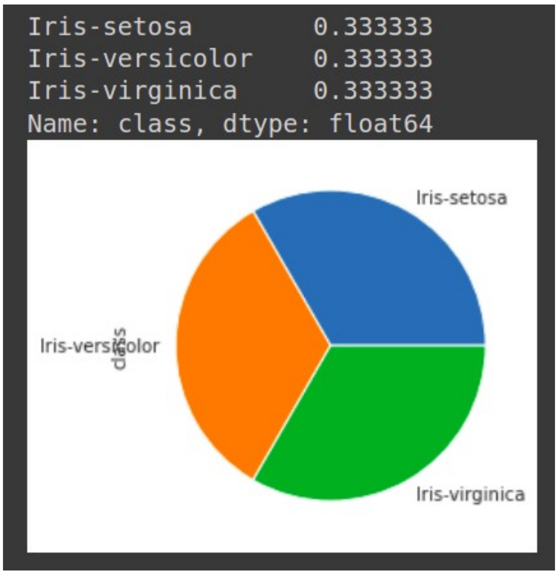
Lab 5
Abu Shahid
B20CS003

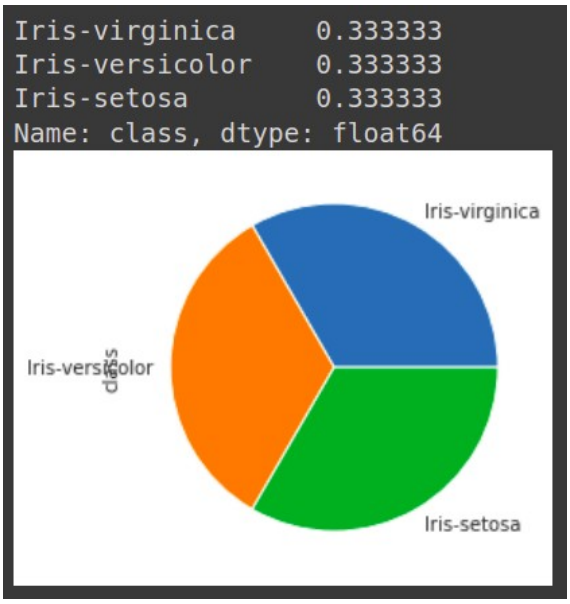
Question 1

Subpart 1: Preprocessing and Vizualization

- No data was missing and no special preprocessing was required.
- To insure all classes are present in test data approximately the same number, we perform Stratified Shuffle Split, with test_size=0.3

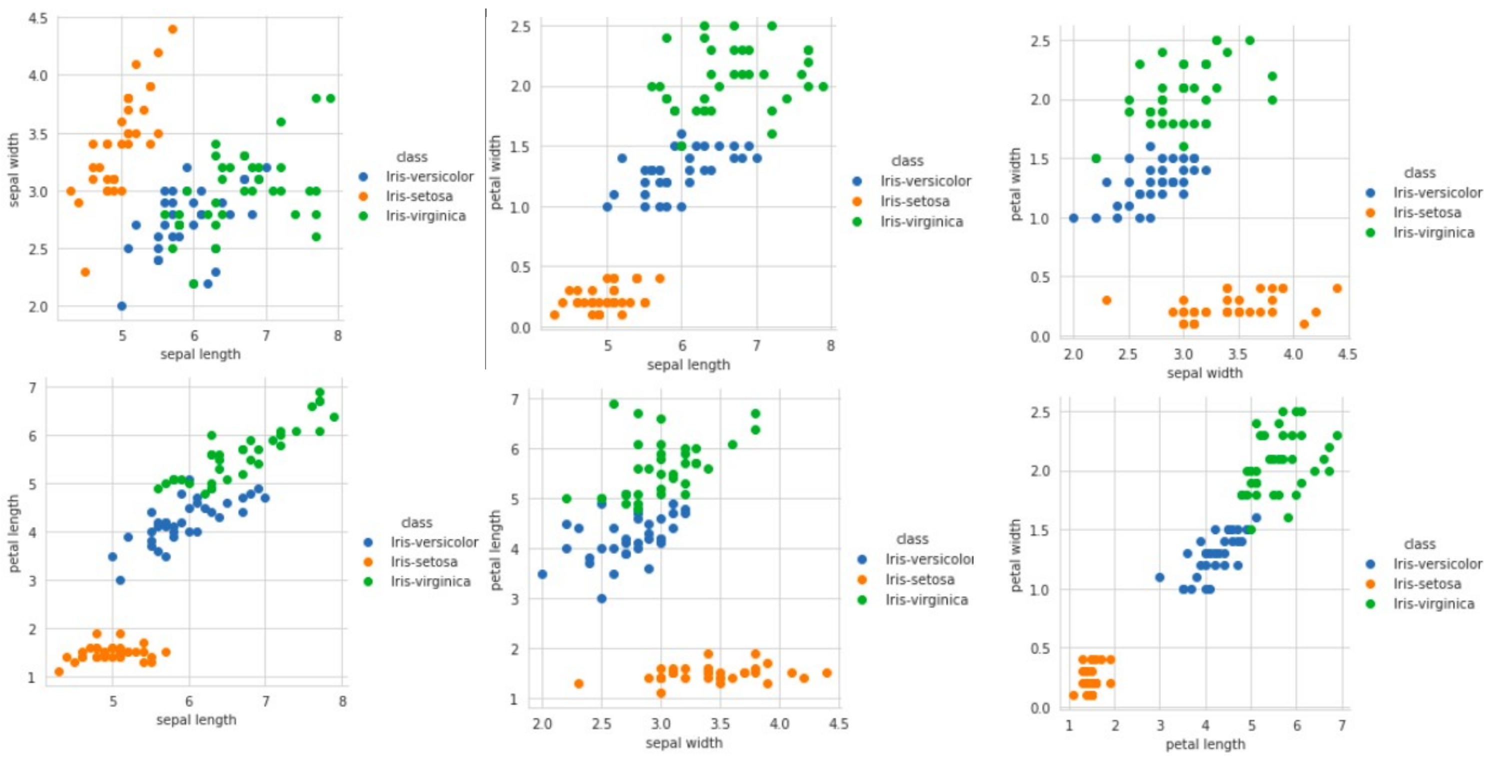


Distribution of dataset



Distribution of test_set

- Next we plotted all the features against each other, taking one pair at a time.



Subpart B, C and D: Feature selection , reporting and Visualization with QDA

- Looking the the plots above, we can clearly see the linear boundaries can be drawn very easily for pairs out of:
  - `> petal length vs petal width;`
  - `> petal length vs sepal length;`
  - `> petal width vs sepal length;`

- These features ensure least misclassifications; therefore for our selection we chose petal length, petal width and sepal length.
- Thereby, taking 2 features at a time QDA was trained:
- Means and corresponding covariance matrices can be reported as:

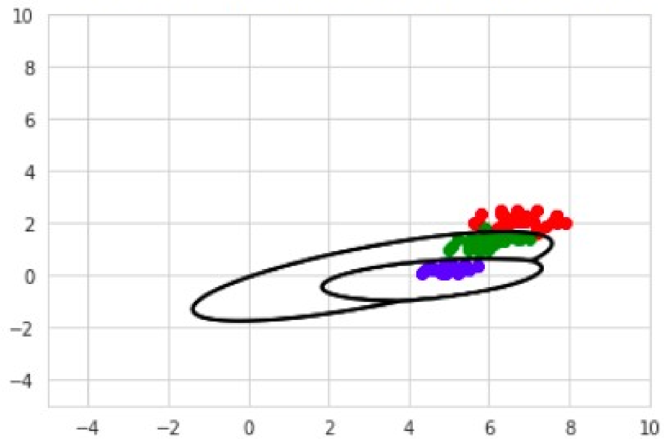
```
for features:  ['sepal length', 'petal width']
[[4.98857143 0.23714286]
 [5.94857143 1.30857143]
 [6.68285714 2.06857143]]

 [[0.10633613 0.01308403]
 [0.01308403 0.01005042]]

 [[0.43734454 0.0347395 ]
 [0.0347395  0.0657479 ]]

 [[0.43734454 0.0347395 ]
 [0.0347395  0.0657479 ]]
```

```
for features:  ['sepal length', 'petal length']
[[4.98857143 1.48857143]
 [5.94857143 4.23714286]
 [6.68285714 5.63142857]]

 [[0.10633613 0.00868908]
 [0.00868908 0.02339496]]

 [[0.43734454 0.33584874]
 [0.33584874 0.33221849]]

 [[0.43734454 0.33584874]
 [0.33584874 0.33221849]]
```

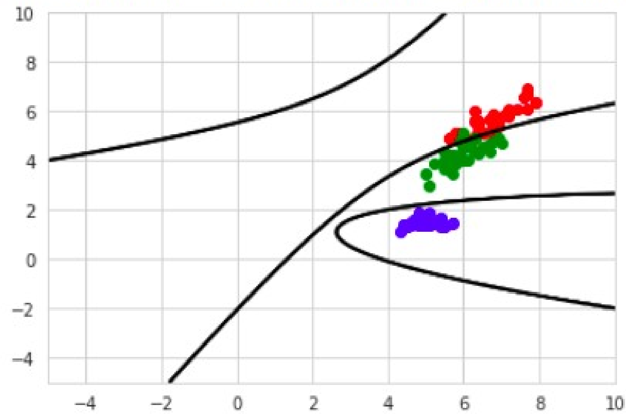
```
for features:  ['petal length', 'petal width']
[[1.48857143 0.23714286]
 [4.23714286 1.30857143]
 [5.63142857 2.06857143]]

 [[0.02339496 0.00337815]
 [0.00337815 0.01005042]]

 [[0.33221849 0.0492521 ]
 [0.0492521  0.0657479 ]]

 [[0.33221849 0.0492521 ]
 [0.0492521  0.0657479 ]]
```

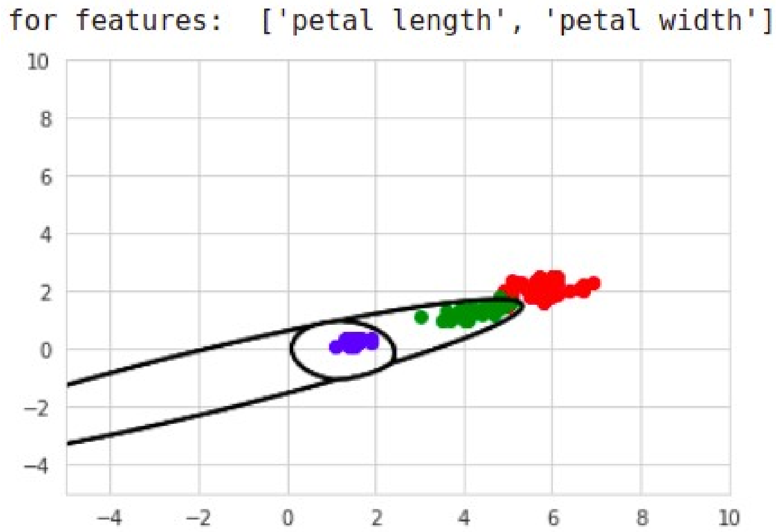
- Decision boundaries can be visualized as:

for features: ['sepal length', 'petal width']



for features: ['sepal length', 'petal length']





Subpart E: Testing QDA

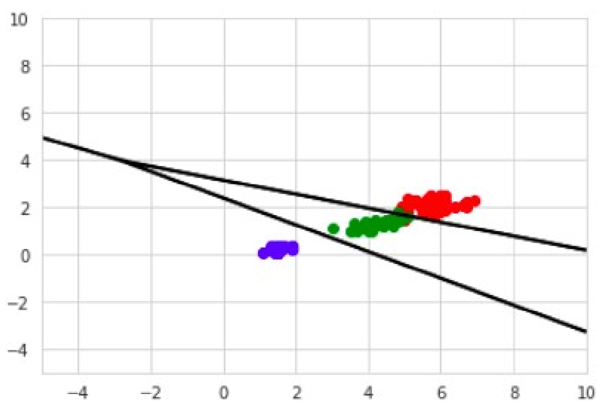
- Prediction of test data was done and classwise accuracy can be reported as:

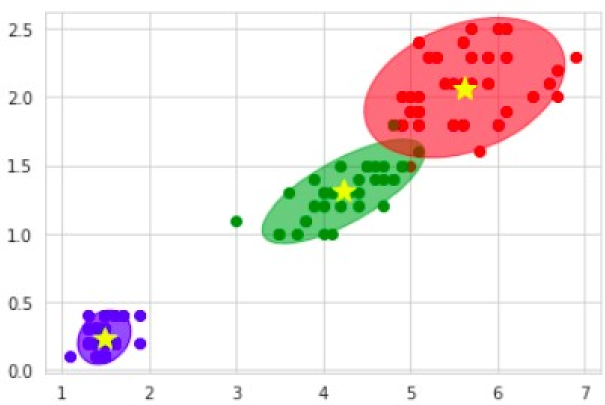
- `for features: ['sepal length', 'petal width'] 0.90`
- `for features: ['sepal length', 'petal length'] 0.9555555555555556`
- `for features: ['petal length', 'petal width'] 0.9666666666666667`

- Petal length v/s petal width gives best accuracy as in biology it is the proportion that is preserved in the various sample of a species. Since petal length is heavily corelated to petal width; it will be giving the best accuracy.

Subpart F, G and H: Training, Plotting and testing LDA

- Here we train an LDA model using petal length and petal width.
- Decision boundary and gaussian curve can be depicted as:



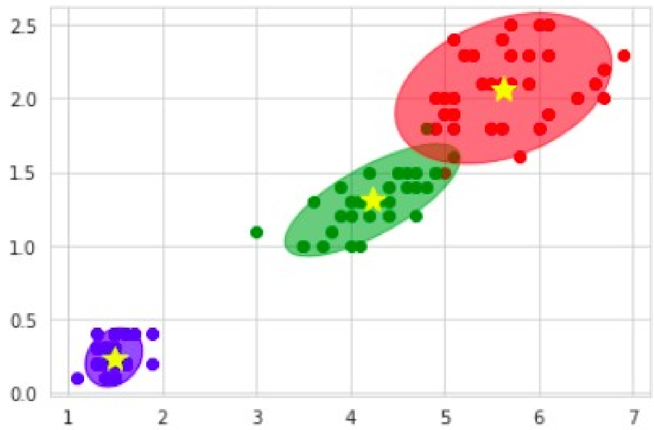


- With LDA accuracy of the classification for features petal length and petal width was found to be: 91.1111111111%
  - compared to 96.66666666666% given by QDA.
- Due to strictly linear boundaries, LDA has substantially lower variance.

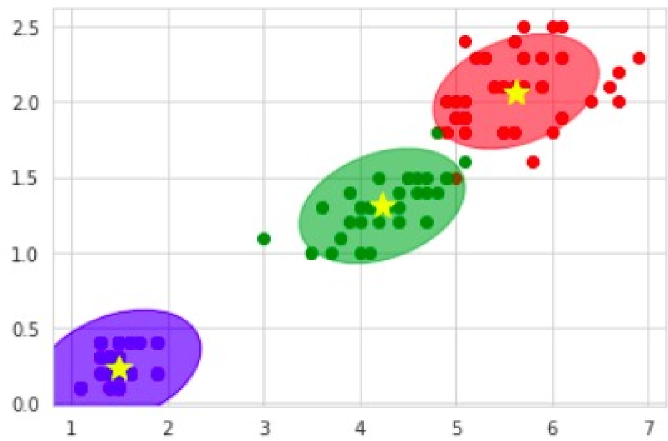
- In layman terms, LDA will find it difficult to classify data that is not linearly separable.

Subpart I: Visualizing LDA and QDA

- Lastly we plotted Gaussian Distributions of our LDA and QDA, given the features petal length v/s petal width.



Plot for QDA



Plot for LDA

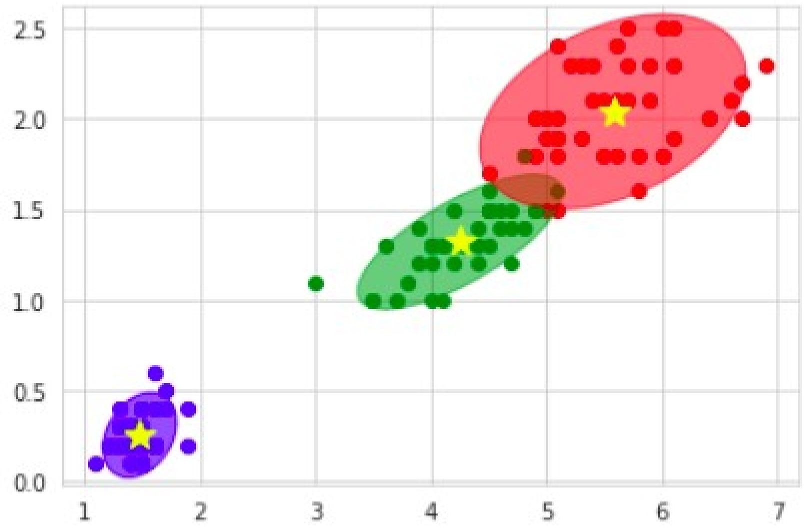
- **Note*:** Accuracy can be pushed further and better decision boundaries can be further obtained from parameter tuning. However, no attempt was made it it was simply not demanded.

Question 2:

Subpart A: Calculation and Visualization

- In this subpart, from the given data we calculated the sample mean and covariance of each class and using them; we plotted the Gaussian distribution of the same.

```
mean of class 1 [1.48, 0.25249999999999984]
mean of class 2 [4.2524999999999995, 1.3199999999999998]
mean of class 3 [5.58, 2.0399999999999987]

 covariance of class 1              petal length  petal width
petal length       0.025744     0.005692
petal width        0.005692     0.013840

 covariance of class 2              petal length  petal width
petal length       0.196404     0.060462
petal width        0.060462     0.034974

 covariance of class 3              petal length  petal width
petal length       0.331897     0.058769
petal width        0.058769     0.072205
```



Plot of gaussian distribution

Subpart B: Computing Likelihood

- This was done by using the formula:

$$\mathcal{N}(\underline{x}\ ;\ \underline{\mu}, \Sigma) = \frac{1}{(2\pi)^{d/2}} |\Sigma|^{-1/2} \exp\left\{-\frac{1}{2}(\underline{x}-\underline{\mu})\Sigma^{-1}(\underline{x}-\underline{\mu})^T\right\}$$

- Implementation of this was rather straightforward.
- Here we initialize a function to calculate likelihoods of data given parameters mean and covariance matrix ***assuming normal distribution.***

Subpart C: MLE

- Here we define a function to perform MLE.
- In a random sampling of N observation vectors $\boldsymbol{x1}$, $\boldsymbol{x2}$, … , $\boldsymbol{xN}$ from Np (μ, Σ), the sample mean vector $\bar{x}$= 1 N ∑ $x\alpha$ N α=1 is the maximum likelihood estimator of μ and Σ = 1 N ∑ ($x\alpha$ − $\bar{x}$) ($x\alpha$ − $\bar{x}$)' N α=1 is the maximum likelihood estimator of Σ.

Reference: https://www.bbau.ac.in/dept/Statistics/TM/Estimation%20of%20Mean%20Vector%20and%20Variance%20Covariance%20Matrix.pdf

- Having defined our function we calculate MLE over training dataset to determine mean and covariance and classify datapoints using Bayes Classifier.
- Parameters obtained for different classes can be summarized as:

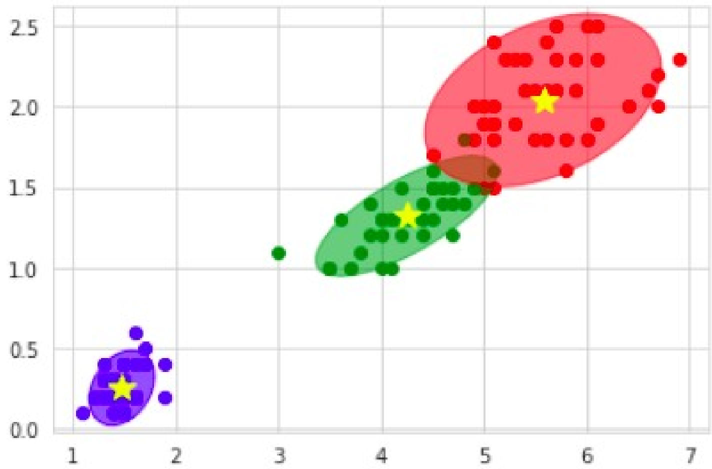
```
mean of class 1

 petal length    1.4800
petal width     0.2525
dtype: float64
mean of class 2

 petal length    4.2525
petal width     1.3200
dtype: float64
mean of class 3

 petal length    5.58
petal width     2.04
dtype: float64
```

```
covariance of class 1

[[0.0251      0.00555    ]
[0.00555     0.01349375]]

covariance of class 2

[[0.19149375 0.05895    ]
[0.05895     0.0341     ]]

covariance of class 3

[[0.3236 0.0573]
[0.0573 0.0704]]
```

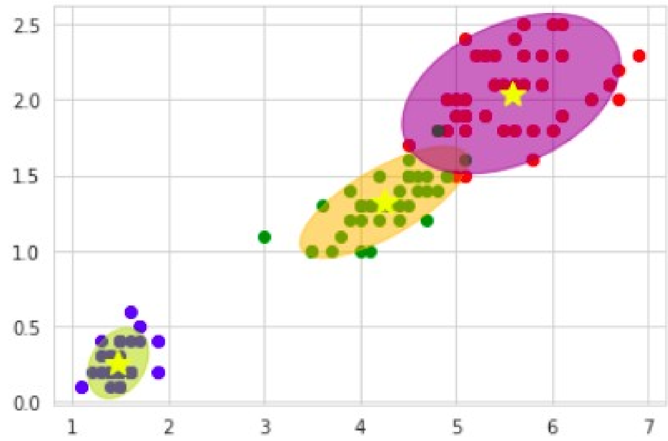
- Accuracy on training dataset was reported to be 96.666%

Subpart D and E:

- Visualization of the above parameters is as:



Plot from QDA



Plot from MLE

- Testing MLE on test set gave an accuracy of 93.3333% compared to 96.6666 with QDA.

Question 3

Subpart A: Compute Likelihood

- Apparently the data file given to us was a sparse matrix. And converting it to a dense matrix was not an option.
- Consequently, a mapping was done from the train.data to train.label and test.data to test.label
- We then calculated priors by using this mapping, by standard formulae.
- Next, using the vocal file given to us, and knowing that we can have 20 different classes; we initialized a likelihood matrix of size |number of terms| **x** 20.
- Next we iterate through all the documents that are labelled c (where c varies from 1 to 20), and calculate the probabilities of all the words across all the classes.

Subpart B: Laplace Smoothing

- On inspecting the likelihood matrix, we found that out of 1223760 entries (11269 documents **x** 61188 terms) , 1022982 entries were zero. This was due to absence of a lot of terms from different documents. This can lead to the posterior calculation yielding a 0 probability.
- So that this does not happen, we apply a technique called Laplace smoothing.

- An empirical fact about language:
- A small number of events occur with high frequency
- A large number of events occur with low frequency

- This was implemented by adding a non-zero term alpha to the count of numerator while calculating likelihood.

Subpart C: Naive Bayes Classification:

- For our final subpart, we defined a function that utilizes the priors and smothened likelihoods that we calculated in the previous parts to return the probabilities of a new document beloging to the 20 different classes.
- Index of Maximum probabilty given was taken as the label of the predicted document.
- For initialization purpose, all predictions were initialized to be belonging to class 1 so when calculating accuracy, we only dealt with those predictions that did not yield in ‘class 1’.
- This model was then evaluated and accuracy was reported to be:

```
accuracy of Naive Bayes 0.7083333333333334
```