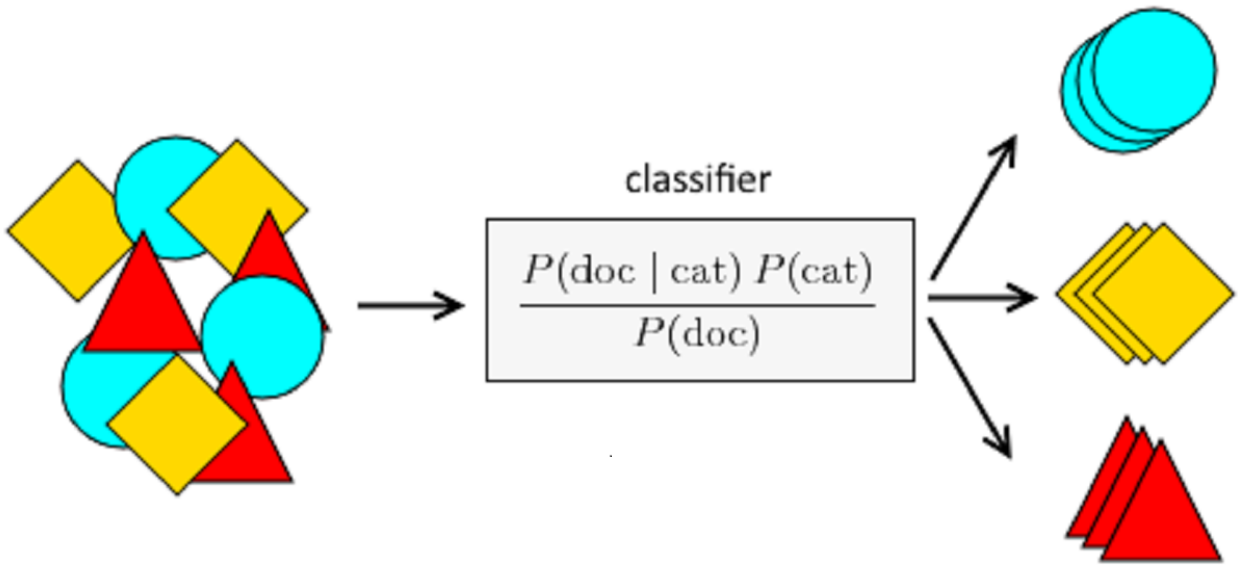
Pattern Recognition and Machine Learning

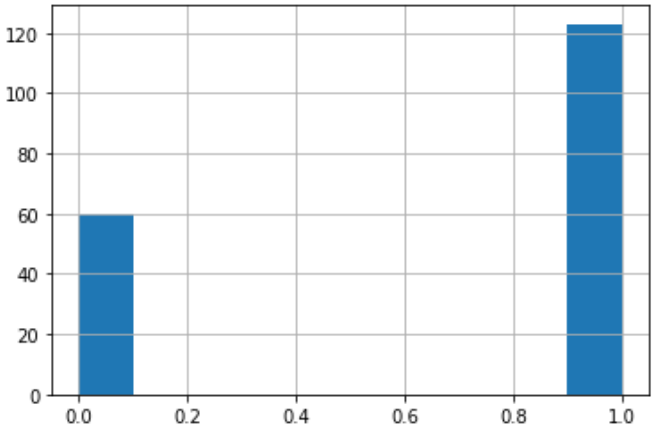
Lab 4 Report (Kartik Choudhary, B2CS025)

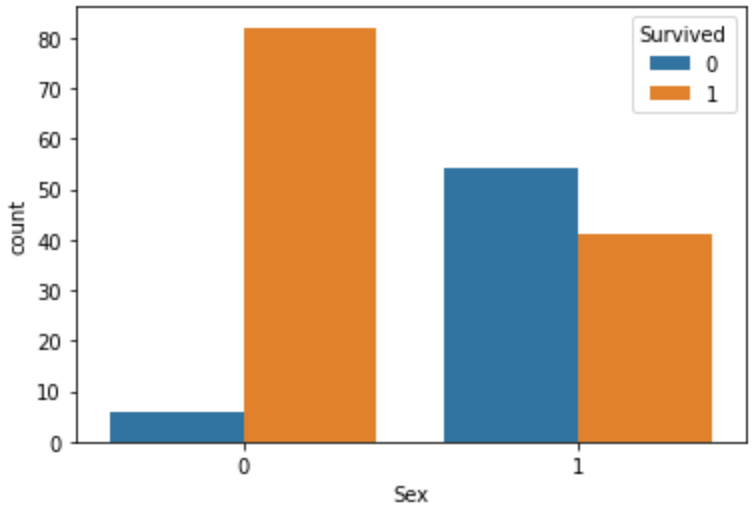
Bayes Classification

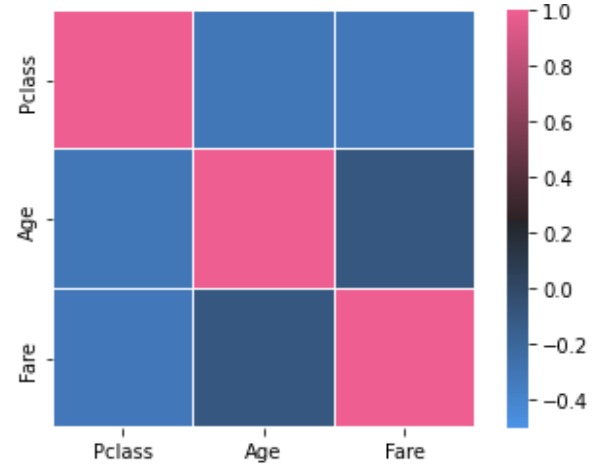


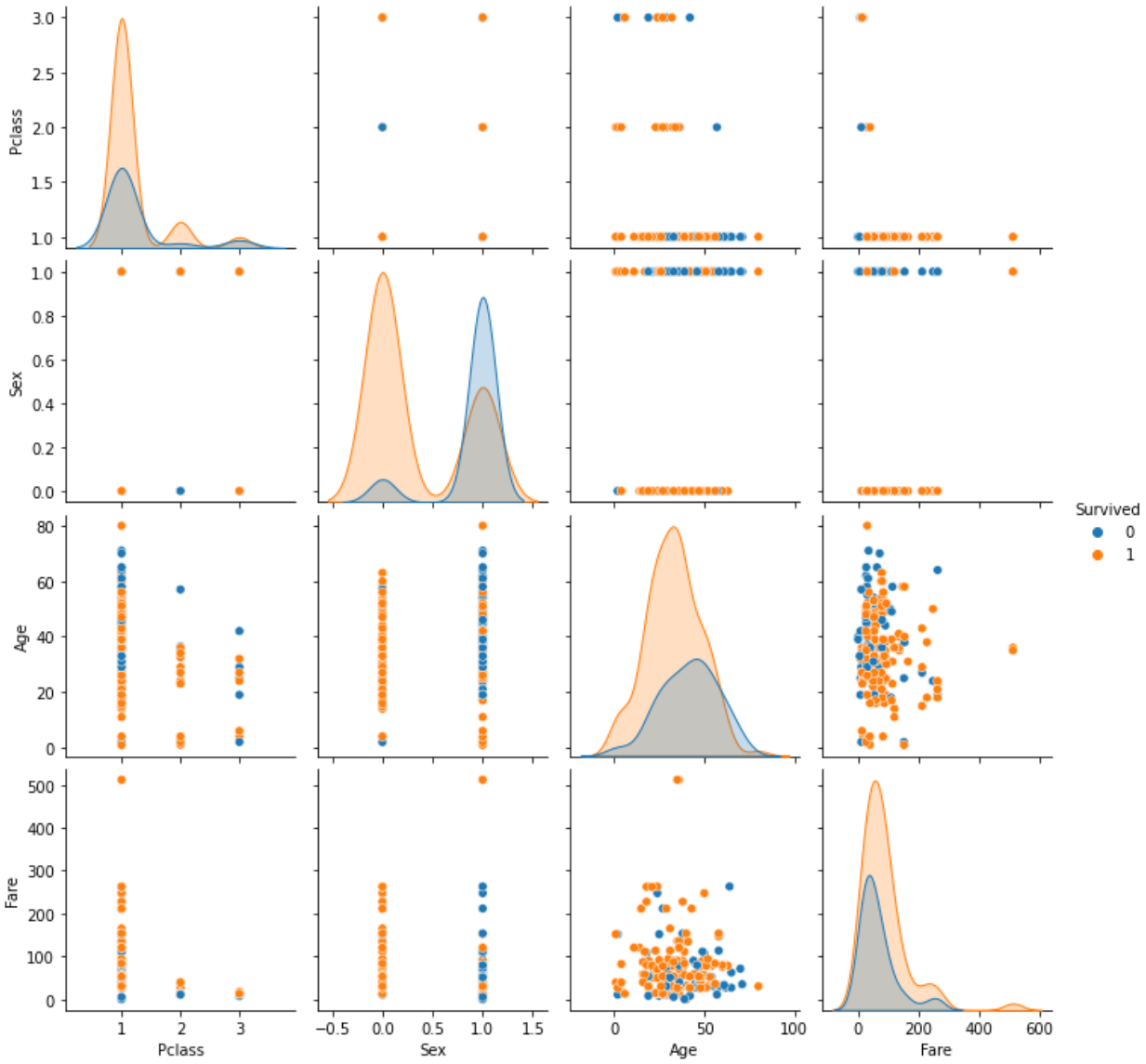
Question 01: TITANIC DATASET

1. Data Preprocessing was done on the dataset to make it fit for the naive Bayes classifier. PassengerId,Name,Ticket,Cabin,Embarked were dropped. Test trains were split into 20:80 ratios. Dataset was visualized as,

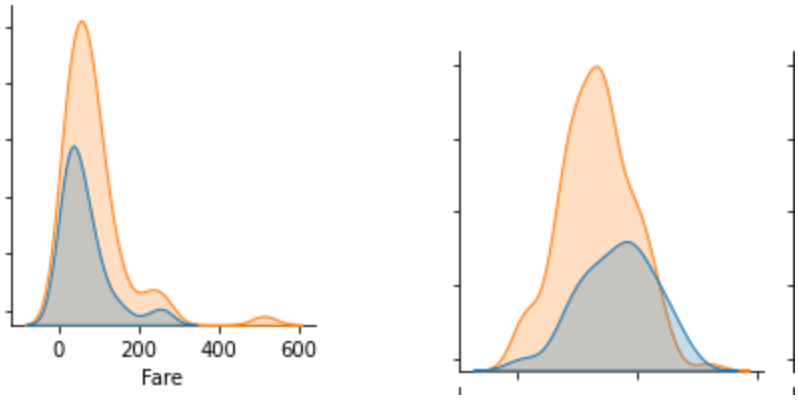



Survived
0
1
count
80
70
60
50
40
30
20
10
0
0
1
Sex


Pclass
Age
Fare
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4


Pclass
Sex
Age
Fare
Survived
0
1

2. Choosing variant - Density plots for the continuous columns were plotted, which can be seen below: For the Fare, Age, columns: As we can see, the 'fare' column was fairly related to the 'survived' column. The 'age' and 'fare' columns were plotted which gave us an idea about their distribution. Their distribution was related to the Gaussian distribution more than any other distribution

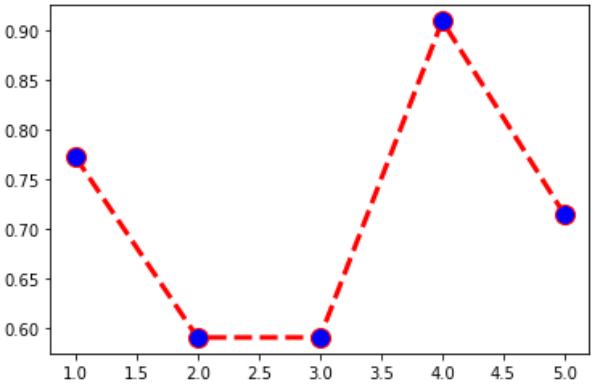


3. The Gaussian Naive Bayes classifier was implemented from scratch as. A function for Gaussian distribution was implemented which handles the continuous columns.
4. 5-fold cross-validation was performed on the whole train dataset and the resul

```
Cross-validation scores:[0.77272727 0.59090909 0.59090909 0.90909091 0.71428571]
Average cross-validation score: 0.7156
```

Average accuracy of 71.56 percent was achieved. Maximum accuracy was achieved at the 4th fold and the minimum was found at the 2nd and 3rd fold.

5. Visualization of results across cross validation sets:



the probability of the top class for each row in the testing dataset.

```
np.array(prob)
```

```
array([9.76676964e-07, 3.82650250e-05, 9.18433537e-07, 3.74457247e-05,
       2.54022181e-05, 6.17404716e-06, 1.01564907e-09, 1.15333338e-08,
       4.60794176e-07, 8.83881588e-06, 1.33152049e-05, 3.99406345e-06,
       2.38297151e-05, 1.25829852e-05, 1.29568990e-07, 1.03339058e-06,
       8.54386902e-07, 2.42181030e-05, 1.09462364e-05, 1.80159446e-06,
       3.41351230e-05, 1.23565974e-05, 2.86750651e-07, 9.07868885e-06,
       3.43192355e-07, 2.83205183e-08, 4.88543077e-09, 3.26615958e-06,
       3.50246385e-05, 1.13540574e-05, 5.01765351e-06, 6.27153039e-07,
       1.89508921e-05, 1.17197063e-05, 1.58319966e-05, 1.07392648e-05,
       3.36912826e-05, 1.03833173e-05, 3.89464183e-07, 1.99150847e-06,
       3.42952503e-07, 9.84016828e-06, 2.52697564e-05, 6.58815710e-06,
       3.11502580e-07, 1.87777174e-05, 3.54077425e-05, 1.48561113e-05,
       1.98168330e-05, 1.26329525e-05, 1.61590407e-05, 4.42347664e-06,
       4.73672679e-07, 2.77778330e-05, 1.42556403e-05, 2.65213535e-06,
       2.01943922e-05, 4.52362887e-06, 7.28383470e-07, 1.84844746e-05,
       6.61083968e-07, 3.91759051e-07, 8.32674939e-07, 4.80582249e-07,
       8.42479626e-07, 2.60138409e-05, 2.42534534e-07, 8.37009080e-06,
       6.87234078e-10, 8.16810721e-09, 3.26161641e-05, 1.23750734e-05,
       6.11468125e-07, 3.66045672e-05, 8.32529153e-07, 3.93045864e-05,
       3.35562388e-05, 1.55269090e-05, 8.88921162e-07, 3.98454008e-05,
       3.34571596e-05, 1.29076848e-05, 3.53585694e-05, 1.48224001e-05,
       1.21313738e-08, 8.25423504e-08, 1.23785591e-05, 2.22624453e-06,
       7.34618111e-07, 3.82495519e-05, 2.77422712e-05, 7.83033461e-06,
       2.29845749e-05, 9.28296188e-06, 2.10782238e-07, 7.85527356e-06,
       5.07624127e-07, 3.30126175e-05, 3.13343103e-05, 1.29927827e-05,
       3.50671065e-07, 6.03700713e-06, 3.36143648e-07, 2.33458429e-05,
       1.39670871e-05, 3.96231087e-06, 1.70770637e-08, 2.69215227e-06,
       1.60628875e-08, 3.56744525e-06, 5.28737715e-07, 9.23114055e-07,
       2.38490048e-07, 1.93169980e-05, 7.76569157e-07, 3.65083499e-05,
       5.26876682e-07, 3.13741079e-05, 3.28206706e-09, 2.42251742e-06,
       2.45911832e-07, 2.46689555e-05, 5.55313198e-07, 3.09044973e-05,
       5.19199590e-07, 3.28258726e-05, 3.77977273e-07, 9.84141015e-06,
       2.72509263e-07, 1.78396047e-05, 3.06274861e-05, 9.81648581e-06,
       1.41612490e-05, 1.11100656e-05, 2.75206807e-05, 1.48152629e-05,
       5.56203360e-07, 3.00358996e-05, 3.25578927e-05, 1.05726071e-05,
       7.09000746e-07, 4.15696948e-07, 5.25444054e-09, 3.43064185e-06,
       2.18719364e-07, 7.79313790e-06, 4.69958497e-07, 2.68059059e-05,
       4.58125786e-07, 8.74697682e-06, 5.48752770e-07, 3.09606349e-05,
       2.16569287e-05, 1.25427776e-05, 1.91438791e-06, 4.62077986e-06,
       2.42993353e-05, 1.45578875e-05, 3.83994603e-07, 2.80781596e-05,
       9.65788408e-09, 7.22703956e-08, 8.10078718e-07, 2.06268612e-05,
```

```
[195] min(prob)

      1.992408105044532e-15

[196] max(prob)

      3.98454007693644e-05
```

6. Scratch implementation was compared with the model imported from sklearn.

   Scratch Implementation-

```
Accuracy:
0.7431693989071039

              precision    recall  f1-score   support

           0       0.57      0.83      0.68        60
           1       0.90      0.70      0.79       123

    accuracy                           0.74       183
   macro avg       0.74      0.77      0.73       183
weighted avg       0.79      0.74      0.75       183
```

   Gaussian Bayes Classifier-

```
Training-set accuracy score using the skicit-learn model is: 0.7615
```

7. Multinomial Bayes classifier was imported and 5-fold cross-validation was performed. Results can be seen below:

```
| accuracy_score(y_test, y2)

 0.6351351351351351
```

Cross Validation Scores

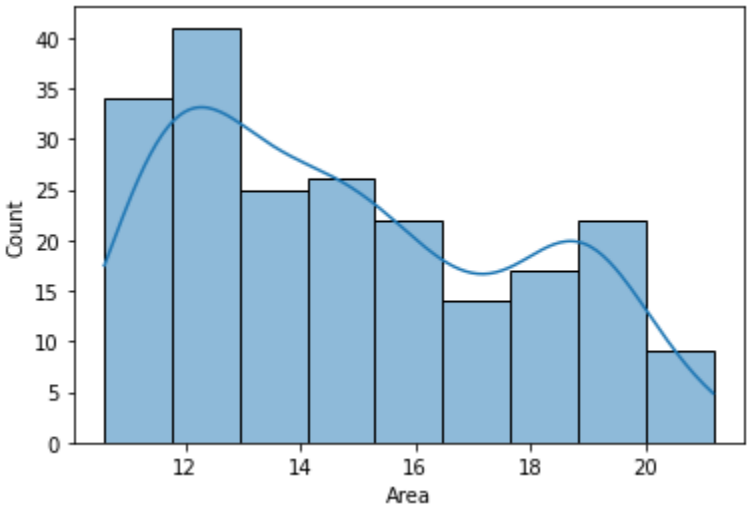
```
array([0.86363636, 0.36363636, 0.63636364, 0.54545455, 0.57142857])
```

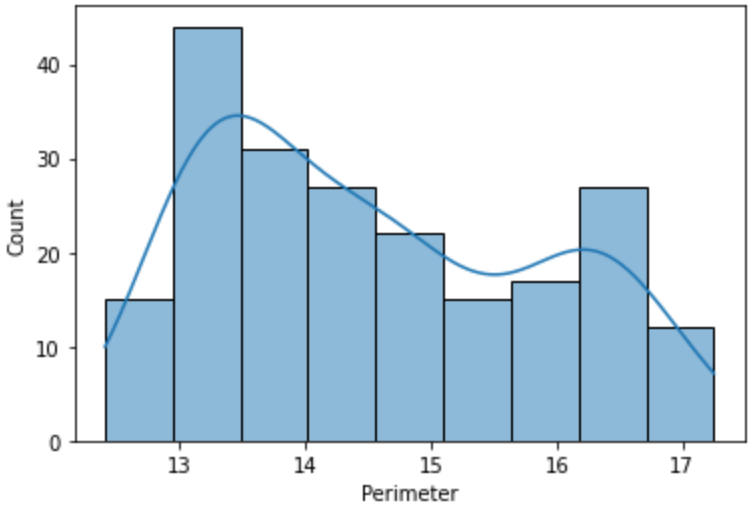
```
Average cross-validation score: 0.5961
```

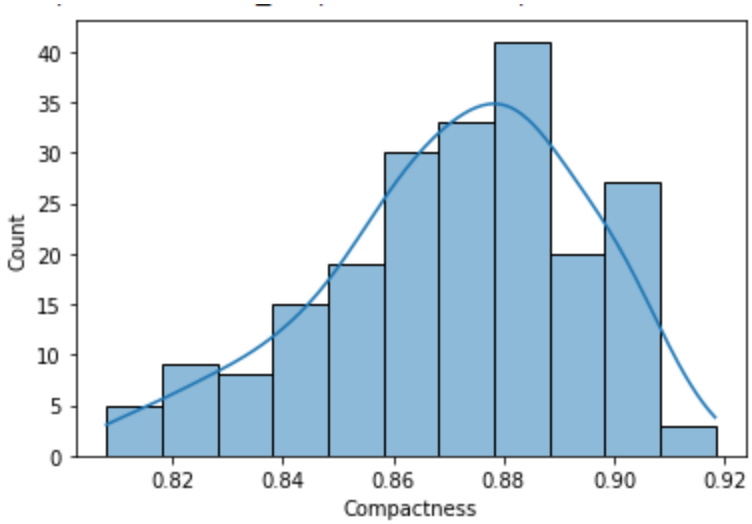
The average accuracy was 59 percent which was far less than the gaussian Bayes classifier. ‘Age’ and ‘fare’ columns were normally distributed and multinomial distribution didn't fit them. This is why we got such poor results on the multinomial Bayes classifier. We got pretty better results than that which can be seen in the above parts.

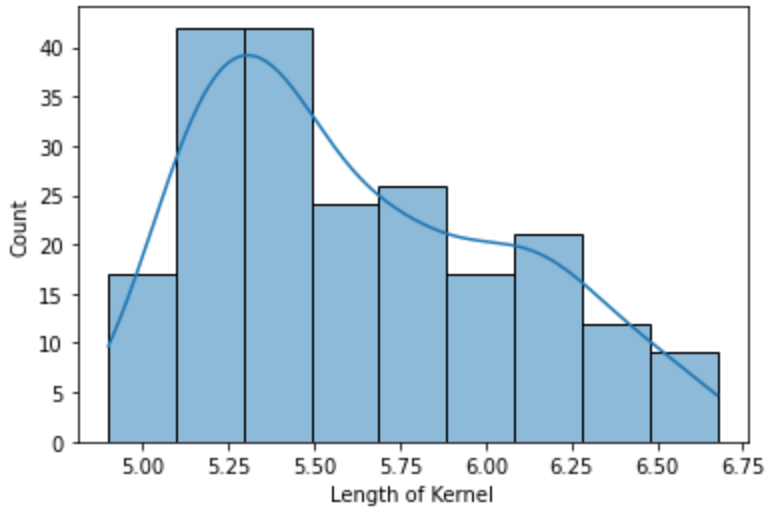
Question 02:

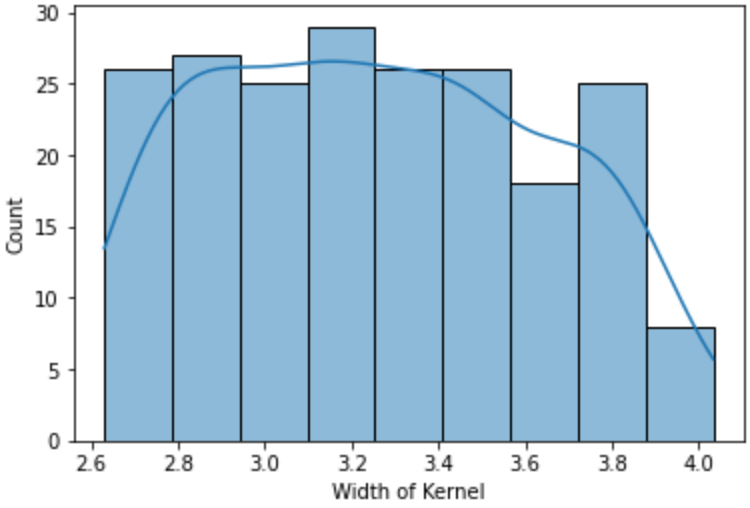
a. Histogram samples:

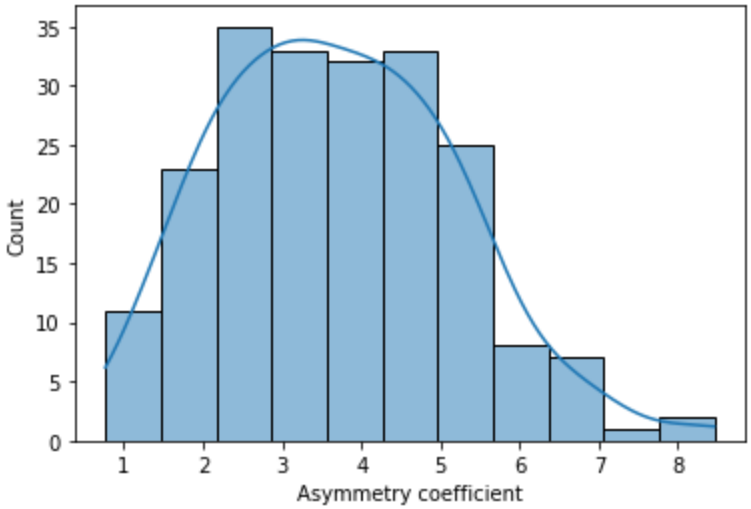

Count
Area
0
5
10
15
20
25
30
35
40
12
14
16
18
20

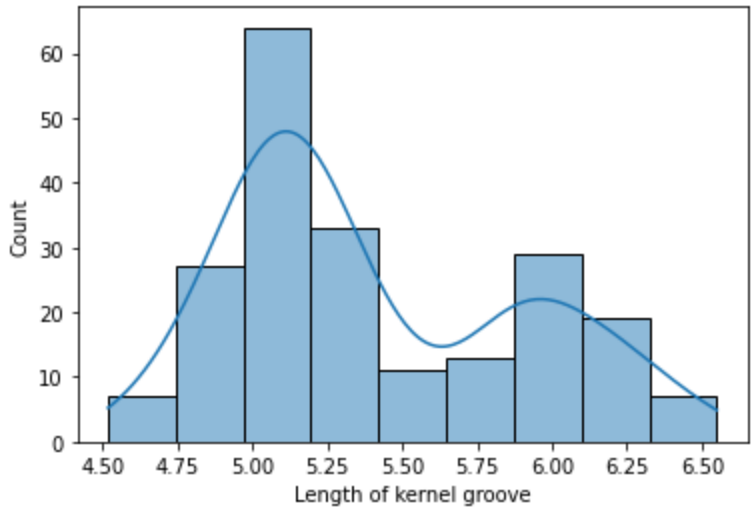

Count
Perimeter
0
10
20
30
40
13
14
15
16
17

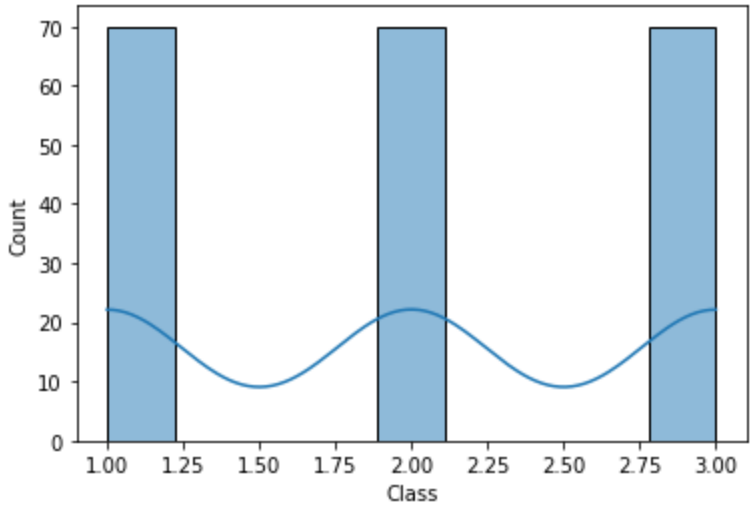

Count
Compactness
0
5
10
15
20
25
30
35
40
0.82
0.84
0.86
0.88
0.90
0.92

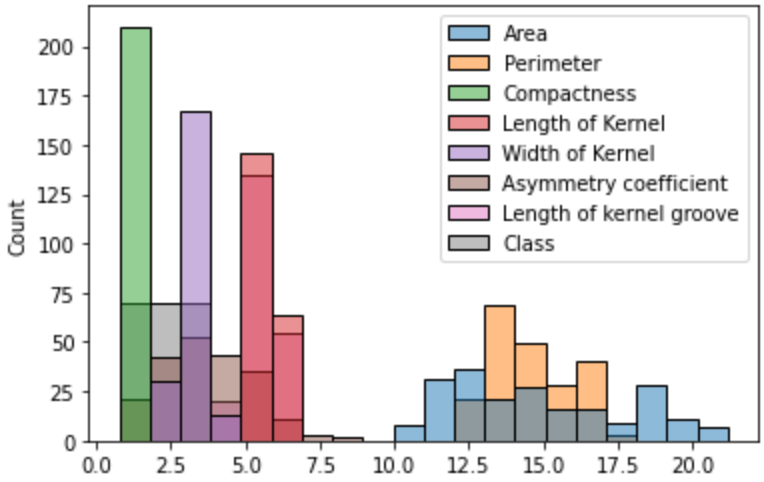

Count
Length of Kernel
0
5
10
15
20
25
30
35
40
5.00
5.25
5.50
5.75
6.00
6.25
6.50
6.75


Count
Width of Kernel
0
5
10
15
20
25
30
2.6
2.8
3.0
3.2
3.4
3.6
3.8
4.0


Count
Asymmetry coefficient
0
5
10
15
20
25
30
35
1
2
3
4
5
6
7
8





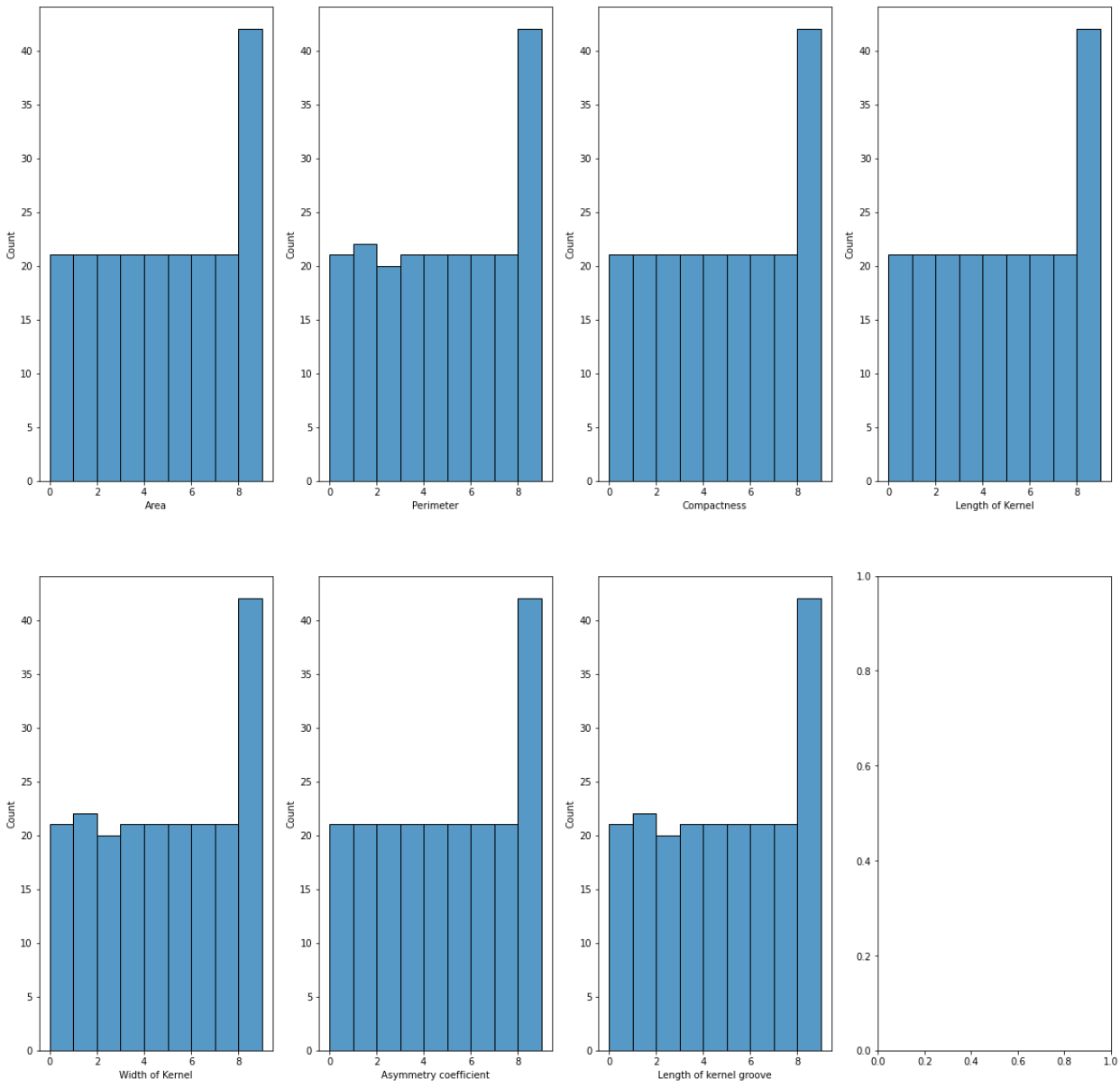


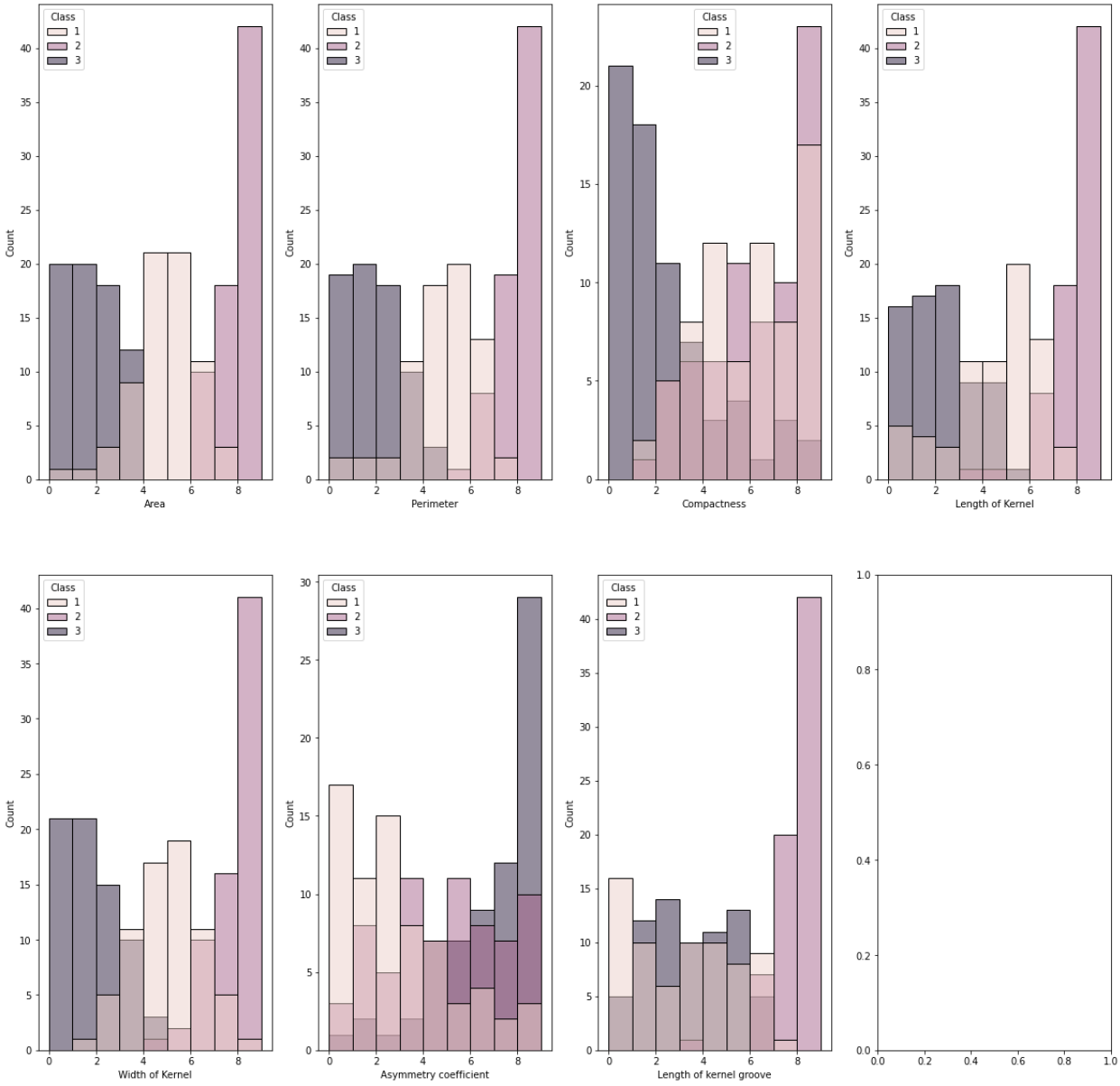
b. Prior probabilities for all the classes were calculated and the results can be seen in the colab notebook.

c. The features were discretized into bins with each bin containing equal samples. Each feature was discretized into 10 bins to make the visualization easier. None of the forbidden libraries were used for this task.

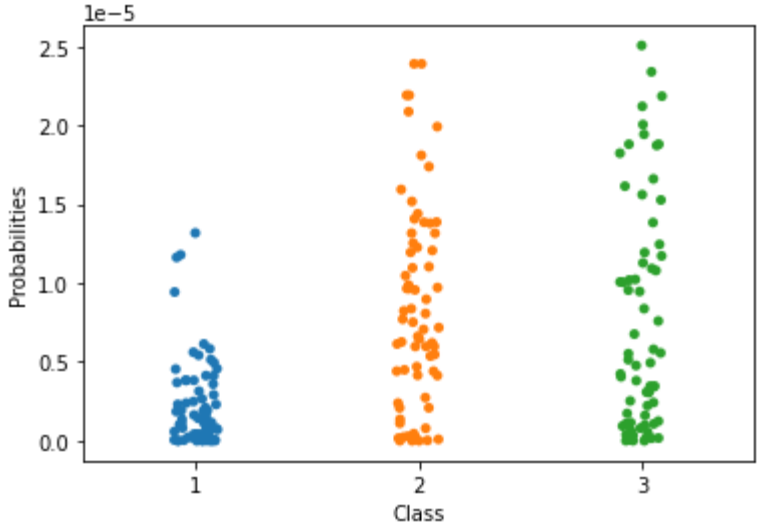
d. Likelihood/Class conditionals were calculated and stored inside a dictionary for faster use. Results can be seen in the colab notebook

e. Plotting the plot required in the e part: (without hue) and with hue as “Class”




Class
1
2
3
Count
Area
Perimeter
Compactness
Length of Kernel
Width of Kernel
Asymmetry coefficient
Length of kernel groove

f. Posterior probabilities were calculated and a strip plot was plotted to analyze the results:



Class 1:- LOW PROBABILITY Value

Whereas the probability is higher for Class 2 and Class 3, if we use the Naive Bayes Classification, it's likely that the model would predict Class 2 and 3 with less error.