Project Progress Report Submission for Grading
Intelligent Dining Decision Assistant Application

Group Members:

- Pauline Brunet, NetID: pbrunet2
- Catherine Orlando, NetID: co24
- Austin Harmon, NetID: austin31
- Mitchell Kopczyk, NetID: kopczyk2
- Louis Hamilton, NetID: louisch3

Recap of Topic Chosen:
Building an application that aids indecisive users in making informed dining choices by providing personalized restaurant recommendations and insights.

Recap of Problem and Relevance to Theme and Class:
Choosing a restaurant can be a complex decision-making process, made even more challenging due to the vast amount of online information available. Our application intends to make this process more manageable and intelligent. The chosen topic aligns with our Text Information Systems class by incorporating concepts such as semantic analysis, relevance, probability ranking, and collaborative filtering to improve the accuracy and relevance of restaurant recommendations and insights.

Completed tasks

1. Initial Project Proposal Draft
   - Conducted research and brainstorming.
   - Drafted the initial proposal outlining the project's goals and methodology.
   - Total hours spent: 10 hours (2 hours/student)

2. Project Proposal Edit and Proofread
   - Revised, refined, and proofread the proposal for clarity.
   - Ensured that the proposal met all project requirements and guidelines.
   - Total hours spent: 5 hours (1 hour/student)

3. First Program Draft Development
   - Developed the initial version of the program, focusing on core functionalities.
   - Conducted basic testing and debugging.
   - Total hours spent: 30 hours (6 hours/student)

6. Initial Progress Report
   - Compiled and documented the project's progress, identifying completed tasks.
   - Total hours spent: 2 hours (0.4 hours/student)

7. Progress Report Edit and Proofread
   - Reviewed and revised the progress report, ensuring accuracy and clarity.
   - Total hours spent: 2 hours (0.4 hours/student)

Tasks in progress

4. Second Program Draft Development
   - Improved and expanded upon the initial code.
   - Conducted further testing, identifying areas for improvement and optimization.
   - Total hours spent: 25 hours (5 hours/student)

Tasks to be completed

5. Final Program Draft Development
   - Conducted final refinements to optimize the code.
   - Carried out comprehensive testing and debugging to ensure functionality and reliability.
   - Total hours spent: 20 hours (4 hours/student)

8. Program Documentation Development
   - Developed detailed documentation explaining the program's code and functionalities.
   - Ensured that the documentation is clear and understandable.
   - Total hours spent: 3 hours (0.6 hours/student)

9. Program Presentation
   - Prepared a cohesive presentation to showcase the project's development process and final product.
   - Practiced the presentation to ensure smooth delivery.
   - Total hours spent: 3 hours (0.6 hours/student)

Challenges faced

- Challenge 1: When the program was initially being developed, every time a user entered a query with the word "the", all documents would be returned. A query, such as "the salad", would return a list of all the restaurants in the database. The word "the" also affected the relevance calculation. Our current solution is to prevent the addition of the words "the" and "and" to the inverted index. This change improves the probability that the user will only see relevant results.

- Challenge 2: Allowing the user to enter multiple queries and keep the program running continuously. To solve this problem, it was necessary to use a Tkinter GUI with a loop. With this new addition, the results of each query can be maintained, since they are added to a Text Widget. Users can scroll through their query and result history by scrolling through the Text Widget with all the results.

- Challenge 3: Addition of a location filtering feature, to improve user experience. The initial version of the program did not allow users to know the location of a restaurant. To solve this issue, the city and state of each restaurant was added to its respective document. The program was modified to check the second line of each file to read the location of each restaurant. This modification allows the user to filter the search results by entering a city and state in a separate text box on the GUI interface.