04.Wizualizacje

Przydatne linki:

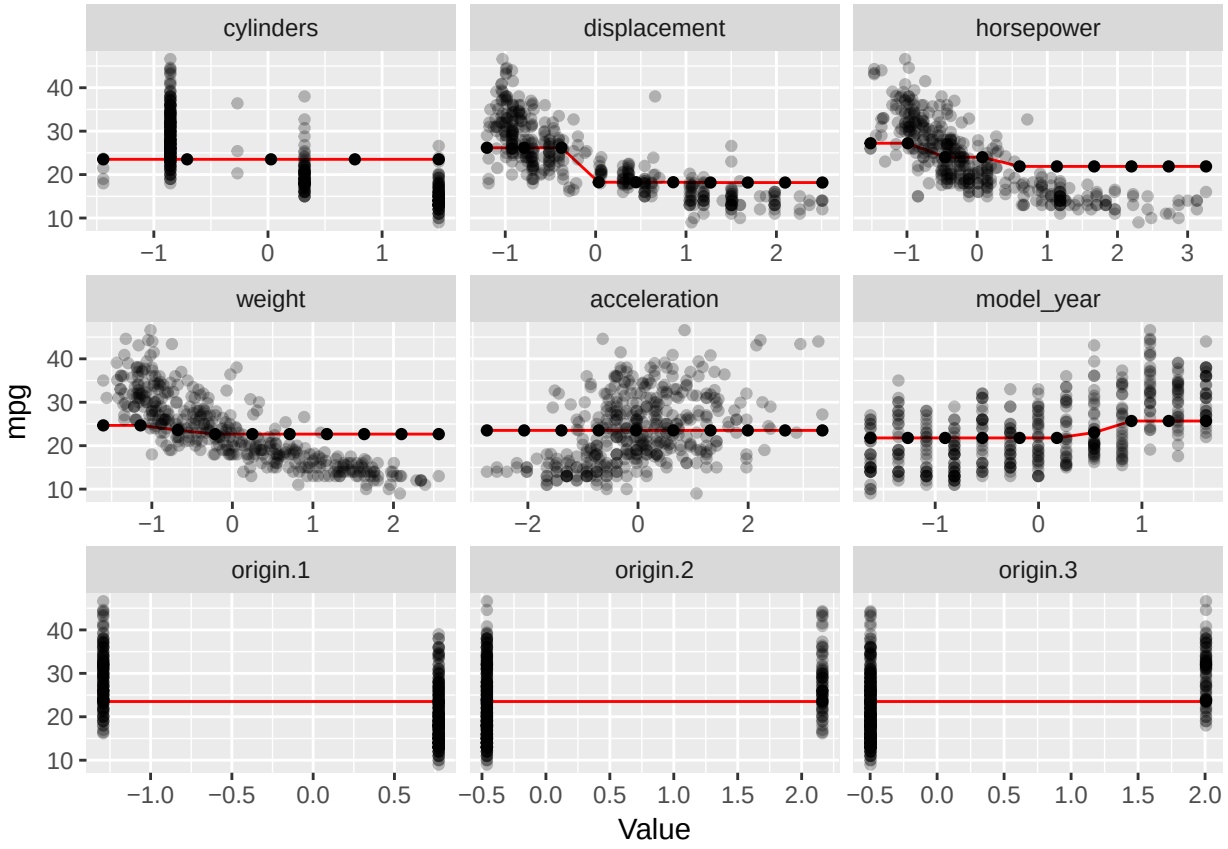
- https://mlr-org.github.io/mlr-tutorial/release/html/visualization/index.html#available-generation-and-plotting-function

Wizualizacje dla zadania `autoMpg`

Wpływ poszczególnych zmiennych na model

```
autoMpgTask = readRDS('data/01_task.RDS')

pd = generatePartialDependenceData(train('regr.rpart', autoMpgTask), autoMpgTask)
plotPartialDependence(pd, data = getTaskData(autoMpgTask))
```



Wpływ wartości hiperparametrów na model

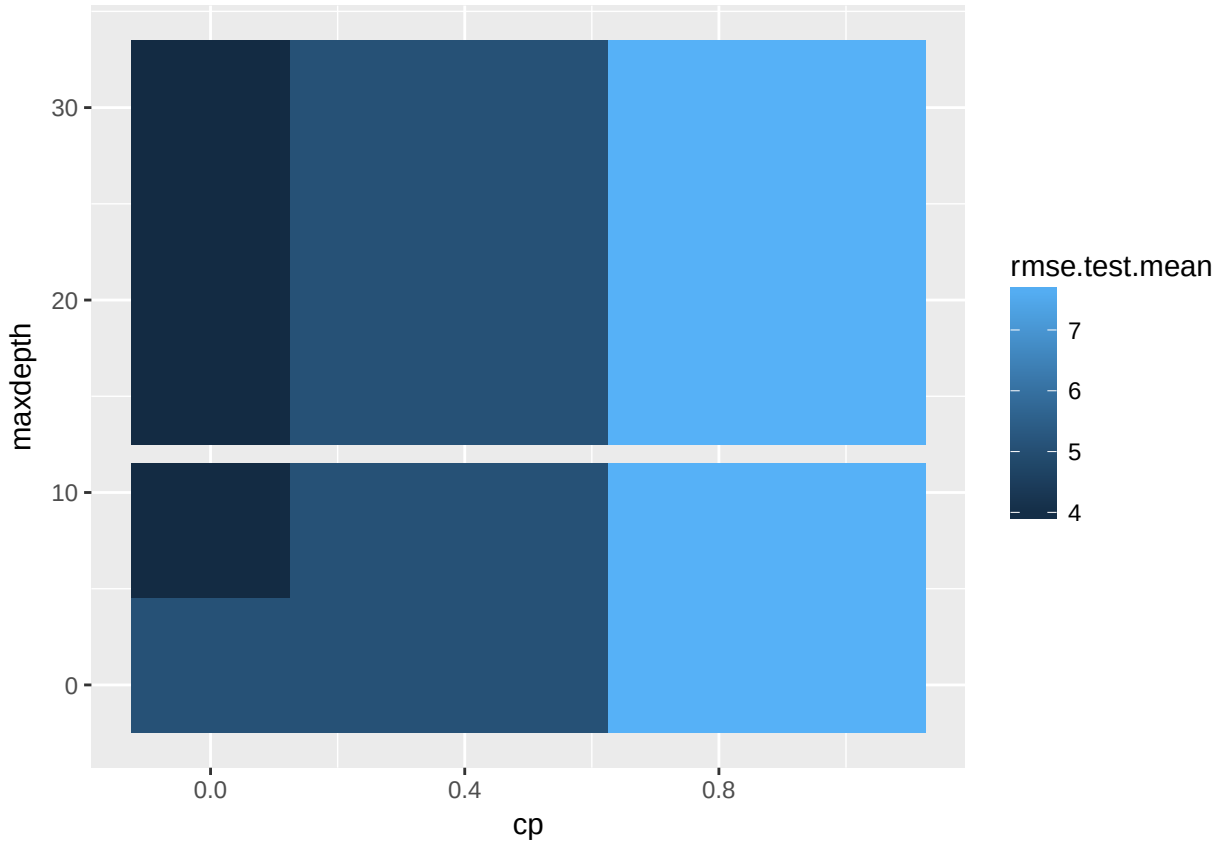
```
rpartLearner <- makeLearner('regr.rpart')

originalParamSet <- getParamSet(rpartLearner)
ps <- makeParamSet(
  originalParamSet$pars$cp,
  originalParamSet$pars$maxdepth
```

```
)

data  <- tuneParams(rpartLearner,
                    task = autoMpgTask,
                    control = makeTuneControlGrid(resolution = 5),
                    measures = setAggregation(rmse, test.mean),
                    resampling = makeResampleDesc("Holdout"),
                    par.set = ps, show.info = FALSE) %>%
                    generateHyperParsEffectData()

plotHyperParsEffect(data, x = "cp", y = "maxdepth", z = "rmse.test.mean", plot.type = "heatmap")
```



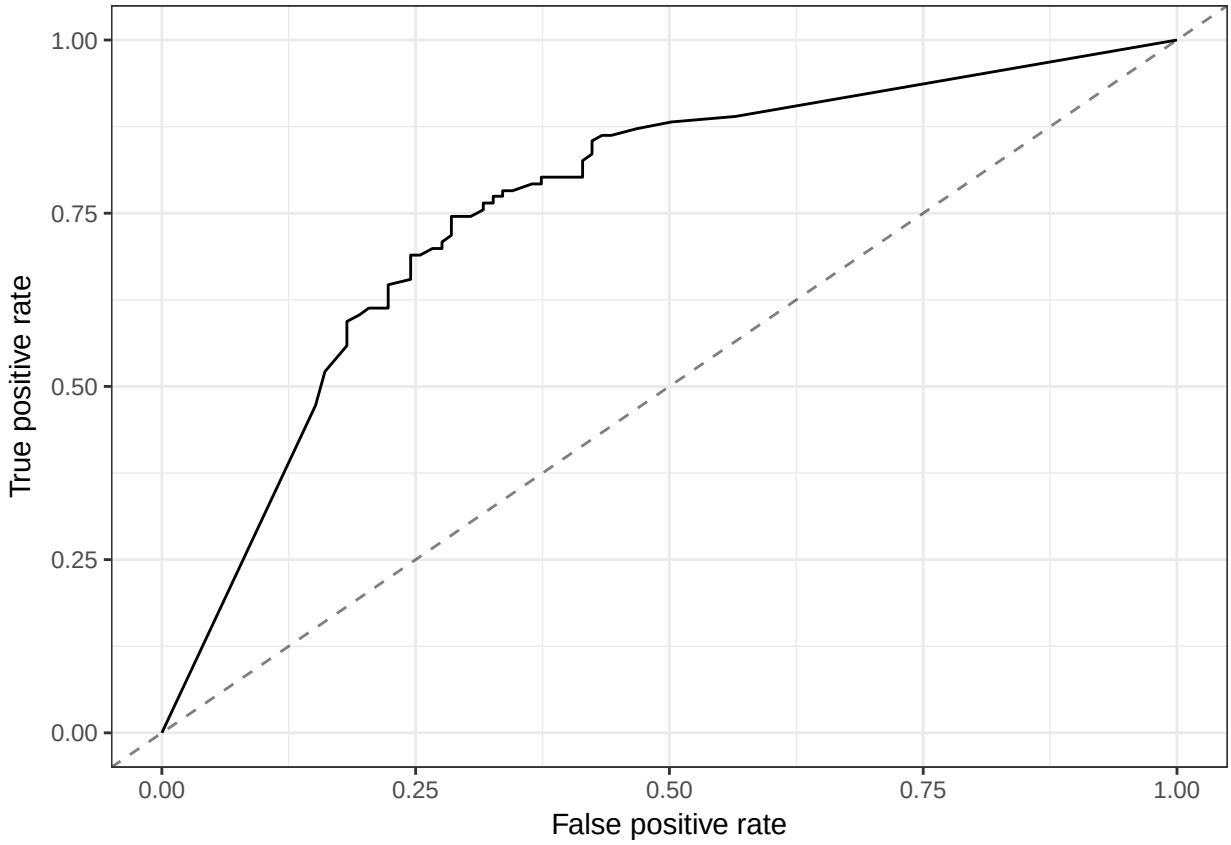
Wizualizacje dla zadania sonar

```
parallelMap::parallelStartMulticore(level = 'mlr.resample')
lrn <- makeLearner("classif.lda", predict.type = "prob")
results =  lrn %>%  resample(task = sonar.task,
                   resampling = cv3,
                   measures = list(acc, setAggregation(acc, test.sd)))
parallelMap::parallelStop()
```

Krzywa ROC

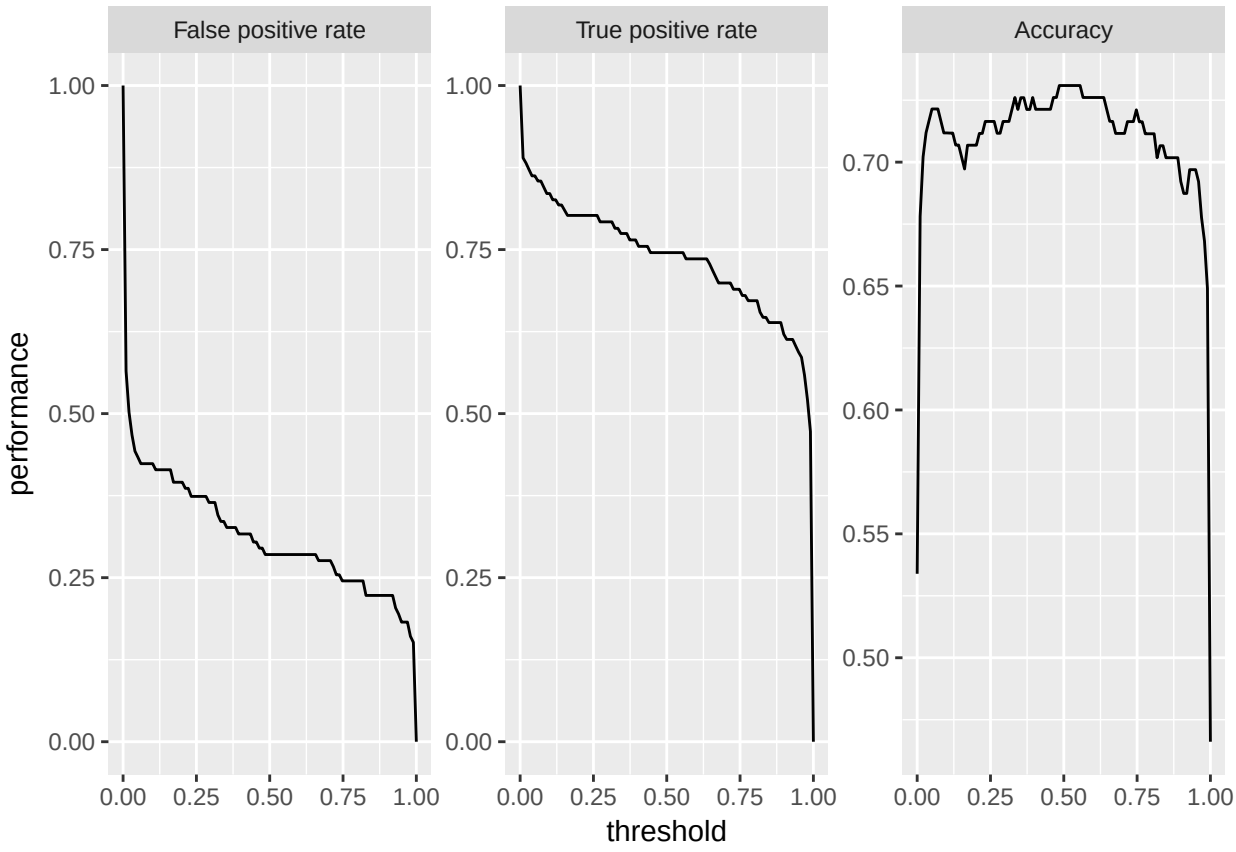
```
df = generateThreshVsPerfData(results, measures = list(fpr, tpr, acc))

plotROCCurves(df) + theme_bw()
```



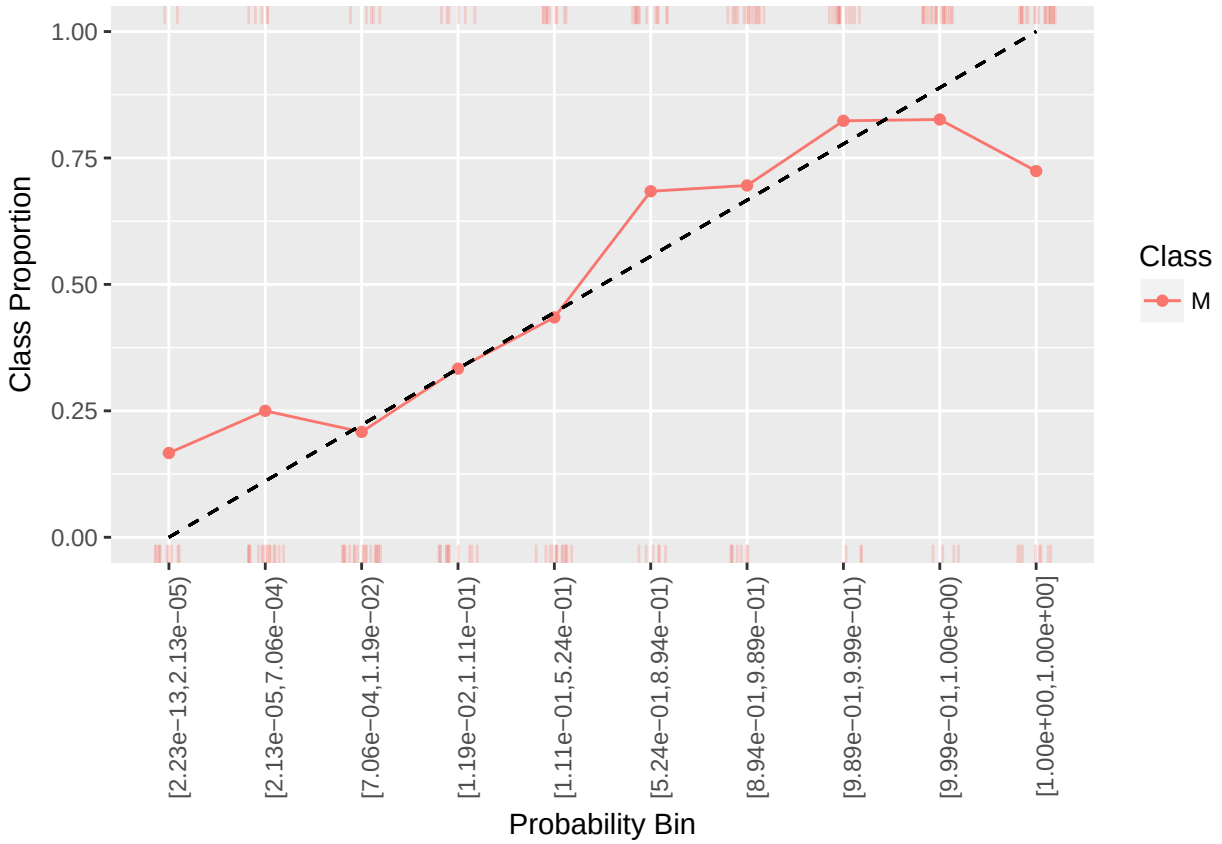
Miary jakości, a wartość progu

```
plotThreshVsPerf(df)
```



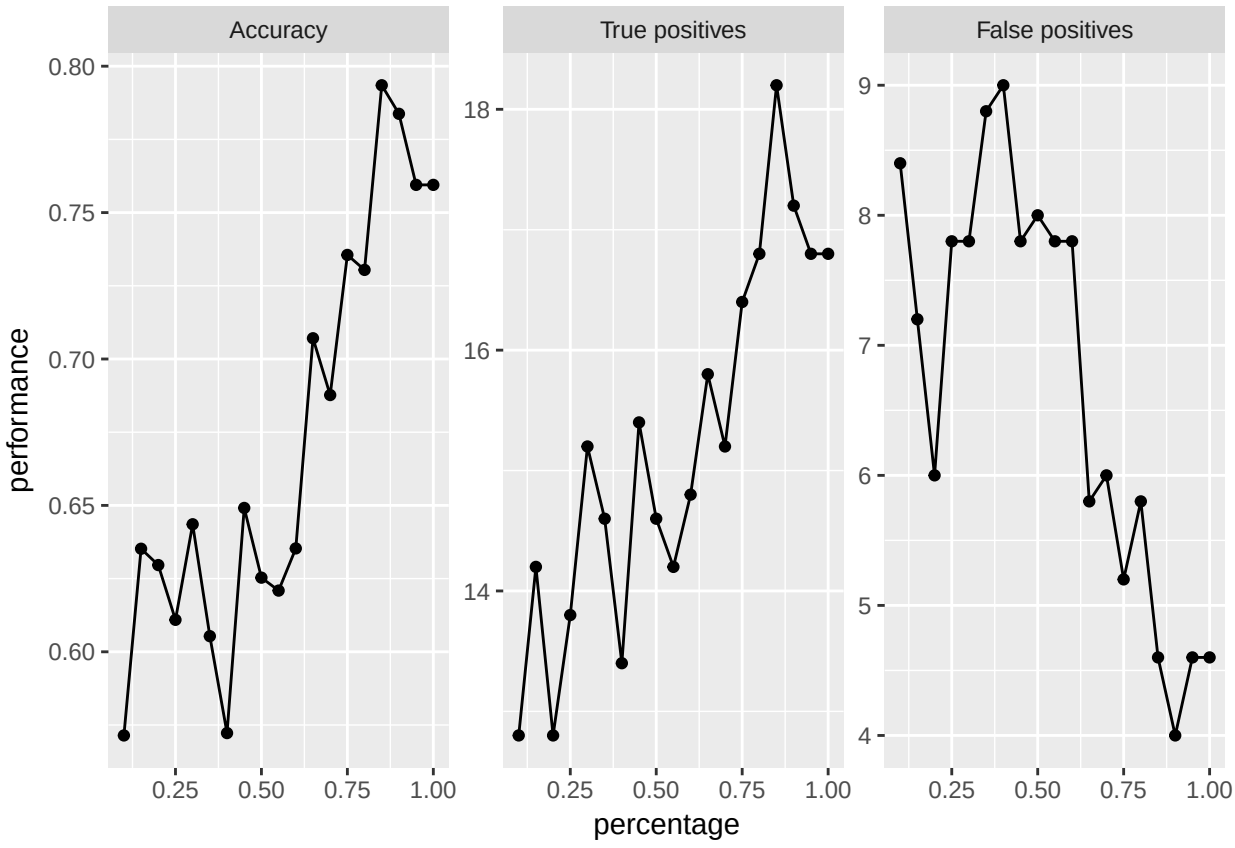
Kalibracja klasyfikatora

```
cal = generateCalibrationData(results, groups = 10)
plotCalibration(cal)
```



Krzywe uczenia

```
r = generateLearningCurveData(
  learners = lrn,
  task = sonar.task,
  percs = seq(0.1, 1, by = 0.05),
  measures = list(acc, tp, fp),
  resampling = makeResampleDesc(method = "CV", iters = 5),
  show.info = FALSE)
plotLearningCurve(r)
```


Accuracy
True positives
False positives
performance
percentage
0.80
0.75
0.70
0.65
0.60
18
16
14
9
8
7
6
5
4
0.25
0.50
0.75
1.00