**ASSIGNMENT #2 – COMP 3106 INTRODUCTION TO ARTIFICIAL INTELLIGENCE**

The assignment is an opportunity to demonstrate your knowledge on informed Bayes' Theorem and fuzzy rule-based systems and to practice applying it to a problem.

The assignment may be completed individually, or it may be completed in small groups of two or three students. The expectations will not depend on group size (i.e. same expectations for all group sizes).

Assignment due date: Friday, November 10, 2023

Assignments are to be submitted electronically through Brightspace. It is your responsibility to ensure that your assignment is submitted properly. Copying of assignments is NOT allowed. Discussion of assignment work with others is acceptable but each individual or small group are expected to do the work themselves

**Components**

The assignment should contain two components: an implementation and a technical document.

*Implementation*

Programming language: Python 3

You may use the Python Standard Library (https://docs.python.org/3/library/). You may also use the NumPy, Pandas, Pillow, scikit-image, and SciPy packages (and whatever packages they depend on). Use of any additional packages requires approval of the instructor.

You must implement your code yourself. Do not copy-and-paste code from other sources, but you may use any pseudo-code we wrote in class as a basis for your implementation. Your implementation must follow the outlined specifications. Implementations which do not follow the specifications may receive a grade of zero. Please make sure your code is readable, as it will also be assessed for correctness. You do not need to prove correctness of your implementation.

You may be provided with a set of examples to test your implementation. Note that the provided examples do not necessarily represent a complete set of test cases. Your implementation may be evaluated on a different set of test cases.

The implementation will be graded both on content and use of good programming practices.

Submit the implementation as a single PY file.

*Technical Document*

Your technical document should answer all questions posed below. Ensure you answers are clear and concise.

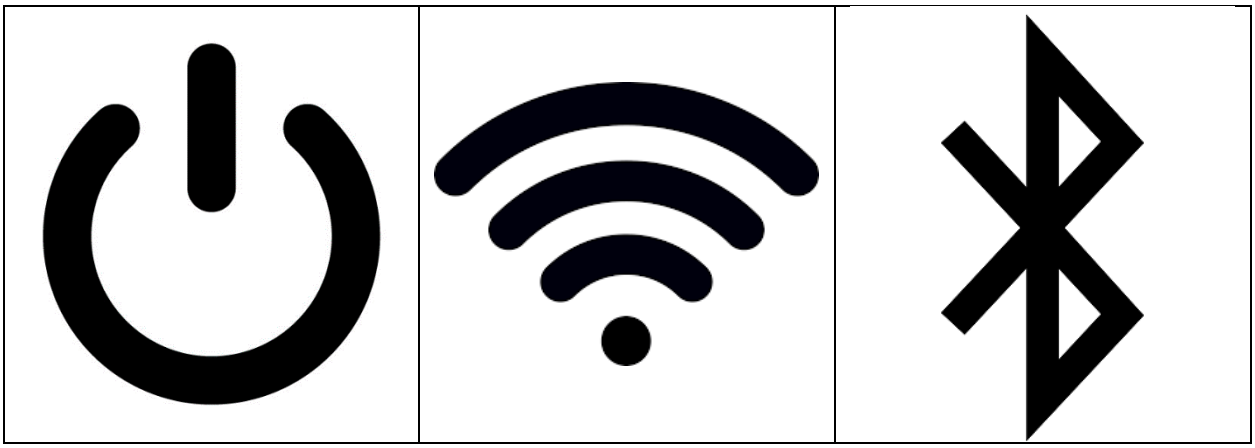
Your technical document must answer all questions posed below. Ensure you answers are clear and concise.
If the assignment was completed in a small group of students, the technical document must include a statement of contributions. This statement should identify: (1) whether each group member made significant contribution, (2) whether each group member made an approximately equal contribution, and (3) exactly which aspects of the assignment each group member contributed to.

Submit the technical document as a single PDF file.

**Implementation**

Consider the problem of classifying what symbol is in an image. In this assignment, we will consider two different classifiers for this task: naïve Bayes' classifier and a fuzzy classifier.

Assume that we only consider three types of symbols: power, Wi-Fi, and Bluetooth (see examples below). We shall assume all images we consider will have one of these symbols.



From the image, we wish to classify which symbol is in the image. To do so, we will assume all images are binary (pixels are either black or white) and extract the following features based on the pixel values.

1. NumBlack: The number of black connected components in the image.
2. NumWhite: The number of white connected components in the image.

Your implementation must contain a single file named "assignment2.py" with two functions (you may have other variables/functions/classes in your file). One function must be named "naive_bayes_classifier"; one function must be named "fuzzy_classifier". The input argument should be the full file path to a comma separated value (CSV) file that contains a matrix representation of the image. The CSV file will have "1" where pixels are black and "0" where pixels are white.

The "naive_bayes_classifier" function should return two values. The first returned value should be a string indicating the most likely class for the input image found using the naïve Bayes' classifier. It should be either "power", "wifi", or "bluetooth". The second returned value should be a Python list indicating the probabilities found using the naïve Bayes' classifier that the input belongs to each class: [power probability, wifi probability, bluetooth probability].

For the naïve Bayes classifier, the probabilities for each feature given each class are provided below. Assume that any probability for a feature value given a class not listed below is zero.

$$P(NumBlack = 1|power) = 0.17$$
$$P(NumBlack = 2|power) = 0.81$$
$$P(NumBlack = 3|power) = 0.02$$

$$P(NumBlack = 2|wifi) = 0.03$$
$$P(NumBlack = 3|wifi) = 0.41$$
$$P(NumBlack = 4|wifi) = 0.52$$
$$P(NumBlack = 5|wifi) = 0.04$$

$$P(NumBlack = 1|bluetooth) = 0.91$$
$$P(NumBlack = 2|bluetooth) = 0.09$$

$$P(NumWhite = 1|power) = 0.79$$
$$P(NumWhite = 2|power) = 0.18$$
$$P(NumWhite = 3|pwer) = 0.03$$

$$P(NumWhite = 1|wifi) = 0.96$$
$$P(NumWhite = 2|wifi) = 0.04$$

$$P(NumWhite = 2|bluetooth) = 0.01$$
$$P(NumWhite = 3|blueooth) = 0.88$$
$$P(NumWhite = 4|bluetooth) = 0.11$$

The prior probability of each setting is given below (following the rough proportion of each setting covering the earth's surface):

$$P(power) = 0.57$$
$$P(wifi) = 0.29$$
$$P(bluetooth) = 0.14$$

The "fuzzy_classifier" function should output two arguments. The first output argument should be a string indicating the output class with the highest membership for the input vector found by the fuzzy classifier. It should be either "power", "wifi", or "bluetooth". The second output argument should be a Python list indicating the combined membership function's value found by the fuzzy classifier at each class: [power value, wifi value, bluetooth value]. The fuzzy membership function and fuzzy rules for the system are provided below. For the fuzzy rules use the Goguen t-norm and the Goguen s-norm.

Assume the fuzzy membership functions are given below.

NumBlack Low:

$$m(x) = \begin{cases} 1 \; if \; x = 0 \\ 0.75 \; if \; x = 1 \\ 0 \; if \; x \geq 2 \end{cases}$$

NumBlack Medium:

$$m(x) = \begin{cases} 0.25 \; if \; x = 1 \; or \; x = 3 \\ 1 \; if \; x = 2 \\ 0 \; if \; x \leq 0 \; or \; x \geq 4 \end{cases}$$

NumBlack High:

$$m(x) = \begin{cases} 0.75 \; if \; x = 3 \\ 1 \; if \; x \geq 4 \\ 0 \; if \; x \leq 2 \end{cases}$$

NumWhite Low:

$$m(x) = \begin{cases} 1 \; if \; x = 0 \\ 0.75 \; if \; x = 1 \\ 0 \; if \; x \geq 2 \end{cases}$$

NumWhite Medium:

$$m(x) = \begin{cases} 0.25 \; if \; x = 1 \; or \; x = 3 \\ 1 \; if \; x = 2 \\ 0 \; if \; x \leq 0 \; or \; x \geq 4 \end{cases}$$

NumWhite High:

$$m(x) = \begin{cases} 0.75 \; if \; x = 3 \\ 1 \; if \; x \geq 4 \\ 0 \; if \; x \leq 2 \end{cases}$$

The fuzzy rules associated with this system are:

IF NumBlack is Medium OR NumWhite is Medium THEN power.
IF NumBlack is High AND NumWhite is Low THEN wifi.
IF NumBlack is Low OR NumWhite is High THEN bluetooth.

Attached are example .csv files and corresponding example text files containing outputs. Note that your function should not write anything to file. These examples are provided in separate files for convenience. Also attached is skeleton code indicating the format your implementation should take.

Hint: The scikit-image package has a built-in function to compute the connected components in an image.

*Grading*

The implementation will be worth 60 marks.

20 marks will be allocated to correctness on a series of test cases for the naïve Bayes classifier, with consideration to each of the two outputs (i.e. most likely class, and class probabilities). 20 marks will be allocated to correctness on a series of test cases for the fuzzy classifier, with consideration to each of the two outputs (i.e. highest membership class, and class memberships). These test cases will be run automatically by calling your implementation from another Python script. To facilitate this, your implementation must adhere exactly to the specifications.

10 marks will be allocated to human-based review of code of the naïve Bayes classifier. 10 marks will be allocated to human-based review of code of the fuzzy classifier. This human-based review will consider both correctness and use of good programming practices.

**Technical Document**

Please answer the following questions in the technical document. For all questions, explain why your answers are correct.

1. Briefly describe how your implementation works. Include information on any important algorithms, design decisions, data structures, etc. used in your implementation. [10 marks]
2. What type of agent have you implemented (simple reflex agent, model-based reflex agent, goal-based agent, or utility-based agent)? [3 marks]
3. Is the task environment: [7 marks]
    a. Fully or partially observable?
    b. Single or multiple agent?
    c. Deterministic or stochastic?
    d. Episodic or sequential?
    e. Static or dynamic?
    f. Discrete or continuous?
    g. Known or unknown?
4. Suppose we wish to measure how well our methods work. Suggest what measure(s) of performance and/or what validation scheme should be used. Assume that we have a labelled set of images available for this task. [6 marks]
5. Suggest a particular feature vector [NumBlack, NumWhite] where the most likely class found using the naïve Bayes classifier is different than the highest membership class found using the fuzzy rule-based system. [4 marks]
6. Suggest a particular feature vector [NumBlack, NumWhite] where the probability of all classes is non-zero for the naïve Bayes classifier. Suggest a particular feature vector [NumBlack, NumWhite] where the membership function for all classes is non-zero for the fuzzy classifier. In both cases, if such a feature vector exists, give the probabilities/memberships associated with it; if such a feature vector does not exist, explain why. [4 marks]
7. In this assignment, the probabilities for the naïve Bayes classifier have been provided. Suppose they were not provided. Suggest how to determine these probabilities given a labelled set of images available for this task. [6 marks]

*Grading*

The technical document will be worth 40 marks, allocated as described above.