

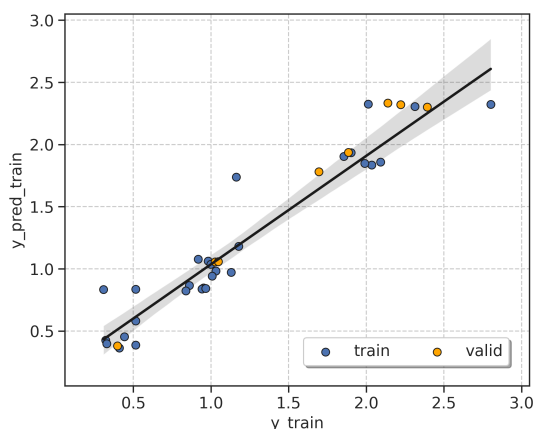


ROBERT v 1.2.0 2024/09/13 20:16:42

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 = NN · Train:Validation = 78:22

Points(train+valid.):descriptors = 37:12

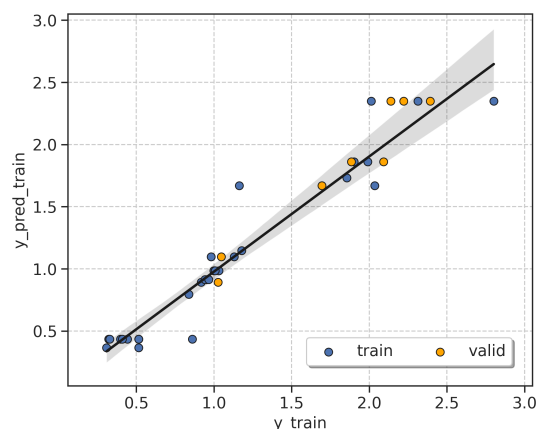
**MODERATE**

Train : $R^2 = 0.9$, MAE = 0.14, RMSE = 0.21
Valid. : $R^2 = 0.99$, MAE = 0.073, RMSE = 0.092

PFI (only most important descriptors):

Model = NN · Train:Validation = 78:22

Points(train+valid.):descriptors = 37:4

**STRONG**

Train : $R^2 = 0.92$, MAE = 0.12, RMSE = 0.19
Valid. : $R^2 = 0.94$, MAE = 0.11, RMSE = 0.13

Severe warnings

- ☒ No severe warnings detected

Moderate warnings

- ☐ Slightly uneven y distribution (Section C)
- ☐ Moderately correlated features (Section D)

Overall assessment

- ☐ Decent model, but it has limitations

Severe warnings

- ☒ No severe warnings detected

Moderate warnings

- ☐ Slightly uneven y distribution (Section C)

Overall assessment

- ☒ The model seems reliable



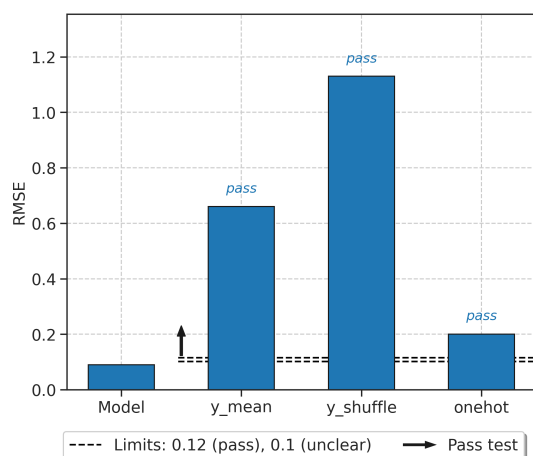
Section B. Advanced Score Analysis

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.

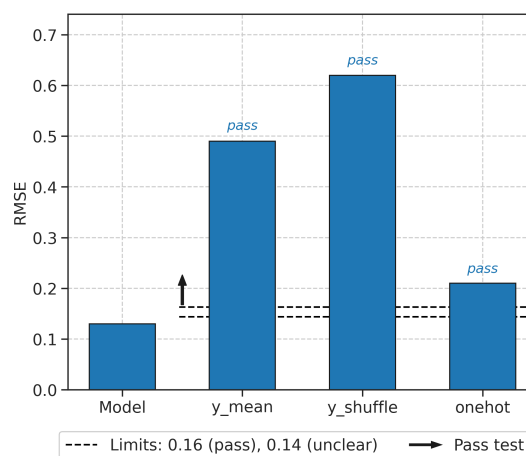
Pass: +1, Unclear: 0, Fail: -1. [Details here.](#)



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 (2 / 2)

Good predictive ability with R^2 (valid.) = 0.99.

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.) = 0.94.

R^2 0.70-0.85: +1, R^2 >0.85: +2.

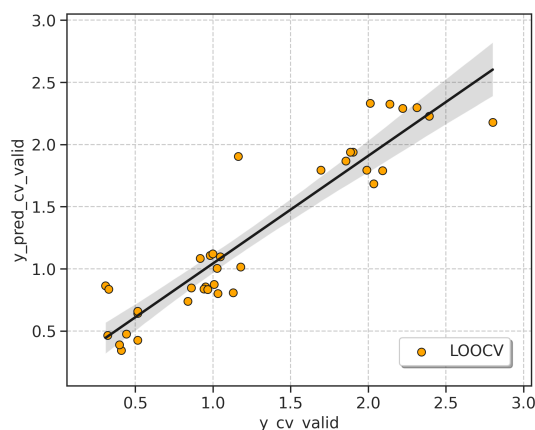
3. Cross-validation (LOOCV) of the model

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

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

Good predictive ability with R^2 (LOOCV) = 0.87.

R^2 0.70-0.85: +1, R^2 >0.85: +2.



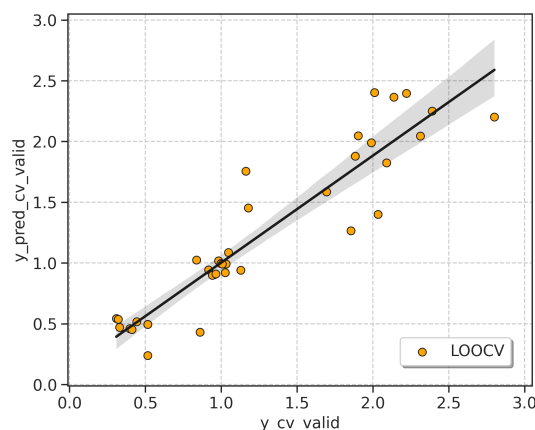
3. Cross-validation (LOOCV) of the model

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

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

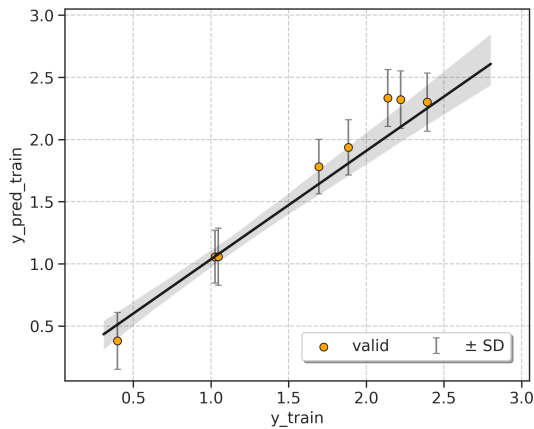
Good predictive ability with R^2 (LOOCV) = 0.86.

R^2 0.70-0.85: +1, R^2 >0.85: +2.



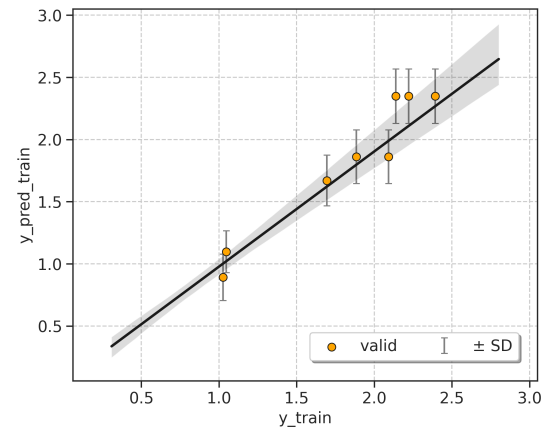
3b. Avg. standard deviation (SD) (1 / 2)

Moderate variation, 4*SD (valid.) = 0.9 (37% y-range).
4*SD 25-50% y-range: +1, 4*SD < 25% y-range: +2.
[Details here.](#)



3b. Avg. standard deviation (SD) (1 / 2)

Moderate variation, 4*SD (valid.) = 0.8 (34% 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 37:12).
5 or more points per descriptor: +1.

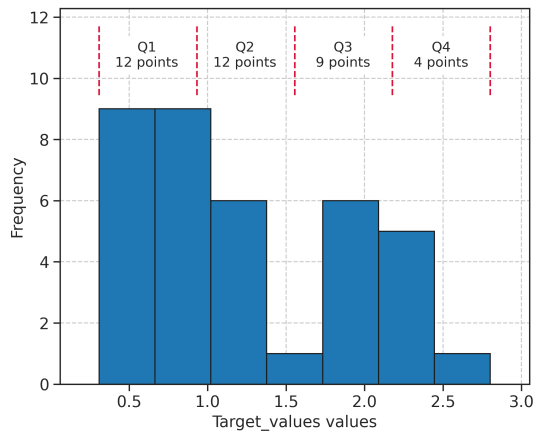
4. Points(train+valid.):descriptors (1 / 1)

Decent number of descps. (ratio 37:4).
5 or more points per descriptor: +1.



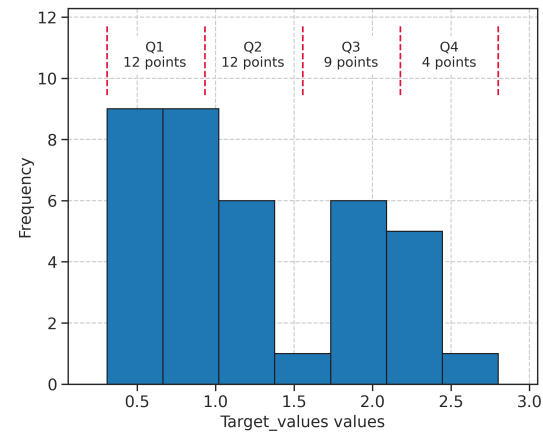
Section C. Distribution of y Values

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



y distribution analysis

x WARNING! Your data is slightly not uniform (Q4 has 4 points while Q1 has 12)



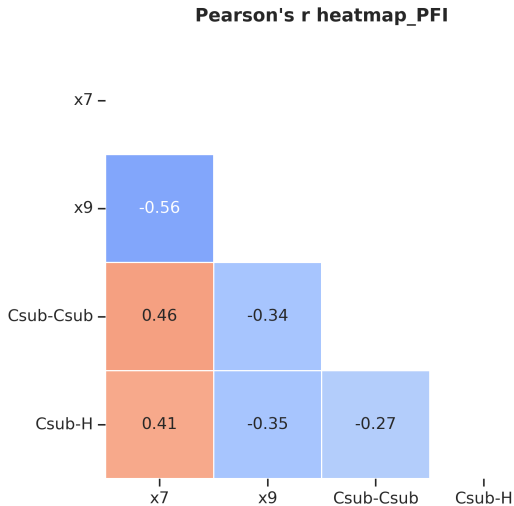
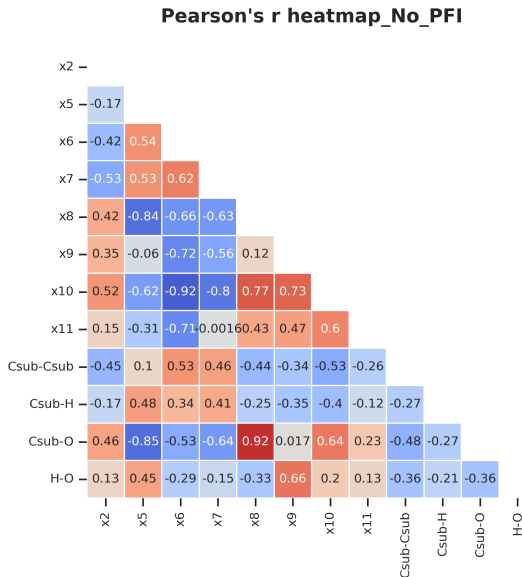
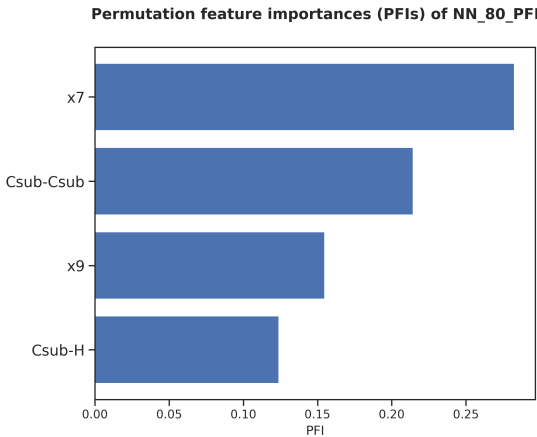
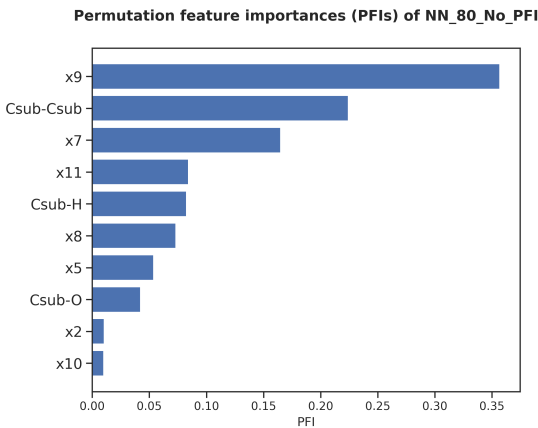
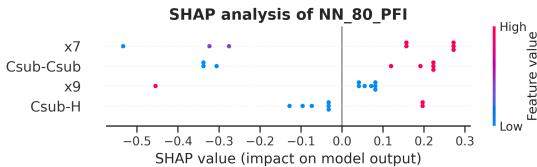
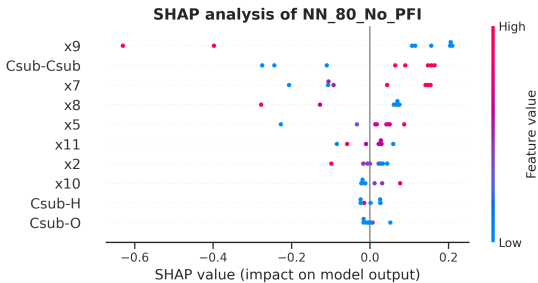
y distribution analysis

x WARNING! Your data is slightly not uniform (Q4 has 4 points while Q1 has 12)



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 $R^2 = 0.85$, for x6 and x10)

Correlation analysis

o Correlations between variables are acceptable

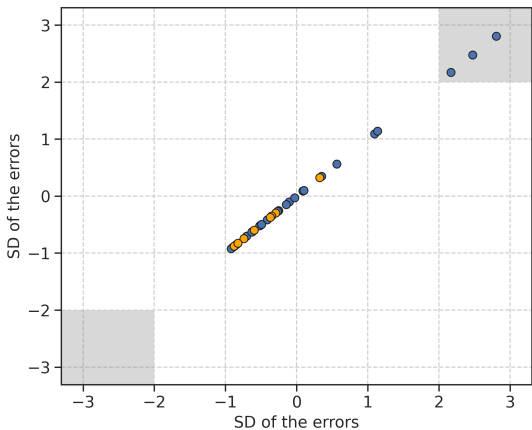


Section E. Outlier Analysis

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

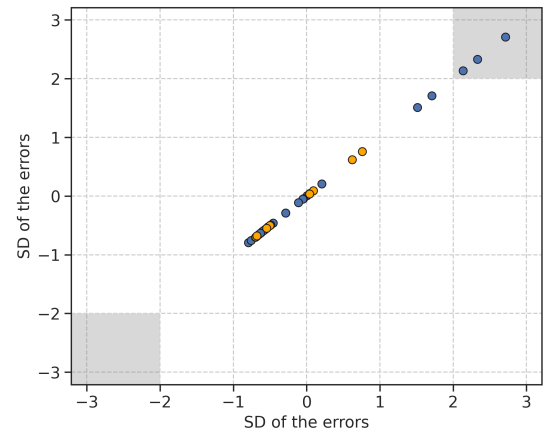
No PFI (standard descriptor filter):

Outliers (max. 10 shown)
Train: 3 outliers out of 29 datapoints (10.3%)
- 6 (2.5 SDs)
- 19 (2.8 SDs)
- 21 (2.2 SDs)
Validation: 0 outliers out of 8 datapoints (0.0%)



PFI (only most important descriptors):

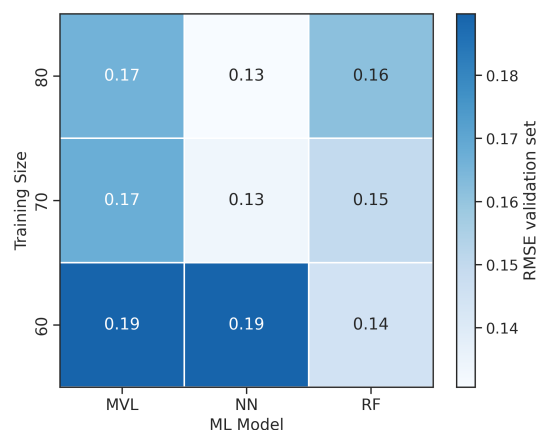
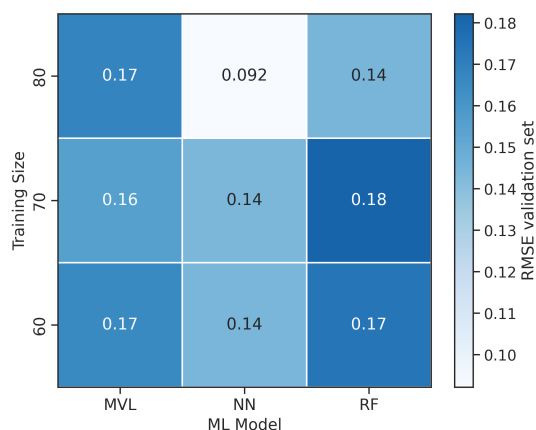
Outliers (max. 10 shown)
Train: 3 outliers out of 29 datapoints (10.3%)
- 3 (2.1 SDs)
- 19 (2.7 SDs)
- 21 (2.3 SDs)
Validation: 0 outliers out of 8 datapoints (0.0%)





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 (Robert_example.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)

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

```
python -m robert --ignore "[Name]" --names "Name" --y "Target_values" --csv_name "Robert_example.csv"
```

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: 101.99 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: MLPRegressor
 random_state: 19
 names: Name
 batch_size: 32
 hidden_layer_sizes: [8, 8, 8]
 learning_rate_init: 0.001
 max_iter: 200
 validation_fraction: 0.3
 alpha: 0.0001
 shuffle: True
 tol: 0.0001
 early_stopping: False
 beta_1: 0.9
 beta_2: 0.999
 epsilon: 1e-08

PFI (only most important descriptors):

sklearn model: MLPRegressor
 random_state: 70
 names: Name
 batch_size: 4
 hidden_layer_sizes: [16, 16]
 learning_rate_init: 0.01
 max_iter: 50
 validation_fraction: 0.2
 alpha: 0.0001
 shuffle: True
 tol: 0.0001
 early_stopping: False
 beta_1: 0.9
 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

**Section J. New Predictions**

Predictions of the external test set added with the csv_test option.

csv_test predictions (sorted, max. 20 shown)

From /PREDICT/csv_test/...No_PFI.csv

Name	Target_va..._pred ± sd
46	1.93 ± 0.22
38	1.9 ± 0.23
39	1.84 ± 0.23
42	1.18 ± 0.25
41	1.06 ± 0.21
40	0.87 ± 0.23
44	0.42 ± 0.24
45	0.4 ± 0.22
43	0.36 ± 0.24

csv_test predictions (sorted, max. 20 shown)

From /PREDICT/csv_test/..._PFI.csv

Name	Target_va..._pred ± sd
46	1.86 ± 0.22
38	1.73 ± 0.17
39	1.67 ± 0.2
42	1.15 ± 0.21
41	0.89 ± 0.19
45	0.44 ± 0.2
44	0.44 ± 0.2
43	0.44 ± 0.2
40	0.44 ± 0.2

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.
-