

# examples.R

*mtst*

*Tue Mar 20 13:34:48 2018*

```
# Examples: explaining single predictions.
# Libraries & data ----
library(e1071)
library(corrplot)

## corrplot 0.84 loaded

library(lime)
library(breakDown)
library(live)

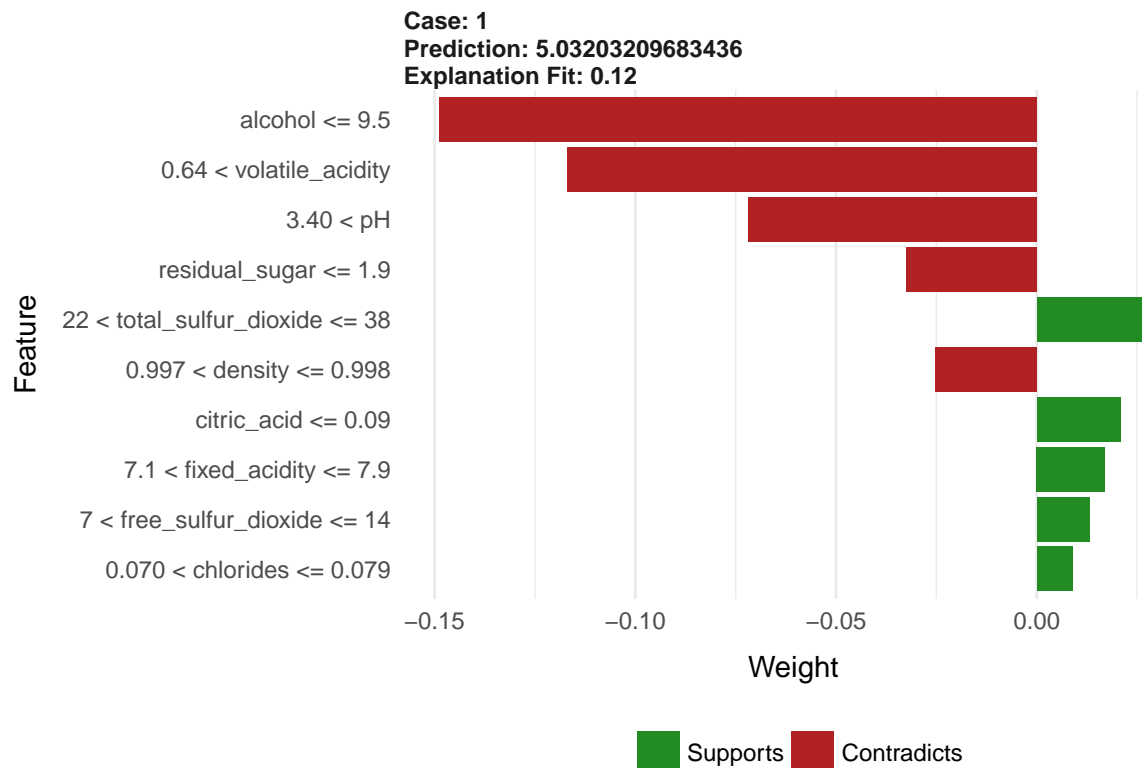
## Loading required package: mlr
## Loading required package: ParamHelpers
##
## Attaching package: 'mlr'
## The following object is masked from 'package:e1071':
##
##      impute
library(shapleyr)

## Loading required package: checkmate
## Loading required package: ggrepel
## Loading required package: ggplot2
## Loading required package: reshape2
## Loading required package: shiny
## Loading required package: shinydashboard
##
## Attaching package: 'shinydashboard'
## The following object is masked from 'package:graphics':
##
##      box
## Welcome to the ShapleyR package!
library(mlr)
library(tidyr)

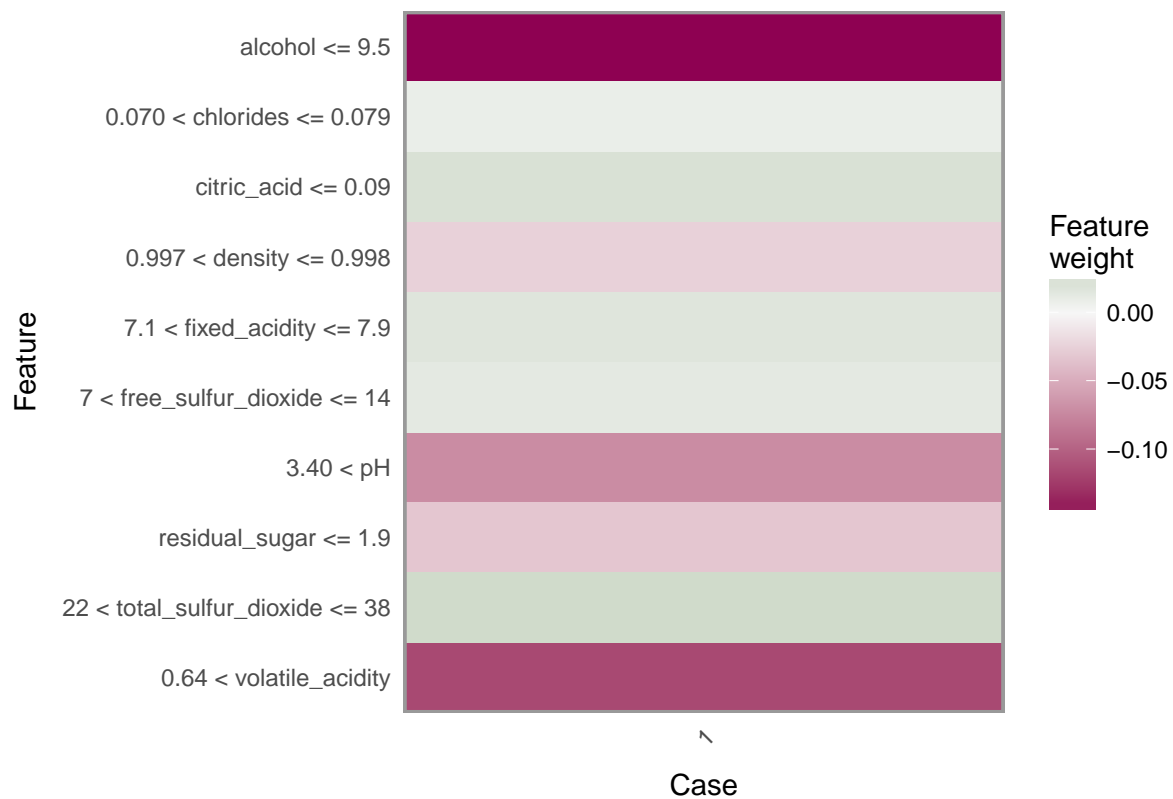
##
## Attaching package: 'tidyr'
## The following object is masked from 'package:reshape2':
##
##      smiths
```

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
## The following object is masked from 'package:lime':
##
##   explain
## The following objects are masked from 'package:stats':
##
##   filter, lag
## The following objects are masked from 'package:base':
##
##   intersect, setdiff, setequal, union
load("winequality_red.rda")
nc <- which(colnames(winequality_red) == "quality")
nobs <- winequality_red[5, -nc]
# Model ----
wine_svm <- svm(quality ~., data = winequality_red)
# LIME ----
wine_expl <- lime(winequality_red[, -nc], wine_svm)
model_type.svm <- function(x, ...) "regression"
svm_explained <- lime::explain(nobs, wine_expl, n_features = 10)
# LIME: plot
plot_features(svm_explained)
```

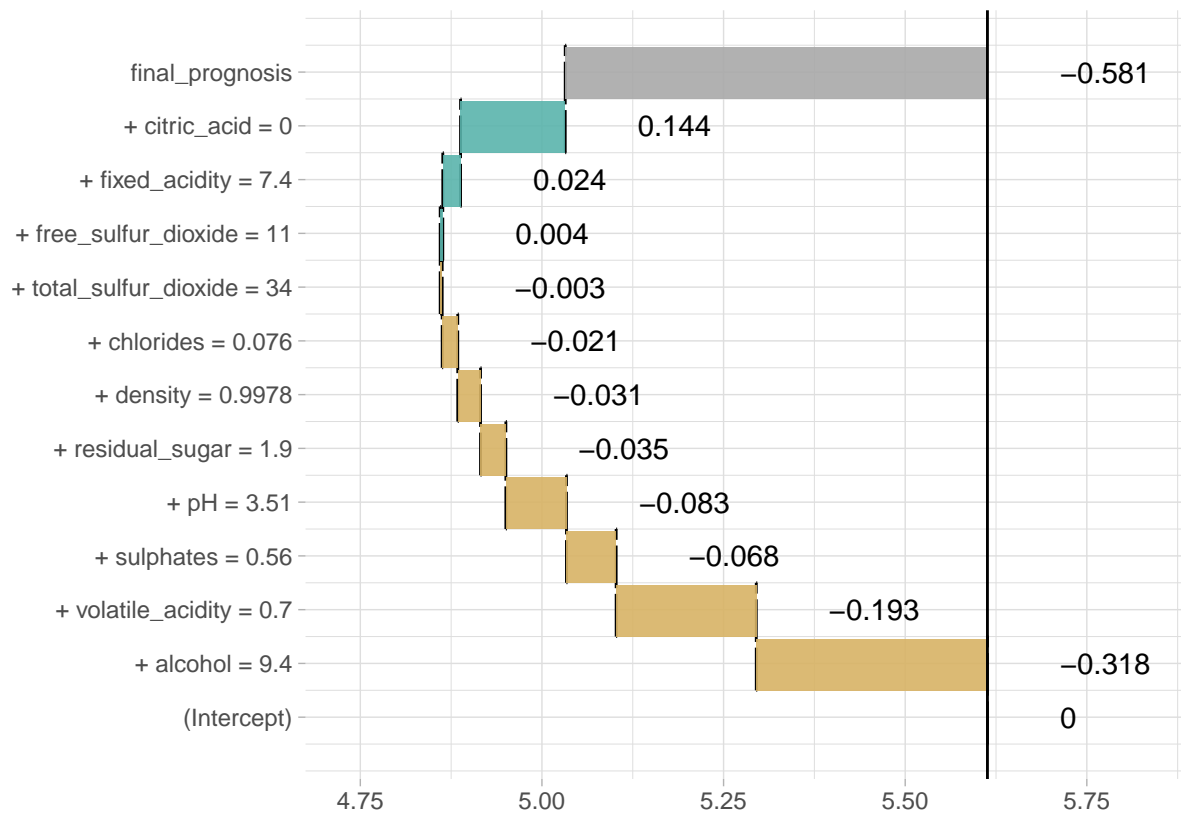


```
plot_explanations(svm_explained)
```

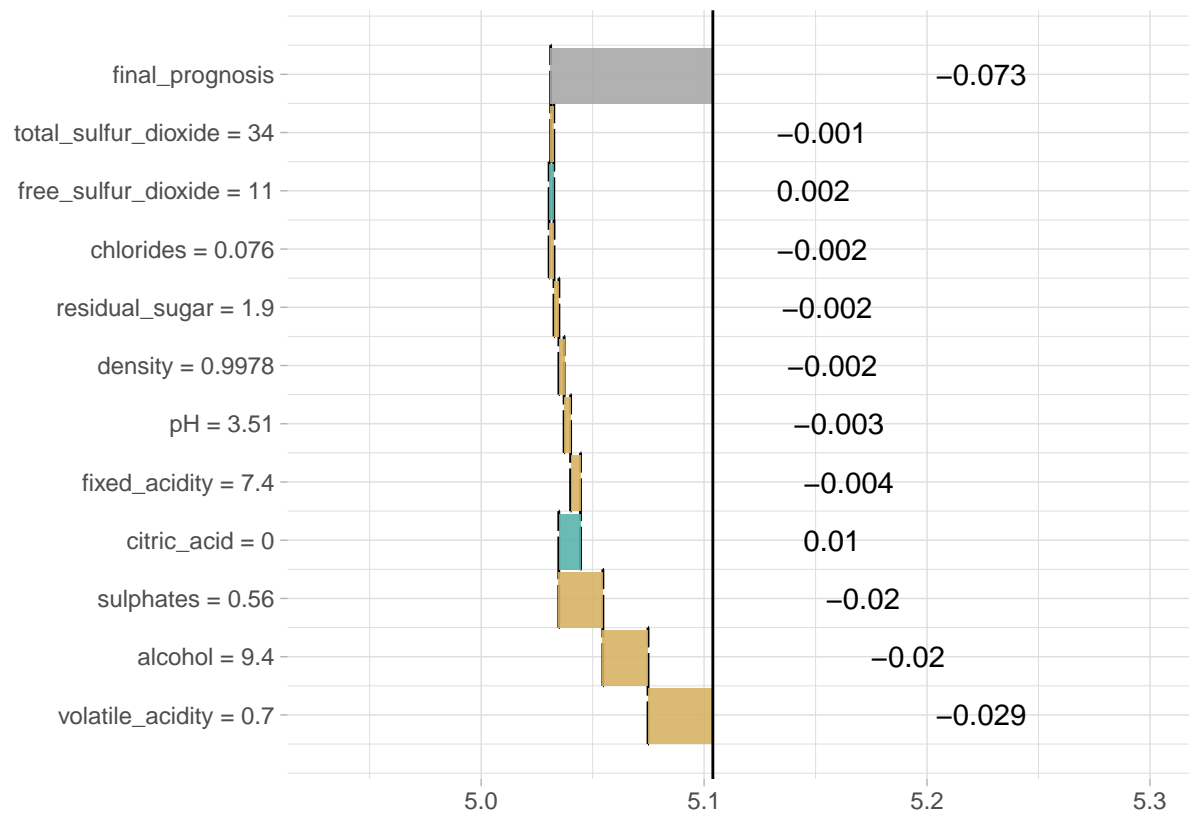


```
# breakDown ----
explain_bd <- broken(wine_svm, new_observation = nobs,
  data = winequality_red[, -nc],
  predict.function = kernlab::predict,
  baseline = "Intercept")

# breakDown: plot
plot(explain_bd)
```














```
# live
wine_sim <- sample_locally(winequality_red, winequality_red[5, ], "quality", 500)
wine_sim_svm <- add_predictions(winequality_red, wine_sim, wine_svm)
wine_expl_live <- fit_explanation(wine_sim_svm, "regr.lm")
plot_explanation(wine_expl_live, "waterfallplot", winequality_red[5, ])
```



```
plot_explanation(wine_expl_live, "forestplot", winequality_red[5, ])
```

```
## Warning in recalculate_width_panels(panel_positions, mapped_text =
## mapped_text, : Unable to resize forest panel to be smaller than its
## heading; consider a smaller text size
```

Variable	N	Estimate		p
fixed_acidity	500		0.10 (0.09, 0.11)	<0.001
volatile_acidity	500		-1.44 (-1.49, -1.38)	<0.001
citric_acid	500		-0.52 (-0.57, -0.46)	<0.001
residual_sugar	500		0.05 (0.04, 0.06)	<0.001
chlorides	500		1.27 (1.06, 1.48)	<0.001
free_sulfur_dioxide	500		-0.00 (-0.01, -0.00)	<0.001
total_sulfur_dioxide	500		0.00 (0.00, 0.00)	<0.001
density	500		-27.62 (-36.46, -18.77)	<0.001
pH	500		-0.17 (-0.24, -0.11)	<0.001
sulphates	500		2.42 (2.32, 2.52)	<0.001
alcohol	500		0.24 (0.23, 0.25)	<0.001

-30 -20 -10 0

```
# SHAP
# requires the use of mlr
tsk <- makeRegrTask("wine", winequality_red, "quality")

## Warning in makeTask(type = type, data = data, weights = weights, blocking
## = blocking, : Provided data is not a pure data.frame but from class tbl_df,
## hence it will be converted.

shp <- shapley(1, model = train("regr.lm", tsk), task = tsk)
# plot.shapley.singleValue(1, model = train("regr.lm", tsk), task = tsk)
brk <- data.frame(var = explain_bd$variable_name, explain_bd$contribution)
shp %>%
  gather(var, score) %>%
  full_join(brk, by = "var") %>%
  arrange(desc(abs(score))) %>%
  rename(bd_score = `explain_bd.contribution`)

## Warning: Column `var` joining character vector and factor, coercing into
## character vector

##           var  score  bd_score
## 1      alcohol -0.255 -0.318177246
## 2 volatile_acidity -0.153 -0.192861726
## 3          pH -0.086 -0.083412476
## 4      sulphates -0.067 -0.068399940
## 5      citric_acid  0.043  0.144089464
## 6 total_sulfur_dioxide  0.031 -0.002767048
## 7      fixed_acidity -0.030  0.024443379
```

## 8	chlorides	0.028	-0.021434037
## 9	free_sulfur_dioxide	-0.024	0.003582795
## 10	residual_sugar	-0.023	-0.034867286
## 11	density	-0.021	-0.031333714
## 12	Intercept	NA	0.000000000
## 13		NA	-0.581137835