forester report

version 1.6.1

2024-11-22 18:48:41.262276

The best models

This is the **binary_clf** task.

The best model evaluated on testing set is: **lightgbm_RS_1**, whereas for the validation set it is: **lightgbm_RS_1**.

The models inside the forester package are trained on the training set, the Bayesian Optimization is tuned according to the testing set, and the validation set is never seen during the training. The training set should not be used for evaluation as the models always perform the best for the data they have seen (overfitting). The least biased dataset is the validation set, however we can also use the testing set, ex. to check if the models overfit.

The names of the models were created by a pattern *Engine_TuningMethod_Id*, where:

- `Engine` - describes the engine used for the training (random_forest, xgboost, decision_tree, lightgbm, catboost),
- `TuningMethod` - describes how the model was tuned (`basic` for basic parameters, `RS` for random search, `bayes` for Bayesian optimization),
- `Id` - is used for separating the random search parameters sets.

More details about the dataset are present at the end of the report.

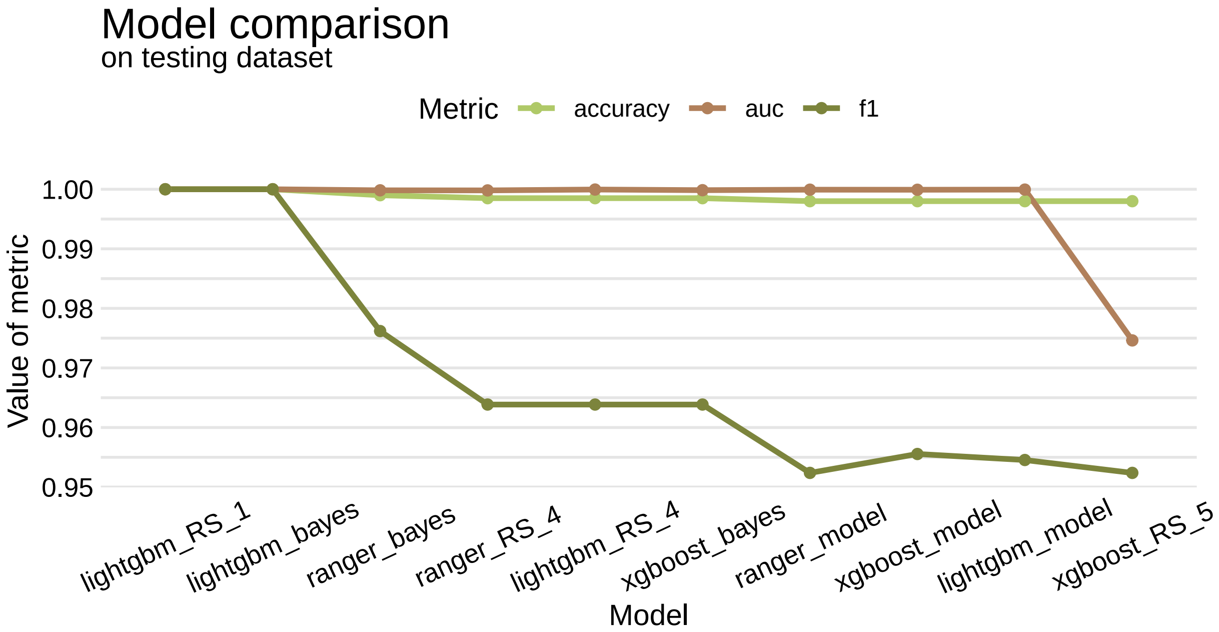
Best models for validation dataset

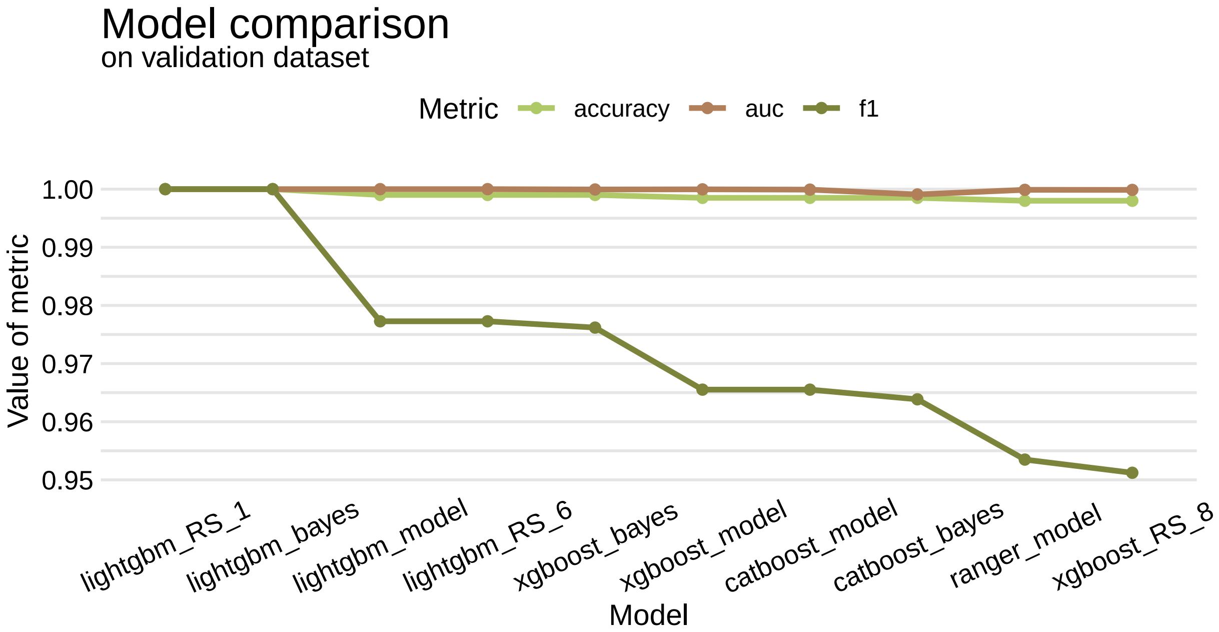
no.	name	accuracy	auc	f1
25	lightgbm_RS_1	1.0000	1.0000	1.0000
47	lightgbm_bayes	1.0000	1.0000	1.0000
3	lightgbm_model	0.9990	1.0000	0.9773
30	lightgbm_RS_6	0.9990	1.0000	0.9773
46	xgboost_bayes	0.9990	0.9999	0.9762
2	xgboost_model	0.9985	1.0000	0.9655
4	catboost_model	0.9985	0.9999	0.9655
48	catboost_bayes	0.9985	0.9991	0.9639
1	ranger_model	0.9980	0.9999	0.9535
22	xgboost_RS_8	0.9980	0.9999	0.9512

Model comparison

Metrics comparison

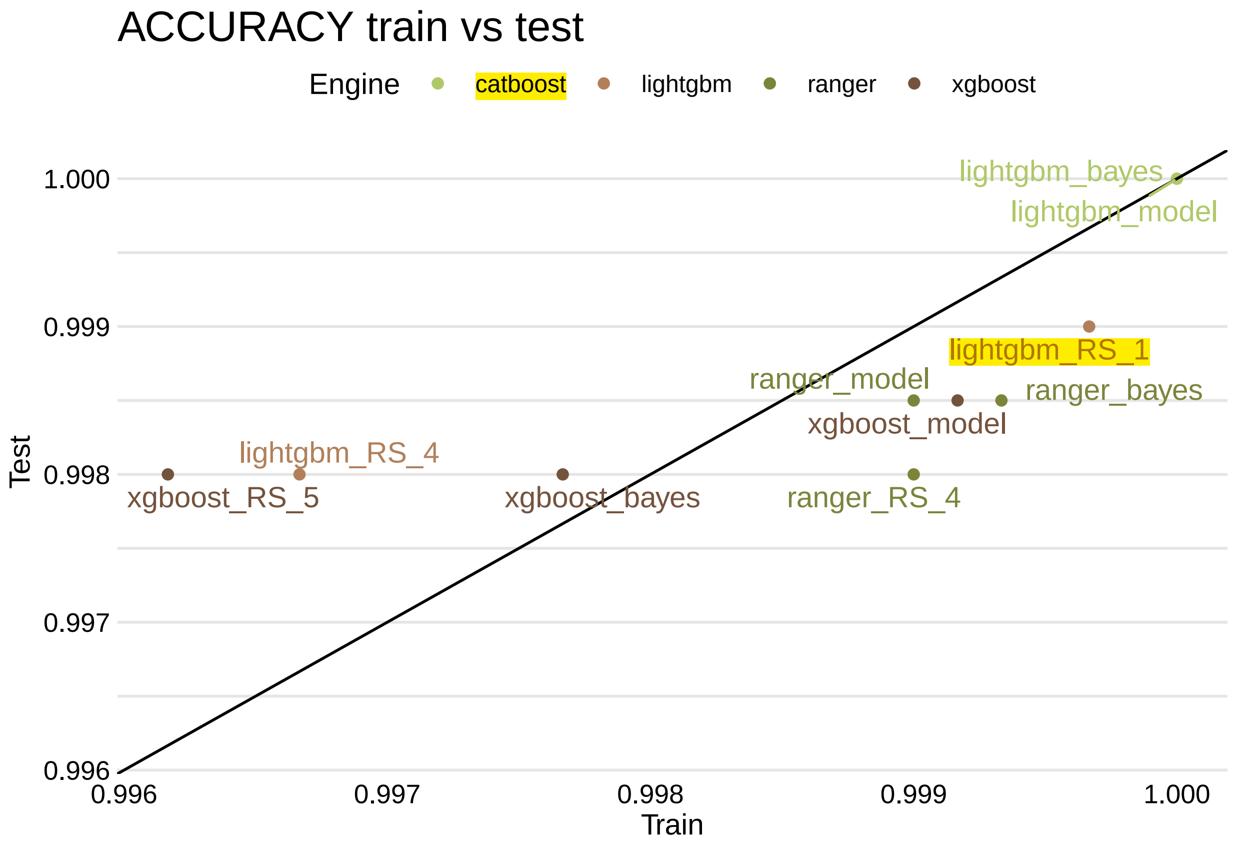
The comparison plot takes a closer look on top 10 performing models, and evaluates their performance in terms of three well-known metrics: accuracy, area under the curve (AUC), and F1 score. For each metric, the larger the value, the better the model is. As the ranked list compares the models only in terms of accuracy, we want to additionally evaluate the performance in terms of other metrics. In some cases it might happen that other model is better in terms of AUC, or F1, but slightly worse in accuracy, and we would like to choose the other model. The results are presented for both testing and validation dataset.





Train vs test plot

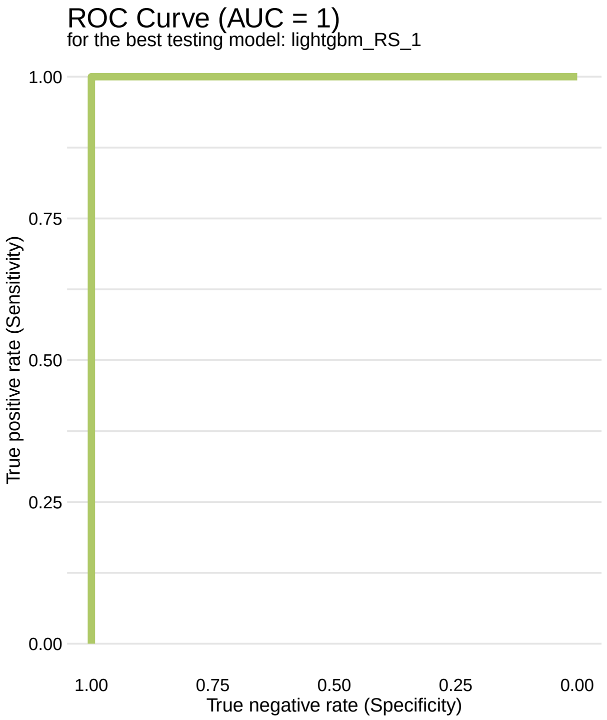
This scatter plot tackles the issue of the overfitting, and compares large amounts of models at once. On the x axis we provide the metrics value evaluated on the training dataset, whereas on the y axis we have the same for the testing dataset. Models performance is assessed in two ways. Firstly, we want the model to have as small value as possible on the testing dataset (so we want it to be lower than other models). Secondly, we want to choose the model which is close to the x = y line, because it means that the model is not overfitted, so it generalizes better. In most cases we want to chose the model that is less overfitted, even though it has worse performance.

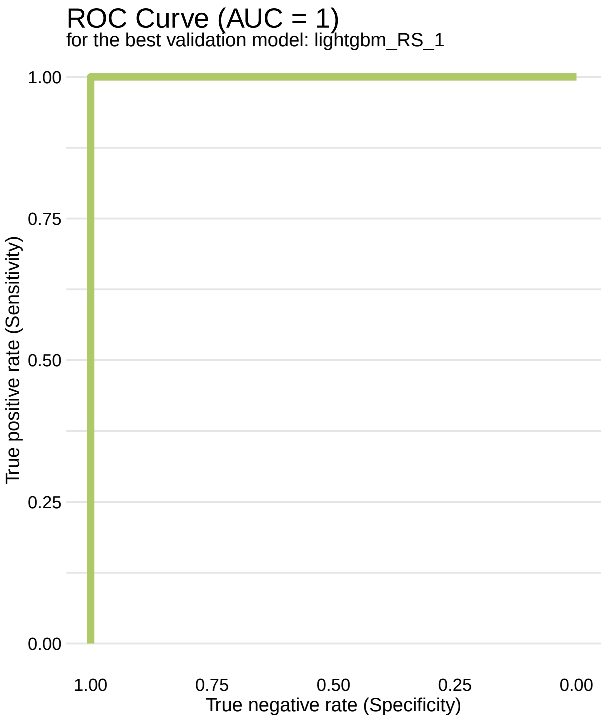


Plots for the best models

ROC

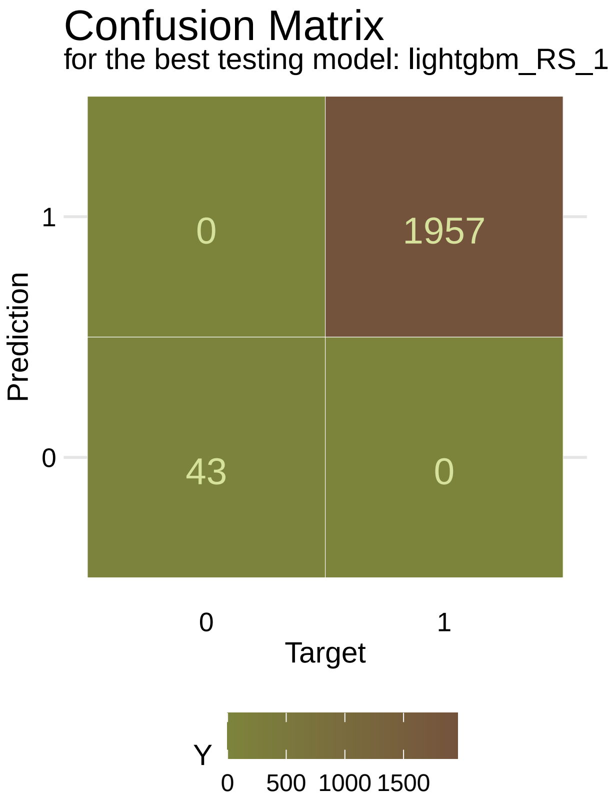
A ROC curve (receiver operating characteristic curve) is a graph showing the performance of a classification model at all classification thresholds. This curve plots two parameters: True Positive Rate (TPR/Sensitivity $= \frac{TP}{TP+FN}$), and True Negative Rate (TNR/Specificity $= \frac{TN}{TN+FP}$). A ROC curve plots TPR vs. TNR at different classification thresholds. Lowering the classification threshold classifies (probability that the prediction is classified as positive) more items as positive, thus increasing both False Positives (FP) and True Positives (TP). AUC stands for “Area under the ROC Curve.” That is, AUC measures the entire two-dimensional area underneath the entire ROC curve (think integral calculus) from (1, 0) to (0, 1). The greater the value, the better the model distinguishes between the two classes.

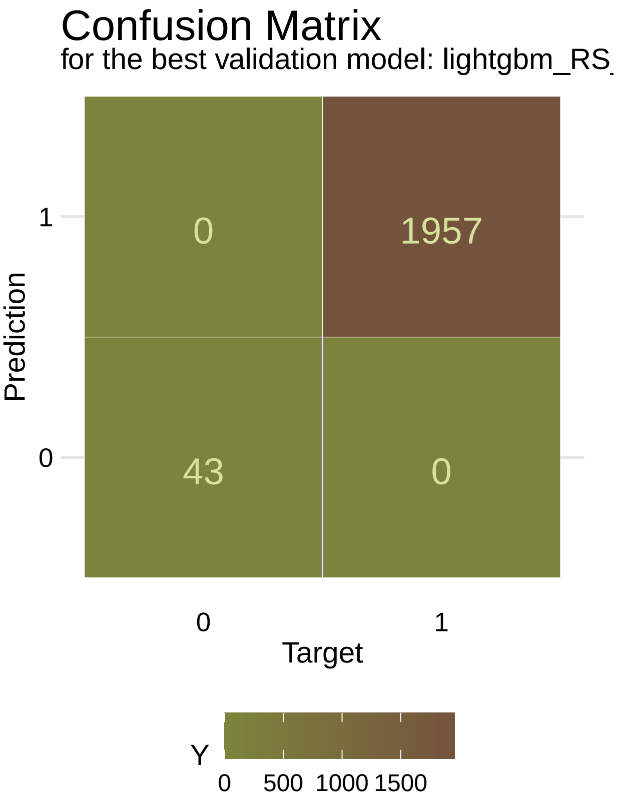

ROC Curve (AUC = 1)
for the best testing model: lightgbm_RS_1


ROC Curve (AUC = 1)
for the best validation model: lightgbm_RS_1

Confusion matrix

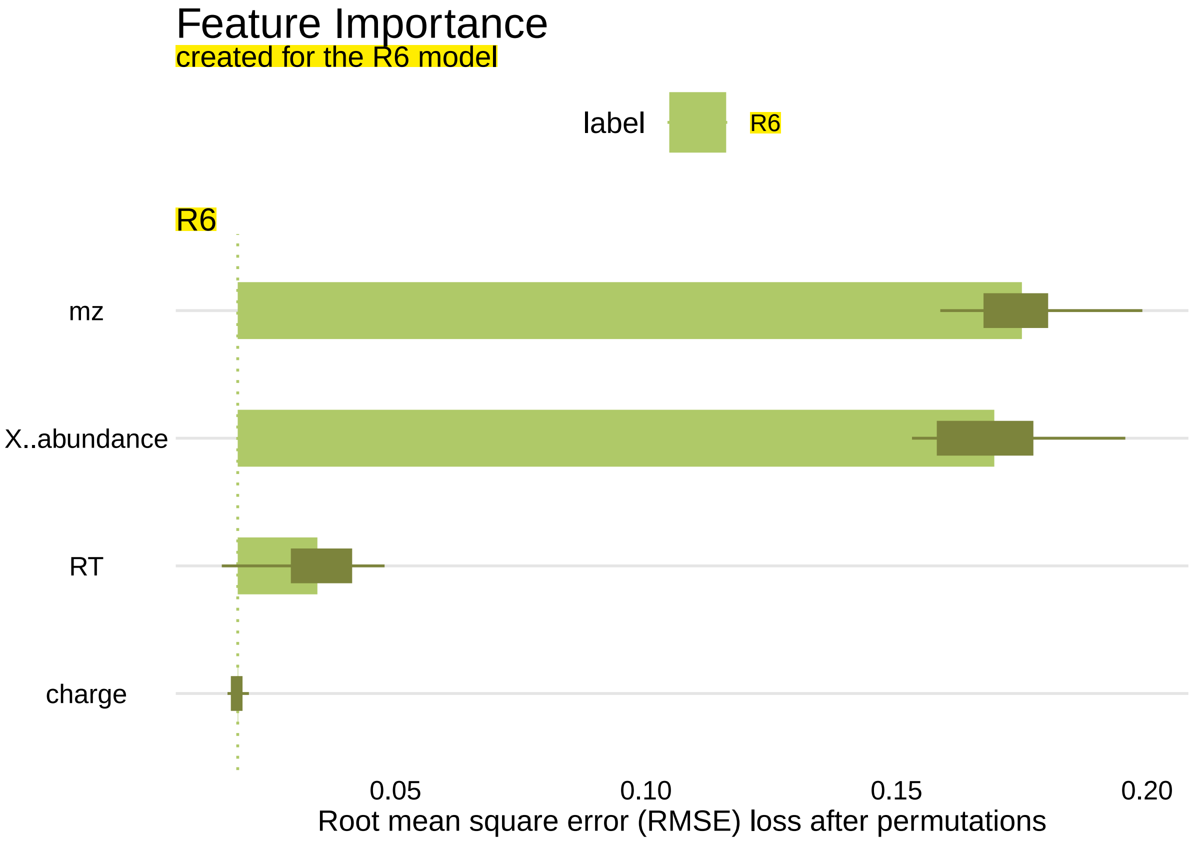
The confusion matrix is a simple way to to visualize which types of errors are made by the model. The plot below presents us the raw values of TP (x = 1, y = 1; True Positive), FP (x = 0, y = 1; False Positive), TN (x = 0, y = 0; True Negative), and FN (x = 1, y = 0; False Negative). Thanks to this visualization we can ex. see if our model has a tendency to predict mostly one class.





Feature Importance

The final visualization presents us with the feature importance plot which lets us understand what's happening inside the best model evaluated on validation set. Feature Importance (FI) shows us the most important variables for the model, and the bigger the absolute value, the more important a variable is. Large FI values for a feature indicate that if we permute the values for the column randomly, it changes the final outcomes drastically.



Details about data

———— CHECK DATA REPORT ————

The dataset has 10000 observations and 5 columns which names are:

charge; RT; mz; type; % abundance;

With the target described by a column type.

Static columns are: charge;

With dominating values: 1;

No duplicate columns.

No target values are missing.

No predictor values are missing.

No issues with dimensionality.

No strongly correlated, by Spearman rank, pairs of numerical values.

There are more than 50 possible outliers in the data set, so we are not printing them. They are returned in the output as a vector.

Dataset is unbalanced with: 45.51163 proportion with common being a dominating class.

Columns names suggest that none of them are IDs.

Columns data suggest that none of them are IDs.

———— CHECK DATA REPORT END ————