CAP 4630 – Introduction to Artificial Intelligence
Project 3 – Recommender System – Report

Joseph Allen
25 March 2018

This recommender system consists of 4 python scripts, 3 test files, a testing information file, a README file, and this PDF report.

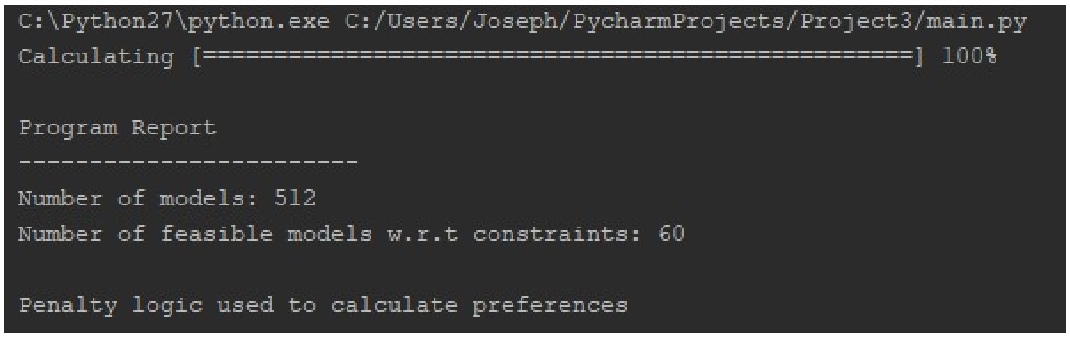
Project Contents:

- *main.py* – main script that runs the gui.py, problem.py, and solver.py scripts. Execute this script to run the program.
- *gui.py* – script the creates GUIs to collect problem data and then returns three file names corresponding to the attributes, constraints, and preferences data.
- *problem.py* – script that creates a Problem object from the files given by gui.py, parses the files for data, and then stores them in a manner readable by Clasp. It also writes the constraints to a file to run clasp on.
- *solver.py* – script that creates a Solver object from a Problem object. The solver performs all the reasoning tasks of the system. Each task is a function within the Solver class and all of this information is printed by the toString() function.
- *attributes.txt, constraints.txt, preferences.txt* – files that contain the data that make up the instance used to test the system.
- *README.txt* – file containing instructions to build and run the system.
- *Project3Report.pdf* – this report which shows how the program works.

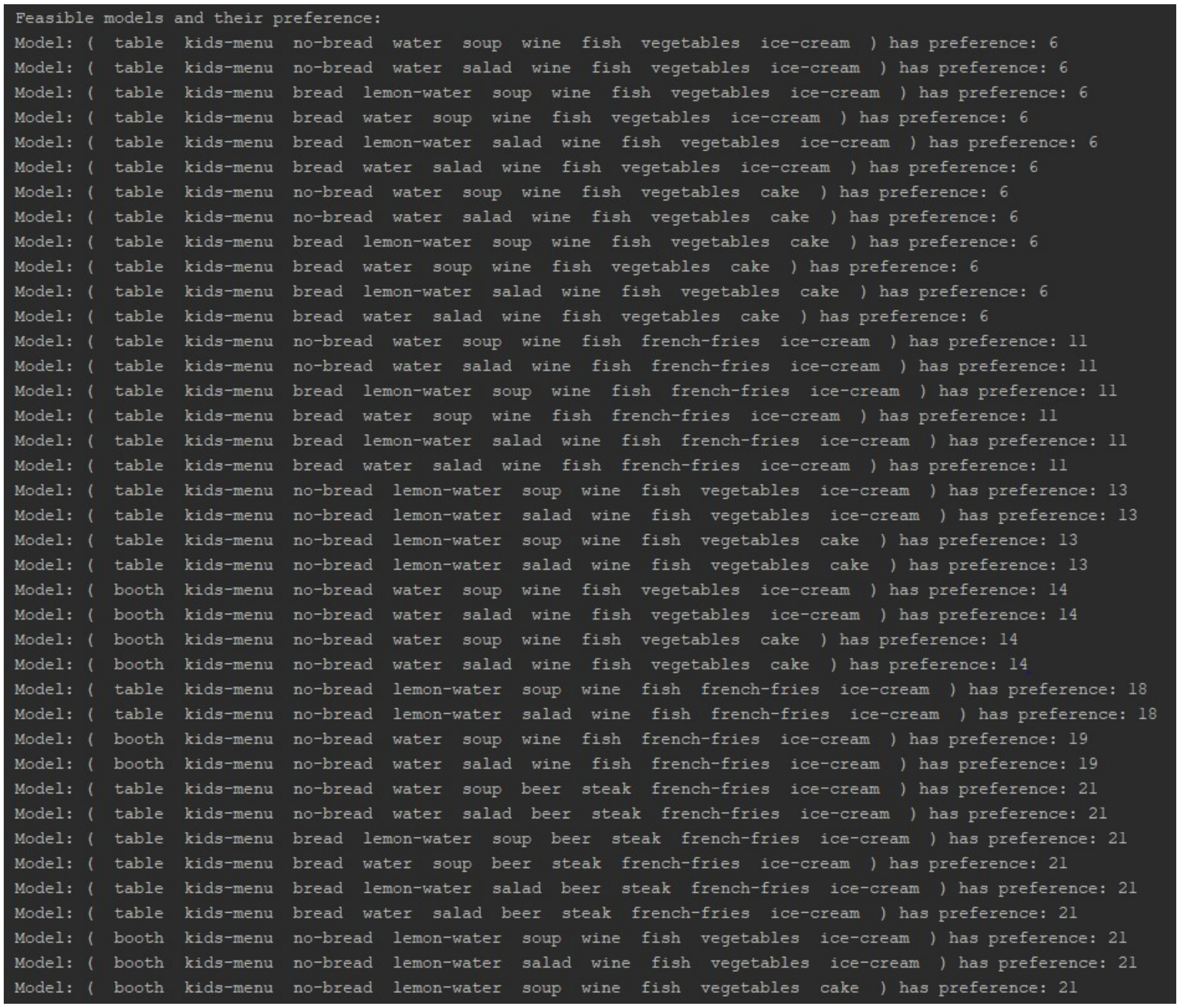
The program performs 5 main reasoning tasks:

- Existence of feasible object – decides whether there are feasible objects with respect to *H* (the constraints), that is, whether there are models of *H* that are truth assignments making H true.
- Calculating preference – finds the total preference of all feasible objects.
- Exemplification – generates, if possible, two random feasible objects, and shows how the preferences between the two compare (strict preference or equivalence).
- Optimization – finds an optimal object with respect to *T* (the preferences).
- Omni-optimization – find all optimal objects with respect to T.

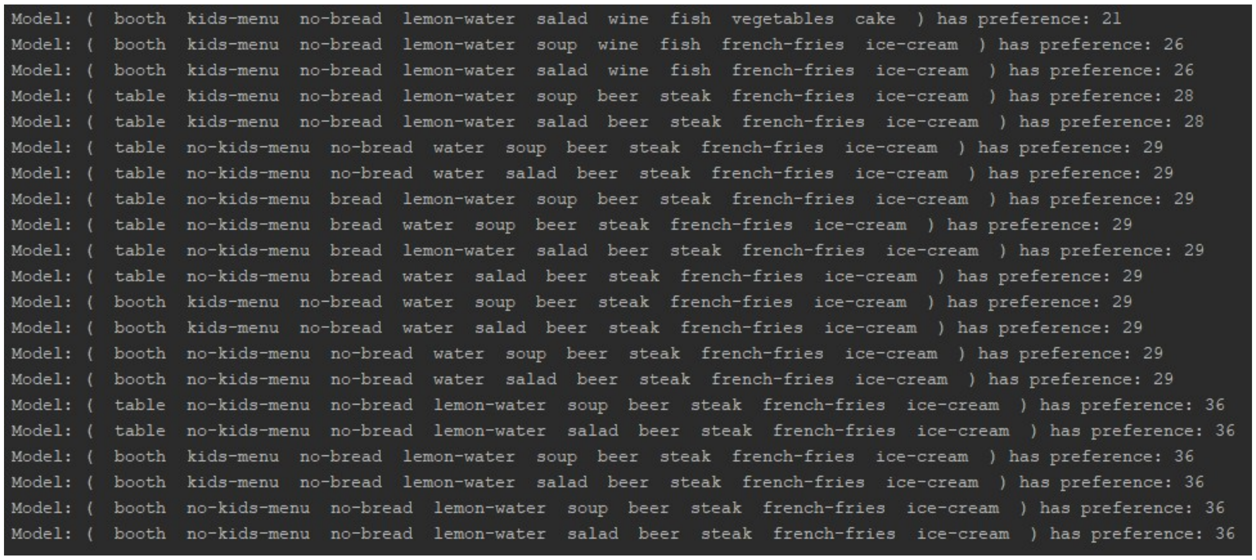
To demonstrate these reasoning tasks, I've included screenshots of the output produced when the program is run on the test instance. The screenshot below shows the beginning of the printed report which displays the total number of objects, and how many of those are feasible with respect to the constraints. The next page demonstrates the next task and shows the objects and their preference.


```
C:\Python27\python.exe C:/Users/Joseph/PycharmProjects/Project3/main.py
Calculating [==================================================] 100%

Program Report
------------------------
Number of models: 512
Number of feasible models w.r.t constraints: 60

Penalty logic used to calculate preferences
```

All feasible models and their calculated preference (Note in the above image which logic was used):

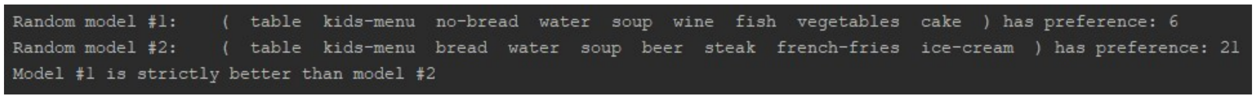


```
Feasible models and their preference:
Model: (  table  kids-menu  no-bread  water  soup  wine  fish  vegetables  ice-cream  ) has preference: 6
Model: (  table  kids-menu  no-bread  water  salad  wine  fish  vegetables  ice-cream  ) has preference: 6
Model: (  table  kids-menu  bread  lemon-water  soup  wine  fish  vegetables  ice-cream  ) has preference: 6
Model: (  table  kids-menu  bread  water  soup  wine  fish  vegetables  ice-cream  ) has preference: 6
Model: (  table  kids-menu  bread  lemon-water  salad  wine  fish  vegetables  ice-cream  ) has preference: 6
Model: (  table  kids-menu  bread  water  salad  wine  fish  vegetables  ice-cream  ) has preference: 6
Model: (  table  kids-menu  no-bread  water  soup  wine  fish  vegetables  cake  ) has preference: 6
Model: (  table  kids-menu  no-bread  water  salad  wine  fish  vegetables  cake  ) has preference: 6
Model: (  table  kids-menu  bread  lemon-water  soup  wine  fish  vegetables  cake  ) has preference: 6
Model: (  table  kids-menu  bread  water  soup  wine  fish  vegetables  cake  ) has preference: 6
Model: (  table  kids-menu  bread  lemon-water  salad  wine  fish  vegetables  cake  ) has preference: 6
Model: (  table  kids-menu  bread  water  salad  wine  fish  vegetables  cake  ) has preference: 6
Model: (  table  kids-menu  no-bread  water  soup  wine  fish  french-fries  ice-cream  ) has preference: 11
Model: (  table  kids-menu  no-bread  water  salad  wine  fish  french-fries  ice-cream  ) has preference: 11
Model: (  table  kids-menu  bread  lemon-water  soup  wine  fish  french-fries  ice-cream  ) has preference: 11
Model: (  table  kids-menu  bread  water  soup  wine  fish  french-fries  ice-cream  ) has preference: 11
Model: (  table  kids-menu  bread  lemon-water  salad  wine  fish  french-fries  ice-cream  ) has preference: 11
Model: (  table  kids-menu  bread  water  salad  wine  fish  french-fries  ice-cream  ) has preference: 11
Model: (  table  kids-menu  no-bread  lemon-water  soup  wine  fish  vegetables  ice-cream  ) has preference: 13
Model: (  table  kids-menu  no-bread  lemon-water  salad  wine  fish  vegetables  ice-cream  ) has preference: 13
Model: (  table  kids-menu  no-bread  lemon-water  soup  wine  fish  vegetables  cake  ) has preference: 13
Model: (  table  kids-menu  no-bread  lemon-water  salad  wine  fish  vegetables  cake  ) has preference: 13
Model: (  booth  kids-menu  no-bread  water  soup  wine  fish  vegetables  ice-cream  ) has preference: 14
Model: (  booth  kids-menu  no-bread  water  salad  wine  fish  vegetables  ice-cream  ) has preference: 14
Model: (  booth  kids-menu  no-bread  water  soup  wine  fish  vegetables  cake  ) has preference: 14
Model: (  booth  kids-menu  no-bread  water  salad  wine  fish  vegetables  cake  ) has preference: 14
Model: (  table  kids-menu  no-bread  lemon-water  soup  wine  fish  french-fries  ice-cream  ) has preference: 18
Model: (  table  kids-menu  no-bread  lemon-water  salad  wine  fish  french-fries  ice-cream  ) has preference: 18
Model: (  booth  kids-menu  no-bread  water  soup  wine  fish  french-fries  ice-cream  ) has preference: 19
Model: (  booth  kids-menu  no-bread  water  salad  wine  fish  french-fries  ice-cream  ) has preference: 19
Model: (  table  kids-menu  no-bread  water  soup  beer  steak  french-fries  ice-cream  ) has preference: 21
Model: (  table  kids-menu  no-bread  water  salad  beer  steak  french-fries  ice-cream  ) has preference: 21
Model: (  table  kids-menu  bread  lemon-water  soup  beer  steak  french-fries  ice-cream  ) has preference: 21
Model: (  table  kids-menu  bread  water  soup  beer  steak  french-fries  ice-cream  ) has preference: 21
Model: (  table  kids-menu  bread  lemon-water  salad  beer  steak  french-fries  ice-cream  ) has preference: 21
Model: (  table  kids-menu  bread  water  salad  beer  steak  french-fries  ice-cream  ) has preference: 21
Model: (  booth  kids-menu  no-bread  lemon-water  soup  wine  fish  vegetables  ice-cream  ) has preference: 21
Model: (  booth  kids-menu  no-bread  lemon-water  salad  wine  fish  vegetables  ice-cream  ) has preference: 21
Model: (  booth  kids-menu  no-bread  lemon-water  soup  wine  fish  vegetables  cake  ) has preference: 21
```



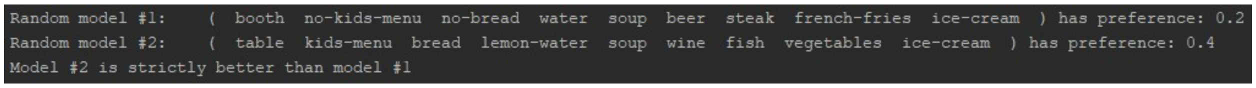
```
Model: (  booth  kids-menu  no-bread  lemon-water  salad  wine  fish  vegetables  cake  ) has preference: 21
Model: (  booth  kids-menu  no-bread  lemon-water  soup  wine  fish  french-fries  ice-cream  ) has preference: 26
Model: (  booth  kids-menu  no-bread  lemon-water  salad  wine  fish  french-fries  ice-cream  ) has preference: 26
Model: (  table  kids-menu  no-bread  lemon-water  soup  beer  steak  french-fries  ice-cream  ) has preference: 28
Model: (  table  kids-menu  no-bread  lemon-water  salad  beer  steak  french-fries  ice-cream  ) has preference: 28
Model: (  table  no-kids-menu  no-bread  water  soup  beer  steak  french-fries  ice-cream  ) has preference: 29
Model: (  table  no-kids-menu  no-bread  water  salad  beer  steak  french-fries  ice-cream  ) has preference: 29
Model: (  table  no-kids-menu  bread  lemon-water  soup  beer  steak  french-fries  ice-cream  ) has preference: 29
Model: (  table  no-kids-menu  bread  water  soup  beer  steak  french-fries  ice-cream  ) has preference: 29
Model: (  table  no-kids-menu  bread  lemon-water  salad  beer  steak  french-fries  ice-cream  ) has preference: 29
Model: (  table  no-kids-menu  bread  water  salad  beer  steak  french-fries  ice-cream  ) has preference: 29
Model: (  booth  kids-menu  no-bread  water  soup  beer  steak  french-fries  ice-cream  ) has preference: 29
Model: (  booth  kids-menu  no-bread  water  salad  beer  steak  french-fries  ice-cream  ) has preference: 29
Model: (  booth  no-kids-menu  no-bread  water  soup  beer  steak  french-fries  ice-cream  ) has preference: 29
Model: (  booth  no-kids-menu  no-bread  water  salad  beer  steak  french-fries  ice-cream  ) has preference: 29
Model: (  table  no-kids-menu  no-bread  lemon-water  soup  beer  steak  french-fries  ice-cream  ) has preference: 36
Model: (  table  no-kids-menu  no-bread  lemon-water  salad  beer  steak  french-fries  ice-cream  ) has preference: 36
Model: (  booth  kids-menu  no-bread  lemon-water  soup  beer  steak  french-fries  ice-cream  ) has preference: 36
Model: (  booth  kids-menu  no-bread  lemon-water  salad  beer  steak  french-fries  ice-cream  ) has preference: 36
Model: (  booth  no-kids-menu  no-bread  lemon-water  soup  beer  steak  french-fries  ice-cream  ) has preference: 36
Model: (  booth  no-kids-menu  no-bread  lemon-water  salad  beer  steak  french-fries  ice-cream  ) has preference: 36
```

With the preference calculations done, the final 3 reasoning tasks are possible. The first image below shows two randomly selected feasible objects and how their preferences compare using penalty logic. In this case, the first random model was strictly better than the second, because in penalty logic a lower preference rating is better.



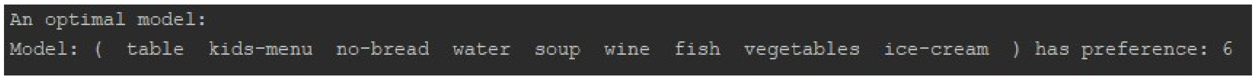
```
Random model #1:    (  table  kids-menu  no-bread  water  soup  wine  fish  vegetables  cake  ) has preference: 6
Random model #2:    (  table  kids-menu  bread  water  soup  beer  steak  french-fries  ice-cream  ) has preference: 21
Model #1 is strictly better than model #2
```

The task is carried out similarly when possibilistic logic is used, except that a higher preference rating is better. See below how two random feasible objects with possibilistic logic preferences compare.



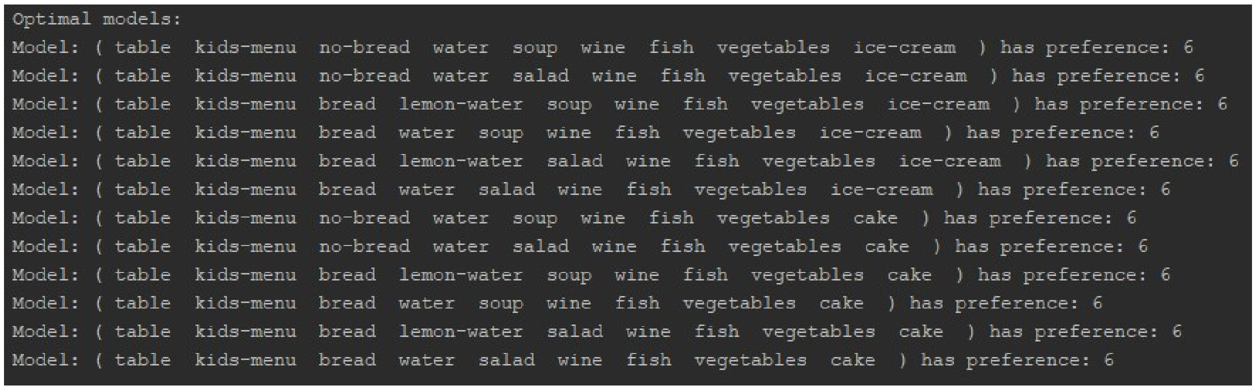
```
Random model #1:    (  booth  no-kids-menu  no-bread  water  soup  beer  steak  french-fries  ice-cream  ) has preference: 0.2
Random model #2:    (  table  kids-menu  bread  lemon-water  soup  wine  fish  vegetables  ice-cream  ) has preference: 0.4
Model #2 is strictly better than model #1
```

The next reasoning task is to simply find any optimal object. Depending on which logic was used to calculate preferences, it is either any object with the highest or lowest preference.



```
An optimal model:
Model: (  table  kids-menu  no-bread  water  soup  wine  fish  vegetables  ice-cream  ) has preference: 6
```

Similar to the previous reasoning task, calculating all of the optimal objects entails finding the set of all objects with the highest or lowest preference rating. In the case of this test instance, the optimal preference rating was a 6.

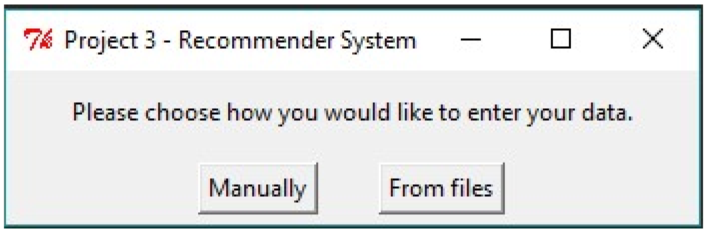


```
Optimal models:
Model: ( table  kids-menu  no-bread  water  soup  wine  fish  vegetables  ice-cream  ) has preference: 6
Model: ( table  kids-menu  no-bread  water  salad  wine  fish  vegetables  ice-cream  ) has preference: 6
Model: ( table  kids-menu  bread  lemon-water  soup  wine  fish  vegetables  ice-cream  ) has preference: 6
Model: ( table  kids-menu  bread  water  soup  wine  fish  vegetables  ice-cream  ) has preference: 6
Model: ( table  kids-menu  bread  lemon-water  salad  wine  fish  vegetables  ice-cream  ) has preference: 6
Model: ( table  kids-menu  bread  water  salad  wine  fish  vegetables  ice-cream  ) has preference: 6
Model: ( table  kids-menu  no-bread  water  soup  wine  fish  vegetables  cake  ) has preference: 6
Model: ( table  kids-menu  no-bread  water  salad  wine  fish  vegetables  cake  ) has preference: 6
Model: ( table  kids-menu  bread  lemon-water  soup  wine  fish  vegetables  cake  ) has preference: 6
Model: ( table  kids-menu  bread  water  soup  wine  fish  vegetables  cake  ) has preference: 6
Model: ( table  kids-menu  bread  lemon-water  salad  wine  fish  vegetables  cake  ) has preference: 6
Model: ( table  kids-menu  bread  water  salad  wine  fish  vegetables  cake  ) has preference: 6
```

That concludes the output of the program and the task it performs. See the next page for images of the GUI that this program uses to collect data.

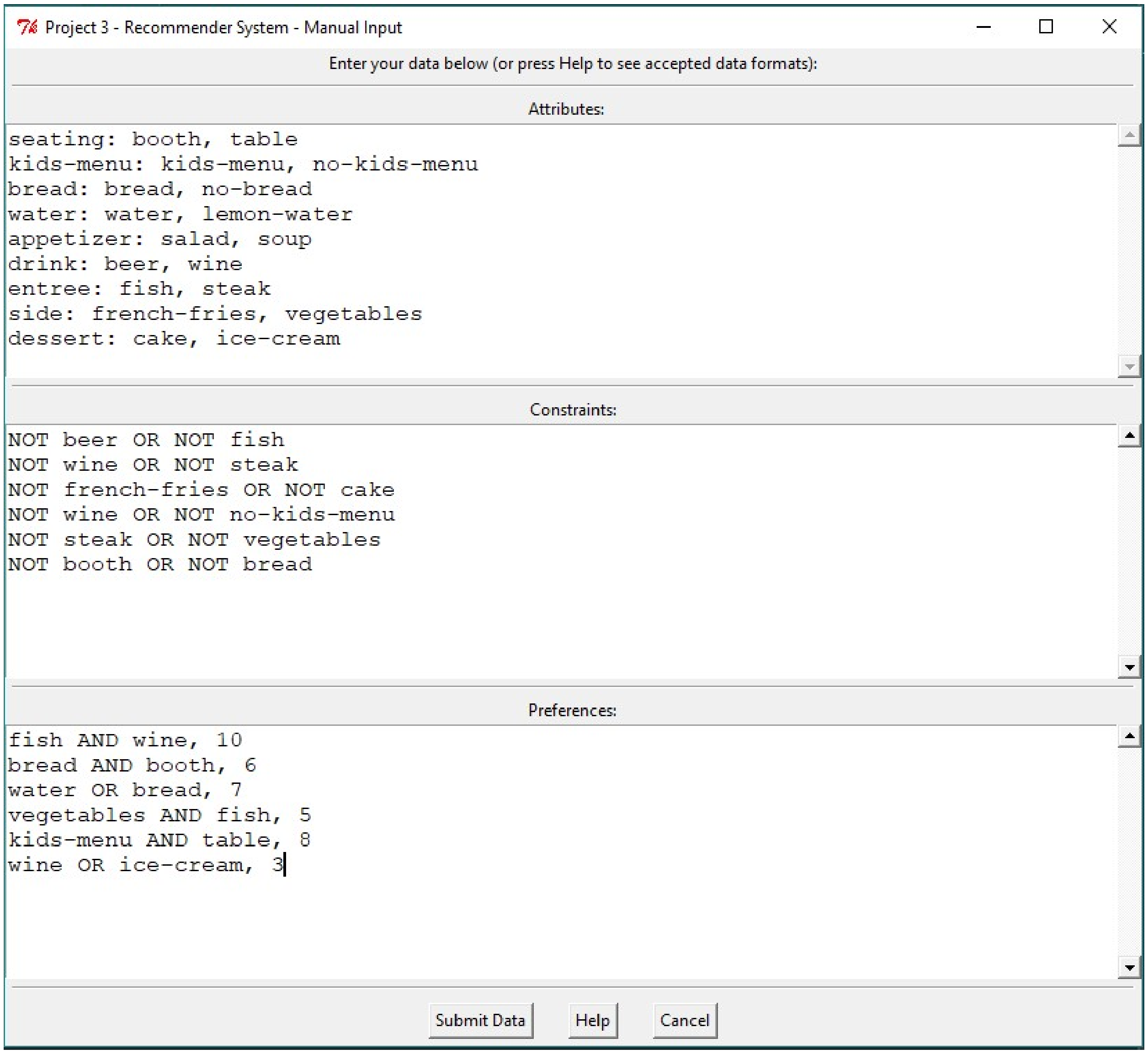
Graphical User Interface Images

The GUI's main window that appears when main is executed:



Please choose how you would like to enter your data.

The window that appears when the user selects to enter their data manually:



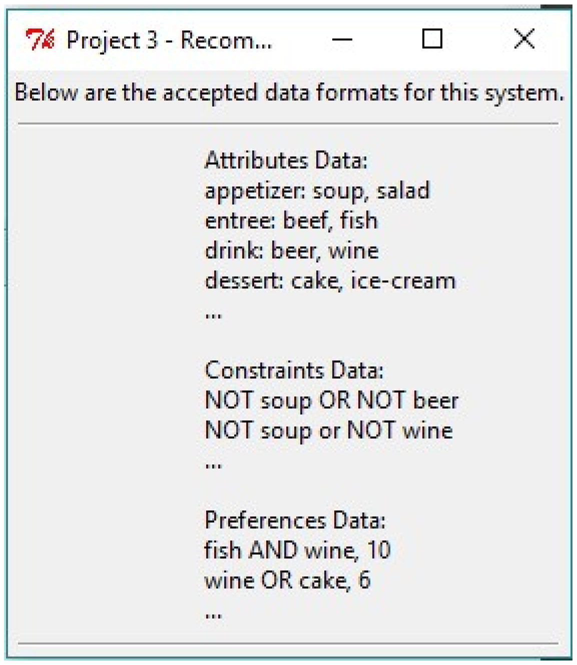
Enter your data below (or press Help to see accepted data formats):

```
seating: booth, table
kids-menu: kids-menu, no-kids-menu
bread: bread, no-bread
water: water, lemon-water
appetizer: salad, soup
drink: beer, wine
entree: fish, steak
side: french-fries, vegetables
dessert: cake, ice-cream
```

```
NOT beer OR NOT fish
NOT wine OR NOT steak
NOT french-fries OR NOT cake
NOT wine OR NOT no-kids-menu
NOT steak OR NOT vegetables
NOT booth OR NOT bread
```

```
fish AND wine, 10
bread AND booth, 6
water OR bread, 7
vegetables AND fish, 5
kids-menu AND table, 8
wine OR ice-cream, 3
```

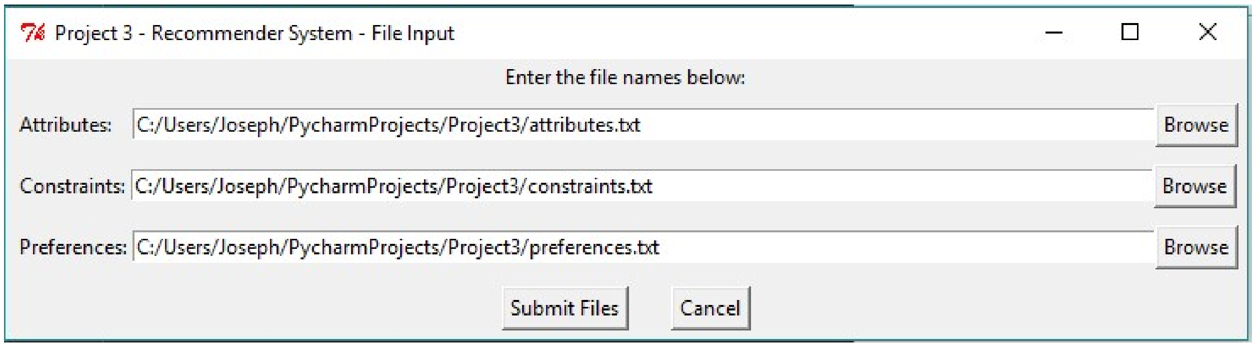
The help window that appears when the help button is pressed:



Below are the accepted data formats for this system.

```
Attributes Data:
appetizer: soup, salad
entree: beef, fish
drink: beer, wine
dessert: cake, ice-cream
...

Constraints Data:
NOT soup OR NOT beer
NOT soup or NOT wine
...

Preferences Data:
fish AND wine, 10
wine OR cake, 6
...
```

The window that appears when the user decides to submit their data through files:



Enter the file names below:

Attributes: C:/Users/Joseph/PycharmProjects/Project3/attributes.txt

Constraints: C:/Users/Joseph/PycharmProjects/Project3/constraints.txt

Preferences: C:/Users/Joseph/PycharmProjects/Project3/preferences.txt