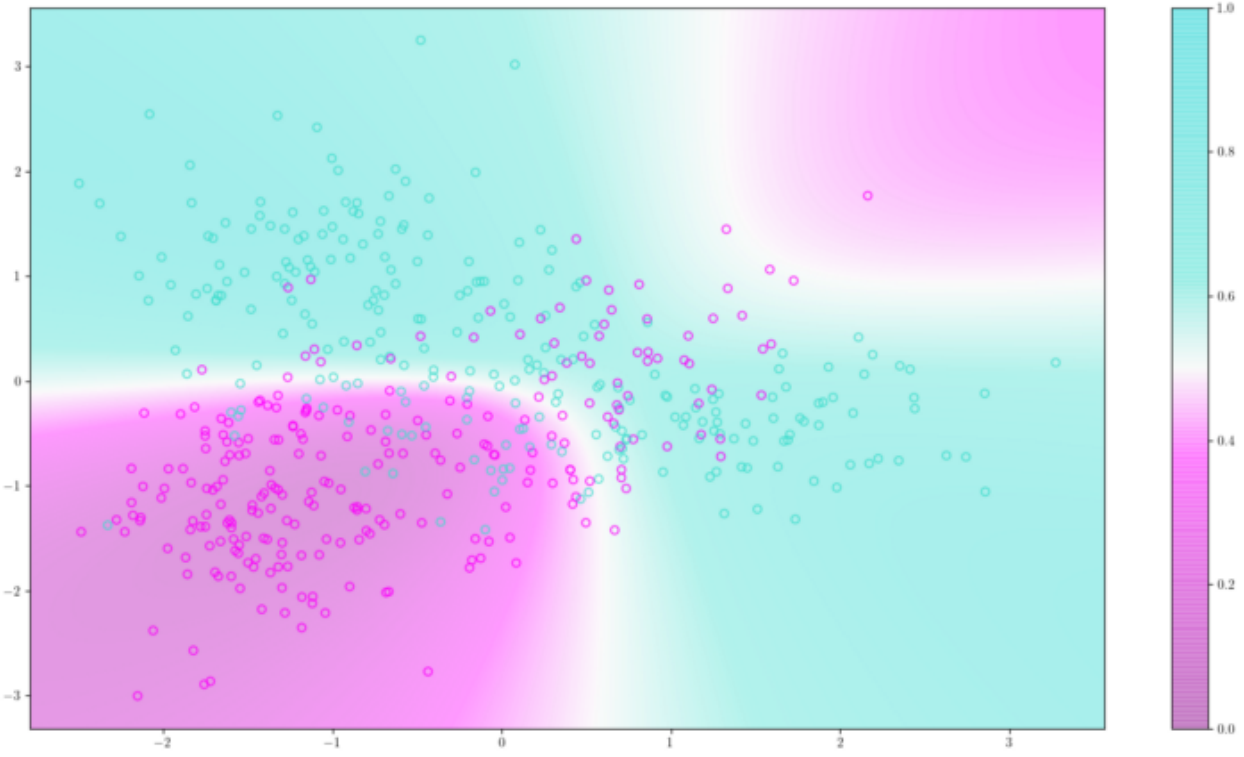
Pattern Recognition and Machine Learning

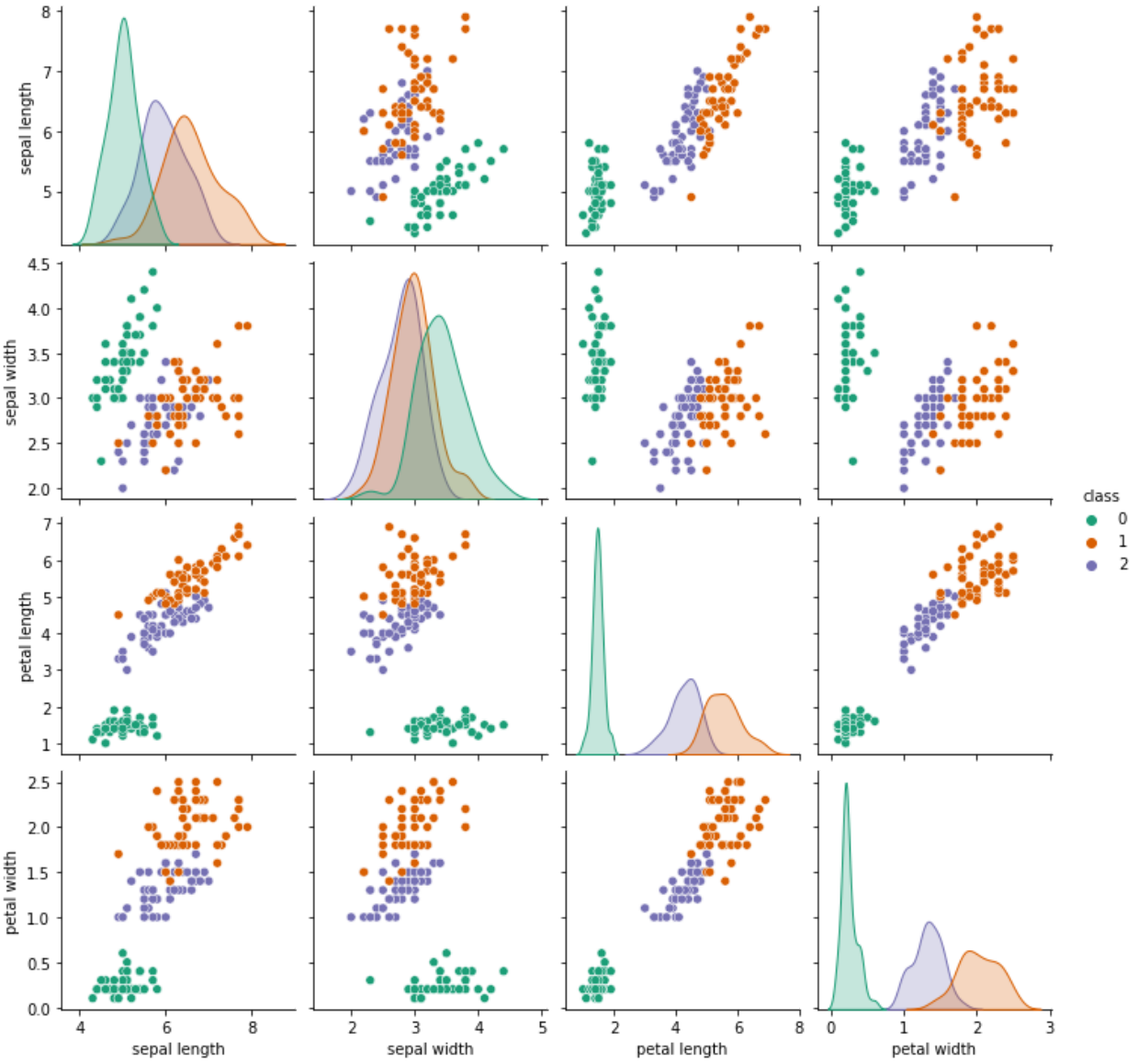
Lab 5 Report (Kartik Choudhary, B2CS025)

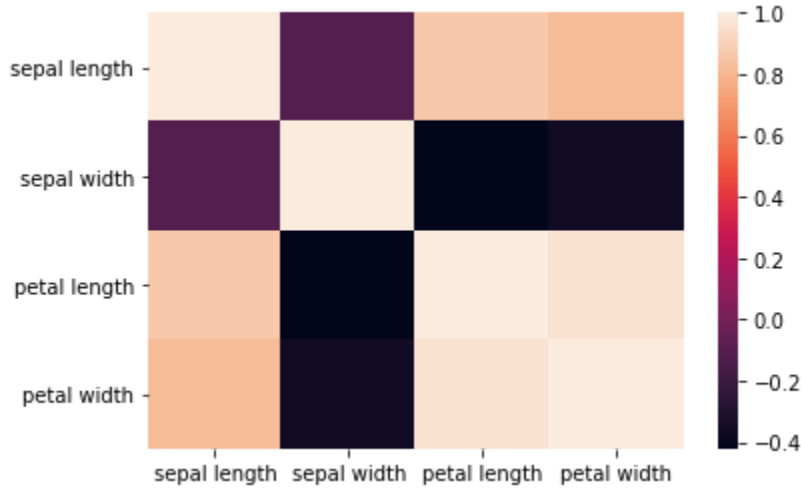
QDA and LDA



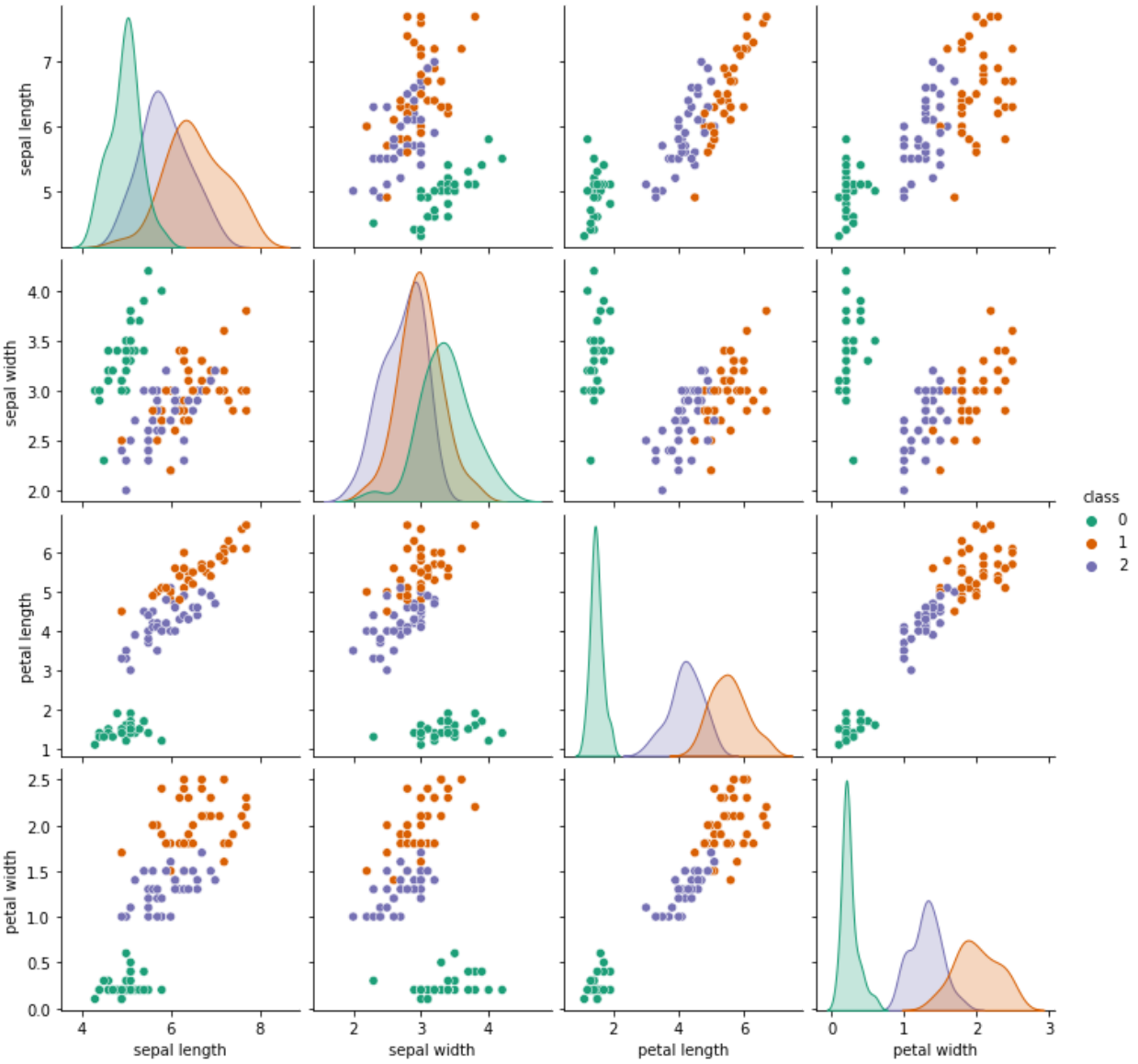
Question 1

A: Data Preprocessing was done on the dataset to make it fit for the Quadratic Discriminant Analysis (QDA).Test trains were split into 30:70 ratios. Dataset was visualized as,




sepal length
sepal width
petal length
petal width
sepal length
sepal width
petal length
petal width
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4

Visualization for training data:



B. Three pairs of features are chosen: the first set of features being (sepal length, sepal width) second being (sepal width ,petal width) and the third being (sepal length, petal width). The Quadratic Discriminative model is trained on the three sets of features.

C. Mean and Covariance for the first distribution is

```
print(qda1.means_)
print(qda1.covariance_)
```

```
[[4.96451613 3.36129032]
 [6.55945946 2.98648649]
 [5.86216216 2.72432432]]
[array([[0.11569892, 0.09924731],
       [0.09924731, 0.14178495]]), array([[0.43414414, 0.09777027],
       [0.09777027, 0.09897898]]), array([[0.28297297, 0.08816817],
       [0.08816817, 0.08966967]])]
```

Mean and Covariance for the second distribution is:

```
#Mean score and Covariance
print(qda2.means_)
print(qda2.covariance_)
```

```
[[3.36129032 0.24516129]
 [2.98648649 2.00540541]
 [2.72432432 1.3027027 ]]
[array([[0.14178495, 0.01047312],
       [0.01047312, 0.01255914]]), array([[0.09897898, 0.06146396],
       [0.06146396, 0.0883033 ]]), array([[0.08966967, 0.04215465],
       [0.04215465, 0.04249249]])]
```

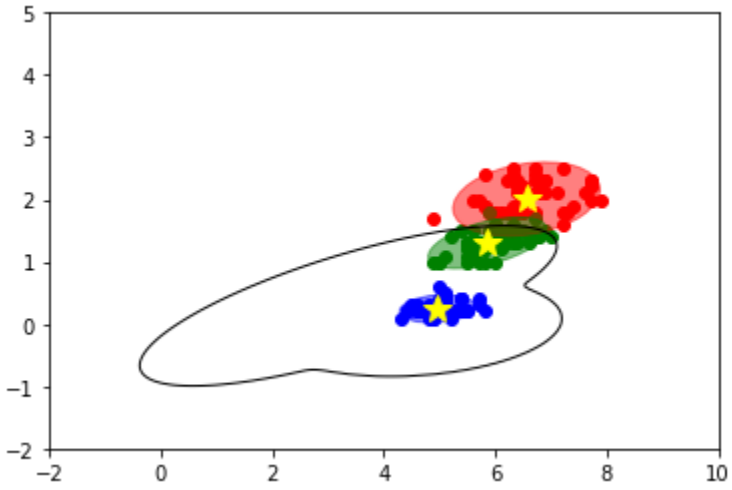
Mean and Covariance for the third distribution is:

```
: #Mean score and Covariance
  print(qda1.means_)
  print(qda1.covariance_)
```

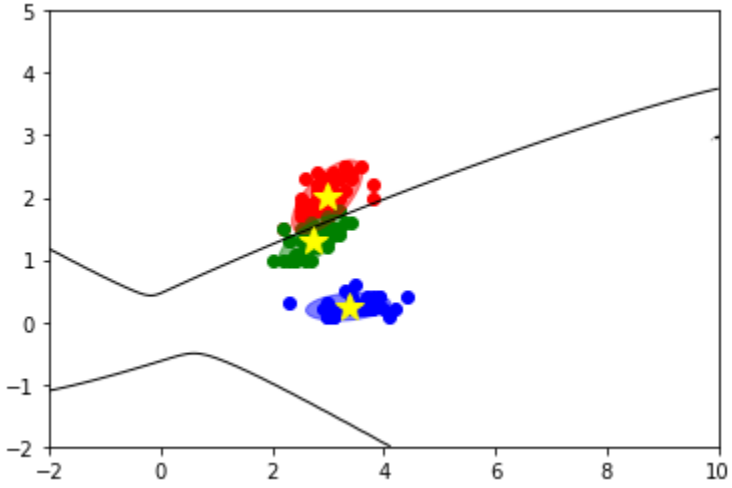
```
[[4.96451613 3.36129032]
 [6.55945946 2.98648649]
 [5.86216216 2.72432432]]
[array([[0.11569892, 0.09924731],
       [0.09924731, 0.14178495]]), array([[0.43414414, 0.09777027],
       [0.09777027, 0.09897898]]), array([[0.28297297, 0.08816817],
       [0.08816817, 0.08966967]])]
```

D. The decision boundary given by the QDA model for the above distributions is:

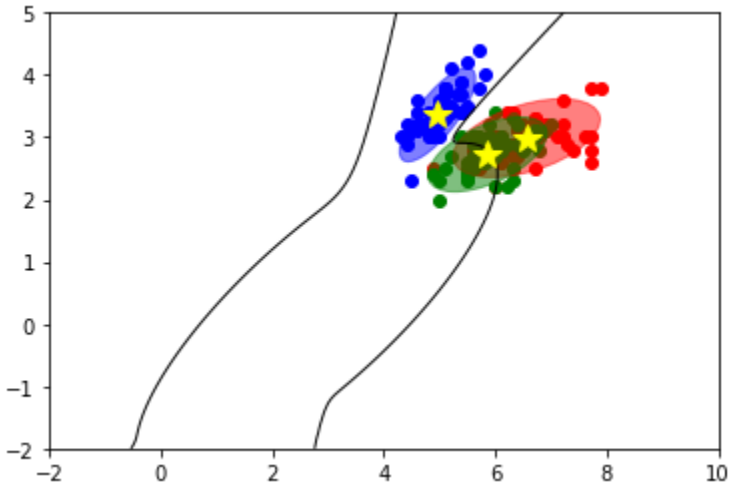
For the third distribution:



For the second distribution:



For the first distribution:



E. Prediction of the test data for the third distribution:

```
accuracy_score(y3_test, pred3)
```

```
: 1.0
```

Error Rate = 0.0

Prediction of the test data for the second distribution:

```
accuracy_score(y2_test, pred2)
```

```
0.9777777777777777
```

Error Rate: 0.0222222222222223

Prediction of the test data for the first distribution:

```
accuracy_score(y1_test, pred1)
```

```
0.8
```

Error Rate: 0.2

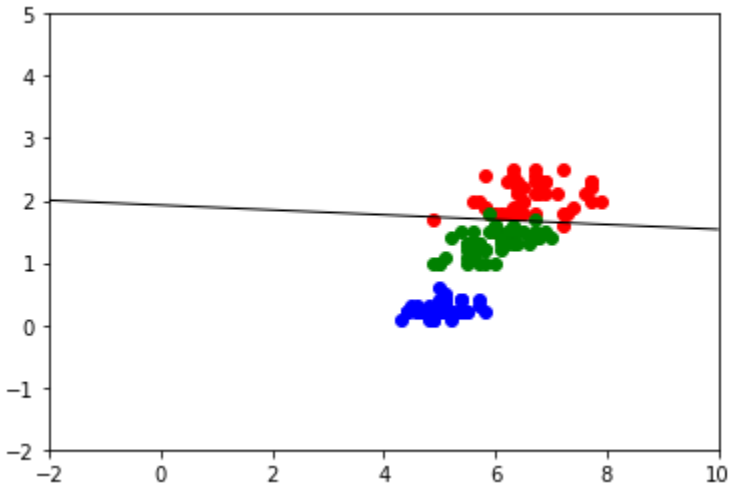
Clearly the third distribution gives the best result i.e choosing the features sepal length, petal width since the error rate is minimum.

F. Taking the pair of features SEPAL LENGTH and PETAL WIDTH and training it on the LDA model with the same training data.

```
lda1 = LinearDiscriminantAnalysis()
lda1.fit(X3_train, y3_train)
```

```
LinearDiscriminantAnalysis()
```

G. The decision boundary given by the LDA model on the top of the scatterplot visualization of the data is:



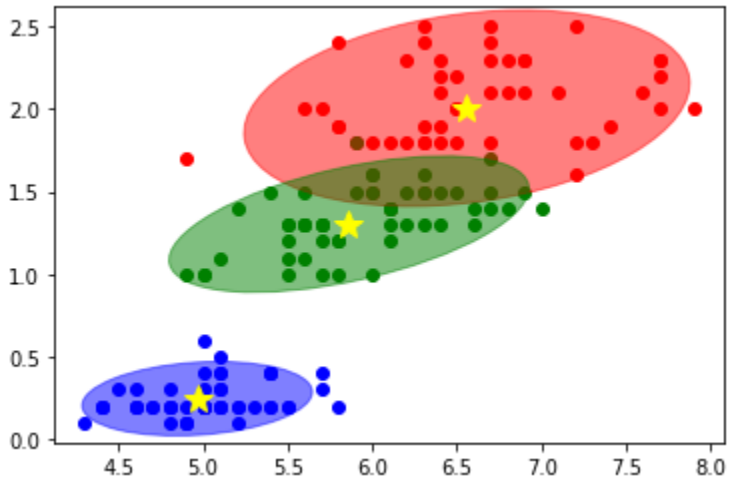
H.

Accuracy by the LDA model = 1.0, Error Rate = 0.0

```
predLDA1 = lda1.predict(X3_test)
accuracy_score(y3_test, predLDA1)
```

```
1.0
```

I.



Question 2:

Preprocessing : Already done in the above question reusing the exact same dataset.Hence,directly going to split the data into train and test with 70:30 ratio.

Initializing the class wise mean as -1 .

Using Petal Length and Petal Width as the features.

The obtained Classwise mean for petal width and length are as follows.

```
classWiseMean
```

```
{0: [1.464, 0.2439999999999999],
 1: [5.552, 2.026],
 2: [4.26, 1.3259999999999998]}
```

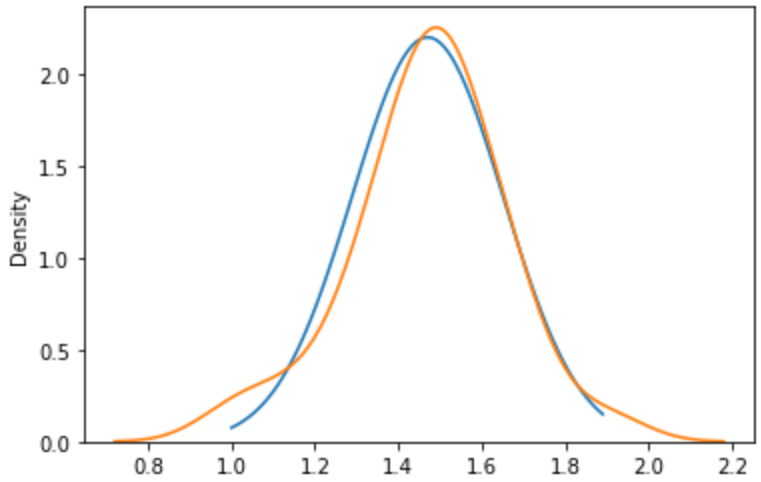
Similarly, Covariance matrix for every class is as follows :

```
dicCovMatrix
```

```
{0: array([[0.029504, 0.005584],
        [0.005584, 0.011264]]),
 1: array([[0.298496, 0.047848],
        [0.047848, 0.073924]]),
 2: array([[0.2164  , 0.07164 ],
        [0.07164 , 0.038324]])}
```

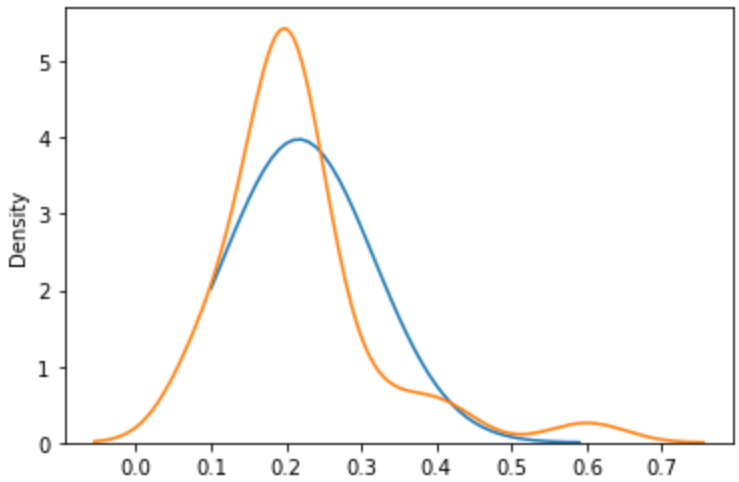
Later, by using a covariance matrix for every class and mean, I calculate the likelihood, giving me an output array of length 150. Later using the obtained array to get the maximum likelihood for petal length and petal width separately.

Plot of Likelihood of Petal length and Petal width is as follows :

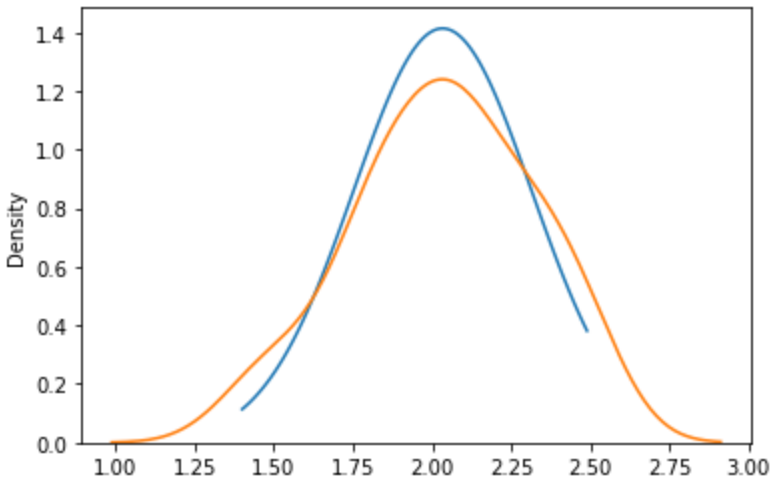


After that Plotting distribution of petal width for every class

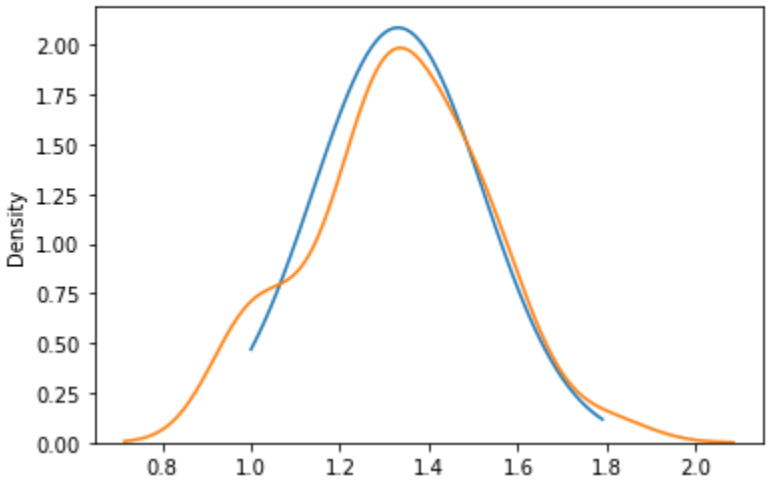
```
plotDistributions(X_train, y_train, 'petal width', 0)
```



```
plotDistributions(X_train, y_train, 'petal width', 1)
```

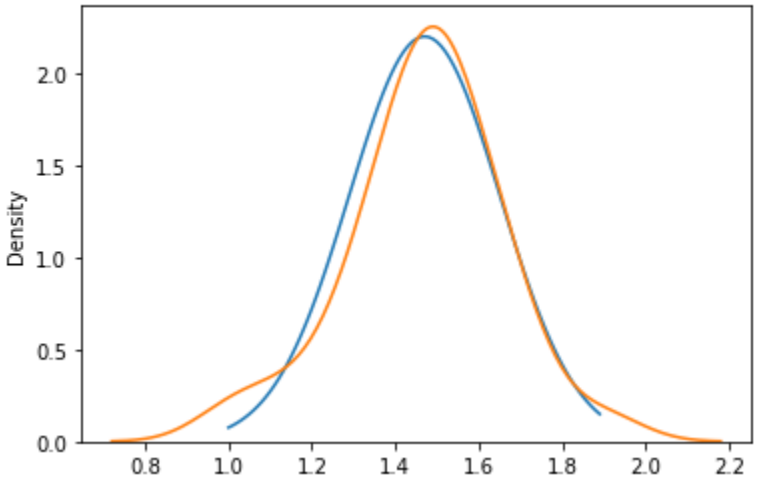



```
plotDistributions(X_train, y_train, 'petal width', 2)
```

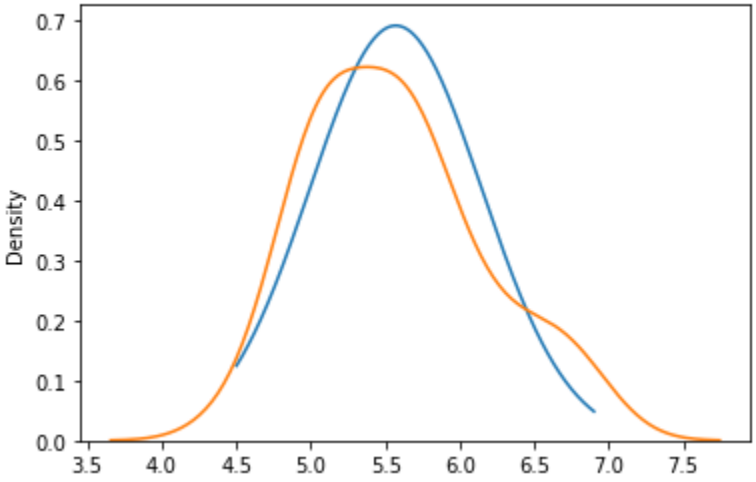


Similarly, for petal length

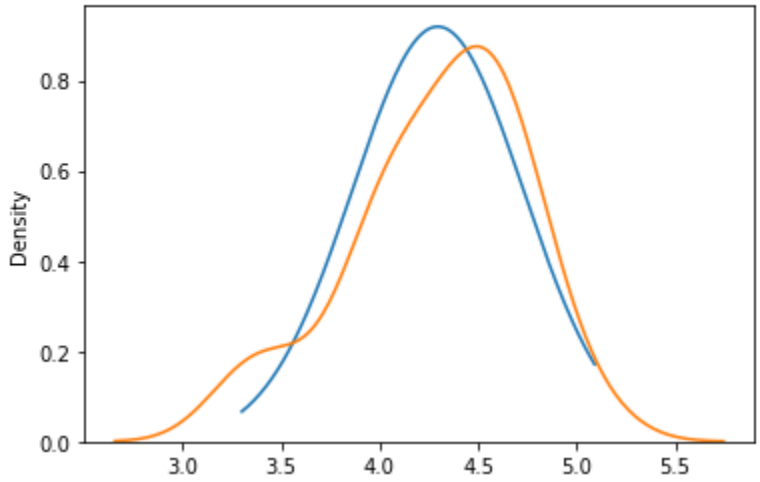
```
plotDistributions(X_train, y_train, 'petal length', 0)
```




```
plotDistributions(X_train, y_train, 'petal length', 1)
```



```
plotDistributions(X_train, y_train, 'petal length', 2)
```



Question 3

For Question 3, it was a text classification using Naive Bayes based problem, all the preprocessing needed in text classification was already done in the data given to us.

I start with importing train.data and train.labels and add columns id's : docId, termId and count. We get a dataframe of following dimensions :

1467345 rows × 3 columns

Then by using the unique values in termId, we obtain a list of total unique words in the provided bag of Words.

Then there is a counter function which plays a major role while computing likelihood, which is further computed with the help of a dictionary. After obtaining the likelihood, we go to the next step which is Laplace smoothing, in which I smooth out likelihoods obtained above.

Repeating the first step and this time importing test.data and test.labels, and again passing through the counter.

And as observed it gives the same result as the above method that is :

Accuracy score of : 0.7846768820786143

```
: accuracy_score(labels, y_pred)

: 0.7846768820786143
```