**Part A. Preliminaries**

1. **Cross Validation**

k-fold cross validation 방법은 주어진 data를 k개의 구간으로 나눈 후 그 중 하나의 집단을 제외한 다른 집단을 training data로 두고 학습을 시킨 후 남은 하나의 집단을 test data로 두어 실행시킵니다. 이 방식을 각각의 집단마다 한번씩, 총 k번 실시한 후 나온 결과를 평균을 내게 되면 k-fold cross validation이라 합니다. 이를 표로 보게 되면 다음과 같습니다. (ex. k = 4)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Fold 1 | Fold 2 | Fold 3 | Fold 4 |
| Try 1 | Test data | Training data | Training data | Training data |
| Try 2 | Training data | Test data | Training data | Training data |
| Try 3 | Training data | Training data | Test data | Training data |
| Try 4 | Training data | Training data | Training data | Test data |

1. **Testing your classifier**

Precision-recall은 모델 결과와 실제 결과를 비교해서 그 값들을 통해 알 수 있는 모델의 성능 수치입니다. 이때 필요한 비교 값을 분류할 때, 모델의 결과와 실제 결과가 같다면 True, 실제 결과와 다르다면 False라 하고, 모델이 참이라고 분류하면 Positive, 거짓이라고 분류하면 Negative이라고 합니다. 따라서 모델의 결과와 실제 결과의 조합을 표로 나타내면 다음과 같습니다.

|  |  |  |
| --- | --- | --- |
| 모델 결과  실제 결과 | True | False |
| True | True Positive | False Negative |
| False | False Positive | True Negative |

FP, FN은 모델링 결과값의 오류인데 FP를 Type Ⅰ error, FN을 Type Ⅱ error라고도 부릅니다. 여기서 precision 즉, 정밀도는 모델이 참으로 판단했을 때 실제결과도 참일 확률을 뜻하므로 으로 나타냅니다. 또한 recall 즉, 재현율은 실제 결과가 참일 때 모델이 판단한 결과도 참일 확률을 뜻하므로 으로 나타냅니다. 이때 precision이 1이라는 의미는 FP가 0이라는 의미이므로 모델이 거짓을 참으로 판단하지 않는다는 의미이고, recall이 1이라는 의미는 FN이 0이라는 의미이므로 모델이 참을 거짓으로 판단하지 않는다는 의미입니다. 따라서 모델의 precision과 recall이 1에 가깝다는 말은 모델이 잘못 판단하는 경우가 적다는 말입니다. 또한 precision과 recall 어느 하나는 1에, 어느 하나는 0에 가까운 경우가 생길 수 있는데 이는 모델이 참이나 거짓을 잘 못 판단하고 있다는 것이므로 precision과 recall 모두 1에 가까운 것이 좋은 모델이라고 할 수 있겠습니다.

1. **Acquiring dataset**

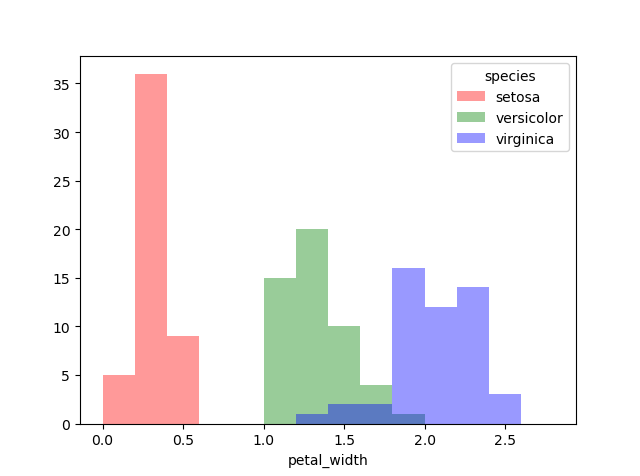
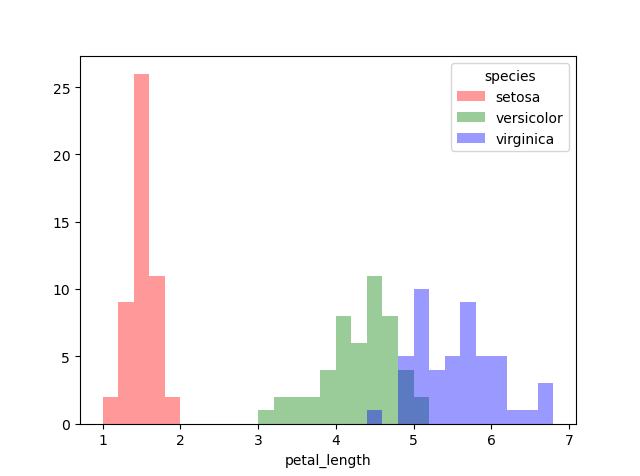
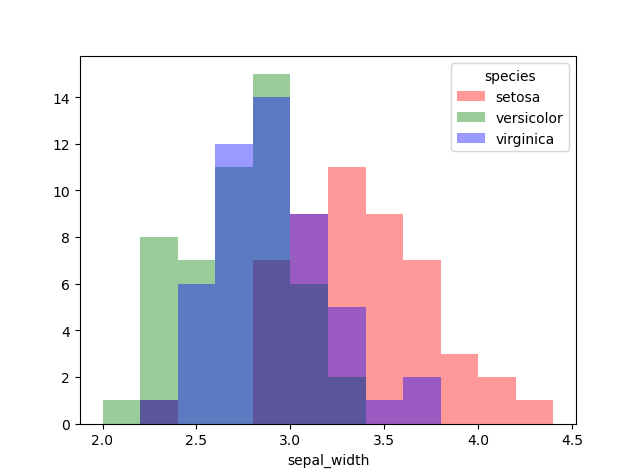
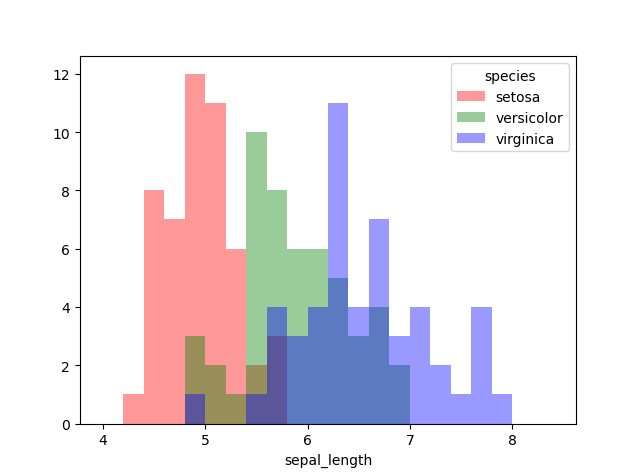
“iris.data”에는 3종류의 iris가 각각 50개의 data를 가지고 있는데, 모든 단위는 cm 단위이며 꽃받침의 길이, 꽃받침의 폭, 꽃잎의 길이, 꽃잎의 폭, iris의 종 순서대로 나열되어 있습니다. “iris.names”에서 35번째 샘플의 4번째 데이터(꽃잎의 폭)과 38번째 샘플의 2번째 데이터(꽃받침의 폭)은 데이터의 오류가 있다고 명시되어 있으므로 수정해 주었습니다. 또한 각 특징에 따른 데이터의 통계는 다음과 같다고 합니다.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Min | Max | Mean | SD | Class Correlation |
| sepal length | 4.3 | 7.9 | 5.84 | 0.83 | 0.7826 |
| sepal width | 2.0 | 4.4 | 3.05 | 0.43 | -0.4194 |
| petal length | 1.0 | 6.9 | 3.76 | 1.76 | 0.9490 |
| petal width | 0.1 | 2.5 | 1.20 | 0.76 | 0.9565 |

**Part B. Iris classification using text data**

1. **4 Histograms of Iris features**

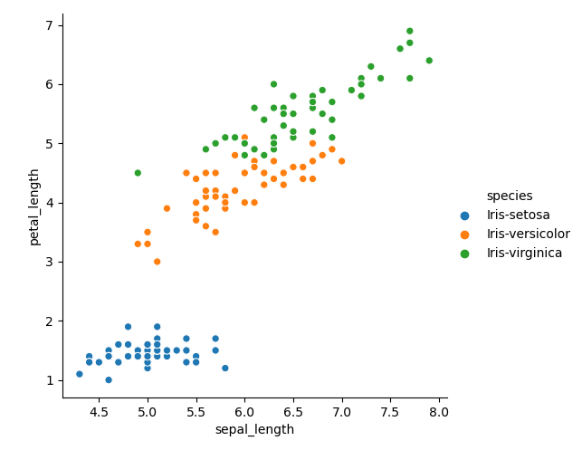
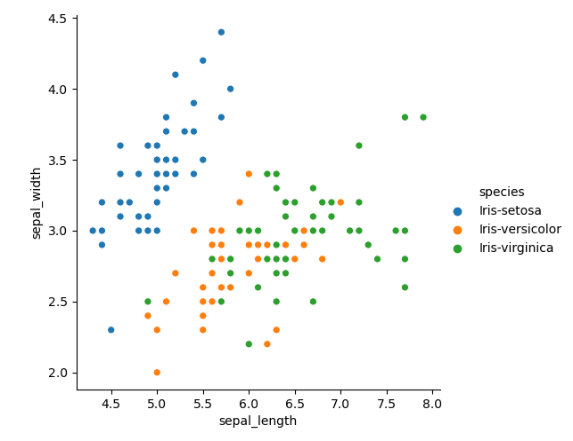
Iris의 특징은 꽃받침의 길이, 폭, 꽃잎의 길이, 폭으로 총 4가지인데 각각의 histogram은 다음과 같습니다.

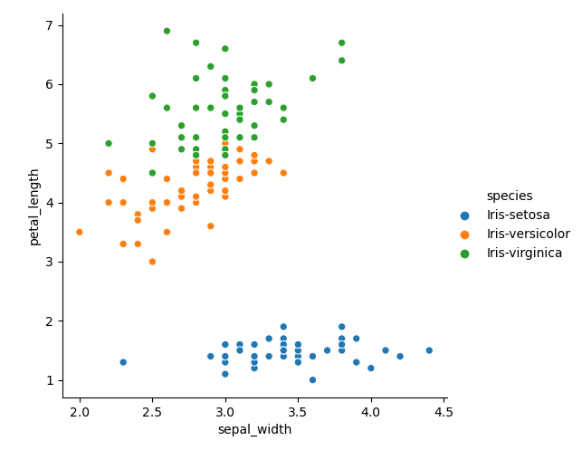
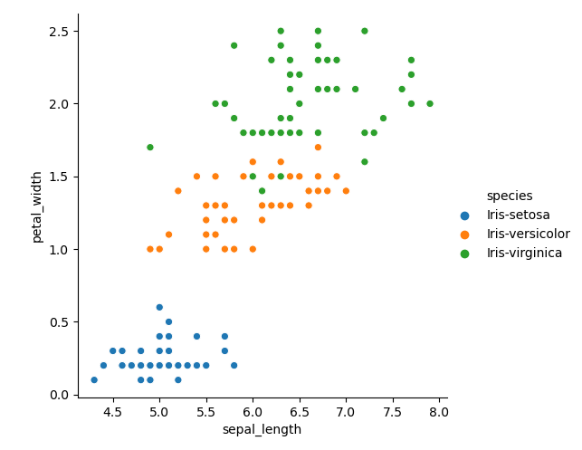


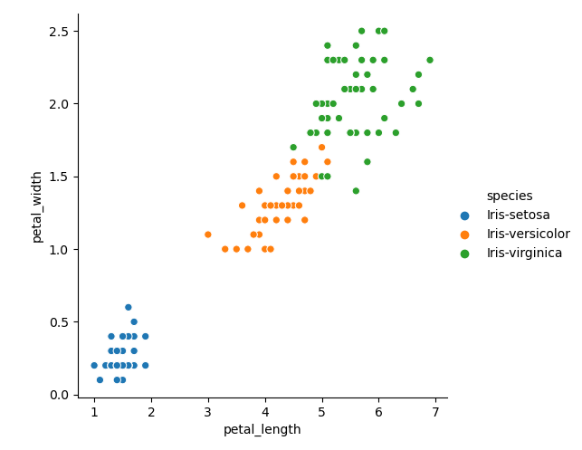
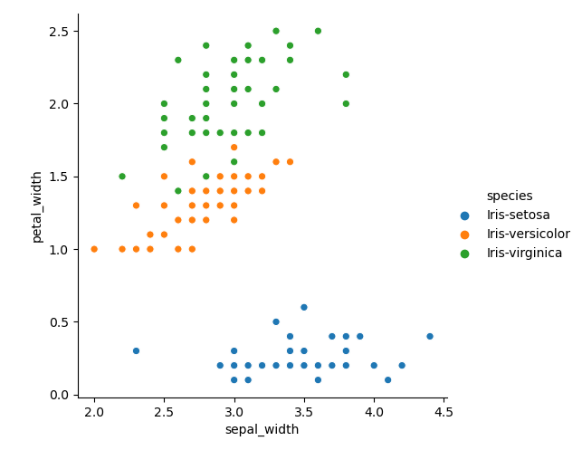
위 히스토그램들은 구간 범위를 0.2cm단위로 설정하여 히스토그램을 만들었습니다. 이 히스토그램들은 error 비용을 최소화하기 위해 최적의 의사 결정 경계를 찾아야 하는데, 여러 시도 끝에 찾은 결과는 sepal\_length는 5.4cm, 6.2cm, sepal\_width는 2.6cm, 3.2cm, petal\_length는 2.4cm, 4.8cm, petal\_width는 0.8cm, 1.8cm로 했을 때 각 의사 결정 경계에 따른 error 비용은 38, 66, 7, 6으로 최소화됐습니다. 이 결과를 통해 sepal, 즉, 꽃받침을 통해 구분하는 것은 length와 width 모두 error비용이 너무 크고, petal\_length, petal\_width를 기준으로 나눴을 때 setosa의 경우 다른 종과 전혀 겹치지 않는 다는 것을 알 수 있습니다. 따라서 한가지의 특징만 가지고 classifier를 만들고자 한다면 petal, 즉, 꽃잎의 길이나 폭을 기준으로 만드는 것이 좋은 선택이라고 할 수 있습니다.

1. **6 scatterplot of Iris features**

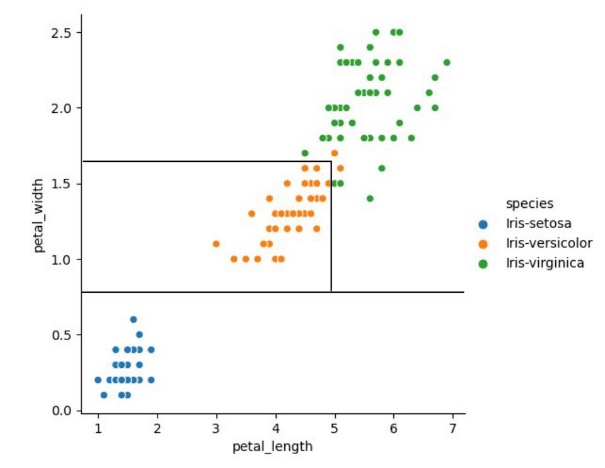
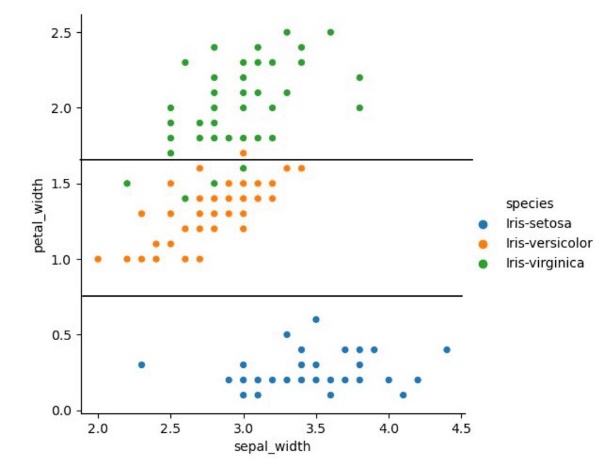
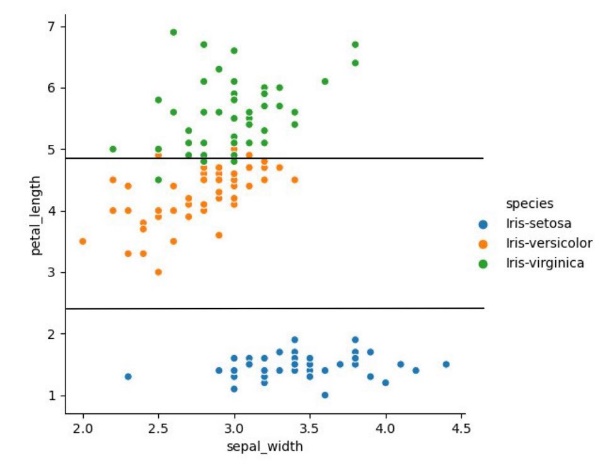
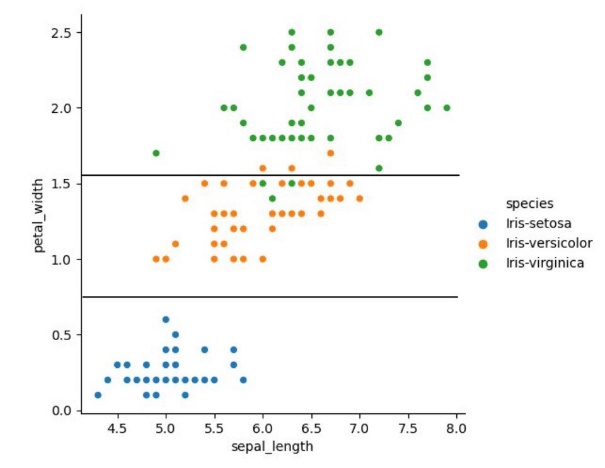
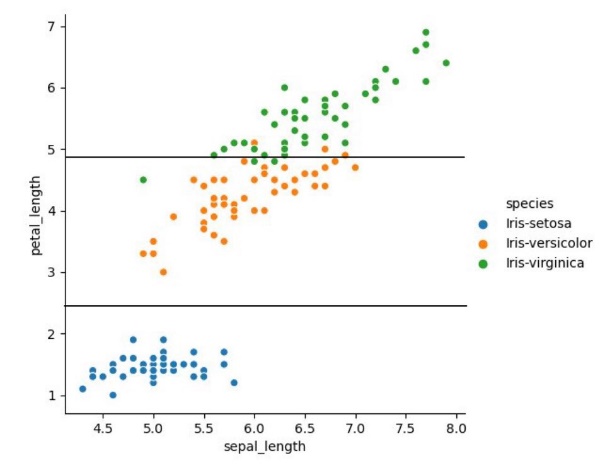
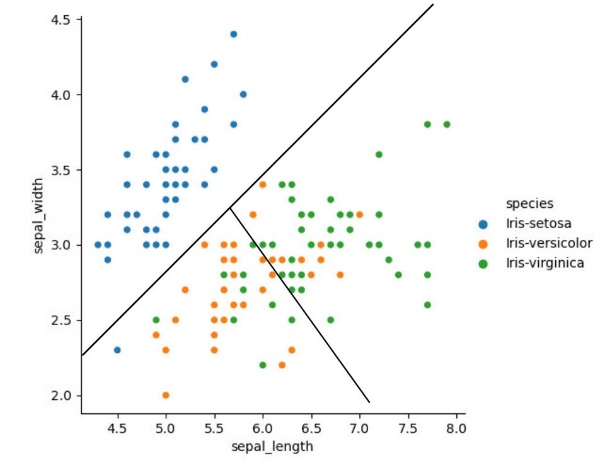
Iris의 4가지 특징의 조합으로 만들어 낼 수 있는 6개의 scatterplot은 다음과 같습니다.







이 scatterplot도 비용을 최소화하기 위해 최적의 의사 결정 경계를 찾아야 하는데, 이번에는 scatterplot들을 직선으로 의사 결정 경계 설정하여 최적의 의사 결정 경계를 계산해보겠습니다. 이를 위해 각 scatterplot에 error가 최소가 되는 직선을 그려보았으며, 그 형태는 다음과 같습니다.



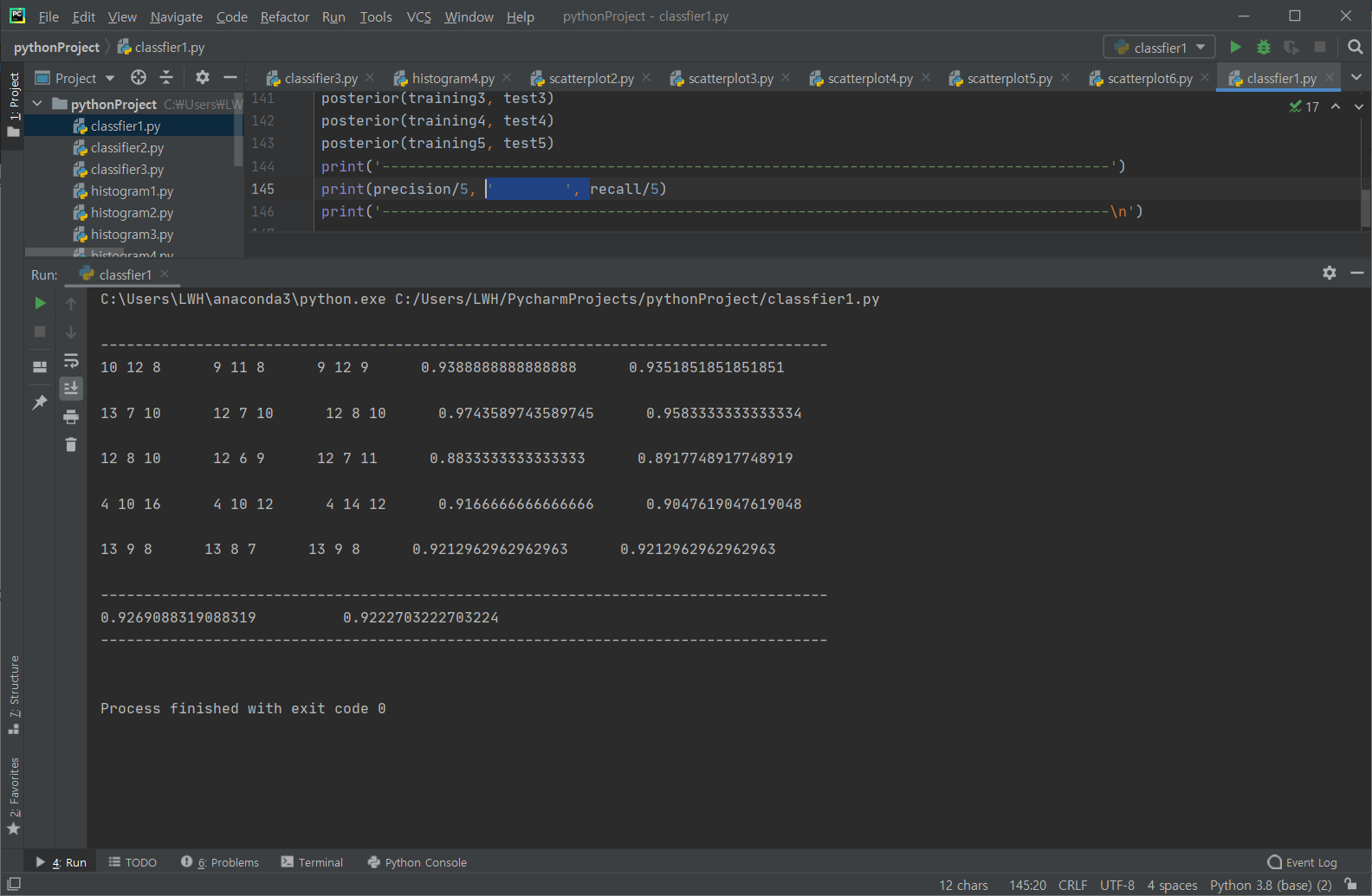
각각의 의사 결정 경계에 따른 error는 19, 6, 6, 6, 5, 2으로 나왔습니다. 이를 통해 petal\_length와 petal\_width 중 하나만 있어도 error가 매우 적게 나오는 것을 알 수 있으며 sepal\_legnth와 sepal\_width를 통해 classifier를 만들게 되면 기울기가 있는 직선으로 판별하게 되기 때문에 다른 방식보단 판별하기 위한 식이 복잡해집니다. 따라서 classifier를 보다 정확하게 만들려면 petal\_length와 petal\_width를 통해 만들면 정확하면서도 간단한 식으로 만들 수 있게 됩니다.

1. **Design classifier to pmf**

Bayesian Decision Theory는 Posterior = Likelihood \* Prior / Evidence 라고 정의하고 있습니다. 이때 을 iris의 종, 를 각각의 특징이라고 하면 Posterior = P(|), Likelihood = P(|), Prior = P(), Evidence = P()이므로 Prior인 P()는 P(setosa), P(versicolor), P(virginica), 즉, 트레이닝 셋에서 획득한 setosa, versicolor, virginica의 비율이며, Likelihood인 P(|)는 트레이닝 셋의 종에서 테스트 셋의 특징이 존재할 확률이며, Evidence인 P()는 전체 중에 가 있을 확률이므로 전체 트레이닝 셋 중에 테스트 셋의 특징이 존재할 확률입니다. 이는 P()P()+P()P()+P()P()로 나타낼 수 있으므로 는 setosa의 Posterior가 됩니다. 또한, 는 iris의 모든 특성이 독립이기 때문에가 되고, 이 부분의 프로그램을 실행한 결과 확률이 0이 되는 경우가 발생했습니다. 이를 해결하기 위해 laplace smoothing이라는 기법을 사용했는데 이 기법은 분모와 분자에 특정 상수를 더해 분모나 분자가 0이 되는 zero frequency를 해결해 줍니다. 또한 class가 3종류 일 때의 A에 대한 TP, TN, FP, FN는 다음과 같습니다.

|  |  |  |  |
| --- | --- | --- | --- |
| 모델 결과  실제 결과 | A | B | C |
| A | TP | A의 FN | A의 FN |
| B | A의 FP | A의 TN | A의 TN |
| C | A의 FP | A의 TN | A의 TN |

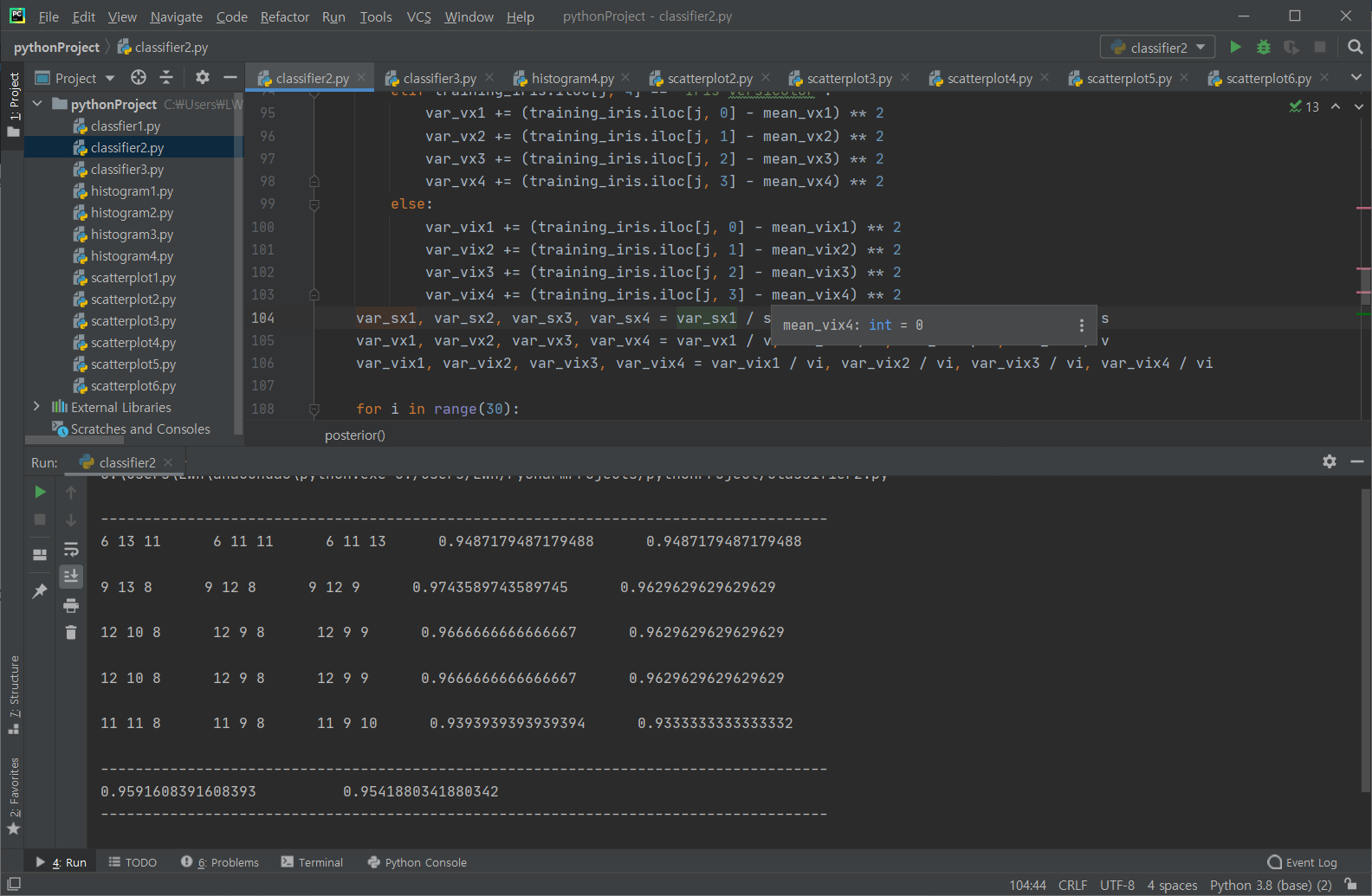
Posterior를 통해 나온 3개의 class에 대한 확률을 비교해 가장 큰 값을 모델의 결과로 하는 classifier를 만들어 5-fold cross-validation을 해보도록 하겠습니다. 다음은 각 종의 모델의 결과(TP+FP), TP, test data의 실제 종 수(TP+FN), 그리고 precision, recall의 값입니다.



Bayesian Decision Theory에 의해 만들어진 classifier는 평균 0.93대의 precision과 0.92대의 recall 값을 가지므로 이 모델은 iris 종의 분류를 잘한다는 것을 알 수 있습니다.

1. **Design classifier to Gaussian distribution**

먼저 이산적인 데이터를 Gaussian distribution으로 만들어 사용해보겠습니다. 이산적인 데이터를 통해 만들어진 Gaussian distribution의 likelihood는 Gaussian distribution의 pdf인 을 이용한 이므로 필요한 정보는 μ, σ입니다. μ는 평균, σ는 표준편차이므로 이산적인 iris 데이터를 기반으로 만든 평균과 분산을 구한 후 위 식에 넣게 되면 x값에 따른 확률을 알 수 있으므로 classifier를 만들 수 있습니다. 다음은 gaussian distribution으로 만든 5-fold cross-validation의 각 종의 모델의 결과(TP+FP), TP, test data의 실제 종 수(TP+FN), 그리고 precision, recall의 값입니다.

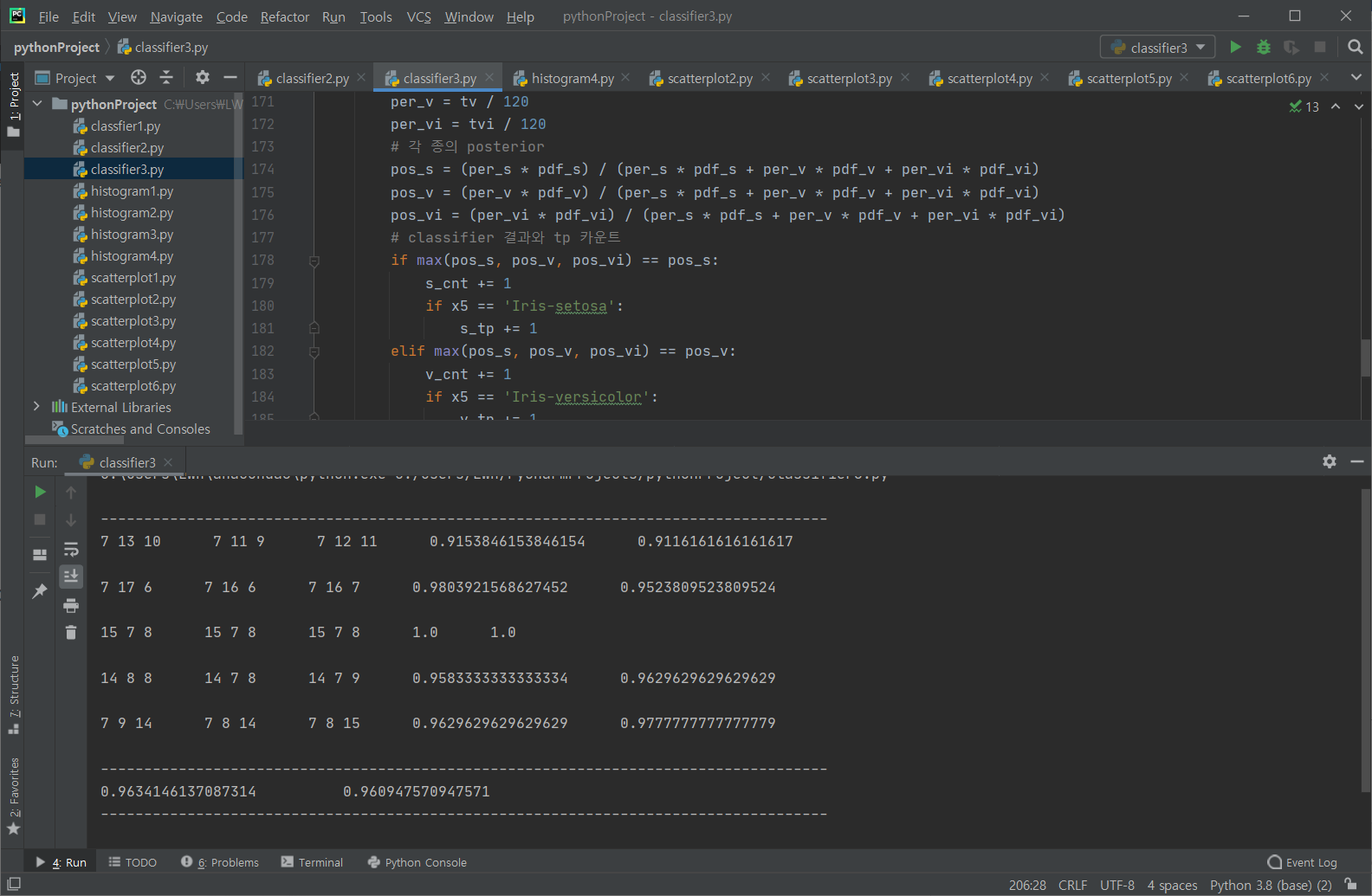


Bayesian Decision Theory에 의해 만들어진 classifier는 평균 0.96대의 precision과 0.95대의 recall 값을 가지므로 이 모델도 iris 종을 분류하는 능력이 좋은 것을 알 수 있습니다. 또한, pmf에서 likelihood가 0이 나왔던 것이 pdf에서는 연속 확률 분포이므로 0이 되는 경우가 없어 laplace smoothing을 사용하지 않아도 되는 장점이 있었습니다.

1. **Extend by using a multi-variate Gaussian model**

multi-variate gaussian model의 경우 평균과 분산을 벡터, 행렬로 나타내는데 이 중에서 변수가 k개인 2차원 multi-variate gaussian model의 평균과 분산은 다음과 같습니다.

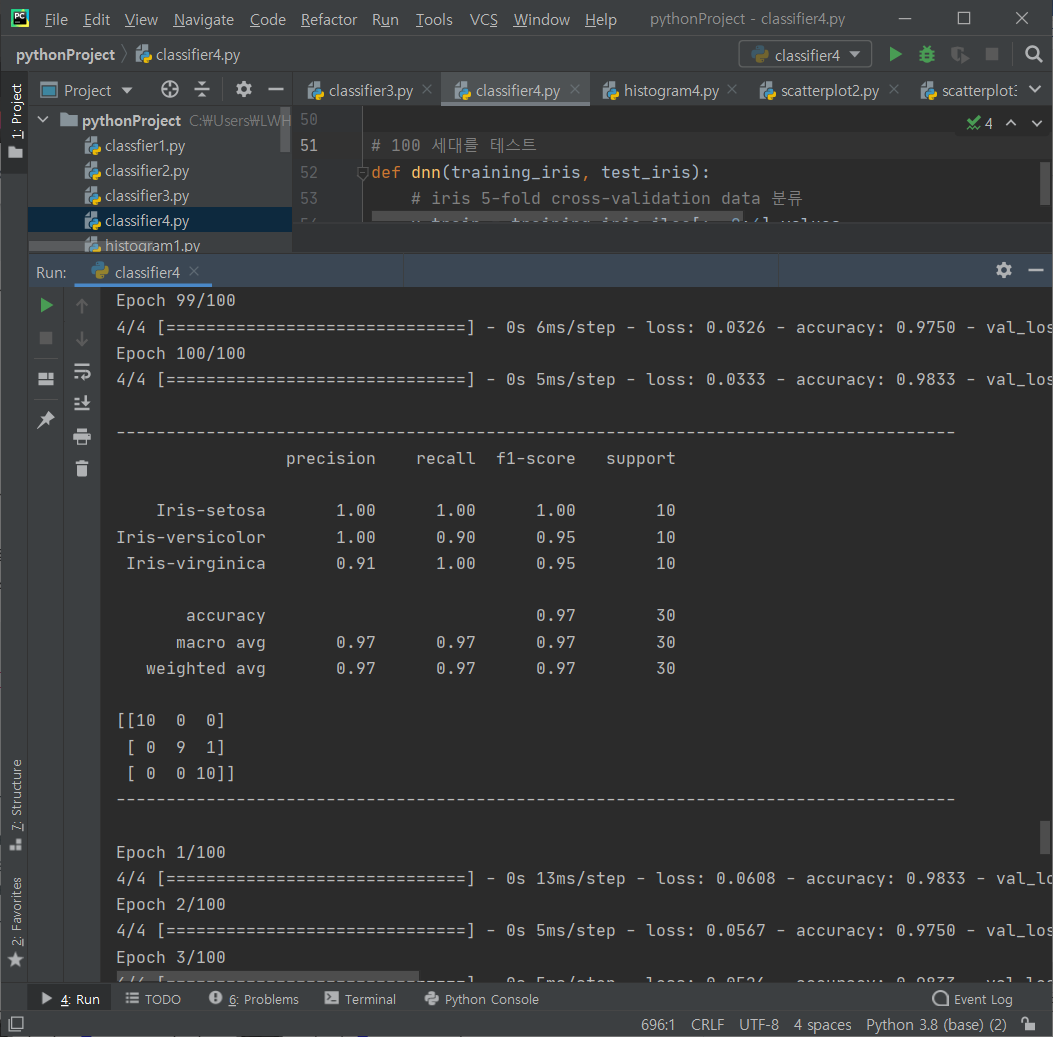
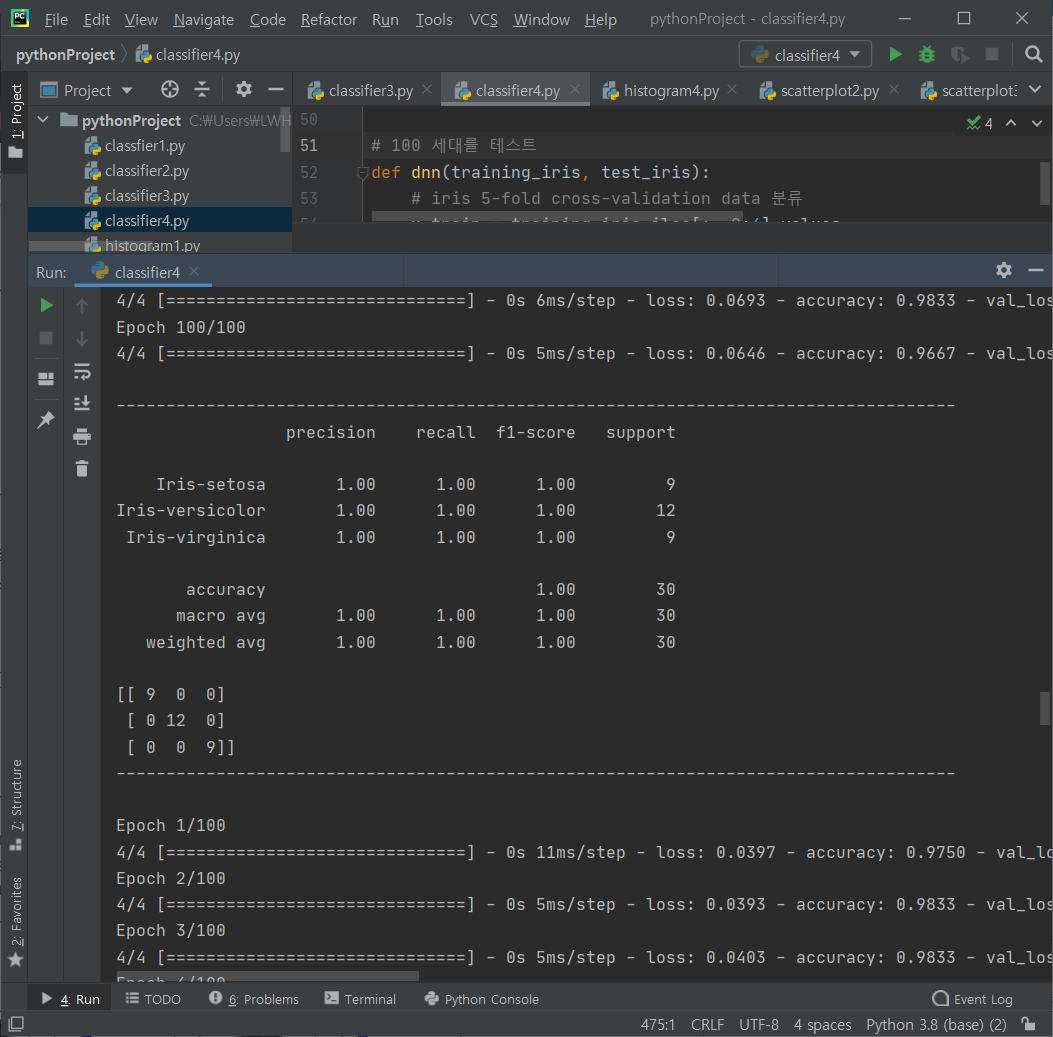
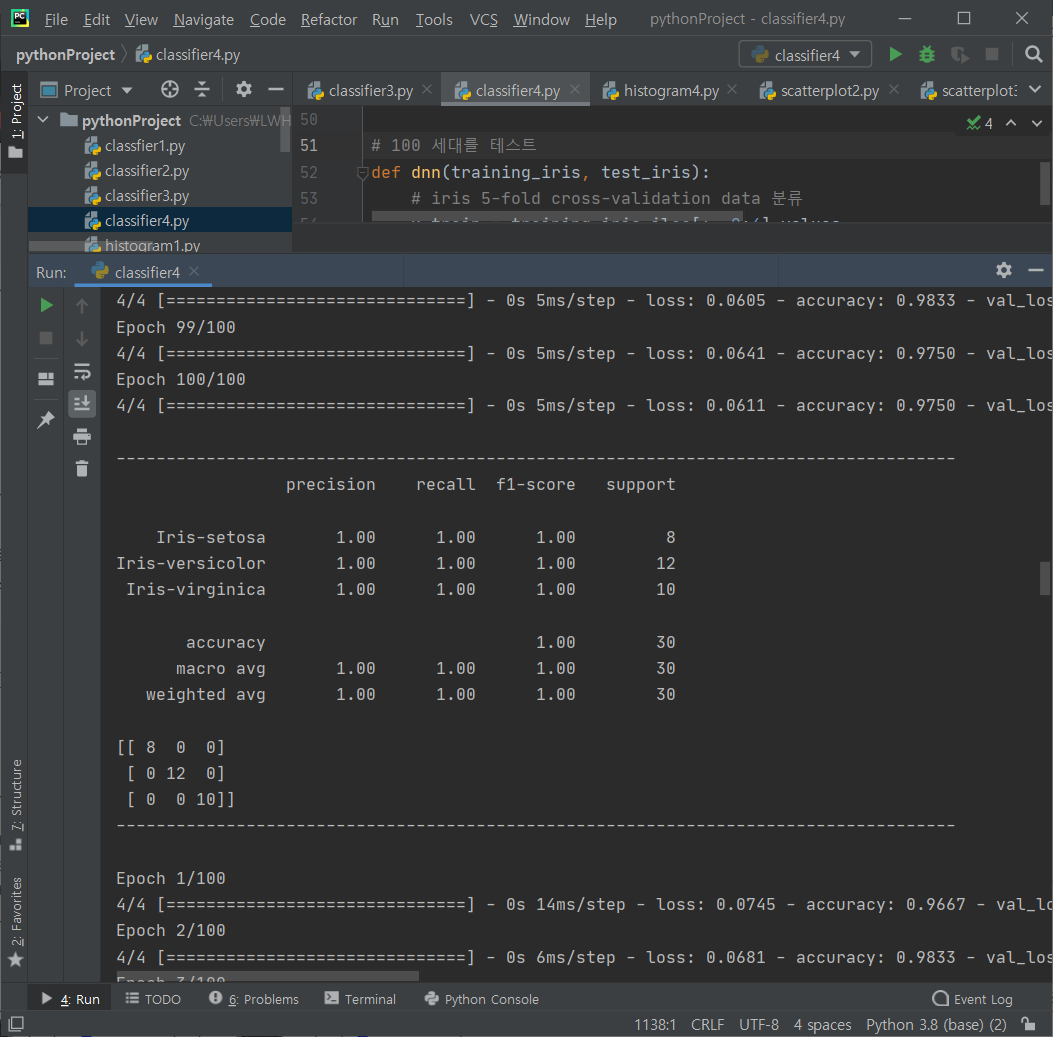
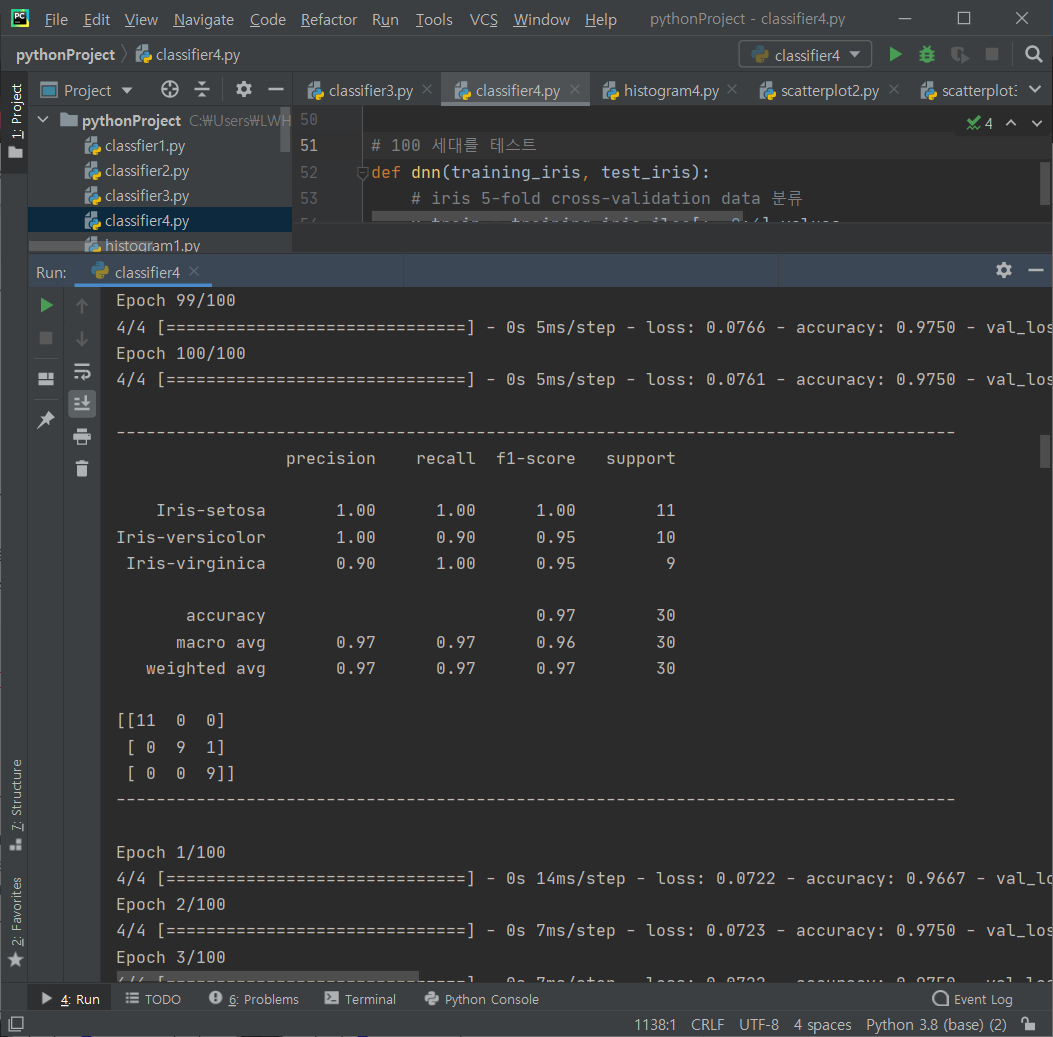
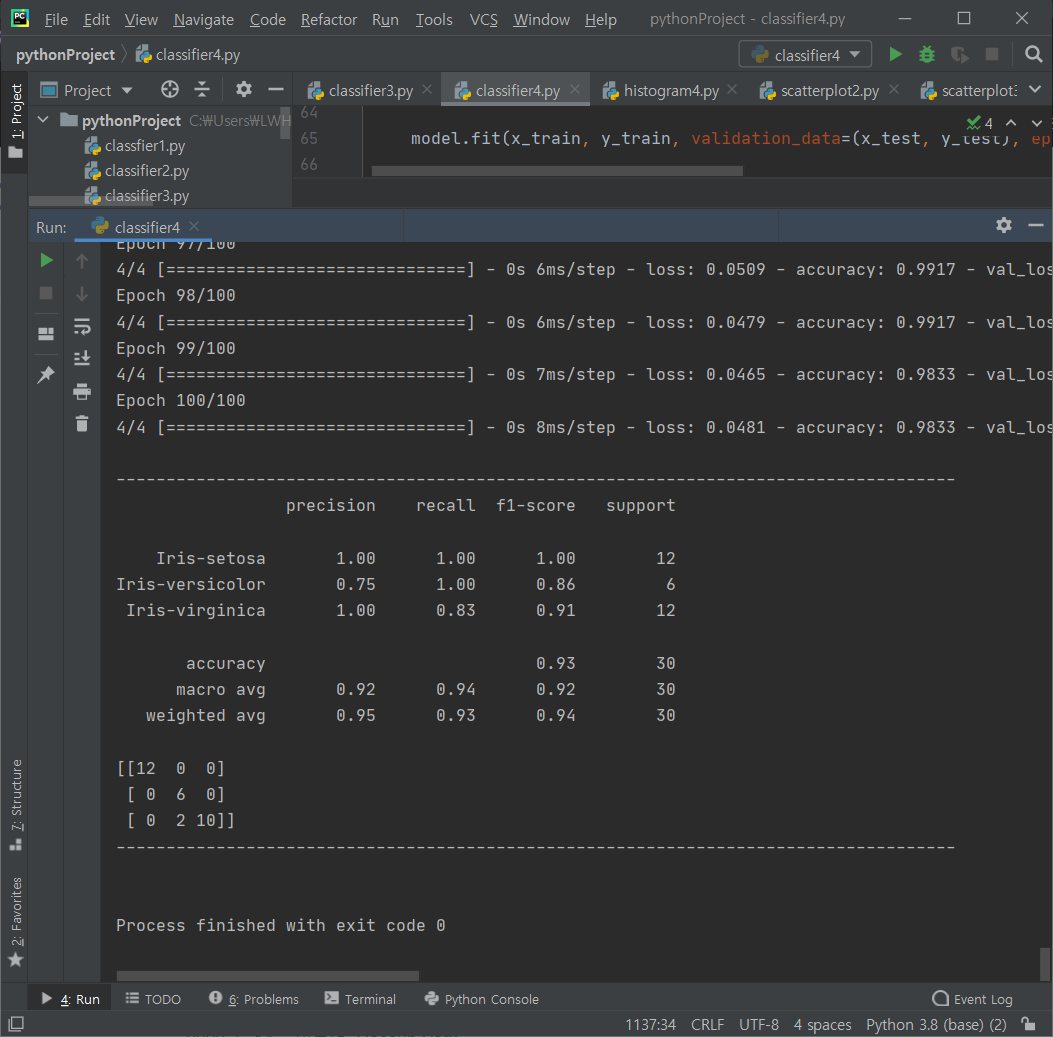
또한, 이 행렬의 대각선 요소는 각 변수의 분산이고, 비대각선 요소는 변수 간의 공분산을 말합니다. 이를 이용한 multi-variate gaussian pdf는 1차원 gaussian model의 pdf와 비슷한 형태로 다음과 같습니다. 이때 는 전치행렬, 는 역행렬, 는 행렬식을 나타냅니다. 이 multi-variate gaussian pdf를 통해 다시 Bayesian Decision Theory로 classifier를 제작해 보겠습니다.



multi-variate gaussian mode에 의해 만들어진 classifier는 평균 0.96대의 precision과 0.96대의 recall 값을 가지므로 이 모델도 iris 종을 분류하는 능력이 좋은 것을 알 수 있습니다. 또한 gaussian model과 마찬가지로 pdf를 통한 연산이므로 pmf에서 likelihood 부분에 필요했던 laplace smoothing을 사용하지 않아도 되는 장점이 있었고, 모든 연산을 행렬을 통해 한번에 하게 되므로 gaussian model보다 식이 간결해지는 장점이 있었습니다.

1. **DNN (Extra credit)**

먼저 DNN에 사용된 신경망은 keras이며, 이 DNN은 4개의 입력 노드 + 64개의 노드를 가지는 2개의 은닉층 + 3개의 출력 노드를 가지는 softmax 층으로 이루어져 있습니다. 또한 모델을 설정한 후에 함수를 만들어 각각의 트레이닝 셋과 테스트 셋을 가지고 프로그램을 실행했으며 각각 100세대씩 진행을 했습니다. 그 결과는 다음과 같습니다.

이 DNN의 결과들은 precision과 recall, f1-score, 총 데이터 순으로 나열되어 있으며 정확도, 라벨 별 합의 평균(macro avg), 가중 산술 평균(weighted avg) 순으로 나와있습니다. 또한 3번 문제에서 나왔던 표를 이용하여 데이터의 TP, TF, FP, FN을 한눈에 보여주도록 작성하였습니다. 또한 5번의 결과를 통한 5-fold cross-validation 결과로는 precision이 평균 0.97, recall이 0.98로 매우 높게 나오는 것을 알 수 있습니다. 따라서 DNN으로 만든 classifier는 상당히 정확하다는 것을 알 수 있습니다.