February Report on Course Recommendation System.

Fuxing Luan, Fuzail Misarwala, Zubin Thampi.

Approach 1: Peer to Peer (Piazza)

The main purpose of this approach was to get new incoming students in touch with their seniors at the university for direct questions and answers for specific doubts they would have about certain courses. We use Piazza as a forum where students can post questions and other, experienced students can answer them based on their knowledge.

We created multiple sections for different courses on Piazza where questions can be posted in the forms of notes or polls, and discussions can be started by anyone.

Some of our primary reasons for using Piazza are:

- It has a good, easy to use user interface.
- Piazza provides an API for text mining algorithms which helps us integrate with our other approach well.
- The content posted on Piazza is persistent and can be accessed by anyone at anytime, even if they join the thread later on, as opposed to a chat mechanism (such as Slack).
- Repetitions of posts can be avoided thanks to persistence of threads.
- Piazza provides a search feature which is very convenient to look up certain issues, and also avoid duplication or repetition of questions.
- Threads on Piazza also incorporate a polling feature which can be greatly useful to decide some things like professors or sections within a course that a student is confused about.

So far, we have created about six different sections on our Piazza forum for six courses, and have added sample threads on it to work with. The primary purpose of this approach is to get direct feedback from students who have already taken the courses that a new student wants to enroll for. This is done in order to answer specific questions they may have that cannot be answered through MyPack or the NCSU website.

Our future work would involve creating more sections for more courses (up until we can add sections for every course being offered in a certain semester), and get as many participants as we can to be able to answer questions. It would be essential to have people from various backgrounds, working towards a variety of specializations as that would add a lot of perspectives and provide the benefit of being able to get answers to questions based on more and more specific scenarios.

We also aim to add more reviews from past students. This would help us gather more data on the text mining system of course recommendation. We also plan to populate data into the section for more general questions which aren’t specific to a certain course.

Approach 2: Web Slurping

This approach involves the consolidation of information essential for course selection, from various sources, into one convenient location. When students are looking to register in a course, they are usually unaware of key information that would help them make their decision. Our idea was to not only provide access to that information, but also make it easy for them by doing so in a single spot.

The first task we accomplished was the collection of the aggregate Grade Distribution data available to all currently enrolled students from MyPack portal. We made use of a script written in Java to convert the information from an html file format to a database.

Secondly, we collected course description information from the NCSU website. A list of all courses and their descriptions were created through the use of a Python script to collect the information. A Python module named Beautiful Soup was used to parse HTML, and collect data.

All of this collected information was loaded on to the database made in SQLite.

We designed and constructed a website to present all this information in a user-friendly manner. The website was built using Ruby on Rails, which is a server side web application framework written in Ruby. The website consists of the following key components.

- Home page that holds all the links to the information pages.
- List of courses.
- Overall grade distribution information, highlighting some statistics such as %As, %Bs, Total number of enrolled students, frequency of offered courses etc.
- The website incorporates a page dedictated to integrating the text mining solution.
- Pages for specific courses providing a more detailed overview of the course including individual grade distribution data, detailed course descriptions, and syllabus information. So far we have detailed information on six of the courses being offered.

Our future work involves:

- Adding more subjects to the database, and to the website, to build a more comprehensive system.
- Getting content from Piazza to display on the website to maximize the amount of information available in one location.
- We are striving to improve the user interface to make it look more appealing and quick, so as to be more user friendly.
- We are attempting to find a way to automate the system of acquiring new and updated information from our sources and add them to the website directly.
- Adding information about professors teaching a certain course by adding information from ratemyprofessor.com

This solution simply aims at providing more insight to course information. There are certain things available to new students but they aren’t even aware of, and the website helps solve that.

Approach 3 : Text Mining

The idea in this approach is to get user feedback from the Piazza forum, and try to score the courses (GOOD vs BAD) on a scale from 0 to 1, on 5 categories – Professor Rating, Grades, Content, Job Prospectives, and Workload. Once we get the requirements from the students on the 5 categories, we can recommend the courses that match closest with their requirements.

We did the following for the approach:

- Pilot project (pilot.js) to check if we can get content from Piazza using an API provided. We had two options – a Python API or NodeJS API. Python was our first preference, but due to incompatibility, we went with NodeJS instead.
- Coded a JavaScript file (get_data_from_piazza.js) to pull data from Piazza's forums, and call the Python file for text mining.
- Text mining was done using a python file called text_mine.py, in the following steps.
    1. Converting the data from documents to vectors (Using Python NLP module doc2vec).
    2. For each course, score five categories on a scale of 0 to 1 on the basis of GOOD vs BAD words.
    3. For each category, we identified a list of relevant words.
    4. We retrieve a list of the related words to above using the doc2vec model.
    5. Calculate a ratio of good words to the total number of words accumulated.

$$Ratio = \frac{n_{good}}{n_{good} + n_{bad}}$$

    The value of the above ratio is between 0 and 1.
    6. Students choose their requirements for each category. (Again on a scale between 0 and 1)
    7. We calculate the value of the squared error (SE) for each course available to us.

$$SE = \sum((x)^2 - (y)^2)$$

$$where\ x = Score\ for\ each\ category\ from\ user\ reviews$$
$$y = Score\ for\ each\ category\ required\ by\ student$$

    8. The courses returned are sorted in ascending order of their value of squared error. This gives the order of courses that matches closest with the student requirements.

We hit some roadblocks with this approach:

- Firstly, we did not have enough data to test this feature on. The sample set that we had was smaller than we would have liked it to be.
- Secondly, the criteria used to measure the error for courses, that is GOOD v BAD words, is not exceptional and may not provide results which are 100% reliable.
- For Course Content, if would be better to rate on an EASY vs HARD scale, as opposed to GOOD vs BAD. But we are not sure how to do that.

We are striving to get rid of these problems by attempting to accomplish the following tasks in the near future:

- Find out ways to rate on EASY vs HARD scale in addition to GOOD v BAD.
- Attempting more regression based models when we have additional data to work with.
- Incorporate data from General Reviews on the forum.

- Adding data and reviews on specific professors which would help a student streamline his search better.
- We also would like to study more regression models in an attempt to find a more suitable one for this project.

Resources made available	**Peer to Peer**	**Web Slurping**	**Text Mining**
Reviews from seniors	Yes	No	No
Grade Distribution	No	Yes	No
Course Content	Yes	Yes	No
Information about Inclusion of Projects, Job Prospects, Workload	Yes	No	No
Frequency of offered course	No	Yes	No
Direct Recommendation	No	No	Yes