

In []: 【ハイパーパラメタ・チューニング】

問7

XGBoostをグリッドサーチでチューニングしてみてください。

やり方は、ランダムフォレストの時と同じですが、**'params'**で設定したパラメータの内容で違うものがあります。問5で調べたものを使ってください。

In [1]: **import** pandas **as** pd

```
df = pd.read_csv('train.csv')
df = df.drop(['PassengerId', 'Age', 'SibSp', 'Parch', 'Ticket',
             'Cabin', 'Name'], axis=1)
df['Embarked'] = df['Embarked'].fillna('S')
df['Embarked'] = df['Embarked'].map({'S': 0, 'C': 1, 'Q': 2})
df['Sex'] = df['Sex'].apply(lambda x: 0 if x=='male' else 1)
df.head()
```

Out [1]:

	Survived	Pclass	Sex	Fare	Embarked
0	0	3	0	7.2500	0
1	1	1	1	71.2833	1
2	1	3	1	7.9250	0
3	1	1	1	53.1000	0
4	0	3	0	8.0500	0

```
In [2]: from xgboost import XGBClassifier
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.metrics import accuracy_score

x = df.drop('Survived', axis=1)
y = df['Survived']
x_train, x_test, y_train, y_test = train_test_split(x, y,
                                                    test_size=0.3,
                                                    random_state=0)

params = {
    'eta': [0.01, 0.1, 0.5, 1],
    'min_child_weight': [1, 2, 3, 5, 10],
    'max_depth': [2, 3, 5, 6, 10],
    'gamma': [0, 0.1, 0.2, 0.3, 0.5, 1, 2, 5, 10]
}
xgb = XGBClassifier()
grid = GridSearchCV(xgb, param_grid=params, cv=5, n_jobs=-1)

grid.fit(x_train, y_train)
```

```
Out[2]: GridSearchCV(cv=5,
                    estimator=XGBClassifier(base_score=None, booster=None,
                    ,
                    colsample_bylevel=None,
                    colsample_bynode=None,
                    colsample_bytree=None, gamma=
None,
                    gpu_id=None, importance_type=
'gain',
                    interaction_constraints=None,
                    learning_rate=None, max_delta
_step=None,
                    max_depth=None, min_child_wei
ght=None,
                    missing=nan, monotone_constra
ints=None,
                    n_estimators=100, n_jobs=None,
                    ,
                    num_parallel_tree=None, rando
m_state=None,
                    ne,
                    scale_pos_weight=None, subsam
ple=None,
                    tree_method=None, validate_pa
rameters=None,
                    verbosity=None),
                    n_jobs=-1,
                    param_grid={'eta': [0.01, 0.1, 0.5, 1],
                                'gamma': [0, 0.1, 0.2, 0.3, 0.5, 1, 2, 5,
10],
                                'max_depth': [2, 3, 5, 6, 10],
                                'min_child_weight': [1, 2, 3, 5, 10]})
```

```
In [3]: print('result : {}'.format(grid.cv_results_))
print()
print('best score : {:.5f}'.format(grid.best_score_))
print()
print('best parameters : {}'.format(grid.best_params_))
```

3, 5, 10,
1, 2, 3, 5, 10, 1, 2, 3, 5, 10, 1, 2, 3, 5, 1
0, 1, 2,
3, 5, 10, 1, 2, 3, 5, 10, 1, 2, 3, 5, 10, 1,
2, 3, 5,
10, 1, 2, 3, 5, 10, 1, 2, 3, 5, 10, 1, 2, 3,
5, 10, 1,
2, 3, 5, 10, 1, 2, 3, 5, 10, 1, 2, 3, 5, 10,
1, 2, 3,
5, 10, 1, 2, 3, 5, 10, 1, 2, 3, 5, 10, 1, 2,
3, 5, 10,
1, 2, 3, 5, 10, 1, 2, 3, 5, 10, 1, 2, 3, 5, 1
0, 1, 2,
3, 5, 10, 1, 2, 3, 5, 10, 1, 2, 3, 5, 10, 1,
2, 3, 5,
10, 1, 2, 3, 5, 10, 1, 2, 3, 5, 10, 1, 2, 3,
5, 10, 1,
2, 3, 5, 10, 1, 2, 3, 5, 10, 1, 2, 3, 5, 10,
1, 2, 3,
5 10 1 2 3 5 10 1 2 3 5 10 1 2

```
In [4]: xgb = XGBClassifier(eta=1, gamma=0.2, max_depth=3,
                           min_child_weight=10)
xgb.fit(x_train, y_train)

pred = xgb.predict(x_test)
acc = accuracy_score(pred, y_test)

print('accuracy score : {:.5f}'.format(acc))
```

accuracy score : 0.80970

```
In [ ]: # 解答例では、学習の際はランダムフォレストを上回ってましたが、テストデータでは
# 負けました。テストデータの結果は、'train_test_split'の'random_state'
# の値で変わることがありますが、それが同じ'0'の場合、ランダムフォレストだった
# ということになりますね。
```