

Assignment #3

|  |  |
| --- | --- |
| 과목명. | 데이터사이언스 |
| 담 당. | 김 상 욱 교수님 |
| 제출일. | 2021년 06월 05일 |
| 공과대학 | 컴퓨터소프트웨어학부 |
| 학 번 | 2016025969 |
| 이 름. | 정지훈 |

****

**목차.**

**1. Summary of algorithm**

**2. Detailed description of codes**

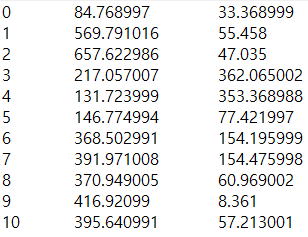
**3. Instructions for compiling source codes at TA's computer**

**4. Any other specification of implementation and testing**

**1. Summary of algorithm**

**- Explain Algorithm**

Input을 통해 Point의 좌표와 Id가 들어오면 DBSCAN 알고리즘을 통해 clustering을 진행하고 cluster에 포함된 점들의 id를 이용해 최종 output을 만드는 과정입니다. DBSCAN은 처음에 임의의 점 P1을 고른 뒤 거기서 주어진 반경 Eps와 반경 안의 최소 점 개수 MinPts를 이용해 모든 가능한 density-reachable인 점을 찾습니다. Density-reachable이란 주어진 P1을 이용해서 해당 점까지 directly density-reachable한 경로를 가지는 점들을 이용해 이어질 때입니다. Directly density-reachable는 임의의 점 P2가 P1로부터 반경 안에 있고 P1의 Eps 안에 MinPts보다 많은 점이 있는 Core Point일 때 성립합니다. 이를 통해 P1이 Core Point이라면 얻은 점들을 Cluster로 형성하고 border point라면 database에서 Cluster 되지 않은 다른 점으로 넘어 갑니다. 이를 반복해 모든 점을 계산한 뒤 종료합니다.



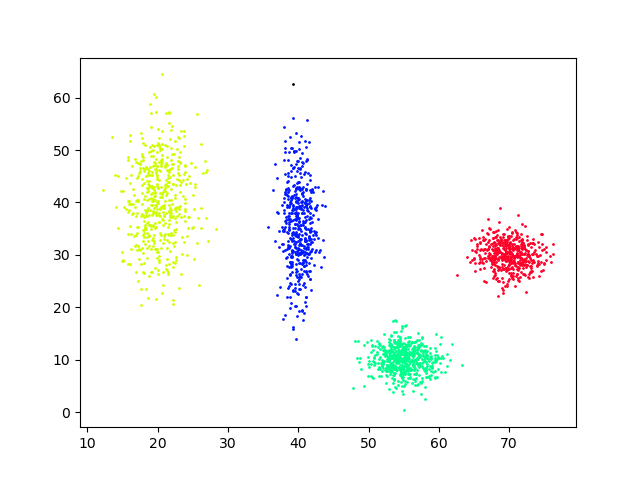
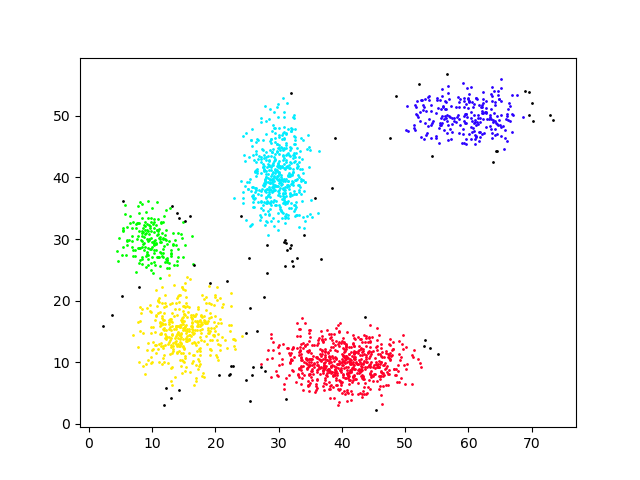
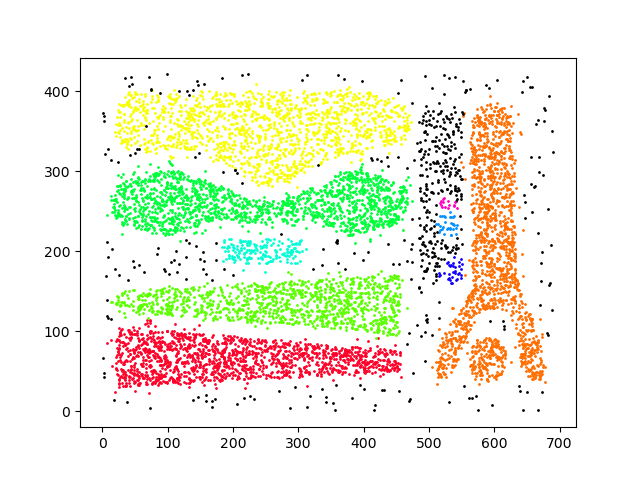
Input1.txt

Cluster 1



Cluster 2

**- Result Figure**

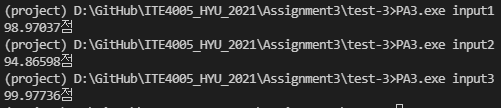


Input 1

Input 2

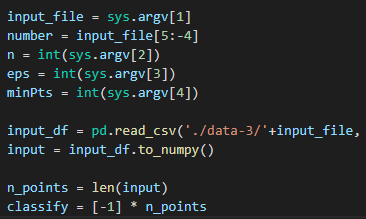
Input 3

**- Final Score**

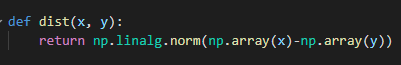


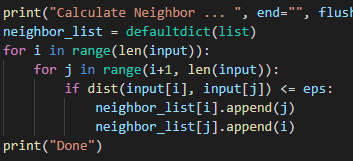
**2. Detailed description of codes**

**- Data Read**

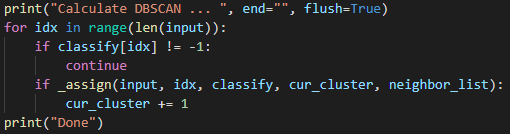
compile에서 주어지는 최대 Cluster 개수와 Eps, MinPts값을 저장한 후 input file을 읽습니다. 이때 input file을 읽을 때 주어진 object\_id를 index로 사용하여 Dataframe을 만듭니다. 만들어진 Dataframe을 이용해 총 점의 개수와 그를 이용해 각 점이 무슨 Cluster에 속하는지 정보를 담는 classify 배열을 만듭니다.

**- Distance**

두 점이 주어지면 Euclidian Distance를 구하는 함수입니다.

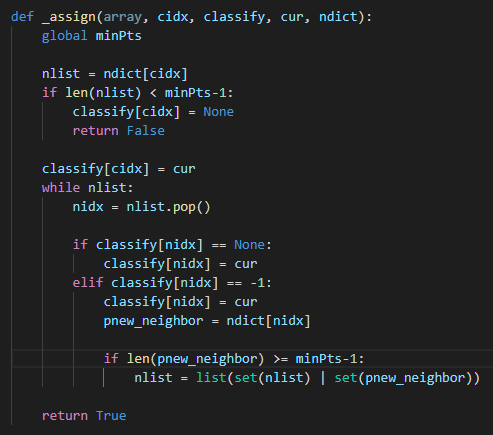
**- Get Neighbor**

기존에 새롭게 추가되어지는 점이 있을 때마다 이웃을 구해주면 중복되는 시간이 많아지기에 처음 Input을 받고 각 점에 대해 Eps 안에 있는 모든 점들을 구해 준 배열을 만듭니다.

**- DBSCAN iterative**

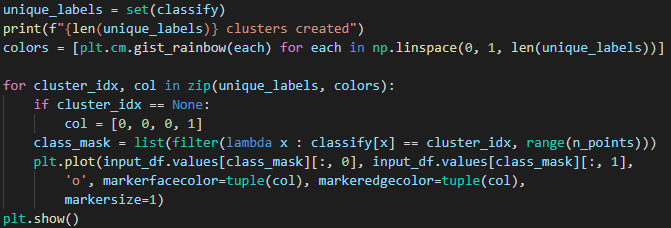
Cluster를 시작하는 점을 선택합니다. 이때 Core Point나 Outlier라고 Assign 되지 않은 점을 선택합니다.

**- Assign Cluster**



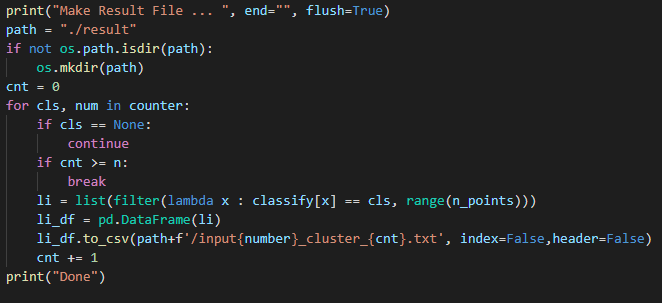
임의로 선택된 Point의 index를 이용해 Point의 이웃 점 집합을 구합니다. 만약 해당 점 포함 집합이 MinPts를 넘지 못한다면 Outlier 또는 Border Point이므로 모르겠다는 None을 할당한 뒤 새로운 Cluster를 만들지 않으므로 False를 반환합니다. 만약 넘었다면 해당 점은 Core Point가 되고 그 점은 Clustering 됩니다 이후 그 점을 통해 얻어지는 directly density-reachable된 모든 점들을 같은 Cluster에 속하게 정해줍니다. 이때 기존 None이었다면 Border Point이므로 Assign만 해주고 -1이었다면 새로운 점의 정보이므로 그 점을 기준으로 다시 directly density reachable을 neighbor list에 넣어 줌으로써 시작 점을 통한 모든 density-reachable 점들을 Assign합니다. 이 때 시간을 줄이기 위해 새로운 Neighbor 배열을 만들 때 중복을 제거함으로써 연산을 줄입니다.

**- Plot Cluster**

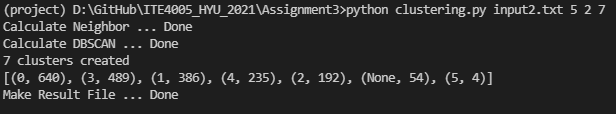


DBSCAN을 통해 구한 Cluster의 결과를 확인하기 위해 해당 점들의 좌표와 classify에 담긴 cluster 정보를 합쳐 Cluster Figure를 얻습니다.

**- Data Write**

Classify 배열의 결과를 통해 Cluster에 포함된 점의 개수가 많은 순으로 Output file을 만듭니다.

**3. Instructions for compiling source codes at TA's computer**



주어진 clustering.py 파일을 python을 통해 compile 하면서 이때 Input 파일과 cluster 개수, Eps, MinPts를 차례대로 입력해야 합니다. 파일은 상대경로를 통해 받으므로 해당 python파일이 위치한 곳에 있는 data-3 folder 안에 input.txt가 있어야 합니다. 또한 Result File은 python 파일이 위치한 경로에 result folder가 없다면 생기는 folder 안에 생기게 됩니다.

**4. Any other specification of implementation and testing**

Python version은 3.8.10이며 numpy의 버전은 1.19.2 pandas의 버전은 1.2.4, 그리고 matplotlib의 버전은 3.3.4 입니다.