Lab 4
Abu Shahid
(B20CS003)

Question 1

Subpart 1: Preprocessing, visualization and finding important features

- Data was loaded and processed.
- Due to the number of missing vales in ‘Cabin’ feature, column was dropped.
- Unnecessary features like ‘Name’, ‘Ticket’ and ‘PassengerId’ were also dropped immediately.
- Filling Age values were filled with the mean.
- ‘Embark’ and ‘Sex’ were one-hot-encoded and ordinally encoded to better understand the relationship between features and target and select most relevant features.
- Dependency of Survivability wrt various features can be summarize as:

```
Survived       1.000000
class_0        0.282368
sex_0          0.282368
Fare           0.255290
embarked_0     0.169966
class_1        0.095002
sex_1          0.095002
embarked_1     0.004536
Age           -0.074673
embarked_2    -0.151777
Embarked_     -0.169718
class_2       -0.320171
Pclass        -0.335549
Sex_          -0.541585
Name: Survived, dtype: float64
```

- Clearly, the most important features are:
    - Sex_
    - Pclass
    - class_0
    - sex_0
    - Fare
- As class_0 and sex_0 are just another representation of Sex_ and Pclass, they were ignored.
- Here we make an assumption that our data preserves much of its integrity with mere 3 features: Sex_, Fare, Pclass

Subpart 2: Identifying best variant of NB

There are three variants of Naive Bayes we can choose for classification:- 1) Gaussian Naive Bayes - continuous features 2) Bernoulli Naive Bayes - binary features 3) Multinomial Naive Bayes - categorical features

Since Fare are continuous features, it would be apropriate to use gaussian naive bayes over the other types since they would not be able to capture the variations in data as well.

Subpart 3: Implementing the above variant from scratch

- Necessary functions and class was declared to implemet Gaussian Naive Bayes from scratch.
- Our Model had following methods:
    - fit(X, y)
    - predict(X): returns a list of 2 lists.
    - The first list are the actual predictions
    - the second list is the confidence level of those predictions

```
accuracy of model on train set:  0.7721518987341772
mean confidence for all predictions:  0.3924050632911392
```

Subpart 4: Performing 5-fold CV

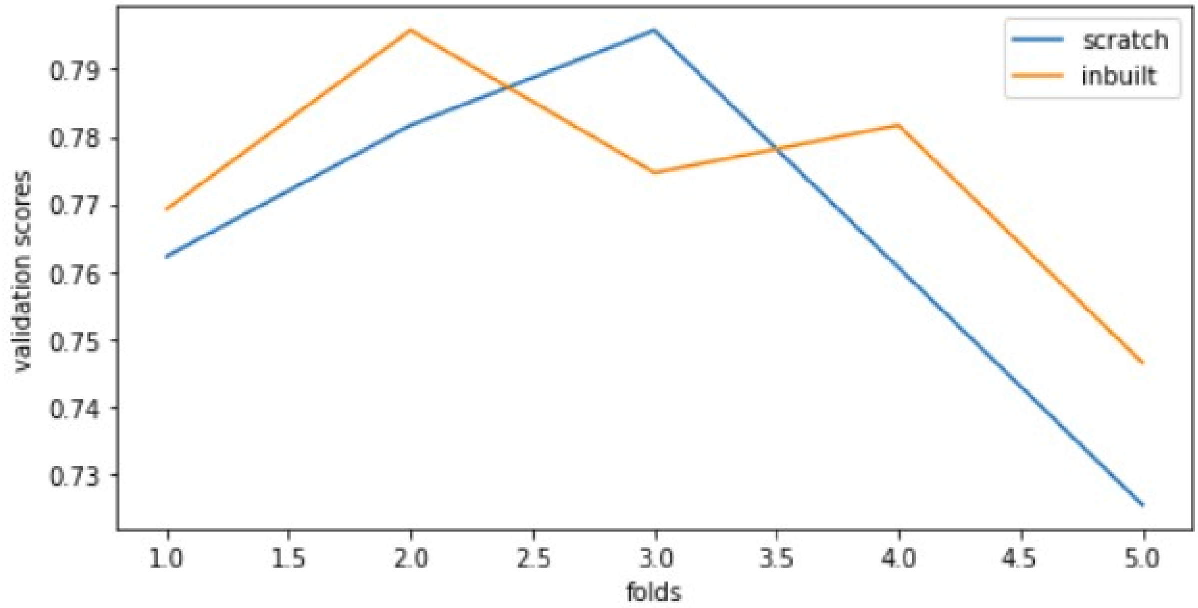
- Declared function cross(model, X, y, fold) from scratch
- Next, 5 fold CV was performed and the results can be summarise as:

```
[0.76223776 0.78169014 0.79577465 0.76056338 0.72535211]
```

- The scores, thought not very great, are consistent and promising; showing that the model is able to generalize the data without overfitting.

Subpart 5: Visualization and summarization of model

- Model performance summary:
- The 5 fold CV results of our model v/s inbuilt can be found below:



- Both the score sets are consistent and in close agreement.
- For our scratch model, validation summary was:

```
mean validation score : 0.7651236087855806
variance in validation score:  0.0005649663759179713

average confidence of top class:  0.8586532602988465
minimum confidence:  0.515027676593242l
```

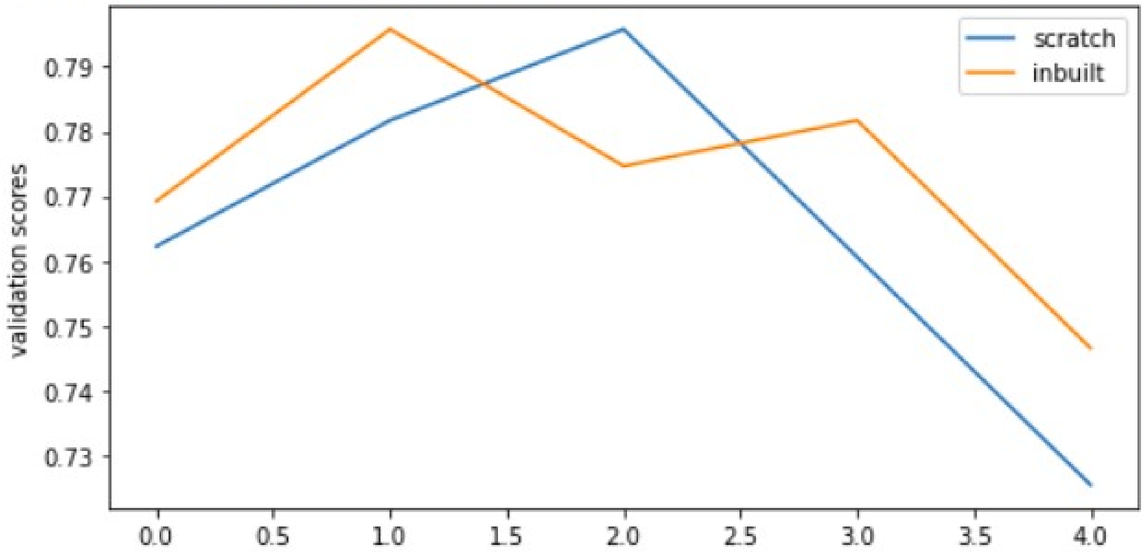
- On close inspction of first 10 predictions:

```
confidence of first 10 prediction:  [0.93317996 0.9991147  0.74549624 0.86612
 0.86612935 0.9333811  0.9983197  0.99774948]
first 10 predictions:  [0 1 1 0 1 0 0 0 1 1]
actual classes:  [0 1 1 0 1 1 0 0 1 1]
accuracy:  0.9
```

  - Confidences are high in general
  - Accuracy is decent

Subpart 6: Comparison with Inbuilt GaussianNB

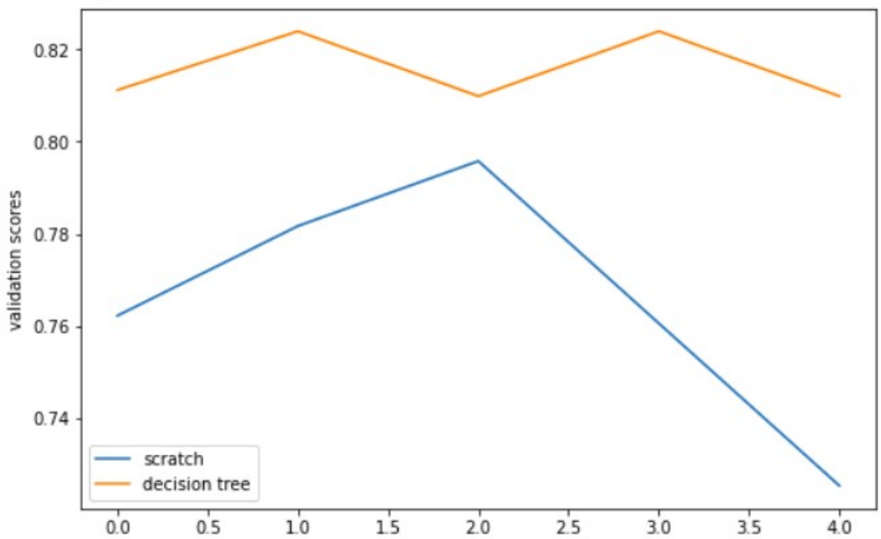
```
mean validation scratch score : 0.7651236087855806
variance in scratch validation score:  0.0005649663759179713
average confidence of top class:  0.8586532602988465
accuracy:  0.7921348314606742

mean validation inbuilt score : 0.7735644637053087
variance in inbuilt validation score:  0.00026258056715993956
average confidence of top class:  0.5
accuracy:  0.7921348314606742
```



Subpart 7: Comparison with another Model(Decision Tree/inbuilt)

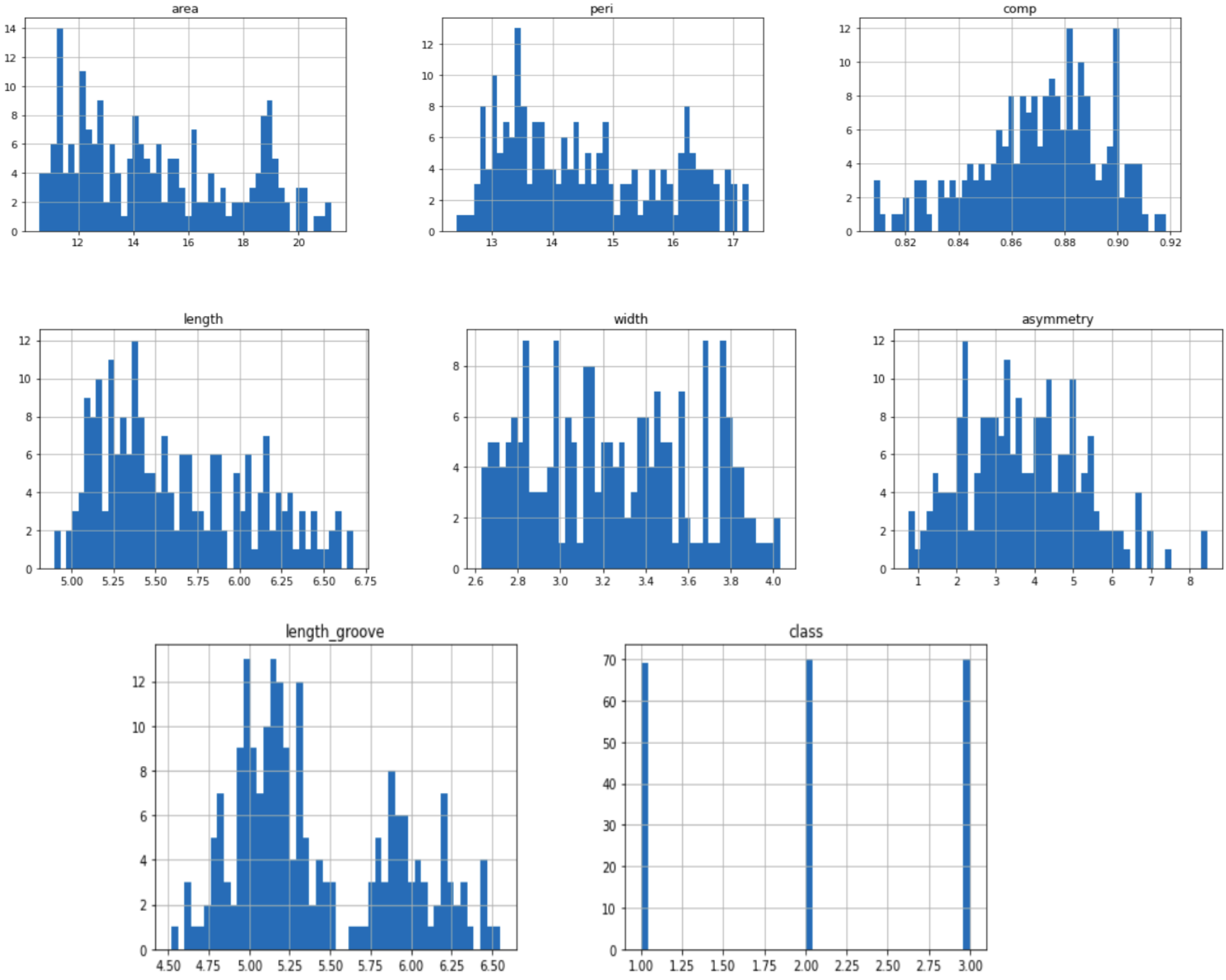
```
mean validation scratch score : 0.7651236087855806
variance in scratch validation score:  0.0005649663759179713
average confidence of top class:  0.8586532602988465
accuracy:  0.7921348314606742
```

```
mean validation dt score : 0.8157588889983256
variance in dt validation score:  4.4896070520911876e-05
average confidence of top class:  0.5
accuracy of decision tree:  0.8033707865168539
```



Question 2

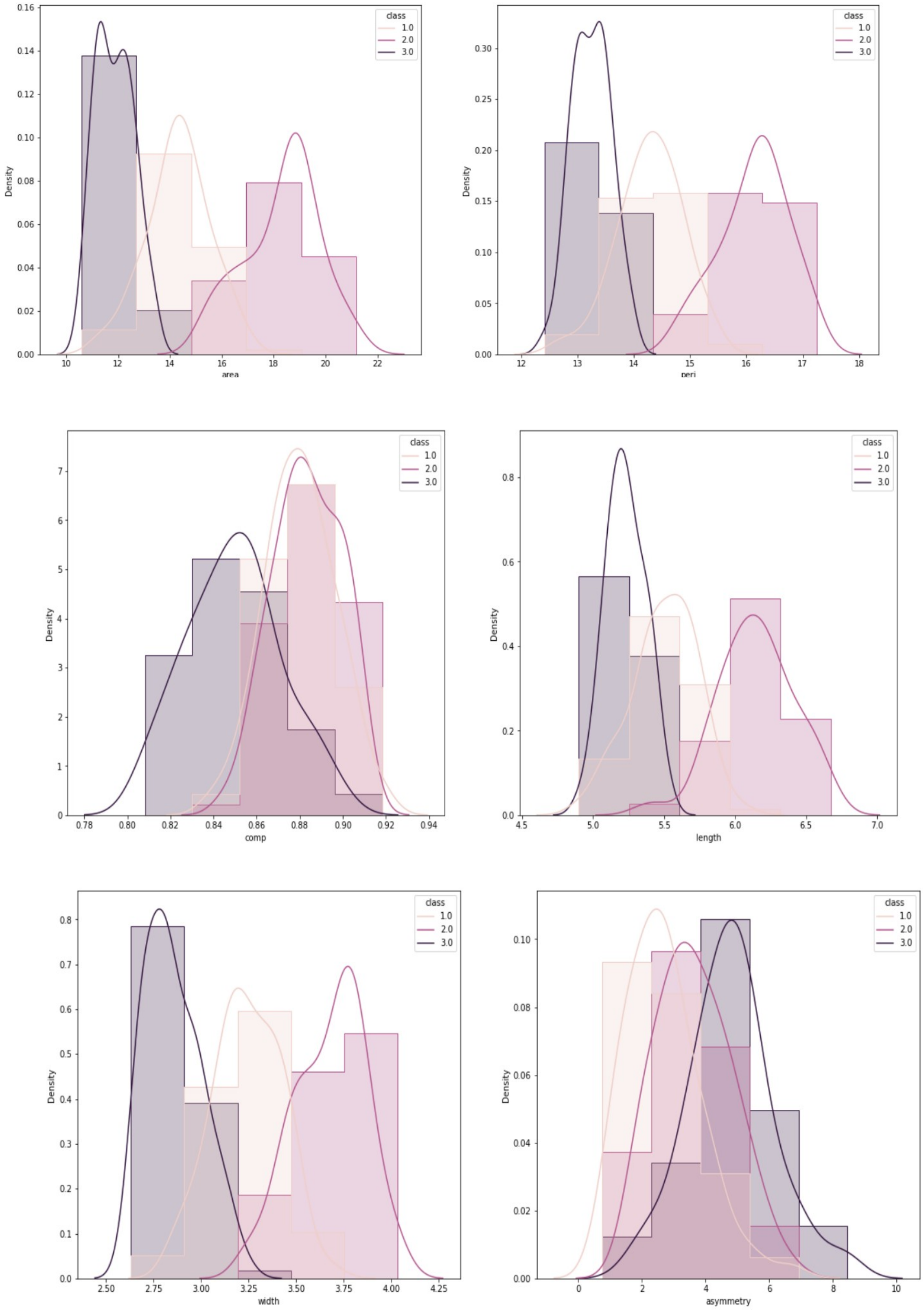
***necessary preprocessing was performed**
Subtask a: Plotting histograms



- Further histplots for individual classes was also plotted and we found the maximum features too be retained at 5 and 7 bins.
- 3 is unable to capture the variations in density across the data while 9+ bins are capturing way too minute features.
- On this analysis, 5 bins are chosen. (hisplots of same can be found on next page:

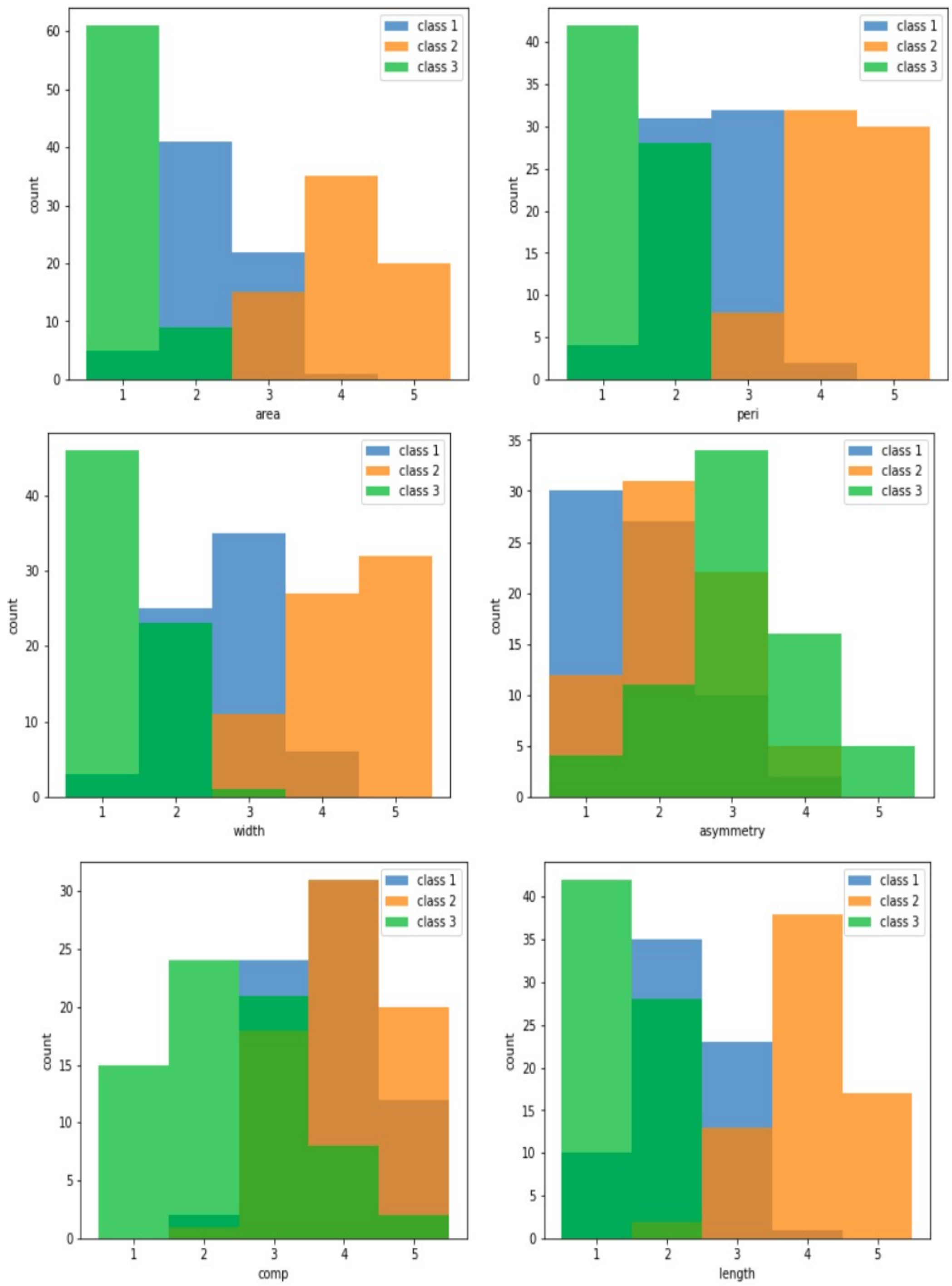
Subtask b: calculating priors
Calculation of priors was straight forward

```
Prior of class 1:  0.33014354066985646
Prior of class 2:  0.3349282296650718
Prior of class 3:  0.3349282296650718
```

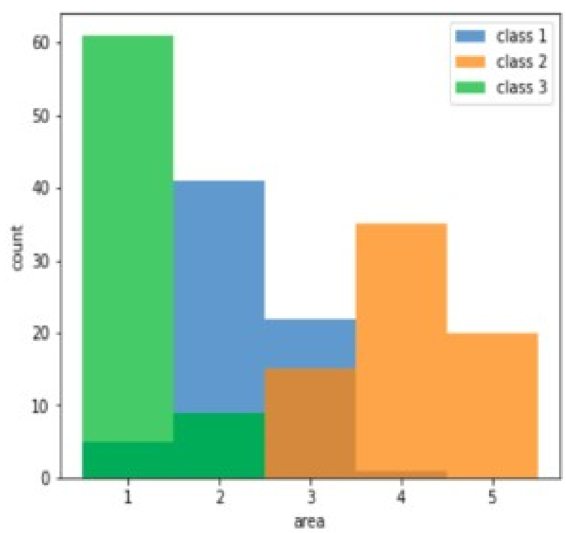


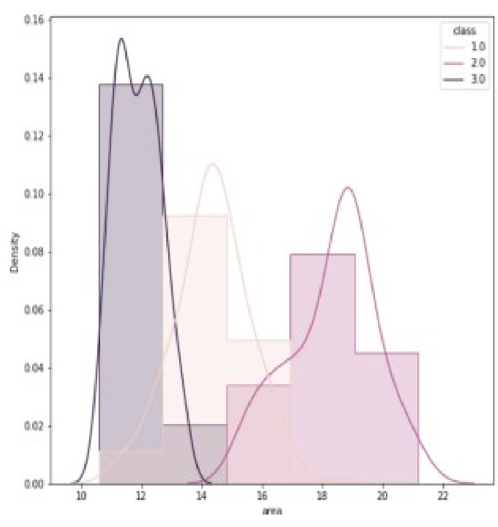
Subtask c: Discretizing into bins (scratch) (and Subtask e)

- Number of bins chosen is 5.
- Here we bin the continous features on the principle of **equal width binning**.
- Visualization of our implementation:



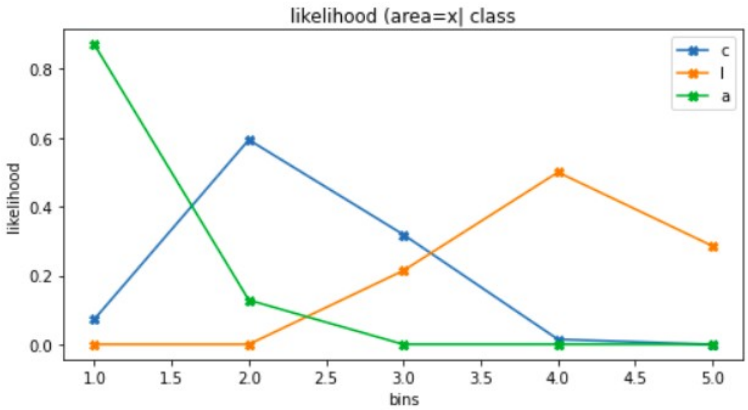
• Clearly we can see, our implementation is representative of the inbuilt.

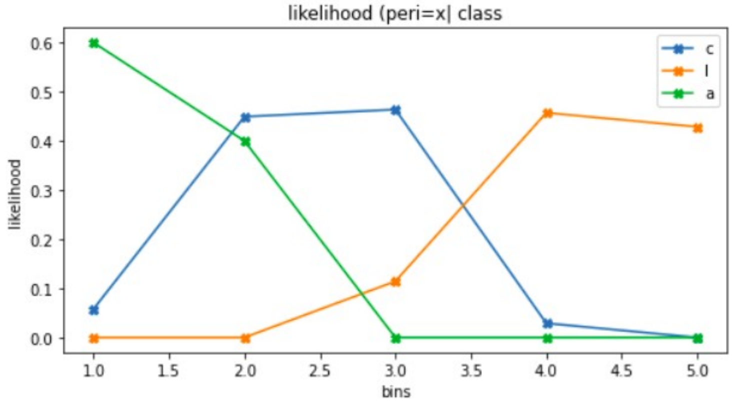


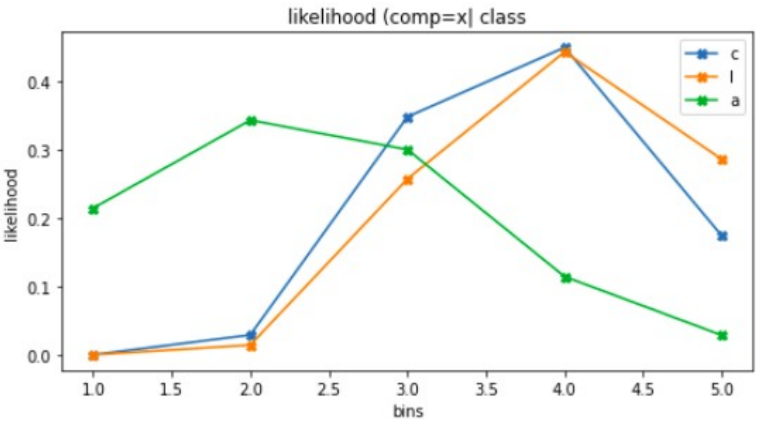


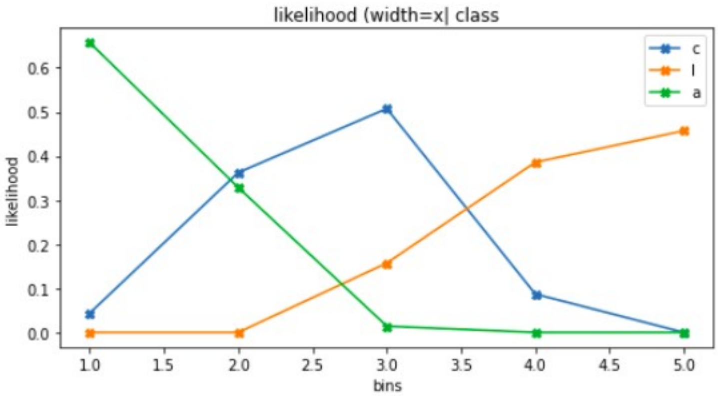
Subtask d: calculating likelihood and plotting

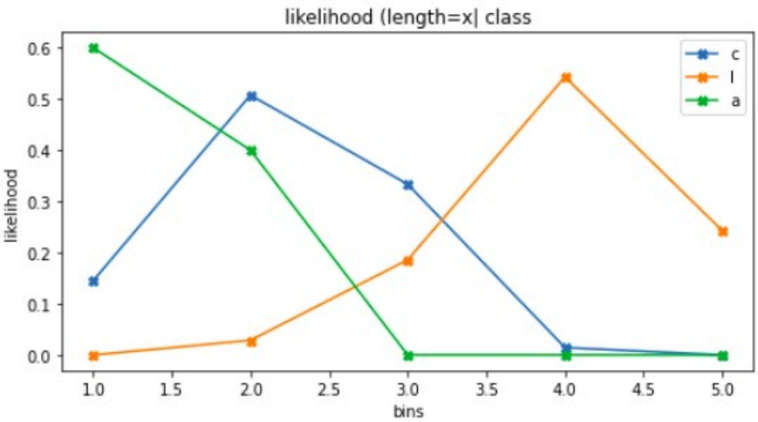
Calculation od likelihood was straight-forward

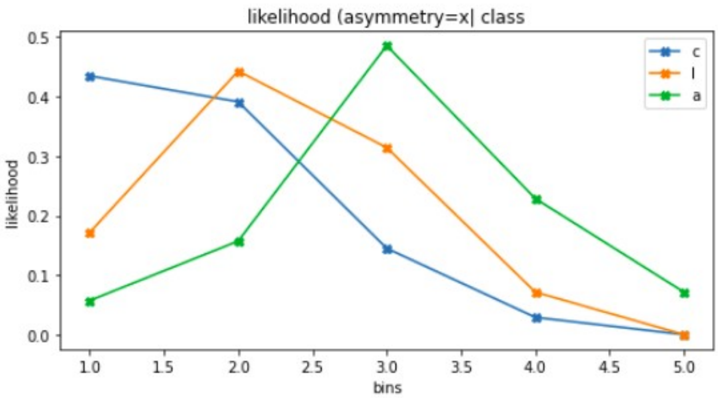


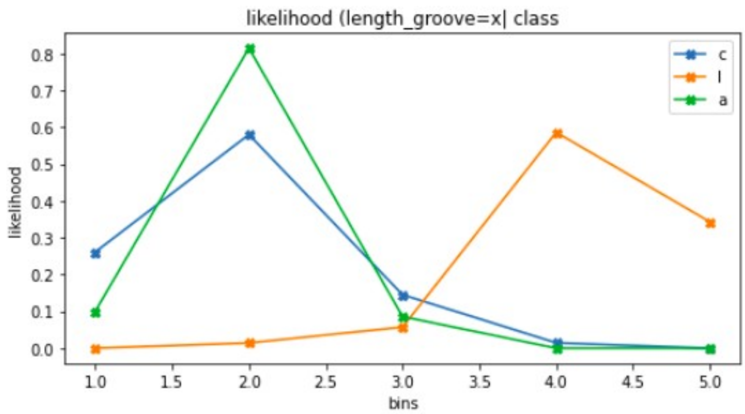








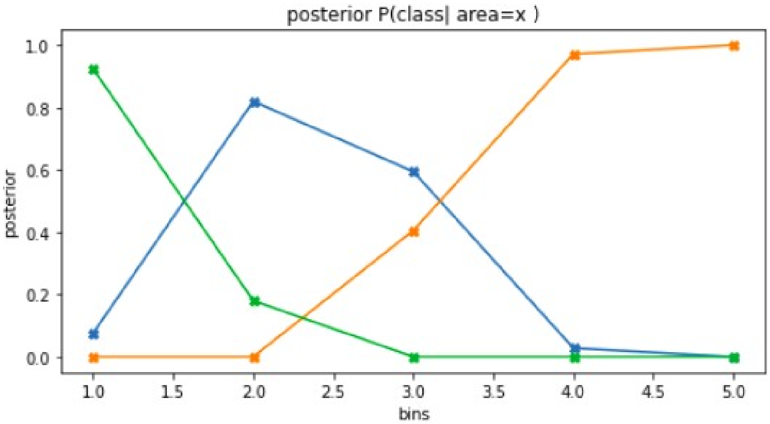


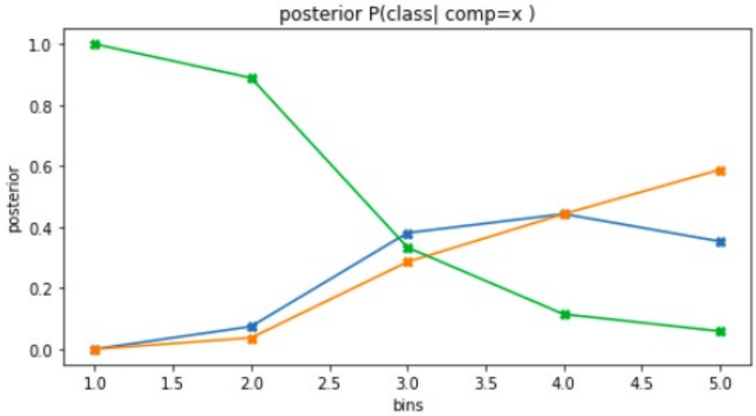


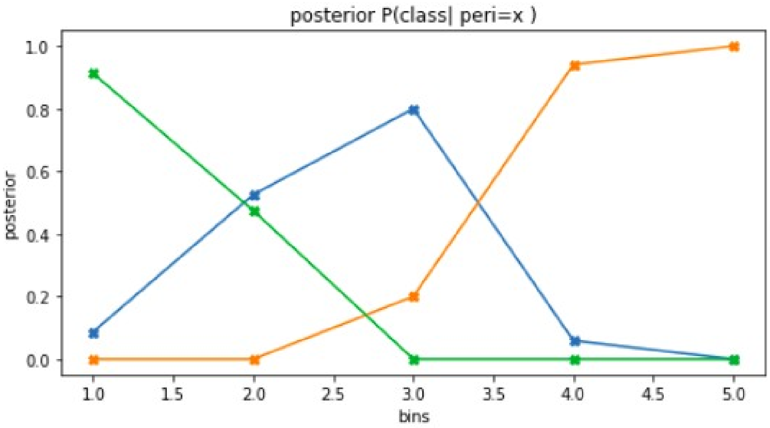
Subtask e: already done in Subtask c.
Subtask f: Calculating and plotting posteriors
Posteriors was calculated feature-wise.

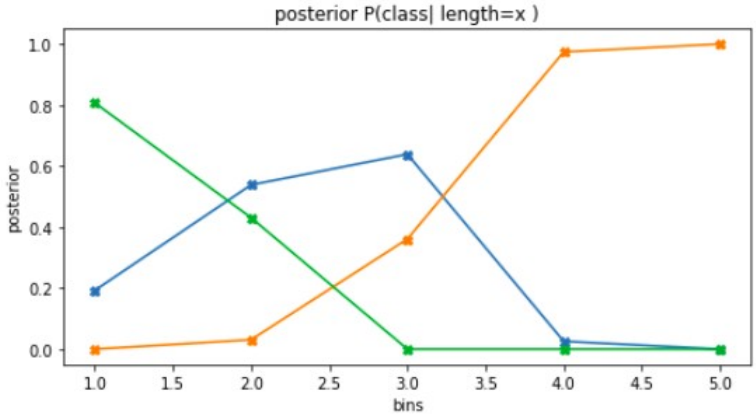
The calculation was straight-forward and formula based.

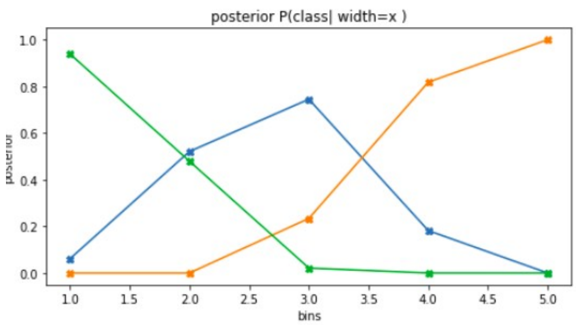
$$\boxed{posterior = \frac{likelihood \times prior}{evidence}}$$

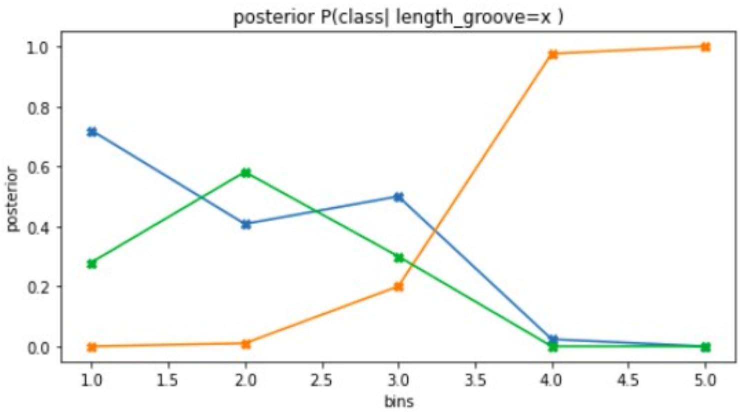


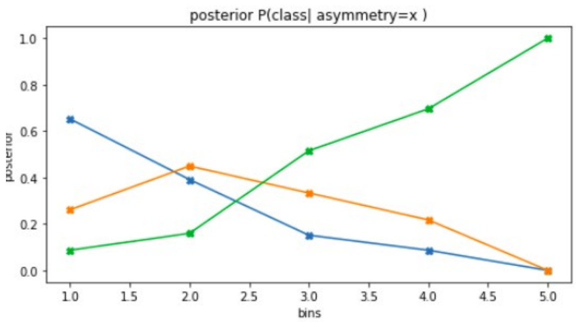




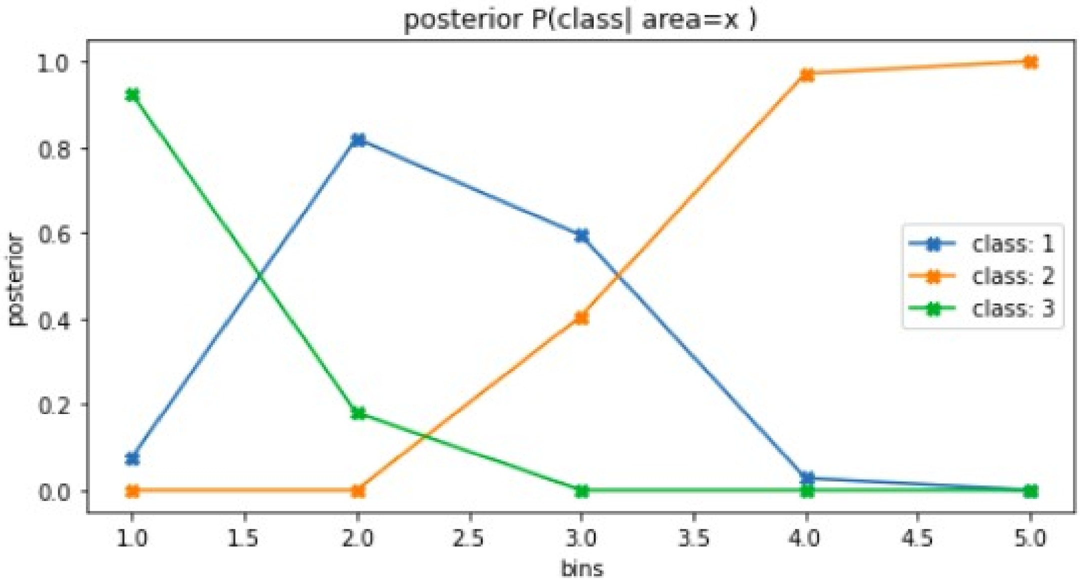








This is done to calculate all the feature-wise posteriors beforehand and also to check if the formula is working (posteriors across a vertical on a graph should sum up to 1).



- Lets just analyze the first posterior graph
- P(class | area= x)
    - smaller values of the feature area(which falls in initial bins) have high chances of corresponding to class 3
    - intermediate values correspond to class 1 and larger values of area correspond to class 2.
- Such as so, other features can also be generalized and give us the results as demonstrated in questions 1.