CS 335: Final Project

Francisco Iacobelli

1 TASK

You will have to find a problem that you think can be solved with: A*, Naïve Bayes, Forward checking, Backtracking, or Minimax.

These can be toy problems in the same category as the n-queens, wumpus, etc.

Please look at the resources section of the course's website for datasets you may want to use if you need them.

1.1 Sample problems you can pick

- Scheduling courses that have time constraints
- Scheduling courses that have instructor constraints
- Finding the best path in a map with max. speed and distance routes.
- Finding researchers with similar interest that may not know each other
- A probabilistic minesweep player
- A minimax Tic-Tac-Toe, connect four, checkers, etc.
- Any classification problem using Naive Bayes (movie reviews, diagnosis, etc.)
- Any other problem that you have run by me.

2 CONSIDERATIONS

Your work must comply with the following:

- Your python program must not contain any external libraries except for those that come with a basic installation of python (math, re, sys, os, etc.)
- Your code should be able to run from the command line.
- Your program should be named `final_project.py`
- filenames and options should be given through the command line, not through user prompts. That is, you should be able to type `python final_project.py [option1 option2 etc.]` where `optionX` can be a filename, or whatever parameters your program requires.
- Your program should allow a "help" option that displays how to use the program. That is, `python final_project help` should display a little help on how to use your program and what to expect.

- You must have a PDF file called `final_project.PDF`(no other names nor formats are allowed) with a description of how the program works internally. You must describe all main functions used. Use the following template:

  Objective
  Describe what the program does in one sentece or two. (for example: this program plays tic-tac-toe in a computer vs. computer mode using minimax).
  Implementation Details
  Describe important details that will help me understand your code. (for example: the tree is stored as a dictionary called `my-game` and the function `get-heur` obtains the heuristic of the board wich consists of counting..... etc.)
  Running the program
  Describe exactly how to run the program in my computer. (for example: The program must be run by executing `python final_project.py number` where `number` is .....)

- If there is any piece of code or data or method you are taking from a foreign source (internet, wikipedia, etc.) mention it in the write-up. At least 70% of your project must be original work.

2.1 Rubric

- Submission complies with rules (20pt)
- Appropriate problem chosen (20pts)
- Document is clear and conforms to specs (20pts)
- Program is correct (does not crash, has a help option, and can be run following the specs) (20pts)
- Program's output is clear and correct (20pts)

3 SUBMIT

Submit only ONE ZIP file with all the files needed to successfully run your program as well as the PDF describing it. Any data files should be included in the ZIP file. The python file(s) must comply with the following convention (THIS IS VERY IMPORTANT)

- The first line of your file should indicate the python version as follows:
  - If you are using a flavor of python 3.x, your first line should be: `#!/usr/bin/env python3`
  - If you are using a flavor of python 2.7.x your first line should be: `#!/usr/bin/env python2`
- The second line should have the name of the homework and optionally a couple of words about it. These should be enclosed in three quotation marks. For example: `""" Eliza homework. Relationship advisor """`

- The third line should have your name assigned to the `__author__` variable. For example if your name is "John Doe" your next line should be:

  `__author__="John Doe"`

- Optionally, you can specify a file encoding on your second line and then follow the convention. You do this by adding the following line as a second line: `# # -*- coding: utf-8 -*-`

A sample hello world file created with python 2.7.x for John Doe looks like this:

```
#!/usr/bin/env python2
# # -*- coding: utf-8 -*-
""" Hello World program """
__author__="John Doe"

print "Hello World"
```

The same file created with python 3.x looks like:

```
#!/usr/bin/env python3
# # -*- coding: utf-8 -*-
""" Hello World program """
__author__="John Doe"

print ("Hello World")
```