Sick dataset analysis - blackbox

Michał Pastuszka

28 04 2020

Loading dataset

Preprocessing

```
train_indices <- read.csv('indeksy_treningowe.txt', sep = ' ', row.names = 1, header = TRUE)
dataset_train <- dataset[train_indices$x,]
dataset_test <- dataset[-train_indices$x,]
```

```
# variable hypopituary contains only one value in training set - we have to drop it
dataset_train <- dataset_train %>%
  select(-hypopituitary)
dataset_test <- dataset_test %>%
  select(-hypopituitary)
```

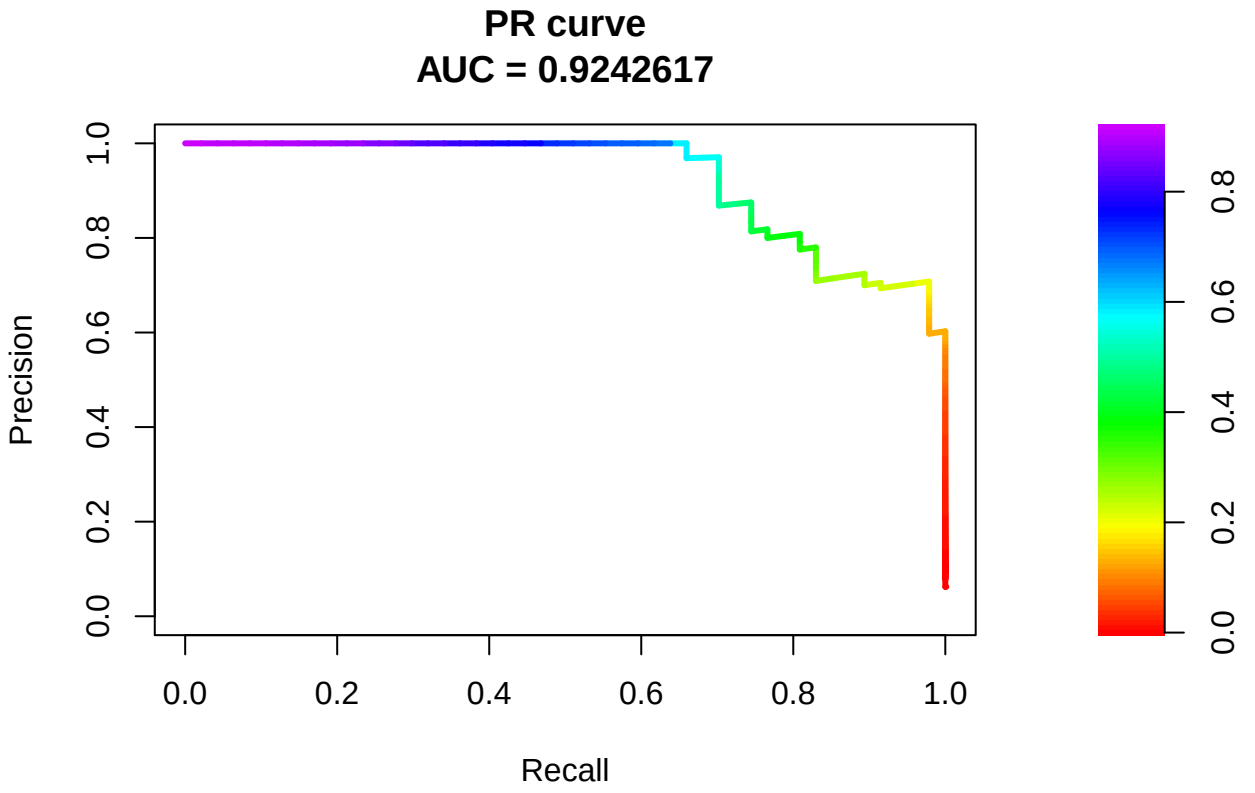
```
dataset_train$sex <- dataset_train$sex %>% replace_na('F')
dataset_test$sex <- dataset_test$sex %>% replace_na('F')
dataset_test$age <- dataset_test$age %>% replace_na(mean(dataset_train$age, na.rm = TRUE))
centerer_bag <- preProcess(dataset_train, method = c('bagImpute'))
dataset_train_bag <- predict(centerer_bag, dataset_train)
dataset_test_bag <- predict(centerer_bag, dataset_test)
```

Blackbox model

We will fit a Ranger random forest using default parameters.

```
get_prauc <- function(responses, test_set){
  fg <- responses[test_set$Class == 'sick']
  bg <- responses[test_set$Class == 'negative']
  pr <- pr.curve(scores.class0 = fg, scores.class1 = bg, curve = T)
  plot(pr)
}
set.seed(10, "L'Ecuyer")
task <- makeClassifTask("ranger", data=dataset_train_bag, target = "Class")
learner_ranger <- makeLearner("classif.ranger", predict.type = 'prob')
model_ranger <- mlr::train(learner_ranger, task)
pred_ranger <- predict(model_ranger, newdata = dataset_test_bag)
probs <- pred_ranger$data$prob.sick

get_prauc(probs, dataset_test_bag)
```

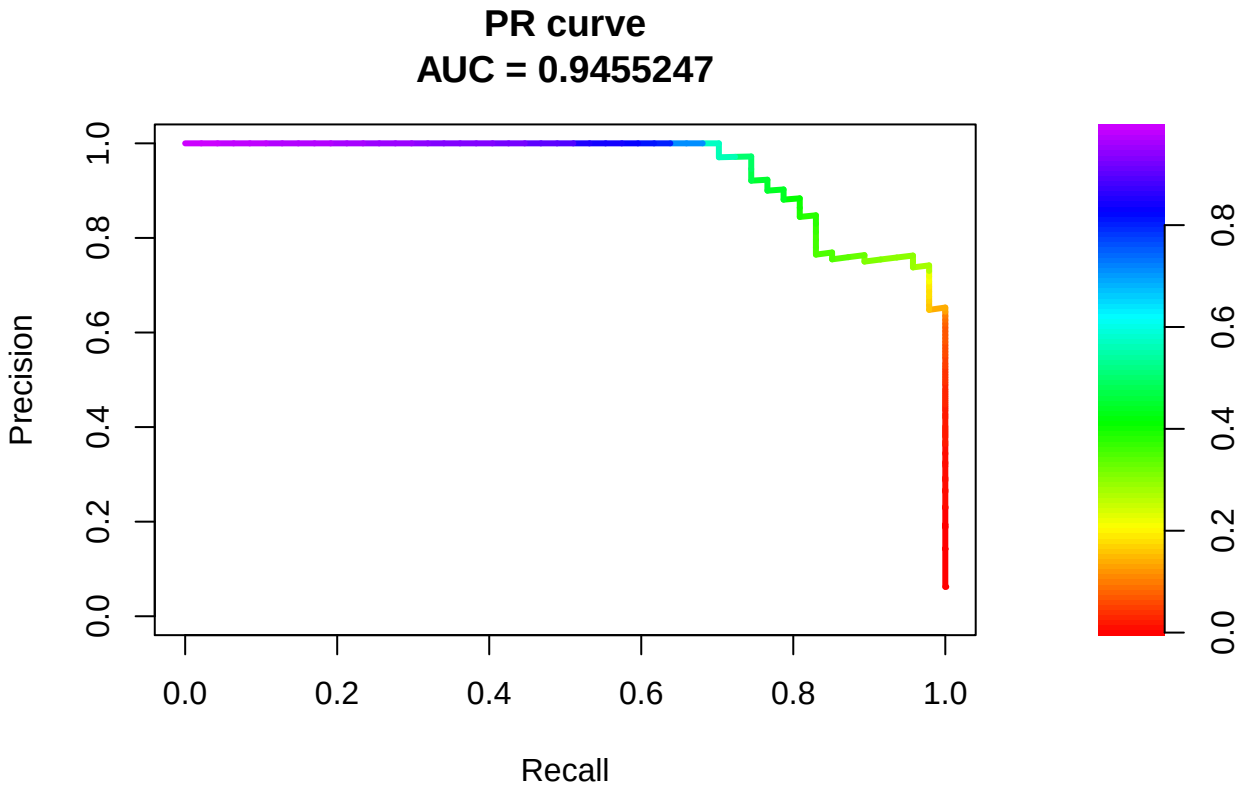


It has completely outclassed the previous attempt using logistic regression. We Now we'll try to improve the performance even more tuning the hyperparamters. We will use 5 fold crossvalidation on the training set and mlrMBO optimizer to tune mtry and min.node.size parameters.

```
task <- makeClassifTask("ranger", data=dataset_train_bag, target = "Class")
learner_ranger <- makeLearner("classif.ranger", predict.type = 'prob')
ps <- makeParamSet(
  makeIntegerParam("mtry", lower = 1, upper = 26),
  makeIntegerParam("min.node.size", lower = 1, upper = 100)
)
ctrl <- makeTuneControlMBO()
rdesc <- makeResampleDesc("CV", iters = 5L, stratify = TRUE)
res <- tuneParams(learner_ranger, task, rdesc, auc, ps, ctrl, show.info = FALSE)
print(res)
```

```
## Tune result:
## Op. pars: mtry=9; min.node.size=1
## auc.test.mean=0.9962700
```

```
learner_ranger_tune <- makeLearner("classif.ranger", predict.type = 'prob', par.vals = res$x)
model_ranger_tune <- mlr::train(learner_ranger_tune, task)
pred_ranger_tune <- predict(model_ranger_tune, newdata = dataset_test_bag)
probs <- pred_ranger_tune$data$prob.sick
get_prauc(probs, dataset_test_bag)
```



It scored even better than previously. It appears that using a blackbox model can drastically improve performance and reduce required work by sacrificing explainability and simplicity of the solution. It is worth noting, that a well tuned decision tree on this set can achieve similar results, while retaining explainability.