Supplemental information

Machine learning workflows beyond linear models in low-data regimes

David Dalmau, Matthew S. Sigman, Juan V. Alegre-Requena*

Table of Contents

Installation of ROBERT 2.0.0

To install the latest version of ROBERT, use the following command in your terminal or Anaconda prompt:

```
pip install robert==2.0.0
```

Ensure that you have Python 3.10 or a compatible version installed. For best performance, we recommend installing the Intel-extension for scikit-learn[1] on supported systems:

```
pip install scikit-learn-intelex==2025.0.0
```

After installation, ROBERT 2.0.0 is ready to use. Detailed installation instructions, usage examples and documentation can be found on our Read the Docs page.[2]

Hyperparameter space for Bayesian optimization of models

The hyperparameter spaces used during Bayesian optimization (BO) of the different machine learning models are shown in Table S1.

Table S1. Hyperparameter space for BO and algorithms used.

Model	Hyperparameter	Range
Random Forest (RF)	n_estimators	(10, 100)
	max_depth	(5, 20)
	min_samples_split	(2, 10)
	min_samples_leaf	(2, 5)
	min_weight_fraction_leaf	(0, 0.05)
	max_features	(0.25, 1.0)
	ccp_alpha	(0, 0.01)
	max_samples	(0.25, 1.0)
Gradient Boosting (GB)	n_estimators	(10, 100)

	learning_rate	(0.01, 0.3)
	max_depth	(5, 20)
	min_samples_split	(2, 10)
	min_samples_leaf	(2, 5)
	subsample	(0.7, 1.0)
	max_features	(0.25, 1.0)
	validation_fraction	(0.1, 0.3)
	min_weight_fraction_leaf	(0, 0.05)
	ccp_alpha	(0, 0.01)
Neural Network (NN) Multi-layer Perceptron (MLP) implemented with scikit-learn's MLPRegressor using the L-BFGS solver.	hidden_layer_1	(1, 10)
	hidden_layer_2	(0, 10)
	max_iter	(200, 500)
	alpha	(0.01, 0.1)
	tol	(0.00001, 0.0001)
LinearRegression (MVL)	Only default parameters were used for the MVL algorithm, with no BO applied.	

This hyperparameter space was explored to identify optimal values that minimize overfitting while enhancing model performance.

Methodology

In this section, we outline the data analysis procedure using ROBERT. Descriptors exhibiting high correlations (correlation coefficient > 0.7) were automatically excluded. This filtering process was applied consistently across all benchmark examples, and one descriptor was removed in examples **C** and **D**.

Table S2 contains the commands used for each example, along with instructions to replicate the results. After executing these commands, ROBERT generates a comprehensive PDF report with a summary of the results. All the reports resulting from this work were provided along with the ESI.

Instructions to Replicate the Results:

1. Ensure that ROBERT 2.0.0 is installed and the appropriate Python environment is activated.
2. Place the data files (A.csv, B.csv, C.csv, …) in the working directory.
3. Execute the commands provided in Table S2 using your terminal or Anaconda Prompt.
4. Once the analysis is complete, a PDF report will appear in the designated directory.

For additional guidance, see the documentation on our Read the Docs page.

Table S2. Command lines used for examples **A**-**H**.

Example	Command line used
A	python -m robert --csv_name A.csv --y G_Exp --names ligand --model [MODEL]
B	python -m robert --csv_name B.csv --y NddG≠ --names Label --model [MODEL]
C	python -m robert --csv_name C.csv --y Nselectivity --names Label --model [MODEL]
D	python -m robert --csv_name D.csv --y NddG≠ --names Label --model [MODEL]
E	python -m robert --csv_name E.csv --y NddG≠ --names Label --model [MODEL]
F	python -m robert --csv_name F.csv --names LNiArCl --y ddG --model [MODEL]
G	python -m robert --csv_name G.csv --y NdG --names Label --model [MODEL]
H	python -m robert --csv_name H.csv --y ln(k)_rate --names Coupling --model [MODEL] --csv_test H_test.csv

MODEL = RF, GB, NN, MVL

Benchmark tables

Table S3. Scaled RMSE values for 10x5-fold CV for examples **A**-**H**.

Example	RF	GB	NN	MVL
A	22.50	21.67	16.67	15.83
B	23.75	19.50	16.00	13.00
C	18.93	17.14	15.36	13.57
D	20.26	19.21	17.37	18.16

E	22.65	18.24	10.59	11.76
F	23.64	23.64	18.18	21.82
G	10.93	9.77	7.91	7.21
H	23.08	18.46	16.92	16.92

Table S4. Scaled RMSE values for external test sets for examples **A**-**H**.

Example	RF	GB	NN	MVL
A	12.50	9.17	6.25	10.00
B	14.50	20.75	11.25	9.25
C	12.50	7.86	9.29	9.29
D	21.05	22.89	28.95	17.63
E	24.41	16.47	9.41	8.53
F	10.91	9.09	8.64	15.45
G	5.35	4.42	3.95	3.95
H	14.15	14.15	16.92	15.38

Table S5. ROBERT scores for examples **A**-**H**.

Example	RF	GB	NN	MVL
A	3	5	6	7
B	0	3	5	7
C	6	7	8	7
D	4	3	1	3
E	3	4	7	7
F	5	6	6	5
G	7	8	8	8
H	5	5	5	6

Evaluating combined metric for BO and dataset size

In this section, we evaluated three different objective function metrics used during BO across the benchmark examples. The resulting hyperparameters were evaluated using model performance on cross-validation (CV) and external test set results.

1. Combined RMSE (standard)

The first approach combines the RMSE from two validation strategies, including intra- and extrapolation:

$$RMSE_{combined} = \frac{RMSE_{10x\ 5-fold\ CV} + RMSE_{sorted\ CV}}{2}$$

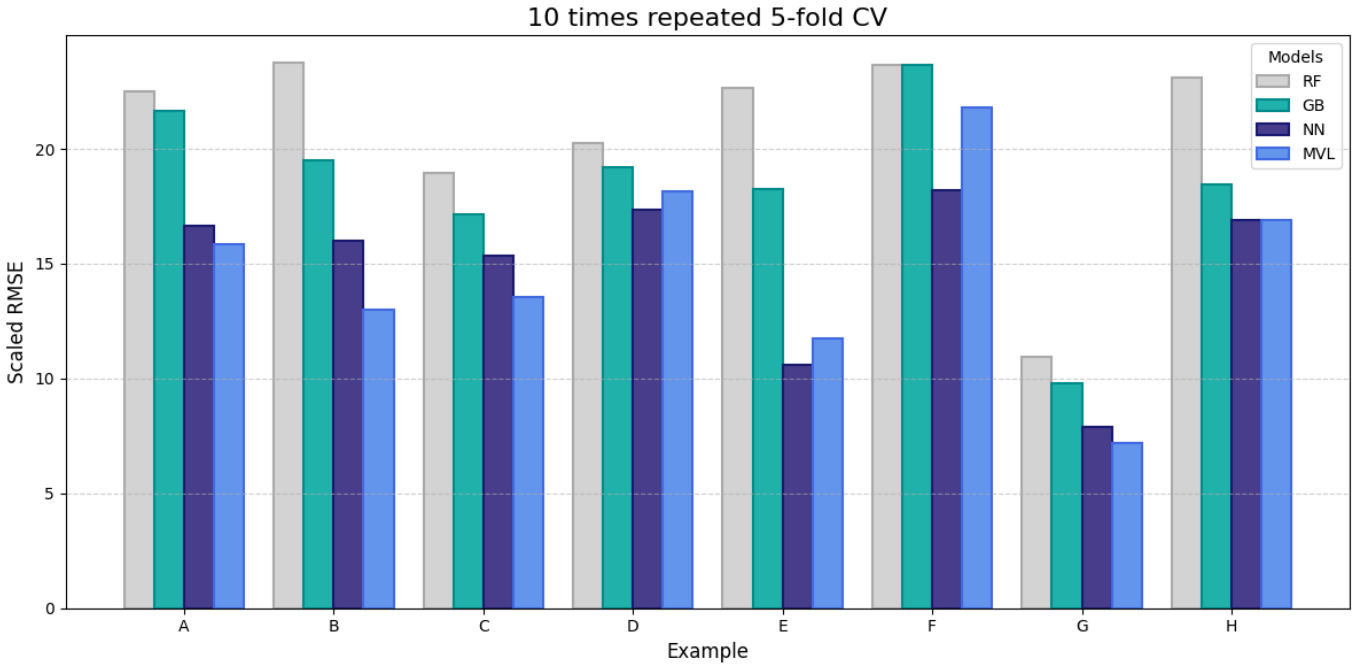


Figure S1. Scaled RMSE for 10-times repeated 5-fold CV using the combined RMSE metric.

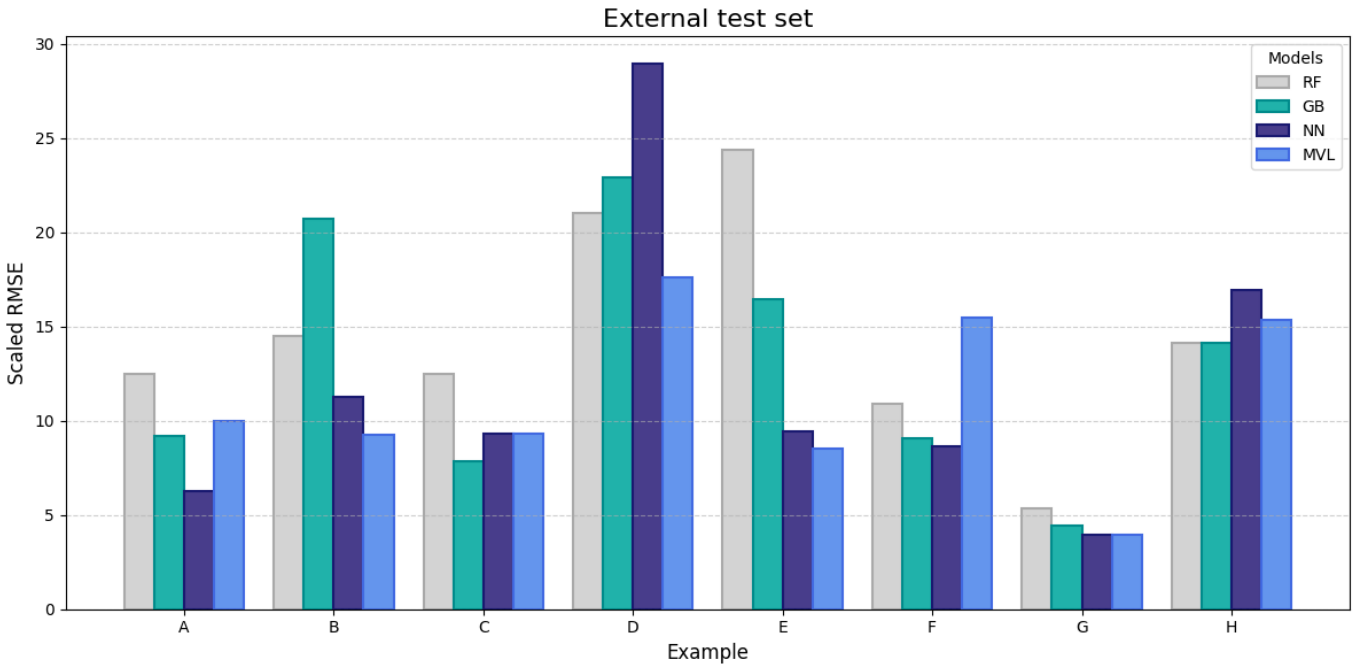


Figure S2. Scaled RMSE for an external test set using the combined RMSE metric.

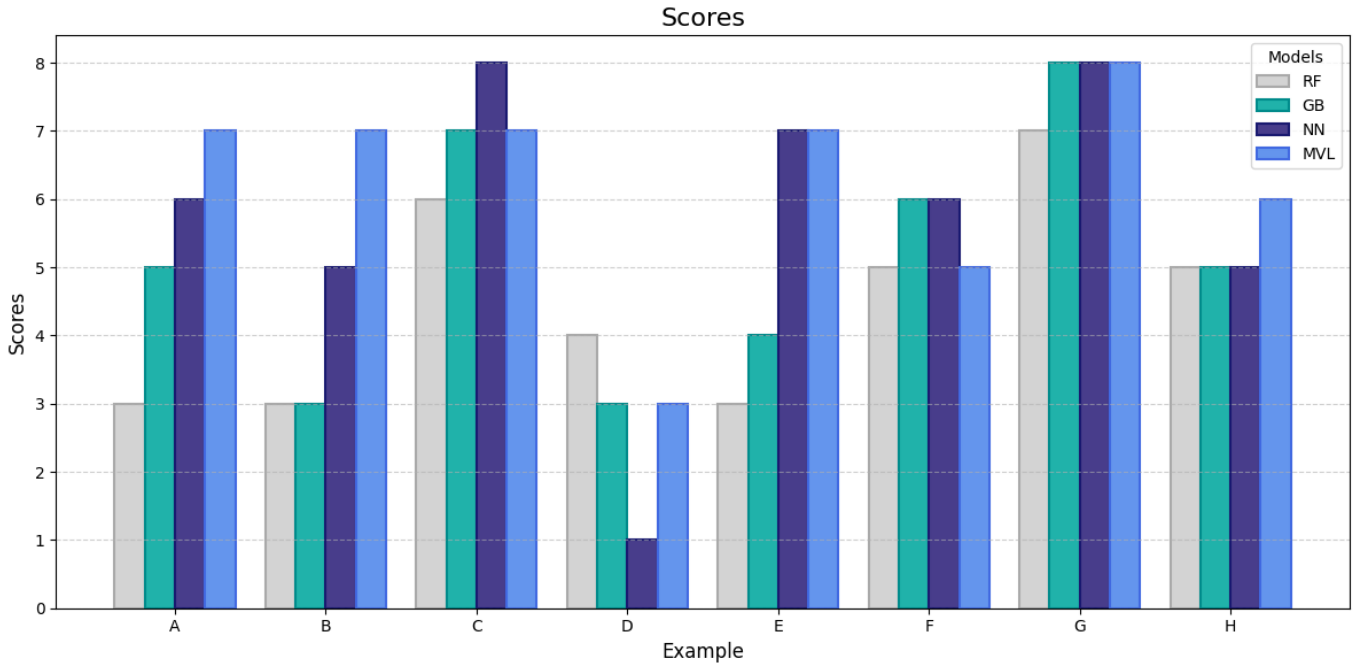


Figure S3. ROBERT model scores using the combined RMSE metric.

2. RMSE from 10x 5-fold CV

This method evaluates model performance using only the RMSE obtained from 10-times repeated 5-fold CV:

$$RMSE_{combined} = RMSE_{10x\ 5-fold\ CV}$$

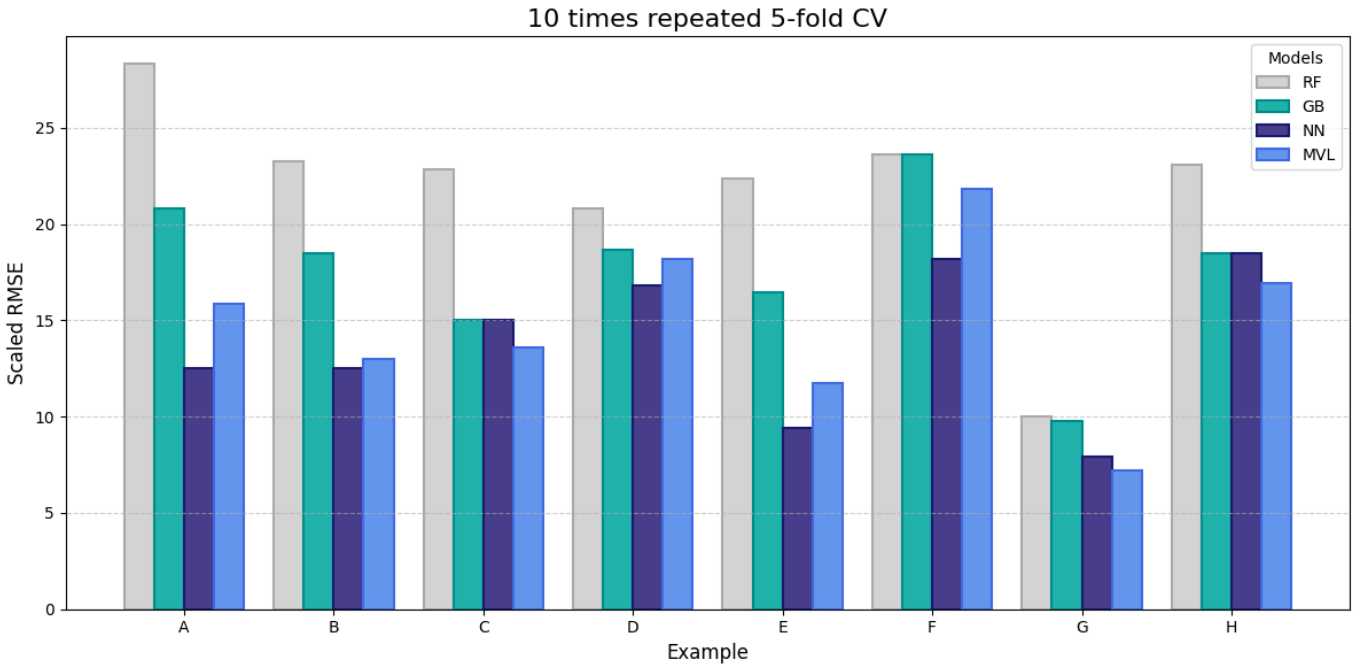


Figure S4. Scaled RMSE for 10-times repeated 5-fold CV.

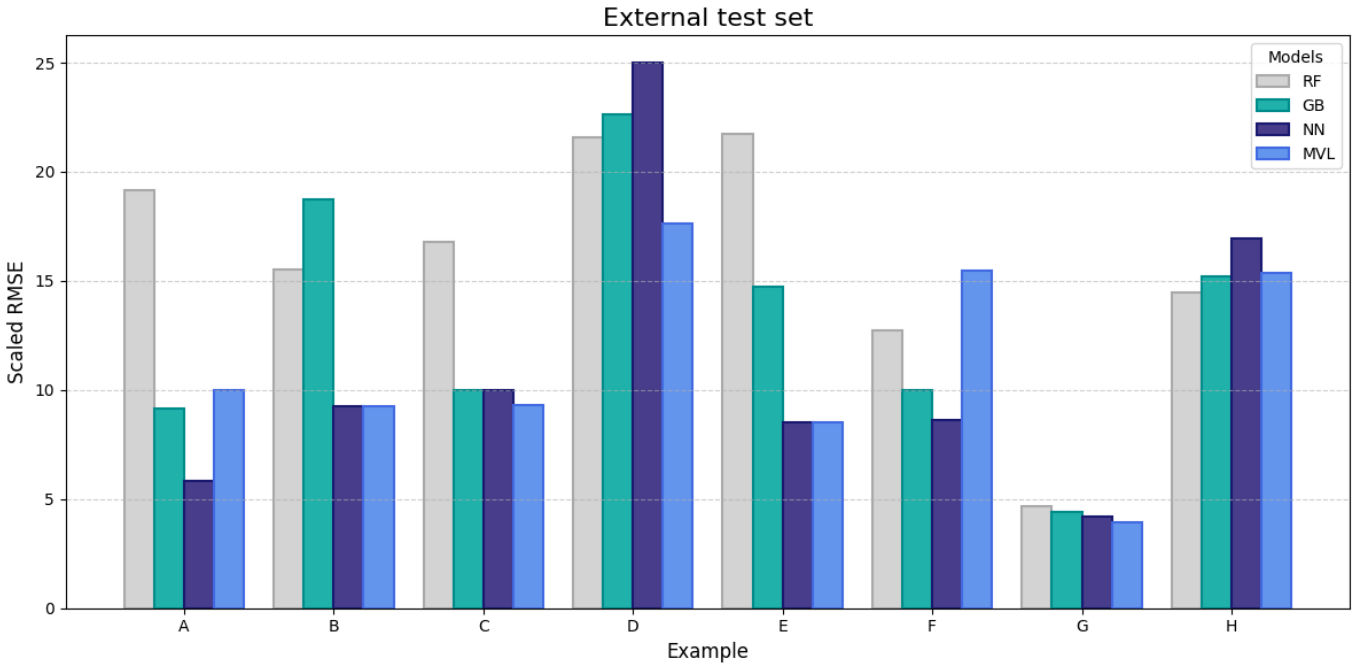


Figure S5. Scaled RMSE for an external test set.

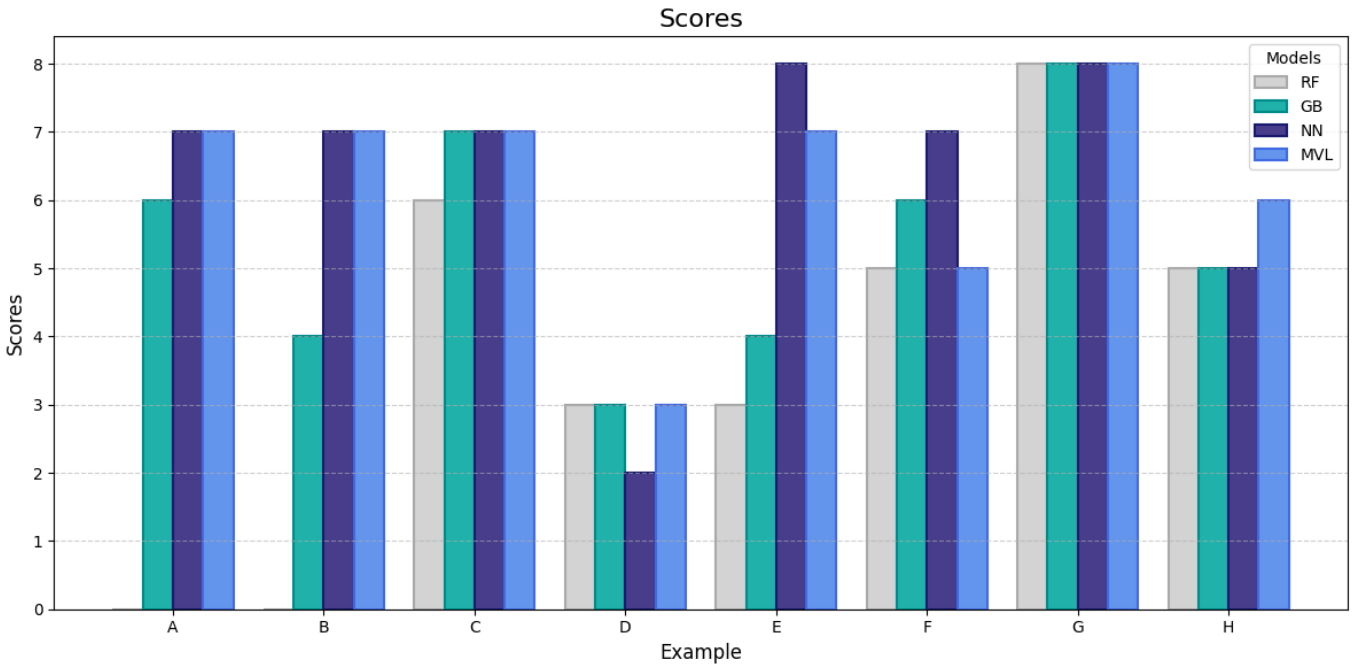


Figure S6. ROBERT model scores using this RMSE metric.

3. RMSE normalized with R^2

In this approach, RMSE is normalized by the coefficient of determination R^2, providing a relative error metric:

$$RMSE_{combined} = \frac{RMSE_{10x\ 5-fold\ CV}}{R^2_{10x\ 5-fold\ CV}}$$

where $R^2_{10x\ 5-fold\ CV}$ corresponds to the R^2 from the same cross-validation procedure.

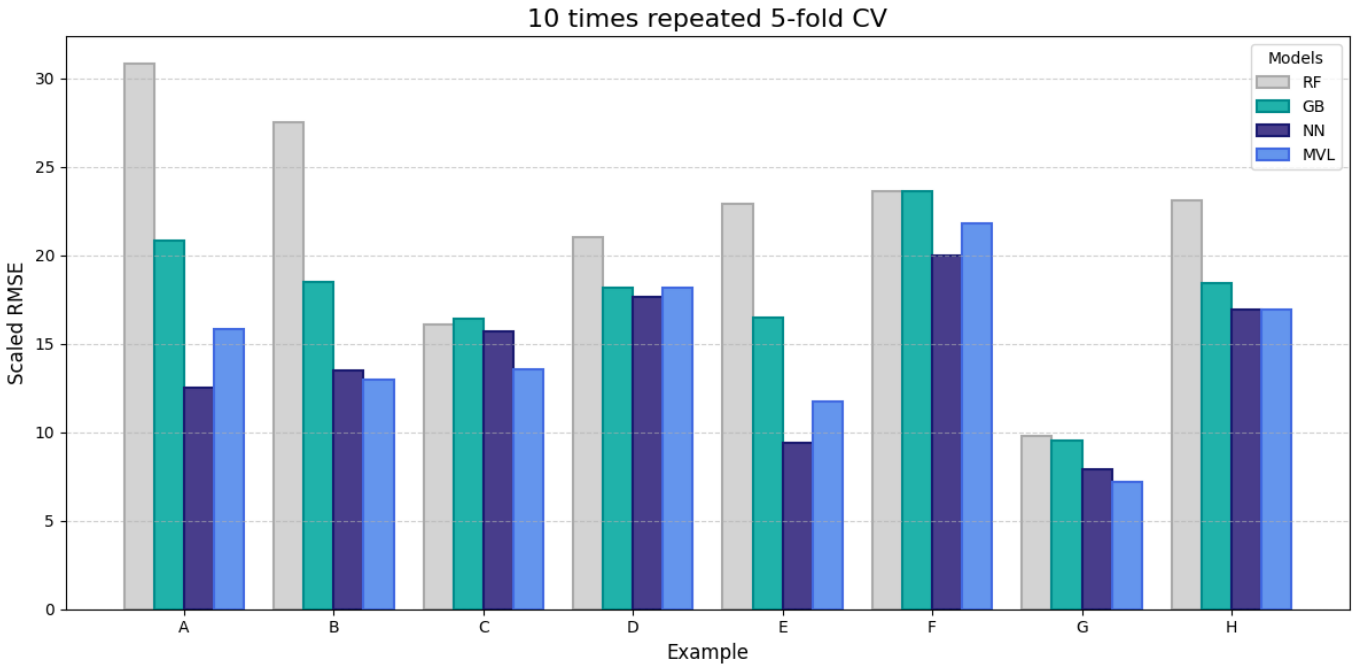


Figure S7. Scaled RMSE for 10-times repeated 5-fold CV using the normalized RMSE metric.

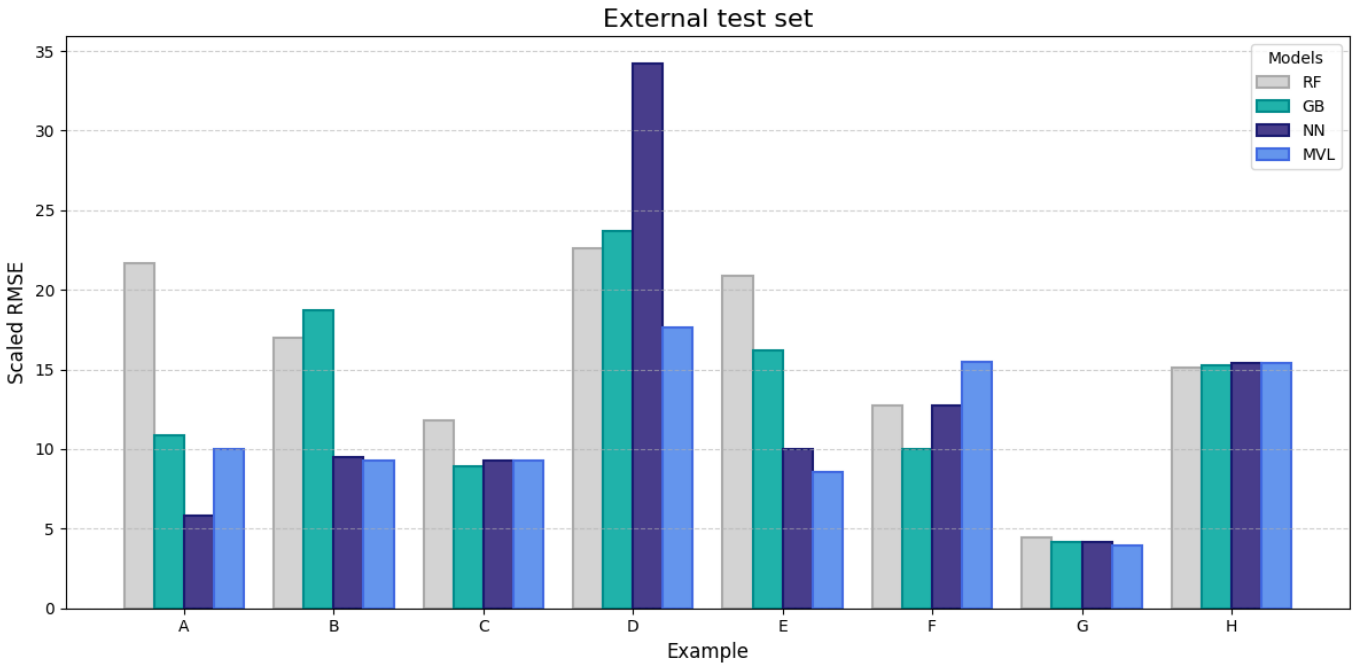


Figure S8. Scaled RMSE for an external test set using the normalized RMSE metric.

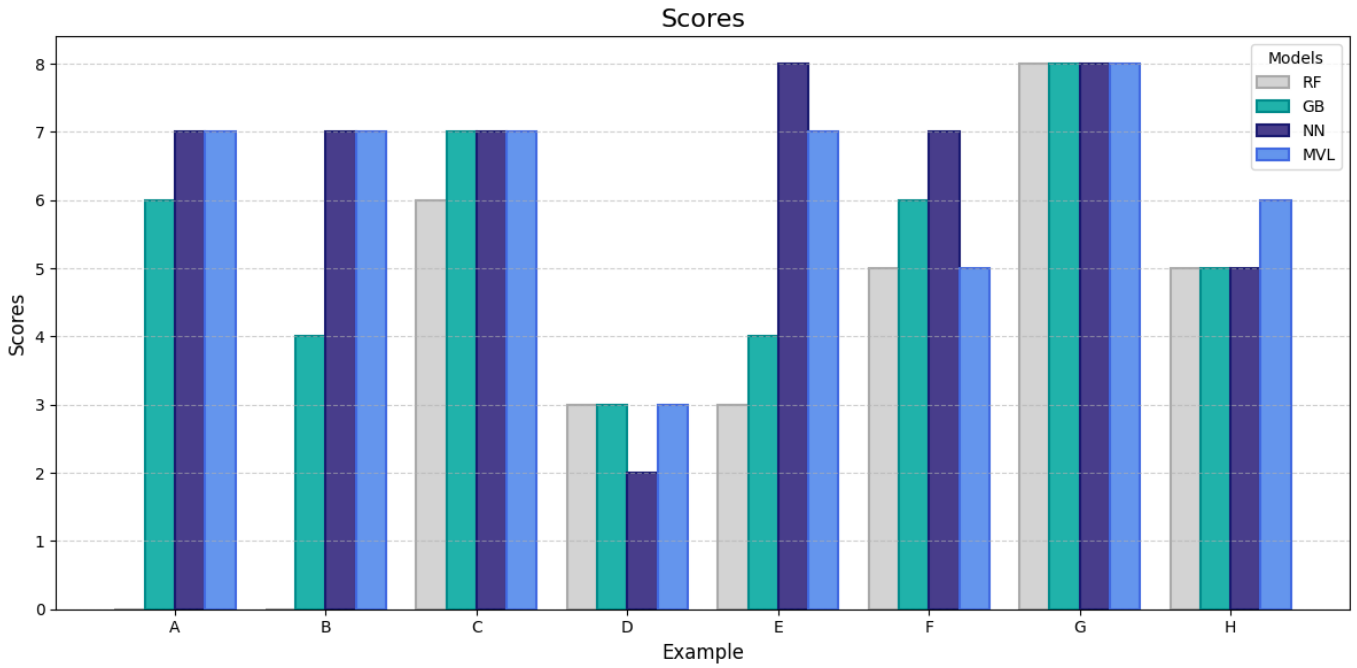


Figure S9. ROBERT model scores based on this metric.

4. Evaluating Dataset Size Impact on RMSE

To understand how dataset size affects model performance, we extended the analysis to three additional datasets (**I**,[3] **J**,[4] **K**[5]) using the standard combined RMSE.

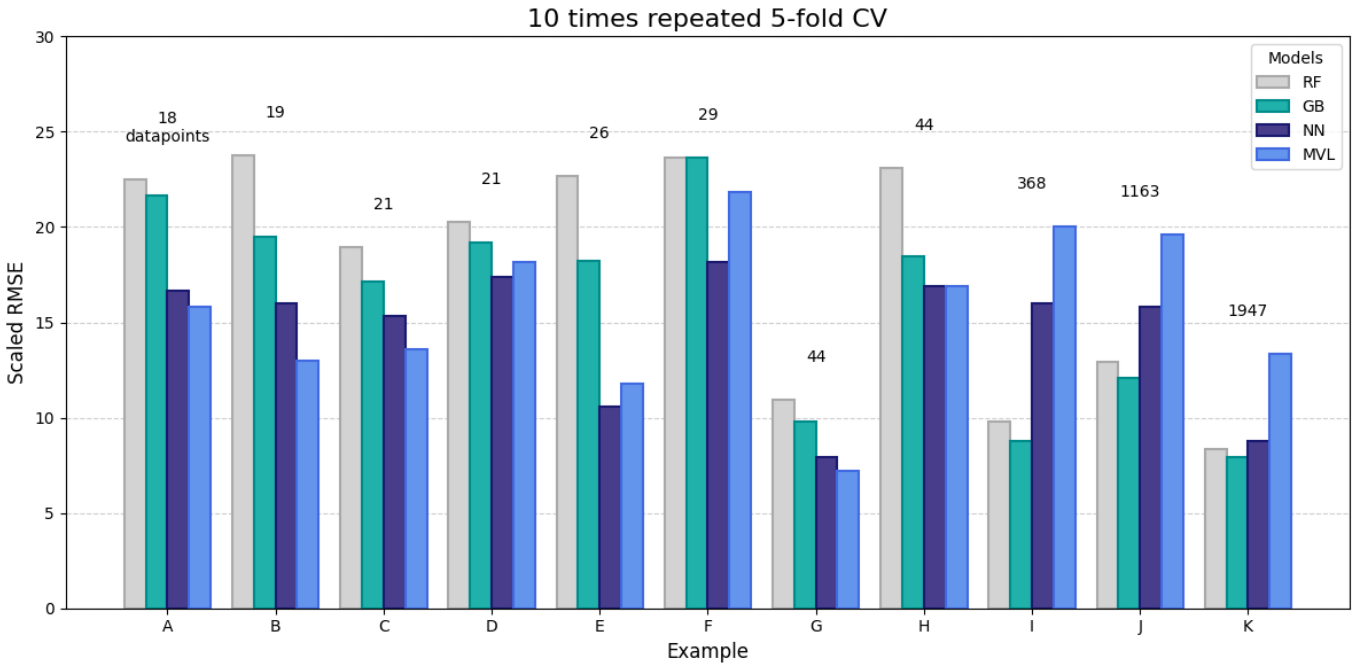


Figure S10. Scaled RMSE for 10-times repeated 5-fold CV across datasets (**A**–**K**).

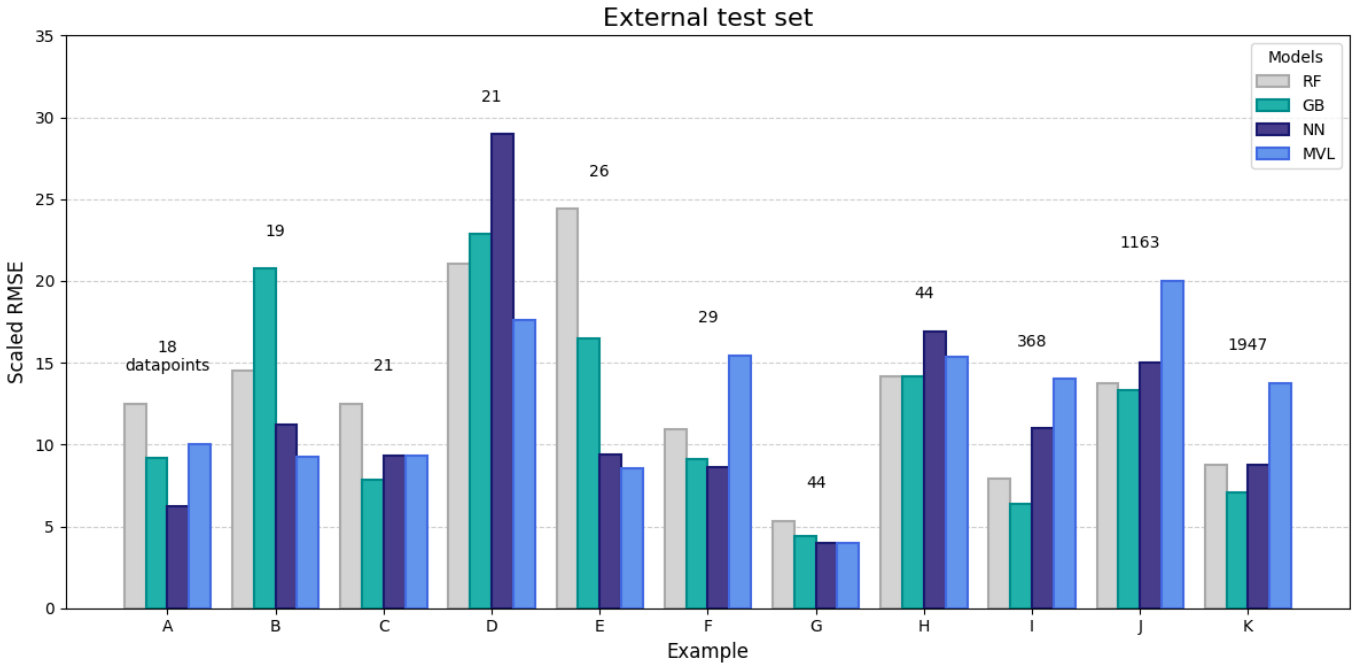


Figure S11. Scaled RMSE for external test set across datasets (**A**–**K**).

ROBERT score

First component: predictive ability and overfitting (up to 8 points)

1. 10x 5-fold CV predictions of the model (min 0; max 2 points)

Two metrics are used to make a more robust evaluation since models might show very good RMSE but low R^2.

Scaled RMSE	Points
≤ 10%	+2
≤ 20%	+1
> 20%	0

R^2 *(penalty)*	Points
< 0.5	–2
< 0.7	–1
≥ 0.7	0

Examples:

a. The 10x 5-fold CV shows a scaled RMSE of 7%, which increases the score by +2, and a R^2 of 0.89, which does not affect the score. Overall, the score increases by +2.
b. The 10x 5-fold CV shows a scaled RMSE of 12%, which increases the score by +1, and a R^2 of 0.80, which does not affect the score. Overall, the score increases by +1.
c. The 10x 5-fold CV shows a scaled RMSE of 25%, which does not affect the score, and a R^2 of 0.4, which reduces the score by –2. Overall, the score is not affected (no negative points in this test, min 0 points).

2. Predictions external test (min 0; max 2 points)

Two metrics are used to make a more robust evaluation since models might show very good RMSE but low R^2.

Scaled RMSE	Points
≤ 10%	+2
≤ 20%	+1
> 20%	0

R^2 *(penalty)*	Points
< 0.5	–2
< 0.7	–1
≥ 0.7	0

Examples: Same as in previous test.

3. Prediction accuracy test vs CV (min 0; max 2 points)

Differences in scaled RMSE between 1 and 2.

Scaled RMSE ratio	Points
Scaled RMSE (test) ≤ 1.25*scaled RMSE (CV)	+2
Scaled RMSE (test) ≤ 1.50*scaled RMSE (CV)	+1
Scaled RMSE (test) >1.50*scaled RMSE (CV)	0

Examples:

a. The scaled RMSE of the test set is 0.72, while that of the 10x 5-fold CV is 0.60. The ratio between the scaled RMSE of the test set and CV is 0.72/0.60 = 1.20, which increases the score by +2.
b. The scaled RMSE of the test set is 0.80, while that of the 10x 5-fold CV is 0.60. The ratio between the scaled RMSE of the test set and CV is 0.80/0.60 = 1.33, which increases the score by +1.
c. The scaled RMSE of the test set is 1.20, while that of the 10x 5-fold CV is 0.60. The ratio between the scaled RMSE of the test set and CV is 1.20/0.60 = 2.00, and the score is not affected.

4. Extrapolation with sorted CV (min 0; max 2 points)

Differences in the RMSE obtained across the five folds of a sorted 5-fold CV (where target values, y, are sorted from minimum to maximum and not shuffled during CV). First, the minimum RMSE among the five folds is identified. Then, the differences between each fold's RMSE and this minimum RMSE are evaluated.

Scaled RMSE difference	Points
Every two folds with RMSE ≤ 1.25*min RMSE	+1

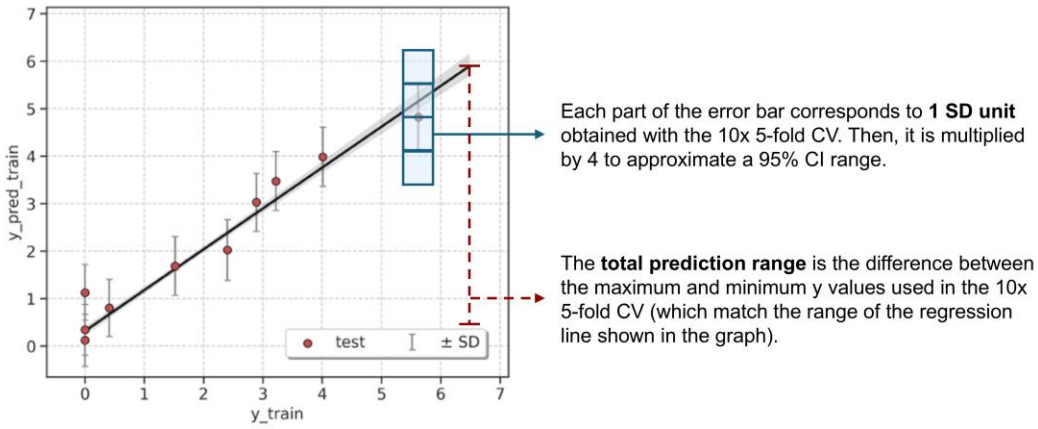
Examples:

a. In a sorted 5-fold CV, the RMSE values for each fold are 0.50, 0.50, 0.45, 0.46, and 0.52. The minimum RMSE is 0.45, so the threshold of 1.25*min RMSE is 0.56. Four folds fall below this threshold (0.50, 0.50, 0.46, and 0.52), increasing the score by +2.
b. In a sorted 5-fold CV, the RMSE values for each fold are 1.20, 0.50, 0.45, 0.46, and 0.52. The minimum RMSE is 0.45, so the threshold of 1.25*min RMSE is 0.56. Three folds fall below this threshold (0.50, 0.46, and 0.52), increasing the score by +1.
c. In a sorted 5-fold CV, the RMSE values for each fold are 1.20, 1.30, 0.45, 0.46, and 1.32. The minimum RMSE is 0.45, so the threshold of 1.25*min RMSE is 0.56. One fold falls below this threshold (0.46), and the score is not affected.

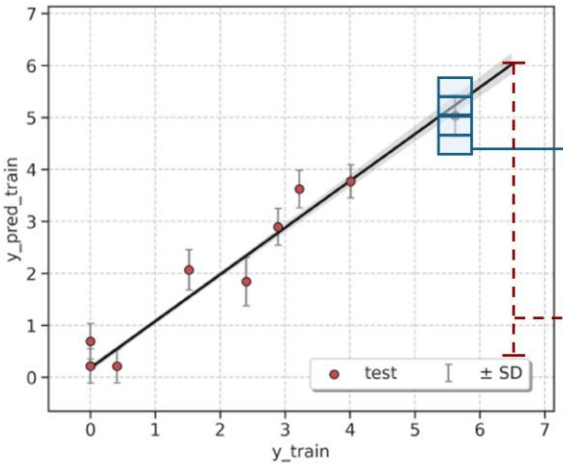
Second component: prediction uncertainty (min 0; max 2 points)

The model's uncertainty is estimated using predictions from the 10 repetitions of the 10x 5-fold CV. ROBERT then computes the average standard deviation (SD) from all predictions and multiplies it by 4 to approximate the 95% confidence interval (CI) of a normally distributed population. The score for this test depends on the uncertainty of the results, measured by the width of the 95% CI across the range of y values.

Scaled RMSE difference	Points
95% CI (or 4*SD) spans ≤ 25% of the y range	+2
95% CI spans between 25% and 50% of the y range	+1
95% CI spans > 50% of the y range	0



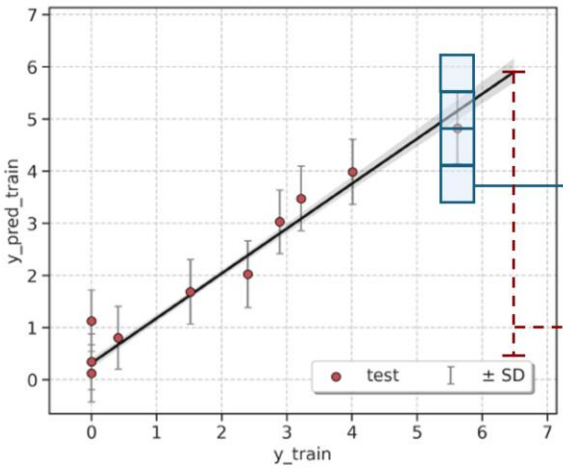
Examples:



Score +2. The 95% CI spans (1.4/5.8)*100 = 24% of the total range of predictions. This value is below 25%, indicating a low prediction uncertainty and increasing the score in +2.

95% CI (or 4*SD) = 1.4

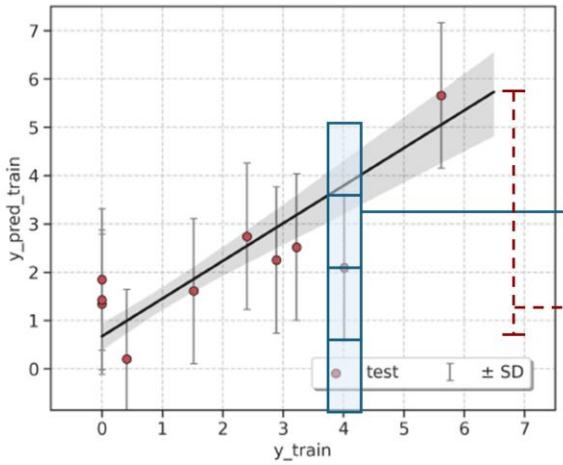
Total prediction range = 5.8



Score +1. The 95% CI spans (2.7/5.6)*100 = 48% of the total range of predictions. This value is between 25% and 50%, indicating a moderate prediction uncertainty and increasing the score in +1.

95% CI (or 4*SD) = 2.7

Total prediction range = 5.6



Score 0. The 95% CI spans (5.8/5.2)*100 = 112% of the total range of predictions. This value is above 50%, indicating a high prediction uncertainty and not affecting the score.

95% CI (or 4*SD) = 5.8

Total prediction range = 5.2

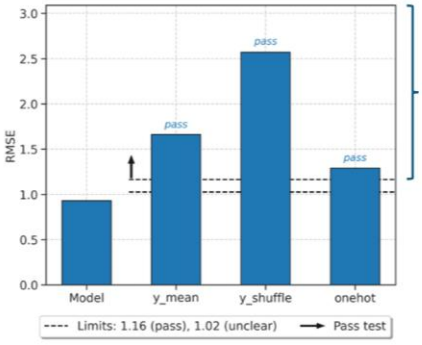
Third component: model vs "flawed" models (*penalty*, min –6; max 0 points)

The model's performance is compared to that of different "flawed" models

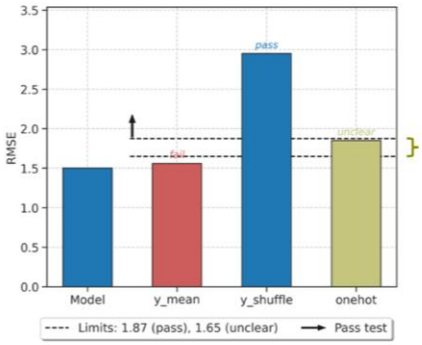
- y-mean test: Error of a model where all predicted y values are fixed to the mean of the measured y values (resulting in a straight line when plotting measured vs predicted y values).
- y-shuffle test:[6] Error of a model trained on randomly shuffled measured y values.
- Onehot encoding test:[7] Error of a model where all descriptors are replaced with 0s and 1s. If the descriptor value is 0, the descriptor remains 0; otherwise, it is set to 1 (useful for combinatorial databases).

Test result	Points
Pass	–2
Unclear	–1
Fail	0

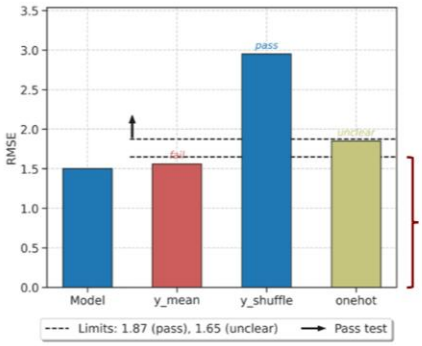
Examples:



Pass, score 0 each test.
In this region, the "flawed" models show more than 30% error compared to our model. Therefore, our model seems to predict correctly "for the right reasons".



Unclear, score –1 each test.
In this region, the "flawed" models show between 15% and 30% error compared to our model. These results are in an "unclear" situation, as the errors are higher than that of our model, but they are dangerously close.



Fail, score –2 each test.
In this region, the "flawed" models show less than 15% error compared to our model. Therefore, our model seems to have important flaws (i.e., overfitting, lack of use of meaningful feature values, etc.).

References

1. Intel(R). https://github.com/intel/scikit-learn-intelex
2. https://robert.readthedocs.io
3. S. Singh, M. Pareek, A. Changotra, S. Banerjee, B. Bhaskararao, P. Balamurugan and R. B. Sunoj, *Proc. Natl. Acad. Sci. U.S.A.*, 2020, **117**, 1339–1345.
4. G. G. Terrones, C. Duan, A. Nandy and H. J. Kulik, *Chem. Sci.*, 2023, **14**, 1419–1433.
5. P. Friederich, G. Dos Passos Gomes, R. De Bin, A. Aspuru-Guzik and D. Balcells, *Chem. Sci.*, 2020, **11**, 4584–4601.
6. C. Rücker, G. Rücker and M. Meringer, *J. Chem. Inf. Model.*, 2007, **47**, 2345–2357.
7. D. T. Ahneman, J. G. Estrada, S. Lin, S. D. Dreher and A. G. Doyle, *Science*, 2018, **360**, 186–190.