A College Major Recommendation System

ABSTRACT

College students are required to select a major to study in depth, but are typically provided with little advice and few resources to assist with this decision. A poor decision is detrimental to the student, since it may result in a subsequent switch to a different major and a delay in graduation, or may cause them to perform poorly and become discouraged and leave the university. A poor choice of major can also impact the university, since the time to graduation and retention rates are key metrics used to evaluate the quality of a university education. There is a general lack of research on recommender systems for college majors, and the most relevant recommender systems focus on course-level recommendations. This study describes and evaluates a recommender system that uses collaborative filtering for selecting an undergraduate major, utilizing nine years of historical student data from a large university. The system bases its recommendations on the courses that the student chose to take in their first few years of college, and how well they performed in those courses. The system is designed to recommend majors that the student is likely to be interested in and will perform well in. Recommendations are evaluated based on the likelihood that the student's actual major was in the top five recommended majors, and whether the student performed above average in that major. The recommendation system dramatically outperforms the baseline strategy of randomly selecting a major, and when the recommendation is followed, the student is 12% more likely to perform above average in that major.

KEYWORDS

Recommender systems, collaborative filtering, nearest neighbor, educational data mining

CCS Concepts

• Information systems→Information systems applications→Data mining→Collaborative filtering
• Information systems→Information systems applications→Decision support systems→Data analytics • Applied computing→Education

1 Introduction

Early in their college career a student must make the life-changing decision of choosing a major. This decision is normally made with minimal information and guidance—perhaps just some general idea of the career paths associated with the major. Many universities do relatively little to assist with this decision, and certainly little relevant hard data is provided. Students may infer some basic strategies—such as if they are good at math and science they should consider a STEM (Science, Technology, Engineering, and Math) discipline—but that still leaves open many choices and does not leverage the historical data that the university maintains. This paper describes a recommender system that utilizes this past historical data to recommend a major that the student is likely to be interested in and perform well in.

Recommender systems are a well-researched and routinely deployed in e-commerce settings and have greatly benefited companies such as Amazon for product recommendations [1]. Recommender systems use collaborative filtering and/or content-based filtering combined with historical data, to make their recommendations. These methods can equally be applied to academic recommendations, including the choice of an academic major. However, there are practical differences, due to the differences between the application domains. E-commerce recommendation systems have access to far more recommendation history, since most customers have selected (i.e., purchased, rented or downloaded) many products, whereas in our domain the vast majority of students will select only a single major. Additionally, a bad decision for an e-commerce site generally has less severe consequences than a poor choice for a student major. A poor choice for a major can lead to a delay in graduation, with an enormous financial impact. Finally, the time scales are very different for these two cases, because a poor choice for a major might not become evident for six months or a year.

This paper describes and evaluates a recommendation system that utilizes collaborative filtering to recommend a set of majors to undergraduate college students. The system bases its recommendations on the courses that the student chose to take in their first few years of college, and how well they performed in those courses. The system is designed to recommend majors in which the student will share like-minded peers and in which the student will perform well. Recommendations are evaluated based on the likelihood that the student's actual major was in the top five recommended majors, and whether the student performed above average in that major. A recommender system must have a sufficient amount of historical data in order to make meaningful recommendations. The recommendation system in this paper is based on over 18,000 student records collected over a nine year period from an undergraduate college within a large university

This work is noteworthy because it applies recommender system technology to a new and important application domain. The use of such a data-driven approach can greatly benefit students by guiding them in their choice of a major. The results in this paper also show that this recommendation system can lead to substantially improved student performance within their major of study. However, it should be pointed out that choosing a major is a significant life-choice and we certainly do not advocate leaving such an important decision up to an automated system. Rather, we view the recommendation system as a decision support tool, to help the student identify and consider a variety of majors, including some that perhaps they would not consider otherwise.

2 Related Work

Recommender systems have been used within the education domain for a variety of tasks, such as selecting courseware for specific courses [5], choosing specific content to improve the learning process [6], and selecting the programs to which a student should apply in order to maximize their chances of acceptance and funding [14]. However, there is a notable lack of recommender system research for selecting a student major. Perhaps the closest thing to our proposed system is one that recommends specific courses to take based on the desired learning outcomes that the student wants to obtain before graduation [13].

Our proposed recommender system also is an example of educational data mining, an emerging field that has its own society (educationaldatamining.org) and associated international conference and journal. Educational data mining spans a wide variety of educational topics, but the one most relevant to our research concerns predicting student academic performance in a course, since our system is designed to recommend majors (i.e., a group of related courses) in which the student will perform well. This prior work has used a variety of methods to predict student performance, including Bayesian Networks [7], Logistic Regression [8], Neural Networks [11] and Decision Trees [9]. Methods from recommender systems have also been utilized to predict student performance [10].

3 Methodology

This section describes the education data set, the data preparation steps necessary to convert the data into a form suitable for the recommender system, the process used to identify a set of majors to recommend for a student, and the metrics used to evaluate the recommendations.

3.1 The Raw Education Data Set

The raw data contains 473,256 records that each describes the performance of one undergraduate student in one class. The data was obtained from a major university and covers a nine-year period. The raw data contains many fields, but only the following are used in this study:

- Student Id: student identifier (anonymized for privacy)
- Class Name: uniquely identifies the class the student is taking
- College Name: necessary since there are multiple undergraduate and graduate colleges
- Graduation Year
- Student Major (e.g., Computer Science, Psychology)
- Student Grade in Class: grade using standard U.S. 4-point scale, where 0=F and 4.0=A

The initial data set was generated for undergraduate colleges within the university, but only data for the largest undergraduate college was kept and used by the recommender system. This was done to facilitate the evaluation of the system, since colleges have overlapping but non-identical sets of majors; it would be unfair to compare performance when some schools have far fewer choices for majors than others.

The data set was left with 68 valid majors once a few invalid majors were removed. The invalid majors were not offered any more, improperly coded, or associated with a major only offered by another college in the university. The popularity of the majors varies widely and the distribution of students to the majors is displayed in Figure 1. Figure 1 illustrates distribution of student majors, sorted from largest to smallest. Probability of major shown in blue and with the left y-axis, and cumulative probability of first N majors shown in red and with the right y-axis. Only students who start and complete their four year degree within the nine year period of the data set are retained. This leaves 7,187 distinct graduating students. The number of students per degree decays rapidly, with the first 28 majors containing 90% of the total students.

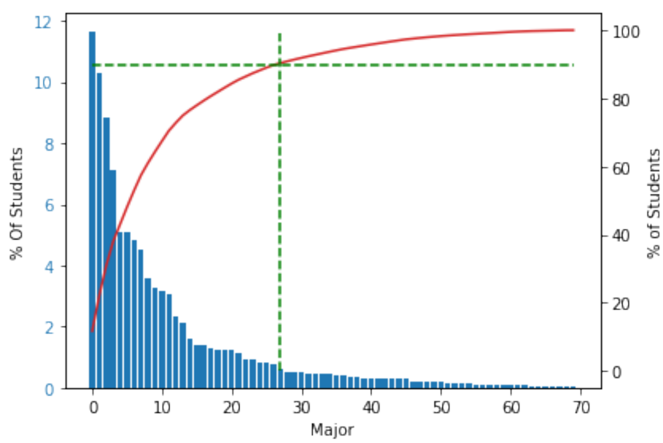


Figure 1 Distribution of student majors

The university has a large core curriculum, which includes diverse topics such as mathematics, social science, and performing arts; students must satisfy the core requirements prior to graduation. We classify all courses as being either "Core" or "Non-Core." These distinctions are important because, as discussed later, we are especially interested in using the core courses to make major recommendations.

All undergraduate courses in the college are numbered such that the first digit indicates the intended year in which the course should be taken. Thus 1000 level courses are intended to be taken in the first year and 4000 level courses in the fourth and final year. The course level is important because we want to restrict the recommender system to using early courses.

3.2 Data Preprocessing and Transformation

A variety of data preprocessing steps are required in order for the data to be converted into a form that can be used by the recommender system. Some of the data preprocessing steps relate to parameters for the recommender system, such as the level of courses to be considered when making the recommendations. At the end of these preprocessing steps, the data will be at the student level, not the student-course level, so that each data point completely summarizes a student.

As shown in Figure 1, the distribution of the course majors is heavily skewed, such that there are many very unpopular majors. Including the most unpopular majors in the recommendation system will not be productive and will only degrade the performance of the system. Consequently the system excludes some of the majors from consideration. This is done by applying a threshold based on the total percentage of students that must be covered. Our main results are presented for a threshold of 90%, which corresponds to the top 28 majors (see Figure 1).

For the system to make relevant recommendations, we cannot utilize all of the courses that the student takes, since a recommendation after four years of college would not be useful. Thus the recommender system only uses a subset of the courses.. Our main results are based on core courses taken during the first two years of study, since most students declare a major toward the end of the second year.

A key component of our data are student grades, since we consider student performance in determining which major to recommend. The student major Grade Point Average (GPA) is computed by averaging the grades from all classes in the major. However, after this is computed, it is normalized to form a normalized GPA, nGPA, such that the mean nGPA is 0 and the standard deviation is 1. Students with negative nGPAs perform below average and those with positive nGPAs perform above average.

The recommender system requires the data to be at the student level, not the student-class level. This transformation is accomplished by aggregating all of the student-class records for each student, such that the resulting record encodes the nGPA of every relevant course the student has taken, and also includes the student's major. The relevant classes are determined based on the level and type of course (e.g., Core or non-Core). The resulting record has 63 fields for our standard configuration that utilizes all level 1 and 2 core courses. Our data set now includes 4,562 records (i.e., students). Some of the 7,187 students dropped because only the 90% of the students in the top 28 majors are included, and others are dropped due to having an insufficient number of courses due to transfer from another university or other data inconsistencies.

3.3 Methodology for Determining which Majors to Recommend

Our recommender system utilizes collaborative filtering to decide on the majors to recommend. A nearest-neighbor algorithm is employed to find the students most similar to the student needing the major recommendations. Similarity is based on the transformed records just discussed, and hence is based on similarity in the courses taken and the nGPA values for those courses. The number of nearest neighbors employed is a parameter to the algorithm, as is the specific distance/similarity metric. Previous research has demonstrated that the distance metric has a large impact on performance [2] and this motivated our desire to evaluate a variety of metrics.

The distance metrics used in this study are described in Table 1 (in the table $|x_i|$ represents the length of vector x, the square root of its dot product with itself). The most basic distance metric in Table 1 is cosine distance, which represents the angle between two vectors. That metric serves a good starting point, but has problems because it does not properly account for the overall performance of the students. To see this, note that the cosine distance between two points (1,1) and (4,4) is 0, which means they look perfectly similar; however, if we interpret these points as (not normalized) GPAs, then one student has two D's and the other has two A's, and hence we consider them to be very dissimilar. The metrics specified in Table 1 after Cosine distance have been shown to often outperform other distance metrics, as well as not having the grade-ratio problem described above[2].

Table 1: Distance Metrics

Metric Name	Abbreviation	Formula
Cosine Distance	CosD(x,y)	$1 - \sum_{i=1}^{n} \frac{x_i \cdot y_i}{\lvert x_i \rvert \times \lvert y_i \rvert}$
Clark Distance	ClaD(x,y)	$\sqrt{\sum_{i=1}^{n} \left(\frac{x_i - y_i}{\lvert x_i \rvert + \lvert y_i \rvert} \right)^2}$
Chi Squared Distance	SCSD(x,y)	$\sum_{i=1}^{n} \frac{(x_i - y_i)^2}{\lvert x_i + y_i \rvert}$
Canberra Distance	CanD(x,y)	$\sum_{i=1}^{n} \frac{\lvert x_i - y_i \rvert}{\lvert x_i \rvert + \lvert y_i \rvert}$
Lorentzian Distance	LD(x,y)	$\sum_{i=1}^{n} \ln(1 + \lvert x_i - y_i \rvert)$

The system should incorporate grades, and the cosine distance cares for difference in ratio of grades between students, and therefore a student performing badly throughout will seem similar to a student performing well throughout. Hence, the closest *n* students, with all other students discarded, have their distances rescaled by a scaling function described in Equation 1:

$$CosD'(x,y) = CosD(x,y) \times (1 - \tanh(nGPA_y \times g)) \qquad [1]$$

Where $CosD(x,y)$ is the original cosine distance between students x and y, $nGPA_y$ is the nGPA student y, and g is a scaling factor determining the effect of nGPA on the distance calculation. This results in the student having an updated vote scaled by their performance. The scaling factor serves the purpose of modulating the effect of tanh, with higher values of g increasing the effect of student performance. We use the formula above as students who have high normalized grades have their distances brought closer and those who do poorly have their distances pushed away. This is visualized in Figure 2 below.

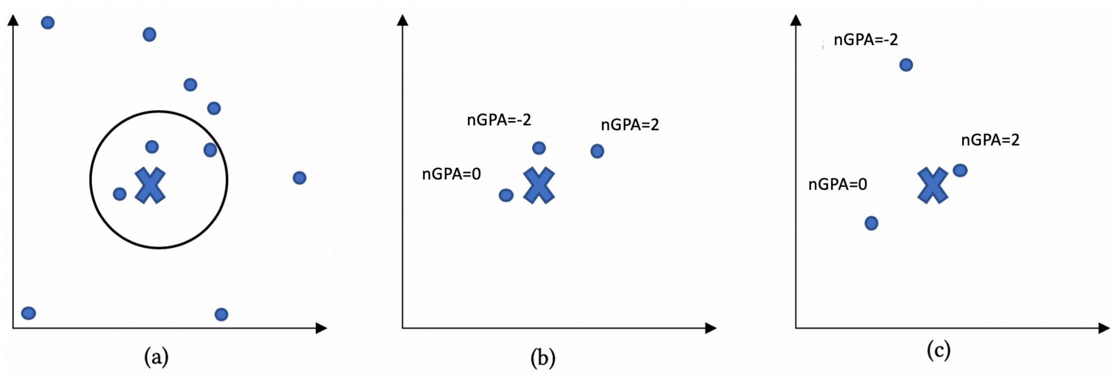


Figure 2 Distance tanh scaling

Figure 2(a) is the initial choice of n nearest neighbors, Figure 2(b) is checking the nGPA of the students, and Figure 2(c) is the student's distance once scaled by 1-tanh($nGPA \times g$). This shows the process of taking the n closest students, scaling their vote by how well they did, and then using these new distances as the final votes each point has.

We are trying to find the majors with the shortest distance to a student, combining distances from nearby students from each potential major. For major M_i, we compute the mean distance $MajorDistance_i$, using the formula in Equation 2.

$$MajorDistance_i = \frac{\sum d_{M_i,j}}{n_i^z} \qquad [2]$$

In the numerator, $d_{M_i,j}$ represents the j[th] distance for a student in major i, n_i is the number of nearby students in major i, and z is a factor between 1 and 2 that rewards a major for having more students. Experiments are provided to assess the impact of different values of z.

As a concrete example, consider the case where we are looking at four nearest neighbors, where three of those neighbors major in Computer Science and one majors in Communications. Further assume that the scaled distance to the three computer science students is 1.0, and the scaled distance for the communications major is 0.8. If z is set to 1.5, then we have the following Major Distances:

$$MajorDistance_{ComputerScience} = \frac{1+1+1}{3^{1.5}} = 0.57$$

$$MajorDistance_{Communications} = \frac{0.8}{1^{1.5}} = 0.80$$

This supports a recommendation of Computer Science, which has the smaller distance due to the larger number of supporting students.

3.4 System Evaluation Metrics

Recommender system evaluation should take into account the domain and how the recommendation will be used. We want to evaluate (1) (weighted) accuracy of predicting the selected major and (2) likelihood of academic success in the predicted major.

Selection prediction accuracy is a common measure for recommender systems. For movie recommendations, this might be measured based on whether they actually followed the recommendation, or perhaps whether the movie is in the top *n* movies they could watch. In the case of our recommender system, since students usually only choose one major, we measure the fraction of students whose actual majors are within the top *n* recommended majors. We refer to this as major recommendation accuracy, or just accuracy for short. But we understand that this is only an approximation of what we would like to assess, since it is possible that the best major for a student could be one that they did not pursue. For example, our system might recommend Physics while the student majored in Mathematics—but it is possible that Physics was the better major for the student. By assessing accuracy based on the top *n* recommendations rather than only the top recommendation, we provide a somewhat more robust and fair measure, but this does not completely address the underlying issue.

Due to a large variance in the number of students within each major, accuracy is not necessarily the most informative metric, similar to why accuracy is generally not the primary evaluation metric for classification problems that include significant class imbalance. Therefore, as in the class imbalance case, we focus on precision and recall, and the F1 score that combines these two measures. However, because there is not one class value (in this case major) of primary importance, we use the mean F1-score, which is provided below, where there are *n* majors, and r_i and p_i are the recall and precision associated with each major.

$$mean\ F1\ score = \frac{1}{n}\sum_{i=1}^{n}\frac{2r_ip_i}{r_i + p_i}$$

Finally, just because we recommended the major that a student selected, does not necessarily mean that this was a suitable major. One way to assess the suitability of a major is via the student's major GPA. We consider performance in a major to be good if the GPA is average or better, which translates to an nGPA ≥ 0. The following two metrics are used to evaluate the performance of the recommender system. The first metric, QOR, is of primary importance.

$$Quality\ of\ Recommendation\ (QOR) = \frac{\sum Major\ Recommended\ \&\ \mathrm{n}GPA \geq 0}{\sum Major\ Recommended}$$

$$Quality\ of\ Not\ Recommended\ (QONR) = \frac{\sum Major\ Not\ Recommended\ \&\ \mathrm{n}GPA \geq 0}{\sum Major\ Not\ Recommended}$$

4 Results

This section presents the results for the recommender system experiments. Section 4.1 provides our main results, which are for our "standard" set of parameter values. Section 4.2 then explores the impact of varying the system parameters. In all cases the experiments were run using leave-one-out cross validation, which means that the recommendation generated for each student is generated using the data for all other students.

4.1 Main Results

The parameter settings used as our standard configuration are provided below. All results in this section are based on these parameter settings.

Distance Metric: CosD', where k =0.5

Number of nearest neighbors: 50

% of Students included: 90

Levels of Study: 1,2 and Only Core

Top N within recommendations to be classified as correct: 5

These results associated with these parameter settings are summarized in Table 2. The results for the recommender system are provided in the last column. The three other recommendation strategies provide useful baselines, in increasing order of sophistication:

Table 2: Recommendation Results The "Random" recommendation strategy randomly selects a major, "Most Common" selects the most common major(s), "Actual Major" selects the major the student actually chose, and "Recommender System" uses the recommendations produced by our system.

Evaluation Metric	Major Recommendation Strategy			
	Random	Most Common	Actual Major	Recommender System
Recommended & nGPA ≥ 0 (QOR)	55%	55%	55%	67%
Recommended & nGPA < 0	45%	45%	45%	33%
Not Recommended & nGPA ≥ 0 (QONR)	55%	55%	55%	44%
Not Recommended & nGPA < 0	45%	45%	45%	56%
Student Major Accuracy	18%	39%	100*	61%
Mean F1-Score	18%	18%	100*	42%

* These values are guaranteed to be 100% based on how accuracy was defined. As discussed earlier, these values are of limited utility since it is not known that the actual student major is the best major for the student.

Table 2 shows that our recommender system performs quite well. The most objectively useful metric is QOR, and our system outperforms all of the baseline strategies in this case—and even substantially outperforms the student's own choice (67% vs. 55%). That means that when our system recommends a major and the student actually choose the major, then the student is more likely to perform well in the major. The metrics associated with what happens when we do not recommend a major are provided for completeness, but are not as important as what happens when we do recommend a major (it is actually a positive that when we do not recommend a major the student's tend to do poorly in it). With respect to the student major accuracy and mean F1-Score, our system significantly outperforms the two simplest baseline strategies. However, it does not outperform and cannot outperform the strategy of choosing the students actual major, since by definition those must be 100%. As mentioned before, these metrics are of limited utility since we cannot say that the actual major that the student chose is the best one. The ability of the system to include the student's actual major in the top 5 recommendations 61% of the time does suggest that the recommender system is making reasonable recommendations. Overall, we can say that if the students follow the system's recommendation, they will generally improve their performance in the major.

A possible explanation for the good performance of the recommendation system with respect to QOR is that its recommendations are more accurate for the high-performing students, who will generally perform well above average. To evaluate this explanation, we compute the Z-value between the initial nGPA distribution and the correctly classified nGPA distribution. Our null hypothesis is that the distributions are similar with a confidence level of $\alpha = 0.1$. As shown below, a Z value of 0.29 is computed, and since this value is below α, we *cannot* reject the null hypothesis that the distributions are different. This provides evidence that our system is not just achieving high QOR values by making accurate recommendations to the high performing students.

$$\boldsymbol{Z} = \frac{\mu_1 - \mu_2}{(\sigma_1 + \sigma_2)^2} = 0.29$$

4.2 Effect of Parameters

The main results that were presented in Section 4.1 were based on five parameter settings. In this section we investigate the impact of these settings by varying each of them and analyzing the results. Some of these changes impact the data that is used for the experiments, such as the level and types of courses upon which the recommendations are based, while others are just parameters that impact the procedure for deciding on which majors to recommend. For simplicity and due to space limitations, we only vary one parameter at a time. The base level parameter settings listed below apply to the experiments in this section, except for the one parameter that is actively being varied.

Distance Metric: CosD', where k = 0.5

Number of nearest neighbors: 10

% Of Students included: 95

Levels of Study: 1,2 and Only Core

Top N within recommendations to be classified as correct: 3

We begin by varying the value of *N* in Top-*N* and the value of *k* in *k*NN. The results are shown in Figure 3, and evaluated with respect to major recommendation accuracy, QOR, and F1 Score.

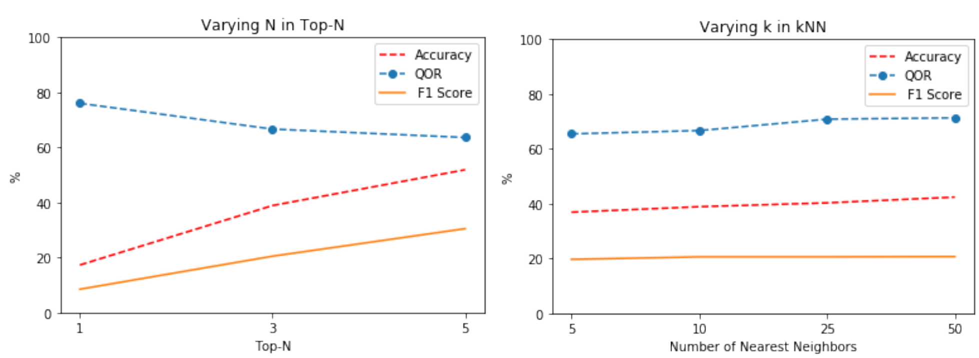


Figure 3: Impact of Top N and Number of Nearest Neighbors on Performance

Figure 3 shows that, as expected, increasing the value of *N* in Top-N, leads to significant increases in accuracy and F1 Score, since with larger N it is easier to obtain a correct recommendation. However this leads to a modest decrease in QOR, so there is a tradeoff between the two. There is no simple way to determine which value of N is best overall, since there is no objective way to assess the relative values of accuracy and QOR in this context. Figure 3 also shows that the results are not especially sensitive to the number of nearest neighbors used by the algorithm, but performance does improve for both accuracy/F1-score and QOR as the value of k increases from 5 to 50. Hence 50 nearest neighbors are recommended. Note that our main results utilize Top-5 and 50 nearest neighbors.

The next set of parameter results involve varying the classes that are utilized in making the recommendations and distance metric used for determining the distances. The results are displayed in Figure 4. The results for the levels of courses are encouraging, since there are no dramatic differences between using core courses for one year, two years, or three years. Thus, it is feasible to issue a recommendation relatively early. Our default configuration uses a values of "Core 1,2", because we feel making a recommendation at the end of the second year is reasonable, and fairly consistent with the practice at our university (our university has a very large core curriculum and that often takes up much of the first two years). We do see that if we use all classes that the student takes during the four years, including non-core classes, then accuracy is increased substantially. However, issuing a recommendation so late in a student's career is clearly not helpful. Aside from this, the non-core courses essentially encode the student's major (e.g., only a chemistry major is likely to take a large number of chemistry classes). That is one reason that we avoid using non-core classes even during the first two years: it may essentially encode the student's decision, assuming the student has already made that decision.

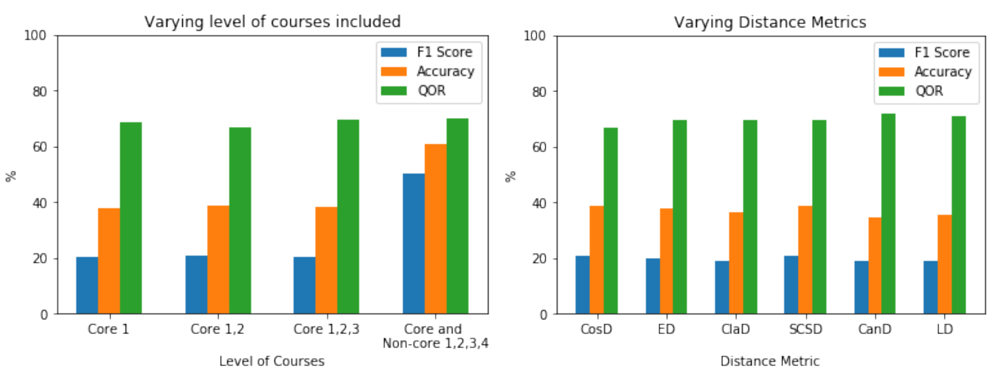


Figure 4: Impact of Courses Included and Distance Metric on Performance

Figure 4 also shows the impact of the different distance metrics. Overall, it demonstrates that the metrics yield generally similar performance. However, there is a clear pattern where some metrics do a bit better on QOR than others, but then perform a bit worse on accuracy and F1 score. These metrics therefore seem to vary a bit on how they tradeoff accuracy of the recommendation and its quality (as measured by QOR).

The results for the final set of parameter values are provided in Figure 5. The leftmost figure shows how the tanh scaling factor, defined in Equation 1 in Section 3.3, affects the trade-off between Accuracy/F1-Score and QOR. As the factor becomes larger, QOR improves to the detriment of the other metrics. This can be associated with weighting better performing students more as the coefficient goes up, which leads to the decrease in accuracy. The right part of the figure shows what happens when we vary the value of *z* from Equation 2 is varied. Increasing the value of *z* generally decreases QOR but increases accuracy and the F1-score. So yet again we see a trade-off between these two metrics.

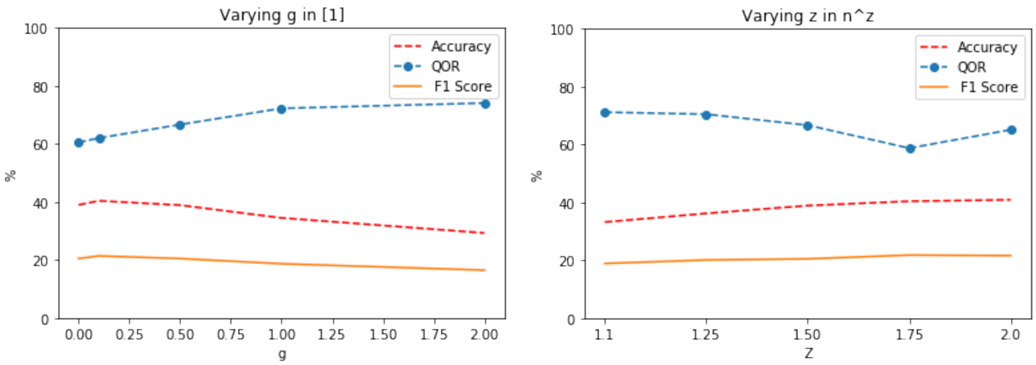
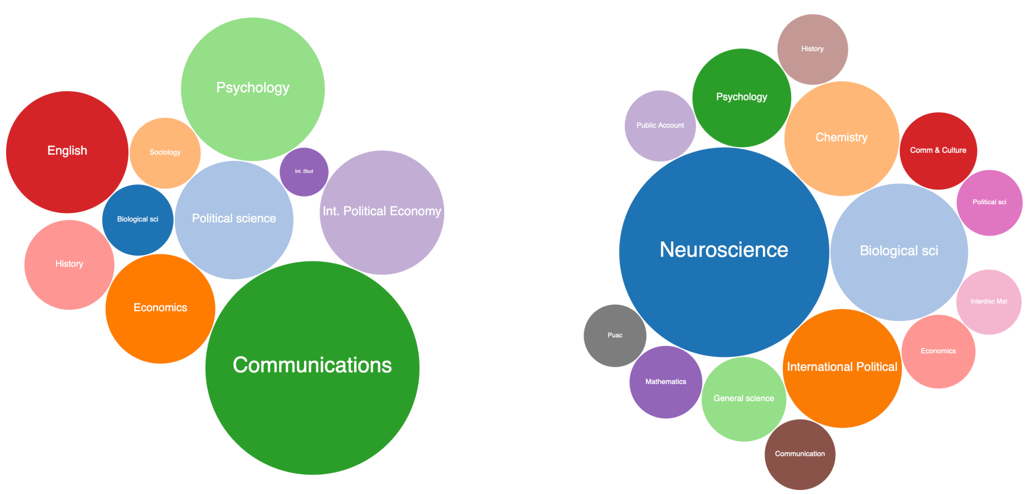


Figure 5: Impact of k in Tanh Adjustment and z in n^z on Performance

5 Majors with Similar Recommendations

This paper focuses on describing and evaluating a system for recommending college majors. However, as a side-effect of its operation, the system can also generate some descriptive data mining results, by showing which majors are close together from a recommendation perspective. That is, if several majors are often recommended together for a student, perhaps with slightly different ranks, then in some sense they are related. This can be useful to college administrators and advisors, and may even provide some insight concerning the relationships between different disciplines. This section visually depicts the proximity of a group of majors from a specified major. In the interest of space we only present figures relative to two majors. These are displayed in Figure 6 for the Communications and Neuroscience majors. In each figure, the proximity to other majors from the designated major is based solely on the size of the other major (i.e., radius of the circle) and is *not* based on its physical distance to the other majors in the figure. The placement of each major in the figure is arbitrary.



(a) **(b)**

Figure 6: Majors similar to (a) Communications and (b) Neuroscience Major Clusters

Figure 6a shows that the Communications major is closest to the Psychology and English Majors. The proximity to English is clear since they two majors do share much in common. (This is even more clear when one recognizes that one specialization within Communications is Journalism.) The connection to Psychology may not be quite as clear, but psychology is not considered a "hard science" and also leads to a Bachelor of Arts degree. With respect to Figure 6b, we see that the four closest disciplines to Neuroscience, which is an interdisciplinary major, are Biology, Chemistry, International Political Economy, and Psychology. Biology and Psychology are both participating department in the Neuroscience major and both play a central role in the major. Chemistry is not quite as central to the major, but neuroscience students must study chemistry to complete the major. Only the connection to the International Political Economy major is a bit odd, but that major is also interdisciplinary and perhaps both attract similar types of students.

6 Conclusion

This paper described and evaluated a system for recommending undergraduate majors. The evaluation was performed on a complete set of student data from a large university, using nine years of data. The recommendation system utilized collaborative filtering using a nearest neighbor method and evaluated a variety of parameter settings and distance metrics. The main results demonstrated that it is possible to generate reasonable recommendations utilizing two years of core curriculum courses, which all students in the university must take. These results showed that 61% of the time the student's actual major was within the top-5 recommended majors, and more significantly, that 67% of the time the student will outperform the average student in the recommended major. More specifically, when students follow the system's recommendation, they are 15% more likely to outperform the average student, then when they just choose their own major. Thus, this system seems well-positioned to be used as a tool to assist students as they explore potential subjects to major in. The ability to predict their future success in the subject may be particularly helpful. The code that implements the recommender system is publicly available [15] and thus represents another contribution of this work.

There are a number of ways in which this research can be extended. First, the methodology can be applied to data from other universities. We encourage other researchers in educational data mining to apply this methodology to their own data, and they are free to use our code to facilitate this process [15]. Also, while this study did explore the use of different parameter settings, not all combinations and all ranges of values were considered, largely due to computation time limits; we expect to explore additional parameter settings in the future.

7 Reproducibility Statement

The data used for this research study includes all of the grades that undergraduate students received in all of their courses over a nine-year period. While every student identifier was replaced with an anonymized version, the data is still too sensitive to share publically, and sharing it could constitute a FERPA (Family Educational Rights and Privacy Act) violation. However, given the nature of this research, the limited access to our data should only constitute a minor concern. We present results only with respect to our own data, and do not make any more general claims about our method or results, nor do we claim to outperform any existing recommendation systems (since there are no similar such systems in this domain). We do provide a public copy of our software [15], which enables others to apply our recommendation system to their own data and generate their own results.

REFERENCES

[1] Asad M Madni, Recommender Systems in E-Commerce, 2014, 2 014 World Automation Congress

[2] V. B. Surya Prasath, Haneen Arafat Abu Alfeilate , Ahmad B. A. Hassanat, Effects of Distance Measure Choice on KNN Classifier Performance - A Review, 2019

[3] J. Ben Schafer, Joseph Konstan, John Riedl, Recommender Systems in E-Commerce, University of Minnesota

[4] Mariela Tapia-Leon,Sergio Lujan-Mora, Recommendation Systems in Education: A Systematic Mapping Study,2018, Proceedings of the International Conference on Information Technology & Systems

[5] E. Garc ́ıa, C. Romero, S. Ventura, C. D. Castro, An architecture for making recommendations to courseware authors using association rule mining and collaborative filtering, User Modeling and User-Adapted Interaction 19 (1-2)

[6] N. Soonthornphisaj, E. Rojsattarat, S. Yim-ngam, Smart E-Learning Using Recommender System, in: International Conference on Intelligent Computing, 518–523, 2006

[7] R. Bekele, W. Menzel, A Bayesian Approach to Predict Performance of a Student (BAPPS): A Case with Ethiopian Students, in: Artificial Intelligence and Applications, Vienna, Austria, 189–194, 2005.

[8] H. Cen, K. Koedinger, B. Junker, Learning Factors Analysis A General Method for Cognitive Model Evaluation and Improvement, in:Intelligent Tutoring Systems, vol. 4053, Springer Berlin Heidelberg, ISBN 978-3-540-35159-7, 164–175, 2006

[9] N. Thai-Nghe, P. Janecek, P. Haddawy, A Comparative Analysis of Techniques for Predicting Academic Performance, in: Proceeding of 37th IEEE Frontiers in Education Conference (FIE’07), Milwaukee, USA, IEEE Xplore, T2G7–T2G12, 2007

[10] N. Thai-Nghe, D. Lucas, A. Krohn-Grimberghe, L Schmidt-Theme, Recommender Systems For Predicting Student Performance, in: Procedia Computer Science 1 (2010) 2811-2819; University of Hildesheim, Hildesheim, Germany

[11] C. Romero, S. Ventura, P. G. Espejo, C. Hervs, Data Mining Algorithms to Classify Students, in: 1st International Conference on EducationalData Mining (EDM’08), Montral, Canada, 8–17, 200

[12] N. Manouselis, H. Drachsler, R. Vuorikari, H. Hummel, R. Koper, Recommender Systems in Technology Enhanced Learning, in: Kantor,P.B., Ricci, F., Rokach, L., Shapira, B. (eds.) 1st Recommender Systems Handbook, Springer-Berlin, 1–29, 2010.

[13] B Bakhshinategh, G Spanakis, O Zaiane, S Elatia, A Course Recommender System Based on Graduating Attributes, in: 9th International Conference on Computer Supported Education

[14] M Hasan, S Ahmed, D Abdullah, S Rahman, Graduate School Recommender System: Assisting admission seekers to apply for graduate studies in appropriate graduate schools, in: International Conference on Informatics, Electronics and Vision (ICIEV), 2010

[15] GitHub Repo of code available at: https://github.com/XXX (address masked to preserve anonymity).