PRML LAB 3
Naïve Bayes Classifier

Vansh Agarwal
B21AI042

Q1. 1 Preprocessing & Visualisation

The imported dataset has been visualised with `data.head()`. The number of null entries and unique values for each column has been printed. **PassengerId** is just the index of each row and doesn't contribute to the classification task, hence we decide to drop it.

Handling Missing Values

Also **Cabin** feature has a huge ratio of missing entries i.e. 687 out of a total of 891 entries have missing value for Cabin column. Thus, we plan to drop this column as well.

The only columns with missing values left now are **Age** and **Embarked**. We use imputation technique to fill in the missing values.
- Embarked is categorical column, hence we use the most frequent value for imputation
- Age is numerical column, hence we use the mean for imputation.

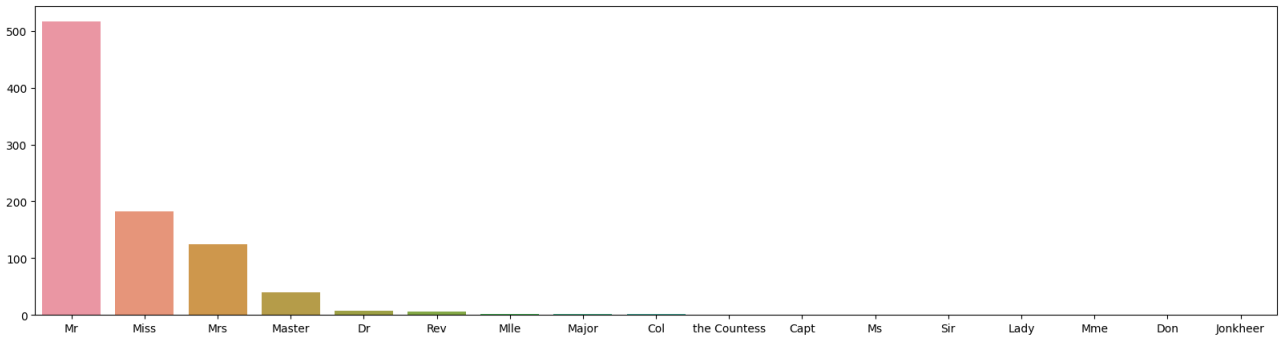
Handling Categorical Columns

The categorical columns in dataset are **Name, Sex, Ticket and Embarked.**
Sex, Ticket & Embarked have been dealt with using Ordinal Encoding. Ordinal Encoding maps each unique value in a categorial feature to a unique integer. For eg: In Sex, Male becomes 1 and female becomes 0.

Working with "Name"

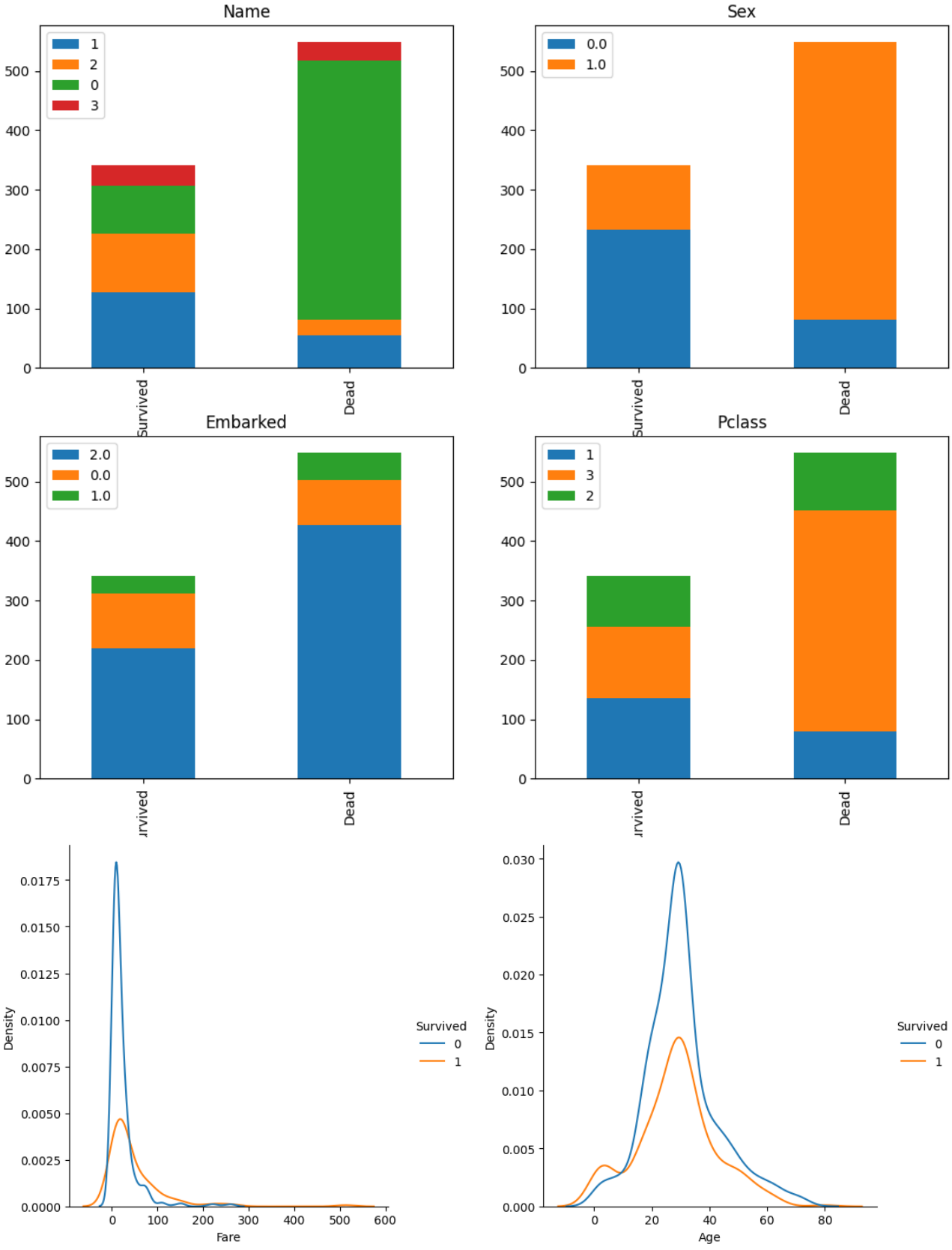
Name has unique value for each entry and might not be useful for classification task. But we plan to extract the titles from these names which might represent the social class of the person and may improve the model performance. For eg: we extract "Mr" from "Braund, Mr. Owen Harris" and so on. The histogram below shows the frequency of each title(mr, mrs etc.) in the dataset.

Instead of using ordinal encoder, we map Mr to 0, Miss to 1, Mrs to 2 and the rest to 3.



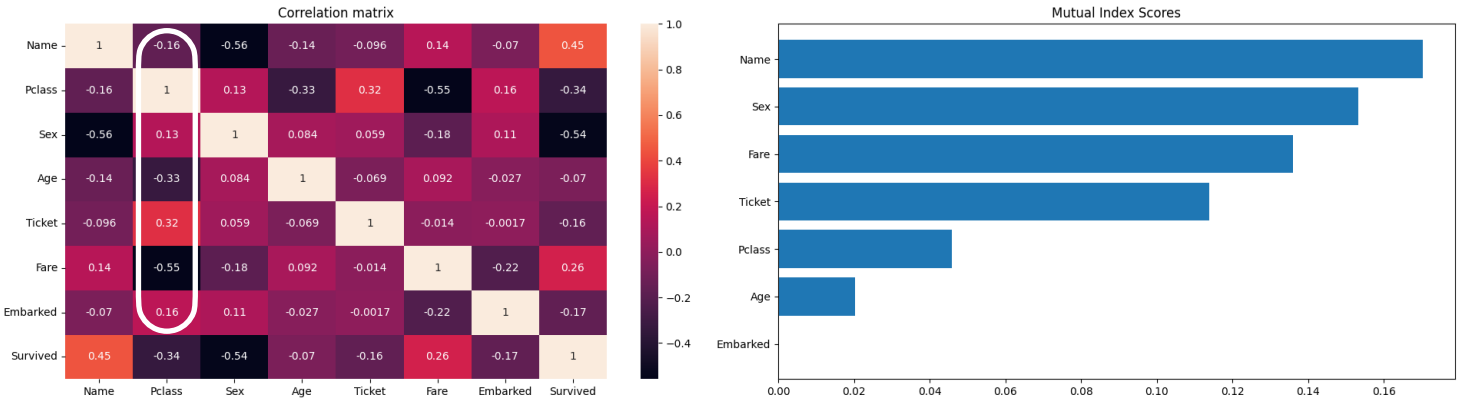
Visualisation

Now we have visualised the discrete features (grouped by Survived or not) using bar graph and continuous features like Fare and Age through their distributions.



Feature Engineering

Correlation matrix and mutual Index Score have been used for Feature Selection.
From the correlation matrix we can see that '**Pclass**' has high correlation with other features as highlighted in the figure.
Highly correlated features more or less represent the same information. So it is preferable to reduce the dimension by dropping one of these features. This also reduces redundancy in some way. Hence we plan to drop Class.
The final dataset contains columns - **'Name', 'Sex', 'Age', 'Ticket', 'Fare', 'Embarked'**

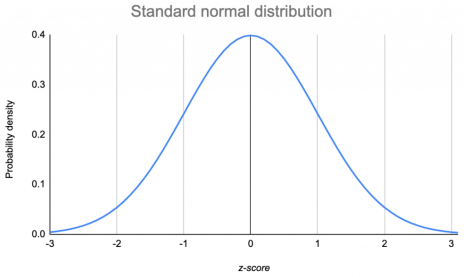


Model Selection & Training

Naïve Bayes classifier has most importantly 3 variants:

- **Gaussian Naive Bayes**: This is used when the features are continuous and follow a normal(gaussian) distribution. The distribution follows the formula given below and looks like a bell shaped curve.

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-(x-\mu)^2}{2\sigma^2}}$$

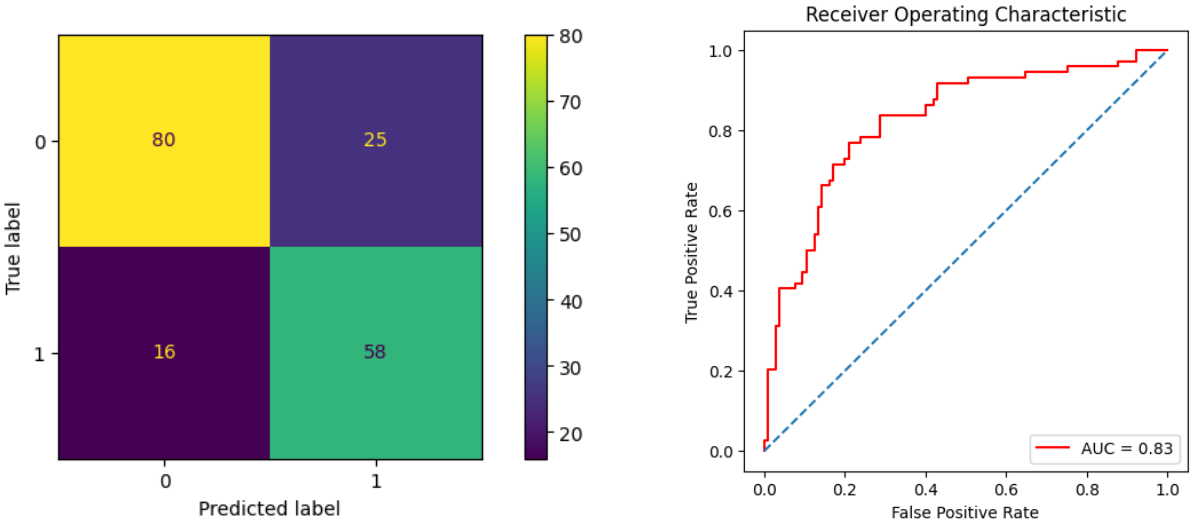


- **Multinomial Naive Bayes**: This is used for discrete variables following multinomial distribution. Multinomial distribution is generalised case for binomial distribution
- **Bernoulli Naive Bayes:** This is used when features are discrete, having binary values like Yes or No / 0 or 1.

For our task we will be using Gaussian Naive Bayes Classifier because of the reasons mentioned above. Most of the features in our dataset are continuous, following Gaussian distribution, as shown in the visualisation part.

Q1.3 Model Performance

Model performance has been measured using **Confusion matrix** and **Receiver Optimisation Characteristics (ROC)** which have been shown below.



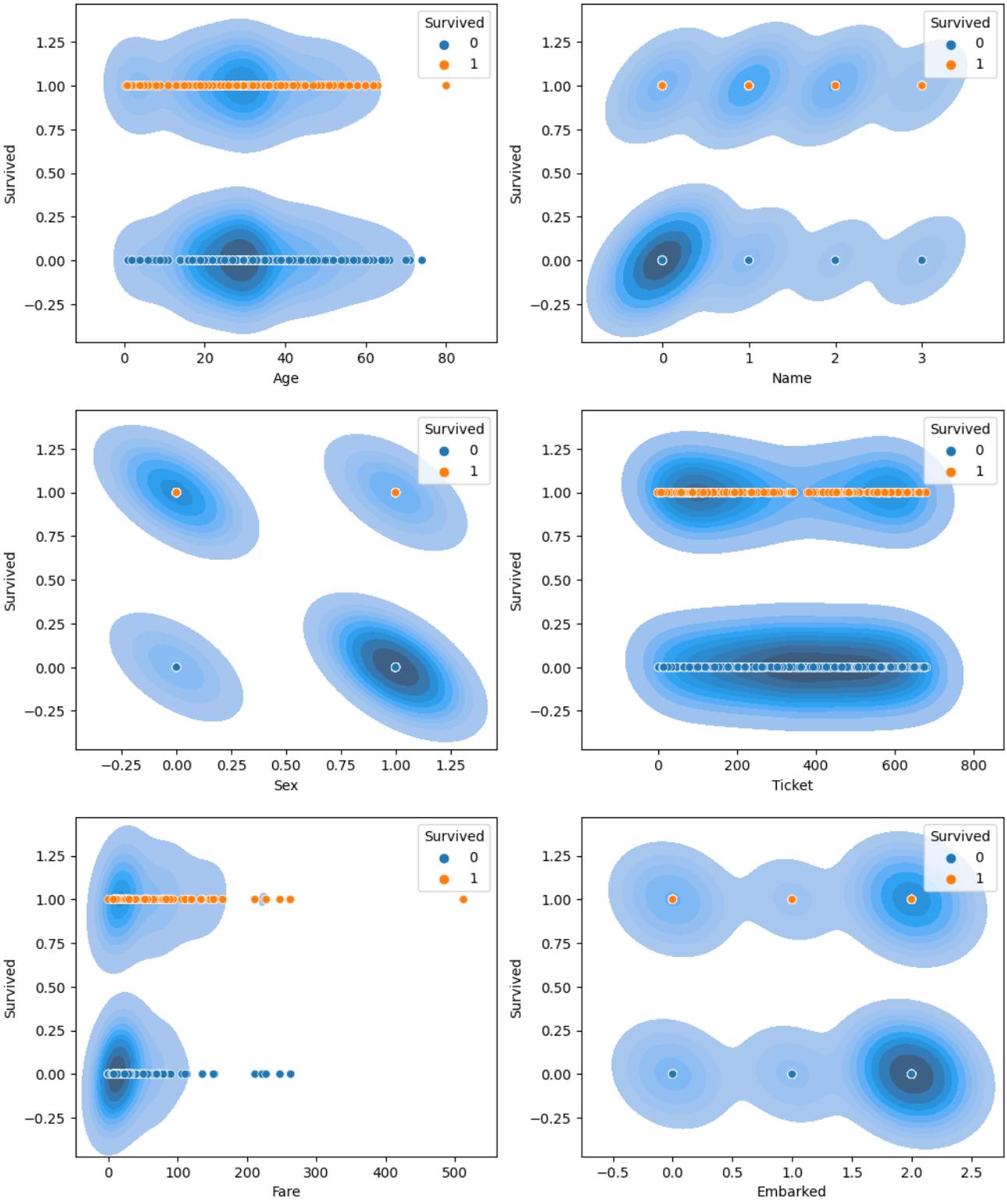
Explanation: If the graph lies more to the left of the dotted diagonal that means, at that particular threshold, the **true positive rate** - proportion of correct predictions for the people who actually Survived increases, and the **False positive rate -** proportion of incorrect predictions for those who didn't survive decreases. So the ROC graph which is more to the left and has more area enclosed with x-axis (representing False positive rate) performs well.
In our case, we achieved an AUC score of **0.83.**

Q1.4 5-Fold Cross validation

In K-fold Cross Validation technique we divide our training set into k subsets. Then for each i^{th} subset we train the model on remaining (k-1) subsets and test it on the i^{th} subset. Finally we take the mean of all the scores. In our case k = 5.
The mean of the 5 fold cross validation scores for our model comes out to **0.7738.**

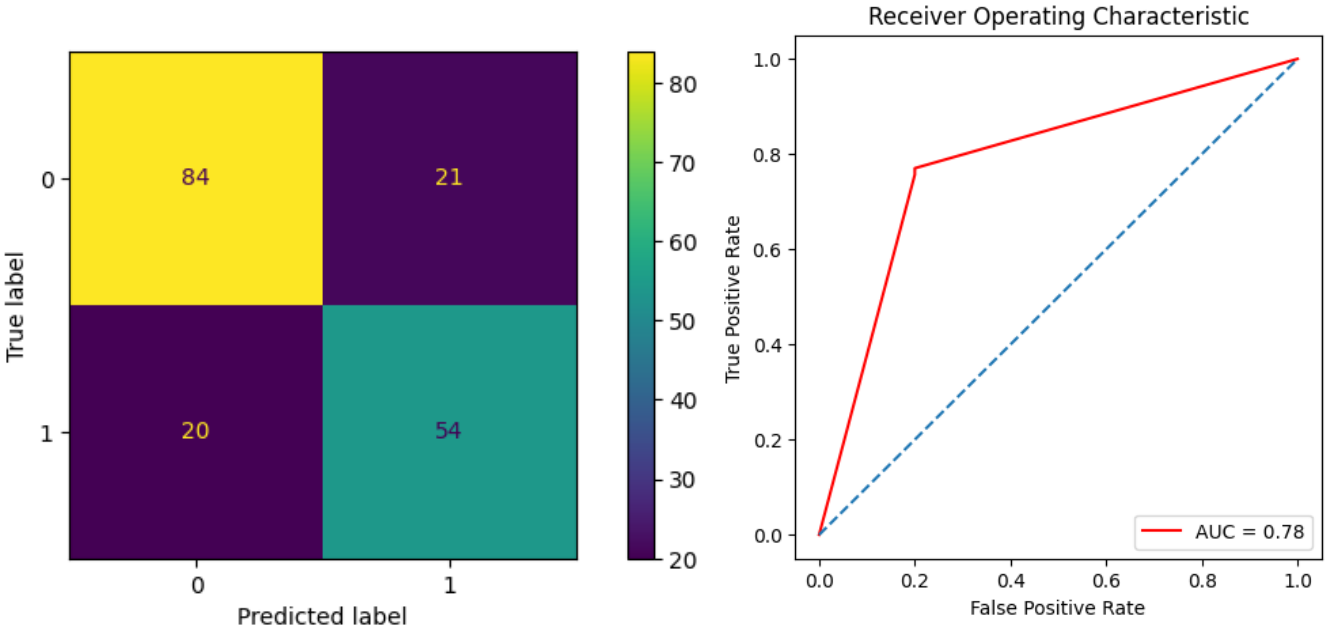
The probabilities for being Survived for the first 5 samples in y_test as predicted by our model are: [0.896, 0.013, 0.017, 0.845, 0.943]

Q1.5 Contour Plots



Q1.4 Decision Tree Classifier

The Decision tree classifier gives a cross validation score of **0.75.**
AUC Score of **0.78.**



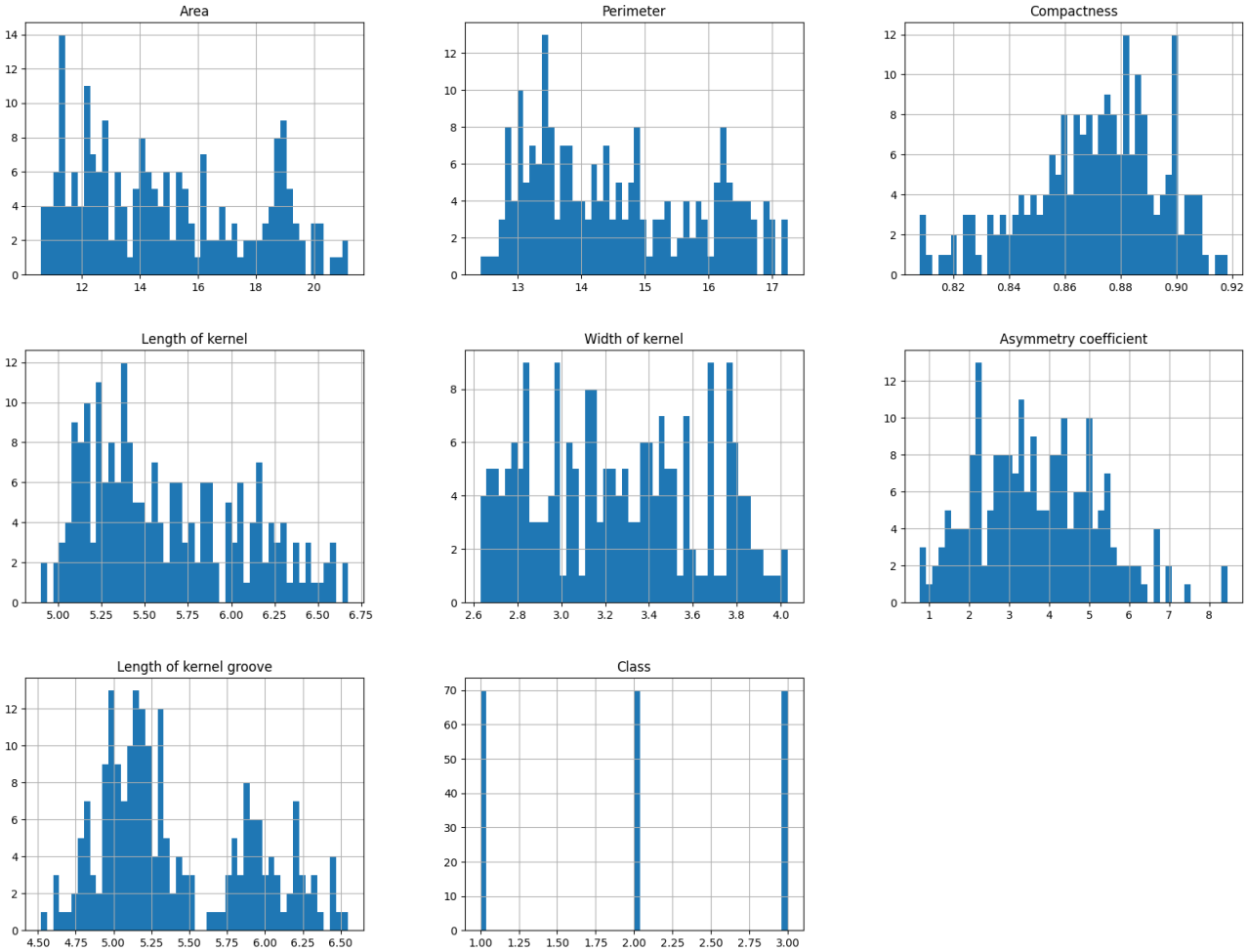
In our case Naive Bayes works slightly better than Decision Tree Classifier.

Explanation: Decision Tree is discriminative model (i.e. it tries to make boundaries between classes). Decision Tree make boundaries by recursively partitioning the space in a manner that reduces Gini Impurity and maximises information Gain.

Naive Bayes is generative model (i.e. it tries to predict how the classes were generated). Naive Bayes assumes every feature is independent and gives same level of importance to every feature.

Naive Bayes also performs better for categorical features as compared to Decision Tree, and our dataset has more number of categorical features. Most of our features are mutually independent (as we removed highly correlated features like Pclass during feature selection). Naive Bayes works on the assumption that features are independent of each other. This is another reason why Naive Bayes performs well in this task.

Q2. 1 Histogram



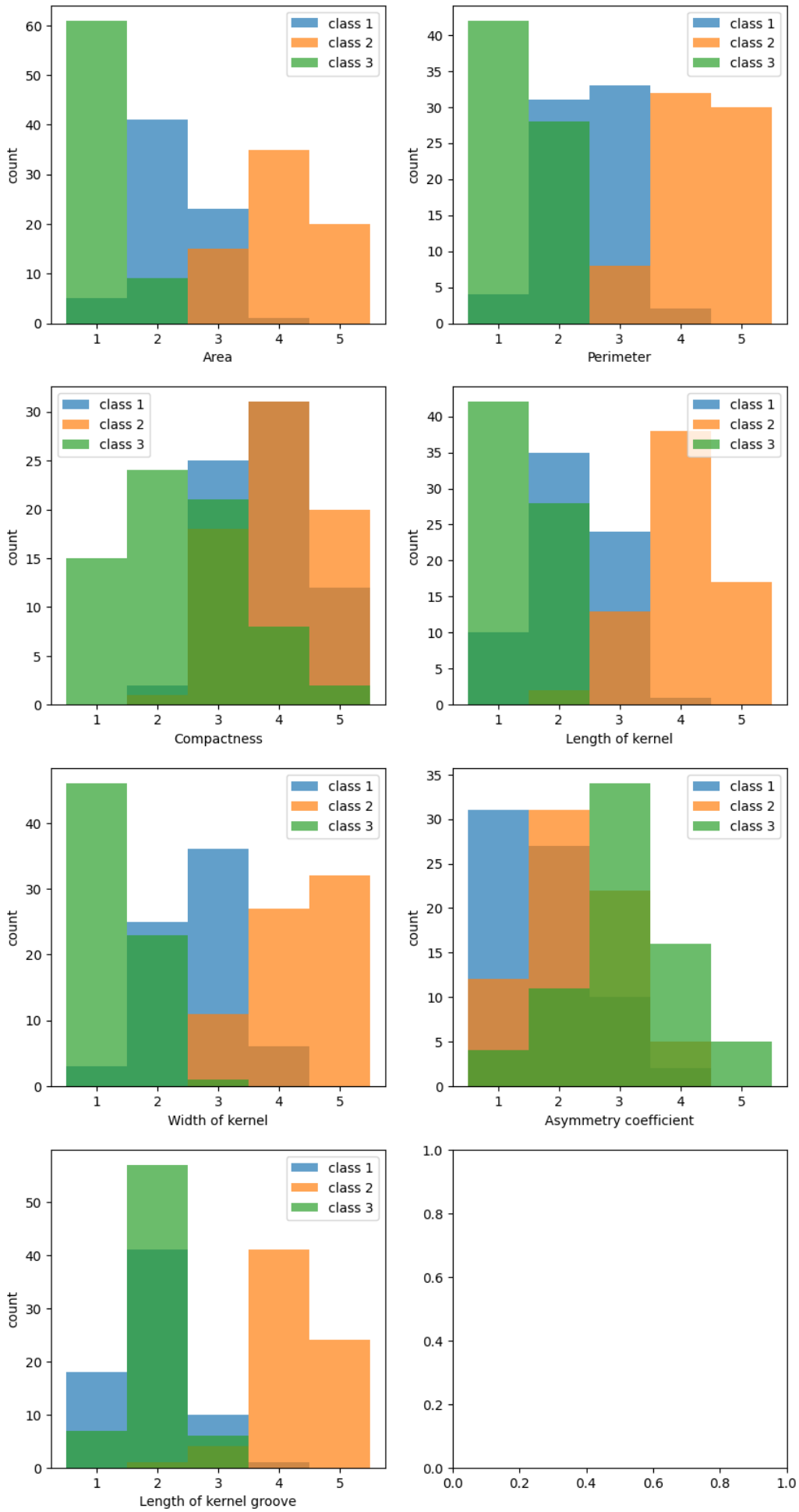
Q2. 2 Prior Probability

Prior Probability of a class is defined as the number of instances of the given class divided by total number of instances.
The prior probabilities for each of the three classes in our case come out to $\mathbf{1/3 = 0.33}$

Q2. 3 Discretisation

Process:
`n_bins` = 5

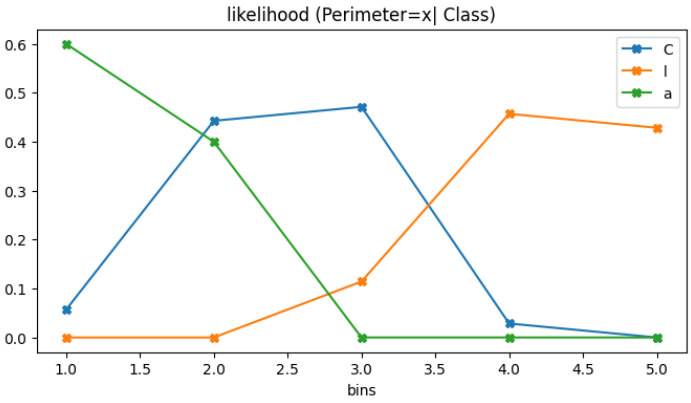
- For each feature the range has been calculated i.e. the `max_value - min_value`. The range has been divided into 5 parts (n_bins).
- Each feature has been grouped by class
- Each entry in feature has been identified with the bin that it falls in and replaced by the

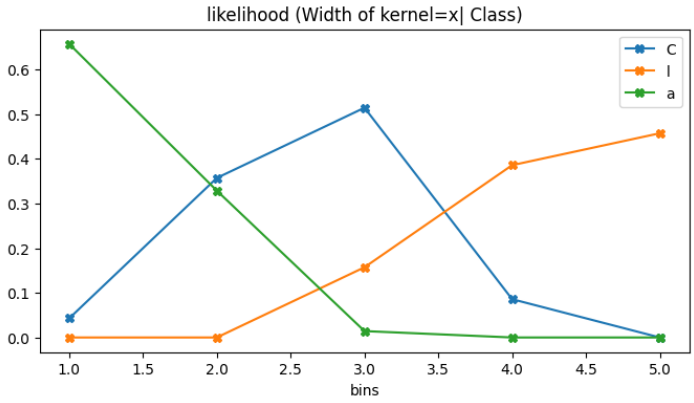

class 1
class 2
class 3
count
Area
Perimeter
Compactness
Length of kernel
Width of kernel
Asymmetry coefficient
Length of kernel groove

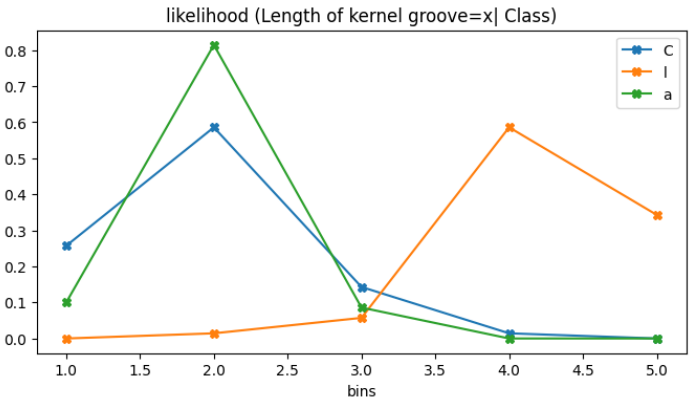
Q2.4 Likelihood

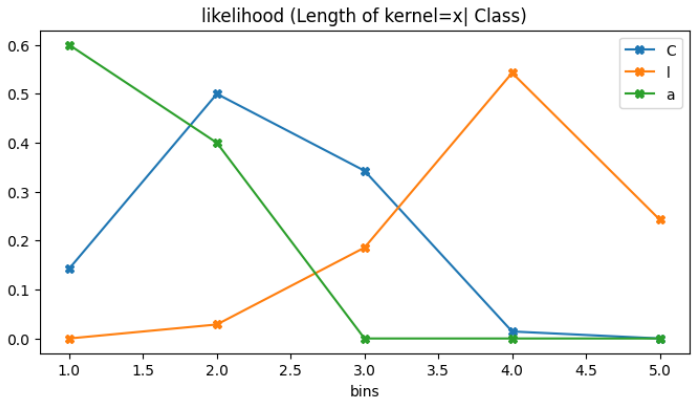
Likelihood is the probability that feature equals a certain value given the entry falls in a certain class. The plot is shown below:

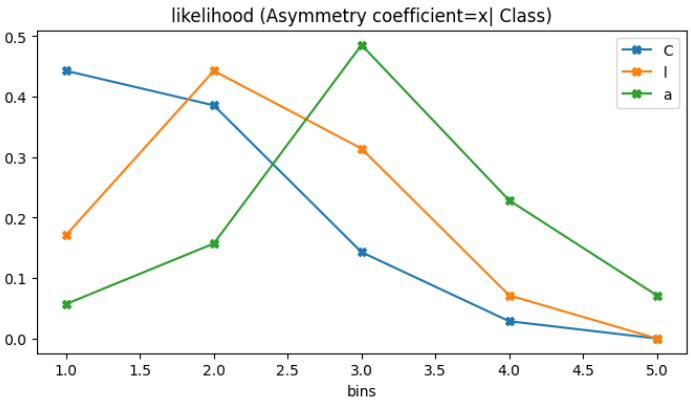
- For each **feature**, for each **bin**, for each **class** we have found the likelihood i.e. (the number of instances where feature value is bin and the class is class) divided by (the total number of instances of that class)

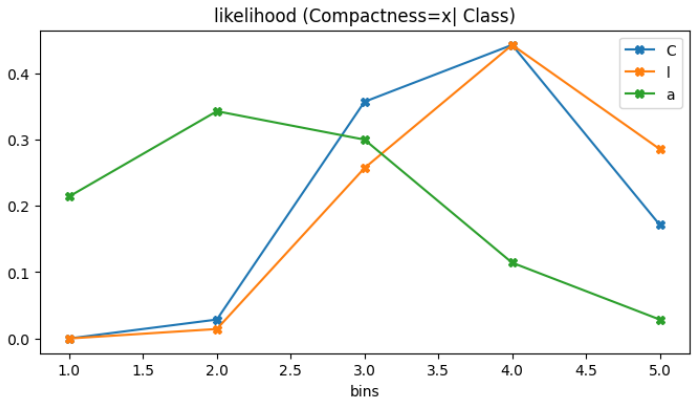


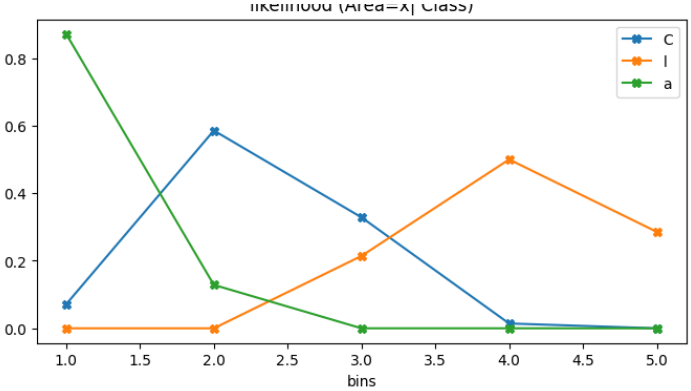












Q2.5 Posterior probability

Posterior Probability is the probability that instance belongs to a certain class given its feature has a certain value.

- For each **feature**, for each **bin**, for each **class** we have already found the likelihood. P(Feature | Class)
- Then we have found the evidence i.e. the probability that feature equals bin. P(Feature)
- We have found prior already i.e. the probability of each class. P(class)
- Next we have found the posterior probability corresponding to each feature and class using the formula:

$$\underset{\text{Posteriori}}{P(y|X)} = \frac{\overset{\text{Likelihood}}{P(X|y)}\overset{\text{Prior}}{P(y)}}{\underset{\text{Predictor Prior}}{P(X)}}$$