ASSIGNMENT #2 – COMP 3106 INTRODUCTION TO ARTIFICIAL INTELLIGENCE

The assignment is an opportunity to demonstrate your knowledge on the Naïve Bayes classifier and 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: 2024-10-29

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 answers to the questions below.

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, and SciPy packages (and any packages they directly 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.

Questions

You must answer all questions posed below. Ensure you answers are clear and concise.

If the assignment was completed in a small group of students, you must also 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 your answers to the questions as a single PDF file.

Implementation

Consider the problem of predicting whether a patient is healthy or diseased using the Naïve Bayes classifier.

In this problem, assume we have a patient who can either be healthy or diseased (treat this as a binary classification problem). We wish to predict whether this patient is healthy or diseased based on two features we measure from the patient.

1. Temperature: The patient's internal body temperature, measured using an oral thermometer (units are degrees Celsius).
2. Heart rate: The patient's resting heart rate, measured using a stethoscope and stopwatch (units are beats per minute).

For this assignment, we will learn the probability density functions for temperature and heart rate for healthy patients and diseased patients from an existing dataset of patients. To do this, assume all features follow a normal distribution, and we can compute the probability a feature takes on a given value as proportional to the normal distribution.

$$P(feature = x|class) \propto \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

The particular μ and σ are the mean and standard deviation for each class and feature. They must be computed from the existing dataset of patients. You may assume the dataset will always have enough patients to compute these quantities.

Likewise, for this assignment, the prior probability of each class (healthy and diseased) must be learned from the existing dataset of patients.

Your implementation must contain a single file named "assignment2.py" with a function named "naïve_bayes_classifier", which takes two input arguments (you may have other variables/functions/classes in your file). The first input argument should be the full file path to a comma separated value (CSV) file that contains the existing patient dataset; the second input argument should be a Python list which contains a never-before-seen patient's measurements [temperature, heart rate].

The CSV file will contain three columns: the first column contains the patient class label (i.e. healthy or diseased), the second column contains temperature, and the third column contains heart rate. Each row in the CSV file refers to a patient in the dataset.

Your function should return two values. The first returned value should be a string indicating the most likely class for the input vector (i.e. "healthy" or "diseased"). The second returned value should be a Python list indicating the probability the input belongs to each class: [healthy probability, diseased probability].

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

Grading

The implementation will be worth 70 marks.

50 marks will be allocated to correctness on a series of test cases, with consideration to each of the two outputs (i.e. most likely class and class probabilities). 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.

20 marks will be allocated to human-based review of code. This human-based review will consider both correctness and use of good programming practices.

Questions

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

1. What type of agent have you implemented (simple reflex agent, model-based reflex agent, goal-based agent, or utility-based agent)? [3 marks]
2. 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?
3. 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. [4 marks]
4. Suggest a particular example instance of this problem where using equal prior probabilities in the naïve Bayes classifier for each class changes the most likely class (compared to using the prior probabilities computed from the dataset). [4 marks]
5. Consider, instead, a fuzzy rule-based system for classifying whether a patient is healthy or diseased with the following rules.
   IF temperature is low AND heart rate is low THEN patient is healthy
   IF temperature is high OR heart rate is high THEN patient is diseased

   All fuzzy membership functions are trapezoidal with parameters given below. Units for temperature are in degrees Celsius and units for heart are in beats per minute.
   Temperature Low: a = 0, b = 0, c = 36, d = 38
   Temperature High: a = 37, b = 39, c = 100, d = 100
   Heart rate Low: a = 0, b = 0, c = 70, d = 90
   Heart rate High: a = 80, b = 100, c = 220, d = 220

   Suppose a patient presents with the following measurements. Compute the highest membership class (i.e. healthy or diseased) and the membership value for each class. [10 marks]
    a. Temperature: 37.5°C, Heart rate: 85 bpm
    b. Temperature: 37.8°C, Heart rate: 88 bpm
    c. Temperature: 37.2°C, Heart rate: 82 bpm
    d. Temperature: 37.7°C, Heart rate: 83 bpm

6. Compare the results from the above question to the results from your naïve Bayes classifier. Briefly comment on why the predicted classes are the same in some cases and different in other cases. [2 marks]

Grading

The technical document will be worth 30 marks, allocated as described above.