CS60050
MACHINE LEARNING
Assignment 1
Question 2 Report

Pranav Mehrotra(20CS10085)
Saransh Sharma(20CS30065)

Code

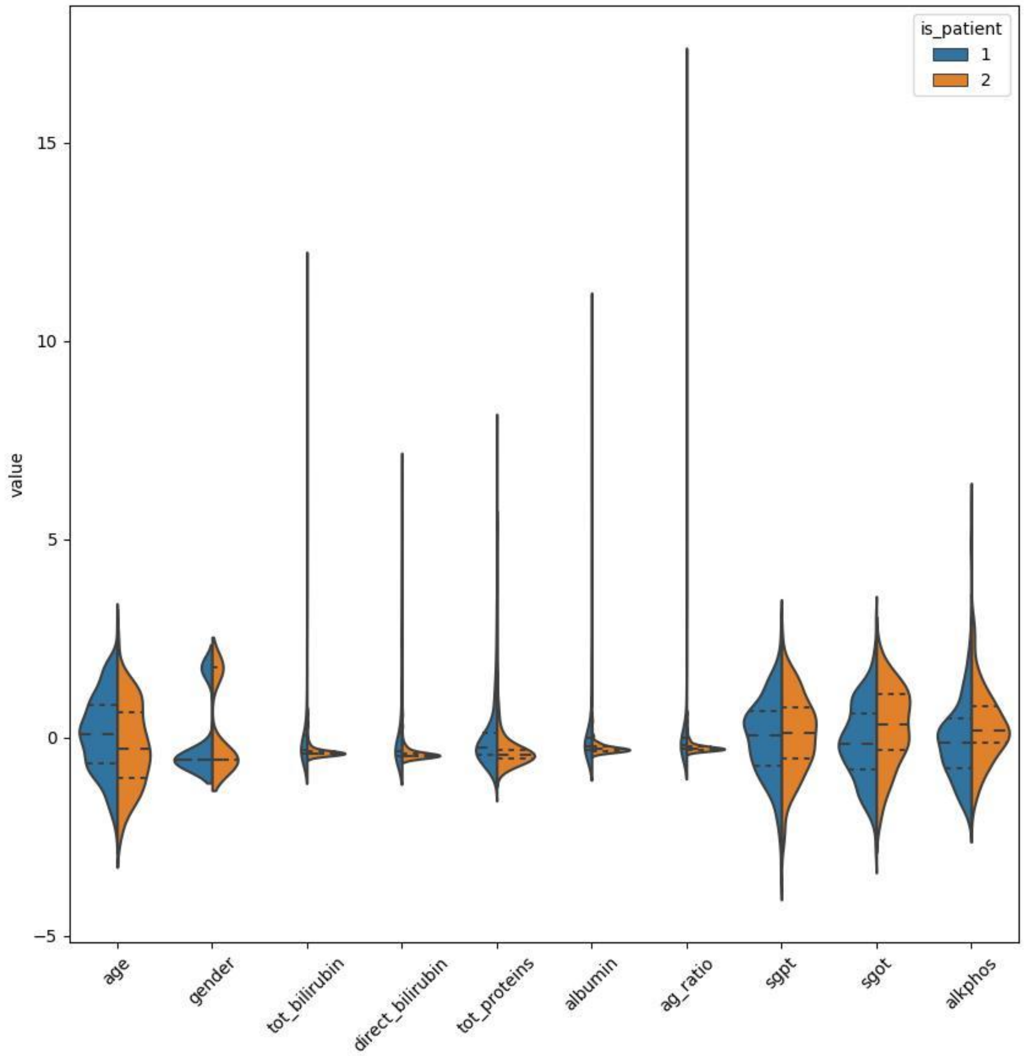
The files of the code:

1. “Naive Bayesian Classifier.py”

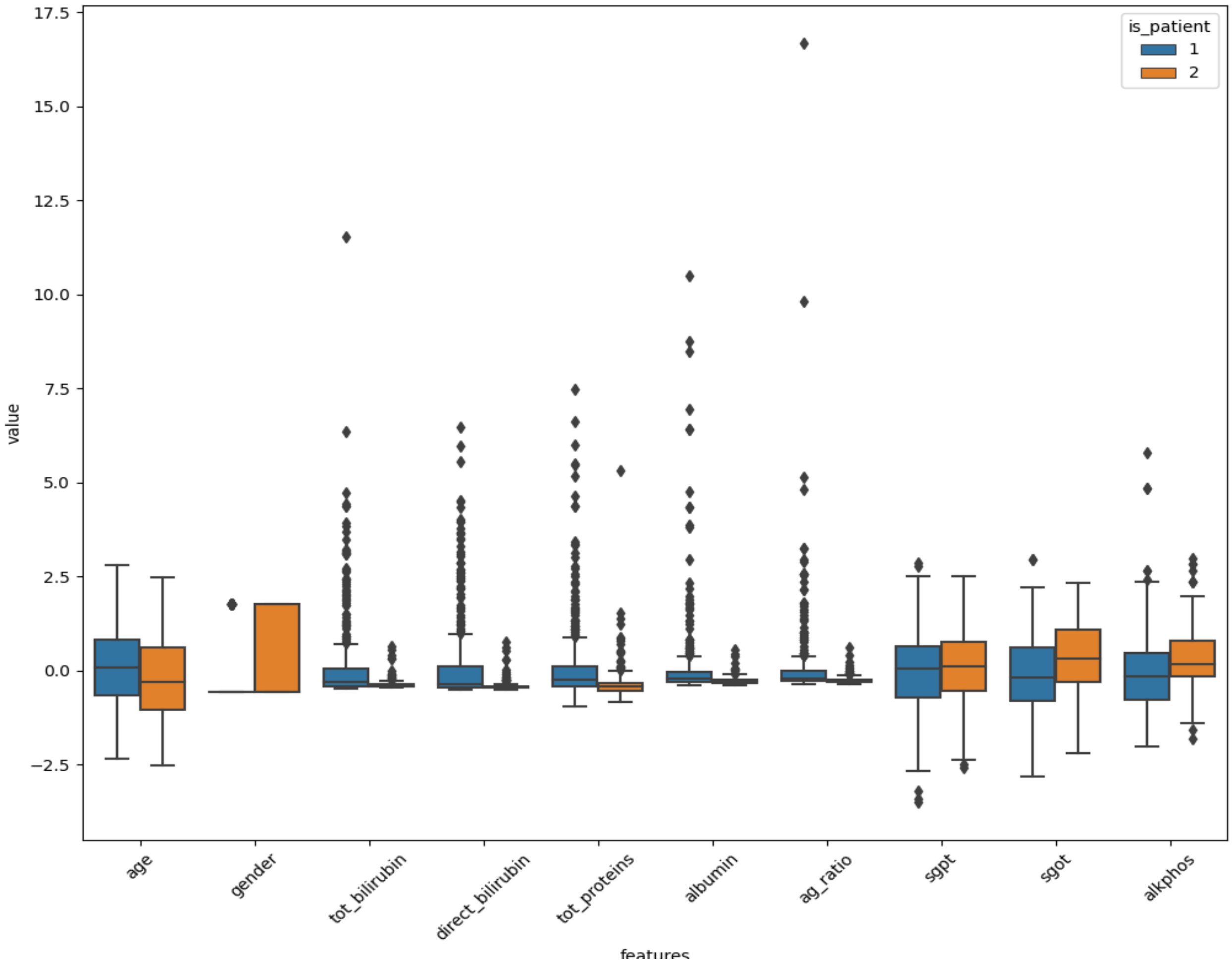
   This file contains the whole python code of the model and data evaluation. It mainly has mainly following functions:

   - Exploratory Data Analysis and Data visualization(EDA()):
     This step is performed to gain insights from data. Data visualizations help us to understand data better and make conclusions about the distribution of data.

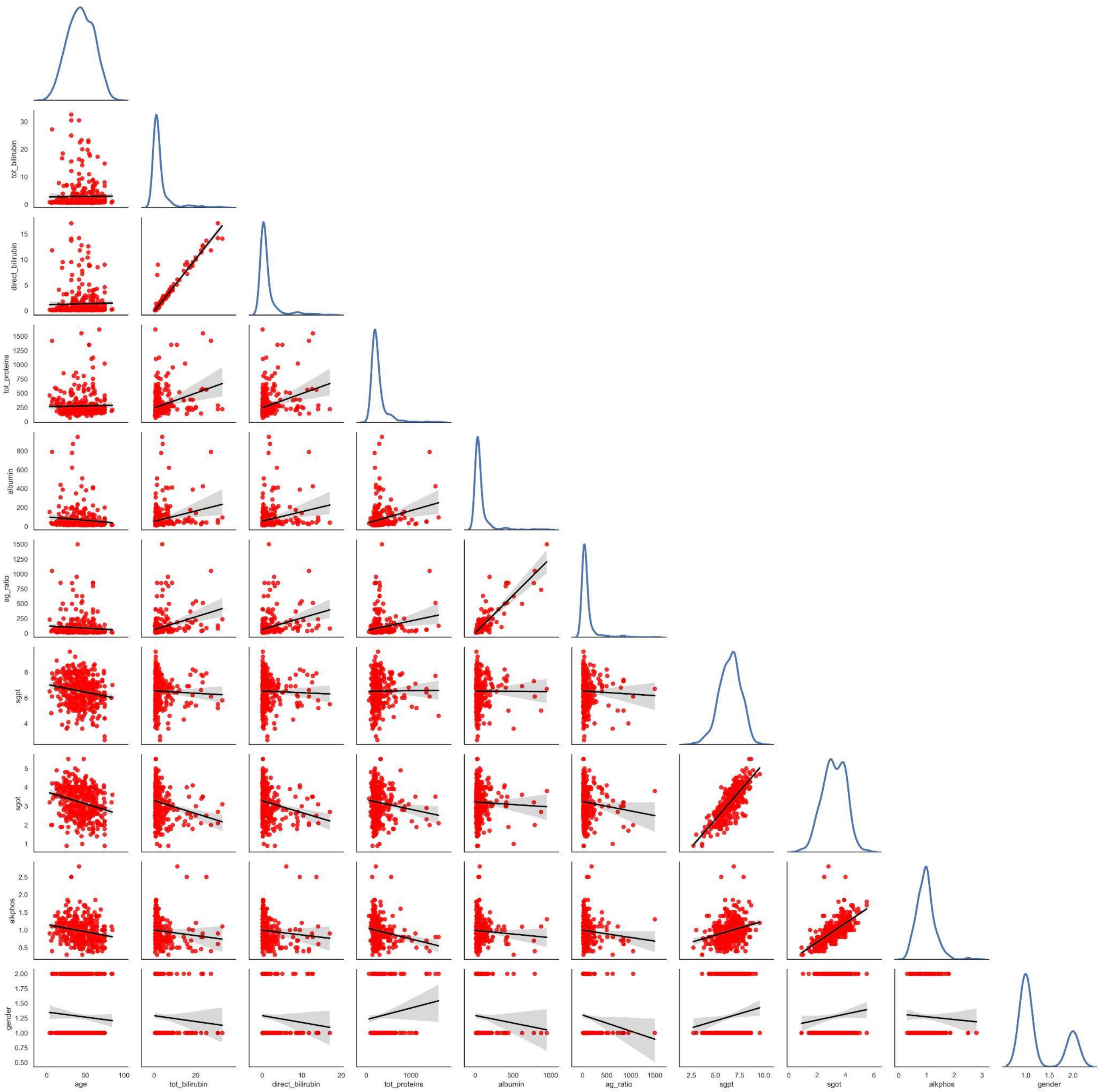
     1. **Violin Plot**: depicts the distribution of different columns of the data along with 25%, 50%(median) and 75% percentile of the data.



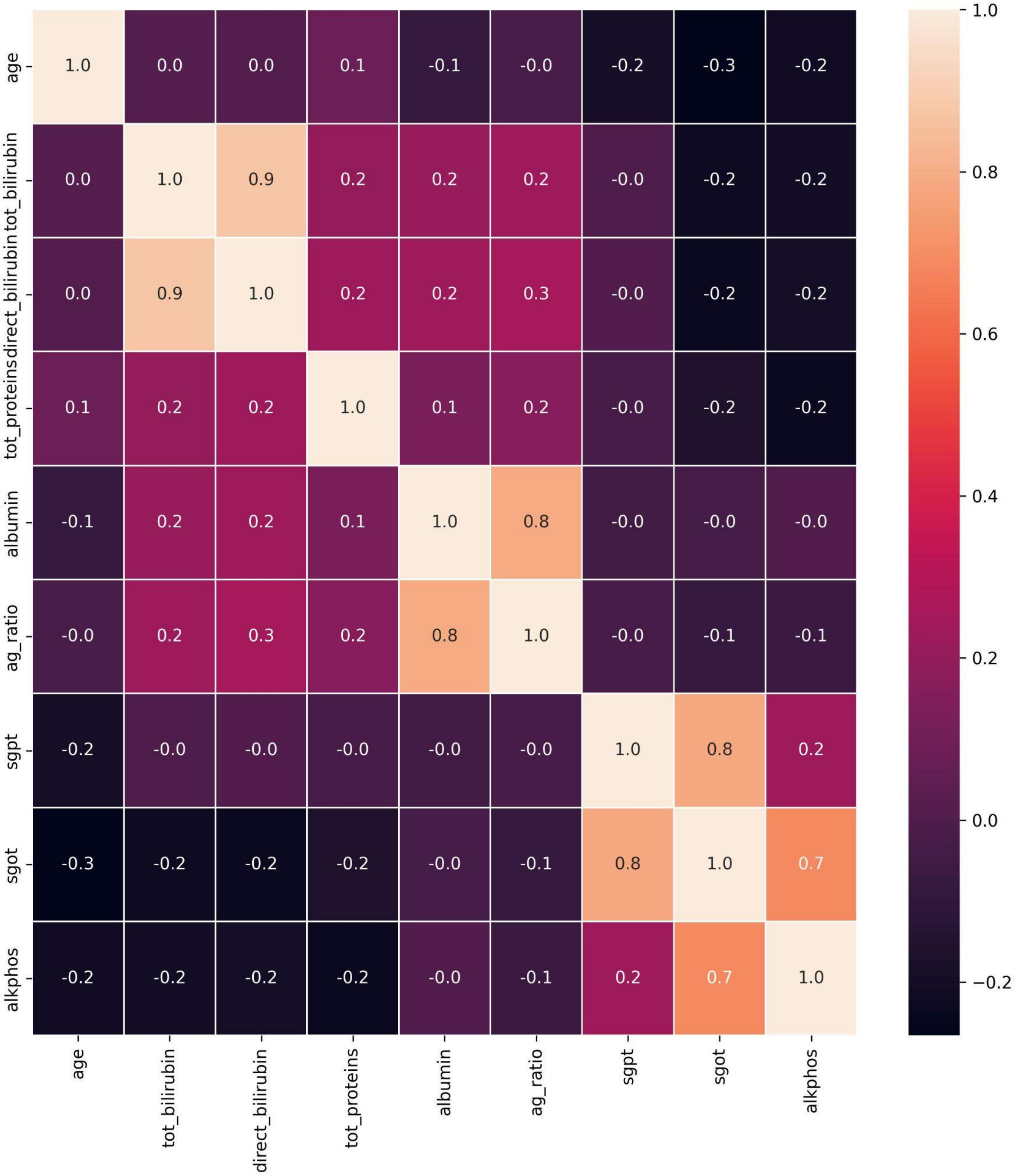
2. **Box plot**: depicts the spread of each column of data.



3. Kernel_Scatter_plot: depicts individual kernel density of features and scatter plots between every pair of features. Scatter plots help us to explore inter feature relationships.



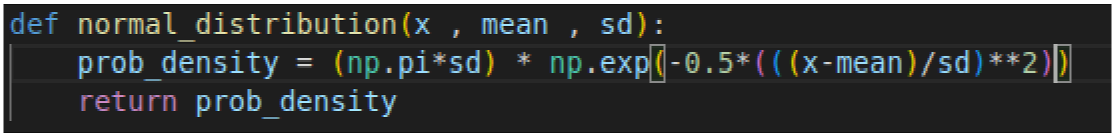
4. Heatmap: depicts the correlation i.e. extent of linear dependability between different features of the data.


age
tot_bilirubin
direct_bilirubin
tot_proteins
albumin
ag_ratio
sgpt
sgot
alkphos
1.0 0.0 0.0 0.1 -0.1 -0.0 -0.2 -0.3 -0.2
0.0 1.0 0.9 0.2 0.2 0.2 -0.0 -0.2 -0.2
0.0 0.9 1.0 0.2 0.2 0.3 -0.0 -0.2 -0.2
0.1 0.2 0.2 1.0 0.1 0.2 -0.0 -0.2 -0.2
-0.1 0.2 0.2 0.1 1.0 0.8 -0.0 -0.0 -0.0
-0.0 0.2 0.3 0.2 0.8 1.0 -0.0 -0.1 -0.1
-0.2 -0.0 -0.0 -0.0 -0.0 -0.0 1.0 0.8 0.2
-0.3 -0.2 -0.2 -0.2 -0.0 -0.1 0.8 1.0 0.7
-0.2 -0.2 -0.2 -0.2 -0.0 -0.1 0.2 0.7 1.0
age
tot_bilirubin
direct_bilirubin
tot_proteins
albumin
ag_ratio
sgpt
sgot
alkphos
1.0
0.8
0.6
0.4
0.2
0.0
−0.2

- category_encoding():
  This function encodes the categorical features to numerical values for easy processing of the data.
- remove_outliers():
  This function removes the outliers found in the data, by the formula given in the assignment.

  $$(2 \times \mu + 5 \times \sigma)$$

  The given data does not contain many outliers(10-15 samples out of 400+), and that too with more than one outlier features in the same sample is very rare(only 3-4 such samples). So, just removing 3-4 samples won't help much in terms of accuracy, so I have also removed the samples with only one outlier feature, so as to help in increasing accuracy.
- normal_distribution():
  This function calculates the probability, given the values of x, Mean and standard deviation, assuming 'x' follows Normal distribution. Using the following formula of PDF. It ignores the constant 1/sqrt(2*pi), because it will be the same for all, so it will be of no use while comparing the relative probabilities of different categories in the Naive Bayesian Classifier.



```
def normal_distribution(x , mean , sd):
    prob_density = (np.pi*sd) * np.exp(-0.5*(((x-mean)/sd)**2))
    return prob_density
```

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

- calc_proba():
  This function calculates the probability of the given data point, assuming that all the features are independent. It uses the following formula.

Naive Bayes classifier:

$$v_{NB} = \underset{v_j \in V}{\operatorname{argmax}} P(v_j) \prod_i P(a_i|v_j)$$

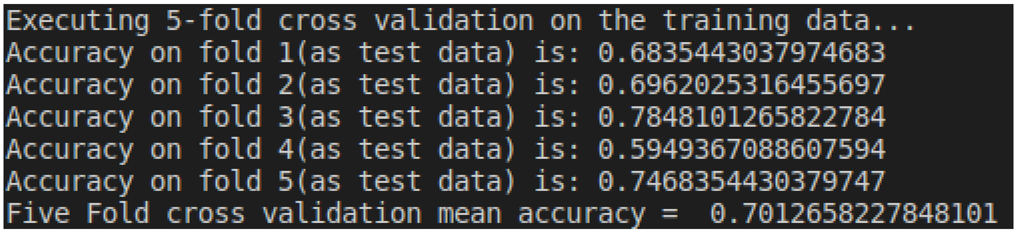
- initialise():
  It trains the model on the given training data and also does Laplace correction if ‘laplace_factor’ is non-zero. It calculates the probability of categorical features and the final category classifications. It also distributes the data into different categories and finds the means and standard deviations of respective features, so that the calc_proba() function can calculate the normal distribution probability of a given data point.
- classify():
  This function calls the calc_proba() function to calculate the relative probabilities of different categories and based on that it decides, the most likely category of the given sample.
- find_accuracy():
  This function takes as input the training and the testing data and laplace_factor. It calls the initialise() function to train the Naive Bayesian Classifier model on the training data and then calls the classify() function to classify the different samples of test data and calculates the accuracy of the model.
- five_fold_cross_validation():
  This function executes Five-fold cross-validation on the training set. It divides the training data into five equal parts and iterates on different parts, considering one part as a test set and the rest four parts as a training set. Then it computes the mean accuracy on all five folds and prints it.

Flow of Model Execution:

1. The data is shuffled randomly, and the random state is preserved. The shuffled data is split into training and test data (70-30).
2. Then the categorical variables are encoded into numbers for ease of processing. After this, the outliers are detected and removed from the data, by the formula given in the assignment.
3. Model executes the five-fold cross-validation on the training set and finds the mean accuracy of the five folds.
4. Then, the model is applied to the test(30%) data, and the final accuracy is found.
5. At last, the model executes Laplace correction, by passing the correction factor as 1, and then finds the accuracy on the test data.

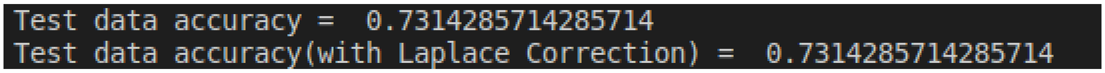
Summary of Results:

1. We can see that the model finds 8-9 samples with outlier features, and it successfully removes those outliers.
2. The average accuracy of five-fold cross-validation is **0.701265823.**



```
Executing 5-fold cross validation on the training data...
Accuracy on fold 1(as test data) is: 0.6835443037974683
Accuracy on fold 2(as test data) is: 0.6962025316455697
Accuracy on fold 3(as test data) is: 0.7848101265822784
Accuracy on fold 4(as test data) is: 0.5949367088607594
Accuracy on fold 5(as test data) is: 0.7468354430379747
Five Fold cross validation mean accuracy =  0.7012658227848101
```

3. The test(30%) data accuracy is **0.73142857.**



```
Test data accuracy =  0.7314285714285714
Test data accuracy(with Laplace Correction) =  0.7314285714285714
```

4. After Laplace correction, the accuracy is **0.73142857**. We can easily see that the accuracy does not change; this is because there are no zero probability features in the dataset. If there would have been categorical features with zero probability, then the Laplace correction would have changed the accuracy.