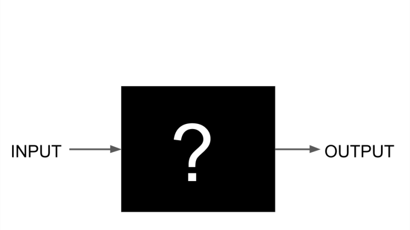
Interpretable Machine Learning in R with iml

Christoph Molnar

2018-07-05

IML theory

INTERPRETABLE MACHINE LEARNING



- Machine learning (ML) has huge potential to improve research, products and processes
- ML models usually operate as intransparent black boxes
- The lack of explanation hurts trust and creates barrier for adoption

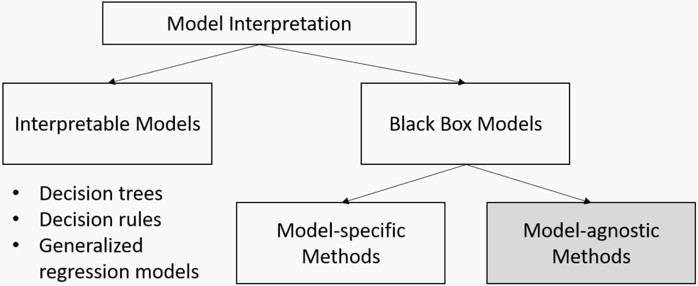
$\Rightarrow$ We need interpretability for machine learning models

WHEN DO WE NEED INTERPRETABILITY?

- Debugging the models
- Increasing trust
- Newly developed systems with unknown consequences
- Decisions about humans
- Critical applications that decide about life and death
- Models using proxies instead of causal inputs
- When the loss function does not cover all constraints

Doshi-Velez, F., and Kim, B. (2017). Towards A Rigorous Science of Interpretable Machine Learning, (Ml), 1-13. Retrieved from http://arxiv.org/abs/1702.08608

WHAT TOOLS DO WE HAVE?



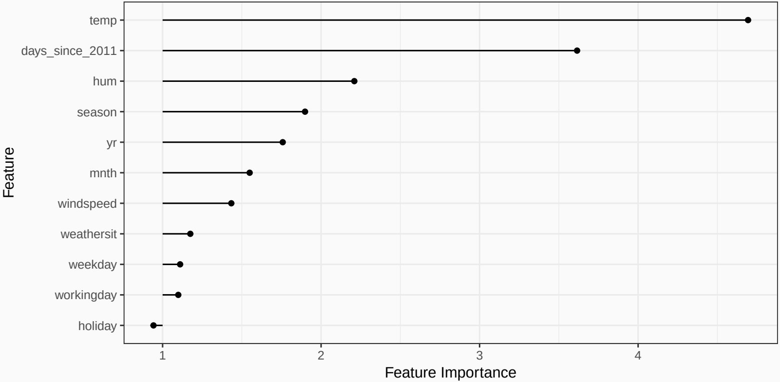
- Decision trees
- Decision rules
- Generalized regression models

- Random Forest Explainer
- Visualizing activations of neural networks

Advantage: Flexibility
- Underlying models can be exchanged
- Several IML Visualizations can be used

PERMUTATION FEATURE IMPORTANCE

- Calculates the increase of the model's prediction error after permuting the feature
- Features are important if permuting one feature's value increases the model error



Fisher, A., Rudin, C., and Dominici, F. (2018). Model Class Reliance.

PERMUTATION FEATURE IMPORTANCE

1. Estimate model error on test data
2. For each feature x_j

- Shuffle the feature

original

x_1	$\dots$	x_j	$\dots$	x_p
3		1.4		6.0
5		1.2		7.2
$\dots$		$\dots$		$\dots$
6		2.0		8.9

$\Rightarrow$

shuffled x_j

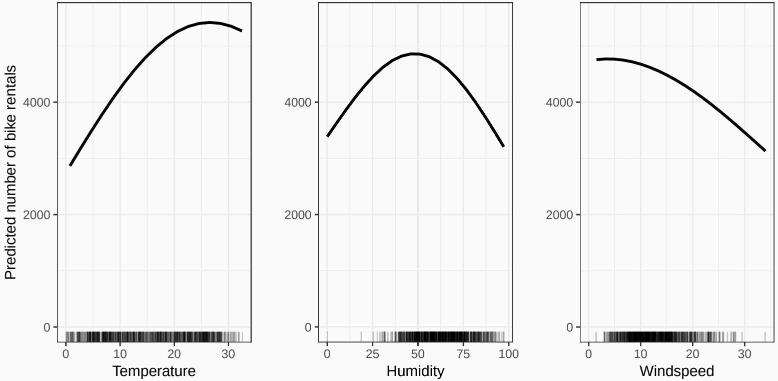
x_1	$\dots$	x_j	$\dots$	x_p
3		2.0		6.0
5		1.4		7.2
$\dots$		$\dots$		$\dots$
6		1.2		8.9

- Estimate the error of the model after shuffling
- Calculate importance as increase in error
- Average the feature importance over shuffle repetitions

PARTIAL DEPENDENCE PLOTS

Show the marginal effect of a feature on the predicted outcome of a fitted model

$$f_{x_S}(x_S) = \mathbb{E}_{x_C} f(x_S, x_C)$$



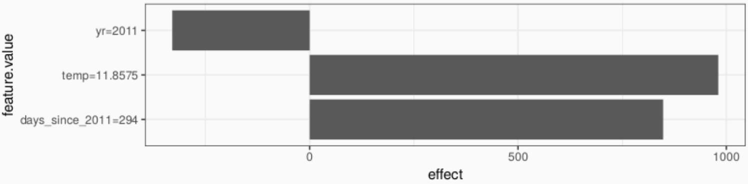
Friedman, J.H. 2001. "Greedy Function Approximation: A Gradient Boosting Machine." Annals of Statistics 29: 1189-1232.

PARTIAL DEPENDENCE PLOTS

- Select a feature x_j
- Choose grid points along x_j
- For each grid point:
  - Overwrite feature x_j in the dataset with the current grid value
  - Get the predictions for these points from the ML model
  - Average the predictions
- Draw a curve with the grid points on the x-axis and the average prediction on the y-axis.

Local Interpretable model-agnostic Explanations

- Fits local, interpretable models that can explain single predictions of any black-box model
- Local surrogate models, that are interpretable like a LM or CART and are learned on predictions of original model

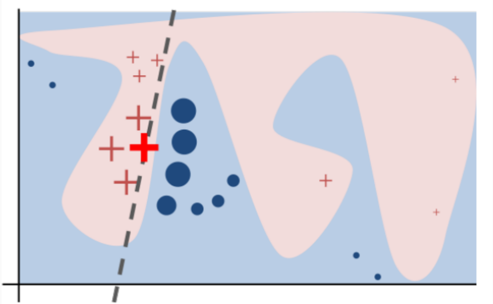


Ribeiro, M. T., (2016, August). Why should i trust you?: Explaining the predictions of any classifier

LIME

How to fit local surrogate model

1. Choose instance of interest x
2. Perturb data and get black box predictions for them
3. Weight new samples by their proximity to x
4. Fit a weighted, interpretable model on this new data set



Ribeiro, M. T., (2016, August). Why should i trust you?: Explaining the predictions of any classifier

IML examples

THE `IML` PACKAGE

- R6 package for **model-agnostic** Interpretable Machine Learning methods
- Analyses a fixed machine learning model
- Available on CRAN and Github: https://github.com/christophM/iml
- Detailed explanations for the methods can be found in the book “Interpretable Machine Learning”: https://christophm.github.io/interpretable-ml-book/agnostic.html

Molnar et al., (2018). iml: An R package for Interpretable Machine Learning . Journal of Open Source Software, 3(26), 786, https://doi.org/10.21105/joss.00786

PACKAGE `IML`

The `iml` package contains the following IML tools

- Permutation Feature Importance (`FeatureImp`)
- Feature Interactions (`Interaction`)
- Partial Dependence Plots (`Partial`)
- LIME (`LocalModel`)
- Shapley Values (`Shapley`)
- Tree Surrogates (`TreeSurrogate`)

EXAMPLE

- Load neccessary packages

```r
library(mlr)
library(iml)
```

- Import data

```r
load("bike.RData")
```

THE BIKE DATA SET

name	type	mean	nlevs
season	factor	NA	4
yr	factor	NA	2
mnth	factor	NA	12
holiday	factor	NA	2
weekday	factor	NA	7
workingday	factor	NA	2
weathersit	factor	NA	3
temp	numeric	15.3	0
hum	numeric	62.8	0
windspeed	numeric	12.8	0
cnt	integer	4504.3	0
days_since_2011	numeric	365.0	0

FIT MLR MODEL AND CREATE IML PREDICTOR

- We have to fit a ML model first

```
task = makeRegrTask(data = bike, target = "cnt")
lrn = makeLearner("regr.randomForest")
mod = train(lrn, task)
```

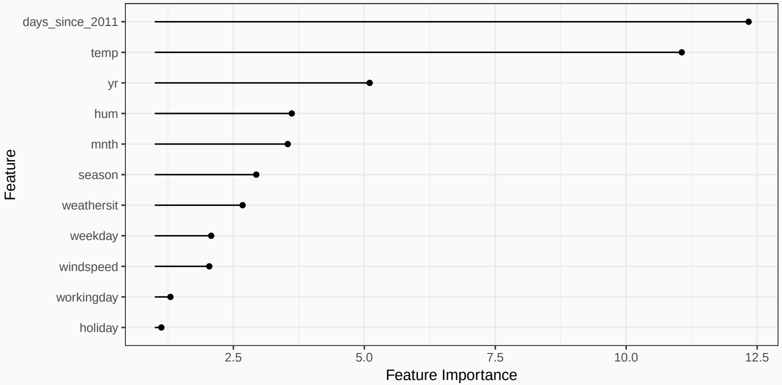
- We can use one IML model for all methods

```
# Create data frame without target column
bike.x = bike[names(bike) != 'cnt']

predictor = Predictor$new(mod, data = bike.x, y = bike$cnt)
```

PERMUTATION FEATURE IMPORTANCE PLOT

```r
importance = FeatureImp$new(predictor, loss = 'mse')
plot(importance)
```



ACCESS RESULTS IN TABLE FORMAT

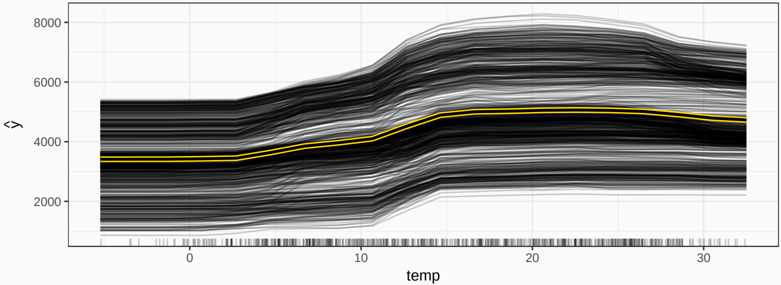
- All results can be viewed in table form

```
importance$results

##           feature original.error permutation.error importance
## 1  days_since_2011          94220           1162402      12.34
## 2             temp          94220           1042154      11.06
## 3               yr          94220            480761       5.10
## 4              hum          94220            340594       3.61
## 5             mnth          94220            333412       3.54
## 6           season          94220            276730       2.94
## 7       weathersit          94220            252174       2.68
## 8          weekday          94220            195678       2.08
## 9        windspeed          94220            192280       2.04
## 10      workingday          94220            122380       1.30
## 11         holiday          94220            106049       1.13
```

PARTIAL DEPENDENCE PLOT

```r
pdp = Partial$new(predictor, "temp", ice = TRUE)
pdp$plot()
```

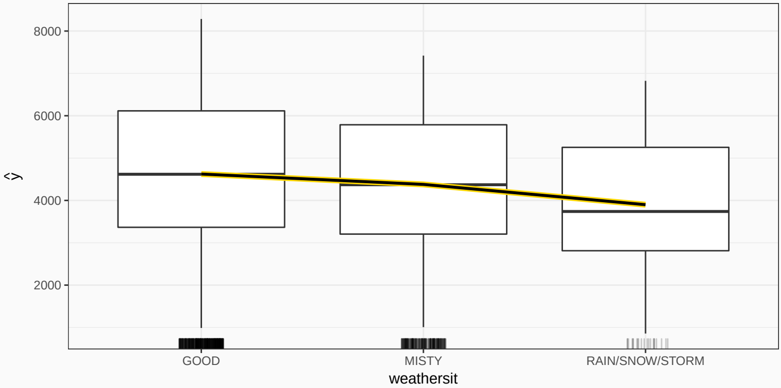


- `ice = TRUE`: Individual Conditional Expectation (ICE) Plots visualizes the relationship between the predicted response and the feature for *individual* observations

REUSE PD OBJECTS

- PD objects can be reused, e.g. for fitting other features

```r
pdp$set.feature("weathersit")
pdp$plot()
```



LIME PLOT

- Select one instance (ml model prediction is 4262.193)

```
bike.x[295,]

##     season   yr mnth    holiday weekday     workingday
## 295 WINTER 2011  OKT NO HOLIDAY     SAT NO WORKING DAY
##     weathersit temp  hum windspeed days_since_2011
## 295       GOOD 11.9 62.9      6.21             294
```

```
lim = LocalModel$new(predictor, x.interest = bike.x[295,], k = 3)
plot(lim)
```