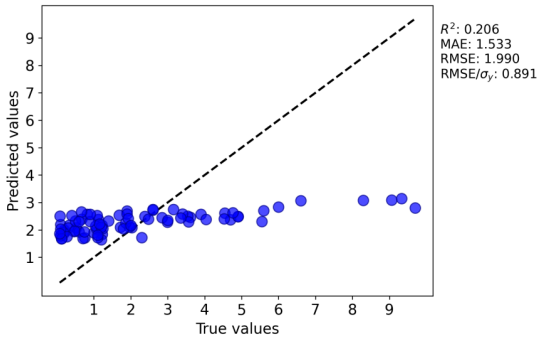
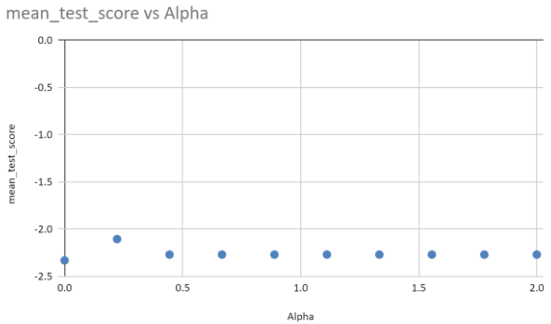
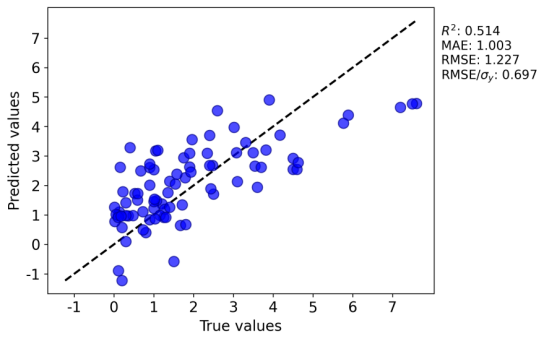
Assessment Figures



	alpha
Best Parameters	0.2222222222





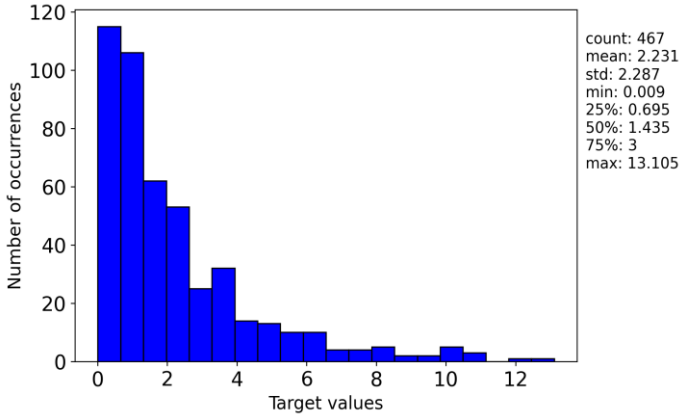
ML4ER - Assignment 5 Activities

Muhammad Zain Azeem,
Informatics Skunkworks (**non-credits),** Week 3
04/08/2024

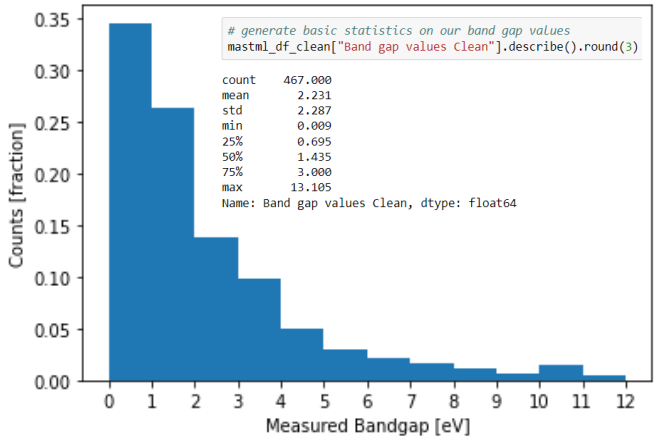
Progress

Question: Let's look through the outputs and compare them to some of the initial dataset analysis and compare to the previous Nanohub workflow. Open the "histogram_target_values.png" file in the newly created DataCleaning folder under our output directory. Compare back to the histogram we made in the previous notebook. Are they the same?

MAST-ML



Nanohub notebook



Conclusion: Results were found to be similar

Progress

Question:

Using the cell block below with outputs the feature object directly compare the features generated to those in the previous workflow. Do we have the same total number?

If they're different can you think of any reasons why?
hint: mastml does some initial cleaning automatically on the features.

MAST-ML

	AtomicNumber_composition_average	AtomicRadii_composition_average
0	41.000000	1.292000
1	32.000000	1.217000
2	17.000000	1.051000
3	49.500000	1.522000
4	34.000000	1.206000
...	...	...
462	36.000000	1.466667
463	35.500000	1.450000
464	32.000000	1.382500
465	35.428571	1.475714
466	23.000000	1.255000

467 rows × 87 columns

Nanohub notebook

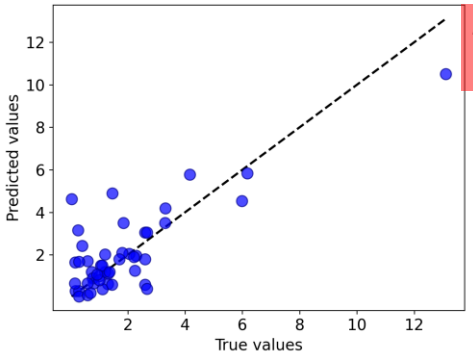
	AtomicNumber_composition_average	AtomicRadii_composition_average
0	6.000000	1.135000
1	10.000000	1.270000
2	19.000000	1.345000
3	15.000000	1.560000
4	28.000000	1.440000
...	...	...
462	60.500000	1.422500
463	83.000000	1.700000
464	35.333333	1.086000
465	50.000000	1.057500
466	36.000000	0.948333

467 rows × 87 columns

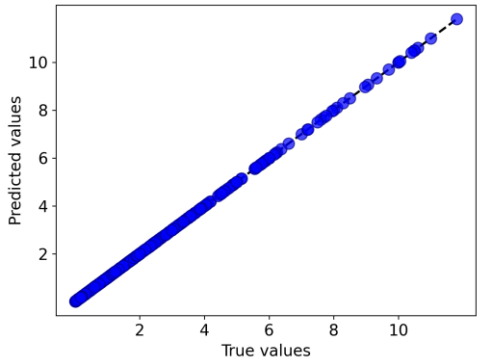
Conclusion: Different results were obtained

Progress

Question: After this run completes we want to go look at how the model is performing. Navigate to the newly created "DecisionTreeRegressor..." folder and find both the "parity_plot_leaveout.png" file as well as the "parity_plot_train.png" file. Compare them both to eachother as well as to the parity plots made during the Nanohub notebook for the default model. Are they the same? Similar?



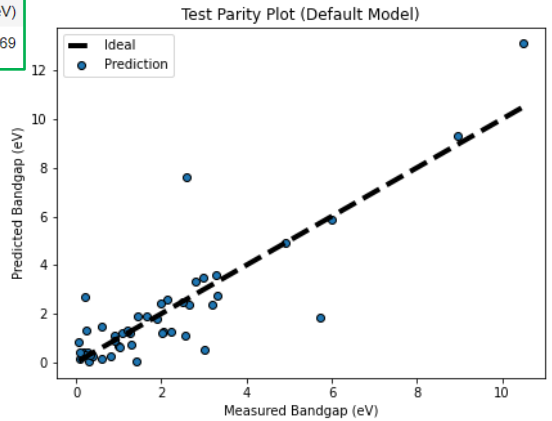
parity_plot_leaveout.png



parity_plot_train.png

	Error Metric	Test Set (Default Model)
0	RMSE	1.2398 (eV)
1	RMSE/std	0.5771
2	MAE	0.723 (eV)
3	R2	0.6669

Nanohub notebook (default model)



Conclusion: Nanohub results showed higher R^2: 0.669 then MAST-ML R^2: 0.622

Progress

Task:

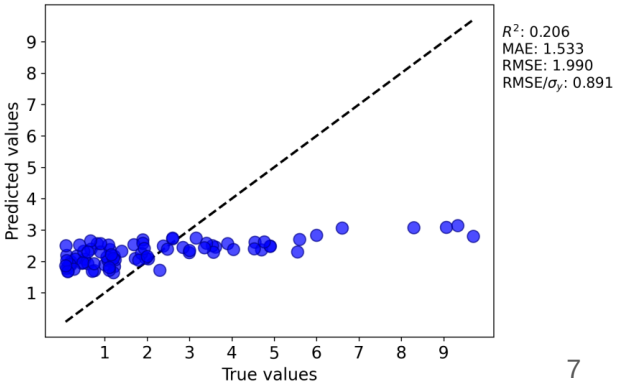
1) pick another model type from scikit-learn. You can see a reference for available models here: https://scikit-learn.org/stable/supervised_learning.html If you're not sure what kind of model to try I might suggest one of the linear type models such as Ridge Regression or LASSO. To see the list of available hyperparameters for each model you can click their respective link.

2) build a default model where you don't change any hyperparameters from the scikit-learn defaults and analyze it's performance both on the Test data and with a 5-fold CV

```
# Activity 5 - Default Model

# Selection of model "LASSO"
default_LASSO = SklearnModel(model='Lasso')
models = [default_LASSO]
selector = [NoSelect()]
metrics = ['r2_score', 'mean_absolute_error', 'root_mean_squared_error', 'rmse_over_stdev']
processor = SklearnPreprocessor(preprocessor='StandardScaler', as_frame=True)

# Set up the 5-fold cross-validation with 2 repeats
splitter = SklearnDataSplitter(splitter='RepeatedKFold', n_repeats=2, n_splits=5)

# Evaluate the model using cross-validation and test set
splitter.evaluate(X=X,
                  y=y,
                  models=models,
                  preprocessor=processor,
                  selectors=selector,
                  metrics=metrics,
                  savepath=savepath,
                  X_extra=X_extra,
                  leaveout_inds=X_testdata,
                  verbosity=3)
```

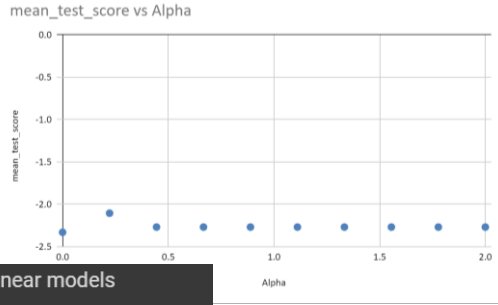
1) Selected "**LASSO**" model
2) Performed 5-CV and splitting

Results:



Progress

	alpha
Best Parameters	0.2222222222



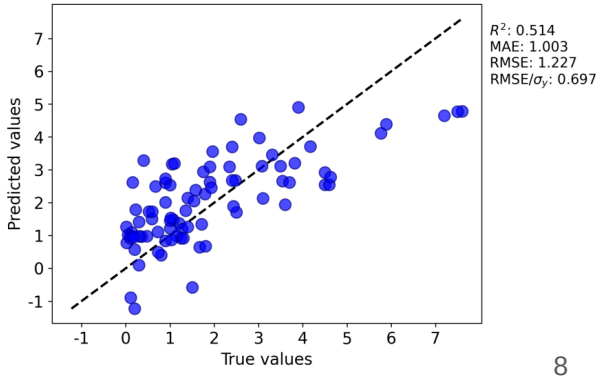
Task:

3) perform a grid search on 1 of the hyperparameters. I'd suggest picking the alpha hyperparameters if using one of the linear models suggested above.

```
# Activity 5

# Selection of model "LASSO" - Optimization
default_LASSO = SklearnModel(model='Lasso',alpha=0.22)
models = [default_LASSO]
selector = [NoSelect()]
metrics = ['r2_score', 'mean_absolute_error', 'root_mean_squared_error', 'rmse_over_stdev']
processor = SklearnPreprocessor(preprocessor='StandardScaler', as_frame=True)

# Set up the 5-fold cross-validation with 2 repeats
splitter = SklearnDataSplitter(splitter='RepeatedKFold', n_repeats=2, n_splits=5)

# Evaluate the model using cross-validation and test set
splitter.evaluate(X=X,
                  y=y,
                  models=models,
                  preprocessor=processor,
                  selectors=selector,
                  metrics=metrics,
                  savepath=savepath,
                  X_extra=X_extra,
                  leaveout_inds=X_testdata,
                  verbosity=3)
```

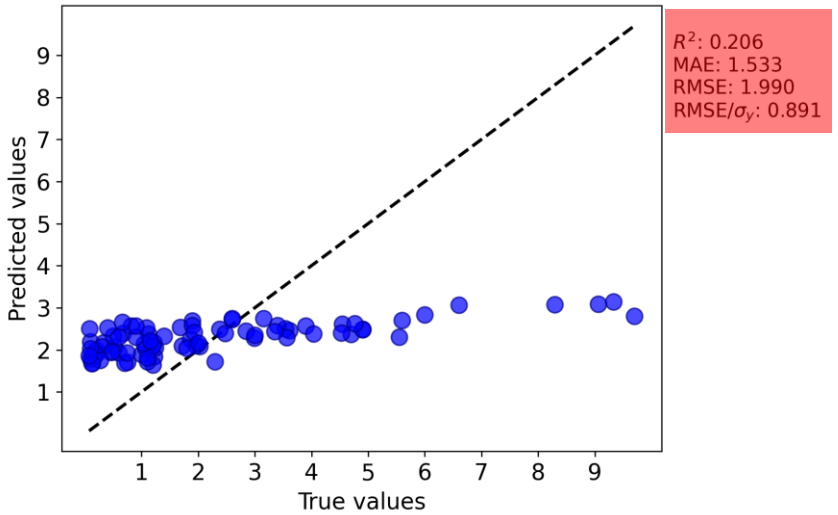
3) Performed grid search
4) Performed 5-CV and splitting

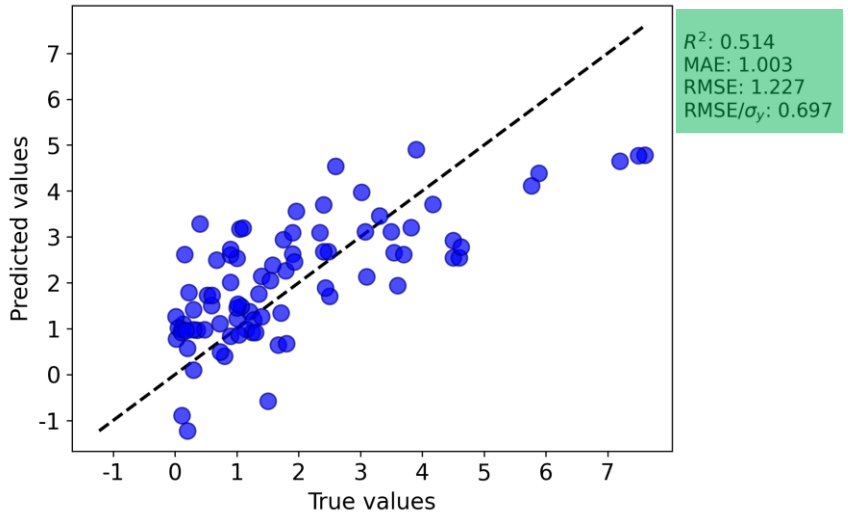
Results:



Progress

Task: 4) Compare the performance with the optimized hyperparameters. Were you able to improve the performance? how much did the RMSE value decrease for the Test set? How about the 5-fold CV test?





Conclusion: The model performed improved from R^2: 0.206 to R^2: 0.514