2-CLASS CLASSIFICATION USING BAYES CLASSIFIER

Linear Algebra and Random Processes (CS6015)
Assignment 3 Input/Output Specifications

Your program (bayes.c/bayes.cpp) written in C/C++ should read the files given under input section and generate the files specified under output section. All the files should be present at the same directory level.

1 Input

1. input_data1
2. input_data2

where input_data1 and input_data2 can be either of the 6 possible cases (i.e., one of the 3 distributions of any of the two combinations of P and N) and it contains two .txt files one for each class.
Divide the data into training and validation data (80%- training and 20%-validation ie **first** 80% samples from class1 followed by **first** 80% samples from class2 makes the training set and the remaining 20% from class1 followed by the remaining 20% from class2 makes the validation set).

1.1 Sample

input_data1 = 1
It will have 2 .txt files : class1_train.txt,class2_train.txt (each containing N samples).

2 Output

1. gamma_no1.txt - Output corresponding to input_data1
2. gamma_no2.txt - Output corresponding to input_data2

where no1, no2 correspond to the numbers allotted to each one of you as mentioned in the table given.

2.1 Sample

gamma_1.txt (corresponding to input_data = 1)
gamma_6.txt (corresponding to input_data = 6)

3 File Format

The first line of each of the two output files should contain a row vector giving the predicted labels for the training data.
The second line should contain the training accuracy.
The third line should contain a row vector giving the predicted labels for the validation data.
The fourth line should contain validation accuracy.

3.1 Sample

1 1 2 2 1 (with a sapce between each label)
95.43
2 1 2 1 1
95.84

The above mentioned output files must exist in the zip file along with the code while submitting. The training data is read from within the code and is not given as argument to the main function.

4 Accuracy

Accuracy is the number of correctly classified labels divided by total number of labels.

5 Evaluation

- The test data is a .txt file (which is not given to you). The format will be the same as that of the input_data given.
- The code should be written in such a way that it should take this .txt file name as input "argument" to the main function and read as a file inside the main functin. It should then create an output .txt file with the name **output.txt** (in the same directory) which should contain the predicted labels as a row vector.
  Example: 1 1 2 1 1 2 (with a space between each label)
- To check if your code is working, make a txt file of your own from the validation data and give it as input argument to the code and check the accuracy as you have also been provided with the corresponding class labels for the validation data.As mentioned

before, the test data will be of the same format as that of the validation data.

The file is executed as:
gcc bayes.c
./a.out test.txt

6 Files expected in the zip file

- Make a separate folder "rollno_A3".
- This is the folder that is expected to be zipped and sent.
- Inside the folder there needs to be one "bayes.c/bayes.cpp" file.
- Two gamma_no.txt (where $no = 1, .., 6$) files.
- **Nothing more should be there inside the folder while submission.**

7 While implementing and testing the code

- Inside the folder, class1_train.txt and class2_train.txt files will be there which will be given as input. (You should not add this to your zip folder as we will add it ourselves during evaluation.)
- From class1_train.txt and class2_train.txt create test.txt on your own which is the validation data. This is just for you to check if the code is creating an "output.txt" file in the same folder as expected. (This should also not be added to the zip folder as we will be testing it with our test.txt file which is not given to you.)