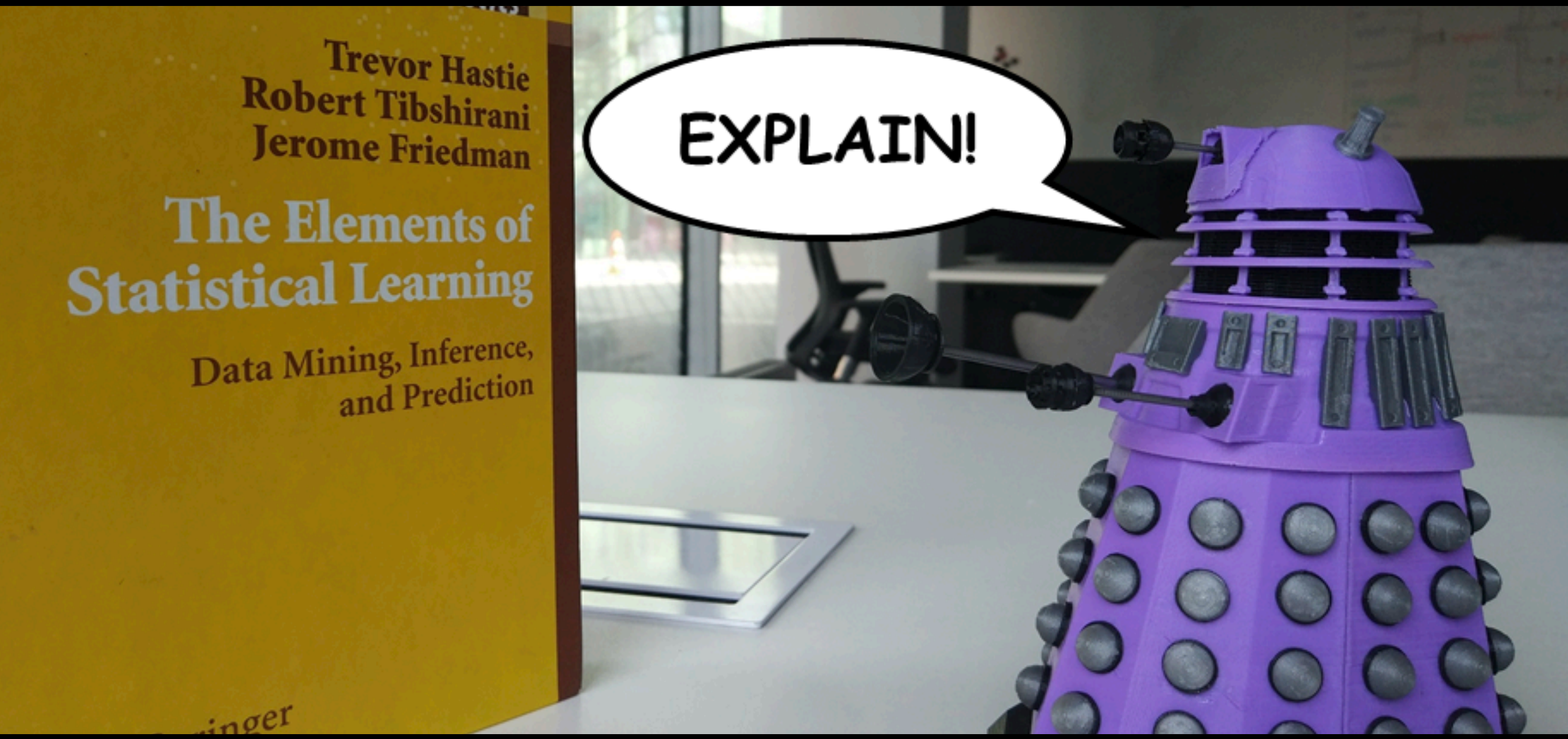
DALEX will help you understand complex predictive models



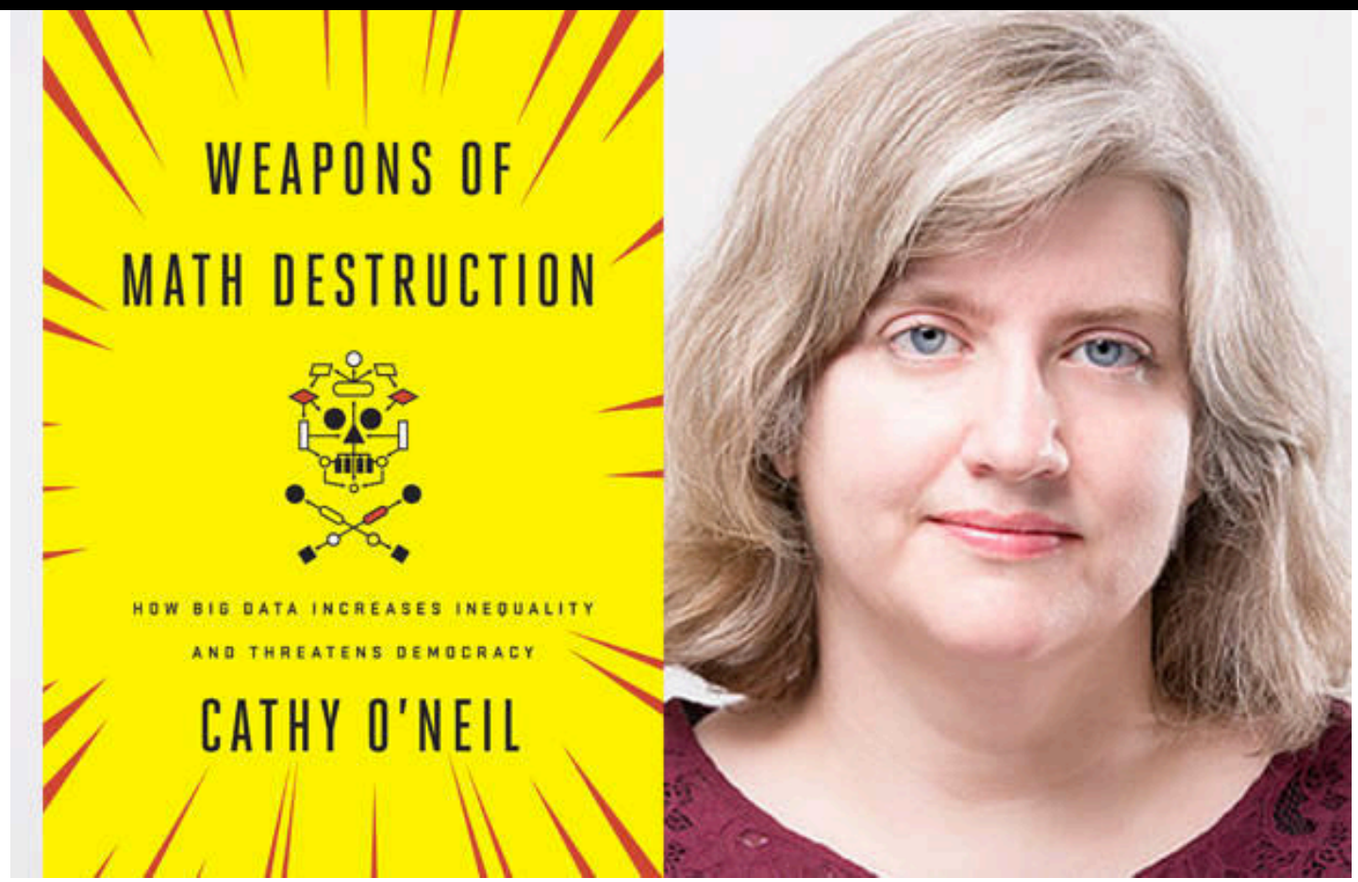
Przemysław Biecek

useR 2018

Why do we need explanations for complex models?

Cathy O'Neil:

The era of blind faith in ~~big data~~ black boxes must end



- "You don't see a lot of skepticism," she says. "The algorithms are like shiny new toys that we can't resist using. We trust them so much that we project meaning on to them."
- Ultimately algorithms, according to O'Neil, reinforce discrimination and widen inequality, "using people's fear and trust of mathematics to prevent them from asking questions".

https://www.theguardian.com/books/2016/oct/27/cathy-oneil-weapons-of-math-destruction-algorithms-big-data

Why do we need explanations for complex models?

Article Talk Read Edit View history Search Wikipedia

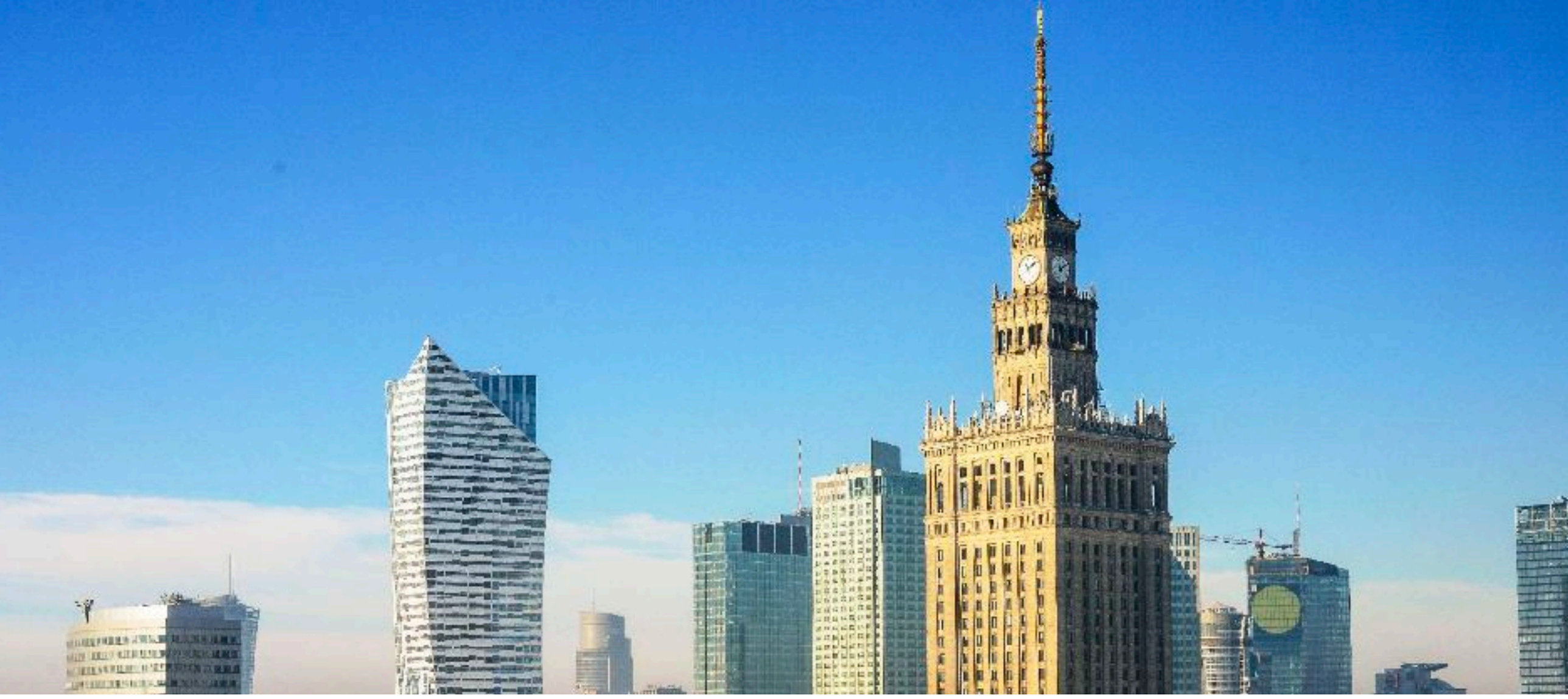
Right to explanation

From Wikipedia, the free encyclopedia

In the regulation of algorithms, particularly artificial intelligence and its subfield of machine learning, a **right to explanation** (or **right to explanation**) is a right to be given an explanation for an output of the algorithm. Such rights primarily refer to individual rights to be given explanation for decisions that significantly affect an individual, particularly legally or financially. For example, a person who applies for is denied may ask for an explanation, which could be "Credit bureau X reports that you declared bankruptcy last year; this is the main considering you too likely to default, and thus we will not give you the loan you applied for."

Some such legal rights already exist, while the scope of a general "right to explanation" is a matter of ongoing debate.

Contents [hide]

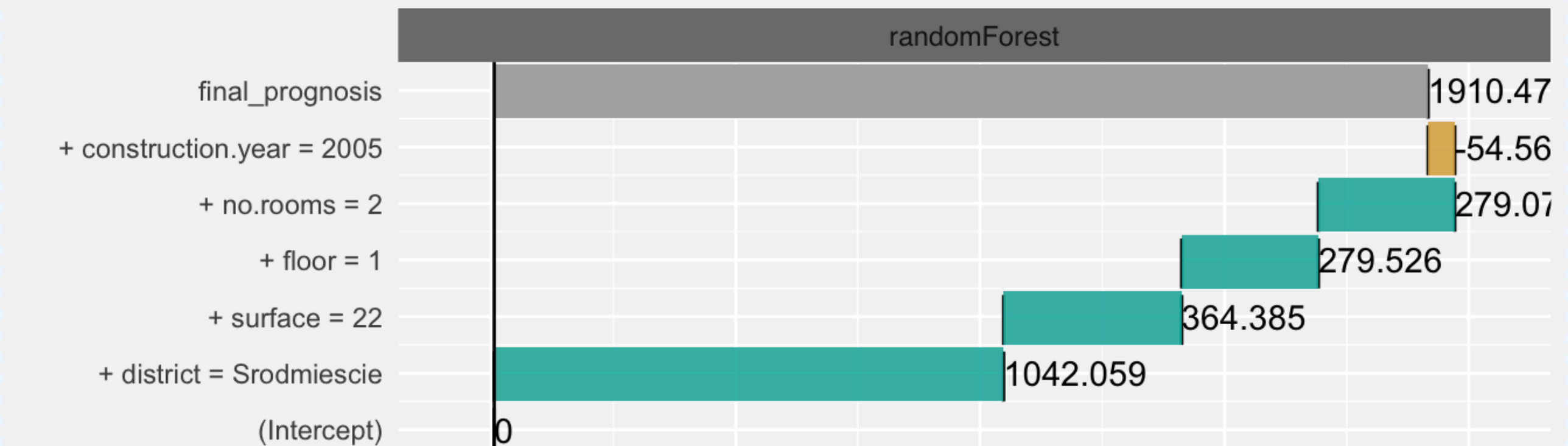


```
library("DALEX")
head(apartments)
```

m2.price	construction.year	surface	floor	no.rooms	district
5897	1953	25	3	1	Srodmiescie
1818	1992	143	9	5	Bielany
3643	1937	56	1	2	Praga
3517	1995	93	7	3	Ochota
3013	1992	144	6	5	Mokotow

breakDown for predictions

```
new_apartment_rf <- single_prediction(explainer_rf, observation = new_apartment)
breakDown:::print.broken(new_apartment_rf)

##                                contribution
## (Intercept)                           0.000
## + district = Srodmiescie           1042.059
## + surface = 22                      364.385
## + floor = 1                         279.526
## + no.rooms = 2                      279.070
## + construction.year = 2005          -54.566
## final_prognosis                    1910.474
```

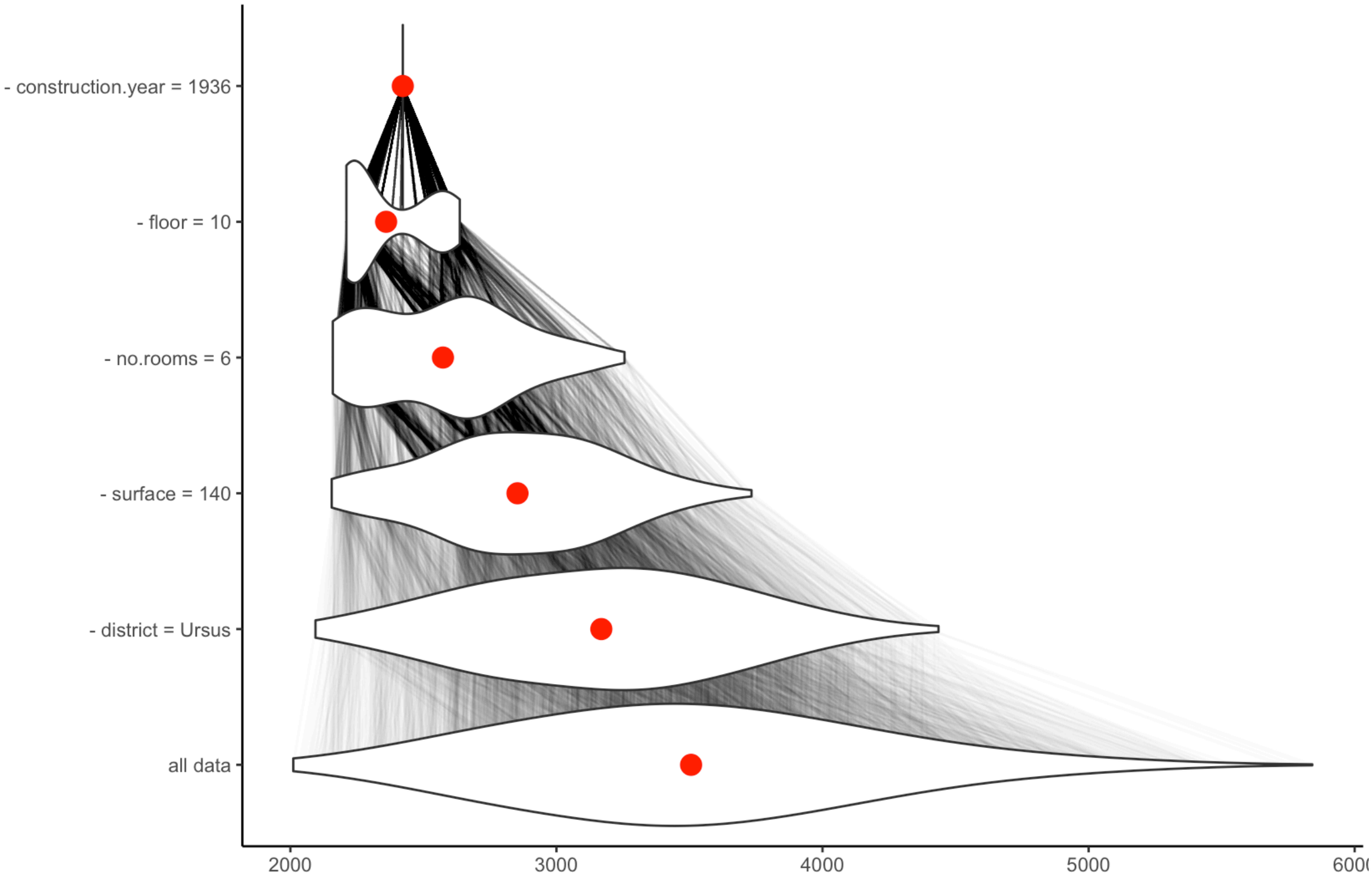


How variables affect this prediction?

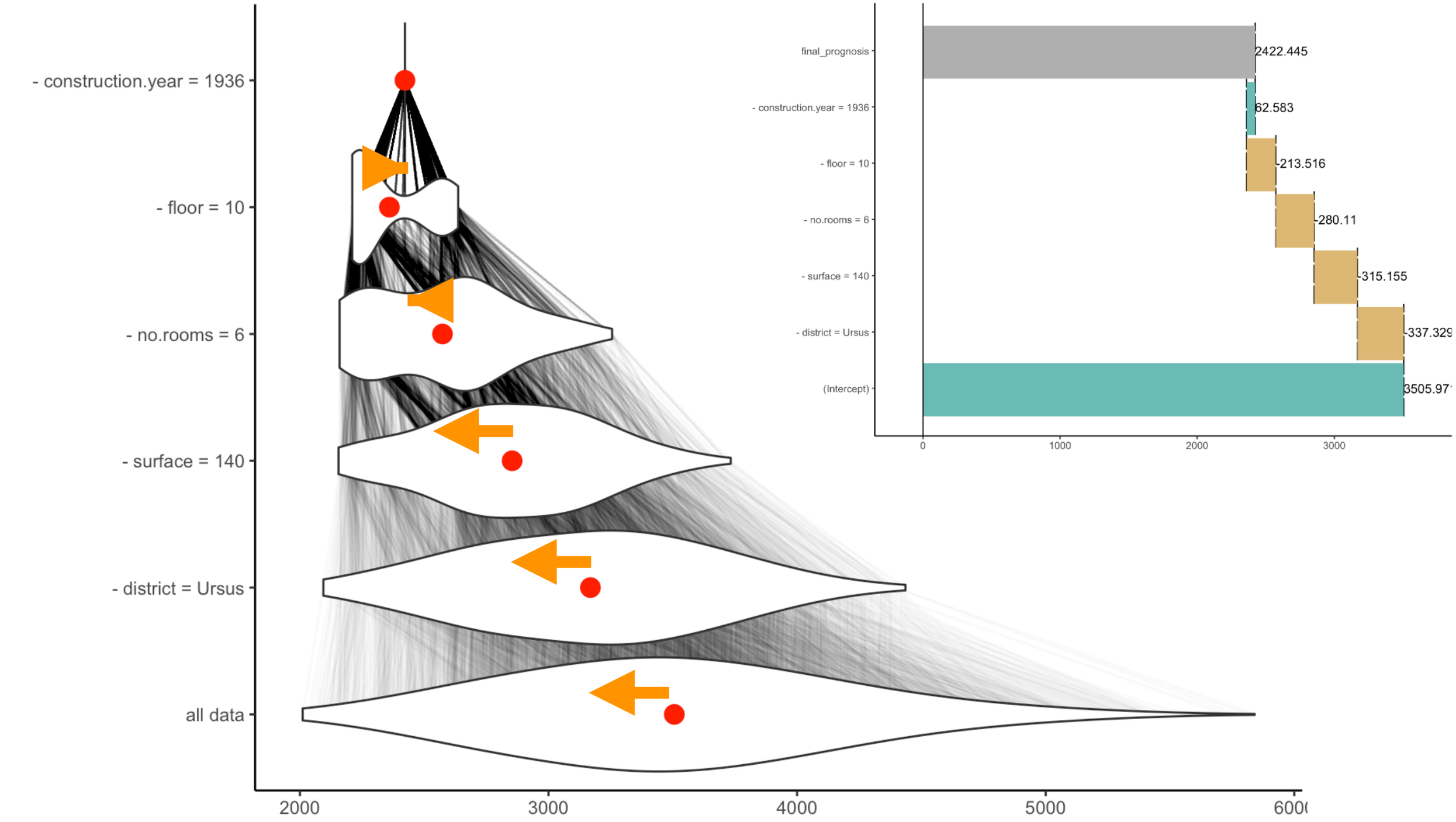
Case:
large (140 m2) flat, 6 rooms,
10 th floor, built in 1936

Prediction from random forest:
3505 EUR/m2

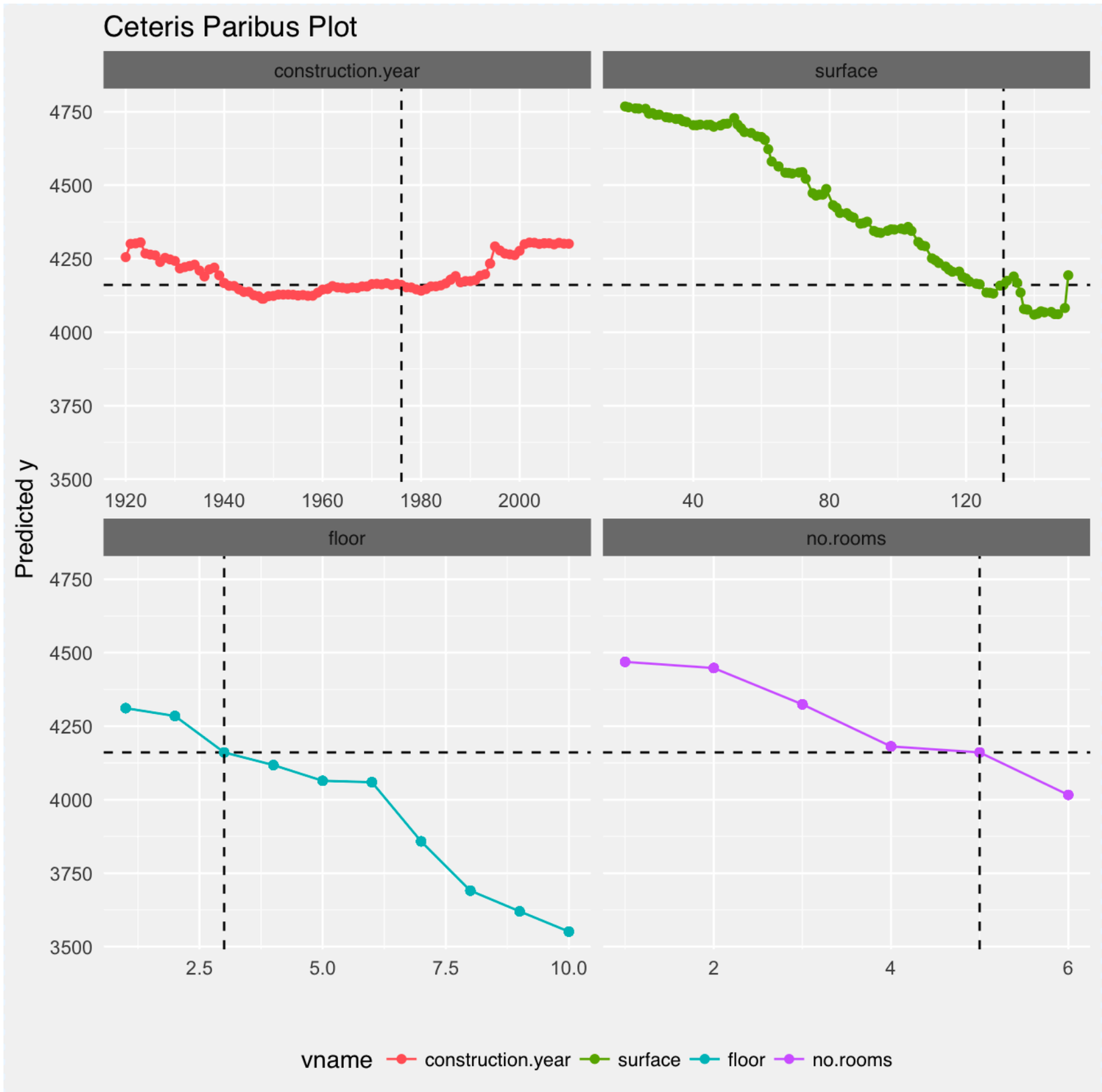
How variables affect this prediction?



How variables affect this prediction?

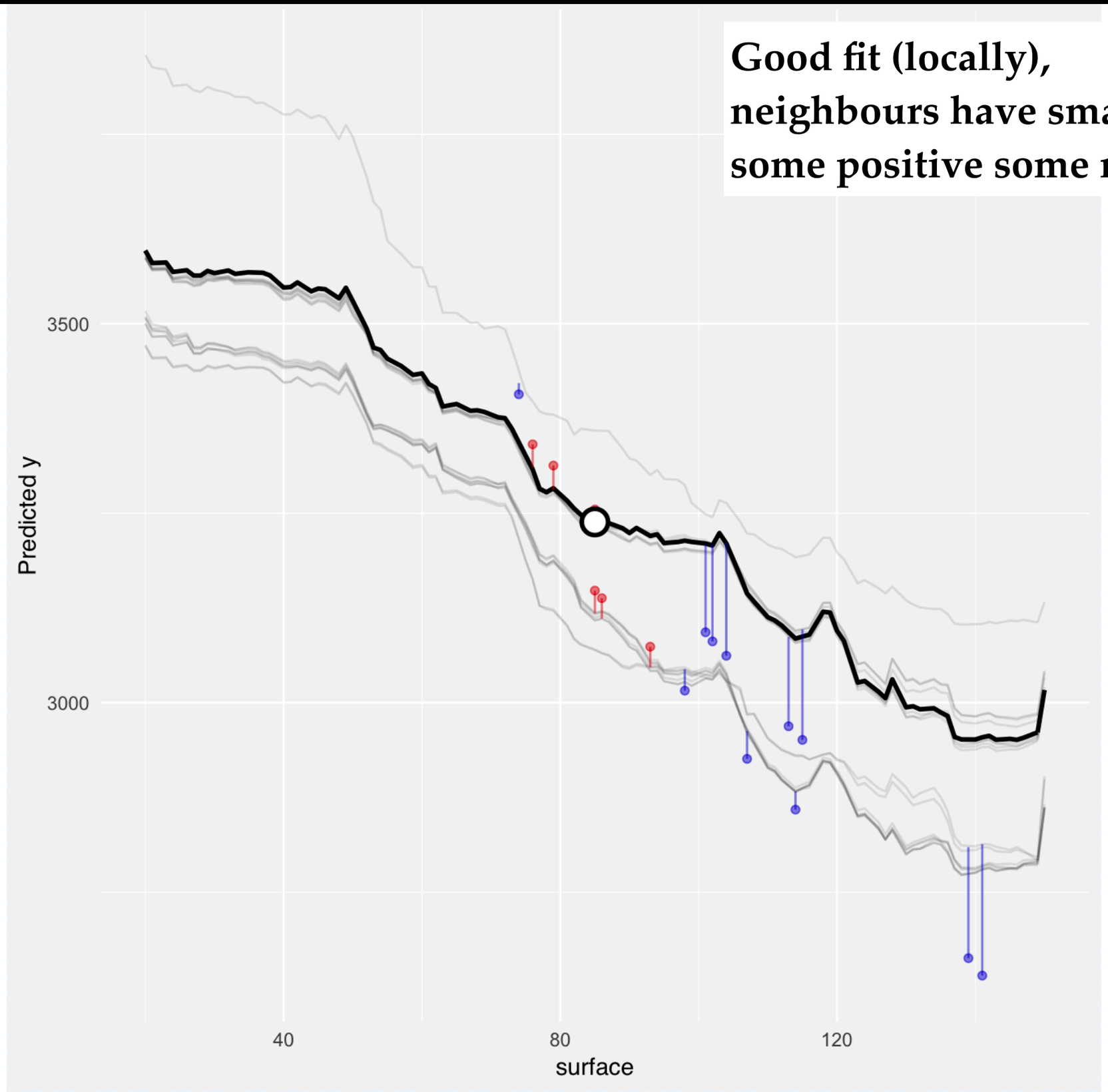


Ceteris Paribus is a Latin phrase for "all else unchanged"



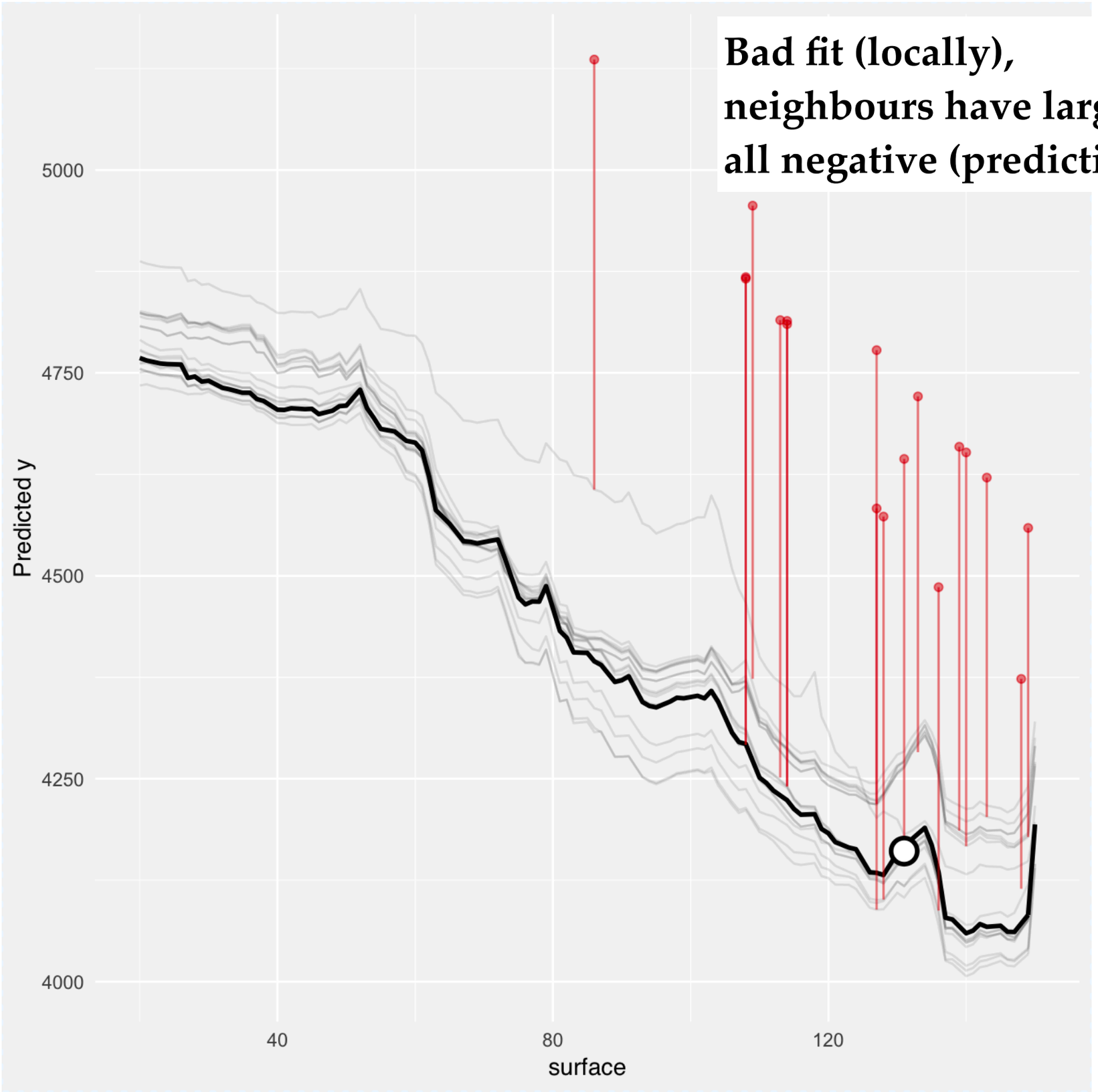
```r
# we need a DALEX object
explainer_rf <- explain(apartments_rf_model,
                        data = apartmentsTest[,2:6],
                        y    = apartmentsTest$m2.price)

# explanations for this data point
new_apartment <- apartmentsTest[1, ]

# as usual, create an explainer and plot it
library("ceterisParibus")
wi_rf <- ceteris_paribus(explainer_rf,
                         observation = new_apartment)

plot(wi_rf,
     split     = "variables",
     color     = "variables",
     quantiles = FALSE)
```

How large are residuals around?



Good fit (locally),
neighbours have small residuals
some positive some negative

How large are residuals around?



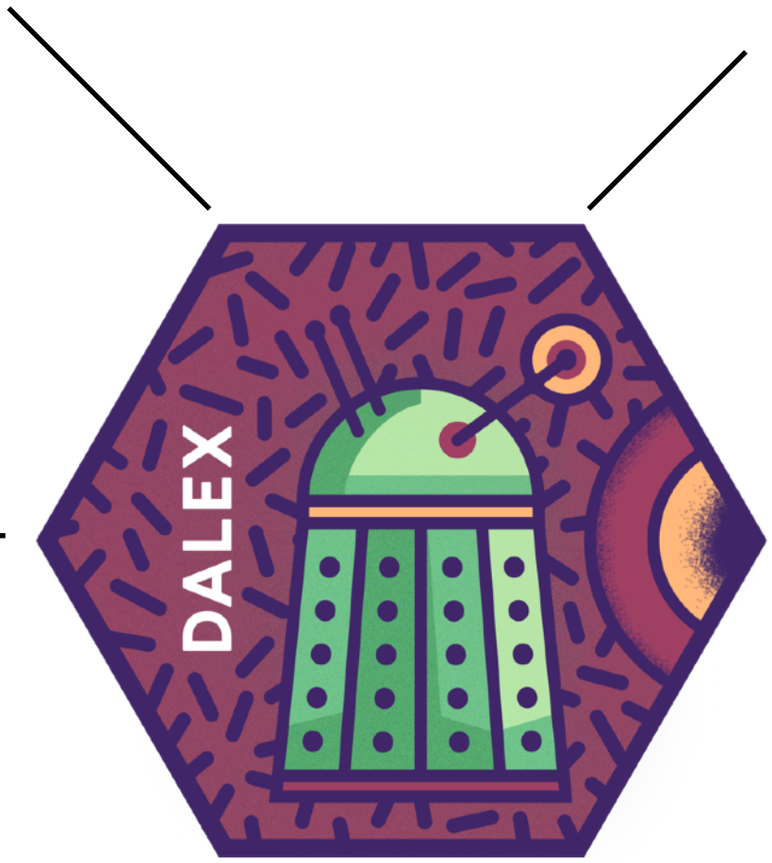
Bad fit (locally),
neighbours have large residuals
all negative (predictions below truth)

Model understanding

Prediction understanding

How good is the model?

How good is a specific model prediction?



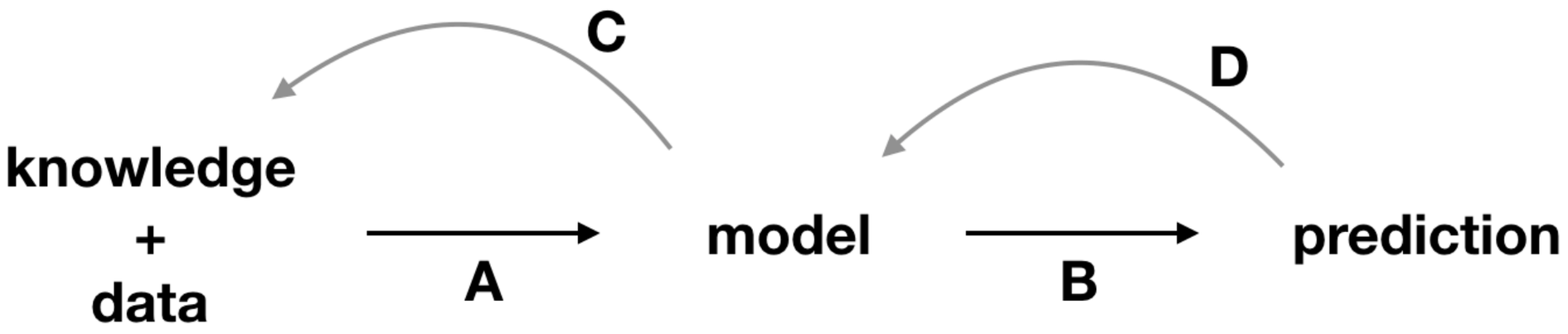
Which variables are important?

Which observations have large residuals?

How a variable is linked with model response?

Which variables influence a specific model prediction?

Typical workflow in ML



A. Modelling is a process in which domain knowledge and data are turned into models.
B. Models are used to generate predictions.
C. Understanding of model structure may increase our knowledge and in consequence leads to a better model. *DALEX helps here.*
D. Understanding of drivers behind particular model predictions may help to correct wrong decisions and in consequence leads to a better model. *DALEX helps here.*

A Tale of Two Models

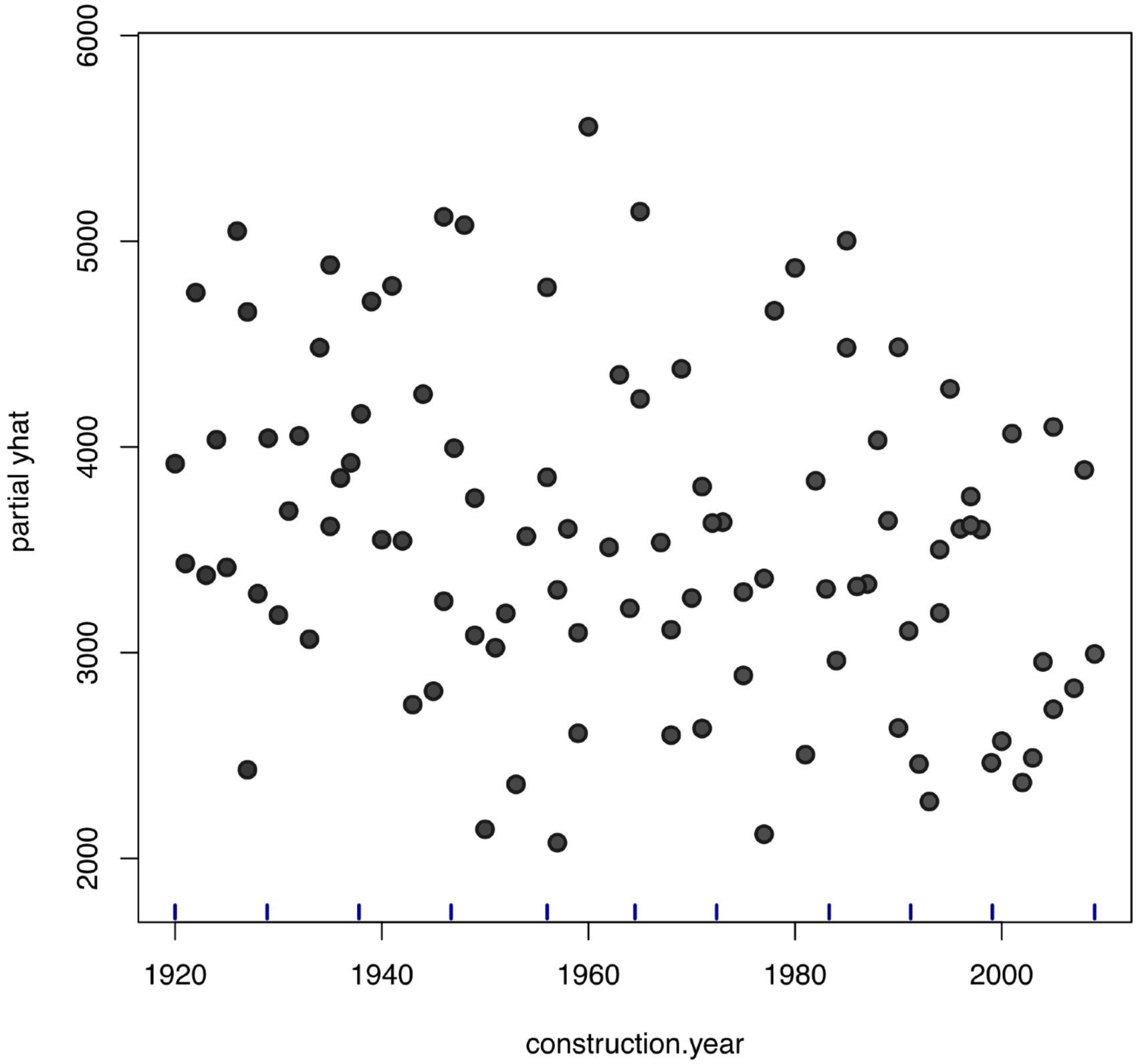
```
apartments_lm_model <- lm(m2.price ~ construction.year + surface + floo
                          no.rooms + district, data = apartments)
library("randomForest")
set.seed(59)
apartments_rf_model <- randomForest(m2.price ~ construction.year + surf
                            no.rooms + district, data = apartments)
```

A Tale of Two Models

Accuracy as a single number is not enough!

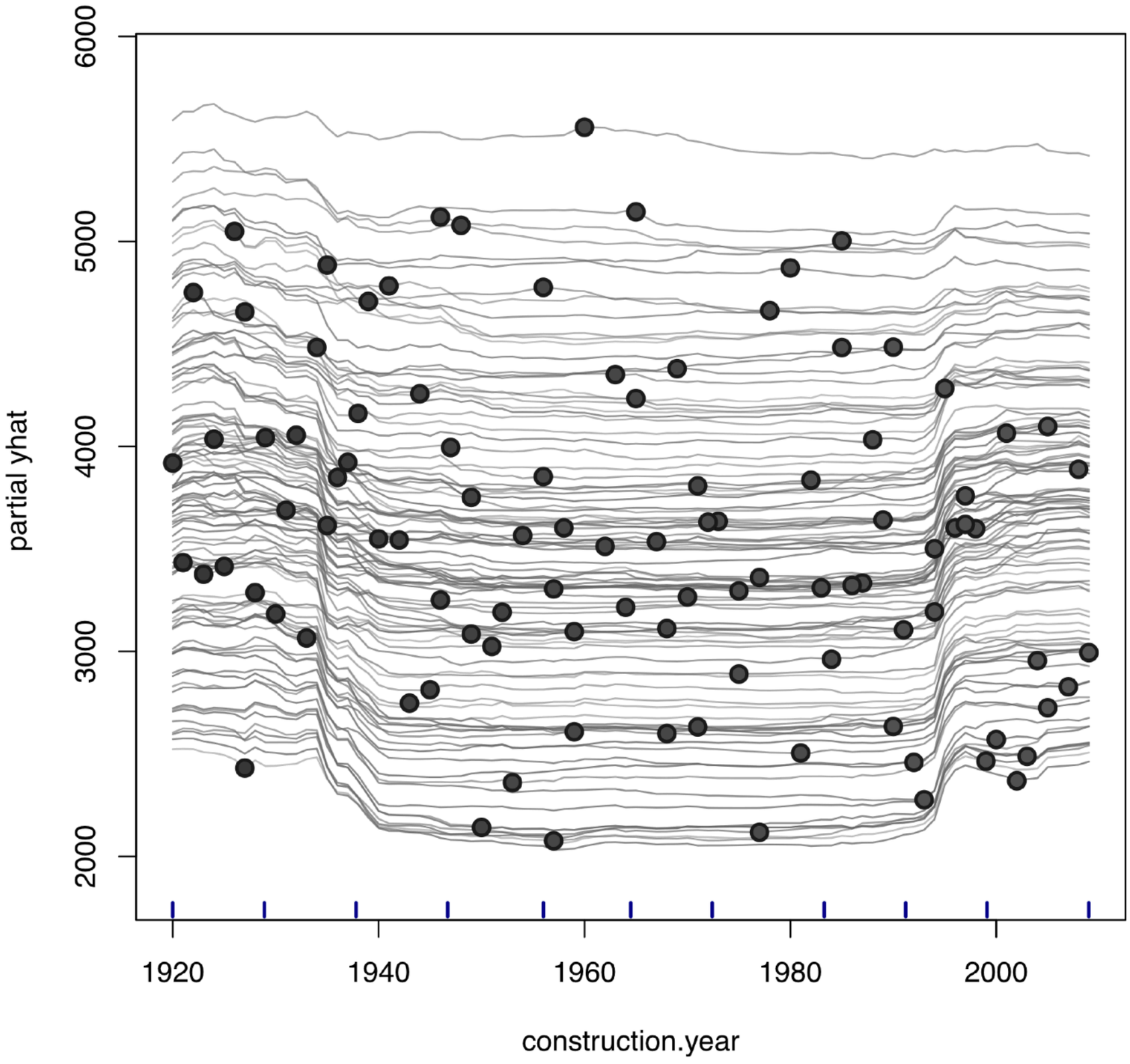
```
# root mean square
predicted_mi2_lm <- predict(apartments_lm_model, apartmentsTest)
sqrt(mean((predicted_mi2_lm - apartmentsTest$m2.price)^2))
## [1] 283.0865

# root mean square
predicted_mi2_rf <- predict(apartments_rf_model, apartmentsTest)
sqrt(mean((predicted_mi2_rf - apartmentsTest$m2.price)^2))
## [1] 283.3479
```

How a single variable affects the response?



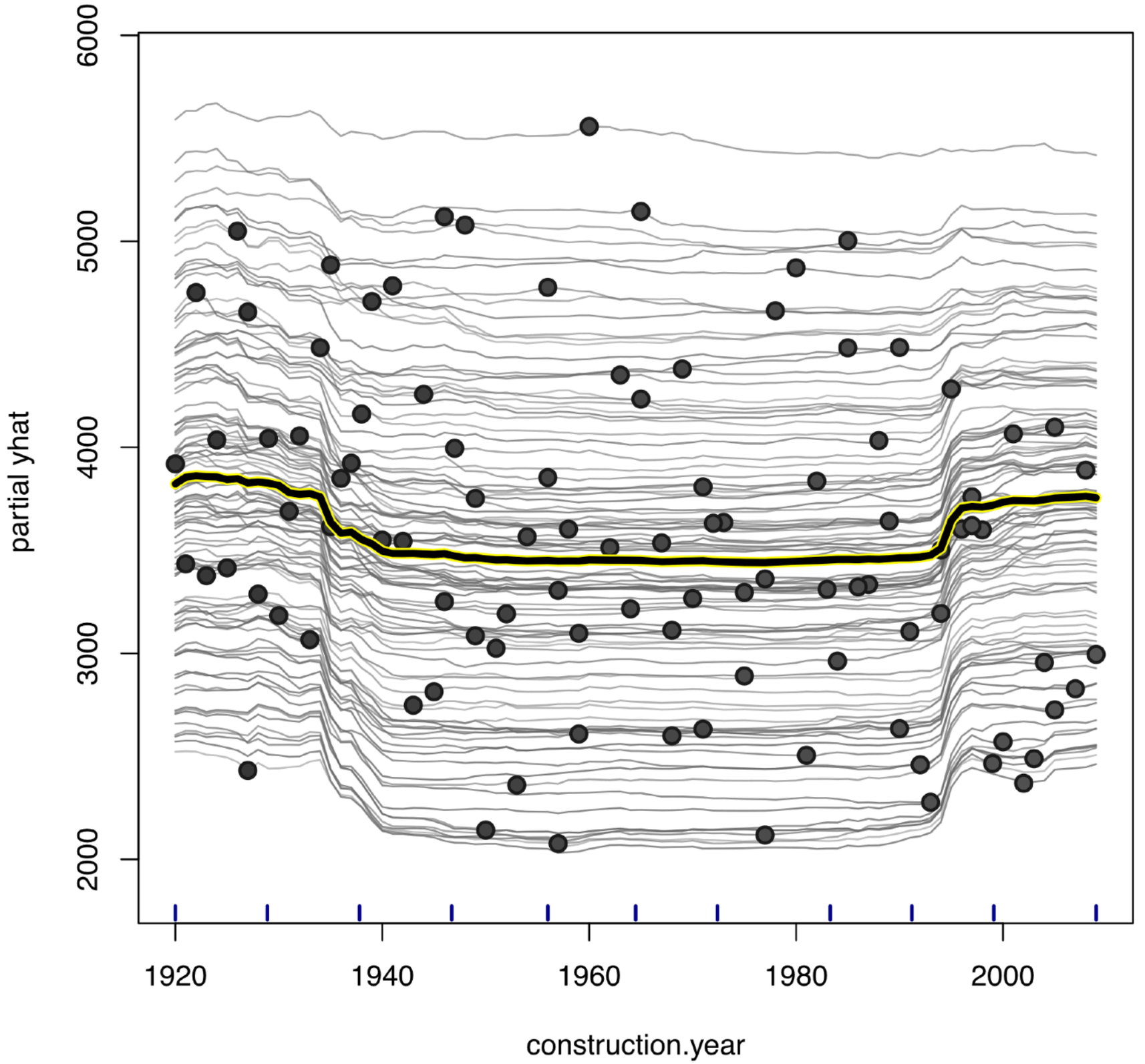
Predictions
vs
Single variable

How a single variable affects the response?



Individual
Conditional
Expectations

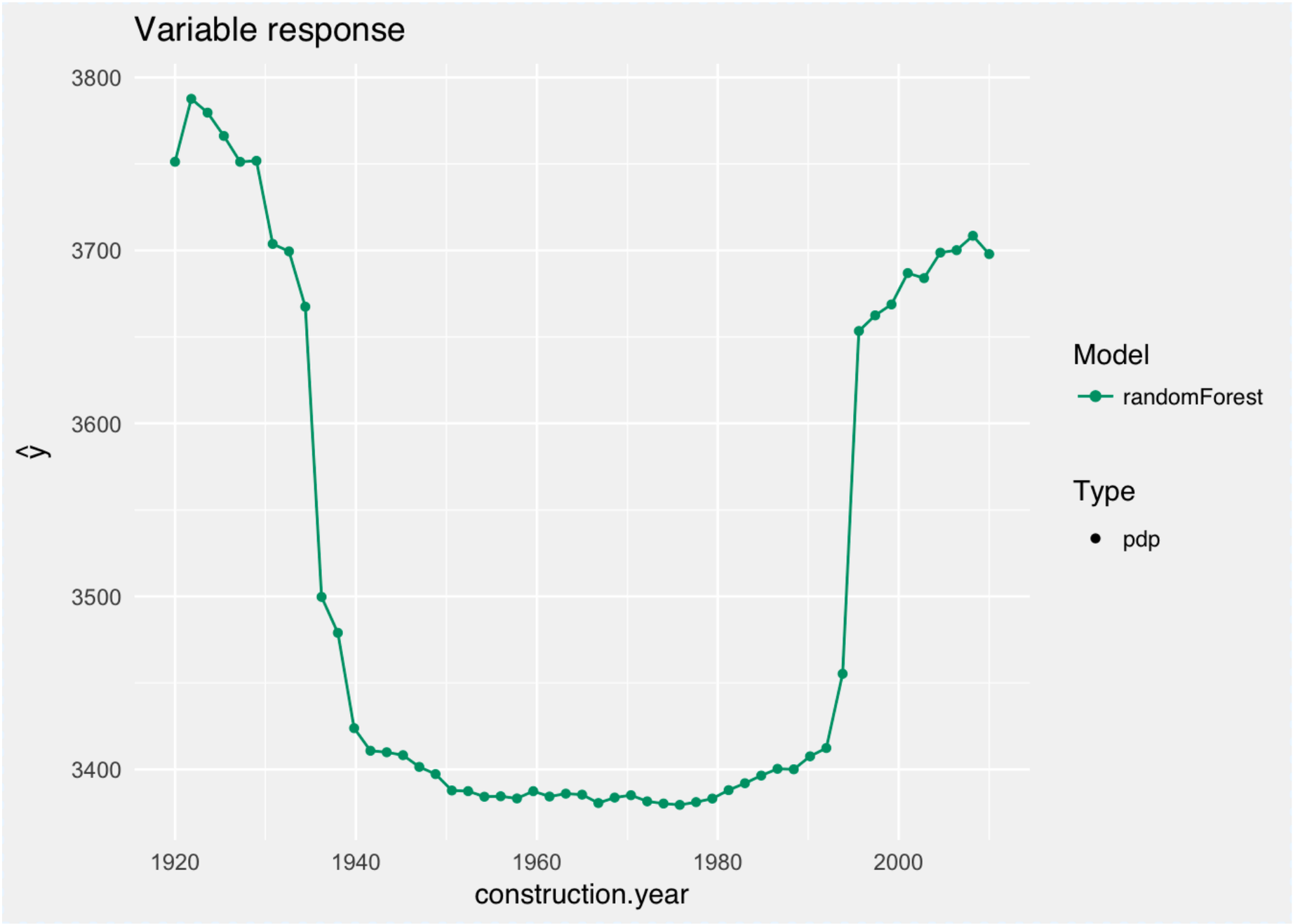
How a single variable affects the response?



Partial Dependence Plot

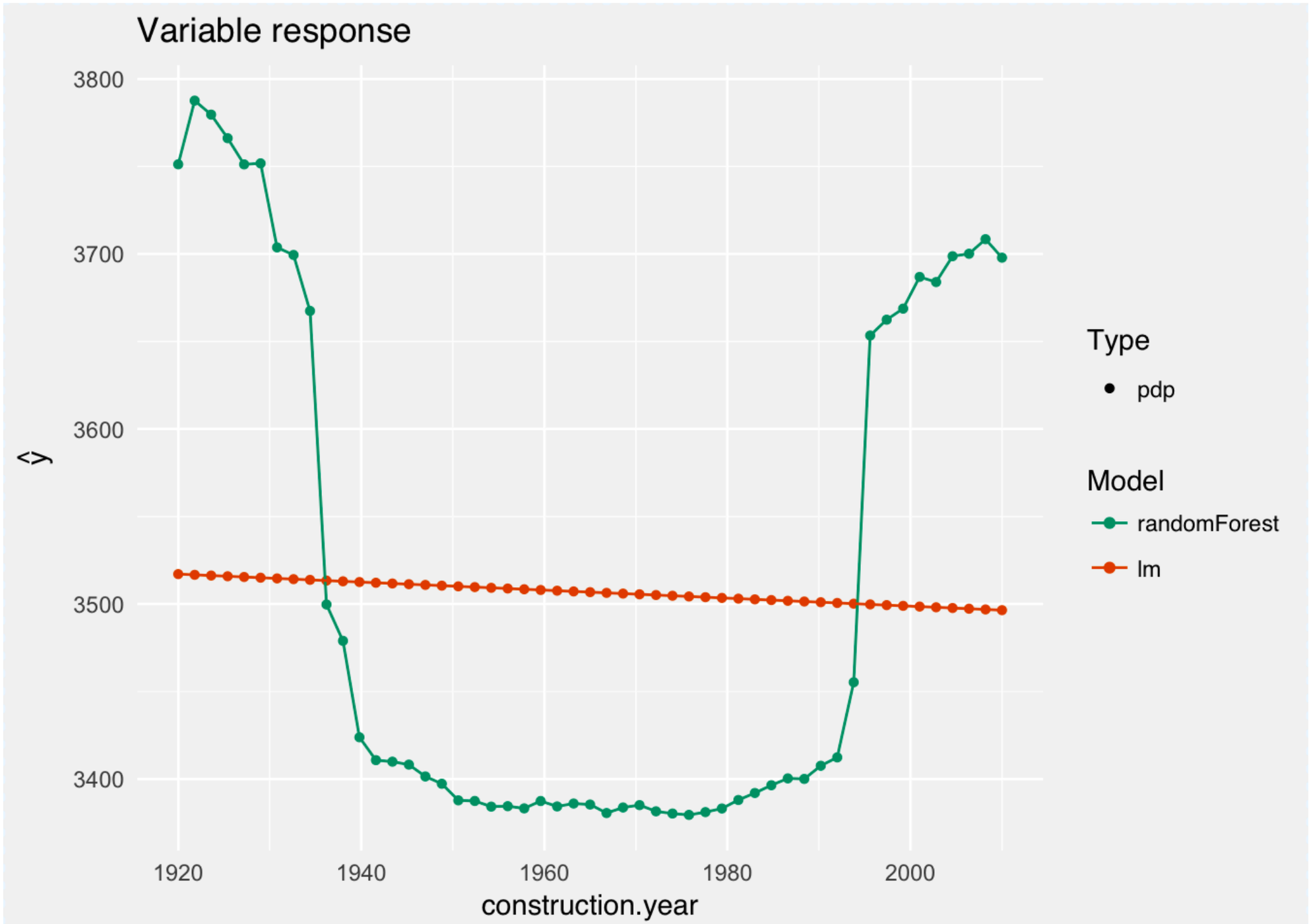
Model response for a continuous variable

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

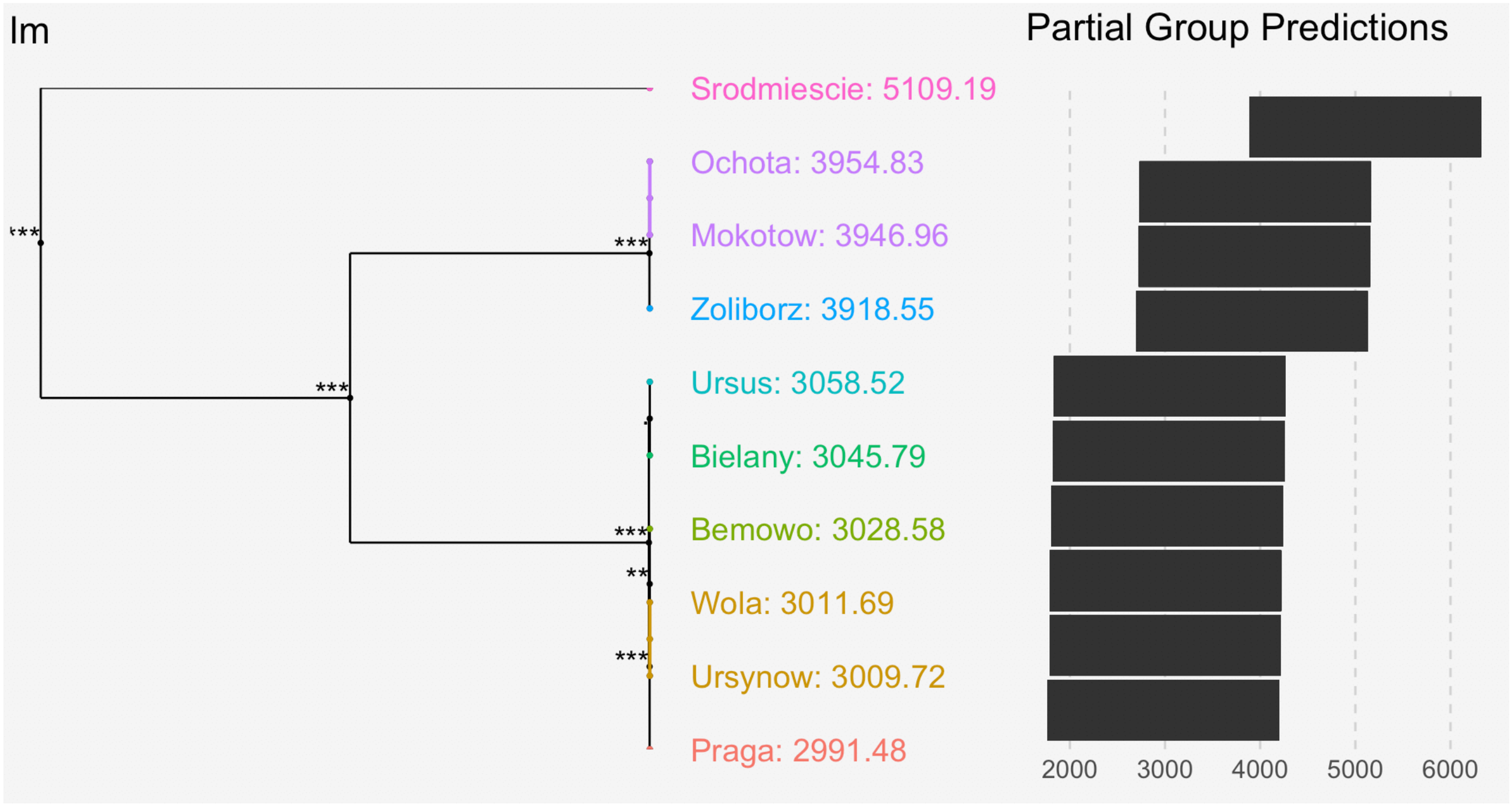


Model response for a continuous variable

```
plot(sv_rf, sv_lm)
```

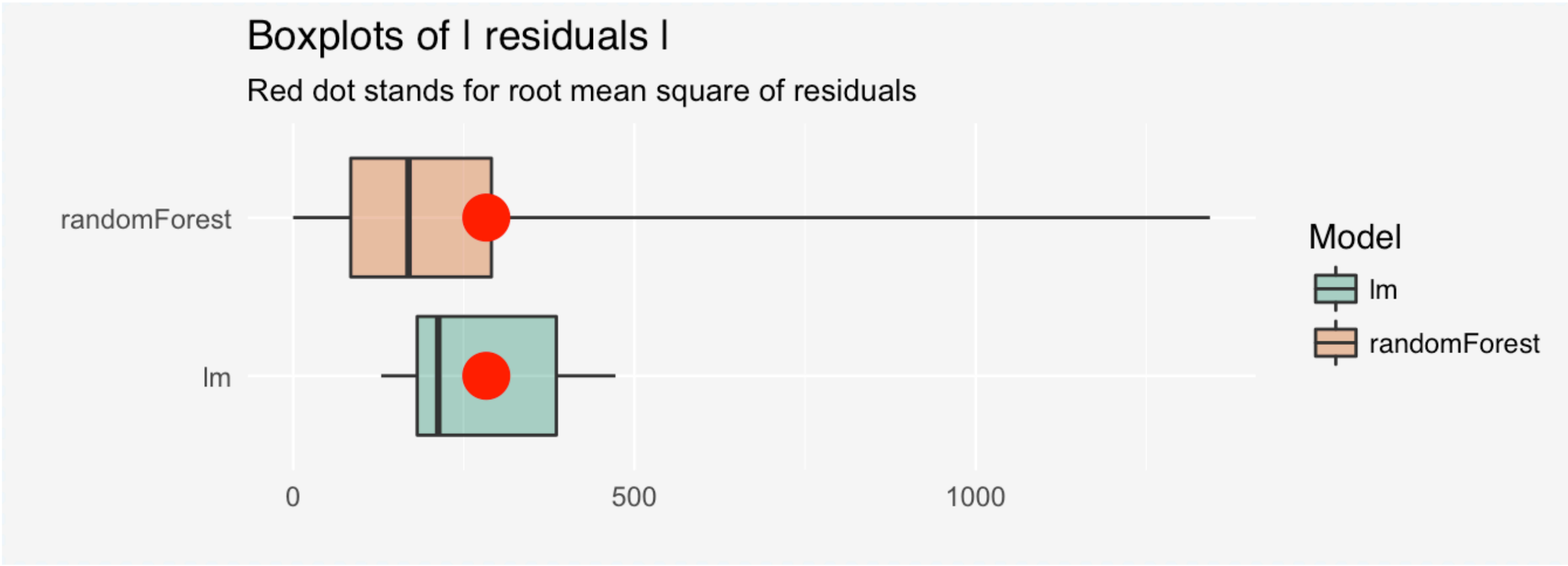


Model response for a categorical variable

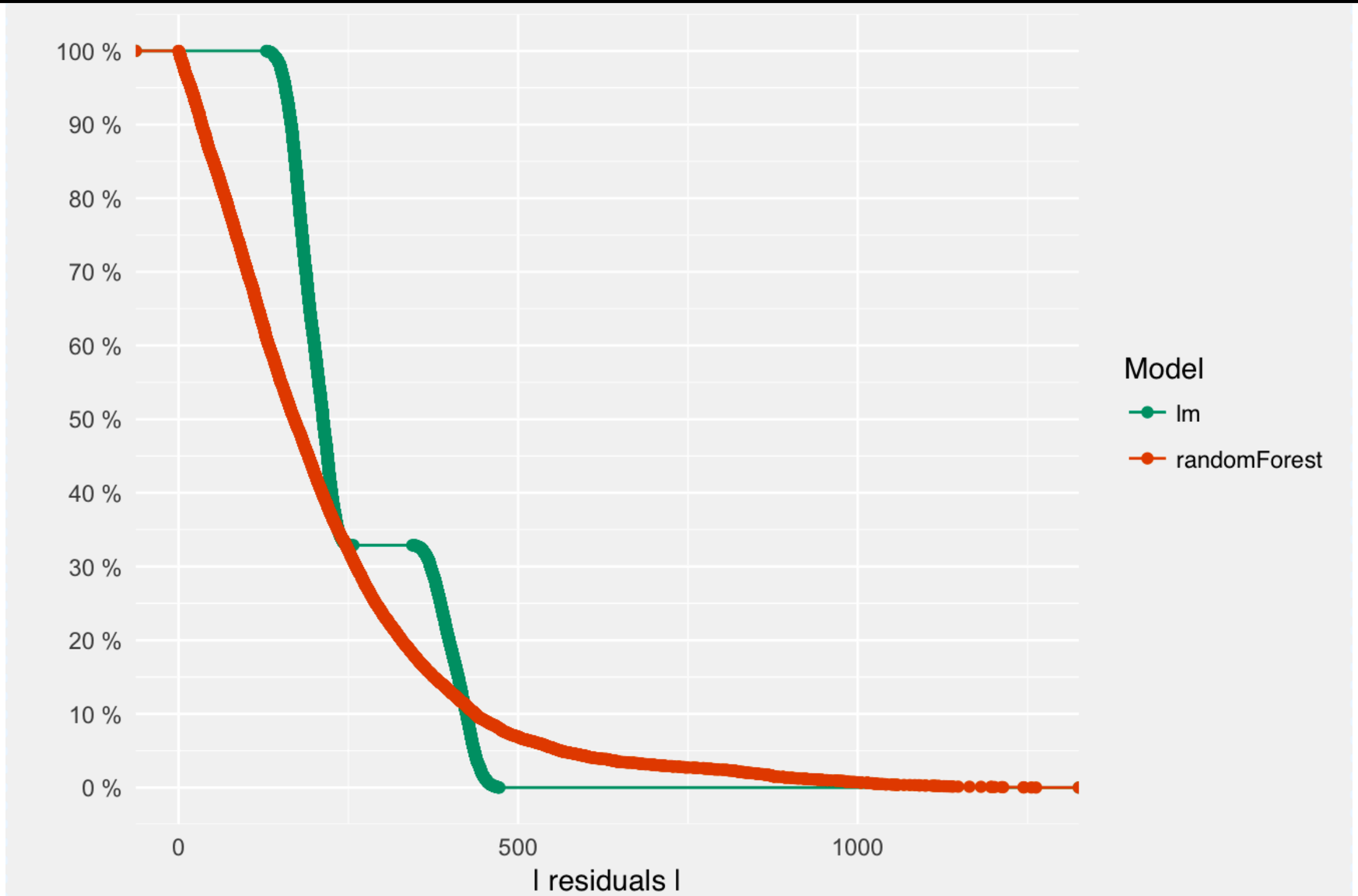


How good are these models?

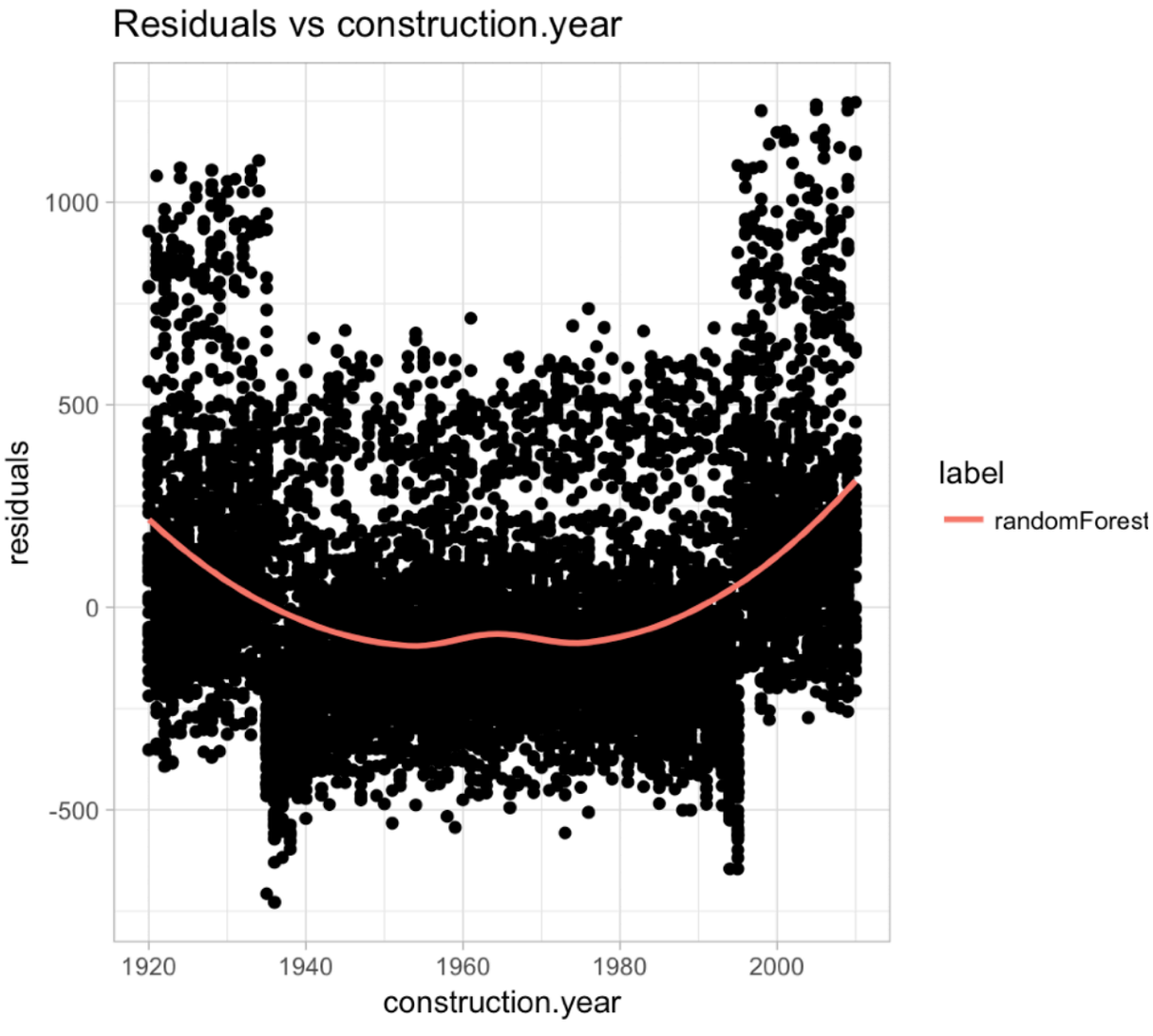
```
plot(mp_lm, mp_rf, geom = "boxplot")
```

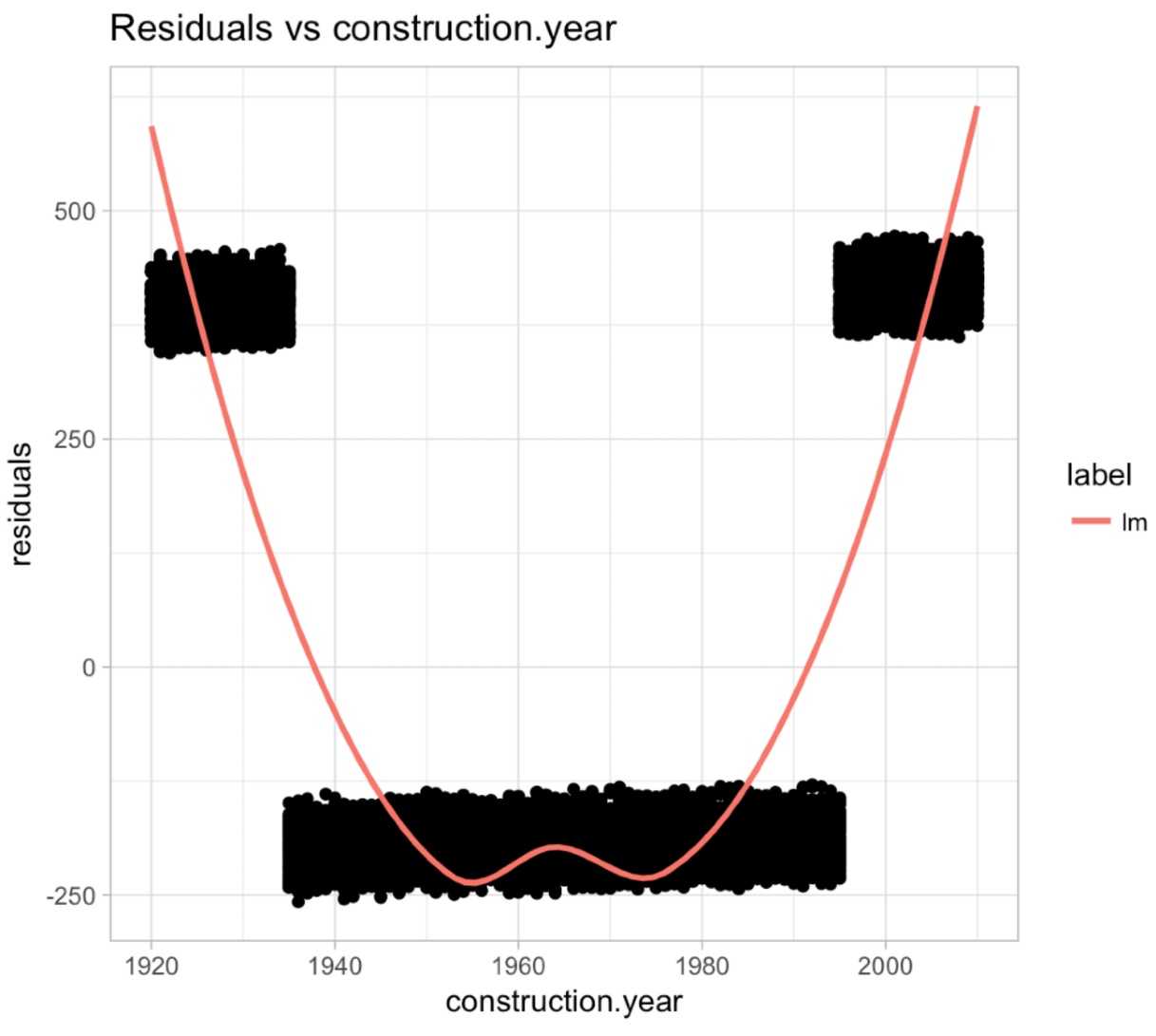


How good are these models?



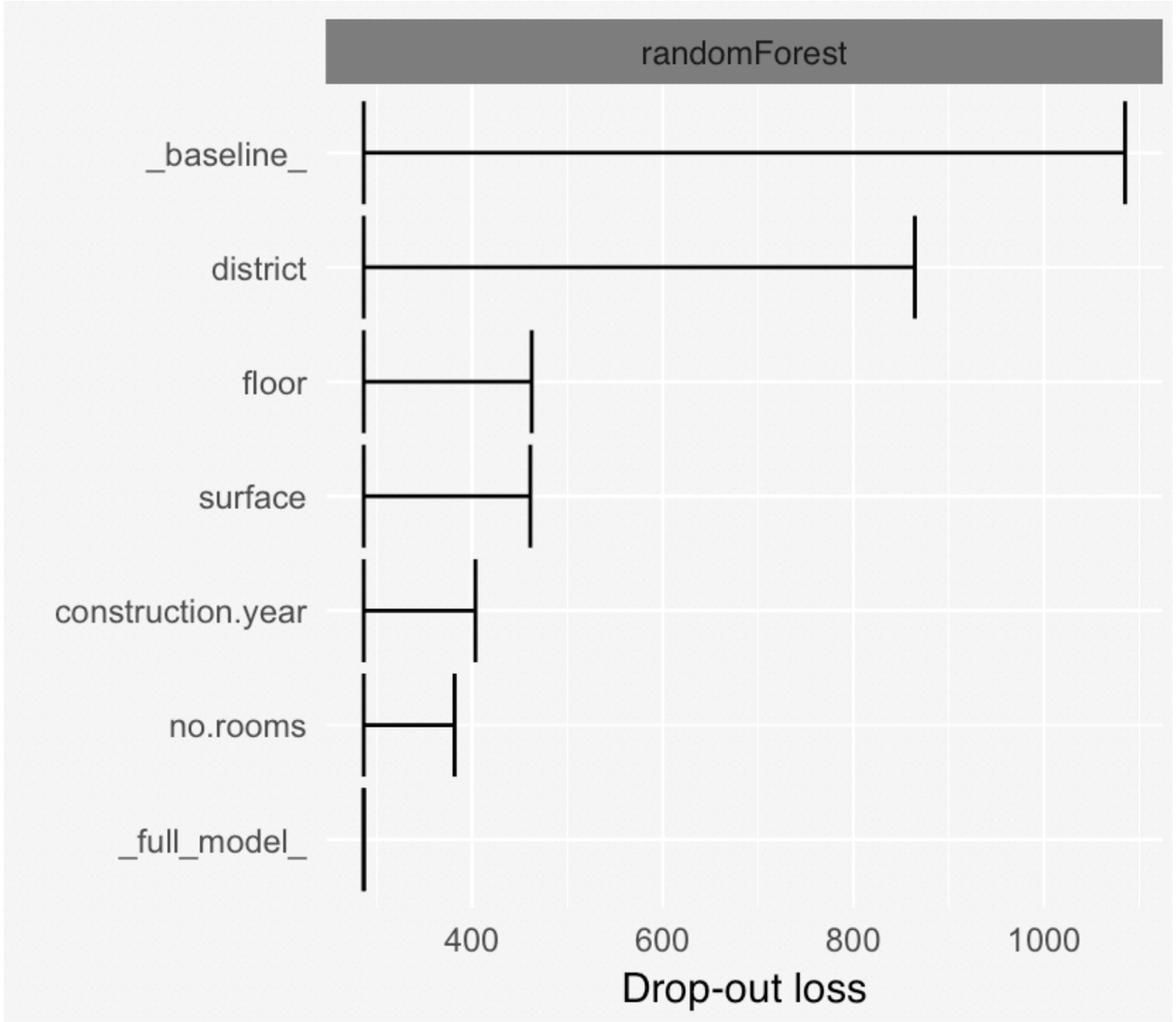
```r
library(auditor)
audit_rf <- audit(explainer_rf)
plotResidual(audit_rf, variable = "construction.year")

audit_lm <- audit(explainer_lm)
plotResidual(audit_lm, variable = "construction.year")
```



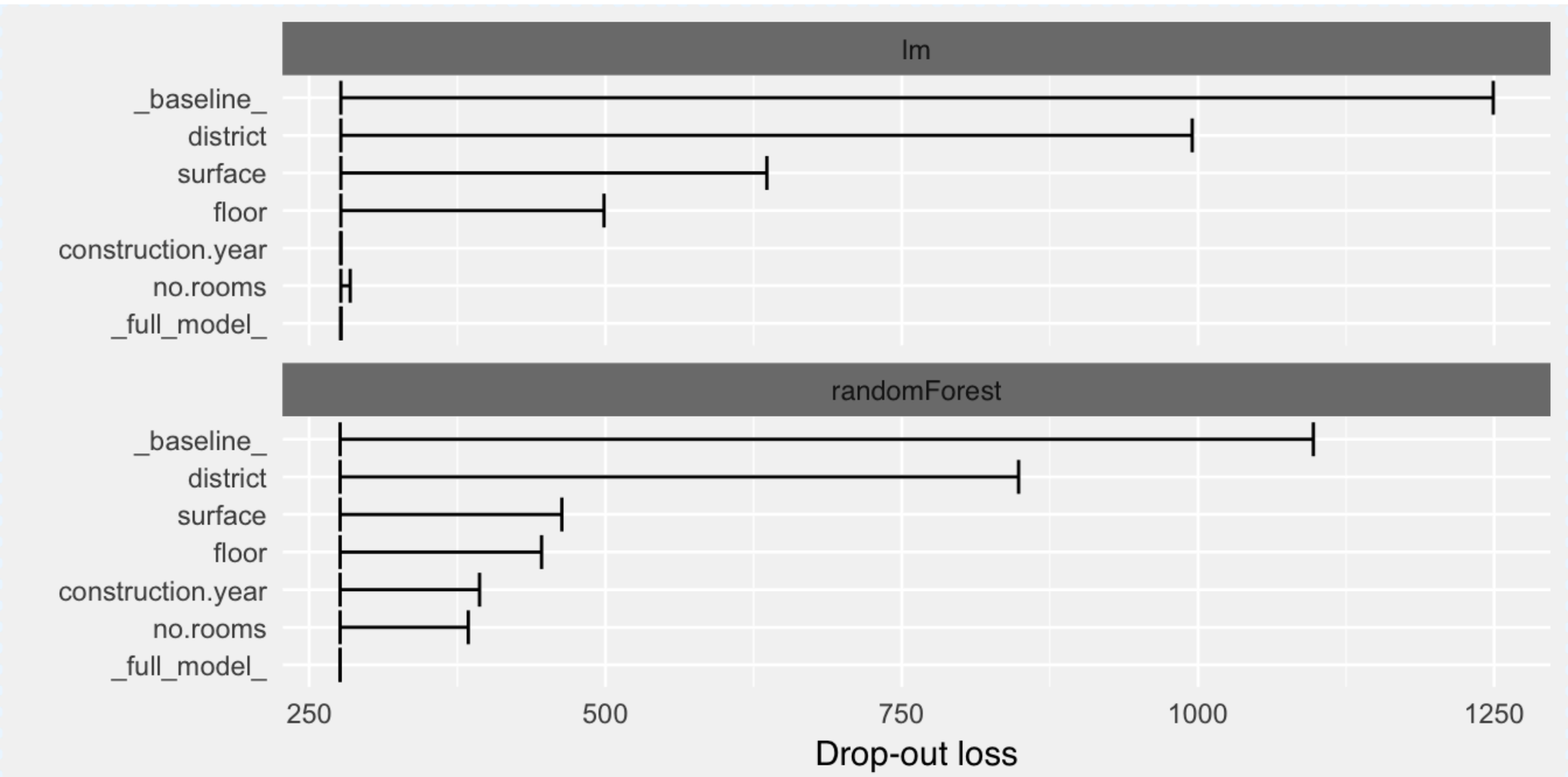


Which variables are important?

```
> vi_rf <- variable_importance(explainer_rf, loss_function = loss_root_mean_square)
> vi_rf
           variable dropout_loss         label
1      _full_model_     286.2676 randomForest
2          no.rooms     381.5975 randomForest
3 construction.year     403.4376 randomForest
4           surface     461.0018 randomForest
5             floor     462.3999 randomForest
6          district     864.4315 randomForest
7         _baseline_    1084.9218 randomForest
>
> plot(vi_rf)
```



Which variables are important?



Agenda

How to understand a black-box model?

Choose the right visual explainer in 2.875 simple steps

1. Want to understand a model or a single prediction?
- entire model
- prediction for a single observation

2. Is it *how to change it* or *why it happened*?
- interested in *what-if* scenarios
- how variables affected this single prediction

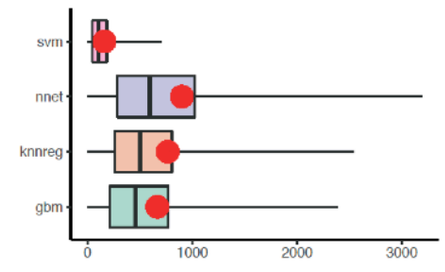
3. Variable attribution or importance?
- decompose prediction (breakDown, Shapley)
- identify key features (live, LIME)

3. Evaluate performance or validate fit?
- compare models performance
- audit residuals and goodness of fit

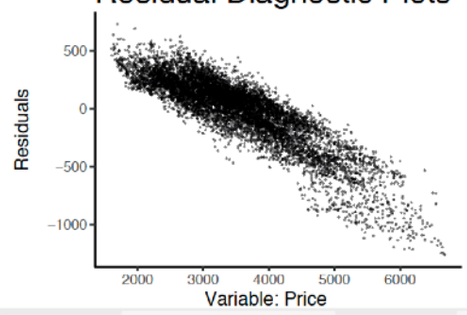
2. Interested in model performance or structure?
- how good is the model
- how does it work

3. Which variable are you interested in?
- all
- a categorical
- a continuous

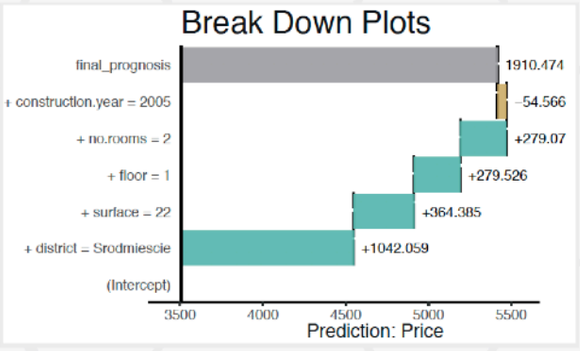
Model Performance Plots



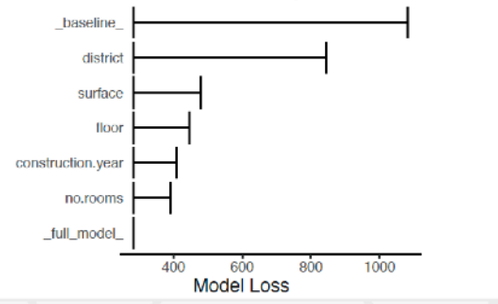
Residual Diagnostic Plots



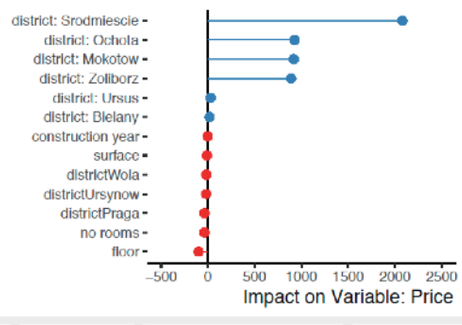
Break Down Plots



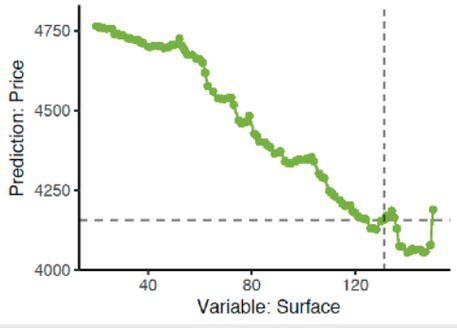
Variable Importance Plots



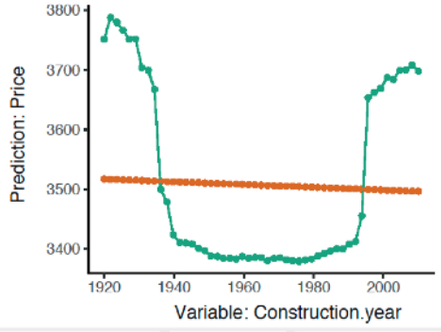
Local Variable Importance



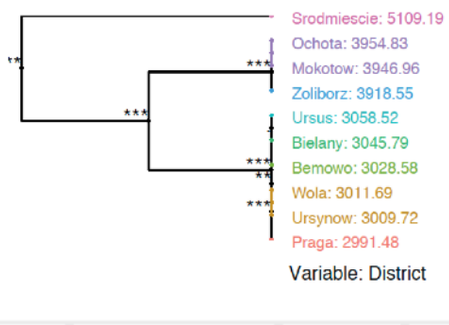
Ceteris Paribus Plots



Partial Dependency Plots



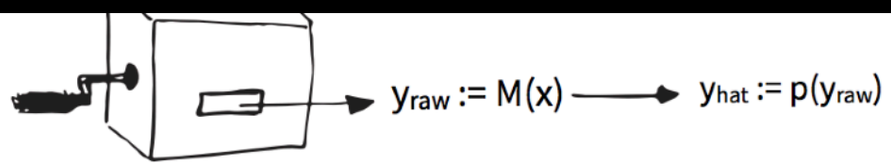
Merging Path Plots



Find more at:
https://github.com/pbiecek/DALEX

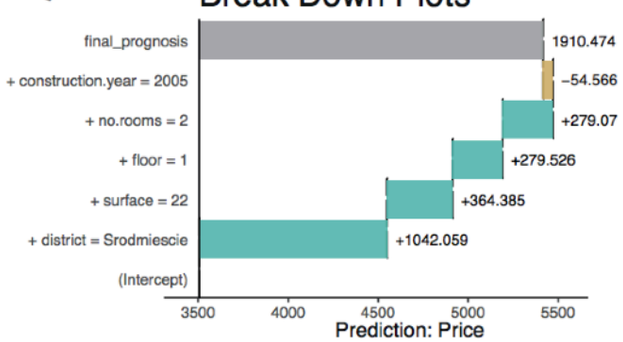
DALEX architecture

A) model

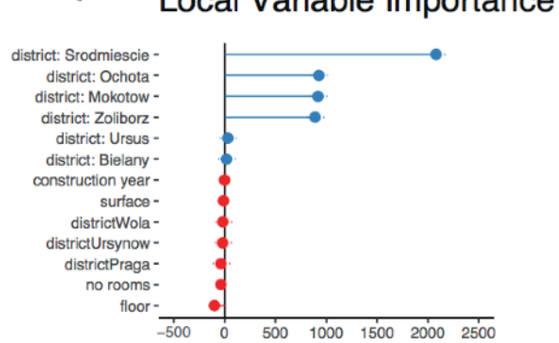


B) explain(model; data; y; predict_function)

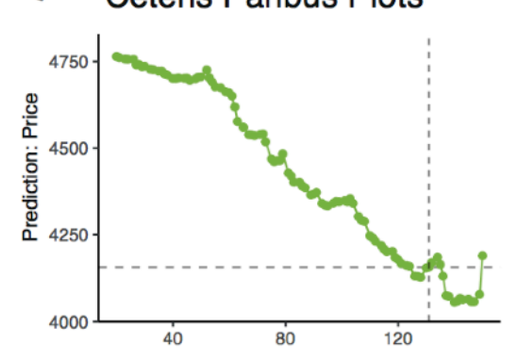
C)



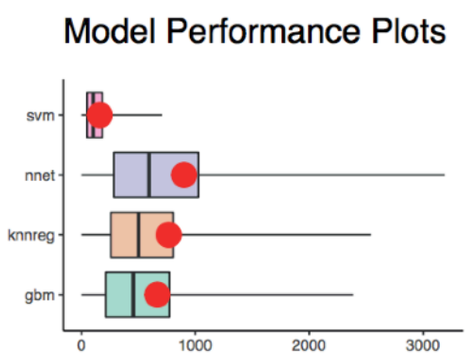
D)



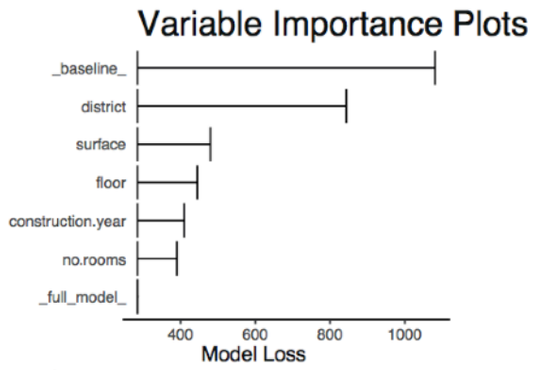
E)



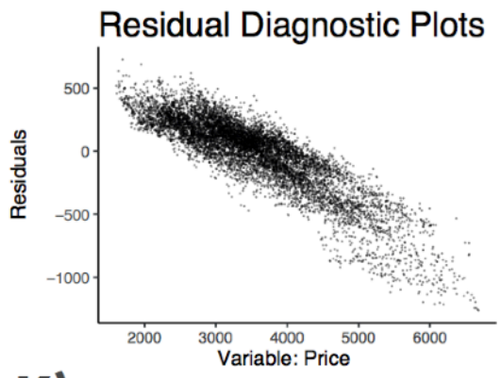
F)



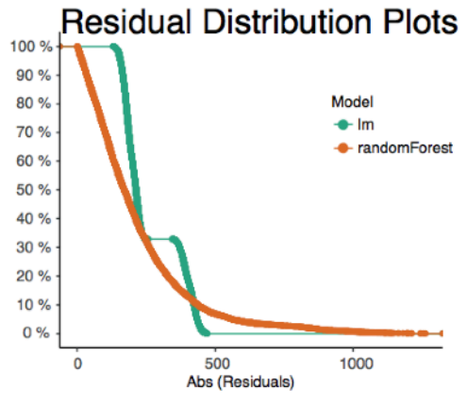
G)



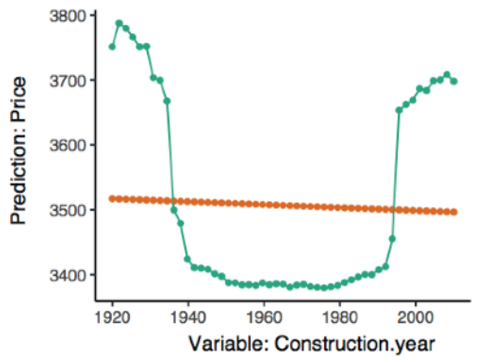
H)



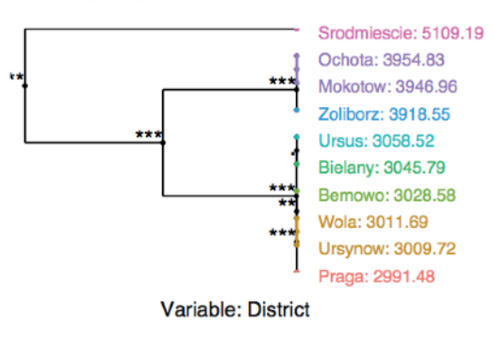
I)



J)



K)



Wrap a model

↓

Create an explainer

↓

Plot the explainer

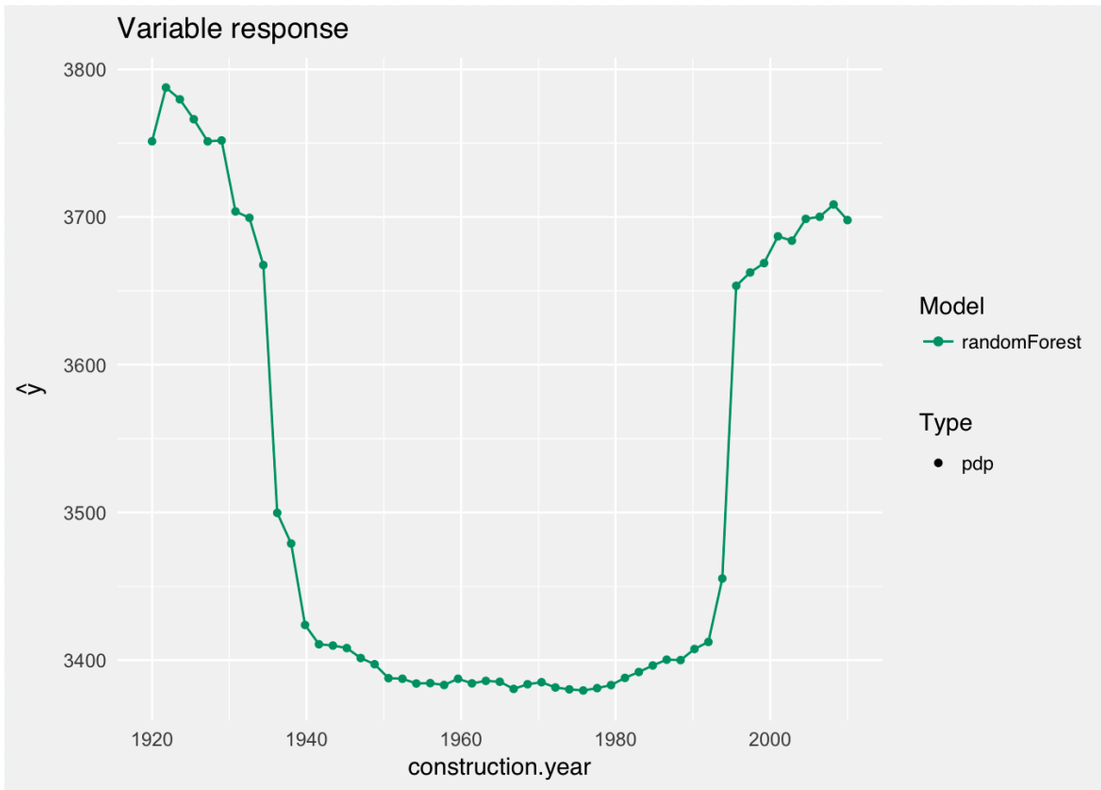
DALEX architecture

```
# wrap model
explainer_rf <- explain(apartments_rf_model,
                        data = apartmentsTest[,2:6],
                        y    = apartmentsTest$m2.price)
# create explainer
sv_rf  <- single_variable(explainer_rf,
                        variable = "construction.year",
                        type     = "pdp")
```

Wrap a model

Create an explainer

```
# plot explainer
plot(sv_rf)
```



Plot the explainer

modelDown: pkgdown for models

https://github.com/MI2DataLab/modelDown

modelDown

build passing

`modelDown` generates a website with HTML summaries for predictive models. Is uses DALEX explainers to compute and plot summaries of how given models behave. We can see how exactly scores for predictions were calculated (Prediction BreakDown), how much each variable contributes to predictions (Variable Response), which variables are the most important for a given model (Variable Importance) and how well out models behave (Model Performance).

`pkgdown` documentation: https://mi2datalab.github.io/modelDown/

An example website for regression models: https://mi2datalab.github.io/modelDown_example/

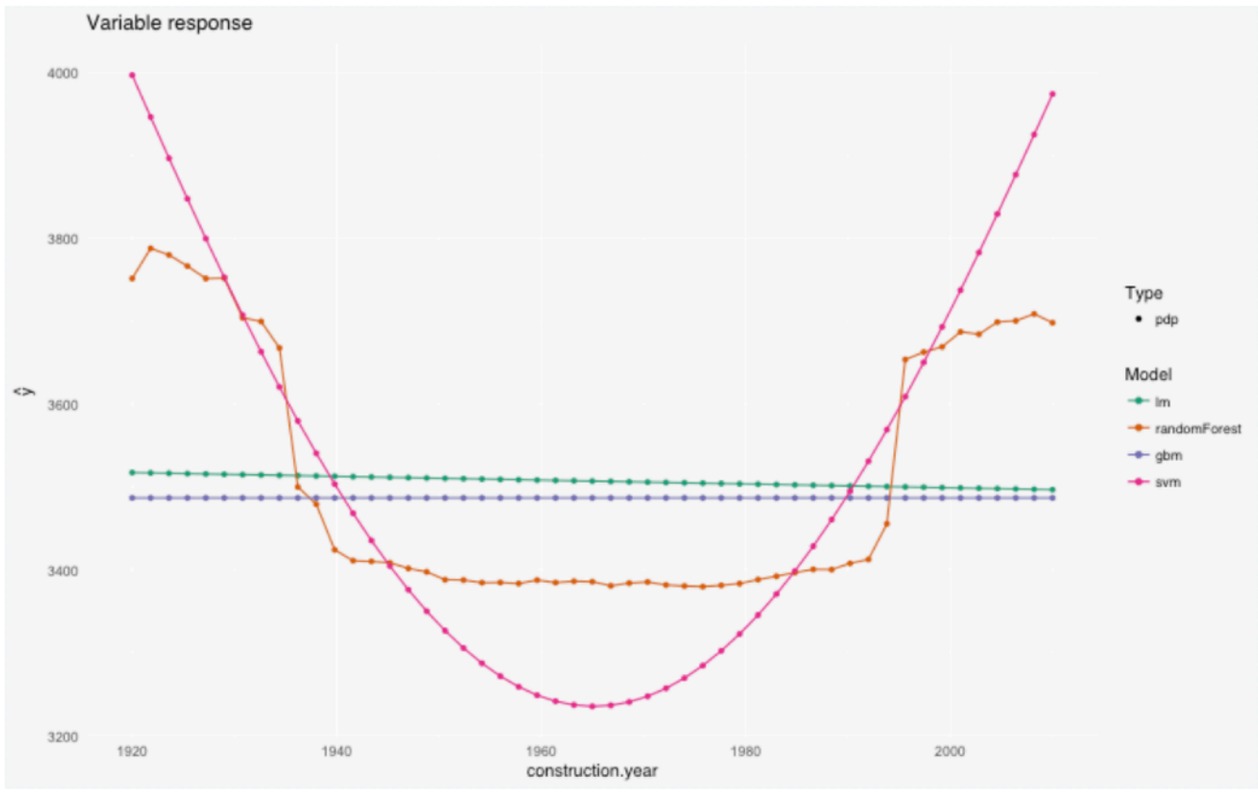
Getting started

Do you want to start right now ? Check out our getting started guide.

modelDown | Model Performance | Variable Importance | Variable Response | Prediction BreakDown

- construction.year
- district
- floor
- no.rooms
- surface

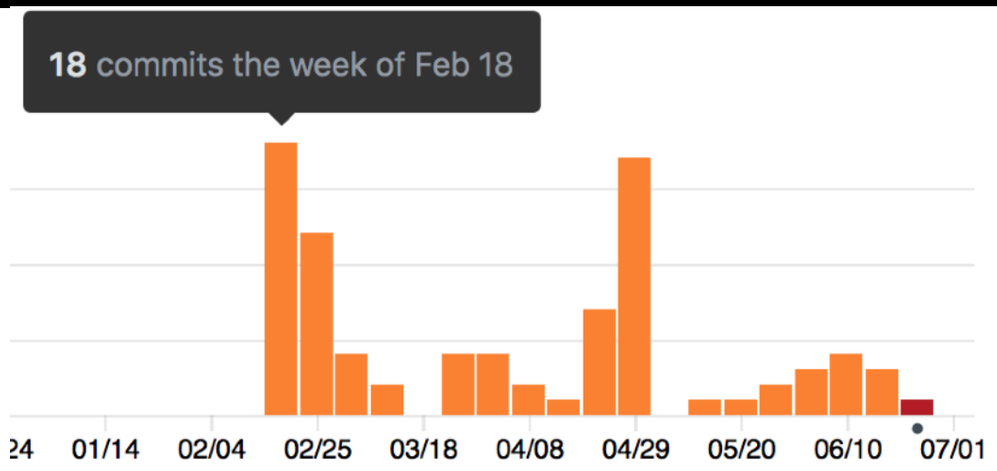
construction.year



https://mi2datalab.github.io/modelDown_example/

DALEX is very young

First commit: 2/18/2018



Active development.
DALEX is meant to be
an *unified interface* to model explainers it's just a glue.

DALEX invasion

- Talk: Complexity Institute @ NTU Singapore, March 2018
- Talk: SER meeting in Warsaw, April 2018
- Workshop: eRum conference, May 2018
- Workshop: STWUR meeting in Wrocław, June 2018
- Workshop: Why R? conference in Wrocław, July 2018
- Workshop: useR! conference in Brisbane, July 2018
- ...

Find more at https://github.com/pbiecek/DALEX

] | https://github.com/pbiecek/DALEX

DALEX

CRAN 0.2.3 | downloads 1841/month | downloads 4050 | build passing | coverage 92%

DALEX: Descriptive mAchine Learning EXplanations

Machine Learning models are widely used and have various applications in classification or regression tasks. Due to increasing computational power, availability of new data sources and new methods, ML models are more and more complex. Models created with techniques like boosting, bagging of neural networks are true black boxes. It is hard to trace the link between input variables and model outcomes. They are use because of high performance, but lack of interpretability is one of their weakest sides.

In many applications we need to know, understand or prove how input variables are used in the model and what impact do they have on final model prediction. `DALEX` is a set of tools that help to understand how complex models are working.

Find more about DALEX in this Gentle introduction to DALEX with examples.

DALEX Stories

- An interactive notebook with examples: launch binder

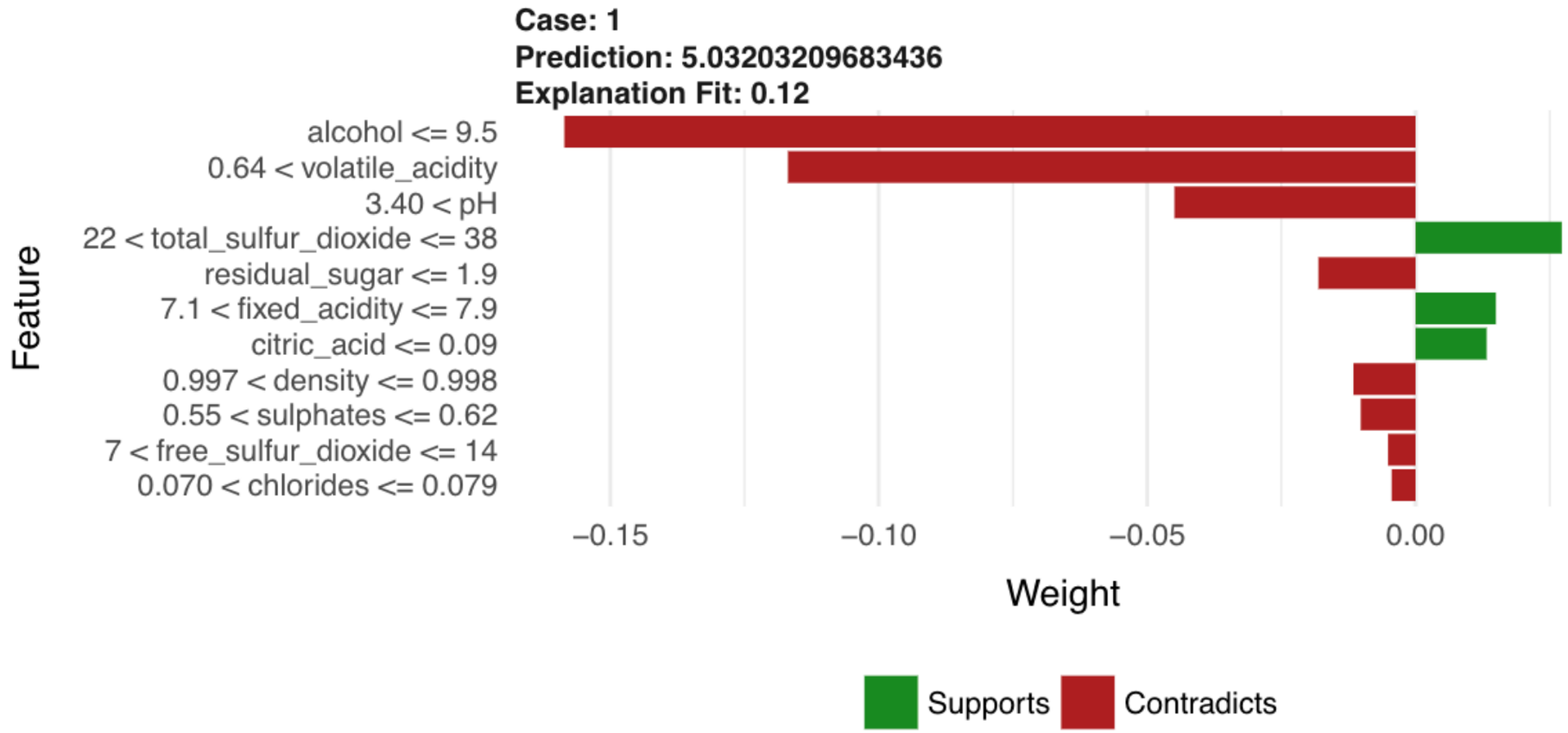
How to use DALEX

- How to use DALEX with caret
- How to use DALEX with mlr
- How to use DALEX with H2O
- How to use DALEX with xgboost package
- How to use DALEX for teaching. Part 1
- How to use DALEX for teaching. Part 2
- breakDown vs lime vs shapleyR

LIME / live

VS

Break Down



(a) Husky classified as wolf

(b) Explanation