ROBERT v 1.2.0 2024/10/08 14:30:03

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.

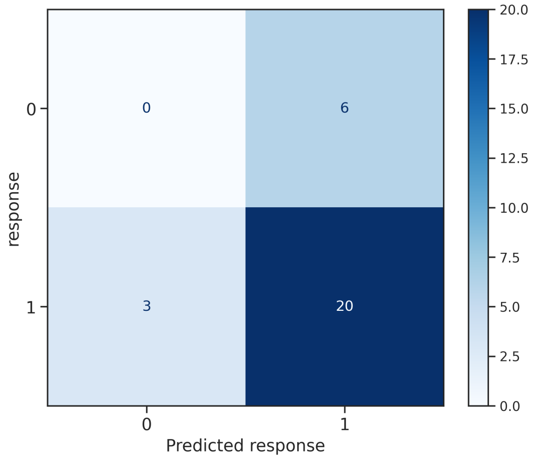
No PFI (standard descriptor filter):

Model = NN · Train:Validation:Test = 81:9:10

Points(train+valid.):descriptors = 268:68

🟨🟨🟨🟨⬜⬜⬜⬜⬜⬜

WEAK



Train : Accuracy = 0.85, F1 score = 0.91, MCC = 0.54
Valid. : Accuracy = 0.85, F1 score = 0.92, MCC = 0.25
Test : Accuracy = 0.69, F1 score = 0.82, MCC = -0.17

Severe warnings

● Very uneven class distribution (Section C)

Moderate warnings

● Moderately correlated features (Section D)

Overall assessment

● The model is unreliable

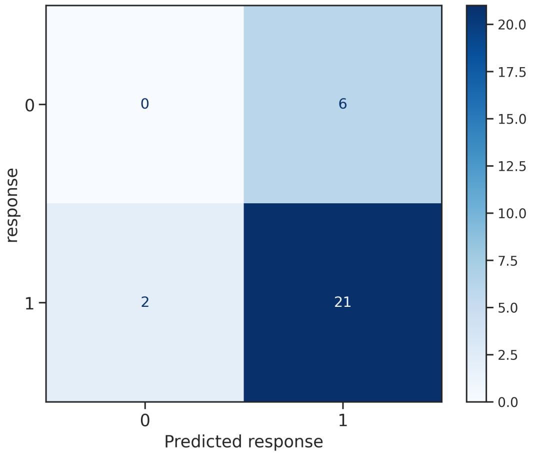
PFI (only most important descriptors):

Model = NN · Train:Validation:Test = 72:18:10

Points(train+valid.):descriptors = 268:23

🟨🟨🟨🟨⬜⬜⬜⬜⬜⬜

WEAK



Train : Accuracy = 0.85, F1 score = 0.91, MCC = 0.52
Valid. : Accuracy = 0.83, F1 score = 0.91, MCC = 0.18
Test : Accuracy = 0.72, F1 score = 0.84, MCC = -0.14

Severe warnings

● Failing required tests (Section B.1)

● Very uneven class distribution (Section C)

Moderate warnings

● Moderately correlated features (Section D)

Overall assessment

● The model is unreliable

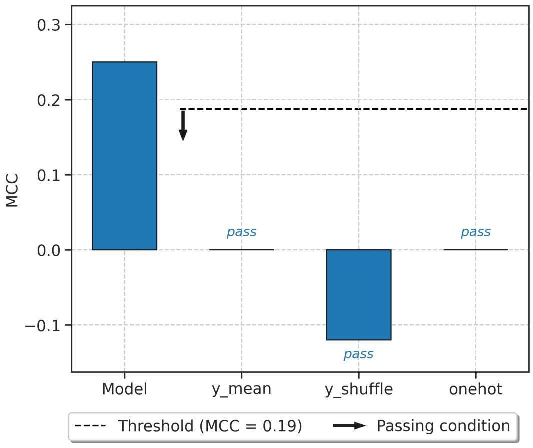
Section B. Advanced Score Analysis

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 (blue): +1, Fail (red): -1. Details here.



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

Low predictive ability with MCC (test) = -0.17.

MCC 0.50-0.75: +1, MCC >0.75: +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 ⬜⬜)

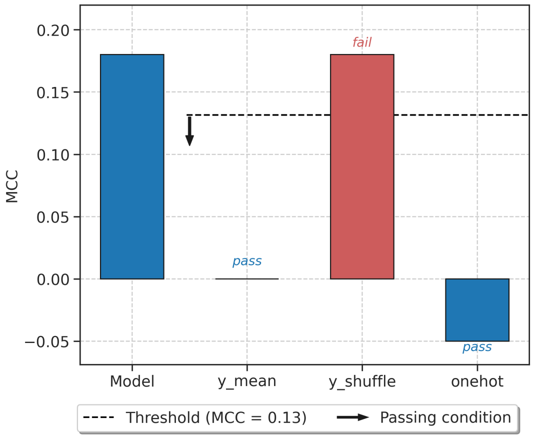
Low predictive ability with MCC (5-fold CV) = 0.06.

MCC 0.50-0.75: +1, MCC >0.75: +2.

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

WARNING! The model might have important flaws.

Pass (blue): +1, Fail (red): -1. Details here.



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

Low predictive ability with MCC (test) = -0.14.

MCC 0.50-0.75: +1, MCC >0.75: +2.

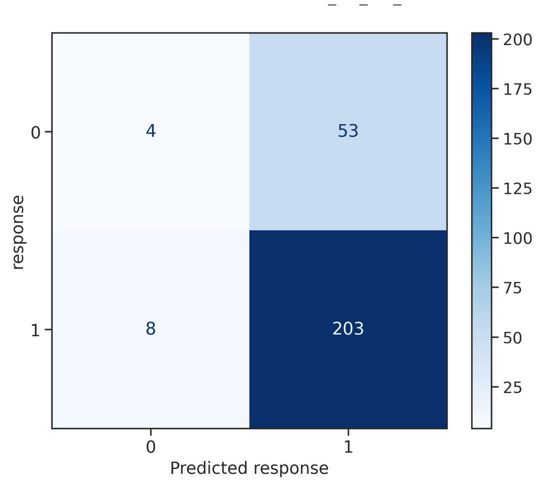
3. Cross-validation (5-fold CV) of the model

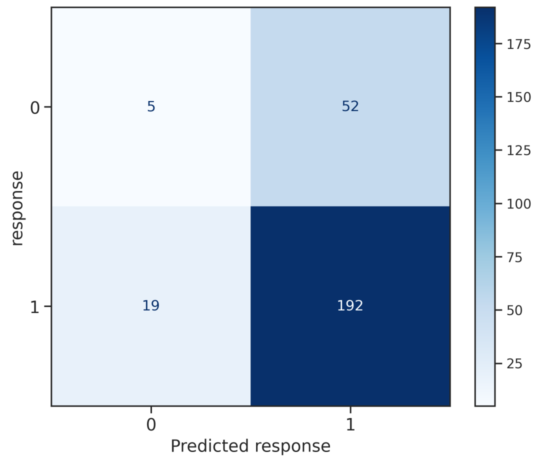
Overfitting analysis on the model with 3a and 3b:

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

Low predictive ability with MCC (5-fold CV) = -0.0.

MCC 0.50-0.75: +1, MCC >0.75: +2.





3b. MCC difference (model vs CV) (1 / 2 🟨⬜)

Moderate variation (test and CV), ΔMCC = 0.23. ΔMCC 0.15-0.30: +1, ΔMCC < 0.15: +2.

3b. MCC difference (model vs CV) (2 / 2 🟦🟦)

Low variation (test and CV), ΔMCC = 0.14. ΔMCC 0.15-0.30: +1, ΔMCC < 0.15: +2.

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

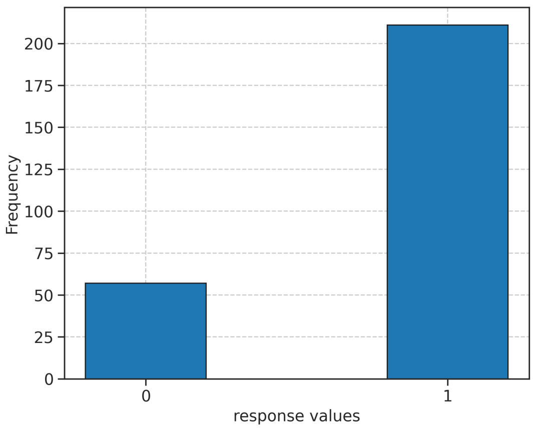
Number of descps. could be lower (ratio 268:68). 5 or more points per descriptor: +1.

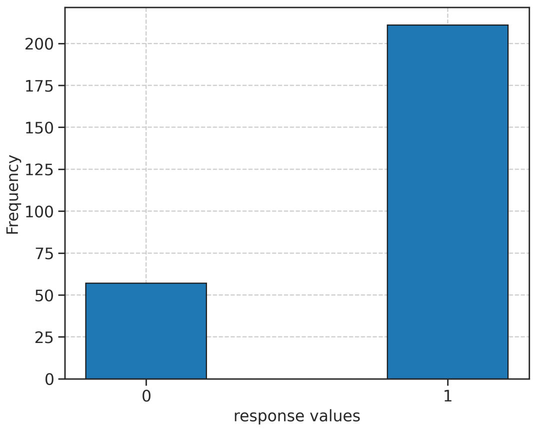
4. Points(train+valid.):descriptors (1 / 1 🟦)

Decent number of descps. (ratio 268:23). 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

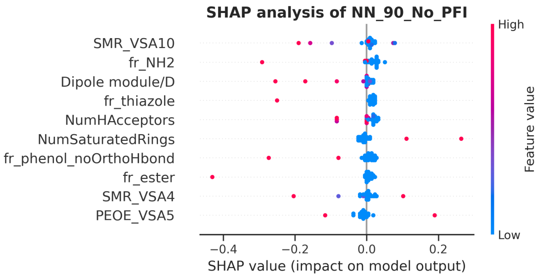
x WARNING! Your data is not uniform (class 0 has 57 points while class 1 has 211)

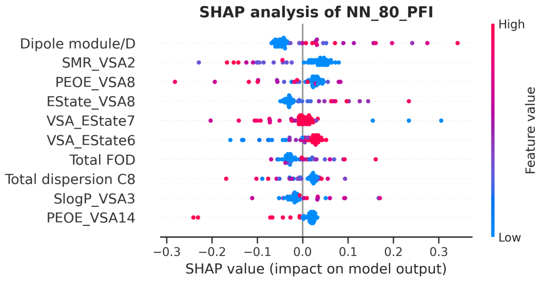
y distribution analysis

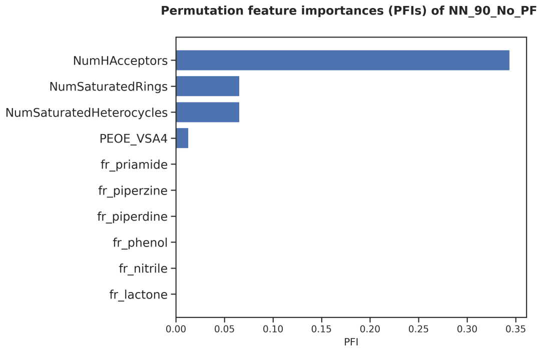
x WARNING! Your data is not uniform (class 0 has 57 points while class 1 has 211)

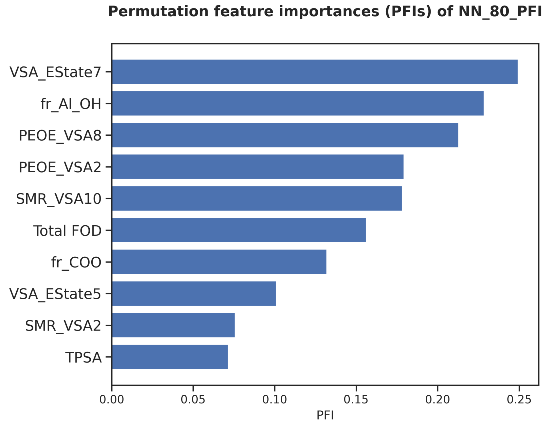
Section D. Feature Importances

This section presents feature importances measured using the validation set.

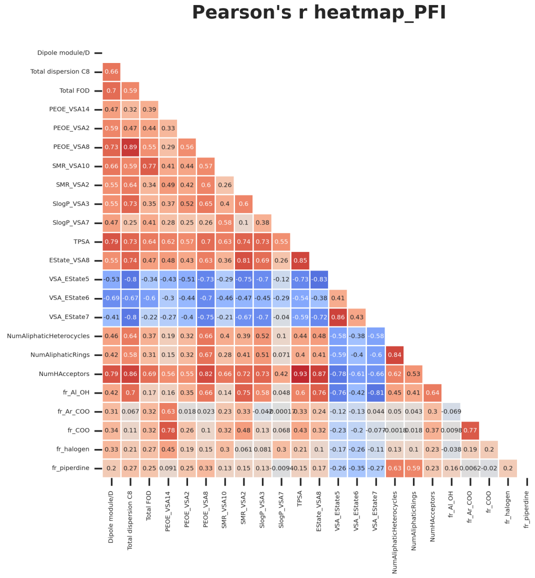

SHAP analysis of NN_90_No_PFI


SHAP analysis of NN_80_PFI


Permutation feature importances (PFIs) of NN_90_No_PFI


Permutation feature importances (PFIs) of NN_80_PFI

Pearson maps not created if >30 descriptors.


Pearson's r heatmap_PFI

Correlation analysis

x WARNING! Noticeable correlations observed (up to r = 0.94 or R^2 = 0.89, for Chi4v and RingCount)

Correlation analysis

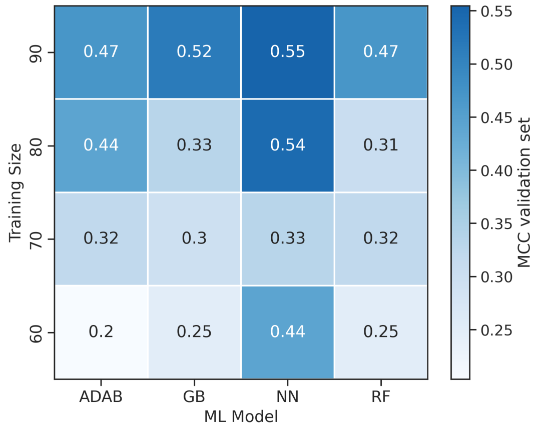
x WARNING! Noticeable correlations observed (up to 0.93 or R^2 = 0.87, for TPSA and NumHAcceptors)

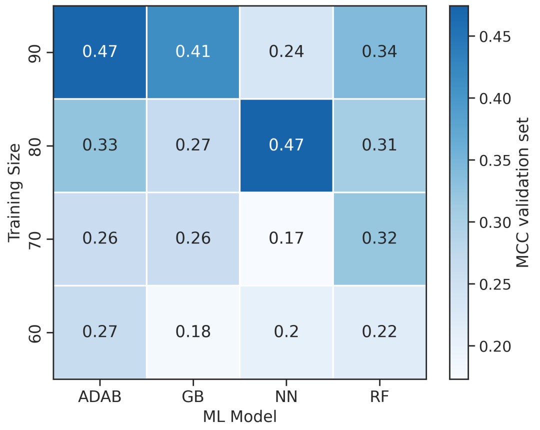
Section E. Outlier Analysis

This feature is disabled in classification problems.

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 (lipros_data.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
- scikit-learn-intelex: not installed

(if scikit-learn-intelex is 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 --csv_name "lipros_data.csv" --y "response" --names "code_name" --aqme --type "clas"
```

4. Execution time, Python version and OS:

Originally run in Python 3.12.5 using Linux #3672-Microsoft Fri Jan 01 08:00:00 PST 2016

Total execution time: 1450.72 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):	PFI (only most important descriptors):
sklearn model: MLPClassifier	sklearn model: MLPClassifier
random_state: 70	random_state: 233
names: code_name	names: code_name
batch_size: 32	batch_size: 4
hidden_layer_sizes: [8, 8, 8]	hidden_layer_sizes: [16, 16]
learning_rate_init: 0.01	learning_rate_init: 0.01
max_iter: 50	max_iter: 200
validation_fraction: 0.1	validation_fraction: 0.2
alpha: 0.0001	alpha: 0.0001
shuffle: True	shuffle: True
tol: 0.0001	tol: 0.0001
early_stopping: False	early_stopping: False
beta_1: 0.999	beta_1: 0.999
beta_2: 0.999	beta_2: 0.999
epsilon: 1e-08	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):	PFI (only most important descriptors):
split: RND	split: RND
type: clas	type: clas
error_type: mcc	error_type: mcc

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.