Assignment 8 - Show a Little Class

Robin Dawes

November 30, 2021

This assignment asks you to use an Object Oriented Programming approach to determine a ranking of restaurants and customers.

.

The Problem

Grubstakes Analytics has collected data based on a number of restaurants and customers. They want to use the data to produce a ranking of the restaurants, and a ranking of the customers.

The Data

The data is in the form of a table with one row for each customer and one column for each restaurant. In the table, the entry for a particular person and a particular restaurant is 1 if the customer has eaten at that restaurant, and 0 if they have not.

The data file is "Customers and Restaurants 2.csv". The values on each line are separated by **commas**

The Task

Your job is to produce a list of the top 10 restaurants and the top 10 customers, based on a scoring method for restaurants and a different scoring method for customers.

Restaurant Scoring

Each restaurant's score is equal to the sum of the number of restaurants each of its customers has eaten at.

Example:

If restaurant "The Angry Ostrich" has 3 customers Tim, Tom and Tam, and those 3 customers have eaten at 4, 9 and 6 restaurants respectively, then "The Angry Ostrich" has score $4 + 9 + 6 = 19$

For ranking Restaurants, we compare the scores. Ties are broken by length of the Restaurant name. If the Restaurants are still tied, alphabetical order of the names is used to break the tie. The name which is alphabetically earlier gets the higher ranking.

Examples:

Suppose "Sally's Soups and Salads" and "Mucho Muncho" have equal scores. "Sally's Soups and Salads" is ranked above "Mucho Muncho" because it has a longer name.

Suppose "Corners" and "Grillit" have equal scores. Their names both have length = 7. "Corners" is ranked above "Grillit" because its name would precede the other name in the dictionary.

Customer Scoring

Each customer's score is the average of the scores of the restaurants at which they have eaten.

Example:

Tam has eaten at 6 restaurants. Suppose those restaurants' scores are $\{12, 3, 19, 23, 19, 6\}$. Tam's score is $\frac{82}{5} = 16.4$

For ranking Customers, we compare the scores. Ties are broken by alphabetical order of Customer name. The name which is alphabetically earlier gets the higher ranking.

Example:

Suppose "Mary Brown" and "Cary Grant" have equal scores. "Cary Grant" is ranked above "Mary Brown" because "Cary Grant" would be placed earlier than "Mary Brown" in alphabetical order.

Method

You have been provided with a class definition called Assignment8. It is designed to be the parent class of two new classes called Restaurant and Customer. You must use these classes to complete the ranking task. You may add attributes and functions to all three classes. In particular, you will need to override the "compute_score" function in each of the Customer and Restaurant classes. You will also add functions to define comparison operators (such as: equal, not equal, less than) so that you can use the built-in "sorted()" function to find the top Restaurants and Customers, as required by the assignment.

Restaurant and Customer Classes

Each class must define (or inherit):

1. an appropriate "__init__" function
2. an attribute to hold links to all the objects of the other type that this object is related to (i.e. for a Restaurant, these links would be to the Customers that eat there, and for a Customer, these links would be to the Restaurants where they eat)
3. a function that properly computes the score of the Object
4. a set of functions for comparing objects of this type
5. an appropriate "add_connection" function that only adds connections to Objects of the proper type. Restaurants can only have connections to Customers, and Customers can only have connections to Restaurants.

Revision to Assignment

Original Requirement:

If you choose to add attributes to any of the classes, follow the convention that all attributes must be named using __ at the start of the name.

Revised Requirement:

Since we established in class that putting __ at the beginning of an attribute name **does not** actually make the attribute **private**, you are not required to use this naming convention in this assignment. Be prepared for CISC-124 though - in Java the use of **private** attributes is important.

It turns out there is a huge debate - sometimes quite heated - about whether Python should have **private** attributes in the same way that Java does. Those who say yes argue that having **private** attributes makes software behaviour more reliable and predictable. Those who say no claim that even with **private** attributes, there are ways to get around the privacy and access/modify these attributes. They claim that **private** attributes only give an illusion of security.

Solution Steps

1. Create the Restaurant Objects and the Customer Objects using the names found in the data file.
2. Use the values in the data file to give each Customer Object a list of the Restaurants at which they have eaten, and to give each Restaurant Object a list of the Customers who have eaten there.
3. Compute the score for each Restaurant, and sort the Restaurants.
4. Compute the score for each Customer, and sort the Customers.
5. Report the top Restaurants and top Customers.

How You Will Be Graded

The assignment will be marked out of 100. 90% of the grade will be for correctness and 10% of the grade will be for programming style.

The grader will read your code and will run your program to test correctness.

What to Submit

For this assignment, you are required to upload to onQ:

- A Python program that defines the required classes and performs the ranking as specified above.
- A text file or pdf containing your ranked list of the top 10 Restaurants and the top 10 Customers.
- You are NOT required to upload the html page generated by pydoc, because I know some students in the class have not been able to get this to work. However, your code must be properly documented with docstrings at the beginning of the program, at the beginning of each class, and in each defined function.

Remember to put your name and student number at the top of your program file, as well as the statement regarding academic integrity (as specified in Assignment 1).

Due Date

The due date for this assignment is 20211206 (December 6), 11:59 PM.