

2_bagging_exercise

May 31, 2019

```
In [ ]: import numpy as np
        from sklearn.tree import DecisionTreeClassifier
        from sklearn.ensemble import BaggingClassifier

In [ ]: X = np.load("./tatanic_X_train.npy")
        y = np.load("./tatanic_y_train.npy")

In [ ]: from sklearn.model_selection import train_test_split
        X_train, X_test, y_train, y_test = train_test_split(X, y, \
                                                            test_size=0.3, random_state=101)

In [ ]: print(X_train.shape)
        print(y_train.shape)
        print(X_test.shape)
        print(y_test.shape)

In [ ]: X_train[0]

In [ ]: y_train[:10]

In [ ]: X_test[0]

In [ ]: y_test[:10]

In [ ]: clf2 = DecisionTreeClassifier(random_state=1)
        eclf = BaggingClassifier(clf2, oob_score=True)

In [ ]: clf2.fit(X_train, y_train)
        preds = clf2.predict(X_test)
        clf2.score(X_test, y_test)

In [ ]: eclf.fit(X_train, y_train)

In [ ]: predictions = eclf.predict(X_test)

In [ ]: eclf.score(X_test, y_test)

In [ ]:
```

```

In [ ]: from sklearn.metrics import confusion_matrix, classification_report

In [ ]: confusion_matrix(y_test, predictions)

In [ ]: print(classification_report(y_test, predictions))

In [ ]: from sklearn.model_selection import cross_val_score
        cross_val_score(eclf, X, y, cv=5).mean()

In [ ]: # parameter bagging parameter.

        params = {
            "n_estimators" : [10,20,30,40,50,55], # estimator
            "max_samples" : [0.5,0.6,0.7,0.8,0.9,1] # sample
        }

In [ ]: # ensemble(BaggingClassifier) + overfittingtunning(CV,GSCV) +
        # crossval(CV) + paramtunning(GridSearchCV) ->

        # quiz - gridsearch operation

In [ ]: grid.best_score_

In [ ]: grid.best_params_

In [ ]: grid.best_estimator_.oob_score_ # oob

In [ ]: grid.best_estimator_

In [ ]: # confusion matrix

In [ ]: # classification_report

In [ ]: grid.score(X_test, y_test)

In [ ]:

```