Assignment #2: Probabilities, Bayes' theorem, Markov chains (total 10 points), due by 11:59 pm Friday, 13 October 2023

Exercise 1: Bayes' on test accuracy. An emergency beacon for hiking (also known as a personal locator beacon, or PLB) is a small handheld device that will send out a distress signal in the event of an emergency. The Altitude Manufacturing Company makes 81% of the PLBs, the Bright Company makes 15% of them, and the Camping Company makes the other 4%. The PLBs made by Altitude have a 4% rate of defects, the Bright PLBs have a 6% rate of defects, and the Camping PLBs have a 9.5% rate of defects.

(a) If a PLB is randomly selected from the general population of all PLBs, find the probability that it was made by the Altitude Manufacturing Company. **(2 points)**

(b) If a randomly selected PLB is then tested and is found to be defective, find the probability that it was made by the Altitude Manufacturing Company. **(2 points)**

You can write your answers for this question using any text editor of your preference, but, please, set the document layout with the following features:

- Page orientation: portrait;
- Page size: 8.5" × 11" (letter);
- Margins: 2.54 cm (top, bottom, left, right);
- Line spacing: between 1.0 and 1.15 pt;
- No double-column format. Write your answers as a 1-column text;
- Paragraph alignment can be 'align left' or 'justify';
- Font type and size: Times New Roman, 12 pt, non-italic, non-bold;
- Use *italic*/**bold**/underline only to emphasize brief pieces of text if needed.
- ONLY `pdf` FILES ARE ACCEPTED. Do not submit editable files such as `.docx`.

Alternatively, you can write your answers on paper and scan your solutions. However, **make sure your answers can be read clearly and the handwriting is legible.** The scanned file to be submitted needs to be in `pdf` format.

*****

Exercise 2: The Ehrenfest Chain (dog–flea model). Another case of a Markov chain is a simple model of diffusion known as *the Ehrenfest Chain*.

(a) Research and describe what is the Ehrenfest chain process. This description should not exceed 1 page. Follow the same text format layout as given in exercise 1. **(2 points)**

(b) Consider two boxes in which a total of four particles ($N = 4$) can be distributed between them. At each step, one of the four particles is chosen at random and moved from the box that it is in into the other box. States can be then defined as the number of particles in the first box. The transition matrix for this example is

$$P = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 1/4 & 0 & 3/4 & 0 & 0 \\ 0 & 1/2 & 0 & 1/2 & 0 \\ 0 & 0 & 3/4 & 0 & 1/4 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

Find its stationary distribution. You can use numerical procedures to find the stationary distribution. Submit your code or script that gives the stationary distribution π in which $\pi = \pi\, P$. **(2 points)**

(c) Write a code that simulates an Ehrenfest chain process with two boxes (A and B) and a total of $N = 10$ particles that can be distributed within the boxes. Let X_n denote the number of particles on box A after n steps. If there are i particles on A, then on the next step, the number of particles on A either goes up by one, if one of the $N - i$ particles on B is transferred to A, or goes down by one, if one of the i particles on A is transferred to B. The process is a Markov chain with $X_n \equiv \{0, 1, ..., N\}$ with transition probabilities

$$P_{ij} = \begin{cases} i/N & \text{if } \; j = i-1 \quad \text{(box had } i \text{ particles, now has } i-1) \\ (N-i)/N & \text{if } \; j = i+1 \quad \text{(box had } i \text{ particles, now has } i+1) \\ 0, \quad \text{otherwise.} \end{cases}$$

Run your simulation up to 1000 time steps. Explain your code and its outputs using comment lines, markdowns, and/or docstrings – depending on your chosen integrated development environment (IDE) software – and show a plot of how the number of particles in one of the boxes varies in time. If you are working with an IDE that does not save figures on their platform, save your figures in `.jpg`. **(2 points)**

IMPORTANT: Whenever applicable, your code needs to output all required numbers of the exercise and they should be printed clearly. The printed values should be properly identified with text saying what is being printed. We cannot debug your code and/or search the answers inside your code for you. This is a graded component of the course, therefore, we can ONLY evaluate what is uploaded and see printed on the screen. We will test other parameter variations and settings in your code, but we will not alter its structure and logistics. Interpret your code as any other graded component as 'in paper', meaning what you submitted is your final answer and what is not provided (or printed) in terms of answers, we cannot consider for marks.

Submission Information

Combine your written answers for exercises 1 and 2 in a single `pdf` file named `report_assignment_2.pdf`. Additionally, name the code for question 2 as `code_assignment_2.ext` (since this is assignment #2) in which `ext` is the file extension associated with your coding platform. You will upload the `pdf` file of your answers, your code, and generated figures to the Gradescope platform. Your code needs to be documented, i.e., introduce comment lines to explain its main command procedures. Pure code lines without explanatory comments will have reduced marks. **It is important that you log in to Gradescope.ca and not Gradescope.com.** You can upload multiple files to the Gradescope platform and you can resubmit your work until the due date. We will test-run your submitted code for syntax errors and check if it generates the requested figures. We will also check if the results/figures presented in your assignment are 'paper-like' quality and that the quantitative results are scientifically/mathematically founded.

ONLY `pdf` (for the written report), `jpg/jpeg` (for figures), and `.py`, `.ipynb`, or `.m` (for codes) FILES ARE ACCEPTED. This means we are accepting Python, Python Notebook, or Matlab codes. If you are more familiar with another programming language, please, contact the instructor in advance.

IMPORTANT: In your code include instructions on how to run it and, if applicable, include testing values for the initial conditions or settings that you attempted so we can reproduce the results. We will test other initial conditions and parameter variations to evaluate the robustness of your code. But we need a place to start. Include also information about the version of your coding platform. If this information is not included, points will be deducted.

*****