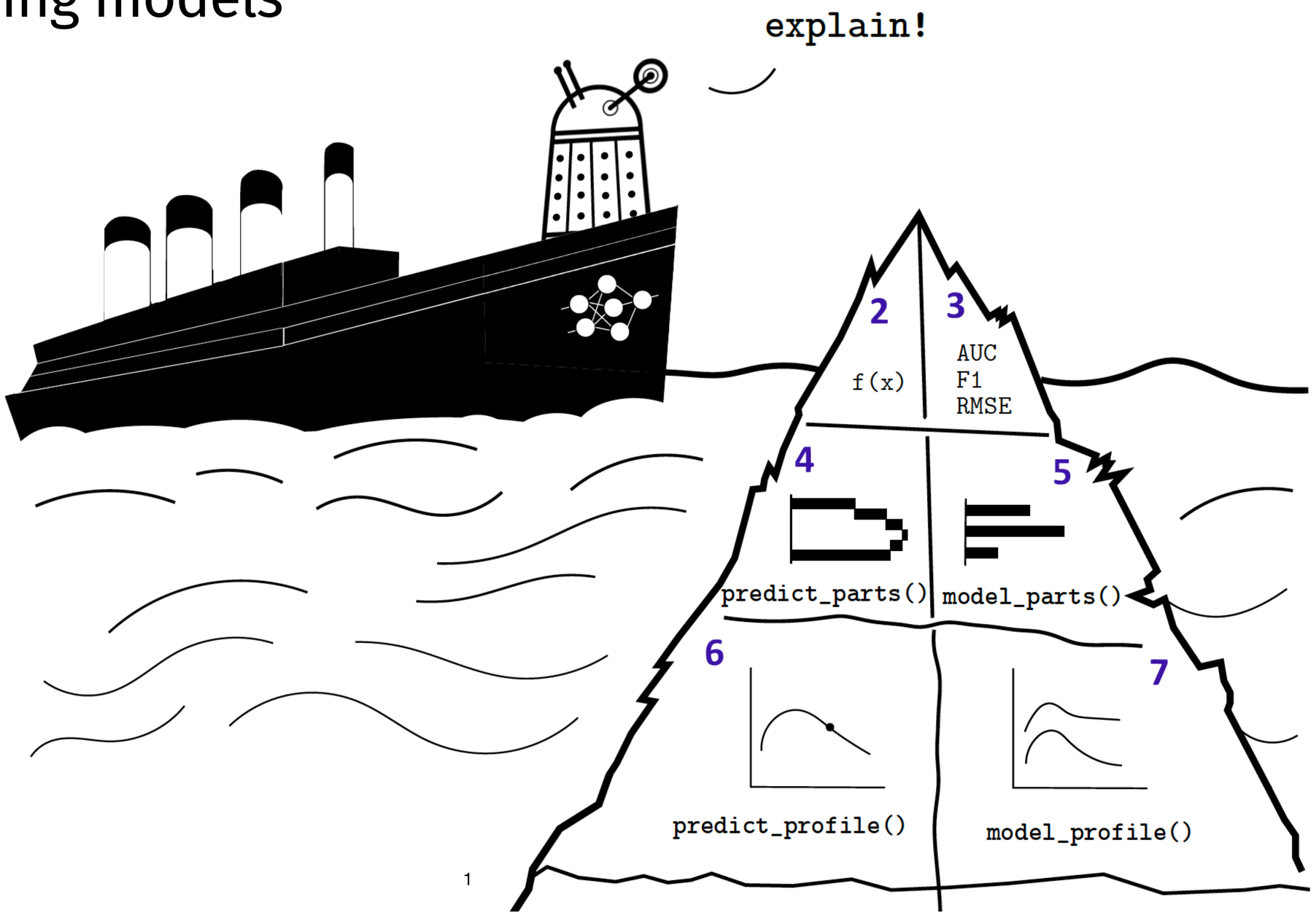
Explanation and exploration of machine learning models with R and DALEX



Przemysław Biecek
Anna Kozak
Szymon Maksymiuk
eRum 20/6/20

Przemysław Biecek

Anna Kozak

Szymon Maksymiuk

MI2DataLab @ Warsaw University of Technology

hbaniecki [python] gh-actions (#228) b8f21a5 12 days ago

5 contributors

102 lines (67 sloc) | 8.36 KB | Raw | Blame | History

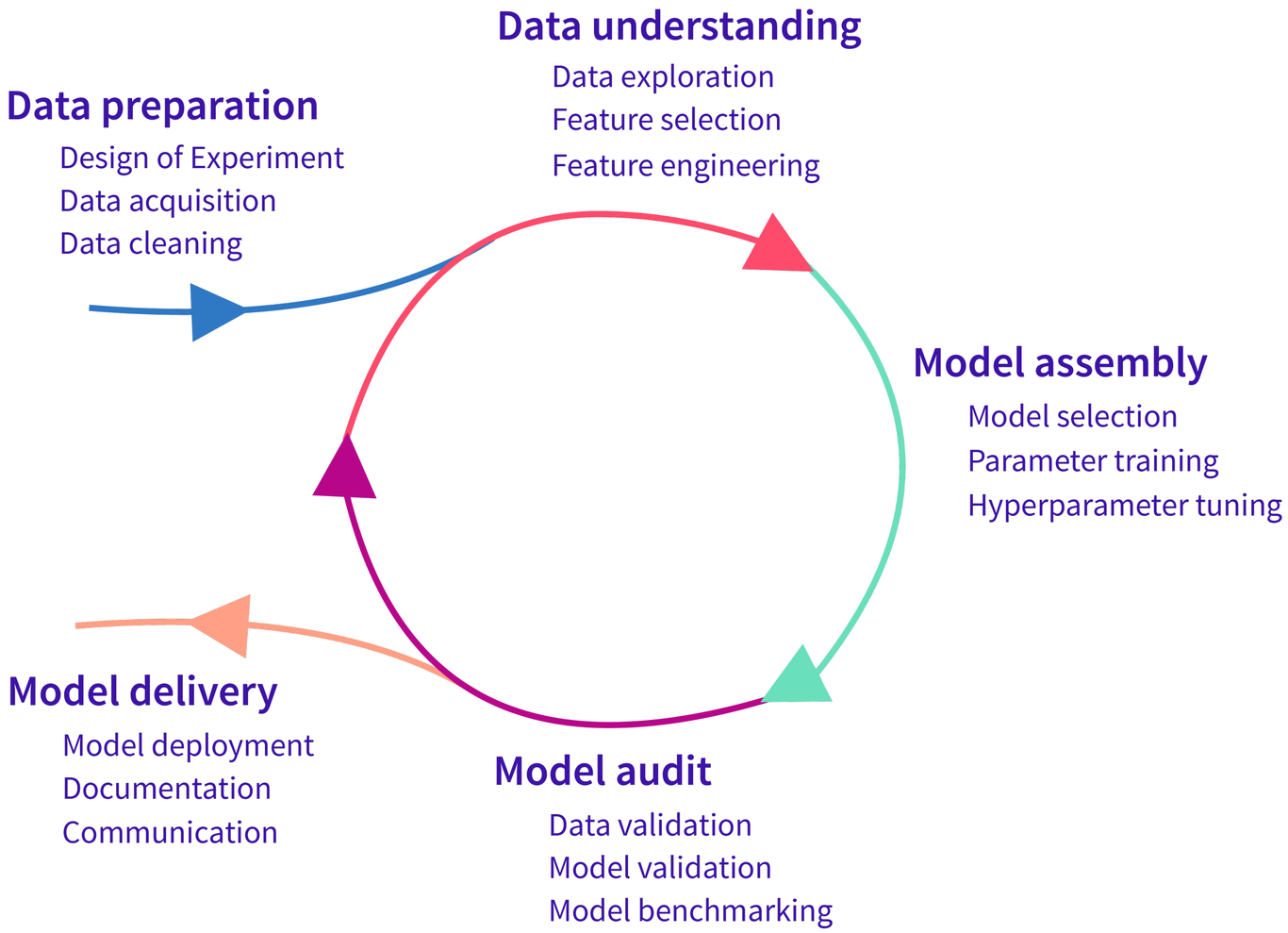
moDel Agnostic Language for Exploration and eXplanation

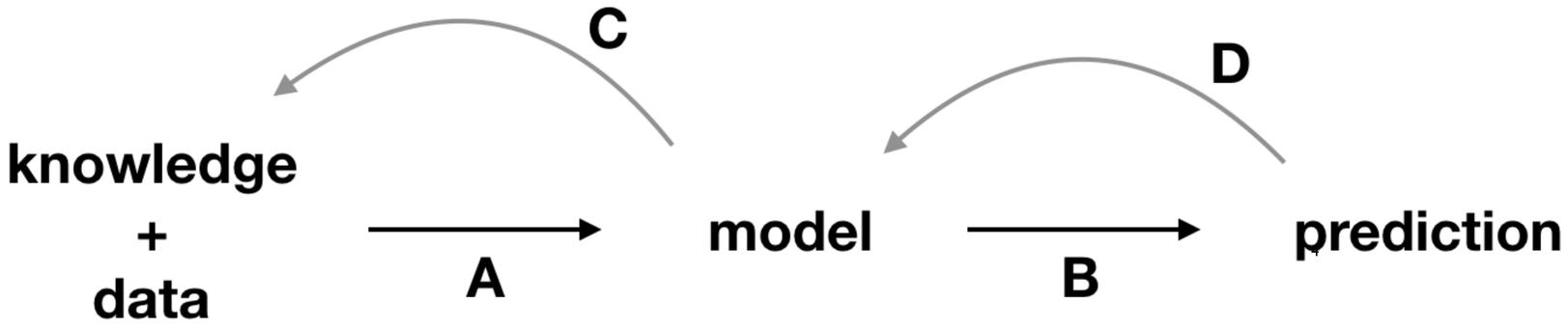
Overview

Unverified black box model is the path to the failure. Opaqueness leads to distrust. Distrust leads to ignoration. Ignoration leads to rejection.

The `DALEX` package xrays any model and helps to explore and explain its behaviour, helps to understand how complex models are working. The main function `explain()` creates a wrapper around a predictive model. Wrapped models may then be explored and compared with a collection of local and global explainers. Recent developents from the area of Interpretable Machine Learning/eXplainable Artificial Intelligence.

The philosophy behind `DALEX` explanations is described in the Explanatory Model Analysis e-book. The `DALEX` package is a


Data understanding
Data exploration
Feature selection
Feature engineering
Data preparation
Design of Experiment
Data acquisition
Data cleaning
Model assembly
Model selection
Parameter training
Hyperparameter tuning
Model delivery
Model deployment
Documentation
Communication
Model audit
Data validation
Model validation
Model benchmarking


C
D
knowledge
+
data
A
model
B
prediction

Fundamental assumption of machine learning models:

The future will be similar to the past

Fundamental assumption of machine learning models:

COVID19

The future will be similar to the past

Google Flu Trends

From Wikipedia, the free encyclopedia

Google Flu Trends was a web service operated by Google. It provided estimates of influenza activity for more than 25 countries. By aggregating Google Search queries, it attempted to make accurate predictions about flu activity. This project was first launched in 2008 by Google.org to help predict outbreaks of flu.[1]

Google Flu Trends is now no longer publishing current estimates. Historical estimates are still available for download, and current data are offerec

Support The Guardian
Contribute
Subscribe
Search jobs | Sign in | Search | International edition
The Guardian
News | Opinion | Sport | Culture | Lifestyle | More
World UK Science Cities Global development Football Tech Business Environment Obituaries

Google

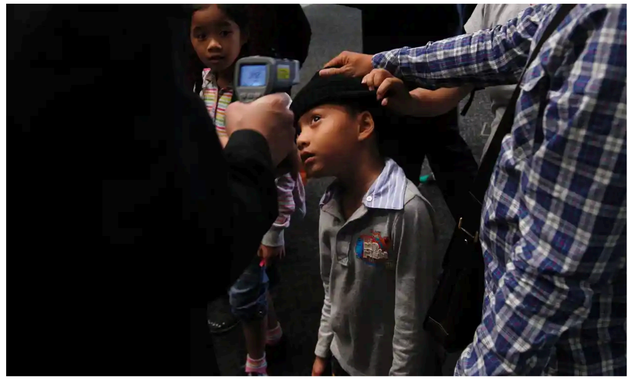
Google Flu Trends is no longer good at predicting flu, scientists find

Researchers warn of 'big data hubris' and the importance of updating analytical models, claiming Google has made inaccurate forecasts for 100 of 108 weeks

Charles Arthur
@charlesarthur
Thu 27 Mar 2014 10.27 GMT

200 7

This article is over 4 years old



Airport security personnel take a body temperature reading of a boy as he arrives at Hong Kong International Airport April 9,2013, following concerns over a deadly strain of bird flu. Photograph: Tyrone Siu/Reuters

most viewed


Dozens of Indian paramilitaries killed in Kashmir car bombing


Live Brexit: blow to May's authority as MPs reject her motion by 303 votes to 258 - as it happened


Theresa May defeated on Brexit again as ERG Tories abstain


Live Trump to sign government funding bill and declare national emergency - live


Andrew McCabe says officials discussed removing Trump after Comey firing

Forbes
Billionaires Innovation Leadership Money Con

61,215 views | Mar 23, 2014, 09:00am

Why Google Flu Is A Failu

Steven Salzberg Contributor
Pharma & Healthcare

It seemed like such a good idea at the time.

https://www.wired.com/2015/10/can-learn-epic-failure-google-flu-trends/

REUTERS World Business Markets Politics TV

Midterm Elections | Imprisoned In Myanmar | Sectors Up Close | Breakingviews | Investing | Future of Money | Charged: The Future of Aut

BUSINESS NEWS OCTOBER 10, 2018 / 5:12 AM / A MONTH AGO

Amazon scraps secret AI showed bias against wor

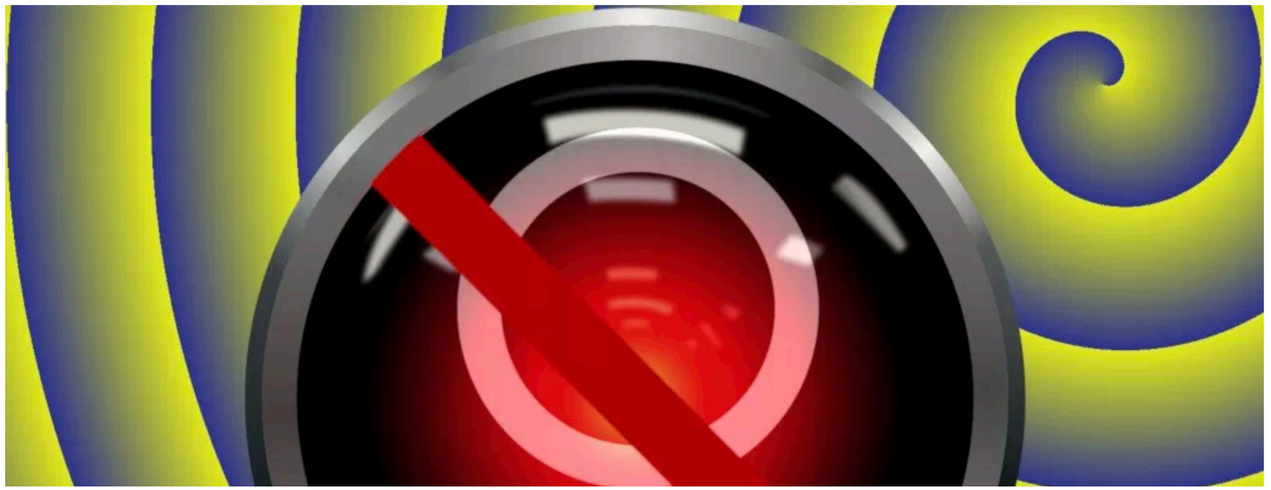
Jeffrey Dastin

SAN FRANCISCO (Reuters) - Amazon.com Inc's specialists uncovered a big problem: their

The group created 500 computer models focused on specific job functions and locations. They taught each to recognize some 50,000 terms that showed up on past candidates' resumes. The algorithms learned to assign little significance to skills that were common across IT applicants, such as the ability to write various computer codes, the people said.

Instead, the technology favored candidates who described themselves using verbs more commonly found on male engineers' resumes, such as "executed" and "captured," one person said.

Amazon trained a sexism-fighting, resume-screening AI with sexist hiring data, so the bot became sexist



THE VERGE TECH SCIENCE C

TECH AMAZON ARTIFICIAL INTELLIGENCE

21

Amazon reportedly scraps internal AI recruiting tool that was biased against women

The secret program penalized applications that contained the word "women's"

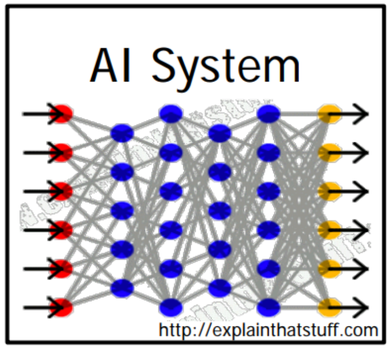
Model debugging

(a) Husky classified as wolf

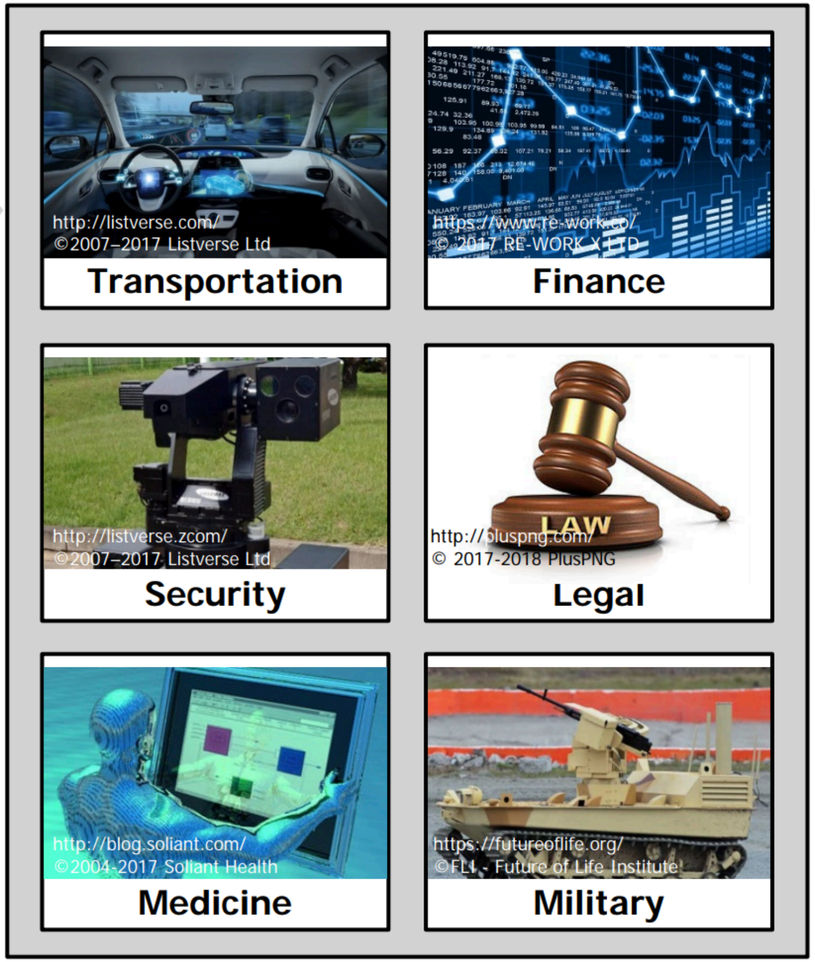
(b) Explanation

"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

The Need for Explainable AI



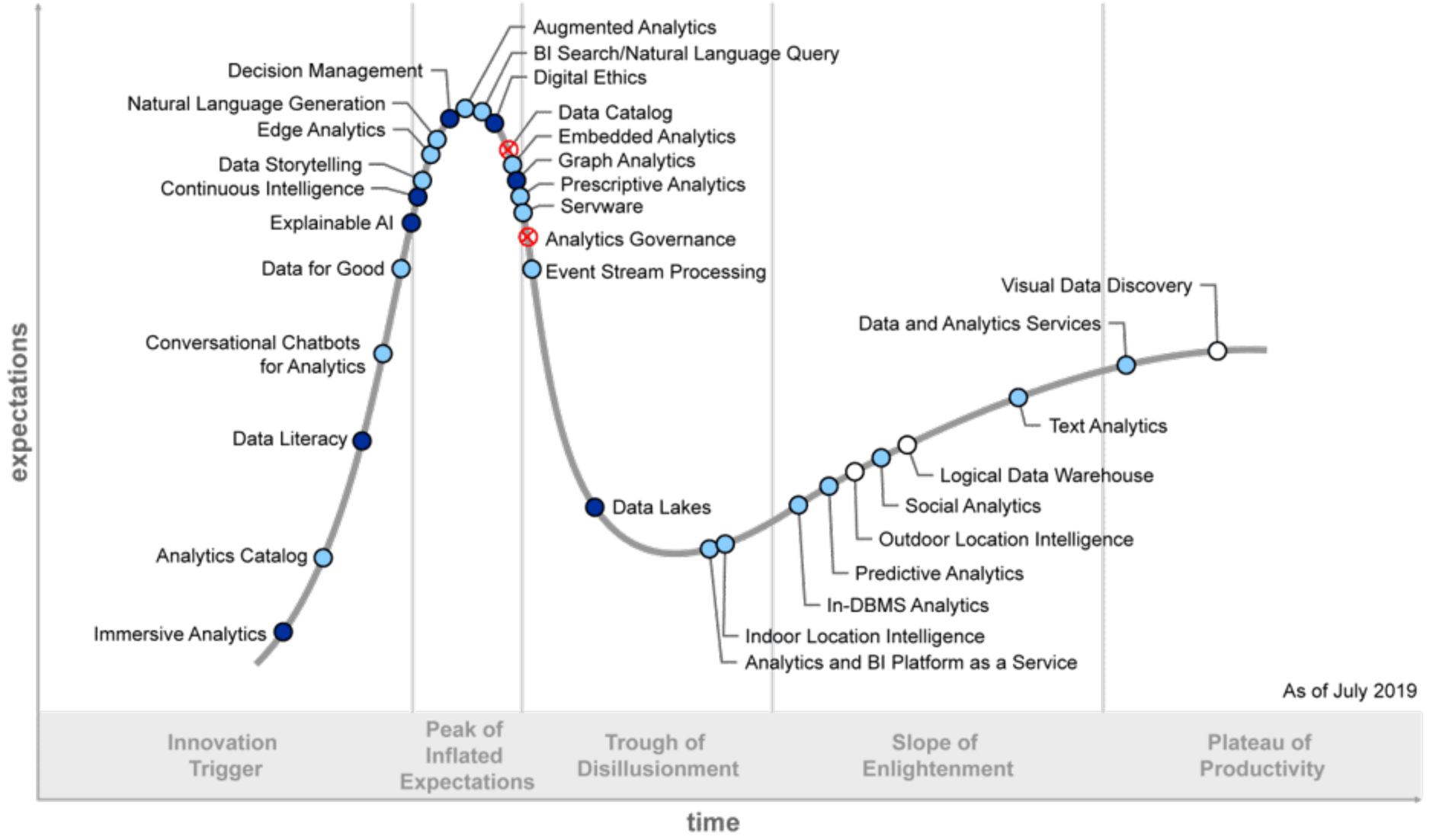
- We are entering a new age of AI applications
- Machine learning is the core technology
- Machine learning models are opaque, non-intuitive, and difficult for people to understand



- Why did you do that?
- Why not something else?
- When do you succeed?
- When do you fail?
- When can I trust you?
- How do I correct an error?

- The current generation of AI systems offer tremendous benefits, but their effectiveness will be limited by the machine's inability to explain its decisions and actions to users
- Explainable AI will be essential if users are to understand, appropriately trust, and effectively manage this incoming generation of artificially intelligent partners

Hype Cycle for Analytics and Business Intelligence, 2019



Source: Gartner

Why XAI ?

D Instance level explanations help **to debug a model** / understand why the model was wrong.

A Dataset level explanations help **to audit a model**, check if it complies with the requirements.

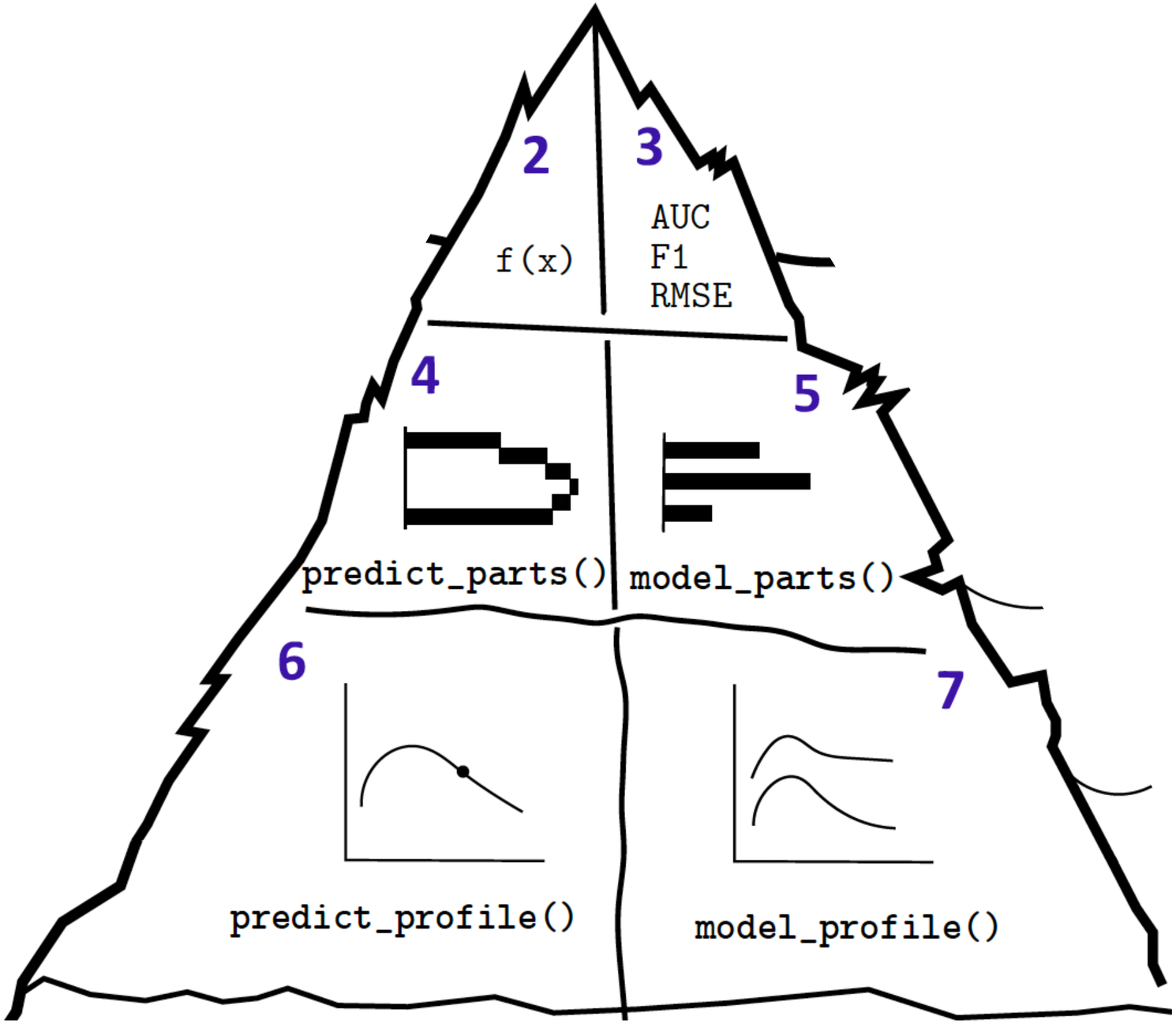
L Understand **what to do to leverage** a prediction for a model.

E Allow to confront the model with **expert knowledge**. If consistent we are more likely to **trust the prediction**.

X **Examine** models e.g. in the **champion-challenger analysis**.

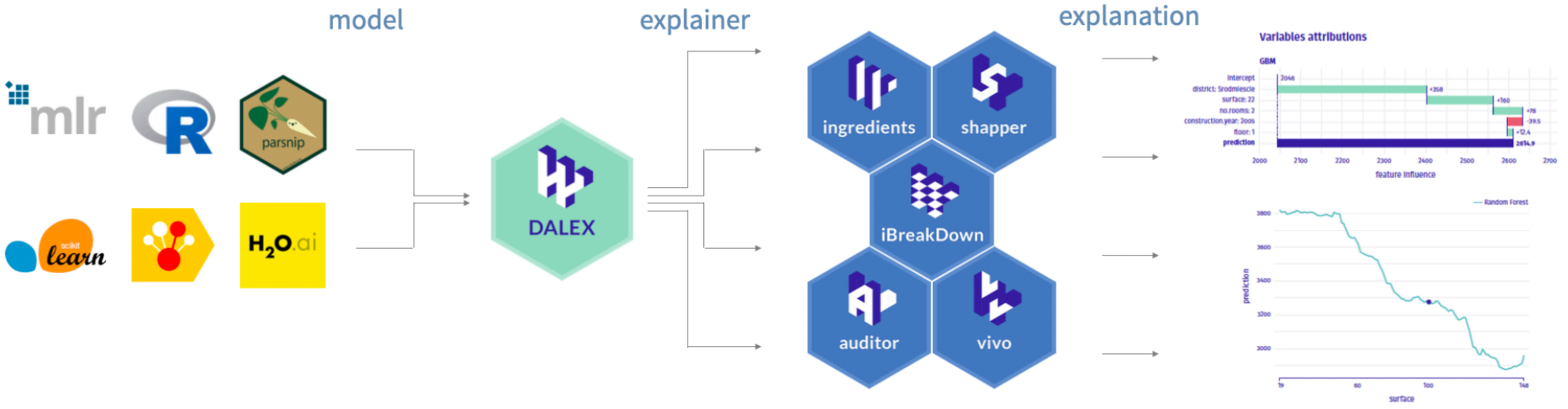
Explanatory Model Analysis is a subfield of eXplainable Artificial Intelligence. It covers the tools and processes for analysing predictive models in order to better understand their behaviour.

Table of contents



Architecture

```
ranger(y ~ ., data = df)  %>%  explain()  %>%  model_parts()  %>%  plot()
```

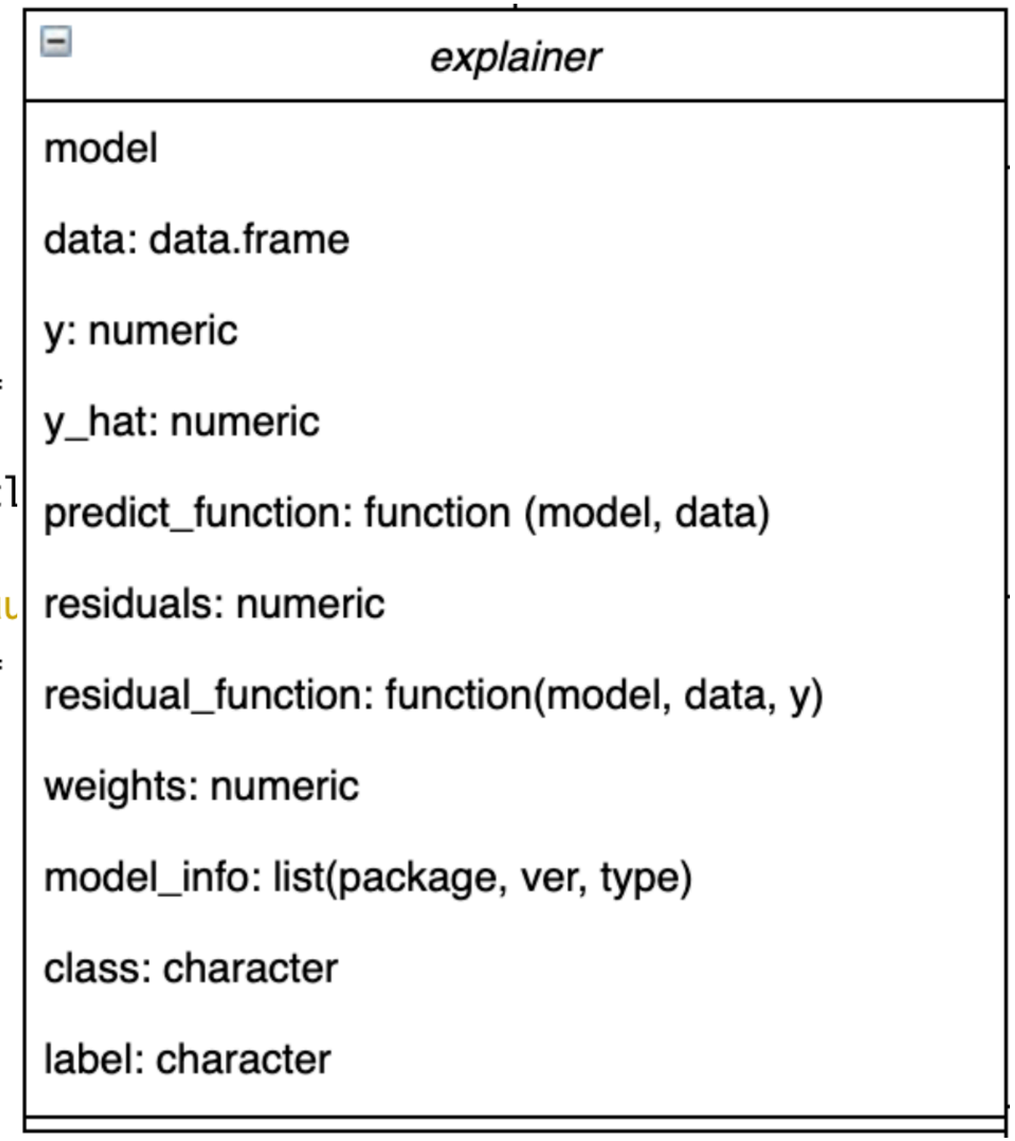


```
> model_ranger <- ranger(survived ~ ., data = titanic_imputed,
+                        classification = TRUE, probability = TRUE)
```

```
> model_ranger <- ranger(survived ~ ., data = titanic_imputed,
+                        classification = TRUE, probability = TRUE)
> exp_ranger <- explain(model_ranger,
+                       data = titanic_imputed[,1:7],
+                       y = titanic_imputed$survived)
Preparation of a new explainer is initiated
  -> model label       :  ranger  (  default  )
  -> data              :  2207  rows  7  cols
  -> target variable   :  2207  values
  -> predict function  :  yhat.ranger  will be used (  default  )
  -> predicted values  :  numerical, min =  0.01149205 , mean =  0.3216525 , max
 =  0.9912358
  -> model_info        :  package ranger , ver. 0.12.1 , task classification (
efault  )
  -> residual function :  difference between y and yhat (  default  )
  -> residuals         :  numerical, min =  -0.7916279 , mean =  0.0005042793 ,
ax =  0.8860613
  A new explainer has been created!
```

```
> model_ranger <- ranger(survived ~ ., data = titanic_imputed,
+                        classification = TRUE, probability = TRUE)
> exp_ranger <- explain(model_ranger,
+                       data = titanic_imputed[,1:7],
+                       y = titanic_imputed$survived)
Preparation of a new explainer is initiated
  -> model label       :  ranger  (  default  )
  -> data              :  2207  rows  7  cols
  -> target variable   :  2207  values
  -> predict function  :  yhat.ranger  will be used (  default  )
  -> predicted values  :  numerical, min =  0.01149205 , mean =  0.3216525 , max
 =  0.9912358
  -> model_info        :  package ranger , ver. 0.12.1 , task classification (
efault  )
  -> residual function :  difference between y and yhat (  default  )
  -> residuals         :  numerical, min =  -0.7916279 , mean =  0.0005042793 ,
ax =  0.8860613
  A new explainer has been created!
> exp_ranger$model_info
Package: ranger
Package version: 0.12.1
Task type: classification
```

```
> model_ranger <- ranger(survived ~ ., data = titanic_imputed,
+                        classification = TRUE, probability = TRUE)
> exp_ranger <- explain(model_ranger,
+                       data = titanic_imputed[,1:7],
+                       y = titanic_imputed$survived)
Preparation of a new explainer is initiated
  -> model label       :  ranger  (  default  )
  -> data              :  2207  rows  7  cols
  -> target variable   :  2207  values
  -> predict function  :  yhat.ranger  will be used (  default
  -> predicted values  :  numerical, min =  0.01149205 , mean =
 =  0.9912358
  -> model_info        :  package ranger , ver. 0.12.1 , task cl
efault  )
  -> residual function :  difference between y and yhat (  defau
  -> residuals         :  numerical, min =  -0.7916279 , mean =
ax =  0.8860613
  A new explainer has been created!
> exp_ranger$model_info
Package: ranger
Package version: 0.12.1
Task type: classification
```

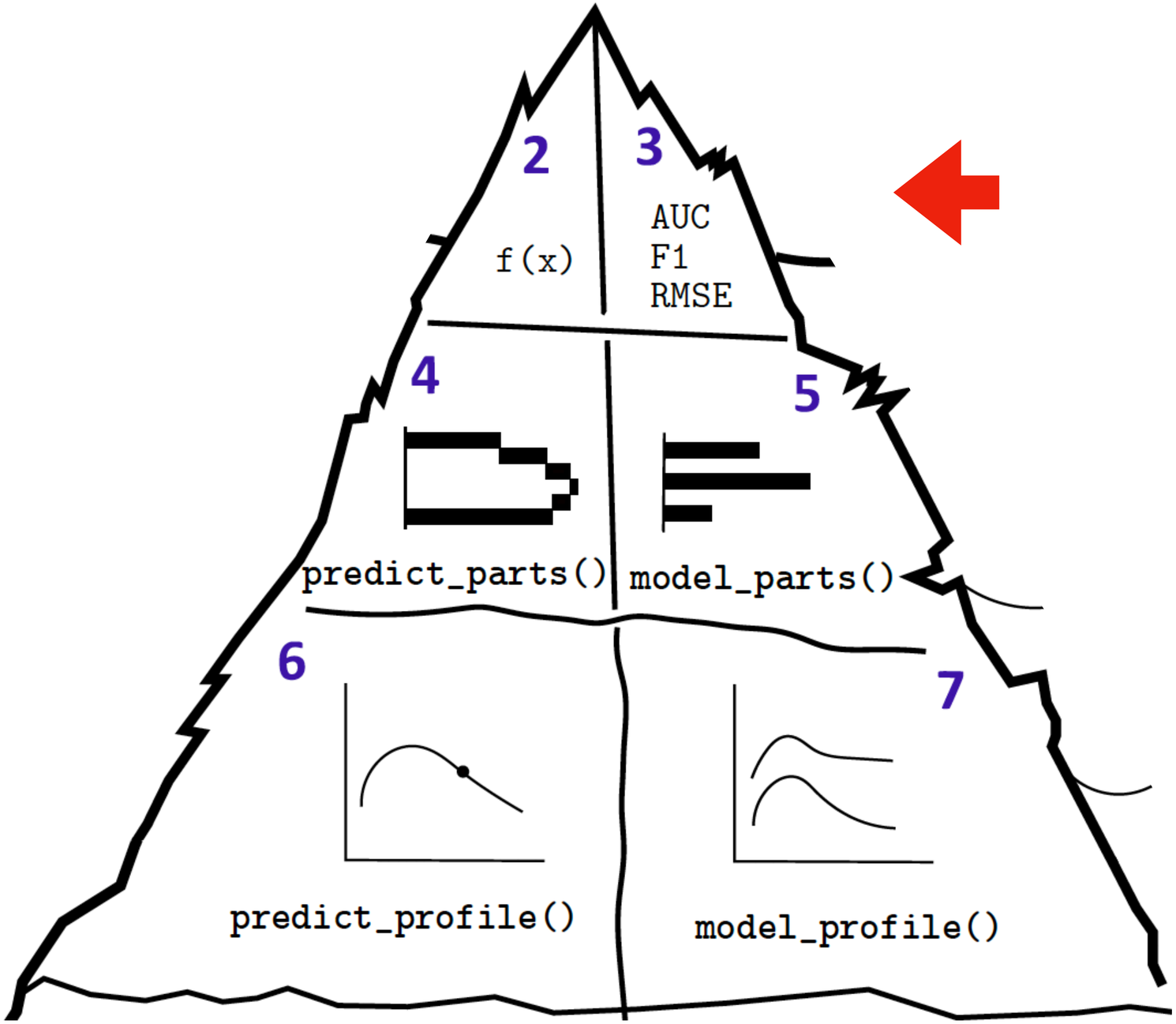


Your turn

1. Use apartments dataset from DALEX package.
2. Build a predictive model for m2.price (with ranger or other ML library).
3. Create an explainer for this model

Model performance

Table of contents



$$MSE(f) = \frac{1}{n}\sum_{i}^{n}(f(x_i) - y_i)^2$$

$$RMSE(f) = \sqrt{MSE(f, X, y)}$$

$$ACC(f) = (TP + TN)/n$$

$$Prec(f) = TP/(TP + FP)$$

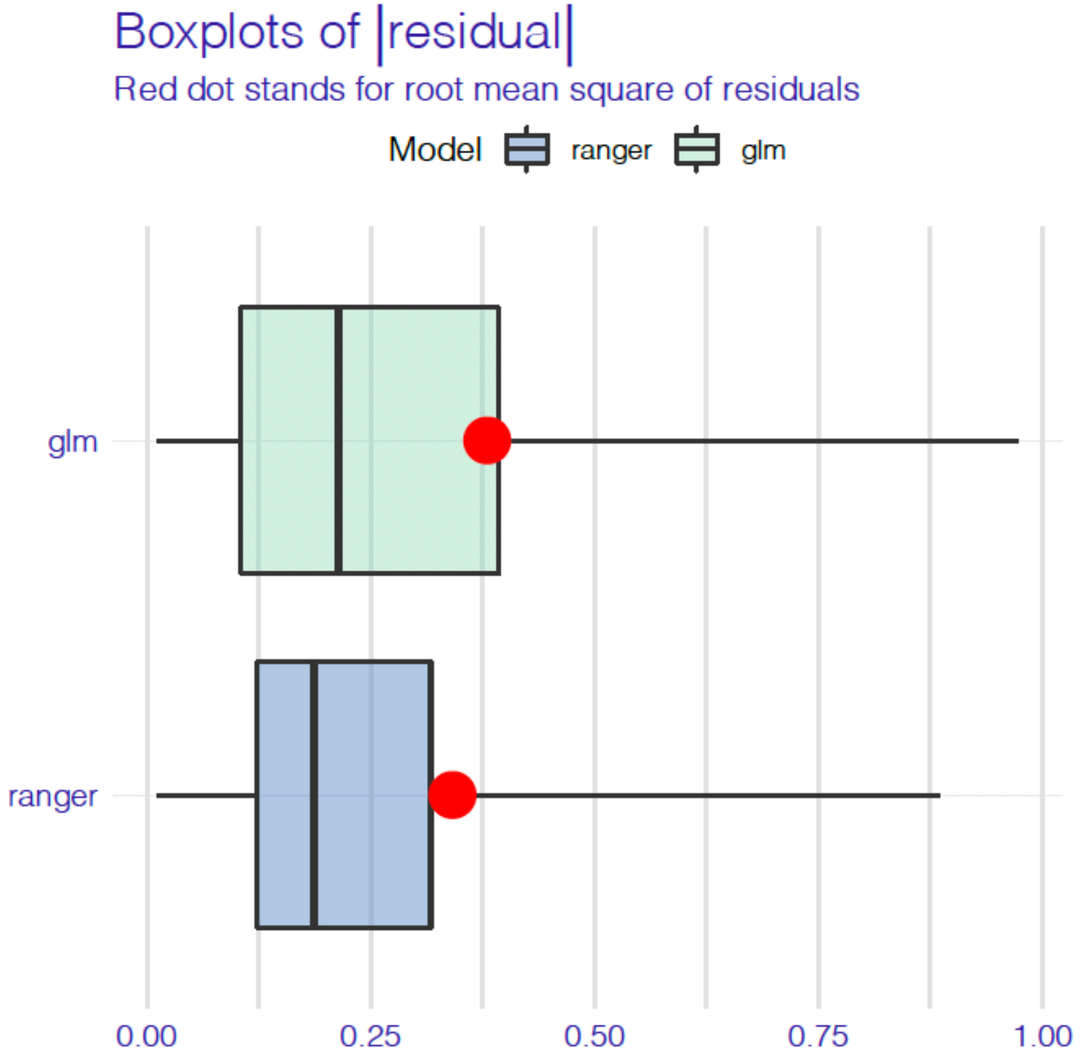
$$Recall(f) = TP/(TP + FN)$$

$$F1(f) = 2\frac{Prec(f) * Recall(f)}{Prec(f) + Recall(f)}$$

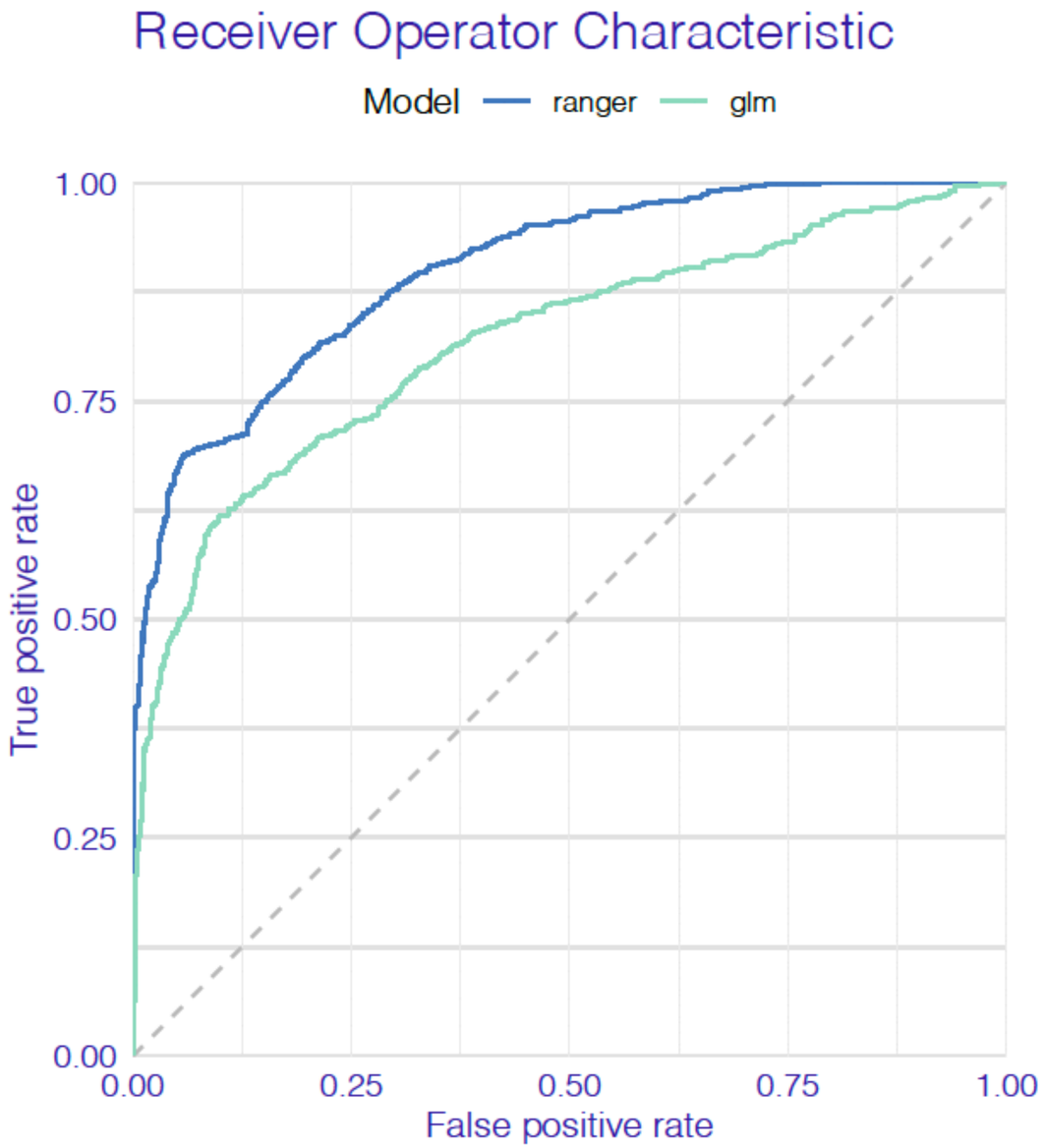
```
mp_ranger <- model_performance(exp_ranger)
mp_ranger
## Measures for:  classification
## recall      : 0.5977496
## precision   : 0.9081197
## f1          : 0.72095
## accuracy    : 0.8509289
## auc         : 0.752529
```

```
mp_rms <- model_performance(exp_rms)
mp_rms
## Measures for:  classification
## recall      : 0.5977496
## precision   : 0.767148
## f1          : 0.6719368
## accuracy    : 0.8119619
## auc         : 0.8174259
```

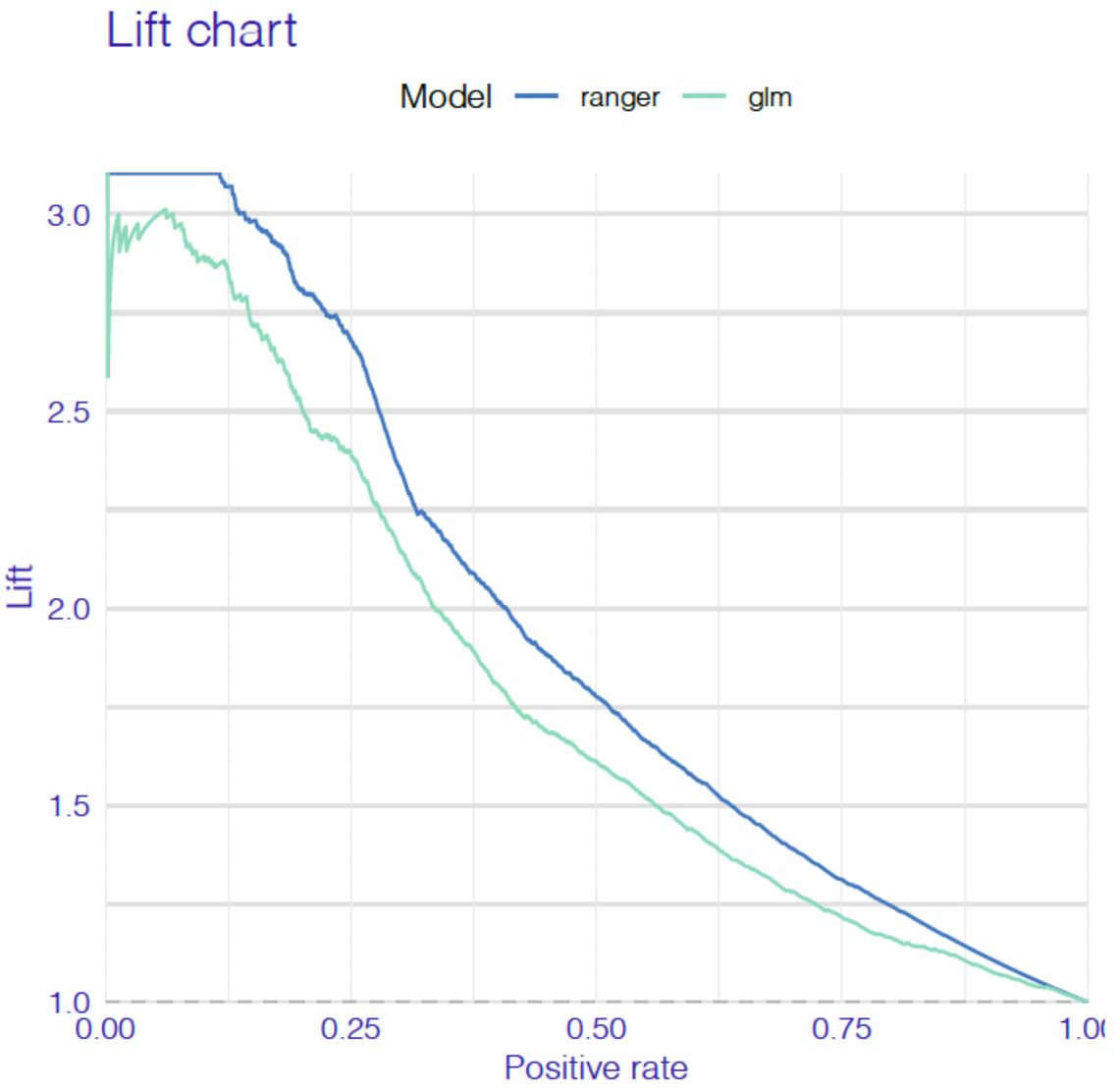
```r
# Boxplots
plot(mp_ranger, mp_rms, geom = "boxplot")
```



```
# ROC curves
plot(mp_ranger, mp_rms, geom = "roc")
```



```
# LIFT curves
plot(mp_ranger, mp_rms, geom = "lift")
```

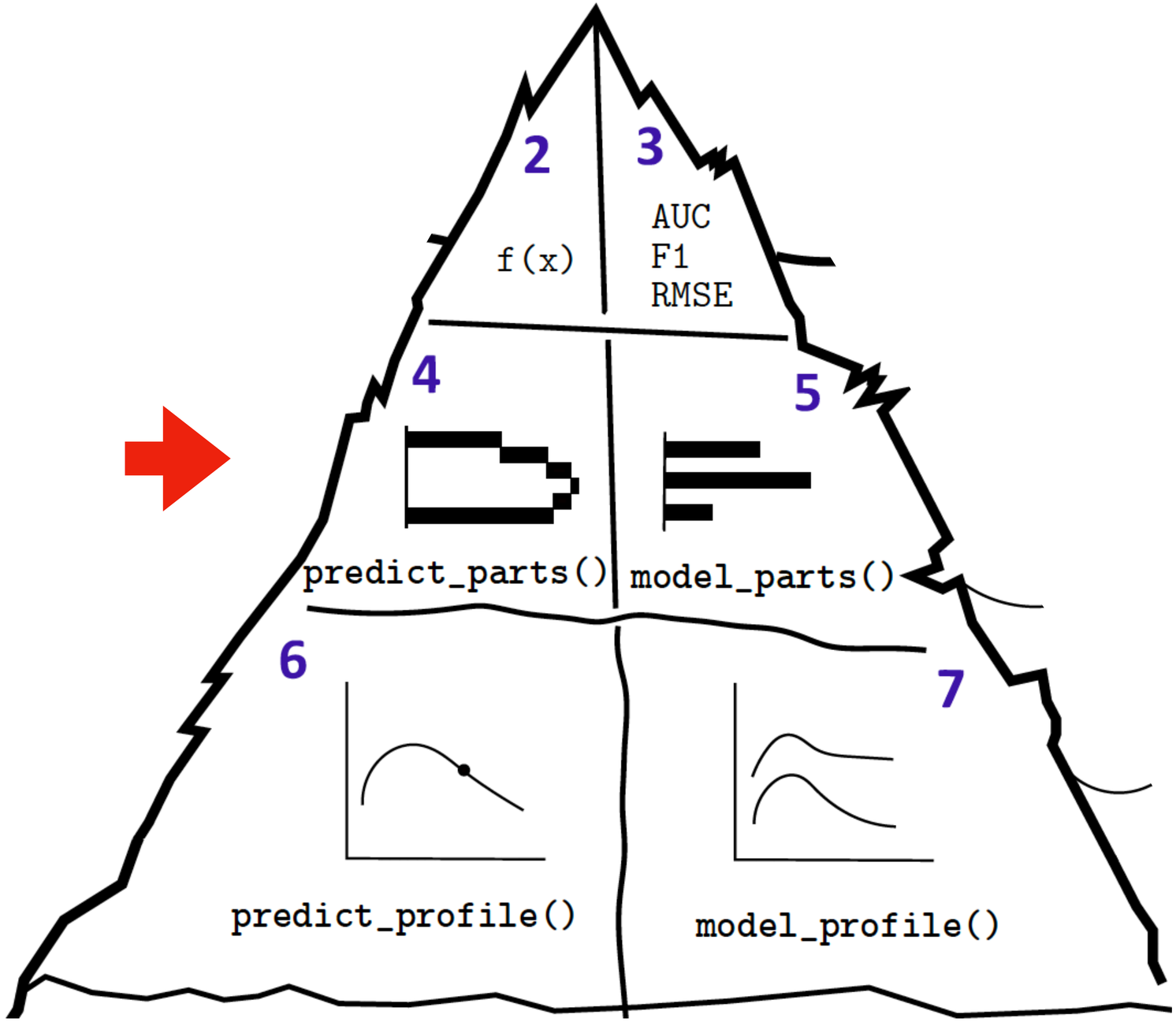


Your turn

1. Create a model_performance object
2. How good is your model?
3. Plot residuals for this model.

Parts of model predictions

Table of contents



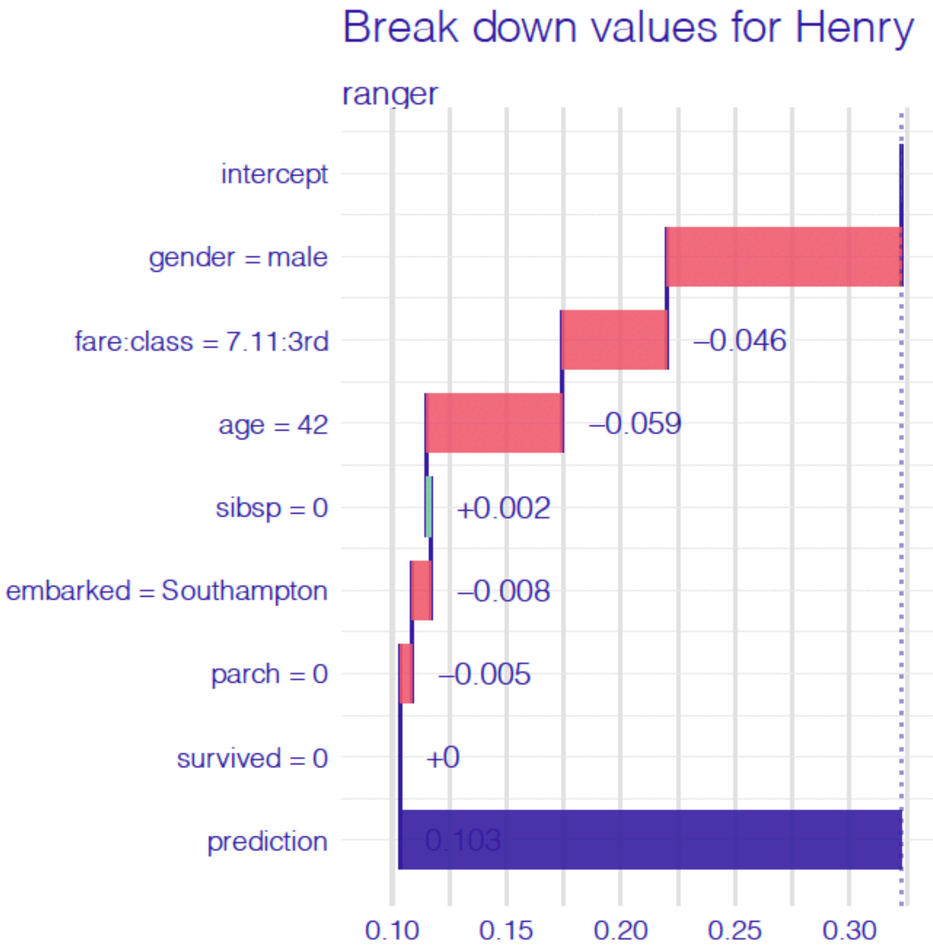
```r
henry <- titanic_imputed[1,]
predict(exp_ranger, henry)
## [1] 0.1032819
```

```
henry <- titanic_imputed[1,]
predict(exp_ranger, henry)
## [1] 0.1032819
```

```
bd_ranger <- predict_parts(exp_ranger, henry, type = "break_down_interactions")
bd_ranger
##                                 contribution
## ranger: intercept                      0.322
## ranger: gender = male                 -0.102
## ranger: fare:class = 7.11:3rd         -0.046
## ranger: age = 42                      -0.059
## ranger: sibsp = 0                      0.002
## ranger: embarked = Southampton        -0.008
## ranger: parch = 0                     -0.005
## ranger: survived = 0                   0.000
## ranger: prediction                     0.103

plot(bd_ranger)
```

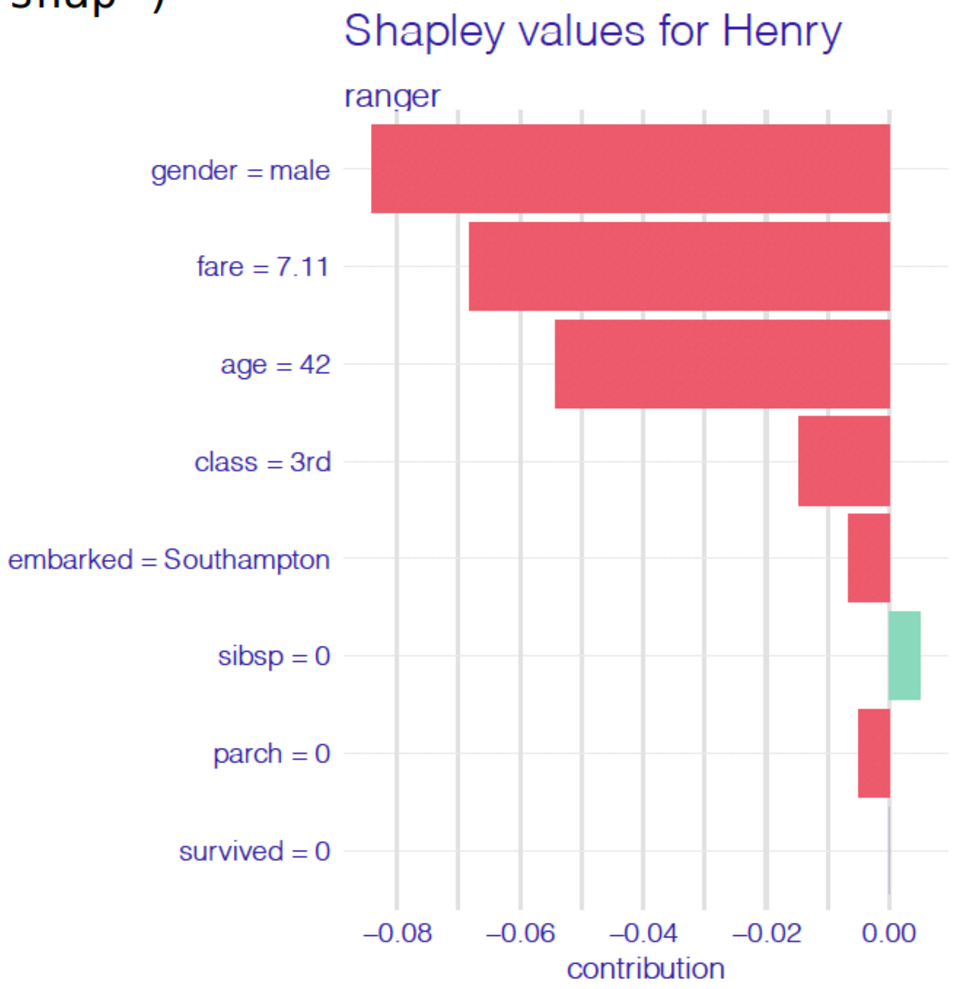
```
henry <- titanic_imputed[1,]
predict(exp_ranger, henry)
## [1] 0.1032819
```

```
bd_ranger <- predict_parts(exp_ranger, henry, type = "break_down_interactions")
bd_ranger
##                                    contribution
## ranger: intercept                         0.322
## ranger: gender = male                    -0.102
## ranger: fare:class = 7.11:3rd            -0.046
## ranger: age = 42                         -0.059
## ranger: sibsp = 0                         0.002
## ranger: embarked = Southampton           -0.008
## ranger: parch = 0                        -0.005
## ranger: survived = 0                      0.000
## ranger: prediction                        0.103

plot(bd_ranger)
```



```
henry <- titanic_imputed[1,]
predict(exp_ranger, henry)
## [1] 0.1032819
```

```
sh_ranger <- predict_parts(exp_ranger, henry, type = "shap")
plot(sh_ranger, show_boxplots = FALSE)
```

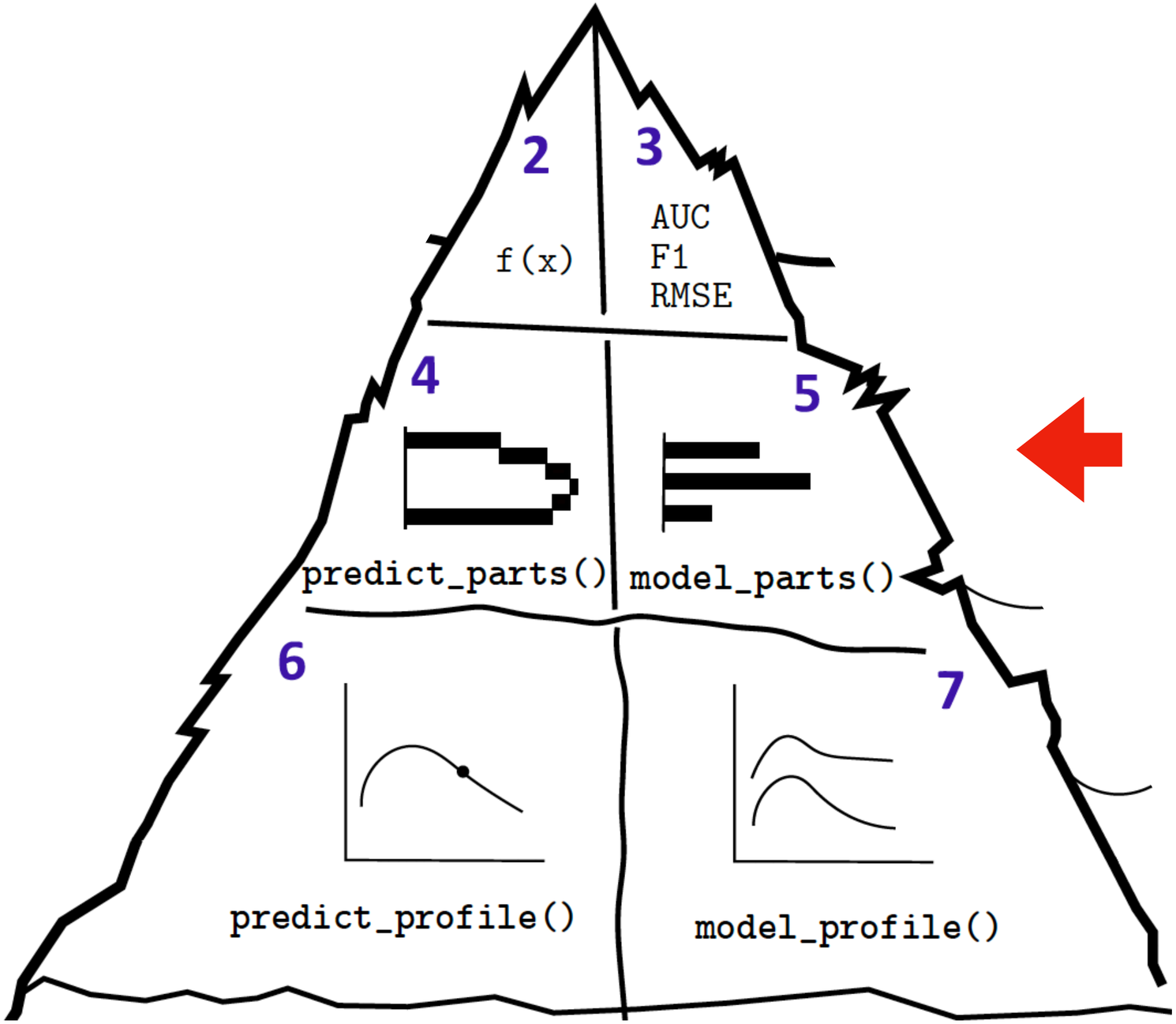


Your turn

1. Select a single instance (e.g. first row)
2. Calculate model prediction for this row
3. Calculate shaply and break-down contributions

Importance of parts of the model

Table of contents



$$V(f,i) = L^{i}_{perm}(f) - L_{org}(f)$$

$$V(f,i) = L^{i}_{perm}(f) - L_{org}(f)$$

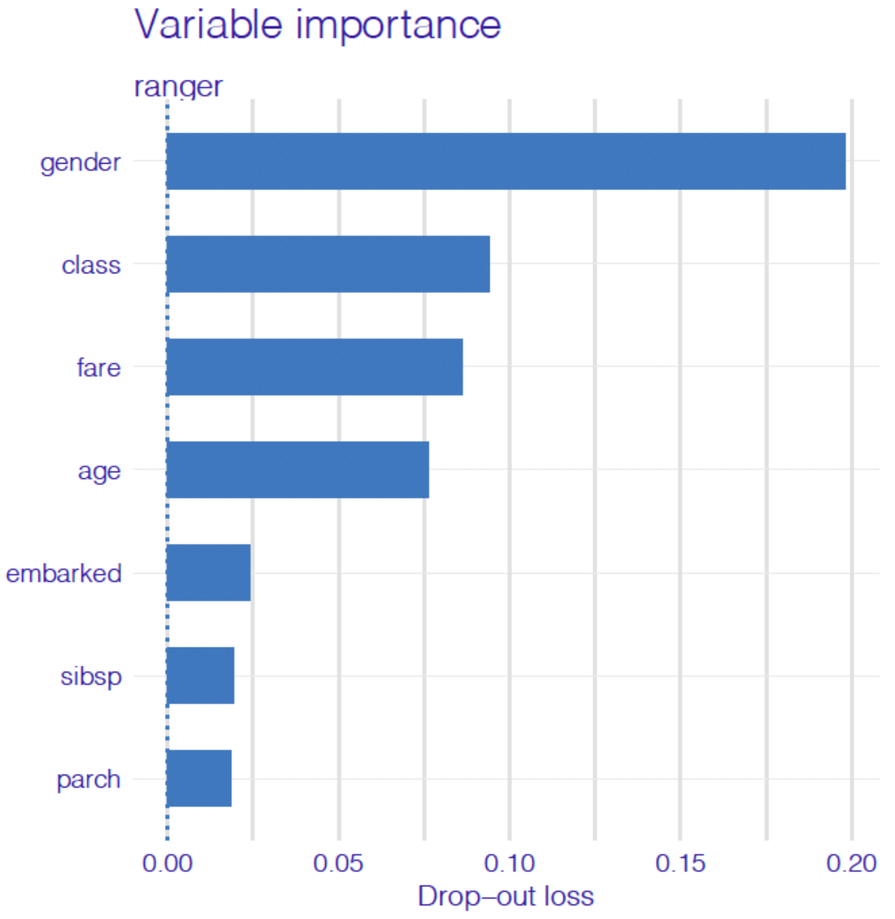
```
mp_ranger <- model_parts(exp_ranger, type = "difference")
```

$$V(f, i) = L^{i}_{perm}(f) - L_{org}(f)$$

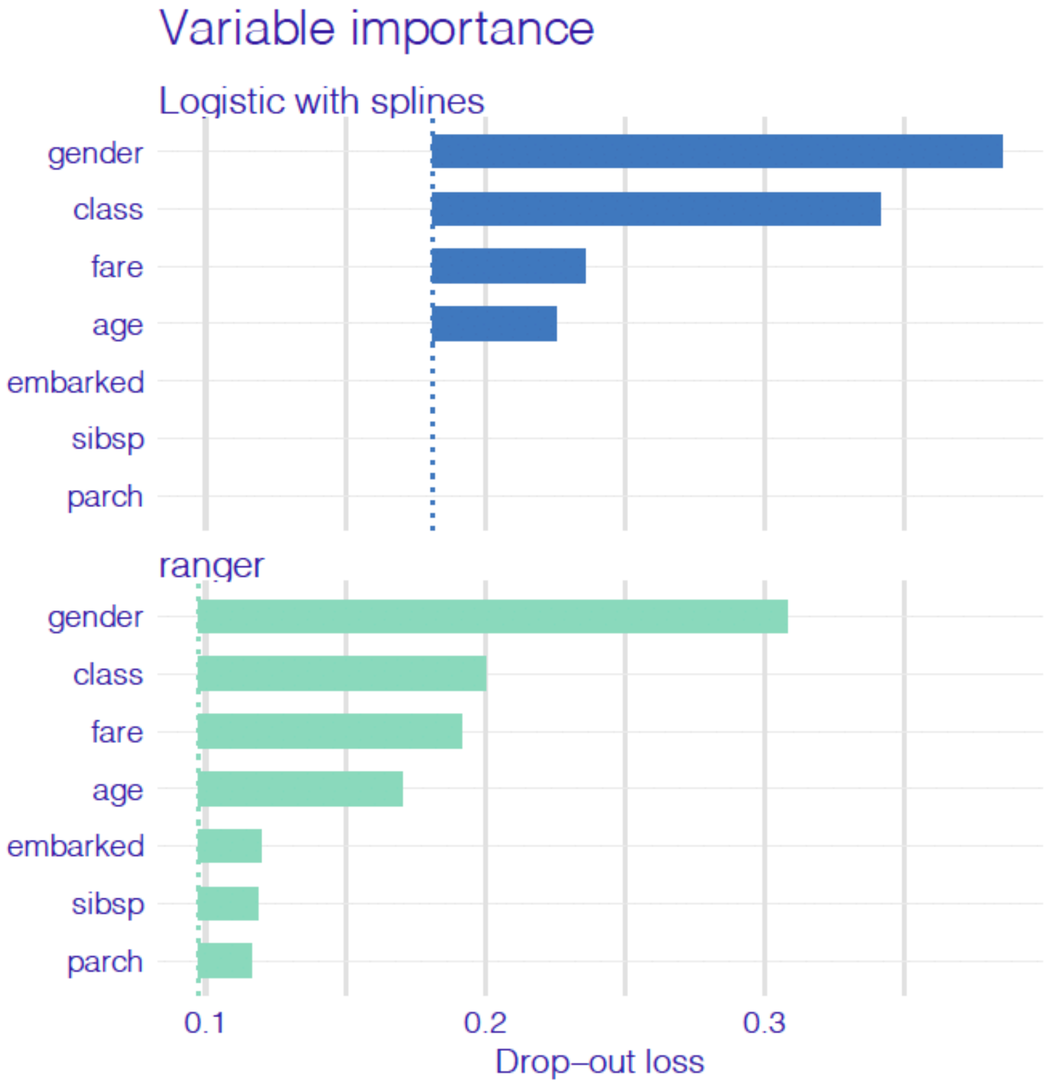
```
mp_ranger <- model_parts(exp_ranger, type = "difference")
mp_ranger
##        variable mean_dropout_loss  label
## 1 _full_model_         0.00000000 ranger
## 2         parch         0.01751121 ranger
## 3         sibsp         0.02057990 ranger
## 4      embarked         0.02507604 ranger
## 5           age         0.07177812 ranger
## 6          fare         0.09148897 ranger
## 7         class         0.09509643 ranger
## 8        gender         0.20906310 ranger
## 9   _baseline_         0.39712142 ranger
```

$$V(f,i) = L^{i}_{perm}(f) - L_{org}(f)$$

```
mp_ranger <- model_parts(exp_ranger, type = "difference")
mp_ranger
##        variable mean_dropout_loss  label
## 1 _full_model_         0.00000000 ranger
## 2         parch         0.01751121 ranger
## 3         sibsp         0.02057990 ranger
## 4      embarked         0.02507604 ranger
## 5           age         0.07177812 ranger
## 6          fare         0.09148897 ranger
## 7         class         0.09509643 ranger
## 8        gender         0.20906310 ranger
## 9   _baseline_         0.39712142 ranger

plot(mp_ranger, show_boxplots = FALSE)
```



```
mp_ranger <- model_parts(exp_ranger)
mp_rms <- model_parts(exp_rms)
plot(mp_ranger, mp_rms, show_boxplots = FALSE)
```

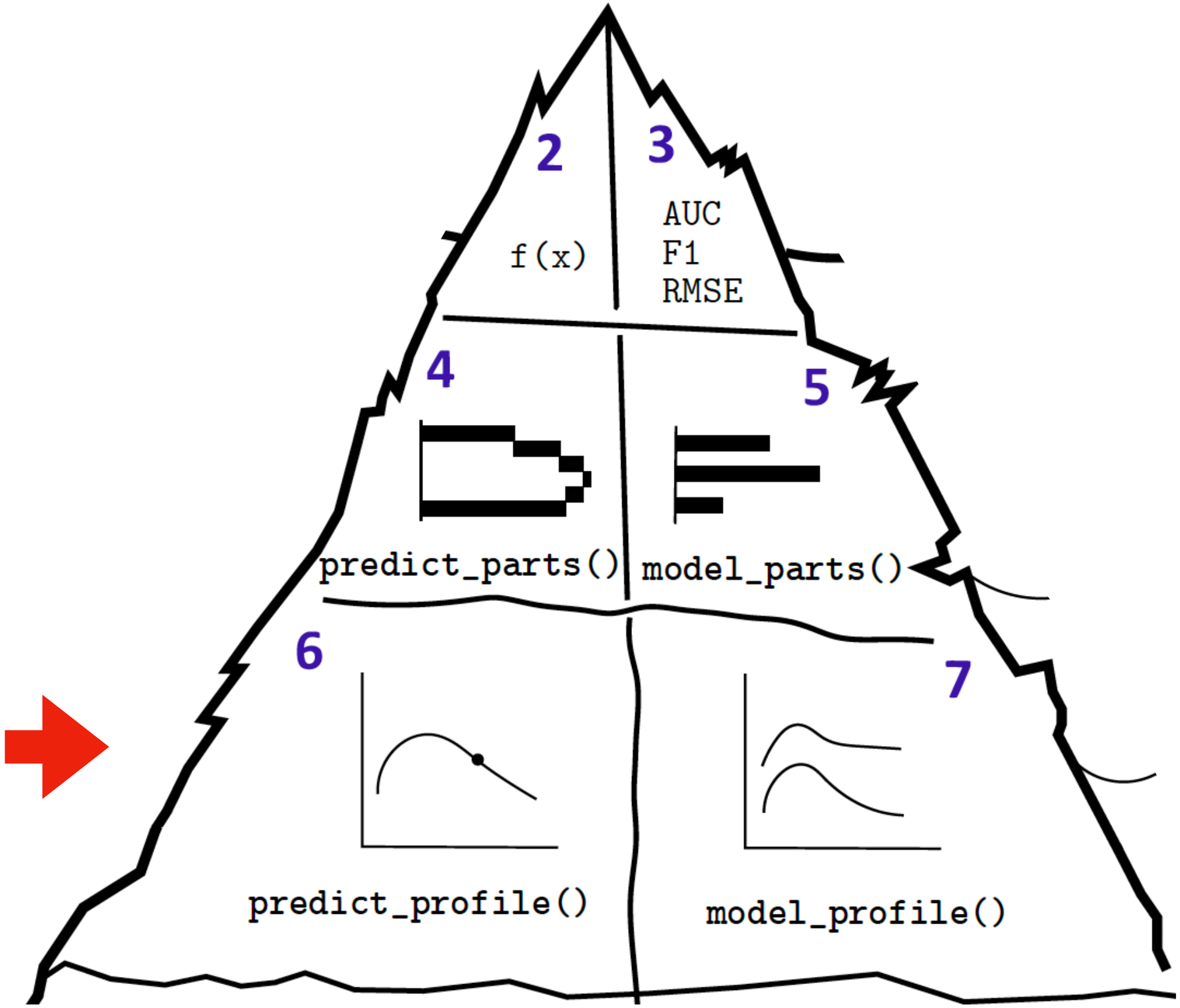


Your turn

1. Calculate variable importance for your model
2. Plot importance scores
3. If you have time, train another model and compare v-importance

Profile for a single prediction

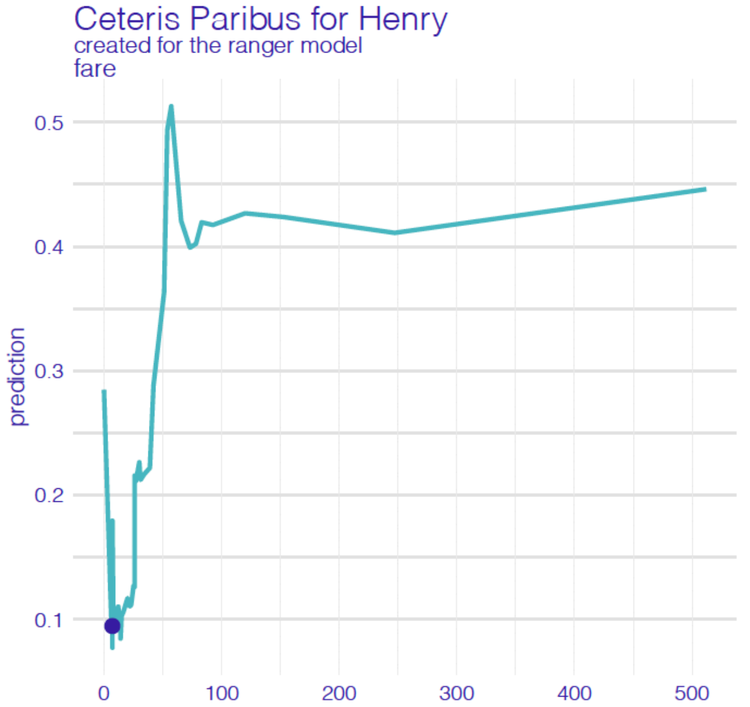
Table of contents

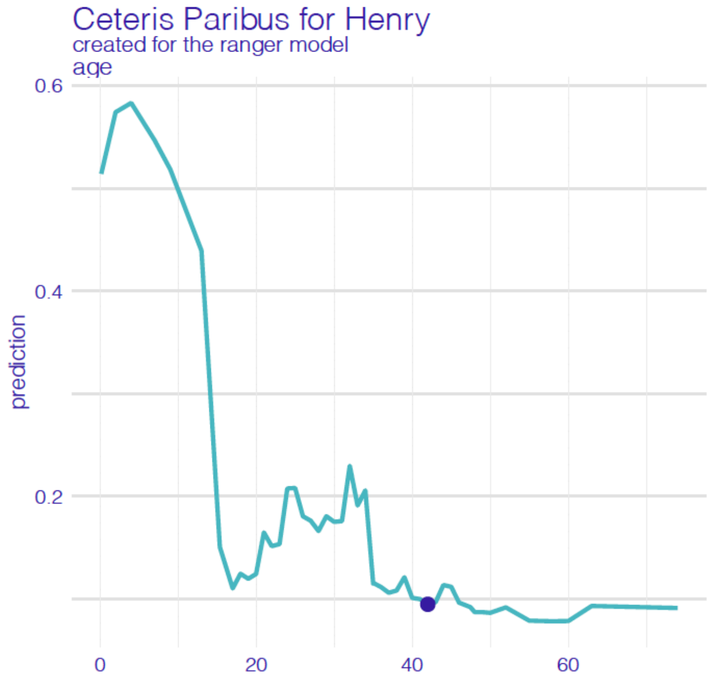


```
cp_ranger <- predict_profile(exp_ranger, henry)
cp_ranger
## Top profiles    :
##       gender        age class    embarked fare sibsp parch     _yhat_
## 1     female 42.0000000   3rd Southampton 7.11     0     0 0.42139892
## 1.1     male 42.0000000   3rd Southampton 7.11     0     0 0.09811404
## 11      male  0.1666667   3rd Southampton 7.11     0     0 0.51780598
##       _vname_ _ids_ _label_
## 1      gender     1  ranger
## 1.1    gender     1  ranger
## 11        age     1  ranger
```

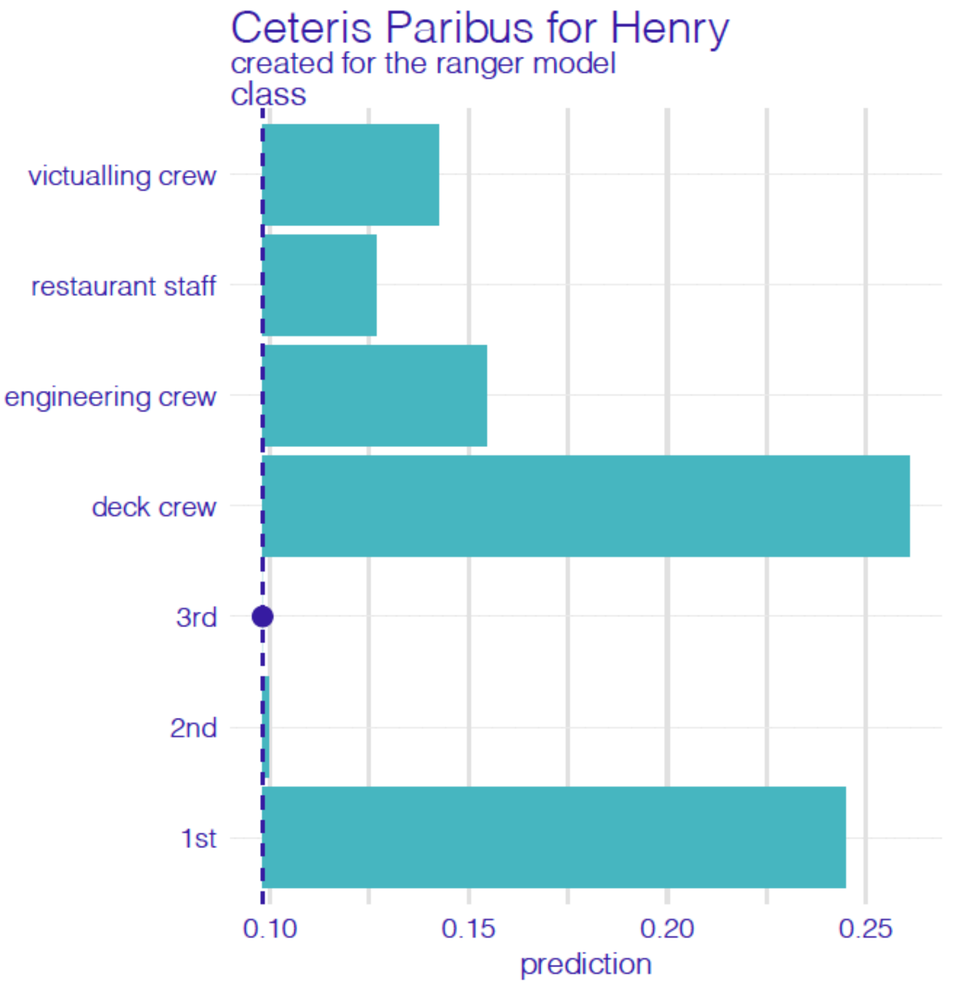
```
cp_ranger <- predict_profile(exp_ranger, henry)
cp_ranger
## Top profiles    :
##       gender        age class    embarked fare sibsp parch
## 1     female 42.0000000   3rd Southampton 7.11     0     0 0.4
## 1.1     male 42.0000000   3rd Southampton 7.11     0     0 0.0
## 11      male  0.1666667   3rd Southampton 7.11     0     0 0.5
##       _vname_ _ids_ _label_
## 1      gender     1  ranger
## 1.1    gender     1  ranger
## 11        age     1  ranger
```

```
plot(cp_ranger, variables = c("age", "fare"))
```

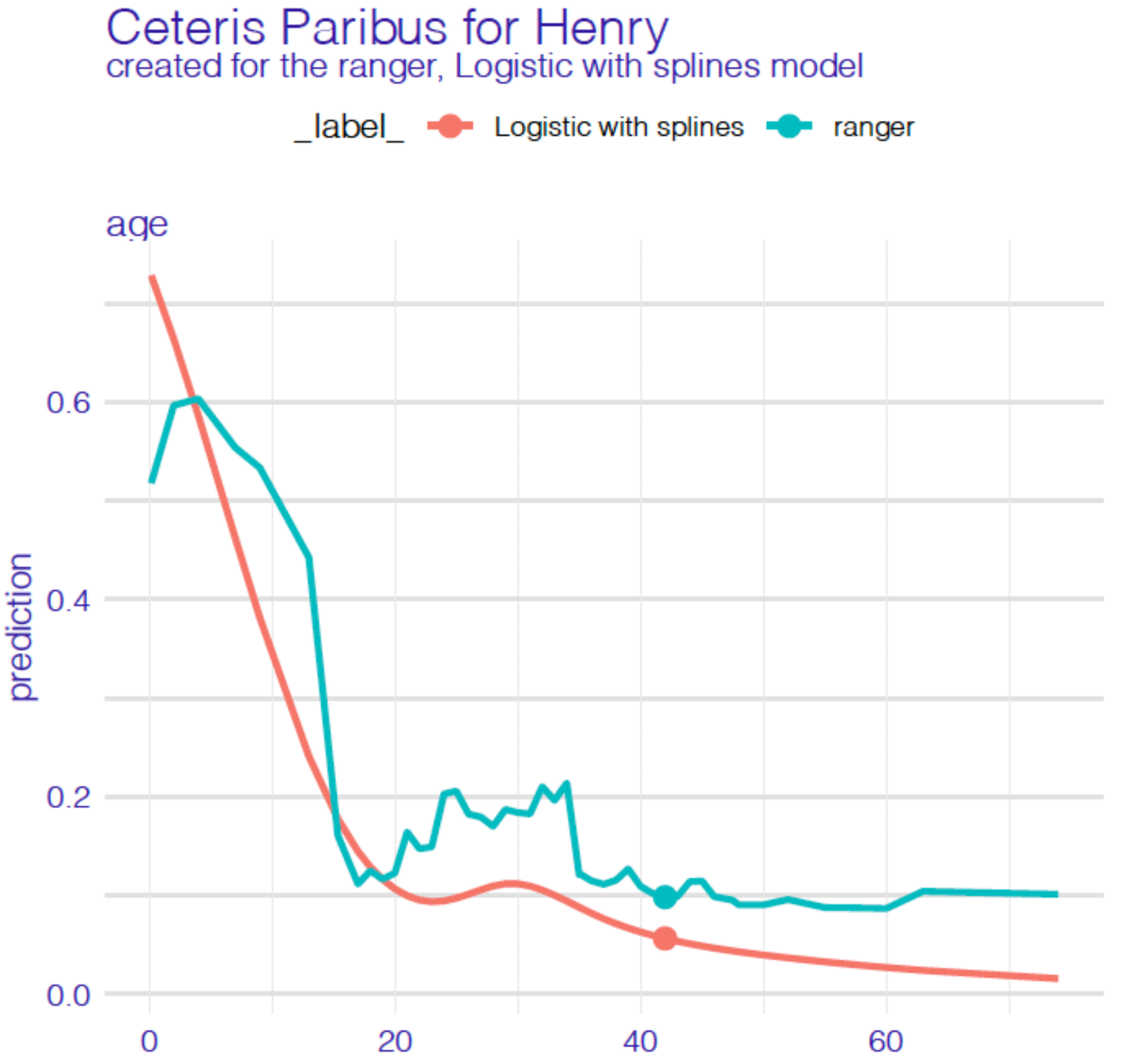




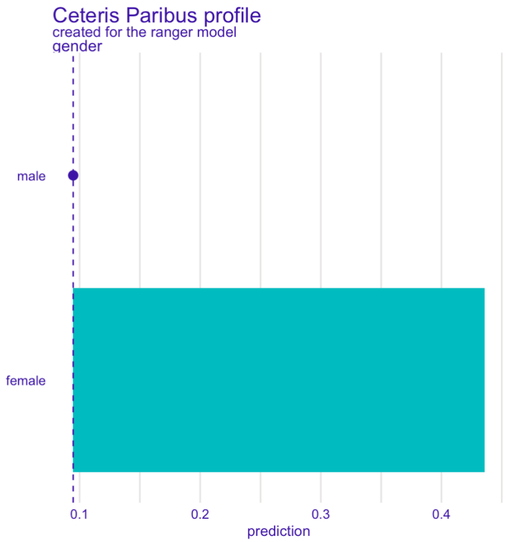
```
plot(cp_ranger, variables = "class", categorical_type = "bars")
```

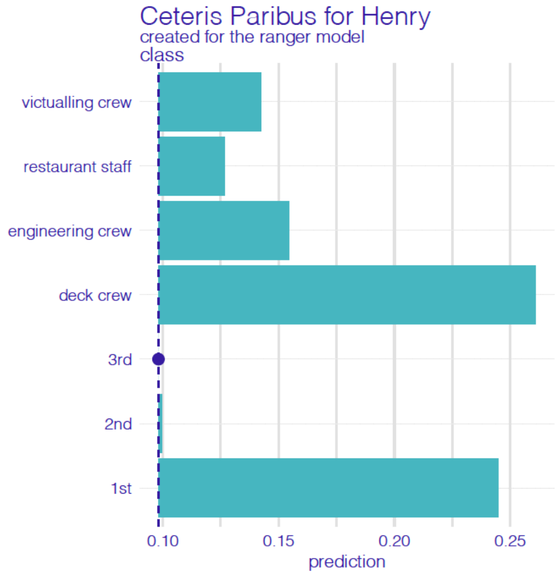


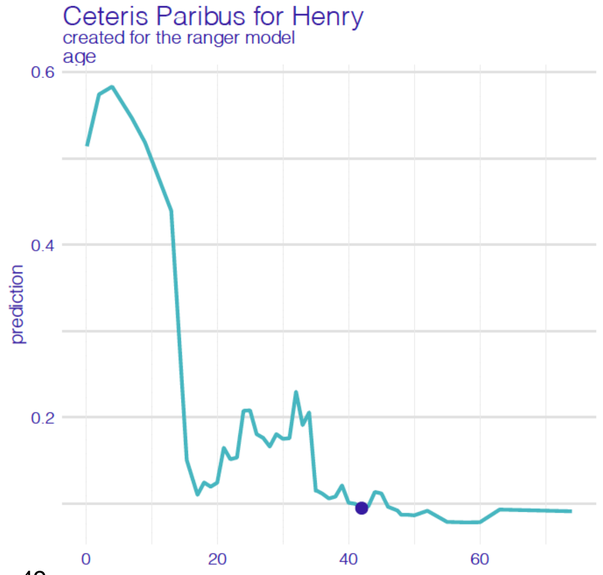
```
cp_rms <- predict_profile(exp_rms, henry)
plot(cp_ranger, cp_rms, variables = "age", color = "_label_")
```

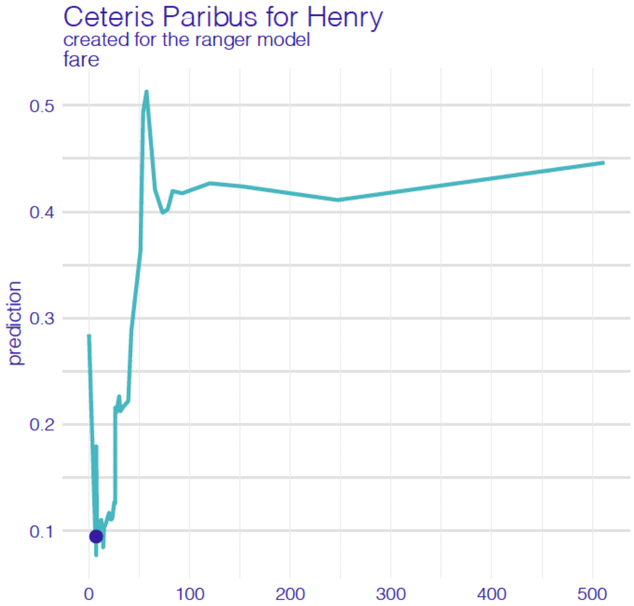


```
predict_parts(exp_rms, henry, type = "oscillations")
##     _vname_ _ids_ oscillations
## 1    gender     1   0.28444819
## 3     class     1   0.13305595
## 2       age     1   0.08664857
## 4      fare     1   0.02131783
```







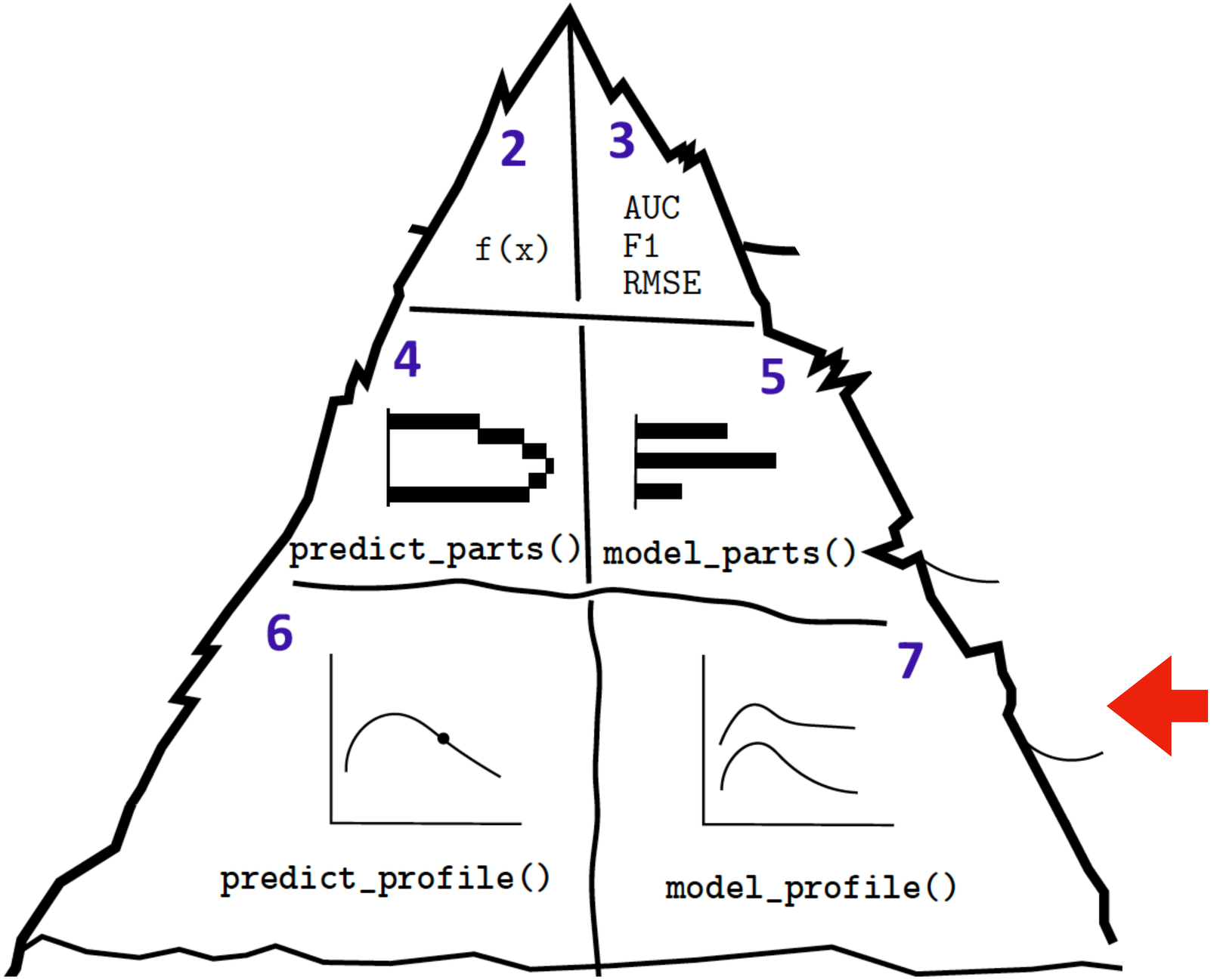


Your turn

1. Calculate profiles for selected variables for your model
2. Plot profiles
3. Which variables are the most important?

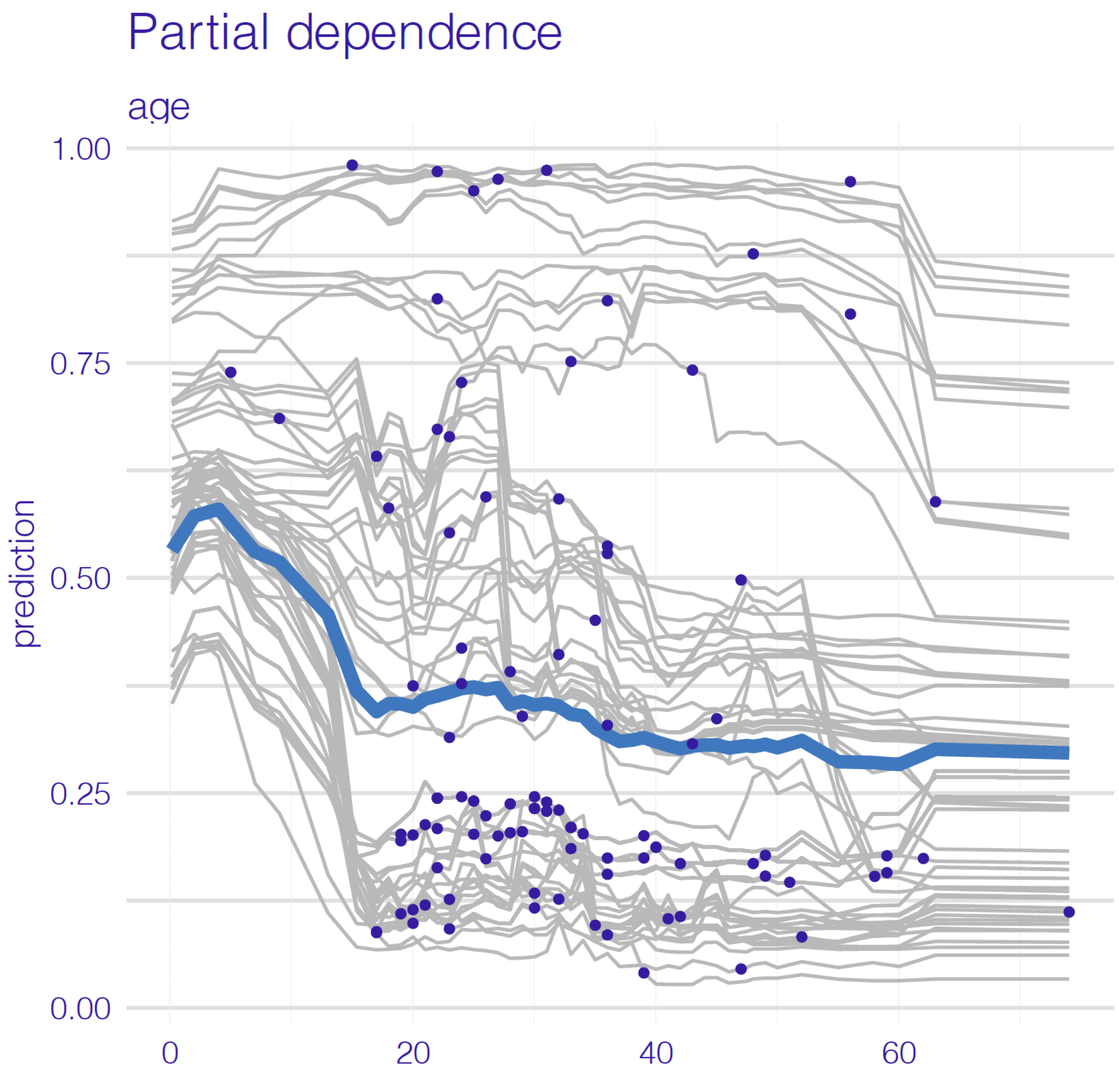
Profile for a model

Table of contents



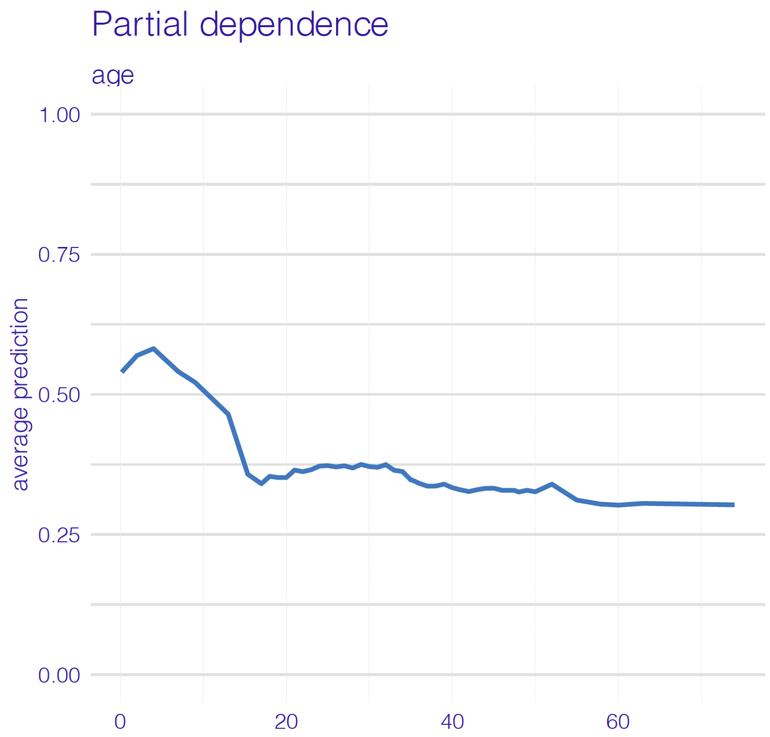
Partial dependence profiles

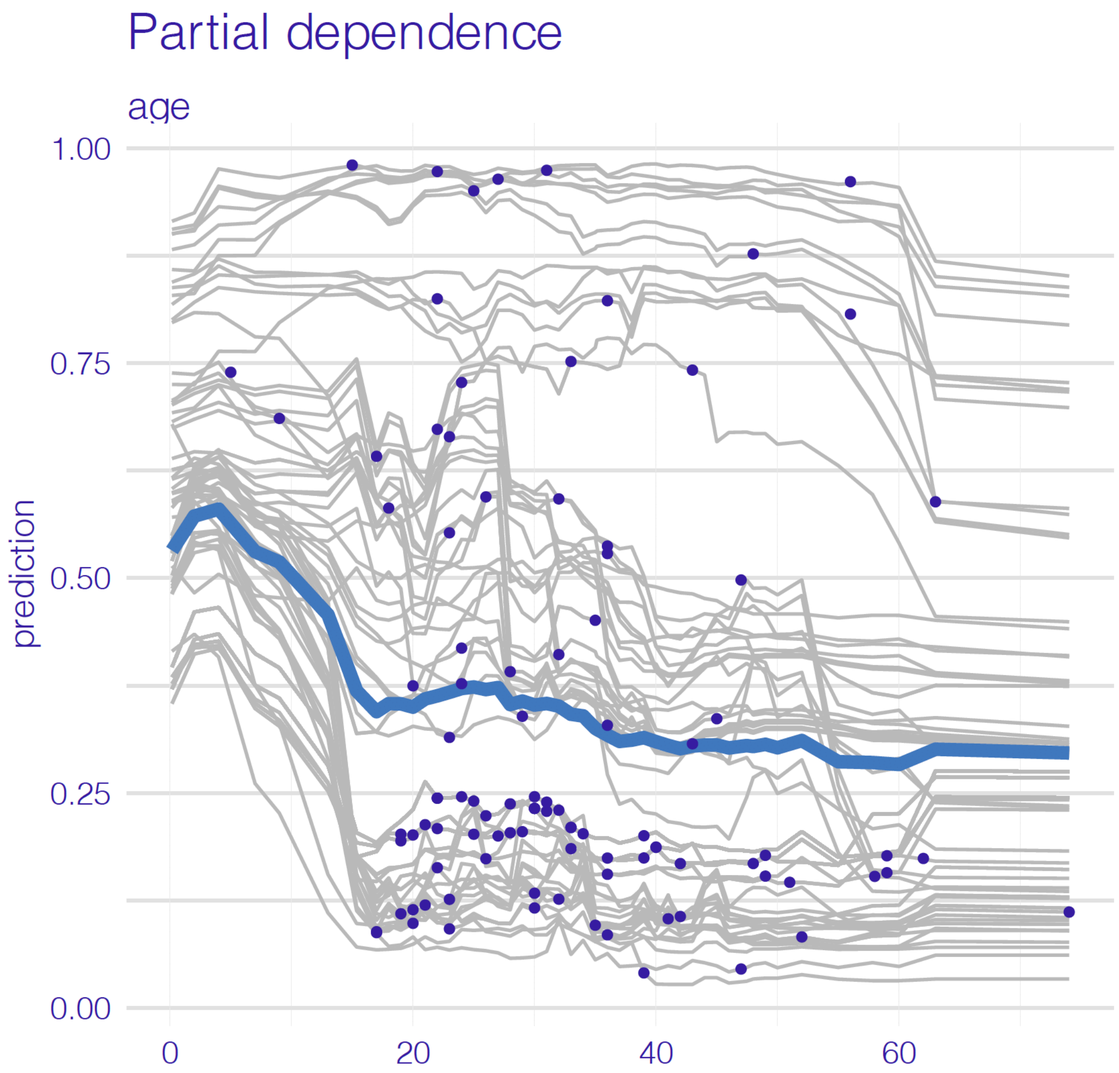
```
mp_ranger <- model_profile(exp_ranger)
```



Partial dependence profiles

```
mp_ranger <- model_profile(exp_ranger)
plot(mp_ranger, variables = "age")
```

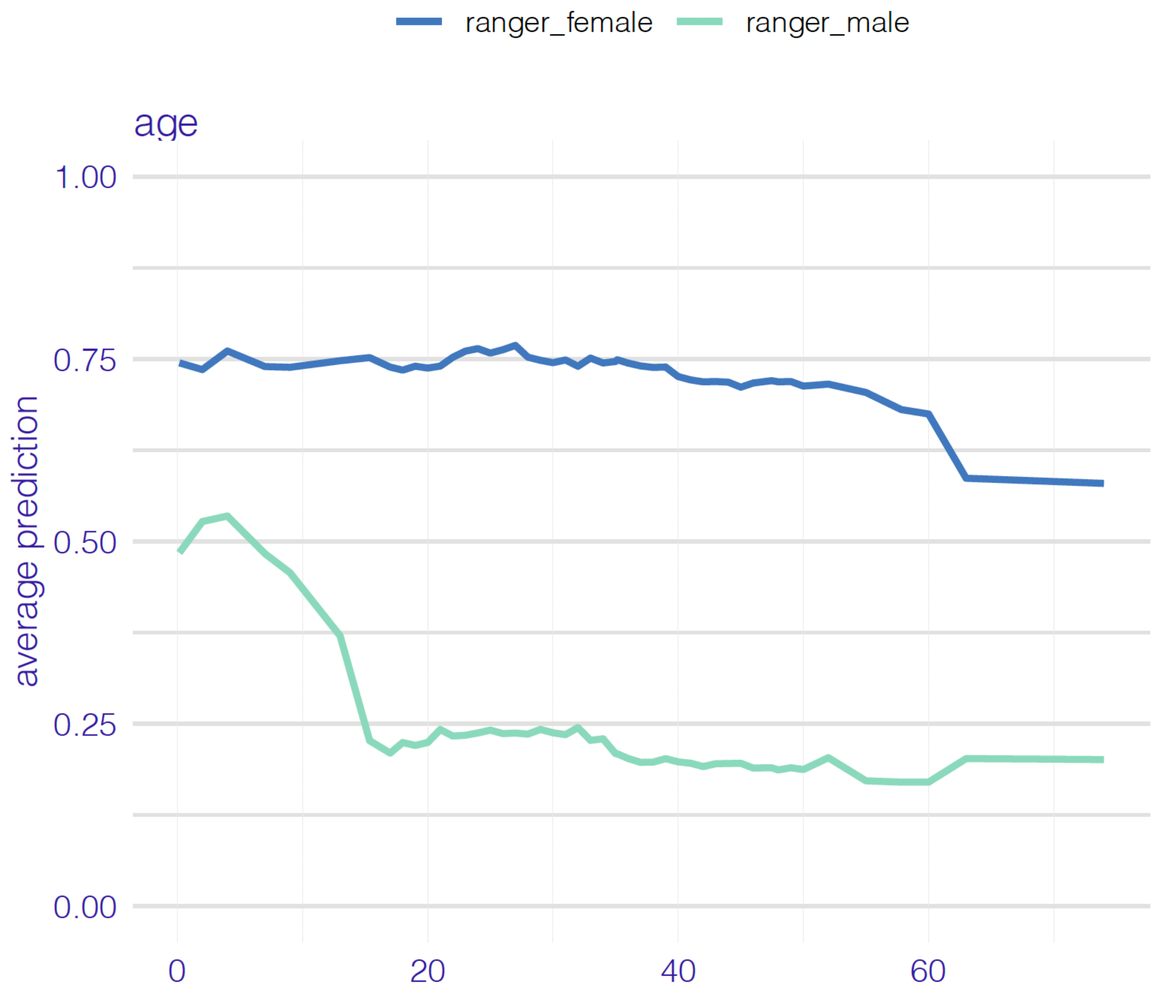




Grouped partial dependence profiles

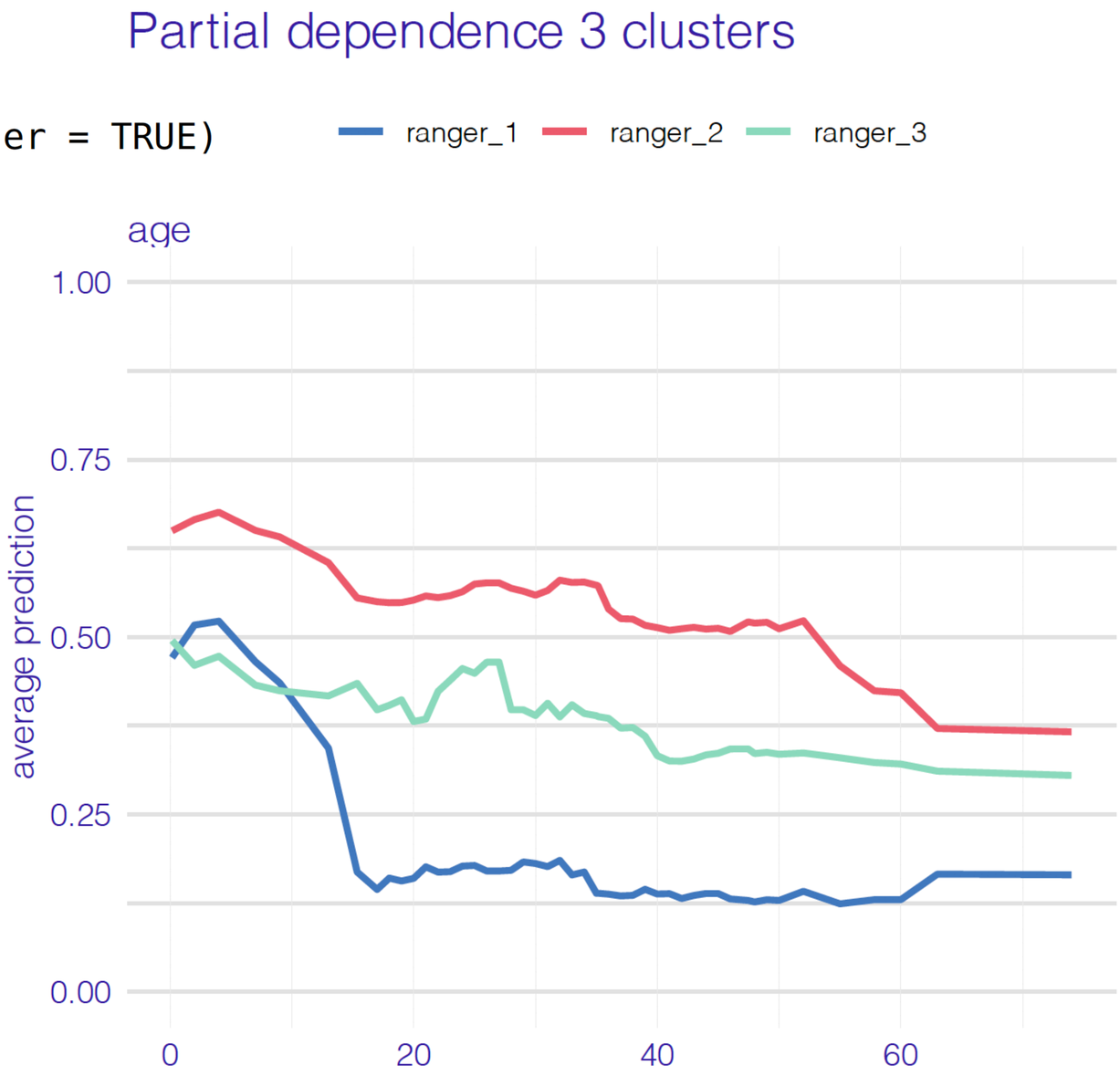
```
mp_ranger <- model_profile(exp_ranger, groups = "gender")
plot(mp_ranger, variables = "age")
```

Partial dependence groups per gender



Clustered partial dependence profiles

```
mp_ranger <- model_profile(exp_ranger, k = 3, center = TRUE)
plot(mp_ranger, variables = "age")
```



Your turn

1. Calculate PD profile for selected variables for your model
2. Plot profiles

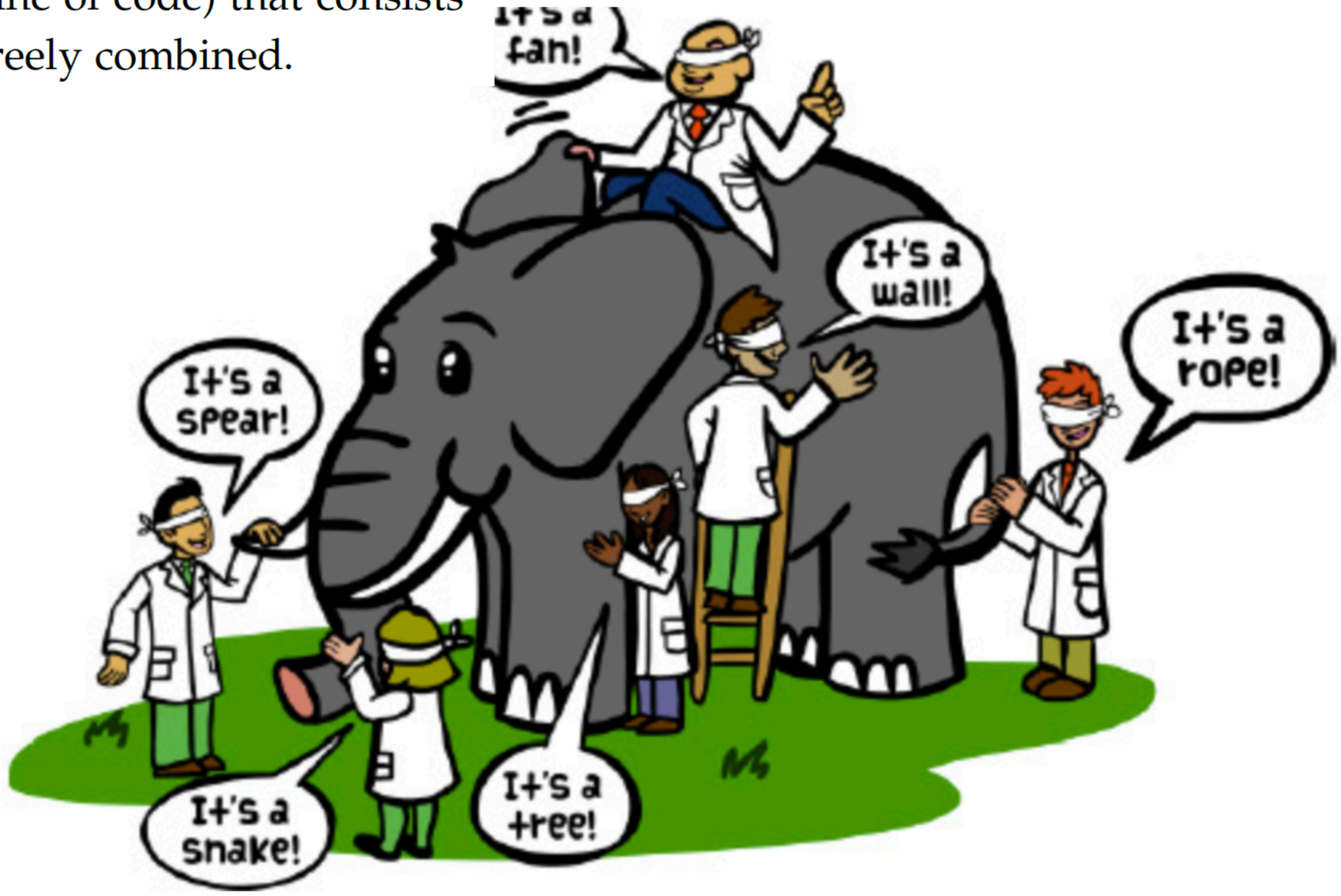
What’s next?

DALEXtra is a package with additional functionalities facilitating work with models created in python (scikitlearn, keras), H2O, mlr, mlr3 or other popular frameworks. All functions described above will also work for models created in other libraries or programming languages, provided that these models are properly wrapped.

```
aps_rf_model4 <- randomForest(m2.price ~., data = apartments)
aps_rf_explainer4 <- explain(aps_rf_model4, data = apartments, label = "model_rf")
#> Preparation of a new explainer is initiated
#>   -> model label       :  model_rf
#>   -> data              :  1000  rows  6  cols
#>   -> target variable   :  not specified! (  WARNING  )
#>   -> predict function  :  yhat.randomForest  will be used (  default  )
#>   -> predicted values  :  numerical, min =  1969.945 , mean =  3487.686 , max =
#>   -> residual function :  difference between y and yhat (  default  )
#>   A new explainer has been created!
```

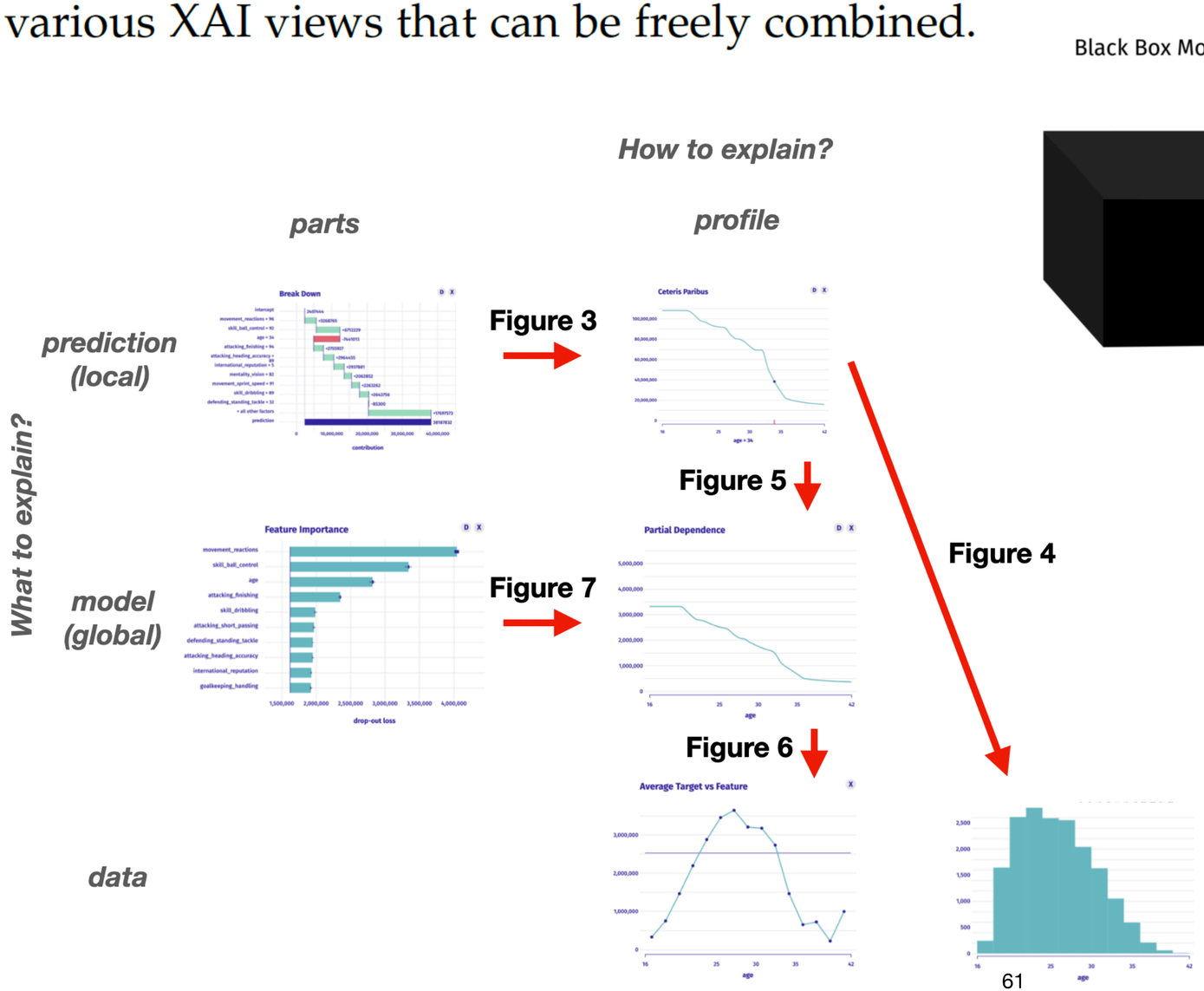
```
# model creation in scikit-learn
from pandas import DataFrame, read_csv
import pandas as pd
import pickle
import sklearn.ensemble
model = sklearn.ensemble.GradientBoostingClassifier()
model = model.fit(titanic_train_X, titanic_train_Y)
pickle.dump(model, open("gbm.pkl", "wb"), protocol = 2)

# explanations in R
library("DALEXtra")
explainer <- explain_scikitlearn("gbm.pkl",
                yml  = "testing_environment.yml",
                data = titanic_test[,1:17],
                y    = titanic_test$survived)
#> Preparation of a new explainer is initiated
#>   -> model label       :  scikitlearn_model  (  default  )
#>   -> data              :  524  rows  17  cols
#>   -> target variable   :  524  values
#>   -> predict function  :  yhat.scikitlearn_model  will be used
#>   -> predicted values  :  numerical, min =  0.02086126 , mean =
#>   -> residual function :  difference between y and yhat (  defa
#>   -> residuals         :  numerical, min =  -0.8669431 , mean =
#>   -> model_info        :  model_info not specified, assuming ty
#>   -> model_info        :  package: Model of class: sklearn.ense
#>   A new explainer has been created!
```

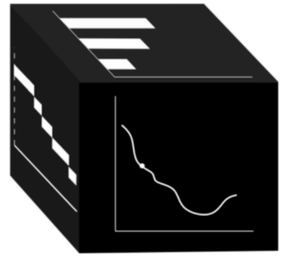
`modelStudio` and `ArenaR` are packages based on the evolving grammar of interactive model exploration. They generate a serverless HTML dashboard (one line of code) that consists of various XAI views that can be freely combined.

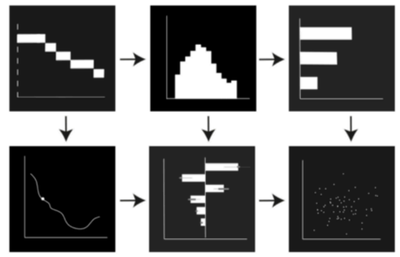


Source: http://www.theblindelephant.com/the_blind_elephant_fable.html

`modelStudio` and `ArenaR` are packages based on the evolving grammar of interactive model exploration. They generate a serverless HTML dashboard (one line of code) that consists of various XAI views that can be freely combined.







FIFA 2020

Players

FIFA 20 MAY 20, 2020

All Added Updated Free On Loan Removed Customized Create Player Calculator

Search Player ...

SETTINGS +

COLUMNS SELECTED +

BASKET

COMPARE CLEAR

CREATE SHORTLIST CREATE SQUAD

SEARCH

All Players

Continents

Poland ×

Nationality / Region

	NAME	AGE	↓OVA	POT	TEAM & CONTRACT	VALUE	WAGE	TOTAL...	HITS
	R. Lewandowski ST	30	91	91	FC Bayern München 2014 ~ 2023	€86M	€300K	2175	165
	W. Szczęsny GK	29	87	88	Juventus 2017 ~ 2024	€41M	€145K	1317	42
	A. Milik ST	25	83	86	Napoli 2016 ~ 2021	€33M	€91K	1962	104
	L. Fabiański GK	34	81	81	West Ham United 2018 ~ 2021	€8M	€70K	1256	18
	P. Zieliński CM LM	25	81	85	Napoli 2016 ~ 2021	€22.5M	€71K	2088	67
	L. Piszczek RB CB	34	80	80	Borussia Dortmund 2010 ~ 2020	€5M	€71K	2120	63

long_name	age	dob	height_cm	weight_kg	nationality	club	overall	potential	value_eur	wage_eur	player_positions	preferred_foot	in
Lionel Andrés Messi Cuccittini	32	1987-06-24	170	72	Argentina	FC Barcelona	94	94	95500000	565000	RW, CF, ST	Left	
Cristiano Ronaldo dos Santos Aveiro	34	1985-02-05	187	83	Portugal	Juventus	93	93	58500000	405000	ST, LW	Right	
Neymar da Silva Santos Junior	27	1992-02-05	175	68	Brazil	Paris Saint-Germain	92	92	105500000	290000	LW, CAM	Right	
Jan Oblak	26	1993-01-07	188	87	Slovenia	Atlético Madrid	91	93	77500000	125000	GK	Right	
Eden Hazard	28	1991-01-07	175	74	Belgium	Real Madrid	91	91	90000000	470000	LW, CF	Right	
Kevin De Bruyne	28	1991-06-28	181	70	Belgium	Manchester City	91	91	90000000	370000	CAM, CM	Right	
Marc-André ter Stegen	27	1992-04-30	187	85	Germany	FC Barcelona	90	93	67500000	250000	GK	Right	
Virgil van Dijk	27	1991-07-08	193	92	Netherlands	Liverpool	90	91	78000000	200000	CB	Right	
Luka Modrić	33	1985-09-09	172	66	Croatia	Real Madrid	90	90	45000000	340000	CM	Right	
Mohamed Salah Ghaly	27	1992-06-15	175	71	Egypt	Liverpool	90	90	80500000	240000	RW, ST	Left	
Kylian Mbappé	20	1998-12-20	178	73	France	Paris Saint-Germain	89	95	93500000	155000	ST, RW	Right	
Kalidou Koulibaly	28	1991-06-20	187	89	Senegal	Napoli	89	91	67500000	150000	CB	Right	
Harry Kane	25	1993-07-28	188	89	England	Tottenham Hotspur	89	91	83000000	220000	ST	Right	
Alisson Ramses Becker	26	1992-10-02	191	91	Brazil	Liverpool	89	91	58000000	155000	GK	Right	
David De Gea Quintana	28	1990-11-07	192	82	Spain	Manchester United	89	90	56000000	205000	GK	Right	
N'Golo Kanté	28	1991-03-29	168	72	France	Chelsea	89	90	66000000	235000	CDM, CM	Right	
Giorgio Chiellini	34	1984-08-14	187	85	Italy	Juventus	89	89	24500000	215000	CB	Left	
Sergio Leonel Agüero del Castillo	31	1988-06-02	173	70	Argentina	Manchester City	89	89	60000000	300000	ST	Right	
Sergio Ramos García	33	1986-03-30	184	82	Spain	Real Madrid	89	89	31500000	300000	CB	Right	
Luis Alberto Suárez Díaz	32	1987-01-24	182	86	Uruguay	FC Barcelona	89	89	53000000	355000	ST	Right	
Robert Lewandowski	30	1988-08-21	184	80	Poland	FC Bayern München	89	89	64500000	235000	ST	Right	
Sergio Busquets i Burgos	30	1988-07-16	189	76	Spain	FC Barcelona	89	89	55000000	300000	CDM, CM	Right	
Antoine Griezmann	28	1991-03-21	176	73	France	FC Barcelona	89	89	69000000	370000	CF, ST, LW	Left	
Paulo Bruno Exequiel Dybala	25	1993-11-15	177	75	Argentina	Juventus	88	92	76500000	215000	CAM, RW	Left	
Paul Pogba	26	1993-03-15	191	84	France	Manchester United	88	91	72500000	250000	CM, CDM	Right	
Ederson Santana de Moraes	25	1993-08-17	188	86	Brazil	Manchester City	88	91	54500000	185000	GK	Left	

long_name	age	dob	height_cm	weight_kg	nationality	club	overall	potential	value_eur	wage_eur	player_positions	preferred_foot	int
Lionel Andrés Messi Cuccittini	32	1987-06-24	170	72	Argentina	FC Barcelona	94	94	95500000	565000	RW, CF, ST	Left	
Cristiano Ronaldo dos Santos Aveiro	34	1985-02-05	187	83	Portugal	Juventus	93	93	58500000	405000	ST, LW	Right	
Neymar da Silva Santos Junior	27	1992-02-05	175	68	Brazil	Paris Saint-Germain	92	92	105500000	290000	LW, CAM	Right	
Jan Oblak	26	1993-01-07	188	87	Slovenia	Atlético Madrid	91	93	77500000	125000	GK	Right	
Eden Hazard	28	1991-01-07	175	74	Belgium	Real Madrid	91	91	90000000	470000	LW, CF	Right	
Kevin De Bruyne	28	1991-06-28	181	70	Belgium	Manchester City	91	91	90000000	370000	CAM, CM	Right	
Marc-André ter Stegen	27	1992-04-30	187	85	Germany	FC Barcelona	90	93	67500000	250000	GK	Right	
Virgil van Dijk	27	1991-07-08	193	92	Netherlands	Liverpool	90	91	78000000	200000	CB	Right	
Luka Modrić	33	1985-09-09	172	66	Croatia	Real Madrid	90	90	45000000	340000	CM	Right	
Mohamed Salah Ghaly	27	1992-06-15	175	71	Egypt	Liverpool	90	90	80500000	240000	RW, ST	Left	
Kylian Mbappé	20	1998-12-20	178	73	France	Paris Saint-Germain	89	95	93500000	155000	ST, RW	Right	
Kalidou Koulibaly	28	1991-06-20	187	89	Senegal	Napoli	89	91	67500000	150000	CB	Right	
Harry Kane	25	1993-07-28	188	89	England	Tottenham Hotspur	89	91	83000000	220000	ST	Right	
Alisson Ramses Becker	26	1992-10-02	191	91	Brazil	Liverpool	89	91	58000000	155000	GK	Right	
David De Gea Quintana	28	1990-11-07	192	82	Spain	Manchester United	89	90	56000000	205000	GK	Right	
N'Golo Kanté	28	1991-03-29	168	72	France	Chelsea	89	90	66000000	235000	CDM, CM	Right	
Giorgio Chiellini	34	1984-08-14	187	85	Italy	Juventus	89	89	24500000	215000	CB	Left	
Sergio Leonel Agüero del Castillo	31	1988-06-02	173	70	Argentina	Manchester City	89	89	60000000	300000	ST	Right	
Sergio Ramos García	33	1986-03-30	184	82	Spain	Real Madrid	89	89	31500000	300000	CB	Right	
Luis Alberto Suárez Díaz	32	1987-01-24	182	86	Uruguay	FC Barcelona	89						
Robert Lewandowski	30	1988-08-21	184	80	Poland	FC Bayern München	89						
Sergio Busquets i Burgos	30	1988-07-16	189	76	Spain	FC Barcelona	89						
Antoine Griezmann	28	1991-03-21	176	73	France	FC Barcelona	89						
Paulo Bruno Exequiel Dybala	25	1993-11-15	177	75	Argentina	Juventus	88						
Paul Pogba	26	1993-03-15	191	84	France	Manchester United	88						
Ederson Santana de Moraes	25	1993-08-17	188	86	Brazil	Manchester City	88	91	54500000	185000	GK	Left	

```
library("gbm")
fifa_gbm <- gbm(Value~.,
        data              = fifa20,
        n.trees           = 250,
        interaction.depth = 3)
```

wikipedia

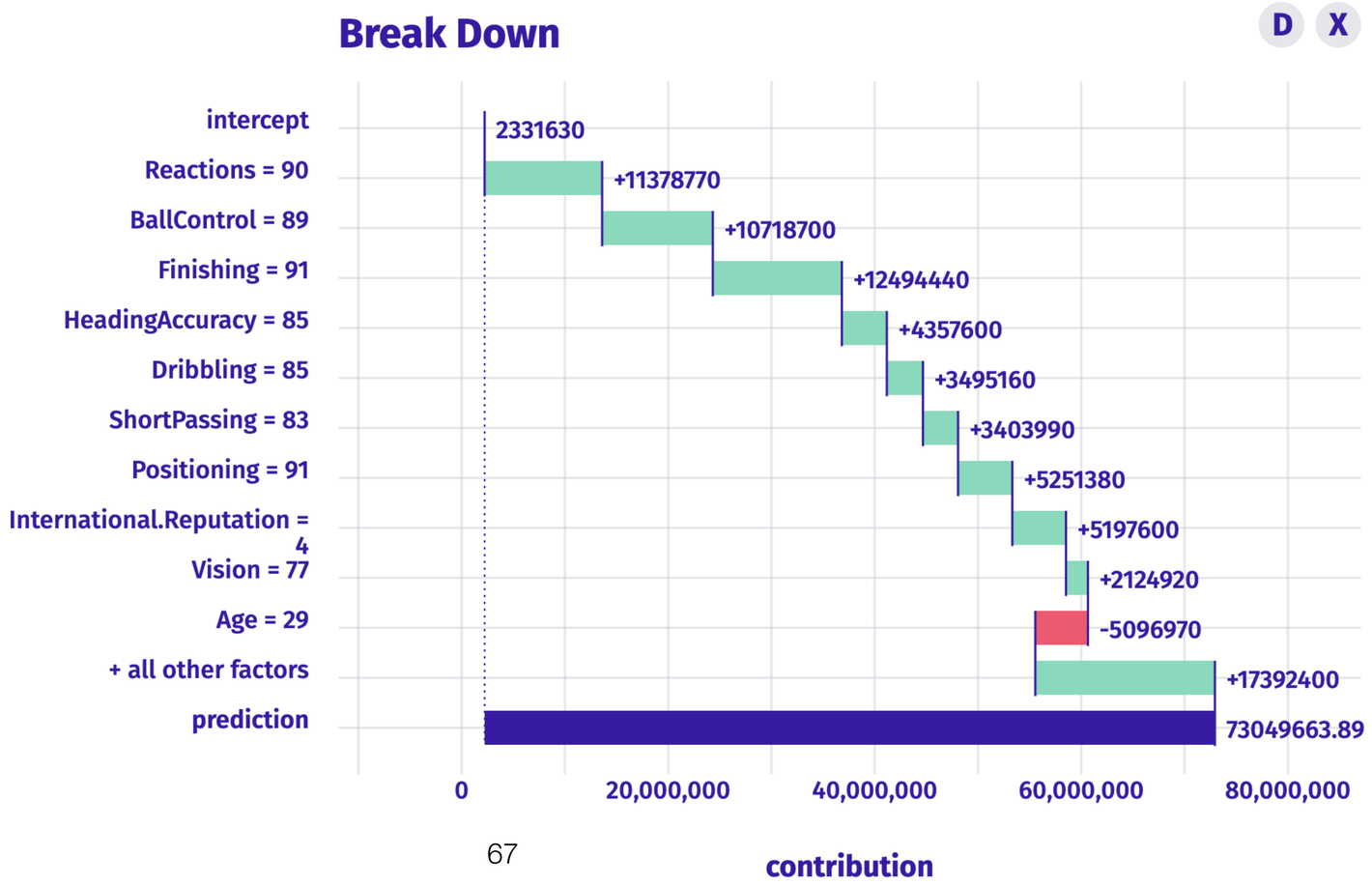
Full name	Robert Lewandowski
Date of birth	21 August 1988
Place of birth	Warsaw, Poland
Playing position	Striker

Prediction from GBM model: 73 049 663 EUR

How? Why? Really?

wikipedia

Full name	Robert Lewandowski
Date of birth	21 August 1988
Place of birth	Warsaw, Poland
Playing position	Striker



Covid-19

https://en.wikipedia.org/wiki/Mortality_due_to_COVID-19

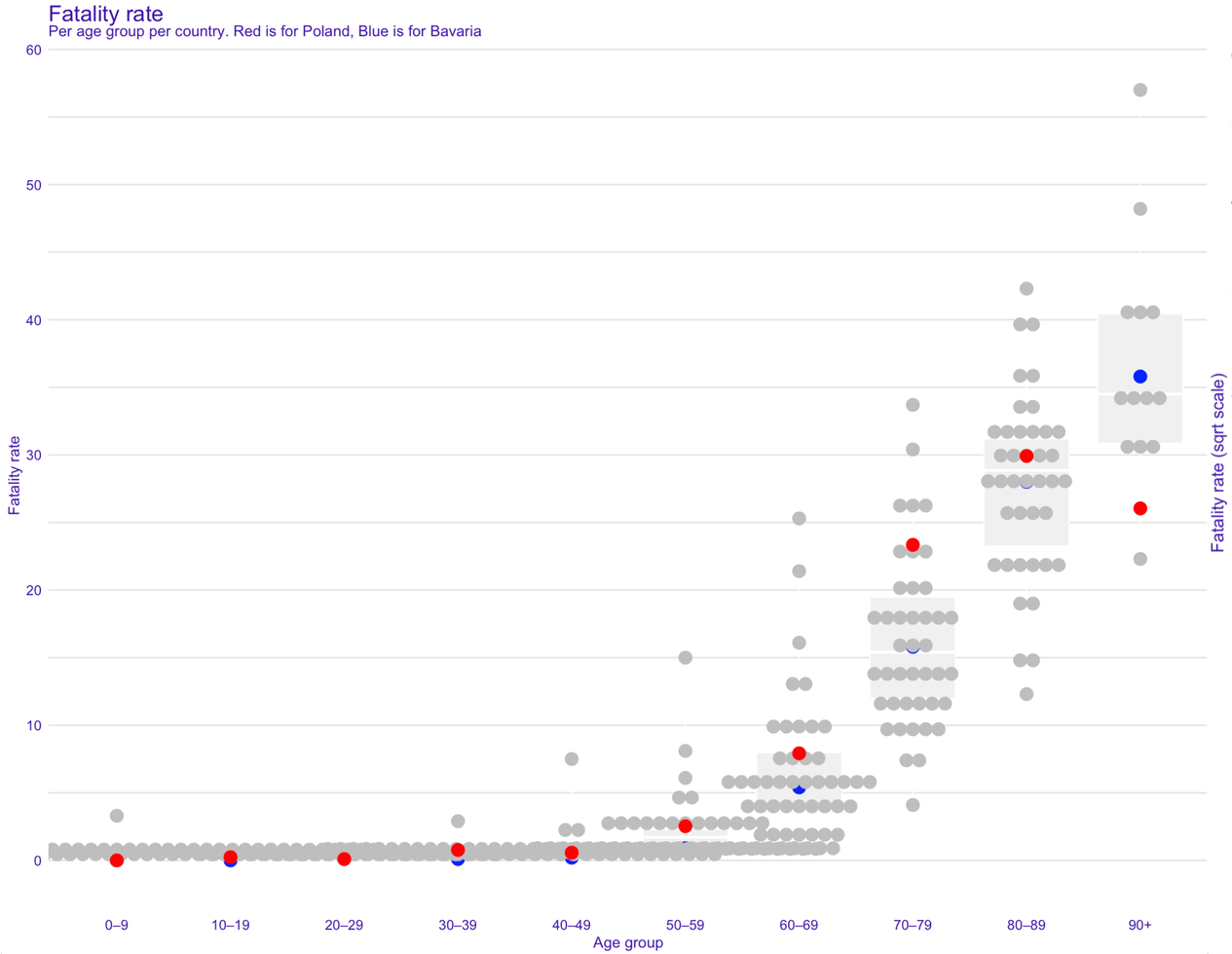
Case fatality rates (%) by age and country [hide]

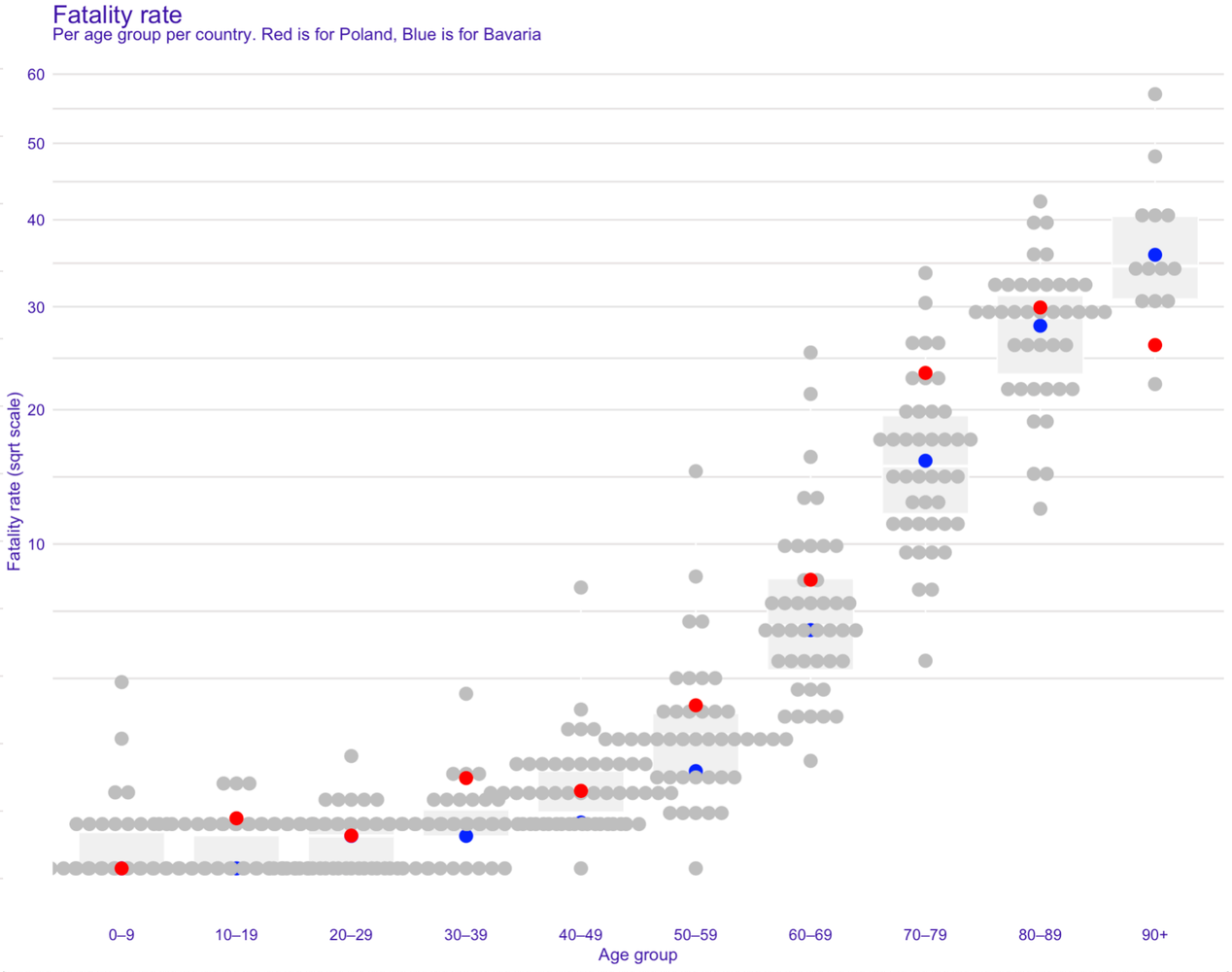
Country	Age									
	0–9	10–19	20–29	30–39	40–49	50–59	60–69	70–79	80–89	90+
Argentina as of 7 May[13]	0.0	0.0	0.1	0.4	1.3	3.6	12.9	18.8	28.4	
Australia as of 4 June[14]	0.0	0.0	0.0	0.0	0.1	0.2	1.1	4.1	18.1	40.8
Canada as of 3 June[15]	0.0		0.1		0.7		11.2		30.7	
Alberta as of 3 June[16]	0.0	0.0	0.1	0.1	0.1	0.2	1.9	11.9	30.8	
Br. Columbia as of 2 June[17]	0.0	0.0	0.0	0.0	0.5	0.8	4.6	12.3	33.8	33.6
Ontario as of 3 June[18]	0.0	0.0	0.1	0.2	0.5	1.5	5.6	17.7	26.0	33.3
Quebec as of 2 June[19]	0.0		0.1	0.1	0.2	1.1	6.1	21.4	30.4	36.1
Chile as of 31 May[20][21]	0.1				0.3	0.7	2.3	7.7	15.6	
China as of 11 February[22]	0.0	0.2	0.2	0.2	0.4	1.3	3.6	8.0	14.8	
Colombia as of 3 June[23]	0.3	0.0	0.2	0.5	1.6	3.4	9.4	18.1	25.6	35.1
Denmark as of 4 June[24]	0.2						4.1	16.5	28.1	48.2
Finland as of 4 June[25]	0.0	0.0	<0.4	<0.4	<0.5	0.8	3.8	18.1	42.3	
Germany as of 5 June[26]	0.0	0.0	0.1			1.9		19.7		31.0
Bavaria as of 5 June[27]	0.0	0.0	0.1	0.1	0.2	0.9	5.4	15.8	28.0	35.8
Israel as of 3 May[28]	0.0	0.0	0.0	0.9	0.9	3.1	9.7	22.9	30.8	31.3
Italy as of 3 June[29]	0.3	0.0	0.1	0.3	0.9	2.7	10.6	25.9	32.4	29.9
Japan as of 7 May[30]	0.0	0.0	0.0	0.1	0.3	0.6	2.5	6.8	14.8	
Mexico as of 3 June[31]	3.3	0.6	1.2	2.9	7.5	15.0	25.3	33.7	40.3	40.6
Netherlands as of 3 June[32]	0.0	0.2	0.1	0.3	0.5	1.7	8.1	25.6	33.3	34.5
Norway as of 4 June[33]	0.0	0.0	0.0	0.0	0.3	0.4	2.2	9.0	22.7	57.0
Philippines as of 4 June[34]	1.6	0.9	0.5	0.8	2.4	5.5	13.2	20.9	31.5	
Portugal as of 3 June[35]	0.0	0.0	0.0	0.0	0.3	1.3	3.6	10.5	21.2	
South Africa as of 28 May[36]	0.3	0.1	0.1	0.4	1.1	3.8	9.2	15.0	12.3	
South Korea as of 15 June[37]	0.0	0.0	0.0	0.2	0.2	0.7	2.6	10.1	25.6	
Spain as of 17 May[38]	0.2	0.3	0.2	0.3	0.6	1.4	4.9	14.3	21.0	22.3

Estimated prognosis by age and sex based on cases from France and Diamond Princess ship[60]

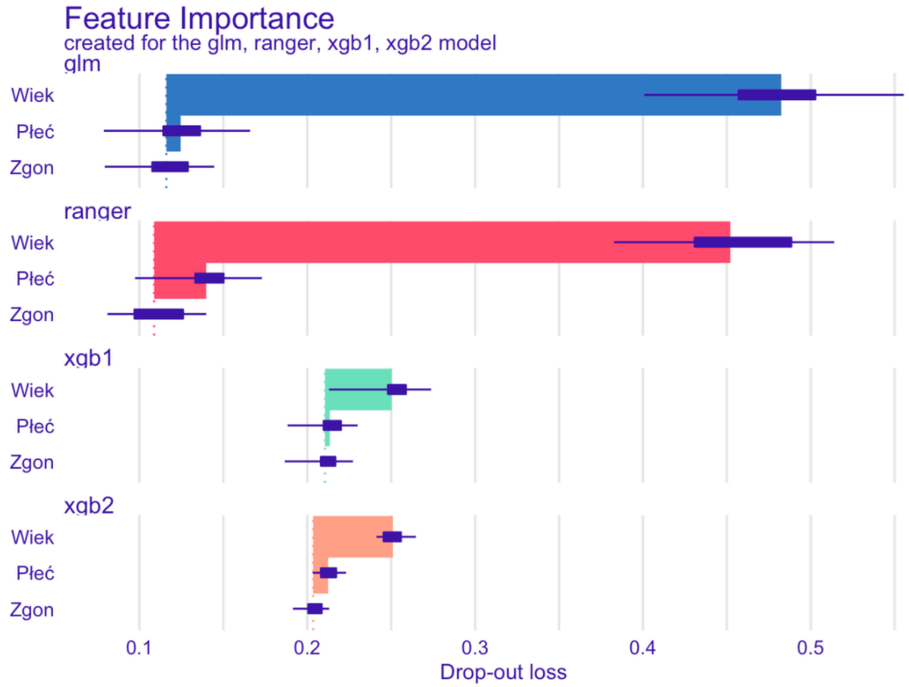
	Percent of infected people who are hospitalized								
	0–19	20–29	30–39	40–49	50–59	60–69	70–79	80+	Total
Female	0.2 (0.1–0.3)	0.6 (0.3–0.9)	1.1 (0.7–1.8)	1.6 (0.9–2.4)	3.2 (1.9–4.9)	6.2 (3.7–9.6)	9.6 (5.7–14.8)	23.6 (14.0–36.4)	3.2 (1.9–5.0)
Male	0.2 (0.1–0.3)	0.7 (0.4–1.1)	1.4 (0.9–2.2)	1.9 (1.1–3.0)	3.9 (2.3–6.1)	8.1 (4.8–12.6)	13.4 (8.0–20.7)	45.9 (27.3–70.9)	4.0 (2.4–6.2)
Total	0.2 (0.1–0.3)	0.6 (0.4–1.0)	1.3 (0.8–2.0)	1.7 (1.0–2.7)	3.5 (2.1–5.4)	7.1 (4.2–11.0)	11.3 (6.7–17.5)	32.0 (19.0–49.4)	3.6 (2.1–5.6)
	Percent of hospitalized people who go to Intensive Care Unit								
	0–19	20–29	30–39	40–49	50–59	60–69	70–79	80+	Total
Female	16.7 (14.4–19.2)	8.6 (7.5–9.9)	11.9 (10.9–13.0)	16.6 (15.6–17.7)	20.7 (19.8–21.7)	23.1 (22.2–24.0)	18.7 (18.0–19.5)	4.2 (4.0–4.5)	14.3 (13.9–14.7)
Male	26.9 (23.2–31.0)	14.0 (12.2–15.9)	19.2 (17.6–20.9)	26.9 (25.3–28.5)	33.4 (32.0–34.8)	37.3 (36.0–38.6)	30.2 (29.2–31.3)	6.8 (6.5–7.2)	23.1 (22.6–23.6)
Total	22.2 (19.2–25.5)	11.5 (10.1–13.2)	15.9 (14.6–17.3)	22.2 (21.0–23.5)	27.6 (26.5–28.7)	30.8 (29.8–31.8)	24.9 (24.1–25.8)	5.6 (5.3–5.9)	19.0 (18.7–19.44)
	Percent of hospitalized people who die								
	0–19	20–29	30–39	40–49	50–59	60–69	70–79	80+	Total
Female	0.5 (0.2–1.1)	0.9 (0.5–1.3)	1.5 (1.2–1.9)	2.6 (2.3–3.0)	5.2 (4.8–5.6)	10.1 (9.5–10.6)	16.7 (16.0–17.4)	25.2 (24.4–26.0)	14.4 (14.0–14.9)
Male	0.7 (0.3–1.5)	1.3 (0.8–1.9)	2.2 (1.7–2.7)	3.8 (3.4–4.4)	7.6 (7.0–8.2)	14.8 (14.1–15.6)	24.6 (23.7–25.6)	37.1 (36.1–38.2)	21.22 (20.8–21.7)
Total	0.6 (0.3–1.3)	1.1 (0.7–1.6)	1.9 (1.5–2.3)	3.3 (2.9–3.7)	6.5 (6.0–7.0)	12.6 (12.0–13.2)	21.0 (20.3–21.8)	31.6 (30.9–32.4)	18.1 (17.8–18.4)
	Percent of infected people who die – infection fatality rate (IFR)								

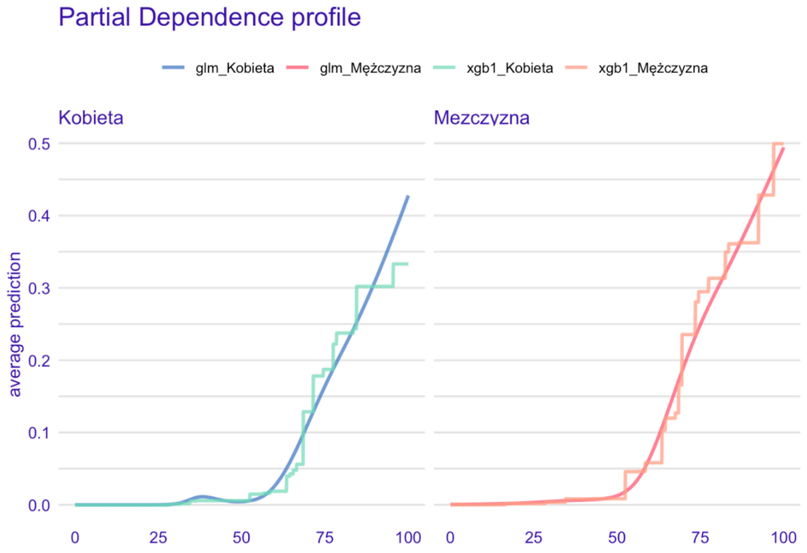
https://en.wikipedia.org/wiki/Mortality_due_to_COVID-19

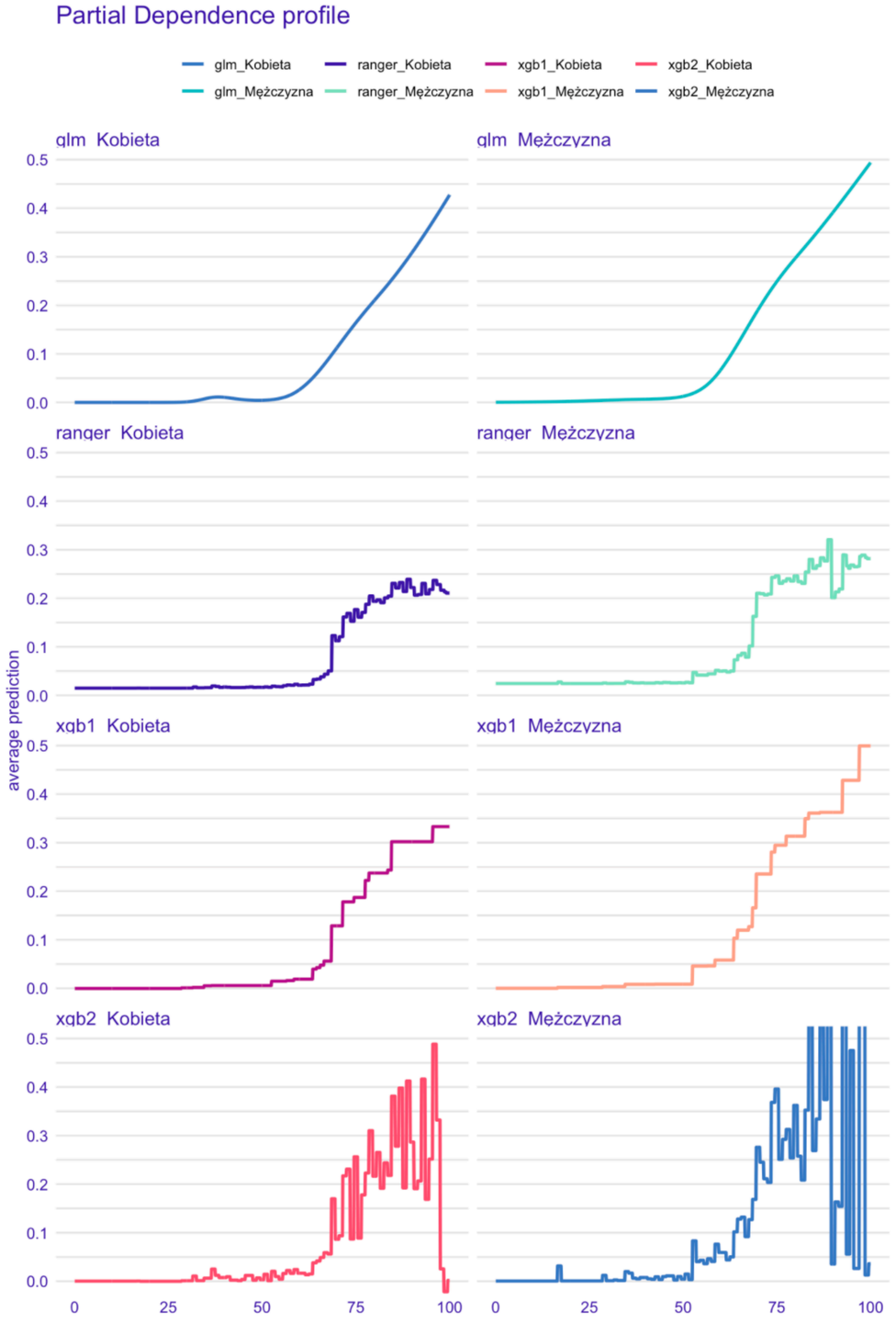




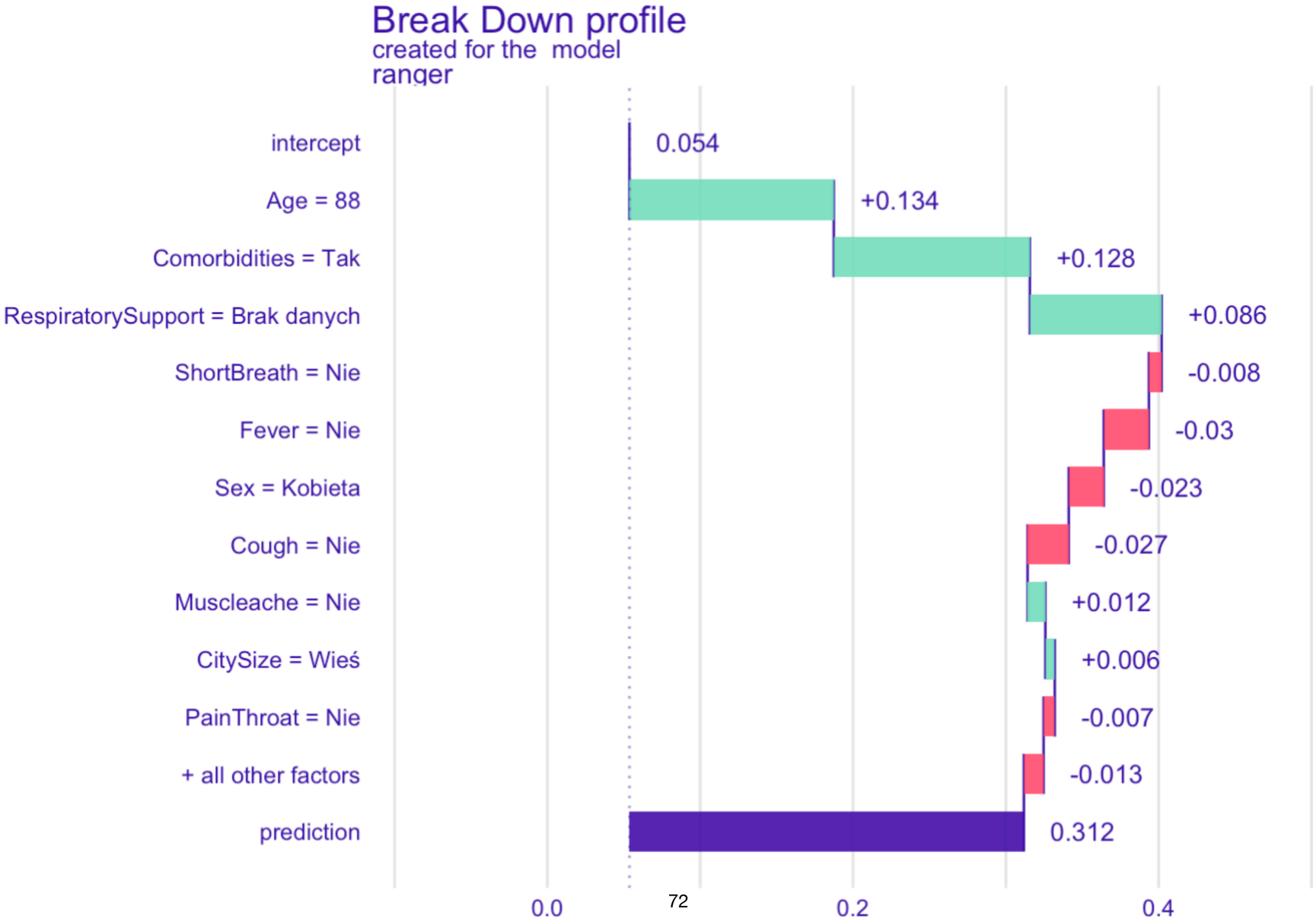
Simple model



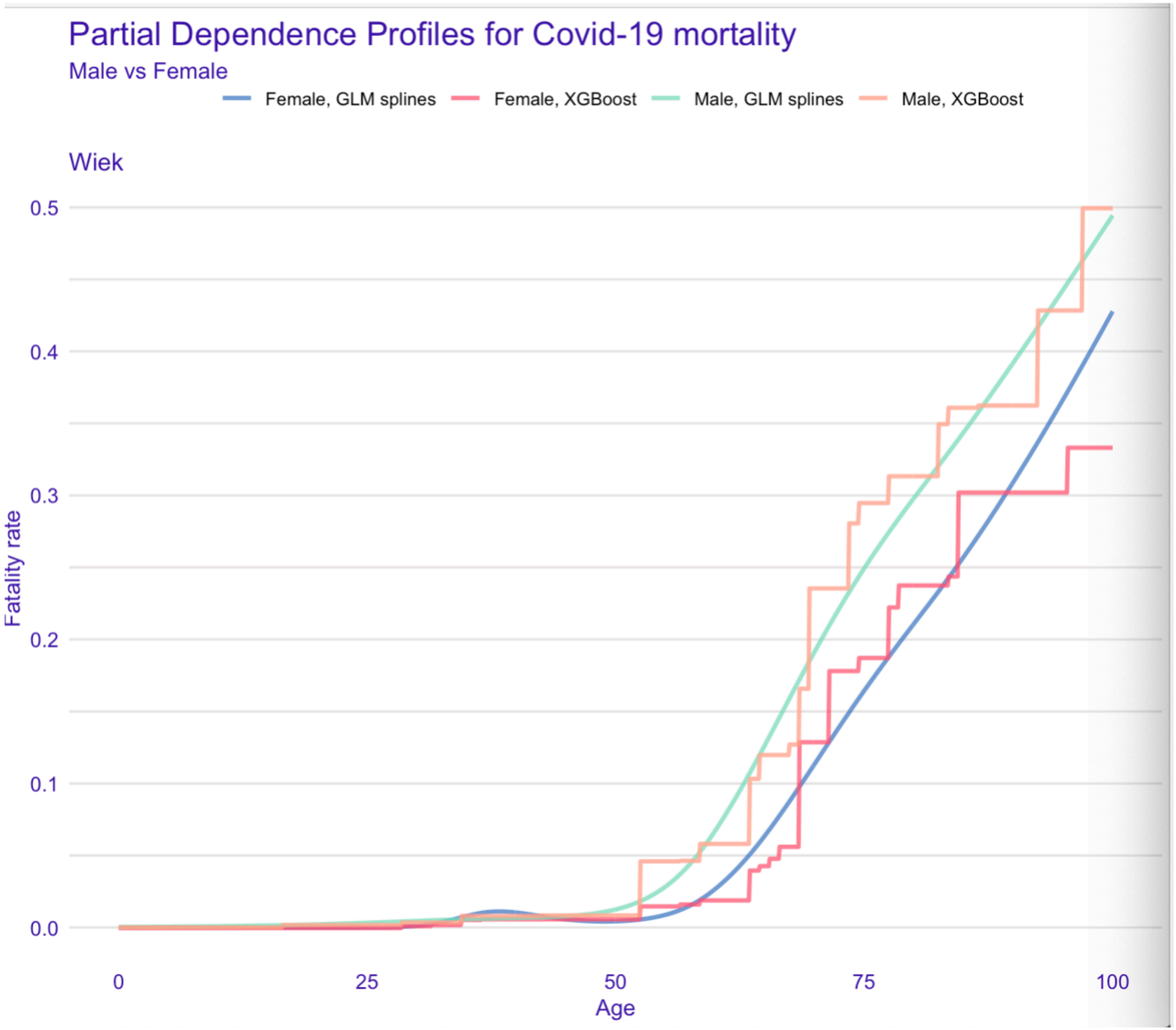


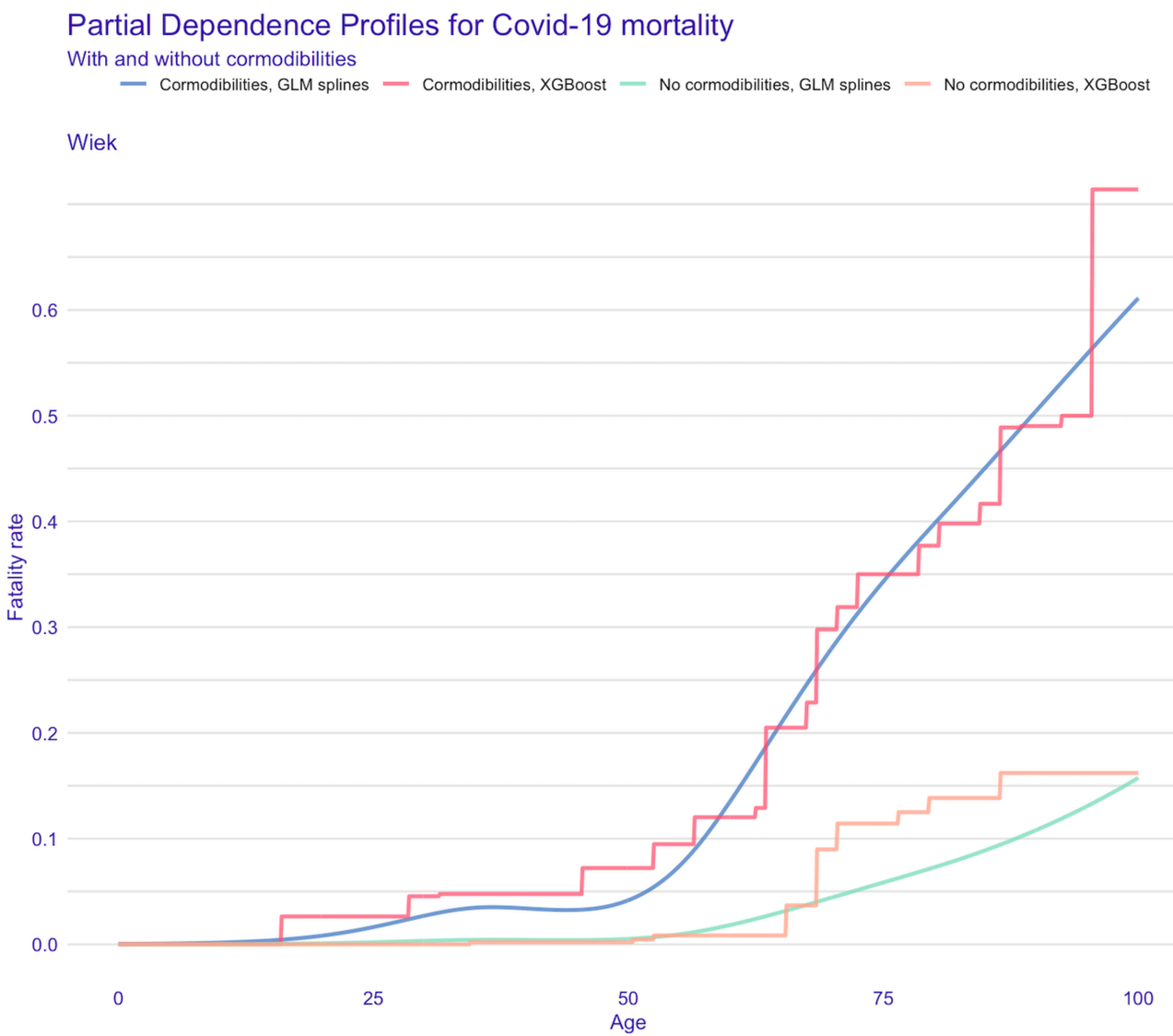


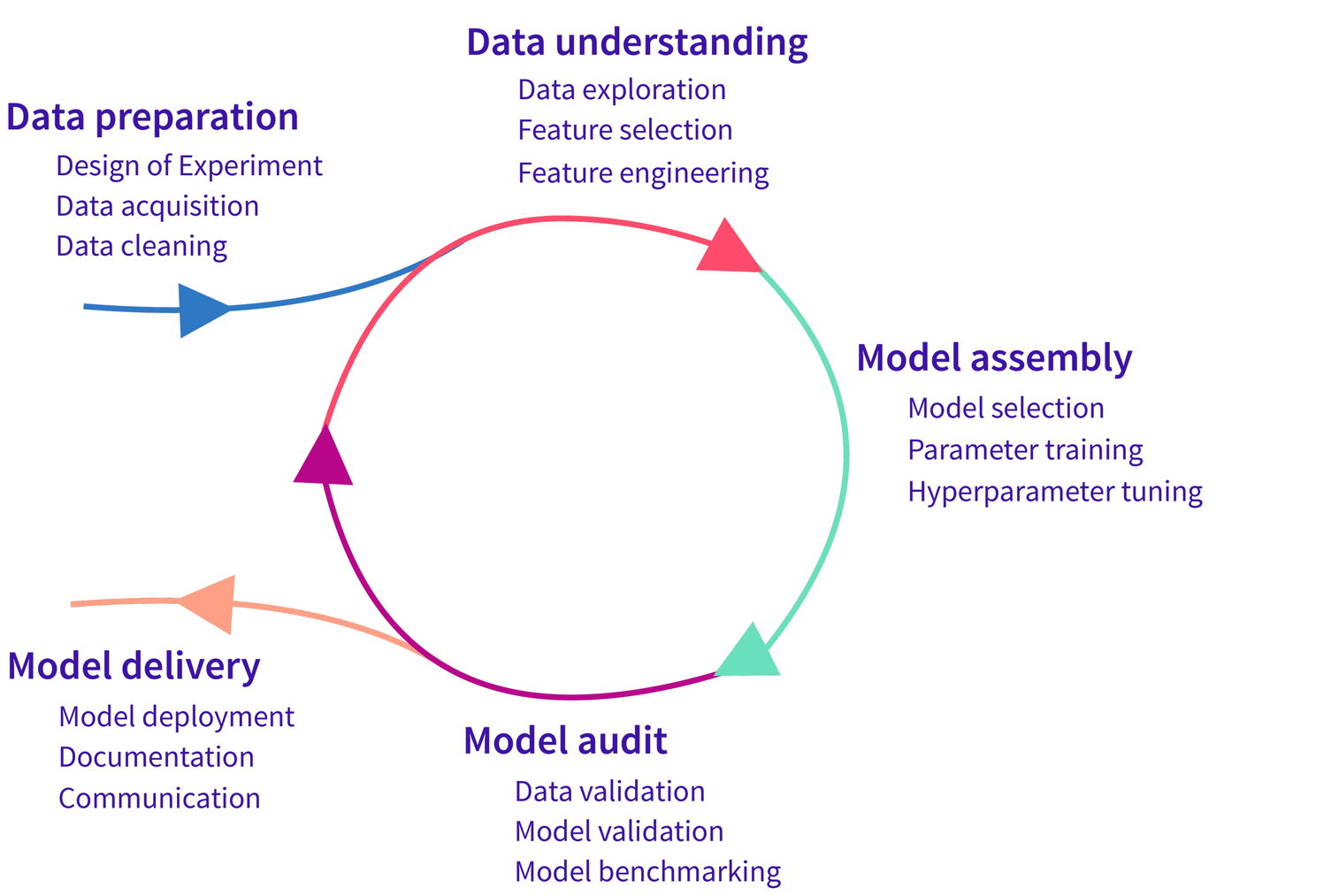
Individual predictions

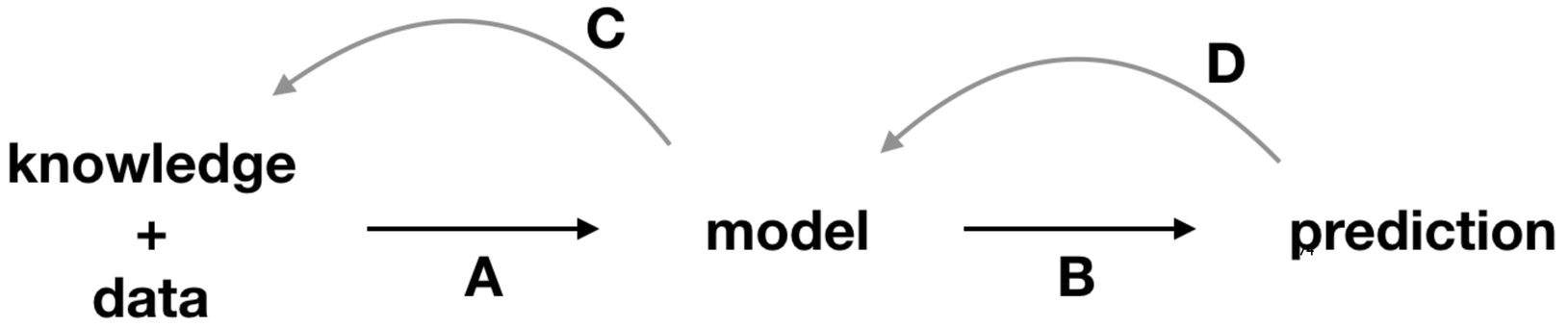


Mortality at better resolution






Data understanding
Data exploration
Feature selection
Feature engineering
Data preparation
Design of Experiment
Data acquisition
Data cleaning
Model assembly
Model selection
Parameter training
Hyperparameter tuning
Model delivery
Model deployment
Documentation
Communication
Model audit
Data validation
Model validation
Model benchmarking


C
D
knowledge
+
data
A
model
B
prediction

hbaniecki [python] gh-actions (#228) b8f21a5 12 days ago

5 contributors

102 lines (67 sloc) | 8.36 KB | Raw | Blame | History

moDel Agnostic Language for Exploration and eXplanation

Overview

Unverified black box model is the path to the failure. Opaqueness leads to distrust. Distrust leads to ignoration. Ignoration leads to rejection.

The `DALEX` package xrays any model and helps to explore and explain its behaviour, helps to understand how complex models are working. The main function `explain()` creates a wrapper around a predictive model. Wrapped models may then be explored and compared with a collection of local and global explainers. Recent developents from the area of Interpretable Machine Learning/eXplainable Artificial Intelligence.

The philosophy behind `DALEX` explanations is described in the Explanatory Model Analysis e-book. The `DALEX` package is a

Questions?