Pattern Recognition and Machine Learning
Bayes Classification
Lab Report

Name: Palak Singh

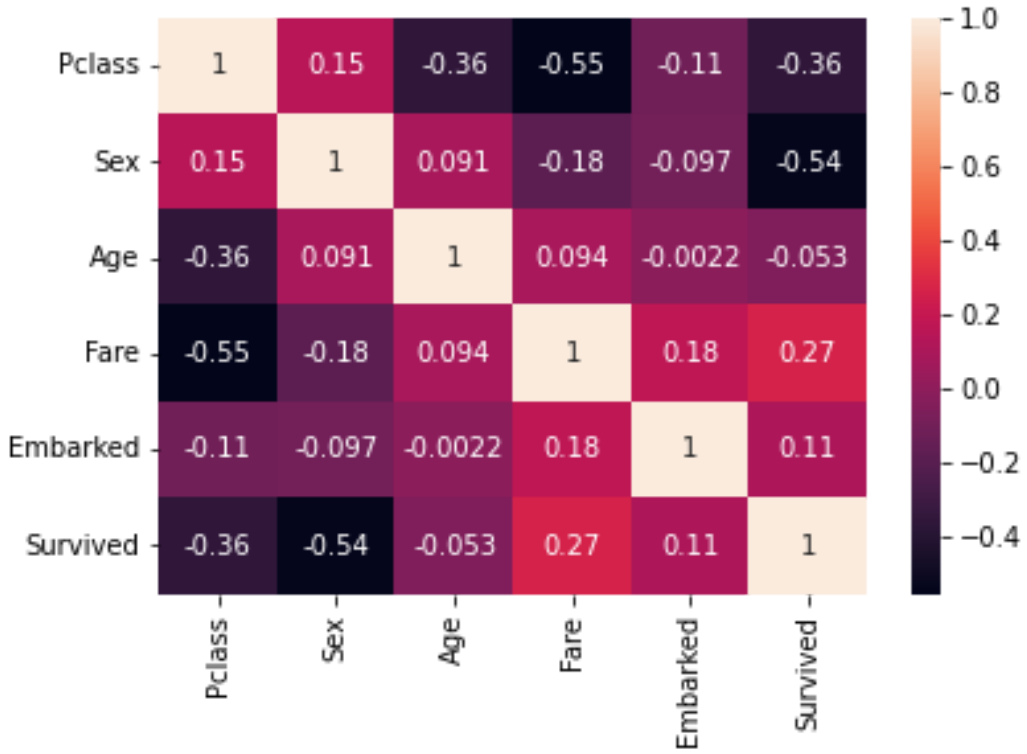
Question 1: Naive Bayes Classification- Titanic Dataset

Step 1: Pre Processing the Data

1. Some irrelevant features were removed from the dataset, the features that were removed were `'PassengerId','Name','Cabin','Ticket'`.
2. Label encoding of the data was done for the features which were not in the machine-interpretable form. These features included Sex and Embarked, another feature 'Age' was also label encoded as it has too many different values, these were divided in categories of 10 years and then label encoded.
3. The fields having NAN entries were dropped.
4. The data was then normalized for easier processing.

Step 2: Visualization of the data

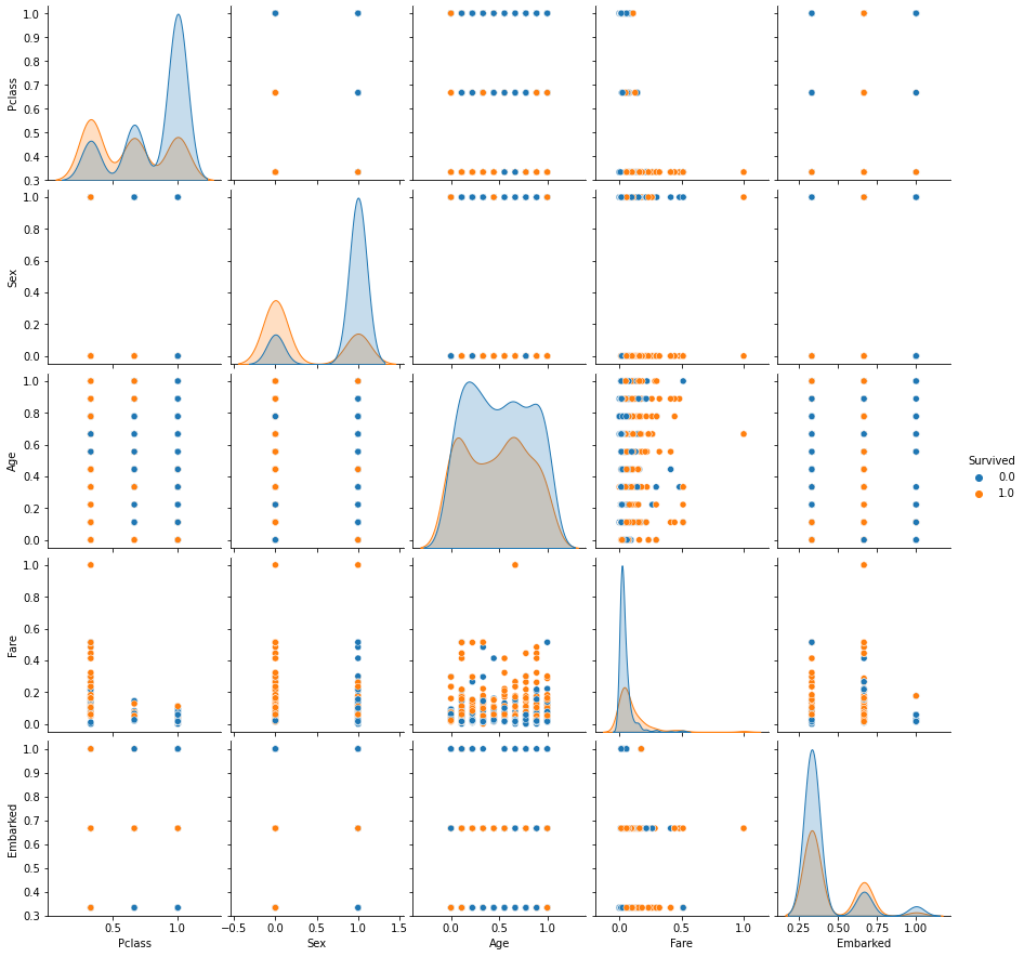
1. Heatmap of the data was generated, which indicated the relation of different features of the dataset.



Ans 2: <u>The type of classifier to be used</u>

From the heatmap we observe that the features are pretty much independent of each other, and the data is found to be continuous, so we find that implementing the Gaussian Naive Bayes Classifier will be the best possible variant for this dataset.

2. Visualization of the complete dataset



Step 3: Implementation of the gaussian naive's bayes

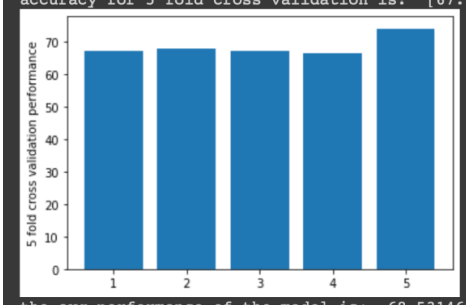
1. The 'Gaussian Model' for our Naive Bayes Classifier was implemented from scratch without using any inbuilt libraries.
2. For implementing gaussian model a class NBC was created and then different functions for calculating prior, likelihood and performance were made.
3. Using all the above functions the predictions were made for the test case.
4. On implementing the model over the entire training set, the accuracy of the model was found.

```
the accuracy of the model is:  55.94405594405595
```

Step 4: 5 fold cross validation

1. The model was used for 5 fold cross validation of the data and the performance of the model was checked.
2. The results of k fold cross validation were plotted for visualization.
3. It was found that the k fold cross validation results were more accurate compared to the results which were found earlier.

```
accuracy for 5 fold cross validation is:  [67.13286713286713, 67.83216783216784, 67.13286713286713, 66.43356643356644, 74.12587412587412]
```




```
the avg performance of the model is:  68.53146853146855
```

Step 5: Implementation of the library function for Gaussian Naive bayes classifier

1. On implementation of the inbuilt function the performance of the model was examined and the results showed that the accuracy of the model was around 81% whereas the performance of the model made from scratch was around 56%, which is comparatively less accurate than the inbuilt model.

```
Gaussian Naive Bayes inbuilt function performance:  81.41592920353983
```

Step 6: Implementation of other model - Decision Tree Classifier

1. The DTC was implemented and the 5- fold cross validation was done on the model, the results came out to be 47% accurate, which is very less then the accuracy of the predictions made by the gaussian model made in the beginning of the assignment.

```
kfold test scores of Decision tree regressor are [0.73460145 0.36719949 0.4510697  0.34415584 0.495671  ]
The avg performance of Decision tree regressor is 0.47853949616737346
```

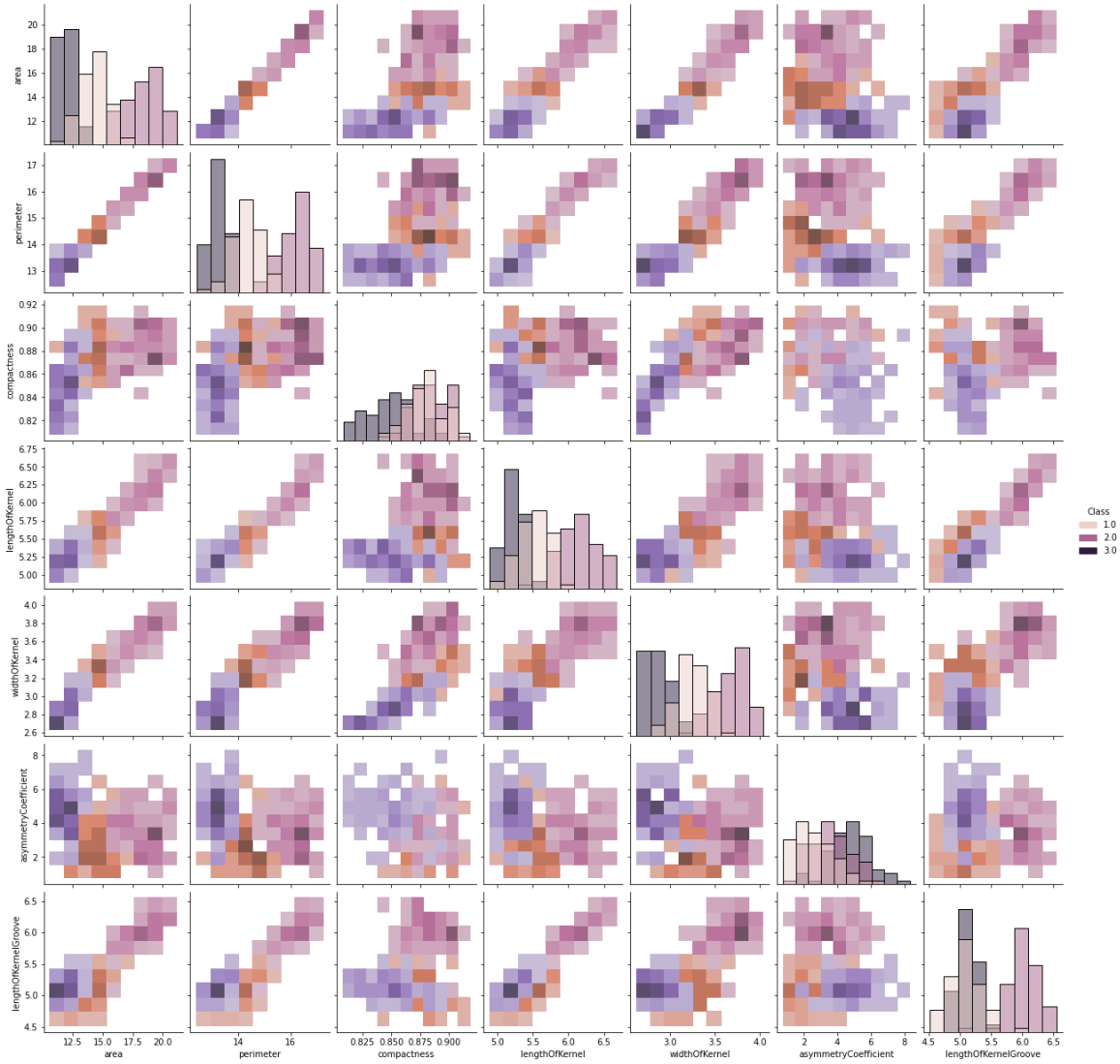
Question 2:

Step 1: Pre processing of the Data

1. The data was pre processed and normalized for implementing the different functions.

Step 2: Visualization of the data

1. Histogram Plot



Step 3: Calculation of prior probability of the three classes

1. The function similar to the one made for gaussian model was used here for calculating the prior probability of different classes.

```
prior probability of class  1  is:  0.3316582914572864
prior probability of class  2  is:  0.3417085427135678
prior probability of class  3  is:  0.32663316582914576
```

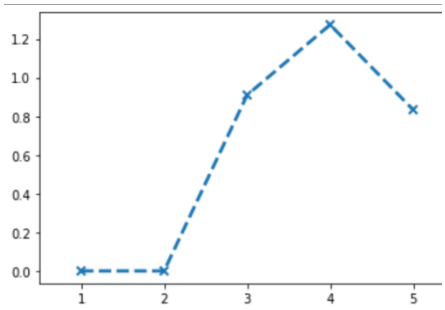
Step 4: Discretizing the data into bins

1. The data in the dataframe is categorized in 5 different bins by a function called bins.
2. We can also alter the number of bins to be made.

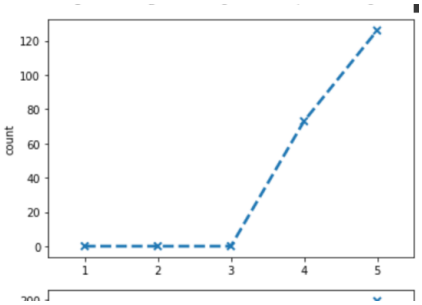
	area	perimeter	compactness	lengthOfKernel	widthOfKernel	asymmetryCoefficient	lengthOfKernelGroove	Class
0	4.0	5.0	5.0	5.0	5.0	2.0	4.0	1.0
1	4.0	5.0	5.0	5.0	5.0	1.0	4.0	1.0
2	4.0	5.0	5.0	4.0	5.0	2.0	4.0	1.0
3	4.0	5.0	5.0	4.0	5.0	2.0	4.0	1.0
4	4.0	5.0	5.0	5.0	5.0	1.0	4.0	1.0
...	...	...	...	...	...	...	...	...
205	3.0	4.0	5.0	4.0	4.0	3.0	4.0	3.0
206	3.0	4.0	5.0	4.0	4.0	3.0	4.0	3.0
207	4.0	4.0	5.0	4.0	5.0	5.0	4.0	3.0
208	3.0	4.0	5.0	4.0	4.0	3.0	4.0	3.0
209	3.0	4.0	5.0	4.0	4.0	4.0	4.0	3.0

Step 5: Calculating the likelihood of all the classes

1. A function called likelihood is used to calculate the likelihood of different classes.
2. The graph for the conditional probability is plotted.



Step 6: Plotting the count of every element of the Features



Step 7: Posterior probability

1. The posterior probability of every class was calculated using the function posterior made from scratch.
2. The posterior probability was pplotted on a graph.