DALEX
Descriptive mAchine Learning EXplanations

Alicja Gosiewska
MI2 Data Lab
Warsaw University of Technology

Data and models

```
> library(DALEX)
> data(apartments)
> data(apartmentsTest)
> head(apartments)
  m2.price construction.year surface floor no.rooms    district
1     5897              1953      25     3        1 Srodmiescie
2     1818              1992     143     9        5     Bielany
3     3643              1937      56     1        2       Praga
4     3517              1995      93     7        3      Ochota
5     3013              1992     144     6        5     Mokotow
6     5795              1926      61     6        2 Srodmiescie
```

Data and models

```
> library(DALEX)
> data(apartments)
> data(apartmentsTest)
> head(apartments)
  m2.price construction.year surface floor no.rooms    district
1     5897              1953      25     3        1 Srodmiescie
2     1818              1992     143     9        5     Bielany
3     3643              1937      56     1        2       Praga
4     3517              1995      93     7        3      Ochota
5     3013              1992     144     6        5     Mokotow
6     5795              1926      61     6        2 Srodmiescie
> apartments_lm_model <- lm(m2.price ~ ., data = apartments)
> library(randomForest)
> set.seed(471)
> apartments_rf_model <- randomForest(m2.price ~ ., data = apartments)
```

Data and models

```
> library(DALEX)
> data(apartments)
> data(apartmentsTest)
> head(apartments)
  m2.price construction.year surface floor no.rooms    district
1     5897              1953      25     3        1 Srodmiescie
2     1818              1992     143     9        5     Bielany
3     3643              1937      56     1        2       Praga
4     3517              1995      93     7        3      Ochota
5     3013              1992     144     6        5     Mokotow
6     5795              1926      61     6        2 Srodmiescie
```

```
> apartments_lm_model <- lm(m2.price ~ ., data = apartments)
> library(randomForest)
> set.seed(471)
> apartments_rf_model <- randomForest(m2.price ~ ., data = apartments)
```

```
> predicted_mi2_lm <- predict(apartments_lm_model, apartmentsTest)
> sqrt(mean((predicted_mi2_lm - apartmentsTest$m2.price)^2))
[1] 283.0865
```

```
> predicted_mi2_rf <- predict(apartments_rf_model, apartmentsTest)
> sqrt(mean((predicted_mi2_rf - apartmentsTest$m2.price)^2))
[1] 283.1138
```

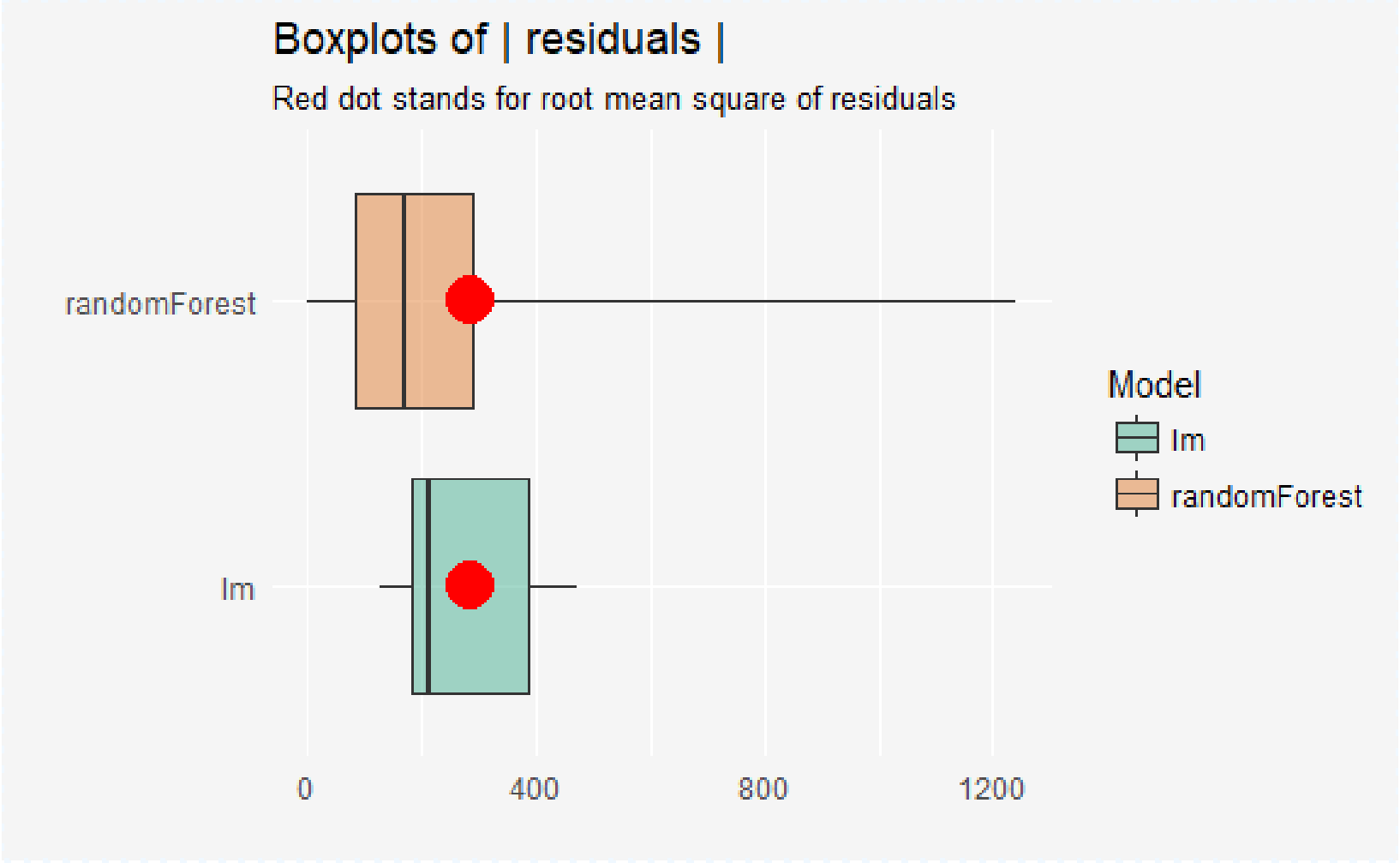
The explain() function

```
explain(model, data, y, predict_function, link, ..., label)
```

```
> explainer_lm <- explain(apartments_lm_model,
+                         data = apartmentsTest[,2:6], y = apartmentsTest$m2.price)
>
> explainer_rf <- explain(apartments_rf_model,
+                         data = apartmentsTest[,2:6], y = apartmentsTest$m2.price)
```

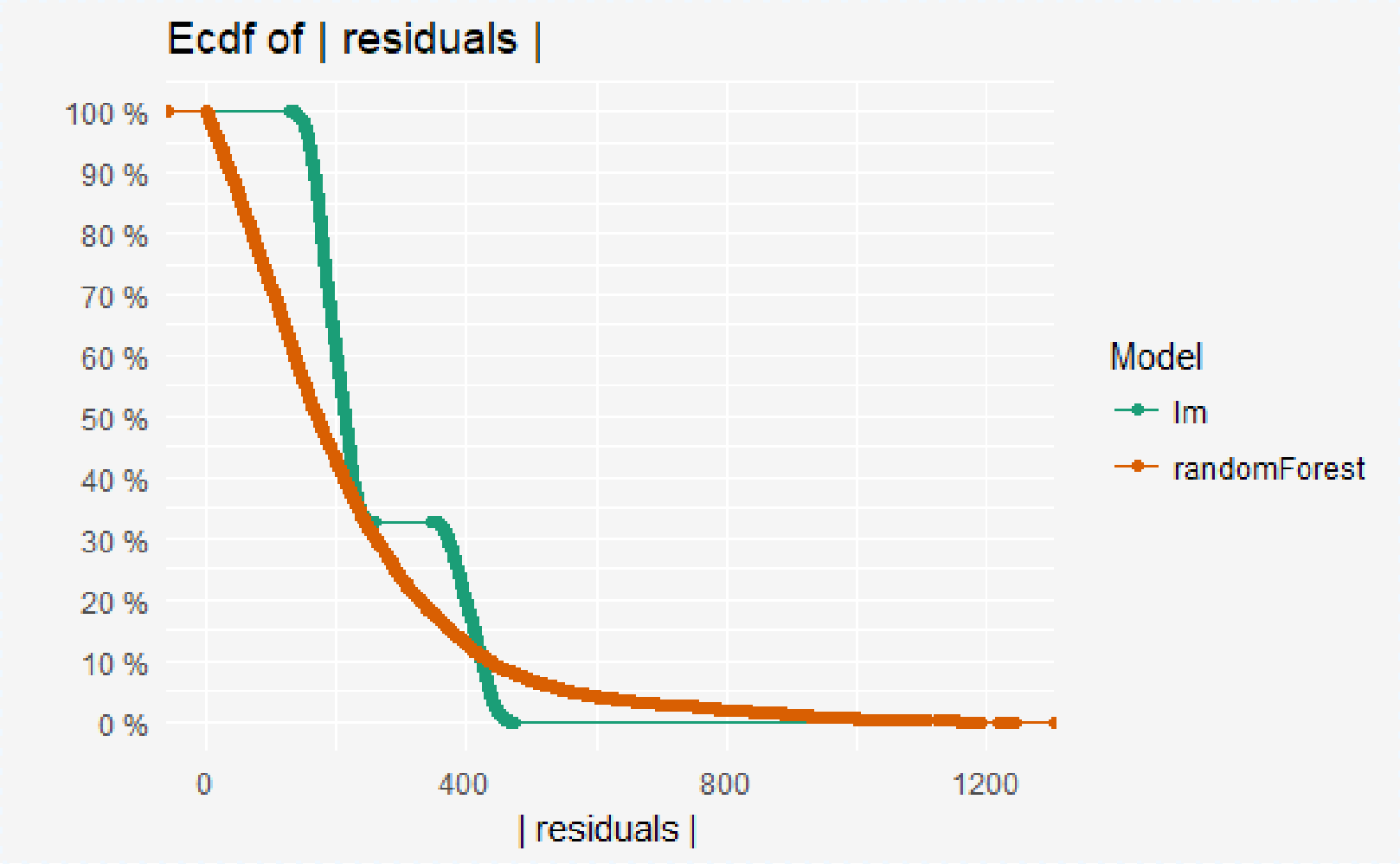
Model performance

```
> mp_lm <- model_performance(explainer_lm)
> mp_rf <- model_performance(explainer_rf)
> plot(mp_lm, mp_rf, geom = "boxplot")
```

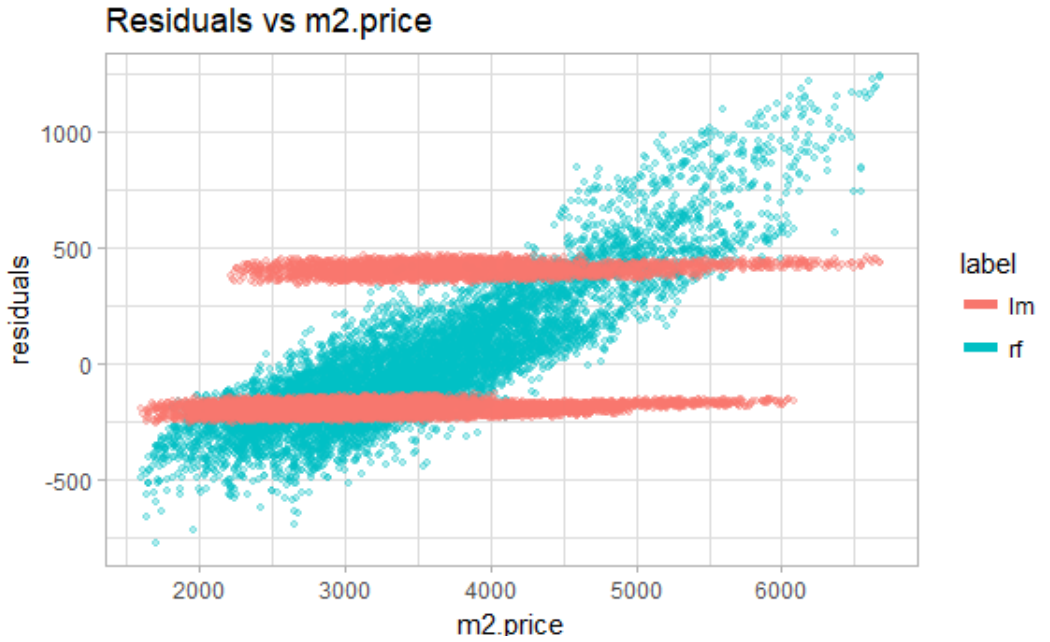


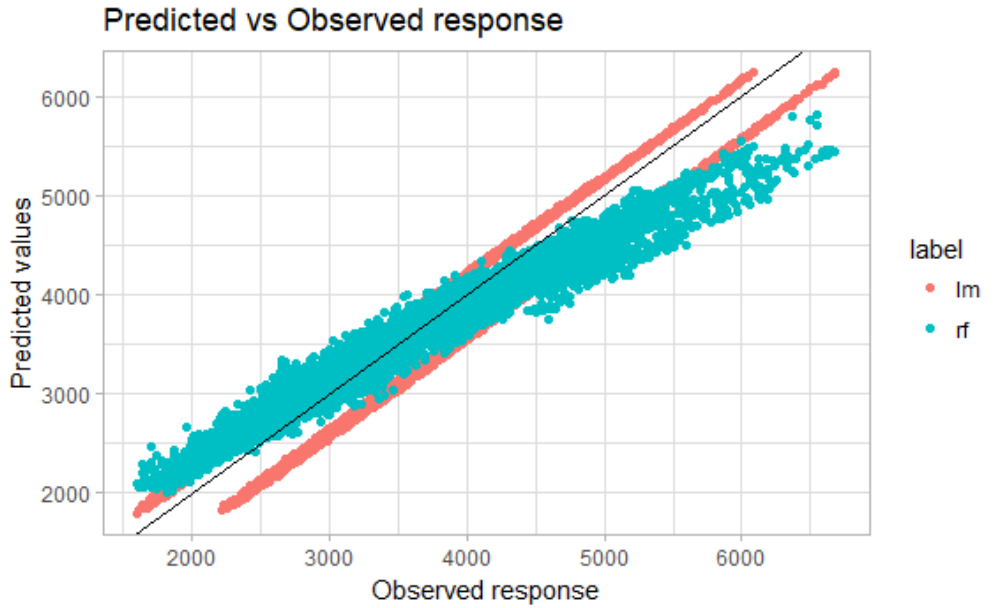
Model performance

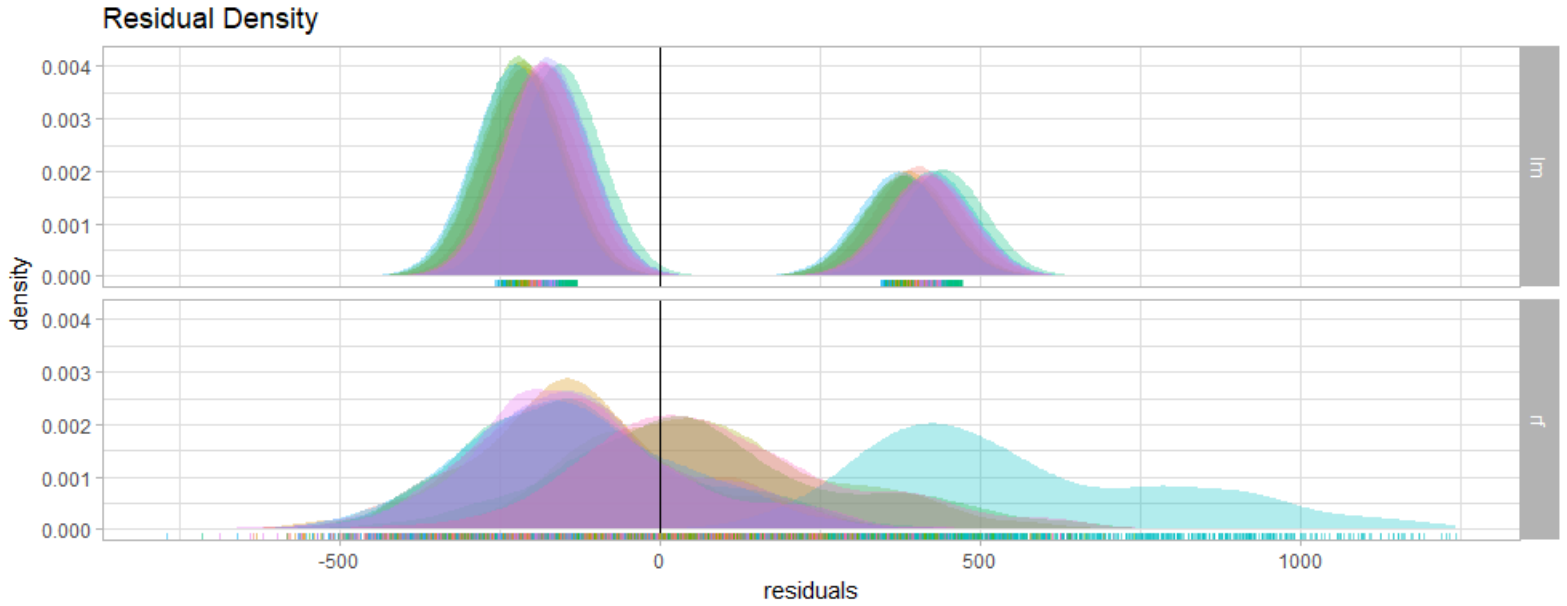
```
> mp_lm <- model_performance(explainer_lm)
> mp_rf <- model_performance(explainer_rf)
>
> plot(mp_lm, mp_rf)
```

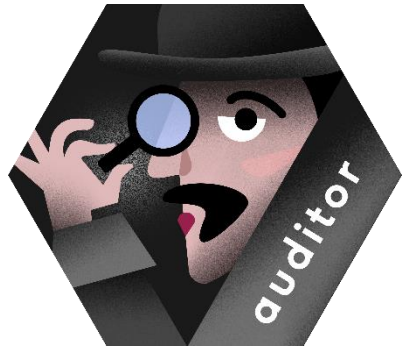


auditor: model performance



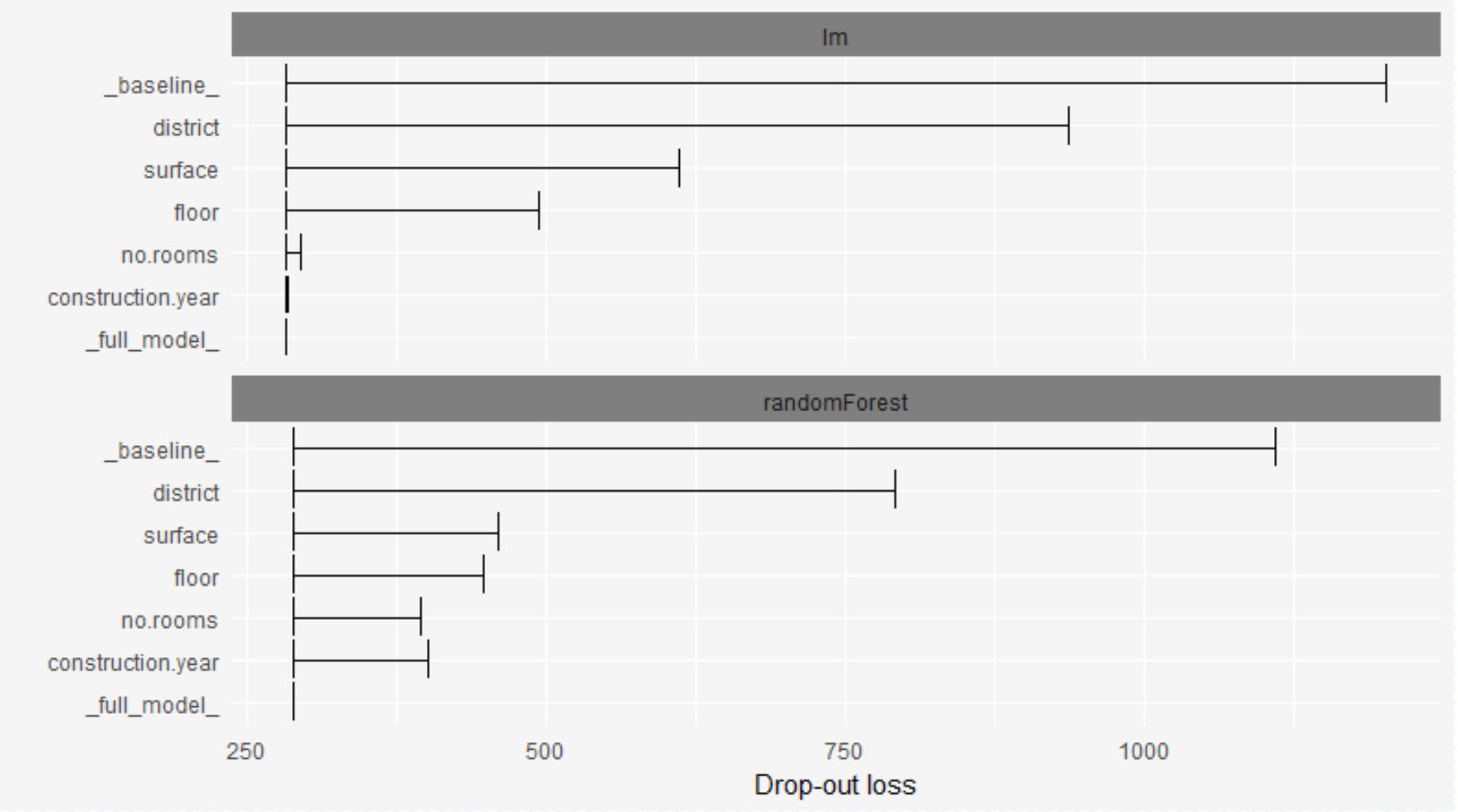






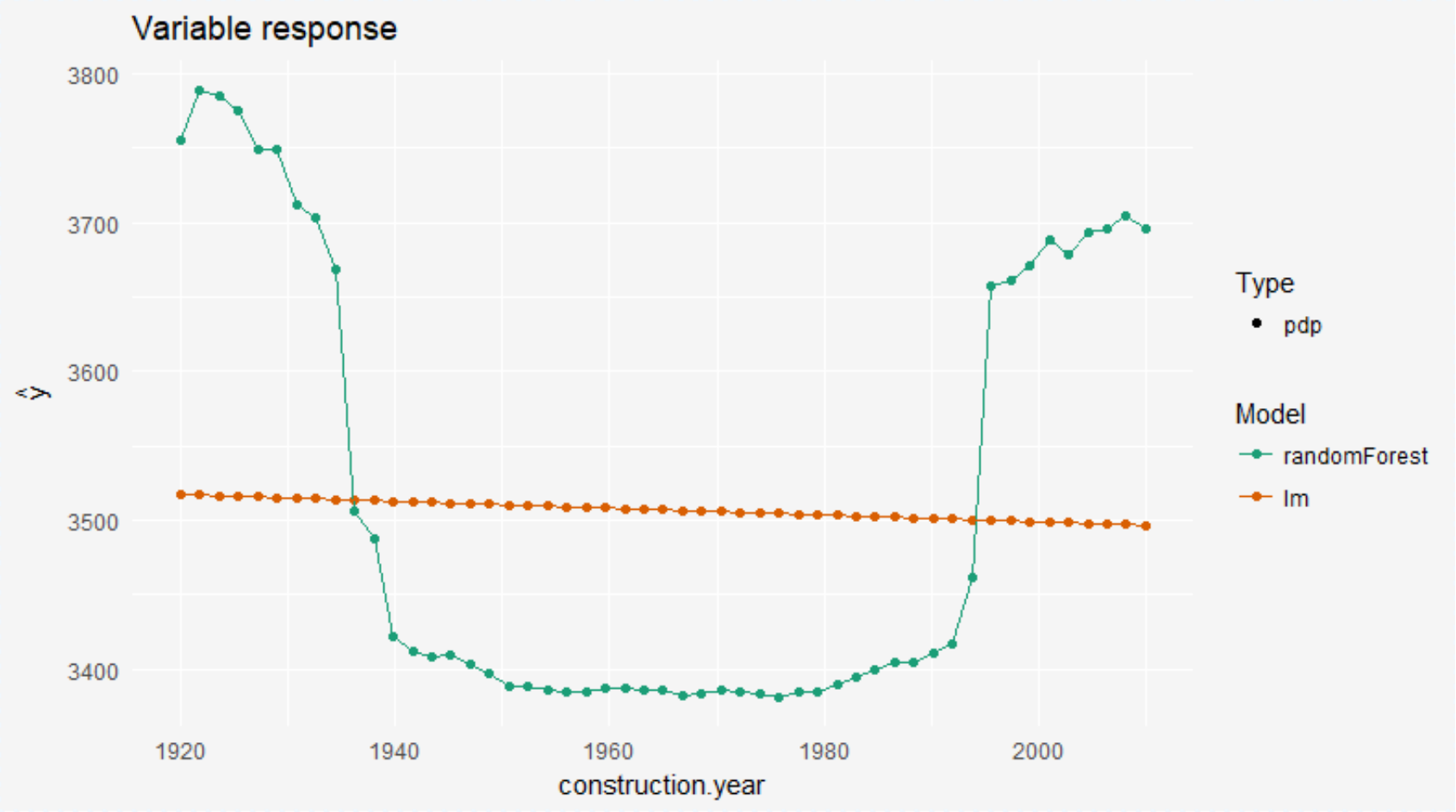
Variable importance

```
> vi_rf <- variable_importance(explainer_rf, loss_function = loss_root_mean_square)
> vi_lm <- variable_importance(explainer_lm, loss_function = loss_root_mean_square)
> plot(vi_lm, vi_rf)
```



Variable response

```
> sv_rf  <- single_variable(explainer_rf, variable =  "construction.year", type = "pdp")
> sv_lm  <- single_variable(explainer_lm, variable =  "construction.year", type = "pdp")
> plot(sv_rf, sv_lm)
```



Improving the model

- Linear model and random forest had equal performance for apartments dataset.
- In general the random forest model has smaller residuals than the linear model but there is a small fraction of very large residuals.
- Random forest model under-predicts expensive apartments. It is not a model that we would like to employ.

Improving the model

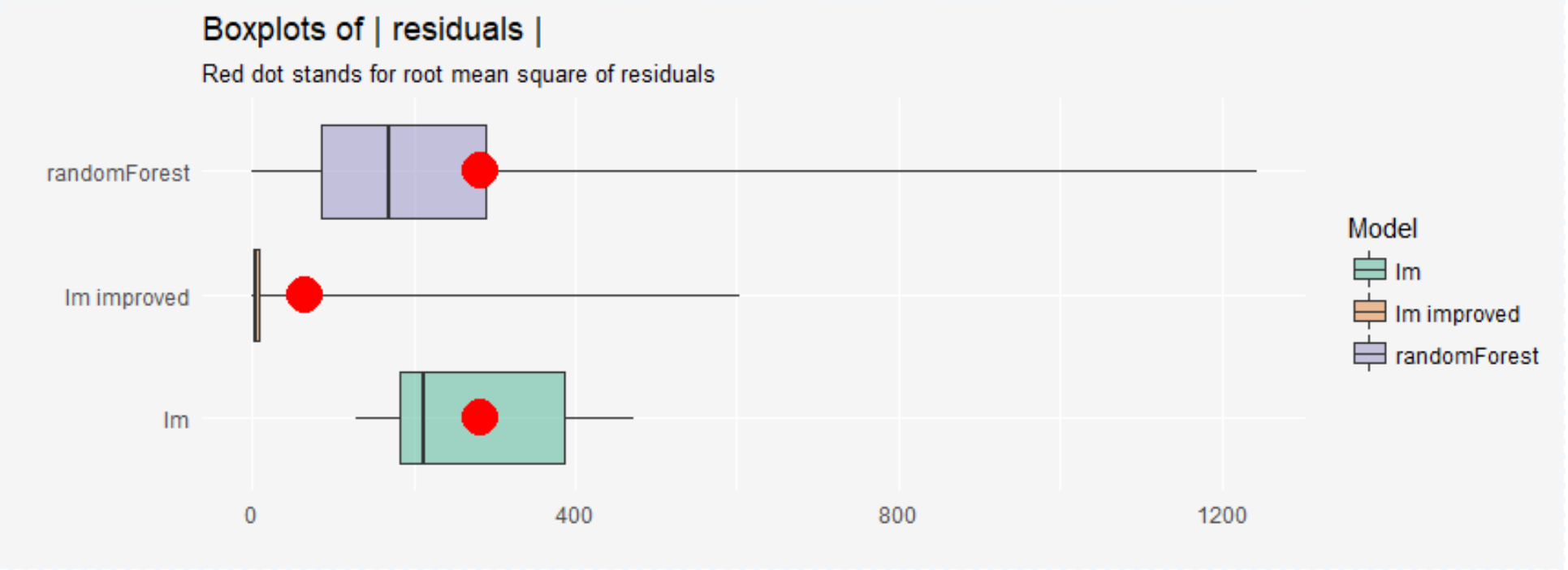
- Linear model and random forest had equal performance for apartments dataset.
- In general the random forest model has smaller residuals than the linear model but there is a small fraction of very large residuals.
- Random forest model under-predicts expensive apartments. It is not a model that we would like to employ.
- `construction_year` is important for the random forest model.
- the relation between `construction_year` and the price of square meter is non linear.

Improving the model

```
> apartments_lm_model_improved <- lm(m2.price ~ I(construction.year < 1935 | construction.year > 1995) +
  surface + floor + no.rooms + district, data = apartments)
```

Improving the model

```
> apartments_lm_model_improved <- lm(m2.price ~ I(construction.year < 1935 | construction.year > 1995) +
  surface + floor + no.rooms + district, data = apartments)
>
> explainer_lm_improved <- explain(apartments_lm_model_improved,
+                            data = apartmentsTest[,2:6], y = apartmentsTest$m2.price,
+                            label = "lm improved")
>
> mp_lm_improved <- model_performance(explainer_lm_improved)
> plot(mp_lm_improved, mp_lm, mp_rf, geom = "boxplot")
```



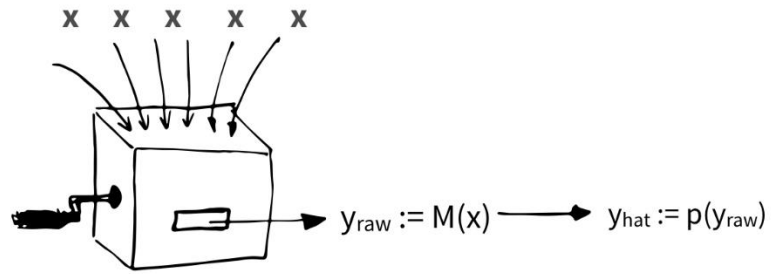
DALEX - Descriptive mAchine Learning EXplanations

DALEX explains black-box models. It's a methodology for better diagnostic of any black-box model.

This approach increases understanding of a model, increases trust in model predictions and allows to further improve the model. It also allows to compare two or more models in the scale space

Notation:

- **(x, y)** - pair of input and output data points. x may be anything (data.frame, factors, numbers, text, image), while here we assume that y is numerical or can be transformed to the numerical variable ($x \in X; y \in R$).
- **M** - a black box model, $M: X \to R$. Its output will be denoted as $y_{raw} = M(x)$
- **p** - a link function, transforms raw model output to the same space as y. Useful for classification, while for regression its usually the identity. $p:R \to R$. Its output will be denoted as $y_{hat} = p(y_{raw})$



explain(model; data; y; predict_function; trans)

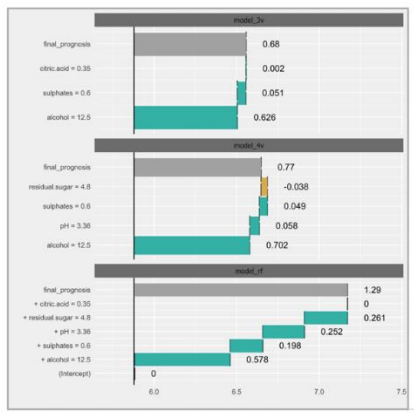
The *explain*() function creates a wrapper over a black-box model. This wrapper contains all necessary components for further processing.

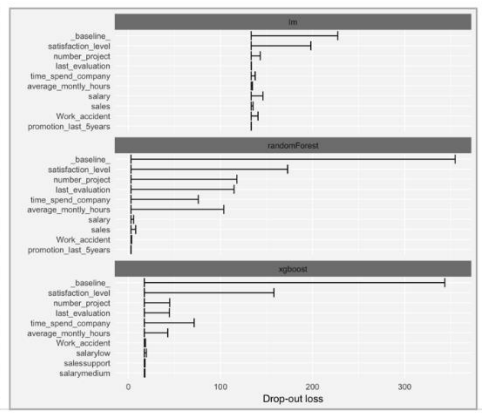
prediction_breakdown(explainer, x) | variable_importance(explainer) | variable_response(explainer, variable)

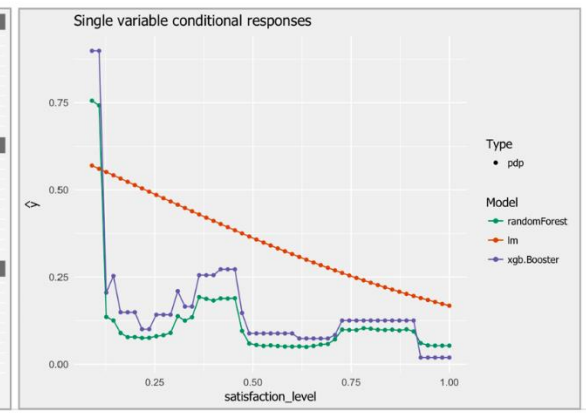
Prediction explainers shows features that drive model response for a selected observation

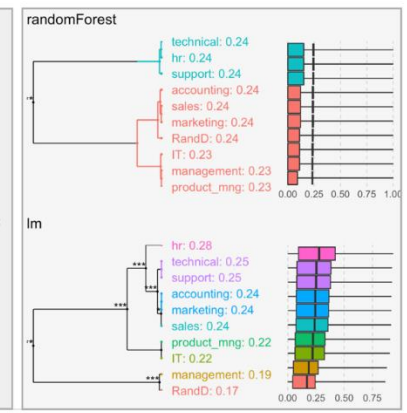
Variable importance explainers shows the drop in the model loss after permutations of a selected variable.

Single variable explainers show conditional relation between model output and a single variable.









Acknowledgements

We acknowledge the financial support from the NCN Opus grant 2016/21/B/ST6/02176