<https://github.com/maravenag/bayesian-autoML/blob/master/bayesian_h2o.ipynb>

<https://medium.com/spikelab/hyperparameter-optimization-using-bayesian-optimization-f1f393dcd36d>

!pip install h2o

import h2o

from h2o.estimators.gbm import H2OGradientBoostingEstimator

from bayes\_opt import BayesianOptimization

h2o.init()

h2o.remove\_all()

data = h2o.upload\_file('winequality-red.csv')

train\_cols = [x for x in data.col\_names if x not in ['quality']]

target = "quality"

train, test = data.split\_frame(ratios=[0.7])

def train\_model(max\_depth,

                ntrees,

                min\_rows,

                learn\_rate,

                sample\_rate,

                col\_sample\_rate):

    params = {

        'max\_depth': int(max\_depth),

        'ntrees': int(ntrees),

        'min\_rows': int(min\_rows),

        'learn\_rate':learn\_rate,

        'sample\_rate':sample\_rate,

        'col\_sample\_rate':col\_sample\_rate

    }

    model = H2OGradientBoostingEstimator(nfolds=5,\*\*params)

    model.train(x=train\_cols, y=target, training\_frame=train)

    return -model.rmse()

bounds = {

    'max\_depth':(5,10),

    'ntrees': (100,500),

    'min\_rows':(10,30),

    'learn\_rate':(0.001, 0.01),

    'sample\_rate':(0.5,0.8),

    'col\_sample\_rate':(0.5,0.8)

}

optimizer = BayesianOptimization(

    f=train\_model,

    pbounds=bounds,

    random\_state=1,

)

optimizer.maximize(init\_points=10, n\_iter=50)

iter | target | col\_sa... | learn\_... | max\_depth | min\_rows | ntrees | sample... |

------------------------------------------------------------------------------------------------------------------

| 13 | -0.292 | 0.8 | 0.01 | 10.0 | 10.0 | 500.0 | 0.8 |

| 15 | -0.6859 | 0.5 | 0.001 | 5.0 | 10.0 | 500.0 | 0.5 |

| 23 | -0.647 | 0.8 | 0.001441 | 10.0 | 30.0 | 500.0 | 0.8 |

optimizer**.**max

OUTPUT

{'target': -0.2920199343134594,

'params': {'col\_sample\_rate': 0.8,

'learn\_rate': 0.0099999,

'max\_depth': 10.0,

'min\_rows': 10.0,

'ntrees': 500.0,

'sample\_rate': 0.8}}

RANDOM FOREST PARAMETERS

[max\_depth](https://docs.h2o.ai/h2o/latest-stable/h2o-docs/data-science/algo-params/max_depth.html): Specify the maximum tree depth. defaults to 20.

[min\_rows](https://docs.h2o.ai/h2o/latest-stable/h2o-docs/data-science/algo-params/min_rows.html): minimum number of observations for a leaf defaults to 1.

[ntrees](https://docs.h2o.ai/h2o/latest-stable/h2o-docs/data-science/algo-params/ntrees.html): Specify the number of trees (defaults to 50).

[sample\_rate](https://docs.h2o.ai/h2o/latest-stable/h2o-docs/data-science/algo-params/sample_rate.html): Specify the row sampling rate (x-axis). (Note that this method is sample without replacement.) The range is 0.0 to 1.0, and this value defaults to 0.6320000291.

* [nbins](https://docs.h2o.ai/h2o/latest-stable/h2o-docs/data-science/algo-params/nbins.html): (Numerical/real/int only) number of bins for the histogram to build, defaults to 20.
* [nbins\_top\_level](https://docs.h2o.ai/h2o/latest-stable/h2o-docs/data-science/algo-params/nbins_top_level.html): (For numerical/real/int columns only) Specify the minimum number of bins at the root level to use to build the histogram. This number will then be decreased by a factor of two per level. This option defaults to 1024.
* [nbins\_cats](https://docs.h2o.ai/h2o/latest-stable/h2o-docs/data-science/algo-params/nbins_cats.html): (Categorical/enums only) Specify the maximum number of bins for the histogram to build, the