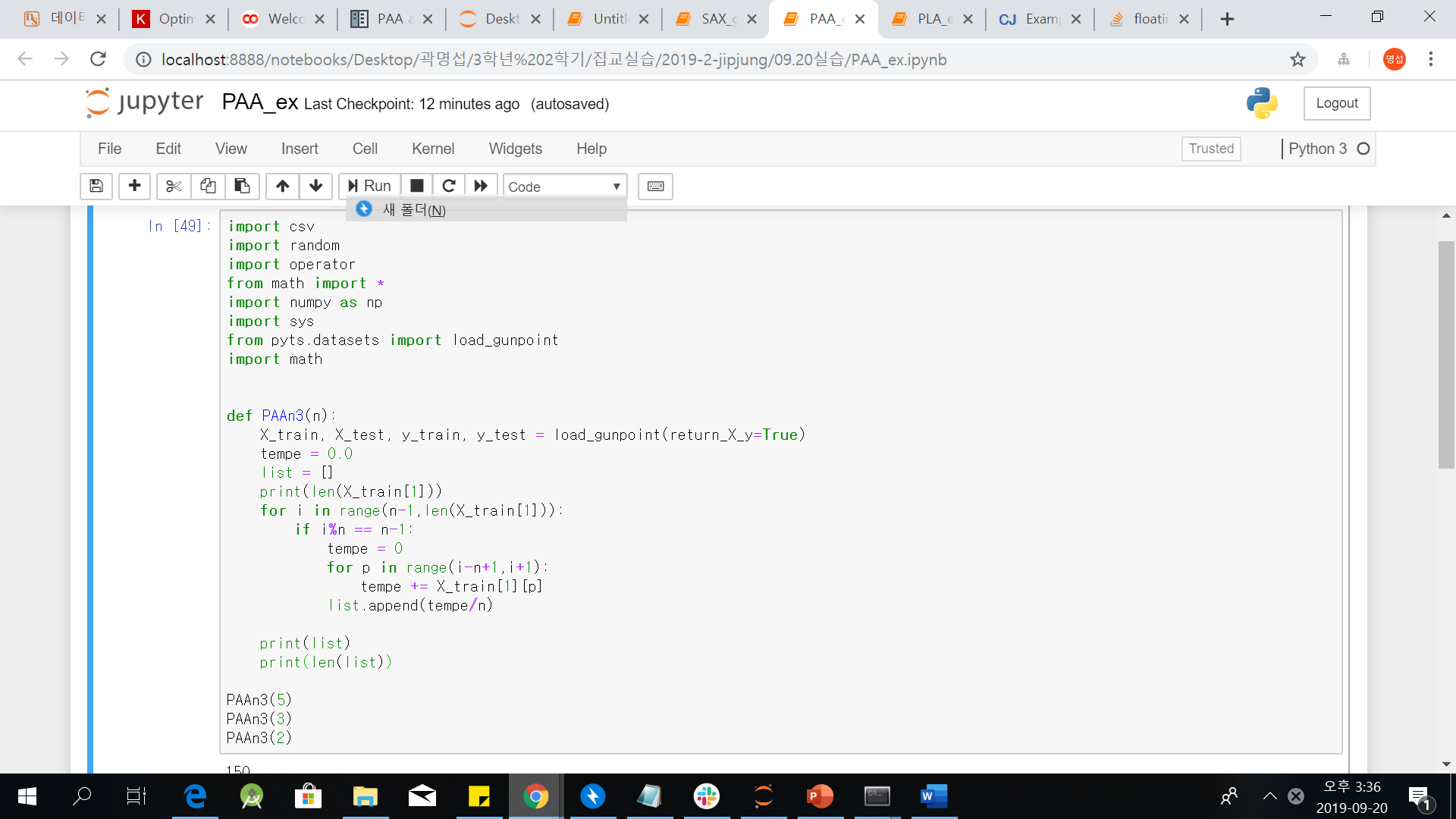
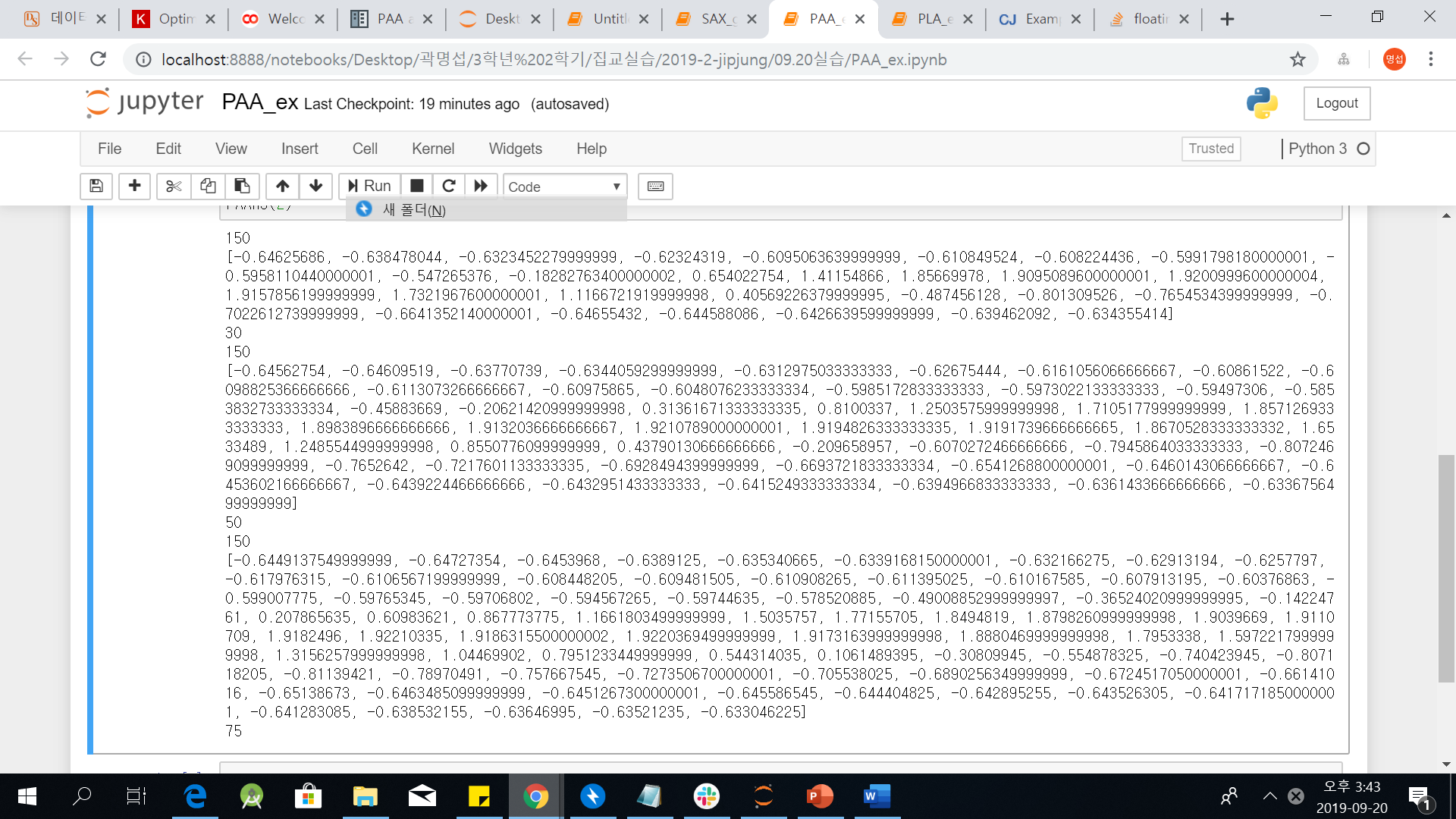
PAA,PLA,SAX구현

201520991

곽명섭

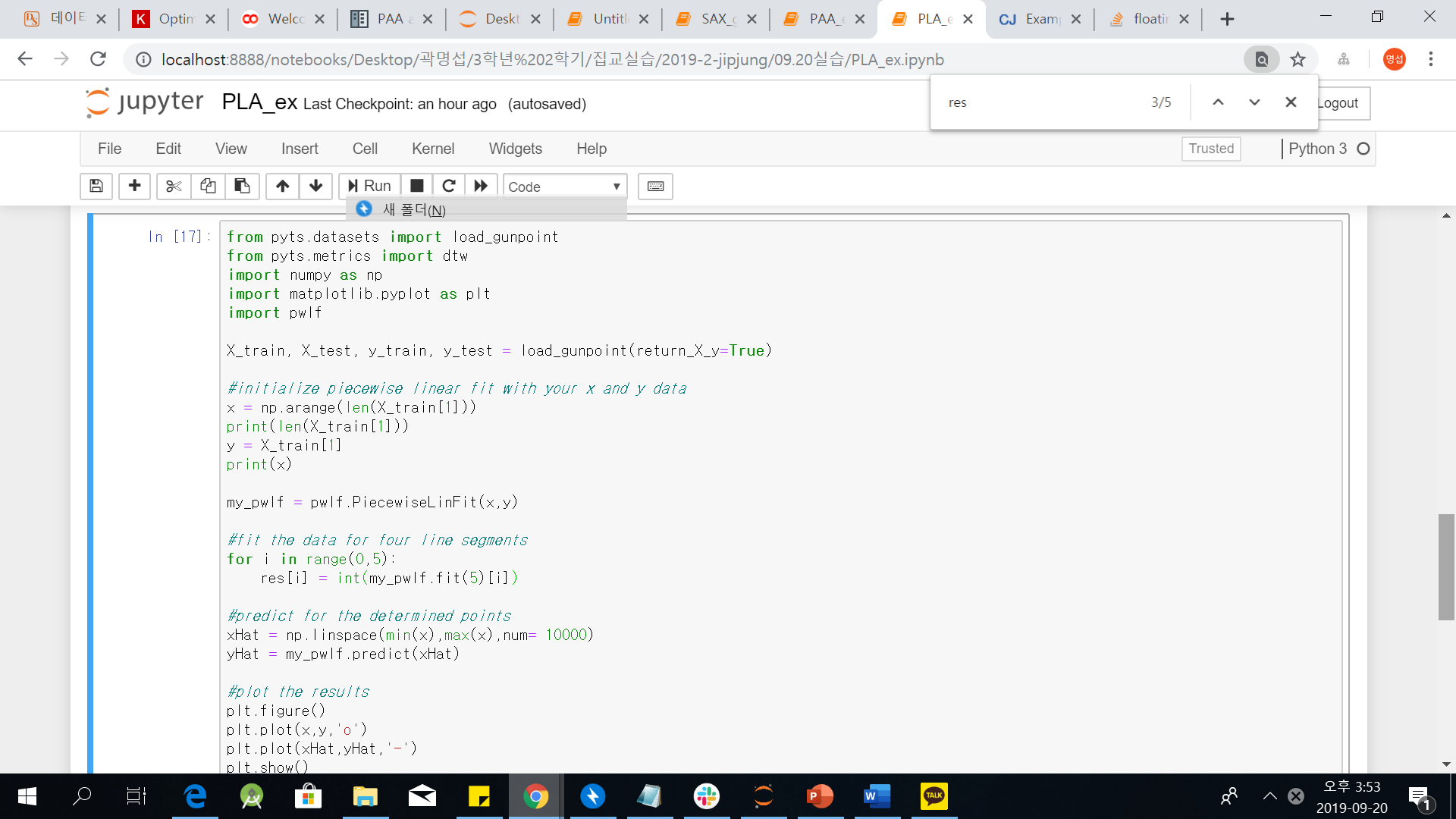
오늘 실습시간에는 여러 데이터들에 PAA,PLA,SAX등을 구현하거나 모듈을 적용하는 실습을 하였습니다. 맨처음으로 PAA를 직접 구현을 해보았는데



맨 처음은 gunpoint 데이터를 불러와서 PAA를 적용하는 함수를 구현 하는 것 인데, 아직 파이썬 문법이 많이 낯설어서 np를 사용하면 더 효율적으로 할 수 있었겠지만 어떻게 이용해야 할지를 모르겠 어서 그냥 이중 for문을 사용하여 데이터의 n값을 parameter로 놓아주어서 새로운 list에 값을 추가해주는 식으로 구현하였다. 

n=3일때의 결과값 출력이다. 올려주신 ppt와 일치한다

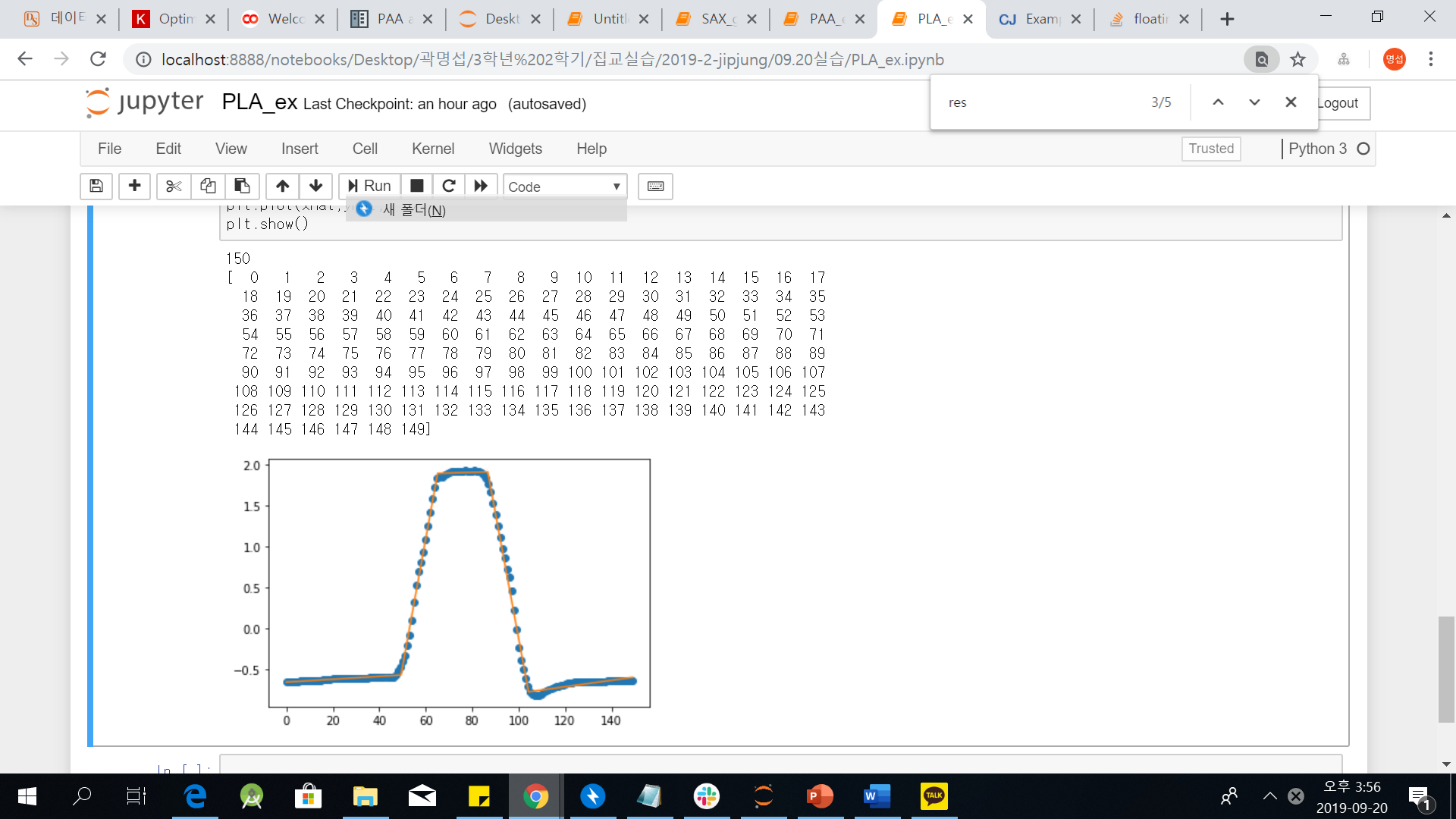
다음은 PLA인데



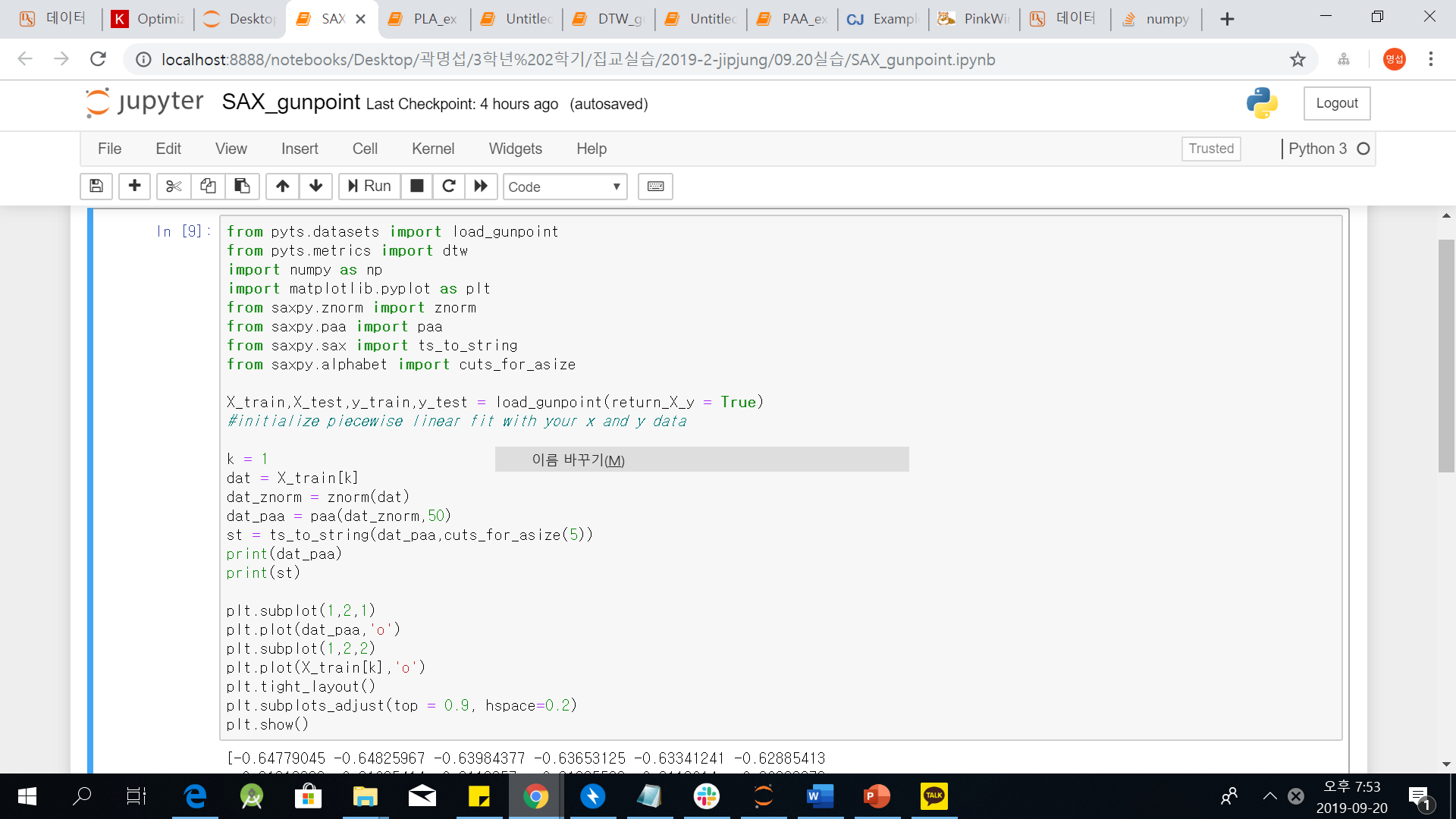
이것도 gunpoint를 불러와서 모듈을 이용해서 PLA를 적용하는 과정과 표를 출력하는 과정이다. 원래 ppt에 올려주신 코드에 exercise의 요구사항 중에

Res의 x값의 Float를 integer로 변환

변환된 값을 반올림한 후 approximation 값 (yHat)을 각 point의 y로 구함이라는 조건이 있어서 원래 코드에서 res라는 배열 값들을 int형으로 값으로 형 변환해서 넣어주는 과정을 넣어 주었습니다. 밑에는 결과 값입니다.

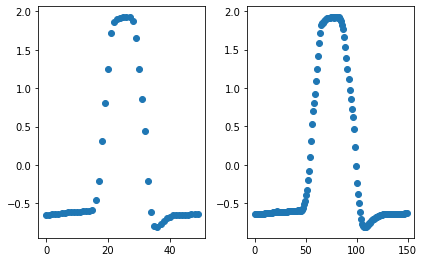


다음은 SAX코드인데

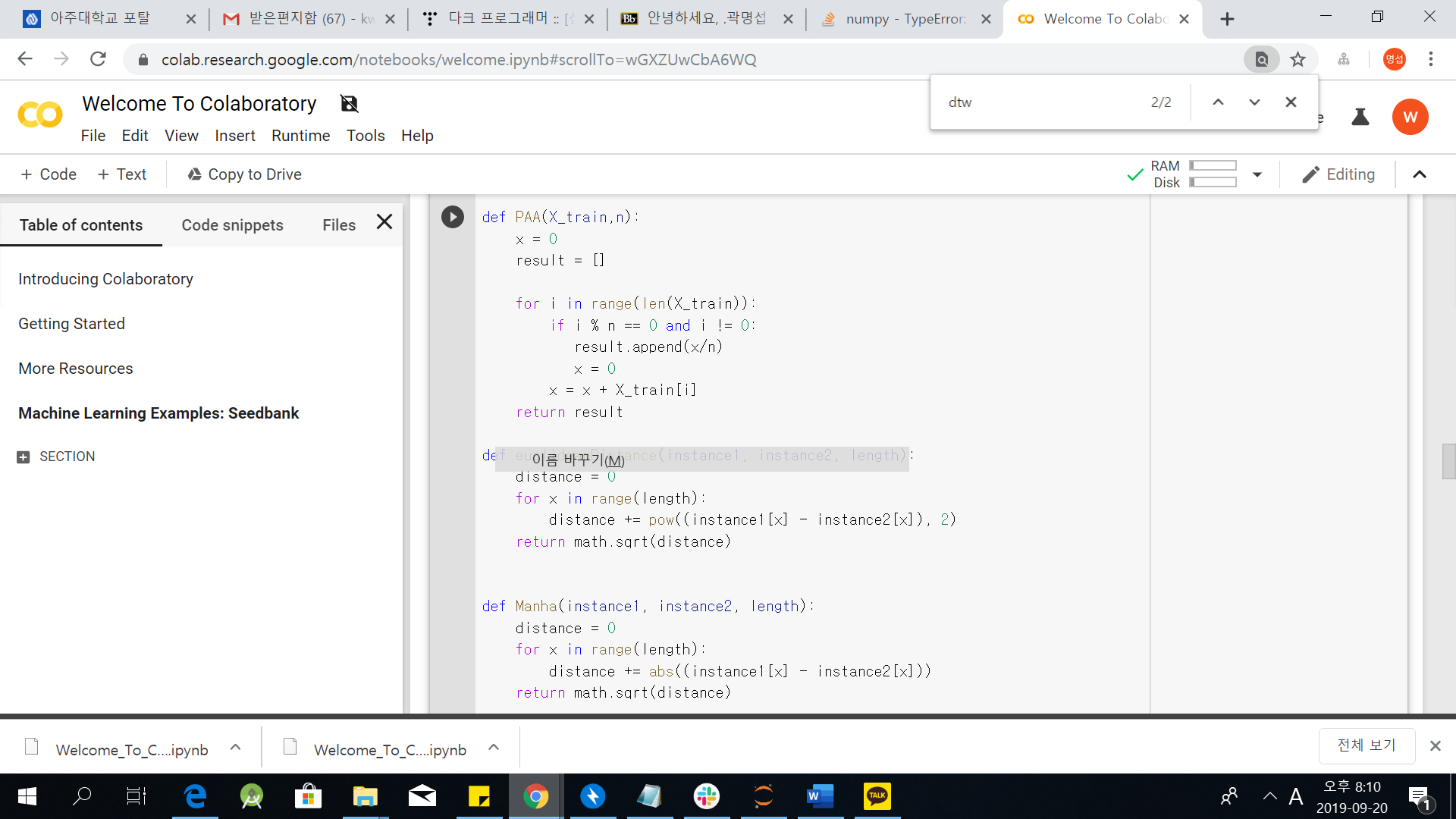


이것도 gunpoint를 불러와서 모듈을 이용하여 SAX를 적용하는 과정과 표를 출력하는 과정이다. 밑에 알파벳은 SAX를 알파벳으로 바꾼 결과 값이고 밑은 그것을 나타낸 표이다. a,b,c,d,e 5가지로 나타낸다. 왼쪽은 SAX를 사용하여 차원을 낮춘 결과이고 오른쪽은 원래 그래프이다

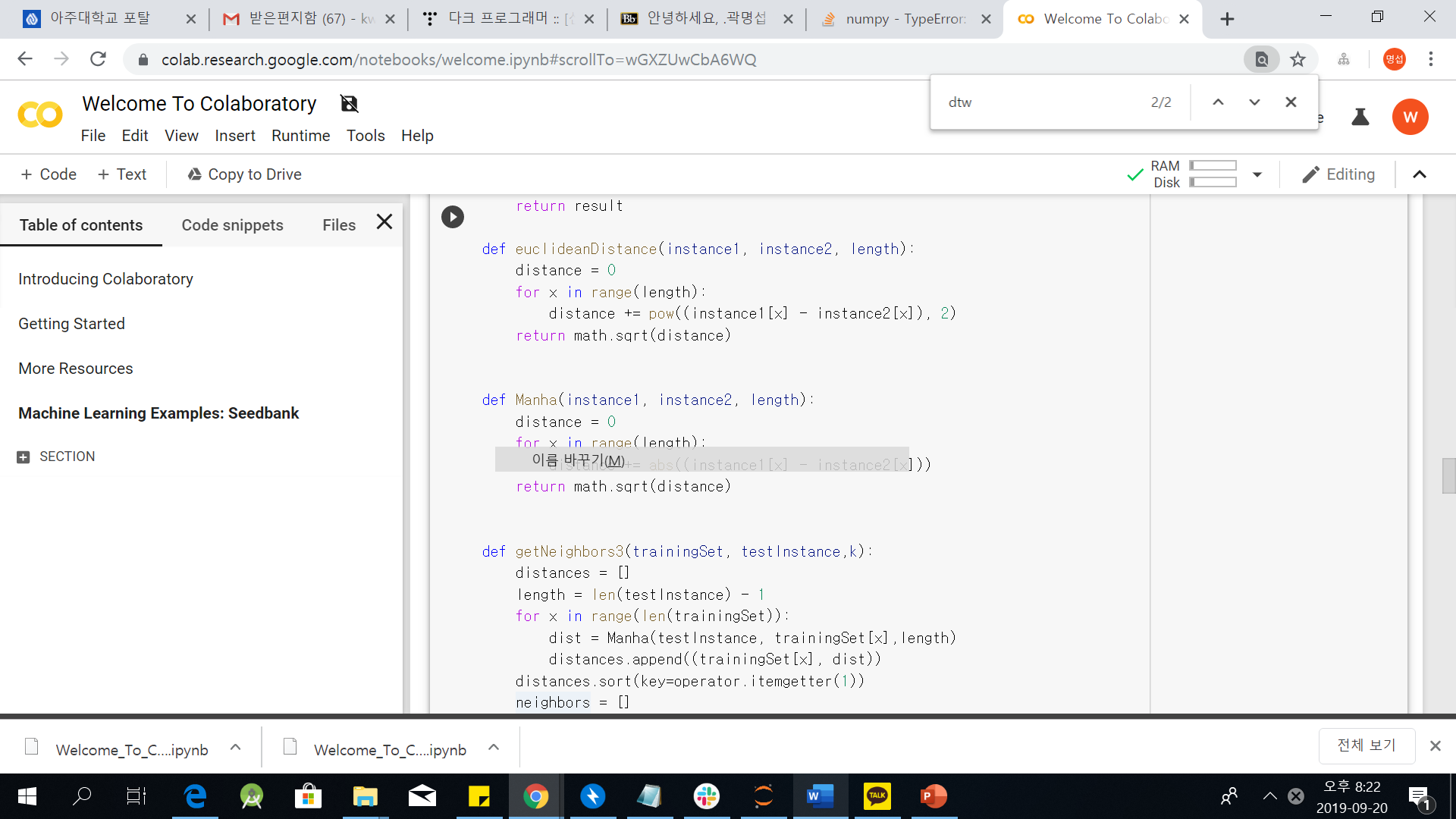
bbbbbbbbbbbbbbbbbcddeeeeeeeeeeeedcbbbbbbbbbbbbbbbb



InlineSkate data를 받아서 timeseries는 PAA를 이용하고 Euclidian distance와 dtw를 이용하는 실습이었습니다. 코드는 전에 썼던 Euclidian distance와 dtw의 코드를 참고하였습니다.



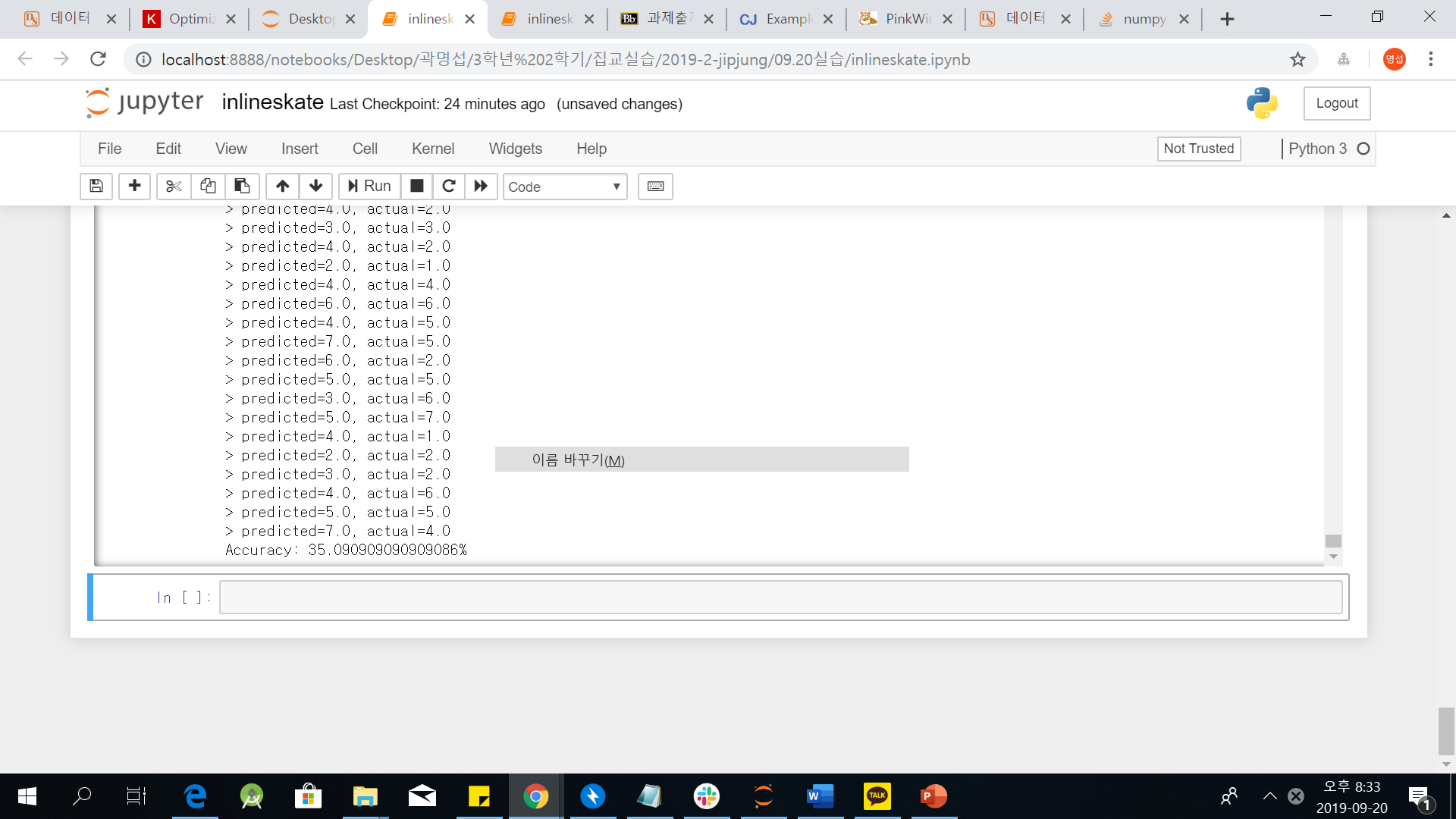
코드를 보시면 n이 몇 개씩 묶는 건지를 나타내는 것이고 result라는 list에 나눠준 데이터를 추가해준다



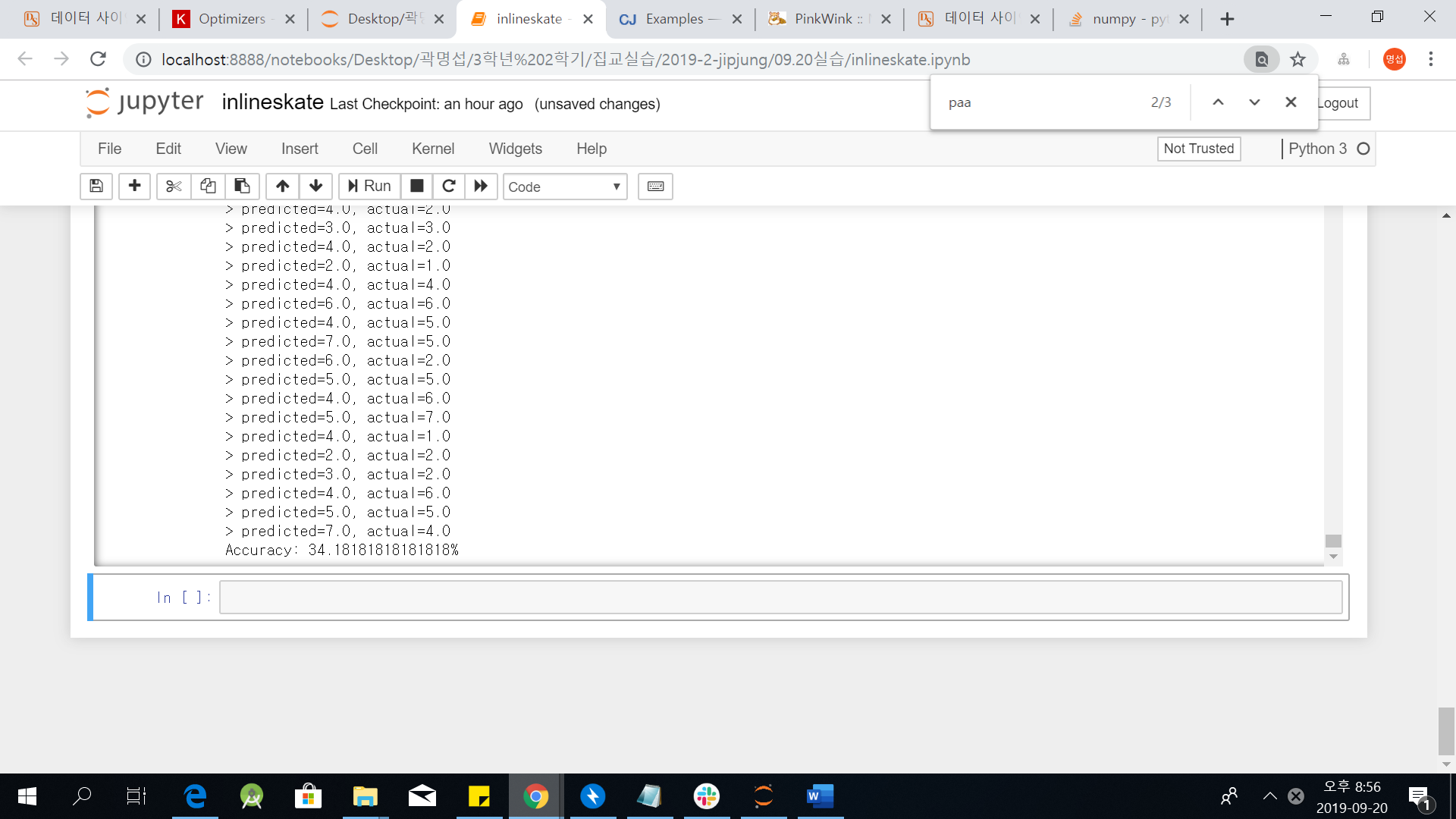
Euclidean distance는 그냥 저번에 사용했던 코드를 그대로 사용하였습니다.

dtw는 저번에 사용했던 코드를 사용하면 for문 때문에 너무 실행시간이 오래 걸리는 관계로 module을 이용해서 하였다.

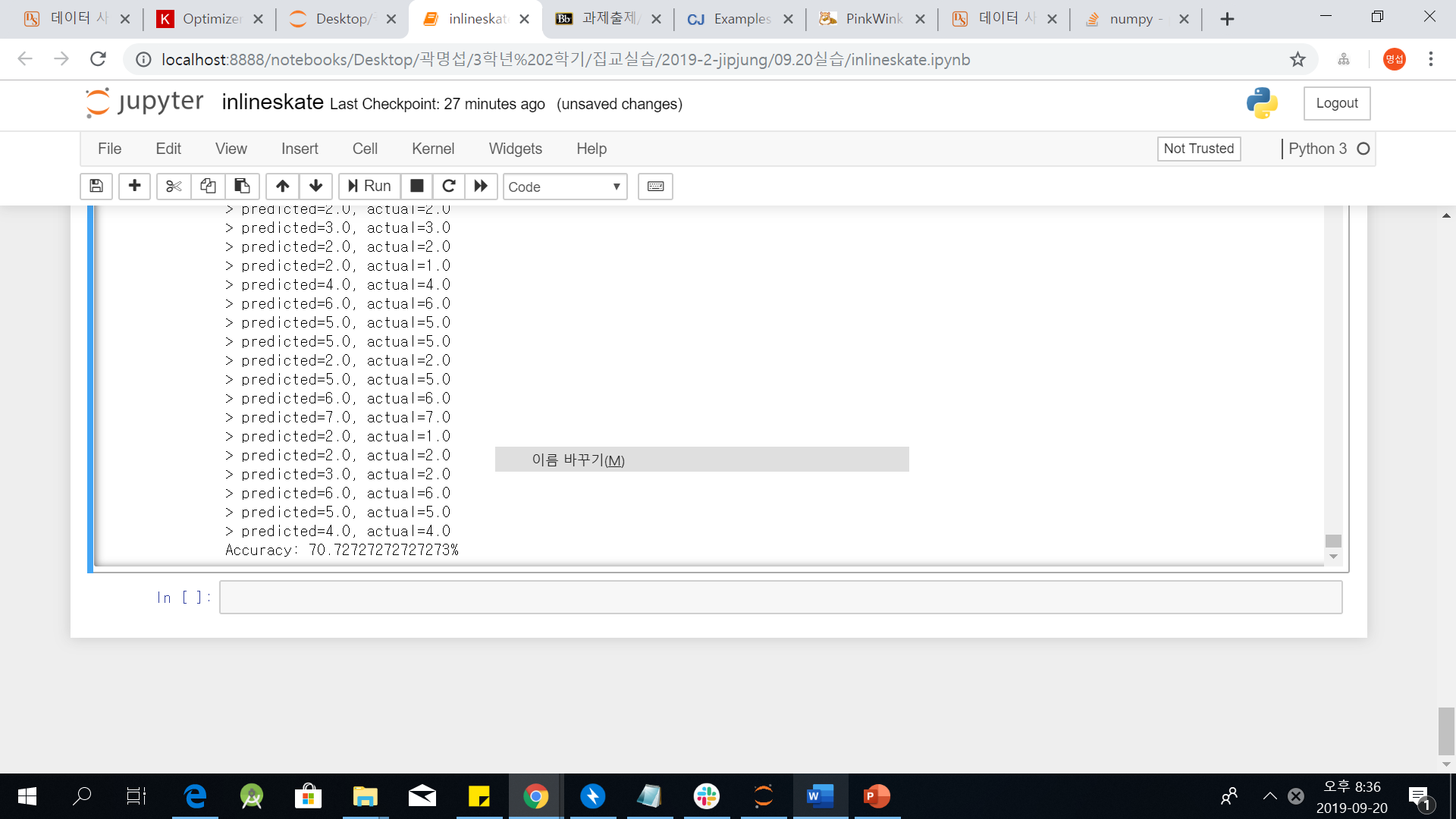
그래서 Euclidean distance를 사용해서 나온 accuracy는 35퍼이다. (aggregation size=10일경우)



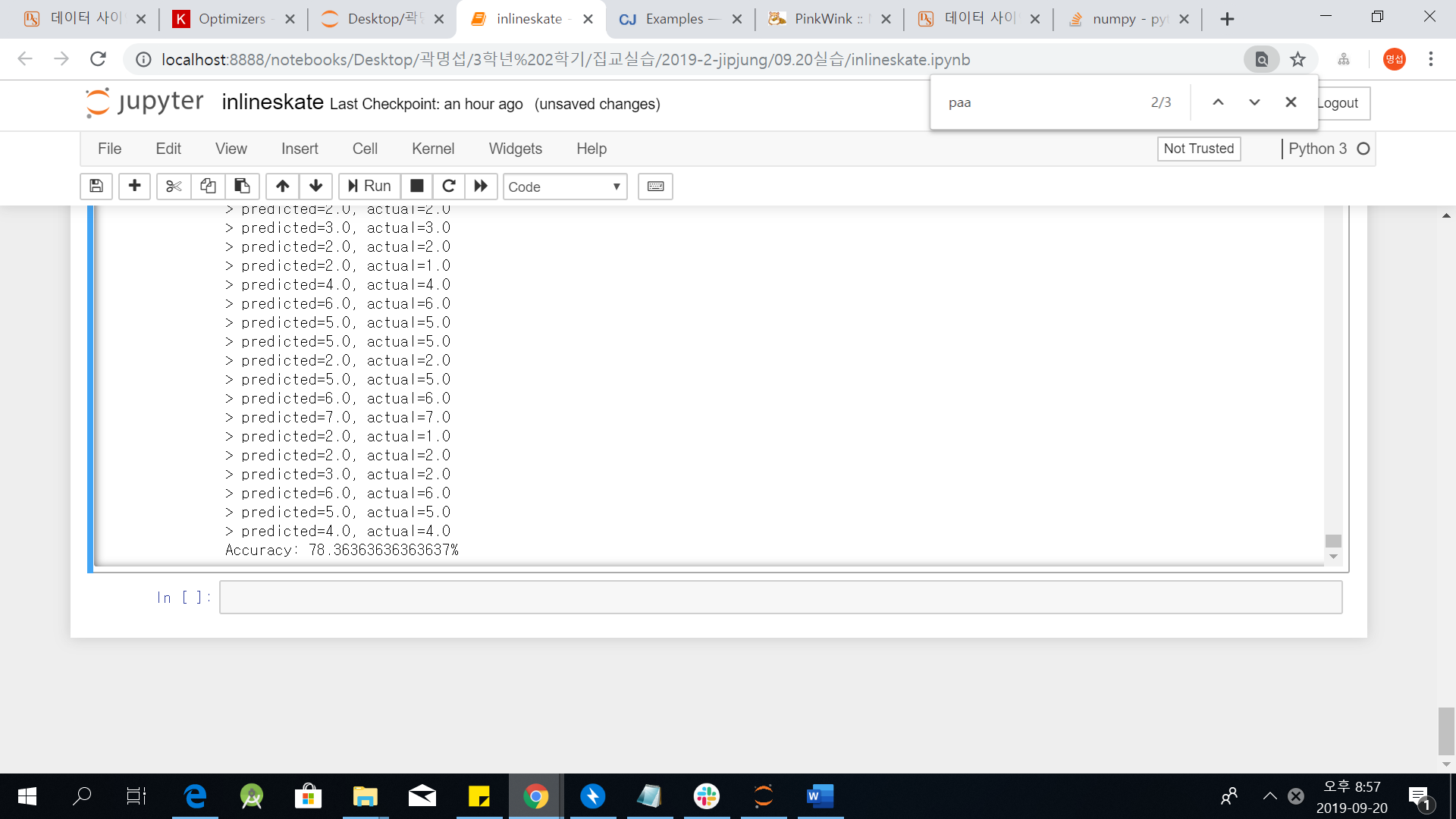
Euclidean distance를 사용해서 aggregation size=20일 경우에는 34퍼이다



그 다음 dtw를 이용해서 aggregation size = 10일 경우에 나타낸 accuracy이며 약 70%정도가 나왔다.



그 다음 dtw를 이용해서 aggregation size = 20일 경우에 나타낸 accuracy이며 약 78%정도가 나왔다.



Dtw를 사용하는경우에 aggregation size를 높였는데도 왜 인지 모르게 accuracy가 올라갔다. 사실 머리로는 이해가 되지 않았다.