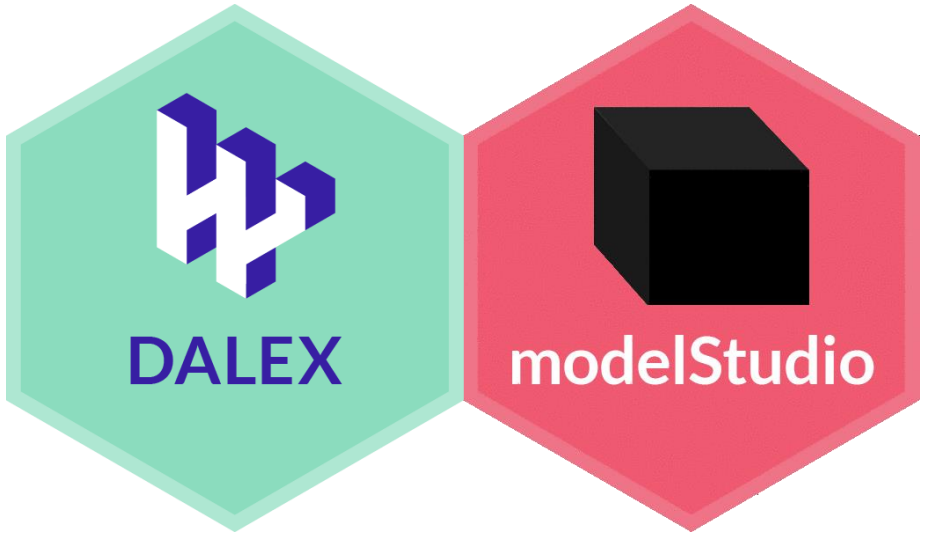
Explaining machine learning predictions with DALEX and beyond

Hubert Baniecki
MI² lab, Warsaw University of Technology, Poland

Ryanair Labs Madrid
January, 2022

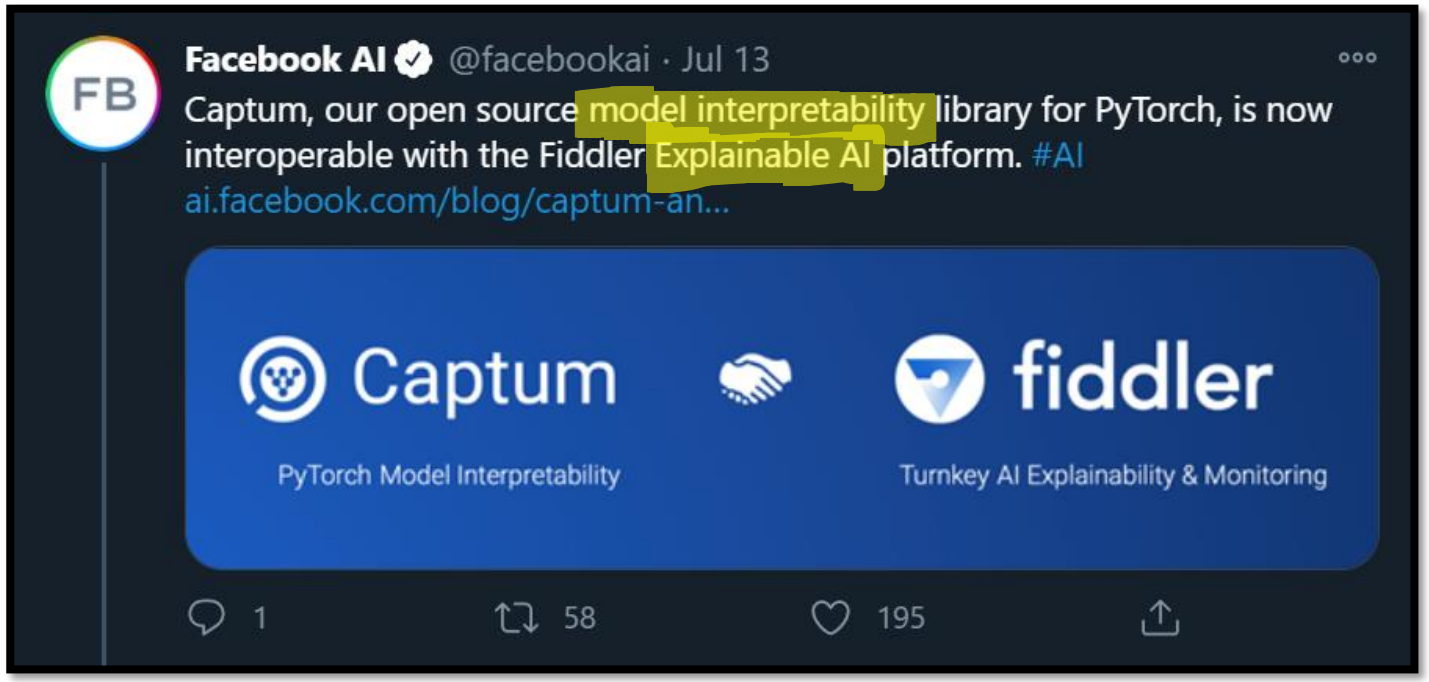
```
hbaniecki:~$ whoami
Researcher and data science student at Warsaw University of Technology
Developing tools for explainable machine learning in R & Python
```



https://hbaniecki.com

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*
-



Fit interpretable models. Explain blackbox machine learning.

machine-learning interpretability gradient-boosting blackbox scikit-learn xai interpretml interpretable-machine-learning machine-learning-interpretability interpretable-ai transparency iml interpretable-ml explainable-ml explainability bias ai artificial-intelligence explainable-ai

Explainable AI (XAI)

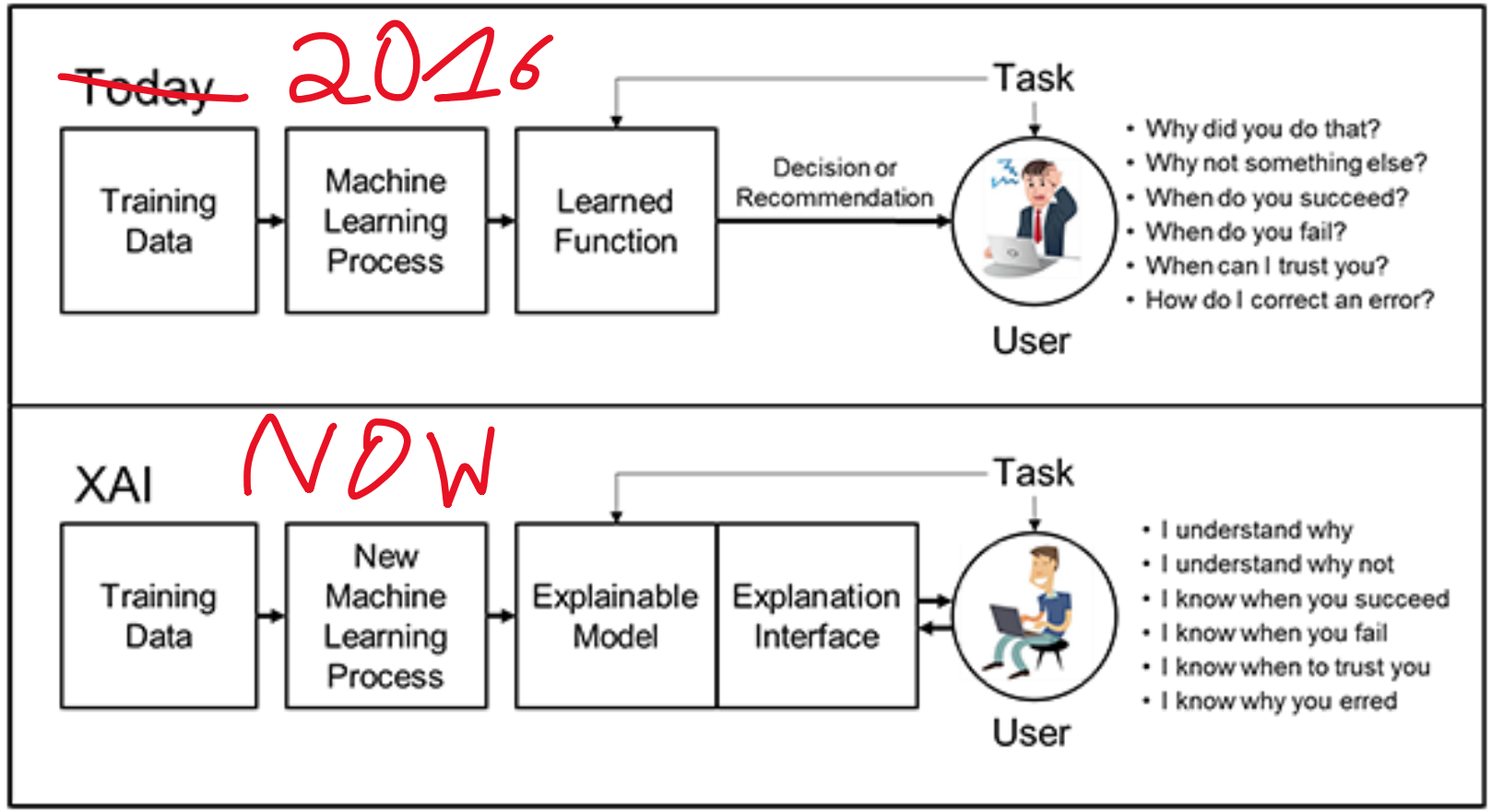
Google:
Tools and frameworks to understand and interpret your machine learning models.

https://cloud.google.com/explainable-ai

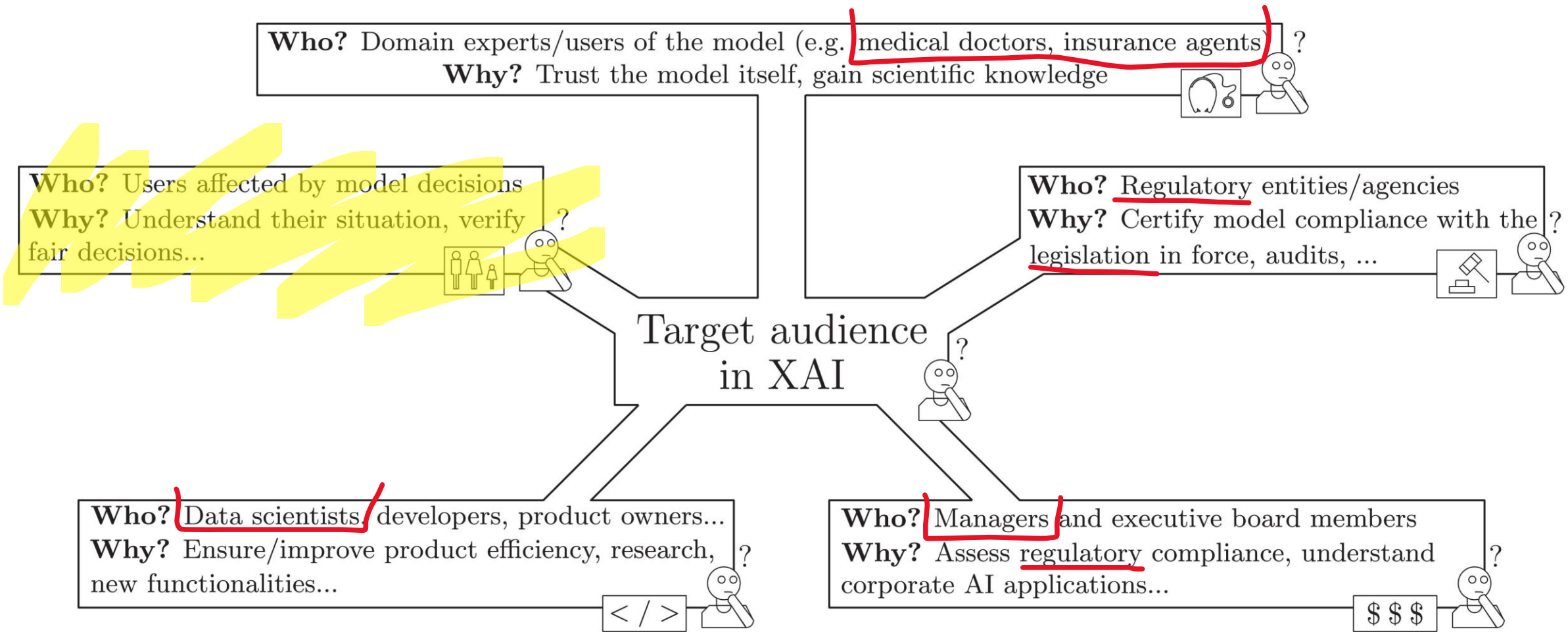
Invest in XAI research 2017-21

DARPA (Defense Advanced Research Projects Agency)

Government agency



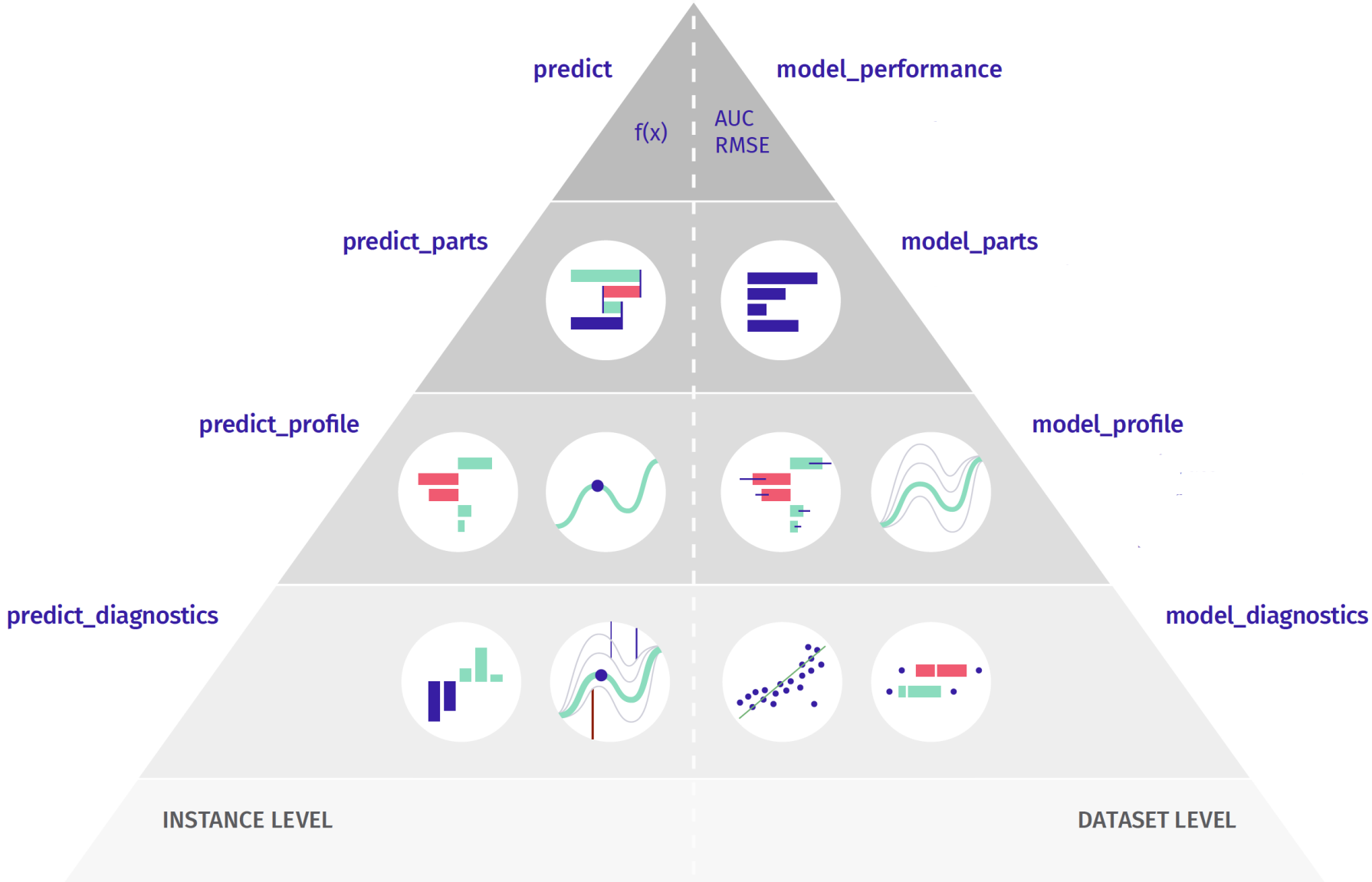
https://www.darpa.mil/program/explainable-artificial-intelligence



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

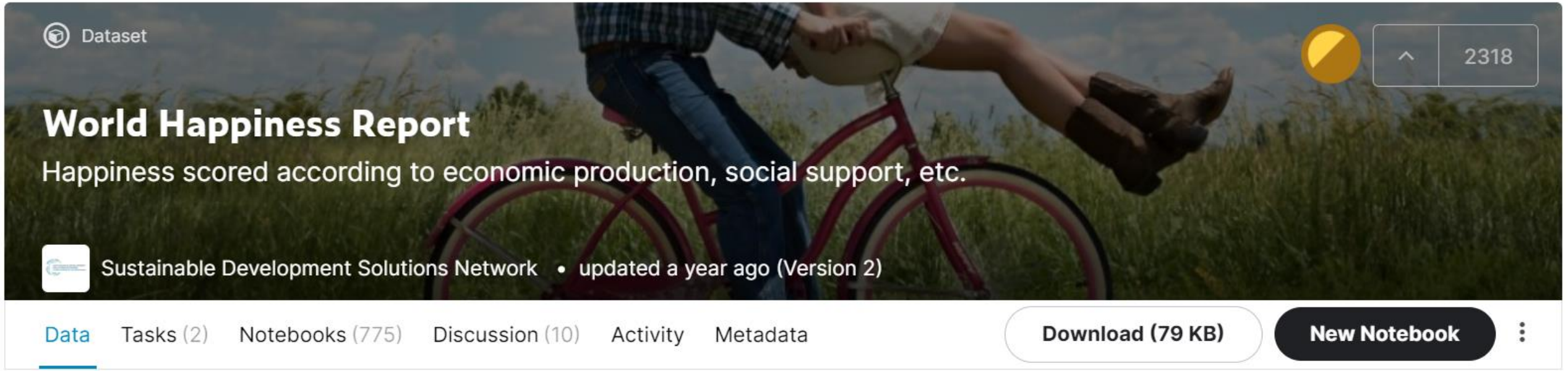
https://doi.org/10.1016/j.inffus.2019.12.012

DALEX: moDel Agnostic Language for Exploration and eXplanation



http://dalex.drwhy.ai

Machine learning predictive task

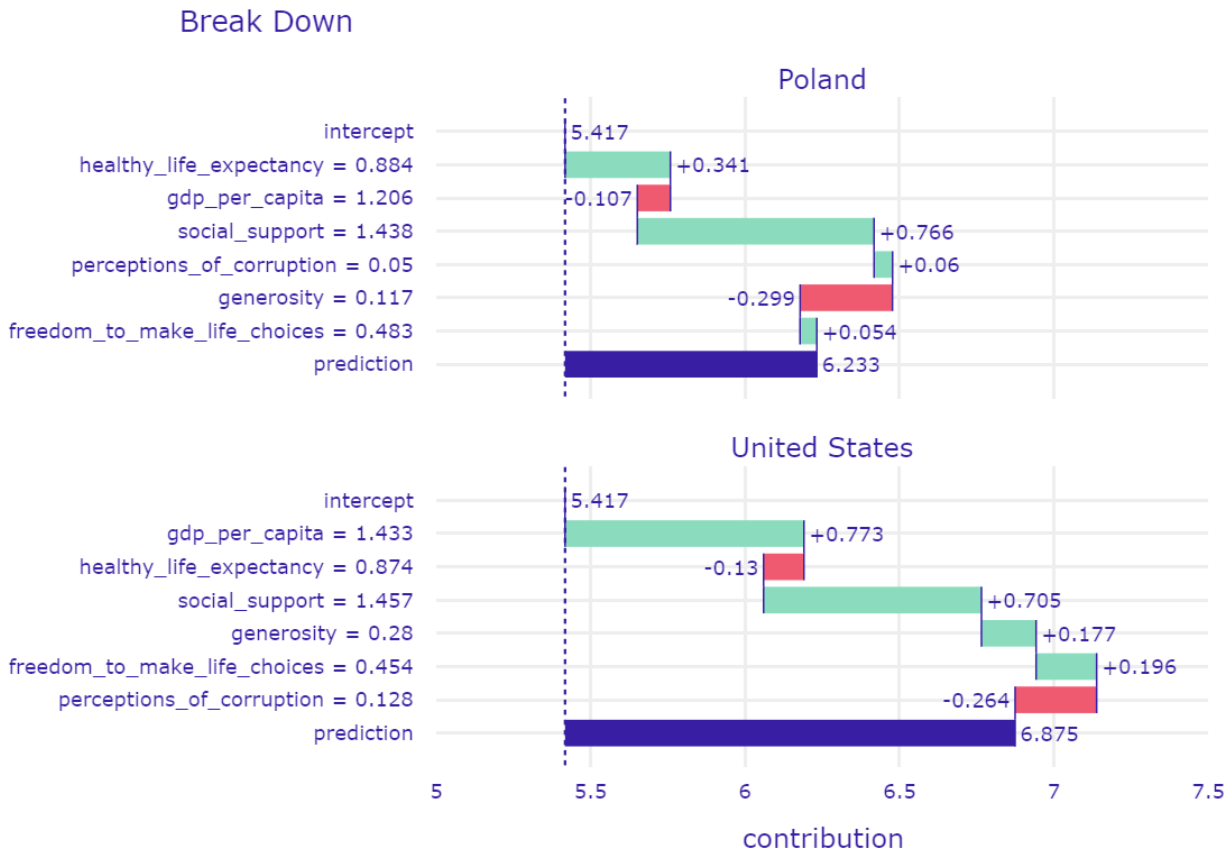


GDP, life expectancy, freedom, social => country happiness score [0, 10]

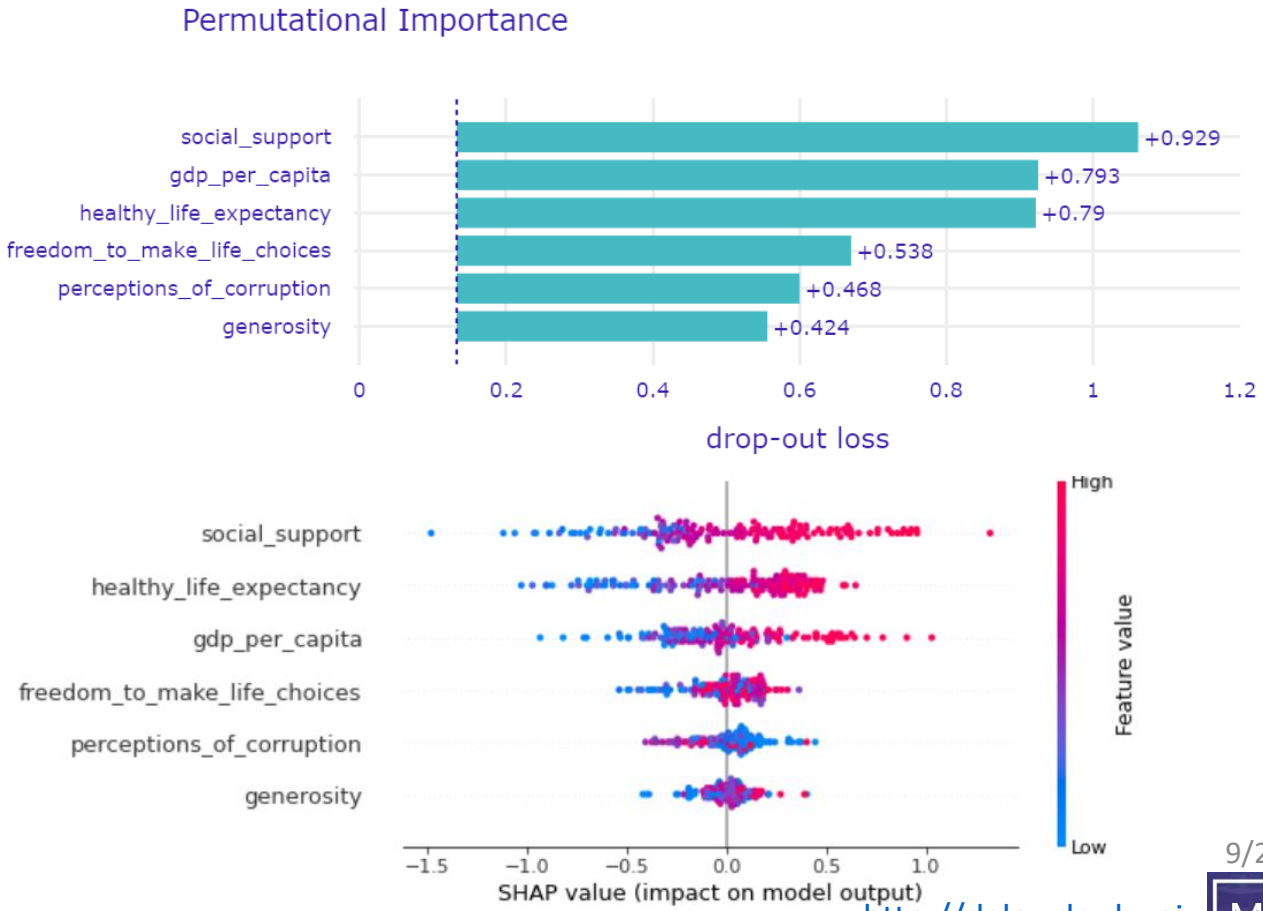
https://www.kaggle.com/unsdsn/world-happiness

parts

predict_parts



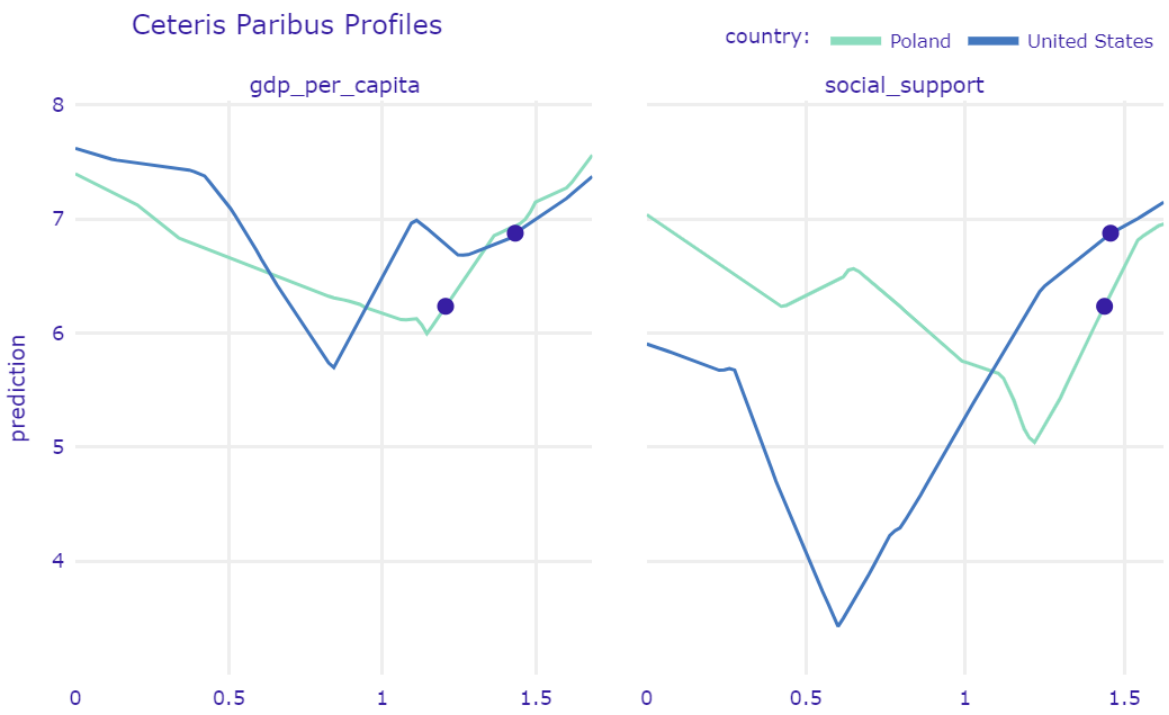
model_parts



http://dalex.drwhy.ai

profile

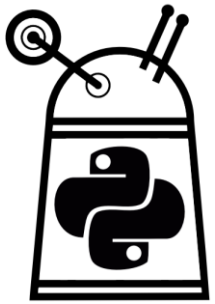
predict_profile



model_profile

http://dalex.drwhy.ai

MI



pip install dalex

import dalex as dx

MODEL

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

DATA

- pandas
- numpy

dx.Explainer

METHODS

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

EXPLANATIONS

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

Explainer

```python
# 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)
```

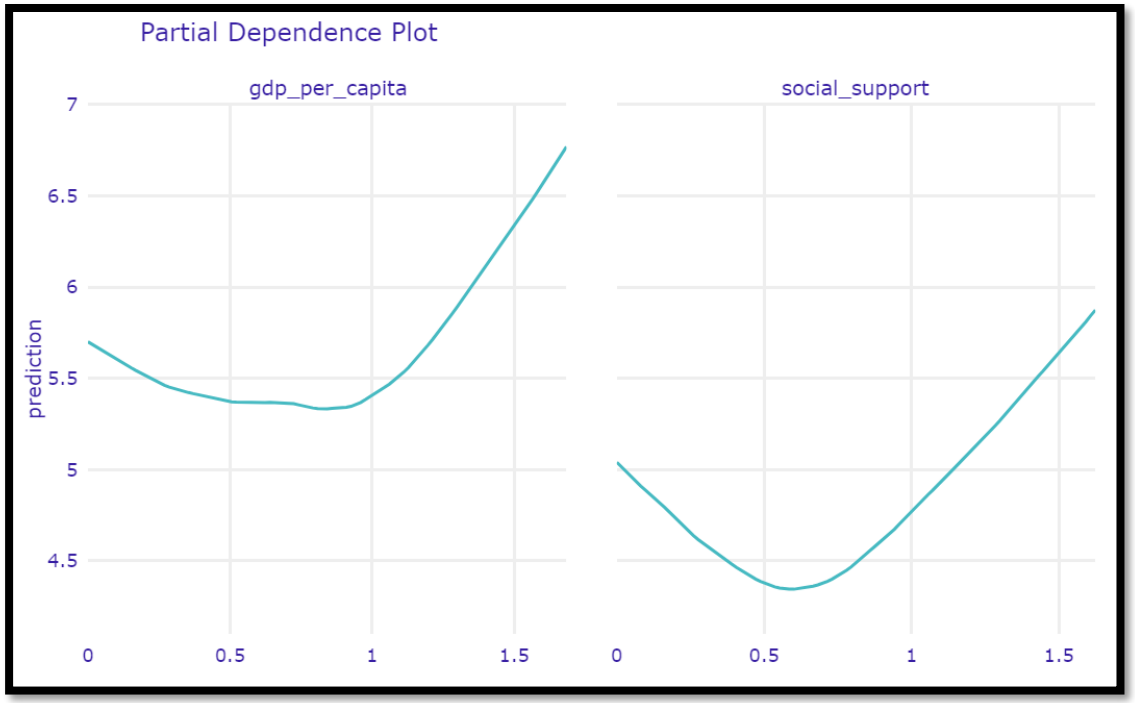
```
Preparation of a new explainer is initiated

  -> 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()
```

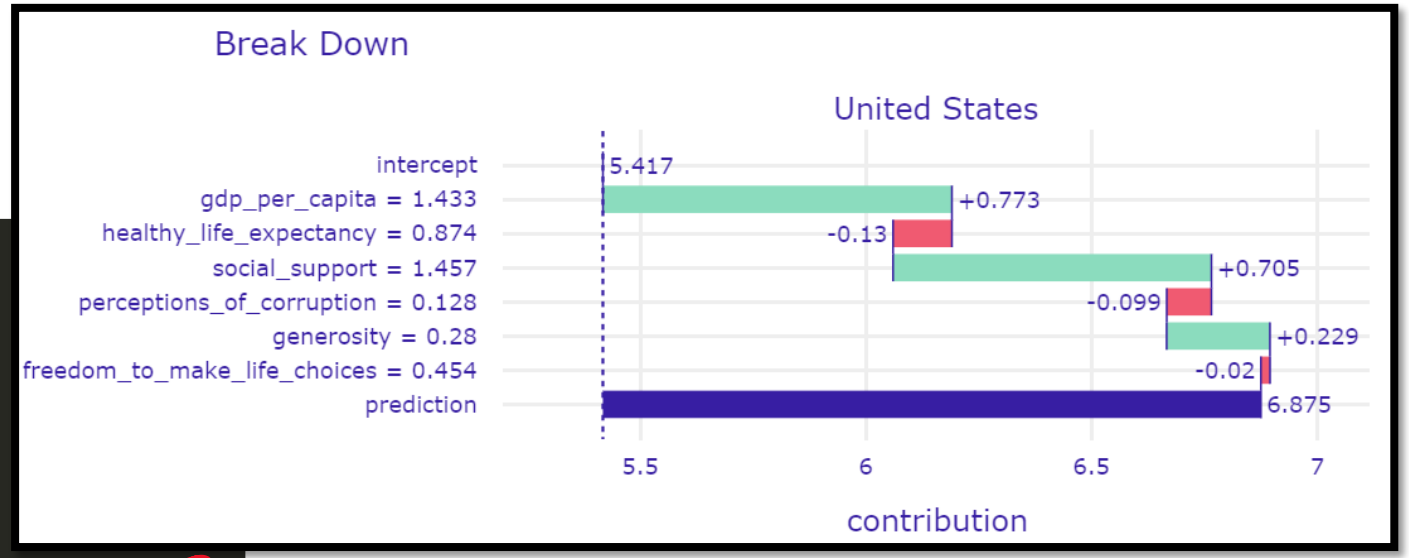
	mse	rmse	r2	mae	mad
	0.017569	0.132549	0.985729	0.072329	0.03636

	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

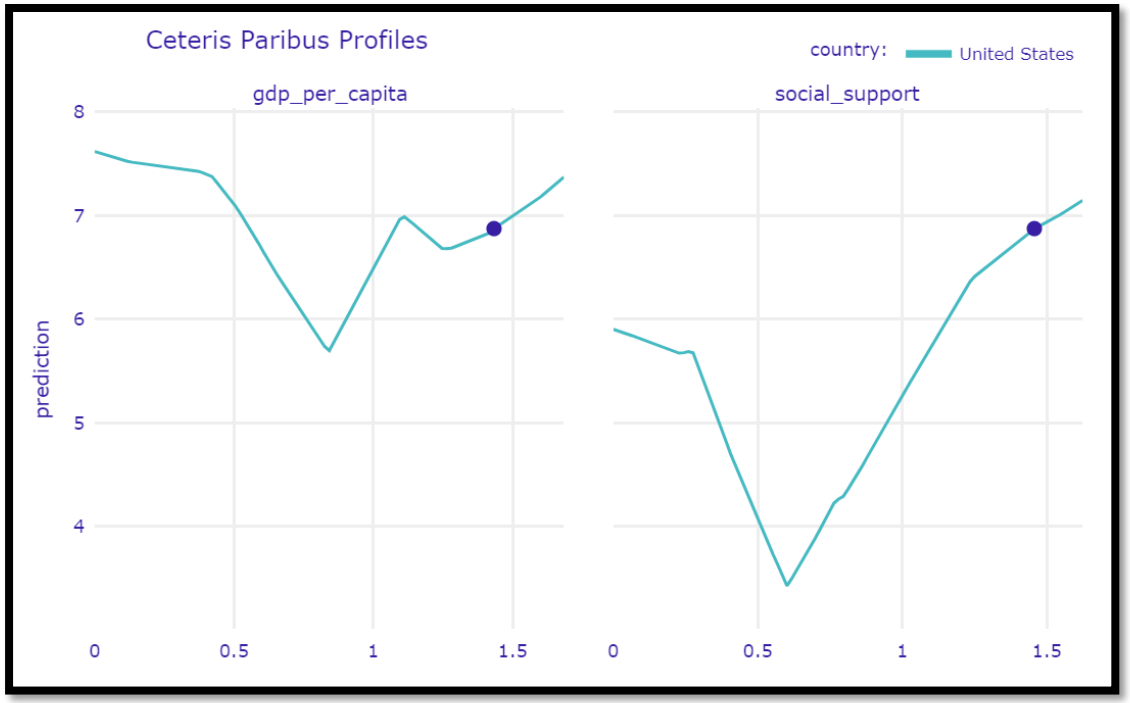


predict

```python
# 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()
```

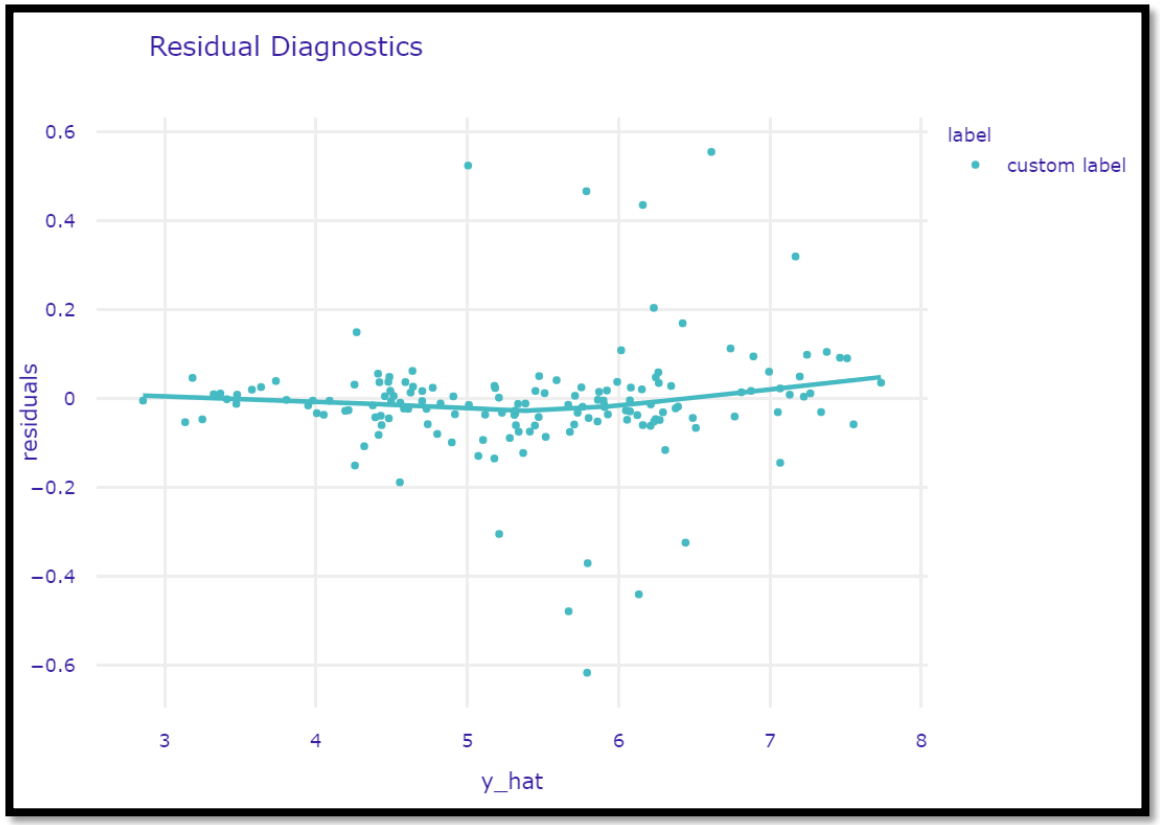


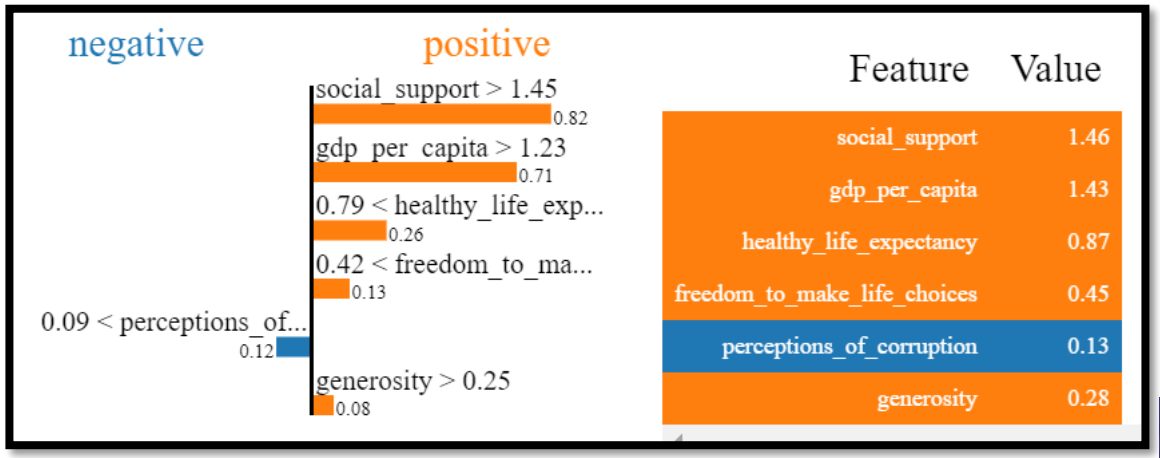
	variable_name	variable_value	variable	cumulative	contribution	sign	position	label
0	intercept	1	intercept	5.417360	5.417360	1.0	7	custom label
1	gdp_per_capita	1.433	gdp_per_capita = 1.433	6.189979	0.772619	1.0	6	custom label
2	healthy_life_expectancy	0.874	healthy_life_expectancy = 0.874	6.059744	-0.130235	-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

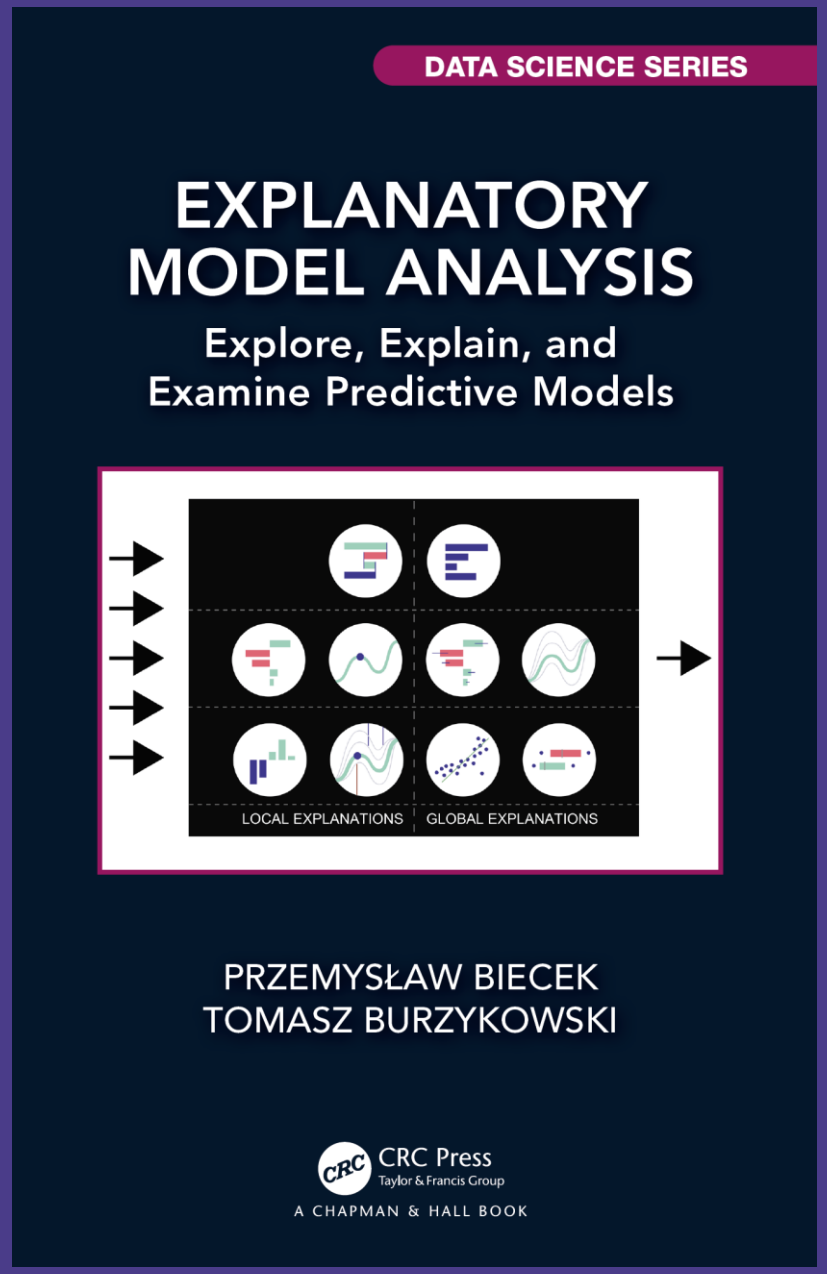


more!

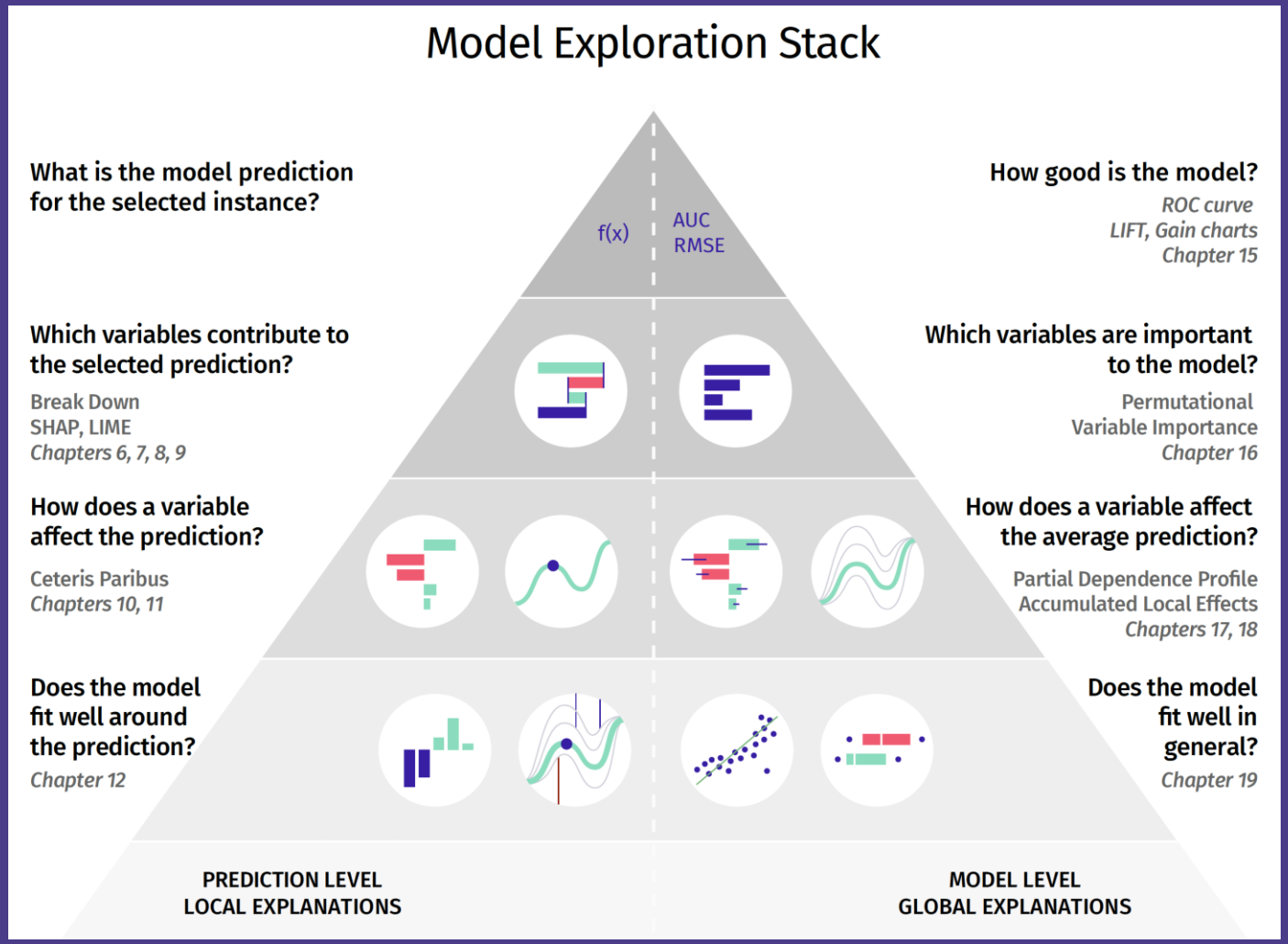
```python
# 10. residuals
explainer.model_diagnostics().plot()

# 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)
```





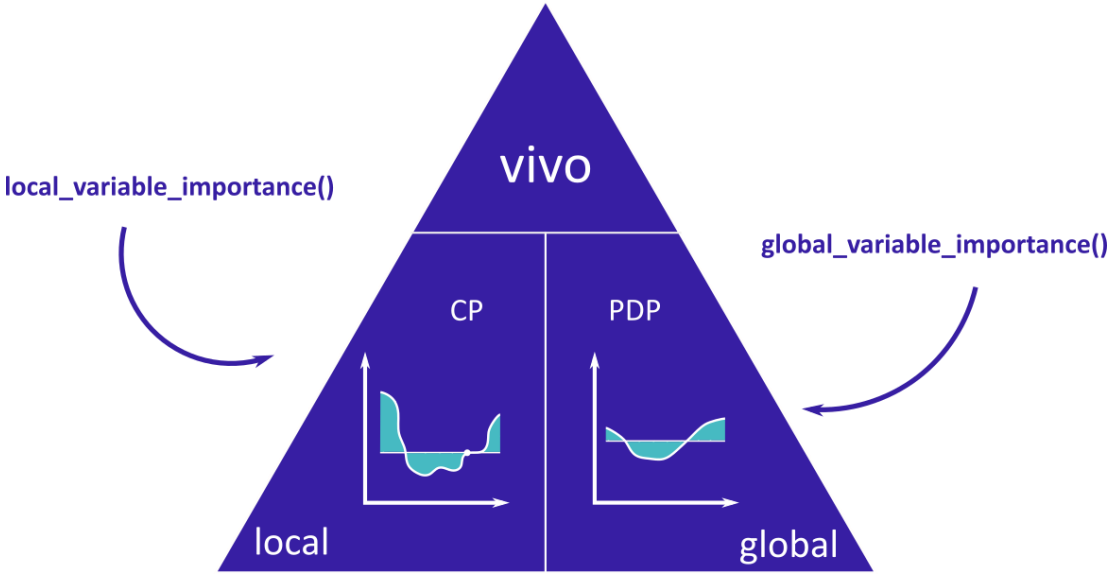


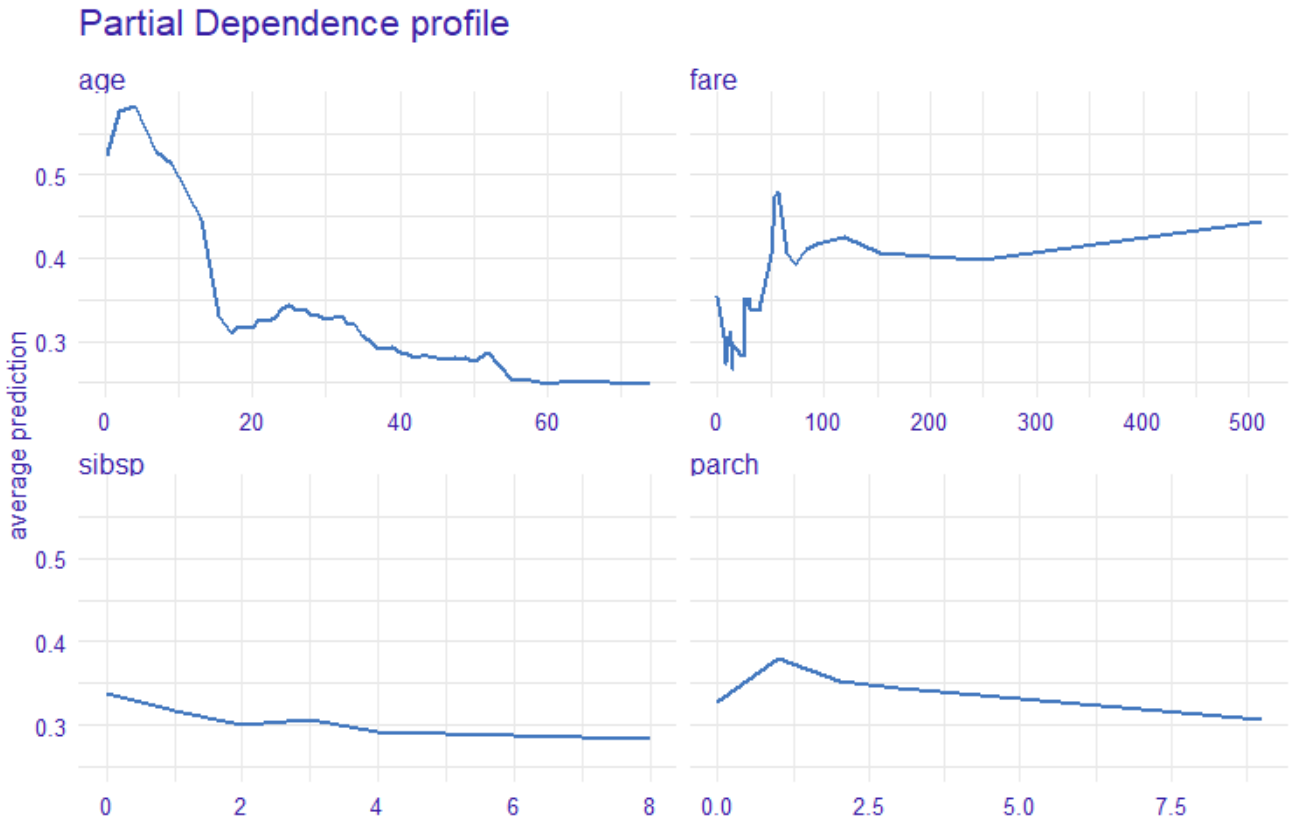
https://ema.drwhy.ai



vivo

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



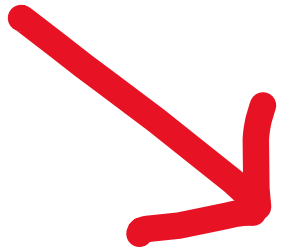


https://github.com/ModelOriented/vivo

MI

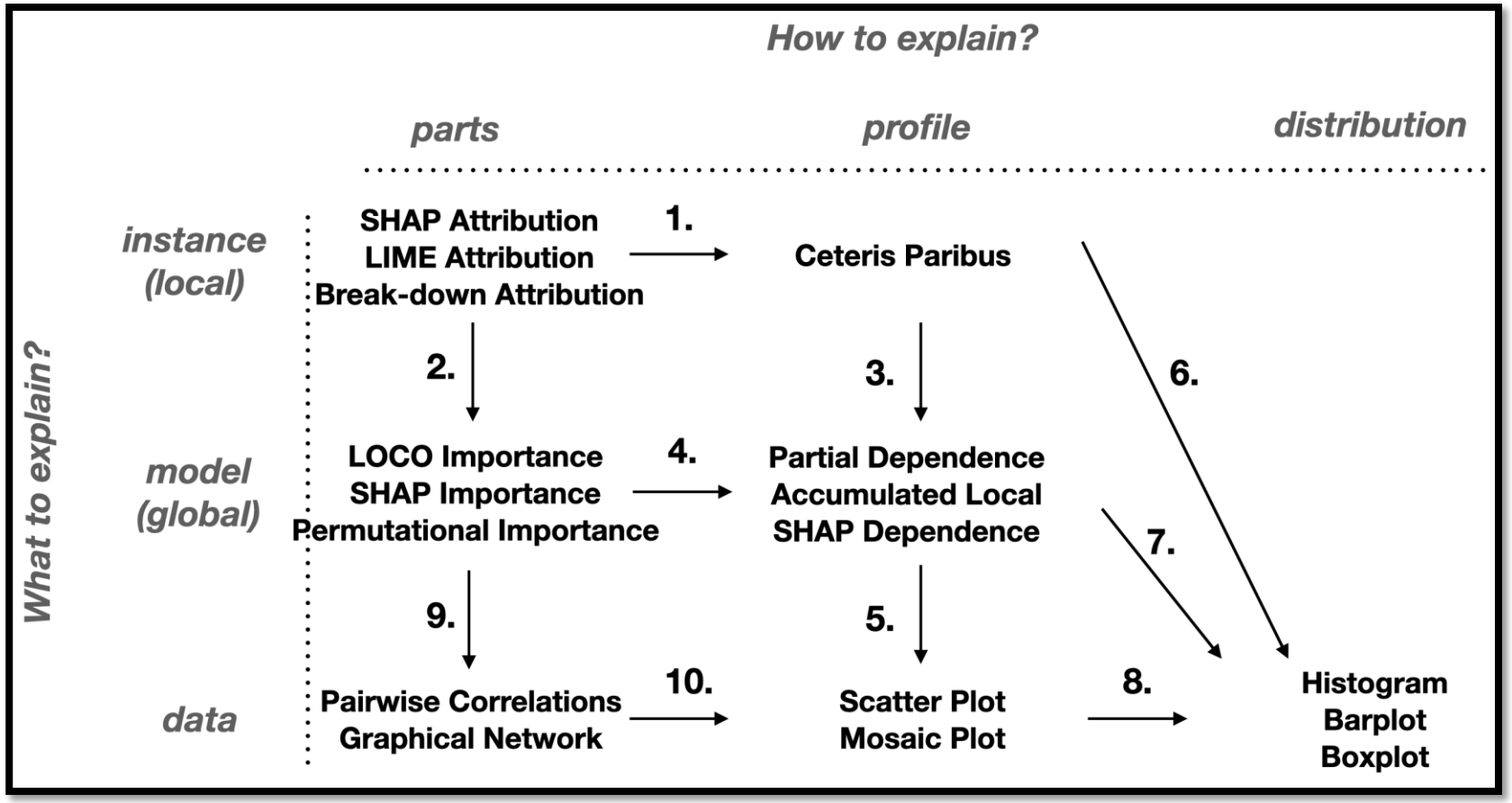
Challenges in explanatory model analysis

- Various stakeholders
- High entry threshold
- Reproducibility
- A lot of coding…



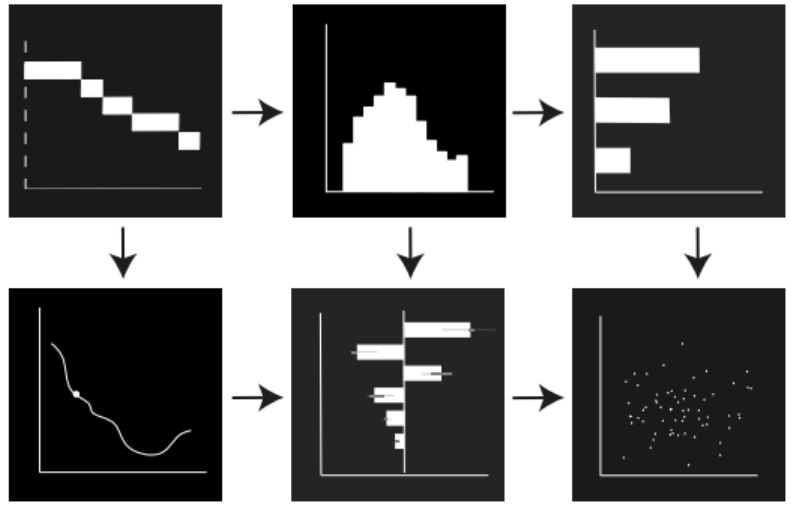
- ✓ Automation
- ✓ Versioning
- ✓ Interactivity
- ✓ Customizability

Interactive explanatory model analysis



II generation explanations

(interactive explanatory model analysis)

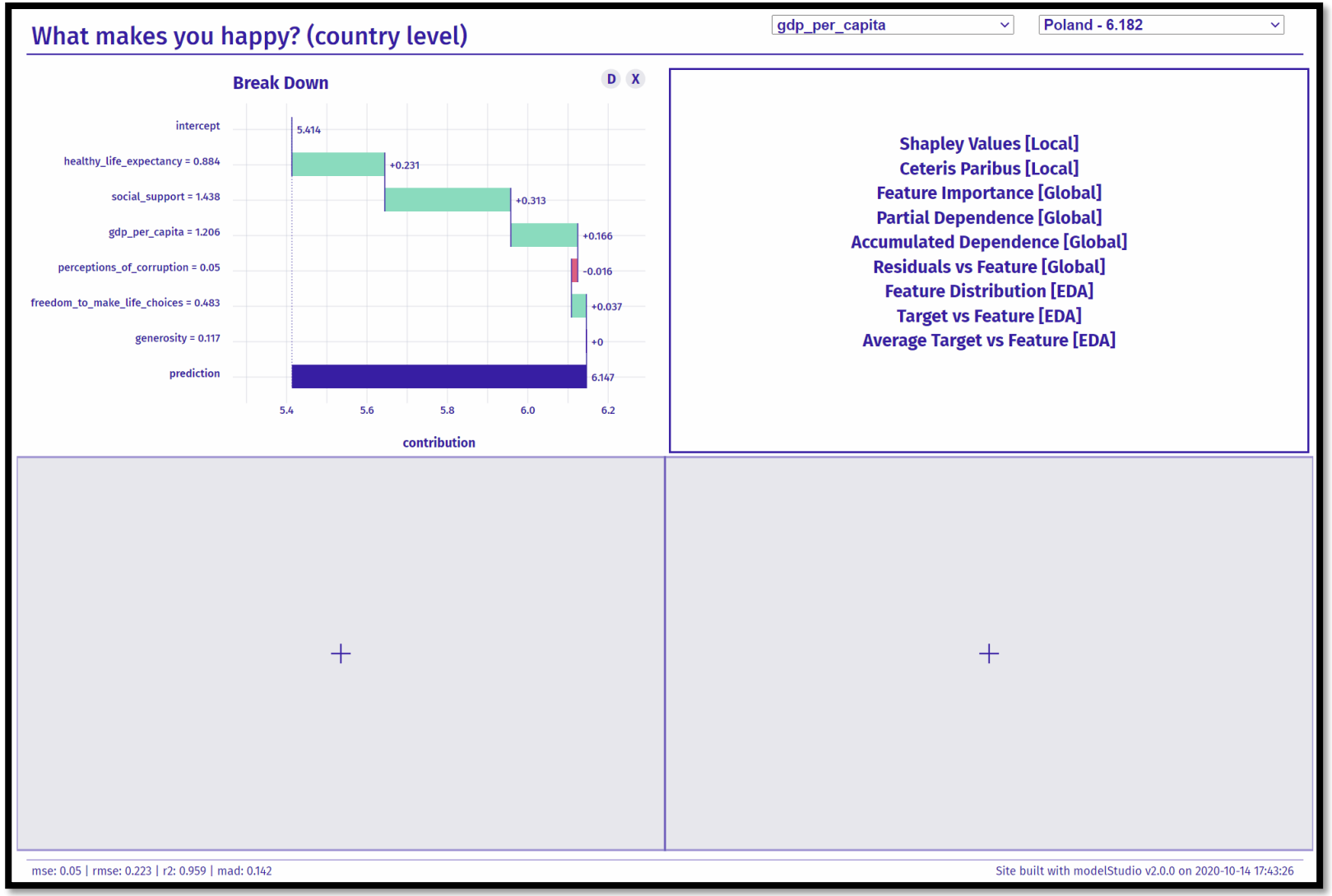


https://iema.drwhy.ai

MI

modelStudio

- creates a dashboard for model analysis
- automated calculations
- customizable interface
- save & share your analysis



http://modelStudio.drwhy.ai

convenient

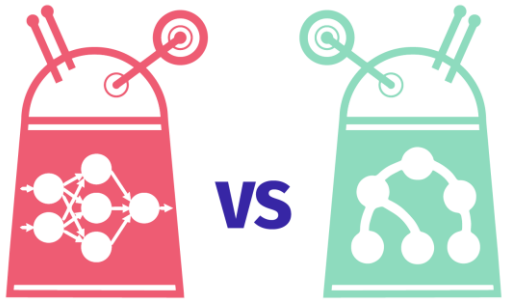
```r
# 0. package
library("DALEX")
library("modelStudio")

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

# 2. model
model <- ...

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

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

DEMO:

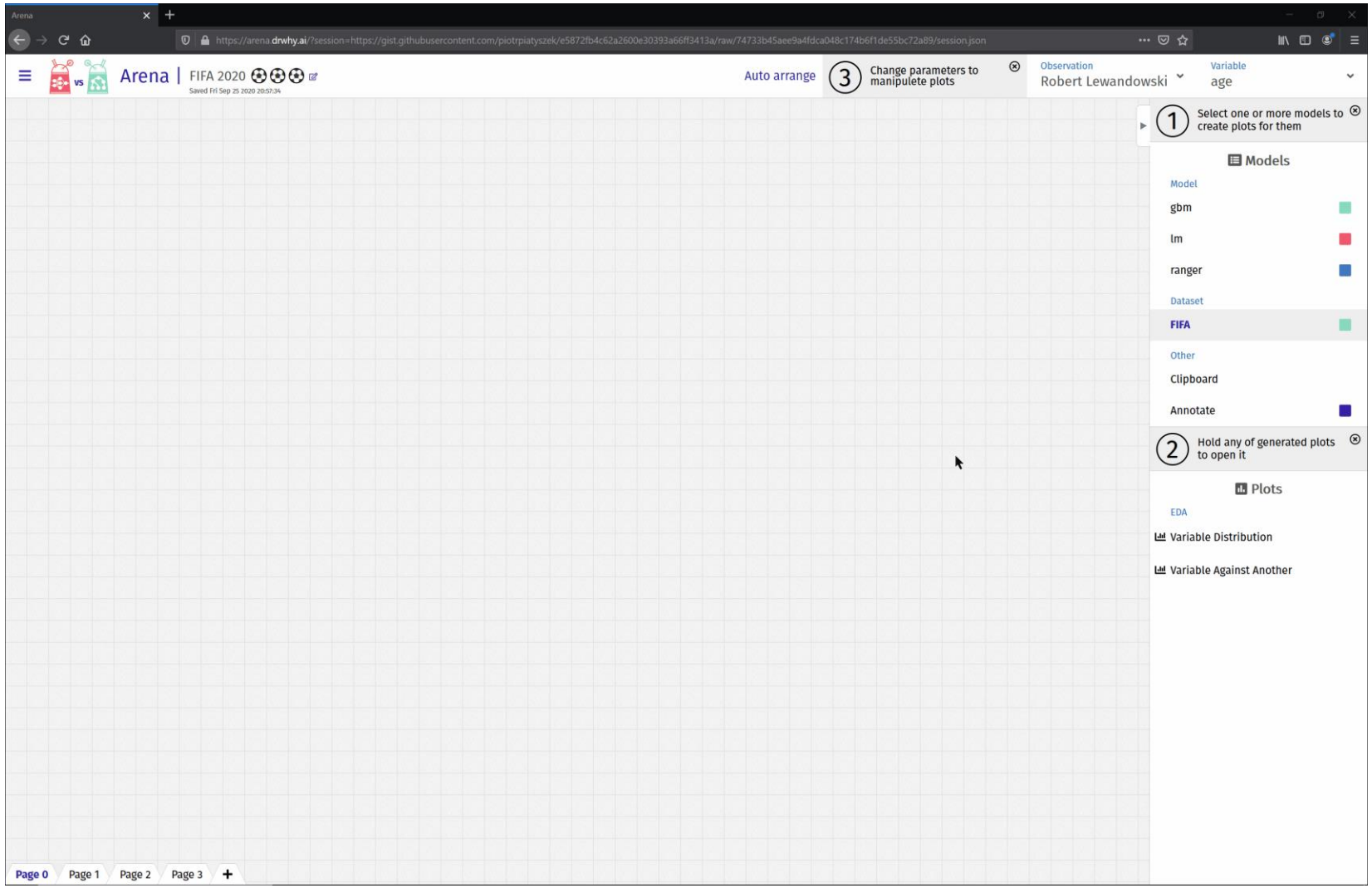
https://github.com/hbaniecki/user-21

Arena

- compare models!
- on multiple datasets!!
- fairness plots
- more pages & cache
- still easy to create...

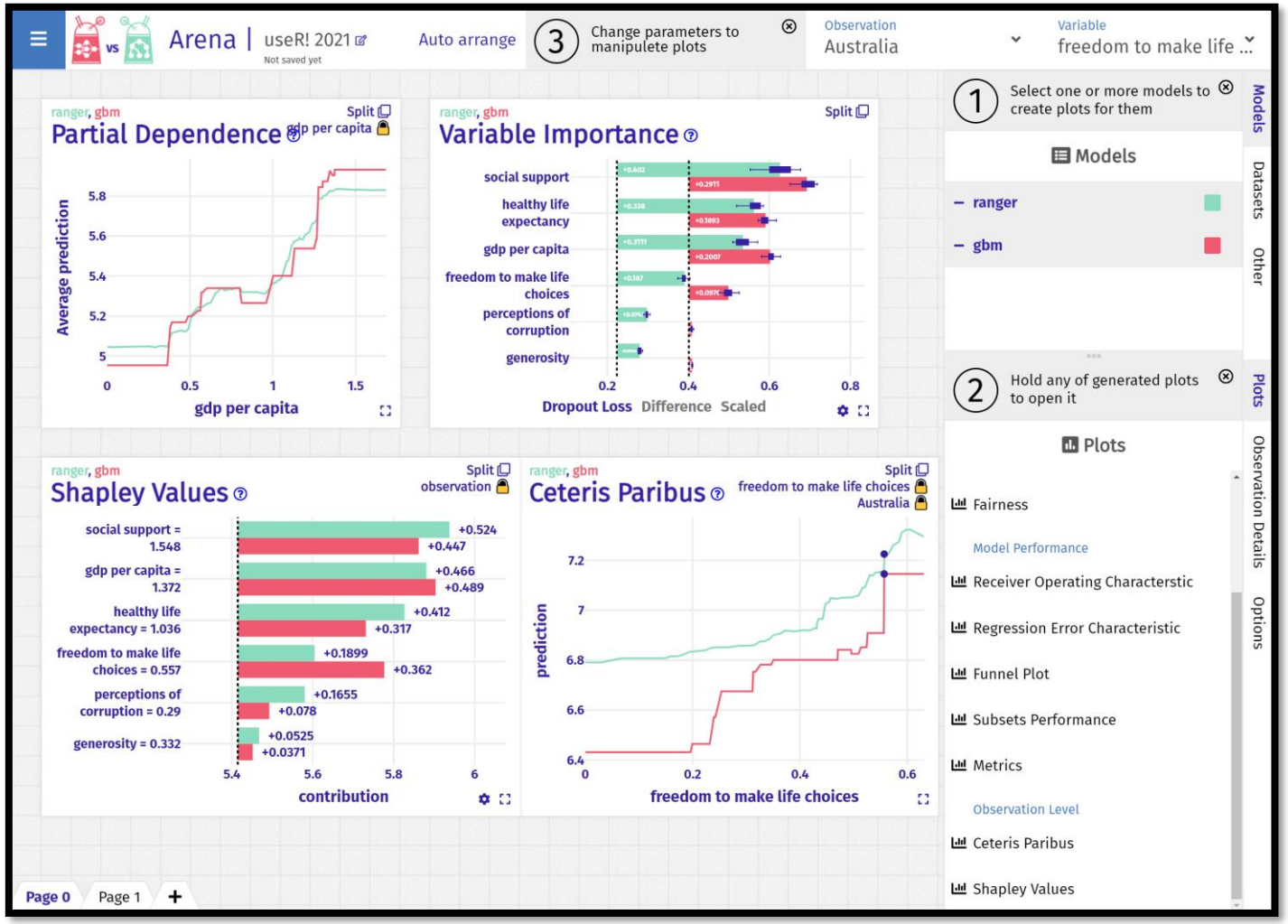


R & Python



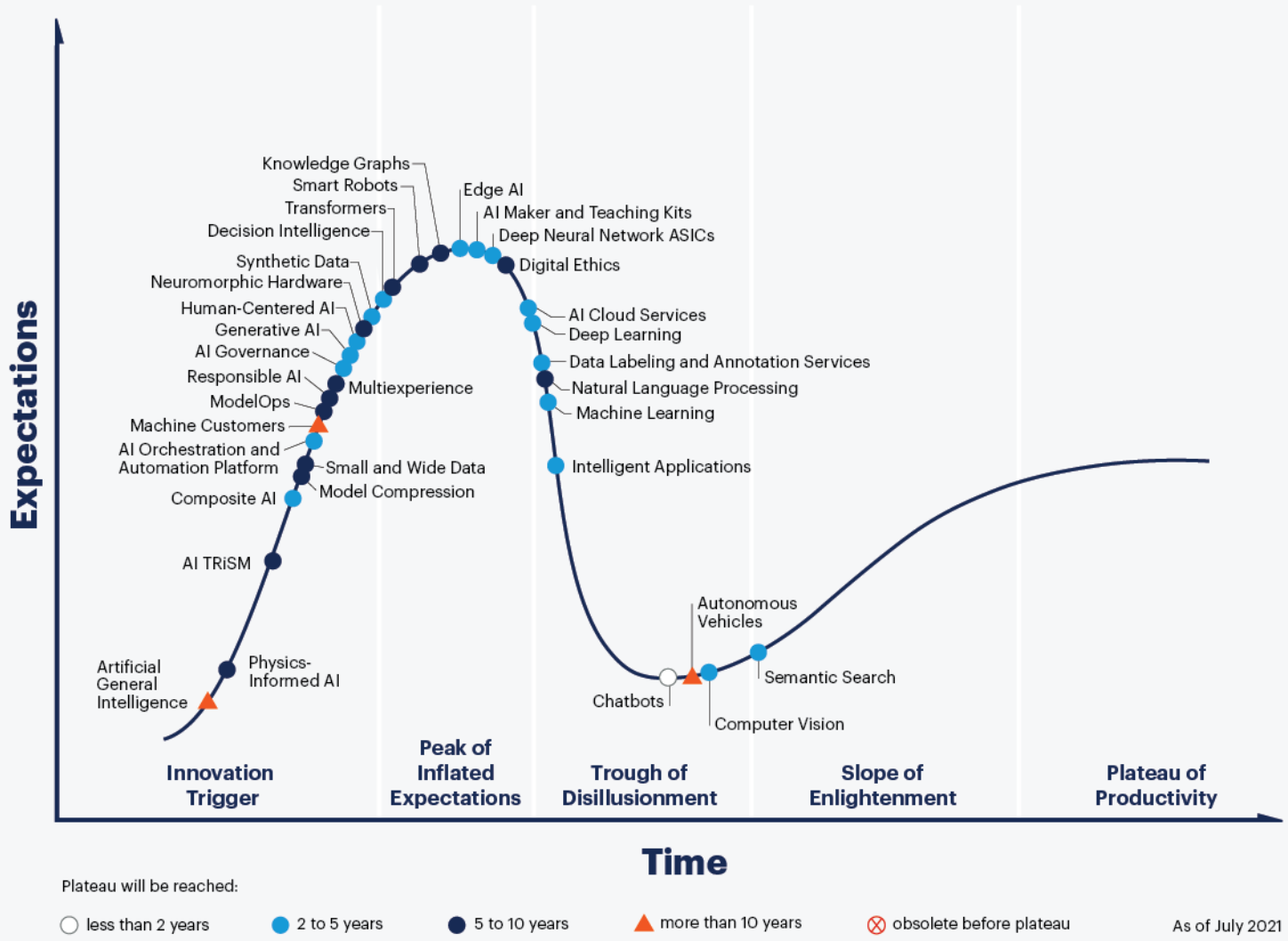
http://arena.drwhy.ai

```r
# 0. package
library("DALEX")
library("arenar")

# 1. data
...
observations <- ...

# 2. models
model_1 <- ...
model_2 <- ...

# 3. explainers
explainer_1 <- DALEX::explain(model_1, X, y)
explainer_2 <- ...

# 4. dashboard
arena <- create_arena(live=TRUE) %>%
  push_model(explainer_1) %>%
  push_model(explainer_2) %>%
  push_observations(observations)

run_server(arena)
```



https://github.com/hbaniecki/user-21

Hype Cycle for Artificial Intelligence, 2021



gartner.com

Source: Gartner

Gartner

Questions?

Feedback appreciated !

Contact	hbaniecki.com
EMA book	ema.drwhy.ai
dalex	dalex.drwhy.ai
modelStudio	modelstudio.drwhy.ai
Arena	arena.drwhy.ai