

In []: **【ハイパーパラメタ・チューニング】**

問2

② C, gammaの値をそれぞれ数通り組合せ、予測精度の変化を確認してください。

C (デフォルト: **1.0**) \in 実数

gamma(デフォルト:1/(特微量の数*分散)) ∈ 実数

```
In [28]: # ②
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns

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[28]:

	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

[illegible]

```
In [30]: from sklearn.svm import SVC
from sklearn.metrics import accuracy_score
from sklearn.model_selection import train_test_split

best_score = 0
for C in [0.01, 0.1, 1, 10]:
    for gamma in [0.01, 0.1, 1, 10]:
        clf = SVC(C=C, gamma=gamma)
        clf.fit(x_train, y_train)
        pred = clf.predict(x_test)
        acc = accuracy_score(pred, y_test)
        if acc > best_score:
            best_score = acc
            best_params = {'C': C, 'gamma': gamma}

print('best score : {:.5f}'.format(best_score))
print('best parameters : {}'.format(best_params))

best score : 0.78731
best parameters : {'C': 10, 'gamma': 1}
```

```
In [31]: # 解説
# Cの取る値を 0.01,0.1,1(デフォルト),10、gammaの取る値を0.01,0.1,1,10
# (gammaのデフォルト値は 1/( 特徴量の数*分散) ) としました。このCとgammaの組
# 合せすべてを二重のforループで計算します(いわゆる、「しらみつぶし」というや
# つです)
# 計算させただけでは、最後に計算したもの(C=10,gamma=10)しかメモリに残らない
# ので、'accuracy_score()'の値が最も大きくなる組合せを if文で判定し、best
# _params={'C': 0, 'gamma': 0} というように辞書型で記録し、スコアが上回るた
# びに辞書を上書きしていきます。
```