Mini-Project 4: Monster Identification (Spring 2023)

In this project, you'll implement an agent that will learn a definition of a particular monster species from a list of positive and negative samples, and then make a determination about whether a newly-provided sample is an instance of that monster species or not. You will submit the code for identifying these monsters to the Mini-Project 4 assignment in Gradescope. You will also submit a report describing your agent to Canvas. Your grade will be based on a combination of your report (50%) and your agent's performance (50%).

About the Project

For the purposes of this project, every monster has a value for each of twelve parameters. The possible values are all known. The parameters and their possible values are:

- size: tiny, small, medium, large, huge
- color: black, white, brown, gray, red, yellow, blue, green, orange, purple
- covering: fur, feathers, scales, skin
- foot-type: paw, hoof, talon, foot, none
- leg-count: 0, 1, 2, 3, 4, 5, 6, 7, 8
- arm-count: 0, 1, 2, 3, 4, 5, 6, 7, 8
- eye-count: 0, 1, 2, 3, 4, 5, 6, 7, 8
- horn-count: 0, 1, 2
- lays-eggs: true, false
- has-wings: true, false
- has-gills: true, false
- has-tail: true, false

A single monster will be defined as a dictionary with those 12 keys. Each value will be one of the values from the corresponding list. The values associated with size, color, covering, and foot-type will be strings; with leg-count, arm-count, eye-count, and horn-count will be integers; and with lays-eggs, has-wings, has-gills, and has-tail will be booleans.

You will be given a list of monsters in the form of a list of dictionaries, each of which has those twelve keys and one of the listed values. Each monster will be labeled as either True (an instance of the species of monster we are currently looking at) or False (*not* an instance of the species of monster we are currently looking at). You will also be given a single unlabeled monster; your goal is to return a prediction—True or False—of whether the unlabeled monster is an instance of the species of monster defined by the labeled list.

Your Agent

To write your agent, download the starter code below. Complete the `solve()` method, then upload it to Gradescope to test it against the autograder. Before the deadline, make sure to select your best performance in Gradescope as your submission to be graded.

Starter Code

Here is your starter code: MonsterClassificationAgent.zip. You may also access the code at the course's Github repository.

The starter code contains two files: MonsterClassificationAgent.py and main.py. You will write your agent in MonsterClassificationAgent.py. You may test your agent by running main.py. You will only submit MonsterClassificationAgent.py; you may modify main.py to test your agent with different inputs.

Your `solve()` method will have two parameters. The first will be a list of 2-tuples. The first item in each 2-tuple will be a dictionary representing a single monster. The second item in each 2-tuple will be a boolean representing whether that particular monster is an example of this new monster species. The second parameter to `solve()` will be a dictionary representing the unlabeled monster.

Each monster species might have multiple possible values for each of the above parameters. One monster species, for instance, include monsters with either 1 or 2 horns, but never 0. Another species might include monsters that can be red, blue, and yellow, but no other colors. Another species might include both monsters with and without wings. So, while each monster is defined by a single value for each parameter, the species as a whole may have more variation.

Returning Your Solution

Your `solve()` method should return True or False based on whether your function believes this new monster (the second parameter) to be an example of the species defined by the labeled list of monsters (the first parameters).

Not every list will be fully exhaustive. Your second parameter could, for example, feature a monster that is a color that never appeared as positive or negative in the list of samples. Your agent's task is to make an educated guess. For example, you might determine, "The only difference between this monster and the positive examples is its color, and its color never appeared in the negative examples, therefore there is a good likelihood that this is still a positive example."

You may assume that the parameters are independent; for example, you will not have any species that has one horn when yellow and two horns when blue, but never one horn when blue. You may assume that all parameters are equally likely to occur; for example, you will not have any species that is yellow 90% of the time and blue only 10% of the time. Those ratios may appear in the list of samples you receive, but the underlying distribution of possibilities will be even. You may assume that these parameters are all that there is: if two monsters have the exact same parameters, they are guaranteed to be the same species. Finally, you should assume that each list is independent: you should not use knowledge from a prior test case to inform the current one.

Submitting Your Solution

To submit your agent, go to the course in Canvas and click Gradescope on the left side. Then, select CS7637 if need be.

You will see an assignment named Mini-Project 4. Select this project, then drag your MonsterClassificationAgent.py file into the autograder. If you have multiple files, add them to a zip file and drag that zip file into the autograder.

When your submission is done running, you'll see your results.

How You Will Be Graded

Your agent will run against 20 test cases. The first four of these will always be the same; these are those contained in the original main.py. The last 16 will be randomly generated.

You can earn up to 40 points. Because the list of labeled monsters is non-exhaustive, it is highly unlikely you can write an agent that classifies every single monster correctly; there will always be some uncertainty. For that reason, you will receive full credit if your agent correctly classifies 17 or more of the monsters. Similarly, because every label is a simple true/false, even a randomly performing agent can likely get 50% correct with no intelligence under the hood. For that reason, you will receive no credit if your agent correctly classifies 7 or fewer monsters.

Between 7 and 17, you will receive 4 points for each correct classification: 4 points for 8/20, 8 for 9/20; 12 for 10/20; and so on, up to 40 points for correctly classifying 17 out of 20 or better.

You may submit up to 40 times prior to the deadline. The large majority of students do not need nearly that many submissions, so do not feel like you should use all 40; this cap is in place primarily to prevent brute force methods for farming information about patterns in hidden test cases or submitting highly random agents hoping for a lucky submission. Note that Gradescope has no way for us to increase your individual number of submissions, so we cannot return submissions to you in the case of errors or other issues, but you should have more than enough submissions to handle errors if they arise.

You **must** select which of your submissions you want to count for a grade prior to the deadline. Note that by default, Gradescope marks your last submission as your submission to be graded. We cannot automatically select your best submission. Your agent score is worth 50% of your overall mini-project grade.

Your Report

In addition to submitting your agent to Gradescope, you should also write up a short report describing your agent's design and performance. Your report may be up to 4 pages, and should answer the following questions:

- How does your agent work? Does it use some concepts covered in our course? Or some other approach?
- How well does your agent perform? Does it struggle on any particular cases?
- How efficient is your agent? How does its performance change as the number of labeled monsters grows?
- Does your agent do anything particularly clever to try to arrive at an answer more efficiently?
- How does your agent compare to a human? Do you feel people approach the problem similarly?

You are encouraged but not required to include visuals and diagrams in your four page report. The primary goal of the report is to share with your classmates your approach, and to let you see your classmates' approaches. You may include code snippits if you think they are particularly novel, but please do not include the entirety of your code.

> **Tip:** Remember, we want to see how you put the content of this class into action when designing your agent. You don't need to use the principles and methods from the lectures precisely, but we want to see your knowledge of the content reflected in your terminology and your reflection.

Submission Instructions

Complete your assignment using JDF, then save your submission as a PDF. Assignments should be submitted to the corresponding assignment submission page in Canvas. You should submit a **single** PDF for this assignment. This PDF will be ported over to Peer Feedback for peer review by your classmates. If your assignment involves things (like videos, working prototypes, etc.) that cannot be provided in PDF, you should provide them separately (through OneDrive, Google Drive, Dropbox, etc.) and submit a PDF that links to or otherwise describes how to access that material.

This is an individual assignment. All work you submit should be your own. Make sure to cite any sources you reference, and use quotes and in-line citations to mark any direct quotes.

Late work is not accepted without advanced agreement except in cases of medical or family emergencies. In the case of such an emergency, please contact the Dean of Students.

Grading Information

Your report is worth 50% of your mini-project grade. As such, your report will be graded on a 40-point scale coinciding with a rubric designed to mirror the questions above. Make sure to answer those questions; if any of the questions are irrelevant to the design of your agent, explain why.

Peer Review

After submission, your assignment will be ported to Peer Feedback for review by your classmates. Grading is *not* the primary function of this peer review process; the primary function is simply to give you the opportunity to read and comment on your classmates' ideas, and receive additional feedback on your own. All grades will come from the graders alone. See the course participation policy for full details about how points are awarded for completing peer reviews.