CSL2050 PATTERN RECOGNITION AND MACHINE LEARNING

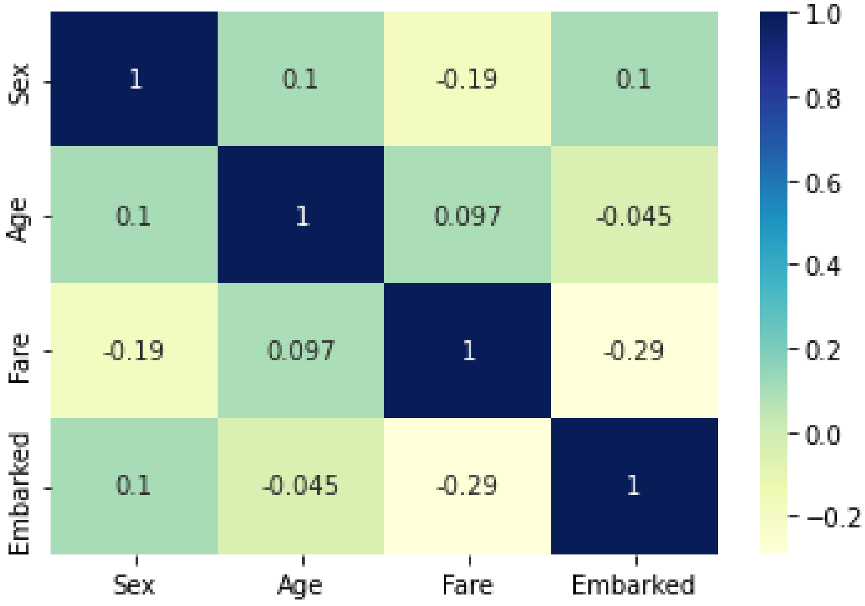
LAB REPORT-03

NAME:	SAMEER SHARMA
ROLL NUMBER:	B21CS066
LAB TITLE:	BAYES CLASSIFICATION

Q1)

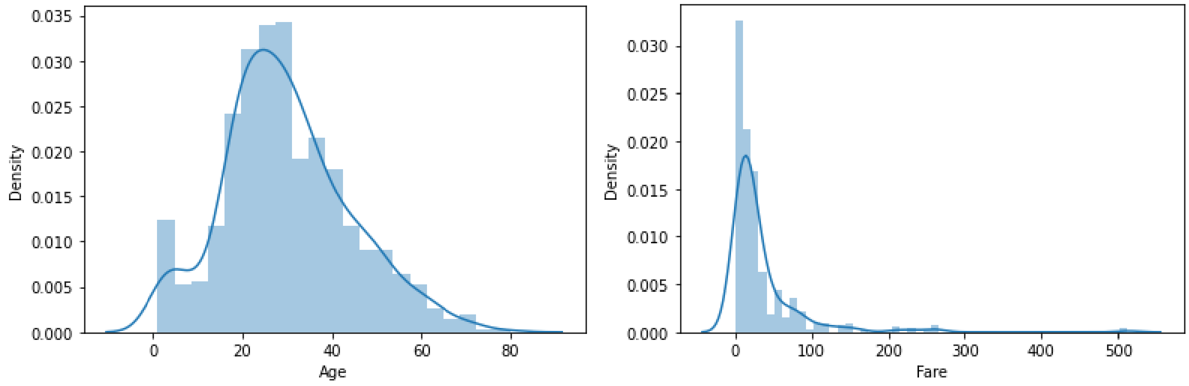
Part 1)

In this part preprocessing on the data is done. For eg: loading data, Dropping unnecessary columns, dropping null values, encoding the data and converting features to their desired data type. Here I only chose Fare, Sex, Embarked and Age as features for my model since they are more relevant and correlated to the output. I got this correlation by plotting the heat map of the features against output column and finding the pearson score. I also dropped such columns which are correlated to other columns like "Pclass" which is correlated to "Fare". Now, the features are not much correlated and can be used to feed a naïve bayes classifier

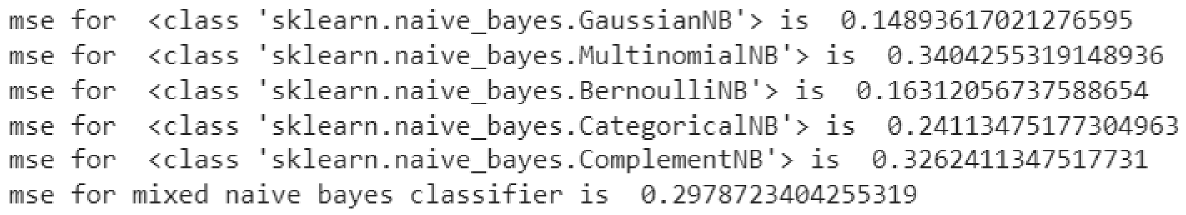


Part 2)

Gaussian Naïve Bayes classifier is best for the given data since “Age” and “Fare” which are the input features nearly resembles the Gaussian distribution. One can confirm from the below distribution plots of “Age” and “Fare”.



One can also confirm that the Gaussian Naïve Bayes classifier is best for the given data by evaluating different models on the data set. I have calculated mean squared error in predictions by different models and got Gaussian to be better than others.

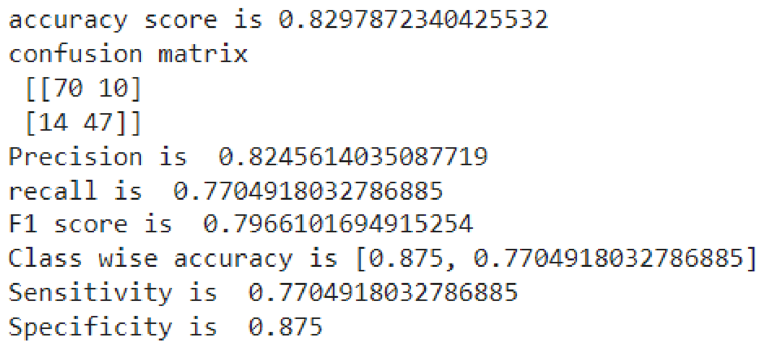


```
mse for  <class 'sklearn.naive_bayes.GaussianNB'> is  0.14893617021276595
mse for  <class 'sklearn.naive_bayes.MultinomialNB'> is  0.3404255319148936
mse for  <class 'sklearn.naive_bayes.BernoulliNB'> is  0.16312056737588654
mse for  <class 'sklearn.naive_bayes.CategoricalNB'> is  0.24113475177304963
mse for  <class 'sklearn.naive_bayes.ComplementNB'> is  0.3262411347517731
mse for mixed naive bayes classifier is  0.2978723404255319
```

Here, mixed Bayes classifier is made from two different classifiers i.e. Gaussian and Categorical Naïve Bayes classifiers trained on numeric/continuous and categorical features respectively and later combined to give predictions.

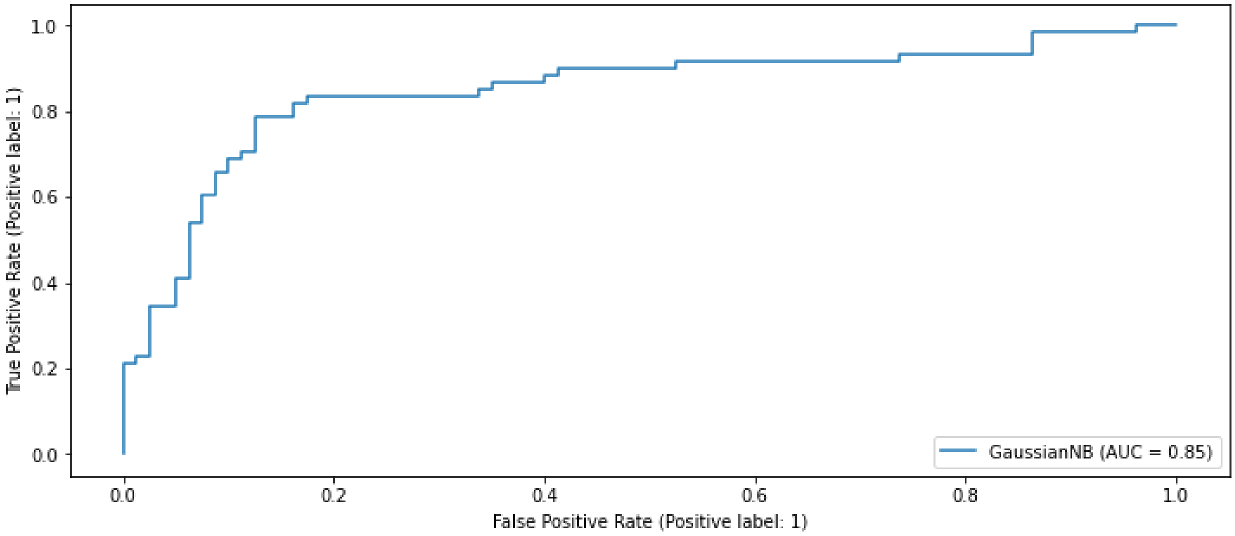
Part 3)

Performance of the model on the testing data is as follows



```
accuracy score is 0.8297872340425532
confusion matrix
 [[70 10]
 [14 47]]
Precision is  0.8245614035087719
recall is  0.7704918032786885
F1 score is  0.7966101694915254
Class wise accuracy is [0.875, 0.7704918032786885]
Sensitivity is  0.7704918032786885
Specificity is  0.875
```

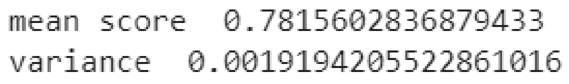
And ROC curve is



Therefore area under curve of the ROC curve is 0.85

Part 4)

The obtained Cross Validation score is as follows

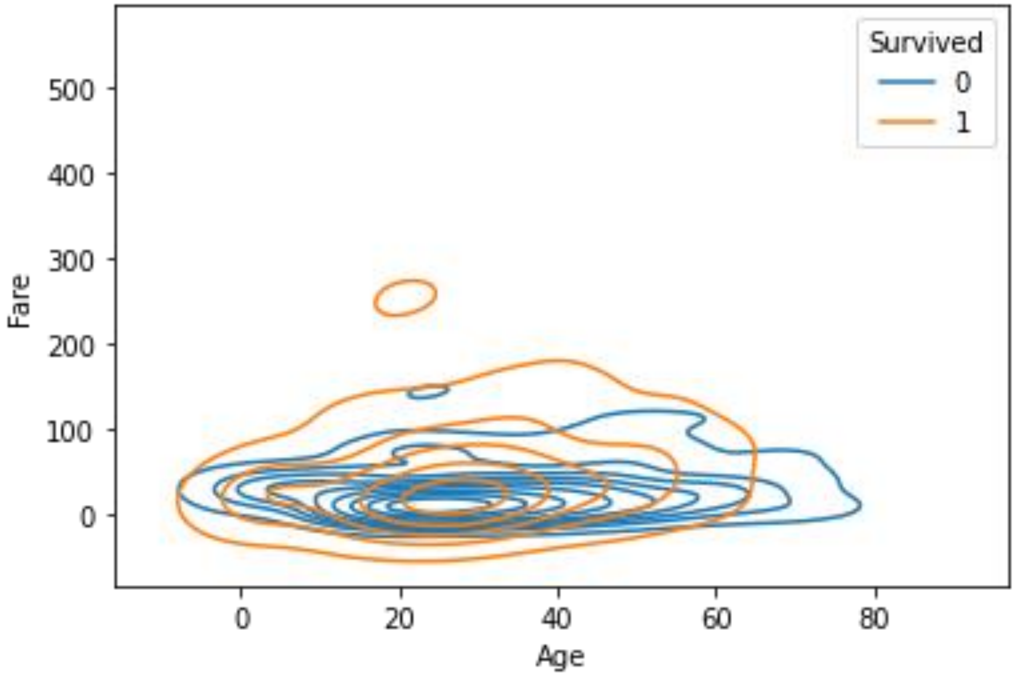


```
mean score  0.7815602836879433
variance  0.0019194205522861016
```

And probabilities for top class for each test data is also printed in the colab file

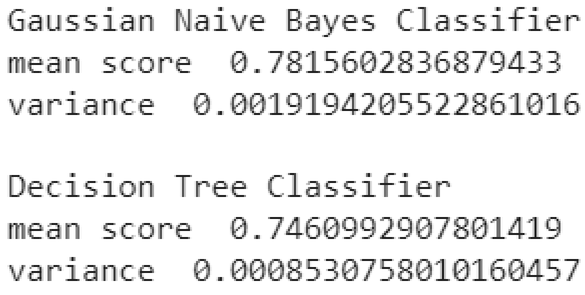
Part 5)

Contour plot of Fare vs Age is given as follows



Part 6)

The accuracy of the two classifiers are nearly same but Gaussian Naïve Bayes classifier performed better than the Decision Tree Classifier. This can be because Decision Tree Classifiers perform better when we have only categorical data. Otherwise it needs much larger dataset for performing better when provided with continuous data. On other hand, Gaussian performs much better when provided with continuous data.



```
Gaussian Naive Bayes Classifier
mean score  0.7815602836879433
variance  0.0019194205522861016

Decision Tree Classifier
mean score  0.7460992907801419
variance  0.0008530758010160457
```

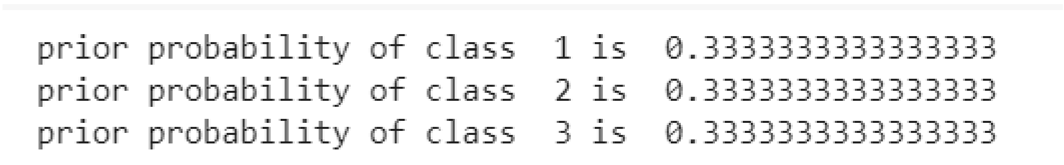
Q2)

Part (a)

- Plotted the Histograms for the given data which you can see in the colab file.

Part (b)

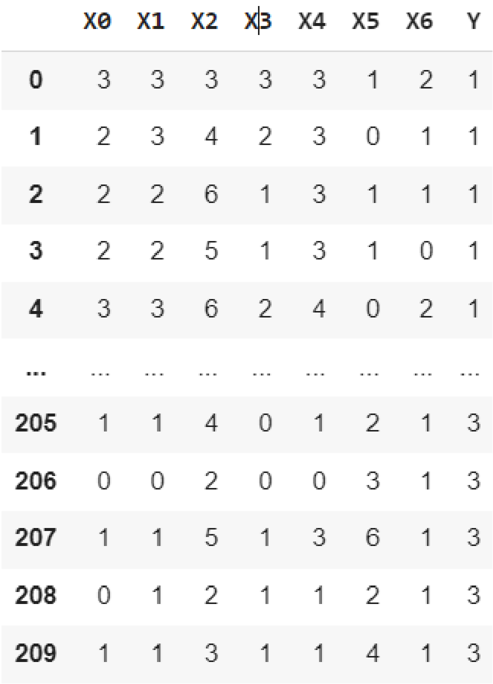
Prior Probabilities of each class are as follows



```
prior probability of class  1 is  0.3333333333333333
prior probability of class  2 is  0.3333333333333333
prior probability of class  3 is  0.3333333333333333
```

Part (c)

Discretized the dataset into seven bins and assigned every data a bin number based on in which range the magnitude of data is lying . The bins are of constant width (range).



	X0	X1	X2	X3	X4	X5	X6	Y
0	3	3	3	3	3	1	2	1
1	2	3	4	2	3	0	1	1
2	2	2	6	1	3	1	1	1
3	2	2	5	1	3	1	0	1
4	3	3	6	2	4	0	2	1
...	...	...	...	...	...	...	...	...
205	1	1	4	0	1	2	1	3
206	0	0	2	0	0	3	1	3
207	1	1	5	1	3	6	1	3
208	0	1	2	1	1	2	1	3
209	1	1	3	1	1	4	1	3

Part (d)

Calculated the likelihood/ class-conditionals for every bin number (bin_no) given the feature column (col) and class (y). This can be done easily by taking the count of data points where the bin number in the specified feature column is equal to bin_no and class (‘Y’) is equal to y dividing it with the count of data points for which class is equal to y.

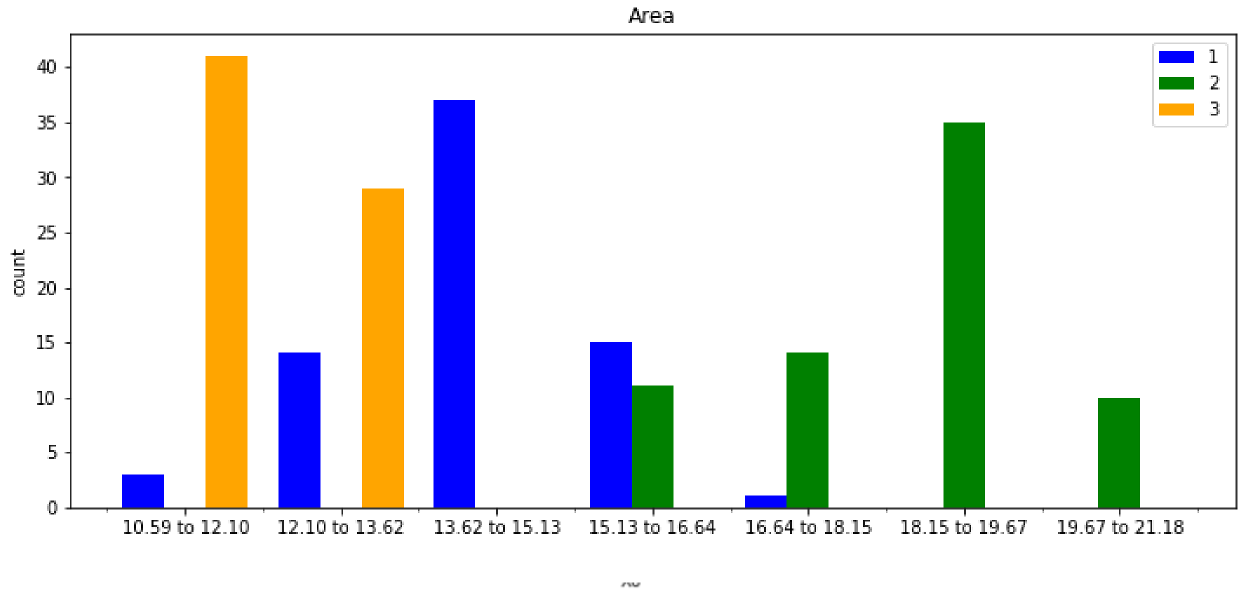
The class conditionals are printed in the colab file alongwith the sum of all conditionals for a given class and feature (which should be ideally equal to one).

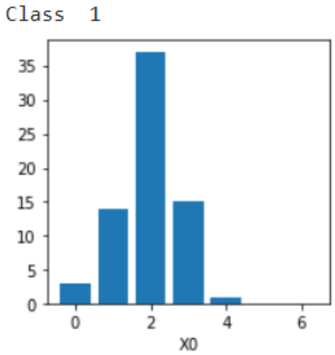
Part (e)

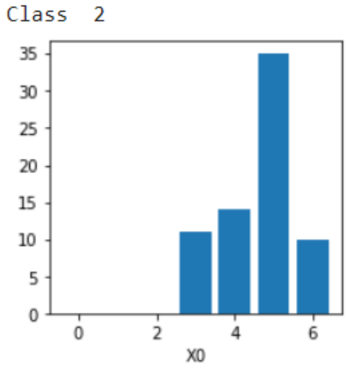
We get the exactly same plots that we got when we used the inbuilt functions in the part (a).

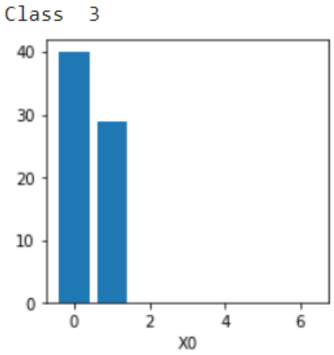
For example let us consider the plots of the area that we plotted in the part (a) and part (e). In part (a) histograms for three classes on the same plot. One can easily observe that the histograms of respective classes in part (a) resemble with one in the part (e).

(Note: Class 1:- Blue, Class 2:- Green, Class 3:-Yellow)



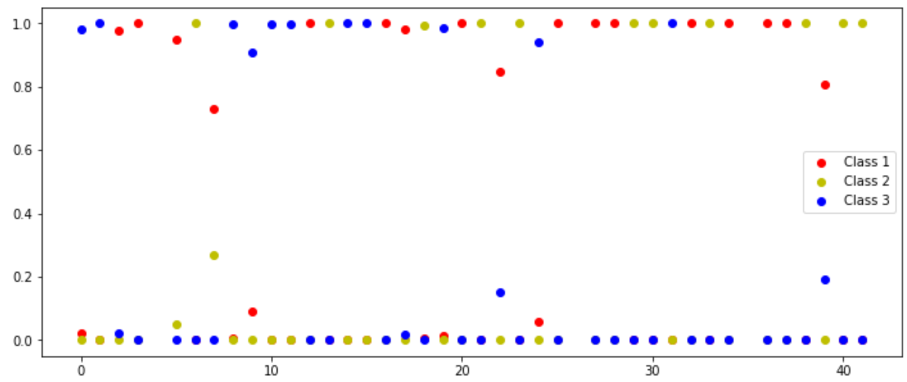






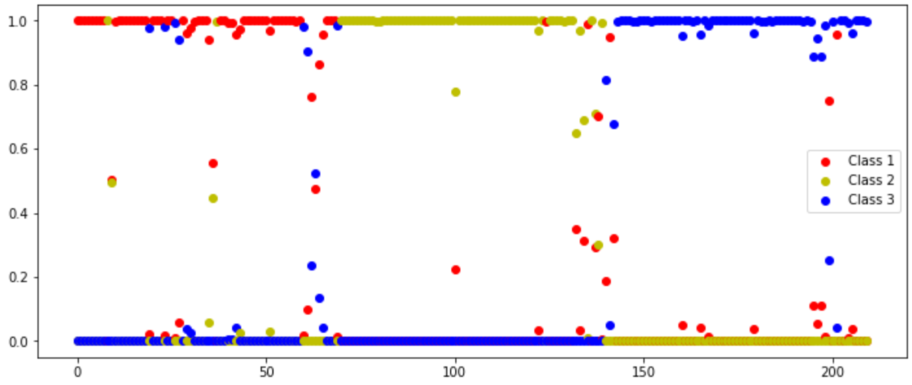
Part (f)

In this part posterior probabilities are calculated. We can easily calculate the posterior probabilities by multiplying the likelihood and prior probability. In this part I calculated the likelihood probabilities from the training data and posterior probabilities are calculated for the testing dataset. I also created the same plot where I used the total dataset for calculating likelihood and posterior probabilities both



When testing data is used to calculate posterior and training data is used for likelihood

•



When whole data set is used to calculate both posterior and likelihood