

An EDA and a white box classification of an OpenML sick dataset

Karol Pysiak

28.04.2020

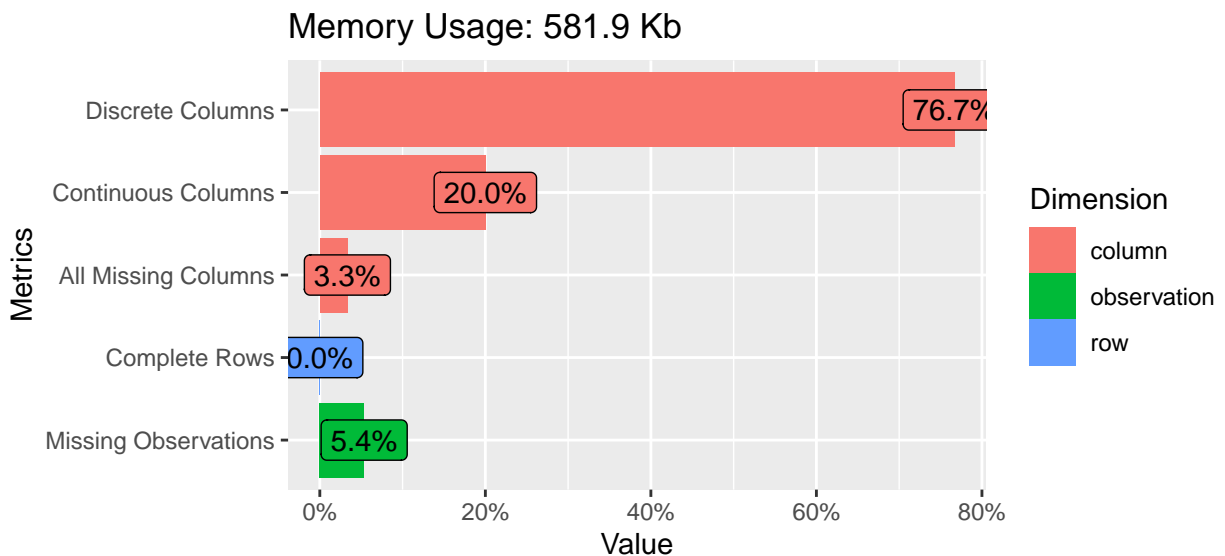
Introduction

In this exercise we will be working on a “sick” dataset from the OpenML database. What will be presented there is an Exploratory Data Analysis and an attempt to predict some classes on a test data with an explainable model, which will be introduced later.

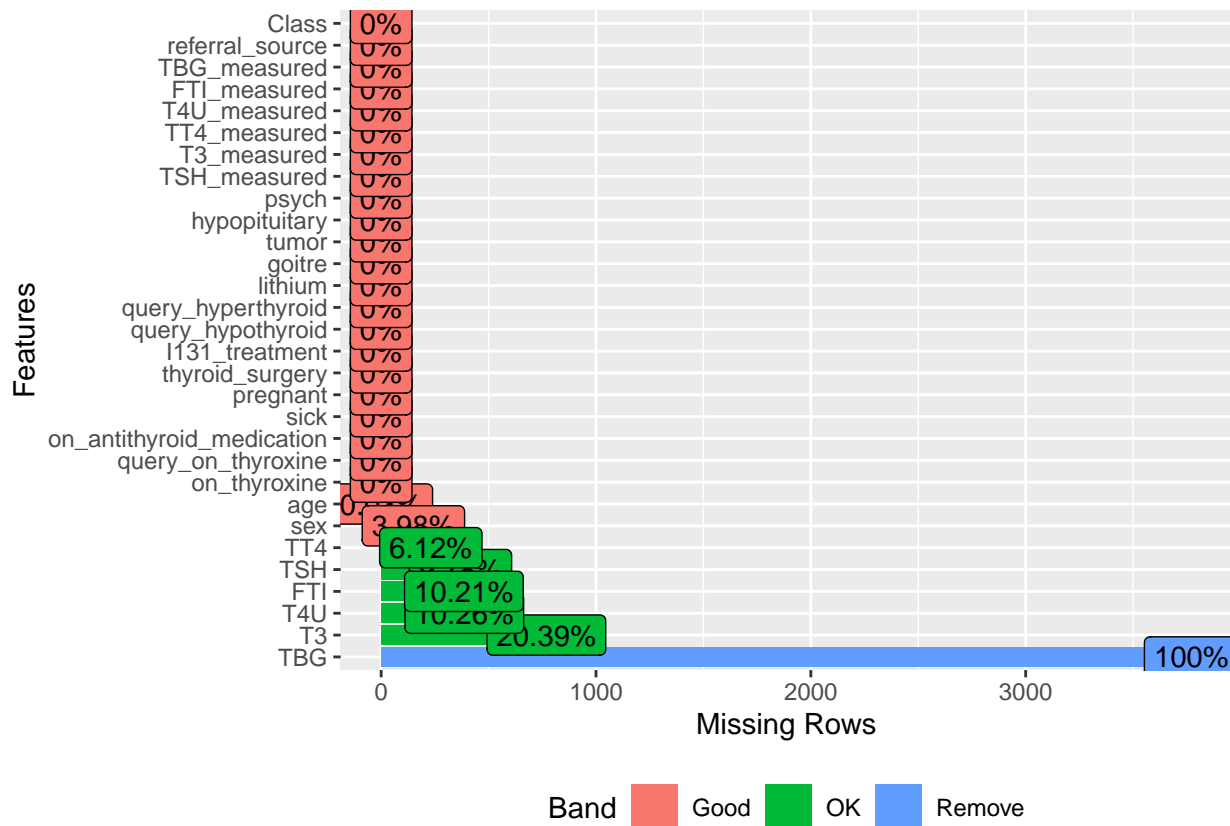
Exploratory Data Analysis

Insight into structure of data

Firstly, we want to check how the dataset is built. We will show if there are missing values, what types are the variables of and how variables are distributed.

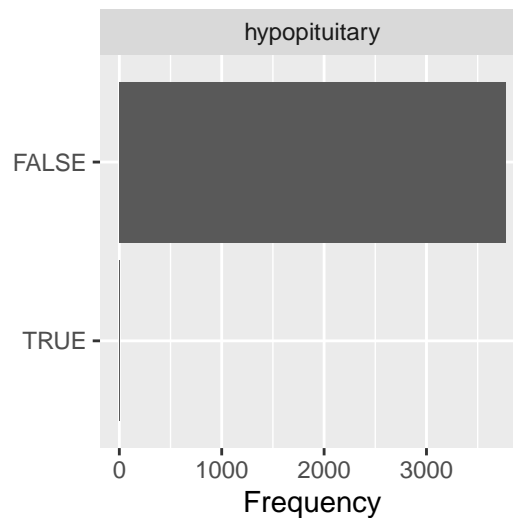


The one thing that can bother us when looking on this graph is that we have some fraction of variables completely missing. The percentage of missing values in total is not that bad. Let's check how this missing values look grouping by variables.



As we have seen above there is one variable completely missing and it is the TBG. We will delete it from dataset, because it is totally insignificant in this situation. The rest of variables have a quite small fraction of missing values, so for the sake of simplicity we will delete rows that contain missing values.

One more thing that can very much affect our classification is that the **hypopituitary** variable, which is a logical variable, has close to 0 occurrences of a **TRUE** value, so it will bring no benefit to the model.



Some of variables are just flags if the other variable contains an **NA** value in that observation. This can give us very little information so we will omit these variables either.

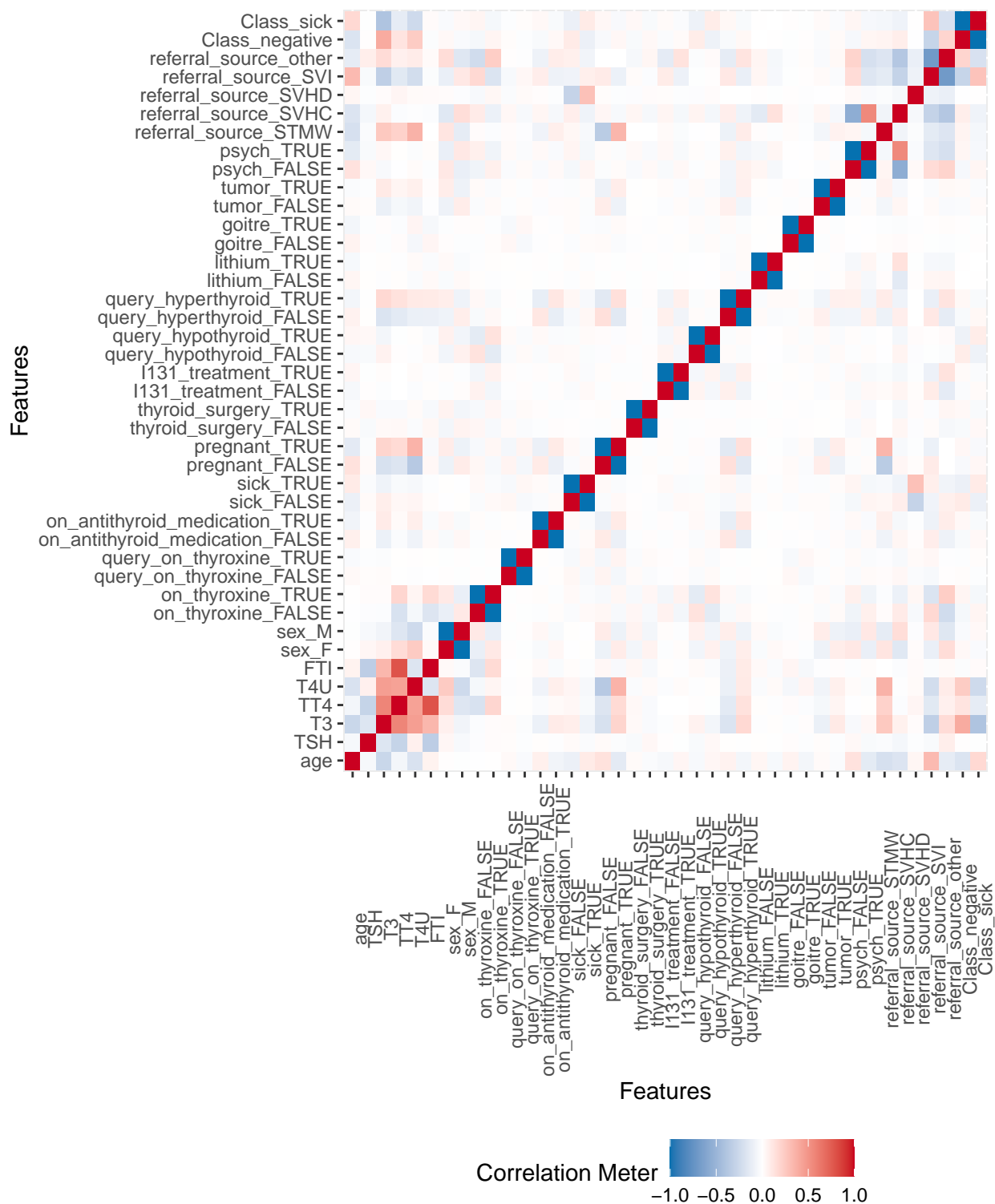
So after all these observations we have a dataset that look like this.

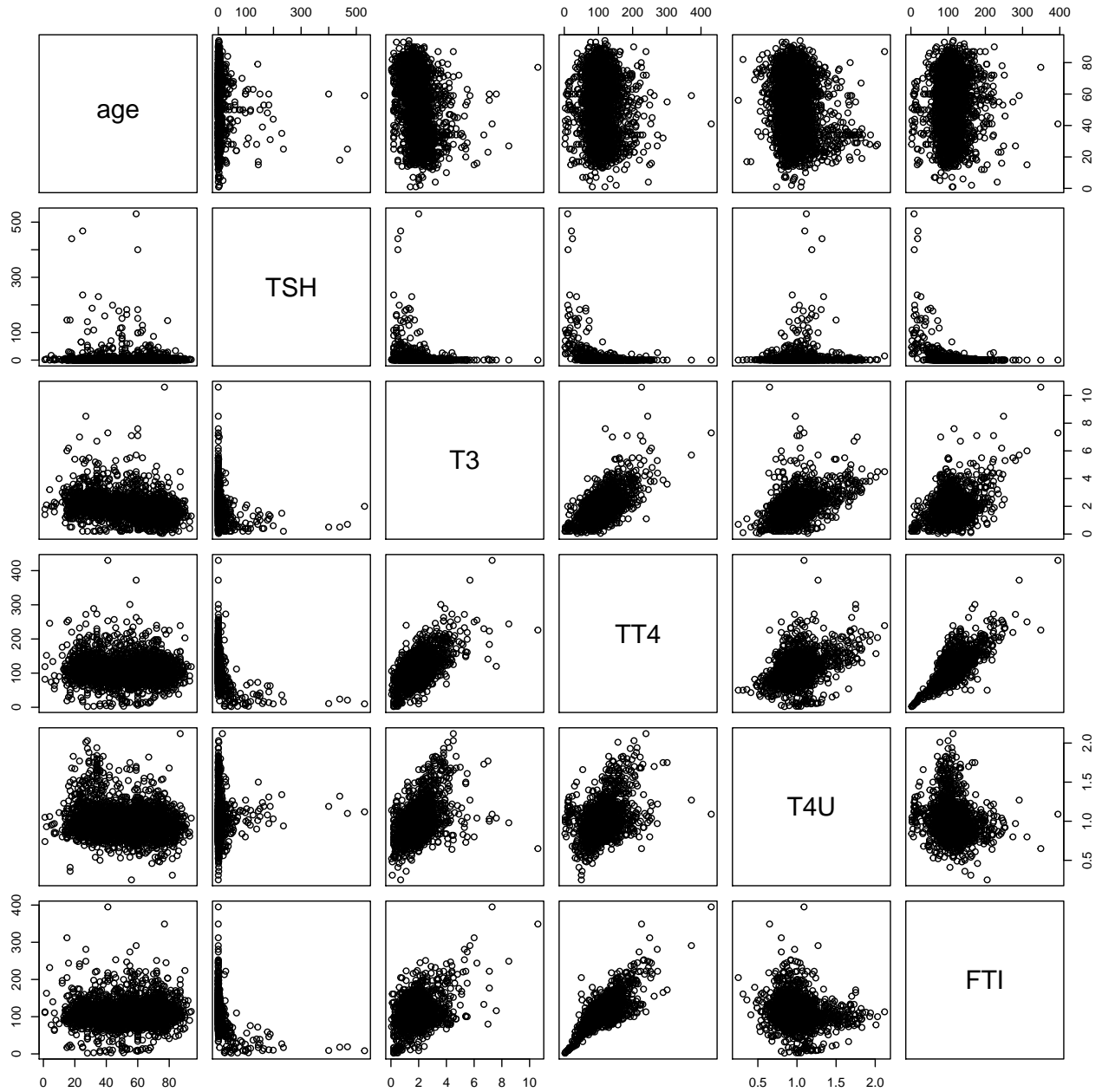
```
## # A tibble: 2,643 x 22
##   age sex  on_thyroxine query_on_thyrox~ on_antithyroid_~ sick  pregnant
##   <dbl> <chr> <lgl>         <lgl>         <lgl>         <lgl> <lgl>
## 1    41 F      FALSE         FALSE         FALSE         FALSE FALSE
## 2    70 F      FALSE         FALSE         FALSE         FALSE FALSE
## 3    80 F      FALSE         FALSE         FALSE         FALSE FALSE
## 4    66 F      FALSE         FALSE         FALSE         FALSE FALSE
## 5    68 M      FALSE         FALSE         FALSE         FALSE FALSE
## 6    84 F      FALSE         FALSE         FALSE         FALSE FALSE
## 7    71 F      FALSE         FALSE         FALSE         TRUE  FALSE
## 8    59 F      FALSE         FALSE         FALSE         FALSE FALSE
## 9    28 M      FALSE         FALSE         FALSE         FALSE FALSE
## 10   63 F      FALSE         FALSE         FALSE         FALSE FALSE
## # ... with 2,633 more rows, and 15 more variables: thyroid_surgery <lgl>,
## #   I131_treatment <lgl>, query_hypothyroid <lgl>, query_hyperthyroid <lgl>,
## #   lithium <lgl>, goitre <lgl>, tumor <lgl>, psych <lgl>, TSH <dbl>, T3 <dbl>,
## #   TT4 <dbl>, T4U <dbl>, FTI <dbl>, referral_source <chr>, Class <chr>
```

While processing the data it turned out that one of observations has invalid **age** of 455. It was 45 or 55 probably, but there is no need to choose it at random, because it is just a one observation, so we will just remove it from our dataset. Below we can see a part of the table of frequencies of ceratain ages of observations.

age	Freq
85	14
86	4
87	11
88	6
89	7
90	5
91	1
92	1
93	2
94	1
455	1

Now we will check the dependencies between variables.





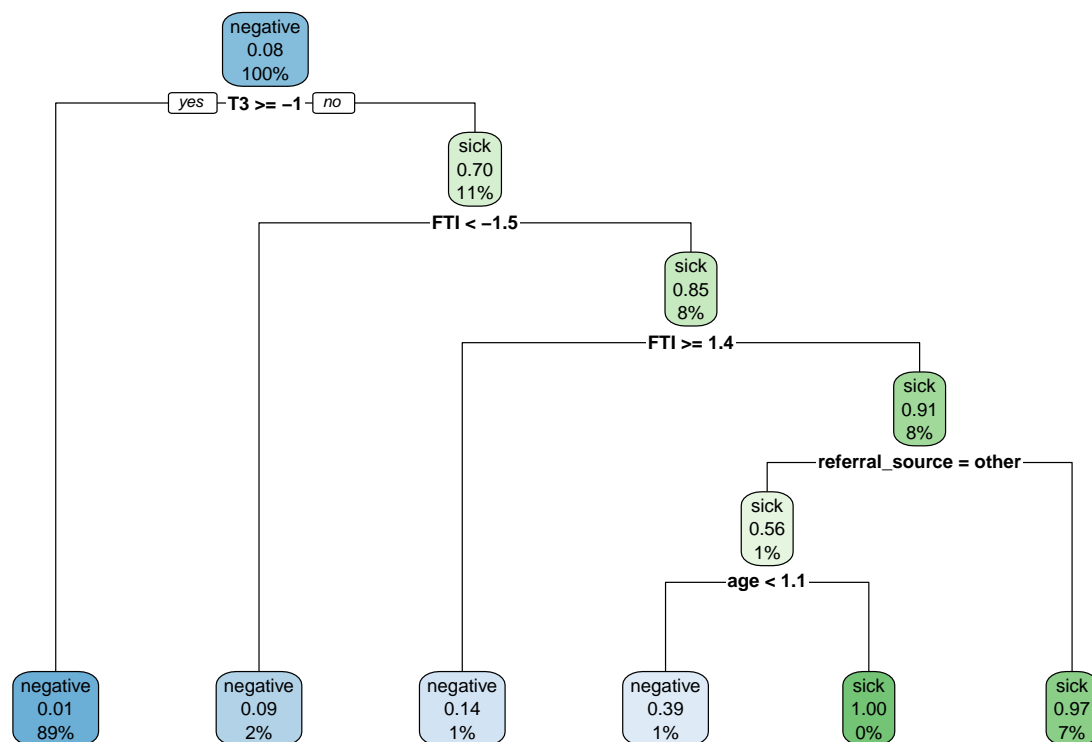
Variables T3, TT4, T4U, FTI looks somehow correlated, especially TT4 and FTI, but not in a way that we should worry about.

White box models testing

Firstly we just standardize our data and split it into the training and testing data. In the training data, as it was mentioned before, we just remove rows that contain missing data. With the testing data we cannot do that because it could affect our end score, which we want to be as close to the reality as possible. We will replace the missing values in the testing data with a mean for numerical variables and a mode for categorical variables.

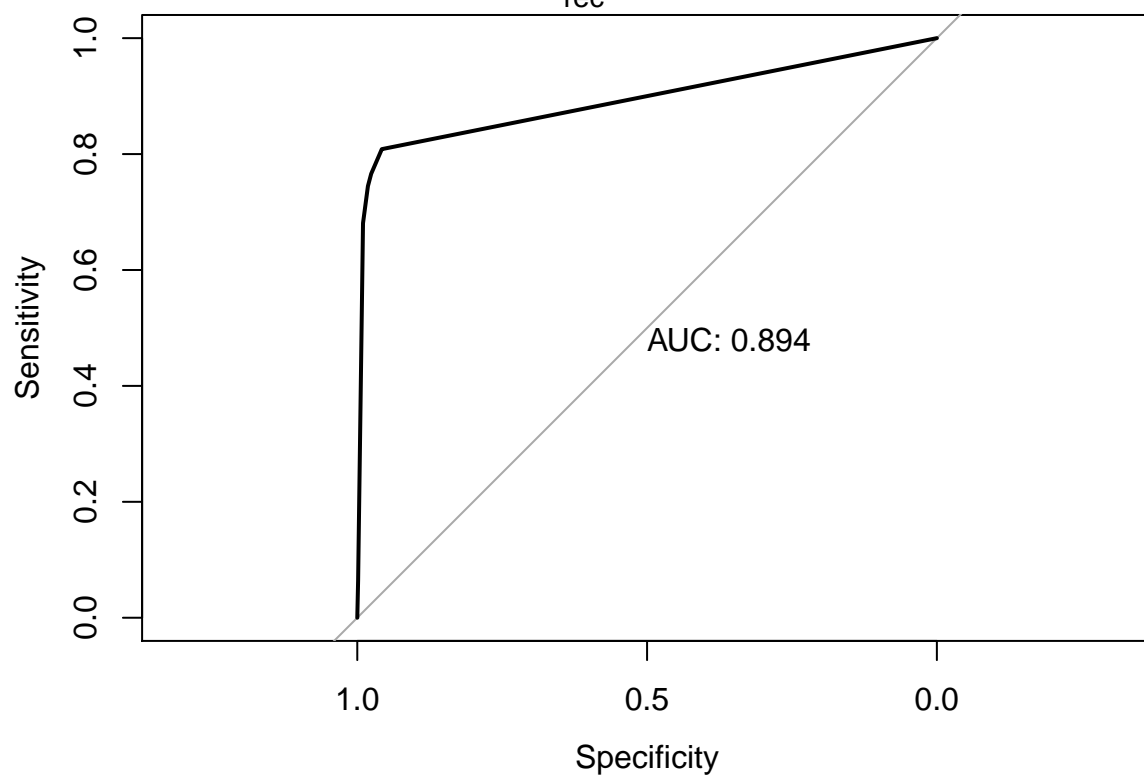
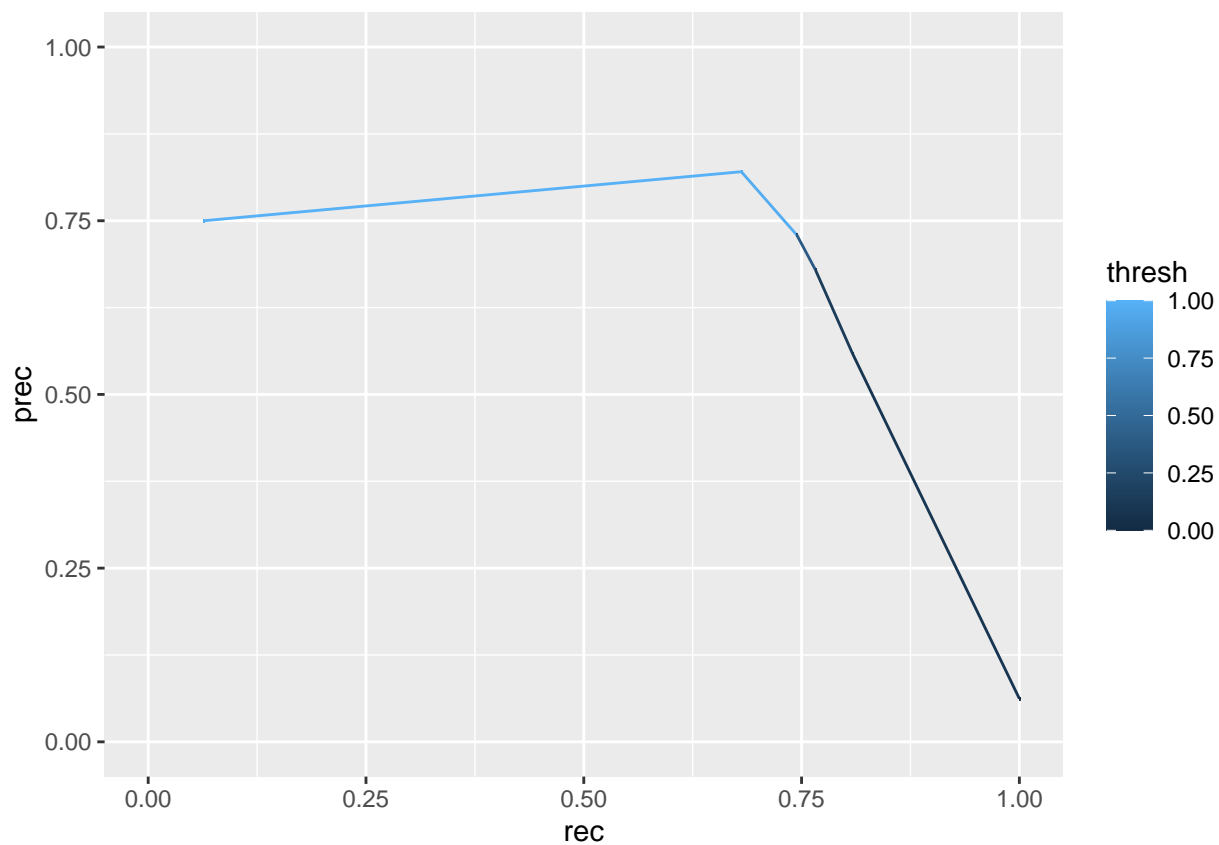
Decision Tree

The first interpretable model that we will test here is Decision Tree. It is probably one of the easiest to describe and draw model. There we will use the `rpart` algorithm from the `mlr` package. After training it on our training data we have a tree that looks like this.



As we can see, model decides which way to go based on value of certain variables. Thresholds are tuned during the training process.

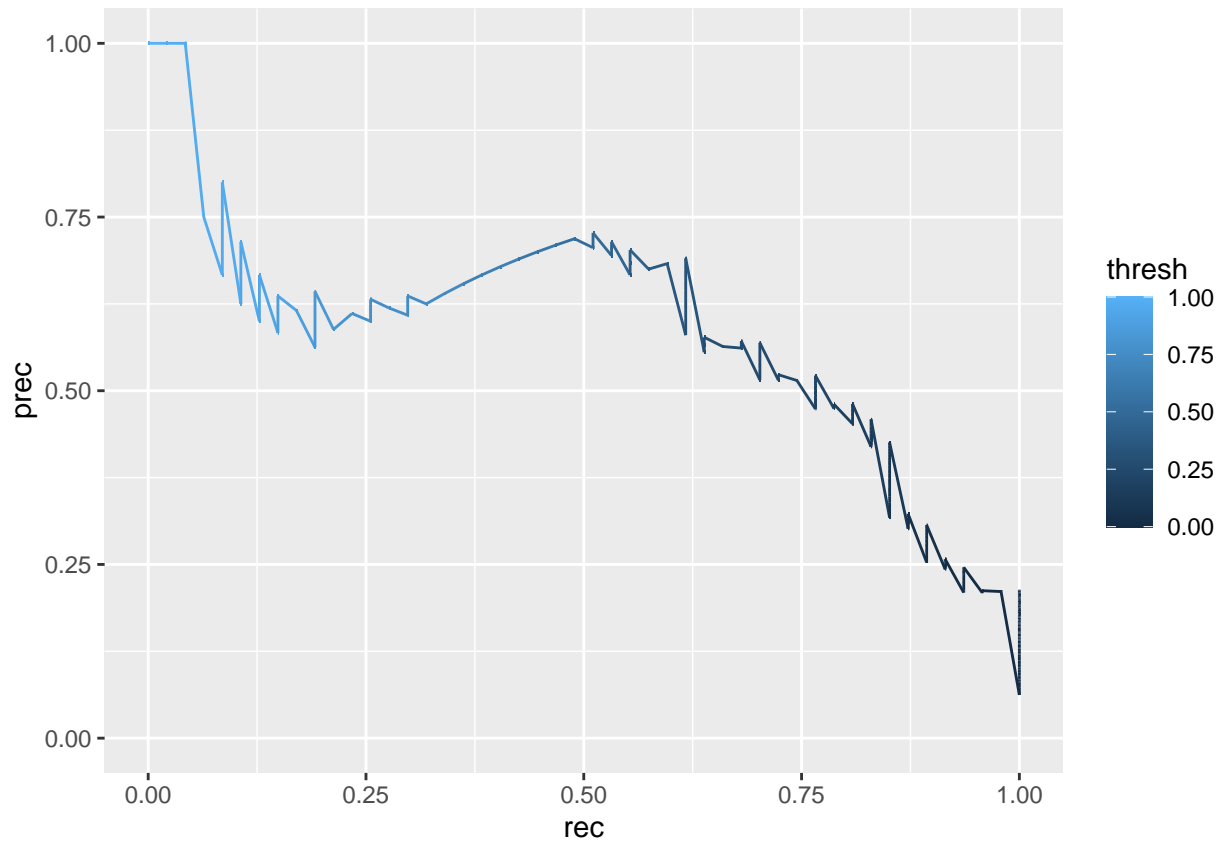
```
## [1] "AUPRC: 0.650825275451872"
```

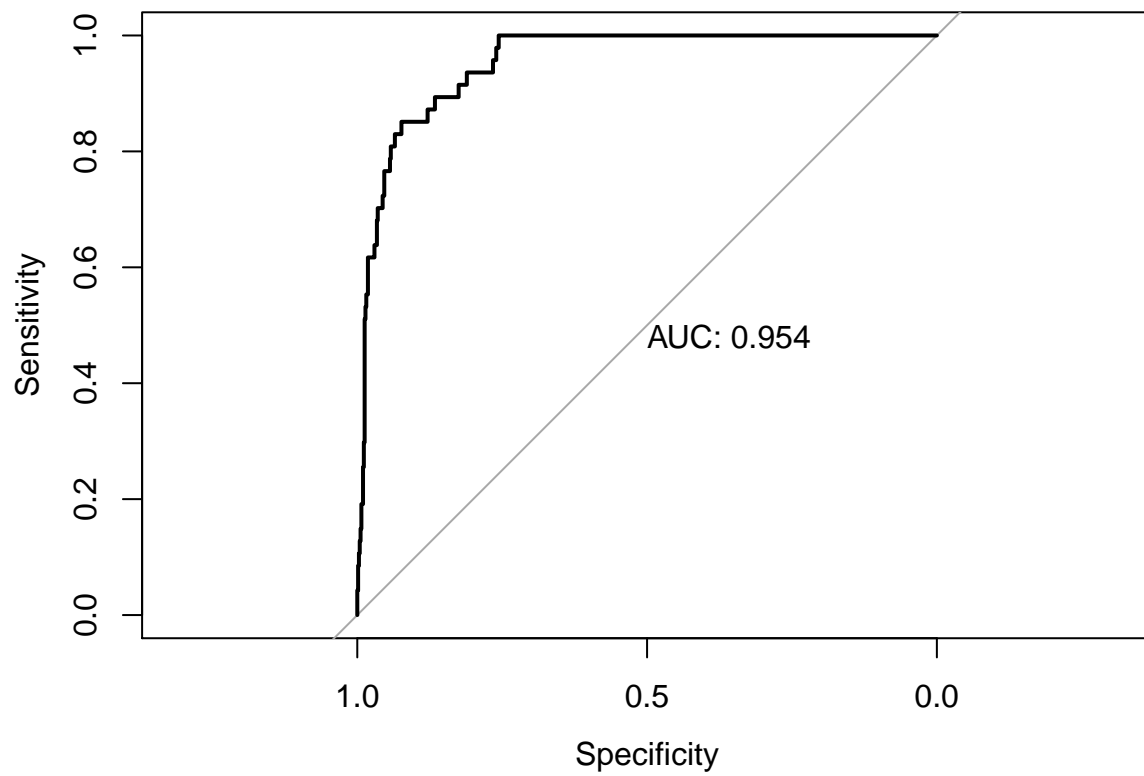


Logistic Regression

Logistic Regression is easy to explain. It is like Linear Regression of classification.

```
## [1] "AUPRC: 0.561059652256059"
```



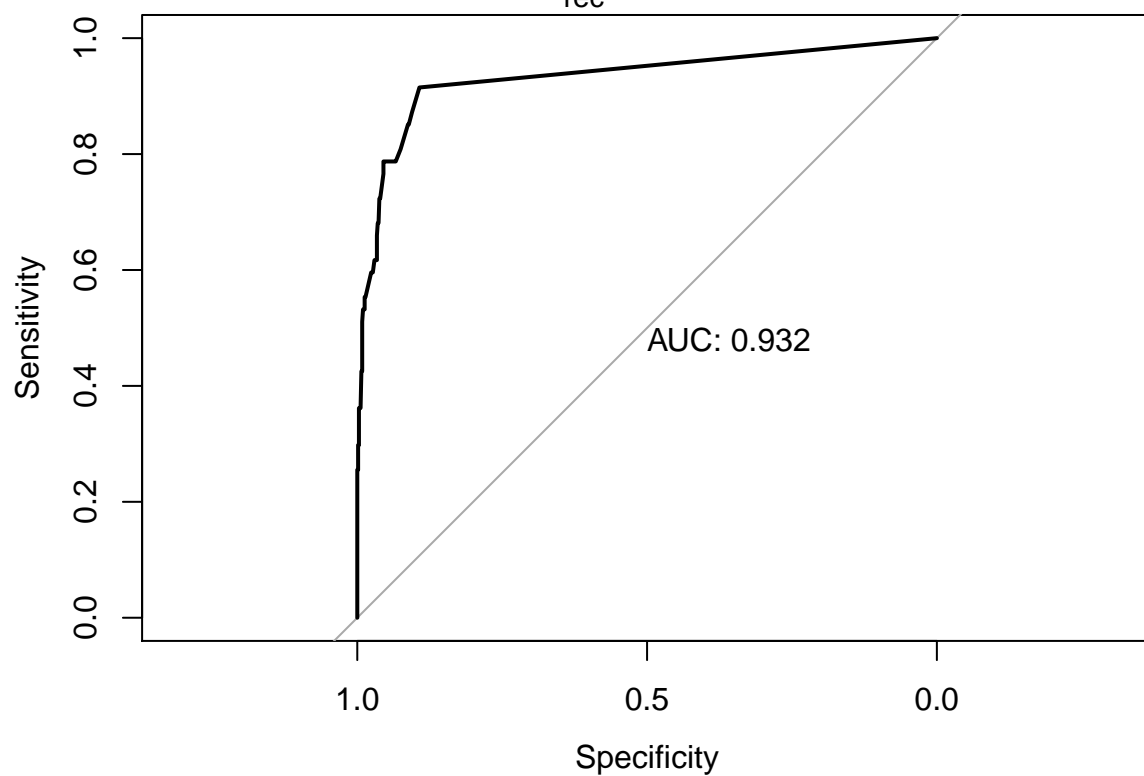
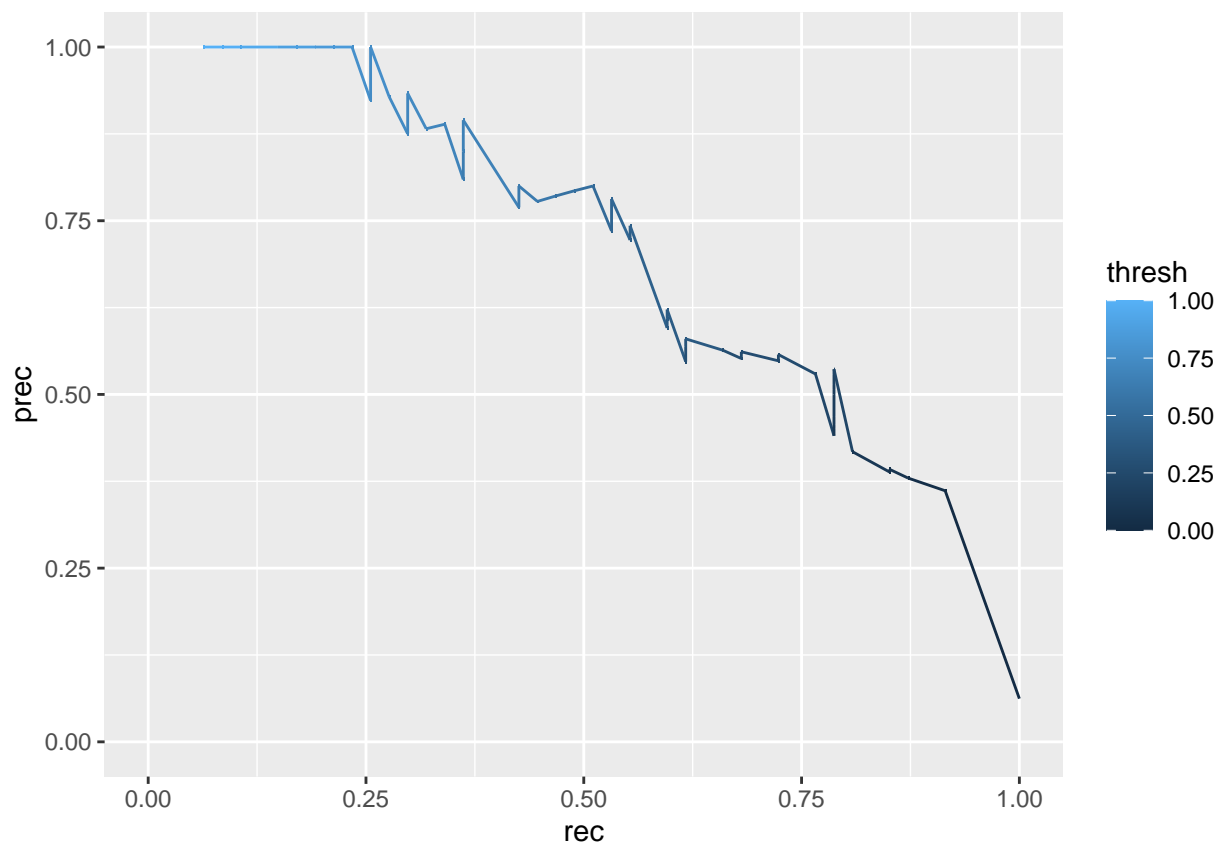


kNN

kNN model is based on how observations are distributed in space. We just look what number of k closest neighbors belong to each class.

Standard tests

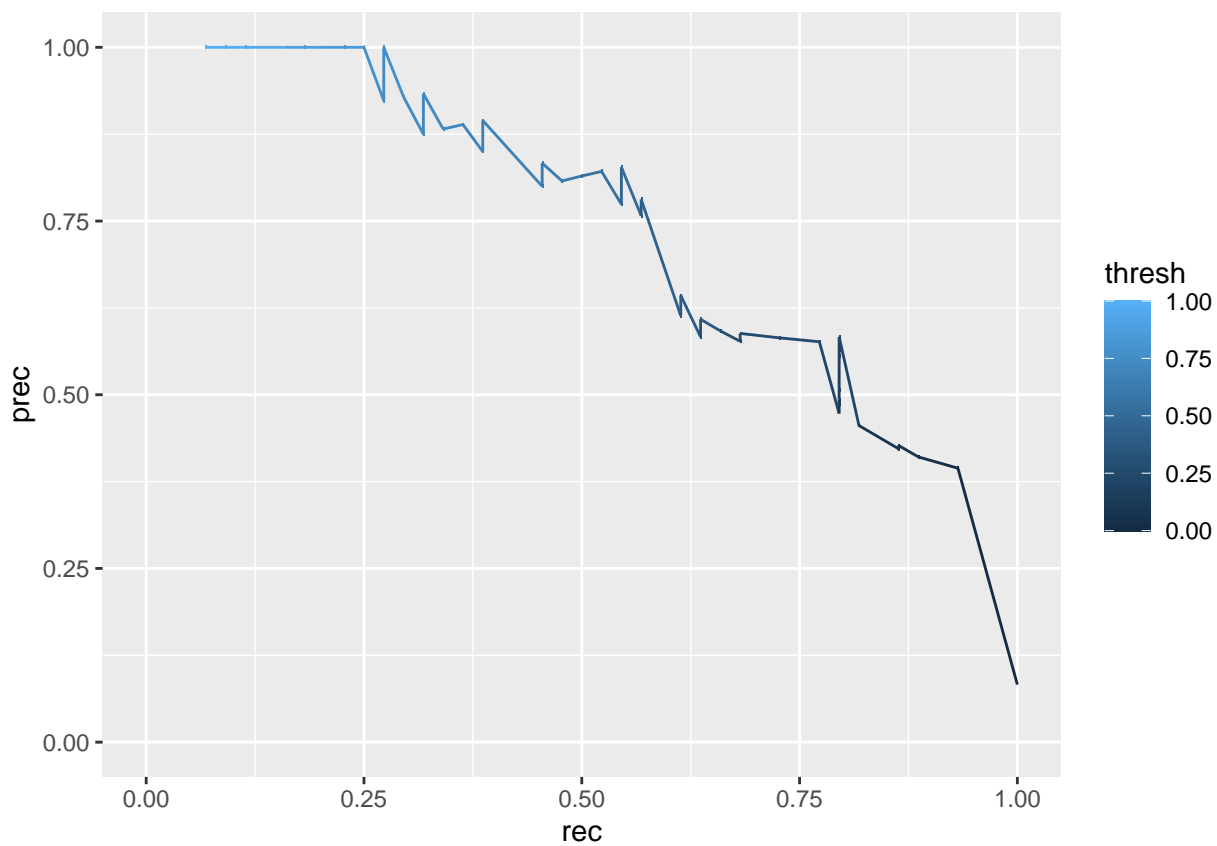
```
## [1] "AUPRC: 0.676345223465442"
```

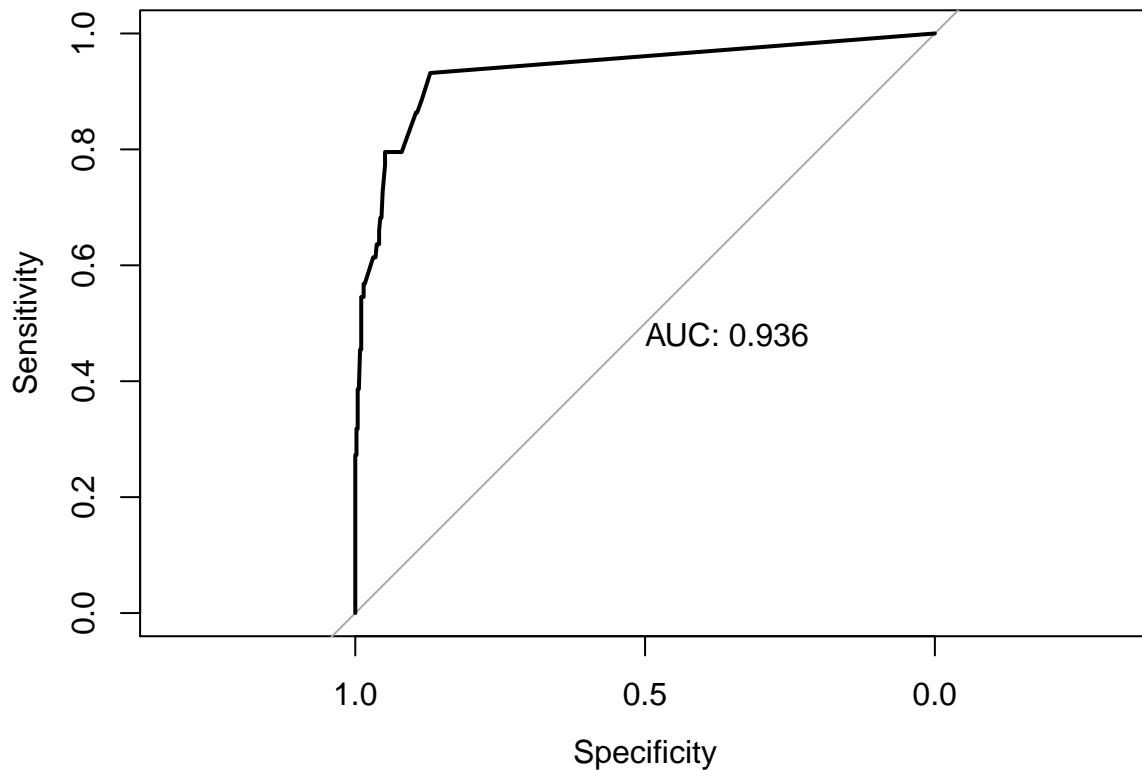


Tests with removed rows with missing values

I just was curious how removing rows of testing data, instead of imputing them, will affect our scores.

```
## [1] "AUPRC: 0.70999610989033"
```





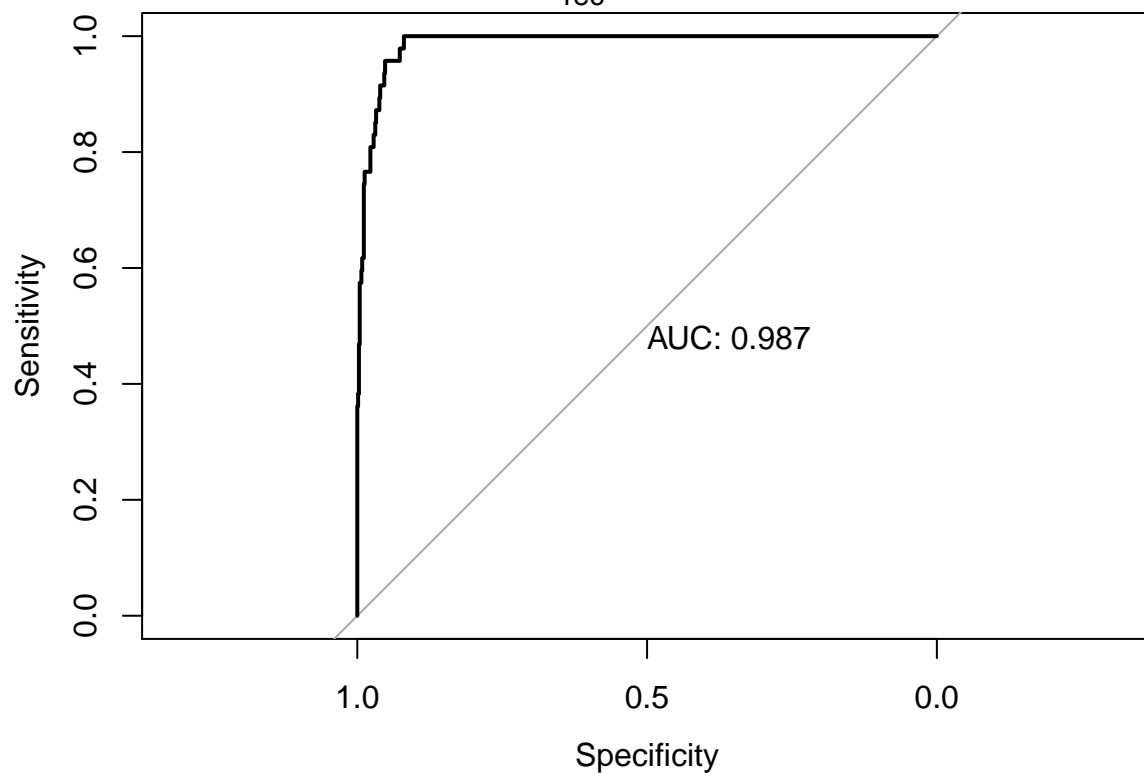
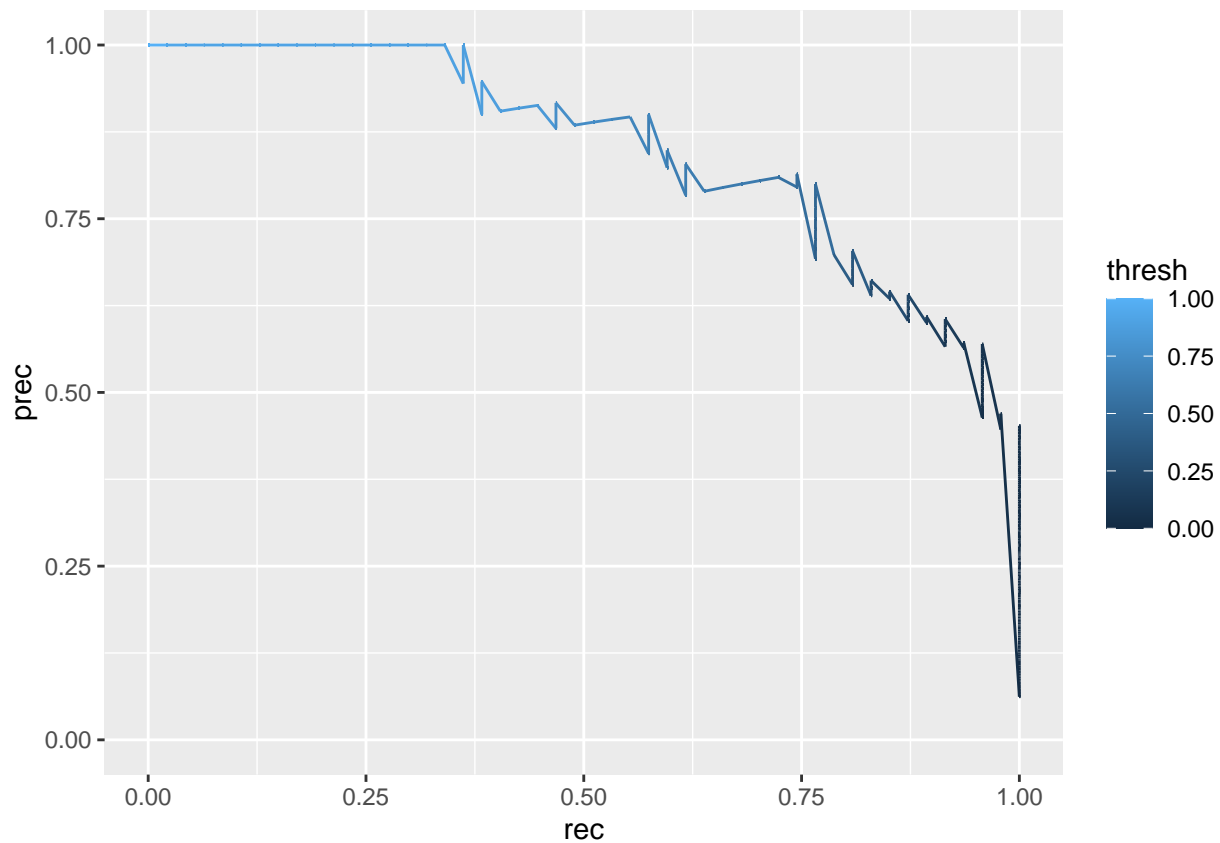
Although it was kind of predictable I find this experiment very informative, because sometimes we want to make something quickly and easily, but we end up with messed up scores and results. There we got better scores than we should and in a real life task it could have devastating effects on our project.

Black box model testing

After testing few white box models we have some knowledge of what we can achieve using those models. Now we will try out a black box model. For this task I chose the `bartMachine` learner from `mlr` package.

```
## bartMachine initializing with 50 trees...
## bartMachine vars checked...
## bartMachine java init...
## bartMachine factors created...
## bartMachine before preprocess...
## bartMachine after preprocess... 40 total features...
## bartMachine sigsq estimated...
## bartMachine training data finalized...
## Now building bartMachine for classification ...Covariate importance prior ON. Missing data feature ON
## evaluating in sample data...done

## [1] "AUPRC: 0.827338280005495"
```



Unfortunately for our white box classifiers `bartMachine` taken just from the shelf reached AUC of 0.99 and AUPRC of 0.83. We got there much better results than in the test with the white box classifiers. However, in this moment we have a model that we do not understand and which we tune only randomly or by basing

on a gut sense.

Summary

It is beyond doubt that some problems require really complicated solutions, but in many cases we can use much simpler and much more interpretable models. This case is an example that even with models like logistic regression or decision tree we can get pretty good results. What is more, when we know how exact model works we can tune it ourselves or augment our data a bit to boost the efficiency of a model. As we can see above, black box classifier had a better performance than the white box analogues. It will be hard to get better scores on the white box models than on the black box ones, because the requirement of low model complexity in the white box models limits in some way the level of a fitting to a complex data. Keeping that in mind we can still use the white box models for less complex data or for the real life solutions that need to be explained. One last thing is very interesting. When we take a look at **AUC** and **AUPRC** measures of the white box models we can see that **AUC** is almost the same for all of the cases, when **AUPRC** differ significantly.

Potwierdzam samodzielność powyższej pracy oraz niekorzystanie przeze mnie z niedozwolonych źródeł. Karol Pysiak