You can use sklearn and Pandas or any other Python package for this lab.

1. **Gene expression cancer RNA-Seq dataset**[1]

   This collection of data is part of the RNA-Seq (HiSeq) PANCAN data set, it is a random extraction of 20531 gene expressions of 801 patients having different types of tumor: BRCA (Breast or Ovarian), KIRC (Kidney renal clear cell carcinoma), COAD (Colon adenocarcinoma), LUAD (Lung adenocarcinoma) and PRAD (Prostate adenocarcinoma). This is a very high dimensional dataset that calls for a method such as Naïve Bayes', which can handle high-dimensional data very well. Remember that because of the curse of dimensionality, it is impossible to find an accurate joint distribution of 20531 features in five classes with only a few hundred observations, while fining the marginal distributions is straightforward.

   The goal of this lab is to classify tumors based on their gene expressions.

   (a) Load the files `data.csv` and `labels.csv` from the Dropbox folder. They contain the Data Set from: `https://archive.ics.uci.edu/dataset/401/gene+expression+cancer+rna+seq`. `data.csv` contains the genetic features for each tumor and `labels.csv` contains the label of each tumor. (5 pts)

   (b) Exploratory data analysis:

      i. Select the first 640 instances as the training set and the rest of the data as the test set. (5 pts)

      ii. Encode the classes as follows BRCA = 0, KIRC = 1, COAD = 2, LUAD = 3, and PRAD = 4. You can use Ordinal Encoder.[2] (5 pts)

   (c) Classification using Gaussian Naïve Bayes

      Because all the features are continuous, one can fit normal/Gaussian marginal pdfs to each of them in each class.

      i. Use sklearn's Gaussian Naïve Bayes method[3] to build a classifier based on training data. Report the training misclassification error rate (the percentage of training data that are misclassified). (20 pts)

      ii. Use sklearn's Gaussian Naïve Bayes method to classify test data, using the model you developed in 1(c)i.[4]. Report test misclassification error rate (the percentage of training data that are misclassified). (20 pts)

   (d) Classification using Bernoulli Naïve Bayes

[1] This Lab is assigned in October in Fall semesters, which is Prostatic Cancer Awareness Month. Please help in raising awareness about prostatic cancer. `https://www.pcf.org/prostate-cancer-awareness-month/`

[2] `https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.OrdinalEncoder.html#sklearn.preprocessing.OrdinalEncoder`

[3] See `https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.GaussianNB.html#sklearn.naive_bayes.GaussianNB`. Most of the default parameters work here, but double check if you need to set any parameters.

[4] Attention: DO NOT use test data to train a new model. You must predict the label of test data using training data only, and compare the labels predicted by training data with the actual labels of the test data to calculate the test misclassification error. Otherwise, there is no point in using training data!

i. Calculate the median of each of the gene features *in the training set.* Binarize the features in the training set: any feature greater than or equal to the median of that feature must be converted to 1 and any feature less than the median must be converted to zero. Binarize the features in the test set using the medians you found for features *in the training set.* Any feature value in the test set that is greater than or equal to the median of the corresponding feature in the training set must be converted to 1 and any feature value less than the median of the corresponding feature in the test set must be converted to 0.[5] (5 pts)

ii. Use sklearn's Bernoulli Naïve Bayes method with Laplace Smoothing[6] to build a classifier based on binarized training data. Report the training misclassification error rate (the percentage of training data that are misclassified). (20 pts)

iii. Use sklearn's Bernoulli Naïve Bayes method to classify test data, using the model you developed in 1(d)ii.[7] Report test misclassification error rate (the percentage of training data that are misclassified). (20 pts)

2. **Extra Credit (Categorical Naïve Bayes)** (20 pts)

(a) Create 10 equally spaced bins between the maximum and minimum of each feature in the training set and convert the features to categorical training data using those bins.[8] Convert the test data based on the bins you calculated for *training daya* into categorical features.(4 pts)

(b) Use sklearn's Categorical Naïve Bayes method with Laplace Smoothing[9] to build a classifier based on categorized training data.[10] Report the training misclassification error rate (the percentage of training data that are misclassified). (8 pts)

(c) Use sklearn's Categorical Naïve Bayes method to classify test data, using the model you developed in 2b.[11] Report test misclassification error rate (the percentage of training data that are misclassified). (8 pts)

[5] You do not include test data in calculations of medians because no information must leak from the test set to the model trained on the training set.

[6] See `https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.BernoulliNB.html#sklearn.naive_bayes.BernoulliNB`. Most of the default parameters work here, but double check if you need to set any parameters.

[7] Attention: DO NOT use test data to train a new model. You must predict the label of test data using training data only, and compare the labels predicted by training data with the actual labels of the test data to calculate the test misclassification error. Otherwise, there is no point in using training data!

[8] You can use Pandas cut `https://absentdata.com/pandas/pandas-cut-continuous-to-categorical/#` and ordinal encoder `https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.OrdinalEncoder.html#sklearn.preprocessing.OrdinalEncoder`.

[9] See `https://scikit-learn.org/stable/modules/naive_bayes.html#categorical-naive-bayes`. Most of the default parameters work here, but double check if you need to set any parameters.

[10] This approach is equivalent to building histograms for each of the features

[11] Attention: DO NOT use test data to train a new model. You must predict the label of test data using training data only, and compare the labels predicted by training data with the actual labels of the test data to calculate the test misclassification error. Otherwise, there is no point in using training data!

Note: You must submit Extra Credit along with your main homework. If you submit Extra Credit Late, your whole homework will be late.