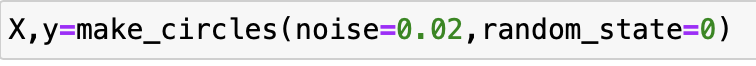
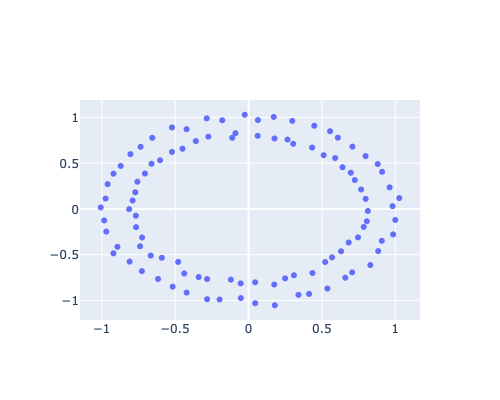
2019150445 신백록

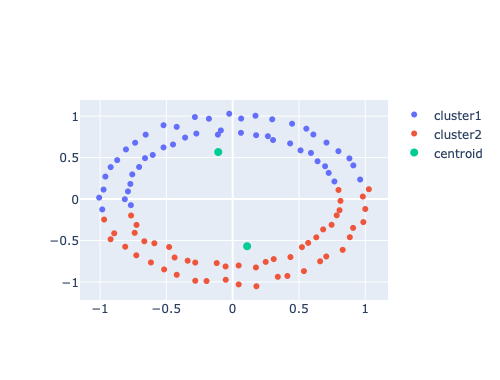
1. Sklearn에서 제공하는 make\_circles data에 적절한 noise를 첨가한 후, K-means\*\*, DBSCAN, HDBSCAN을 적용하여 비교하라.



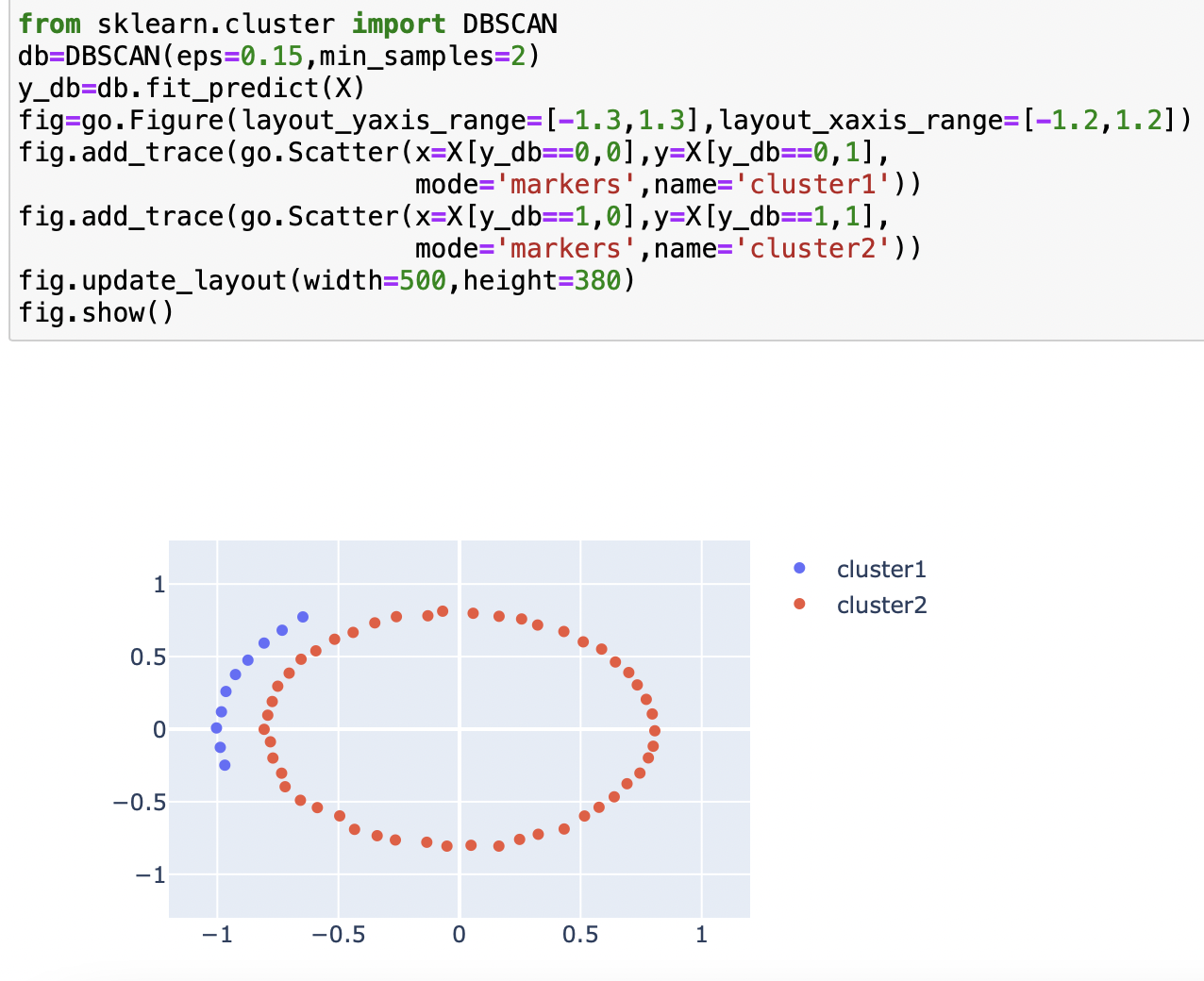
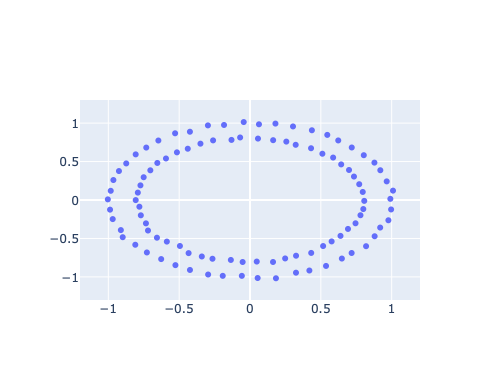
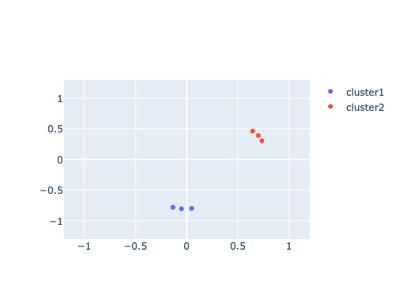
Noise를 0.02를 주고 Plot을 해보았다. Cluster 두 개가 큰 원, 작은 원을 이루고 있다는 것을 plot으로 확인할 수 있다.



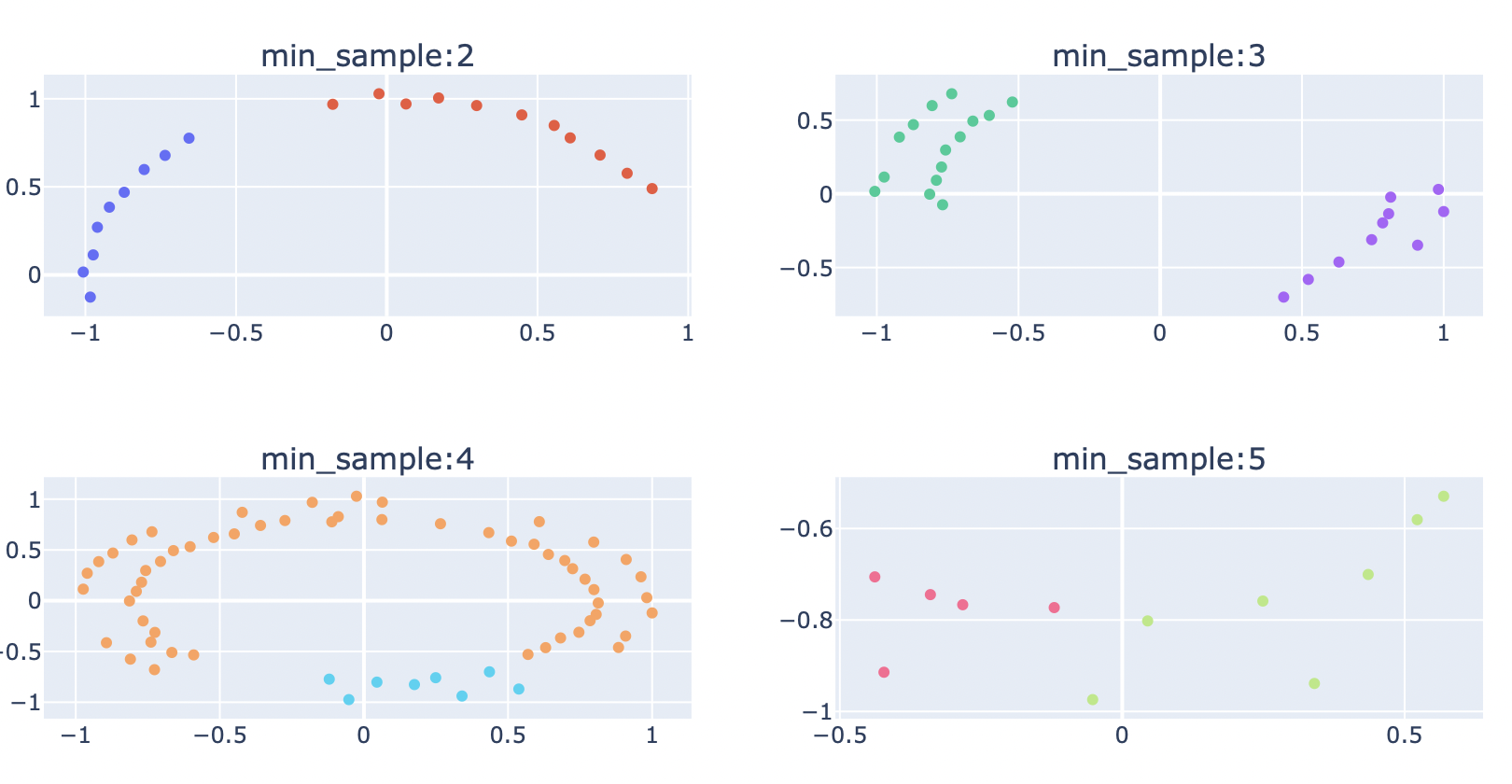
거리를 유사성 측도로 설정해 중심과의 거리를 계산하는 k-means cluster는 당연히 이 cluster 분류와 맞지 않는다. 안쪽 원과 바깥쪽 원 사이의 거리가 원의 반대쪽 점들과의 거리보다 가깝기 때문이다. 따라서 원 바깥쪽과 원 안쪽이 분류되는 것이 아니라 원을 반 나눠서 cluster가 분류된다.



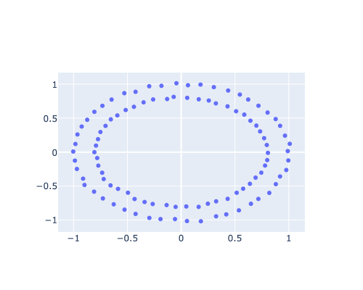
다음으로 DBSCAN을 통해서 clustering을 해보았는데, 적절한 eps와 min\_samples를 설정하기가 쉬운 일이 아니다. 여기서 min\_samples는 plot을 보니 점들이 많지 않고, dense가 그리 크지 않기에 2나 3이 적당한 것 같았고, 이거보다 크게 잡으면 core point가 줄어들어 남은 점들이 별로 없어진다. Eps는 원 안과 바깥 점들 사이의 거리가 0.1~0.2정도 차이 나기에 그 주변에서 여러 번 설정하며 보았다. 하지만 최적의 eps를 찾지 못했고, 그나마 eps=0.15 정도일 때가 cluster2가 살아남았다. Cluster1의 대부분 점들이 Noise로 분류되긴 했지만 말이다. 여기서 Eps를 키우면 두 번째 그림과 같이 모든 점들이 하나의 cluster로 분류가 될 것이다. Eps를 줄이면 세 번째 그림과 같이 대부분의 점들이 noise로 분류되고, 많은 점들이 없어진다.

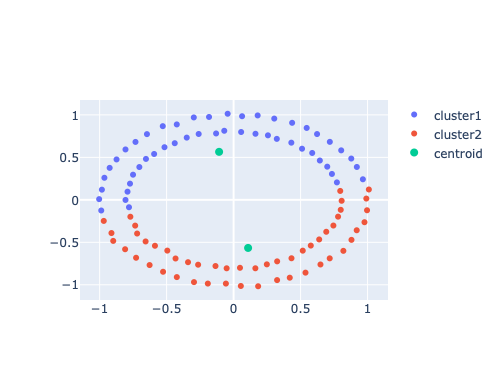
HDBSCAN도 다음과 같이 좋은 성능을 보이지 못했다.



그래서 noise를 0.01로 줄여서 시도해보았다.



물론 k-means는 다음과 같이 위와 같은 결과가 나왔고,



DBSCAN과 HDBSCAN 모두 cluster을 안 쪽과 바깥 쪽으로 잘 분류할 수 있었다.

