モデルを跨いでデータを見たい

“model-agnostic data explanation”

IML本をベースに、（機械学習）モデルの説明用パッケージをいくつか紹介します

第76回R勉強会@東京（#TokyoR）

1. Interpretability
2. 使ってみた： **mlr**
3. 使ってみた： **DALEX**
4. 使ってみた： **iml**

？誰

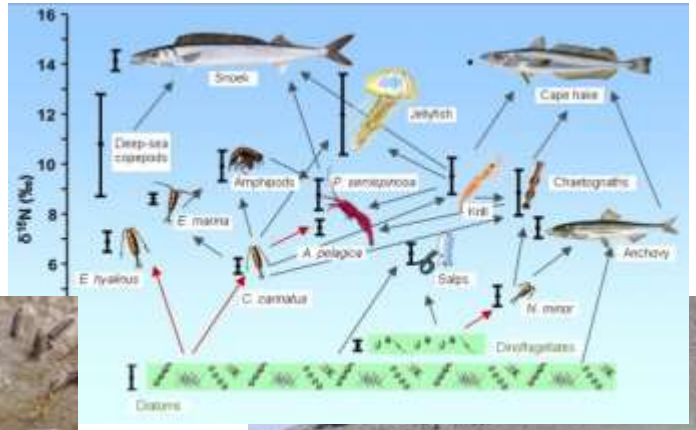
臨床検査事業 の なかのひと

？専門

遊牧@モンゴル（生態学／環境科学）

▼

臨床検査事業の研究所（医療情報学／疫学）



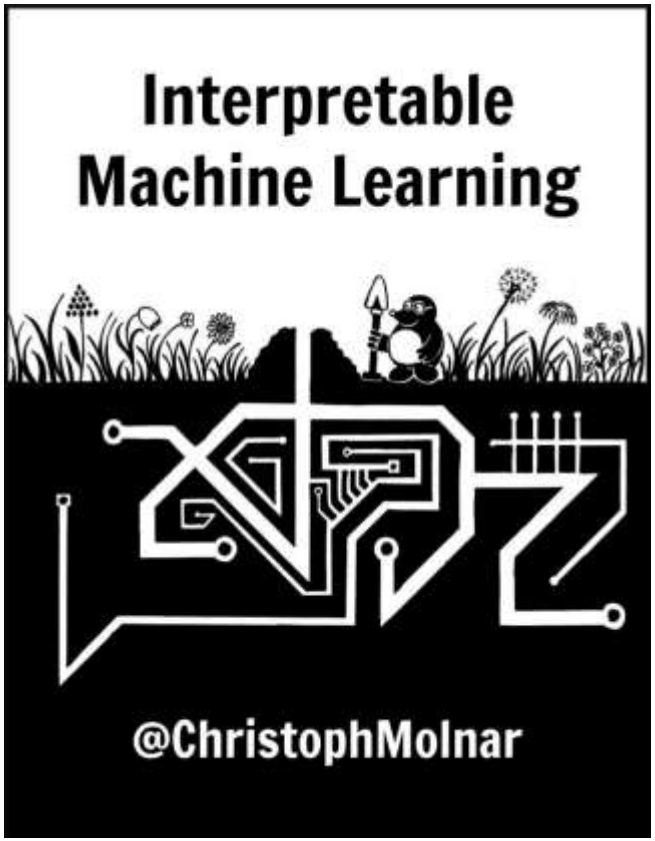
1cm

Interpretable Machine Learning

A Guide for Making Black Box Models Explainable.

Christoph Molnar

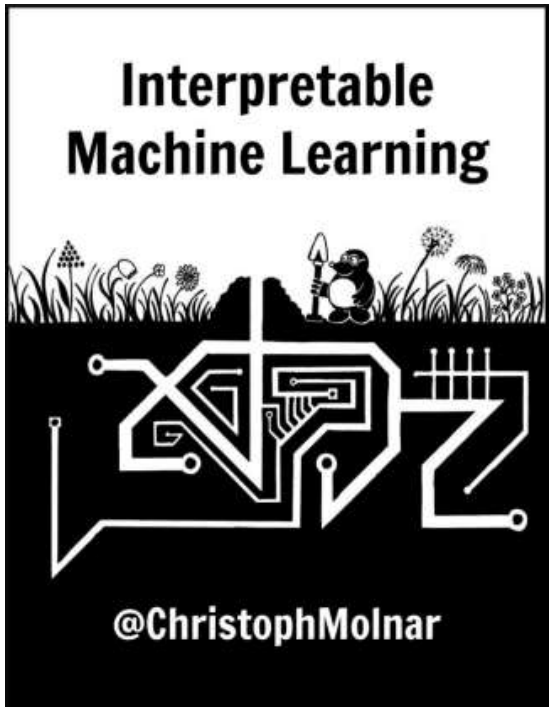
2019-02-21



https://christophm.github.io/interpretable-ml-book/

人にやさしく

- 2.6 **Human-friendly Explanations** は、是非原著を！
  - https://christophm.github.io/interpretable-ml-book/explanation.html
  - 2.6.1 What Is an Explanation?
  - 2.6.2 What Is a Good Explanation?
- 約2,200 words..**ガンバレ！**

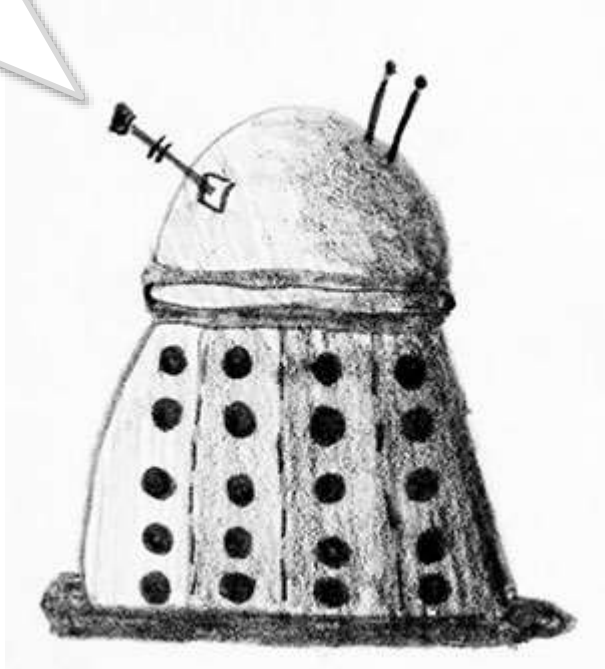


DALEX: Descriptive mAchine Learning EXplanations

Przemysław Biecek

2018-08-11

Explain! Explain! Explain!



The Daleks are a fictional extraterrestrial race portrayed in the Doctor Who BBC series. Rather dim aliens, known to repeat the phrase Explain! very often. Daleks were engineered. They consist of live bodies closed in tank-like robotic shells. They seem like nice mascots for explanations concerning Machine Learning models.

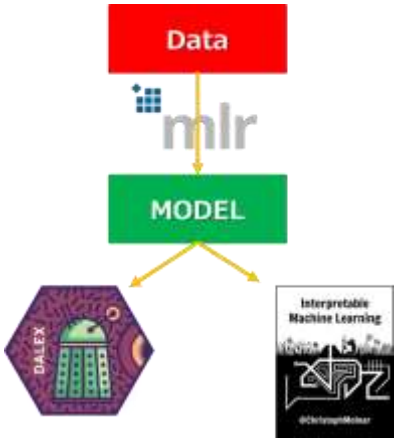
https://pbiecek.github.io/DALEX_docs/

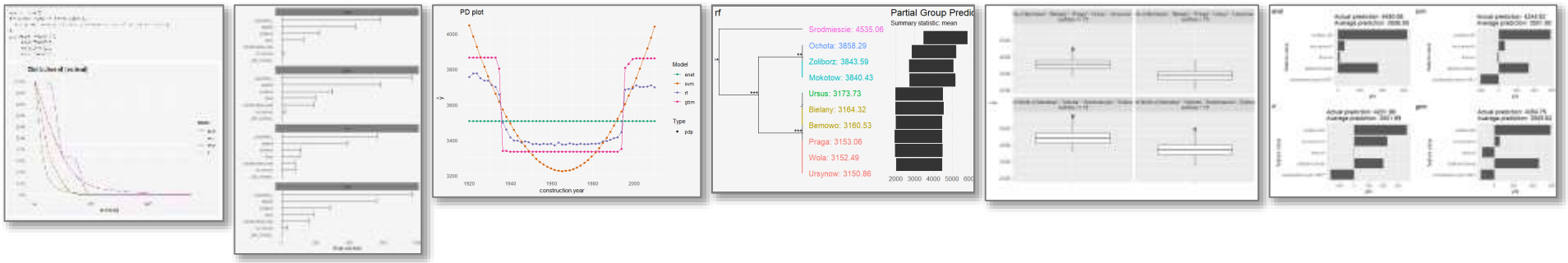
（機械学習）モデルの説明用パッケージをいくつか紹介します

- 深層学習関連の話題は出ません

	Intrinsic	Post hoc
Model-Specific Methods	• Linear Regression • Logistic Regression • GLM, GAM and more • Decision Tree • Decision Rules • RuleFit • Naive Bayes Classifier • K-Nearest Neighbors	• Feature Importance (OOB error@RF; gain/cover/weight @XGB) • Feature Contribution (forestFloor@RF, XGBoostexplainer, lightgbmExplainer) • Alternate / Enumerate lasso (@LASSO) • inTrees / defragTrees (@RF) • Actionable feature tweaking (@RF/XGB)
Model-Agnostic Methods	Intrins Model	• Partial Dependence Plot • Individual Conditional Expectation • Accumulated Local Effects Plot • Feature Interaction • Permutation Feature Importance • Global Surrogate • Local Explanation (LIME, Shapley Values, breakDown)
Example-based Explanations	??	• Counterfactual Explanations • Adversarial Examples • Prototypes and Criticisms • Influential Instances

DALEX & iml package





（機械学習）モデルの説明用パッケージをいくつか紹介します

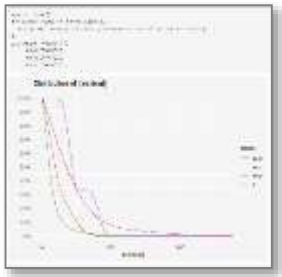
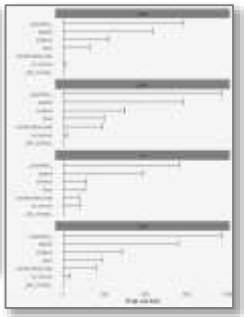
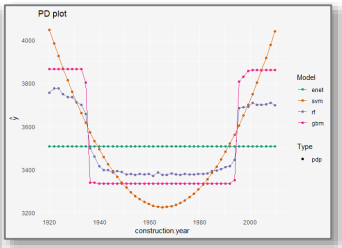
- 深層学習関連の話題は出ません

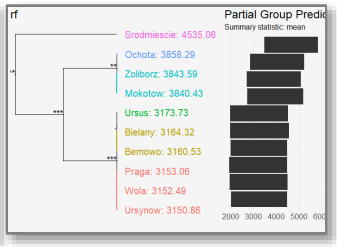
	Intrinsic	Post hoc
Model-Specific Methods	• Linear Regression • Logistic Regression • GLM, GAM and more	• Feature Importance (OOB error@RF; gain/cover/weight @XGB) • Feature Contribution (forestFloor@RF,
		• Influential Instances

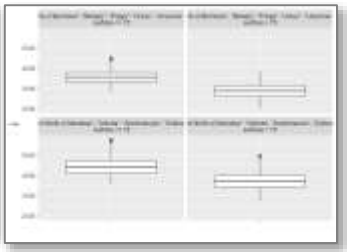
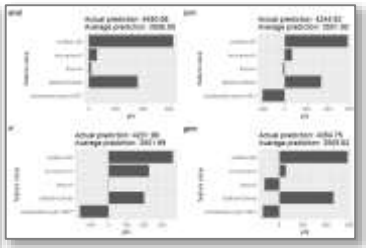
Rで使いたい

本日のコードはすべてgithubで公開しています

https://github.com/katokohaku/compareModels_with_MLR-IML



こっちを見ましょう

機械学習モデルの判断根拠の説明

1,082 回視聴　20　1　共有　保存

STAIR Lab
2019/01/15 に公開

チャンネル登録 157

第20回ステアラボ人工知能セミナー (2018年12月21日)

1. https://www.youtube.com/watch?v=Fgza_C6KphU
2. https://www.slideshare.net/SatoshiHara3/ss-126157179

こっちを見ましょう

https://speakerdeck.com/line_developers/machine-learning-and-interpretability

Properties of Explanation Methods（1）

Expressive Power（表現力）

- 何をどう説明しているか？
- IF-THEN rules, decision trees, a weighted sum, natural language

Algorithmic Complexity（複雑性）

- 説明を生成する手続きがどのくらい複雑か？
- メソッドの計算時間がボトルネックとなる場合がある

https://christophm.github.io/interpretable-ml-book/properties.html

Properties of Explanation Methods（2）

Translucency (不透明性)

- 説明方法が機械学習モデルにどの程度依存しているか？
- 高い：　たとえば、線形回帰モデル（重みを見ればわかる）
- 低い：　入力→出力しか得られない（ブラックボックス）

Portability （移植性）

- 説明方法を適用できる機械学習モデルの範囲
- 高い：　ブラックボックスでも説明できる方法（入力→出力さえあればよい）
- 低い：　モデル固有の性質やアルゴリズムに依存する説明方法

https://christophm.github.io/interpretable-ml-book/properties.html

Properties of Individual Explanations (1)

Accuracy

- 説明モデル自体のデータの予測精度
- ブラックボックスモデルの予測の代わりに説明モデルを使用する場合には重要

Fidelity（忠実度）

- ブラックボックスの予測を説明モデルがどれだけうまく近似しているか
- 説明モデルの忠実度が高い場合、説明モデルの予測精度も高くなる

Degree of Importance（重要度）

- ある説明や説明に用いられる変数が、説明全体においてどれだけ重要か
- ある予測に、どのルールのどの条件が最も重要だったか明らかか？

https://christophm.github.io/interpretable-ml-book/properties.html

Properties of Individual Explanations (2)

Novelty（新規性）

- 未見の予測対象に対するブラックボックスモデルの振る舞いが、説明モデルにおいても反映されるか？
- 予測対象が、訓練データの分布から離れているとき、学習済モデルの予測が不正確になり、説明が役に立たなくなる可能性がある

Representativeness（代表性）

- 説明のカバレッジがどの程度あるか？
  - モデル全体を説明する（例えば、線形回帰モデルにおける重みの解釈）
  - 個々の予測だけを説明する（例えば、Shapley値）

https://christophm.github.io/interpretable-ml-book/properties.html

Properties of Individual Explanations (3)

Consistency（一貫性）

- 複数のモデルで同じタスクについて同様の予測が得られるとき、モデル間で説明がどの程度異なるか
  - 予測が一貫したデータの背景に依存している場合
    - どんなモデルであれ説明は非常に似ることが期待される
  - 異なるモデルが、異なる特徴量セットで同じような予測を生成する場合
    - 似たような予測に対する説明が異なる（羅生門効果）

羅生門効果（らしょうもんこうか、英: Rashomon effect）とは、ひとつの出来事において、人々がそれぞれに見解を主張すると矛盾してしまう現象のことであり、心理学、犯罪学、社会学などの社会科学で使われることがある。 映画『**羅生門**』に由来する。

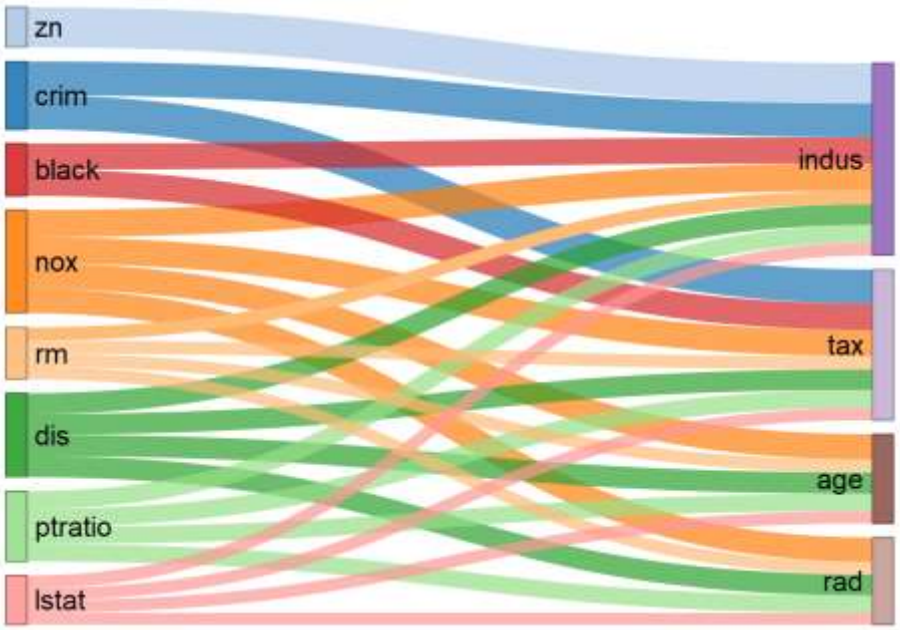
羅生門効果 - Wikipedia
https://ja.wikipedia.org/wiki/羅生門効果

https://christophm.github.io/interpretable-ml-book/properties.html

the alternate features search

- Lasso変数選択の際の見落としの問題に対する提案
- 代替可能なモデル(変数候補)を提示する
- 予測はほとんど変わらないが、**納得感のある別の説明**を得られる可能性

元の変数



置換候補

S. Hara, T. Maehara, Finding Alternate Features in Lasso, arXiv:1611.05940, 2016.
https://github.com/sato9hara/LassoVariants
https://github.com/katokohaku/AlternateLassoR

Taxonomy of Interpretability Methods

Intrinsic

- モデルの複雑性を制約することで、モデルそのものが解釈可能

Post hoc

- データからモデルを学習した後で説明手法を適用する

	Intrinsic	Post hoc
Model-Specific Methods	• Linear Regression • Logistic Regression • GLM, GAM and more • Decision Tree • Decision Rules • RuleFit • Naive Bayes Classifier • K-Nearest Neighbors	• Feature Importance (OOB error@RF; gain/cover/weight @XGB) • Feature Contribution (forestFloor@RF, XGBoostexplainer, lightgbmExplainer) • Alternate / Enumerate lasso (@LASSO) • inTrees / defragTrees (@RF) • Actionable feature tweaking (@RF/XGB)
Model-Agnostic Methods	Intrinsic interpretable Model にも適用可能	• Partial Dependence Plot • Individual Conditional Expectation • Accumulated Local Effects Plot • Feature Interaction • Permutation Feature Importance • Global Surrogate • Local Explanation (LIME, Shapley Values, breakDown)
Example-based Explanations	??	• Counterfactual Explanations • Adversarial Examples • Prototypes and Criticisms • Influential Instances

Taxonomy of Interpretability Methods

Model-specific

- モデルの設計思想やアルゴリズムを説明に転用する

Model-agnostic

- モデルの入力→出力に対する反応を観察する

Example-based

- 代表的・イレギュラーなデータに対するモデルの反応を観察する

	Intrinsic	Post hoc
Model-Specific Methods	• Linear Regression • Logistic Regression • GLM, GAM and more • Decision Tree • Decision Rules • RuleFit • Naive Bayes Classifier • K-Nearest Neighbors	• Feature Importance (OOB error@RF; gain/cover/weight @XGB) • Feature Contribution (forestFloor@RF, XGBoostexplainer, lightgbmExplainer) • Alternate / Enumerate lasso (@LASSO) • inTrees / defragTrees (@RF) • Actionable feature tweaking (@RF/XGB)
Model-Agnostic Methods	Intrinsic interpretable Model にも適用可能	• Partial Dependence Plot • Individual Conditional Expectation • Accumulated Local Effects Plot • Feature Interaction • Permutation Feature Importance • Global Surrogate • Local Explanation (LIME, Shapley Values, breakDown)
Example-based Explanations	??	• Counterfactual Explanations • Adversarial Examples • Prototypes and Criticisms • Influential Instances

本日の話題

	Intrinsic	Post hoc
Model-Specific Methods	• Linear Regression • Logistic Regression • GLM, GAM and more • Decision Tree • Decision Rules • RuleFit • Naive Bayes Classifier • K-Nearest Neighbors	• Feature Importance (OOB error@RF; gain/cover/weight @XGB) • Feature Contribution (forestFloor@RF, XGBoostexplainer, lightgbmExplainer) • Alternate / Enumerate lasso (@LASSO) • inTrees / defragTrees (@RF) • Actionable feature tweaking (@RF/XGB)
Model-Agnostic Methods	Intrin Mode	• Partial Dependence Plot • Individual Conditional Expectation • Accumulated Local Effects Plot • Feature Interaction • Permutation Feature Importance • Global Surrogate • Local Explanation (LIME, Shapley Values, breakDown)
Example-based Explanations	??	• Counterfactual Explanations • Adversarial Examples • Prototypes and Criticisms • Influential Instances

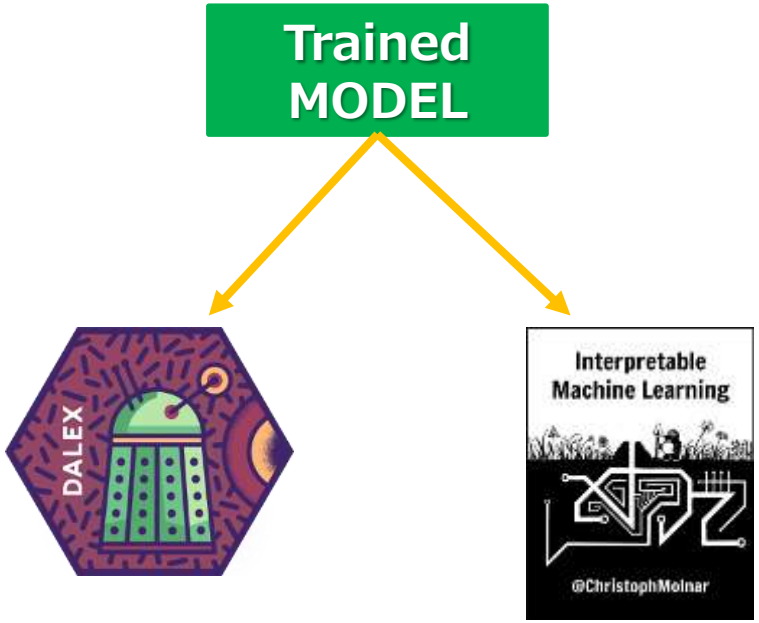
DALEX & iml package

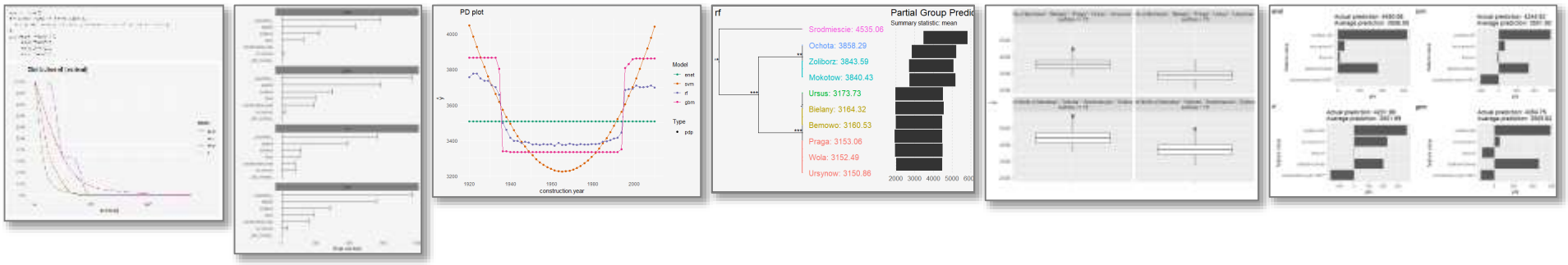
Why model-agnostic ?

- モチベーション
  - **モデル横断的に**データと予測を評価・比較したい
- うれしさ
  - モジュール化できる ＝ **解釈のプロセスを自動化**できる
  - ブラックボックスモデルの**置き換えを容易化**できる
- ※ intrinsically methods には別の使い道がある

説明のアプローチ

0. 説明の準備
1. モデルの性能や特性の評価
2. 特徴量（変数）に対するモデルの応答をみる
3. あるデータに対する予測がどのように得られたか説明する





本日のパッケージ： iml

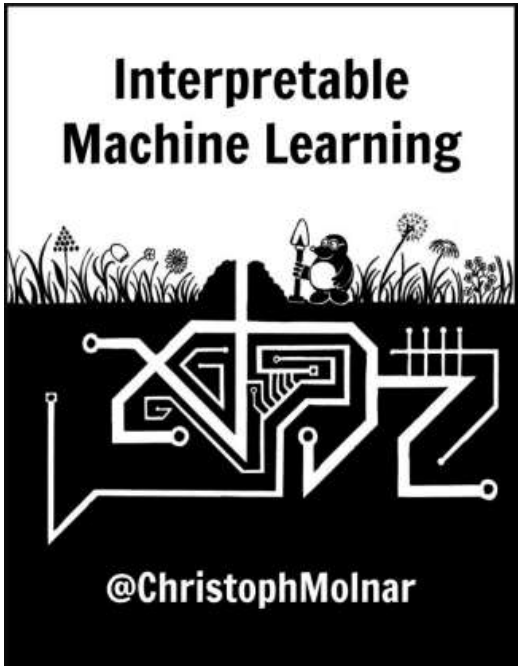
Interpretable Machine Learning

A Guide for Making Black Box Models Explainable.

Christoph Molnar

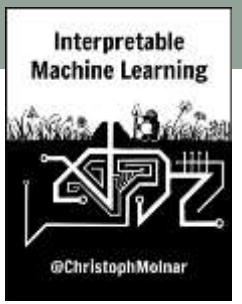
2019-02-21

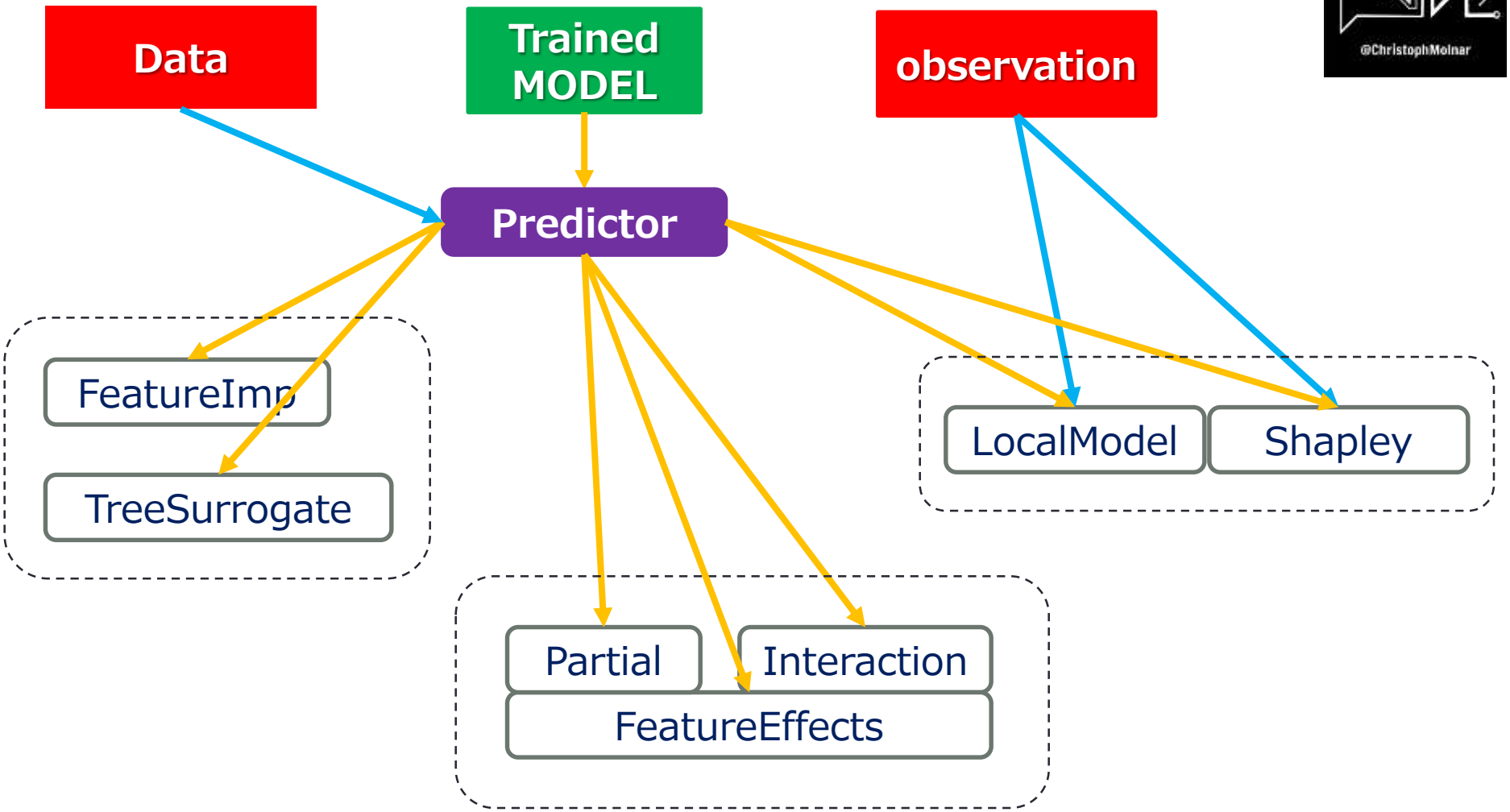
- 書籍の中で紹介されている解析手法をまとめてパッケージとして提供
- **R6クラスをベース**とする独自実装
- 書籍ではその他の実装も、(あれば)紹介しているので、お好みで。
- CRANからインストール可能



https://cran.r-project.org/package=iml

ワークフロー： iml





https://github.com/christophM/iml

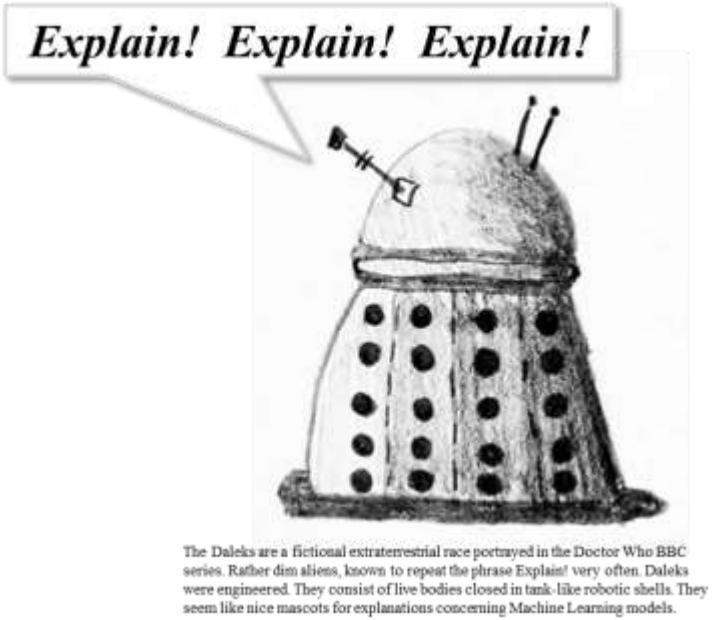
本日のパッケージ： DALEX

DALEX: Descriptive mAchine Learning EXplanations

Przemysław Biecek

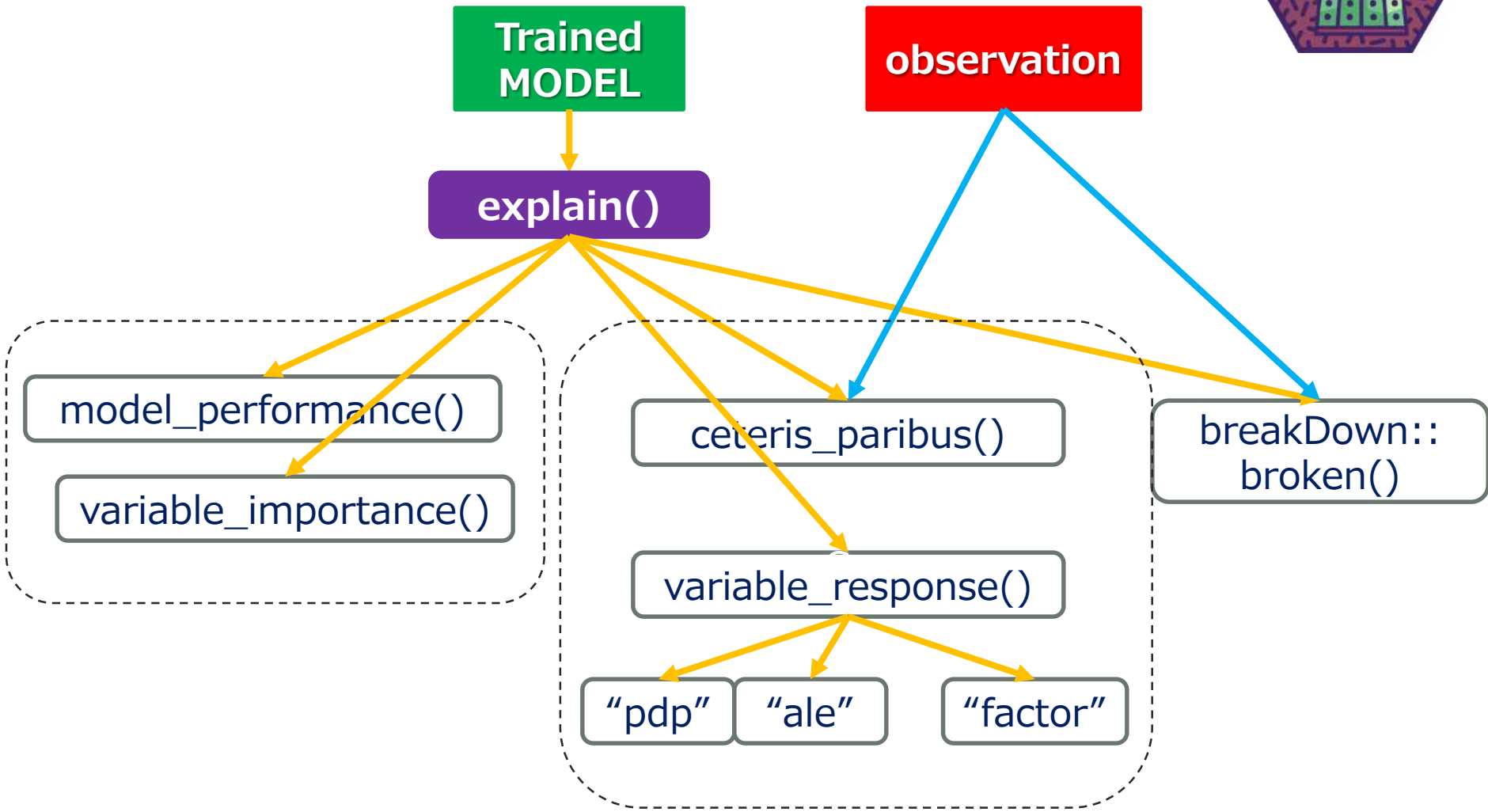
2018-08-11

- 既存の手法＋著者たちがこれまでに開発・公表したアルゴリズムのラッパー
- **モデル間の比較を強く意識した設計**
- 基本的には既存の外部パッケージに依存
- CRANからインストール可能



https://cran.r-project.org/package=DALEX

ワークフロー： DALEX



https://pbiecek.github.io/DALEX_docs/2-architecture.html#architecture

iml vs DALEX

policy	method name	iml	DALEX
understand entire model	residuals and goodness of fit	✗	✓
	permutation importance	✓	✓
	global surrogate	Tree surrogate	✗
understand feature(s)	Merging Path Plot (PDP for categorical data)	✗	✓
	Partial Dependence Plot (PDP for continuous data)	✓	✓
	Individual Conditional Expectation (ICE)	✓	Ceteris Paribus Plots
	Accumulated Local Effects (ALE) Plot	✓	✓
	Feature Interaction	✓	✗
local interpretation (for single prediction)	LIME	✓	✗
	SHAPLY value	✓	✗
	breakDown	✗	✓

mlr : Machine Learning in R

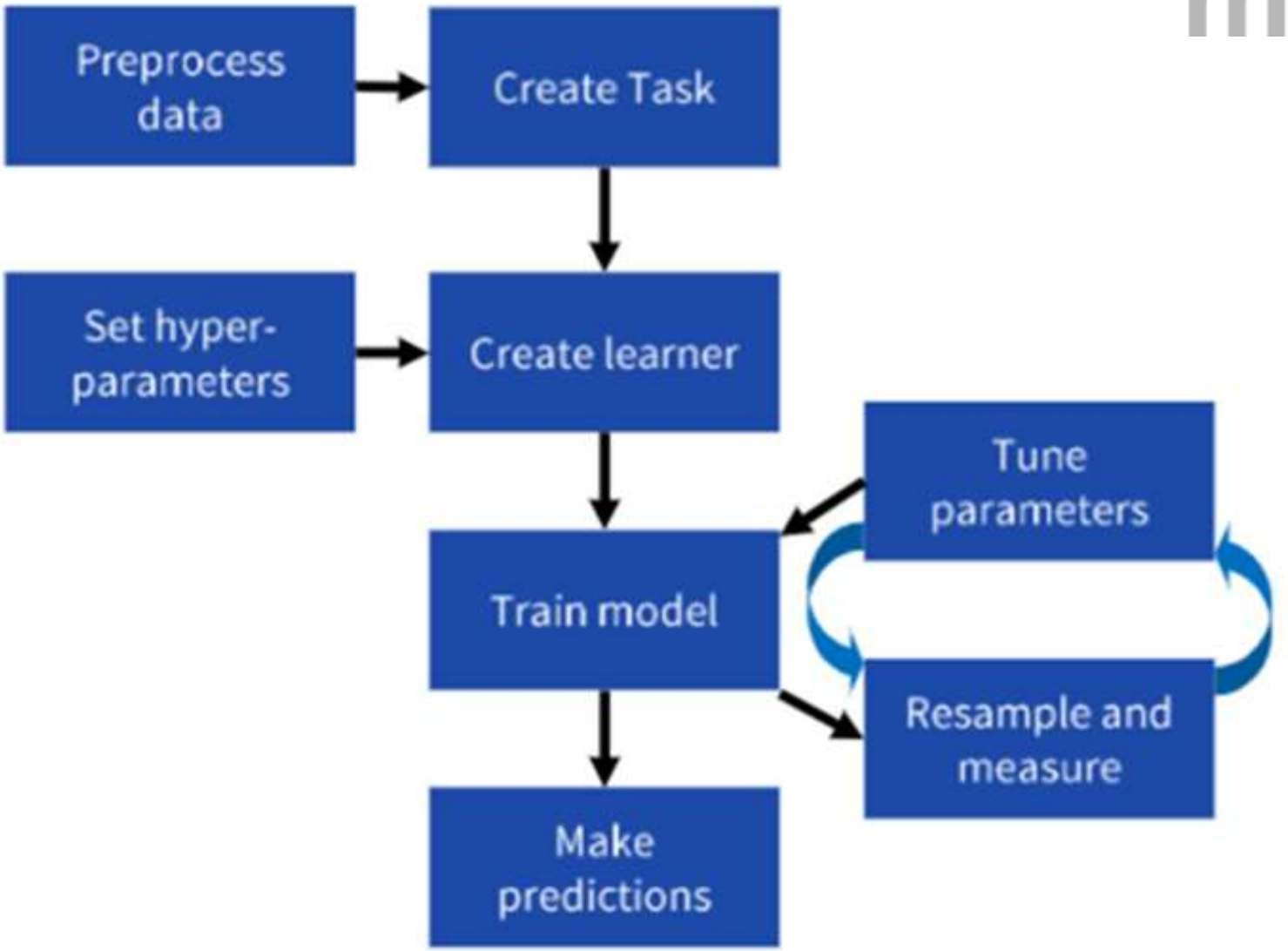
- Rで提供されるさまざまな機械学習モデル（パッケージ）に対するgenericなフレームワークを提供する
  - Classification / Regression / Survival analysis / Clustering

- 各モデルのパラメータなどを隠蔽することで、統一的なインターフェースによって、一連のワークフローを共通化する

- R6ベースに移行するかも？　→　**mlr3**

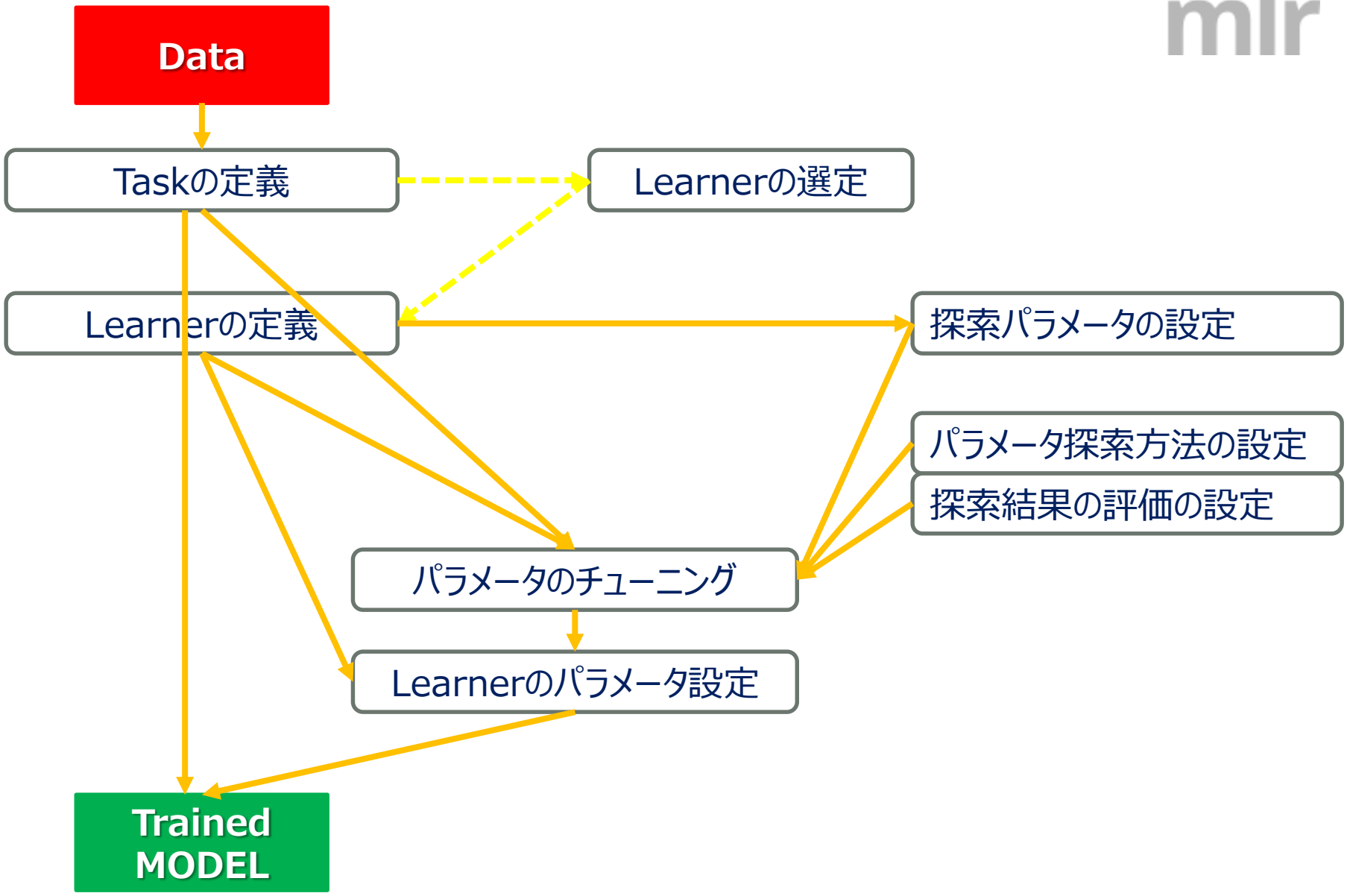
https://mlr-org.com/
https://mlr3.mlr-org.com/

ワークフロー： mlr

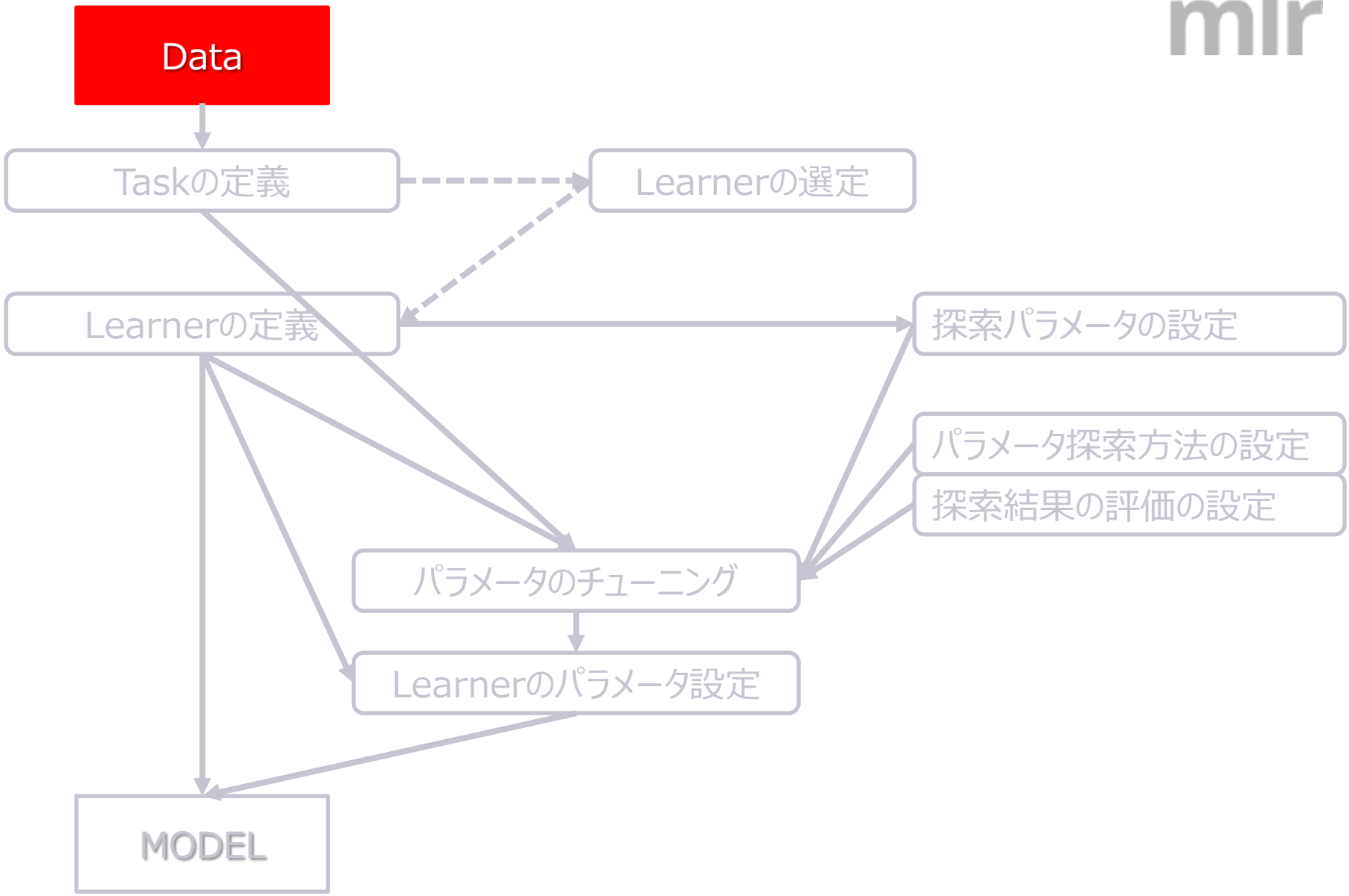


https://github.com/mlr-org/mlr/blob/master/addon/cheatsheet/MlrCheatsheet.pdf

ワークフロー： mlr


Data
Taskの定義
Learnerの選定
Learnerの定義
探索パラメータの設定
パラメータ探索方法の設定
探索結果の評価の設定
パラメータのチューニング
Learnerのパラメータ設定
Trained MODEL

ワークフロー： mlr


Data
Taskの定義
Learnerの選定
Learnerの定義
探索パラメータの設定
パラメータ探索方法の設定
探索結果の評価の設定
パラメータのチューニング
Learnerのパラメータ設定
MODEL

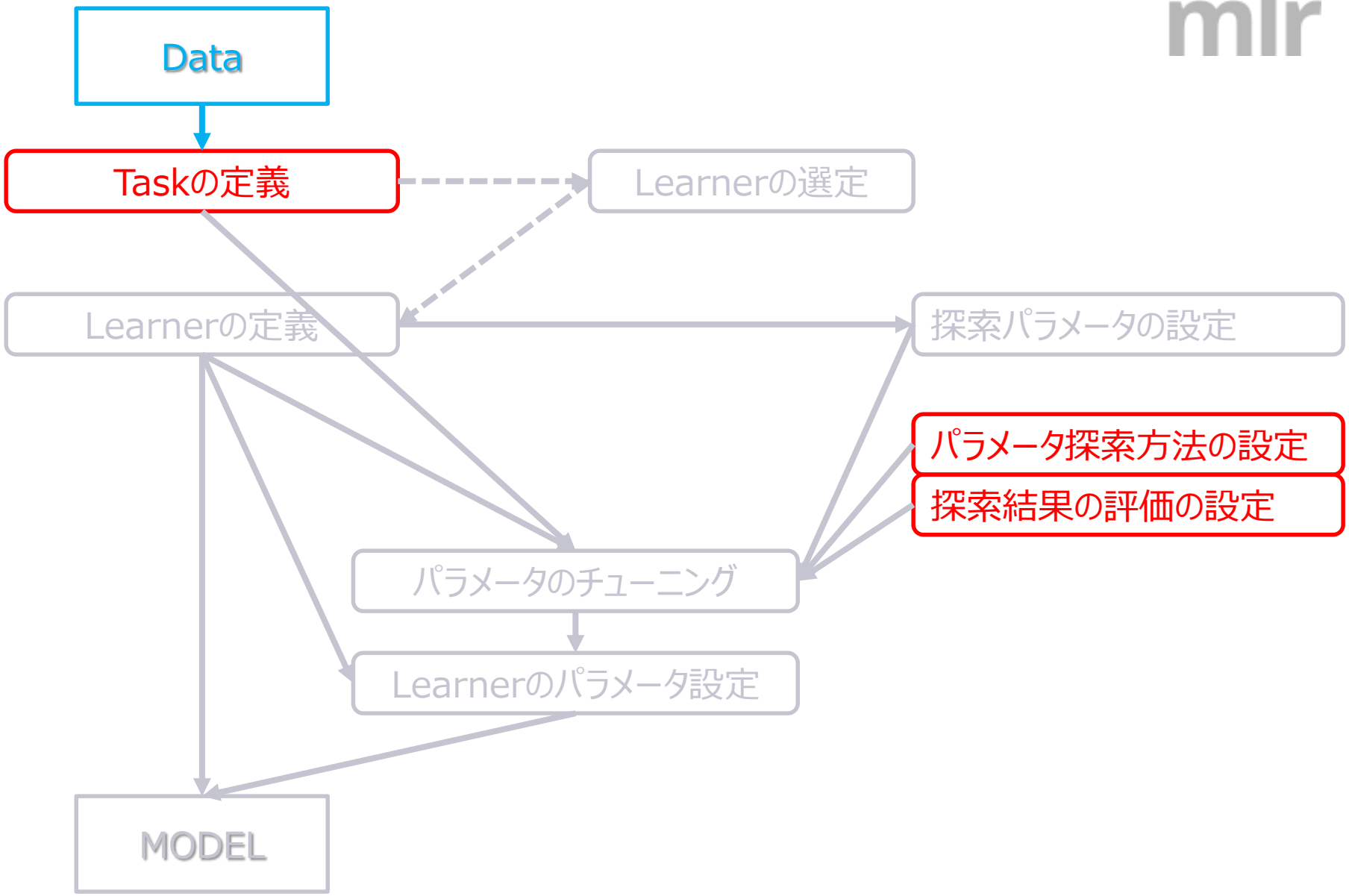
apartment データセット

PBImisc packageからコピーされたデータセット
1. m2.price = 面積当たり価格
2. surface = 面積 (m^2)
3. n.rooms = 部屋数 (面積と相関)
4. district = アパートがある地区（10 levels）
5. floor = 階数
6. construction.date = 築年数

```
data(apartments, package = "DALEX")
data(apartmentsTest, package = "DALEX")

apartments %>% str()
```

```
'data.frame':   1000 obs. of  6 variables:
 $ m2.price          : num  5897 1818 3643 3517 3013 ...
 $ construction.year: num  1953 1992 1937 1995 1992 ...
 $ surface           : num  25 143 56 93 144 61 127 105 145 112 ...
 $ floor             : int  3 9 1 7 6 6 8 8 6 9 ...
 $ no.rooms          : num  1 5 2 3 5 2 5 4 6 4 ...
 $ district          : Factor w/ 10 levels "Bemowo","Bielany",..: 6 2 5 4 3 6 3 7
```

ワークフロー： mlr


Data
Taskの定義
Learnerの選定
Learnerの定義
探索パラメータの設定
パラメータ探索方法の設定
探索結果の評価の設定
パラメータのチューニング
Learnerのパラメータ設定
MODEL

```
task <- makeRegrTask(id = "ap", data = apartments, target = "m2.price")
task %>% print()
```

```
Supervised task: ap
Type: regr
Target: m2.price
Observations: 1000
Features:
   numerics     factors     ordered functionals
          4           1           0           0
Missings: FALSE
Has weights: FALSE
Has blocking: FALSE
Has coordinates: FALSE
```

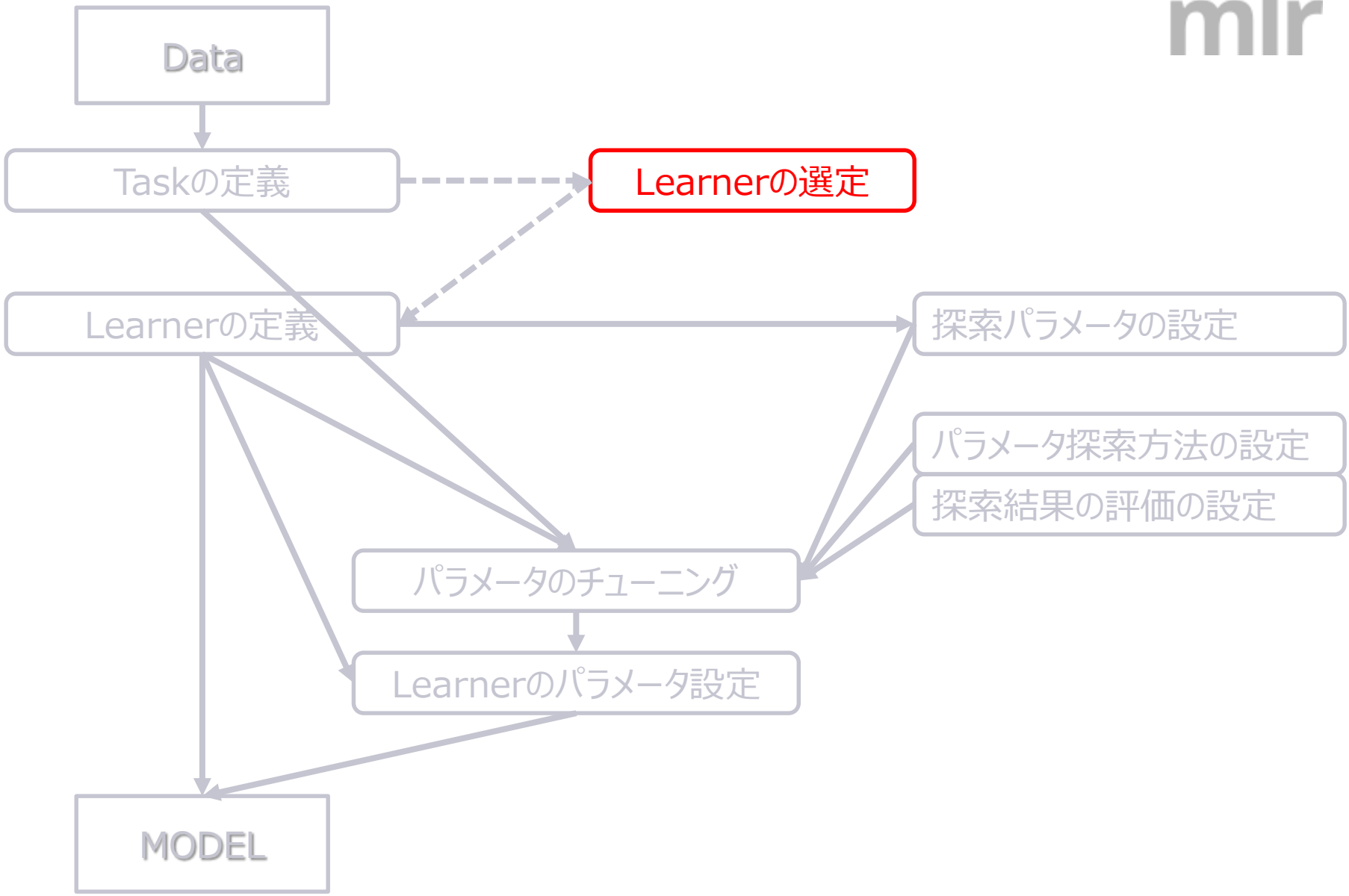
```
tune.ctrl <- makeTuneControlRandom()
tune.ctrl %>% print()
```

```
Tune control: TuneControlRandom
Same resampling instance: TRUE
Imputation value: <worst>
Start: <NULL>
Budget: 100
Tune threshold: FALSE
Further arguments: maxit=100
```

```
res.desc <- makeResampleDesc("CV", iters = 2)
res.desc %>% print()
```

```
Resample description: cross-validation with 2 iterations.
Predict: test
Stratification: FALSE
```

ワークフロー： mlr

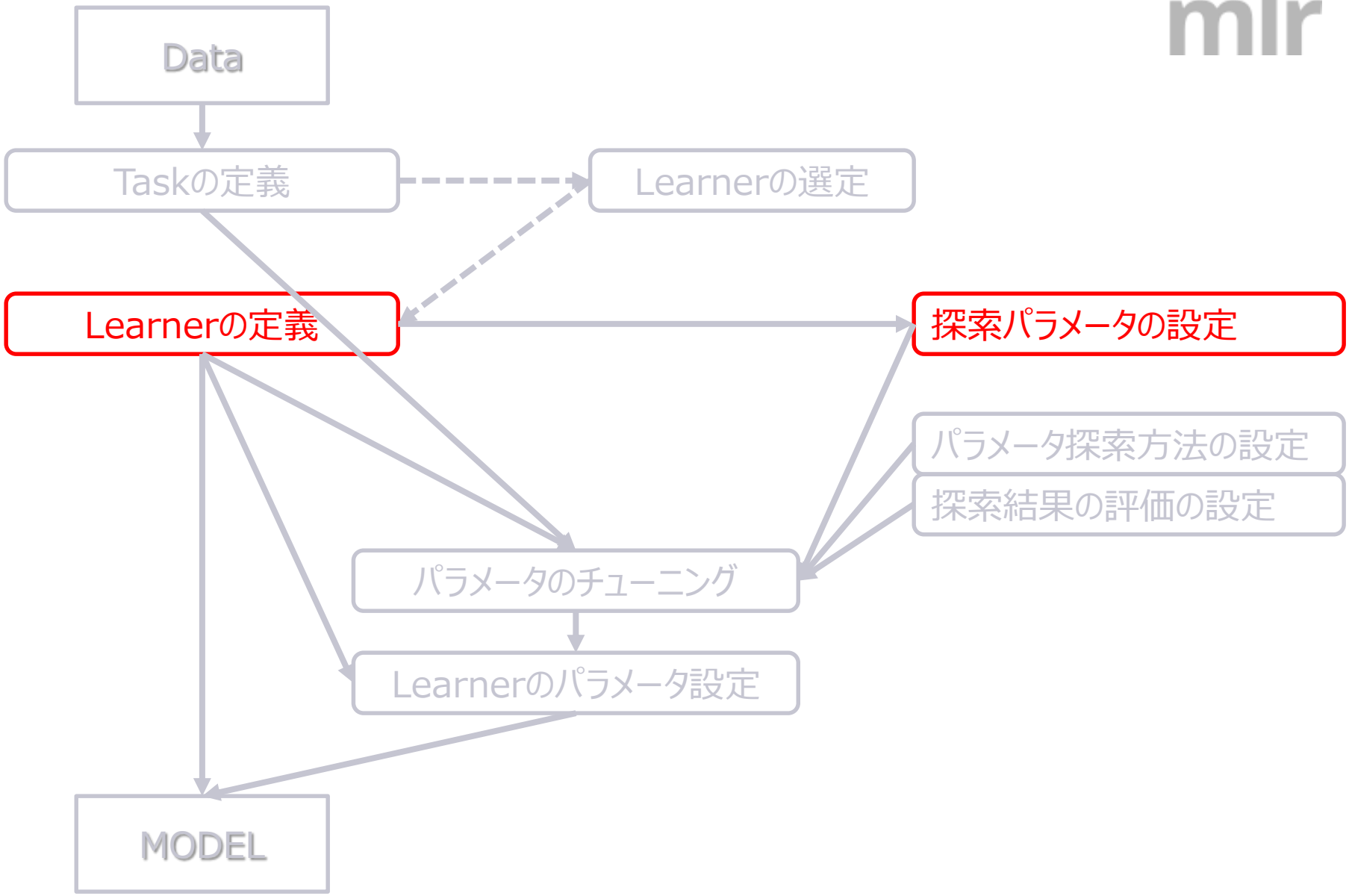

Data
Taskの定義
Learnerの選定
Learnerの定義
探索パラメータの設定
パラメータ探索方法の設定
探索結果の評価の設定
パラメータのチューニング
Learnerのパラメータ設定
MODEL

ワークフロー： mlr


```
listLearners(task) %>%
  select(class, short.name, package)
```

```
              class  short.name     package
1 regr.bartMachine bartmachine bartMachine
2       regr.bcart       bcart         tgp
3        regr.brnn        brnn        brnn
4        regr.btgp        btgp         tgp
5     regr.btgpllm     btgpllm         tgp
6        regr.btlm        btlm         tgp
... (#rows: 41, #cols: 3)
```

ワークフロー： mlr


Data
Taskの定義
Learnerの選定
Learnerの定義
探索パラメータの設定
パラメータ探索方法の設定
探索結果の評価の設定
パラメータのチューニング
Learnerのパラメータ設定
MODEL

linear regression with penalty (elastincnet)

```
# getLearnerParamSet(makeLearner("regr.glmnet"))
learner[["enet"]]<- makeLearner("regr.glmnet")
par.set[["enet"]] <- makeParamSet(
  makeNumericParam("alpha", lower = 0, upper = 1),
  makeNumericParam("s",     lower = 1, upper = 10^3))
```

Support vector machine (SVM)

```
# getLearnerParamSet(makeLearner("regr.ksvm"))
learner[["svm"]]  <- makeLearner("regr.ksvm", kernel = "rbfdot")
par.set[["svm"]]  <- makeParamSet(
  makeNumericParam("C",     lower = -3, upper = 3, trafo = function(x) 10^x),
  makeNumericParam("sigma", lower = -3, upper = 3, trafo = function(x) 10^x))
```

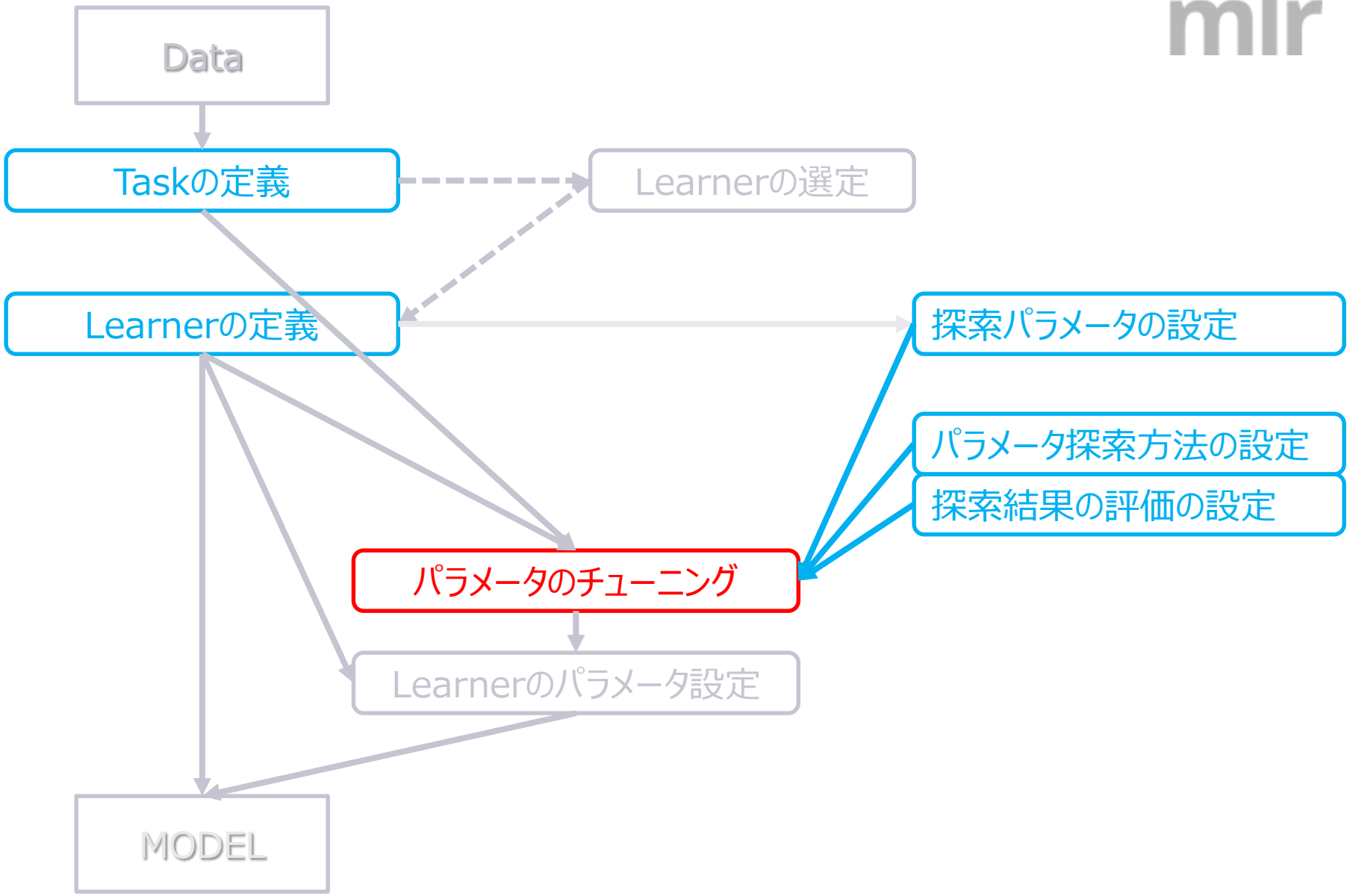
random forest (RF)

```
# getLearnerParamSet(makeLearner("regr.randomForest"))
learner[["rf"]] <- makeLearner("regr.randomForest")
par.set[["rf"]] <- makeParamSet(
  makeIntegerParam("ntree", lower=50, upper=1000))
```

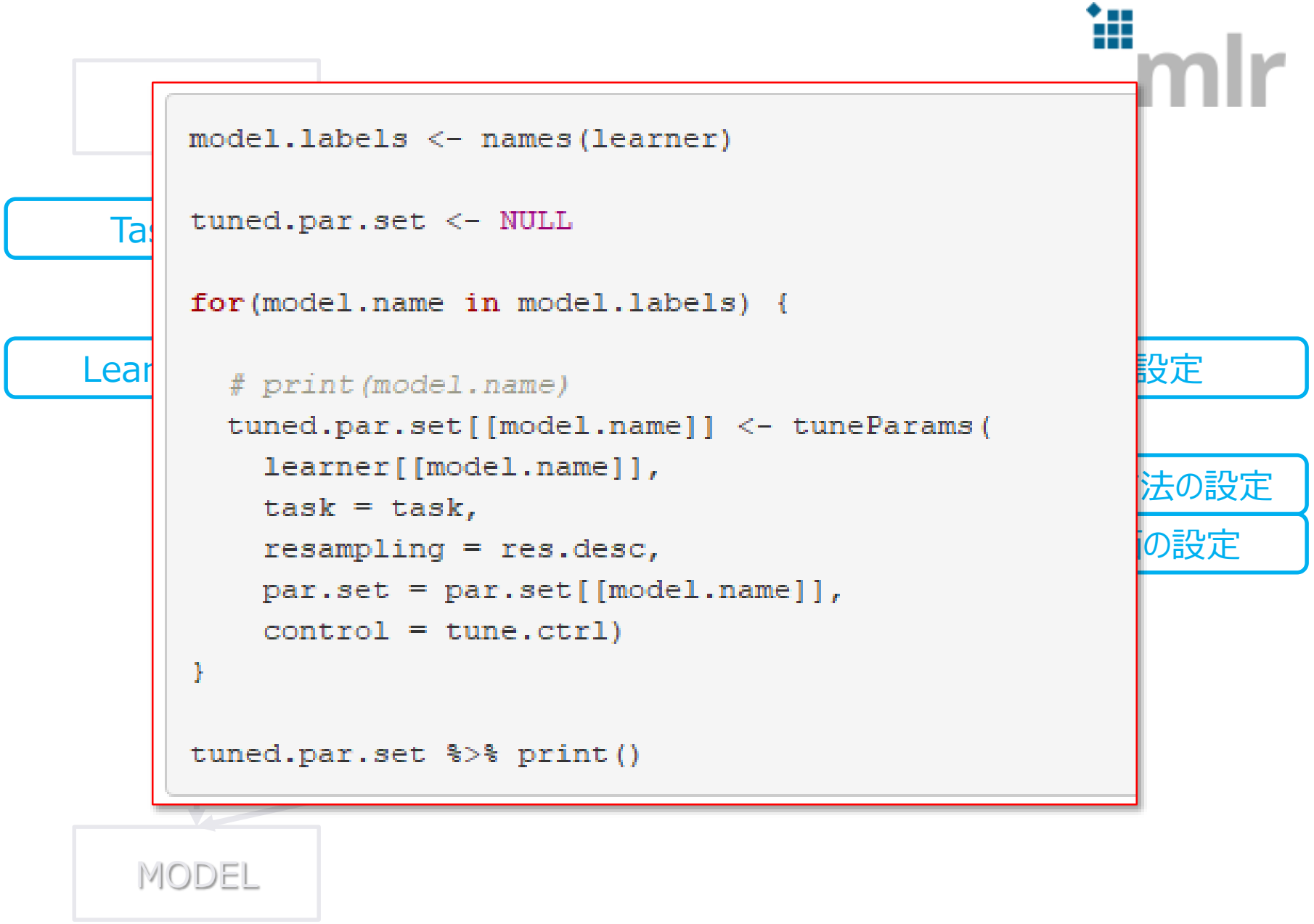
Gradient Boosting Machine (GBM)

```
# getLearnerParamSet(makeLearner("regr.gbm"))
learner[["gbm"]] <- makeLearner("regr.gbm")
par.set[["gbm"]] <- makeParamSet(
  makeIntegerParam("n.trees",           lower = 3L, upper = 50L),
  makeIntegerParam("interaction.depth", lower = 3L, upper = 20L))
```

ワークフロー： mlr


Data
Taskの定義
Learnerの選定
Learnerの定義
探索パラメータの設定
パラメータ探索方法の設定
探索結果の評価の設定
パラメータのチューニング
Learnerのパラメータ設定
MODEL

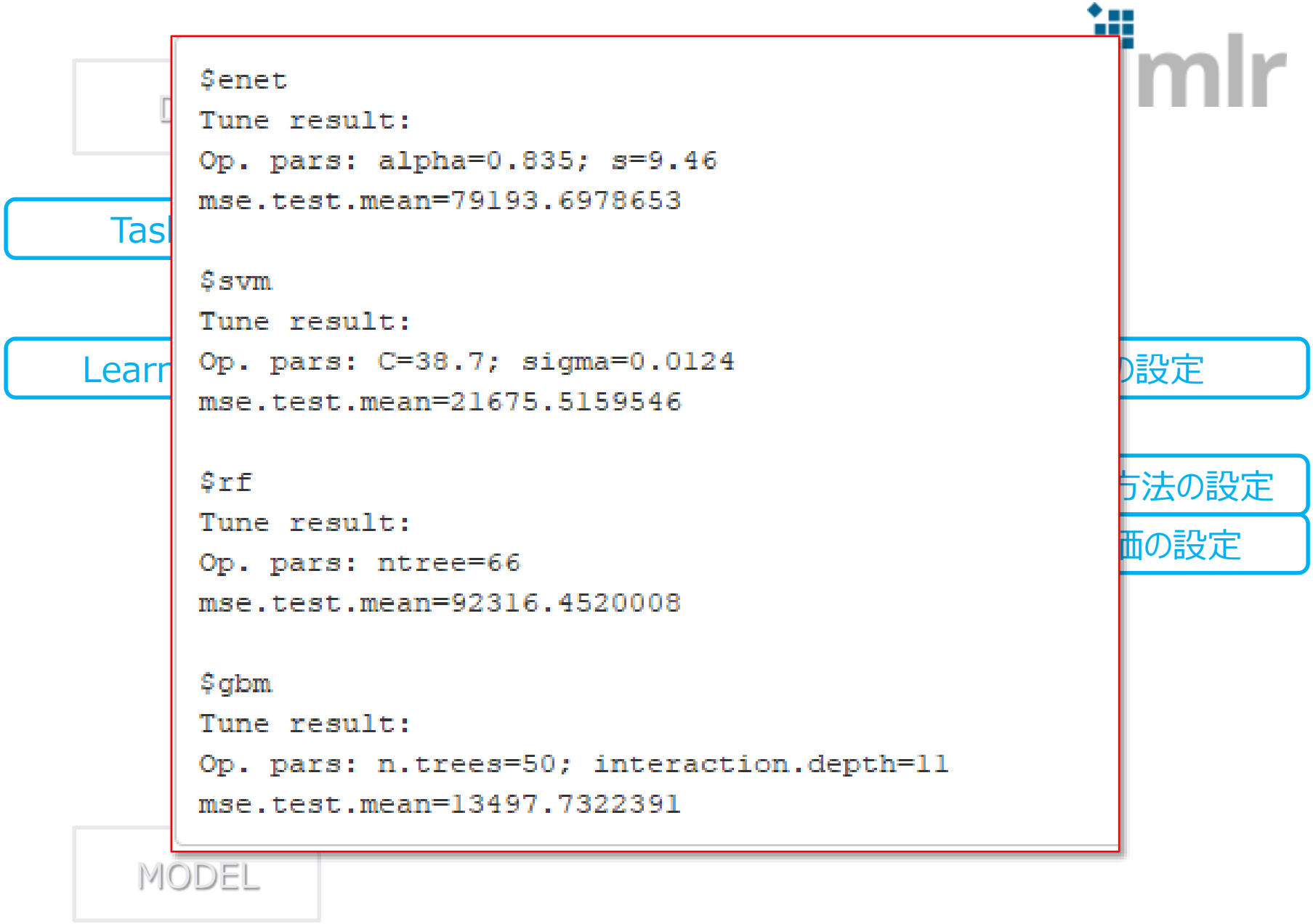
ワークフロー： mlr



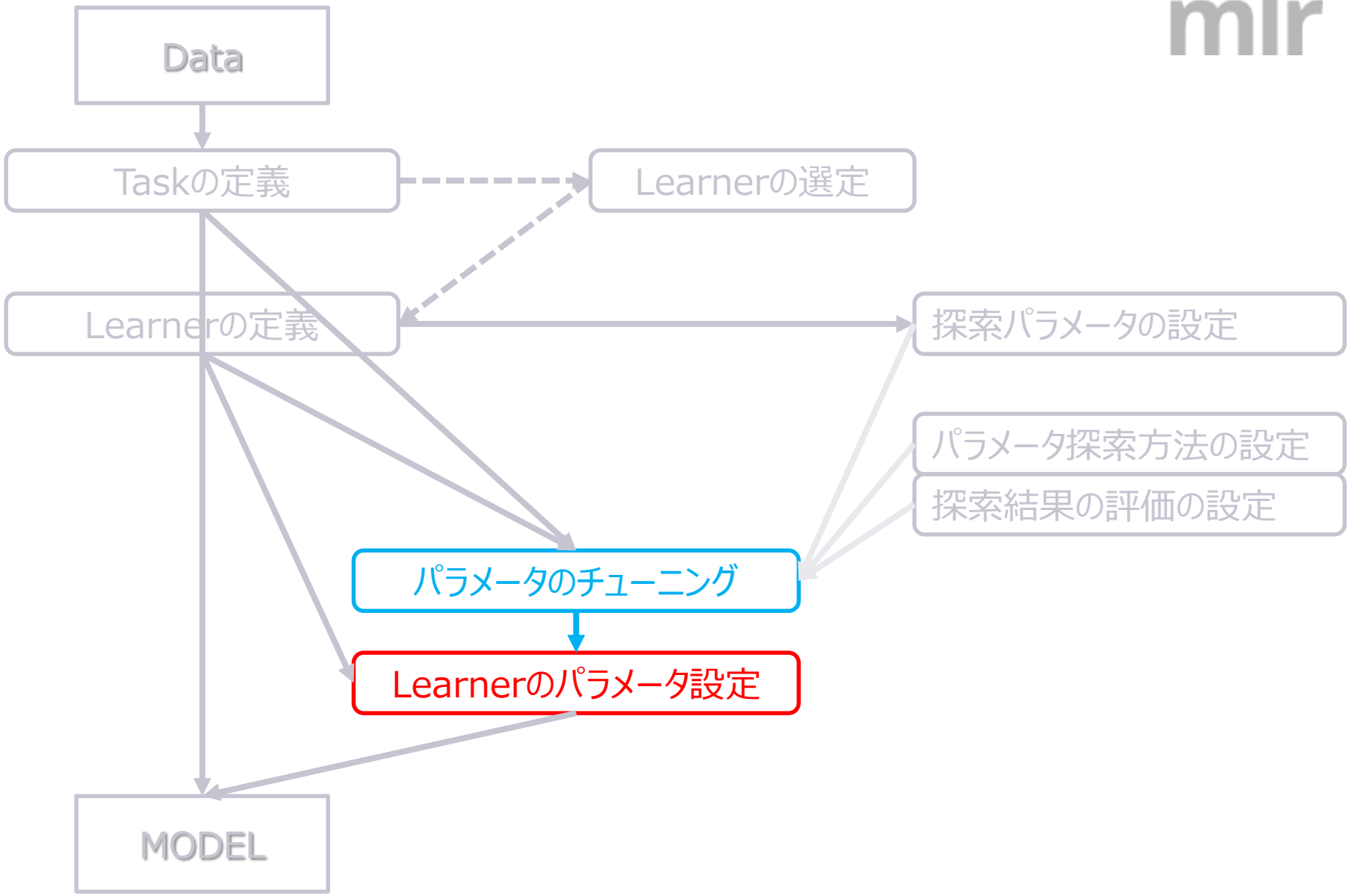

```
model.labels <- names(learner)

tuned.par.set <- NULL

for(model.name in model.labels) {

  # print(model.name)
  tuned.par.set[[model.name]] <- tuneParams(
    learner[[model.name]],
    task = task,
    resampling = res.desc,
    par.set = par.set[[model.name]],
    control = tune.ctrl)
}

tuned.par.set %>% print()
```

ワークフロー： mlr

```
$enet
Tune result:
Op. pars: alpha=0.835; s=9.46
mse.test.mean=79193.6978653

$svm
Tune result:
Op. pars: C=38.7; sigma=0.0124
mse.test.mean=21675.5159546

$rf
Tune result:
Op. pars: ntree=66
mse.test.mean=92316.4520008

$gbm
Tune result:
Op. pars: n.trees=50; interaction.depth=11
mse.test.mean=13497.7322391
```



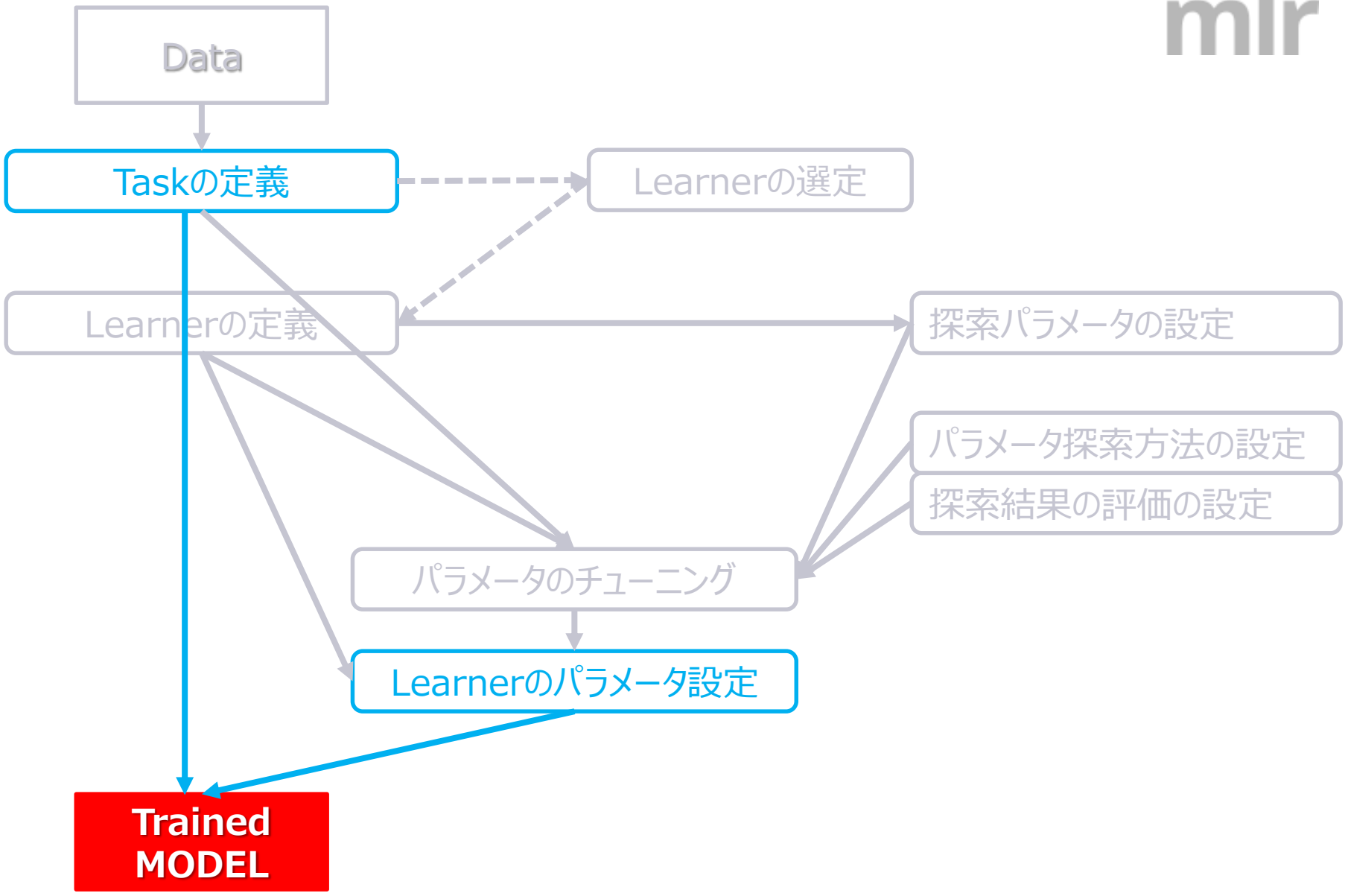
ワークフロー： mlr


Data
Taskの定義
Learnerの選定
Learnerの定義
探索パラメータの設定
パラメータ探索方法の設定
探索結果の評価の設定
パラメータのチューニング
Learnerのパラメータ設定
MODEL

ワークフロー： mlr


Data
tuned.learner <- list()
for(model.name in model.labels) {
# print(model.name)
tuned.learner[[model.name]] <- setHyperPars(
learner = learner[[model.name]],
par.vals = tuned.par.set[[model.name]]$x
)
}
MODEL

ワークフロー： mlr


Data
Taskの定義
Learnerの選定
Learnerの定義
探索パラメータの設定
パラメータ探索方法の設定
探索結果の評価の設定
パラメータのチューニング
Learnerのパラメータ設定
Trained MODEL

```
tuned.learner %>% print()
```

```
$enet
Learner regr.glmnet from package glmnet
Type: regr
Name: GLM with Lasso or Elasticnet Regularization; Short name: glmnet
Class: regr.glmnet
Properties: numerics,factors,ordered,weights
Predict-Type: response
Hyperparameters: s=9.46,alpha=0.835


$svm
Learner regr.ksvm from package kernlab
Type: regr
Name: Support Vector Machines; Short name: ksvm
Class: regr.ksvm
Properties: numerics,factors
Predict-Type: response
Hyperparameters: fit=FALSE,kernel=rbfdot,C=38.7,sigma=0.0124


$rf
Learner regr.randomForest from package randomForest
Type: regr
Name: Random Forest; Short name: rf
Class: regr.randomForest
Properties: numerics,factors,ordered,se,oobpreds,featimp
Predict-Type: response
Hyperparameters: ntree=66


$gbm
Learner regr.gbm from package gbm
Type: regr
Name: Gradient Boosting Machine; Short name: gbm
Class: regr.gbm
Properties: missings,numerics,factors,weights,featimp
Predict-Type: response
Hyperparameters: distribution=gaussian,keep.data=FALSE,n.trees=50,interaction.depth=11
```

ワークフロー： mlr

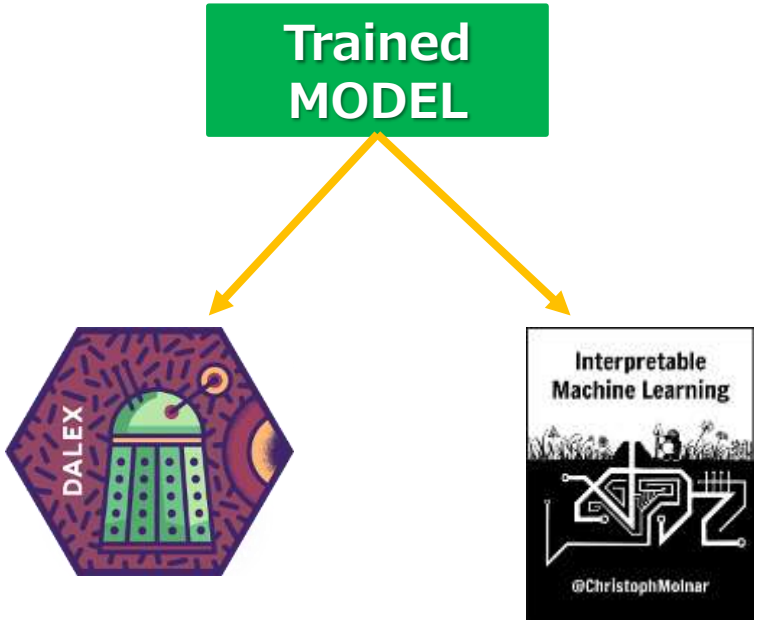
```
tuned.model   <- NULL

for(model.name in model.labels) {
  tuned.model[[model.name]] <- train(tuned.learner[[model.name]], task)
}

tuned.model %>% print()
```

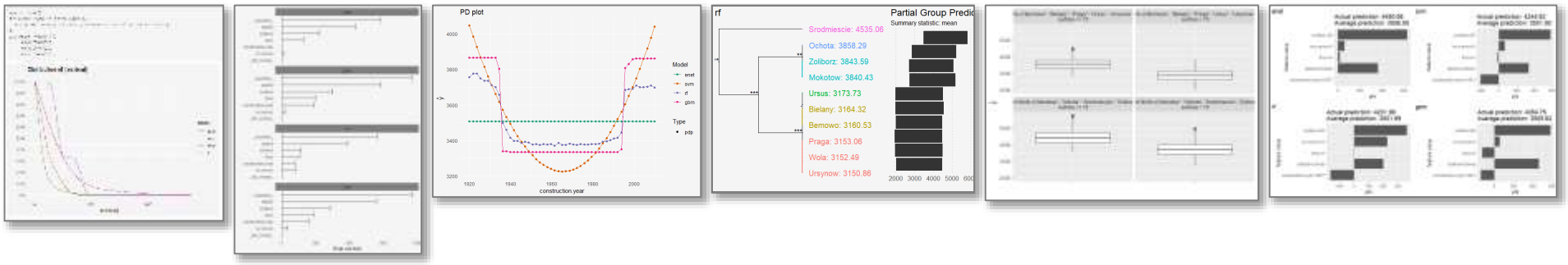
```
$enet
Model for learner.id=regr.glmnet; learner.class=regr.glmnet
Trained on: task.id = ap; obs = 1000; features = 5
Hyperparameters: s=9.46,alpha=0.835

$svm
Model for learner.id=regr.ksvm; learner.class=regr.ksvm
Trained on: task.id = ap; obs = 1000; features = 5
Hyperparameters: fit=FALSE,kernel=rbfdot,C=38.7,sigma=0.0124

$rf
Model for learner.id=regr.randomForest; learner.class=regr.randomForest
Trained on: task.id = ap; obs = 1000; features = 5
Hyperparameters: ntree=66

$gbm
Model for learner.id=regr.gbm; learner.class=regr.gbm
Trained on: task.id = ap; obs = 1000; features = 5
Hyperparameters: distribution=gaussian,keep.data=FALSE,n.trees=50,interaction.depth=11
```

説明のアプローチ

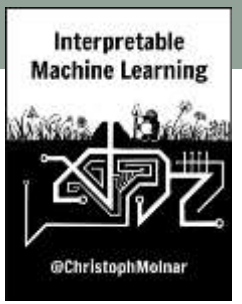
0. 説明の準備

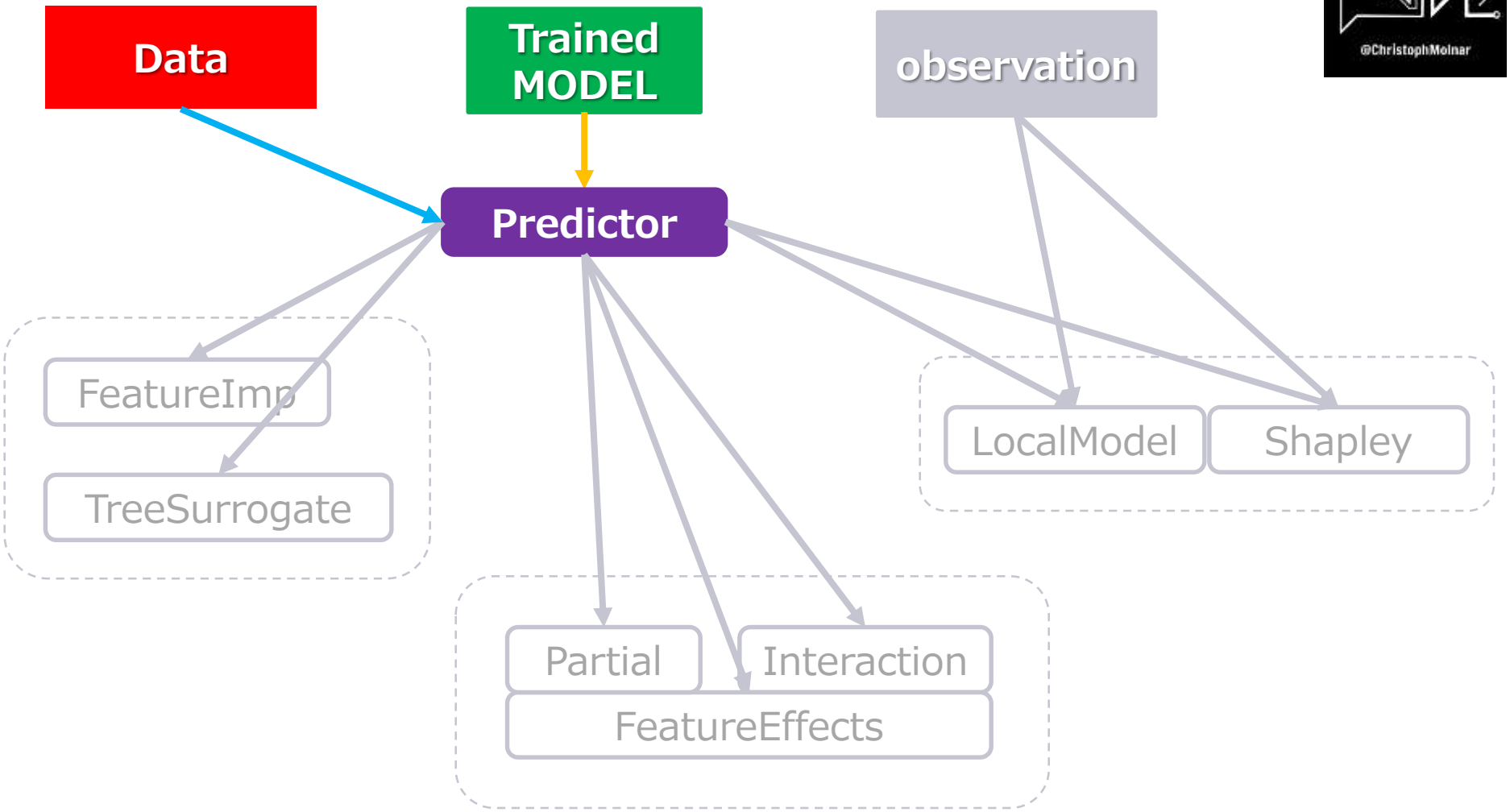
1. モデルの性能や特性の評価
2. 特徴量（変数）に対するモデルの応答をみる
3. あるデータに対する予測がどのように得られたか説明する





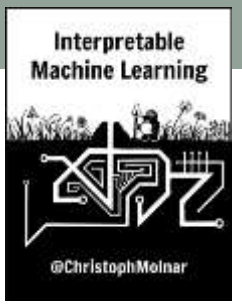
ワークフロー： iml

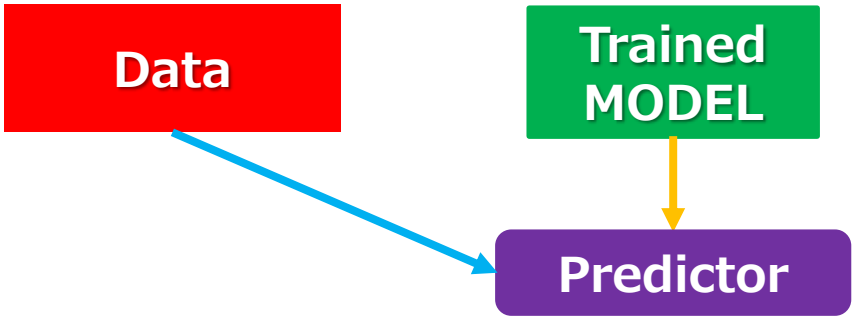




https://github.com/christophM/iml

準備： iml



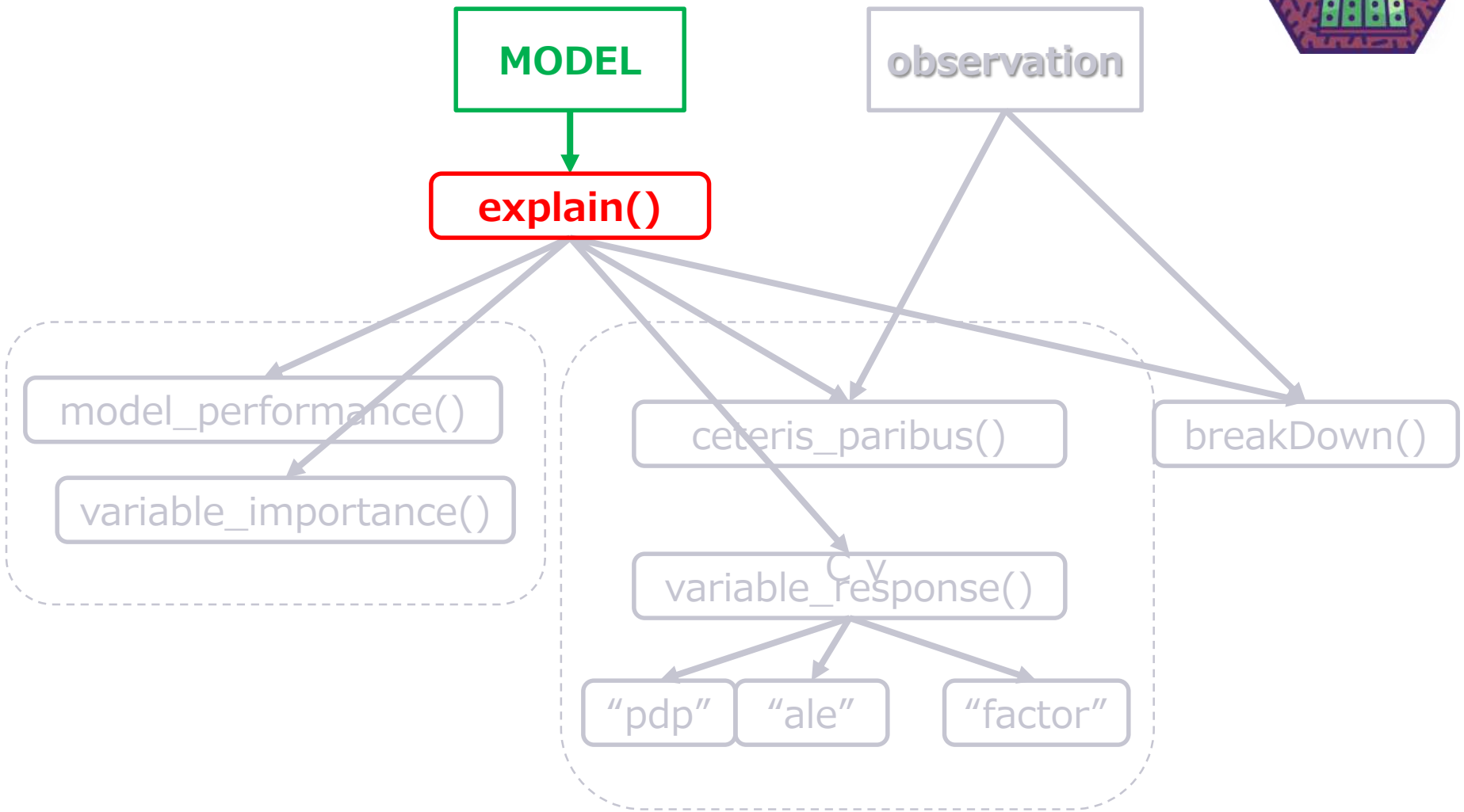



```
require("iml")
```

```
data("apartmentsTest", package = "DALEX")
X = apartmentsTest %>% select(-m2.price)
Y = apartmentsTest$m2.price

predictor.rf <- Predictor$new(tuned.model[["rf"]], data = X, y = Y)
```

https://github.com/christophM/iml

学習したモデルの評価と比較： DALEX（準備）



https://pbiecek.github.io/DALEX_docs/2-architecture.html#architecture

準備： DALEX

MODEL

observation

explain()

```
model.regr.rf <- tuned.model[["rf"]]
```

variable_importance()

```
explainer.rf <-  explain(model = model.regr.rf,
                         label = "rf",
                         data  = apartmentsTest %>% select(-m2.price),
                         y     = apartmentsTest$m2.price,
                         predict_function = predictMLR)
```

https://pbiecek.github.io/DALEX_docs/2-architecture.html#architecture

準備： DALEX

MODEL

observation

mlrの予測結果から**予測値を抜き出す**関数を定義する

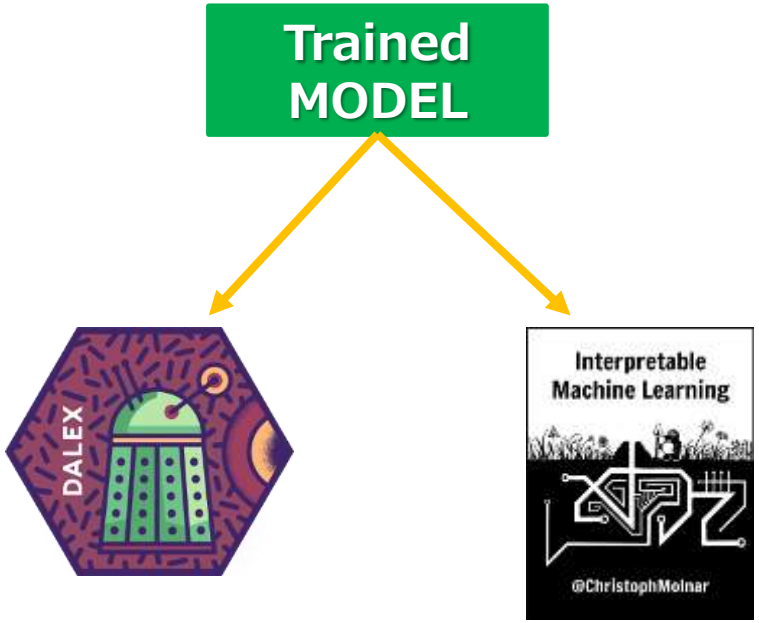
```
predictMLR <- function(object, newdata) {
  pred <- predict(object, newdata=newdata)
  response <- pred$data$response
  return(response)
}
```

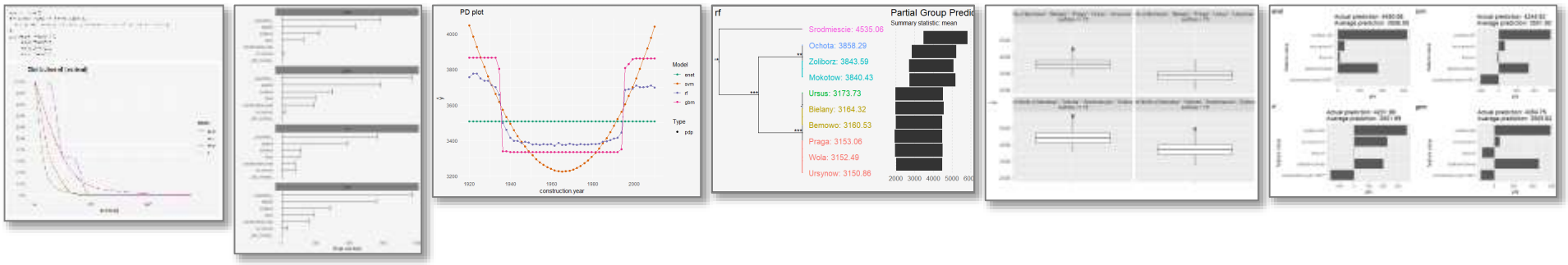
```
explainer.rf <- explain(model = model.r
                        label = "rf",
                        data  = apartmentsTe      % select(-m2.price),
                        y     = apartmentsTest    2.price,
                        predict_function = predictMLR)
```

https://pbiecek.github.io/DALEX_docs/2-architecture.html#architecture

説明のアプローチ

1. **モデルの性能や特性の評価**
2. 特徴量（変数）に対するモデルの応答をみる
3. あるデータに対する予測がどのように得られたか説明する

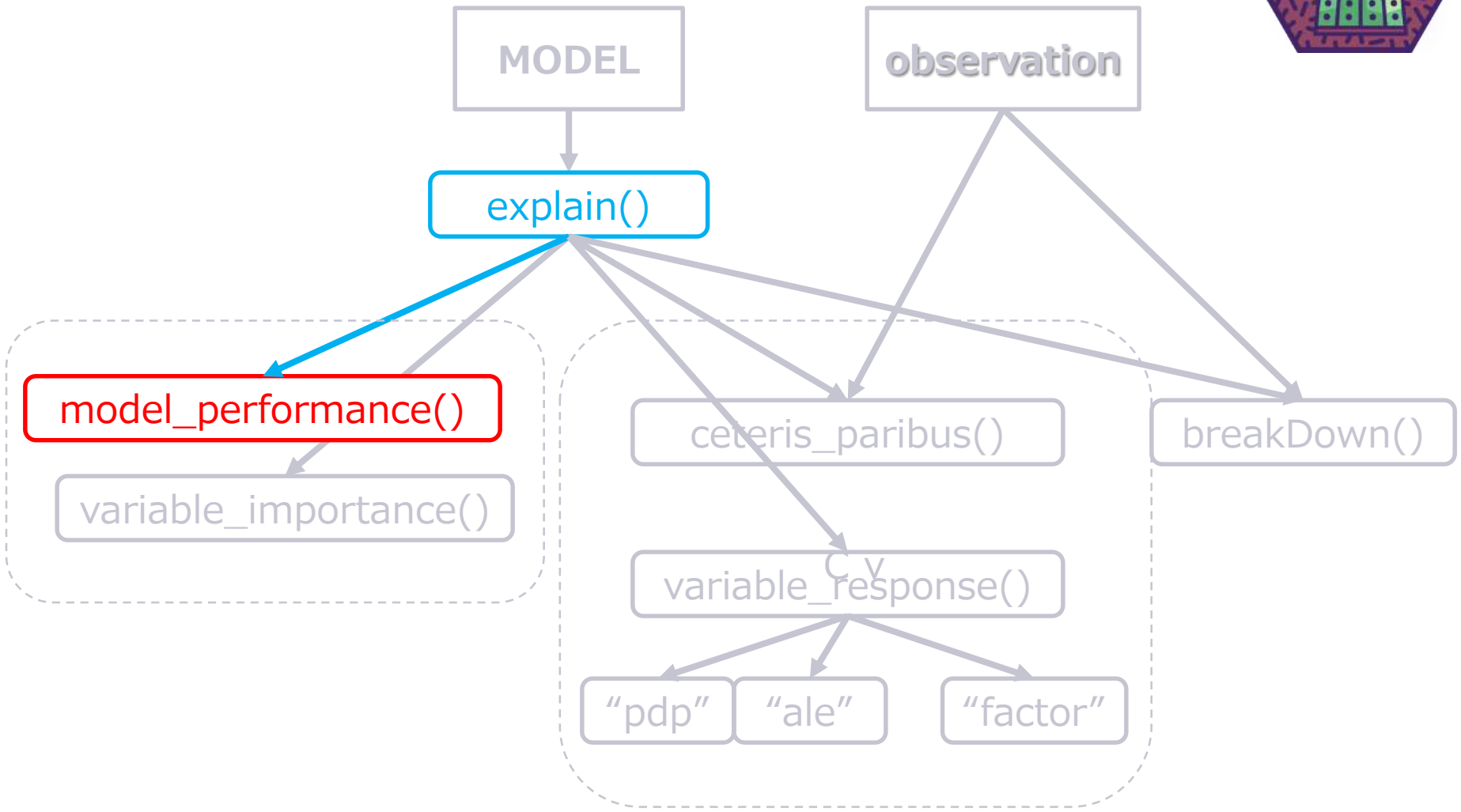




Residuals & goodness-of-fit

policy	method name	iml	DALEX
understand entire model	residuals and goodness of fit	✗	✓
	permutation importance	✓	✓
	global surrogate	Tree surrogate	✗
understand feature(s)	Merging Path Plot (PDP for categorical data)	✗	✓
	Partial Dependence Plot (PDP for continuous data)	✓	✓
	Individual Conditional Expectation (ICE)	✓	Ceteris Paribus Plots
	Accumulated Local Effects (ALE) Plot	✓	✓
	Feature Interaction	✓	✗
local interpretation (for single prediction)	LIME	✓	✗
	SHAPLY value	✓	✗
	breakDown	✗	✓

Residuals & goodness-of-fit



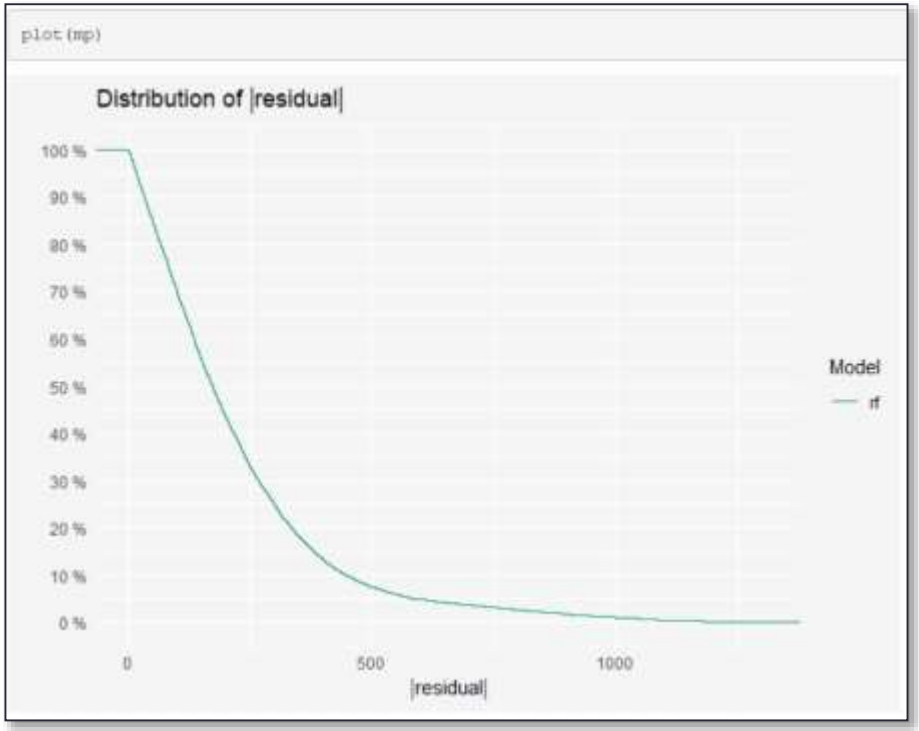
https://pbiecek.github.io/DALEX_docs/2-architecture.html#architecture

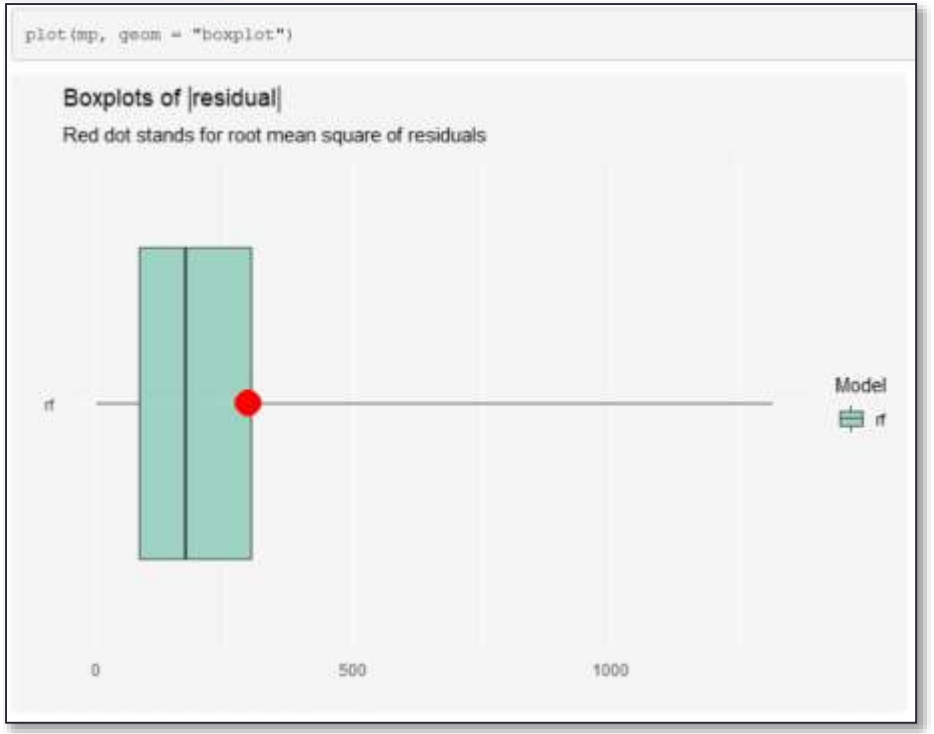
予測残差の大きさの頻度分布を観察する

```
mp <- model_performance(explainer.rf)
str(mp)
```

```
Classes 'model_performance_explainer' and 'data.frame': 9000 obs. of  4 variables:
 $ predicted: num  4187 3319 2744 2682 2904 ...
 $ observed : num  4644 3082 2498 2735 2781 ...
 $ diff     : num  -456.7 237.2 246.2 -53.2 123.3 ...
 $ label    : chr  "rf" "rf" "rf" "rf" ...
```

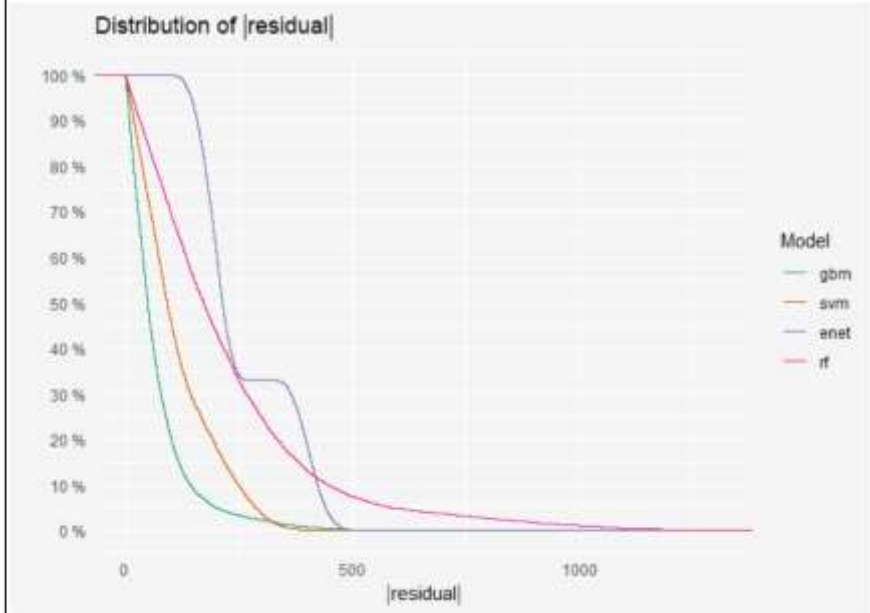
```
plot(mp)
```



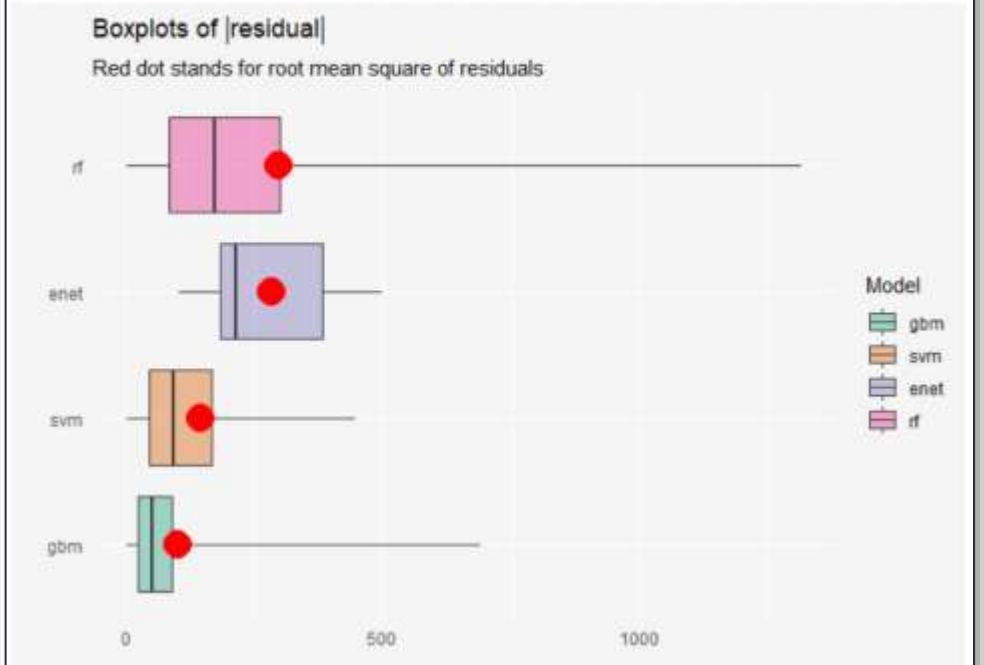


モデルの比較（当てはまりの良さ）

```
mps <- list()
for(model.name in model.labels){
  mps[[model.name]] <- model_performance(explainer[[model.name]])
}
plot(mps[["enet"]],
     mps[["svm"]],
     mps[["rf"]],
     mps[["gbm"]])
```

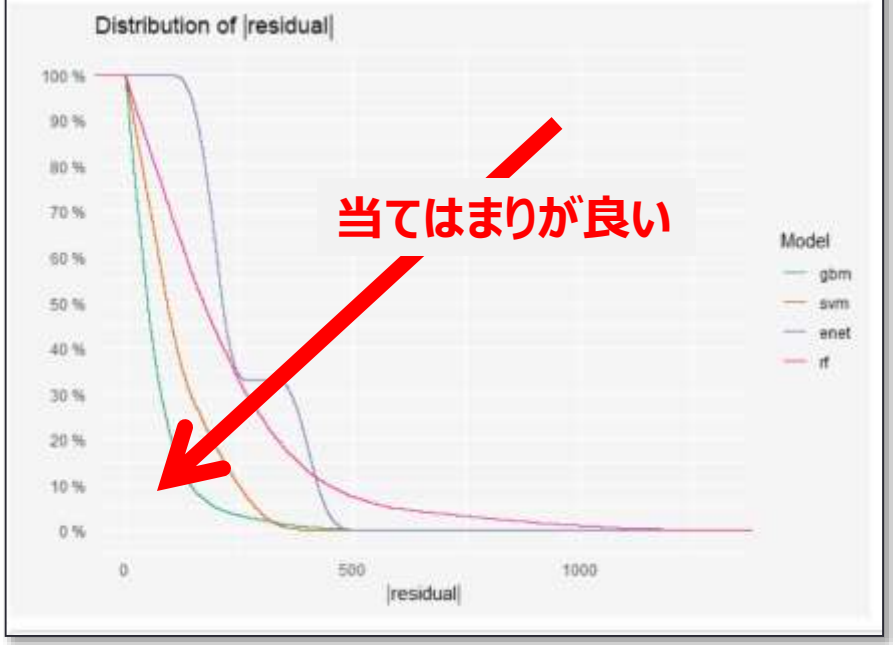



```
plot(geom = "boxplot",
     mps[["enet"]],
     mps[["svm"]],
     mps[["rf"]],
     mps[["gbm"]])
```



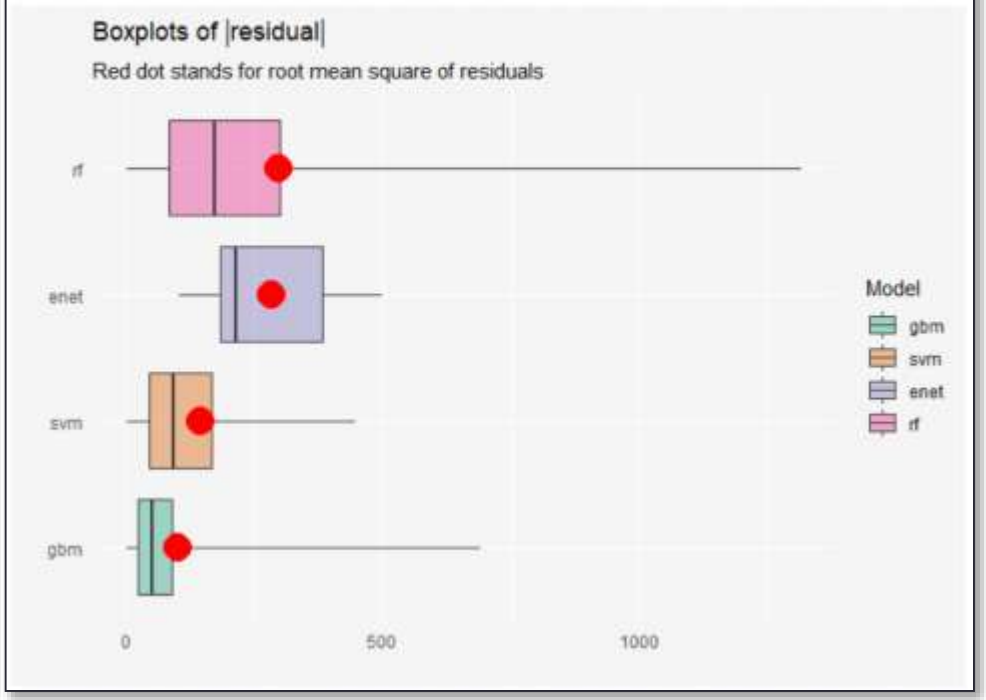
Empirical Cumulative Distribution Function (ecdf) of residual error

```
mps <- list()
for(model.name in model.labels){
  mps[[model.name]] <- model_performance(explainer[[model.name]])
}
plot(mps[["enet"]],
     mps[["svm"]],
     mps[["rf"]],
     mps[["gbm"]])
```



当てはまりが良い

```
plot(geom = "boxplot",
     mps[["enet"]],
     mps[["svm"]],
     mps[["rf"]],
     mps[["gbm"]])
```



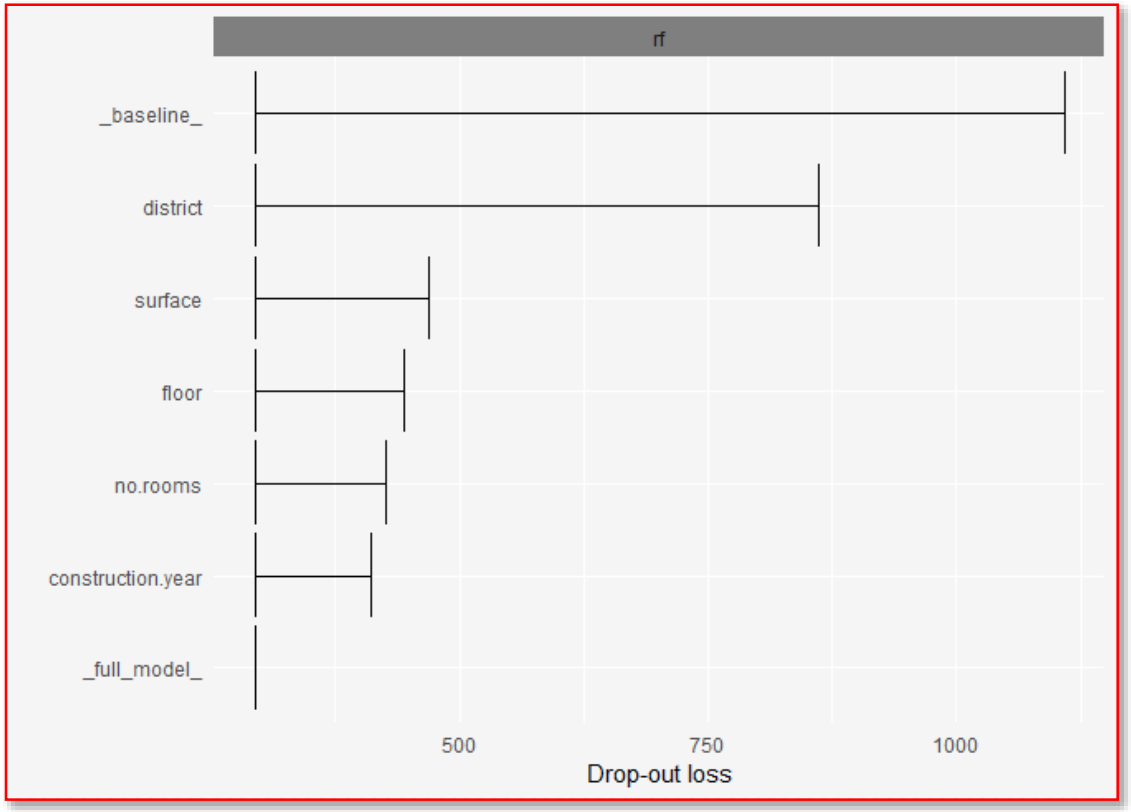
Permutation importance

policy	method name	iml	DALEX
understand entire model	residuals and goodness of fit	✗	✓
	permutation importance	✓	✓
	global surrogate	Tree surrogate	✗
understand feature(s)	Merging Path Plot (PDP for categorical data)	✗	✓
	Partial Dependence Plot (PDP for continuous data)	✓	✓
	Individual Conditional Expectation (ICE)	✓	Ceteris Paribus Plots
	Accumulated Local Effects (ALE) Plot	✓	✓
	Feature Interaction	✓	✗
local interpretation (for single prediction)	LIME	✓	✗
	SHAPLY value	✓	✗
	breakDown	✗	✓

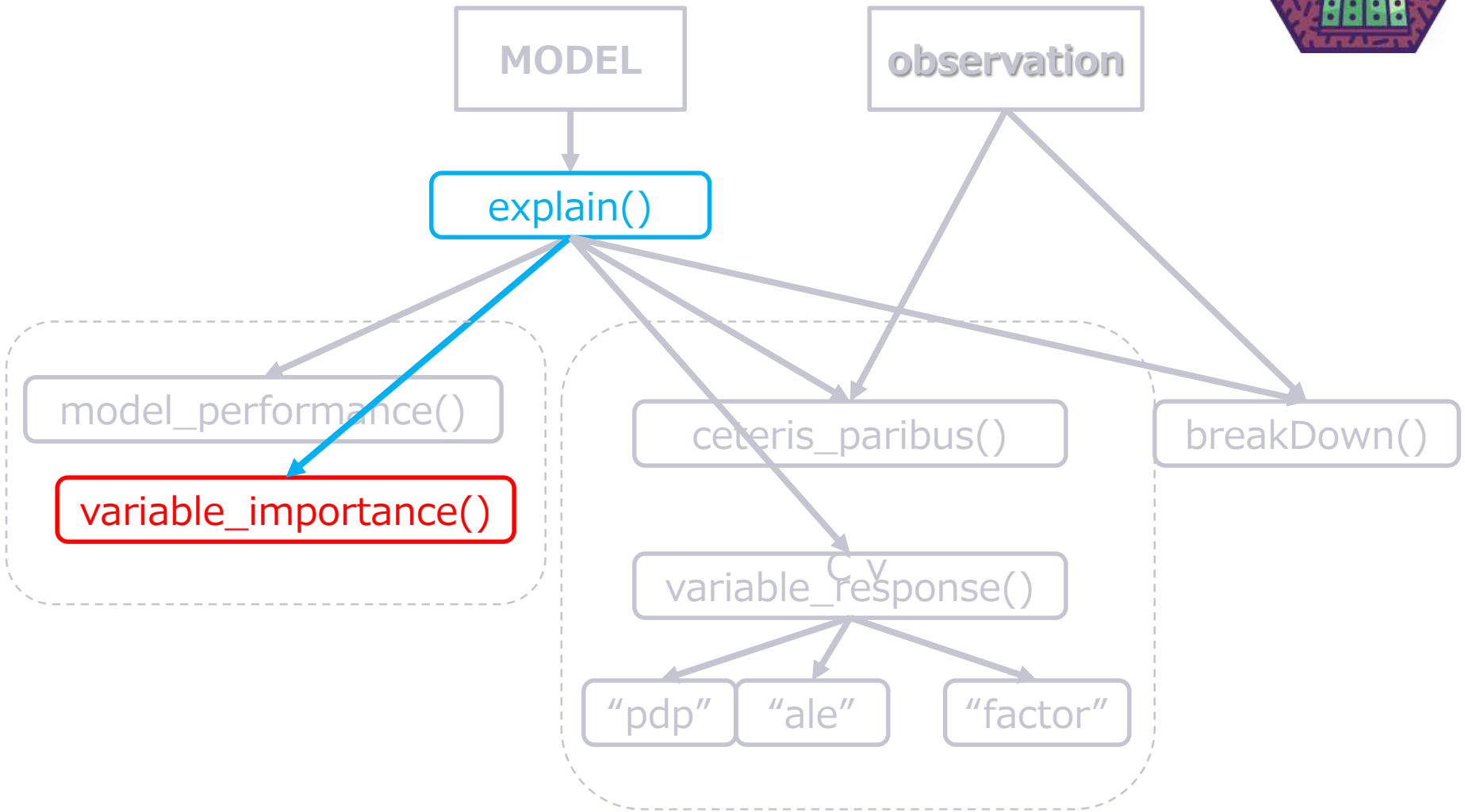
Permutation importance

- ある変数列の値をシャッフルした時にすると
  予測精度が大きく低下する＝重要度が高い

```
var.imp <- variable_importance(explainer     = explainer.rf,
                               loss_function = loss_root_mean_square)
plot(var.imp)
```



Permutation importance



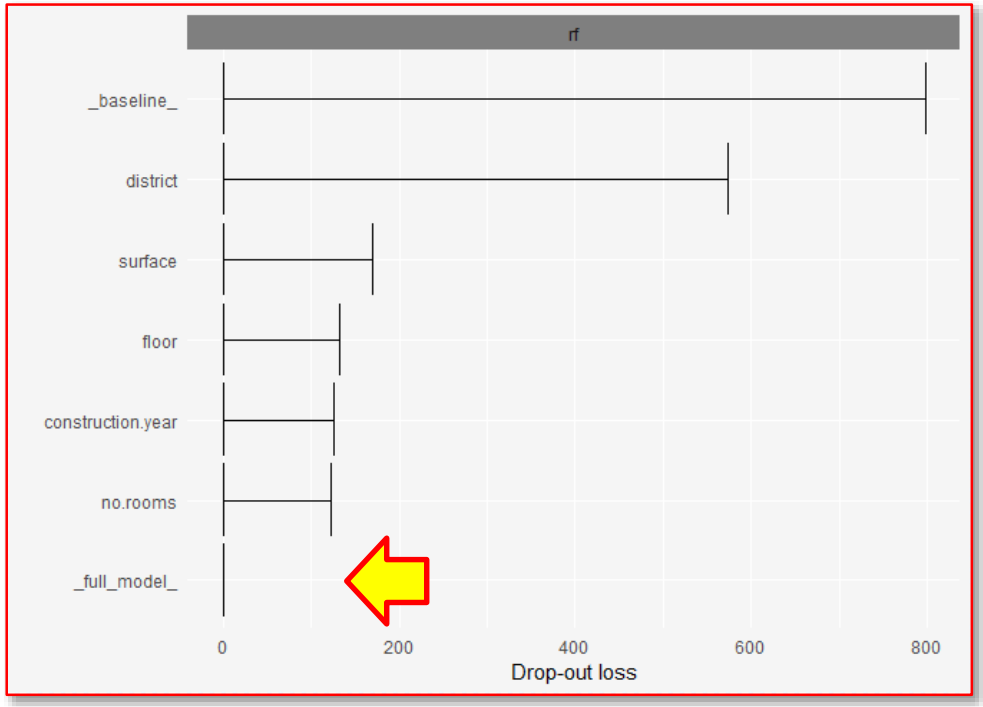
https://pbiecek.github.io/DALEX_docs/2-architecture.html#architecture

variable_importance()

- type = “difference“を指定すると、フルモデルからの相対値（drop_loss - drop_loss_full_model）で表示する

→ モデル同士の比較用

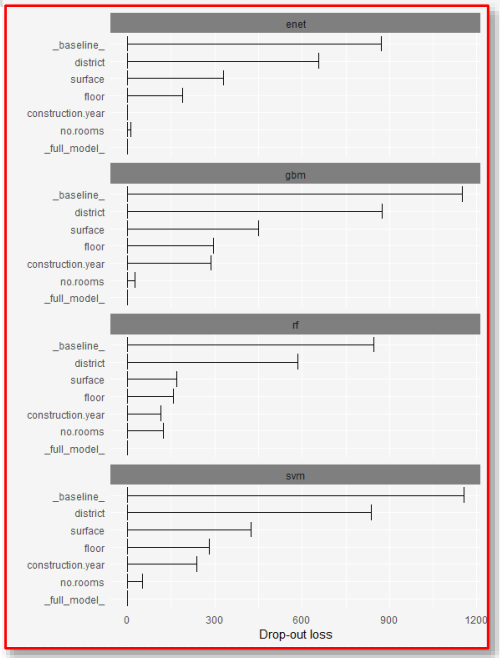
```
var.imp.diff <- variable_importance(explainer     = explainer.rf,
                                    loss_function = loss_root_mean_square,
                                    type          = "difference")

plot(var.imp.diff)
```



variable_importance()

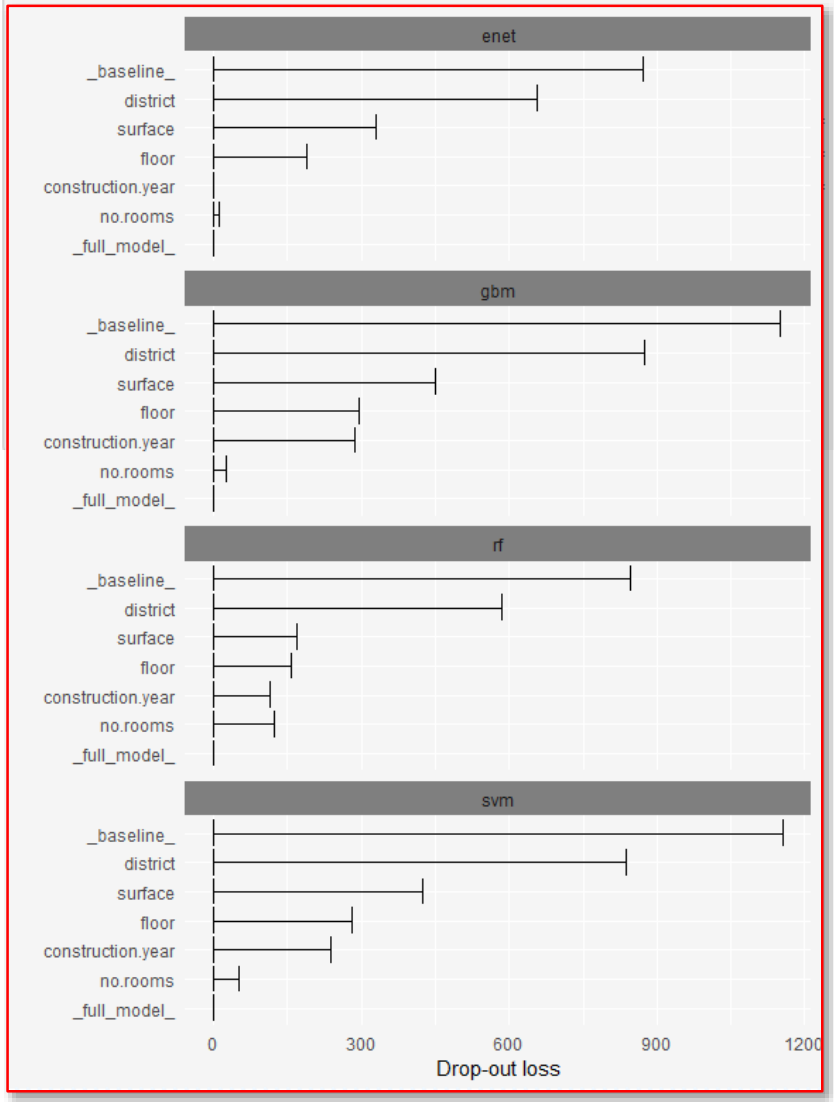
- データのシャッフルに対する予測精度のロスを評価するため、すべての予測モデルを同じ尺度で評価できる

```
vid <- list()
pvid <- list()
for(model.name in model.labels){
  vid[[model.name]] <- variable_importance(explainer     = explainer[[model.name]],
                                           loss_function = loss_root_mean_square,
                                           type          = "difference")
  pvid[[model.name]] <- plot(vid[[model.name]])
}

plot(vid[["enet"]],
     vid[["svm"]],
     vid[["rf"]],
     vid[["gbm"]])
```



variable_importance()

- データのシャッフルに対する予測精度のロスを評価するため、すべての予測モデルを同じ尺度で評価できる




```
explainer[[model.name]],
loss_root_mean_square,
"difference")
```

Permutation importance

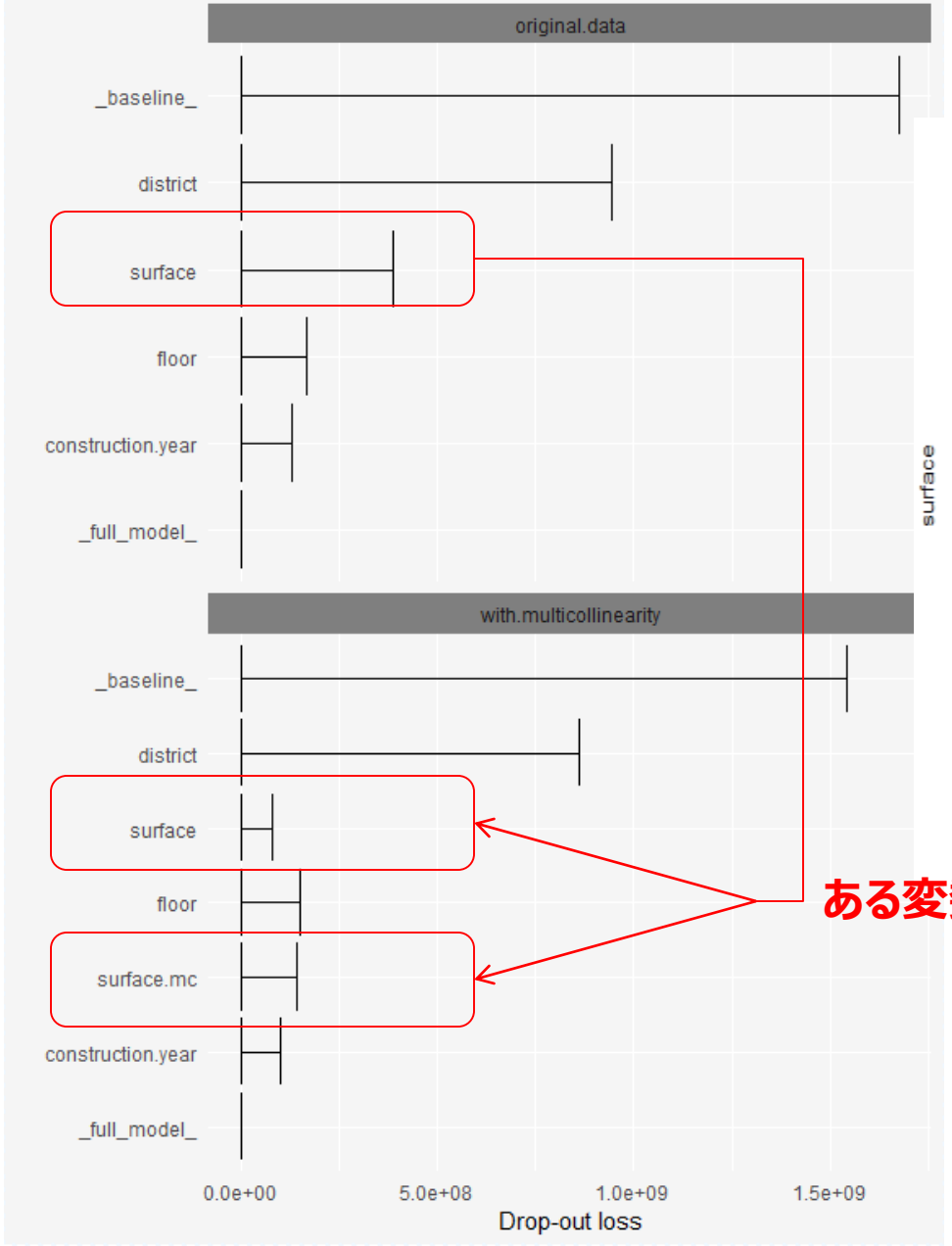
Advantage

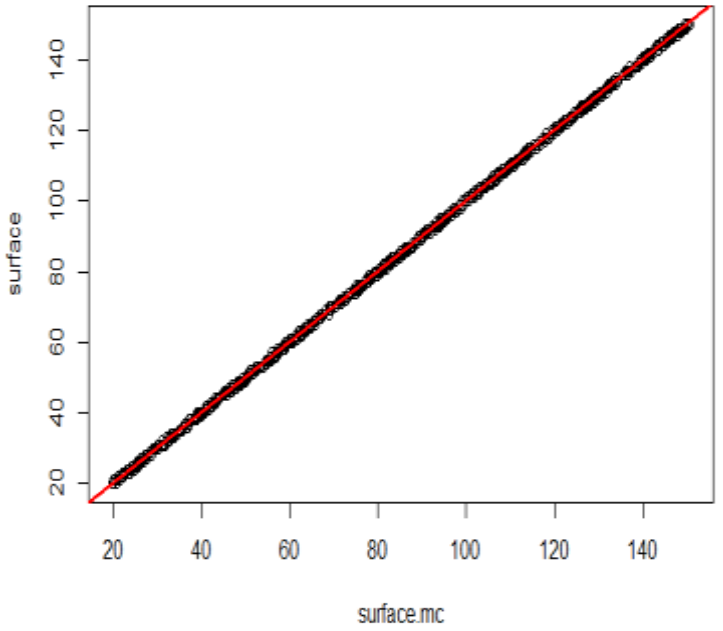
- 解釈しやすい
  - ターゲット特徴量の情報を壊すと予測性能が下がる
- モデルの再学習を必要としない
  - 予測対象のデータを操作するだけで良い

Disadvantage

- 訓練データで計算するべき？テストデータを使用するべき？
  - よくわかっていない。
- Permutationによるランダム性（値の「ゆれ」が生じる）
  - 安定化させるためには繰り返しをとって平均すればよいが、計算コストがかかる
- 特徴量の間に強い相互作用がある場合、重要度にバイアスが発生する
- 相関の高い変数間で重要度が「分割」される
  - 相関の高い別の特徴が、ある変数のシャッフルに対して補完的に振る舞うため

相関の高い変数間で重要度が「分割」される
相関の高い別の特徴が、ある変数のシャッフルに対して補完的に振る舞うため

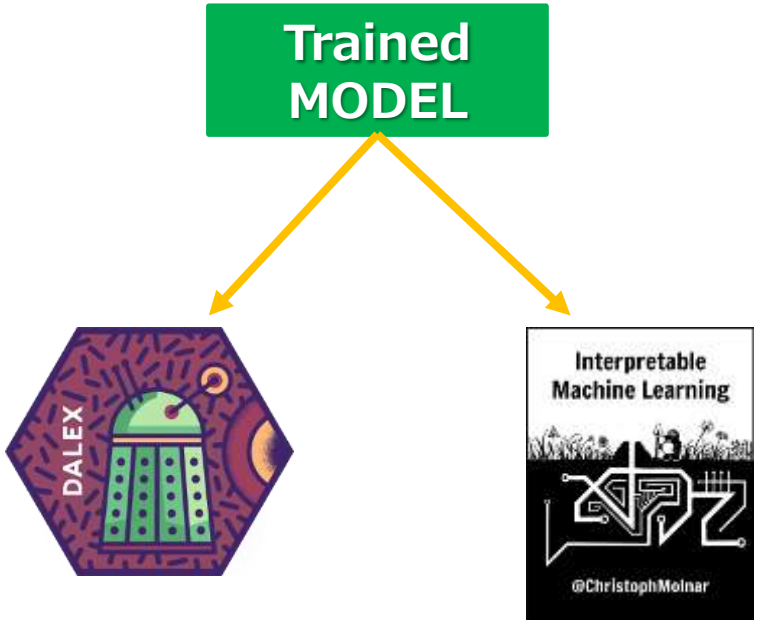


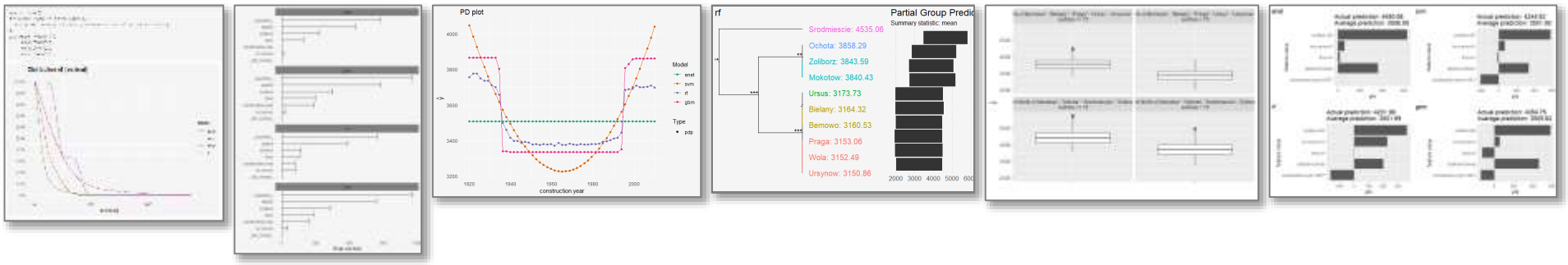


ある変数と相関の高い別の特徴変数がある場合

説明のアプローチ

1. モデルの性能や特性の評価
2. **特徴量（変数）に対するモデルの応答をみる**
3. あるデータに対する予測がどのように得られたか説明する



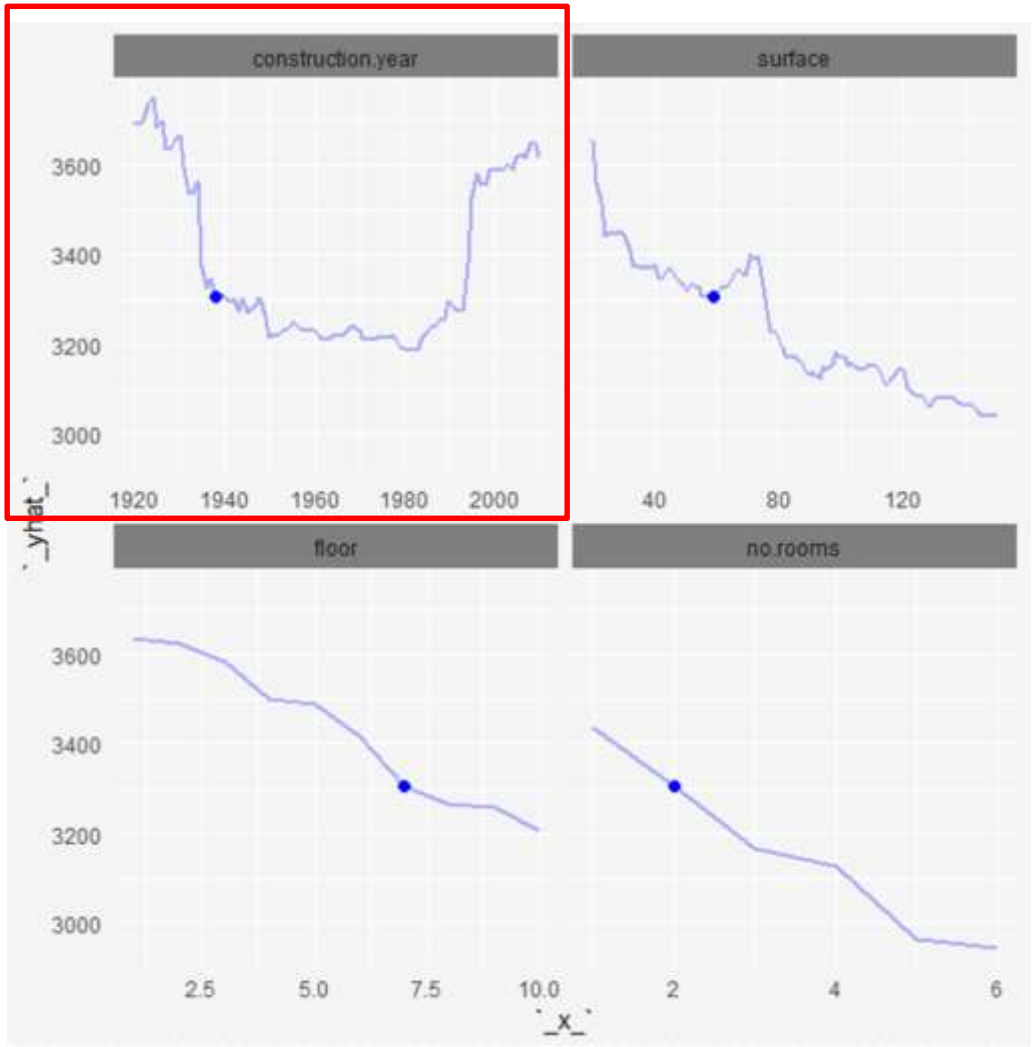


Partial Dependence Plot

policy	method name	iml	DALEX
understand entire model	residuals and goodness of fit	✗	✓
	permutation importance	✓	✓
	global surrogate	Tree surrogate	✗
understand feature(s)	Merging Path Plot (PDP for categorical data)	✗	✓
	Partial Dependence Plot (PDP for continuous data)	✓	✓
	Individual Conditional Expectation (ICE)	✓	Ceteris Paribus Plots
	Accumulated Local Effects (ALE) Plot	✓	✓
	Feature Interaction	✓	✗
local interpretation (for single prediction)	LIME	✓	✗
	SHAPLY value	✓	✗
	breakDown	✗	✓

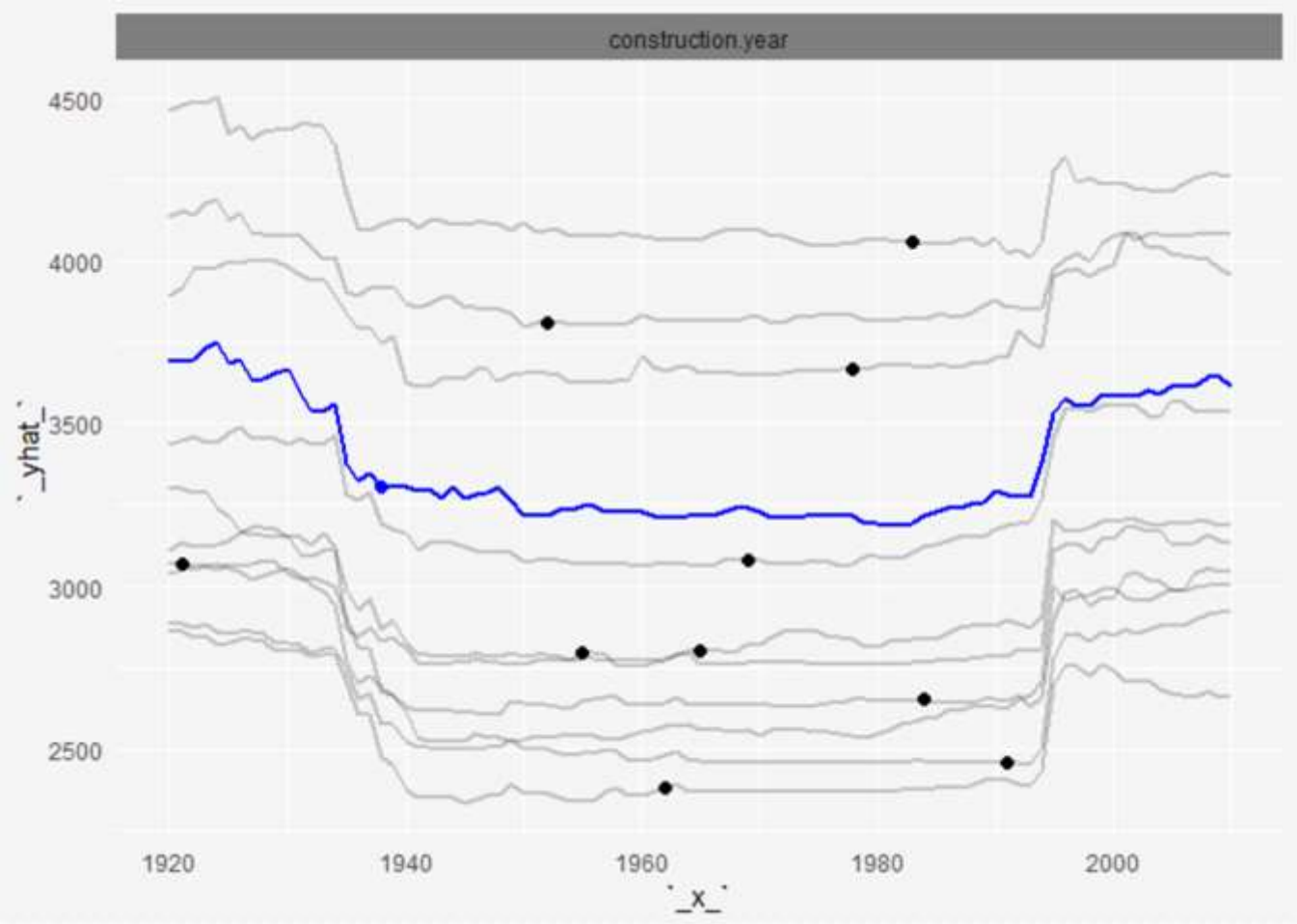
What-If (*ceteris paribus*) plot for single observation

ある観察に対して、**注目する変数以外はすべて同じ値**を持つ観察がたくさんあったとき、対象の変数の値の増減によって、予測値がどう変化するか？



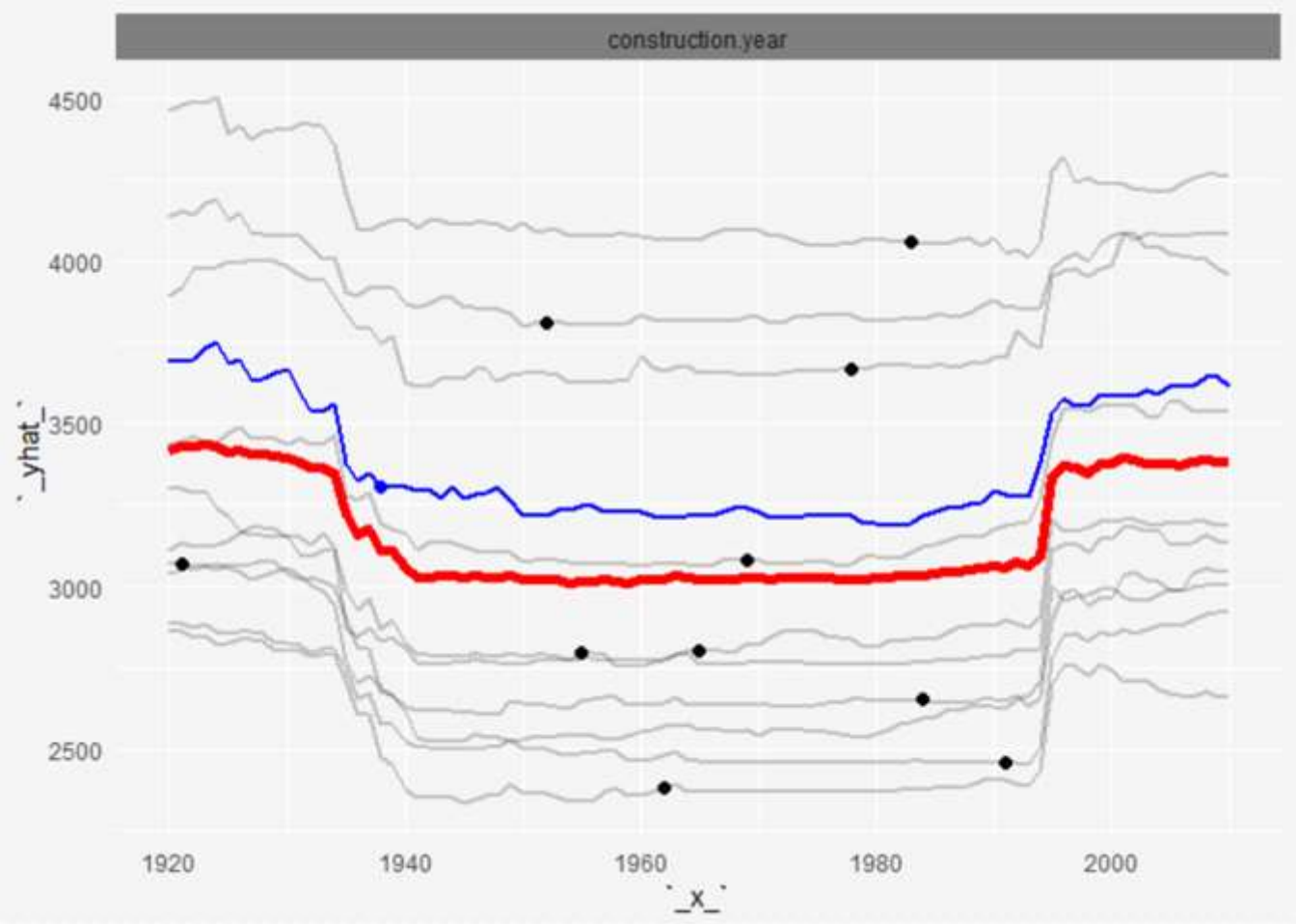
What-If plot for single feature + other observation (ICE)

ある観察の予測値の変化を観察したとき
ほかの観察の予測値はどう振る舞っているか？



What-If + other observation（ICE) + average line（PD)

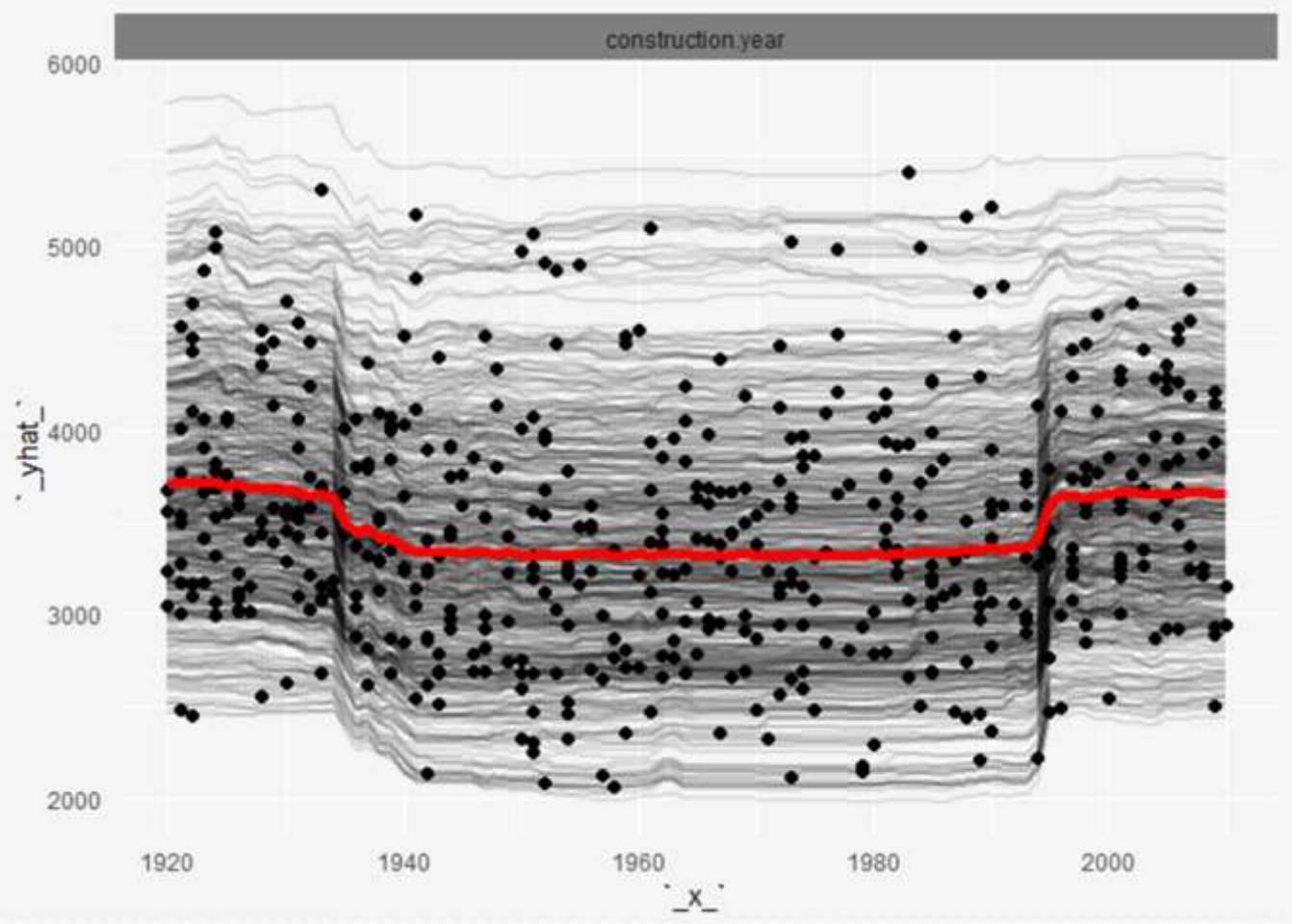
ある観察の予測値の変化を観察したとき
ほかの観察との平均的な振る舞いはどうなっているか？



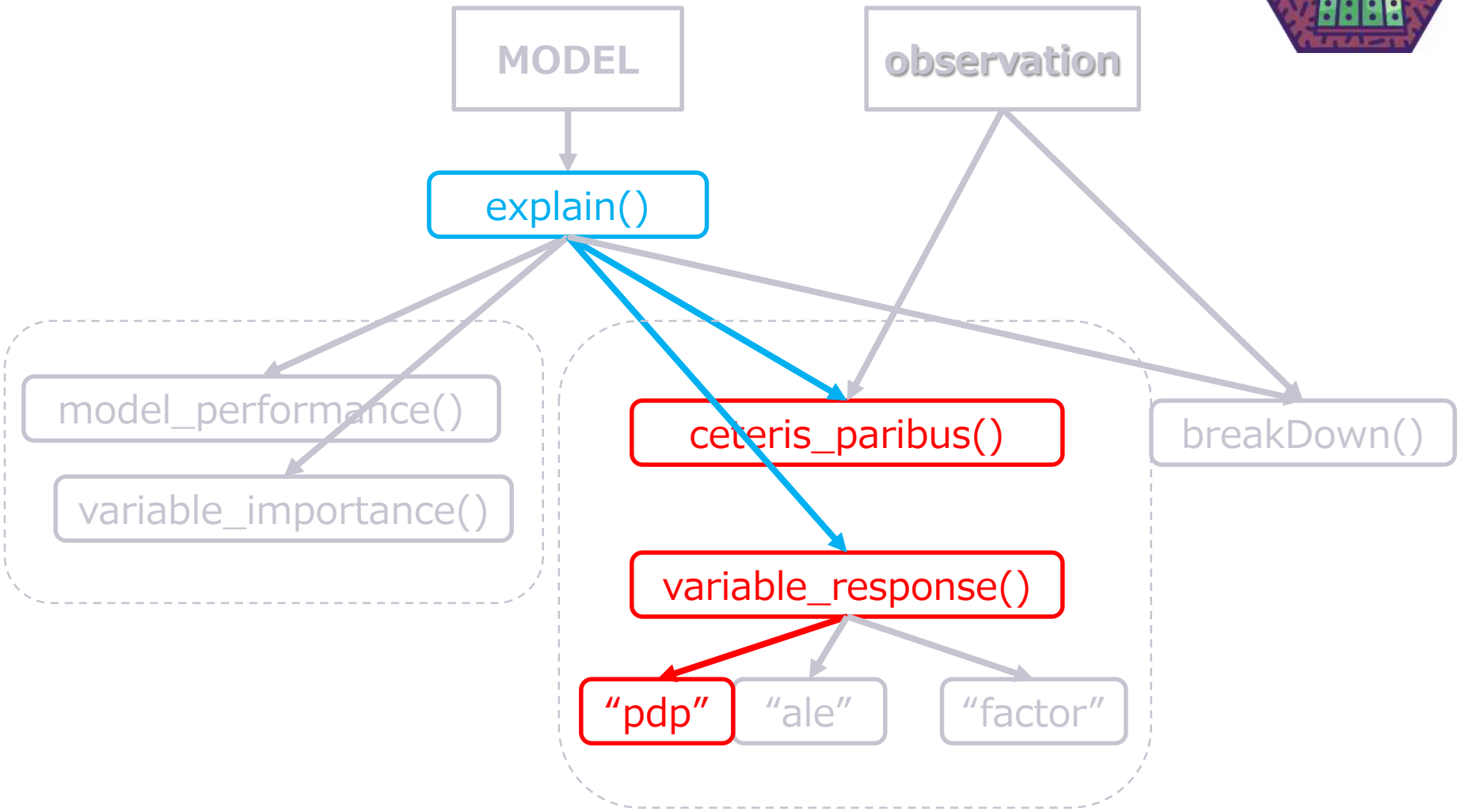
Partial Dependence (PD) + Individual Conditional Expectation (ICE)

全ての観察における平均的な振る舞いはどうなっているか？

全ての観察で、注目する変数について周辺平均を取る



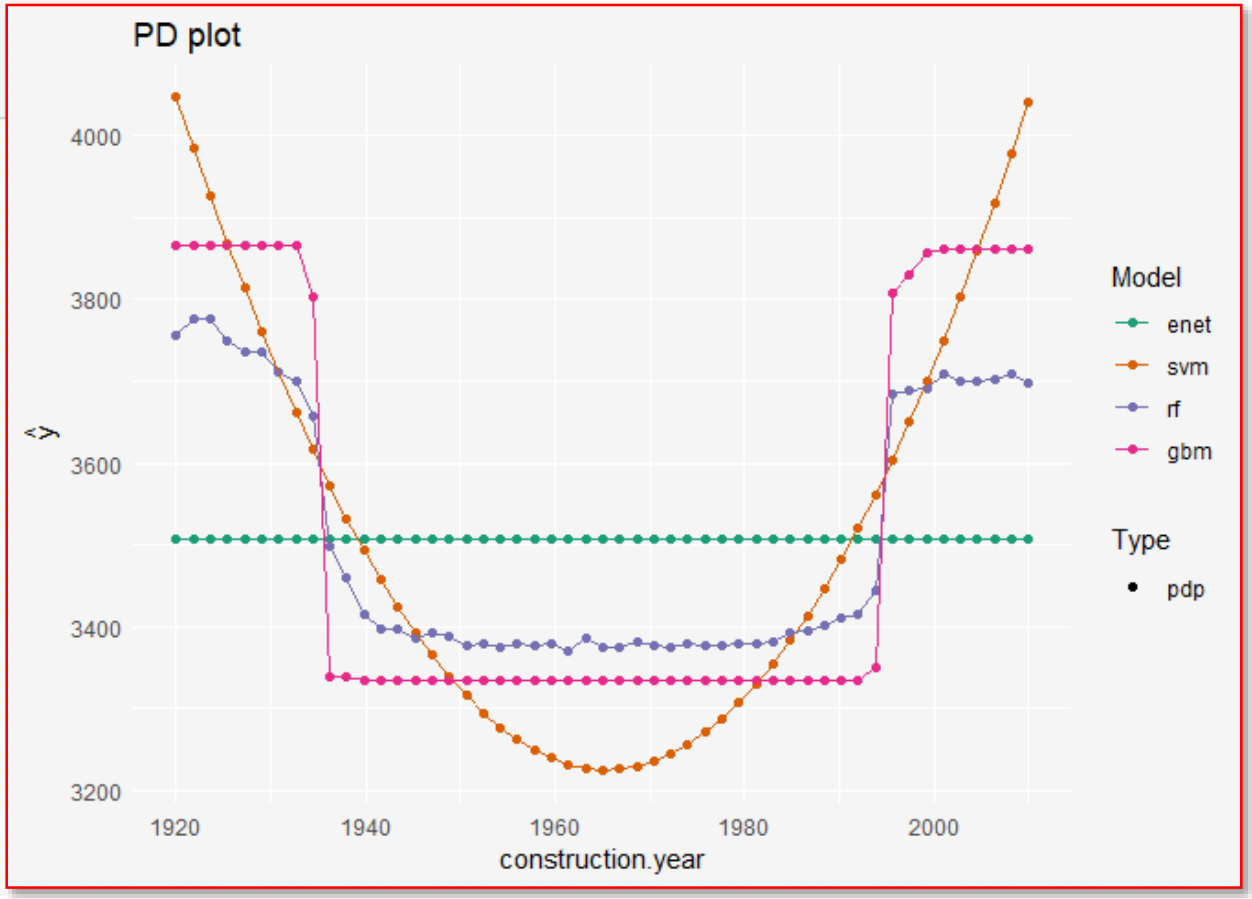
Partial Dependent Plot



https://pbiecek.github.io/DALEX_docs/2-architecture.html#architecture

モデル間の比較： variable_response(..., type = "pdp")

```
PDPs <- list()
for(model.name in model.labels){
  PDPs[[model.name]] <- variable_response(explainer[[model.name]],
                                          variable =  target.feature,
                                          type = "pdp")
}
plot.pdps <- plot(PDPs[["enet"]],
                  PDPs[["svm"]],
                  PDPs[["rf"]],
                  PDPs[["gbm"]]) +
  ggtitle("PD plot")

plot.pdps
```



Partial Dependence Plot

Advantage

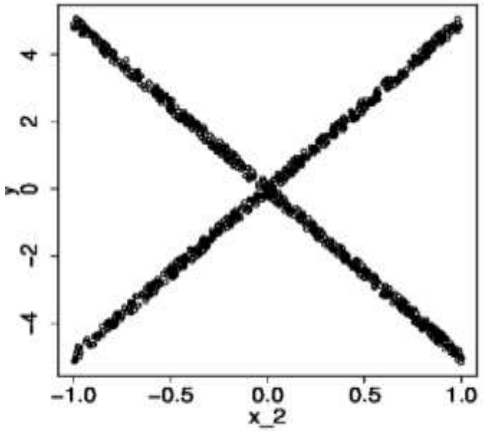
- 素人にも直感的にわかりやすい
- 実装も簡単

Disadvantage

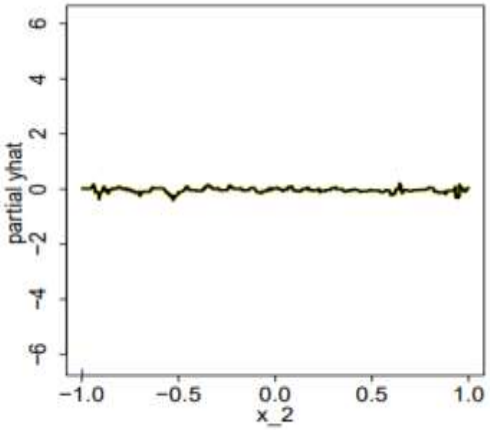
- 2変数以上は視覚化できない
  - 通常のヒトは3次元に暮らしているため
- 実際のデータ点の分布を省略すると誤解を招く可能性がある
  - x軸にデータポイントのラグをプロットする等の工夫をするとよい
- データの不均一性が平均されることでマスクされる可能性がある
  - ICE Plotと併用するとよい
- 周辺分布の計算のために特徴同士に独立性を仮定している。
  - 単純な値の置換により、身長2m体重30Kgとか出てきても気にしない
  - 周辺分布ではなく条件付き分布で扱うAccumulated Local Effect (ALE) plotsが解決策のひとつ。

Disadvantage : Partial Dependence Plot

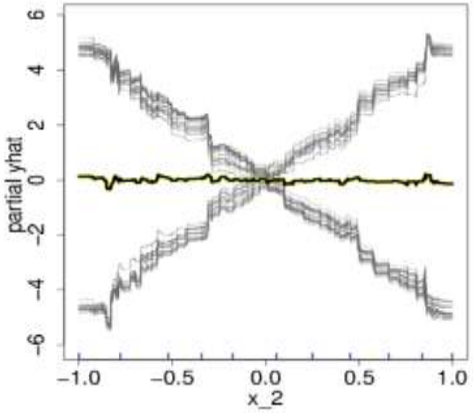
- データの不均一性が平均されることでマスクされる可能性がある
  - ICE Plotと併用するとよい
  - ICE Plotはすべてのデータを描画すると煩雑(というか真っ黒)になるので、適当にサンプリングするとよいとのこと。



(a) Scatterplot of Y versus X_2



(b) PDP



PD + ICE Plot

https://arxiv.org/abs/1309.6392

Partial Dependence (PD) + Individual Conditional Expectation (ICE)

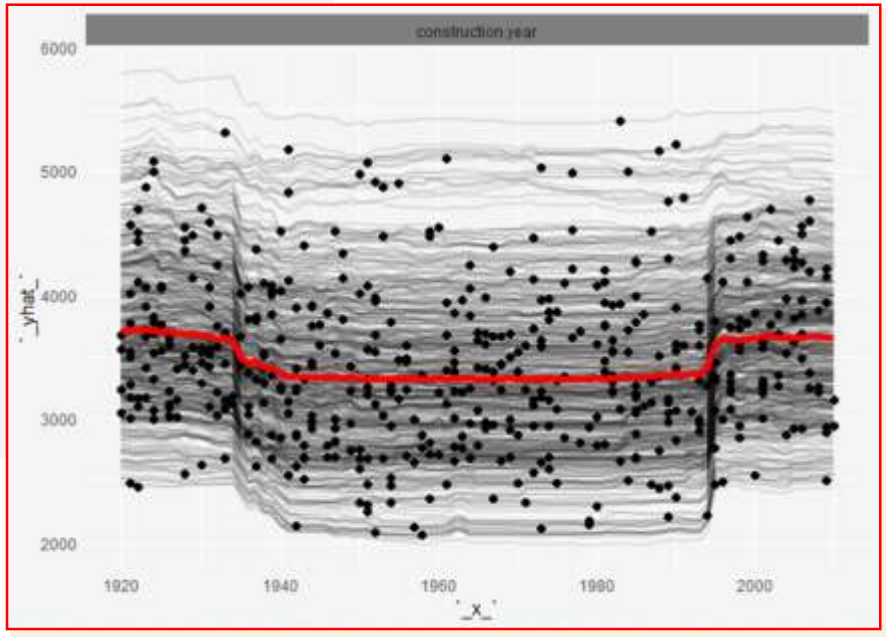
- **DALEX**自体はICEplotに対応していないが、同作者のceterisParibusで描画できる
- imlパッケージでは普通に“ice”として機能提供

ceteris paribus plot

```
install.packages("ceterisParibus", dependencies = TRUE)
```

```
library("ceterisParibus")
```

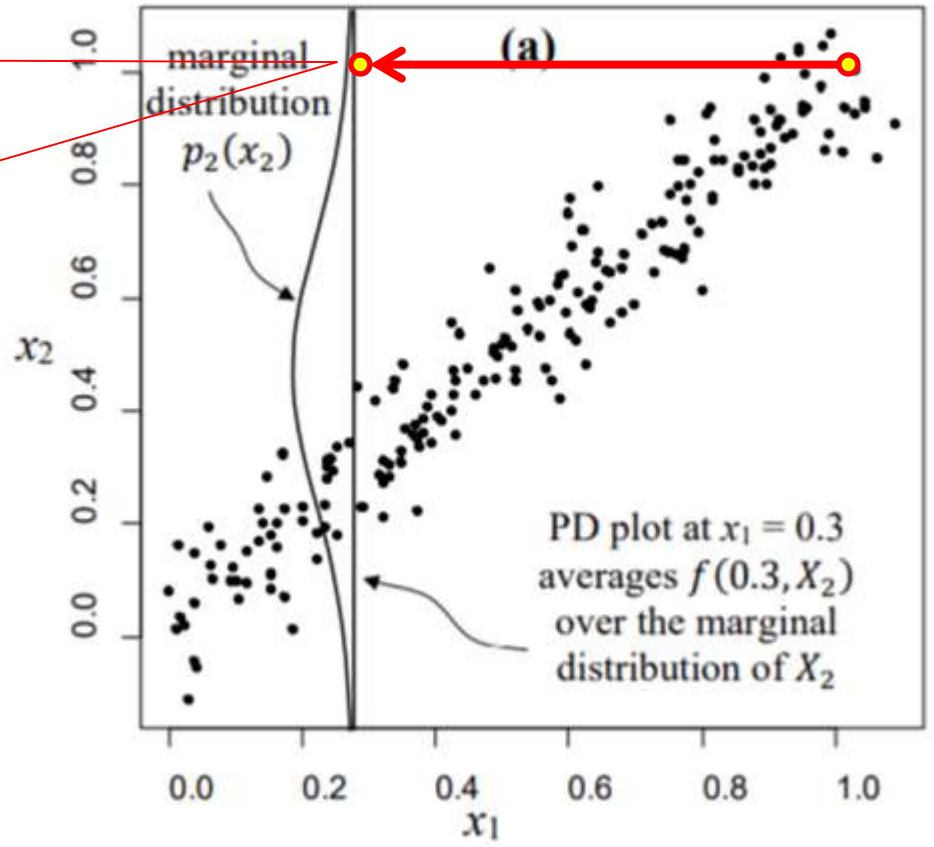
```
apartmentsTest.sub <- apartmentsTest %>% sample_n(500)
profile.rf.sub  <- ceteris_paribus(explainer.rf,
                                   observations = apartmentsTest.sub,
                                   y = apartmentsTest.sub$m2.price)

plot(profile.rf.sub,
     selected_variables = target.feature,
     show_residuals = FALSE,
     show_observations = TRUE,
     size = 1, alpha = 0.1) +
  ceteris_paribus_layer(
    profile.rf.sub,
    selected_variables = target.feature,
    aggregate_profiles = mean,
    show_observations = FALSE,
    size = 2, alpha = 1, color = "red")
```



Disadvantage： Partial Dependence Plot

- 周辺分布の計算のために特徴同士に独立性を仮定している。
  - その仮定の下で、残りの変数を固定したまま対象の変数を置き換える
  - x1とx2の２変数による予測 f(x1, x2)についてのPDを考えたとき、x1とx2の間に相関があると、推定にバイアスが生じる

X1=30kgの平均を求めるとき、
単純な値の置換によって
身長2mの人が
体重30Kgだとしたら？
のような、
極端なwhat-ifが生成されても
気にせず予測して平均する



解決策のひとつが、
周辺分布ではなく条件付き分布で扱う
Accumulated Local Effect (ALE) plots

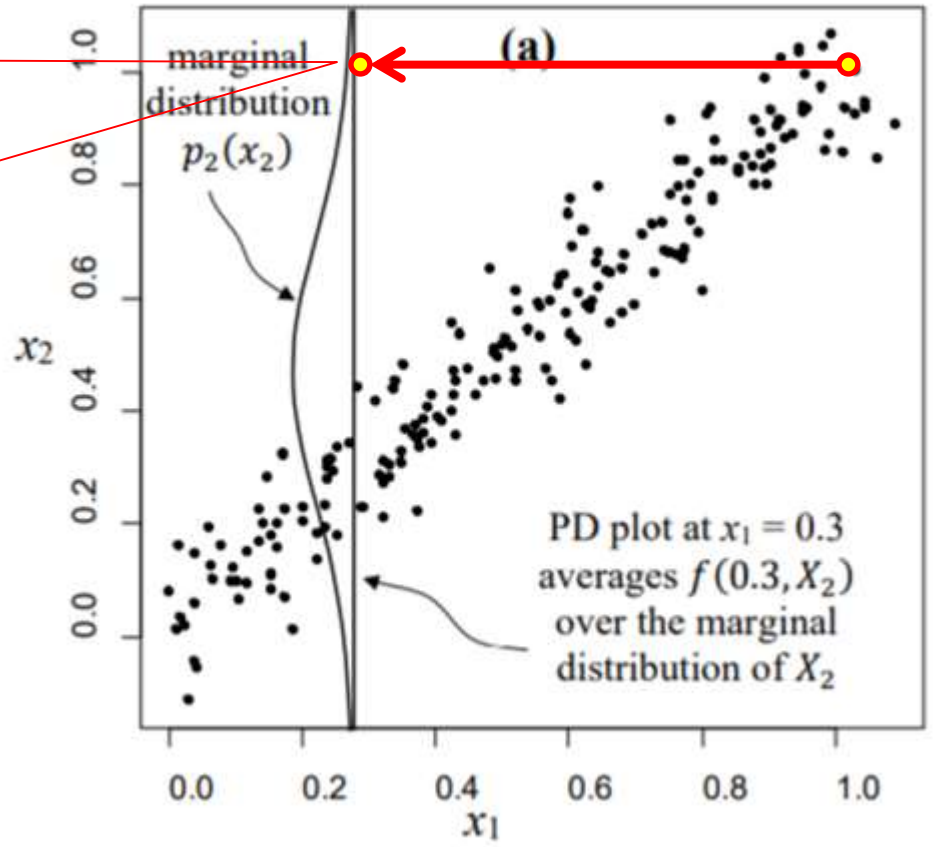
Accumulated Local Effects (ALE) Plot

policy	method name	iml	DALEX
understand entire model	residuals and goodness of fit	✗	✓
	permutation importance	✓	✓
	global surrogate	Tree surrogate	✗
understand feature(s)	Merging Path Plot (PDP for categorical data)	✗	✓
	Partial Dependence Plot (PDP for continuous data)	✓	✓
	Individual Conditional Expectation (ICE)	✓	Ceteris Paribus Plots
	Accumulated Local Effects (ALE) Plot	✓	✓
	Feature Interaction	✓	✗
local interpretation (for single prediction)	LIME	✓	✗
	SHAPLY value	✓	✗
	breakDown	✗	✓

(再掲) Partial Dependence Plotの問題点

- 周辺分布の計算のために特徴同士に独立性を仮定している。
  - その仮定の下で、残りの変数を固定したまま対象の変数を置き換える
  - x1とx2の２変数による予測　f(x1, x2)についてのPDを考えたとき、x1とx2の間に相関があると、推定にバイアスが生じる

X1=30kgの平均を求めるとき、
単純な値の置換によって
身長2mの人が
体重30Kgだとしたら？
のような、
極端なwhat-ifが生成されても
気にせず予測して平均する



注目する変数の変化に対する予測値の応答が知りたい

- 特徴の「値」の置き換えで生成したインスタンスの**全平均（PD)でなくても**よいのでは？

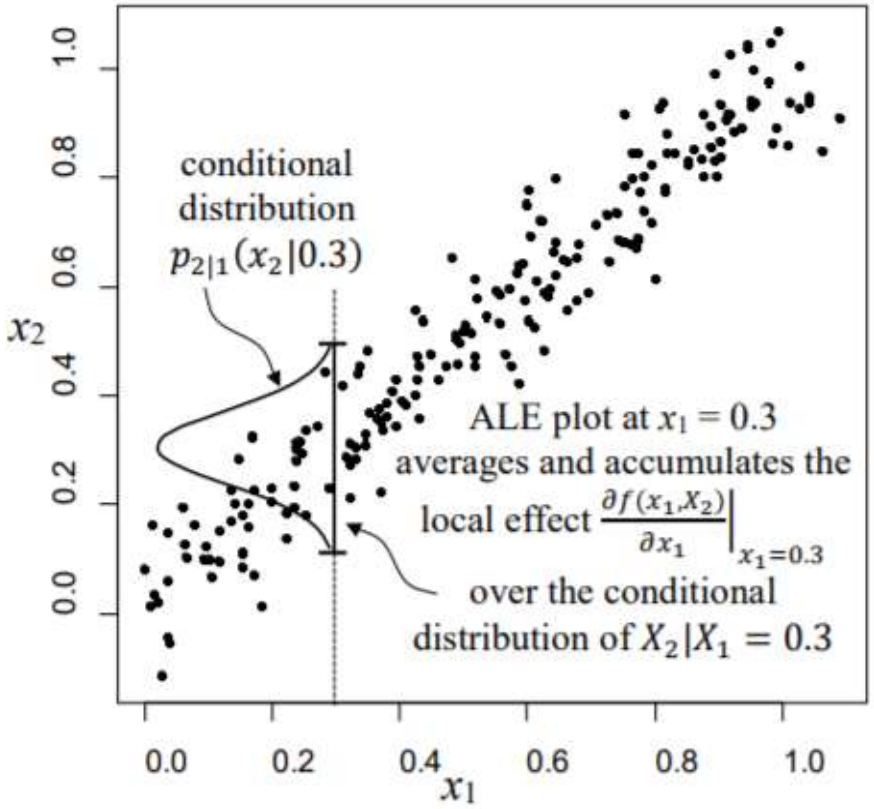


Figure 2. Illustration of the computation of $f_{1,ALE}(x_1)$ at $x_1 = 0.3$.

https://arxiv.org/abs/1612.08468

- 注目する特徴の「値」の置き換えで生成したインスタンスを平均するのではなく
- 注目する特徴の**「値」の近傍で予測値がどう動くか**を観察したらよいのでは？

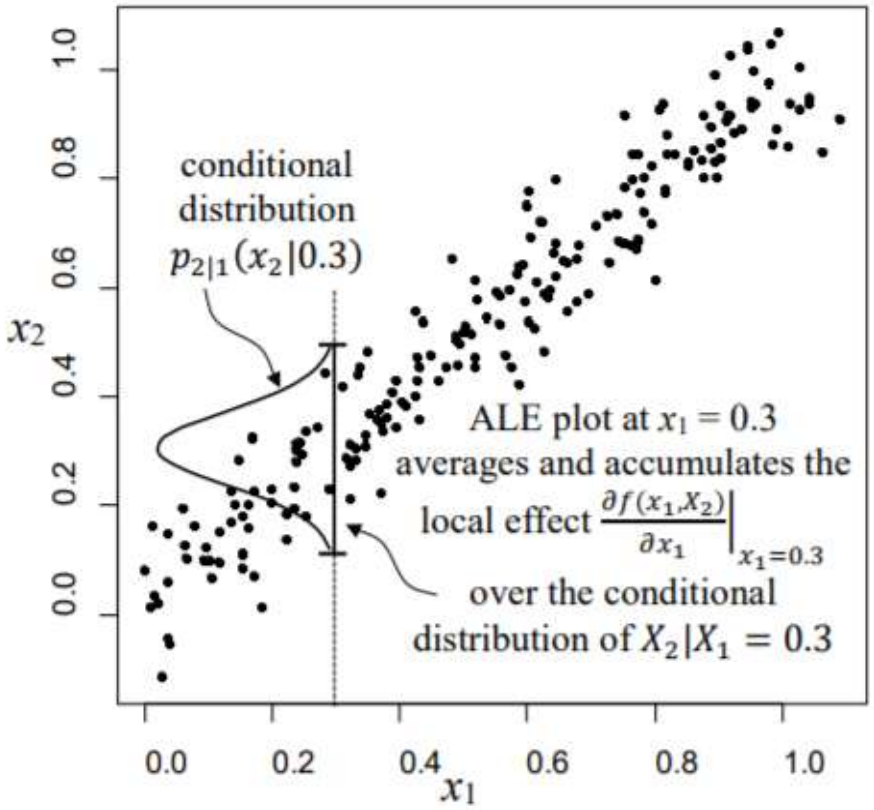


Figure 2. Illustration of the computation of $f_{1,ALE}(x_1)$ at $x_1 = 0.3$.

https://arxiv.org/abs/1612.08468

- 注目する特徴の「値」の置き換えで生成したインスタンスを平均するのではなく
- 注目する特徴の**「値」の近傍で予測値がどう動くか**を観察したらよいのでは？
- 注目する特徴の**「値」の周りでモデルの勾配**を観察したらよいのでは？

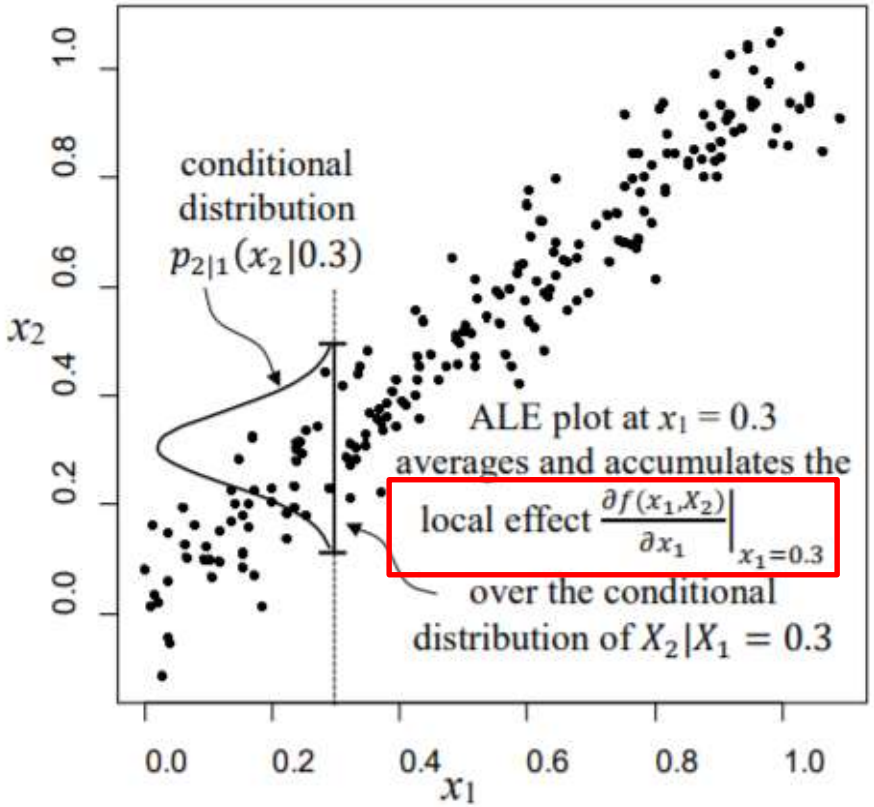
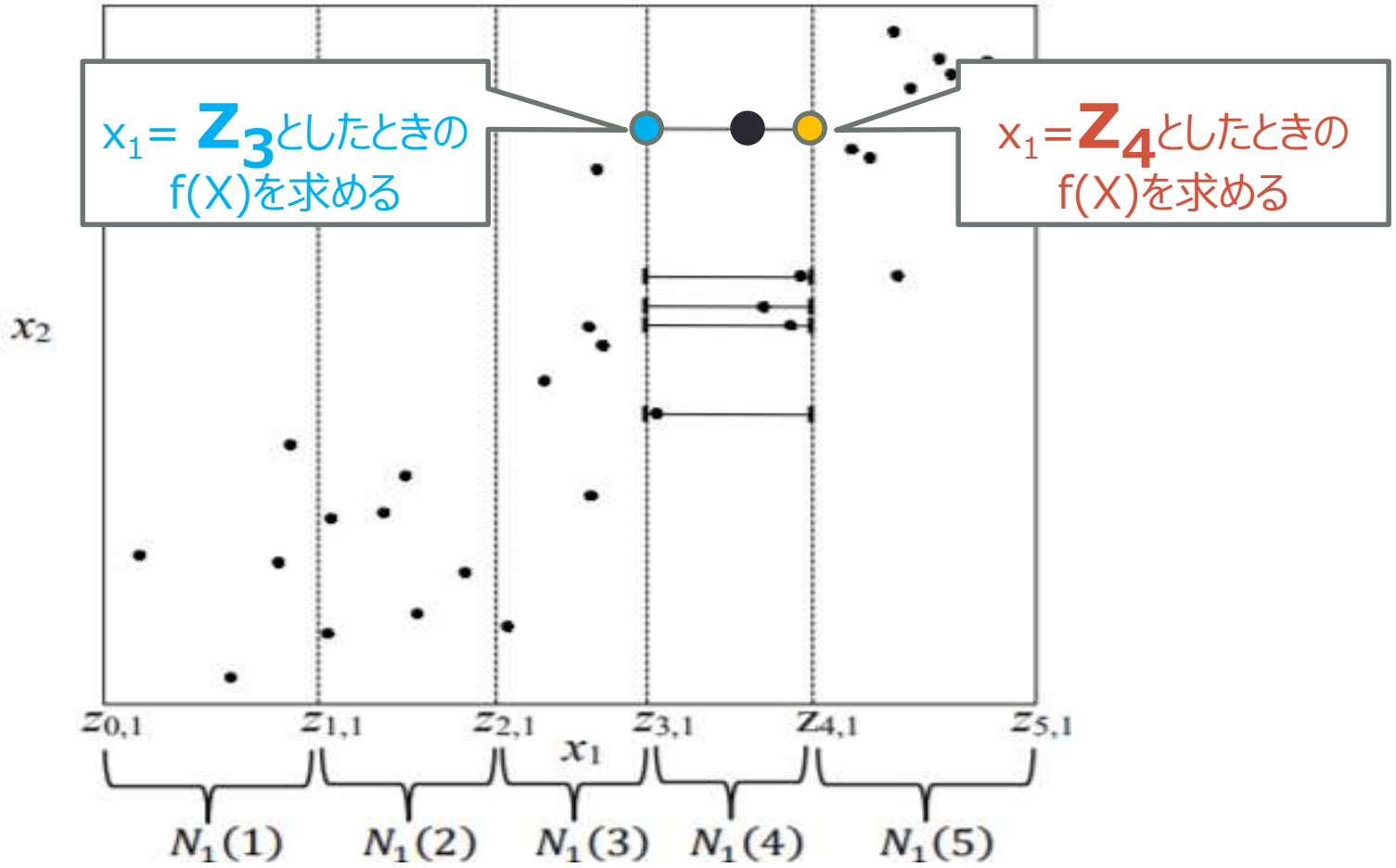


Figure 2. Illustration of the computation of $f_{1,ALE}(x_1)$ at $x_1 = 0.3$.

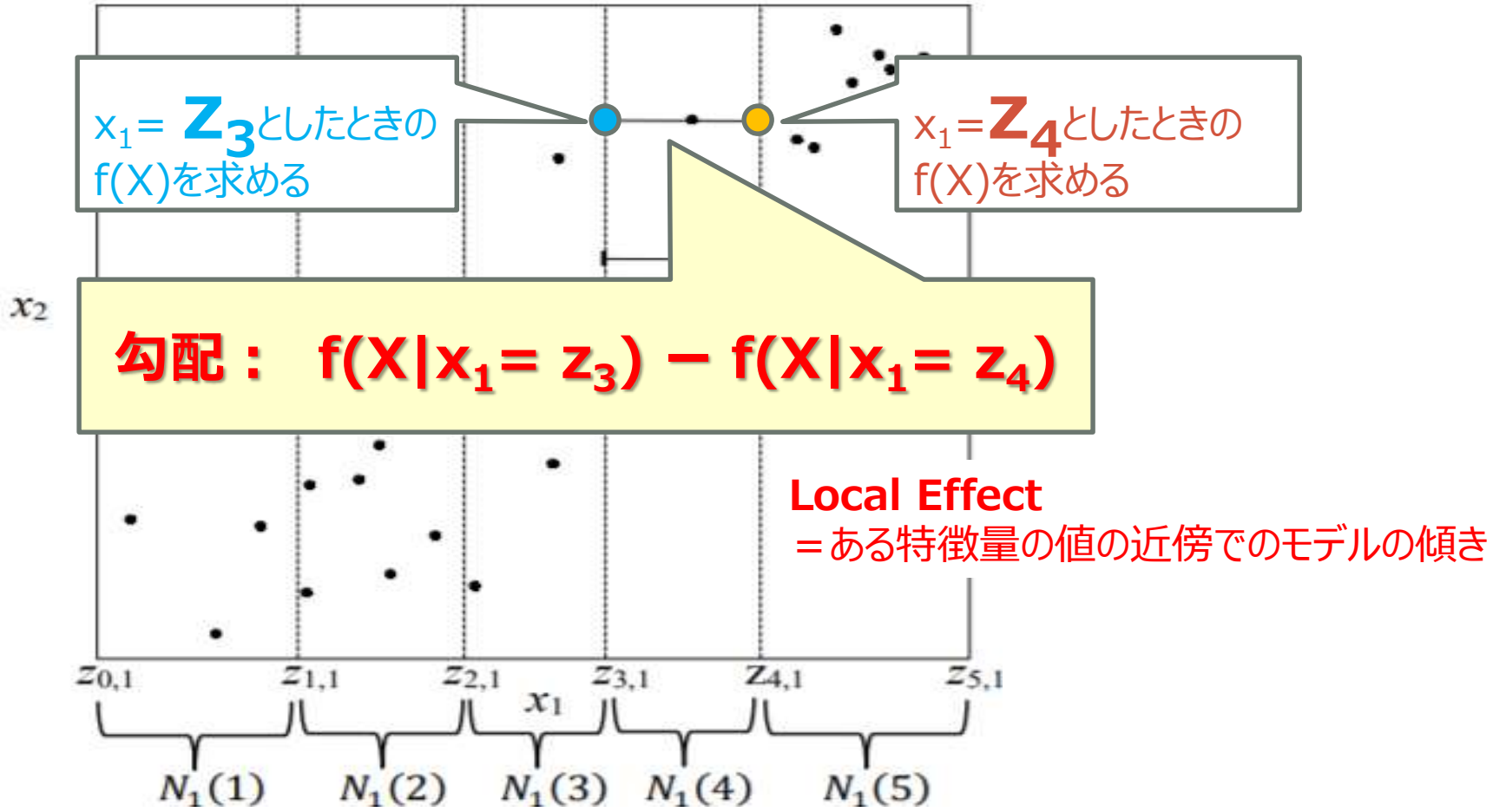
https://arxiv.org/abs/1612.08468

モチベーション

- 注目する特徴の「値」の置き換えで生成したインスタンスを平均するのではなく
- 注目する特徴の「値」の近傍で予測値がどう動くかを観察したらよいのでは？
- 注目する特徴の「値」の**周りの区間でモデルの勾配**を観察したらよいのでは？

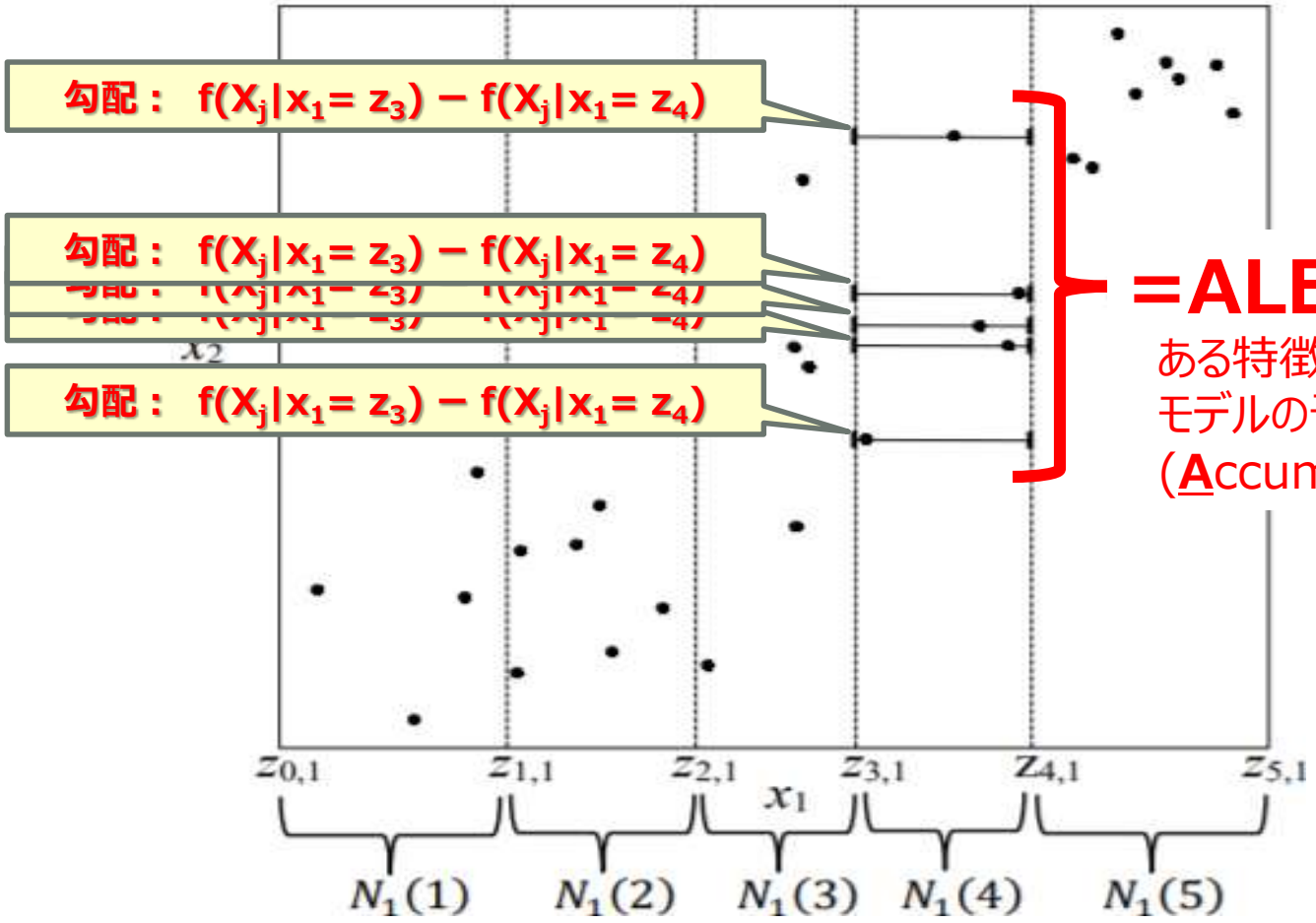


- 注目する特徴の「値」の置き換えで生成したインスタンスを平均するのではなく
- 注目する特徴の「値」の近傍で予測値がどう動くかを観察したらよいのでは？
- 注目する特徴の「値」の**周りの区間でモデルの勾配**を観察したらよいのでは？



Accumulated Local Effects (ALE) Plot

- 注目する特徴の「値」の置き換えで生成したインスタンスを平均するのではなく
- 注目する特徴の「値」の近傍で予測値がどう動くかを観察したらよいのでは？
- 注目する特徴の「値」の**周りの区間でモデルの勾配**を観察したらよいのでは？

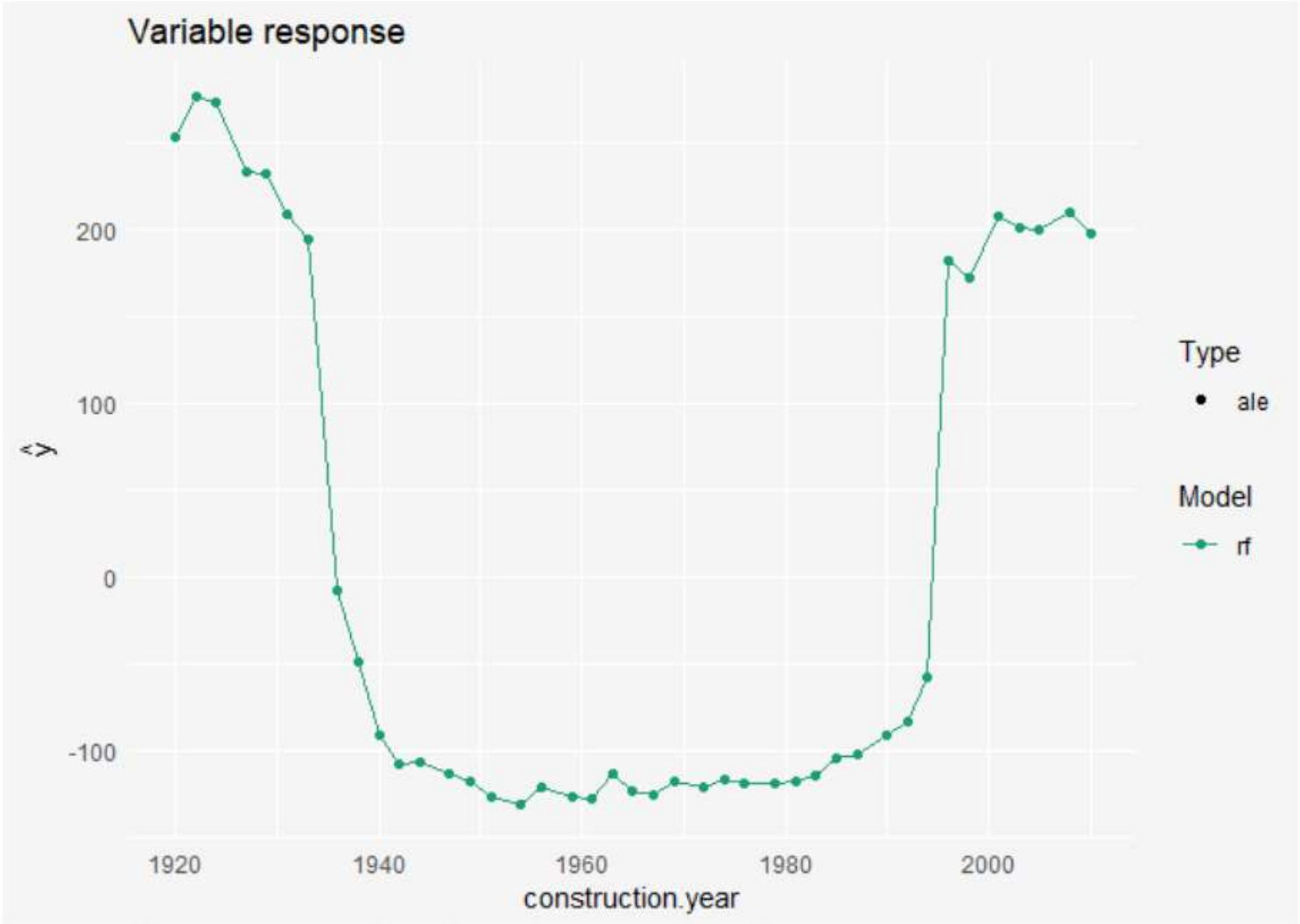


=ALE

ある特徴量の近傍で、**平均的に**
モデルの予測値がどう変化するか？
(**A**ccumulated **L**ocal **E**ffect)

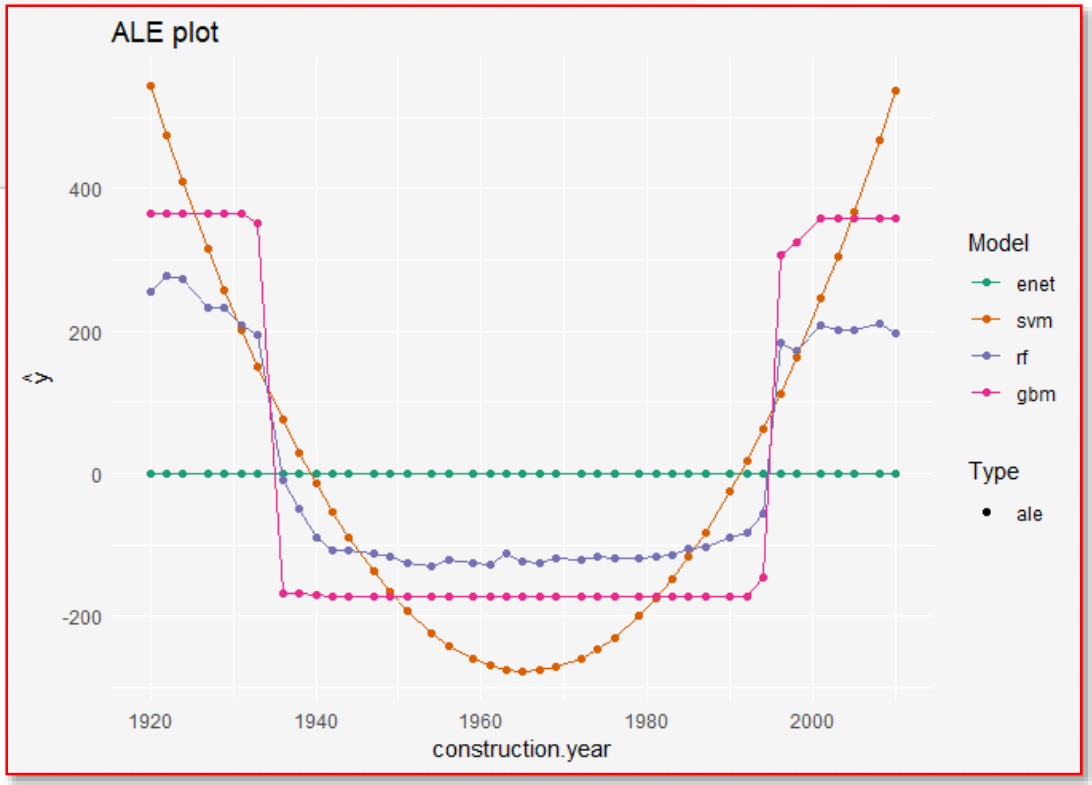
モデル間の比較： variable_response(..., type = “ale")

```
ale  <- variable_response(explainer.rf, variable =  target.feature, type = "ale")

plot(ale)
```



モデル間の比較： variable_response(..., type = “ale")

```
ALEs <- list()
for(model.name in model.labels){
  ALEs[[model.name]] <- variable_response(explainer[[model.name]],
                                          variable =  target.feature,
                                          type = "ale")
}
plot.ales <- plot(ALEs[["enet"]],
                  ALEs[["svm"]],
                  ALEs[["rf"]],
                  ALEs[["gbm"]]) +
  ggtitle("ALE plot")

plot.ales
```



Accumulated Local Effects (ALE) Plot

Advantage

- PDPとくらべて：
  - 特徴が相関していてもうまく行く
  - 計算量が少ない

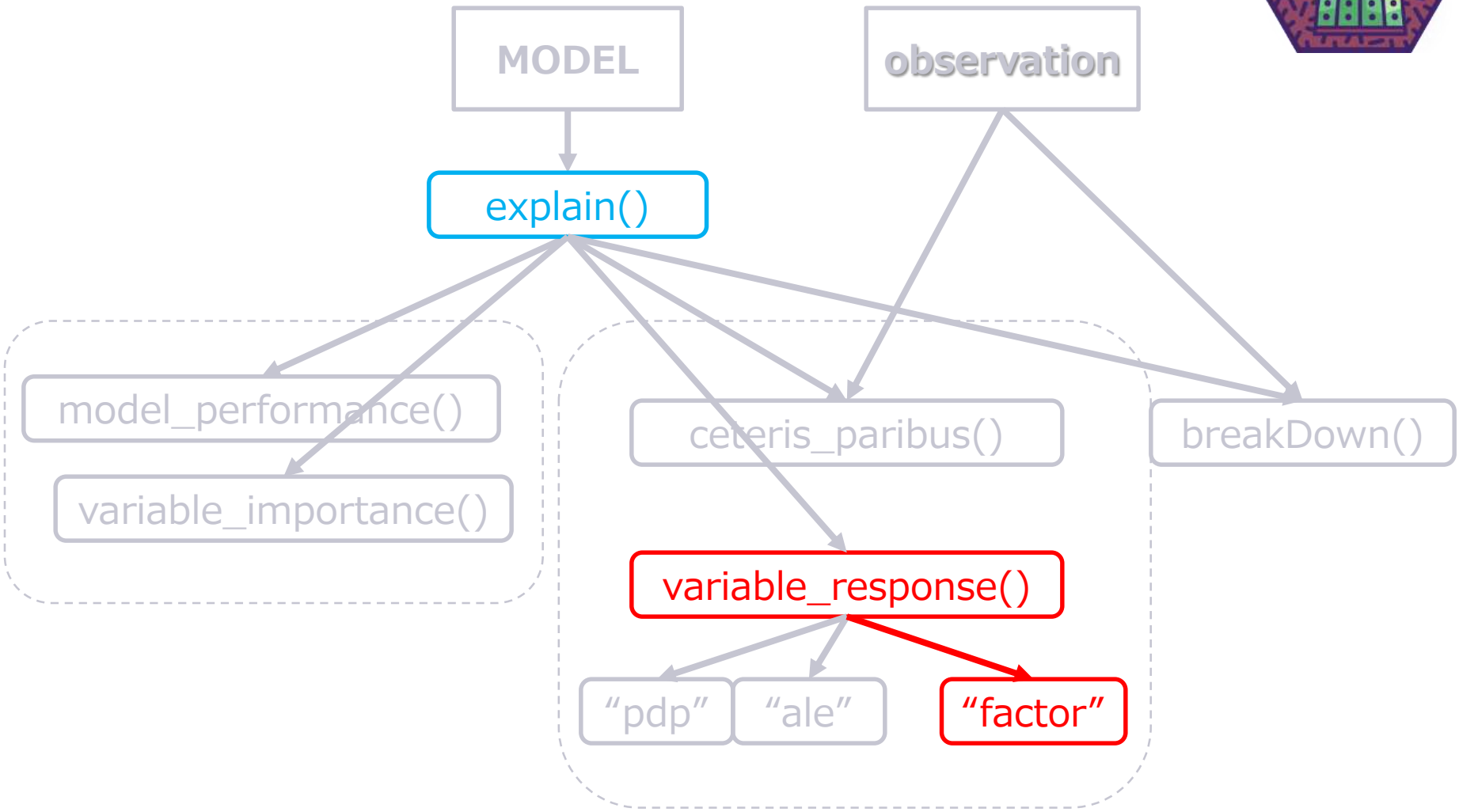
Disadvantage

- PDPとくらべて：
  - コンセプトが分かりにくい
  - ICEプロットとの対照が出来ない
  - 変数が独立ならPDPでもよい

Merging Path Plot (factorMerger)

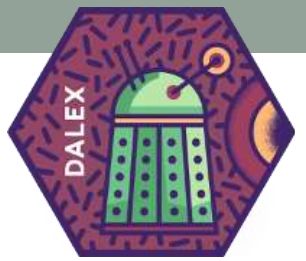
policy	method name	iml	DALEX
understand entire model	residuals and goodness of fit	✗	✓
	permutation importance	✓	✓
	global surrogate	Tree surrogate	✗
understand feature(s)	Merging Path Plot (PDP for categorical data)	✗	✓
	Partial Dependence Plot (PDP for continuous data)	✓	✓
	Individual Conditional Expectation (ICE)	✓	Ceteris Paribus Plots
	Accumulated Local Effects (ALE) Plot	✓	✓
	Feature Interaction	✓	✗
local interpretation (for single prediction)	LIME	✓	✗
	SHAPLY value	✓	✗
	breakDown	✗	✓

DALEX:: Variable_response(…, type = “factor”)



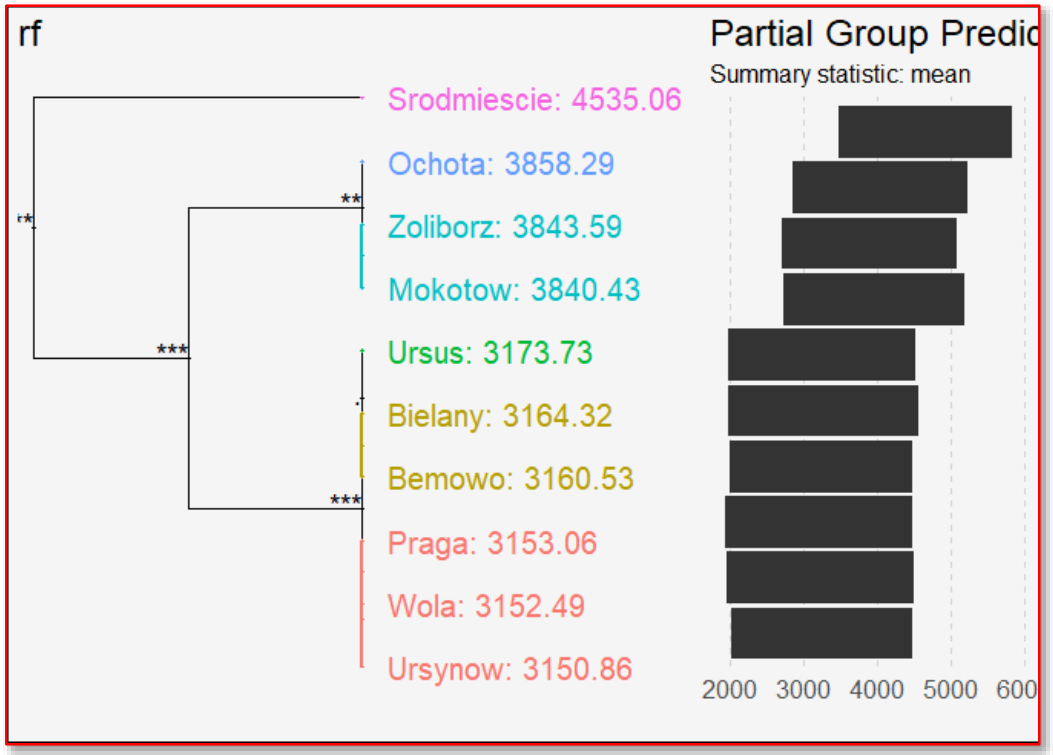
https://pbiecek.github.io/DALEX_docs/2-architecture.html#architecture

DALEX:: Variable_response(…, type = “factor”)



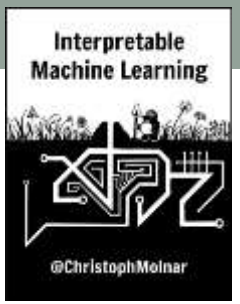
- 因子型（Factor）の変数に対するPDP、みたいなもの
- ANOVA によるpost hoc testにもとづいて、因子同士をグループ化する
- 内部でfactorMergerパッケージが呼ばれている

```
mpp <- DALEX::variable_response(explainer.rf, variable =  "district", type = "factor")

plot(mpp)
```



Feature interaction

policy	method name	iml	DALEX
understand entire model	residuals and goodness of fit	✗	✓
	permutation importance	✓	✓
	global surrogate	Tree surrogate	✗
understand feature(s)	Merging Path Plot (PDP for categorical data)	✗	✓
	Partial Dependence Plot (PDP for continuous data)	✓	✓
	Individual Conditional Expectation (ICE)	✓	Ceteris Paribus Plots
	Accumulated Local Effects (ALE) Plot	✓	✓
	Feature Interaction	✓	✗
local interpretation (for single prediction)	LIME	✓	✗
	SHAPLY value	✓	✗
	breakDown	✗	✓



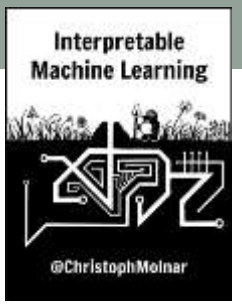
概要

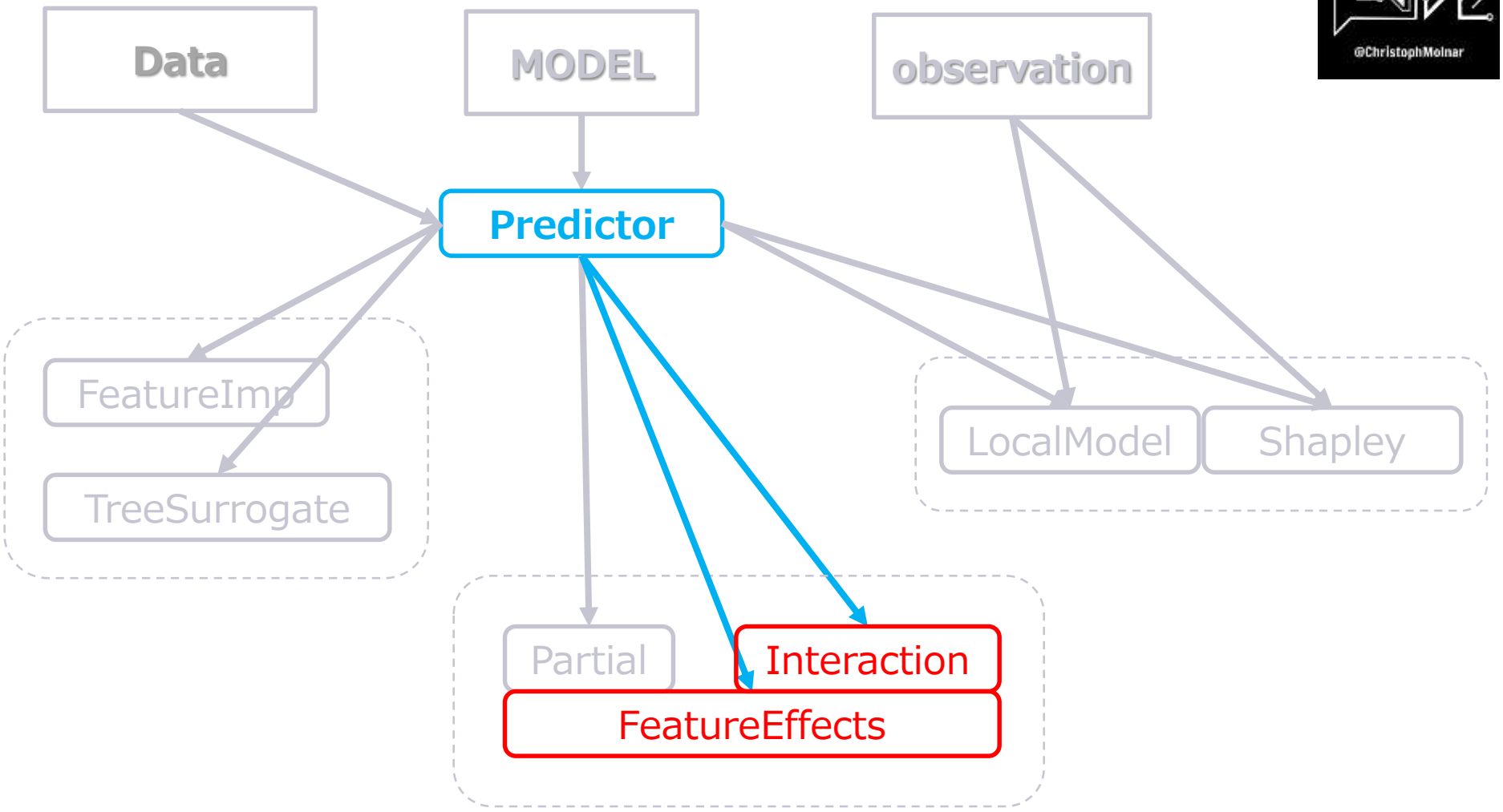
- RuleFit ※の論文中で提案されている H-statistic を指標とする
- H-statistic は partial dependenceを分解することで得られる
  - 0 (interactionなし) ～
  - 1 (f(x)の分散のうち、100% がinteractionsに由来する).

※ アンサンブル木をLASSOで枝狩りする手法

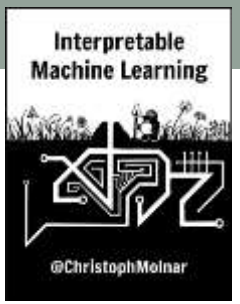
Friedman, Jerome H, and Bogdan E Popescu. “Predictive learning via rule ensembles.” The Annals of Applied Statistics. JSTOR, 916–54. (2008)

Feature interaction



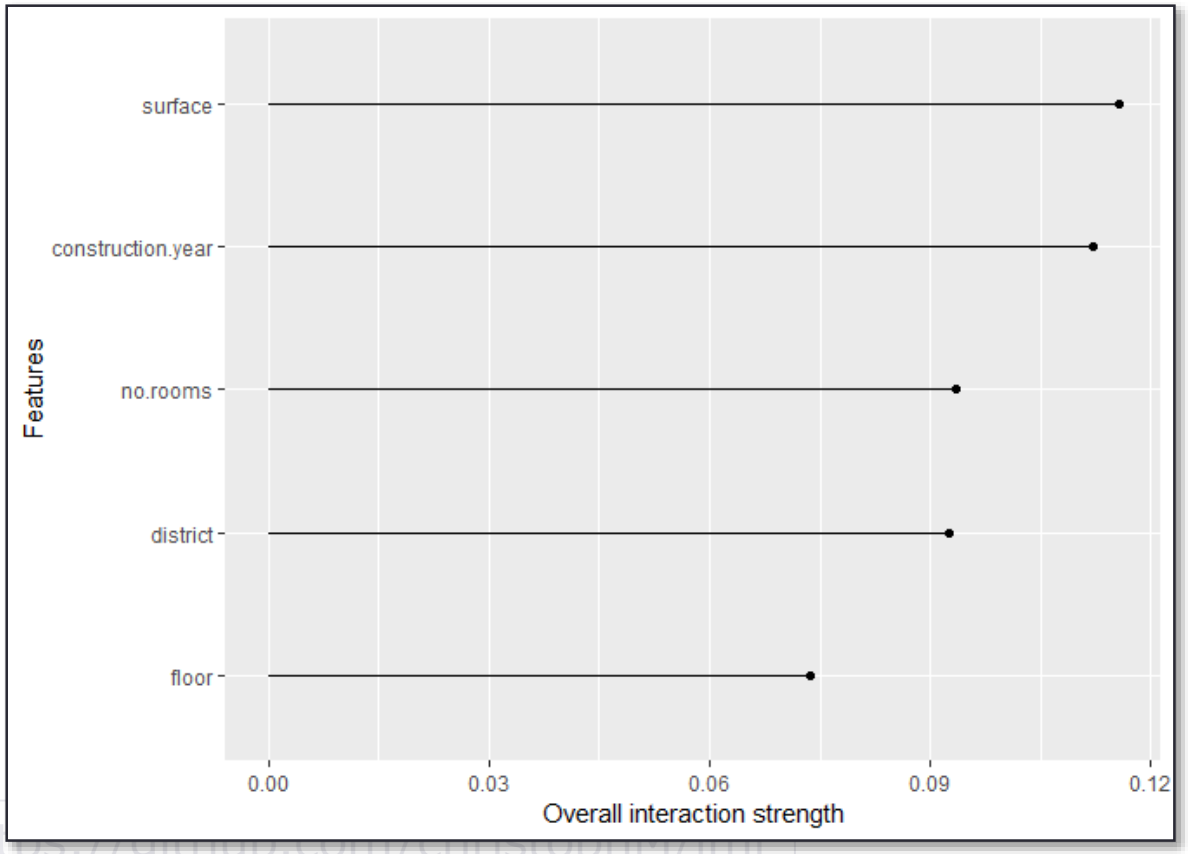


https://github.com/christophM/iml



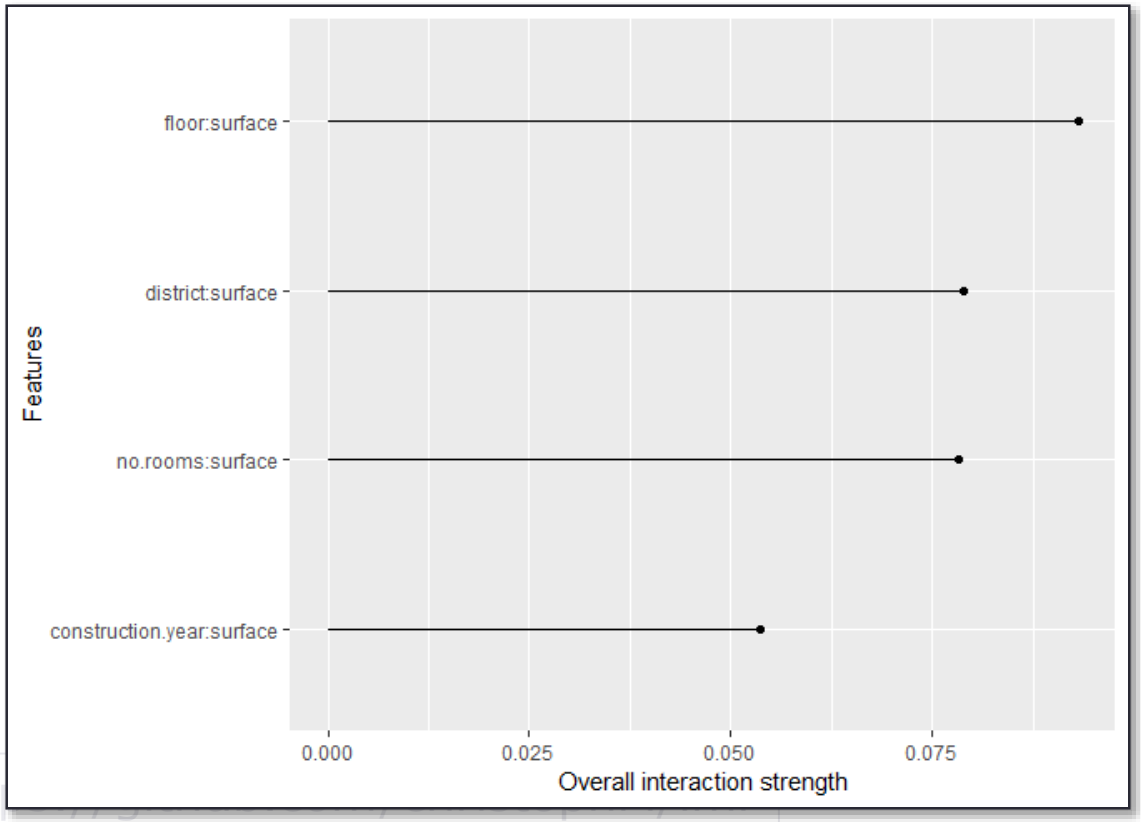
Overall The interaction strength (H-statistic) for each feature with all other features

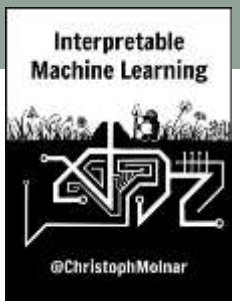
```
interact.rf <- Interaction$new(predictor.rf)
plot(interact.rf)
```



2-way interactions strength (H-statistic) between the selected feature and the other features

```
interact.2way.rf <- Interaction$new(predictor.rf, feature = "surface")

plot(interact.2way.rf)
```





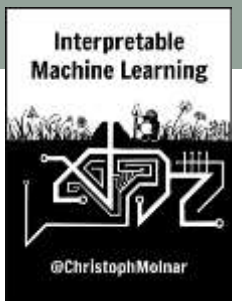
Advantage（H-statistic）

- 意味のある解釈ができる
  - 相互作用によって説明できる分散の割合
  - [0, 1]の無次元な量なので、特徴間・モデル間で同等な比較ができる。
- 2次元以上の任意の相互作用を分析することも可能。

Disadvantage （H-statistic）

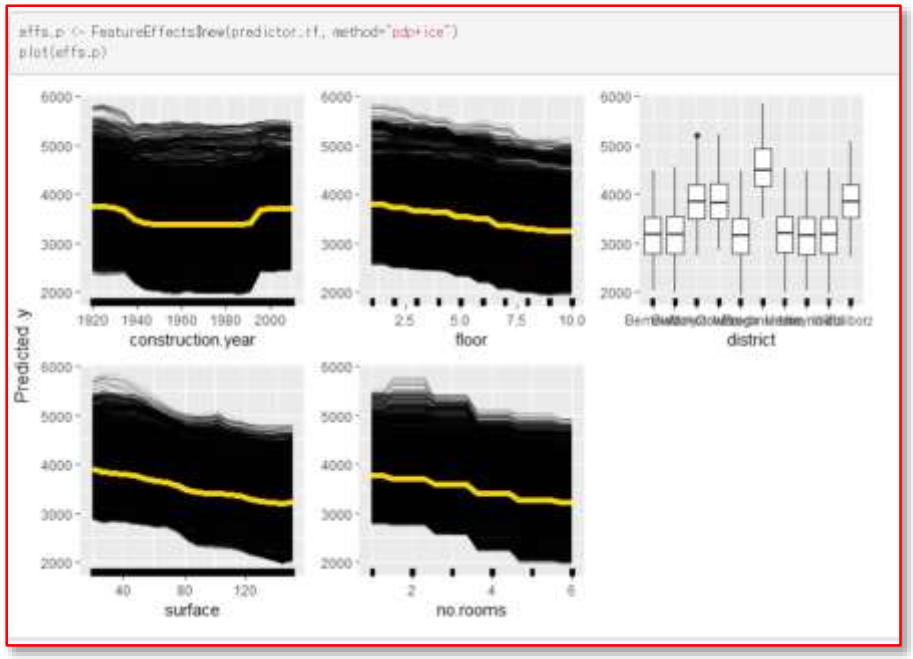
- 部分従属プロットと同様の問題を持つ
  - 計算コストが高い。サンプリング近似をすると、推定結果が不安定になる可能性がある。
  - 独立して特徴をシャッフルできるという仮定を置いている
- どの程度大きければ、相互作用が強いといえるのかも難しい。
  - （特殊な場合を除いて）有意性の検定法がない
- 相互作用の強さはわかるが、どのような相互作用なのか具体的にはわからない
  - 関心のある相互作用について2次元の部分従属プロットを作成するとよい
- 画像分類には使えない。
  - 入力がピクセルの場合、H-statisticは意味をもたない

iml:: FeatureEffect$new()

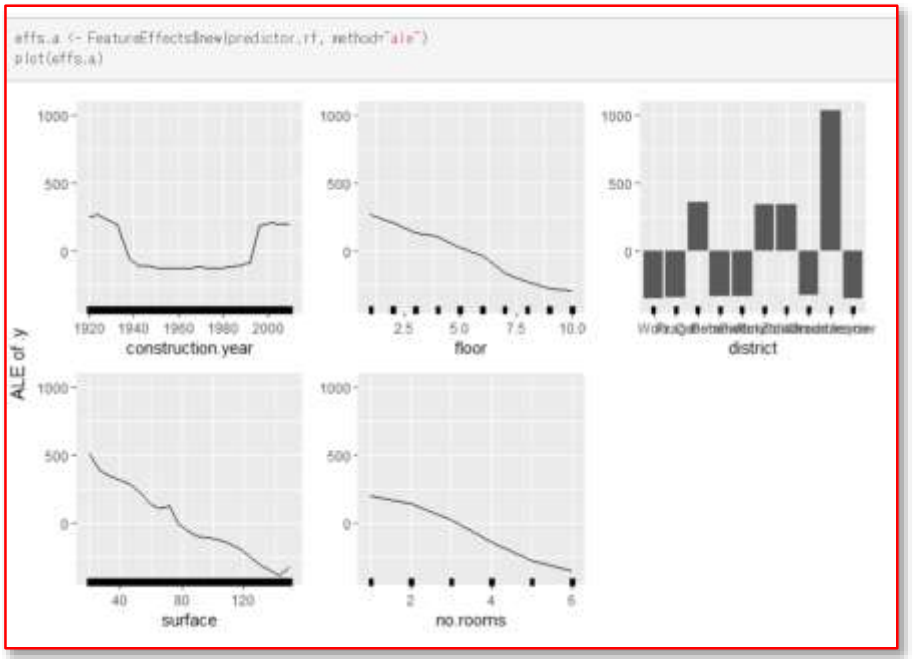


Interactionを確認した後は、PD/ALE plotで挙動を確認

PDP+ICE plot

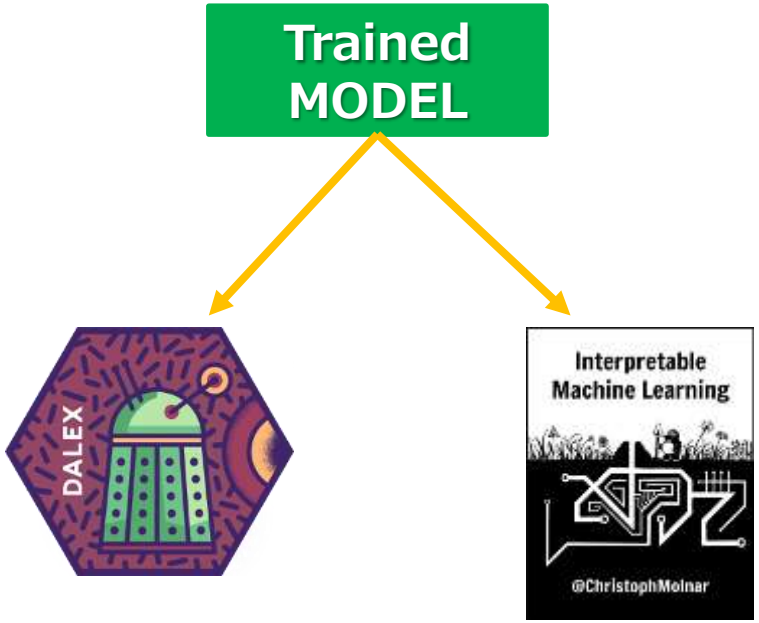


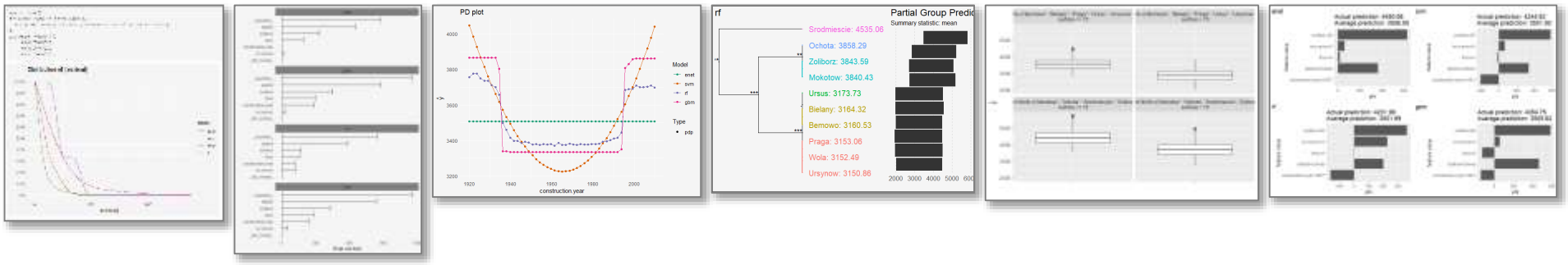
ALE plot



説明のアプローチ

1. モデルの性能や特性の評価
2. 特徴量（変数）に対するモデルの応答をみる
3. **あるデータに対する予測がどのように得られたかを説明する**

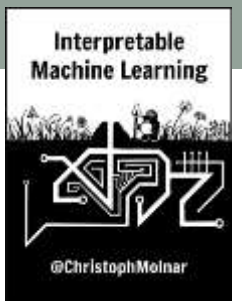


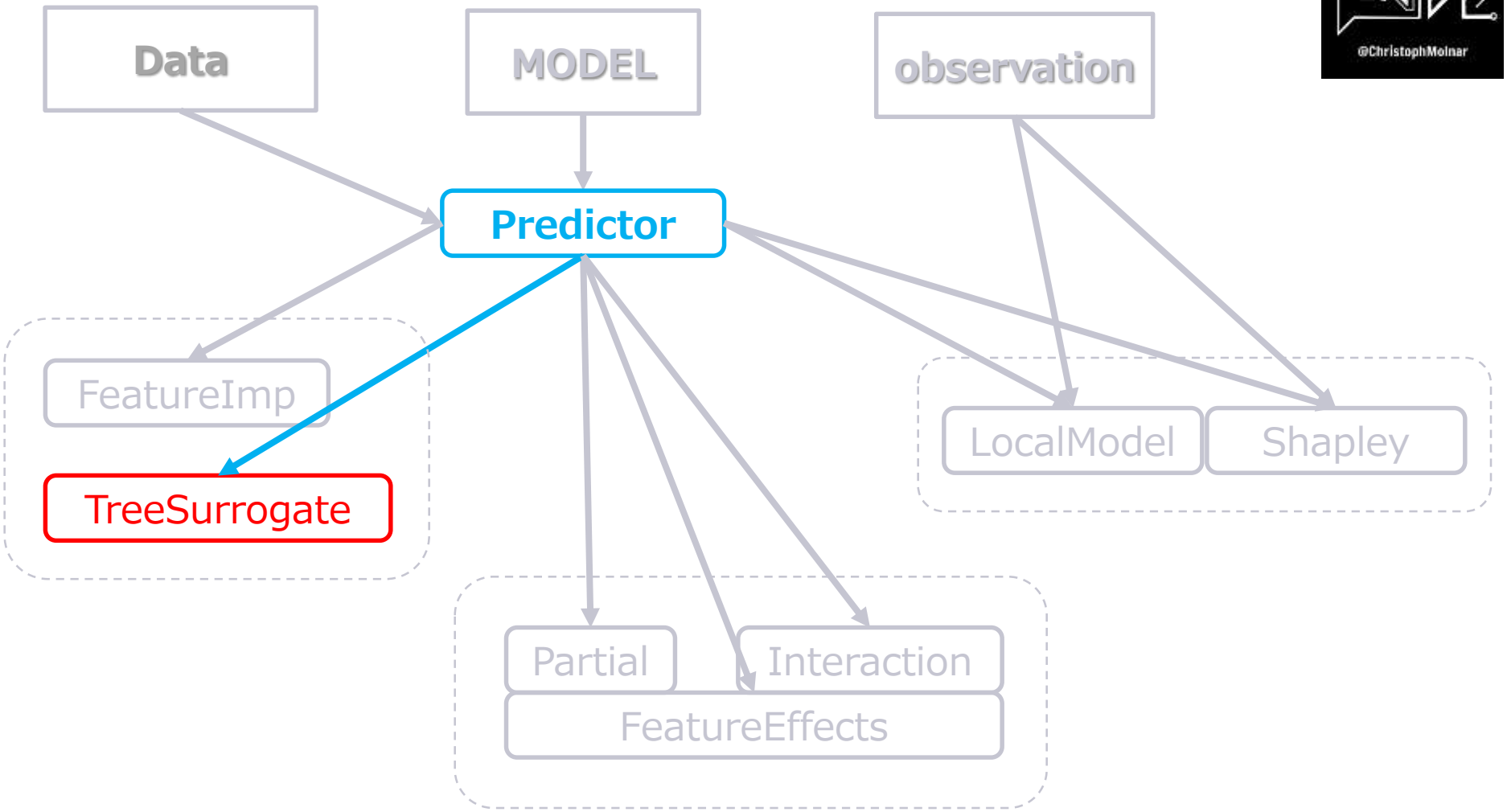


Global surrogate

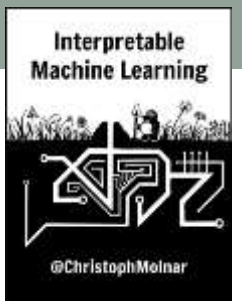
policy	method name	iml	DALEX
understand entire model	residuals and goodness of fit	✗	✓
	permutation importance	✓	✓
	global surrogate	Tree surrogate	✗
understand feature(s)	Merging Path Plot (PDP for categorical data)	✗	✓
	Partial Dependence Plot (PDP for continuous data)	✓	✓
	Individual Conditional Expectation (ICE)	✓	Ceteris Paribus Plots
	Accumulated Local Effects (ALE) Plot	✓	✓
	Feature Interaction	✓	✗
local interpretation (for single prediction)	LIME	✓	✗
	SHAPLY value	✓	✗
	breakDown	✗	✓

Global surrogate





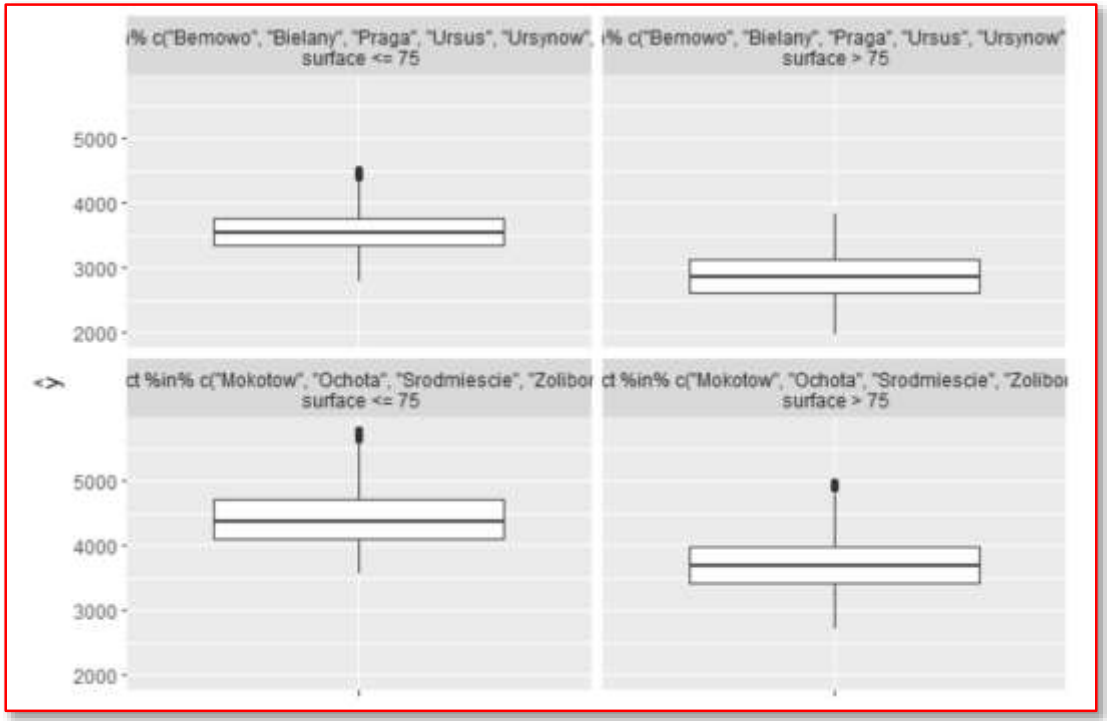
https://github.com/christophM/iml



概要

- 複雑なモデルの入力と予測をデータとして、単純なモデルでフィットしなおす
- Imlは決定木でのrefitのみ提供

```
tree = TreeSurrogate$new(predictor.rf, maxdepth = 2)
plot(tree)
```



Advantage

- あらゆるブラックボックスモデルに対して使用できる。
- 実装と説明が容易
- R^2値などでブラックボックスによる予測に対する近似精度を簡単に測定できる

Disadvantage

- 現実のタスクで、単純なサロゲートで良好な近似が得られるとは限らない
- 選択した解釈可能なモデルの長所と短所をすべて引き継ぐ。
- **モデルが出力する予測値について説明する（データの説明ではない）**

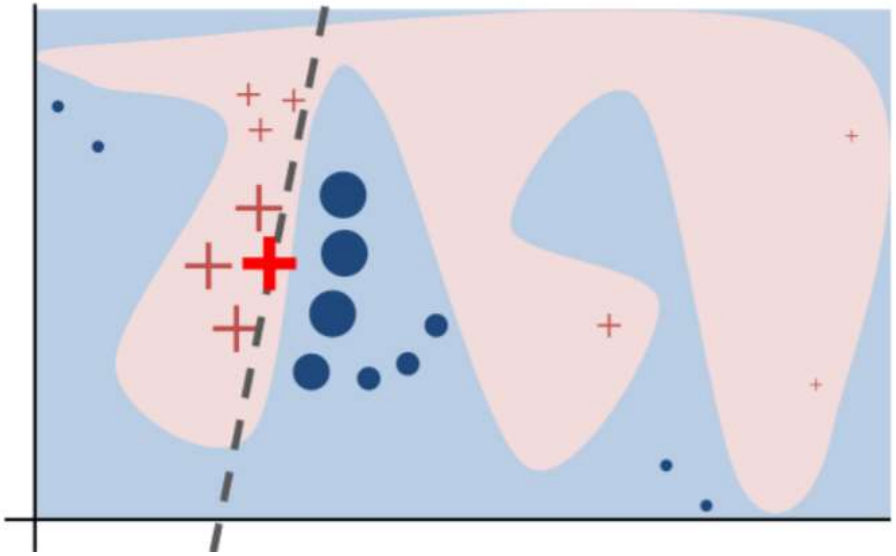
LIME

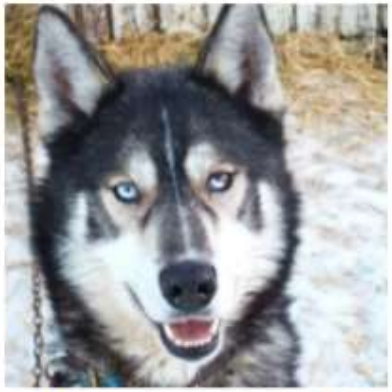
policy	method name	iml	DALEX
understand entire model	residuals and goodness of fit	✗	✓
	permutation importance	✓	✓
	global surrogate	Tree surrogate	✗
understand feature(s)	Merging Path Plot (PDP for categorical data)	✗	✓
	Partial Dependence Plot (PDP for continuous data)	✓	✓
	Individual Conditional Expectation (ICE)	✓	Ceteris Paribus Plots
	Accumulated Local Effects (ALE) Plot	✓	✓
	Feature Interaction	✓	✗
local interpretation (for single prediction)	LIME	✓	✗
	SHAPLY value	✓	✗
	breakDown	✗	✓

LIME (Locally Interpretable Model-agnostic Explanations)

- 説明したい観察の周辺で、単純な線形モデルによる近似を行う
- 近似モデルの重みが、各変数の予測に対する説明を表す

"Why Should I Trust You?"





(a) Husky classified as wolf

(b) Explanation

https://arxiv.org/abs/1602.04938

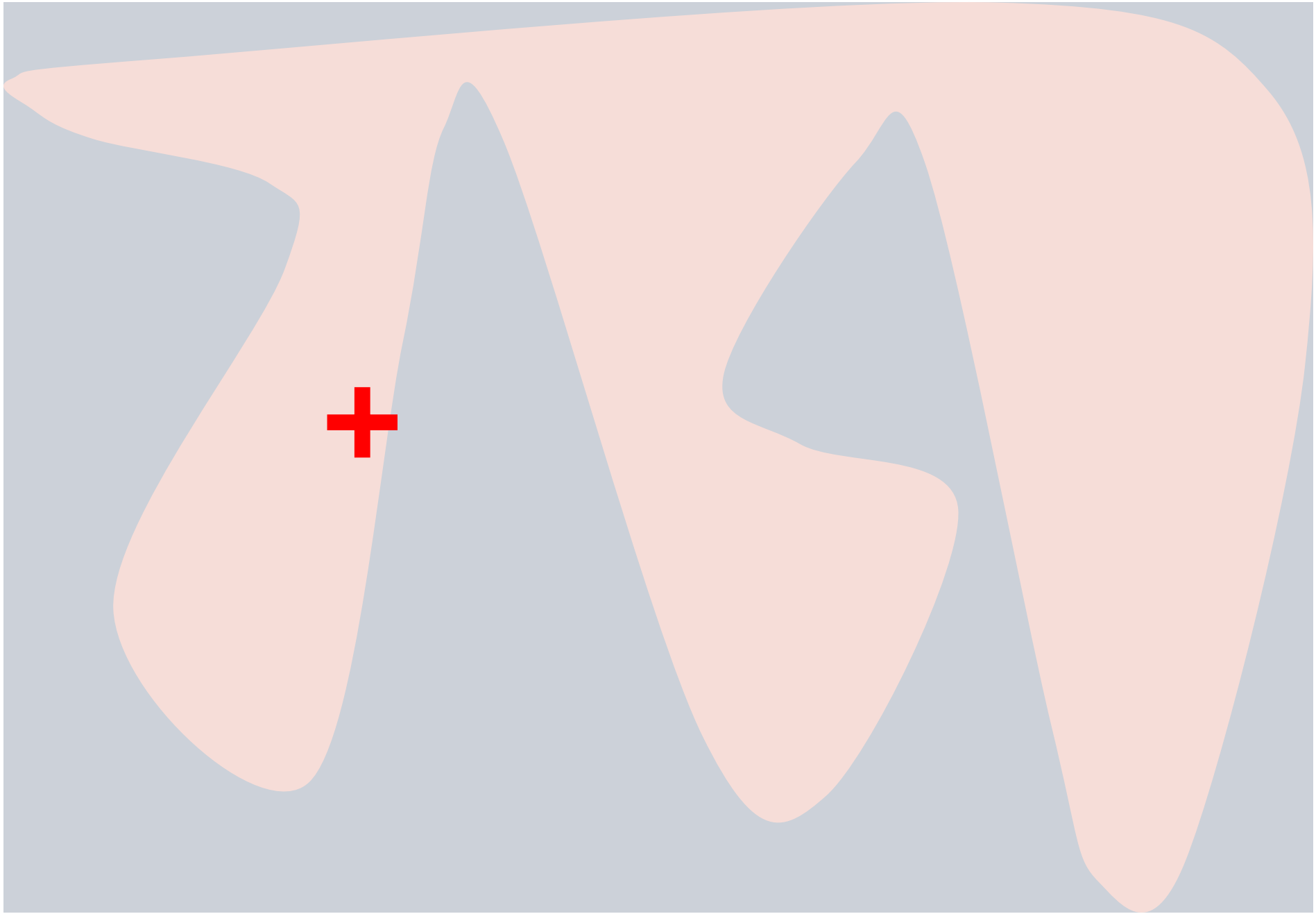
推定ステップ

- The general approach lime takes to achieving this goal is as follows:

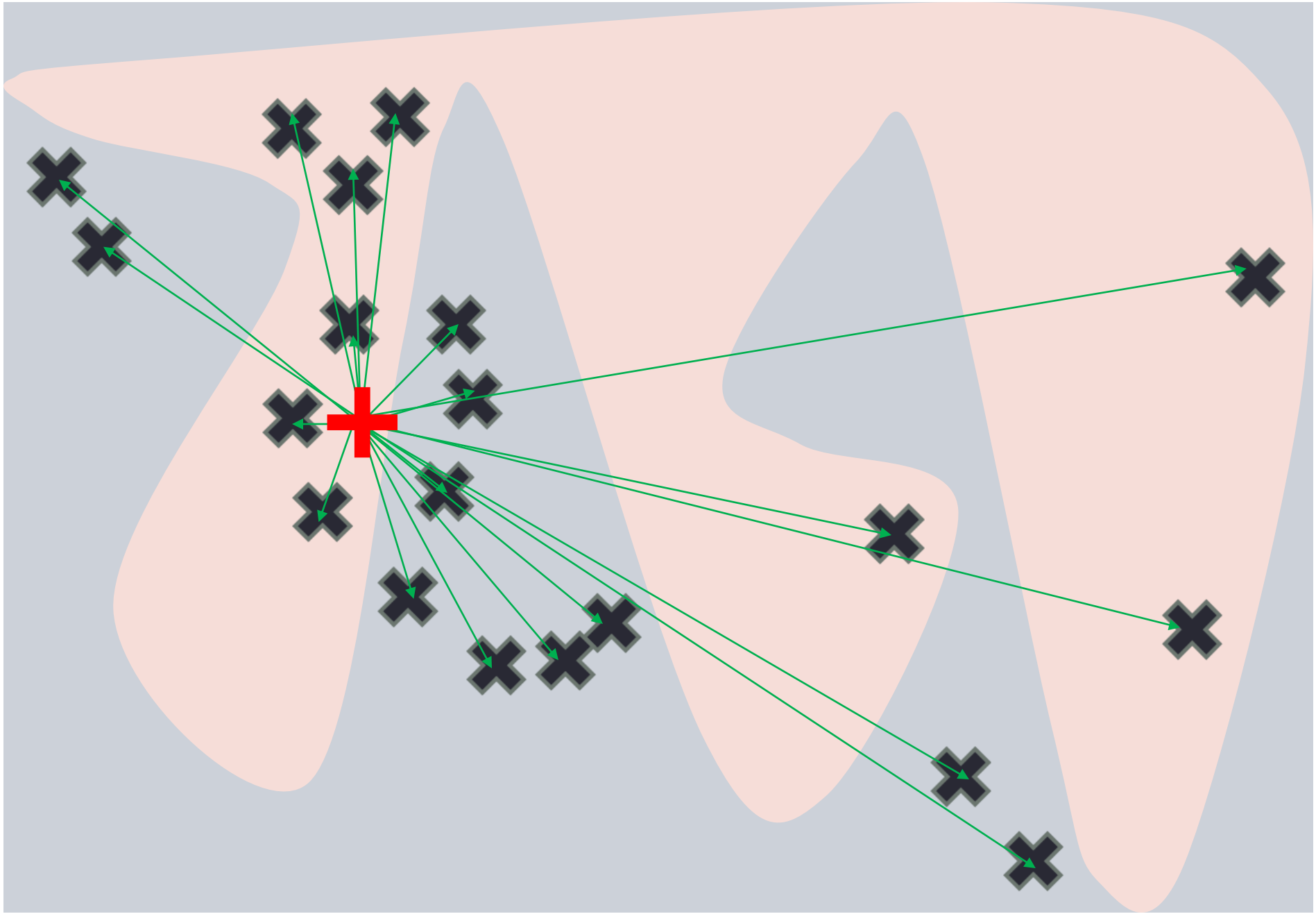
1. For each prediction to explain, permute the observation n times.
2. Let the complex model predict the outcome of all permuted observations.
3. Calculate the distance from all permutations to the original observation.
4. Convert the distance to a similarity score.
5. Select m features best describing the complex model outcome from the permuted data.
6. Fit a simple model to the permuted data, explaining the complex model outcome with the m features from the permuted data weighted by its similarity to the original observation.
7. Extract the feature weights from the simple model and use these as explanations for the complex models local behavior..

https://cran.r-project.org/web/packages/lime/vignettes/Understanding_lime.html

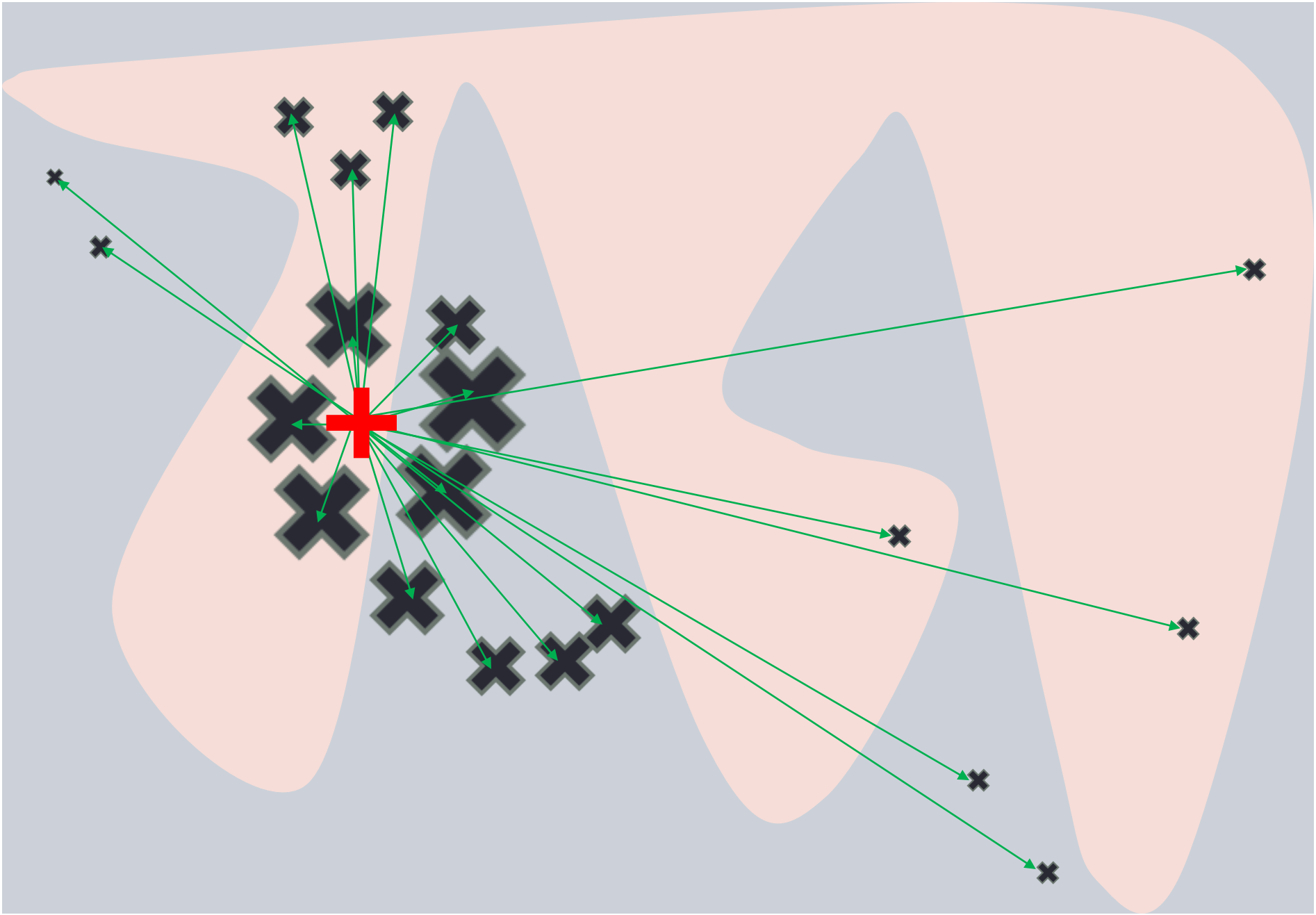
LIME : 1. Predict an observation with complex model



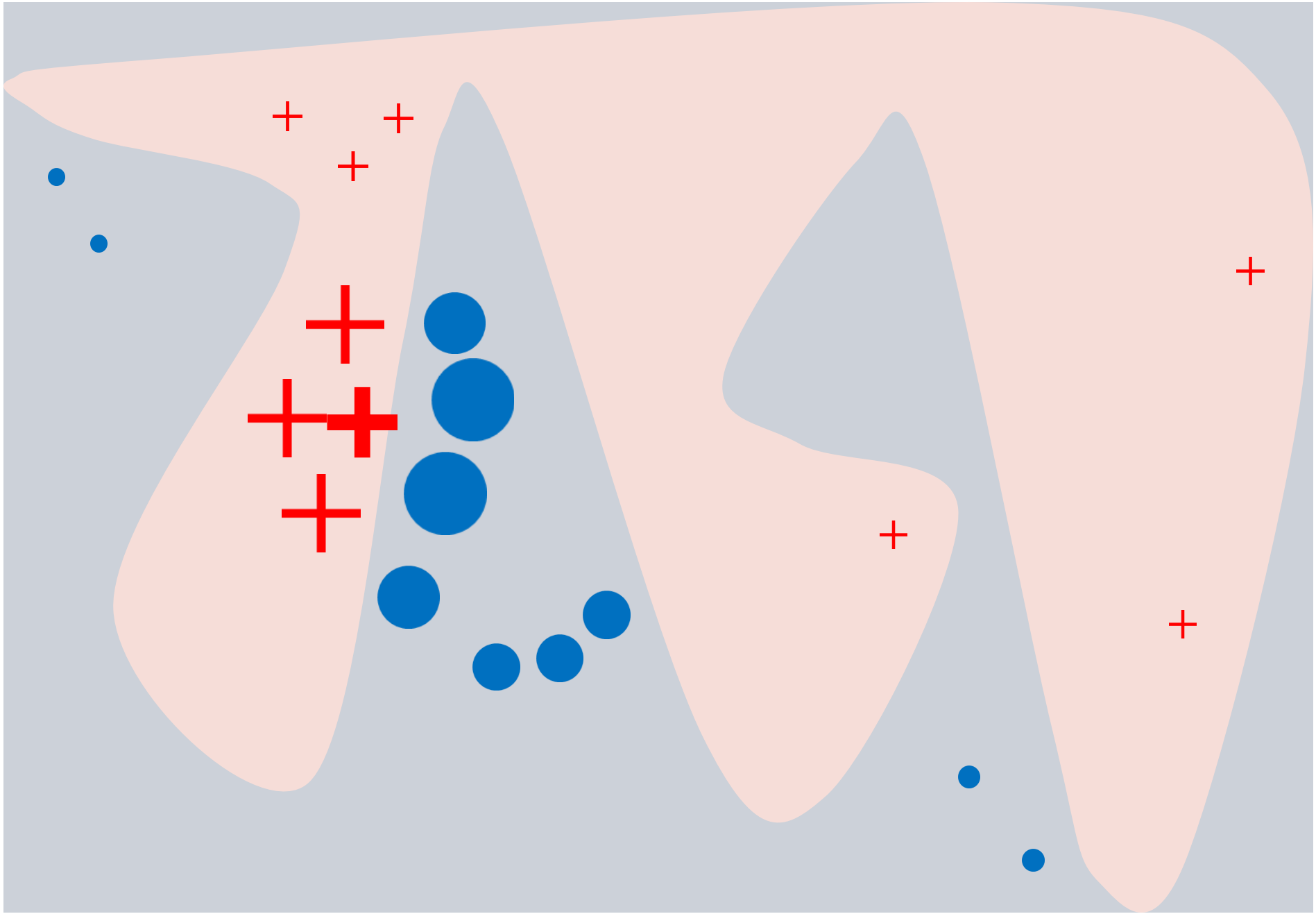
LIME : 2. Scatter instances with Permutated feature values



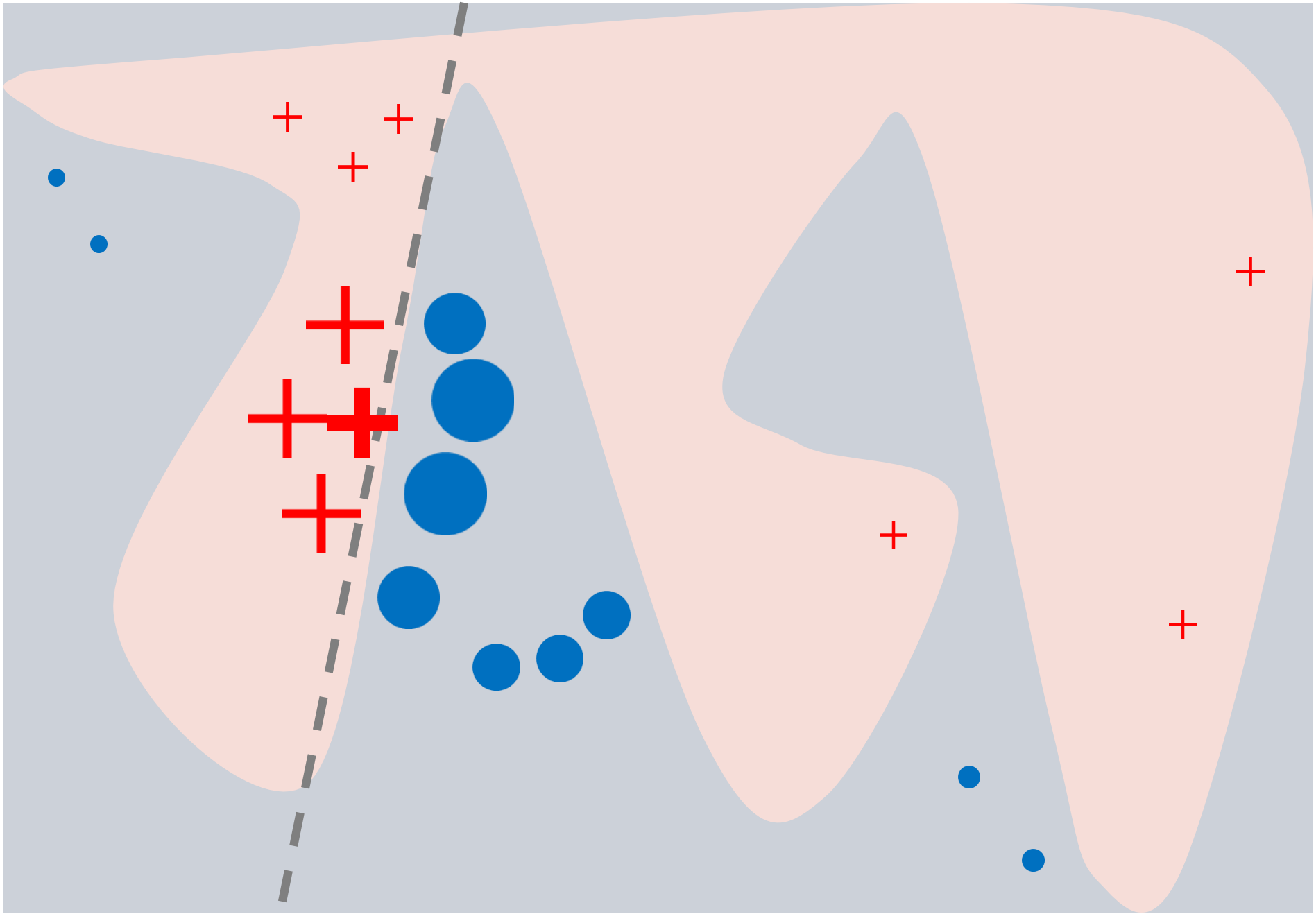
LIME : 3. Scoring similarity distance from observation to permutations



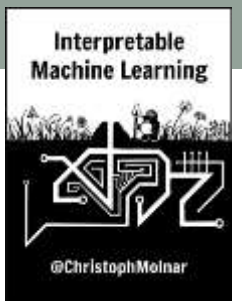
LIME : 4. Predict permutations with Complex model

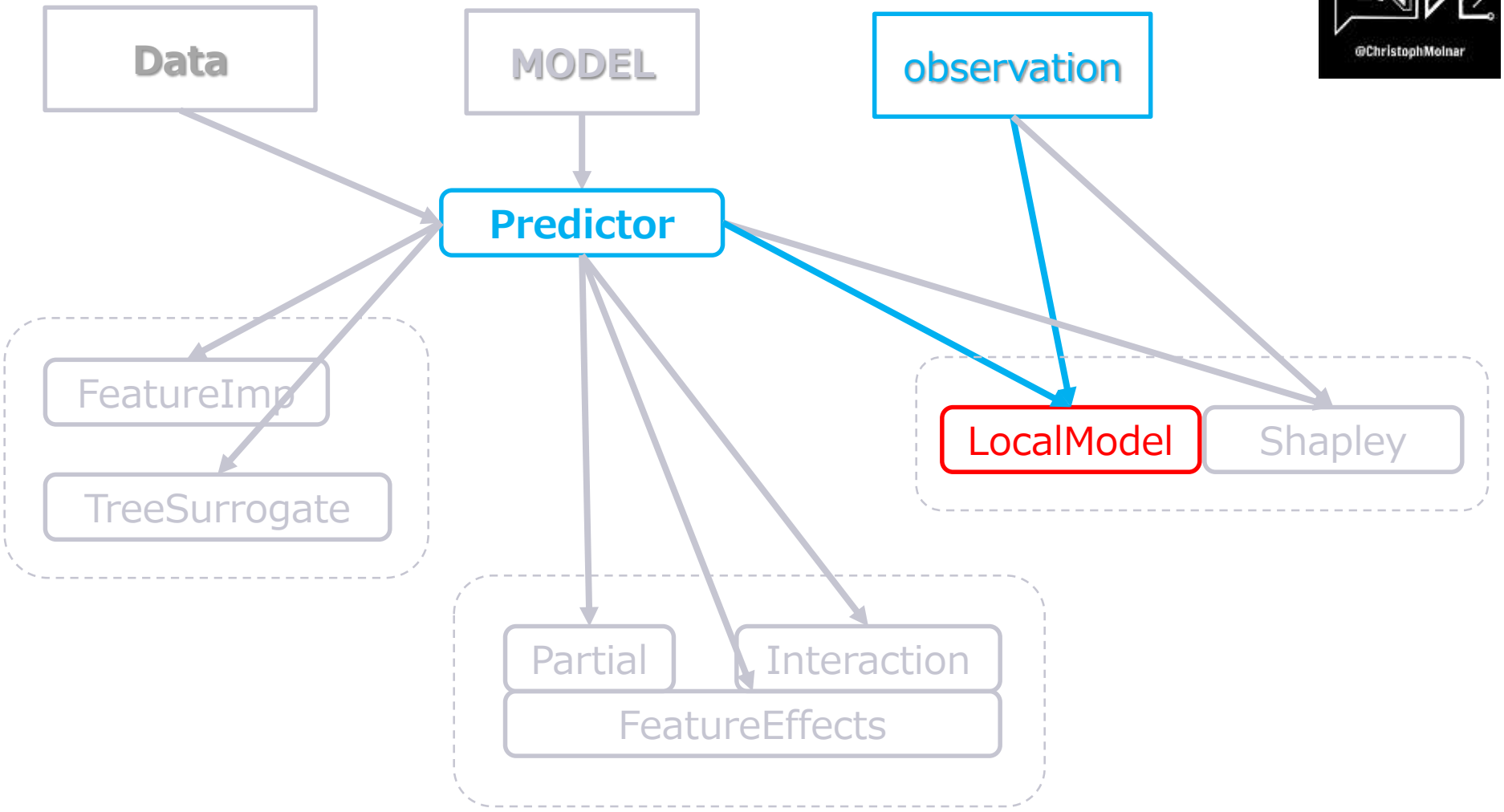


LIME : 5. Simple model for weighted permutations



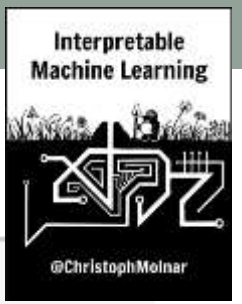
iml::LocalModel() ／LIME





https://github.com/christophM/iml

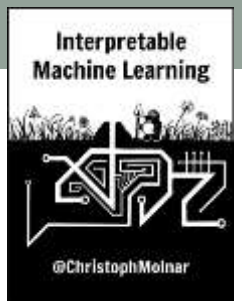
iml::LocalModel() ／LIME




```
lime.explain <- LocalModel$new(predictor.rf, x.interest = X[1,])
lime.explain$results
```

	beta	x.recoded	effect	x.original
surface	-4.579322	131	-599.89123	131
no.rooms	-2.538297	5	-12.69148	5
district=Srodmiescie	568.030712	1	568.03071	Srodmiescie

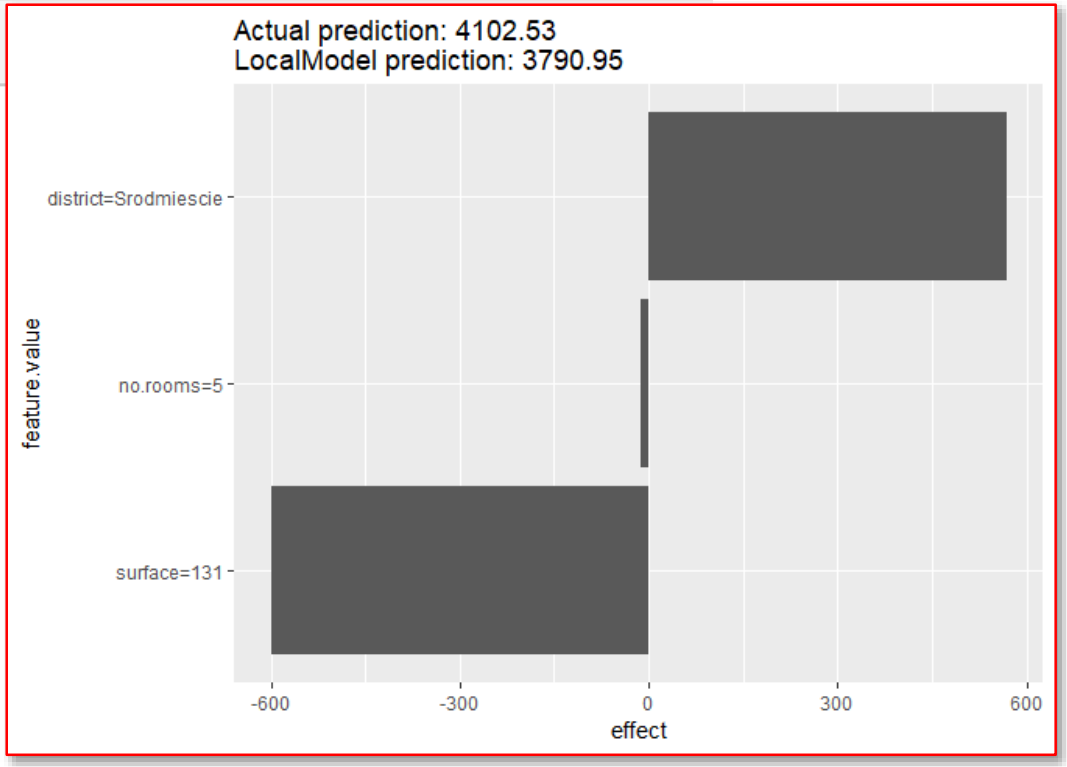
iml::LocalModel() ／LIME



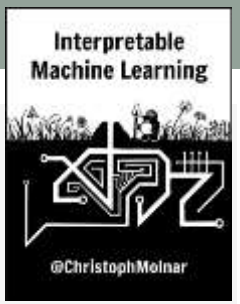

```
lime.explain <- LocalModel$new(predictor.rf, x.interest = X[1,])
lime.explain$results
```

```
                          beta x.recoded     effect  x.original
surface                -4.579322       131 -599.89123         131
no.rooms               -2.538297         5  -12.69148           5
district=Srodmiescie 568.030712         1  568.03071 Srodmiescie
                                  feature        feature.value
surface                           surface          surface=131
no.rooms                        no.rooms           no.rooms=5
district=Srodmiescie district=Srodmiescie district=Srodmiescie
```

```
plot(lime.explain)
```

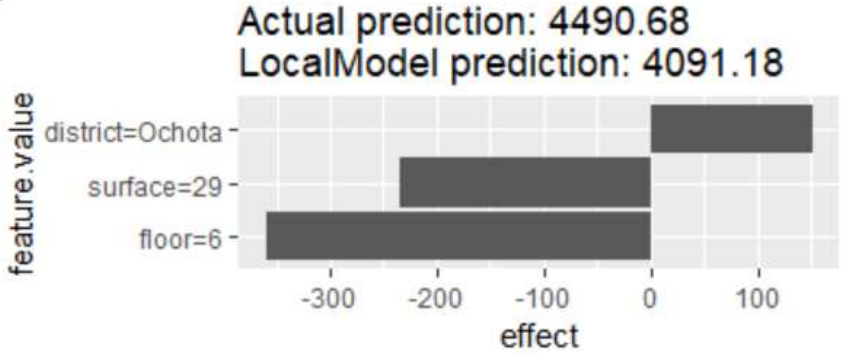


iml::LocalModel() ／LIME

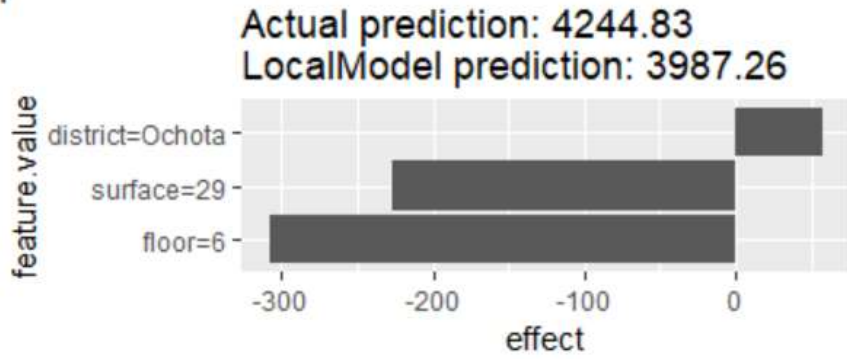



```
lime <- plime <- list()

for(model.name in model.labels){
  lime[[model.name]]  <- LocalModel$new(predictor[[model.name]], x.interest = X[10,])
  plime[[model.name]] <- plot(lime[[model.name]]) + labs(tag = model.name)
}

gridExtra::grid.arrange(grobs = plime, ncol=2)
```

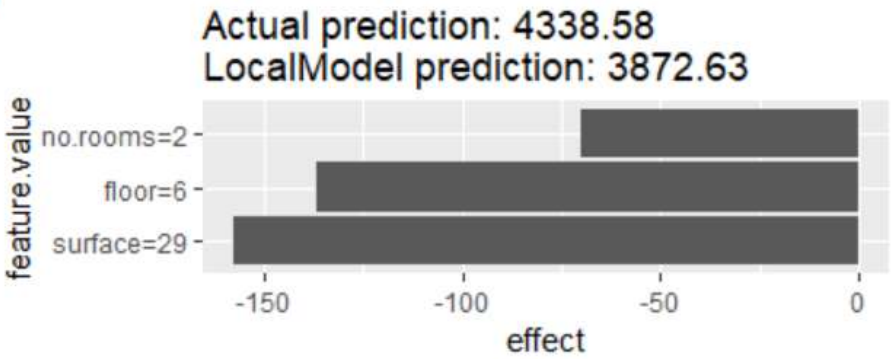
enet



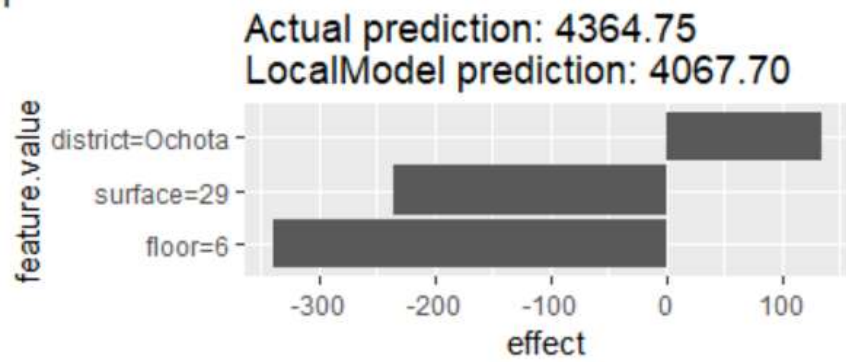
svm



rf



gbm



Advantage

- 表形式のデータ、テキスト、および画像のどれに対しても機能する。
- ヒトに優しい説明になりやすい
  - 得られる説明は短く（＝選択的）違いが分かりやすい

Disadvantage

- あくまで近似
  - ある特徴の貢献（attributions）を、完全に表すわけではない
- チューニングに注意
  - カーネルの選定と設定次第で推定が変わる
    - 説明モデルの複雑さはユーザーが事前に定義する必要がある
  - サンプリングデータ点は、特徴間の相関を無視して、ガウス分布からサンプリングされる（必ずしも特徴空間での近傍と限らない）
  - ブラックボックスモデル（の決定境界）の複雑性に依存して説明が安定しない。サンプリングプロセスを繰り返すと、説明が変わる可能性がある。

Shapley value

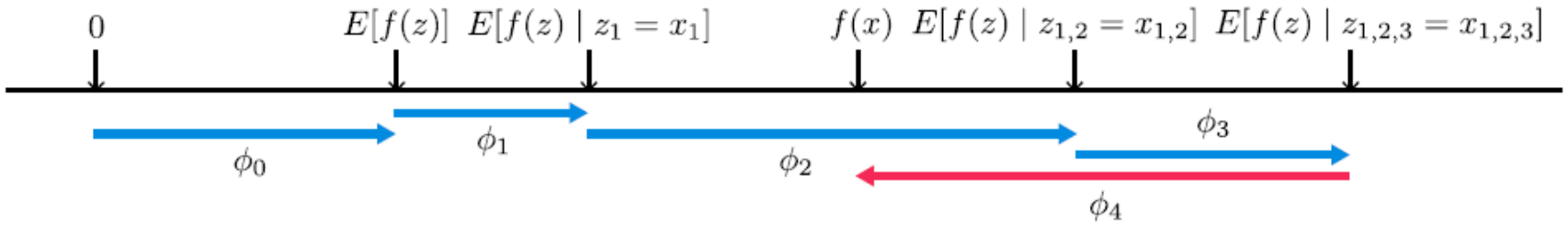
policy	method name	iml	DALEX
understand entire model	residuals and goodness of fit	✗	✓
	permutation importance	✓	✓
	global surrogate	Tree surrogate	✗
understand feature(s)	Merging Path Plot (PDP for categorical data)	✗	✓
	Partial Dependence Plot (PDP for continuous data)	✓	✓
	Individual Conditional Expectation (ICE)	✓	Ceteris Paribus Plots
	Accumulated Local Effects (ALE) Plot	✓	✓
	Feature Interaction	✓	✗
local interpretation (for single prediction)	LIME	✓	✗
	SHAPLY value	✓	✗
	breakDown	✗	✓

Shapley value

- ある予測が得られる過程を協力ゲームと考える：
  - 予測値＝「報酬」
  - 各変数＝ゲームの「プレーヤー」
- 各変数の貢献度を、特徴間で「報酬」を公平に配分する
  - 協力した　＝　元の予測値
  - 協力しない＝　変数をシャッフルしたときの予測値
  - 両者の差分をすべての組み合わせで評価する
  - 特徴量を取り除いて学習したモデルの予測値と元のモデルの予測値との差ではないことに注意。

$$\phi_i(f, x) = \sum_{z' \subseteq x'} \frac{|z'|!(M - |z'| - 1)!}{M!} [f_x(z') - f_x(z' \setminus i)] \tag{8}$$

where $|z'|$ is the number of non-zero entries in z', and $z' \subseteq x'$ represents all z' vectors where the non-zero entries are a subset of the non-zero entries in x'.



https://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions.pdf

推定ステップ

1. Mは、観察Xについて、x_iを含む変数の組み合わせ
2. 全ての組合せについて、Mの予測値とM からx_iの情報をシャッフルした予測値との差を計算して平均する
3. 全ての特徴iについて、１～２を繰り返す

X			f(X)	f^i(X)	diff	Φi
x1	**x2**	x3	with xi	without xi	= f(X) - f^i(X)	= mean(diff)
1	0	0	15			
0	0	0	5	5	10	
1	1	0	40			
0	1	0	45	45	-5	
1	0	1	65			
0	0	1	50	50	15	
1	1	1	35			
0	1	1	40	40	-5	
						Φ1＝3.75

推定ステップ

1. Mは、観察Xについて、x_iを含む変数の組み合わせ
2. 全ての組合せについて、Mの予測値とMからx_iの情報をシャッフルした予測値との差を計算して平均する
3. 全ての特徴iについて、１～２を繰り返す

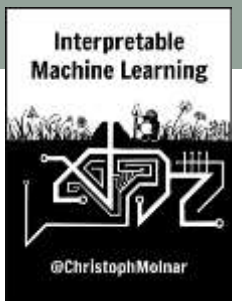
X			f(X)	f^i(X)	diff	Φi
x1	**x2**	x3	with xi	without xi	= f(X) - f^i(X)	= mean(diff)
0	1	0	45			
0	**0**	0	5	5	40	
1	1	0	40			
1	**0**	0	15	15	25	
0	1	1	40			
0	**0**	1	50	50	-10	
1	1	1	35			
1	**0**	1	65	65	-30	
						Φ2＝6.25

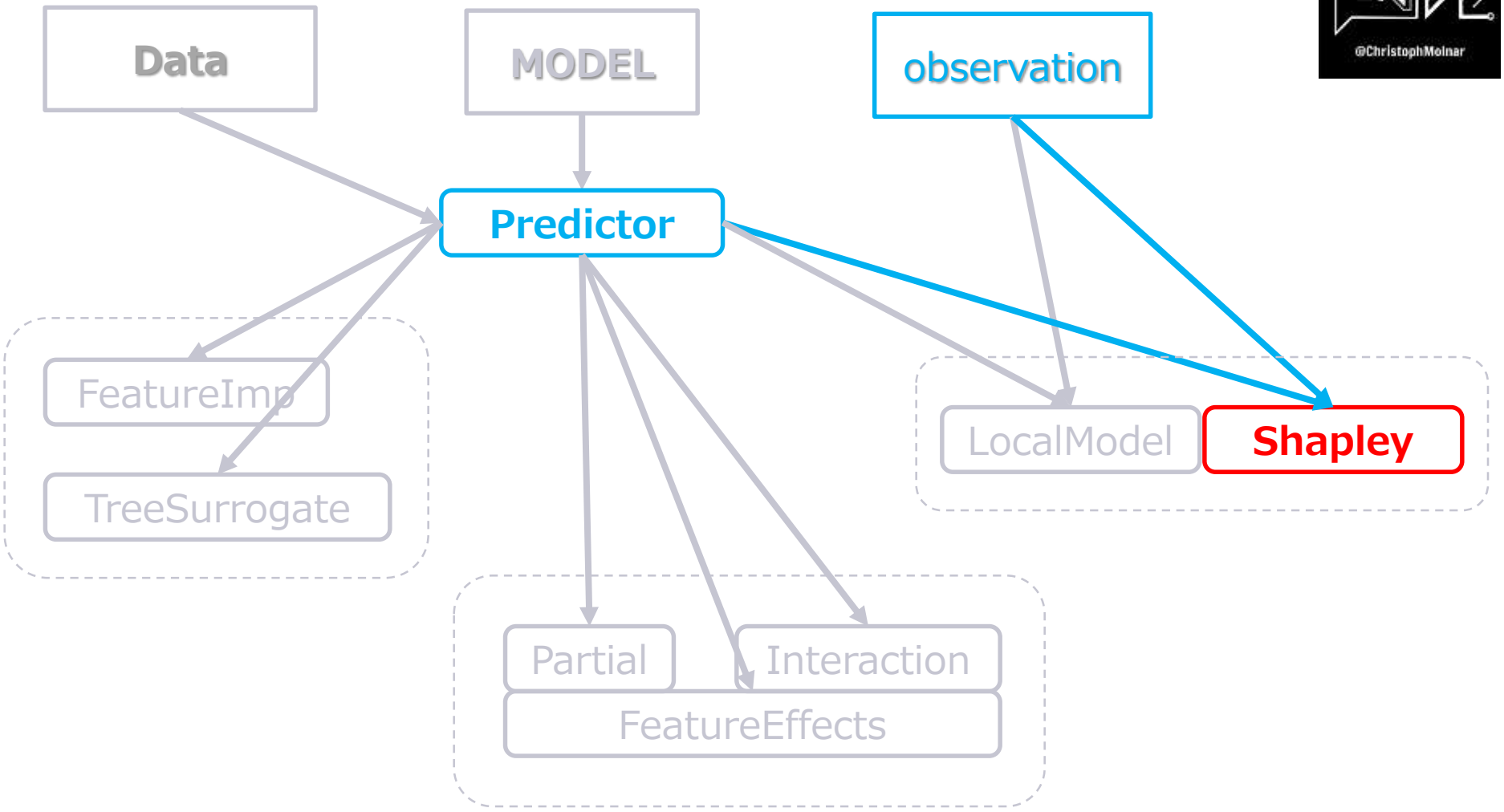
推定ステップ

1. Mは、観察Xについて、x_iを含む変数の組み合わせ
2. 全ての組合せについて、Mの予測値とMからx_iの情報をシャッフルした予測値との差を計算して平均する
3. 全ての特徴iについて、１～２を繰り返す

X			f(X)	f^i(X)	diff	Φi
x1	x2	**x3**	with xi	without xi	= f(X) - f^i(X)	= mean(diff)
0	0	1	50			
0	0	**0**	5	5	10	
1	0	1	65			
1	0	**0**	15	15	50	
0	1	1	40			
0	1	**0**	45	45	-5	
1	1	1	35			
1	1	**0**	40	40	-5	
						Φ3＝12.5

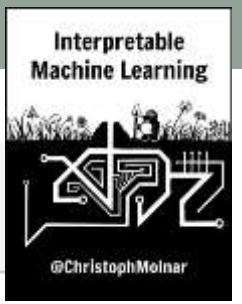
Iml::Shapley



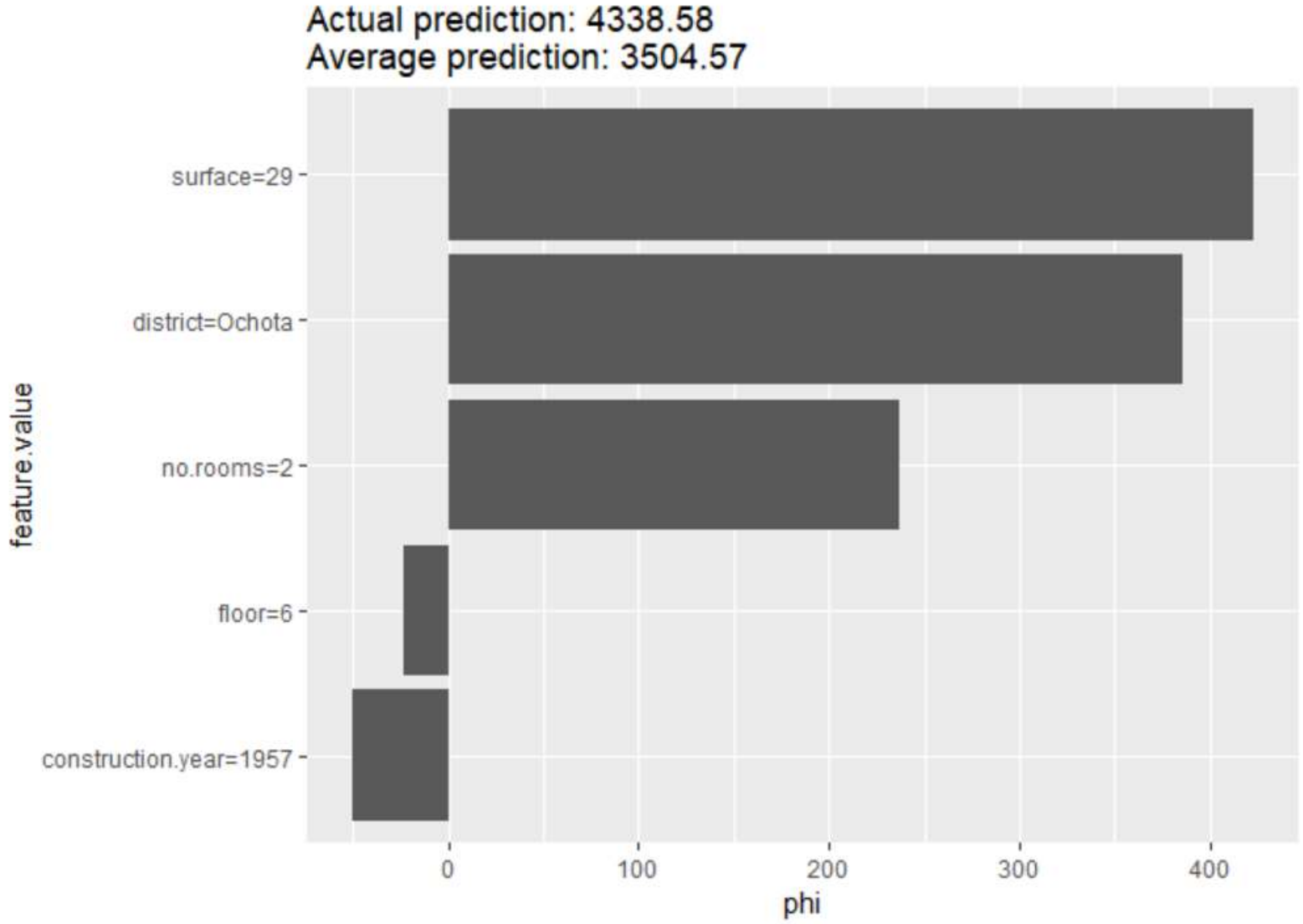


https://github.com/christophM/iml

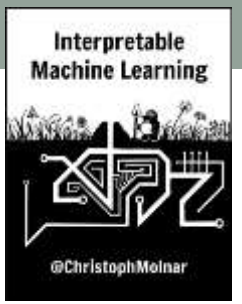
Iml::Shapley$new()




```
shapley <- Shapley$new(predictor.rf, x.interest = X[10,])
plot(shapley)
```

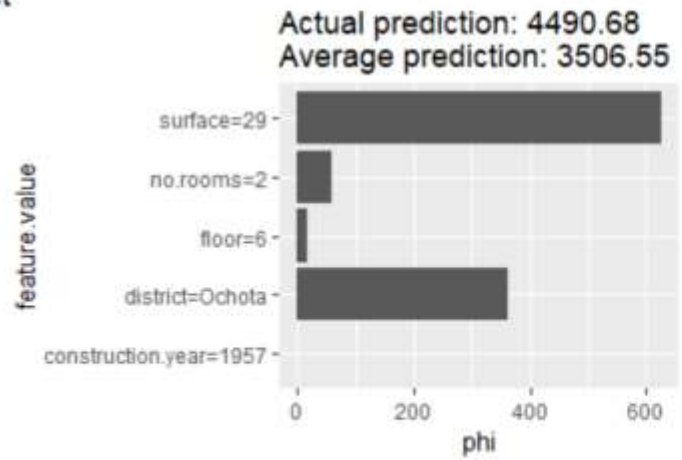


Iml::Shapley

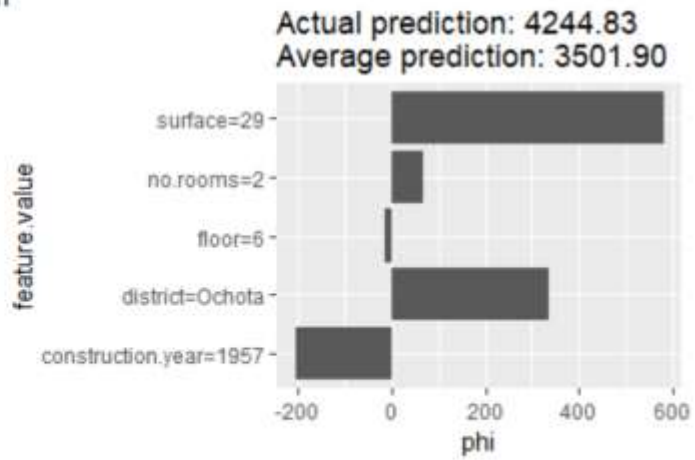



```
set.seed(8)
shap <- pshap <- list()

for(model.name in model.labels){
  shap[[model.name]]  <- Shapley$new(predictor[[model.name]], x.interest = X[10,])
  pshap[[model.name]] <- plot(shap[[model.name]], sort=FALSE) + labs(tag = model.name)
}

gridExtra::grid.arrange(grobs = pshap, ncol=2)
```

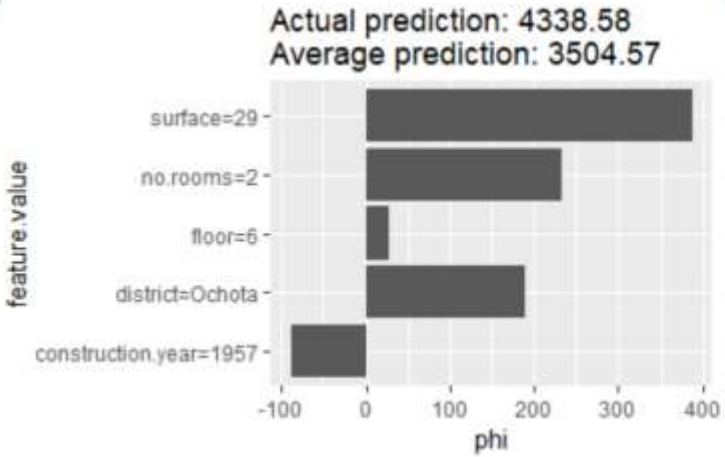
enet



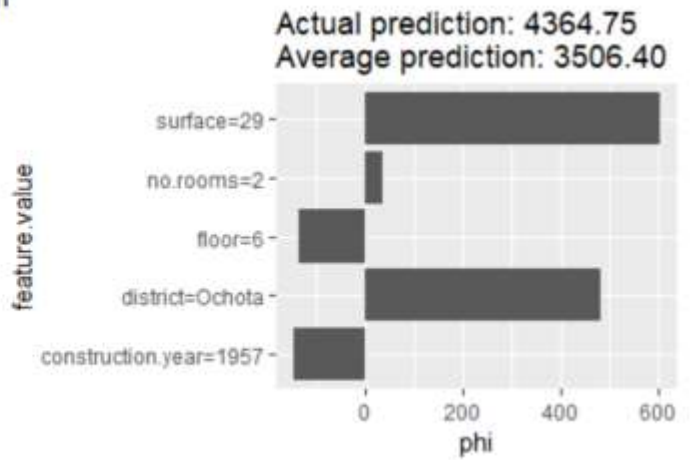
svm



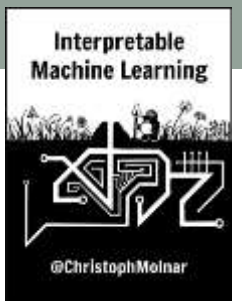
rf



gbm



Iml::Shapley



- すべての観察についてすべての特徴量の組み合わせを総当たりにすると、
  計算コストが高いので、imlではモンテカルロサンプリングによる近似を採用

$$\hat{\phi}_j = \frac{1}{M}\sum_{m=1}^{M}\left(\hat{f}\left(x_{+j}^m\right) - \hat{f}\left(x_{-j}^m\right)\right)$$

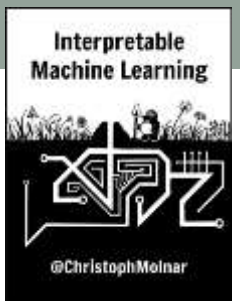
Approximate Shapley estimation for single feature value:

- Output: Shapley value for the value of the j-th feature
- Required: Number of iterations M, instance of interest x, feature index j, data matrix X, and machine learning model f
- For all m = 1,...,M:
  - Draw random instance z from the data matrix X
  - Choose a random permutation o of the feature values
  - Order instance x: $x_o = (x_{(1)}, \ldots, x_{(j)}, \ldots, x_{(p)})$
  - Order instance z: $z_o = (z_{(1)}, \ldots, z_{(j)}, \ldots, z_{(p)})$
  - Construct two new instances
  - $x_{+j} = (x_{(1)}, \ldots, x_{(j-1)}, x_{(j)}, z_{(j+1)}, \ldots, z_{(p)})$
  - $x_{-j} = (x_{(1)}, \ldots, x_{(j-1)}, z_{(j)}, z_{(j+1)}, \ldots, z_{(p)})$
  - $\phi_j^m = \hat{f}(x_{+j}) - \hat{f}(x_{-j})$
- Compute Shapley value as the average: $\phi_j(x) = \frac{1}{M}\sum_{m=1}^{M}\phi_j^m$

https://christophm.github.io/interpretable-ml-book/shapley.html

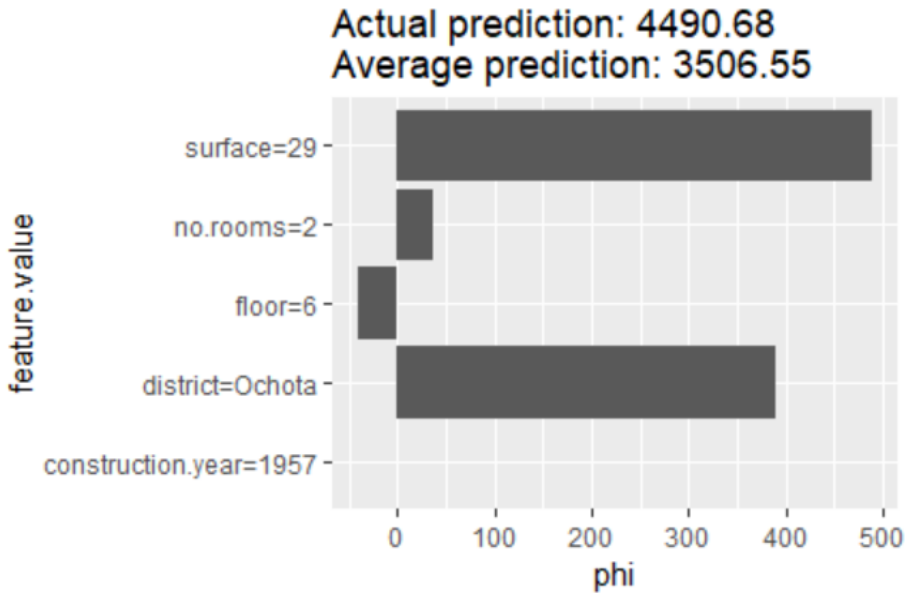
Štrumbelj, Erik, and Igor Kononenko. “Explaining prediction models and individual predictions with feature contributions.” Knowledge and information systems 41.3 (2014): 647-665.

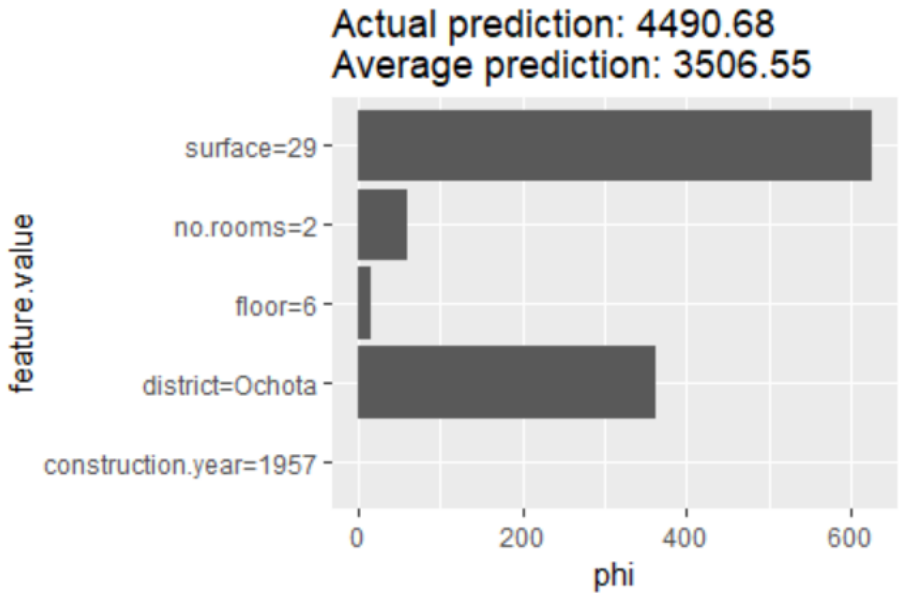
Iml::Shapley(…, sample.size =100)



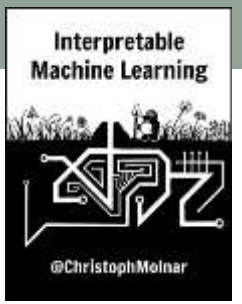
- すべての観察についてすべての特徴量の組み合わせを総当たりにすると、
  計算コストが高いので、imlではモンテカルロサンプリングによる近似を採用
- そのため、サンプリングが小数だと推定結果が不安定に。

```
set.seed(1)
shapley.1 <- Shapley$new(predictor[["enet"]], x.interest = X[10,])
p1 <- plot(shapley.1, sort=FALSE)

set.seed(8)
shapley.2 <- Shapley$new(predictor[["enet"]], x.interest = X[10,])
p2 <- plot(shapley.2, sort=FALSE)

gridExtra::grid.arrange(p1, p2, ncol=2)
```



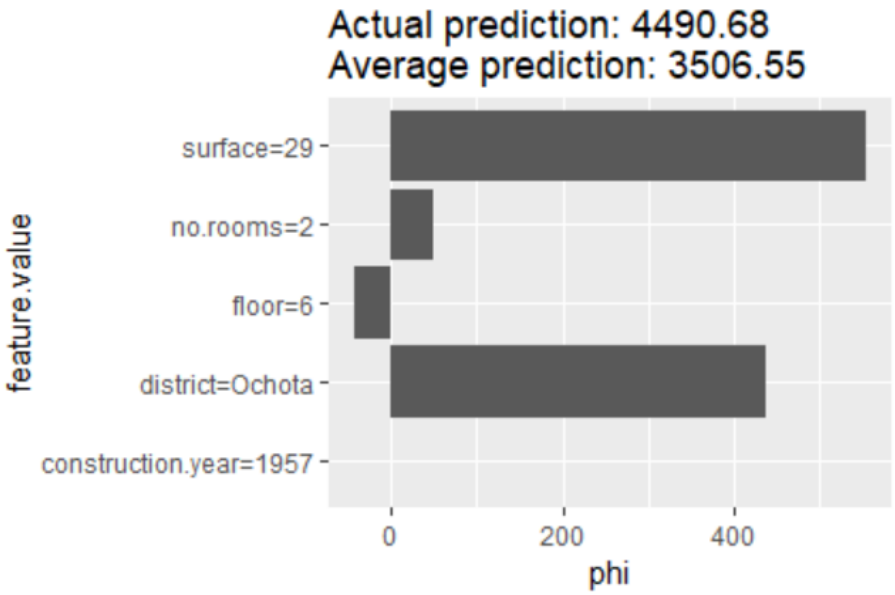


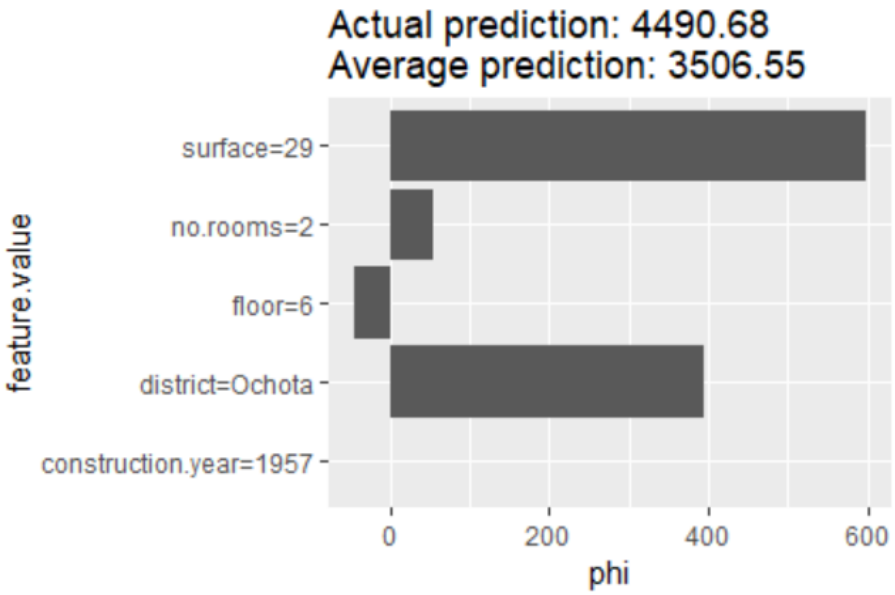
Iml::Shapley(…, sample.size = 100 *10)



- **サンプルサイズを大きくする**と推定結果は安定する。
- 計算時間は増える。

```
set.seed(1)
shapley.1 <- Shapley$new(predictor[["enet"]], x.interest = X[10,], sample.size = 100 * 10)
p1 <- plot(shapley.1, sort=FALSE)

set.seed(8)
shapley.2 <- Shapley$new(predictor[["enet"]], x.interest = X[10,], sample.size = 100 * 10)
p2 <- plot(shapley.2, sort=FALSE)

gridExtra::grid.arrange(p1, p2, ncol=2)
```





Iml::Shapley()

Advantage

- 完全な説明を提供する強固な理論保証がある
- 単一のデータポイント（またはサブセット）に対する説明ができる
  - LIMEの場合、あくまでもモデル全体の振る舞いを局所的に近似することで「なぜ予測が得られたのか」説明する

Disadvantage

- 計算コストが高い
  - 完全な説明を提供するためには2^kの組み合わせを総当たりする必要がある
  - 現実的にはサンプリングによる近似となるが、計算時間と分散の大きさはトレードオフ。
- 説明がスパースにならない
  - 常にすべての特徴を使って説明を生成する
- 訓練セットのデータを保持しておく必要がある。
  - 新しい観察データのShapley値を計算する場合は、予測関数とデータが両方必要。
- 特徴ごとに単純な値を返す
  - 入力の変化に対する予測の変化は説明できない。
- 特徴が相関するときに非現実的な置換が起き、推定の中に入れてしまう。

breakDown

policy	method name	iml	DALEX
understand entire model	residuals and goodness of fit	✗	✓
	permutation importance	✓	✓
	global surrogate	Tree surrogate	✗
understand feature(s)	Merging Path Plot (PDP for categorical data)	✗	✓
	Partial Dependence Plot (PDP for continuous data)	✓	✓
	Individual Conditional Expectation (ICE)	✓	Ceteris Paribus Plots
	Accumulated Local Effects (ALE) Plot	✓	✓
	Feature Interaction	✓	✗
local interpretation (for single prediction)	LIME	✓	✗
	SHAPLY value	✓	✗
	breakDown	✗	✓

breakDown

- ある観察 xについて、変数選択後の予測値と、もとの予測値　との距離を１変数ずつ繰り返し評価する。
  - 増減の順序がキモになるが、当然、組み合わせ爆発を起こす。
  - breakDownパッケージでは、stepwise selectionを採用。
- **Step-down**:
  - いわゆる Backward elimination
  - full model → model with empty set
  - ロスを最小にするように変数を減らす
- **Step-up**
  - いわゆる Forward selection
  - model with empty set → Full model
  - ゲインを最大にするように変数を増やす

https://arxiv.org/abs/1804.01955

- ある観察 xについて、変数選択後の予測値と、もとの予測値　との距離を１変数ずつ繰り返し評価する。
  - 増減の順序がキモになるが、当然、組み合わせ爆発を起こす。
  - breakDownパッケージでは、stepwise selectionを採用。
- **Step-down**:
  - いわゆる Backward elimination
  - full model → model with empty set
  - ロスを最小にするように変数を減らす

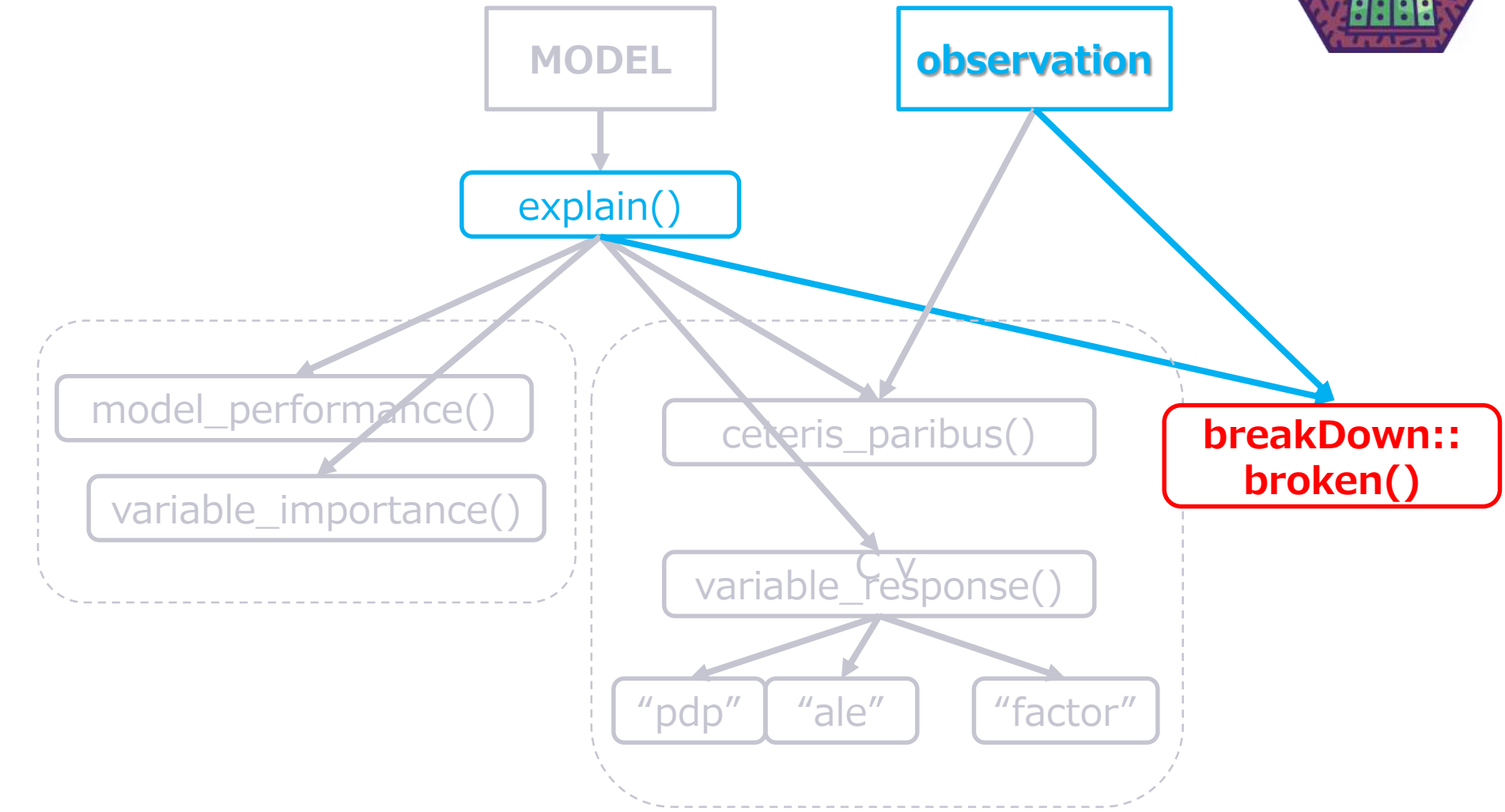
Algorithm 2 Model agnostic break down of model predictions. The *step-down* approach.

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

Step-upは、この逆をする (model with empty set → Full model)

https://arxiv.org/abs/1804.01955

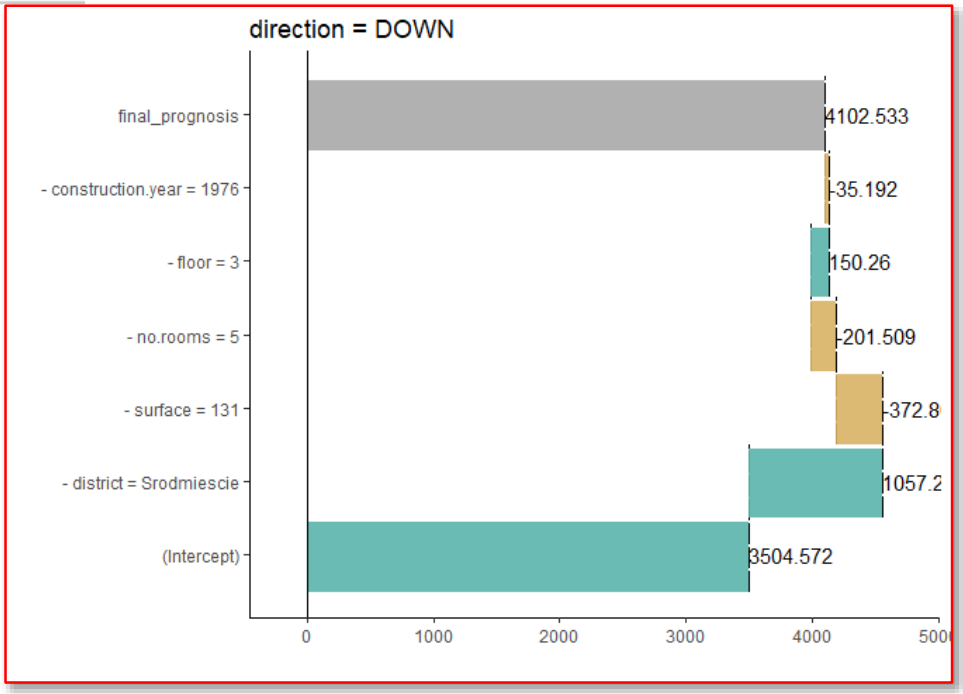
breakDown



https://pbiecek.github.io/DALEX_docs/2-architecture.html#architecture

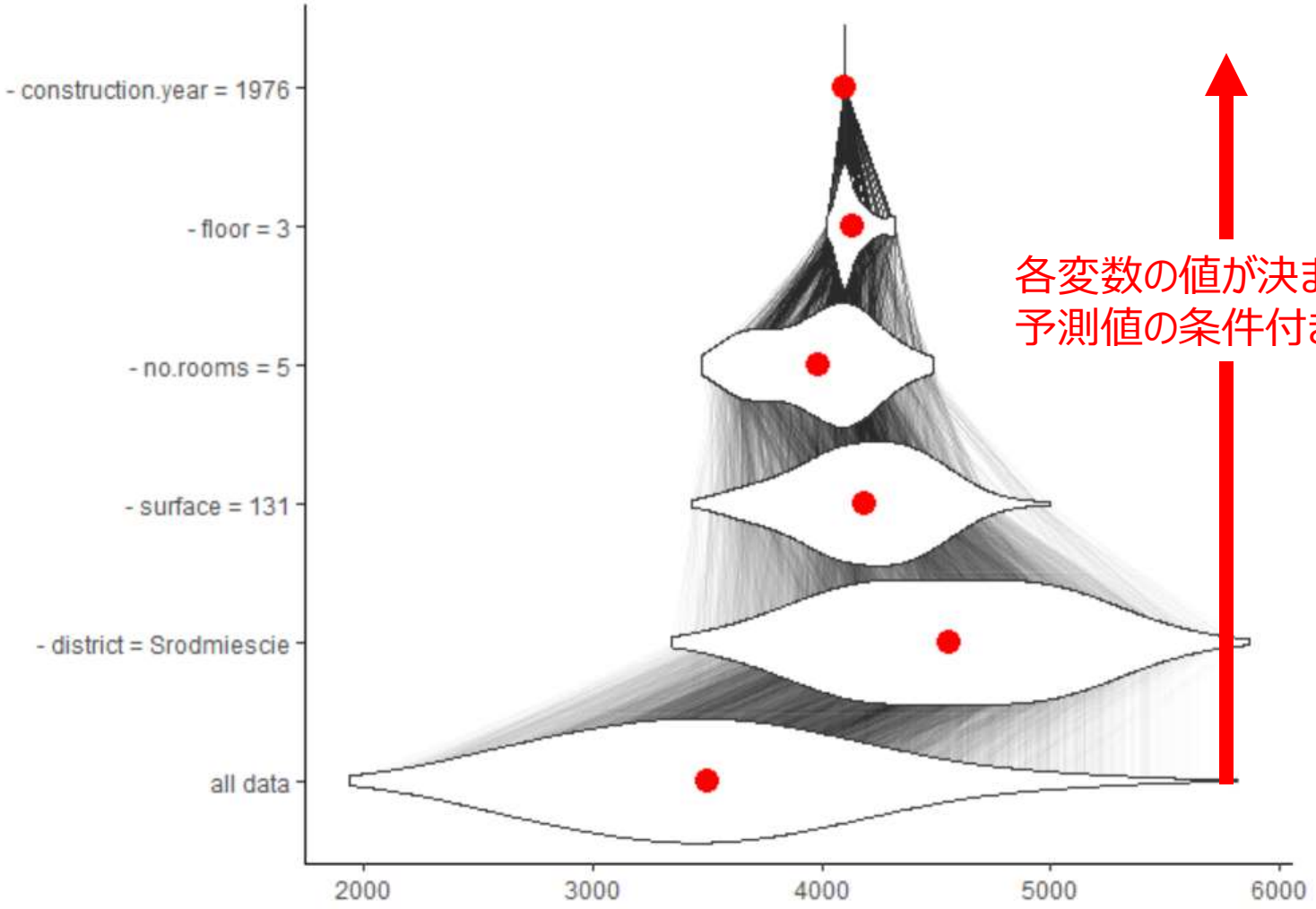
breakDown

```
data("apartmentsTest", package = "DALEX")
X <- apartmentsTest[,-1]

target <- 1

br.down.rf <- broken(
  model = tuned.model[["rf"]],
  data  = X,
  new_observation = X[target, ],
  direction = "down",
  predict.function = predictMLR,
  keep_distributions=TRUE)

pbr.down.rf <- plot(br.down.rf) + ggtitle("direction = DOWN")
pbr.down.rf
```



breakdown (Conditional proportion plot)

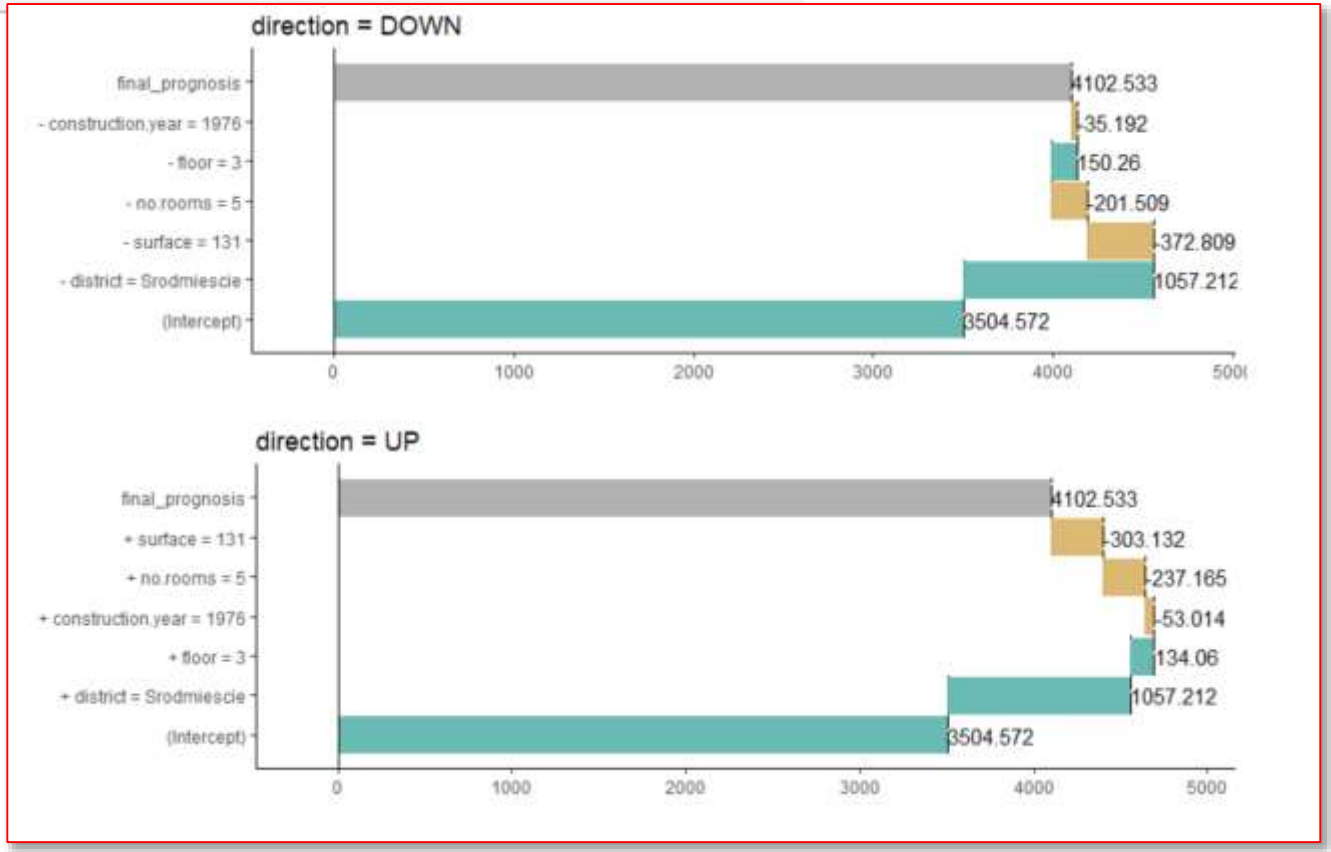
```
plot(br.down.rf, plot_distributions =TRUE)
```



各変数の値が決まるごとに
予測値の条件付き分布が収束

breakdown (Step-down & Step-up)

```
br.up.rf <- broken(
  model = tuned.model[["rf"]],
  data  = X,
  new_observation = X[target, ],
  direction = "up",
  predict.function = predictMLR,
  keep_distributions=TRUE)

pbr.up.rf <- plot(br.up.rf) + ggtitle("direction = UP")

gridExtra::grid.arrange(pbr.down.rf, pbr.up.rf, ncol=1)
```



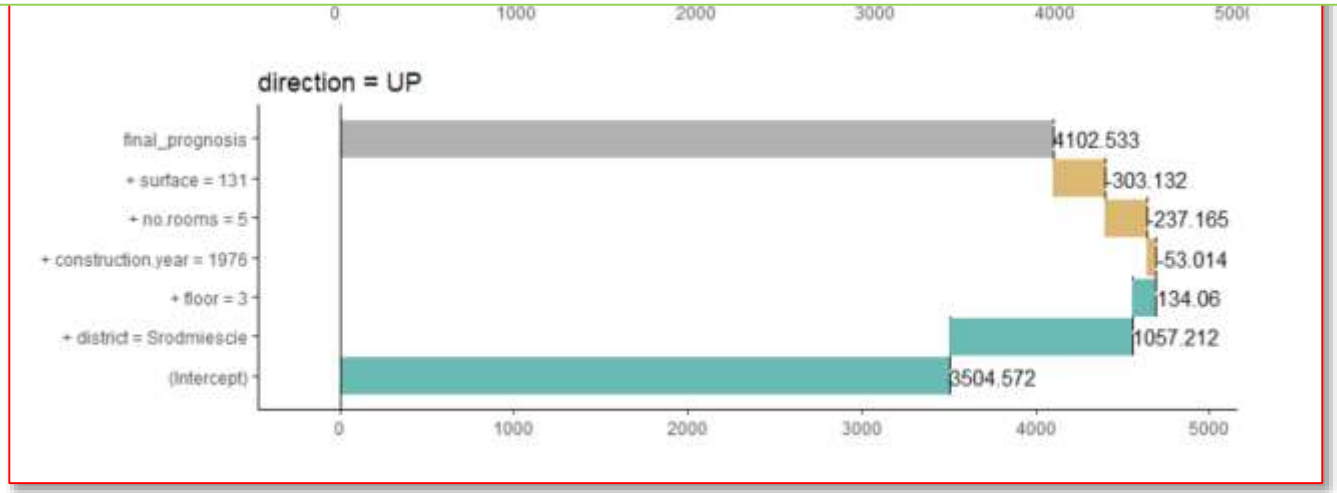
breakdown (Step-down & Step-up)

```
br.up.rf <- broken(
  model = tuned.model[["rf"]],
  data  = X,
  new_observation = X[target, ],
  direction = "up",
  predict.function = predictMLR,
  keep_distributions=TRUE)

pbr.up.rf <- plot(br.up.rf) + ggtitle("direction = UP")

gridExtra:
```

完全な加法モデルでは、予測結果の完全説明になる。

$$y = \sum(w \cdot x) + \text{Intercept}$$

breakDownパッケージではlm, glmに適用可能



Advantage

- 元の予測値の再構成になっている

Disadvantage

- ステップワイズ法の性質を引き継ぐ
  - 変数の評価順番を決めるので、局所解にはまりやすい
- 説明がスパースにならない
  - すべての特徴を使って説明を生成する
- 特徴ごとに単純な値を返す
  - LIMEのように予測モデルを返すわけではないので、入力の変化に対する予測の変化は説明できない。

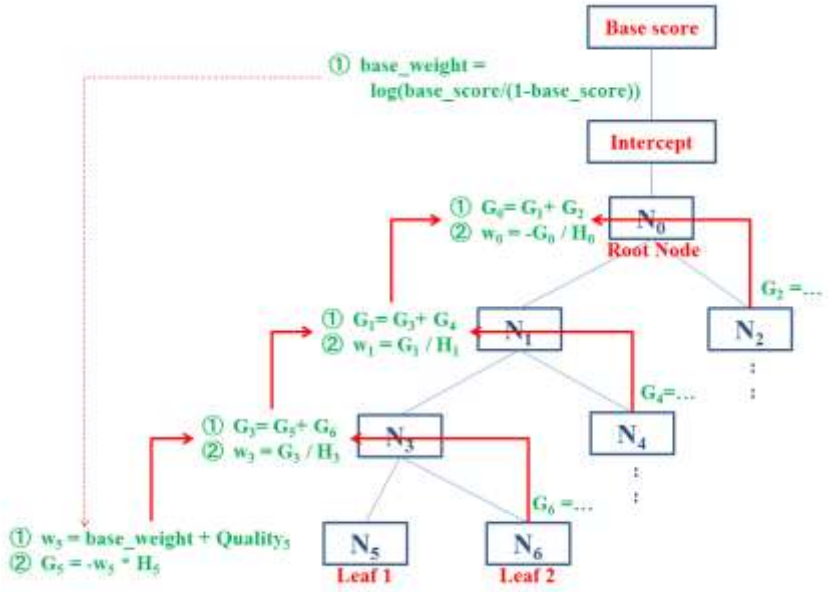
（参考） 予測値の再構成

モデル依存的な方法では、予測結果がどのように得られたか
直接理解・抽出できるものも多い

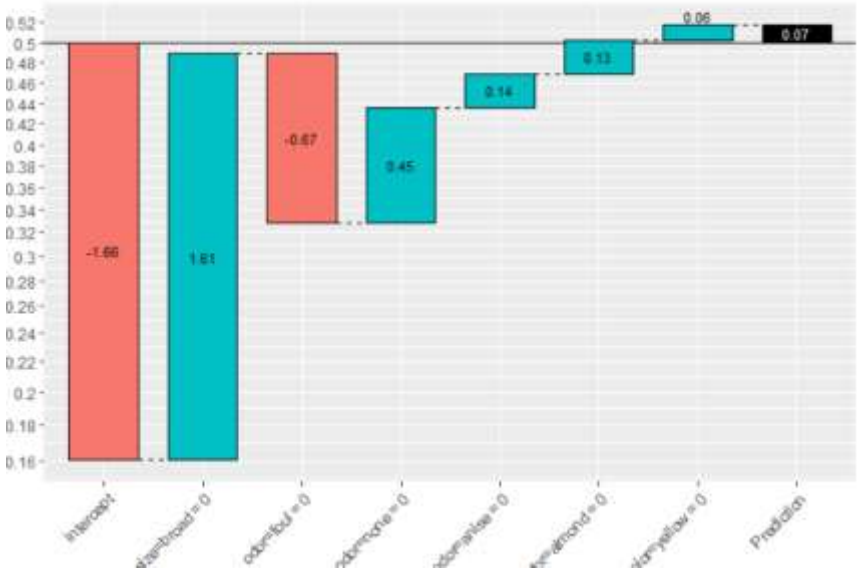
	Intrinsic	Post hoc
Model-Specific Methods	• **Linear Regression** • **Logistic Regression** • **GLM, GAM and more** • **Decision Tree** • **Decision Rules** • **RuleFit** • Naive Bayes Classifier • K-Nearest Neighbors	• Feature Importance (OOB error@RF; gain/cover/weight @XGB) • Feature Contribution (forestFloor@RF, **XGBoostexplainer, lightgbmExplainer**) • Alternate / Enumerate lasso (@LASSO) • inTrees / defragTrees (@RF) • **Actionable feature tweaking (@RF/XGB)**
Model-Agnostic Methods	Intrinsic interpretable Model にも適用可能	• Partial Dependence Plot • Individual Conditional Expectation • Accumulated Local Effects Plot • Feature Interaction • Permutation Feature Importance • Global Surrogate • Local Explanation (LIME, Shapley Values, breakDown)
Example-based Explanations	??	• Counterfactual Explanations • Adversarial Examples • Prototypes and Criticisms • Influential Instances

XGBoostExplainer

- ある観察に対して抽出した予測パスから、予測結果の分解再構成を行う。
- 再推定ではなく、XGBoostの予測を完全に再現する




```
#> Prediction:  0.5178885
#> Weight:  0.07158444
#> Breakdown
#>        intercept  gill-size=broad        odor=foul        odor=none
#>      -1.65714093       1.61423045      -0.67129347       0.45408751
#>       odor=anise      odor=almond cap-color=yellow
#>       0.13628094       0.13073006       0.06468987
```

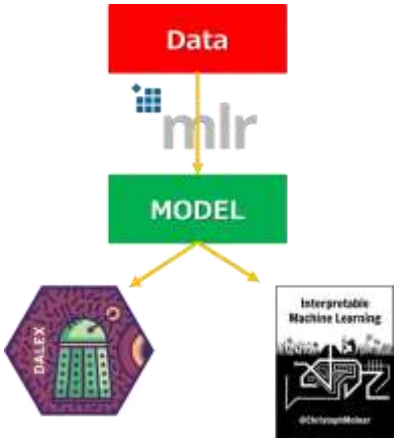


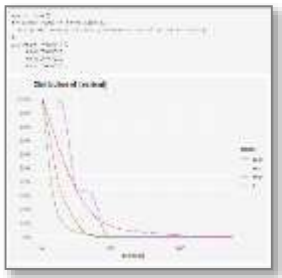
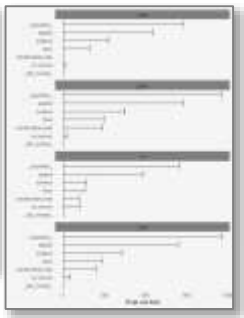
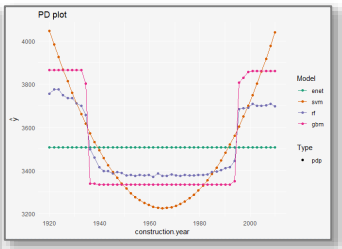
http://kato-kohaku-0.hatenablog.com/entry/2018/12/14/002253

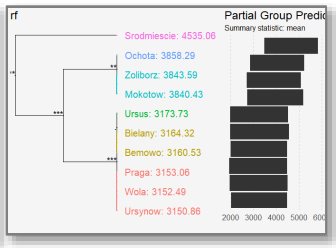
（機械学習）モデルの説明用パッケージをいくつか紹介しました

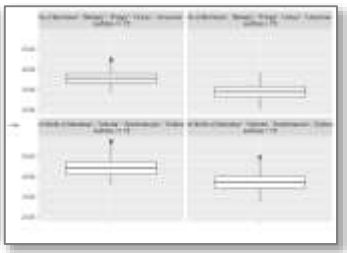
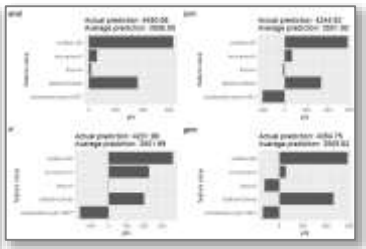
	Intrinsic	Post hoc
Model-Specific Methods	• Linear Regression • Logistic Regression • GLM, GAM and more • Decision Tree • Decision Rules • RuleFit • Naive Bayes Classifier • K-Nearest Neighbors	• Feature Importance (OOB error@RF; gain/cover/weight @XGB) • Feature Contribution (forestFloor@RF, XGBoostexplainer, lightgbmExplainer) • Alternate / Enumerate lasso (@LASSO) • inTrees / defragTrees (@RF) • Actionable feature tweaking (@RF/XGB)
Model-Agnostic Methods	Intrins Model にも適用可能	• Partial Dependence Plot • Individual Conditional Expectation Accumulated Local Effects Plot • Feature Interaction • Permutation Feature Importance • Global Surrogate • Local Surrogate (LIME, Shapley Values)
Example-based Explanations	??	• Counterfactual Explanations • Adversarial Examples • Prototypes and Criticisms • Influential Instances

DALEX & iml package





Interpretability book

Christoph Molnar(2019) "Interpretable Machine Learning: A Guide for Making Black Box Models Explainable."

- https://christophm.github.io/interpretable-ml-book/

Przemysław Biecek (2018) "DALEX: Descriptive mAchine Learning EXplanations"

- https://pbiecek.github.io/DALEX_docs/

総説（和文）

原 聡 私のブックマーク「機械学習における解釈性（Interpretability in Machine Learning）」

- 人工知能33巻3号（2018年5月）
- https://www.ai-gakkai.or.jp/my-bookmark_vol33-no3/

原 聡「機械学習モデルの判断根拠の説明」

- 第20回ステアラボ人工知能セミナー (2018年12月21日)
- https://www.youtube.com/watch?v=Fgza_C6KphU
- https://www.slideshare.net/SatoshiHara3/ss-126157179

吉永 尊洸「機械学習と解釈可能性」

- ソフトウェアジャパン2019（2019年2月5日）
- https://speakerdeck.com/line_developers/machine-learning-and-interpretability

R packages

mlr: Machine Learning in R

- https://cran.r-project.org/package=mlr

iml: Interpretable Machine Learning

- https://cran.r-project.org/package=iml

DALEX: Descriptive mAchine Learning EXplanations

- https://cran.r-project.org/package=DALEX

breakDown: Model Agnostic Explainers for Individual Predictions

- https://cran.r-project.org/package=breakDown

その他

mlr

- Machine Learning in R
  - https://mlr-org.com/
- Machine Learning in R - Next Generation (mlr3)
  - https://mlr3.mlr-org.com/

ALE plot

- Visualizing the Effects of Predictor Variables in Black Box Supervised Learning Models
  - Daniel Apley
  - the Joint Statistical Meetings 2017
  - https://arxiv.org/abs/1612.08468
  - https://ww2.amstat.org/meetings/jsm/2017/onlineprogram/AbstractDetails.cfm?abstractid=324823

その他

H-statistic in RuleFit

- Predictive learning via rule ensembles.
  - Friedman, Jerome H, and Bogdan E Popescu
  - The Annals of Applied Statistics. JSTOR, 916–54. (2008)
  - https://projecteuclid.org/euclid.aoas/1223908046

LIME

- Marco Tulio Ribeiro, Sameer Singh, Carlos Guestrin (2016) "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
  - https://arxiv.org/abs/1602.04938
- lime: Local Interpretable Model-Agnostic Explanations
  - https://cran.r-project.org/web/packages/lime/vignettes/Understanding_lime.html

その他

Shapley value

- A Unified Approach to Interpreting Model Predictions
  - Scott M. Lundberg and Su-In Lee
  - Advances in Neural Information Processing Systems 30 (NIPS 2017)
  - https://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions
- Explaining prediction models and individual predictions with feature contributions
  - Štrumbelj, Erik and Igor Kononenko
  - Knowledge and information systems 41.3 (2014): 647-665.
  - https://www.semanticscholar.org/paper/Explaining-prediction-models-and-individual-with-%C5%A0trumbelj-Kononenko/eb89cd70cbcfda0c350333b5a938d5da3b7b435f

breakDown

- Explanations of Model Predictions with live and breakDown Packages
  - Mateusz Staniak and Przemysław Biecek
  - The R Journal (2018) 10:2, pages 395-409.
  - https://journal.r-project.org/archive/2018/RJ-2018-072/index.html

その他

the alternate features search

- S. Hara, T. Maehara, Finding Alternate Features in Lasso, arXiv:1611.05940, 2016.
  - https://github.com/sato9hara/LassoVariants
  - https://github.com/katokohaku/AlternateLassoR
  -

XGBoostExplainer: An R package that makes xgboost models fully interpretable

- https://github.com/AppliedDataSciencePartners/xgboostExplainer
- https://medium.com/applied-data-science/new-r-package-the-xgboost-explainer-51dd7d1aa211
- http://kato-kohaku-0.hatenablog.com/entry/2018/12/14/002253