Question 1

Imported the dataset to the local and investigated it and found 4 features and 1 label columns. Feature columns are sepal-length, sepal-width, petal-length, and petal-width. The label column was variety.

After importing the dataset, the 1st thing, I did was there any null values or not and found it contains no null value, so no need to drop any column or row from the dataset.

Label columns were containing string values so I used label encoding method for better readability.

Also divided the dataframe into features and labels.

To perform the Naïve-bayes classification operation I defined a class called **NaiveBayesClassifier** which contains the below methods

fit:

This function takes argument class object, features and label as vector and calculate the mean variance for each feature for each class and store as vectors in class variables

It also calculates the prior probability for each class

gaussian_distribution:

This function takes argument class object, sample feature, mean and variance vector and returns the likelihood of each feature as a vector

Likelihood function: $f(x)=1/(sqrt(2*pi*var)*e^{(-.5*((x\text{-}mean)^\wedge 2/var))}$

X=sample

calculate_postirior:

This function takes argument class object and sample and returns the posterior probability of each class as vector

Posterior probability= logarithmic sum of prior probability and likelihood of each feature

predict classes:

This function takes argument as class object and samples and returns the predicted label for all samples as a vector

predict class:

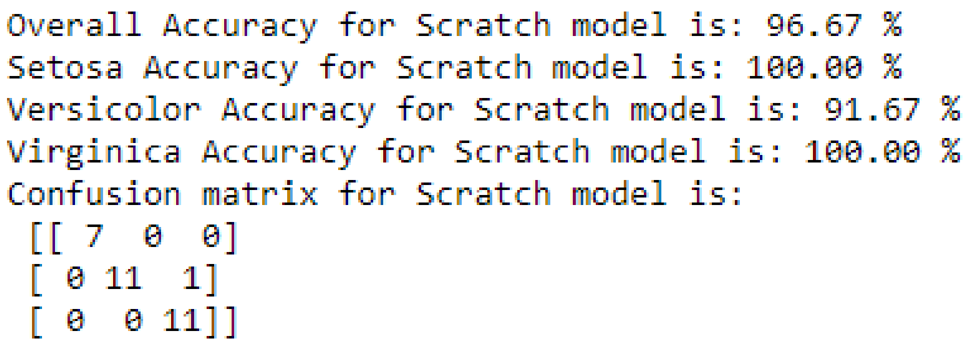
This function takes argument as class object and sample and returns as predicted label from the posterior vector which class having maximum posterior probability.

accuracy:

This function takes argument as class object and true labels and predicted labels as vectors and returns accuracy of prediction

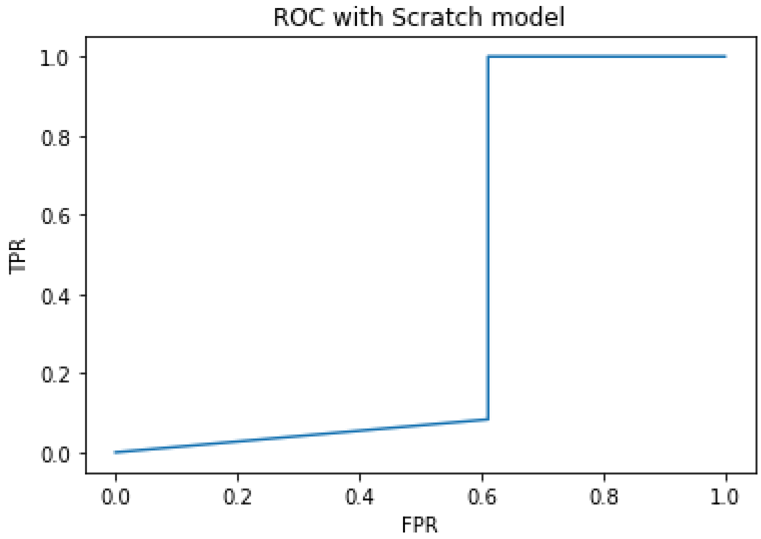
I split the features and labels dataset into training and testing dataset using sklearn train_test_split function.

Then I fit the training data to my **NaiveBayesClassifier** class object and validated with test data set and found the below results

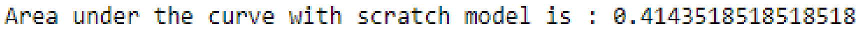


```
Overall Accuracy for Scratch model is: 96.67 %
Setosa Accuracy for Scratch model is: 100.00 %
Versicolor Accuracy for Scratch model is: 91.67 %
Virginica Accuracy for Scratch model is: 100.00 %
Confusion matrix for Scratch model is:
 [[ 7  0  0]
 [ 0 11  1]
 [ 0  0 11]]
```

Used roc_curve function of skelarn to get the tpr and fpr points and got the below ROC curve on plotting the same using pyplot of Matplotlib

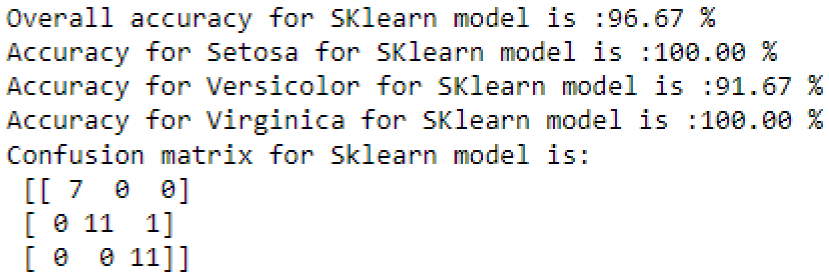


Used auc function from sklearn to get area under the curve and got the below result

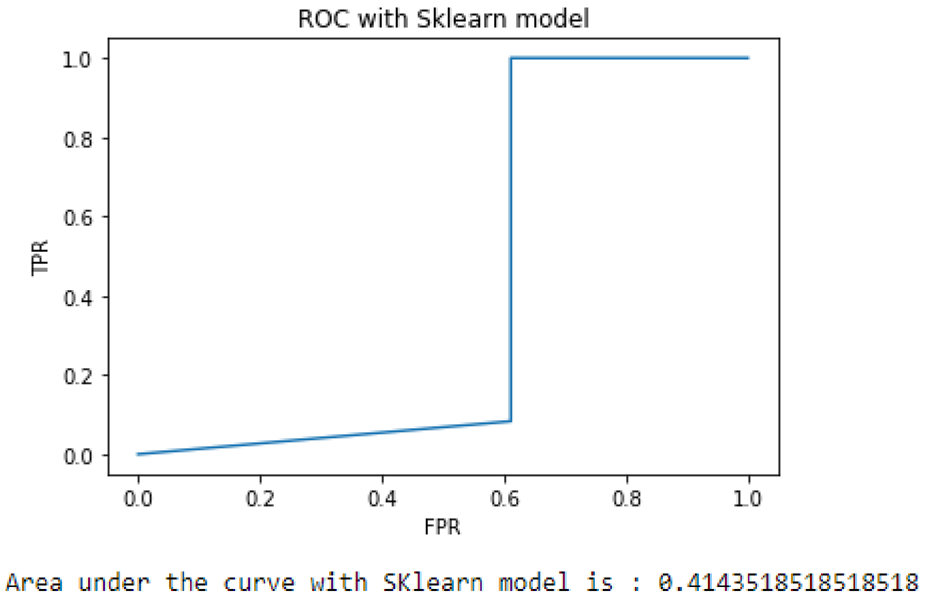


```
Area under the curve with scratch model is : 0.4143518518518518
```

Did the same operation from fitting same training dataset and validation of testing dataset on sklearn GaussianNB model and found the below result



```
Overall accuracy for SKlearn model is :96.67 %
Accuracy for Setosa for SKlearn model is :100.00 %
Accuracy for Versicolor for SKlearn model is :91.67 %
Accuracy for Virginica for SKlearn model is :100.00 %
Confusion matrix for Sklearn model is:
 [[ 7  0  0]
 [ 0 11  1]
 [ 0  0 11]]
```



```
Area under the curve with SKlearn model is : 0.4143518518518518
```

On risk factor calculation I did matrix element wise multiplication for my model confusion matrix and loss function matrix and sum of all matrix elements gave me the Bayes Risk and got the below result

```
Bayes risk : 51
```

Question 2

Note: Used only one dataset of 3 due to size constraint

Imported the dataset in the local and on inspection I found that the column “unnamed: 0” contribute to the dataset nothing so I dropped the column

On the pre-processing part removed all special characters and alpha-numeric characters

Also found Null values in the dataset so dropped those rows from the dataset

on inspection of the pre-processed dataset, I found that the 2nd column contains the labels.

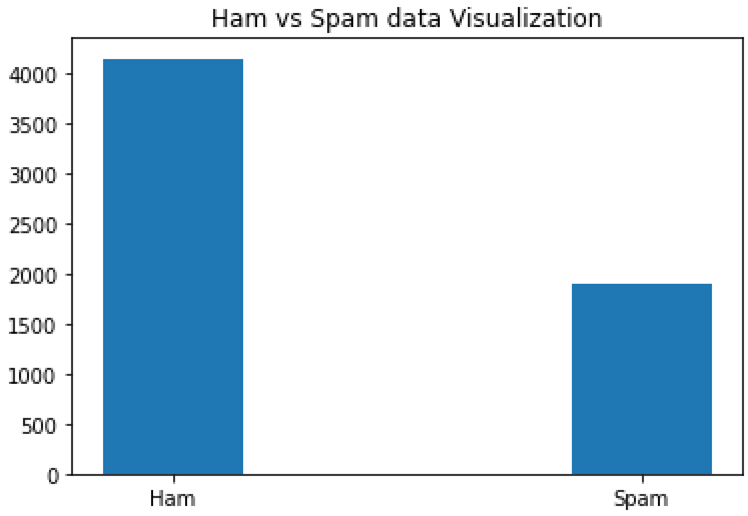
Labes contains only 2 values 0 and 1

1: Ham mails

2: Spam mails

To visualize the Ham and spam mails I get the sum of number of ‘1’ label for Spam and ‘0’ for Ham

To Visualize the dataset, I used Matplotlib bar plot and found the below result



To plot the wordcloud need to process further, for that I created a new class **Word_handling** which contains the below functions

split_sentence_in_words:

This function takes argument as class object and a string a returns lower case string

split_words:

This function takes argument as class object and a string

It splits the strings into words separated by space and store it as a list and finally returns it

split_sentences_in_words:

It takes argument as class object and list of strings

Split each string into list of string using split_words function and returns

get_unique_words:

It takes argument as class object and list of string and return list of unique words presents in the list of strings

I got the unique words present in Ham and Spam dataset using the above class object

Now I can create wordcloud using wordcloud library

And plotted the same using imshow function of Matplotlib and got the below figures

Ham wordcloud

aaliyah
ab
abba
abc
abductee
abound
abuse
absolute
abstraction
accident
abduction
academic
acceleration
accommodation
accountant
activist
act
add
addict
achieve

Spam wordcloud

abacha
ability
ab
achieve
abuse
accept
acknowledge
accumulate
acid
account
accountant
abonnement
accommodation
activation
affect
affliction
adapte
acquire
acquisition
addressyour
administrator
advertisement

Question 4

I imported the dataset from the below link

link="http://archive.ics.uci.edu/ml/machine-learning-databases/breast-cancer/breast-cancer.data"

After importing the dataset on inspection of the dataset I found that the dataset contains missing values, so removed those rows which contains missing values as a part of pre-processing.

Also found that the features contain string values, ranges which are not suitable for Naïve bayes operation so I did label encoding as part of pre-processing. Now our dataset is ready to split in training and testing dataset.

Also, from the dataset got the info that the “class_” is the target labels

So next I did to split the dataset into features and labels

I took “class_” column as label and rest other columns as features.

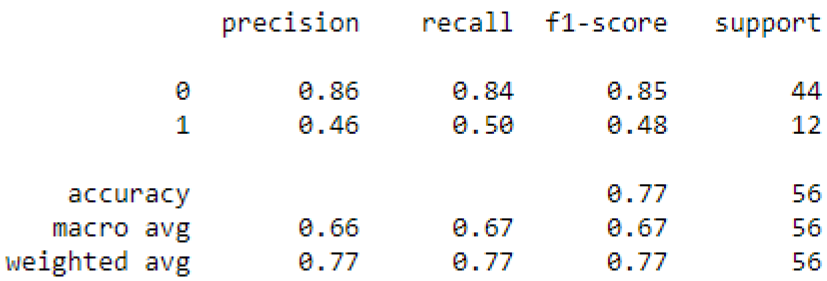
Now I used train_test_split function from sklearn to split the dataset into training and testing dataset.

I used **GaussianNB** model from Sklearn and fit the training dataset.

Later validated with testing dataset.

Also, used **roc_curve** function from sklearn to get tpr and fpr points and area under the curve in the ROC respectively

Used classfication_report from sklearn to generate classification report and got the below report

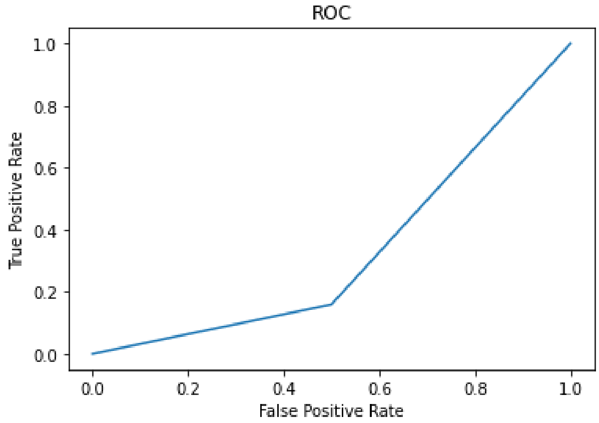


```
              precision    recall  f1-score   support

           0       0.86      0.84      0.85        44
           1       0.46      0.50      0.48        12

    accuracy                           0.77        56
   macro avg       0.66      0.67      0.67        56
weighted avg       0.77      0.77      0.77        56
```

Plotted tpr and fpr points using pyplot from matplotlib library and found the below figure


ROC
True Positive Rate
False Positive Rate