

UC Berkeley is very fortunate to offer over 150 different undergraduate majors and minors, which provides its students with the opportunity to experience a multitude of different fields of study. With that being said, there are almost too many options to choose from. In fact, UC Berkeley offered almost 7,000 different classes in just the fall 2020 semester alone. This makes it not only impossible to know all the classes offered but also difficult to decide. Not to mention, class selection can be especially challenging for new majors like data science, that do not yet have a set path. This often leads to an extensive amount of time spent switching between sites such as the berkeley class catalog, berkeleytime.com for grade distributions, and ratemyprofessor.com for professor ratings.

However, a class recommendation system such as Cal Class Recommender, or CCR for short, can make the class selection process much easier. CCR provides data science related class recommendations based on classes that were enjoyed by other data science students while considering the natural order of classes. With the use of CCR, class selection is now quicker and easier than ever which is especially important when self-navigating during these times of virtual learning.

CCR utilizes content-based and collaborative filtering. The recommendation engine takes in a class that a student is currently taking as its input, analyzes the contents (such as the title of the class). Then, it figures out which other users have taken similar classes. It will then rank similar students according to their similarity scores and recommend the most relevant classes to the student. For example, if the system detects that user A is the most similar to user B, then if user A has taken a class that user B has not, the class will get recommended to user B and vice-versa.

The features that are utilized by the model for CCR are the classes students' took last semester, classes they are taking this semester, and classes they recommend. Therefore, we transformed the list of classes into columns, which are inputted into a function for combining the values of these columns into a single string, called *combine_features*. Next, the similarity between two users was found through the cosine similarity score of their coursework. This was done by representing the class titles as vectors and by importing CountVectorizer from sci-kit learn's feature extraction text package. Ultimately, we fed the combined string into CountVectorizer and called *cosine_similarity* on the matrix to get similarities among users. Our next step was to get the title of the class the user recommends. After obtaining the title, we found the index of the class and accessed the row corresponding to this class in the similarity matrix. Thus, we get the similarity scores of all the other classes from the current one. Afterwards, we enumerated through all the similarity scores to make a tuple of class index and similarity score. Next, we sorted the list of similar classes according to similarity scores in descending order and output the first entry at the top of the list. This complete process is done in the function: *ccr_recommendation*, which requires the user's classes as input in array form. It also defaults the user's major to data science, but in the future we hope to expand CCR to all majors.