

ENGR 212 Programming Practice

Mini Project 6

May 9, 2016 (Due: May 23, 2016)

Student life is full of challenges and difficulties like deciding what movie to watch each night or deciding at what campus you should have your lunch so that you would minimize your walking time to the afternoon classes. Perhaps, the hardest to deal with challenge is to decide on what classes you would take during the registration time at the beginning of each semester. Most students would like to choose classes that they will be successful and get good grades. In this mini project, you are going to help students with this challenge.

What? : You are going to write a Python program that will act as a fortune teller for students, and make predictions about their future grades for courses that they have not taken.

How? : Using the web scraping and document filtering skills (?) that you have learnt in the class!

Input? : Past grades of a student mapped on a curriculum (in an excel file), and the links to the course descriptions as provided on SEHIR web pages.

Output? : Your software will provide the predicted grades for the courses that are listed on the curriculum, and have not been taken by the student yet.

Your program will have a graphical user interface (GUI) which will look like as shown in Figure 1. Details about how it should work are provided below.

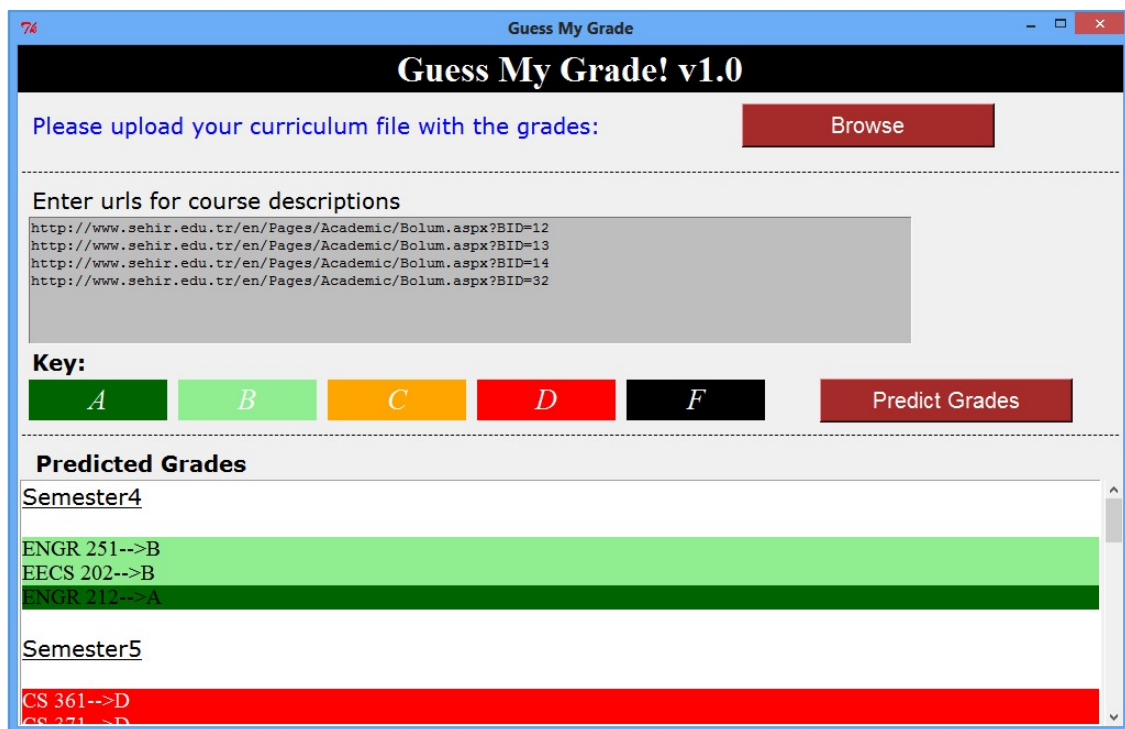


Figure 1

- The user will first provide his/her past grades entered into his/her department's curriculum file which is the same excel file that you used for Mini Project 1 extended with an additional grade column. Sample test curriculum files with some grades are provided as part of the project files.
- Besides, the user will provide four different URLs to web pages that list course descriptions for CS, EE, IE, and UNI courses, respectively, in the mentioned order. For testing you may use the following links as input:

<http://www.sehir.edu.tr/en/Pages/Academic/Bolum.aspx?BID=12>
<http://www.sehir.edu.tr/en/Pages/Academic/Bolum.aspx?BID=13>
<http://www.sehir.edu.tr/en/Pages/Academic/Bolum.aspx?BID=14>
<http://www.sehir.edu.tr/en/Pages/Academic/Bolum.aspx?BID=32>

- Then, the user will click on “Predict Grades” button. Your tool will process the uploaded curriculum and the web pages at the provided URLs, and extract the course descriptions. You may need to use Selenium module (please see practice session material for Week 9). Selenium can work with many browsers, but using Firefox web driver is suggested. You may need to have your program sleep for 5 seconds (e.g., `time.sleep(5)`) after clicking on a link through Selenium to allow for the loading of the web pages. Then, you can get the page html source from the web driver, and process it with the BeautifulSoup to extract the required information.

You will consider grade prediction task as a special case of document filtering as follows:

- You will assume that there are 5 classes (based on course grades): A, B, C, D, and F.
 - You will train a Naïve Bayes classifier by using the courses that you have already taken where course description for each course that you have taken will be a training document, and the grade that you got for that course will be its class/label. You will ignore + and – in letter grades. For instance, if a student got B+ or B- in a course, it will be considered as B. Similarly, other grades will be adjusted in the same way.
 - The courses with P, U, and S letter grades will be ignored.
 - You will assume that each class has threshold 1.
- Once you train your classifier, you will predict the grade for each course that you have not taken in your curriculum. Courses should be grouped under semesters according to the curriculum organization as shown in the example run in Figure 1. You may ignore general electives. However, for UNI electives and departmental electives, you need to still include predictions in your output. You may list your predictions for all available UNI and departmental electives at the end of the obligatory courses. See Figure 2 as an example (example run with CS curriculum).

Predicted Grades	Predicted Grades
Semester8	EECS 415-->D
ENGR 498-->A	EECS 416-->B
	EECS 415-->D
Departmental Electives	UNI COURSES
EECS 481-->B	UNI 222-->B
EECS 468-->B	UNI 221-->B
EECS 483-->D	UNI 115-->B
EECS 214-->A	UNI 100-->B

Figure 2

- In your output area, your predictions should be color-coded for easy differentiation based on the predicted grade. You should use the coloring provided in Figure 1. Besides, you should provide the color legend/key on your GUI to let your users know what each color means (please see Figure 1).

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

- You **should** import `docclass` module and use the methods/functions from there appropriately.
- You may use Text widget for the the result area. You may want to use tag feature of Text widget for color and font configuration of particular regions in the results.

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. **Do not** work together if you are not in the same group. 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 for best learning experience). 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 in your submission). Name your code file 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.
 - Those who **do not** follow the above naming conventions **will get -5 off** of their grade.
- Submit it online on LMS (Go to the Assignments Tab) by **17:00 on Monday, May 23, 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 (%3)	Classes and objects used (%4)	Sufficient commenting (%4)	Compiles? (20)	GUI Design (10)	Retrieving course descriptions (15)	Parsing curriculum and existing grades (15)	Predicting grades for courses (15)	Showing predictions properly (15)

- Interview evaluation
 - Your grade from interview will be between 0 and 1, and it will be used as a coefficient to compute your final grade. For instance, if your initial grade was 80 before the interview, and your interview grade is 0.5, then your final grade will be $80 \times 0.5 = 40$. Not showing up for the interview appointment will **result in** grade 0.

Have further questions? :

- Please contact your TAs (Jareth or Dogukan are focusing on projects. You may want to talk to them first, but you may talk to Ali and Bekir as well) 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. **Do not walk in randomly (especially on the last day) into your TAs’ or the instructor’s offices. Make an appointment first. This is important. Your TAs have other responsibilities. Please respect their personal schedules!**