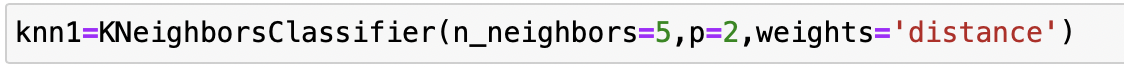
2019150445 신백록 4장 과제

1.

일단 다음과 같이 k=5로 설정하고 weight=’distance’로 준 것과 weight=’uniform’을 준 knn과 결과를 비교하였더니 둘 다 97.8%의 test set에 대한 acc로 똑같은 성능을 보였다.



하지만 k를 늘려보니 weight=’distance’를 준 model이 uniform을 준 모델보다 확실히 성능이 좋게 나타났다. K=40일 때, weight=’distance’를 준 model의 acc는 97.8%로 K=5일 때와 같은 성능을 보였지만, weight=’uniform’을 준 모델은 88.8%로 성능이 현저히 떨어졌다.

텍스트이(가) 표시된 사진

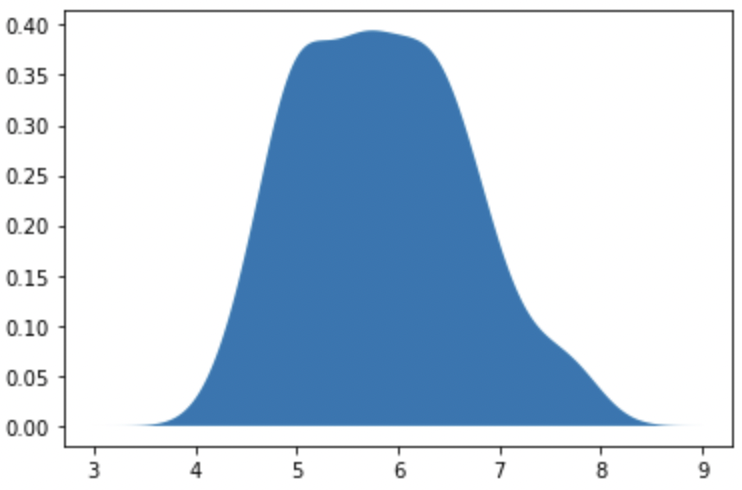
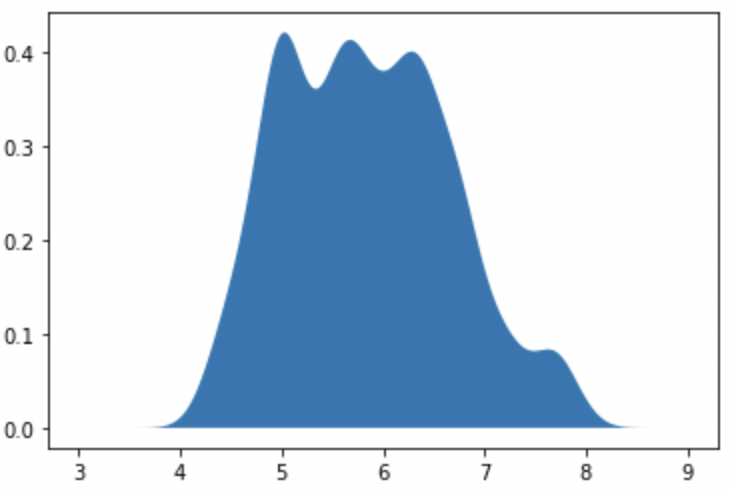
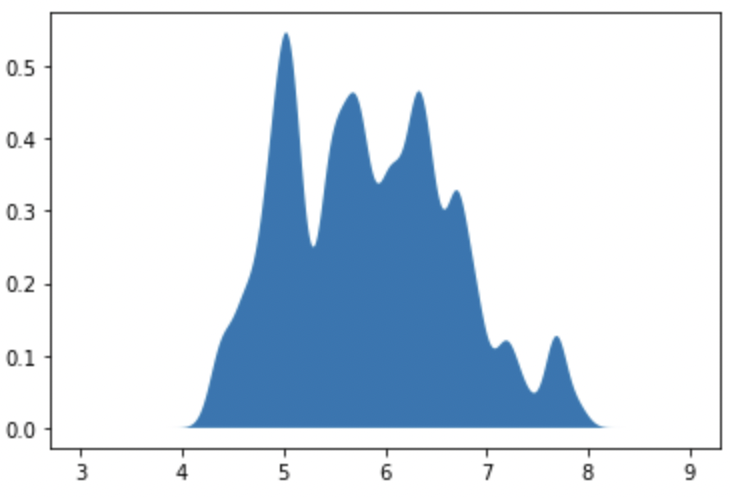
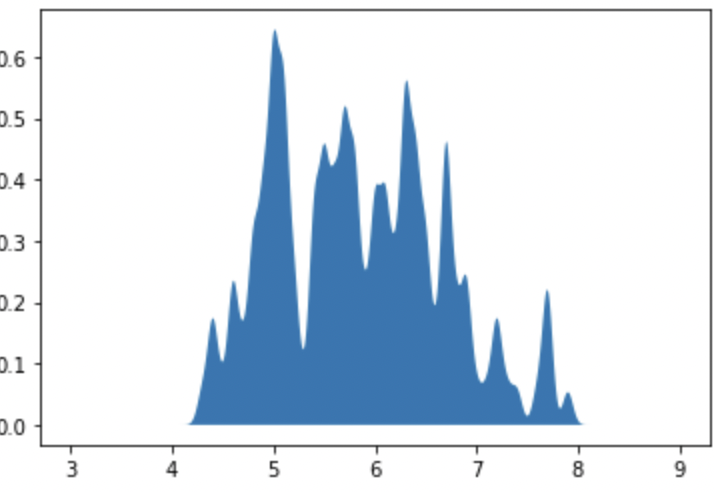
자동 생성된 설명

k를 크게 잡으면 잡을 수록 실제 class의 바운더리를 넘어가 다른 class의 instance까지 포함하기 때문에 weight을 똑같이 주면 성능이 낮아질 수밖에 없다. 하지만 weight을 거리에 가까울 수록 더 크게 주게 된다면 해당 클래스의 feature을 나타내지 못하는 instance가 포햠되더라도 weight이 낮기 때문에 weight=’uniform’을 준 모델보다 제대로 예측할 수 있을 것이다. 또한 weight=’distance’로 주면 데이터가 sparse해질수록 큰 역할을 할 수 있을 것이다. 낮은 K를 주더라도 데이터가 sparse하다면 멀리 있는 instance까지 neighbor에 포함될 수 있기 때문이다.

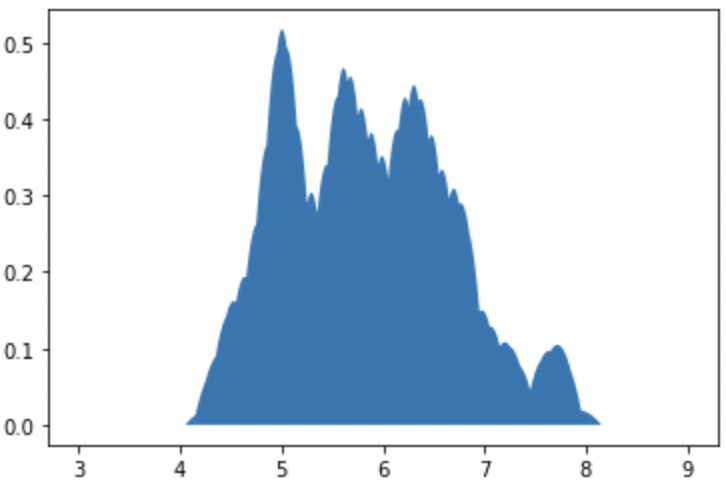
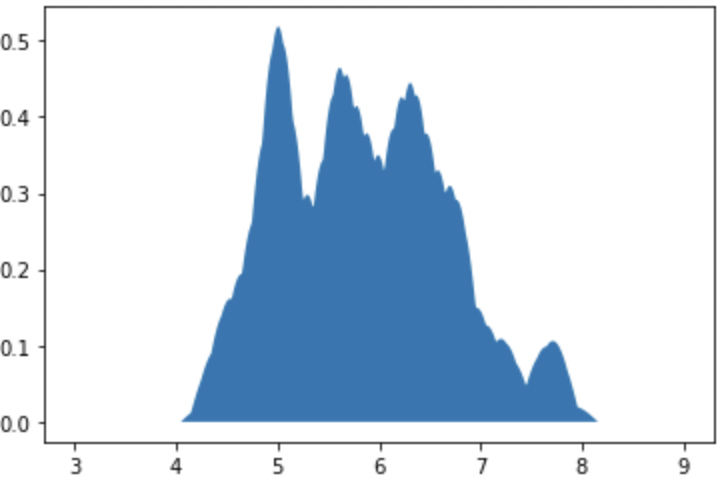
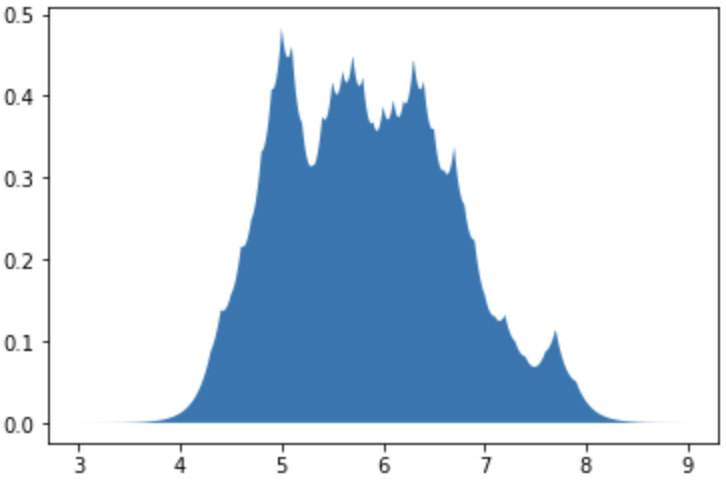
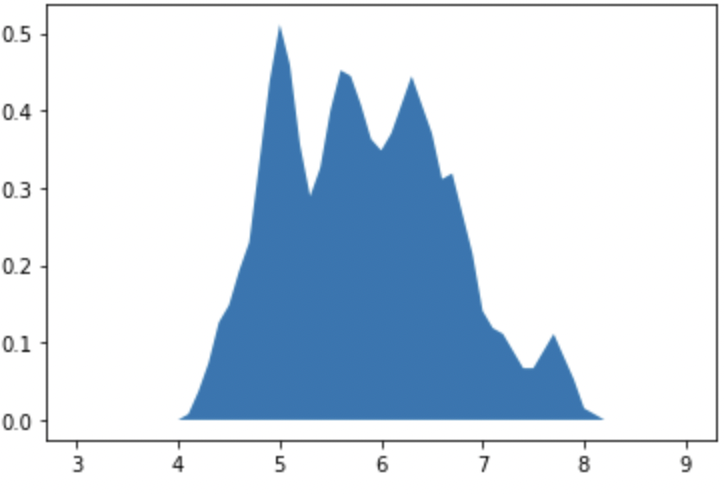
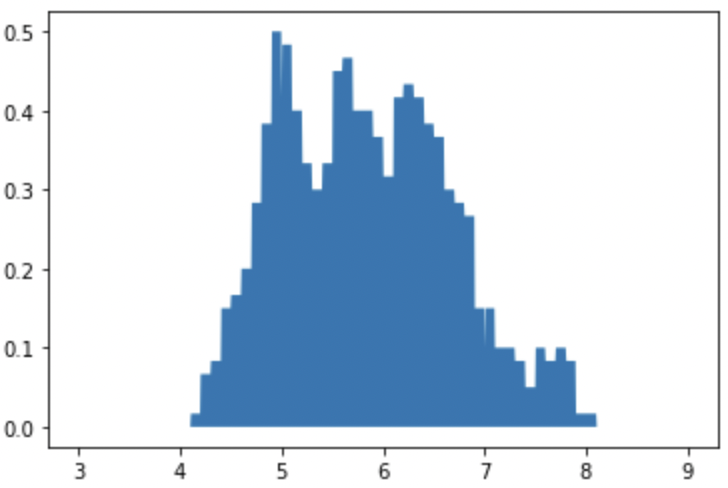
특히 거리를 계산하지 않는다는 것은 거리에 대한 정보를 아는데도 uniform을 사용함으로써 weight을 1 또는 0을 줘 정보의 손실을 가져오는 것과 같다고 생각한다.

2.

Kernel의 bandwidth를 조정하였다. 왼쪽부터 차례대로 0.05, 0.1, 0.2, 0.3이다. Bandwidth가 커지면 smoothing이 강해져 분포에 대한 정보까지 뭉개졌고, bandwidth가 작아지면 너무 flexible해져 데이터 하나 하나에 대한 커널함수가 그대로 드러나 정보를 나타내기 때문에 데이터의 분포에 대한 정보를 얻기 힘들다.



아래는 kernel 함수를 조절해가며 결과를 비교한 것이다. 왼쪽부터 ‘tophat’, ‘linear’, ‘exponential’, ‘cosine’, ‘epanechnikov’의 그래프이다. 각각의 bandwidth는 임의로 데이터가 가장 잘 나타내는 방향으로 조정하였다.



일단 tophat 함수도 거리에 따라서 가중치를 부여하는데 그 가중치가 1또는 0이다. 따라서bandwidth가 작은 히스토그램과 비슷한 모양을 갖고 있고, uniform 함수의 모양 떄문에 부드럽게 smoothing 되기가 어렵다.

Linear 함수와 exponential 함수도 마찬가지로 함수의 모양 때문에 뾰족한 부분이 그대로 드러날 수 밖에 없지만 bandwidth를 늘리면 부드러운 smoothing이 어느정도 가능해지기는 한다. 하지만 그만큼의 분포에 대한 정보 손실이 생기게 된다. 이는 거리에 대한 가중치가 점점 감소한다는 면에서 tophat 함수와는 다르고 그나마 커널함수에 다른 데이터들의 정보가 어느정도 포함되어있다는 면에서 합쳤을 때 전체 분포를 찾기에 용이하다.

cosine과 epanechnikov, gaussian 함수는 exponential에 비해 함수가 천천히 감소한다는 면에서 특정 데이터와 다른 데이터와의 거리가 가까울수록 특정 데이터의 커널함수에 그 정보가 많이 포함된다. 그래서 좀 더 통계적으로 합리적인 그래프가 만들어진다.

위 그래프에서만 그런건지는 몰라도, bandwidth를 조절하면 6개의 그래프가 거의 비슷한 형태를 보여 데이터의 분포를 잘 나타내주고 있다. kernel을 조절하는 것보다 bandwidth를 데이터에 맞게 잘 조절하는 것이 더 중요해 보인다.