

1_ensemble_exercise

May 30, 2019

```
In [ ]: import numpy as np
        from sklearn.tree import DecisionTreeClassifier
        from sklearn.linear_model import LogisticRegression
        from sklearn.naive_bayes import GaussianNB
        from sklearn.ensemble import VotingClassifier

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

In [ ]: X[:5] # x features

In [ ]: y[:5]

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)

In [ ]: clflog = LogisticRegression(random_state=1)
        clfdt = DecisionTreeClassifier(random_state=1)
        clfgn = GaussianNB()
        eclf_h = VotingClassifier(estimators=[('lr', clflog), ('rf', clfdt), \
                                                ('gnb', clfgn)], voting='hard')
        eclf_s = VotingClassifier(estimators=[('lr', clflog), ('rf', clfdt), \
                                                ('gnb', clfgn)], voting='soft')

In [ ]: models = [clflog, clfdt, clfgn, eclf_h, eclf_s]

In [ ]: for model in models:
        model.fit(x_train, y_train)
        pred = model.predict(x_test)
        score = model.score(x_test, y_test)
        print(model)
        print(score)
        print('-'*20)

In [ ]: from sklearn.model_selection import cross_val_score
        for model in models:
            scores = cross_val_score(model, x_train, y_train, cv=5)

            print(scores)
            print(scores.mean())
            print('-'*20)
```

```

In [ ]: clf1 = LogisticRegression(random_state=1)
        clf2 = DecisionTreeClassifier(random_state=1)
        eclf1 = VotingClassifier(estimators=[('lr', clf1), ('dt', clf2)], voting='hard')
        eclf2 = VotingClassifier(estimators=[('lr', clf1), ('dt', clf2)], voting='soft')

In [ ]: models = [clf1, clf2, eclf1, eclf2]

        for model in models:
            scores = cross_val_score(model, x_train, y_train, cv=5)

            print(scores)
            print(scores.mean())
            print('-'*20)

In [ ]: # eclf = VotingClassifier(estimators=[('lr', clf1), ('dt', clf2)], voting='hard')

In [ ]: c_params = [0.1, 5.0, 7.0, 10.0, 15.0, 20.0, 100.0]

        params = {
            "lr__solver" : ['liblinear'], "lr__penalty" : ["l2"], \
            "lr__C" : c_params, "dt__criterion" : ["gini", "entropy"],
            "dt__max_depth" : [10,8,7,6,5,4,3,2],
            "dt__min_samples_leaf": [1,2,3,4,5,6,7,8,9]
        }

In [ ]: # cross_validation(cv), grid_search(param_grid), ensemble(estimator) =>

        from sklearn.model_selection import GridSearchCV
        grid = GridSearchCV(estimator=eclf, param_grid=params, cv=5)
        grid = grid.fit(X, y)

In [ ]: grid.best_score_

In [ ]: grid.best_params_

In [ ]: c_params = [0.1, 5.0, 7.0, 10.0, 15.0, 20.0, 100.0]

        params = {"solver" : ['liblinear'], "penalty" : ["l2"], "C" : c_params}
        grid = GridSearchCV(clf1, param_grid=params, cv=5)
        grid = grid.fit(X, y)

In [ ]: grid.best_score_

In [ ]: grid.best_params_

In [ ]:

```