

그리드 서치 두번째 시간

학습 내용

- SVC의 kernel 매개변수에 따른 주의 사항을 알아본다.

비대칭 매개변수 그리드 탐색

- SVC는 kernel 매개변수를 가지고 있다. 어떤 커널을 사용하는지에 따라 관련 있는 매개변수들이 결정.
 - kernel='rbf'이면 C와 gamma를 모두 사용. 이런 경우에 모든 조합을 조사하는 것은 맞지 않음.
 - kernel='linear'이면 선형 모델. C매개변수만 사용.
 - kernel='linear'이면 gamma를 사용하지 않으므로 gamma의 값에 대해 조사하는 것이 시간 낭비

In [5]:

```
from sklearn.model_selection import GridSearchCV
from sklearn.svm import SVC
import pandas as pd
```

In [6]:

```
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
iris = load_iris()
```

In [7]:

```
X_train, X_test, y_train, y_test = train_test_split(iris.data, iris.target,
                                                    random_state=0)
```

In [8]:

```
param_grid = [{ 'kernel': ['rbf'],
                  'C': [0.001, 0.01, 0.1, 1, 10, 100],
                  'gamma': [0.001, 0.01, 0.1, 1, 10, 100]},

               { 'kernel': ['linear'],
                  'C': [0.001, 0.01, 0.1, 1, 10, 100]}]

print("그리드 목록:\n{}".format(param_grid))
```

그리드 목록:

```
[{'kernel': ['rbf'], 'C': [0.001, 0.01, 0.1, 1, 10, 100], 'gamma': [0.001, 0.01, 0.1, 1, 10, 100]}, {'kernel': ['linear'], 'C': [0.001, 0.01, 0.1, 1, 10, 100]}]
```

In [9]:



```
grid_search = GridSearchCV(SVC(), param_grid, cv=5, return_train_score=True)
grid_search.fit(X_train, y_train)
print("최적 파라미터: {}".format(grid_search.best_params_))
print("최고 교차 검증 점수: {:.2f}".format(grid_search.best_score_))
```

최적 파라미터: {'C': 10, 'gamma': 0.1, 'kernel': 'rbf'}

최고 교차 검증 점수: 0.97

In [10]:



```
results = pd.DataFrame(grid_search.cv_results_)
# 좀 더 나은 출력을 위해 결과를 전치시킵니다
display(results.T)
```

	0	1	2	3	4	
mean_fit_time	0.00418963	0.0013916	0.00121255	0.00100226	0.00101514	0.00019969
std_fit_time	0.00158592	0.000483649	0.00041019	1.7953e-05	1.98375e-05	0.00039939
mean_score_time	0.00158443	0.000401497	0.000804567	0.000998831	0.000390482	
std_score_time	0.000802272	0.000491736	0.000403458	1.33685e-06	0.000478431	
param_C	0.001	0.001	0.001	0.001	0.001	0.001
param_gamma	0.001	0.01	0.1	1	10	100
param_kernel	rbf	rbf	rbf	rbf	rbf	rbf
params	{'C': 0.001, 'gamma': 0.001, 'kernel': 'rbf'}	{'C': 0.001, 'gamma': 0.01, 'kernel': 'rbf'}	{'C': 0.001, 'gamma': 0.1, 'kernel': 'rbf'}	{'C': 0.001, 'gamma': 1, 'kernel': 'rbf'}	{'C': 0.001, 'gamma': 10, 'kernel': 'rbf'}	{'C': 0.001, 'gamma': 100, 'kernel': 'rbf'}
split0_test_score	0.347826	0.347826	0.347826	0.347826	0.347826	0.347826
split1_test_score	0.347826	0.347826	0.347826	0.347826	0.347826	0.347826
split2_test_score	0.363636	0.363636	0.363636	0.363636	0.363636	0.363636
split3_test_score	0.363636	0.363636	0.363636	0.363636	0.363636	0.363636
split4_test_score	0.409091	0.409091	0.409091	0.409091	0.409091	0.409091
mean_test_score	0.366403	0.366403	0.366403	0.366403	0.366403	0.366403
std_test_score	0.0224845	0.0224845	0.0224845	0.0224845	0.0224845	0.0224845
rank_test_score	27	27	27	27	27	27
split0_train_score	0.370787	0.370787	0.370787	0.370787	0.370787	0.370787
split1_train_score	0.370787	0.370787	0.370787	0.370787	0.370787	0.370787
split2_train_score	0.366667	0.366667	0.366667	0.366667	0.366667	0.366667
split3_train_score	0.366667	0.366667	0.366667	0.366667	0.366667	0.366667
split4_train_score	0.355556	0.355556	0.355556	0.355556	0.355556	0.355556
mean_train_score	0.366092	0.366092	0.366092	0.366092	0.366092	0.366092
std_train_score	0.00558129	0.00558129	0.00558129	0.00558129	0.00558129	0.00558129

23 rows × 42 columns

- 0~42는 각각의 다른 파라미터 값을 가지고 결과를 확인

그리드 서치를 교차 검증 적용하여 결과를 확인

In [11]:



```
from sklearn.model_selection import cross_val_score
```

In [12]:



```
param_grid = { 'C'      : [0.001, 0.01, 0.1, 1, 10, 100],
                'gamma': [0.001, 0.01, 0.1, 1, 10, 100] }

scores = cross_val_score(GridSearchCV(SVC(), param_grid, cv=5),
                          iris.data, iris.target, cv=5)

print("교차 검증 점수: ", scores)
print("교차 검증 평균 점수: ", scores.mean())
print(param_grid)
```

```
교차 검증 점수: [0.96666667 1.          0.96666667 0.96666667 1.          ]
교차 검증 평균 점수: 0.9800000000000001
{'C': [0.001, 0.01, 0.1, 1, 10, 100], 'gamma': [0.001, 0.01, 0.1, 1, 10, 100]}
```