ROBERT v 2.0.1 2025/04/15 11:09:05

How to cite: Dalmau, D.; Alegre Requena, J. V. WIREs Comput Mol Sci. 2024, 14, e1733.

Section A. ROBERT Score

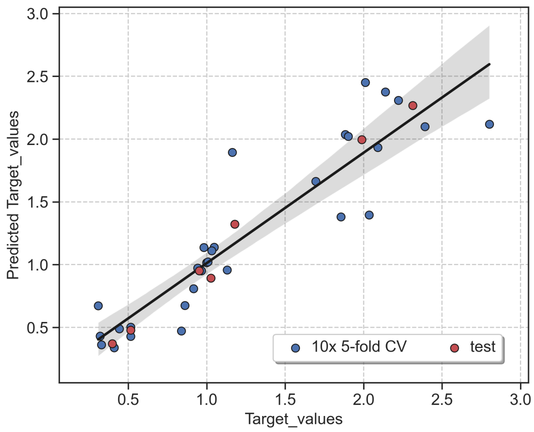
This score is designed to evaluate the models using different metrics.

No PFI (standard descriptor filter) · Score 5

Model = NN · CV (train+valid.):Test = 81:19

Points(train+validation):descriptors = 30:9

WEAK



10x 5-fold CV : R^2 = 0.81, MAE = 0.23, RMSE = 0.31

Test : R^2 = 0.99, MAE = 0.056, RMSE = 0.078

Severe warnings

- Failing required tests (Section B.1)

Moderate warnings

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

Overall assessment

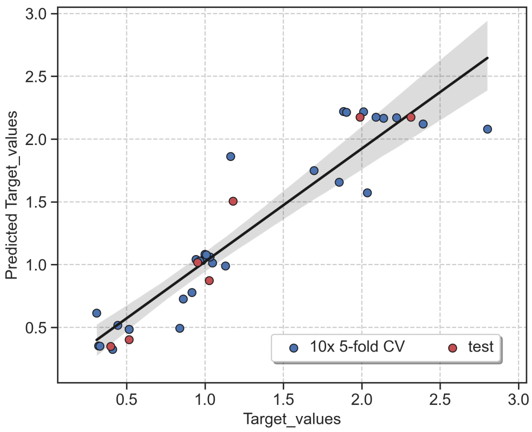
- The model is unreliable

PFI (only important descriptors) · Score 6

Model = NN · CV (train+valid.):Test = 81:19

Points(train+validation):descriptors = 30:3

WEAK



10x 5-fold CV : R^2 = 0.86, MAE = 0.18, RMSE = 0.26

Test : R^2 = 0.94, MAE = 0.15, RMSE = 0.17

Severe warnings

- No severe warnings detected

Moderate warnings

- Some tests are unclear (Section B.1)
- Slightly uneven y distribution (Section C)
- Moderately correlated features (Section D)

Overall assessment

- Moderate model, with important limitations

Section B. Advanced Score Analysis

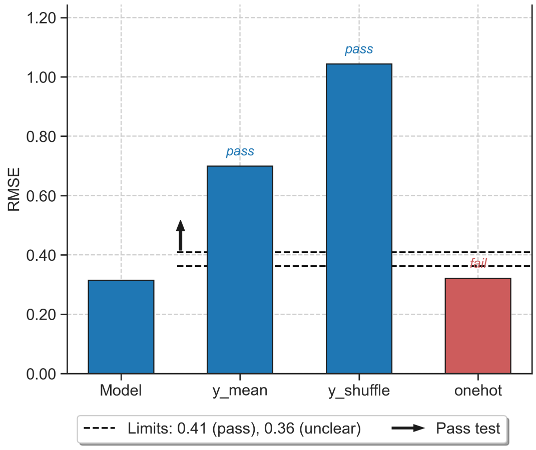
This section explains each component that comprises the ROBERT score. *[More details here.]*

1. Model vs "flawed" models (-2 / 0)

Warning! The model probably has important flaws.

· *Scoring from -6 to 0* ·

Pass: 0, Unclear: -1, Fail: -2.

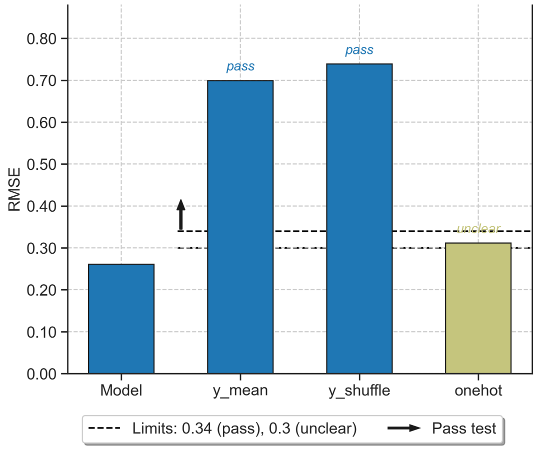


1. Model vs "flawed" models (-1 / 0)

Warning! The model probably has important flaws.

· *Scoring from -6 to 0* ·

Pass: 0, Unclear: -1, Fail: -2.



2. CV predictions of the model (1 / 2 🟨⬜)

Scaled RMSE (10x 5-fold CV) = 12.4%.

R^2 (10x 5-fold CV) = 0.81.

· *Scoring from 0 to 2* ·

Scaled RMSE ≤ 10%: +2, Scaled RMSE ≤ 20%: +1.

$R^2 < 0.5$: -2, $R^2 < 0.7$: -1

2. CV predictions of the model (1 / 2 🟨⬜)

Scaled RMSE (10x 5-fold CV) = 10.4%.

R^2 (10x 5-fold CV) = 0.86.

· *Scoring from 0 to 2* ·

Scaled RMSE ≤ 10%: +2, Scaled RMSE ≤ 20%: +1.

$R^2 < 0.5$: -2, $R^2 < 0.7$: -1

3. Predictive ability & overfitting

3a. Predictions test set (2 / 2 🟦🟦)

Scaled RMSE (test set) = 3.12%.

R^2 (test set) = 0.99.

· *Scoring from 0 to 2* ·

Scaled RMSE ≤ 10%: +2, Scaled RMSE ≤ 20%: +1.

$R^2 < 0.5$: -2, $R^2 < 0.7$: -1

3b. Prediction accuracy test vs CV (2 / 2 🟦🟦)

Relative differences in values from sections 2 and 3a.

RMSE in test is 0.25*scaled RMSE (CV).

· *Scoring from 0 to 2* ·

Scaled RMSE (test) ≤ 1.25*scaled RMSE (CV): +2.

Scaled RMSE (test) ≤ 1.50*scaled RMSE (CV): +1.

3. Predictive ability & overfitting

3a. Predictions test set (2 / 2 🟦🟦)

Scaled RMSE (test set) = 6.8%.

R^2 (test set) = 0.94.

· *Scoring from 0 to 2* ·

Scaled RMSE ≤ 10%: +2, Scaled RMSE ≤ 20%: +1.

$R^2 < 0.5$: -2, $R^2 < 0.7$: -1

3b. Prediction accuracy test vs CV (2 / 2 🟦🟦)

Relative differences in values from sections 2 and 3a.

RMSE in test is 0.65*scaled RMSE (CV).

· *Scoring from 0 to 2* ·

Scaled RMSE (test) ≤ 1.25*scaled RMSE (CV): +2.

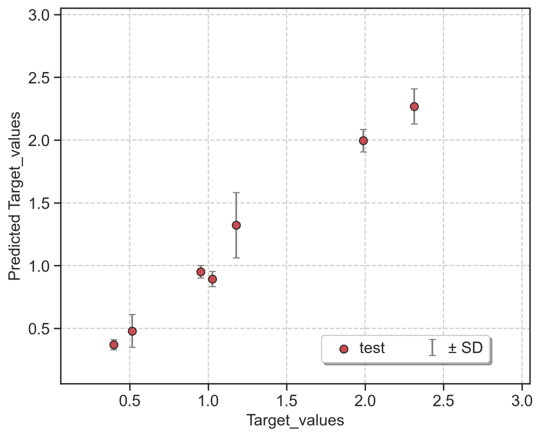
Scaled RMSE (test) ≤ 1.50*scaled RMSE (CV): +1.

3c. Avg. standard deviation (SD) (2 / 2 ▰▰)

Low variation, 4*SD = 0.4 (18% y-range).

· *Scoring from 0 to 2* ·

4*SD ≤ 25% y-range: +2, 4*SD ≤ 50% y-range: +1.



3d. Extrapolation (sorted CV) (0 / 2 ▱▱)

Scaled RMSEs across 5-fold CV:

[9.2%, 14.8%, 12.4%, 17.2%, 18.4%]

· *Scoring from 0 to 2* ·

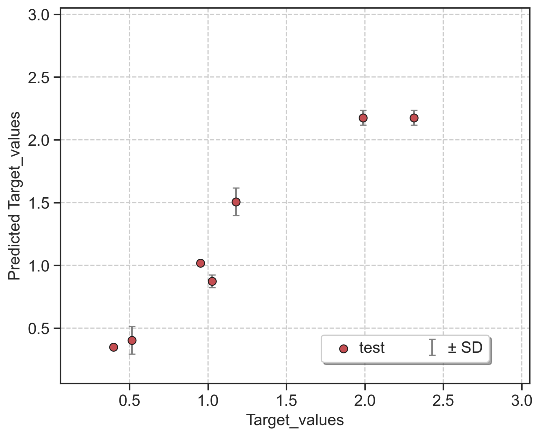
Every two folds with RMSEs ≤ 1.25*min RMSE: +1.

3c. Avg. standard deviation (SD) (2 / 2 ▰▰)

Low variation, 4*SD = 0.2 (10% y-range).

· *Scoring from 0 to 2* ·

4*SD ≤ 25% y-range: +2, 4*SD ≤ 50% y-range: +1.



3d. Extrapolation (sorted CV) (0 / 2 ▱▱)

Scaled RMSEs across 5-fold CV:

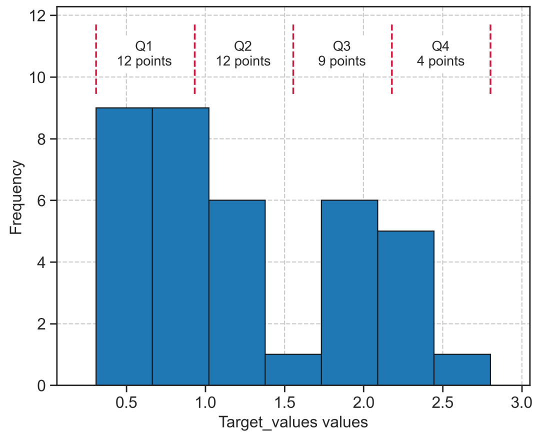
[7.6%, 8.0%, 11.2%, 18.8%, 19.2%]

· *Scoring from 0 to 2* ·

Every two folds with RMSEs ≤ 1.25*min RMSE: +1.

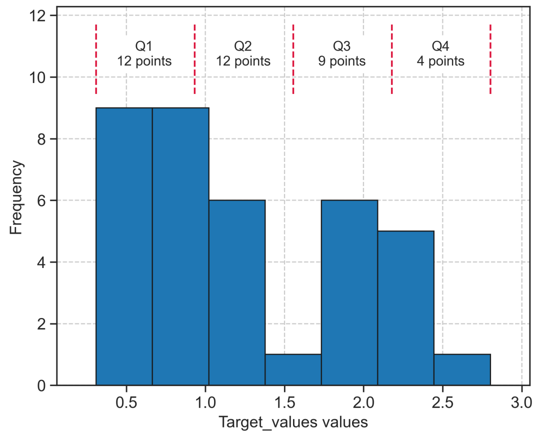
Section C. Distribution of y Values

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)

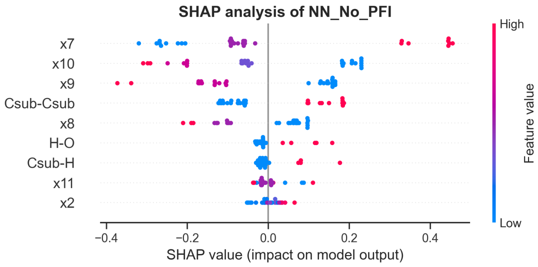


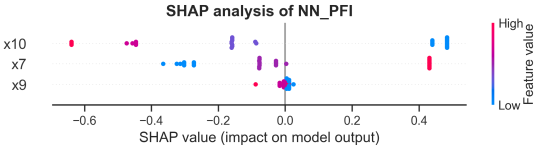
y distribution analysis

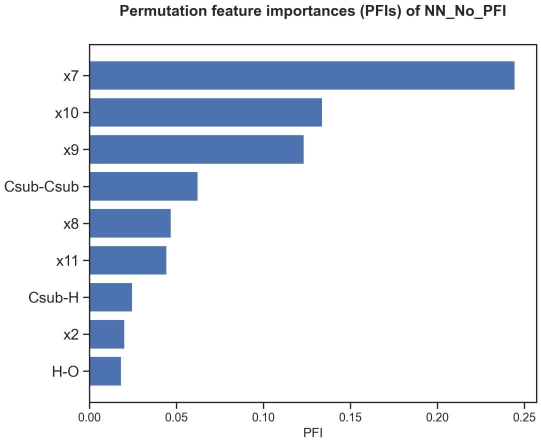
x WARNING! Your data is slightly not uniform (Q4 has 4 points while Q1 has 12)

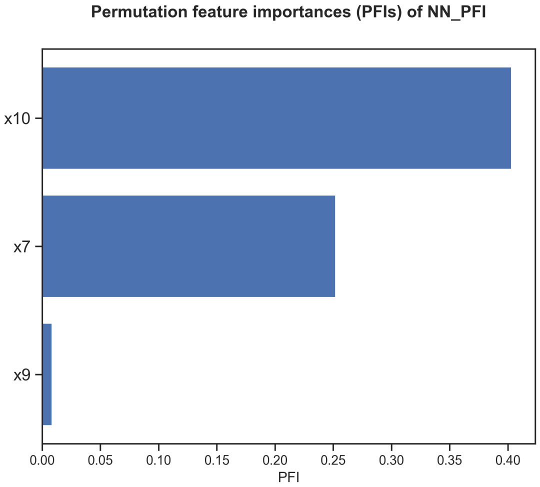
Section D. Feature Importances

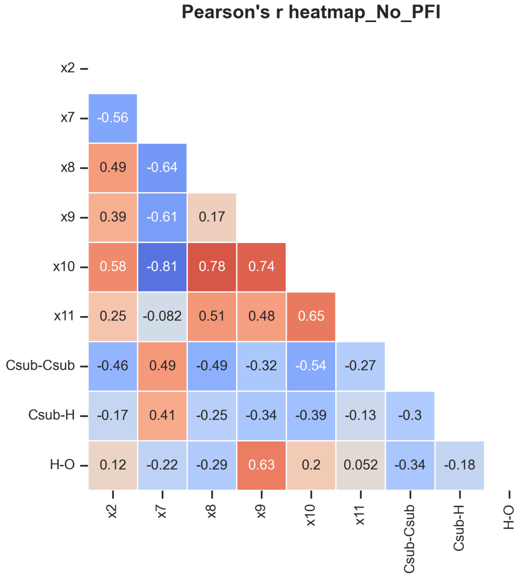
This section presents feature importances measured using the validation set.

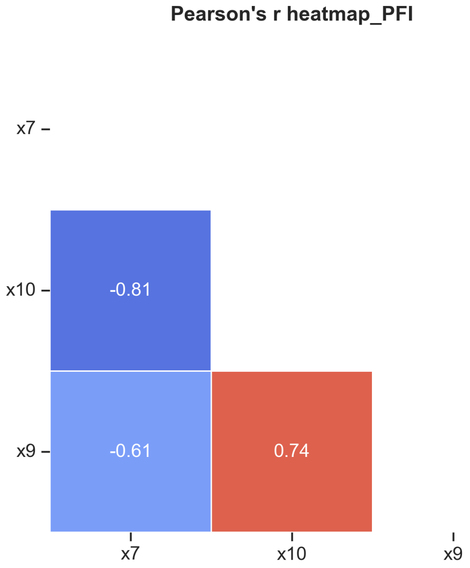

SHAP analysis of NN_No_PFI


SHAP analysis of NN_PFI


Permutation feature importances (PFIs) of NN_No_PFI


Permutation feature importances (PFIs) of NN_PFI


Pearson's r heatmap_No_PFI


Pearson's r heatmap_PFI

Correlation analysis

x WARNING! Noticeable correlations observed (up to r = -0.81 or R^2 = 0.66, for x7 and x10)

Correlation analysis

x WARNING! Noticeable correlations observed (up to r = -0.81 or R^2 = 0.66, for x7 and x10)

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 30 datapoints (10.0%)

- 2 (2.2 SDs)
- 19 (2.6 SDs)
- 21 (2.4 SDs)

Test: 0 outliers out of 7 datapoints (0.0%)

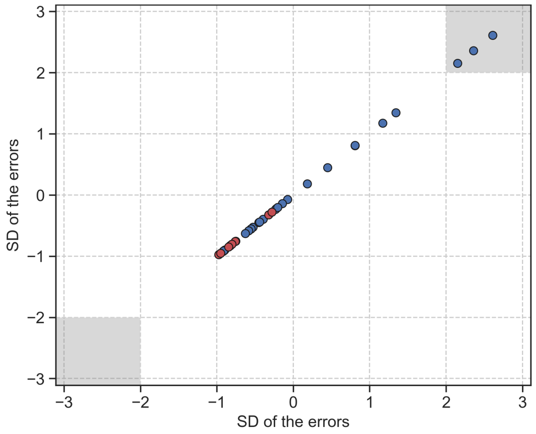
PFI (only important descriptors):

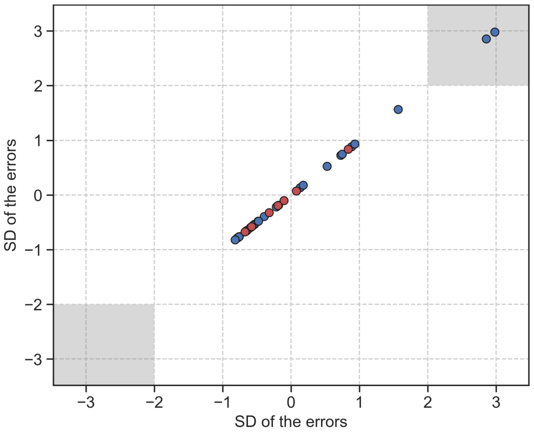
Outliers (max. 10 shown)

Train: 2 outliers out of 30 datapoints (6.7%)

- 19 (2.9 SDs)
- 21 (3.0 SDs)

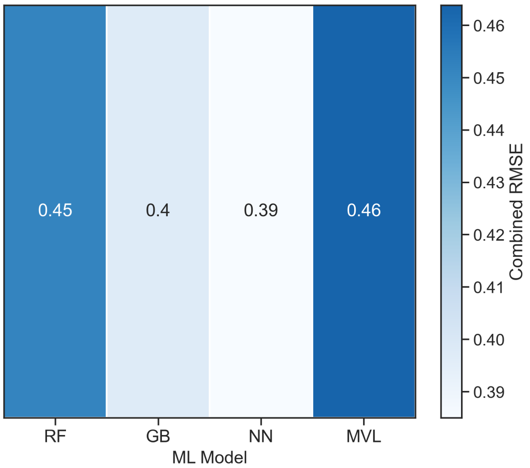
Test: 0 outliers out of 7 datapoints (0.0%)

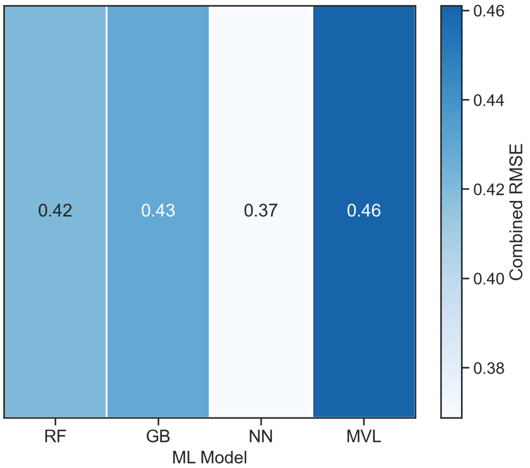




Section F. Model Screening

This section compares different combinations of hyperoptimized algorithms and partition sizes. The combined error is calculated as the product of the training error, validation error, and cross-validation error.





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 -y -c conda-forge robert
- Adjust ROBERT version: pip install robert==2.0.1
- Install scikit-learn-intelex: pip install scikit-learn-intelex==2025.2.0

(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 "Name" --y "Target_values" --csv_name "Robert_example.csv"
```

4. Execution time, Python version and OS:

Originally run in Python 3.10.17 using Windows 10.0.19045

Total execution time: 205.2 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 important descriptors):**
sklearn model: MLPRegressor	sklearn model: MLPRegressor
hidden_layer_1: 7	hidden_layer_1: 7
hidden_layer_2: 2	hidden_layer_2: 2
max_iter: 239	max_iter: 239
alpha: 0.03838855158317655	alpha: 0.03838855158317655
tol: 4.273396938483603e-05	tol: 4.273396938483603e-05
random_state: 0	random_state: 0
solver: lbfgs	solver: lbfgs

2. ROBERT options, including prediction type (REG or CLAS), folds and repeats used for CV, etc:

No PFI (standard descriptor filter):	**PFI (only important descriptors):**
type: reg	type: reg
kfold: 5	kfold: 5
repeat_kfolds: 10	repeat_kfolds: 10
seed: 0	seed: 0
error_type: rmse	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.

External test predictions (sorted, max. 20 shown)

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

Name	**Target_va..._pred ± sd**
46	1.95 ± 0.07
38	1.75 ± 0.2
39	1.66 ± 0.19
42	1.32 ± 0.26
41	0.89 ± 0.06
40	0.82 ± 0.09
44	0.37 ± 0.05
43	0.37 ± 0.04
45	0.35 ± 0.04

External test predictions (sorted, max. 20 shown)

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

Name	**Target_va..._pred ± sd**
46	2.18 ± 0.06
39	1.69 ± 0.11
38	1.69 ± 0.11
42	1.51 ± 0.11
41	0.87 ± 0.05
40	0.81 ± 0.05
45	0.35 ± 0.01
44	0.35 ± 0.01
43	0.35 ± 0.01

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.