

ROBERT v 1.2.1 2025/08/05 18:55:27

How to cite: Dalmau, D.; Alegre Requena, J. V. WIREs Comput Mol Sci. 2024, DOI: 10.1002/WCMS.1733



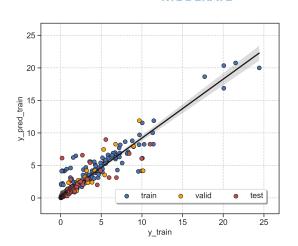
Section A. ROBERT Score

This score is designed to evaluate the models using different metrics.

No PFI (standard descriptor filter):

Model = RF · Train:Validation:Test = 81:9:10 Points(train+valid.):descriptors = 1008:6 Score = 7 / 10

MODERATE



Train: $R^2 = 0.92$, MAE = 0.2, RMSE = 0.62 Valid. : $R^2 = 0.84$, MAE = 0.31, RMSE = 0.78 Test: $R^2 = 0.69$, MAE = 0.43, RMSE = 1.1

Severe warnings

No severe warnings detected

Moderate warnings

- Uneven y distribution (Section C)
- Potential "faulty" outliers (Section E)

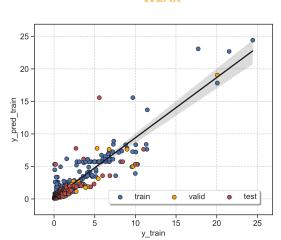
Overall assessment

Decent model, but it has limitations

PFI (only most important descriptors):

Model = RF · Train:Validation:Test = 81:9:10 Points(train+valid.):descriptors = 1008:4 Score = 6 / 10

WEAK



Train: $R^2 = 0.87$, MAE = 0.38, RMSE = 0.78 Valid.: $R^2 = 0.9$, MAE = 0.44, RMSE = 0.91 Test: $R^2 = 0.51$, MAE = 0.61, RMSE = 1.5

Severe warnings

No severe warnings detected

Moderate warnings

- Uneven y distribution (Section C)
- Potential "faulty" outliers (Section E)

Overall assessment

The model is unreliable

ROBERT v 1.2.1 Page 1 of 8

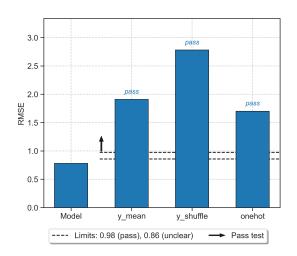
Section B. Advanced Score Analysis

This section explains each component that comprises the ROBERT score.

1. Model vs "flawed" models (3 / 3

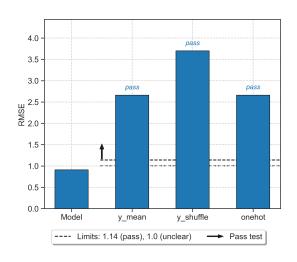
The model predicts right for the right reasons.

Pass: +1, Unclear: 0, Fail: -1. Details here.



1. Model vs "flawed" models (3 / 3

The model predicts right for the right reasons. Pass: +1, Unclear: 0, Fail: -1. *Details here.*



2. Predictive ability of the model (0 / 2)

Low predictive ability with R^2 (test) = 0.69. R^2 0.70-0.85: +1, R^2 >0.85: +2.

2. Predictive ability of the model (0 / 2)

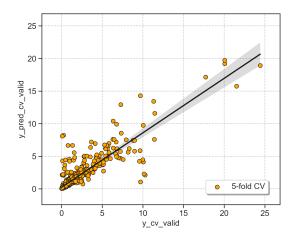
Low predictive ability with R^2 (test) = 0.51. R^2 0.70-0.85: +1, R^2 >0.85: +2.

3. Cross-validation (5-fold CV) of the model

Overfitting analysis on the model with 3a and 3b:

3a. CV predictions train + valid. (1 / 2)

Moderate predictive ability with R^2 (5-fold CV) = 0.8. R^2 0.70-0.85: +1, R^2 >0.85: +2.

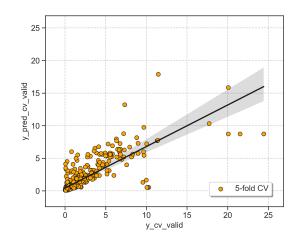


3. Cross-validation (5-fold CV) of the model

Overfitting analysis on the model with 3a and 3b:

3a. CV predictions train + valid. (0 / 2)

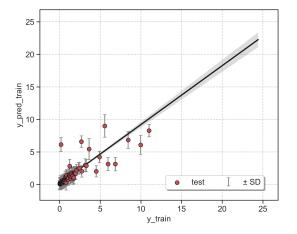
Low predictive ability with R^2 (5-fold CV) = 0.68. R^2 0.70-0.85: +1, R^2 >0.85: +2.



ROBERT v 1.2.1 Page 2 of 8

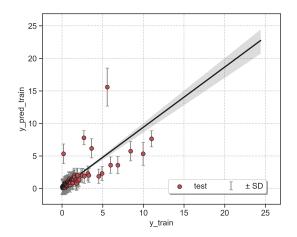
3b. Avg. standard deviation (SD) (2 / 2 Low variation, 4*SD (test) = 3.7 (15% y-range).

4*SD 25-50% y-range: +1, 4*SD < 25% y-range: +2. *Details here.*



3b. Avg. standard deviation (SD) (2 / 2

Low variation, 4*SD (test) = 4.5 (19% y-range). 4*SD 25-50% y-range: +1, 4*SD < 25% y-range: +2. Details here.



4. Points(train+valid.):descriptors (1 / 1

Decent number of descps. (ratio 1008:6). 5 or more points per descriptor: +1.

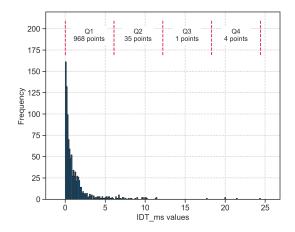
4. Points(train+valid.):descriptors (1 / 1 ===)

Decent number of descps. (ratio 1008:4). 5 or more points per descriptor: +1.



Section C. Distribution of y Values

This section shows the distribution of y values within the training and validation sets.



200 - Q1 Q2 Q3 Q4 4 points 175 - 150 - 25 - 0 0 5 10 15 20 25 IDT ms values

y distribution analysis

x WARNING! Your data is not uniform (Q3 has 1 points while Q1 has 968)

y distribution analysis

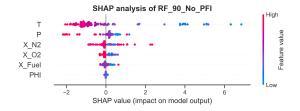
x WARNING! Your data is not uniform (Q3 has 1 points while Q1 has 968)

ROBERT v 1.2.1 Page 3 of 8

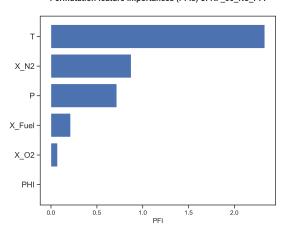


Section D. Feature Importances

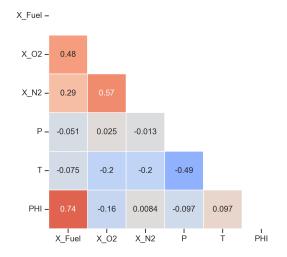
This section presents feature importances measured using the validation set.



Permutation feature importances (PFIs) of RF_90_No_PFI

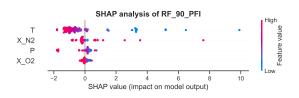


Pearson's r heatmap_No_PFI

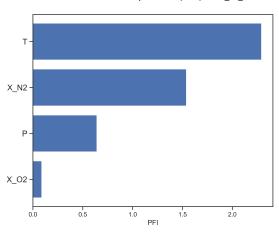


Correlation analysis

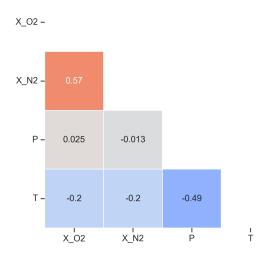
o Correlations between variables are acceptable



Permutation feature importances (PFIs) of RF_90_PFI



Pearson's r heatmap_PFI



Correlation analysis

o Correlations between variables are acceptable

ROBERT v 1.2.1 Page 4 of 8



Section E. Outlier Analysis

This section detects outliers using the standard deviation (SD) of errors from the training set.

No PFI (standard descriptor filter):

Outliers (max. 10 shown)

Train: 33 outliers out of 907 datapoints (3.6%)

- 101 (7.4 SDs)
- 102 (6.0 SDs)
- 105 (3.2 SDs)
- 292 (5.9 SDs)
- 293 (6.2 SDs)
- 294 (6.4 SDs)
- 204 (0.4 000)
- 295 (6.5 SDs)
- 296 (3.0 SDs)- 297 (2.4 SDs)
- 300 (3.0 SDs)

Validation: 6 outliers out of 101 datapoints (5.9%)

- 322 (9.7 SDs)
- 524 (3.4 SDs)
- 648 (2.2 SDs)
- 662 (2.9 SDs)
- 664 (2.1 SDs)
- 989 (3.4 SDs)

Test: 11 outliers out of 112 datapoints (9.8%)

- 301 (9.8 SDs)
- 140 (2.3 SDs)
- 560 (6.3 SDs)
- 1118 (2.7 SDs)
- 683 (3.9 SDs)
- 991 (5.5 SDs)
- 363 (4.5 SDs)
- 362 (6.0 SDs) - 710 (2.4 SDs)
- 324 (6.3 SDs)

10.0 7.5 5.0 9 0.0 -2.5 -5.0 -7.5 -10.0 -7.5 -5.0 -2.5 SD of the errors

PFI (only most important descriptors):

Outliers (max. 10 shown)

Train: 33 outliers out of 907 datapoints (3.6%)

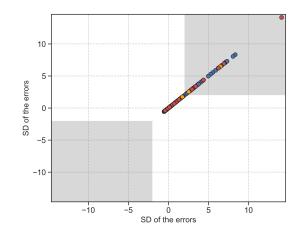
- 27 (3.1 SDs)
- 292 (2.5 SDs)
- 293 (2.8 SDs)
- 294 (3.0 SDs)
- 295 (3.1 SDs)
- 296 (5.9 SDs)
- 298 (5.5 SDs)
- 299 (3.8 SDs)
- 300 (4.0 SDs) - 302 (7.1 SDs)

Validation: 6 outliers out of 101 datapoints (5.9%)

- 319 (6.3 SDs)
- 322 (6.5 SDs)
- 444 (2.9 SDs)
- 524 (3.1 SDs)
- 662 (2.8 SDs)
- 664 (2.5 SDs)

Test: 11 outliers out of 112 datapoints (9.8%)

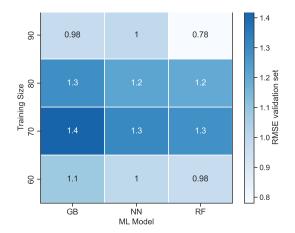
- 301 (7.0 SDs)
- 560 (6.9 SDs)
- 1118 (3.1 SDs)
- 683 (3.3 SDs)
- 378 (3.3 SDs)
- 991 (1.4e+01 SDs)
- 363 (2.9 SDs)
- 362 (4.2 SDs)
- 710 (3.4 SDs)
- 324 (6.2 SDs)

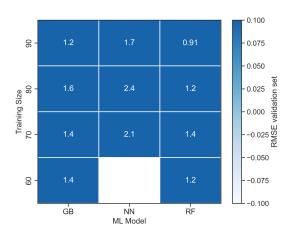


ROBERT v 1.2.1 Page 5 of 8

Section F. Model Screening

This section compares different combinations of hyperoptimized algorithms and partition sizes.







Section G. Reproducibility

This section provides all the instructions to reproduce the results presented.

1. Download these files (the authors should have uploaded the files as supporting information!):

- CSV database (JetFuel_Ignition.csv)

2. Install and adjust the versions of the following Python modules:

- Install ROBERT and its dependencies: conda install -y -c conda-forge robert
- Adjust ROBERT version: pip install robert==1.2.1
- Install scikit-learn-intelex: pip install scikit-learn-intelex==2024.7.0

(if scikit-learn-intelex is not installed, slightly different results might be obtained)

3. Run ROBERT using this command line in the folder with the CSV database:

python -m robert --names "Point" --y "IDT ms" --model "[RF,GB,NN]" --csv name "JetFuel Ignition.csv"

4. Execution time, Python version and OS:

Originally run in Python 3.12.2 using Linux #1 SMP Fri Apr 20 16:44:24 UTC 2018

Total execution time: 473.05 seconds (the number of processors should be specified by the user)

ROBERT v 1.2.1 Page 6 of 8



Section H. Transparency

This section contains important parameters used in scikit-learn models and ROBERT.

1. Parameters of the scikit-learn models (same keywords as used in scikit-learn):

No PFI (standard descriptor filter): PFI (only most important descriptors):

sklearn model: RandomForestRegressor sklearn model: RandomForestRegressor

random state: 0 random state: 43 names: Point names: Point n estimators: 5 n estimators: 5 max depth: 60 max depth: 5 max features: 1.0 max features: 1.0 min samples split: 2 min samples split: 2 min_samples_leaf: 1 min_samples_leaf: 1

min_weight_fraction_leaf: 0 min_weight_fraction_leaf: 0

ccp_alpha: 0 ccp_alpha: 0 oob_score: False oob_score: True max_samples: 0.75 max_samples: 0.75

2. ROBERT options for data split (KN or RND), predict type (REG or CLAS) and hyperopt error (RMSE, etc.):

No PFI (standard descriptor filter): PFI (only most important descriptors):

split: KN split: KN type: reg type: reg

error_type: rmse error_type: rmse



Section I. Abbreviations

Reference section for the abbreviations used.

ACC: accuracy KN: k-nearest neighbors **REG:** Regression ADAB: AdaBoost MAE: root-mean-square error RF: random forest

CSV: comma separated values RMSE: root mean square error MCC: Matthew's correl. coefficient

CLAS: classification ML: machine learning RND: random

CV: cross-validation MVL: multivariate lineal models SHAP: Shapley additive explanations

F1 score: balanced F-score NN: neural network VR: voting regressor

GB: gradient boosting PFI: permutation feature importance R2: coefficient of determination GP: gaussian process

ROBERT v 1.2.1 Page 7 of 8

Miscellaneous

General tips to improve the models and instructions to predict new values.

Some general tips to improve the score

1. Adding meaningful datapoints might help to improve the model. Also, using a uniform population of datapoints across the whole range of y values usually helps to obtain reliable predictions across the whole range. More information about the range of y values used is available in Section C.

2. Adding meaningful descriptors or replacing/deleting the least useful descriptors used might help. Feature importances are gathered in Section D.

How to predict new values with these models?

- 1. Create a CSV database with the new points, including the necessary descriptors.
- 2. Place the CSV file in the parent folder (i.e., where the module folders were created)
- 3. Run the PREDICT module as 'python -m robert --predict --csv_test FILENAME.csv'.
- 4. The predictions will be shown at the end of the resulting PDF report and will be stored in the last column of two CSV files called MODEL_SIZE_test(_No)_PFI.csv, which are in the PREDICT folder.

ROBERT v 1.2.1 Page 8 of 8