

ENGR 102 Programming Practice

Mini Project 5

(Posted on May 06, 2019, Due on May 20, 2019)

Last Update: May 06, 17:00

The quality of an academic institution can be evaluated according to the research projects developed by the academicians and the funds received by these projects. The Department of Computer Science at Istanbul Sehir University is home to many research projects developed by academicians and funded by authorized institutions. In this project, you will build a tool that automatically predicts the principal investigators of research projects which are carried out by Sehir CS Faculty Members.

How should it look like?

The initial opening graphical user interface of your tool should look like as the following one.

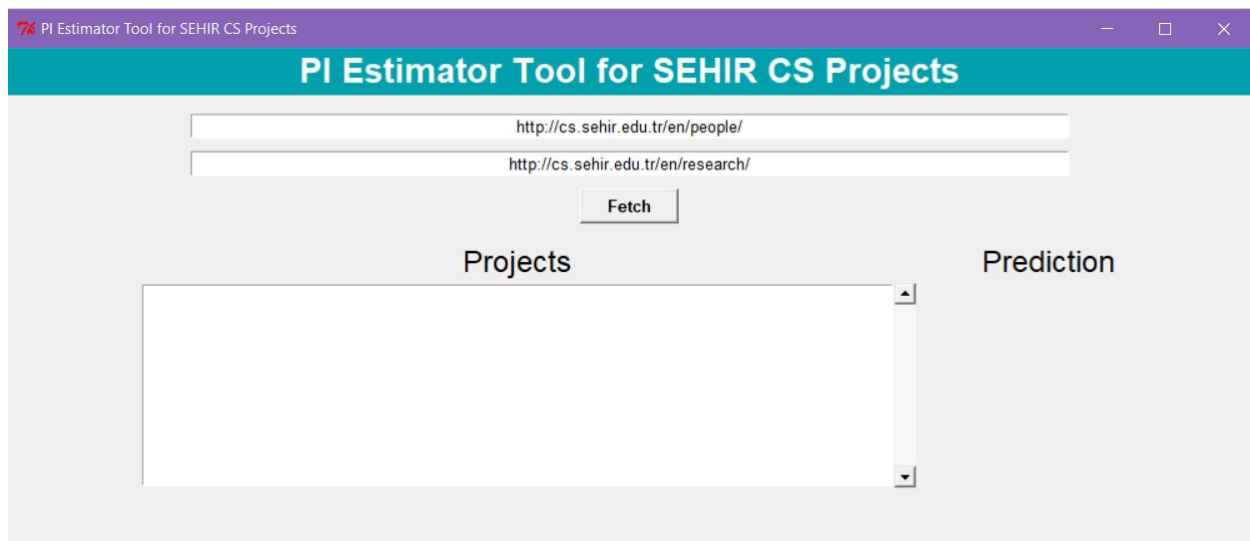


Figure1

How should it work?

1- Fetching Data:

Firstly, the user will provide two URLs in the corresponding entry widgets at the top. The first URL lists the Sehir CS faculty members, and the second URL lists the research project details. By default, these widgets should have the following URLs when the app is launched.

<http://cs.sehir.edu.tr/en/people/>

<http://cs.sehir.edu.tr/en/research/>

When the user clicks on “Fetch” button, using urllib2 and BeautifulSoup (or even Selenium if you want to), the program will fetch the following information.

1. From the first link, your program should collect the links to the profile pages of CS faculty members. Then, your program should go to the profile of each faculty member, and collect the faculty member's first and last name (ignore middle names, if any), and the publication information. Each publication would be considered as a separate document. For instance, Prof. Ahmet Bulut has 24 listed publications which will be considered as 24 documents with category name 'Ahmet Bulut'. These 24 documents will be part of your training data. The first training document for category 'Ahmet Bulut' will include the following:

[Ammar Rashed, Abdullah Ihsan Seer, Abdurrahman Aboudakika, Ahmet Bulut: Orienting Social Event Streams as Data Stories. MIKE 2018: 138-147.](#)

2. From the second link, your program should fetch the research projects detail. You need to fetch the title, principal investigator, and summary for each research project.

Note: Please ignore the projects whose principal investigator is not included in the first URL, as these projects were carried out by ex-faculty members.

When data fetching is completed, the listbox should be populated with the fetched research project titles, sorted in alphabetical order. Your program should create a Naive Bayes classifier object, and train it with the training data that you build out of publications of CS faculty members. As a remark, you should consider each publication info in a profile page as a training document and the name of the corresponding faculty member as its category. In total, there are 6 faculty members. Therefore, there will be 6 categories. Train your classifier with these training documents and their categories. When the fetching process is completed, your program should have the following GUI.

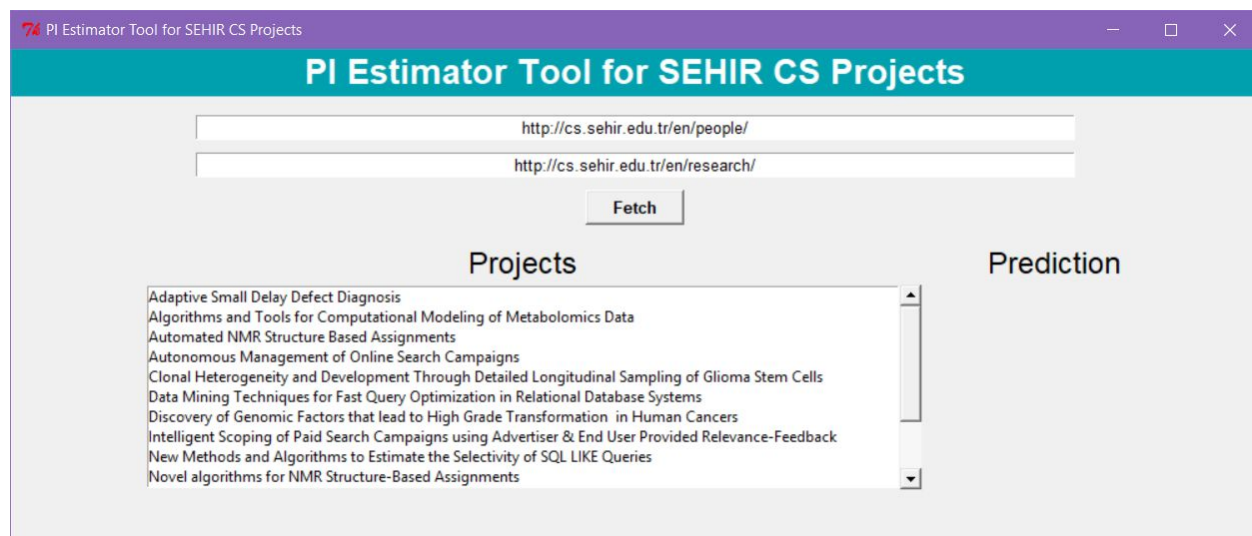


Figure2

2- Predicting Principal Investigator for Research Project:

When the user selects a research project from the project list, your program will employ the created classifier object to predict the principal investigator (PI) for the selected research project. To this end, you will combine the title and summary of the selected project into a document, and use your already trained classifier to predict the category (i.e., PI name) of this document. The predicted PI name will be shown in a label on the right. If the prediction is correct, the PI name label will have a green background (Figure 3). Otherwise, the background color of the PI label would be red (Figure 4). To verify whether the prediction is correct or wrong, you will need the PI name that you extracted from the project details page.

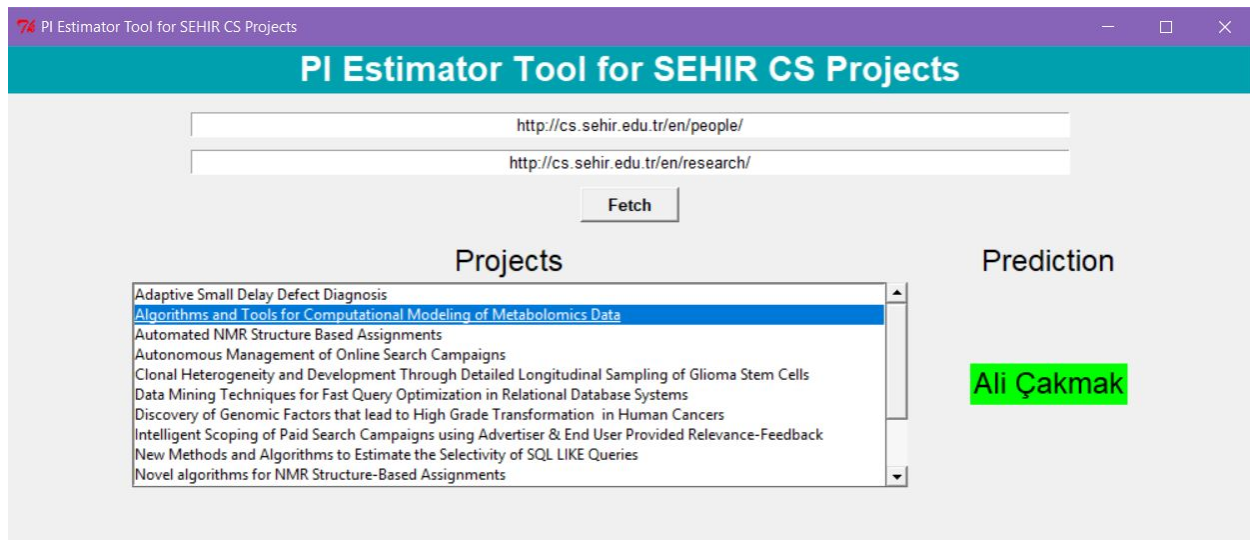


Figure3

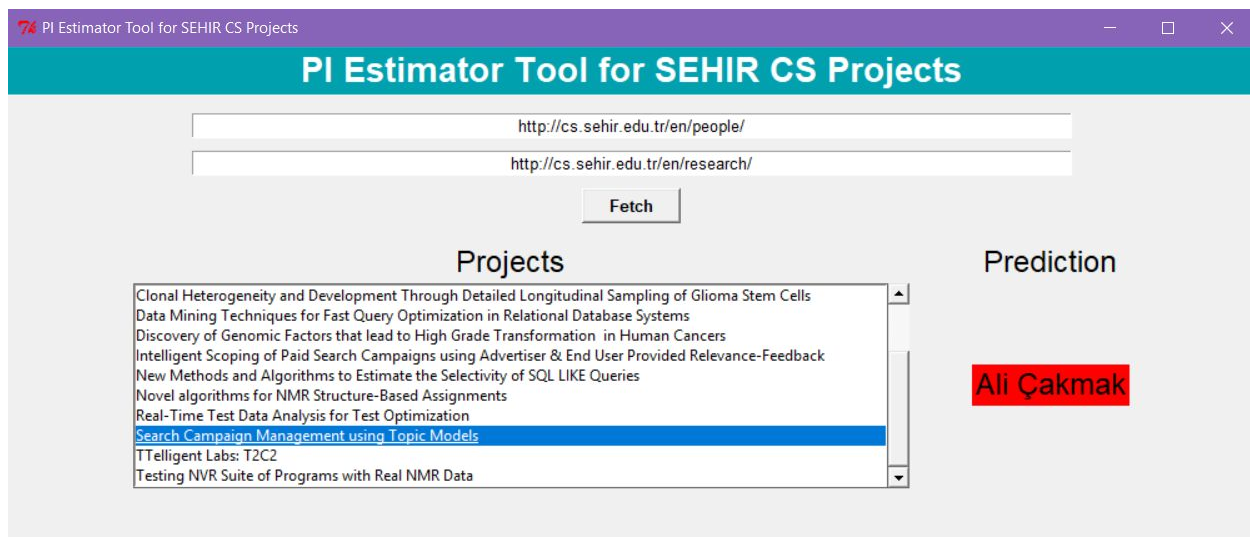


Figure4

Implementation Notes:

- As part of this project, you are provided with docclass.py module for creating and using a classifier. Please download the copy that is attached to MP5 on LMS.
- You need to have at least the following classes.
 - A class that represents a FacultyMember with the following attributes: name [first and last name] (string), profile_url (string), publications (list of strings).
 - A class that represents a ResearchProject with the following attributes: title (string), summary (string), PI name (string).
 - A Predictor class with the following attributes: classifier (Naive Bayes object), faculty_members (dict - key: faculty member name, value: the corresponding faculty member object), projects (dict - key: project title, value: the corresponding research project object). It will have the following methods:
 - fetch_members
 - fetch_publications
 - fetch_projects
 - train_classifier
 - predict_PI
 - A class that represents GUI which will an object of the Predictor class as one of its attributes.

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 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 the best learning experience). If you are doing it as a group,

only **one** of the members should submit the project. Filename 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 the 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 20, 2019.

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 profiles properly (20)	Fetching projects properly (20)	Creating and training a classifier properly (20)	Employing the trained classifier properly to predict the PI information and display it (20)
----------------------------	--	--	---	--

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. This is important. Your TAs have other responsibilities. Please respect their personal schedules.**