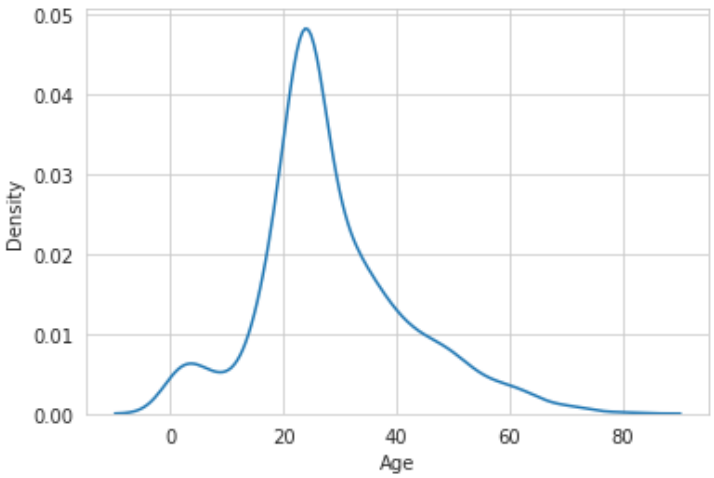
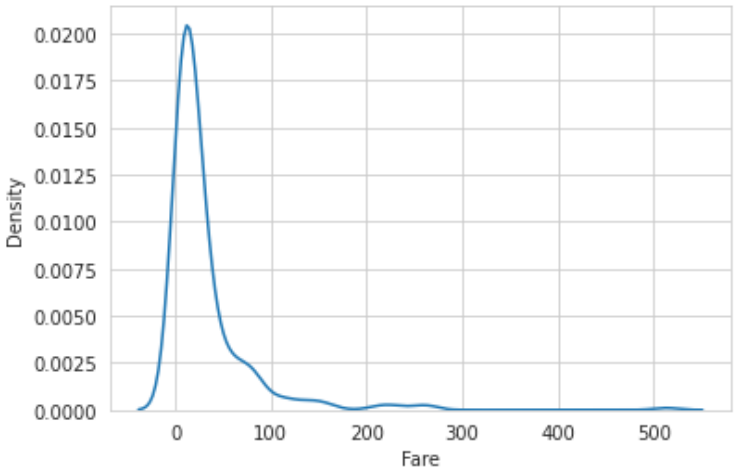
Pattern Recognition and Machine Learning
Report for : Lab 4

Name : Khushi Parikh
Roll No : B20EE029

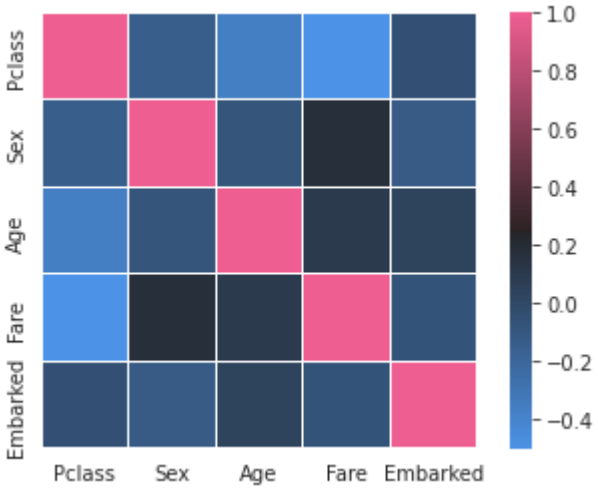
Question 1:

- **(1.1)** After importing the necessary libraries, we imported the data("titanic.csv"). After this I checked the dataset and dropped `'Name','PassengerId' and 'Ticket'` as they were irrelevant to if the passenger survived or not. There were 3 columns which had missing values - 'Cabin' was completely dropped as it had around 80 percent of its values missing. 'Embarked' has only 0.2% values missing and they were replaced by the mode of the column. 'Age' has 19.86% values missing and they were replaced by mode of the column which is the age group with the maximum number of people. The data was split into a training and testing set.
- (1.2) The distribution of the features was plotted and each class was found to have approximately a normal distribution. For example the range of Age and Fare columns -





- The feature correlation is also very less, i.e. the features are independent of each other -



- This shows that the best possible variant of naive bayes classifier is the Guassian Naive Bayes. This assumes that the probability distribution is normally distributed which we can see in the above graphs.
- (1.3) First we find the mean and variance of each target class for each feature. For the training set -

```
Mean          PassengerId    Pclass       Sex        Age       Fare  Embarked
Survived
0          438.199482  2.569948  0.150259  28.726684  22.396447  1.686528
1          433.721519  1.924051  0.687764  27.746835  51.296484  1.540084

Variance           PassengerId    Pclass       Sex         Age         Fare  Embarked
Survived
0          69333.599065  0.505484  0.128013  155.891987  1130.622264  0.402779
1          63030.354323  0.739970  0.215655  203.661326  5248.417292  0.444361
```

- We then divide it into two parts, one for target class 0 and the other for target class 1, each part consisting of a list of the mean and variance for each feature. This helps to ease the calculation.
- We define a function ‘gaussian’ which computes and returns the following value -

$$P(x_i \mid y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp\left(-\frac{(x_i - \mu_y)^2}{2\sigma_y^2}\right)$$

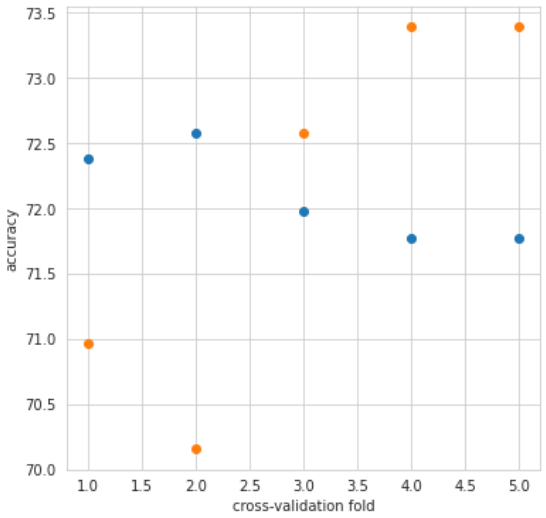
- We then fit the training data. For each row in the testing data, we predict the output value by calculating the probabilities for each feature in each class and then calculating the prior. We then compare it with Y_test to get the accuracy.
- (1.4) 5-fold cross validation was performed. The training data was divided into 5 parts and 4 of these were used for training and the 5th for testing. This was repeated for each fold as the testing data. The average accuracy of each fold was calculated.

- (1.5) Result on each of the folds and the average -

```
Accuracy on training set : [72.37903225806451, 72.58064516129032, 71.9758064516129, 71.7741935483871, 71.7741935483871]
Average of training folds : 72.09677419354838

Accuracy on testing set : [70.96774193548387, 70.16129032258064, 72.58064516129032, 73.38709677419355, 73.38709677419355]
Average of testing folds : 72.09677419354838
```

Using a scatter plot to visualize the accuracy on train(blue) and test(orange) sets -



- The top class in each row was 0. The probabilities are as shown :

```
Max value occurence in fold 0 : 0
Probability of above occurence : 0.717741935483871
Max value occurence in fold 1 : 0
Probability of above occurence : 0.6935483870967742
Max value occurence in fold 2 : 0
Probability of above occurence : 0.7419354838709677
Max value occurence in fold 3 : 0
Probability of above occurence : 0.7338709677419355
Max value occurence in fold 4 : 0
Probability of above occurence : 0.7580645161290323
```

- (1.6) Using the inbuilt GaussianNB classifier, we get the following results. Comparing them with my results -

```
Average score of scratch on training set : 72.09677419354838
Average score of inbuilt on training set : 78.81219903691814

Average score of scratch on testing set : 72.09677419354838
Average score of inbuilt on testing set : 77.23880597014924
```

Also, the result of cross validation compared to my cross-validation in 1.5 -

```
Training set : [82.4        72.         80.         74.19354839 83.06451613]
Testing set : [74.07407407 79.62962963 77.77777778 77.35849057 83.01886792]
```

- (1.7) Using MultinomialNB, we get an accuracy of 66 percent. The results of cross- validation are -

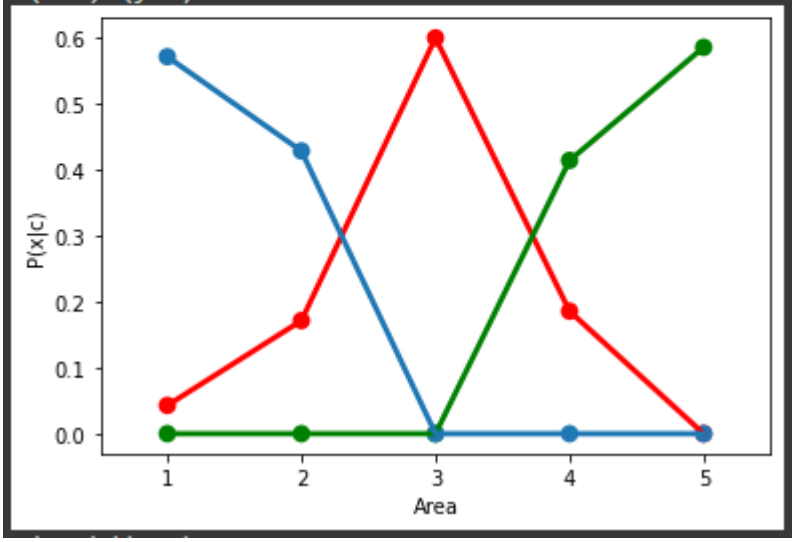
```
Training set : [68.         70.4        68.         66.12903226 79.83870968]
Testing set : [57.40740741 64.81481481 70.37037037 60.37735849 67.9245283 ]

Avg on training set : 70.47354838709677
Avg on testing set : 64.17889587700908
```

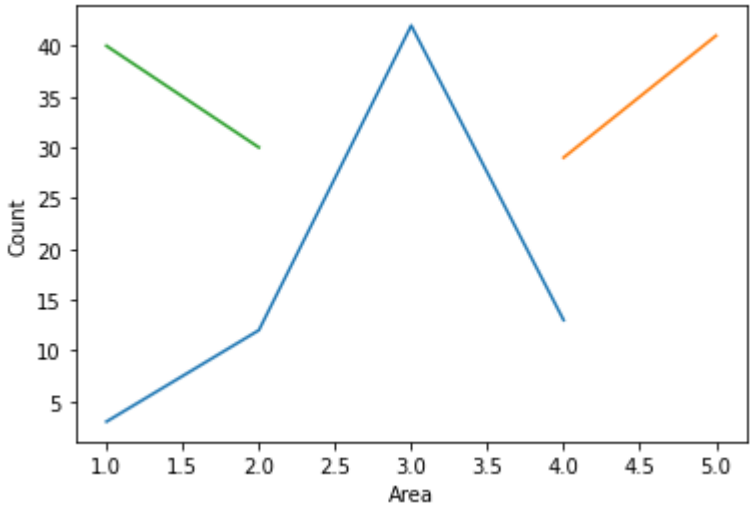
Question 2:

- **(2.1)** After importing the necessary libraries, we imported the data("dataset.txt") and converted it into a pandas dataframe with the respective columns. Histograms were plotted for each feature using the matplotlib library.
- (2.2) Each class had 70 samples. The prior of each class was computed and it was found to be equal.
- (2.3) Each feature was divided into 5 bins. This was done by sorting the data of the entire column and dividing it into 5 parts. Then I assigned the value 1 to the first bin, 2 to the second bin and so on.
- (2.4) A labels function was made which computed the number of elements for each class in each feature. It returned the values of the feature when the target class was a particular value. For example, x_c1 is a list of the values of the feature when the target class was 1. The probability was then calculated by dividing each of them by the length of x_c1. This was then plotted. For example - (y=1 is red, y=2 is green and y=3 is blue). Any point of the graph represents the probability of (x=x-value given y=class represented by that colour).

```
P(x=3)/(y=1) : 0.6
P(x=4)/(y=1) : 0.18571428571428572
P(x=2)/(y=1) : 0.17142857142857143
P(x=1)/(y=1) : 0.04285714285714286
P(x=4)/(y=2) : 0.4142857142857143
P(x=5)/(y=2) : 0.5857142857142857
P(x=2)/(y=3) : 0.42857142857142855
P(x=1)/(y=3) : 0.5714285714285714
```



- (2.5) The unique counts of each class was plotted



This plot is very similar to the distribution plot as it is just scaled. This is because the length of each target class is the same, i.e. each has 70 elements. If the target classes had a different number of elements, then the plot would have been different.

- (2.6) The evidence array was calculated. To calculate the posterior, the likelihood was multiplied by the prior and then divided by the evidence. We get the values as :

```
Evidence Array:  [0.2       0.2       0.2       0.2047619 0.1952381]
Posterior C1:  [0.07142857142857142, 0.2857142857142857, 0.9999999999999999, 0.3023255813953489, 0.0]
Posterior C2:  [0.0, 0.0, 0.0, 0.1380952380952381, 0.19523809523809524]
Posterior C3:  [0.9523809523809523, 0.6976744186046512, 0.0, 0.0, 0.0]
```

Graph for Area -