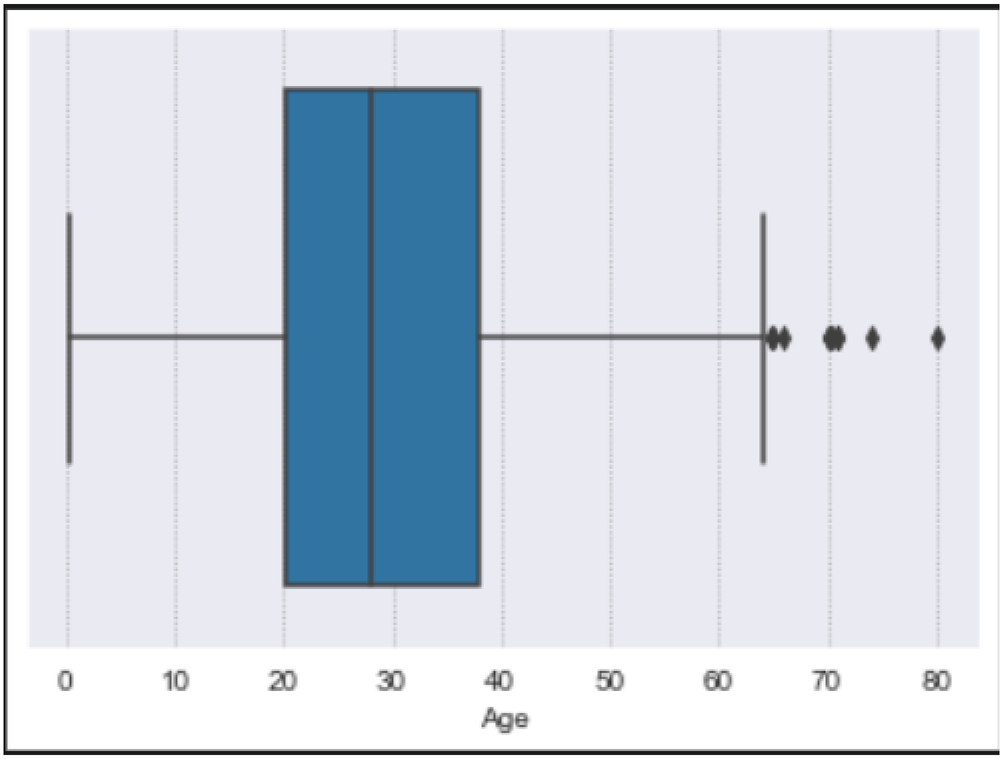
**Pattern Recognition and Machine Learning**
**Lab 4**

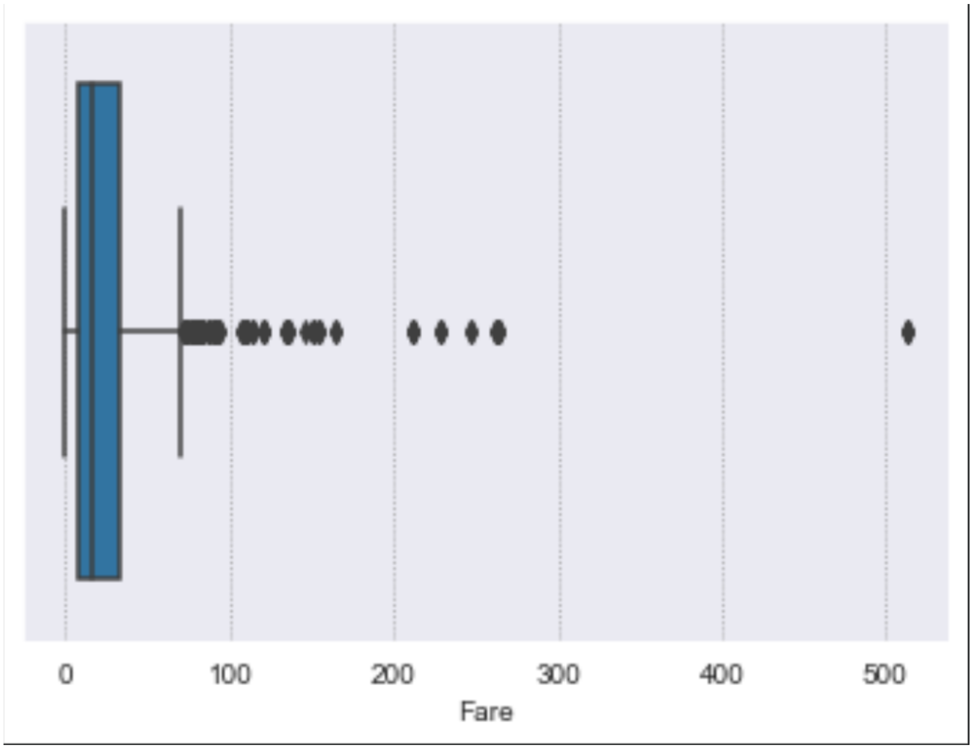
Question 1 - Titanic Dataset

1. Pre-processing - Preprocessing was done on the dataset to make it fit for the naive Bayes classifier. First of all commands like df.head() and df.describe() were used to get an overlook of the data. The ‘Cabin’ column was dropped because it contained more than 70 percent null values. Then the ‘Ticket’ and ‘name’ columns were dropped because they didn't contain any relevant information and there was nothing that naive Bayes could do about it. ‘Sex’ column was encoded giving ‘female’ a value of 1 and ‘male’ a value of 0. Box plot for the ‘Age’ column was plotted to find out the outliers.

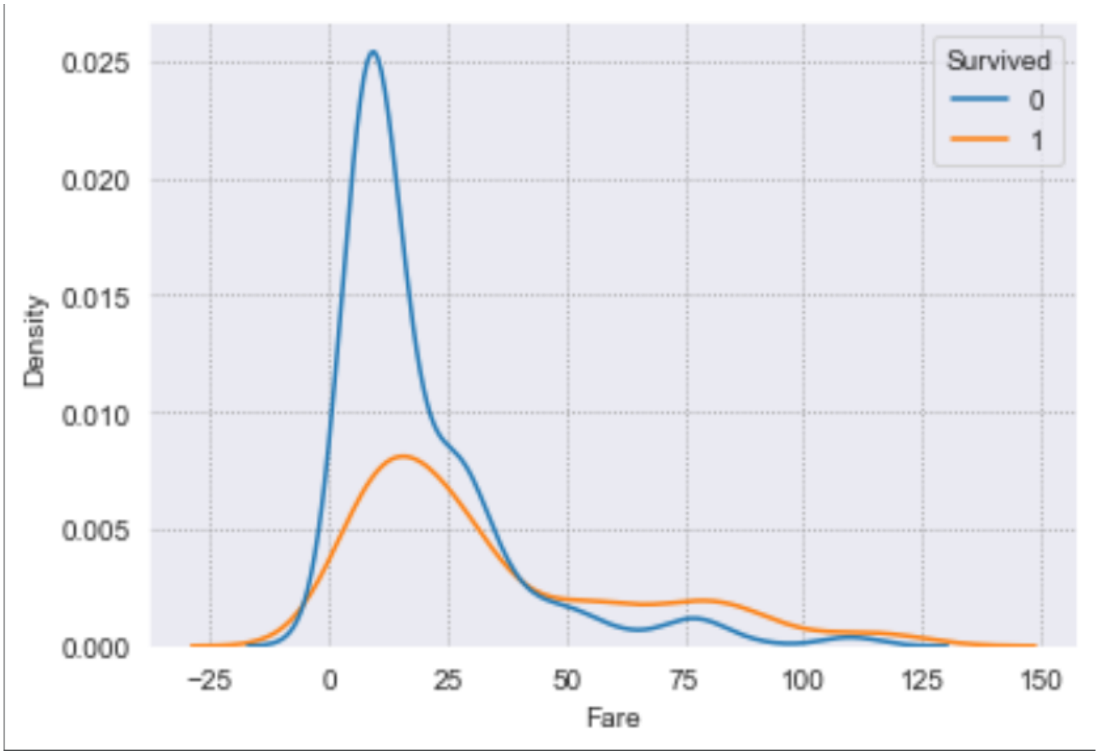


As we can see, some outliers were removed from the dataset. The same kind of procedure was done for the ‘fare’ column.

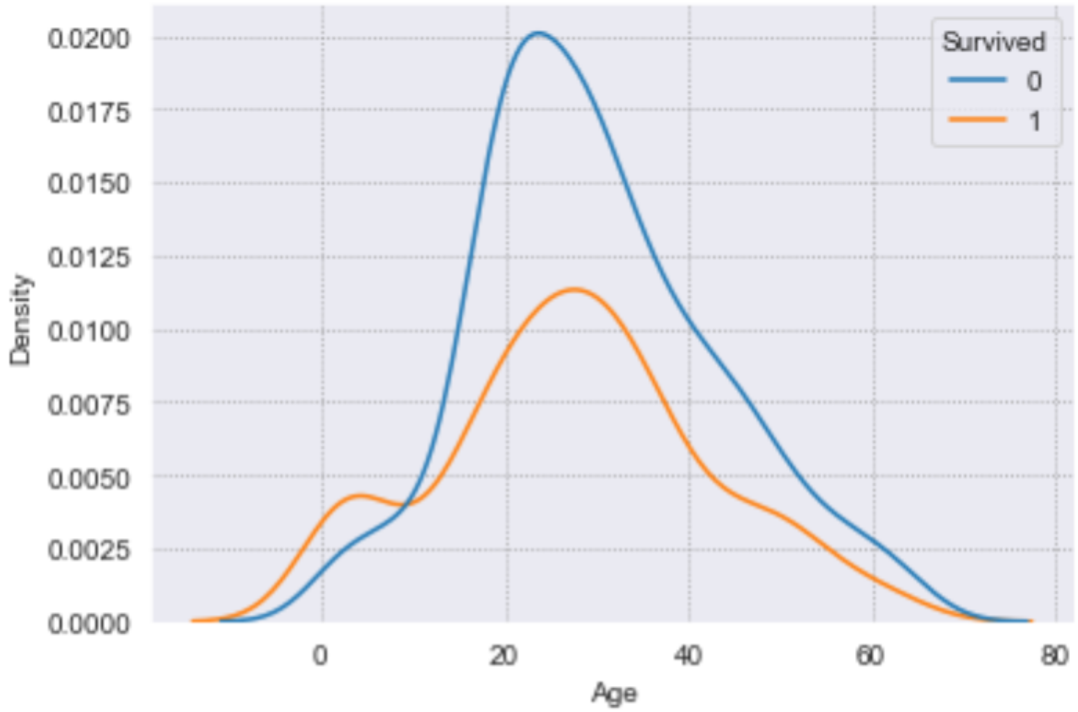
Box plot can be seen below:



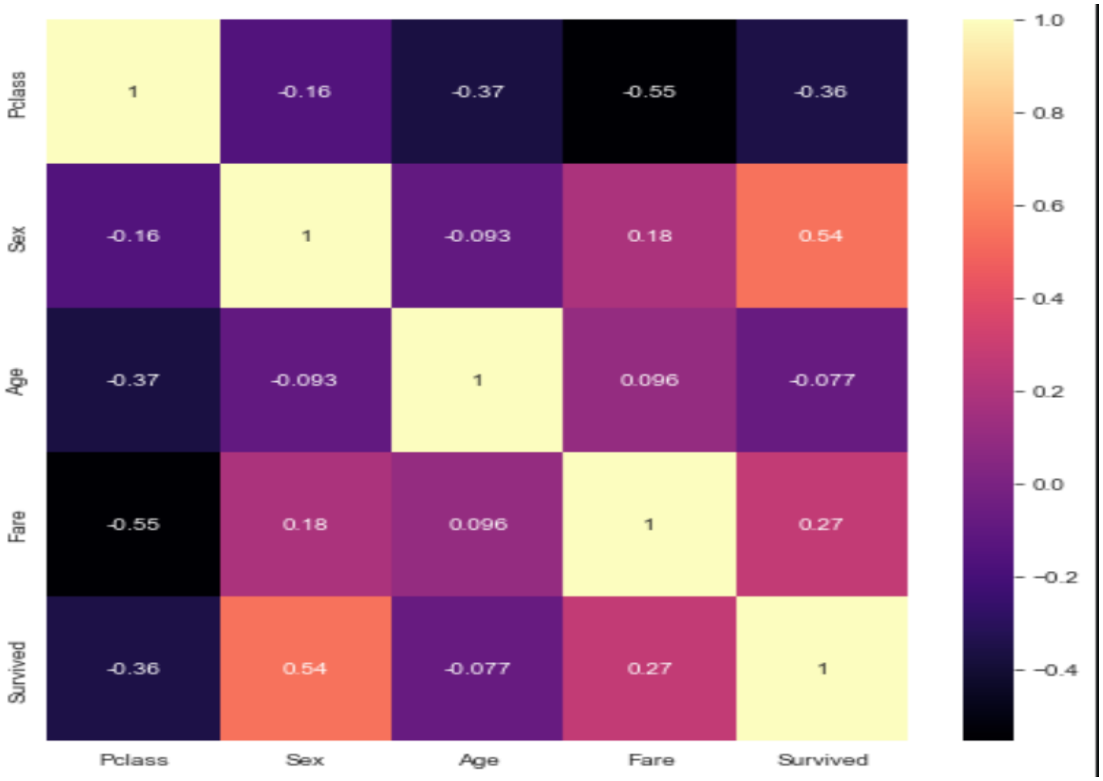
2. Choosing variant - Density plots for the continuous columns were plotted, which can be seen below:
   For the 'fare' column:



For the ‘Age’ column:



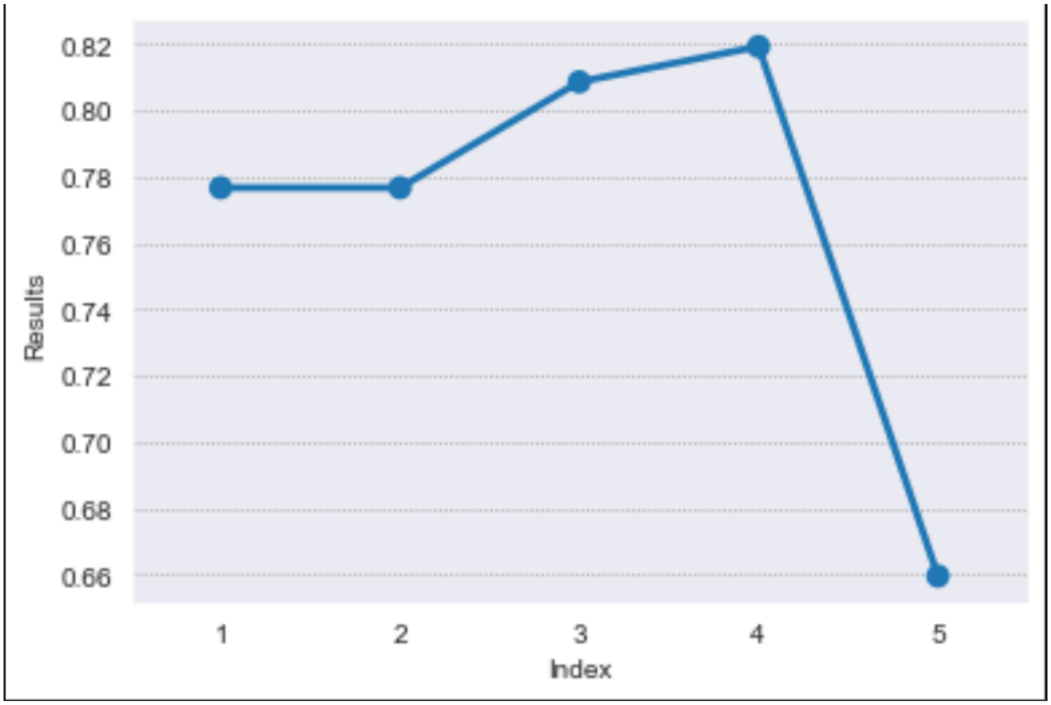
A heatmap was plotted describing the correlation between the different columns of the dataset.



| | Pclass | Sex | Age | Fare | Survived |
|---|---|---|---|---|---|
| Pclass | 1 | -0.16 | -0.37 | -0.55 | -0.36 |
| Sex | -0.16 | 1 | -0.093 | 0.18 | 0.54 |
| Age | -0.37 | -0.093 | 1 | 0.096 | -0.077 |
| Fare | -0.55 | 0.18 | 0.096 | 1 | 0.27 |
| Survived | -0.36 | 0.54 | -0.077 | 0.27 | 1 |

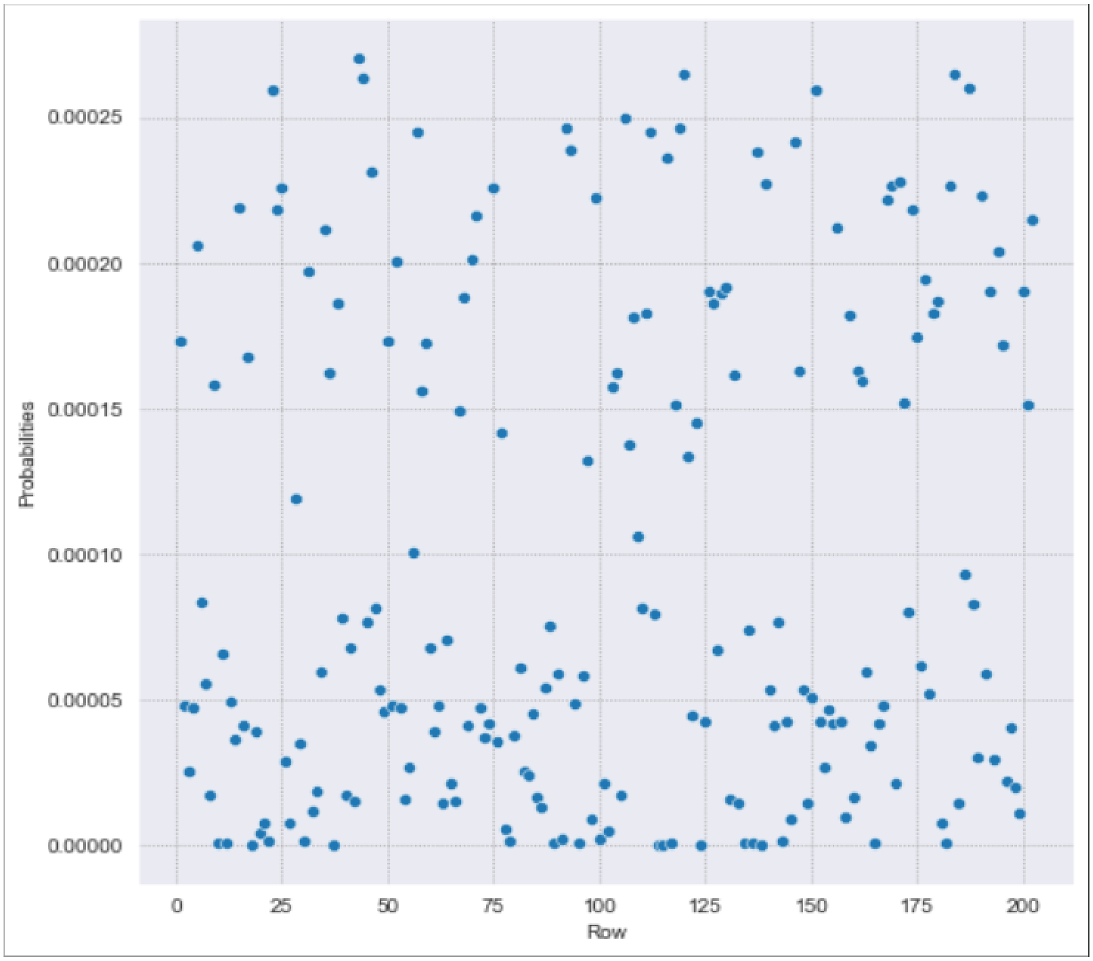
As we can see ‘fare’ column was fairly related to the ‘survived’ column. The ‘age’ and ‘fare’ columns were plotted which gave us an idea about their distribution. Their distribution was related to the Gaussian distribution more than any other distribution. ‘Sex’ column had the most correlation with the ‘survived’ column. Binomial would have been the best way to describe this distribution but the Gaussian would also give the same result if used (Giving higher priority to the female sex). So the Gaussian Naive Bayes classifier was decided to predict on X_test.

3. Gaussian Naive Bayes classifier was implemented from scratch as justified above. 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 results have been plotted below:



Average accuracy of 76.8 percent was achieved. Maximum accuracy was achieved at the 4th fold and the minimum was found at the 5th fold. Point plot was used to describe and visualize the results.

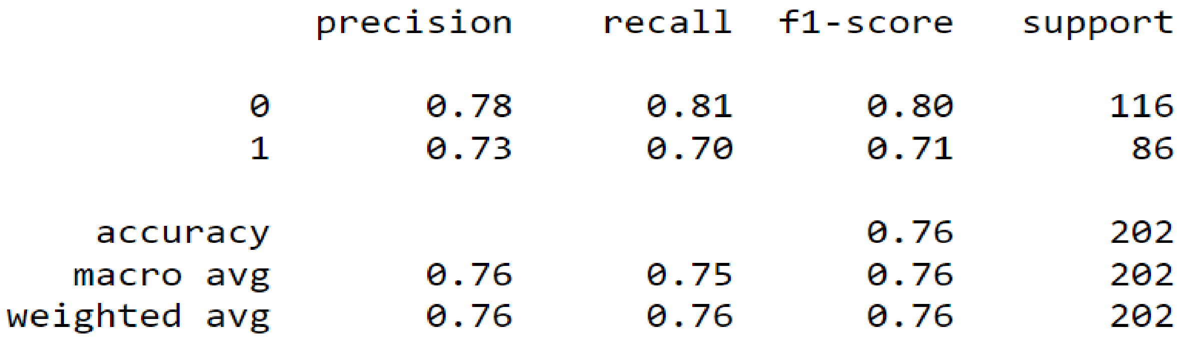
5. The probability of top-class for each row in the testing dataset was plotted and the results are shown below:



The minimum value of probability for any row was 8.7e-9 and the maximum value was 0.00028.

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

Scratch Implementation-

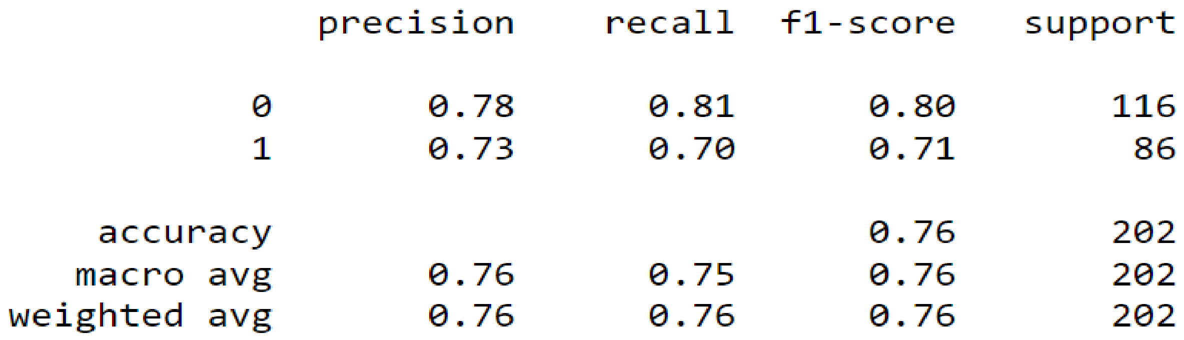


```
              precision    recall  f1-score   support

           0       0.78      0.81      0.80       116
           1       0.73      0.70      0.71        86

    accuracy                           0.76       202
   macro avg       0.76      0.75      0.76       202
weighted avg       0.76      0.76      0.76       202
```

Gaussian Bayes Classifier-



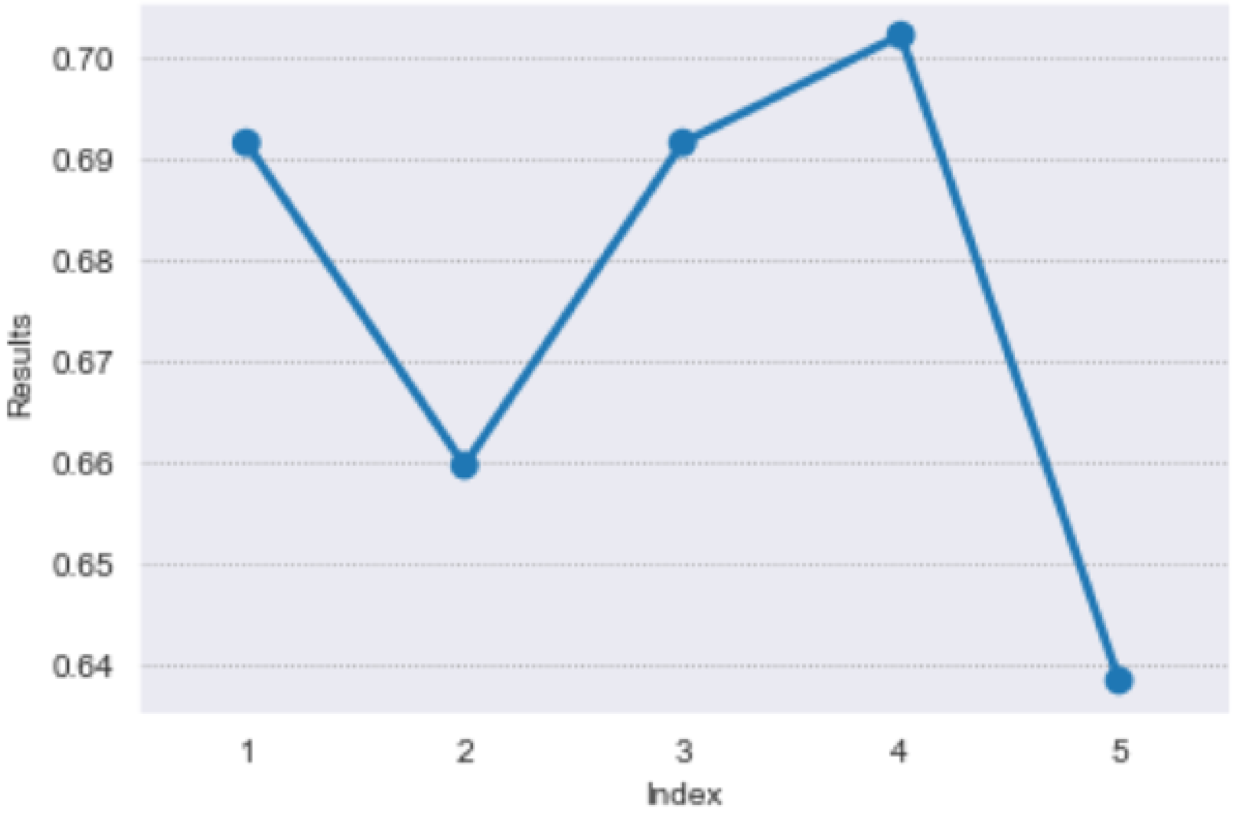
```
              precision    recall  f1-score   support

           0       0.78      0.81      0.80       116
           1       0.73      0.70      0.71        86

    accuracy                           0.76       202
   macro avg       0.76      0.75      0.76       202
weighted avg       0.76      0.76      0.76       202
```

Results for both the models were the same. Giving about 76 percent precision and 76 percent recall. Results can be seen above in the classification report.

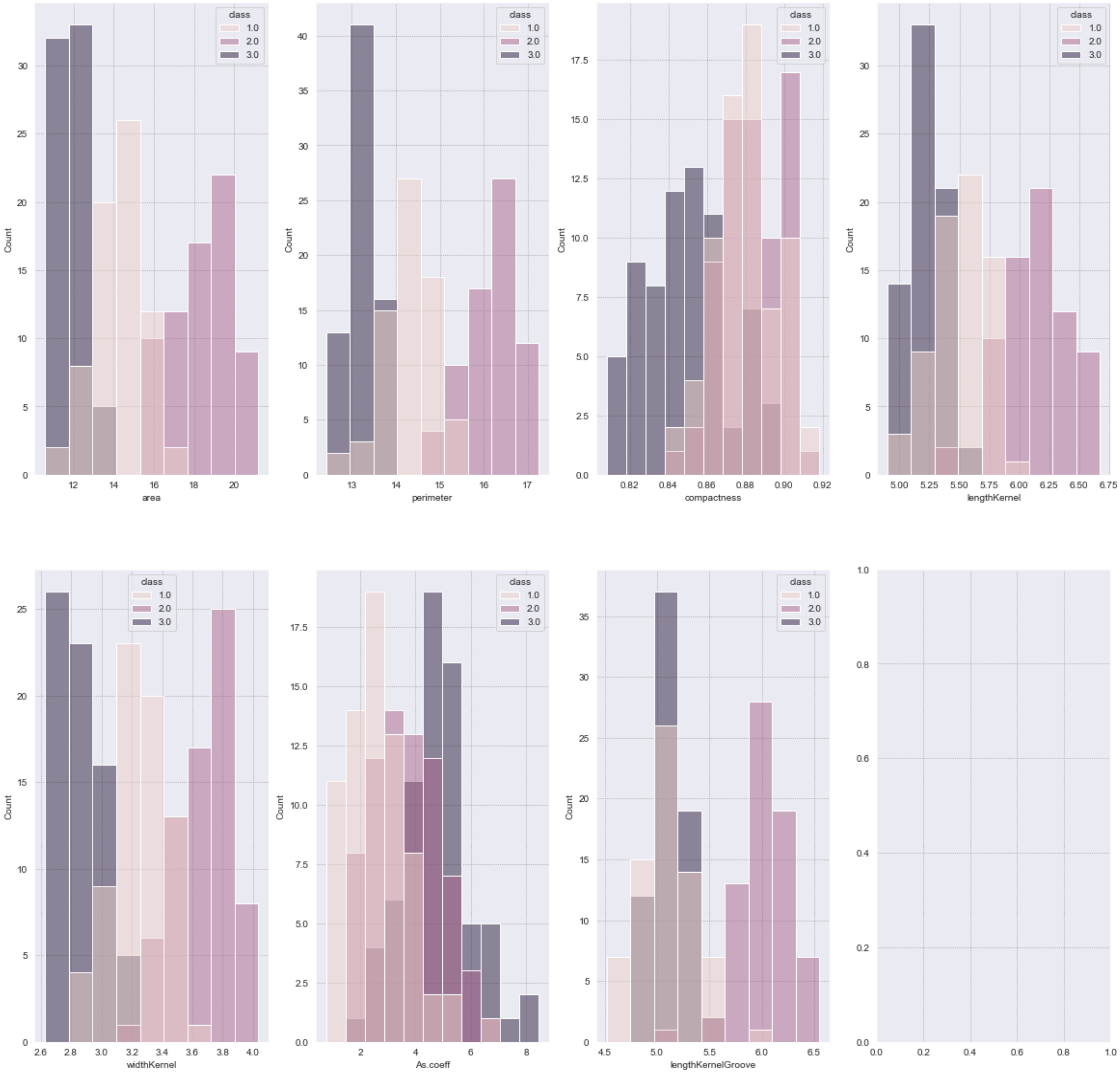
7. Multinomial Bayes classifier was imported and 5-fold cross-validation was performed. Results can be seen below:



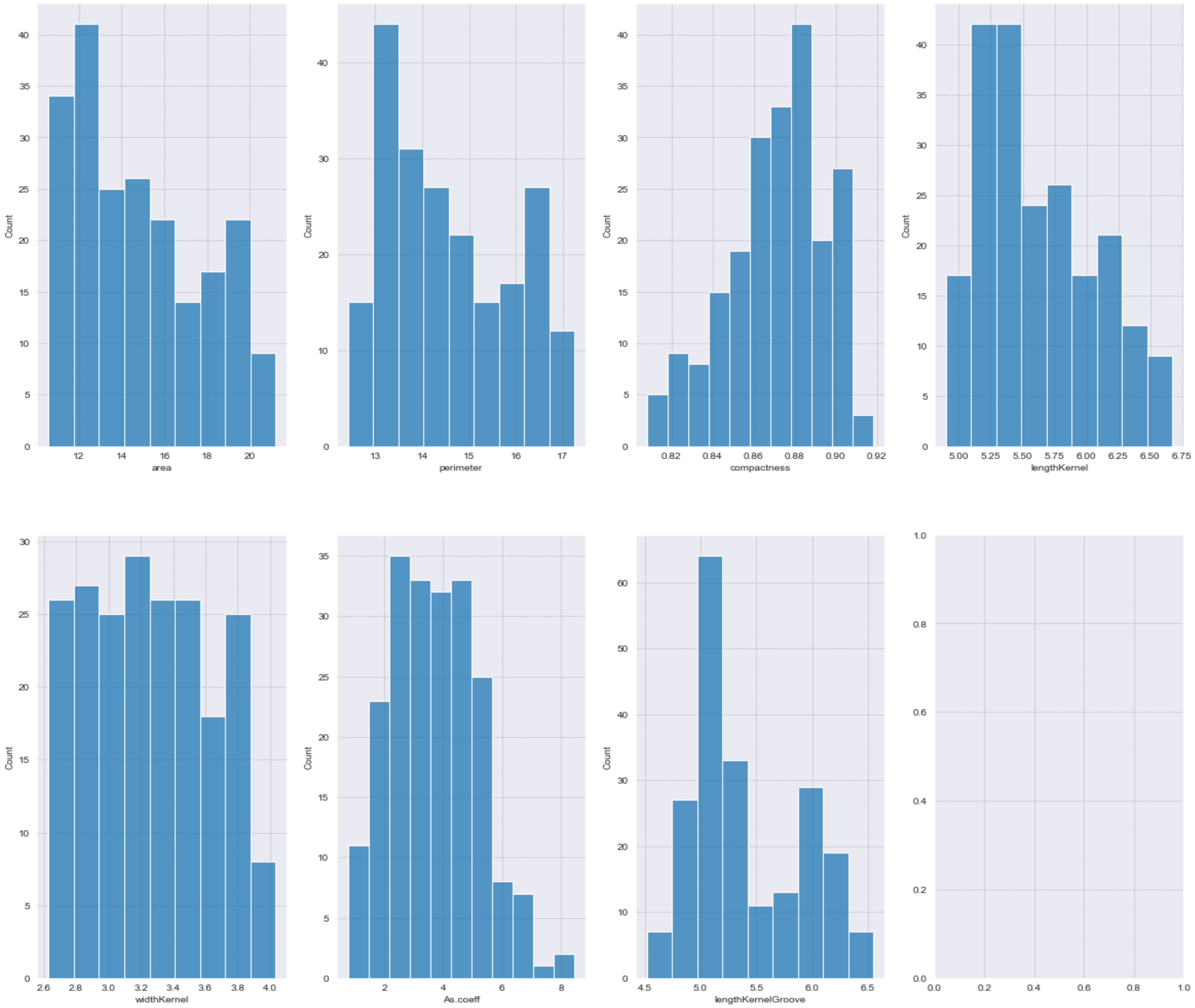
The average accuracy was 67 percent which was far less than the gaussian Bayes classifier. ‘Age’ and ‘fare’ columns were normally distributed and multinomial distribution didn't fit them. This is why we got such poor results on the multinomial Bayes classifier. We got pretty better results than that which can be seen in the above parts.

Question 2 -

1. Data was imported using open() command and preprocessed to give the final output. A histogram was plotted to plot the distribution of the samples. The plot can be seen below with hue as ‘class’ column:

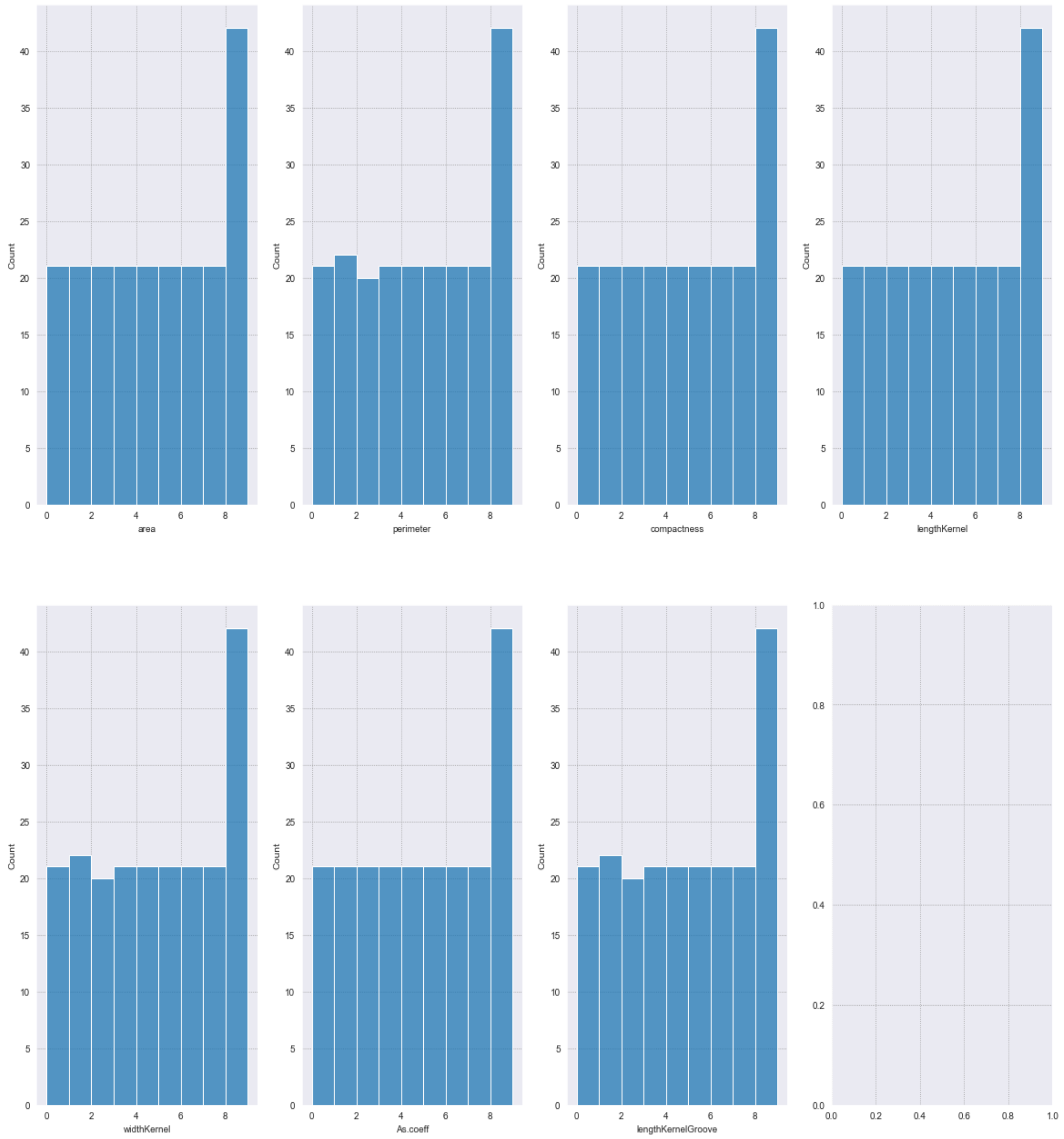


Plot without hue:

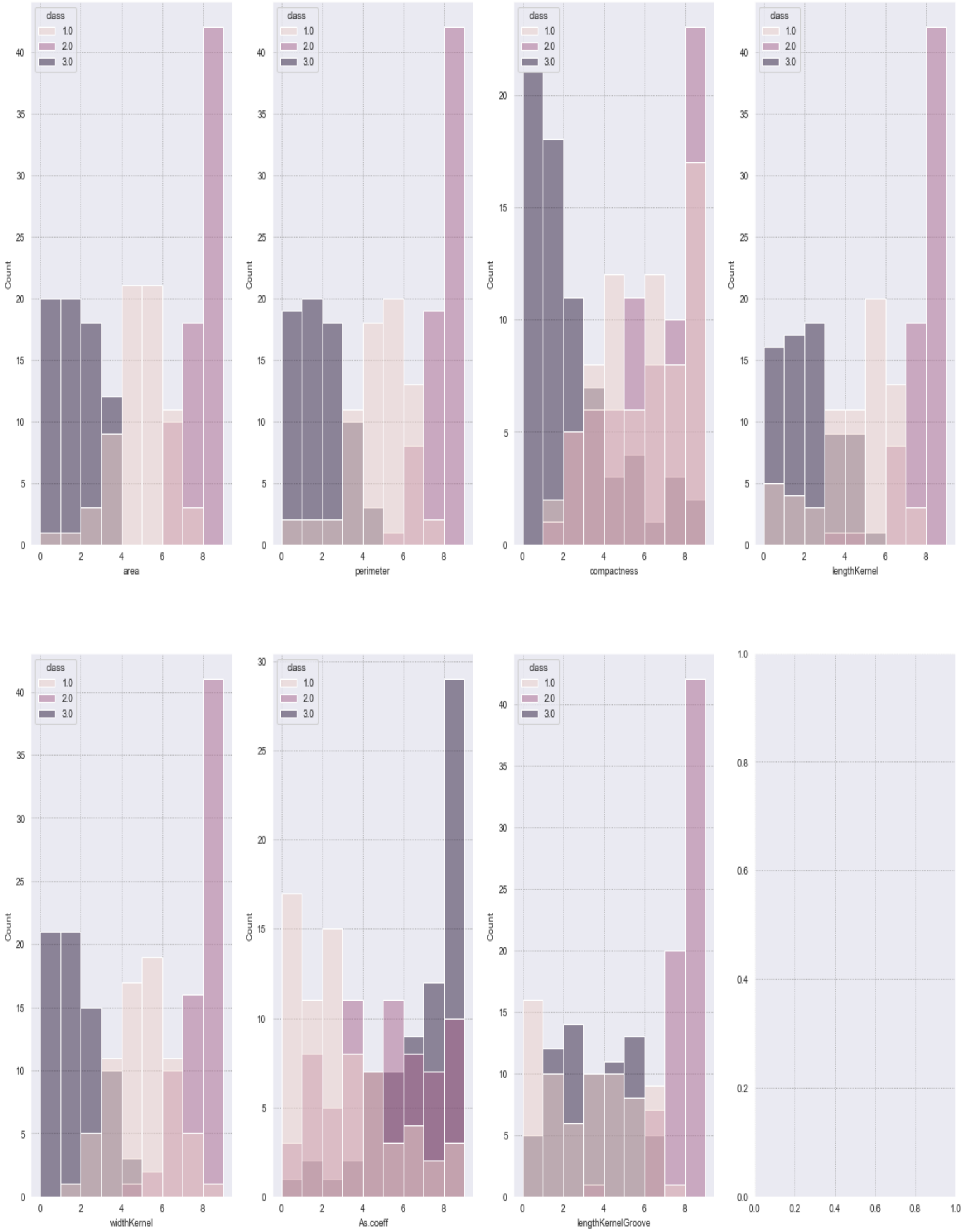


2. Prior probabilities for all the classes were calculated and the results can be seen in the colab notebook.

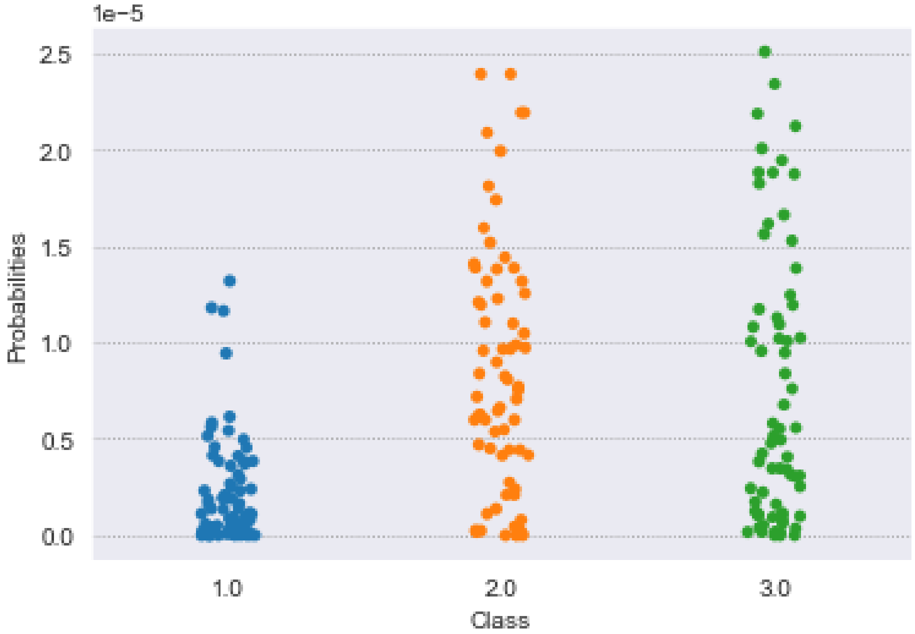
3. The features were discretized into bins with each bin containing equal samples. Each feature was discretized into 10 bins to make the visualization easier. None of the forbidden libraries were used for this task.
4. Likelihood/Class conditionals were calculated and stored inside a dictionary for faster use. Results can be seen in the colab notebook.
5. Plotting the plot required in the 5th part: (without hue)



With hue as ‘class’:



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



The results achieved can be seen above. For class 1 probabilities had a lower absolute value. Majority of which had a value lying between 0.5e-5 and 0 (exclusive). Class ‘2’ and class ‘3’ had kind of similar results with probability ranging from 0 (exclusive) to 2.5e-5. This implied if we used a naive Bayes classifier to predict the results for this dataset, it would have predicted the classes ‘2’ and ‘3’ with lower error than the class ‘1’. Error for class ‘2’ and class ‘3’ would be lesser than the class ‘1’. The strip plot proves and justifies the result.