**Data Science**

**Assignment3 : Clustering (DB Scan)**

2016025532 컴퓨터전공 심수정

**1. Summary of Algorithm**

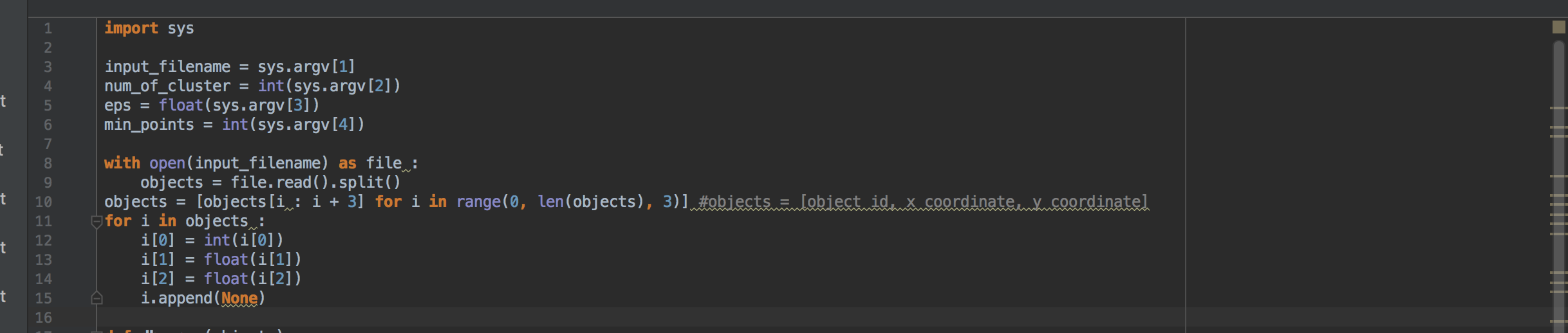
clustering은 주어진 data들의 특성을 보고 어느 object들끼리 가장 유사한것인지 알아내는 algorithm입니다. 이는 크게 distance 기반, density 기반이 있는데, 이번에 과제로 수행할 algorithm인 db scan은 density 기반입니다.

Density based clustering은 기본적으로 2개의 parameter가 존재합니다. Eps와 min points입니다. Eps는 maximum radius of their neighborhood이고, min points는 minimum number of points in an Eps-neighborhood입니다. 여기서 eps-neighborhood는 한 object로부터 eps 반경 내에 존재하는 object들을 말합니다.

여기에는 directly density reachable이라는 개념이 나옵니다. 이는 일정 min points를 만족하는 한 object로부터 eps 내에 도달할 수 있는 경우를 말합니다. 또한 density connected라는 개념도 존재합니다. 이는 directly density reachable은 아니지만, 다른 density reachable한 object들을 통하여 연결될 수 있는 경우를 말합니다.

Db scan에서 하나의 cluster는 maximal set of density connected points를 말합니다.

**2. Detailed Description of Code**



실행시 넘겨받은 argument로부터 input filename, number of cluster, maximum radius, minimum number of point in an eps of a point를 받아와서 각각 input\_filename, num\_of\_cluster, eps, min\_points에 string, int, float, int의 형태로 넣어줍니다.

그리고 input file을 열어 파일을 읽어 objects에 [[object id1, x1, y1], [object id2, x2, y2], …]의 형태로 저장한 후 이 list item에서 각각은 int, float, float의 형태로 바꿉니다. 그리고 cluster label이 아직 전부 지정되지 않았기 때문에 마지막으로 None을 추가해줍니다.

최종적인 obejcts의 형태는[[object id1, x1, y1, label1], [object id2, x2, y2, label2], …]의 형태가 됩니다.



db scan을 통해 clustering하는 algorithm입니다. 각각의 DB obejct을 모두 확인하되 만약 label이 이미 확정된 object라면 바로 건너뜁니다.

만약 label이 확정되지 않은 object라면 eps 반경의 object들의 density가 일정 수를 넘는지 확인한 후 넘지 않는다면 noise에 해당하는 -1을 label에 넣어줍니다.

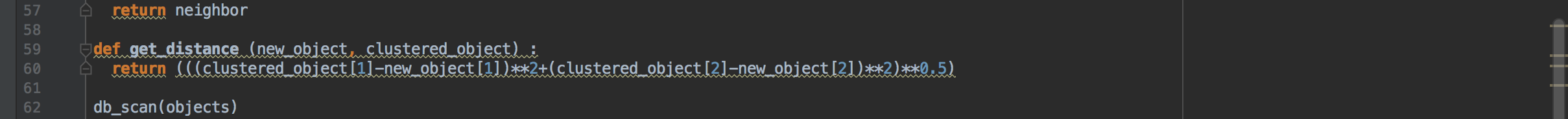
만약 넘는다면 이 반경에 해당하는 object들에서 자기 자신을 제거한 후 이웃 object들에 대해 iteration을 합니다. 원래 해당 object가 noise였다면 이를 현재 label로 바꿔줍니다. 밑에서 label이 이미 존재하는지 확인한 후 존재한다면 바로 다음 obejct를 확인합니다.

이웃 object에 label이 존재하지 않는다면 다시 이가 일정 이상의 density를 가지고 있는지 확인합니다. 일정 이상의 density를 가지고 있다면 이를 원래 이웃 obejct에 더 추가하여, 이에 대해서도 이 일을 반복하도록 합니다.

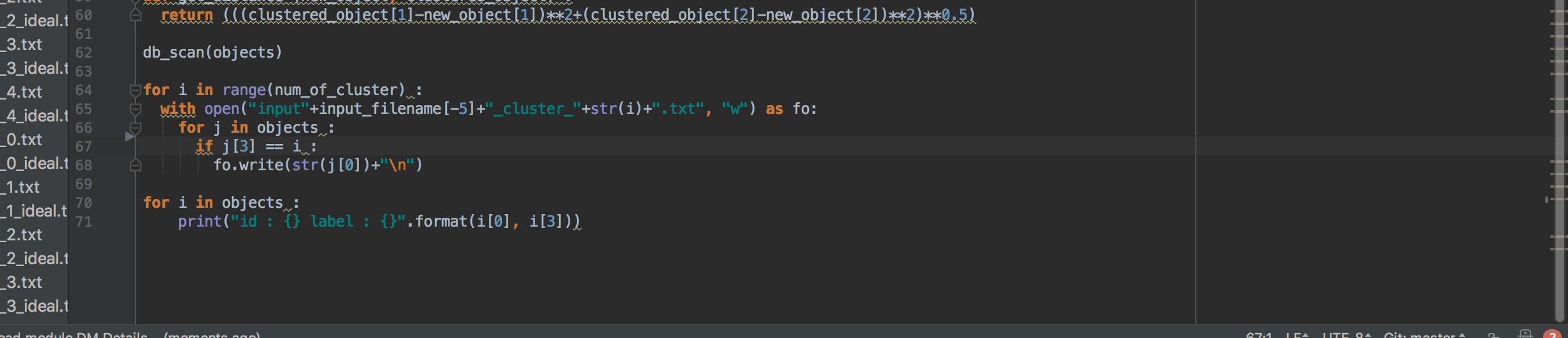
더 이상 추가된 object가 없고, 모든 이웃 object에 대해 이 일을 반복했다면 다음 label에 대해 이를 하도록 count의 수를 올리고, 만약 이가 정해진 number of cluster를 벗어난다면 algorithm을 종료합니다.



나와 object DB의 다른 object간의 거리를 계산하여 eps 안이라면 나의 이웃 object목록에 이를 추가하고, 최종적으로 이를 return해줍니다.



clustering되고 있는 object와 DB에 들어있는 다른 obejct와의 거리를 계산하여 return합니다.



input file에서 받은 object들에 대해 db scan을 수행하고, 이를 각각의 cluster label file에 써줍니다. 써줄 때는 object id를 쓰고 줄을 바꾸는 format으로 써줍니다.

후에는 각각의 object id와 label을 출력합니다. 이는 과제 명세와는 관련이 없는 cluster의 결과를 바로 확인하기 위한 code입니다.

**3. Instruction for Compiling Source Code**

이 코드는 python3을 기반으로 작성되었습니다. 따라서 python3가 설치되어 있어야 합니다.

“**python3 clustering.py [input file name] [number of cluster] [maximum radius] [min points]**”의 형태로 complie, 실행하면 됩니다.

**4. Other Sepcification of Implementation and Testing**

각각의 object는 [object id, x position, y position, cluster label] 형태의 list를 가지고 있습니다.

또한, 데이터가 많아질수록 program의 실행시간이 다소 길어집니다. 무한히 도는 loop는 아니기에 program이 정상적으로 끝날 때까지 기다리면 clustering의 결과를 이상 없이 볼 수 있습니다.

test 결과는 아래와 같이 input1에 대해서는 약 99%, input 2에 대해서는 약 95%, input3에 대해서는 약 100%에 달하는 성능을 내는 것을 확인할 수 있습니다.

