**Applied Probability for Computer Science homework2**

**12171628 방솔찬**

Part A: k-fold cross-validation method가 무엇인가?

머신러닝은 dataset를 크게 3가지로 분류한다. training, validation, test Dataset으로 나눠진다. 이 3가지는 마치 고등학교 공부에 비교할 수 있는데 우리는 대학을 가기 위해서 공부를 하고 모의고사를 치고 수능을 본다. 공부를 하면서 우리는 개념을 이해 할려고 노력하고 공부가 끝나면 수능을 치기 전 모의고사를 봄으로써 내가 몇점을 맞을지 모의시험을 본다. 그러나 모의시험에서 잘 봤다고 수능을 잘본다는 보장은 없다. 그래서 모의고사를 많이 풀어보고 수능을 보게 된다. 이 예에서 공부는 training 모의고사는 validation 수능은 test dataset이라고 할 수 있다. data가 충분할 만큼 주어지면 3가지로 나눠서 쓰면 된다. 그러나 dataset이 부족할 경우에 이 방법을 쓰는데 validation을 삭제하고 training과 test dataset 2가지로 나눠진다. validation의 목적은 training dataset을 학습해서 나온 모형이 test상황에서 얼마나 잘 작동하는지를 알고자 하는 것이지만 사용을 안 할 경우에는 test dataset의 결과를 좋게 하는 방향으로 만들기 떄문에 특정 상황에 overfiting될 수 있다. 그러나 오히려 validation dataset과 test dataset의 결과가 비슷하다면 우리는 그 모형에 대해 좀더 신뢰할 수 있을 것이다.

그러면 testdata가 부족할 떄 training과 test data를 나누는 방법을 무엇일까?

그것은 k에 따라 달렸다. 일단 data를 k개의 fold로 나눈다. 1개의 test data와 k-1개의 taring data로 나눌 떄 k번 다르게 선택할 수 있다. 그로 인해서 k번의 traing을 통해서 모델을 만들 수 있다.

Part B-1 Iris classification using text data

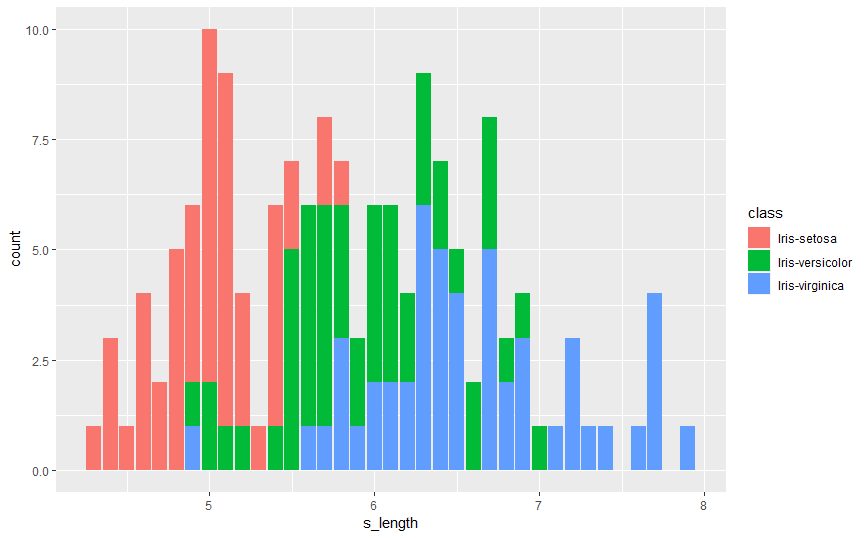
저는 R언어를 사용해서 그래프를 그렸습니다.

library(ggplot2)

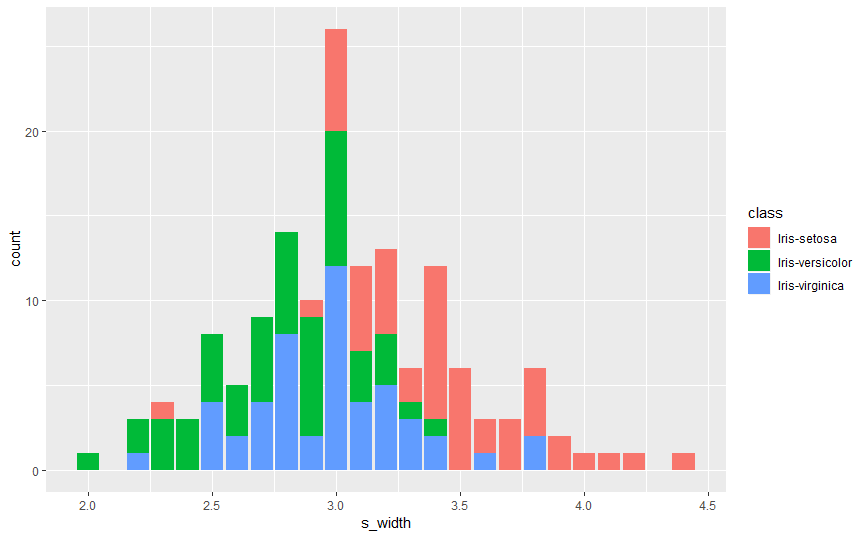
library(readxl)

library("dplyr")

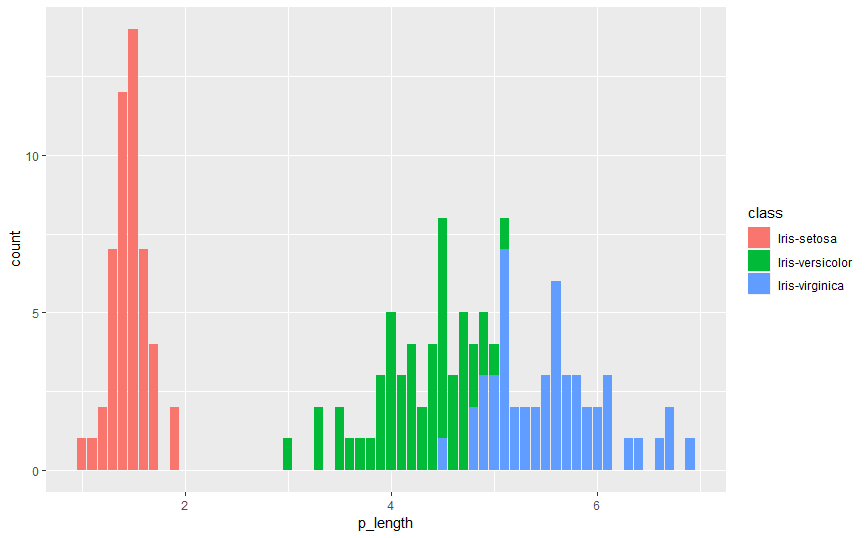
df\_csv<-read.csv("csv\_all.csv",stringsAsFactors = F)



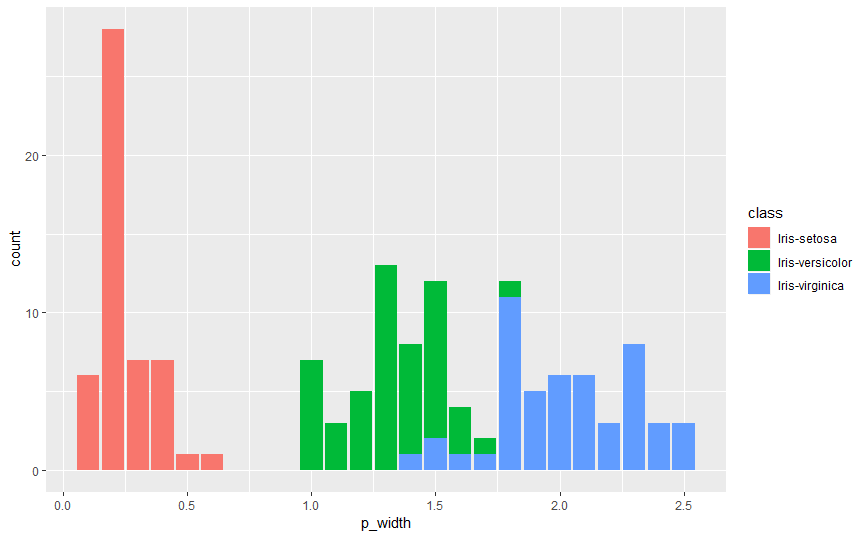
#problem 1-1 ggplot(data=df\_csv,aes(x=s\_length,fill=class))+geom\_bar()



#problem 1-2 ggplot(data=df\_csv,aes(x=s\_width,fill=class))+geom\_bar()



#problem 1-3 ggplot(data=df\_csv,aes(x=p\_length,fill=class))+geom\_bar()



#problem 1-4 ggplot(data=df\_csv,aes(x=p\_width,fill=class))+geom\_bar()

1. 전처리:

그래프를 R에서 그리고 사용한 data를 all.txt파일로 만들고 파이썬환경에 불러왔습니다. 이것을 kfold\_training과 kfold\_test로 나눴습니다.그리고 각각 값을 float로 변환을 시켰습니다. 2차원 배열로 만들었는데 첫번쨰 파라미터에는 data의 순서가 들어가고 두번쨰 파라미터에는

# [0]:s\_length [1]:s\_width [2]:p\_length [3]p\_width [4]:class

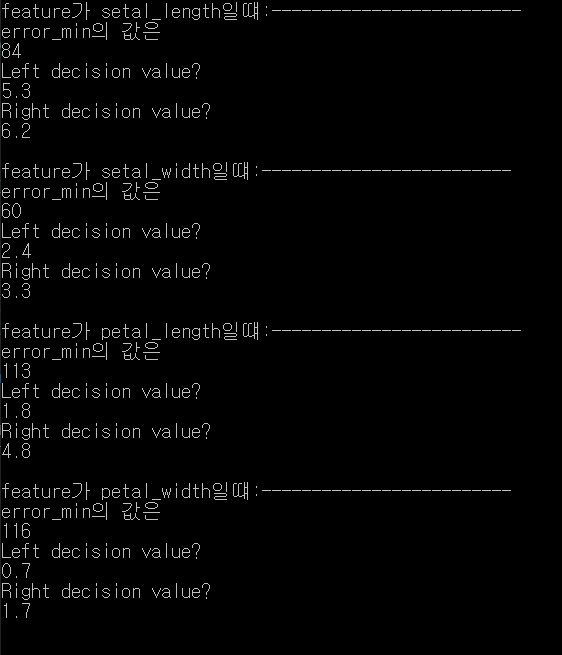
각각 해당하는 값이 들어갔습니다.

B-1번

4개의 특징과 3개의 class가 존재할 떄 우리는 3개의 class의 각각 다른 점을 보여야 한다.원하는 구간에 원하는 class가 있기 위해서 cost를 줄이고 optimal decision boundary를 설정해야 한다. 그래프에서 선2개를 그으면 3개의 구간이 나타난다. 그 구간안에 원하는 클래스가 최대한 많이 있으면 된다고 판단했습니다. 각각 class별로 data들의 평균을 구해서 어떤 class가 이 feature에서 대체로 어느부분에 존재하는지를 파악했습니다. 선 2개를 data의 처음과 끝으로 전체를 0.1단위로 돌리기 위해서 data의 제일 작은 값과 제일 큰 값을 구해야 했습니다.

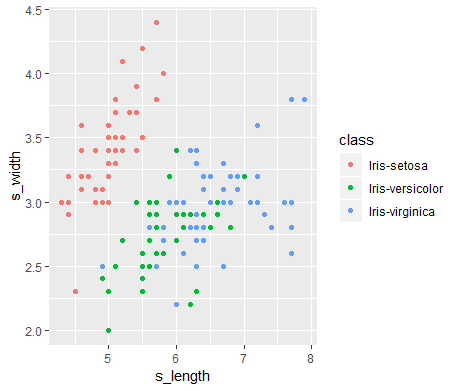
return\_max\_min\_value(parameter):파라미터의 data중 제일 큰값과 작은값리턴

return\_order\_str(parameter):데이터의 평균을 통해 class의 순서파악



각 feature별로 에러가 최소화되는 boundary일떄 내가 예측한class 와 실제 class가 같은 경우 count를 하여 error\_min으로 출력했습니다. Left decision과 right decision은 각각 class를 나누는 decision boundary이다.

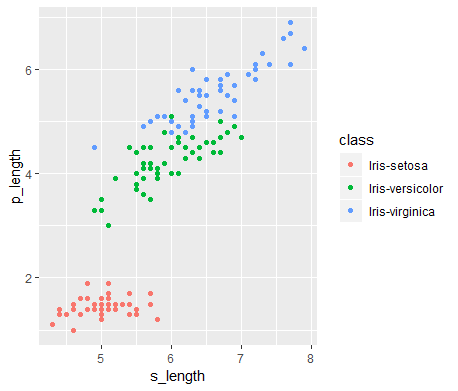
Part B-2 Iris classification using text data



#problem 2-1

ggplot(data=df\_csv,aes(x=s\_length,y=s\_width,col=class, fill=class))+

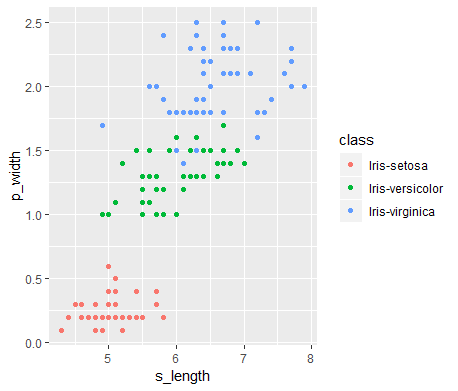
geom\_point()



#problem 2-2

ggplot(data=df\_csv,aes(x=s\_length,y=p\_length,col=class, fill=class))+

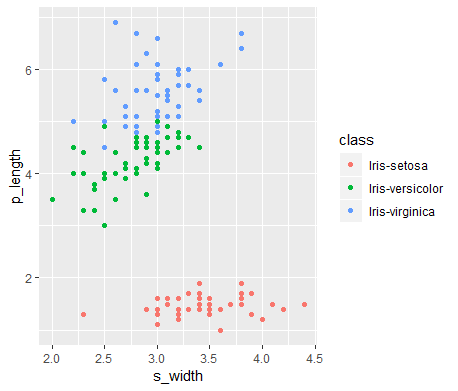
geom\_point()



#problem 2-3

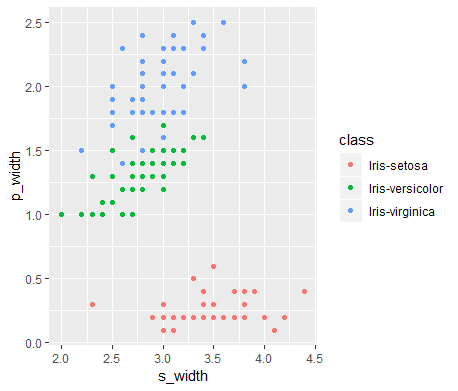
ggplot(data=df\_csv,aes(x=s\_length,y=p\_width,col=class, fill=class))+

geom\_point()



#problem 2-4

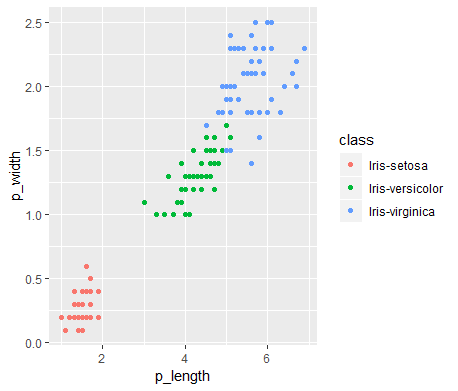
ggplot(data=df\_csv,aes(x=s\_width,y=p\_length,col=class, fill=class))+geom\_point()



#problem 2-5

ggplot(data=df\_csv,aes(x=s\_width,y=p\_width,col=class, fill=class))+

geom\_point()



#problem 2-6

ggplot(data=df\_csv,aes(x=p\_length,y=p\_width,col=class, fill=class))+

geom\_point()

로지스틱 분류 방법을 이용하였습니다.

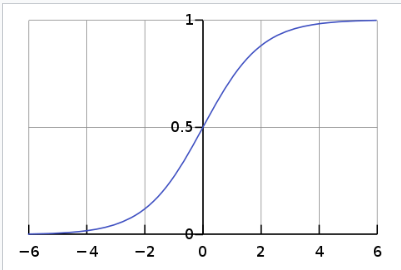
Logistic Regression Model에서 2가지 클래스중 하나로 분류하기 위해서

If h(x)>=0, predict “y=1”

If h(x)<0.5, predict “y=0”으로 classification한다고 할 때

0<=h(x)<=1을 원합니다. 그래서 우리는 h(x)에 sigmoid 함수를 취해줍니다.

시그모이드 함수특성은 아무리 작아도 0이상이고 아무리 커도 1이하이기 떄문입니다. 이 함수의 특성을 활용해서 로지스틱 분류를 한다.



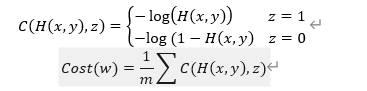
가설함수를 다음과 같이 설정했습니다.

H(x,y)=Sigmoid(1+w0\*x+W1\*y) 상수를 1로 설정하는 이유는 w0와 w1에 의해서 실질적으로 y=ax+b라 할떄 a와 b가 정해지기 떄문 입니다.

decision boundary에서 if h(x)>=0일떄 y=1로 예측하고

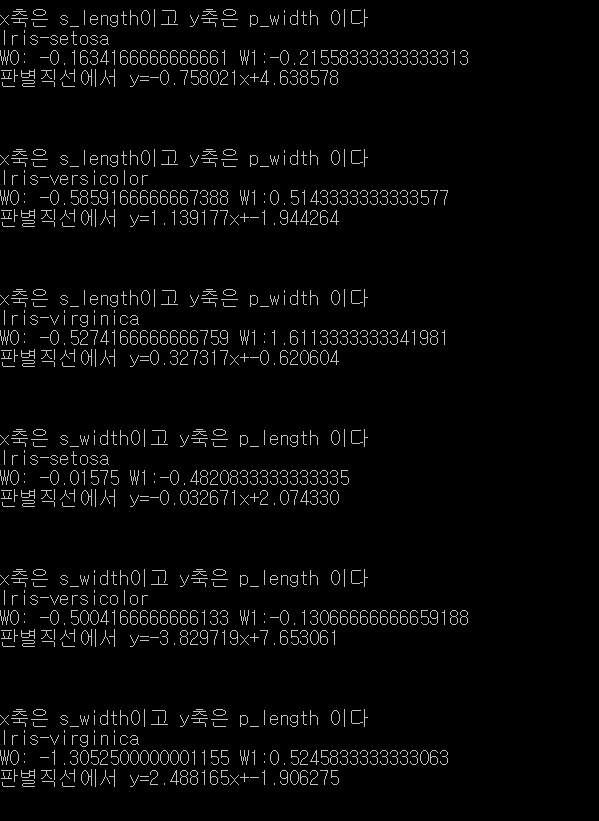
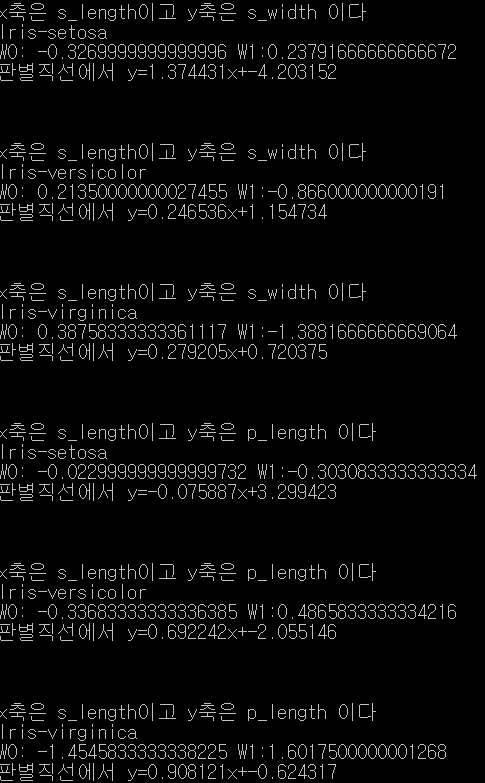
decision boundary에서 if h(x)<=0일떄 y=0로 예측합니다.

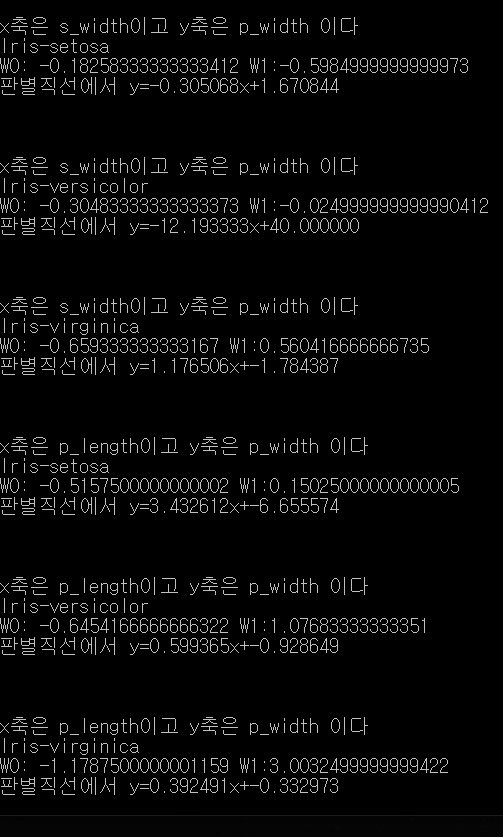
여기서 비용함수는



입니다. Cost를 최소화하는 x,y를 구해서

여기서 log를 취하는 이유는 경사하강법을 써서 cost를 최소화 해야 된다. log가 없는 cost함수는 h의 도함수가 0인 부분이 많아서 실제로 h(x)가 최소화되는 값이 구해지지 않는다. 그래서 log를 취함으로써 h의 도함수가 0인 부분이 하나가 되게 한다. 경사하강법은 W=W-a\*(Cost(W)의 도함수)에서 (Cost(W)의 도함수)를 구하기 위해서 1/m 시그마(H(x,y)-z)\*(x or y)이다. 이것을 저는 만번 돌리면서 W를 정의한다. 그래서 H(x)를 정한다. Multi class 에서 선을 2개를 그어야 되는데 iris-setosa와 나머지 class,iris-virginica와 나머지 class, iris-versicolor와 나머지 class로 나눠서 로지스틱 분류를 한다. 그러나 iris versicolor와 나머지 class를 나눌떄 거의 모든 feature가 setosa,versicolor,virginica순서대로 있어서 versicolor는 신뢰도가 높지않다. 그래서 setosa, virginica를 분류하는 선이 decision boundary이다.

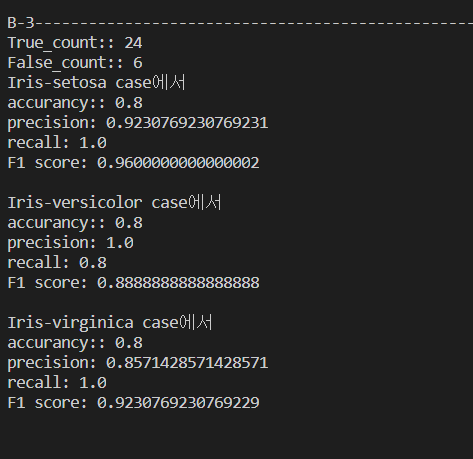




B-3

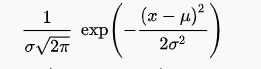
Bayesian decision Rule에서 featrue는 4개 class 는 3개이다. 확률 x를 class c라고 예측할 확률이라고 할 떄 f1,f2,f3,f4는 각각의 특징을 가질 확률이라고 하자.

Bayesian decision rule에 따라서 P(x|f1,f2,f3,f4)=P(f1,f2,f3,f3|x)\*P(x)/P(f1,f2,f3,f4)이다. Random 함수를 돌려서 5-fold -cross- validation을 하였을 떄 p(Xc0)=0.2 p(Xc1)=0.4 p(Xc2)=0.4가 나왔다. P(f1,f2,f3,f4)에서 모든 posterior값은 같기 떄문에 고려하지 않아도 된다. B-1번에서 그린 그래프에서 가로의 길이를 0.1단위로 나누고 해당 feature의 범위일떄 내가 구하고 싶은 class의 확률이 먼지 구하면 그것이 liklihood가 된다. 그러나 그범위에만 혹시 구하는 class 가 없을경우 posterior가 0이 된다. 이것은 옳지 않음으로 이럴 떄는 B-1번에서 구한 decision boundary와 비교하여 범위안에 있을 때는 확률을 1로 가정하고 그렇지 않을 경우에는 0.01로 가정해서 각각 class 별로의 posterior을 비교하여 classification하였다. 이 방법으로 예측시 정확도는 0.8 정도 나왔다.

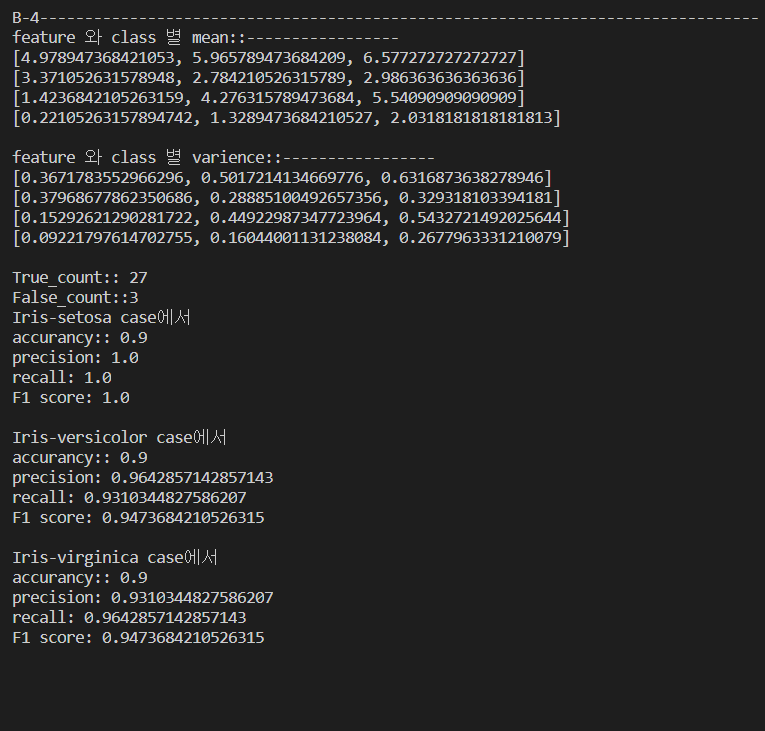


B-4

4번에서는 likelihood를 구할 떄 feature data를 0.1단위로 나누고 해당 클래스 범위에 어떤 확률로 있는지 확인 하는 것인데 이때는 내가 가지고 있는 trainingdata를 Gaussian distribution으로 근사해서 해당범위에 해당하는 pmf를 구할수 있다.이떄 likelihood를 설정할떄 naïve gaussian에서 해당 class의 pmf/(모든 class의 pmf의 확률의 합)이 된다. Likelihood를 통해서 posterior를 비교해 실제 class와 같은지 비교하면 된다.

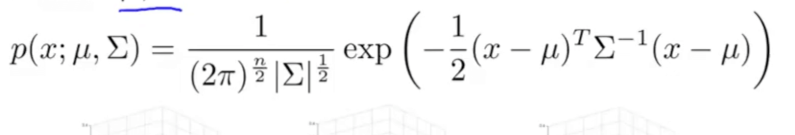
가우시안 분포로 근사 시킬 떄는 training data를 class별로 나눠서 mean 과 varience를 구하고 해당식에 평균과 분산을 대입하면 정규분포로 근사가 가능하다.

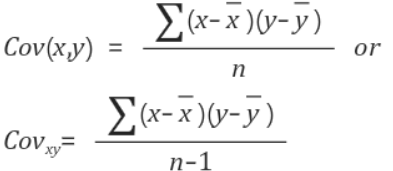
이방법으로 근사시 정확도는 90프로까지 올라갔다.

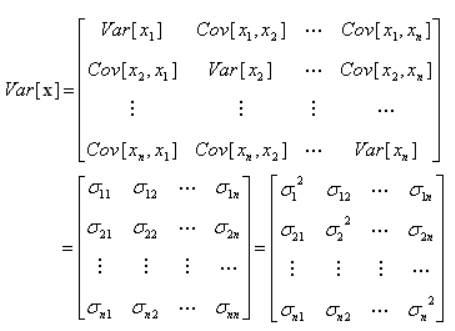


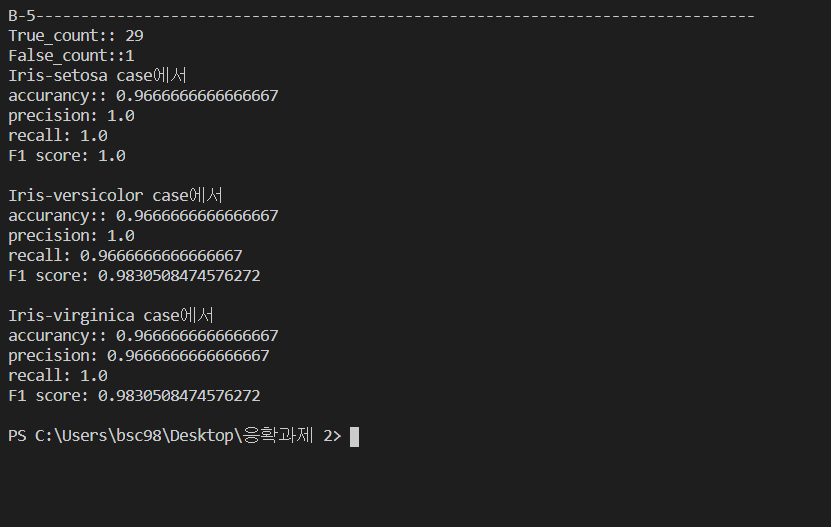
B-5

Multivariate Gaussian Distribution에서 Gaussian Distribution과 다른점은 single variable이 multi variable로 바뀌는 것이다 즉 다른 특성들 끼리의 연관성을 구해서 posterior을 구하는 것인데 그 역할을 공분산이 한다. 그래서 공분산을 통해 연관성을 파악하고 likelihood를 구해 posterior를 구하는 것이다. 평균과 공분산의 행렬을 구하고 test case를 x로 벡터화 시킨후 p를 구하면 된다.









이떄 정확도는 97프로로 지금까지 classification한 모델 중 제일 정확도가 높았다.

|  |  |  |
| --- | --- | --- |
| 사용 package | import | 속성 |
| csv | Open(file) | Data를 불러오기 |
| random | Random.shuffle() | Kfold-cross-validation |
| numpy | Exp,determinant,transpose,inverse | 5번에서 p(x)구할 때 |
| math | Sqrt,pi | 4번에서 정규화할떄 사용 |