**ENGR 102 Programming Practice**

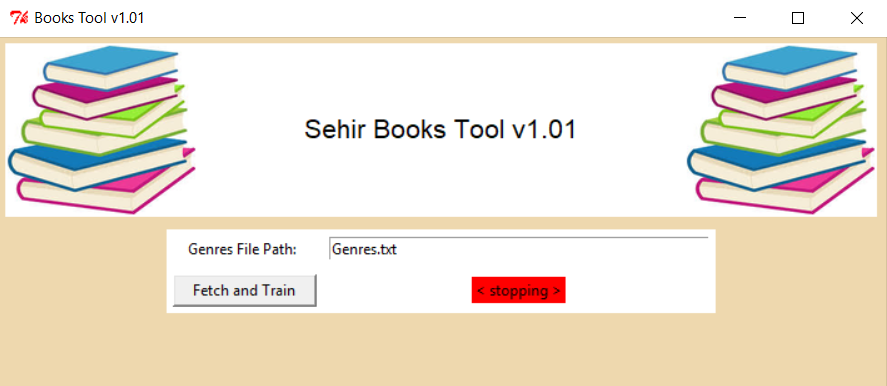
**Mini Project 5**

*Tags: Naive Bayes, document filtering, classification, GUI*

*26th December, 2018*

(***Due*** *on 9th January, 2019*)

In this mini project, you are going to build an Estimator tool to predict which Genre do books correspond to. You will use different analysis approaches to find the most related Genre using all the words from the book names as features. At the end, the user should be able to also see the analysis for each Genre and true/false Genre classifications. Your GUI should initially look like as shown in Figure 1. On both sides of the GUI there will be books picture (“books.jpg”included with the project files).



**Figure 1**. The initial look of the GUI.

**How should it work?**

**Fetching Books Data:**

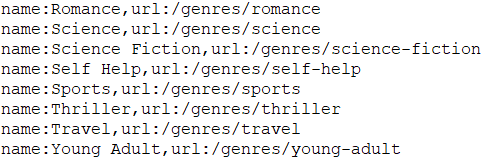
First, the user should provide the path to Genres.txt file (the Entry Widget should have the file name by default when app runs). Genres.txt file will have the names of the Genres and a partial URL. To use the partial URL you need to use this link:

https://www.goodreads.com

For example, if we wanted to access Science Books page, our link should be:

https://www.goodreads.com/genres/science

You can check the example contents of Genres.txt file as shown in Figure 2 below.



**Figure 2**. Genres.txt contents

When you access a specific Genre page, you should collect books only from Popular Books section at the end of the page.

When the user clicks on ***Fetch and Train*** button in the GUI, using urllib2 and BeautifulSoup, your program should fetch all books’ names and create an id for each book. For example, if we went to Science Genre in popular books Section we can see the book "A Brief History of Time". You should create the ID Science1 for this book. For the next book following "A Brief History of Time", an ID of Science2 should be created. All IDs should be created for all books until there is no more books in that section.

The program should fetch all the books’ names since each word is going to be a feature. At the end, all the books should be fetched and a list of all the genres should be created and the corresponding listbox should be populated as shown in Figure 3.

Before starting the fetching process, the label next to the ***Fetch and Train*** button will be red (Figure 2). Once ***Fetch and Train*** button is clicked, your program should start fetching and change the color of the label to yellow and the text to “***Trained***” (Figure 3).

When data fetching process is completed, your GUI will look like as shown in Figure 3.



**Figure 3**. GUI after fetching process completed.

**Analysis Part:**

**PLEASE USE *prob* method in *naive Bayes* classifier in *docclass.py* for your predictions.**

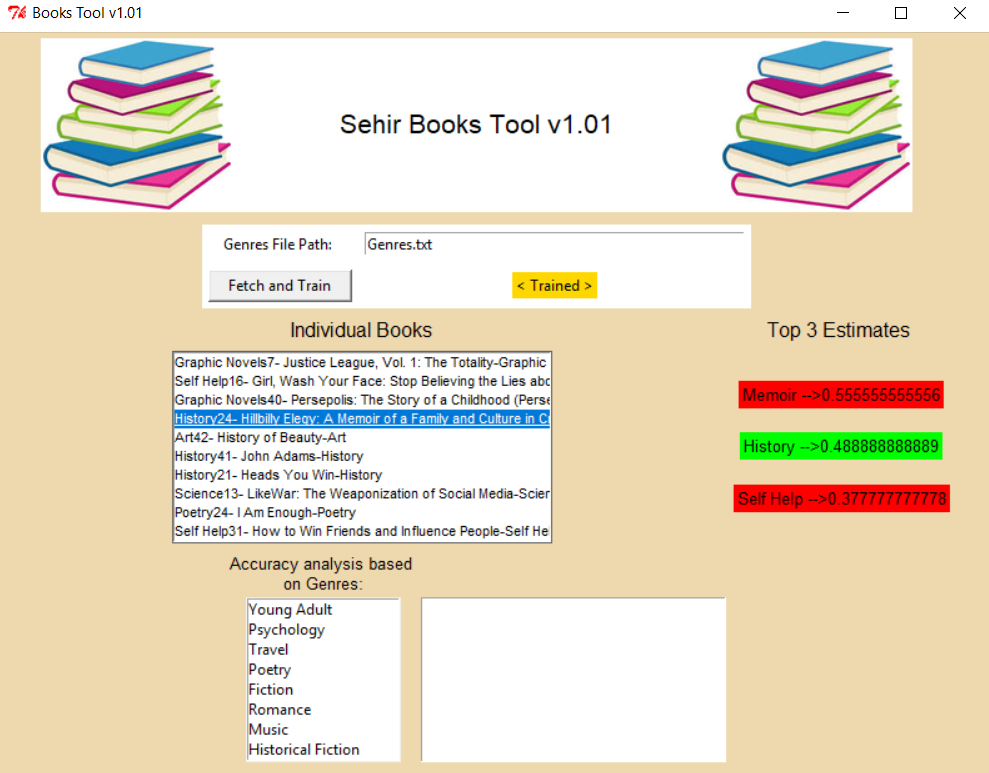
The program will allow two types of analysis:

**The user will see top three predictions for any book offered by selecting from the list:**

When the user clicks one of the Books from the ***Individual Books*** ListBox, the top three predictions (predicted Genre) for the selected Book should show up at the right hand side under ***Top 3 Estimates*** label (Figure 4).

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

An example is shown in Figure 4. When the History24 Book id is selected from the listbox, the top three genre prediction appears on the right. The highest being Memoir is highlighted in red since it is not correct and the second prediction History being the correct prediction is highlighted in green.



**Figure 4**. Genre predictions for the books selected from ***Individual Books*** listbox.

**User may perform an accuracy analysis per Genre classification:**

When the user selects one of the Genre from the ***Accuracy Analysis Based on Genres*** listBox, the textbox on the right will show the following information regarding the selection:

**Genre Name:** Genre that has been clicked from the listbox

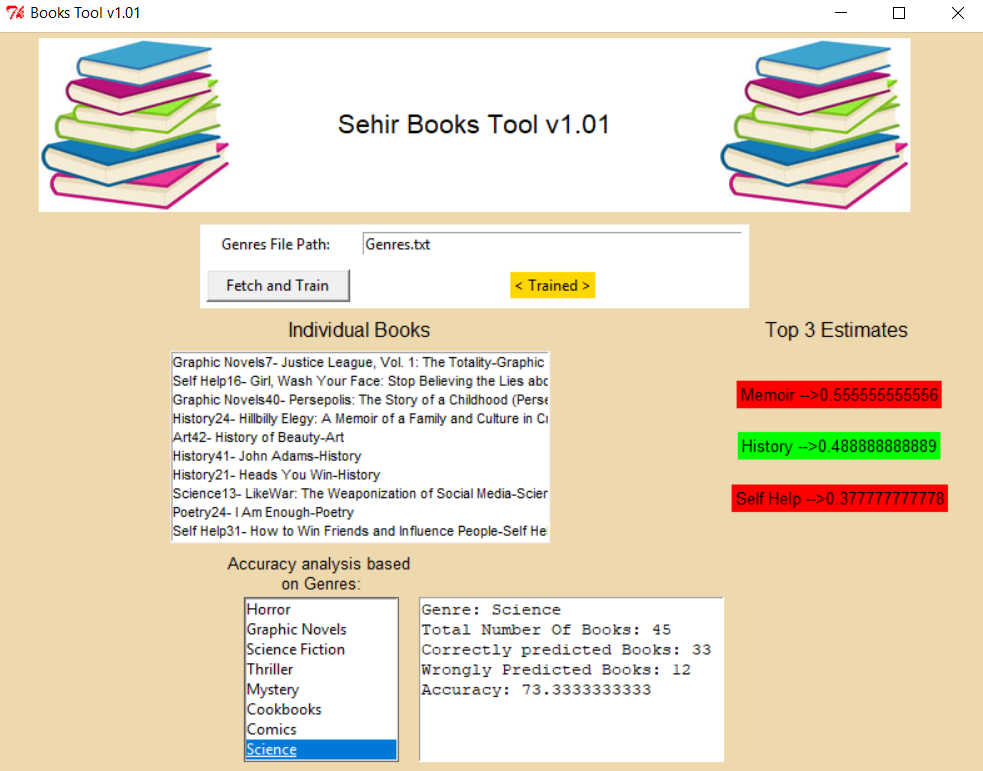
**Total Number of Books**: The number of books in this Genre

**Accurately Classified**: The number of the books that are in this Genre and their predicted Genre was true.

**Inaccurate Classification**: The number of books that are in this Genre but were predicted false.

**Accuracy**: True classifications in that Genre divided by total number of books in that Genre, calculated as a percentage.

In **Figure 5,** an example of Accuracy Analysis based on Genre is shown. When Science Genre is selected, the information, as explained above, will be displayed in the textbox. Science program has 33 accurate and 12 inaccurate classifications and so the accuracy calculated as 73.33% (33\*100/45).



**Figure 5**. Accuracy Analysis based on Genres.

**Further Pointers:**

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

**Warnings:**

You **CANNOT** use place for geometry, only grid and packare 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 **January 09, 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 Data**  **(30)** | **Genre prediction for individual Books**  **(20)** | **Accuracy analysis for each Genre**  **(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.**

**IMPORTANT NOTES:**

**Note 1: Plagiarism**:

* + Zero tolerance
  + Cases will be referred to the Ethics Committee
  + Both parties (provider and receiver) are responsible
  + Process:
    - Automated computerized checks for pre-filtering
    - Human review for confirmation
    - Referral to the Ethics Committee if true positive