Evaluation of Automated Machine Learning(AutoML)

—

Honyu Xin, Junyi Li, Yanping Wang

Outline

- The Purpose of using AutoML for QSAR datasets
- Design of AutoML framework
- Result
- Conclusion and recommendation

Goals

- Development of end-to-end automated machine learning workflows is important in order to deploy them in a robust way in real life applications
- Design of AutoML framework and carry out evaluation on QSAR data sets .
- Stretch goal: design a systematic simulation study that would evaluate the AutoML framework in a controlled setting.

Design of AutoML framework

- Datasets(page5-6)
- Performance metrics
  - Using r^2 to evaluate prediction performance
  - Using less time to find the best model
- Hardware choice and resource specifications
  - 4 core Intel(R) Xeon(R) CPU E5-2670 0 @ 2.60GHz
  - Sufficient RAM
  - Build on Amarel cluster
- Frameworks and their configuration(page 7-11)

We use the framework to conduct a thorough comparison of 3 AutoML systems across 15 QSAR datasets and analyze the results.

Data

The Training and Test sets each consist of 15 biological activity data sets in comma separated value(csv) format.

Why we need automl for QSAR datasets?

QSAR datasets are biological or physico-chemical properties of molecules description. With the help of automl, we can find the medicine structure more efficiently.

QSAR_1_train_x.csv	May 27, 2020 at 4:42 PM	684.5 MB
QSAR_6_train_x.csv	May 27, 2020 at 4:51 PM	645.7 MB
QSAR_10_train_x.csv	May 27, 2020 at 4:54 PM	122.1 MB
QSAR_8_train_x.csv	May 27, 2020 at 4:52 PM	111.8 MB
QSAR_2_train_x.csv	May 27, 2020 at 4:43 PM	97.1 MB
QSAR_12_train_x.csv	May 27, 2020 at 4:55 PM	86.6 MB
QSAR_13_train_x.csv	May 27, 2020 at 4:56 PM	67.7 MB
QSAR_3_train_x.csv	May 27, 2020 at 4:44 PM	61.9 MB
QSAR_11_train_x.csv	May 27, 2020 at 4:54 PM	60.7 MB
QSAR_15_train_x.csv	May 27, 2020 at 4:57 PM	53.6 MB
QSAR_9_train_x.csv	May 27, 2020 at 4:53 PM	49.4 MB
QSAR_14_train_x.csv	May 27, 2020 at 4:56 PM	47.7 MB
QSAR_5_train_x.csv	May 27, 2020 at 4:44 PM	37.1 MB
QSAR_4_train_x.csv	May 27, 2020 at 4:44 PM	15.3 MB
QSAR_7_train_x.csv	May 27, 2020 at 4:51 PM	13.8 MB
QSAR_1_train_y.csv	May 27, 2020 at 4:42 PM	259 KB
QSAR_6_train_y.csv	May 27, 2020 at 4:51 PM	197 KB
QSAR_10_train_y.csv	May 27, 2020 at 4:54 PM	77 KB
QSAR_8_train_y.csv	May 27, 2020 at 4:52 PM	69 KB
QSAR_2_train_y.csv	May 27, 2020 at 4:43 PM	60 KB
QSAR_12_train_y.csv	May 27, 2020 at 4:55 PM	59 KB
QSAR_11_train_y.csv	May 27, 2020 at 4:54 PM	46 KB
QSAR_3_train_y.csv	May 27, 2020 at 4:44 PM	42 KB
QSAR_13_train_y.csv	May 27, 2020 at 4:56 PM	42 KB
QSAR_9_train_y.csv	May 27, 2020 at 4:53 PM	37 KB
QSAR_15_train_y.csv	May 27, 2020 at 4:57 PM	35 KB
QSAR_14_train_y.csv	May 27, 2020 at 4:56 PM	30 KB
QSAR_5_train_y.csv	May 27, 2020 at 4:44 PM	22 KB
QSAR_4_train_y.csv	May 27, 2020 at 4:44 PM	13 KB
QSAR_7_train_y.csv	May 27, 2020 at 4:51 PM	6 KB

Data

The challenge is to predict the activity value for each molecule/data set combination in the test set.

- Column 1-9177: Molecular descriptors/features
- Last column: Activity

```
print(QSAR_1.shape)
QSAR_1.head()
```

```
(37241, 9178)
```

	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10	...	x9169	x9170	x9171	x9172	x9173	x9174	x9175	x9176	x9177	y
0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	6.0179
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	4.3003
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	5.2697
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	6.1797
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	4.3003

5 rows × 9178 columns

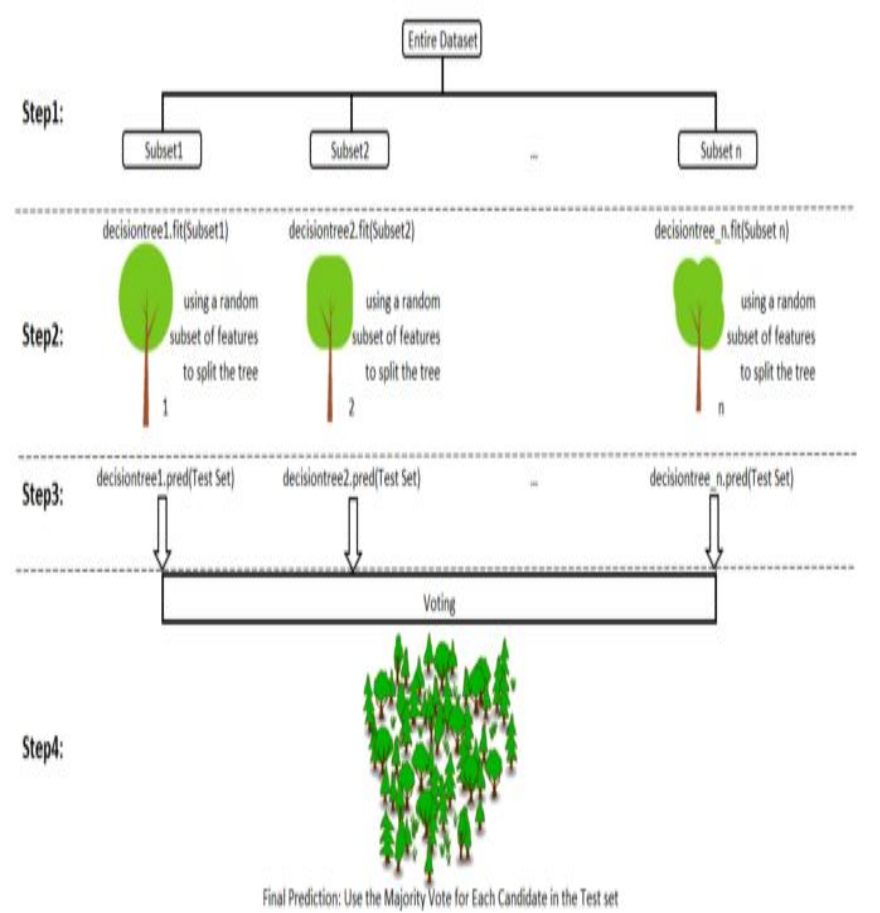
Frameworks and their configuration

- Baseline Model: Random Forest
- AutoMl Framework:
  - Auto-sklearn(Bayesian Optimization)
  - H20 AutoML(Random Search)
  - TPOT(Genetic Programming)

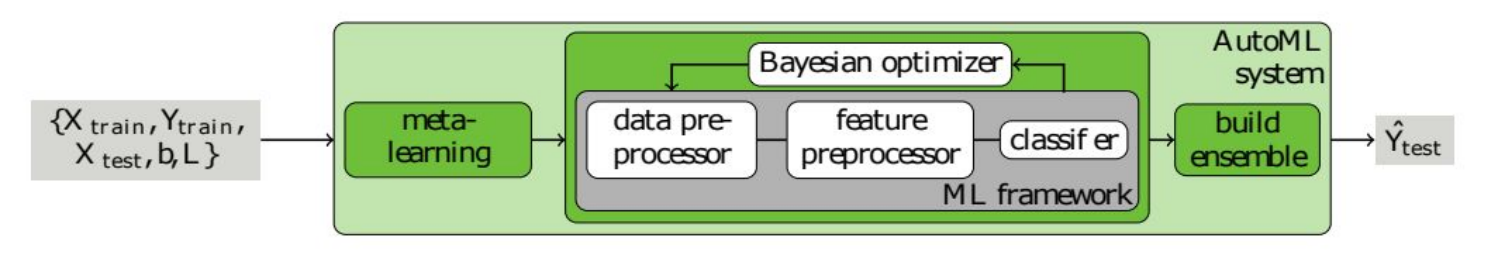
Random forest

- RF are among the most predictive and have few adjustable parameters
- Due to its high prediction accuracy, ease of use, and robustness to adjustable parameters, RF has been something of a “gold standard” to which other QSAR methods are compared.
- used with the default hyperparameter values and search spaces

```
reg = RandomForestRegressor(bootstrap=True, ccp_alpha=0.0, criterion='mse',
                max_depth=None, max_features='log2', max_leaf_nodes=None,
                max_samples=None, min_impurity_decrease=0, min_samples_leaf=1,
                min_samples_split=2, min_weight_fraction_leaf=0,
                n_estimators=100, n_jobs=None, oob_score=False,
                random_state=None, verbose=0, warm_start=False)
```



Auto-sklearn



Auto-sklearn which used Bayesian optimization to select and tune the algorithms in a machine learning pipeline and also added meta-learning to warm-start the search with the best pipelines on similar datasets, as well as ensemble construction.

```
reg = autosklearn.regression.AutoSklearnRegressor(
    time_left_for_this_task=3600,
    per_run_time_limit=600,
    n_jobs=4,
    memory_limit=20*1024,
    tmp_folder='/home/hx152/tmp/autosklearn_regression_example_tmp',
    metric=autosklearn.metrics.r2)
reg.fit(X_train, y_train, dataset_name='QSAR')
```

H2o.ai

H2o uses random grid search to optimize the hyperparameters.

H2o redesigned the data structure. That's the reason why H2o has better memory management and multi-thread performance.

Excellent industrial implementation.

```
aml = H2OAutoML(max_runtime_secs=3600, exclude_algos=['DeepLearning'], seed = 1, verbosity="NULL")
aml.train(x=x_names, y=y_names, training_frame=hf_train)
```

Tpot

Tree-Based Pipeline Optimization Tool, is a genetic programming-based optimizer that generates machine learning pipelines.

```
print('start')
tpot = TPOTRegressor(random_state=1, max_time_mins=60, n_jobs=4, verbosity=3, generations=10, population_size=20)
tpot.fit(X_train, y_train)
print('end')
```

Experiment design

- The performance of fitting in different datasets

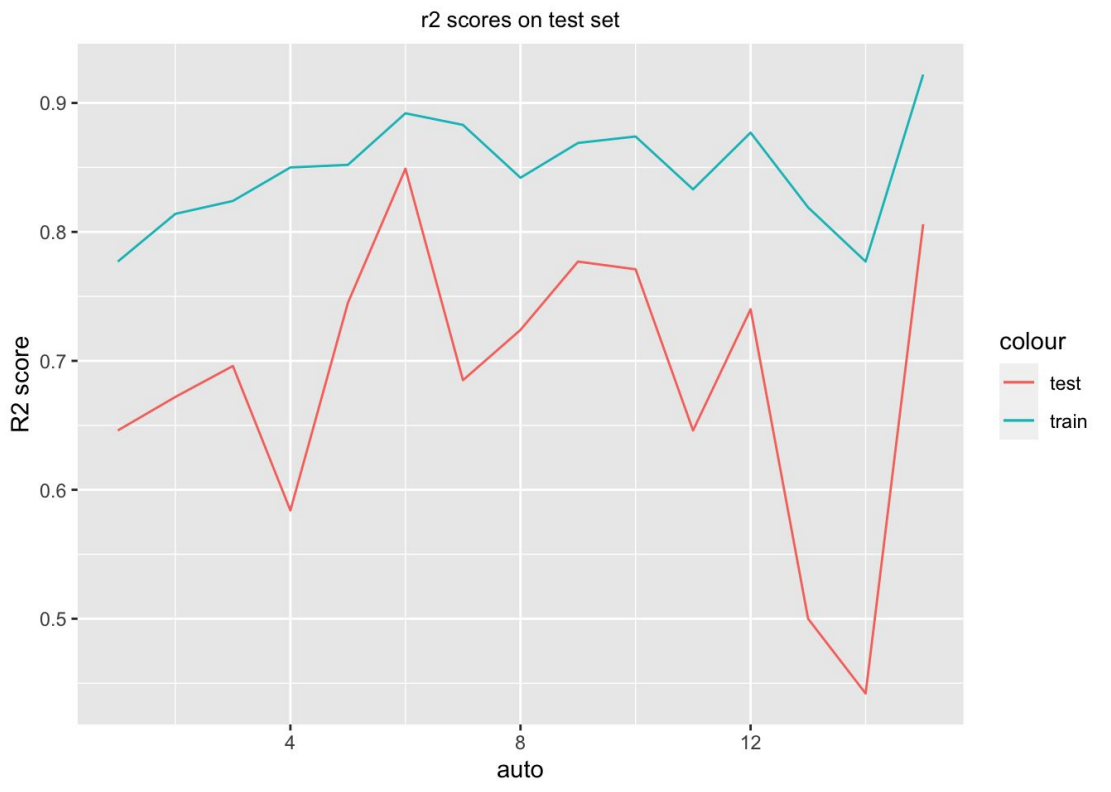
  We set fitting time limit as 1 hour, compare r2 of different framework in same dataset.

- The performance of fitting in different time limit

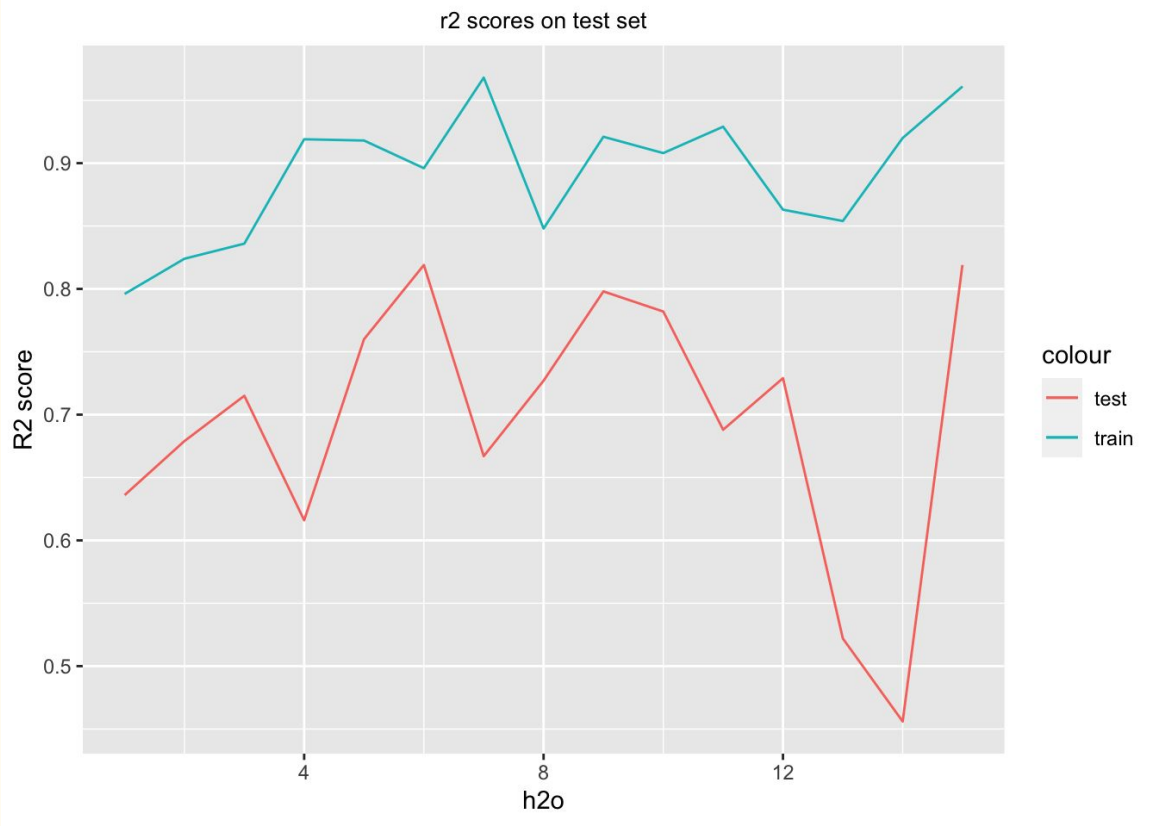
  We set different time limit, compare r2 of different framework in same dataset.

- Memory management and usability

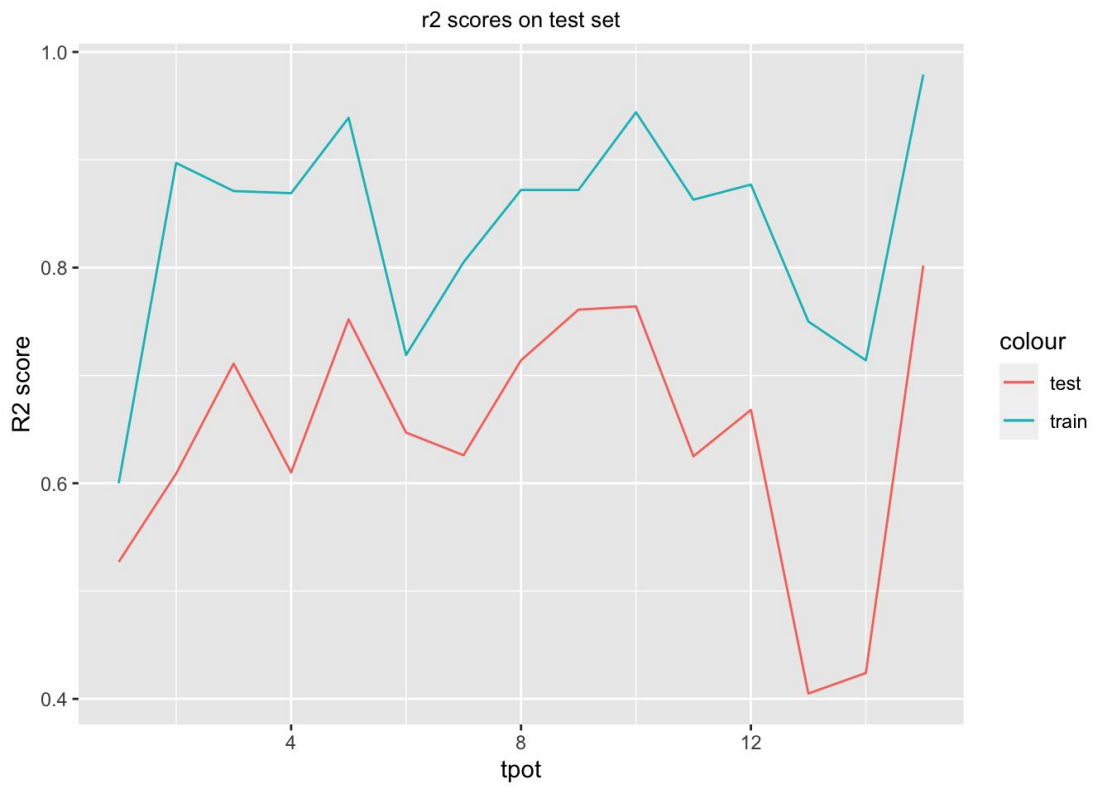
The result of auto-sklearn



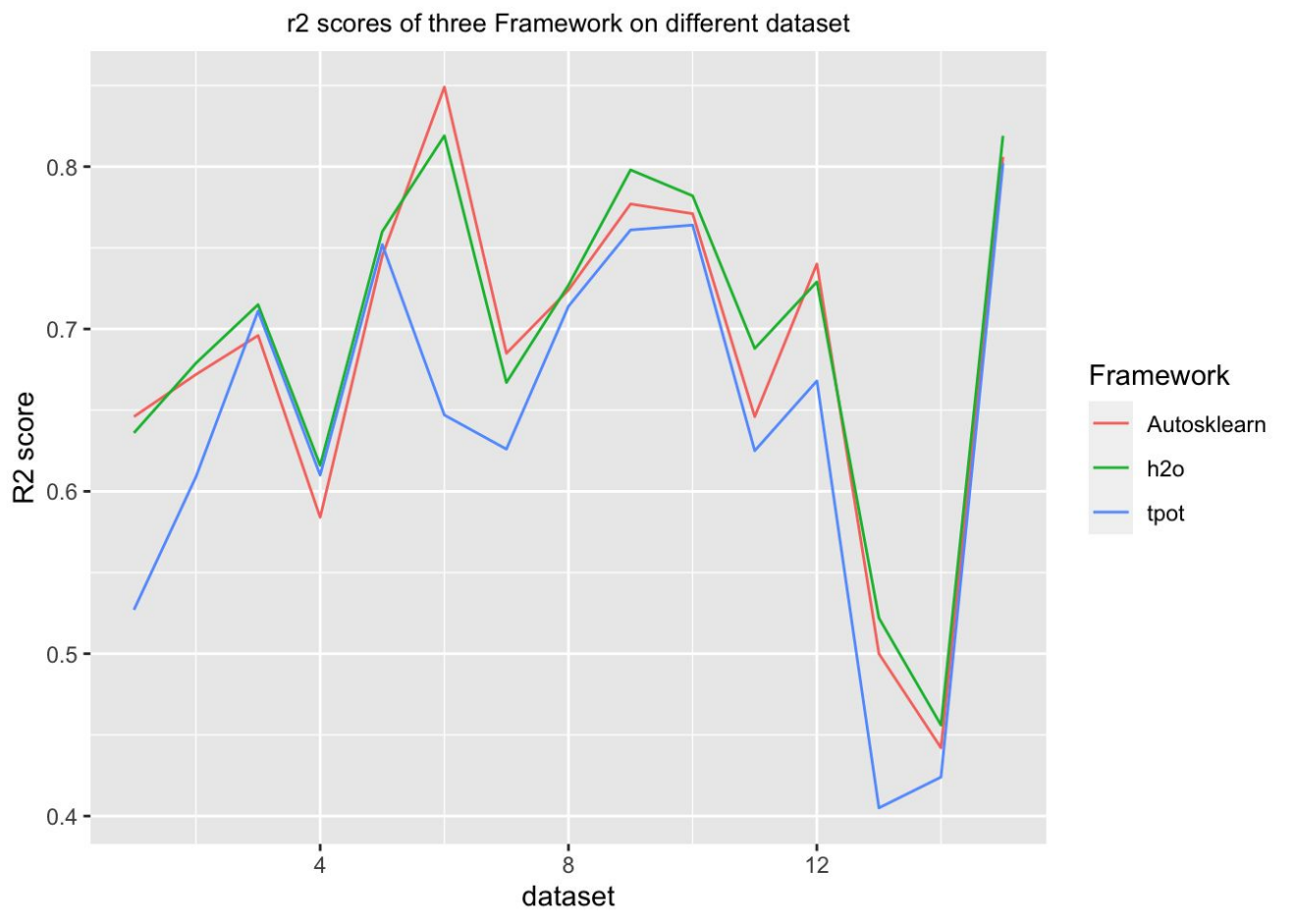
The result of H2o.ai



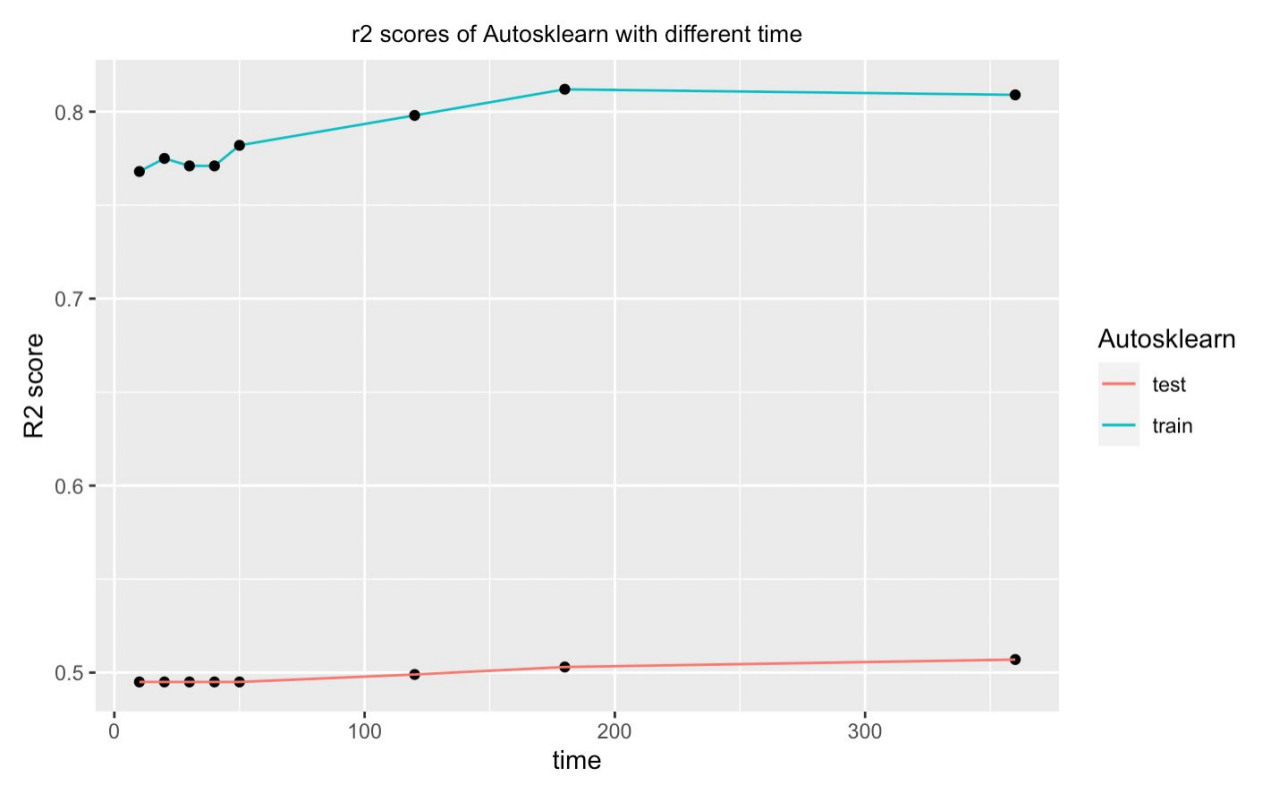
The result of tpot



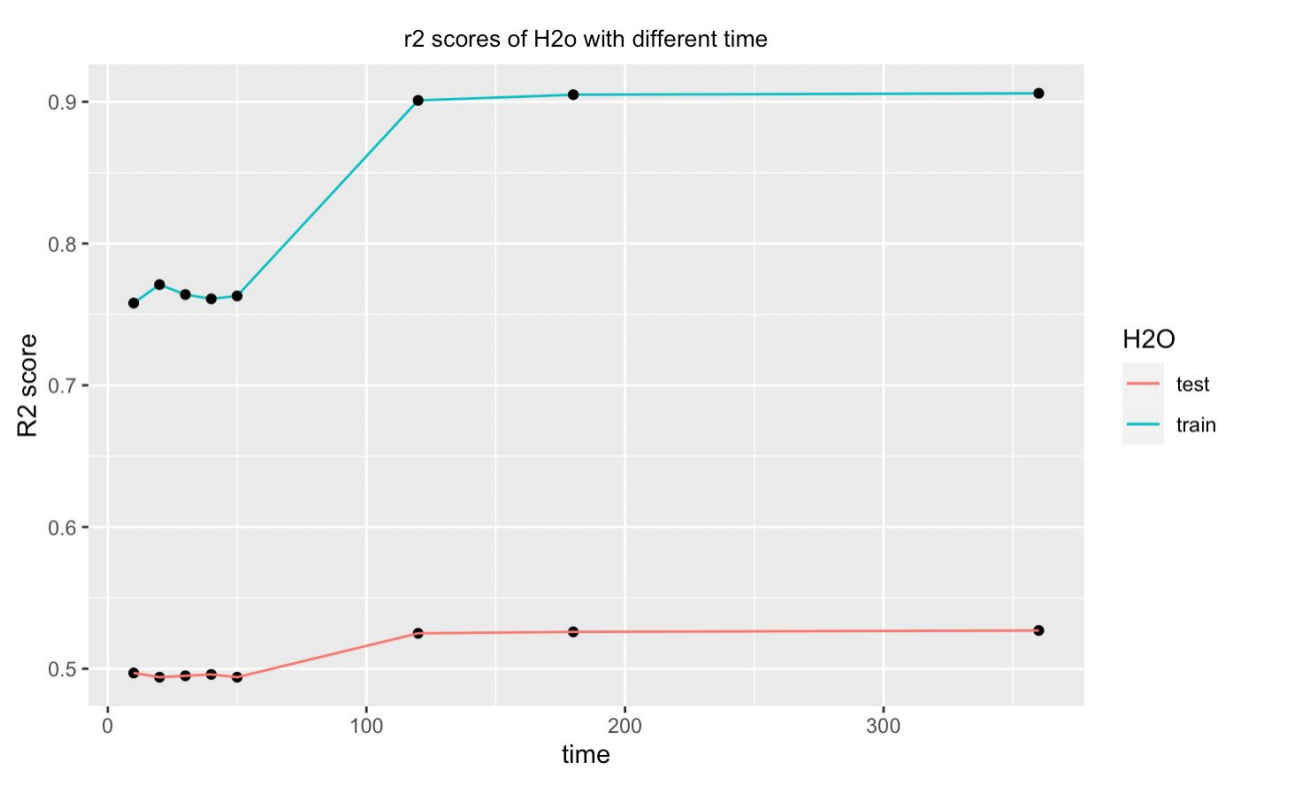
The result of different dataset



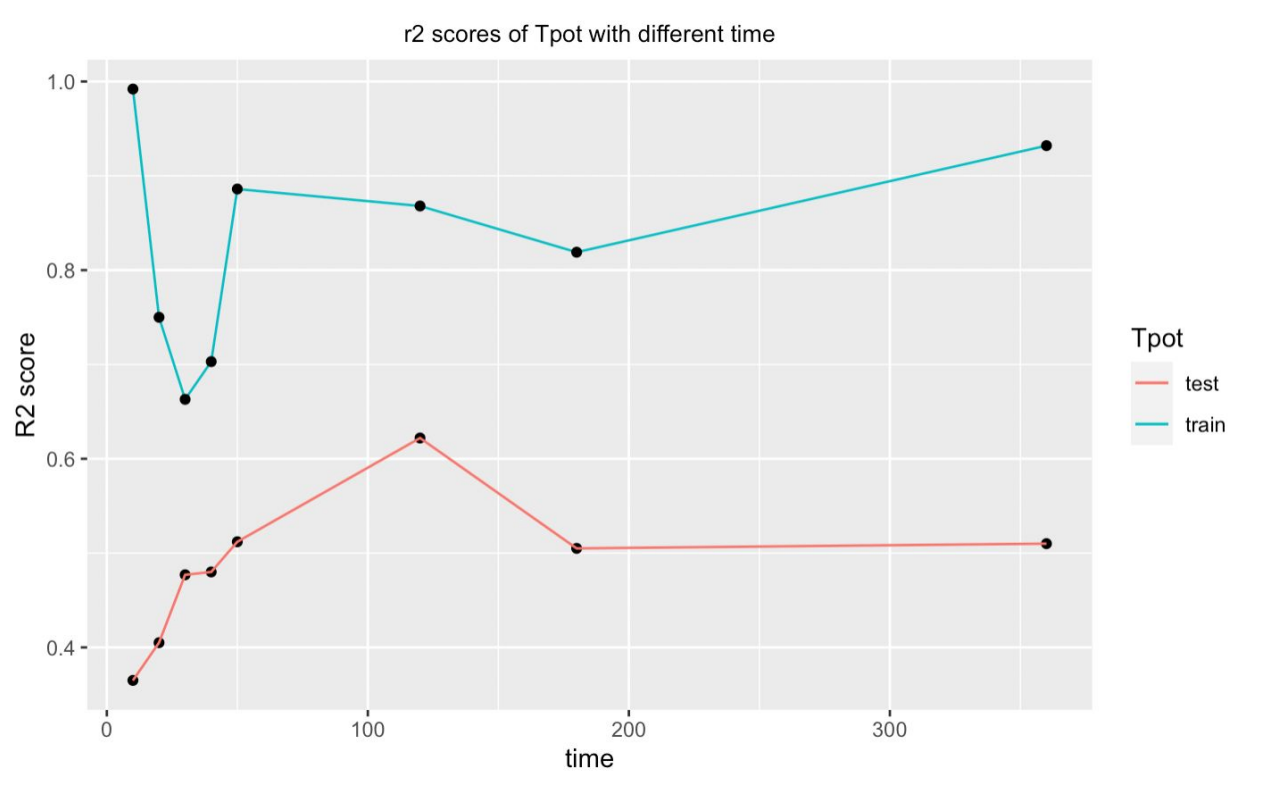
The result of auto-sklearn



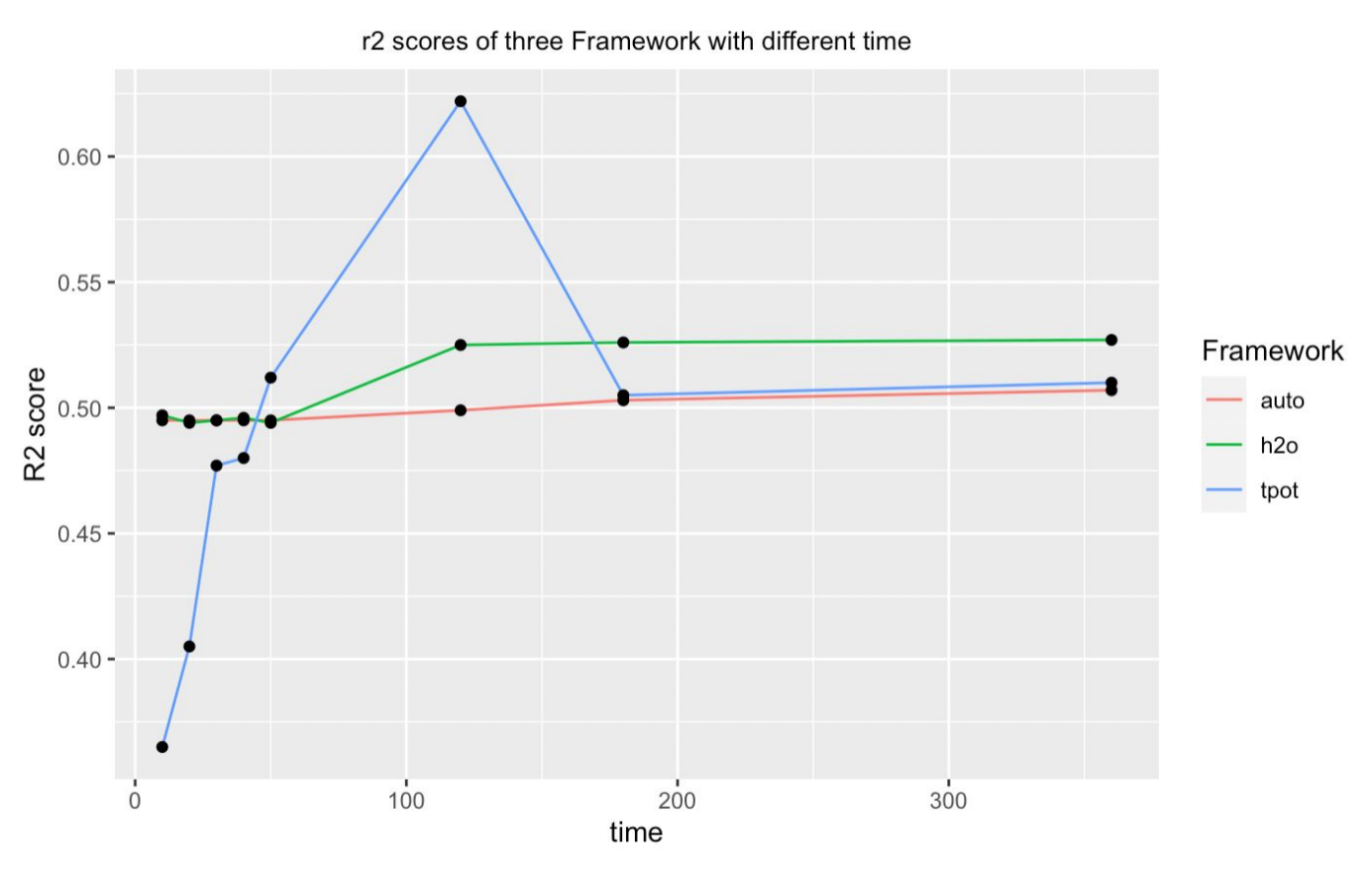
The result of H2o.ai



The result of tpot



The result of different time



Memory management and usability

Auto-sklearn

Data_storage memory cost * threads + 20G memory

H20 AutoML

Data_storage memory cost + 20G memory

TPOT

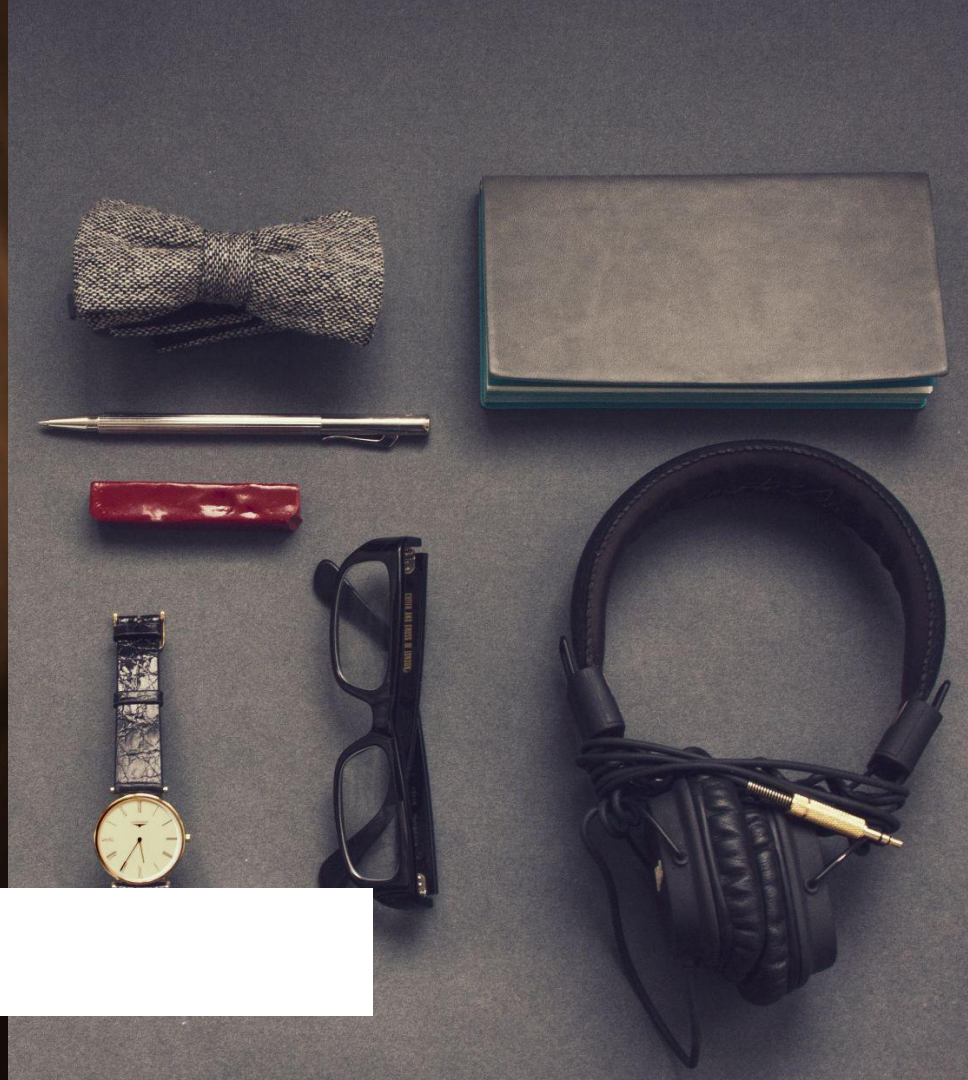
Hard to estimate. The memory cost will increase with the grow of the searching space.

Conclusion and Recommendations

H2o.ai is the best framework for QSAR dataset.

Github : https://github.com/arthurxin/workflow_of_automl_framework

Code



That’s all, thank you

References

https://arxiv.org/pdf/1907.00909.pdf

https://openml.github.io/automlbenchmark/paper.html

https://arxiv.org/pdf/1808.06492.pdf