**ENGR 421/DASC 521:** Introduction to Machine Learning
**Homework 1:** Naive Bayes Classifier
**Deadline:** October 31, 2022, 11:59 PM

In this homework, you will implement a naive Bayes classifier using Python. Here are the steps you need to follow:

1. Read Section 5.7 from the textbook.

2. You are given a multivariate classification data set, which contains 400 nucleotide sequences of length 7. These sequences are from two distinct classes, namely, 1 (a splice site) and 2 (not a splice site), where we have 200 sequences from each class. You are given two data files:

   a. `hw01_data_points.csv`: nucleotide sequences,

   b. `hw01_class_labels.csv`: corresponding class labels.

3. Divide the data set into two parts by assigning the first 300 sequences to the training set and the remaining 100 sequences to the test set. (10 points)

---

**Hint:** Let us define the following probabilities.

$$p_{\mathrm{A}cd} = \text{probability of having adenine (A) for class } c \text{ at location } d,$$
$$p_{\mathrm{C}cd} = \text{probability of having cytosine (C) for class } c \text{ at location } d,$$
$$p_{\mathrm{G}cd} = \text{probability of having guanine (G) for class } c \text{ at location } d,$$
$$p_{\mathrm{T}cd} = \text{probability of having thymine (T) for class } c \text{ at location } d.$$

---

4. Estimate the model parameters $\hat{p}_{\mathrm{A}11}, \ldots, \hat{p}_{\mathrm{A}17}$, $\hat{p}_{\mathrm{A}21}, \ldots, \hat{p}_{\mathrm{A}27}$, $\hat{p}_{\mathrm{C}11}, \ldots, \hat{p}_{\mathrm{C}17}$, $\hat{p}_{\mathrm{C}21}, \ldots, \hat{p}_{\mathrm{C}27}$, $\hat{p}_{\mathrm{G}11}, \ldots, \hat{p}_{\mathrm{G}17}$, $\hat{p}_{\mathrm{G}21}, \ldots, \hat{p}_{\mathrm{G}27}$, $\hat{p}_{\mathrm{T}11}, \ldots, \hat{p}_{\mathrm{T}17}$, $\hat{p}_{\mathrm{T}21}, \ldots, \hat{p}_{\mathrm{T}27}$, $\widehat{\Pr}(y = 1)$, and $\widehat{\Pr}(y = 2)$ using the data points you assigned to the training set in the previous step. Your parameter estimations should be like the following figures. (30 points)

```
print(pAcd)
[[0.28       0.68 0.09333333 0.56666667 0.68       0.14 0.19333333]
 [0.22666667 0.24 0.26666667 0.18666667 0.20666667 0.18 0.26      ]]
print(pCcd)
[[0.4  0.08666667 0.01333333 0.02       0.12       0.06       0.08      ]
 [0.16 0.23333333 0.09333333 0.17333333 0.21333333 0.26666667 0.19333333]]
print(pGcd)
[[0.21333333 0.09333333 0.82666667 0.35333333 0.1 0.76        0.20666667]
 [0.29333333 0.27333333 0.22666667 0.36       0.2 0.23333333  0.2       ]]
print(pTcd)
[[0.10666667 0.14       0.06666667 0.06 0.1  0.04 0.52      ]
 [0.32       0.25333333 0.41333333 0.28 0.38 0.32 0.34666667]]
print(class_priors)
[0.5 0.5]
```

**Hint:** You can use the following equations to estimate the parameters.

$$\hat{p}_{\mathrm{A}cd} = \frac{\sum_{i=1}^{N} 1(x_{id} = \mathrm{A}) 1(y_i = c)}{N_c}$$

$$\hat{p}_{\mathrm{C}cd} = \frac{\sum_{i=1}^{N} 1(x_{id} = \mathrm{C}) 1(y_i = c)}{N_c}$$

$$\hat{p}_{\mathrm{G}cd} = \frac{\sum_{i=1}^{N} 1(x_{id} = \mathrm{G}) 1(y_i = c)}{N_c}$$

$$\hat{p}_{\mathrm{T}cd} = \frac{\sum_{i=1}^{N} 1(x_{id} = \mathrm{T}) 1(y_i = c)}{N_c}$$

$$\widehat{\Pr}(y = c) = \frac{\sum_{i=1}^{N} 1(y_i = c)}{N} = \frac{N_c}{N}$$

5. Calculate the confusion matrix for the data points in your training set using the parametric classification rule you will develop using the estimated parameters. Your confusion matrix should be like the following matrix. (30 points)

```
print(confusion_train)
y_truth    1    2
y_pred
1        145   14
2          5  136
```

**Hint:** You can use the following equation to calculate the scores.

$$\begin{aligned} g_c(\boldsymbol{x}) &= \log\left[\prod_{d=1}^{D} p(x_d | y = c)\right] + \log \widehat{\Pr}(y = c) \\ &= \log\left[\prod_{d=1}^{D} \left(\hat{p}_{\mathrm{A}cd}^{1(x_d=\mathrm{A})} \hat{p}_{\mathrm{C}cd}^{1(x_d=\mathrm{C})} \hat{p}_{\mathrm{G}cd}^{1(x_d=\mathrm{G})} \hat{p}_{\mathrm{T}cd}^{1(x_d=\mathrm{T})}\right)\right] + \log \widehat{\Pr}(y = c) \end{aligned}$$

6. Calculate the confusion matrix for the data points in your test set using the parametric classification rule you will develop using the estimated parameters. Your confusion matrix should be like the following matrix. (30 points)

```
print(confusion_test)
y_truth   1   2
y_pred
1        48   8
2         2  42
```

---

**What to submit:** You need to submit your source code in a single file (`.py` file) named as `STUDENTID.py`, where `STUDENTID` should be replaced with your 7-digit student number.

**How to submit:** Submit the file you created to Blackboard. Please follow the exact style mentioned and do not send a file named as `STUDENTID.py`. Submissions that do not follow these guidelines will not be graded.

**Late submission policy:** Late submissions will not be graded.

**Cheating policy:** Very similar submissions will not be graded.

---