adaboost_find_hyperparameter

January 30, 2019

```
In [1]: import numpy
        X = numpy.loadtxt("./data/Train/X_train.txt")
        y = numpy.loadtxt("./data/Train/y_train.txt")
```

```
In [2]: from models import adaboost
        best_model = adaboost.find_hyperparameter(X,y)
```

```
cross validation scores of best moddel are : [0.85622593 0.84768638 0.85649936 0.84912959 0.8572
mean of cross validation scores of best model is: 0.8533552796860722
```

```
In [3]: best_model.get_params()
```

```
Out[3]: {'algorithm': 'SAMME.R',
         'base_estimator__class_weight': None,
         'base_estimator__criterion': 'gini',
         'base_estimator__max_depth': 4,
         'base_estimator__max_features': None,
         'base_estimator__max_leaf_nodes': None,
         'base_estimator__min_impurity_decrease': 0.0,
         'base_estimator__min_impurity_split': None,
         'base_estimator__min_samples_leaf': 1,
         'base_estimator__min_samples_split': 2,
         'base_estimator__min_weight_fraction_leaf': 0.0,
         'base_estimator__presort': False,
         'base_estimator__random_state': None,
         'base_estimator__splitter': 'best',
         'base_estimator': DecisionTreeClassifier(class_weight=None, criterion='gini', max_depth
                     max_features=None, max_leaf_nodes=None,
                     min_impurity_decrease=0.0, min_impurity_split=None,
                     min_samples_leaf=1, min_samples_split=2,
                     min_weight_fraction_leaf=0.0, presort=False, random_state=None,
                     splitter='best'),
         'learning_rate': 0.6,
         'n_estimators': 104,
         'random_state': None}
```