# Raport WB 2-1

## Comparison of deep learning and tree-based models in the clinical prediction of the course of COVID-19 illness

*Kacper Grzymkowski, Jakub Fołtyn, Konrad Komisarczyk*

## EDA

We are using data from (Xiaoran et al. 2020) article. The data have already been preprocessed: records with insufficient test results have been removed and other missing data have been imputed with predictive mean modeling using the Multivariate Imputation.n The authors have also used Boruta software to pick the most important features from the dataset. As a result, 5 features were selected from the ICU admission dataset and 6 from the death prediction dataset. It is worth mentioning that an independent Boruta test conducted by us has selected several additional features for each dataset.

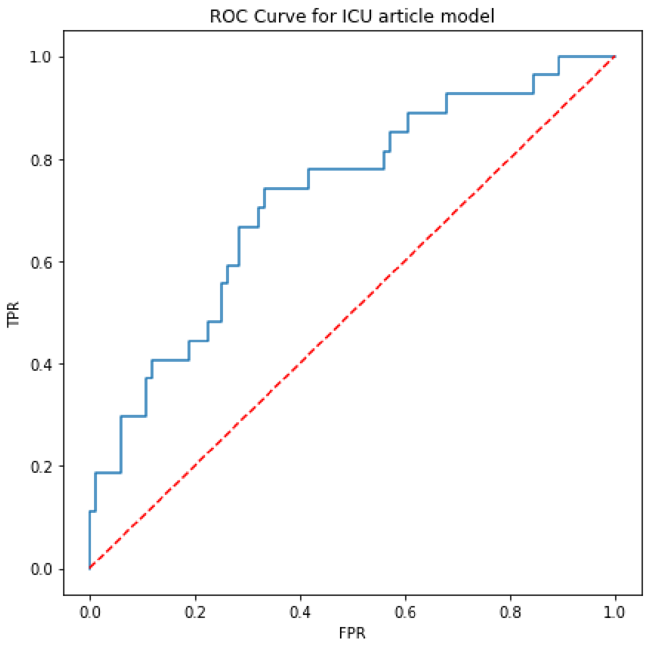
## Reproduction of the DNN model from the article

We followed the reproduction steps described in the article (Xiaoran Li et al. 2020) using the Keras Python library, with two modifications:

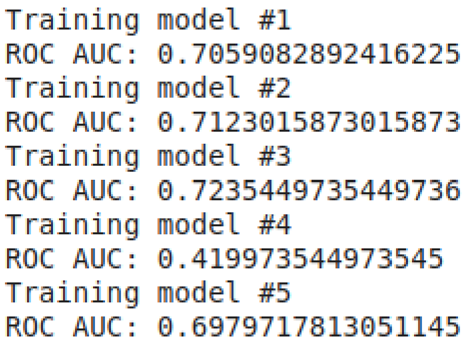
- We scaled the input data using sklearn's RobustScaler
- We skipped the 5-Fold cross validation and weight aggregation, instead using 0.1 validation set size and the resulting training set was used in an automatic 0.2 split for testing.

### ICU admission model

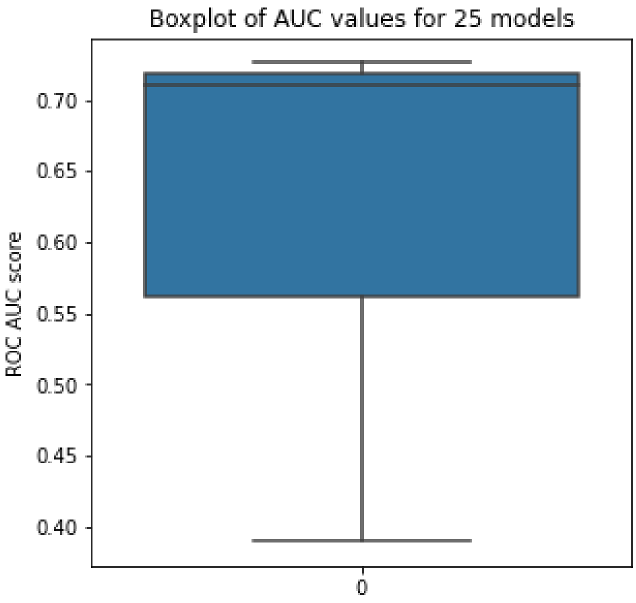
The resulting ROC Curve of the model was as follows, with a AUC of around 0.722. The results were not as good as the ones in the article (AUC=0.780), we suspect that this worse performance is caused by skipping the 5-Fold cross validation and aggregation step.



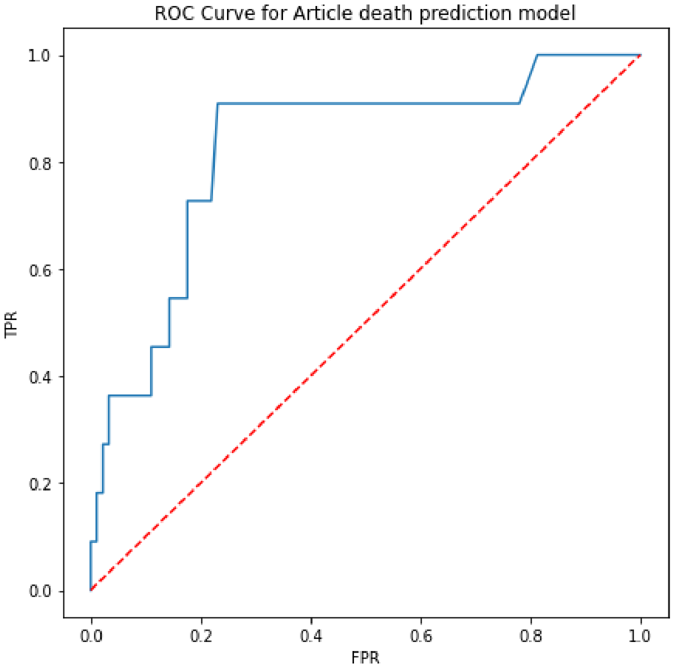
However, more worryingly, the results from the model were quite inconsistent. We noticed that sometimes the model would not converge at all, and stay around a AUC = 0.5, while other times even performing worse than a random classifier with AUCs as low as 0.3. We later determined the cause of this behavior to be the loss function that was chosen in the article. Mean square error is a metric usually used for regression tasks and proved inadequate here. Moreover, it’s common to use categorical cross entropy as a loss function for a classification task



```
Training model #1
ROC AUC: 0.7059082892416225
Training model #2
ROC AUC: 0.7123015873015873
Training model #3
ROC AUC: 0.7235449735449736
Training model #4
ROC AUC: 0.419973544973545
Training model #5
ROC AUC: 0.6979717813051145
```



## Death prediction model



The death prediction model performed better and didn’t have convergence issues like the ICU model. With an AUC score of 0.825, which we decided was close enough to article value of 0.844.

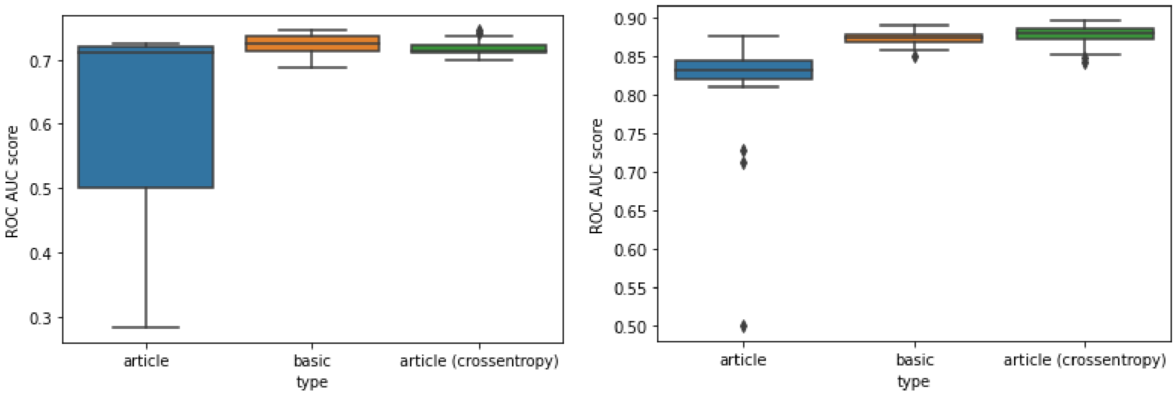
# Our modifications to the DNN models

The convergence issue with the ICU admission prediction model sparked us to experiment with new models, and we created two new architectures:

- Basic: a densely connected, two hidden layer network with 32 neurons per layer.
- Cross entropy: copy of the article model, with binary cross entropy as the loss function instead of mean square error.

The results were a clear increase in model stability - with the basic model and the modified article model showing similar results. Both were better than the original model in stability and mean performance.

*Left is ICU models, right is death prediction models.*



# Tree-based models on the new data

We trained tree-based models using two datasets for both icu admission and death prediction: features selected with Boruta algorithm and all features shared by the article authors. Sets were splitted randomly into train and test fraction in ratio 0.75:0.25.
We decided to use following algorithms:

- gradient boosting from R xgboost package
- random forest from R ranger package

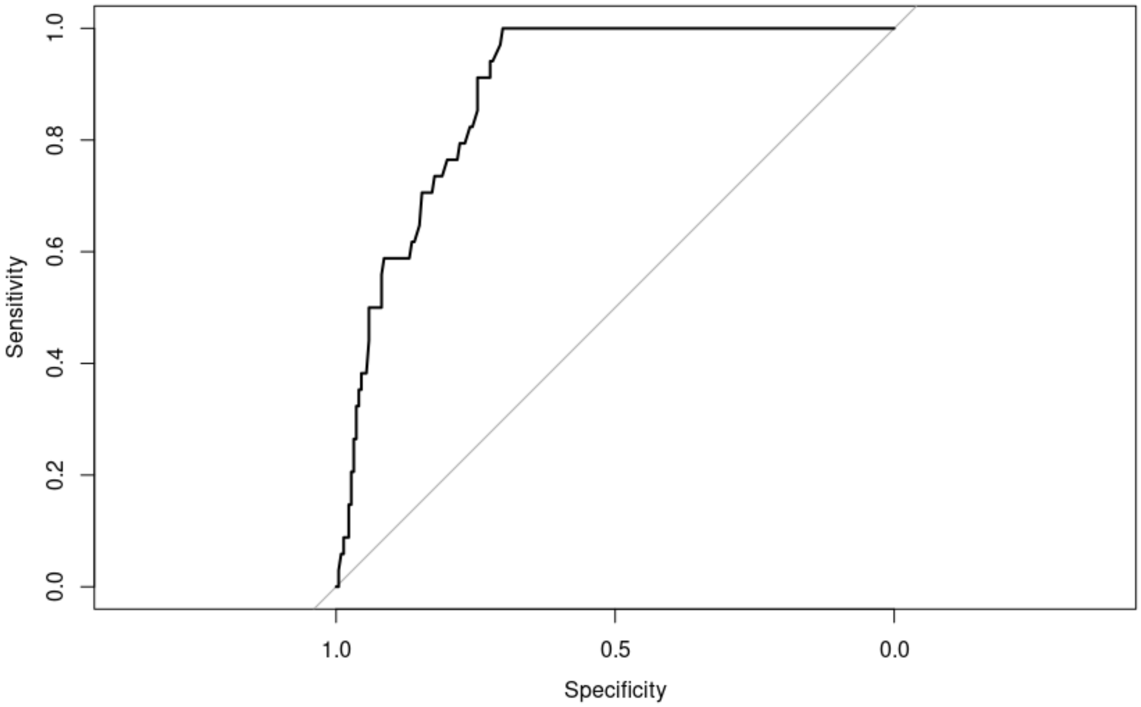
For both methods we performed hyperparameter tuning and chose parameters producing the highest AUROC score.
Then, we looked at ROC curves for each final model, chose threshold to maximize specificity, while keeping possibly high sensitivity and compared model's confusion matrix based scores, AUROCs and SHAP based feature importances.
Finally, we compared results achieved using XGBoost and Random Forest to results achieved with DNN.

## XGBoost models

We tuned two following hyperparameters: *nrounds* (number of trees), *max_depth* (max depth of a tree). We searched through a grid of possible nrounds values = (1, 2, 3, 4, 5, 6, 8, 10, runif(10, 11, 300)), and possible max_depth values = c(2, 3, 4, 5, 6, 7).

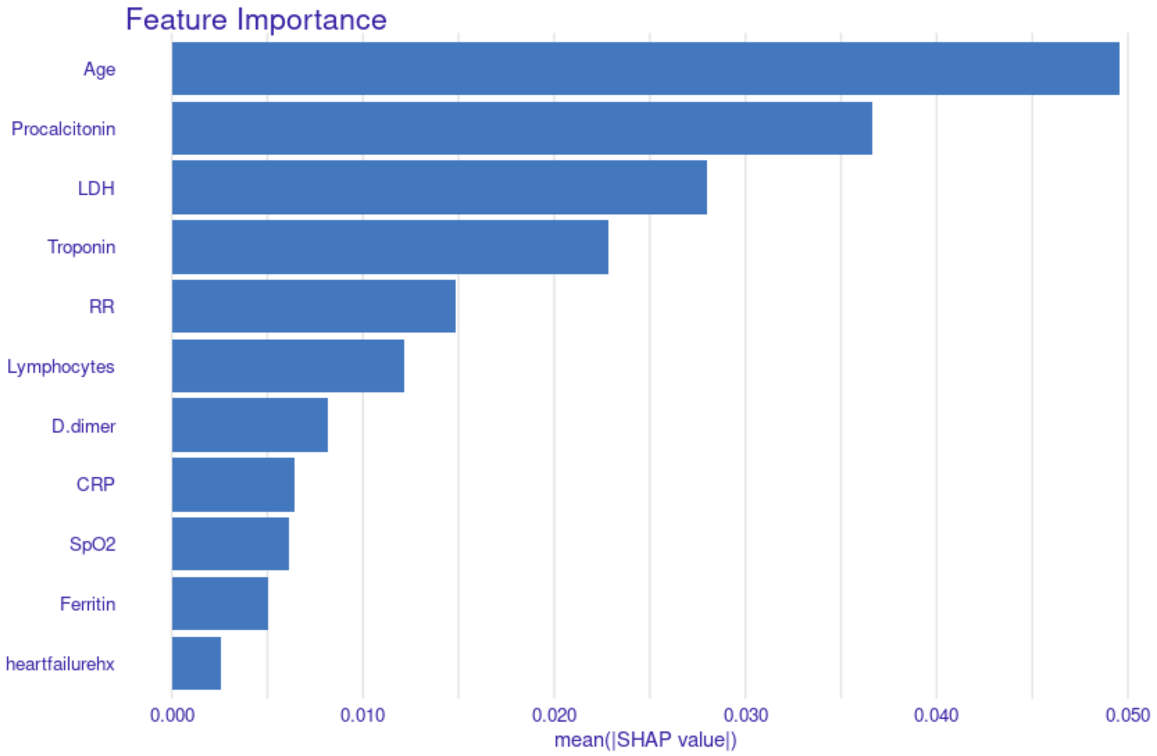
### Death prediction, all features

Model consisting of 4 trees of max_depth = 4 performed best with AUROC = 0.8869.



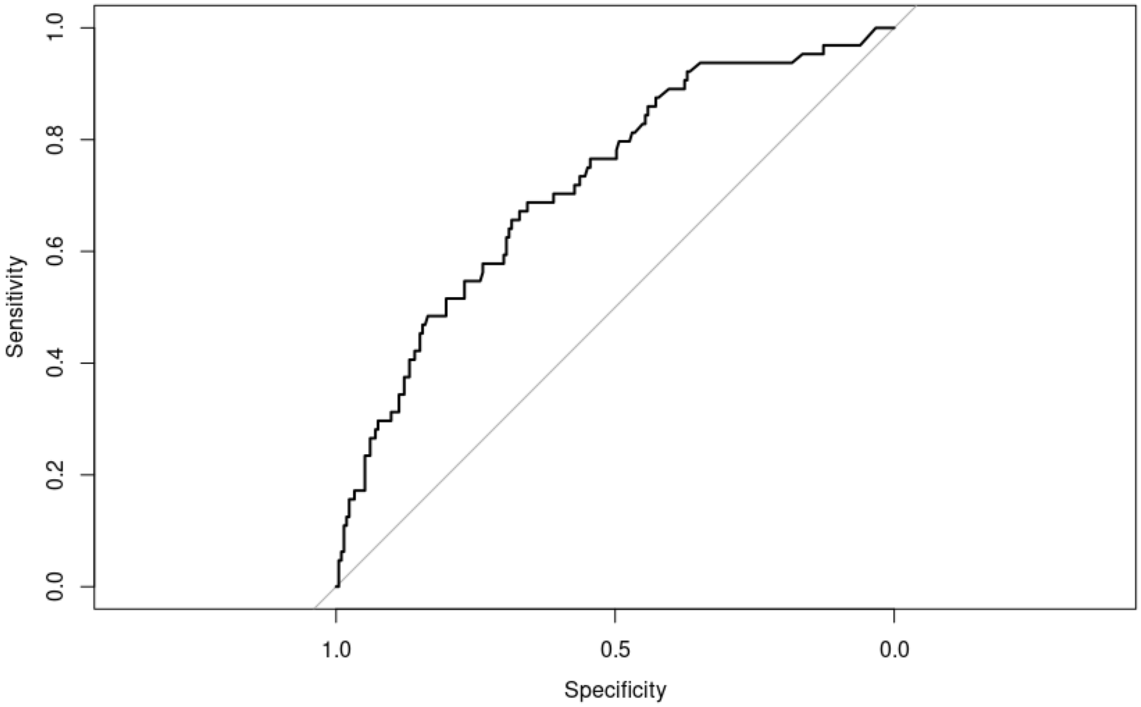
Confusion matrix with chosen threshold = 0.2:

```
Reference
Prediction   0   1
0 155   0
1  66  34

Accuracy : 0.7412
95% CI : (0.6828, 0.7938)
No Information Rate : 0.8667
P-Value [Acc > NIR] : 1

Kappa : 0.3851

Mcnemar's Test P-Value : 1.235e-15

            Sensitivity : 0.7014
            Specificity : 1.0000
         Pos Pred Value : 1.0000
         Neg Pred Value : 0.3400
             Prevalence : 0.8667
         Detection Rate : 0.6078
   Detection Prevalence : 0.6078
      Balanced Accuracy : 0.8507
```



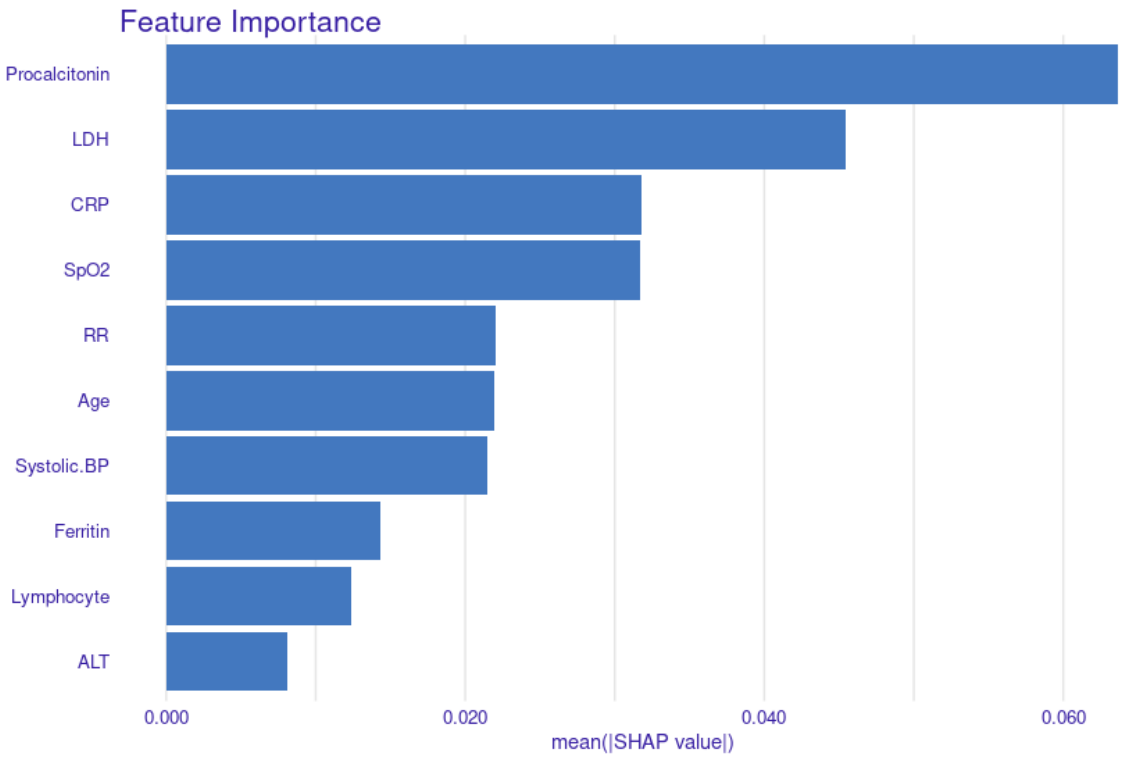
### ICU admission prediction, all features

Model consisting of 6 trees of max_depth = 4 performed best with AUROC = 0.7215.



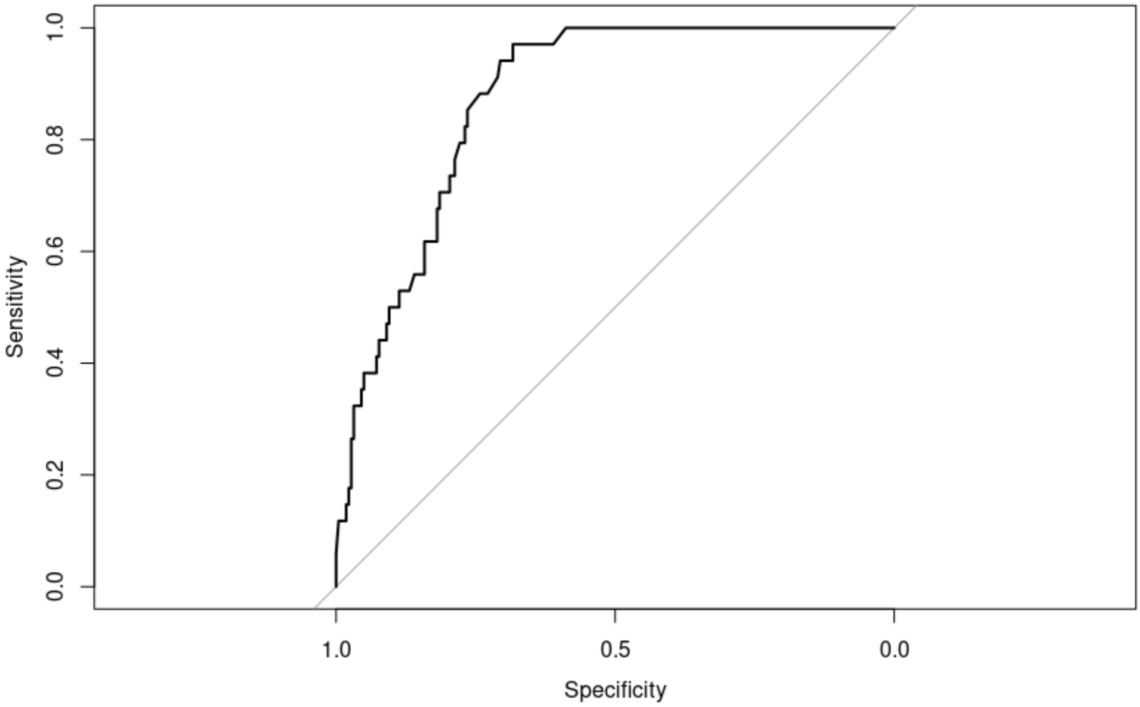
Confusion matrix with threshold = 0.2:

```
Reference
Prediction   0   1
0 107  15
1 106  49

Accuracy : 0.5632
95% CI : (0.5026, 0.6224)
No Information Rate : 0.769
P-Value [Acc > NIR] : 1

Kappa : 0.179

Mcnemar's Test P-Value : 2.796e-16

          Sensitivity : 0.5023
          Specificity : 0.7656
          Pos Pred Value : 0.8770
          Neg Pred Value : 0.3161
          Prevalence : 0.7690
          Detection Rate : 0.3863
  Detection Prevalence : 0.4404
          Balanced Accuracy : 0.6340
```



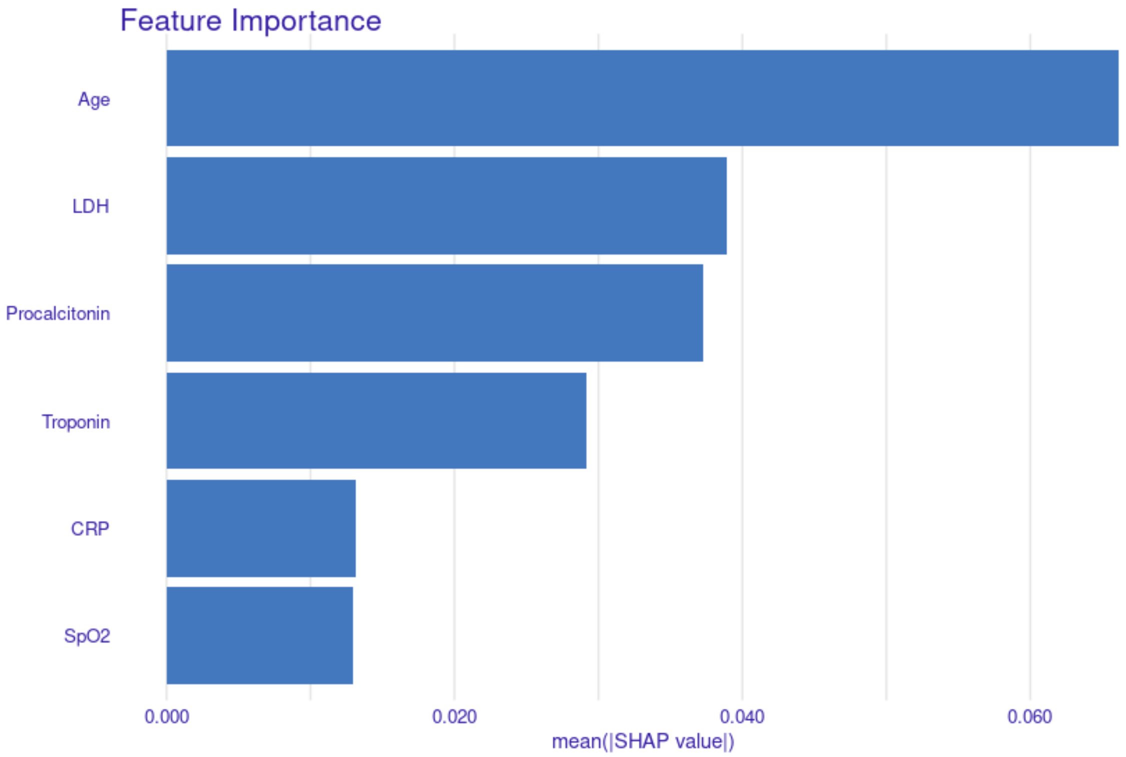
**Death prediction, selected features**

Model consisting of 5 trees of max_depth = 5 performed best with AUROC = 0.8715.



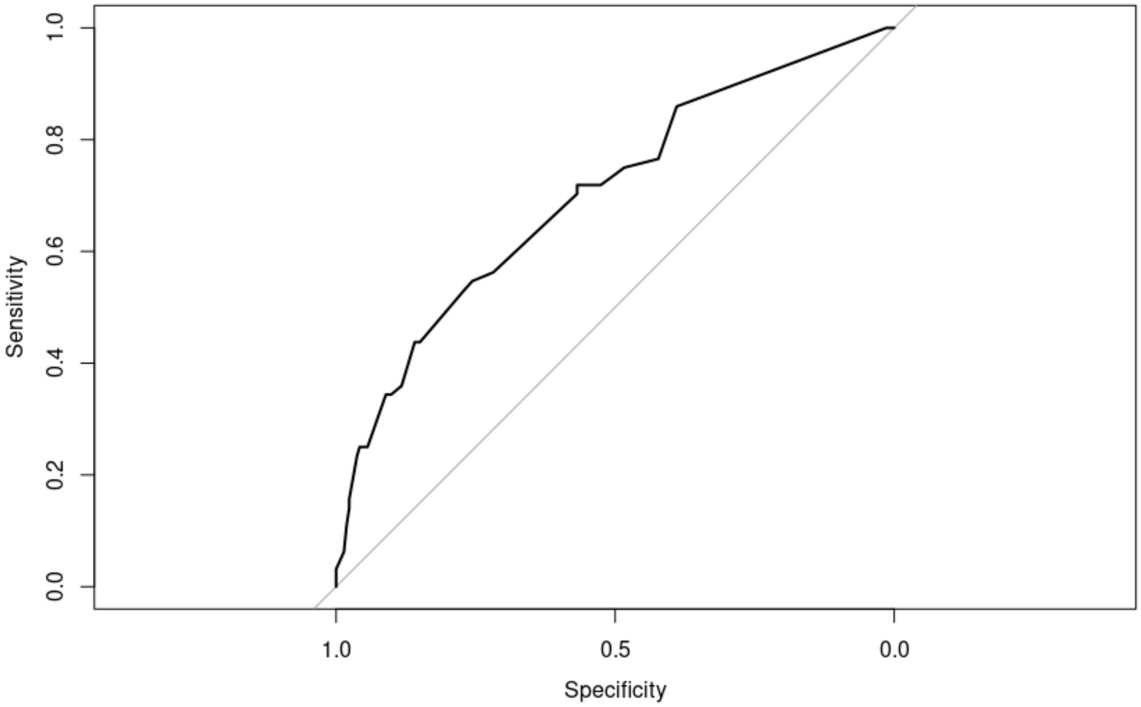
Confusion matrix with threshold = 0.16:

```
Reference
Prediction   0   1
0 151   1
1  70  33

Accuracy : 0.7216
95% CI : (0.6622, 0.7757)
No Information Rate : 0.8667
P-Value [Acc > NIR] : 1

Kappa : 0.3518

Mcnemar's Test P-Value : 7.023e-16

            Sensitivity : 0.6833
            Specificity : 0.9706
            Pos Pred Value : 0.9934
            Neg Pred Value : 0.3204
            Prevalence : 0.8667
            Detection Rate : 0.5922
   Detection Prevalence : 0.5961
            Balanced Accuracy : 0.8269
```



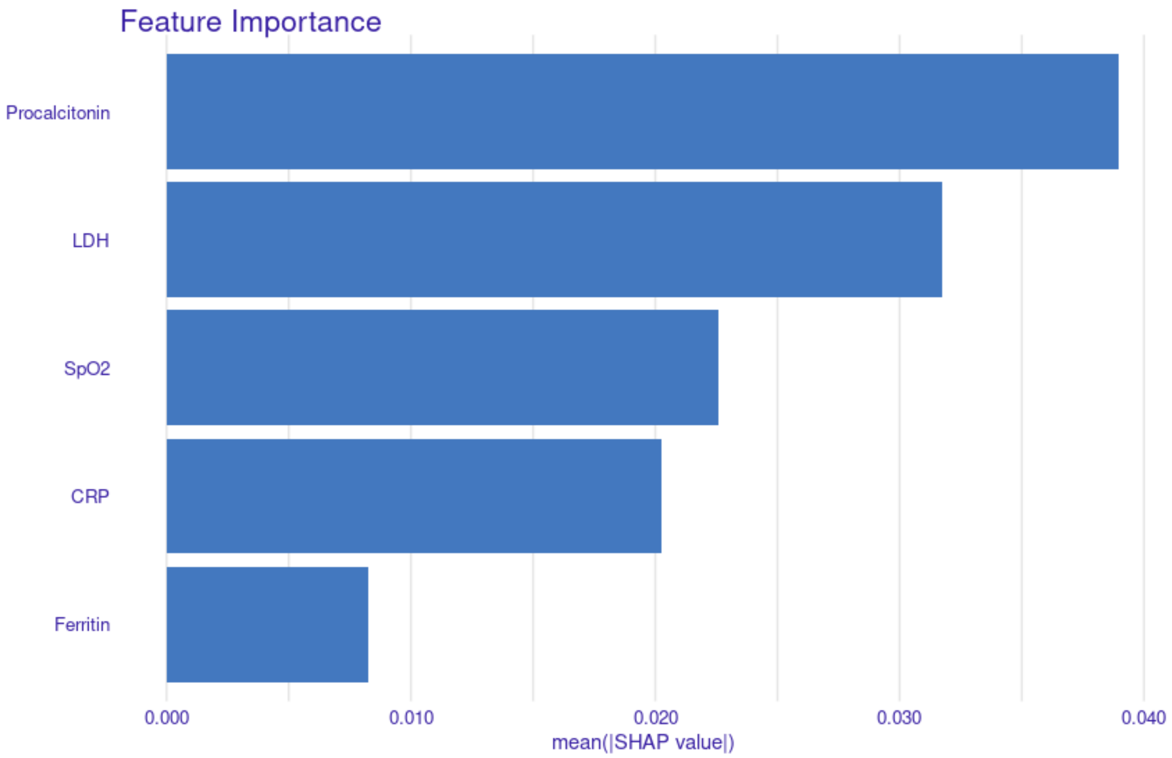
### ICU admission prediction, selected features

Model consisting of 2 trees of max_depth = 3 performed best with AUROC = 0.7019.



Confusion matrix with threshold = 0.35:

```
Reference
Prediction   0   1
0 112  18
1 101  46

Accuracy : 0.5704
95% CI : (0.5098, 0.6295)
No Information Rate : 0.769
P-Value [Acc > NIR] : 1

Kappa : 0.1683

Mcnemar's Test P-Value : 5.608e-14

            Sensitivity : 0.5258
            Specificity : 0.7188
            Pos Pred Value : 0.8615
            Neg Pred Value : 0.3129
            Prevalence : 0.7690
            Detection Rate : 0.4043
  Detection Prevalence : 0.4693
            Balanced Accuracy : 0.6223
```

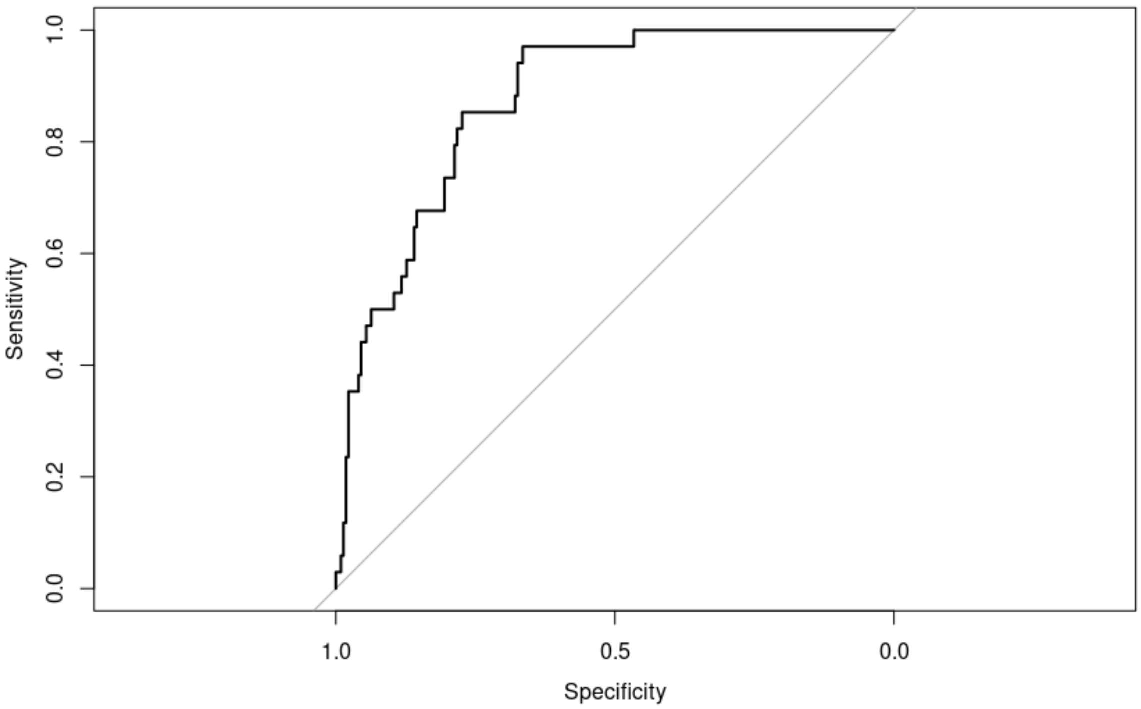


## Random Forest models

We tuned two following hyperparameters: *num.trees* (number of trees), *max.depth* (max depth of a tree). We searched through a grid of possible *num.trees* values = c(1, 3, 5, runif(11, 6, 500)), and possible *max.depth* values = c(2, 3, 4, 5, NULL). Due to the randomness in the model building process causing small fluctuations of AUC score for a fixed (*max.depth, num.trees*) pair for following tries, we tried each pair 5 times and aggregated the score of a pair as mean AUROC score of tries.

### Death prediction, all features

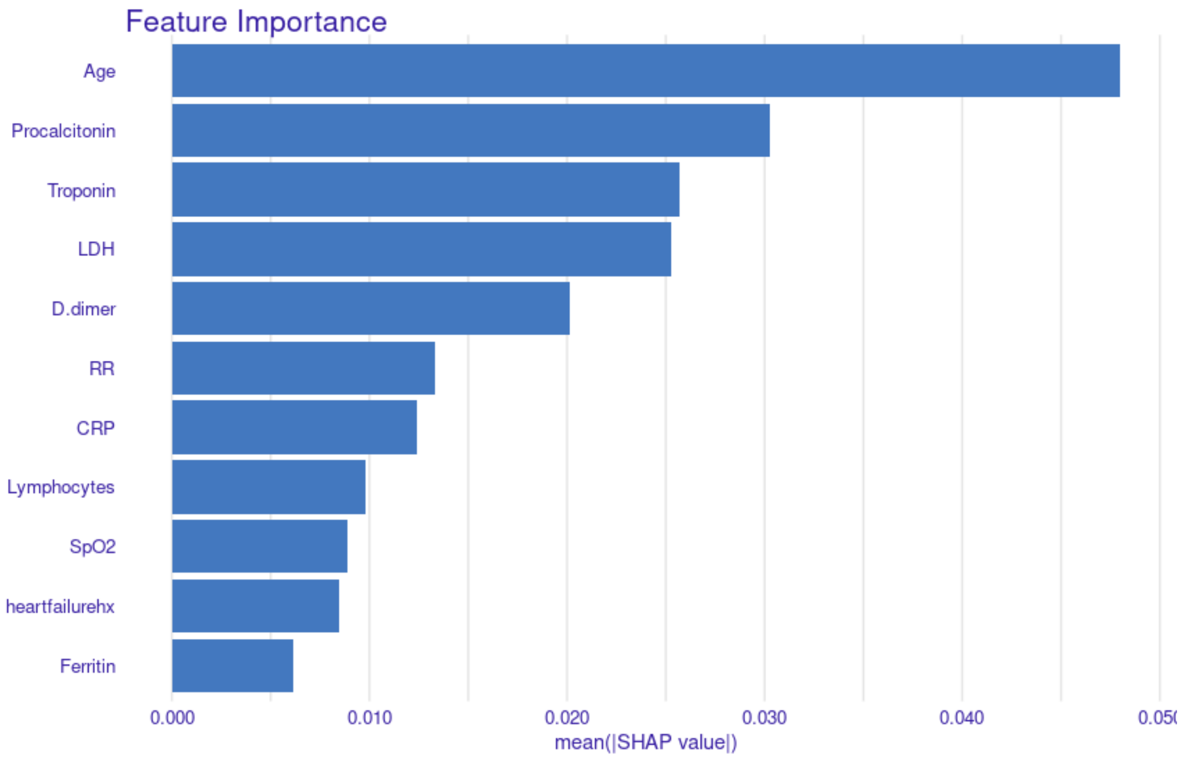
Model consisting of 135 trees of *max.depth* = 5 performed best with mean AUROC = 0.871.



AUROC = 0.8729

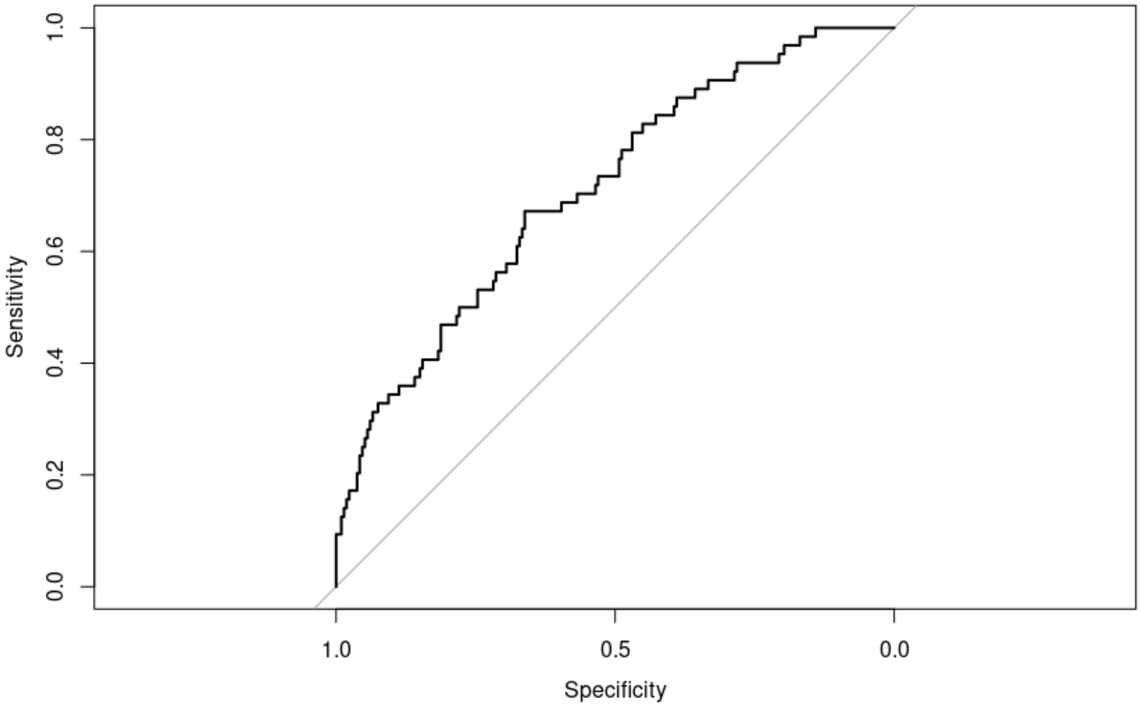
Confusion matrix with threshold = 0.11

```
Reference
Prediction   0   1
0 142   1
1  79  33

Accuracy : 0.6863
95% CI : (0.6254, 0.7427)
No Information Rate : 0.8667
P-Value [Acc > NIR] : 1

Kappa : 0.3111

Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.6425
            Specificity : 0.9706
            Pos Pred Value : 0.9930
            Neg Pred Value : 0.2946
            Prevalence : 0.8667
            Detection Rate : 0.5569
  Detection Prevalence : 0.5608
            Balanced Accuracy : 0.8066
```



### ICU admission prediction, all features

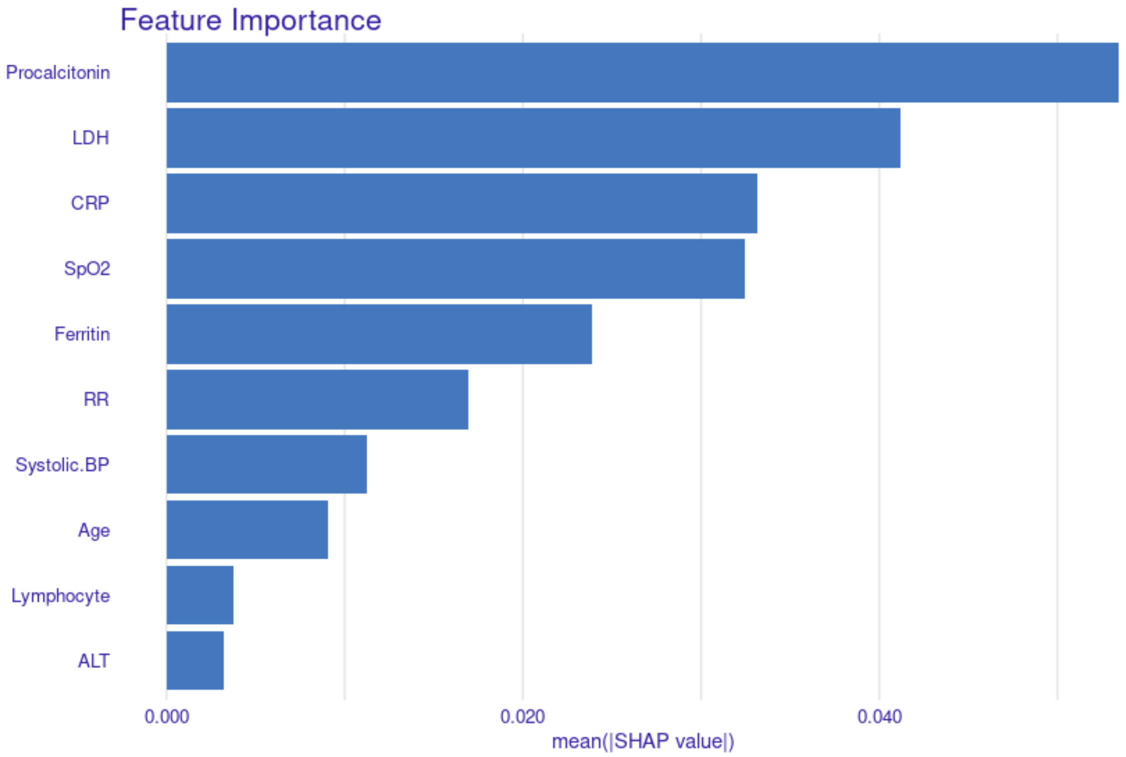
Model consisting of 389 trees of *max.depth* = 4 performed best with mean AUROC = 0.717.



AUROC = 0.7107

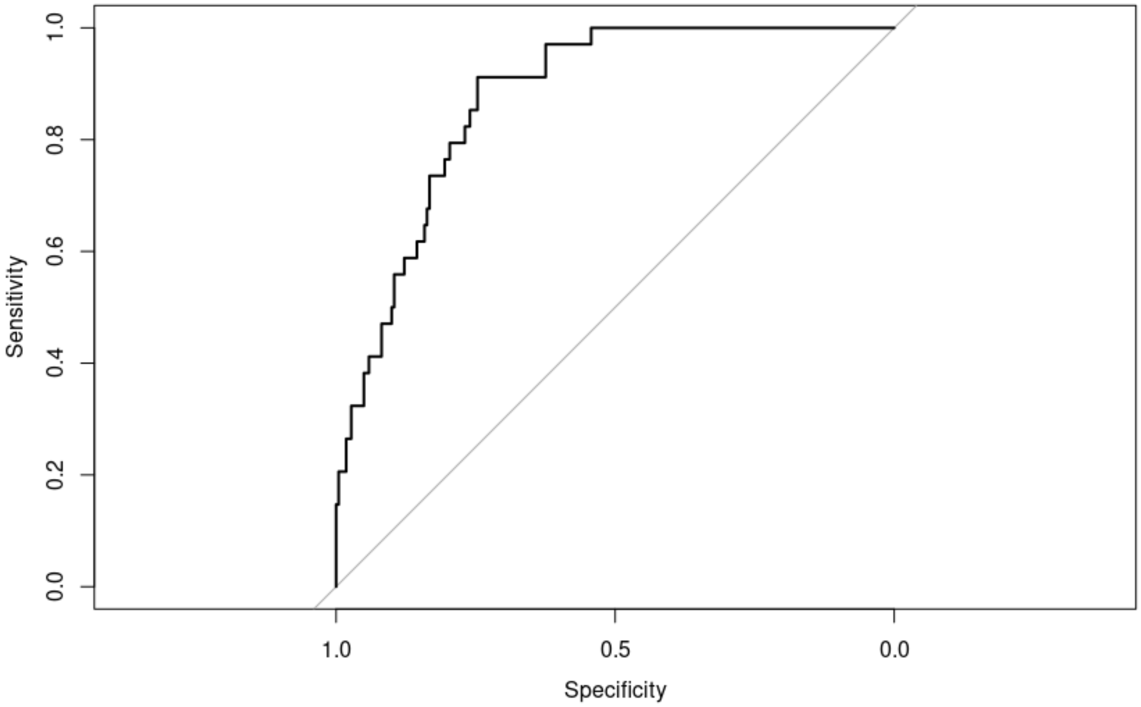
Confusion matrix for threshold = 0.2:

```
Reference
Prediction   0   1
0 114  19
1  99  45

Accuracy : 0.574
95% CI : (0.5134, 0.633)
No Information Rate : 0.769
P-Value [Acc > NIR] : 1

Kappa : 0.1658

Mcnemar's Test P-Value : 3.528e-13

            Sensitivity : 0.5352
            Specificity : 0.7031
         Pos Pred Value : 0.8571
         Neg Pred Value : 0.3125
             Prevalence : 0.7690
         Detection Rate : 0.4116
   Detection Prevalence : 0.4801
      Balanced Accuracy : 0.6192
```



### Death prediction, selected features

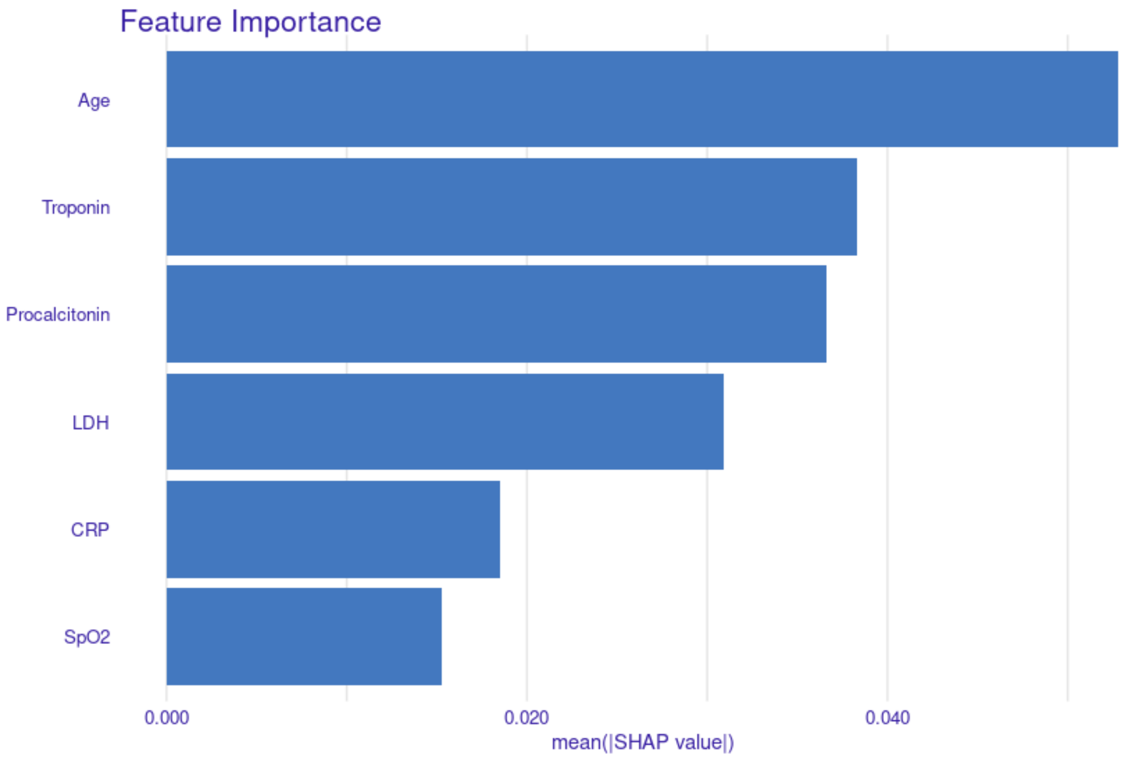
Model consisting of 144 trees of *max.depth* = 5 performed best with mean AUROC = 0.876.



AUROC = 0.8754

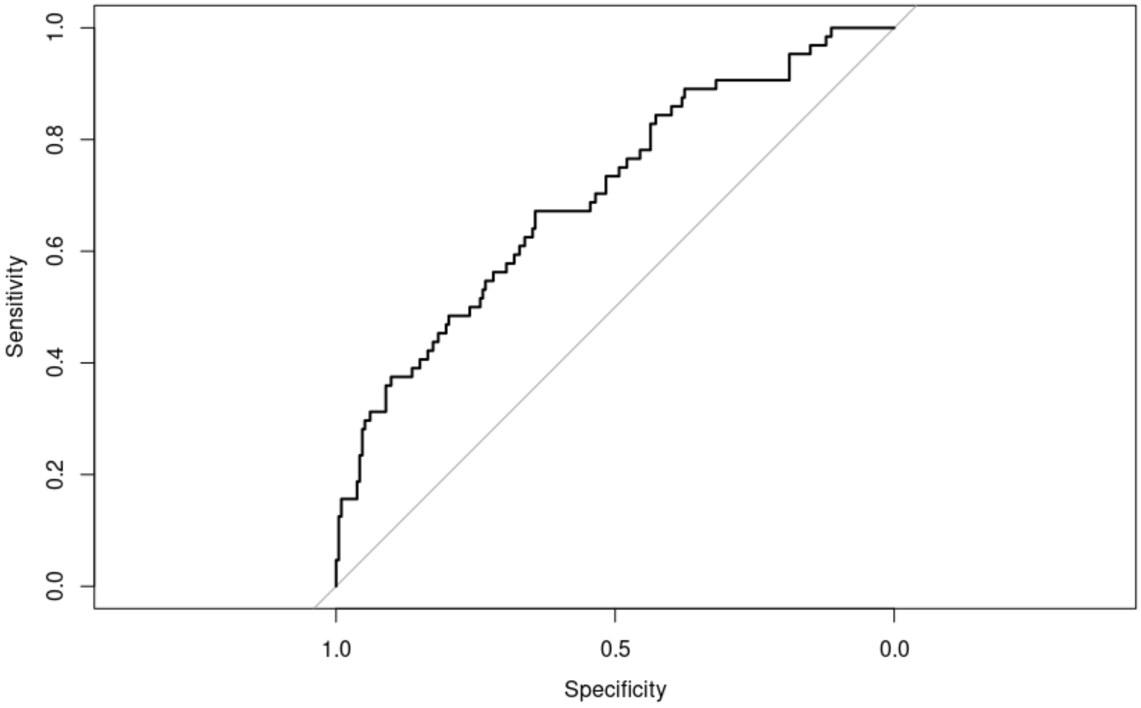
Confusion matrix with threshold = 0.1:

```
Reference
Prediction   0   1
0 138   1
1  83  33

Accuracy : 0.6706
95% CI : (0.6092, 0.7279)
No Information Rate : 0.8667
P-Value [Acc > NIR] : 1

Kappa : 0.2945

Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.6244
            Specificity : 0.9706
            Pos Pred Value : 0.9928
            Neg Pred Value : 0.2845
            Prevalence : 0.8667
            Detection Rate : 0.5412
  Detection Prevalence : 0.5451
            Balanced Accuracy : 0.7975
```



**ICU admission prediction, selected features**

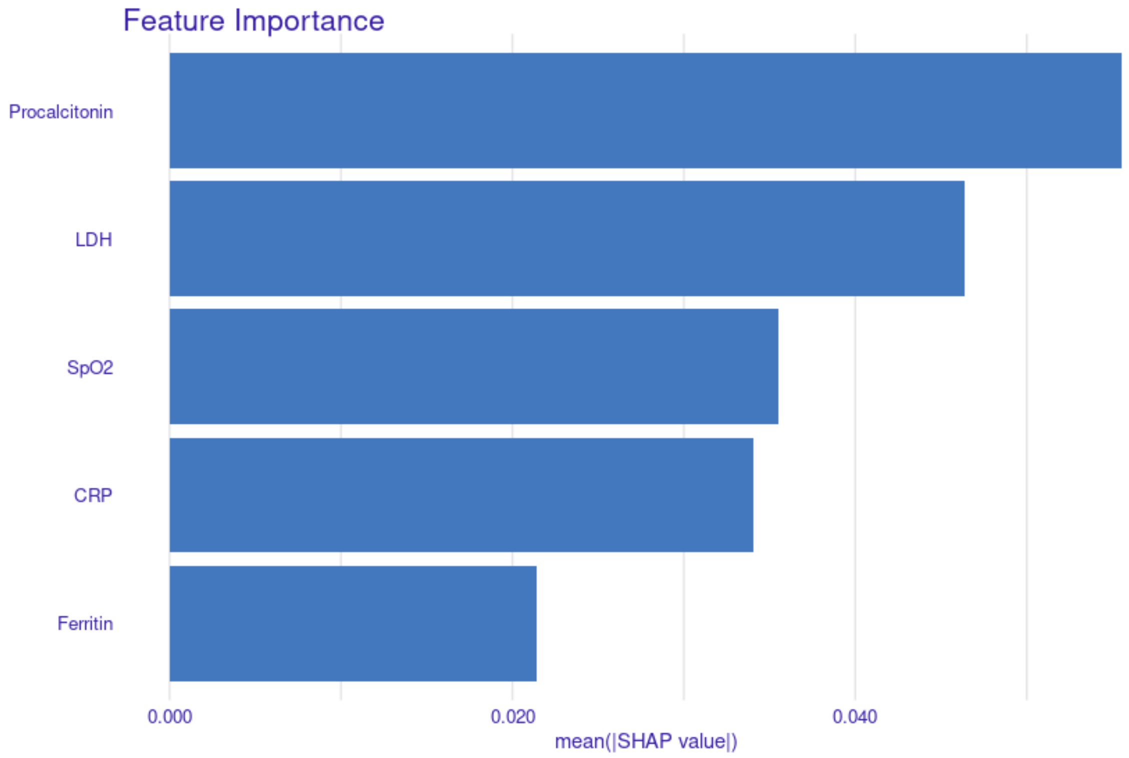
Model consisting of 359 trees of *max.depth* = 3 performed best with mean AUROC = 0.673.



AUROC = 0.7025

Confusion matrix with threshold = 0.2:

```
Reference
Prediction   0   1
0 112  19
1 101  45

Accuracy : 0.5668
95% CI : (0.5062, 0.626)
No Information Rate : 0.769
P-Value [Acc > NIR] : 1

Kappa : 0.1581

Mcnemar's Test P-Value : 1.422e-13

            Sensitivity : 0.5258
            Specificity : 0.7031
            Pos Pred Value : 0.8550
            Neg Pred Value : 0.3082
            Prevalence : 0.7690
            Detection Rate : 0.4043
  Detection Prevalence : 0.4729
            Balanced Accuracy : 0.6145
```



## XGBoost and Random Forest comparison

All models performed better at the death prediction task. They all achieved around 0.85 AUROC while predicting patient's death compared to around 0.7 AUROC while predicting patient’s ICU admission.

In death prediction applying feature selection yielded in no changes in AUC scores.
In ICU admission prediction feature selection minimally decreased achieved AUCs.

Looking at feature importances for ICU admission prediction we can see that selected features almost match ones most important in all-feature models. Order of features by their importance is almost not affected, too.

As XGBoost death prediction model based on 6 features had only slightly worse performance than based on all 10 features and outperformed both random forests, we would choose it as most clinically applicable of examined models. Difference in feature importances suggests that it may be worth checking if selection of different features would produce better results. What is more, chosen gradient boosting models are much simpler than ranger models, they consist of fewer trees, the trees are also rather shallow. It makes XGBoost models more interpretable - interpretations can be computed for them faster (although it does not mean much, as for random forest they can still be computed very fast) and they can be examined by eye.

Same goes for ICU prediction. XGBoost performed slightly better than random forest, and was less complex. The model on selected features was only a sum of 2 shallow trees, what makes it possible to comprehend even for a clinician.

Unfortunately single decision trees have not scored near to chosen best ones.

## DNN and tree-based models comparison

When comparing the performance, tree-based models and neural networks with our modifications produced similar results. Original DNN from the article performed even worse. But the biggest difference between DNNs and tree-based models lies in the interpretability. XGBoost models chosen by us are so simple, that they can be grasped by eye. We use SHAP values to measure feature importance. SHAP values for tree-based models can be calculated precisely in polynomial time, while for neural networks they can only be approximated. There are also many more explanation methods that are available for tree models but not for DNNs.