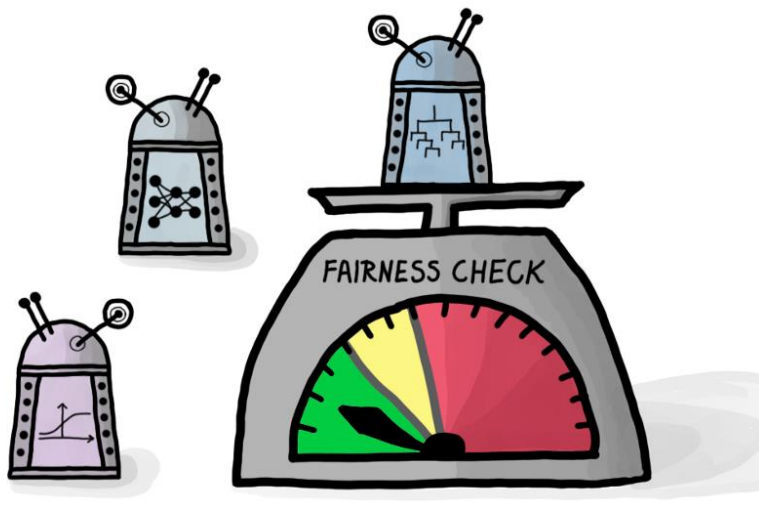
gartner.com/SmarterWithGartner

Source: Gartner
© 2020 Gartner, Inc. and/or its affiliates. All rights reserved. Gartner and Hype Cycle are registered trademarks of Gartner, Inc. and its affiliates in the U.S.

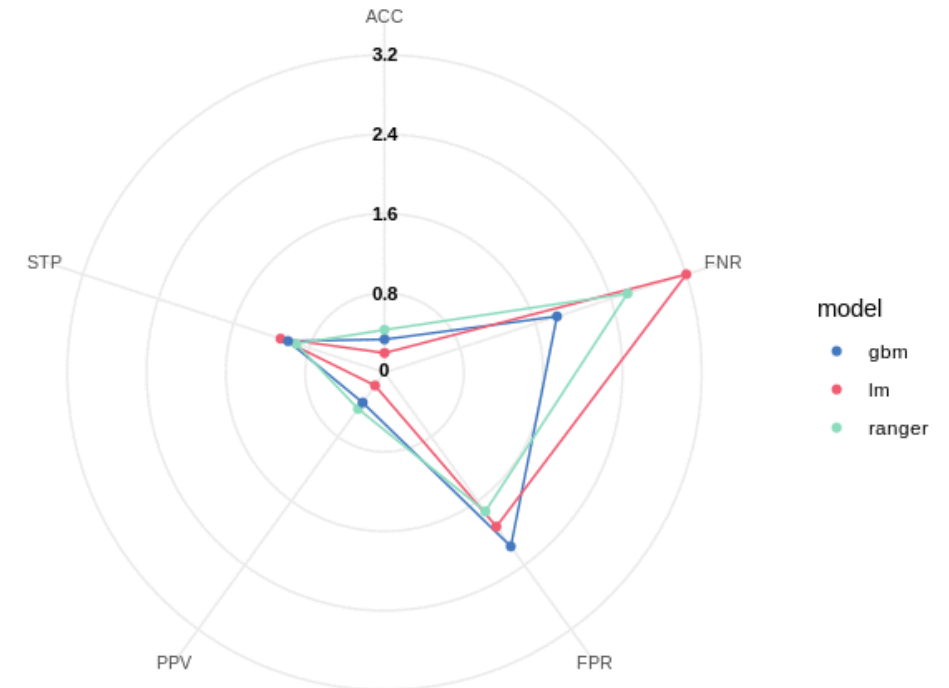
Gartner

fairmodels

- check model fairness in respect to sensitive categorical variables
- pre- and post- bias mitigation
- compare measures for multiple models
- various techniques and visualisations



Parity loss metric radar plot



fairness check



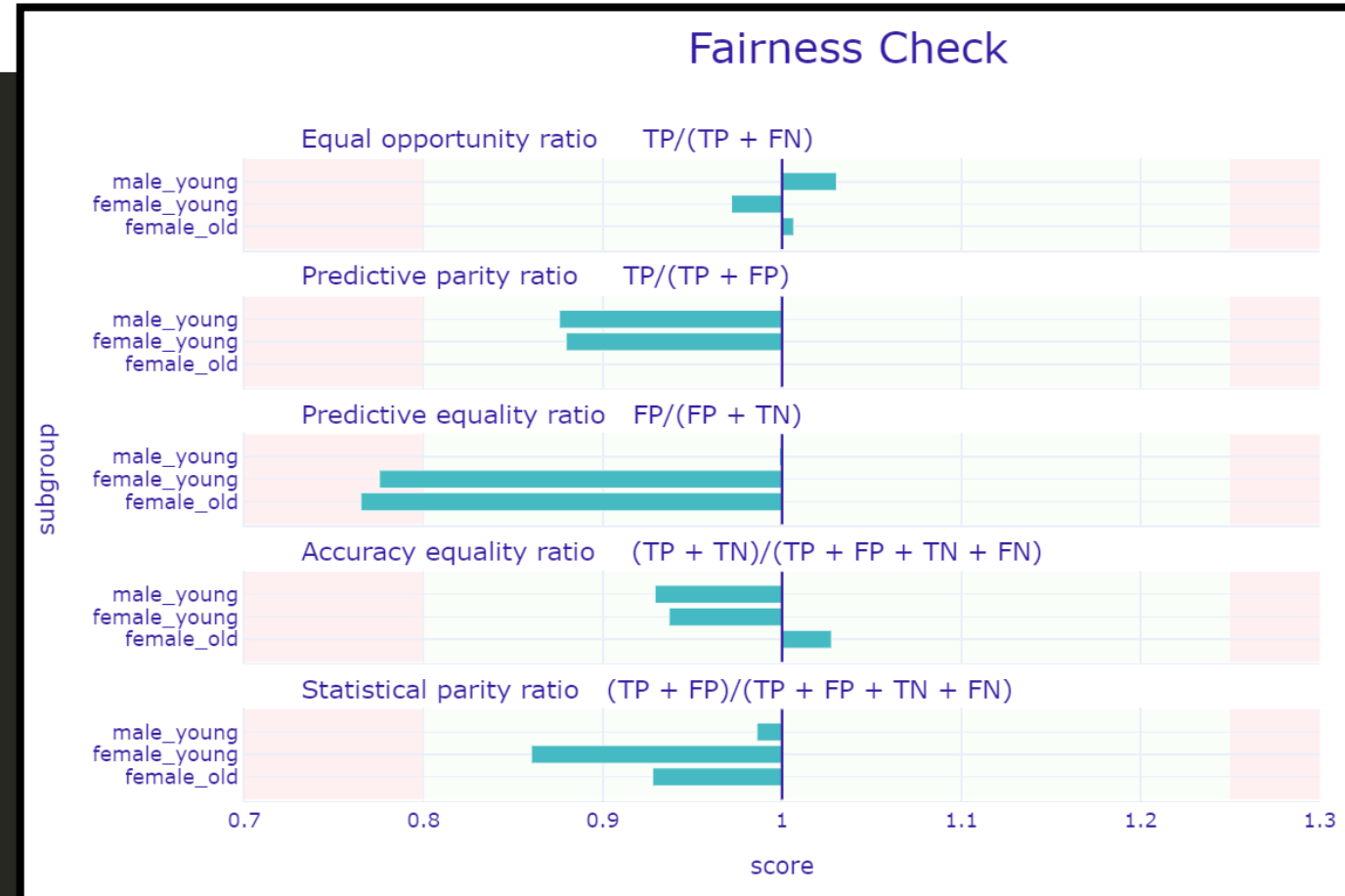
```
# 1. protected variable with subgroups  
protected = [race + sex + age for ...]
```

```
# 2. privileged subgroup  
privileged = 'white_male_young'
```

```
# 3. fairness  
explanation = explainer.model_fairness(  
    protected, privileged  
)
```

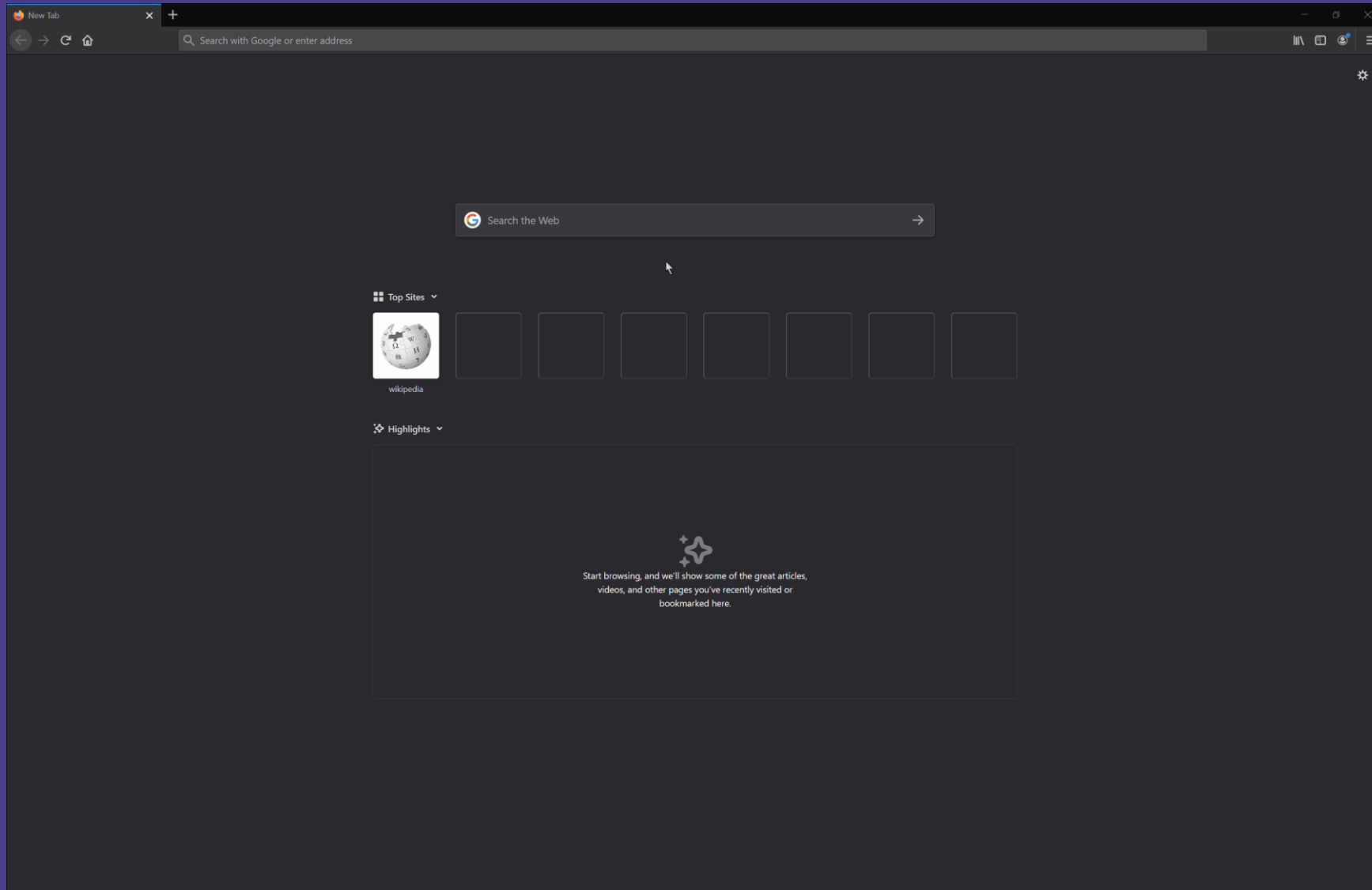
```
# 4. check  
explanation.fairness_check()
```

```
# 5. explain  
explanation.result  
explanation.plot()
```



DrWhy.AI blog: Responsible ML

<https://medium.com/responsibleml>



Feedback appreciated !

Contact me	linkedin.com/in/hbaniecki
DALEX	dalex.drwhy.ai
DrWhy.AI	drwhy.ai
Blog	medium.com/responsibleml