1-NN classification gunpoint

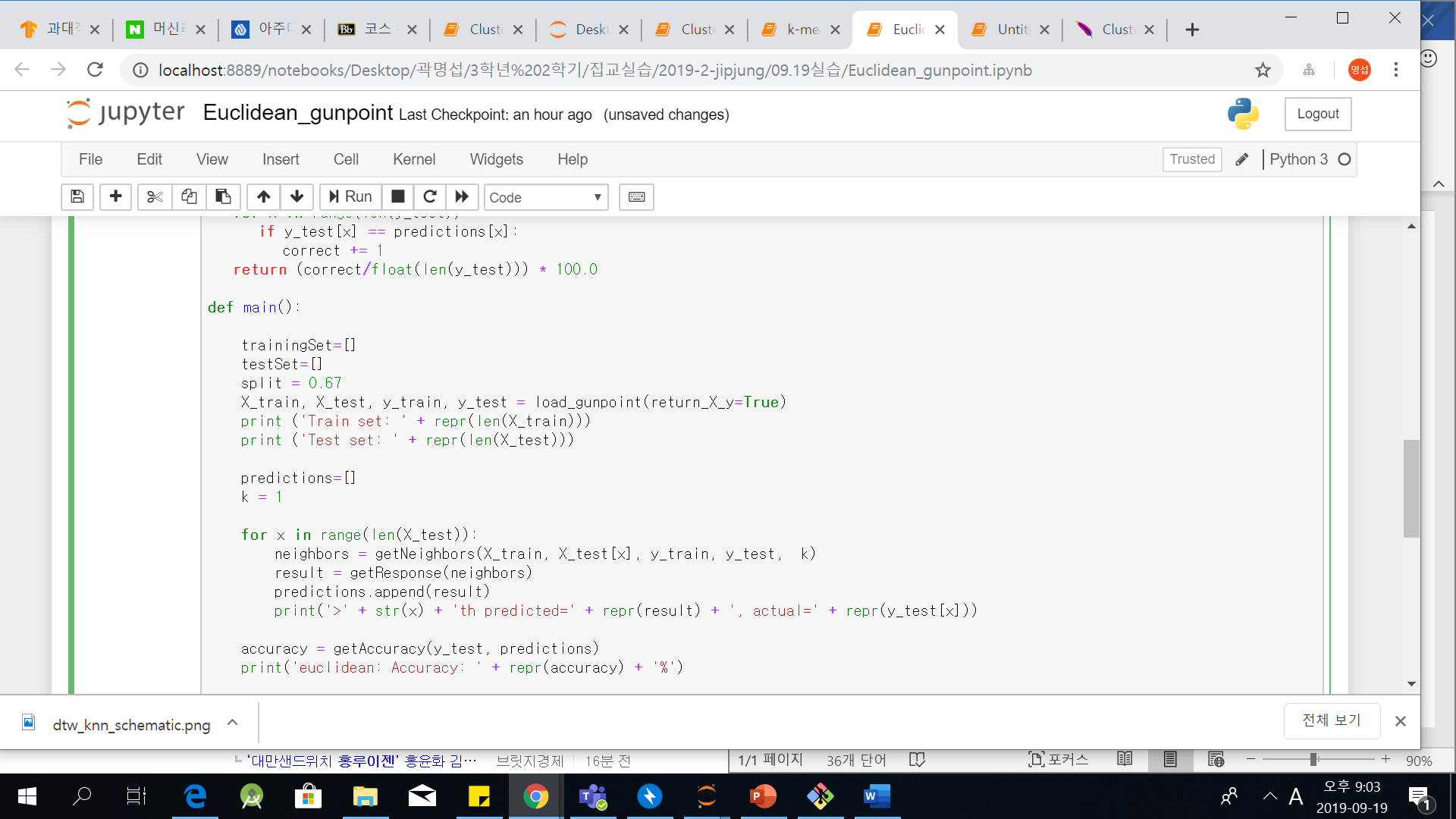
201520991

곽명섭

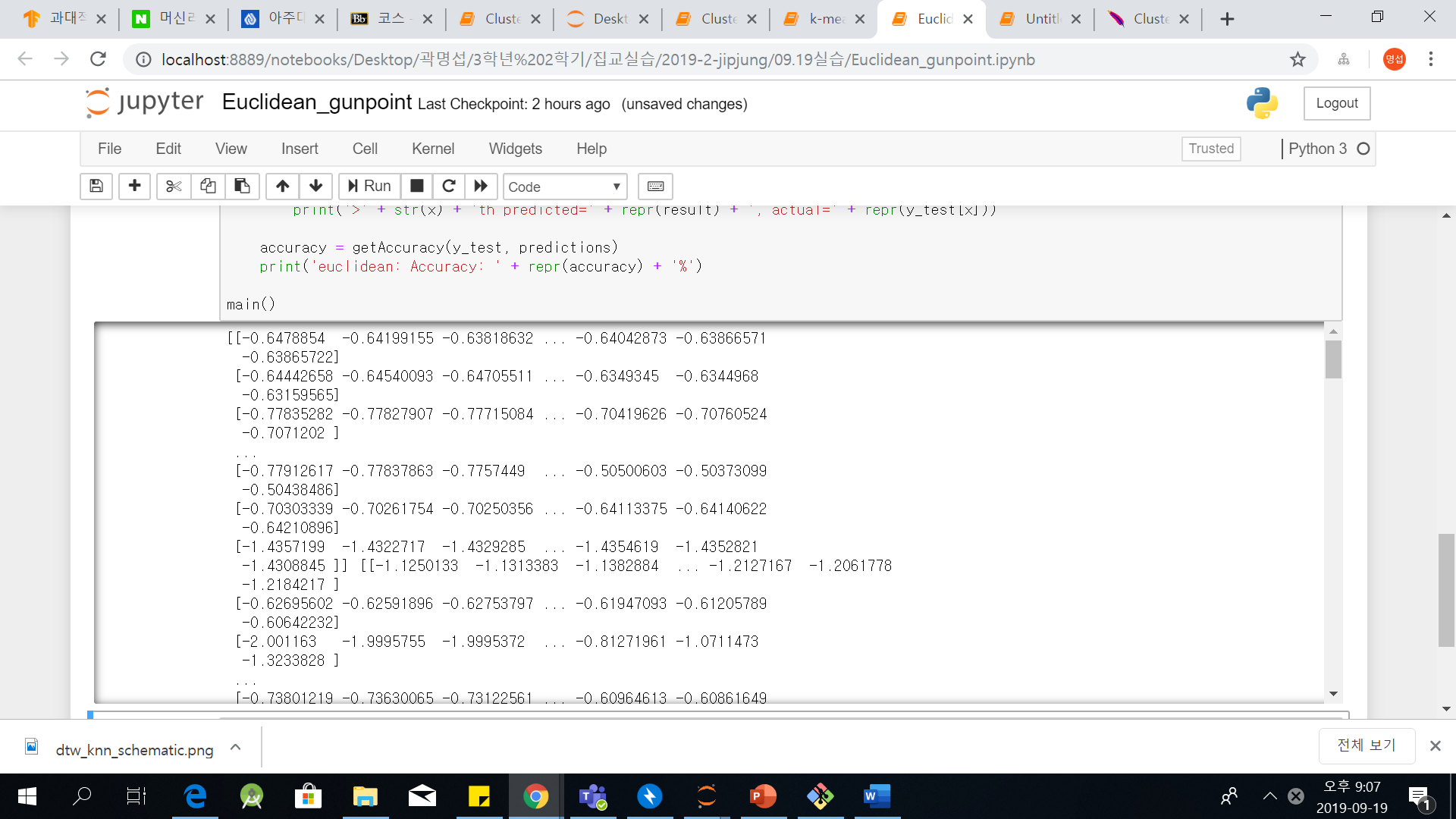
오늘 실습 과제는 gunpoint라는 자료를 불러와서 1-NN classification을 하는데 Euclidean distance, DTW distance, Manhattan distance를 각각 이용하여 하는 실습이었는데 일단 뼈대가 되는 코드는 학기초에 했던 KNN실습의 코드를 이용하여 실습을 하였다.

우선 데이터를 로드 하는 부분인데

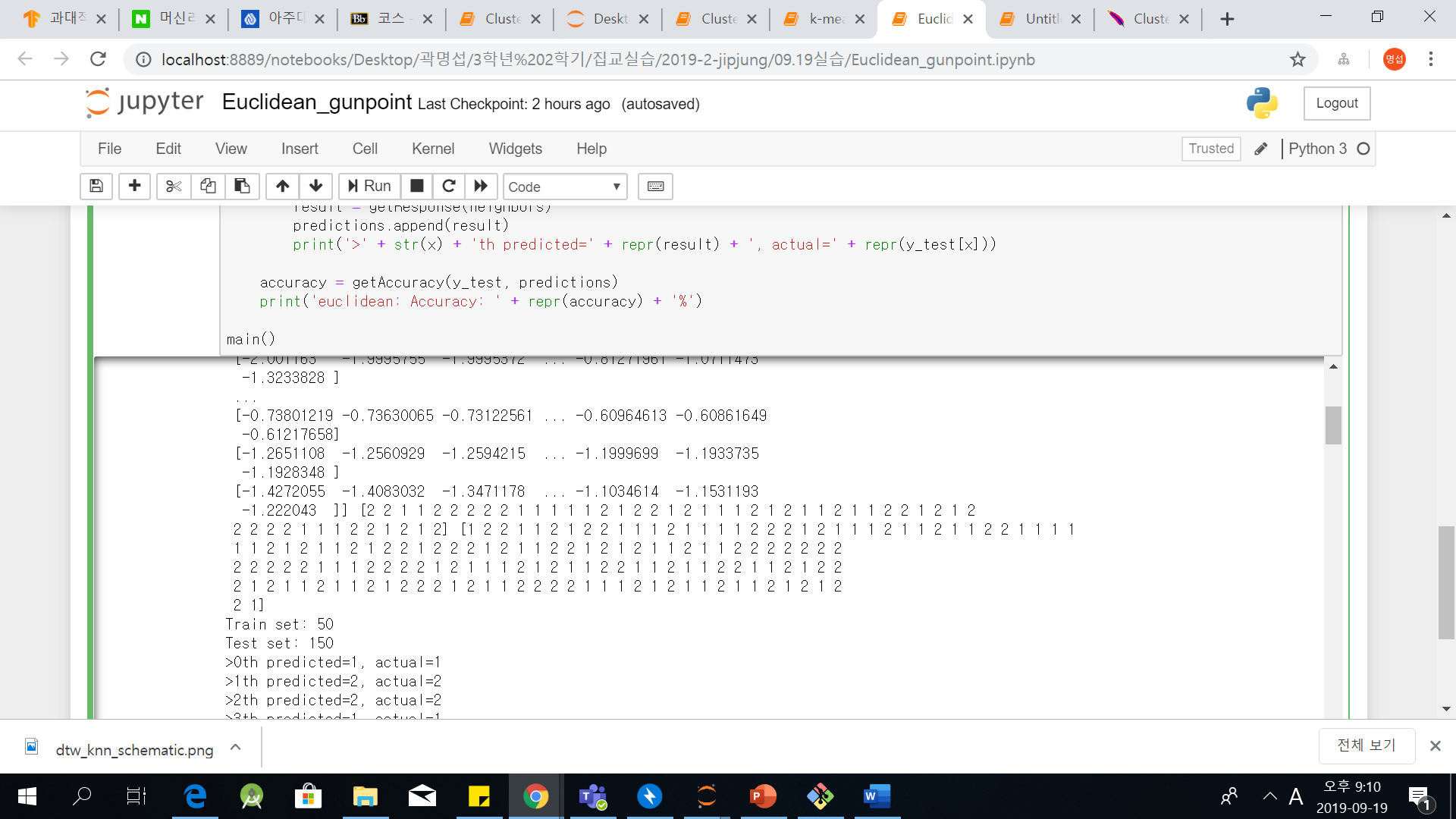
from pyts.datasets import load\_gunpoint



이렇게 import만 해주고 load문 한줄만 써주면 데이터는 쉽게 불러올 수가 있었다. 그리고 실습과제의 뼈대에서 데이터의 틀을 맞춰 주고 파싱을 해주기 위해서 gun point data가 어떤식으로 생겼는지를 알았어야 했는데 처음에는 이것조차도 헤매고 있다가 print 문을 쓰면 되겠다는 생각에 데이터를 출력해 보았다. 출력해본 결과

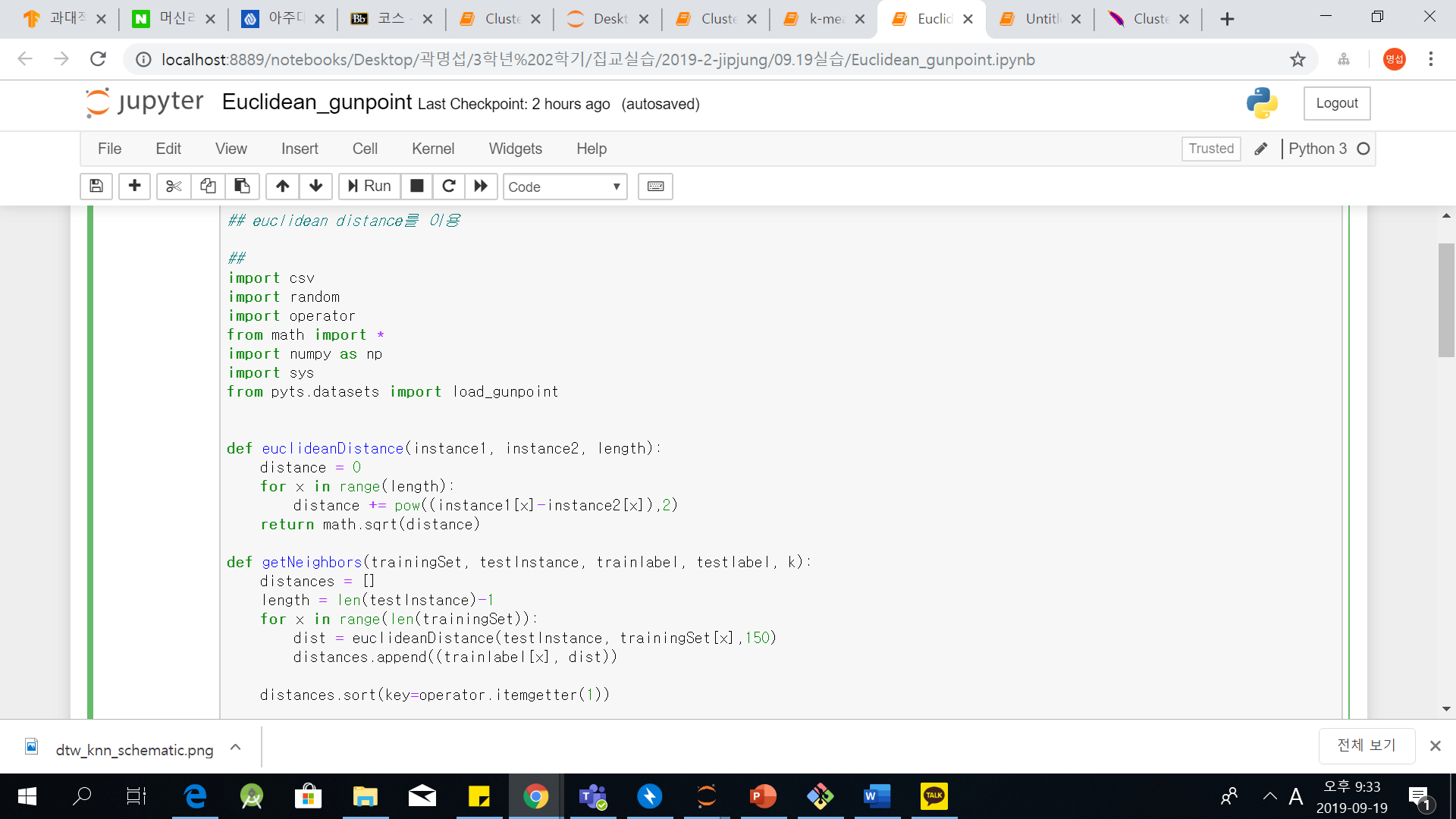


X\_train과 X\_test에는 이런식으로 실제 데이터가 저장이 되어 있었고



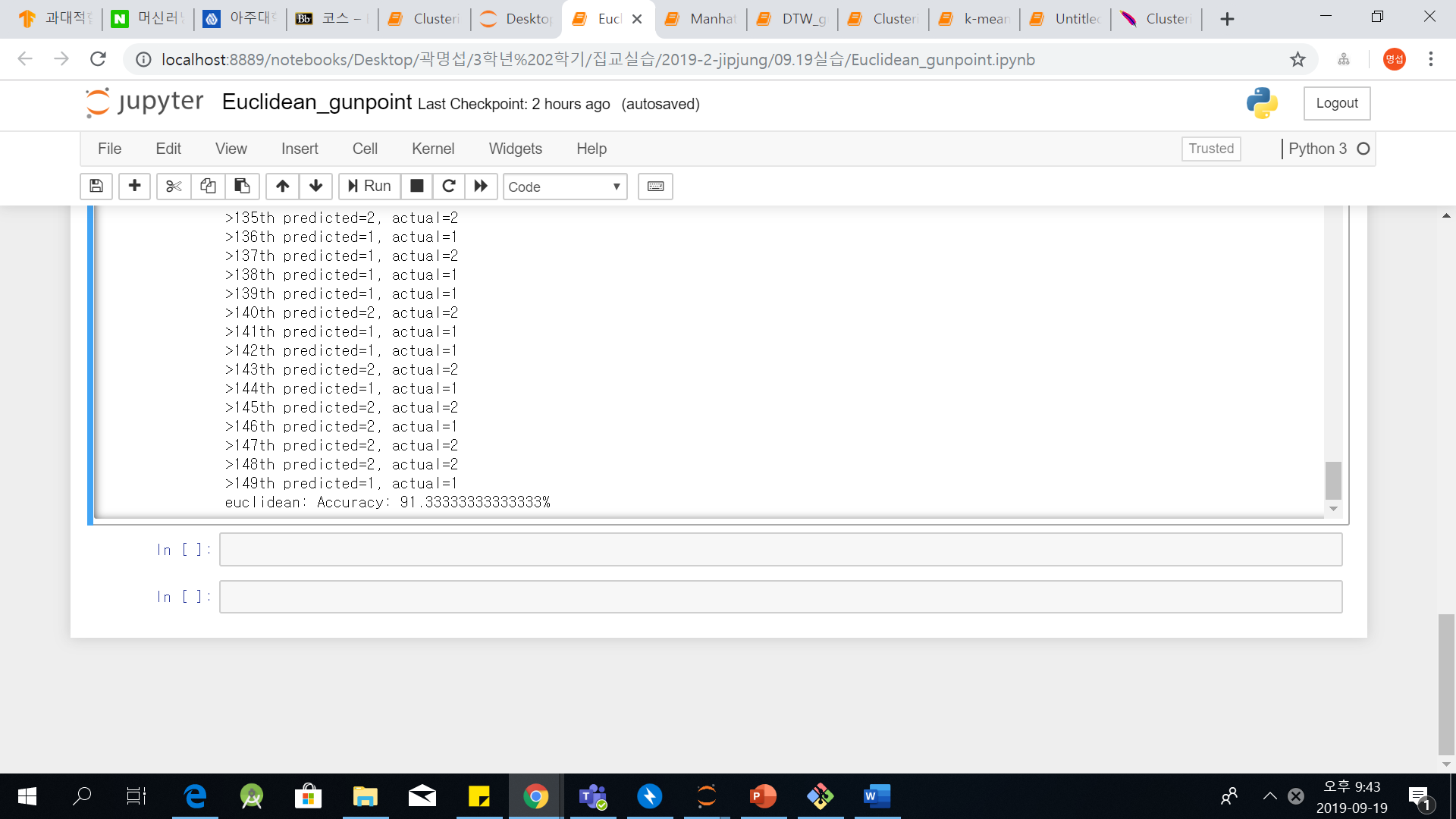
y\_train과 y\_test에는 이런 식으로 class값이 1과 2로 저장되어 있었다.

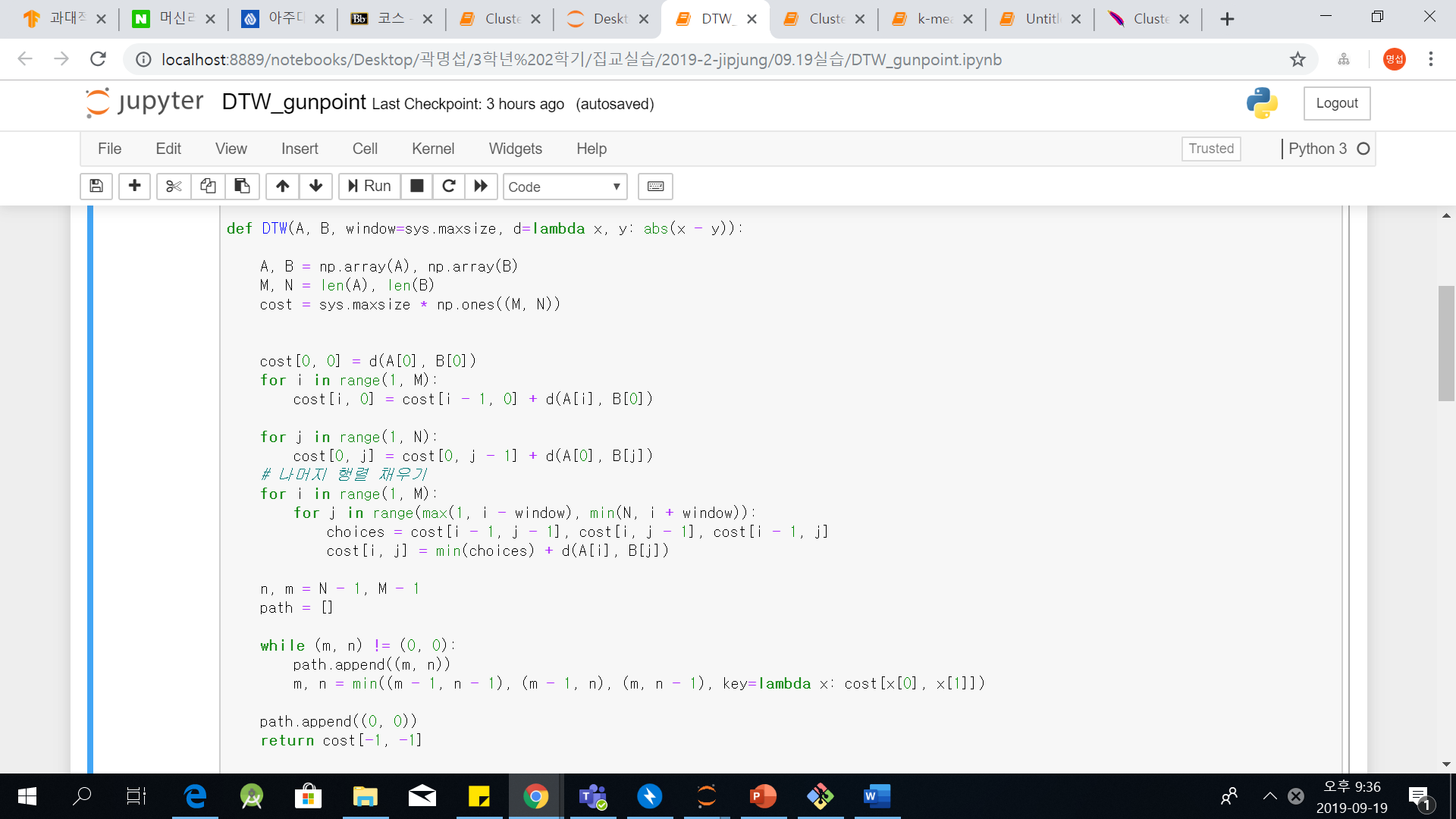
그래서 이것에 맞추어서 데이터형을 맞춰 주는데 만 상당히 오랜 시간이 걸렸다.



Euclidean distance를 구하는 알고리즘이다.

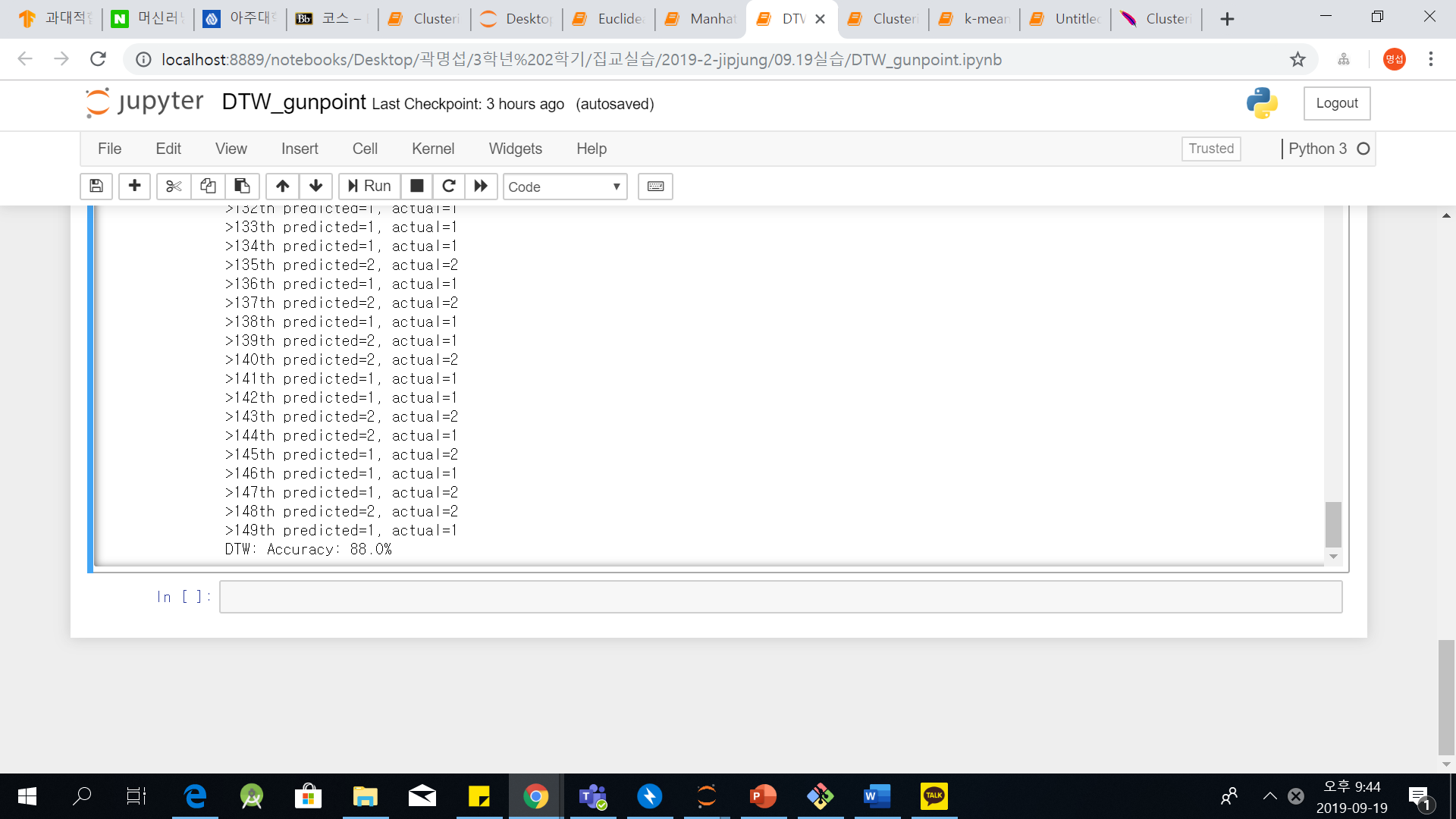
그리고 Euclidean distance를 이용하여 gunpoint자료를 classification한 결과 accuracy는 91.3333%정도가 나왔다.

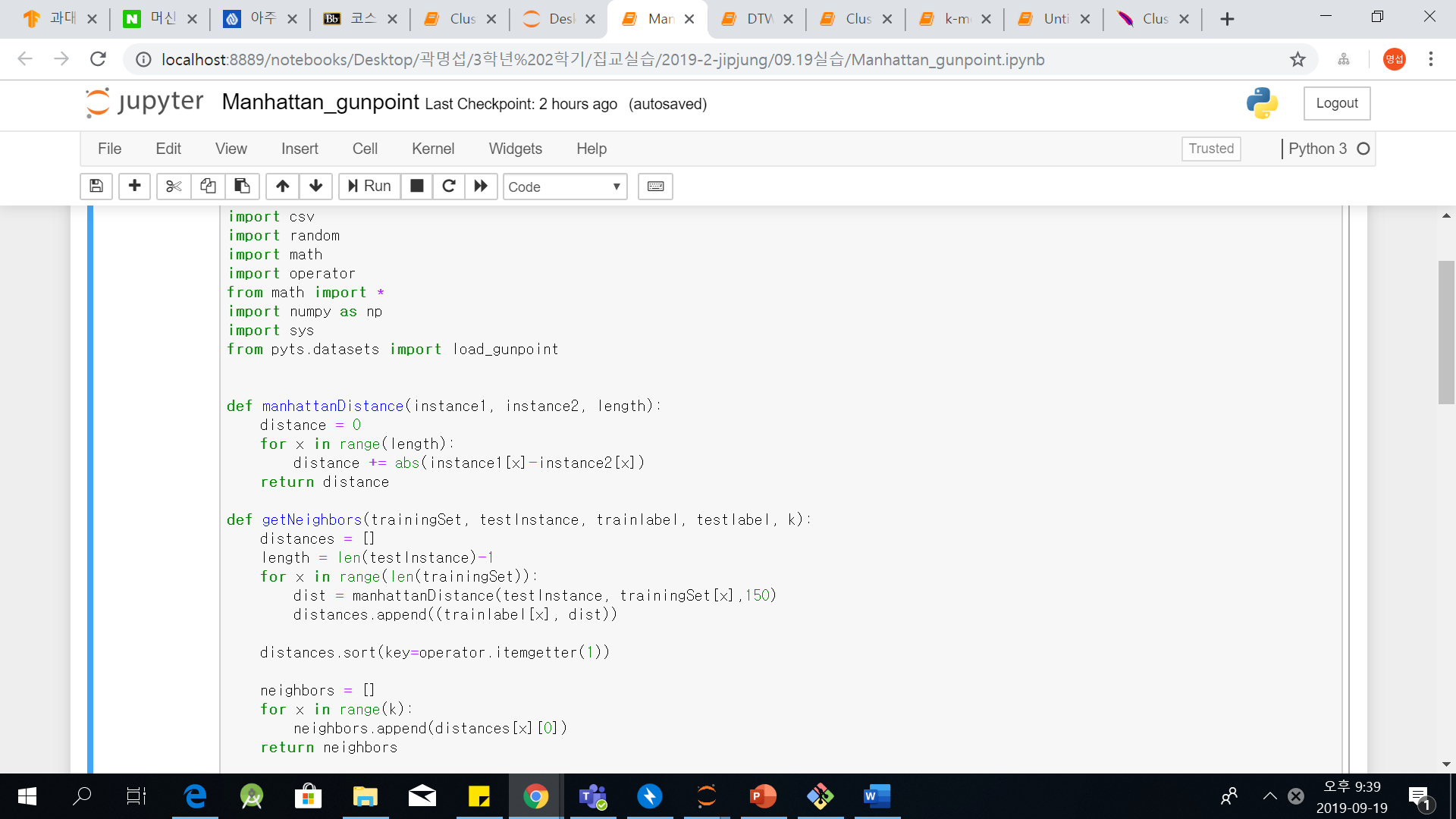




DTW distance를 구하는 알고리즘이다.

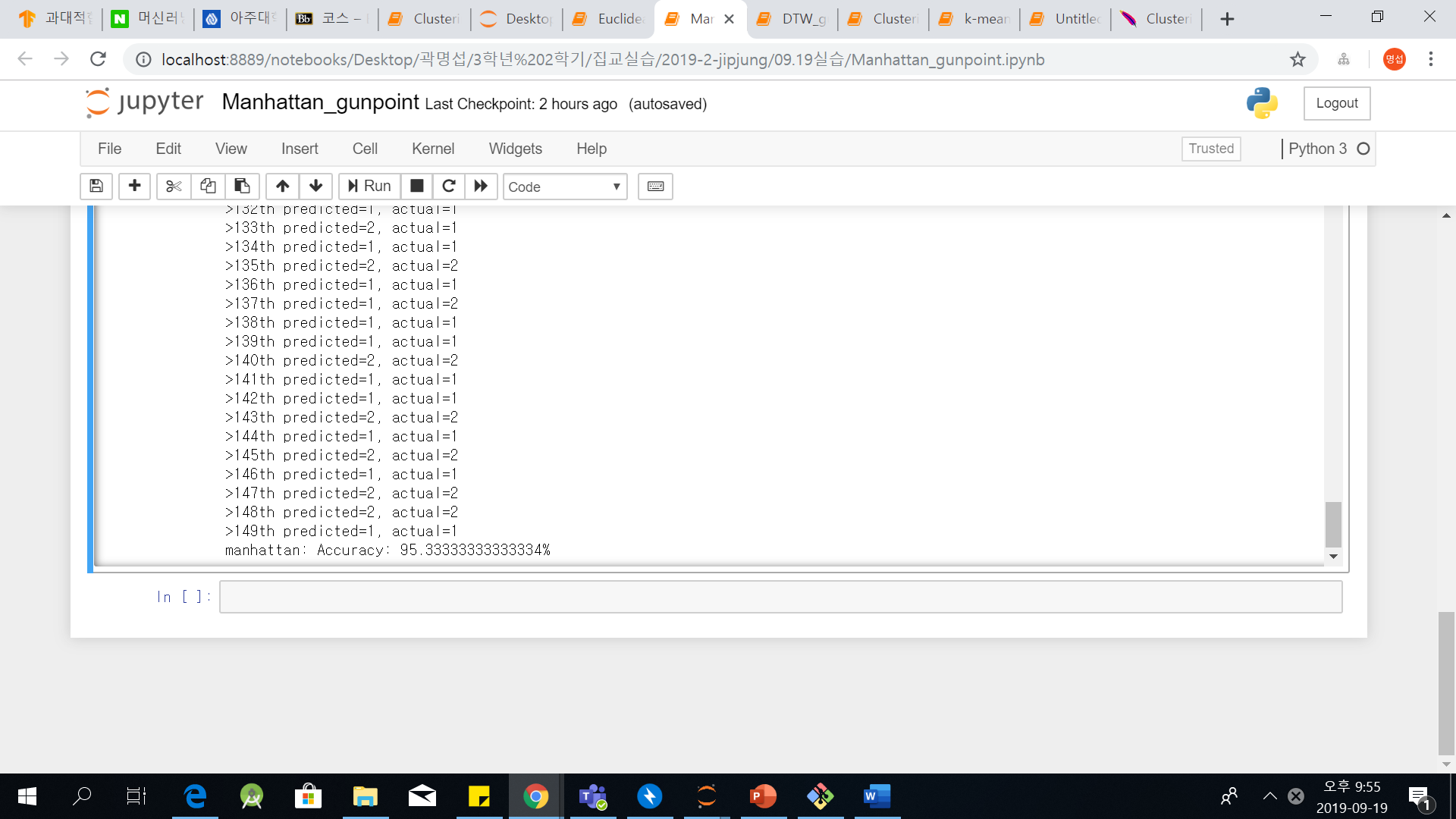
DTW distance를 적용하여 나온 accuracy는 대략 88%정도가 된다.





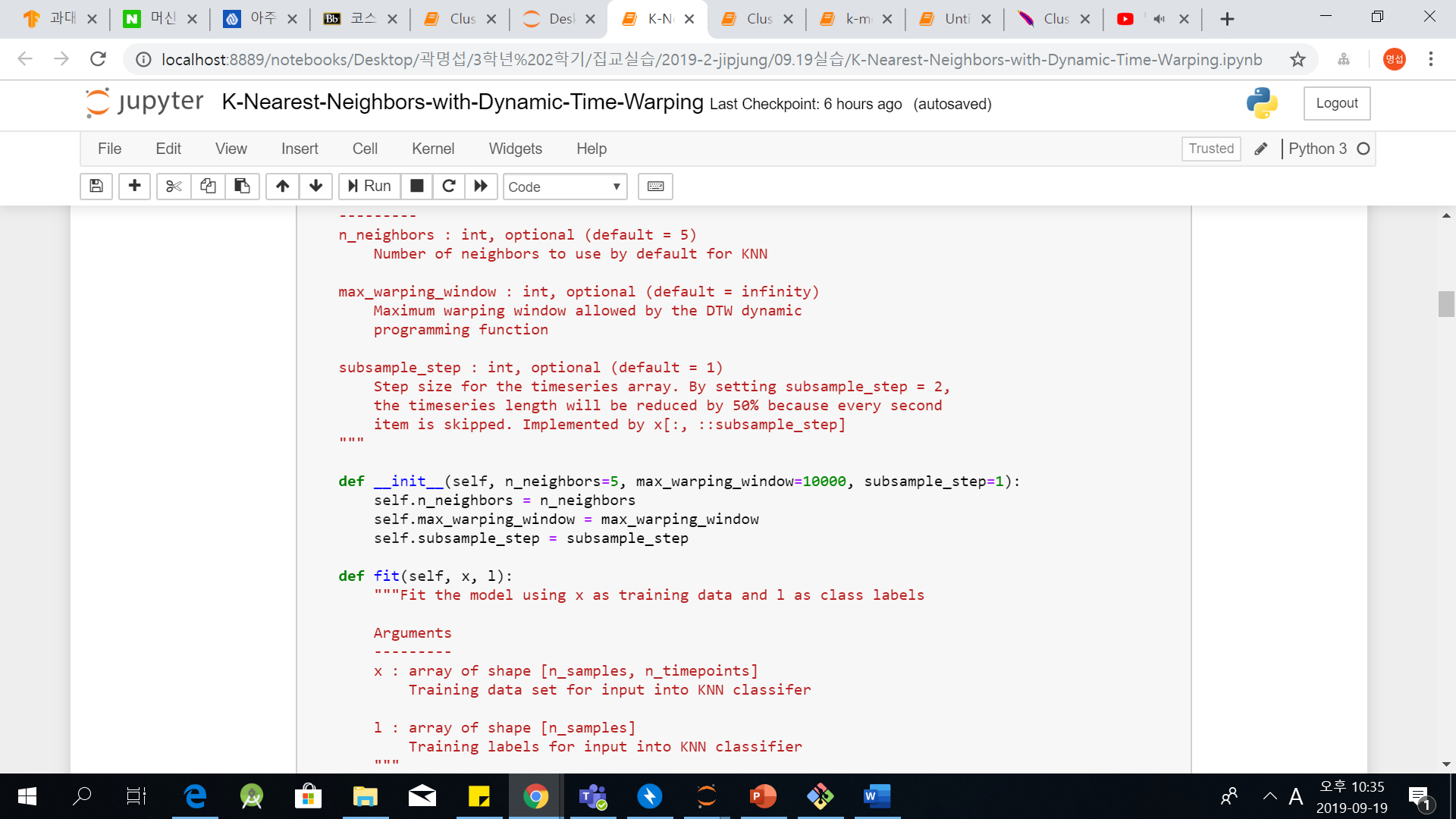
manhattanDistance를 구하는 알고리즘이다.

DTW distance를 적용하여 나온 accuracy는 대략 95%정도가 된다.

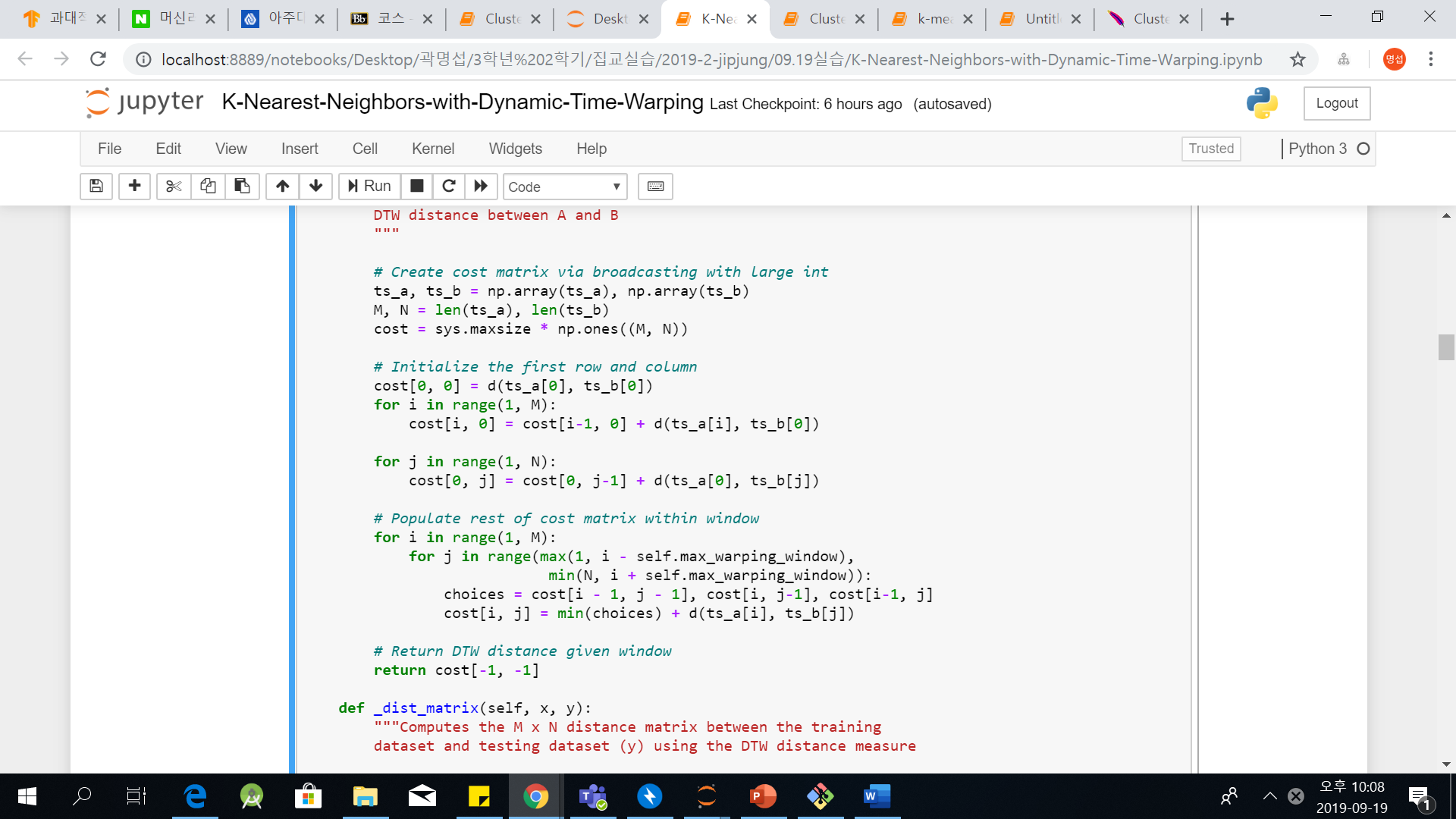


코드는 따로 첨부되어 있습니다

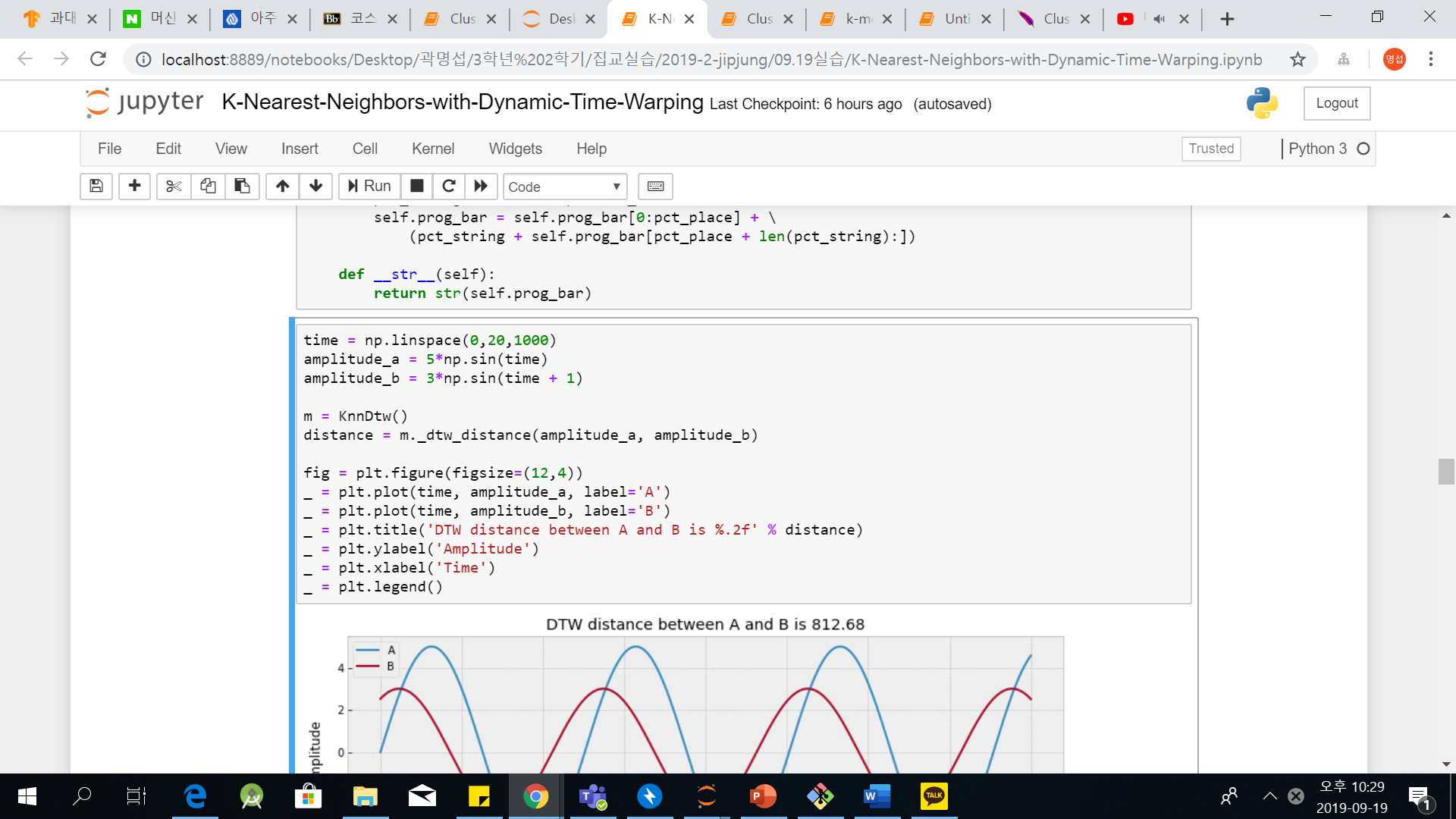
KNN with DTW

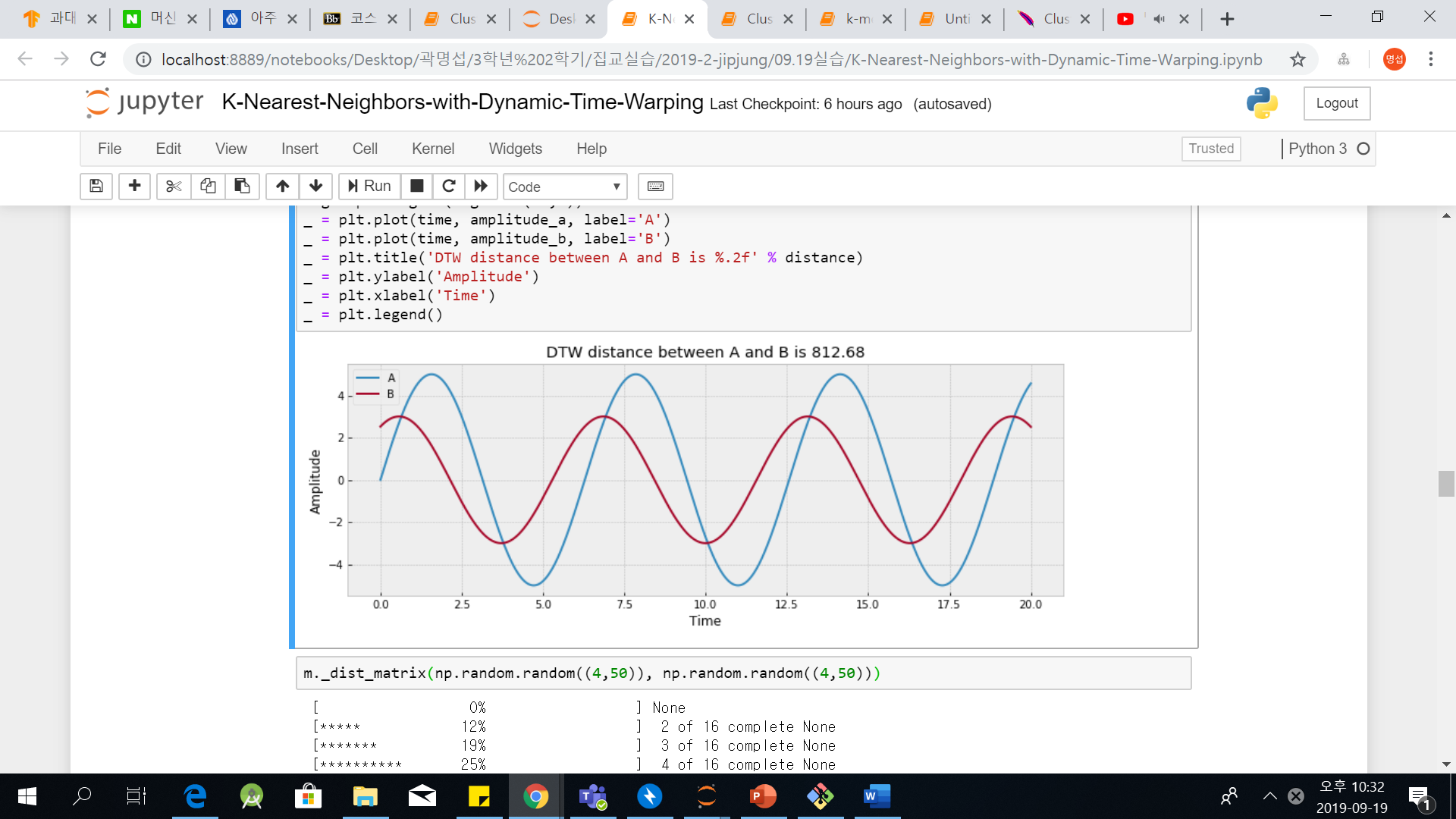


DTW를 시작할 때 초기화를 해주는 코드부이다.

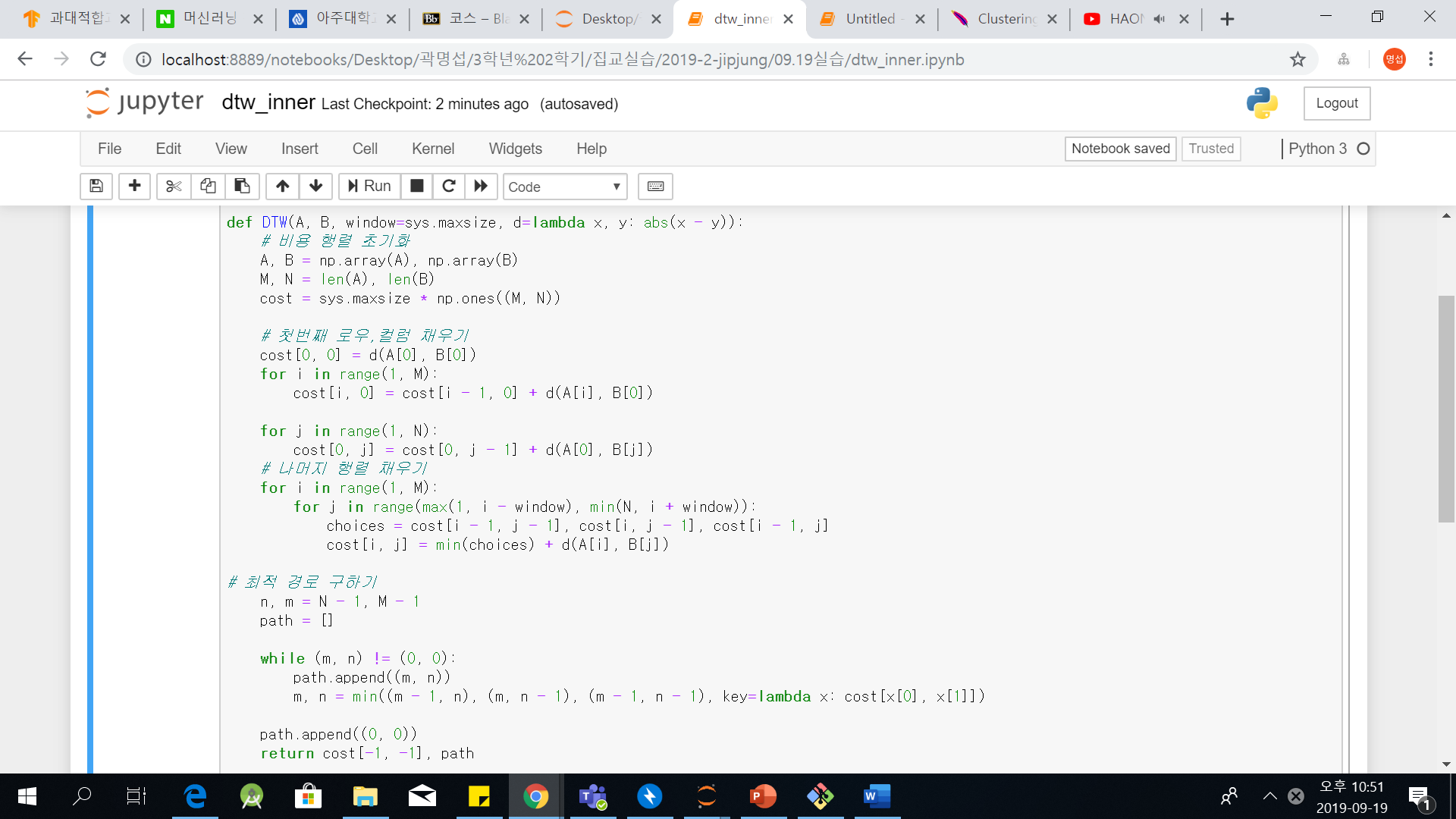


여기가 DTW distance를 구하는 알고리즘인데 2개의 timeseries numpy 배열간의 약간 성능을 위해 변형시킨 dtw distance를 반환해주는 역할을 한다.



A와 B간의 time series data를 표로 출력해주는 부분이다.  


DTW 내부 코드이해



DTW의 내부를 이해하기위한 코드인데 아까 위에 gunpoint data를 불러와서 dtw를 사용해서 classification을 할때도 사용을 했던 코드인데

맨 위에서 array를 초기화를 해주고 첫번째 low,column을cost 행렬도 만들어주는데 이때 (0,0)에는 A,B 요소의 첫번째 값의 차로 넣어주고 행에는 A의 처음부터 끝까지의 요소와 B의 첫번째 요소간의 차이를 더해서 넣어주고 열에는 B의 처음부터 끝까지의 요소와 A의 첫번째 요소간의 차이를 넣어준 것을 더해서 넣어준다 이후에 DTW 알고리즘을 통해서 cost 행렬을 채워주는데 이때 choice는 (i,j)를 채울때 (i-1,j-1),(i,j-1),(i-1,j)의 3요소를 선택하여 최소인 값과 A[i],B[j]의 차를 더해주면 된다. 그리고 그밑은 path를 구해주는 알고리즘을 나타낸 것이다