

ENGR 102 Programming Practice

Mini Project 5

May 30, 2018

(Due on June 4, 2018)

Course code and course name reflects the corresponding programs content. In this mini project, you are going to build an Estimator tool to predict which program do courses correspond to. You will use different analysis to find most related programs using all the words that appear on the course names as features. At the end, the user will be able to also see the analysis for each program and the correct and wrong course classifications. Your program will have a GUI as the one shown in Figure 1.

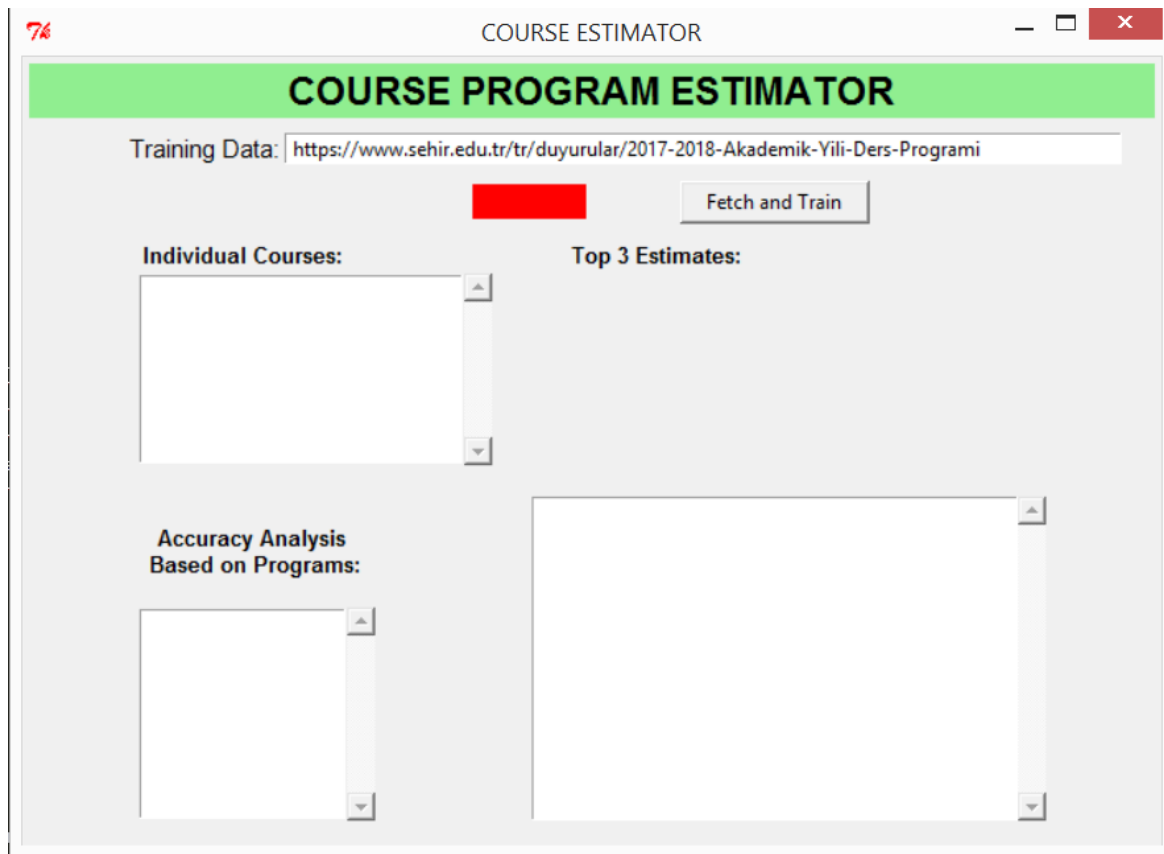


Figure 1. GUI of your program for Mini Project 5

1. Fetching Data:

First, the user will enter SEHIR's programs URL into the corresponding entry widget at the top. By default this widget should have the following URL when the app is run: <https://www.sehir.edu.tr/tr/duyurular/2017-2018-Akademik-Yili-Ders-Programi>.

When the user clicks on "Fetch and Train", from this link, using urllib2 and BeautifulSoup, your program will fetch all course names and their corresponding course codes. The first part of the course code (i.e., ENGR for ENGR 102) will be used as the program code that the course belongs to. The program should fetch all the course names where each word is going to be a feature. At the end all the courses should be fetched (i.e. ADP 102 01, ENGR 102,...) and a list of all the programs should be created and the corresponding Listboxes should be populated as seen on Figure 2.

Before starting the fetching process, the label next to the “Fetch and Train” button will be red. Once “Fetch and Train” button is clicked, your program should start fetching and change the color of the label as yellow. At the end of the fetching phase, the label should be colored green.

When data fetching process is completed, your program will have a GUI as shown in Figure 2.

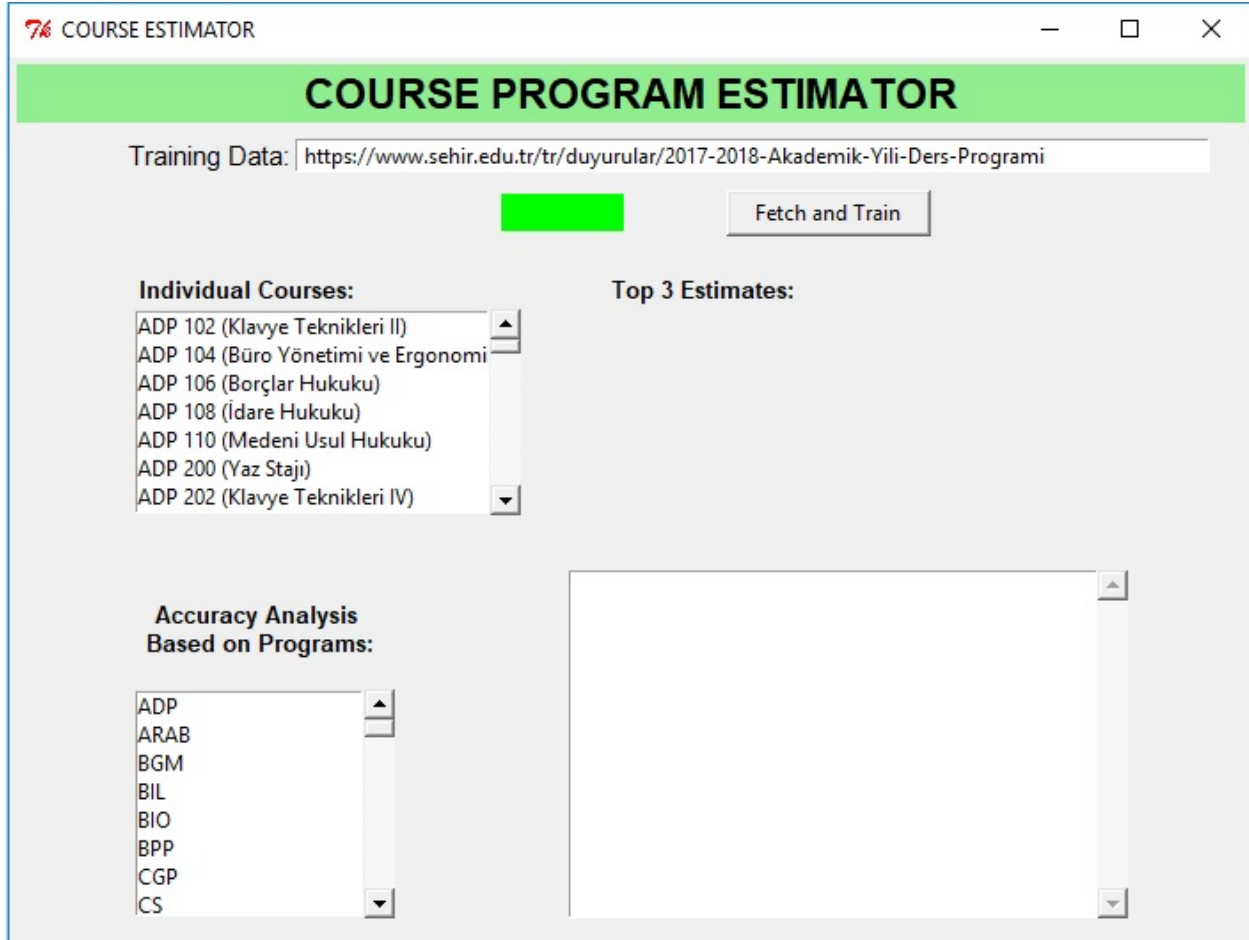


Figure 2. GUI after fetching process completed

Analysis Part:

The program will allow two types of analysis:

- 1. The user will see top three predictions for any course offered by selecting from the list.**

When the user clicks one of the courses from the Individual Courses ListBox, on the right side under Top 3 Estimates Label the top three prediction (predicted programs) for the selected course should be shown.

When listing the top three programs for the selected course if the prediction is correct the background of the label should be highlighted in green and red otherwise(if the prediction is not correct).

In **Figure 3** an example is shown when selecting ADP 106 course top three prediction appear on the right, the highest being HUK which is highlighted in red since it is not correct and the second prediction ADP which is the correct prediction hence highlighted in green.

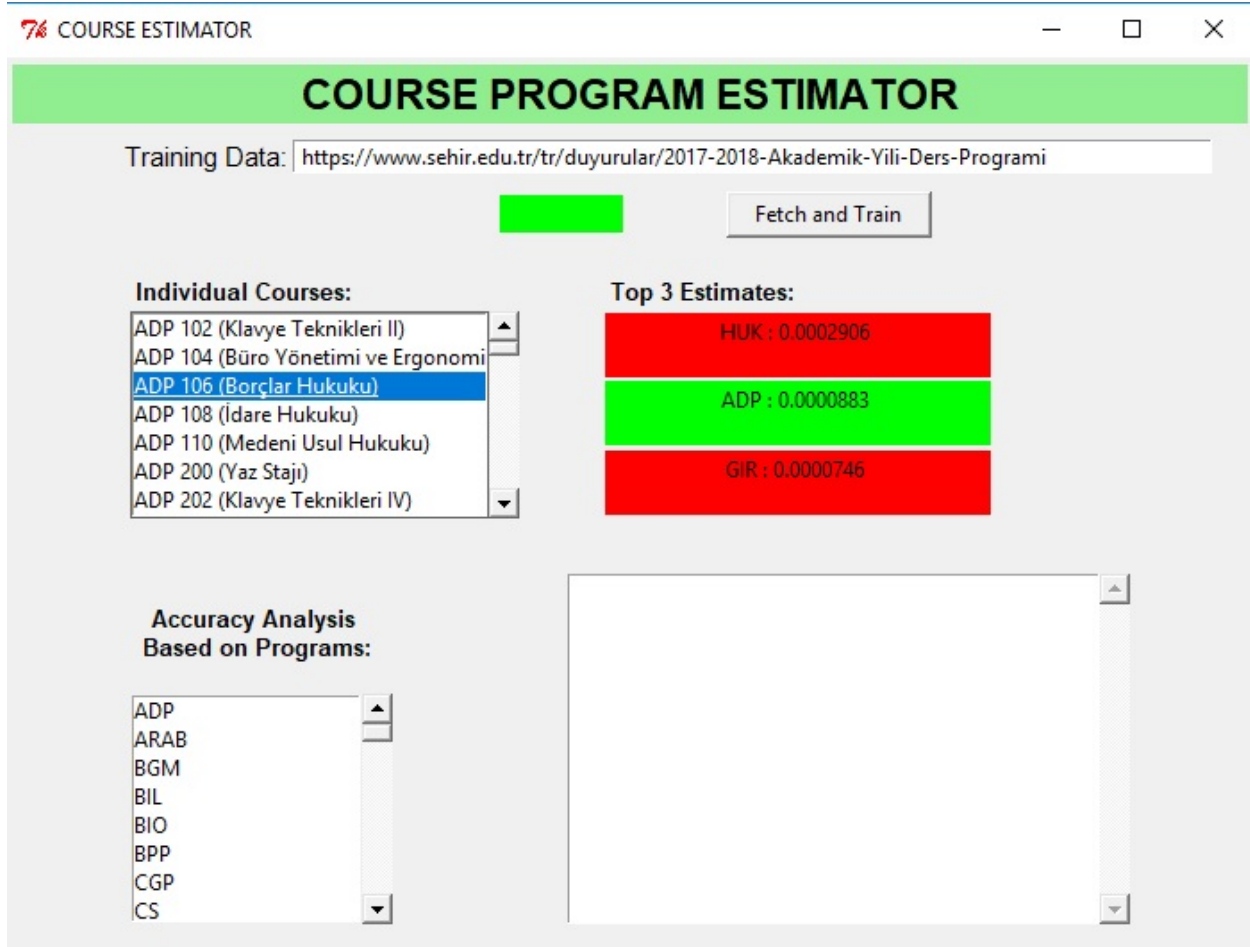


Figure 3. Predictions for an individual course

2. User may perform an accuracy analysis per program

When the user selects one of the programs from the Accuracy Analysis Based on Programs List Box, the Textbox on the right will show the following information regarding the selected program:

Accuracy: (in percentage)

Total Number of Courses: (the number of courses on this program)

Accurately Classified: (the number of the courses that are in this program that their predicted program was correct as well)

Inaccurate Classification: (the number of courses that are in this program but were predicted to be on a different program)

In **Figure 4** an example of Accuracy Analysis based on Programs is shown. When HUK program is selected the above described information will be displayed on the Textbox for the program. HUK program has 37 accurate classifications, and 6 inaccurate one of them is HUK 202 which as seen from Figure 3 is being wrongly predicted to be an ADP program course.

The list of the inaccurate classifications will have the following information:

Course Code --> Top predicted program (i.e. HUK 202 -->ADP)

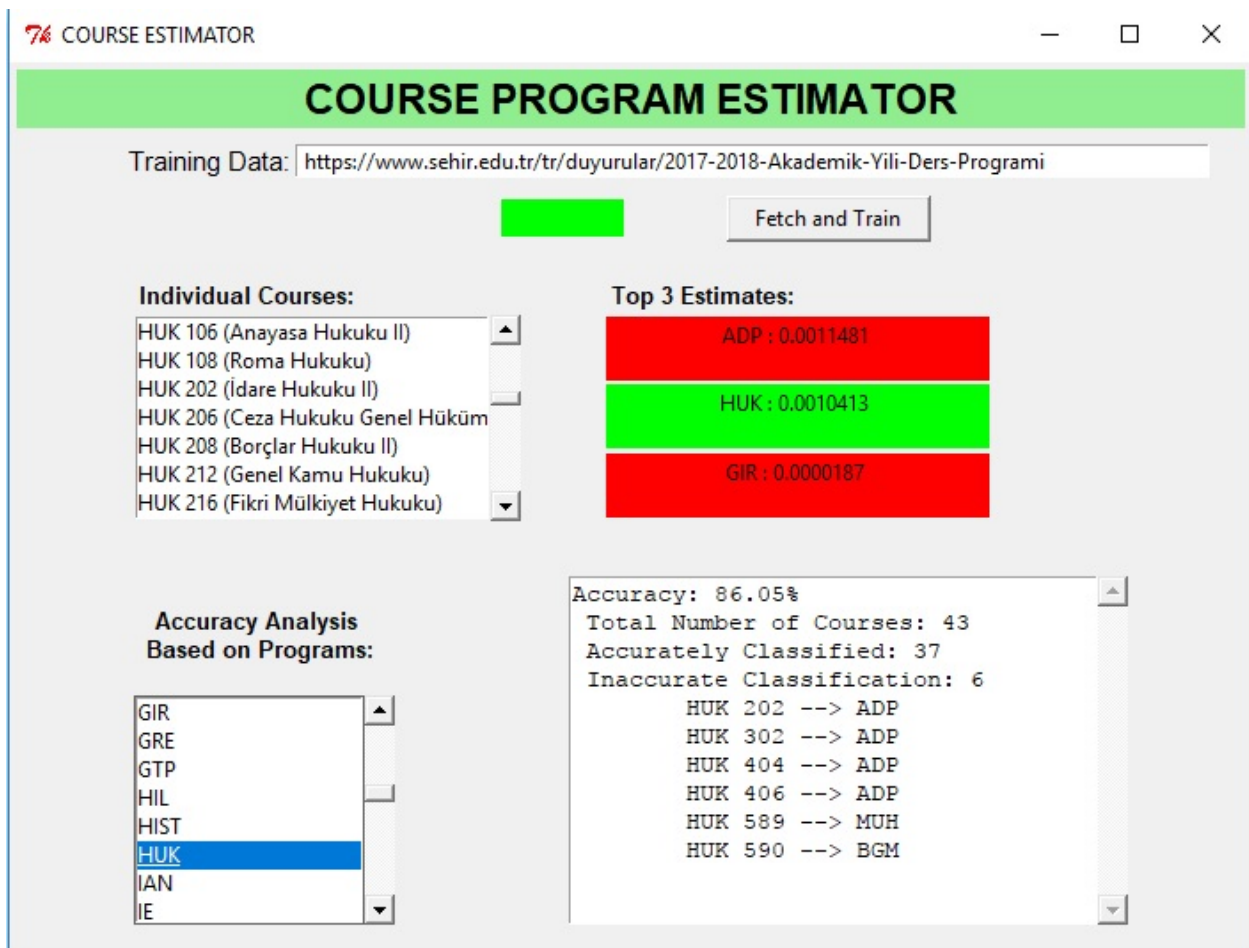


Figure 4. Accuracy Analysis based on Programs

Further Pointers:

- You may call `self.update()` in the interface class at proper points to make sure that color changes on the fetching status label are visible on the interface.

Warnings:

- You **CANNOT** use place for geometry, only grid and pack are allowed.
- 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. Last but not the least, you need to understand code pieces that you may get some other resources. This is one of the goals of the mini projects.
- 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.

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 **strongly** 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 file naming details).
- Submit your own code in a single Python file. Name it with your and your partner’s first and last names. As an example, 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** 10% **off** of their project grade.
- Submit it online on LMS by 17:00 on May 22, 2018.

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 after which are up-to 24 hours late.

- -50%: Submissions which are up-to 48 hours late.
- Submission more than 48 hours late will not be accepted.

Grading Criteria?

GUI Design (20)	Fetching Data (30)	Program prediction for individual courses (20)	Accuracy analysis for each program (30)
----------------------------	-------------------------------	---	--

From your overall grade, we will deduct points by the specified percentage for the following items:

- Inappropriate/cryptic variable names (-10%)
- Classes and objects are not used properly (-30%)
- Insufficient commenting (-10%).
- Inappropriate file naming (-10%)

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.**