# Storyboard: Course Recommendation System for Duke Registrar

**Authors:** Vanessa Tang, Sangseok Lee, Yiran Chen

## Epic 1

The purpose of this project is to build a course recommendation system for Duke undergraduate students that will help students choose which classes to take because the current course selection process is actually quite complex, and students often spend many hours trying to figure out which courses to take. Students take classes for a variety of reasons, including major or minor requirements, graduation requirements, timing, location, professors, or peer suggestion to name just a few contributing factors. For this reason, we aim to build a course recommendation system to facilitate this process. By recommending courses based on student information, the system can help with course planning, order, and selection.

### Theme 1.1

We aim to focus on building a recommender system based primarily on three things:

1. The student's planned major
2. The course history from other students within the same major
3. The idea of an academic pathway. Though this can be defined in many ways, one intuitive way is tracking the most popular courses for each term within a specific major during a student's time at Duke.

**Story 1.1.1**

We build a dashboard to show the most popular courses students of a specific major took in a specific semester. For example, we can see the 5 most popular courses for Econ majors to take their Freshman Fall term. Based on this dashboard, one might recommend taking Economic Principles Freshman Fall in addition to the required academic writing course.

**Story 1.1.2**

We build a similar dashboard but tracking courses through the entire 4 years at Duke. This dashboard shows the top 5 most popular courses for Econ majors to take during all 8 semesters. This can help students plan their courses throughout all 4 years, which can help with course load planning and ensuring major requirements are met.

### Theme 1.2

Develop a recommendation model based on past class completion, enrollment data, and findings from Theme 1.1. This may be in the form of a more traditional recommendation engine that enables students to compare pathways within a majors.

**Story 1.2.1**

We build a recommendation system using three models with three subsets of data. The first and smaller subset consists of students who enrolled at Duke in 2005 and graduated in 2009 with a BS in economics. The second dataset consists of students who enrolled in 2005 and graduated in 2009 with any of the top 5 majors. The third dataset consists of all students who majored in Engineering. Using these three datasets, train and test sets are split by year. For example, if we want to recommend courses for Senior year, we train on the first three years of classes. There exist slight variations for the train-test split methods used in each of the following model.

**Story 1.2.2 (Sangseok Lee)**

The first model is a collaborative filtering model using the neighborhood method defined by cosine similarity. We define accuracy as the number of courses actually taken in the top 10 recommendations divided by the total number of courses taken during the fourth year. We can define the potential pathways the students are taking beside their major. These pathways could involve all different kinds of interest.

**Story 1.2.3 (Vanessa Tang)**

The second model is a matrix factorization model model using Singular Value Decomposition (SVD), where students' grades can be predicted by multiplying $U$ (the student grade matrix), $\sigma$ (the diagonal matrix of weights), and $V^T$ (all the courses).

**Story 1.2.4 (Yiran Chen)**

The third model is LightFM, which is one of the hybrid matrix factorization model that is able to integrate three pieces of information: the information on the student, the information on the course, and the interaction information of which students took which courses.

In the case of Light FM, we get the latent representation of each feature for every item and user. Following this idea, the latent representation of a item is just the sum of the latent representations of the item’s features. Similarly for users, we just add the latent representations of the user’s features to get the latent representation for a user. The score for a item-user pair is again the cosine similarity of the latent representations of the item and the user.

**Story 1.2.5**

In order to assess the model performance, we use a form of accuracy. We calculate this by counting the number of courses actually taken in the top k recommendations divided by the total number of courses taken during the year of interest. For example, if a student took 4 of the recommended 15 courses and took a total of 8 courses that year, then the accuracy would be 4/8 or 50%.