**인공지능확률론 homework 2**

12195304

컴퓨터공학과

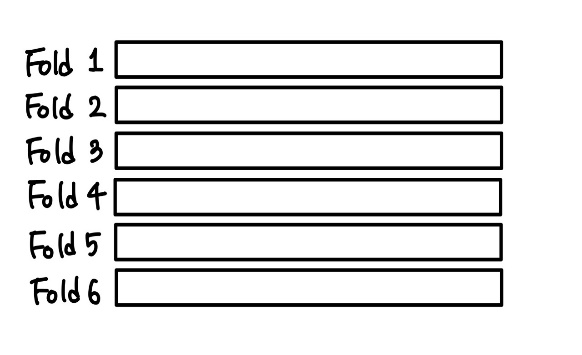
이지윤

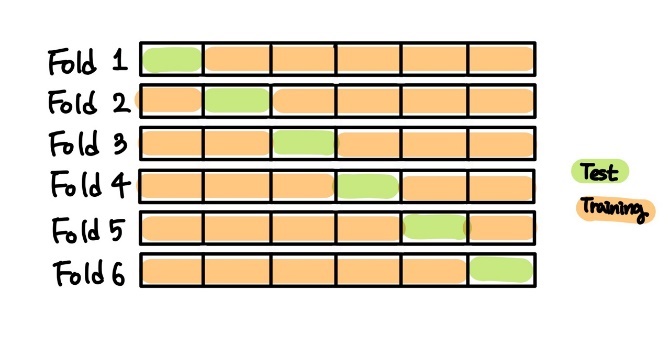
**코드실행은 Jupyter notebook을 사용하여 작성했습니다.**

**A.**

**Cross Validation**

k개의 fold를 만들어서 진행하는 교차검증을 k fold cross validation이라고 한다.

Training set과 test set으로 데이터를 나누고 training을 k개의 fold로 나눈다. 만약 data를 6개의 fold로 나누고 싶다면 왼쪽 그림처럼 하면 된다. 즉, fold하나에 전체 data가 들어가고 총 6개의 fold에 검증하고 싶은 데이터의 전체 데이터가 각각 들어간다.



6개의 fold로 나누었으면 데이터를 6개로 분리하여 1개는 test set으로, 나머지 k-1개는 training set으로 한다. 각 fold의 data set의 역할을 바꿔가면서 6개의 fold를 검증할 수 있게 한다.

Fold1에 있는 training set으로 평균치를 내고 test set으로 평균치와 test set을 검증하여 error값을 구한다. Fold2, 3, 4, 5, 6도 동일하게 진행한다. 그 후 6개의 error값을 바탕으로 6개의 fold중 가장 최적의 모델을 찾고 그 모델로 전체 training set의 학습을 진행시킨다. 마지막으로 해당 모델을 처음에 분할하였던 test set으로 evaluation한다.

**B. Iris classification using text data**

**1.**

(코드 일부 발췌)

seto\_u=slen\_seto.mean() # setosa의 feature중에 sepal\_length의 평균을 구하고

seto\_sigma=math.sqrt(slen\_seto.var()) # sepal\_length의 표준편차를 구하여

seto\_pre=seto\_u-seto\_sigma

seto\_back=seto\_u+seto\_sigma

seto\_cnt=0

for i in slen\_seto['s\_len']: # 해당 범위외의 값이 있다면 +1 하며 cost를 늘린다.

if (i<seto\_pre).bool():

seto\_cnt+=1

elif (i>seto\_back).bool():

seto\_cnt+=1

**2**.

(코드 일부 발췌)

tmp\_x = abs(seto\_ux-ver\_ux)/2 # 두 개의 class에 대한 mean x값의 차를 이용하여 중간값을 구하고 그 값을 yellow\_x에 넣는다. Yellow\_y도 동일하다.

tmp\_y = abs(seto\_uy-ver\_uy)/2

yellow\_x=0

if (seto\_ux<ver\_ux).bool():

yellow\_x = seto\_ux + tmp\_x

else:

yellow\_x = ver\_ux + tmp\_x

if (seto\_uy < ver\_uy).bool():

yellow\_y = seto\_uy + tmp\_y

else:

yellow\_y = ver\_uy +tmp\_y

slope = (yellow\_y['s\_wid'])/(yellow\_x['s\_len']) # 직선의 기울기를 각 mean의 중간값을 이용하여 구한다.

def linear\_red\_green(x):

return slope\*x+0.1 # 함수로 구현하여 cost 구할 수 있게끔 하였다.

list\_x=[]

list\_y=[]

for i in np.arange(4.0,8.0,0.2):

list\_x.append(i)

list\_y.append(linear\_red\_green(i))

산점도를 이용한 그래프에서 x축이 petal length, y축이 petal width 일 때, cost가 가장 적어 분류

하기에 가장 적절하다고 생각했다.