ENGR 212 – PROGRAMMING PRACTICE
SPRING 2015
MINI PROJECT 5

May 17, 2015

On Istanbul Sehir University's official website, the list of faculty members for each school and department is provided[1]. Each faculty member's name is linked to a profile page which usually provides a short bio of the faculty member[2]. Sometimes, publications of the faculty member is also listed on the profile page[3].

In this project, you are going to build a tool that will predict the department of a given SEHIR faculty member based on his/her profile on SEHIR web site. Figure 1 shows a sample view of your program's user interface.

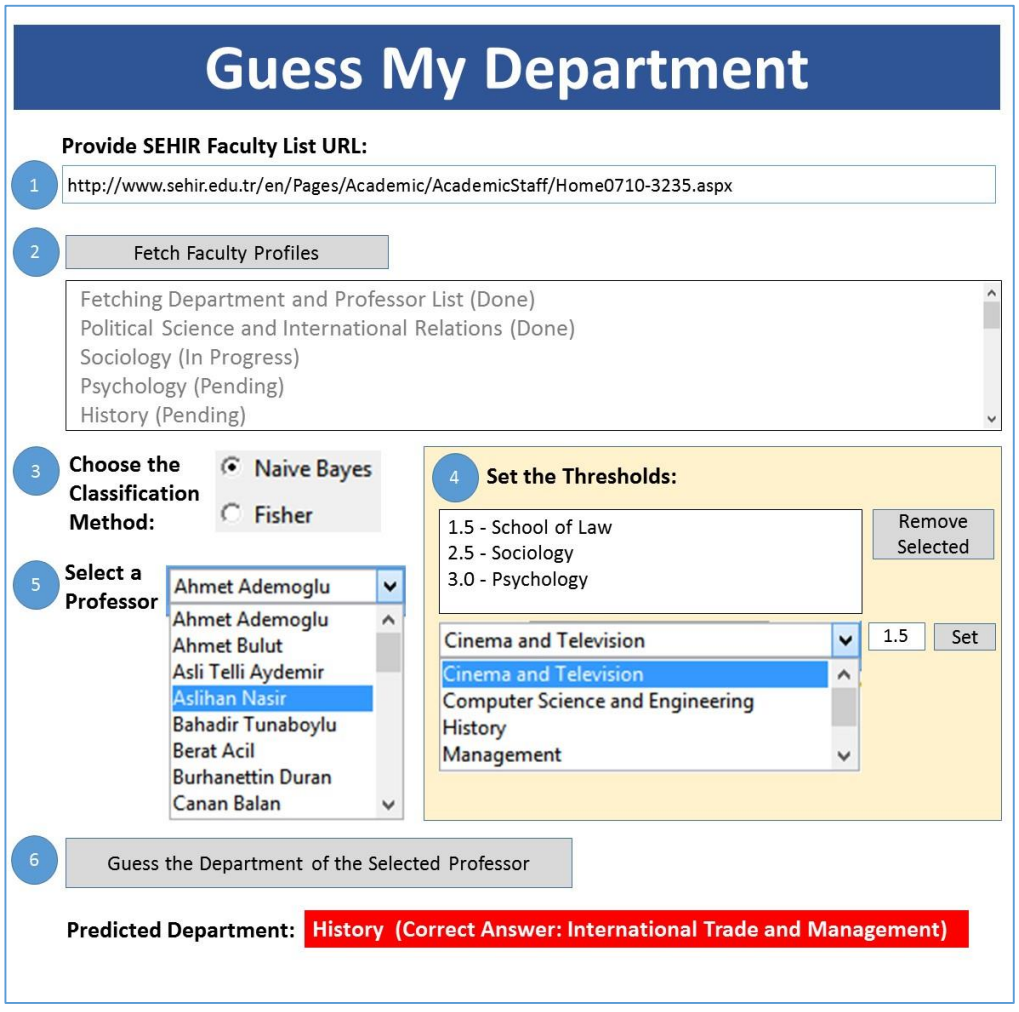

Figure 1

Below are detailed explanations describing how your program should work.

- First, the user will provide the URL to a SEHIR web page that lists faculty members for each school and department (Step 1). For testing this project, please use the link in footnote 1.
- Then, the user will click on "Fetch Faculty Profiles" button (Step 2). At this step, your program first fetch and extract the list of schools/departments and faculty members for each school/department. Your program should show the status messages as shown in Figure 1. The

[1] http://www.sehir.edu.tr/en/Pages/Academic/AcademicStaff/Home0710-3235.aspx
[2] http://www.sehir.edu.tr/en/Pages/Academic/AcademicList.aspx?akademid=260
[3] http://www.sehir.edu.tr/en/Pages/Academic/AcademicList.aspx?akademid=149

status panel should be populated as soon as the list of departments/schools are fetched with all departments initially in "pending" status. Then, your program should download professor profile pages (by following links on professor names) for each department/school one by one (in the same order as listed on the SEHIR web page). As one department/school is completed, and the other is started, the status panel should be updated properly so that the user can follow your program's progress. On faculty profile pages, your program should exclude the parts that are marked in Figure 2. In the classification task, each profile page will be considered as a document, and department name of the faculty member will be considered as the category/label of the document.

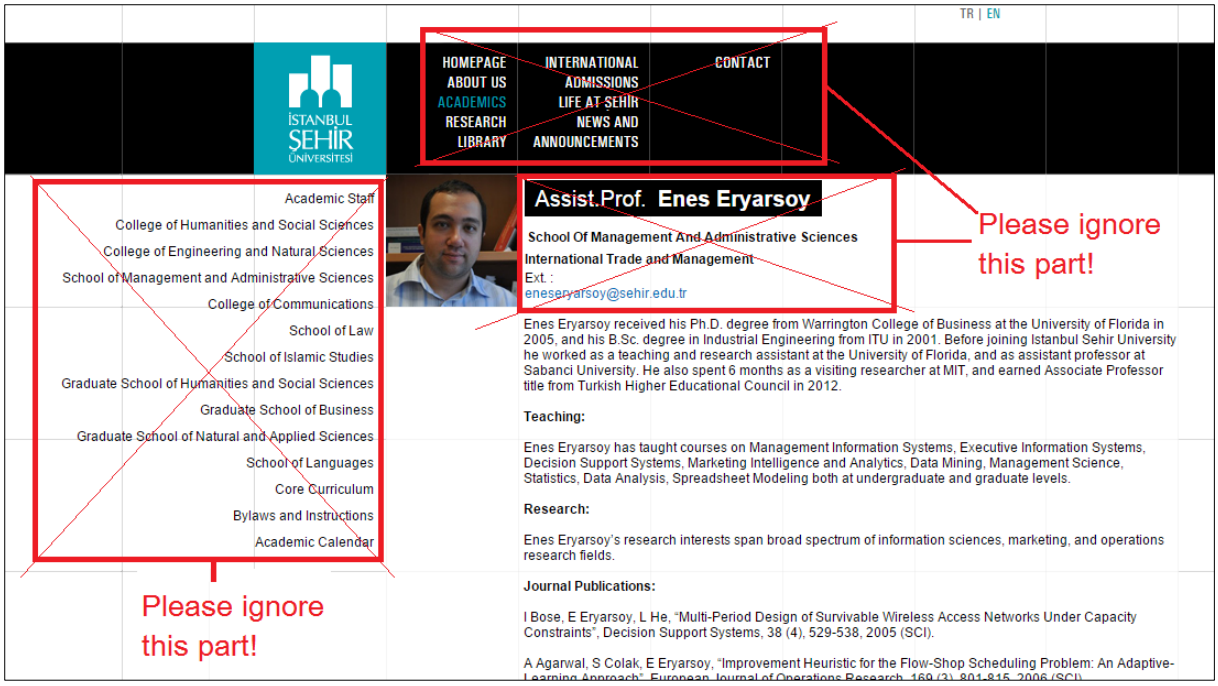

Figure 2

- After all professor profiles are downloaded, in step 3, the user will choose the classification method they prefer to use.
- Next, optionally, the user may set thresholds for categories (i.e., departments/schools in this project) that will be considered during classification (Step 4). Here, a listbox (initially empty) will be used to list the currently set threshold values for each department/school. The user can set a threshold for a department/school by selecting a school from the combo box (located under the listbox), and entering a value in the entry field next to the combo box. This combo box will be automatically populated by your program after your program fetches the list of department names in the initial phase of step 2. The department names should be sorted alphabetically for easy navigation. When the user clicks on the set button, the threshold should be properly set using the methods in docclass.py (setthreshold() in naivebayes, and setminimum() in fisherclassifier). The set threshold should be listed in the above listbox as shown in Figure 1. The user should be able to select any previously set threshold in the listbox, and click on the "Remove Selected" button to remove the selected threshold from the listbox (this corresponds to setting the removed threshold to their default values, i.e., 1 for naivebayes, 0 for fisherclassifier). If the user sets a new threshold for a department/school for which the threshold is already set, the latest set threshold should be used and displayed in the listbox properly. You may use "unknown" if no department could be assigned to a professor after setting some thresholds.
- Next, through a combo box, the user will choose a professor whose department will be predicted by the tool (step 5). This combo box will be automatically populated by your program after your program fetches the list of faculty names in the initial phase of step 2. The professor names should be sorted alphabetically for easy navigation.
- Finally, the user will click on "Guess the Department of the Selected Professor" button (step 6), and see the predicted department at the bottom (The values in this document are not from a

real run. Hence, do not compare them to yours). Here, when this button is clicked, your program should (i) create a proper classifier object depending on the user-selected classifier type, (ii) train it (by calling the train() method) with all faculty member profile pages **except the profile page of the faculty member selected in Step 5**, and then (iii) predict the department of the selected professor (by calling the classify() method), If the prediction is not correct, the tool should color the background of the prediction result text as red, and provide the correct department information in parenthesis (Figure 1). If the prediction is accurate, background color of the prediction result text should be green (Figure 2).

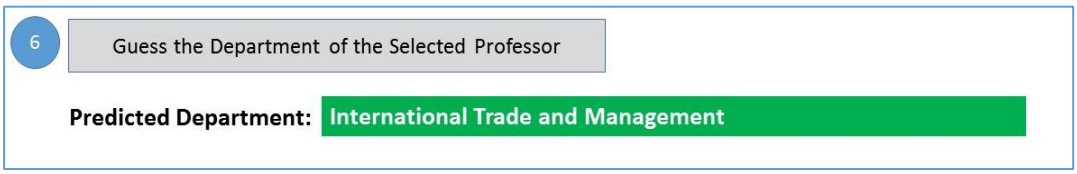

Figure 2

- Your program should allow the user to repeat step 6 by changing parameters in any of steps 3, 4, and 5 without needing to re-download faculty profiles (that is, without repeating step 2).

Important Notes:

- Most schools/colleges have departments under them (e.g., College of Engineering and Natural Sciences has three departments (EE, CS, and IE)). However, some schools/colleges have no individual departments under them (e.g., School of Law). When there is no department under a school/college, you program should consider school/college name as a department name. In all other cases where there are individual departments under schools/colleges, the college/school name should be ignored, and individual department names should be used in your program (e.g., in combo boxes, status messages, etc.). You should not hardcode which schools have no departments under them. Your program should be able to figure this out automatically during the parsing in step 2.
- You should **not** hard code any professor name or department/school name in your code. Your program should extract such information from the URL provided in step 1. This will make sure that if a new department/school is added to the university, or a faculty member joins/leaves the university, your program is not needed to be updated.

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

- As for the GUI, you **should use** Tkinter that we have covered last semester (see ENGR 211 last week's slides). If you do not like Tkinter, you may use PyQt (we have not covered that in ENGR 211), but in any case, you are **not** allowed to use a designer or any other GUI module (other than the above ones).
- In order to be able to place widgets as shown in Figures, you should heavily use frames. Please see ENGR 211, last week's slides for creating row, col, and grid frames, and their example uses with Swampy.
- To display status messages (in step 2) and to list the set thresholds (in step 4), please use ListBox. These ListBox widgets should have a vertical scrollbar. The following link shows an example for how to add a vertical scrollbar.
    - http://www.java2s.com/Tutorial/Python/0360__Tkinker/ListBoxwithscrollbar.htm
- To remove the selected item in the listbox in step 4, you need to find out selected item in the listbox. With the project files, a sample code is included in testlistboxselection.py that demonstrates getting the selected value or index in a listbox. You may use that as an example.
- You may want to use update_idletasks() call to make your update messages appear in the status box in step 2.
    - http://stackoverflow.com/questions/6588141/update-a-tkinter-text-widget-as-its-written-rather-than-after-the-class-is-fini

- If you have problems with Turkish characters in professor names, please check out the Unicode tutorial that you were presented in the practice session two weeks ago (available on LMS). In particular, you may sometimes need to use str.encode('utf8') method,

How and when do I submit my project? :

- Projects may be done individually or as a small group of two students. 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 docclass.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.
- Do **not** copy/paste code from docclass.py into your own code file. Anything that you need from docclass.py should be called with proper dot notation after importing that module.
- Do **not** use any external module other than Swampy, urllib2, BeautifulSoup which are not included in standard Python installation.
- Do **not** use Python 3.x. Use Python 2.7.x.
- Submit it online on LMS (Go to the Assignments Tab) by **17:00 on May 31 (Sunday)**.

    Late Submission Policy:

    - -20%: Submissions between 17:01 – midnight (00:00) on the due date.
    - -40%: 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? :

- Does it run? (Submissions that do not run will get some partial credit which will not exceed 30% of the overall project grade).
- Does it implement all the features according to the specifications and produce correct results?
- Code organization (Meaningful names, sufficient and appropriate comments, proper organization into functions and classes, clean and understandable, etc.)?
- Interview evaluation.

Have further questions? :

Please contact your TAs (Mehmet Aytimur and Muhammed Esad Unal) if you have further questions. If you need help with anything, **please use the office hours** of your TAs and the instructor to get help. If office hours are not suitable, please **get** an appointment through email before walking in your TAs offices.