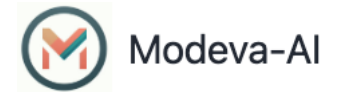
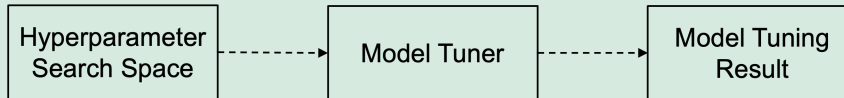


Model Tuning with Modeva :: CHEATSHEET



Model Tuning is to find the optimal hyperparameters that maximizes the performance of a machine learning model on a given task. Modeva provides multiple model tuners that do the search.



```
from modeva.models import
    ModelTuneGridSearch,
    ModelTuneRandomSearch,
    ModelTuneOptuna,
    ModelTunePSO
```

Grid Search

```
hpo = ModelTuneGridSearch(dataset=ds, model=model)
result = hpo.run(param_grid=param_grid,
    metric=("ACC", "AUC", "Brier", "LogLoss"))
```

Random Search

```
hpo = ModelTuneRandomSearch(dataset=ds, model=model)
result = hpo.run(param_distributions=param_space,
    n_iter=20, metric=("ACC", "AUC"))
```

Optuna Tuner

```
import optuna
hpo = ModelTuneOptuna(dataset=ds, model=model)
result = hpo.run(param_distributions=param_space,
    sampler="tpe", #{"grid","random","tpe","gp","cma-es","qmc"}
    cv=5, metric=("ACC", "AUC", "LogLoss"))
```

PSO (Particle Swarm Optimization) Search

```
hpo = ModelTunePSO(dataset=ds, model=model)
result = hpo.run(param_bounds=param_bounds,
    param_types=param_types,
    n_iter=2, n_particles=10, metric="AUC")
```

Hyperparameter Search Space

Example: Regular Grid

```
param_grid = {
    "n_estimators": [100, 500],
    "eta": [0.001, 0.01, 0.1]}
```

Example: Random Space

```
from scipy.stats import uniform, randint
param_space = {
    "max_depth": randint(1, 10),
    "n_estimators": randint(100, 1000),
    "eta": uniform(0.001, 0.3)}
```

Example: PSO Bounds and Types

```
param_bounds = {
    "max_depth": [1, 10],
    "eta": [0.001, 0.3]}
param_bounds = {
    "max_depth": "int",
    "eta": "float"}
```

Model Tuning Result

```
result.table
result.plot("parallel", figsize=(8, 5))
result.plot("x:param", "y:metric")
```