



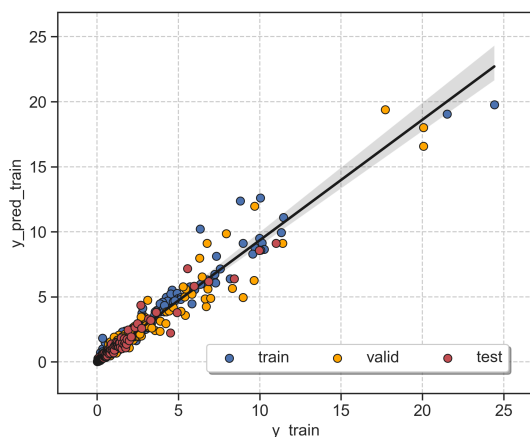
ROBERT v 1.2.1 2025/08/05 18:58:07

**How to cite:** Dalmau, D.; Alegre Requena, J. V. WIREs Comput Mol Sci. 2024, DOI: 10.1002/WCMS.1733**Section A. ROBERT Score***This score is designed to evaluate the models using different metrics.***No PFI (standard descriptor filter):**

Model = RF · Train:Validation:Test = 63:27:10

Points(train+valid.):descriptors = 1008:9

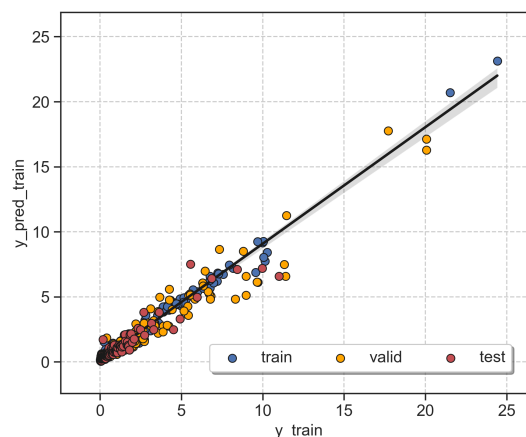
Score = 10 / 10

**STRONG**Train :  $R^2 = 0.97$ , MAE = 0.14, RMSE = 0.38Valid. :  $R^2 = 0.94$ , MAE = 0.29, RMSE = 0.63Test :  $R^2 = 0.94$ , MAE = 0.23, RMSE = 0.48**PFI (only most important descriptors):**

Model = GB · Train:Validation:Test = 54:36:10

Points(train+valid.):descriptors = 1008:6

Score = 10 / 10

**STRONG**Train :  $R^2 = 0.99$ , MAE = 0.19, RMSE = 0.31Valid. :  $R^2 = 0.94$ , MAE = 0.32, RMSE = 0.66Test :  $R^2 = 0.9$ , MAE = 0.34, RMSE = 0.68**Severe warnings**

- ☒ No severe warnings detected

**Moderate warnings**

- ☐ Uneven y distribution (Section C)
- ☐ Highly correlated features (Section D)
- ☐ Potential "faulty" outliers (Section E)

**Overall assessment**

- ☐ Reliable model, but examine warnings

**Severe warnings**

- ☒ No severe warnings detected

**Moderate warnings**

- ☐ Uneven y distribution (Section C)
- ☐ Potential "faulty" outliers (Section E)

**Overall assessment**

- ☒ The model seems reliable

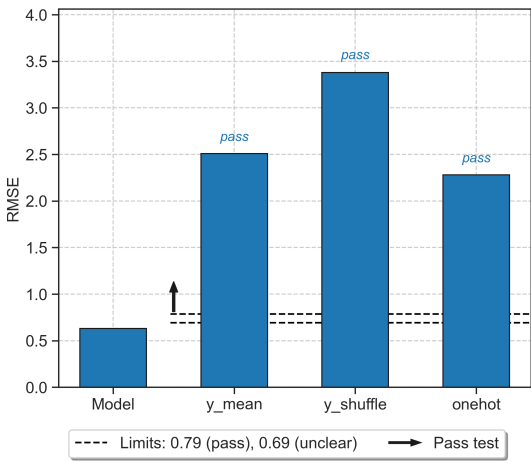


Section B. Advanced Score Analysis

This section explains each component that comprises the ROBERT score.

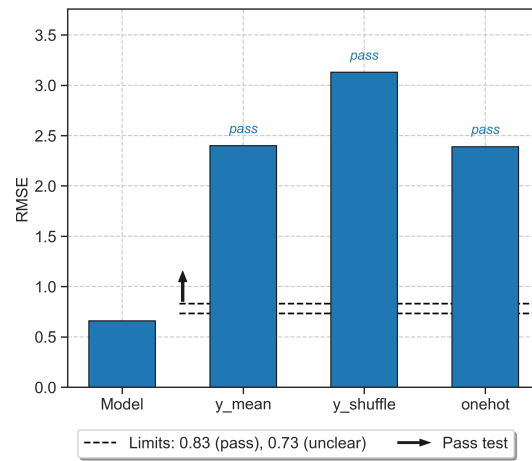
1. Model vs "flawed" models (3 / 3)

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  $R^2$  (test) = 0.94.  
 $R^2$  0.70-0.85: +1,  $R^2$  >0.85: +2.

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

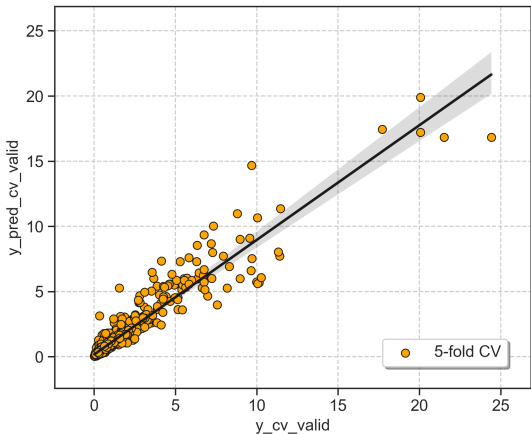
Good predictive ability with  $R^2$  (test) = 0.9.  
 $R^2$  0.70-0.85: +1,  $R^2$  >0.85: +2.

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

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

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

Good predictive ability with  $R^2$  (5-fold CV) = 0.91.  
 $R^2$  0.70-0.85: +1,  $R^2$  >0.85: +2.

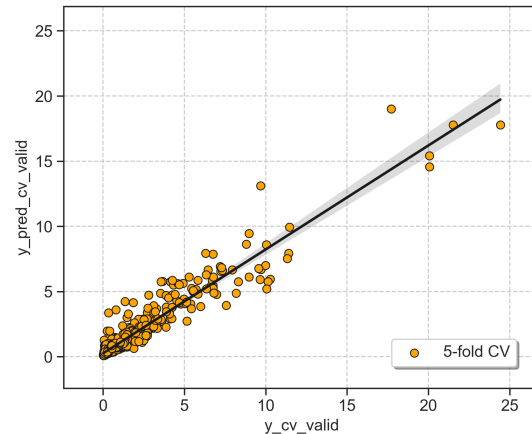


3. Cross-validation (5-fold CV) of the model

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

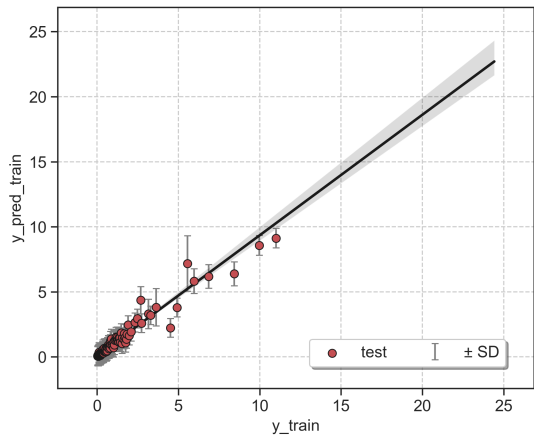
3a. CV predictions train + valid. (2 / 2)

Good predictive ability with  $R^2$  (5-fold CV) = 0.92.  
 $R^2$  0.70-0.85: +1,  $R^2$  >0.85: +2.



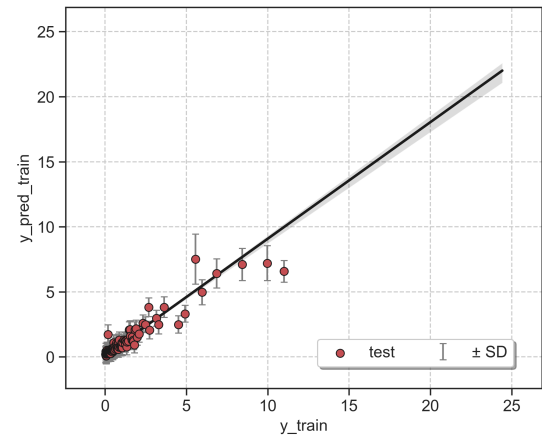
3b. Avg. standard deviation (SD) (2 / 2 )

Low variation, 4\*SD (test) = 3.0 (12% y-range).  
4\*SD 25-50% y-range: +1, 4\*SD < 25% y-range: +2.  
[Details here.](#)



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

Low variation, 4\*SD (test) = 2.6 (11% y-range).  
4\*SD 25-50% y-range: +1, 4\*SD < 25% y-range: +2.  
[Details here.](#)



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

Decent number of descps. (ratio 1008:9).  
5 or more points per descriptor: +1.

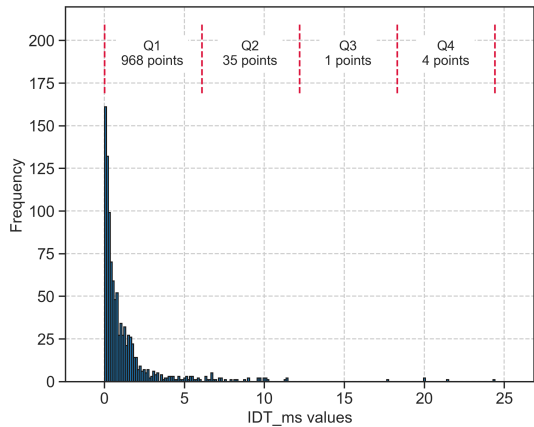
4. Points(train+valid.):descriptors (1 / 1 )

Decent number of descps. (ratio 1008:6).  
5 or more points per descriptor: +1.



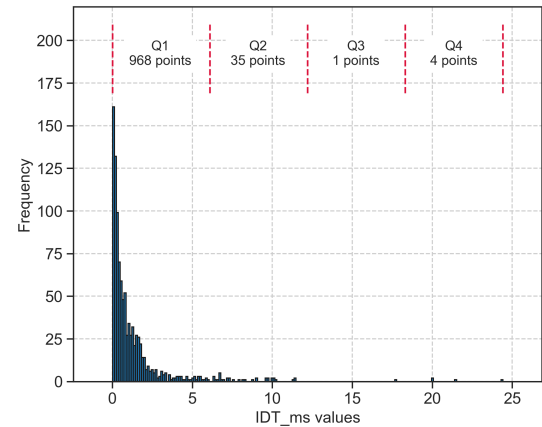
Section C. Distribution of y Values

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



y distribution analysis

x WARNING! Your data is not uniform (Q3 has 1 points while Q1 has 968)



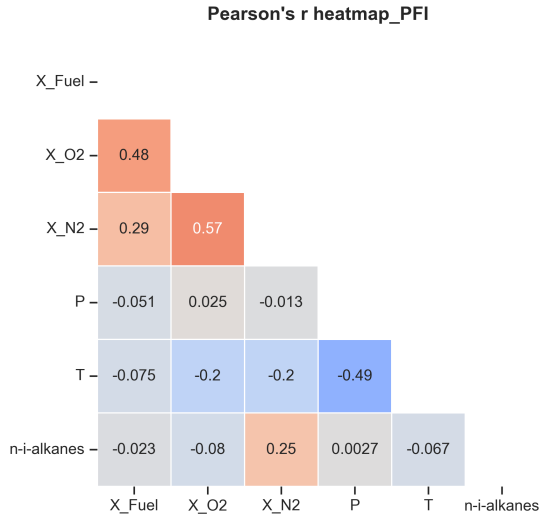
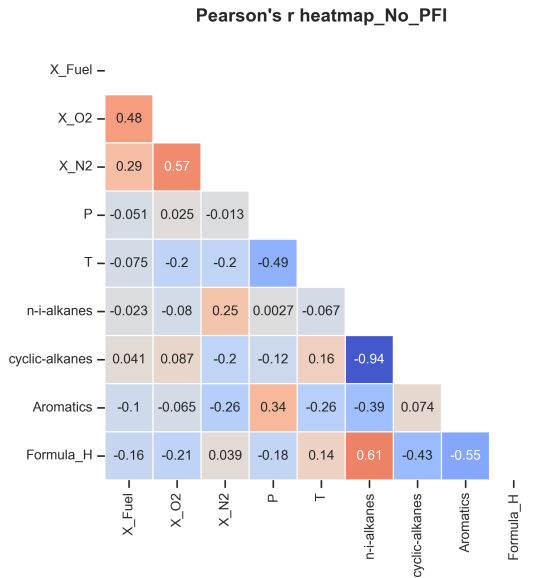
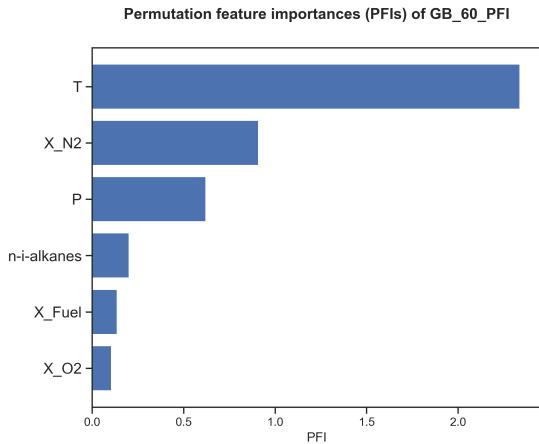
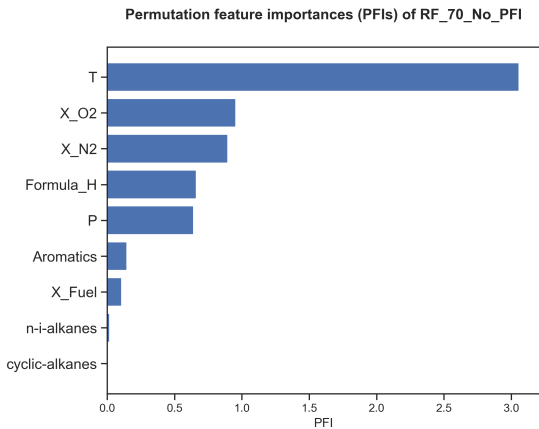
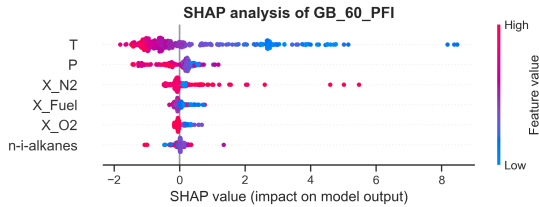
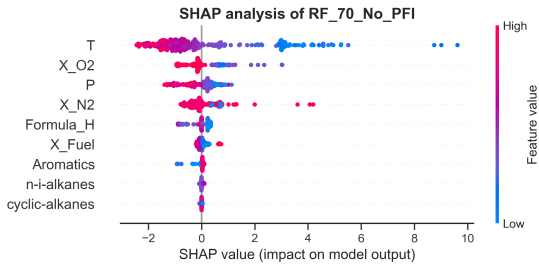
y distribution analysis

x WARNING! Your data is not uniform (Q3 has 1 points while Q1 has 968)



Section D. Feature Importances

This section presents feature importances measured using the validation set.



Correlation analysis

x WARNING! High correlations observed (up to  $r = -0.94$  or  $R^2 = 0.89$ , for n-i-alkanes and cyclic-alkanes)

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: 21 outliers out of 705 datapoints (3.0%)

- 27 (2.4 SDs)
- 150 (2.4 SDs)
- 319 (2.2 SDs)
- 320 (3.2 SDs)
- 322 (2.4 SDs)
- 323 (4.1 SDs)
- 325 (3.7 SDs)
- 364 (2.2 SDs)
- 403 (2.4 SDs)
- 426 (3.3 SDs)

Validation: 23 outliers out of 303 datapoints (7.6%)

- 402 (9.0 SDs)
- 406 (2.8 SDs)
- 424 (2.9 SDs)
- 427 (4.2 SDs)
- 440 (1.1e+01 SDs)
- 441 (6.4 SDs)
- 444 (3.2 SDs)
- 522 (7.0 SDs)
- 597 (4.6 SDs)
- 611 (5.4 SDs)

Test: 7 outliers out of 112 datapoints (6.2%)

- 560 (4.3 SDs)
- 683 (6.0 SDs)
- 378 (2.8 SDs)
- 991 (4.1 SDs)
- 710 (5.3 SDs)
- 324 (3.5 SDs)
- 654 (4.9 SDs)

### PFI (only most important descriptors):

#### Outliers (max. 10 shown)

Train: 23 outliers out of 604 datapoints (3.8%)

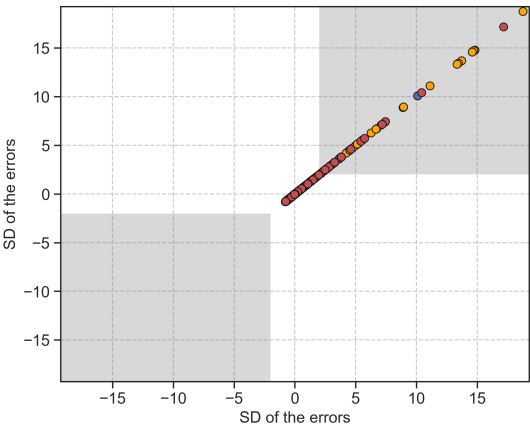
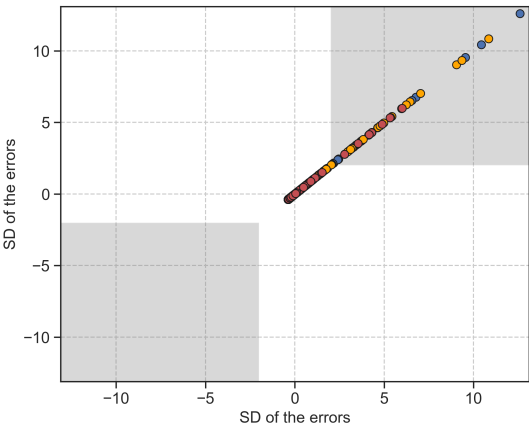
- 11 (3.1 SDs)
- 300 (4.4 SDs)
- 320 (1e+01 SDs)
- 321 (2.6 SDs)
- 322 (8.9 SDs)
- 323 (6.7 SDs)
- 325 (7.4 SDs)
- 361 (2.6 SDs)
- 403 (3.3 SDs)
- 405 (2.3 SDs)

Validation: 36 outliers out of 404 datapoints (8.9%)

- 68 (2.9 SDs)
- 299 (5.0 SDs)
- 319 (1.4e+01 SDs)
- 366 (2.2 SDs)
- 402 (1.3e+01 SDs)
- 404 (3.7 SDs)
- 406 (4.8 SDs)
- 424 (6.3 SDs)
- 425 (4.6 SDs)
- 427 (3.2 SDs)

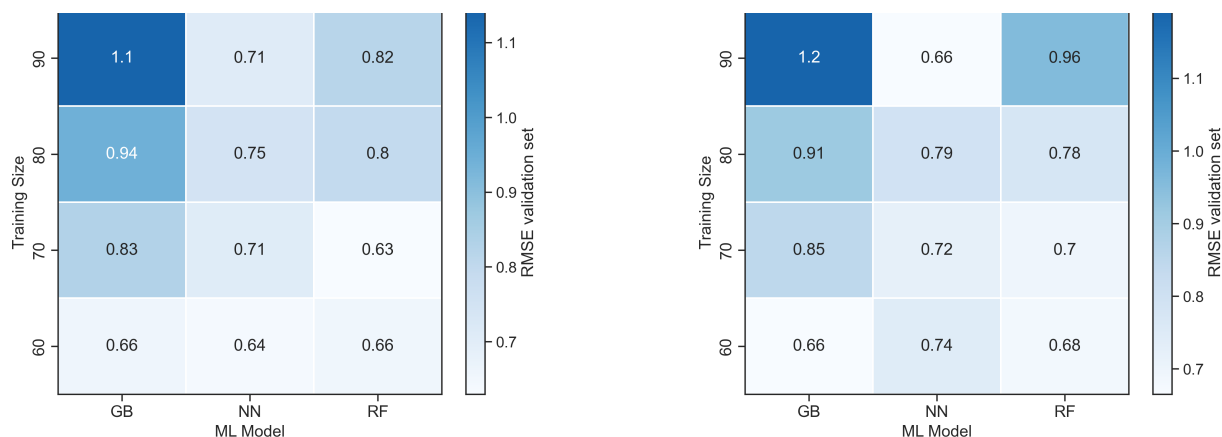
Test: 11 outliers out of 112 datapoints (9.8%)

- 301 (5.4 SDs)
- 908 (2.8 SDs)
- 560 (3.8 SDs)
- 747 (2.5 SDs)
- 683 (7.4 SDs)
- 378 (5.7 SDs)
- 991 (7.2 SDs)
- 363 (3.3 SDs)
- 710 (4.6 SDs)
- 324 (1e+01 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 (JetFuel\_Ignition\_Three\_Type.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==1.2.1`
- Install scikit-learn-intelex: `pip install scikit-learn-intelex==2024.7.0`

(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 "Point" --y "IDT_ms" --model "[RF,GB,NN]" --csv_name "JetFuel_Ignition_Three_Type.csv"
```

### 4. Execution time, Python version and OS:

Originally run in Python 3.12.2 using Linux #1 SMP Fri Apr 20 16:44:24 UTC 2018

Total execution time: 589.97 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: 19  
 names: Point  
 n\_estimators: 20  
 max\_depth: 20  
 max\_features: 1.0  
 min\_samples\_split: 2  
 min\_samples\_leaf: 1  
 min\_weight\_fraction\_leaf: 0  
 ccp\_alpha: 0  
 oob\_score: True  
 max\_samples: 0.75

#### PFI (only most important descriptors):

sklearn model: GradientBoostingRegressor  
 random\_state: 0  
 names: Point  
 n\_estimators: 60  
 max\_depth: 5  
 max\_features: 0.5  
 min\_samples\_split: 2  
 min\_samples\_leaf: 1  
 min\_weight\_fraction\_leaf: 0  
 ccp\_alpha: 0  
 learning\_rate: 0.05  
 subsample: 1.0  
 validation\_fraction: 0.3

### 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.*

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



## 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.
-