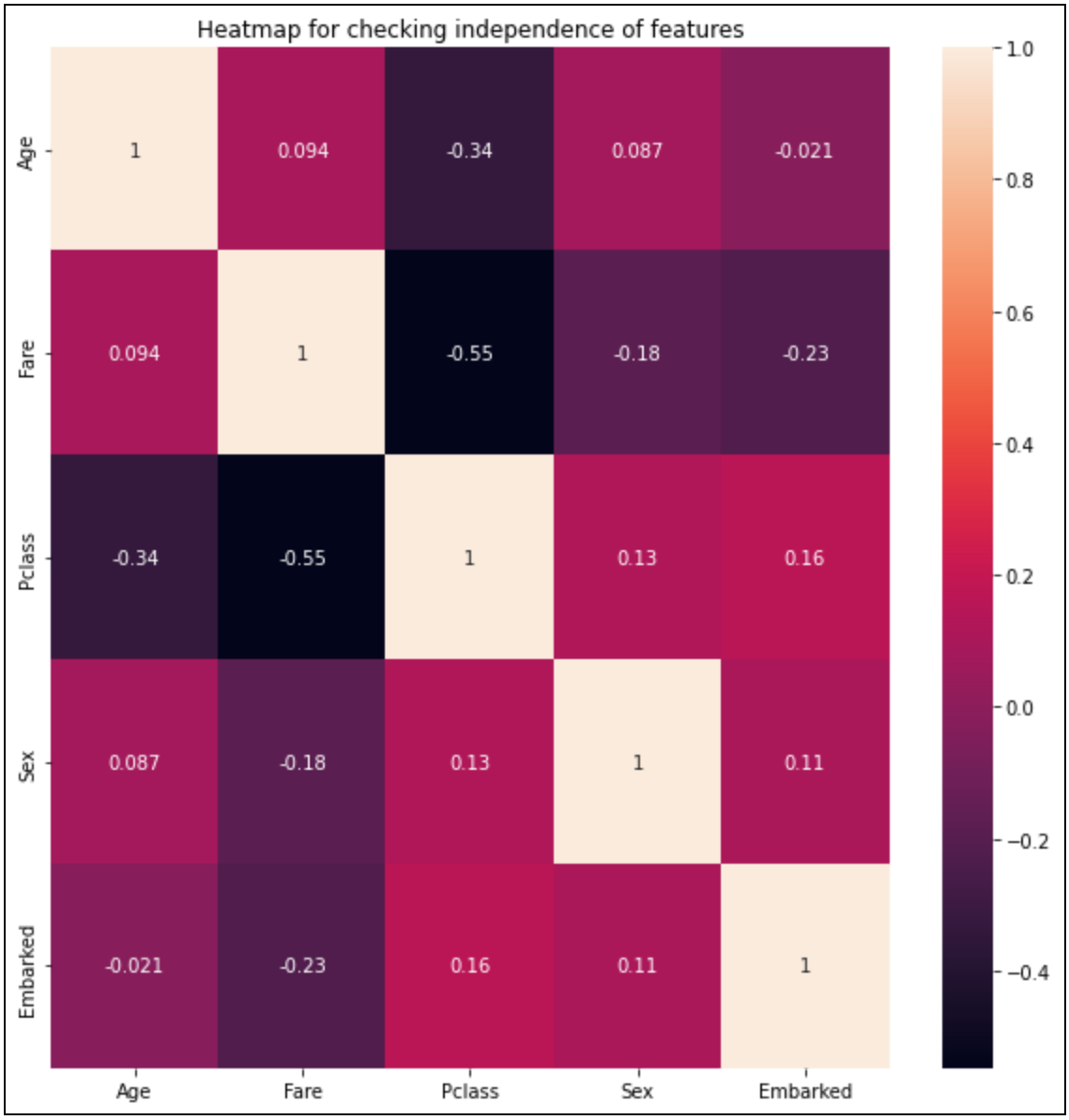
Pattern Recognition and Machine Learning
2022 Winter Semester

Report - Lab Assignment - 4

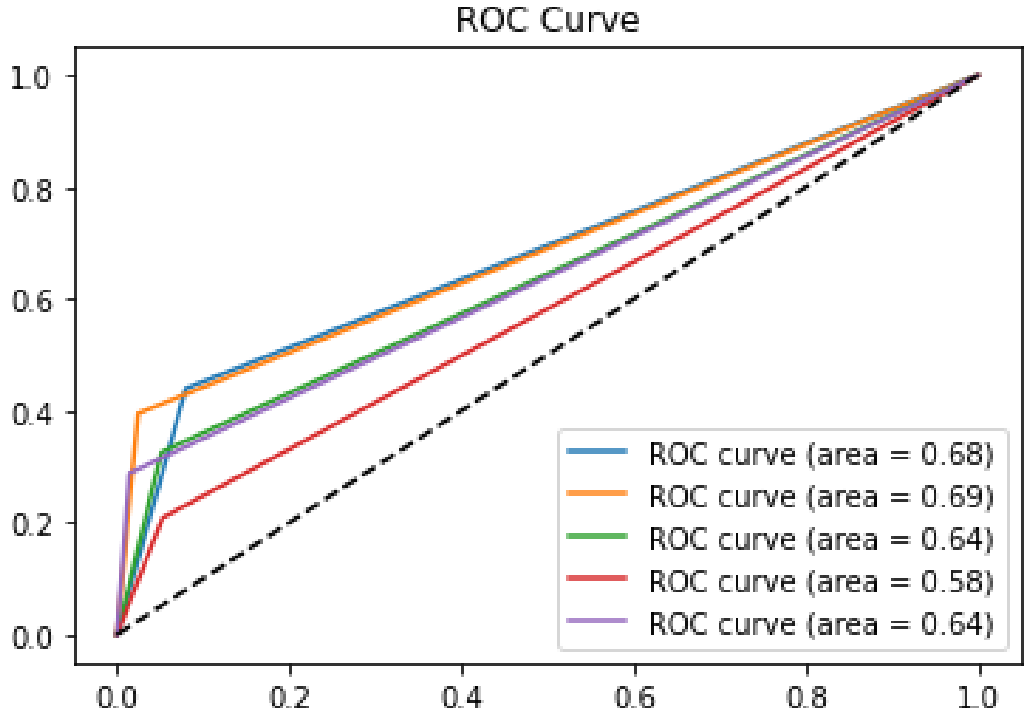
Question - 1

- Initially, I uploaded the dataset and read the dataset and also imported the necessary data manipulation and visualization libraries such as pandas and numpy.
- Then, I dropped the features ‘PassengerId’, ‘Name’, ‘Ticket’, ‘Cabin’ which were unnecessary and didn’t have much effect on the classification.
- Then, I handled the missing values and used the mean of the feature values to replace the NaN values.
- Then, I encoded the ‘Sex’ and ‘Embarked’ features using LabelEncoder() class of sklearn library.
- Then, I splitted the data into a 70:30 ratio using the train_test_split function of sklearn library.
- Following is the diagram of the correlation of every feature with each other. If the correlation is near to zero then the features are nearly independent of each other. Here we can see that most of the features have a correlation value near to zero which means they are independent. Besides this, most of the features are continuous and we have a large amount of data due to which we nearly get a normal distribution. These all are favorable conditions for applying the Gaussian Naive Bayes Classifier.

- 

- After this, I also plotted the gaussian probability distributions for the features ‘Age’, ‘Fare’ and ‘Pclass’ which were continuous in nature.
- Then, I implemented the Naive Bayes Class from Scratch in which I included the functions which calculated the Prior probabilities, gaussian densities, and post prior probabilities. The class also included the functions for fitting the model, making predictions based on the calculated Posteriors and calculating accuracy of the model.

- The function posterior probability returned the predictions based on the maximum probability among all the classes of the target feature(“Survived”)
- The accuracy score for the scratch Gaussian Naive Bayes Classifier came out to be 74.906 %.
- Then, I Implemented five fold cross validation on my training data and plotted the ROC Curves for the folds which are used in cross validation and even the auc is also given in the legend domain of the curve.

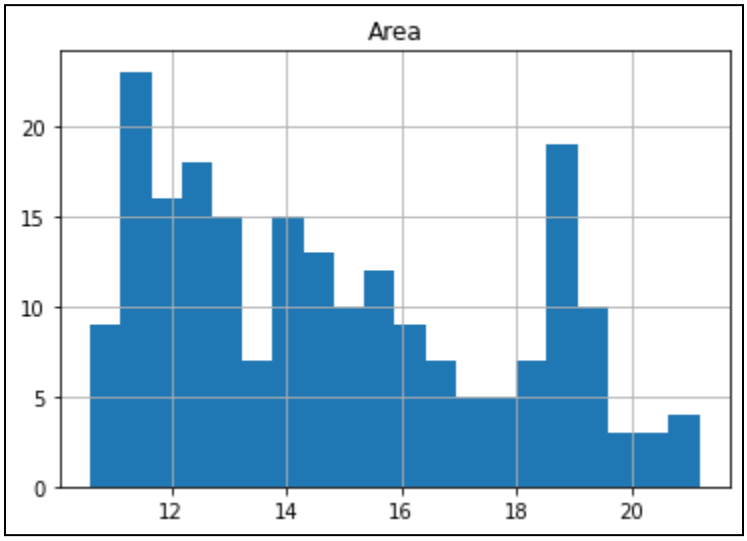


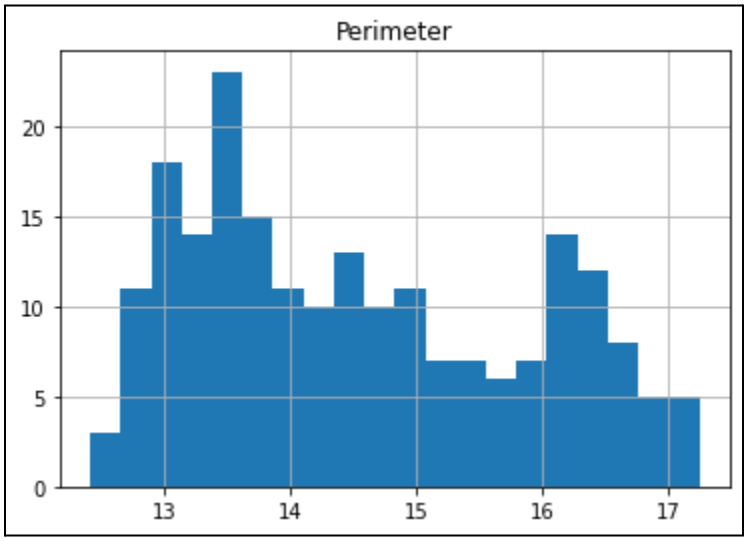
- The results of five fold cross validation were:
  [0.728, 0.776, 0.717741935483871, 0.6612903225806451, 0.6935483870967742]
- The above plot clearly depicts that at a low false positive rate we have a high true positive rate for all the folds of 5 cross validation.
- Then, I applied the Gaussian Classifier from the sklearn library and the accuracy score was coming out to be 78.27%.

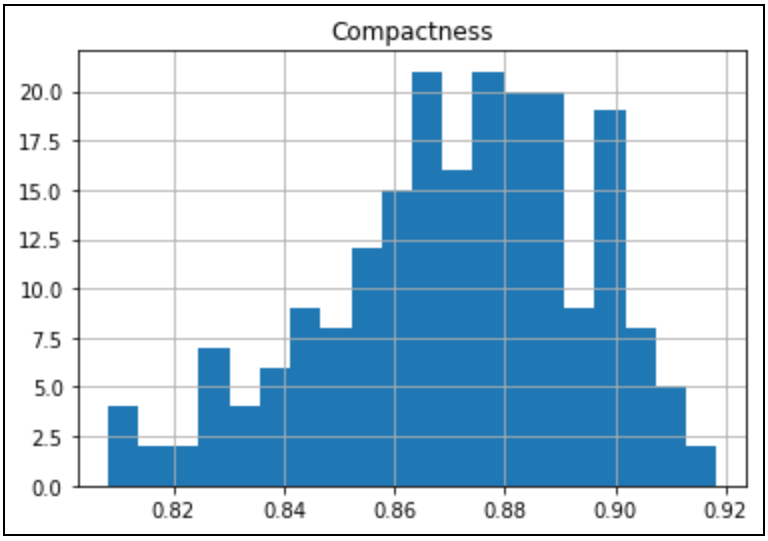
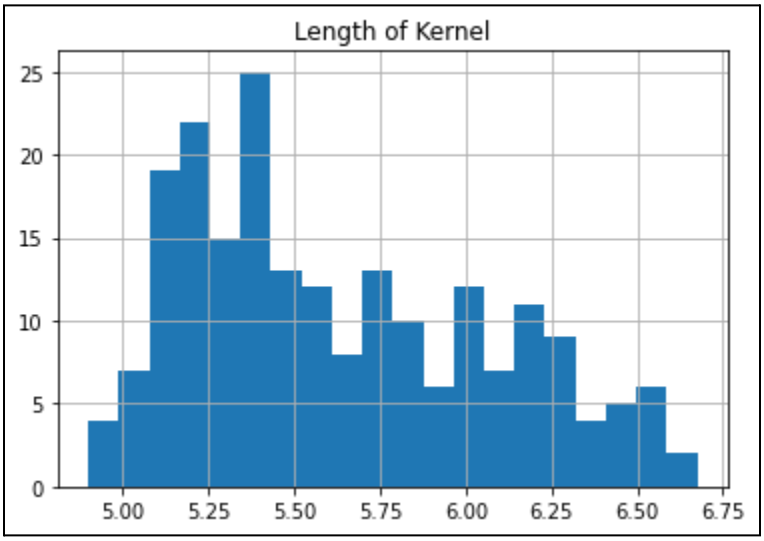
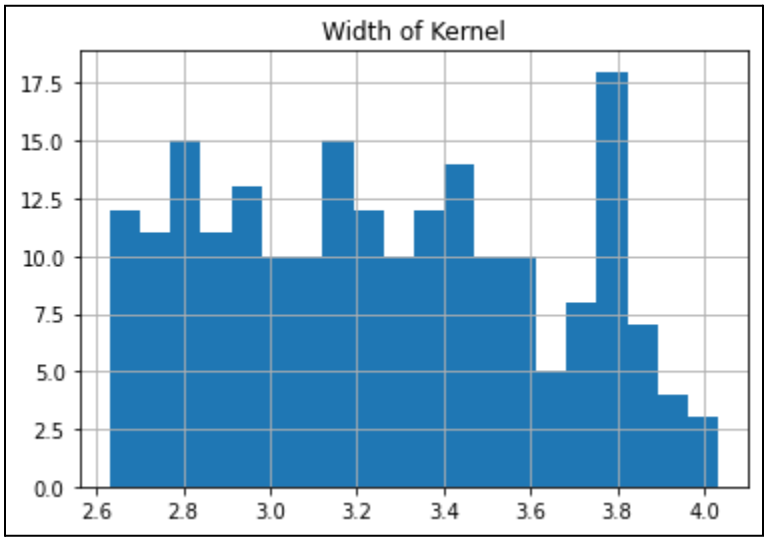
- The cross validation scores for the inbuilt gaussian classifier were: [0.736 0.8 0.77419355 0.75 0.80645161], which is a little more as compared to our scratch model.
- Then, I implemented the Multinomial Naive Bayes Classifier and Bernaulli Naive Bayes Classifier of the sklearn library and calculated the accuracies which came out to be 67.7% and 82.02% respectively.
- The reason for lesser accuracy and avoidance of the Multinomial NB is that generally it is applied to that data where counting or frequency of occurring or something related to it is used and even we have a pretty large dataset which leads to normal distribution according to central limit theorem which made the favorable condition for preferably using gaussian naive bias Classifier.
- And the reason for avoidance of Bernaulli Classifier even if its accuracy is higher is because this classifier is used in cases where the labels are either present with us or absent.

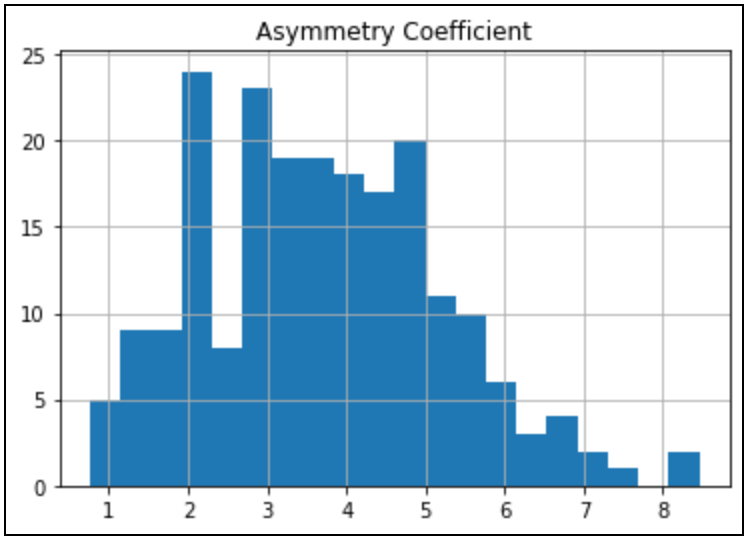
Question - 2

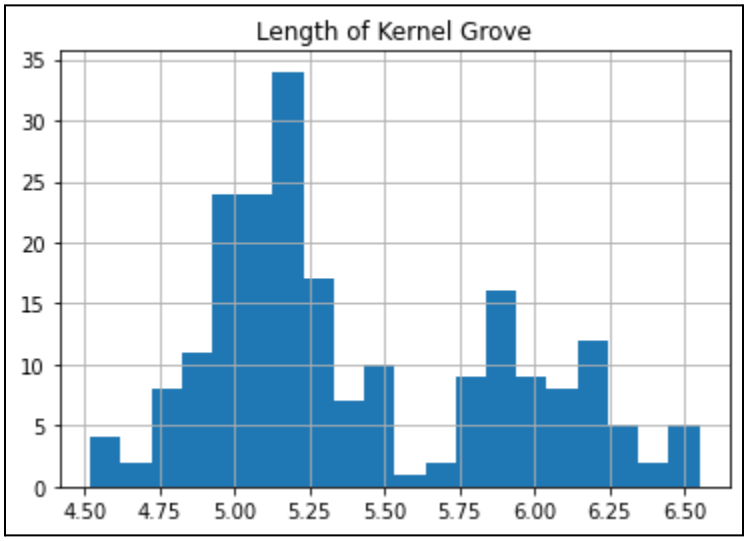
- Initially, I uploaded the dataset and read the dataset and also imported the necessary data manipulation and visualization libraries such as pandas and numpy.
- Then, I renamed the columns as the parameters given to us in the question.
    - Area.
    - Perimeter.
    - Compactness
    - Length of kernel.
    - Width of kernel.
    - Asymmetry coefficient.

    - Length of kernel groove.
    - Class (1, 2, 3).
- Then, I plotted the histograms to show the distribution of the samples.
- 

- 

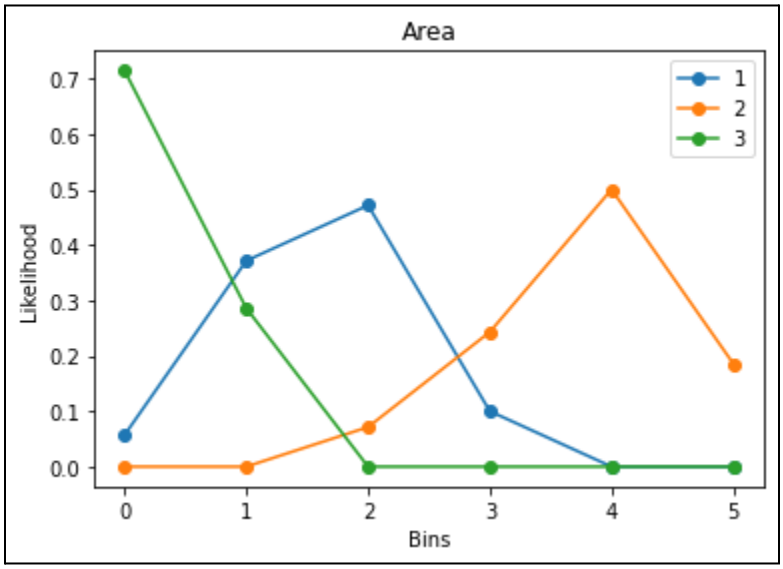
- 
Compactness
20.0
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
0.82
0.84
0.86
0.88
0.90
0.92
- 
Length of Kernel
25
20
15
10
5
0
5.00
5.25
5.50
5.75
6.00
6.25
6.50
6.75
- 
Width of Kernel
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
2.6
2.8
3.0
3.2
3.4
3.6
3.8
4.0

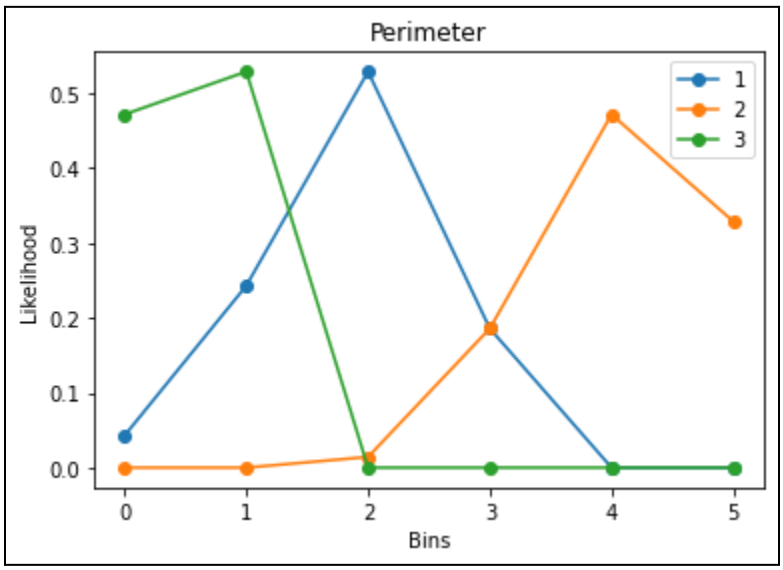
- 

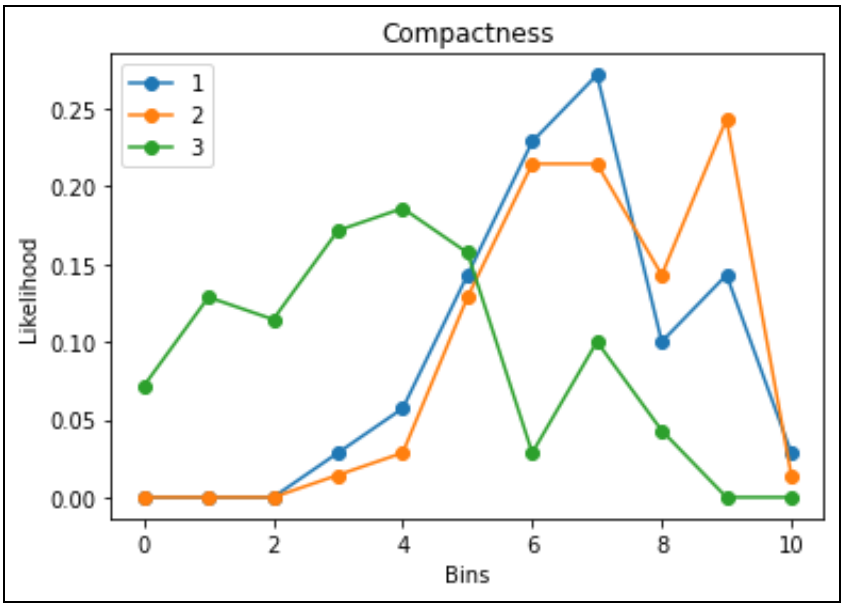
- 

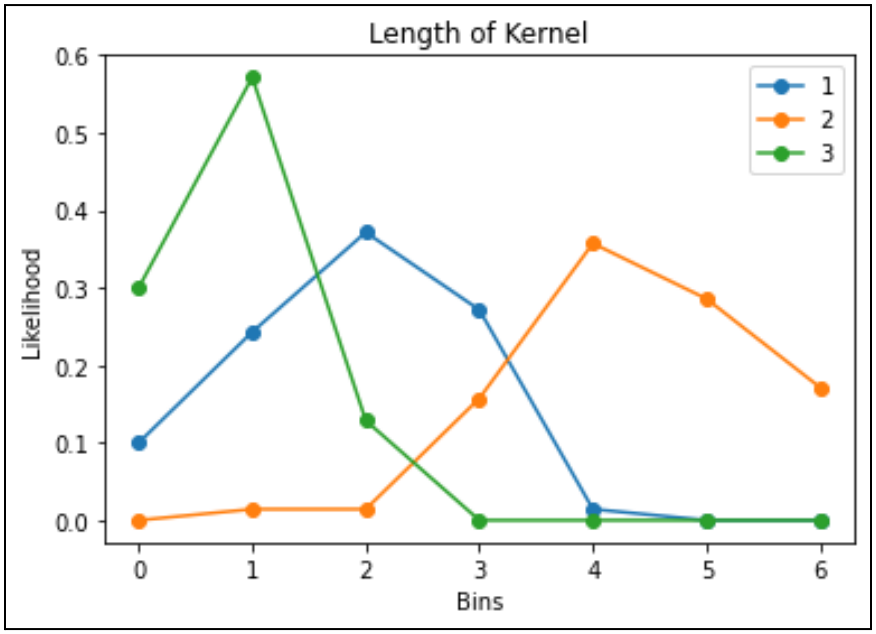
- For discretizing the features into bins, firstly the bins were considered. To have the best bin size, I used the freeman algorithm which says Size=2*(IQR)/(n)^⅓ where IQR is the interquartile range(75%-25%).
- Now after selecting the size of the bins, the features of the dataset were segregated into their respective bins and a new copy of the dataset was created with discrete features into bins.
- Then, I calculated the likelihood conditional probabilities for all the features considering all the 3 classes. Likelihood of a particular feature given a particular class was calculated for

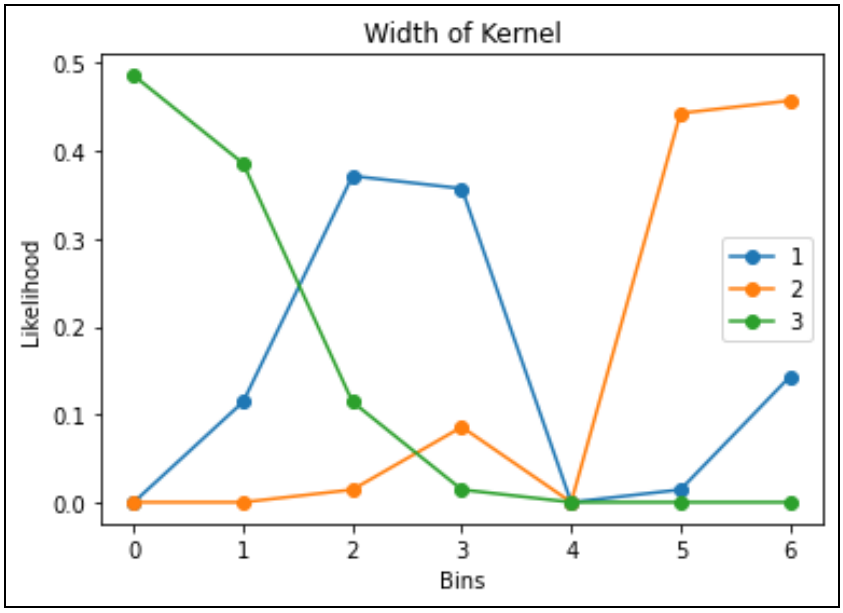
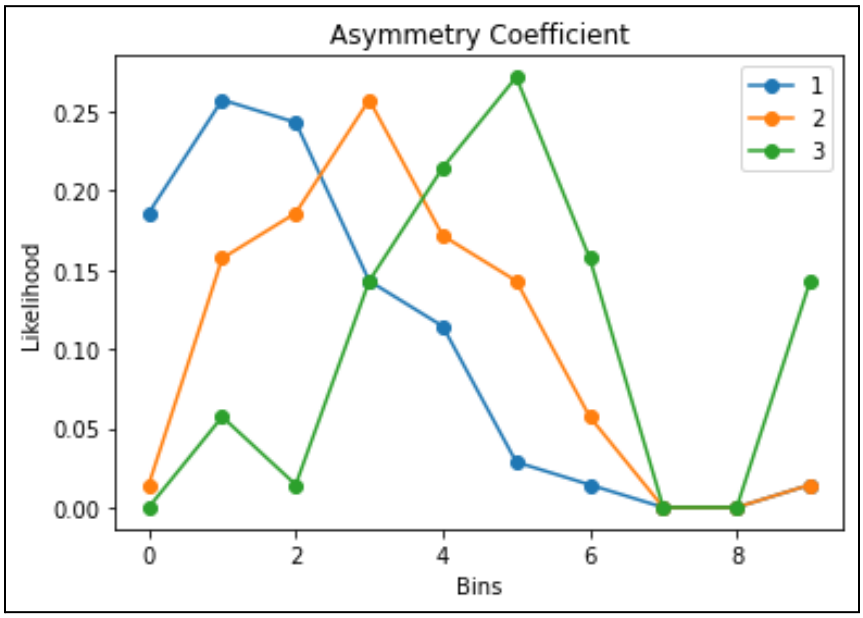
every feature given all the 3 classes and all this data was stored in a dictionary format and then printed accordingly.

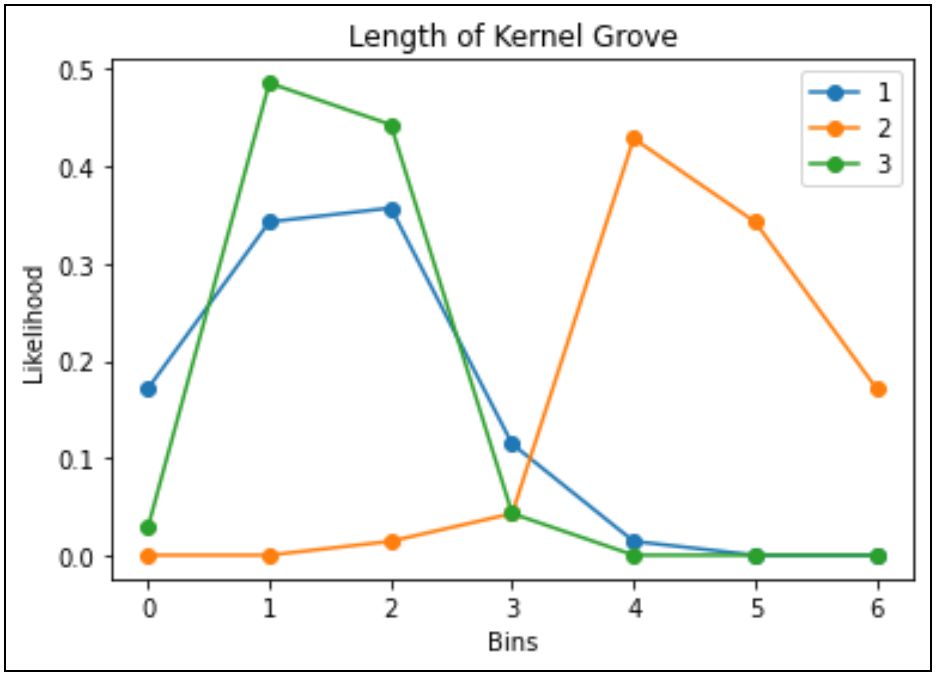
- Then, I plotted the likelihood plots for all the features given all the 3 classes in the dataset.
- Following are the likelihood plots:
- 

- 

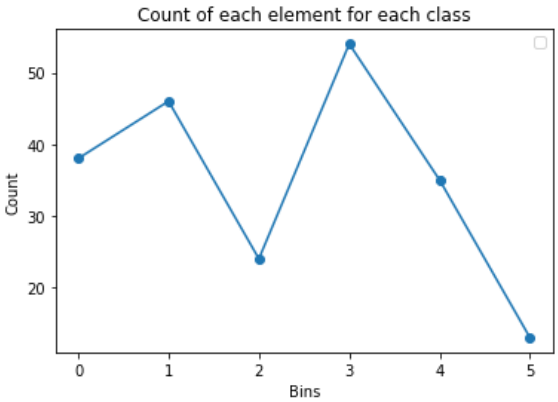
- 
Compactness
1
2
3
Likelihood
0.25
0.20
0.15
0.10
0.05
0.00
0
2
4
6
8
10
Bins

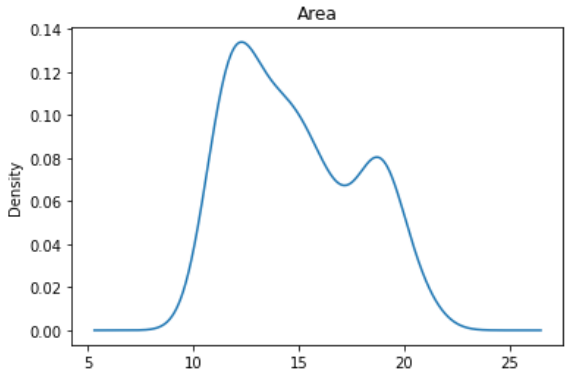
- 
Length of Kernel
1
2
3
Likelihood
0.6
0.5
0.4
0.3
0.2
0.1
0.0
0
1
2
3
4
5
6
Bins

- 
Width of Kernel
Likelihood
Bins
0.0
0.1
0.2
0.3
0.4
0.5
0
1
2
3
4
5
6
1
2
3
- 
Asymmetry Coefficient
Likelihood
Bins
0.00
0.05
0.10
0.15
0.20
0.25
0
2
4
6
8
1
2
3

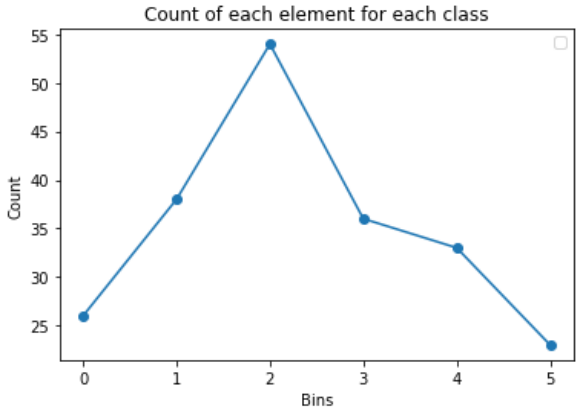
- 

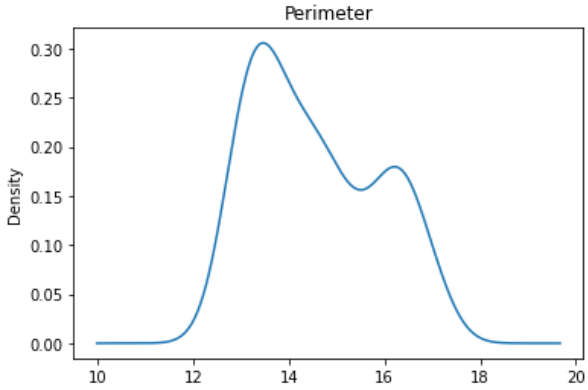
- Then, I plotted the count of each unique element for each class. and compared the plot with the plots of distribution.
    - Area



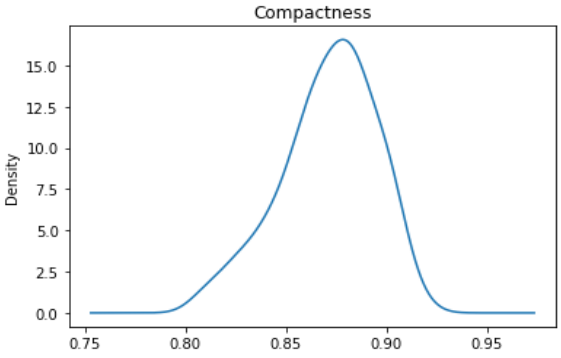


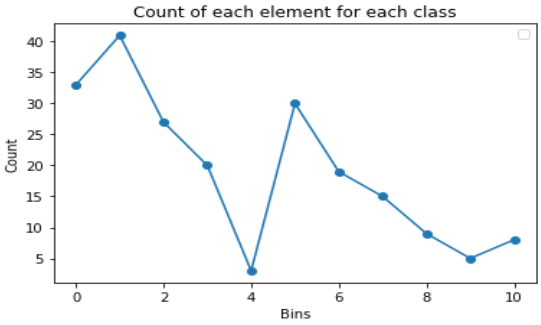
- Perimeter



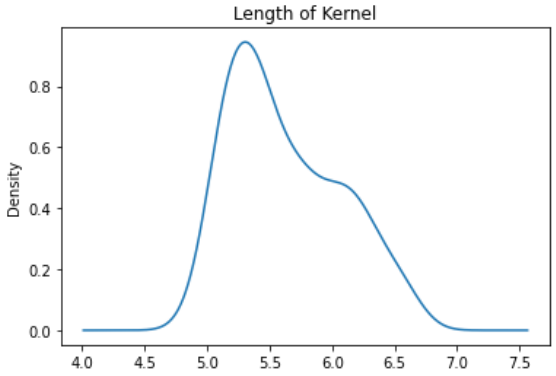


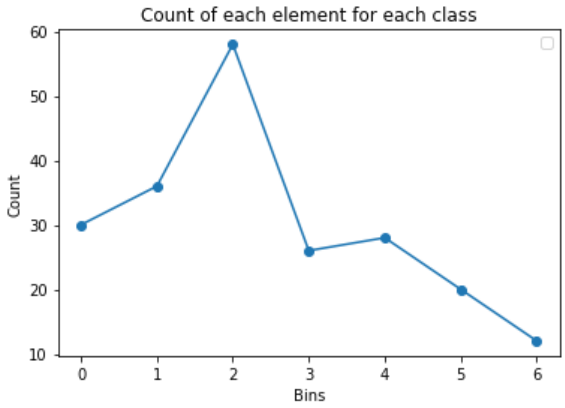
- Compactness



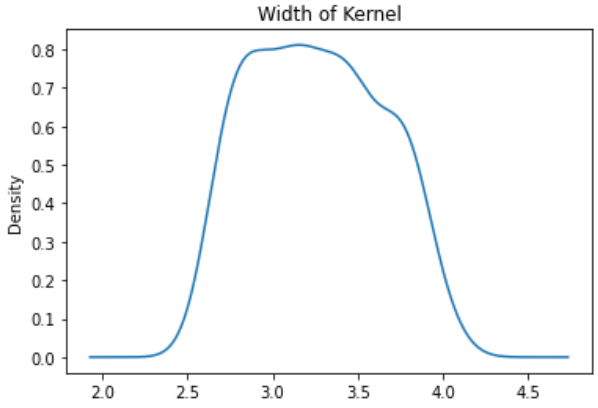


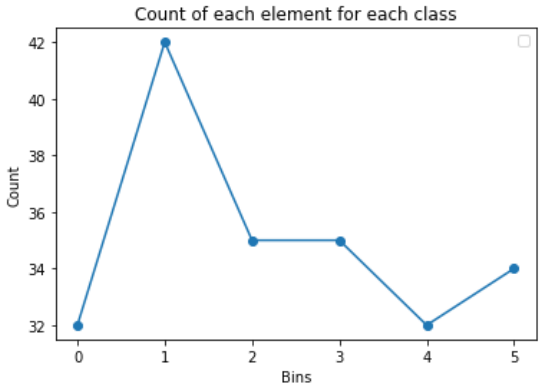
- Length of Kernel



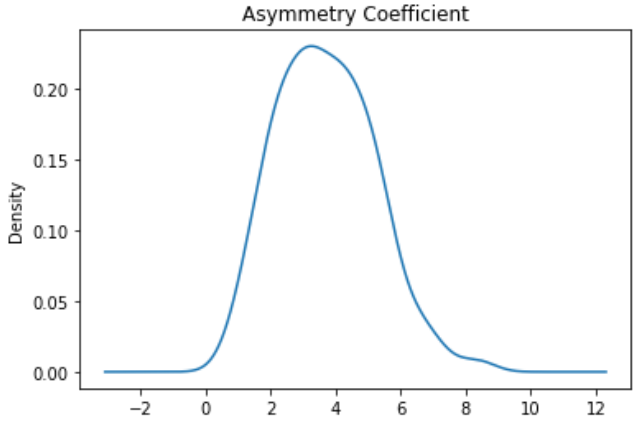


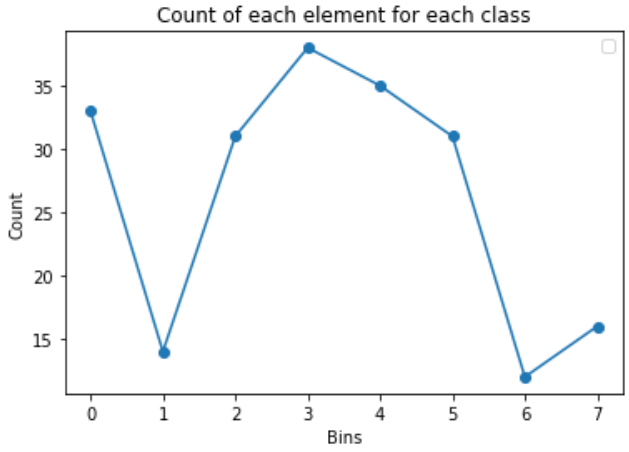
- Width of Kernel



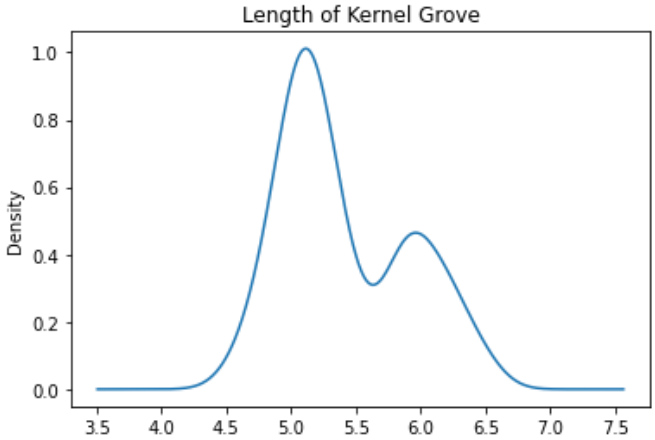


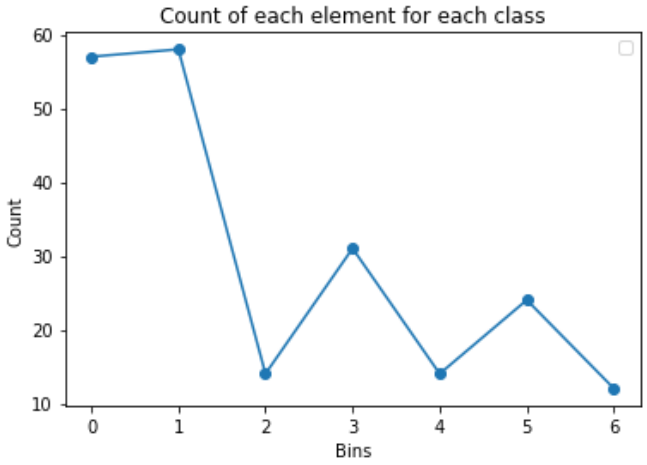
- Asymmetry Coefficient



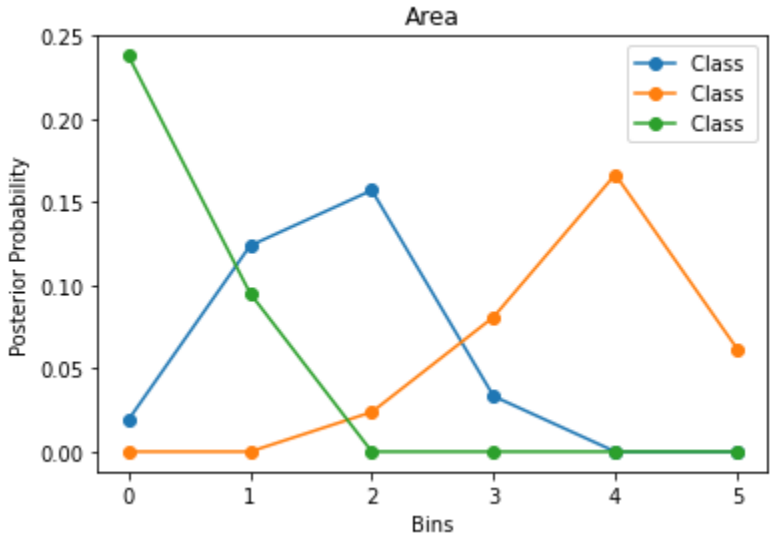


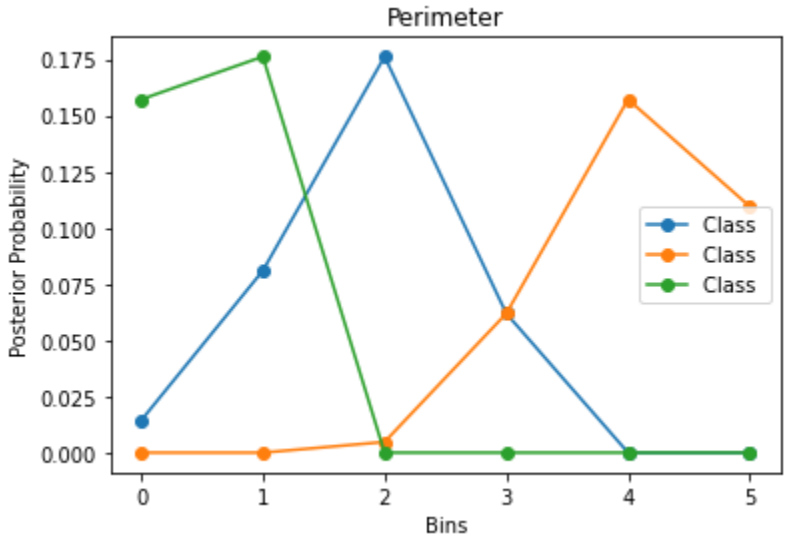
- Length of Kernel Groove

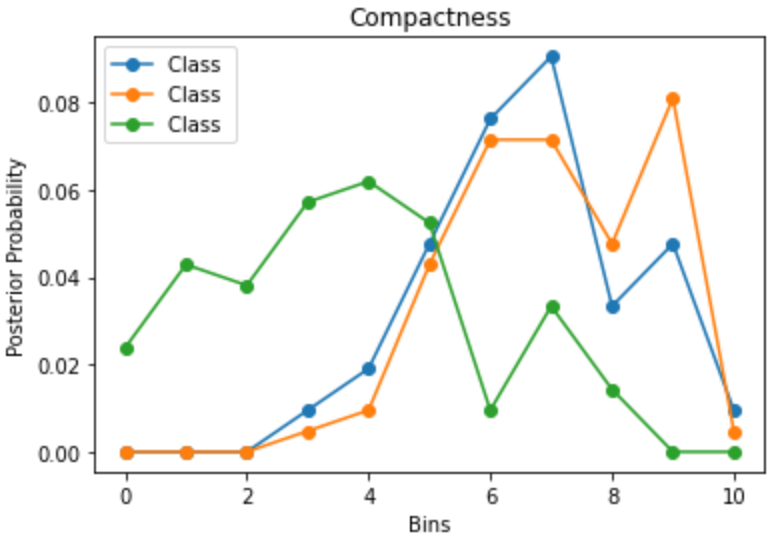
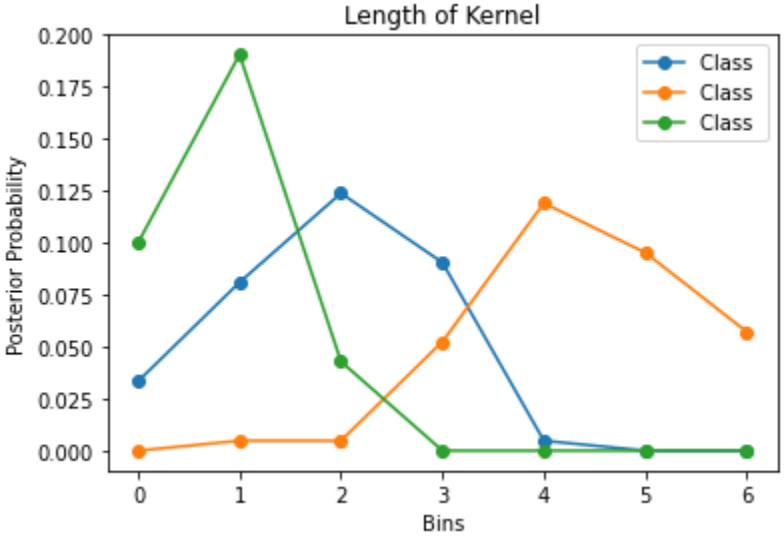
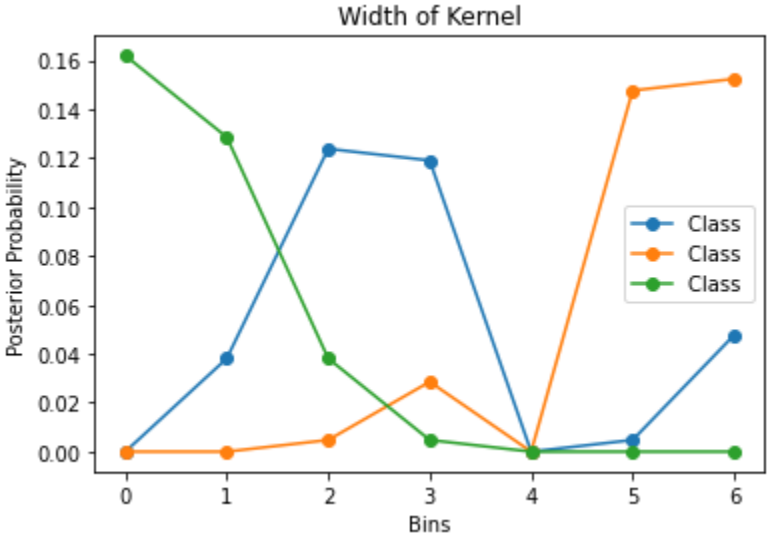


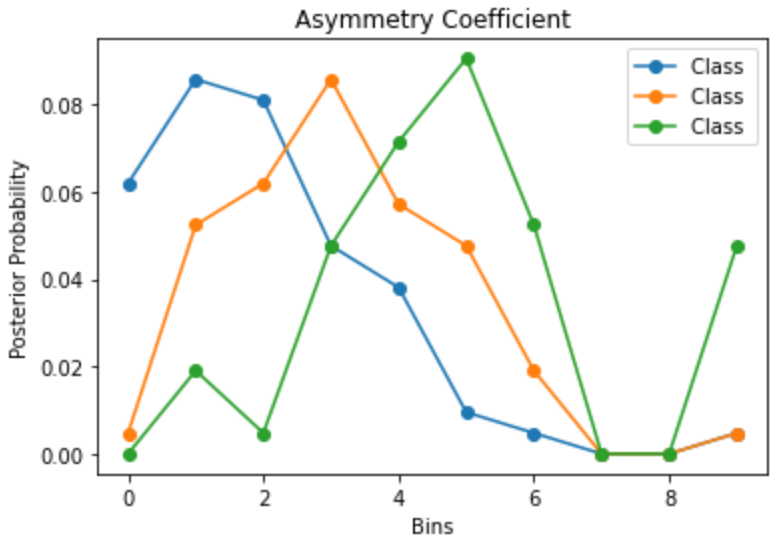


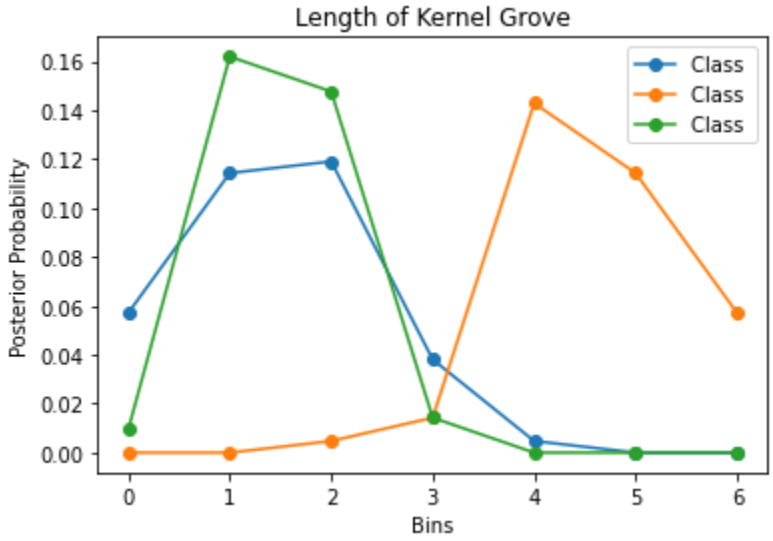
- If we compare the distribution of features along with the line graph then we can see that they are synchronizing with each other,i.e. if a positive slope is observed in distribution then an increase in the count is observed if it's a line graph.

- The plots of Posterior Probability calculated at the last are
- 

- 

- 
Compactness
Class
Class
Class
Posterior Probability
0.00
0.02
0.04
0.06
0.08
0
2
4
6
8
10
Bins
- 
Length of Kernel
Class
Class
Class
Posterior Probability
0.000
0.025
0.050
0.075
0.100
0.125
0.150
0.175
0.200
0
1
2
3
4
5
6
Bins
- 
Width of Kernel
Class
Class
Class
Posterior Probability
0.00
0.02
0.04
0.06
0.08
0.10
0.12
0.14
0.16
0
1
2
3
4
5
6
Bins

- 

- 

- Here we have the plots for posterior probability of the feature and we have fine bifurcations according to the features and classes. So for prediction, that value of posterior probability of a particular feature is selected which is maximum among them and then we repeat that procedure for all other features.