

CatBoost cheat sheet

Important CatBoost classes

```
# CatBoost feature data type
from catboost import Pool
# cross-validation generator
from catboost import cv
# classifier
from catboost import CatBoostClassifier
# regressor
from catboost import CatBoostRegressor
# general purpose (e.g. ranking)
from catboost import CatBoost
```

Pool

```
train_set = Pool(data, # data
                 labels, # labels
                 # indices/column names of
                 # categorical features
                 cat_features,
                 # workers to read from files:
                 thread_count=-1
                 # ... more functions
                 # to define how input
                 # data should be handled
                 )
```

Classifier/Regressor options

```
params = {'iterations': # int,
         'learning_rate': # float,
         'depth': # range [1,16],
         'task_type': 'GPU' # CPU/GPU,
         ''
        )
clf = CatBoostClassifier(params)
```

Attributes

```
# similar to scikit-learn
clf = CatBoostClassifier()
clf.fit()
clf.predict()
clf.predict_proba()

# CatBoost-specific
# cross-validation
# (can be passed to grid_search)
scores = cv(train_set # pool,
            params # model params,
            fold_count # (int) folds,
            shuffle # bool,
            stratified # bool,
            )
```

gridsearch for hyperparameter tuning

```
clf.grid_search(params # dict,
               X # Pool or list,
               y # list or none (if Pool),
               cv # cv function or no. of folds
               )
```

```
# compare two CatBoost models
clf.compare(otherCBModel,
            evalDataset,
            metric # list of metric(s))
```

Model I/O

Save (export) models

```
# Python format
clf.save_model(filename, format="python")

# C++ format
clf.save_model(filename, format="CPP")

# JSON format
clf.save_model(filename, format="json")

# PMML format
clf.save_model(filename, format="pmml")

# CoreML format
clf.save_model(filename, format="coreml",
               export_parameters={
                   'prediction_type': predictionType
               })

# Java, Rust (CatBoost binary model format)
clf.save_model(filename, format="cbm")
```

Load (import) models

```
clf = CatBoostClassifier()

# Python format
clf.load_model(filename, format="python")

# C++ format
clf.load_model(filename, format="CPP")

# CoreML format
clf.load_model(filename, format="AppleCoreML")

# CatBoost binary format
clf.load_model(filename, format="cbm")
```

Other hints

- Many attributes have a option called “plot” to plot processes/metrics/etc.
- Do not use One-Hot-Encoding with CatBoost.