ENGR 212 Programming Practice
Mini Project 2

October 30, 2016

In this mini Project you are going to develop a simple GUI based book recommendation system for you and your friends to choose what to read based on your personal preferences, and ratings by several different people. You will be given a dataset containing the ratings given by several people to a number of books. Your program will also give you the proper interface in order to input several other ratings via a user interface. Details regarding the requirements are as follows:

Your program should have a graphical user interface (GUI) which will look like as shown in Figure. You may use any coloring-scheme you like. The below one is for illustration purposes only.

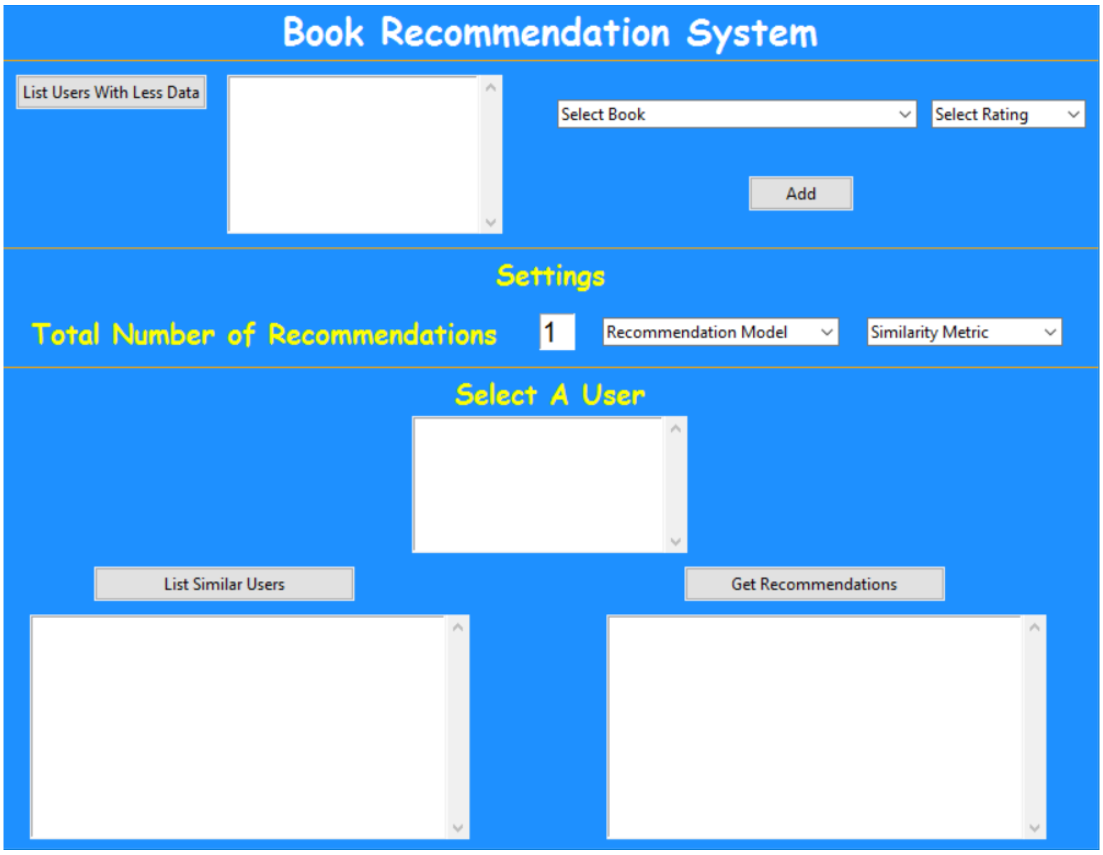
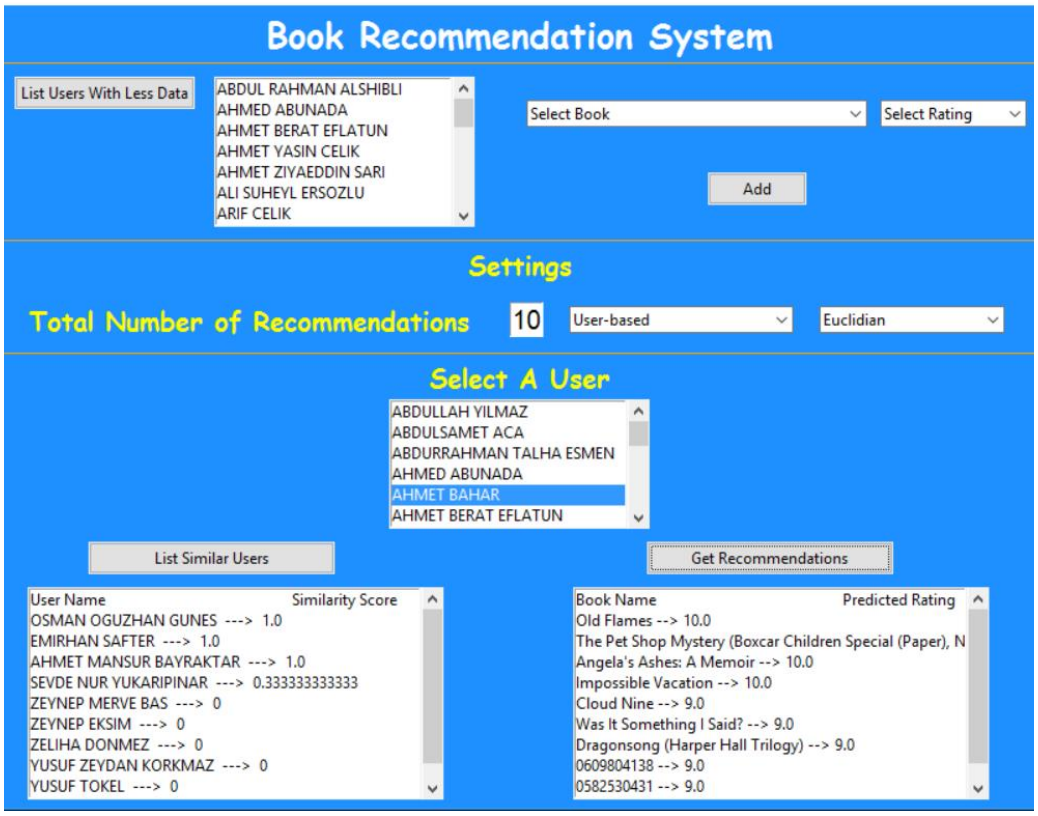


Figure 1

- Your program, upon startup, should input and parse a dataset from a CSV (comma separated values) files containing the ratings given by several people to several book titles, as input. The example file is provided within the project folder on LMS. Your programs will be tested with either that files or with files in similar formats. After parsing the file, you should construct a similar data structure we have seen in the lectures for keeping track of user ratings, so that you can use the Python codes directly from the textbook. Your data structure should be based on book names rather than the ISBN strings, just like the example in the text book.

- As we know, for a recommendation system, the more data, the better the resulting recommendations will be. When the button labelled "**List Users With Less Data**" is pressed, the program should list the names, **sorted alphabetically**, of your friends who have rated less than **two** books in the **listbox** shown in the figure above. The user should be able to enter more ratings for those people. The next part of the user interface is for this purpose. The available book titles should be extracted from the dataset loaded during the above step, and you should populate the corresponding **combo box** (**Select Book**) with appropriate entries. The user will select a friend name from the list of names produced, a book name from the Select Book combo box, then select his/her rating (within a 1 to 10 scale, **Select Rating**) for the selected book, and whenever the "**Add**" button is pressed, adds the entered data to the user ratings data structure.

- In the next section, the user will have the ability to edit the configuration settings for the recommendation engine. "Total Number of Recommendations" specifies the maximum number of selections to be used for various recommendation types. The "recommendation model" should be selected either as "User-based", or "Item-based", and specifies the approach to be used while getting the recommendations (see below). Finally, the "similarity metric" should be selected as one of "Euclidian", "Pearson", or "Jaccard" to be used later for similarity score computations.
- The part next to the "Settings" will give the user an option to select one of his/her friends, and see other people with similar tastes regarding the book preferences. When the "List Similar Users" button is pressed by the user, a number of user equal to the setting given before should be listed together with their respective similarity scores.
- Finally, the bottom part will show the actual book recommendations to the friend selected above. When the "Get Recommendations" button is pressed, depending on the recommendation approach selected above, the program should execute the proper functions, and list the resulting recommendations in a list-box.



```
ABDUL RAHMAN ALSHIBLI
AHMED ABUNADA
AHMET BERAT EFLATUN
AHMET YASIN CELIK
AHMET ZIYAEDDIN SARI
ALI SUHEYL ERSOZLU
ARIF CELIK
```

Total Number of Recommendations 10 User-based Euclidian

```
ABDULLAH YILMAZ
ABDULSAMET ACA
ABDURRAHMAN TALHA ESMEN
AHMED ABUNADA
AHMET BAHAR
AHMET BERAT EFLATUN
```

```
User Name                       Similarity Score
OSMAN OGUZHAN GUNES ---> 1.0
EMIRHAN SAFTER ---> 1.0
AHMET MANSUR BAYRAKTAR ---> 1.0
SEVDE NUR YUKARIPINAR ---> 0.333333333333
ZEYNEP MERVE BAS ---> 0
ZEYNEP EKSIM ---> 0
ZELIHA DONMEZ ---> 0
YUSUF ZEYDAN KORKMAZ ---> 0
YUSUF TOKEL ---> 0
```

```
Book Name                                  Predicted Rating
Old Flames --> 10.0
The Pet Shop Mystery (Boxcar Children Special (Paper), N
Angela's Ashes: A Memoir --> 10.0
Impossible Vacation --> 10.0
Cloud Nine --> 9.0
Was It Something I Said? --> 9.0
Dragonsong (Harper Hall Trilogy) --> 9.0
0609804138 --> 9.0
0582530431 --> 9.0
```

Figure 2

After getting a set of recommendations, the user should be able to modify any part in any of the regions of the interface, and can then click on the "List Similar Users" or "Get Recommendations" button again to get a new set of users and recommendations based on the latest settings and ratings. You **should not** use the place geometry manager in any part of the project.

Can you provide any further pointers that may be helpful? :

- You may use csv module to read csv files or you can use plain open() command in python. There are various tutorials on the Internet.
- For various selections (select track, own rating, recommendation model, etc.), you can use the ComboBox widget of Tkinter. The following example code piece may help:
    - http://stackoverflow.com/questions/17757451/simple-ttk-combobox-demo
- In several parts, you will need to provide a vertical scrollbar to accommodate entries that does not fit to the widget. The following link shows an example for how to add a vertical scrollbar.
    - http://www.java2s.com/Tutorial/Python/0360__Tkinker/ListBoxwithscrollbar.htm
- If you get an error while trying to parse books as some have Unicode characters in their titles, you may have a look at this page
    - http://stackoverflow.com/questions/10268518/python-string-to-unicode,

Warnings:

- **Do not** talk to your classmates on project topics when you are implementing your projects. **Do not** show or email your code to others. If you need help, talk to your TAs or myself, not to your classmates. If somebody asks you for help, explain them the lecture slides, but do not explain any project related topic or solution. Any similarity in your source codes will have **serious** consequences for both parties.
- Carefully read the project document, and pay special attention to sentences that involve "**should**", "**should not**", "**do not**", and other underlined/bold font statements.
- If you use code from a resource (web site, book, etc.), make sure that you reference those resource at the top of your source code file in the form of comments. You should give details of which part of your code is from what resource. Failing to do so **may result in** plagiarism investigation.
- Even if you work as a group of two students, each member of the team should know every line of the code well. Hence, it is **important** to understand all the details in your submitted code. You may be interviewed about any part of your code.

How and when do I submit my project? :

- Projects may be done individually or as a small group of two students (doing it individually is recommended). If you are doing it as a group, only **one** of the members should submit the project. File name will tell us group members (Please see the next item for details).
- Submit your own code in a **single** Python file (Do **not** include recommendations.py that you import). Name it with your and your partner's first and last names (see below for naming).
    - If your team members are Deniz Barış and Ahmet Çalışkan, then name your code file as deniz_baris_ahmet_caliskan.py (Do **not** use any Turkish characters in file name).
    - If you are doing the project alone, then name it with your name and last name similar to the above naming scheme.

- Submit it online on LMS (Go to the Assignments Tab) by **17:00 on November 13, 2016**.

Late Submission Policy:

- -10%: Submissions between 17:01 – 18:00 on the due date
- -20%: Submissions between 18:01 – midnight (00:00) on the due date
- -30%: Submissions which are 24 hour late.
- -50%: Submissions which are 48 hours late.
- Submission more than 48 hours late will not be accepted.

Grading Criteria? :

Code Organization			Functionality						
Meaningful variable names (%5)	Classes and objects used (%5)	Sufficient commenting (%5)	Compiles % Runs? (30)	GUI Design (without the Place Geometry Manager) (15)	Listing users with less data (20)	Adding new data to the dataset (20)	Settings (10)	Listing Similar Users & Getting Recommendations (10)	ReRun Functionality (10)

Have further questions? :

- Contact your TA's during their office hours, and please do NOT hesitate to ask the instructor (me) also (by e-mail or in-person), about anything related to the project.

Good Luck!