군집 DBSCAN (2)

#01. 패키지 참조

```
import sys
import seaborn as sb
from matplotlib import pyplot as plt
from pandas import read_excel
from sklearn.cluster import DBSCAN
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import silhouette_score
```

```
plt.rcParams["font.family"] = 'AppleGothic' if sys.platform == 'darwin' else 'Malgun G
plt.rcParams["font.size"] = 14
plt.rcParams['axes.unicode_minus'] = False
```

#02. 데이터 가져오기

```
origin = read_excel("https://data.hossam.kr/G02/customer.xlsx", index_col="고객ID")
print(origin.info())
origin.head()
```

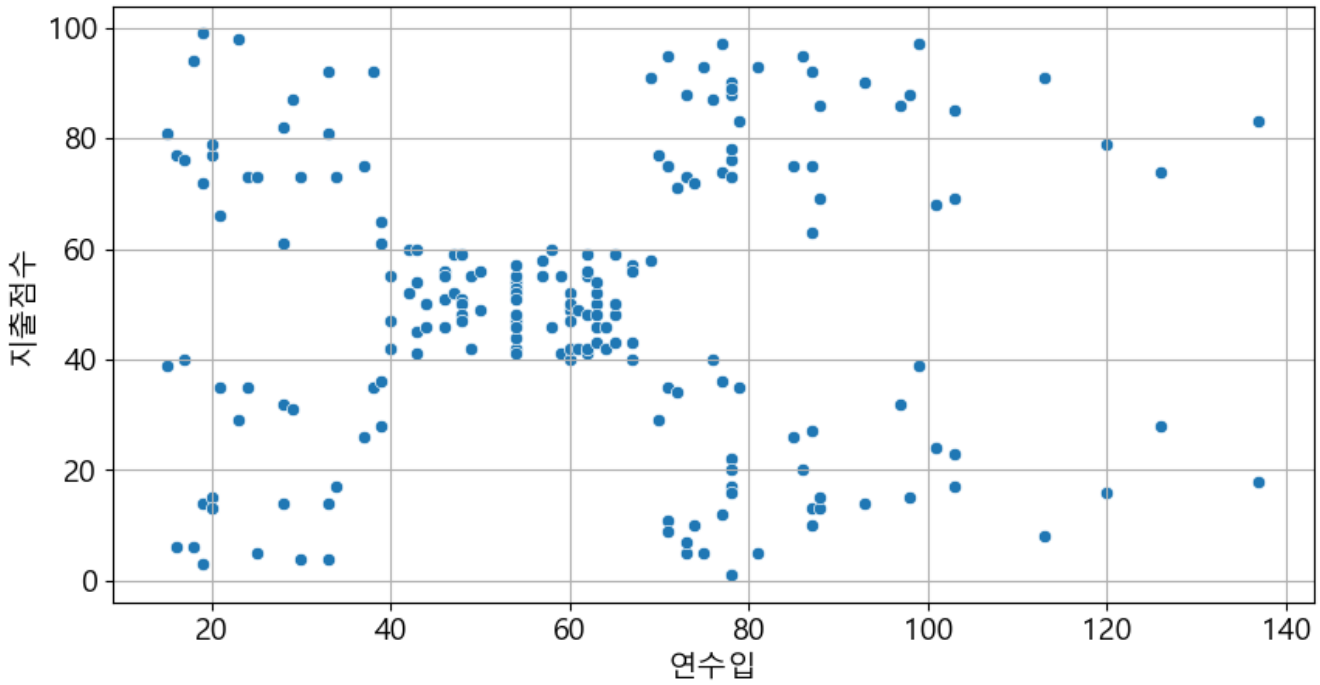
```
<class 'pandas.core.frame.DataFrame'>
Index: 200 entries, 1 to 200
Data columns (total 4 columns):
 #   Column  Non-Null Count  Dtype 
---  ------  --------------  ----- 
 0   성별      200 non-null    object
 1   나이      200 non-null    int64 
 2   연수입     200 non-null    int64 
 3   지출점수    200 non-null    int64 
dtypes: int64(3), object(1)
memory usage: 7.8+ KB
None
```

	성별	나이	연수입	지출점수
고객ID				
1	Male	19	15	39
2	Male	21	15	81
3	Female	20	16	6
4	Female	23	16	77

	성별	나이	연수입	지출점수
고객ID				
5	Female	31	17	40

연수입에 따른 지출점수 확인

```
plt.figure(figsize=(10, 5))
sb.scatterplot(data=origin, x='연수입', y='지출점수')
plt.grid()
plt.show()
plt.close()
```



#03. 데이터 전처리

필요한 필드 추출

```
x = origin.filter(['연수입', '지출점수'])
x.head()
```

	연수입	지출점수
고객ID		
1	15	39
2	15	81
3	16	6
4	16	77
5	17	40

데이터 표준화

```
scaler = StandardScaler()
scaler.fit(x)
n_data = scaler.transform(x)
n_data[:5]
```

```
array([[-1.73899919, -0.43480148],
       [-1.73899919,  1.19570407],
       [-1.70082976, -1.71591298],
       [-1.70082976,  1.04041783],
       [-1.66266033, -0.39597992]])
```

#04. DBSCAN 구현

모델 구축

```
dbscan = DBSCAN(eps=0.3, min_samples=5)
dbscan.fit(n_data)
```

DBSCAN

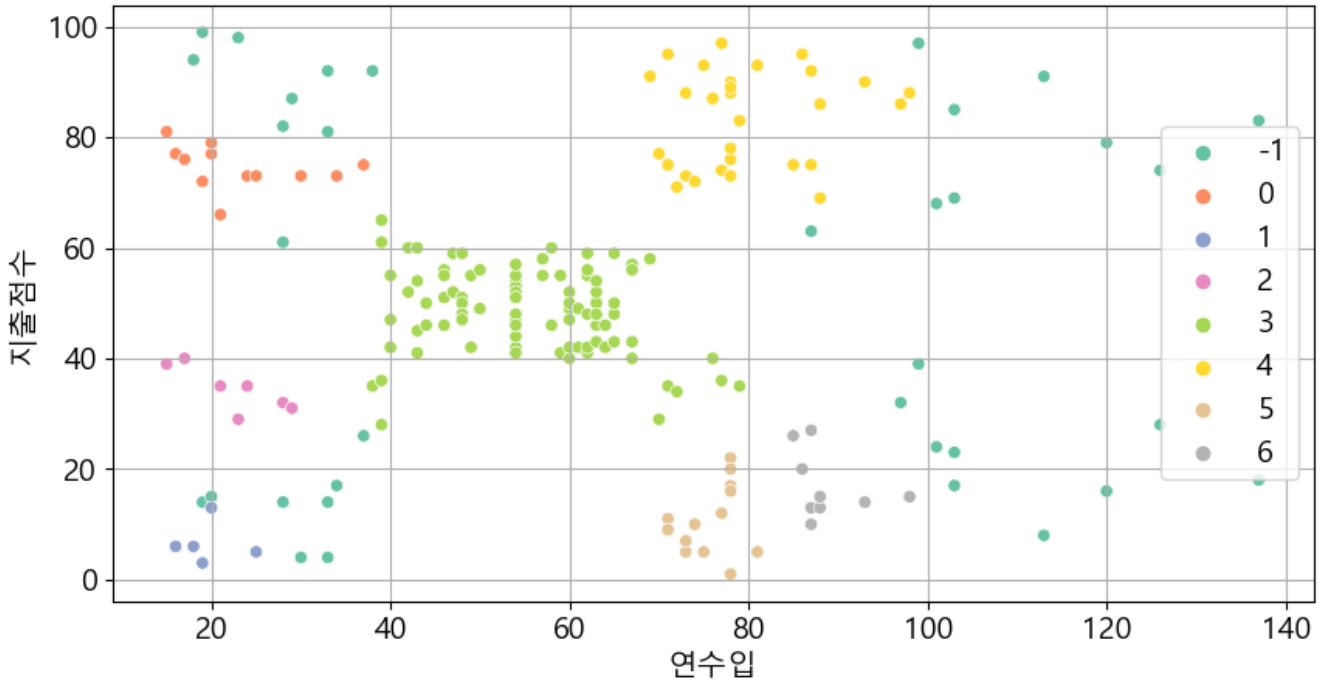
```
DBSCAN(eps=0.3)
```

군집 결과

```
cluster_label = dbscan.labels_
cluster_label
```

```
array([ 2,  0,  1,  0,  2,  0,  1, -1,  1,  0, -1, -1, -1,  0,  1,  0,  2,
        0,  2, -1,  2,  0,  1,  0, -1, -1,  2, -1,  2, -1, -1,  0, -1, -1,
       -1, -1, -1,  0, -1,  0,  3, -1,  3,  3,  3,  3,  3,  3,  3,  3,  3,
        3,  3,  3,  3,  3,  3,  3,  3,  3,  3,  3,  3,  3,  3,  3,  3,  3,
        3,  3,  3,  3,  3,  3,  3,  3,  3,  3,  3,  3,  3,  3,  3,  3,  3,
        3,  3,  3,  3,  3,  3,  3,  3,  3,  3,  3,  3,  3,  3,  3,  3,  3,
        3,  3,  3,  3,  4,  3,  4,  3,  4,  5,  4,  5,  4,  3,  4,  5,  4,
        5,  4,  5,  4,  5,  4,  3,  4,  5,  4,  3,  4,  5,  4,  5,  4,  5,
        4,  5,  4,  5,  4,  5,  4,  3,  4,  5,  4,  6,  4,  6,  4,  6, -1,
        6,  4,  6,  4,  6,  4,  6,  4,  6,  4, -1,  4,  6,  4, -1, -1, -1,
       -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1], dtype=int64)
```

```
plt.figure(figsize=(10, 5))
sb.scatterplot(data=origin, x='연수입', y='지출점수', hue=cluster_label, palette='Set2')
plt.grid()
plt.show()
plt.close()
```



#05. 핵심 포인트 확인

핵심포인트의 인덱스

```
core_sample_indices = dbscan.core_sample_indices_
core_sample_indices[:5]
```

```
array([1, 2, 3, 5, 6], dtype=int64)
```

해당 인덱스의 실 데이터

```
components = dbscan.components_
components[:5]
```

```
array([[-1.73899919,  1.19570407],
       [-1.70082976, -1.71591298],
       [-1.70082976,  1.04041783],
       [-1.66266033,  1.00159627],
       [-1.62449091, -1.71591298]])
```

학습 데이터 중에서 핵심 포인트의 인덱스와 일치하는 데이터 찾기

```
is_core_samples = []

for i in range(0, n_data.shape[0]):
    if i in core_sample_indices:
        is_core_samples.append(1)
    else:
        is_core_samples.append(0)
```

```
print(is_core_samples)
```

```
[0, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1
```

시각화

```
plt.figure(figsize=(10, 5))
sb.scatterplot(x=n_data[:,0], y=n_data[:,1],
               hue=is_core_samples, palette='Set2')
plt.grid()
plt.show()
plt.close()
```