Aabaan Samad

JavaCC1 - Attempt 2

Test Platform: Visual Studio Code Version 1.97.0

Language: Java version "20" 2023-03-21

Java(TM) SE Runtime Environment (build 20+36-2344)

Java HotSpot(TM) 64-Bit Server VM (build 20+36-2344, mixed mode, sharing)

Description:

This is a program to rank diners based on their preferences using a similarity matrix calculated using Spearman’s Rank Correlation and a recursive function that finds the optimal permutation of the diners using a HashMap to keep track of all possible similarity scores and its corresponding permutation and an ArrayList to keep track of the positions of the diners. I use a Diner class to add structure to the program and to make keeping track of each diner easier. The algorithm is exhaustive with an O(N!) time complexity and goes through all possible combinations. It uses the Collections library to generate random choices for each of the diners using the in built Collections shuffle method and an ArrayList containing the choices 1 through 5 to represent each choice.

I had first thought of using a more comparative method as can be seen in the commented out method (AssignPositions)to compare each score in each row of the similarity matrix, however, that method only considered the first best score it came across and disregarded subsequent scores, so although it allowed the first 2 diners to be ranked correctly and to the most accurate degree, it did not allow the last 2 diners to be ranked correctly, and instead just listed them in numerical order. It was dependent on where the first best score occurred and did not backtrack to check for future score comparisons. The updated code is good for small inputs of data but will take more space and time if the number of inputs increases.

I gained help from my Database Management professor who gave me ideas of how to go about calculating the concept of similarity who gave me the idea of using K means clustering which led me to eventually using Spearman’s Ranking Coefficient. I also used Wikipedia to find the formula to calculate Spearman’s Ranking Coefficient. One of my friends gave me the idea to use an exhaustive permutation-based algorithm which I then implemented myself in Java.

Testing Screenshots: A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.