ROBERT v 1.2.0 2024/09/13 20:45:22

How to cite: Dalmau, D.; Alegre Requena, J. V. ChemRxiv, 2023, DOI: 10.26434/chemrxiv-2023-k994h

Section A. ROBERT Score

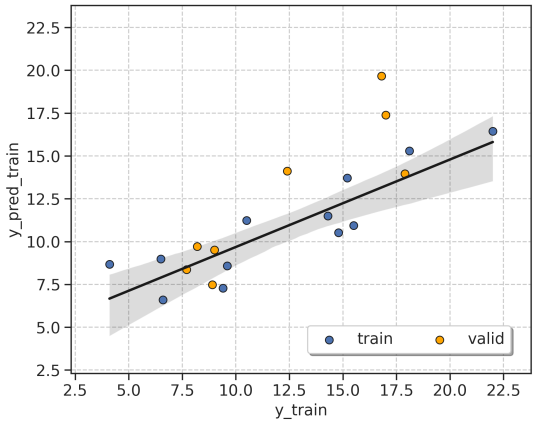
This score is designed to evaluate the models using different metrics.

No PFI (standard descriptor filter):

Model = RF · Train:Validation = 60:40

Points(train+valid.):descriptors = 20:6

WEAK



Train : R^2 = 0.77, MAE = 2.7, RMSE = 3.2

Valid. : R^2 = 0.78, MAE = 1.6, RMSE = 2.0

Severe warnings

- No severe warnings detected

Moderate warnings

- Imprecise predictions (Section B.3b)
- Moderately correlated features (Section D)

Overall assessment

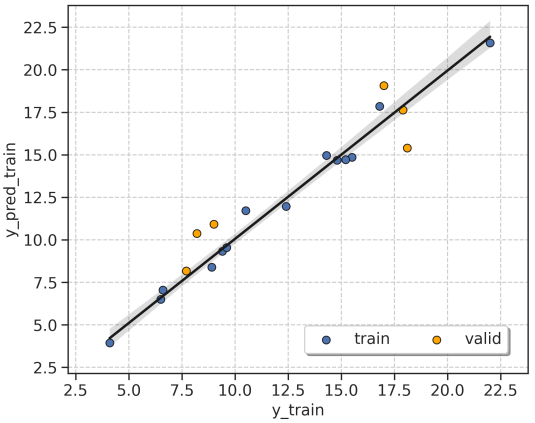
- The model is unreliable

PFI (only most important descriptors):

Model = NN · Train:Validation = 70:30

Points(train+valid.):descriptors = 20:4

MODERATE



Train : R^2 = 0.99, MAE = 0.45, RMSE = 0.57

Valid. : R^2 = 0.87, MAE = 1.6, RMSE = 1.8

Severe warnings

- No severe warnings detected

Moderate warnings

- Imprecise predictions (Section B.3b)

Overall assessment

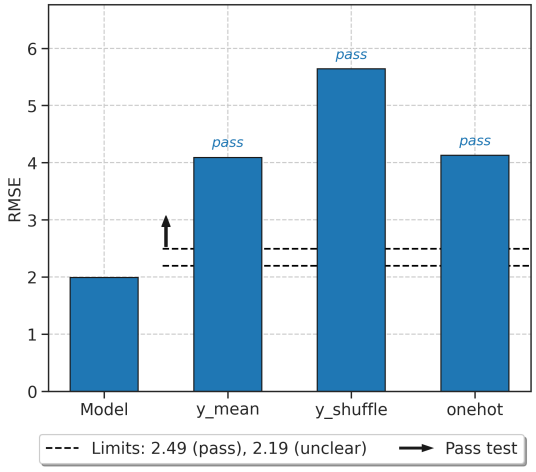
- Decent model, but it has limitations

Section B. Advanced Score Analysis

This section explains each component that comprises the ROBERT score.

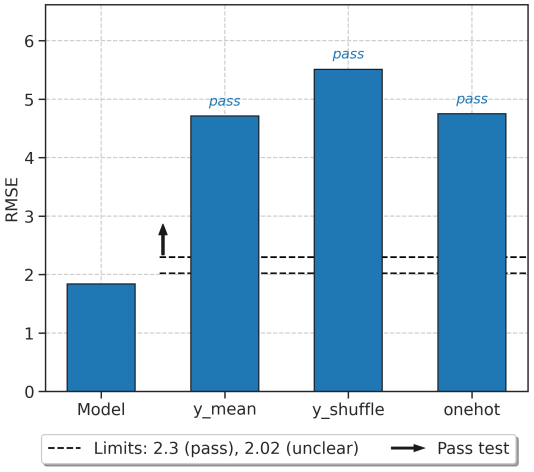
1. Model vs "flawed" models (3 / 3)

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 (1 / 2)

Moderate predictive ability with R^2 (valid.) = 0.78.
R^2 0.70-0.85: +1, R^2 >0.85: +2.

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

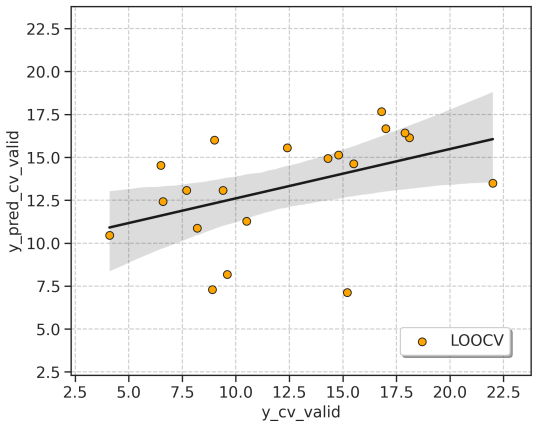
Good predictive ability with R^2 (valid.) = 0.87.
R^2 0.70-0.85: +1, R^2 >0.85: +2.

3. Cross-validation (LOOCV) 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 (LOOCV) = 0.19.
R^2 0.70-0.85: +1, R^2 >0.85: +2.

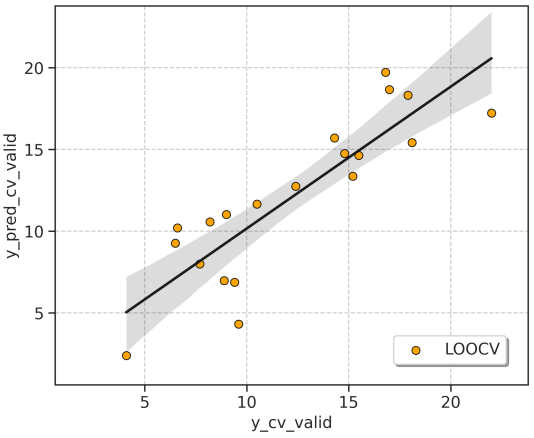


3. Cross-validation (LOOCV) of the model

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

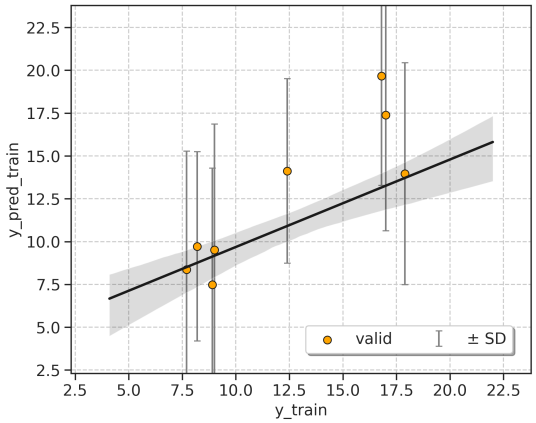
3a. CV predictions train + valid. (1 / 2)

Moderate predictive ability with R^2 (LOOCV) = 0.75.
R^2 0.70-0.85: +1, R^2 >0.85: +2.



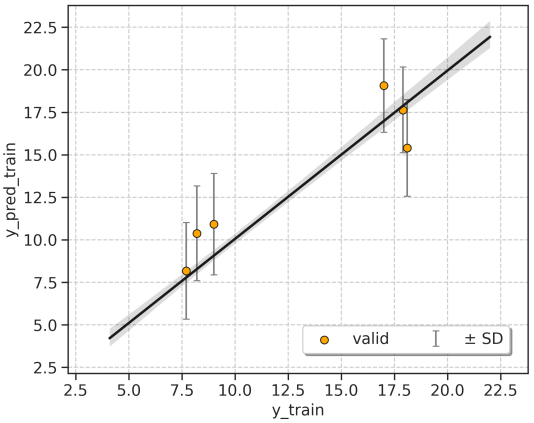
3b. Avg. standard deviation (SD) (0 / 2)

High variation, 4*SD (valid.) = 25.8 (144% y-range).
4*SD 25-50% y-range: +1, 4*SD < 25% y-range: +2.
Details here.



3b. Avg. standard deviation (SD) (0 / 2)

High variation, 4*SD (valid.) = 11.2 (62% y-range).
4*SD 25-50% y-range: +1, 4*SD < 25% y-range: +2.
Details here.



4. Points(train+valid.):descriptors (0 / 1)

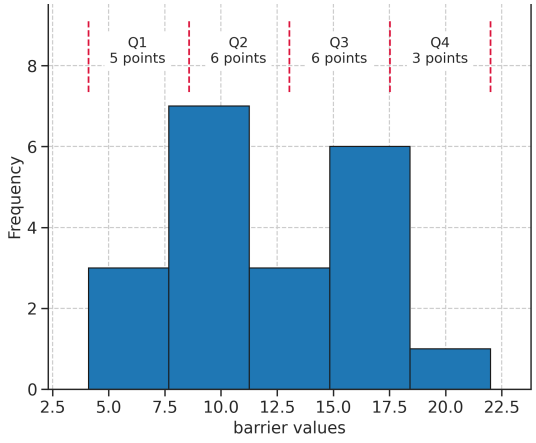
Number of descps. could be lower (ratio 20:6).
5 or more points per descriptor: +1.

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

Decent number of descps. (ratio 20:4).
5 or more points per descriptor: +1.

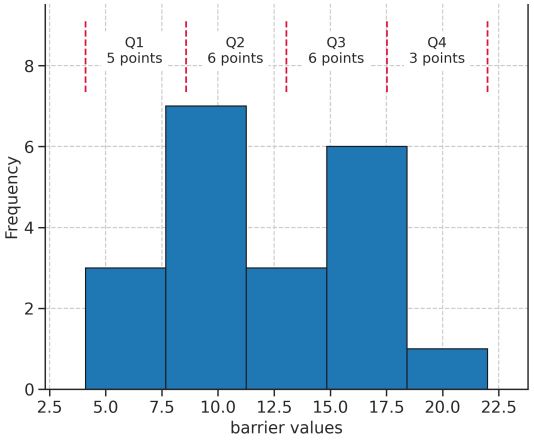
Section C. Distribution of y Values

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



y distribution analysis

o Your data seems quite uniform

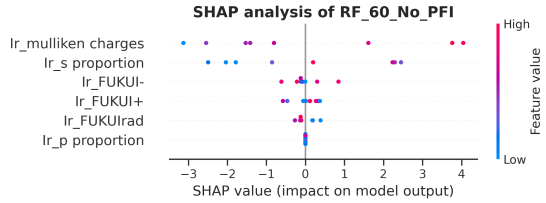


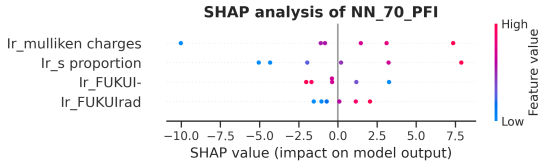
y distribution analysis

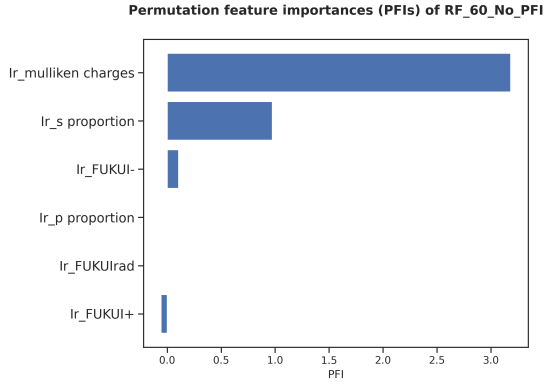
o Your data seems quite uniform

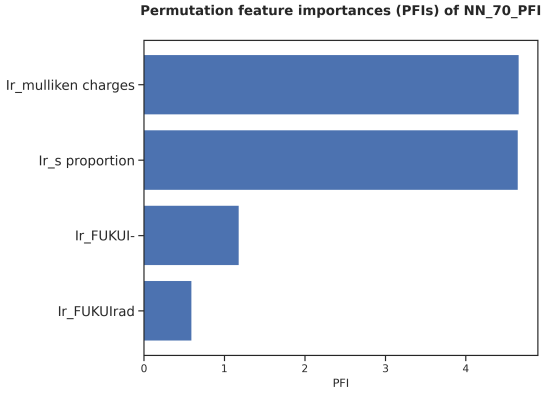
Section D. Feature Importances

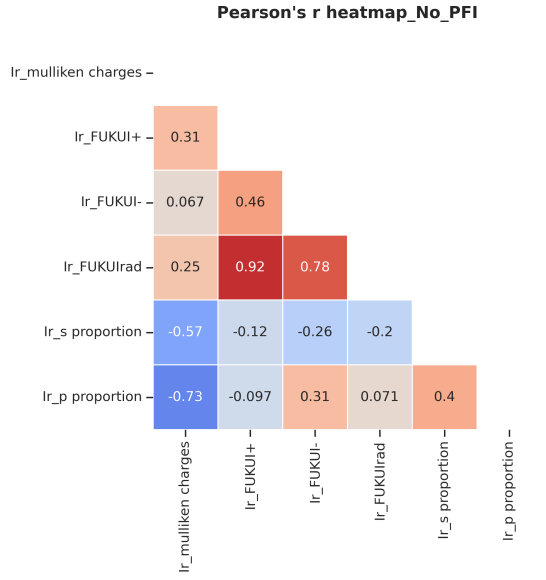
This section presents feature importances measured using the validation set.

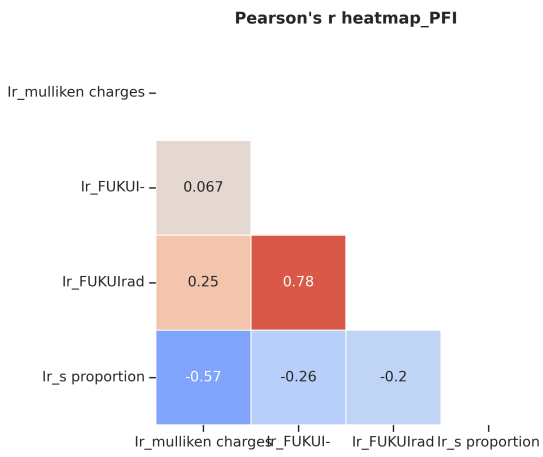












Correlation analysis

x WARNING! Noticeable correlations observed (up to r = 0.92 or R^2 = 0.84, for lr_FUKUI+ and lr_FUKUIrad)

Correlation analysis

o Correlations between variables are acceptable

Section E. Outlier Analysis

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

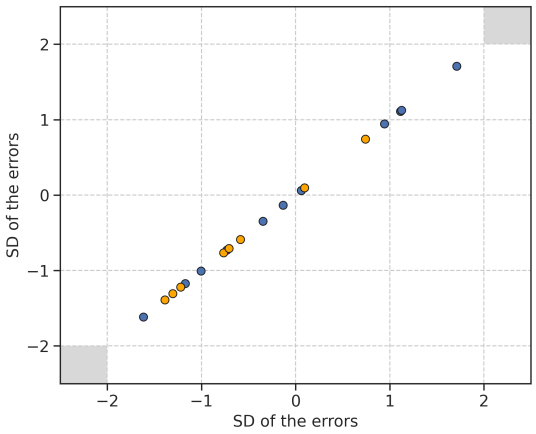
No PFI (standard descriptor filter):

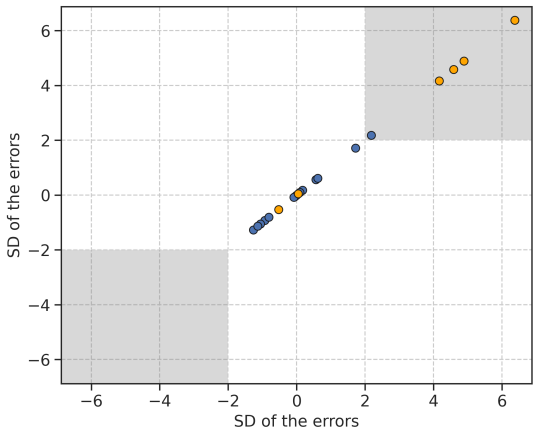
Outliers (max. 10 shown)
Train: 0 outliers out of 12 datapoints (0.0%)
Validation: 0 outliers out of 8 datapoints (0.0%)

PFI (only most important descriptors):

Outliers (max. 10 shown)
Train: 1 outliers out of 14 datapoints (7.1%)
- ir_tbp_1_dft-pyz_1_dft-pme3_1_dft-hicn_1_chloride_1_smi1_1_s_1 (2.2 SDs)

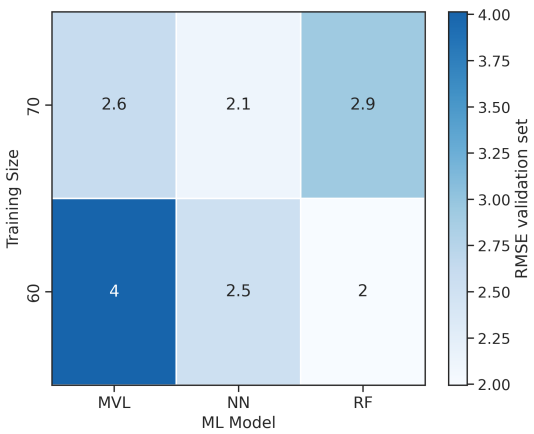
Validation: 4 outliers out of 6 datapoints (66.7%)
- ir_tbp_1_dft-py_1_dft-sime_1_dft-hicn_1_dft-no2_1_smi1_1_s_1 (4.2 SDs)
- ir_tbp_1_dft-pet3_1_dft-ime_1_dft-co_1_dft-cch_1_smi1_1_s_1 (4.9 SDs)
- ir_tbp_1_dft-pyz_1_dft-sime_1_dft-hicn_1_dft-icn_1_smi1_1_s_1 (4.6 SDs)
- ir_tbp_1_dft-ime_1_dft-nme3_1_dft-hicn_1_dft-icn_1_smi1_1_s_1 (6.4 SDs)

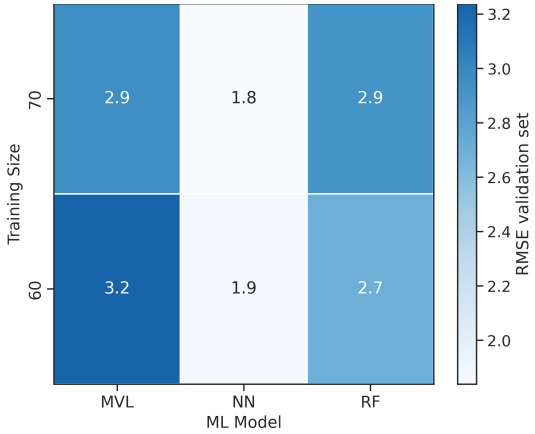




Section F. Model Screening

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 (vaska_short.csv)

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

- Install ROBERT and its dependencies: conda install -c conda-forge robert
- Adjust ROBERT version: pip install robert==1.2.0
- Install scikit-learn-intelex: pip install scikit-learn-intelex==2024.5.0

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

- Install AQME and its dependencies: conda install -c conda-forge aqme
- Adjust AQME version: pip install aqme==1.6.1
- Install xTB: conda install -c conda-forge xtb
- Adjust xTB version (if possible): conda install -c conda-forge xtb=6.6.1

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

python -m robert --aqme --y "barrier" --csv_name "vaska_short.csv" --qdescp_keywords "--qdescp_atoms ['Ir']"

4. Execution time, Python version and OS:

Originally run in Python 3.10.13 using Linux #1 SMP Fri Mar 29 23:14:13 UTC 2024

Total execution time: 175.52 seconds *(the number of processors should be specified by the user)*

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):

sklearn model: RandomForestRegressor
random_state: 43
names: code_name
n_estimators: 5
max_depth: 40
max_features: 0.25
min_samples_split: 2
min_samples_leaf: 1
min_weight_fraction_leaf: 0
ccp_alpha: 0
oob_score: False
max_samples: 0.75

PFI (only most important descriptors):

sklearn model: MLPRegressor
random_state: 19
names: code_name
batch_size: 4
hidden_layer_sizes: [16, 16]
learning_rate_init: 0.01
max_iter: 50
validation_fraction: 0.3
alpha: 0.0001
shuffle: False
tol: 0.0001
early_stopping: False
beta_1: 0.8
beta_2: 0.999
epsilon: 1e-08

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

No PFI (standard descriptor filter):

split: KN
type: reg
error_type: rmse

PFI (only most important descriptors):

split: KN
type: reg
error_type: rmse

Section I. Abbreviations

Reference section for the abbreviations used.

ACC: accuracy
ADAB: AdaBoost
CSV: comma separated values
CLAS: classification
CV: cross-validation
F1 score: balanced F-score
GB: gradient boosting
GP: gaussian process
KN: k-nearest neighbors
MAE: root-mean-square error
MCC: Matthew's correl. coefficient
ML: machine learning
MVL: multivariate lineal models
NN: neural network
PFI: permutation feature importance
R2: coefficient of determination
REG: Regression
RF: random forest
RMSE: root mean square error
RND: random
SHAP: Shapley additive explanations
VR: voting regressor

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.