

군집-DBSCAN(4) - 교통사고 밀집지역

#01. 패키지 참조

```
import sys
import seaborn as sb
import numpy as np
from itertools import product
from matplotlib import pyplot as plt
from pandas import read_excel
from sklearn.neighbors import NearestNeighbors
from sklearn.cluster import DBSCAN
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import silhouette_score
```

#02. 데이터 가져오기

```
origin = read_excel('https://data.hossam.kr/G02/traffic.xlsx')
print(origin.info())
origin.head()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2658 entries, 0 to 2657
Data columns (total 23 columns):
#   Column                Non-Null Count  Dtype
---  -
0   발생년                2658 non-null  int64
1   발생년월일시          2658 non-null  datetime64[ns]
2   주야                  2658 non-null  object
3   요일                  2658 non-null  object
4   사망자수              2658 non-null  int64
5   부상자수              2658 non-null  int64
6   중상자수              2658 non-null  int64
7   경상자수              2658 non-null  int64
8   부상신고자수          2658 non-null  int64
9   발생지시도            2658 non-null  object
10  발생지시군구           2658 non-null  object
11  사고유형_대분류        2658 non-null  object
12  사고유형_중분류        2658 non-null  object
13  사고유형               2658 non-null  object
14  가해자법규위반         2658 non-null  object
15  도로형태_대분류        2658 non-null  object
16  도로형태               2658 non-null  object
17  가해자_당사자종별      2658 non-null  object
18  피해자_당사자종별      2658 non-null  object
19  발생위치X_UTMK         2658 non-null  int64
20  발생위치Y_UTMK         2658 non-null  int64
21  경도                   2658 non-null  float64
```

22 위도 2658 non-null float64
 dtypes: datetime64[ns](1), float64(2), int64(8), object(12)
 memory usage: 477.7+ KB
 None

	발생 년	발생년 월일시	주 야	요 일	사 망 자 수	부 상 자 수	중 상 자 수	경 상 자 수	부 상 신 고 자 수	발 생 지 시 도	...	사 고 유 형	가 해 자 법 규 위 반	도 로 형 태 - 대 분 류	도 로 형 태	가 해 자 - 당 사 자 종 별	피 해 자 - 당 사 자 종 별
0	2022	2022-01-01 10:00:00	주	토	1	0	0	0	0	전남	...	공작물 충돌	안전운전의무불이행	단일로	교량 위	승용차	없음
1	2022	2022-01-01 14:00:00	주	토	1	0	0	0	0	전북	...	추돌	안전운전의무불이행	단일로	기타 단일로	원동기장치자전거	승용차
2	2022	2022-01-01 17:00:00	주	토	1	0	0	0	0	경기	...	도로 이탈 추락	안전운전의무불이행	단일로	기타 단일로	이륜차	없음
3	2022	2022-01-01 20:00:00	야	토	1	0	0	0	0	부산	...	기타	안전운전의무불이행	단일로	고가도로 위	승용차	승용차
4	2022	2022-01-02	주	일	1	0	0	0	0	서울	...	횡단	안전	단일	기타	승합	보행

	발생 년	발생년 월일시	주 야	요 일	사 망 자 수	부 상 자 수	중 상 자 수	경 상 자 수	부 상 신 고 자 수	발 생 지 시 도	...	사 고 유 형	가 해 자 법 규 위 반	도 로 형 태 - 대 분 류	도 로 형 태	가 해 자 - 당 사 자 종 별	피 해 자 - 당 사 자 종 별
		06:00:00										중	운 전 의 무 불 이 행	로	단 일 로	차	자

5 rows × 23 columns

#03. 데이터 전처리

필요한 데이터만 추출

```
df = origin.query("발생지시도=='서울' and 사망자수 > 0")
print(df.shape)
df.head()
```

(215, 23)

	발생 년	발생년 월일시	주 야	요 일	사 망 자 수	부 상 자 수	중 상 자 수	경 상 자 수	부 상 신 고 자 수	발 생 지 시 도	...	사 고 유 형	가 해 자 법 규 위 반	도 로 형 태 - 대 분 류	도 로 형 태	가 해 자 - 당 사 자 종 별	피 해 자 - 당 사 자 종 별
4	2022	2022-01-02 06:00:00	주	일	1	0	0	0	0	서울	...	횡단 중	안전운 전 의 무 불 이 행	단 일 로	기 타 단 일 로	승 합 차	보 행 자
11	2022	2022-01-03	야	월	1	0	0	0	0	서울	...	횡단	안전	단 일	기 타	승 용	보 행

	발생 년	발생년 월일시	주 야	요 일	사 망 자 수	부 상 자 수	증 상 자 수	경 상 자 수	부 상 신 고 자 수	발 생 지 시 도	...	사 고 유 형	가 해 자 법 규 위 반	도 로 형 태 - 대 분 류	도 로 형 태	가 해 자 - 당 사 자 종 별	피 해 자 - 당 사 자 종 별
		18:00:00										중	운전의무불이행	로	단일로	차	자
20	2022	2022-01-05 04:00:00	야	수	1	0	0	0	0	서울	...	차도통행중	안전운전의무불이행	교차로	교차로내	승용차	보행자
22	2022	2022-01-05 06:00:00	주	수	1	0	0	0	0	서울	...	횡단중	보행자보호의무위반	단일로	기타단일로	건설기계	보행자
31	2022	2022-01-06 21:00:00	야	목	1	0	0	0	0	서울	...	횡단중	신호위반	단일로	기타단일로	이륜차	보행자

5 rows × 23 columns

군집에 사용할 데이터를 표준화

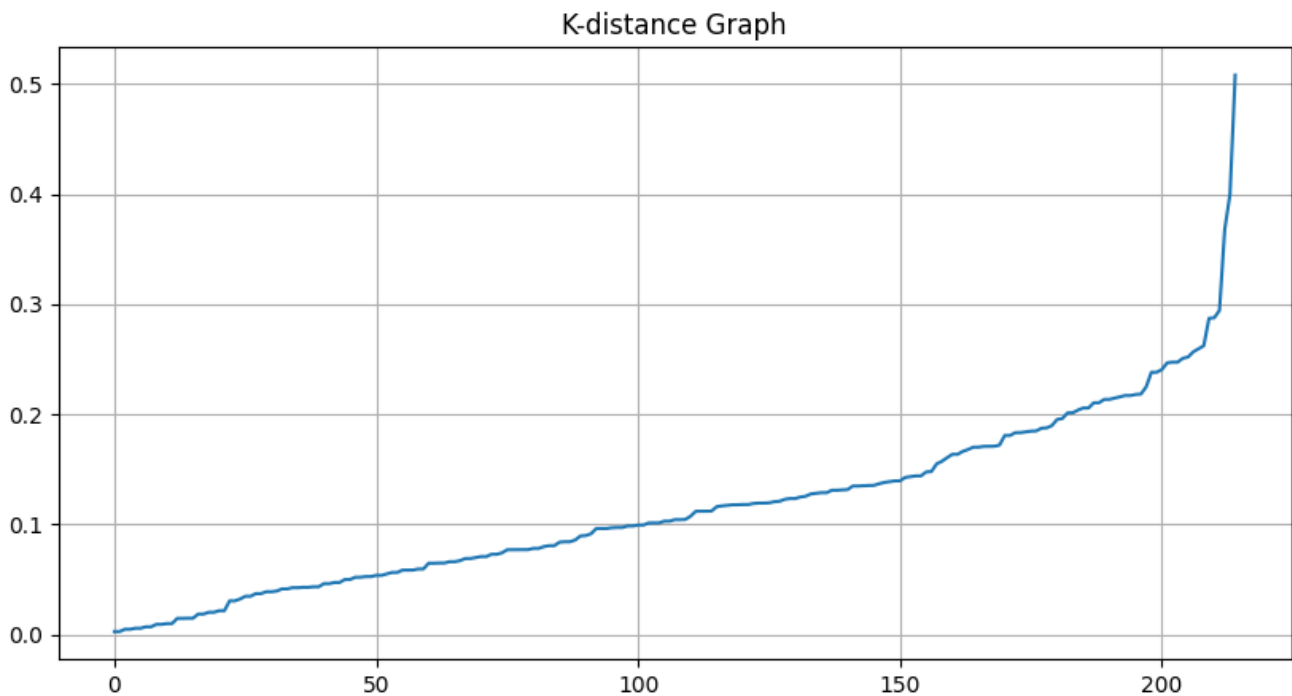
```
x = df[['경도', '위도']]
scaler = StandardScaler()
n_data = scaler.fit_transform(x)
n_data[:5]
```

```
array([[ -0.49295687,  0.81624643],
       [-0.88651335, -0.68436506],
       [-0.0892039 , -1.17366995],
       [-0.00405861, -1.32094612],
       [ 0.37764499,  0.11784443]])
```

#04. sorted-k-dist 그래프

```
neighbors = NearestNeighbors(n_neighbors=4)
neighbors_fit = neighbors.fit(n_data)
dist, indi = neighbors_fit.kneighbors(n_data)
s_dist = np.sort(dist, axis=0)
```

```
plt.figure(figsize=(10, 5))
plt.plot(s_dist[:, 1])
plt.grid()
plt.title('K-distance Graph')
plt.show()
plt.close()
```



#05. DBSCAN 모델 구축

1. 조건값 설정

```
eps = [0.2, 0.3]
minPts = [4, 5]
```

```
for e, m in list(product(*[eps, minPts])):
    dbscan = DBSCAN(eps=e, min_samples=m, n_jobs=-1)
```

```

clusters = dbscan.fit_predict(n_data)

print('eps=', e, 'min_samples=', m)
print('cluster label:', np.unique(clusters))
print('silhouette score:', silhouette_score(n_data, clusters))

plt.figure(figsize=(10, 5))
plt.scatter(n_data[:, 0], n_data[:, 1], c=clusters, cmap='Paired')
plt.grid()
plt.title('eps=%s, min_samples=%s → %s' % (str(e), str(m), clusters) )
plt.show()
plt.close()
print()

```

```

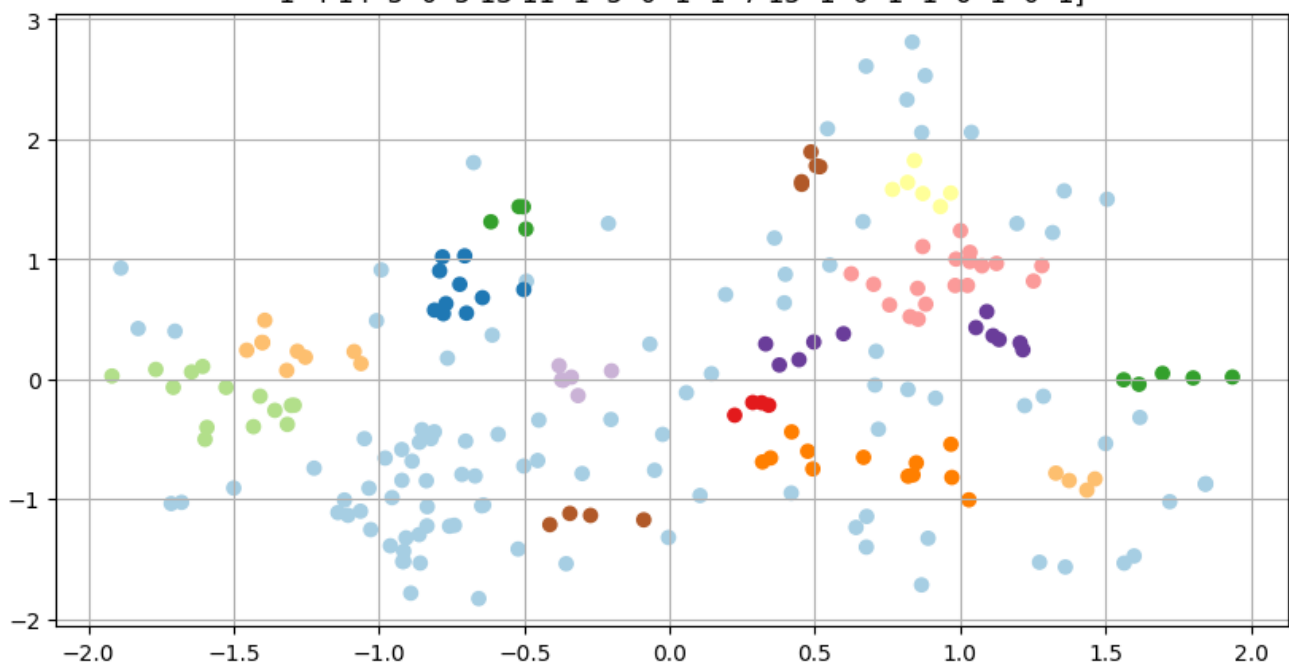
eps= 0.2 min_samples= 4
cluster label: [-1  0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15]
silhouette score: 0.0745649278380414

```

```

eps=0.2, min_samples=4 -> [-1 0 15 -1 11 1 7 2 0 2 0 2 -1 3 4 0 5 2 -1 -1 6 9 0 -1
    7 8 -1 3 1 0 0 -1 1 -1 9 3 -1 0 9 0 1 -1 9 9 -1 -1 6 5
   -1 9 -1 -1 2 -1 0 10 9 7 7 -1 -1 2 13 0 -1 0 10 2 0 -1 -1 1
    8 9 2 0 -1 7 0 5 0 2 -1 -1 2 -1 -1 -1 -1 -1 15 -1 -1 4 -1
    0 1 8 11 5 0 -1 7 7 9 7 -1 0 10 15 -1 12 5 -1 0 2 1 -1 -1
    0 5 9 8 -1 5 0 -1 -1 -1 -1 -1 0 0 10 -1 -1 5 0 -1 13 5 -1 10
   -1 4 1 0 13 12 5 13 2 2 10 -1 14 6 12 12 9 11 0 14 -1 1 0 1
   -1 5 -1 0 3 2 14 13 -1 12 5 12 3 5 -1 5 14 11 5 0 9 -1 -1 0
   -1 4 14 5 0 5 13 11 -1 5 0 -1 -1 7 15 -1 0 -1 -1 6 -1 0 -1]

```



```

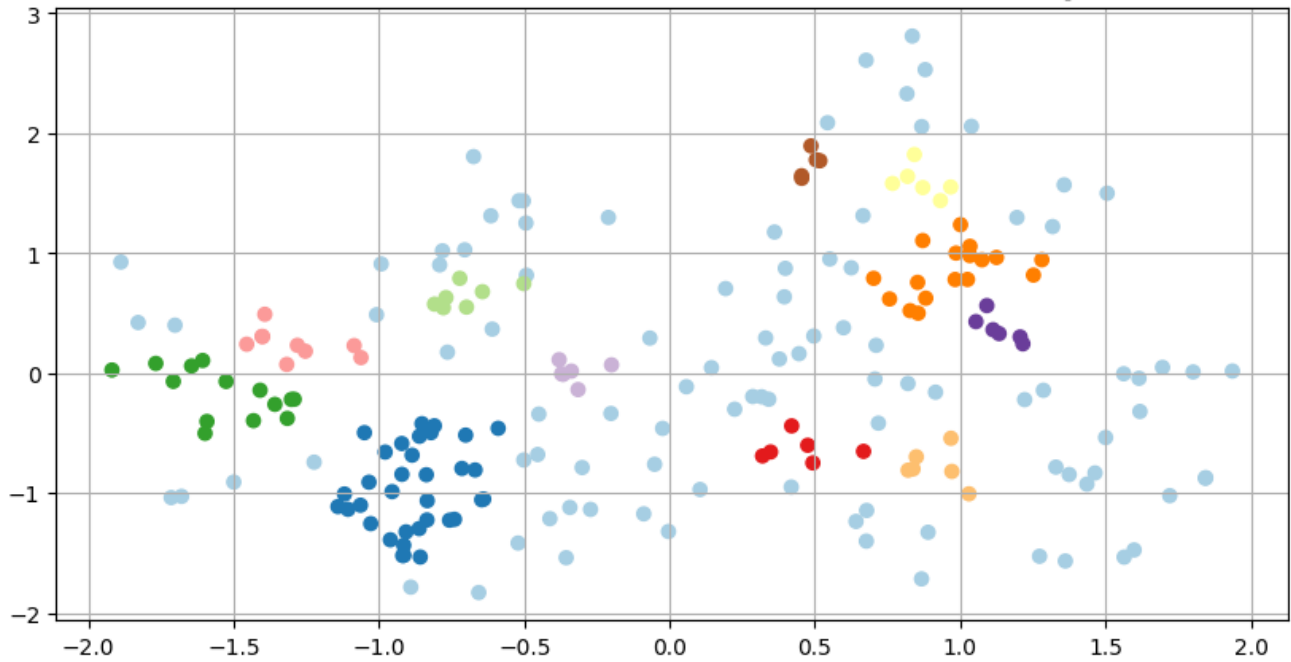
eps= 0.2 min_samples= 5
cluster label: [-1  0  1  2  3  4  5  6  7  8  9 10]
silhouette score: -0.030130955148575292

```

```

eps=0.2, min_samples=5 -> [-1 0 -1 -1 -1 1 3 2 0 2 0 2 -1 -1 -1 0 6 2 -1 -1 -1 5 0 -1
 3 -1 -1 -1 1 0 -1 -1 -1 -1 4 -1 -1 0 5 0 -1 -1 5 4 -1 -1 -1 6
-1 4 -1 -1 2 -1 0 7 4 3 3 -1 -1 2 9 0 -1 0 7 2 0 -1 -1 1
-1 5 2 0 -1 3 0 6 0 2 -1 -1 2 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
0 -1 -1 -1 6 0 -1 3 3 4 3 -1 0 7 -1 -1 8 6 -1 0 2 1 -1 -1
0 -1 4 -1 -1 6 0 -1 -1 -1 -1 -1 0 0 7 -1 -1 6 0 -1 9 6 -1 7
-1 -1 1 0 9 8 6 9 2 2 7 -1 10 -1 8 8 5 -1 0 10 -1 1 0 1
-1 6 -1 0 -1 2 10 9 -1 8 6 8 -1 6 -1 6 10 -1 6 0 5 -1 -1 0
-1 -1 10 6 0 6 9 -1 -1 6 0 -1 -1 3 -1 -1 0 -1 -1 -1 -1 0 -1]

```



```

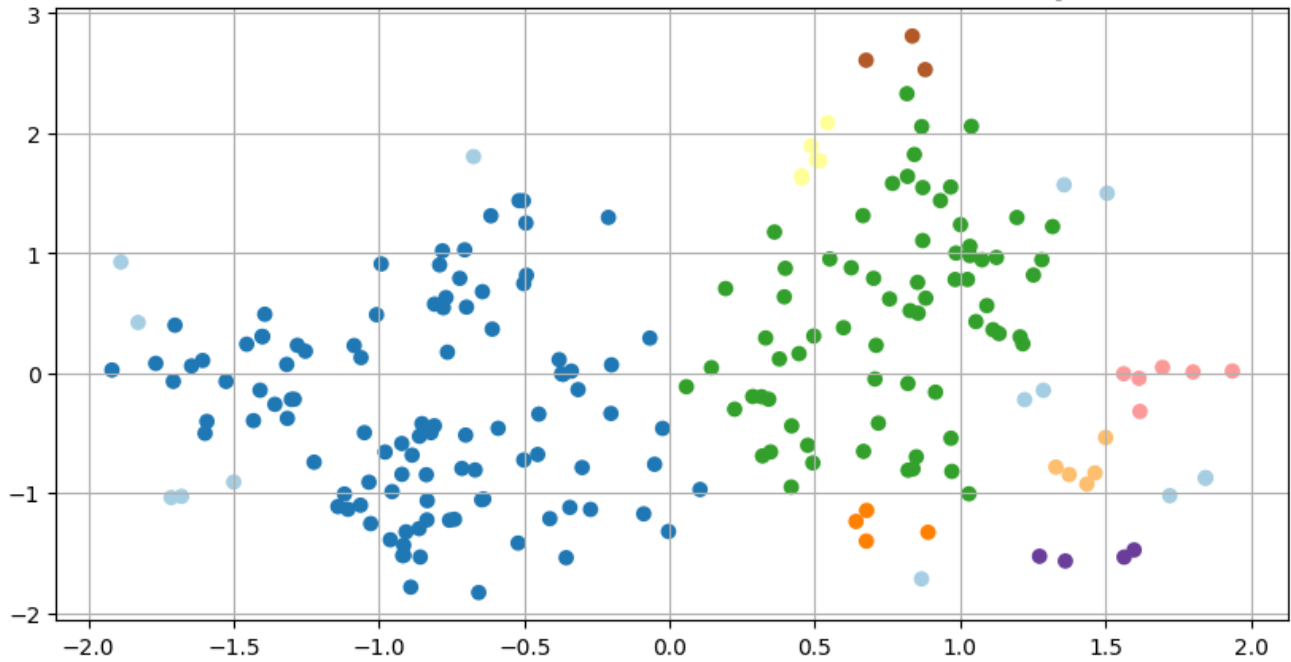
eps= 0.3 min_samples= 4
cluster label: [-1 0 1 2 3 4 5 6 7]
silhouette score: 0.13047665687331503

```

```

eps=0.3, min_samples=4 -> [ 0 0 0 0 1 0 0 0 0 0 0 0 0 -1 2 0 0 1 0 0 0 1 1 0 5
    0 3 -1 2 0 0 0 0 0 1 1 2 1 0 1 0 0 1 1 1 1 1 1 1 1
    6 1 -1 -1 0 1 0 0 1 0 0 7 1 0 1 0 1 0 0 0 0 0 0 4 0
    3 1 0 0 0 0 0 1 0 0 -1 5 0 7 -1 1 1 0 1 0 0 1 0 0
    0 0 3 1 1 0 -1 0 0 1 0 1 0 0 0 0 0 1 1 4 0 0 0 1 4
    0 1 1 3 0 1 0 -1 -1 0 1 0 0 0 0 0 0 1 1 0 2 1 1 1 0
    -1 0 0 0 1 1 1 1 0 0 0 0 0 6 1 1 1 1 1 0 6 0 0 0 0
    0 1 3 0 2 0 6 1 -1 1 1 1 2 1 -1 1 6 1 1 0 1 7 5 0
    1 0 6 1 0 1 1 1 0 1 0 5 0 0 0 -1 0 -1 1 1 4 0 0]

```



```

eps= 0.3 min_samples= 5
cluster label: [-1 0 1 2 3 4 5 6]
silhouette score: -0.11868242696621044

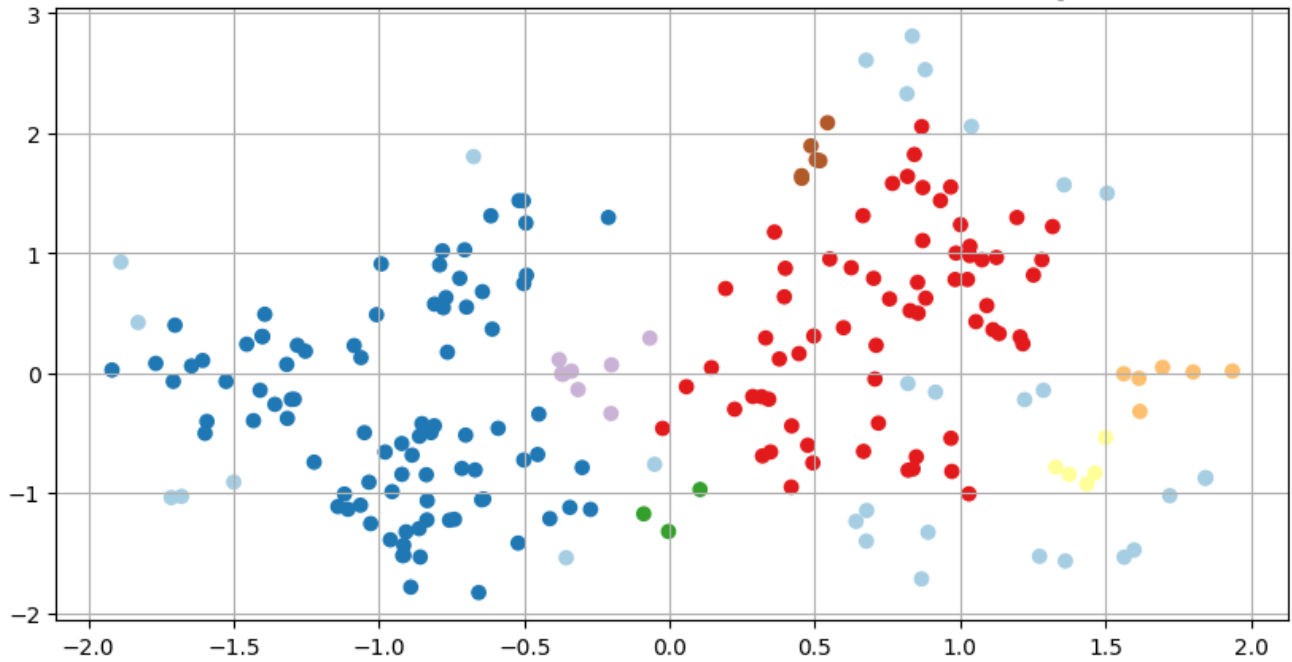
```



```

eps=0.3, min_samples=5 -> [ 0 0 1 1 2 0 0 0 0 0 0 0 -1 3 0 0 2 0 0 0 2 2 0 -1
 0 5 -1 3 0 0 0 4 0 2 2 3 2 0 2 0 0 -1 2 2 2 -1 2 2
 6 2 -1 -1 0 2 0 4 2 0 0 -1 2 0 2 0 -1 0 4 0 0 0 -1 0
 5 2 0 0 0 0 0 2 0 0 -1 -1 0 -1 -1 -1 2 0 2 0 0 2 0 2
 0 0 5 2 2 0 -1 0 0 2 0 2 0 4 0 0 2 2 -1 0 0 0 2 -1
 0 2 2 5 4 2 0 -1 -1 0 2 0 0 0 4 -1 2 2 0 3 2 2 2 4
 -1 0 0 0 2 2 2 2 0 0 4 0 6 2 2 2 2 2 0 6 1 0 0 0
 0 2 5 0 3 0 6 2 -1 2 2 2 3 2 -1 2 6 2 2 0 2 -1 -1 0
 2 0 6 2 0 2 2 2 0 2 0 -1 0 0 0 -1 0 -1 2 2 -1 0 -1]

```



```

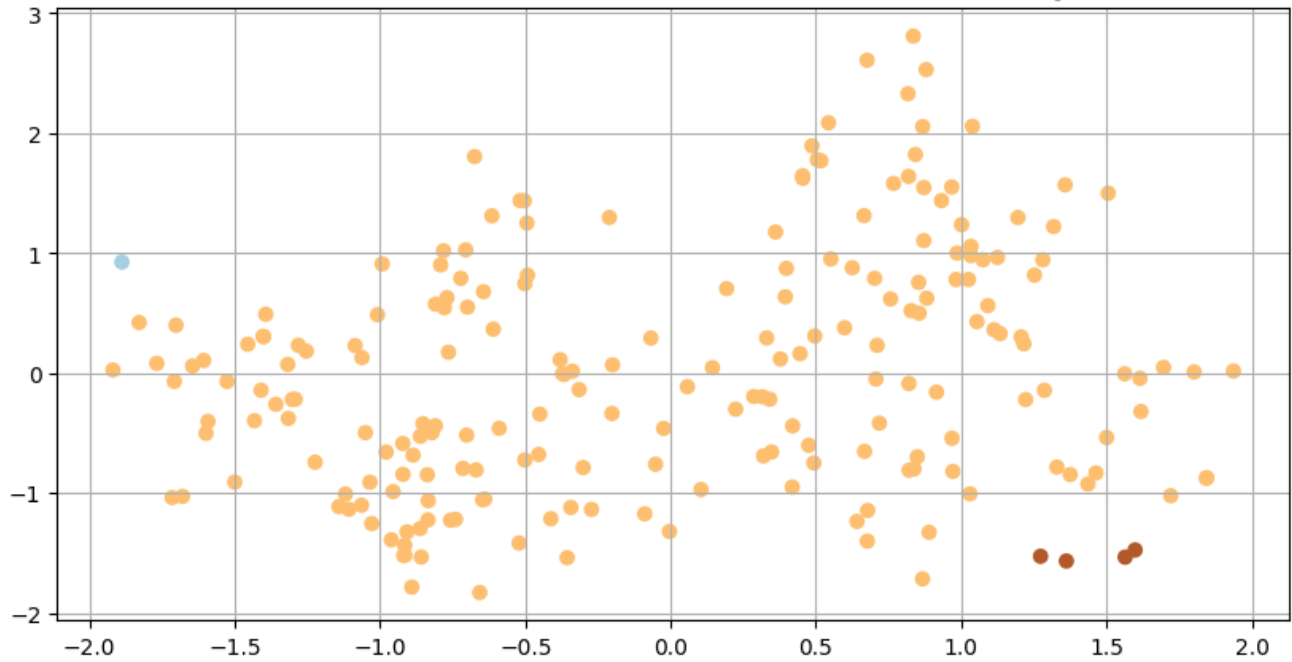
eps= 0.4 min_samples= 4
cluster label: [-1  0  1]
silhouette score: -0.001510096088665934

```

```

eps=0.4, min_samples=4 -> [ 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1
    0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
    0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
    0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0
    0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
    0 0 0 0 0 0 0 -1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
    0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
    0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0
    0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0]

```



```

eps= 0.4 min_samples= 5
cluster label: [-1 0]
silhouette score: 0.27951834880841014

```

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_indis:
        is_core_samples.append(1)
    else:
        is_core_samples.append(0)

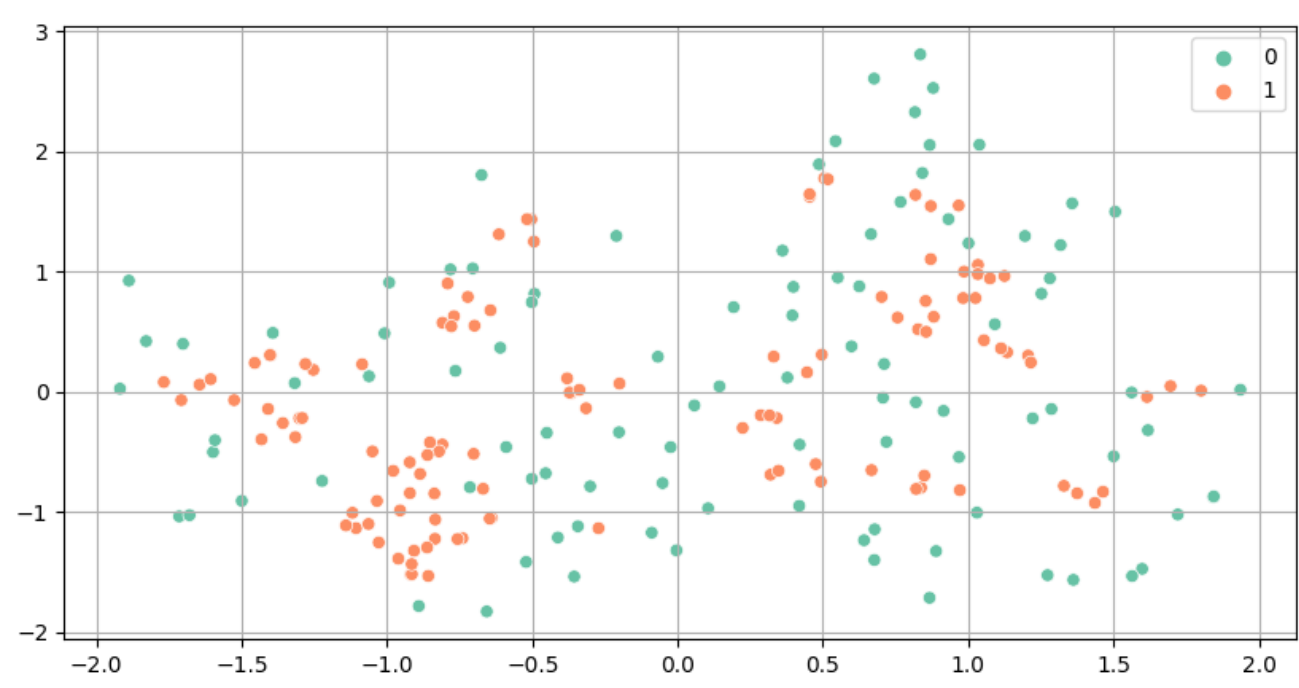
print(is_core_samples)
```

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

결과 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()
```



#06. 분석결과

```
result_df = df.filter(['발생지시도', '발생지시군구', '사고유형', '가해자법규위반',
                        '도로형태_대분류', '가해자_당사자종별', '피해자_당사자종별',
                        '경도', '위도'])

result_df.head()
```

	발생 지시 도	발생지 시군구	사고 유형	가해자법 규위반	도로형 태_대분 류	가해자_ 당사자종 별	피해자_ 당사자종 별	경도	위도
4	서울	서대문 구	횡단 중	안전운전 의무 불이 행	단일로	승합차	보행자	126.936182	37.579800
11	서울	영등포 구	횡단 중	안전운전 의무 불이 행	단일로	승용차	보행자	126.900927	37.507591
20	서울	동작구	차도 통행 중	안전운전 의무 불이 행	교차로	승용차	보행자	126.972351	37.484045
22	서울	동작구	횡단 중	보행자 보 호의무 위 반	단일로	건설기계	보행자	126.979979	37.476959
31	서울	성동구	횡단 중	신호위반	단일로	이륜차	보행자	127.014172	37.546193

```

result_df['cluster'] = clusters
result_df['core'] = is_core_samples
result_df.head()

```

	발 생 지 시 도	발생 지시 군구	사 고 유 형	가해 자법 규위 반	도로 형태_ 대분 류	가해 자_당 사자 종별	피해 자_당 사자 종별	경도	위도	cluster	core
4	서울	서대문구	횡단중	안전운전 의무 불이 행	단일로	승합차	보행자	126.936182	37.579800	-1	0
11	서울	영등포구	횡단중	안전운전 의무 불이 행	단일로	승용차	보행자	126.900927	37.507591	0	1
20	서울	동작구	차도 통행중	안전운전 의무 불이 행	교차로	승용차	보행자	126.972351	37.484045	15	0
22	서울	동작구	횡단중	보행자 보 호의무 위 반	단일로	건설기계	보행자	126.979979	37.476959	-1	0

	발 생 지 시 도	발 생 지 시 군 구	사 고 유 형	가 해 자 법 규 위 반	도 로 형 태 _ 대 분 류	가 해 자_ 당 사 자 종 별	피 해 자_ 당 사 자 종 별	경도	위도	cluster	core
31	서울	성동구	횡단중	신호위반	단일로	이륜차	보행자	127.014172	37.546193	11	0

result_df[result_df['core'] == 1]

	발 생 지 시 도	발 생 지 시 군 구	사 고 유 형	가 해 자 법 규 위 반	도 로 형 태 _ 대 분 류	가 해 자_ 당 사 자 종 별	피 해 자_ 당 사 자 종 별	경도	위도	cluster	core
11	서울	영등포구	횡단중	안전운전의무불이행	단일로	승용차	보행자	126.900927	37.507591	0	1
46	서울	서대문구	추돌	안전운전의무불이행	단일로	승용차	승용차	126.911360	37.570810	1	1
77	서울	영등포구	정면충돌	중앙선침범	단일로	화물차	이륜차	126.897744	37.512262	0	1
100	서울	영등포구	정면충돌	신호위반	단일로	승용차	승용차	126.858617	37.527982	2	1
127	서울	구로구	측면충돌	안전거리미확보	단일로	원동기장치자전거	승용차	126.894754	37.492957	0	1
...
2547	서울	금천구	기타	안전운전의무불이행	단일로	승합차	보행자	126.894220	37.473630	0	1
2595	서울	관악구	기타	안전운전의무	교차로	화물차	보행자	126.955938	37.485828	15	1

	발 생 지 시 도	발 생 지 시 군 구	사 고 유 형	가 해 자 법 규 위 반	도 로 형 태 _ 대 분 류	가 해 자_ 당 사 자 종 별	피 해 자_ 당 사 자 종 별	경 도	위 도	cluster	core
				불이 행							
2609	서울	영등포구	추돌	안전 운전 의무 불이 행	단일 로	이륜 차	건설 기계	126.903197	37.515174	0	1
2636	서울	용산구	횡단 중	안전 운전 의무 불이 행	기타	화물 차	보행 자	127.000383	37.526035	6	1
2649	서울	영등포구	공작 물 충돌	안전 운전 의무 불이 행	단일 로	승용 차	없음	126.886223	37.516619	0	1

116 rows × 11 columns