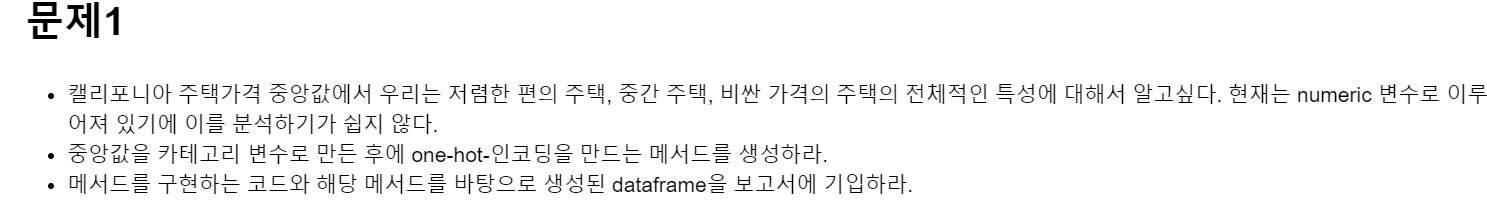
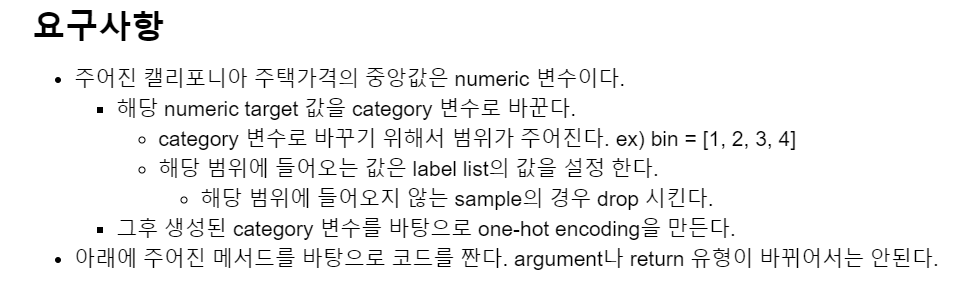
# 데이터 마이닝

## assignment

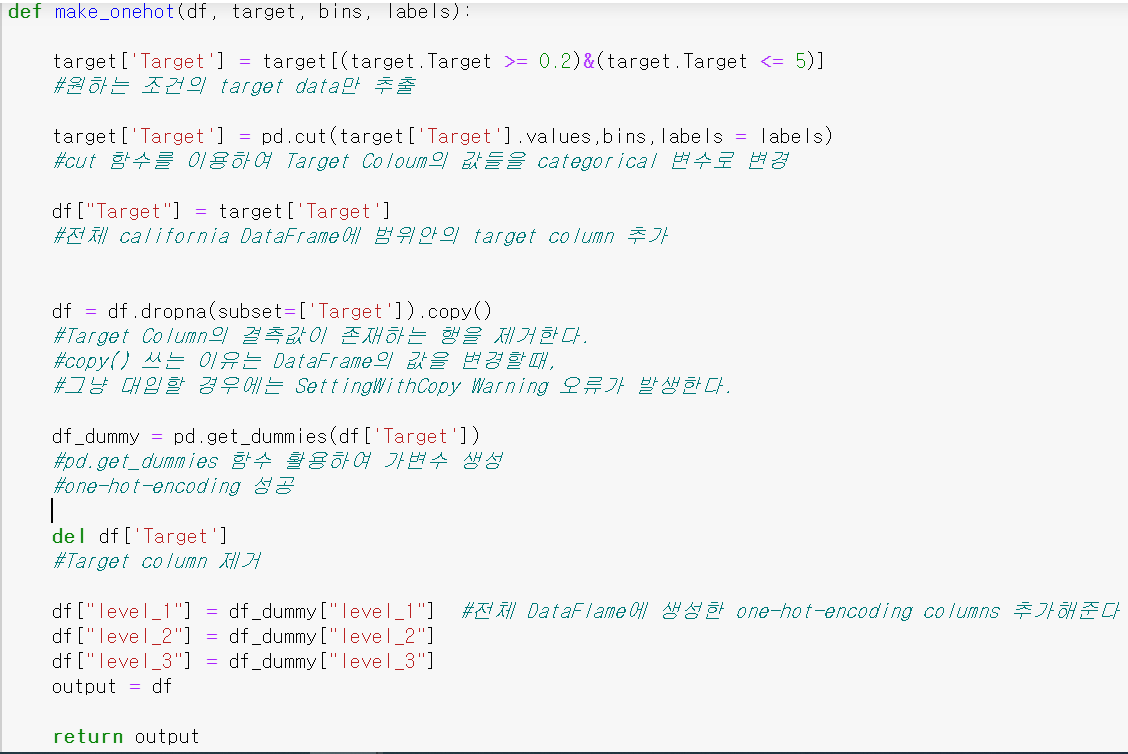
사이버보안학과

201620641 유상정



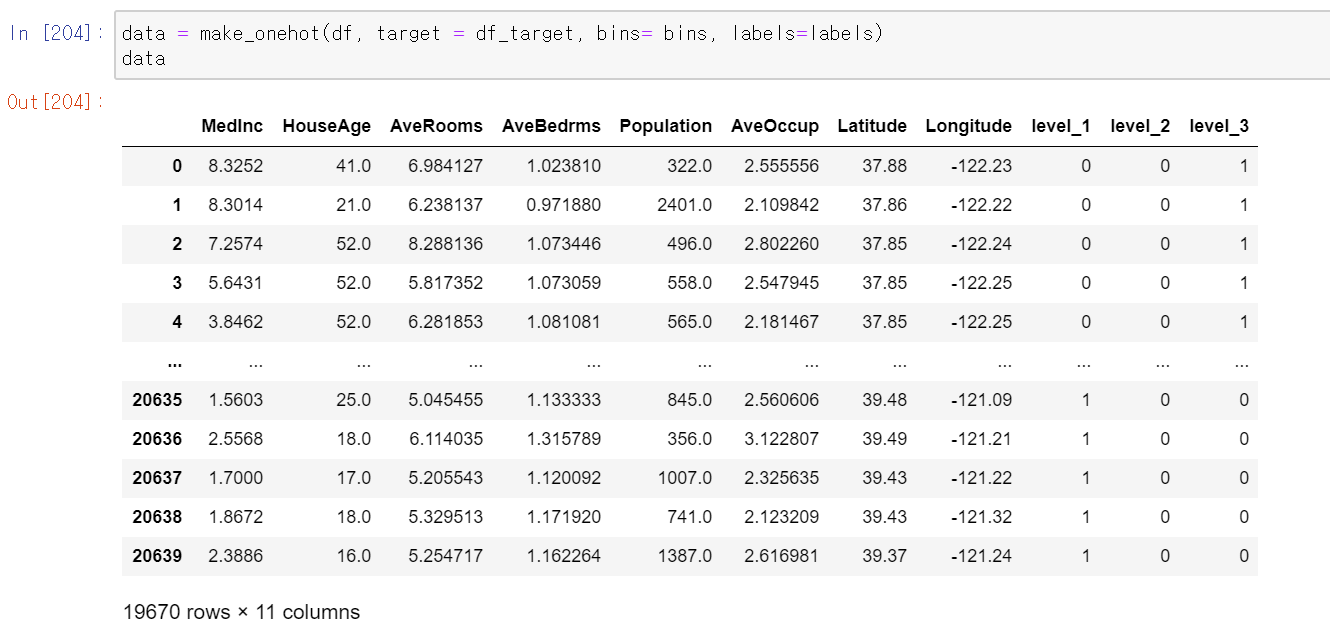


(1번문제 코드)

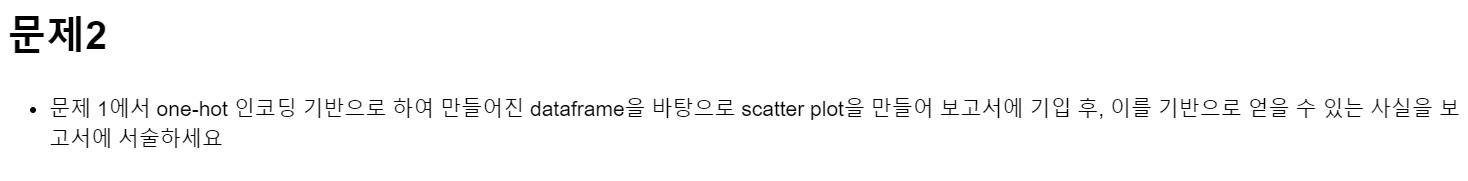


코드를 위와 같이 작성하였다. 각 코드 한 줄, 한 줄 마다 주석처리로 설명을 적어 두었다.

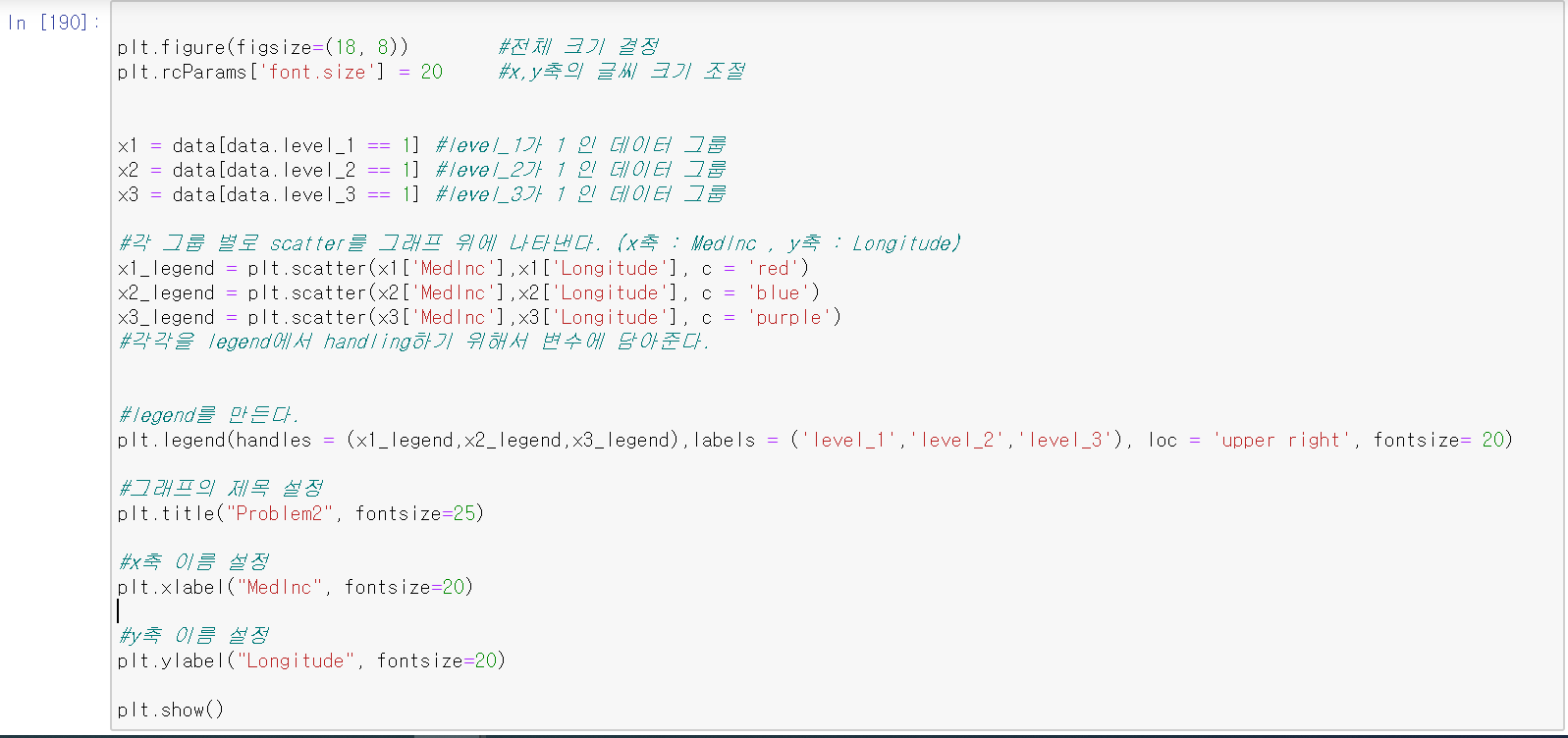
(결과)



함수 실행결과인 make\_onehot 함수를 통해 생성된 DataFrame이 예시로 나타난 DataFrame과 같음을 사진으로 알 수 있다. 또한 bins 범위 밖의 값들은 예외처리가 되어서, Data (raw)의 총 개수가 20640개에서 19670개로 감소하였고, one\_hot\_encoding 처리를 하여, column의 수가 11개로 증가 하였다.



(작성 코드)

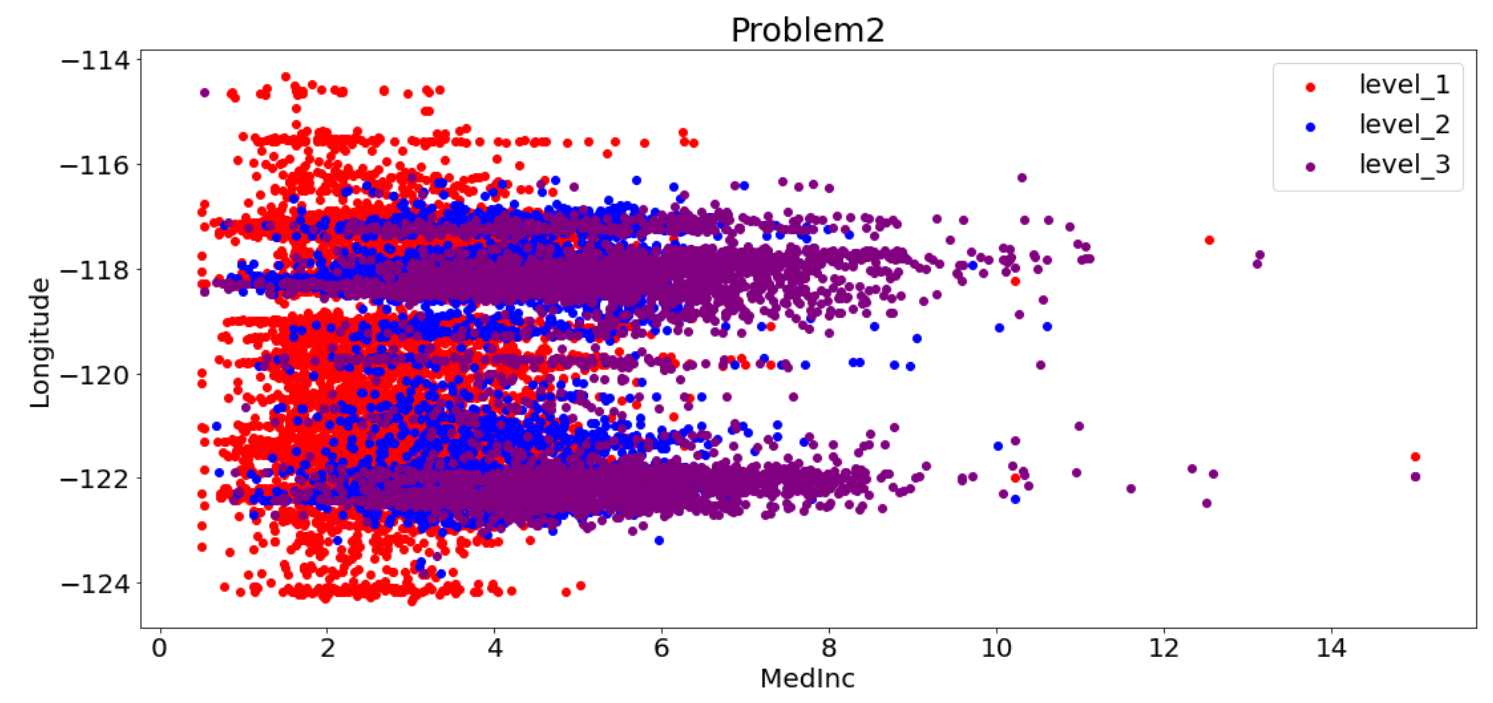


위와 같이 코드를 작성하였고, 각 코드의 한 줄, 한 줄 마다 코드에 대한 설명을 주석처리 해 두었다.



위 코드 이전에, 그래프를 나타내어 주는 matplotlib.pyplot 이 import 되어 있다.

(실행결과)



위의 scatter plot을 만들면서, 처음에는 어려움을 많이 느꼈다. 왜냐하면, jupyter 자체에 대한 지식이 전무할 뿐만 아니라, matplotlib 과 pandas를 처음 다뤄보았다. 이를 기반으로 얻을 수 있는 사실을 말하기 이 전에, python은 정말 편리한 도구임을 깨 달았다. 다음으로 위의 scatter plot으로 얻을 수 있는 사실을 알아보겠다.

￥이를 통해 얻을 수 있는 사실

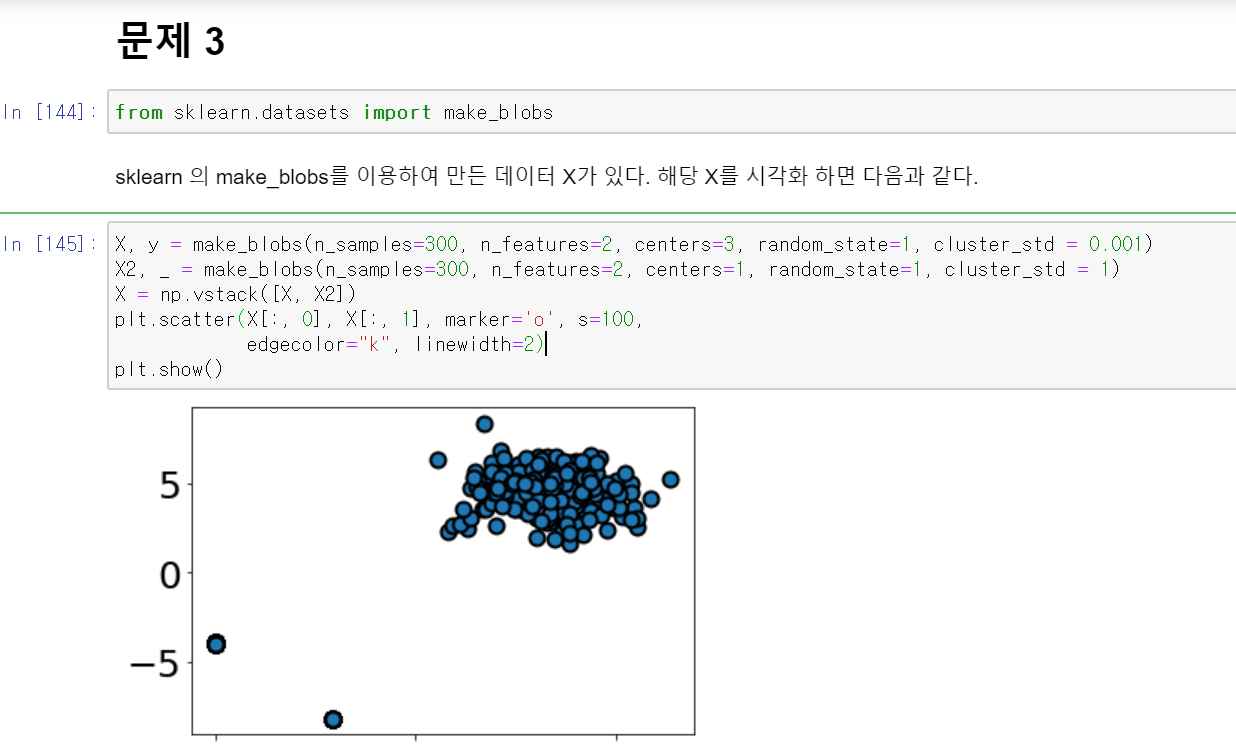


* data를 통해서 target 변수는 캘리포니아 지역들에 대한 집값의 중앙값임을 알 수 있다.

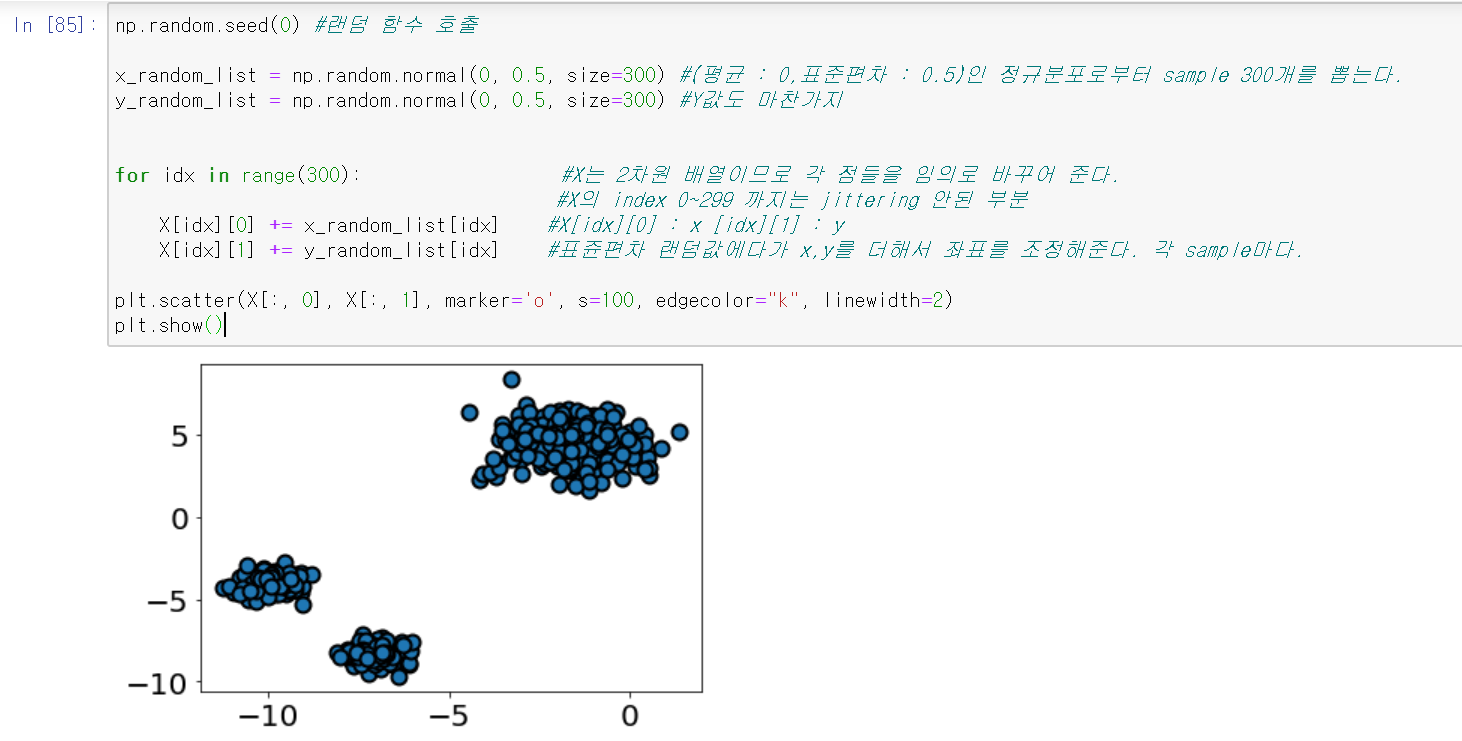
1. MedInc가 높게(12를 넘는 정도) 분포한 지역이 거의 없다. (outlier는 존재한다.)
2. Logitude(해당 행정 구역의 경도)와 상관없이 MedInc(행정 구역 내 소득의 중앙값)이 낮은 data가 넓게 분포해 있다.
3. Level1 , level2, level3인 data가 많이 겹쳐져 있지만, 경도가 중앙에서 멀어질수록 level1이 분포한다.
4. MedInc(행정 구역 내 소득의 중앙값)이 증가할수록 상대적인 집값의 중앙값이 높아진다. 물론, x축이 4쯤될때는 모든 level이 존재한다. 이 부분에 대한 조사는, 조금 더 할 필요가 있다. 또한, MedInc(행정 구역 내 소득의 중앙값)이 높은데, level1인 지역도 존재하지만, outlier로 생각하고 제외해야 한다고 생각한다.
5. 최종적으로, 위 그래프에서 얻을 수 있는 결론

* 대부분의 경우 MedInc(행정 구역 내 소득의 중앙값)이 증가할수록 행정구역 내 집값의 중앙값이 증가한다.

문제 3번 아래 장



(작성 코드 및 결과)



우선 코드 설명은 각 한 줄, 한 줄마다 주석처리로 설명해 뒀다. 또한 결과를 보면 , 문제 예시에서 나온 scatter plot과 유사한 모양이 완성되었음을 알 수 있다. 본인은 jittering을 만들기 위해서 정규분포로부터 랜덤 값을 뽑아내는 numpy.random.normal 함수를 사용하였다.

￥Jittering이 필요한 경우

Jittering은 데이터에다가 일부로 조금의 noise를 줌으로서 한 곳에 뭉쳐 있지않고, 흩어져 있게 만드는 것이다. 이 것이 필요한 경우로는 데이터형이 카테고리 형일 때, 그 카테고리에 포함된 데이터들은 한 점에 똑같이 위치하게 된다. 그래서 data 시각화를 하였을 때, 실제의 데이터는 굉장히 많지만, 한 점으로 보여서 데이터가 얼마나 많은 지 쉽게 판단할 수 없다.

이러한 경우에 jittering을 줌으로서, 한점에 똑같이 있던 데이터 하나하나가 조금씩 그 위치에서 벗어나게 된다. 그렇게 되면 데이터는 한 곳에 뭉쳐 있는 것이 시각적으로 잘 보이게 된다. 즉, 시각적인 효과를 높여 줌으로서, data분석가의 분석을 좀 더 용이하게 한다. 예시로는 3번문제의 코드 결과가 되겠다. 또 다른 예를 하나 들자면,

Bins = [1,6,10] , labels = [a,b]

Data = [1,2,3,4,5,7,8,9]

Datas = pd.cut(Data, bins = Bins, labels = labels)

[1,2,3,4,5] 🡪 a카테고리 , [7,8,9] -> b 카테고리

이렇게 되었을 때, 시각화를 카테고리 별로 처리하면 그래프에는 두 점 (a,b) 밖에 나타나지 않는다. 이럴 때, 각 좌표를 일부로 jittering하여서 풍부한 데이터가 눈에 보이도록 한다.